Command Line

When launching SQLing from the Command Line you can include the following options in arbitrary order:

Automatic connection to the SQL Server

[-O**oledb_provider**] -S**server_name0** [-D**database_name0** -U**user_name0** -P**user_password0** -T] ensures automatic connection to the SQL Server
The -O option means that the connection will be of OLE DB type. Other connect options will be interpreted as follows: server_name = data dource; database_name = location; the remaining parameters have the same meaning.
Include the -T option for trusted connection. If -D, -U, -P are not specified, the following default values are used: user=sa, password = empty string, current database will be set as the default database of user **sa**.
Other command line options include:

-S1**server_name1** [-D1**database_name1** -U1**user_name1** -P1**user_password1** -T1] ,

these specify a second connection to the SQL Server configured as above.

If both server connections were successful, SQLing selects **server0.database0** as the current (active) connection.
Note: Because server can be also IP address, it can be delimitated by the following brackets "{}". (e.g. -S1{192.126.13.15})
As soon as connection to server is achieved, the command line is checked for the following options:

Load file into query editor.

-Q**file_name** opens the query window and loads the file specified by **file_name**

-L**saved_downsize** loads the saved downsize of the current database (which has to be set in the -D option!) upsizes it without asking and without performing the Save Downsize command (if **Save Downsizes** is enabled in options). If an error occurs while the Upsize script is generated, an error message appears and SQLing remains running. SQLing closes if no errors occur. Set the -L option to simplify synchronization of the actual database structure with the development database, as any downsized databases are saved before Upsize. You can avoid having to launch the sequence **Load downsize, Upsize, Load downsize, Upsize, Load downsize, Upsize** manually if you make a batch file (.BAT) which runs SQLing with the -L command line parameter.

The following options automate activities so that SQLing can be used by inexperienced users. Create batch files, which launch SQLing with one of the following command line options.

Compare the content of selected tables between two databases

-Cd**data_compare_settings_file_name.tse** Performs command-line comparison of data in between the selected tables. This option is corresponds to the following sequence of operations:

1. Set **server_name0** and **database_name0** as the current server and database.

2. Select Data in tables … in menu **Current Database/Compare.**
3. Select **Load Settings** from the File menu of the dialog box that appears and load file **data_compare_settings_file_name.tse** (which may be specified after the -Cd parameter).
4. Click on **Start** to start comparing.
5. Save the generated results in file **server_name0_database_name0__dserver_ddb.txt** located in the folder specified in the **In Directory** field of the Compare Dialog Box. In case no folder is specified the file is created in the current folder. Parameters **dserver** and **ddb** specify the server and the database with which the current database has been compared. The server and database were specified in the **.tse** file used in step 3.
6. Close SQLing.

Compare the structure between two databases

-CS compared the structure of two databases, **server0.database0** and **server1.database1**. Results are stored in file **strdiff_server0_database0__server1_database1.txt** located in the current folder. When the comparison is finished, SQLing closes automatically.

Compare procedures between two databases

-CP compares the source code of all stored procedures in two databases, **server0.database0** and **server1.database1**. Results are exported to file **codediff_proc_server0_database0__server1_database1.txt** located in the current folder. Whenever procedures differ at a syntactically important point a descriptive line is saved into the file. An additional settings file named **pc_proc_database0.tse** is created for the Synchronize function. When this file is opened by selecting Load from menu File of the Synchronize dialog window a synchronization script is created. This script eliminates the differences between procedures. SQLing closes after the comparison.

Compare triggers between two databases

-CT compares the source codes of all triggers in two databases, **server0.database0** and **server1.database1**. Results are saved in file **codediff_trig_server0_database0__server1_database1.txt** located in the current folder. Whenever triggers differ at a syntactically important point a descriptive line is saved into the file. An additional settings file named **pc_trig_database0.tse** is created for the Synchronize function. When this file is opened by selecting Load from menu File of the Synchronize dialog window a synchronization script is created. This script eliminates the differences between triggers. SQLing closes after the comparison.

Compare script containing procedures with in-database procedures

-CF**script_file_name.sql** compares the source codes of procedures and other source-code objects scripted in a file with those located on server **server_name0,** database **database_name0**. SQLing generates a file **codefiff_server0_database0.txt** reporting differences similarly to -CT and -CP options.
SQLing closes automatically. This function has no non-command line equivalent.

Synchronize databases

-CY**synchronize_settings_file_name.tse** wholly or partially synchronizes a database.
Its function is identical with the following sequence of operations:

1. Set the current server and database as **server_name0, database_name0**.
2. Select Synchronize in the **Current Database** menu.
3. Open file **synchronize_settings_file_name.tse** (also specified after the -CY option) by selecting **Load Settings** in the File menu of the Dialog Box that appears.
4. Press **Start** to start the synchronization.
5. Close SQLing.

Offline database export

-CI**offline_export_settings_file_name.tse** exports a database wholly or partially while offline.
Its function is identical with the following sequence of operations:

1. Set the current server and database as **server_name0, database_name0**.
2. Select Offline Export … from menu **Current Database/Transfer.**
3. Open file **offline_export_settings_file_name.tse** (also specified after the -CI option) by selecting **Load Settings** in the File menu of the Dialog Box that appears.
4. Press **Transfer** to start the offline transfer.
5. Close SQLing.

Create database script

-CC**offline_export_settings_file_name.tse** creates a script of the database, wholly or partially.
Its function is identical with the following sequence of operations:

1. Set the current server and database as **server_name0, database_name0**.
2. Select Offline Export … from the menu **Current Database/Transfer.**
3. Open file **offline_export_settings_file_name.tse** (also specified after the -CI option) by selecting **Load Settings** in the File menu of the Dialog Box that appears.
4. Press **Create Script** to generate the script.
5. Close SQLing.

Offline database import

-CJ**directory** launches offline import from the folder **directory**.

Print or export report

-RN**report_name** - specifies a report (database and server are specified in -S and -D) to be printed or exported to file. Unless this option is specified all other options described in this paragraph are ignored.
-RP**parameter_set_name** - if the report is parameter-based (this is determined in Report Designer Properties) you can specify the name of parameter set for which reports should be generated.
-RF**export_file_name** - specifies the file to which report data should be exported in tabulated format. If this option is not specified the report is printed on the default printer.

-RP**printer_name** - name of the printer on which the report is to be printed. Enter printer name as \\server\printer. Only specify this option if you do not wish to use the default printer or if no default printer is defined for the user who is running SQLing.
-RL - it is not desirable that error messages waiting for confirmation appear when the application is launched from the command line. This option ensures that any error messages are stored in table SQLing..report_error and that no dialog windows are displayed.
Because the names of reports, parameter sets and printers may contain spaces, it is advisable to enclose any such names in quotation marks (””).
In case that SQLing fails to print a report it may be necessary to check the SQLing..report_error table for any error messages that could explain what went wrong.

Log errors and suppress error messages

-RL - if this option is used without the -RN switch it ensures that no error message/query dialog opens and that any error messages are logged in the file $temp\SQLingBlackBox.txt.

Export/Import text data

The Offline Export function stores table data in bulkcopy (.bcp) files which may then be imported by Offline Import into newly created tables. Because these files are binary, problems may occur if they are used to transfer data between Pentium and Alpha -based servers. These processors use different binary data encoding patterns. The **-FC** command line switch solves this by ensuring that the .bcp files are not generated as binary, but as text.

Debug mode

If the application is launched with the -debug switch the Upsize function prompts before its execution whether only a modification script should be created or whether the script should be created, implemented and the working database erased.

Help

The last supported option switch is -?. It lists all possible options.

SQLing Server Options

The **SQLing** database on the SQL Server contains a table named **options,** which has two rows. Both specify the behavior of SQLing on the current server.
The row entitled **Save downsizes** specifies whether the downsized database should automatically be saved before every Upsize. Database is saved by the Save Downsize function into the file named **wns_server_database.pdbxxxx** where **database** is the name of the downsized database and **xxxx** is a four-digit file version number. The version number begins at 0001 and is increased with each new upsize of the database. The file is created in the folder specified in column **opt_text** of the **Save downsizes** row. Downsizes are only automatically saved if the opt_int column contains a non-zero number.

The row entitled **Don't rollback (in trigger) in nested transaction** specifies how triggers ensuring reference integrity should be generated. If the **opt_int** column in this row contains a non-zero value the condition *IF @@TRANCOUNT=1* is inserted before every *ROLLBACK TRAN* during trigger source code generation. This ensures that the triggers generated by SQLing do not rollback attempts to disrupt reference integrity which occur during a Transaction. It has proved inconvenient for triggers to rollback nested transactions because a *ROLLBACK TRAN* rolls back to the first *BEGIN TRAN.* The procedure which launched the trigger would not expect this and as a result, inconsistencies in tables could arise.
The row entitled **Historic Table Indexes** specifies which indexes are generated for historic tables. This information is located in column **opt_text** as text which consists of the following characters:

- p - generate index for the primary key of the historic table
- i - generate index for the inherited primary key of the monitored table
- d - generate index of modification date
- c - generate index for column with the ordinal number of the modified attribute

If the string consist of all these characters, all indexes will be generated for the historic tables. If the string only contains some of the characters, only the respective indexes are generated.

0 The column entitled **last_update** in the table **options** contains the NT user name of the last user who changed any value in the row.

1 Tip:
2 If the user decides to regenerate all triggers in a database after changing the **Don't rollback (in trigger) in nested transaction** option, the following steps should be made:

1. Change the option setting.
0 2. Downsize the current database.
1 3. Use the function Current Database/Touch to set the regeneration of all triggers.
2 4. Upsize to the actual database.

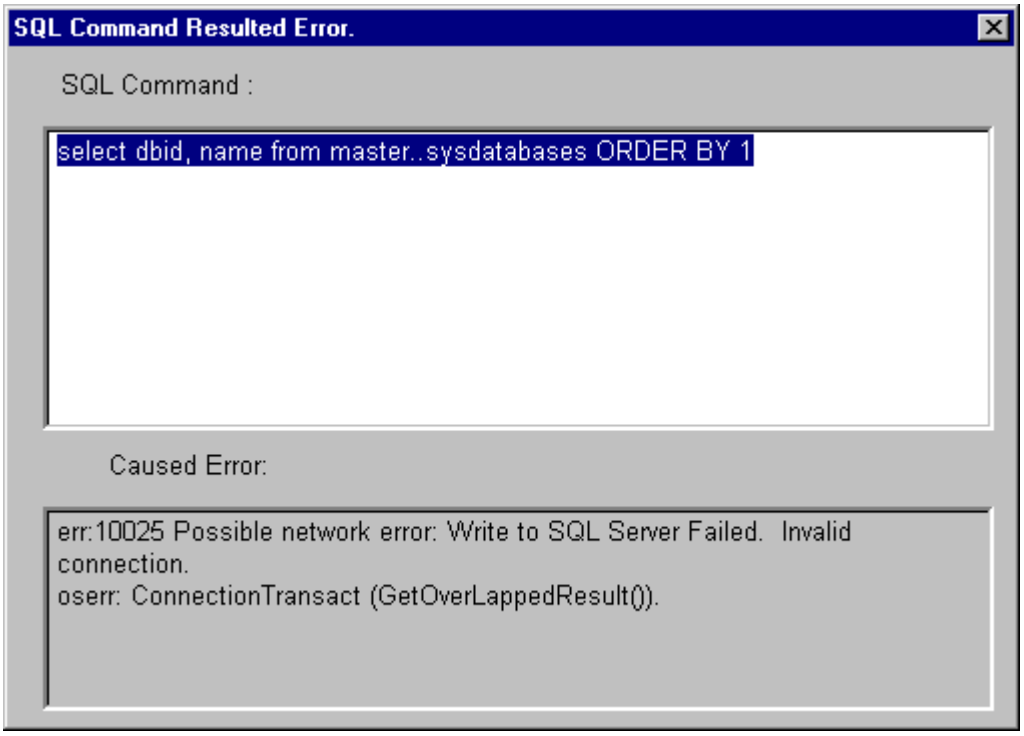
Basic Application Principles.

Error Handling
Detailed Description of the Upsize and Downsize Functions
Structure of the SQLing database
SQLing Server Options

Error Handling

0 Unexpected Errors

Handling of application errors arising unexpectedly. An error might either occur through incorrect use of SQLing or by a failure on the side of the SQL Server, communication breakdown or any other exception. In such cases, an error message dialog box appears.

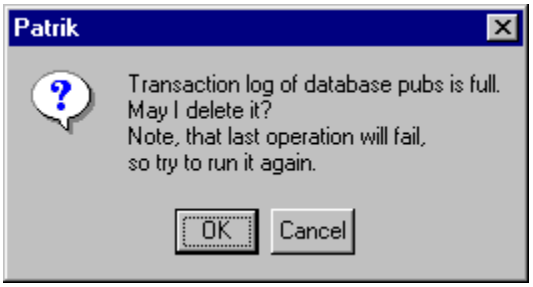


0

1 The upper part of the window contains the command (or a description of command) that was to be executed and the lower part contains the error message. The Dialog Box should facilitate error diagnostics.

2 Transaction log overflow.

3 From time to time the database transaction log may overflow. In such cases, this question appears:



4

5 The user is asked whether the log should be dumped immediately.

Process stall

One of the possible exceptions is a process stall (kill). The standard SQL Error Dialog Box appears and the process (Connect) will become unusable. Connection to the server where the process stalled will have to be renewed.

Detailed Description of the Upsize and Downsize Functions

Downsize

The process of downsizing is the creation of a database which is structurally identical with the actual database but contains no data. Downsizing involves the creation of system-like tables in the **SQLing** database and copying the source database system tables into these new tables. The layer of the source database which was added by SQLing is then copied: this layer includes comments, relation views, table positions in the relations etc. In the process, only tables stored in the **SQLing** database are replicated into themselves. In the end, a record is added into table **SQLing..workdb**. This table contains information about downsized databases. Once connected to the server, it is this table and the sysdatabases table that are used to retrieve the list of databases which can be browsed in SQLing. Downsize will not proceed and an error message is displayed in the following cases:

1. If the source database contains tables with the prefix **upsizetmp_** (refer to section on Upsize). Delete any such tables and re-run Downsize.
02. If some other user has already downsized the source database.
13. If the last Downsize did not finish correctly or completely. In such case, run the **DropDatabase.sql** script located in the SQLing install folder and then execute the new procedure **SQLing..DropDatabase** with the name of the incompletely downsized database as a parameter. You can obtain this name from the **SQLing** database if you are familiar with the Downsize process described at the end of the section named *Structure of the SQLing Database*.
24. If the **SQLing** database is not compact. In such case, run the script **CompactMetaDB.sql** located in the install folder of SQLing.

0 Drop Working Database.

1 Dropping a database which was created by this process only involves the following steps: remove the record in **SQLing..workdb** which refers to the database, drop the copies of system tables in the **SQLing** database, delete the appropriate records from the standard tables of the **SQLing** database.

Upsize

Upsizing is the application of the changes made in the working database to the actual database. This process efficiently uses the resources because it only modifies objects which were changed in the working database. First, user data types are modified (added, deleted), next users and then tables. The types of changes to tables which are recognized by SQLing can be divided into three groups:

1. Add or Remove Table.
0 2. Simple modification of a table - can be effected by T-SQL commands and/or calls to system stored procedures. This type of change is supported by many of the database managers existing on the market.
1 3. Complex table modification - has to be brought about by means of a little trick. The existing table is renamed (temporary names like **upsizetmp_xx** are used,

where xx is a number), a new table is created, retaining the properties of the old one, but incorporating those of the table in the working database, transfer of table content, dropping of the old table and the establishment of relations and triggers over the new table. The establishment of relations and triggers only after the data has been transferred is necessitated by the unfavorable referential integrity.

When changes are made to tables the triggers and relations of modified tables are updated and in the end the working database is deleted (dropped) and the result set is displayed. It is important to be able to read the results of the Upsize process. Do not be surprised if you see a message saying that an object was successfully renamed although you were not renaming anything. The renamed object is very likely a table of the third category. Practically no errors caused by SQLing should appear during Upsizing after the initial message "Foreign Key Dropped". Any errors are probably caused by the user's lack of attention or a SW/HW/POWER failure. Loss of data is practically impossible. The only possibility of any loss of data from the original table (though the data would not be erased from the server) is while modifying a table of the third type. In such case failure to transfer data from the temporary table **upsizetmp_xx** to the original table could occur but the respective temporary table would not be dropped so the data would remain on the server. The user would then have to manually transfer the data from the table named **upsizetmp_xx** (*< INSERT … SELECT … FROM upsizetmp_xx >)* and drop the temporary table. Because no databases containing any **upsizetmp_xx** tables can be Downsized, these tables should be deleted.

0 Upsize consists of three parts.

1. Create Upsize-script - <name_downsize-u>.sql.

0 2. Execute Upsize-script.

1 3. Drop the working database.

An error message informing the user of any potential negative effects of the Upsize script may be displayed between phases 1 and 2. You can still halt the Upsize process at this point.

0 When upsizing is finished the results are saved in the file <name_downsize-u>.txt and displayed on the screen in a new window.

1 For completeness, here is a description of the structure of the SQLing database .

Structure of the SQLing Database.

Knowledge of the structure of the ”SQLing” database is not required before one begins to work with the SQLing application. This chapter is intended for advanced users of the SQL Server who may use their skills to many purposes, some of which are mentioned at the end of this section.
The database consists of the following tables:

rel - this table stores the positions of windows which display table attributes in the Relational

- **db - name of the database to which the table belongs**
- **tbl - table name**
- **vy - coordinates of the top left corner of the table window**
- **cx - window width**
- **cy - window height**

2 This table is always read when loading/refreshing relations and stored when Save Layout is selected from the shortcut menu of the relational model window.

3 Tables describing the generated triggers.

4 Triggers have to be re-generated during upsizing because of the changes made to the downsized database. The triggers which maintain referential integrity contain part that is modified in the re-generation process. Such part in the body of a trigger begin with the comment /*SQLingB*/ and end with /*SQLingE*/. The rest of the trigger contains the Business Rules encoded by the user and is not re-generated during upsizing. This part that has to be newly generated is based on the **keysadd** table. If a table has any non-constraint relations recorded in the **keysadd** table by a non-zero **flags** attribute, then a sub-routine is linked to the trigger to ensure that this referential integrity is not destroyed. Such sub-routine is located at the beginning of the trigger to detect any disruption and to cancel the transaction before any Business Rules initiate.

5 After upsize SQLing uses the information in this table instead of parsing the source code of the triggers. If any records were deleted from the above table then the triggers would again be harmonized with the table after an upsize. This would result in a different database condition than was originally defined by the designer. For this reason it is advisable to transfer the **keysadd** table together with the main database, either by using the function **Transfer/Metadata Layer …** or by making sure that the database transfer is handled by SQLing. It is not necessary to transfer these tables if the database will not be modified by SQLing on its new destination.

Keysadd - this table contains properties and descriptions of relations. It also contains records on constraint relations for commenting purposes. In such case the flags value is not relevant.

db - database into which the relation belongs
id - table name on the *many* side of the relation (child table)
depid - table name on the *one* side of the relation (parent table)
flags - relation properties encoded by the following bits:
- **1 - one to one relationship**
- **02 - delete trigger exists (or has to be created)**
- **14 - update trigger exists**
- **28 - delete trigger deletes child table rows in a cascade**
- **316 - update trigger updates child table rows in a cascade**

str - relation comment up to 255 characters

Comments

0 Comments or notes to objects are maintained in two tables, **conspects** and **notes**, the structure of which is obvious.

1 Relational Model Views.

2 The table **rel_view** contains a record for every view, consisting of its name and creator. The view structure is stored in table **view_item**, which is linked to **rel_view** by a *One to Many* relation. The **view_item** table contains the position and visibility of every table included in the particular view.

3

rel_view - views table

- id - primary table key
- name - view name
- db - database to which the view belongs
- usr - user who created the view

view_item - this table contains the positions of windows which display attributes of tables in the Relational Model window.

- **id - the view to which the table belongs**
- **tbl - table name**
- **vy - coordinates of top left corner of the table window**
- **cx - width and height of the window**
- **flags - table visibility**

- ***Workdb - contains information on fictitious (downsized) databases***

- **name - name of the fictitious database without the ”#” prefix**
- **sourcedb - name of the source (downsized) database**
- **status - database status. Can be set in menu ”Current Database/Options” of SQLing.**

2 The SQLing database also contains the Options table and tables storing definitions of reports. These are not described here.

3 This was a description of the tables contained in the SQLing database. Along with these tables a cascaded delete trigger on the **rel_view** table was created to delete window positions in **view_item** when the related view is deleted.

4 Tables created with the Upsize function.

5 More tables appear in the SQLing database after a downsize operation. If, for instance, a database named ”pubs” was downsized as a fictitious database ”work”, the following tables would be created in the SQLing database during upsizing:

6 sysalternates_work, syscolumns_work, syscomments_work, sysconstraints_work, sysidentities_work, sysobjects_work, syskeys_work, sysprotects_work, sysreferences_work, systypes_work and **sysusers_work.**

7 All these tables have the same structure as their counterparts (without the suffix) in database ”pubs”. SQLing utilizes some of the unused columns, described below:

8 sysobjects.sysstat - describes the changes which happened to an object. This field is used during a database upsize. The following bits can be set in this field:

9 1 - table index was changed
10 2 - primary table key was changed
11 4 - data type of a table column was changed
12 8 - name of a table column was changed
13 16 - rule or default of a table column was changed
14 32 - the table was ”significantly” modified
15 64 - new column was added
16 128 - table name was changed
17 256 - table was added into database
18 512 - the comment for a table column was changed
19 1024 - table IDENTITY was changed
20 2048 - check or default constraint for a table column was changed
21 4096 - relation for the table was changed

22 systypes.printfmt - describes changes that happened to a data type

- 1 - user group was changed
- 2 - user rights were changed

- 4 - user was significantly changed (other changes)
- 8 - user was added to database

2 sysusers.environ - describes changes that happened to database user

3 Another table is copied into the SQLing database during downsizing. The structure of this table is different from its model table in the downsized database (pubs, in our case).

- ***created table sysindexes_work***
- **name** index name
- **id** id of the table to which the index belongs
- **indid** index id
- **segment** number of segment where index is created
- **status** index type
- **fill** Reserved
- **keys** of type char(65), contains numbers of columns to which the index relates, ordered by size and separated by commas (columns are numbered from 1)

Together with the above tables the cascade delete trigger is created on the **sysobjects_work** table. This trigger deletes rows in child tables **syscolumns_work** and **sysindexes_work**.

Why know anything about these tables? Here are several possible applications of such knowledge:

1. When upsizing a database with the MS Access Upsizing Tool the Access fields of the type text are converted to varchar fields in SQL-i. Because work with the varchar type is more awkward it is convenient to change the varchar types to char in SQLing and upsize the database from SQLing. This would be simple if the database contained a few tables but who would be willing to do all the necessary changes in a large database? With the knowledge of the ”SQLing” database tables, however, the procedure could be as follows:

- *Downsize the database as ”work”*
- *Run the following query in the Query Editor:*

```
UPDATE SQLing..sysobjects_work  SET sysstat=sysstat|4
FROM SQLing..sysobjects_work a, SQLing..syscolumns_work b
WHERE a.type='U' AND a.id=b.id AND b.usertype=2
```

```
UPDATE SQLing..syscolumns_work SET type=47, usertype=1
WHERE usertype=2 AND name NOT LIKE('@%')
```

- *Upsize database ''work''.*

2. If for various reasons (e.g. power failure) the database does not downsize completely, the user knows which tables can be dropped from the SQLing database. To drop these tables comfortably, use the **DropDatabase.sql** script supplied with SQLing.

See Detailed Description of the Upsize and Downsize functions .

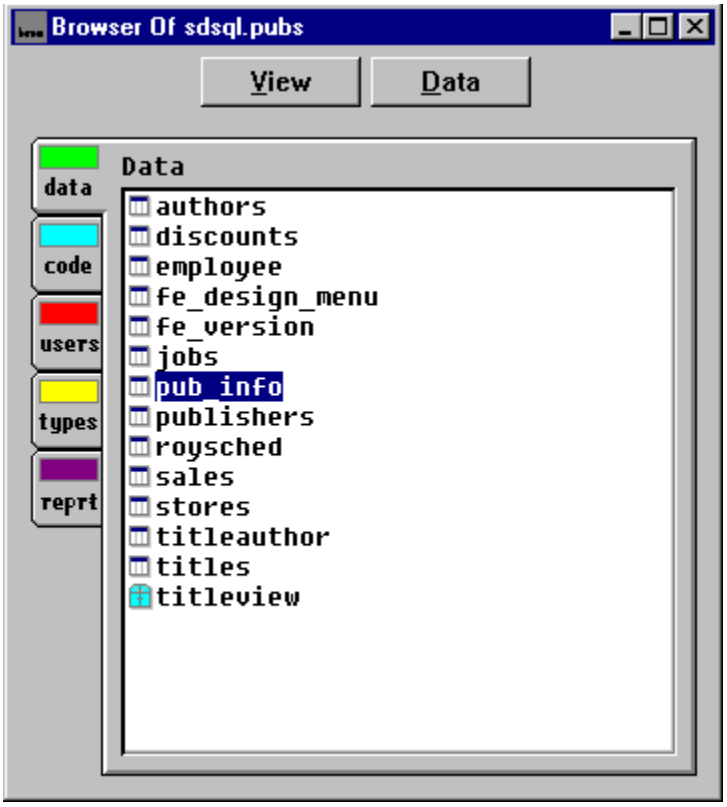
MDI Database Object Browser

This chapter describes the Browser Of *server_name.database_name* window and a method of quickly finding database objects.

Displaying the Browser Of *server_name.database_name* window

1. Click on the Browser () button in the main application toolbar

   A window entitled Browser Of *server_name.database_name* for the current database will be displayed.



The database Browser enables you to browse through database objects. To ensure clarity, the objects are divided into five groups represented by five tabs in the window.

- Data - tables, views
- Code - stored procedures, triggers, defaults, rules. By default, triggers are not displayed because they are usually not interesting to the user. Select Edit to display the triggers.
- Filter
- Users - users of the database
- Types - user data types
- Report – reports

0 Use the mouse or the keyboard shortcuts Ctrl+PgUp and Ctrl+PgDn to switch between the tabs.

1 Specific functions and buttons.
2 For each of the object groups, specific functions described in the respective chapters (see the above references) are available.

3 General Browser Functions

0 Search for an object in the object list

1. Press ALT and type the first letters of the name of the object you would like to find (hold Alt pressed).
0 The typed text is displayed in the information bar of the application.
1 2. Whenever you make a typing error, erase the typed text by pressing Space (Alt space, in fact).

0 SQLING automatically jumps to the object the name of which begins with the text you are typing.

0 Note: This function is not available for the Users group.

Refresh
0 Reload the window content by pressing F5.

1 Object owner
2 If the object was created by an other user than ”sa” the name of the user is shown after the object name as:
3 object_name (user_name)

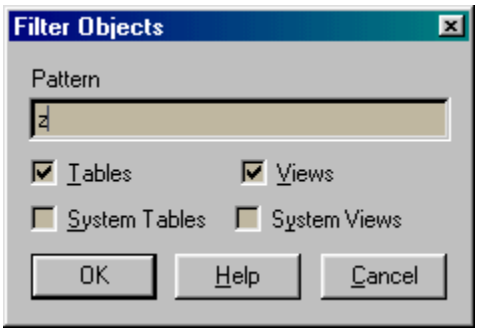
Edit Filter

Select Edit Filter to view only those tables you are interested in. The Edit Filter function also filters database objects in groups Code and Report.

0 Filter objects in the Data group.

1 1. Select Edit Filter from the browser Shortcut menu.

2 A dialog window entitled Filter Objects appears.

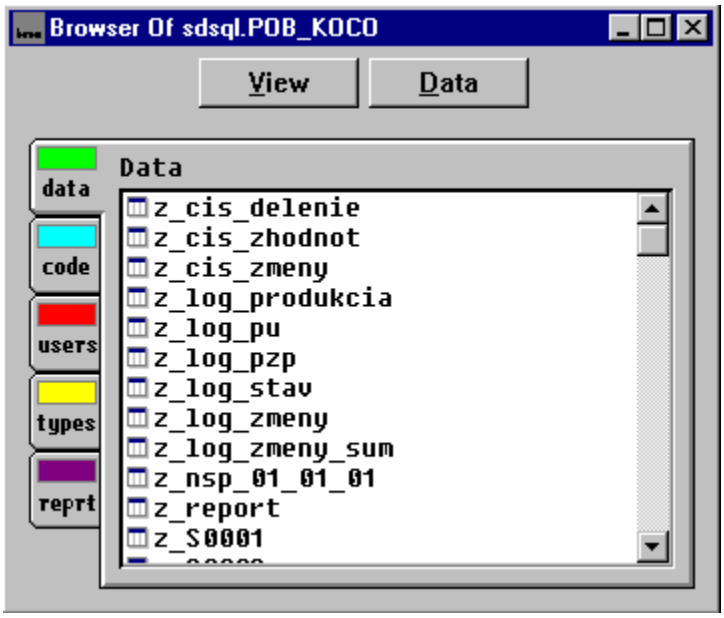


0 3. Enter the first letters (the prefix) of the objects which you wish to be displayed in the browser window in the Pattern field.

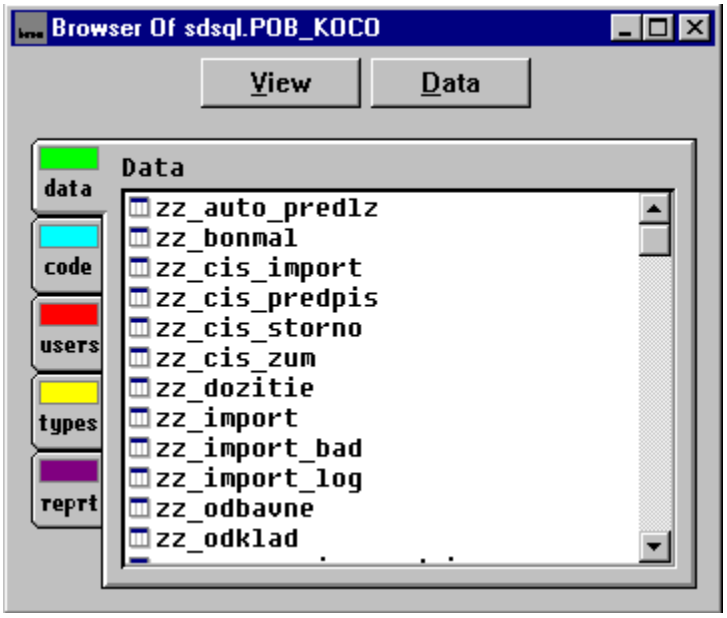
1 4. Check the check boxes of those objects you wish to display in the browser window.

2 5. Press OK.

3 Example.
4 If you type ”z” in the Pattern field, SQLING will only display tables beginning with the letter ”z” in the browser.



0 If you enter ”zz” in the Pattern field, only tables whose name begins with ”zz” are displayed in the SQLING browser window.



0 Disable the previously defined filter
1 The simplest method is to close and reopen the Browser for the current database.

Data - as a Browser object group.

This section describes the possible methods of browsing the structure of data objects - tables () and views () (only tables will be mentioned in the following as they are the most frequent data objects), creating and editing table notes. You will also learn how to edit and browse database table data comfortably and easily.

0 Available Window buttons differ according to the type of the browsed database.

- When browsing an actual database on an SQL server the following buttons are available in the window
  - View - displays the definition (structure) of the selected table, or the selected view.
  - Data - displays a window containing the data of the selected table.
- When browsing a working (downsized) database the following buttons are available in the window
  - New - create a new table (not view).
  - Design - edit table structure (not view). Views can only be edited and created in an actual database.

Shortcut menu offers additional functions for working with tables. The menu is different when browsing an actual and when browsing a working (downsized) database.

0 The Shortcut Menu for an actual database contains the following functions:

- Edit Filter… - set the subset of tables you wish to view in the browser.
- Drop Object… - delete the selected table.
- Edit Notes - edit the notes to the selected table.
- View All Notes - display the notes of all users appended to the selected table.
- Script Data… - save (the defined subset of) the selected table data into a file as an SQL script.
- Select COUNT(*) - diagnose the size of the selected table.
- Last Line WHERE… - display the last record in the table.
- WHERE Open… - display the table data which matches the specified WHERE condition.

The Shortcut menu of the browser of a working (downsized) database does not contain functions operating with table data (Save Data…, Load Data…, Select Count(*), Last Line WHERE and WHERE Open) because the tables in a working database never contain data. The menu contains the additional function

- Clone table - replicate the selected table.

Drop Object

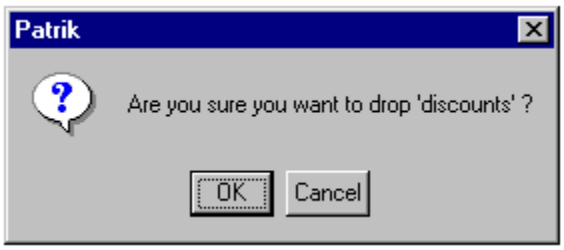
0 Use this command to delete the selected object.

1 Drop an object

1. Select the object you would like to delete from the database in the browser window. Click the right mouse button to bring up the Shortcut Menu.

0 2. Select Drop Object from the Shortcut menu.

1 The following dialog box appears.



0 3. Press OK to confirm you really wish to drop the object.

Edit Notes

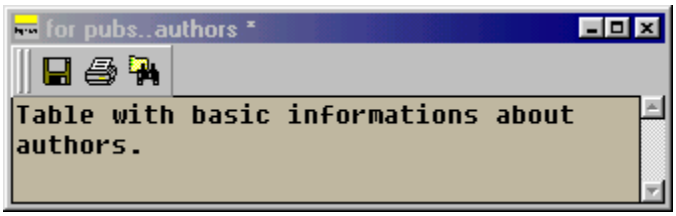
0 You can add your notes to any database object.

1 Adding notes to a database object

1. Select the object to which you would like to add a comment (or edit an existing note) in the browser window. Click the right mouse button to bring up the Shortcut Menu.

0 2. Select Edit Notes from the Shortcut menu.

1 A window named Edit Notes for *table_name* appears.



0 3. Type a new note or edit an existing one.

0 4. Click on the Save () button in the toolbar. The note will be saved immediately.

Display All Notes to an Object

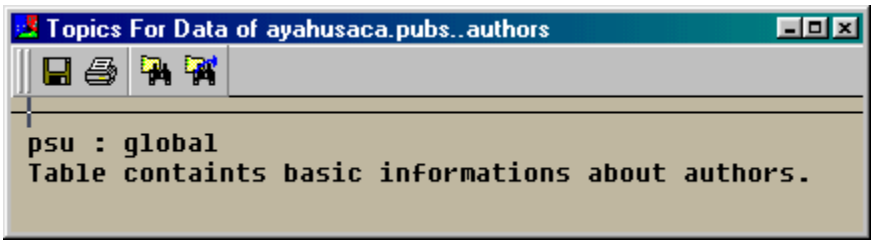
0 You can view all notes including the notes created by other users but you only can edit your own notes.

1 Displaying all notes

1. Select the object for which you would like to display notes in the browser window. Click the right mouse button to bring up the Shortcut Menu.

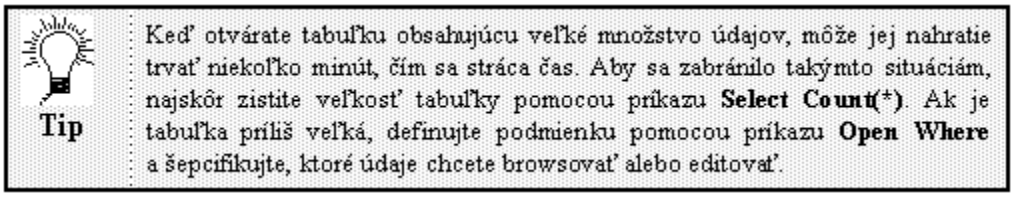
0 2. Select View All Notes from the Shortcut menu.

A window containing notes is displayed. The window is of the Results type.



The first line of each table note contains NT-login of user that created the note.

Select count(*)



0 Use the Select Count(*) function to determine the size of the selected table.

1 Determine table size

1. Select the table whose size you would like to know in the browser window.

0 2. Click on Select Count(*) in the Shortcut menu.
SQLING displays the number of rows and the size of the table in kilobytes in the main application window information bar.

```
Table fe_cinnost has 623 rows, used 746 KB
```

WHERE Open

0 Display the table data matching the user-specified WHERE Condition.

1. In the browser window, select the table for which you would like to view a subset of records. Click the right mouse button to bring up the Shortcut Menu.

0 2. Select WHERE Open… from the Shortcut menu.
A dialog box named Type Where appears.

0 3. Enter your WHERE Condition.

1 4. Press OK.

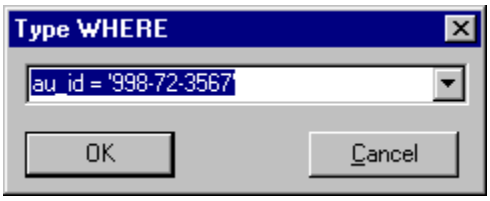
SQLING displays all records which satisfy the entered WHERE condition in a new window.

Last Line WHERE

0 Display the last table row - only works for tables with a primary key.

1. Select the table whose last row you would like to view in the browser window. Click the right mouse button to bring up the Shortcut Menu.

0 2. Select Last Line WHERE… from the Shortcut menu.
A dialog box entitled Type Where appears containing a WHERE condition which eliminates all table records except the one containing the highest values in the primary key columns. Usually MAX + 1 values are inserted into the primary key column when tables are filled, so the above condition will very likely return the last row in the table:



0 3. Press OK.

SQLING displays the last table record in a new window.

Clone Table

0 The Clone Table command is used to quickly create a table which is similar to or identical with an existing table.

1 Cloning a table

1. Highlight the table you would like to clone in the browser window. Click the right mouse button to bring up the Shortcut Menu.

0 2. Select Clone Table.

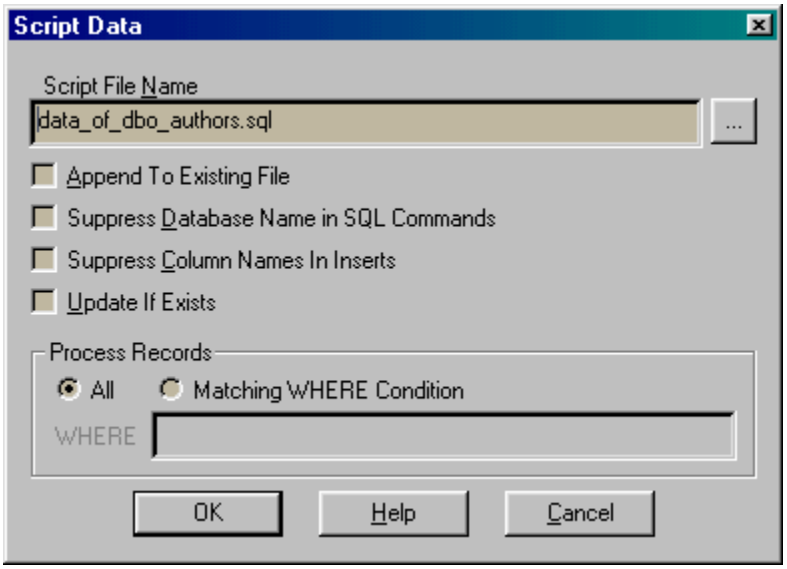
1 A dialog box entitled Clone To appears.

0 3. Enter the name of the new table and press OK.

SQLING opens the window named Design Table *new_table_name* in which you can edit the structure of the newly created table.

Script Data

The Script Data function is used to save the table data as a script consisting of INSERTs and other SQL commands for inserting large texts/binary data. Use the following dialog box



0 to set
1 1. Script File Name - the file where the script will be saved.
2 2. Append to Existing File - whether the script should be appended to an existing file.
3. Suppress Database Name in SQL Commands - whether any references to the table should contain the name of the database where the scripted table is stored.
4. Suppress Column Names In Insert - whether INSERT commands should contain column lists.

0 5. Update if Exists - whether any INSERT instructions in the script should be replaced by the following construction:

IF EXISTS (SELECT COUNT(*) FROM <table> WHERE <the where condition of the inserted row>.)
0 INSERT <table> VALUES (value1, value2, ..., valueN
1 ELSE
UPDATE <table> SET attr1=value1 WHERE <the where condition of the inserted row>
0 UPDATE <table> SET attr2=value2 WHERE <the where condition of the inserted row>
1 ...
2 UPDATE <table> SET attrN=valueN WHERE <the where condition of the inserted row>
END

0 6. Process Records - either All table data or only a subset of the data matching the WHERE Condition will be saved in the file.

Code Object Group

This section describes how the source code structure of coded objects (mainly procedures) can be created, edited and browsed. You will also learn how to create and edit comments to source-code objects in the window named Browser of *server_name.table_name* and how to create and retrieve help topics related to procedures.

0 Object type

Five object types are available in the browser tab named Code. Each type is represented by a different symbol in front of the object name.

- - stored procedure
- - trigger. Triggers are not displayed by default. Use the Edit filter function if you wish to view triggers.
- – default
- – rule
- - extended procedure

0 The following buttons are available in the Browser window.

1 

- New - creates a new object (e.g. a procedure). This button is only available in the window if browsing an actual database.
- Design - opens the procedure designer with the definition of the selected object.
- Execute - execute procedure.

The Shortcut Menu contains the following commands.

- Edit filter - filter objects in the browser window.
- Drop object - delete procedure.
- Edit notes - create notes to procedures.
- Plan size - determine procedure size.
- View help - view the first comment in the procedure body.

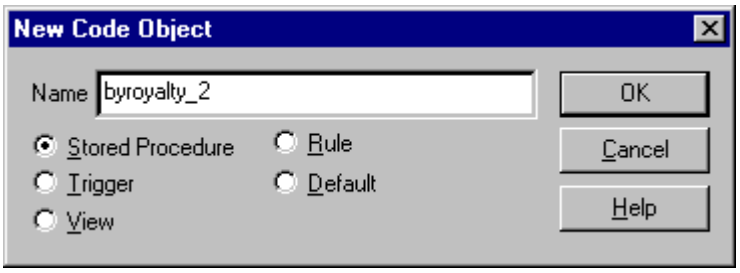
The New Button

0 Click on this button to create a new object in the Code object group.

1 Creating a new code object

0 1. Click on the New button in the Code object group in the browser window. The following dialog box is displayed

0



1 2. Enter the name of the new object in the field entitled Name, select the appropriate object type and press OK.

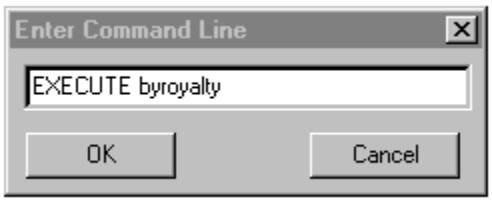
3. The Procedure designer window is opened and you can use it to program the code of the object.

The Execute Button

0 Execute a procedure.

1 1. Select the procedure to be executed in the browser.

2 2. Click on Execute.
A dialog box entitled Enter Command Line appears, with the setting: EXECUTE - procedure_name.



0 3. Enter procedure parameter values and press OK.

4. SQLING executes the procedure and displays the results in the Query Results window.

0 A new process is opened on the SQL server before the procedure is executed. The procedure runs in the new process.

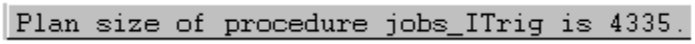
Plan Size

The Plan Size command is used to determine the size of the procedure after compilation. Use this function to maintain the procedure size under 64 kB and avoid incorrect functioning of the coded object.

1. Select the procedure the size of which you would like to know in the browser and click the right mouse button.

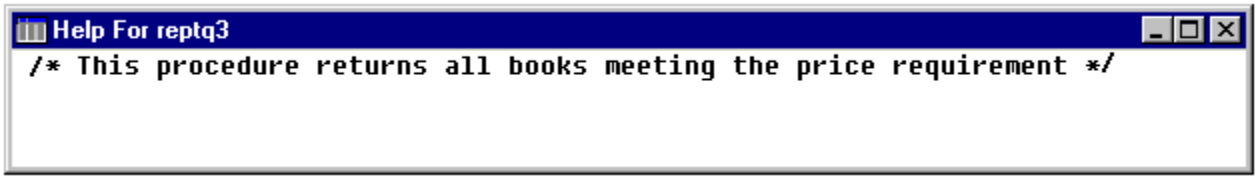
0 2. Select Plan Size from the Shortcut menu.

1 SQLING displays the size of the compiled procedure in bytes.



View Help

Select View Help to display the first comment in the source code of the selected procedure. Nothing is displayed if no comments are present in the source code. The first comment in the procedure body usually contains a description of its function.



The window entitled Help for *table_name* is of the Results type.

Group of Users

This section describes how users can be created, edited and browsed and how notes on users can be modified in the browser window.
Two types of items are located in the Users group window. They are distinguished by different symbols placed before their names.

0 Browser window buttons available for an actual database

- View - display the definition of the selected user.

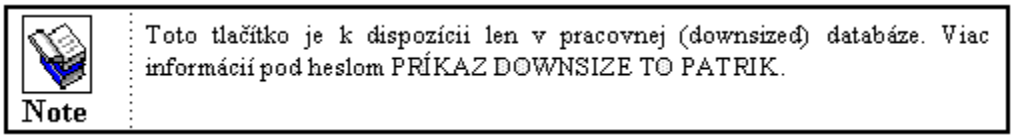
0 Browser window buttons available for a working (downsized) database

- New - create a new user.
- Design - modify the selected user. It is not possible to create or edit groups.

The Shortcut menu available in the browser window contains the following function:

- Drop object - delete a user.

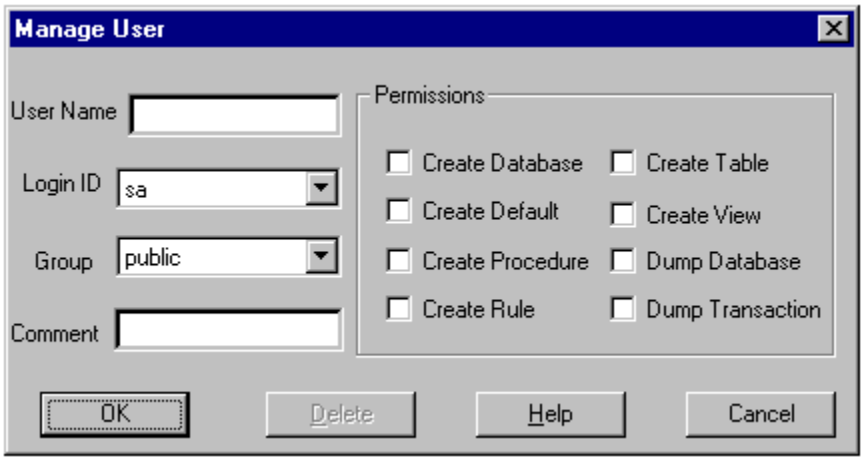
The New Button



0 Click on this button to add new users into the database.

1 Creating a new user

2 1. Click on the New button in the browser.



0 2. Enter the name of the new user in the field entitled User Name.

1 3. Select the Login ID in the combo box named Login ID. This will be the user's login on the server.

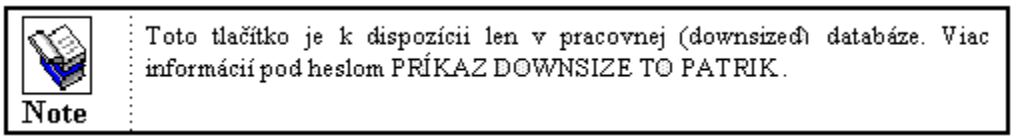
2 4. Select the group into which the new user will belong in the Group combo box.

3 5. Enter a comment attached to the user into the field named Comment.

4 6. Define the rights of the new user in the group named Permissions.

5 7. Press OK.

The Design Button



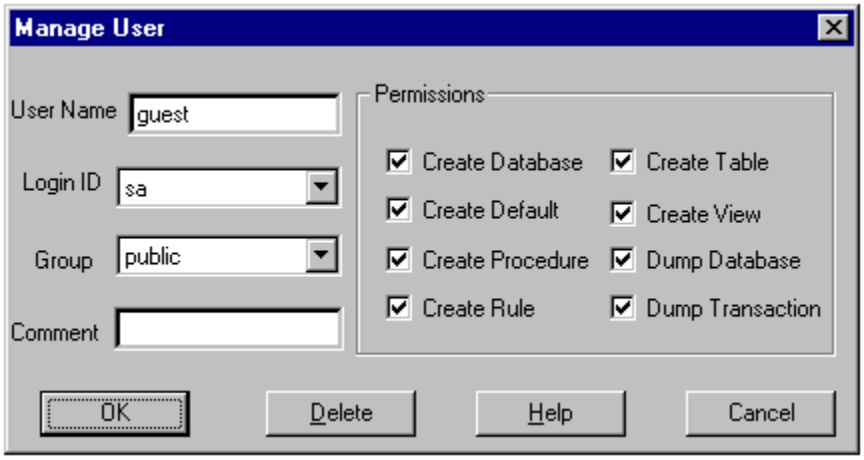
0 This button is used to edit user definitions.

1 Editing user definitions

2 1. Select the user whose definition you wish to edit in the browser window.

0 2. Press Design.

A dialog box entitled Manage User appears.

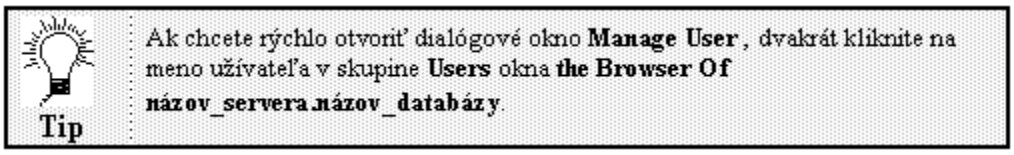


The dialog box consists of these fields:

- User Name
- Login ID - the user's login for the database
- Group - the group into which the user belongs
- Comment - user description
- Permissions - user rights

Press Delete to erase the user

0 3. Press OK to confirm any changes you have made.



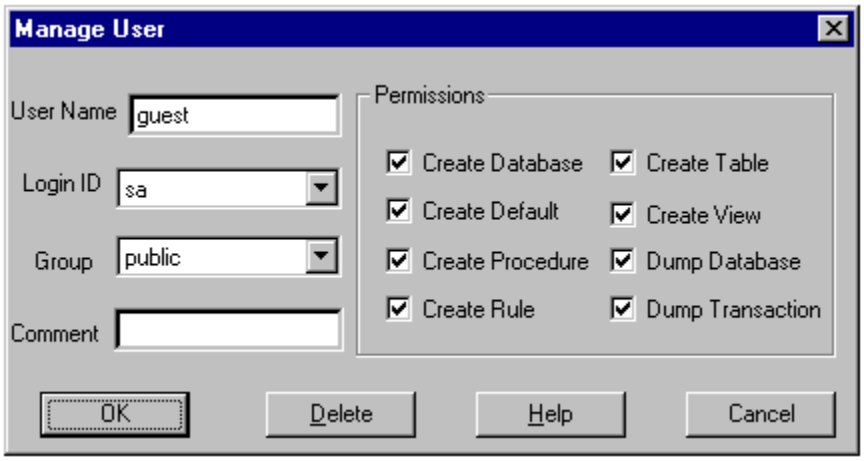
The View Button

0 View user definitions.

1 1. Select the user whose definition you wish to view in the browser window.

0 2. Press View.

A dialog box entitled Manage User appears.



The dialog box consists of these fields:

- User Name
- Login ID - the user's login for the database
- Group - the group into which the user belongs
- Comment - user description
- Permissions - user rights

0 Press Delete to erase the user.

User Data Types

This section describes how user data types can be created, edited and browsed, and how notes to user data types can be created and edited in the window named Browser of *server_name.table_name*.

0 Buttons available in the browser window for an actual database

- View - display the definition of the user data type.

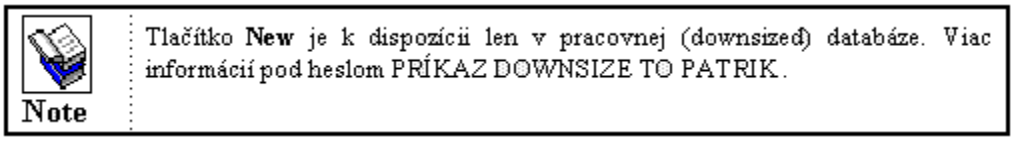
Buttons available in the browser window for a working (downsized) database

- New - create a new data type.
- Design - modify the selected user data type.

The Shortcut menu available in the browser window contains this function:

- Drop object - delete user data type.

The New Button

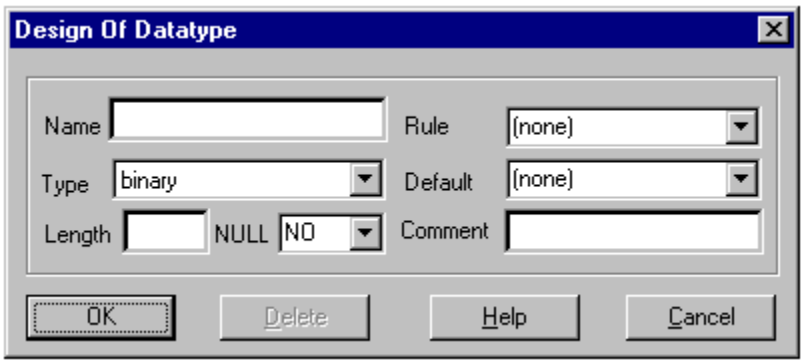


0 Use this button to define new user data types in the group named Types.

1 Define new user data types

0 1. Press New in the window of the Types group in the window named Browser of *server_name.database_name*.

1 A dialog box entitled Design of Datatype appears.



0 2. Enter the new data type name in the field entitled Name.

1 3. From the Type combo box select a system-defined data type on which the new data type will be based.

2 4. Enter the length of the new data type in the field named Length.

3 5. Set whether the data type may contain NULL values in the Null combo box.

4 6. Select the validation rule in the Rule combo box.

5 7. Select the DEFAULT for generating a standard value for the new data type in the combo box named Default.

6 8. Write a comment appended to the new data type in the field named Comment.

7 9. Press OK.

The Design Button

0 This button is used to modify an existing data type.

1 Edit the definition of a user data type

1. Select the data type whose definition you would like to edit in the Types group in the window entitled Browser of *server_name.database_name*.

0 2. Press Design.
A dialog box entitled Design of Datatype appears.

The dialog box contains the following fields

- Name - name of the data type
- Type - system data type on which the selected data type is based
- Length - size of the new data type
- Null - permit or restrict NULL values for the data type
- Rule - validation rule for values of this data type
- Default - generating a standard value for the data type

Press Delete to discard the data type.

0 3. Press OK to confirm the modifications you have made.

The View Button

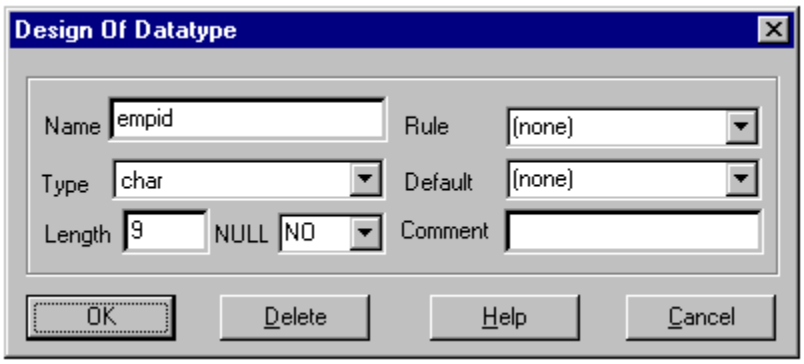
0 This button can only be used to view data types.

1 Viewing the definitions of data types

1. Select the data type the definition of which you would like to view in the Types group in the window entitled Browser of *server_name.database_name*.

0 2. Press View.

A dialog box entitled Design of Datatype appears.



0 The dialog box contains the following fields.

- Name - name of the data type
- Type - system data type on which the selected data type is based
- Length - size of the new data type
- Null - permit or restrict NULL values for the data type.
- Rule - validation rule for values of this data type
- Default - generating a standard value for the data type

Press Delete to discard the data type.

Reports

This chapter deals with reports. Reports are a method of viewing data and may contain graphs representing the data and the related statistical indicators.

The group named Report contains the names of all reports existing in a database. Unlike other groups (data, code, etc.) reports are not standard database objects and therefore they cannot be ”seen” from other applications. Reports are only used for presenting data, not for editing it. Permission to read cannot be set explicitly but is inherited from the tables which participate in the report.

Browser buttons for the Report group

Reports and downsized databases

There is no sense in making a connection between reports and downsized databases, because reports are used to display actual table data. All functions related to reports are only available on an actual database containing data. The following buttons are visible in the browser:

- New
- Design
- Execute

Click on the buttons New and Design to open the report designer in which you can develop a New report or modify/view an existing report (Design). The Designer offers the best possible image of the report structure. While these two buttons will mostly be used by somebody designing reports, the third function, Execute, is mainly intended for the end user.

0 Click on Execute to view the report data without any additional report structure information.

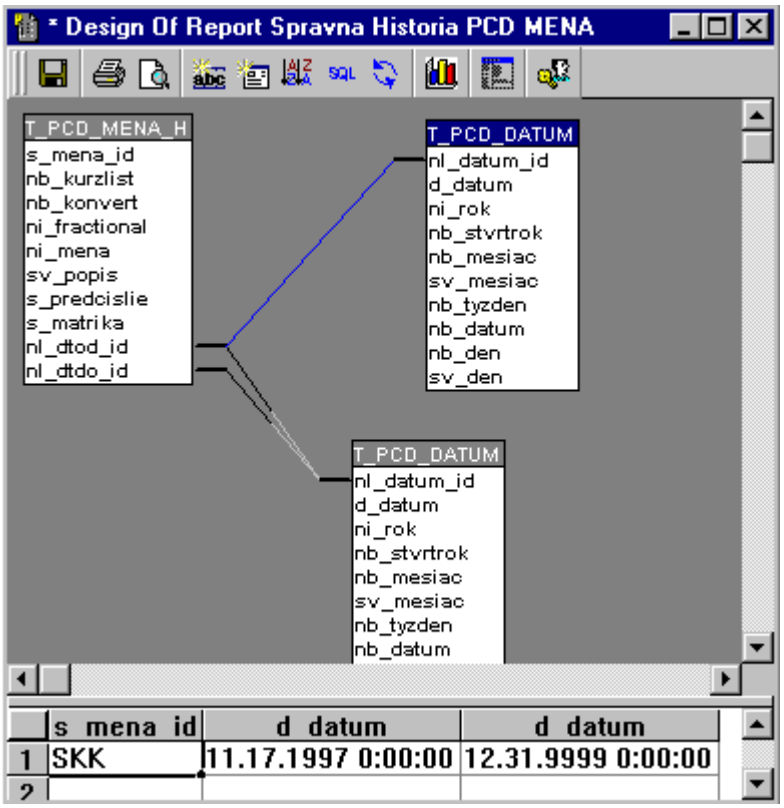
0 The Report group Shortcut Menu in the Browser.

1 When the right mouse button is clicked over a report, a menu containing the following functions appears

- Edit filter - filter objects in the browser window.
- Drop object - delete a report.
- Edit notes - create a note to a report.

Report Designer

Reports are essentially ordinary SELECT instructions operating on several chosen database tables. It is neither necessary (nor possible) to write such SELECT commands manually. If you are creating a new report the Add Table dialog box appears and you can choose the table (tables) which will be used in the report. Reports need not be based exclusively on table data. You can use reporting to visualize the data of Views and other reports. This is enabled by structured report definition. In similar cases the report SELECT command is more complex and aliased Sub-SELECTS are included in place of tables after the FROM clause.



0

The designer window is horizontally divided in two parts. Report tables are shown in the upper part and report columns in the lower.

0 Modification Flag

1 The window title contains the text Design of Report <report name>. An asterisk * precedes this text if the report has been modified.

2 Autorefresh

If no autorefresh is set for the report in its Properties, the sample of data used in the report is not reloaded after modifications. Click on Refresh ( ) on the toolbar to reload the sample data into the changed report.

0 Toolbar

1 The toolbar consists of the following functions

-  Save – stores the report in the database.
-  Print – prints the report.
-  Preview – sets report printing properties.
-  Add Column – adds a new column to the table and opens the dialog box entitled Column Options.
-  Add Table – adds a table to the report.
-  Order By – defines the ordering of the report data.
-  Show SQL – displays the SELECT instruction which retrieves the data used in the report.
-  Refresh - reloads the report data sample.
-  Open New Graph – opens a new graph. If any report columns are highlighted when this function is called, the graph will only show the highlighted columns. If no columns are highlighted then the graph shows all columns of the report. Two types of columns are distinguished - numeric and text columns. Numeric columns represent vectors with data and text columns represent node titles. If more text columns have to be displayed in a graph, their values (texts) are collated into a single body of text (divided by spaces). The graph will not be displayed unless the columns to be displayed include at least one numeric and one non-numeric column.
-  Auto Read Comments – loads the comments to all columns of all tables included in the relational model. Every table added to the report is automatically loaded along with its comments.
-  Properties.

0 Saving Reports

All reports are stored in the SQLing database on the SQL server. This database contains all optional settings of the reports including graphs and the preview information. Any report may have an unlimited number of graphs. Every graph has its title which may contain spaces and any other characters to make it as expressive as possible. The same applies to the name of the report.

0 Locking reports

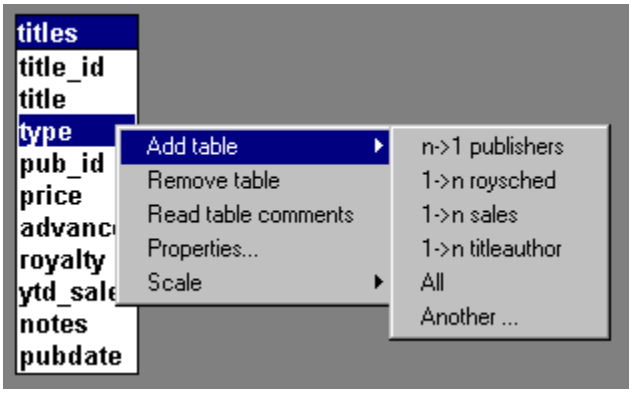
SQLing automatically locks the report while it is edited to ensure safety in case more users would like to edit a single report simultaneously.

Report Tables – Relational Model.

The upper part of the designer window contains a relational model of the tables used in the report. If the report consists of more tables and is correctly built all tables should be interconnected by relations, whether actual (displayed in black color) or created for the purposes of the report only (colored blue). Both types of relations act as real in the report and are only distinguished by color so that the user does not become confused by the temporary relations.

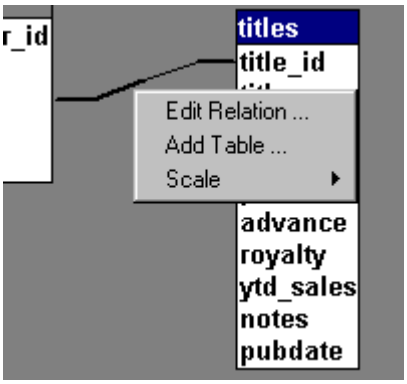
0 The Shortcut Menu in the Relational Model of the Report Designer.

Click the right mouse button to bring up the shortcut menu, as usual. The shortcut menu changes according to context - according to the location of the mouse pointer.



0 If the mouse pointer is located in the area of a table, the shortcut menu contains the following set of functions:

- Add Table – opens a submenu containing the names of all tables which are connected to the current table by a relation and the commands All and Another…. Select any of the table names to add the table into the relational model. Select All to add all displayed tables into the relational model. If you wish to add a table (tables) not listed in the menu (not connected by a relation to the current table) select Another….
- Remove Table – select this function to remove a table from the relational model of the report. This will also remove that table from the report. If any of the table columns is also a column in the report, SQLing displays a warning and prompts you to confirm your intention. If you do not cancel the process, the columns of the deleted table will be removed from the report.
- Read Table Comments | Hide Table Comments - only one of these reverse functions is available at a time. Read Table Comments loads comments to the columns of the table. The comments are created in the Design Table window. Displaying comments is not always convenient because the comments may take up most of the space in the relational model. Usually the comments are loaded only when the report is composed and are then disabled to save space.
- Properties – this function fine-tunes and optimizes the SELECT.
- Scale.

0 

If the mouse pointer is not located over any table but a relation is activated (make a relation active by clicking on it with the left mouse button) the following menu is displayed

- Delete Relation – only if an unreal relation is active (colored blue). Actual relations can only be deleted in the Relations Module.
- Edit Relation – view/change the relation properties connected with the created SELECT.
- Add Table…
- Scale

0 In any other case the menu contains the following functions:

- Add Table…
- Scale

Moving tables

You can move the tables in the relational model as you wish, this does not affect the functioning of the report in any way.

Creating new (unreal) relations

It is sometimes necessary to connect the columns of two tables by a relation without any referential integrity or persistence of such relation outside the current report. The Designer supports such fictitious relations and displays them using blue color. Follow this procedure to create a new false relation:

1. Select the columns to be connected by a relation in one of the tables.
2. Press and hold the left mouse button and move the mouse pointer into the area of the other table.
3. Release the left mouse button.
1. A dialog box entitled Edit Relation appears. In this dialog box, specify the pairs of columns to be connected and the type of the connection (join).

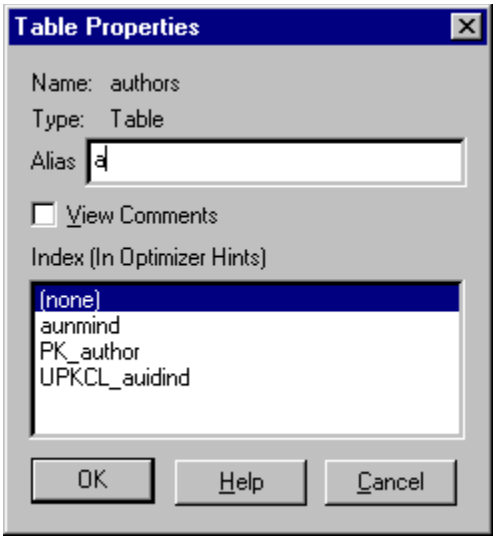
Adding columns into the report

The columns included in the report are listed in the lower part of the report designer

window. Add a column either by double clicking (the left mouse button) on the desired table column in the relational model or dragging and dropping one or more columns into the report column list.

Table Properties

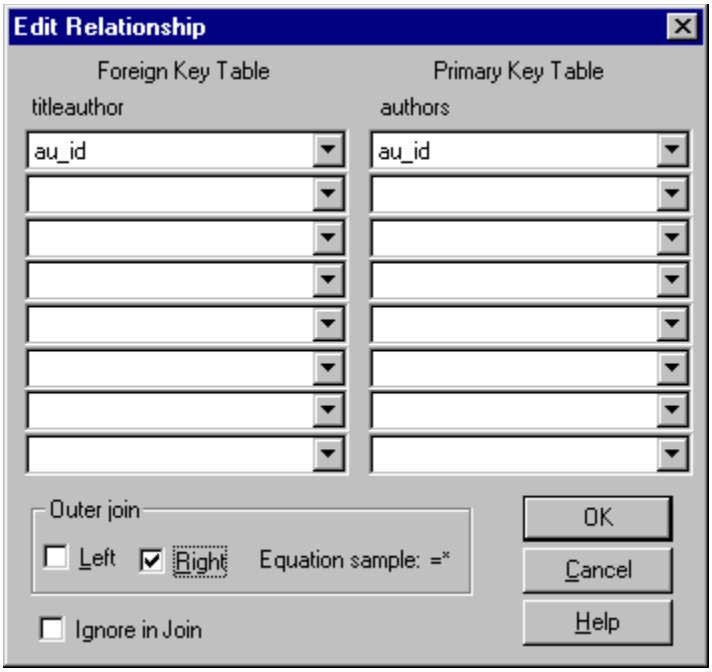
This function is retrieved by clicking the right mouse button on any of the tables displayed in the relational model of the report designer and selecting the Properties item.
The following dialog window appears



The name of the current table is displayed in the field entitled Name.
Type of object is shown in the field named Type. The object type is most likely a Table but it also can be a View or another Report.
In the editable Alias field, enter the alias name for quicker table identification in any SELECT. Aliases are used when the user wants to build a SELECT in the designer and then use it in a Procedure or a Query. If several instances of the same table exist in one report (relational model of the designer) it is necessary to use aliases because the SELECT instruction generated by SQLing would otherwise be returned by the SQL server as incorrect.
The View Comments check box shows whether comments to table columns are loaded. You can toggle this check box.
The list box named Index (In Optimizer Hints) is used to optimize the SELECT instruction returning the report data. For any table you can specify an index which will be suggested to the optimizer by the SELECT as default. The list box includes the row (none) and the names of indexes in the current table. Select (none) if you wish to leave the selection of indexes to the SQL optimizer (which should be able to make the best decision).
This function has been included in SQLing because the SQL Server fails to select the optimum index from time to time.

Setting Relation Properties.

This function is invoked when a new relation is added, when an existing relation is double clicked, when
Edit Relation is selected in the shortcut menu while the mouse pointer is over an active relation or when a relation is added. The following dialog box appears:



It lists the names of columns in both connected tables. You can only change this connection if it is a false relation (colored blue), or a newly added relation. Real relations cannot be modified in this dialog box but you can do so in the Edit Relation window.
The section entitled Outer Join displays the type of connection (join). Such connections can be constructed for any relation type as they do not depend on the declaration of the relation and only have relevance to building the JOIN in a SELECT instruction.
Check Ignore in Join to prevent the relation from being used when creating a SELECT instruction. In such case another (auxiliary) relation is usually built so that the report does not show the Cartesian product of tables.

Add Table

This function is automatically invoked whenever a new report is created, but it can also be called later, by clicking on this button on the Report Designer toolbar.

The following dialog box is displayed

It contains a list of all tables, views and reports (including those that are no longer shown in the relational model of the designer). Objects (tables) are added into the relational model either by double clicking on their names or by highlighting the object and clicking on the Add Table button. When you attempt to add the same table into a report for a second time its relations are not loaded again. SELECTS are only generated correctly for such tables if you set an alias in the Properties.

Scale

The Scale Shortcut menu contains these items

- 100 %
- 75 %
- 50 %
- 25 %

0 Select the scale at which you wish to display the relational model.

Report Columns

0 The report table containing a data sample is displayed in the lower part of the report designer.

Table Properties

0 Tables support most of the standard table properties, including

- column/row/rectangular cell area selection
- clipboard functions
- widen/shrink rows/columns
- sort columns by double clicking on the first table row.

Changing column order.

Drag and drop columns within the report to change their positions:

1. Highlight the column (or several adjoining columns) which you wish to move and press the left mouse button.
2. Move the mouse pointer to the column before which you wish to insert the moved column.
3. Release the left mouse button.

Removing columns from the report

1. Highlight the column (columns) which you wish to remove.
2. Press Delete.

Shortcut Menu

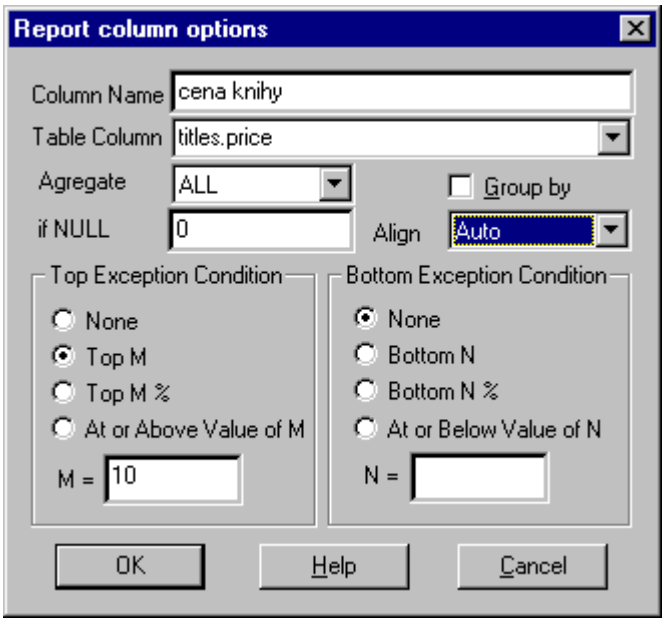
Press the right mouse button to retrieve a menu containing the following functions

- Open Graph – opens a submenu containing the names of graphs which were created for the report and a command, <New>. Select any of the existing graphs to view it. This will not work if the columns of the report visualized in the graphs were removed. In such case SQLing shows a warning and enables you to delete the invalid graph. Select <New> to display a new graph. If any report columns are highlighted when this function is called, the graph will only show the highlighted columns. If no columns are highlighted then the graph shows all columns of the report. Two types of columns are distinguished - numeric and text columns. Numeric columns represent vectors with data and text columns represent node titles. If more text columns have to be displayed in a graph, their values (texts) are collated into a single body of text (divided by spaces). The graph will not be displayed unless the columns to be displayed include at least one numeric and one non-numeric column.
- Column Options – sets report column options. The menu only contains this item if any report column was highlighted when the shortcut menu was invoked.
- Delete Column – removes a column from the report. The menu only contains this item if any report column was highlighted when the shortcut menu was invoked.
- Add Column – adds a new column to the table and opens a dialog box entitled Column Options.
- Refresh – reloads report data sample.
- Show Cursor Script - generates and displays a script browsing SELECT records of

the report according to a cursor. First asks for the name of the cursor.

Column Options

Use this function to set detailed properties of a column of the designed report in the following dialog box:



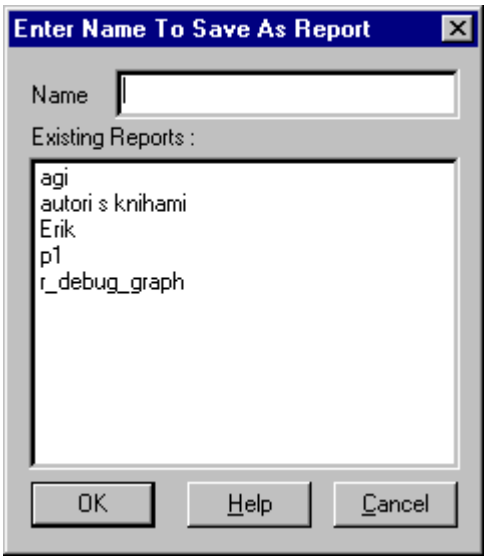
0

- Column Name - displays text with the name of the report column.
- Table Column - the column and table displayed by the selected report column. Names of columns of all tables included in the report are listed in the combo box named table column. If you do not select anything from this list, enter your own text, e.g. a function (or an asterisk, *). This text will be handed to a SELECT - the SELECT list of the report.
- Aggregate - this combo box contains the names of all aggregate functions of Transact-SQL and the pseudo-function ALL which ensures that all table column data is displayed and no aggregate function is used. If you decide to use an aggregate function for any of the columns, the created SELECT will automatically be grouped by other report columns. This is the only possible way of ensuring the correctness of such SELECT.
- Group By - means that Grouping by the selected column is used in the report SELECT.
- If NULL - the value to replace any NULLs in the column. The value must be of the same data type as the column.
- Align - defines how column values are aligned (left, right or centered). Select Auto to align values automatically (align text left and numbers right).
- Top Exception Condition - this function highlights the greatest values in the column. Toggle highlight (red)
    - Top M - M-count of the greatest values in the column.
    - Top M % - top M-per cent of the column values.
    - At or Above Value of M - column values greater than or equal to M. This

option is only available for integer columns because M is an integer.
    - None - if you do not wish to highlight any great values.
    - M = - an integer value.
- Bottom Exception Condition - this function highlights the lowest values in the column. Toggle highlight (green)
    - Bottom N - N-count of the lowest values in the column.
    - Bottom N % - Bottom N-per cent of the column values.
    - At or below Value of N - column values smaller than or equal to N. This option is only available for integer columns because N is an integer.
    - None - if you do not wish to highlight any small values.
    - N = - an integer value.

Storing Report in the Database

If the report was not just created it is stored in the database under its original name. If the report is newly created the following dialog box opens



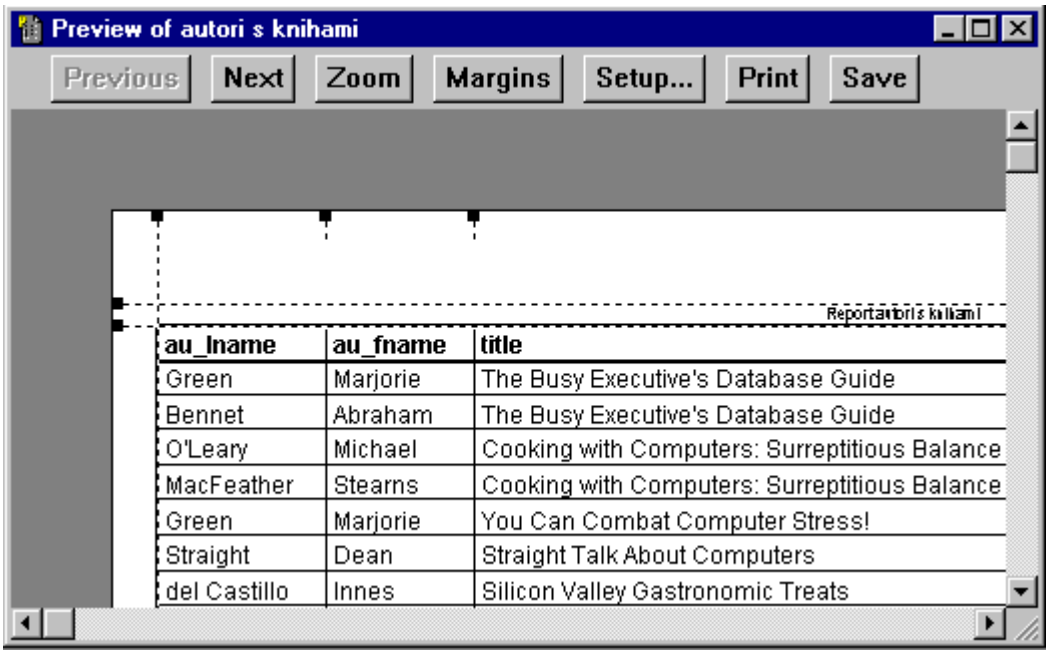
0

1 containing these fields:

- Name - enter the name of the saved report.
- Existing Reports - lists all reports existing in the database. You can also specify one of these names as the name of the saved report. The report with the selected name is replaced by the saved report. SQLing prompts whether you really wish to overwrite the report.

Preview

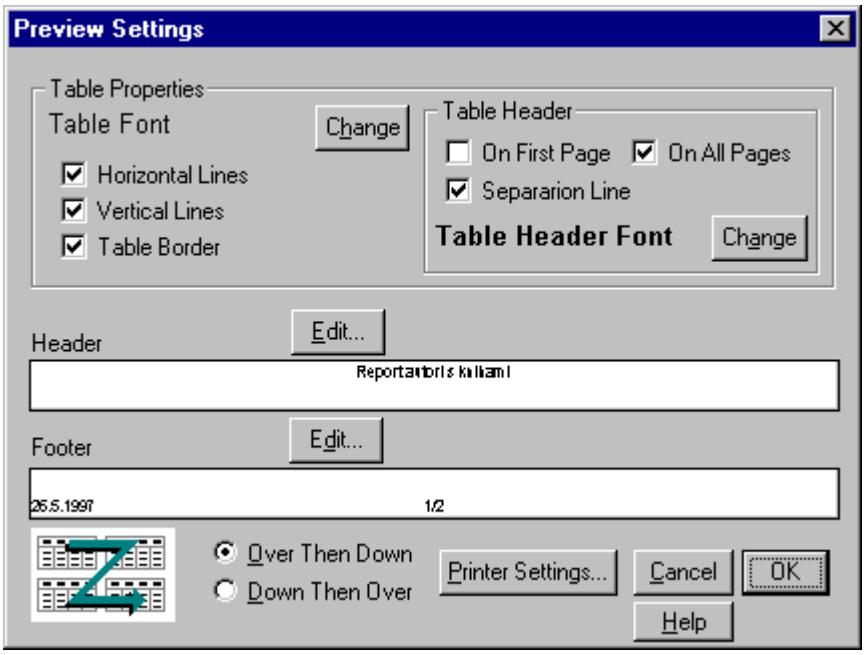
0 Enables you to determine the properties of a printed report.

1 

- Previous, Next - click on these buttons to change the page displayed in Preview.
- Zoom - use this button to switch between a minimized view - the whole page is visible in the window, and an enlarged view - the Preview window only displays a scrollable part of the page.
- Margins - displays dotted lines which enable you to set the margins and all report columns. Click on the margins button again to disable the dotted lines.
- Setup - enables you to set other details (fonts, header, footer) of the printed report.
- Print - prints the report.
- Save - saves the settings which were set in the preview mode. The button is only available if any modifications were made in the preview mode. You can only save settings for a report which is already stored in the database.

Preview Setup

0 Opens the following dialog box where you can set the properties of the printed report

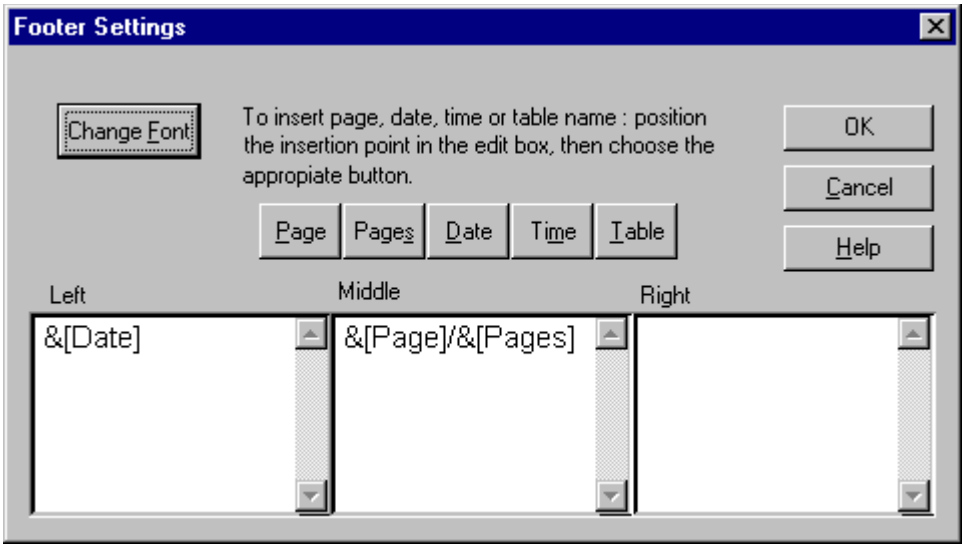


1

2 using the following fields

- Table Font - Change - opens a dialog box in which you can set the font to be used for table cells. The active font is applied on the sample text Table Font.
- Horizontal Lines - set whether horizontal lines separating table rows will be printed.
- Vertical Lines - set whether vertical lines separating table columns will be printed.
- Table Border - set whether table border (thicker line) will be printed.
- Table Header - table header settings (header contains column names)
  - On First Page - the table header will only be located on the top page of the report.
  - On All Pages - the table header will be located on every report page. If neither of the check boxes is selected the table header is not printed on any pages of the report.
  - Separation Line - set whether the header (if it is printed) should be separated from table cells by a thicker line.
  - Table Header Font - Change - opens a dialog box where you can set the font to be used for the table header. The currently active font is used on the sample text Table Header Font.
- Header - Edit - enables you to set page headers in the Header Settings dialog box.
- Footer - Edit - enables you to set page footers in the Footer Settings dialog box. Blank fields below these functions are used to preview the header and the footer.
- Over then Down, Down then over - set the order in which the tables are to be printed in case the report is wider than one page.
- Printer Settings - opens the printer driver dialog box.

Header/Footer Settings



0

In this dialog window you can set the header/footer of the printed report pages. All printed pages have the same definition of header/footer (only headers further on). The definition of a header consists of three parts Left - part aligned left, Middle - centered part, Right - part aligned right. Each of these parts has its own text editor where you can enter the text to be displayed in that section of the header. If you wish the text to contain a substituted value, place the cursor on the desired location and click on one of the following buttons:

- Page - insert the page number
- Pages - insert the total number of all printed pages
- Date - insert the date when the report is printed
- Time - insert the time when the report is printed
- Table - insert the name of the printed report (table)

The text may also include identifiers beginning with a tilde ”~”. If a variable of such name is used in the report the identifier is replaced with the value of the variable. Report variables (parameters) are specified in Properties.

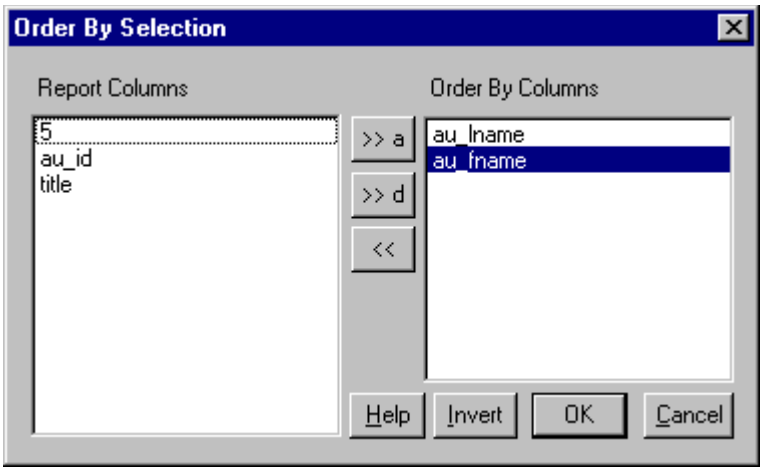
Click on Change Font to open a dialog box in which you can set the font of the section over which the text cursor is located.

Print

Prints the report. The appearance of the printed report can be set and/or viewed prior to printing in the Preview window.

Order By

In the following dialog box

0 

1 you can set the ordering of report data.

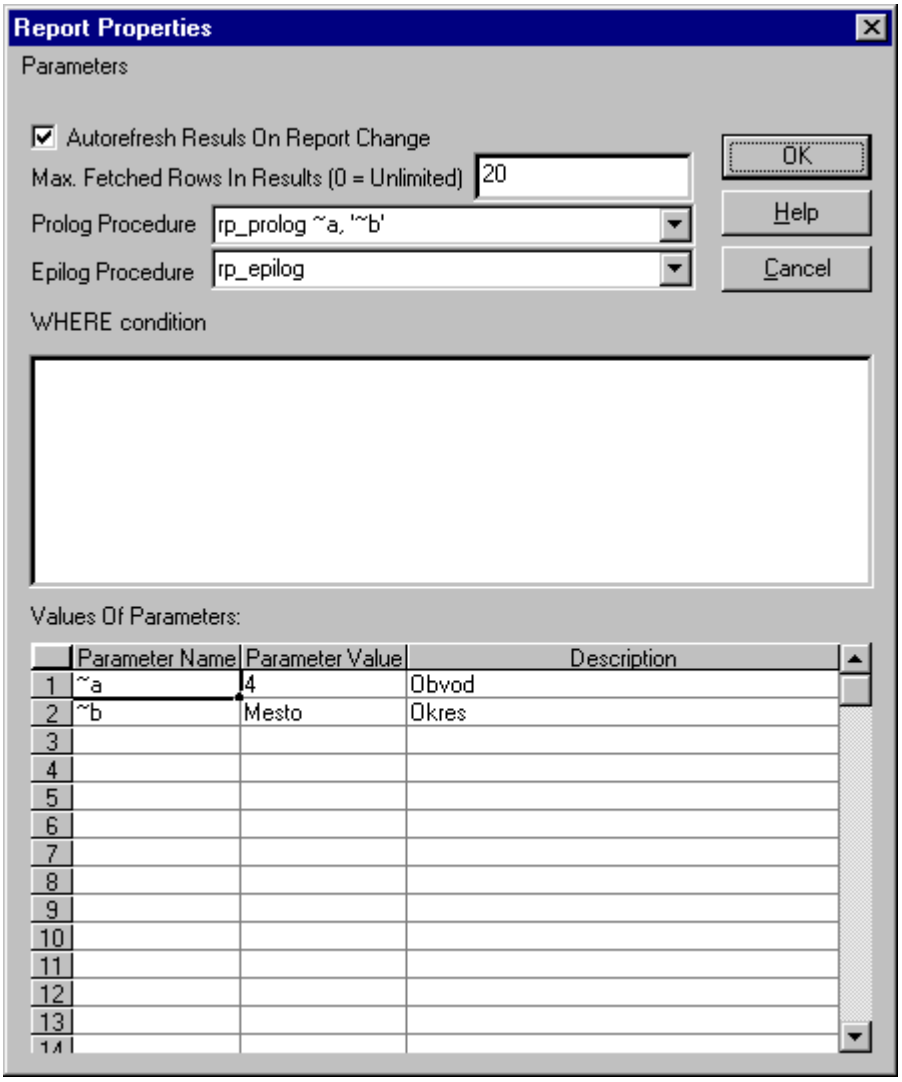
- The list box entitled Report Columns contains the names of all report columns (alphabetically ordered) which are not used in the ordering. The list box may contain numbers. The numbers determine the report column order and are calculated, not retrieved from tables.
- The list box entitled Order by Columns includes the names of columns which are used in the ordering. The columns are ordered by their priority in the ordering. If a column name is followed by "- Desc", then the ordering by that column is descending (otherwise it is ascending).
- The button >> a adds the columns highlighted in the Report Columns list box at the end of the list of columns in the Order by Columns list box. Ordering by any newly added columns is ascending.
- The button >> d adds the columns highlighted in the Report Columns list box at the end of the list of columns in the Order by Columns list box. Ordering by any newly added columns is descending.
- The button << moves the columns highlighted in Order by Columns back into Report Columns.
- The button named Invert swaps ascending and descending ordering in columns highlighted in Order by Columns.

Show SQL

Opens a dialog box with a read-only editor containing the SELECT instruction which represents the report. The text can be highlighted, copied into clipboard (Ctrl C) and pasted into the Query Editor.

Properties

0 Select this function to configure the WHERE Conditions and report procedures, and to set the designer options.



1

- Autorefresh Results On Report Change - a new data sample is automatically loaded from the SQL server every time the report changes. Use this check box to enable/disable this function. When Autorefresh is disabled, reload the data sample manually, by selecting Refresh in the shortcut menu when between report columns or by clicking on the Refresh button on the toolbar.
- Max Fetched Rows in Results - limit the number of report data sample rows in this field. Zero (0) means the report sample row count will be unlimited.

0 Report Procedures

1 SQLing enables you to define two types of procedures:

- Prolog procedure - the procedure which is automatically executed before

    - report data is loaded, i.e. before the SELECT instruction is launched during report Execute
    - the whole report definition is loaded when opening the Report Designer.
- Epilog procedure - the procedure which is automatically executed after
    - the data is loaded during report Execute
    - the Designer is closed.

If the report is so complex that it cannot be constructed using a single SELECT, you can create a procedure which inserts the rows that should be the result of the report into a (purpose-created) table, displayed later by the report. In such case it is advisable to create a paired Epilog procedure which removes the data (or drops the whole table) created by the first procedure.
You may specify parameters for the procedures. The parameters can be constants or variables, as described in the section entitled Values of Parameters.

The Where Condition

The Where Condition of the report's SELECT consists of two parts.

1. The parts containing the joins of the relationally connected tables. This part is automatically generated in the SELECT by the designer.
1. User defined part of the WHERE Condition. This part can be edited in the WHERE Condition field of the dialog box.

Both parts are connected by AND.

Values of parameters

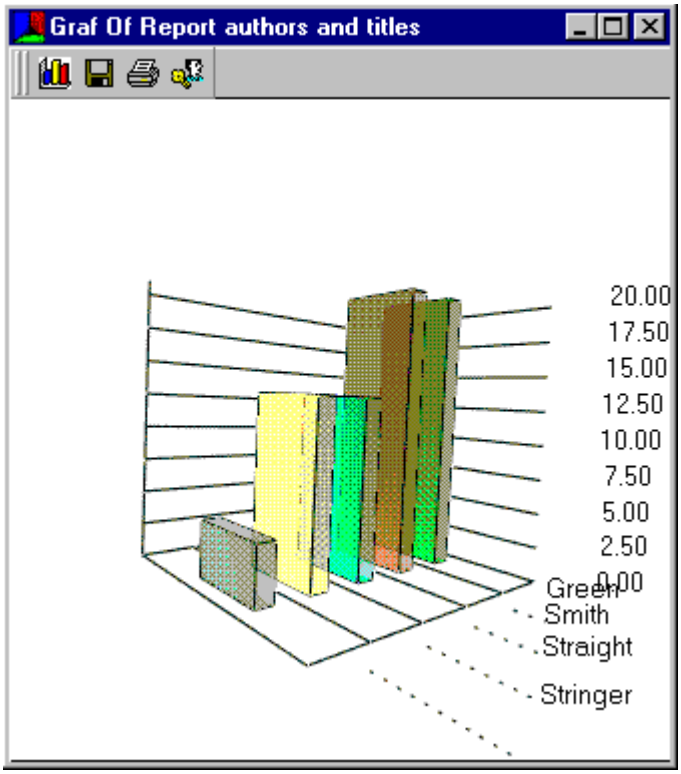
The user-defined part of the WHERE Condition, as well as the parameters of the prolog and epilog procedures, may contain identifiers beginning with a tilde "~". Such identifiers are translated by the designer as parameters. Assign values to parameters in the table named Values of Parameters by entering the name of the variable in the column entitled Parameter Name, variable value in the column entitled Parameter Value and a description of the variable in the column entitled Description.

The Parameters menu of the dialog box

parameter values are stored in the SQLing..param table. (The names and descriptions of parameters are stored with the report). Each report may have an unlimited number of parameter sets. Every parameter set may contain values for several parameters and has its unique name.

- Load - loads the set of parameters from the param table
- Save as - saves the set of parameters into the param table. Displays a dialog box which lists the names of all parameter sets in the current report and enables you to enter a new name for the saved set or overwrite an existing set.
- Save as Default - saves the parameter values in the param table as Default. This name has a special position among all other parameter set names:
    - when a report is loaded into the designer the Default set is automatically opened
    - when opening a report using the Execute function SQLing prompts for parameters (if the report includes any) and offers the values from the Default set (if any such set exists for the report).

Graph

0 

Graphs are used as graphical representation of the report data. At least one numeric column of the report is transformed into graph nodes (if one column is used, the graph will be two-dimensional, if more are used, three-dimensional). At least one text column of the report is displayed in the graph as node description.

0 Shortcut Menu

1 Press the right mouse button on the area of a graph to display a shortcut menu containing the following functions:

- Graph Type - change the graph type
- Options
- Rotation - opens a dialog window where you can change the position of a 3D graph. This function is only available if the graph is of the Bar 3D or the Pie 3D type.
- Size of Graph - change the graph size. The size is independent form the graph window size.
- Save Graph ”<graph name>” - the function saves any modifications made to the graph into the database.
- Save Graph As
- Print Graph … - opens the Printer Setup dialog box where you can set printer options and print the graph.

0 The Toolbar contains the following functions: (all of which are also available in the Shortcut Menu)

- Graph Type - change the graph type.
- Save Graph - the function saves any modifications made to the graph into the database. If the graph has not been saved before the function calls Save Graph as.
- Print Graph … - opens the Printer Setup dialog box where you can set printer options and print the graph.
- Options.

Set graph colors by opening Colors in Options.

Save graph as

0 Opens the following dialog box

1 

2 containing the fields:

- Name - enter the name of the saved graph here.
- Existing Graphs - lists the names of all graphs existing in the report. You can select one of these names when saving the current graph. The selected graph will then be replaced by the saved graph and SQLing will prompt whether you really wish to overwrite the existing graph.

0 If you enter a new name the saved graph will be added to the report.
1 You can only save graphs if the report has been saved.

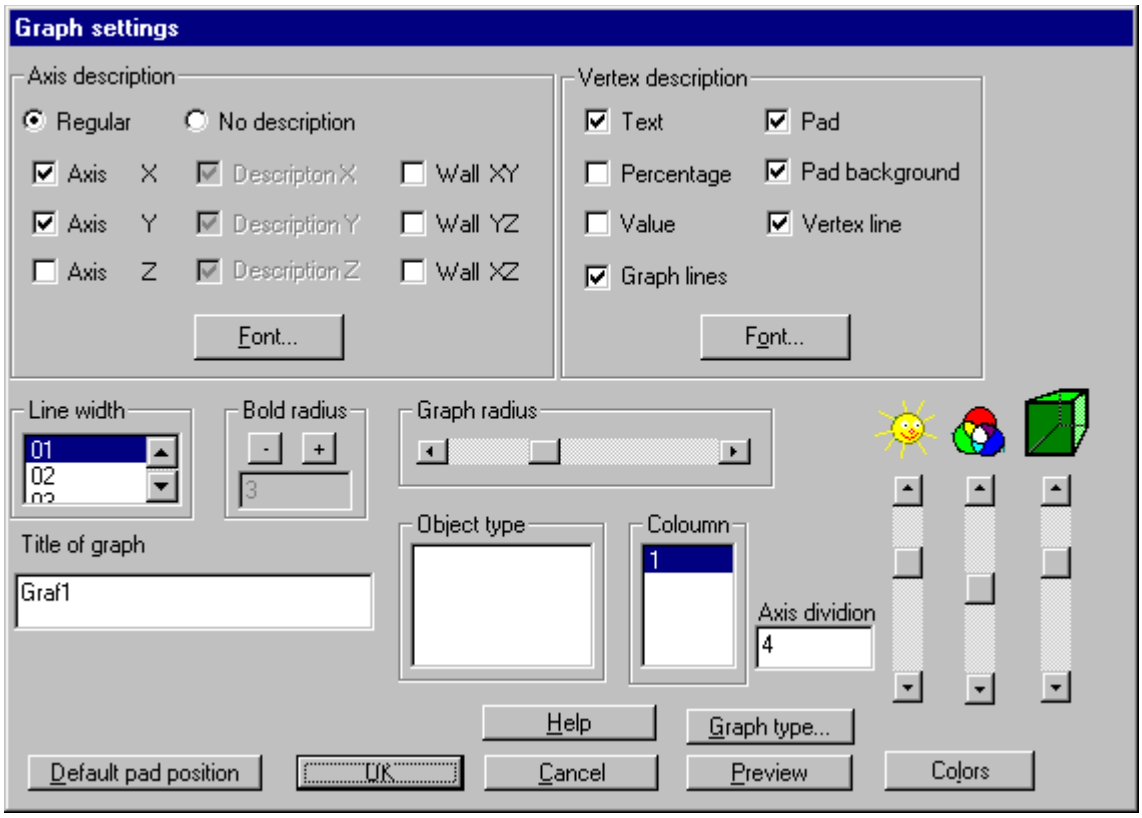
Graph Types

0 The dialog box entitled Graph Type contains radio buttons with the names of the various graph types available:

- Simple Line - simple 2D graph where the node values are represented as points in a coordinate system, connected by a line. This graph only displays the values of a single column, even for multi-column graphs. Set the displayed column in Graph Options.
- Multi Line - simple 2D graph where the node values are represented by points in a coordinate system, connected by a line. The graph displays all columns of a multi-column graph.
- Bar 2D - bar graph. Only values of a single column are displayed, even for multi-column graphs. Set the displayed column in Graph Options .
- Multi Bar - bar graph. The graph displays all columns of a multi-column graph.
- Bar 3D - 3D bar graph. The graph displays all columns of a multi-column graph. This graph type is ideal for multi-column graphs.
- Pie 2D - two-dimensional pie chart. Only values of a single column are displayed, even for multi-column graphs. Set the displayed column in Graph Options.
- Pie 3D - three-dimensional pie chart. Only values of a single column are displayed, even for multi-column graphs. Set the displayed column in Graph Options.

Graph Options

In this dialog box



0

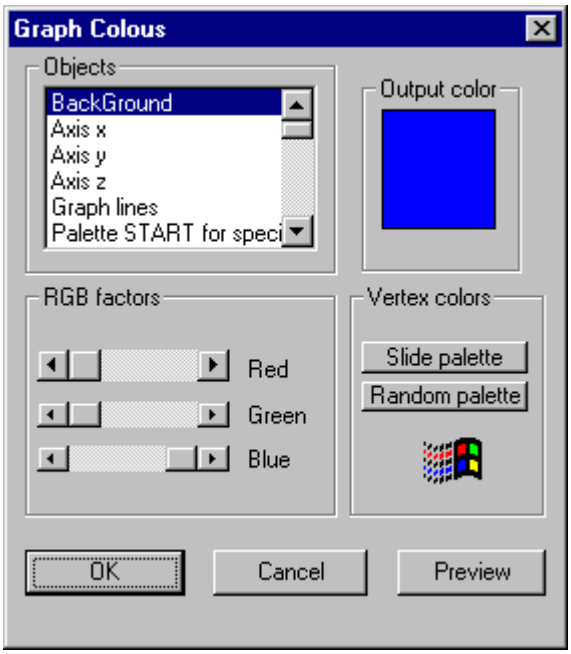
1 you can change graph settings using these fields:

- Axis Description
  - Regular and No Description - axis description Yes/No
  - Axis X, Y, Z - display axes x, y, z
  - Description X, Y, Z - description of the selected axes
  - Wall XY, YZ, XZ - wall color (Bar 3d)
  - Font - set axis font
- Vertex Description - with 2D graphs (does not apply to multi-graphs) and pie charts you can use description pads which contain: node name, (per cent values in pie charts), node value. The graph name uses the same settings as node descriptions.
  - Pad - node description in a frame
  - Pad Background - color background of the frame
  - Vertex line - the line connecting the node and the pad (if moved)
  - Text - node name
  - Percentage - percentage value of pie chart slices
  - Value - node value
  - Graph lines - node outlines
  - Font - select font for node description

- Line Width - width of the graph lines
- Bold Radius - size of the points in Simple Line and Multi Line graphs
- Graph radius - In bar graphs: the radius of nodes; in pie charts: the displacement of the slices from the center.
- Title of graph - display graph title
- Object Type - type of objects used
    - In Line graphs:
        - Sphere
        - Triangle
        - Quad - square
    - In Bar 3D graphs
        - Box - block
        - Cylinder
        - Pyramid - cone
        - Spiral
        - Flower
- Column list box - shows the data column to be displayed. In Bar 3D, Multi Bar and Multi Line graphs this item is ignored as these graph types display all columns.
- Axis Division - into how many parts the axes should be divided (and described by values).
- Scrollbars
    - Light intensity (the Sun icon)
    - Material transparency (the icon with the three RGB circles)
    - Perspective (the Cube icon)
- Buttons
    - Default Pad Position - places all description pads next to their respective nodes - only for the current column.
    - Graph type
    - Colors
    - Preview - render graph

Colors

0 In this dialog box you can change graph colors

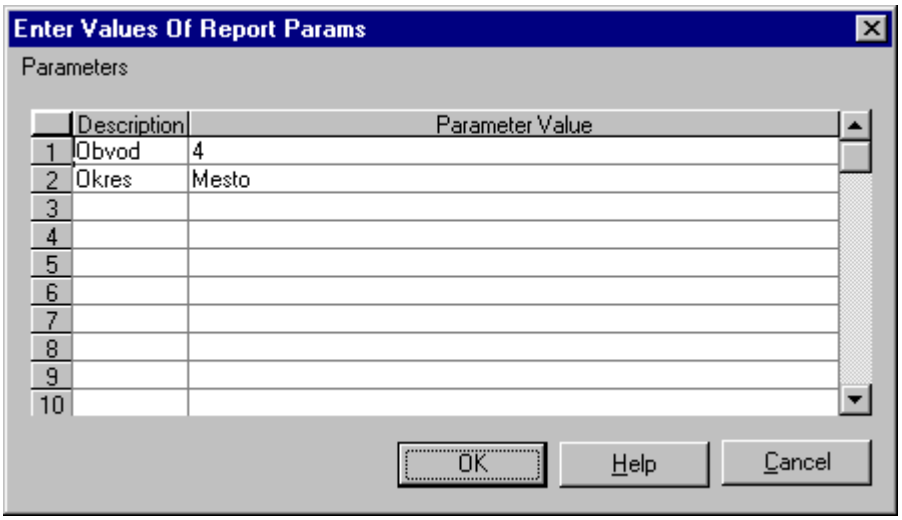


1

- modify colors of:
  - Background
  - Axis X, Y, Z - graph axes
  - Graph Lines
  - Palette START - see Slide Palette
  - Palette END
  - Text - text color
  - Remark Background - the background of node description pads.
  - Wall 1, 2, 3 - the walls of a Bar 3D graph
  - Vertex 1, …., N - the graph nodes.
- whenever it is inconvenient to specify the colors of nodes individually you can use the following functions
  - Slide Palette - generates a smooth transition from Palette START to Palette END
  - Random Palette - generates random colors for nodes

Report Data Window

If the report (or report procedures, or any WHERE Condition of the report) contains any parameters then SQLing displays the following window where you can set the values of the parameters before loading the report data:



0

The field entitled Description contains a description of the parameter. If no description was assigned to the parameter in the Properties of the report then this field only displays the name of the parameter.
Enter the value for which the report will be calculated in the Parameter Value field.
To store the entered values of variables in the database as a parameter set, select Save as from the Parameters menu. To avoid having to enter the parameter values again, select Load… from the Parameters menu to retrieve a parameter set from the database.
The parameters in the table may already have certain default values (values which were stored in the database using the Save as Default function located in the Parameters menu). You can also modify these default values.
When you have entered the values and pressed OK in the parameter dialog box, the report data itself is opened.

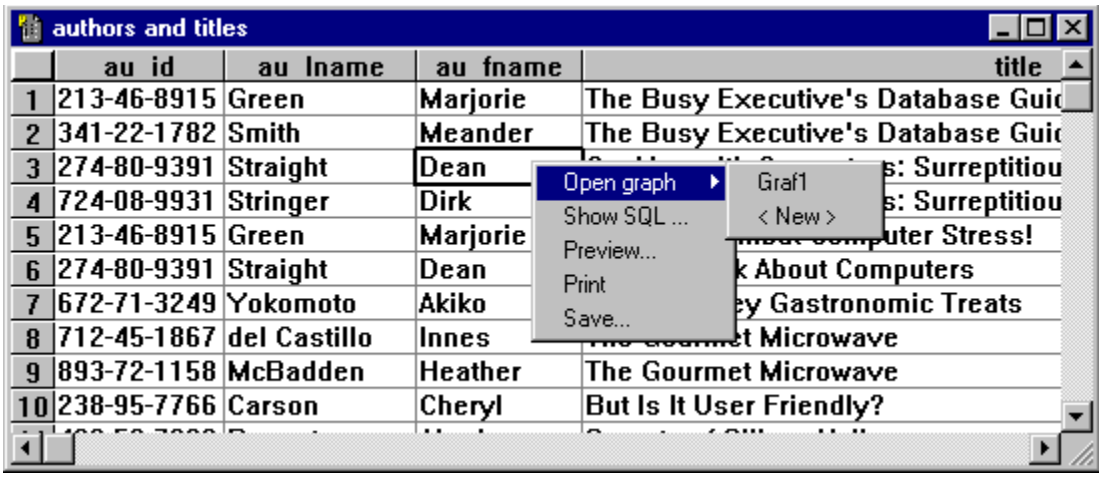


Table Properties

0 Tables support most of the standard table properties, including

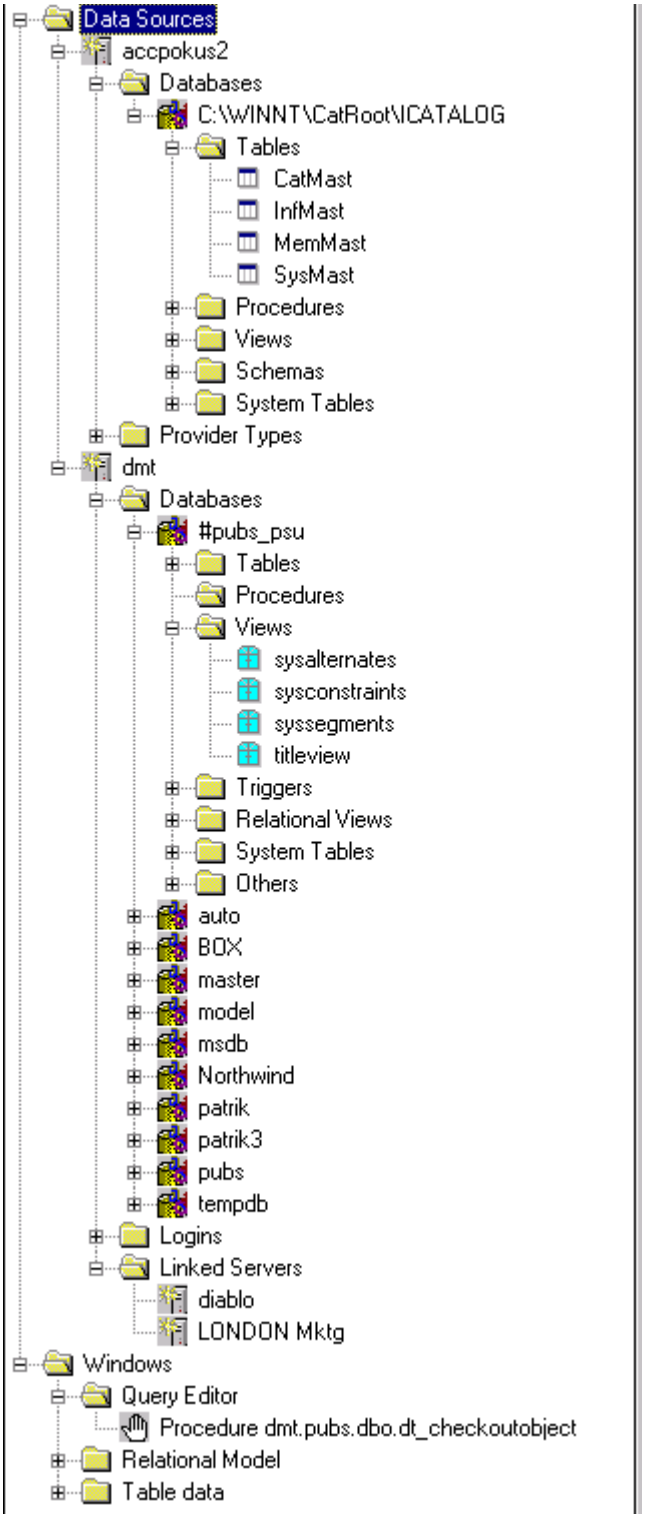
- column/row/rectangular cell area selection
- clipboard functions
- widen/shrink rows/columns
- sort columns by double clicking on the first table row

0 Shortcut menu

Press the right mouse button to open a menu containing the following functions:

- Open Graph – opens a submenu containing the names of graphs which were created for the report and a command, <New>. Select any of the existing graphs to view it. This will not work if the columns of the report visualized in the graphs were removed. In such case SQLing shows a warning and enables you to delete the invalid graph. Select <New> to display a new graph. If any report columns are highlighted when this function is called, the graph will only show the highlighted columns. If no columns are highlighted then the graph shows all columns of the report. Two types of columns are distinguished - numeric and text columns. Numeric columns represent vectors with data and text columns represent node titles. If more text columns have to be displayed in a graph, their values (texts) are collated into a single body of text (divided by spaces). The graph will not be displayed unless the columns to be displayed include at least one numeric and one non-numeric column.
- Show SQL – displays the SELECT instruction which selects the report data.
- Preview – set the report printing options.
- Print – print report.
- Save - saves the report data in a file as tab delimited text.

Tree Browser



0

Trees are frequently used in Windows and it is assumed that the user is sufficiently familiar with their function and controls. This section only describes the extended

functions and object type-specific popup menus.

The tree was designed to be completely controllable by keyboard. In addition to the standard use of the keyboard, i.e. browsing by arrow keys and searching for an object in the current branch by entering the prefix of its name, the following keyboard shortcuts are available:
Change branch in the current database:
- Alt T - table
- 0 Alt P - procedure
- 1 Alt V - view
- 2 Alt S - system table
- 3 Alt G - triggers
- 4 Alt R - relational model view

F5 - refresh branch
0 F1 - show the full tree path to the current object (e.g. *Data Sources/Diablo/Databases/pubs/Tables/authors*) in the info bar
1 Alt F2 or Tab - move focus to the main area with MDI windows.
2 Ctrl t F2 - as Alt F2 + minimize the tree browser
3 Alt F1 - set focus to the tree browser (and any restore of the minimized browser by Ctrl F2)
4 Enter - display the data of the table/view/report, execute procedure, design other object types
5 Ctrl Enter - design a table/procedure/view/report
6 Delete - drop the selected object
Ctrl X, C, V - cut, copy, paste object (the Copy/Paste function is in fact a transfer) Same results can be achieved by applying the Drag&Drop method on the objects in the tree: catch/hold an object of one database and drag/drop it over another database to move it. If the object is dropped over the source database, its copy is created and a number is appended to its name (authors -> authors1).

Objects are generally divided into databases, database objects, open windows and other, miscellaneous objects (logins, linked servers, etc.)

The popup menus of all objects except windows with no popup menu contain an item, Refresh, which is used to refresh the list of objects or the subtree of the selected object, and another item, Drop, which removes the current object (this function is obviously not available for Server/Data Source objects).

Data Source - this subtree contains all connected servers. To connect to new servers/data sources from the popup menu select Connect…, to disengage existing connections select Disconnect, to view the activity on the selected server select View Processes, or open the Query Editor.

The popup menus of Databases contain the same functions as the Current Database menu, but the popup functions always work with the database over which the menu was

invoked.

0 The Browser knows the following types of database objects:
1 Tables
2 Procedures - stored procedures
3 Views
4 Triggers
5 Relational Views - views of the relational model of the database
6 System Tables
7 Extended Procedures
8 Defaults - defaults and default constraints
9 Rules - checking rules
10 Users - database users
11 Reports - reports and queries created in a database
12 Graphs - report graphs
13 Pset - report parameter set

Some of the (less frequently used) object classes are included in the Others subtree of the current database. Any database object classes can be moved into this subtree (Drag the class -> Drop it over the field named Others), and any classes can be removed from this subtree (Drag the class -> Drop it over a database name). This obviously does not apply to subclasses like Graphs or Pset.

All database object types contain functions similar to New, View and Design, and also Open. These functions are described the appropriate sections dealing with the class. One of the functions View, Open, Design is also executed as the main browsing function when the user double clicks on the object or presses Enter on the object name.

0 The classes can logically be divided as follows:

Data objects - (tables, views, system tables). The related popup menu contains (the available items depend on the type of the current object and database) a subset of the following functions:

- New - create a new table (not view).
- Open - display a window with the table data. System tables are of another table type, although this type is similar to a user table. The most significant difference is apparent in the table named sysprocesses which is so important that its functionality had to be extended.
- View or Design- display the (structure) definition of the selected table or the selected view.
- Edit Notes - edit notes to the selected table.
- View All Notes - display the notes of all users attached to the selected table.
- Script Data… - save the data of the selected table into a file.
- Select COUNT(*) - determine the size of the selected table.
- Last Line WHERE… - display the last record in the table.
- WHERE Open… - display the table data matching the defined WHERE Condition.

0 Source-code objects- (procedures, triggers, views, defaults, rules). The Popup menu may contain the following:

- New - create a new objects (e.g. a procedure). This button is only available while browsing an actual database.
- Design - open the procedure designer with the object definition.
- Execute - execute the procedure.
- Edit notes - create notes to the procedure.
- Plan size - determine procedure size.
- View help - display the first comment in the procedure body.

0 Users

- View - display the definition of a user.
- New - create a new user.
- Design - modify the selected user. It is not possible to create and edit groups.

0 User Data Types

- View - display the data type definition.
- New - create a new data type.
- Design - modify the selected user data type.

The following object classes are added to the data model by SQLing. These classes are only visible from within SQLing.

Relational Views - views of the relational model of the database containing the (selected) database tables and relational model notes. The layout of the tables may be different than in the global relational model. Create a relational view from the global relational model using the following functions available in its popup menu: Select/Deselect, View/Save… and Add Note.

Open - opens a relational model of the database and automatically loads the view.

Reports - reports and queries existing over a database

Graphs - report graphs

Pset - report parameter set

Reports and Downsized Databases

Reports display real table data which are not present in a downsized database. Reports can only be created on actual databases containing real data.

Clicking on New or Design will open the report designer in which you can develop a New report or modify/view an existing report (Design). The Designer offers the best possible image of the report structure. While these two buttons will mostly be used by somebody designing reports, the third function, Execute, is mainly intended for the end user.

Click on Execute to view the report data without any additional report structure information.

Current Database Menu

This section explains the options in the Current Database menu which is located on the main menu level. You will learn how to modify a database on the SQL Server correctly and conveniently, using a working (downsized) database, how to retrieve information about the current database and how to quickly and easily create database documentation.

Current database

All commands included in this menu work with the database specified in the Database Combobox

Menu commands can be divided into four groups.

- Tools
  - MDI Browser
    - Tables
    - Procedures - opens an MDI browser for code objects
  - Open Relational Model
  - Open Fixed Query Editor - opens the Query Editor. Any queries executed in the Query Editor will apply exclusively to the defined current database and not to the database that might be current at the moment when the query is executed.
- Database Modification Commands
  - Downsize …
  - Upsize …
  - Save downsize…
  - Load downsize
  - Touch…
- Database Maintenance Commands
  - Options…
  - Compare
  - Synchronize…
  - Transfer
  - Rename
  - Drop…
- Other Commands
  - Info…
  - Make document…

Downsize…

When changes have to be made to the database on the SQL Server, it may often be more convenient to create a so-called working or downsized database the structure of which is identical with the SQL Server database. The working database can be extensively modified and then the database on the SQL Server is modified. Modification of the SQL Server database by means of a working (downsized) database has two major advantages:

1. Changes may take a long time to implement on the SQL Server database. When a working (downsized) database is used, more changes can be made in succession and then the actual database located on the SQL Server is updated.

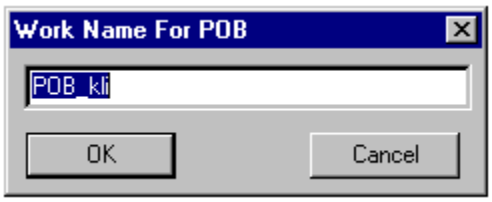
0 2. The database on the SQL Server remains consistent.

1 Downsizing the SQL Server database to SQLing

1. Select the server and the database you would like to downsize (make a working copy of) from the lists in the combo boxes Connection and Database.
2. Select command Downsize … from the menu Current Database.

A dialog box is displayed on the screen showing the name of the new working database. You do not have to devise new names as SQLING automatically prepares the title by fitting the current data into this template

<current database name>_<NT username>



3. Press OK if you agree with the ready-made title.

4. If you wish to change the name, just type it in and press OK.

0 Working database

SQLING automatically adds a double cross prefix (#) in front of the name of the working (downsized) database. The working database is then created by duplicating the structure of the current database. The name of the newly created database is added into the Database combo box in the main application window.
The working database can be modified arbitrarily while the function of the original database remains undisrupted. When all the desired changes have been made to the working database, run the function Upsize To SQL to implement the same changes on the original database.

For each real database you can create only one downsize database. If you try to create another database, Patrik will display dialog window Program Error and it does not react

on downsize request.

Any changes made to the working (downsized) database can be saved by selecting Save downsize from the Current Database menu. For more information refer to the topic Save downsize.

More experienced users may be interested in a description of the downsize process.

Upsize…

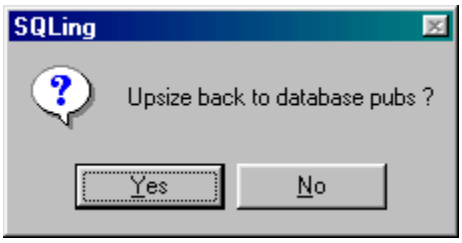
Upsize a working (downsized) database into an actual database immediately after modifications have been made to the working (downsized) database or upsize a previously saved working (downsized) database. If you wish to retrieve a previously saved working (downsized) database select Load Downsize from the Current Database menu.

0 Upsizing a working database

1 1. Select a working database you wish to upsize in the Database combo box.

2 2. Select Upsize… in the Current Database menu.

3 A dialog box is displayed.



0 3. Press OK.

The upsize process will be launched. This process applies all the changes that were made to the working (downsized) database on the full-sized, original database. The upsize process is continuously logged (displays its results) in the Query Results window. The last status message displayed in the window will be: Upsize Finished… . This means that the function has been completed. The working database is then no longer needed and it is deleted (dropped).

More experienced users may be interested in a description of the upsize process .

Save downsize

The Save Downsize command is used to save the changes made to the working database. This function enables the upsize process to be executed later or be applied to several real databases on different SQL servers, provided you manage two or more identical real databases on two or more SQL Servers and you wish to implement the changes on all of them.

0 Save working database

1. Select Save Downsize from the Current Database menu.
2. The Save As dialog box appears.
3. Type in a filename and select the folder where you wish to save the file.
4. Press OK.
4. You can load the saved downsize file into an identical database (on a different server) and upsize it there. This will save time because identical changes will not need to be made twice on databases with the same structure (e.g. the databases of your company and your customer).

Load downsize

Open a previously saved downsized (working) database

1. Select Load Downsize in the Current Database menu.
The Open dialog box appears.

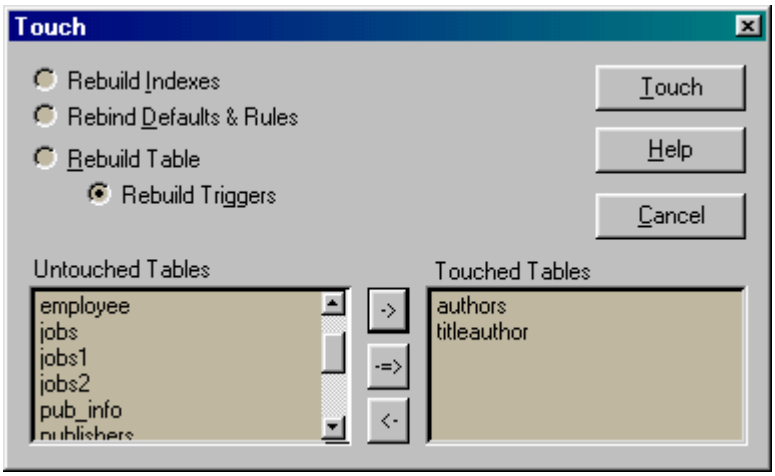
2. Select the file you wish to load.

3. Press OK.

Touch

Use this command to simulate that selected tables of the working database were changed (though no real modifications are made). This will cause certain table properties (or even whole tables) to be re-generated when the Upsize function is executed.

0 In this dialog box

1 

2 1. Select the type of re-generation to be run for the selected tables.

- Rebuild Table - re-generated everything related to the tables, including whole tables
- Rebuild Triggers - re-generate triggers of the selected tables
- Rebuild Indexes - rebuild tables indexes
- Rebind Defaults & Rules - unbind and rebind defaults and rules of table columns

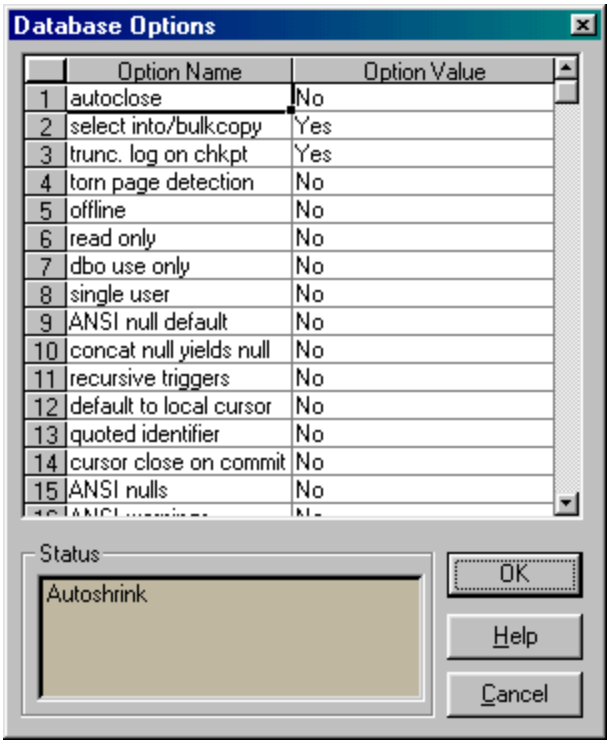
2. Use these buttons: -> (move selected tables) and -=> (move all tables) to move the tables you want to have re-generated from the Untouched Tables field into Touched Tables.

0 3. Press Touch.

For more information on the re-generation process see the topic detailed description of the Upsize function.

Options

If you select Options in the Current Database menu the dialog box Database options is displayed.



0 The Database Options dialog box contains two subgroups.

- Options - consists of a database options table. Set the options and press OK to apply changes.
- Status - displays the potential exceptional status of the current database.

For more information on the options and database states see SQL Server Books Online.

Compare

In this chapter you will learn how to detect differences between two databases and how to synchronize data and structures of two databases. Using the Compare menu commands you can compare the structure of code objects, table structures, history and table content (data).

Compare - With Another Database… - compare the structure of two databases
Compare - Structure Of Selected Tables… - compare the structure of selected tables between two databases
Compare - Source Codes… - compare procedure, trigger, etc. source codes
Compare - Data In Tables… - compare and/or synchronize table data
Compare - Permissions... - compare user permissions with relation to objects

Compare with another database

The command Compare With Another Database is used to compare the structure of databases. It does not perform a comparison of table content (data) or source codes of procedures, triggers, defaults or rules. These elements are compared by:

Compare - Data In Tables…
Compare - Source Codes…

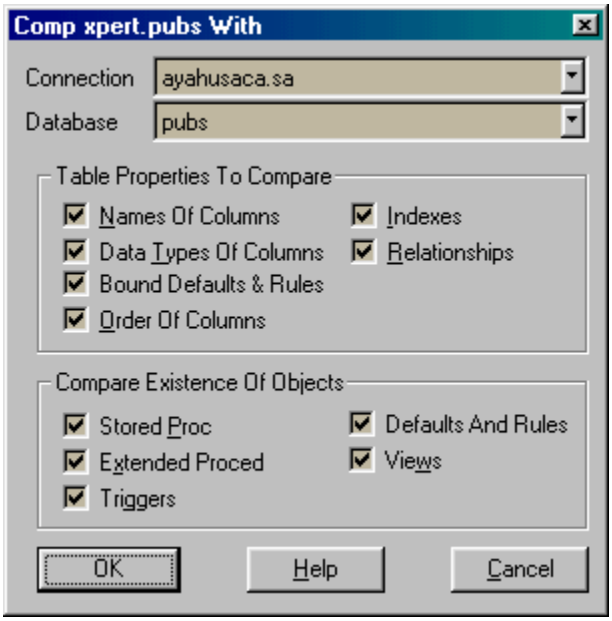
Compare the structure of the current database with another database

1. Select the database you would like to compare in the Database combo box.

2. Enter the Compare menu in Current Database.

3. Select With Another Database.

A dialog box appears entitled Comp *server_name.database_name* With.



1. In combo boxes Connection and Database select the server and database with which you want to compare the current database.

2. In the To Compare group of Table Properties select the check boxes corresponding to those table characteristics you want to compare.

Names Of Columns - for each of the columns of a table checks whether a column of the same name exists in the other table.
Note: Couples of columns with the same name will be compared in the comparison of other features.

Datatypes Of Columns - discovers the differences between data types (including

their length and NULL possibilities) of the compared columns.

Bound Defaults & Rules - checks the existence of rules and defaults for the compared columns. This is no longer a comparison of the source codes of the compared objects.

Order - determines the differences in the order of the compared columns in the table.

Indexes -discovers differences in table indexes.

Relationships - discovers differences in relations between tables.

3. In the Existence group check the check boxes of those database objects whose existence you would like to compare.

4. Press OK to confirm your selection.

The Query Results window which displays results of the comparison is of the Results type.

Compare Structure of Selected Tables

Use the command Compare Structure of Selected Tables to compare the structure of the selected tables in one database with tables of the same name in another database. This function does not compare table content (data). Compare table content by Compare - Data In Tables….

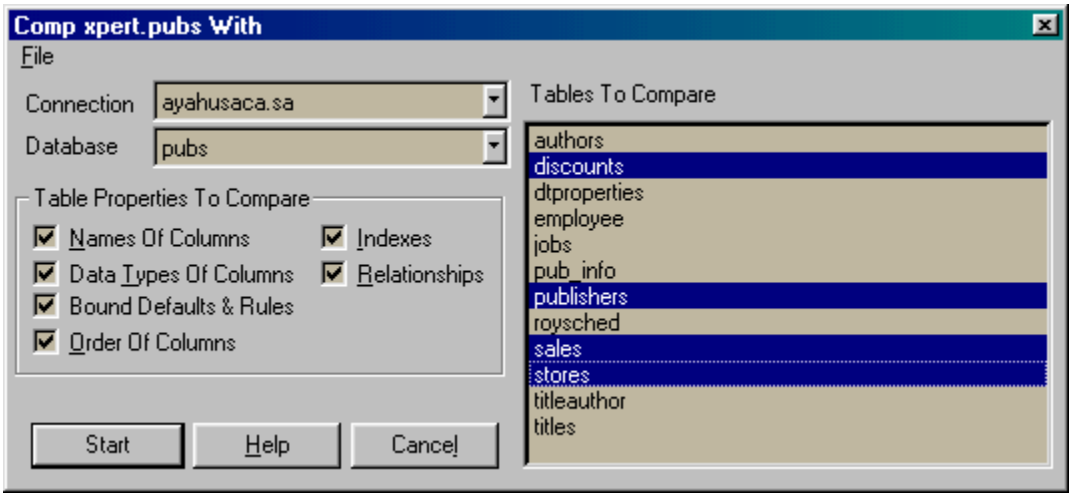
Compare the structure of selected tables of one database with tables of the same name in another database.

1. Select the database whose tables you wish to compare in the Database combo box.

2. Open the Compare menu in Current Database.

3. Select Structure of Selected Tables.

The following dialog box appears.



1. In combo boxes Connection and Database select the server and database with which you want to compare the tables of the current database.

2. In the To Compare group of Table Properties select the check boxes corresponding to those table characteristics you want to compare.

Names of Columns - for each of the columns of a table checks whether a column of the same name exists in the other table.
Note: Couples of columns with the same name will be compared in the comparison of other features.

Datatypes of Columns - discovers the differences between data types (including their length and NULL possibilities) of the compared columns.

Bound Defaults & Rules - checks the existence of rules and defaults for the compared columns. This is no longer a comparison of the source codes of the compared objects.

Order - determines the differences in the order of the compared columns in the table.

Indexes -discovers differences in table indexes.

Relationships - discovers differences in relations between tables.

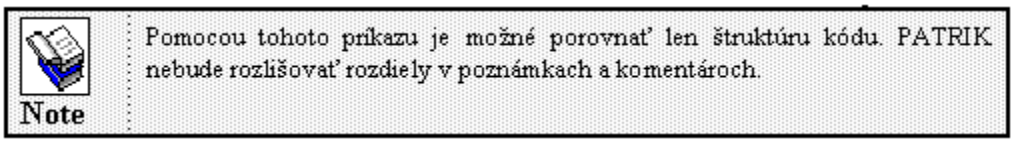
3. In the Tables to Compare list box select those tables whose structure you wish to compare with tables of the same names from another database.

4. Press OK to confirm your selection.

The Query Results window which displays results from the comparison is of the Results type.
The File menu of the dialog box.

Compare Source Codes

The Compare Source Codes function compares Transact SQL source codes (procedures) of two databases.

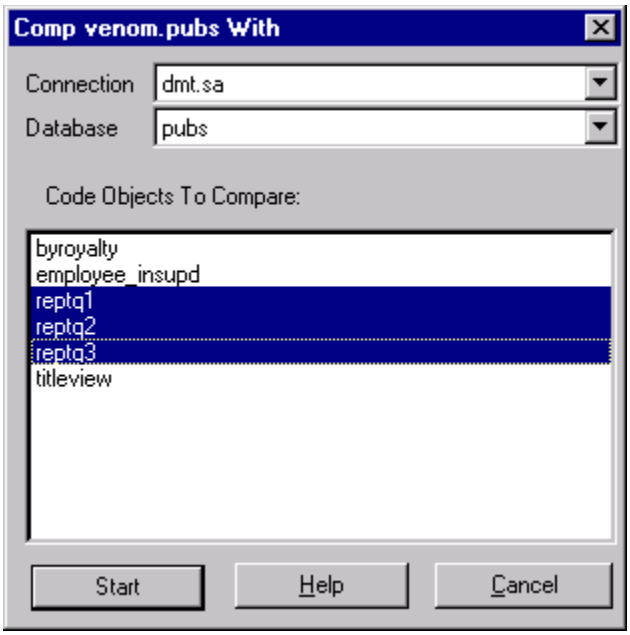


Compare source codes of two databases

1. In combo boxes Connection and Database select the server and database whose source codes you want to compare with the source codes of objects bearing the same name, located in another database.

2. Open menu Compare in Current Database.

A dialog box entitled Comp *server_name.database_name* With appears.



1. In combo boxes Connection and Database select the database whose source codes you would like to compare with the current database.

2. Select the source objects you wish to compare in the list of the Code Objects to Compare field.

3. Press Start to confirm your selections.

Query Results are displayed showing the row, column and the first different word in a procedure for each of the differing pairs of procedures. This function only displays the first difference and then moves to the next object.

The Query Results window is of the Results type.

Compare Data in Tables

Important Notes

1. Data can only be compared in primary key tables.
2. With MSSQL 7.0, correct comparison of text/image fields (i.e. comparison over the whole length of the fields) is only possible when connecting via OLE DB.

The Compare Data in Tables command is used to compare table data between two databases.

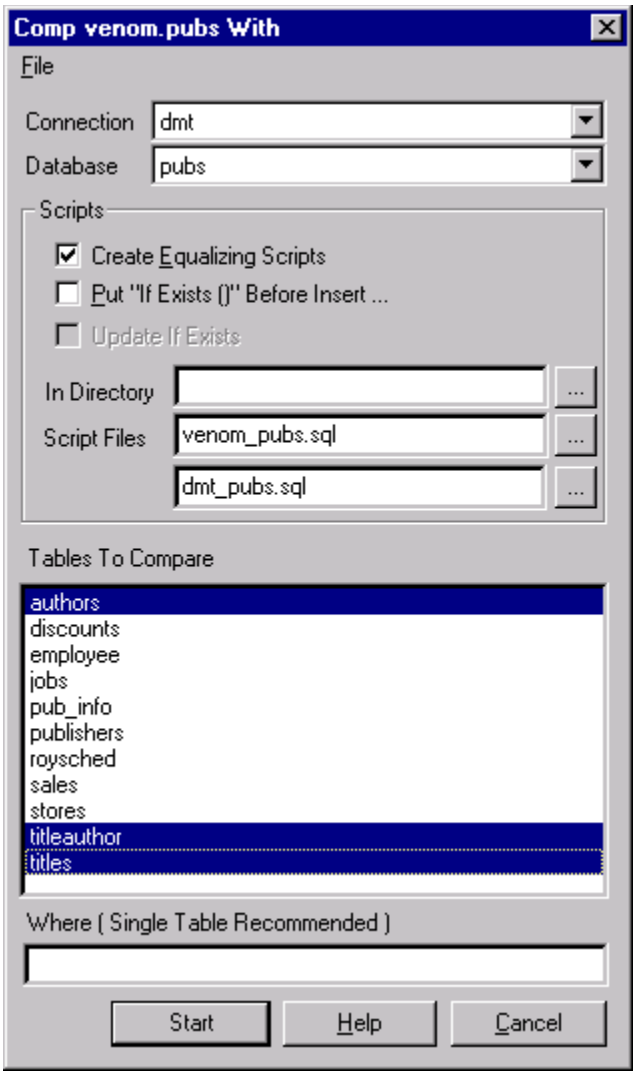
Comparing table data between two databases

1. In combo boxes Connection and Database select the server and the database whose table data you want to compare with data in tables bearing the same name in another database.

2. Open menu Compare in Current Database.

3. Select Data In Tables

A dialog box entitled Comp *server_name.database_name* With is displayed.



1. In combo boxes Connection and Database select the server and the database whose table data you want to compare with the current database.

2. Locate the group Scripts
    1) If you check Create Equalizing Scripts SQLING will create two files which can later be used to synchronize the table data of the two compared databases. The default script names are:
        1. *Server1_database1*.sql - contains a set of Insert, Delete and Update commands. When this file is run on server1, the data in database1 will be equalized with the table data in database2 on server2.
        2. *Server2_database2*.sql - contains a set Insert, Delete and Update commands. When this file is run on server2, the data in database2 will be equalized with the table data in database1 on server1.

        These default names can be changed in fields In Directory and Script Files.

2) Put "If Exists ()" Before Insert… - if checked, every occurrence of the INSERT command in the equalizing scripts will be preceded by a condition which will prevent duplicate insertion of records into the database in case the script is run more than once.

3) Update If Exists - can only be checked if Put "IF Exists ()" Before Insert is selected. This option inserts the following structure into the equalizing scripts

ELSE
UPDATE <table> SET attr1=value1 WHERE <the Where Condition of the Inserted row>
UPDATE <table> SET attr2=value2 WHERE <the Where Condition of the Inserted row>
...
UPDATE <table> SET attrN=valueN WHERE <the Where Condition of the Inserted row>
END

after this line: *IF EXISTS (...) INSERT <table> VALUES (value1, value2, ..., valueN)*.

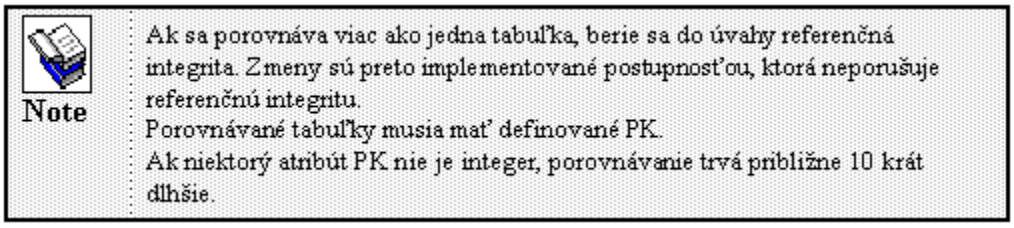
This function can be used in a specific case when M similar tables and one template table exist, so that a single equalizing script can be run on all of the tables

3. Select the tables you want to compare in the field Tables to Compare.

4. Enter a condition to compare only the portion of the table which contains the data matching the condition in the field Where (Single Table Recommended). The Where condition applies to all selected tables, it is therefore recommended to compare only one table.

5. Press Start to confirm your selection.

A query is executed and the differences between the compared tables are displayed in the Query Results window.
Query Results is a Results-type window.



Synchronize Data in Tables
Dialog Box File Menu.

Synchronize Data in Tables

Synchronize data between two tables

1. Set the database in which you wish to synchronize tables as the current database.

2. Press Query () in the main application window.
The Query Editor window is displayed.

3. Select Load from the Shortcut Menu
An Open dialog box appears.

4. Highlight the file created by the data compare function.
The name of the file which will synchronize the data in the current database with the data in the compared database has the following format:
server_name_current_database_name.sql.

5. Press OK.
The file is loaded into the Query Editor window.

6. Press the Execute button () in the Query Editor window.

Compare permissions

Use the command Compare Permissions to compare permissions between two databases.

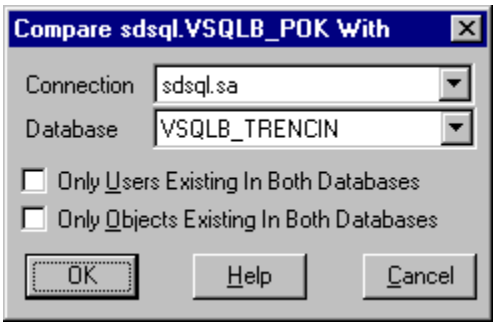
Compare permissions between two databases:

1. In the combo boxes Connection and Database select the server and the database for which you wish to compare permissions with the permissions of objects bearing the same name, located in another database.

2. Open the Compare menu in Current Database.

3. Select Permissions

A dialog box entitled Compare *server_name.database_name* with appears.



1. In the combo boxes Connection and Database select the server and the database whose permissions you wish to compare with the permissions in the current database.

2. Check option Only Users Existing in Both Databases if you do not want the results to contain references to permissions of users who only exist in one of the databases.

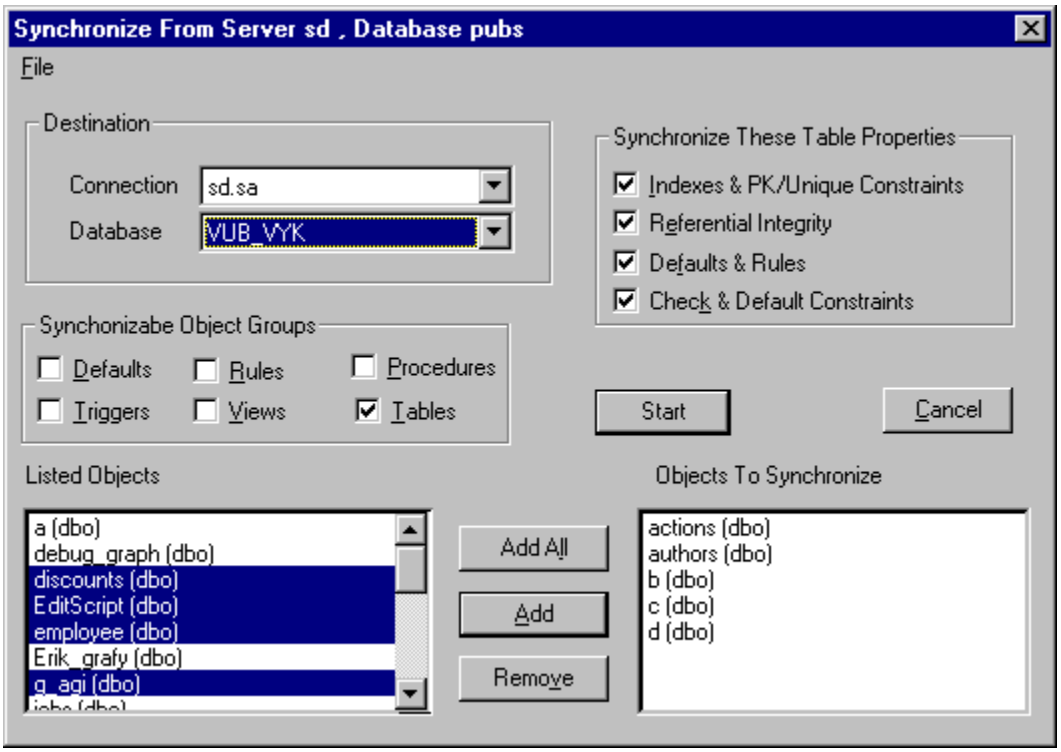
3. Check option Only Objects Existing in Both Databases if you do not want the results to contain references to permissions for objects (tables, views, procedures) existing in only one of the databases.

4. Press OK.

The Query Results window appears, displaying a line for each of the differences discovered in permission tables of the two databases.

Synchronize Database Structure.

The Current Database menu contains the function Synchronize which synchronizes the structure of databases containing data (without destroying the data). The function synchronizes to the widest possible extent until ambiguity appears. Only those tables are synchronized which have been selected and are both missing from the synchronized database and have the same attributes, data types and their order in both databases. This means the synchronization affects indexes, bound defaults and rules, defined constraints (all types), relations and triggers. Only those source-code objects (e.g. procedures or defaults etc.) are synchronized which either do not exist in the synchronized database or the executive portions of their source code differ between the synchronized database and the source database.
When **Synchronize** is selected a dialog box similar to **Transfer** appears.



The user can select the Defaults, Rules, Procedures, Triggers, Views and Tables, which should be synchronized. With tables, the properties to synchronize can be specified. The user can select the synchronized database for which a script named **synch_server_database.sql** will be generated. **Server** is the synchronized server and **database** the synchronized database. Press **Start** to generate the script. When this script is run in the synchronized database it synchronizes the structure of objects selected for synchronization. If the structure of a table differs between the databases the table is not synchronized and the following line is added for that table into the script:

PRINT '----------> Table %s has different structure in each of selected databases, therefore no synchronization in this table is built

If all objects of the same class were synchronized the function generates another script named **drop_server_database.sql** where **server** is the synchronized server and **database** is the synchronized database. The script contains commands to remove the superfluous objects of those classes for which all objects were synchronized.

Transfer

This section describes how to move the current database on the SQL server into another database on the same or on another SQL server provided that both servers are connected either by LAN or by WAN. You will also learn how to transfer the current database on the SQL server into a target database on another server by means of portable storage media.

Online export… .- online database transfer (or its part)
Off-line export… - offline database transfer (or its part) to disk
Off-line Import… - transfer database from disk to database on SQL server
Metadata Layer… - transfer a layer of data stored in the SQLingdatabase

Online export

Use this command to export some or all objects of the current database into a target database which may be located on another SQL server provided that the servers are connected by LAN or WAN.

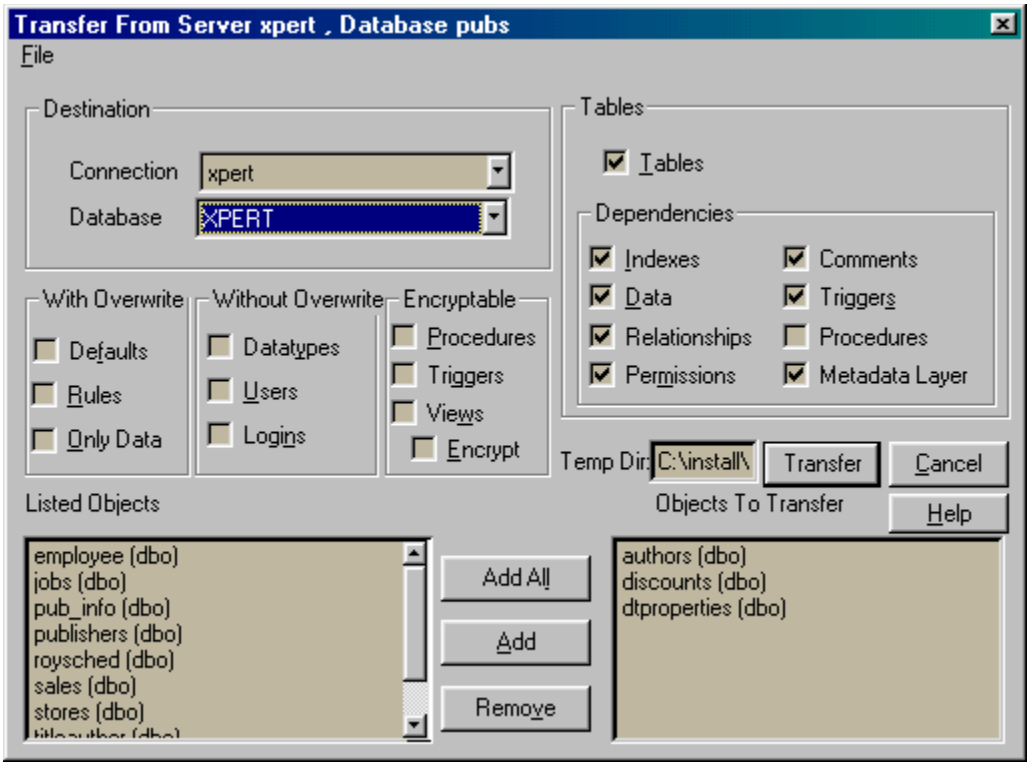
Online database transfer

1. Select the database you wish to transfer.

2. Select Transfer in the Current Database menu.

3. Select Online Export.

A dialog box entitled Transfer from Server *server_name*, Database *database_name* appears.



4. In the group Destination
   a) In the combo box Connection select the name of the target server where you wish to transfer the current database.
   b) In the combo box Database select the name of the target database into which the current database should be transferred.

5. In the group With Overwrite

(If the selected objects already exist in the target database SQLING replaces their content with the content of objects of the current database.)

Repeat the following steps on each item of the group With Overwrite:
Select the check boxes of the objects you would like to transfer.
The field named Listed Objects will list all objects of the selected type.
In the field Listed Objects select the object to be transferred and press Add.
Press Add All if you want to transfer all objects of the specified type.
To prevent the transfer of an object that has been added into the field named Objects to Transfer select that object and press Remove.

6. In the group Without Overwrite

(If the selected objects are already located in the target database they will not be replaced by the objects of the current database.)
Repeat the following steps on each item of the group Without Overwrite:
Select the check boxes of the objects you would like to transfer.
The field named Listed Objects will list all objects of the selected type.
In the field Listed Objects select the objects to be transferred and press Add.
Press Add All if you want to transfer all objects of the specified type.
To prevent the transfer of an object that has been added into the field named Objects to Transfer select that object and press Remove.

7. In the group Encryptable

(When the check box labeled Encrypt is checked the source codes of Procedures, Triggers and Views are encrypted to prevent the misuse of the principles they contain. If the check box Encrypt is left unchecked the source codes of Procedures, Triggers and Views will be transferred without encryption.)
Repeat the following steps on each item of the group Encryptable:
Select the check boxes of the objects you would like to transfer.
The field named Listed Objects will list all objects of the selected type.
In the field Listed Objects select the object to be transferred and press Add.
Press Add All if you want to transfer all objects of the specified type.
To prevent the transfer of an object that has been added into the field named Objects to Transfer select that object and press Remove.

8. In the group Tables

Check the check box named Tables.
A list of tables in the current database appears in the field Listed Objects.
In the field Listed Objects select the tables to be transferred and press Add.
Press Add All if you want to transfer all tables.
To prevent the transfer of tables that have been added into the field named Objects to Transfer select a table and press Remove.

9. In the group Dependencies

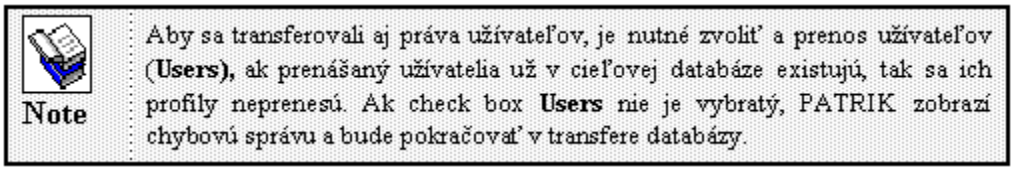
Check the appropriate check boxes to select the attributes which you wish to be transferred with the tables.

Indexes - table indexes

Data - table data
Relationships - relationships between tables (if the Metadata Layer check box is checked relationship attributes and table positions in the relational model will also be transferred.)
Permissions - user permissions for certain tables (see note below)
Comments - attribute and table comments
Triggers - triggers
Procedures - procedures
Metadata Layer - table history

10. In the field entitled Temp Dir select a folder located on your computer. This folder will be temporarily used to store table data during the transfer. The temporary directory must have enough free space to store the data of the largest transferred table.

11. Press Transfer to confirm your selection.



Dialog box File menu.

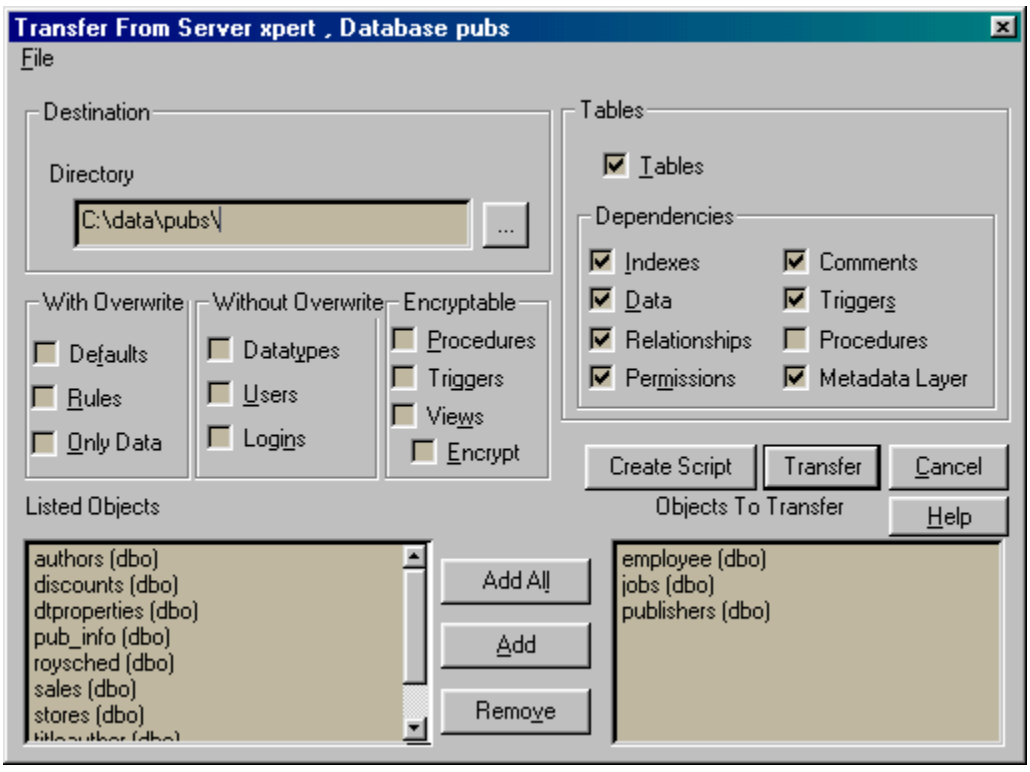
OFF-line Export

This function exports some or all objects of the current database on the SQL server into a target database on another SQL server when the servers are not connected by LAN or WAN. Portable storage media is used in the transfer.
The function can also create a structural script for the selected portion of the database.

Offline Export of some or all objects.

1. Select the database you wish to transfer.
2. Select Transfer in the Current Database menu.
3. Select Off-line Export.

A dialog box entitled Transfer from Server *server_name*, Database *database_name* appears.



1. Select the temporary location of the database in the field entitled Directory under Destination. The transferred portion of the database can be restored into another database on another SQL server from this folder (or its copy elsewhere) by the Offline Import function.

5. In the group With Overwrite

(If the selected objects already exist in the target database SQLING replaces their content with the content of objects of the current database.)

Repeat the following steps on each item of the group With Overwrite:
Select the check boxes of the objects you would like to transfer.
The field named Listed Objects will list all objects of the selected type.
In the field Listed Objects select the object to be transferred and press Add.
Press Add All if you want to transfer all objects of the specified type.
To prevent the transfer of an object that has been added into the field named Objects to Transfer select that object and press Remove.

6. In the group Without Overwrite

(If the selected objects are already located in the target database they will not be replaced by the objects of the current database.)
Repeat the following steps on each item of the group Without Overwrite:
Select the check boxes of the objects you would like to transfer.
The field named Listed Objects will list all objects of the selected type.
In the field Listed Objects select the objects to be transferred and press Add.
Press Add All if you want to transfer all objects of the specified type.
To prevent the transfer of an object that has been added into the field named Objects to Transfer select that object and press Remove.

7. In the group Encryptable

(When the check box labeled Encrypt is checked the source codes of Procedures, Triggers and Views are encrypted to prevent the misuse of the principles they contain. If the check box Encrypt is left unchecked the source codes of Procedures, Triggers and Views will be transferred without encryption.)
Repeat the following steps on each item of the group Encryptable:
Select the check boxes of the objects you would like to transfer.
The field named Listed Objects will list all objects of the selected type.
In the field Listed Objects select the object to be transferred and press Add.
Press Add All if you want to transfer all objects of the specified type.
To prevent the transfer of an object that has been added into the field named Objects to Transfer select that object and press Remove.

8. In the group Tables

Check the check box named Tables.
A list of tables in the current database appears in the field Listed Objects.
In the field Listed Objects select the tables to be transferred and press Add.
Press Add All if you want to transfer all tables.
To prevent the transfer of tables that have been added into the field named Objects to Transfer select the tables and press Remove.

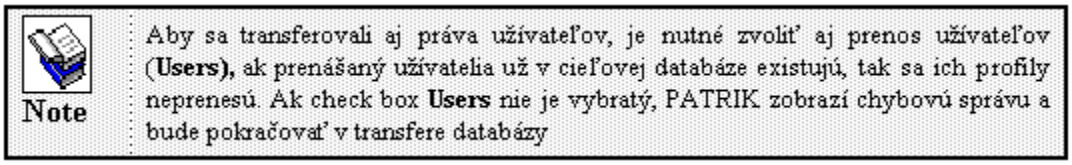
9. In the group Dependencies

Check the appropriate check boxes to select the attributes which you wish to be transferred with the tables.

Indexes - table indexes
Data - table data
Relationships - relationships between tables (if the Metadata Layer check box is checked relationship attributes and table positions in the relational model will also be transferred.)
Permissions - user permissions for certain tables (see note below)
Comments - comments on attributes and tables
Triggers - triggers
Procedures - procedures referring to tables
Metadata Layer - table history

10. Press Transfer to confirm your selection and to transfer objects to disk (if selected for transfer, data will be stored in BulkCopy .bcp files in the specified folder).

Press Create Script if you want to create a T-SQL script which consists of connected .sql files normally created by the Offline Export function including data INSERTs. SQLing will prompt you for the file name under which the script should be saved. When connected to MSSQL 7.0 text/image field data can only be correctly scripted if connecting via OLE DB.



Dialog box File menu.

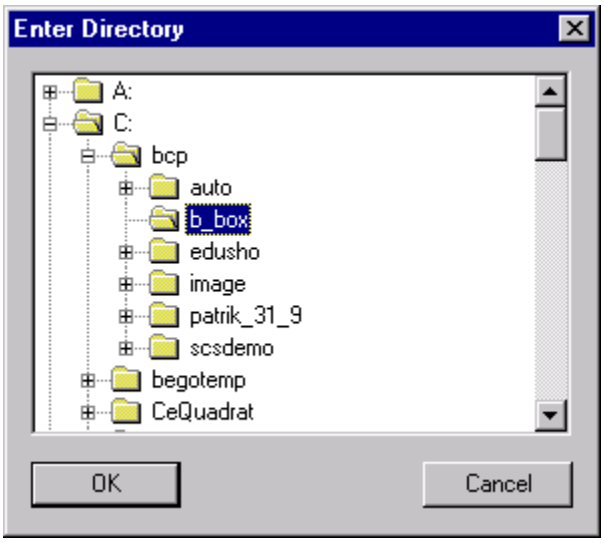
OFF-line Import

This function imports database objects into the current database from the source database located on another SQL server not connected via LAN or WAN. Portable storage media are used in the offline transfer.

Import database objects from a disk folder

1. Select Transfer in the Current Database menu.
2. Select Off-line Import.

A dialog window entitled Enter Directory appears.



3. Select the folder from which object will be imported into the current database.
4. Press OK.

Transfer Metadata Layer

This function performs an additional transfer of the layer that SQLING adds to any processed database. If a database was transferred by another application then it does not contain the layer stored in the SQLing database. This layer includes comments to tables and attributes, position of windows in the relational model, properties of relationships, history of the existing tables and relationships in views.

Click here to learn about the structure of the SQLing database.

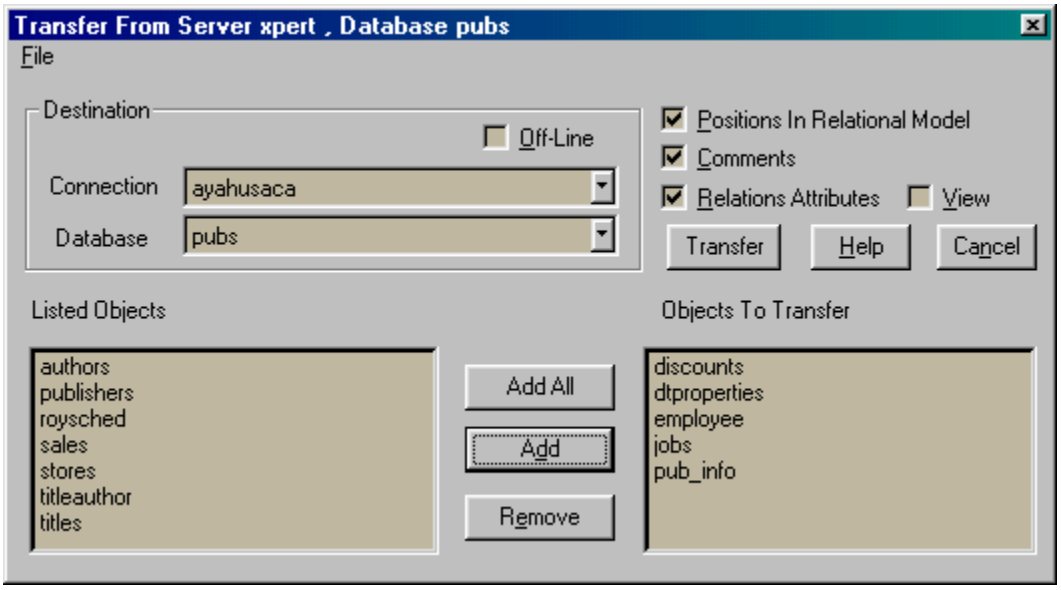
The SQLing database can be transferred both online and offline.

Online transfer of the Metadata Layer (SQLing database)

1. Select Transfer in the Current Database menu.

2. Select Metadata Layer.

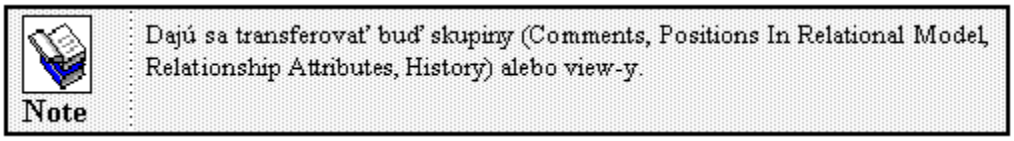
A dialog box entitled Transfer from Server *server_name*, Database *database_name* appears.



3. In the group Destination
   a) In the combo box labeled Connection select the name of the target server into which the SQLing database should be transferred.
   b) In the combo box labeled Database select the name of the target database into which the SQLing database will be transferred.

4. Select the groups (Comments, Positions In Relational Model, Relationship Attributes, History) or Views you wish to transfer (see note below).

5. In the field Listed Objects select the tables (in group Views select the names of views) to be transferred and press Add.
   Press Add All if you want to transfer all objects.
   To prevent the transfer of tables that have been added into the field named Objects to Transfer highlight the object and press Remove.
   Press Transfer.



Offline Export of the Metadata Layer (SQLing database)

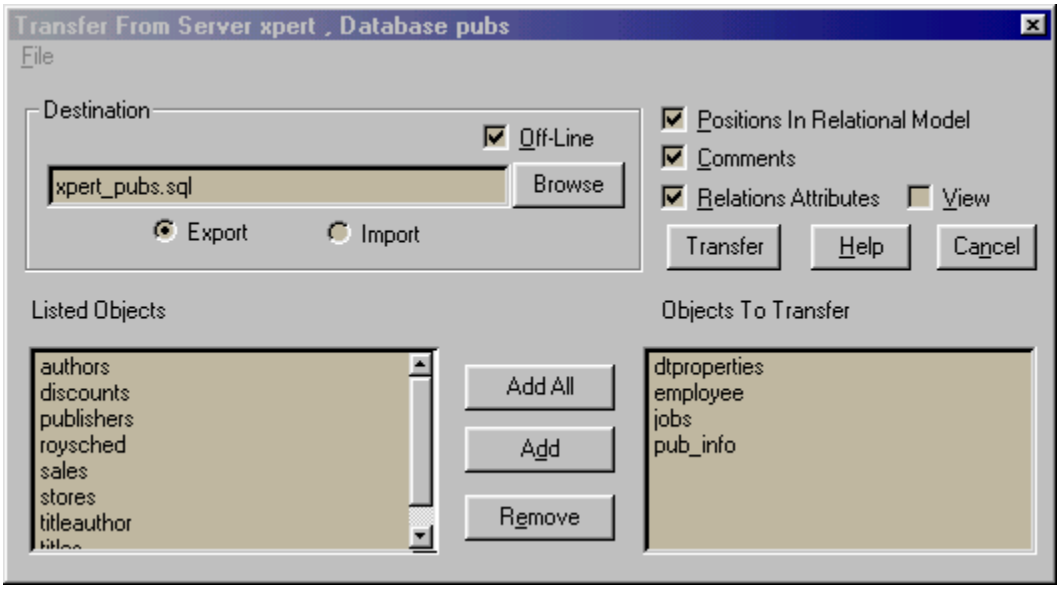
1. Select Transfer in the Current Database menu.

2. Select Metadata Layer.

A dialog box entitled Transfer from Server *server_name*, Database *database_name* appears.

3. Group Destination
   Check offline.

The subwindow is changed.



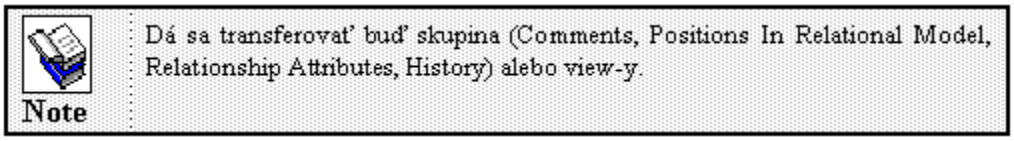
4. In Destination
   Select the Export radio button.
   Press Browse and specify the location where the database will be temporarily

saved.

5. Select the groups (Comments, Positions In Relational Model, Relationship Attributes) or Views you wish to transfer (see note below).

6. In the field Listed Objects select the tables (in group Views select the names of views) to be transferred and press Add.
   Press Add All if you want to transfer all tables (or selected views in the Views group).
   To prevent the transfer of tables (or views selected in the Views group) that have been added into the field named Objects To Transfer highlight the object and press Remove.

7. Press Transfer.



Offline Import of the Metadata Layer (SQLing database)

1. Press Transfer in the Current Database menu.

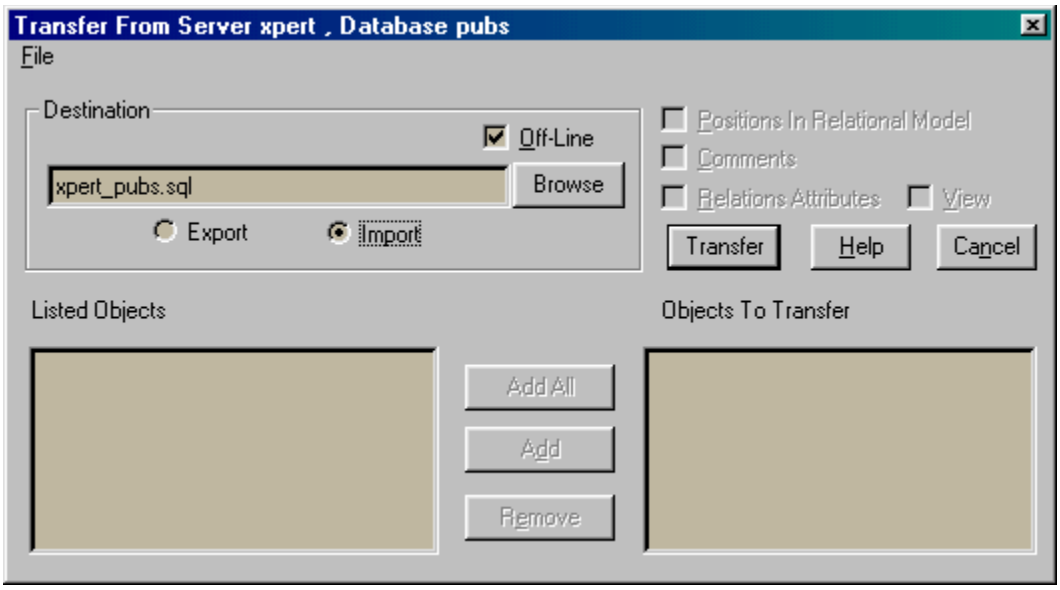
2. Select Metadata Layer.

   A dialog box entitled Transfer from Server *server_name*, Database *database_name* appears.

3. Group Destination
   Check Offline.
   Subwindow changes.

4. In Destination
   Select the Import radio button.

   A dialog box entitled Transfer from Server *server_name*, Database *database_name* appears.



5. Press Browse.
   A dialog box entitled Open appears.

6. Select the database to import.

7. Press Transfer.

Dialog box File menu.

Dialog Box File Menu

The settings you made in a dialog box can be saved to disk and loaded later to save time when using the same dialog box more than once.

Save dialog box settings defined by user

1. Select Save Settings from the dialog box File menu.
A dialog box entitled Save as appears.

2. Type in the file name, select the disk drive and folder where you want to save the file.

3. Press OK.

Restore the settings saved before

1. Select Load Settings from the dialog box File menu.
A dialog box entitled Open appears.

2. Select the file which contains the settings.

3. Press OK.

Merge dialog window settings from several files

1. Select Include Settings from the dialog box File menu.
A dialog box entitled Open appears.

2. Select the file which you want to add to the dialog window settings.

3. Press OK.

Repeat steps 1 to 3 until you have selected all settings that you wanted to add to the dialog box.

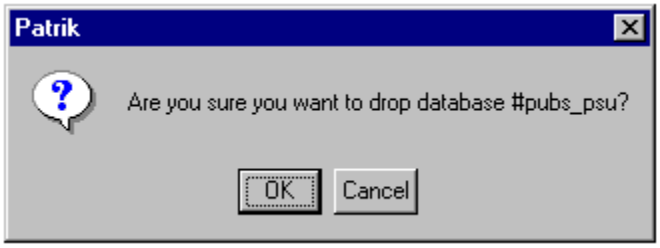
Drop Database

Select Drop Database to delete the whole of the current database.
When a database is dropped all its objects are removed and the space it occupied is released.

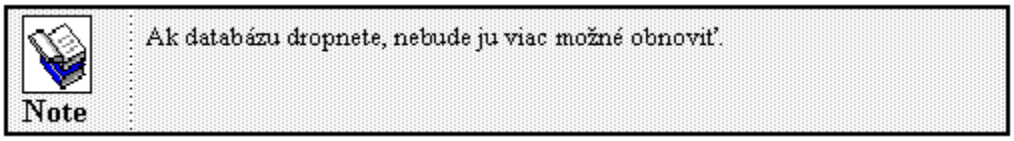
Drop database

1. Select the database you would like to drop in the combo box entitled Database.
Select Drop Database from the Current Database menu.

This dialog window appears.



2. Press OK to confirm you really wish to drop the database.



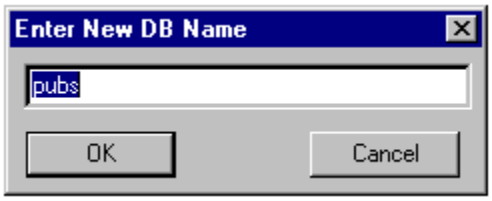
Rename

To change the name of the current database, select Rename.

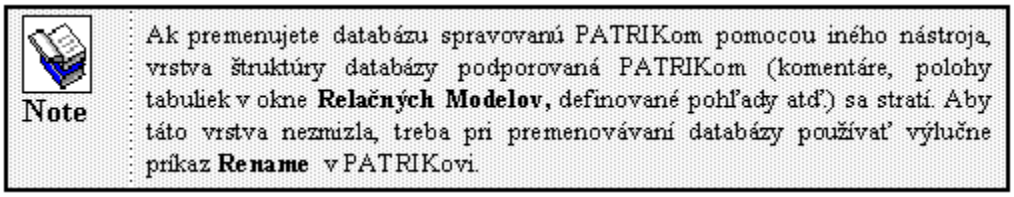
Rename the current database

1. Select the database you would like to rename in the combo box entitled Database.
2. Select Rename in the Current Database menu.

A dialog box entitled Enter New DB Name appears.



3. Type in the new name of the database.
4. Press OK.



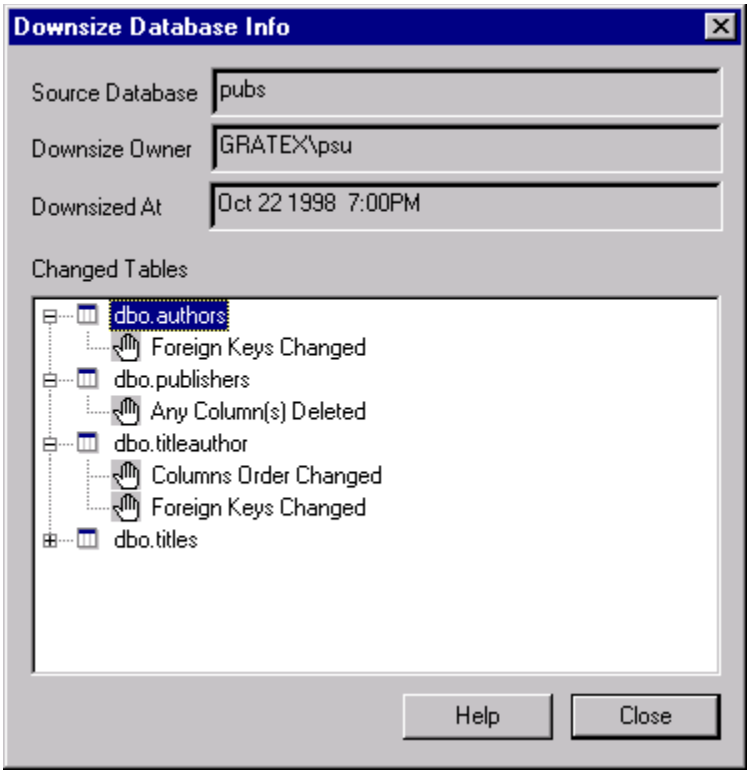
Database Info

Select Info to view information about the current database.

Viewing current database information

Select Info in the Current Database menu.

A window entitled Database Info appears. This window contains information about the database specific to the server version.
Database info for a downsized database is displayed in the following dialog box



which contains this information:

- the database which was downsized
- who and when executed the downsize
- which tables were changed and how

Make Document

Select Make Document to create a text document containing information about the tables of the current database. This function is useful in creating printed as well as electronic documentation.

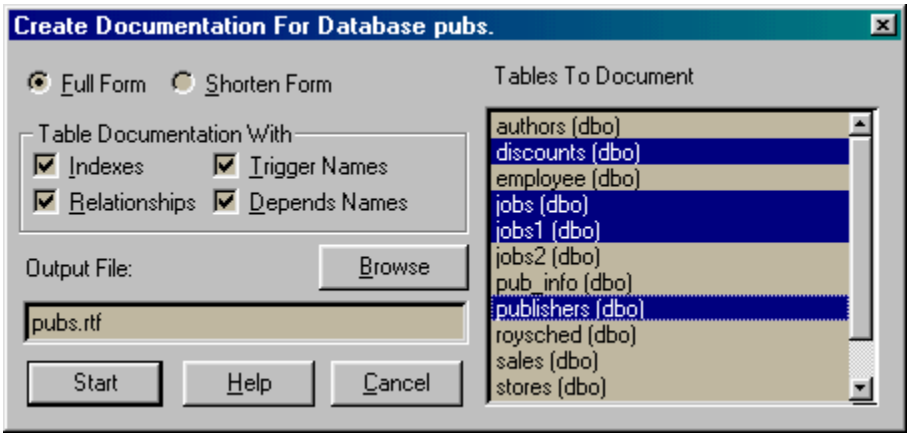
The created document is in the rich text format (file_name.rtf). It can be viewed by any text editing application which supports this format (e.g. MS Word, Wordpad).

Create Database Documentation

1. Select the database which you wish to document in the combo box entitled Database.

2. Select Make Document in the Current Database menu.

A dialog box entitled Create Documentation for Database *database_name* appears.



3. Select the items you wish to be documented in group Table Documentation with.

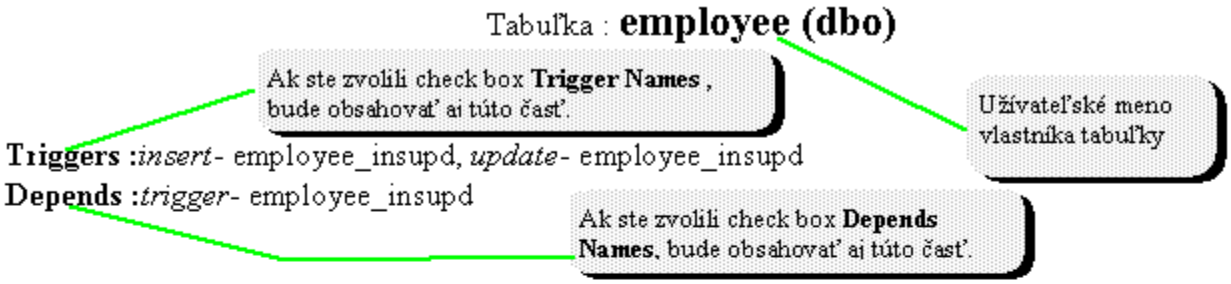
4. Select the tables you wish to be documented in group Tables to Document.

5. Enter the name and location of the document in the field entitled Output File. SQLING will automatically suggest a location.
   If you want to change the default location enter the path and the name of the document or press Browse. SQLING will display a dialog box entitled Choose Directory where you can select the output file name and folder.

6. Use the radio buttons Full Form, Shortened Form to determine the type of documentation to be generated. Standard (Full Form) is shown on the figure below, simplified (Shortened Form) consists of one table with the columns: Database, Table, Column Name, Column Data Type, Column Comment describing all columns of the tables selected in section 4.

7. Press Start.

Príklad dokumentovanej tabuľky zamestnancov:

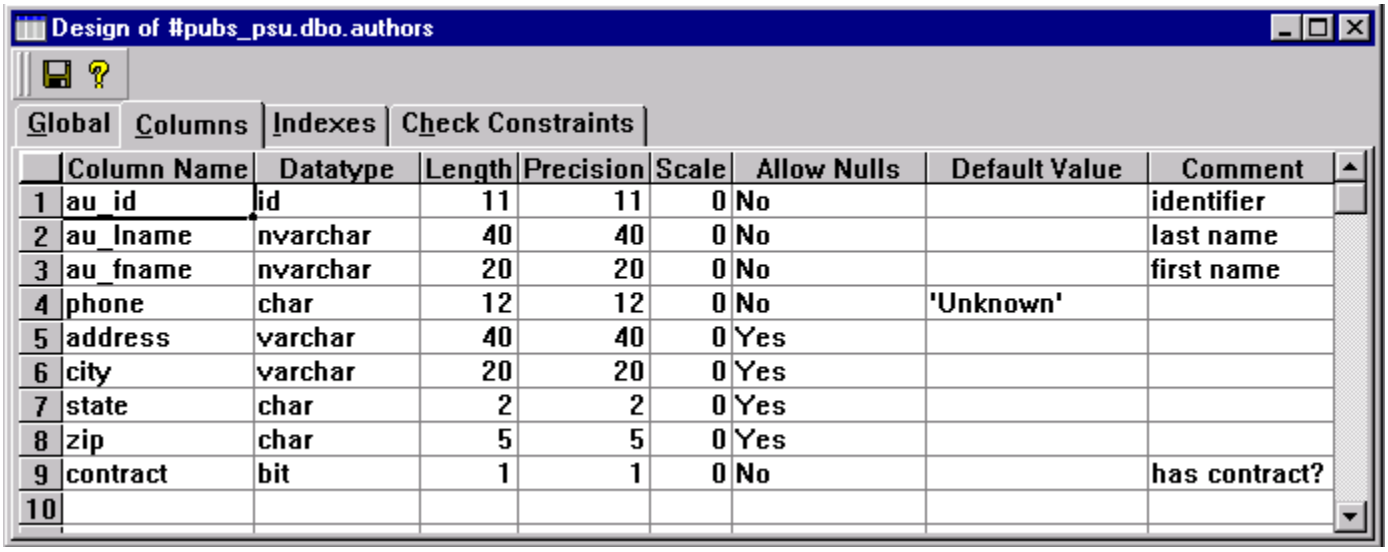


Column name	Type	Rule	Default	Comment
emp_id	empid			employee identifier
fname	varchar(20)			employee first name
minit	char(1) NULL			
lname	varchar(30)			employee latst name
job_id	smallint		DF__employee__job_id__2FAF1EF9	employee job
job_lvl	tinyint		DF__employee__job_lv__3197676B	employee job level
pub_id	char(4)		DF__employee__pub_id__328B8BA4	id of publisher where employee works
hire_date	datetime		DF__employee__hire_d__3473D416	the date when employee was hire

Displaying and Editing Table Structure

Designer or Viewer?

Window titles are the first thing to indicate which operations can be done on the structure displayed in the window. If the window contains the structure of a table from an actual database, the title will say View of *<table_name>* and although you are able to edit the table structure, it cannot be stored (applied). Only column notes or modifications to table description can be saved.
In the rest of the cases, when the window contains the structure of a table of a downsized database or when the structure (still empty) belongs to a newly created table in a downsized database the window title says Design of *<table_name>* and you are able to save any changes made to the table structure.

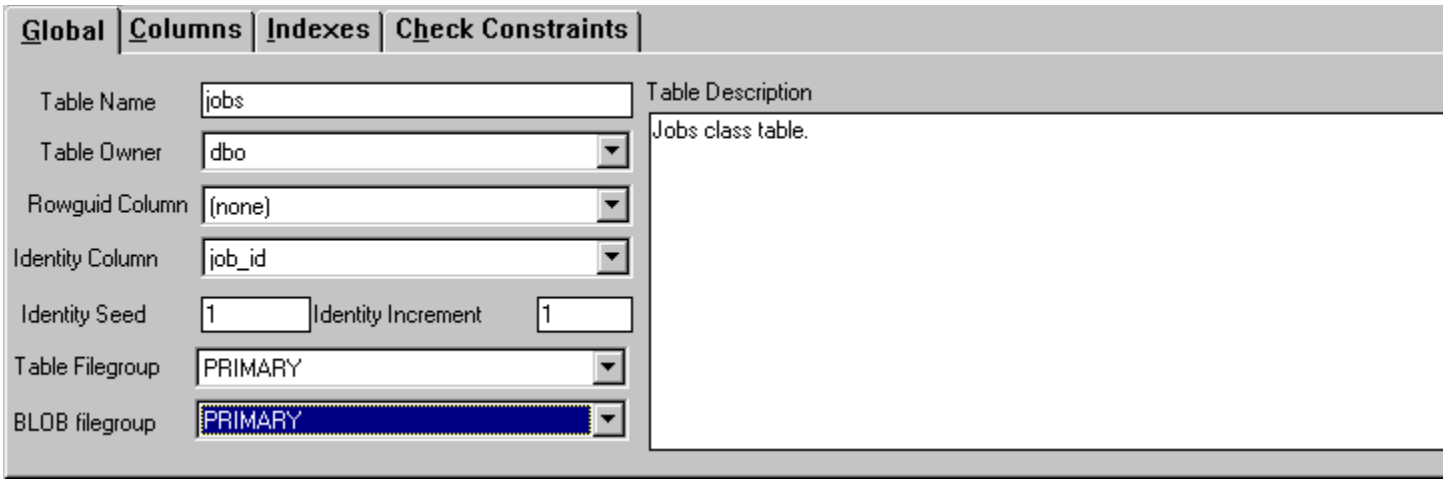


	Column Name	Datatype	Length	Precision	Scale	Allow Nulls	Default Value	Comment
1	au_id	id	11	11	0	No		identifier
2	au_lname	nvarchar	40	40	0	No		last name
3	au_fname	nvarchar	20	20	0	No		first name
4	phone	char	12	12	0	No	'Unknown'	
5	address	varchar	40	40	0	Yes		
6	city	varchar	20	20	0	Yes		
7	state	char	2	2	0	Yes		
8	zip	char	5	5	0	Yes		
9	contract	bit	1	1	0	No		has contract?
10								

Window Layout

The window contains a Save button and the structure description is broken down into four tabs.

The Global Tab



this tab contains general table properties, including the table name, owner, description, segments/filegroups where the table is located and its text/image data. Furthermore this tab indicates whether the table has a RowGuidCol or an Identity column (and if the Identity column exists whether it is seed or increment).
The tab contains original names of table properties and some of the fields may be grayed. This depends on the SQL server where the table is located.

The Columns Tab

This tab contains the actual description of the table structure - the columns.

Table Properties

The table supports most of the standard table properties including:

- clipboard functionality
- widen/shrink rows/columns
- cell replication
- edit cell values in edit boxes/combo boxes (by pressing F2 or double clicking on the cell)

but it does not support sorting by column - by double clicking on the column title.

These properties are described in the section Table with SQL Table Data.

Moreover the following additional features are available:

Move Table Columns

The order of rows in a GDI table can be changed using the drag & drop technique to move the selected rows to a new place.

Data Type Naming Convention

When editing the Column Name column of a new row (i.e. when adding the name of a new column into the table) the column named Datatype is automatically filled according to the following:

attribute name prefix	data type

s_	char, varchar
nl_	int
*ni_*	*smallint*
*nb_*	*tinyint*
d_	datetime
nd_	decimal, float, number
c_	money
ts_	timestamp
bn_	binary, varbinary
m_	text, image
b_	bit

If you do not wish to use this naming convention you can change the newly added data type.

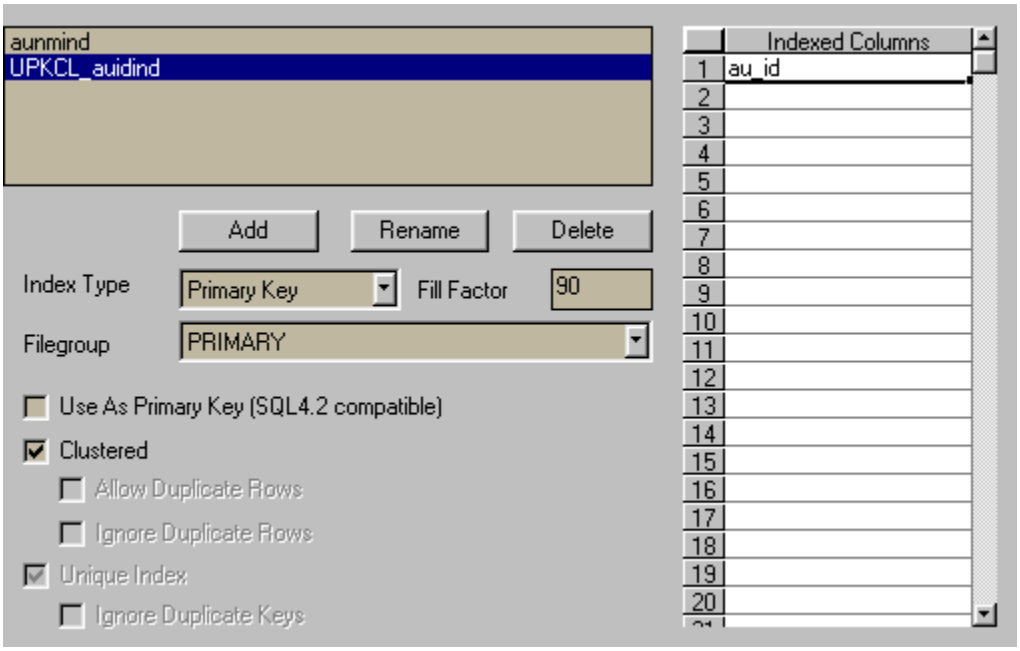
Length, Precision, Scale

The Length parameter only has to be provided for the data types [var]char and [var]binary. The parameters Precision and Scale are optional. If any parameters required by the data type used were not provided, default values are used.

Default Value

Either enter the default value to create a default constraint or bind an existing default to the column by selecting the default name in the combo box.

The Indexes Tab



add/change/delete indexes, and unique and primary key constraints. The only thing that could be unclear here is the checkbox named Use as Primary Key which can be

selected for a Unique Index, Unique constraint or PrimaryKey constraint. The primary key is compatible with SQL4.2 and SQLing requires this key when connecting the table to other (foreign) tables by relations with a definable referential integrity which is ensured in triggers.
Tip: If you want to avoid deciding when to check this check box, select it only for Primary Key Constraints.

The Check Constraints Tab

used to create/change/delete table check constraints.

Saving changes to table structure/comment

Saving changes to structure or to comments?

The title of the window indicates what can be done with the displayed structure. If the structure belongs to a table of an actual database the window title says View of <*table_name*> and the table structure can be edited but not saved (applied). You can only save the changed column comments or the table description.
In all other cases, i.e. when the structure belongs to a table of a downsized database or when the structure (still empty) belongs to a newly created table in a downsized database the window title says Design of <*table_name*> and any changes made to the table structure can be saved.

The correctness of changes to the table structure and consistency issues

When saving modifications to the structure SQLing checks that the changes are correct and consistent. If the application discovers the changes made are incorrect, one of the following error messages is displayed, the saving process is halted and you will be enabled to correct any errors in the structure. These are some of the possible error messages:

Column <column name> Does Not Match Identifier Rules
Table cannot have more than one Primary key constraint.
Constraint expression for constraint <constraint name> is not defined
You want use as primary key more than one index?!?
Index <index name> has no column set
SQL server can handle max. 1 clustered index
and others.

If no errors were discovered the consistency of the changes is checked. The user can decide whether to have consistency checked automatically or whether to skip it, risking problems during upsizing.
Here are some examples of possible consistency issues:

Q: One or more columns are set to allow nulls but they participate in primary key <PK constraint name>, which is impossible.
Set these columns as Not NULL?
A: both Yes and No are possible but if you select No you risk that during upsize neither the table nor its primary key will be created.

Q: The table contains at least one column disallowing NULL-s with no default bound on it. Upsize of this table may result in errors.
A: The issue may arise if a column was added which neither has a default value nor has NULLs permitted, so the SQL server will not know what to enter into this column (on the existing rows). By selecting Cancel you risk that the table data will not be placed in the proper table after upsize, but into the table named upsizetmp_<xxx>.

Q: I must erase all [one to many] relationships (<their number>) with this table.

A: This issue arises when the primary key (or its part) is revoked. If you press Cancel the saving process will be interrupted and you will return to the designer.

Check constraints are further parsed and in case of failure the function closes and returns to the designer.

The table is stored after all checks are finished.

Creating a New Database Table

The Design Table window is displayed without any attributes.

Enter the necessary information and close the window. The table will be stored into the database.

Installing SQLing

In this chapter you will learn about the system requirements of SQLING , how to tall or uninstall SQLING successfully and what to do when you are connecting to an SQL server for the first time.

System Requirements
Uninstalling SQLing
Creating Database Devices
Connecting to an SQL Server and Creating the SQLing Database

System Requirements

This is the minimum hardware and software configuration required by SQLING:

- Windows NT 4.0, MS Windows 95, 98 OS
- 16 MB RAM
- MS-SQL Client
- connection to an MS SQL Server
- 10 MB free disk space on the workstation
- 20 MB free disk space on every SQL Server with which SQLing will cooperate

Scripts Provided with the Application.

The install folder of the application contains several T-SQL scripts (.sql) written to help in specific situations.

1. CompactMetaDB.sql - this script repairs the SQLing database if it is not compact. This situation may occur when databases are renamed and deleted other than from SQLing, when the SQLing database is transferred frequently or, indeed, by long use of the application. It is advisable to schedule automatic execution of the script with a period of, for instance, one year.
2. DropDatabase.sql - this program drops an incorrectly downsized database not included among the databases which can be browsed using SQLing. When this script is executed for the first time a procedure named DropDatabase is created. Run this procedure with the name of the downsized database as the parameter to drop the failed downsize.

Uninstalling SQLing

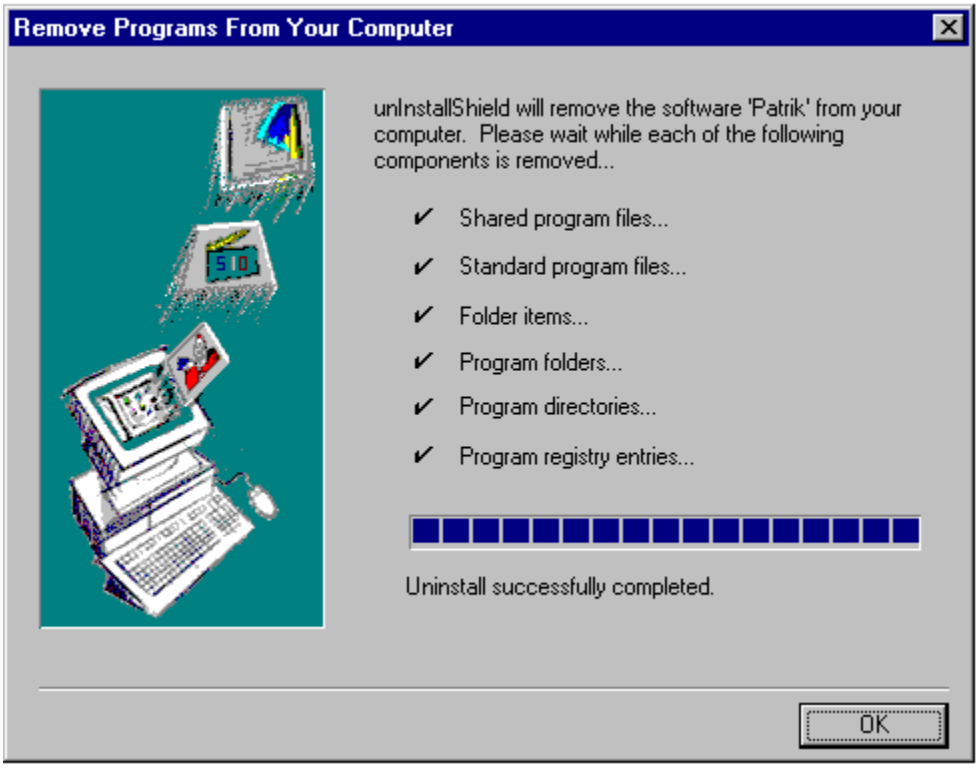
If you wish to remove SQLING entirely from a workstation hard drive:

1. Double click on the UninstallShield icon in the SQLING program group.

A dialog box entitled Confirm File Deletion appears.

2. Press Yes to uninstall SQLING.

A dialog box entitled Remove Programs >From Your Computer appears.



1. Press OK.

Creating Database Devices

Note: Only if connecting to SQL Servers 6.5 and lower. No devices have to be created on SQL Server 7.0.

Before you connect to an SQL Server for the first time, two database devices must exist on the server. One of them - SQLing_log - servers as a transactional database log and the second - SQLing_dat - stores data. Both devices should have about 10 MB.

Creating Database Devices

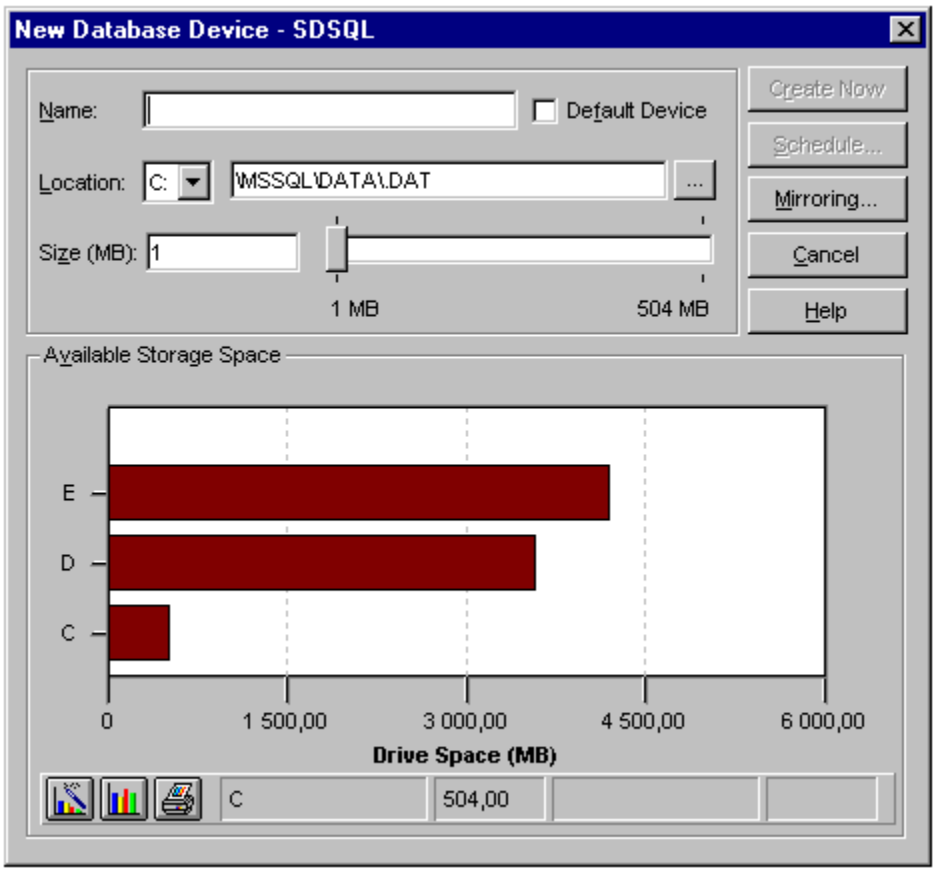
1. Open the Microsoft SQL Enterprise Manager application.

2. Select the SQL Server to which you would like to connect using SQLING.

3. Click on the folder Database Devices.

4. Click the right mouse button.
The Shortcut menu appears.

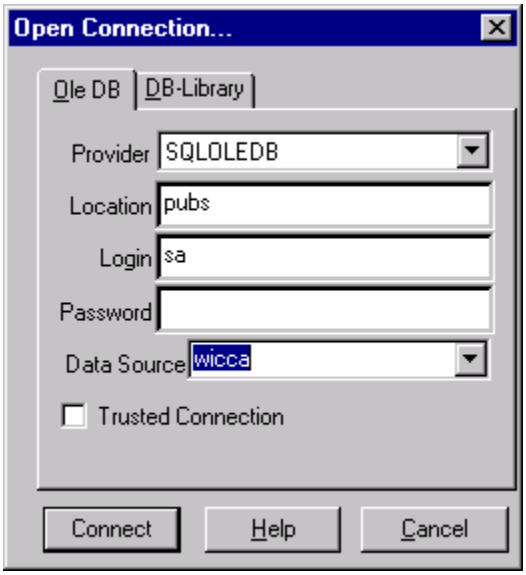
5. Select New Device.
New Database Device - *server_name* appears



6. Enter the device name in the field named Name.
7. Enter the device size in the field named Size (MB).
1. Press Create Now.

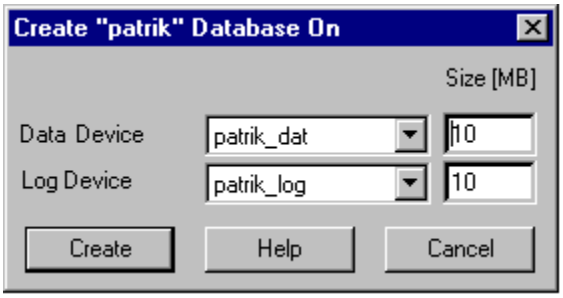
Connecting to an SQL Server and Creating the SQLing Database

Double click on the SQLing icon to run the SQLING application. SQLING opens the main application window and a dialog box entitled Connect.



Enter information about your connection and press Connect.

If you are connecting to an SQL Server for the first time, SQLING will attempt to create its own database named SQLing on the server.
If you are connecting to the SQL Server 7.0 the **SQLing** database will be created automatically. If you are connecting to SS 6.5 or lower a dialog box named Create SQLing Database On is displayed, awaiting the names (and sizes) of two devices: Data Device and Log Device. The SQLing database will be created on these devices.



1. Select the SQLing_dat device in the combo box named Database Device.
2. Select the SQLing_log device in the combo box named Log Device.
3. Press Create.

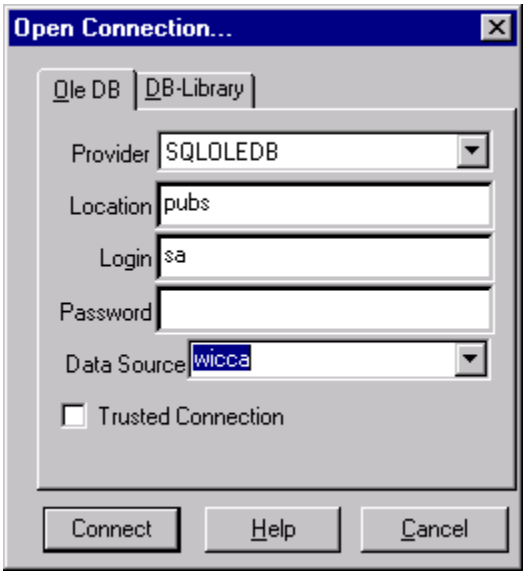
Project Menu

This chapter describes the commands of the Project and the main menus.

Connect - connection to a (new) SQL server
Disconnect - disconnect from the current SQL Server
Query - SQL Query Editor
Options
Exit

Connect

This dialog box is used to initiate connection to an SQL Server via either the DB Library or OLE DB interface. The dialog box contains two tabs named DB Library and OLE DB. The former, DB Lib, was the most efficient interface for communicating with the MS SQL Server until MS SQL 7.0 was introduced. Microsoft then decided to stop supporting this interface and version 7.0 of the MS SQL Server only supports DB-Lib, compatible with version 6.5. New functions of the MS SQL Server, including the support of long varchars, Unicode, etc. are not handled by the interface. Due to this obvious shortcoming, SQLing was extended to provide connection via OLE DB which is not limited to the SQL Server and can communicate with any DBMS, or anything else for which an OLE DB provider exists. This enables SQLing to browse any database systems (e.g. Oracle, Db2, MSAccess, TextFiles etc.). SQLing is primarily developed to cooperate with the SQL Server, therefore the other DBMS's can only be browsed at present. SQLing will be scaled up to other DBMS's according to market demand and the development of OLE DB which still suffers from certain problems.

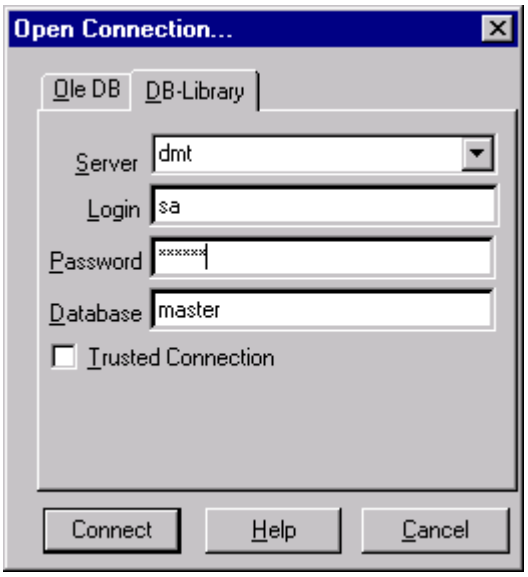


It is recommended to use OLE DB when connecting to MSSQL 7.0 and other DBMS's. The OLE DB Connect has the following fields:

- Provider - OLE DB Provider.
- Location - usually the database name.
- Login - user account name to use in the connection.
- Password - password of the specified Login (account).
- Data Source - most often the name of a server or other source of data.
- Trusted Connection - Windows user authentication.

0 Technical Note: If the provider is MSDASQL and the field Location is not empty SQLing does not send connection information to the provider in the standard way but by means of an ODBC Provider String, as follows:

1 *DRIVER={<Data Source>};SERVER=<location>;UID=<login>;PWD=<Password>;*

2 

3 When connecting to MSSQL 6.5, 6.0, 4.21 it is recommended to use the DB-Library. The connect dialog contains these fields:

- Server - name or IP address of the SQL Server to which SQLing is connecting. The field contains a history of all servers to which the application connected successfully from the present workstation.
- Login - Login name on the SQL server.
- Password - Login Password
- Database - name of the database to be opened by SQLing when connected. If this field is left blank (or the database name is invalid) the application opens the default database for the current Login.
- Trusted Connection - if mixed security is enabled on the current SQL server check this checkbox to specify whether integrated security (checkbox checked) or standard security should be applied.

If you are connecting to an SQL server for the first time, go to the section First connection to the SQL server and setting up the SQLing database.

Possible Problems with OLE DB

At the time when this document was written the following problems in OLE DB were recognized and are handled in SQLing:
- When connecting to SQL 6.5 and lower no information about the database scheme could be retrieved unless a script supplied with OLE DB was run on the SQL Server. The script creates a set of scheme rowset procedures on the server.
- OLE DB does not support data in tables containing more than one text/image column.

Disconnect

Select this command to close the active connection. SQLing will display a warning if there are open transactions in the active connection. By disconnecting, all transactions will be cancelled (Rollback).

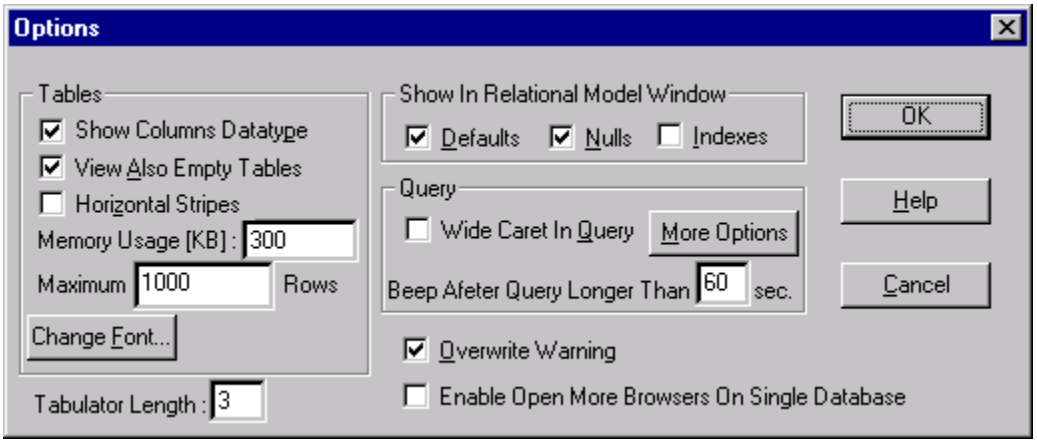
Options

This section describes the setting of SQLING options.

Setting the Options table in SQLING

1. Press Options .

2. The Dialog Box entitled Options appears.



1. Tables group
   a) Check the check box Show Column Datatype if you want SQLING to display column data types in data window headers and in query results.
   b) Check the check box View Also Empty Tables if you want SQLING to open even data windows containing no records.
   c) Horizontal Stripes - set to display even-numbered records in data tables with darker background.
   d) Memory Usage, Maximum .... rows - amount of memory in kilobytes or the maximum number of lines to which the sizes of loaded data tables and query results are truncated.
   e) Change Font - opens a dialog box where you can set the font used in the data tables.

2. Select check boxes in the group Show In Relationship Window to specify which table column properties you wish to view in the tables displayed in the Relational Model window.

3. Query group
   a) When the Wide Caret in Query check box is checked, SQLING displays a wide caret in the Query Editor window.
   b) Enter a time period in seconds in the field Beep after query longer than and SQLING will beep when an executed query lasts longer than the specified time.

c) More Options - general query options.

4. Use the field Tab Length to set the number of spaces, which will represent a tab in the Query Editor window.

5. If you select the check box Overwrite Warning, SQLING will display a warning message whenever you attempt to overwrite an existing file while saving to disk.

6. Enable Open More Browsers on Single Database - if not checked, SQLING will not allow you to open more than one Browser window for a single database.

7. Press OK to confirm your settings.

Options are saved to disk automatically so you do not have to set them next time you run SQLing.

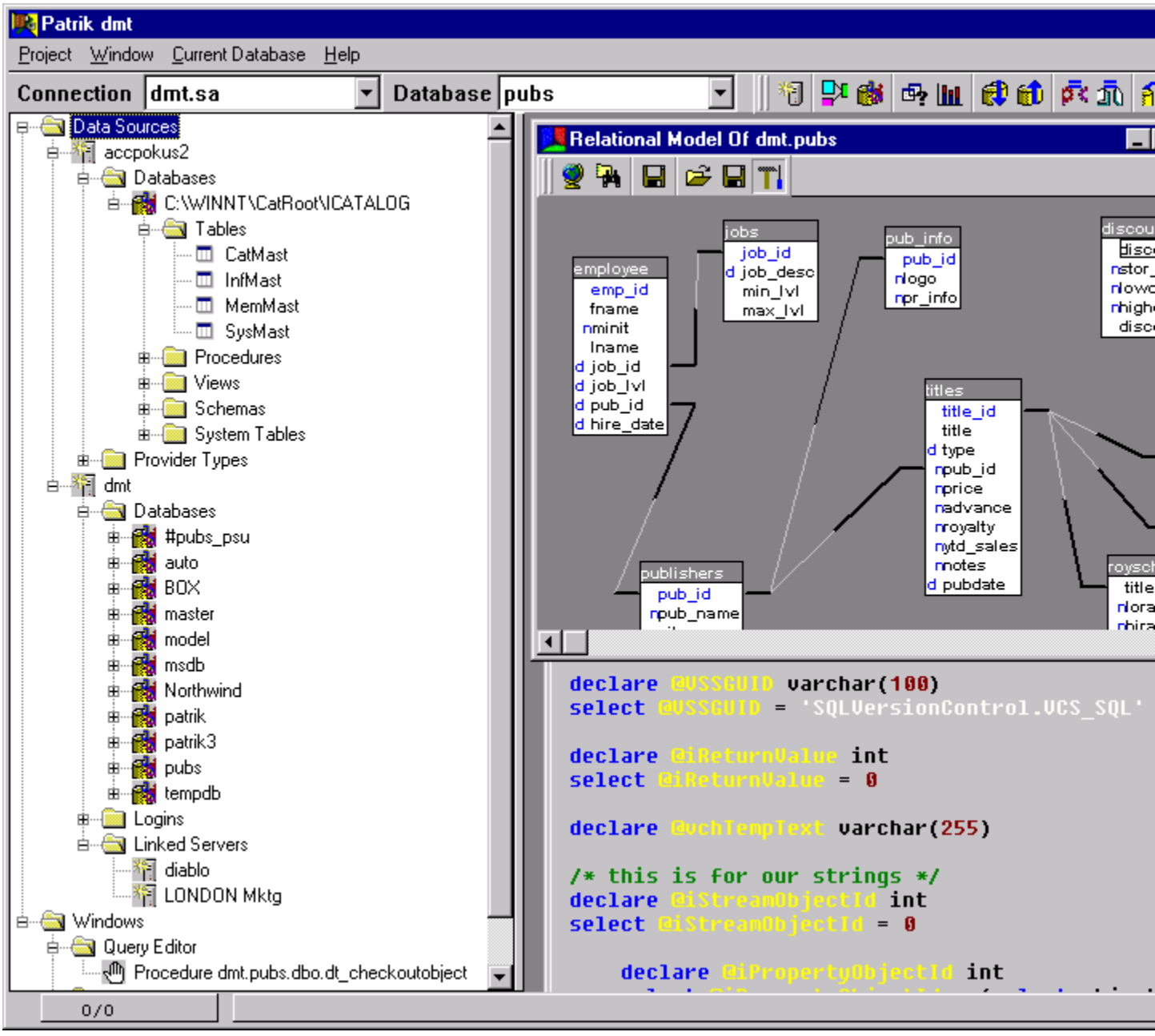
Exit

To exit SQLING select Exit in the Project menu. SQLing will display a warning if there are open transactions in any of the active SQL connections. By disconnecting, all transactions will be cancelled (Rollback).

Main Application Window

This section describes the items that appear on the screen and provides basic information about the items in the application menu of SQLING. It also explains the elements present in the Main Application Window.

The Main Window appears when the application is launched. Unless the name of an SQL Server was specified on the Command Line-e, a dialog box entitled Connect is automatically displayed so that you can (and must) choose the server to which SQLing will connect.



The working area of the window consists of the following elements

- Window title - contains the application name and the name of the server to which SQLing is connected.
- Main Application Menu
- Toolbar - contains the Connection Combo Box, Database Combo Box and main application buttons
- Tree Database Browser
- Info bar - located at the bottom of the window, contains
    - Numeric information - the meaning changes according to the window of the application user interface which is active at the moment
    - Text information - application messages
    - Time

0 Important keyboard shortcuts

1 The following key shortcuts are important to users accustomed to frequent use of the keyboard

2 Alt F1 and Alt F2 switch the focus of the cursor between the tree browser and the main working area of the application (the MDI window).

3 Alt Shift F2 minimizes the browser desktop. The browser is restored automatically when invoked by Alt F1.

4 Toolbar buttons

The Main Application Window contains buttons for the most frequently used commands and functions:

-  Connect… - connects to a (another) SQL Server.
-  Relational Model- opens the window with a relational model of the current database.
-  Browser - opens the database browser
-  Query Editor - opens an SQL script editor
-  Table sysprocesses - maps process activity on the server
-  Downsize - prepare database for modifications
-  Upsize - implement database modifications
-  Compare Schema - compare the structure of two databases
-  Compare Data - compare the data of the selected tables
-  Online Transfer - transfer (a part) of a database into another database
-  Offline Transfer - transfer (a part) of a database to disk
-  Options - change application settings

Main Menu

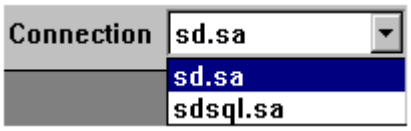
The Main menu in the Main Application Window contains the following items:

0 Project - general application control functions
1 Windows - functions for ordering windows on the application desktop
2 Current Database - operations on the Current Database
3 Help - contains functions Help and About...

Connection Combo Box

0 Use the Connection combo box to select the SQL Server you want to work with from a list of SQL servers to which SQLING is connected. The records in the Connection combo box have the format *server_name.user_name*.

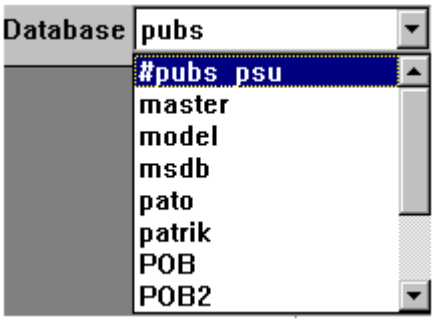
1 To add a new SQL server into this combo box select Connect… in the Project menu and establish a connection to a new SQL server.

2 

Database Combo Box

All databases present on the SQL Server you are currently connected to are listed in the Database combo box. The database which is highlighted in this combo box is considered to be the current database. All commands and functions from the Current Database menu relate to this database.

0 To make a database current, select its name in the Database combo box.

1 

2 Some database names begin with a double cross (**#**). These are working databases (created by the Downsize function) which are not located on the SQL server. Working databases are defined in the SQLing database.

3 The content of the combo box is refreshed whenever the same server name is selected in the Connection combo box.

SQLING enables a number of commands and functions to be selected from the Shortcut menu. These commands are always related to the objects you are working with. To access the Shortcut menu for a particular window or form field, place the mouse cursor over its area and press the right mouse button.

The Sysprocesses Table

Contains the records of the system table named sysprocesses. This table has a record for each of the processes currently running on the server. Its Shortcut Menu contains the following functions:

- Related Table - displays relation records
- Order By - orders records by the selected column
- Edit WHERE - defines a WHERE condition for the table records
- Refresh - reloads the table
- Show Input Buffer - shows the last command executed by the selected process
- Send Message To Owners - transmits a net send to the owners of the selected processes. Enter the text of the message in a dialog box that appears. The text may contain a %s string which SQLing replaces when sending the message by the name of the application that opened the selected process on the server.
- Autorefresh, Stop Autorefresh - start/stop automatically refreshing window content at the specified interval.

Autorefresh

Use this dialog box to set the refresh interval (in seconds) for the table data window. Turn the autorefresh function off by selecting **Stop Autorefresh** in the popup menu of the sysprocesses table window.

Tip: Those processes you wish to monitor can be highlighted and stay highlighted even after a data refresh.

System Table Data

The window displays the records stored in the selected system table. The window's Shortcut Menu contains the following functions:

- Related Table - displays relation records.
- Order By - orders the records by the selected column.
- Edit WHERE - defines a WHERE condition for table records.
- Refresh - reloads records from the table.

Query Editor

The Query Editor is used for editing (writing, viewing, creating, executing) of Free Queries. Procedures, views, triggers, defaults and rules are edited a dedicated designer which takes into consideration object structure and enables greater comfort of work than if the object (e.g. a procedure) was edited in the Query Editor. A free query, however, can also be a command (set of commands) creating a procedure.
Press the toolbar button or select Query in the Project menu to open the Query Editor.
The title bar of the window informs that a query is edited, shows the name under which the query is saved on the disk (or unnamed for new queries) and informs whether the query was edited since it was last saved (an asterisk * appears before query name to show that it was edited). The toolbar of the Query Editor contains the following commands:

- Load... - loads the content of the selected file into the editor.
- Save - saves the Query to file.
- Print... - prints text.
- Find - Searches for information (Alt F3)
- Find Next - finds the next occurrence of the selected expression (F3).
- Properties
- Execute - executes the Query or its highlighted part (Ctrl E).
- Go - launches the query debugger (F5).
- Insert/Remove Breakpoint - inserts/deletes a breakpoint at the line with the cursor in the editor (F9).
- Remove All Breakpoints - removes all set breakpoints.

0 The space under the toolbar is divided into tabs. When the window is first opened only the tab Query which represents the Editor is displayed. A tab entitled Results appears when the query is first executed using the function Execute. This tab lists the results and any error messages returned by the SQL server during the last execution of the query. A tab entitled Show Plan is added if Show Plan is selected in Options. This tab contains query optimization information.
1 Press Alt + <the underlined letter in tab name> to switch between tabs.

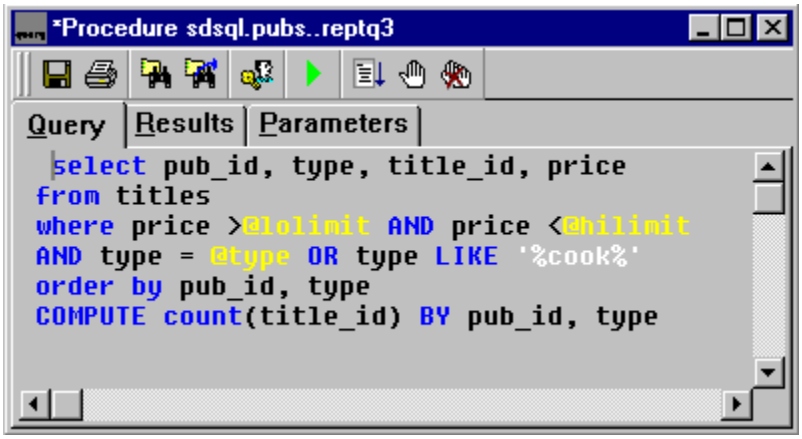
2 To access the functions which are not available in the toolbar, press the right mouse button in the editor to display the following shortcut menu:

- Save as... - saves the Query in a different file.
- Find Prev - finds the previous occurrence of the selected expression (Shift F3)
- Replace - replaces text. (Ctrl H)
- Colors... - sets editor colors for different word groups.
- Define Vocabulary - define a user word group which will be shown using a different color.
- Run to Begin - runs the debugger and halts before the first query command.

- Run to Cursor - runs the debugger and executes the query until a breakpoint or the cursor position is reached.

Procedure Designer

Use the Designer to view, edit and create stored procedures, views, triggers, defaults and rules. The Procedure Designer does not differ visually from the Query Editor but it has additional functions. Its editor window does not contain the procedure initialization text (CREATE PROCEDURE) nor parameter definition. The parameters of a procedure which is edited in the Designer can be specified in a tab entitled Parameters. This tab shows the table where procedure parameters and their properties are stored and is not shown when Defaults, Rules, or Triggers are edited. The Parameters tab is displayed while editing Views but it is used to list and edit column names instead of parameters.



The Designer title bar shows object name, type and location. If the object (procedure) was modified in the designer since it was last saved into the database, an asterisk (*) appears in front of its name.
The Designer toolbar contains the following commands:

- Save - scripts the procedure (or any other edited object) into a file.
- Print... - prints text.
- Find - The find function (Alt F3)
- Find Next - finds the next occurrence of the selected expression (F3).
- Properties - sets additional properties of Procedures, Views, Triggers, Defaults, or Rules
- Execute - executes and/or saves an object back into the database. The function works differently for Procedures , Views and for Triggers, Defaults and Rules (Ctrl E).

The following buttons are only shown while a procedure is being edited:

- Go - runs procedure debugger (F5).
- Insert/Remove Breakpoint - inserts/removes a breakpoint at the line with the cursor in the editor window (F9).
- Remove All Breakpoints - removes all set breakpoints.

0 The space below the toolbar is divided into the following tabs, some of which are optional:

- Query - this tab is the editor window containing the source code of the procedure (Alt q).
- Results - only opens after the procedure is executed or saved; contains the results/status of the execution/saving (Alt r).
- Parameters - contains the definition of procedure parameters or the column names of a View (Alt p).

0 Press Alt + <the underlined letter in tab name> to switch between tabs.

1 To access the functions which are not available in the toolbar, press the right mouse button in the editor to display the following shortcut menu:

- Save as... - scripts the procedure into a different file.
- Find Prev - finds the previous occurrence of the selected expression (Shift F3).
- Replace - replaces text.
- Colors... - sets editor colors for word groups.

0 The following items are only included in the shortcut menu while a procedure is edited.

- Run to Begin - runs the debugger and halts before the first command in the procedure.
- Run to Cursor - runs the debugger and executes the query until a breakpoint or the cursor position is reached.

0 Sometimes the menu contains a submenu Dependencies with the items Called by <procedure name> and Calls <procedure name>. These functions open the procedure which is called by or calls the currently edited procedure.

Editor

Use the editor to type commands or procedure code, which will be sent to the SQL server when executed. The editor automatically analyzes the typed words and expressions and divides them into syntactical word groups, which are displayed using different colors. The following word groups are recognized in the T-SQL mode:

- Keywords (e.g. SELECT, from, Where - letter case is not important)
- Strings (delimited by quotation marks or inverted commas)
- Numbers
- Variables and temporary tables (e.g. @@rowcount, #tmp)
- Comments
- User-defined word groups
- Other words - those that are not included in any of the above groups
- Identifiers delimited by square brackets

0 The following syntactical word groups are recognized when the editor is used to edit HTML (e.g. if the editor is included in a data table and is used to edit an HTML text field - press F3 to launch the editor; HTML text editing is not available when the editor operates under the query editor):

- HTML Element Name
- HTML Attribute Name
- HTML Attribute Value
- HTML Comment
- HTML Tag Delimiter

0 Color text
1 Select Colors from the popup menu of the Query editor to set the colors for the various groups of words.
2 The Write-ahead function for keywords may be helpful when you are typing. If you want to type SELECT, type the first few characters (e.g. SE) only and press Esc. SQLing will complete the remaining letters of the word. If the word was not completed correctly, continue pressing Esc until the desired word appears (SELECT in this case).

3 Online Help
4 Online Help works for keywords, tables and procedures. Position the cursor over the word for which you wish to obtain help and press F1. The editor will decide whether the word is the name of a table, of a procedure or a keyword and displays the appropriate shortcut menu. If the word is a table (even tables referred to as *pubs..authors* are recognized by the help function) the displayed menu contains the items View Table pubs..authors, Open Table pubs..authors and WHERE Open pubs..authors. The menu displayed over a procedure contains the item Design Procedure <procedure name>. In both cases you can select the item or press Esc to close the menu. The situation is more complicated when you want to see help for keywords, which may not always consist of single words and often do not constitute whole commands on their own (e.g.

the word CREATE itself does not have any meaning but the pairs *CREATE TABLE, CREATE DATABASE, CREATE RULE* are actual commands). This ambiguity is removed by displaying all possible topics which begin with the selected word (or part of word) so that the user can decide what to view help on. The help topics are simple, without pictures or numerous examples. The help text (and all help topics) are stored in a table named helpsql which is located on any master SQL Server. SQLing ”only” displays this help.

5 Jump to coupled bracket (quotation mark, comment marker)
6 Use the key shortcut Ctrl { to find the other paired bracket. The function works on all types of brackets - ( ), {}, [] but also on quotation marks, opening and closing comment markers (/*, */) and the keywords BEGIN and END.

7 Write Ahead
8 This function helps to type keywords faster. Type the first few letters of a keyword, then press Esc and the rest of the word is added automatically. Press Esc more than once to cycle through words beginning with the typed letters.

9 Indent
10 Highlight several lines of text and press the Tab key to indent the left margin of the text to the next tab position, press Shift+Tab to move the left margin to the left.

11 Auto Text
12 If you need to type sections of the code repeatedly use the Auto Text function. Highlighted sections can be saved to file (e.g. Ctrl Shift 1) and pasted into the edited text at a later time (Ctrl 1). You can copy and paste more than one script (procedure) segment at a time, using the key combinations Ctrl Shift 2 and Ctrl 2, 3, 4, …., 9, 0 which function similarly to Ctrl Shift 1 and Ctrl 1.

13 Markers in the Query Editor
14 If, while editing a long procedure, you need to ”jump” elsewhere in the code and then return to the place you were editing, use the following key combinations:

- Ctrl F2 – set a marker on the current line
- Ctrl F2 - remove a marker from the current line
- F2 – jump to the nearest marker below (if no more markers lower in the text can be found, jump to the farthest marker above)

Markers are displayed as cyan asterisks on the left margin.

0 Text Highlight
1 To highlight text press and hold Alt while dragging the mouse pointer over the text.

0 The row and column where the cursor is located in the editor window are shown on the info bar at the bottom of the application.
1 The following standard key combinations are available in the editor:

- Ctrl C - copy - copy the selected text into clipboard

- Ctrl V - paste - paste clipboard content into the text
- Ctrl X - cut - copy selected text into clipboard and remove it from the editor window.
- Ctrl Z - undo - undo the last typing action
- Ctrl Y - redo - do the previously undone action again
- Ctrl A - select all
- Ctrl] - jump to paired bracket
- Esc - Write Ahead
- Ctrl F2 - insert/delete marker
- F2 - jump to marker

0 The following additional key combinations become available when the editor operates under the query editor:

- Ctrl E , Ctrl F5 - Execute
- F5 - start debugging
- Alt F5, Shift F5 - stop debugging
- Ctrl S - save
- Ctrl G – jump to line
- Ctrl F - generate **SELECT table.column1, ..., table.columnN FROM table** for table name currently under the cursor
- Ctrl Shift F - generate a select without <table name> preceding every column name
- Alt F3 - Find - the Find function
- F3 - Find next (find the next occurrence of the selected expression)
- Shift F3 - Find previous (find the previous occurrence of the selected expression)
- F9 - set/remove breakpoint
- Ctrl D - find
- Ctrl B - breakpoint editor

Execute

Execute Highlighted

This function executes a Query or its highlighted part. If no portion is highlighted the whole content of the Query window is executed.

More Batches in One Text

When executing batches SQLing observes the convention of the GO command as described in MS SQL Server manuals. The pseudo-command GO is used to separate two batches in a single body of text.

Autosave When Executing

When executing queries, there is some danger that the editor "hangs". Therefore, all modified Queries are automatically saved into files query1.sql to query<n>.sql located in the $temp folder (usually c:\temp or d:\temp).

Database and Server

The target database and server are by default the active database and server.
In Properties, the user may set the database/server that Queries should always be executed on, without respect which database/server is active.

Independent process and thread

Before executing the Query SQLing opens a new SQL Server connection which is the same as that in which the Query would normally be launched. The application runs the Query in a separate thread. This ensures that the running Query can be stopped by the user at any time. When the Query is launched a Stop button () appears on the toolbar. Click on this button (or hit the Pause key) to stop the running procedure. This is technically done by closing the connection (disconnect) and stopping the thread.
The connection is also closed when the Query finishes normally, so it is not possible to write Queries for multiple launches (as when expecting that a temporary table created during the first launch would be filled with data during the second run of the same Query).

Set of Results

When enabled, the Results tab is automatically brought forward. This tab continuously displays the obtained results. Because the volume of results could be too large, every returned table is truncated so that it does not exceed the size limit specified in the Memory Usage field in Options.

Script directives added by SQLing – dynamic connection to SQL Server

If a single comment line containing the string $Connect S=server U=user P=password D=database T=1 is found at the beginning of a batch, SQLing first connects to the specified server and then launches all batches that follow the line. The same method is used to change the server back or to connect to another server.
There is no need to specify the database (master by default), user (sa by default), password (empty by default). The optional parameter T=1 means that the connection is

to be made in the Trusted mode (Windows NT Integrated Security).
Example:

```
select @@servername
go
-- $Connect S=diablo U=psu P=ppp
select @@servername
go
-- $Connect S=sdsql
select @@servername
go
$Connect S=diablo
select @@servername
```

Note: The second attempt to connect to the diablo server did not require user and password to be specified because existing connections are not closed when connecting to another server. Therefore, when connection to server diablo is opened for a second time, the previous connection is used. All connections to SQL servers created during the run of a query are disconnected when the query finishes.

Interserver INSERT … SELECT

You can use SQLing to write queries, which transfer data between servers connected in a query.
The set of servers connected in a query includes all servers connected by the above-mentioned directive - $Connect and the server on which the query is executed.
If Transact-SQL included an INSERT … SELECT construction, it would be sufficient to execute the following on server1:

```
INSERT server2.db2.user2.table2 SELECT ….
```

Start the transfer by executing the following in SQLing:

```
PRINT ”DINSERT server2.database2.user2.table2”
SELECT …
```

When loading script results, SQLing recognizes the keyword DINSERT (must be capitalized) and remembers that the results of the next SELECT should not be displayed in the Results window, but recorded in a table named table2 in the database named database2 on server2. Server2 must be one of the connected servers for the query and database2 must already exist on the server. If table2 does not exist it will be automatically created.

Example: a connection exists to a server named sdsql, to which we want to copy data from the server named diablo.master..sysdatabases:

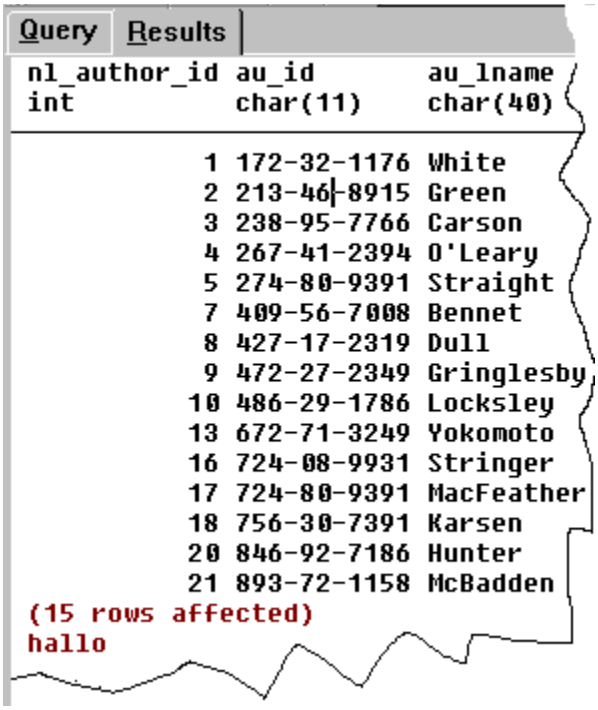
```
--$Connect S=Diablo D=pubs        connects to the diablo server, the remaining lines of the batch will be executed on that server
select * from authors             stores data from the table named diablo.pubs.authors in the results buffer
```

print 'DINSERT sdsql.pubs..help' this line does not display anything, it only informs SQLing that another set of results will be written to the table named help on the server sdsql

*select * from master..sysdatabases* this line does not display anything, stores data in sdsql.pubs..help

*select * from jobs* loads the table jobs into results

Results



The tab with results contains the most recent results received from the SQL server after a Query/Procedure is run. The results consist of two types of information:

- Tables - displayed in black color. In addition to data the tables also contain other information. The first line shows the names of columns in the table (the table does not have to be real - it could be a fictitious table without any column names such as can be loaded into the results buffer by executing a Query: *SELECT 1*). The second line shows the data types of columns and is followed by the data itself, separated by a horizontal line. Data is not present if the SELECT command returned an empty table. The number of data rows is truncated to match the length specified in the Memory Usage field in Options. Table data is followed by information about the total number of lines and a statement whether the table was loaded completely or whether it had to be truncated.
- Information and error messages - displayed in magenta color.

0 Results and the Clipboard

1 The Results window is a Read Only Edit Control. This means that it contains the cursor which can be moved and used to highlight text which can then be copied into the clipboard using the standard keyboard shortcut Ctrl C.

2 Tables into Table View

3 Although the table data is displayed and can partially be copied into the clipboard it lacks the comfort of work available in other table views in SQLing (sorting by columns,

clipboard operations on table level). Table data can be copied into the more effective table views by pressing the right mouse button over the data and selecting the only item of the shortcut menu that appears: View in Table. This function opens a table view containing the required data.

4 The function is only available if the option Another Delimiter in the Result Output Format field of Query Options is selected.

5 In the table view a shortcut menu containing these two items is available:

- Save - save table data into a text file delimited by tabs.
- Save To SQL Table - prompts for the name of a table (in the current database on the current server) into which the data is INSERT-ed. If no such table exists it is automatically created by SQLing. The name of the table can include a complete T-SQL table reference in the format database.user.table.

0 Go To Error

1 If any error messages are registered during the execution of a Query/Procedure it is not difficult to discover where the error occurred because the error messages contain the number of the source code line. It is much more difficult to determine the position of an error in queries consisting of several batches, because line numbering begins with 1 in each of the batches and therefore does not denote an absolute position in the whole query. You can, however, place the cursor on the error message and press F4. The Query tab is brought forward and the cursor is placed over the line where the error occurred. In this way even errors in large scripts can be tracked.

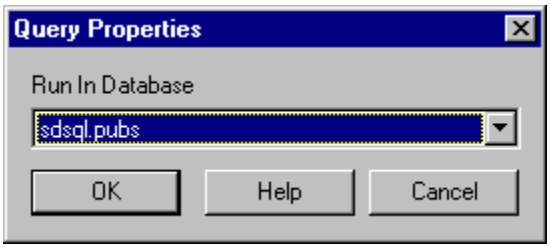
2 Save, Print, Find, Find Next

3 If the Results tab is active these toolbar buttons apply to the Results window, not to Query.

-  Save - saves the content of the window into a text file.
-  Print... - prints text.
-  Find - finds an expression (Alt F3)
-  Find Next - finds the next occurrence of the selected expression (F3).

Properties

Activate this function by pressing the button on the Query Editor toolbar. The following dialog box appears.



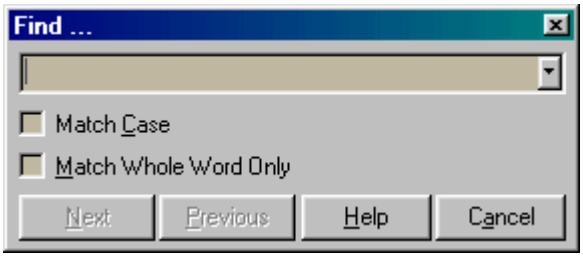
Specify the database and server on which the Query will be executed. The default value is <Auto>. This means the query will be run on the database/server set in the main application toolbar. Use the combo boxes to select from all available databases.
Press the Query Options button to change other specific properties of the Query Editor.

Find

0 Search for text in the Query Editor window

0 1. Select Find ( ) on the toolbar.

1 A dialog box entitled Find appears.



0

1 2. Enter the expression you want to find in the field Find What.
3. Check Case Sensitive to search for text which contains capitals and lowercase letters exactly as typed in the Find field.
4. Select Next to search for the next occurrence of the text.
0 5. Select Previous to search for the previous occurrence of the text.

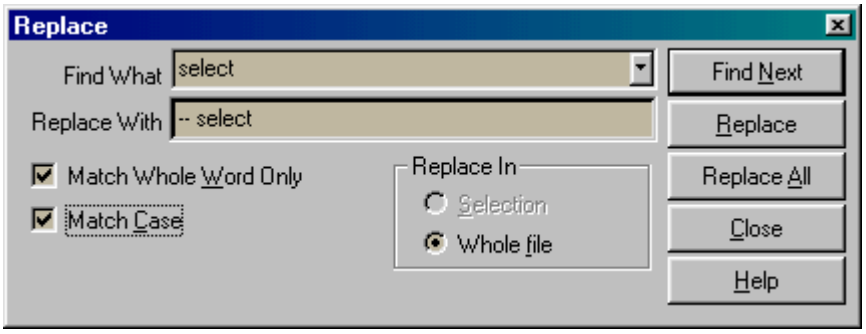
Replace

Find and replace a word or expression in the Editor window.

1. Select Replace from the editor shortcut menu.

A dialog box entitled Replace appears.

0



1 2. Enter the expression you want to find in the Find what field.
2 3. Enter the expression with which you want to replace the found text in the field Replace With.
3 4. Check Match Whole Word Only to search only for whole words, not their parts.
5. Check Match Case, to search for text which contains capitals and lowercase letters exactly as typed in the Find What field.
6. Group Replace In
   a) Select the radio button entitled Selection if you wish to replace the highlighted text.
   0 b) Select the radio button entitled Whole file if you wish to replace the whole text in the Query Editor window.
7. Do the following:
   a) Click on Find Next if you do not wish to make any changes and proceed to the next occurrence of the text.
   0 b) Click on Replace to replace text and find the next occurrence.
   1 c) Click on Replace All to change all occurrences without prompting.

Define Vocabulary

This function enables you to define a group of words which will be displayed in the Query Editor window using a special color.
What else could this user-defined vocabulary contain than the names of databases, tables, their columns and data types? Normally, no other words are used in the query editor. Such vocabulary would, however, be quite extensive and it might be impracticable to enter all the special words manually. Therefore, the special vocabulary can be defined as one or several SELECT commands which return the required set of words. All the words are stored in the system tables of the SQL server and by writing the appropriate SELECTs the user can save a lot of effort.

1. In the query editor, create a Transact SQL definition (SELECT) of the special vocabulary.
Examples:

a) SELECT name FROM systypes
returns the set of all data types defined in the current database.

b) SELECT name FROM master..sysdatabases
returns the set of all databases on an SQL server.

c) SELECT 'go'
defines the word 'go'.

0 3. Select Define Vocabulary from the Query menu in the Query Editor menu.

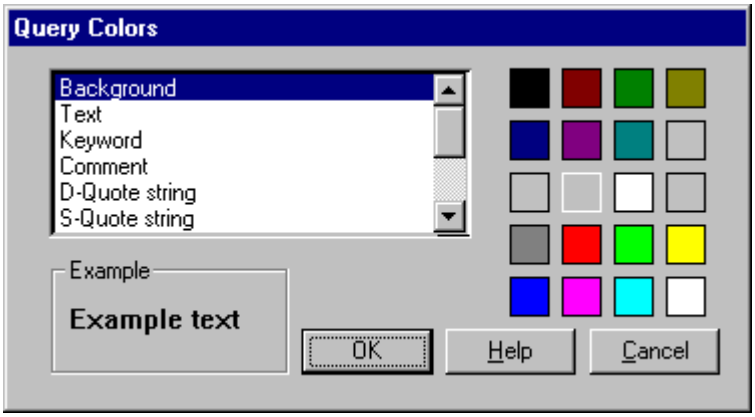
1 4. The words defined by the query result are displayed using different color immediately after the query has been executed. You can customize the user color by selecting Colors... from the shortcut menu of the Query Editor. Change the color for the syntactical group named User Defined.

Colors

Customize colors in the Query Editor.

1. Select Color... from the shortcut menu of the Query Editor.

A dialog box entitled Query Colors appears.



0

1 2. Select the syntactical group (class) of words and click on one of the color rectangles to assign a color to it. (You can select from twenty different colors).

2 3. You can select one of the following syntactical groups:
3 SQL syntax:
Text - unassigned words
0 Keyword - keywords
1 Comment - comments
2 D-Quote String - strings delimited by double quotation marks
3 S-Quote String - strings delimited by single quotation marks
4 Number - numbers
5 Variable - variable and temporary tables
6 User Defined - user defined words
HTML syntax:
0 HTML Element Name
1 HTML Attribute Name
2 HTML Attribute Value
3 HTML Comment
4 HTML Tag Delimiter - brackets (< >)
5 Margin markers:
6 Editor Bookmark
7 Breakpoint - debugger breakpoint
8 Current Statement - currently debugged line

9 You can also change:

Background - background color
0 Block text - text highlight color
1 Block Background - highlighted text background

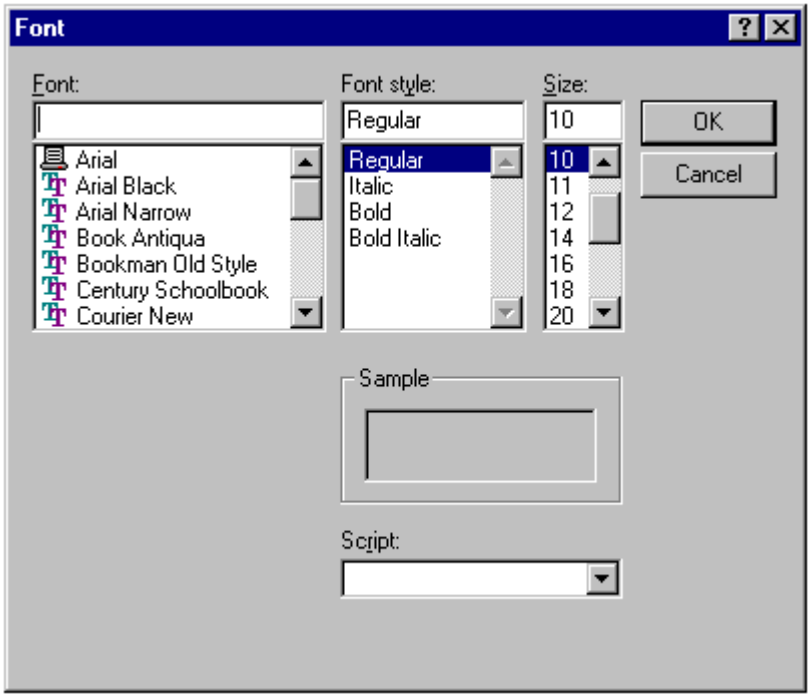
0 4. Press OK to confirm your selection.

Print

Print the text typed in Query Editor or displayed in the Results window.

1. Click on the Print ( ) button on the toolbar.

0 A dialog box entitled Font appears.



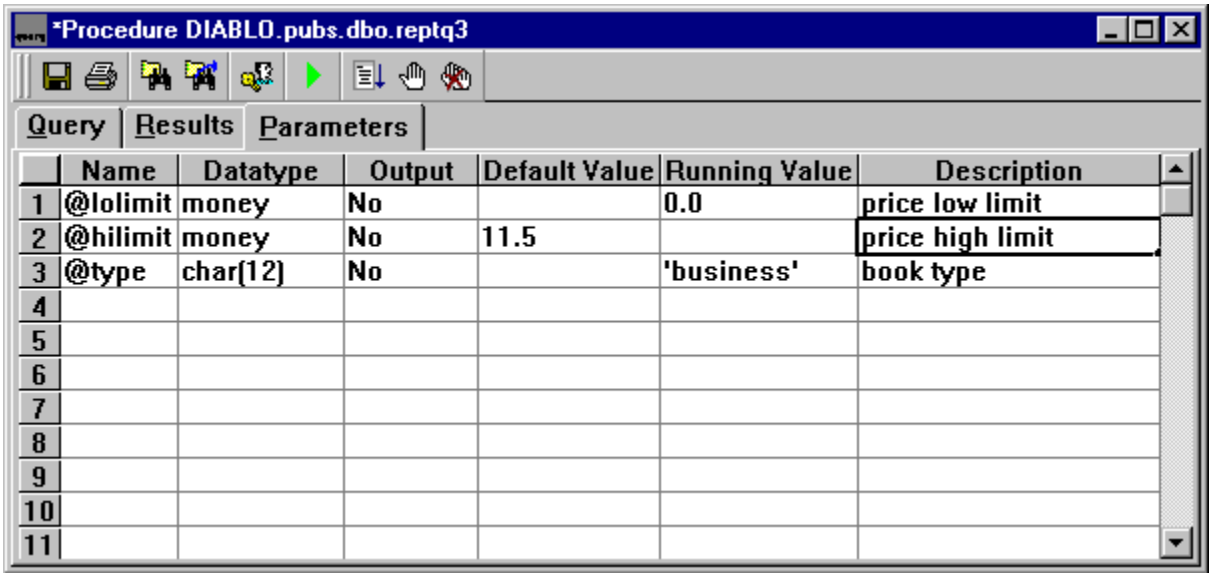
0

1 2. Choose the font, font style and size.

2 3. Press OK.

Parameters

The Parameters tab of the Procedure Designer contains the definitions of procedure parameters.



	Name	Datatype	Output	Default Value	Running Value	Description
1	@lolimit	money	No		0.0	price low limit
2	@hilimit	money	No	11.5		price high limit
3	@type	char(12)	No		'business'	book type
4						
5						
6						
7						
8						
9						
10						
11						

Parameters are located in a table with these columns:

- Name - name of the variable which represents the parameter.
- Datatype - variable data type
- Output - Yes/No value defining whether the parameter is of the output type or not.
- Default Value - default value assigned to a parameter when the procedure is called without explicit parameter value specification.
- Running Value - value assigned to a parameter when the procedure is executed from the Designer (Execute).
- Description - parameter description stored in the source code of the procedure as a comment following parameter definition.

0 Not all columns are used to define parameter structure. The Running Value column enables values to be assigned to parameters when the procedure is run/debugged from the Designer. Changes to the values in this column are not regarded as modifications to procedure structure (no asterisk appears on the left side of the Designer window title bar).

1 Table Properties

2 The tables have most of the standard table properties, e.g.

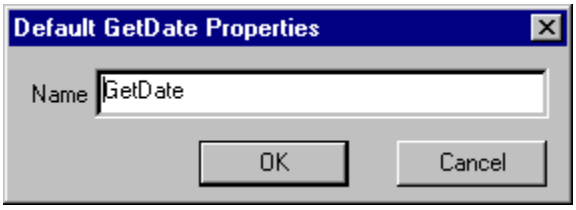
- clipboard functionality
- changeable width of rows/columns
- cell replication
- cell values editable in editboxes/comboboxes (by pressing F2 or double clicking on the cell)

0 but do not support sorting by column - by double clicking on the column title.

These properties are described in the chapter Table Containing SQL Table Data.

Default, Rule - Properties

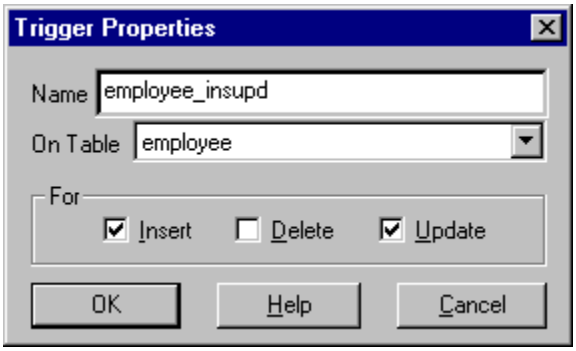
Click on the Properties ( ) button on the toolbar of the default or rule designer to display the following dialog box



where you can change the name of the default or rule.

Trigger Properties

Click on the Properties ( ) button on the toolbar of the trigger designer to display the following dialog window

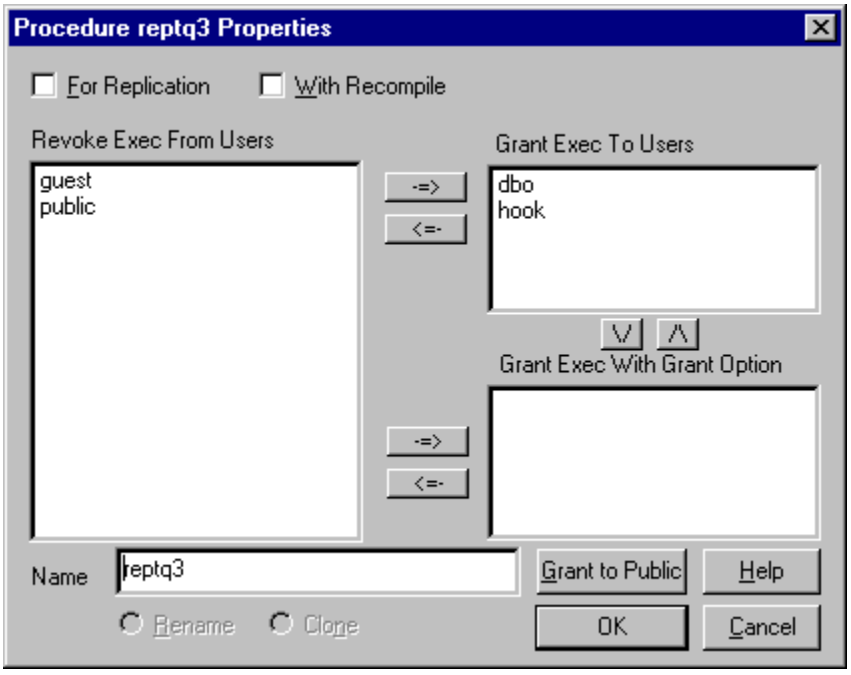


where you can view/change

- Trigger name - column Name
- Table on which the trigger is created - On Table
- Type of action (Update, Delete and/or Insert) validated by the trigger
- Click on Query Options to set other specific options of the editor.

Procedure and View Properties.

Click on the Properties ( ) button on the toolbar of the procedure or view designer to display the following dialog box



where you can view/change

- User permissions for the procedure/view - collated from three listboxes: Revoke Exec from Users, Grant Exec to Users and Grant Exec with Grant Option. Includes names of all database users (user groups). Move users as desired using buttons with arrows. The Grant to Public button is only a shortcut which moves the public group into Grant Exec to Users - this function is frequently used.
- Procedure/view name. Once you change the name, two more buttons become enabled: Rename and Clone. Rename changes the procedure name when it is next saved. Select Clone if you wish to create a procedure similar to an existing one without having to write it from scratch. A procedure with the new name will be created when next saving the work and only the new procedure will be shown in the Designer when you continue working.
- For Replication and With Recompile options. These options are only displayed for a procedure. Views with these properties cannot be created.
- Select Query Options to set other specific editor options.

Trigger, Default, Rule - Execute

0 Click on the Execute () button on the toolbar of the trigger, default, rule designer to save the object in the original database on the original SQL server.

1 Set of Results
2 The Results tab is automatically brought to foreground while the saving process is in progress. This tab lists any error messages. If no errors occur while saving the object, the Results window remains empty.

3 Transaction
4 If you work with SQL Server 6.5 or higher any objects are saved in a transaction and if they cannot be saved the original object is not dropped.

View - Execute

0 If the view was modified click on the Execute () button on the Designer toolbar to save the view in the original database on the original SQL Server.

1 Set of Results
2 The Results tab is automatically brought to foreground while the saving process is in progress. This tab lists any error messages. If no errors occur while saving the view, the Results window remains empty.

3 Transaction
4 If you work with SQL Server 6.5 or higher any views are saved in a transaction and if they cannot be saved the original view is not dropped.

5 Modification Flag
6 If the view was saved successfully the designer modification flag is removed (the asterisk from the designer window title bar disappears).

7 If the view was not modified (modification flag is not set) click on the Execute () button on the designer toolbar to run the view.

*8 SELECT * FROM <view name>*

9 and display the Results data represented by the view.

Procedure Execute

If the procedure was modified click on the Execute ( ) button on the Designer toolbar to save the procedure in the original database on the original SQL Server.

Set of Results

The Results tab automatically switches to foreground while saving. This tab lists any error messages. If no errors occur while saving the procedure, the Results window remains empty.

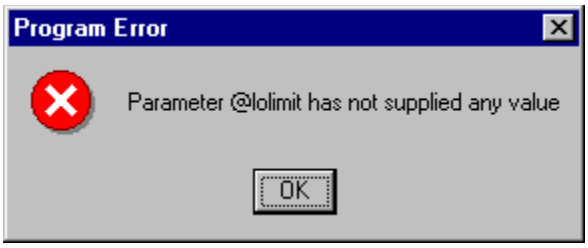
Transaction

If you work with SQL Server 6.5 or higher any procedures are saved in a transaction and if they cannot be saved the original procedure is not dropped.

Modification Flag

If the procedure was saved successfully the designer modification flag is removed (the asterisk from the designer window title bar disappears) and the procedure is not saved when next executed, although:

If the procedure was not modified (modification flag is not set) clicking on the Execute ( ) button on the designer toolbar the SQL procedure named EXEC is run. If any parameters are specified for the procedure (in the Parameters tab), their Running Values must be defined. These values will be passed to the procedure as parameters. The parameters for which no Running Values are defined must either be of the OUTPUT type or should have a defined Default Value. If this condition is not met SQLing displays the following warning



and the procedure is not executed.
If values for all parameters are defined the procedure is executed and its results are continuously displayed in the Results tab. Because the results can be extensive, any returned tables are truncated to meet the size limit specified in the Memory Usage field in Options.

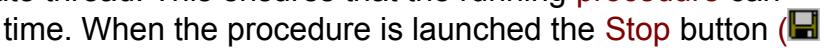
Output Parameters

Any procedures containing output parameters are executed using the following structure

```
DECLARE @i int
0  EXEC Procedure1 @i=@i OUTPUT
```

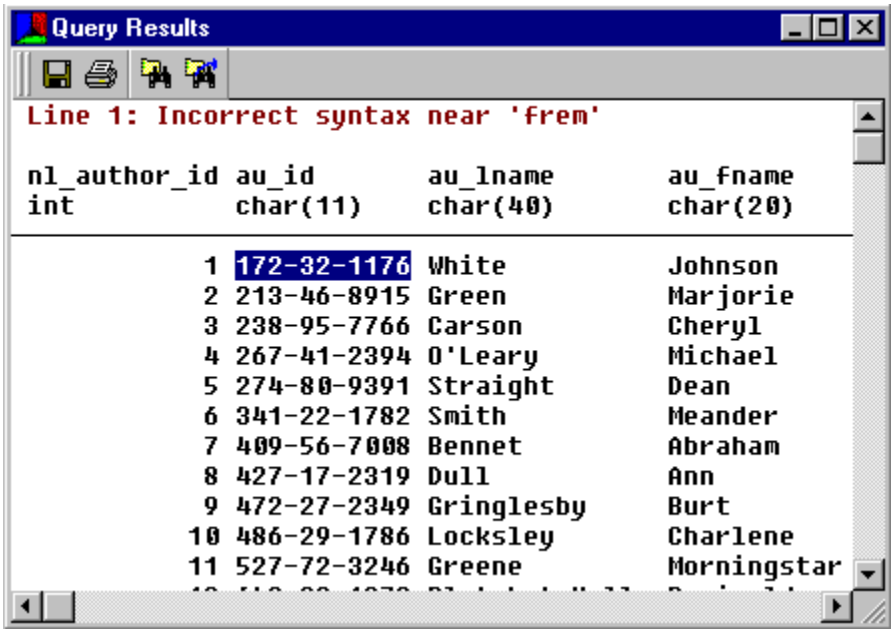
```
1  PRINT 'Values Of Output Parameters'
2  SELECT '@i'=@i
```

0 which enables the output parameters to be viewed after the procedure finishes.

1 Independent process and thread.
2 Before executing any procedure SQLing opens a new SQL Server connection which is the same as that in which the procedure would normally be launched. The application runs the procedure in a separate thread. This ensures that the running procedure can be stopped by the user at any time. When the procedure is launched the Stop button () appears on the toolbar. Click on this button (or hit the Pause key) to stop the running procedure. This is technically done by closing the connection (disconnect) and stopping the thread.

Results Window

This window is used to display the results returned when a script is executed on the SQL server or to view other text results (e.g. the results of database comparison).



The results consist of two types of information:

- Tables - displayed in black color. In addition to data the tables also contain other information. The first line shows the names of columns in the table (the table does not have to be real - it could be a fictitious table without any column names such as can be loaded into the results buffer by executing a Query: *SELECT 1*). The second line shows the data types of columns and is followed by the data itself, separated by a horizontal line. Data is not present if the SELECT command returned an empty table. The number of data rows is truncated to match the length specified in the Memory Usage field in Options. Table data is followed by information about the total number of lines and a statement whether the table was loaded completely or whether it had to be truncated.
- Information and error messages - displayed in magenta color.

0 Results and the Clipboard

1 The Results window is a Read Only Edit Control. That means it contains a cursor which can be moved and used to highlight text which can be copied into the clipboard using the standard keyboard shortcut Ctrl C. Highlight the whole text by Ctrl A.

2 Tables into Table View

3 Although the table data is displayed and can partially be copied into the clipboard it lacks the comfort of work available in other table views in SQLing (sorting by columns,

clipboard operations on table level). Table data can be transformed into the more effective table views by pressing the right mouse button over the data and selecting the only item of the shortcut menu that appears: View In Table. This function opens a table view containing the required data.

4 Save, Print, Find, Find Next

5 The window has a toolbar which contains the following buttons:

-  Save - saves window content into a text file.
-  Print... - prints text.
-  Find - searches for information (Alt F3)
-  Find Next - finds the next occurrence of the selected expression (F3).

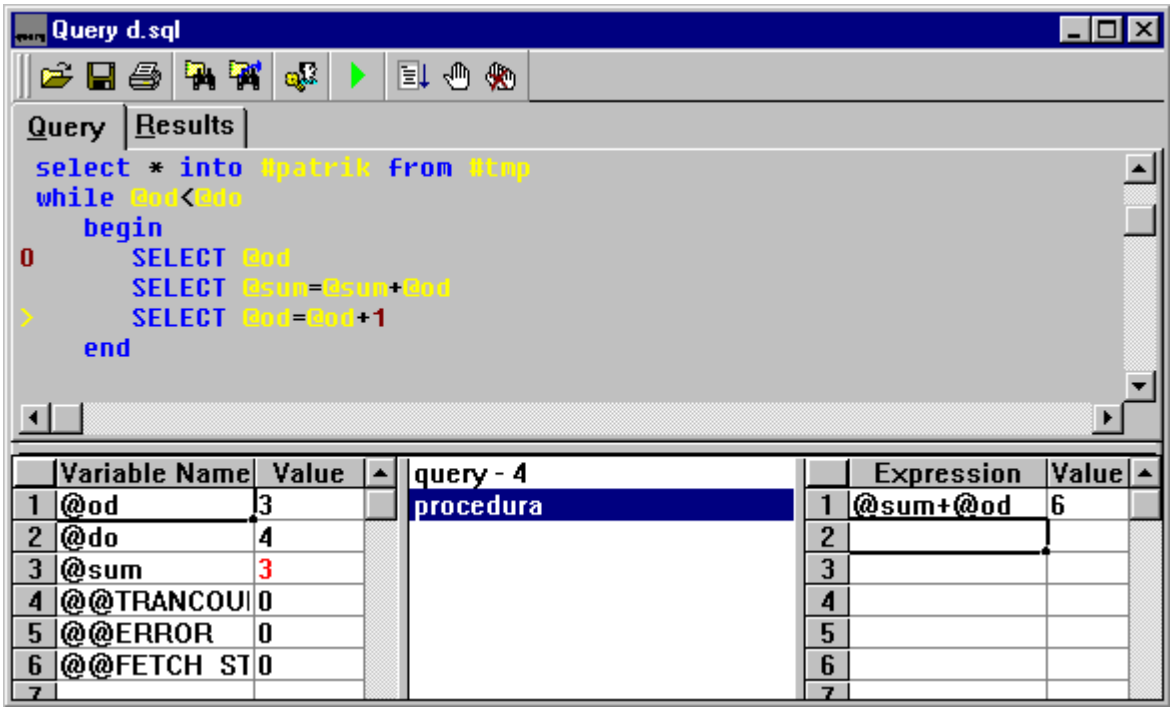
Debugger

The Debugger is controlled using the following keys

- F11 – run the debugger and stop at the first command of the query/procedure.
- F5 (💾) – run the query/procedure (only procedure hereafter) in the debugger. The procedure is executed step by step and any called procedures are also debugged. Debugging stops whenever any of these three cases occurs:
  - The end of the procedure is reached.
  - A breakpoint is reached (breakpoints are set/removed using the F9 (💾) key); the breakpoint editor can be retrieved by pressing Ctrl B.
  - When the Pause/Break key is hit or the toolbar button Stop (💾) clicked.
- F7 - as F5, execution is stopped when the debugger reaches the line where the cursor was located when the function was invoked.

0 If the debugger is running but the execution of the procedure/query is halted the current position is marked by a yellow marker (”>”) on the left margin next to the current line.

1 Three subwindows are shown in the bottom part of the Query window while debugging



2

3

4 The first (from left to right) contains a list of procedure parameters, variables used and some useful global variables including their current values.

5 The second subwindow contains a list of procedure stack calls.

6 The third subwindow is an expression Watch. The second column of the table contains the current values of all the expressions listed in the first column. The value of the expression is obtained as: *SELECT expression*. Here are some examples of

expressions:

- @a+@b
- COUNT(*) FROM authors
- 1

At this moment the following keys/toolbar buttons are available:

- Alt F5 () – stop debugging
- F5 () – continue to run the procedure (see above)
- F8 (), F11 () – execute the current command in the procedure. If the current command is *EXEC procedure* then the debugger either steps into the called procedure (Step In – F8) or runs the procedure without debugging its content (Step Over – F11). Whenever a similar portion of code is executed all monitored (Watched) variable/expression values in their respective windows are reset – this may slow down the debugging process, so you should regularly remove any unnecessary expressions from the list.
- F7 () - continue execution until the line with the cursor.
- Changing values in Local Variables table is possible and accepted in time of debugging.

Restrictions

1. Triggers cannot be debugged.
2. Step In is not available for an *INSERT .. EXEC* structure.
3. Step In is not possible when calling system procedures or procedures located on another server.
4. Contrary to procedures executed normally, the debugged procedures do not run in cloned processes, so you can browse table data without risking process blocking. It is not recommended to run more complicated functions (upsize, compare etc.) while debugging.

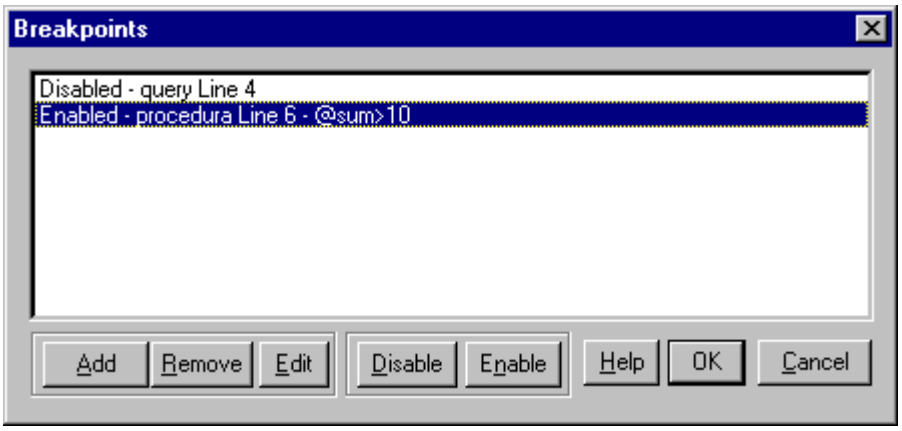
Breakpoint Editor

SQLing supports three types of breakpoints

- Positional - the execution of the debugged procedure stops before the line with the breakpoint. Set/remove the breakpoints of this type by F9 or the button in the editor toolbar. Positional breakpoints are marked in the text by a magenta circle (”O”) on the left margin.
- Conditional - a set of conditions (Boolean expressions). The execution of the procedure is stopped if at least one expression is TRUE. Every conditional breakpoint must be bound to a procedure in which (and only in there) its validity is tested.
- Combined - conditional breakpoints bound to a particular position in the particular procedure.

0 All these types of breakpoints can be edited in the Breakpoint Editor which is retrieved by pressing Ctrl B.

1

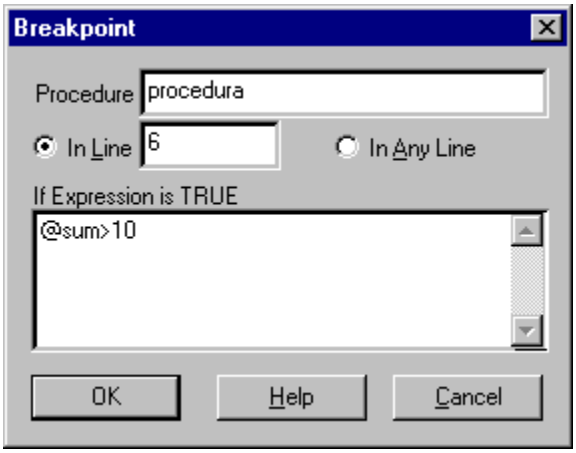


2 The selected breakpoint can be temporarily disabled and later enabled using the Enable and Disable buttons.

3 Use the buttons Add, Edit and Remove to add new, edit/remove existing breakpoints.

Breakpoint Editor

Add a new breakpoint or edit the existing one in this dialog box



containing the columns

- Procedure - name of the procedure in which the breakpoint will initiate (if you want to make the breakpoint active in the debugged query, set procedure name to ”query”).
- In Line - if the breakpoint is bound to a line in the procedure/query this radio button is selected and the number of the line is shown in the edit field.
- In Any Line - this radio button is selected if the breakpoint is not bound to a line in the procedure.
- If Expression is TRUE - contains a condition (unless the breakpoint is not conditional or combined). If this field contains no text the breakpoint is considered as positional.

0 Examples of conditions

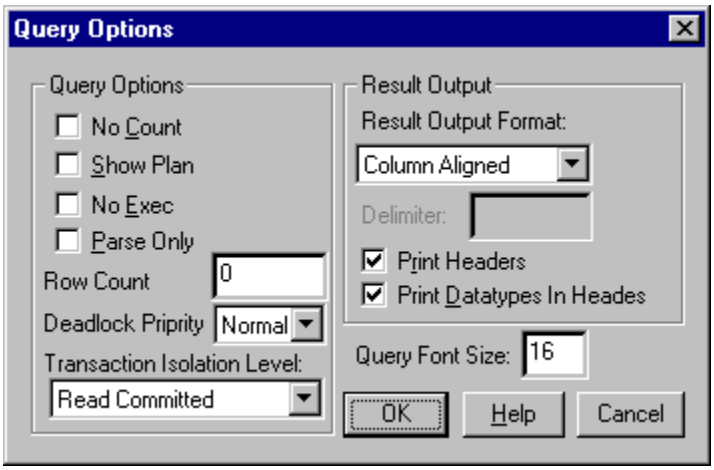
- 0<(SELECT COUNT(*) FROM #tab WHERE a=@a)
- @@TRANCOUNT>2
- @A+@B<@C

Query - More Options

Local and Global Query Options

Any opened Query Editor window has its local options. Modify the local options in the Properties of the query (procedure) editor.
Set the Global Query Options in Options. Local options of the query editors opened after global options were changed will be modified accordingly. After changing global options you will be prompted whether you wish to reset the local options of all currently opened query editors accordingly.

Query Options Dialog Box



Query options fields:

- No Count - do not show the number of lines after each table in the results.
- Show Plan - SQL query optimization output.
- No Exec - only compilation instead of execution. Used in conjunction with Show Plan.
- Parse only - syntactical analysis instead of execution.
- Row Count - limit the number of rows processed in the queries. 0 = unlimited.
- Deadlock priority - process priority when deadlock is reached.
- Transaction Isolation Level - see SQL Books Online.

0 Result Output fields:

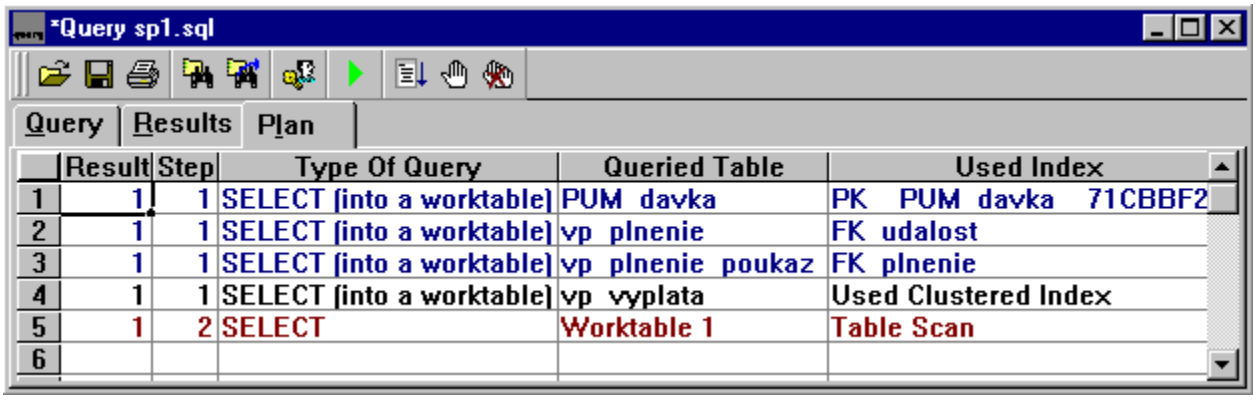
- Column Output Format - table format in Results .
  - Column Aligned - columns are aligned. Column width is set according to the longest text value.
  - Tab Delimited - values in columns are separated by tabs.
  - Another Delimiter - values in columns are separated by the character entered in Delimiter field (see below).
- Delimiter - the character delimiting column values in the Results window.
- Print Headers - the tables in options contain a header with the names of the columns and their data types.

- Print Datatypes In Headers - list column data types in table headers.

0 Font Size field
- Query Font Size - font size in the query editor.

Show Plan

This tab contains tabulated query optimization information for the last executed query/procedure.



	Result	Step	Type Of Query	Queried Table	Used Index
1	1	1	SELECT (into a worktable)	PUM_davka	PK_PUM_davka_71CBBF2
2	1	1	SELECT (into a worktable)	vp_plnenie	FK_udalost
3	1	1	SELECT (into a worktable)	vp_plnenie_poukaz	FK_plnenie
4	1	1	SELECT (into a worktable)	vp_vyplata	Used Clustered Index
5	1	2	SELECT	Worktable 1	Table Scan
6					

Every (data modification language) command (returning a Result) is processed in one or more Steps. The type of the command is recorded in the column named Query Type (do not be confused by the fact that SELECT could be of query type INSERT - this is caused by the query being gradually processed in several working tables).
The column named Used Index contains important information. The type of this value determines the color of the whole row. The possible values (sorted by optimized status):

- Used Clustered Index - Clustered Index was used in the optimization; highest possible optimized status of the query step.
- <Index Name> - the index <index name> was used in the optimization.
- Table Scan - no index was used in the optimization, the whole table had to be scanned.
- Dynamic Index - a temporary dynamic index was created while processing the query, the temporary index was deleted after evaluation.

The first two values mean that the query step is optimized, the last two mean that it is not optimal (lines marked red).

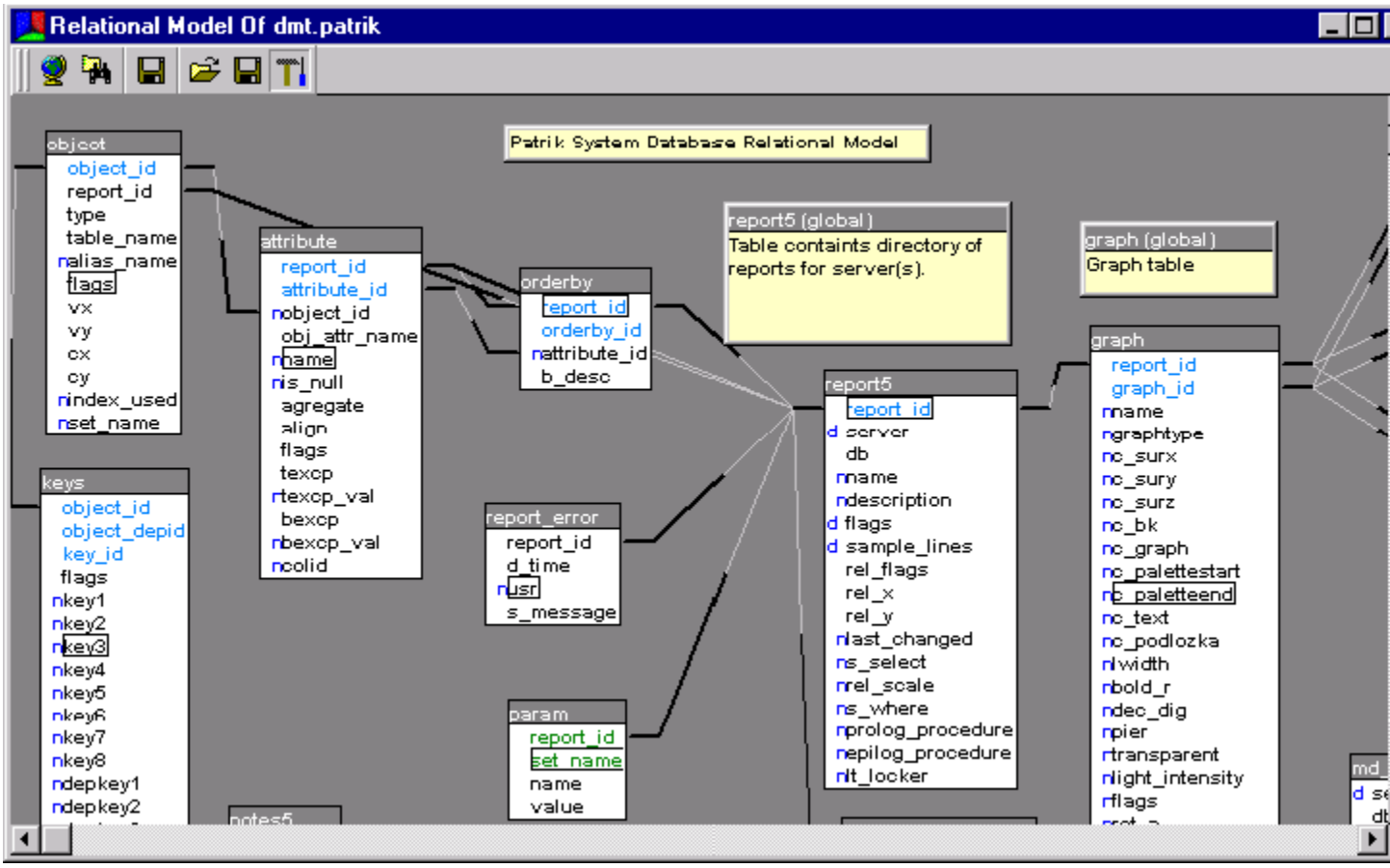
Relational Model

In this chapter, you will learn about the graphical representation of a relational model, about editing relations directly in the Relational Model window, defining views which only display the tables you select, searching for and transferring tables in the Relational Model window while using a map of relations, and printing relational models.

The Relational Model Window

SQLING enables you to browse and edit the Relational model (database structure and the relations between tables). In SQLING you can create, print and edit table relations. Relations are displayed in the window entitled Relational Model.

Click on the Relational Model... button ( ) in the application toolbar to display the Relational model.



Tables

SQLING displays tables of the current database with columns as windows and the relations between the relational tables as linking lines. You can move the table windows around as you like. Select Relations - Save Layout in the shortcut menu shortcut menu to save the layout of the tables.

Relations

The Primary Side of a relation is designated by a lighter line and the Foreign side by a darker line. A one to one relation line has uniform thickness along the whole length whereas a one to many relation line is thicker on the Foreign (many) side. If you click on a relation, its line will become thicker and new editing functions will be added into the shortcut menu. If you double click on a relation the Relation Editor is opened.

Notes

Although notes are not normally displayed, they may be included in relational views of a database. There are two types of notes:

1. Table comment - the caption in the note window contains the table name and the origin of the comment.
2. Free comment - the note window has no caption and contains an arbitrary description of (a part of) the model or anything else.

Add a note from the popup menu. Remove a note by pressing the Delete key. Edit a note by double clicking on its text.
Notes are saved automatically in relational views. Refer to Select/Deselect and Save View.

Creating a relation

MS SQL Server only supports one to many relations (one to one relations are one to many relations with the unique index on the foreign key). Because new relations are a modification to the database, they can only be created in a downsized database. Follow these steps to create relations:

1. Select the primary table key which will be bound by a relation to another table. Hold the left mouse button pressed after the last column of the primary key has been selected.
2. Drag the mouse pointer to the other table and release its button. A window entitled Edit Relation appears. Enter the necessary information and confirm to create a new one to many relation between the selected pair of tables.

Note: You can also add a relation by selecting Add Relation….

Table Columns

Table columns are displayed as the rows of a window which bears the name of the table. The columns which are primary table keys are shown in

- blue color - if the key is a Primary Key Constraint
- green color - if the key is an ordinary primary key (e.g. defined by the sp_primarykey system procedure)
- cyan color - if the key is a mixed type primary key, comprising both of the above types

Blue-colored letters may appear before the columns of a table to indicate that:

- i - an index is set for the column
- d - a default is set for the column
- n - the column may contain NULL values

In Options you can set whether these letters are displayed.

0 The Relational Window toolbar contains the following functions

-  Show Map - shows the relational model extremely minimized
-  Find Table - searches for tables in the relational model
-  Save Layout - saves the location of table windows into a database
-  Load View - loads a relational view from the database
-  Save View - saves the relational view into the database
-  Edit Mode - switches to the editing mode of the relational view

0 Other functions are located in the shortcut menu.

Shortcut Menu in the Relational Model Window

0 To display the Shortcut Menu containing commands for working with the relational model place the mouse pointer over the work area of the Relational Model window and click the right mouse button.

1 The SQL Server Shortcut Menu in the window showing the Relational Model of an actual database located on an SQL server contains the following commands:

- Relations
  - Show Map - shows a scaled-down map of a relational model.
  - Save Layout - saves the positions of windows in the relational model into a database.
- View Relation… - displays the definition of the selected relation. This function is only available if a relation is selected.
- Find Table… - search for tables.
- Move Table Here - moves the table (or several tables) to the specified location. This function is only available when the menu is invoked outside any table window.
- View table - displays table structure.
- Open Table - opens a window containing table data. This function is only available if any of the windows in the relational model is active.
- WHERE Open… - opens a window containing the data of the table which meets the specified WHERE condition. This function is only available if any of the windows in the relational model is active.
- Select COUNT(*) - displays the number of lines in the active table. This function is only available if any of the windows in the relational model is active.
- Select or Deselect - used in editing a View. The option is only available if the Edit Mode is enabled.
- Scale - scale the relational model window.
- Add Note - adds a note into the model view. May contain the following items:
  - Global Note - adds a global note to the selected table. Global notes are edited in the Table description field of the Global tab in the Design Table window.
  - Of User <user name> - adds a note with user-specific text to the selected table. User notes are edited by the Edit Note function.
  - Free Text - opens an empty note not bound to any table.
- View - contains input/output operations on views.
  - Load…
  - Save…
  - Delete…

0 The Shortcut menu for a working (downsized) database does not contain certain commands which are present in the SQL server Shortcut menu of an actual database. The missing commands include Open table, WHERE open, Select COUNT all of which work with the actual data not present in a working database. On the other hand, the menu contains commands modifying the relational model of the database:

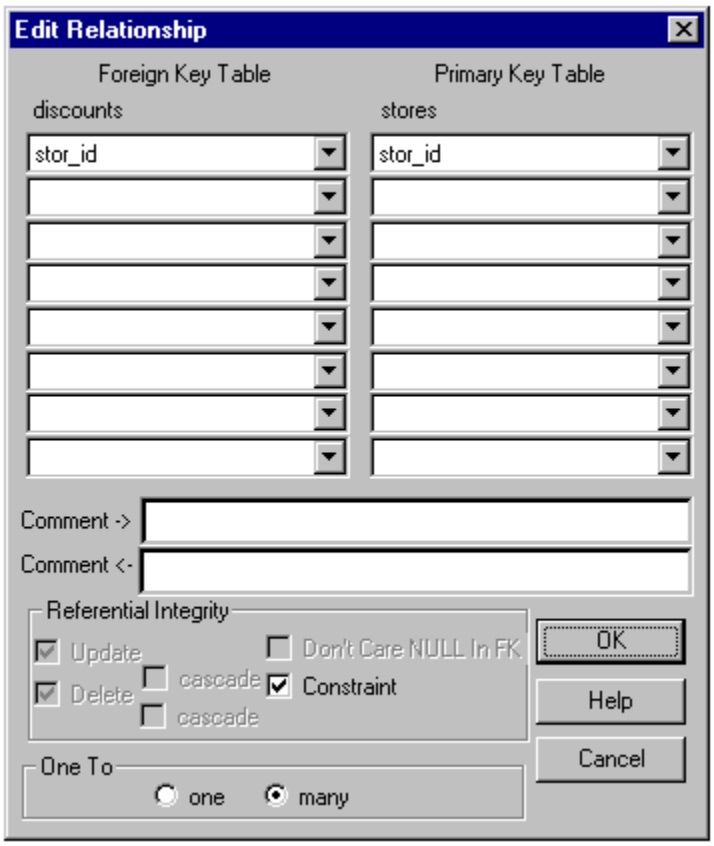
- Edit Relation… - displays the definition of the selected relation. This function can be used to redefine the relation. The function is only available if any relation is selected.
- Delete Relation - deletes the selected relation. The function is only available if any relation is selected.
- Add Relation… - adds a One to Many relation to the selected table.

Editing Relations between Tables

Relations between tables are represented in the relational model by lines linking the relational tables. The ends of the lines are colored in different shades of gray according to the type of relation. If the line end is light the table it connects is on the ”one” side of a relation. Dark end of a line means that the connected table is on the ”many” side (see the figure below).

Editing relations in the Relational Model window

1. In the Relational Model window, double click on the relation you wish to edit.
A dialog box entitled Edit Relationship appears.



2. Select the Pk attribute of the table in one of the eight combo boxes located on the left.

3. Select the FK attribute of the table in the relation combo box located on the right.
Repeat steps 3 and 4 to create combined relations.

4. Select relation relationship in the One to group.

5. In the group Referential Integrity:

- Check Constraint and specify the declarative referential integrity (foreign key constraints).
- Uncheck the Constraint check box to enable the check boxes Update, Delete and Cascade and to select referential integrity ensured by triggers.
- Check Update to ensure that the primary key value in the primary key table does not change immediately after the T-SQL UPDATE command is later executed if the record has any relational records.
- Select the Update check box to enable the Cascade check box next to Update.
- Check Cascade to activate Cascading Update in triggers which will execute the update of all relational records.
- Check Delete to ensure that primary table record is not deleted immediately after the T-SQL command DELETE is executed if identical records exist in the relational table.
- If the Delete check box is checked the Cascade check box next to Delete will become enabled.
- Check Cascade to activate Cascading Delete in triggers which will delete all relational records for every deleted record of the primary table.
- Select Don't Care Null In FK - to enable NULL value for the foreign key (this will permit orphans with NULL values in their foreign keys)

6. Add comments to the relation using Comments and Comments .

7. Press OK.

Show Map

Databases often contain a large number of tables and it is not practicable to display all tables on the screen at once. This makes it necessary to use the scrollbars of the relational model window to scroll the relational model view during browsing or editing. For a better overview of the relational model, SQLING provides a window named Map of Relational Model. In this window, tables are represented by small black rectangles which match the actual database tables as displayed by the Relational Model window.

Open the Map of Relational Model

1. Select Show Map from the Relationship shortcut menu in the Relational Model window.

Tables

The tables of the relational model are represented in the map view mode by black rectangles. Relations between tables are not drawn to increase display refresh rate. If you enlarge the map window all tables (black rectangles) are enlarged in proportion. If you click on any of the rectangles the name of the table which the rectangle represents is displayed in the lower (information) part of the map window. Drag and drop any table to change its position in the window.

Relational Model Window View

The position of the frame visible in the window is shown on the map as a white rectangle. Double click on any place on the map to shift the relational model view.

In the Map of Relational Model you can:

- Move a group of tables
- Print the selected tables

Move a group of tables

1. Frame the tables you would like to move in the white rectangle (see figure below)
   a) Place the mouse pointer anywhere in the Relational Model Map window (except any of the black rectangles).
   b) Hold the left mouse button pressed and move the mouse until all the tables you wish to select are inside the white frame.

2. When you release the left mouse button a Shortcut menu appears.

3. Select Move Area.

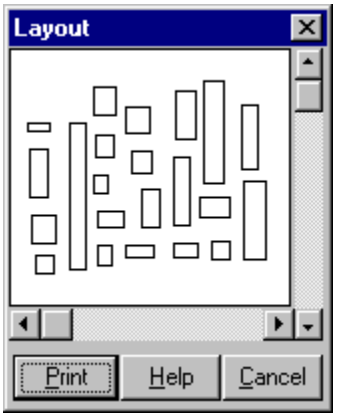
4. Move the rectangle to another place.

Print a group of tables using the Relational Model Map Window

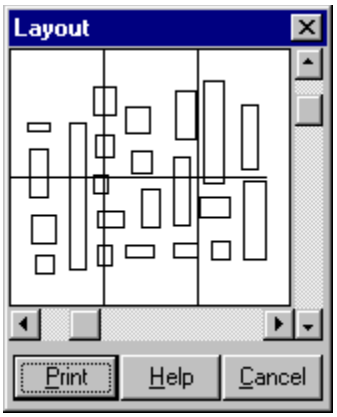
1. Frame the tables you would like to print in the white rectangle (see figure below)
   a) Place the mouse pointer anywhere in the Relational Model Map window (except the black rectangles).
   b) Hold the left mouse button pressed and move the mouse until all the tables you wish to select are inside the white frame.

2. When you release the left mouse button a Shortcut menu appears.
   Select Print Area.

Set the printing format in the window entitled Layout which appears.



3. Move the vertical scrollbar down and the horizontal scrollbar to the right to set the printing format.



SQLING will print the pages in the following order (this is the order for 12 pages).

1	4	7	10
2	5	8	11
3	6	9	12

Save Layout

Save the changes to the positions of tables in the Relational Model

1. Select Save Layout in the Relationships shortcut menu of the Relational Model window.

The layout of the tables is stored in the SQLing database

- As a result, the window layout in the relational model will be the same for all users on all workstations.
- If, when transferring the database, you do not use the transfer function of SQLing the tables in the target database will not retain the layout they had in the source database, they will completely lack any order. Use the command Transfer - Metadata Layer to transfer the relational window layout alone.

Find Table

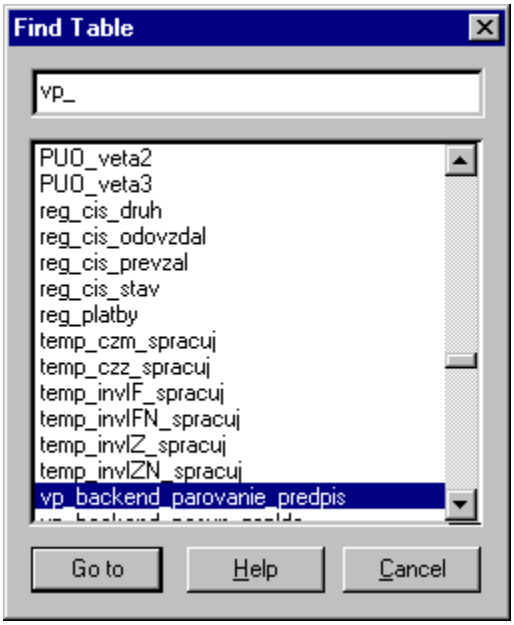
Use this command to find the table you need in the Relational Model window quickly.

Find Table in the Relational Model Window

1. Place the mouse pointer over any table in the Relational Model window and click the right mouse button.

2. Select Find Table from the Shortcut Menu.

A dialog box entitled Find Table appears.



3. Do the following:
   a) In the dialog box, enter the name of the table you would like to find in the blank field named Find Table.
   0 b) Select the table in the field containing a list of tables.

0 4. Press Go to.

1 SQLING highlights the found table and centers the viewing frame of the relational model window on the table.

Add Table

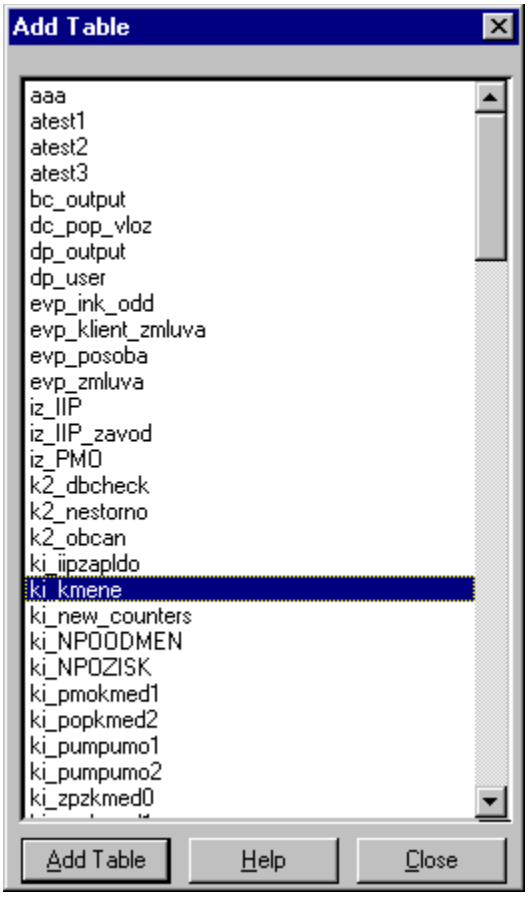
Only tables with defined primary keys or tables connected by relations with other tables are shown in the Relational Model window. Tables without a primary key or without relations are not displayed in the Relational Model window but can be added using the Add Table command.

Adding a Table into the Relational Model Window

1. Place the mouse pointer over any table in the Relational Model window and click the right mouse button.

2. Select Add Table.

A dialog box entitled Add Table appears, containing the tables not displayed in the relational model window.



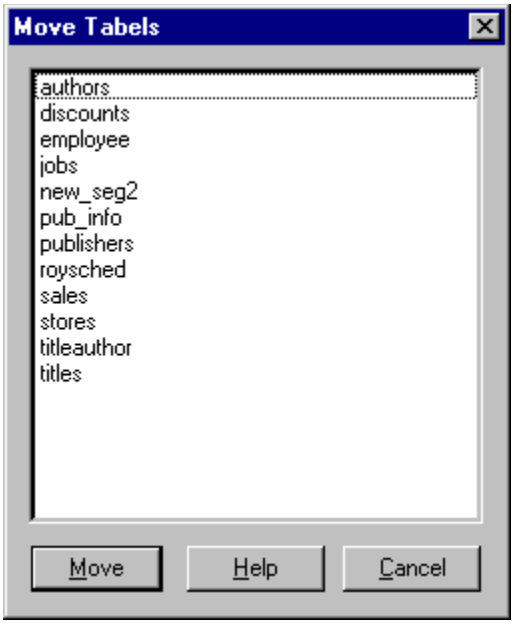
3. Select the table you would like to add into the relational model.

4. Press Add Table.

SQLING will add the selected table into the Relational Model window. It is possible to define relations between tables directly in the Relational Model window. For more information, see the section Editing Relations Between Tables.

Move Table Here

Move the mouse pointer to the place in the relational model where you wish to place a table (or several tables) and select the function Move Table Here from the popup menu. A dialog box containing the names of all tables included in the relational model view appears.



0 In the dialog box, highlight the table you wish to move to the selected location and press Move. The table will be moved in the relational model. If you want to move more tables, repeat the selection and the Move command for each of the tables.

Select Count

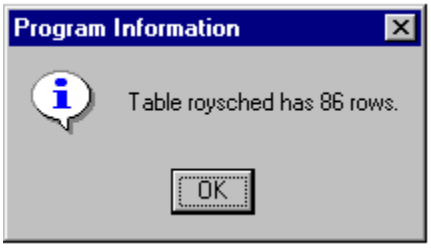
Use the Select Count(*) command to count the number of records in the selected table.

Determining the size of the selected table

1. Place the mouse pointer over any table in the Relational Model window and click the right mouse button.

2. Choose Select Count(*) in the Shortcut menu.

A dialog box entitled Program Information appears.



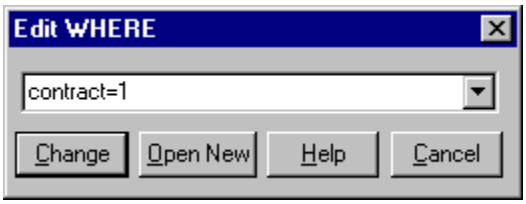
Where Open

Display table data with the defined condition.

1. Place the mouse pointer over any table in the Relational Model window and click the right mouse button.

2. Select WHERE Open… in the Shortcut menu.

A dialog box entitled Type Where appears.



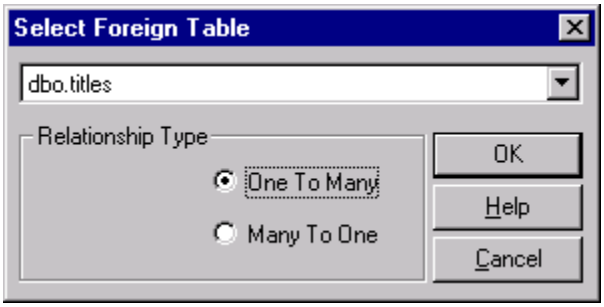
3. Enter a WHERE condition.

4. Press OK.

SQLING will open a new window and display the part of the table which contains the data that meets the defined condition.

Add Relation

1. The following dialog box appears, containing a combo box with all tables which are included in the relational model.



2. Select the table which you wish to be in relation with the connected table.
3. Select the type of relation (many to one or one to many) between the connected table and the table selected in point 2.
4. Set all other relation attributes in the Edit Relation dialog box, which appears.

Here is another method for adding a relation:

1. In the Relational Model window, place the mouse pointer over the primary key attribute of the table on the ”one” side of the ”one to many” relation.
2. Press the left mouse button and drag the primary key attribute to the foreign key attribute in the table which you want to engage in the relation.
3. Set all other relation attributes in the Edit Relation dialog box, which appears.

Select, Deselect Commands

Define a View

1. Place the mouse pointer over any table in the Relational Model window and click the right mouse button.

2. Choose Select from the Shortcut menu.
   a) Choose Alone to select the desired table.
   b) Choose With Children to select the group of tables connected to the current table by a ”one to many” relation (provided the current table is on the ”one” side of the relation).
   0 c) Choose With Whole Family to select the group of tables connected to the current table by any relation.

Repeat these steps to select all the tables you wish to be included in the View.

0 3. Select Save As in the shortcut menu to store the definition of the View in the database.

1 Cancel the selection of a table using the Deselect command.

2 1. Place the mouse pointer over any table in the Relational Model window and click the right mouse button.

3 2. Choose Deselect from the Shortcut menu.
   a) Choose Alone to deselect the selected table.
   b) Choose With Children to deselect the group of tables connected to the current table by a ”one to many” relation (provided the current table is on the ”one” side of the relation).
   0 c) Choose With Whole Family to deselect the group of tables connected to the current table by any relation.

3. Choose Deselect from the Shortcut menu.

Load Command

Load a defined View

1. Select Load in the View menu of the Relationship window.
A dialog box entitled Choose View to Load appears.

2. Select the view you wish to load.

3. Press OK.

4. The View is loaded and the relational model will exit the Edit Mode.

Save Command

Save the changes made to a View

1. Select Save in the View menu of the Relationship window.
A dialog box entitled Choose View to Save as appears.

2. Enter the name of the changed View.

3. Press OK.

Delete Command

Delete a defined View

1. Select Delete in the View menu of the Relationship window.
A dialog box entitled Choose View to Delete appears.

2. Select the view you wish to delete from a field listing all views.

3. Press OK.

View Dialog Box

is used by functions

- Save
- Load
- Delete

0 which operate on the views of the relational model.

The Edit Mode Button

Switches between two modes.

The Relational Model can operate in these two modes:

- Edit Mode on - the toolbar button  is depressed - The commands Select, Deselect are shown in the shortcut menu; highlighted/selected tables are those with blue background. Unselected tables are shown on white background.
- Edit Mode off - the toolbar button  is not depressed - The commands Select, Deselect are not available in the shortcut menu and only tables which were selected (blue background) in the Edit Mode are shown.

When relations are opened the Edit Mode is on and no tables are highlighted.

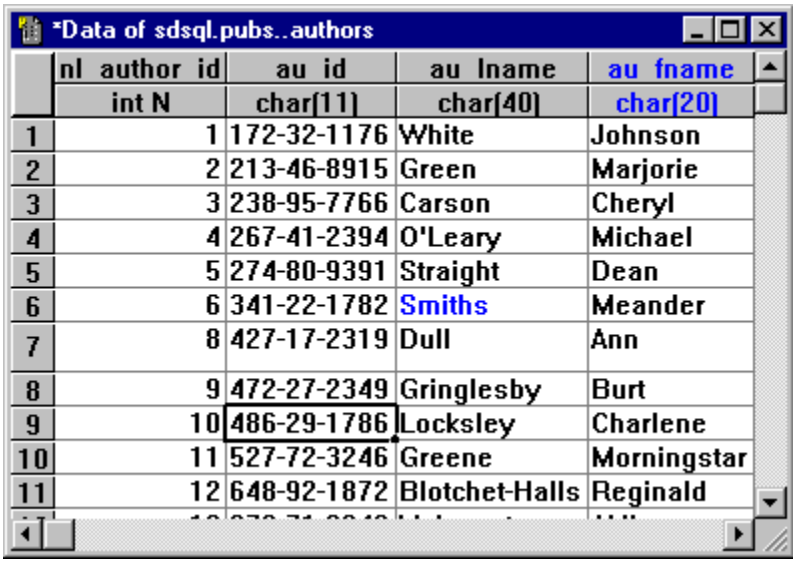
Table Containing SQL Table Data.

Large Tables.

Large SQL Tables may sometimes have to be truncated to fit in the view window. The size limit is set in the Memory Usage field in Options. If the table is not loaded completely, the status bar of the application contains an asterisk (*) and the following ratio

<number of loaded records>/<total number of records in the table (meeting any WHERE/autofilter that may have been set)>

In such case it is advisable to use the Edit WHERE function to specify a condition excluding those records of the table you do not wish to be displayed.



	nl_author_id	au_id	au_lname	au_fname
	int N	char(11)	char(40)	char(20)
1	1	172-32-1176	White	Johnson
2	2	213-46-8915	Green	Marjorie
3	3	238-95-7766	Carson	Cheryl
4	4	267-41-2394	O'Leary	Michael
5	5	274-80-9391	Straight	Dean
6	6	341-22-1782	Smiths	Meander
7	8	427-17-2319	Dull	Ann
8	9	472-27-2349	Gringlesby	Burt
9	10	486-29-1786	Locksley	Charlene
10	11	527-72-3246	Greene	Morningstar
11	12	648-92-1872	Blotchet-Halls	Reginald

Window Title Bar

The window title bar contains the following description of the SQL Table whose data is displayed in the window: Data of <server>.<database>..<table>. If the window only displays some part of the table which satisfies the WHERE condition, the condition is also displayed in the window title bar.

Table Header

The first row of the table (with gray background) contains the names of columns of the displayed SQL table.
If Show Columns Datatype is selected in Options the table header also contains a second row showing the data types of the table columns. Each data type may be followed by the following characters:

- D - if a Default is set for the column.
- N - if NULL values are permitted for the column.

0 Table Row Numbers

1 Rows in an SQL table are normally not numbered but to achieve greater transparency and to enable row height changes and row highlight the table rows are numbered in the view.

2 Data Sorting

3 Table data can be sorted by any column - by double clicking on the column header. Numeric columns are sorted by size, date columns by date succession and text column by alphabetical order. The records are initially sorted upwards. Double click on the column header to sort the records downward. The sorting is static, i.e. identical values retain the order they had in the unsorted table (in other words, rows with equal values in the sorted column are sorted by the last sorted column). This works for up to eight levels which means that table data can be sorted by eight columns at once. If you are sorting a table containing a telephone directory, first sort it by the address column, then by the first name and at last by surname to obtain a table sorted primarily by surname, secondarily by the first name and then by address.

4 Highlighting rows, columns or blocks of cells

5 Certain functions (e.g. clipboard operations) work with highlighted rows/columns/cells.

6 Highlight Rows by clicking the left mouse button on the row number. To highlight several rows hold Ctrl and to highlight a series of rows hold Shift - the standard Windows key usage.

7 Highlight Columns by clicking the left mouse button on the column header. To highlight several columns hold Ctrl and to highlight a series of columns hold Shift - the standard Windows key usage.

8 Highlight a block of cells by dragging the mouse while holding the left mouse button pressed, or by holding Shift and using the arrow keys.

9 Highlight the whole rows in which a block of cells is highlighted by pressing F6. Because it is sometimes inconvenient to use the mouse to highlight rows, use the keyboard to highlight a block of cells including the rows you would like to highlight and press F6.

10 Press Ctrl A to highlight all rows and all columns.

11 Change the width of columns and the height of rows same as you would in Windows. In addition you can double click on the line separating the columns (in the header) to set the column width to the default value (the width of the widest text in the column).

12 Change Data in Table

13 You can change the value of records in the table (UPDATE), delete the highlighted row using the Del key (DELETE) and modify the values of empty cells - so adding records to the table (INSERT). To navigate in the text use the cursor (thick-bordered rectangle) which can be moved using the arrow keys, Tab and Shift Tab, or click on the desired place with the mouse. Press F2 to switch the cursor into edit mode and change the cell content. The content of the modified cells is displayed in blue color to make previous changes easily visible. Any modifications to the table are not made directly to

the SQL table. Select Update changes in the shortcut menu to apply all modifications. After the update all tables are reloaded from the database.
14 Tip: Non-string fields are entered into the generated INSERTs and UPDATEs exactly as typed by user, so it is possible to use system functions (e.g. GetDate() in date columns or NewId() in uniqueidentifier columns) instead of their values.

15 Long Text Fields

16 The data in text or image columns is sometimes extensive. SQLing only loads the first 4Kb of such data items. Truncated cells (i.e. those larger than 4Kb) are displayed in the table using gray color. If the cursor is located on such cell, pressing F3 will load the whole content of the cell and the area of the cell will be temporarily (until you press one of the following: Esc = undo, F3 = disable large view, F2 = stop editing field) enlarged to full window so that you can comfortably view/edit long text.

17 Developer Editor in table fields

18 It is common that some table fields in databases contain scripts in some language, usually Transact-SQL or HTML. These scripts are edited more comfortably in an editor which offers context-sensitive coloring of text, online help, indenting, write-ahead etc. Such editor is opened instead of the standard editor when you press F3 if the field text begins with:
19 1. The text <HTML>.
20 2. CAPITALIZED letters.
21 HTML and T-SQL text respectively will be syntactically colored. In any other cases the text is interpreted normally. You can change the colors assigned to the syntactical word groups in the Colors field of the Query Editor popup menu.

22 Modification Flag

23 If you change any data in the table an asterisk (*) will be displayed before its name in the title bar.

24 Replicate filled cells.

25 When inserting new rows into the table which are similar to any existing rows you can highlight a block of cells in the row and drag the ball in the lower right corner of the text cursor to the lower right corner of the newly created rows. The highlighted cells are replicated into all the newly created rows.

26 Clipboard + import/export from MS Office

27 You can use all clipboard functions while editing the table:

- Cut - Ctrl X
- Delete - Del
- Copy - Ctrl C
- Paste - Ctrl V

Use the Clipboard to copy data between SQLing and MS Office tables.

Change column order.

Use the drag and drop technique to move columns within the table:

1. Highlight the column (or more adjacent columns) which you wish to move, press and hold the left mouse button.
2. Move the mouse pointer to the column before which you want to insert the moved column.
3. Release the left mouse button.

Shortcut menu

Other operations available for tables are included in the Shortcut Menu .

- Related Table - display the relational records of the table related to the current table

0 The following two items are only displayed in the menu if the menu was invoked over the foreign key column of a simple (not combined) relation. The table on the primary side of the relation is called a counter.

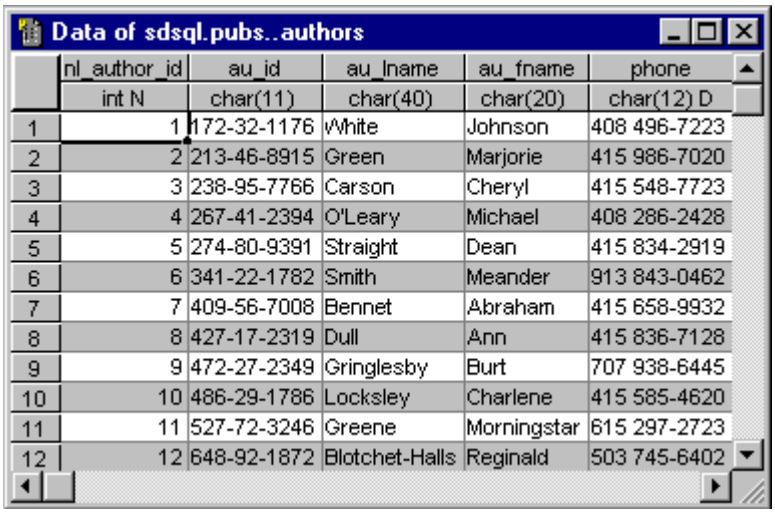
- Open <table name> - Opens a counter table named <table name>, related to the current column.
- Expand <table name> - replaces (mostly numeric) values in the current column by text values of the related counter.

- Edit WHERE… - filter the displayed data
- Refresh - reload table data
- Edit Notes - create table notes
- View All Notes - display all table notes
- Save… - saves the table data into a file as tab-delimited text.
- Save Table Options - saves the current table appearance (autofilter, where, sort, expand, order and width of columns) as its default appearance.
- Record Script… - log table changes
- Update Changes - apply the modifications made to the SQL table data
- Field Chooser... - this function chooses the list of loaded and therefore displayed columns for the selected table. Used with OLE DB connect for tables containing more than one text/image column. OLE DB cannot load all columns of such tables.
- Auto Filter <column name> - filter column data conveniently. This function is only available in the menu if the mouse pointer is located over a column header when the menu is invoked.

0 Table Appearance

1 You can set the following elements of table appearance in Options.

- Horizontal Stripes - if selected, alternate table records will be displayed against a darker background.
- Change Font - opens a dialog box where you can set the font used in the tables.

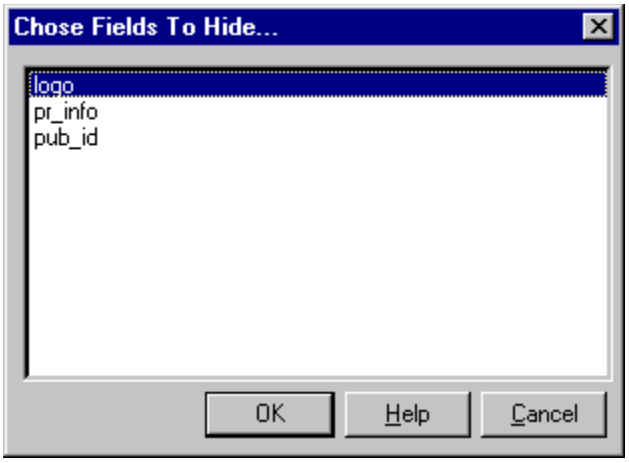


Data of sdsql.pubs..authors

	nl_author_id	au_id	au_lname	au_fname	phone
	int N	char(11)	char(40)	char(20)	char(12) D
1	1	172-32-1176	White	Johnson	408 496-7223
2	2	213-46-8915	Green	Marjorie	415 986-7020
3	3	238-95-7766	Carson	Cheryl	415 548-7723
4	4	267-41-2394	O'Leary	Michael	408 286-2428
5	5	274-80-9391	Straight	Dean	415 834-2919
6	6	341-22-1782	Smith	Meander	913 843-0462
7	7	409-56-7008	Bennet	Abraham	415 658-9932
8	8	427-17-2319	Dull	Ann	415 836-7128
9	9	472-27-2349	Gringlesby	Burt	707 938-6445
10	10	486-29-1786	Locksley	Charlene	415 585-4620
11	11	527-72-3246	Greene	Morningstar	615 297-2723
12	12	648-92-1872	Blotchet-Halls	Reginald	503 745-6402

0

Table Field Chooser



This dialog window is used to select the column (columns) which will not be loaded or displayed when the table is next used. If the table properties are saved using Save Table Options the selected columns will never be loaded and displayed in the future.

Save Table Options

Select Save Table Options from the shortcut menu to save table appearance information and the
user name (NT login) into the SQLing database. When the data table is next loaded by the same user the table will have the saved appearance.

Table appearance information includes:

- Autofiltre for all columns
- Expand of the columns
- WHERE table condition
- Column order (if changed)
- Column width (if changed from the default best to fit)
- Table sorting information
- The columns which should not be loaded (Field chooser)

0 Ignore the saved appearance
1 If a saved appearance exists for a table but you would like to open the table with the default appearance click on the Data button in the Browser while holding Ctrl.

2 Saving information
3 Information is saved in the table named SQLing..tabexpand.

Expand <table name>

Counters

0 In relational databases use is often made of tables (counters) which were created by the decomposition of relations to increase spatial efficiency. Counters usually contain a numeric column and a text column where the text represents numeric values which are used in other tables bound to the counter by an N-1 relation.

1 If a counter is bound to the current column the shortcut menu invoked over this column contains an item named Expand <counter name>. Select this item to open a dialog box containing a single combo box listing the names of all columns of the counter. Any column in the combo box is automatically selected if its name contains the substrings ”name” or ”description”. The user should browse through the list in the combo box to select the text column of a counter. Press OK to confirm the selection and the values in the active table column will be replaced by the values from the selected column of the counter. From now on values in the current table column will be edited in a combo box (not an edit box) which will offer all text values of the descriptive column of the counter.

2 The best way to explain this is by means of an example:

3 We are going to expand column object_id in this table

Data of sdsql.patrik..attribute

	report_id	attribute_id	object_id	obj_attr_name
	int	int	int N	char(255)
1	2	0	2	start
2	2	1	2	totnum
3	2	2	2	status
4	1	0	1	db
5	1	1	1	id
6	1	100	1	id
7	5	0	9	au_lname
8	5	1	9	nl_autor_id
9	5	2	10	fname
10	5	3	10	minit
11	5	4	10	nl_pub_id
12	5	5	10	job_lvl
13	5	6	11	nl_job_id
14	5	7	11	min_lvl

4

5 The counter

6

Data of sdsql.patrik..object

	object_id	report_id	type	table_name
	int	int	int	char(255)
1	1	1	0	keysadd
2	2	2	0	spt_committab
3	6	1	0	report
4	7	4	2	keysadd
5	8	4	0	graph
6	9	5	0	authors
7	10	5	0	employee
8	11	5	0	jobs

7 The result

Data of sdsql.patrik..attribute

	report_id	attribute_id	object_id	obj_attr_name
	int	int	int N	char(255)
1	2	0	spt_committab	start
2	2	1	spt_committab	totnum
3	2	2	spt_committab	status
4	1	0	keysadd	db
5	1	1	keysadd	id
6	1	100	keysadd	id
7	5	0	authors	au_lname
8	5	1	authors	nl_autor_id
9	5	2	employee	fname
10	5	3	employee	minit
11	5	4	employee	nl_pub_id
12	5	5	employee	job_lvl
13	5	6	jobs	nl_job_id
14	5	7	jobs	min_lvl

8

Saving Data to Disk

0 The data of the active table can be saved into a .txt file as tab-delimited text which can be imported into MS Excel.

1 Saving Table Data into a File

0 1. Open the Shortcut menu.

1 2. Select Save from the Shortcut menu.
A dialog box entitled Save as appears.

0 3. Enter the file name, select the disk drive and the folder where you would like to save the file.
The extension .txt will be appended to the saved file.

The Related Table Menu

The Related Table function is used to show the data of tables which are related to the current table. Only those records of the related tables are displayed which are bound to the selected records of the current table.

The Related Table Menu contains a submenu with items for every table connected to the present table by a relation. If at least two tables connected to the current table by a 1->n relation exist the menu will include the command All 1->n (the same is true for n->1). Select any of the table names to display the records of the selected table which are related to the selected records of the current table. Select All 1->n (n->1) to display the records of all tables connected with the current table by a 1->n (n->1) relation which are related to the selected records of the current table.

SQLING opens a window entitled Data of *server_name.database_name..table_name WHERE foreign (primary) key value=primary (foreign) key - the value of the selected records of the current table*, which contains the data of the relational table referring to the previously selected row (rows) of the current table.

Applying Changes Made to a Table on the SQL Database

If you modify table data a modification flag (an asterisk, *) will be displayed in the top left corner of its title.

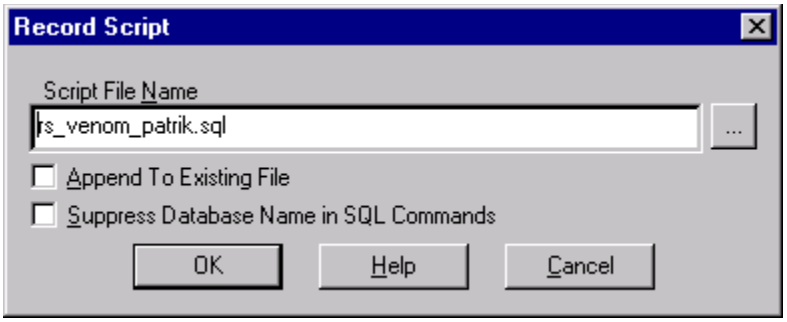
0 Select Update Changes in the shortcut menu to save the modifications on the SQL server. Not all changes made in the table are always stored. If any of your modifications interfere with the referential, domain or entity integrity of the database, an error message is displayed and the changes are not saved.
1 When the SQL table has been updated it is reloaded. This will cause all blue (modified) cells to be displayed as black and all unsuccessful changes to be reverted. Those rows which could not have been stored in the SQL table are displayed red. Correct any errors and run Update Change again.
2 The function removes the modification flag.

Record script

Select Record script to save the changes you made to the database on one SQL server and implement them on a database on another SQL server, provided the data on both SQL servers is identical.

Saving the changes you made to the database content in the window entitled Data of *server_name.database_name..table_name*

1. Select Record Script from the Shortcut menu.
   A dialog window entitled Record Script appears



0 2. Enter the file name into which the script will be generated in the field named Script File Name.

1 3. Select Append to Existing File if you wish to append the new script to the end of an existing file.

4. Select Suppress Database Name in SQL Commands if you do not wish the tables referred to by the script to contain the database name in their identifiers.

0 5. Make the necessary changes to the table data (on the SQL server).

1 6. Click the right mouse button and select Update Changes from the Shortcut menu.

7. Click the right mouse button and select Stop Recording from the Shortcut menu.
   This will save the file where the changes are scripted.

0 Implement the modifications made in the database on another SQL server

1 1. Click on the Query Editor ( ) button in the main application toolbar.
   A window entitled Query Editor is displayed.

0 2. Click on the Load ( ) button in the query window toolbar.
   A dialog box entitled Open appears.

3. Select a previously saved script.

4. Click on Execute () in the Query Editor window.

View All Notes

Notes to the current table can be displayed using the command View My Notes located in the shortcut menu. A Results window containing all notes that were ever connected with the table is displayed:

The first line of each table note contains NT-login of user that created the note.

Editing Notes

Select Edit Notes to edit the notes connected with a table.

Editing the notes to the selected table

1. Select Edit Notes from the Shortcut menu.
   A dialog box entitled Edit Notes for *table_name* appears

0 

1 3. Write a new note or change the existing note.

0 4. Click on Save () in the toolbar. The note will be saved without prompting.

Refresh

The data displayed on the screen for a longer time may become obsolete when you are working with a table of a database used by several users simultaneously.

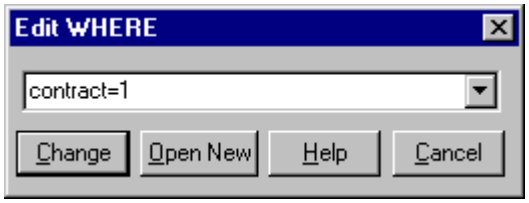
Select Refresh in the shortcut menu to reload the data from the SQL server.

Edit WHERE

Select the Edit WHERE command from the shortcut menu to change the WHERE condition valid for the table records displayed in the window.

Changing the Where condition

1. Select Edit WHERE from the Shortcut menu.
   A dialog box entitled Edit Where appears



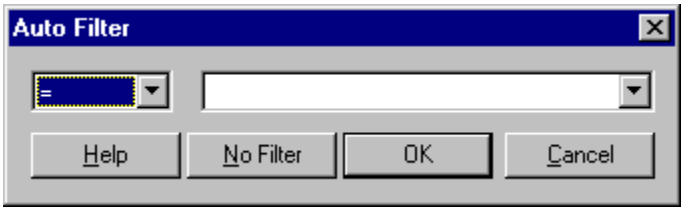
0

1 2. Enter a new WHERE condition or select a cached WHERE condition from the combo box.

0 4. Do the following:
   a) Click on Change to display the data satisfying the new condition in the currently opened table window.
   b) Click on Open New to display the data satisfying the condition in a new window.

Auto Filter

Is a convenient way of filtering the table data. The function must always be applied to a table column. This dialog box appears when the function is called.



The dialog box contains two combo boxes

- Operators - (=, <, >, <=, >=, <>, in, like) select the operator you wish to use in the filter.
- Values - this is a sorted list of all values available in the selected column at the time when the function was called. Select one of these values or enter a new value.

Set the values in both combo boxes and press OK. The table is reloaded but only the records which satisfy the filter condition are displayed. The header of the selected column is colored blue to indicate that the table records are filtered by the selected column. You can set filters for more than one column.

Remove Filter.
If you wish to remove filtering from a column press No Filter and the content of the above combo boxes will be disregarded.

SQLing

Installation

SQLing: Examples of Use.
What Is SQLing
What You Can Do With SQLing

When launching the application, you can specify Command Line parameters .

Main Application Window
Elementary Application Principles

This help and the manual to the SQLING application are intended for experienced users of the MS SQL Server. It is assumed that you are already familiar with the basics of database design and administration. (For more information on database administration please refer to the Microsoft SQL Server Books Online).

What Is SQLing

SQLING is a database administration tool created out of the need to facilitate the development of projects which involve more than 1000 SQL tables, 800 table relations, 1500 stored procedures and gigabytes of critical data. The whole database structure was designed and administered in SQLing from beginning to the end.

SQLING can be used to:

-Browse, edit and print relational models
-Create and modify database structure
-Browse and edit data tables
-Create, compare, edit and debug procedures
-Compile database documentation
-Transfer databases wholly or partially, online or offline
-Employ the client/server architecture
-Compare databases
-Easily build queries and reports
-and much more

What You Can Do With SQLing

Use the Query Editor with the context color and debugging functions

The Query Editor window can display up to ten syntactical word classes (one of which is User defined) in different colors to help you find your way through the text. The editor also contains an integrated debugger which helps to eliminate errors in procedures.

Visualize the relational models of databases

SQLING enables you to browse and edit relational models (i.e. database structure and relations between tables). The Relational Model window displays tables with a defined primary key or those related to a primary key table. SQLING can create, print and edit relations between tables. In addition to that you will be able to understand the structure of large databases with many tables and relations easier if you print the relational model on several pages and tile them into a large, wall-hung poster. For more information, go to the section Relational Model.

Define views to work more efficiently

If the database contains too many tables it is useful to open different views in the Relational Model window, each of which only shows the selected tables. This will enable you to browse and edit relations between tables much more efficiently. For more information go to the section Select, Deselect Commands.

Compare and synchronize structure and data of two databases

You can compare the structure of tables and other objects in a database, as well as the content of tables selected specifically to discover differences between two databases. You can turn on automatic generation of balancing scripts which will synchronize the structure or content of databases. For more information, go to section Compare.

Modify database structure comfortably

When modifying a database on an SQL Server it is convenient to create a working, downsized database which is structurally identical with the database located on the SQL Server. You can make as many modifications to this working database as you wish before you decide to upsize it. The changes will then be implemented on the actual database located on the SQL Server. Modifying the database on the SQL Server by first changing its downsized version has two advantages:

1. The implementation of any changes on the SQL Server database may take a long time. When you use a working database you can make several changes and then upsize the working database to the actual SQL Server database. You may launch the database upsize process and then leave. This will save you the time you would have to

spend waiting for the implementation of the changes.
2. The actual database on the SQL Server remains consistent - it will transfer from the original consistent state to the new consistent state during the Upsize process which does not last very long.

For more information, refer to the Downsize … or the Upsize … command.

Avoid recurring operations by activating automatic tools for database structure and content modification

The Save Downsize command is used to save the changes you made to the database structure on one SQL Server and their implementation on another SQL Server. Database structure on both SQL Servers must be identical. For more information see the Save Downsize ….

The Record Script command is used to save the changes you made to the database content on one SQL Server and their implementation on another SQL Server, provided that the data on both SQL Servers is identical. For more information see the Record Script ….

Make substantial modifications to the database structure - see Design Table

a) Change the sequence of attributes in a table.
b) Change the data type of an attribute.
c) Delete an attribute located in the middle of a table.
d) Add an attribute to any part of the table.

Quickly diagnose a table

You can quickly determine the number of records and the size of a table. Find more information under the Select count(*) ….

Transfer a database from one SQL Server to another without an online link

You can export all or some of the objects of the current database on one SQL Server into a target database located on another SQL Server which is not connected via LAN (Local Area Network) or WAN (Wide Area Network). Portable data storage media is used to transfer the information offline. Find more information under the OFF-Line Export and OFF-line import commands.

Increase database transparency by an added layer of comments

Add comments to table relations, database objects, tables and their attributes. (Find more information under Edit Notes, Editing Table Relations).

Create database documentation

Quickly and comfortably create database documentation containing the comments you added to database objects and table content. Find more information in section Make Document.