What's new in Database Desktop

See also

In addition to the new Windows 95 look and feel—including support for long file names —this version of Database Desktop adds ease-of-use features:

- Changing object properties is easier with new tabbed dialog boxes.
- New, movable Toolbars add convenience —and you can dock them, or let them float.
- Menu enhancements make it easier to find the commands you need.
- File-handling enhancements use Windows common dialog boxes for convenience and suite compatibility.
- New and improved tools add functionality and convenience.
- Online Help enhancements make it easier to find the information you need when you need it.
- SQL enhancements let you work with SQL tables directly through the Database Desktop user interface and make it easier to query SQL tables.

Toolbar enhancements

See also

You can now move Toolbars, dock them at the sides and bottom of the window, or let them float. For more information, see About Toolbars.

Toolbar tips make it easier to see what a Toolbar button is for. Just point to a button and its description appears.

Menu enhancements

See also

Menus have been redesigned so commands are easier to find. These are some of the more significant changes:

- To define aliases in the Alias Manager, choose Tools|Alias Manager.
- The View menu lets you specify which Toolbars to display and also sets Field View and Persistent Field View. Use the object-specific menu (Table) to toggle in and out of Edit mode (or press F9).
- Changes and additions to the Tools menu make Database Desktop more convenient and easier to use.
- The Windows menu offers vertical or horizontal tiling.

File-handling enhancements

See also

File-handling enhancements include the use of Windows common dialog boxes for convenience and suite compatibility.

For example, you can manage some file operations in the common dialog boxes, such as File|Open. Just right-click a file name for a menu. You can also create directories from these dialog boxes, without having to return to the Windows Explorer. For more information, see About using common dialog boxes.

▪

Tool enhancements

See also

Tools menu changes

Changes and additions to the Tools menu make Database Desktop more convenient and easier to use.

- The Alias Manager is now available on the Tools menu.
- Passwords now appears at the top level of the Tools menu.

New and improved tools

- You can perform multi-table live queries.

▪

Online Help enhancements

See also

Online Help enhancements make it easier to find the information you need.

- The entire Database Desktop User's Guide contents are now included in Help.
- Database Desktop Help is organized into a familiar book-like table of contents that's easier to use. An expanded index and full-text searching offer quick access to online assistance when you need it.

For details, see About the Help system.

▪

SQL enhancements

See also

The following enhancements to SQL support in Database Desktop make it easier to access and query tables on remote and local servers:

- You can now work with SQL tables directly through the Database Desktop user interface.
- The SQL Editor has new features, described in About the SQL Editor.
- You can now do the following in SQL queries:
- Use underlying indexes in live queries
- Constrain updates to satisfy query conditions
- Create calculated fields in live queries
- Database Desktop is ANSI-92 SQL-compatible for remote operations.

▪

Introduction to Database Desktop

See also

Database Desktop lets you create, view, sort, modify, and query data tables in a variety of Paradox, dBASE, and SQL formats.

Edit|Paste Link lets you create live links to data in other applications with DDE (Dynamic Data Exchange).

For details on Database Desktop and its uses, click the Index or Contents button of this Help window and browse or search for the subject you are interested in.

If you're familiar with other versions of Database Desktop, see What's new in Database Desktop for a description of enhancements in this version.

About links

See also

You link tables by defining a relationship between a field of one table and a field of another table. What you are really doing is telling Database Desktop that the values in a field of one table match values in an indexed field of another table. Because the values match, Database Desktop has a way to relate the data of one table to the data of the other.

One or more of the fields used to define the link must be indexed because Database Desktop must match values from the fields. Without an index, these values could be anywhere. When an index is used, Database Desktop has a maintained file that lists the locations of all the records in the table. Database Desktop can then find and link the records quickly and efficiently.

About keys and indexes

See also

You link tables by defining a relationship from the field of one table to the field of another table. These fields must meet certain requirements. The most important requirement is that one or both of these fields have indexes. An index is a file that Database Desktop uses to keep track of the location of records in a table. This makes it easy for Database Desktop to

- Maintain a sorted order of a table
- Group like values together

Both Paradox and dBASE tables can contain indexes to specify the order in which records are accessed. However, the way indexes work is different for Paradox and dBASE tables.

In Paradox tables, primary indexes are sometimes called keys.

For more information, see About keys and indexes in tables.

▪

Keys

See also

A Paradox table can have many indexes defined, but you usually identify one of them as the primary index. In Paradox tables, the primary index is called the key. A table that has a key defined is said to be a keyed table.

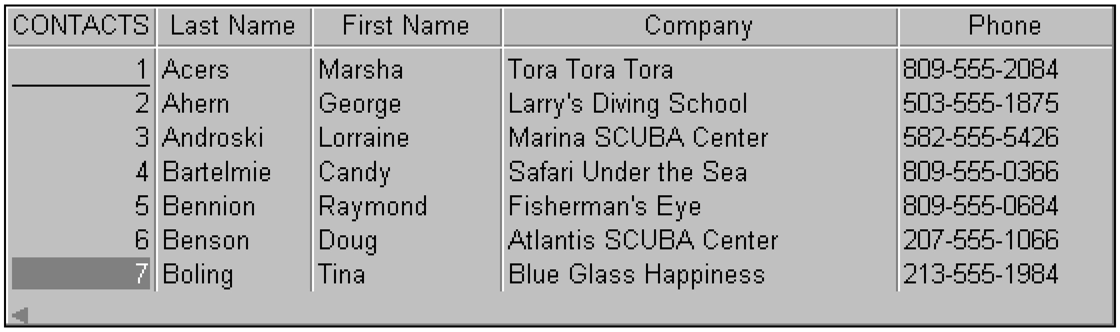
When you create a key, Database Desktop enforces rules about the data that can be contained in the keyed field(s).

- Each value in the field must be unique. This ensures you do not have duplicate records in the table.

  Note: You can leave only one record's key blank. Database Desktop considers all subsequent blanks to be duplicates and does not accept records containing them.
- The key establishes the default sort order for the table. Database Desktop sorts the table's records based on the values in the field(s) you define as the table's key.

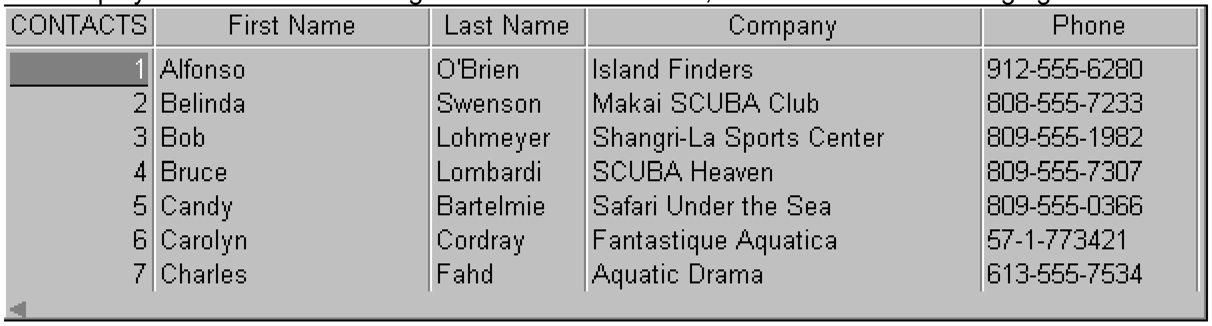
If you define a key on a table that already contains data, Database Desktop moves the records of the table into the correct sort order. The physical location of records is determined by sorting the values of the keyed field(s) in an ascending order (A to Z and 0 to 9). New records you add are moved to their correct position in the sorted table.

For example, if you create a key on the Last Name field of the sample Contacts table, you are telling Database Desktop to organize the table by the values in the Last Name field, as shown in the following figure.



CONTACTS	Last Name	First Name	Company	Phone
1	Acers	Marsha	Tora Tora Tora	809-555-2084
2	Ahern	George	Larry's Diving School	503-555-1875
3	Androski	Lorraine	Marina SCUBA Center	582-555-5426
4	Bartelmie	Candy	Safari Under the Sea	809-555-0366
5	Bennion	Raymond	Fisherman's Eye	809-555-0684
6	Benson	Doug	Atlantis SCUBA Center	207-555-1066
7	Boling	Tina	Blue Glass Happiness	213-555-1984

If you prefer to organize the table by first names, you can make First Name the key. Database Desktop then displays the records according to the value in that field, as shown in the following figure.



CONTACTS	First Name	Last Name	Company	Phone
1	Alfonso	O'Brien	Island Finders	912-555-6280
2	Belinda	Swenson	Makai SCUBA Club	808-555-7233
3	Bob	Lohmeyer	Shangri-La Sports Center	809-555-1982
4	Bruce	Lombardi	SCUBA Heaven	809-555-7307
5	Candy	Bartelmie	Safari Under the Sea	809-555-0366
6	Carolyn	Cordray	Fantastique Aquatica	57-1-773421
7	Charles	Fahd	Aquatic Drama	613-555-7534

If you use more than one field in a key, the index is called a composite key.
For more information, see About keys and indexes in tables.

Composite keys

See also

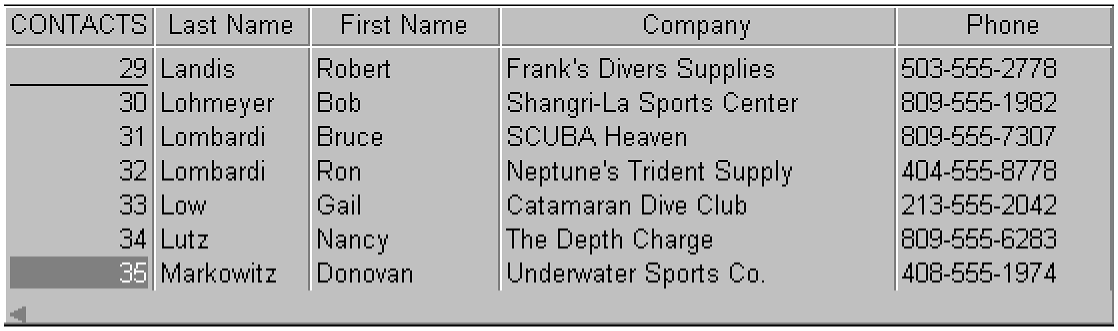
You can create a key on a single field or group of fields. When you specify a group of fields as a table's key, the group is called a composite key.

Database Desktop allows duplicate values in individual fields of a composite key, as long as values are not duplicated across all fields of the key. The fields of the key, taken as a whole, must identify each record as unique.

For example, the Contacts table may have several entries with the last name Lombardi. Likewise, it may have many entries with the first name Ron. Neither of these fields (Last Name or First Name) is enough to identify a record as unique. But the combination of them may be. (There may be only one Ron Lombardi.) So the key for the Contacts table could be a composite of Last Name and First Name.

Of course, even this may not be enough. It is entirely possible to have duplicate first and last names in the table (like several entries for John Smith). It may be a good idea to include another field of the table in the composite key. You must always include enough fields in a composite key to ensure the uniqueness of each record of the table. If you cannot reasonably expect a composite key to handle all cases of duplicate data, it is a good idea to define an identification field that identifies one and only one record of the table. Customer No in the sample Customer table is such an identification field.

When you create a composite key, Database Desktop creates a primary composite index, which organizes the records by the first field of the key (according to the table's structure), then the next field, and so on. The following figure shows the Contacts table with a composite key made up of the Last Name and First Name fields.



CONTACTS	Last Name	First Name	Company	Phone
29	Landis	Robert	Frank's Divers Supplies	503-555-2778
30	Lohmeyer	Bob	Shangri-La Sports Center	809-555-1982
31	Lombardi	Bruce	SCUBA Heaven	809-555-7307
32	Lombardi	Ron	Neptune's Trident Supply	404-555-8778
33	Low	Gail	Catamaran Dive Club	213-555-2042
34	Lutz	Nancy	The Depth Charge	809-555-6283
35	Markowitz	Donovan	Underwater Sports Co.	408-555-1974

For more information, see About keys and indexes in tables.

▪

Indexes

See also

When you create an index, Database Desktop creates one or more files that contain the indexed field's values and their locations. Database Desktop refers to the index file when locating and displaying the records in a able. This is true of both primary indexes (keys) and secondary indexes.

For details, see About keys and indexes in tables and About secondary indexes.

About Database Desktop objects

See also

In Database Desktop, the database components that store, display, retrieve, and present data are called objects. The main objects you work with in Database Desktop are tables, queries, and SQL files.

Database Desktop uses objects to store, display, and present information.

Objects include

- Files on a disk
- Tables, queries, and SQL files.

Database Desktop uses object icons to represent objects when they are minimized.

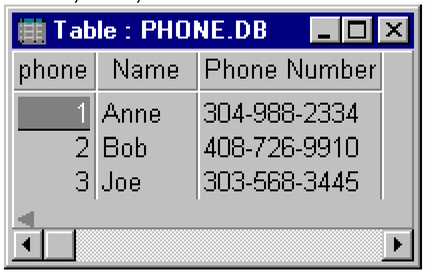
Each object has a different extension. For a list, see File extensions for Database Desktop objects.

Tables

See also

Database Desktop stores data in tables. Tables have rows and columns. Each row contains information about a particular item (like a person, place, or thing). This is called a record. Each column contains one category of the data that makes up a record. This is called a field.

The simple table below is named Phone.db. It has two fields▪Name and Phone Number
▪and three records
▪Anne, Bob, and Joe.



phone	Name	Phone Number
1	Anne	304-988-2334
2	Bob	408-726-9910
3	Joe	303-568-3445

Temporary tables

See also

Certain Database Desktop operations create temporary tables that last only until you change your private directory or end the Database Desktop session.

Database Desktop stores all temporary tables in your private directory (see About directories and aliases for more information). You can edit and query a temporary table as you would any other table. If you want to save one of these tables, you must rename it (see About renaming objects).

Caution: You should not use any reserved temporary table name as the name of an object you create. If you do use a temporary table name, Database Desktop deletes your object when you change your private directory or end the Database Desktop session.

The following table lists the temporary tables Database Desktop creates when performing certain operations. Database Desktop places these tables in the private directory.

Name	Created during	Contains
Answer	Query	Results from a query
Changed	CHANGETO query or Add operation (update)	Unchanged copy of changed records
Deleted	DELETE query	Deleted records
Errorchg	CHANGETO query	Records that could not be changed
Errordel	DELETE query	Records that could not be deleted
Errorins	INSERT query	Records that could not be inserted
Inserted	INSERT query	Inserted records
Keyviol*	Restructure or Add operations (append)	Records with duplicate key values and records that violate referential integrity rules
Problems*	Restructure or Import operations	Unconverted records

* If you perform more than one operation that results in this temporary table within one session, Database Desktop creates additional temporary tables with the same name and numbers them. For example, Keyviol1, Keyviol2, and so on.

▪

Queries

See also

A Database Desktop query is a question you ask about the data in your tables. You can use queries to

- Find or select data from a table
- Combine data from more than one table
- Perform calculations on the data in a table

Database Desktop gives you a simple, yet powerful, way to ask questions about a table's data. In the Database Desktop Query window, you choose which tables you want to ask questions about. Then you enter an example of the data you want, and Database Desktop gives you an answer based on your example. This is called query by example (QBE).

Database Desktop provides powerful live query views that let you define and run a query that generates a live editable view of the data you described in the query. When you edit the live query view, you actually change the data in the table you queried. Live query views give you a simple way to limit your view of data to just what you need to work with.

For more information on queries, see About queries.

SQL Files

See also

An SQL file is an object that contains code you write in SQL (Structured Query Language). For more information about using SQL with Database Desktop and about using Database Desktop to work with remote data, see About SQL.

You can use the SQL Editor to write SQL code to perform operations on remote data using Borland SQL Links. You can also write query scripts using SQL that you can run on local Paradox or dBASE data. For more information about using the SQL Editor, see About the SQL Editor.

▪

File extensions for Database Desktop objects

See also

The following table lists the file extensions used by Database Desktop.

Extension	Type of object
.CFG	Configuration file
.DB	Paradox table
.DBF	dBASE table
.DBT	Memos for a dBASE table
.FAM	Database Desktop's listing of related files (like a table's .TV file)
.INI	Configuration file
.LCK	Lock file
.MB	Memos for a Paradox table
.MDX	Maintained index of a dBASE table
.NDX	Non-maintained index of a dBASE table
.PX	Primary index of a Paradox table
.QBE	Saved query
.SQL	Saved SQL file
.TV	Table view settings for a Paradox table
.TVF	Table view settings for a dBASE table
.TVS	Table view setting for SQL data
.VAL	Validity checks and referential integrity for a Paradox table
.Xnn	Secondary single-field index for a Paradox table, numbered
.Ynn	Secondary single-field index for a Paradox table, numbered
.XGn	Composite secondary index for a Paradox table
.YGn	Composite secondary index for a Paradox table

▪

About the Database Desktop window

See also

The Database Desktop window is the first thing you see when you start Database Desktop. It is the primary Database Desktop workspace. In this window, you can

- Manage tables
- Create and run queries
- Create and run SQL statement files

Each type of major object in Database Desktop (like tables or queries) appears in its own type of window. For example, tables always appear in a Table window, and queries always appear in a Query window. For a list of Database Desktop object windows, see Database Desktop window and child windows.

The Database Desktop window and child windows

See also

The Database Desktop window is the parent window in Database Desktop. All other windows are child windows, meaning that they have some degree of independence, but cannot exist alone.

Each type of object in Database Desktop (like tables, queries, or SQL files) appears in its own type of window. Each type of window has some specialized commands that apply only to that type. All commands and features of the Database Desktop window remain available in child windows.

The following are the object windows available in Database Desktop:

Query window
SQL Editor
Table window

About Toolbars

See also

Below the menu is a collection of buttons and tools called the Toolbar. The buttons available on the Toolbar depend on the Toolbar type and the kind of object active on the Database Desktop window. For example, if a table is active, the Standard Toolbar buttons will be those that help you perform tasks with a table. Many Toolbar buttons provide quick equivalents to menu commands or keystrokes. Others provide handy ways for you to navigate through your data.

From within Database Desktop, to get quick help on what a tool or button does, point to it. Database Desktop displays a description of the button next to it and in the status bar.

Database Desktop Toolbars can be moved away from their standard position near the top of the Database Desktop window. You can dock them at either side or the bottom of the Database Desktop window, or let them float undocked. For instructions, see To move a Toolbar.

You can display more than one Toolbar on the Database Desktop window. For instructions, see To display additional Toolbars.

The Standard Toolbar

This is the default Toolbar which usually appears immediately below the menus. This Toolbar displays buttons and tools which are shortcuts to menu commands for the current active window. As you change the focus from one window to another, the Toolbar changes to provide buttons that match the window.

The Global Toolbar

This Toolbar displays buttons and tools which are shortcuts to commonly-used menu commands that are not window-specific such as saving a file, opening a file, and so on. Therefore, its buttons do not change when the active window changes. This means that some buttons on this Toolbar will not have any effect if the currently selected window does not support that action.

This Toolbar does not appear by default. To display the Global Toolbar, see To display additional Toolbars.

To move and dock Toolbars

See also

You can move a Toolbar from the top of the Database Desktop window and dock it at either side of the window or the bottom. If you prefer, you can let the Toolbar float, without docking it. You can then drag the top of the floating Toolbar to move it where you want it.

To move a Toolbar,

- Click within it (but not on a button) and drag it to a new location.

To dock a Toolbar,

- Drag it toward an edge of the Database Desktop window. Move it toward the edge until a dashed outline appears. Release the mouse button when the outline touches the edge.

Note: You can display more than one Toolbar. For instructions, see To display additional Toolbars.

To display additional Toolbars

See also

You can choose to display more than one Toolbar, or to display another Toolbar instead of the one that appears by default.

To change the Toolbar display,

1. Choose Edit|Preferences and click the Toolbars tab to display the Toolbar page of the Preferences dialog box.
2. Choose one or more Toolbars to display:

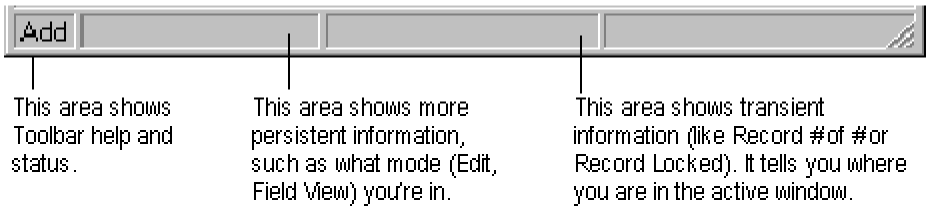
- Standard: The default Toolbar for each child window in the Database Desktop window.
- Global: A Toolbar with buttons to display each Database Desktop object: tables, queries, and SQL files.

When you close Database Desktop, these settings are saved for the next session.

About the Database Desktop status bar

See also

The status bar displays information about Database Desktop and actions you perform in Database Desktop. The following figure shows what different areas of the status bar typically display.


Add
This area shows Toolbar help and status.
This area shows more persistent information, such as what mode (Edit, Field View) you're in.
This area shows transient information (like Record # of # or Record Locked). It tells you where you are in the active window.

To exit Database Desktop

See also

You can exit from Database Desktop in a variety of ways:

- Choose File|Exit
- Choose Close from the window Control menu
- Double-click the window Close box
- Press Alt+F4

▪
About directories and aliases

See also

Working directory

A Database Desktop working directory is the directory Database Desktop uses by default to open and save files. The working directory controls which files are displayed in dialog boxes during open and save operations.

When you install Database Desktop on a local drive (not a network drive), Database Desktop creates a directory named WORKING below the system directory. This is your default working directory. You can change it later, if you want. (See To change your working directory for details.)

You'll probably find it convenient to you use working directories to organize your files. Then, when you want to use the files in a specific directory, you can make it your working directory. For example, if you're working with tables and queries in a directory named C:\DATAFILES\BUDGET, you could change your working directory to C:\DATAFILES\BUDGET.

Private directory

In a multiuser environment, you need a place to put your temporary objects. You need to store temporary tables, such as Answer or Inserted, in a nonshared directory, or other users could overwrite them. All Database Desktop users need their own private directory when they run Database Desktop.

Your default private directory is PRIVATE, created below the main Database Desktop directory on your hard drive, or on your network home directory if you have no hard drive. You can change to another private directory if you want. See To specifiy a private directory for details.

Aliases

An alias is a name you can assign as a shortcut to a directory. By default, your working directory has the alias :WORK: and your private directory has the alias :PRIV:.

There are two kinds of aliases:

- public aliases
- project aliases.

See About aliases for more information.

Note: When you change any of these directory or alias settings, Database Desktop automatically saves the changes.

To change your working directory

See also

Your Database Desktop working directory is the default data directory Database Desktop uses to open and save files.

1. Choose File|Working Directory. Database Desktop opens the Set Working Directory dialog box.
2. Enter the location (full path) of the directory you want in the Working Directory text box, or choose Browse to open the Directory Browser. You can choose any directory, path, or alias from the Directory Browser. The directory you choose appears in the Working Directory text box. You can also choose an alias from the Aliases drop-down list.
3. Choose OK.

Note: Database Desktop assigns your working directory the temporary alias **:**WORK**:** (even if it already has another alias name).

If you create a project alias, Database Desktop creates a file called PDOXWORK.CFG and stores it in your working directory. This file contains all project aliases (public aliases are stored in IDAPI32.CFG).

To specify a private directory

See also

To specify your private directory,

1. Choose File|Private Directory.
2. Specify a private directory in the Private Directory text box. You can type it in or choose Browse to display the Directory Browser.

   Database Desktop assigns the **:**PRIV**:** alias to your private directory.

If you do not specify a private directory, Database Desktop uses the PRIVATE directory, which is installed below your system directory when you install Database Desktop on a local (non-network) drive. If you have no local hard disk, the network home directory on the file server should be used as the private directory.

Note: When you change private directories, Database Desktop releases any locks you have placed on any tables and deletes all your temporary tables. Make sure you do not need any of your temporary tables before you change private directories.

▪

About aliases

See also

An alias is a name you can assign as a shortcut to a directory.

At any time, you can use the Alias Manager dialog box to:

Create a new alias

Modify an existing alias

Remove an alias

Example

Suppose you have a collection of tables, text files, and graphics all in one directory. This collection of files is located in a directory called C:\DATAFILES\PROJECTS\NEW\PLANNER. Using the Alias Manager dialog box, you can give that full path a name▪an alias. For example, if you create an alias for this directory called **:**PLAN**:**, you can then simply use **:**PLAN**:** instead of the full directory path when you need files from C:\DATAFILES\PROJECTS\NEW\PLANNER.

Advantages

Aliases give you several advantages:

- You avoid typing long path names.
- Using the alias, you can connect to or disconnect from your remote database server.
- You can change the definition of an alias at any time. All Database Desktop objects that refer to the alias automatically refer to the new definition of the alias.

Kinds of aliases

There are two kinds of aliases: public aliases and project aliases. See Public and project aliases for more information.

▪

Public and project aliases

See also

In Database Desktop you can create aliases that are available from all directories or specific to a working directory:

- Public aliases are stored in the BDE configuration file. They are available from any working directory and visible to any application that uses BDE (the Borland Database Engine).
- Project aliases are stored in the PDOXWORK.CFG file in the working directory. They are available only when you are using Database Desktop and are in the working directory you created them in.

For example, if your working directory is C:\PROGRAM FILES\BORLAND\SAMPLE and you are creating an alias named CONNECT for a directory called C:\DATAFILES\CONNECT directory, you have two choices:

- A public alias that is available from Database Desktop or any other application that uses BDE (for example, an application developed using ObjectPAL). If you create CONNECT as a public alias, you will see :CONNECT: in the Drive (Or Alias) list of all file selection dialog boxes.
- A project alias that is available only when C:\PROGRAM FILES\BORLAND\SAMPLE is your working directory. If you change working directories or use other applications that use BDE, you will not see the :CONNECT: alias.

Whenever you change working directories, Database Desktop unloads all project aliases associated with the old working directory and loads those project aliases that are specific to the new working directory. Public aliases are available from any working directory.

If a project alias has the same name as a public alias, Database Desktop does not load the project alias.

To create a new alias

See also

You can create aliases for local or network directories, or for remote databases. See Alias Manager dialog box (SQL Link) for information specific to Borland SQL Links and creating an alias for a remote database.

Use the Alias Manager dialog box to create new aliases. To open this dialog box, choose Tools|Alias Manager in the Database Desktop window or Table window.

To create a new alias,

1. Choose New in the Alias Manager dialog box.
2. Type the name (alias) you want to give the directory in the Database Alias text box.
3. Choose the driver you want from the Driver Type drop-down list. The Driver Type drop-down list shows all the drivers you are connected to. To create a database of Paradox and dBASE tables, choose STANDARD.
4. Enter the full path of the directory location, including the drive letter, in the Path text box. (You can choose Browse to open the Directory Browser and select the path from there.)
5. Choose whether this alias will be public or project-specific. Check Public Alias if you want the alias to be available no matter which directory you are working in.
6. Choose Keep New if you want to keep the alias but do not want to close the dialog box. The Keep New button becomes the New button. You can then create another alias. If you want the alias you just created to be temporary (exist only until you exit Database Desktop), choose OK and do not proceed to step 7.
7. To save the alias you have created, choose Save As. Database Desktop opens the Save File As dialog box. By default, Database Desktop stores saved public aliases in IDAPI32.CFG and project aliases in PDOXWORK.CFG. You are prompted to overwrite the existing .CFG file. This is OK; when you overwrite, Database Desktop appends the new alias without changing any existing configuration settings. You can remove the alias from the .CFG file at any time (using the Alias Manager dialog box).
8. Choose OK. The Alias Manager dialog box closes. If you choose OK before saving the alias you have created, Database Desktop prompts you to save the alias in the appropriate .CFG file.

Tip: To create an alias similar to one you already have, select the appropriate alias from the drop-down list. Select New, then make your changes to the alias and choose Keep New to save.

To modify an alias

See also

You can change an alias definition using the Alias Manager dialog box.

To modify an alias,

1. Choose Tools|Alias Manager to open the Alias Manager dialog box.
2. In the Alias Manager dialog box, choose the alias whose path you want to change from the Database Alias list. Then type the new path in the Path text box.
3. Choose Save to overwrite the existing alias, or Save As to create a new alias.

To remove an alias

See also

To remove an existing alias,

1. Choose Tools|Alias Manager to open the Alias Manager dialog box.
2. Select the alias you want to remove from the Database Alias drop-down list.
3. Choose Remove.
4. When you choose OK, Database Desktop prompts you to save the change in the appropriate .CFG file.

About manipulating objects

See also

You can work with Database Desktop objects in many ways. For example,

- You can select windows and table columns, then use the mouse to move or resize them. See To manipulate objects directly with the mouse for details.
- You can right-click most objects to set their properties and perform other operations, such as copying or renaming them. For details, see About copying objects and About renaming objects.
- You can use choose commands from the menu bar to perform many of the same operations available with right-clicking, and more.

To manipulate objects directly with the mouse

See also

Although you can perform most functions in Database Desktop using the keyboard, you should take advantage of the speed and flexibility of the mouse. A mouse is not required for data entry; but it is required for many operations in designing documents. Virtually all mouse actions start with selecting an object. See To select objects with the mouse for instructions.

- Left-click selects an object or activates a command. In tables, you can move to a field by positioning the pointer on it and clicking the left mouse button.
- Drag moves an object or changes its size, shape, or position:
- To resize a table column, drag the column border to where you want it.
- To move a table column, position the mouse at the top of the column and drag it to its new position.
- Right-click displays a menu so you can work with an object or change its properties.

To select objects with the mouse

See also

You can use the mouse to select fields in tables and queries and text in SQL statements.

To select	Do this
A field	Point and click.
Specific text	Point and click, then click to place the insertion point and drag to select the text.

For information on selecting more than one item at a time in lists, see To select from lists.

To select from lists

See also

In some Database Desktop lists, you can select more than one item at a time.

To select	Do this
A single item	Click it.
A contiguous block	Click the first item, then Shift+click the last item.
Scattered items	Ctrl+click (hold down Ctrl while you click the items).

About using common dialog boxes

See also

Many dialog boxes in Database Desktop have common features. These include a file list, a Save In or Look In drop-down list, a Files Of Type drop-down list and an Alias drop-down list. Dialog boxes that you use to open and save files are good examples of this kind of dialog box.

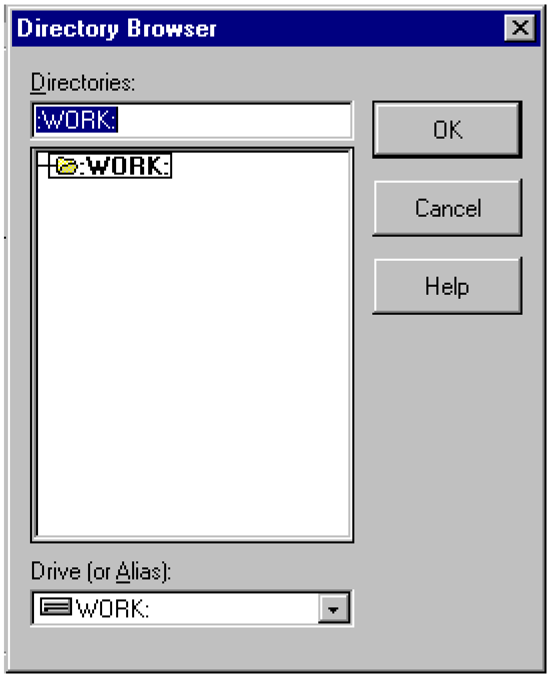
Choose any of the icons beside the Save In or Look In list to navigate, create a folder, or change the display of folders.

To view a common dialog box, choose File|Open, choose Table, and click the Help button that appears in the Open Table dialog box. The Open Document dialog box help topic describes how to use that dialog box. The same general techniques apply to the other common dialog boxes as well.

About the Directory Browser

See also

Dialog boxes that don't have a Save In or Look In drop-down list (such as the Alias Manager dialog box) have a Browse button. When you choose Browse, Database Desktop opens the Directory Browser:



If the directory you want isn't available in the Directories list, pull down the Drive (Or Alias) drop-down list to locate it. The Directory Browser Help topic contains more information.

About browsing files

See also

Some file selection dialog boxes have a Browse button that opens the Select File dialog box instead of the Directory Browser. The Select File dialog box is one of the common dialog boxes that lets you choose a directory and a file. If you have defined aliases for directories, you can use them to browse in the Select File dialog box.

▪

About navigating and editing

See also

You can use keys and the mouse to navigate through Database Desktop tables.

- The navigation buttons in the Standard Toolbar helps you move between records.
- There are many keyboard shortcuts to simplify navigation and editing. These topics list many of them:

  Data entry shortcuts

  Navigation and selection keys

  Keys used in Edit mode

  For a complete list, see About keyboard commands.

About saving Database Desktop objects

See also

Different Database Desktop objects are saved in different ways.

Tables

You don't have to save table data; Database Desktop does this automatically. See To save table data and properties for more information.

Queries and SQL statements

You can use File|Save or Save As to save queries and SQL files. For instructions, see To save queries and SQL files.

Copying and renaming

You can use File|Save As to copy and rename many Database Desktop files, but for best results, use Copy or Rename instead. For more information, see About copying objects and About renaming objects.

To save table data and properties

See also

To save table data,

You don't use the File|Save or Save As commands; Database Desktop automatically saves the data you enter as soon as you leave each record.

To save table properties,

- Choose Table|Table View Properties|Save from the Table window. (If you make changes to a table's properties and don't save them, Database Desktop prompts you to save them when you close the table.)

To save queries and SQL files

See also

To save a newly created query or SQL file,

1. Choose File|Save.

   The Save File As dialog box appears.
2. Type the name of the file you're saving in the File Name text box. You don't need to type an extension; Database Desktop recognizes the type of file you're saving from the Save As Type specification. Database Desktop saves the file when you choose Save.
3. Database Desktop saves the file in the working directory unless you specify otherwise by choosing a different directory in the Save In drop-down list or the Alias drop-down list.

To save existing Database Desktop files,

To save existing Database Desktop files,

- Choose File|Save.

After you've named a file for the first time, choosing Save doesn't open a dialog box; it simply saves the active file.

To save a file under a different name

See also

To save a file under a different name,

- Choose File|Save As to display the Save File As dialog box.

Use it as though you were saving the file for the first time.

Note: For best results, use Copy or Rename instead of Save As to copy or rename Database Desktop objects. For more information, see About copying objects and About renaming objects.

▪

About copying objects

See also

You can copy tables, queries, SQL files, and text files from within Database Desktop. For instructions, see To copy objects.

For best results, always use the Database Desktop Copy utility to copy Paradox or dBASE tables and other Database Desktop objects. Using the DOS COPY command or the Windows Explorer may not copy all related files that make up a table (for example, the files containing a table's primary index, secondary indexes, validity checks, or BLOB data). The Database Desktop Copy command, however, copies all files correctly.

When you copy a table, Database Desktop copies both its structure and the data contained in it. Database Desktop also copies the table's

- Key (primary index)
- Secondary index(es) (except .NDX files on dBASE tables)
- Validity checks
- see Copying referential integrity
- Table properties (as you've set them in the Table window)

Note: These elements are copied only when you copy the table to another table of the same type. That is, they are copied only when you copy a Paradox table to another Paradox table or a dBASE table to another dBASE table.

Copying on a network

See also

When you copy a table, Database Desktop must acquire a read lock on the original table and an exclusive lock on the copy. This means

- No user can change the contents or the structure of the table you're copying during the Copy operation.
- If you copy to an existing table, there can be no locks open on that table.

If there is a record lock, write lock, or exclusive lock on the table you're copying, you won't be able to make the copy until the lock is removed.

Note: Windows lets you open several instances of the same table at the same time, so you could be considered another use of the table, preventing the records from being copied. Be sure to close the table window and any of its associated-document windows before using Copy.

▪

Copying referential integrity

See also

When you define referential integrity, you create a parent/child relationship between two tables.

- If you copy the parent table, Database Desktop doesn't copy the referential integrity.
- If you copy the child table, Database Desktop copies the referential integrity. This means the copied table must meet the requirements of the referential integrity. To delete the referential integrity, you must restructure the table.
- Both tables in the referential integrity relationship must be in the same directory. When you copy the child table to a different directory you break the referential integrity link.

For more information on referential integrity, see About referential integrity.

To copy objects

See also

For best results, use the Database Desktop Copy command instead of using the DOS COPY command or Windows Explorer to copy Database Desktop objects, such as tables.

To copy an object,

1. Choose Tools|Utilities|Copy, then choose the name of the file to copy in the Copy dialog box and choose Open.

   Paradox opens the Copy <file name> To dialog box.
2. Choose a file from the list to copy to, or type the name of the file to create when you save the copy.
3. Choose Save to save a copy of the selected source file.

▪

Copying to a different table type

See also

You can copy a Paradox table to a dBASE table, or a dBASE table to a Paradox table, by typing the file extension you want (.DB for Paradox and .DBF for dBASE) for the copied table. For example, if you want to copy the Paradox Customer table to a dBASE Customer table, type `CUSTOMER.DBF` as the name of the copied table.

If the new dBASE table contains no production index (.MDX file), no float number field type, and no memo field type, Database Desktop creates a dBASE III+ table. If the dBASE table contains an OLE or binary field, Database Desktop creates a dBASE for Windows table. Otherwise, Database Desktop creates a dBASE IV table.

Database Desktop automatically changes field types when you change table types. For a list of field conversions and side effects, see:

- Copying from Paradox to dBASE tables
- Copying from dBASE to Paradox tables

Copying from Paradox to dBASE tables

See also

For general information on copying from a Paradox table to a dBASE table, see Copying to a different table type.

Database Desktop automatically changes field types when you change table types. The following table shows what to expect when you copy from a Paradox table to a dBASE table.

From Paradox type	To dBASE type	Side effects
Alpha	Character	
Number	Number	Assigns size (20) and dec. (4)
Money	Number	Assigns size (20) and dec. (4)
Short	Number	Assigns size (6) and dec. (0)
Long Integer	Number	Assigns size (11) and dec. (0)
BCD	Number	Assigns size (20) and dec. (4)
Date	Date	
Time	Character	Assigns size (8)
Timestamp	Character	Assigns size (30)
Memo	Memo	
Formatted memo	Memo	Formatting is lost
Graphic	Binary	
OLE	OLE	
Logical	Logical	
Autoincrement	Number	Assigns size (11) and dec. (0)
Binary	Memo	Data cannot be displayed
Bytes	Memo	Data cannot be displayed

If the new dBASE table contains no production index (.MDX file), no float number field type, and no memo field type, Database Desktop creates a dBASE III+ table. If the dBASE table contains an OLE or binary field, Database Desktop creates a dBASE for Windows table. Otherwise, Database Desktop creates a dBASE IV table.

If in the BDE Configuration Utility, the Level parameter for the dBASE driver is set to 4 instead of 5, graphic and OLE Paradox fields fields convert to dBASE memo fields. Also, bytes fields cannot be converted.

Copying from dBASE to Paradox tables

See also

For general information on copying from a dBASE table to a Paradox table, see Copying to a different table type.

Database Desktop automatically changes field types when you change table types. The following table shows what to expect when you copy from a dBASE table to a Paradox table.

From dBASE type	To Paradox type	Side effects
Character	Alpha	
Float	Number	Removes size
Number	Number	Removes size
Logical	Logical	
Date	Date	
Memo	Memo	Adds size (1)*
OLE	OLE	
Binary	Graphic	

*Paradox assumes the data in the dBASE memo is in text form. If the memo contains a different type of data, you should use the Add utility and add the memo to the appropriate Paradox BLOB field type.

About renaming objects

See also

You can rename tables, queries, and SQL files from within Database Desktop.

Always use the Database Desktop Rename command to rename tables from within Database Desktop. Using the DOS RENAME command or the Windows Explorer may not rename all related files that make up a table (for example, the files containing a table's primary index, secondary indexes, validity checks, or BLOB data). The Database Desktop Rename command renames all files correctly.

To rename an object, choose Tools|Utilities|Rename. For instructions, see To rename objects.

Tip: Be careful when renaming tables. Once renamed, a table cannot be found by associated documents. Queries that refer to a table under one name will not be bound to the table under its new name. The next time you open an unbound object, Database Desktop asks you to supply the name of the table to which you'd like it to be bound.

▪

Rules for renaming objects

See also

Follow these rules when renaming objects:

- You cannot rename a table to change its type. A Paradox table must be renamed as a Paradox table, and a dBASE table must be renamed as a dBASE table.

  You can copy a table to change its type. For more information, see Copying to a different table type.
- You cannot rename a table that is identified as the parent table in a referential integrity relationship. You must first either delete the referential integrity (by restructuring the child table) or delete the child table.
- When renaming an object, you can type a full path when you type the object's new name. This both renames the object and moves it to a new location.

Caution: Be careful when renaming tables. Once renamed, a table can't be found by associated documents. Queries that refer to a table under one name won't be bound to the table under its new name. The next time you open an unbound object, Database Desktop asks you to supply the name of the table to which you'd like it to be bound.

▪

Renaming tables on a network

See also

When you use Rename, Database Desktop must acquire an exclusive lock on the table. This means

- No user can access the table in any way.
- If there is a lock of any type open on the table, you must wait until it's released before you can use the Rename utility.
- If you rename an object with an existing object's name, Database Desktop deletes the existing object.

To rename objects

See also

To rename any type of object,

1. Choose Tools|Utilities|Rename, then choose the name of the file to rename in the Rename dialog box and choose OK.
2. Type the new name in the File Name text box, or choose an existing file name from the list to replace that file with the renamed one.
3. Choose Rename to rename the object.

About deleting objects

See also

You can delete tables, queries, and SQL files from within Database Desktop. For instructions see To delete an object.

Always use the Database Desktop Delete command to delete tables from within Database Desktop. Using the DOS DELETE command or the Windows Explorer may not delete all related files that make up a table (for example, the files containing a table's primary index, secondary indexes, validity checks, referential integrity, or BLOB data). The Database Desktop Delete command, however, deletes all files correctly.

You cannot delete a table that is identified as the parent in a referential integrity relationship. You must first either delete the referential integrity (from the child table), empty the child table, or delete the child table.

Caution: Be careful when deleting objects. You can't undo a deletion. Be sure that a table isn't used in any queries before you delete it. Queries that depend on the table are not deleted when the table is deleted.

▪

Deleting tables on a network

See also

When you use Delete to delete a table, Database Desktop must acquire an exclusive lock on the table. This means

- No user can access the table in any way.
- If there is a lock of any type open on the table, you must wait until it's released before you can use the Delete utility. This means you cannot delete a table that is open in the Database Desktop window.

Windows lets you open several instances of the same table at the same time, so you could be considered another user of the table, preventing the records from being deleted. Be sure to close the Table window and any of its associated-document windows before using Delete.

To delete an object

See also

To delete an object,

1. Choose Tools|Utilities|Delete, then choose the name of the file to delete in the Delete dialog box.
2. Paradox opens a dialog box that asks you to confirm the deletion. Choose Yes to delete the object or No to cancel the operation.

You cannot delete a table that is identified as the parent in a referential integrity relationship. You must first either delete the referential integrity (from the child table), empty the child table, or delete the child table.

Caution: Be careful when deleting objects. You can't undo a deletion. Be sure that a table isn't used in any queries before you delete it. Queries that depend on the table are not deleted when the table is deleted.

▪

About the Help system

See also

The Help menu is one way of using the Database Desktop Help system. You can also choose any Help button or press F1 at any time to open the Help system.

- When you use the Help menu, you use its menu commands to choose the subject you want help on.
- When you choose any Help button from a dialog box, you get help on using that dialog box.
- When you press F1 with a menu command highlighted, Database Desktop assumes you want help with that command and selects a Help topic accordingly. This type of help is called context sensitive; the context in which you ask for help determines the help provided. In a chain of menu commands, you must highlight the last one to get help.

As with other versions of Windows, any time the pointer changes to a hand, you can click the left mouse button to view more information:

- Click text with a dotted underline to view a related pop-up topic.
- Click text with a solid underline to jump to a related topic. Use the Back button or click an open Help window to return to a previous topic.
- Click See Also under the topic title to view a list of related topics.

In Windows 95, you can also

- Highlight a "book" in the Help Contents and choose the Print button to print all topics in that sequence.
- Choose Index to display the Help Index and search for a defined index entry.
- Choose Contents or Index, then choose Find to search for any text you enter.

To learn more about the features of Help in Windows 95 and Windows NT, see the following instructions.

For details about using Help in Windows 95,

1. Choose Start in the lower left corner of the Windows Desktop, then choose Help.

   Windows Help appears.

2. In the Help Contents, open the How To book, then open Use Help.

   To choose a book in the Help Contents, double-click it or move to it with the arrow keys and press Enter.

For details about using Help in Windows NT,

- In the Program Manager, choose Help|How To Use Help.

▪

International issues

See also

The following topics provide an overview of international features of Database Desktop and how use them to adhere to different conventions, such as

- Character set issues
- Sorting conventions
- Data formats

Note: These issues are of particular interest to those working in international environments, but they apply to all users.

▪

Preparation and assumptions

See also

Before installing Database Desktop in (or using tables from) an international setting, make sure of the following:

- The International settings of Windows Control Panel correspond to your needs.
- You understand the differences between (and implications of) the Windows (ANSI) character set and your DOS (OEM) code page.
- You know how to use Alt and the numeric keypad to enter extended characters in Windows applications and files. (Make sure NumLock is on before attempting this.)

For more information about these issues and concepts, consult your Windows and DOS documentation.

Sorting conventions

See also

Database Desktop uses language drivers to sort tables according to different conventions. If you are using a workstation with non-U.S. settings or are working with tables created on non-U.S. workstations, make sure Database Desktop is using the language driver(s) closest to the conventions you are used to.

In most cases, you should not have to worry about a table's language drivers after setting the default drivers for your workstation. When sharing tables between workstations, make sure the workstations are using the same default language drivers.

▪

Character set issues

See also

Because Database Desktop is a Windows application, it supports the ANSI character set for files that can be used only by other Windows applications. Database Desktop stores OEM characters in tables. This means Database Desktop translates ANSI characters to those in your OEM code page when saving table data.

For example, if you are using code page 437 (the default code page for U.S. workstations that support ASCII) and place an "Æ" (ANSI character 198) in a field, Database Desktop saves it as OEM character 146. You will see the same character when viewing the table, but it is not literally the same one you originally entered.

Most of the time, this is transparent; that is, there is no loss of data. However, if you enter a character that is not supported by your code page, Database Desktop converts it to one that is. For example, if you are using code page 437 and type an "Õ", Database Desktop converts it to an "O" because your code page does not support the original character. In this example, a mild form of data loss occurs; the tilde (~) is removed.

If you enter an ANSI character that cannot be converted to a similar character in your code page, Database Desktop replaces it with OEM character 254(□).

Character conversion occurs when you

- Enter data into a table
- Name a file
- Export data to OEM files or applications

In all other operations, Database Desktop uses and saves characters from the ANSI character set.

▪

Working with tables using different language drivers

See also

Although you can use different language drivers for different Database Desktop tables, we advise against linking tables using different language drivers. This includes (but is not limited to) operations like the following:

- Joining tables using different language drivers in queries
- Adding or subtracting two tables based on different language drivers
- Creating multi-table documents based on tables with different language drivers
- Defining referential integrity between tables based on different language drivers
- Defining a lookup table using a table based on a language driver different than the master table.

Database Desktop is designed to handle such operations; however some language driver combinations may yield unexpected results. For best results, choose one language driver, then restructure your tables so they use that driver.

Note: You can use Database Desktop language drivers only for Paradox tables and dBASE language drivers for dBASE tables.

Data formats

See also

In Database Desktop, the default symbols and formats used for your data are based on the International settings in the Windows Control Panel. For example, if you set your Windows Control Panel money symbol to "$", Database Desktop uses it when displaying money values. Similarly, Database Desktop formats date and time values according to the settings used by Windows.

To set Database Desktop's defaults permanently to different formats, begin by altering the settings in the Windows Control Panel. When you alter these settings it affects the default formats used in all Database Desktop operations, except some internal data conversion operations.

▪

Internal data conversion

See also

Many operations require Database Desktop to convert a string of characters to a number, date, or time value. For example, when you enter a date into a table, Database Desktop converts the characters you type to a value representing a date. This process is automatic and generally uses format settings in Windows Control Panel to control the conversion. However, a few operations use BDE to convert character strings to number, date, or time values and to convert these data formats to character strings. These operations include

- Queries that use selection criteria or perform pattern matching on number and date fields
- Table restructures that change alpha fields to number or date fields (and the reverse)
- Adding or subtracting records between tables that do not have the same structure

Because these operations use BDE for this internal data conversion, you must ensure the BDE configuration file uses the same data format conventions and settings as Windows Control Panel; otherwise, unexpected results might occur.

Note: This does not affect the way data is formatted when you display it; Database Desktop uses the settings in Windows Control Panel to display data.

We recommend using standard data formats when possible.

In most cases, you will not have to worry about these settings, because BDE (when installed) is configured to the data format conventions of the country defined in Windows Control Panel. However, if you customize Windows Control Panel so it uses settings that are different from your country's data format conventions, you should also configure BDE to the same settings; otherwise, queries that match date and numeric values may yield unexpected results.

For example, the "forward-slash" (/) is commonly used in U.S. workstations as a date separator. If you change this to an ampersand (&) in Windows Control Panel, you should also configure BDE so it uses an ampersand for a date separator.

Caution: Make a backup of the BDE configuration file before changing it with the BDE Configuration Utility.

To configure BDE to specialized date, time, or number formats,

1. Start the BDE Configuration Utility.
2. Choose either the Date, Time, or Number page, by clicking the respective tab. You will probably need to change the formats in all three sections.
3. Type in the parameters in the various format sections that match those of the new data format. For information on the specific sections of the format pages, see the BDE Configuration Utility's online Help.
4. Choose File|Save to save your changes.
5. If don't want to save your changes, select File|Exit and choose No when prompted to save your changes.

You can also update BDE to the current settings in Windows Control Panel by reinstalling BDE.

Note: If you customize BDE so it uses special number formats, you may not be able to share saved queries with other workstations unless those workstations have also been customized to the same settings.

To prevent character conversion

See also

If you want to prevent Database Desktop from converting characters, use the Strict Translation command from the Table menu while editing a table. If you turn Strict Translation on and try to save a record containing characters that are not supported by your code page, Database Desktop displays the message "Character(s) not supported by Table Language" and prevents you from saving the record until you

- Remove (or replace) the unsupported characters
- Turn Strict Translation off

To change the default language drivers

See also

To change the Database Desktop default language drivers, run the BDE Configuration Utility. On the Drivers page click the driver that you want to change in the Driver Name box. In the Parameters box there is a setting called LANGDRIVER: you can type in the new language driver that you want to use. Each language driver is appropriate only for a particular code page; for example, the Database Desktop International (Database Desktop 'intl') driver works with code page 437 only. Use language drivers appropriate for your code page. To see the language drivers appropriate for each code page, see the BDE Configuration Utility Online Help.

To change a table's language driver

See also

You can assign different language drivers to different tables. To change a table's language driver,

1. Restructure it (by choosing Tools|Utilities|Restructure or Table|Restructure).
2. Choose Table Language from the Table Properties panel.
3. Choose Modify, then choose the driver you want.
4. Choose OK to save your changes.

Use the Configuration Utility to change the default language driver for your tables.

▪

About creating tables

See also

Database Desktop supports several Paradox, dBASE, and SQL file formats. When you create a table, you can:

- Name the fields of the table (required)
- Specify field types (required) and sizes (required for some field types)
- Specify a table language to control sort order and available character set
- Assign indexes to the table
- Borrow the structure of an existing table

In addition, when you create a Paradox table you can:

- Assign a key, or primary index, to the table
- Assign secondary indexes to the table
- Define validity checks for individual fields
- Establish a table lookup to another table
- Establish referential integrity with another table
- Specify password security for the table or individual fields

Once you create a table, you can sort its records or search for data with commands and queries.

If you need to add or delete fields or make other structural changes to an existing table, you can restructure the table. For more information, see About restructuring tables.

▪

Guidelines for creating tables

See also

Planning is the first step in creating a table. You need to decide what you want the table to contain and how you want to lay it out. When you plan a table, keep these guidelines in mind:

- Put as little information as possible in each field. This allows for more flexible data maintenance and more straightforward querying. For example, if you break an address into separate fields for street, city, and state, you can easily query on these specific field values. This is where designing a database table differs from designing a spreadsheet.
- Be complete. Try to include fields for all the information you think you'll need, but don't clutter the table with information you don't need. If you discover later that you need another field, you can add it then.
- Use small tables. If you have a great deal of information to organize, it's generally better to put it in several small, related tables rather than in one all-encompassing table.
- Keep your tables familiar. It's often best to create tables that correspond to the kinds of documents you already use.
- Avoid redundancy. Beyond the common fields necessary for linking tables, don't duplicate information in different tables.
- Consider what kind of table you need. Because you can easily create either Paradox tables or dBASE tables, weigh the advantages of each. For example, Paradox tables support passwords, validity checks, referential integrity, and a greater variety of field types. dBASE tables support soft deletions, can have more than 255 fields, and are fully compatible with existing dBASE applications. Determine your needs before you choose a table type.

To create a simple table

See also

The following instructions describe how to create a Paradox table. To create another type of table, choose a different table type in step 2. Slightly different dialog boxes appear in the other steps. You can choose Help when viewing them for a description.

To create a new Paradox table from the Database Desktop window,

1. Choose File|New|Table. Or right-click the Open Table Toolbar button, and choose New.

   Database Desktop opens the Create Table dialog box.
2. If you want a table type other than Paradox 7.0, click the arrow next to the list box and select one from the drop-down list.
3. Choose OK.

   Database Desktop opens the Create Table dialog box, where you can specify the structure of the new table.

   You can borrow the structure of an existing table. For details, see About borrowing structures.
4. Type the name of the first field in the Field Name column of the Field Roster. See Rules for Paradox field names for more information.
5. Move to the Type column.

   You can move among the columns of the Field Roster by pressing Tab, Shift+Tab, or Enter, or by using the arrow keys or the mouse. Database Desktop automatically skips over any columns that are not required.
6. Press Spacebar or right-click the Type column to display a list of field types. Type the symbol for the field type you want. See Paradox field types and sizes for more information.
7. Move to the Size column and type an appropriate field size (if a size is required). See Paradox field types and sizes for more information.
8. Press the down arrow key. Repeat the above steps until you've specified as many fields as you want.
9. If you want, define a key (Paradox tables only) and set table properties. When creating any table, you can:

- Specify a table language
- Assign indexes to the table

Furthermore, when creating a Paradox table you can

- Define validity checks for individual fields
- Assign secondary indexes to the table
- Establish a table lookup to another table
- Establish referential integrity with another table
- Specify password security for the table or individual fields

When the table structure is defined, choose Save As to name and save the table. After you save the structure, you can enter data in your new table.

Once you save a new table, you must restructure it to add and delete fields or change any other part of its structure.

▪

Rules for Paradox field names

See also

Follow these rules when specifying field names for Paradox tables:

- The maximum length of a field name is 25 characters.
- A field name cannot start with a blank space (unless it is enclosed in quotation marks), but it can contain blank spaces.
- Each field name in a table must be unique. (You can not have two identical field names.) You cannot make a name unique by doing one of the following
- Adding a blank space at the end of the name
- Changing the capitalization of the name
- A field name should not contain certain characters if you plan to use the table in a query, because these characters have special significance. These characters are:
- The comma (,), the pipes (|), and the exclamation point (!)
- Avoid using SQL keywords, such as SELECT and COUNT.

Rules for dBASE field names

See also

Follow these rules when specifying field names for dBASE tables:

- A field name cannot exceed 10 characters.
- A field name cannot contain blank spaces.
- Each field name in a table must be unique. You cannot have two identical field names. You cannot make a name unique by
- Adding a blank space at the end of the name
- Changing the case of the name

To create a field

See also

To create a field from the Desktop,

1. Open the Create Table or the Restructure Table dialog box.

   Choose File|New|Table and specify the table type to open the Create Table dialog box, or right-click a table in the Project Viewer and choose Restructure to open the Restructure Table dialog box.
2. Use the arrow keys to move to a new row in the Field Roster, if necessary.
3. Enter the name of the field in Field Name. See Rules for Paradox field names and Rules for dBASE field names for more information.
4. Enter the type of field in Type. Right-click or press the Spacebar to choose from a list of field types. For more information on field types, see Paradox field types and sizes and dBASE field types and sizes.
5. Enter the size of the field in Size, if it is necessary for your field type.
6. If you are creating a Paradox table, specify whether the field is a key. Move to the Key column and follow the instructions on the screen. For more information on keys, see About primary indexes (key fields).
7. If you are creating a dBASE table, specify the number of decimal places in Dec, if it is necessary for your field type.

Rules for Informix field names

See also

- The maximum length of a field name is 18 characters.
- A field name must begin with a letter (A-Z, a-z).
- A field name can contain digits from 0 to 9, uppercase or lowercase letters, and underscore (_) characters.
- Each field name in a table must be unique. (You cannot have two identical field names.)

Rules for InterBase field names

See also

- The maximum length of a field name is 31 characters.
- A field name must begin with a letter (A-Z, a-z).
- A field name can contain letters (A-Z, a-z), digits, $, or underscore (_) characters.
- You cannot use InterBase reserved words for table names. See the InterBase *Programmer's Reference* for a list of reserved words.
- Each field name in a table must be unique. (You cannot have two identical field names.)

▪

Rules for Oracle field names

See also

- The maximum length of a field name is 30 characters.
- A field name must begin with a letter (A-Z, a-z).
- A field name can contain letters (A-Z, a-z), digits (0-9), or the _, $, or # characters.
- You cannot use ORACLE reserved words for remote table names, quoted table names, or quoted index names. For a list of reserved words and other naming restrictions, see the ORACLE *Programmer's Reference.*
- Each field name in a table must be unique. (You cannot have two identical field names.)

Rules for Sybase field names

See also

- The maximum length of a field name is 30 characters.
- A field name must begin with a letter (A-Z, a-z).
- A field name can contain letters (A-Z, a-z), digits (0-9), or the _, $, or # characters.
- You cannot use SQL Server reserved words for remote table and column names. See the SQL Server *Programmer's Reference* for a list of reserved words.
- Field names may be case sensitive, depending on how SQL Server is installed.
- Each field name in a table must be unique. (You cannot have two identical field names.)

Paradox field types and sizes

See also

The valid Paradox field types and sizes are

Symbol	Size	Type
A	1 - 255	Alpha
N		Number
$		Money
S		Short
I		Long Integer
#	0 - 32*	BCD
D		Date
T		Time
@		Timestamp
M	1 - 240**	Memo
F	0 - 240**	Formatted Memo
G	0 - 240***	Graphic
O	0 - 240***	OLE
L		Logical
+		Autoincrement
B	0 - 240***	Binary
Y	1 - 255	Bytes

* Number of digits after the decimal point

** Memo and formatted memo fields can be virtually any length. The value you specify in the Create Table dialog box refers to the amount of the memo Database Desktop stores in the table (1 to 240 characters for memos and 0 to 240 characters for formatted memos). The entire memo is stored outside the table. For example, if you assign a size value of 45 to the field, Database Desktop stores the first 45 characters in the table. It stores the whole memo field in another file (with the extension .MB) and retrieves it as you scroll through the records of the table.

*** Optional

Tip: If all your memos are smaller than a given size (for example, 200 characters), you can save space and time by setting the memo field size equal to or larger than this size. Database Desktop stores the entire memo in the table if it is less than the given size.

▪

Paradox alpha fields

See also

Paradox alpha fields contain strings consisting of

- Letters
- Numbers
- Special symbols like %, &, #, or =
- Other printable ASCII characters

▪

Paradox number fields

See also

Paradox number fields must contain only numbers. Number fields can hold positive or negative values. The range of values possible for a number field is from -10^{307} to 10^{308} with 15 significant digits. Use number fields when you plan to perform calculations on the values in the fields.

Number fields are best used when you want to perform calculations on the values in the field.

Tip: It is a good idea to use an alpha field rather than a number field for phone numbers or zip codes. In an alpha field, you can include parentheses and hyphens.

To change the default display of a number field, use Paradox.

Paradox money fields

See also

Paradox money fields, like number fields, can contain only numbers. They can hold positive or negative values. But by default, money fields are formatted to display decimal places and a money symbol. Regardless of the number of decimal places displayed, Database Desktop recognizes up to six decimal places when performing internal calculations on money fields.

To change the default display of a money field, use Paradox.

Paradox short fields

See also

Paradox short fields are special number fields that can contain only whole numbers in the range -32,767 to 32,767. Short fields require less disk storage than ordinary number fields. They are available only in Paradox-type tables.

Paradox long integer fields

See also

Paradox long integer fields are 32-bit signed integers that contain whole numbers (nonfractional) with complete accuracy in the range -2147483648 to 2147483647 (plus or minus 2 to the 31st). Long integer fields require more space to store than short fields.

Paradox BCD fields

See also

Paradox BCD fields contain numeric data in a BCD (Binary Coded Decimal) format. Use BCD fields when you want to perform calculations with a higher level of precision than that available with the use of other numeric fields. Calculations on BCD fields are not performed as quickly as those on other numeric fields.

The BCD field type is provided primarily for compatibility with other applications that use BCD data. Database Desktop correctly interprets BCD data from other applications that use the BCD type. However, when Database Desktop performs calculations on BCD data, it converts the data to the numeric float type, then converts the result back to BCD.

Note: Although BCD fields can handle larger numbers, you can only enter a number with 15 significant digits or less into a BCD field.

▪
Paradox date fields

See also

Paradox date fields can contain any valid date from January 1, 9999 BC to December 31, 9999 AD. Database Desktop correctly handles leap years and leap centuries and checks all dates for validity. Database Desktop treats all BC years as leap years.

Database Desktop correctly handles leap years and leap centuries and checks all dates for validity.

To change the default display of a date field, use Paradox.

▪

Paradox time fields

See also

Paradox time fields contain times of day, stored in milliseconds since midnight, and are limited to 24 hours.

To change the default display of a time field, use Paradox.

Paradox timestamp fields

See also

Paradox timestamp fields contain both time and date values. To enter today's date and the current time, press Spacebar repeatedly until Database Desktop enters the data. Rules for this field type are the same as those for date fields and time fields.

To change the default display of a timestamp field, use Paradox.

Paradox memo fields

See also

Use memo fields for text strings that are too long to store in an alpha field.

Memo fields can be virtually any length. The size value you assign refers to the amount of the memo Database Desktop stores in the table. This can be from 1 to 240 characters. Database Desktop stores the whole memo outside the table (in the .MB file). Database Desktop retrieves the data from the .MB file as you scroll through the records of the table. The amount of data a memo field contains is limited only by the disk space available on your system.

Tip: If all your memos are smaller than a given size (for example, 200 characters), you can save space and time by setting the memo field size to be equal to or larger than this given size. You will still have an .MB file, but Database Desktop will not have to access it to display the field's data.

Memo fields can contain letters, numbers, special symbols (such as %, &, #, and =), or any other printable ASCII character (except null). You can enter line breaks, tabs and other print control characters in memo fields.

To view memo fields, use Paradox.

▪

Paradox formatted memo fields

See also

Paradox formatted memo fields are like memo fields except you can format the text (in Paradox). Paradox recognizes text attributes (typeface, style, color and size) and stores them with the text.

To define a field as a formatted memo field, use the Create Paradox Table dialog box or the Restructure Paradox Table dialog box

To view and format this type of field, use Paradox.

▪

Paradox graphic fields

See also

Paradox graphic fields contain pictures. You can create graphics in a painting or drawing application, or scan in images.

You can select .BMP, .PCX, .TIF, .GIF, and .EPS file formats. When you paste a graphic into a graphic field, Database Desktop converts the graphic into the .BMP format.

Graphic fields do not require a size because they are not stored in the table, but in separate files.

To view a graphic field, use Paradox.

▪

Paradox OLE fields

See also

Use the OLE field to store different kinds of data, such as images, sound, documents, and so on. The OLE field provides you with a way to view and manipulate this data within Paradox. Database Desktop does not support OLE data.

You do not need to specify a size for OLE fields because they are not stored in the table, but in separate files.

To view and add data to OLE fields, use Paradox.

Paradox logical fields

See also

Paradox logical fields contain values representing true or false (yes or no). By default, valid entries include "True" and "False" (case is not important).

To change the default display of a logical field, use Paradox.

Paradox autoincrement fields

See also

Paradox autoincrement fields contain long integer, read-only (non-editable) values. Database Desktop begins with the number 1 and adds one number for each record in the table.

Deleting a record does not change the field values of other records.

When creating a Paradox table, you can specify the starting number of an autoincrement field by specifying a minimum value for it. See About minimum and maximum values for more information.

Paradox binary fields

See also

Binary fields should be used only by advanced users who need to work with data that Database Desktop cannot interpret. Database Desktop cannot display or interpret binary fields. A common use of a binary field is to store sound.

Unlike bytes fields, binary fields do not require a size because they are stored in a separate file (the .MB file), not in the table.

Paradox bytes fields

See also

Bytes fields should be used only by advanced users who need to work with data that Database Desktop cannot interpret. A common use of a bytes field is to store bar codes or magnetic strips.

Unlike binary fields, bytes fields are stored in the Paradox table (rather than in the .MB file), allowing for faster access.

Paradox 4 field types

See also

The valid Paradox 4 field types and sizes are

Symbol	Size	Type
A	1 - 255	Alpha
N		Number
$		Money
D		Date
S		Short
M	1 - 240*	Memo
F	0 - 240*	Formatted Memo
B	0 - 240**	Binary
G	0 - 240**	Graphic
O	0 - 240**	OLE

* Memo and formatted memo fields can be virtually any length. The size value you specify in the Create Table dialog box refers to the amount of the memo Database Desktop stores in the table (1 to 240 characters for memos and 0 to 240 characters for formatted memos). The whole memo is stored outside the table. For example, if you assign a size value of 45 to the field, Database Desktop stores the first 45 characters in the table. It stores the whole memo field in another file (with the extension .MB) and retrieves it as you scroll through the records of the table.

** Optional

Tip: If all your memos are smaller than a given size (for example, 200 characters), you can save space and time by setting the memo field size equal to or larger than this size. Database Desktop stores the entire memo in the table if it is less than the given size.

Paradox 3.5 field types

See also

The valid Paradox 3.5 field types and sizes are

Symbol	Size	Type
A	1 - 255	Alpha
N		Number
$		Money
D		Date
S		Short

▪

dBASE field types and sizes

See also

The valid dBASE field types and sizes are

Symbol	Size	Decimal Point	Type
C	1 - 254		Character (alpha)
F*	1 - 20	0 - 18, and <=Size - 2	Float (numeric)
N	1 - 20**	0 - 18, and <= Size - 2	Number (BCD)
D			Date
L			Logical
M***			Memo
O****			OLE
B****			Binary

*Available only in dBASE IV and later versions.

** For dBASE III+ tables, the size can be from 1 - 19

***Memo field formats differ between dBASE III+ and later versions of dBASE.

****Available only in dBASE for Windows and later versions

dBASE character fields

See also

dBASE character fields can contain any printable character (including blank spaces). The maximum size of a dBASE character field is 254.

dBASE float fields

See also

dBASE provides two ways to store numeric data. The float number type contains numeric data in a binary floating-point format. Use the float number type on fields that will not require precise calculations to be performed on them; some degree of precision is rounded or truncated during calculation. Float number fields are best used to contain whole numbers, or numbers of up to two decimal places.

The size of a dBASE float number field can be from 1 to 20.

Setting decimal places

You set the number of decimal places in the Dec column of the Field Roster in the Create dBASE Table or Restructure dBASE Table dialog box.

In the Dec column, you specify how many decimal places to store. Enter a number at least 2 less than the field size. This is because Database Desktop counts the decimal point and sign (if any) as part of the field size.

dBASE number fields

See also

dBASE number fields contain numeric data in a Binary Coded Decimal (BCD) format. Use number fields when you will need to perform precise calculations on the field data. Calculations on number fields are performed more slowly, but with greater precision than on float number fields.

The size of a dBASE number field can be from 1 to 20 (except for dBASE III+ tables, which can be from 1 to 19).

Setting decimal places

Set the number of decimal places in the Dec column of the Field Roster in the Create/Restructure dialog box.

In the Dec column, you can specify how many decimal places to store. Enter a number at least 2 less than the field size. This is because Database Desktop counts the decimal point and sign (if any) as part of the field size.

dBASE date fields

See also

Date fields contain dates. The default date entry and display format is Windows Short (which uses the short date format you defined from the Windows Control Panel Date/Time Properties dialog box), but you can format dBASE date fields in other ways using Paradox or dBASE. The size for a date field is always 8.

dBASE logical fields

See also

Logical fields contain a single character representing True or False (Yes or No) values. In dBASE logical fields, logical true can be entered as T, t, Y, or y. Logical false can be entered as F, f, N, or n. The size for a dBASE logical field is always 1.

To change the default format of dBASE logical fields, use Paradox or dBASE.

dBASE memo fields

See also

dBASE memo fields contain blocks of text that are too large to be stored in a character field. The contents of memo fields are stored externally to the table. You do not specify a field size for dBASE memo fields.

▪

dBASE OLE fields

See also

Use the OLE field to store different kinds of data, such as images, sound, documents, and so on. The OLE field provides you with a way to view and manipulate this data within Paradox or dBASE. Database Desktop does not support OLE data.

You do not need to specify a size for OLE fields because they are not stored in the table, but in separate files.

dBASE binary fields

See also

Binary fields should be used only by advanced users who need to work with data that Database Desktop cannot interpret. Database Desktop cannot display or interpret binary fields. A common use of a binary field is to store sound. Binary fields do not require a size because they are stored in a separate file (the .DBT file), not in the table.

dBASE record lock fields

See also

In a multiuser environment, each user can place record locks on a shared table. For example, if user JSMITH is editing record number 12 of Stock, user MBROWN cannot access that record until it is unlocked. This prohibits one user from unintentionally overwriting another user's work.

The dBASE table type gives you the Record Lock option to show you information about a locked record. If you check Record Lock, Database Desktop adds a hidden field to the table. This field shows you when a record was locked and by whom.

Note: Although Database Desktop adds the Record Lock field to the table, you will not see it when you view the table. You see a record's Record Lock field only if you are locked out of that record.

Use the Create dBASE Table dialog box or Restructure dBASE Table dialog box to create the Record Lock field for a dBASE table. Record Lock is not available for dBASE III+ tables.

The information you see when you find a locked field depends on the Info Size you specify. The Record Lock field can be from 8 to 24 characters. The default is 16.

- The first two characters tell whether a user has changed the record.
- The next three characters tell the time a user placed the lock.
- The next three characters tell the date a user placed the lock.
- The remaining 16 characters are optional. They tell the name of the user that placed the lock.

The default size of 16 displays the changed status of the record, the time and date of the lock, and the first 8 characters of the user who placed the lock.

Informix field types and sizes

See also

The following table list valid Informix field types and sizes. For detailed information on field types and sizes, see your Informix documentation.

Name	Size	Dec	Description
CHAR	1-32,769		Fixed-length character data
SMALLINT			Whole number -32,767 to +32,767
INTEGER			Integer -2,147,483,647 to +2,147,483,647
SMALLFLOAT			Single-precision floating-point number with approximately 8 significant digits
FLOAT			Single-precision floating-point number with up to 16 significant digits
MONEY	0-32	0-32	Fixed-point number with up to 32 significant digits
DECIMAL	0-32	0-32	Decimal floating-point number with up to 32 significant digits
DATE			Calendar date Jan 1, 1900 to Dec 31, 9999
DATETIME			Calendar date Jan 1, 0001 to Dec 31, 9999 and 24-hour time of day
INTERVAL			Span of time (year-month or day-time)
SERIAL			Sequential number up to 2,147,483,647 assigned automatically by the database server when a row is inserted
BYTE			Any type of binary data
TEXT			Variable-length character data to 2,147,483,647 bytes
VARCHAR	1-255		Variable-length character data

Note: In Database Desktop you can create all Informix field types, and you can view and edit data in all fields except BYTE, CHAR > 255, and TEXT.

InterBase field types and sizes

See also

The following table list valid InterBase field types and sizes. For detailed information on field types and sizes, see your InterBase documentation.

Name	Size	Dec	Description
SHORT			Integer -32,768 to +32,767
LONG			Integer -2,147,483,647 to +2,147,483,647
FLOAT			Floating-point number with up to 7 digits of precision
DOUBLE			Floating-point number with up to15 digits of precision
CHAR	0-32,767		Fixed-length character data
VARCHAR	0-32,767		Variable-length character data
DATE			Calendar date Jan 1, 0100 to Dec 11, 5941
BLOB			Any type of binary data
ARRAY			You cannot create an ARRAY field

Note: In Database Desktop you can create all InterBase field types except ARRAY, and you can view and edit data in all fields except BLOB and ARRAY.

Oracle field types and sizes

See also

The following table list valid Oracle field types and sizes. For detailed information on field types and sizes, see your Oracle documentation.

Name	Size	Dec	Description
CHAR	1-255		Fixed-length character data
RAW	1-255		Binary data to 255 bytes
DATE			Calendar date Jan 1, 4712 BC to Dec 31, 4712 AD and 24-hour time of day
NUMBER	0-38		Floating-point number with up to 38 digits of precision
LONG			Variable-length character strings up to 2 gigabytes ((2**32)-1 bytes)
LONG RAW			Binary data up to 2 gigabytes
FLOAT			Floating-point number with up to 38 digits of precision
VARCHAR2	1-2000		Variable-length character data
VARCHAR	1-255		Variable-length character data

Note: In Database Desktop you can create all Oracle field types, and you can view and edit data in all fields except LONG, LONG RAW, and RAW.

Sybase field types and sizes

See also

The following table list valid Sybase field types and sizes. For detailed information on field types and sizes, see your Sybase documentation.

Name	Size	Dec	Description
CHAR	1-255		Fixed-length character data
VARCHAR	1-255		Variable-length character data
INT			Integer -2,147,483,647 to +2,147,483,647
SMALLINT			Integer -32,768 to +32,767
TINYINT			Integer 0 to 255
FLOAT			8-byte floating-point number
MONEY			-922,337,203,685,477.5808 to +922,337,203,685,477.5808
TEXT			Variable-length character data up to 2,147,483,647 bytes
BINARY	1-255		Fixed-length binary data up to 255 bytes
VARBINARY	1-255		Variable-length binary data up to 255 bytes
IMAGE			Variable-length binary data 0 to 2,147,483,647 bytes
BIT			Either 0 or 1. Cannot be NULL. Integer values other than 0 or 1 are interpreted as 1
DATETIME			Calendar date Jan 1, 1753 to Dec 31, 9999 and 24-hour time of day
TIMESTAMP			Binary timestamp
REAL			4-byte floating-point number
SMALLMONEY			-214,748.3648 to +214,748.3647
SMALLDATETIME			Calendar date Jan 1, 1900 to Jun 6, 2079 and 24-hour time of day

Note: In Database Desktop you can create all Sybase field types, and you can view and edit data in all fields except BINARY, IMAGE, TEXT, TIMESTAMP, and VARBINARY.

▪

About keys and indexes in tables

See also

An index is a file that determines the order in which Database Desktop accesses the records in a table. Paradox, dBASE, and SQL tables use indexes to organize the records in a table, but their indexes work differently.

Indexes can be primary or secondary. In Paradox tables, the primary index is also called the key.

Paradox tables

Database Desktop organizes the records of a keyed Paradox table according to the values in the key field(s). This is its primary index. By default, all indexes organize and access data in ascending order (A to Z, or 0 to 9).

When a composite key is defined for a table, Database Desktop creates a primary composite index, which organizes the records by the first field of the key (according to the table's structure), then the next, and so on.

In Paradox tables, a secondary index defines an alternate view order to temporarily change the display order of the records. The physical location of the records in the table does not change.

Secondary indexes are also used for queries, to speed performance, and to establish links between tables. For more information on secondary indexes, see About secondary indexes.

dBASE tables

In dBASE tables, an index is used to organize the records in a table according to the values in one or more fields.

SQL tables

SQL tables use unique and non-unique indexes, but they do not use the primary keys that Paradox tables use. You can create multiple indexes for an SQL table; for each index, you specify whether it is unique or non-unique. SQL indexes, unlike Paradox and dBASE indexes, are always maintained.

You can use Database Desktop to create and modify indexes on SQL tables, but you cannot specify which index to use in Database Desktop.

When you use an SQL table in Database Desktop, the table should have a unique index. If it does not have a unique index and you edit the table's data, you may not be able to view the edits as you are making them.

About primary indexes (key fields)

See also

A Paradox table's key establishes the primary index and sort order for the table.

Database Desktop organizes the records of a keyed table according to the values in the key of the table. These fields, which make up the table's key, are its primary index.

By default, all indexes organize and access data in ascending order (A to Z or 0 to 9). By creating a key field, you tell Database Desktop to organize the table by the values in that field. Changing the key changes where Database Desktop physically stores each record in the table.

A key requires each value in the field(s) that defines the key to be unique. For example, if the Customer No field is identified as the key of the Customer table, each value in the Customer No field must be unique. Likewise, if the Order No and Stock No fields are identified as the key of the Lineitem table, the field values (taken as an ordered group) must be unique. This guards against duplication of data within the table.

The key for a table must be the first field or group of fields in the Field Roster.

When you use an autoincrement field type as the table's key, Database Desktop automatically creates a unique value for each record in the table.

Keys are required for most types of table links and for using Paradox table data integrity features.

Keys are also used to speed up queries, searches, and locates for Paradox tables. If a key is a composite key, Database Desktop only uses the first field of the key to speed up such operations.

A primary index from a composite key

If you identify more than one field as keyed, it is known as a composite key. These fields, taken as a group, must be unique for each record of the table. When you define a composite key, Database Desktop creates a primary composite index, which organizes the records by the first field of the key (according to the table structure), then the next, and so on. For more information about composite keys, see Composite key fields.

▪

Rules for defining key fields

See also

Follow these rules when you define a key:

- A table can have only one key. This key can be made up of one or more fields.
- Keys cannot contain memo, formatted memo, graphic, OLE, binary, logical, or bytes fields.
- If a key is defined as a single field, that field must be the first field in the Field Roster.
- If you identify more than one field as keyed, you create a composite key. These fields, taken as a group, must be unique for each record of the table. The composite key must begin on the first field in the Field Roster.

The effect of restructuring tables on key fields

See also

You might rearrange fields so that the key fields are no longer the first consecutive fields. When you click OK, Database Desktop alerts you to correct any violation of key field rules in the Restructure Table dialog box.

If you add keys to a table that was previously unkeyed or had different keys, you can cause a key violation: Data already entered into the table violates a rule established by the new key. Database Desktop writes the key-violating records to a special temporary table called Keyviol.

Records that are key violations are deleted from your table. You can change the records in the Keyviol table so they comply with the key requirements, then add them back to your original table using Tools|Utilities|Add.

Composite key fields

See also

A Paradox type table can have more than one field defined as a key field. The fields are treated as a group or composite. Composite key fields must be the first fields of the table.

Use composite key fields when there is no single field in a table where every value is unique.

When a table has a composite key field, duplicate values are allowed in an individual key field, as long as values are not duplicated across all key fields. In other words, the key fields, taken as a group, must uniquely identify a record.

Database Desktop sorts tables that have composite key fields by starting with the first field, then sorting on following fields.

To create a key

See also

To define a Paradox field as a key field,

1. Display the structure of the table in the Create Paradox Table dialog box or the Restructure Paradox Table dialog box.
2. Select the Key column.
3. Press the Spacebar or double-click to toggle on the key field marker.

   Database Desktop displays an asterisk (*) in the Key column for that field.

To remove a key

See also

To remove a key from a field or group of fields,

1. Display the structure of the table in the Create Paradox Table dialog box or the Restructure Paradox Table dialog box.
2. Select the Key column.
3. Press the Spacebar or double-click to toggle on the key field marker.

   Database Desktop displays an asterisk (*) in the Key column for that field.

Removing one or more fields from a composite key might cause duplicate values in the remaining field(s) of the key. Database Desktop places duplicate records in the temporary Keyviol table as discussed in The effect of restructuring tables on key fields.

If you remove a key located above other keys, an error message appears when you try to save the table structure. Make sure all key fields are the first fields in your table structure.

▪

About secondary indexes

See also

A secondary index is a field or group of fields that you define as

- An alternate sort order for the table
- A field you can link the table on
- A way to speed up certain search and locate operations

A table can have more than one secondary index. In fact, you can identify each field of the table as a secondary index, so you can sort the table on any of its fields. You can also create composite secondary indexes by combining two or more fields, up to a total of 1 fields.

You cannot create a secondary index on a memo, formatted memo, binary, OLE, graphic, logical or bytes field.

When you use a secondary index, you change only the view order of the records. The physical location of the records in the table does not change.

Paradox tables have these options for secondary indexes: Composite, Unique, Case-sensitive, Maintained, and Ascending/Descending. For details, see Types of secondary indexes.

Alternate sort orders

When sorting a keyed table, you must use a secondary index. Only an explicitly defined secondary index can override the primary sort order established by a table's key definition.

Linking tables

Secondary indexes are also used in linking Paradox tables.

For example, you may want to link the sample Customer and Orders tables so you can see the orders that each customer has placed. The Orders table has a secondary index identified on its Customer No field. This means Database Desktop can quickly find all the records with a given Customer No value. When you link the tables, Database Desktop identifies each Customer No value in Customer, then finds and displays all matching Customer No values in Orders.

Search and locate operations

Database Desktop uses a secondary index to speed up some search and locate operations on Paradox tables if the index is:

Single-field

Case-sensitive

Maintained

Types of secondary indexes

See also

Paradox tables provide many different options for secondary indexes.

Composite

Composite secondary indexes use more than one field in a table. You can define a secondary indexes on a group of up to 16 fields. Composite indexes organize the data by the first field of the index first, then by the next field, and so on.

Unique

Unique secondary indexes determine whether records can have duplicate values in the secondary index field or fields. If Unique is checked and two or more records have the same value in the secondary index field, the attempt to define the secondary index fails. You can eliminate duplicate values and define the secondary index again.

Case-sensitive

Case-sensitive indexes use capitalization, or case, as a criterion for sorting. In a case-sensitive index, uppercase letters sort before lowercase letters.

Maintained

When a secondary index is maintained, Database Desktop automatically updates the index whenever you update the table. A table must have a key before you can create a maintained secondary index.

A non-maintained index is not automatically updated when you update the table.

When you choose to view a table with a non-maintained index, it is temporarily locked and cannot be updated.

Ascending/Descending

You can specify either ascending or descending sort orders for secondary indexes on Paradox tables. Ascending indexes sort in the following ways:

- Alpha fields sort characters in the order a, b, c, and so on.
- Number, money, short, and long integer fields sort in the order 1, 2, 3, and so on.
- Date, time, and timestamp fields sort in the order 12/1/95, 12/2/95, 12/3/95, and so on.

Note: Unique and Descending indexes cannot be applied to tables saved in Paradox file types earlier than version 7.

To create secondary indexes

See also

To define a field or group of fields as a secondary index,

1. Display the structure of the table in the Create Paradox Table dialog box or the Restructure Paradox Table dialog box.
2. Select Secondary Indexes from the Table Properties drop-down list. The Define button becomes available and any existing secondary indexes appear.
3. Choose Define to open the Define Secondary Index dialog box. The Fields list displays the fields you can use as a secondary index. BLOB fields are dimmed.
4. Select the field you want to create the secondary index on, then choose the Add arrow or press Alt+A to move it to the Indexed Fields list.
5. Check the Index Options you want: Unique, Descending, Case Sensitive, and Maintained. For a description of each option, see Define Secondary Index dialog box. (Note: You can only check Maintained if the table has a key.)
6. Choose OK. Database Desktop automatically names case-sensitive indexes you create on a single field with the field's name. If you created a composite secondary index (using more than one field), you have to give the index a name in the Save Index As dialog box.

To create another secondary index, choose Define again. As you create secondary indexes, they are listed in the box below the Define button in the Create Paradox Table (or Restructure Paradox Table) dialog box.

Composite secondary indexes

To create a composite secondary index, follow the same steps, except add more than one field to the Indexed Fields list in step 4. For details, see About composite secondary indexes and To create a composite secondary index.

To modify a secondary index

See also

To modify a secondary index,

1. Display the structure of the table in the Create Paradox Table dialog box or the Restructure Paradox Table dialog box.
2. Select Secondary Indexes from the Table Properties drop-down list.
3. Select the index you want to modify in the list below the Define button.
4. Choose Modify.

   The Define Secondary Index dialog box opens with the selected index specification filled in. Change the specifications to what you want, then choose OK.

To delete a secondary index

See also

To delete a secondary index,

1. Display the structure of the table in the Create Paradox Table dialog box or the Restructure Paradox Table dialog box.
2. Select Secondary Indexes from the Table Properties drop-down list.
3. Select the index you want to delete in the list below the Define button.
4. Choose Erase.

   The index is deleted.

About composite secondary indexes

See also

You can create a composite secondary index by adding more than one field to the Indexed Fields list in the Define Secondary Index dialog box.

Database Desktop creates the composite index in the order that the fields appear in the Indexed Fields list. When you use this index, Database Desktop sorts the table by the top field first, then by the next, and so on.

To create a composite secondary index

See also

To define a composite secondary index,

1. Display the structure of the table in the Create Paradox Table dialog box or the Restructure Paradox Table dialog box.
2. Select Secondary Indexes from the Table Properties drop-down list. The Define button becomes available and any existing secondary indexes appear.
3. Choose Define to open the Define Secondary Index dialog box. The Fields list displays the fields you can use as a secondary index. BLOB fields are dimmed.
4. Select each field you want to include in the index, then choose the Add arrow or press Alt+A to move it to the Indexed Fields list.

   Database Desktop creates the composite index in order of the Indexed Fields list. When you use the index, Database Desktop sorts the table by the top field first, then by the next, and so on.
5. Check the Index Options you want: Unique, Descending, Case Sensitive, and Maintained. For a description of each option, see Define Secondary Index dialog box. (Note: You can only check Maintained if the table has a key.)
6. Choose OK.

   Database Desktop displays the Save Index As dialog box.
7. Enter a name for the index and choose OK.

To add a field to a composite secondary index

See also

A composite secondary index can have up to 16 fields.

To add a field to a composite secondary index,

1. Display the structure of the table in the Create Paradox Table dialog box or the Restructure Paradox Table dialog box.
2. Select Secondary Indexes from the Table Properties drop-down list.
3. Select the index that you want to change, then click Modify to open the Define Secondary Index dialog box.
4. Select the field you want to include in the Fields list, then choose the Add arrow or press Alt+A to move it to the Indexed Fields list.

   Database Desktop adds the field below the selected field in the Indexed Fields list.

To remove a field from a composite secondary index

See also

To remove a field from a composite secondary index,

1. Display the structure of the table in the Create Paradox Table dialog box or the Restructure Paradox Table dialog box.
2. Select Secondary Indexes from the Table Properties drop-down list.
3. Select the index that you want to change, then click Modify to open the Define Secondary Index dialog box.
4. Select the field you want to remove in the Indexed Fields list, then choose the Remove arrow or press Alt+R to move it to the Fields list.

   To remove all fields from Indexed Fields, choose Clear All.

To rearrange fields in a composite secondary index

See also

To rearrange fields in a composite secondary index,

1. Display the structure of the table in the Create Paradox Table dialog box or the Restructure Paradox Table dialog box.
2. Select Secondary Indexes from the Table Properties drop-down list.
3. Select the index that you want to change, then click Modify to open the Define Secondary Index dialog box.
4. Select the field you want to move in the Indexed Fields list, then use the Change Order arrows to move it up or down.

About dBASE indexes

See also

When working with dBASE tables, Database Desktop uses an index to organize the records in a table according to the values in one or more fields.

When you create an index on a dBASE table, Database Desktop creates a file that contains the indexed field's values and their corresponding record numbers. Database Desktop refers to the index file when locating and displaying the records in a table.

When you use an index on a dBASE table, the records appear in a different order. However, the records remain stored in the same physical location in which you originally entered them.

Although Database Desktop supports both .MDX files and .NDX files, it is recommended that you use a dBASE production index (the .MDX file which uses the table name as its file name) whenever possible. Although you can create non-production .MDX files as well as .NDX files, Database Desktop automatically maintains only the production index.

▪

Maintained dBASE indexes

See also

You tell Database Desktop to automatically maintain a dBASE index in the Define Index dialog box.

When you check the Maintained option, Database Desktop updates the index every time the table changes. This speeds up certain operations like queries.

- Database Desktop saves a maintained index as part of an .MDX file and gives the .MDX file the same name as the table. This is your production index. It is recommended that you use production indexes when working in Database Desktop.
- When you save a maintained index, Database Desktop asks you for a tag name. The .MDX file can contain several maintained index specifications.
- Maintained is unavailable for dBASE III+ tables.
- Non-maintained indexes are assigned the .NDX file extension. You cannot have a production .NDX file.
- You cannot restructure a non-maintained index.

▪

Creating a dBASE expression index

See also

Expression indexes are useful for creating a multi-field (composite) index on a dBASE table.

You create an expression index on a value that you express using any formula that results in a value, using dBASE expression syntax. For example, you could create an expression index such as FIRST_NAME + LAST_NAME, where both FIRST_NAME and LAST_NAME are field names and of the same data type.

Some elements of dBASE expressions are not allowed; for example, memory variables, user-defined functions, macro substitution, and references to fields in other tables.

You create an expression index on a dBASE table from the Create dBASE Table dialog box or the Restructure dBASE Table dialog box.

Follow the instructions in To create an index on a dBASE table.

To use field names in an expression index, position the insertion point in the appropriate text box and click the field you want in the Field list.

For example, to create the expression index FIRST_NAME + LAST_NAME, click the Expression Index button to position the insertion point in the Expression Index text box, then click FIRST_NAME in the Field list. FIRST_NAME appears in the text box. Enter + and click LAST_NAME in the Field list.

Creating a subset condition expression

See also

A subset condition expression (also called a filter) is an expression that evaluates to true or false. Database Desktop creates for a dBASE table an index that points only to values that meet the filter's requirements. For example, if you create the subset condition expression State=CA, you tell Database Desktop to create an index on those values in the State field that match the value CA.

You create a subset condition expression on a dBASE table from the Create Table dialog box or the Restructure Table dialog box.

Follow the instructions in To create an index on a dBASE table.

To create a subset condition expression, enter the expression in the Subset Condition (filter) Expression text box.

To use field names in a subset condition, position the insertion point in the appropriate text box and click the field you want in the Fields list. For example, to create the expression index FIRST_NAME + LAST_NAME, position the insertion point in the Subset Condition (filter) Expression text box, then click FIRST_NAME in the Fields list. FIRST_NAME appears in the text box. Enter + and click LAST_NAME in the Fields list.

To create an index on a dBASE table

See also

To define a field or group of fields as an index,

1. Display the structure of the table in the Create dBASE Table dialog box or the Restructure dBASE Table dialog box.
2. Select Indexes from the Table Properties drop-down list.
3. Choose Define to open the Define Index dialog box. The Fields list displays the fields you can use in the index.
4. Select the field you want to create the index on from the Field list. Database Desktop adds it to the Indexed Field box.
5. Check the options you want: Unique, Maintained, or Descending.

   See the Define Index dialog box for a description of these options.
6. If you want an expression index, click Expression Index and add an expression in the Expression Index text box.

   To use field names in an expression index, position the insertion point in the appropriate text box and click the field you want in the Fields list.

   For example, to create the expression index FIRST_NAME + LAST_NAME, position the insertion point in the Expression Index text box, then click FIRST_NAME in the Fields list. FIRST_NAME appears in the text box. Enter + and click LAST_NAME in the Fields list.

   See Creating a dBASE expression index for more information.
7. Type in a Subset Condition (filter) Expression if you want one.

   You can enter fields as described in step 6.

   See Creating a subset condition expression for more information.
8. Choose OK to open the Save Index As dialog box, where you can enter an Index File Name and Index Tag Name.

About borrowing structures

See also

Sometimes you might want to create a new table that is similar (or identical) in structure to an existing table. You can borrow the structure from the existing table and change it to meet your needs.

In addition to borrowing the structure of a table, you can also borrow its primary or secondary indexes, its validity check definitions, its referential integrity, its table lookup definitions, or any combination of these options. Use the Options settings in the Select Borrow Table dialog box to specify the definitions you want to borrow with the table.

If you borrow a table's key (the Primary Index option) you must make sure that the keyed field is the first field in the new table's Field Roster.

To borrow a table structure

See also

When creating a table, you can borrow the structure of another table. You must begin from a blank table structure to borrow another table's structure.

To borrow a table structure,

1. In the Create Table dialog box, choose Borrow.

   Database Desktop opens the Select Borrow Table dialog box, which shows you a list of tables in the current directory (by default, the working directory). The list includes only table types that match the type of table you are creating.

2. Select the source table from the list.

   To borrow from a table not in the current directory, you can choose another directory in the Look In list box. For an explanation of the icons next to the list box, see Select Borrow Table dialog box.

   If the directory you want has an alias, you can choose it in the Alias drop-down list.

3. In the Options area, specify any table properties you want to borrow.
4. Choose Open.

   Database Desktop puts a copy of the selected table's structure in the field specification area of the Create Table dialog box. You can now add data or change the borrowed structure.

About validity checks

See also

In Paradox tables, validity checks are rules imposed on a field to ensure that the data entered in the field meets certain requirements. The way you define a validity check determines what can be entered in a field. Database Desktop provides five kinds of validity checks:

Validity check	Meaning
Required Field	Every record in the table must have a value in this field.
Minimum Value	The values entered in this field must be equal to or greater than the minimum you specify here.
Maximum Value	The values entered in this field must be less than or equal to the maximum you specify here.
Default Value	The value you specify here will be entered in this field automatically, if no other value is entered.
Picture	You specify a character string that acts as a template for the values that can be entered in this field.

When you save a table, Database Desktop saves validity checks in a file with the table's name and the .VAL file extension.

To create validity checks

See also

When you select a field in the Field Roster, Database Desktop shows its validity checks down the right side of the window. As you select fields, the validity checks change to reflect the constraints for the selected field.

To specify a validity check,

1. Display the structure of the table in the Create Paradox Table dialog box or the Restructure Paradox Table dialog box.
2. In the Field Roster, select the field for which you want to define a validity check.
3. Select Validity Checks from the Table Properties drop-down list.
4. In the panel below, enter the information for any validity checks you want. For details on each type of validity check, see About validity checks.

To view validity checks

See also

To view validity checks for a field,

1. Display the structure of the table in the Create Paradox Table dialog box or the Restructure Paradox Table dialog box,

   or

   Right-click the table in the Project Viewer and choose Info Structure.
2. In the Field Roster, select the field whose validity checks you want to view.
3. Select Validity Checks from the Table Properties drop-down list.
4. Current validity check values appear in the panel below the Table Properties drop-down list. For details on each type of validity check, see About validity checks.

To remove a validity check

See also

To remove a validity check,

1. Display the structure of the table in the Create Paradox Table dialog box or the Restructure Paradox Table dialog box.
2. In the Field Roster, select the field for which you want to remove a validity check.
3. Select Validity Checks from the Table Properties drop-down list.
4. In the panel below, remove any validity checks you want. For details on each type of validity check, see About validity checks.

Example of creating validity checks

See also

Suppose you want the default value of the State/Prov field in Customer.db to be CA. To specify this default value validity check,

1. Display the structure of the table in the Create Paradox Table dialog box or the Restructure Paradox Table dialog box.
2. In the Field Roster, select the State/Prov field.
3. Select Validity Checks from the Table Properties drop-down list.
4. Type CA in the Default text box.

When you insert a new record in the table, Database Desktop automatically enters the value CA in the State/Prov field. You can move to the field and edit the value if you want.

About required fields

See also

You can define required fields for Paradox and SQL tables.

A required field must contain data before the record is inserted into the table. Database Desktop checks that the required field constraint has been met when the record is posted.

If you enter a record in a table that doesn't have a value in a required field, Database Desktop informs you that the validity check has failed. You can't move off the record or leave Edit mode until you've entered a value in the required field.

You can place a required field validity check on any field type.

To create required fields

See also

Paradox tables

To define a required field for a Paradox table,

1. Display the structure of the table in the Create Paradox Table dialog box or the Restructure Paradox Table dialog box.
2. In the Field Roster, select the field you want to define as a required field.
3. Select Validity Checks from the Table Properties drop-down list.
4. Check the Required Field check box.

SQL tables

To define a required field for an SQL table, use the Create Table dialog box:

1. In the Field Roster, select the field to be required.
2. Click the Required Field check box.

Note: To clear a required field definition in Paradox and SQL tables, click the Required Field check box to remove the check.

▪

About minimum and maximum values

See also

Use a minimum-value validity check to define the minimum allowable value for a field. Use a maximum-value validity check to define the maximum allowable value for a field.

You can use minimum-value and maximum-value validity checks only for alpha, number, short, long integer, money, timestamp, time, and date field types. You can use only a minimum validity check on an autoincrement field. You cannot use minimum-value and maximum-value validity checks on BC dates; instead, you can define a picture validity check on BC dates.

You can specify an initial value for an autoincrement field using a minimum validity check. Enter the initial field value in the Minimum Value text box. You can do this only when creating a new table.

To create minimum and maximum values

See also

To specify a minimum or maximum value for a field,

1. Display the structure of the table in the Create Paradox Table dialog box or the Restructure Paradox Table dialog box.
2. In the Field Roster, select the field.
3. Select Validity Checks in the Table Properties drop-down list.
4. Type the minimum or maximum value in the text box. To define a range, enter both minimum and maximum values.

   When you define a numeric minimum or maximum, you must use the number format currently selected in the Windows Control Panel. During data entry, however, you can use any format and the validity check still works.

▪

About default values

See also

Database Desktop automatically enters the value you define as a field's default in each record of the table as soon as you insert the record. For example, if most of your customers are located in the United States, you can define USA as the default value for the Country field in Customers. When you insert a new record, it appears with the value USA already in the Country field.

You can override the default value by moving to the field and typing a different value. You can also delete the default value and leave the field blank, unless it also has a required-field validity check.

You can use default value validity checks for alpha, number, short, long integer, money, logical, and date field types (including date, time, and timestamp).

To create default values

See also

To specify a default value for a field,

1. Display the structure of the table in the Create Paradox Table dialog box or the Restructure Paradox Table dialog box.
2. In the Field Roster, select the field.
3. Select Validity Checks from the Table Properties drop-down list.
4. Type the default value in the edit box.

- When you enter numeric values as a default, you must use the number format currently selected in the Windows Control Panel.
- You can use the TODAY operator to define today's date as the default value in a date field.

About pictures

See also

A picture acts as a template that formats the value you enter in a field.

For example, if you specify the picture (###)###-#### (a common template for U.S. phone numbers) and enter the value (4085551234, Database Desktop formats the value into (408)555-1234. For other examples, see Examples of pictures.

You can choose a standard picture when creating a validity check, or create a custom picture. For instructions, see To use standard pictures and To create custom pictures.

See Picture string characters for a table of the characters you can use in a picture and their meanings. If you use any printable (visible) character in a picture string different from those listed in the table, Database Desktop treats it as a constant.

When you enter a value in a field that has a picture validity check, and you come to a point at which a constant is specified, Database Desktop automatically enters the constant. For example, if you create the picture (408)###-#### and then type (5551234 in the field, Database Desktop inserts (408)555-1234 in the table.

If you create a picture validity check when restructuring a table that contains data, Database Desktop does not reformat existing data to match the picture nor validate existing data to check that it matches.

Picture string characters

See also

You can use these characters in a picture string:

Character	Stands for
#	Numeric digit
?	Any letter (uppercase or lowercase)
&	Any letter (convert to uppercase)
~	Any letter (convert to lowercase)
@	Any character
!	Any character (convert to uppercase)
;	(semicolon) Interpret the next character as a literal, not as a special picture-string character.
*	Any number of repeats of the following character
[abc]	Optional characters a, b, or c
{a,b,c}	Optional characters a, b, or c

Examples of pictures

See also

Following are some examples of valid pictures. For more examples, use the Assist button as described in To use standard pictures.

Picture	Description
#&#&#&	Canadian postal code; for example, 1A2B3C
#####[-####]	U.S. postal code; for example, 12345 or 12345-6789
*!	Any entry; all letters will be in uppercase
{Yes,No}	Either "Yes" or "No"

To use standard pictures

See also

To specify a standard picture string for a selected field,

1. Display the structure of the table in the Create Paradox Table dialog box or the Restructure Paradox Table dialog box.
2. In the Field Roster, select the field.
3. Select Validity Checks from the Table Properties drop-down list.
4. Choose Assist.

   The Picture Assistance dialog box appears.
5. Choose a picture from the Sample Pictures list.

   A description of the picture appears in the message area of the dialog box.
6. Choose Use to place the sample in the Picture text box.
7. Choose OK.

To create custom pictures

See also

To create a custom picture,

1. Display the structure of the table in the Create Paradox Table dialog box or the Restructure Paradox Table dialog box.
2. In the Field Roster, select the field.
3. Select Validity Checks from the Table Properties drop-down list.
4. Choose Assist.

   The Picture Assistance dialog box appears.
5. Do either of the following:

- Select a picture from the Sample Pictures list and choose Use to place the sample in the Picture text box.

  You can modify this standard template. If you make a mistake, choose Restore Original to return to the standard template you copied to the Picture text box.
- Type the picture string characters in the Picture text box.

6. Choose Verify Syntax to check the syntax of the picture you created.

   If the syntax is correct, a message appears in the Picture Assistance dialog box.
7. If you want, enter a value in the Sample Value text box and choose Test Value to test the picture.
8. To keep the picture for reuse in other tables, choose Add To List. In the Save Picture dialog box, type a description for the picture in the Description text box. Then, choose OK.

   The picture in the Picture text box is added to the Sample Pictures list.
9. Choose OK to use the picture and close the dialog box.

▪

About table lookups

See also

The table lookup feature lets you

- Refer to another table to look up acceptable values for a field
- Automatically copy values from the lookup table to the table you are editing (automatic fill in)
- Require that the values you enter into a field exist in the first field of another table

When you specify a lookup table for a field, you are saying the field can contain only values that exist in the first field of another table you specify, the lookup table. You also specify whether the person entering data in the field will be allowed to see the lookup table and copy values from it, or will be required to match the lookup table's values without being able to see them.

The major advantage of table lookup is its ability to automatically enter correct values in your table. For information on using table lookup, see Rules for table lookups and To use table lookup.

The difference between table lookup and referential integrity

Table Lookup is primarily a data entry tool. It is provided to help enter data that already exists in another table. To establish a more powerful tie between two tables, define a referential integrity relationship. While table lookup ensures that data is copied accurately from one table to another, referential integrity ensures that the ties between like data in separate tables cannot be broken. For more information on referential integrity, see About referential integrity.

Rules for table lookups

See also

Follow these rules when setting up a lookup table:

- The lookup table contains data you want to copy to another table. That data must be in the lookup table's first field.
- The field that you're defining as a table lookup must be the same field type and size as the first field of the lookup table.
- For best performance, the lookup table should be keyed. See About primary indexes (key) for more information.
- You can use a table lookup across different directories. When you define table lookup on a table from a different directory, Database Desktop stores the full path to the table. If you move your lookup table to a different directory, you must recreate the same path or redefine the table lookup.

To create table lookups

See also

To specify a lookup table for a field,

1. Display the structure of the table in the Create Paradox Table dialog box or the Restructure Paradox Table dialog box.
2. Select Table Lookup from the Table Properties drop-down list.
3. Choose Define. The Table Lookup dialog box appears.
4. In the Fields list, select the lookup field. Click the Add Field arrow → or press Alt+A. The field name appears in the Field Name box.
5. In the Lookup Table panel, specify the name of the table to use as the lookup table. Choose Drive (or Alias) to choose an alias or your private directory, or choose Browse to open the Select File dialog box.

   You can specify a table from a different directory, but you'll need to recreate the lookup if you ever move the lookup table (for details, see Rules for table lookups).
6. Click the Add Field arrow ←. The field name of the first field in the lookup table appears in the Lookup Field box.
7. Select the option you want in Lookup Type.

   See Table Lookup dialog box for more information.
8. Select the option you want in Lookup Access.

   See Table Lookup dialog box for more information.

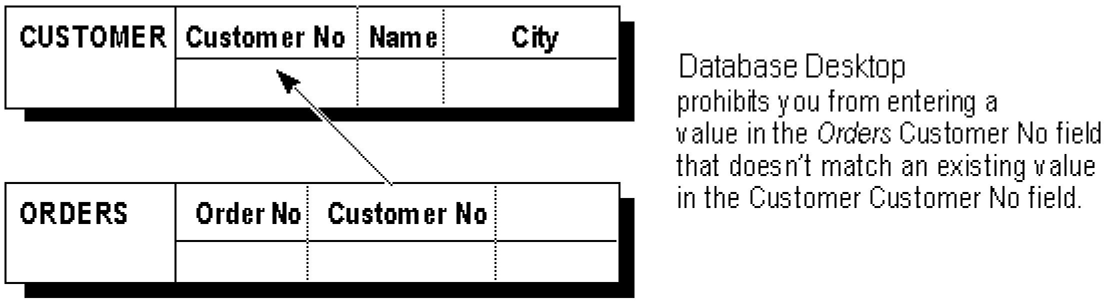
About referential integrity

See also

Referential integrity means that a field or group of fields in one table (the "child" table) must refer to the key of another table (the "parent" table). Only values that exist in the parent table's key are valid values for the specified field(s) of the child table.

Using referential integrity, Database Desktop checks the validity of a value before accepting it in the referential integrity table. For example, if you establish referential integrity between Customer and Orders on their Customer No fields, then enter a value in the Customer No field of Orders, Database Desktop searches the Customer No field of Customer and

- Accepts the value in Orders if it exists in Customer
- Rejects the value in Orders if it doesn't exist in Customer


CUSTOMER
Customer No
Name
City
ORDERS
Order No
Customer No
Database Desktop prohibits you from entering a value in the *Orders* Customer No field that doesn't match an existing value in the Customer Customer No field.

Database Desktop lets you establish referential integrity for any file type that supports it. You cannot establish referential integrity between .DBF files, Paradox 3.5 tables, or tables that do not have a key. You can use .DB files and also some SQL server tables if you need referential integrity. See your server documentation to determine if your table type supports referential integrity.

Referential integrity and indexes

See also

When you create or modify a referential integrity relationship, Database Desktop creates an index on the referential integrity fields if it does not already exist.

Database Desktop names the index with the name of the field (if it's a single-field definition) or the name you gave the referential integrity (if it's a multiple-field definition). The index appears in the list of secondary indexes when you choose Secondary Indexes from the Table Properties list in the Create Table or Restructure Table dialog box.

If you delete the referential integrity, Database Desktop does not automatically delete this index. You must delete it manually.

▪

Self-referential integrity

See also

A referential integrity relationship between a field in a table and the same table's key field is called a self-referential integrity relationship.

For example, suppose you are using a table of employees keyed on the Employee ID field. If this table has a Supervisor field, you may want to create a self-referential integrity relationship between Supervisor and Employee ID, because the supervisors are also employees.

When you create a self-referential integrity relationship,

- You must use the Prohibit update rule in the Referential Integrity dialog box.
- You cannot create a circular reference. That is, you cannot create a relationship in which a field refers to itself.

Referential integrity guidelines

See also

Follow these guidelines when you establish referential integrity:

- You can establish referential integrity only between like fields that contain matching values.
  For example, you can establish referential integrity between the sample Customer.db and Orders.db tables on their Customer No fields. The field names do not matter as long as the field types and sizes are identical.
- You can establish referential integrity only between tables in the same directory.
- The referential integrity parent table must be keyed.
- If you define referential integrity on a table that already contains data, some existing values may not match a value in the parent's key field. When this happens, Database Desktop places the existing records that do not match into the temporary Keyviol table in your private directory.

To create referential integrity

See also

To define a referential integrity relationship,

1. Display the structure of the referential integrity child table in the Create Paradox Table or the Restructure Paradox Table dialog box.
2. Select Referential Integrity from the Table Properties drop-down list.
3. Choose Define to open the Referential Integrity dialog box.
4. Select a referential integrity parent table from the Table list and click the Add Field arrow.

   The table's key field appears in the Parent's Key box.
5. Select the child table's field in the Fields list and click the Add Field arrow.

   The field name appears in the Child Fields box.

- If you choose a field that is not the same logical type as the parent's key field, Database Desktop displays a message and doesn't add the field. In most cases, this means the field types must be identical; however, autoincrement and long integer are of the same logical type.
- If you make a mistake and add the wrong field, click the Remove Field arrow or press Alt+R.
- If the parent table has a composite key, add fields to match all of the parent's key fields.

6. Select the update rule you want.
7. Specify whether you want to use strict referential integrity.
8. Choose OK to display the Save Referential Integrity As dialog box.
9. Complete the dialog box and choose OK.

To change referential integrity

See also

You can change the following attributes of a referential integrity relationship:

- The update rule
- The Strict Referential Integrity setting

Database Desktop must obtain locks on all tables involved in a referential integrity relationship when you modify it.

To change a referential integrity relationship,

1. Choose Tools|Utilities|Restructure.

   Database Desktop displays the Select File dialog box.
2. Enter the name of the referential integrity child table in the File Name field, or use the dialog box to navigate to it, then choose Open.

   Database Desktop displays the Restructure Table dialog box.
3. Select Referential Integrity from the Table Properties drop-down list.

   Database Desktop displays the referential integrity relationships in the list box on the right.
4. Select the relationship you want to change, then choose Modify.

   Database Desktop displays the Referential Integrity dialog box.
5. Make the desired changes, then choose OK to close the Referential Integrity dialog box and return to the Restructure Table dialog box.
6. Choose Save to save the new table structure and to close the Restructure Table dialog box.

To delete referential integrity

See also

To delete a referential integrity relationship,

1. Choose Tools|Utilities|Restructure.

   Database Desktop displays the Select File dialog box.
2. Enter the name of the referential integrity child table in the File Name field, or use the dialog box to navigate to it, then choose OK.

   Database Desktop displays the Restructure Table dialog box.
3. Select Referential Integrity from the Table Properties drop-down list.

   Database Desktop displays the referential integrity relationships in the list box on the right.
4. Select the relationship you want to delete, then choose Erase.

   Database Desktop deletes the referential integrity relationship.
5. Choose Save to save the new table structure and to close the Restructure Table dialog box.

▪

About password security

See also

You can ensure that the table you create is protected from access by unauthorized users. This is especially important in a multiuser environment. Not only can you establish a password for the table as a whole, you can assign specific rights to the table or individual fields.

Once you specify password security, only those users who know the password can access the table. This includes you, so do not forget your password! Whenever users try to access a password-protected table, Database Desktop prompts them to supply the password (if they haven't already done so).

Types of passwords

Database Desktop provides two types of passwords:

- Master passwords control all access to an entire table. You must specify a master password before creating additional access restrictions.
- Auxiliary passwords provide different levels of access privileges for different users in a group.

Typically, one person▪such as a database administrator

▪has access to master passwords. A group of users who need to perform different tasks with the table have different auxiliary passwords that provide different levels of access.

Using passwords

See also

You can define passwords for your tables from the Create Paradox Table dialog box or the Restructure Paradox Table dialog box. When you try to open a password-protected table, Database Desktop prompts you for the password. You must enter the password to open the table.

Suppose you close the table, then attempt to open it again. If you have not exited Database Desktop, you will be allowed to open the table without giving the password another time. Database Desktop remembers that you accessed the table previously and opens the table again. Database Desktop releases all passwords when you exit the program.

Releasing passwords

To release a password without exiting Database Desktop, choose Tools|Utilities|Passwords. The Enter Password(s) dialog box opens.

Enter the password you want to release from Database Desktop's memory in the Password text box. Asterisks (*) represent the characters you type. Choose Remove to remove this password from Database Desktop's memory. You will be required to give the password the next time you open the table.

You can choose Remove All to remove all passwords from Database Desktop's memory. This means any table you have opened using a password, then closed, will again be protected. (Tables that are still open are not affected.)

Using one password for several tables

If you assigned the same password to several tables, you can use the Enter Password(s) dialog box to give Database Desktop the password once to access all applicable tables. Type the password and choose Add or OK (or press Enter).

To create a master password

See also

To create a master password,

1. Display the structure of the table in the Create Paradox Table dialog box or the Restructure Paradox Table dialog box.
2. Select Password Security from the Table Properties drop-down list.
3. Choose Define.

   The Password Security dialog box appears.
4. Type the password you want in the Master Password text box. You'll see asterisks (*) representing the characters you type. A password can be from 1 to 15 characters long and can contain spaces. Passwords are case-sensitive.
5. Verify the password by typing it again in the Verify Master Password text box. Again, you'll see asterisks in place of the characters you type.
6. Choose OK.

- If the two passwords are identical, Database Desktop saves the password and closes the Password Security dialog box.
- If the two passwords aren't identical (including capitalization), you'll see an error message prompting you to enter either one of them again.

If you want more specific security, you can choose Auxiliary Passwords from the Password Security dialog box.

You can assign the same password to several tables. For details, see Using passwords.

To change a master password

See also

To change a master password,

1. Display the structure of the table in the Restructure Paradox Table dialog box.
2. Select Password Security from the Table Properties drop-down list.
3. Choose Modify.

   The Password Security dialog box appears.
4. Choose Change.
5. Type the new password in the Master Password text box.
6. Verify the password by typing it again in the Verify Master Password text box.
7. Do either of the following:

- Accept the new password by choosing OK.
- Cancel the new password and restore the previous one by choosing Revert, then choose Cancel to close the Password Security dialog box.

To delete a master password

See also

To delete a master password,

1. Display the structure of the table in the Restructure Paradox Table dialog box.
2. Select Password Security from the Table Properties drop-down list.
3. Choose Modify.

   The Password Security dialog box appears.
4. Choose Delete.
5. Choose OK to close the Password Security dialog box.

About auxiliary passwords

See also

Auxiliary passwords use table rights and field rights to provide different levels of access privileges for different users in a group.

- Table rights determine the overall level of access to a table.
- Field rights determine the level of access to an individual field within the table.

The type of table rights you specify for a user determines the type of field rights you can specify for that user, as shown in the following table:

Table rights	Possible field rights
All	All
Insert & Delete	All
Data Entry	All, Read Only, or None
Update	All, Read Only, or None
Read Only	All, Read Only, or None

To create an auxiliary password

See also

To specify an auxiliary password,

1. Choose Auxiliary Passwords in the Password Security dialog box.
2. Type the auxiliary password you want to assign in the Current Password text box.
3. Select the level of table rights for the password from the Table Rights panel.
4. Select a field in the Field Rights panel, then click the Field Rights button to assign the field rights (All, Read Only, or None) for the password.
5. Choose Add to place the password in the Passwords list.

   If you want to clear the auxiliary password and start over, choose New.
6. Repeat the process to specify as many auxiliary passwords as you need.
7. Choose OK to save the auxiliary passwords and return to the Password Security dialog box.

To change an auxiliary password

See also

To change an auxiliary password,

1. Choose Auxiliary Passwords in the Password Security dialog box.
2. Select the password you want to change in the Passwords list.
3. Choose Change.

   The password moves into the Current Password box, and its definition appears.
4. Make the modifications to the password.
5. When you are done modifying the password, do either of the following:

- Choose Accept to accept the changes and place the password back in the Passwords list.
- Choose Revert to cancel the changes and place the password back in the Passwords list.

6. Choose OK to save the auxiliary passwords and return to the Password Security dialog box.

To delete an auxiliary password

See also

To delete an auxiliary password,

1. Choose Auxiliary Passwords in the Password Security dialog box.
2. Select the password you want to delete in the Passwords list.
3. Choose Delete.

   The password is removed from the Passwords list.
4. Choose OK to save the auxiliary passwords and return to the Password Security dialog box.

▪

About table language drivers

See also

A table's language driver determines the table's sort order and available character set. You choose a default language driver for Paradox and dBASE tables from the BDE Configuration Utility. (Refer to the BDE Configuration Utility Help system for more information.)

Database Desktop uses table language drivers to:

- Open any Paradox table with the correct language driver. This includes opening tables with different language drivers in the same Database Desktop session.
- Open dBASE tables with the correct language driver.
- Create tables generated by copy operations using the language driver of the copied table.
- Create tables generated by import operations with the default language driver of the file format (Paradox or dBASE).
- Determine the language driver of a query's Answer table using the language driver of the first (topmost) query image in the query.

To choose a table language driver

See also

A table's language driver determines the table's sort order and available character set.

Caution: If you change a table language driver when restructuring a table, you risk losing special characters in the table.

To change the default table language driver,

1. Display the structure of the table in the Create Table dialog box or the Restructure Table dialog box.
2. Select Table Language from the Table Properties drop-down list.
3. Choose Modify.

   The Table Language dialog box appears.
4. Choose the language driver that you want to use from the Language drop-down list.
5. Choose OK to close the Table Language dialog box.

▪

About restructuring tables

See also

You can change the structure of a table with the Restructure Table dialog box, even if the table already has data in it.

The Restructure Table dialog box lets you:

- Add fields
- Delete fields
- Rearrange field order
- Change field types
- Change field sizes
- Change indexes
- Borrow the structure of an existing table

In addition, when you restructure a Paradox table you can:

- Change the key, lookup tables, validity checks, secondary indexes, and referential integrity (even if you do not change the basic table structure)
- Establish table language drivers, and passwords

Before restructuring a table, make sure no queries are running that use the table. If you or any other user (in a multiuser environment) have such a document open, you will not be able to restructure the table.

If you want to rename a table but not restructure it, use Tools|Utilities|Rename.

Restructuring on a network

See also

When you restructure a table on a network or with more than one session of Database Desktop open, Database Desktop automatically places a lock on the table. This means other users cannot access the table during the restructuring.

If another application has started an operation using the table you want to restructure, you cannot begin restructuring until that application finishes working with the table.

Restructure warnings

See also

When you restructure a table, you might make changes that could result in a loss of data. Changes such as shortening field sizes, creating validity checks, or changing field types can cause existing data to become invalid. Whenever this is the case, Database Desktop opens the Restructure Warning dialog box, upon leaving the Restructure Table dialog box.

Temporary tables created when restructuring

See also

Restructuring sometimes results in the creation of temporary tables, such as a Problems table, that Database Desktop uses to store records that are incompatible with the table as you've restructured it.

Database Desktop numbers these temporary tables consecutively (up to 99) and stores them in your private directory. For example, if you restructure twice, and both operations cause data loss, Database Desktop creates both a Problems and a Problem1 table.

Temporary tables are deleted at the end of a session. If you do not want a temporary table deleted at the end of a session, you must rename it. All temporary tables are stored in your private directory (:PRIV:).

Keyviol

If you add a primary key to a table that was previously unkeyed or had different keys, you might cause key violations. This means data already entered into the table violates the rules established by the new key. Database Desktop moves the key-violating records to a temporary table called Keyviol, located in your private directory.

Database Desktop deletes key-violating records from your table. You can change the records in Keyviol so they comply with the key requirements, and then add them back to your original table using Tools|Utilities|Add.

For more information see The effect of restructuring tables on key fields.

Rules for restructuring

See also

Follow these rules when restructuring a table:

- You cannot change a table's type. For example, you cannot change a Paradox table into a dBASE table when you restructure. You can choose Tools|Utilities|Copy to copy a table of one type into a table of another type.)
- If you restructure a table that was created in Paradox for DOS in such a way that Database Desktop must convert it to a Paradox for Windows table, the Restructure Warning dialog box warns you of the conversion and asks you to confirm it.
- If you add a primary key to a table that was previously unkeyed or had different keys, you might cause key violations. This means data already entered into the table violates the rules established by the new key. Database Desktop moves the key-violating records to a temporary table called Keyviol, located in your private directory.

  If there is already a Keyviol table, Database Desktop adds a number to the new temporary table, so it might appear as Keyviol1 or Keyviol2. Database Desktop can create up to 100 temporary tables of the same name (the first is not numbered and the last is number 99).

  Database Desktop deletes key-violating records from your table. You can change the records in Keyviol so they comply with the key requirements, and then add them back to your original table using Tools|Utilities|Add.

  For more information see The effect of restructuring tables on key fields.
- If you change a field's type, and Database Desktop cannot convert some of the data in the field to the new type, Database Desktop prompts you to confirm the change. If you do, Database Desktop moves the records containing data that could not be converted into a special temporary table called Problems.

  You can change the records in Problems so they comply with the new structure of the table, and then add them back into the table using Tools|Utilities|Add.
- If you decrease a field's size, Database Desktop prompts you to trim existing data in the Restructure Warning dialog box. If you choose not to trim data, Database Desktop moves the records containing data that does not fit in the new field size to the Problems table.
- If you add or change a validity check, you have the option of enforcing the new validity check on existing data (make this choice from the Restructure Warning dialog box). If you choose to enforce the new validity check on existing data, and any data that does not comply with it, Database Desktop places the non-compliant data in the Keyviol table. Database Desktop does not do this if the validity check is a picture. You can change the records in Keyviol and then add them back to the table using Tools|Utilities|Add.
- If you add a new field that has a default validity check on it, and choose to enforce the validity check on existing data, Database Desktop creates the new field and places the default value in each record of the table. If you define a default validity check on an existing field that contains data, Database Desktop does not overwrite the existing data with the new default value.
- If you change a table's language driver when restructuring a table, you risk losing any special characters that may exist in the table. Database Desktop converts table structure information such as field names and index names, but not the data in the table. To convert the data in a table:
- You can create a new table by choosing Save As from the Restructure Table dialog box.
- Choose Tools|Utilities|Add and add the records to the new table.

To restructure a table

See also

1. Open the Restructure Table dialog box, using one of these techniques:

- From the Project Viewer, click the Tables icon, select a table, then right-click and choose Restructure.
- From the Table window, choose Table|Restructure.
- From the Desktop, choose Tools|Utilities|Restructure, then specify the table name in the Select File dialog box and choose Open.

2. Change any field and key information you want to change.
3. Change any table properties you want to change.

   Note: If you want, you can borrow a table structure from an existing table. Choose Borrow in the Restructure Table dialog box. For details, see About borrowing structures.

4. When all restructuring changes are complete, choose Save to save the table.

   You can choose Save As to save the restructured table with a different name.

Note: For more information about the settings you can change when restructuring, see the dialog box topic for the type of table you are restructuring:

Restructure Paradox Table dialog box

Restructure dBASE Table dialog box

Restructure INTRBASE Table dialog box

Restructure INFORMIX Table dialog box

Restructure ORACLE Table dialog box

Restructure SYBASE Table dialog box

Restructure Table dialog box (other SQL)

Shortening fields

See also

When you shorten a field that already has data in it, you may lose some data. When this is the case, Database Desktop displays the Restructure Warning dialog box, which lets you choose whether to trim existing data, or to save records that contain data too long for the new field size in the Problems table.

▪

Deleting fields

See also

Deleting fields from an existing table is different than deleting fields when you create a new table:

- Deleting a field usually results in a loss of data (unless the field is empty). Database Desktop displays a dialog box warning you of the loss and asking you to confirm the deletion.
- If a field you delete from a table appears as a field object in any form, report, or query, then the next time you open the form, report, or query, you must either redefine or delete the field object.
- If you delete a key, you must also either delete any secondary indexes, or convert them to non-maintained.

▪

Rearranging fields

See also

Rearrange field order in either the Create Table dialog box or the Restructure Table dialog box. In the Field Roster, click the number of the field you want to move and drag it to the position you want it to occupy.

You can place a field in the following locations:

- Between the rows of existing fields
- In the row above the first field
- In the row below the last field

You cannot move fields in a way that violates the rules for key fields. See About primary indexes (key fields) for more information.

Converting a non-keyed field to a keyed field

See also

When you convert a field from non-keyed to keyed, remember that keyed fields must be consecutive and start with the first field in the Field Roster.

You can move a field if necessary. See Rearranging fields.

For more information on restructuring non-keyed and keyed fields, see Temporary tables created when restructuring and The effect of restructuring tables on key fields.

▪
Packing a table

See also

To pack a table, check Pack Table in the Restructure Table dialog box when you restructure the table. Packing differs for Paradox and dBASE tables.

Paradox tables

Packing a Paradox table reclaims disk space used by deleted records.

dBASE tables

Packing a dBASE table permanently removes records that are marked for deletion from the table.

To add fields

See also

When you add fields to an existing table, Database Desktop does not automatically add those fields to any forms, reports, or queries associated with the table. If you want the new fields added to associated objects, you must explicitly add them.

To add a field to a table,

1. Open the Restructure Table dialog box.
2. Use the arrow keys to move to a new row in the Field Roster, or press Ins to insert a new field above the current field.
3. Enter the name of the field under Field Name.
4. Enter the type of field under Type. Press the Spacebar to choose from a list of field types.
5. Enter the size of the field under Size, if it is necessary for your field type.
6. If you are creating a Paradox table, specify whether the field is a key. Move to the Key column and follow the instructions on the screen.
7. If you are creating a dBASE table, specify the number of decimal places in Dec, if it is necessary for your field type.

To delete fields

See also

When you delete a field from an existing table, Database Desktop unbinds the field from previously created forms and reports.

Since a deletion can cause loss of data, when you choose Save or Save As, a dialog box appears to let you confirm the deletion or cancel the operation.

To delete a field from a table,

1. Open the Restructure Table dialog box.
2. Select the row number of the field you want to delete, then press Ctrl+Del.

   The field is removed from the table specification.

To rename fields

See also

To rename a field in a table,

1. Open the Restructure Table dialog box.
2. Select the field you want to edit.
3. Click the field again to place the insertion point in the field. Edit the field name as desired, using standard editing techniques.

To change validity checks

See also

For rules about changing validity checks, see Rules for restructuring.

To change validity checks,

1. Open the Restructure Paradox Table dialog box.
2. Select Validity Checks from the Table Properties drop-down list.
3. Change any Validity Check settings you want to.
4. When all restructuring changes are complete, choose Save to save the table.
   You can choose Save As to save the restructured table with a different name.

To change table lookup

See also

To change table lookup,

1. Open the Restructure Paradox Table dialog box.
2. Select Table Lookup in the Table Properties drop-down list.
3. You can define a new table lookup or modify an existing one.

- To define a new lookup, choose Define. The Table Lookup dialog box appears. Create the new lookup, then choose OK to return to the Restructure Paradox Table dialog box.
- To change an existing table lookup, select the lookup table to change. Then, choose Modify to open the Table Lookup dialog box. Change any settings you want to, then choose OK to return to the Restructure Paradox Table dialog box.

4. When all restructuring changes are complete, choose Save to save the table.

   You can choose Save As to save the restructured table with a different name.

To change secondary indexes

See also

To change secondary indexes,

1. Open the Restructure Paradox Table dialog box.
2. Select Secondary Indexes in the Table Properties drop-down list.
3. You can define a new secondary index or modify or erase an existing definition.

- To define a new secondary index, choose Define. The Define Secondary Index dialog box appears. Change the definition, then choose OK to return to the Restructure Paradox Table dialog box.
- To change an existing secondary index, select the index you want to change in the list and choose Modify. Then, change the definition in the Define Secondary Index dialog box. Choose OK to return to the Restructure Paradox Table dialog box.
- To delete an existing secondary index, select the index you want to change in the list and choose Erase. The index is removed from the list.

4. When all restructuring changes are complete, choose Save to save the table.

   You can choose Save As to save the restructured table with a different name.

To change password security

See also

To change password security,

1. Open the Restructure Paradox Table dialog box.
2. Select Password Security in the Table Properties drop-down list.
3. Choose Define to add new passwords or choose Modify to change existing passwords.
   The Password Security dialog box appears.
4. Add or change passwords, then choose OK to return to the Restructure Paradox Table dialog box.
5. When all restructuring changes are complete, choose Save to save the table.
   You can choose Save As to save the restructured table with a different name.

To change table languages

See also

If you need to change the table language of an existing table that has data, you must first duplicate the table's structure with the language driver you want, then append the original table's data to the new table.

To change table languages,

1. Choose File|New|Table and choose the table type you want.
2. In the Create Table dialog box, choose Borrow to borrow the structure of the table that you want to change.
3. In the Select Borrow Table dialog box, check all options you want to borrow (usually this is all options).
4. Specify the new table's language driver by choosing Table Language from the Table Properties drop-down list. Choose Modify to open the Table Language dialog box. Choose the table language you want.
5. Save the new table.

   The new table uses the language driver you specified, and all its structure information (such as field names) is in the format of that language driver.
6. Move the data from the original table to the new table using Tools|Utilities|Add. Choose to append the data. When the data is appended to the new table, it is transliterated to the format specified by the new language driver. See About adding records for more information about using the Add command.

To change field types

See also

To change field types,

1. Open the Restructure Table dialog box.
2. In the Field Roster, select the Type column of the field you want to change.
3. Type the symbol for the field type or select from the drop-down list. You can get the list two ways:

- Right-click the Type column.
- Press Spacebar.

If the change causes data loss, Database Desktop prompts you to confirm it. If you confirm the change, Database Desktop writes the records containing data that could not be converted to a temporary table called Problems.

You can change the records in the Problems table so they comply with the new structure, then add them back into the table using Tools|Utilities|Add.

Compatible Paradox field types

See also

When converting a Paradox field from one type to another, use the following chart to determine field type compatibility.

To	A	N	$	D	S	M	F	B	G	O	L	I	T	@	#	+	Y
From A	Y	P	P	P	P	Y					P	P	P	P	P		Y
From N	Y	Y	Y		P						Y	Y			Y	<	
From $	Y	Y	Y		Y						Y	Y			Y		
From D	Y			Y										Y			
From S	Y	Y	Y		Y						Y	Y			Y	<	
From M	Y					Y	Y	Y									
From F						Y	Y	Y									
From B								Y									
From G								Y	Y								
From O								Y		Y							
From L	Y	Y	Y		Y						Y	Y			Y	<	
From I	Y	Y	Y		Y						Y	Y			Y		
From T	Y												Y	Y			
From @	Y			Y									Y	Y			
From #	Y	Y	Y		Y						Y	Y			Y		
From +	Y	Y	Y		Y						Y	Y			Y	Y	
From Y	Y																Y

Y:	Database Desktop allows the conversion, but may trim data. If Database Desktop must trim data, you will see the Restructure Warning dialog box, which asks you to confirm the conversion.
Blank:	The field type conversion is not allowed.
P:	The conversion is allowed, but might generate the Problems table.
<:	Conversion to autoincrement is allowed only from a single-field key containing data that is <2147483647

Alpha field conversions

See also

The result of converting another field type to an alpha field varies. All formatting and other definitions associated with the other field type are lost.

When you convert a field of another type to an alpha field, you must specify a size for the field. If some data already in the field contains more characters than the newly specified length of the alpha field, you can trim the data or move records containing such data to the Problems table.

If you convert between an alpha field and a date, number, short, or money field, make sure the settings in your Windows Control Panel match the settings in IDAPI32.CFG.

▪

Number, money, short, and long integer field conversions

See also

In a Paradox table, you can convert a money, long integer, BCD, autoincrement or short field to a number field. In fact, you can convert among all of these field types without loss of data, except when a value is too large for a short field or includes decimals. In that case, you can either trim the values, or have Database Desktop write records containing those values to the temporary Problems table.

You can convert an alpha field to a number field if it contains no data inconsistent with a number. If data in the field is inconsistent with a number field, you must do one of the following:

- Have Database Desktop place the records in a Problems table
- Delete the inconsistent data and then make the conversion
- Insert a new field and delete the original field (losing all data)

Autoincrement field conversions

See also

In a Paradox table, you can convert existing number, short, and long integer fields to autoincrement fields without losing data only if the number, short, or long integer field is the table's single-field primary index (key). This ensures that the data to be converted to an autoincrement field meets the requirements of being unique and sorted in ascending order.

Date field conversions

See also

In a Paradox table, you can convert alpha and timestamp fields to date fields. Database Desktop saves any invalid data in a Problems table. If any record contains data in that field that cannot be interpreted as a date, Database Desktop removes the record and writes it to the temporary Problems table.

Here are examples of what kinds of alpha strings can and cannot be converted to dates:

Can be converted	Cannot be converted
7/04/1776	July 4, 1776
3/30/91	The 30th of March, 1991
25-Dec-1066	Christmas Day, 1066
11-Nov-18	Armistice Day
1.01.2000	New Year's Day, the year 2000
13.06.80	Herb's 29th birthday

If you customize your date format using the BDE Configuration Utility, date values are converted according to your customized settings.

▪

Compatible dBASE field types

See also

You restructure a dBASE table the same way you do a Paradox table. Changing field types of dBASE fields has different consequences for each field type.

Number to character

Data in number fields or float number fields can be converted to text in a character field with no loss of data. However, you cannot perform calculations on numeric data stored in a character field.

Character to number or float number

You can convert a character field to a number or float number field with the following results:

- If the data in the character field is numeric (digits), Database Desktop converts it to a number or float number field with no data loss.
- If the data in the character field is a mixture of text and digits beginning with digits, Database Desktop converts the digits to a number or float number format and deletes all text.
- If the data in the character field is a mixture of text and digits beginning with text, Database Desktop assigns the value 0 to the number or float number field.

Logical to character

Logical values are converted to T or F text values.

Logical to number or float

True values are converted to 1 and False values are converted to 0.

Character to logical

The characters T, t, Y, and y are converted to logical true, and all other values are converted to logical false.

Date to character

You can convert a date value to a text value. The text value will be eight characters in the format MM/DD/YY.

Character to date

You can convert a text value to a date value only if it is an eight-character value in the format MM/DD/YY. Any other value sizes or formats are not recognized as dates and are not converted.

Memo to character

Values that are longer than the size of the character field are trimmed.

About restructuring and referential integrity

See also

When restructuring the parent table in a referential integrity relationship, you might be prohibited from performing certain restructure operations.

To see if the table you are restructuring is the parent in a referential integrity relationship, choose Dependent Tables from the Table Properties drop-down list in the Restructure Paradox Table dialog box. Database Desktop lists all child tables that depend on the table you are restructuring.

The basic rule to remember when restructuring a parent table is that you cannot perform any operation that causes records to be removed from the table. If you remove records from the parent table, you risk orphaning records in the child table. This is in violation of the rules of referential integrity. Each record in the child table must have a valid parent record.

▪

Guidelines for restructuring tables in referential integrity relationships

See also

Follow these guidelines as you restructure tables that are linked by referential integrity:

- If you resize any field in the parent table, you must choose to trim data that does not fit in the new field size, rather than save such data in the Problems table.
- You cannot change the parent table's key definition or the child table's foreign key definition in such a way that will cause records to be saved in the Keyviol table.
- You can change field names, but not types or sizes, of fields that are part of the referential integrity definition.
- You can add a validity check to either table, but you must choose not to enforce it on existing data. (Use the Restructure Warning dialog box to make this choice.) The exception to this rule is the creation of a default validity check on a new field in the table.
- To make a parent table the child of another table, that table and all its existing child tables must be empty. For example, if Orders is the parent table of Stock, you cannot make Orders the child of Customer unless both Orders and Stock are empty.
- When working with tables that contain data, if you link more than two tables by referential integrity you must create the first link to the table that has no parent. For example, to define referential integrity among the Customer, Orders, Lineitem, and Stock tables, you must
    1. First create the link from Orders to Customer.
    2. Then create the link from Lineitem to Orders.
    3. Then create the link from Stock to Lineitem.
- To create a cyclic referential integrity relationship (as in "Table A refers to Table B, which refers to Table C, which refers back to Table A") all the tables must be empty.

Windows on your data

See also

Database Desktop uses the Table window to display data in columns and rows. You can either use the default table format or change rows and columns to get the view you want.

▪

Opening a table

See also

To open a table, use the menu or the Toolbar.

- From the Desktop, choose File|Open|Table, choose a table, and click OK.
- Click the Open Table button on the Toolbar, choose a table, and click OK.

Database Desktop displays the default view of the table.

When you open a table, the menu and Toolbar change to show operations you can perform on the table.

Note: Commands that involve data entry operations are dimmed until you enter Edit mode. See About Edit mode for information about working in the Table window in Edit mode.

Navigation buttons

See also

Use the navigation buttons on the Toolbar to move quickly among a table's records:

⏮	First record of the table
⏪	Previous set of records
◀	Previous record of the table
▶	Next record of the table
⏩	Next set of records
⏭	Last record of the table

A set of records is the number of records currently visible onscreen.

You can also use the Record menu for the above operations.

Table scroll bars

See also

Use the up and down scroll arrows on the vertical scroll bar to scroll through a table one record at a time. Use the left and right scroll arrows on the horizontal scroll bar to scroll through the columns of the table.

Dragging the vertical scroll bar

When you drag the box on the vertical scroll bar to scroll through the records of the table, the records themselves do not move. Instead, Database Desktop displays the range of record numbers as it would appear if you released the scroll box on the Desktop status bar. When you see the range you want to scroll to, release the scroll box. Database Desktop updates the view of the table.

In dBASE tables, the vertical scroll box is always centered vertically.

Note: If the table is keyed, Database Desktop displays the range of values in the key field (or the first field of a composite key) on the status bar as you move the vertical scroll box.

Dragging the horizontal scroll bar

When you drag the box on the horizontal scroll bar to scroll through the fields of the table, the fields themselves do not move. Instead, Database Desktop displays the field name that would appear if you released the scroll box. When you see the field you want to scroll to, release the scroll box. Database Desktop updates the view of the table.

Scroll lock

See also

To lock one or more columns into place as you move horizontally through the table's columns, you can place a scroll lock to the right of the column(s) you want to remain onscreen. You can do this only when the horizontal scroll box is set all the way to the left.

The scroll lock looks like a triangle ◀ in the lower left corner of the Table window. To place a lock, drag the triangle to the right side of the column to lock. The pointer changes to a double-headed arrow and the lock itself changes to two triangles. Position the scroll lock on the right grid line of the right-most column to lock. All columns to the left of the lock remain stationary as you move through the table's columns.

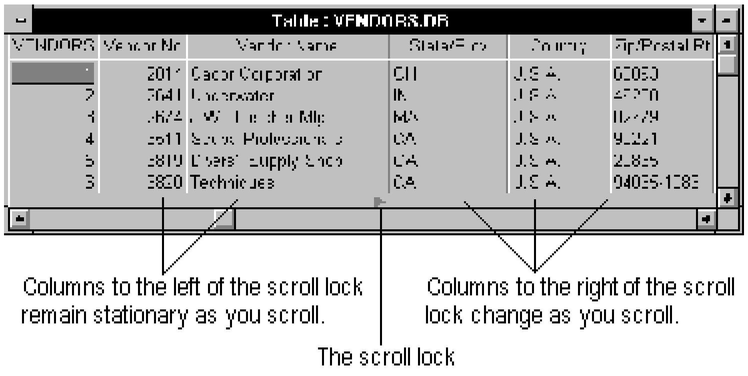


Table : VENDORS.DB

VENDORS	Vendor No	Vendor Name	State/Prov	Country	Zip/Postal Pr
1	2014	Cacor Corporation	OH	U.S.A.	02050
2	2041	[illegible]water	IN	U.S.A.	47270
3	2674	[illegible] Mfg	MA	U.S.A.	02479
4	2511	[illegible] Professionals	CA	U.S.A.	9[illegible]21
5	2819	Divers' Supply Shop	CA	U.S.A.	2[illegible]
3	3820	Techniques	CA	U.S.A.	94035-1083

Columns to the left of the scroll lock remain stationary as you scroll.
The scroll lock
Columns to the right of the scroll lock change as you scroll.

Keyboard actions in Table windows

See also

You can move to different fields and records using the keyboard as follows:

Key	Effect/Action
Left arrow	Selects the field to the left of the selected field. (If the selected field is the first field in the record, selects the last field of the previous record.)
Right arrow	Selects the field to the right of the selected field. (If the selected field is the last field in the record, selects the first field of the next record.)
Down arrow	Selects the same field in the record below the current one.
Up arrow	Selects the same field in the record above the current one.
Home	Selects the first field in the current record.
End	Selects the last field in the current record.
Ctrl+Home	Selects the first field of the first record in the table.
Ctrl+End	Selects the last field of the last record in the table.
PgDn	Displays the next set of records.
PgUp	Displays the previous set of records.
Ctrl+PgDn	Scrolls the window to the next set of fields.
Ctrl+PgUp	Scrolls the window to the previous set of fields.

For additional keys, see Table Operation Shortcuts

▪

About table views

See also

The default view of a table is the way it initially looks when you open it in its Table window. The default view depends on your Windows screen colors and the structure of the table.

But you can change the way your table looks, and the way you view your data.

You can change these features by dragging with the mouse:

- Order of the columns
- Column width
- Heading height
- Spacing between records (row height)
- Placement of a scroll lock on a column

For details, see About dragging with the mouse.

About dragging with the mouse

See also

You can use the mouse to point, click, and drag directly on the object to change. You can directly manipulate the size, shape, or position of most onscreen objects.

The pointer changes shape as the mouse passes over places where you can click and drag to resize or move columns or change the heading or row height.

Pointer	Property	To manipulate
⇕	Heading height	Drag the table name up or down.
⇕	Row height	Drag the line under the first record number in the window up or down.
◀	Horizontal scroll lock	Drag the triangle at the lower left edge of the Table window to the right.
⇔	Column width	Drag the top of the column's right grid line to the left or right.
▭	Order of columns	Drag the column heading to the new location.

To move, resize, or rotate columns

See also

You can move, resize, or rotate columns in a Table window.

To move a column

1. Click and hold the column's heading.

   The pointer changes shape.

2. Drag the column to its new position.

To resize a column

1. Click its right grid line in either the heading area or the top row of data.

   You'll know it's in the correct area when the pointer changes to the double-headed arrow. ▪

2. Drag the grid line to the right or left to increase or decrease the width of the column.

To rotate columns

▪ Select the column to move, and press Ctrl+R.

Database Desktop moves the column to the last position on the right of the table and shifts all other columns one position to the left.

To resize rows

See also

You can resize the table's rows (increase or decrease the row height of all rows) by dragging the line under the first record number in the table.

- Drag the line up to decrease the row height.
- Drag the line down to increase the row height.

Database Desktop resizes all rows to match the row height you specify.

To save table properties

See also

When you resize rows and columns and make other changes to a table, you change its properties.

To save table properties,

- Choose Table|Table View Properties|Save to save all the property changes you make in a Table window.

This saves the appearance of the table as you have changed it. Database Desktop saves data as it is entered, so File|Save and File|Save As are not necessary and are dimmed in the Table window.

Database Desktop saves the properties you define in the .TV file. (Properties for dBASE tables are saved in the .TVF file.) For example, the properties you define for the Customer table are saved in CUSTOMER.TV.

To restore table properties

See also

If you change table properties, then change your mind about them, you can restore previous settings.

To restore table properties,

- Choose Table|Table View Properties|Restore.

  Database Desktop restores all properties to the settings they had when you opened (or previously saved) the table properties.

If you try to close a Table window without saving property changes, Database Desktop asks if you want to save your changes.

To delete table properties

See also

When you delete a table's unique property file (.TV or .TVF), Database Desktop uses default property settings.

To delete table properties,

- Choose Table|Table View Properties|Delete.

▪

About sorting tables

See also

When you sort a table, Database Desktop rearranges the order of the records in the table and displays them in the order you specify. To change the actual location of records in the table, you can sort the keyed table to a new table.

Database Desktop cannot sort on the following field types:

- BLOB, BCD, logical, or bytes fields in Paradox tables
- Memo, binary, OLE, or logical fields in dBASE tables

Fields of these types are displayed in the Fields list, but are dimmed and cannot be selected for placement in the Sort Order list.

Note: You cannot sort SQL tables.

Sorting keyed tables

See also

If a table is keyed, Database Desktop keeps the records sorted according to the values in the key field (or fields).

You cannot override the sort order established by a table's key. What you can do, however, is use a maintained secondary index to change the view order of the keyed table. This gives you a sorted view of the records, but doesn't change the physical location of the records in the keyed table.

If you want to change the actual location of the records in a keyed table, you can sort the keyed table to a new table. The new table created by the sort operation is unkeyed. The original table remains unchanged.

Sorting unkeyed tables

See also

If a table is not keyed, records appear in the table in the order in which you entered them. (See About indexes and keys for information on creating keys.)

When you sort an unkeyed table, you change the actual location of the records in the table. You tell Database Desktop the fields on which you want the table sorted. Database Desktop then rearranges the records based on field values. You can sort an unkeyed table to itself, or create a new sorted table, leaving the original intact.

For dBASE tables, the default order is chronological; for Paradox tables, it is positional.

▪

Sorting on a network

See also

When you sort tables in a multiuser environment, Database Desktop automatically places a lock on the table you are sorting. This means other users cannot modify its contents or structure. If another user has a lock on the table, you will not be able to begin sorting until that user finishes working with it.

When you sort to a new table, Database Desktop automatically places a lock on that table as well as the original table for the duration of the sort.

To sort a table

See also

1. Choose Tools|Utilities|Sort, then choose the table you want to sort from the Select File dialog box.

   Database Desktop displays the Sort Table dialog box.
2. Specify the order to sort the records of the table in by moving fields from the Fields list to the Sort Order list. See To add fields to the Sort Order list.
3. Specify whether to sort the fields in ascending or descending order by selecting a field in the Sort Order list and choosing Sort Direction.
4. If the table is keyed, enter a file name for the new table in the New Table text box.
5. Specify whether you want to Sort Just Selected Fields.
6. Specify whether you want to Display The Sorted Table.
7. Choose OK.

To sort an Answer table

See also

You can sort Answer tables from queries.

1. Make the Query window active.
2. Choose Query|Properties, then choose the Sort tab.

   Use the arrows to move Answer table fields from the Answer Fields list to the Sort Order list in the order you want them to sort.
3. Choose OK.

Now, when you run the query, the Answer table is sorted in the order you specify.

To specify the sort order

See also

You can specify fields to sort on and the order to sort them in. You can also specify whether to sort in ascending (aa-zz) or descending (zz-aa) order.

To specify the sort order,

1. Choose Tools|Utilities|Sort, then choose the table you want to sort in the Select File dialog box.

   Database Desktop displays the Sort Table dialog box.
2. Select the fields to sort by in the Fields list and add them to the Sort Order list as described in To add fields to the Sort Order list.
3. Specify whether to sort the fields in ascending or descending order as described in To specify ascending or descending sort order.
4. If the table is keyed, type a new table name in the New Table text box.
5. Specify whether you want to Sort Just Selected Fields.
6. Specify whether you want to Display The Sorted Table.
7. Choose OK.

When Database Desktop performs the sort, it sorts records on the values in the first field in the Sort Order list, then on the values in the second field, and so on.

You do not have to put all the fields from the Fields list in the Sort Order list. Database Desktop adds any fields you do not explicitly put in the Sort Order list to the end of that list before performing the sort (unless you have checked Sort Just Selected Fields). In any case, Database Desktop includes all fields in the result (whether the result is the same or a new table).

Note: If you do not add any fields to the Sort Order list, Database Desktop sorts the table in the order of the fields in the Fields list. If you check Sort Just Selected Fields, you must place at least one field in the Sort Order list.

To add fields to the Sort Order list

See also

To add fields to the Sort Order list,

1. Choose Tools|Utilities|Sort, then choose the table you want to sort in the Select File dialog box.

   Database Desktop displays the Sort Table dialog box.

2. Select one or more fields in the Fields list. To select multiple fields from the Fields list, do one of the following:

- Click a field at one end of the range and drag to the other end of the range.
- Using the keyboard, move to the top field in the range, hold Shift and press the down arrow ↓ key until all the fields you want are selected.
- Hold Shift, then click the fields at the beginning and end of the range.

3. Choose the Add Field arrow ▪, press Alt+A, or double-click the field. The field appears in the Sort Order list. The field name remains in the Fields list, but it is dimmed to show that it is no longer available.

   If you select a range of fields that extends over fields that cannot be sorted on, or over fields already added to the Sort Order list, Database Desktop ignores them.

Database Desktop cannot sort on BLOB, BCD, logical, or bytes fields in Paradox tables, or on memo, binary, OLE, or logical fields in dBASE tables. Database Desktop displays these types of fields in the Fields list, but they are dimmed and cannot be placed in the Sort Order list.

To add fields to the top of the Sort Order list

See also

To insert fields at the top of the Sort Order list,

1. Choose Tools|Utilities|Sort, then choose the table you want to sort in the Select File dialog box.
   Database Desktop displays the Sort Table dialog box.
2. Select the top field in the Sort Order list.
3. Select a field in the Fields list and add it to the Sort Order list. The new field appears selected below the top field.
4. Use the Change Order up arrow to move the field to the top position.

To remove fields from the Sort Order list

See also

Removing selected fields

1. Choose Tools|Utilities|Sort, then choose the table you want to sort in the Select File dialog box.

   Database Desktop displays the Sort Table dialog box.

   Suppose the table was sorted previously and you want to remove some of the sort fields listed in the Sort Order list.
2. Select the fields to remove in the Sort Order list.

   To remove a range of fields, select the range by dragging.
3. Choose the Remove Field arrow ▪ or press Alt+R. The field moves to the Fields list.

Removing all fields

To remove all fields from the Sort Order list, making those fields available again in the Fields list,

1. Display the Sort Table dialog box as described in Step 1 above.
2. Choose Clear All or press Alt+C.

To rearrange the sort order

See also

To rearrange fields in the Sort Order list,

1. Choose Tools|Utilities|Sort, then choose the table you want to sort in the Select File dialog box.

   Database Desktop displays the Sort Table dialog box.
2. Select the fields to sort by in the Fields list and add them to the Sort Order list as described in To add fields to the Sort Order list.
3. Select the field to reorder in the Sort Order list.
4. Click the up or down Change Order arrow below the Sort Order list.

   The Change Order arrows are available only when the Sort Order list contains two or more fields.

To specify ascending or descending sort order

See also

Each field in the Sort Order list is preceded by a sort order indicator that shows whether the sort order within the field is ascending (**+**) or descending (**-**). The default is ascending (aa-zz).

1. Choose Tools|Utilities|Sort, then choose the table you want to sort in the Select File dialog box.

   Database Desktop displays the Sort Table dialog box.
2. Select the fields to sort by in the Fields list and add them to the Sort Order list as described in To add fields to the Sort Order list.
3. To reverse the sort order for a field, double-click the sort order indicator or select the field and choose Sort Direction. The sort order indicator and Sort Direction are toggles that change from ascending to descending sort order and back again with each successive click.

▪

About table structure information

See also

You can use the Info Structure command on the Table or Tools|Utilities menus and Structure Information dialog box to get information about a table. The Structure Information dialog box shows the structure of the table, as well as any key, validity check, index, table lookup, or referential integrity information.

You cannot make changes to the table structure from the Structure Information dialog box; you can only view the structure. You can choose Tools|Utilities|Restructure or Table|Restructure to restructure a table.

Paradox tables

The Table Properties drop-down list displays the following information about a Paradox table:

- Validity Checks shows validity checks defined for each field. Move through the fields in the Field Roster to see validity checks for each one.
- Table Lookup shows any tables that this table uses as a lookup table.
- Secondary Indexes shows all secondary indexes for the table.
- Referential Integrity shows whether this table refers to a parent table for valid data.
- Table Language shows the language driver for the table.
- Dependent Tables shows any table that this table recognizes as a child in a referential integrity relationship.

dBASE tables

The Table Properties drop-down list displays the following information about a dBASE table:

- Indexes shows all indexes for the table.
  You can select an index and choose Detail Info to see information about the index in the Index Info dialog box (dBASE tables) dialog box.
- Table Language shows the language driver for the table.

SQL tables

- The Required Field check box (in the panel on the right) specifies whether the selected field is required.
- The panel on the right lists indexes for the table. You can select an index and choose Detail Info to see information about the index in the Index Info dialog box (SQL tables) dialog box.

To get table structure information

See also

To view the structure of a table,

1. Do either of the following:

- Right-click the table in the Project Viewer and choose Info Structure.
- Choose Tools|Utilities|Info Structure from the Desktop, then choose the table in the Select File dialog box.

   The Structure Information dialog box appears.

2. Review the structure information. Use the Table Properties drop-down list to review information about properties such as validity checks, secondary indexes, and so on.

   For dBASE and SQL tables, you can select an index from the list and choose Detail Info to see information about the index.Database Desktop opens the Index Info dialog box dialog box for dBASE or SQL tables.

3. To save the structure information as a Paradox table, choose Save As.

   The Save Structure Information As dialog box appears.

4. Specify a name and path for the table and choose Save.
5. Choose Done to close the Structure Information dialog box when you have finished viewing the table structure.

▪

About adding records

See also

To quickly add many records to a table, you can merge the records from another table that has the same structure.

Use the Add command to add a copy of the records in one table to another table.

You can use the Options area in the Add Records In <table> To dialog box to either add new records, update existing records, or both.

For details on source and destination table requirements, see Rules for adding records.

Adding records to a different table type

When you add records from one table type to another, consider whether the field types in the table you add records to are compatible with the field types in the table you add records from. The rules for adding records from one type to another are the same as those for restructuring from one table type to another.

- For information about adding between Paradox and dBASE tables, see Adding compatible field types and Merging Paradox and dBASE tables.
- For information about adding between local tables and SQL tables, see your SQL Links documentation.

Note: Some field type conversions can result in invalid records being written to the temporary Problems table. If this happens, edit the records in the Problems table and then add them again. The Problems table is not generated for SQL tables; the invalid records are dropped.

Rules for adding records

See also

When performing an Add operation, keep these rules in mind:

- You can add records from one table type to another only if the tables have a compatible structure. This means compatible field types in the same order. For more information, see Adding compatible field types. Some field type conversions can result in invalid records being written to the temporary Problems table. If this happens, edit the records in the Problems table and then add them again.
- The table you add records to can have more fields than the source table, as long as the first fields of the table you add the records to are compatible with all fields of the source (compatible fields types in the same order). Database Desktop places null values in the extra fields.
- The source table can have more fields than the table you add the records to, as long as the fields of the table you add the records to are compatible with the first fields of the source (compatible field types in the same order). Database Desktop ignores the extra fields.
- If the table you add the records to is keyed, the added records must conform to the rules of the key. Database Desktop places records that do not conform in the temporary Keyviol table in your private directory. The source table is never changed during an Add operation; it does not matter if it is keyed or not.

▪

Adding compatible field types

See also

The two tables you use in the Add operation must have compatible (though not necessarily identical) field types in the same order.

For fields to be compatible, Database Desktop must be able to change from the existing field type to the new field type in a Restructure operation. For example, Paradox number (N) and money ($) fields are compatible, but Paradox number (N) and graphic fields (G) are not.

- For information about compatible Paradox and dBASE field types, see
  Compatible Paradox Field Types
  Compatible dBASE Field Types
  Merging Paradox and dBASE tables
- For information about compatible field types for SQL tables, see your SQL Links documentation.

Merging Paradox and dBASE tables

See also

In an Add operation, the rules for adding records from one type to another are the same as those for restructuring from one table type to another.

You can add records from one table type to another only if the tables have a compatible structure. This means compatible field types in the same order. The following table shows which Paradox and dBASE field types are compatible.

	dBASE C	dBASE F	dBASE N	dBASE D	dBASE L	dBASE M	dBASE O	dBASE B
Paradox								
A	Yes	P	P	P	P	Yes	No	No
N	Yes	Yes	Yes	No	P	No	No	No
$	Yes	Yes	Yes	No	No	No	No	No
D	Yes	No	No	Yes	No	No	No	No
S	Yes	Yes	Yes	No	P	No	No	No
M	No	No	No	No	No	Yes	No	Yes
F	No	No	No	No	No	Yes	No	Yes
B	No	No	No	No	No	Yes	No	Yes
G	No	No	No	No	No	Yes	No	Yes
O	No	No	No	No	No	Yes*	Yes	Yes
I	Yes	Yes	Yes	No	Yes	No	No	No
#	Yes	Yes	Yes	No	Yes	No	No	No
T	Yes	No	No	Yes	No	No	No	No
@	Yes	No	No	Yes	No	No	No	No
L	Yes	Yes	Yes	No	Yes	No	No	No
+	Yes	Yes	Yes	No	Yes	No	No	No
Y	P	No	No	No	No	Yes	No	No

Yes	The field types are compatible.
No	The field types are not compatible.
P	The field types are somewhat compatible, but conversion can result in a Problems table.

* You can add from a Paradox OLE field to a dBASE IV memo field, but not to a dBASE memo field.

When you add data from a Paradox formatted memo to a dBASE memo, Database Desktop removes all formatting and converts the data to straight text.

When you add data from a Paradox graphic, OLE, or binary field to a dBASE memo, the dBASE table can accept the data, but cannot display it.

The table you add the records to can have more fields than the source table, as long as the first fields of the table you add the records to are compatible with all fields of the source (compatible field types in the same order). Database Desktop places null values in the extra fields.

Merging tables on a network

See also

When you merge tables using Add, Database Desktop needs to acquire a read lock on the source table and a write lock on the table you add records to. This means that until the records are added, other users

- Cannot change the contents or structure of either table
- Cannot perform any operation that requires a write or exclusive lock on the target table

If another user has already placed a write or exclusive lock on either table, you must wait until the lock is removed before using Add.

Windows lets you open several instances of the same table at the same time, so you could be considered another user of the table, preventing the records from being added. You can add records to an open table only if you are viewing the table; you cannot add records to a table that is open in Edit mode.

To add records from another table

See also

To quickly add many records to a table, you can merge the records from another table that has the same structure. The two tables can be of different types, as long as their fields are compatible. To verify that the source and destination tables are suitable for this procedure, see Rules for adding records.

To add records from another table,

1. Choose Tools|Utilities|Add. Database Desktop opens the Add Records In dialog box.

   All tables in the working and private directories are shown in the list below the Look In list box.

   You can use the Look In or Alias drop-down lists to access files in different directories. You can perform an Add operation across directories. Choose an alias for an SQL server to display a list of tables for that server.
2. Select the table you want to add records from, then choose Open.

   Database Desktop opens the Add Records In <table> To dialog box.
3. Choose the table you want to add records to.
4. If you want to add new records or update existing records (or both), specify append and update options.
5. Choose Add to add the records.

▪

About moving records

See also

In certain situations, you may have a record in one table that corresponds to a record in another table. This can happen:

- In a referential integrity relationship, where one record in a parent table is related to one or more records in a child table
- In a multi-table form, where one record of the master table is related to one or more records in the detail table

In either of these kinds of relationships, you can use Move Help to move, or reassign, a dependent record from one master to a different master.

Example:

For example, suppose you've linked Customer and Orders in a one-to-many relationship in a form. If you select a value in Customer No in the Orders table, then choose Record|Move Help (or press Ctrl+Shift+Spacebar), you'll see the Customer table in a dialog box. When you choose a value from the Customer No field in this lookup table, Database Desktop changes the Customer No value for the selected record, assigning it to a different master.

To move dependent records with Move Help

See also

In certain situations, you may have a record in one table that corresponds to a record in another table. You can use Move Help to move, or reassign, a dependent record from one master to a different master.

To move dependent records,

1. Open the dependent table.
2. Press F9 to enter Edit mode.
3. Select the record to move, or reassign, by clicking in the field of that record that corresponds to the first field of the master table in a referential integrity relationship. You can click in any field of a detail table.
4. Choose Record|Move Help or press Shift+Ctrl+Spacebar to display the Move Help dialog box.
5. Select the master record to receive the reassigned dependent record.
6. Click OK to complete the move.

▪

About subtracting records that exist in another table

See also

You can use the Subtract command to remove from one table those records that match records in another (called the subtraction table). For example, after a mass mailing, you might want to create a table of all customers who did not answer their letters. You could then subtract the records in this table from your Customer table.

You can subtract records only from a keyed table. Because dBASE and SQL tables do not support Paradox keys, you cannot subtract records from dBASE or SQL tables. Instead, use a DELETE query.

SQL: You cannot use an SQL table as the source of a subtract operation.

During a subtract operation, Database Desktop removes any record that contains a value in its key that exactly matches the corresponding field(s) of a record in the subtraction table.

Rules for subtracting tables

- The two tables you use in the Subtract operation must have compatible structures. This means compatible fields in the same field order.
- If the table you subtract from is the parent table in a referential integrity relationship, the Subtract operation is not allowed. You must first either delete the referential integrity (by restructuring the child table) or delete the child table.

▪

Subtracting records on a network

See also

When you use Subtract, Database Desktop needs to acquire a read lock on the table that contains the records you are subtracting and a write lock on the table you are subtracting records from. This means that until the records are subtracted, other users cannot

- Change the contents or structure of either table
- Perform any operation that requires a write or exclusive lock on either table

If another user has already placed a write or exclusive lock on either table, you must wait until the lock is removed before using Subtract.

Windows lets you open several instances of the same table at the same time, so you could be considered another user of the table, preventing the records from being subtracted. You can subtract records from an open table only if you are viewing the table; you cannot subtract records from a table that is open in Edit mode.

To subtract records that exist in another table

See also

The Subtract command lets you subtract records from one table that match records in another. To verify that the tables meet Subtract command requirements, check the rules listed in About subtracting records that exist in another table.

To subtract one table from another,

1. Choose Tools|Utilities|Subtract. Database Desktop opens the Subtract Records In dialog box.

   All tables in the working and private directories are shown in the list below the Look In list box.

   You can use the Look In or Alias drop-down lists to access files in different directories. You can perform an Add operation across directories. Choose an alias for an SQL server to display a list of tables for that server.
2. Select the table with records you want to subtract, then choose Open.

   Database Desktop opens the Subtract Records In <table> From dialog box.
3. Choose the table you want to subtract records from.
4. Choose Subtract to subtract the records.

▪

About emptying tables

See also

You can use the Empty command to remove all records from a table, leaving the table's structure (including all keys, indexes, validity checks, and so on) intact.

You can use Empty on Paradox, dBASE, and SQL tables. When you empty a dBASE table, all records in the table are marked as deleted.

You cannot empty a table that is identified as the parent in a referential integrity relationship. You must first either delete the referential integrity (from the child table) or delete the child table.

Emptying tables on a network

When you use Empty, Database Desktop must acquire an exclusive lock on the table. This means

- No user can access the table in any way.
- If there is a lock of any type open on the table, you must wait until it is released before you can use the Empty utility.

To empty tables

See also

To empty a table,

- Choose Tools|Utilities|Empty. Database Desktop opens the Empty dialog box.

  All tables in your working directories are shown in the file list. You can use the Look In and Alias drop-down lists to locate files in different directories. Choose an alias for an SQL server to display a list of tables for that server.

  Enter the name of the table you want to empty in the Look In drop-down list. When you choose Empty, Database Desktop displays a message asking you to confirm the Empty operation. Choose Yes to remove all records from the table or No to cancel the operation.

Note: You cannot empty a table that is identified as the parent in a referential integrity relationship. You must first either delete the referential integrity (from the child table) or delete the child table.

About entering and editing data

See also

You can enter data into tables by typing in Edit mode. For details on Edit mode, see About Edit mode.

In addition to typing values in fields, you can cut or copy data from a field and paste into different fields. Data you cut or copy to the Clipboard remains there until you change it, clear it, or exit Windows. The Clipboard provides temporary storage for data you want to move to a different location.

You can also insert, edit, and delete data records with INSERT, CHANGETO, and DELETE queries.

Edit commands

With a table open in Edit mode, you can use the following commands on the Edit menu to work with your data.

Choose:	To:
Undo	Undo all changes to the current record. This does not undo any changes you posted. You must choose Undo before leaving the record.
Cut	Delete a value from a selected field or fields in a table and place it on the Windows Clipboard.
Copy	Copy a value from a selected field or fields in a table and place it on the Windows Clipboard. In a Table window, you can copy more than one field at a time. When you make your selection, lines appear around the selected data.
Paste	Paste the contents of the Windows Clipboard into the selected field. **Note:** You can paste only a valid value into a field. For example, you cannot paste a graphic value into an alpha field.
Paste Link	Establish a link using Dynamic Data Exchange (DDE) from another Windows application to your table.
Delete	Remove the value. Database Desktop does not place it on the Windows Clipboard.
	Note: You can remove an entire record with Edit|Delete but not with Edit|Cut.
Select All	Select all fields in the table (the entire table). Database Desktop places a box around the table.

▪

Guidelines for entering and editing data

See also

To enter or change data, follow these guidelines:

- Open a table with File|New, File|Open, or the Open Table Toolbar button, then follow the steps in this topic:

  To enter or edit data in a Table window

- Make your edits in Edit mode and Field View or Persistent Field View. For details, see

  About Edit mode

  About Field View

- Use Edit|Undo to undo changes that you make to data.
- Database Desktop automatically saves the data you enter as soon as you leave a record. So, you do not use the Save or Save As commands to save table data.
- You use Tools|Utilities|Rename to rename a table.
- You save changes to table properties by choosing Table|Table View Properties|Save. (If you make changes to table properties and do not save them, Database Desktop prompts you to save them when you close the table.)

▪

About Edit mode

See also

Before you can edit data in a table, you must enter Edit mode. This tells Database Desktop that you don't just want to look at the data, you want to change it.

The look of the table changes slightly when you enter Edit mode.

Once you're in Edit mode, you can move the insertion point to any of the table's fields and begin typing. (This replaces the existing contents of the field.) In most field types, you simply select the field you want and type a value in it.

If you need to position the insertion point at some particular point within the field (for example, to change a spelling or typing error), enter Field View. For details, see About Field View.

Entering Edit mode

Use any of these ways to enter Edit mode:

- Choose Table|Edit Data.
- Click the Edit Data Toolbar button.
- Press F9.

Exiting Edit mode

Use any of these ways to exit Edit mode:

- Choose Table|View Data.
- Click the Edit Data Toolbar button.
- Press F9.

▪

About Field View

See also

In normal Edit mode, whatever you type in a field overwrites the data entered there. To change only part of a field, use Field View.

Entering Field View

To enter Field View, select the field, then either

- Click the Field View Toolbar button
- Choose View|Field View
- Press F2
- Click the selected field again

In Field View, you can use the Left and Right arrows, as well as Backspace and Del.

Exiting Field View

To exit Field View, either

- Click the Field View Toolbar button again
- Choose View|Field View
- Press F2
- Select a different field

Also, pressing Enter, Tab, or Alt with the arrow keys lets you exit Field View and move to a different field.

Tip: If you want to move from field to field and remain in Field View, press Ctrl+F2 to enter Persistent Field View. Press Ctrl+F2 again to exit Persistent Field View. For more information, see About Persistent Field View.

About Persistent Field View

See also

In Edit mode and Field View, you can edit part of a field without overwriting the rest of the data in the field. But, when you leave a field, you exit Field View.

You can use Ctrl+F2 to enter into Persistent Field View, where you can move from field to field without leaving Field View.

In Persistent Field View, press Tab, Enter, or Alt plus an arrow key to move from field to field. Press arrow keys to move character-by-character within a field.

Press Ctrl+F2 again to leave Persistent Field View.

▪

Data entry shortcuts

See also

Use these keyboard shortcuts for faster data entry. You can also use the navigation buttons on the Toolbar. For details, see Table navigation buttons.

Press	To
Home	Move to the first field of the table, remaining on the selected record.
Ctrl+Home	Move to the first field of the first record of the table.
End	Move to the last field of the table, remaining on the selected record.
Ctrl+End	Move to the last field of the last record of the table.
Ctrl+Backspace	Delete the word to the left of the insertion point. **Note:** Ctrl+Backspace works only when you are in Field View and do not have text selected.
Ctrl+D	Duplicate the information from the record above the selected field to the selected field.
Esc	Undo a field edit (you must press Esc **before** you leave the field!).
Spacebar	Enter current date, time, or both in date, time, or timestamp fields. You must press the spacebar for each part of the field's format.

These topics give additional keyboard shortcuts for entering and editing data:

Navigation and selection keys

Keys used in Edit mode

Why can't I leave a field?

See also

The status line at the bottom of the Desktop tells you what the problem is. If you cannot see a status line, maximize the Database Desktop window.

Several things can prevent you from leaving a field:

- The field requires that a value be entered, and you have not entered one (for example, maximum or minimum values have been specified, or a picture string has been specified). Type any character or number that satisfies any validity checks defined for that field to get out of this field.
- The field requires specific values from a lookup table, and you have not provided an acceptable one. To get out of such a field:
- Press Ctrl+Spacebar to see the lookup table and choose a value from that.
  If the lookup table was defined with Help And Fill checked, it appears for you to choose from.
- If no lookup table appears, press Esc to undo your entry. Find out what the acceptable values are before you continue.
- The value you entered violates referential integrity requirements. Press Esc to undo your entry. Find out what the acceptable values are before you continue.
- The value you entered violates the table's key. Choose Edit|Undo to remove the current record.

To enter or edit data in a Table window

See also

To enter or edit data in a Table window,

1. View the table, then either

- Click the Edit Data Toolbar button

- Choose Table|Edit Data
- Press F9

2. Place the insertion point in the field you want to edit. Whatever you type replaces what is in the field.

   You can edit just a portion of the field by using Field View.

In addition to the usual Edit menu commands, you can press Ctrl+D in any field to copy a field value from the record above it.

To insert today's date in a date field, press Spacebar three times. Database Desktop adds the three elements of a date separately.

In Edit mode, your data is saved automatically every time you move off a record.

To exit Edit mode, choose Table|View Data, press F9, or click the Edit Data Toolbar button.

▪

About inserting, posting, and deleting records

See also

You can insert new blank records or delete existing records from a table.

Inserting and posting records

To insert a blank record above the selected record, follow the steps in To insert records. Database Desktop saves the new information as soon as you move off the record or choose Record|Post/Keep Locked.

Saving a record is often called posting or committing a record. When working in a multiuser environment, other users do not see changes you've made until you've posted them.

- When you post a record in a keyed table, Database Desktop automatically moves it to its proper position in the table. If the record's proper position is off screen, the record may seem to disappear as it is posted. However, if you look at the record count on the status bar, you'll see that the record has been added. Your view of the table might not change when Database Desktop posts the record, but the insertion point remains where it was when you pressed Ins.
- When you post a record in a non-keyed table, the record always stays where it is inserted.

If you insert a record into a live query view, and the record does not meet the criteria established by the query, you won't see the record when it is posted.

Deleting records

To delete records, follow the steps in To delete records.

When using a Paradox table, you cannot retrieve a deleted record. When using a dBASE table, deleting a record does not permanently remove it; to do that, you must restructure the table with Pack Table checked.

To insert records

See also

To insert records,

1. Open a table and click the Edit Data Toolbar button to enter Edit mode.
2. Press Ins.

   Database Desktop opens a new blank record above the insertion point position.
3. Enter data into the fields of the new record.

   Type the values you want in the new record's fields, then either move off the record or choose Record|Post/Keep Locked to save the new record in the table.
4. Click the Edit Data Toolbar button again to exit Edit mode, or navigate to a different record.

- If the table is keyed, Database Desktop automatically moves the record to its correct location in the table.
- If the table is not keyed, the new record stays in the location where you added it.

Database Desktop saves the changes you make to a record when you exit Edit mode or move to a different record. This is called posting the record. For more information on posting records, see About inserting, posting, and deleting records.

To add records from another table, choose Tools|Utilities|Add from the Desktop. The tables must have compatible structures. For details, see About adding records.

To delete records

See also

To delete records,

1. Open a table and click the Edit Data Toolbar button to enter Edit mode.
2. Navigate to the record that you want to delete.
3. Press Ctrl+Del to delete the selected record. Database Desktop deletes the record from the table.
4. Click the Edit Data Toolbar button again to exit Edit mode.

When using a Paradox table, you cannot retrieve a deleted record. When using a dBASE table, deleting a record does not permanently remove it; to do that, restructure the table with Pack Table checked in the Restructure dBASE Table dialog box.

▪

About cutting, copying, and pasting data

See also

In addition to typing values in fields, you can cut or copy data from one field and paste it into a different field or a different application. Data you cut or copy remains on the Windows Clipboard until you change it, clear it, or exit Windows. The Clipboard provides temporary storage for data you want to move to a different location.

For information on entering and deleting data with the Clipboard, see About using the Clipboard.

Database Desktop also has commands for copying entire files (tables, queries, text, and SQL files). For information on these commands, see Tools|Utilities|Copy.

▪

About using the Clipboard

See also

You can use the Windows Clipboard to help you enter, copy, and move Database Desktop data:

- Choose Edit|Cut to delete the selected field's value from the table and place it on the Clipboard.
- Choose Edit|Copy to copy the selected field's value from the table to the Clipboard.

  When using a table, you can copy the values of more than one field at a time. For instructions, see To copy data.

When you cut or copy an object using Edit|Cut or Edit|Copy, Database Desktop stores the object on the Clipboard. You can paste these stored objects from the Clipboard back into Database Desktop designs:

- Choose Edit|Paste to paste the contents of the Clipboard into the selected field. (You can paste into only one field at a time.)

  You can paste only a valid value into a field. For example, you can't paste a graphic value into an alphanumeric field.

The Clipboard stores only one image at a time. Each time you place an object on the Clipboard, Database Desktop discards the previous image.

You can delete objects without storing them on the Clipboard. For details, see To delete data.

Undoing Clipboard actions

Edit|Undo does not work for Clipboard actions, like cut, copy, and paste. To undo a paste, delete the pasted object. To undo a cut, paste the object back in. You cannot undo a copy.

To copy data

See also

To copy data to the Clipboard, select the data you want and choose Edit|Copy. You can paste data you have copied to the Clipboard into other fields or other Windows applications.

To copy from a field

Select a field, then choose Edit|Copy to copy the entire field value. To copy only a portion of a field's data, enter Field View and select the data you want. Then choose Edit|Copy. When using a Table window, you can copy more than one field at a time. When you make your selection, lines appear around the selected data. Database Desktop must be in Edit mode.

To copy from a column

Double-click the column heading to select the column, then choose Edit|Copy.

To copy from a row

Double-click an unselected record number. (if the record number is selected when you double-click, you enter Field View). Then, choose Edit|Copy.

To copy multiple field values

Either choose Edit|Select All followed by Edit|Copy (this copies all the values in the table to the Clipboard), or drag over the specific fields you want to select and choose Edit|Copy.

You can copy multiple field values only in a table. You cannot paste multiple field values back into a table. You can, however, paste them into any other application which accepts them (for example, Quattro Pro for Windows).

To delete data

See also

To delete data and store it on the Clipboard,

- Select the data to delete, then choose Edit|Cut or the Cut Toolbar button.

To delete data without storing it on the Clipboard,

- Select the data to delete, then choose Edit|Delete or press Del.

In this case, the object is not saved and cannot be pasted from the Clipboard. To retrieve it, choose Edit|Undo immediately.

To paste data

See also

To paste data from the Clipboard,

- In Edit mode, select the field, text, or object to paste into, then choose Edit|Paste or the Paste Toolbar button.

The contents of the Clipboard are not deleted when you paste, so you can paste as many times as you want.

Use Edit|Paste Link to create DDE links.

▪

Wildcards

See also

You can use two wildcards in any search string you specify using Search |Find and Search|Replace.

Wildcard	Represents
@	Any single character
..	Any value

To search for these characters as literals, you must precede them with a backslash (\ ).

For examples, see Sample search strings with wildcards.

Extended list of wildcards

See also

You can use an extended set of wildcards in a search string when you check Advanced Pattern Match in the Search|Find and Search|Replace dialog boxes.

Wildcard	Represents
@	Any single character
..	Any value
^	Beginning of field
$	End of field
*	Match none or more of the expression before the *
+	Match one or more of the expression before the +
?	Match one or none of the expression before the ?
|	Match either the characters before or after the vertical bar
[abc]	Match any of the characters contained within the brackets
[^abc]	Match any characters not contained within the brackets
(abc)	A group (a series of literals)
\	Use the following wildcard operator as a regular character
\r	Carriage return
\n	Line feed
\t	Tab
\f	Form feed

For examples, see Sample search strings with wildcards.

Sample search strings with wildcards

See also

Here are some examples of wildcard characters in a search string and what they find when you choose Advanced Pattern Match in the Find or Find And Replace dialog box.

Search string	Finds
co@l	cool and coal, but not col
s**..**ch	search, scorch, and such
^any	any only when it occurs at the start of a paragraph (when Case-sensitive is not checked)
able$	able only when it occurs at the end of a paragraph (and is not followed by a period)
(success)	success
[success]	Any s, u, c, or e
[^success]	Any character except s, u, c, or e
a | (an)	Either a or an ("an" is a group here)
hands?	hand and hands (hand with or without the s)
suc?es?	success or Sue (when Case Sensitive is not checked). The ? stands for one "c" or none and one "s" or none.
suc*es*	success or Sue (when Case Sensitive is not checked). The ***** stands for any number of c's or none at all, and any number of s's or none at all.
suc+es+	success only; the + stands for one or more c's and one or more s's
4\^2	4^2 (read "four squared"). Without the backslash, only paragraphs ending in 4 followed by a paragraph starting with 2 would be found.
apples\\pears	apples\pears
apples\\\\pears	apples\\pears

Note: You can use **?**, *****, or **+** if you are not sure how to spell success.

About editing fields with validity checks

See also

Validity checks impose restrictions on a field to ensure that the data entered in the field meets certain requirements. For example, you can define a maximum value validity check for a field so Database Desktop doesn't accept any value higher than the maximum.

When a validity check is defined, you can't post or leave a record until its requirements are met. If you enter invalid data, Database Desktop prevents you from moving off the record. Either correct the data or undo changes to the record before you move.

The following table describes how validity checks determine the kind of data you can enter into a field. For information on defining validity checks, see About validity checks.

Type of validity check	Description
Required field	You can't move from the record until you enter a value. This ensures that important fields always have data in them for each record.
Minimum value	Database Desktop won't accept any value less than the minimum value.
Maximum value	Database Desktop won't accept any value greater than the maximum value.
Default value	Database Desktop automatically enters the default value in the field when you insert a new record.
	To enter a value different from the default, select the field and enter the value you want. To enter a blank value, select the field and press Backspace or Del.
	Database Desktop inserts default values only in new records. Moving through an existing record will not cause a default value to be inserted
Picture	Pictures are patterns Database Desktop uses to validate and help you enter correctly the data you place in a field.
	For example, a common picture is (###)###-####. This is the pattern of most U.S. telephone numbers. If you have defined this picture for a field, you can just enter numbers, without the parentheses or hyphen. Database Desktop enters the numbers correctly (according to the picture) and adds the parentheses and hyphen to the value.
	Picture validity checks provide an editing aid (automatically checking your data for you) as well as enforce rules that ensure the data you enter meets with the requirements you established for valid data in the field when you created the table. See About pictures for information on the different types of pictures you can create.

▪

About locking records

See also

Database Desktop automatically locks a record when you start editing it and removes the lock when you leave the record. A message appears in the status bar to inform you of these automatic locks.

You can also manually lock a record. Select the record, then choose Record|Lock (or press F5 or Ctrl+L). The status bar tells you that the record is locked.

Locking is important if you use Database Desktop in a multiuser environment. When you lock a record, other users can view it, but can't edit or delete it.

Locking a record prevents other users from placing a read or write lock on the table. It also prevents users from performing any operations that require a read or exclusive lock (such as restructuring the table).

To lock a record

See also

- Choose Record|Lock or press F5 to place a lock on the selected record.
  The Desktop status bar tells you when you have locked a record.

After you lock a record, the Lock command changes to Unlock. You must unlock the record before another user can change it.

On a network

Locking is important if you are using Database Desktop in a multiuser environment. When a record is locked, other users can view it, but cannot edit or delete it. If you try to change a record locked by another user, Database Desktop tells you the record is locked by another user. A record is automatically locked for you when you begin to edit it. A record is automatically unlocked when you move to another record.

On a single computer

Traditionally, the term multiuser has been equivalent to the term network. This is true in Database Desktop too, but you can also place yourself in a multiuser situation working on a standalone system.

For example, if you open a table, Database Desktop places a lock on it. This ensures you an accurate view of the table; it cannot be restructured or deleted while you are using it. This is true when you open the table in a query.

Sometimes these automatic locks prevent you from performing an operation on a table. For example, you are prevented from deleting an open table.

In these circumstances, your various windows of table data act as various users of the table.

To unlock a record

See also

- Database Desktop automatically unlocks a record when you move off it or exit Edit mode.

After you've locked a record, the Lock command changes to the Unlock command. Choose Unlock if you want to release a record for other users' access without moving off it. (You must unlock records before other users can edit or delete them.)

To post a record without unlocking it

See also

Database Desktop automatically saves (posts) any changes you make when you leave the record, but if you want to save your edits before you leave the record,

- Choose Record|Post/Keep Locked.

Sometimes Database Desktop moves a record to a different location when you post it. This happens if the table is keyed and the new record is not in its correct location in the table. Database Desktop moves the record to its correct location. When you choose Record|Post/Keep Locked, the moved record remains selected, and Database Desktop updates your view of the table if necessary.

-

Types of table lookup

See also

Database Desktop provides two types of table lookup:

- Just Current Field: The value in the current field is the only value from the lookup table that Database Desktop checks or fills in for you.
- All Corresponding Fields: Database Desktop checks the field on which the table lookup is defined and fills values in all fields that match fields in the lookup table. (Database Desktop determines if fields match by the field names.)

Whether you'll be able to view the lookup table from the table you're editing depends on the type of lookup access specified when the table lookup was defined:

- Help And Fill: You can view the lookup table from the table you're editing; the default.
- Fill No Help: You can't view the lookup table from the table you're editing.

When the lookup access is Fill No Help, you can't open the lookup table automatically. You can, however, view the lookup table by opening it in its own Table window.

For information on defining table lookup, see About table lookups.

For examples, see Examples of table lookup.

To use table lookup

See also

Table lookup lets you refer to another table to look up the acceptable values for a field and then automatically copy values in the lookup table to the table you are editing. Before table lookup can be used, you must define one using the Table Lookup dialog box. For information on defining table lookup, see About table lookups.

To use table lookup,

1. Press Ctrl+Spacebar to view the lookup table. For this to work, you must have defined the lookup table with Help And Fill selected in the Table Lookup dialog box.

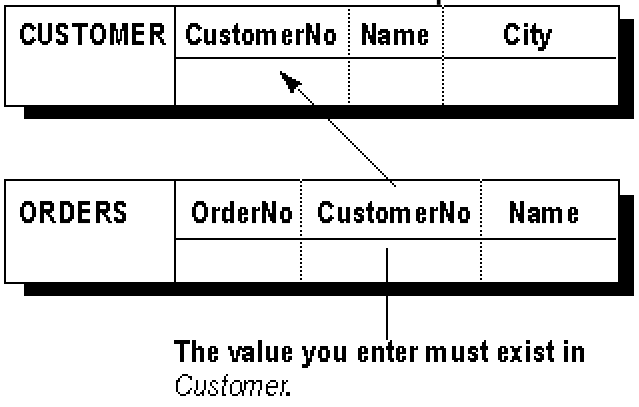
   Select the value you want from the highlighted field.
2. To make sure you have the right value, you can scroll to other fields; the scroll lock is on in a lookup table, so the lookup field stays onscreen while you scroll.
3. Choose OK to close the lookup table and insert the selected value into your table. Some table lookups are designed to also fill in other fields with the same name and type as the fields in the lookup table.

Examples of table lookup

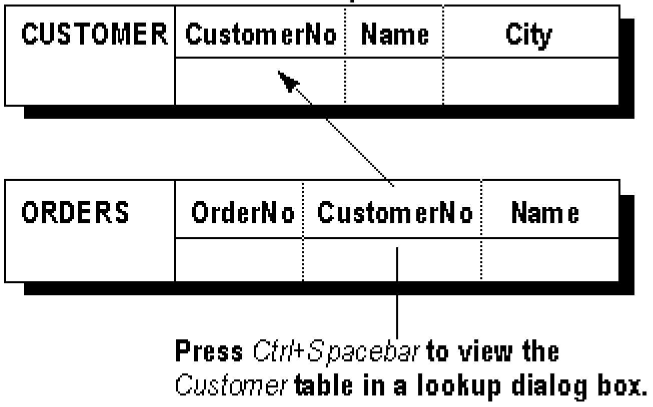
See also

The following figures illustrate table lookup.

Just Current Field with Fill No Help

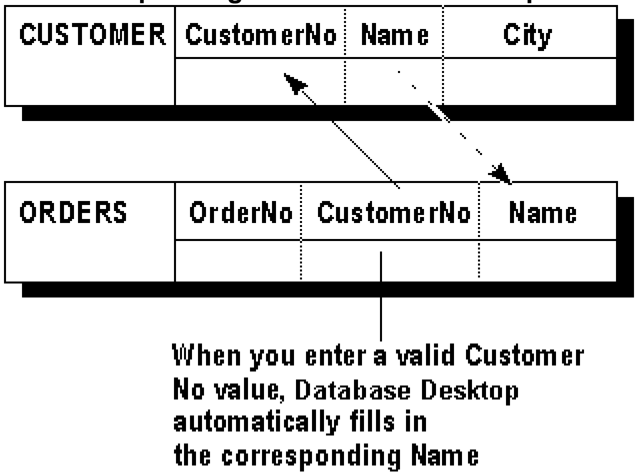


Just Current Field with Help And Fill



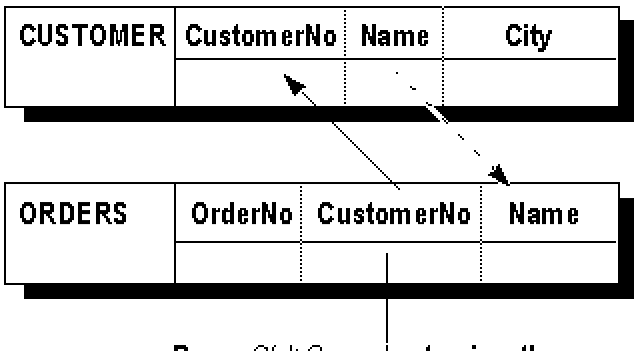
CUSTOMER
CustomerNo
Name
City
ORDERS
OrderNo
CustomerNo
Name
Press *Ctrl+Spacebar* to view the *Customer* table in a lookup dialog box.

All Corresponding Fields with Fill No Help



CUSTOMER
CustomerNo
Name
City
ORDERS
OrderNo
CustomerNo
Name
When you enter a valid Customer No value, Database Desktop automatically fills in the corresponding Name

All Corresponding Fields with Help And Fill



Press *Ctrl+Spacebar* **to view the** *Customer* **table and fill in Customer No and Name field values.**

▪

Example of using Just Current Field with Fill No Help

See also

Suppose you are editing an Orders table in which the Customer No field has a table lookup defined as Just Current Field and Fill No Help to the Customer No field of a Customer table. This means any value entered in the Customer No field of the Orders table must be a value that already exists in the Customer No field of the Customer table. With Fill No Help, the data entry person must already know what these values are.

If you enter an invalid value, Database Desktop displays the message "Field value fails lookup validity check." You cannot move off the record until you enter a valid value.

To get out of a field that requires a value to be entered, type any character or number.

If the field is constrained by referential integrity requirements, choose Edit|Undo to undo your entry. Find out what the acceptable values are before you continue.

▪

Example of using Just Current Field with Help And Fill

See also

Suppose you are editing an Orders table in which the Customer No field has a table lookup defined as Just Current Field and Help And Fill to the Customer No field of a Customer table. In Orders, when the Customer No field is selected, you will see the message "Press Ctrl+Space for lookup" in the Desktop status bar. You can either enter a valid value in the field or press Ctrl+Spacebar.

If you press Ctrl+Spacebar, the lookup table (Customer) appears in a dialog box on top of the table you are editing. A scroll lock is placed to the right of the lookup field (Customer No). If there is a valid value in the table you are editing, the current record marker indicates that value in the lookup table. For example, if you enter 1320 and then press Ctrl+Spacebar, the current record marker is on the value 1320 in the lookup table.

From the lookup table, select the value you want to enter. When you choose OK, the value is filled in and the dialog box containing the lookup table disappears.

Example of using All Corresponding Fields with Fill No Help

See also

Suppose you are editing an Orders table in which the Customer No field has a table lookup defined as All Corresponding Fields and Fill No Help to the Customer No field of a Customer table. This Orders table also has a Name field that contains the customer's name.

When you enter a valid value in the Customer No field of Orders, the correct value for the Name field is automatically filled in. This is because the Name field of Orders corresponds to the Name field of Customer.

If you enter an invalid value, Database Desktop displays an error message. You cannot move off the record until you enter a valid value.

To get out of a field that requires a value to be entered, type any character or number.

If the field is constrained by referential integrity requirements, choose Edit|Undo to undo your entry. Find out what the acceptable values are before you continue.

Example of using All Corresponding Fields with Help And Fill

See also

Suppose you are editing an Orders table in which the Customer No field has a table lookup defined as All Corresponding Fields and Help And Fill to the Customer No field of a Customer table. This Orders table also has a Name field that contains the customer's name.

You can enter data into the Customer No field by typing it in, or you can press Ctrl+Spacebar to display the lookup table (Customer) in a dialog box. When you choose a Customer No value, Database Desktop enters it and all corresponding field values (like Name) in the Orders table.

About DDE

See also

Dynamic Data Exchange (DDE) lets you communicate with other applications that support DDE. You can use DDE to send field values from Database Desktop to other applications, or to send data from other applications to a table or query in Database Desktop.

DDE links are shown as text, not icons or data. For an example, see To use Database Desktop as a DDE client (tables).

To use Database Desktop as a DDE server

See also

When you take the values from a field in Database Desktop and place them in another application, you are using Database Desktop as a DDE server.

Using Database Desktop as a DDE server

Suppose you have a spreadsheet that performs a series of calculations on a value. The value you want to perform the calculations on is in a field of a Paradox table.

1. In a Database Desktop Table window, select any value in the field, then choose Edit|Copy to copy the field to the Clipboard.
2. In the DDE-client spreadsheet, use Paste Link (or Paste Special, in some applications) to place the field in the appropriate spreadsheet cell. Remember, you do not place an actual value in the spreadsheet. Instead, you use DDE to tell the spreadsheet where to look for the value.

As you move through the records of your Paradox table, the values in the spreadsheet change because the value in the field is different for different records. The spreadsheet displays the field value for the selected Paradox record.

Note: You can use DDE to place Paradox table fields in any type of application that is a DDE client. Spreadsheets, word processors, and a variety of other applications can accept Paradox field values through DDE.

To link an entire table through DDE, choose Edit|Select All, then Edit|Copy.

To use Database Desktop as a DDE client (tables)

See also

When you use Database Desktop as a DDE client, you place link information about a value from another application into an alpha field in a Paradox table.

A common use of Databse Desktop as a DDE client is to use values from another application and perform queries on them in Database Desktop.

To use Database Desktop as a DDE client,

1. Copy the value you want to use (your DDE server can be a spreadsheet, word processor, or any other DDE-capable application).
2. In Database Desktop, select the alpha field where you want to place the DDE value, then choose Edit|Paste Link.

   You see link information like @DDE:"QPW"!"C:\QPW\NOTEBK1.WB1"!"$A$D$2"!@. This is a string that tells Database Desktop where to look for the DDE value. This particular string tells Database Desktop to look for a Quattro Pro for Windows file located on C:\QPW in Notebook 1, page A, cell D2.

In Database Desktop, you view the link information rather than the DDE value. To view the value in the DDE server, select the field and press Shift+F2. Database Desktop displays a message telling you it is launching the DDE server, then opens the application and the correct file.

To use Database Desktop as a DDE client (queries)

See also

1. Highlight the item in the server, then copy it to the Clipboard. Most servers use Edit|Copy to place a copy of the object on the Clipboard.
2. Return to the client (Database Desktop) Query window.
3. Select the QBE field to receive its value from the server.
4. Choose Edit|Paste Link from the menu. The DDE link information appears in the query.
5. Choose Query|Wait For DDE to tell Database Desktop to execute the query each time data is sent from the server.

To disconnect a DDE link

See also

After a DDE link is pasted into a DDE-client application, the Table|Notify On command is activated in Database Desktop. While this command is active, the link is live. For example, when you select another record in the linked table (in Database Desktop), the new value is delivered to the DDE client.

To disconnect the link,

- Uncheck Table|Notify On in Database Desktop.

While this command is inactive, no changes are delivered to the DDE client.

To reconnect the link at any time, choose Table|Notify On.

If you create a DDE link to an entire table, Table|Notify On works similarly. When any record in the linked table changes, the entire table is refreshed in the DDE client. Changes are posted in the table whenever the person editing the table moves off the record.

Example of Database Desktop as a DDE server

See also

For example, suppose you want to place a Paradox field's value in a cell in a Quattro Pro for Windows spreadsheet. The following example shows how to do this using the sample Orders table.

1. In Database Desktop, open Orders, a Paradox table. Select the first record's Total Invoice value.
2. Click the Copy to Clipboard Toolbar button. Database Desktop places the value on the Clipboard.
3. Open Quattro Pro for Windows. Select a notebook cell and choose Edit|Paste Link.
4. To see how DDE works, place your Database Desktop window and your Quattro Pro window together on the screen.

   Select the Total Invoice field in Database Desktop and press the up and down arrows to move through invoice values. Notice how the value shown in the notebook cell in Quattro Pro changes to display the Total Invoice value in the currently selected Paradox table record.

   In Quattro Pro, you can create calculations that use the value from Database Desktop. As the DDE value is updated, the calculated result is updated along with it.

Example of Database Desktop as a DDE client and server (queries)

See also

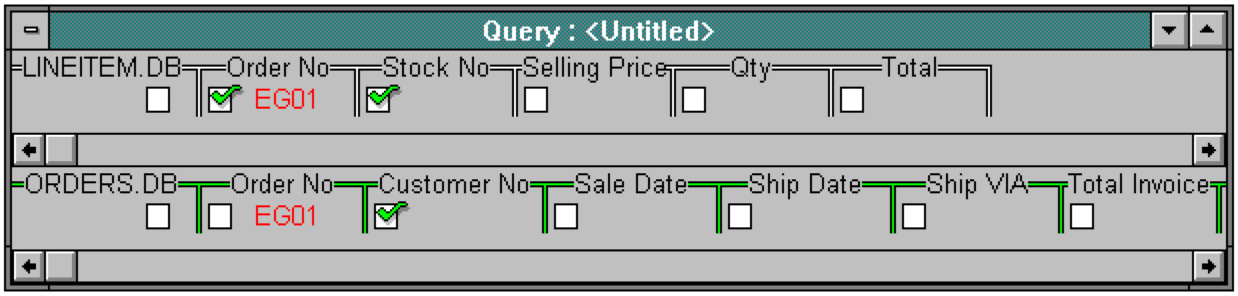
When you use Database Desktop as both DDE client and server, all actions can be performed within Database Desktop.

For example, a linked field can run a query (the DDE client). When the field value changes in the source table (DDE server), an updated Answer table appears.

Using DDE to run a query

Suppose you want to run a separate query for each customer in the Customer table. Follow these steps:

1. Open the Query window and add the Orders and Lineitem tables to it.
2. Construct a query that looks like this:



3. Open Customer in a Table window.
4. In Customer, select Customer No 1221 and click the Copy button on the Toolbar.
5. In the Query window, position the text insertion point in the Customer No field of the Orders table. Choose Edit|Paste Link. Link information from the Customer table appears in the field.
6. Click the Run Query Toolbar button. Database Desktop creates an Answer table listing all of Customer No 1221's items.

Using DDE to run a query interactively

Create a DDE link, following steps 1 through 5 above. Then,

1. Click the Query window's title bar to activate the window. Choose Query|Wait For DDE.
2. Click the Customer table's title bar to activate the window. Select Customer No 1221. Press the Down arrow to move to Customer No 1231. When you select the new value, Database Desktop activates the DDE link and runs the query again, updating the Answer table with the new value's data.

   You can uncheck Query|Wait For DDE if you want to scroll quickly through the Customer table without running a query on each record's value.

▪

About queries

See also

What is a query?

A query is a way to retrieve information from your tables. Queries are usually in the form of a question. For example, you can find out

- Which customers have placed orders this month?
- What is the total amount of all orders placed by each customer?
- What orders have not been paid?

Uses of queries

By constructing queries that build on each other, you can play "what if?" with your data. For example, you can find out

- How much would total sales increase if sales to Oregon residents increased by 8%?
- How much would our travel costs increase if airline prices went up 10%?

You can also use a query to perform calculations on your data. And you can insert, delete, and change records using INSERT, DELETE, and CHANGETO queries.

QBE

The query method Database Desktop uses is called query by example (QBE). To perform a QBE query, you give Database Desktop an example of the result you want. You use selection conditions and example elements to define the query. Then, you can save the query definition to use again.

In a query, you can specify

- Tables to ask questions about
- Fields you want to see in the Answer table
- Records you want to select
- Calculations you want to perform
- New fields you want to create

You can query one table or several tables to get just the information you need. Database Desktop finds the records that meet the conditions you specify and presents the results to you in an Answer table.

If a query does not quite obtain the results you want, you can easily refine it and perform the query again.

Query results

By default, Database Desktop prepares an Answer table for queries that yield a table of results. You can edit Answer tables, but your edits don't update the original table or the tables included in the query. If you want to update related tables by editing query results, you can create a live query view instead of an Answer table. For more information, see About query results.

Query properties and preferences

You can set properties for each query, such as the type and name of the results table, whether the results are sorted, and more. For more information, see About query properties.

To open a query

See also

To open a query from the Desktop,

1. Do one of the following:

- Click the Open Query button.
- Choose File|Open|Query.

The Open Query dialog box appears.

2. Choose a query from the list. Database Desktop opens a Query window displaying the selected query.

To run a query

See also

To run a query from the Query window, do one of the following:

- Click the Run Query ⚡ button.
- Choose Query|Run Query
- Press F8.

If the query contains no errors, Database Desktop displays a window to tell you the status of the query. After Database Desktop completes the query, depending on the kind of query it is, Database Desktop either displays an Answer table or changes data in a table. See About query results for more information.

To save a query

See also

You can save a query for later use.

- Choose File|Save or File|Save As.

If you close the Query window without saving, Database Desktop prompts you to save the query.

When you save a query, it becomes an object like any other Database Desktop object. You can open it, minimize it and display it.

▪

Guidelines for creating queries

See also

The types of queries you can create with Database Desktop and QBE are almost limitless. You can use query operators and calculation statements to extract just the information you need. No matter what kind of query you're creating, the technique you use to create it has very little variation.

When you create a query, you

- Choose the table or tables you want to ask about.
- Link the tables you've selected (if you're creating a multi-table query). See Using example elements to link tables.
- Select the fields you want displayed in the Answer table.
- Specify selection conditions for choosing specific records (optional).
- Specify calculations to perform on the data (optional).
- Set query properties, such as table type and sort order (optional).
- Run the query.
- Save the query (optional).

Additionally, you can customize the Answer table and to save it under a different name. Or, you can choose to view a live query instead of an Answer table by changing a query property setting. For more information, see About query results.

To create a query from a table

See also Example

1. Do one of the following:

- Right-click the Open Query
- button and choose New.
- Choose File|New|Query.

2. The Select File dialog box appears. Type the name of the table you want to query or select one or more tables from the list of files. (To select multiple tables, see To select from lists.)
3. Choose Open.

   Database Desktop places an image of each table chosen in the Query window.
4. Enter selection conditions and/or example elements and specify fields to display in the Answer table. For an overview, see Working with the query image.

Note: If your query contains more than one table, you must link the tables with example elements before you run the query. See Using example elements to link tables.

To create a query based on another query

See also

1. Do one of the following:

- Right-click the Open Query
- button and choose New.
- Choose File|New|Query.

The Select File dialog box appears.

2. Choose Queries in the Files Of Type drop-down list.
3. Type the name of the query you want to use as a base for the new query or select a query from the list of files.
4. Choose Open.

   Database Desktop places in the Query window an image of each table used in the chosen query. Any existing selection conditions, example elements, (and inclusion operators, if necessary) are included in the query images.

▪

Example of creating a simple query

See also

This topic provides an example of creating a query. For generic instructions, see To create a query from a table.

To create a simple query that results in a list of customer names and phone numbers, follow these steps. (This example uses the Customer table located in your SAMPLE directory.)

1. Make sure your working directory is set to SAMPLE under the directory with your Database Desktop program files. To change it, see To change your working directory.
2. Choose File|New|Query (or right-click the Open Query ▪ button and choose New). You will see the Select File dialog box.
3. From the Select File dialog box, select CUSTOMER.DB and choose Open (or double-click CUSTOMER.DB). Database Desktop places an image of the Customer table in the Query window.
4. Click the check box in the Name field. Database Desktop places a checkmark in the check box. The default checkmark type is Check, which shows only unique records (the first record that has each value) for that field.
5. Use the scroll bar at the bottom of the table's query image to move to the right of the image until you see the Phone field.
6. Click the check box in the Phone field. Database Desktop places a checkmark in the check box.
7. Choose Query|Run Query (or click the Run Query button, or press F8). Database Desktop displays a status window to track the progress of the query.

When Database Desktop finishes gathering the data you want, it displays the data in an Answer table on the Desktop on top of the Query window.

▪

Using the Query window

See also

The Query window appears when you open a query or create a new query.

The Query window contains query images of each table in the query. For information on query images, see Working with query images.

Query table scroll bars

Each table represented in the query has its own horizontal scroll bar. This lets you scroll to any columns that are not visible.

The master vertical scroll bar on the right side of the Query window lets you scroll the whole query. This lets you view any table query images that are not visible.

Tiling or cascading query images

Database Desktop provides two ways for you to display multiple query images in a Query window.

- Choose View|Tile Tables to view multiple query images tiled vertically in the Query window. (This is the default setting.)
- Choose View|Cascade Tables to view multiple query images cascaded in the Query window.

To add tables to a query

See also

You can open a query with one or more tables and add tables to it.

To add tables to a query,

1. Do one of the following:

- Click the Add Table button.
- Chose Edit|Add Table.

The Select File dialog box appears.

2. Type the name of the table or select one or more tables from the list of tables. (To select multiple tables, see To select from lists.)

   When you choose Open, Database Desktop places an image of each table chosen in the Query window.

Queries that contain more than one table must be linked with example elements. See Querying more than one table for more information.

To remove tables from a query

See also

You can open a query with one or more tables and remove tables from it.

1. Do one of the following:

- Click the Remove Table button.
- Chose Edit|Remove Table.

   The Remove Table dialog box appears.

2. Select one or more tables from the list of tables. (To select multiple tables, see To select from lists.)

   When you choose OK, Database Desktop removes the images of the selected tables from the Query window.

Working with query images

See also

The Query window contains query images of each table in the query. The query image has the same fields, in the same order, as the table it represents, but no data. If you have changed the table's properties (for example, changed the column order or the way heading text is displayed), the query image does not reflect them. However, you can change the column order of the query image.

Working with a query image

Type selection conditions and/or example elements into the fields of the query image. (See About selection conditions and About example elements for details.)

You type data into and navigate through the fields of a query image the same way you would in a table in Edit mode. For example:

To...	Do this...
Add a row	Press the Insert key (this only works if you have made some change to the current row).
Delete a row	Press Ctrl+Del.
Enter Field View	Press F2.

Moving among fields

To move among fields within a query image using the keyboard, press Tab or Shift+Tab. To do this in multi-table queries, press Super Tab (F4) or Super Back Tab (F3).

Selecting fields to display in the Answer table

Use the query image in the Query window to tell Database Desktop what fields of the table to include in the Answer table. For details, see About selecting fields to display.

Query image check boxes

See also

Each field of a query image has a check box. The column on the far left under the table name also has a check box. Click a field's check box to include that field in the Answer table for the query. When you right-click a field's check box, you see the different types of checks you can use. Each has its own meaning.

Use **Check** to show all unique values for the checked field. The values are displayed in ascending order (A to Z or 0 to 9). When used with a summary operator, a checkmark specifies that the records be divided into groups based on the values in the checked field.

Use **CheckPlus** to show all values in a field, including duplicates, without sorting. When you use CheckPlus, the values from the checked field appear in the Answer table in the same order they appear in the queried table.

When you use CheckPlus in any field of the query image, it overrides any Checks or CheckDescendings you have placed in any other field. This is because Database Desktop cannot sort and exclude duplicates which is what the Check and CheckDescending tell it to do

and not sort and include duplicates

which the CheckPlus tells it to do.

Note: Although you can place Checks and CheckDescendings in BLOB fields, Database Desktop treats them as CheckPluses in these fields. This is because Database Desktop cannot sort or distinguish unique from duplicate values in these field types.

Use **CheckDescending** to show unique values sorted in descending order (Z to A or 9 to 0).

Use **GroupBy** to specify a group of records to use in a set query. (A field with the GroupBy checkmark does not appear in the Answer table.) For more information, see About querying sets of records.

Use this to remove a check.

Query image fields

See also

The fields of a query image hold the selection conditions for your query. You define query selection conditions by typing them directly into the query image fields.

You can use the Edit menu to perform cut, copy, and paste operations on any selection condition or portion of a selection condition in a field of a query image. Use standard Windows procedures to select material to be cut or copied. Then use the Edit menu to perform the cut, copy, and paste operations.

To place a checkmark in a query image

See also

To place the default checkmark,

Usually a Check, do one of the following:

- Click the field's check box.
- Select the field and press F6.

To place another type of checkmark,

Usually CheckPlus, CheckDescending, or GroupBy, do one of the following:

- Right-click the field's check box to display the check menu, then choose the type of check you want from the menu.
- Select the field and press Shift+F6 repeatedly until the type of check you want is displayed.

Shortcut: To include all fields in the Answer table, click the check box in the left-most column (under the table name).

Note: For a description of the different types of checkmarks, see Query image check boxes.

To rotate columns in a query image

See also

Do one of the following:

- Use the mouse to drag a column heading to a new location.
- Select the column you want to move and press Ctrl+R. The selected column becomes the last column in the table.

To change the order of columns in a query Answer table, first fill out the query image. Then, before running the query, choose Query|Properties and click the Structure page. You can use the arrows to change the order of the fields.

▪

About selecting fields to display

See also

In a query image, you need to specify what fields you want to see in the Answer table.

▪ If you place a Check in one field of a query image, Paradox displays only unique values from that field in the Answer table.

▪ If you want to see all values, including duplicates, select CheckPlus instead of the Check from the check box menu.

When you use CheckPlus, the values are not sorted.

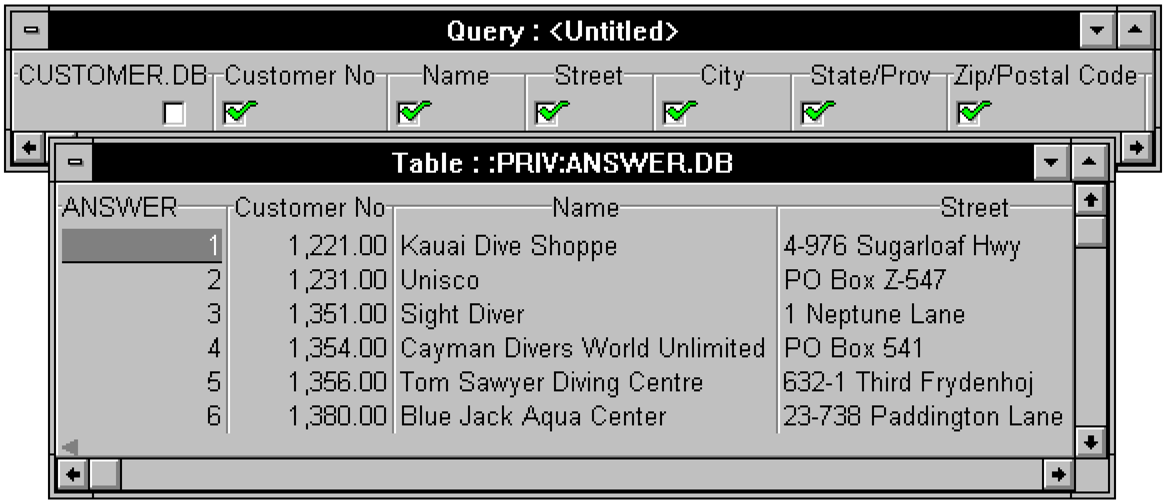
For more information on the effects of these and other checkmarks, see Query image check boxes.

Including a field

To include a field in the Answer table, place a checkmark in the field's check box. For instructions, see To place a checkmark in a query image.

Selecting all fields

To select all fields, check the box under the table name in the leftmost column.



Query : <Untitled>

CUSTOMER.DB	Customer No	Name	Street	City	State/Prov	Zip/Postal Code
☐	✓	✓	✓	✓	✓	✓

Table : :PRIV:ANSWER.DB

ANSWER	Customer No	Name	Street
1	1,221.00	Kauai Dive Shoppe	4-976 Sugarloaf Hwy
2	1,231.00	Unisco	PO Box Z-547
3	1,351.00	Sight Diver	1 Neptune Lane
4	1,354.00	Cayman Divers World Unlimited	PO Box 541
5	1,356.00	Tom Sawyer Diving Centre	632-1 Third Frydenhoj
6	1,380.00	Blue Jack Aqua Center	23-738 Paddington Lane

Unchecking a field

To uncheck a field, click the check box again, or press F6.

To specify field names in the Answer table

See also

Paradox displays a field in the Answer table with the same name it has in the original table, or, in some cases, with a name Paradox assigns. If you want a field in the Answer table to have a different name, use the AS operator.

To specify a different name,

1. Type your selection condition, if any, in the field, then type AS, followed by a space.
2. Type the name you want the field to be called in Answer.

In the Answer table, Paradox displays the values under the field name you specified.

See To rename Answer table fields.

-

About selection conditions

See also

In most queries, you want to see only records that meet certain conditions. You specify the conditions you want records to meet by typing the conditions in the fields of a query image.

You use the query operators to define selection conditions.

You can define selection conditions that test for these types of matches:

- Exact matches
- Matching a range of values: comparison operators
- Inexact matches: the LIKE operator
- Non-matches: the NOT operator
- Blank values: the BLANK operator
- Today's date: the TODAY operator
- Using wildcards to match a pattern

You can also use AND and OR to indicate whether a record must match all the defined selection conditions or just one of them.

You can type a selection condition in a field without checking that field. You do not have to include a field in the Answer table to use its values to select records. For example, you can query a table containing names and addresses for a list of people living in a particular state without including the state field in the Answer table.

You must follow certain rules when entering selection conditions and calculation statements in query images. For details, see

- Entering numbers in queries
- Using reserved words or symbols in selection conditions

▪

Query operators

See also

Paradox query operators are grouped into seven types:

Category	Operator	Meaning
Reserved symbols	Check	Display unique field values in Answer
	CheckPlus	Display field values including duplicates in Answer
	CheckDescending	Display field with values sorted in descending order
	GroupBy check	Specify a group for set operators
Reserved words	CALC	Calculate a new field
	INSERT	Insert records with specified values
	DELETE	Remove records with specified values
	CHANGETO	Change specified values in fields
	SET	Define specific records as a set for comparisons
Arithmetic operators	**+**	Addition or alphanumeric string concatenation
	-	Subtraction
	*****	Multiplication
	/	Division
	( )	Group arithmetic operations
Comparison operators	**=**	Equal to (optional)
	>	Greater than
	<	Less than
	>=	Greater than or equal to
	<=	Less than or equal to
Wildcard operators	**..**	Any series of characters
	@	Any single character
Special operators	LIKE	Similar to
	NOT	Does not match
	BLANK	No value
	TODAY	Today's date
	OR	Specify OR conditions in a field
	, (comma)	Specify AND conditions in a field
	AS	Specify the name of a field in Answer
	! (exclamation mark)	Display all values in a field, regardless of matches
Summary	AVERAGE	Averages the values in a group
	COUNT	Counts the number of values in a group
	MAX	Finds the maximum value of a group
	MIN	Finds the minimum value of a group
	SUM	Totals the values in a group
	ALL	Calculate summary based on all values in a group,

		including duplicates
	UNIQUE	Calculate summary based on unique values in a group
Set comparison operators	ONLY	Display records that match only members of the defined set
	NO	Display records that match no members of the defined set
	EVERY	Display records that match every member of the defined set
	EXACTLY	Display records that match all members of the defined set and no others

▪

Operator precedence in queries

See also

Paradox evaluates operators in queries in a certain order.

In expressions containing more than one operator, the operators are evaluated in the order of precedence shown in the following table.

Precedence	**Operator**
1	()
2	* /
3	+ -
4	= <> < <= > >=
5	NOT
6	OR
7	, (comma)

Any expression contained in parentheses is evaluated first, and inner levels of parentheses are evaluated before outer levels. When two or more operators of equal precedence are in a single expression, they are evaluated from left to right.

▪

Using arithmetic operators

See also

You can use arithmetic expressions in number, date, time, and money fields of a query image.

Operator	Meaning
+	Addition or string concatenation
-	Subtraction
*****	Multiplication
/	Division
( )	Used to group expressions

Use parentheses ( ) to combine and group operations and to indicate which calculations should be performed first. In expressions without parentheses, multiplication and division are performed before addition and subtraction. Operations with equal precedence are calculated from left to right. For more information on operator precedence, see Operator precedence in queries.

Arithmetic operators are especially useful with the TODAY operator, the CALC operator, and with example elements.

You can use arithmetic expressions with date values and the TODAY operator to

- Add a number of days to a date
- Subtract a number of days from a date
- Subtract a date from a date resulting in a number of days

Use arithmetic operators to create arithmetic expressions with field values. You can use any of the arithmetic operators in the numeric fields▪Paradox number, short, long integer, BCD, and money and dBASE number and floating number fields.

For a list of Paradox and dBASE field types you can use, see:

- Paradox field types allowing arithmetic operators
- dBASE field types allowing arithmetic operators

You can use the addition (+) operator in alpha fields to combine or concatenate alpha values.

For general information on all the query operators, see Query operators.

Paradox field types allowing arithmetic operators

See also

This table shows which arithmetic operators can be used in each Paradox field type.

Operator	A	N	$	S	I	#	D	T	@	M	F	G	O	L	+	B	Y
+	Y	Y	Y	Y	Y	Y	Y	Y	Y								
-		Y	Y	Y	Y	Y	Y	Y	Y								
*		Y	Y	Y	Y	Y		Y	Y								
/		Y	Y	Y	Y	Y		Y	Y								
( )	Y	Y	Y	Y	Y	Y	Y	Y	Y								

dBASE field types allowing arithmetic operators

See also

This table shows which arithmetic operators can be used in each dBASE field type.

Operator	C	F	N	D	L	M	O	B
+	Y	Y	Y	Y				
-		Y	Y	Y				
*		Y	Y					
/		Y	Y					
( )		Y	Y	Y				

Entering numbers in queries

See also

When you type a number into a numeric field (Paradox number, short, long integer, or money field and dBASE number or floating number fields) of a query image,

- Do not type dollar signs.
- Do not type parentheses to signify a negative value.
- Do not type thousand separators (a comma in U.S. convention and a period in international convention) when specifying a pattern match with the **..** or @ wildcard operators. See Using wildcards to match a pattern.

On the other hand,

- Do type decimal separators (a period in U.S. convention and a comma in international convention).
- Do type the minus symbol to signify a negative value.
- Optionally, do type thousand separators when specifying an exact match numeric selection condition.

Paradox determines when a comma or a period is a whole-number or a decimal separator, first based on whether you have U.S. or international number convention set, and second, based on the symbol's position and context. Ambiguity arises when a comma could be Paradox's AND operator, which is a comma, and when a period could be part of Paradox's **..** wildcard operator, which is two periods in a row.

If a comma's or period's meaning is not clear, then you must help Paradox understand the symbol's meaning with double quotation marks or spaces. A comma's or period's meaning will not be clear as a thousand separator if you are specifying a pattern match with the **..** or @ wildcard operators; thus, do not type thousand separators when specifying a numeric pattern with **..** or @.

If you have the U.S. number format set,

- Paradox interprets a single period in a numeric field as a decimal separator.
- Paradox interprets the first two periods in a row as the **..** wildcard operator.

In a numeric field, if Paradox encounters three periods in a row, it interprets them as the **..** wildcard operator followed by the decimal separator. To make Paradox interpret the first period as the decimal separator, enclose it in double quotation marks.

- Paradox interprets a comma in a numeric field as a thousand separator if you are specifying an exact match and if the comma is in the proper position to be a thousand separator. To make Paradox interpret a comma as the AND operator where this meaning might not be clear, type a space or any other non-numeric character except @ or a period after the AND comma. For example, you could type a comparison operator.

If you are using the international number format,

- Paradox interprets the first comma in a numeric field within a number as the decimal separator.
- Paradox interprets a comma followed by a space or any other non-numeric character except @ or a period as the AND operator in a numeric field.
- Paradox interprets a single period in a numeric field as a thousand separator if you are specifying an exact match and if the period is in the proper position within a numeric selection condition to be one.

▪

Using reserved words or symbols in selection conditions

See also

In a query image, to specify an alphanumeric value that contains a period or comma or a Paradox reserved word, enclose the value in double quotation marks. Paradox then recognizes the quoted characters as a value and does not act on their special meaning.

If the value itself contains a double quotation mark, precede the quotation mark with a backslash (\):

```
   Thomas E. \"Ned\" Lawrence
```

If the value contains a backslash, precede that backslash with another backslash (\\).

You do not need quotation marks to enclose blank spaces in a value. You do need them, however, for all other symbols and operators that have special meanings in Paradox, like commas, periods, and asterisks.

Exact matches

See also

If you want a query to retrieve only records that have a specific value in a field, type the value you are looking in the appropriate field of the query image.

Paradox includes in the Answer table only records with that value in that field.

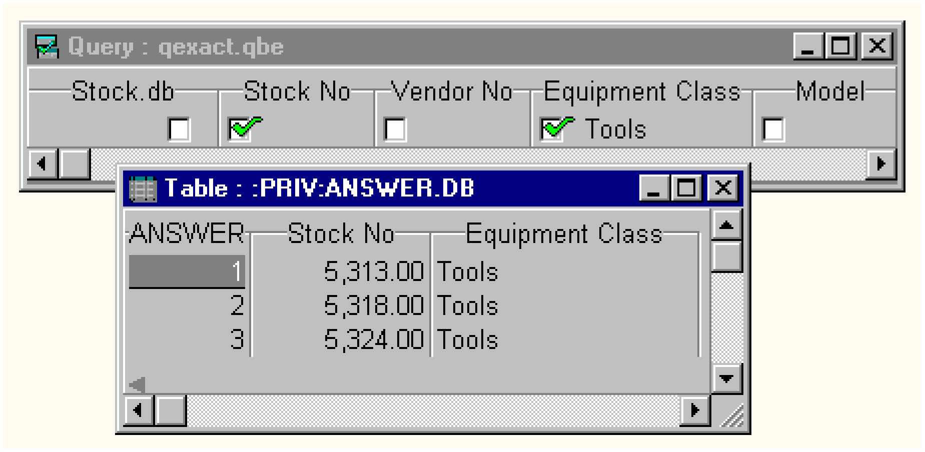
Exact matches are case-sensitive. You can specify exact matches for as many different fields as you like. Type all of the values you want to see exactly as they appear in the table in the appropriate fields of the query image.

Remember to check the field if you want it displayed.

Note: You cannot specify exact matches for BLOB fields. You must use the **..** wildcard operator to specify selection conditions in memo and formatted memo fields. See Using wildcards to match a pattern.

Exact matches of logical fields include uppercase or lowercase T and F and any combination of uppercase and lowercase letters of the entire words True and False.

Example



Stock.db	Stock No	Vendor No	Equipment Class	Model
	✓		✓ Tools	

ANSWER	Stock No	Equipment Class
1	5,313.00	Tools
2	5,318.00	Tools
3	5,324.00	Tools

Matching a range of values: comparison operators

See also

If you want a query to retrieve records that match a range of values, use comparison operators, also known as range operators. Comparison operators let you specify a range of values in a single field. For example, you might want to see any quantity greater than 10, any price less than $500, any date before June 13, 1992, or any name that comes before Smith in alphabetical order.

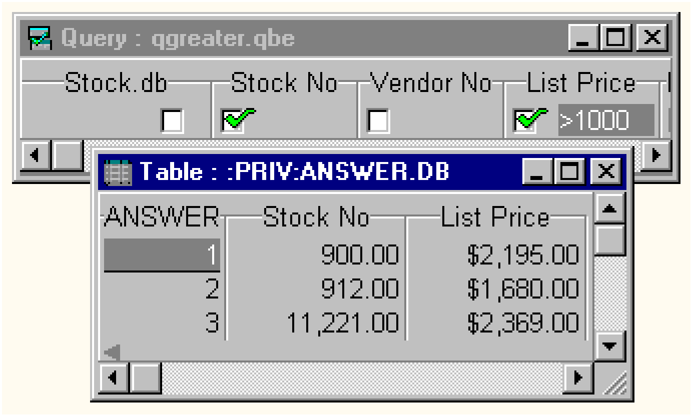
To use a comparison operator, type it in front of the value you are using to define the range.

You can use comparison operators with alphanumeric values and all number, date, and logical values. You cannot use them with BLOB or dBASE memo values; you can only use the equal to (=) operator with these types.

Operator	Meaning	Examples	Match
=	Equal to*	= 3/17/81	Only March 17,1981
		= Ralph	Only Ralph
		= False	Only False
>	Greater than	> 3/17/81	Dates later than March 17, 1981
		> "Ralph"	"Rat", "Rudolph", etc.
		> "False"	True, T, Yes, 1
<	Less than	< 3/17/81	Dates before March 17, 1981
		< "Ralph"	"Charles", etc.
		< "True"	False (by convention, False < True)
>=	Greater than	>= 3/17/81	March 17,1981 and later dates
	or equal to	>= "Ralph"	"Ralph", "Raphael", "Randolph", etc.
<=	Less than	<= 3/17/81	March 17, 1981 and earlier dates
	or equal to	<= "Ralph"	"Ralph", "Manny", "Charles", etc.

*The **=** operator is optional in these cases, because it is assumed when no other comparison operator is used.

To use a comparison operator, type it before the value you are interested in. If you are typing an alphanumeric value, you can use any combination of uppercase and lowercase letters to produce the same results. The example of all stock that costs more than $1000 is shown in the figure below.


Query : qgreater.qbe

Stock.db	Stock No	Vendor No	List Price
☐	✓	☐	✓ >1000

Table : :PRIV:ANSWER.DB

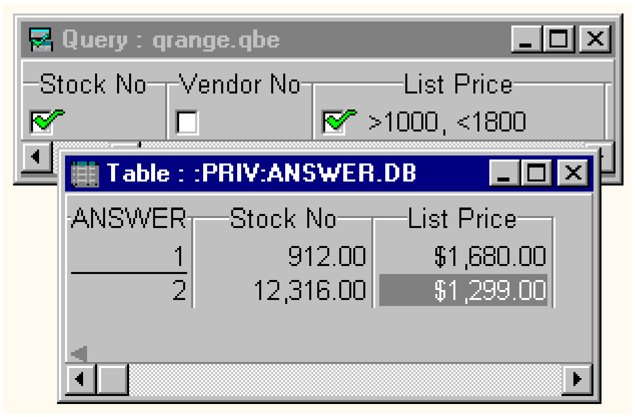
ANSWER	Stock No	List Price
1	900.00	$2,195.00
2	912.00	$1,680.00
3	11,221.00	$2,369.00

You can specify ranges for any number of fields in a query image.

Combining operators

You can combine comparison operators to construct a limited range of values. Separate all the comparison conditions with a comma. For example, the following query requests records with a List

Price greater than $1,000 and less than $1,800.



Query : qrange.qbe

Stock No	Vendor No	List Price
✓	☐	✓ >1000, <1800

Table : :PRIV:ANSWER.DB

ANSWER	Stock No	List Price
1	912.00	$1,680.00
2	12,316.00	$1,299.00

▪

Inexact matches: the LIKE operator

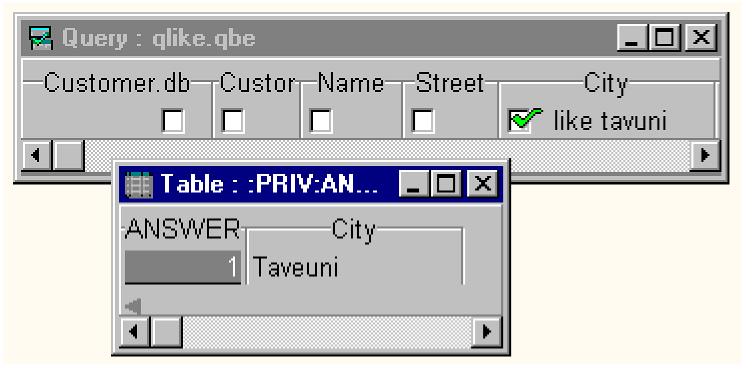
See also

Use the LIKE operator in a query image to match inexact alphanumeric values. This is particularly useful for finding values that contain typographical errors or alternate spellings.

If the Answer table to a query does not include some records you expected to see, try using LIKE with one or more alpha fields; the records you are looking for might contain typographical errors, misspellings, or alternate spellings.

To use the LIKE operator, type LIKE in front of the value you think will match the records you want.

Example



Customer.db	Custor	Name	Street	City
☐	☐	☐	☐	✓ like tavuni

ANSWER	City
1	Taveuni

Two general rules for obtaining a match with the LIKE operator are

- The first character of the pattern you specify with the LIKE operator must match exactly (though case does not matter). "LIKE California" does not match Kalifornia.
- A pattern matches if at least half to two-thirds of the characters match.

Field types

You cannot use LIKE on BLOB fields or dBASE memo fields.

While you can use LIKE in numeric and date fields, you will get better results using the wildcard operators **..** and @ to specify a numeric or date pattern.

Non-matches: the NOT operator

See also

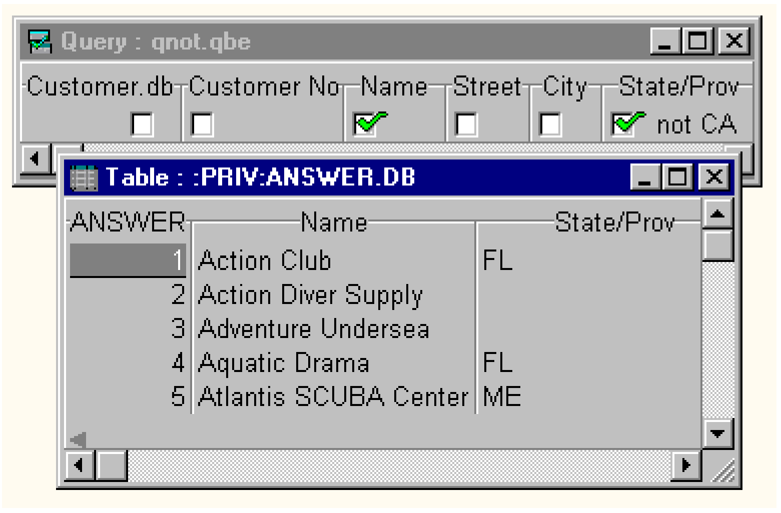
In a query image, use the NOT operator to select records that do not have a specified value in a particular field.

To use the NOT operator, type NOT before the example of the value you do not want to see.

NOT can precede exact values, ranges, wildcard patterns, or other selection conditions. In fact, you can precede any valid Paradox selection condition with NOT.

If the selection condition you specify after NOT is an exact match condition, you must type the condition exactly as the matching value appears in the table, with respect to capitalization and spelling. (Values in logical fields are an exception to this rule.) As with all of Paradox's operators, the case of the NOT operator does not matter.

Example



Customer.db	Customer No	Name	Street	City	State/Prov
☐	☐	✓	☐	☐	✓ not CA

ANSWER	Name	State/Prov
1	Action Club	FL
2	Action Diver Supply	
3	Adventure Undersea	
4	Aquatic Drama	FL
5	Atlantis SCUBA Center	ME

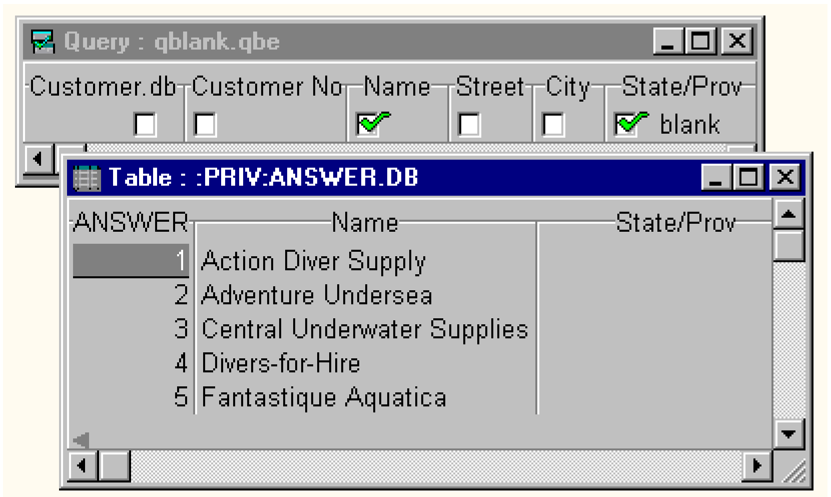
Blank values: the BLANK operator

See also

In a query image, use the BLANK operator to find records with no value in a specified field.

In some cases, the absence of a value is in itself a useful piece of information. Or you might want to find records with a blank field so you can fill in information unavailable when the record was entered.

To use the BLANK operator, type BLANK in the appropriate field.



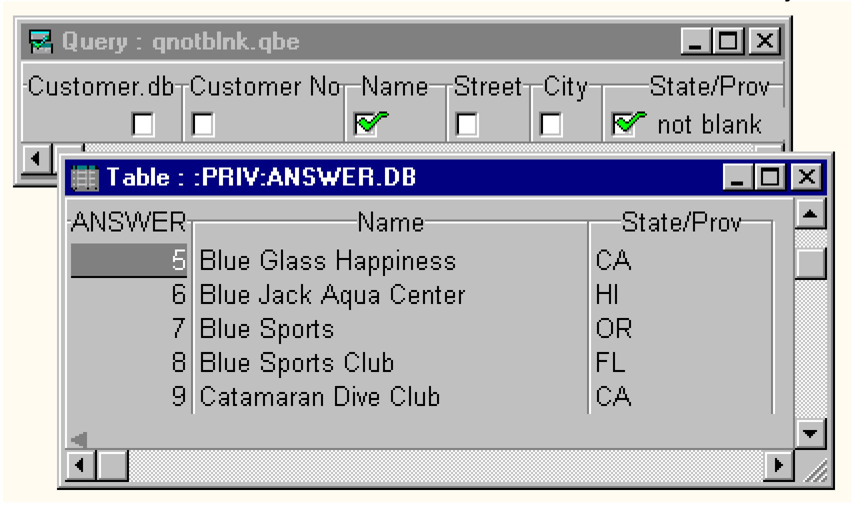
Query : qblank.qbe

Customer.db	Customer No	Name	Street	City	State/Prov
☐	☐	✓	☐	☐	✓ blank

Table : :PRIV:ANSWER.DB

ANSWER	Name	State/Prov
1	Action Diver Supply	
2	Adventure Undersea	
3	Central Underwater Supplies	
4	Divers-for-Hire	
5	Fantastique Aquatica	

You can combine NOT with BLANK to find all records that have any value in the specified field.



Query : qnotblnk.qbe

Customer.db	Customer No	Name	Street	City	State/Prov
☐	☐	✓	☐	☐	✓ not blank

Table : :PRIV:ANSWER.DB

ANSWER	Name	State/Prov
5	Blue Glass Happiness	CA
6	Blue Jack Aqua Center	HI
7	Blue Sports	OR
8	Blue Sports Club	FL
9	Catamaran Dive Club	CA

Note: Searching for blank field values is entirely different from leaving a field blank in a query image. Using the BLANK operator tells Paradox you want to see only those records that have no value in the specified field. When you leave the field of a query image blank, on the other hand, Paradox does not consider the field at all when selecting records.

When you use comparison operators or sort by a field that has blank values, blank fields are considered to be less than any nonblank value.

Today's date: the TODAY operator

See also

In date fields of a query image, the TODAY operator always stands for today's date. Make sure your computer's calendar is set properly.

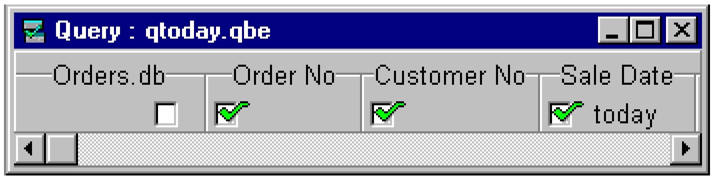
TODAY is especially useful for aging payables and receivables when used with Paradox's arithmetic operators.

For example:

Expression	Meaning
< TODAY	Finds dates earlier than today's date
< TODAY - 90	Finds dates earlier than 90 days ago
TODAY + 30	Finds dates 30 days ahead of today's date

Example

Suppose you want to query the sample Orders table to see what orders were placed today. This is how you would set up the query:



You could save this query and run it at the end of each day to see what orders were placed each day.

▪

Using wildcards to match a pattern

See also

Paradox provides two wildcard operators to match patterns of characters in queries:

- The .. operator, which matches any series of alphabetical or numeric characters
- The @ operator, which matches any single alphabetical or numeric character

Although the LIKE operator is useful for finding inexact matches in alpha fields, wildcard operators give you more flexibility.

You can use these operators in any field except in binary, graphic, OLE, or logical fields. You can type any combination of uppercase and lowercase letters, and your query will produce the same results.

Note: To retrieve values from a memo or formatted memo field, you must use the **..** wildcard operator to specify a pattern selection condition. (Typing an exact match in these field types means typing the entire memo value; to prevent this unnecessary effort, Paradox does not allow it.) You can also use the @ wildcard operator to specify a pattern match in these field types, but you must use it in combination with the **..** wildcard operator.

Special guidelines apply when you use wildcard operators with dates and numbers. See

- Using wildcards with dates
- Using wildcards with numbers

For an example of using wildcard operators with comparison operators, see

- AND conditions in the same field

The .. wildcard operator

See also

The **..** wildcard operator matches any series of any number of characters, including blank spaces. The .. wildcard operator is case-insensitive.

Pattern	Matches
G**..**	Giant, gigantic, Georgia
g**..**t	Giant, gross weight
..D	Grand, Elm Road
..e**..**s	Phillip Edward Wilson, roses
7**..**5	7485, 70,005
6/**..**/96	6/01/96, 6/25/96

Note: To retrieve values from a memo or formatted memo field, you must use the **..** wildcard operator to specify a pattern selection condition. (Typing an exact match in these field types means typing the entire memo value; to prevent this unnecessary effort, Paradox does not allow it.) You can also use the @ wildcard operator to specify a pattern match in these field types, but you must use it in combination with the **..** wildcard operator.

Suppose you want to find shops in the Customer table with the word Dive in their names. If you used the LIKE operator and typed LIKE dive in the Name field of the Customer table, you would only get dive shops whose names started with the word Dive and for whom Dive represented at least half of the letters of the entire name value. If, instead, you type ..dive.. in the Name field, Paradox generates an Answer table that shows customers with the word Dive anywhere in their name.

The @ operator

See also

The @ wildcard operator matches any single character (letter or number). You can use any number of @ characters to specify a pattern.

When you know how many characters are in the pattern you're looking for, you can use that number of @ wildcard operators instead of using the .. wildcard operator. For example, if you don't know if a person spells her name Kathy or Cathy, you can type @athy to match the value.

Pattern	Matches
m@@e	Mike, more, made
wom@n	Woman, women
s@@@@	Smith, Smyth, scent
19@2	1922, 1972, 1992

The @ wildcard operator is case-insensitive.

Note: You cannot use the @ wildcard operator by itself to specify a pattern in a memo or formatted memo field. You can use it to represent single characters in a memo or formatted memo field, but you must also use the .. wildcard operator to retrieve memo field values.

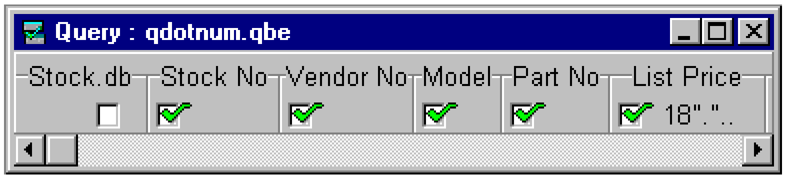
Using wildcards with numbers

See also

If a comma's or period's meaning is not clear, then you must help Paradox understand the symbol's meaning with double quotation marks or spaces. A comma's or period's meaning will not be clear as a thousand separator if you are specifying a pattern match with the **..** or @ wildcard operators; thus, do not type thousand separators when specifying a numeric pattern with **..** or @.

If there is a chance that a decimal or thousand separator will be confused with the **..** or @ wildcard operator, use quotation marks. For details, see Entering numbers in queries.

For example, here is a query to find all stock having a list price of $18 and any number of cents.



Stock.db	Stock No	Vendor No	Model	Part No	List Price
☐	✓	✓	✓	✓	✓ 18"."..

Paradox considers only significant digits in Paradox number fields when you use wildcard operators. For example, @@@.@ matches 400.70, because the last 0 isn't significant. By contrast, @@@.@@ doesn't match 400.70 for the same reason.

dBASE numbers

A dBASE number field has trailing zeros to the right of the decimal place, so add the **..** operator to the end of a numeric pattern, even if you are trying to match the last digits. For example, **...**95**..** will match all numeric values ending in **.**95, but **...**95 will not match.

Using wildcards with dates

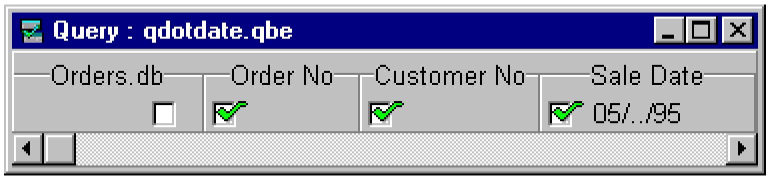
See also

In a query image, when entering date values for exact matches, you can use any date format that Paradox supports, including custom formats.

However, when you use a wildcard to find a date, the pattern you define with the wildcard operator must reflect the date format you have set in both the BDE Configuration Utility and the Windows Control Panel Regional settings. (The BDE and Control Panel date settings must match.)

Example

If the date format set in both BDE and Control Panel is mm/dd/yy, you can find orders placed in May of 1995 like this:



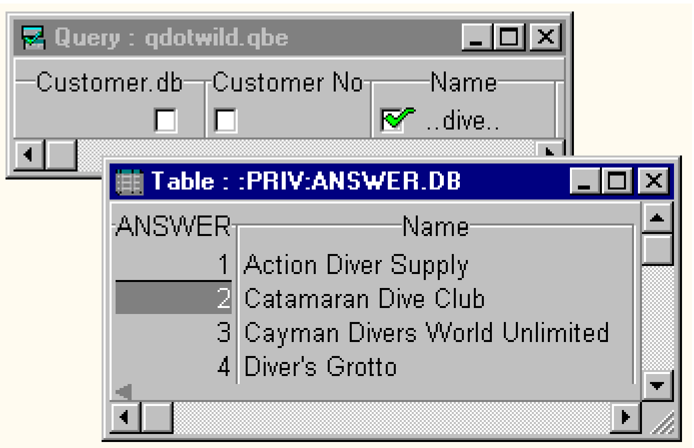
Orders.db	Order No	Customer No	Sale Date
	✓	✓	✓ 05/../95

If you have another date format set, use that in the wildcard query.

Example of using wildcards to match a pattern

See also

The following example shows the use of the **..** operator to find the name of all customer shops with Dive in their name.



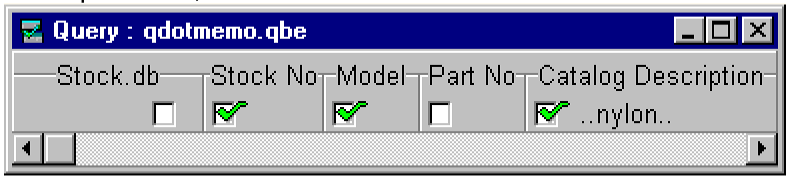
Query : qdotwild.qbe

Customer.db	Customer No	Name
☐	☐	✓ ..dive..

Table : :PRIV:ANSWER.DB

ANSWER	Name
1	Action Diver Supply
2	Catamaran Dive Club
3	Cayman Divers World Unlimited
4	Diver's Grotto

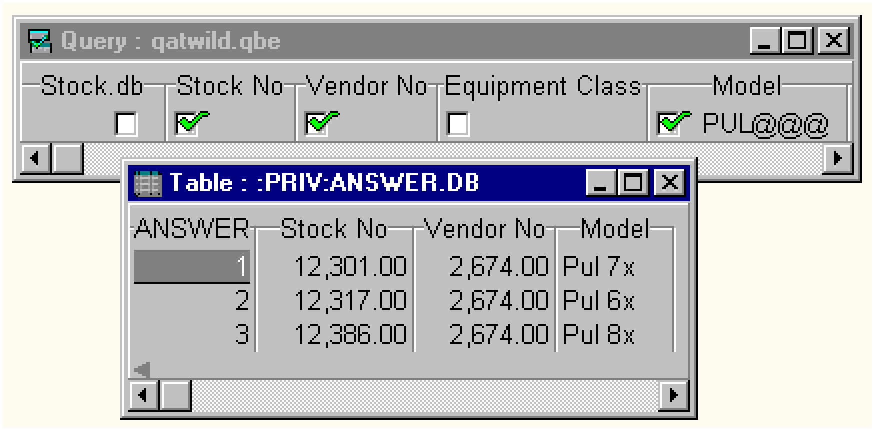
This example retrieves from the sample Stock table all records that have the word "nylon" in the Catalog Description field, which is a memo field.



Query : qdotmemo.qbe

Stock.db	Stock No	Model	Part No	Catalog Description
☐	✓	✓	☐	✓ ..nylon..

The next example shows the use of the @ operator to find all stock with Model name beginning with PUL plus 3 and only 3 characters. Notice that @ retrieves the blank space character as well as letters and numbers. If you used the **..** operator in this case (Pul**..**) the Answer table could give you anything from Pulse to Pullman.



Query : qatwild.qbe

Stock.db	Stock No	Vendor No	Equipment Class	Model
☐	✓	✓	☐	✓ PUL@@@

Table : :PRIV:ANSWER.DB

ANSWER	Stock No	Vendor No	Model
1	12,301.00	2,674.00	Pul 7x
2	12,317.00	2,674.00	Pul 6x
3	12,386.00	2,674.00	Pul 8x

About AND conditions

See also

When you enter selection conditions in separate fields on the same line of a query image, all conditions on that line must be met by a record in the table for the query to retrieve that record. This type of operation is called a logical AND, and means that all conditions must be met.

You can also express a logical AND in a single field▪that is, enter more than one condition in a field and require that they all be met

▪by separating the conditions with commas.

The comma acts as an AND operator, telling Paradox that both (or all) conditions must be met for a match to occur.

Note: If you want to enter a comma into a query without Paradox interpreting it as the AND operator, enclose it in quotation marks.

You can use the AND operator in all field types including BLOBs. Whenever you query a memo or formatted memo field, you must use the .. wildcard operator in addition to any other selection conditions or operators you use.

▪

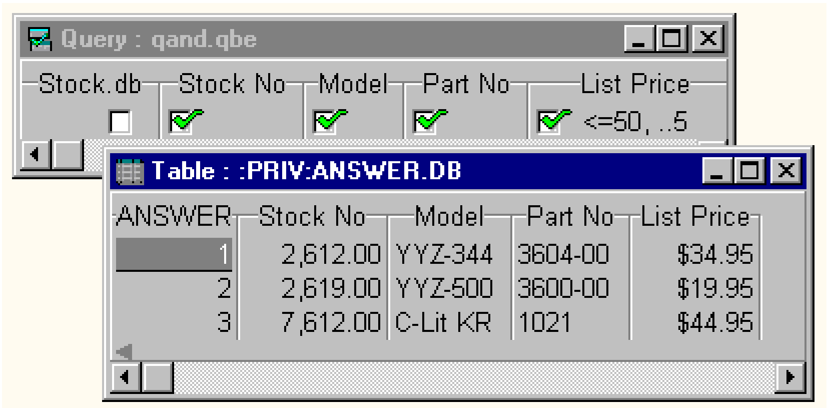
AND conditions in the same field

See also

Use a comma (,) to separate AND conditions in a single field of a query image. Type the entire AND expression on the same line of the field. The comma acts as an AND operator, telling Paradox that both (or all) conditions must be met for a match to occur. Because a value in a single field cannot be two or more values at the same time, the AND conditions you will be specifying in a single field will be any kind except exact match conditions▪for example, two or more types of patterns, or two range conditions.

Example

The following figure shows a query that asks to see list prices from the Stock table that are less than or equal to $50.00 and that end with the number 5.



Query : qand.qbe

Stock.db	Stock No	Model	Part No	List Price
☐	✓	✓	✓	✓ <=50, ..5

Table : :PRIV:ANSWER.DB

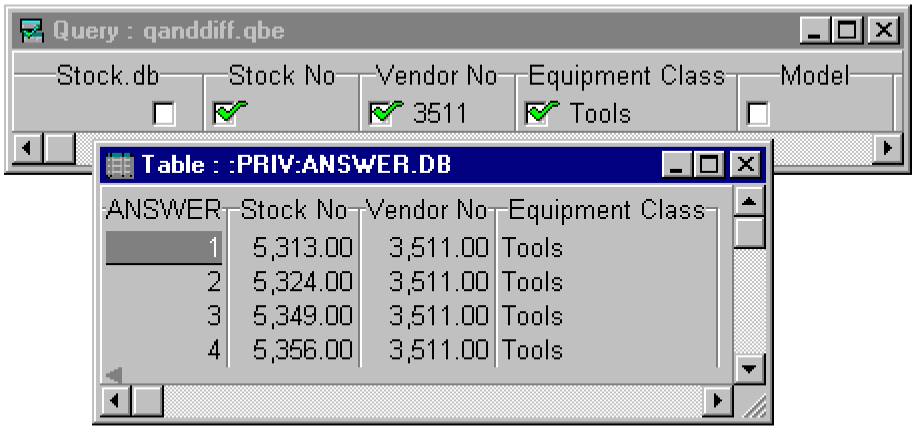
ANSWER	Stock No	Model	Part No	List Price
1	2,612.00	YYZ-344	3604-00	$34.95
2	2,619.00	YYZ-500	3600-00	$19.95
3	7,612.00	C-Lit KR	1021	$44.95

If you have the U.S. number format set, spaces are not necessary between the conditions and the AND (,) operator. If you have the international number format set, a space is necessary on one side of the comma.

▪
AND conditions in different fields

See also

To specify AND conditions in different fields▪that is, conditions that must all be met for a match to occur ▪type the conditions on the same line of the query image, each condition in its respective field.
You can specify exact matches on more than one field in a single query. Type all of the values you want to see▪exactly as they appear in the table
▪in the appropriate fields of the query image. The following figure shows such a query.



Stock.db	Stock No	Vendor No	Equipment Class	Model
☐	✓	✓ 3511	✓ Tools	☐

ANSWER	Stock No	Vendor No	Equipment Class
1	5,313.00	3,511.00	Tools
2	5,324.00	3,511.00	Tools
3	5,349.00	3,511.00	Tools
4	5,356.00	3,511.00	Tools

▪

AND conditions with linked tables

See also

To specify AND conditions with linked tables, type all selection conditions that you want to be met on the same line of each linked query image. As usual, specify AND conditions within a single field by separating all conditions that you want to be met with a comma (,), which is the AND operator.

About OR conditions

See also

You can set logical OR operations in a query. That is, you can retrieve records that meet either of two (or any of several) conditions.

To express an OR condition in a single field, use the OR operator. See OR conditions in the same field for details.

To express an OR condition between different fields, use separate lines of the query image, **not** the OR operator. See OR conditions in different fields for details.

You can create a query that specifies OR conditions in two or more tables. For details, see OR conditions with linked tables (multi-table queries).

Note: You can use the OR operator in all field types, including BLOBs. Whenever you query a memo or formatted memo field, you must use the **..** wildcard operator in addition to any other selection conditions or operators you use.

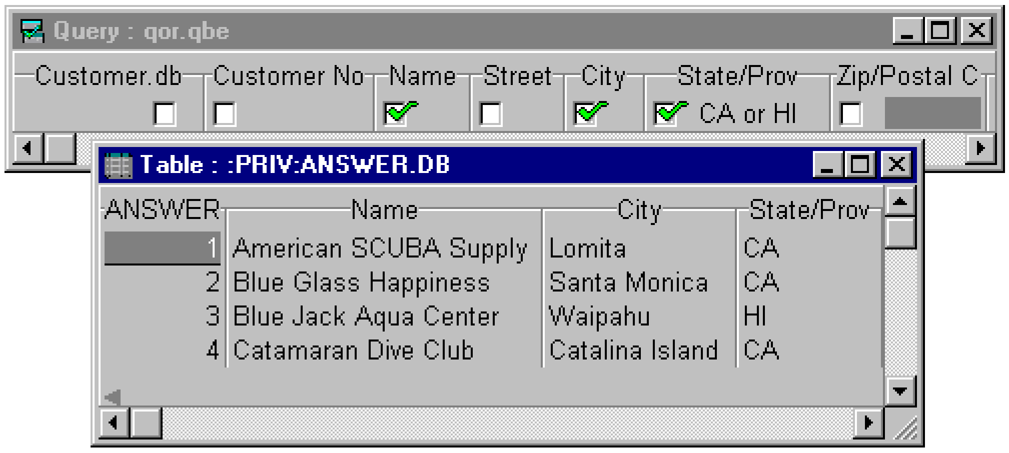
OR conditions in the same field

See also

Specify conditions in a single field on the same line of a query image to tell Paradox you want records that meet any of two or more conditions in that field. Type the operator OR between conditions.

Example

This query retrieves a list of all dive shops from the sample Customer table that are in either California or Hawaii.



Query : qor.qbe

Customer.db	Customer No	Name	Street	City	State/Prov	Zip/Postal C
☐	☐	✓	☐	✓	✓ CA or HI	☐

Table : :PRIV:ANSWER.DB

ANSWER	Name	City	State/Prov
1	American SCUBA Supply	Lomita	CA
2	Blue Glass Happiness	Santa Monica	CA
3	Blue Jack Aqua Center	Waipahu	HI
4	Catamaran Dive Club	Catalina Island	CA

OR conditions in different fields

See also

You can specify OR conditions for different fields of the table you are querying. You perform this kind of OR operation by putting selection conditions on different lines of the query image.

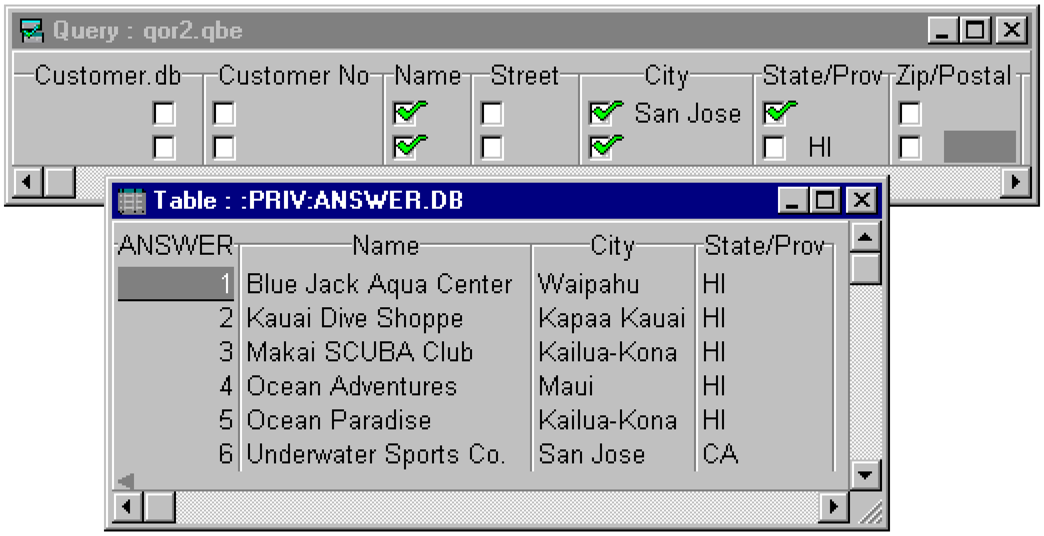
To add additional lines to the query, follow the editing instructions in Working with query images.

To display fields in the Answer table with this kind of query, you must check the check boxes in the same field on each line. For example, if you check the Name field in the first line, you must also check the Name field in all other lines of the query. Otherwise, Paradox displays error messages stating that the query appears to ask two unrelated questions or that one or more query rows do not contribute to the Answer.

Note: You do not use the OR operator for this kind of query. You use the OR operator for OR conditions in the same field.

Example

The following figure shows a query that asks to see records from the Customer table that are in either the city of San Jose, California (this condition is on the first line), or in the state of Hawaii (this condition is on the second line). The same fields are checked in both lines of the query.


Query : qor2.qbe

Customer.db	Customer No	Name	Street	City	State/Prov	Zip/Postal
☐	☐	✓	☐	✓ San Jose	✓	☐
☐	☐	✓	☐	✓	☐ HI	☐

Table : :PRIV:ANSWER.DB

ANSWER	Name	City	State/Prov
1	Blue Jack Aqua Center	Waipahu	HI
2	Kauai Dive Shoppe	Kapaa Kauai	HI
3	Makai SCUBA Club	Kailua-Kona	HI
4	Ocean Adventures	Maui	HI
5	Ocean Paradise	Kailua-Kona	HI
6	Underwater Sports Co.	San Jose	CA

OR conditions with linked tables (multi-table queries)

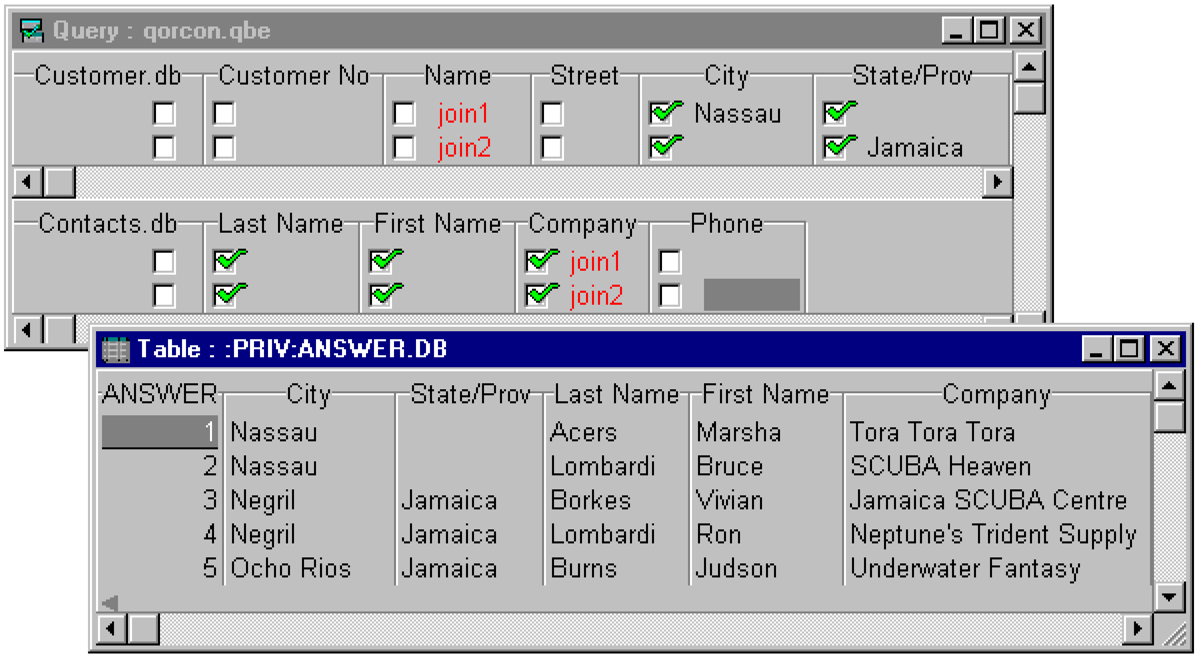
See also

To specify OR conditions with linked tables, type all selection conditions for different fields of a single table, any of which a given record can meet, on separate lines of the table's query image. All query images of linked tables must have the same number of lines and be linked with different example elements for each line of the common field. Specify OR conditions within a single field by separating all conditions, any of which you want to be met, with the OR operator.

Note: You can't use the OR operator on example elements. The condition Qty or Price, where Qty and Price are example elements, returns an error message. This is because an example element stands for all the values in the field. You can't tell Paradox that either Qty or Price can represent all the values in the field.

Example

This query uses the sample Customer and Contacts tables to find the names of the contacts for customers located either in the city of Nassau or in the province of Jamaica. The same example elements are used on corresponding lines of the query images (join1 on the top lines and join2 on the bottom lines).



Query : qorcon.qbe

Customer.db	Customer No	Name	Street	City	State/Prov
☐	☐	☐ join1	☐	✓ Nassau	✓
☐	☐	☐ join2	☐	✓	✓ Jamaica

Contacts.db	Last Name	First Name	Company	Phone
☐	✓	✓	✓ join1	☐
☐	✓	✓	✓ join2	☐

Table : :PRIV:ANSWER.DB

ANSWER	City	State/Prov	Last Name	First Name	Company
1	Nassau		Acers	Marsha	Tora Tora Tora
2	Nassau		Lombardi	Bruce	SCUBA Heaven
3	Negril	Jamaica	Borkes	Vivian	Jamaica SCUBA Centre
4	Negril	Jamaica	Lombardi	Ron	Neptune's Trident Supply
5	Ocho Rios	Jamaica	Burns	Judson	Underwater Fantasy

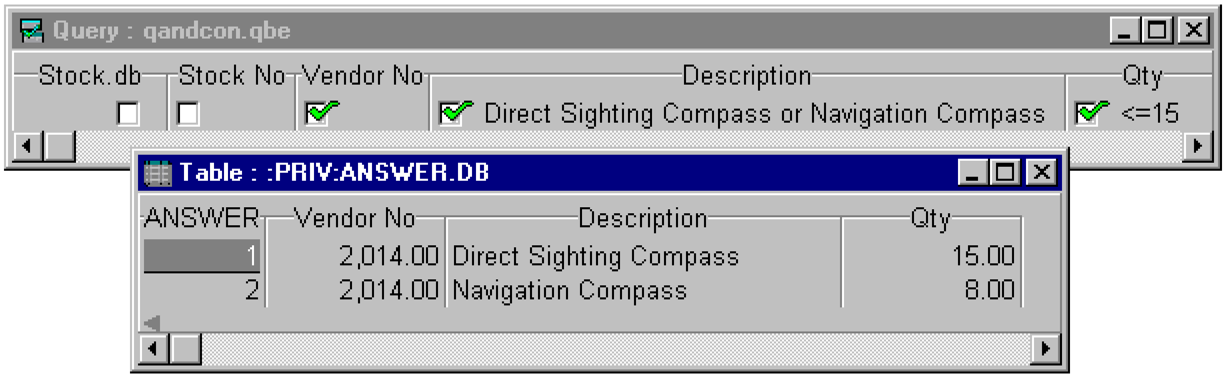
Combining AND and OR conditions

See also

You can combine AND and OR conditions in a single query.

Example

The following example uses the STOCK table to find out if you have 15 or fewer Direct Sighting Compasses and 15 or fewer Navigation Compasses in stock. You also want to see which vendors supply these items.



Stock.db	Stock No	Vendor No	Description	Qty
☐	☐	✓	✓ Direct Sighting Compass or Navigation Compass	✓ <=15

ANSWER	Vendor No	Description	Qty
1	2,014.00	Direct Sighting Compass	15.00
2	2,014.00	Navigation Compass	8.00

Combining two conditions in one field

See also

You can enter two or more selection conditions in the same field of a query image, separating the conditions with commas. The comma acts as an AND operator, telling Paradox that both (or all) of the selection conditions must be met for a match to occur.

Example

Suppose that in the sample Stock table, a list price ending in 5 indicates an item is on sale. You want to see all items that are on sale and cost $50 or less. Here is how you would set up the query:

If you have the U.S. number format set, spaces are not necessary between the conditions and the AND (,) operator. If you have the international number format set, a space is necessary on one side of the comma.

You can also combine AND and OR conditions in a single query.

Notes

To match a value that includes a comma (like Acme, Inc.) you must enclose the value in quotation marks, or Paradox interprets the comma as an AND operator. For example, you would type "Acme, Inc".

Sometimes you use the OR query when you are asking an "and" question. For example, if you want all records in CA and HI, you have to query for CA OR HI because no single record has both values.

▪

About example elements

See also

An example element represents values in the field it is placed in. Example elements are used in two ways in Paradox:

- In single-table queries, you can use example elements with query operators to perform calculations with the values in a particular field. An example element represents each value in turn from that field in the selection condition.
- In multi-table queries, you use example elements to link tables by common fields. The example elements tell Paradox that two fields contain common data even though their field names might differ. Each example element acts as a place marker and means "If a record selected from Table A has a value in this field, link it with all the records from Table B that have the same value in the corresponding field."

You can use example elements in all fields except BLOB fields.

For information on creating example elements, see Creating example elements.

Creating example elements

See also

You can create your own example elements by pressing F5 and typing them. Or you can let Paradox do it for you by clicking the Join Tables button. For details, see

- To create an example element by typing
- To place example elements with the Toolbar

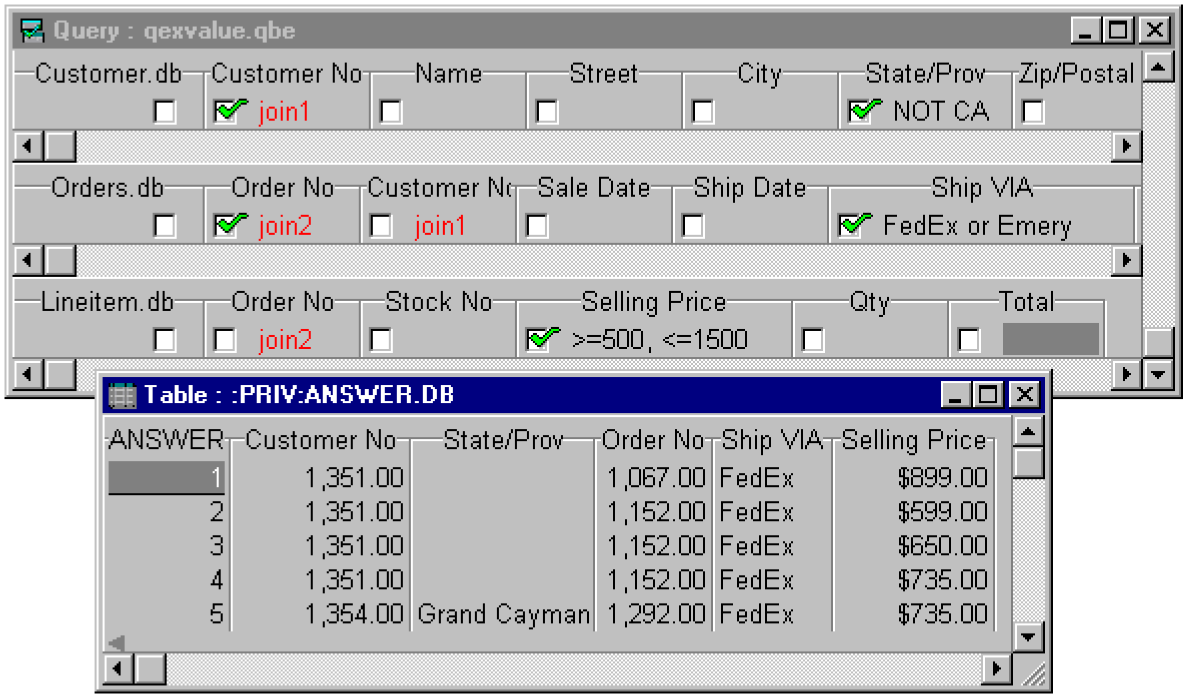
When you create your own example elements, you can use nonsense syllables or names that are meaningful to you. Example elements can contain any alphabetic characters (A-Z, a-z), digits (0-9), or both. They must not contain spaces.

Using an example element in a selection condition

See also

When you use example elements to link tables, you can add as many selection conditions as you want. You can place conditions in any query image. The only requirement of a multi-table query is that all tables in the Query window be linked to each other.

In the following example, you want to know which dive shops outside of California have placed orders for items from $500 to $1,500 in selling price and have had these items shipped via Federal Express or Emery.



Query : qexvalue.qbe

Customer.db	Customer No	Name	Street	City	State/Prov	Zip/Postal
☐	✓ join1	☐	☐	☐	✓ NOT CA	☐

Orders.db	Order No	Customer No	Sale Date	Ship Date	Ship VIA
☐	✓ join2	☐ join1	☐	☐	✓ FedEx or Emery

Lineitem.db	Order No	Stock No	Selling Price	Qty	Total
☐	☐ join2	☐	✓ >=500, <=1500	☐	☐

Table : :PRIV:ANSWER.DB

ANSWER	Customer No	State/Prov	Order No	Ship VIA	Selling Price
1	1,351.00		1,067.00	FedEx	$899.00
2	1,351.00		1,152.00	FedEx	$599.00
3	1,351.00		1,152.00	FedEx	$650.00
4	1,351.00		1,152.00	FedEx	$735.00
5	1,354.00	Grand Cayman	1,292.00	FedEx	$735.00

Note: You cannot use the OR operator with example elements. The statement Qty OR Price, where Qty and Price are example elements, is not a logical question and returns an error message. This is because an example element represents all the values in the field. You cannot tell Paradox that either Qty or Price can represent all the values in the field.

▪

Using an example element to represent a value

See also

You can use an example element in a selection condition when the value you want to use is stored in a table. The example element stands for whatever value Paradox retrieves.

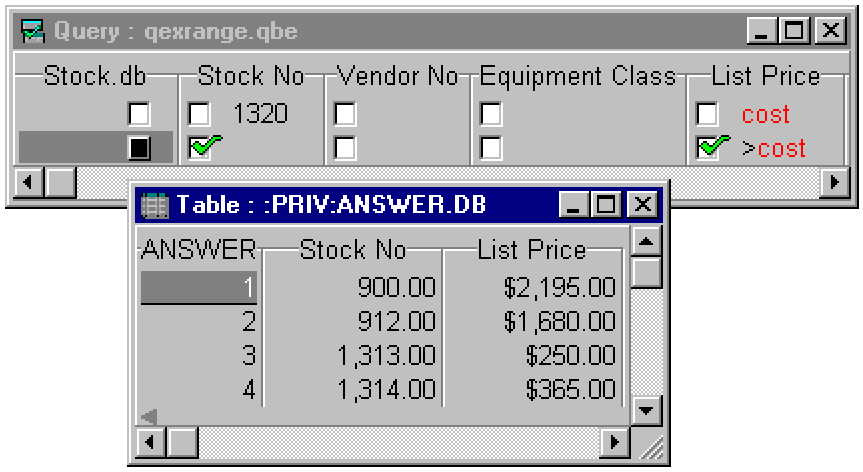
For example, suppose you want to know what dive shops in the Customer table are located in the same city as the VIP Divers Club. Rather than ask what city that is, then ask what cities match it (a two-query process), you can find the value and all matching values in one query, following these steps:

1. Open a Query window and select the Customer table.
2. In the Name field, type `VIP Divers Club`.
3. In the City field, press F5 and type `city` as the example element to represent the city where VIP Divers Club is located.
4. Press the down arrow to create a second line in the query image.
5. On the second line of the query image, check the Customer No, Name, and City fields.
6. In the City field on the second line, press F5 and type `city` again to retrieve all records whose City values are the same as the City value for VIP Divers Club.
7. Run the query.

Using an example element in a range

See also

You can use example elements in queries to retrieve records that match a range of values. For example, suppose you want to list all the stock items whose cost is greater than the cost of item number 1320. You would construct a query like the one below.



Query : qexrange.qbe

Stock.db	Stock No	Vendor No	Equipment Class	List Price
☐	☐ 1320	☐	☐	☐ cost
■	✓	☐	☐	✓ >cost

Table : :PRIV:ANSWER.DB

ANSWER	Stock No	List Price
1	900.00	$2,195.00
2	912.00	$1,680.00
3	1,313.00	$250.00
4	1,314.00	$365.00

The first line of this query retrieves the record that contains Stock No 1320 from the sample Stock table. The cost of item 1320 is represented by the example element cost. The same example element is used in the second line to retrieve all records with a cost greater than that of item 1320. The cost of 1320 is $171.00.

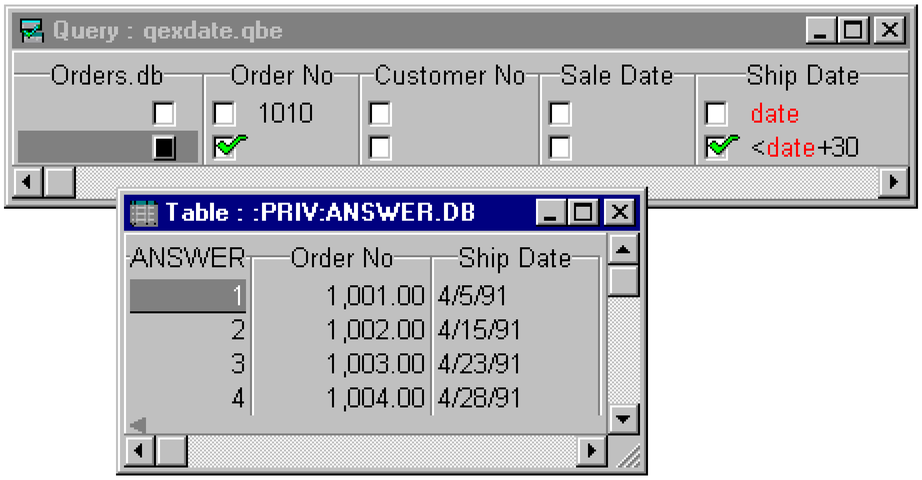
▪

Using an example element in a date condition

See also

You can use an example element in a date expression. For example, suppose you want to list all orders that were shipped less than 30 days after order number 1010 (this includes orders that were shipped before order number 1010). Order 1010 shipped on 5/14/91.

You would construct a query like the one below.


Query : qexdate.qbe

Orders.db	Order No	Customer No	Sale Date	Ship Date
☐	☐ 1010	☐	☐	☐ date
■	✓	☐	☐	✓ <date+30

Table : :PRIV:ANSWER.DB

ANSWER	Order No	Ship Date
1	1,001.00	4/5/91
2	1,002.00	4/15/91
3	1,003.00	4/23/91
4	1,004.00	4/28/91

This query uses

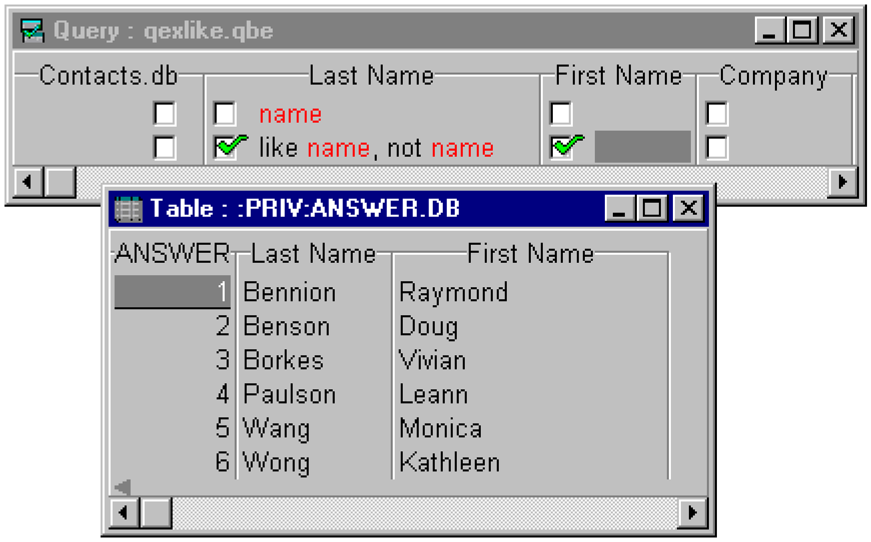
- An example element to represent the shipping date of order number 1010.
- An arithmetic expression to calculate the date 30 days after.
- The < (less than) operator to select the records with shipping dates earlier than the date 30 days after the shipping date of Order No 1010.

Using LIKE or NOT with an example element

See also

You can use example elements with the LIKE and NOT operators.

Suppose you want to find contacts who have been entered more than once in the Contacts table with slightly different last name spellings. You could use LIKE to look for alternative-spelling duplicates of each name, one at a time, or you could use LIKE and NOT with example elements to find all alternative-spelling duplicates at once.



Contacts.db	Last Name	First Name	Company
☐	☐ name	☐	☐
☐	✓ like name, not name	✓	☐

ANSWER	Last Name	First Name
1	Bennion	Raymond
2	Benson	Doug
3	Borkes	Vivian
4	Paulson	Leann
5	Wang	Monica
6	Wong	Kathleen

The statement "like name, not name" specifies last names that are like one another and at the same time not exactly one another-just names that have in common at least half to two-thirds of the same letters. (The space after the comma is not necessary but makes the expression easier to read.)

To create an example element by typing

See also

1. In the Query window, click the field where you want to add an example element.
2. Press F5.
3. Type the example element in the field.

Paradox displays example elements in a different color (usually red), except on monochrome monitors.

You can use any characters that make sense to you. Example elements can contain any letters and numbers.

The following characters cannot be part of an example element:

***   (   )   -   +   /   .**

You cannot put a space in an example element.

When you do one of these things, Paradox assumes you have completed the example element:

- Move to another field, line, or query image.
- Press Spacebar.
- Type one of the characters that can't be part of an example element.

Subsequent characters you type appear in normal text.

If you prefer, you can use the Join Tables button to link two or more tables. This method is usually more efficient than typing. For instructions, see To place example elements with the Toolbar.

To place example elements with the Toolbar

See also

Although you can use the manual method of placing example elements to link two or more tables, the most efficient way to place example elements for this purpose is with the Join Tables ▪ button.

When you click the Join Tables button, the word join appears to the lower right of the pointer and Paradox displays the message `Performing Join` on the status bar. This indicates that you're in join mode. Paradox ends join mode automatically when you place two example elements (by clicking in two fields). You can click the Join Tables button again to leave join mode at any time.

The first pair of example elements Paradox creates is join1, the next is join2, and so on.

The fields you link must be compatible field types (not necessarily the exact same field type▪numeric and money fields are interchangeable) and must contain corresponding data for the link to work.

Example

Suppose you want to see the names of dive shops that have placed orders. The Orders table shows only the Customer No▪not the dive shop's name. The Customer table contains dive shop names. To get the information you want, you must link Customer and Orders on their common Customer No fields.

1. Open a Query window and select the Customer and Orders tables.
2. Check Customer No and Name in the Customer query image, and Order No in the Orders query image.
3. Click the Join Tables button. The join indicator appears to the lower right of the pointer.
4. Click the Customer No field in the Customer query image. Paradox places `join1` in that field.
5. Click the Customer No field in the Orders query image. Paradox places `join1` in that field too.
6. Run the query.

Calculating values with queries: the CALC operator

See also

Uses

The CALC operator performs calculations on the information in your tables. Use CALC to

- Construct and evaluate mathematical expressions
- Combine values from two or more fields
- Combine field values with constants
- Create a new field with a constant value

Capabilities

You can

- Specify selection conditions to define the records to perform calculations on
- Type the CALC expression itself in any field of the query image
- Use CALC with alphanumeric values and with summary operators
- Use values from several tables in a calculation.
- Use example elements in the CALC expression to refer both to values in the same table and to values in other tables.

Rules

When you use CALC in a query, the Answer table generated by that query contains an additional field for the calculated result. This means that

- When you create tables, there is no need to include fields for any data that can be calculated from the values in other fields.
- It does not matter what field of the query image you type the CALC expression in.
- You don't need to check the field in which you enter the CALC expression, because the CALC operator always causes Paradox to create a new field in the Answer table.

  Note: If you *do* check the field in which you enter the CALC operator, this changes the grouping and alters the results.

Paradox gives the new field a name based on the calculation. You can use the AS operator to give the calculated field another name. For instructions, see To rename Answer table fields.

▪

Using CALC with arithmetic operators

See also

You can use CALC in any field of a query image. Following the CALC reserved word, type the expression for the calculation you want to perform.

Expressions can contain

- Constants like 154 or 12/24/91
- Example elements like QTY
- Arithmetic operators like **+ - * / ( )**
- Summary operators like SUM or MAX
- Comparison operators like **= < > <= >=**

Example

Suppose you want to multiply the values of the Quantity (Qty) field of Stock.db by the values in the List Price field to obtain total costs of the stock you have on hand.

1. Choose File|New|Query and select Stock.db.
2. Check the Stock No, Part No, Description, Qty, and List Price fields in the query image.
3. Place an example element in the Qty field (press F5 and type something like `Qty`).
4. Place an example element in the List Price field (press F5 and type something like `Lp`).

   After you've defined the field values you want to work with by placing example elements in the List Price and Qty fields, you can type the CALC expression using these example elements in any field of the query image.
5. In any field, type CALC , then place the example element you're using for the Qty field, then type * , then place the example element you're using for the List Price field. Your query statement should look something like this: `CALC qty * Lp`. (You can choose to type spaces or not; Paradox disregards them.)
6. Run the query.

▪

Using CALC with alphanumeric values

See also

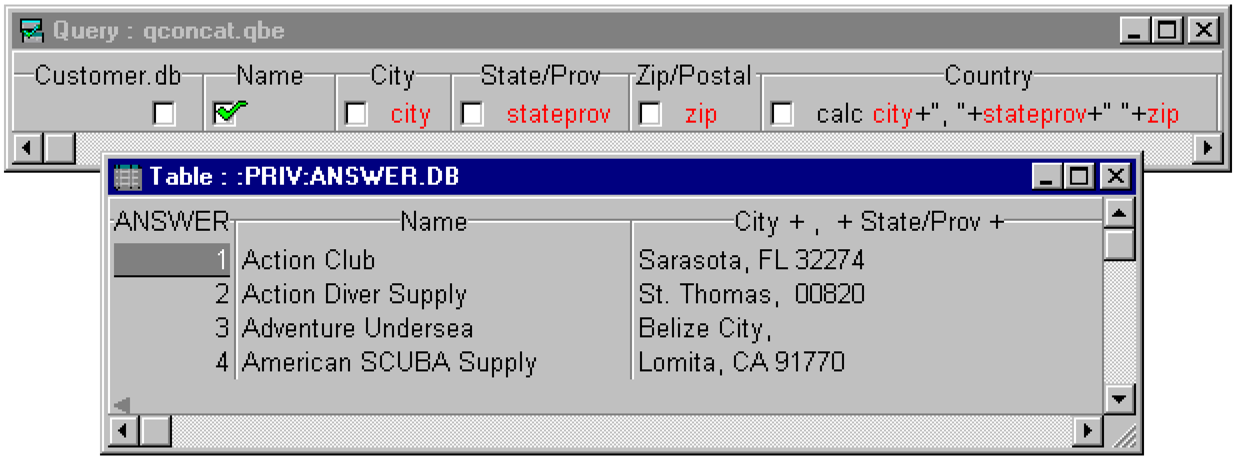
You can combine (concatenate) alphanumeric values and constants by using CALC and the **+** operator. For example,

▪ You can add "Ms. " in front of a list of last names when the value in the Sex field is F.

▪ You can use CALC to combine values from the City, State, and ZIP fields into a single Address field.

Example

Suppose you want to combine the City, State/Prov, and Zip/Postal Code fields of the sample Customer table into one field in an Answer table. Here is how you would set up the query:



Query : qconcat.qbe

Customer.db	Name	City	State/Prov	Zip/Postal	Country
☐	✓	☐ city	☐ stateprov	☐ zip	☐ calc city+", "+stateprov+" "+zip

Table : :PRIV:ANSWER.DB

ANSWER	Name	City + , + State/Prov +
1	Action Club	Sarasota, FL 32274
2	Action Diver Supply	St. Thomas, 00820
3	Adventure Undersea	Belize City,
4	American SCUBA Supply	Lomita, CA 91770

To include the country name for dive shops outside the U.S., you can add the Country field to this concatenation.

Creating a new answer field with a constant value

See also

You can create a new Answer table field that contains a constant value (numeric, date, or alphanumeric) rather than the result of a calculation. When creating a numeric or date constant, type the reserved word CALC, a space, and the constant numeric or date value in any field of the query image. When creating an alphanumeric constant, type CALC, a space, double quotation marks, the alphanumeric constant (with respect for case) and end with double quotation marks.

Paradox names the new field in the Answer table the same name as the constant value. (To name the new field something else, use the AS operator, as described in To rename Answer table fields.) If the new field is alpha, it has as many character spaces as necessary to hold the constant value.

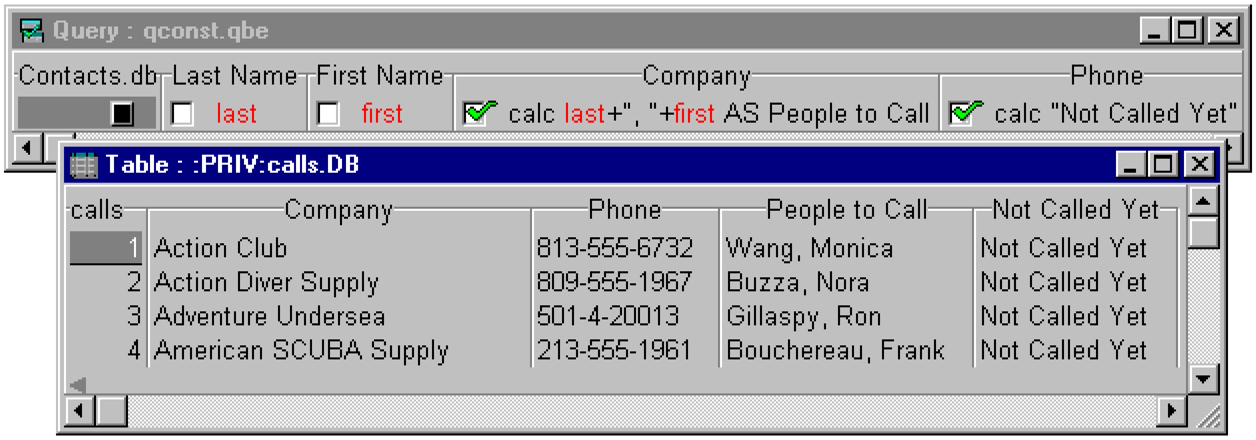
You can create a new blank field by typing CALC BLANK. In this case, you must type the CALC expression in a field of the type that you want the resulting new Answer field to be▪number, short, long integer, money, date, or alpha.

Example

Suppose you need to call all of the dive shop customer contacts in the sample Contacts table to conduct a survey of customer satisfaction. You want a way to keep track of the contacts you have yet to call so that you do not call anyone twice by mistake.

You can create a new table from the Contacts table called Calls. You want to combine the Last Name and First Name fields of Contacts in the Calls table, and you want to create a new field in Calls with the alphanumeric constant "Not called yet." Here is how you would set up the query:

1. Start by giving the Answer table the name Calls. Choose Query|Properties, click the Answer page, then type `Calls` in the Table Name text box and choose OK.
2. Then, set up the following query and run it.



Query : qconst.qbe

Contacts.db	Last Name	First Name	Company	Phone
■	☐ last	☐ first	✓ calc last+", "+first AS People to Call	✓ calc "Not Called Yet"

Table : :PRIV:calls.DB

calls	Company	Phone	People to Call	Not Called Yet
1	Action Club	813-555-6732	Wang, Monica	Not Called Yet
2	Action Diver Supply	809-555-1967	Buzza, Nora	Not Called Yet
3	Adventure Undersea	501-4-20013	Gillaspy, Ron	Not Called Yet
4	American SCUBA Supply	213-555-1961	Bouchereau, Frank	Not Called Yet

Note: You must type the CALC expression and AS operator condition in the same field. If you type them in either the Last Name or First Name fields, which already have example elements in them, you must separate the example element from the CALC expression and AS operator condition with a comma.

Calculating with numeric values from different tables

See also

You can link tables and perform calculations that call on values from different tables in a single query. For examples, see Examples of calculating values with queries, Example 2.

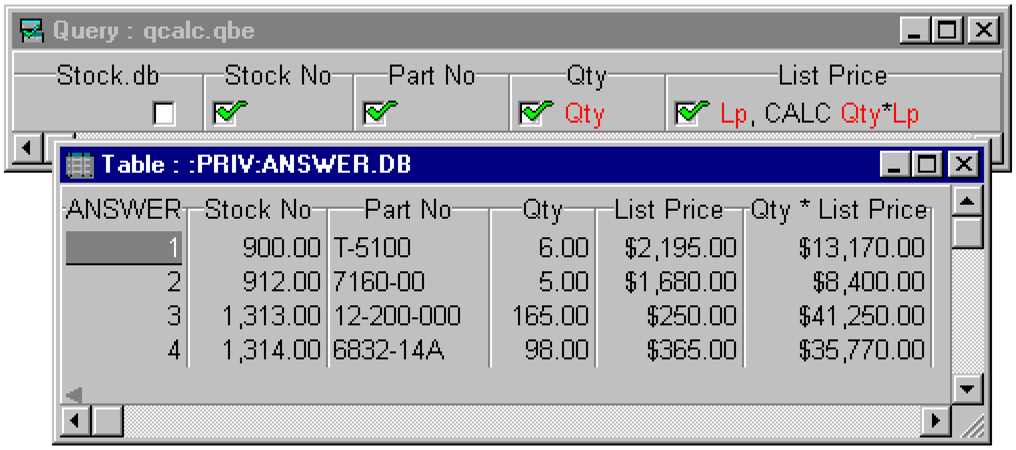
▪

Examples of calculating values with queries

See also

Example 1

Suppose in the sample Stock table you want to multiply the values of the Quantity (Qty) field by the values in the List Price field to obtain total costs of the stock you have on hand. Here is how you would set up the query:



Query : qcalc.qbe

Stock.db	Stock No	Part No	Qty	List Price
☐	✓	✓	✓ Qty	✓ Lp, CALC Qty*Lp

Table : :PRIV:ANSWER.DB

ANSWER	Stock No	Part No	Qty	List Price	Qty * List Price
1	900.00	T-5100	6.00	$2,195.00	$13,170.00
2	912.00	7160-00	5.00	$1,680.00	$8,400.00
3	1,313.00	12-200-000	165.00	$250.00	$41,250.00
4	1,314.00	6832-14A	98.00	$365.00	$35,770.00

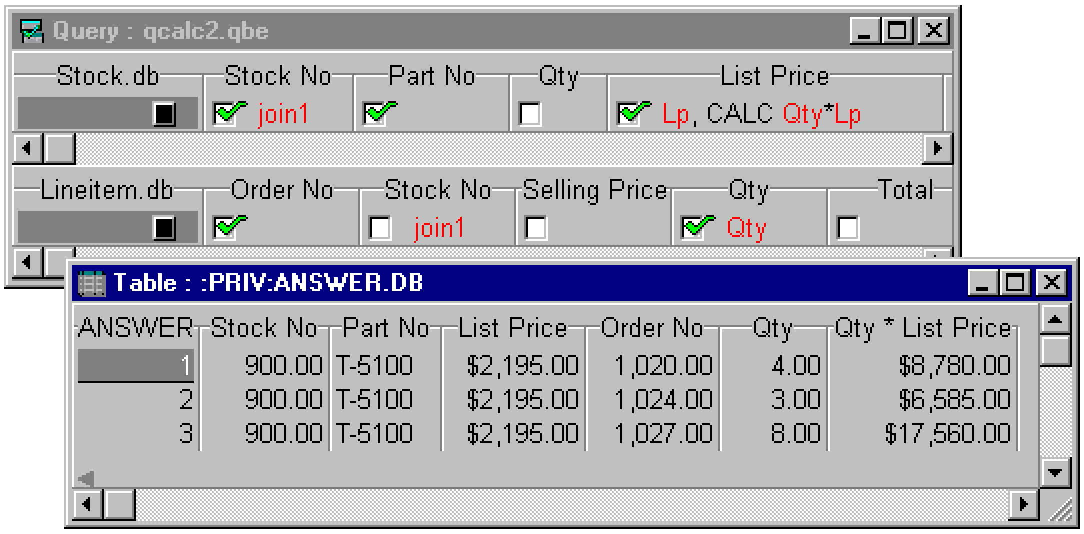
▪ The first occurrence of each example element defines the example. The example elements say, "This variable represents the values in this field."

▪ The second occurrence of each example element uses the values the example elements represent. It says, "Do this with each value in this field."

Example 2

Suppose you want to calculate a total dollar amount of all currently on-order items based on List Price (in STOCK.DB) rather than on Selling Price (in LINEITEM.DB).

To find this information, you need to multiply the list price of all items by the quantity of that item ordered. For steps, see Calculating with numeric values from different tables. The following figure shows the results:



Query : qcalc2.qbe

Stock.db	Stock No	Part No	Qty	List Price
■	✓ join1	✓	☐	✓ Lp, CALC Qty*Lp

Lineitem.db	Order No	Stock No	Selling Price	Qty	Total
■	✓	☐ join1	☐	✓ Qty	☐

Table : :PRIV:ANSWER.DB

ANSWER	Stock No	Part No	List Price	Order No	Qty	Qty * List Price
1	900.00	T-5100	$2,195.00	1,020.00	4.00	$8,780.00
2	900.00	T-5100	$2,195.00	1,024.00	3.00	$6,585.00
3	900.00	T-5100	$2,195.00	1,027.00	8.00	$17,560.00

Create the following example elements by pressing F5 and then typing:

`Qty` in the Qty field of the Lineitem query image

`Price` in the List Price field of the Stock query image

- Then type a comma, and type the expression `CALC Qty * Price` (entering Qty and Price as example elements).
- Use the Join Tables
- button to place example elements in the Stock No fields of both query images.

Querying more than one table

See also

Two or more tables usually contain different information about the same person, place, or thing. To combine this information, you can query more than one table at the same time.

Multi-table queries are similar to single-table queries, except that

- You fill out a separate query image for each table.
- You use example elements to identify common fields among the tables. For details, see Using example elements to link tables.

▪

Using example elements to link tables

See also

When you query more than one table, you must use example elements to link the tables by a common field. These linking fields are fields in each table that contain the same kind of information. For example, Customer and Orders both have a field containing customer identification numbers called Customer No. Because the information in both fields is compatible, you can link these two tables on that field.

Linking fields

The linking fields

- Do not need to have the same field name.
- Must be of compatible types. You cannot, for example, link a number field in one table to an alpha field in another.
- Cannot be memo, formatted memo, graphic, OLE, or binary fields.

You can link up to 24 tables in a single query. For more information, see To add tables to a query and Linking more than two tables.

Example elements

To enter an example element, do one of the following:

- Click the Join Tables
- button. Then click in the appropriate field of each query image. Paradox places example elements that join the tables.
- Select the field, then press F5 and type the example characters in the field.

When you use an example element to link tables, you need to check the field in only one of the tables to display the field.

Existing queries

You can link tables automatically using an existing query. For details, see To create a query based on another query.

▪

Linking more than two tables

See also

Sometimes, three or more tables you are querying have the same field in common. In that case, you use the same example element to link all the tables.

The more usual case occurs when three or more tables have different fields in common: For example, Table 1 and Table 2 have one field in common, Table 2 and Table 3 have a different field in common, and Table 1 and Table 3 have no fields in common.

Use a unique example element for each link. In the case above, you could use the example element abc to link Tables 1 and 2 and use xyz to link Tables 2 and 3.

Note: You can query as many as 24 tables in a single query.

For an overview of linking tables with example elements, see Using example elements to link tables. Or, you can use an existing multi-table query; see Using a multi-table query to link tables.

▪

Using a multi-table query to link tables

See also

If you have already set up a multi-table query, you can open and modify it.

The Query window must be new or empty to start with other multi-table documents.

To use a multi-table query to set up your query, follow the steps in this topic:

- To create a query based on another query

Paradox adds the tables used in the existing query to the Query window, and places example elements to join the tables according to the existing query.

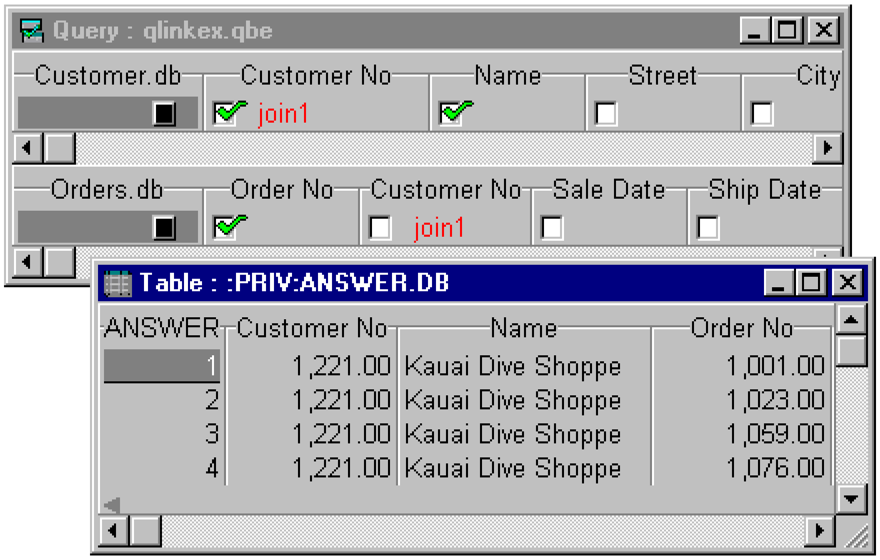
Example of multi-table queries

See also

Example 1

Suppose you want to use the sample tables to see which dive shops have placed orders. The Orders table, however, only shows the Customer ID number and not the dive shop names. The Customer table contains the dive shop names. Therefore, you want to use example elements to link Customer and Orders on their common Customer No fields to retrieve

- Orders information from Orders
- The names of the dive shops that have placed orders from Customer


Query : qlinkex.qbe

Customer.db	Customer No	Name	Street	City
■	✓ join1	✓	☐	☐

Orders.db	Order No	Customer No	Sale Date	Ship Date
■	✓	☐ join1	☐	☐

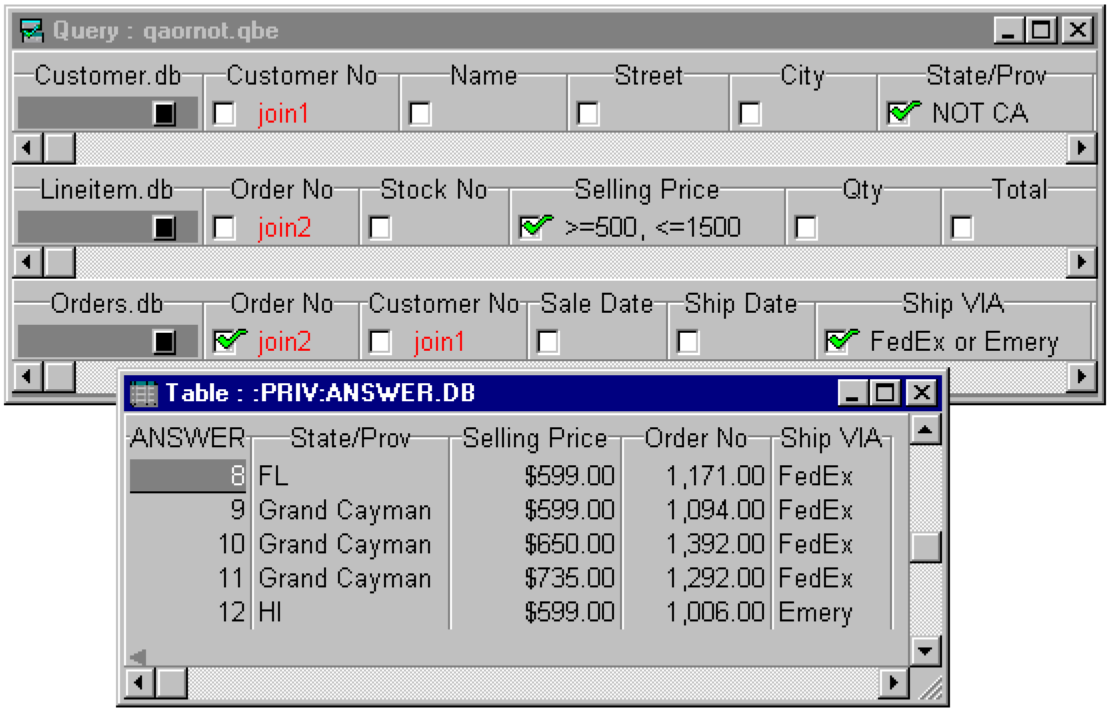
Table : :PRIV:ANSWER.DB

ANSWER	Customer No	Name	Order No
1	1,221.00	Kauai Dive Shoppe	1,001.00
2	1,221.00	Kauai Dive Shoppe	1,023.00
3	1,221.00	Kauai Dive Shoppe	1,059.00
4	1,221.00	Kauai Dive Shoppe	1,076.00

Example 2

Suppose you want to know which dive shops outside of California have placed orders for items from $500 to $1,500 in selling price and have had these items shipped via Federal Express or Emery.

The following figure shows the use of two example elements to link three tables.



Query : qaornot.qbe

Customer.db	Customer No	Name	Street	City	State/Prov
	join1				✓ NOT CA

Lineitem.db	Order No	Stock No	Selling Price	Qty	Total
	join2		✓ >=500, <=1500		

Orders.db	Order No	Customer No	Sale Date	Ship Date	Ship VIA
	✓ join2	join1			✓ FedEx or Emery

Table : :PRIV:ANSWER.DB

ANSWER	State/Prov	Selling Price	Order No	Ship VIA
8	FL	$599.00	1,171.00	FedEx
9	Grand Cayman	$599.00	1,094.00	FedEx
10	Grand Cayman	$650.00	1,392.00	FedEx
11	Grand Cayman	$735.00	1,292.00	FedEx
12	HI	$599.00	1,006.00	Emery

About query results

See also

When run, most queries display an Answer table, which is placed in your private directory. However, if your query uses the INSERT, DELETE, or CHANGETO reserved words, Database Desktop does not display an Answer table. Instead, it changes the data in one of the tables represented in the query and creates an Inserted, Deleted, or Changed table. See About queries that change data.

Note: If you choose the Query|Properties and check the Fast Queries setting on the QBE page, Database Desktop will not create the Inserted, Deleted, and Changed tables.

Live query views

You can edit the Answer table, but any changes you make are not reflected in the original table or tables that you queried. If you want to create an Answer table that does update the original table when you change it, create a live query view instead of an Answer table. See About live query views for more information.

About the Answer table

See also

A Database Desktop query that retrieves data or performs calculations gives you an Answer table. The Answer table is a temporary table that Database Desktop stores in your private directory and replaces each time you perform a query. Database Desktop deletes the Answer table when you exit Database Desktop. If you want to save the Answer table, you must rename it, or save it to a different directory.

To specify a different name for the Answer table or otherwise change it before or after you run the query, see Modifying and renaming the Answer table.

Live query views

You can make a query produce a live query view instead of an Answer table. See About live query views for more information.

Modifying and renaming the Answer table

See also

By default, Database Desktop names the result of a query ANSWER.DB and places it in your private directory. The structure of the Answer table closely reflects the structure of the query example: the leftmost field checked in the first image becomes the leftmost field of the Answer table, and so on.

Modifying the Answer table before running a query

You can change the properties of the Answer table before you run the query. To change the way Database Desktop displays the Answer table, do one of the following:

- Click the Answer Table Properties Toolbar button
- Choose Query|Properties

When the Properties dialog box opens, click the Answer tab, and follow the instructions in the Help for the Answer page (Query properties dialog box).

You can use this dialog box to

- Give the Answer table a different name
- Save Answer to a directory other than your private directory
- Create the Answer table as a Paradox or dBASE table
- Produce a live query view instead of an Answer table

When you finish setting the properties for the table, choose OK to return to the Query window.

Note: For a complete description of available query properties, see About query properties.

Renaming the Answer table after running a query

The Answer table is a temporary table. Every time you run a query, Database Desktop overwrites the Answer table with the new Answer table. To save an Answer table, you must rename it before you run another query. To rename the Answer table, choose Table|Rename or Tools|Utilities|Rename. For information, see About renaming objects.

Caution: If you give Answer the same name as an existing table in the directory to which you save it, Database Desktop overwrites the existing table with no warning. If you save Answer to a directory location that already contains an Answer table, Database Desktop overwrites the existing table with no warning.

When you give Answer a new name, Database Desktop does not treat it as a temporary table, and does not delete it when you change working directories or exit the program.

Changing the Answer table structure and field names

Unless you change it, the structure of the Answer table closely reflects that of the query example: the leftmost field checked in the first query image becomes the leftmost field of the Answer table, the next leftmost field checked in the first query image becomes the second field of the Answer table, and so on through the checked fields of all the query images. You can change the order of fields in the query image or the final table by dragging columns to the desired position or pressing Ctrl+R. Or, choose Query|Properties and rearrange the field order on the Structure page.

If the Answer table contains fields with duplicate field names from two or more tables, Database Desktop names the first field by its exact field name and numbers the duplicates, calling them Name_1, Name_2, and so on.

Database Desktop places new calculated fields at the end of the Answer table and names them according to the calculation. You can rename Answer table fields, including calculated fields. For instructions, see To rename Answer table fields.

To rename Answer table fields

See also

When you check a field in a query image, it is displayed in the Answer table with the same name it had in the original table. To change the field name in the Answer table, use the AS operator:

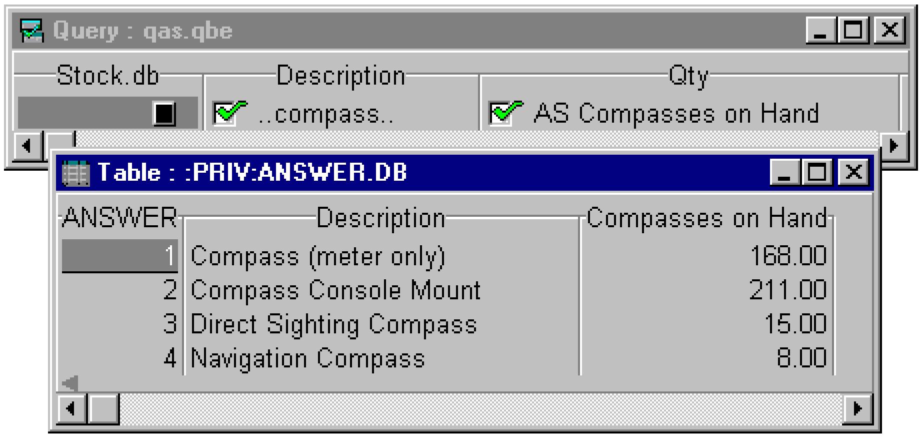
- Type the selection condition, if any, in the field, then type AS, followed by a space, then type the new field name you want.

The AS operator changes field names only in the Answer table. It doesn't change field names in the table(s) you query.

Note: When you use the CALC operator, Database Desktop creates a new field in the Answer table that contains the results of a calculation. Database Desktop automatically places the new calculated field at the end of the Answer table and gives it the name of the calculation. To specify a different name for a calculated field, follow the CALC expression with the AS operator and the new name. For an example, see Creating a new answer field with a constant value.

Example

This example shows how to make the Qty field of the Stock table appear as Compasses on Hand in the Answer table.



Query : qas.qbe

Stock.db	Description	Qty
■	✓ ..compass..	✓ AS Compasses on Hand

Table : :PRIV:ANSWER.DB

ANSWER	Description	Compasses on Hand
1	Compass (meter only)	168.00
2	Compass Console Mount	211.00
3	Direct Sighting Compass	15.00
4	Navigation Compass	8.00

To sort the Answer table

See also

You can sort the Answer table before you run the query. To do this,

1. In the Query window, choose Query|Properties.
2. Click the Sort tab to display the Sort page (Query properties dialog box).
3. Use the Add Field arrow ▪ to move the fields from the Answer Fields list to the Sort Order list. Add the fields in the order you want the Answer table sorted.

To remove a field from the Sort Order list, select it and choose the Remove Field arrow: ▪

To change the order of the fields in the Sort Order list, select a field and use the Change Order arrows ⬆ ⬇ to move it up or down in the list.

Choose OK. Database Desktop will sort the Answer table according to the Sort Order list.

To sort Answer table values in descending order

See also

By default, Database Desktop sorts records in the Answer table in ascending order, based on the values in fields you check, from left to right. That is, it sorts on the leftmost field first, then the next field, and so on.

Here is how sort order applies to the different Paradox field types:

Field type	**Examples of sorted values**	
	from low	**to high**
Number	0	10
Alpha	A, a	Z, z
Date	1/1/91	12/31/91
Money	$1.99	$99.99
Memo	Not sorted	
Graphic	Not applicable	
Time	00:00:01	23:59:59
Logical	False	True
	F	T
	No	Yes
	0	1

Numbers and other nonalphabetic characters are sorted according to the sort order you installed. Alphanumeric "10" sorts before "2" even though it is numerically larger.

To specify that values be sorted in descending order,

▪ Select CheckDescending
▪ from the check box menu for the field you want sorted in descending order.
(To display the check box menu, right-click the check box where you want to enter a checkmark.)

Note: In BLOB fields and in dBASE memo fields, Database Desktop treats ▪ and
▪ as if they were
▪. You cannot use
▪ in BLOB fields or dBASE memo fields.

▪

About live query views

See also

When you create a Database Desktop query that generates an Answer, the Answer table generated by the query does not maintain a relationship with the original table you queried. Edits you make to Answer are not reflected in the original table.

If you prefer, you can create a live query view. When you create a live query view, Database Desktop generates an Answer set that is a limited, direct view into the table you queried. The view is limited by the selection conditions you specify in the query. When you edit the Answer table, you are really editing the table you queried, and using this limited, direct view to see only the data you want from that table.

When Database Desktop creates a live query view, live-field indicators appear next to each live field. Changes you make to data in a live field are also made to the source table.

Multi-table QBE queries can't return live query views. SQL queries on up to three tables can return live query views.

To create a live query view, follow the instructions in To create a live query view.

Tip: You will get better performance on single-table queries if you use a live query view. You can set your query options to default to creating live query views and using the CheckPlus ▪. See Modifying and renaming the Answer table.

Note: Even though you request a live query view on the Answer page of the Query Properties dialog box, Database Desktop might not actually produce a live query view. Here are some common reasons why a live query view might not appear:

- The query didn't use CheckPlus; Check and CheckDescending cause sorting instead.
- You performed an INSERT, DELETE, or CHANGETO query.
- You performed a CALC query.

If a query view can't be live, you still get a query view, not an Answer table, but all the fields are read-only. You'll still see updates other users make to the table (if it's a Paradox table and the Refresh Rate is not 0).

▪

Rules for live query views

See also

Not all queries can return live query views. A live query view must meet the following conditions.

- You can create a live query view only on single-table queries.
- You must use the CheckPlus
- operator. The live query view cannot be sorted, so Check, CheckDescending, and GroupBy checks are not allowed.
- You cannot use the the Sort Answer Table

button or the Sort settings of Query|Properties on a live query view.
- You cannot use calculated fields in a live query view.
- Multi-line OR queries are not allowed.
- The selection conditions you specify in the query must be capable of being expressed as a filter. This means the following query structures are not allowed:
- References to one field in the selection condition of another field
- References to aggregates in the selection condition
- Use of the @ wildcard operator
- Use of the .. wildcard operator before selection conditions. (Use of the .. wildcard operator after a selection condition is allowed, as in the example Canada.., and produces a case-insensitive answer set.)

To create a live query view

See also

1. Choose File|New|Query and select a table to query.
2. In the query image, place CheckPlus ▪ marks in the fields you want to include in the live query view.
3. Choose Query|Properties, click the Answer tab, then choose Live Query View.
4. Run the query.

To edit a live query view

See also

When Database Desktop creates a live query view, you'll see the words Query View and the name of the .QBE file that generated the live query view (if the .QBE file has been saved) in the title bar of the live query view.

To edit the live query view,

- Work with it as if it were any other table you work with in Database Desktop. You must enter Edit mode to make changes. All changes to data in fields from the queried table immediately appear in the original table.

Note: You can use Ctrl+Del to delete a record from the live query view, and this also deletes the record from the table you queried. This deletion (like all Database Desktop deletions) cannot be undone.

To save a live query view

See also

You can save the live query as a .QBE file. From the Query window, choose File|Save or File|Save As and save the live query view as you would any other query. Each time you open and run the query, Database Desktop generates a new live query view of the table's data.

The live query view is a temporary view of the table you queried. You can save this query view as a new table. Choose File|Save to convert the live query view to a standard Answer table and place it in your private directory. Edits you make after saving the live query view are no longer reflected in the table you queried.

If you prefer, you can choose File|Save As to save the live query view as a table you name in a location you specify. This converts the live query view to a standard, permanent table like any other. It no longer maintains a relationship with the table you queried.

About query properties

See also

In the Query window, you can use Query|Properties to specify how you prefer Database Desktop to run your queries and how to display the results. These settings apply to the active query during the current work session.

When you choose Query|Properties, the Properties dialog box appears. It contains the following pages:

Answer	Whether the results appear as an Answer table or live query view, whether the table type is Paradox or dBASE, plus the name and directory of the Answer table.
QBE	Whether queries are to be run locally, remotely, or either; and whether to create auxiliary tables for queries that change data (INSERT, DELETE, CHANGETO queries)
Sort	What Answer table fields are to be included in a sort and in what order
Structure	The order of fields in the Answer table

Query properties are saved with the query.

▪

Handling table updates

See also

When using Database Desktop on a network, multiple users can make changes concurrently to a shared table in a shared data directory. You can choose whether you want your Answer table to reflect changes made to the source table(s) of your query while the query is running.

To change table update handling for the current work session,

- Open the Query menu, then choose the setting you want:
- Choose Restart On Changes to make Database Desktop restart the query when it detects a change to the source table(s).
- Choose Lock Tables to lock all tables in your query, preventing any changes to them while Database Desktop runs the query. Database Desktop releases the locks when it finishes running the query. (If someone else is already using the table(s) you want to lock and query, Database Desktop can't place your locks. You'll see a message informing you that a table is locked.)
- Choose Ignore Changes to allow other users to make changes to the source table(s) while Database Desktop runs your query and to prevent Database Desktop from restarting the query if they do. (This is the default selection.)

▪

Setting Answer table properties

See also

The Answer table properties let you specify the results of a query that can produce an Answer table.

To change Answer table properties for the current query,

1. Choose Query|Properties.
2. Choose the Answer page of the query properties dialog box, then choose the settings you want:

- Choose Answer Table to generate a temporary Answer table; choose Live Query View to generate a live query view
- which you can edit to update the original tables queried
- instead of an Answer table.
- Choose a type for the Answer table: Paradox or dBASE.
- If you want, choose a different name and/or directory for the Answer table so it won't be overwritten the next time you run a query.

Settings made with Query|Properties are saved with the query.

▪

Setting auxiliary table properties

See also

The INSERT, DELETE, and CHANGETO queries generate more than the Answer table. For example, CHANGETO queries create the Changed table and INSERT queries create the Inserted table. Creating these extra tables takes a certain amount of time, and you might not be interested in them.

The auxiliary table preferences and properties let you specify whether to create these tables for queries that change data.

To change auxiliary table options for the current query,

1. Choose Query|Properties.
2. Choose the QBE page of the Query Properties dialog box, then choose the setting you want.

Settings made with Query|Properties are saved with the query.

INSERT, DELETE, and CHANGETO queries are discussed in the following topics:

- About INSERT queries
- About DELETE queries
- About CHANGETO queries

▪
Setting remote query properties

See also

When creating a query that uses data from a remote database server, you can choose whether you want Database Desktop to process the query locally (on your hard drive) or remotely (on the server). Or, you can let Database Desktop decide how the query can be run most efficiently.

To change remote query settings for the current query,

1. Choose Query|Properties.
2. Choose the QBE page of the Query Properties dialog box, then choose the setting you want.

Settings made with Query|Properties are saved with the query.

Whether you create a query on local (Paradox or dBASE) or remote (SQL) data, Database Desktop can translate your QBE statement into valid SQL syntax. This is done automatically when you query remote data. You can view this SQL syntax by choosing Query|Show SQL or clicking the Show SQL button. Database Desktop opens the SQL Editor with the translated SQL syntax in it.

If you prefer writing SQL syntax to creating QBE statements, you can use the SQL Editor to write SQL statements to be run against local (Paradox or dBASE) or remote (SQL) tables. The only restriction is that QBE must be able to interpret the SQL syntax correctly.

Setting query sort properties

See also

You can use Query|Properties to determine how an Answer table will sort before you run a query. For details see To sort the Answer table.

Because Sort is a query property, sort information is saved with the query.

Setting query structure properties

See also

You can use Query|Properties to determine the field order of an Answer table before you run a query.

To change field order for the Answer table of the current query,

1. Choose Query|Properties.
2. Choose the Structure page of the query properties dialog box, then use the Change Order arrows to arrange the Answer Fields list in the order you want.

You can use Undo to restore the previous order at any time.

Because Structure is a query property, field order information is saved with the query.

To save query settings

See also

Settings made with Query|Properties are saved with the query when you choose File|Save or Save As.

▪

About advanced queries

See also

Database Desktop can perform a variety of advanced queries on groups and sets of records. You can

- Work with groups of records using summary operators and other analysis tools.
- Define and compare sets of records to show records that are and aren't part of a set.
- Create and use inclusive links to retrieve all the records in a table, whether they match a selection condition or not.

About querying groups of records

See also

You can use Database Desktop to answer questions about groups of records taken together. You can

- Select records based on characteristics of a group, such as items that appear in two or more orders
- Calculate statistics on groups of records, such as the average invoice total of orders placed in each state
- Compare characteristics of a group with other records, such as which customers have placed more orders than any Hawaii customer

These questions all consider more than one record at a time. No individual record can answer them▪you have to look at the group of records together.

You can use the summary operators to answer these and other questions about groups of records.

▪

About summary operators

See also Examples

A summary operator performs an operation on a group of records that you define by checking a field or fields. You specify which records to group with selection conditions. Database Desktop has five summary operators:

AVERAGE	Averages the values in a group
COUNT	Counts the number of values in a group
MAX	Finds the maximum value of a group
MIN	Finds the minimum value of a group
SUM	Totals the values in a group

As with Database Desktop's other reserved word operators, the case (uppercase or lowercase) in which you type any of the summary operators or summary operator modifiers does not matter.

Fields

You cannot use summary operators in Paradox BLOB fields or dBASE memo fields. In addition, AVERAGE and SUM cannot be used in alpha, date, time, or timestamp fields. See these topics for information on field types that allow summary operators:

- Paradox field types allowing summary operators
- dBASE field types allowing summary operators

Summary modifiers

Summary modifiers let you specify whether to use all values in a group or only unique values. For details, see:

- Using summary operator modifiers

Defining groups

You can use summary operators and checkmarks to define groups of data. For more information, see:

- Selecting records based on group definitions

Paradox field types allowing summary operators

See also

Operator	A	N	$	S	I	#	D	T	@	M	F	G	O	L	+	B	Y
AVERAGE		Y	Y	Y	Y	Y	Y	Y	Y						Y		
COUNT	Y	Y	Y	Y	Y	Y	Y	Y	Y					Y	Y		Y
MAX	Y	Y	Y	Y	Y	Y	Y	Y	Y					Y	Y		Y
MIN	Y	Y	Y	Y	Y	Y	Y	Y	Y					Y	Y		Y
SUM		Y	Y	Y	Y	Y									Y		
ALL*	Y	Y	Y	Y	Y	Y	Y	Y	Y					Y	Y		Y
UNIQUE*	Y	Y	Y	Y	Y	Y	Y	Y	Y					Y	Y		Y

* By default, SUM and AVERAGE operate on all values in a field, while COUNT, MAX, and MIN operate only on unique values. You can override these default groupings by adding the word ALL or UNIQUE to a CALC statement.

dBASE field types allowing summary operators

See also

Operator	C	F	N	D	L	M	O	B
AVERAGE		Y	Y					
COUNT	Y	Y	Y	Y	Y			
MAX	Y	Y	Y	Y	Y			
MIN	Y	Y	Y	Y	Y			
SUM		Y	Y					
ALL*	Y	Y	Y	Y	Y			
UNIQUE*	Y	Y	Y	Y	Y			

* By default, SUM and AVERAGE operate on all values in a field, while COUNT, MAX, and MIN operate only on unique values. You can override these default groupings by adding the word ALL or UNIQUE to a CALC statement.

Using summary operator modifiers

See also

All of the summary operators except COUNT perform their operation on all of the values in a group by default. COUNT counts only unique values in a group by default. To change the default behavior, apply one of the summary operator modifiers:

ALL	Considers all values in a group, including duplicates. You must use ALL with COUNT, in the format COUNT ALL, to make COUNT count all values in a group, including duplicates.
UNIQUE	Considers only unique values in a group. You must use UNIQUE with all summary operators except COUNT to make them perform their operation on unique values in a group instead of on all values.

▪

Selecting records based on group definitions

See also

Use summary operators and checkmarks to define groups of data. Checkmarks (Check, CheckPlus, and CheckDescending) that appear on the same line as a summary operator serve two functions:

- They divide the records into groups based on the values in the checked field.
- They include the checked field in the Answer table (their usual function).

The following examples show how to use summary operators and checkmarks to define groups of data:

- Example of using COUNT: selecting records based on a group count
- Example of using SUM: selecting records based on a group sum
- Example of using AVERAGE: selecting records based on a group average
- Example of using MAX and MIN: selecting records based on a group maximum or minimum

Example of using COUNT: selecting records based on a group count

See also

Use the COUNT summary operator to count unique values in each group.

For example, suppose you want to know which countries have three or more dive shop customers.

In a Query window with a blank CUSTOMER.DB query image,

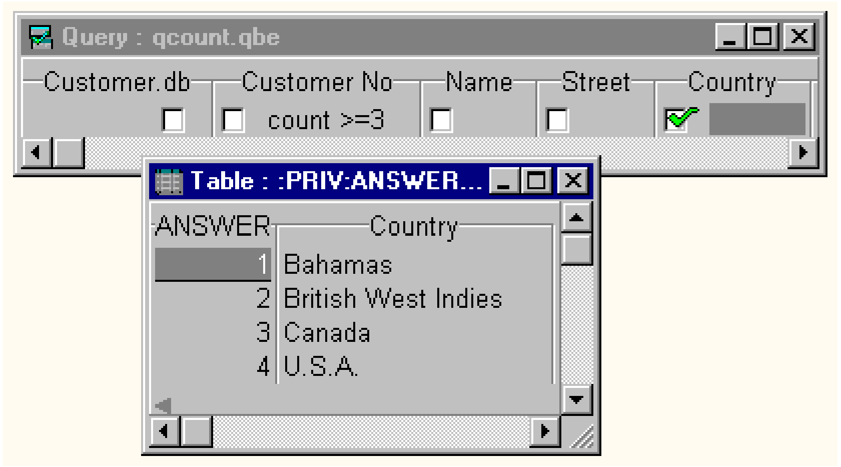
1. Check the Country field.

   The checkmark in the Country field groups the records by country and includes the Country field in the Answer table.

2. Type `count >=3` in the Customer No field.

   The expression count >=3 tells Database Desktop to count all the different customer numbers for each group (country) and to select groups for which the count is three or more.

3. Run the query.



Query : qcount.qbe

Customer.db	Customer No	Name	Street	Country
☐	☐ count >=3	☐	☐	✓

Table : :PRIV:ANSWER...

ANSWER	Country
1	Bahamas
2	British West Indies
3	Canada
4	U.S.A.

Customer is a keyed table, and Customer No is the keyed field, so you know that all customer numbers are unique. The COUNT operator counts unique values by default. If you want to count all values, including duplicates, use COUNT ALL. See Example of counting unique values and Example of counting all values.

▪

Example of using SUM: selecting records based on a group sum

See also

Use the SUM summary operator to sum values within each group in a query.

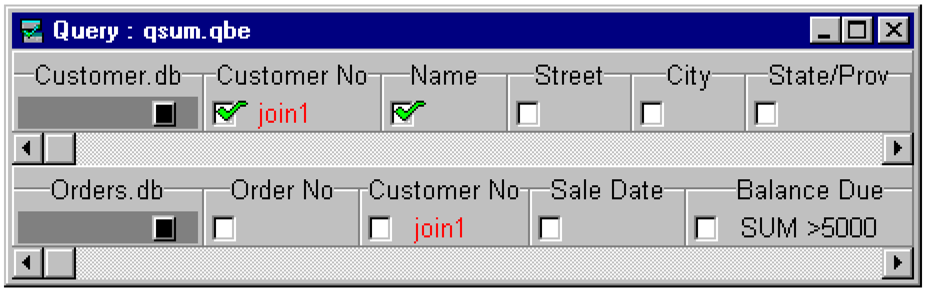
For example, suppose you want to know which customers have placed orders for which they owe $5,000 or more.

In an open Query window with blank CUSTOMER.DB and ORDERS.DB query images,

1. Use the Join Tables ▪ button to place corresponding example elements in the Customer No fields of both query images.
2. Check the Customer No and Name fields of the CUSTOMER.DB query image.
   The Check in Customer No groups the records by customer and includes this field in the Answer table. The Check in Name also groups records by customer and includes this field in the Answer table.

   The Check in Name does not form a different group from the Check in Customer No, because there's a one-to-one correspondence between Customer No and Name; both checks form the same group. See Example of grouping by more than one field.
3. Type `sum >5000` in the Balance Due field of the ORDERS.DB query image.

   The expression sum >5000 sums the balance due for each group (customer) and selects those with balances greater than $5,000.
4. Run the query.


Query : qsum.qbe

Customer.db	Customer No	Name	Street	City	State/Prov
■	✓ join1	✓	☐	☐	☐

Orders.db	Order No	Customer No	Sale Date	Balance Due
■	☐	☐ join1	☐	☐ SUM >5000

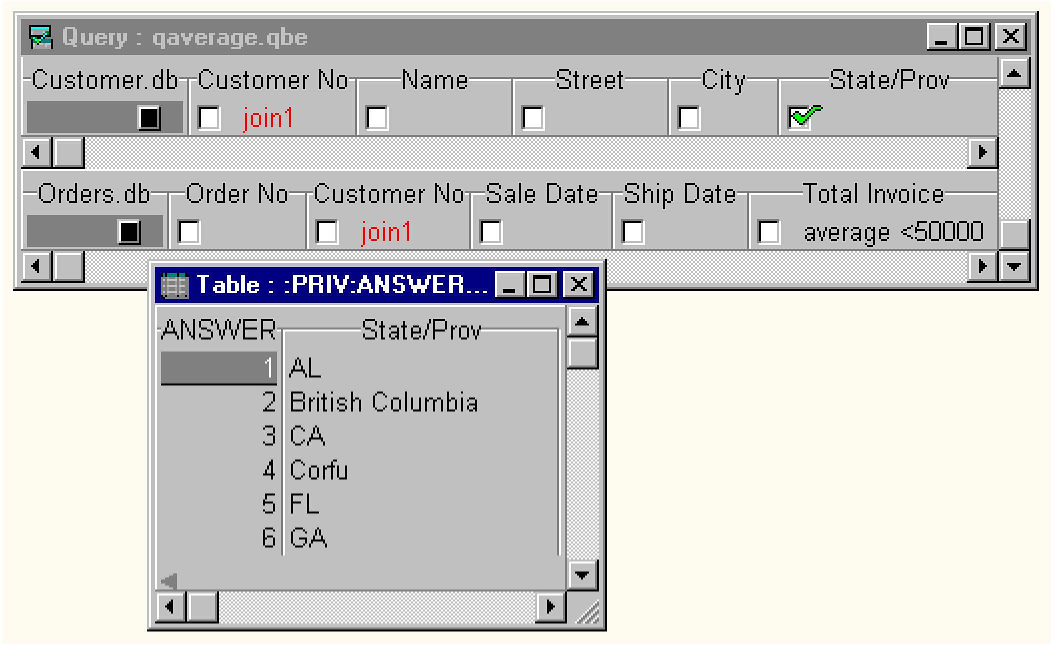
Example of using AVERAGE: selecting records based on a group average

See also

Use the AVERAGE summary operator to average the values in each group in a query.

For example, suppose you want to know the states in which the average invoice total is less than $50,000. In an open Query window with blank CUSTOMER.DB and ORDERS.DB query images,

1. Use the Join Tables button to place example elements in the Customer No fields of both query images.
2. Check in the State/Prov field of the CUSTOMER.DB query image to group the table's records by State/Prov values and include this field in the Answer table.
3. Type `average <50000` in the Total Invoice field of the ORDERS.DB query image.
   The expression average <50000 averages the invoices for each group (state/province) and selects those groups with less than $50,000.
4. Run the query.



Query : qaverage.qbe

Customer.db	Customer No	Name	Street	City	State/Prov
	join1				✓

Orders.db	Order No	Customer No	Sale Date	Ship Date	Total Invoice
		join1			average <50000

Table : :PRIV:ANSWER...

ANSWER	State/Prov
1	AL
2	British Columbia
3	CA
4	Corfu
5	FL
6	GA

▪

Example of using MAX and MIN: selecting records based on a group maximum or minimum

See also

Use the MAX summary operator to find the maximum value in a group. Use the MIN summary operator to find the minimum value in a group.

The following example shows a query using the MAX summary operator. You could do the same query with the MIN summary operator to retrieve the minimum value from the same group.

Suppose you want to know the countries in which the highest total invoice is $200,000 or less. In an open Query window with blank CUSTOMER.DB and ORDERS.DB query images,

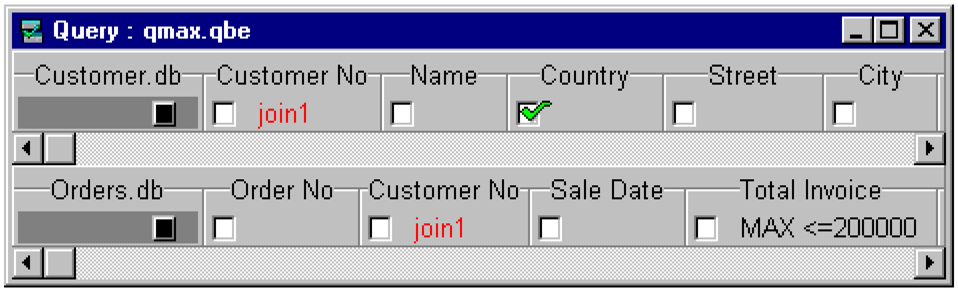
1. Use the Join Tables ▪ button to place example elements in the Customer No fields of both query images.

2. Check the Country field of the CUSTOMER.DB query image to group the table's records by Country values and include this field in the Answer table.

3. Type `max <=200000` in the Total Invoice field of the ORDERS.DB query image.

   The expression max <=200000 finds the total invoice for each group (country) and selects those with $200,000 or less.

4. Run the query.



Customer.db	Customer No	Name	Country	Street	City
■	☐ join1	☐	✓	☐	☐

Orders.db	Order No	Customer No	Sale Date	Total Invoice
■	☐	☐ join1	☐	☐ MAX <=200000

▪

About calculations on groups

See also Examples

In a query, in addition to calculating new fields for each record, you can also calculate statistics (like total and average) for groups of records. For example, you can ask

- How many of each stock item have been ordered?
- What is the total amount of sales for each customer?
- How many customers live in each country or state?
- What are the highest and lowest priced stock items?

Use summary operators with the CALC operator to count, summarize, average, and find the minimum or maximum values in the fields of your tables. To do this, type `CALC` and the appropriate summary operator in the field you want calculated.

Like all CALC queries, those using groups also create a new field in the Answer table. Database Desktop automatically names the new Answer table field according to the group calculation. You can use the AS operator to rename the new field. For instructions, see To rename Answer table fields.

For examples of calculations on groups, see these topics:

- Basic example of a calculation on a group
- Example of grouping by more than one field
- Example of performing a group calculation on the entire table
- Example of displaying summary values without grouping by them
- Example of counting unique values
- Example of counting all values

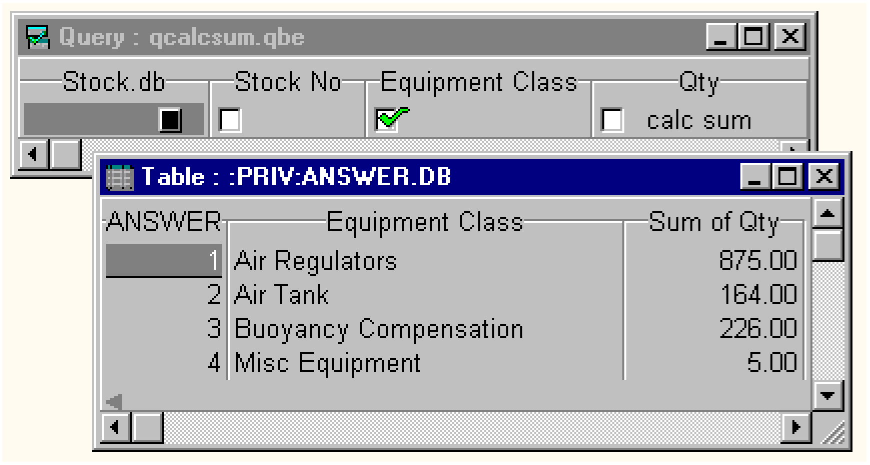
Example of a basic calculation on a group

See also

Suppose you want to know how many of each class of items you have in stock.

In an open Query window with a blank STOCK.DB query image,

1. Check the Equipment Class field to group the table's records by equipment classification and include this field in the Answer table.
2. Type `calc sum` in the Qty field to calculate the sum of the values in this field.
3. Run the query.



Query : qcalcsum.qbe

Stock.db	Stock No	Equipment Class	Qty
■	☐	✓	☐ calc sum

Table : :PRIV:ANSWER.DB

ANSWER	Equipment Class	Sum of Qty
1	Air Regulators	875.00
2	Air Tank	164.00
3	Buoyancy Compensation	226.00
4	Misc Equipment	5.00

Example of grouping by more than one field

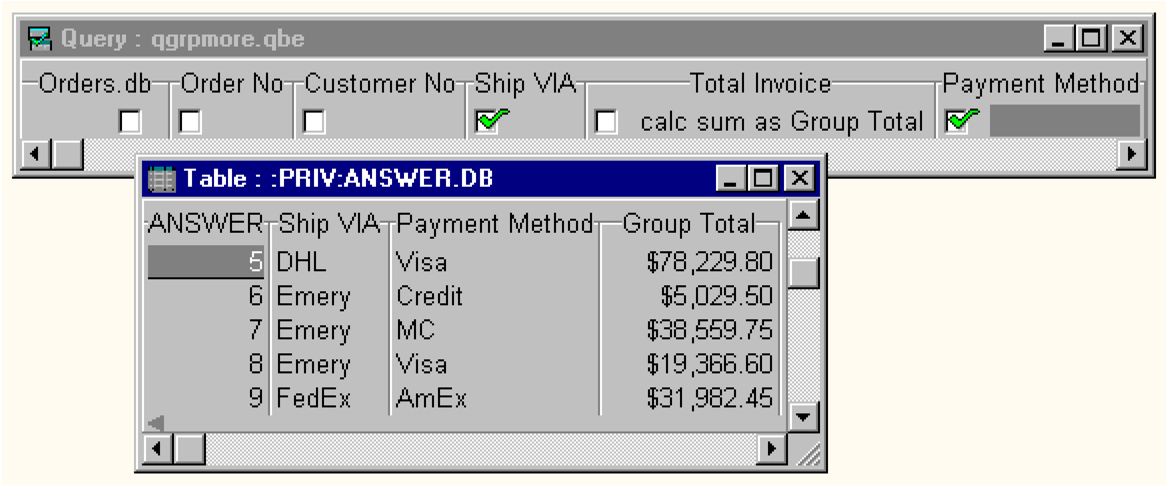
See also

You can group by more than one field in a query. To do this, place checkmarks in all the fields you want to group the table's records by.

Suppose you are interested in a relationship between a payment method and a preferred shipment method. You can group by both the Payment Method and Ship VIA fields of the Orders table:

In an open Query window with a blank ORDERS.DB query image,

1. Check the Payment Method and Ship VIA fields to group the table's records by the values in both fields and include these fields in the Answer table.
2. Type `calc sum as Group Total` in the Total Invoice field to calculate the sum of the values in this field for each group and rename the new calculated field of the Answer table Group Total instead of Sum of Total Invoice. See To rename Answer table fields for instructions.
3. Run the query.



Orders.db	Order No	Customer No	Ship VIA	Total Invoice	Payment Method
☐	☐	☐	✓	☐ calc sum as Group Total	✓

ANSWER	Ship VIA	Payment Method	Group Total
5	DHL	Visa	$78,229.80
6	Emery	Credit	$5,029.50
7	Emery	MC	$38,559.75
8	Emery	Visa	$19,366.60
9	FedEx	AmEx	$31,982.45

In this example, the group of customer numbers and the group of names turned out to be the same group.

SUM operator

Example of using SUM: selecting records based on a group sum demonstrates a query grouping by more than one field. However, in that circumstance, a one-to-one correlation exists between the two fields (Customer No and Name) by which the query is grouping. Thus, the two separate groups (the group of customer numbers and the group of names) are actually the same group.

▪

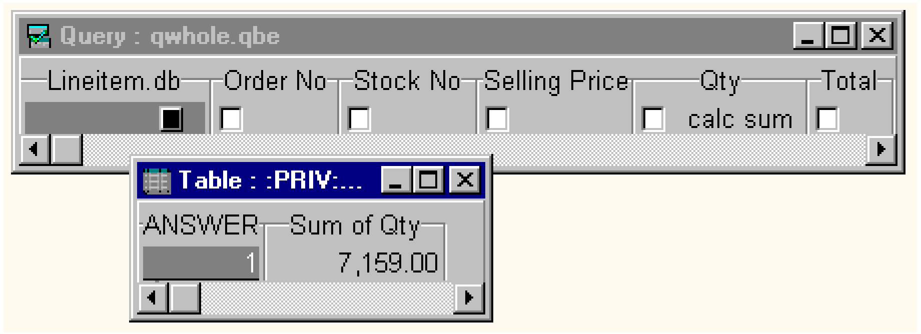
Example of performing a group calculation on the entire table

See also

If you do not check any fields in a query, Database Desktop performs the summary operation or summary calculation on all the records in the table▪the whole table is the group.

Suppose you want to know the total number of items ordered, regardless of who ordered them or what they are or cost. In an open Query window with a blank LINEITEM.DB query image,

1. Type `calc sum` in the Qty field to calculate the total number of items ordered.
2. Run the query.



Lineitem.db	Order No	Stock No	Selling Price	Qty	Total
■	☐	☐	☐	☐ calc sum	☐

ANSWER	Sum of Qty
1	7,159.00

No field is checked, so the group is the whole Lineitem table, and the only field in the Answer table is the Sum of Qty field (the result of the CALC SUM operation).

▪

Example of displaying summary values without grouping by them

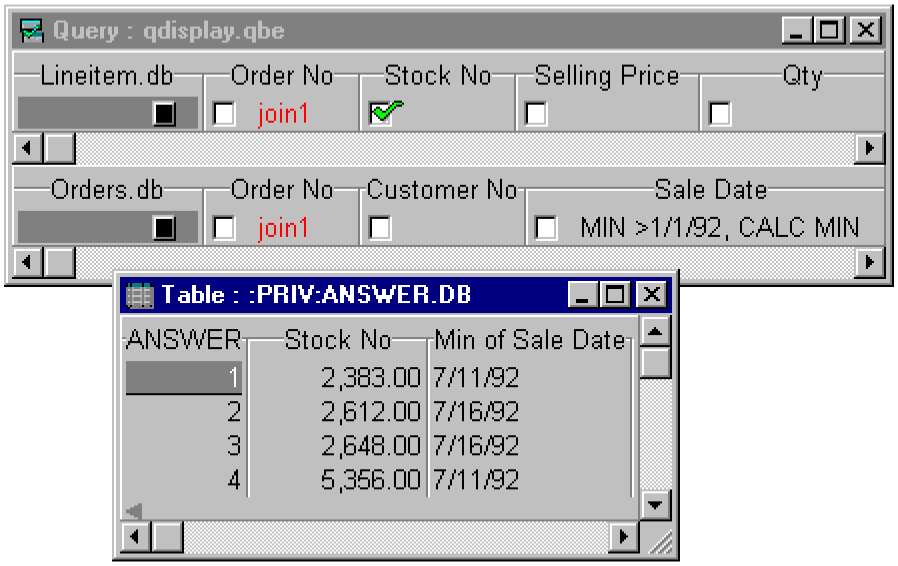
See also

In a query, to display values from a field for which you specify a summary operation without grouping by that field, use the CALC operator in that field with the summary operator you used to specify the operation. The CALC operator causes Database Desktop to create a new calculated field in the Answer table, and this new field will contain the values meeting the summary condition.

Suppose you want to know which items were sold for the first time after January 1, 1989, and you want to display the dates on which these items were ordered.

In an open Query window with blank LINEITEM.DB and ORDERS.DB query images,

1. Use the Join Tables ▪ button to place example elements in the Order No fields of both query images.
2. Check the Stock No field of the LINEITEM.DB query image to group the table's records by Stock No values and include this field in the Answer table.
3. Type `min >1/1/92, calc min` in the Sale Date field of the ORDERS.DB query image.
   Because placing a checkmark in the Sale Date field would cause Database Desktop to attempt to group records by that field, as well as by the LINEITEM.DB Stock No field, you cannot use a checkmark to display the sale dates. Instead, CALC MIN causes Database Desktop to create a new calculated field, Min Of Sale Date, which contains sale dates meeting the summary condition MIN >1/1/92, while preserving the correct grouping.
4. Run the query.


Query : qdisplay.qbe

Lineitem.db	Order No	Stock No	Selling Price	Qty
	join1	✓		

Orders.db	Order No	Customer No	Sale Date
	join1		MIN >1/1/92, CALC MIN

Table : :PRIV:ANSWER.DB

ANSWER	Stock No	Min of Sale Date
1	2,383.00	7/11/92
2	2,612.00	7/16/92
3	2,648.00	7/16/92
4	5,356.00	7/11/92

▪

Example of counting unique values

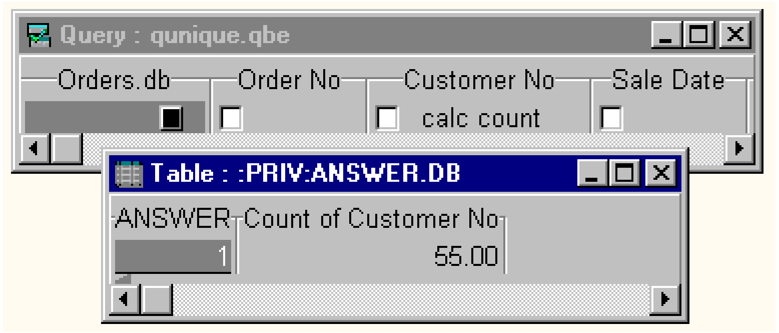
See also

The CALC COUNT query operator counts only unique values. You cannot use COUNT in Paradox BLOB fields and dBASE memo fields. In these field types, CALC COUNT counts all values, even if you specify the UNIQUE operator.

Suppose you want to know how many customers have placed orders with your firm.

In an open Query window with a blank ORDERS.DB query image,

1. Type `calc count` in the Customer No field.
2. Run the query.



Orders.db	Order No	Customer No	Sale Date
■	☐	☐ calc count	☐

ANSWER	Count of Customer No
1	55.00

No field is checked, so the whole Orders table is the group, and the only field in the Answer table is the Count Of Customer No field (the result of the CALC COUNT operation).

▪

Example of counting all values

See also

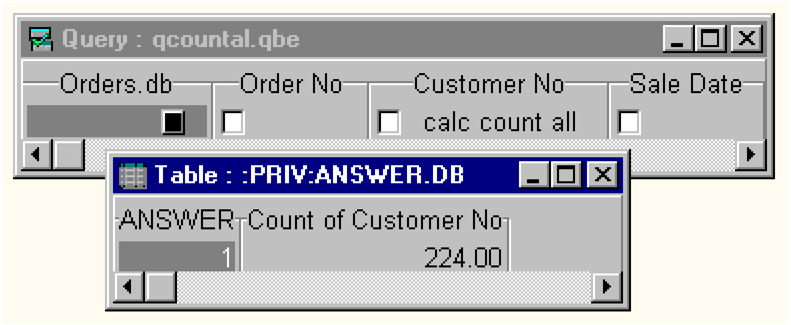
To include duplicates in a query COUNT operation, type `ALL` after the CALC COUNT operator. Database Desktop then counts all values, regardless of duplication.

One way of finding out how many orders have been placed is to use CALC COUNT ALL in the Customer No field of the Orders table. Instead of learning how many unique customers have placed orders, you learn the total number of orders placed.

In an open Query window with blank ORDERS.DB query image,

1. Type `calc count all` in the Customer No field.

2. Run the query.



Query : qcountal.qbe

Orders.db	Order No	Customer No	Sale Date
■	☐	☐ calc count all	☐

Table : :PRIV:ANSWER.DB

ANSWER	Count of Customer No
1	224.00

No field is checked, so the whole Orders table is the group, and the only field in the Answer table is the Count Of Customer No field (the result of the CALC COUNT ALL operation).

▪

Example of using ONLY: selecting records containing only one value

See also

The ONLY operator works the same way as summary operators in that it selects groups whose records all contain the same value and no others. However, ONLY is not a query summary operator since you cannot perform calculations with it.

You can use ONLY in all field types except Paradox BLOB fields and dBASE memo fields.

For example, suppose you want to find customers who have ordered only small instruments. In an open Query window with blank ORDERS.DB, LINEITEM.DB, and STOCK.DB query images,

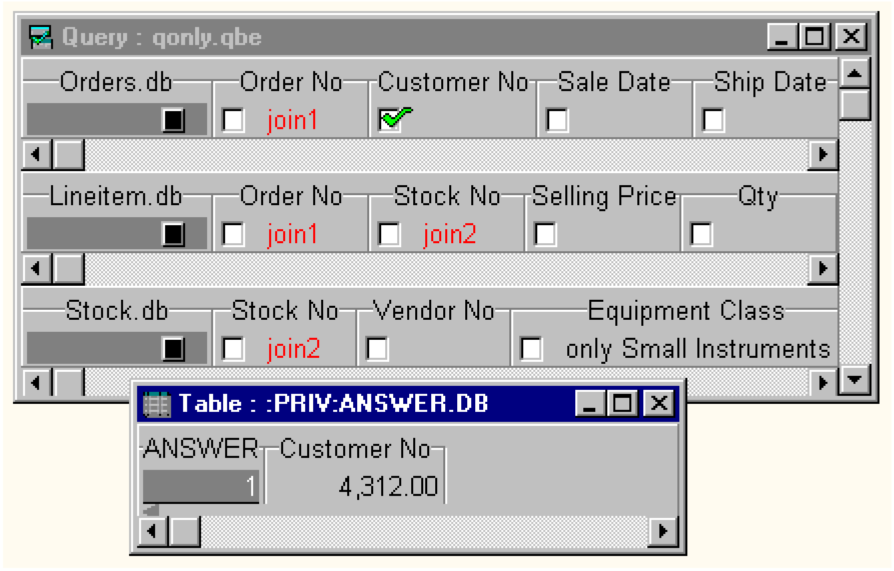
1. Use the Join Tables ▪ button to place example elements in the Order No fields of the ORDERS.DB and LINEITEM.DB query images.

2. Use the Join Tables button to place example elements in the Stock No fields of the LINEITEM.DB and STOCK.DB query images.

3. Check the Customer No field of the ORDERS.DB query image to group the table's records by customer number and include this field in the Answer table.

4. Type `only Small Instruments` in the Equipment Class field of the STOCK.DB query image to select all customers who have ordered small instruments and nothing else.

5. Run the query.


Query : qonly.qbe

Orders.db	Order No	Customer No	Sale Date	Ship Date
■	☐ join1	✓	☐	☐

Lineitem.db	Order No	Stock No	Selling Price	Qty
■	☐ join1	☐ join2	☐	☐

Stock.db	Stock No	Vendor No	Equipment Class
■	☐ join2	☐	☐ only Small Instruments

Table : :PRIV:ANSWER.DB

ANSWER	Customer No
1	4,312.00

▪

About querying sets of records (SET queries)

See also

In general, a set is a collection of objects. In Database Desktop, a set is a specific group of records that you intend to query.

You can use a SET query to answer a question that might otherwise take two or more queries. Use a SET query when you need to ask questions about the characteristics of a group rather than about individual records.

Components of SET queries

Every SET query consists of the following components:

- One or more lines that define a set
- One or more lines, all of which define other records that meet certain comparisons to the set
- Optionally, one or more lines that display related information

SET queries are particularly useful for revealing trends and patterns in data.

Guidelines for querying sets

To query a set, follow these general steps. For examples based on the Stock and Orders tables, click the underlined text.

- Step 1: Define the set
- Step 2: Define groups to compare to the set
- Step 3: Select special groups with set comparisons

Use these set comparison operators to compare the set to other records or groups of records:

ONLY

NO

EVERY

EXACTLY

For more about these steps, see

- Defining a set
- Performing set comparisons

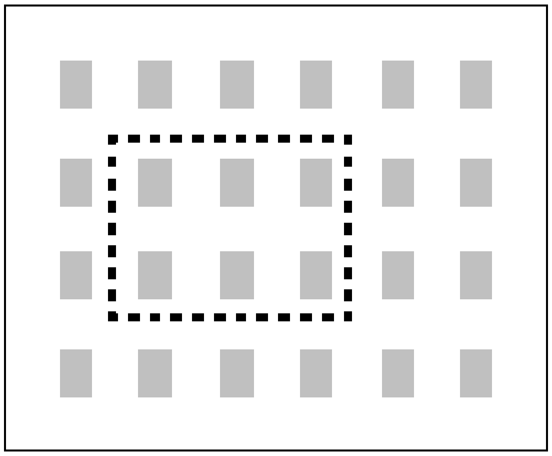
Using the GroupBy check

Sometimes you might want to group records in a query by the values in a specified field without including those values in the Answer table. To do so, choose the GroupBy check ▪ from the menu of checks for the field. You can use the GroupBy check only with SET queries. You cannot use it in BLOB fields. For an example of a query that uses the GroupBy check, see Example of the ONLY set comparison operator.

SET queries, Step 1: Define the set

Each box represents a stock item.

The dotted line surrounds all the stock items that are small instruments, forming a set of all small instrument stock items.

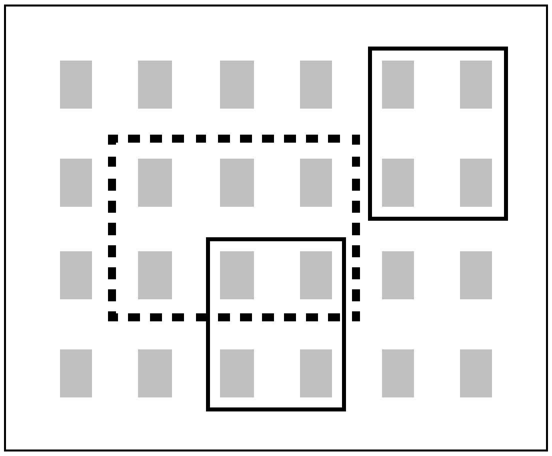


SET queries, Step 2: Define groups to compare to the set

Each box represents a stock item.

The dotted line surrounds all the stock items that are small instruments, forming a set of all small instrument stock items.

Each group in solid-line borders represents an order. The order at the upper right has four line items; none of them is a small instrument. The order at the bottom has four line items; two of them are small instruments.

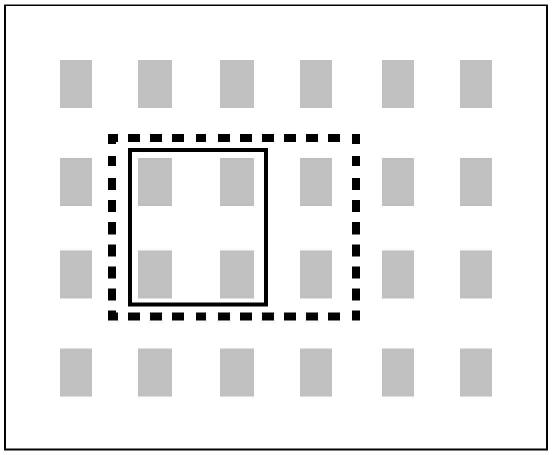


SET queries, Step 3: Select special groups, ONLY

Each box represents a stock item.

The dotted line surrounds all the stock items that are small instruments, forming a set of all small instrument stock items.

Each group in solid-line borders represents an order. This order's line items are *only* for small instruments, not any other type, but they don't include all small instruments.

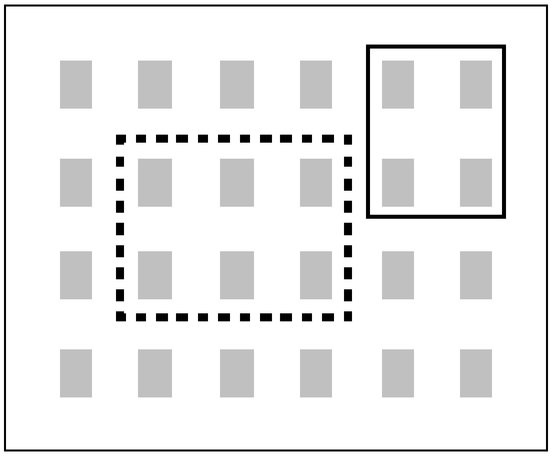


SET queries, Step 3: Select special groups, NO

Each box represents a stock item.

The dotted line surrounds all the stock items that are small instruments, forming a set of all small instrument stock items.

Each group in solid-line borders represents an order. This order has *no* line items for small instruments.

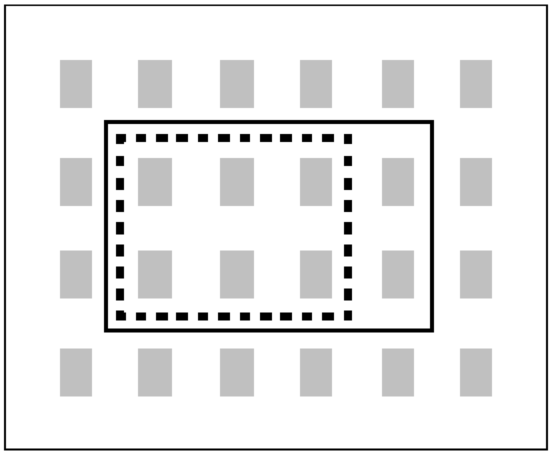


SET queries, Step 3: Select special groups, EVERY

Each box represents a stock item.

The dotted line surrounds all the stock items that are small instruments, forming a set of all small instrument stock items.

Each group in solid-line borders represents an order. This order's line items include *every* small instrument plus other stock items.

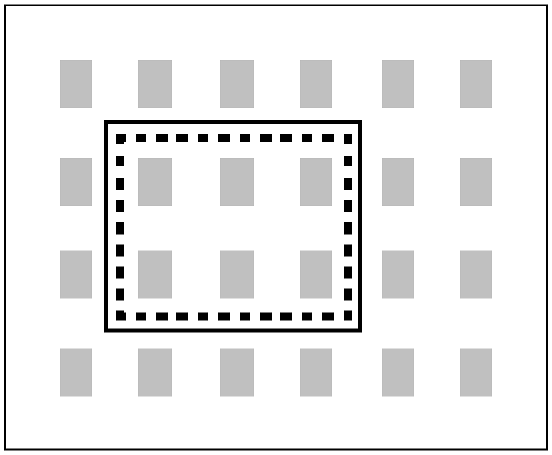


SET queries, Step 3: Select special groups, EXACTLY

Each box represents a stock item.

The dotted line surrounds all the stock items that are small instruments, forming a set of all small instrument stock items.

Each group in solid-line borders represents an order. This order is for *exactly* the items in the small instrument set, no more and no less.



▪

Defining a set

See also

Defining a set of records in a query is very much like selecting the records to be included in the Answer table. A set definition is a query within a query.

1. In the query image(s), enter selection conditions that define the records to be included in the set. If the records are in more than one table, use example elements to link the tables.
2. Choose Set from the menu of query operations in the leftmost field of all query lines that define the set. (To display that menu, right-click in a blank area of the leftmost field.)
3. Where you would ordinarily put checkmarks to define fields, use example elements instead.

   This is necessary because lines that are part of the set definition cannot contain checkmarks or summary operators.

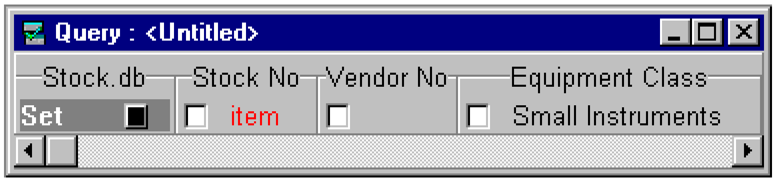
When you go on to compare and retrieve records, you will use these same example elements to link the comparison lines to the set definition.

For an example, see Example of defining a set.

Example of defining a set

See also

This example uses the sample Stock table. The single line of this query defines the set of stock items that are Small Instruments, but it is not a complete query. You still need to compare the set to another factor. (See Example of performing a set comparison for an example of a complete SET query.)


Query : <Untitled>

Stock.db	Stock No	Vendor No	Equipment Class
Set ■	☐ item	☐	☐ Small Instruments

▪
Performing set comparisons

See also Examples

After you have defined a set in a query (see Defining a set), you can compare it to other records. One way of doing this is to compare groups of records to the set.

You can make set comparisons of two different kinds:

- You can compare other groups of records to the set.
- You can use the summary operators to compute the SUM, COUNT, AVERAGE, MIN, and MAX of a set's values, and then compare the results to values in other records.

Database Desktop provides four special set comparison operators to define different sets of records.

Operator	**Field types**	**Description**
ONLY	All*	Displays only records that match members of the set
NO	All*	Displays records that match no members of the set
EVERY	All*	Displays records that match all members of the set
EXACTLY	All*	Displays records that match all members of the set and no others

* You can use set comparison operators in all field types except Paradox BLOB fields and dBASE memo fields.

For example, the records of the Orders table make up the set of all orders placed by customers. From this table, you can formulate subsets of orders for different classes of equipment, such as tools, vehicles, and so on. You can use the four set comparison operators to define sets of orders that

- Are for *only* small instruments
- Have *no* items over $50 in price
- Are for *every* vehicle
- Are for *exactly* all vehicles and no other equipment class item

To form groups of records to compare to the defined set, you use checkmarks. The method is the same as for summary operators.

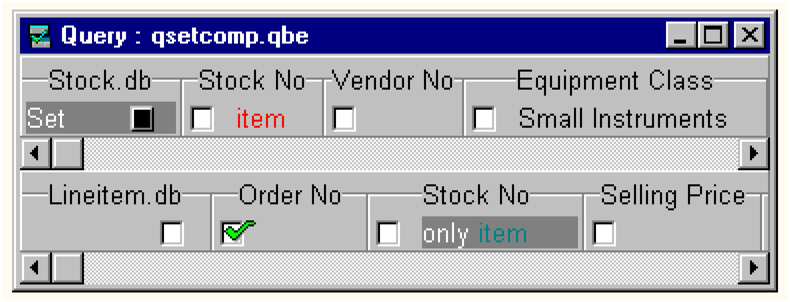
For an example, see Example of performing a set comparison.

▪

Example of performing a set comparison

See also

The Stock query image of this query (created in Example of defining a set) defines the set of stock items that are Small Instruments, but it is not a complete query. To complete the query, you can add the Lineitem table and check the Order No field to display the group of order numbers containing records that meet the conditions of the set. Then type the set comparison operator ONLY, followed by the example element item, in the Stock No field of Lineitem. The query looks like this:



Stock.db	Stock No	Vendor No	Equipment Class
Set	☐ item	☐	☐ Small Instruments

Lineitem.db	Order No	Stock No	Selling Price
☐	✓	☐ only item	☐

The query does several things:

- Defines the set of stock items that are of the equipment class Small Instruments
- Groups the records in the Lineitem table by order number
- Displays the Order No field of Lineitem in the Answer table
- Compares the group of line items of each order number to the set of stock items that are small instruments, selecting those orders whose line items are only small instrument stock items

The Answer table shows those order numbers whose line items are only of the equipment class Small Instruments.

You can use the NO, EVERY, and EXACTLY set comparison operators the same way you use ONLY.

▪

Example of the ONLY set comparison operator

See also

When you use the ONLY set comparison operator in a query, you ask Database Desktop to display only the members of the set you specify.

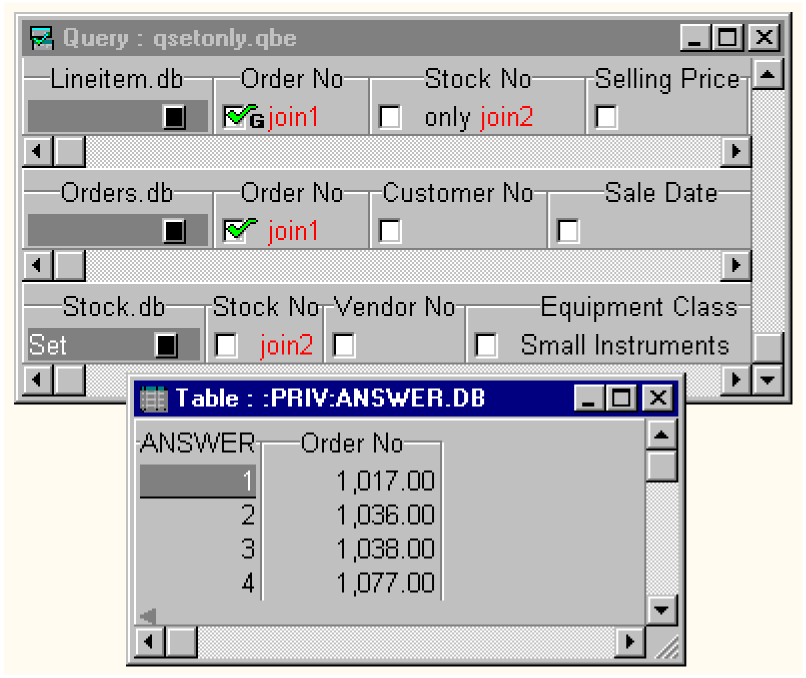
The following example demonstrates another SET query almost like the one in Example of performing a set comparison, except it includes the Orders table. Both queries produce the same Answer table. The difference between the two is where you define the group of order numbers.

Orders is a parent table to Lineitem, and the two tables are linked by their Order No fields, so Lineitem shouldn't have any order numbers that don't exist in Orders. If records with order numbers that don't exist in Orders were present in Lineitem, those records would be orphans▪you'd have line items for nonexistent orders. If those orphan records were in Lineitem, their order numbers would have appeared in the query of Example of performing a set comparison, but not in the query of the following example.

Suppose you want to query the sample tables to see orders placed for the Small Instruments equipment class and no other class of equipment. In an open Query window with blank LINEITEM.DB, ORDERS.DB, and STOCK.DB query images,

1. Use the Join Tables ▪ button to place example elements in the Order No fields of the LINEITEM.DB and ORDERS.DB query images.
2. Use the Join Tables button to place example elements in the Stock No fields of the LINEITEM.DB and STOCK.DB query images.
3. In the leftmost column of the STOCK.DB query image, type `s`, or choose Set from the menu of query operations.
4. In the Equipment Class field of the STOCK.DB query image, type `Small Instruments` to define the set of stock items that are small instruments.
5. Check the Order No field of the ORDERS.DB query image to group by the values of this field and display the field in the Answer table.
6. Place a GroupBy Check in the Order No field of the LINEITEM.DB query image to group by the values of this field but not display this field in the Answer table.
7. Type `only` before the example element in the Stock No field of the LINEITEM.DB query image to cause Database Desktop to select orders placed for only Small Instrument stock numbers.

   (If you were to do this query without the ONLY set operator and without SET in the leftmost column of STOCK.DB, you would get orders placed for Small Instruments in combination with any other equipment class items.)
8. Run the query.



Query : qsetonly.qbe

Lineitem.db	Order No	Stock No	Selling Price
	✓G join1	only join2	

Orders.db	Order No	Customer No	Sale Date
	✓ join1		

Stock.db	Stock No	Vendor No	Equipment Class
Set	join2		Small Instruments

Table : :PRIV:ANSWER.DB

ANSWER	Order No
1	1,017.00
2	1,036.00
3	1,038.00
4	1,077.00

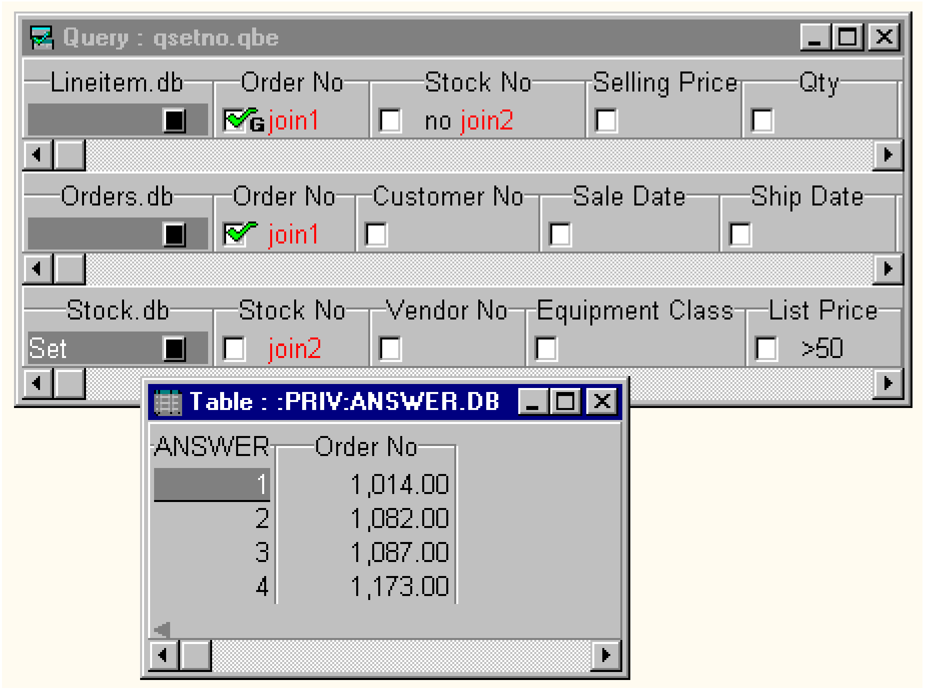
Example of the NO set comparison operator

See also

When you use the NO set comparison operator in a query, you ask Database Desktop to display the groups in which no record matches any record of the set you specify.

For example, suppose you want to find which orders are for no items over $50 in price. The NO SET query asks to see all records outside the set you specify.

1. Duplicate the query shown in Example of the ONLY set comparison operator.
2. Remove the Small Instruments selection condition in the Equipment Class field of the STOCK.DB query image.
3. In the List Price field of the STOCK.DB query image, type `>50` to define the set of stock items over $50 in price.
4. Replace the `only` in front of the example element in the Stock No field of the LINEITEM.DB query image with `no` to retrieve orders placed for items not greater than $50.
5. Run the query.



Query : qsetno.qbe

Lineitem.db	Order No	Stock No	Selling Price	Qty
	✓G join1	no join2		

Orders.db	Order No	Customer No	Sale Date	Ship Date
	✓ join1			

Stock.db	Stock No	Vendor No	Equipment Class	List Price
Set	join2			>50

Table : :PRIV:ANSWER.DB

ANSWER	Order No
1	1,014.00
2	1,082.00
3	1,087.00
4	1,173.00

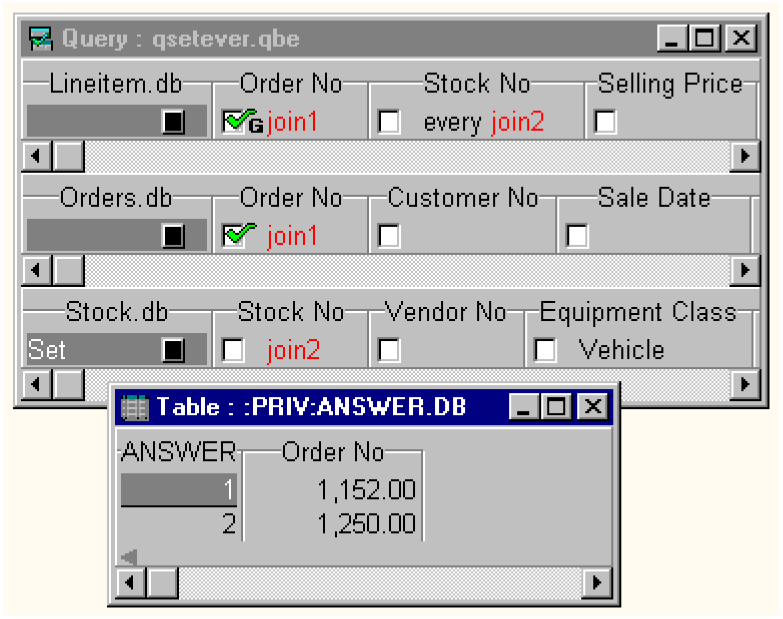
Example of the EVERY set comparison operator

See also

When you use the EVERY set comparison operator in a query, you create a set and ask to see groups containing records that match every item in the set.

For example, suppose you want to see all orders placed for every item in the Vehicle equipment class.

1. Duplicate the query shown in Example of the NO set comparison operator.
2. Remove the `>50` selection condition in the List Price field of the STOCK.DB query image.
3. In the Equipment Class field of the STOCK.DB query image, type `Vehicle` to define the set of stock items that are vehicles.
4. Replace the `no` in front of the example element in the Stock No field of the LINEITEM.DB query image with `every` to cause Database Desktop to select orders placed for all Vehicles.
5. Run the query.



Query : qsetever.qbe

Lineitem.db	Order No	Stock No	Selling Price
	✓G join1	every join2	

Orders.db	Order No	Customer No	Sale Date
	✓ join1		

Stock.db	Stock No	Vendor No	Equipment Class
Set	join2		Vehicle

Table : :PRIV:ANSWER.DB

ANSWER	Order No
1	1,152.00
2	1,250.00

▪

Example of the EXACTLY set comparison operator

See also

When you use the EXACTLY set comparison operator in a query, you create a set and ask to see groups containing records that match every item of the set and only items of the set.

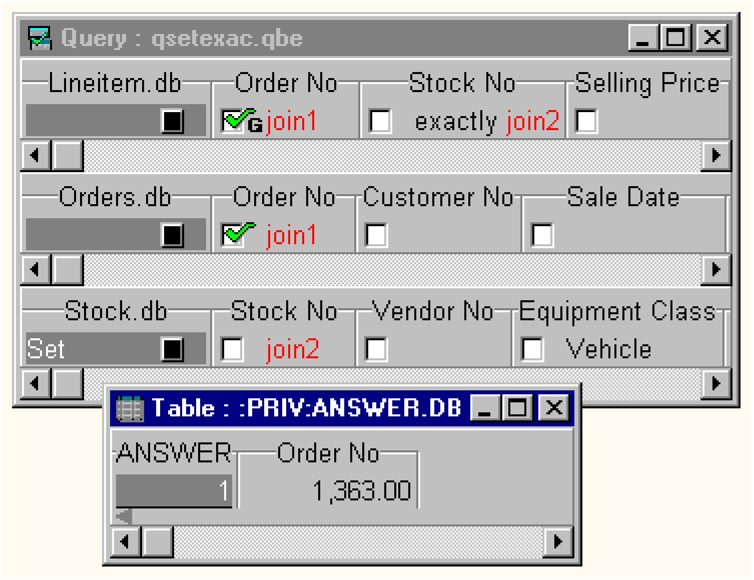
For example, suppose the Sight Diver dive shop calls you and wants to change an order they just placed, order number 1363. This order is for one of the vehicles and an air regulator. Instead of the air regulator, the Sight Diver shop wants the other vehicle. You change this order in the Lineitem table. After you do, you decide to query for other orders that might have been placed for every vehicle and only vehicles:

First, edit the Lineitem table, changing the record for the air regulator, Stock No 1390, in order number 1363 to the following:

Field	**Old value**	**New value**
Stock No	1390	912
Selling Price	170.00	1680.00
Qty	8	1
Total	1360.00	1680.00

Then query for other orders that might have been placed for every vehicle and only vehicles. In the Query window containing the linked query images of Example of the EVERY set comparison operator,

1. Replace the `every` in front of the example element in the Stock No field of the LINEITEM.DB query image with `exactly` to cause Database Desktop to select orders placed for all stock items that are vehicles and no other stock items.
2. Run the query.



Query : qsetexac.qbe

Lineitem.db	Order No	Stock No	Selling Price
	✓G join1	exactly join2	

Orders.db	Order No	Customer No	Sale Date
	✓ join1		

Stock.db	Stock No	Vendor No	Equipment Class
Set	join2		Vehicle

Table : :PRIV:ANSWER.DB

ANSWER	Order No
1	1,363.00

Suppose that when you finished running the query, the Sight Diver shop called you back with another change of mind. They want eight air regulators after all and not the other vehicle.

1. Edit the Lineitem table again to change the record for the 912 vehicle of order number 1363 back to the original 1390 air regulator values; use the Old value column of the table in this example.

This returns the sample data to its original state.

Example of a SET query involving more than one set

See also

SET queries can retrieve records based on comparisons involving more than one set. The comparison in the following example involves two sets.

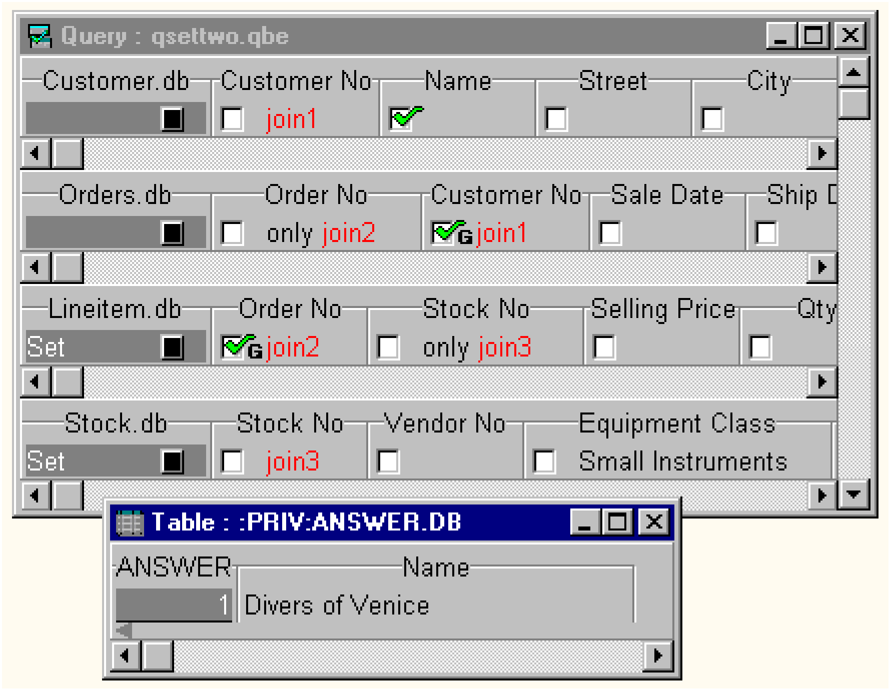
In an open Query window with blank CUSTOMER.DB, ORDERS.DB, LINEITEM.DB, and STOCK.DB query images, in that order,

1. Use the Join Tables button to place example elements in the Customer No fields of CUSTOMER.DB and ORDERS.DB, in the Order No fields of ORDERS.DB and LINEITEM.DB, and in the Stock No fields of LINEITEM.DB and STOCK.DB.

2. Define the set of Small Instruments stock items in the Stock table by choosing Set from the menu of query operations in the leftmost field of STOCK.DB and by typing `Small Instruments` in the Equipment Class field.

3. Retrieve the records from Lineitem that meet the stock item set conditions and only those set conditions by typing `only` in front of the example element in the Stock No field of LINEITEM.DB and by placing a GroupBy Check in the Order No field of LINEITEM.DB.

4. Define the line items that meet the "only Small Instruments" set as a set itself by choosing Set from the menu of query operations in the leftmost field of LINEITEM.DB.
5. Retrieve the records from Orders that meet the line item set conditions and only those set conditions by typing `only` in front of the example element in the Order No field of ORDERS.DB and by placing a GroupBy Check in the Customer No field of ORDERS.DB.
6. Retrieve the customers from Customer who have placed the orders that meet the set conditions and only those set conditions by placing a checkmark in the Name field of CUSTOMER.DB.
7. Run the query.


Query : qsettwo.qbe

Customer.db	Customer No	Name	Street	City
	join1	✓		

Orders.db	Order No	Customer No	Sale Date	Ship D
	only join2	✓G join1		

Lineitem.db	Order No	Stock No	Selling Price	Qty
Set	✓G join2	only join3		

Stock.db	Stock No	Vendor No	Equipment Class
Set	join3		Small Instruments

Table : :PRIV:ANSWER.DB

ANSWER	Name
1	Divers of Venice

▪

Example of summary operators in a SET query

See also

You can compare groups of records to a defined set.

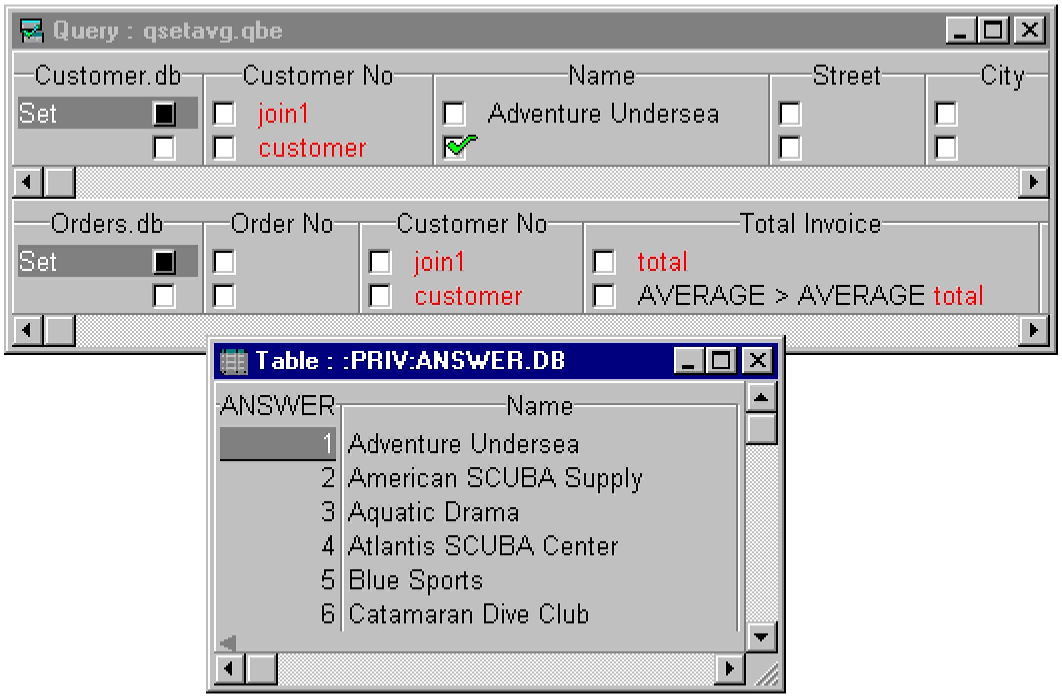
You can also compare groups of records to summary values derived from a set. To do this, you define the set as usual. In the line of the query that selects the records to compare to the set, however, use a summary operator instead of a set comparison operator. You can place the summary operator in an arithmetic expression.

For example, suppose you want to know which dive shops' total invoice averages are more than the total invoice average for a particular dive shop, specifically the Adventure Undersea dive shop. In an open Query window with blank CUSTOMER.DB and ORDERS.DB query images,

1. Use the Join Tables ▪ button to place example elements in the Customer No fields of both query images.

2. Choose Set from the menu of query operations in the leftmost fields of both query images.
3. In the CUSTOMER.DB query image, type `Adventure Undersea` in the Name field to define the set of dive shops that consists of just Adventure Undersea.
4. In the CUSTOMER.DB query image, add a second line. Then, in the second line of the Customer No field, press F5 and type `customer` as an example element representing each customer number value.

5. In the second line of CUSTOMER.DB, check the Name field.
6. In the Total Invoice field of the ORDERS.DB query image, press F5 and type `total` as an example element representing the set of the single invoice total for the Adventure Undersea dive shop.
7. In the ORDERS.DB query image, add a second line. Then, in the second line of the Customer No field, press F5 and type `customer` as an example element representing each customer number value.
8. In the second line of the Total Invoice field in the ORDERS.DB query image, type `average > average` and then press F5 and type `total` to select only those dive shops whose total invoice averages are greater than the total invoice average for Adventure Undersea.
9. Run the query.


Query : qsetavg.qbe

Customer.db	Customer No	Name	Street	City
Set ■	☐ join1	☐ Adventure Undersea	☐	☐
☐	☐ customer	✓	☐	☐

Orders.db	Order No	Customer No	Total Invoice
Set ■	☐	☐ join1	☐ total
☐	☐	☐ customer	☐ AVERAGE > AVERAGE total

Table : :PRIV:ANSWER.DB

ANSWER	Name
1	Adventure Undersea
2	American SCUBA Supply
3	Aquatic Drama
4	Atlantis SCUBA Center
5	Blue Sports
6	Catamaran Dive Club

About inclusive links (! operator)

See also Examples

Queries that use example elements to link tables together usually retrieve all the records in one table that match records in another table. This type of query represents an exclusive link and is sometimes called an inner join.

To produce an Answer table that includes those records that do not match records in the table to which they are linked, use the Database Desktop inclusion operator (**!**). This type of query represents an inclusive link and is sometimes called an outer join.

Add the **!** operator to an example element in a query to retrieve all of the records in that table, whether they match records in another table or not. You can also add selection conditions to define the set of master records included in the answer. You can

- Use multiple inclusion (**!**) operators to retrieve all the records from more than one table
- Use **!** in a query containing an arithmetic expression
- Use both inclusive and exclusive links in the same query

Linking to all records in a table

See also

Sometimes you want all records from one table in a query to appear in the Answer table even if they are not matched in the joined table. This is called an inclusive link and it uses **!** (the inclusion operator).

When you use the inclusion operator in one of two tables, that table is the master table. The other table is the lookup table.

Database Desktop first retrieves all records from the master table. It then looks for and retrieves any matching records in the lookup table. The resulting Answer table contains all records from the master table but only matched records from the lookup table.

You can also use the inclusion operator on both sides of the link. For example, you might want to know which students did not sign up for any courses and which courses have no students.

Note: It is important which table you put the inclusion operator in. That table is the master table and is always processed first. Thus, two queries that are identical, except for the placement of the inclusion operator, can produce significantly different results.

Rules for linking tables

See also

You cannot use an inclusive and an exclusive link in two linked lines.

For any two linked lines in a query, you can use either an inclusive link (**!**) or an exclusive link to associate them, but you cannot use both. This is because an inclusive link includes all the records from the master table, while an exclusive link includes only records whose values in the linked fields match each other. If you use both kinds, Database Desktop has no way to decide which link to process first. The resulting Answer table would be different depending on the sequence.

You will not violate this rule if you remember that you can use **!** with any given example element only once per line and twice per query. In other words, you can use only one type of link to associate any two lines in a query.

You can use an inclusive and an exclusive link in the same query.

You can use both exclusive and inclusive links in the same query as long as they do not both involve the same pair of lines. When you have both types of link in one query, they are processed in order from least to most inclusive:

1. Exclusive links, which do not retrieve records that are not matched by records in another table, are processed first.
2. Asymmetrical inclusive links (with both master and lookup tables), which retrieve all of the records from the master table but only the matched records from the lookup table(s), are processed next.
3. Symmetrical inclusive links (with only master tables), which include all records from both tables, are processed last.

By processing exclusive links before inclusive links, Database Desktop guarantees consistent results to its queries. If you want Database Desktop to process the links in some other order, you must break your question into separate queries.

Selection conditions with inclusive links

See also

You can specify selection conditions for inclusive links just as you can in other queries. This lets you fine-tune either the set of master records or the lookup records to be matched with them.

If you set selection conditions for the master table, the resulting Answer table contains only those records that match the specified selection condition. But it still contains all of those matching records, whether or not they are matched in the lookup table.

▪

Example of linking to all records in a table

See also

Suppose you want to find out if the Customer table contains customers who have never placed an order. If you link Customer and Orders by placing an example element in both Customer No fields, then check the fields you want to see in the Answer table, you will see only those customer records that match one or more records in Orders.

If, however, you add the inclusion (**!**) operator after the example element in the Customer No field of Customer, you will see all customer records, including those of customers who have never placed an order.

In an open Query window with blank CUSTOMER.DB and ORDERS.DB query images,

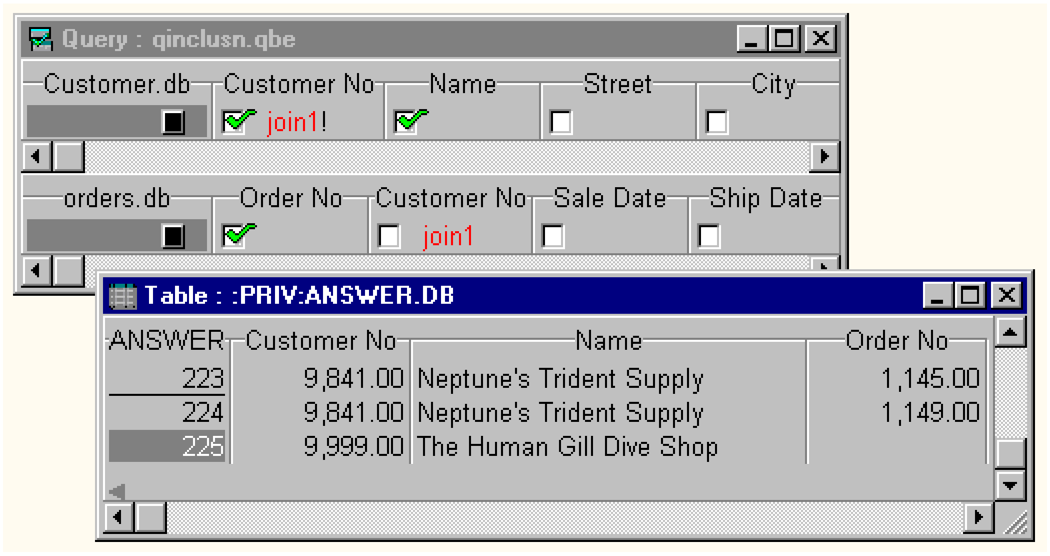
1. Open the Customer table (by choosing File|Open|Table) and add a new record to the end of it (by scrolling to the end and pressing F9 to edit and pressing the down arrow to append a blank record). Add a record for a new dive shop customer, using the following data:

Field Name	**Data**
Customer No	9999
Name	The Human Gill Dive Shop
Street	1225 E. River St.
City	Savannah
State/Prov	GA
Zip/Postal Code	30541
Country	U.S.A.
Phone	404-555-1451
First Contact	5/31/92

2. After adding the new record for The Human Gill Dive Shop to the Customer table, press F9 again to end Edit mode and close the table.
3. Use the Join Tables ▪ button to place example elements in the Customer No fields of both the CUSTOMER.DB and ORDERS.DB query images.

4. Type ! after the example element in the Customer No field of the CUSTOMER.DB query image to include all customers from the Customer table in the Answer table, even if they don't have a matching record in the Orders table.

5. Check the Customer No and Name fields of the CUSTOMER.DB query image.
6. Check the Order No field of the ORDERS.DB query image.
7. Run the query and scroll to the end of the Answer table. Customers without an Order No entry appear there.



Query : qinclusn.qbe

Customer.db	Customer No	Name	Street	City
■	✓ join1!	✓	☐	☐

orders.db	Order No	Customer No	Sale Date	Ship Date
■	✓	☐ join1	☐	☐

Table : :PRIV:ANSWER.DB

ANSWER	Customer No	Name	Order No
223	9,841.00	Neptune's Trident Supply	1,145.00
224	9,841.00	Neptune's Trident Supply	1,149.00
225	9,999.00	The Human Gill Dive Shop	

Note: For a more direct way to accomplish this same task, see Example of retrieving records from one table that are not in another table.

▪

Example of using the Inclusion operator in a query that performs a calculation

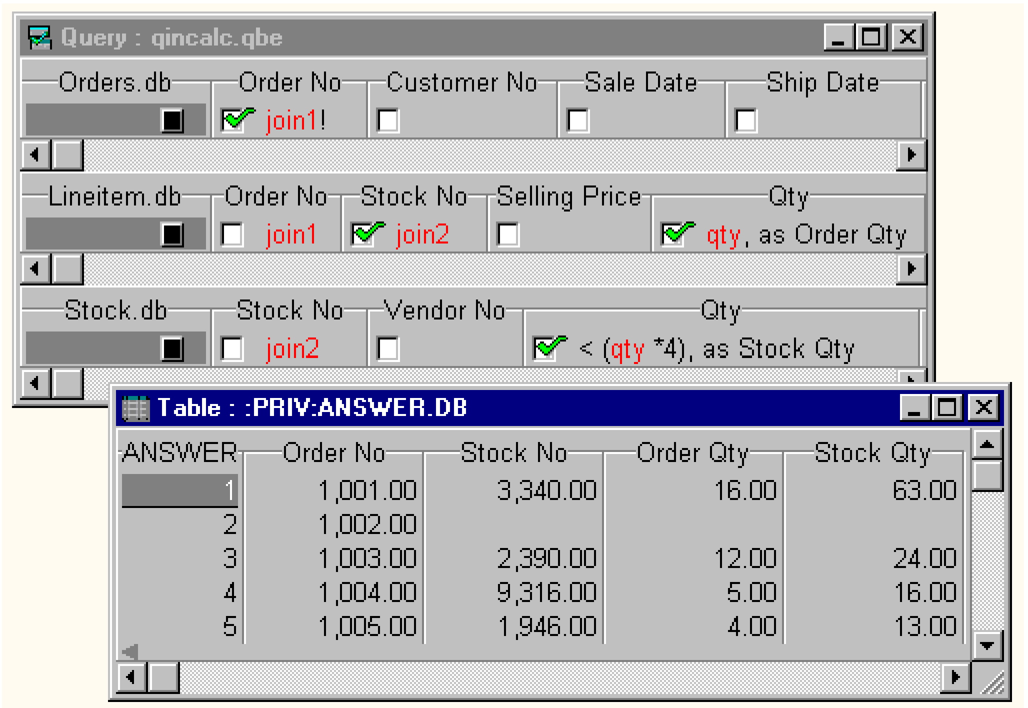
See also

You can use inclusion operators in a query that performs a calculation.

For example, suppose you're concerned about orders you can't fill with your current inventory. More specifically, you want a list of all orders, highlighting those for quantities that exceed one quarter of the quantities in stock.

In an open Query window with blank ORDERS.DB, LINEITEM.DB, and STOCK.DB query images, in that order,

1. Use the Join Tables ▪ button to place example elements in the Order No fields of the ORDERS.DB and LINEITEM.DB query images.
2. Use the Join Tables button to place example elements in the Stock No fields of the LINEITEM.DB and STOCK.DB query images.
3. Type `!` after the example element in the Order No field of the ORDERS.DB query image to see all order numbers. Check the Order No field in the ORDERS.DB query image.
4. Check the Stock No and Qty fields of the LINEITEM.DB query image.
5. In the Qty field of the LINEITEM.DB query image, press F5 and type `qty` as the example element representing all the values, in turn, of the Lineitem table's Qty field.
6. Still in the Qty field of the LINEITEM.DB query image, type `, as Order Qty` after the qty example element.
7. Check the Qty field of the STOCK.DB query image.
8. In the Qty field of the STOCK.DB query image, type `< (`, then press F5 and type `qty` and a space. Then type `* 4)`, as Stock Qty.
9. Run the query.



Orders.db	Order No	Customer No	Sale Date	Ship Date
	✓ join1!			

Lineitem.db	Order No	Stock No	Selling Price	Qty
	join1	✓ join2		✓ qty, as Order Qty

Stock.db	Stock No	Vendor No	Qty
	join2		✓ < (qty *4), as Stock Qty

ANSWER	Order No	Stock No	Order Qty	Stock Qty
1	1,001.00	3,340.00	16.00	63.00
2	1,002.00			
3	1,003.00	2,390.00	12.00	24.00
4	1,004.00	9,316.00	5.00	16.00
5	1,005.00	1,946.00	4.00	13.00

The **!** operator in Orders ensures that the Answer table contains all orders. The qty example element is used in the expression `qty * 4` to multiply each stock item quantity value in the Qty field of the Lineitem table (representing the order quantity of each stock item) by four. The **<** comparison operator then looks for actual stock quantities less than this amount, thus retrieving records of orders that exceed one quarter of the inventory. Records in Answer that contain only an order number are those that do not meet the selection conditions, but are included because the inclusion operator was used.

▪

Example of retrieving records from one table that are not in another table

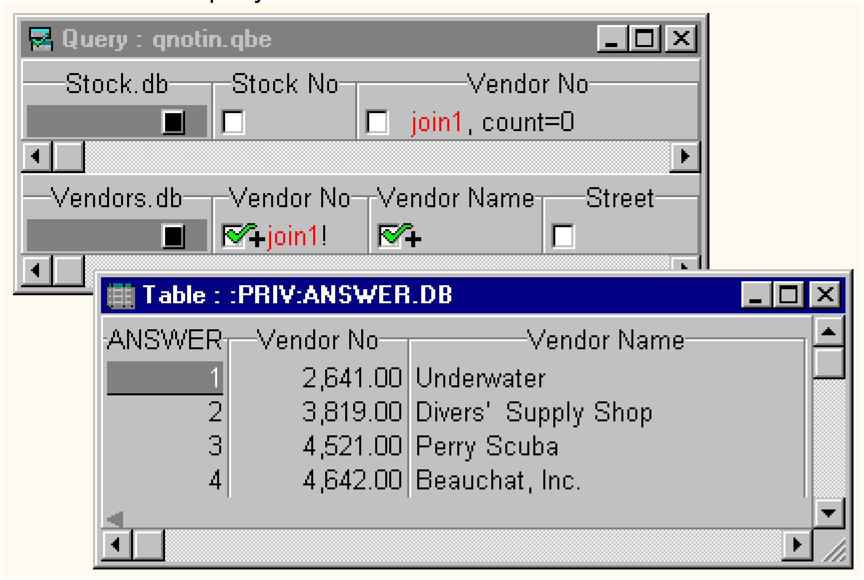
See also

You can use an inclusive link with the COUNT summary operator and CheckPlus to retrieve records from one table that are not in another table.

For example, suppose you want to know which vendors you have in the Vendors table from whom you have yet to buy any stock. That means you want to know which vendors are in the Vendors table that are not in the Stock table.

In an open Query window with blank STOCK.DB and VENDORS.DB query images,

1. Use the Join Tables ▪ button to place example elements in the Vendor No fields of both query images.
2. Type `!` after the example element in the Vendor No field of VENDORS.DB.
3. Place CheckPluses in the Vendor No and Vendor Name fields of VENDORS.DB to retrieve all records, including duplicates.
4. After the example element in the Vendor No field of STOCK.DB, type a comma and a space and then `count = 0`.
5. Run the query.



Stock.db	Stock No	Vendor No
■	☐	☐ join1, count=0

Vendors.db	Vendor No	Vendor Name	Street
■	✓+join1!	✓+	☐

ANSWER	Vendor No	Vendor Name
1	2,641.00	Underwater
2	3,819.00	Divers' Supply Shop
3	4,521.00	Perry Scuba
4	4,642.00	Beauchat, Inc.

▪

Example of using both inclusive and exclusive links in a query

See also

The following example uses the sample tables to demonstrate a complicated query containing both inclusive and exclusive links.

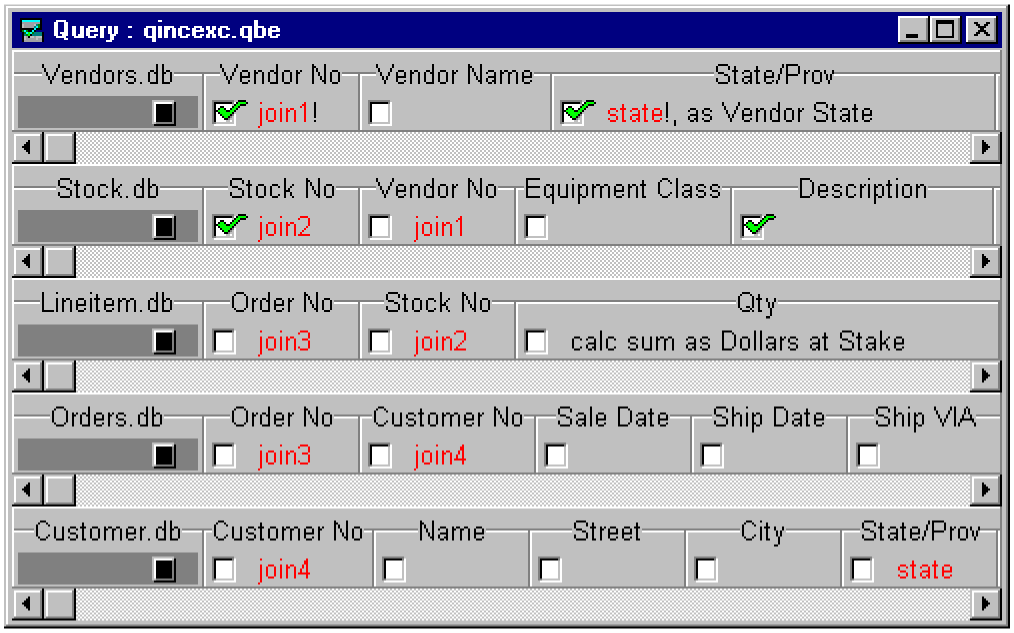
Suppose you have recently agreed with your vendors not to sell items to customer dive shops in the same state as the vendor. You can determine how current orders would be affected by these new agreements by summing their total dollar values.

In an open Query window with blank VENDORS.DB, STOCK.DB, LINEITEM.DB, ORDERS.DB, and CUSTOMER.DB query images, in that order,

1. Use the Join Tables ▪ button to place example elements in the Vendor No fields of VENDORS.DB and STOCK.DB, in the Stock No fields of STOCK.DB and LINEITEM.DB, in the Order No fields of LINEITEM.DB and ORDERS.DB, and in the Customer No fields of ORDERS.DB and CUSTOMER.DB.

2. Type `!` after the example element in the Vendor No field of VENDORS.DB and check it to see all vendor numbers, whether you've ordered stock from them or not.
3. Check the State/Prov field of VENDORS.DB, then press F5 and type `state` as the example element representing each State/Prov value in the Vendors table.

4. Still in the State/Prov field, type `!` after the state example element to see all vendor states and then type `, as Vendor State` to rename the field in the Answer table.
5. Check the Stock No and Description fields of STOCK.DB to see these fields in the Answer table.
6. In the Total field of LINEITEM.DB, type `calc sum as Dollars at Stake` to generate a new calculated field in the Answer table. This new field contains summary values of the total order cost for each stock item ordered by each customer located in the same state as a vendor selling that stock item.
7. In the State/Prov field of the CUSTOMER.DB query image, press F5 and type `state` as the example element representing each customer's state.
8. Run the query.



Query : qincexc.qbe

Vendors.db	Vendor No	Vendor Name	State/Prov
	✓ join1!		✓ state!, as Vendor State

Stock.db	Stock No	Vendor No	Equipment Class	Description
	✓ join2	join1		✓

Lineitem.db	Order No	Stock No	Qty
	join3	join2	calc sum as Dollars at Stake

Orders.db	Order No	Customer No	Sale Date	Ship Date	Ship VIA
	join3	join4			

Customer.db	Customer No	Name	Street	City	State/Prov
	join4				state

The Answer table contains

▪ All vendors, whether or not you have ordered stock from them

▪ The states that those vendors are located in, and that are, by extension, the same states dive shop customers are located in who have ordered stock from you, which you, in turn, could have purchased from a vendor in the same state (Vendor State, inclusively linked with State/Prov in

CUSTOMER.DB)

- All stock items that have been ordered (Stock No and Description
- if blank, you have not ordered stock from that vendor)
- The sum of total orders for each stock number for which a customer could have purchased the same stock item from a vendor selling it in the same state (Dollars at Stake)

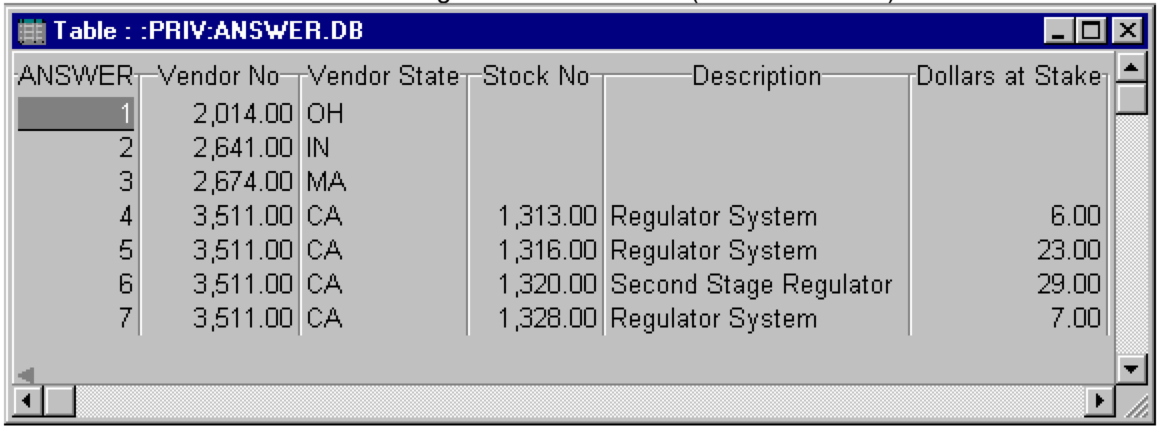


Table : :PRIV:ANSWER.DB

ANSWER	Vendor No	Vendor State	Stock No	Description	Dollars at Stake
1	2,014.00	OH			
2	2,641.00	IN			
3	2,674.00	MA			
4	3,511.00	CA	1,313.00	Regulator System	6.00
5	3,511.00	CA	1,316.00	Regulator System	23.00
6	3,511.00	CA	1,320.00	Second Stage Regulator	29.00
7	3,511.00	CA	1,328.00	Regulator System	7.00

About queries that change data

See also

Use INSERT, DELETE, and CHANGETO queries to change the data in a table.

INSERT	Inserts new records from one table into another.
DELETE	Deletes records that match conditions you specify.
CHANGETO	Changes existing values to a new value you specify.

The table you change with these queries does not have to be open in a window.

INSERT, DELETE, and CHANGETO queries produce temporary tables, which appear in a separate window. The temporary table holds data that was inserted, deleted, or changed so you can restore the table to what it was before the query if you need to.

You choose INSERT and DELETE from a menu in the leftmost field of a query image. You place CHANGETO in the field containing the value you want to change.

You can combine several operations in a single query. If you do, Database Desktop performs all DELETEs first, then all CHANGETOs, then all INSERTs.

▪

Operation order in a query involving multiple operations

See also

You can perform multiple table-changing operations in a single query. If you have more than one query image in a Query window, the only basic requirement for the query to work is that all tables be linked with example elements.

You can, for example, perform a single query that deletes records from one table, inserts records into another table, and changes values in yet another table. You can also do a query that does an INSERT, DELETE, and CHANGETO operation in a single table.

In such multi-operation queries, these rules describe the order in which Database Desktop performs operations:

1. Database Desktop first retrieves records based on all selection conditions.
2. It next performs any INSERTs specified in the order Database Desktop finds them▪that is, Database Desktop looks in the first query image first, then the second, and so on.
3. Next, Database Desktop performs any CHANGETOs specified in the order it finds them.
4. Next, it performs any DELETEs specified in the order it finds them.
5. Finally, Database Desktop displays the temporary tables that result, including an Answer table, if you checked any fields.

Because Database Desktop performs all DELETEs after it performs all INSERTs, be careful not to design a query that undoes itself, first inserting records and then deleting them from the same table.

You can design intricate queries that save you from having to perform multiple, sequential queries. The more operations you design into a single query, however, the harder it becomes for you to undo the query.

▪

About INSERT queries

See also Examples

Use an INSERT query to insert records from one or more sources into a single target table. INSERT queries let you map which values from your source(s) to insert into fields of your target table.

With INSERT you can insert records from one table type to another, for example, from dBASE to Paradox or Paradox to dBASE tables. For example, you can put

- Any numeric data into any numeric field type, Paradox or dBASE
- Alphanumeric or character data into any alpha or character field
- Dates into date fields

Fields you leave blank (with no example element) in the target table receive no values from the source table(s). You cannot put example elements in Paradox BLOB or bytes fields or in dBASE memo fields, so you cannot insert these types of values into these types of fields.

Instead of producing an Answer table, an INSERT query produces a temporary table called Inserted, which includes only the records inserted.

INSERT query temporary tables

See also

Database Desktop generates one or two temporary tables during an INSERT query.

Inserted

An INSERT query produces a temporary table called Inserted. As with an Answer table, Database Desktop saves Inserted to your private directory, overwrites it each time you run an INSERT query, and deletes it when you exit the program. You can use Tools|Utilities|Rename to save Inserted under a different name.

Note: If you choose Query|Properties, then check Fast Queries on the QBE page of the Query Properties dialog box, Database Desktop does not create the Inserted table.

You can produce an Answer table in addition to the Inserted table if you check fields on a separate line of the target query image. If you also supply selection conditions on that line, the records in the Answer table will reflect those conditions. However, such an Answer table does not contain any information that has to do with the INSERT operation. See Operation order in a query involving multiple operations for more information.

You can use the Inserted table along with DELETE to undo an insertion.

Errorins

If you try to insert records that violate the referential integrity of the target table or that violate validity checks established for that table (except picture validity checks), Database Desktop places the new records into a temporary table called Errorins. Those records that do not violate referential integrity or validity checks are placed in Inserted.

To perform an INSERT query

See also

1. Add the source and target tables to the Query window. (If the target table is new, you must create it before you create the query.)
2. Link all tables using example elements.
3. For each source table, specify any selection conditions.
4. In the target table, place the word `Insert` in the leftmost column (under the table name) by doing any of the following in that column:
   - Type the letter `i`.
   - Right-click and choose Insert from the menu of query operations.
   - Press Spacebar, then choose Insert from the menu of query operations that appears.

   Do not check any of the fields on the same line as the INSERT operator, or you will get an error.
5. Run the query.

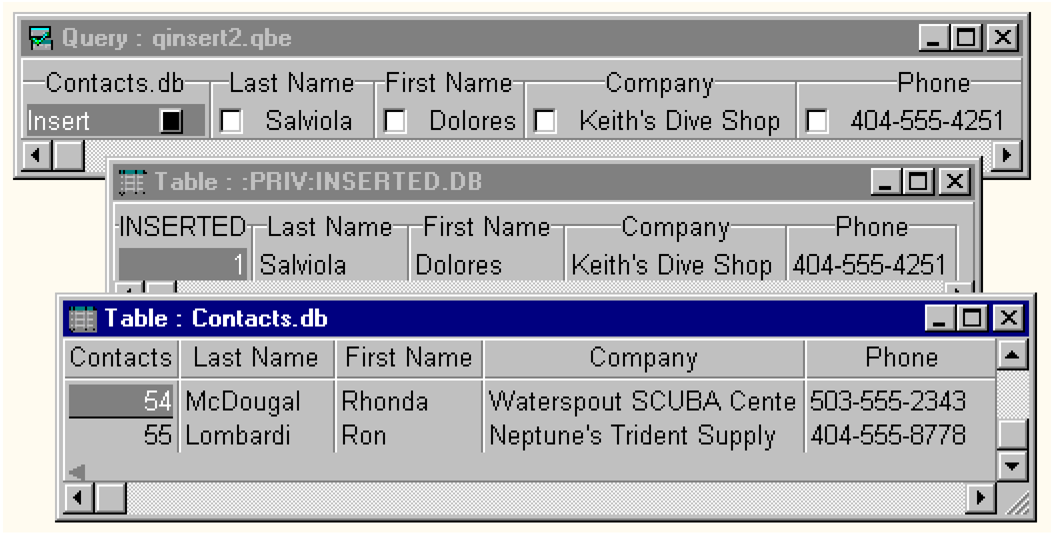
Database Desktop inserts the records from the source into the target table for every field you specified. The source table is not affected by the INSERT query.

Example of inserting a record with an INSERT query

See also

Suppose you want to insert a record of literal values into the Contacts table using an INSERT query. In a Query window that has the CONTACTS.DB query image,

1. Choose Insert from the menu of query operations in the leftmost column of CONTACTS.DB.
2. In the Last Name field, type `Salviola`.
3. In the First Name field, type `Dolores`.
4. In the Company field, type `Keith's Dive Shop`.
5. In the Phone field, type `404-555-4251`.
6. Run the query. Database Desktop opens the Inserted table.
7. Choose File|Open|Table and, from the Open Table dialog box, select the Contacts table. Move to the end of Contacts to see the record you inserted.



Contacts.db	Last Name	First Name	Company	Phone
Insert	☐ Salviola	☐ Dolores	☐ Keith's Dive Shop	☐ 404-555-4251

INSERTED	Last Name	First Name	Company	Phone
1	Salviola	Dolores	Keith's Dive Shop	404-555-4251

Contacts	Last Name	First Name	Company	Phone
54	McDougal	Rhonda	Waterspout SCUBA Cente	503-555-2343
55	Lombardi	Ron	Neptune's Trident Supply	404-555-8778

Example of inserting new values with an INSERT query

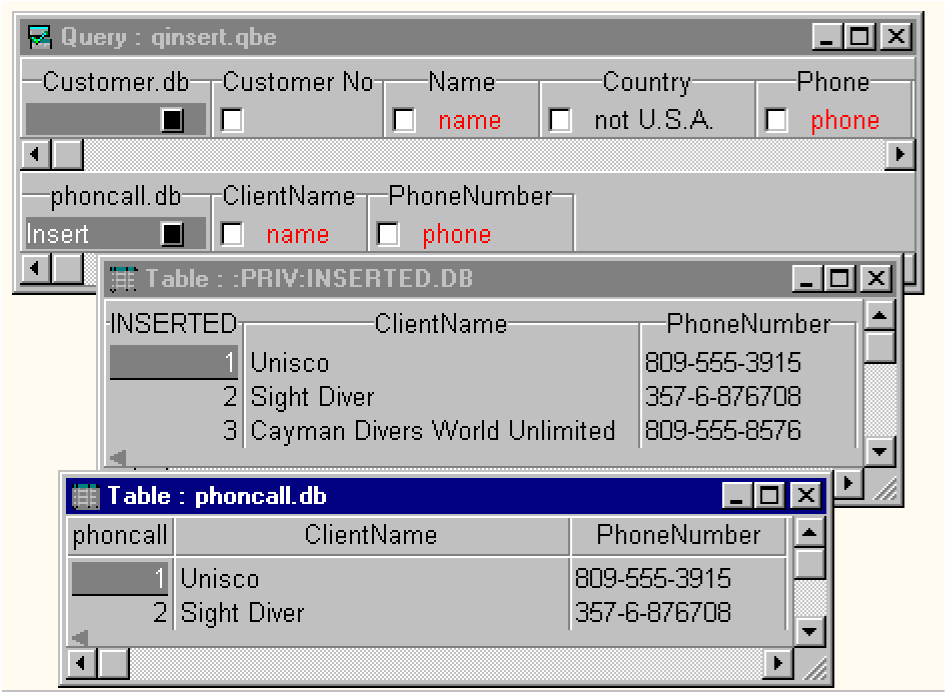
See also

Suppose you find out you can get a cheaper phone rate for international calls if you switch to a different long distance service. Before you switch long distance companies, however, you want to see just how many customers are located outside the U.S.

This example uses demonstrates an INSERT query that places all international customers in a new Phoncall table. You must define the structure of the Phoncall table before you can use INSERT to add data to it. Create the Phoncall table by borrowing its structure from the Customer table, deleting all fields except Name and Phone, renaming Name as ClientName, and renaming Phone as PhoneNumber.

In a Query window that has the CUSTOMER.DB and PHONCALL.DB query images,

1. In the Name field of CUSTOMER.DB, press F5 and type `name` for the example element.
2. In the Country field of CUSTOMER.DB, type `not U.S.A.` This inserts into Phoncall only those dive shops not in the U.S.
3. In the Phone field of CUSTOMER.DB, press F5 and type `phone` for the example element.
4. In PHONCALL.DB, right-click under the table name and choose Insert from the menu of query operations.
5. In the Client Name field of PHONCALL.DB, press F5 and type `name` for the example element.
6. In the Phone Number field of PHONCALL.DB, press F5 and type `phone` for the example element.
7. Run the query. Database Desktop opens the Inserted table on the Desktop.
8. Choose File|Open|Table and, from the Open Table dialog box, select PHONCALL.DB. Because Phoncall was empty before this operation, its records should exactly match the records in the Inserted table.



Customer.db	Customer No	Name	Country	Phone
■	☐	☐ name	☐ not U.S.A.	☐ phone

phoncall.db	ClientName	PhoneNumber
Insert ■	☐ name	☐ phone

INSERTED	ClientName	PhoneNumber
1	Unisco	809-555-3915
2	Sight Diver	357-6-876708
3	Cayman Divers World Unlimited	809-555-8576

phoncall	ClientName	PhoneNumber
1	Unisco	809-555-3915
2	Sight Diver	357-6-876708

You can get the results of this particular INSERT query much faster by doing a CheckPlus query, placing a CheckPlus in the Name and Phone fields of CUSTOMER.DB, and saving the Answer table as Phoncall. A CheckPlus query is not always a more efficient alternative to an INSERT query, however, so this example provides the framework for more complex ones.

About DELETE queries

See also Examples

Use DELETE queries to remove selected records from a table. DELETE queries are effective when the records to be deleted have something in common that you can specify in one or more selection conditions.

DELETE removes only records, not specific field values within records. Use CHANGETO to change or remove specific field values.

Instead of producing an Answer table, a DELETE query produces a temporary table in the Private directory called Deleted, which includes only the records deleted.

▪
DELETE query temporary tables

See also

Database Desktop generates one or two temporary tables during a DELETE query.

Deleted

A DELETE query produces a temporary table called Deleted, which contains only the deleted records. Database Desktop saves Deleted to your private directory, overwrites it each time you run a DELETE query, and deletes it when you exit the program. You can use Tools|Utilities|Rename to save Deleted under a different name.

Note: If you choose Query|Properties, then check Fast Queries on the QBE page of the Query Properties dialog box, Database Desktop does not create the Deleted table.

You can produce an Answer table in addition to the Deleted table if you check fields on a separate line of the query image. If you also supply selection conditions on that line, the records in the Answer table will reflect those conditions, as you might expect. However, such an Answer table is not particularly valuable, since it does not contain any information that has to do with the DELETE operation.

You can use Deleted, along with INSERT, to undo a deletion. Use Deleted as the source table and insert Deleted's records back into the table from which they were deleted. If you are reinserting records you deleted from an unkeyed table, the records are inserted at the end of the table and thus will not necessarily be in their original order.

You can also reinsert the deleted records in Deleted into the original table with Tools|Utilities|Add. Apart from these two methods, you have no other way of recovering records deleted from a Paradox table. (With a dBASE table, you can view the table, enter Edit mode, and choose Table|Show Deleted, then undelete each deleted record one at a time using Record|Undelete.)

Errordel

If you try to delete records that violate the referential integrity of the target table or that violate validity checks established for that table (except picture validity checks), Database Desktop places the new records into a temporary table called Errordel. Those records that do not violate referential integrity or validity checks are placed in Deleted.

To perform a DELETE query

See also

1. Add to the Query window the table from which you want to delete records and the table(s), if any, you want to join to the target table and use to define selection conditions.
2. Place the word Delete in the leftmost column (under the table name) of the table whose records you want to delete by doing any of the following in that column:

- Type the letter `d`.
- Right-click and choose Delete from the menu of query operations that appears.
- Press Spacebar, then choose Delete from the menu of query operations that appears.

   Do not check any of the fields on the same line of the query image as the DELETE operator, or you will get an error.

3. Enter any selection condition to select the records to be deleted. You can enter selection conditions in several fields of the same query image or in fields of tables linked by example elements.

   Caution: If you do not enter any selection conditions, Database Desktop deletes all the records from the table.

4. Run the query.

Database Desktop deletes from the table all records that meet the selection conditions.

Instead of producing an Answer table, a DELETE query produces a temporary table called Deleted, which includes only the records deleted.

▪

Example of removing a record with a DELETE query

See also

Suppose Larry's Diving School has gone out of business and you want to remove this dive shop from the Contacts table.

In a Query window with the Contacts table query image,

1. In the leftmost column, right-click and choose Delete from the menu of query operations.
2. In the Company field, type:

```
  Larry's Diving School
```

3. Run the query.

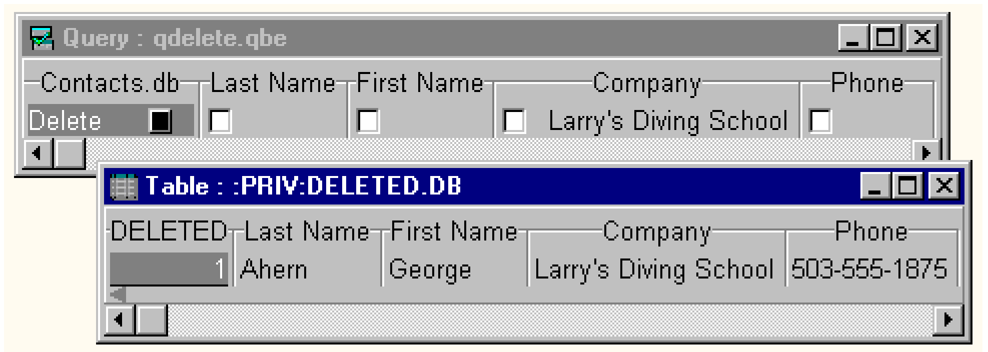
Database Desktop opens the Deleted table. To undo this query, follow the steps in Example of undoing a DELETE query.

▪

Example of undoing a DELETE query

See also

You can undo a DELETE query with an INSERT query. For example, suppose Larry's Diving School has gone out of business and you want to remove this dive shop from the Contacts table. Here is how you could do that (see Example of removing a record with a DELETE query):



Query : qdelete.qbe

Contacts.db	Last Name	First Name	Company	Phone
Delete	☐	☐	☐ Larry's Diving School	☐

Table : :PRIV:DELETED.DB

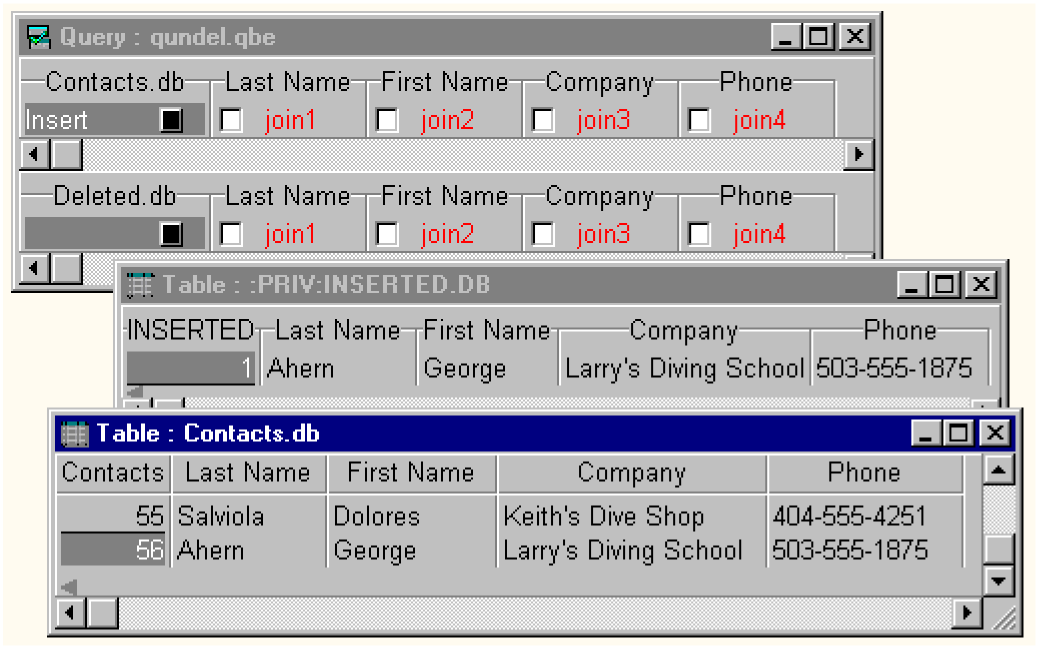
DELETED	Last Name	First Name	Company	Phone
1	Ahern	George	Larry's Diving School	503-555-1875

Suppose you change your mind and decide after you have deleted the contact for Larry's Diving School that you want to keep George Ahern as a contact for potential dive shop customers.

The easiest way to undo the deletion in this case would be to use Tools|Utilities|Add, adding the deleted record in Deleted back into Contacts. This example shows you another way to undo. The method you use will depend on the complexity of the deletion you are trying to undo. With any method, you should make copies of the tables at each stage in case you make a mistake in the recovery process and have to undo it.

Using the Query window from the previous example,

1. Clear the existing selection conditions in the CONTACTS.DB query image by pressing Ctrl+Del in any field of the image.
2. Add the DELETED.DB query image to the Query window. (Follow the instructions in To add tables to a query; choose PRIV: in the Alias drop-down list.)
3. Use the Join Tables ▪ button to place corresponding example elements in each pair of matching fields in CONTACTS.DB and DELETED.DB.
4. In the leftmost column of CONTACTS.DB, choose Insert from the menu of query operations.
5. Run the query.
6. Choose File|Open|Table and, from the Open Table dialog box, select the Contacts table. George Ahern's record is back in Contacts, at the very end.



Query : qundel.qbe

Contacts.db	Last Name	First Name	Company	Phone
Insert	join1	join2	join3	join4

Deleted.db	Last Name	First Name	Company	Phone
	join1	join2	join3	join4

Table : :PRIV:INSERTED.DB

INSERTED	Last Name	First Name	Company	Phone
1	Ahern	George	Larry's Diving School	503-555-1875

Table : Contacts.db

Contacts	Last Name	First Name	Company	Phone
55	Salviola	Dolores	Keith's Dive Shop	404-555-4251
56	Ahern	George	Larry's Diving School	503-555-1875

About CHANGETO queries

See also Examples

Use CHANGETO queries to change specific field values in a table based on conditions you specify in a query. CHANGETO provides you with a kind of global search-and-replace capability. It is particularly useful when you want to change many values that have something in common in a similar way.

Instead of producing an Answer table, a CHANGETO query produces a temporary table called Changed, which contains a copy of the records you changed as they existed before you changed them.

CHANGETO query temporary tables

See also

Database Desktop generates one or two temporary tables during a CHANGETO query.

Changed

CHANGETO produces a temporary table called Changed, which contains a copy of the records you changed as they existed before you changed them. Database Desktop saves Changed to your private directory, overwrites it each time you run a CHANGETO query, and deletes it when you exit the program. You can use Tools|Utilities|Rename to save Changed under a different name.

Note: If you choose Query|Properties, then check Fast Queries on the QBE page of the Query Properties dialog box, Database Desktop does not create the Changed table.

You can produce an Answer table in addition to the Changed table if you check fields on a separate line of the query image. If you also supply selection conditions on that line, the records in the Answer table will reflect those conditions, as you might expect. However, such an Answer table is not particularly valuable since it does not contain any information that has to do with the CHANGETO operation.

You can use the Changed table to back out changes made with CHANGETO. See To undo changes using the Changed table.

Errorchg

If you try to change records in a way that violates the referential integrity of the table or that violate validity checks established for that table (except picture validity checks), Database Desktop places the new records into a temporary table called Errorchg. Only those records that do not violate referential integrity or validity checks are placed in Changed.

To perform a CHANGETO query

See also

1. Type the value you want to change in the field of the query image.
2. After the value you want to change, type a comma.
3. After the comma, type `CHANGETO` and a space. (As with all of Database Desktop's operators, you can type it in uppercase or lowercase.)
4. After CHANGETO and the space, type the new value you want to change the current value to. You can also type selection conditions in other fields to specify further which records to change.

   The CHANGETO operator must be on the same line in the query image as any selection conditions. Do not check any of the fields on this line of the query image, or you will get an error.
5. Run the query.

Database Desktop changes all records that meet the selection conditions.

Instead of producing an Answer table, a CHANGETO query produces a temporary table called Changed, which contains a copy of the records you changed as they existed before you changed them.

To undo changes using the Changed table

See also

Use the Changed table to verify that the correct records have been changed. If you changed records you did not mean to change, you can delete the changed records from the queried table and reinsert the original records back into the table from Changed.

1. Run a DELETE query on the table whose records you accidentally changed, using the new field value(s)▪the ones you changed to
▪as a selection condition(s).
   This removes the incorrect records.
2. Insert Changed's records back into the original table, using Changed as the source table and the original table as the target table, in an INSERT query. (For instructions, see To perform an INSERT query.)
   This restores the queried table back to its original state. (If you are reinserting records into an unkeyed table, Database Desktop inserts them at the end of the table. Thus, they will not necessarily be in the same order they were originally in before you deleted them.)

To perform a multi-table CHANGETO query

See also

You can create a CHANGETO query to change the records in one table to match the records in another table that is linked to it through referential integrity:

1. Place query images of both the parent and child tables in a Query window.
2. Use the Join Tables ▪ button to place example elements in each corresponding field of both query images.
3. In the query image of the parent table, type `changeto` and a space before the example element in each field that you want to change.
4. Run the query.

   Database Desktop changes the values of the appropriate fields of the parent table to match those of the child table.

Example of changing data with a CHANGETO query

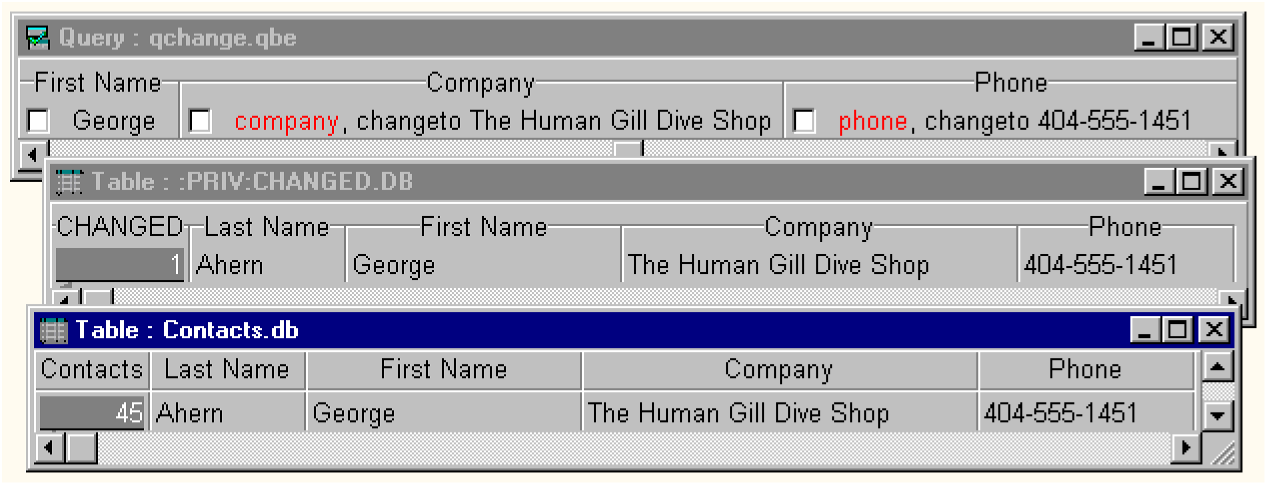
See also

Suppose you learn that George Ahern, the previous contact for the now out-of-business Larry's Diving School, has gotten a job at The Human Gill Dive Shop in Savannah, Georgia. You want to contact George so you can perhaps gain his new employer as one of your customers. You also need to change the company and phone number information about George in the Contacts table. Here is how you would set up the query:

In a Query window that has the CONTACTS.DB query image,

1. In the Last Name field, type `Ahern`.
2. In the First Name field, type `George`.
3. In the Company field, type `company` for the example element, then type a comma and type `changeto`, then type a space and type `The Human Gill Dive Shop`.
4. In the Phone field, type phone for the example element, then type a comma and type `changeto`, then type a space and `404-555-1451`.
5. Run the query.

   Database Desktop opens the Changed table.



Query : qchange.qbe

First Name	Company	Phone
☐ George	☐ company, changeto The Human Gill Dive Shop	☐ phone, changeto 404-555-1451

Table : :PRIV:CHANGED.DB

CHANGED	Last Name	First Name	Company	Phone
1	Ahern	George	The Human Gill Dive Shop	404-555-1451

Table : Contacts.db

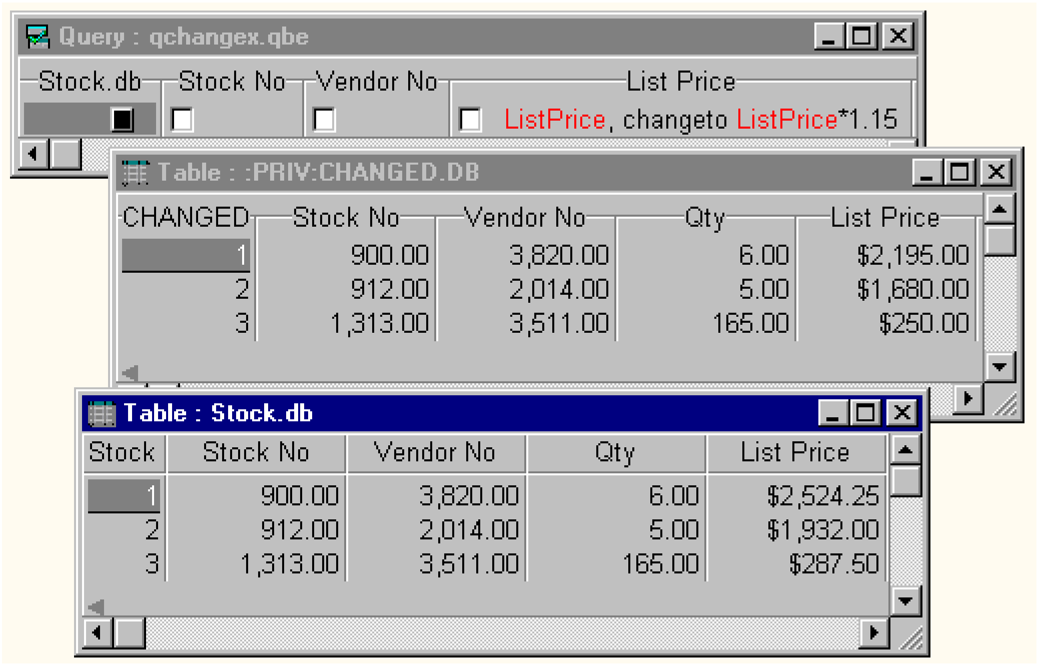
Contacts	Last Name	First Name	Company	Phone
45	Ahern	George	The Human Gill Dive Shop	404-555-1451

Example of using CHANGETO with example elements

See also

You can use a CHANGETO query with example elements to perform a calculation on values in a field and change the original values to the new calculated values in the same field. (If you were to perform calculations using the CALC operator, Database Desktop would create a new field to hold the results in an Answer table and would leave the original values unchanged.)

This query increases the list price of all stock items in the Stock table by 15%. In the query image, ListPrice is an example element that represents the value in the List Price field.



Query : qchangex.qbe

Stock.db	Stock No	Vendor No	List Price
■	☐	☐	☐ ListPrice, changeto ListPrice*1.15

Table : :PRIV:CHANGED.DB

CHANGED	Stock No	Vendor No	Qty	List Price
1	900.00	3,820.00	6.00	$2,195.00
2	912.00	2,014.00	5.00	$1,680.00
3	1,313.00	3,511.00	165.00	$250.00

Table : Stock.db

Stock	Stock No	Vendor No	Qty	List Price
1	900.00	3,820.00	6.00	$2,524.25
2	912.00	2,014.00	5.00	$1,932.00
3	1,313.00	3,511.00	165.00	$287.50

▪

Property

No help is available for this property.

▪

Add Records In dialog box

See also

Use the Add Records In dialog box to add the records in one table to those in another without having to retype them. The Add Records In dialog box indicates the table to add the records from.

To open this dialog box, choose Tools|Utilities|Add.

Dialog box options

Look In

By default, Database Desktop displays the working directory. To choose another directory, use this drop-down list to browse until you reach the directory you want. All files of the selected type in that directory appear in the list below the Look In drop-down list.

If the directory you want has an alias, choose it in the Alias drop-down list. The name of the directory appears in the Look In list box and its files appear in the file list.

Choose any of the icons to navigate, create a folder, or change the display of folders.

File Name

Type the name of the file or select one from the list box below the Look In list. You don't need to type an extension; Database Desktop recognizes the type of file you want based on the type shown in the Files Of Type drop-down list.

Files Of Type

Displays the types of files you can use for the addition operation you are performing.

Alias

If the directory you want has an alias, you can select it in the Alias drop-down list. The name of the selected directory appears in the Look In drop-down list and the files in that directory appear in the file list.

▪

Add Records In <table> To dialog box

See also

Use the Add Records In <table> To dialog box to add the records in one table to those in another without having to retype them.

You can use the Options area in this dialog box to either append new records, update existing records, or both.

To open this dialog box, choose OK in the Add Records In dialog box.

Dialog box options

Look In

By default, Database Desktop displays the working directory. To choose another directory, use this drop-down list to browse until you reach the directory you want. All files of the selected type in that directory appear in the list below the Look In drop-down list.

If the directory you want has an alias, choose it in the Alias drop-down list. The name of the directory appears in the Look In list box and its files appear in the file list.

Choose any of the icons to navigate, create a folder, or change the display of folders.

File Name

Type the name of the file to add records to or select one from the list box below the Look In list. You don't need to type an extension; Database Desktop recognizes the type of file you want based on the type shown in the Files Of Type drop-down list.

Files Of Type

Displays the types of files you can use for the addition operation you are performing.

Alias

If the directory you want has an alias, you can select it in the Alias drop-down list. The name of the selected directory appears in the Look In drop-down list and the files in that directory appear in the file list.

Options

These settings let you append records, update records, or both.

Append

Adds new records from the source table to the target table without affecting any existing records:

- If the target table is not keyed, the records are placed after existing records. Records that violate validity checks are placed in the temporary Keyviol table in your private directory.
- If the target table is keyed, added records that meet the key criteria are inserted in their proper sort order. Records that do not meet the key criteria are stored in the temporary Keyviol table in your private directory. If you want, you can edit these records to meet the key criteria, then use Add again to place them in the target table.

Update

Updates records that already exist in the table you are adding records to. Any records in the source table that do not match an existing record are not added.

When you choose Update, the records of the source table overwrite matching records in the table you are adding records to. Database Desktop places the records that are overwritten in the temporary Changed table in your private directory.

Note: The table you add records to must be keyed to use Update.

Append & Update

Adds new records to a table (following the rules stated above) and updates existing records in the target table (following the rules just stated).

Note: The table you add records to must be keyed to use Append & Update.

Alias Manager dialog box

See also

Use the Alias Manager dialog box to create or modify aliases for local or network directories.

Creating aliases lets you give logical names to directories and is strongly encouraged, since it frees you from typing long path names, and makes your files more portable.

To open this dialog box, choose Tools|Alias Manager.

Dialog box options

Public Alias

Check this check box to make an alias a public alias available from all applications that use BDE. Uncheck this check box to make an alias a project alias

available only to Database Desktop applications in the current working directory.

Database Alias

Choose an alias from the list. To create a new alias, first choose New, then type the name (alias) you want to give the database.

Driver Type

Choose the driver you want. The Driver Type drop-down list shows all the drivers you are connected to. If you want to create a database of Paradox and/or dBASE tables, choose STANDARD.

Path

Type the full path of the directory location, including the drive letter.

Show options

Show Public Aliases Only
Choose this button if you want to see only public aliases.

Show Project Aliases Only
Choose this button if you want to see only project aliases.

Show All Aliases
Choose this button if you want to see both public and project aliases.

Browse

Choose Browse to look for a directory using the Directory Browser.

New

Choose New to open an empty box where you can type in a new alias. After you choose New, the button becomes the Keep New button.

Keep New

Choose Keep New if you want this to be a temporary alias, existing only until you exit. Then click OK, and the Alias Manager dialog box closes.

Note: Keep New does not close the dialog box. It lets you do a temporary save which does not take effect until you click OK. If you click Cancel, whatever you temporarily saved with Keep New is canceled.

Choose Keep New if you are creating several aliases and do not want to open this dialog box to create each one.

Remove

Choose Remove to tag the selected alias for removal. The alias is removed when you exit the box without specifying the removed name again or when you choose Save As and overwrite the current file containing the alias.

Save As

Choose Save As if you want this alias to be permanent▪usable any time you use Database Desktop. You see the Save File As dialog box. By default, Database Desktop stores saved public aliases in IDAPI32.CFG and project aliases in PDOXWORK.CFG. You are prompted to overwrite the existing .CFG file.

Important: When you overwrite, Database Desktop appends the new alias without changing any existing configuration settings. You can undo the change by deleting the alias (using the Alias Manager dialog box).

OK

Choose OK if you want to save any changes you have made in the dialog box, but only for the current Database Desktop session. All Windows applications currently running are affected by any changes.

Cancel

Cancels only the changes in type-in boxes. Any changes you made with Save As remain.

Auxiliary Passwords dialog box

See also

Use the Auxiliary Passwords dialog box to assign passwords for table and field rights.

To open this dialog box, choose the Auxiliary Passwords button in the Password Security dialog box.

Dialog box options

Passwords
Lists the passwords for the current table.

Current Password
To specify an auxiliary password, type it in the Current Password text box.

Add
After choosing the table and field rights for your auxiliary password, click Add to place the password in the Passwords list.

Table Rights
Choose the level of table rights for the password from the Table Rights panel. You can choose only one type of table rights for each auxiliary password. If you want a user to have more than one (but not all) rights, you must assign more than one auxiliary password.

> **All**
> Gives a user all rights to any function of the table, including the ability to restructure or delete it, and to change or delete passwords.
>
> **Insert & Delete**
> Gives a user the right to insert, delete, or empty records, but not to delete or restructure the table.
>
> **Data Entry**
> Gives a user the right to edit data and insert records, but not to delete records, restructure, or empty the table.
>
> **Update**
> Gives a user the right to view the table and change non-key fields, but not to insert or delete records or change key fields.
>
> **Read Only**
> Gives a user the right to view the table, but not to change it in any way.

Field Rights
Assigns rights to individual fields. The default right in the Field Rights list is All. To choose another option, double-click the field or choose the Field Rights button.

> **All**
> Gives a user all rights to the data in that field (within the limits of the table rights you specify).
>
> **Read Only**
> Gives a user the right to view, but not to change, the data in that field.
>
> **None**
> Prevents a user from viewing or changing the data in that field. Database Desktop hides the values in the field when the table is opened.

New
Choose New when you have finished adding one auxiliary password to the list and want to add another

before leaving this dialog box. You can repeat this process to assign any number of auxiliary passwords.

Change
Change a password that's already on the Passwords list by selecting it and then choosing Change.

Delete
Remove a password by selecting it in the Passwords list and choosing Delete.

Copy dialog box

See also

Use the Copy dialog box to specify the file you want to copy. You can copy tables, queries, and SQL files from within Database Desktop.

Note: Do not try to copy tables using the DOS COPY command or the Windows Explorer.

To open this dialog box, choose Tools|Utilities|Copy.

Dialog box options

Look In

By default, Database Desktop displays the working directory. To choose another directory, use this drop-down list to browse until you reach the directory you want. All files of the selected type in that directory appear in the list below the Look In drop-down list.

If the directory you want has an alias, choose it in the Alias drop-down list. The name of the directory appears in the Look In drop-down list box and its files appear in the file list.

Choose any of the icons to navigate, create a folder, or change the display of folders.

File Name

Type the name of the file to copy or select one from the list box below the Look In drop-down list. You don't need to type an extension; Database Desktop recognizes the type of file you want based on the type shown in the Files Of Type drop-down list.

Files Of Type

Displays the types of files you can copy.

Alias

If the directory you want has an alias, you can select it in the Alias drop-down list. The name of the selected directory appears in the Look In drop-down list and the files in that directory appear in the file list.

Copy <file name> To dialog box

See also

Use this dialog box to specify a file name and directory for the destination file in a file copying operation.

To display this dialog box, right-click a file name in the Project Viewer and choose Copy or use Tools|Utilities|Copy and choose OK in the Copy dialog box

Dialog box options

Save In

By default, Database Desktop displays the working directory. To choose another directory, use this drop-down list to browse until you reach the directory where you want to save the file. All files of the selected type in that directory appear in the graphics list below the Save In drop-down list.

If the directory you want has an alias, choose it in the Alias drop-down list. The name of the directory appears in the Save In drop-down list and its files appear in the file list.

Choose any of the icons to navigate, create a folder, or change the display of folders.

File Name

Type the name of the file to save or select one from the list box below the Save In list. You don't need to type an extension; Database Desktop recognizes the type of file you want based on the type shown in the Save As Type drop-down list.

Save As Type

Displays the types of files you can save.

Note: To copy a Paradox table to a dBASE table or vice versa, specify the appropriate extension. For example, if you want to copy NAMES.DBF to a Paradox table, type `NAMES.DB` in the File Name text box. See Copying to a different table type for important information on copying between Paradox and dBASE table types.

Alias

If the directory you want has an alias, you can select it in the Alias drop-down list. The name of the selected directory appears in the Save In drop-down list and the tables in that directory appear in the file list.

Copy To File dialog box

See also

Use the Copy To File dialog box to copy values in a table's binary fields to non-Database Desktop files.

To open this dialog box, choose Edit|Copy To with a binary field selected.

Dialog box options

Save In

By default, Database Desktop displays the working directory. To choose another directory, use this drop-down list to browse until you reach the directory where you want to save the file. All files of the selected type in that directory appear in the file list below the Save In drop-down list.

If the directory you want has an alias, choose it in the Alias drop-down list. The name of the directory appears in the Save In drop-down list and its files appear in the file list.

Choose any of the icons to navigate, create a folder, or change the display of folders.

File Name

Type the name of the file to save or select one from the list box below the Save In list. You don't need to type an extension; Database Desktop recognizes the type of file you want based on the type shown in the Save As Type drop-down list.

Save As Type

Displays the types of files you can save.

Alias

If the directory you want has an alias, you can select it in the Alias drop-down list. The name of the selected directory appears in the Save In drop-down list and the files in that directory appear in the graphics list.

Copy To Graphic File dialog box

See also

Use the Copy to Graphic File dialog box to copy values in graphic fields to non-Database Desktop files.

To open this dialog box, select a graphic object or field and choose Edit|Copy To.

Dialog box options

Save In

By default, Database Desktop displays the working directory. To choose another directory, use this drop-down list to browse until you reach the directory where you want to save the file. All files of the selected type in that directory appear in the graphics list below the Save In drop-down list.

If the directory you want has an alias, choose it in the Alias drop-down list. The name of the directory appears in the Save In drop-down list and its files appear in the graphics list.

Choose any of the icons to navigate, create a folder, or change the display of folders.

File Name

Type the name of the file to save or select one from the list box below the Save In list. You don't need to type an extension; Database Desktop recognizes the type of file you want based on the type shown in the Save As Type drop-down list.

Save As Type

Displays the types of files you can save.

Alias

If the directory you want has an alias, you can select it in the Alias drop-down list. The name of the selected directory appears in the Save In drop-down list and the graphic files in that directory appear in the graphics list.

▪

Create dBASE Table dialog box

See also

Use the Create dBASE Table dialog box to specify the structure of a dBASE table.

This dialog box has two main panels: Field Roster and Table Properties. You can move between them using the keyboard: Use the Super Tab key (F4) to move from Field Roster to Table Properties; to return, press Shift+Tab.

To open this dialog box, choose a dBASE table format in the Create Table dialog box.

Dialog box options

Field Roster

Use the Field Roster to specify the fields of a table. You can add, delete, or rename fields, and change field types and sizes.

- To insert a field between two existing fields in the Field Roster, select a field and press Ins. Database Desktop opens a blank row above the selected field.
- To delete a field from the Field Roster, select it and press Ctrl+Del. Database Desktop deletes the entire row.

The order in which fields are listed in the Field Roster is the order in which the fields appear in the table. To change the field order, click the row number of the field and drag it to a new position.

Field Name
Specifies the name of the field. See Rules for dBASE field names for more information. This is a required field. (When a field is required, you must enter a value in the field for every record in the table. See About required fields for more information.)

Type
Specifies the type of the field. Right-click the Type column or press the Spacebar to display a list of field types. See dBASE field types and sizes for more information. This is a required field.

Size
Specifies the size of the field. See dBASE field types and sizes for more information. This is a required item for some field types.

Dec
Specifies the number of decimal places for number or float fields.

Table Properties

In the Table Properties panel you can specify the following:

Indexes
Creates an index on the current field in the Field Roster. See About dBASE indexes for more information.

Choose Indexes, then choose Define to open the Define Index dialog box.

Once you create an index, you can choose Modify to change it or Erase to remove it.

Table Language
Specifies the language driver. See About table language drivers for more information.

Choose Table Language, then choose Modify to open the Table Language dialog box.

Borrow

Specifies whether to create this table structure by using the structure of another table as a template. Choose Borrow to open the Select Borrow Table dialog box and choose from the list of tables. The Field Roster must be empty to borrow another table's structure.

Record Lock
Contains information about records locked by other users.

Info Size
Specifies whether to keep track of record locking information in a multiuser environment. When you check Info Size, Database Desktop adds a hidden field to the table that shows when a record was locked and by whom.

The amount of information you see when you encounter a locked field depends on the Info Size you specify. The default size is 16 characters. You can choose a size from 8 to 24 from the Info Size drop-down list. See dBASE Record Lock fields for more information.

Save As
Saves the table you are creating and closes the Create dBASE Table dialog box. Choose Save As to open the Save Table As dialog box, where you type a name for your new table. You can save the table in the current directory or another one.

Create INFORMIX Table dialog box

See also

Use the Create INFORMIX Table dialog box to specify the structure of an Informix table.

This dialog box has two main sections: the Field Roster panel and the panels on the right. You can move between them using the keyboard: Use the Super Tab key (F4) to move from Field Roster to the panels on the right; to return, press Shift+Tab.

To open this dialog box, choose INFORMIX in the Create Table dialog box.

Field Roster

In the Field Roster, you specify the fields of a table. When you are creating a table, you can add, delete, or rename fields, and change field types and sizes:

Field Name	Required for every field
Type	Required for every field. Right-click or press Spacebar to choose a field type.
Size	Table type determines which fields require this
Dec	Table type determines which fields require this

Required Field

The Required Field check box (in the panel on the right) specifies whether the selected field is required. Check to make the selected field required. When a field is required, you must enter a value in the field for every record in the table. See About required fields for more information.

List of Indexes

In the panel on the right, you create indexes for the table. You can add indexes, modify existing indexes, and erase indexes.

Define Index	Choose Define Index to create an index. Database Desktop opens the Define Index dialog box.
Modify Index	Choose Modify Index to change the selected index. Database Desktop opens the Define Index dialog box.
Erase Index	Choose Erase Index to remove the selected index. Database Desktop erases the index.

Borrow

You can borrow another table's structure. Choose Borrow to open the Select Borrow Table dialog box and choose from the list of tables. The Field Roster must be empty to borrow another table's structure.

▪

Create INTRBASE Table dialog box

See also

Use the Create INTRBASE Table dialog box to specify the structure of an InterBase table.

This dialog box has two main sections: the Field Roster panel and the panels on the right. You can move between them using the keyboard: Use the Super Tab key (F4) to move from Field Roster to the panels on the right; to return, press Shift+Tab.

To open this dialog box, choose INTRBASE in the Create Table dialog box.

Field Roster

In the Field Roster, you specify the fields of a table. When you are creating a table, you can add, delete, or rename fields, and change field types and sizes:

Field Name	Required for every field
Type	Required for every field. Right-click or press Spacebar to choose a field type.
Size	Table type determines which fields require this
Dec	Table type determines which fields require this

Required Field

The Required Field check box (in the panel on the right) specifies whether the selected field is required. Check to make the selected field required. When a field is required, you must enter a value in the field for every record in the table. See About required fields for more information.

List of Indexes

In the panel on the right, you create indexes for the table. You can add indexes, modify existing indexes, and erase indexes.

Define Index	Choose Define Index to create an index. Database Desktop opens the Define Index dialog box.
Modify Index	Choose Modify Index to change the selected index. Database Desktop opens the Define Index dialog box.
Erase Index	Choose Erase Index to remove the selected index. Database Desktop erases the index.

Borrow

You can borrow another table's structure. Choose Borrow to open the Select Borrow Table dialog box and choose from the list of tables. The Field Roster must be empty to borrow another table's structure.

▪

Create ORACLE Table dialog box

See also

Use the Create ORACLE Table dialog box to specify the structure of an Oracle table.

This dialog box has two main sections: the Field Roster panel and the panels on the right. You can move between them using the keyboard: Use the Super Tab key (F4) to move from Field Roster to the panels on the right; to return, press Shift+Tab.

To open this dialog box, choose ORACLE in the Create Table dialog box.

Field Roster

In the Field Roster, you specify the fields of a table. When you are creating a table, you can add, delete, or rename fields, and change field types and sizes:

Field Name	Required for every field
Type	Required for every field. Right-click or press Spacebar to choose a field type.
Size	Table type determines which fields require this
Dec	Table type determines which fields require this

Required Field

The Required Field check box (in the panel on the right) specifies whether the selected field is required. Check to make the selected field required. When a field is required, you must enter a value in the field for every record in the table. See About required fields for more information.

List of Indexes

In the panel on the right, you create indexes for the table. You can add indexes, modify existing indexes, and erase indexes.

Define Index	Choose Define Index to create an index. Database Desktop opens the Define Index dialog box.
Modify Index	Choose Modify Index to change the selected index. Database Desktop opens the Define Index dialog box.
Erase Index	Choose Erase Index to remove the selected index. Database Desktop erases the index.

Borrow

You can borrow another table's structure. Choose Borrow to open the Select Borrow Table dialog box and choose from the list of tables. The Field Roster must be empty to borrow another table's structure.

▪

Create Paradox Table dialog box

See also

Use the Create Paradox Table dialog box to specify the structure of a Paradox table.

This dialog box has two main panels: Field Roster and Table Properties. You can move between them using the keyboard: Use the Super Tab key (F4) to move from Field Roster to Table Properties; to return, press Shift+Tab.

To open this dialog box, choose a Paradox table format in the Create Table dialog box.

Dialog box options

Field Roster

Use the Field Roster to specify the fields of a table. You can add, delete, or rename fields, and change field types and sizes.

- To insert a field between two existing fields in the Field Roster, select a field and press Ins. Database Desktop opens a blank row above the selected field.
- To delete a field from the Field Roster, select it and press Ctrl+Del. Database Desktop deletes the entire row.

The order in which fields are listed in the Field Roster is the order in which the fields appear in the table. To change the field order, click the row number of the field and drag it to a new position.

Field Name
Specifies the name of the field. See Rules for Paradox field names for more information. This is a required field.

Type
Specifies the type of the field. Right-click the Type column or press the Spacebar to display a list of field types. See Paradox field types and sizes for more information. This is a required field.

Size
Specifies the size of the field. See Paradox field types and sizes for more information. This is a required field for some field types.

Key
Specifies whether the field is a key field. The table type determines rules for Paradox key fields. See About primary indexes (key fields) for more information.

Table Properties

In the Table Properties panel you can specify the following:

Validity Checks
Specifies requirements and defaults for a field. You must have a valid entry selected in the Field Roster area to specify validity check information. For more information about validity checks, see About validity checks.

You can specify the following types of validity checks:

- **Required Field**: Specifies that the selected field in the Field Roster is a required field. When a field is required, you must enter a value in the field for every record in the table. See About required fields for more information.
- **Minimum**: Specifies a minimum value for the selected field in the Field Roster. When a field has a minimum validity check, the values entered in the field must be greater than or equal to the minimum you specify here. See About minimum and maximum values for more information.
- **Maximum**: Specifies a maximum value for the selected field in the Field Roster. When a field has a maximum validity check, the values entered in the field must be less than or equal to the maximum you specify here. See About minimum and maximum values for more information.
- **Default**: Specifies a default value for the selected field in the Field Roster. When a field has a

default validity check, Database Desktop enters the value you specify here if you do not enter another value when you edit this field. See About default values for more information.

- **Picture**: Restricts the types of information you can enter in a field. When a field has a picture validity check, you specify a character string as a template for the values that can be entered into this field. See About pictures for more information.
- **Assist**: Opens the Picture Assistance dialog box, where you can select or modify a predefined string to use as a picture.

Table Lookup
Specifies a lookup table for the current field in the Field Roster. A lookup table is another table that contains values that are valid for the current field. See About table lookups for more information.

Choose Table Lookup, then choose Define to open the Table Lookup dialog box.

Secondary Indexes
Creates a secondary index on the current field in the Field Roster. A secondary index lets you sort data in an order different from the key field, and lets you form links between tables. See About secondary indexes for more information.

Choose Secondary Indexes, then choose Define to open the Define Secondary Index dialog box.

Referential Integrity
Creates a referential integrity relationship between the current field and the key field in another table. A referential integrity relationship ensures that ties between like data in separate tables cannot be broken. See About referential integrity for more information.

Chose Referential Integrity, then choose Define to open the Referential Integrity dialog box.

Password Security
Creates passwords to protect your tables from unauthorized access. See About password security for more information.

Choose Password Security, then choose Define to open the Password Security dialog box.

Table Language
Specifies the language driver. See About table language drivers for more information.

Choose Table Language, then choose Modify to open the Table Language dialog box.

Dependent Tables
Displays all tables that depend on the current table for referential integrity.

Borrow
Specifies whether to create this table structure by using the structure of another table as a template. Choose Borrow to open the Select Borrow Table dialog box and choose from the list of tables. The Field Roster must be empty to borrow another table's structure.

Save As
Saves the table you are creating and closes the Create Paradox Table dialog box. Choose Save As to open the Save Table As dialog box, where you type a name for your new table. You can save the table in the current directory or another one.

Create SYBASE Table dialog box

See also

Use the Create SYBASE Table dialog box to specify the structure of an Sybase table.

This dialog box has two main sections: the Field Roster panel and the panels on the right. You can move between them using the keyboard: Use the Super Tab key (F4) to move from Field Roster to the panels on the right; to return, press Shift+Tab.

To open this dialog box, choose SYBASE in the Create Table dialog box.

Field Roster

In the Field Roster, you specify the fields of a table. When you are creating a table, you can add, delete, or rename fields, and change field types and sizes:

Field Name	Required for every field
Type	Required for every field. Right-click or press Spacebar to choose a field type.
Size	Table type determines which fields require this
Dec	Table type determines which fields require this

Required Field

The Required Field check box (in the panel on the right) specifies whether the selected field is required. Check to make the selected field required. When a field is required, you must enter a value in the field for every record in the table. See About required fields for more information.

List of Indexes

In the panel on the right, you create indexes for the table. You can add indexes, modify existing indexes, and erase indexes.

Define Index	Choose Define Index to create an index. Database Desktop opens the Define Index dialog box.
Modify Index	Choose Modify Index to change the selected index. Database Desktop opens the Define Index dialog box.
Erase Index	Choose Erase Index to remove the selected index. Database Desktop erases the index.

Borrow

You can borrow another table's structure. Choose Borrow to open the Select Borrow Table dialog box and choose from the list of tables. The Field Roster must be empty to borrow another table's structure.

▪

Create Table dialog box

See also

Use the Create Table dialog box to specify the type of table to create.

Table type determines

- The table's file-name extension.
- Which tables you can borrow a structure from.
- What are valid field names. For example, Paradox tables permit spaces and punctuation in field names, while dBASE tables do not.
- What are valid field types and sizes.
- The rules for specifying indexes.

To open this dialog box, choose File|New|Table from the Desktop. Or right-click the Open Table Toolbar button and choose New.

Dialog box options

Table Type

Specifies the type of table to create. You can choose any table type on the drop-down list.

Tables created using the Paradox 4 option are compatible with Paradox 4.5 for Windows and Paradox 4.0 for DOS.

▪

Create Table dialog box (other SQL)

See also

Use the Create Table dialog box to specify the structure of an SQL table.

This dialog box has two main sections: the Field Roster panel and the panels on the right. You can move between them using the keyboard: Use the Super Tab key (F4) to move from Field Roster to the panels on the right; to return, press Shift+Tab.

To open this dialog box, choose an SQL driver in the Create Table dialog box.

Field Roster

In the Field Roster, you specify the fields of a table. When you are creating a table, you can add, delete, or rename fields, and change field types and sizes:

Field Name	Required for every field
Type	Required for every field. Right-click or press Spacebar to choose a field type.
Size	Table type determines which fields require this
Dec	Table type determines which fields require this

Required Field

The Required Field check box (in the panel on the right) specifies whether the selected field is required. Check to make the selected field required. When a field is required, you must enter a value in the field for every record in the table. See About required fields for more information.

List of Indexes

In the panel on the right, you create indexes for the table. You can add indexes, modify existing indexes, and erase indexes.

Define Index	Choose Define Index to create an index. Database Desktop opens the Define Index dialog box.
Modify Index	Choose Modify Index to change the selected index. Database Desktop opens the Define Index dialog box.
Erase Index	Choose Erase Index to remove the selected index. Database Desktop erases the index.

Borrow

You can borrow another table's structure. Choose Borrow to open the Select Borrow Table dialog box and choose from the list of tables. The Field Roster must be empty to borrow another table's structure.

Define Index dialog box or Index Info dialog box (dBASE tables)

See also

Use the Define Index and Index dialog box to define or display indexes for dBASE tables.

To open the Define Index dialog box, choose Define in the Create dBASE Table dialog box or the Restructure dBASE Table dialog box.

To open the Index Info dialog box, select an index in the Structure Information dialog box and choose Detail Info.

Dialog box options

Field List

Database Desktop displays the fields in your table. Select the one you want to appear in the Indexed Field box.

Indexed Field

Displays the field you have selected.

In the Define Index dialog box, you can select the field you want in the Field List and use the Add Field arrow to add it to the Indexed Fields list (or press Alt+A). To remove a selected field, use the Remove Field arrow

.

The Add Field and Remove Field arrows are unavailable in the Index Info dialog box.

Expression Index/Index Field (button)

Specifies whether the index is an expression index or a field index. By default, the button reads Expression Index and Database Desktop is set to create a field index.

- In the Define Index dialog box, you can choose Expression Index to create an expression index. The Expression Index box becomes available.
- If the Index Field box isn't available, you can choose Index Field to create an index on a field.

Options

Indicates how you want Database Desktop to treat your indexes.

Unique

Tells Database Desktop that each value in the index must be unique.

Unique is not equivalent to a Paradox table key. It does not prevent you from entering duplicate values for fields in the index; rather, it only shows you one record for a given set of values for the index.

Maintained

Tells Database Desktop to maintain the index automatically. This means every time the table changes, Database Desktop updates the index.

Descending

When checked, the index sorts the table in descending (Z to A) order.

Expression Index

Specifies the expression to use for an expression index. In the Define Index dialog box, you can type any formula that results in a value. For example, you could create an expression index such as FIRST_NAME + LAST_NAME, where FIRST_NAME and LAST_NAME are field names. For more information, see Creating a dBASE expression index.

Subset Condition (filter) Expression

In the Define Index dialog box, lets you create an expression (sometimes called a filter) that evaluates to true or false. For details, see Creating a subset condition expression.

▪

Define Index dialog box or Index Info dialog box (SQL tables)

See also

Use the Define Index and Index Info dialog boxes to define or display indexes for SQL tables.

To open the Define Index dialog box, do one of the following:

- Choose Define Index in the Create Table dialog box or the Restructure Table dialog box.
- Select an index in the Create Table dialog box or the Restructure Table dialog box and choose Modify Index.

To open the Index Info dialog box:

- Select an index in the Structure Information dialog box and choose Detail Info.

Dialog box options

Field List

Displays the fields in your table. Select the fields you want to appear in the Indexed Field box.

Indexed Field

Displays the field for the index.

In the Define Index dialog box, you can select the field you want in the Field List and use the Add Field arrow ▪ to add it to the Indexed Fields list (or press Alt+A). To remove a selected field, use the Remove Field arrow

▪.

The Add Field and Remove Field arrows are unavailable in the Index Info dialog box.

Change Order

Changes the order of the fields in the Indexed Fields list.

In the Define Index dialog box, you can select a field and use the Change Order arrows to move it up or down. These arrows become available when two or more fields appear in the Indexed Fields list. Change the order of the fields to change the sort order of the index.

This field is unavailable in the Index Info dialog box.

Index Options

Indicate how you want Database Desktop to treat your indexes. These options are available only if they are supported by your server.

In the Index Info dialog box, these options are for information only and cannot be changed.

Unique

Tells Database Desktop that each value in the index must be unique. The index accepts only the first occurrence of any duplicate field values.

Descending

When checked, the index sort the table in descending (Z to A) order. With Descending checked, if you try to link to another table that is sorted in ascending (A to Z) order, you will not be able to perform the link.

Case Sensitive

When Case Sensitive is checked, Database Desktop pays attention to capitalization in sorting.

Note: Capitalizing a value does not make it unique in a case-insensitive index.

▪

Define Secondary Index dialog box (Paradox tables)

See also

Use the Define Secondary Index dialog box to define secondary indexes for Paradox tables.

To open this dialog box, choose Secondary Indexes, then choose Define in the Create Paradox Table dialog box or the Restructure Paradox Table dialog box. Or select an index and choose Modify.

For Paradox 3.5 tables, you can't define a secondary index based on more than one field.

Dialog box options

Fields

Database Desktop displays a list of the fields you can use as a secondary index. Bytes, logical, and BLOB fields are dimmed▪you cannot create a secondary index on these fields.

Indexed Fields

Select the field you want and use the Add Field arrow to add it to the Indexed Fields list, or press Alt+A. To remove a selected field, use the Remove Field arrow.

If you add more than one field, Database Desktop creates a composite secondary index and sorts on all included fields, starting at the top of the list. For details, see About composite secondary indexes.

Add Field arrow ▪

Adds the selected field in the Fields list to the Indexed Fields list.

Remove Field arrow ▪

Removes the selected field in the Indexed Fields list and places it in the Fields list.

Clear All

Removes all fields in the Indexed Fields list and places them in the Fields list.

Change Order

To move a field in the Indexed Fields, select the field and use the Change Order arrows to move it up or down. These arrows become available when two or more fields are in the Indexed Fields. Change the order of the fields to change the sort order of the index.

Index Options

Choose how you want Database Desktop to treat your secondary indexes.

Unique

Specifies whether more than one record can have the same value in the secondary index fields. If Unique is checked and Database Desktop encounters a duplicate value, the index isn't applied and a warning message appears. You can edit the field data and try indexing again after the duplicate value has been removed.

Descending

Specifies whether the secondary index is ascending or descending.

Case Sensitive

Specifies whether to pay attention to capitalization in sorting.

▪ When checked, Case Sensitive sorts the following sets of characters in this order:
`Abcd, aBcd, aaaa`

▪ When Case Sensitive is unchecked, the following sets of characters are sorted in this order:
`aaaa, Abcd, aBcd`

Values such as `Abcd` and `aBcd` are considered duplicates in a case-insensitive index. They appear in the order in which they were entered in the table.

Database Desktop automatically names single-field, case-sensitive indexes with the field's name. You must name a case-insensitive index when you save it. This enables you to create two indexes on the same field, one case-sensitive and one case-insensitive.

Maintained
Specifies whether to maintain the secondary index automatically.

- Maintained indexes are updated by Database Desktop every time the table changes.
  This speeds up certain operations like queries.
  Maintained indexes are available for keyed tables only.
- Non-maintained indexes are updated only when the index is used; for example, when you link tables or run a query.

The operation that uses the secondary index takes slightly longer using a non-maintained index, because Database Desktop must first update the index to recognize values that you have added, deleted, or changed, and then sort the table according to the new index. Also, if a non-maintained index becomes out of date, you cannot use it to change the viewing order of records.

Non-maintained indexes are most useful on tables that are read-only.

Delete dialog box

See also

Use the Delete dialog box to delete a file from disk. You can delete tables, queries, SQL files, and text files from within Database Desktop.

To open this dialog box, choose Tools|Utilities|Delete.

Warning: Be careful when deleting objects! You cannot undo a deletion. Make sure the table is not used in any associated objects like queries. Associated documents are not deleted when you delete the table; you must delete them yourself.

Dialog box options

Look In

By default, Database Desktop displays the working directory. To choose another directory, use this drop-down list to browse until you reach the directory you want. All files of the selected type in that directory appear in the list below the Look In drop-down list.

If the directory you want has an alias, choose it in the Alias drop-down list. The name of the directory appears in the Look In drop-down list box and its files appear in the file list.

Choose any of the icons to navigate, create a folder, or change the display of folders.

File Name

Type the name of the file to delete or select one from the list box below the Look In drop-down list. You don't need to type an extension; Database Desktop recognizes the type of file you want based on the type shown in the Files Of Type drop-down list.

Files Of Type

Displays the types of files you want to delete.

Alias

If the directory you want has an alias, you can select it in the Alias drop-down list. The name of the selected directory appears in the Look In drop-down list and the files in that directory appear in the file list.

When you choose Delete, Database Desktop displays a message asking you to confirm the deletion of each object. Choose Yes to delete the object, or No to cancel the operation.

▪

Directory Browser

See also

Use the Directory Browser to select a directory.

Dialog box options

Directories and Drive (Or Alias)

Select a drive or alias from the Drive (Or Alias) drop-down list, then select a directory from the Directories list box.

Colors page (Editor Preferences dialog box)

See also

Use the Colors page of the Editor Preferences dialog box to specify how you want the different elements of your code to appear in an SQL Editor window. You can specify both colors and text attributes.

Dialog box options

Elements

Select a code element whose color or text attributes you want to specify.

Color

To specify colors, use the color grid to set the foreground (FG) and background (BG) colors for the element.

To select colors using the mouse,

1. Click a color to select it as the foreground color. (FG appears in the colored box.)
2. Right-click a color to select it as the background color. (BG appears in the colored box.)

To select colors using the keyboard,

1. Use the arrow keys to select a color.
2. Press F to set it as the foreground color, or B to set it as the background color.

Use Default

If a Use Default check box is checked, the SQL Editor will use your Windows system colors for whatever is checked (the foreground, the background, or both) to display a code element. Unchecking either option restores the color you selected previously or, if no color has been previously selected, sets the code element to the Windows system color.

Note: To change the Windows system colors, use Control Panel (under Settings on the Windows Start menu).

Text Attributes

If you want a code element to appear in bold, italic, or underlined, select it in the Elements list, and then check the attribute(s) you want.

Display page (Editor Preferences dialog box)

See also

Use the Display page of the Editor Preferences dialog box to set various SQL Editor display preferences.

Dialog box options

Keystroke Mapping

Use this drop-down list to select from three sets of keystrokes:

- The SQL Editor's default
- uses key bindings that match CUA mappings (with some WordStar additions). This is closest to the keystroke mappings in the Database Desktop 5.0 Editor.
- BRIEF
- uses key bindings that match most of the standard BRIEF keystrokes.
- Epsilon
- uses key bindings that match a large part of the Epsilon editor.

Note: When you are in an SQL Editor window, you can press Shift+F1 to see the keystrokes for the current keymap selection.

Options

You can also set the following preferences by checking the check box beside the option:

Prompt To Save
The SQL Editor will prompt you to save changes when you close the SQL Editor window.

BRIEF Cursor Shapes
Uses an underline instead of a vertical cursor in insert mode.

Show Sidebar
Shows the sidebar with breakpoint symbols.

Custom Size
Opens the next SQL Editor window to the size of the active SQL Editor window (if one is open), or the size of the last SQL Editor window open.

Hints On Status Bar
Shows Toolbar help messages on the status bar. When unchecked, displays only SQL Editor messages.

Font

Use the Name and Size drop-down lists in the Font panel to choose a font name and size. The SQL Editor uses only monospaced screen fonts, such as Courier. A sample of what you choose appears in the Sample box.

Editor page (Editor Preferences dialog box)

See also

Use the Editor page of the Editor Preferences dialog box to customize the behavior of the SQL Editor.

Option When selected

Option	When selected
Auto Indent	Indents the next line to the indent of the current line, when you press Enter.
Backspace Outdents	Aligns the insertion point to the previous indent level (outdents it) when you press Backspace, if the insertion point is on the first character of a line.
Insert Mode	Inserts text at the insertion point without overwriting existing text. If Insert Mode is not checked, text is overwritten. (Use the Ins key to toggle Insert Mode in the SQL Editor without changing this setting.)
Use Tab Character	Inserts a true tab character (ASCII 9). If not checked, inserts spaces instead. If Smart Tab is enabled, this option is off.
Cursor Through Tabs	Enables the arrow keys to move uniformly (column by column) through tabs. If this option is not checked, the insertion point jumps several columns when you move it over a tab.
Smart Tab	Indents to the next character of the previous line. Especially useful when you want to create tabular-looking code.
Group Undo	Undoes as a group your last editing command (for example, a typed character or a overstrike) and all preceding commands of the same type, when you choose Edit|Undo. A "group" starts the last time Enter was pressed. If Group Undo is not checked, Edit|Undo undoes a single command or keystroke.
Undo After Save	Enables you to perform an Edit|Undo command after a file has been saved.
Persistent Blocks	Keeps marked blocks selected even when the insertion point is moved, until a new block is selected.
Overwrite Blocks	Replaces a marked block of text with whatever is typed next. If Persistent Blocks is also checked, text you enter is added to the currently selected block.
Cursor Beyond EOF	Lets you move the insertion point beyond the end-of-file character.
Cursor Beyond EOL	Lets you move the insertion point past the last column of the line.

Use Default

Choose this button to set the above options to the default of the current keystroke mapping▪default, BRIEF, or Epsilon. (Select the keyboard mapping you want on the Display page of the Editor Preferences dialog box.)

Tab Size

Use Tab Size to specify the number of columns you want between tab stops.

Block Indent

Use Block Indent to specify how many columns to indent and outdent a block.

Undo Limit

Use Undo Limit to specify the number of undo actions stored before undo information is discarded.

▪

Empty dialog box

See also

Use the Empty dialog box to remove all records from a table.

When you choose Empty, Database Desktop displays a message asking you to confirm the emptying operation for each table. Choose Yes to remove all records from the table or No to cancel the operation.

To open this dialog box, choose Tools|Utilities|Empty.

Dialog box options

Look In

By default, Database Desktop displays the working directory. To choose another directory, use this drop-down list to browse until you reach the directory. All files of the selected type in that directory appear in the list below the Look In drop-down list.

If the directory has an alias, choose it in the Alias drop-down list. The name of the directory appears in the Look In list box and its files appear in the file list.

Choose any of the icons to navigate, create a folder, or change the display of folders.

File Name

Type the name of the file to empty or select one from the list box below the Look In list. You don't need to type an extension; Database Desktop recognizes the type of file based on the type shown in the Files Of Type drop-down list. You can select more than one table. See To select from lists for more information.

Files Of Type

Displays the types of files you can use for the emptying operation you are performing.

Alias

If the directory has an alias, you can select it in the Alias drop-down list. The name of the selected directory appears in the Look In drop-down list and the files in that directory appear in the file list.

▪

Enter Password(s) dialog box

See also

Use the Enter Password(s) dialog box to supply a password when Database Desktop requests it, or to specify passwords to use for your Database Desktop session:

- The Enter Password(s) dialog box appears when you attempt to load or run a query based on a Paradox table that has been password protected. You must enter the password to open the table or run the query.
- Open the Enter Password(s) dialog box to specify whether to use or stop using the passwords for your Database Desktop session. To open the Enter Password(s) dialog box, choose Tools|Utilities|Passwords. The Enter Password(s) dialog box is helpful for users working with protected tables or tables on a network.

The password you enter is added to Database Desktop's password list. The password list contains all passwords that have been entered in the current Database Desktop session. Once you enter a specific password, you can gain access to any table that recognizes that password until you exit Database Desktop.

You define passwords for Paradox tables in the Create Paradox Table dialog box or the Restructure Paradox Table dialog box.

This dialog box is used only for Paradox tables.

Dialog box options

Password

Type the password in the Password text box. Asterisks (*) represent the characters you type.

Add

Adds the password you typed in Password to Database Desktop's memory. When you choose Add, the dialog box remains open so you can enter additional passwords for tables that you intend to open later in the session. Press Add again to add another password.

Remove

Deletes the password from Database Desktop's memory. By default, if you have closed a password-protected table, you can open it again before exiting Database Desktop, without supplying the password again. Selecting Remove, however, requires you to give the password the next time you open the table.

Use Remove when you want to open a new table but have exceeded the maximum number of passwords per session (if this occurs, you will be notified with an error message).

Remove All

Removes all passwords from Database Desktop's memory. This means any table you have opened using a password will again be protected.

Note: Database Desktop releases all passwords when you exit the program. Through the Enter Password(s) dialog box, you can release a password without exiting Database Desktop.

File Browser

See also

Use the File Browser to select files in other directories.

Dialog box options

Directories and Drive (Or Alias)

Select a drive or alias from the Drive (Or Alias) drop-down list, then select a directory from the Directories list box. The list box to the right of these fields displays the files in the directory you select.

File Name

The text box displays the file extension of the type of file you are browsing for. Choose a file from the list box below the text box.

File Types

The type of file you are browsing for. You can choose a file type if there is more than one displayed in the list.

Document has changed. Save it?

You tried to close the active document or query before saving changes you made to the file. Choose Yes to save the file. Choose No to close the file without saving your changes. Choose Cancel to continue working with the file.

▪

Newly created document. Save it?

You tried to close the active document or query before saving the file. Choose Yes to save the file. Choose No to close the file without saving it. Choose Cancel to continue working with the file.

▪
Find dialog box (SQL Editor)

See also

Use the Find dialog box to find strings of text in your SQL statement.

To open this dialog box, choose Search|Find, or press Ctrl+Z.

Dialog box options

Search For
Type the text you want to look for.

Case Sensitive
Check this to search for the text exactly as you typed it, including capitalization.

Backwards
Check this if you want to search backward from the insertion point instead of forward.

Whole Words Only
Check this if you don't want to see partial matches.

Advanced Pattern Match
Check this to use Database Desktop's extended wildcards.

Find
Choose Find to begin searching for the specified value. If found, the matching string is highlighted. You can then press

- Ctrl+A to find the next occurrence of the string
- Ctrl+R to replace the selected text with the replacement value specified in the Find and Replace dialog box. If there is not a replacement value, the selected text is deleted.
- Ctrl+L to replace all occurrences of the string with the replacement value specified in the Find and Replace dialog box.

▪

Find And Replace dialog box (SQL Editor)

See also

Use the Find And Replace dialog box to find and replace text strings in your SQL statement.

Note: If you have used the Find And Replace dialog box in the current session, you can press Ctrl+R while in the Editor to perform a find and replace based on the current values and settings in the Find And Replace dialog box.

To open this dialog box, choose Search|Replace.

Dialog box options

Search For

Type the text you want to look for (or paste it from the Clipboard).

Replace With

Type the text you want to insert (or paste it from the Clipboard).

Case Sensitive

Check this to search for the text exactly as you typed it, including capitalization.

Backwards

Check this if you want to search backward from the insertion point instead of forward.

Font dialog box

Use this dialog box to specify the default system font.

To open this dialog box, choose Change in the Default System Font panel of the General page of the Preferences dialog box.

Dialog box options

Font
Specify the font you want to use by default by typing it in the text box or selecting one from the list box.

Font Style
Specify a style for the font that appears in the Font text box. You can type it in the Font Style text box or select a style from the list box.

Size
Specify a size, in points, for the font that appears in the Font text box. You can type it in the Size text box or select a size from the list box.

Sample
This field displays a sample of the specified font.

Script
Lists the available language scripts for the specified font. Pick the one appropriate for the language your computer is set up for.

Note: You must exit and restart Database Desktop for the system font change to take effect.

Lookup Help dialog box

See also

When you define table lookup for a table with Help And Fill, and then press Ctrl+Space when resting in the lookup field in Edit mode, this dialog box appears.

Its contents vary according to the way you have defined the lookup table and field. It shows the lookup field you are filling plus related fields if you requested them.

Move Help dialog box

See also

Use the Move Help dialog box to move a detail record to a new master record.

To open this dialog box, select the field of a dependent record that has referential integrity, or any field of a record in a detail table, and choose Record|Move Help, or press Ctrl+Shift+Spacebar.

Dialog box options

The entire master table appears in the Move Help dialog box.

Select the new master record from the master table displayed and click OK. The selected detail record is now assigned to the new master record.

Note: Move Help is available only in fields for which a one → many relationship or a referential integrity relationship has been defined.

Open Document dialog box

See also

Use the Open Document dialog box to specify the file you want to open.

To open this dialog box, choose File|Open, then choose a file type.

Dialog box options

Look In

By default, Database Desktop displays the working directory. To choose another directory, use this drop-down list to browse until you reach the directory you want. All files of the selected type in that directory appear in the list below the Look In drop-down list.

If the directory you want has an alias, choose it in the Alias drop-down list. The name of the directory appears in the Look In list box and its files appear in the file list.

Choose any of the icons to navigate, create a folder, or change the display of folders.

File Name

Type the name of the file or select one from the list box below the Look In list. You don't need to type an extension; Database Desktop recognizes the type of file you want based on the type shown in the Files Of Type drop-down list.

Files Of Type

Displays the types of files you can use for the addition operation you are performing.

Alias

If the directory you want has an alias, you can select it in the Alias drop-down list. The name of the selected directory appears in the Look In drop-down list and the files in that directory appear in the file list.

Password Security dialog box

See also

Use the Password Security dialog box, to create passwords to protect your tables from unauthorized access.

To open this dialog box in the Create Paradox Table or Restructure Paradox Table dialog box, choose Password Security from the Table Properties drop-down list, then choose Define.

Dialog box options

Master Password

Type your password in the Master Password text box. Only asterisks appear onscreen. A password can be from 1 to 15 characters and can contain spaces.

Verify Master Password

Type your password again in the Verify Master Password text box. Passwords are case-sensitive. If the two passwords are not identical, an error message prompts you to enter either of them again.

Auxiliary Passwords

Choose this to open the Auxiliary Passwords dialog box. (This button is not enabled until you verify a master password.)

Change

Changes the master password. Choose Change, then enter the new password and verify it.

This button only appears when a master password already exists.

Delete

Deletes the master password. Choose Delete to remove the password.

This button only appears when a master password already exists.

▪

Picture Assistance dialog box

See also

Use the Picture Assistance dialog box to get assistance with pictures you create or with the standard pictures Database Desktop provides.

To open this dialog box, the following:

- In the Create Paradox Table or Restructure Paradox Table dialog box, choose Validity Checks in the Table Properties drop-down list, then choose the Assist button.

Dialog box options

Picture

Using picture string characters, type the picture you want in the Picture text box or the field object you right-clicked. You can fill this text box with the contents of the Sample Pictures drop-down list box. For details, see "Sample Pictures" below.

Verify Syntax

Choose Verify Syntax to ensure Database Desktop can interpret the picture. If the syntax is correct, a message appears telling you the picture is valid.

Restore Original

If you make a mistake, choose Restore Original to return to the picture that was originally in the Picture text box or the field object you right-clicked.

Sample Value

Type a value, then choose Test Value to see if your picture works.

Test Value

Choose Test Value to ensure Database Desktop can interpret the value you typed into the Sample Value Text box. The message area below the button reports the result of the test.

Sample Pictures

Database Desktop provides several standard pictures, available from the Sample Pictures drop-down list of the Picture Assistance dialog box. Click the drop-down arrow to see this list. When you choose one of these pictures, you see an explanation of it in the message area. For example, when you choose the picture "5{#}[-4{#}]," you see the message that this picture is for either a 5-digit or a 9-digit U.S. zip code."

Add To List

Choose Add To List to open the Save Picture dialog box where you can describe your picture and add it to the Sample Pictures drop-down list. The description you type in the Save Picture dialog box will appear in the message area of the Picture Assistance dialog box whenever you select it from the Sample Pictures list.

Delete From List

Choose Delete From List to delete your picture from the Sample Pictures drop-down list.

Use

Choose Use to copy one of the sample pictures to the Picture text box or to the field object you right-clicked.

General page (Preferences dialog box)

See also

Use this dialog box to specify a default system font.

To open this dialog box, choose Edit|Preferences and click the General tab.

Dialog box options

Default System Font

This field displays the default system font, used in text objects. To change the default system font, click the Change button. Database Desktop opens the Font dialog box.

▪

Toolbars page (Preferences dialog box)

See also

Use this dialog box to display other Toolbars instead of or in addition to the standard Toolbar for each Database Desktop window.

To display this dialog box, choose Edit|Preferences and click the Toolbars tab.

Dialog box options

Toolbars

Indicates which Toolbars to display. Certain Toolbars are only available in certain windows.

Standard
When checked, displays the Standard Toolbar for each window.

Global
Displays the Global Toolbar, with a button to open each type of Database Desktop object and save appropriate objects.

Answer page (Query Properties dialog box)

See also

Use the Answer page settings to specify the type of answer a query generates.

This dialog box page is available only when you have created or opened a valid query that could display an Answer table. To open this dialog box in the Query window, choose Query|Properties or click the Query Properties Toolbar button. If you do not see this page in the Query Properties dialog box, either the query is invalid or does not produce an Answer table.

Dialog box options

Query Answer Type

Live Query View
Choose this button if you want the query to generate a live query view. For important information on how to use this option, see About live query views.

Answer Table
Choose this button if you want the query to generate an Answer table.

Table Type

Paradox/dBASE
Choose whether to save the Answer table as a Paradox or dBASE table.

Table Name
Lets you create a new results table instead of ANSWER.DB. Type the new table name in the text box. When you run the query, the result appears in a table with the new name, rather than ANSWER.DB. This named table is permanent; ANSWER.DB is a temporary table that is overwritten each time you run a query.

Or, you can keep the name ANSWER.DB and type another path name in the box. When you save ANSWER.DB to a directory other than your private directory, Database Desktop does not delete it when you exit the program. (**NOTE:** While you can change the path, renaming ANSWER.DB is preferable.)

Caution: If the path you type already contains an Answer table, Database Desktop will overwrite this with no warning when you run the query.

Browse
Displays the Save File As dialog box, where you can choose a path and name for the results table specified in Table Name.

QBE page (Query Properties dialog box)

See also

Use the QBE page settings to specify how to handle queries against remote tables (SQL tables) and whether to generate auxiliary tables for queries that change data.

To open this dialog box page in the Query window, choose Query|Properties or click the Query Properties Toolbar button, then click the QBE tab.

Dialog box options

Queries Against Remote Tables

These options apply only to queries of remote data on SQL servers. Running queries locally is slower, but might be necessary for example, if you're querying joined tables from more than one server.

Query May Be Local Or Remote
Choose this button to make Database Desktop attempt to run the query remotely (as described below). If this fails, Database Desktop runs the query locally (as described below).

Run Query Remotely
Choose this button to make Database Desktop request that the server run the query and send back only the answer data.

Run Query Locally
Choose this button to make Database Desktop run the query locally. This means that Database Desktop

1. Requests all data in queried tables from the server.
2. Runs the query on your computer system.

Auxiliary Table Option

Choose one of the following buttons to affect the way Database Desktop runs queries that change data (INSERT, DELETE, and CHANGETO queries). Creating auxiliary tables is slower than running a query without creating them, but they can be useful if you want to undo a query. For more information, see Setting auxiliary table properties.

Fast Queries (No Auxiliary Tables)
Choose this button to keep Database Desktop from generating auxiliary tables when running queries that change data. Because Database Desktop does not create the Inserted, Deleted, and Changed tables, the query is performed faster.

Generate Auxiliary Tables
Choose this button to make Database Desktop create auxiliary tables when running queries that change data. These tables are stored in your private directory by default.

Sort page (Query Properties dialog box)

See also

Use the Sort page to sort the Answer table before you run a query.

This dialog box page is available only when you have created or opened a valid query that could display an Answer table. To open this dialog box in the Query window, choose Query|Properties or click the Query Properties Toolbar button, then click the Sort tab. If you do not see this page in the Query Properties dialog box, either the query is invalid or does not produce an Answer table.

Dialog box options

Answer Fields
Lists the fields that will appear in the Answer table.

Sort Order
Lists the fields you select to sort by.

Add Field Arrow
Moves selected fields from the Answer Fields list to the Sort Order list. Add the fields in the order to sort the Answer table by. Or use the Change Order arrows to change the order.

Remove Field Arrow
Removes a selected field from the Sort Order list.

Change Order Arrows

Change the order of the fields in the Sort Order list. Select a field, then click the appropriate arrow to move it up or down in the list. Or, drag the fields to the desired position.

Clear All
Removes all fields from the Sort Order list and returns them to the Answer Fields list.

SQL page (Query Properties dialog box)

See also

Use the SQL page settings to specify how to handle queries against remote tables (SQL tables), whether to generate auxiliary tables for queries that change data, and whether to limit data entered in live query views to values that are "legal" for the current query selection conditions.

To open this dialog box page in the SQL Editor window, choose SQL|Properties or click the Query Properties Toolbar button, then click the SQL tab.

Dialog box options

Queries Against Remote Tables

These options apply only to queries of remote data (tables that you access using Borland SQL Link). Running queries locally is slower, but might be necessary▪for example, if you're querying joined tables from more than one server.

Query May Be Local Or Remote
Choose this button to make Database Desktop attempt to run the query remotely (as described below). If this fails, Database Desktop runs the query locally (as described below).

Run Query Remotely
Choose this button to make Database Desktop request that the server run the query and send back only the answer data.

Run Query Locally
Choose this radio button to make Database Desktop run the query locally. This means that Database Desktop

1. Requests all data in queried tables from the server.
2. Runs the query on your Desktop system.

Auxiliary Table Option

Choose one of the following buttons to affect the way Database Desktop runs queries that change data (INSERT, DELETE, and CHANGETO queries).

Fast Queries (No Auxiliary Tables)
Choose this radio to make Database Desktop not create auxiliary tables when running queries that change data. Because Database Desktop does not create the Inserted, Deleted, and Changed tables, the query is performed faster.

Generate Auxiliary Tables
Choose this radio button to make Database Desktop create auxiliary tables when running queries that change data.

Constraints

Controls the update of SQL tables from live query views.

Constrained Updates
When checked, you are only able to update an SQL table in a live query view with values that meet the selection conditions for the query; you are unable to enter any values that don't match the query conditions.

Structure page (Query Properties dialog box)

See also

Use the Structure page to determine the order of fields in the Answer table before you run a query.

This dialog box is available only when you have created or opened a valid query that could display an Answer table. To open this dialog box page in the Query window, choose Query|Properties or click the Query Properties Toolbar button, then click the Structure tab. If you do not see this page in the Query Properties dialog box, either the query is invalid or does not produce an Answer table.

Dialog box options

Answer Fields

Lists the fields that will appear in the Answer table.

Change Order Arrows

Change the order of the fields in the Sort Order list. Select a field, then click the appropriate arrow to move it up or down in the list. Or, drag the fields up and down to reorder them.

Undo

Restores the previous order in the Answer Fields list.

Referential Integrity dialog box

See also

Use the Referential Integrity dialog box to define a referential relationship between two tables. First select a field from the table you are creating or restructuring (the child table), then select a table containing all valid values for your selected field (the parent table).

For a list of referential integrity tasks, click See also.

To open this dialog box, choose Referential Integrity from the Table Properties list of the Create Paradox Table or the Restructure Paradox Table dialog box, and then choose the Define button.

Dialog box options

Fields

The Fields list displays all the fields from the referential integrity child table. Memo, formatted memo, graphic, binary, OLE, logical, autoincrement, BCD and bytes fields are dimmed in the Fields list, because you cannot create referential integrity from these field types.

Add Field arrow

Choose the referential integrity child field; then click the Add Field arrow or press Alt+A. The field appears in the Child Fields box.

If you choose a field that is not the same logical type as the parent's key field, Database Desktop displays a message on the status bar, and doesn't add the field. In most cases, this means the field types must be identical; however, autoincrement and long integer are of the same logical type.

Child Fields

Displays the fields you select from the Fields list.

Parent's Key

Displays the fields in the referential integrity parent's key.

Add Field arrow

Choose the referential integrity parent table in the Table list; then click the Add Field arrow. The parent table's key appears in the Parent's Key box.

You must choose a table with a key that is the same logical type as the field in the Child Fields box.

Table

Database Desktop displays tables from the working directory in the Table list. Choose the referential integrity parent table and click the Add Field arrow.

Remove Field arrow

To remove a field from the diagram, select the field, then click the Remove Field arrow or press Alt+R.

Update Rule

Database Desktop provides two update rules for tables that use referential integrity. Update rules specify what happens when a user tries to update or delete data in a parent table that has dependent records in a child table.

Prohibit

Prohibit specifies that you cannot change or delete a value in the parent's key if there are records that match the value in the child table.

For example, if the value 1356 exists in the Customer No field of Orders, you cannot change that value in the Customer No field of Customer. (You can change it in Customer only if you first delete or change all records in Orders that contain it). If, however, the value doesn't exist in any records of the child table, you can change the parent table.

Cascade

Cascade specifies that any change you make to the value in the key of the parent table is automatically made in the child table. If you delete a value in the key of the parent table, dependent records in the child table are also deleted. Cascade is the default update rule.

To cascade an update across tables, Database Desktop must place a lock on the target table. If the lock is denied (because another user has already placed a lock), Database Desktop cannot perform the cascade update.

If you are working with SQL tables, the availability of cascading updates and deletes varies according to the table type and software version.

Strict Referential Integrity

Choose Strict Referential Integrity to protect your data from being corrupted by earlier versions of Database Desktop (the default). Strict Referential Integrity specifies that Paradox for DOS cannot access a table on which you've defined referential integrity.

Suppose you use a version of Paradox for DOS to open a Paradox for Windows table that uses referential integrity. You could add data that violates the referential integrity, because the version of Paradox you're using doesn't recognize the referential integrity. To prevent versions of Paradox for DOS from opening the table, check the Strict Referential Integrity check box.

Rename dialog box (tables)

See also

Use the Rename dialog box to give the table you are viewing a different name. This dialog will also rename the associated files such as the indexes.

To open this dialog box, choose Table|Rename in a Table window.

Dialog box options

From

Shows the name of the table to be renamed.

To

Type the name you want to give the table.

When you click OK, Database Desktop renames the table.

Rename dialog box (objects)

See also

Use the Rename dialog box to give a file a different name. You can rename tables, queries, and SQL files, from within Database Desktop.

Note: Do not try to rename tables using the DOS RENAME command or the Windows Explorer.

To open this dialog box, choose Tools|Utilities|Rename.

Dialog box options

Look In

By default, Database Desktop displays the working directory. To choose another directory, use this drop-down list to browse until you reach the directory you want. All files of the selected type in that directory appear in the list below the Look In drop-down list.

If the directory you want has an alias, choose it in the Alias drop-down list. The name of the directory appears in the Look In list box and its files appear in the file list.

Choose any of the icons to navigate, create a folder, or change the display of folders.

File Name

Type the name of the file or select one from the list box below the Look In list. You don't need to type an extension; Database Desktop recognizes the type of file you want based on the type shown in the Files Of Type drop-down list.

Files Of Type

Displays the types of files you can rename.

Alias

If the directory you want has an alias, you can select it in the Alias drop-down list. The name of the selected directory appears in the Look In drop-down list and the files in that directory appear in the file list.

Rename <file name> To dialog box

See also

Use this dialog box to specify a file name and directory for the destination file in a file renaming operation.

To display this dialog box, right-click a file name in the Project Viewer and choose Rename or use Tools|Utilities|Rename and choose OK in the Rename dialog box.

Dialog box options

Save In

By default, Database Desktop displays the working directory. To choose another directory, use this drop-down list to browse until you reach the directory where you want to save the file. All files of the selected type in that directory appear in the file list below the Save In drop-down list.

If the directory you want has an alias, choose it in the Alias drop-down list. The name of the directory appears in the Save In drop-down list and its files appear in the graphics list.

Choose any of the icons to navigate, create a folder, or change the display of folders.

File Name

Type the name of the file to save or select one from the list box below the Save In list. You don't need to type an extension; Database Desktop recognizes the type of file you want based on the type shown in the Save As Type drop-down list.

Save As Type

Displays the types of files you can save.

Alias

If the directory you want has an alias, you can select it in the Alias drop-down list. The name of the selected directory appears in the Save In drop-down list and the tables in that directory appear in the file list.

▪

Restructure dBASE Table dialog box

See also

Use the Restructure dBASE Table dialog box to specify the structure of a dBASE table.

This dialog box has two main panels: Field Roster and Table Properties. You can move between them using the keyboard: Use the Super Tab key (F4) to move from Field Roster to Table Properties; to return, press Shift+Tab.

To open this dialog box, do one of the following:

- From the Table window, choose Table|Restructure.
- From the Desktop, choose Tools|Utilities|Restructure, then specify the table name in the Select File dialog box.

Dialog box options

Field Roster

Use the Field Roster to specify the fields of a table. You can add, delete, or rename fields, and change field types and sizes.

- To insert a field between two existing fields in the Field Roster, select a field and press Ins. Database Desktop opens a blank row above the selected field.
- To delete a field from the Field Roster, select it and press Ctrl+Del. Database Desktop deletes the entire row.

The order in which fields are listed in the Field Roster is the order in which the fields appear in the table. To change the field order, click the row number of the field and drag it to a new position.

Field Name
Specifies the name of the field. See Rules for dBASE field names for more information. This is a required field.

Type
Specifies the type of the field. Right-click the Type column or press the Spacebar to display a list of field types. See dBASE field types and sizes for more information. This is a required field.

Size
Specifies the size of the field. See dBASE field types and sizes for more information. This is a required field for certain field types.

Dec
Specifies the number of decimal places for number or float fields.

Table Properties

In the Table Properties panel you can specify the following:

Indexes
Creates an index on the current field in the field roster. See About dBASE indexes for more information.

Choose Indexes, then choose Define to open the Define Index dialog box.

Table Language
Specifies the language driver. See About table language drivers for more information.

Choose Table Language, then choose Modify to open the Table Language dialog box.

Pack Table

When Pack Table is checked, Database Desktop deletes records that you have marked for deletion.

Record Lock

Contains information about records locked by other users.

Info Size

Specifies whether to keep track of record locking information in a multiuser environment. When you check Info Size, Database Desktop adds a hidden field to the table that shows when a record was locked and by whom.

The amount of information you see when you encounter a locked field depends on the Info Size you specify. The default size is 16 characters. You can choose a size from 8 to 24 from the Info Size drop-down list box. See dBASE Record Lock fields for more information.

Save

Overwrites the old structure with the new structure. If the restructure could cause data loss, the Restructure Warning dialog box opens, where you can tell Database Desktop what to do about the problem.

Save As

Saves the table you are creating and closes the Restructure dBASE Table dialog box. Choose Save As to open the Save Table As dialog box, where you type a name for your new table. You can save the table in the current directory or another one.

▪

Restructure INFORMIX Table dialog box

See also

Use the Restructure INFORMIX Table dialog box to modify the indexes of an existing Informix table.

To open this dialog box, choose one of the following:

- Table|Restructure in a Table window, to restructure the current table
- Tools|Utilities|Restructure, then specify the table name in the Select File dialog box

Dialog box options

Field Roster

The Field Roster panel (on the left), specifies the fields of a table. You cannot modify fields of SQL tables.

Required Field

The Required Field check box (in the panel on the right) specifies whether the selected field is required. You cannot change whether a field of an SQL table is required.

List of Indexes

The panel on the right lists existing indexes for the table. You can add indexes, modify existing indexes, and erase indexes.

Define Index	Choose Define Index to create an index. Database Desktop opens the Define Index dialog box.
Modify Index	Choose Modify Index to change the selected index. Database Desktop opens the Define Index dialog box.
Erase Index	Choose Erase Index to remove the selected index. Database Desktop erases the index.

▪
Restructure INTRBASE Table dialog box

See also

Use the Restructure INTRBASE Table dialog box to modify the indexes of an existing InterBase table.

To open this dialog box, choose one of the following:

- Table|Restructure in a Table window, to restructure the current table
- Tools|Utilities|Restructure, then specify the table name in the Select File dialog box

Dialog box options

Field Roster

The Field Roster panel (on the left), specifies the fields of a table. You cannot modify fields of SQL tables.

Required Field

The Required Field check box (in the panel on the right) specifies whether the selected field is required. You cannot change whether a field of an SQL table is required.

List of Indexes

The panel on the right lists existing indexes for the table. You can add indexes, modify existing indexes, and erase indexes.

Define Index	Choose Define Index to create an index. Database Desktop opens the Define Index dialog box.
Modify Index	Choose Modify Index to change the selected index. Database Desktop opens the Define Index dialog box.
Erase Index	Choose Erase Index to remove the selected index. Database Desktop erases the index.

▪

Restructure ORACLE Table dialog box

See also

Use the Restructure ORACLE Table dialog box to modify the indexes of an existing Oracle table.

To open this dialog box, choose one of the following:

- Table|Restructure in a Table window, to restructure the current table
- Tools|Utilities|Restructure, then specify the table name in the Select File dialog box

Dialog box options

Field Roster

The Field Roster panel (on the left), specifies the fields of a table. You cannot modify fields of SQL tables.

Required Field

The Required Field check box (in the panel on the right) specifies whether the selected field is required. You cannot change whether a field of an SQL table is required.

List of Indexes

The panel on the right lists existing indexes for the table. You can add indexes, modify existing indexes, and erase indexes.

Define Index	Choose Define Index to create an index. Database Desktop opens the Define Index dialog box.
Modify Index	Choose Modify Index to change the selected index. Database Desktop opens the Define Index dialog box.
Erase Index	Choose Erase Index to remove the selected index. Database Desktop erases the index.

▪

Restructure Paradox Table dialog box

See also

Use the Restructure Paradox Table dialog box to specify the structure of a Paradox table.

This dialog box has two main panels: Field Roster and Table Properties. You can move between them using the keyboard: Use the Super Tab key (F4) to move from Field Roster to Table Properties; to return, press Shift+Tab.

To open this dialog box, do one of the following:

- From the Table window, choose Table|Restructure.
- From the Desktop, choose Tools|Utilities|Restructure, then specify the table name in the Select File dialog box.

Dialog box options

Field Roster

Use the Field Roster to specify the fields of a table. You can add, delete, or rename fields, and change field types and sizes.

- To insert a field between two existing fields in the Field Roster, select a field and press Ins. Database Desktop opens a blank row above the selected field.
- To delete a field from the Field Roster, select it and press Ctrl+Del. Database Desktop deletes the entire row.

The order in which fields are listed in the Field Roster is the order in which the fields appear in the table. To change the field order, click the row number of the field and drag it to a new position.

Field Name
Specifies the name of the field. See Rules for Paradox field names for more information. This is a required field.

Type
Specifies the type of the field. Right-click the Type column or press the Spacebar to display a list of field types. See Paradox field types and sizes for more information. This is a required field.

Size
Specifies the size of the field. See Paradox field types and sizes for more information. This is a required field for certain field types.

Key
Specifies whether the field is a key field. The table type determines rules for Paradox table key fields. See About primary indexes (key fields) for more information.

Table Properties

In the Table Properties panel you can specify the following:

Validity Checks
Specifies requirements and defaults for a field. You must have a valid entry selected in the Field Roster area to specify validity check information. For more information about validity checks, see About validity checks.

You can specify the following types of validity checks:

- **Required Field**: Specifies that the selected field in the Field Roster is a required field. When a field is required, you must enter a value in the field for every record in the table. See About required fields for more information.
- **Minimum**: Specifies a minimum value for the selected field in the Field Roster. When a field has a minimum validity check, the values entered in the field must be greater than or equal to the minimum you specify here. See About minimum and maximum values for more information.
- **Maximum**: Specifies a maximum value for the selected field in the Field Roster. When a field has

a maximum validity check, the values entered in the field must be less than or equal to the maximum you specify here. See About minimum and maximum values for more information.

- **Default**: Specifies a default value for the selected field in the Field Roster. When a field has a default validity check, Database Desktop enters the value you specify here if you do not enter another value when you edit this field. See About default values for more information.
- **Picture**: Restricts the types of information you can enter in a field. When a field has a picture validity check, You specify a character string as a template for the values that can be entered into this field. See About pictures for more information.
- **Assist**: Opens the Picture Assistance dialog box, where you can select or modify a predefined string to use as a picture.

Table Lookup
Specifies a lookup table for the current field in the Field Roster. A lookup table is another table that contains values that are valid for the current field. See About table lookups for more information.

Choose Table Lookup, then choose Define to open the Table Lookup dialog box.

Secondary Indexes
Creates a secondary index on the current field in the field roster. A secondary index lets you sort data in an order different from the key field, and lets you form links between tables. See About secondary indexes for more information.

Choose Secondary Indexes, then choose Define to open the Define Secondary Index dialog box.

Referential Integrity
Creates a referential integrity relationship between a field or group of fields and the key field in another table. A referential integrity relationship ensures that ties between like data in separate tables cannot be broken. See About referential integrity for more information.

Chose Referential Integrity, then choose Define to open the Referential Integrity dialog box.

Password Security
Creates passwords to protect your tables from unauthorized access. See About password security for more information.

Choose Password Security, then choose Define to open the Password Security dialog box.

Table Language
Specifies the language driver. See About table language drivers for more information.

Choose Table Language, then choose Modify to open the Table Language dialog box.

Dependent Tables
Displays all tables that depend on the current table for referential integrity.

Pack Table

Check to reuse disk space left over from deleting records. Some restructure operations automatically pack the table. Check this check box to be sure that Database Desktop packs the table when you choose Save or Save As.

Save

Overwrites the old structure with the new structure. If the restructure could cause data loss, the Restructure Warning dialog box opens, where you can tell Database Desktop what to do about the problem.

Save As

Saves the table you are creating and closes the Restructure Paradox Table dialog box. Choose Save As to open the Save Table As dialog box, where you type a name for your new table. You can save the table in the current directory or another one.

▪

Restructure SYBASE Table dialog box

See also

Use the Restructure SYBASE Table dialog box to modify the indexes of an existing Sybase table.

To open this dialog box, choose one of the following:

- Table|Restructure in a Table window, to restructure the current table
- Tools|Utilities|Restructure, then specify the table name in the Select File dialog box

Dialog box options

Field Roster

The Field Roster panel (on the left), specifies the fields of a table. You cannot modify fields of SQL tables.

Required Field

The Required Field check box (in the panel on the right) specifies whether the selected field is required. You cannot change whether a field of an SQL table is required.

List of Indexes

The panel on the right lists existing indexes for the table. You can add indexes, modify existing indexes, and erase indexes.

Define Index	Choose Define Index to create an index. Database Desktop opens the Define Index dialog box.
Modify Index	Choose Modify Index to change the selected index. Database Desktop opens the Define Index dialog box.
Erase Index	Choose Erase Index to remove the selected index. Database Desktop erases the index.

▪

Restructure Table dialog box (other SQL)

See also

Use the Restructure Table dialog box to modify the indexes of an existing SQL table.

To open this dialog box, choose one of the following:

- Table|Restructure in a Table window, to restructure the current table
- Tools|Utilities|Restructure, then specify the table name in the Select File dialog box

Dialog box options

Field Roster

The Field Roster panel (on the left), specifies the fields of a table. You cannot modify fields of SQL tables.

Required Field

The Required Field check box (in the panel on the right) specifies whether the selected field is required. You cannot change whether a field of an SQL table is required.

List of Indexes

The panel on the right lists existing indexes for the table. You can add indexes, modify existing indexes, and erase indexes.

Define Index	Choose Define Index to create an index. Database Desktop opens the Define Index dialog box.
Modify Index	Choose Modify Index to change the selected index. Database Desktop opens the Define Index dialog box.
Erase Index	Choose Erase Index to remove the selected index. Database Desktop erases the index.

▪

Restructure Warning dialog box

See also

When you restructure a table, you often make changes that could result in a loss of data. Changes such as shortening field sizes, creating validity checks, or changing field types can cause existing data to become invalid. Whenever this happens, Database Desktop opens the Restructure Warning dialog box when you leave the Restructure Table dialog box. Choose any of the following options to answer that data handling question the same way for every restructured field. Otherwise, you will be asked to answer these questions repetitively.

Dialog Box Options

Field Trim

Choose how Database Desktop treats data in fields.

Trim All Fields
Truncates all data that does not fit in the new field, without asking for confirmation on each field.

Trim No Fields
Extracts all records containing data that exceed the new maximum length of the shortened field, and saves these records in a Problems table.

Skip confirmation for each deleted field

When this is checked, Database Desktop deletes fields without asking for confirmation for each one.

Validity Checks

Check this, then choose whether or not to apply validity checks to existing data:

Apply to Existing Data
When this is checked, any existing data that does not meet the conditions of new validity checks is written to the Keyviol table. You can change the records in Keyviol, and then add them back to the table using Tools|Utilities|Add. (Note: Database Desktop does not apply a Picture validity check to existing data.)

Do not apply
When this is checked, Database Desktop does not enforce the new validity checks on existing data.

▪

Save File As dialog box

See also

Use the Save File As dialog box to save a file under another name or to copy data to a file.

Dialog box options

Save In

By default, Database Desktop displays the working directory. To choose another directory, use this drop-down list to browse until you reach the directory. All files of the selected type in that directory appear in the file list below the Save In drop-down list.

If the directory has an alias, choose it in the Alias list box. The name of the directory appears in the Look In drop-down list and its files appear in the file list.

Choose any of the icons to navigate, create a folder, or change the display of folders.

File Name

Type the name of the file to save or select one from the list box below the Save In list. You don't need to type an extension; Database Desktop recognizes the type of file based on the type shown in the Save As Type drop-down list.

Save As Type

Displays the types of files you can save.

Alias

If the directory has an alias, you can select it in the Alias drop-down list. The name of the selected directory appears in the Save In drop-down list and the files in that directory appear in the file list.

Save Index As dialog box (dBASE tables)

See also

Use the Save Index As dialog box to specify a file name or tag name for your dBASE table index.

To open this dialog box, choose OK in the Define Index dialog box.

Dialog box options

Index File Name

If you have specified a nonmaintained index, the Index File Name text box is available. If you have specified a single-field index, Database Desktop enters the field's name as the file name. If you have specified an expression index, enter the name to assign to it. Database Desktop saves the index with the .NDX extension.

Index Tag Name

If you have specified a maintained index, the Index Tag Name text box is available. Enter a name to give the index. This name appears in the Create Table (or Restructure Table) dialog box below the Define button. Database Desktop creates a file using the table's name and the .MDX extension to store all maintained indexes.

Save Index As dialog box (Paradox tables)

See also

Use the Save Index As dialog box to name and save a composite or case-insensitive secondary index you have constructed in the Define Secondary Index dialog box.

To open this dialog box, choose OK in the Define Secondary Index dialog box after defining a composite or case-insensitive secondary index.

The Save Index As dialog box appears only when you create an index that is not case-sensitive or not based on a single field. You cannot give such an index the name of a field as its name.

Dialog box options

Index Name

The name you type in this dialog box appears only in the Secondary Index list in the Create Table or Restructure Table dialog box. A secondary index name can be up to 25 characters and include any printable character except: tab, carriage return, line feed, , comma, *, >, <, =, [ ], |, +, ?, : (colon), and \.

Database Desktop automatically names single-field, case-sensitive indexes with the field's name and warns you if you are overwriting an existing index.

Save Index as dialog box (SQL tables)

See also

Use the Save Index As dialog box to name and save an index.

To open this dialog box, choose OK in the Define Index or Define Secondary Index dialog box after defining the index.

Dialog box options

Index Name

The name of the index.

For SQL tables other than Sybase, Database Desktop supplies the prefix "<table>_" for the index name. This prefixes the index name with the table name to ensure that the index name is unique within the database, as described in Creating indexes on SQL tables.

▪

Save Referential Integrity As dialog box

See also

Use the Save Referential Integrity As dialog box to name and save a relationship you constructed in the Referential Integrity dialog box. Database Desktop saves referential integrity definitions in a file with the table's name and the .VAL file extension when you save the table's structure.

For a list of referential integrity tasks, click See also.

To open this dialog box, choose OK in the Referential Integrity dialog box.

Dialog box options

Referential Integrity Name

Type the name to give the referential integrity relationship. Referential integrity names can be up to 31 printable characters and require no file extension. When you choose OK, the Referential Integrity dialog box is closed, and the referential integrity name appears in the list area below the Define button in the Create Paradox Table (or Restructure Paradox Table) dialog box.

The name you type in this dialog box appears only in the Referential Integrity list in the Create Paradox Table dialog box or the Restructure Paradox Table dialog box. When you complete all restructures, the referential integrity relationship is saved in a .VAL file of the same name as your table in the working directory.

Save Table As dialog box

See also

Use the Save Table As dialog box to save a new table, or to save a restructured table under a new name, leaving your original table intact.

When you restructure a table, you should use Save As when you are not sure what some of the potential problems, key violations, or trimming options might do to your data. If you like the new table, you can delete the old one or use Tools|Utilities|Rename to rename the new table and overwrite the old.

To open this dialog box, choose Save As in the Create Table or Restructure Table dialog box.

Dialog box options

Save In

By default, Database Desktop saves tables to the working directory. To choose another directory, use this list box to browse until you reach the directory. All files of the appropriate type in that directory appear in the table list below the Save In drop-down list.

If the directory has an alias, choose it in the Alias drop-down list. The name of the directory appears in the Save In list box and its files appear in the table list.

Choose any of the icons to navigate, create a folder, or change the display of folders.

File Name

Type the name of the table or select one from the list box below the Save In drop-down list. You don't need to type an extension; Database Desktop recognizes the type of file based on the table type you chose in the Create Table dialog box.

Save As Type

Displays the type of table you are saving.

Alias

If the directory has an alias, you can select it in the Alias drop-down list. The name of the selected directory appears in the Save In drop-down list and the tables in that directory appear in the table list.

Options

Display Table
Check to open the new table after you save it.

Add Data to New Table
Check this to add to the new table as much data from the old structure as is applicable to the new structure. This option is available only when you are restructuring tables.

Select Alias dialog box

See also

Use the Select Alias dialog box to choose an alias for the remote database to send your SQL query to. Database Desktop displays the aliases you created in the Alias Manager dialog box. Select an alias from the drop-down list box and choose OK.

To execute the SQL statement, choose SQL|Run SQL or press F8. You can also click the Run SQL button on the Toolbar .

To create an alias for the remote database, choose Tools|Alias Manager.

▪

Select Borrow Table dialog box

See also

Uses the structure of an existing table as a template for the table you are creating. You can borrow the structure, and then modify it for your new table.

Dialog box options

Look In

By default, Database Desktop looks for a source table in the working directory. To choose another directory, use this list box to browse until you reach the directory. All files of the appropriate type in that directory appear in the table list below the Look In drop-down list.

If the directory has an alias, choose it in the Alias drop-down list. The name of the directory appears in the Look In drop-down list and its files appear in the table list.

Choose any of the icons to navigate, create a folder, or change the display of folders.

File Name

Type the name of the table or select one from the list box below the Look In list. You don't need to type an extension; Database Desktop recognizes the type of file based on the table type you chose to create in the Create Table dialog box.

Files of Type

Displays the type of table structure you can borrow, based on the type of table you specified earlier in the Create Table dialog box.

Alias

If the directory has an alias, you can select it in the Alias drop-down list. The name of the selected directory appears in the Look In drop-down list and the tables in that directory appear in the table list.

Options

The options you can borrow vary according to the type of table you are creating.

- **Paradox**

  Primary Index
  Check to borrow the primary index from the source Paradox table.

  Validity Checks
  Check to borrow the validity checks from the source Paradox table.

  Lookup Table
  Check to borrow the lookup table assignments from fields in the source Paradox table.

  Secondary Indexes
  Check to borrow the secondary indexes from the source Paradox table.

  Referential Integrity
  Check to borrow the referential integrity relationships from fields in the source Paradox table.

- **dBASE**

  Indexes
  Check to borrow the indexes from the source dBASE table.

Select File dialog box

See also

Use the Select File dialog box to specify a file.

This dialog box opens in several situations; for example, if you are pasting from a text file into a memo field.

Dialog box options

Look In

By default, Database Desktop displays the working directory. To choose another directory, use this drop-down list to browse until you reach the directory. All files of the selected type in that directory appear in the list below the Look In drop-down list.

If the directory has an alias, choose it in the Alias drop-down list. The name of the directory appears in the Look In list box and its files appear in the file list.

Choose any of the icons to navigate, create a folder, or change the display of folders.

File Name

Type the name of the file or select one from the list box below the Look In list. You don't need to type an extension; Database Desktop recognizes the type of file based on the type shown in the Files Of Type drop-down list.

Files Of Type

Displays the types of files you can use for the operation you are performing.

Alias

If the directory has an alias, you can select it in the Alias drop-down list. The name of the selected directory appears in the Look In drop-down list and the files in that directory appear in the file list.

▪

Select File dialog box (queries)

See also

Use the Select File dialog box to create new queries or add tables to a query you are editing.

When you first create a new query, you have the option to add individual tables or copy an existing query.

Creating a query

Use this dialog box to create a query if you did one of the following before this dialog box appeared:

- Chose File|New|Query
- Right-clicked the Open Query

Toolbar button and chose New

See To create a query from a table for step-by-step instructions on creating a query.

Adding tables to a query

Use this dialog box to add tables to a query if you did one of the following before this dialog box appeared:

- Chose Edit|Add Table
- Clicked the Add Table
- Toolbar button

See To add tables to a query for step-by-step instructions on adding tables to a query.

Dialog box options

Look In

By default, Database Desktop displays the working directory. To choose another directory, use this drop-down list to browse until you reach the directory. All files of the selected type in that directory appear in the list below the Look In drop-down list. You can use Ctrl+click to select more than one file in the file list.

If the directory has an alias, choose it in the Alias drop-down list. The name of the directory appears in the Look In list box and its files appear in the file list.

Choose any of the icons to navigate, create a folder, or change the display of folders.

File Name

Type the name of the file or select one from the list box below the Look In list. You don't need to type an extension; Database Desktop recognizes the type of file based on the type shown in the Files Of Type drop-down list.

Files Of Type

Displays the types of files you can use for the operation you are performing. You can choose tables or other queries.

Alias

If the directory has an alias, you can select it in the Alias drop-down list. The name of the selected directory appears in the Look In drop-down list and the files in that directory appear in the file list.

Sort Table dialog box

See also

Use the Sort Table dialog box to sort a table.

You cannot sort SQL tables.

To open this dialog box, choose Table|Sort or Tools|Utilities|Sort.

Dialog box options

Fields

Select the fields to add or remove to the Sort Order list. All fields from the table are listed.

Sorted Table

Use these options to specify how to sort a table.

Same Table

The sort overwrites the existing sort order of the table. Keep the following in mind:

- Same Table is available only if you are sorting an unkeyed table. Sorting a keyed table to the same table would conflict with the primary index established by the key, which Database Desktop does not allow.

New Table

The sort creates a new table. Type the name of the new table in the text box that appears when you choose this option.

- If you enter the name of an existing table, Database Desktop prompts you to confirm overwriting the existing table.
- If you overwrite an existing table, you must close all windows that include that table's data before you perform the sort.

Sort Just Selected Fields

When you check this option, Database Desktop sorts only on the fields that appear in the Sort Order list. All the fields of the source table are included in the resulting sorted table, but they are not sorted beyond the fields listed in the Sort Order list.

If two or more records have identical values in these fields, Database Desktop cannot sort those records and places them in the table as a group, but unsorted within the group.

When you do not check this option, Database Desktop performs the sort first on the fields in the Sort Order List, then if there are two or more records with identical values in their sorted fields

on the fields remaining in the Fields List (in the order in which they appear).

Display Sorted Table

Check this option to see the results of the sort immediately. Database Desktop opens the sorted table when you close the dialog box.

Fields

Select the fields to remove or add to the Sort Order:

- Places a selected field on the Sort Order (keyboard shortcut: Alt+A).
- Takes a selected field off the Sort Order (keyboard shortcut: Alt+R)

You do not have to put all the fields from the Fields list in the Sort Order list. Database Desktop adds the rest to the end of the list before performing the sort (unless Sort Just Selected Fields is checked).

A field is dimmed when you add it to the Sort Order list.

Note: Database Desktop cannot sort on BLOB, BCD, logical, or bytes fields. These fields are unavailable in Fields.

Clear All

Removes all fields from the Sort Order list, making those fields available again in the Fields list.

Sort Order
Displays fields to include in the sort.

Sort Direction
Switches between ascending and descending sort order for the selected field in the Sort Order list.

The default sort order is ascending, indicated by a **+** in front of the field name in the Sort Order. To change to descending, double-click the field name or click the Sort Direction button; **+** changes to **-**, indicating descending sort order.

Change Order
Changes the order of the fields in the Sort Order list. To move a selected field up or down in the Sort Order, click the Up arrow ▪ or Down arrow
▪ below the list.

▪

Structure Information dialog box

See also

Use the Structure Information dialog box to get information about a table's structure or to save that structure information to a table.

- For Paradox tables, the Structure Information dialog box shows you validity checks, table lookups, secondary indexes, referential integrity, table language, and dependent tables.
- For dBASE tables, the Structure Information dialog box shows you indexes and table language.
- For SQL tables, the Structure Information dialog box shows you indexes and whether each field is required.

You cannot change the table structure from this dialog box. To change the table structure, choose Table|Restructure or Tools|Utilities|Restructure.

To open this dialog box, do either of the following:

- Choose Tools|Utilities|Info Structure.
- Choose Table|Info Structure in the Table window.

Dialog box options

Field Roster

The table's fields and field types are shown in the Field Roster.

Field Name
Specifies the name of the field.

Type
Specifies the type of the field. See Paradox field types and sizes for more information on Paradox fields. For information on other field types, click the Index button and search for Field Types.

Size
Specifies the size of the field. See Paradox field types and sizes for more information on Paradox fields. For information on other field types, click the Index button and search for Field Types.

Key
Specifies whether the field is a key field. See About primary indexes (key fields) for more information.

Table Properties

Use the Table Properties drop-down list to view information about the table. This list is available only for local tables. If you view a dBASE table, the Table Properties drop-down list shows only the table indexes and table language.

Validity Checks
Shows each field's defined validity checks. Move through the fields in the Field Roster to see each one's validity checks.

Table Lookup
Shows any tables that this table uses as a lookup table.

Secondary Indexes
Shows all the table's secondary indexes.

Referential Integrity
Shows whether this table refers to a parent table for valid data.

Table Language
Shows the table's language driver.

Dependent Tables

Shows any table that uses this table as a parent table for valid data.

Indexes
Shows a dBASE table's indexes.

Required Field
Shows whether this field is required to have a value in every record.

Detail Info

Select an index and choose Detail Info to see information about the index. This option is available only when you choose Secondary Indexes or Indexes in the Table Properties drop-down list.

Save As

Choose Save As to create a table that shows the structure information for the table you are working with. The structure table's fields correspond to the settings in the Structure Information dialog box. The structure table does not include information about secondary indexes, table language, and referential integrity.

▪

Subtract Records In dialog box

See also

Use the Subtract Records In dialog box to subtract the records in one table from those in another. Records are looked up and subtracted from the second table based on key value. The Subtract Records In dialog box indicates the table to subtract from another table.

To open this dialog box, choose Tools|Utilities|Subtract.

Dialog box options

Look In

By default, Database Desktop displays the working directory. To choose another directory, use this drop-down list to browse until you reach the directory. All files of the selected type in that directory appear in the list below the Look In drop-down list.

If the directory has an alias, choose it in the Alias drop-down list. The name of the directory appears in the Look In list box and its files appear in the file list.

Choose any of the icons to navigate, create a folder, or change the display of folders.

File Name

Type the name of the file or select one from the list box below the Look In list. You don't need to type an extension; Database Desktop recognizes the type of file based on the type shown in the Files Of Type drop-down list.

Files Of Type

Displays the types of files you can use for the subtraction operation you are performing.

Alias

If the directory has an alias, you can select it in the Alias drop-down list. The name of the selected directory appears in the Look In drop-down list and the files in that directory appear in the file list.

Subtract Records In <table> From dialog box

See also

Use the Subtract Records In <table> From dialog box to subtract the records in one table from those in another.

To open this dialog box, choose OK in the Subtract Records In dialog box.

Dialog box options

Look In

By default, Database Desktop displays the working directory. To choose another directory, use this drop-down list to browse until you reach the directory. All files of the selected type in that directory appear in the list below the Look In drop-down list.

If the directory has an alias, choose it in the Alias drop-down list. The name of the directory appears in the Look In list box and its files appear in the file list.

Choose any of the icons to navigate, create a folder, or change the display of folders.

File Name

Type the name of the file to subtract records from or select one from the list box below the Look In list. You don't need to type an extension; Database Desktop recognizes the type of file based on the type shown in the Files Of Type drop-down list.

Files Of Type

Displays the types of files you can use for the subtraction operation you are performing.

Alias

If the directory has an alias, you can select it in the Alias drop-down list. The name of the selected directory appears in the Look In drop-down list and the files in that directory appear in the file list.

▪

Table Language dialog box

See also

Use the Table Language dialog box to override the default table language driver you set using the BDE Configuration Utility.

To open this dialog box, choose Table Language in the Table Properties panel of the Create Table or the Restructure Table dialog box, then choose Modify.

Dialog box options

Language

Choose a different language from the Language drop-down list. Each selection corresponds to a different set of language properties, including the available character set, language sort order, and upper/lower case handling.

▪

Table Lookup dialog box

See also

Use the Table Lookup dialog box to specify a lookup table for a field.

To open this dialog box, select the field to define a table lookup for in the Create or Restructure dialog box. In the Table Properties panel, choose Table Lookup and choose Define.

Dialog box options

Fields
Database Desktop displays the fields in your table. Select the field to specify a Table Lookup for, then click the Add arrow ▪ above the list. The field name appears in the Field Name box.

Add Field arrow ▪
Adds the selected field in the Fields list to the Field Name box and replaces the existing field.

Field Name
Shows the field you are specifying the table lookup for. Choose any available field in the Fields list.

Lookup Field
Shows the first field of the table you have specified as the lookup table. Choose any table in the Lookup Table list.

Add Field arrow ▪
Adds the selected table in the Lookup Table list and places the first field of the selected table in the Lookup Field box.

Lookup Table
Database Desktop shows the tables in the current directory. (Use Browse to see tables in other directories.) Select the table to use as the lookup table, then choose the Add arrow ▪ above the list. The name of the first field of that table appears in the Lookup Field box.

Lookup Type
Choose the type of table lookup:

> **Just Current Field**
> Only the current field gets its value from the lookup table, even if the current table and the lookup table have other fields in common.
>
> **All Corresponding Fields**
> All fields of the current table that correspond to fields in the lookup table take their values from the lookup table. Corresponding fields must have identical field names and compatible field types in both tables. Only the first field of the lookup table is used as part of the validity check.

For examples of these types, see:

- Example of using Just Current Field with Fill No Help
- Example of using Just Current Field with Help And Fill
- Example of using All Corresponding Fields with Fill No Help
- Example of using All Corresponding Fields with Help And Fill

Lookup Access
Choose the type of viewing access:

> **Fill No Help**
> You cannot view the lookup table from the table you are entering. You can view the lookup table by opening it in its own window.

Help and Fill
You can view the lookup table from the table you are editing.

Drive (or Alias)
Use the list to choose an alias or your private directory.

Browse
Choose Browse to see files in other directories in the Select File dialog box.

Navigation icons

To move up a level from the selected area, click the Up One Level icon.

To create a new folder in the current directory, click the Create New Folder icon. To name the new folder, type over the default name.

To show only folder names, click the List icon.

To show folder names plus details, click the Details icon.

▪

File | Close

See also

Choose File|Close to close the active child window.

File | Exit

See also

Choose File|Exit to leave Database Desktop and close the application.

If you have a window open that has not been saved, Database Desktop displays a dialog box asking if you want to save it. Choose

Yes	To save the file. Database Desktop opens the Save File As dialog box if you have not yet named and saved the file.
No	To exit without saving changes you made to the file.
Cancel	To close the dialog box and go back to what you were doing in Database Desktop.

▪

File | New

See also

Choose File|New to

- Design a new table or query
- Write a new SQL file

▪ QBE Query

Choose File|New|QBE Query to create a new QBE (Query by Example) query. The Select File dialog box opens, where you can choose the table you want to query. When you choose a table and choose Open, Database Desktop opens a Query window, where you specify your query criteria. Once in the Query window, you can add or remove tables with the Add Table and Remove Table Toolbar buttons. For more information, see Queries.

SQL File

Choose File|New|SQL File to create a new SQL (Structured Query Language) query statement. SQL is the standard language for storing and manipulating data in relational databases. When you choose File|New|SQL File, Database Desktop opens the SQL Editor. You can use SQL locally with Paradox and dBASE tables as well as with remote SQL databases like InterBase and Oracle. For more information, see SQL Editor.

Table

Choose File|New|Table to create a table. The Create Table dialog box opens. Select the table type you want from the list box and choose OK. This takes you to the Create Table dialog box. For more information, see Tables.

File | Open

See also

Choose File|Open to

- Work with a table,
- View or run a query
- View or run an SQL statement (if you have an SQL driver installed)

Shortcut: Click the appropriate Desktop Toolbar button.

QBE Query

Choose File|Open|QBE Query to open a QBE (Query by Example). In the Select File dialog box, type the name of the query you want or select it from the list. Database Desktop opens the file in the Query window.

SQL File

Choose File|Open|SQL File to open an SQL (Structured Query Language) statement file. SQL is the standard language for storing and manipulating data in relational databases. When you choose File|Open|SQL File, Database Desktop opens the SQL Editor. You can use SQL locally with Paradox and dBASE tables as well as with remote SQL databases like InterBase and Oracle. For more information, see SQL Editor.

Table

Choose File|Open|Table to open the Open Document dialog box. Type the table name you want or select it from the list, then choose Open. Database Desktop opens the file in the Table window.

▪

File | Save

See also

Use File|Save periodically to save the changes in your current Database Desktop file to disk.

Database Desktop does **not** prompt you for a file name, once you have named the file.

The application is written to the file you most recently specified using File|Open or File|Save As. You can see the name of the file in the title bar of the child window, or if the child is maximized, inside the Database Desktop window in the title bar of Database Desktop.

Note: Save and Save As are always dimmed in a Table window. This is because

- Database Desktop automatically saves the data you enter as soon as you leave each record.
- You save a table's property changes by choosing Table|Table View Properties|Save from the Table window.
- You use Tools|Utilities|Copy or Tools|Utilities|Rename to copy or rename a table.

▪

File | Save As

See also

Choose File|Save As to specify the file name and path where you want Database Desktop to save your current Database Desktop file to disk in a new file. Use File|Save As to save your changed application in a new file without overwriting the original file.

Database Desktop opens the Save File As dialog box, where you can specify file name and path.

Note: Save and Save As are always dimmed in a Table window. This is because

- Database Desktop automatically saves the data you enter as soon as you leave each record.
- You save a table's property changes by choosing Table|Table View Properties|Save Properties from the Table window.
- You use Tools|Utilities|Copy or Tools|Utilities|Rename to copy or rename a table.

▪

File | Working Directory

See also

Choose File|Working Directory to set your working directory. Setting a working directory is the easiest way to quickly get to a group of tables, queries and other objects that are in the same directory. See About directories and aliases for more information.

Effects of changing your working directory

When you change your working directory,

- Project aliases for that working directory become available, and those for the previous working directory become unavailable.
- If you have any open tables or SQL files, they close automatically. If they have been changed, you are prompted to save them.

Set Working Directory dialog box options

Working Directory

Enter the location (the full path) of your working directory or choose Browse to select another directory from the Directory Browser.

Browse

Choose Browse to look for a directory using the Directory Browser.

Aliases

Choose an alias from the list if you want to change the working directory to a directory that already has an alias.

▪

File | Private Directory

See also

Choose File|Private Directory to open the Set Private Directory dialog box where you can identify a directory to use as your private directory. The directory you identify is where Database Desktop stores any temporary tables you create. This avoids conflicts with any other network user's temporary tables. See About directories and aliases for more information.

Private Directory

Enter the location (the full path) of your private directory or choose Browse to select another directory from the Directory Browser.

Browse

Choose Browse to look for a directory using the Directory Browser.

If you do not specify a private directory, Database Desktop uses the PRIVATE directory, which is installed below your system directory if you install Database Desktop on a local (non-network) drive. If you have no local hard disk, the network home directory on the file server should be used as the private directory. You cannot use a floppy drive as a private directory.

Note: Your private directory must be different from any other user's private or working directory. You can choose :PRIV: (the alias for your private directory) from any Alias list in a dialog box.

Edit | Add Table

See also

Choose Edit|Add Table to add a table to the Query window. When you choose Edit|Add Table, Database Desktop opens the Select File dialog box. Select the table or tables you want to add to the Query window and choose Open.

Shortcut

Toolbar

▪

Edit | Copy

See also

Copies the selected text or objects onto the Clipboard. Edit|Copy does not delete anything from your table or query.

To paste the contents of the Clipboard into your document, use either

- Edit|Paste
- Shift+Ins
- Paste button

The contents of the Clipboard are not deleted when you paste, so you can paste as many times as you want.

Shortcut key: Ctrl+Ins

In queries

Use Edit|Copy on expressions and example elements.

In tables

Edit|Copy is available only in Edit mode.

Edit | Copy To

See also

Choose Edit|Copy To to copy binary, and graphic values to external files. This command isn't activated in Database Desktop.

Edit | Cut

See also

Removes selected text or objects and places them on the Clipboard.

You can then use the Paste button or choose Edit|Paste to paste the contents of the Clipboard into another file or somewhere else in the same file.

The contents of the Clipboard are not deleted when you paste, so you can paste as many times as you want.

To delete a selection without affecting the Clipboard contents, press Del or choose Edit|Delete.

Shortcut key: Shift+Del

In queries

Use Edit|Cut on selection conditions and example elements.

In tables

Edit|Cut is available only in Edit mode.

Edit | Delete

See also

Deletes the selected text or object.

In this window	Edit|Delete removes the selected
Table	Value in the current field(s)
Query	Example element or expression for a field

Shortcut key: Del

▪
Edit | Editor Preferences

See also

Choose Edit|Editor Preferences to customize the SQL Editor, used to create SQL statements.

The SQL Editor Preferences dialog box contains the following pages:

Editor
Display
Colors

▪

Edit | Paste

See also

Inserts information previously put onto the Clipboard by Edit|Cut, Edit|Copy, or other applications.

The effects of Edit|Paste depend on which window is active and whether you are editing or viewing data.

The contents of the Clipboard are not deleted when you paste, so you can paste as many times as you want.

Shortcut key: Shift+Ins

In Table windows

When in Field View, choose Edit|Paste to

- Insert the contents of the Clipboard into a field at the insertion point.
- Replace the selected contents of the current field with the contents of the Clipboard.

In queries

Choose Edit|Paste to insert the contents of the Clipboard into a query.

Edit | Paste From

See also

Choose Edit|Paste From to paste a value from an external file into a selected field. This command isn't activated in Database Desktop.

▪

Edit | Paste Link

See also

Choose Edit|Paste Link to

▪ Create a linked duplicate of data entered through Dynamic Data Exchange (DDE), so that any change you make to the source is automatically made to the duplicate. Paste Link is available for linking data through DDE into a query or into a large alpha field in a table.

Edit | Preferences

See also

Choose Edit|Preferences to set your Database Desktop preferences in the following areas:

General

Toolbars

Note: These settings affect the current and future Database Desktop work sessions and supply default settings for object properties. In general, property settings override preferences.

Edit | Remove Table

See also

Choose Edit|Remove Table to remove one or more tables from the Query window. When you choose Edit|Remove Table, Database Desktop opens the Remove Table dialog box. Select the table you want to remove and choose OK.

You can remove only one table at a time.

Shortcut

Toolbar

Edit | Select All

See also

In Table windows

Choose Select All to select all fields of a table (the entire table). Database Desktop places a box around the table.

In the SQL Editor

Selects all text in the active SQL Editor window.

Edit | Undo

See also

Shortcut key: Alt+Backspace

In Table windows

Choose Edit|Undo to undo all changes to all fields in the current record and unlock the current record. If the current record has not been changed, Edit|Undo does nothing. Because Database Desktop updates data as soon as you move off a record, you must use Undo before you leave the record.

To discard changes to a single field, press Esc before you leave the field. Database Desktop restores the original contents of the field.

Caution: You cannot use Edit|Undo to retrieve a record you have deleted. Once you delete a record in a Paradox table, there is no way to get it back except to enter it again.

In SQL Editor windows

Undoes your last edit.

View | Cascade Tables

See also

Overlaps multiple table query images in the Query window.

Cascading maximizes the amount of information visible in the active query image while still showing the table names of other query images.

▪

View | Field View

See also

Choose View|Field View to toggle in and out of Field View. When your table or query is in Field View, the insertion point is blinking. Whatever you type is entered at the insertion point and does not overwrite the rest of the field.

Field View makes it possible to

- Place the insertion point between characters in an alpha, number, money, date, or other non-BLOB field
- Select part of a field instead of the whole entry
- Use navigation keys (arrows, Home, End, and so on) to move within a field instead of the whole table or query

Persistent Field View

Choose View|Persistent Field View to keep your table or query in Field View.

Shortcuts (Entering Field View)

Toolbar	▪
Keyboard	F2, Ctrl+F
Mouse	Click in the field after it has been selected.

Shortcuts (Leaving Field View)

Toolbar	▪
Keyboard	F2, Ctrl+F
Mouse	Click another field

▪

View | Persistent Field View

See also

Choose View|Persistent Field View to keep your table or query in Field View. When your table or query is in Field View, the insertion point is blinking. Whatever you type is entered at the insertion point and does not overwrite the rest of the field.

Shortcut key: Ctrl+F2

Field View makes it possible to

- Place the insertion point between characters in an alpha, number, money, date, or other non-BLOB field
- Select part of a field instead of the whole entry
- Use navigation keys (arrows, Home, End, and so on) to move within a field instead of the whole table or query

Field View

Choose View|Field View to put your table or query into Field View until you move the insertion point to another field or field object.

View | Tile Tables

See also

Choose View|Tile Tables to arrange multiple table query images in the Query window without overlapping.

Tiling shows you the same amount of each query image at once. This is the default arrangement.

Search | Find (SQL Editor)

See also

Choose Search|Find to find a particular text string (word or phrase) in your SQL statement.

When you choose Search|Find, Database Desktop opens the Find dialog box, where you specify the text to search for, the case sensitivity to use, and other search options.

To search and replace text, choose Search|Replace.

Search | Find Next (SQL Editor)

See also

Moves to the next occurrence of the text you specified in the Find dialog box.

Note: Find Next is dimmed if you have not searched for anything in the active SQL Editor window.

To search and replace text, choose Search|Replace.

Shortcut

Ctrl+A

Search | Replace (SQL Editor)

See also

Choose Search|Replace to search for text and replace it with a value you specify. When you choose Search|Replace, the Find And Replace dialog box appears. Use it to specify the text you are searching for and what you want to replace it with.

Shortcut

Shift+Ctrl+Z

Search | Replace Next (SQL Editor)

See also

Replaces the next occurrence of the text specified in the Find And Replace dialog box.

To replace all occurrences of the text, choose Search|Replace and check Replace All.

Note: Replace Next is dimmed if you have not replaced anything in the active SQL Editor window.

Shortcut

Ctrl + R

Table | Edit Data/View Data

See also

Choose Table|Edit Data to enter data in a table. Choose Table|View Data when you are through entering data.

In Edit mode, records are automatically locked and unlocked as you edit them. This prevents one user from deleting or changing the same record at the same time as another user.

In Edit mode, changes are saved automatically every time you move to another record.

Shortcut key: F9

▪

Table | Info Structure

See also

Use Table|Info Structure to see the structure of the selected table. You can view field types and sizes, as well as key, index, referential integrity, table language driver, password, and lookup information.

When you choose Table|Info Structure, Database Desktop opens the Structure Information dialog box. This dialog box displays, for viewing only, the same information that is in the Restructure Table dialog box.

▪

Table | Notify On

See also

When Database Desktop is the server in a DDE link, Notify On controls when data is sent to the client application.

For example, if a Database Desktop field is linked to a spreadsheet cell through DDE, you have two options:

- When Notify On is checked, the value in the spreadsheet is changed every time a new record is selected in the Database Desktop table.
- When Notify On is not checked, data is sent to the client only if the client requests it.

Table | Restructure

See also

Choose Table|Restructure to change the structure of the selected table. You can change field types and sizes, as well as key, index, referential integrity, table language driver, password, and lookup information.

When you choose Table|Restructure, Database Desktop opens the Restructure Table dialog box for the type of table you have selected.

Shortcut

Toolbar

Table | Table View Properties

See also

Use the commands on the Table View Properties menu to save, restore, or delete the changes you have made to a table view. This menu is available only when a table is open.

Table|Table View Properties|Save

Table|Table View Properties|Restore

Table|Table View Properties|Delete

Table | Table View Properties | Delete

See also

Choose Table|Table View Properties|Delete to delete a Paradox table's .TV file (or a dBASE table's .TVF file). When you delete a table's unique property file, Database Desktop uses default property settings.

Table | Table View Properties | Restore

See also

Choose Table|Table View Properties|Restore to undo any property changes you have made to the Table window since they were last saved. If the properties have never been saved, Database Desktop restores the default view of the table. Database Desktop does not restore data.

Table | Table View Properties | Save

See also

Choose Table|Table View Properties|Save to save all the property changes you have made to a table, including property changes to individual fields. This saves the appearance of the table as you have changed it. Database Desktop saves data as it is entered, so File|Save and File|Save As are not necessary and are dimmed in the Table window.

This command is available only when a table is open.

Database Desktop saves the properties you define for a Paradox table in the .TV file, and the properties you define for a dBASE table in the .TVF file.

If you try to close a Table window without saving property changes, Database Desktop displays a dialog box asking if you want to save your changes.

Tip: If you change properties, then change your mind about them, choose Table|Table View Properties|Restore to restore your previous properties.

Table | Strict Translation

See also

Choose Table|Strict Translation to limit available characters to the DOS character set supported by the table's language driver. These are characters common to both the OEM and ANSI character sets.

When Strict Translation is checked, you cannot move off a field where you have entered a character that is not a member of the table's DOS character set.

When Strict Translation is not checked, you can enter a character not in the set, but when you move off the field, that character changes to a character that does occur in the DOS character set supported by the table's language driver.

It is also possible that a table that has been edited with a DOS application might contain characters not found in the Windows ANSI character set. If you use Database Desktop or Paradox for Windows to edit such a table with Strict Translation checked, a warning is issued whenever you enter Field View (in Edit mode) in a field containing non-ANSI characters. If you leave the field without editing, the characters are not changed; if you edit the field, the characters are converted to ones that are common to both the ANSI and OEM character sets.

Record | Delete

See also

Choose Record|Delete to delete the current record from the table. You must be in Edit mode.

In most types of tables, you cannot retrieve a deleted record, so be sure you want to delete the entire record before you choose Delete.

In a dBASE table, deleting a record does not immediately remove it. You can permanently remove deleted records by restructuring the table and checking Pack Table in the Restructure dBASE Table dialog box.

Shortcut key: Ctrl+Del

Record | Insert

See also

Choose Record|Insert to insert a blank record above the selected record. You can also press Ins.

When you insert a record into a keyed table, then enter a value in it, Database Desktop automatically moves it to its proper position in the table. (The record might move from the place where you inserted it.) Records inserted in non-keyed tables stay where they are inserted.

▪
Record | Lock

See also

Choose Record|Lock to place a lock on the record you are viewing. The Desktop status bar tells you when you have locked a record.

Shortcut key: F5

You do not have to manually lock each record before making changes to it. Database Desktop locks a record automatically when you begin editing it. The message `Record is now locked` appears in the Desktop status bar. Database Desktop removes the lock when you leave the record.

Locking is important if you use Database Desktop in a multiuser environment, or if you run two Database Desktop sessions simultaneously. When a record is locked, other users can view it but cannot edit or delete it.

After you lock a record, the Lock command changes to Unlock. You must unlock records before other users can access them. Choose Record|Unlock or press Shift+F5.

▪

Record | Lookup Help

See also

Choose Record|Lookup Help when you enter data in a field that has a lookup table specified.

When you choose Record|Lookup Help, the lookup table opens in a window where you can choose the value you want. You must be in Edit mode to use Record|Lookup Help.

Shortcut key: Ctrl+Spacebar

Record | Move Help

See also

Choose Record|Move Help to move a detail record to a new master record.

In certain situations, you might have a record in one table that corresponds to a record in another table. This can happen in a referential integrity relationship, where one record in a parent table is related to one or more records in a child table. In this kind of relationship, you can use Move Help to move a dependent record from one master to a different master.

Shortcut key: Ctrl+Shift+Spacebar

▪

Record | Post/Keep Locked

See also

Choose Post/Keep Locked to write your changes to the current record and move the record to its place in a keyed table. Other users can see it, but the record is locked so you can continue editing it.

Use Post/Keep Locked to make sure no key violation occurs before you fill in the rest of the record.

Shortcut key: Ctrl+F5

Record menu

See also

Use the commands on the Record menu to quickly find, insert, delete, or lock records in a table. Record commands are available only when you are viewing data in a table. To use Insert, Delete, Lock, and Post/Keep Locked, you must be in Edit mode.

You can also use the navigation Toolbar buttons to move through records in the table.

Command	Toolbar	Action
Next	▶	Find the next record.
Previous	◀	Find the previous record.
Next Set	▪	Find the next set of records.
Previous Set	◀◀	Find the previous set of records.
First	⏮	Find the first record.
Last	⏭	Find the last record.
Insert		Insert a record.
Delete		Delete a record.
Lock		Lock a record you are editing, then unlock it when you are through.
Post/Keep Locked		Hold a lock on a record even after you have posted (saved) its value.
Lookup Help		Display the lookup table containing valid values for a field that has a table lookup.
Move Help		Move a detail record to a new master record in a referential integrity relationship.

Tools | Alias Manager

See also

Opens the Alias Manager dialog box. Use this dialog box to view, change, or add aliases.

An alias name is a name you give to a database. Once you've named it, the alias name is all you have to type in the future, rather than the entire path plus its original name. The alias name also appears in the Alias list in File|Open dialog boxes.

▪

Tools | Passwords

See also

Choose Tools|Passwords to open the Enter Password(s) dialog box, where you can specify which table passwords to use in the current Database Desktop session. Passwords can be defined for your Paradox tables in the Create Paradox Table or the Restructure Paradox Table dialog boxes.

When you use a password-protected table, the password can either be entered through the Enter Password dialog box or it will be requested when you open the table. Passwords are remembered for later use by Database Desktop, even if you subsequently close the table.

Tools | Utilities

Use the Utilities menu to manage your Database Desktop database.

The Utilities menu contains commands that affect tables. You can add or subtract records from tables, copy, delete, rename, restructure, or sort tables, get information about a table, and empty tables.

You can also copy, delete and rename other Database Desktop objects from the Utilities menu.

The commands available from the Tools|Utilities menu are shown below.

Add
Copy
Delete
Empty
Info Structure
Rename
Sort
Restructure
Subtract

▪

Tools | Utilities | Add

See also

Choose Tools|Utilities|Add to add the records in one table to those in another without having to retype them. Database Desktop opens the Add Records In dialog box.

Note: The two tables must have identical structures, except that

- Number and money fields are interchangeable.
- You can add from an autoincrement field to a long integer field.

Tools | Utilities | Copy

See also

Choose Tools|Utilities|Copy to make a copy of a file. Database Desktop opens the Copy dialog box.

You can copy tables, queries, SQL, and text files from within Database Desktop. When you copy a table, Database Desktop copies both its structure and the data contained in it.

Warning: Always use the Database Desktop Copy command to copy tables. Using the DOS COPY command or the Windows Explorer might not copy all related files that make up a table. For example, the contents of memo fields are stored externally to a table and are not copied by copying the .DB file. A Database Desktop Copy command, however, copies all files and pointers correctly.

Tools | Utilities | Delete

See also

Choose Tools|Utilities|Delete to delete a file from disk. Database Desktop opens the Delete dialog box.

You can delete tables, queries, SQL, and text files from within Database Desktop.

Always use the Database Desktop Delete command to delete tables. Using the DOS DELETE command or the Windows Explorer might not delete all related files that make up a table.

Warning: Be careful when deleting objects! You cannot undo a deletion. Make sure the table is not used in any associated objects like queries.

Tools | Utilities | Empty

See also

Choose Tools|Utilities|Empty to remove all records from a table, leaving the table structure (including all keys, indexes, validity checks, and so on) intact. Database Desktop opens the Empty dialog box.

Tools | Utilities | Info Structure

See also

Choose Tools|Utilities|Info Structure to get information about a table's structure. In the Select File dialog box, choose a table. Database Desktop opens the Structure Information dialog box.

If you want structure information on an open table, choose Table|Info Structure.

The Structure Information dialog box shows you validity checks, table lookup, secondary indexes, referential integrity, table language, and dependent tables.

Note: Depending on the display monitor you have or the way you set colors in the Windows Control Panel, information in the Structure Information dialog box might not be visible on your screen. For example, the contents of Referential Integrity list boxes might be gray on gray, and therefore invisible. If you are missing information, adjust your screen colors using the Windows Control Panel.

You cannot change the table structure from this dialog box. To change a table's structure, choose Tools|Utilities|Restructure or Table|Restructure.

Saving a table's structure to a table

Choose the Save As button to create a table that shows the structure information for the table you are working with. The structure table's fields correspond to the settings in the Structure Information dialog box.

Tools | Utilities | Rename

See also

Choose Tools|Utilities|Rename to give a file a different name. Database Desktop opens the Rename dialog box.

You can rename tables, queries, SQL, and text files from within Database Desktop.

Always use the Database Desktop Rename command to rename tables. Using the DOS Rename command or the Windows Explorer might not rename all related files that make up a table (for example, the files containing table's primary index, secondary indexes, validity checks, or BLOB data). The Database Desktop Rename command, however, renames all files correctly.

Be careful when renaming tables. Queries that refer to a table under one name will not be bound to the table under its new name.

▪

Tools | Utilities | Restructure

See also

Choose Tools|Utilities|Restructure to change the structure of a table. In the Select File dialog box, choose the table to restructure. If you choose, a .DB table, Database Desktop opens the Restructure Paradox Table dialog box, where you can change

- Field names or types
- Which fields are included
- Key fields
- Validity checks
- Lookup fields
- Secondary indexes
- Referential integrity
- Passwords
- Table language character set

If you want to restructure an open table, choose Restructure from the Table menu.

Tools | Utilities | Sort

See also

Choose Tools|Utilities|Sort to sort the data in a table. In the Select File dialog box, choose the table to sort. Database Desktop opens the Sort Table dialog box. You can sort into the same table or a different table.

Note: Sorting to the same table is only available on unkeyed tables.

Tools | Utilities | Subtract

See also

Choose Tools|Utilities|Subtract to remove from one table records that exist in another. Database Desktop opens the Subtract Records In dialog box.

You can subtract records only from a keyed table. Because dBASE does not support Paradox table keys, you cannot subtract records from a dBASE table. Instead, use a DELETE query.

During a subtract operation, Database Desktop removes any record that contains a value in its key fields that exactly matches the corresponding fields of a record in the subtraction table.

Window | Arrange Icons

See also

Choose Window|Arrange Icons to reorder the arrangement of icons on the Desktop.

Windows arranges the icons across the bottom of the Desktop in a straight line, maintaining the same order it found them in, left to right.

▪
Window | Cascade

See also

Choose Window|Cascade to overlap all open windows on the Desktop so only the title bars of inactive windows show.

The titles of all open windows appear on the Windows menu. When you choose a title to activate the window, it moves to the top of the stack.

▪
Window | Close All

See also

Choose Window|Close All to close all open windows on the Desktop. Database Desktop prompts you to save any changes before closing each window.

The titles of all open windows appear on the Windows menu. Click a title to activate its window.

▪

Window | n

See also

Choose the name of the child window you want to make active.

- If the window is hidden behind another it will be brought to the top.
- If the window is minimized, it will be restored or maximized (same as the currently active window) and made active.

The name you see in the list is the same name you will see on the window's title bar.

Window | Tile Side-By-Side

See also

Choose Window|Tile Side-By-Side to fit all open windows side-by-side in the Database Desktop window without overlapping. The currently active window will be the leftmost window.

The titles of all open windows appear on the Windows menu. Click a title to activate its window.

Window | Tile Top And Bottom

See also

Choose Window|Tile Top And Bottom to fit all open windows one above the other in the Database Desktop window without overlapping. The currently active window will be the topmost window.

The titles of all open windows appear on the Windows menu. Click the title to activate its window.

Query | Ignore Changes

See also

Choose Query|**I**gnore Changes to allow other users to make changes to the source table(s) while Database Desktop runs your query and to prevent Database Desktop from restarting the query if they do.

Query | Lock Tables

See also

Choose Query|Lock Tables to lock all tables in your query, preventing any changes to them while Database Desktop runs the query. Database Desktop releases the locks when it finishes running the query. (If someone else is already using the table(s) you want to lock and query, Database Desktop can't place your locks. You'll see a message informing you that a table is locked.

Query | Properties

See also

Choose Query|Properties to specify how you want Database Desktop to run the current query. Database Desktop opens the Query Properties dialog box.

If you have specified a valid query, the Query Properties dialog box contains the following pages:

Answer

QBE

Sort

Structure

▪

Query | Restart On Changes

See also

Choose Query|Restart On Changes to make Database Desktop restart the query when it detects a change to the source table(s).

Query | Run Query

See also

Choose Query|Run Query to run a query. If the query contains no errors, Database Desktop displays a window to tell you the status of the query. After Database Desktop completes the query, depending on the kind of query it is, Database Desktop either displays an Answer table or changes data in a table. See About query results for more information.

Shortcuts

Toolbar	
Keyboard	F8

Query | Show SQL

See also

Choose Query|Show SQL to translate your query to Structured Query Language (SQL) and have the code displayed in the SQL Editor.

Shortcut

Toolbar

Query | Wait For DDE

See also

If Wait For DDE is on, the query refreshes every time the DDE value changes. If Wait For DDE is off, you must explicitly tell Database Desktop when to run the query, and it will take the current DDE value.

SQL | Properties

See also

Choose SQL|Properties to control where the SQL query is performed (local or remote) and how the results are presented.

Database Desktop displays the following pages of the Query Properties dialog box:

- Answer page
- SQL page

You can also click the Query Properties button on the Toolbar.

SQL | Run SQL

See also

Choose SQL|Run SQL to execute the active SQL statement. Database Desktop displays a status window to tell you the status of the query and displays the Answer table or a live query view when the query is successfully completed. If you have not already selected an alias for the remote database, Database Desktop displays the Select Alias dialog box, where you can specify an alias before running the query.

Shortcut key: F8

You can also click the Run SQL button on the Toolbar.

The Answer table is a temporary table. It is overwritten every time you run another query and deleted when you leave Database Desktop. To change the options for the Answer table, choose SQL|Properties. Database Desktop displays the Answer page of the Query Properties dialog box, where you can choose to create an Answer table or a live query view, change the table name, and specify whether to save the table as a Paradox or dBASE table.

To create a new SQL statement, choose File|New|SQL File. To select a different SQL statement to execute, choose File|Open|SQL File.

Alias Manager Dialog Box (SQL Editor)

SQL | Select Alias

See also

Choose SQL|Select Alias to select the alias of the remote database you want to connect to. Database Desktop opens the Select Alias dialog box where you can choose one of the aliases you created in the Alias Manager dialog box.

You can also click the Select Alias button on the Toolbar.

To execute the SQL statement, choose SQL|Run SQL or press F8. You can also click the Run SQL button on the Toolbar .

To create an alias for a remote database, choose Tools|Alias Manager.

▪

Help | About Database Desktop

See also

Choose Help|About Database Desktop to open the About Database Desktop dialog box, which displays the version of Database Desktop that you are using.

Help | User's Guide Topics

See also

Choose Help|User's Guide Topics to open the User's Guide Table of Contents.

Delete (right-click menu)

See also

When you right-click in the left-most field of a query image and choose Delete, this indicates that you want to change a table by deleting data that matches the query. For details, see About DELETE queries.

Global (right-click menu)

See also

Displays the Global Toolbar, with a button to open each type of Database Desktop object and save appropriate objects.

▪

Insert (right-click menu)

See also

When you right-click in the left-most field of a query image and choose Insert, this indicates that you want to change a table by inserting data that matches the query. For details, see About INSERT queries.

▪

Set (right-click menu)

See also

When you right-click in the left-most field of a query image and choose Set, this indicates that you want to perform a SET query on a group of records. For details, see About querying sets of records (SET queries).

▪

Standard (right-click menu)

See also

When checked, displays the Standard Toolbar for each window.

No help available

No help is available for the item you selected. Click to view the Table of Contents of the Database Desktop User's Guide.

▪

Query window

See also Toolbar

You can use the Query window to retrieve information from your tables. For example, you can find out

- Which customers have placed orders this month?
- What is the total amount of all orders placed by each customer?
- What orders have not been paid?

You can also use a query to perform calculations on your data. And you can insert, delete, and change records using INSERT, DELETE, and CHANGETO queries.

The Query window appears when you open a *.QBE file or create a new one.

SQL Editor

See also

Use the SQL Editor to enter, save and execute SQL statements for your server. Choose File|Open|SQL File or File|New|SQL File to open the SQL Editor.

You can also use the SQL Editor to view the SQL statement that a Database Desktop query sends to your server. When you save an SQL statement to your local hard disk, Database Desktop places it in an unformatted text file with an .SQL extension.

Local SQL queries

You can also use the SQL Editor to execute SQL statements against local databases. See About Local SQL for more information.

Table window

See also Toolbar

The Table window appears when you open or create a new Paradox, dBASE, or SQL table in Database Desktop.

You can use the Table window to enter data or restructure tables.

Database Desktop default keymap

See also

If you're using the Database Desktop default keystroke mapping (Editor Preferences dialog box, Display page) the keystrokes in the left column will perform the actions shown on the right.

Left Arrow	Move cursor left one column.
Right Arrow	Move cursor right one column.
Up Arrow	Move cursor one line up.
Down Arrow	Move cursor one line down.
Home	Move cursor to start of line.
End	Move cursor to end of line.
Page Up	Move cursor up one page.
Page Down	Move cursor down one page.
Backspace	Delete character to the left of the cursor.
Delete	Delete character to the right of the cursor. (On the cursor if box-shaped cursor).
Insert	Toggle insert / overstrike mode.
Tab	Inserts a tab, or indents to indent of previous line if smart-tab is turned on.
Ctrl+Left Arrow	Move cursor left one word.
Ctrl+Right Arrow	Move cursor right one word.
Ctrl+Up Arrow	Scroll window up one line.
Ctrl+Down Arrow	Scroll window down one line.
Ctrl+Home	Move cursor to start of editor.
Ctrl+End	Move cursor to end of editor.
Ctrl+Page Up	Move cursor to top of screen.
Ctrl+Page Down	Move cursor to bottom of screen.
Ctrl+Backspace	Delete the word to the left of the cursor.
Ctrl+Delete	Delete the word to the right of the cursor.
Ctrl+Tab	Smart tab, indents to the indent of the previous line.
Shift+Left Arrow	Select from the character to the left of the cursor.
Shift+Right Arrow	Select from character to the right of the cursor.
Shift+Up Arrow	Select from one line up.
Shift+Down Arrow	Select from one line down.
Shift+Home	Select to start of line.
Shift+End	Select to end of line.
Shift+Page Up	Select to previous page.
Shift+Page Down	Select to next page.
Shift+Tab	Move cursor back one tab position.

Ctrl+Shift+Left Arrow	Select from the word to the left of the cursor.
Ctrl+Shift+Right Arrow	Select from the word to the right of the cursor.
Ctrl+Shift+Home	Select from start of editor.
Ctrl+Shift+End	Select from end of editor.
Ctrl+Shift+Page Up	Select from start of screen.
Ctrl+Shift+Page Down	Select from end of screen.
Ctrl+Ins	Copy to Clipboard.
Shift+Del	Cut to Clipboard.
Shift+Ins	Paste from Clipboard.
Ctrl+C	Copy to Clipboard.
Ctrl+X	Cut to Clipboard.
Ctrl+V	Paste from Clipboard.
Alt+Backspace	Undo.
Alt+Delete	Redo.
Ctrl+]	Find matching parenthesis.
F1	Context sensitive help.
F5	Go To Line.
F8	Run.
F9	Check syntax.
Shift+F2	Save the source and close the SQL Editor.
Ctrl+A	Find next.
Ctrl+L	Replace All.
Ctrl+N	Next warning.
Ctrl+R	Replace next.
Ctrl+S	Incremental search.
Ctrl+Y	Delete current line.
Ctrl+Z	Find first.
Ctrl+Shift+I	Indent block.
Ctrl+Shift+U	Outdent block.
Ctrl+Shift+Z	Replace first.
Ctrl+K+B	Set start of block (Persistent blocks must be on).
Ctrl+K+C	Copy block to Clipboard.
Ctrl+K+E	Change word under cursor to lowercase.

Ctrl+K+F	Change word under cursor to uppercase.
Ctrl+K+H	Hide block.
Ctrl+K+I	Indent block.
Ctrl+K+K	Set end of block (Persistent blocks must be on).
Ctrl+K+L	Mark current line.
Ctrl+K+N	Uppercase block.
Ctrl+K+O	Lowercase block.
Ctrl+K+R	Read block from file.
Ctrl+K+S	Save form.
Ctrl+K+T	Mark word under cursor.
Ctrl+K+U	Outdent block.
Ctrl+K+V	Move block (Persistent blocks must be on).
Ctrl+K+W	Write block to file.
Ctrl+K+Y	Delete current block.
Ctrl+O+C	Set column block.
Ctrl+O+I	Set inclusive block.
Ctrl+O+K	Set non inclusive block.
Ctrl+O+L	Set line block.
Ctrl+O+O	Toggle case of block.

BRIEF keymap

See also

If you selected the BRIEF keystroke mapping in the Editor Preferences dialog box (Display page) the keystrokes in the left column will perform the actions shown on the right.

Left Arrow	Moves (or extends selection) one column left of cursor.
Right Arrow	Moves (or extends selection) one column right of cursor.
Up Arrow	Moves (or extends selection) one line up from cursor.
Down Arrow	Moves (or extends selection) one line down from cursor.
Home	Moves (or extends selection) to start on line, then start of screen, then start of editor.
End	Moves (or extends selection) to end of line, then end of screen, then end of editor.
Page Up	Moves (or extends selection) one page up.
Page Down	Moves (or extends selection) one page down.
Backspace	Delete character to the left of the cursor.
Delete	Delete character to the right of the cursor. (On the cursor if box shaped cursor).

Tab	Inserts a tab, or indents block if a block exists.
Ctrl+Left Arrow	Moves (or extends selection) one word left of cursor.
Ctrl+Right Arrow	Moves (or extends selection) one word right of cursor.
Ctrl+Home	Moves (or extends selection) to top of screen.
Ctrl+End	Moves (or extends selection) to bottom of screen.
Ctrl+Page Up	Moves (or extends selection) to start of editor.
Ctrl+Page Down	Moves (or extends selection) to end of editor.
Ctrl+Backspace	Delete the word to the left of the cursor.
Alt+Backspace	Delete the word to the right of the cursor.
Shift+Home	Moves (or extends selection) to left of screen.
Shift+End	Moves (or extends selection) to right of screen.
Shift+Tab	Moves back to previous tab, or outdents block if a block exists.
Insert	Paste from Clipboard.
Minus (Num Keypad)	Cut block to Clipboard, cut current line if no block is selected.
Plus (Num Keypad)	Copy block to Clipboard; copy current line if no block is selected.
Star (Num keypad)	Undo.
F1	<Not assigned. Reserved for later use>
F5	Find first.
F6	Replace first.
F9	Run.
Alt+F2	Zoom window.
Alt+F5	Reverse search first.
Alt+F6	Reverse replace first.
Alt+F10	Syntax check.
Ctrl+F5	Toggle case sensitive in search.
Ctrl+F6	Toggle advanced match in search.
Ctrl+F9	Run.
Shift+F5	Search next.
Shift+F6	Replace next.

Ctrl+B	Scroll current line to bottom of window.
Ctrl+C	Scroll current line to center of window.
Ctrl+D	Scroll window one line down.
Ctrl+E	Scroll window one line up.
Ctrl+K	Delete to beginning of line.
Ctrl+M	Enter key.
Ctrl+N	Next warning.
Ctrl+S	Incremental search.
Ctrl+T	Scroll current line to top of window.
Ctrl+U	Redo.
Alt+A	Start non-inclusive block marking.
Alt+C	Start column marking.
Alt+D	Delete line.
Alt+H	Help.
Alt+I	Toggle insert / overwrite mode.
Alt+K	Delete to end of line.
Alt+L	Start line marking.
Alt+M	Start inclusive block marking.
Alt+R	Read block from file.
Alt+S	Search first.
Alt+T	Replace first.
Alt+U	Undo.
Alt+X	Close editor without saving code.
Ctrl+Q+[	Find matching parenthesis.
Ctrl+Q+]	Find matching parenthesis.
Ctrl+O+O	Toggle case of block.

Epsilon keymap

See also

If you selected the Epsilon keystroke mapping in the Editor Preferences dialog box (Display page) the keystrokes in the left column will perform the actions shown on the right.

Left Arrow	Move cursor left one column.
Right Arrow	Move cursor right one column.
Up Arrow	Move cursor one line up.
Down Arrow	Move cursor one line down.
Home	Move cursor to top of screen.
End	Move cursor to bottom of screen.
Page Up	Move cursor up one page.
Page Down	Move cursor down one page.

Backspace	Delete character to the left of the cursor.
Delete	Delete character to the right of the cursor. (On the cursor if box shaped cursor).
Insert	Toggle insert / overstrike mode.
Tab	Inserts a tab, or indents to indent of previous line if smart-tab is turned on.
Ctrl+Left Arrow	Move cursor left one word.
Ctrl+Right Arrow	Move cursor right one word.
Ctrl+Home	Move cursor to start of editor.
Ctrl+End	Move cursor to end of editor.
Esc+Left Arrow	Move cursor to start of line.
Esc+Right Arrow	Move cursor to end of line.
F1	Context sensitive help.
F8	Run.
F9	Undo.
Shift+F2	Save the source and close the editor.
Ctrl+F9	Undo.
Ctrl+F10	Redo.
Ctrl+A	Move cursor to start on line.
Ctrl+B	Move cursor left one column.
Ctrl+D	Delete character to the right of the cursor. (On the cursor if box shaped cursor).
Ctrl+E	Move cursor to end of line.
Ctrl+F	Move cursor right one column.
Ctrl+H	Delete character to the left of the cursor.
Ctrl+K	Delete current line and copy/append to Clipboard.
Ctrl+L	Scroll current line to center of window.
Ctrl+M	Enter.
Ctrl+N	Move cursor one line down.
Ctrl+P	Move cursor one line up.
Ctrl+S	Incremental search.
Ctrl+T	Switch the 2 characters on each side of the cursor.
Ctrl+V	Move cursor down one page.
Ctrl+W	Delete block and copy/append to Clipboard.
Ctrl+Y	Paste from Clipboard.
Ctrl+Z	Scroll one line up.

Ctrl+_	Context sensitive help.

Ctrl+X, I	Read block from file.
Ctrl+X, U	Undo.
Ctrl+X, W	Write block to file.
Ctrl+X, Tab	Indent block.

Ctrl+X, Ctrl+I	Indent block.
Ctrl+X, Ctrl+R	Redo.
Ctrl+X, Ctrl+T	Switch current and previous line.
Ctrl+X, Ctrl+U	Undo.
Ctrl+X, Ctrl+W	Write block to file.
Ctrl+X, Ctrl+X	Exchange cursor and block marker.

Alt+B	Esc, B	Move cursor left one column.
Alt+C	Esc, C	Uppercase word.
Alt+D	Esc, D	Delete word and copy/append to Clipboard.
Alt+F	Esc, F	Move cursor right one word.
Alt+L	Esc, L	Lowercase word.
Alt+M	Esc, M	Move cursor to first character on current line.
Alt+T	Esc, T	Switch the words before and after the cursor.
Alt+U	Esc, U	Uppercase word.
Alt+V	Esc, V	Move cursor to previous page.
Alt+W	Esc, W	Copy block to Clipboard.
Alt+Z	Esc, Z	Scroll window one line down.
Alt+ ,	Esc, ,	Move cursor to top of window.
Alt+ .	Esc, .	Move cursor to end of window.
Alt+ \	Esc, \	Delete white space on both sides of cursor.
Alt+)	Esc,)	Find the matching parenthesis.
Alt+ <	Esc, <	Move cursor to start of editor.
Alt+ >	Esc, >	Move cursor to end of editor.

Alt+ ?	Esc, ?	Context sensitive help.
Alt+ %	Esc, %	Replace first.
Alt+ *	Esc, *	Replace first (advanced match).
Alt+ &	Esc, &	Replace first (match string).
Alt+ @	Esc, @	Set marker to current cursor position.
Alt+Backspace	Esc, Backspace	Delete word left of cursor and copy/append to Clipboard.
Alt+Tab	Esc, Tab	Indents to indent of previous line.
Alt+Ctrl+B	Esc, Ctrl+B	Find matching parenthesis.
Alt+Ctrl+F	Esc, Ctrl+F	Find matching parenthesis.
Alt+Ctrl+H	Esc, Ctrl+H	Delete the word left of cursor and copy/append to Clipboard
Alt+Ctrl+R	Esc, Ctrl+R	Search first (backward, advanced match).
Alt+Ctrl+S	Esc, Ctrl+S	Search first (forward, advanced match).
Alt+Ctrl+W	Esc, Ctrl+W	Append to the keyboard (until next non Clipboard action).
Alt+Ctrl+ \	Esc, Ctrl+ \	Block indent.

If the directory you want has an alias, you can select it in the Alias drop-down list. The name of the selected directory appears in the Look In drop-down list and the files in that directory appear in the file list.

View the Form opens the form in its view window.

Edit the Form Design opens the form in its design window.

Print the Report prints the form as a report if you also choose Open as Report.

Open as a Report opens the form as a report. This is a quick way to use a form layout to specify the layout of a report.

Run the Query runs the query and displays the Answer table.

Edit the Query opens the query in its design window.

View the Report opens the report in its view window.

Edit the Report Design opens the report in its design window.

Print the Form prints the report as a form if you also choose Open as Form.

Open as a Form opens the report as a form. This is a quick way to use a report layout to specify the layout of a form.

Run the Script runs the script.

Edit the Script opens the script in its design window.

Run the Query runs the query and displays the Answer table.

Edit the Query opens the query in its design window.

Opens a form or report using a different master table▪a different table from the one on which it was originally designed. When you choose Change Table, Database Desktop opens the Select File dialog box, where you specify the new master table.

Empty Desktop Toolbar

See also

The following buttons appear on the Standard Toolbar when the Database Desktop window is empty. Click a button below to see specific information on it.

Open or Create Table			Open or Create Query
Open or Create SQL File			

Like the menus, the Toolbar changes when the active window changes. Each window has a unique Toolbar.

Open or Create

To open an object of this type, click the button to display the Open dialog for this file type. Choose the file you want to open. To create a new instance of the object, right-click the button and choose New from the menu that appears.

Query window Toolbar

See also

The Standard Toolbar for a Query window is available whenever a Query window is active. The Toolbar provides some shortcuts to common menu commands, as well as a method of joining tables with example elements. Click a button below to see specific information on it.

Cut to Clipboard	Copy to Clipboard
Paste from Clipboard	Run Query
Join Tables	Query Properties
Sort Answer Table	Show SQL
Add Table	Remove Table
Field View	

SQL Editor Toolbar

See also

The Standard Toolbar for a SQL Editor window provides some shortcuts to common menu commands. Click a button below to see specific information on it.

Cut to Clipboard	Copy to Clipboard
Paste from Clipboard	Run SQL
Find	Find Next
Select Alias	Query Properties

Table window Toolbar

See also

Use the Toolbar buttons as quick shortcuts for navigating through the table or performing menu commands. As you point to each button, tool tips appear with the name of the button.

The following figure shows the buttons available on the Table window's Standard Toolbar, which displays when a table is selected in the Database Desktop window. The buttons that control movement among records are collectively called navigation buttons. Click a button below to see specific information on it.

Cut to Clipboard	Copy to Clipboard
Paste from Clipboard	Restructure
First Record	Previous Record Set
Previous Record	Next Record
Next Record Set	Last Record
Field View	Edit Data

Global Toolbar

See also

Use the Global Toolbar's buttons as quick shortcuts for performing commonly used menu commands.

To display the Global Toolbar, choose View|Toolbars, and check Global. The Global Toolbar is always displayed regardless of the active window.

The following figure shows the buttons available on the Global Toolbar. Click a button below to see specific information on it.

Open or Create Table	Open or Create Query
Open or Create SQL File	

Help button

Clicking the Help ? button is the same as pressing F1.

Join Tables button

To join two tables using the Join Tables ▪ button,

1. Click the Join Tables button. When you move the pointer over a query image, the pointer changes to a join symbol.
2. Click the corresponding field of each table. Database Desktop places example elements that join the tables.

After you click the two corresponding fields, Database Desktop returns the mouse pointer to normal behavior. To return the mouse pointer to normal behavior without joining the tables, click the Join Tables button again.

About keyboard commands

Most Database Desktop mouse operations have keyboard equivalents. These keyboard commands usually have an abbreviated series of keystrokes called shortcuts.

General Keys

Menu command keys

Application access keys

Control menu keys

Help system keys

Dialog box keys

Shortcuts in Tables

Navigation and selection keys

Table operation shortcuts

Edit mode keys

Function Key Actions

Function keys in tables

Function keys in queries

Function keys in the SQL Editor window

Special Procedures

Rotating columns in a table

Selecting multiple fields

Super Tab operations

▪

Menu command keys

See also

Use the following keys to access the menus in Database Desktop.

Use	To
Alt	Select the menu bar
Alt then Hyphen	Open the child window Control Menu.
Alt then Spacebar	Open the Database Desktop Control Menu
Arrow keys	Select menu items
Enter or ▪	Open the selected menu
Esc	Exit from a menu without choosing a command

Each menu name has an underlined letter. To open a menu, press Alt plus the underlined letter. For example,

Alt+F	Opens the File menu

To choose a command from a menu, select it with the arrow keys and press Enter. Or press the underlined letter (also referred to as the access key) in the command name. For example, when the File menu is open, press the letter O to open a file, S to save a file, or X to exit Database Desktop and unload it from memory.

Application access keys

See also

Use	To
Esc	Close a Control menu but leave the Control-menu box selected. Press Esc a second time to deselect the Control-menu box.
Alt+Esc	Display the next application on the Windows desktop. This will restore a minimized application. The name of the application will not be shown before displaying it.
Ctrl+Esc	Access the Start button on the taskbar.
Alt+Hyphen	Open the active window's Control Menu.
Alt+Spacebar	Open the Database Desktop Control menu which allow you to Restore, Move, Size, Minimize, Maximize or Close Database Desktop. The Restore option is available only if the Database Desktop window has been minimized.To move the Database Desktop window, choose Move, use the arrow keys to reposition the window, and press Enter when you finish.
Alt+Tab	Display the next application on the Windows desktop. This will restore a minimized application. The icons of all loaded applications are displayed in a box. Hold down Alt and press Tab repeatedly until the icon you want is selected, then release Alt.

▪

Control menu keys

See also

The Control menu opens when you click the icon at the left end of a window title bar.

Use	To
Alt+Hyphen	Open the active window's Control menu.
Alt+Spacebar	Open the Database Desktop Control menu which allow you to Restore, Move, Size, Minimize, Maximize or Close Database Desktop. The Restore option is available only if the Database Desktop window has been minimized.To move the Database Desktop window, choose Move, use the arrow keys to reposition the window, and press Enter when you finish.
Alt+F4	Close the window. Windows prompts you for confirmation if your current work is unsaved.
Esc	Close a Control menu but leave the Control-menu box selected. Press Esc a second time to deselect the Control-menu box.
Ctrl+Esc	Access the Start button on the taskbar.

Help system keys

See also

To access Help, press the F1 key. Depending on the current context,

When	Press F1 to display help for
In a menu	The selected command.
In a dialog	That dialog.
In a child window	That window.
Empty desktop	Database Desktop User's Guide Help topics.

When a help topic is displayed, you may use the keys listed below to navigate within a Help topic, between topics, and within the Help system,

Use	To
Esc	Close the current Help window.
Tab	Move to the next underlined Help topic. For a jump, press Enter. For a glossary definition, press Enter once to see the definition; press Enter again to make it go away.
Shift+Tab	Move to the previous underlined Help topic. For a jump, press Enter. For a glossary definition, press Enter once to see the definition; press Enter again to make it go away.
▪	Scroll down one line (if a vertical scrollbar is displayed).
↑	Scroll up one line (if a vertical scrollbar is displayed).
PgUp, PgDn	Scroll the Help window (if vertical scrollbars are displayed).
Alt+C	Display the Help Contents for the current Help topic.
Alt+B	Return to the last Help screen you viewed.
Alt+I	Search for a particular term in the Help index.

Dialog box keys

See also

Use	To
Esc	Cancel a dialog box, leaving the settings unchanged. Same as the Cancel button.*
Spacebar	Select. (Same as left-click.) Depending on the context, pressing the Spacebar can: ▪ Toggle a selected check box. ▪ Set a selected option button. ▪ Select an item in a list box or drop down list box. ▪ Activate a push button.
Ctrl+Spacebar	Select non-contiguous items in a multi-select list box. (Same as Ctrl+Left-click.)
Shift+Spacebar	Select a contiguous group of items in a multi-select list box. (Same as Shift+Left-click.)
Enter	Activate a selected command button.
Tab	Move to the next named option or group of options. Or press Alt plus an underlined letter in the option name to choose that option directly.
Shift+Tab	Move to the previous named option or group of options.
▪ and ↑	Depending on the context, the up and down arrows can: ▪ Select a radio button. ▪ Highlight a selection list item. Or press the first letter of the item name to select the first item in the list beginning with that letter. ▪ Move to another check box inside a group of check boxes.
Alt▪	Drop a list down for the selected list box.
F1	Display context-sensitive Help on the dialog box.
Alt+F4	Closes the dialog box.
F3	Super Back Tab. In multi-region dialog boxes, Super Back Tab moves backward from panels with tables in them to other panels. (Tab and Shift+Tab move among objects or fields within the region.)
F4	Super Tab. In multi-region dialog boxes, Super Tab moves forward from panels with tables in them to other panels. (Tab and Shift+Tab move among objects or fields within the region.)

* If there is no Cancel button, ESC will not close the dialog. Use the Close or OK buttons as appropriate instead.

▪

Rotating columns in a table

See also

To rotate the order of a table's columns with the keyboard,

1. Select the column to move.
2. Press Ctrl+R.

   This moves the selected column to the last place on the right of the table.

You can rotate the order of columns in a Table window.

Selecting multiple fields

See also

You can select multiple fields across rows and columns in a Table window by dragging a box around the ones you want. Fields selected this way must be adjoining.

To select a group of fields using the keyboard,

1. Select the field where you want to begin (do not enter Field View).
2. Hold down Shift while using the arrow keys to place a box around the fields you want.

Selecting all fields

To select all fields in the table (the entire table), choose Edit|Select All. Database Desktop places a box around the whole table.

Note: You can select multiple fields in a Table window only.

▪

Function keys in tables

See also

Key	Action
F1	Displays Help for the Table window
F2	Field View
Ctrl+F2	Persistent Field View
Ctrl+F3	Refresh data
F5	Lock record
Shift+F5	Post record
Ctrl+F5	Post/Keep Lock
F6	View current field's right-click menu
F9	Edit/End edit
F10	View Menu
Shift+F10	View current field's right-click menu
F11	Previous record
Shift+F11	Previous set
Ctrl+F11	First record
F12	Next record
Shift+F12	Next set
Ctrl+F12	Last record

*When viewing SQL data, you must press Ctrl+F3 to perform a data refresh. Changes made by others do not automatically refresh the screen.

For alpha fields containing DDE links, Shift+F2 launches the server application.

For additional keys, see

- Table operation shortcuts
- Navigation and selection keys
- Keyboard actions in Table windows

▪

Function keys in queries

See also

Key	Action
F1	Displays Help
F2	Field View
Ctrl+F2	Persistent Field View
F3	Moves to the previous table image
F4	Moves to the next table image
F5	Tells Database Desktop that you are about to define an example element
F6	Check/uncheck toggle ▪. The check type (Check or CheckPlus) is specified on the Query page of the Preferences dialog. If pressed when in the column on the far left, F6 will check/uncheck all fields in the table.
Shift+F6	Cycle checks ▪, ▪, ▪, ▪ and ▪
F8	Run query
F10	Menu

▪

Function keys in the SQL Editor window

See also

The SQL Editor supports the following keystroke mappings,

- The default Database Desktop keymap
- BRIEF keymap
- Epsilon keymap

▪

Super Tab operations

See also

Use Super Tab (F4) and Super Back Tab (F3) to move among panels in multi-region windows and dialog boxes.

Create/Restructure dialog box

The Field Roster is one panel in a multi-region dialog box. You use Tab or Shift+Tab to move from column to column in the Field Roster. The Table Properties panel▪its list boxes and buttons ▪is another panel in this multi-region dialog box. Tab and Shift+Tab move through all the objects in the Table Properties panel, then through all other buttons in the dialog box.

- Press Super Back Tab (F3) to move from the Field Roster panel to the Help button.
- Press Super Tab (F4) to move from the Field Roster panel to the Table Properties panel.
- To return to the Field Roster panel from the Table Properties panel, press Shift+Tab until you get back (you see no insertion point or highlighted box in the Table Properties panel). Or Tab forward through all objects in the Table Properties panel and the dialog box until you reach the Field Roster panel.

Query images

In multi-table queries, use Super Tab (F4) to move forward among the query table images. (Tab and Shift+Tab move right and left among fields within a query image.) Use Super Back Tab (F3) to move backward.

Other multi-region dialog boxes

In dialog boxes that contain embedded table images, Super Tab and Super Back Tab move you out of the table. (Tab and Shift+Tab move right and left among fields within a table.)

Key	Action
F3	Super Back Tab
F4	Super Tab

▪

Table operation shortcuts

See also

Key combination	Action
Alt+Backspace	Undo
Ctrl+F	Field View (same as F2)
Ctrl+Ins	Copy to the Clipboard
Ctrl+L	Lock the current record
Ctrl+Shift+L	Post record
Ctrl+R	Rotate columns
Del	Clear or delete (as appropriate)
Shift+Del	Cut to the Clipboard
Shift+Ins	Paste from the Clipboard
Spacebar	Enter current date, time, or both in date, time, or timestamp fields. You must press the Spacebar for each part of the field's format.

For additional keys, see

- Keyboard actions in Table windows
- Function keys in tables
- Navigation and selection keys

▪

Navigation and selection keys

See also

This table shows the keys you can use to navigate with when you are looking at data in tables. Make sure Num Lock is off when you use Alt in combination with a keypad key. Field view and non-field-view keys are listed.

Key	Non-field view	Field view
PgUp	Up one set of records	Up one set of records
Ctrl+PgUp	Left one screen	Left one screen
PgDn	Down one set of records	Down one set of records
Ctrl+PgDn	Right one screen	Right one screen
Home	First field of record	Beginning of field
Shift+Home	Select to first field of record	Select to beginning of field
Ctrl+Home	First field of first record	First field of first record
Alt+Home	First field of record	First field of record
End	Last field of record	End of field
Shift+End	Select to last field of record	Select to end of field
Ctrl+End	Last field of last record	Last field of last record
Alt+End	Last field of record	Last field of record
←	Left one field	Left one character
Shift ←	Select left one field	Select left one character
Ctrl ←	First column	Left one word
Ctrl+Shift▪	Select to first field of record	Extend selection left one word
Alt▪	Left one field	Left one field
▪	Right one field	Right one character
Shift▪	Select right one field	Select right one character
Ctrl▪	Last column	Right one word
Ctrl+Shift▪	Select to last field of record	Extend selection right one word
Alt▪	Right one field	Right one field
▪	Up one field	Up one line in multi-line field or up one record in single-line field
Shift▪	Select up one field	Select up one line within multi-line field or up one record in single-line field
Alt▪	Up one field	Up one field
▪	Down one field	Down one line within multi-line field or down

		one record in single-line field
Shift▪	Select down one field	Select down one line within multi-line field or down one record in single-line field
Alt▪	Down one field	Down one field

Edit mode keys

See also

This table shows the keys to use while editing. Entering Field View does not change the action of these keys.

Key	Action
Ins	Insert record
Shift+Ins	Paste (same as Edit|Paste)
Ctrl+Ins	Copy (same as Edit|Copy)
Del	Delete selected text (same as Edit|Delete)
Shift+Del	Cut (same as Edit|Cut)
Ctrl+Del	Delete record
Backspace	Delete character to the left or delete selected text
Ctrl+Backspace	Delete word to left
Alt+Backspace	Undo record edit (same as Edit|Undo)
Esc	Undo field edit
Tab	Post value and move to next field
Shift+Tab	Post value and move to previous field
Enter	Post value and move to next field
Ctrl+Spacebar	Lookup Help (if defined.)**
Ctrl+Shift+Spacebar	Move Help (if applicable).***

**For more information on Table Lookup and Lookup Help, see About Table Lookups.

***For more information on Move Help, see Record|Move Help.

▪

Glossary

{button A,JI(`',`gloss_a')} {button B,JI(`',`gloss_b')} {button C,JI(`',`gloss_c')} {button D,JI(`',`gloss_d')} {button E,JI(`',`gloss_e')} {button F,JI(`',`gloss_f')} {button G,JI(`',`gloss_g')} {button H,JI(`',`gloss_h')} {button I,JI(`',`gloss_i')} {button J,JI(`',`gloss_i')} {button K,JI(`',`gloss_k')} {button L,JI(`',`gloss_l')} {button M,JI(`',`gloss_m')} {button N,JI(`',`gloss_n')} {button O,JI(`',`gloss_o')} {button P,JI(`',`gloss_p')} {button Q,JI(`',`gloss_q')} {button R,JI(`',`gloss_r')} {button S,JI(`',`gloss_s')} {button T,JI(`',`gloss_t')} {button U,JI(`',`gloss_v')} {button V,JI(`',`gloss_v')} {button W,JI(`',`gloss_w')} {button X,JI(`',`gloss_z')} {button Y,JI(`',`gloss_z')} {button Z,JI(`',`gloss_z')}

A

active

alias

alpha field

ANSI

Answer table

arithmetic operators

ascending order

ASCII

asymmetrical outer join

B

BDE

binary field

blank field

BLOB

C

cascade

check box

checkmark

client

Clipboard

command

comparison operator

composite key

concatenate

constant

D

data

data integrity

data type

database

Database Desktop window

DDE

default

default action

default value

dependent tables

descending order

detail table

dialog box
drop-down list box

E
example element
exclusive link

F
field
field type
field value
Field View
file
file name
font
function keys

G
group (n.)
GroupBy operator

H
highlight

I-J
icon
inclusion operator
inclusive link
index

K
key

L
link
list box
lock
logical operator
logical value
lookup table

M
Main menu
master table
multi-value relationship

N
normalized data structure
number field

O
object

OEM
OLE
OLE container
OLE server
operator
outer join

P
picture
pop-up definitions
primary index
private directory
prompt
properties

Q
query
query by example (QBE)
query operators
query statement

R
record
record number
referential integrity
reserved words
restructure

S
secondary index
selection condition
server
set
set comparison operator
short field
single-value relationship
special field
SQL
string
structure
summary operator
syntax error

T
table
table header
table language driver
Toolbar

active
The object or window to which the next keystroke or mouse action will apply.

alias

A name you assign to a database in addition to its original name. When you use an alias, you don't need to specify the directory path.

alpha field
A field containing letters, numbers, or a combination of both.

ANSI
American National Standards Institute. The character set supported by Windows.

Answer table
A temporary table used to store the results of a query.

arithmetic operators
The **+**, **-**, *****, **/**, and ( ) operators used to construct arithmetic expressions in queries and calculated fields.

ascending order
A sort order: alphabetic order in alpha fields (most often A to Z case-sensitive, but the order depends on the language driver you are using); low to high in numeric fields, earliest to latest in date fields.

ASCII
American Standard Code for Information Interchange. A sequence of 128 standard characters.

asymmetrical outer join
A query in which an inclusive link is specified for only one of the tables involved.

BDE
Borland Database Engine (formerly IDAPI). Database Desktop uses this database engine to access and deliver data. BDE maintains information about your PC's environment in the BDE configuration file (usually called IDAPI32.CFG). Use the BDE Configuration Utility to change the settings in this configuration file.

binary field
A field used to store data Database Desktop cannot interpret. A common use of a binary field is to store sound.

blank field
A field that does not contain a value.

BLOB
Binary large object. Field types that can contain BLOBs include binary, memo, formatted memo, graphic, and OLE. Certain rules apply to these fields as a whole, and they are sometimes discussed collectively as BLOB fields.

cascade
To use referential integrity to update child tables when a value changes in the parent table.

check box
A box you can check or uncheck to set an option. You can check more than one check box in a set.

checkmark
The symbol ▪ used in query statements to indicate that a field is to be displayed in the Answer table.

client
The application that starts a DDE or OLE conversation and usually receives data from the other application, called the server.

Clipboard
A temporary area used to copy and paste information from one location to another.

command
A word on a menu or button that you choose to perform an action.

comparison operator
In a query, the operators (<, >, <=, >=, and =) you can use to compare two values.

composite key

A key comprised of two or more fields of a Paradox table which, together, provide a unique value for the table.

concatenate
To combine two or more alphanumeric values using the **+** operator.

constant
A specific, unchanging value used in calculations.

data
The information in a table.

data integrity
The assurance that only valid data can be entered in a field and that links between common fields in separate tables cannot be broken; supported by validity checks and referential integrity.

data type

The kind of data a field can contain.

- Paradox data types are alpha, number, money, short, long integer, BCD, date, time, timestamp, memo, formatted memo, graphic, OLE, logical, autoincrement, binary, and bytes.
- dBASE data types are character, float, number, date, logical, memo, OLE, and binary.

database

An organized collection of information; in Database Desktop, a collection of related tables and queries in a given directory.

DDE
Dynamic Data Exchange. A way for two or more applications to share data.

default
What Database Desktop automatically does or looks like in the absence of an overriding command.

default action

The choice that Database Desktop determines to be the most logical or safest and the one that will be carried out unless otherwise specified. Default actions are performed by double-clicking an object or its icon.

default value
In validity checks, the value automatically entered in a field if no other value is entered.

descending order
A sort order: reverse alphabetical order in alpha fields (most often Z to A, case-sensitive, but the order depends on the language driver you are using); high to low in numeric fields, latest to earliest in date fields.

dependent tables
Tables that depend on the current table for referential integrity.

Database Desktop window
The main window in Database Desktop.

detail table
In multi-table relationships, the table whose records are subordinate to those of the master table.

dialog box

A box that requests user input or provides information. Many dialog boxes present options to choose among before you can perform an action; others display warnings or error messages.

All dialog boxes require interaction with the user; a modal dialog box, however, keeps the focus until you respond to it.

drop-down list box
A single-line text box that opens to display more choices when you click the arrow beside it.

example element
A character or group of characters that represents a value in a field of a query.

exclusive link

In a query, the use of an example element to retrieve from one table only those records that match the records in another table.

field
A column of information in a table. A collection of related fields makes up one record.

field type

The type of data a field can contain.

- Paradox data types are alpha, number, money, short, long integer, BCD, date, time, timestamp, memo, formatted memo, graphic, OLE, logical, autoincrement, binary, and bytes.
- dBASE data types are character, float, number, date, logical, memo, OLE, and binary.

field value
The data contained in one field of a record. If no data is present, the field is considered blank.

Field View

A mode that lets you move through a field character by character. Use this mode to view field values that are too large to be displayed in the current field width, or to edit a field value. Pressing F2 puts you into Field View.

file
A collection of information stored under one name on a disk. For example, Paradox tables are stored in files.

file name

When entering a file name, you can simply type the name of the file; or, you can specify the file name combined with a drive, path, alias, or a combination of these elements. Some examples of valid file names:

File name	Description
MYFILE.DB	File name
D:\MYFILE.DB	File name combined with drive
:MYWORK:MYFILE.DB	File name combined with alias
:MYWORK:\PHONEDIR\MYFILE.DB	File name combined with alias and path

font
A design applied to all characters. Fonts are typically available in different sizes, measured in points; 1 point equals 1/72 of an inch. Font styles usually include bold, italic, and underline.

function keys
The 12 keys across the top of the keyboard labeled F1 through F12. (Some keyboards have 10 function keys at the left.) These keys provide fast access to Database Desktop operations.

group

In a query, a set of records that either

- Have the same value in one or more fields
- Fall within a range of values
- Are displayed in a fixed number of records

GroupBy operator

In a query, the operator (indicated by ▪) that groups records by a field without displaying the field's values in the Answer table.

highlight
To select by dragging the mouse across a line or lines of text.

icon
A graphical representation of an object.

inclusion operator

The symbol **!** used with an example element to include a complete set of records in the Answer table, whether or not they match records in another table.

inclusive link
A query whose answer includes all the values in a field of one table, whether or not there are matching values in the linked field of another table.

index
A file that determines an order in which Database Desktop can access the records in a table. A Paradox table's key establishes its primary index.

key

A field or group of fields in a Paradox table used to order records or ensure referential integrity. Establishing a key has three effects:

- The table is prevented from containing duplicate records.
- The records are maintained in sorted order based on the key fields.
- A primary index is created for the table.

link
To establish a relationship between tables by linking corresponding fields.

list box
A list of selectable items in a dialog box.

lock
A device placed on a table in a multi-user environment that prevents other users from viewing, changing, or locking the table.

logical operator
One of three operators (AND, OR, or NOT) that can be used in queries.

logical value
A value (True or False) assigned to an expression when it is evaluated.

lookup table
A table that assures that a value entered in one table matches an existing value in another table.

Main menu
The menu bar across the top of the Database Desktop window.

master table
In a multi-table relationship, the primary table.

modal
A modal dialog box keeps the focus until you respond to it. You cannot move nor resize a modal dialog box.

multi-value relationship

A multi-value relationship exists between tables if, for every record in one table, no records, one record, or more than one record from another table is related to it.

normalized data structure

An arrangement of data in tables in which each record includes the fewest number of fields necessary to establish unique categories. Rather than using a few redundant fields to provide all possible information within a single table, normalized tables distribute information over many tables using fewer fields. Normalized tables provide more flexibility in terms of analysis.

number field
A field that can contain only numbers, a sign, and a decimal point.

object
A table, query, or SQL file. All entities that can be manipulated in Database Desktop are objects.

OEM
Stands for Original Equipment Manufacturer and refers to the character set your computer uses.

OLE
Object linking and embedding. A means of linking to or embedding data from other applications (servers) in another application (the client).

OLE container

An application that requests the services of and cooperates with an OLE server to offer the user a way to view and edit objects created by that server.

OLE server
An application that provides services to and cooperates with an OLE client to offer the user a way to view and edit objects created by that server.

operator
A symbol that represents an operation to be performed on a value or values. For example, the **+** operator represents addition, and the ***** operator represents multiplication.

outer join
A type of query that uses the inclusion operator (**!**) to retrieve all records in a table, whether or not they match records in another table.

picture
A pattern of characters that defines what you can type into a field during editing or data entry.

pop-up definitions
Introduce you to terms that might be unfamiliar or words that Database Desktop uses in a special way.

primary index

An index on the key fields of a Paradox table. A primary index

- Determines the location of records
- Lets you use the table as the detail in a link
- Keeps records in sorted order
- Speeds up operations

private directory
A nonshared directory for storing temporary objects, such as Answer tables. The default private directory is PRIVATE, created below the main Database Desktop directory on your hard drive, or on your network home directory if you have no hard drive. You can change to another private directory if you want.

prompt
Instructions displayed on the screen. Prompts ask for information or guide you through an operation.

properties
The attributes of an object. You can right-click an object to view or change its properties.

query
A way to retrieve data from your tables. See About queries.

query by example (QBE)
The method of retrieving data by providing an example of what you are looking for.

query operators
The reserved words Database Desktop uses in queries. See Query operators for more information.

query statement
One or more filled out query images in the Query window.

record
A horizontal row in a data table that contains a group of related fields of data.

record number
A unique number that identifies each record in a data table.

referential integrity
A way of ensuring that the ties between like data in separate tables cannot be broken.

reserved words
The names of commands, functions, system variables, and operators.

restructure
To change the structure of an existing table. You can change the field names, field types, field order, key, indexes, validity checks, referential integrity, password protection, table language, and table lookup.

secondary index
An index used for linking, querying, and changing the view order of tables.

selection condition
Expressions typed in the fields of a query image to specify the conditions that records must meet to appear in the Answer table or live query.

server
The application that responds to the calling application, or client, in a DDE or OLE conversation. The server usually sends data to the client.

set
In a query, the specific group of records you intend to query.

set comparison operator
One of the reserved words (ONLY, NO, EVERY, EXACTLY) used to compare a defined set of records to other records.

short field
A Paradox field type that can contain numbers from -32,768 through 32,767 with no decimal values.

single-value relationship

A single-value relationship exists between tables if, for every record in one table, no records or only one record from another table is related to it.

special field
A field that contains information about a table. Special fields include Today and Now.

SQL
Structured Query Language (abbreviated SQL and commonly pronounced "sequel"). The standard language for storing and manipulating data in relational databases.

string
An alphanumeric value, or an expression consisting of alphanumeric characters.

structure
The arrangement of fields in a table.

summary operator
One of the operators (AVERAGE, COUNT, MAX, MIN, or SUM) that retrieves data from groups of records in queries.

syntax error
An error caused by an incorrectly expressed statement.

table
A structure made up of rows (records) and columns (fields) that contains data.

table header

Paradox tables consist of two sections: the header and the data blocks. The header contains information about the number of fields, passwords, write protection, sort order, and the version of Paradox or Database Desktop that created the table. Indexes and memos are stored in separate files.

table language driver
Determines the table's sort order and available character set. The BDE Configuration Utility lets you set the default language driver for Paradox and dBASE tables.

Toolbar
The set of buttons and tools for frequently performed tasks. The Toolbar is under the menu bar and changes according to the window you are using.

To get quick help on what a tool or button does, point to it▪Database Desktop displays a description of the button in the status line.

validity check
A constraint on the values you can enter in a field.

wildcard operators
Special characters Database Desktop uses to match patterns in queries or when locating values.

working directory
The default data directory Database Desktop uses to open and save files.

Alias Manager dialog box (SQL Link driver)

See also

Use the Alias Manager dialog box to create or modify aliases for remote database directories, or to connect to or disconnect from the target SQL server.

You are working with an alias for a SQL Link driver. The settings in the Alias Manager dialog box reflect the information stored in your BDE configuration file. Refer to the documentation for more information on using the driver selected in the Driver Type list box.

Note: If you are working with an ODBC driver connection, some of the options below are not available.

Dialog box options

Public Alias
Check this check box to make an alias a public alias available from all applications that use BDE. Uncheck this check box to make an alias a project alias

available only to Paradox applications in the current directory.

Database Alias
Choose an alias from the list. To create a new SQL Link driver alias, choose New, then choose the appropriate driver type, then type the new alias name.

Alias configuration options

Driver Type
To create an SQL Link driver alias, choose the appropriate driver type from the list.

Server Name
The name of the target SQL server.

User Name
The default name for accessing the SQL server.

Open Mode
The mode in which SQL Link opens the SQL database. Can be READ/WRITE (default) or READ ONLY.

Schema Cache Size
The number of SQL tables whose schema information will be cached. Can be any whole number from 0 to 32 (default=8).

Langdriver
The language driver used to display SQL data to Database Desktop (U.S. default=blank). Choose the language driver that uses the same character set in which the server passes data to Database Desktop, and a collation sequence that matches your server's collation sequence. [more]

SQLqrymode
SQL query mode; the method for handling queries to SQL data. Can be NULL (blank setting, which is the default), SERVER, or LOCAL. [more]

SQLpassthru Mode
SQL pass-through mode; specifies whether or not Database Desktop users can access the SQL server via both QBE and the SQL Editor (pass-through SQL) in the same alias connection. Can be NOT SHARED, SHARED AUTOCOMMIT (default), or SHARED NO AUTOCOMMIT. [more]

Password
Type the password needed to connect to the server. Asterisks (*) represent the characters you type.

Show options

Show Public Aliases Only
Click this button if you want to see only public aliases.

Show Project Aliases Only
Click this button if you want to see only project aliases.

Show All Aliases
Click this button if you want to see both public and project aliases.

Connect
Choose Connect to log on to the server named in the Server Name text box, using the current User Name and Password. Depending on your server, this may take a while.

Disconnect
Choose Disconnect to log off the server named in the Server Name text box.

New
Choose New to open an empty box where you can type in a new alias name. After you click New, the button becomes the Keep New button.

Keep New
Choose Keep New if you want this to be a temporary alias, existing only until you exit. Then click OK to close the Alias Manager dialog box.

Note: Choosing Keep New does not close the dialog box. It lets you save the alias temporarily when you click OK. If you click Cancel, whatever you saved with Keep New is deleted.

Choose Keep New if you are creating several aliases and do not want to re-open this dialog box to create each one. Then choosing Save As will save them all.

Remove
Choose Remove to tag the selected alias for removal. The alias is removed when you exit the box or when you choose Save As and overwrite the current file containing the alias.

Save As
Choose Save As if you want your new alias to be permanent-usable any time you use Database Desktop. You will see the Save File As dialog box. Unless you choose a different configuration file, Database Desktop stores saved aliases in your default BDE configuration file.

Note: The message, "File already exists. Overwrite?", appears when you click Save As and choose a file name. Choose Yes. Database Desktop appends the aliases to the file; it does not replace the ones already there.

OK
Choose OK if you want to save any changes you have made in the dialog box, but only for the current Database Desktop session.

Cancel
Cancels only the changes in type-in boxes. Any changes you made with Save As remain.

▪

Alias Manager dialog box (InterBase SQL Link)

See also

Use the Alias Manager dialog box to create or modify aliases for local, network, or remote database directories. You can also choose to connect or disconnect from a server. Most of the options described below are available only if you have installed a Borland SQL Link driver and have chosen the INTRBASE option from the Driver Type list.

The settings in this dialog box reflect the information stored in your BDE configuration file.

Dialog box options

Public Alias
Check this check box to make an alias a public alias▪available from all applications that use BDE. Uncheck this check box to make an alias a project alias
▪available only to Paradox applications in the current directory.

Database Alias
Choose an alias from the list. To create a new InterBase alias, choose New, then choose the INTRBASE driver type, then type the new alias name.

Alias configuration options

Driver Type
To create an InterBase alias, choose INTRBASE.

Server Name
The name of the target InterBase SQL server, usually a path that includes the server name.

User Name
The default name for accessing the InterBase SQL server.

Open Mode
The mode in which SQL Link opens the InterBase database. Can be READ/WRITE (default) or READ ONLY.

Schema Cache Size
The number of SQL tables whose schema information will be cached. Can be any whole number from 0 to 32 (default=8).

Langdriver
The language driver used to display SQL data to Database Desktop (U.S. default=blank). Choose the language driver that uses the same character set in which the server passes data to Database Desktop, and a collation sequence that matches your server's collation sequence. [more]

SQLqrymode
SQL query mode; the method for handling queries to SQL data. Can be NULL (blank setting, which is the default), SERVER, or LOCAL. [more]

SQLpassthru Mode
SQL pass-through mode; specifies whether or not Database Desktop users can access the InterBase SQL server via both QBE and the SQL Editor (pass-through SQL) in the same alias connection. Can be NOT SHARED, SHARED AUTOCOMMIT (default), or SHARED NO AUTOCOMMIT. [more]

Schema Cache Time
The time (in seconds) that a table list is cached. A value of -1 causes the schema list to be cached until the database is closed and reopened.

Password

Type the password needed to connect to the server. Asterisks (*) represent the characters you type.

Show options

Show Public Aliases Only
Click this button if you want to see only public aliases.

Show Project Aliases Only
Click this button if you want to see only project aliases.

Show All Aliases
Click this button if you want to see both public and project aliases.

Connect
Choose Connect to log on to the server named in the Server Name text box, using the current User Name and Password. Depending on your server, this may take a while.

Disconnect
Choose Disconnect to log off the server named in the Server Name text box.

New
Choose New to open an empty box where you can type in a new alias name. After you click New, the button becomes the Keep New button.

Keep New
Choose Keep New if you want this to be a temporary alias, existing only until you exit. Then click OK to close the Alias Manager dialog box.

Note: Choosing Keep New does not close the dialog box. It lets you save the alias temporarily when you click OK. If you click Cancel, whatever you saved with Keep New is deleted.

Choose Keep New if you are creating several aliases and do not want to re-open this dialog box to create each one. Then choosing Save As will save them all.

Remove
Choose Remove to tag the selected alias for removal. The alias is removed when you exit the box or when you choose Save As and overwrite the current file containing the alias.

Save As
Choose Save As if you want your new alias to be permanent-usable any time you use Database Desktop. You will see the Save File As dialog box. Unless you choose a different configuration file, Database Desktop stores saved aliases in your default BDE configuration file.

Note: The message, "File already exists. Overwrite?", appears when you click Save As and choose a file name. Choose Yes. Database Desktop appends the aliases to the file; it does not replace the ones already there.

OK
Choose OK if you want to save any changes you have made in the dialog box, but only for the current Database Desktop session.

Cancel
Cancels only the changes in type-in boxes. Any changes you made with Save As remain.

Alias Manager dialog box (Informix SQL Link)

See also

Use the Alias Manager dialog box to create or modify aliases for local, network, or remote database directories. You can also choose to connect or disconnect from a server. Most of the options described below are available only if you have installed a Borland SQL Link driver and have chosen the INFORMIX option from the Driver Type list.

The settings in this dialog box reflect the information stored in your BDE configuration file.

Dialog box options

Public Alias
Check this check box to make an alias a public alias available from all applications that use BDE. Uncheck this check box to make an alias a project alias
available only to Paradox applications in the current directory.

Database Alias
Choose an alias from the list. To create a new Informix alias, choose New, then choose the INFORMIX driver type, then type the new alias name.

Alias configuration options

Driver Type
To create an Informix database alias, choose INFORMIX from the list.

Server Name
The name of the target Informix SQL server.

Database
The name of the target Informix database.

User Name
The default name for accessing the Informix SQL server.

Open Mode
The mode in which SQL Link opens the Informix database. Can be READ/WRITE (default) or READ ONLY.

Langdriver
The language driver used to display SQL data to Database Desktop (U.S. default=blank). Choose the language driver that uses the same character set in which the server passes data to Database Desktop, and a collation sequence that matches your server's collation sequence. [more]

Schema Cache Size
The number of SQL tables whose schema information will be cached. Can be any whole number from 0 to 32 (default=8).

SQLqry Mode
SQL query mode; the method for handling queries to SQL data. Can be NULL (blank setting, which is the default), SERVER, or LOCAL. [more]

Date Mode
The format in which the Informix driver sends dates to the server. Must match the Informix server's DBDATE environment variable. Can be either 0 (MDY; default) or 1 (DMY).

Password
Type the password needed to connect to the server. Asterisks (*) represent the characters you type.

Show options

Show Public Aliases Only

Click this button if you want to see only public aliases.

Show Project Aliases Only

Click this button if you want to see only project aliases.

Show All Aliases

Click this button if you want to see both public and project aliases.

Connect

Choose Connect to log on to the server named in the Server Name text box, using the current User Name and Password. Depending on your server, this may take a while.

Disconnect

Choose Disconnect to log off the server named in the Server Name text box.

New

Choose New to open an empty box where you can type in a new alias name. After you click New, the button becomes the Keep New button.

Keep New

Choose Keep New if you want this to be a temporary alias, existing only until you exit. Then click OK to close the Alias Manager dialog box.

Note: Choosing Keep New does not close the dialog box. It lets you save the alias temporarily when you click OK. If you click Cancel, whatever you saved with Keep New is deleted.

Choose Keep New if you are creating several aliases and do not want to re-open this dialog box to create each one. Then choosing Save As will save them all.

Remove

Choose Remove to tag the selected alias for removal. The alias is removed when you exit the box or when you choose Save As and overwrite the current file containing the alias.

Save As

Choose Save As if you want your new alias to be permanent-usable any time you use Database Desktop. You will see the Save File As dialog box. Unless you choose a different configuration file, Database Desktop stores saved aliases in your default BDE configuration file.

Note: The message, "File already exists. Overwrite?", appears when you click Save As and choose a file name. Choose Yes. Database Desktop appends the aliases to the file; it does not replace the ones already there.

OK

Choose OK if you want to save any changes you have made in the dialog box, but only for the current Database Desktop session.

Cancel

Cancels only the changes in type-in boxes. Any changes you made with Save As remain.

Alias Manager dialog box (Oracle SQL Link)

See also

Use the Alias Manager dialog box to create or modify aliases for local, network, or remote database directories. You can also choose to connect or disconnect from a server. Most of the options described below are available only if you have installed a Borland SQL Link driver and have chosen the ORACLE option from the Driver Type list.

The settings in this dialog box reflect the information stored in your BDE configuration file.

Dialog box options

Public Alias
Check this check box to make an alias a public alias▪available from all applications that use BDE.
Uncheck this check box to make an alias a project alias
▪available only to Paradox applications in the current directory.

Database Alias
Choose an alias from the list. To create a new Oracle alias, choose New, then choose the ORACLE driver type, then type the new alias name.

Alias configuration options

Driver Type
To create an Oracle alias, choose ORACLE.

Server Name
The name of the target Oracle server.

User Name
The default name for accessing the Oracle server.

Net Protocol
Network transport used to communicate with the database server. [more]

Open Mode
The mode in which SQL Link opens the Oracle database. Can be READ/WRITE (default) or READ ONLY.

Schema Cache Size
The number of SQL tables whose schema information will be cached. Can be any whole number from 0 to 32 (default=8).

Langdriver
The language driver used to display SQL data to Database Desktop (U.S. default=blank). Choose the language driver that uses the same character set in which the server passes data to Database Desktop, and a collation sequence that matches your server's collation sequence. [more]

SQLqrymode
SQL query mode; the method for handling queries to SQL data. Can be NULL (blank setting, which is the default), SERVER, or LOCAL. [more]

SQLpassthru Mode
SQL pass-through mode; specifies whether or not Database Desktop users can access the InterBase SQL server via both QBE and the SQL Editor (pass-through SQL) in the same alias connection. Can be NOT SHARED, SHARED AUTOCOMMIT (default), or SHARED NO AUTOCOMMIT. [more]

Password
Type the password needed to connect to the server. Asterisks (*) represent the characters you type.

Show options

Show Public Aliases Only

Click this button if you want to see only public aliases.

Show Project Aliases Only

Click this button if you want to see only project aliases.

Show All Aliases

Click this button if you want to see both public and project aliases.

Connect

Choose Connect to log on to the server named in the Server Name text box, using the current User Name and Password. Depending on your server, this may take a while.

Disconnect

Choose Disconnect to log off the server named in the Server Name text box.

New

Choose New to open an empty box where you can type in a new alias name. After you click New, the button becomes the Keep New button.

Keep New

Choose Keep New if you want this to be a temporary alias, existing only until you exit. Then click OK to close the Alias Manager dialog box.

Note: Choosing Keep New does not close the dialog box. It lets you save the alias temporarily when you click OK. If you click Cancel, whatever you saved with Keep New is deleted.

Choose Keep New if you are creating several aliases and do not want to re-open this dialog box to create each one. Then choosing Save As will save them all.

Remove

Choose Remove to tag the selected alias for removal. The alias is removed when you exit the box or when you choose Save As and overwrite the current file containing the alias.

Save As

Choose Save As if you want your new alias to be permanent-usable any time you use Database Desktop. You will see the Save File As dialog box. Unless you choose a different configuration file, Database Desktop stores saved aliases in your default BDE configuration file.

Note: The message, "File already exists. Overwrite?", appears when you click Save As and choose a file name. Choose Yes. Database Desktop appends the aliases to the file; it does not replace the ones already there.

OK

Choose OK if you want to save any changes you have made in the dialog box, but only for the current Database Desktop session.

Cancel

Cancels only the changes in type-in boxes. Any changes you made with Save As remain.

▪

Alias Manager dialog box (Sybase SQL Link)

See also

Use the Alias Manager dialog box to create or modify aliases for local, network, or remote database directories. You can also choose to connect or disconnect from a server. Most of the options described below are available only if you have installed a Borland SQL Link driver and have chosen the SYBASE option from the Driver Type list.

The settings in this dialog box reflect the information stored in your BDE configuration file.

Dialog box options

Public Alias
Check this check box to make an alias a public alias▪available from all applications that use BDE.
Uncheck this check box to make an alias a project alias

▪available only to Paradox applications in the current directory.

Database Alias
Choose an alias from the list. To create a new Sybase or Microsoft SQL Server alias, choose New, then choose the SYBASE driver type, then type the new alias name.

Alias configuration options

Driver Type
To create a Sybase or Microsoft SQL Server database alias, choose SYBASE from the list.

Database
The name of the target database.

Server Name
The name of the target SQL Server.

User Name
The default name for accessing the SQL Server.

Open Mode
The mode in which SQL Link opens the Informix database. Can be READ/WRITE (default) or READ ONLY.

Schema Cache Size
The number of SQL tables whose schema information will be cached. Can be any whole number from 0 to 32 (default=8).

BLOB Edit Logging
Enables or disables the logging of BLOB edits. Can be TRUE (default) or FALSE. When FALSE, this option helps minimize BLOB space requirements and increase performance.

Langdriver
The language driver used to display SQL data to Database Desktop (U.S. default=blank). Choose the language driver that uses the same character set in which the server passes data to Database Desktop, and a collation sequence that matches your server's collation sequence. [more]

SQLqry Mode
SQL query mode; the method for handling queries to SQL data. Can be NULL (blank setting, which is the default), SERVER, or LOCAL. [more]

Password
Type the password needed to connect to the server. Asterisks (*) represent the characters you type.

Show options

Show Public Aliases Only

Click this button if you want to see only public aliases.

Show Project Aliases Only

Click this button if you want to see only project aliases.

Show All Aliases

Click this button if you want to see both public and project aliases.

Connect

Choose Connect to log on to the server named in the Server Name text box, using the current User Name and Password. Depending on your server, this may take a while.

Disconnect

Choose Disconnect to log off the server named in the Server Name text box.

New

Choose New to open an empty box where you can type in a new alias name. After you click New, the button becomes the Keep New button.

Keep New

Choose Keep New if you want this to be a temporary alias, existing only until you exit. Then click OK to close the Alias Manager dialog box.

Note: Choosing Keep New does not close the dialog box. It lets you save the alias temporarily when you click OK. If you click Cancel, whatever you saved with Keep New is deleted.

Choose Keep New if you are creating several aliases and do not want to re-open this dialog box to create each one. Then choosing Save As will save them all.

Remove

Choose Remove to tag the selected alias for removal. The alias is removed when you exit the box or when you choose Save As and overwrite the current file containing the alias.

Save As

Choose Save As if you want your new alias to be permanent-usable any time you use Database Desktop. You will see the Save File As dialog box. Unless you choose a different configuration file, Database Desktop stores saved aliases in your default BDE configuration file.

Note: The message, "File already exists. Overwrite?", appears when you click Save As and choose a file name. Choose Yes. Database Desktop appends the aliases to the file; it does not replace the ones already there.

OK

Choose OK if you want to save any changes you have made in the dialog box, but only for the current Database Desktop session.

Cancel

Cancels only the changes in type-in boxes. Any changes you made with Save As remain.

LANGDRIVER settings

Long driver name	Short name	Character set	Collation seq.
Paradox 'ascii'	ascii	DOS code page 437	Binary
Paradox 'intl'	intl	DOS code page 437	Paradox 'intl'
Paradox 'intl' 850	intl850	DOS code page 850	Paradox 'intl' 850
Paradox 'nordan'	nordan	DOS code page 865	Paradox 'nordan'
Paradox 'nordan40'	nordan40	DOS code page 865	Paradox 'nordan40'
Paradox 'swedfin'	swedfin	DOS code page 437	Paradox 'swedfin'
Paradox ANSI INTL	ANSIINTL	ISO8859.1 (ANSI)	Paradox 'intl'
Paradox ESP 437	SPANISH	DOS code page 437	Paradox ESP 437
Paradox ISL 861	iceland	DOS code page 861	Paradox ISL 861
Pdox ANSI INTL850	ANSII850	ISO8859.1 (ANSI)	Pdox 'intl' 850
Pdox ANSI NORDAN40	ANSINOR4	ISO 8859.1 (ANSI)	Pdox 'nordan40'
Pdox ANSI SWEDFIN	ANSISWFIN	ISO 8859.1 (ANSI)	Pdox 'swedfin'
Pdox ESP ANSI	ANSISPAN	ISO 8859.1 (ANSI)	PDox ESP437
SQL Link ROMAN8	BLROM800	ROMAN8	Binary
Borland ENU Latin-1	BLLT1US0	ISO 8859.1 (ANSI)	Binary

SQLQRYMODE settings

Setting	Meaning
NULL (blank setting)	Server-local mode (default). Query goes first to the server. If the server is unable to perform the query, the query is performed at the Desktop.
SERVER	Server-only mode. Query is sent to the server. If the server is unable to perform the query, the query fails.
LOCAL	Local-only mode. Query is always performed at the Desktop.

SQLPASSTHRU MODE settings

NOT SHARED

(blank setting) (Default for InterBase, Oracle, Sybase)
Pass-through SQL and non-pass-through SQL do NOT share the same connection.

SHARED AUTOCOMMIT

(Default for Informix)
Pass-through SQL and non-pass-through SQL will share the same connection, and (as long as you are not in an explicit client transaction or batch mode) pass-through SQL will be automatically committed.

SHARED NOAUTOCOMMIT

Pass-through SQL and non-pass-through SQL share the same connection, but pass-through statements will not be automatically committed.

NET PROTOCOL settings

Value	Description
3270	IBM 3270 protocol
APPC	IBM APPC LU 6.2 protocol
ASYNC	Asynchronous (dial-up) access protocol
DECNET	Digital Equipment Corporation DECnet protocol
NAMED PIPES	Named Pipes protocol, as used by OS/2
NETBIOS	NetBios protocol, as used by LAN Manager and other PC LANs
SPX/IPX	SPX/IPX protocol, as used by Novell NetWare
TCP/IP	Transport Control Protocol/ Internet Protocol, as used by Unix and VAX workstations
VINES	Banyan VINES protocol

▪

Database Information dialog box

See also

Use the Database Information dialog box to view or modify the connection parameters you set for accessing remote servers. You need to modify these parameters when

- You connect to a server for the first time in a session
- You change connections to access data in a different location

Database Desktop displays the parameter settings you entered in the Alias Manager dialog box. In most cases, all you need to add or modify is the user name and password.

Dialog box options

Database Alias
Database Desktop displays the alias name you entered in the Alias Manager dialog box, or specified when you tried an operation against a remote database.

Server Name
Database Desktop displays the full path of the database specified in the alias. If necessary, type a new path for the database, including the name of the server.

User Name
Type the name of the user recognized by the database server.

Default BDE configuration file

The BDE configuration file used at Database Desktop startup. The default configuration file is listed in the Windows registry.

The BDE configuration file that comes with Database Desktop is called IDAPI32.CFG. However, you can give your BDE configuration file any name as long as it ends in ".CFG" and is no more than 12 characters long.

ODBC driver connection

A connection from your BDE application to an ODBC driver. The connection requires your BDE application, a vendor-supplied ODBC driver, the Microsoft ODBC Driver Manager, a BDE alias on the workstation side, and an ODBC data source on the server side.

Once you create an ODBC driver connection, it appears on the list of available drivers in the BDE Configuration Utility. This enables you to set up an alias for the target ODBC data source and connect to it through your BDE application.

About SQL

See also

SQL (Structured Query Language) descended from SEQUEL (or Structured English QUEry Language) is a language for constructing relational database management systems (RDBMS) on any hardware platform. It is now the standard language for network queries across different hardware and software platforms.

SQL servers run on local area network (LAN) file-server systems, minicomputers, and mainframes. They handle requests in logical units of work called transactions. Transaction processing protects your data against conflicts that may arise when more than one person is working on a table at the same time.

In SQL, all transactions can be explicitly ended with a command to either accept or discard the changes. Once you are satisfied that no errors occurred during the transaction, you can end that transaction with a COMMIT command. The database then changes to reflect the operations you have just performed. If an error occurs, you can abandon the changes with the ROLLBACK command.

Transaction
A transaction is a group of related operations that must all be performed successfully before the database management system will finalize any changes to the database.

SQL terminology

See also

SQL	Paradox	Description
Table	Table	A structure of rows (records) and columns (fields) that contains information.
Row	Record	A group of columns (fields) in a table that contain related information about a single record.
Column	Field	A category of information (column) in a table that cuts across all rows in the table.

▪

Preparing to connect to an SQL database

See also

Before you can begin to access an SQL database you must complete the following steps:

Action to complete	Description
Enable SQL database access	Make sure you have a valid user ID and password on the SQL server, and at least Read access privileges for the SQL database. See your database administrator.
Install necessary client software	Install any client software libraries required to communicate with the SQL server. Test software is usually included. Make sure this test software can successfully connect to the SQL server before using an SQL alias.
Install an SQL driver	Install an SQL driver for your SQL server; Database Desktop supports Borland SQL Links as well as ODBC.
Configure the SQL driver	When you first install the SQL driver it uses all the default driver settings. Make sure these default settings are right for your server installation before you create any aliases for your SQL database. For details, see Help for the BDE Configuration Utility, installed in the Program Group with Database Desktop.
Create at least one SQL alias	Your SQL database alias includes your user name and password on the target SQL server, and is required to access any SQL data. A generic SQL alias is automatically created the first time you modify the default link driver parameters after installation. See the Borland SQL Links Help or Help for the BDE Configuration Utility.

▪

Connecting to the SQL server

See also

The first time you try to query or view a table in your SQL database through Database Desktop, the Database Information dialog box appears. To complete the connection, enter your password in the Database Information dialog box and click OK.

If the connection is successful, Database Desktop continues with the operation you requested. The connected database remains connected until you either exit Database Desktop or manually disconnect.

Connecting manually

If you want to connect to a database without first performing a database action, you can connect manually through the Alias Manager:

1. Choose Tools|Alias Manager to open the Alias Manager dialog box.
2. Choose the desired alias from the Database Alias drop-down list. Database Desktop displays the alias and its connection parameters.
3. If necessary, modify the alias connection parameters. Then enter your password and choose Connect. If the connection is successful, the Alias Manager displays "Connection is successful. Database is open."
4. To close the Alias Manager dialog box, click OK.

Disconnecting manually

To disconnect from the SQL server without exiting Database Desktop, redisplay the Alias Manager, select the appropriate alias, and choose Disconnect.

Changes in the Database Desktop window

See also

Since Database Desktop supports the use of SQL operations against local (Paradox or dBASE) tables, the SQL Editor is visible in the Toolbar whether you have an SQL driver installed or not. When an SQL driver is installed, Database Desktop is said to be SQL-enabled.

The Desktop changes in a number of ways:

New icons
Whenever you access the SQL server you see the SQL hourglass.

Working directories
Since you cannot store objects such as documents, queries, reports, and forms on SQL servers, you cannot set an SQL server as the location for your working directory.

You might want to set up a local directory to hold all the forms, reports, .TVS files, and queries you use when you work with a particular SQL database. Once you connect to that database you can then open the local directory and put your working tools at your fingertips. You can also easily apply the tools across other SQL databases.

Refreshing data displays
When you are using local tables, as soon as one user makes a change to a shared database all users see their view of the data refreshed. However, when you are working on an SQL server, this does not occur.

If you are working with indexed SQL tables, you can update the active window by pressing Ctrl+F3 periodically. Ctrl+F3 shows any updates made to a table while you are viewing it.

▪

Using Table windows

See also

When Database Desktop is SQL-enabled, Table windows change in the following general ways:

- Table windows of SQL data do not display record numbers; the scroll box is always in the center of the vertical scroll bar.
- If an SQL table has any indexes, it will always be viewed in order by some index (you can select which one). This allows fast and reliable updates.
- When you query an SQL table using QBE, Database Desktop stores the SQL table properties in a file with the extension .TVS. This helps distinguish them from Table window property files for Paradox tables (which end in .TV) and dBASE tables (which end in .TVF).

Note: .TVS files for SQL tables are not automatically deleted when you delete the SQL table. Also, if you change your private directory the table will no longer be displayed with the properties you set.

- If you try to view an SQL table when someone else is editing data, you may have to wait until the other user is finished editing.

Changes in QBE

See also

The characteristic behavior of SQL update queries means that updates to SQL data are performed either completely or not at all. When you use QBE to perform updates on SQL data, Database Desktop does not generate any of the following auxiliary tables:

CHANGED.DB	INSERTED.DB
DELETED.DB	ERRORCHG.DB
ERRORINS.DB	ERRORDEL.DB

For detailed information on how to use QBE to query and update SQL data, see About querying SQL data.

About querying SQL data

See also

Database Desktop provides several different ways to perform operations against SQL data:

Users who are unfamiliar with SQL can frame queries to SQL tables in QBE. They can also view and edit data directly through Table windows and Form windows.

Users who are familiar with SQL can pass SQL statements directly to the database through the SQL Editor window.

Note: You can also use the SQL Editor window to perform SQL operations against local (Paradox or dBASE) data. For further information, see About Local SQL.

▪

Using QBE to query SQL data

See also

Query by example (QBE) provides you with a graphical format that helps you show what kind of information you want in your Answer table. When you use QBE to query a table in an SQL database, Database Desktop attempts to translate your query to an equivalent SQL statement and pass it to the SQL server. If successful, the server processes your query, then passes the answer set back to you through the SQL driver. The SQL Editor lets you view the equivalent SQL statement for the query at any time during query construction, or after it is processed.

Note: If the SQL database does not support an equivalent SQL statement for a QBE query, a message confirms that the query is processing in the QBE environment.

Querying an SQL table works exactly the same way as querying a local table in Database Desktop:

1. Choose File|New|Query.
2. Select the Alias to the SQL server you want to query.
3. Select the SQL table you want to query.
4. Fill out the Query image, specifying data selection criteria.
5. Press F8 (Run) or click the Run Query button to process your query.

Note: You cannot interrupt a query while it is processing as long as the SQL hourglass is visible. The size of the SQL table determines query retrieval time.

Borland SQL Links also supports the use of queries that join SQL tables with local tables, or SQL tables from different SQL databases (heterogeneous queries). Heterogeneous queries are always processed according to QBE rules.

Once you become familiar with the syntax of SQL queries, you may prefer to use the SQL Editor to write SQL statements and send them directly to your server. This type of query is always processed by the rules of your SQL server. For more information, see Using pass-through SQL.

Note: QBE queries sent to an SQL server are automatically under transaction control. However, if you run the SQL equivalent of a QBE query, those SQL statements are not under automatic transaction control. Non-QBE transactions must be explicitly begun and either committed or rolled back.

To view the SQL translation for a query you constructed using QBE,

1. Connect to the SQL database as described in Connecting to the SQL server.
2. Use QBE to construct a query to the SQL database.
3. Open the SQL Editor window in either of the following ways:

- Choose Query|Show SQL.
- Click the SQL button.

Depending on the type of query you just created, the SQL translation will be one of the following types of statements:

Desired result of query	Equivalent SQL statement
Display specific data	SELECT
Add new data	INSERT
Change existing data	UPDATE
Remove existing data	DELETE

Using pass-through SQL

See also

Programmers familiar with SQL can use the SQL Editor window to directly enter, execute, or save an SQL statement on a remote SQL server. In Database Desktop, this is called "using pass-through SQL." The remote SQL server performs all error or syntax checking. You can save the SQL statement to a disk file (the SQL Editor automatically saves the file with the extension .SQL), and then later load, modify, or execute it.

The following topics describe two alternatives to using pass-through SQL:

Using QBE to query SQL data

About Local SQL

For information about the SQL Editor, see About the SQL Editor.

▪
Creating an SQL table

See also

When you create an SQL table,

- You specify the driver type in the Table Type dialog box after choosing File|New|Table.
- You can define the table structure (fields & types), specify required fields, and define indexes.

Other features of Paradox tables, such as validity checks and referential integrity, are not supported on SQL tables.

- On the Create Table dialog box, the Dec field is the number of decimal places for numeric fields.
- You name indexes as described in Creating indexes on SQL tables.

When you use an SQL table in Database Desktop, the table should have a unique index. If it does not have a unique index and you insert a record, you may not be able to view the record until you close the table and reopen it. To add a unique index, choose Utilities|Restructure.

You can create an SQL table using pass-through SQL in the SQL Editor, as described in Using pass-through SQL.

▪

Creating indexes on SQL tables

See also

You can use Database Desktop to create and modify indexes on SQL tables.

To create an index for an SQL table,

1. Display the Create Table or the Restructure Table dialog box.
2. Choose Define Index.

   Database Desktop displays the Define Index dialog box.

When you use an SQL table in Database Desktop, the table should have a unique index. If it does not have a unique index and you insert a record, you may not be able to view the record until you close the table and reopen it.

Naming SQL Indexes

For most database servers, index names must be unique for all tables in a database (or in some other predefined workspace). Index names must start with a letter, not a number. When you create an index on an SQL table, Database Desktop prefixes the index name with the table name to ensure that the index name is unique.

Sybase note: Sybase index names do not need to be unique within a database, so Database Desktop does not prefix Sybase index names with table names.

When you create an SQL index and choose OK from the Define Index dialog box, Database Desktop supplies the prefix "<table>_" for the index name. For example, if you are creating the index "last_name" on the Customer table, Database Desktop gives the index the name "customer_last_name".

You can include the table name with the index name or omit it:

- If you type the index name following "<table>_", Database Desktop prefixes the index name with the table name and an underscore.
- If you delete "<table>_", Database Desktop omits the table name from the index name. If the index name is not unique, an error will occur when Database Desktop saves the table.

This index naming scheme also affects copying and restructuring.

▪

Restructuring an SQL table

See also

When you restructure an SQL table using Database Desktop, you can add, modify, and drop indexes. You cannot otherwise change the structure of a table on a server with Database Desktop, unless you use pass-through SQL.

When you use an SQL table in Database Desktop, the table should have a unique index. If it does not have a unique index and you insert a record, you may not be able to view the record until you close the table and reopen it.

Prefixing the Index Name with the Table Name

Database Desktop prefixes some index names with the table name, as described in Creating indexes on SQL tables. These index names are also affected when you restructure an SQL table as follows:

- If you create a new index during a restructure, Database Desktop prefixes the index name with the table name unless you delete the "<table>_" string from the index name.
- If you modify an index during a restructure, Database Desktop does not modify the index name, unless you rename the index as part of your modification.
- If you choose Save As during a restructure, Database Desktop renames all index names with the new table name, even if the index names are not prefixed with the current table name. (Otherwise, a duplicate index name would be guaranteed.) For example, suppose the EMPLOYEE table contains the following indexes:
- EMPLOYEE_DEPT_NO
- EMPLOYEE_EMP_NO
- FULL_NAME
- JOB

If you restructure the table and save it as MY_DEPT, Database Desktop renames the indexes as follows:

- MY_DEPT_DEPT_NO
- MY_DEPT_EMP_NO
- MY_DEPT_FULL_NAME
- MY_DEPT_JOB

Note: If, during a restructure operation, you add an index and omit the "<table>_" string or modify an index name in any way, Database Desktop does not prefix the index name with the table name during the Save As operation.

For example, suppose you restructure an InterBase table EMPLOYEE which contains an index EMPID. While saving the index, you change the index name to DEPT105_EMPID. When you choose Save As, Database Desktop saves the table and does not prefix the DEPT105_EMPID index name with the new table name.

▪

About the SQL Editor

See also

The SQL Editor is a full featured text editor that includes color highlighting, smart tab indent, and many other features. It also supports BRIEF- and Epsilon-style editing.

Use the SQL Editor window to directly enter, execute, or save an SQL statement. This is sometimes called pass-through SQL. You specify the SQL statement in your server's dialect. The SQL server performs all error or syntax checking and executes the statement without any involvement by Database Desktop.

The SQL Editor appears when you open or create a new SQL file.

As you work with the SQL Editor, you can use the keyboard or the SQL Editor Toolbar.

By default, keywords appear in bold, and comments in italics. You can change colors and text attributes in the Editor Preferences dialog box on the Colors page.

Note: The SQL Editor does not automatically wrap lines of text. A line extends to the right as you type until you press Enter to begin a new line.

Customizing the Editor

You can customize the SQL Editor by choosing Edit|Editor Preferences and choosing your preferences on the various pages of the Editor Preferences dialog box. Many options are available, such as color highlighting, incremental search, smart tab indent, and so on. You can also choose BRIEF or Epsilon keymaps, instead of the Database Desktop default. See below.

Shift+F1 help

For a listing of keystrokes that correspond to the keymap you choose in the Editor Preferences dialog box, place the insertion point on a blank space in the SQL Editor and press Shift+F1.

Keystroke mappings

You can choose from three keystroke mappings in the SQL Editor:

- The default Database Desktop keymap
- BRIEF keymap
- Epsilon keymap

Of the three, the default is the only CUA keymap. The BRIEF and Epsilon mappings do not allow standard menu access through hotkeys, and standard MDI keys are not available.

Menus

Using the BRIEF and Epsilon keymaps, you can access the menus by pressing F10 or by pressing and releasing the Alt key. This moves the focus to the menu. Then press the shortcut key for the wanted menu.

The Default keymap allows menu access as for BRIEF and Epsilon, but in addition the menus can be reached by the standard Alt+Key combination, for example, Alt+E for the Edit menu.

Standard MDI system keys

Standard MDI system keys are only available for the Default keymap. Examples of these keys are:

Ctrl+F6	The MDI window toggle
Alt+F6	The SDI window toggle
Ctrl+F4	Closing an MDI window

For more information on keys, see To move around the SQL Editor with the keyboard.

To open the SQL Editor

See also

To open the SQL Editor, do one of the following:

- To enter (and execute) a new SQL statement,
  Choose File|New|SQL Statement or right-click the Open SQL Script button and choose New.
- To open (and edit or execute) an existing .SQL file,
  Choose File|Open|SQL Statement, click the Open SQL Script button, or right-click the Open SQL Script button and choose Open.
- To view the SQL equivalent of an open QBE query,
  Choose Query|Show SQL or click the Open SQL Script button.

To move around the SQL Editor with the keyboard

See also

Use the following keys to move around in the SQL Editor:

Ctrl+left arrow	Moves the cursor one word to the left.
Ctrl+right arrow	Moves the cursor one word to the right.
Home	Moves the cursor to the beginning of a line.
End	Moves the cursor to the end of a line.
Ctrl+Home	Moves the cursor to the beginning of the text.
Ctrl+End	Moves the cursor to the end of the text.
Page up	Moves one screenful back.
Page down	Moves one screenful forward.
Backspace	Deletes the character to the left of the cursor.
Delete	Deletes the character to the right of the cursor.
Insert	Has no effect because the SQL Editor is always in insert mode. As you type, characters are pushed to the right. You cannot overwrite characters.
Ctrl+C	Copies selected text to the clipboard.
Ctrl+X	Copies selected text to the clipboard and deletes it from the window.
Ctrl+V	Pastes text from the clipboard into your method.
Tab	Inserts a Tab character and pushes text to the right.

To select text in the SQL Editor

See also

You can select a block of text by dragging with the mouse, using the arrow keys with Shift held down, or clicking with Shift held down to extend the selection.

- To select a word, double-click it.
- To select an entire line, click to the left of the line and drag the insertion point. (The mouse is in position when the I-beam changes to an arrow.)

To select a block of text, either

- Click and drag the mouse
- Press Shift and use the arrow keys
- Click to indicate the starting position, then press Shift to extend the selection

The keymapping you choose in the Editor Preferences dialog box also affects selection.

When text is selected, what happens when you type a character (or paste from the Clipboard) depends on whether you checked Overwrite Blocks in the Editor Preferences dialog box.

To search for text in the SQL Editor

See also

To find and/or replace text in an SQL Editor window,

- Choose Search|Find or Search|Replace.

You can use these two commands to search for text from the insertion point forward (or backward if you check the Backwards option). The Find And Replace dialog box (displayed with Search|Replace) lets you replace the specified text with a specified value.

To exit the SQL Editor

See also

- Double-click the SQL Editor's Control menu (or choose Close from the SQL Editor window's Control menu.

To run your statement from the SQL Editor window, which closes the SQL Editor, click the Run SQL button.

To save an SQL statement before you exit, use File|Save or Save As. For more information, see To save an SQL statement.

▪

About SQL statements

See also

SQL statements are the instructions you use to communicate with databases on SQL servers. If you are only accessing one remote server, you can use the particular SQL syntax required by that server. If you are accessing several servers and/or Paradox and dBASE tables on your local system, you can use Local SQL. (For more information, see About Local SQL.)

You can create SQL statements by typing them directly into the SQL Editor or you can run a query using QBE and display the SQL equivalent of that query in the SQL Editor. For instructions, see To view the SQL translation of a QBE query.

Note: Before you can access a database on an SQL server, you must give the database an alias. For instructions, see To create a new alias.

To specify an alias in the SQL Editor

See also

Before running an SQL statement, you must specify the alias that the statement will run against. To specify an alias, do one of the following:

- Choose SQL|Select Alias.
- Click the Select Alias button.

Database Desktop opens the Select Alias dialog box, where you can choose one of the aliases you created in the Alias Manager dialog box or the BDE Configuration Utility.

If you do not specify an alias, Database Desktop uses the alias `:WORK:`.

You can include aliases in the text of the SQL statement only if you are using Local SQL.

If you need to join local and remote tables (in a heterogeneous join), specify a local alias, then include the remote alias in the text of the SQL statement by using Local SQL.

To enter an SQL statement

See also

To enter an SQL statement, type the statement in the SQL Editor. You can enter multiple SQL statements if your server allows it and you include only one SELECT statement.

You can include aliases in the text of the SQL statement only if you are using Local SQL.

Use the following commands on the Edit menu to select, locate, and replace text:

Command	Description
Find	Search for strings of text in your code.
Find Next	Search for the next occurrence of the text you specified using Find.
Replace	Search for text and replace it with a value you specify.
Replace Next	Replace the next occurrence of the text specified using Replace.
Select All	Select all text in the SQL Editor window.

To run an SQL statement

See also

To run an SQL statement that you have typed in the SQL Editor window, do one of the following:

- Click the Run SQL button.
- Choose SQL|Run SQL.

The SQL server performs all error or syntax checking and executes the statement.

If your SQL statement is a query, the query results are displayed in an Answer table.

Note: Before running an SQL statement, specify the alias the statement will run against by choosing SQL|Select Alias.

To save an SQL statement

See also

- Choose File|Save or File|Save As.

When you save an SQL statement to your local hard disk, Database Desktop places it in an unformatted text file with an .SQL extension.

If the Prompt To Save option in the Editor Preferences dialog box is not checked, you are not prompted to save your changes when you close an SQL Editor window or run SQL code from an open SQL Editor window.

If the Prompt To Save option in the Editor Preferences dialog box (Display page) is checked, a confirmation dialog box lets you save or cancel your changes when you close an SQL Editor window or run a query from an open SQL Editor window.

To view the SQL translation of a QBE query

See also

When you use QBE to query an SQL table, SQL Links attempts to translate your query to an equivalent SQL statement and pass it to the SQL server. The server processes your query, then passes the answer set back to you through SQL Links. Database Desktop lets you view the equivalent SQL statement for the query at any time during query construction, or after it is processed.

To view the SQL translation for a QBE query,

1. Connect to the SQL database as described in Connecting to the SQL server.
2. Use QBE to construct a query to the SQL database.
3. Open the SQL Editor window in one of the following ways:

- Choose Query|Show SQL.
- Click the SQL button.

Database Desktop opens the SQL Editor and displays the SELECT statement for your query.

Note: If the SQL database does not support an equivalent SQL statement for a QBE query, a message confirms that the query is processing in the QBE environment.

Depending on the type of query you create, the SQL translation will be one of the following types of statements:

Desired query result statement	**SQL**
Display specific data	SELECT
Add new data	INSERT
Change existing data	UPDATE
Remove existing data	DELETE

▪

About Local SQL

See also

The Borland Database Engine (BDE) enables access to both local and remote database tables through Local SQL (Structured Query Language). Local SQL (sometimes called "client-based SQL") is a subset of ANSI-92 SQL enhanced to support Paradox and dBASE (standard) naming conventions for tables and fields (called "columns" in SQL).

Local SQL lets you use SQL to query "local" standard database tables that do not reside on a database server (specifically Paradox or dBASE tables) as well as "remote" SQL servers. Local SQL is also essential to make multi-table queries across both local standard tables and those on remote SQL servers.

Naming conventions

For a summary of naming conventions for tables and columns, syntax enhancements, and syntax limitations for Local SQL, see Naming conventions.

SQL statements

The SQL statements are broken down into two different categories: Data Manipulation Language (DML) and Data Definition Language (DDL).

- DML statements are used for selecting, inserting, updating, and deleting table data. Syntax and usage examples are included.
- DDL statements are used for creating, altering, and dropping tables, and for creating and dropping indexes. The DDL transforms directly into BDE function calls. Syntax and usage examples are included.

For a complete introduction to ANSI-standard SQL, see one of the many third-party books.

Naming conventions (Local SQL)

See also

ANSI-standard SQL confines each table or column name to a single word comprised of alphanumeric characters and the underscore symbol (_). Local SQL, however, is enhanced to support more comprehensive names.

Tables

Local SQL supports full file and path specifications for table names. Table names with path or file-name extensions must be enclosed in single or double quotation marks. For example,

```
SELECT * FROM 'PARTS.DBF'
SELECT * FROM "C:\SAMPLE\PARTS.DBF"
```

Local SQL also supports BDE aliases for table names. For example,

```
SELECT * FROM ":PDOX:TABLE1"
```

If you omit the file extension for a local table name, the table is assumed to be the table type specified in the Default Driver setting in the System page of the BDE Configuration Utility, or the default driver type for the standard alias associated with the query or table.

Finally, Local SQL permits table names to duplicate SQL keywords as long as those table names are enclosed in single or double quotation marks. For example,

```
SELECT PASSID FROM "PASSWORD"
```

Columns

Local SQL supports Paradox multi-word column names and column names that duplicate SQL keywords as long as those column names are

- Enclosed in single or double quotation marks
- Prefaced with an SQL table name or table correlation name

For example, the following column name is two words:

```
SELECT E."Emp Id" FROM EMPLOYEE E
```

In the next example, the column name duplicates the SQL DATE keyword:

```
SELECT DATELOG."DATE" FROM DATELOG
```

About Data Manipulation Language (DML) statements

See also

With some restrictions, Local SQL supports the following statements for data manipulation:

- SELECT, for retrieving existing data
- INSERT, for adding new data to a table
- UPDATE, for modifying existing data
- DELETE, for removing existing data from a table

The following sections describe functions available to DML statements in Local SQL.

- Aggregate functions
- String functions
- Date function
- Operators
- Updateable (live) queries

For additional illustrative examples, see:

- DML examples

▪

Live query views (SQL)

See also

SQL Links offers expanded support for both single table and multi-table live query views.

Restrictions on live queries

Single-table queries or views are updateable provided that:

- There are no JOINs, UNIONs, INTERSECTs, or MINUS operations.
- There is no DISTINCT key word in the SELECT.
- Everything in the SELECT clause is a simple column reference or a calculated field, no aggregation is allowed.
- There is no GROUP BY or HAVING clause.
- There are no subqueries that reference the table in the FROM clause and no correlated subqueries.
- Any ORDER BY clause can be satisfied with an index.

Restrictions on live joins

Live joins depend upon composite cursors. Live joins may be used only if:

- All joins are left-to-right outer joins or inner joins.
- All joins are equi-joins.
- All join conditions can be satisfied by indexes (for Paradox and dBASE).
- Output ordering is not defined.
- The query contains no elements listed above that would prevent single-table updatability.

Constraints

You can constrain any live query view by checking Constrained Updates on the SQL page of the Query Properties dialog box. An error will then be returned whenever a modify or insert would cause a new record to disappear from the result set.

Calculated fields

For live query views with calculated fields, the calculated field is updated whenever dependent fields are updated.

▪

Heterogeneous joins

See also

Local SQL supports joins of tables in different database formats; such a join is called a "heterogeneous join."

When you perform a heterogeneous join, you may select a local alias. To select an alias, choose SQL|Select Alias. If you have not selected an alias, Local SQL will attempt to find the table in the current directory of the database which is being used. For example, the alias **:**WORK**:** might be the database handle passed into the function.

When you specify a table name after selecting a local alias:

- For local tables, specify either the alias or the path.
- For remote tables, specify the alias.

The following statement retrieves data from a Paradox table and a dBASE table:

```
SELECT DISTINCT C.CUST_NO, C.STATE, O.ORDER_NO
FROM "CUSTOMER.DB" C, "ORDER.DBF" O
WHERE C.CUST_NO = O.CUST_NO
```

You can also use BDE aliases in conjunction with table names.

INSERT

See also

In Local SQL, INSERT can insert a list of values or values can be obtained from a SELECT statement, a query that returns row values.

Examples

The following statement adds a row to a table, assigning values to two columns:

```
INSERT INTO EMPLOYEE_PROJECT (EMP_NO, PROJ_ID) VALUES (52, "dgpii");
```

The next statement uses SELECT to specify values to insert into a table:

```
INSERT INTO PROJECTS
  SELECT * FROM NEW_PROJECTS
    WHERE NEW_PROJECTS.START_DATE > "6-JUN-1994";
```

UPDATE

See also

There are no restrictions on or extensions to the ANSI-standard UPDATE statement.

DELETE

See also

There are no restrictions on or extensions to the ANSI-standard DELETE statement.

SELECT

See also

The SELECT statement is used to retrieve data from one or more tables. A SELECT that retrieves data from multiple tables is called a "join." Local SQL supports the following form of the SELECT statement:

```
SELECT [DISTINCT] column_list
FROM table_reference
[WHERE search_condition]
[ORDER BY order_list]
[GROUP BY group_list]
[HAVING having_condition]
[UNION select_expr]
```

Except as noted elsewhere, all clauses are handled as in ANSI-standard SQL. Clauses in square brackets are optional.

The column_list indicates the columns from which to retrieve data. For example, the following statement retrieves data from two columns:

```
SELECT PART_NO, PART_NAME
FROM PARTS
```

Choose one of the following topics for more information on using SELECT:

- FROM clause
- WHERE clause
- ORDER BY clause
- GROUP BY clause
- HAVING clause
- UNION clause
- Heterogeneous joins

FROM clause (SELECT statement)

See also

The FROM clause specifies the table or tables from which to retrieve data. Table_reference can be a single table, a comma-delimited list of tables, or can be an inner or outer join as specified in the SQL-92 standard. For example, the following statement specifies a single table:

```
SELECT PART_NO
FROM "PARTS.DBF"
```

The next statement specifies a left outer join for table_reference:

```
SELECT * FROM PARTS LEFT OUTER JOIN INVENTORY
ON PARTS.PART_NO = INVENTORY.PART_NO
```

▪

WHERE clause (SELECT statement)

See also

The optional WHERE clause reduces the number of rows returned by a query to those that match the criteria specified in search_condition. For example, the following statement retrieves only those rows with PART_NO greater than 543:

```
SELECT * FROM PARTS
WHERE PART_NO > 543
```

The WHERE clause can include the IN predicate, followed by a parenthesized list of values. For example, the next statement retrieves only those rows where a part number matches an item in the IN predicate list:

```
SELECT * FROM PARTS
WHERE PART_NO IN (543, 544, 546, 547)
```

ORDER BY clause (SELECT statement)

See also

The ORDER BY clause specifies the order of retrieved rows. For example, the following query retrieves a list of all parts listed in alphabetical order by part name:

```
SELECT * FROM PARTS
ORDER BY PART_NAME ASC
```

The next query retrieves all part information ordered in descending numeric order by part number:

```
SELECT * FROM PARTS
ORDER BY PART_NO DESC
```

Calculated fields can be ordered by correlation name or ordinal position. For example, the following query orders rows by FULL_NAME, a calculated field:

```
SELECT LAST_NAME || ', ' || FIRST_NAME AS FULL_NAME, PHONE,
FROM CUSTOMER
ORDER BY FULL_NAME
```

▪

GROUP BY clause (SELECT statement)

See also

The GROUP BY clause specifies how retrieved rows are grouped for aggregate functions.

HAVING clause (SELECT statement)

See also

The HAVING clause specifies conditions records must meet to be included in the return from a query. It is a conditional expression used in conjunction with the GROUP BY clause. Groups that do not meet the expression in the HAVING clause are ommitted from the result set.

Subqueries are supported in the HAVING clause. A subquery works like a search condition to restrict the number of rows returned.

In addition to scalar comparison operators (=, <, > ...) additional predicates using IN, ANY, ALL, EXISTS are supported.

UNION clause (SELECT statement)

See also

The UNION clause combines the results of two or more SELECT statements to produce a single Answer table.

-

Aggregate functions (SQL)

See also

The following ANSI-standard SQL aggregate functions are available to Local SQL for use with data retrieval:

- SUM(), for totaling all numeric values in a column
- AVG(), for averaging all non-NULL numeric values in a column
- MIN(), for determining the minimum value in a column
- MAX(), for determining the maximum value in a column
- COUNT(), for counting the number of values in a column that match specified criteria
- COUNT(*), for counting non-NULL numeric values in a column

Complex aggregate expressions are supported, such as:

```
SUM( Field * 10 )
SUM( Field ) * 10
SUM( Field1 + Field2 )
```

▪

String functions (SQL)

See also

Local SQL supports the following ANSI-standard SQL string manipulation functions for retrieval, insertion, and updating:

- UPPER(), to force a string to uppercase
- LOWER(), to force a string to lowercase
- TRIM(), to remove repetitions of a specified character from the left, right, or both sides of a string
- SUBSTRING() to create a substring from a string

substring

SUBSTRING() takes a string and creates a substring of that string.

```
   SELECT SUBSTRING( CUSTNAME FROM 1 FOR 10 ) FROM CUSTOMER
```

This query return the first 10 characters of the CUSTNAME column.

You could also use the SUBSTRING expression `SUBSTRING( CUSTNAME FROM 1)`, which starts returning characters at the specified number and continues to the end.

▪
Date functions

See also

Local SQL supports the EXTRACT() function for isolating a single numeric field from a date/time field on retrieval using the following syntax:

```
EXTRACT (extract_field FROM field_name)
```

For example, the following statement extracts the year value from a DATE field:

```
SELECT EXTRACT(YEAR FROM HIRE_DATE)
FROM EMPLOYEE
```

You can also extract MONTH, DAY, HOUR, MINUTE, and SECOND using this function.

Note: EXTRACT does not support the TIMEZONE_HOUR or TIMEZONE_MINUTE clauses.

Operators (SQL)

See also

Local SQL supports the following operators:

Type	Operator
Arithmetic	+, –, *, /
Comparison	<, >, =, <>, IS NULL, IS NOTNULL, >=, =<
Logical	AND, OR, NOT
String concatenation	||
String pattern match	LIKE

▪

DML examples

See also

The DML syntax supports these clauses:

```
   SELECT FROM, WHERE, ORDER BY, GROUP BY, and HAVING
```

The following aggregates are supported:

```
   SUM, AVG, MIN, MAX, COUNT
```

The following operators are supported:

```
   +, -, *, /, =, < >, IS NULL, IS NOTNULL, >=, =<, AND, OR, NOT, ||, LIKE
```

UPDATE, INSERT, DELETE operations are fully supported.

The following examples show DML statements used with standard databases:

Example 1: UPDATE

```
update goods
  set city = 'Santa Cruz'
  where goods.city = 'Scotts Valley'
```

Example 2: INSERT

```
insert
  into goods ( part_no, city )
  values ( 'aa0094', 'San Jose' )
```

Example 3: DELETE

```
delete
  from goods
  where part_no = 'aa0093'
```

Example 4: SELECT used to join

The following example illustrates how the SELECT statement is supported as an equivalent to a JOIN:

```
select distinct p.part_no, p.quantity, g.city
  from parts p, goods g
  where p.part_no = g.part_no
  and p.quantity > 20
  order by p.quantity, g.city, p.part_no
```

A SELECT statement that contains a join must have a WHERE clause in which at least one field from each table is involved in an equality check.

Example 5: Sub-selects

Sub-select queries are supported. The following example illustrates this syntax:

```
select p.part_no
  from parts p
  where p.quantity in
   (select i.quantity
     from inventory i
     where  i.part_no = 'aa9393')
```

Example 6: GROUP BY

The following examples illustrate the GROUP BY clause:

```
select part_no, sum(quantity) as PQTY
  from parts
  group by part_no
```

Note: Aggregates in the SELECT clause must have GROUP BY clause if a projected field is used, as shown in the first example above.

Example 7: ORDER BY

The following example illustrates the ORDER BY with a DESCENDING clause:

```
select distinct customer_no
  from c:\data\customer
  order by customer_no descending
```

▪

About Data Definition Language (DDL) statements

See also

Local SQL supports Data Definition Language (DDL) for creating, altering, and dropping tables, and for creating and dropping indexes.

Views are supported.

Local SQL does not permit the substitution of variables for values in DDL statements.

The following DDL statements are supported:

- CREATE TABLE
- ALTER TABLE
- DROP TABLE
- CREATE INDEX
- DROP INDEX
- CREATE VIEW

For additional illustrative examples see:

- DDL examples

▪

CREATE TABLE

See also

CREATE TABLE is supported with the following limitations:

- Column definitions based on domains are not supported.
- Constraints are limited to PRIMARY KEY for Paradox tables. Constraints are unsupported in dBASE tables.

For example, the following statement creates a Paradox table with a PRIMARY KEY constraint on the LAST_NAME and FIRST_NAME columns:

```
CREATE TABLE "employee.db"
(
LAST_NAME CHAR(20),
FIRST_NAME CHAR(15),
SALARY NUMERIC(10,2),
DEPT_NO SMALLINT,
PRIMARY KEY(LAST_NAME, FIRST_NAME)
)
```

The same statement for a dBASE table should omit the PRIMARY KEY definition:

```
CREATE TABLE "employee.dbf"
(
LAST_NAME CHAR(20),
FIRST_NAME CHAR(15),
SALARY NUMERIC(10,2),
DEPT_NO SMALLINT
)
```

Creating Paradox and dBASE tables

You create a Paradox or dBASE table using Local SQL by specifying the file extension when naming the table:

- ".DB" for Paradox tables
- ".DBF" for dBASE tables

If you omit the file extension for a local table name, the table created is the table type specified in the Default Driver setting in the System page of the BDE Configuration Utility.

Data type mappings for CREATE TABLE

The following table lists SQL syntax for data types used with CREATE TABLE, and describes how those types are mapped to Paradox and dBASE types by the BDE:

SQL Syntax	BDE Logical	Paradox	dBASE
SMALLINT	fldINT16	Short	Number (6,10)
INTEGER	fldINT32	Long Integer	Number (20,4)
DECIMAL(x,y)	fldBCD	BCD	N/A
NUMERIC(x,y)	fldFLOAT	Number	Number (x,y)
FLOAT(x,y)	fldFLOAT	Number	Float (x,y)
CHARACTER(n)	fldZSTRING	Alpha	Character
VARCHAR(n)	fldZSTRING	Alpha	Character
DATE	fldDATE	Date	Date
BOOLEAN	fldBOOL	Logical	Logical
BLOB(n,1)	fldstMEMO	Memo	Memo
BLOB(n,2)	fldstBINARY	Binary	Binary
BLOB(n,3)	fldstFMTMEMO	Formatted memo	N/A

BLOB(n,4)	fldstOLEOBJ	OLE	OLE
BLOB(n,5)	fldstGRAPHIC	Graphic	N/A
TIME	fldTIME	Time	N/A
TIMESTAMP	fldTIMESTAMP	Timestamp	N/A
MONEY	fldFLOAT, fldstMONEY	Money	Number (20,4)
AUTOINC	fldINT32, fldstAUTOINC	Autoincrement	N/A
BYTES(n)	fldBYTES(n)	Bytes	N/A

x = precision (default: specific to driver)
y = scale (default: 0)
n = length in bytes (default: 0)
1-5 = BLOB subtype (default: 1)

ALTER TABLE

See also

Local SQL supports the following subset of the ANSI-standard ALTER TABLE statement. You can add new columns to an existing table using this ALTER TABLE syntax:

```
  ALTER TABLE table ADD column_name data_type [, ADD column_name
data_type ...]
```

For example, the following statement adds a column to a dBASE table:

```
  ALTER TABLE "employee.dbf" ADD BUILDING_NO SMALLINT
```

You can delete existing columns from a table using the following ALTER TABLE syntax:

```
  ALTER TABLE table DROP column_name [, DROP column_name ...]
```

For example, the next statement drops two columns from a Paradox table:

```
  ALTER TABLE "employee.db" DROP LAST_NAME, DROP FIRST_NAME
```

ADD and DROP operations can be combined in a single statement. For example, the following statement drops two columns and adds one:

```
  ALTER TABLE "employee.dbf" DROP LAST_NAME, DROP FIRST_NAME, ADD FULL_NAME
CHAR[30]
```

DROP TABLE

See also

DROP TABLE deletes a Paradox or dBASE table. For example, the following statement drops a Paradox table:

```
  DROP TABLE "employee.db"
```

CREATE INDEX

See also

CREATE INDEX enables users to create indexes on tables using the following syntax:

```
CREATE INDEX index_name ON table_name (column [, column ...])
```

Using CREATE INDEX is the only way to create indexes for dBASE tables. For example, the following statement creates an index on a dBASE table:

```
CREATE INDEX NAMEX ON "employee.dbf" (LAST_NAME)
```

Paradox table users can create only secondary indexes with CREATE INDEX. Primary Paradox indexes can be created only by specifying a PRIMARY KEY constraint when creating a new table with CREATE TABLE.

Note: The index created in nonmaintained, nonunique, not case-sensitive, and in ascending order. If the table has a primary key, then a maintained index is created.

DROP INDEX

See also

Local SQL provides the following variation of the ANSI-standard DROP INDEX statement for deleting an index. It is modified to support dBASE and Paradox file names.

```
  DROP INDEX table_name.index_name | PRIMARY
```

The PRIMARY keyword is used to delete a primary Paradox index. For example, the following statement drops the primary index on EMPLOYEE.DB:

```
  DROP INDEX "employee.db".PRIMARY
```

To drop any dBASE index, or to drop secondary Paradox indexes, provide the index name. For example, the next statement drops a secondary index on a Paradox table:

```
  DROP INDEX "employee.db".NAMEX
```

CREATE VIEW

See also

A view creates a virtual table from a SELECT statement. You can look at just the data you need within this movable frame or window on the table, while the technical underpinnings are hidden. Instead of entering a complex qualified SELECT statement, the user simply selects a view.

CREATE VIEW describes a view of data based on one or more underlying tables in the database. The rows to return are defined by a SELECT statement that lists columns from the source tables. A view does not directly represent physically stored data. It is possible to perform select, project, join, and union operations on views as if they were tables.

CREATE VIEW enables users to create views on tables by using the following syntax:

```
  CREATE VIEW view_name [ (column_name [, column_name]...)]
```

CREATE VIEW is supported in conjunction with the Client Data Repository (CDR). The CDR stores the SELECT statement that defines the view.

The "WITH CHECK OPTION" is supported to create a constrained view.

Views of Views are supported. However, the CASCADE/LOCAL view attribute is not supported, because all updateable views CASCADE the constraints.

DDL examples

See also

The following examples show the use of DDL statements with standard databases.

Example 1a: DDL (DROP TABLE)

When the table name contains a period "." character, enclose the name in quotation marks:

```
drop table "c:\data\customer.db"
```

Example 1b: DDL (DROP TABLE)

No quotation marks are used if the table name does not contain the "." character:

```
drop table clients
```

Example 2: DDL (CREATE INDEX)

```
create index part on parts (part_no)
```

Paradox: Paradox primary indexes can be created only when creating the table. Secondary indexes are created as case insensitive and maintained, when possible.

dBASE: dBASE indexes are created as maintained. The Index name specified is the tag name.

For more information about different types of indexes, see DbiAddIndex in the *Borland Database Engine Online Reference*.

Example 3: DDL (DROP INDEX)

The syntax for drop index is tablename.indexname:

```
drop index parts.part_no
```

Paradox: For Paradox only, the syntax tablename.primary indicates the primary index:

```
drop index parts.primary
```

▪

Ordering a SQL Links disk set

See also

SQL Links 3.0 provides you with a single user/developer license.

Borland's SQL Links Deployment License is required to distribute SQL Links with any applications. This deployment license allows a single client/server developer to deploy an unlimited number native SQL Links 3.0 with his/her applications to an unlimited number of servers for use by end-users.

To order a SQL Links disk set,

- In the U.S., please fax your order to 510-657-0186. For faster service, call toll-free 1-800-682-9229.
- In Canada, please fax your order to 1-800-825-2225. For faster service, call toll-free 1-800-461-3327.
- Or, choose File|Print Topic to print this topic, complete the information, and mail it to:

U.S. Customers	**Canadian Customers**
SQL Links Order Processing	Borland Canada
Borland International, Inc.	200 Konrad Crescent
100 Borland Way	Markham, Ontario
P. O. Box 660005	Canada L3R8T9
Scotts Valley, CA 95067-0005	

(Please print)

Product Name____

Name____

Company Name____

Street Address____

City___________________________________ State___________

Zip/Postal Code_________________ Country_______________________

Daytime Phone (________)____

(in case we have a question about your order)

The cost of the SQL Links disk set is U.S.$9.95 plus $5.00 shipping and handling and sales tax where applicable.*

SQL Links disk	$ 9.95	*Residents in CA, CO, CT, DC, FL, GA, IL, MA, MD, MI,
State sales tax	$________	MN, MO, MC, NJ, NY, OH, PA, TN, TX, UT, VA, WA:
Freight*	$ 5.00	Please add appropriate sales tax. CO, MI, NY, PA,
TOTAL	$________	TX, WA residents: please add tax to freight charges

Please make your check payable to Borland International, Inc. and enclose it with this form. Or fill in your credit card name, number, expiration date, your printed name, and signature.

___ Check enclosed ___ MasterCard ___ Visa ___ American Express

Card number __ __ __ __ - __ __ __ __ - __ __ __ __ - __ __ __ __

Expiration date __ __ - __ __

Name as it appears on the card___________________________________

Signature____

Please allow two to three weeks for delivery. Pricing and offer good in the United States only.

For international orders: Please refer to the Borland Registration card for the telephone number of the Borland office nearest you. Prices and shipping may vary by country.