

SQLBuilder Contents



The SQL builder allows you to visually create and run a SQL Select, Update or delete query and to run that query.

Select queries show their results in the result table. This result table can also be saved to disk (see [Saving Result Table](#)).

Update and Delete type queries use the result table to preview the rows that will be affected by the statement. The changes are not actually made until you click the "Post Changes" button

To create a query you simply;

- a) Select the tables you want to query
- b) Select the fields you want to see or update
- c) Set the criteria for the query

If that sounds a bit daunting at the moment you can use the Query Wizard to set-up the initial query for select type queries for you.

Select any of the topics below for more specific help

[The main screen](#)

[Adding Tables](#)

[Joining Tables](#)

[Selecting Columns](#)

[Setting Query Criteria](#)

[Using the Query Wizard](#)

SQL Builder Main Screen

The main screen of SQL builder can be divided into three main parts. Click the highlighted text for more detailed information;

1) The Button Bar

The button bar allow for quick selection of any of SQLBuilder's main functions such as selecting the type of query, adding and removing tables, defining calculated fields and editing table joins.

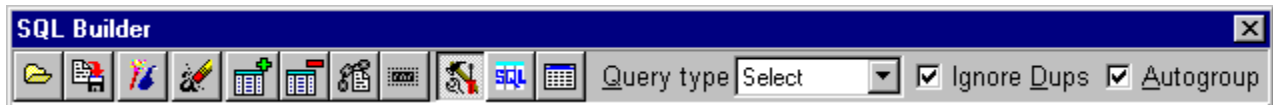
2) Selected Tables Panel

This panel contains a field grid for each table selected. Any individual field in any of the table grids may then be selected to be included in the query.

3) Selected Columns Panel

This panel contains a grid that has a column in the grid for each column from the table that has been selected. Criteria, sort order grouping are then set for each column as necessary.

The Button Bar



Overview

The button bar allows quick access for all common functions. See detailed description below of what each item does.



Opening a saved query

This button allows you to open a previously created query.



Opening a saved query

This button allows you to save the current query.



Table & Field Wizard

This button starts the Table & Field Wizard to lead you through table and field selection step by step.



Adding Tables

The first part of building any query is to select the tables that are to be queried. As each table is selected a grid is shown displaying the fields available in that table. Clicking this button will display a file-open dialog listing all tables in the current database alias



Removing Tables

To remove a table first click on the grid for that table to select it. When a table's grid is removed any selected fields and table links for that table are also automatically removed.



Clearing Query

This button removes all tables and fields from a query.



Linking Tables

Display all links set between the various tables in the query.



Editing Defined Fields

Displays a dialog box that allows the user to add a user-defined calculated field to the query.



Show Visual Query

Displays the standard SQL Visual Builder window



Show SQL Text

Displays the SQL query text that would be generated if the OK button is clicked with the current settings.



Show Query Results

Attempts to open any databases used and run the visual query currently being designed. If all necessary databases can be opened and the query is valid for those databases the results are displayed in a read-only table format. This function only operates with "Select" queries for the data safety reasons.

Query Type

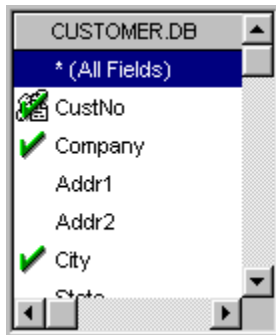
Whether any query type can be selected or not depends on the internal settings given to SQLBuilder by the application.

See [Query Types](#)

Remove Duplicates

When checked the "DISTINCT" reserved word is added to the query to remove any duplicate rows on a [Select](#) query.

Selected Tables Panel



Overview

This panel shows a grid for each table that has been selected for the query (see [Selecting Tables](#)). Each grid contains a list of columns for that table. Clicking the right mouse button over the grid will display the [Popup Menu](#).

The table-grid currently selected has a different color background to any other table-grid. These table-grids control which columns are selected (see [Selecting Columns](#)) and how tables are joined (see [Joining Tables](#)).

Selected Columns Panel

* SELECT *	EMP_NO	FIRST_NAME	LAST_NAME	TotalOrders
Table	EMPLOYEE	EMPLOYEE	EMPLOYEE	User Defined
Field Type	SmallInt	String	String	Currency
Show Field	True	True	True	True
Sort Type				
Where value is	<input type="text"/>			
Or value is				
Group Field	True	True	True	False
Having				

Overview

This panel contains a grid that has a column in the grid for each column from the table that has been selected. The rows that are displayed in this grid vary according to what [Query Types](#) is being generated. It is in this grid that you set the search criteria, sort order and grouping as necessary for each column. See [Setting Query Criteria](#).

Changing Column Order

To change the order of columns in the query simply drag and drop the columns displayed in the grid. ie. To change the order, click and hold down the left mouse button on the title of the column you want to move, drag it to where you want it and release the button.

SELECT Queries

Overview

Select queries are queries that return rows from the tables that match the criteria entered. The row returned may not necessarily contain any actual data from the table it only be a total, average or count but it is still a row.

Where more than one table is used in a query they are usually joined by a common field such as a customer account code or an order number etc. There are various kinds of joins see [Joins](#) for more information.

Syntax

The basic Select Query has the following format, note that only the "SELECT" and "FROM" clauses are compulsory and that the reserved word "DISTINCT" may be optionally be used to remove duplicate rows from the query results;

```
SELECT      (DISTINCT) select-list
FROM table-name(s)
WHERE       criteria
GROUP BY    column-name(s)
HAVING      criteria
ORDER BY    column-name(s)
```

Examples

Example 1;

```
SELECT "species_name", "country"
FROM "Biolife.db"
WHERE "country" = "Australia"
ORDER BY "species_name"
```

Example 2;

```
SELECT "species_name", AVG("length")
FROM "Biolife.db"
WHERE "country" = "Australia"
GROUP BY "species_name"
```

UPDATE Queries

Overview

Update queries are queries that modify the data in the rows in the tables that match the criteria entered. Where more than one table is used in a query they are usually joined by a common field such as a customer account code or an order number etc. There are various kinds of joins see [Joins](#) for more information.

Syntax

The basic Update Query has the following format.

```
UPDATE      table-name(s)
SET  update-list)
WHERE      criteria
```

A update-list is one or more update statements, separated by commas, specifying the updates to be applied. An update statement has the form;
column-name = expression / NULL

Examples

Example 1;

```
UPDATE "Biolife.db"
SET  "country" = "Australia"
WHERE "Common_Name" = "Funnel Web Spider"
```

DELETE Queries

Overview

Update queries are queries that modify the data in the rows in the tables that match the criteria entered.

Syntax

The basic Update Query has the following format.

```
DELETE FROM table-name(s)
```

```
WHERE criteria
```

Examples

Example 1;

```
DELETE FROM "Biolife.db"
```

```
WHERE "Common_Name" = "Loch Ness monster"
```

Where more than one table is used in a query they are usually joined by a common field such as a customer account code or an order number etc. There are various kinds of joins see [Joins](#) for more information.

Joins

The basic table join is a simple link formed between two or more like fields in two or more tables in the "WHERE" clause to create either a "one to many" or "one to one" join.
eg. WHERE "Accounts"."Ac_Code" = "Trans"."Ac_Code"

What sort of joins you can successfully perform is totally dependent on the abilities of your database server. More advanced servers can also perform what are called Inner Joins and Outer Joins joins.

Inner Joins

Overview

Inner Joins are used in the FROM clause to link the columns in the tables in various ways. The word INNER is optional as all joins are assumed to be INNER unless an Outer Join is specified.

Syntax

The syntax of an Inner join is;
FROM table1 (INNER) JOIN table2
ON criteria

eg.
FROM table1 (INNER) JOIN table2
ON table1.IdCode = table2.IdCode

Types of Join

The types of INNER JOIN you can construct are;

Equi-Joins : Links rows based on a common value.

Non-Equi-Joins : Links rows where the column values are not equal.

Self-Joins : Compares values in a single table.

Outer Joins

Overview

Outer Joins are used in the FROM clause to link the columns in the tables in various ways much the same as Inner Joins. Outer joins return every row from one table and a subset of rows from another table. Note that Outer joins also require you to specify the type of join you want.

There are three types of Outer Join;

Left Outer Join

Retrieves all the rows from the left table and any of the rows in the right table that match the criteria.

Right Outer Join

Retrieves all the rows from the right table and any of the rows in the left table that match the criteria.

Full Outer Join

Retrieves all the rows from both tables regardless of any criteria specified in the ON clause.

Syntax


The syntax of an Outer join is;

```
SELECT column-name(s) FROM LeftTable [ LEFT|RIGHT|FULL ] OUTER JOIN table2 ON criteria
```


Example

```
SELECT * FROM table1 LEFT OUTER JOIN table2 ON table1.IdCode = table2.IdCode
```

Selecting Columns

Selecting columns for a query is done by double clicking the left mouse button on the required field in the Table's grid. Any selected field has a checkmark beside it like this . To deselect a column double click it again. Columns may also be dragged and dropped to the selected columns panel.

Joining Tables

Simple joins may be made by dragging from the "join" field from first table and dropping it on the "join" field of the second table. Columns that are joined have a link icon like this  beside them. More advanced join operations and removing joins may all be done using the [Table Join Editor](#).

Popup Menu

Clicking the right mouse button anywhere over the selected table's field list will pop-up a menu. This menu contains items specific to the selected tables.

The menu items are;

Add Table

Remove Table

Join Editor

If the selected table-grid is the "User Defined" table-grid then you will also have the following extra menu items for dealing with user defined fields;


1/ Add Defined Field

2/ Edit Defined Field

3/ Delete Defined Field

Each of these menu items apply to the selected "defined field".

Selecting Tables

To select a table click the  Add Table Button. You will then be shown all the tables available in the selected database alias or path to choose from. Select a table and click the OK button.

The alias box contains a drop-down list of all database aliases registered with the database engine. The alias shown in the box is the current alias that will be used for any add-table operations. Whether you can change the alias or not depends on the internal settings given to SQLBuilder by your application.

The path box displays the last selected local directory (if any). To select a table by path click the button on the right hand side of the box and select a directory from the dialog box displayed. Whether you can change the path or not depends on the internal settings given to SQLBuilder by your application.

Query Types

SQL queries can be generated for three standard query types, they are;

<u>SELECT Queries</u>	: Returns rows that match criteria
<u>UPDATE Queries</u>	: Updates rows that match criteria
<u>DELETE Queries</u>	: Deletes rows that match criteria

Setting Query Criteria

Overview

The criteria for any query are set in the Selected Columns Panel. When you click on any of the editable rows for a column either a drop-down list of possible values or an edit box into which you can type anything and a button will be displayed. Clicking on the button will activate the Expression Builder.

Syntax

When setting search criteria for a query, criteria can be in one of two formats;

1. First non-blank character is an operator ie. <,>,+,-,/,*,=.

eg. > 01/01/95 2. First non-blank character is NOT an operator.

eg. UPPER("Customer.db"."Company") = 'IBM'

The Table Join Editor

The table join editor allows you to edit the join criteria that links each of the tables in the query .

The editor can be used in "Simple" or "Custom" modes. In "Simple" mode the Table and fields to join are simply selected from drop down lists.

- When a table is added TSQLBuilder looks for any natural join and if it can find one it sets the join criteria accordingly.
- When you drag and drop fields to create a join the join criteria will contain the source and target fields.

In "Custom" mode you can set all aspects of the join and use complex SQL expressions as the join criteria.

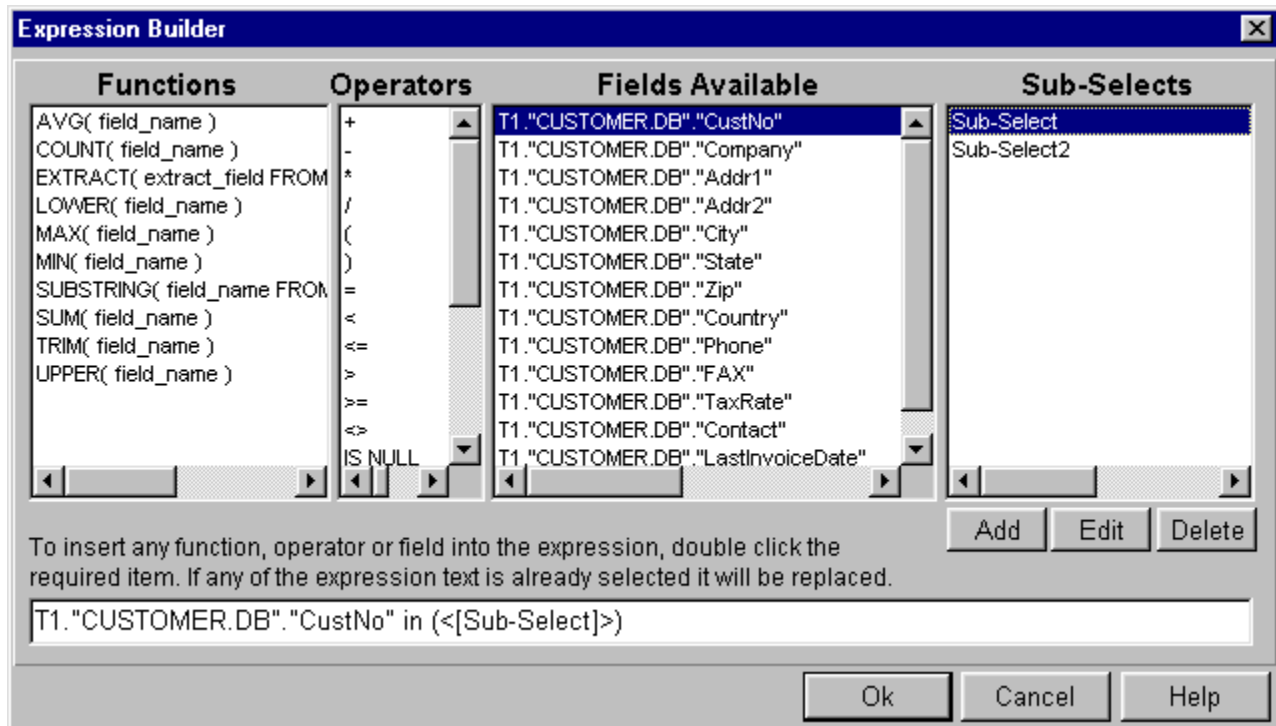
see;

Joins

Removing Tables

To remove any table from the query you must first select the table you want to remove by clicking the left mouse button on it and then clicking the Remove Table button or selecting Remove tables from the pop-up menu.

Expression Builder



Overview

The Expression Builder lets you build complex SQL expressions by pointing and clicking on the appropriate functions, operators and fields all of which are displayed in lists for you to choose from. If any text is highlighted in the expression being built then that text is replaced by the next function, operator or field selected.

SQL Functions

This list contains all the SQL functions that you require to build a query. If a SQL function requires parameters then those parameters that need to be replaced are listed with the function using fairly descriptive names eg. AVG(field_name); In this case you would replace "field_name" with the actual field name for which you want to calculate an average. Which SQL functions are visible to you depends on the settings given to SQLBuilder by the application, some SQL functions may be hidden as they will not work with your database server. As the functions that may be used and the results they give vary from one type of SQL database server to another we will not try to describe each function here, refer to the documentation for your SQL server.

Operators

This list contains all the SQL operators that you require to build a query.

- + : Add
- : Subtract
- * : Multiply
- / : Divide
- (: Opening Parenthesis
-) : Closing Parenthesis
- = : Equals
- < : Less than

<= : Less than or equal to
> : Greater than
>= : Greater than or equal to
<> : Not equal to
IS NULL: Has no value
LIKE : Matches this substring/wildcard
AND : Logical AND
OR : Logical OR
NOT : Logical NOT
|| : String join operator

Field List

This list contains a list of all the fields available to the expression builder to use in expressions. See [Selecting Columns](#).

Sub-Selects List

This list contains a list of all the sub-select statements defined for this query. Sub-Select statements are ordinary select statements that are used inside another select statement usually with the "IN" or "EXISTS" functions. eg.

```
SELECT T1."CustNo"  
FROM "CUSTOMER.DB" T1  
WHERE T1."CustNo" IN  
    ( SELECT  
        ST1."CustNo"  
        FROM  
        "ORDERS.DB" ST1)
```

Click the "Add" button to visually build and test a Sub-Select statement. One of the beauties of using Sub-Selects is that once you build and test the statement you can then simply use it as a predefined function in your main query. Sub-Selects can also be saved to disk in the same way as with any ordinary query for use at some future time in other queries.

The Query Wizard

The Query Wizard takes you through setting up the tables, the joins between them and the fields to be queried in a step by step fashion. You can click the "Next" button to go on to the next step or the "Previous" button to go back at any time.

Step 1. The first screen shows the tables available in the current database alias. If alias changing is enabled you can change the current database alias by selecting a new alias from the drop-down list. Click on the table you require to highlight it and then click the small square ">" button to add it to the selected tables list. As each subsequent table is added the Table Join Editor is displayed with default join criteria for your confirmation or editing. When all required tables have been selected click the "Next" button.

Step 2. The Wizard now displays a list of all the fields available in the tables selected plus any user defined calculated fields you may want to add (see User Defined Fields). The Autogroup checkbox turns AutoGrouping on or off.

Step 3. Click finished and TSQLBuilder is now loaded and ready to either run the query the wizard has built for you immediately or allow you to set the set sort or search criteria to refine the query further.

User Defined Fields

These fields are those fields that are returned by a SQL query that do not actually exist in the database.

For example, running this query;

```
SELECT COUNT(*) as TotalsRows
```

```
FROM "MyTable.DB"
```

generates an user defined column called TotalRows defined by the SQL expression "COUNT(*) as TotalsRows".

What can be defined and calculated in these fields is limited only by your database's capabilities. Add these field by clicking the add user field button to display the User Defined Field Editor. The editor uses the [Expression Builder](#) to help generate the sql expression

AutoGrouping

If Autogrouping only has any effect if the query will contains a summary column in the result.

If Autogrouping is on then all summary columns have "Group By" set to false and all non-summary columns have "Group By" set to true in the generated query.

Saving Result Table

If the result table was generated using local SQL syntax then you can export the query results as a file to disk.

This result table may be exported to Paradox, DBase or ASCII format files. If saved as a Paradox or DBase table then this table can then be queried just like any other table.