

# SQL Studio (BC)



**Release 4.6C**



## Copyright

© Copyright 2001 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft<sup>®</sup>, WINDOWS<sup>®</sup>, NT<sup>®</sup>, EXCEL<sup>®</sup>, Word<sup>®</sup>, PowerPoint<sup>®</sup> and SQL Server<sup>®</sup> are registered trademarks of Microsoft Corporation.

IBM<sup>®</sup>, DB2<sup>®</sup>, OS/2<sup>®</sup>, DB2/6000<sup>®</sup>, Parallel Sysplex<sup>®</sup>, MVS/ESA<sup>®</sup>, RS/6000<sup>®</sup>, AIX<sup>®</sup>, S/390<sup>®</sup>, AS/400<sup>®</sup>, OS/390<sup>®</sup>, and OS/400<sup>®</sup> are registered trademarks of IBM Corporation.

ORACLE<sup>®</sup> is a registered trademark of ORACLE Corporation.

INFORMIX<sup>®</sup>-OnLine for SAP and Informix<sup>®</sup> Dynamic Server<sup>™</sup> are registered trademarks of Informix Software Incorporated.

UNIX<sup>®</sup>, X/Open<sup>®</sup>, OSF/1<sup>®</sup>, and Motif<sup>®</sup> are registered trademarks of the Open Group.

HTML, DHTML, XML, XHTML are trademarks or registered trademarks of W3C<sup>®</sup>, World Wide Web Consortium, Massachusetts Institute of Technology.

JAVA<sup>®</sup> is a registered trademark of Sun Microsystems, Inc.

JAVASCRIPT<sup>®</sup> is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

SAP, SAP Logo, R/2, RIVA, R/3, ABAP, SAP ArchiveLink, SAP Business Workflow, WebFlow, SAP EarlyWatch, BAPI, SAPPHIRE, Management Cockpit, mySAP.com Logo and mySAP.com are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other products mentioned are trademarks or registered trademarks of their respective companies.

## Icons

Icon	Meaning
	Caution
	Example
	Note
	Recommendation
	Syntax
	Tip

## Contents

<b>SQL Studio (BC)</b> .....	<b>6</b>
Starting SQL Studio .....	7
Support Functions .....	8
Logging On to the Database .....	9
Logging On Using File ( Connect .....	10
Logging On Directly Using the File Menu.....	11
Automatic Logon Using Connect in the Standard Toolbar.....	12
WAN Mode.....	13
Logging Off from the Database.....	14
Catalog Manager .....	15
Tables .....	16
Favorites.....	17
Owned Users.....	18
Direct SQL Dialog.....	19
Help Creating SQL Statements .....	20
Executing SQL Statements .....	21
Saving an SQL Statement.....	22
Saving Results.....	23
Importing and Exporting SQL Statements.....	24
Entry of Several SQL Statements .....	25
Setting Parameters for SQL Statements.....	26
Settings .....	27
Query-By-Example Dialog .....	28
Selecting Data Records Using Selection Criteria .....	29
Example for Setting Predicates .....	30
Inserting New Data Records.....	31
Changing Data Records.....	32
Deleting a Data Record .....	33
Report Presentation .....	34
Saving a Query By Example.....	35
Importing and Exporting LONG Columns.....	36
Visual Query Dialog .....	37
Visual Query Window.....	38
Example of a Visual Query Window .....	39
Choosing Columns from Tables .....	40
Synonyms for the Column Names (Synonym).....	41
Definition of Sequence (Sort).....	42
Displaying Columns in the Result (Visible).....	43
Grouping Results (Group).....	44
Choosing Data Records (Criteria).....	45
Setting Parameters for Visual Queries.....	46

---

The AND Link .....	47
The OR Link .....	48
The JOIN Predicate .....	49
Viewing the Relevant SQL Statement .....	50
Saving a Visual Query.....	51
Executing an Existing Visual Query .....	52
Saving Results.....	53
Stored Objects (SQL Studio Objects) .....	54
SQL Studio User Profile.....	55

## SQL Studio (BC)

SAP DB is a relational database system with an SQL-compatible user interface.

"Relational" means that SAP DB provides all its information to users in the form of tables. The standard language SQL (Structured Query Language) provides a set of instructions with which these tables can be managed, updated, and analyzed.

SQL Studio provides easy access to the data held on SAP DB servers. SQL Studio is a multi-document interface application, supporting drag and drop, clipboard transfer, and OLE. SQL Studio thus assists other Microsoft Windows tools with the exchange of data with SAP DB.

SQL Studio is also a helpful tool for application programmers as it allows them to test SQL statements before using them in programs.

## Starting SQL Studio

### Procedure

To start SQL Studio, select *Start* → *Programs* → *SAP Database Technology* → *SQL Studio*.

## Support Functions

### Toolbar

As well as key combinations, SQL Studio also provides a standard toolbar with a menu bar showing frequently used commands. If you hold the mouse pointer on a symbol for a moment, the relevant command will be shown.

### Right Mouse Button

You can also access the menu commands described in the documentation by using the right mouse button. To do this, you must position the mouse exactly on the field to which the command relates.

### F1 Help

Choose *Help* → *SQL Studio Topics* or **F1** and the system displays the documentation for the relevant topic according to your current dialog with the database.

### Display of activated menu commands

A ✓ shows you, that the menu command is switched on.

## Logging On to the Database

### Procedure

Start SQL Studio and use one of the following options for logging on to the database:

[Logging On Using File → Connect \[Page 10\]](#)

[Logging On Directly Using the File Menu \[Page 11\]](#)

[Automatic Logon Using Connect in the Standard Toolbar \[Page 12\]](#)

[WAN Mode \[Page 13\]](#)

## Logging On Using *File* → *Connect*

### Procedure

1. Choose *File* → *Connect*.
2. Select a data source. You can create this data source using the ODBC administrator.
3. Enter the necessary values in the logon screen.

*User name/password*: When the user name and password are entered, they are automatically shown in capitals. If upper and lower case is required, the user name and password must be enclosed between double quotation marks. The double quotation marks are also necessary if the name includes special characters. The name itself must not contain any double quotation marks.

*Serverdb*: Enter the name of the database (SERVERDB).

*Server node*: Enter the network node name of the database server.

Once logged on, the logon is entered in the list of last used logons held in the *File* menu to allow direct log on to a subsequent SQL Studio terminal session. The user name, serverdb name, and name of the network node (server node) are saved.

Depending on the [user profile settings \[Page 55\]](#), the password will also be saved.

- Password stored: The stored password is used for a direct logon with the corresponding entry.
- Password not stored: The logon screen is displayed with the logon data and requires the password be entered.

Up to ten entries can be made in the list of last used logons held in the *File* menu for logging on directly. If ten entries exist already and another is added, the logon which has been out of use for the longest time will be deleted. The entries for logging on direct are sorted chronologically. The logon used most recently is at the top of the list.

You may delete the list with *Clear LRU List* provided you have not yet logged on to the database.

## Logging On Directly Using the *File* Menu

### Procedure

The list of last used logons for a direct logon contains the following information:

- User
- Serverdb where the user has logged on
- Server node to which the user has logged on
- Data source through which the user has logged on

Select the relevant entry to log the required user on to the required database directly. Depending on the settings, direct log on will take place with or without a password prompt.

---

**Automatic Logon Using Connect in the Standard Toolbar**

## **Automatic Logon Using *Connect* in the Standard Toolbar**

If entries for direct logons were made in the list of last used logons stored in the menu *File*, the last logon can be repeated by using *Connect* from the standard toolbar.

If there is no entry for a direct logon, the logging on will take place via the logon screen.

## WAN Mode

### Use

If the connection to the database is poor, you can work with the SQL Studio, albeit with fewer functions, in WAN mode (WAN = Wide Area Network).

### Features

The following functions are provided in WAN mode:

- [Direct SQL Dialog \[Page 19\]](#)
- [User Profile \[Page 55\]](#)

However, you cannot access the catalog manager.

### Activities

Choose *WAN Mode* in the standard toolbar and then log on to the database.

---

**Logging Off from the Database**

## Logging Off from the Database

To close an existing connection with the database, choose *File* → *Disconnect*.

You may want to log off if, for example, you are only authorized to connect to the database once and you want to log on to it from another SAP DB component without ending the current SQL Studio session. Logging off is not necessary before logging on to a different database or elsewhere. In this case, log off is automatic.



For automatic logoff choose *Disconnect* from the standard toolbar.

## Catalog Manager

[Tables \[Page 16\]](#)

[Favorites \[Page 17\]](#)

[Owned Users \[Page 18\]](#)

## Tables

### Tables

*Tables* displays the tables you have access to according to your database and SQL Studio user profile. The tables are arranged by user.

You can use the *Catalog Manager* in the menu bar to edit the database tables:

<i>Drop Table</i>	Delete a database table
<i>Drop View</i>	Delete a view table
<i>Delete All Rows</i>	Delete all rows or data records in a database table
<i>Rename Table</i>	Rename a database table
<i>Rename View</i>	Rename a view table
<i>Show Columns</i>	(or double-click) Display the column definitions for a database table



You cannot edit system tables.

If you select the node for system tables, the system only displays a list of tables approximately the length of a screen (and not the entire list as this is usually a large amount of data). Choose *Continue* at the end of the list to continue the read.

## Favorites

You can use the *Favorites* node to compile a selection of database tables so that you can access them more quickly.

<i>Add to Favorites</i>	Add a copy of a database table to the <i>Favorites</i>
<i>Clear Favorites</i>	Delete the entire contents of the <i>Favorites</i>
<i>Remove Favorite</i>	Delete a database table copy from the <i>Favorites</i>

---

**Owned Users**

## Owned Users

*Owned Users* displays the users the logged on user has owner rights for.

You can edit these users using *Catalog Manager* in the menu bar:

<i>Drop User</i>	Delete a user
<i>Rename User</i>	Rename a user

## Direct SQL Dialog

If you are familiar with SQL queries, want to use other database queries besides the ones offered by your database server, or just want to create an ad-hoc query, you can choose a direct SQL dialog (*Direct SQL*).

1. Choose *View* → *Opening Properties* and set the SQL mode and isolation level of the SQL statement.
2. Choose *View* → *Direct SQL* or *View* → *Direct SQL Auto Commit Off*.
3. Enter the SQL statement.

[Help Creating SQL Statements \[Page 20\]](#)

[Executing SQL Statements \[Page 21\]](#)

[Saving Results \[Page 53\]](#)

[Importing and Exporting SQL Statements \[Page 24\]](#)

[Entry of Several SQL Statements \[Page 25\]](#)

[Setting Parameters for SQL Statements \[Page 26\]](#)

[Settings \[Page 27\]](#)

## Help Creating SQL Statements

- *Suppose List*

When you are creating a new SQL statement, you can use a list of existing statements to help you. Choose `CTRL+SPACE` to view the list of SQL statements. A list of SQL keywords is then displayed.

You can also add your own statements to the list by selecting them and then choosing *Add to Suppose List*. You can delete SQL statements from the list using the function *Remove from Suppose List*.

- [Stored Objects \[Page 54\]](#)

If you have already created and saved SQL statements with the SQL dialog, or if you have been provided with SQL statements from other SQL Studio users, you can view them by choosing *Stored Objects* → *Stored Statements*.

- *Recent Statements*

The statements you have displayed in a window are saved automatically and you can call them by choosing *Direct SQL* → *Recent Statements*. These statements are deleted automatically when you log off from the database.

- *Previous/Next Statement*

You can scroll through the statement history of a database session by choosing *Direct SQL* → *Previous Statement* or *Next Statement*.

## Executing SQL Statements

### Procedure

Choose *Direct SQL* → *Execute*.

You can also select certain sections of your SQL statement if you only want to execute that part of it.

### Result

The SQL Studio status bar shows whether the statement was executed successfully, and how long it took to execute the statement.

The results (if any) of your SQL statement are displayed in a new window.

If you would like to reuse your SQL statement at a later date, save it by choosing *Direct SQL* → *Save As*. They are then stored under *Stored Statements* in the [SQL Studio Objects \[Page 54\]](#) window.



In *Internal Mode*, you can override the settings in the [user Profile \[Page 55\]](#) to display the results (clipped result view).

Choose *Direct SQL* → *Clipped Result View* to override the setting that determines which sections of the results are displayed. This only applies to the window you are currently in.

---

## Saving an SQL Statement

# Saving an SQL Statement

## Use

You can save your SQL statement to be able to use it later.

## Activities

Choose *Direct SQL* → *Save as* and enter a name.

## Result

The SQL statement is stored in the [SQL Studio Objects \[Page 54\]](#) window where you can edit it and use it at a later date.

## Saving Results

### Features

You can save the result of an SQL statement as a text file so that you can export it to an Excel table, for example.

### Procedure

Position your mouse pointer in the results window and choose *Result to file* with the right mouse button. Specify the path where you want to save the file.

### Result

The result is saved as a text file. Columns are separated by tabs. Rows are separated by line breaks. Long columns up to max. 16 characters and characters that cannot be displayed are replaced with blank characters.

## Importing and Exporting SQL Statements

You can import and export SQL statements.

<i>Direct SQL → Load from File</i>	Import SQL statement from a file
<i>Direct SQL → Save as File</i>	Export SQL statement to a file
<i>Direct SQL → Import from Classic Query</i>	Import SQL statements from the program XQUERY

## Entry of Several SQL Statements

You can enter several SQL statements consecutively. They will then be executed consecutively.

To separate several SQL statements from one another, specify a separator line. To be identified as a separator line, the line must start with `//`.

This line can also be used for inserting comments because all further characters in the line will be disregarded. When saving, all separator lines and SQL statements will be included.

If there is more than one result, you can alternate between results using the dropdown list which is displayed.

## Setting Parameters for SQL Statements

# Setting Parameters for SQL Statements

## Procedure

To set parameters for an SQL statement, insert square brackets in all the places where variable entries are possible. A prompt may be formulated as a variable input. This prompt appears in the dialog box where the text for the variable part(s) of the SQL statement is entered.

Various sets of parameters can be stored for each SQL statement.



Table `Article` with `Article No.` and `Unit Price` Columns

If you want to set parameters for the columns `Article No.` and `Unit Price`, you can set the following parameters among the selection criteria:

```
SELECT "Article No.", "Unit Price" FROM "Article"  
WHERE "Article No." > [article number (of):]  
AND "Article No." < [article number (up to):]  
AND "Unit Price" <= [upper price limit:]
```

Make sure you use the correct syntax.



If you want to compare a CHAR column with a parameter, the parameters must also be enclosed by quotation marks:

```
"Name" = '[Name:]'
```

By setting parameters you can also replace entire sections of an SQL statement.



You can formulate a SELECT and, for example, add a WHERE condition using a parameter:

```
SELECT * FROM USERS  
[Your WHERE condition:]
```

## Settings

You can set font and font size separately in input and result windows. Choose *Direct SQL* → *Font*.

## Query-By-Example Dialog

# Query-By-Example Dialog

If you want to use a mask to display or edit your data records, choose the query-by-example dialog (*Query By Example*).

- You want to generate a new *query by example*:  
Choose *View* → *Query By Example*.
- You want to edit an existing *query by example*:  
Choose *Stored Queries By Example* in the *SQL Studio Objects* window. See [Stored Objects \[Page 54\]](#).

Using drag and drop, you can copy the required table to your *query by example*.

If there are no data records in the selected table, the system displays an empty mask in which you can enter a new data record.



You can display the selected tables in read-only mode to avoid setting locks on the table. To do this, go to the [user profile \[Page 55\]](#) and set the function *Open Query By Example read only* to **True**.

[Selecting Data Records Using Selection Criteria \[Page 29\]](#)

[Inserting New Data Records \[Page 31\]](#)

[Changing Data Records \[Page 32\]](#)

[Deleting Data Records \[Page 33\]](#)

[Report Presentation \[Page 34\]](#)

[Importing and Exporting LONG Columns \[Page 36\]](#)

## Selecting Data Records Using Selection Criteria

### Use

Initially, all data records are displayed for editing when you select a table.

You can then either display one particular data record or select a number of records for display.

### Procedure

To define which criteria (predicates) you want to use when selecting data records, choose *Query By Example* → *Predicates*.

- The predicates for each column (*Column Predicates*)
  - If you want to define a new predicate (criterion), first of all select the relevant column and choose *Add*. Now define your predicate.
  - To change the predicate, choose *Edit*.
  - To remove the predicate, choose *Delete*.
  - You can also define how all predicates are linked and how the predicates assigned to a column interact with other predicates.
- Predicates that apply to a whole table (*Table Predicates*)
  - You can define the predicates ROWCOUNT and ORDER.

The lower window then displays the resulting condition for the relevant SQL statement.

To activate the predicates you have set (and thus set a filter), choose *Query By Example* → *Use Predicates*.

### Result

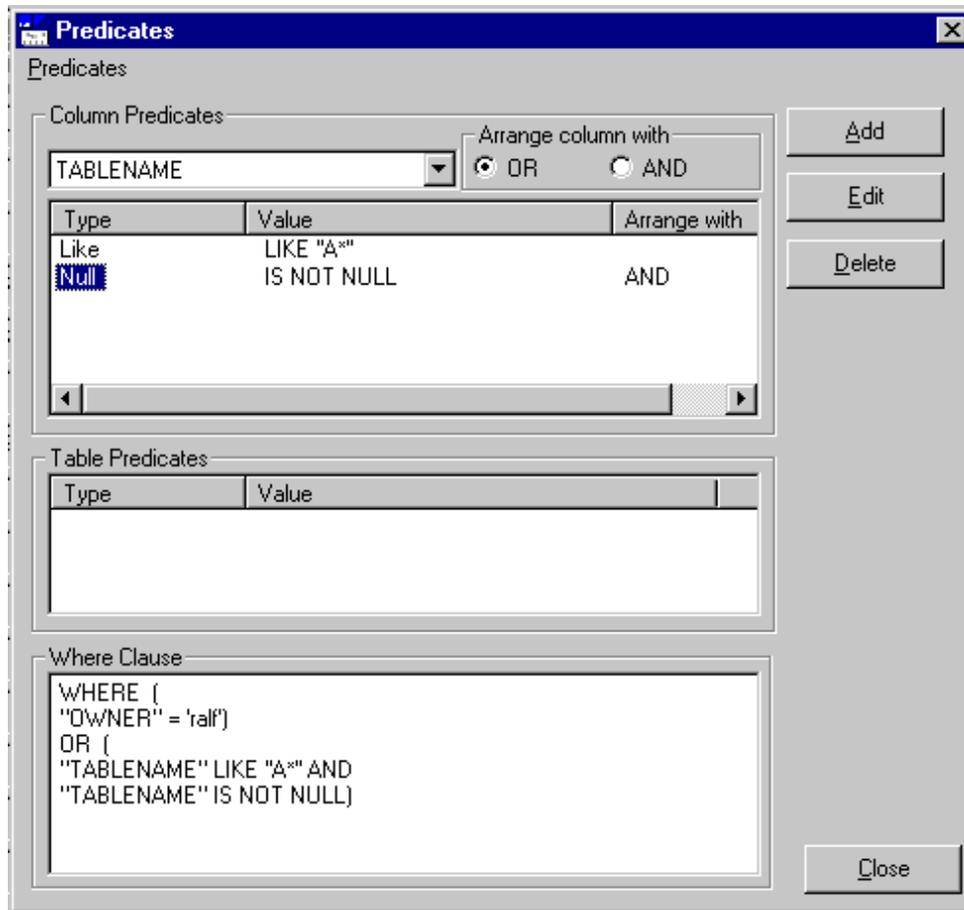
Your *query by example* is then restructured and data records are read from the database. The records read depend on the criteria you have defined.

If necessary, you can activate or deactivate the filter option by choosing *Query By Example* → *Use Predicates*.

[Example \[Page 30\]](#)

## Example for Setting Predicates

## Example for Setting Predicates



## Inserting New Data Records

### Procedure

- Inserting a data record

Choose *Query By Example* → *Insert*. The current data record will be copied to the end of the table. You can overwrite the data.



The data must be confirmed with *Execute*.

- Inserting several data records

To insert several new data records, change to input mode by choosing *Query by Example* → *New*. An empty input mask will appear into which you can enter your data. When the data has been entered a new empty input mask appears.

If you do not wish to insert any more data records, choose *Query by Example* → *New* again or **ESC**.



Columns which may be empty or have a **DEFAULT** value do not need to be specified. On the other hand, there may be columns which have to be specified if they are part of the table key, for example.

If you do not know the type of column, hold the mouse over the relevant column for a moment to display the column type. Lead columns are printed bold in a *query by example* dialog.

If you have specified column values, these will be automatically selected as insertion values.



If it is a **LONG** column, you can import a file or the contents of the clipboard, as long as it consists of text, as the value for this column.

See [Importing and Exporting LONG Columns \[Page 36\]](#)

---

**Changing Data Records**

## Changing Data Records

### Procedure

To change an existing data record, select it and enter the new values.

If you do not know the type of column, hold the mouse over the relevant column for a moment to display the column type. Lead columns are printed bold in a *query by example* dialog.

If the column to be changed is a LONG column, importing for this will be similar to importing when inserting a new data record.

To update a data record, confirm using *Execute* or choose *Query by Example* → *Update Row*. If you wish a change to affect not just the current data records but the entire table, choose *Query by Example* → *Update Table*.



Care needs to be taken when changing values relating to all the records in a table because in fact all the records in a table are altered regardless of whether a filter has been set or not.

## Deleting a Data Record

To delete the current data record, choose *Query by Example* → *Delete*.



The data record is deleted straight away.

---

**Report Presentation**

## Report Presentation

If you want to view all data records at once, you can use *Query by Example* → *Report* to move between the report and mask.

## Saving a Query By Example

### Use

You can save your query by example to be able to use it later.

### Activities

Choose *Query By Example* → *Save as* and enter a name.

### Result

The query by example is stored in the [SQL Studio Objects \[Page 54\]](#) window where you can edit it and use it at a later date.

## Importing and Exporting LONG Columns

### Use

You can import the contents of a LONG column from a file or the clipboard and export it to a file or the clipboard.

### Procedure

Importing a file	<i>Query By Example→Import Long→Import from File</i>
Importing text from the clipboard	<i>Query By Example→Import Long→Import Text from Clipboard</i>
Exporting to a file	<i>Query By Example→Export Long→Export to File</i>
Exporting text to the clipboard	<i>Query By Example→Export Long→Export to Clipboard as Text</i>

## Visual Query Dialog

If you want comprehensive support for formulating SQL statements, choose the visual query dialog (*Visual Query*).

- You want to generate a new *visual query*:  
Choose *View* → *Visual Query*.
- You want to edit an existing *visual query*:  
Choose *Stored Visual Queries* in the window *SQL Studio Objects* See [Stored Objects \[Page 54\]](#).

Using drag and drop, you can copy the required tables to your *visual query*.



You can copy up to 16 tables from one user by selecting the appropriate user.

[Visual Query Window \[Page 38\]](#)

[Choosing Columns From Tables \[Page 40\]](#)

[Synonyms for the Column Names \(Synonym\) \[Page 41\]](#)

[Definition of Sequence \(Sort\) \[Page 42\]](#)

[Displaying Columns in the Result \(Visible\) \[Page 43\]](#)

[Grouping Results \(Group\) \[Page 44\]](#)

[Choosing Data Records \(Criteria\) \[Page 45\]](#)

[Set Parameters for Visual Queries \[Page 46\]](#)

[The AND Link \[Page 47\]](#)

[The OR Link \[Page 48\]](#)

[The JOIN Predicate \[Page 49\]](#)

[Viewing the Relevant SQL Statement \[Page 50\]](#)

[Executing a Stored Visual Query \[Page 52\]](#)

---

**Visual Query Window**

## Visual Query Window

The Visual Query window is divided in two:

- Table selection window (upper half of *visual query* window)

The table selection window contains the tables selected for the SQL statement which is to be formulated. You can select the columns required for the result and link the tables to one another.

The selected tables are located in the table selection window where you placed the tables.

- You can move the tables in the table selection window as you like.
- Use the scroll bars to the right and bottom of the displayed tables to show tables which are not visible. To return to the upper left-hand corner use the button at the bottom right of the table selection window.
- You have the option to sort the tables in the visual query again. To do this, choose *Visual Query → Arrange Table* or *Visual Query → Cascade Table*.
- To delete a table in your visual query, select it and choose *Visual Query → Delete Table*.

- Column selection window (lower half of the *visual query* window)

The result is further defined by a table in the column selection window. You can set the position, possibly a synonym, the sort sequence, and other conditions for the individual columns here.

[Example \[Page 39\]](#)

## Example of a Visual Query Window

The screenshot shows the SQL Studio Visual Query Window. On the left is a tree view of database objects. The main area contains two tables: 'DOMAIN\_SHOW\_TABLE' and 'DOMAIN\_SHOW\_TABLEDEF'. Arrows indicate a relationship between the 'TABLENAME' field in the first table and the 'TABLENAME' field in the second table. Below the tables is a grid showing the query results.

Field	DOMAIN_SHOW_TABLE.OWNER	DOMAIN_SHOW_TABLE.TABLENAME	DOMAIN_SHOW_TABLE.PRIVILEGES	DOMAIN_SHOW_TABLE.TYPE	DOMAIN_SHOW_TABLEDEF.OWNER	DOMAIN_SHOW_TABLEDEF.TABLENAME	DOMAIN_SHOW_TABLEDEF.COLUMNNAME	DOMAIN_SHOW_TABLEDEF.MODE	DOMAIN_SHOW_TABLEDEF.DATATYPE	DOMAIN_SHOW_TABLEDEF.CODETYPE	DOMAIN_SHOW_TABLEDEF.LEN	DOMAIN_SHOW_TABLEDEF.DEC	DOMAIN_SHOW_TABLEDEF.COLUMNPRIVILEGES	DOMAIN_SHOW_TABLEDEF.DEFAULT
Synonym:	ownername	table	userprivilege	tabletype			columnname							
Visible:	Not Visible	Visible	Visible	Visible	Visible	Visible								V
Group:	No	No	No	No	No	No								
Sort:	(None)	(None)	Ascending	Descending										
Criteria:	= 'domain'	like 'A'												
Or:	like 'S'													

At the bottom of the window, the status bar shows: Auto Commit: ON | SQL Mode: INTERNAL | Isolation Level: Committed | No Result 14:23:43.857 : 14:23:44.067 | ODBC 7.2 | U1 | db72 | p26233

---

**Choosing Columns from Tables**

## Choosing Columns from Tables

### Procedure

To select a column for the result: Choose the required column in the selected tables and pull it into the column selection window.

You can select several columns or the entire table (\* or double-click) and display these.



For a drag and drop from one table into another, a join is made to the corresponding column in the destination table (see [The JOIN Predicate \[Page 49\]](#)).

To delete a column in the column selection window, select the column and choose *Visual Query* → *Delete Column*.

## **Synonyms for the Column Names (Synonym)**

Column names in the column selection window are made up of the owner, the table name, and the name of the column.

The name does not always convey the actual function of the column. For this reason you can specify a synonym for the column name.

---

**Definition of Sequence (Sort)****Definition of Sequence (Sort)**

A selected column is not used for sorting the result (standard).

To use a particular column for sorting the result, choose *Sort* and a certain sort sequence.

## Displaying Columns in the Result (Visible)

A selected column is displayed in the result (standard).

To avoid displaying a selected column in the result, this must be set to *Not Visible* in the line Visible.

This can be very useful if you want to make an [AND link \[Page 47\]](#) using a column.

---

**Grouping Results (Group)**

## Grouping Results (Group)

### Use

To get a better overview of the results, you can sort them into columns and display them in groups.

### Procedure

Define the columns you want to group the results in by double clicking on *Yes* in the row *Group*.

### Result

The contents of the columns that were set to *Yes* are now displayed as a navigation tree.

## Choosing Data Records (Criteria)

To formulate selection criteria for a column the cells are used from the row *Criteria* onwards.

When formulating a selection criterion, the comparison values for the particular column must be inserted according to the type of data.



If the data type for the particular column is a character string, the comparison value must be placed in single quotation marks.

To link several selection criteria, AND or OR links can be selected. Generally AND links are formulated in a single criterion line and OR links in several such rows.

[The AND Link \[Page 47\]](#)

[The OR Link \[Page 48\]](#)

---

**Setting Parameters for Visual Queries**

## Setting Parameters for Visual Queries

### Procedure

The parameters of visual queries can be set in order to formulate selection criteria. To do this, insert square brackets where the selection criterion of a visual query has to be variable. A prompt can be formulated between square brackets and then appears in the dialog box in which the text for the variable character of the visual query is entered.

Several sets of parameters can be entered for each visual query and will be retained when storing the visual query.



If you have selected the column `Order_no.` in the table `Orders`, you can set the following parameters for the selection criteria:

```
= [order number:]
```



If you want to compare a CHAR column with a parameter, the parameters must also be enclosed by quotation marks:

```
= '[Name:]'
```

## The AND Link

The criteria for all columns formulated in a row are interpreted as AND links.



If a column is to be used more than once to formulate an AND link but is not to be displayed in the result, set all of the repetitions for the column to *Not Visible*.

[Displaying Columns in the Result \(Visible\) \[Page 43\]](#)

---

## The OR Link

### The OR Link

Criteria which appear in different rows are interpreted as OR links.



A column can be used for an OR link without being visible in the result if the column is set to *Not Visible*.

[Displaying Columns in the Result \(Visible\) \[Page 43\]](#)

## The JOIN Predicate

### Use

The JOIN predicate is used to link columns from different tables.

### Procedure

To link two columns from two different tables, first select a column from the first table and drag this to a column in the second table. You can identify a link between columns in different tables by the line connecting the columns.

To specify the join more accurately, select the join to be specified with the alternate mouse button and then *Show Join Definition*. You can now specify a type and relational operator for the join.

To delete a join, select the join using the right mouse button and then choose *Delete Join*.



If this table is used twice, the SQL Studio assigns aliases. This allows the table to be used for self joins.

---

**Viewing the Relevant SQL Statement**

## Viewing the Relevant SQL Statement

### Procedure

To view the SQL statement relating to a visual query, select *Visual Query*→*Show SQL*.

### Result

A window will appear with the corresponding SQL statement which you can copy to the clipboard for further use.

## Saving a Visual Query

### Use

You can save your visual query to be able to use it later.

### Activities

Choose *Visual Query* → *Save as* and enter a name.

### Result

The visual query is stored in the [SQL Studio Objects \[Page 54\]](#) window where you can edit it and use it at a later date.

---

## Executing an Existing Visual Query

### Executing an Existing Visual Query

To view the results of a visual query, select *Visual Query*→*Execute*.



You can also override the settings in the [user profile \[Page 55\]](#) to display the results (Clipped Result View).

Select *Visual Query* → *Clipped Result View* to override the setting that determines which sections of the results are displayed. This only applies to the window you are currently in.

## Saving Results

### Features

You can save the result of an SQL statement as a text file so that you can export it to an Excel table, for example.

### Procedure

Position your mouse pointer in the results window and choose *Result to file* with the right mouse button. Specify the path where you want to save the file.

### Result

The result is saved as a text file. Columns are separated by tabs. Rows are separated by line breaks. Long columns up to max. 16 characters and characters that cannot be displayed are replaced with blank characters.

## Stored Objects (SQL Studio Objects)

## Stored Objects (SQL Studio Objects)

### Use

With SQL Studio, objects created in the various dialogs can be stored in the database. This allows you to reuse SQL statements or make them accessible to other users.

Once you have logged on to the database, a window opens to show the SQL Studio objects which are available:

- *Stored Statements*
- *Stored Visual Queries*
- *Stored Queries By Example*

and the [SQL Studio User Profile \[Page 55\]](#).

### Activities

You can change the sort sequence for stored objects by *name*, *owner*, *AutoCommit mode*, *SQL mode*, and *isolation level*.

<i>Stored Objects</i> → <i>Modify</i>	(or double-click) Open the requested objects for editing
<i>Stored Objects</i> → <i>Only Own</i>	Display only own memory objects or optionally those made available to you

Depending on how an object was made available to you, you may only be able to execute the object, or read it as well.



If you change an object made available to you, you can only save it as your own object.

<i>Stored Objects</i> → <i>Sharing</i>	Make objects available to others
<i>Stored Objects</i> → <i>New</i>	Create a new object
<i>Stored Objects</i> → <i>Execute</i>	Execute an object
<i>Stored Objects</i> → <i>Delete</i>	Delete an object



If you wish to edit several objects, you can combine the quantity of objects to be edited using `CTRL` or `SHIFT`.

## SQL Studio User Profile

### Use

You can personalize the settings in SQL Studio during operation. When you exit the SQL Studio, these settings are stored in the database where the user with which you logged on to the SQL Studio was created.

### Features

To change the settings, choose *User Profile* in the *SQL Studio Objects* window.

You can change the following functions by simply double-clicking them:

Function	Explanation	Standard user profile
<i>Clipped Result View</i>	Results displayed in segments/groups	True
<i>Confirmation</i>	A prompt that appears before closing a window asking whether or not a new object should be saved	True
<i>Limit in Bytes for Long Columns</i>	The number of bytes that should be called from the database for a LONG column	1024
<i>Open Query By Example read only</i>	<i>Query by example</i> dialog is in read-only mode	True
<i>Replace Non Chars in Long Columns</i>	ASCII-Codes < 32 are replaced by a space when LONG columns are displayed	False
<i>Representation Of Null Value</i>	Determines how the value zero is displayed (freely definable)	?
<i>Restore Catalogstate</i>	The <i>Tables</i> tree structure and the <i>SQL Studio Objects</i> window are restored when you log on again to the database and when you exit the SQL Studio	False
<i>Select Table Types</i>	Displays table type in the <i>Tables</i> tree structure	True
<i>Store Password</i>	The password you used to log on is stored in the registry so you can log on directly	False

If you want to store your modified user profile in the database immediately instead of after leaving the SQL Studio, choose *User Profile* → *Save*.

You can also display the current user profile settings by choosing *User Profile* → *Refresh*. This is particularly useful if you change your user profile in another SQL Studio session or if several users are working with the same database user and make changes to the user profile.

