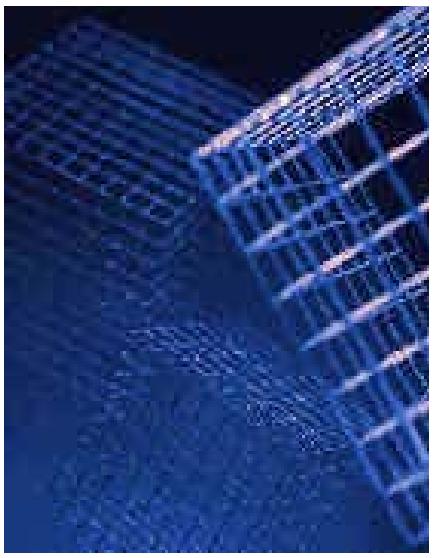
SQL Studio: SAP DB



Version 7.3

Copyright

Icons

Icon	Meaning
	Caution
	Example
	Note
	Recommendation
	Syntax

Typographic Conventions

Type Style	Description
Example text	Words or characters that appear on the screen. These include field names, screen titles, pushbuttons as well as menu names, paths and options.
	Cross-references to other documentation
Example text	Emphasized words or phrases in body text, titles of graphics and tables
EXAMPLE TEXT	Names of elements in the system. These include report names, program names, transaction codes, table names, and individual key words of a programming language, when surrounded by body text, for example, SELECT and INCLUDE.
`Example text`	Screen output. This includes file and directory names and their paths, messages, names of variables and parameters, source code as well as names of installation, upgrade and database tools.
`Example text`	Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.
`<Example text>`	Variable user entry. Pointed brackets indicate that you replace these words and characters with appropriate entries.
`EXAMPLE TEXT`	Keys on the keyboard, for example, function keys (such as `F2`) or the `ENTER` key

SQL Studio: SAP DB 7.3

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, Structured Query Language (SQL), 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&Drop, clipboard transfer, and OLE.

As well as its use as a query definition tool, SQL Studio is a comfortable tool, both for application programmers who want to test SQL statements to be used in their programs, and also for database designers who can benefit from using SQL Studio and the graphical support it provides for the creation and modification of database objects.

Starting SQL Studio

Procedure

- To start SQL Studio, select *Start* → *Programs* → *SAP DB* → *SQL Studio*. Log on to the database instance.
- You can also launch SQL Studio (SQLSTO) from the command line. If you do launch SQL Studio from the command line, you can also specify options for it.

Starting from the command line

1. Switch to the directory where the program **`sqlsto.exe`** is located.
2. Enter the following command:
   `sqlsto [<options>]`

You can use the following options:

-?	Display all command line options
-n <server_node>	Network node
-d <database_name>	Name of the database instance
-u <username[,password]>	User
-version	Display SQL Studio version
-c <sql_mode>	SQL mode
-t <trace_file>	Log file for logging SQL statements

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.

Display database errors

If an error occurs when communicating with the database, you can view more specific information on this error. Choose View → *Show last DB error*.

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 7]

Direct log on using saved log on data [Page 8]

WAN Mode

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).

In WAN mode, only Direct SQL Dialog [Page 19] is available. In addition, you are able to access your SQL Studio user profile [Page 9].

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

NOTIMEOUT

The database will usually terminate an existing database connection automatically if it has been inactive for a certain amount of time.

You can, however, change this.

To do so, activate the *NOTIMEOUT* checkbox in the standard toolbar and then log on to the database.

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 9], 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.

Direct log on using saved log on data

Procedure

SQL Studio provides a list (the Last Recently Used (LRU) list) of the ten last used logons to allow fast and comfortable log on to the database. This list can be found in the *File* menu.

The list contains the following information for each log on:

- User
- Database to which the user has logged on
- Network node to which the user has logged on

Select the relevant entry to log the required user on to the required database directly. Depending on the user settings, [Page 9] direct log on will take place with or without a password prompt.

The most recently saved log on can be repeated using the *Connect* button on the toolbar.

If there is no entry for a direct log on, the log on will occur via the log on screen.

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.

User Settings

Use

You can personalize the settings in SQL Studio during operation. These settings are stored when you exit SQL Studio.

Procedure

To change your settings, choose *View* → *Settings*.

Features

You can now make your changes.

General

Stored password	The password you used to log on is stored in the registry so you can log on directly
Confirmation for saving SQL Studio Objects	A prompt that appears before closing a window asking whether or not a new object should be saved
Query By Example read only	*Query by example* dialog is in read-only mode
Remove failed connection attempts from LRU list	Failed connection attempts are removed from the Last Recently Used (LRU) list
Identifiers in create dialogs upper case	Identifiers are automatically written in upper case letters when creating SQL Studio objects

Result window

Limit for long columns in bytes	The number of bytes that should be called from the database for a LONG column
Representation of null value	Determines how null values are displayed (freely definable)
Clipped result view	Results displayed in segments/groups
Replace non chars in long columns	ASCII-Codes < 32 are replaced by a space when LONG columns are displayed
Copy columns tab separated to clipboard	Separates columns using tabs when copying result sets to the clipboard
Copy result with column header	Column headers are also copied when copying results sets

Catalog Manager

Selected table types	Displays table type in the *Tables* tree structure
Path for local folder	Path to be used to save SQL Studio Objects in a local folder
Restore catalog state	Restore the Catalog Manager when logging back on to the database to the state is was in on exiting SQL Studio
Convert input for catalog filter to uppercase	Automatically convert data entered in the Catalog Manager filter into uppercase

Use local folder DB independent	Use of the local folder is independent of which database the user is logged on to

Query dialog

Isolation level	Isolation level for creation of SQL statements
SQL mode	SQL mode for creation of SQL statements in the direct SQL dialog

Catalog Manager

The Catalog Manager is displayed in the left-hand pane of the SQL Studio window. It shows you all database objects to which you have at least read access.

SQL Studio Objects [Page 10]	Stored database queries and draft database tables
Tables [Page 11]	Database and system tables, view tables and synonyms arranged according to user
Indexes [Page 17]	Indexes that have been created for tables, arranged according to table and its owner
Number sequences [Page 18]	Generated number sequences, arranged according to owner
Favorites [Page 18]	Database tables selected by the user
User [Page 18]	Users for which the user logged onto the database has owner rights
Datenbase procedures [Page 19]	Database procedures that the user created in the database

You can specify that only certain objects be displayed by defining a filter for the Catalog Manager using the object name. The filter is defined using the input field *Catalog Objects start with...* in the toolbar.

* can be used as a wildcard symbol.

This filter does not refer to your SQL Studio objects [Page 10].

SQL Studio Objects

Use

SQL Studio enables you to save the following types of user-created SQL Studio objects in the database or locally in the *Local Folder* subdirectory.

- SQL statements from the Direct SQL, Query By Example and Visual Query dialogs.
- Draft database tables

This functions enables you to reuse or edit the objects, or make them available to other users.

You can enter the path for the *Local Folder* in your user profile settings [Page 9].

You can create a new folder by choosing *Catalog Manager* → *New* → *Folder*.

Procedure

To edit an object, select the object and choose *Catalog Manager* → *Show* or double-click on it to open.

Tables

Tables displays the database tables, view tables and synonyms to which you have access, according to your user profile settings [Page 9]. The entries are arranged by user.

You can create views and synonyms in SQL Studio; database tables can be created and edited.

Database tables [Page 11]

View tables [Page 16]

Synonyms [Page 17]

When you click on the node for a directory under Tables, only approximately one screen full of tables is displayed. Choose *Continue* at the end of the list to continue the read.

Database Tables

Catalog Manager → *New* → *Table*	Create a new database table [Page 12]
Catalog Manager → *Alter Table*	Change the definition of a database table [Page 15]
Catalog Manager → *Drop Table*	Delete a database table
Catalog Manager → *Delete All Rows*	Delete all rows or data records in a database table
Catalog Manager → *Rename Table*	Rename a database table
Catalog Manager → *Show Table*	Display the definition of a database table
Catalog Manager → *Show Content*	Display the contents of the table

You cannot edit system tables.

Creating Database Tables

Use

SQL Studio allows you to create database tables for the database to which you are currently logged on.

Procedure

1. Choose *Tables* → *Catalog Manager*→ *New* → *Table*. The database table definition window opens.

   Assign a table name under *Table Name* and define

   Defining Columns [Page 12]

   Defining Constraints [Page 13]

   Defining Foreign Key Dependencies (Referential Integrity) [Page 14]

   Defining Miscellaneous Information [Page 15]

   - You can copy the definition of a table from the *Catalog Manager* to be used as a template for your new database table. Use *Drag&Drop* to drag the required table to your new table. Change the table definition as necessary.
   - You can use Drag&Drop to copy the content and definition of a stored database query from your SQL Studio Objects [Page 10] to your new database table. You can then change the table columns and miscellaneous information.

     If a database query created in the direct SQL dialog contains several individual queries, the first SELECT statement in the query is copied to the new database table.

2. Choose *Table Definition* → *Create Table* to create the table in the database.

   - *Table Definition* → *Save/Save as* can be used to save the current version of your table in your SQL Studio Objects [Page 10] in draft form. You can then edit this version at any time.
   - To export the SQL statement for creating the database table to the direct SQL dialog, choose *Table Definition* → *Export to Direct SQL*.

Defining columns

Procedure

To define a table column, you must provide the name (`Name`) and data type (`Datatyp`). Depending on the data type, you may be able to define further properties for the column.

`Dim`	Dimension
`Code`	Code type
`Key`	Is the column a key column?
`Not Null`	Can the column accept a null value?
`Default`	System default value

	user-defined / database-dependent
Unique	Exclusivity of the column contents
Comment	Comment text for the column

Add new column

Choose *New Column*. The new column is added to the end of the table.

Copy column

Select the column to be copied, and choose *Copy Column*.

The new column is added to the end of the table.

Import column

Choose *Import Column*. A window opens, containing all columns defined by the current user. Select one or more columns and choose *OK*. The new column(s) is/are added to the end of the table.

Delete column

In order to remove a column from the table, select the column and choose *Remove Column*.

Change column sequence

You can change the columns sequence by selecting one or more columns and moving them upwards or downwards using the arrows ⇑⇓.

Defining Constraints

Prerequisites

A column constraint is composed of one or more conditions that you formulate line-by-line and can link together.

You can only formulate constraints for valid column definitions.

Procedure

Name	The entry for the constraint name is optional.
Column	Choose one of the columns that you have defined in order to set the condition.
Predicat	Select a predicate.
!	Activation of the Not operator makes the formula condition negative. This is not possible with relational operators.
Expression	Expression of the constraint.
Single Value	Single Value
All	Expression must be fulfilled for all elements in a list  `col1>(100,200,300)` means that col1 must be greater than 300
Any	Expression must be fulfilled for at least one element in a list
Default	The column default value is used as an expression

<non>	You can set the field to AND or OR, thus inserting a further line for formulation of another condition which is then linked by AND or OR.  If you subsequently reset the relation type back to none, all following conditions are deleted immediately.

Insert new constraint

Choose *New Constraint*. The new constraint is added to the end of the table.

Copy constraint

Select the constraint to be copied, and choose *Copy Constraint*.

The new constraint is added to the end of the table.

Import Constraint

Choose *Import Constraint*. A window opens, containing all constraints defined by the current user. Select one or more constraints and choose *OK*. Thew new constraint(s) is/are added to the end of the table.

Remove constraint

In order to remove a constraint from the table, select the constraint and choose *Remove Column*.

 Defining Foreign Key Dependencies (Referential Integrity)

Foreign key dependencies (Referential Integrity) describe the dependencies between data of two tables. Under *Ref. Integrity* you can define links for columns in your table to columns in other tables.

Foreign Key Name	Name of foreign key (optional)
Referencing Columns	Referencing columns
Reference Referenced Table Indexed Column	Reference Table to which the link points Indexed columns in the referenced table
On Delete	Behavior of the referenced table when a record is deleted in the referencing table

Insert new link

Choose *New Foreign Key*. The new link is added to the end of the table.

Copy link

Select the link to be copied, and choose *Copy Foreign Key*.

The new link is added to the end of the table.

Import link

Choose *Import Foreign Key*. A window opens containing all links defined by the current user. Select one or more links and choose *OK*. The new link(s) is/are added to the end of your table.

Remove link

In order to remove one or more links from the table, select the link and choose *Remove Foreign Key*.

Defining Miscellaneous Information

Under *Misc* you can enter additional information for your table.

`Comment`	Comment
`Statistics`	
`Sample Value`	Sample Value specifies how large the portion of data is that is included for optimization of the database strategy.
`Sample Unit`	Unit for `Sample Value` (%, rows)
`Others`	
`Temporary Table` `Ignore Rollback`	The table is a temporary table and is saved locally under *Temp*. You can deactivate the rollback option for temporary tables.
`Identifiers upper case`	You can overwrite the user setting for the style of the identifiers.  A change to this setting affects the entire table definition immediately. When changing the definition of an existing database table, you can only change the setting from *Upper case* to *Upper/Lower case*, and not the other way around. If you use a database table as a template for the creation of another database table, the user setting is overwritten.

Changing database table definitions

Use

You can change the definition of database tables in SQL Studio.

Prerequisites

You must have the authorizations required to change the table.

Procedure

Select the required table and choose *Catalog Manager* → *Alter Table*.

A window containing the table definition is opened. You can now make your changes.

Columns [Page 12]

Constraints [Page 13]

Referential Integrity [Page 14]

Miscellaneous Information [Page 15]

Choose *Table Definition* → *Alter Table* to save your changes.

- You can save the modified table in draft form under SQL Studio Objects [Page 10]. Choose *Table Definition* → *Save/Save as*. You can then edit this draft at any time.
- By choosing *Table Definition* → *Show* SQL, you can display the corresponding SQL statement used to modify the database table.

View Tables

Catalog Manager → *New* → *View*	Create a new view table [Page 16]
Catalog Manager → *Drop View*	Delete a view table
Catalog Manager → *Delete All Rows*	Delete all rows or data records in a view table
Catalog Manager → *Rename View*	Rename a view table
Catalog Manager → *Show Columns*	Display the definition of a view table
Catalog Manager → *Show Content*	Display the contents of the view table

Creating View Tables

Use

SQL Studio allows you to create view tables for the database to which you are currently logged on.

Procedure

3. Choose *Tables* → *Catalog Manager*→ *New* → *View*. The view table definition window opens.
4. Using Drag&Drop, copy a database query from your SQL Studio Objects [Page 10] into the definition window. This action copies the structure of the results set for this database query into the new view table.
5. Assign a name to your view table.

   You can rename the basis table column names for your view table under *Alias in View*.
6. If necessary, change the following settings:

Replace existing view	Replace the existing view table
Check on insert or update	Check the view conditions when changing values
Identifiers upper case	Identifier written in upper case letters

7. Choose *Table Definition* → *Create View*, in order to create the view table in the database.

To export the SQL statement for creating the view database to the direct SQL dialog, choose *Table Definition* → *Export to Direct SQL*.

Synonyms

Catalog Manager → *New* → *Synonym*	Create a new synonym [Page 17]
Catalog Manager → *Drop Synonym*	Remove a synonym
Catalog Manager → *Delete All Rows*	Delete all rows or records in a database table that is referred to via a synonym
Catalog Manager → *Rename Synonym*	Rename a synonym
Catalog Manager → *Show Columns*	Display the column definitions for the database table that is referred to via a synonym
Catalog Manager → *Show Content*	Display the contents of database table that is referred to via a synonym

Creating synonyms

Use

SQL Studio allows you to create synonyms for database tables to which you have access.

Procedure

1. In the *Tables* directory, select the database table for which you wish to create a synonym and choose *Catalog Manager* → *New* → *Synonym*. The synonyms definition window opens.
2. Assign the synonym for the database table.
3. Choose *Table Definition* → *Create Synonym* to create the synonym in the database.

By choosing *Table Definition* → *Show* SQL, you can display the corresponding SQL statement used to create the synonym.

Indexes

Indexes contains the tables for which indexes have been created. You can create your own indexes in SQL Studio.

Creating Indexes for Tables

1. Select the required table and choose *Catalog Manager* → *New* → *Index*.
2. Enter a name for the index.

3. Choose the column whose values are to be indexed in the *In Index* column, and set it to `Yes`.
4. If necessary, change the following settings:

Unique	Uniqueness of the indexed column
Index name upper case	Identifier written in upper case letters

5. Choose *Table Definition* → *Create Index*.

To export the SQL statement for creating the index to the direct SQL dialog, choose *Table Definition* → *Export to Direct SQL*.

Number Sequences

Sequences contains the number sequences that the user who is currently logged on has created in the database. You can also generate your own sequences in SQL Studio.

Generating a New Sequence

1. Choose *Catalog Manager*→ *New* → *Sequence*.
2. Enter a name and define the required values.
3. Choose *Sequence Definition* → *Create Sequence*.

To export the SQL statement for generating the number sequence to the direct SQL dialog, choose *Table Definition* → *Export to Direct SQL*.

Favorites

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

Add to Favorites	Add a copy of a database table to the *Favorites*
Clear Favorites	Delete the entire contents of the *Favorites*
Remove Favorite	Delete a database table copy from the *Favorites*

User

Owned Users displays the users for whom the user who is currently logged on has owner rights.

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

Drop User	Delete a user
Rename User	Rename a user

Database Procedures

Under *Procedures*, SQL Studio displays the database procedures that the user created in the database to which the user is currently logged on.

You can process the database procedures using the Catalog Manager in the menu bar.

Show dbproc	View database procedure
Drop dbproc	Delete database procedure

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 the direct SQL dialog.

Creating an SQL statement [Page 19]

Executing an SQL statement [Page 20]

Saving an SQL statement [Page 20]

Saving the results of the query [Page 30]

Importing and exporting SQL Statements [Page 21]

Entering several SQL Statements [Page 22]

Setting parameters for SQL Statements [Page 22]

Creating an SQL statement

Procedure

1. Under User Settings [Page 9] → *Query Dialog*, define the isolation level for your SQL statement.
2. Choose *View* → *Direct SQL* for Auto Commit: ON or *View* → *Direct SQL Auto Commit Off* for Auto Commit: OFF.
3. Enter the SQL statement in the SQL statement window.

You can read the current setting at any time in the status bar for the window.

To comment out SQL statements, use `//` or `--` at the beginning of the line.

You can find help for creating SQL statements as follows:

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 expand the list yourself. Select your SQL statement and choose *Add to Suppose List*. You can delete SQL statements from the list using the function *Remove from Suppose List*.

SQL Studio Objects [Page 10]

If you have already created and saved SQL statements with the direct SQL dialog, or if you have been provided with SQL statements from other SQL Studio users, you can view them in the *SQL Studio Objects* directory in the Catalog Manager.

Recent Statements

The statements that you have used during a database session are saved automatically and can be called up again. Choose *Direct SQL* → *Recent Statements* to do this. 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 an SQL statement

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.

If you would like to reuse your SQL statement at a later date, save it in the SQL Studio Objects [Page 10] folder in the Catalog Manager by choosing *Direct SQL* → *Save As*.

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.

You can hide and show columns in the results window. To do this, click the right mouse button, and choose *Customize* and deactivate or activate *Visible*. Choose *OK*.

You can fix columns to allow you to scroll in the result set more easily. To do this, click the right mouse button and choose *Customize*. Enter the number(s) – beginning at the left - of the column(s) that you want to fix in the field *Fixed Columns*. Choose *OK*.

In *Internal Mode*, you can override the user settings [Page 9] to display the result set (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

Use

You can save your SQL statements in order to reuse them later.

Activities

Choose *Direct SQL* → *Save as* and assign a name for the SQL statement.

Result

The statement is stored in the SQL Studio Objects [Page 10] folder and can be edited and reused there.

Saving the results of the query

Features

You can either export all the results of an SQL statement or just individual cells.

Procedure

Position your mouse pointer in the results window and right-mouse click.

Export results

- *Export Result to → File*

  Saves the result as a text file.

  The 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.

- *Export Result to → Excel*

  Saves the result as an Excel file.

  Each column is saved in cells in the Excel file. You can transfer a maximum of 1024 bytes into one Excel cell. Any data in the CHARACTER format greater than 1021 bytes is truncated and flagged with three periods (...).

Export cells

- *Export Cell to → File*

  Saves the cell contents as a text file.

- *Export Cell to → Zoom Window*

  Displays the cell contents in a separate window.

Exporting and/or displaying the contents of cells is particularly helpful with LONG columns.

Importing and Exporting SQL Statements

Use

You can import and export SQL statements as ASCII format text files.

Procedure

Choose the *Import File/Export File* menu item from the right-click context menu or the corresponding button in the application window toolbar.

Entering Several SQL Statements

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

To separate 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 each one using the dropdown list which appears under the results window.

Setting parameters for an SQL statement

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:]
```

Query-By-Example Dialog

If you want to use a mask to display or edit your data records, choose the query-by-example dialog.

Creating a query by example [Page 23]

Selecting data records using selection criteria [Page 23]

Creating a Query By Example

4. Under User Settings [Page 9] → *Query Dialog*, define the isolation level for your SQL statement.
5. Choose *View* → *Query By Example*.

   The Query By Example window opens.
6. Use Drag&Drop to drag the required tables from the Catalog Manager to the window.

   The data record is built up in the left half of the window. In the right half of the window, you can enter predicates to select specific records [Page 23] .

   To work with an existing Query By Example, select the required Visual Query from your SQL Studio Objects [Page 10] and double-click to open it.

Selecting Data Records Using Selection Criteria

Use

Selection criteria, or predicates, are used to decide which specific information should be displayed from your database table. You can define one selection criterion for each column in the table.

Procedure

In the right half of the Query By Example window, formulate the selection criteria for the table coumns. In the first field, enter the selection type, and in the second enter the value. At the end of the line, choose AND or OR in order to link the selection criterion with the criterion for the next table column.

Result

Choose *Query By Example* → *Refresh* to refresh the new data record.

Changing Data Rows

Procedure

To change an existing data row, 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 the *Query By Example* dialog.

If the column to be changed is a LONG column, the import procedure for the LONG column is similar to importing when inserting a new data row.

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

Changes to the values in all rows in a table are applied irrespective of whether a filter is set or not.

Inserting Data Rows

Procedure

- Inserting a data row

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

  The data must be confirmed with *Execute*.

- Inserting several data rows

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

  If you do not wish to insert any more data records, choose *Query by Example* → *New Row* 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 the *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 25]

Deleting a Data Row

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

The data row is deleted immediately.

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

Saving a Query By Example

Use

You can save your *query by example* in order to reuse it later.

Activities

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

Result

The Visual Query is stored in the SQL Studio Objects [Page 10] directory and can be edited and reused there.

Visual Query Dialog

If you want comprehensive support for formulating SQL statements, choose the visual query dialog.

Creating a visual query [Page 26]

Executing a visual query [Page 29]

Saving a visual query [Page 30]

Saving the results of the query [Page 30]

Viewing the SQL statement for a visual query [Page 31]

Creating a Visual Query

Procedure

7. Under User Profiles [Page 9] → *Query Dialog*, define the isolation level for your SQL statement.
8. Choose *View* → *Visual Query.*

   The Visual Query window opens.

   Table selection window [Page 26] (upper half of *visual query* window)

   Copy the tables that your want to use for your query into this window.

   Column selection window [Page 26] (lower half of the *visual query* window)

   Copy the columns from the selected tables that your want to use for your query into this window. You can set the position, possibly a synonym, the sort sequence, and other conditions for the individual columns here.

   To work with an existing Visual Query, select the required Visual Query from your SQL Studio Objects [Page 10] and double-click to open it.

Selecting Tables for a Visual Query

Using drag and drop, you can copy tables for your query into the Visual Query window.

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

You can select the columns [Page 26] required for the result and link tables [Page 29] to one another.

To delete a table in your visual query, select it and choose *Visual Query* → *Delete Table*.

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 re-sort the tables in the visual query. To do this, choose *Visual Query* → *Arrange Table* or *Visual Query* → *Cascade Table*.

Select Columns

Procedure

To select a column for your query, drag the required column from your tables into the column selection window.

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

If you use drag and drop from one table into another, a Join is created on the corresponding column in the destination table (see Joins [Page 29]).

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

Synonyms for the Column Names (Synonym) [Page 27]

Definition of Sequence (Sort) [Page 27]

Displaying Columns in the Result (Visible) [Page 27]

Grouping Results (Group) [Page 27]

Choosing Data Records (Criteria) [Page 28]

Synonyms for Column Names (Synonym)

In the column selection window, the column names are derived from the owner, the table name, and the name of the column.

These names do 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)

A selected column is not by default used to sort the results.

In order to define a specific column by which to sort the results, choose *Sort* and the required sort sequence.

Displaying Columns in the Result (Visible)

A selected column is by default displayed in the results.

To ensure that a selected column is not displayed in the result, the column must be set to *Not Visible* in the Visible line.

This is especially useful if you want to use the column to perform an AND link [Page 28].

Structuring the Results of a Visual Query

Use

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

Procedure

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

Result

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

Selecting Data Records (Criteria)

The cells following the *Criteria* line are used to formulate selection criteria on which to base the results.

When formulating selection criteria, the comparison values for each column must be correspond to the required data type.

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

You can use AND or OR links to link several selection criteria. Generally AND links are formulated in a single criterion line and OR links in several such rows.

Set Parameters for Visual Queries [Page 28]

The AND Link [Page 28]

The OR Link [Page 29]

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

Where a single *Criteria* row contains more than one entry, it is assumed that the predicates for all the columns involved are linked by 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 results more than once, set all of the repetitions for the column to *Not Visible*.

Displaying Columns in the Result (Visible) [Page 27]

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 27]

Joins

Use

A Join is used to link columns in 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 precisely, select the join to be specified with the right mouse button and then choose *Show Join Definition.* You can now specify the join type and a corresponding relational operator.

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

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

Executing a Visual Query

Procedure

Choose *Visual Query* → *Execute*.

If you would like to reuse your Visual Query at a later date, save it in the SQL Studio Objects [Page 10] folder in the Catalog Manager by choosing *Visual Query* → *Save As*.

Result

The status bar of the Visual Query window shows whether the query was successfully executed, and how long it took to execute the query.

The results (if any) of your Visual Query are displayed in a new window.

You can hide and show columns in the results window. To do this, click the right mouse button, and choose *Customize* and deactivate or activate *Visible*. Choose *OK*.

You can fix columns to allow you to scroll in the result set more easily. To do this, click the right mouse button and choose *Customize*. Enter the number(s) – beginning at the left - of the column(s) that you want to fix in the field *Fixed Columns*. Choose *OK*.

In *Internal Mode*, you can override the user settings [Page 9] to display the result set (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 a visual query

Use

You can save your Visual Query in order to reuse it later.

Activities

Choose *Visual Query* → *Save as* and assign it a name.

Result

The Visual Query is stored in the SQL Studio Objects [Page 10] directory and can be edited and reused there.

Saving the results of the query

Features

You can either export all the results of an SQL statement or just individual cells.

Procedure

Position your mouse pointer in the results window and right-mouse click.

Export results

- *Export Result to* → *File*

  Saves the result as a text file.

  The 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.

- *Export Result to* → *Excel*

Saves the result as an Excel file.

Each column is saved in cells in the Excel file. You can transfer a maximum of 1024 bytes into one Excel cell. Any data in the CHARACTER format greater than 1021 bytes is truncated and flagged with three periods (...).

Export cells

- *Export Cell to → File*

  Saves the cell contents as a text file.

- *Export Cell to → Zoom Window*

  Displays the cell contents in a separate window.

Exporting and/or displaying the contents of cells is particularly helpful with LONG columns.

Viewing the SQL statement for a visual query

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.