

Using TSQLBuilder Version 1.40

Table of Contents

The Button Bar

2

Opening Saved Queries

2

Saving Queries

2

Table, Joins and Field Selection Wizard

2

Adding Tables

2

Removing Tables

2

Table Joins

2

User Defined Fields

2

Display SQL Builder

2

Display SQL Text

2

Display SQL Query Results Table

2

Query Type

2

Ignore Dups

2

Autogroup

3

The Selected Table Grids

3

Adding joins between table fields

3

The Table Joins Editor

3

Adding joins between field of different tables

3

The Selected Fields Grid

4

Table

4

Field Type

4

Show Field

4

Sort Type

4

Where value is

4

Or value is

5

Group Field

5

Having

5

Set To Value (Update Queries Only)

5

The Expression Builder

5

The Query Tables and Fields Wizard

6

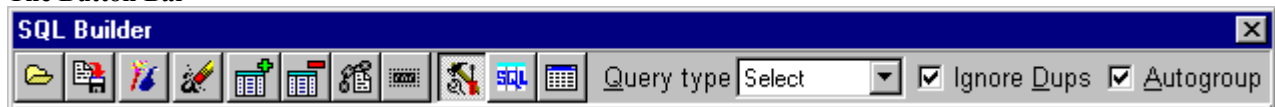
General Hints

6

How to contact the author

7

The Button Bar



Opening Saved Queries

This button displays the open file dialog box to allow you to select a previously created and saved Tsqlbuilder query to open. The current query will be replaced by the opened query.



Saving Queries

This button displays the save file dialog box to allow you to save the current query to file.



Table, Joins and Field Selection Wizard

This button displays the Table, Joins and Field Selection wizard. It takes the user through setting up the initial query 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.



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 joins for that table are also automatically removed.



Table Joins

Display all table join details.



User Defined Fields

Displays a dialog box to add a user defined field to the query.



Display SQL Builder

Displays the visual SQL query builder panel



Display SQL Text

Displays the generated SQL text in the Query Editor panel. If the appropriate option is set then the query may also be modified.



Display SQL Query Results Table

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 obvious reasons.

Query Type

Selects the type of query that will be generated from a list. The query type also determines which rows are displayed in the Selected Fields Grid.

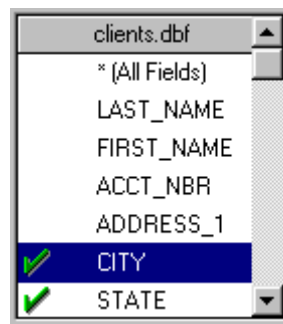
Ignore Dups

Adds the reserved word "DISTINCT" to a "Select" type Query to remove any duplicate rows.

Autogroup

If checked a query that contains summary columns will be grouped on all non-summary columns.

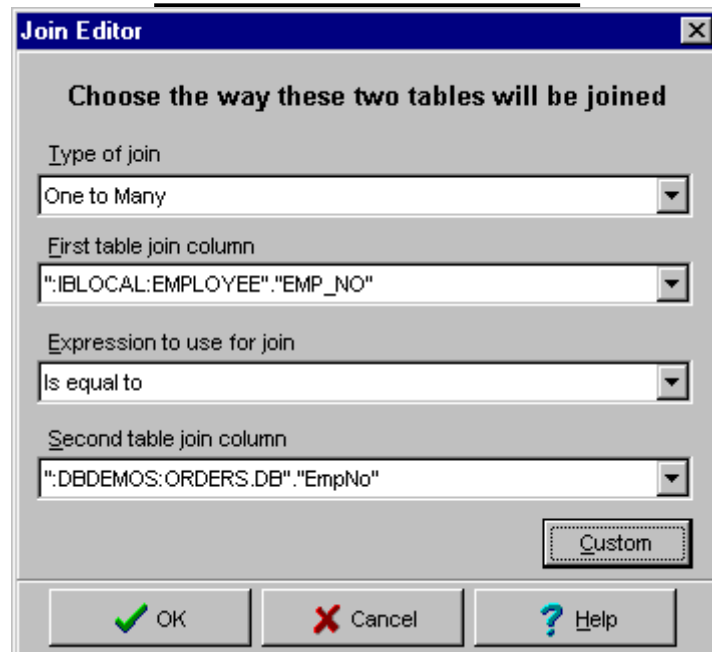
The Selected Table Grids



Adding and Removing Fields from the query

The table grids display a list of fields available for each selected table. To add a field to the query simply double click the required field (If at least one field is already selected you can drag fields to the Selected Fields grid). A green check mark will appear showing that the field is selected and a column for that field will appear in the Selected Fields Grid. To deselect a field simply double click it again.

The Table Joins Editor



Adding joins between table fields

Whenever you add a table to the query the join properties editor will then displayed and if there is a natural join between this and the previously added table then information will already be setup for you to confirm.

You can select different join fields or even enter complex sql92 type joins by clicking the “custom” button and filling in the join type and criteria.

To add a join between fields in two tables after they are already in the query simply use the left mouse button to drag the first of the fields to be joined to the other field and drop it.

Note: Many join types are not defined in standard SQL89 but many different vendors have implemented these and other extensions in various non-standard ways. For more detail on how joins work with your database refer to your database manager documentation.

The Selected Fields Grid

* 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				

Each column in this grid represents a field that is either user defined or has been selected from the available tables. The order of the columns determines the sort and grouping orders where applicable, any column can be dragged to whatever position you like.

Some columns can not be edited;a) The values in the first two rows (table and field name) cannot be edited.b) If grouping is used and this is not a grouped field then the “Where value is” and “Or value is” rows are protected and “Having” is open, if it is a grouped field then the reverse applies.

To edit any of the other values click on the cell containing that value. If the possible values are true or false then a drop down list will appear. In other cases a edit box and Expression Builder button appear, if the criteria is simple just type it in, else click the button and let the Expression Builder do the work for you.

Table

The table name of the selected field or “USER DEFINED” if field is user defined.

Field Type

The data format used to store the data for this field.

Show Field

If true then the field is included in the select clause of the query

Sort Type

Can be ASC (ascending), DESC (descending) or not sorted on this field.Note: Column order also affects the sort. Columns are naturally sorted from left to right.

Where value is

Criteria for the Where clause of the query. Criteria can be in one of two formats;;

- First non-blank character is an operator ie. <,>,+,-,/,*,=. eg. > 01/01/95 entered in the “DateOpened” column of a table.Use this format to simply compare the column’s value to the entered value (note: quote characters are not necessary they will be added automatically when needed).
- First non-blank character is NOT an operator. Enter the full expression you want to appear in the “Where” clause of the query of **any** selected field’s column.eg.
UPPER(“Customer.db”.”Company”) = ‘IBM’

Or value is

Criteria for the OR clause of the query. Same format as “Where Value Is”;

- First non-blank character is an operator ie. <,>,+,-,/,*,=. eg. > 01/01/95
- First non-blank character is NOT an operator.
eg. UPPER(“Customer.db”.”Company”) = ‘IBM’

Group Field

If true then query will group data on this field. If the Autogroup box is checked then this will be handled for you when the query is viewed or run.Note:You should group all table columns whenever your query returns one or more columns created by a summary operator (Sum, Avg, Count etc.) and one or more table columns.

Having

If “Group Field” is true this restricts the groups returned by the query.

Same format as “Where Value Is”;

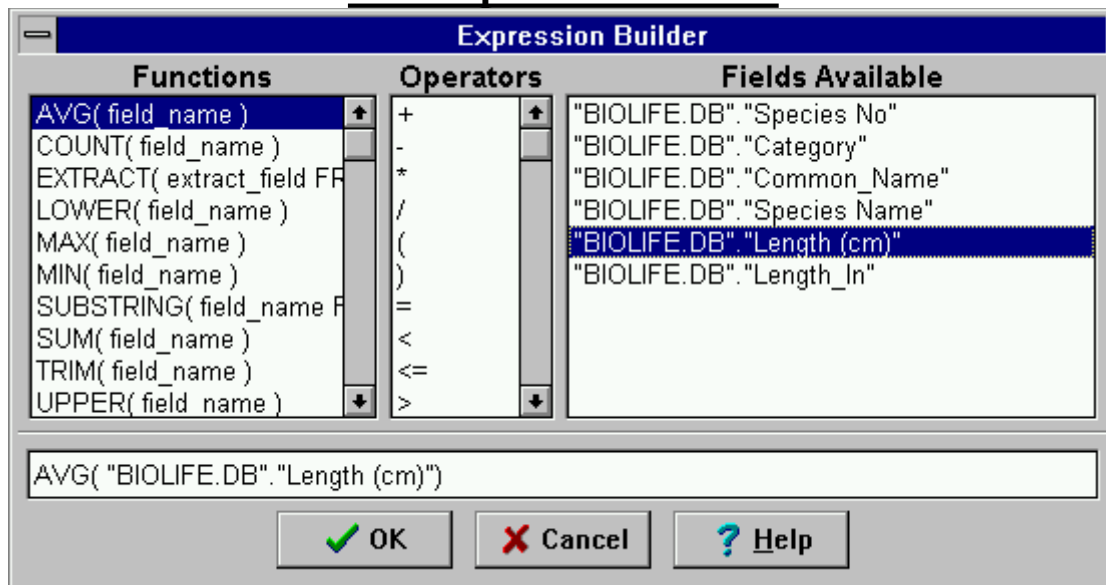
- First non-blank character is an operator ie. <,>,+,-,/,*,=. eg. > ‘01/01/95’

2. First non-blank character is NOT an operator.
eg. UPPER("Customer.db"."Company") = 'IBM'

Set To Value (Update Queries Only)

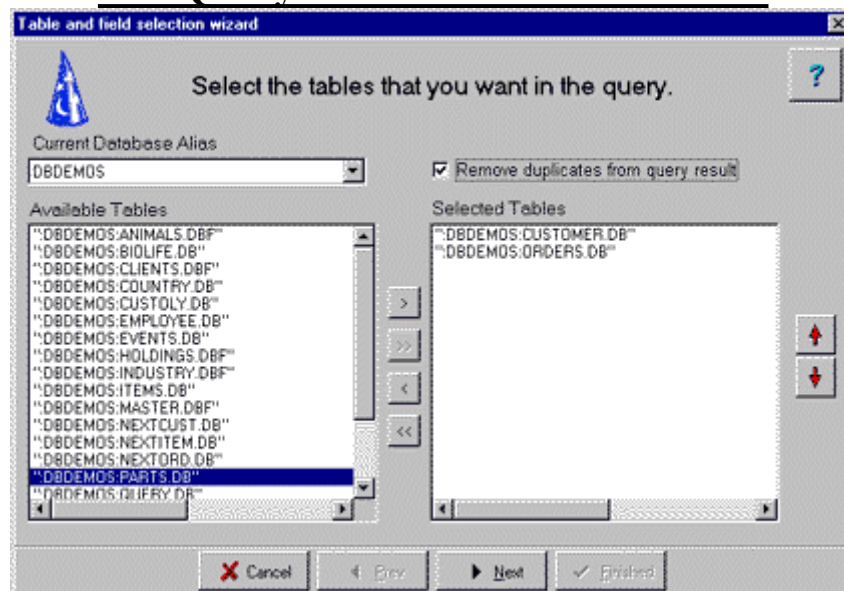
Sets the value for that field for all rows that meet the query criteria.

The Expression Builder



The Expression builder allows you to assemble fairly complex expressions simply by pointing and double clicking. Double clicking any function, operator or field will add the item to the expression. If any part of the expression is selected then the selected portion will be replaced by the double clicked item. In the example above, starting with an empty string, double clicking on AVG(field_name) enters this into the expression with “field_name” selected. Then double clicking on “BIOLIFE.DB”.“Length (cm)” replaces “field_name” to give the final result.

The Query Tables and Fields Wizard



The Wizard takes the inexperienced user through the steps in creating a valid SQL query one at a time. The user simply picks what they want and then clicks the “Next” or “Previous” buttons to go forward or backward through the process.

The first screen displays a list of tables available in the selected database, as tables are selected the joins between those tables are set up (automatically where possible).

When the next button is clicked the second screen with a list of fields available in the selected tables and a list of fields already selected is displayed. Select the required fields and set their order as required with the up and down buttons. If you want to include a user defined calculated field then click the add button, complete the details and it will be added to the selected fields list. The first stage of building the query is now complete, clicking the finished button will return you and the query you've created to TSQLBuilder. The query can now be either fine tuned or run as is.

General Hints

Not all SQL databases are equal, each manufacturer has interpreted the "details" of the various SQL standards in their own way. Many perfectly standard legal queries may fail (especially local SQL) with an error of "Capability not supported" and what works with one database may not work identically without modification with another. The crucial phrase in that last sentence is "without modification", in all likelihood there is a way to perform the same operation but some detail will vary between the two, the most common problems appear to be;

- handling of quote characters This seems to vary more than any other detail from one database to the next. Use the SQL Editor panel to remove/add any quote characters and email the author giving details of the problem and database type.
- heterogeneous queries A TQuery component will only open the database listed in the TQuery.DatabaseName property, if any other database/alias is involved in the query it must be opened manually. A list of all the databases used in the query is kept in the UsedAliasList property. The Results panel of SQLBuilder uses this property to automatically open and close databases as required to run a query, Registered users can see how all this works by looking up the PrepareAndRunQuery procedure in sqlbuild.pas
- grouping with summary functions If you want to extract a data set that uses a summary function (eg. listing a count of orders taken by each employee) then, in general, you have to group all the fields that make up the summary (ie. everything except the actual summary field) or you will get the dreaded and uninformative Delphi 1.x "Capability Not Supported" error.

How to contact the author

Jeffrey Cooke
P.O. Box 106
Kemps Creek
NSW 2171
Australia

compuserve: 100026,3107
internet: 100026.3107@compuserve.com
Fax: 61 47 749086

© Jeffrey Cooke, 1996.