



# DB2 Maestro

## **User's guide**

# Table of Contents

Foreword .....	0
<b>I Welcome to DB2 Maestro!</b> .....	<b>1</b>
1 System Requirements .....	3
2 Installation .....	4
3 How can I purchase DB2 Maestro? .....	5
4 License Agreement .....	6
5 About SQL Maestro Group .....	8
6 What's new .....	11
<b>II Getting Started</b> .....	<b>12</b>
1 Connect to a database .....	13
2 Connection parameters .....	14
3 Explaining user interface .....	16
First time started .....	17
Tabbed MDI .....	18
Switching between windows .....	20
4 Shortcut keys .....	22
<b>III Databases and Database Profiles</b> .....	<b>23</b>
1 Creating Database Profiles .....	25
Setting connection properties .....	25
Setting profile options .....	25
2 Editing Database Profile .....	27
Editing connection properties .....	27
Setting profile options .....	28
Setting default directories .....	30
Editing obligatory scripts to execute .....	30
Setting log options .....	32
3 Database Editor .....	34
<b>IV Work with Objects</b> .....	<b>35</b>
1 Create Objects .....	36
Create Database Object Dialog .....	36
Overview of Create Objects Wizards .....	36
Setting object name .....	37
Viewing common information .....	38
2 Edit Objects .....	39
Overview of Object Editors .....	39
Permissions of the Object .....	40
Object grants .....	40
Object Dependencies .....	41
SQL Definition .....	42

Parameter Editor .....	43
Executing functions and procedures .....	43
<b>Modify Object Properties</b> .....	<b>44</b>
<b>Describe Objects</b> .....	<b>44</b>
<b>3 Duplicate Objects</b> .....	<b>45</b>
<b>Duplicate Object Wizard</b> .....	<b>45</b>
Selecting source and destination databases .....	45
Selecting object to duplicate .....	46
Modifying new object definition .....	47
<b>Duplicate Selected Object</b> .....	<b>47</b>
<b>Copy, Paste and Drag-n-Drop features</b> .....	<b>48</b>
<b>4 Browse Objects</b> .....	<b>50</b>
<b>Database Explorer</b> .....	<b>50</b>
Filtering explorer content .....	53
<b>Object Browser</b> .....	<b>53</b>
<b>Object Manager</b> .....	<b>55</b>
<b>Filter Builder Dialog</b> .....	<b>56</b>

## V Database Objects 58

<b>1 Schemas</b> .....	<b>59</b>
<b>Create Schema Wizard</b> .....	<b>60</b>
<b>Schema Editor</b> .....	<b>62</b>
<b>2 Tables</b> .....	<b>65</b>
<b>Create Table Wizard</b> .....	<b>66</b>
<b>Table Editor</b> .....	<b>68</b>
Editing table properties .....	68
Managing table data .....	70
<b>Fields</b> .....	<b>72</b>
Setting field identity .....	75
<b>Indexes</b> .....	<b>76</b>
<b>Foreign Keys</b> .....	<b>78</b>
<b>Checks</b> .....	<b>81</b>
<b>Triggers</b> .....	<b>83</b>
Create Trigger Wizard .....	84
Trigger Editor .....	86
<b>Foreign Key References</b> .....	<b>88</b>
<b>3 Views</b> .....	<b>91</b>
<b>Create View Wizard</b> .....	<b>92</b>
<b>View Editor</b> .....	<b>97</b>
Editing view properties .....	98
Viewing data .....	99
<b>4 Procedures</b> .....	<b>101</b>
<b>Create Procedure Wizard</b> .....	<b>102</b>
Specifying procedure options .....	103
Managing parameters .....	105
Specifying procedure definition .....	105
<b>Procedure Editor</b> .....	<b>106</b>
Editing properties .....	107
<b>5 External Procedures and Functions</b> .....	<b>109</b>
<b>Create External Procedure Wizard</b> .....	<b>110</b>
<b>External Procedure Editor</b> .....	<b>112</b>

<b>6 Functions</b>	<b>115</b>
<b>Create Function Wizard</b>	<b>116</b>
<b>Function Editor</b>	<b>119</b>
Editing properties	119
Viewing Function results	121
<b>7 Distinct Types</b>	<b>123</b>
<b>Create Distinct Type Wizard</b>	<b>124</b>
Specifying Distinct Type properties	124
<b>Distinct Type Editor</b>	<b>126</b>
Editing properties	127
<b>8 Aliases</b>	<b>128</b>
<b>Create Alias Wizard</b>	<b>129</b>
Specifying alias options	130
<b>Alias Editor</b>	<b>131</b>
Editing alias properties	132
Managing alias data	133
<b>9 Sequences</b>	<b>136</b>
<b>Create Sequence Wizard</b>	<b>137</b>
<b>Sequence Editor</b>	<b>138</b>
<b>10 Structured Types</b>	<b>141</b>
<b>Create Structured Type Wizard</b>	<b>142</b>
Specifying type properties	143
Setting type subitems	145
<b>Structured Type Editor</b>	<b>146</b>
Editing type properties	147
UDS Attributes	150
Methods	151
Extended methods	154
<b>11 MQ Tables</b>	<b>157</b>
<b>Create MQ Table Wizard</b>	<b>158</b>
Selecting fields	158
Specifying table properties	159
Specifying WHERE condition	161
<b>MQ Table Editor</b>	<b>164</b>
Editing table properties	165
Viewing table data	167

## VI Server Objects 169

<b>1 Server Editor</b>	<b>171</b>
<b>2 Users</b>	<b>173</b>
<b>Create User Wizard</b>	<b>173</b>
Specifying user options	174
<b>User Editor</b>	<b>176</b>
Editing user properties	177
Managing user objects	179
<b>3 Groups</b>	<b>180</b>
<b>Create Group Wizard</b>	<b>180</b>
Specifying group options	181
<b>Group Editor</b>	<b>183</b>
Editing group properties	184

4 Tablespaces .....	187
Create Tablespace Wizard .....	188
Tablespace Editor .....	189
Editing tablespace properties .....	190
5 Buffer Pools .....	192
<b>VII Queries .....</b>	<b>194</b>
1 SQL Editor .....	196
SQL Formatter .....	198
Executing query .....	199
Query Parameters .....	201
2 Visual Query Builder .....	202
Creating query diagram .....	203
Working with editor area .....	208
Executing query .....	209
<b>VIII Working with data in DB2 Maestro .....</b>	<b>211</b>
1 Data View .....	212
Working with data grid .....	213
Working with info cards .....	218
Data input form .....	219
Data filtering .....	220
2 BLOB Editor .....	223
Editing as image .....	223
Editing as hexadecimal dump .....	224
Editing as plain text .....	224
Editing as HTML .....	225
Editing as PDF document .....	226
3 Export Data Wizard .....	228
Setting destination file name and format .....	228
Setting header and footer .....	229
Selecting fields for export .....	230
Adjusting data formats .....	231
Setting format-specific options .....	231
Setting common export options .....	234
4 Get SQL Dump .....	235
Selecting fields .....	235
Specifying dump options .....	236
5 Import Data Wizard .....	238
Setting source file name and format .....	239
Setting the accordance between source and target columns .....	241
Map builder .....	242
Data formats .....	244
Customizing common options .....	244
<b>IX Database Tools .....</b>	<b>246</b>
1 Script Runner .....	247
2 SQL Script Editor .....	248
3 Generate Database Report .....	250

Selecting reporting elements and setting other report options .....	250
Reporting objects and editing styles options .....	250
Editing database report style .....	251
<b>4 BLOB Viewer .....</b>	<b>252</b>
Viewing as hexadecimal dump .....	252
Viewing as plain text .....	253
Viewing as image .....	254
Viewing as HTML .....	255
Viewing as PDF .....	256
<b>5 Diagram Viewer .....</b>	<b>258</b>
Customizing diagram properties .....	259
Exporting diagram image .....	260
<b>6 Data Analysis .....</b>	<b>262</b>
Input SELECT query .....	263
Managing report data .....	264
<b>7 Report Designer .....</b>	<b>267</b>
Designer Tools and Objects .....	269
Object Inspector .....	271
<b>8 Schema Designer .....</b>	<b>274</b>
Designer Navigation Bar .....	276
Schema Designer Toolbox .....	276
<b>9 Dependency tracker .....</b>	<b>279</b>
<b>10 SQL Generator .....</b>	<b>280</b>
<b>11 DML procedures generation .....</b>	<b>281</b>
<b>12 Generation of updatable views .....</b>	<b>283</b>
<b>13 Split table .....</b>	<b>284</b>
<b>14 Nullable Column Checker .....</b>	<b>286</b>
<b>15 Dialogs .....</b>	<b>288</b>
Find Text dialog .....	288
Replace Text dialog .....	289

## **X Options 292**

<b>1 Application .....</b>	<b>293</b>
Preferences .....	293
Confirmations .....	294
Directories .....	296
Tools .....	296
Explorer .....	298
Object Manager .....	299
SQL Editor .....	299
SQL Script Editor .....	300
Query Builder .....	301
Colors .....	303
BLOB Viewer .....	303
Data Export .....	304
Database Designer .....	305
<b>Object Editors .....</b>	<b>306</b>
Table .....	308
<b>Data Grid .....</b>	<b>308</b>

Options .....	309
Colors .....	311
Formats .....	311
Filter .....	313
<b>2 Editors &amp; Viewers .....</b>	<b>315</b>
General .....	315
Display .....	316
SQL highlight .....	317
XML highlight .....	318
PHP highlight .....	319
Code Insight .....	320
Code Folding .....	321
<b>3 Appearance .....</b>	<b>323</b>
Bars and menus .....	323
Trees and lists .....	324
Edit controls .....	325
Check boxes .....	326
Buttons .....	327
Page controls .....	328
Group boxes .....	329
Splitters .....	330
<b>4 Export Settings .....</b>	<b>332</b>
Specifying destination file .....	332
Selecting setting categories .....	332
Selecting database profiles .....	333
Saving settings .....	333
<b>Index .....</b>	<b>335</b>

# 1 Welcome to DB2 Maestro!

**DB2 Maestro** is the premier Windows GUI admin tool for DB2 development and management. It allows you to make all the database operations easy and fast.

## **Basic DB2 Maestro features**

### **Support of the latest DB2 features**

Use DB2 Maestro to work with DB2 8.x-10.x. Among other features and objects implemented in the latest versions of the server, our software supports materialized views, login mappings, triggers, functions, procedures, and a lot of other useful things.

### **Easy database management**

Database profiles give you the opportunity to connect to databases in one touch and work with the selected databases only. See the [Database Management](#)<sup>[23]</sup> for details.

### **Powerful database object management**

DB2 Maestro provides you with an ability to manage database objects in various ways. For example, you can perform operations with a group of objects as well as with a single object in Object Manager, sort, group and filter the database objects within Object Browser, copy an object from one database to another by a drag-and-drop operation inside the explorer tree, use Windows clipboard to copy a set of objects and so on. For details turn to the [Database Object Management](#)<sup>[58]</sup> section.

### **Working with tables and table subobjects**

DB2 Maestro wizards and editors allow you to create, edit and drop tables as well as their *fields*, *indexes*, and *foreign keys* in a couple of simple operations. See the [Tables](#)<sup>[65]</sup> section to learn more.

### **Building and executing queries**

DB2 Maestro provides two powerful tools which allow you either to edit query text directly with syntax highlighting and code completion or to build a query diagram visually selecting tables and fields, setting links between tables and so on. You can find the detailed description in the [Queries](#)<sup>[194]</sup> section.

### **Powerful data management tools**

DB2 Maestro puts at your disposal a complete set of data management tools with viewing, editing, grouping, sorting and filtering abilities, lookup editors, master-detail data view, BLOB Viewer/Editor, data export, data import and SQL dump modules and more. See the [Data Management](#)<sup>[217]</sup> to learn the details.

### **Wide choice of additional tools**

DB2 Maestro provides you with a number of tools for working with database metadata and SQL scripts, including Script Runner, SQL Script Editor with code folding and script explorer. Moreover, it gives such tools as Schema Designer, BLOB Viewer, Diagram Viewer, Data Analysis, Dependency Tracker, SQL Generator, Report Designer, and a lot of others. To learn more, see the [Database Tools](#)<sup>[246]</sup> section.

### **Security management**

DB2 Maestro gives you a comfortable access to DB2 security features.

### **Full customization according to your preferences and needs**

In DB2 Maestro you can customize the behavior of all its tools, select a user interface scheme and set a lot of other preferences. All the options and their meanings are listed within the [Options](#)<sup>292</sup> dialog description.

## 1.1 System Requirements

### Client environment

- Pentium PC or higher;
- Windows NT4/2000/XP/Vista/Windows 7/Windows 8;
- 256 MB RAM (512 MB recommended);
- 15 MB of free hard disk space;
- SVGA-compatible video adapter
- any DB2 client.

### Server environment

- DB2 8.x-10.x.

## 1.2 Installation

To install **DB2 Maestro** for the first time on your PC:

- download the DB2 Maestro distribution package from the [download page](#) at our site;
- run setup.exe from the local folder and follow the instructions of the installation wizard;
- find the DB2 Maestro shortcut in the corresponding program group of the Windows Start menu after the installation is completed.

To upgrade the installed copy of DB2 Maestro to the latest version:

- download the DB2 Maestro executable file from the [download page](#) at our site;
- unzip downloaded file to any local folder, e.g. *c:\unzipped*;
- exit from DB2 Maestro if it is running;
- replace previous version of DB2 Maestro by copying unzipped files to the DB2 Maestro folder;
- run DB2 Maestro using its shortcut in the Windows Start menu.

You can also use the full distribution package to upgrade your current version of DB2 Maestro. In this case you should repeat the steps of the first-time installation. Note that the full distribution package is larger than a single executable file.

## 1.3 How can I purchase DB2 Maestro?

Thank you for your interest in purchasing **DB2 Maestro**!

You can select licensing options and register DB2 Maestro at its [on-line order page](#). It is possible to purchase on-line, by fax, mail, toll-free phone call, or place a purchase order. We send the software activation key by email within 24 hours after completion of the order process. If you have not received the activation key within this period, please contact our [sales department](#).

All our products and bundles are shipped with 12 months of free upgrades (minor and major ones) or with 36 months of free upgrades for a quite small additional fee. After this period you may renew your license for the next 12(36) months with a 50% discount.

DB2 Maestro has a free 30-day trial. Upon purchasing the product you confirm that you have tested it and you are completely satisfied with its current version.

To obtain technical support, please visit the [appropriate section](#) on our website or contact us by email to [support@sqlmaestro.com](mailto:support@sqlmaestro.com).

## 1.4 License Agreement

**Notice to users:** carefully read the following legal agreement. The use of the software provided with this agreement (the "SOFTWARE") constitutes your acceptance of these terms. If you do not agree to the terms of this agreement, do not install and/or use this software. The use of this software is conditioned upon the user's compliance with the terms of this agreement.

- **License grant.** SQL Maestro Group grants you a license to use one copy of the version of this SOFTWARE on any single hardware product for as many licenses as you purchase. "You" means a company, an entity or an individual. "Use" means storing, loading, installing, executing or displaying the SOFTWARE. You may not modify the SOFTWARE or disable any licensing or control features of the SOFTWARE except as an intended part of the SOFTWARE's programming features. This license is not transferable to any other company, entity or individual. You may not publish any registration information (serial numbers, registration keys, etc.) or pass it to any other company, entity or individual.
- **Ownership.** The SOFTWARE is owned and copyrighted by [SQL Maestro Group](#). Your license confers no title or ownership of the SOFTWARE and should not be construed as a sale of any rights for the SOFTWARE.
- **Copyright.** The SOFTWARE is protected by the United States copyright law and international treaty provisions. You acknowledge that no title to the intellectual property in the SOFTWARE is transferred to you. You further acknowledge that title and full ownership rights to the SOFTWARE will remain the exclusive property of [SQL Maestro Group](#) and you will not acquire any rights to the SOFTWARE except as expressly set forth in this license. You agree that any copies of the SOFTWARE will contain the same proprietary notices which appear on and in the SOFTWARE.
- **License and distribution.** An unregistered copy of the SOFTWARE ("UNREGISTERED SOFTWARE") may be used for evaluation purposes. The UNREGISTERED SOFTWARE may be freely copied and distributed to other users for their evaluation. If you offer this UNREGISTERED SOFTWARE installation package for download, then you agree to:
  - replace existing version of the UNREGISTERED SOFTWARE installation package with the new package immediately after a new version of the SOFTWARE is released by SQL Maestro Group, or
  - delete an obsolete version of the UNREGISTERED SOFTWARE installation package immediately upon written email notice by SQL Maestro Group.

A registered copy of the SOFTWARE ("REGISTERED SOFTWARE") allows you to use the SOFTWARE only on a single computer and only by a single user at a time. If you wish to use the SOFTWARE for more than one user, you will need a separate license for each individual user. You are allowed to make one copy of the REGISTERED SOFTWARE for back-up purposes.

- **Reverse engineering.** You affirm that you will not attempt to reverse compile, modify, translate, or disassemble the SOFTWARE in whole or in part.
- **Unauthorized use.** You may not use, copy, rent, lease, sell, modify, decompile, disassemble, otherwise reverse engineer, or transfer the SOFTWARE except as provided in this agreement. Any such unauthorized use shall result in immediate and

automatic termination of this license.

- **No other warranties.** [SQL Maestro Group](#) does not warrant that the SOFTWARE is error-free. SQL Maestro Group disclaims all other warranties with respect to the SOFTWARE, either express or implied, including but not limited to implied warranties of merchantability, fitness for a particular purpose and noninfringement of third party rights. Some jurisdictions do not allow the exclusion of implied warranties or limitations on how long an implied warranty may last, or the exclusion or limitation of incidental or consequential damages, so the above given limitations or exclusions may not apply to you. This warranty gives you specific legal rights and you may also have other rights which vary from jurisdiction to jurisdiction.
- **Limited warranty.** This SOFTWARE is provided on an "AS IS" basis. [SQL Maestro Group](#) disclaims all warranties relating to this SOFTWARE, whether expressed or implied, including but not limited to any implied warranties of merchantability or fitness for a particular purpose. Neither SQL Maestro Group nor anyone else who has been involved in the creation, production, or delivery of this SOFTWARE shall be liable for any indirect, consequential, or incidental damages arising out of the use or inability to use such SOFTWARE, even if SQL Maestro Group has been advised of the possibility of such damages or claims. The person using the SOFTWARE bears all risk as to the quality and performance of the SOFTWARE.

Some jurisdictions do not allow limitation or exclusion of incidental or consequential damages, so the above given limitations or exclusion may not apply to you to the extent that liability is by law incapable of exclusion or restriction.

- **Severability.** In the event of invalidity of any provision of this license, the parties agree that such invalidity shall not affect the validity of the remaining portions of this license.
- **No liability for consequential damages.** In no event shall [SQL Maestro Group](#) or its suppliers be liable to you for any consequential, special, incidental or indirect damages of any kind arising out of the delivery, performance or use of the SOFTWARE, even if SQL Maestro Group has been advised of the possibility of such damages. In no event will SQL Maestro Group's liability for any claim, whether in contract, tort or any other theory of liability, exceed the license fee paid by you, if any.
- **Entire agreement.** This is the entire agreement between you and [SQL Maestro Group](#) which supersedes any prior agreement or understanding, whether written or oral, relating to the subject matter of this license.
- **Reserved rights.** All rights not expressly granted here are reserved to [SQL Maestro Group](#).

## 1.5 About SQL Maestro Group

**SQL Maestro Group** is a privately-held company producing high-quality software for database administrators and developers. The united team of eminently qualified developers is pleased to create new software products for commercial, academic and government customers worldwide. We do our best to design and develop products that remove complexity, improve productivity, compress time frames, and increase database performance and availability. We are glad to realize that our products take usual chores upon themselves, so that our customers could have more time left for their creative work.

The company was founded in 2002 as an essential partner for every business that is trying to harness the explosive growth in corporate data. SQL Maestro Group employs an international team concentrating their efforts on cutting-edge DBA tools development.

The slogan of our company is **The Shortest Path to SQL**. It is aimed to denote that we set to create easy-to-use products meant for those who appreciate comfort, friendly program interface and support when working with SQL servers.

- We are pleased to facilitate your job.
- We aim at being of considerable assistance to our clients.
- We feel contented doing our beloved work.

At present, our company offers a series of Windows GUI admin tools for SQL management, control and development of the following servers: **MySQL, Microsoft SQL Server, PostgreSQL, Oracle, SQL Anywhere, DB2, SQLite, Firebird, and MaxDB**. We also produce universal tools to be used for administering any database engine accessible via ODBC driver or OLE DB provider. Such products may be the clear-cut decision for those who constantly work with several database servers.

**SQL Maestro** is the premier Windows GUI admin tool for database development, management, and control.

It provides you with the ability to perform all the necessary database operations such as creating, editing, copying, extracting and dropping database objects; moreover, you can build queries visually, execute queries and SQL scripts, view and edit data including BLOBs, represent data as diagrams, export and import data to/from most popular file formats, manage users and their privileges (if possible), and use a lot of other tools designed for making your work with your server comfortable and efficient.



**SQL PHP Generator** is a powerful tool for creating database-driven web applications visually. It allows you to generate high-quality PHP scripts for working with tables, views and queries through the web. You needn't have any programming background to use it.



**SQL Data Wizard** is a high-capacity Windows GUI utility for managing your data.

It provides you with a number of easy-to-use wizards for performing the required data manipulation easily and quickly. The tool allows you to convert any ADO-compatible database to the Oracle database, export data from Oracle tables and queries to most popular formats, import data into the tables, and generate ASP.NET scripts for the selected tables and queries.



**SQL Code Factory** is a premier GUI tool aimed at the SQL queries and scripts development.

It allows you to manage SQL queries and scripts using such useful features as code folding, code completion and syntax highlighting, build query visually, execute several queries at a time, execute scripts from files, view and edit result data with filtering, sorting and grouping abilities, export data to as many as 14 file formats including Excel, RTF and HTML, import data from Excel, CSV, XML and text files, view and edit BLOBs in various way, build diagrams based on Oracle data, and much more.



**Service Center for MySQL** is a multi-functional Windows GUI utility for MySQL server maintenance and data reservation.

It allows you to create tasks for backing your data into a script, restoring data from a script, analyzing, checking, repairing and optimizing database tables. In addition to these functions, Service Center for MySQL provides you with a powerful task scheduler, an ability of using scenarios and other useful features.



The software products are constantly optimized for the latest server versions support.

You can use the following contact information if necessary:

Our web-site [www.sqlmaestro.com](http://www.sqlmaestro.com)

Postal address: **SQL Maestro Group**

140 Broadway, Suite 706  
New York City, New York 10005  
United States

**Thank you for your interest to our company!**

## 1.6 What's new

Please find out the latest DB2 Maestro news at <http://www.sqlmaestro.com/products/db2/maestro/news/>

## 2 Getting Started

The topics in this section provide some basic information about DB2 Maestro, what it is for and what you can do with it.

### How to get started:

- [Connect to a database with DB2 Maestro](#)<sup>[13]</sup>
- [Explaining user interface](#)<sup>[16]</sup>
- [How DB2 Maestro looks when you start it for the first time](#)<sup>[17]</sup>
- [Shortcut keys](#)<sup>[22]</sup>

### Learning more:

- ❑ Study the [Overview of Database Object Management](#)<sup>[35]</sup> section for detailed instructions on using DB2 Maestro.
- ❑ See [Database Tools](#)<sup>[246]</sup> and [Queries](#)<sup>[194]</sup> sections for instructions on more advanced procedures!
- ❑ Find out more about [Working with Data in DB2 Maestro](#)<sup>[211]</sup>.
- ❑ Customize the way DB2 Maestro works, see [Program Options](#)<sup>[292]</sup> for full details.

## 2.1 Connect to a database

To manage an existing database with DB2 Maestro, you have to [create the according database profile](#)<sup>[26]</sup> first. A profile stores database connection settings, and some additional options to customize the way the software works with the database. After the creation database profiles appear as nodes in the Explorer tree on the left (profile properties can be later changed with [Database Profile Editor](#)<sup>[27]</sup>).

When the profile is created you can connect to the database. To do so, select the database in the Explorer tree, or either select the [Database | Connect to Database](#) main menu item or use the [Connect to Database](#) item of the popup menu. You can also double click the database node in the explorer tree. If connection succeeds, the database node expands displaying the tree of database objects (tables, views, procedures, etc). The database becomes ready for your activities.

### How can I disconnect from a database?

In order to disconnect from a database you should first select the database in the explorer tree, then either

- select the [Database | Disconnect from Database](#) main menu item
- or
- use the [Disconnect from Database](#) item of the popup menu.

See also: [Connection parameters](#)<sup>[14]</sup>

## 2.2 Connection parameters

DB2 Maestro unites databases in groups at the Explorer tree. **Database group** is a pure logical structure and can be named by any convenient appellation (i.e. a database name, a database host, etc.).

To setup a connection to a DB2 database stored on a remote server, you need:

1. Get any DB2 client with ODBC driver installed on your computer. We recommend to use clients with the DB2 Command Line Processor (CLP) tool. It seems that IBM Data Server Runtime Client is the best solution in this situation. It is free and is not overloaded with various drivers. To obtain this client, visit the [publisher's site](#) and download *IBM Data Server Runtime Client (Windows 32-bit AMD and Intel x86)*.

2. Get this database catalogued by db2 on the client machine. The DB2 client can work only with catalogued databases assumed with cataloged nodes (servers). To work with a DB2 database with DB2 Maestro, you have to get this database catalogued on the client machine (the operation is accomplished only once) and to specify the *name of user* registered on DB2 server and the corresponding *password*. To catalogue a DB2 database, follow the steps below:

2.1 Catalog the node of server. To get all the catalogued nodes of the client machine, enter the following command to the DB2 Command Window or similar (It's time to use DB2 Command Line Processor ;) ):

```
db2 list node directory
```

To catalog a new server node, use the following command:

```
db2 catalog tcpip node <nodename> remote <hostname or ip address>  
server <db2servicename> [remote <instance_name> ostype <OSname>]
```

nodename – the node name to be used on your computer

hostname or ip address – hostname/ipaddress of server storing the DB2 database

db2servicename – ServiceName or port number of db2instance (db2 get dbm cfg | grep SVCENAME)

instance\_name – instance name of db2 database.

OSname – Solaris, windows, AIX, Linux etc.

Example:

```
db2 catalog tcpipnode testnode remote testhost server 50000  
db2 catalog tcpipnode testnode remote testhost server 50000 remote db2inst1
```

2.2 Catalog the necessary database on the corresponding node. To get the list of catalogued databases, use the following command:

```
db2 list db directory
```

To add the necessary database to the catalog, enter the following command:

```
db2 catalog database <dbname> as [<db_alias>] at node <nodename>
```

dbname – database name

db\_alias – optional database alias, if you want database name different on client.

nodename – nodename you created on first step

Example:

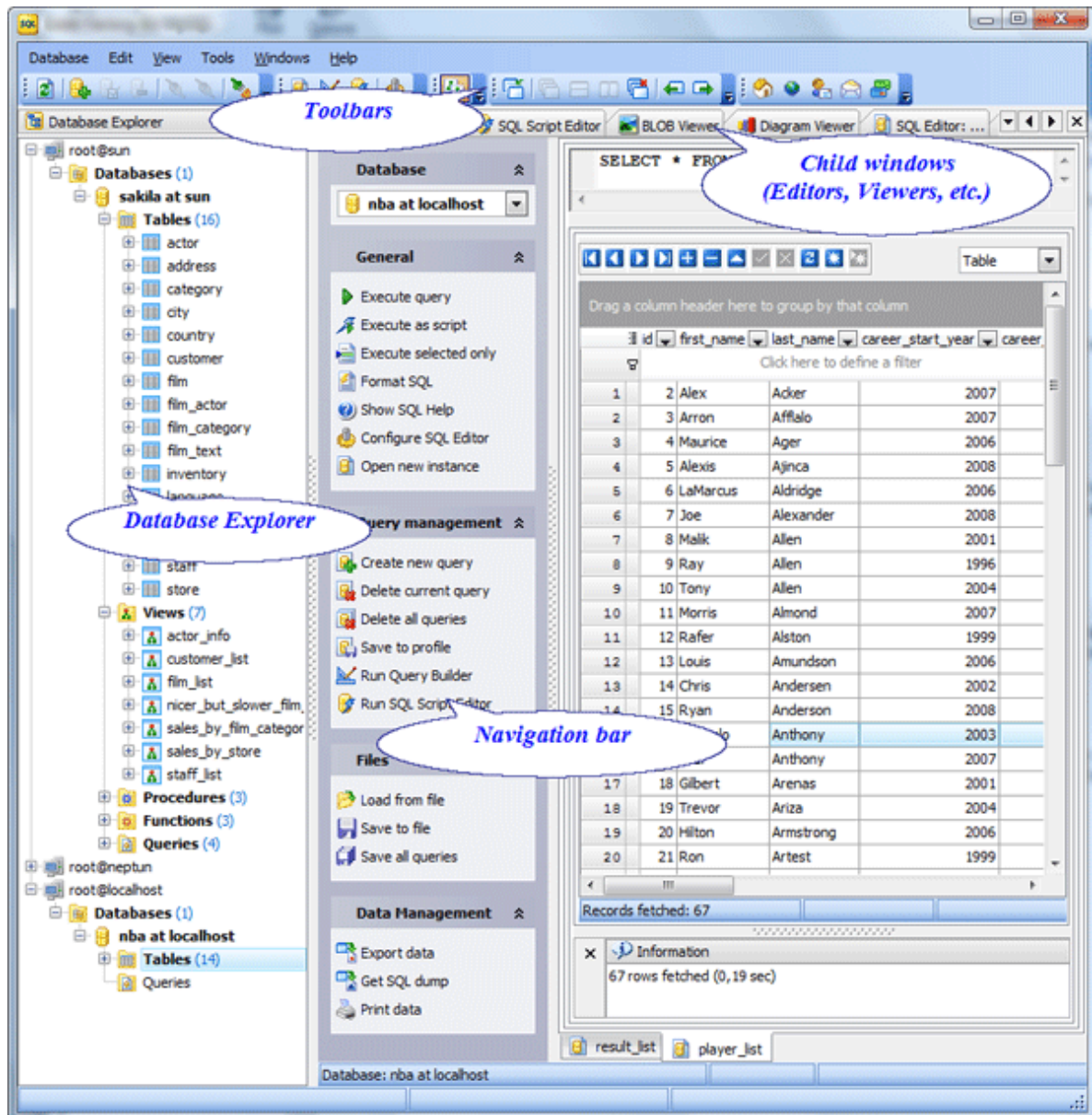
```
db2 catalog database testdb as testdb at node testnode
```

```
db2 catalog database sample at node testnode
```

## 2.3 Explaining user interface

The SQL Maestro Group products are famous for their clear and intuitive user interface. The programs are built around the three-pane workspace that includes the [database explorer](#) and child windows consisting of the [navigation bar](#) and [work area](#).

This topic provides a brief guide to the components of DB2 Maestro's user interface. For detailed descriptions, see below.



### Database Explorer

The [Database Explorer](#) occupies the left side of DB2 Maestro main window. It represents all the connected databases objects [including system objects](#)<sup>[28]</sup>.

The explorer provides the fastest way to reach the object properties, to perform the following operations with database profiles using the popup menu:

- create new objects (database profiles, database objects, table objects...);
- edit currently selected objects;
- remove currently selected objects from the explorer tree;
- duplicate objects;
- rename objects if available and edit object comments out of the object editor.

See also: [Filtering explorer content](#)

## Editors and Viewers

According to the MDI style implementation the DB2 Maestro tools and editors are opened in appropriate windows. Each window consists of a navigation bar and work area. The software supports Classic and Tabbed MDI.

See also: [Switching between windows](#)<sup>[20]</sup>, [Tabbed MDI](#)<sup>[18]</sup>

## Navigation bar

The [Navigation Bar](#) contains a set of logically grouped links provided to realize the corresponding actions. Just position the mouse over a link and wait for a second to display the appropriate action shortcut making it possible for experienced users to control the program almost entirely with the keyboard.

See also: [Shortcut keys](#)<sup>[22]</sup>

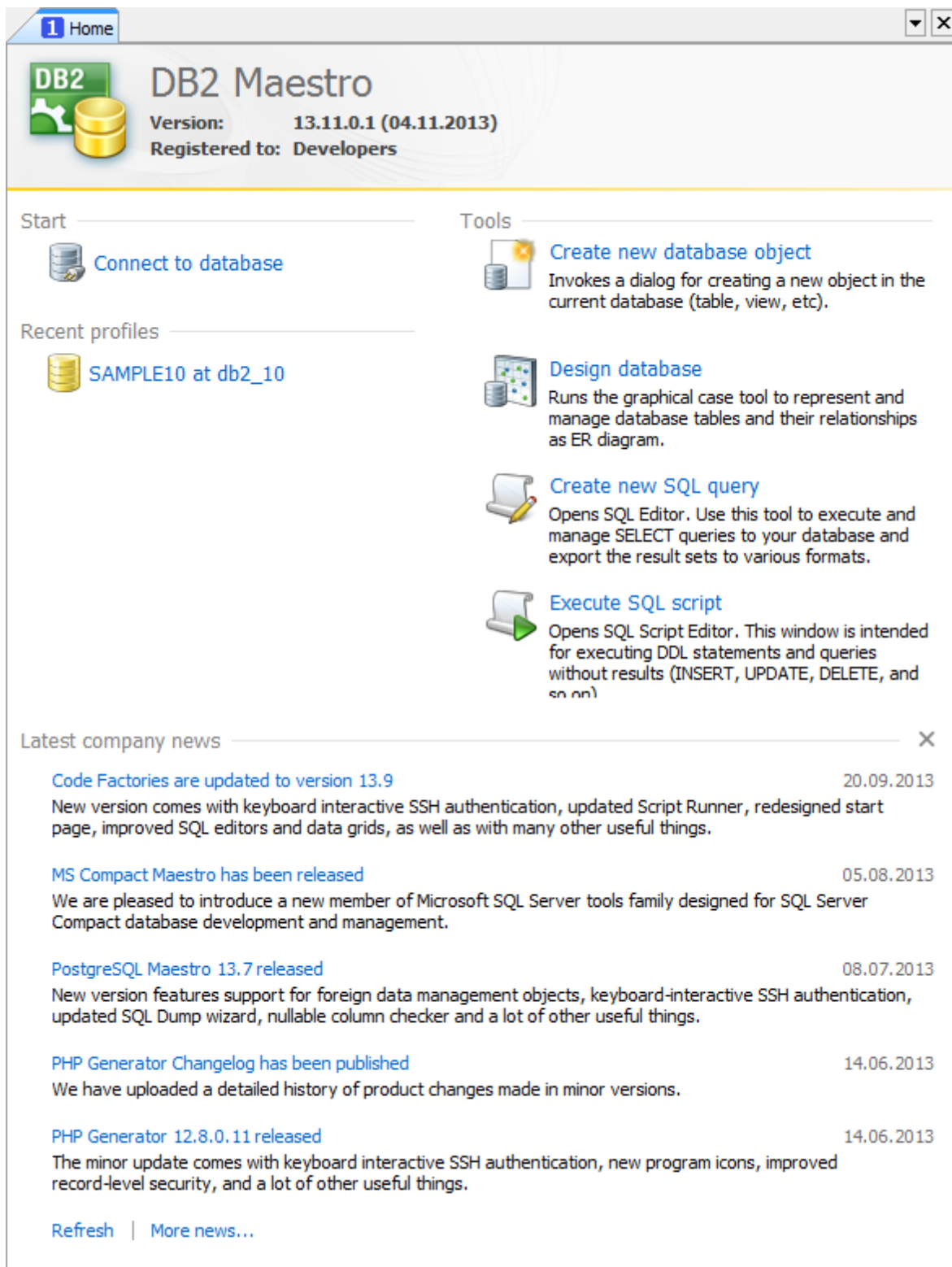
## Toolbars

The bars occupy the top of the main window. The [Toolbars](#) provide quick access to the most frequently-used functions. Just position the mouse over a tool and wait for a second to display a brief text describing what it is for.

### 2.3.1 First time started

This is how DB2 Maestro looks when you run it for the first time. The [Connect to database](#) link allows you to start working with existing databases. Follow the link to open [Create database profile](#)<sup>[25]</sup>

The window provides you with quick access to the [Create Database Object](#)<sup>[36]</sup> dialog, [Schema Designer](#)<sup>[274]</sup>, [SQL Editor](#)<sup>[196]</sup>, and [SQL Script Editor](#)<sup>[248]</sup> recently used database profiles. At the bottom of the page the latest company news and current discount programs are represented.

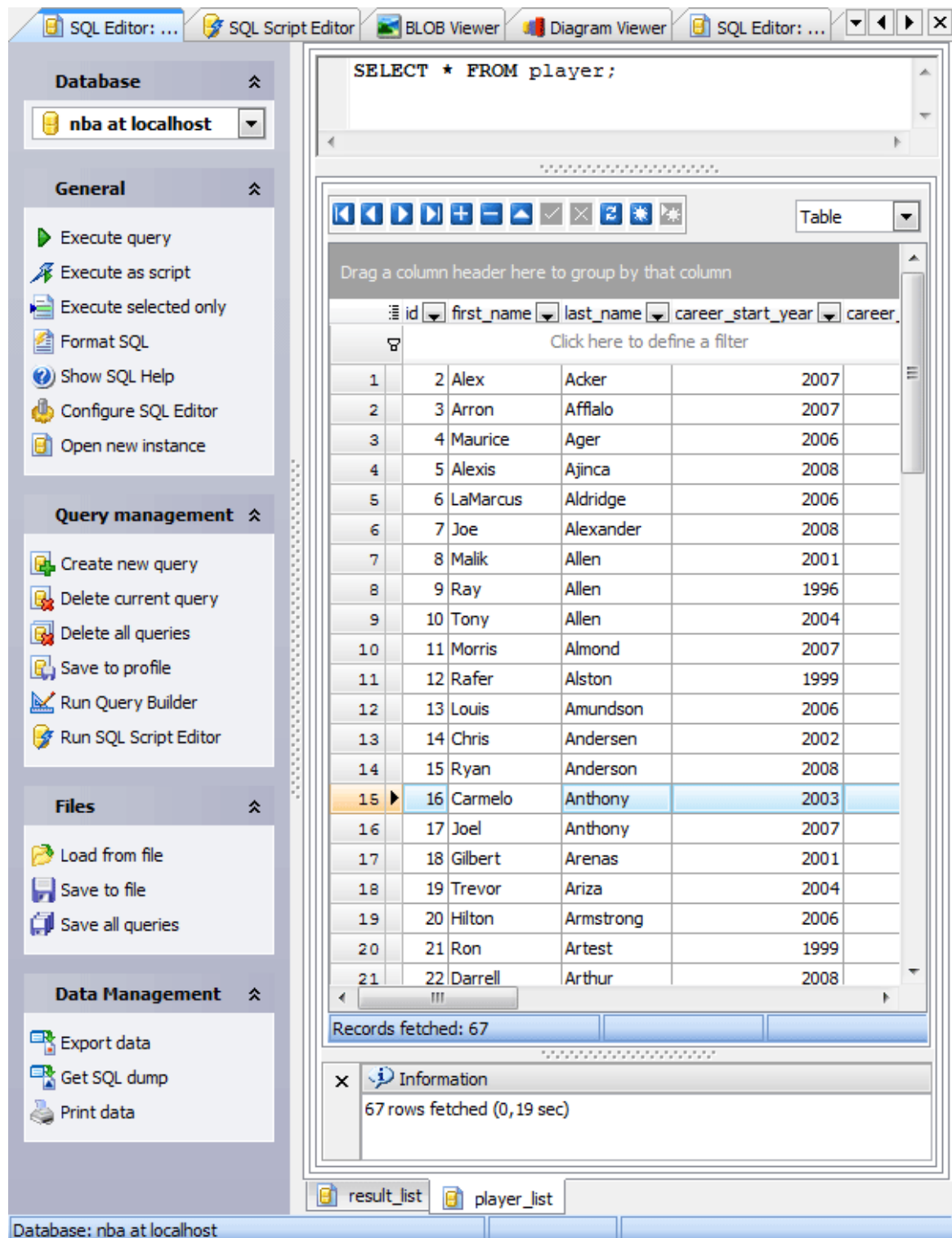


### 2.3.2 Tabbed MDI

DB2 Maestro provides you with a possibility to choose ([Options|Application](#)) your favorite UI. Among the **classic MDI style** the **tabbed MDI style** is also available.

Applying the style you'll get all the objects editors opening on separate sheets. You can move from one sheet to another by clicking the sheet tabs at the bottom of the working area. The tab for the active sheet is underlined in the color you choose; tabs for inactive sheets are fully colored.

You can switch between the sheets with corresponding sheet tabs or using **Ctrl+Tab**. If you don't see the tool you want, click the tab scrolling buttons to display the tab, and then click the tab. You can also move the sheets.

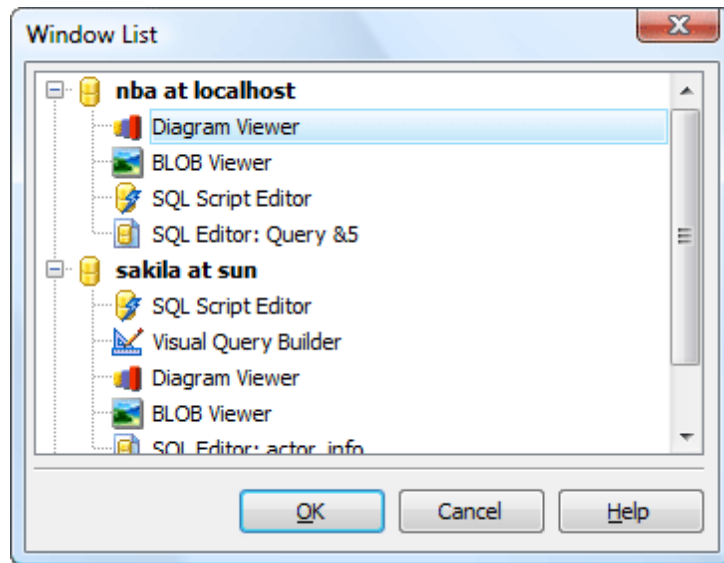


### 2.3.3 Switching between windows

The [Window List](#) dialog allows you to switch the child application windows quickly. To open the dialog select the [Windows | Window List...](#) item of the main menu or use the

**Alt+O** hot keys combination.




Most of the windows are linked according to their active databases and displayed in the form of a tree, e.g. [Table Editor](#), [SQL Editor](#), [Diagram Viewer](#), etc. Windows which are common for the entire program are shown as separate nodes of the tree.



To activate the window you need, select one of the window tree items and click the **OK** button.

## 2.4 Shortcut keys

The following table describes the default shortcut keys in DB2 Maestro.

<b>Interface</b> 	
Window list	<b>Alt+O</b>
Previous Window	<b>F6</b>
Next Window	<b>Ctrl+F6</b>
Show Database Explorer	<b>F11</b>
Refresh	<b>F5</b>
Exit	<b>Alt+F4</b>
DB2 Maestro help	<b>F1</b>
<b>Clipboard</b>	
Cut	<b>Ctrl+X</b>
Copy	<b>Ctrl+C</b>
Paste	<b>Ctrl+V</b>
Select all	<b>Ctrl+A</b>
Find	<b>Ctrl+F</b>
Replace	<b>Ctrl+H</b>
Search again	<b>F3</b>
Undo	<b>Ctrl+Z</b>
Redo	<b>Shift+Ctrl+Z</b>
<b>SQL Editors</b> 	
Open SQL Editor	<b>Ctrl+E</b>
Open SQL Script Editor	<b>Ctrl+R</b>
Open Visual Query Builder	<b>Ctrl+Q</b>
Execute query	<b>(F9) or (F8)</b>
Execute query as script	<b>(Shift+F9) or (Shift+F8)</b>
Execute selected only	<b>(Alt+F9) or (Alt +F8)</b>
Go to line	<b>Ctrl+G</b>
Format selected SQL	<b>Ctrl+Alt+F</b>
Create new query	<b>Ctrl+N</b>
Delete current query	<b>Ctrl+Alt+D</b>
Load script from file	<b>Ctrl+O</b>
<b>Database management</b> 	
Create a new database profile	<b>Shift+Ctrl+P</b>
Edit an existing database profile	<b>Shift+Ctrl+E</b>
Rename a database profile (object)	<b>F2</b>
Remove database profile	<b>Shift+Ctrl+R</b>
Connect to the database	<b>Shift+Ctrl+C</b>
Disconnect from the database	<b>Shift+Ctrl+D</b>
Create a database object	<b>Shift+Ctrl+N</b>
Object Browser	<b>Shift+Ctrl+O</b>
Open BLOB Viewer	<b>Ctrl+B</b>

### 3 Databases and Database Profiles

DB2 Maestro allows you to manipulate databases by means of database profiles. Profile contains database connection settings and a set of options to automatize common manipulations with databases (a possibility to connect to the database at DB2 Maestro startup, login prompt before connection, etc.). To start working with databases in DB2 Maestro, you should create database profile(s) first.

Use the following links for details:

#### ■ **How can I create new database profiles?**

In DB2 Maestro database profiles are created within [Create Database Profiles Wizard](#)<sup>[26]</sup>. In order to run the wizard you should either

- select the [Database | Create Database Profiles...](#) main menu item
- or
- use the [Create Database Profiles...](#) item of the popup menu.

Using [Create Database Profiles Wizard](#) set the necessary connection and authorization options and click the [Ready](#) button to complete the operation.

#### ■ **How can I edit existing database profile options?**

Database connection properties and profile options are edited within the [Database Profile Properties](#)<sup>[27]</sup> dialog window. In order to open the dialog for the selected database profile you should either

- select the [Database | Edit Database Profile...](#) main menu item
- or
- use the [Edit Database Profile...](#) item of the popup menu.

#### ■ **How can I remove database profiles?**

In order to remove a database profile you should first select the database profile in the explorer tree, then either select the [Database | Remove Database Profile](#) main menu item, or use the [Remove Database Profile](#) item of the popup menu and confirm removing profile in the dialog window to complete the operation.

#### ■ **How can I connect to a database?**

In order to connect to a database you should first select the database in the explorer tree, then either

- select the [Database | Connect to Database](#) main menu item
- or

- use the [Connect to Database](#) item of the popup menu.

#### ■ **How can I disconnect from a database?**

In order to disconnect from a database you should first select the database in the explorer tree, then either

- select the [Database | Disconnect from Database](#) main menu item
- or
- use the [Disconnect from Database](#) item of the popup menu.

## 3.1 Creating Database Profiles

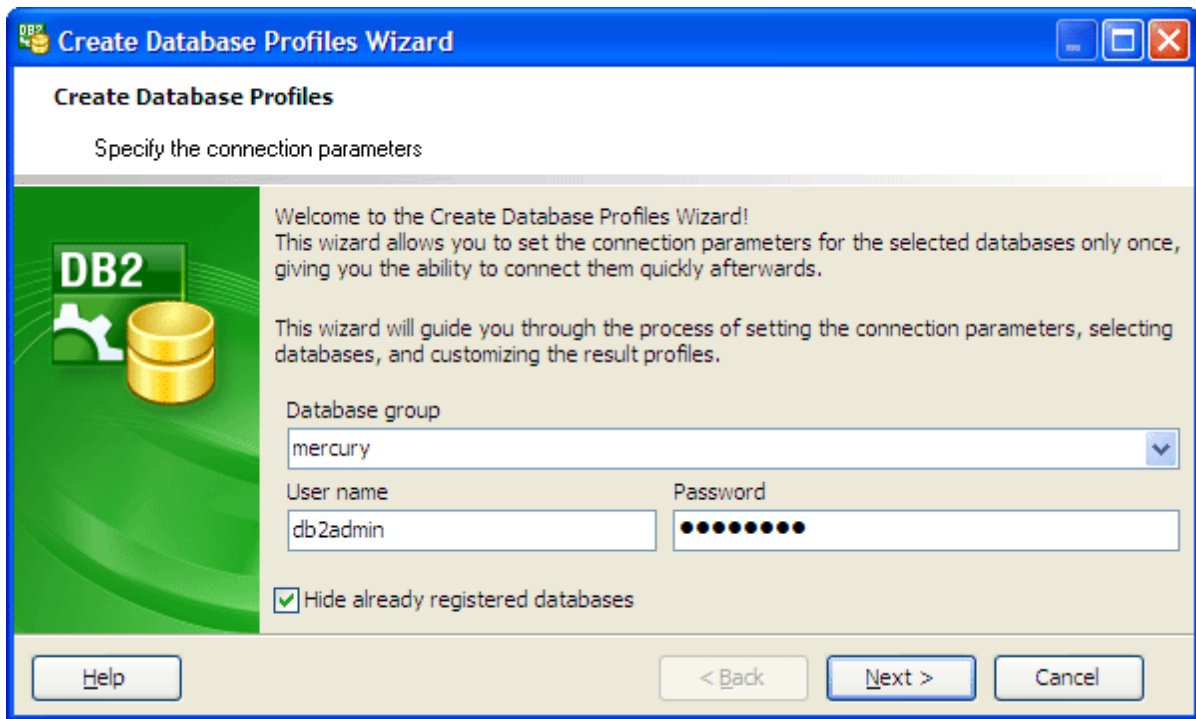
Create Database Profiles Wizard allows you to create a single database profile or several profiles from one host. To run the wizard, select the [Database | Create Database Profiles...](#) main menu item, or press the **Shift+Ctrl+P** hot keys combination. You can also use the [Create Database Profiles](#) button of the main toolbar.

- [Set connection properties](#)<sup>[25]</sup>
- [Specify database profile options](#)<sup>[25]</sup>

**See also:** [Edit Database Profile Dialog](#)<sup>[27]</sup>

### 3.1.1 Setting connection properties

Specify DB2 [connection properties](#)<sup>[14]</sup> to be used on further connections.

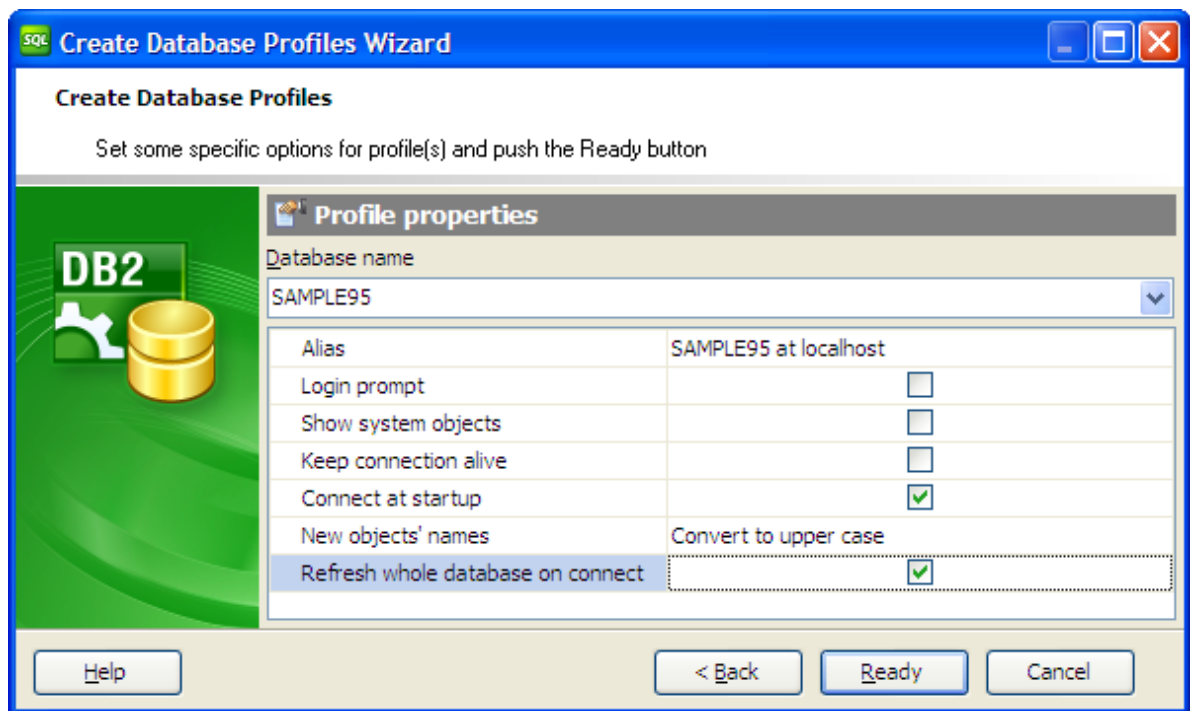


☒ Hide already registered databases

Check the box to shorten the databases list on the next wizard step.

### 3.1.2 Setting profile options

Set the alias and other profile options (the description is given below). Click the [Ready](#) button when done to start working with the selected databases in DB2 Maestro.



☒ **Login prompt before connection**

Use the option to enable DB2 Maestro to prompt for user name and password every time you connect to the database.

☒ **Show system objects**

Check the box to make system objects visible.

☒ **Keep connection alive**

Check the box for pinging server before each query execution.

☒ **Connect at startup**

With this option on connection to the profile database is automatically established at the application startup.

**New objects' names (Don't change case, Convert to upper case, Convert to lower case)**  
The option allows you to specify the newly created objects case.

☒ **Refresh whole database on connect**

Use the option along with the [Show empty schemas](#) explorer options to hide/show empty schemas in the explorer tree.

**Profile text color**

Select the color to be used to represent the database profile name at the Explorer tree. For example this option may be useful to mark development and production databases in different colors in order to prevent casual metadata or data changes in the production.

## 3.2 Editing Database Profile

Use the [Edit Database Profile](#) dialog to edit the profile properties set on its creation. To open the dialog, select the database in the explorer tree, then select the [Database | Edit Database Profile...](#) main menu item or press the **Shift+Ctrl+E** hot key combination. You can also use the [Edit Database Profile](#) button of the main toolbar.

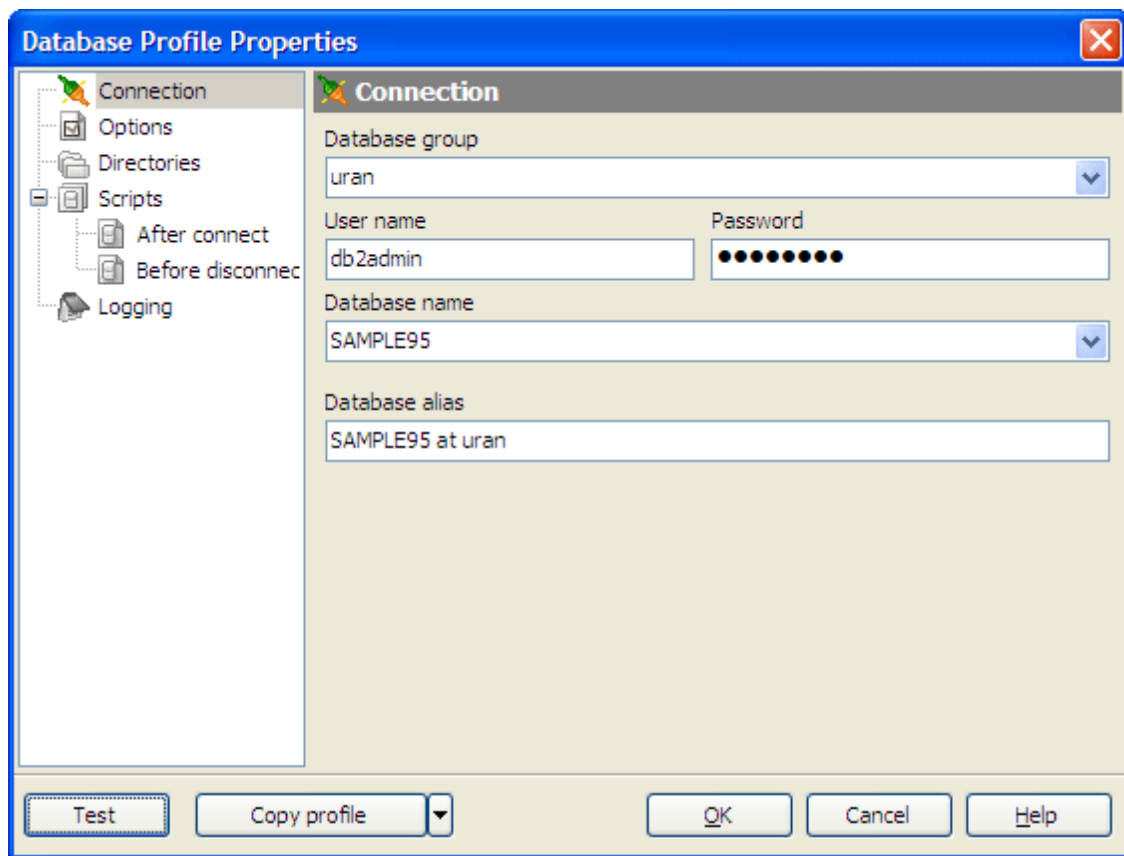
Instead of manual profile options editing you can copy all the options from the another existing profile with the [Copy profile](#) button.

- [Editing database connection properties](#)<sup>[27]</sup>
- [Settings database options](#)<sup>[28]</sup>
- [Setting default directories for database tools](#)<sup>[30]</sup>
- [Editing obligatory scripts to execute](#)<sup>[30]</sup>
- [Setting log options and file names](#)<sup>[32]</sup>

**See also:** [Create Database Profile Wizard](#)<sup>[25]</sup>

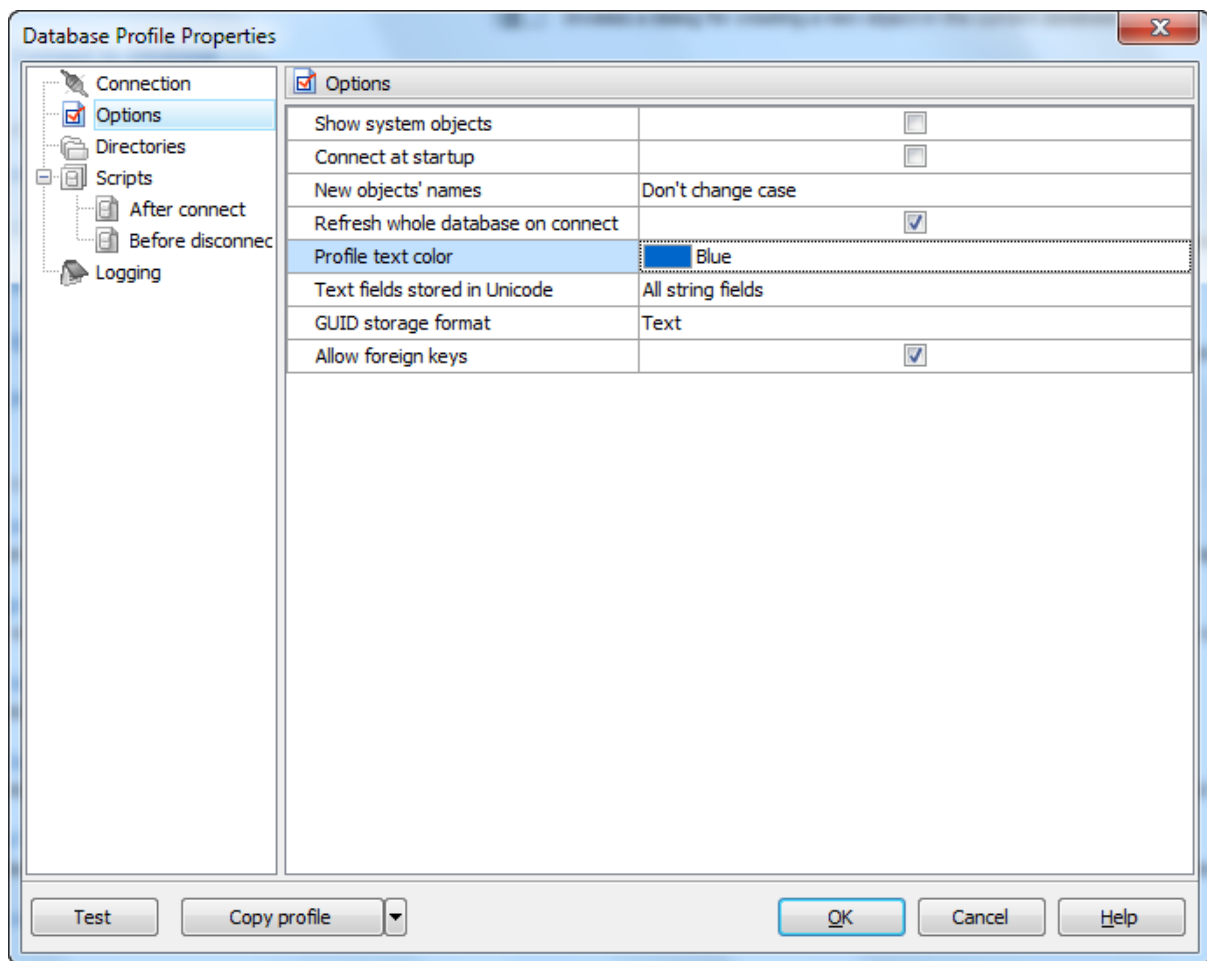
### 3.2.1 Editing connection properties

The tab allows you to change [connection properties](#)<sup>[14]</sup> of an existing database profile. Here you can change the database group, database info and edit the database alias, an optional name to display the database in the Explorer tree and in all the application tools.



### 3.2.2 Setting profile options

Customize database options according to your needs. The detailed description is given below.



☒ **Login prompt**

Use the option to enable DB2 Maestro to prompt for user name and password every time you connect to the database.

☒ **Show system objects**

Check the option to make system objects visible.

☒ **Keep connection alive**

Check the box for pinging server before each query execution.

☒ **Connect at startup**

With this option on connection to the profile database is automatically established at the application startup.

**New objects' names (Don't change case, Convert to upper case, Convert to lower case)**

Use the option to change the case for newly created objects.

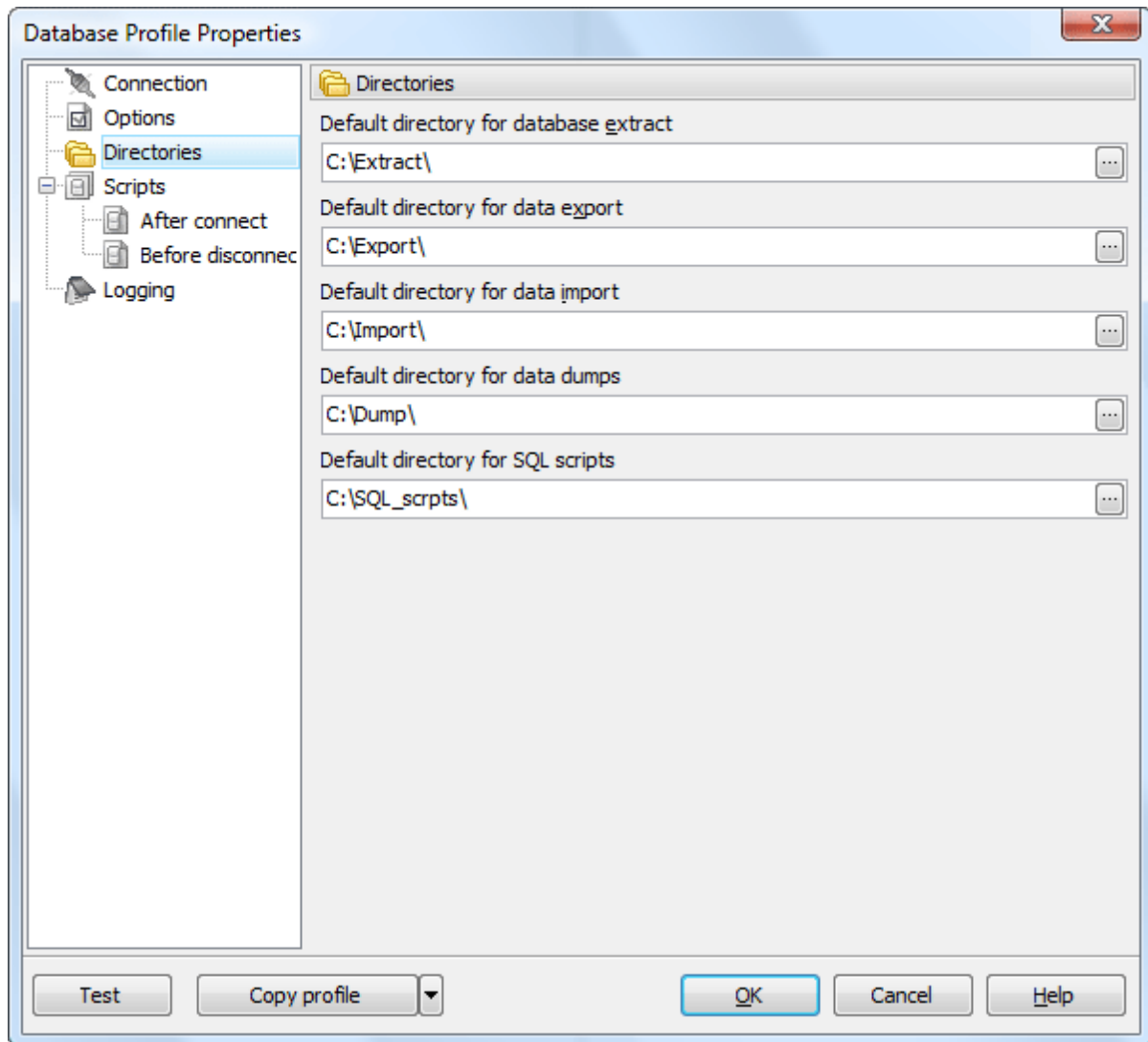
☒ **Refresh whole database on connect**

Use the option along with the [Show empty schemas](#) <sup>298</sup> explorer options to hide/show empty schemas in the explorer tree.

You can also change here the font color the profile name is represented at the Explorer tree.

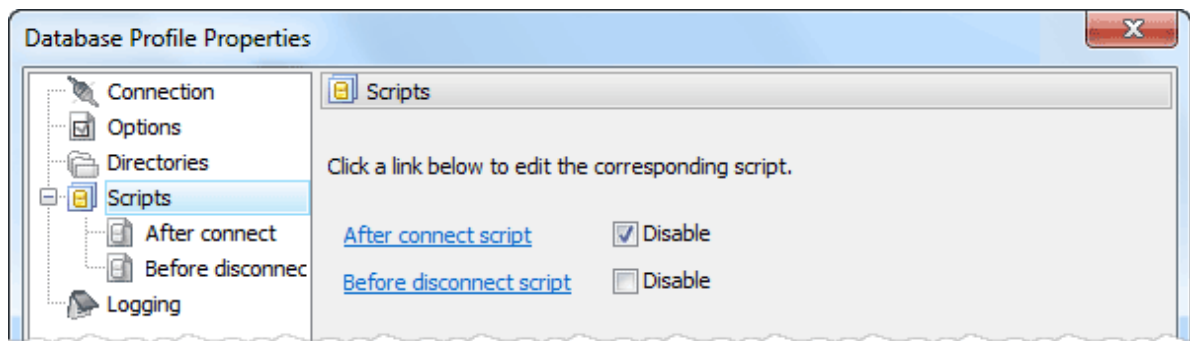
### 3.2.3 Setting default directories

Use the tab to specify the [default directories](#) respectively for database extract, data export, data import, and data dump.

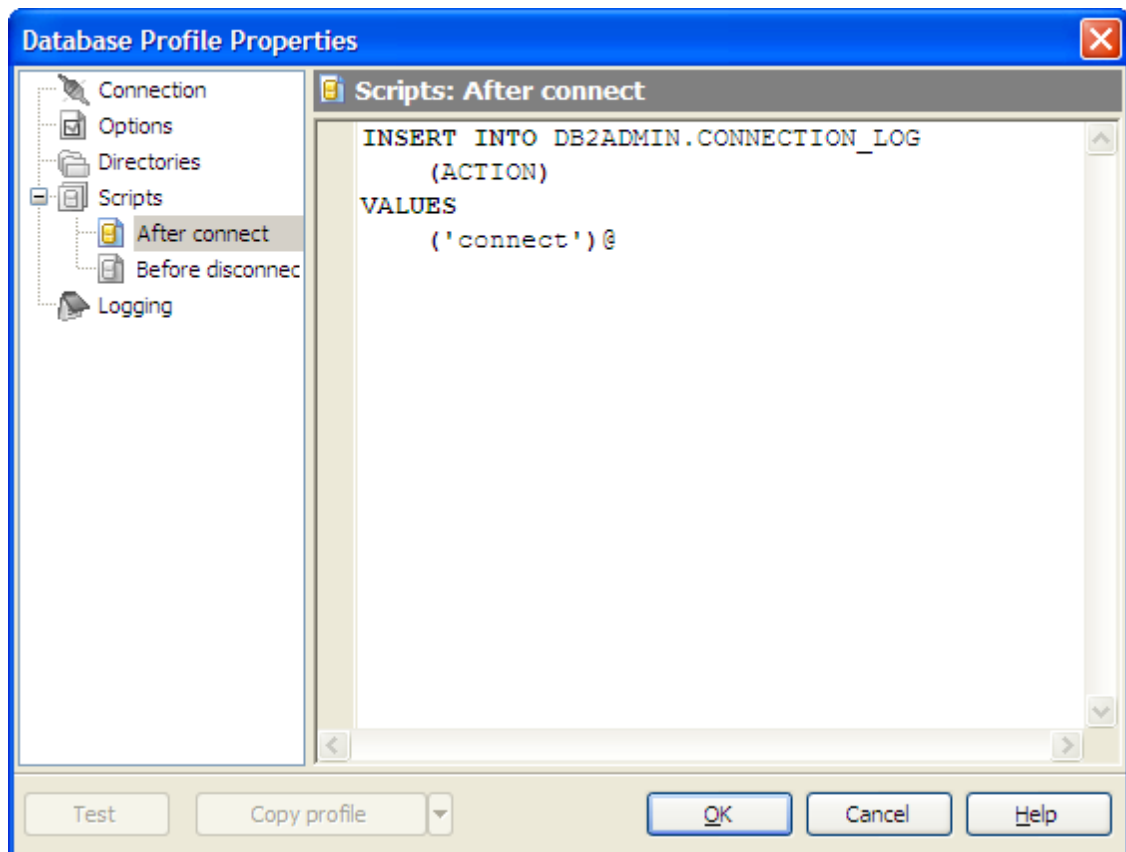


### 3.2.4 Editing obligatory scripts to execute

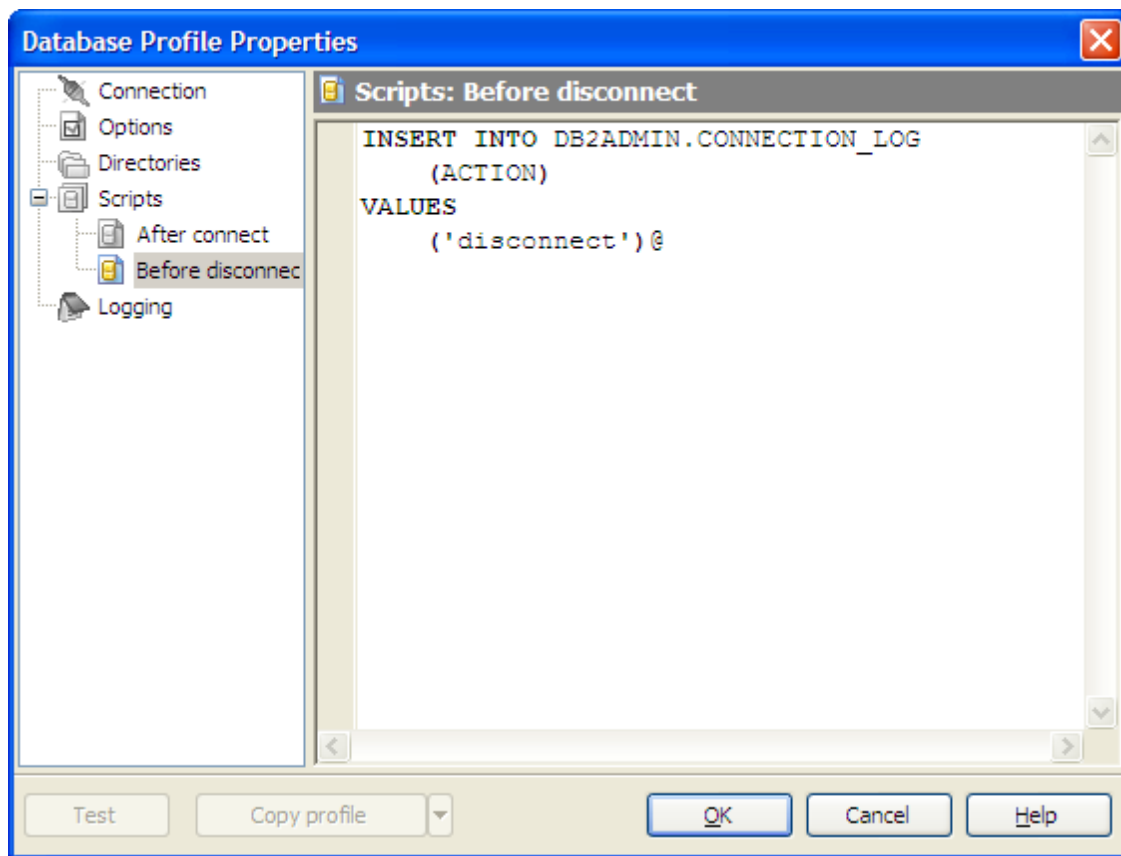
Use the tab to specify the obligatory scripts to execute after DB2 Maestro will connect to the database or before disconnect from the database. You can also enable/disable a written script.



Below you can find an example of an obligatory script to execute after DB2 Maestro will connect to the database. The script writes a connect time to the log table.

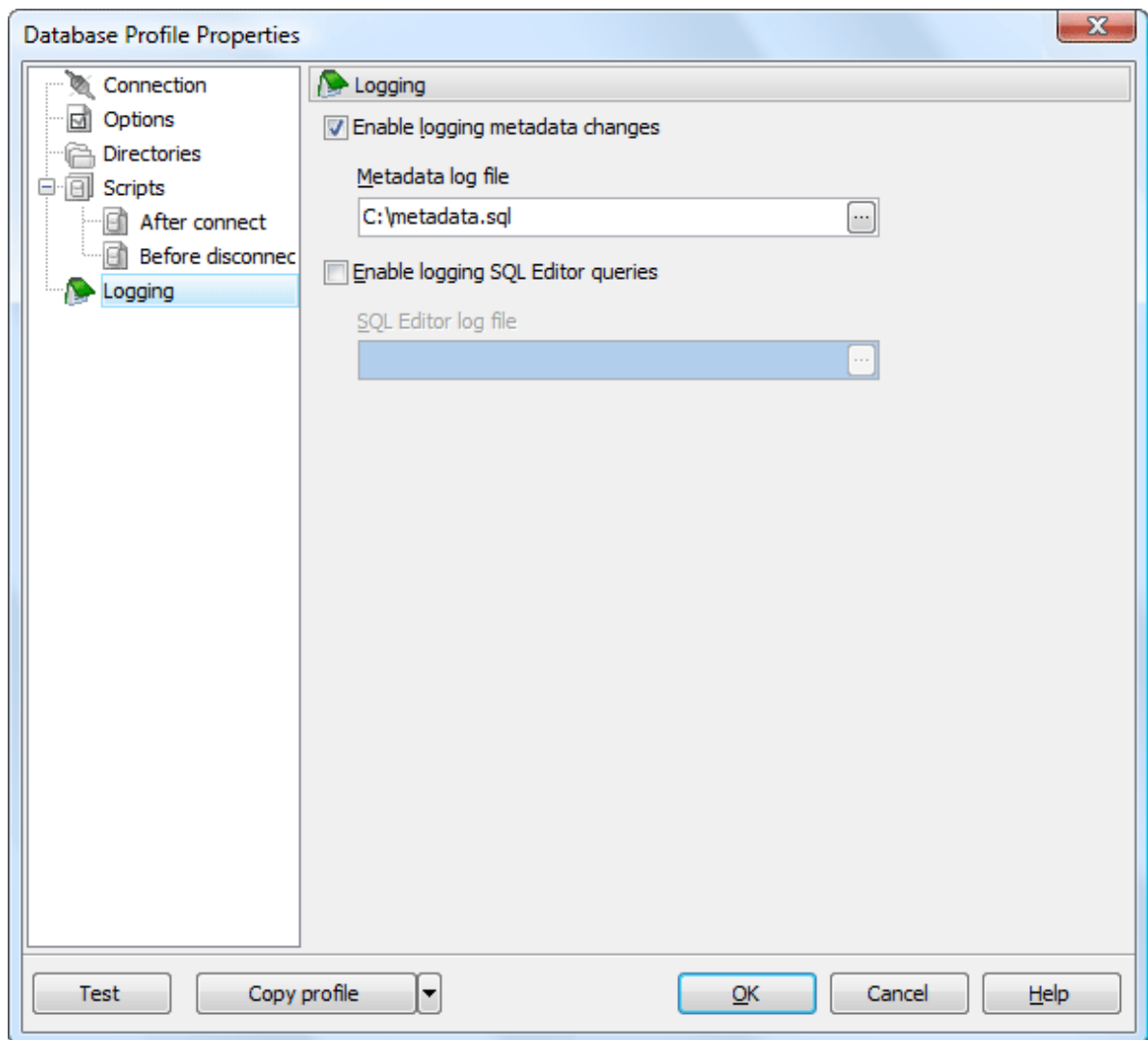


The next screen represents the example of an obligatory script to execute before DB2 Maestro will disconnect from the database. The script writes a disconnect time to the log table.



### 3.2.5 Setting log options

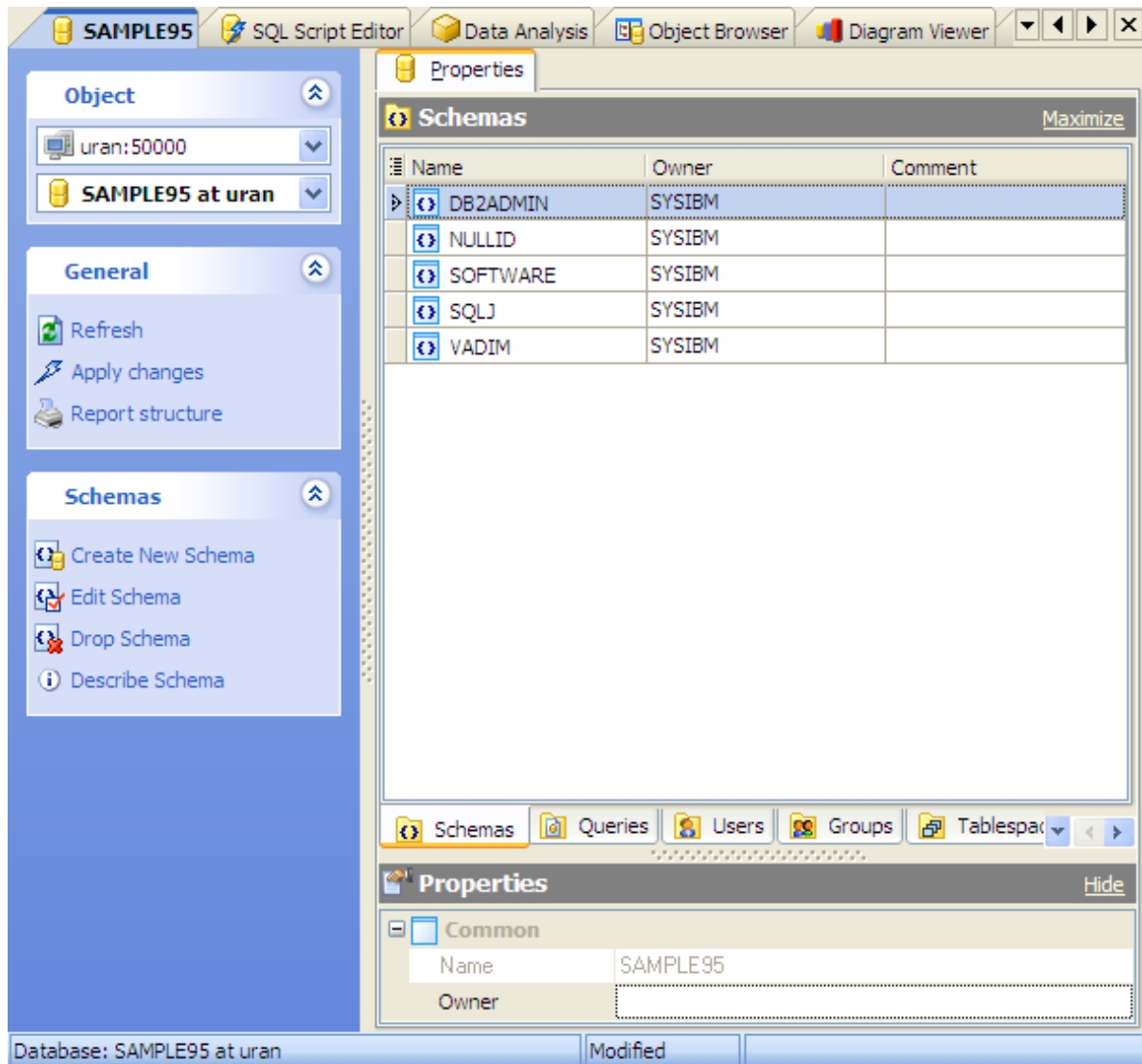
Enable/disable [metadata changes logging](#) and [SQL query logging](#) and specify the corresponding log file names if necessary.



### 3.3 Database Editor

Database Editor allows you to browse, add, edit and delete all objects of the selected database and its main properties.

To open the editor, use popup menu of the database node at the Explorer tree.



#### Subitems

Every tab is intended for managing corresponding database objects (e.g. *tables*, *views*, *queries*, etc.). Open the object in its editor by double-clicking or pressing the **Enter** key. The popup menu allows you to create new, edit or drop the selected database objects. Using this menu you can also create a copy of the object.

You can operate on several objects at a time. For this you have to select database objects with the **Shift** or the **Ctrl** key pressed. After the group of objects is selected, you can operate on it, e.g. delete several objects at once, as it was a single object.

## 4 Work with Objects

DB2 Maestro provides you with several tools to manage and navigate DB2 objects. To browse and modify objects, at least one connection to a database should be established.

- Browse Database Objects
- [Create New Objects](#) <sup>36</sup>
- [Edit Existing Objects](#) <sup>39</sup>
- [Duplicate Objects](#) <sup>45</sup>

The options to create or edit an object in DB2 Maestro follow the parameters defined by DB2. If you need clarification on what an option means or how it should be used, see DB2's documentation for more information. The documentation provides detailed description of objects, including their purpose, properties, and restrictions. The DB2 Maestro manual provides you with only brief review of DB2 objects.

## 4.1 Create Objects

DB2 Maestro provides a number of [Create Object Wizards](#)<sup>[36]</sup> to accomplish the most facile DB2 object creation.

There are several ways to invoke the necessary Create Object Wizard:

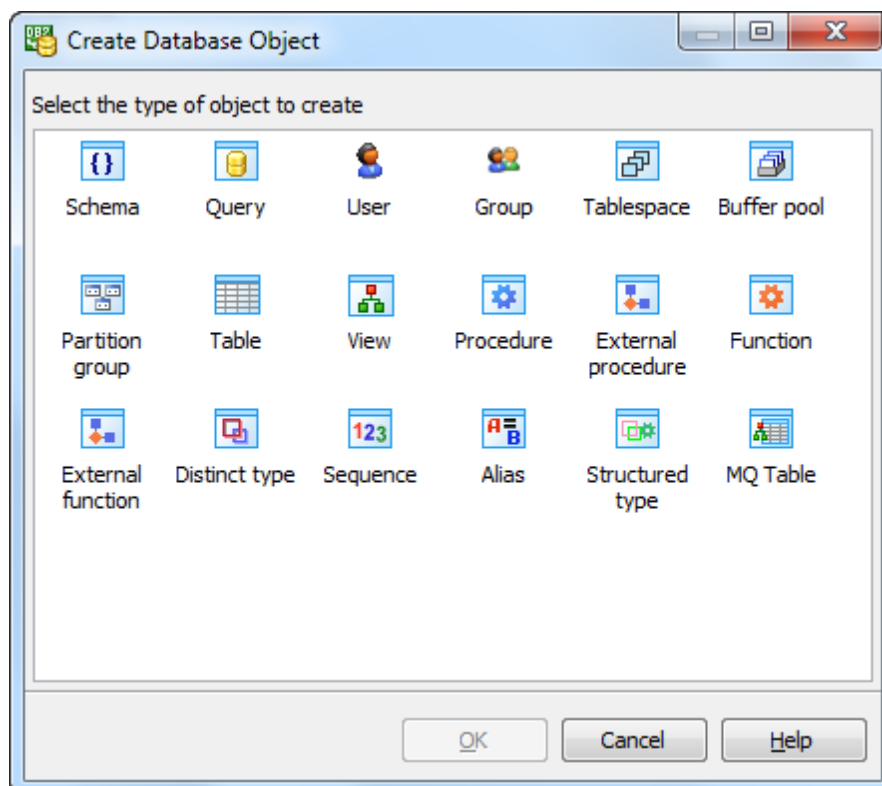
- select the **Object | Create Database Object...** main menu item;
- select the necessary icon (table, query, view, trigger, etc.) in the [Create Database Object](#)<sup>[36]</sup> dialog

or

- select the object list (Tables, Views, Triggers, etc.) or any object from that list in the Explorer tree (**Object Manager** and **Object Browser**);
- select the **Create New Table (View, Trigger, etc.)...** item from the popup menu or press **Insert**.

### 4.1.1 Create Database Object Dialog

The [Create Database Object](#) dialog allows you to create any type of database object supported by DB2 Maestro. To open the dialog select the **Object | Create Database Object...** main menu item or use the **Shift+Ctrl+N** hot keys combination. Select an object type icon and click the **OK** button to invoke the corresponding wizard or dialog.



### 4.1.2 Overview of Create Objects Wizards

Several steps of Create Object Wizards are common for all of them. This part purpose is the formulation of the basic principles for the [Create Object Wizard](#) organization.

- On the [first wizard step](#)<sup>[37]</sup> you need to specify the new object name.
- On the second one you have to define all the object properties. To clear up the object properties meanings see the appropriate topic of the respective Create Object Wizard section.
- Some objects has subitems (e.g. each table contains fields, indexes, procedures have parameters, etc). In this case the next step allows you to manage such subobjects of the object being created. We recommend you to store the following shortcuts in order to speed your work: the Ins key adds a new subobject, the Enter key displays the subobject's editor, and the Del key drops the subobject.
- The [next wizard step](#)<sup>[38]</sup> is final. It is provided to sum up the [Create Object Wizard](#) operation.

**Note:** There are some objects to have an additional [Create Object Wizard](#) steps. The detailed description of the steps you can find at the appropriate topic of the corresponding section.

**See also:**

- [Create Schema Wizard](#)<sup>[60]</sup>
- [Create Table Wizard](#)<sup>[66]</sup>
- [Create View Wizard](#)<sup>[92]</sup>
- [Create Function Wizard](#)<sup>[116]</sup>
- [Create Distinct Type Wizard](#)<sup>[124]</sup>
- [Create Sequence Wizard](#)<sup>[137]</sup>
- [Creation of a New Query](#)<sup>[194]</sup>
- [Create Procedure Wizard](#)<sup>[102]</sup>
- [Creation of External Functions and Procedures](#)<sup>[110]</sup>
- [Create Structured Type Wizard](#)<sup>[142]</sup>
- [Creation of a Materialized Query Table](#)<sup>[158]</sup>

#### 4.1.2.1 Setting object name

Select the container (table, schema, database, etc.) for the new object from the list of available containers and enter the new object [name](#) in the respective box.

**Note:** the name of the object must be unique among all the object names in its container. Moreover, all the objects that are source of data need unique names among themselves. You can use any identifier that is allowed by DB2 server.

Welcome to the Create Table Wizard!  
This wizard allows you to set up all the table properties, and create the new table according to these settings.

This wizard will guide you through the process of building the initial table structure and setting its properties.

Schema  
NBA

Table name  
PLAYER

#### 4.1.2.2 Viewing common information

At this step common information about the object to be created is displayed. Select the [Open object editor after creating option](#) to open the appropriate [Object Editor](#) after the new object is created. Click the [Ready](#) button to complete creation of the object.

The following schema is selected:  
NBA

The following table will be created:  
PLAYER

☒ Open table editor after creating

Click "Ready" to create a new table.

## 4.2 Edit Objects

DB2 Maestro allows you to view and modify existing database objects in several ways:

- edit object comment with the [Describe Object](#)<sup>[44]</sup> dialog;
- briefly view and modify [object properties](#)<sup>[44]</sup>;
- view and modify the object including subitems within the object editor.

To open an [Object Editor](#)<sup>[39]</sup>, just double click its node in the [Database Explorer](#) tree. Of course this action is also available through popup menus, navigation bars, and so on.

### 4.2.1 Overview of Object Editors

[Database Object Editors](#) are the basic DB2 Maestro tools for working with existing objects. The proper editor can be opened automatically after the object is created and is used for specifying and editing necessary object properties. You can also open the necessary object editor within the [Explorer Tree](#) or [Object Manager](#).

The editors consist of a several tabs. Several tabs are similar for all editors. This part purpose is the formulation of the basic principles for the [Object Editor](#) organization.

- To edit main table properties such as name, owner, comment, and others use the [Properties](#) tab. To clear up the object properties meanings see the appropriate topic of the respective [Object Editor](#) section.
- In case the object has subitems (e.g. each table contains fields, indexes, procedures have parameters, etc), its editor contains the corresponding tab named [Subitems](#). You can add, edit or drop subobjects using the grid's popup menu or editor's navigation bar. To change the subitems' options, use the corresponding editor. For example, to modify function or procedure parameter, use [Parameter Editor](#)<sup>[43]</sup>.
- Use the [Permissions](#)<sup>[40]</sup> tab to manage access privileges (grants) of the corresponding object.
- In a similar manner, some objects called grantees (e.g. users or roles) can have rights to do something with other objects (e.g. a user can read data from a table). This relationship can be set up at the [Grants](#)<sup>[40]</sup> tab.
- Most of objects have a possibility to be created from an SQL script (SQL definition). If so, the corresponding script is available at the [SQL](#)<sup>[42]</sup> tab of the editor.
- There is a [Result](#) tab in editors of such routines as functions and procedures that can take parameters, perform calculations or other actions, and return a result. You can [execute](#)<sup>[43]</sup> any routine directly from its editor.

**Note:** Some object editors have additional tabs. The detailed description of them you can find at the appropriate topic of the corresponding section.

To apply the changes, select the [Apply Changes](#) item in the [Navigation bar](#) or use **Ctrl**

**+F9** or **Ctrl+F7** shortcut keys.

It is also possible to modify object properties without opening the object editor: use the [Object Properties](#) item of the popup menu of the selected object from the explorer tree.

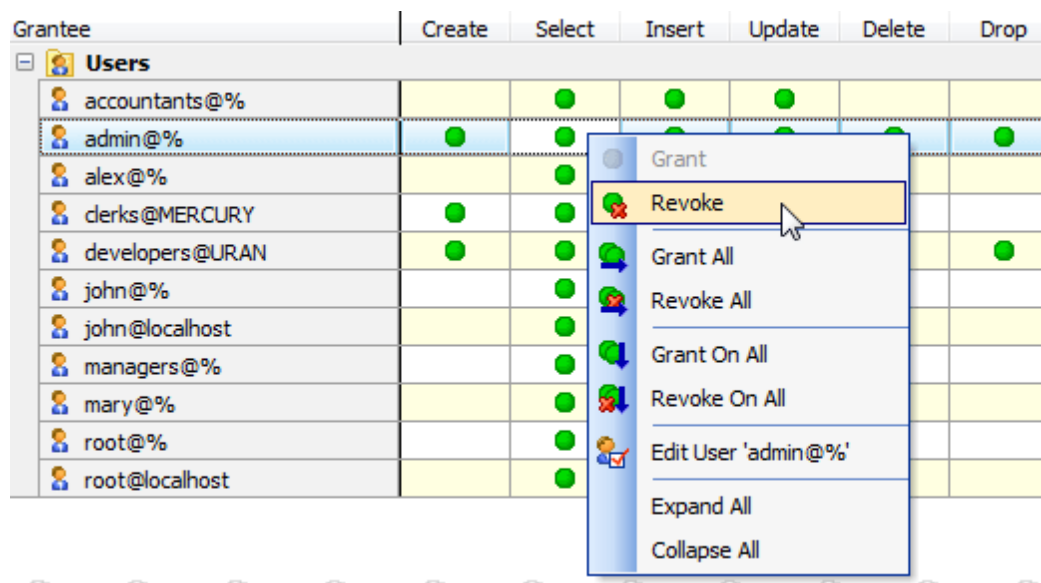
**See also:**

- [Schema Editor](#)<sup>[62]</sup>
- [Table Editor](#)<sup>[66]</sup>
- [View Editor](#)<sup>[97]</sup>
- [Function Editor](#)<sup>[119]</sup>
- [Distinct Type Editor](#)<sup>[126]</sup>
- [Sequence Editor](#)<sup>[138]</sup>
- [Procedure Editor](#)<sup>[106]</sup>
- [Editing of External Functions and Procedures](#)<sup>[112]</sup>
- [Structured Type Editor](#)<sup>[146]</sup>
- [Editing of a Materialized Query Table](#)<sup>[164]</sup>

#### 4.2.1.1 Permissions of the Object

The [Permissions](#) grid allows you to manage access privileges (grants) of users.

Grants give specific privileges for an object to one or more users.



Using the grid you can grant/revoke privileges as well as sort and filter displayed grantees.

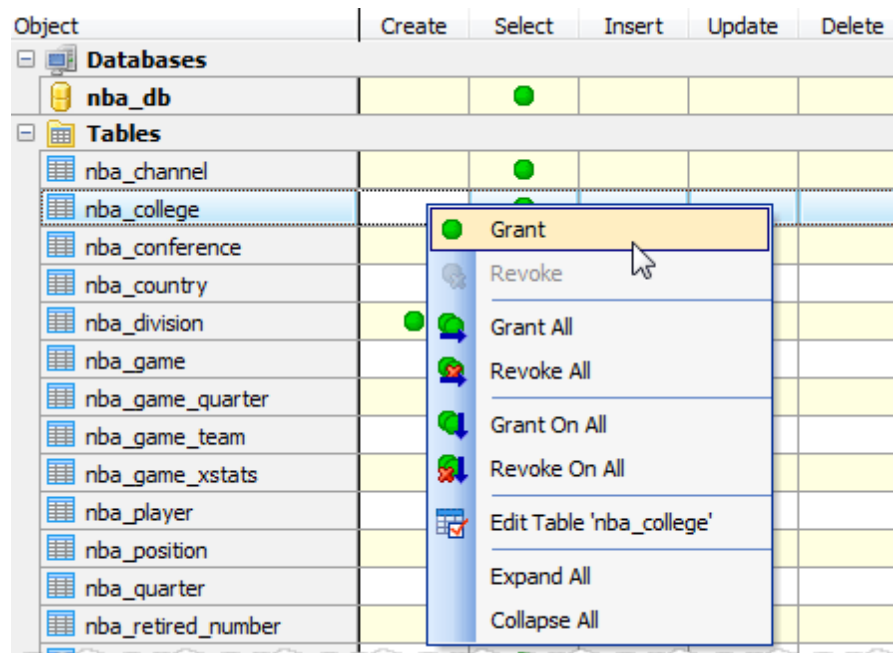
**See also:**

#### 4.2.1.2 Object grants

The Grants grid allows you to manage access privileges (grants) of the current object.

A grant gives specific privileges on an object (*table*, *view*, *procedure*) to the current one.

All objects are grouped by kind. Filter the object kinds using the checkboxes at the bottom of the window. Using the grid you may sort and filter data.



To grant the subject privilege on the object double-click an empty field; to revoke the privilege double-click a grant with grant option.

Use grid's popup menu to *grant*, *grant all*, *grant with grant option*, *grant all with grant option*, *grant on all*, *grant on all with grant option*, *revoke*, *revoke all* and *revoke on all*:

- select the **Grant** item to grant the subject privilege on the object;
- select the **Grant All** item to grant all the privileges on the object;
- if the **Grant With Grant Option** item is selected, the recipient of the privilege may in turn grant it to others (without a grant option specified, the recipient cannot do that; at present, grant options can only be granted to individual subject, not to groups or Public);
- select the **Grant All With Grant Option** item to grant with grant options the privilege on all the objects of the kind;
- select **Grant On All** or **Grant On All With Grant Option** to grant or grant with grant options respectively the subject privilege on all the objects;
- to revoke the privilege, all the privileges on the object or the privileges on all the server objects select the **Revoke**, **Revoke All** or **Revoke On All** items respectively.

Using the popup menu you can also collapse or expand all the object kinds.

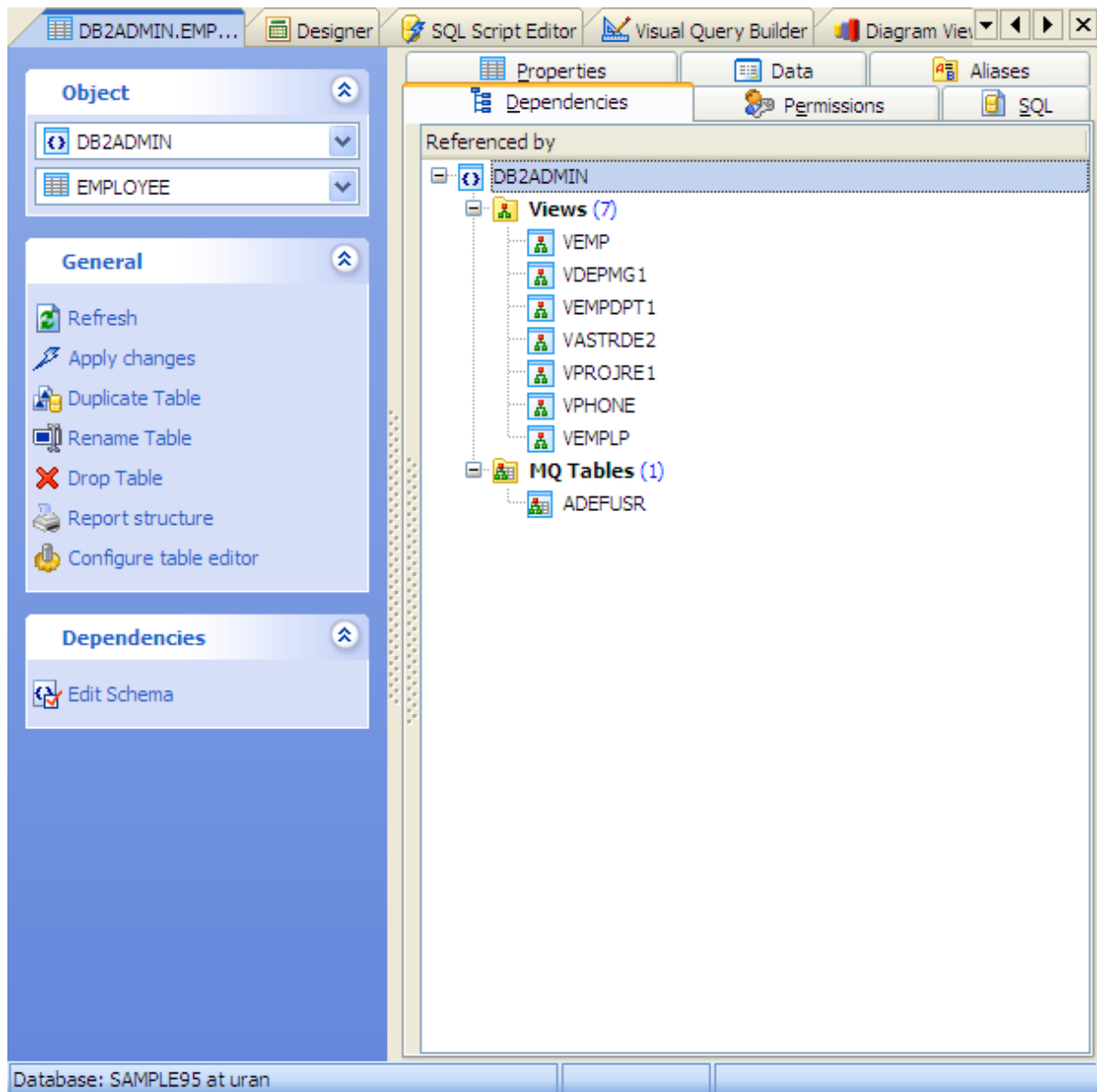
#### 4.2.1.3 Object Dependencies

When you create complex database structures involving many tables with foreign key constraints, views, triggers, Functions, etc. you will implicitly create a net of dependencies between the objects. For instance, a table with a foreign key constraint depends on the table it references. We tried to create a software to make your work easier, i.e. to give you the tools for efficient database objects management. The **Dependencies** tab allows you to control the correlation of objects efficiently.

The **Depends on** window represents all the objects the current object depends on.

The **Referenced by** window contains the tree of database objects constituting the dependency relationship on the current object.

Use items of the popup menu to edit the selected object in **Object Editor** or to drop the object.



#### 4.2.1.4 SQL Definition

The **SQL** tab displays the SQL definition for the object with all its properties. Bear in mind that this text is read-only. If you want to change the object definition, use the appropriate editor tabs instead, or copy the text to the Windows Clipboard to paste it in **SQL Editor** or **SQL Script Editor**.

The SQL definition window allows you to browse the text effectively. The popup menu and the extensive system of hot keys give you the opportunity to search expressions within the text, to select the whole text for copying it to the Windows Clipboard, to save the definition to the \*.sql or \*.txt files, to print the document, etc.

You can customize the displayed definition using the [Editors & Viewers](#)<sup>315</sup> options.

The [Properties](#) item of the popup menu displays the [Options](#) dialog in which you can establish optional settings concerning the current database.

The [Code Folding](#) item group makes it possible to view either the whole text or its logical parts (regions). Each region can be collapsed and extended.

In [extended mode](#) the whole text is displayed (set by default)

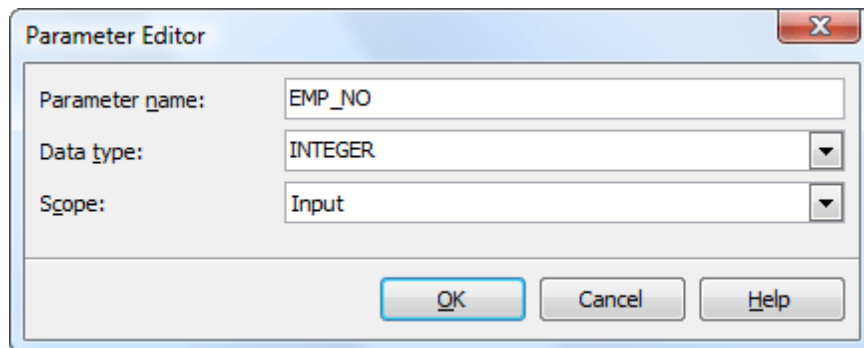
In [collapsed mode](#) the text is hidden behind one text line denoting the first line of the collapsed region.

[Navigation Bar](#) on the [SQL](#) tab allows you to copy the object's SQL definition (DDL) to the [SQL Script Editor](#) for future modifications.

#### 4.2.1.5 Parameter Editor

The editor allows you to change the [Parameter name](#), [Data type](#).

Use the [Scope](#) box to set whether parameter is *Input*, or *Output*, or *Input-Output*.



#### 4.2.1.6 Executing functions and procedures

[Editor](#) provides an opportunity to execute current routine by opening the [Results](#) tab, by clicking the [Execute](#) item of the [Navigation Bar](#), or by pressing the **F9** key.

If the has parameters, DB2 Maestro will ask you to specify the values for these parameters in the [Input parameters](#) dialog which appears before the procedure execution. [Input parameters](#) dialog allows you to specify the values for all input parameters. After changes are made, click the [OK](#) button to execute the Function, or the [Cancel](#) button to abort the execution.

Parameter Values	
(↕) @conference_id int	2
(↕) @division_id int	2
(↕) @team_id int	1
</	

DB2 Maestro supports [Parameter History](#). Values that have been set previously as the routine parameters are represented in the [History](#) tab of the [Input Parameter](#) dialog with a date and time of their last using. Double click a necessary set of values to set them as the routine parameters. You can manage the [Parameter History](#) with [Delete history item](#) and [Clear history](#) links.

The result of the successfully executed routine can be found within the [Results](#) tab of [Editor](#).

**Note:** If any unsaved changes are applied to the routine being currently edited, the execution of the routine is impossible until changes are saved by the [Compile](#) procedure item of the [Navigation Bar](#).

## 4.2.2 Modify Object Properties

You can rename all objects those can be renamed with the corresponding option of the popup menu of the object at the Explorer tree. To edit other properties of the selected object without opening its editor, use the [Object Properties](#) dialog. To open this dialog, select the according item of the same popup menu. To clear up the object properties meanings, see the appropriate topic of the respective [Object Editor](#) section.

## 4.2.3 Describe Objects

Essentially a comment is the most often altered object property. To simplify it's editing, the DB2 Maestro provides an ability to [Describe the object](#) within the [Database Explorer](#) immediately without opening of the object's editor.

Step-by-step:

- Select the necessary object in the explorer tree;
- Choose the [Describe Object...](#) item in the popup menu;
- Edit object comments within the [Describe Object](#) window;
- To commit the changes, push [OK](#) button.

## 4.3 Duplicate Objects

DB2 Maestro offers several ways of objects duplicating.

1. **Duplicate Object Wizard.** The wizard is the most flexible tool of the coping. Along with a possibility to adjust the new object definition it allows you to copy data (for tables). But it consists of [several steps](#)<sup>[45]</sup> and takes more time than other manners.
2. **Duplicate Object** window allows you to attune new object's SQL definition. It is preferred for creation a copy of selected object. [Here](#)<sup>[47]</sup> you can find some additional info.
3. By [Drag-n-Drop](#)<sup>[48]</sup> operation.

### 4.3.1 Duplicate Object Wizard

The [Duplicate Object Wizard](#) allows you to create a new database object with the same properties as the existing one. It is the most flexible tool of copying objects provided by DB2 Maestro. It also allows you to copy data of the selected table to the new one.

To run the wizard select the [Object | Duplicate Database Object...](#) main menu item.

- [Selecting source and destination databases](#)<sup>[46]</sup>
- [Selecting object to duplicate](#)<sup>[46]</sup>
- [Modifying definition of a new object](#)<sup>[47]</sup>

**See also:** [Create Database Object](#)<sup>[36]</sup>

#### 4.3.1.1 Selecting source and destination databases

Select the database containing a source object from the list of connected databases, and then specify the database for the duplicated object.

You should connect to the destination database beforehand (see [Database Management](#)<sup>[23]</sup>).

Welcome to the Duplicate Object Wizard!  
This wizard allows you to create a new table or other database object with the same properties as one of the existing objects has.

This wizard will request the name of the new object and of the source object, generate the SQL statement for creating the new object, and execute this statement.

Source database  
nba\_db at localhost

Destination database  
nba\_db\_new at sun

#### 4.3.1.2 Selecting object to duplicate

Specify a database object to create the new one with the same properties.

1. Select the type of the object to duplicate from the **Source objects** drop-down list.
2. Pick up the necessary object from the list.

Source objects  
Tables

Objects

nba_channel	nba_season
nba_college	nba_season_team
nba_conference	nba_stats_figure
nba_country	nba_team
nba_division	
nba_game	
nba_game_quarter	
nba_game_team	
nba_game_xstats	
nba_player	
nba_position	
nba_quarter	
nba_retired_number	
nba_round	

#### 4.3.1.3 Modifying new object definition


The last wizard step allows you to edit the new object definition directly.

Use this step to edit the name of object been creating ([New Object Name](#)). By default DB2 Maestro generates the new object definition with the same name if the duplicating is to the source database, or like "%SOURCE\_OBJECT\_NAME%01" otherwise.

**Note:** the name of the object must be unique among all the object names in its container. Moreover, all the objects that are source of data need unique names among themselves. You can use any identifier that is allowed by DB2 server.

You can edit the result SQL statement manually, add or remove fields, change field types, using the [New object definition](#) text area. Click the [Ready](#) button to complete the operation.

Check the according boxes to [Copy Data](#) (only for tables) and to [Open Object Editor](#) after the duplicating.



New object name

nba\_player

Definition

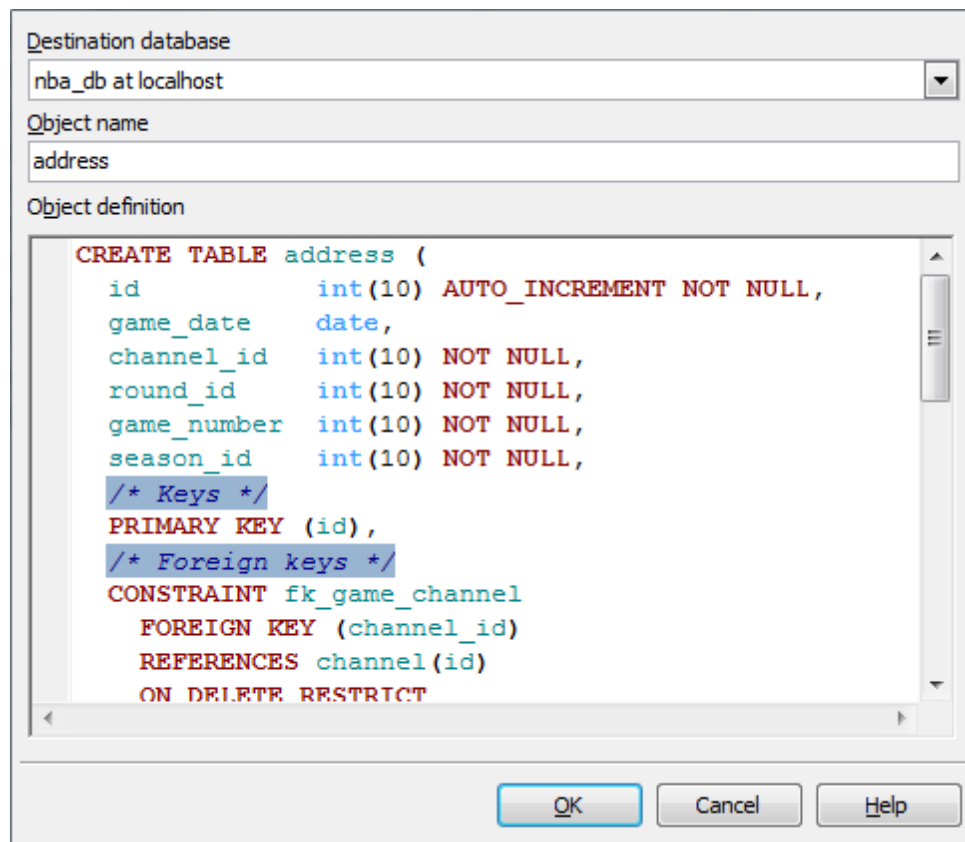
```
CREATE TABLE nba_player (  
  id                int(10) NOT NULL DEFAULT '0',  
  first_name        varchar(50) NOT NULL,  
  last_name         varchar(50) NOT NULL,  
  full_name         varchar(102) NOT NULL,  
  career_start_year int(10) NOT NULL DEFAULT '0',  
  career_end_year   int(10) NOT NULL DEFAULT '0',  
  position_id       int(10) NOT NULL DEFAULT '0',  
  photo             longblob,  
  country_id        int(10) NOT NULL DEFAULT '0',  
  height            int(10),  
  birthday          datetime,
```

☐ Open Object Editor ☒ Copy Data

#### 4.3.2 Duplicate Selected Object

Within the [Duplicate Object](#) window you can duplicate a selected object fast and with some modifications.

It is available from the corresponding link of the object's popup menu at the [Database Explorer](#).



Select the [database](#) for a new object from the list of connected databases first.

Enter the [name](#) for the new object.

**Note:** the name of the object must be unique among all the object names in its container. Moreover, all the objects that are source of data need unique names among themselves. You can use any identifier that is allowed by DB2 server.

You can also edit the SQL [definition](#) of the object if necessary (add or remove fields, change field types, etc.).

### 4.3.3 Copy, Paste and Drag-n-Drop features

DB2 Maestro provides you with an ability of copying database objects within the database or even from one database to another (in this case you should connect to both the source and the destination databases first).

To copy an object, just drag the object in a source window (such as [Database Explorer](#), [Object Manager](#), [Object Browser](#)) and drop it to the target container in another window. You also can use the [Edit | Copy](#) and the [Edit | Paste](#) main menu items or the **Ctrl+C**/**Ctrl+V** hot keys combinations respectively. Copying several objects at a time is also available.

It is also possible to drag and drop objects between [Database Explorer](#), [Object Manager](#), [Object Browser](#) and [SQL Editor or SQL Script Editor](#). This works as follows:

**SQL Editor:** after dropping the object you will get a query to retrieve object data (e.g. `SELECT * FROM table_name`) or the full name of the object if it doesn't contain data (domains, indexes, etc.).

**SQL Script Editor:** after dropping the object you will get its SQL definition if applicable.

**See also:** Database Explorer, Object Manager, and Object Browser

## 4.4 Browse Objects

DB2 Maestro allows to browse objects stored in a Remote Server database in several ways:

- [Database Explorer](#)<sup>[50]</sup>: objects are represented as a hierarchy (grouped by kind and listed under the according DB2 database node, provided with subobjects if exist)
- [Object Browser](#)<sup>[53]</sup>: an extension of explorer with ability to sort, group, filter and multiple select objects.
- [Object Manager](#)<sup>[55]</sup>: an extension of the explorer with ability to select several objects at a time (to copy, drop, etc.)

All tool allows you to drag-and-drop between them and to perform all necessary operations upon database objects.

### 4.4.1 Database Explorer

Database Explorer is the basic feature of DB2 Maestro which allows you to perform practically all necessary operations upon databases and their objects. The Database Explorer area occupies the left side of the DB2 Maestro main window. All the objects at the Explorer tree are grouped by kind and listed under the according DB2 database node.

To start working with a database you need to create its profile first. The conception of database profiles gives you an opportunity to connect to databases in one touch and work with the selected databases only.

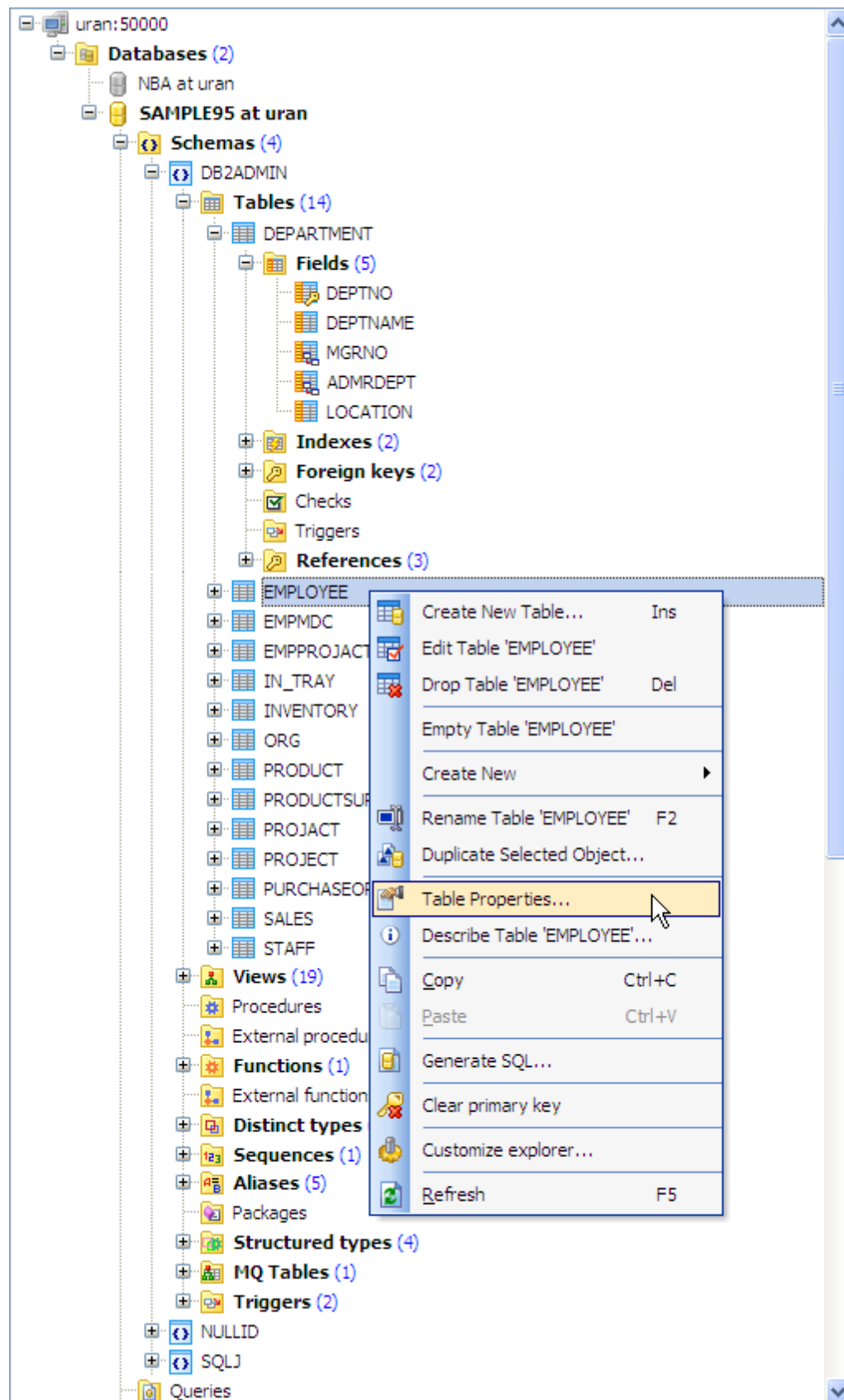
**See also:** [Object Manager](#)<sup>[55]</sup>, [Object Browser](#)<sup>[53]</sup>

**Note:** In case your databases have a large quantity of objects you can speed up the object search by typing first letters of the object name in the explorer area.

**Note:** [Explorer options](#)<sup>[298]</sup> allow you to hide/display table subobjects, represent system objects in different color, etc.



The sections below describe each of these actions in detail.

- [What operation can I accomplish upon database profiles within the Explorer Tree?](#)<sup>[52]</sup>
- [How can I connect to a database?](#)<sup>[50]</sup>
- [How can I disconnect from a database?](#)<sup>[52]</sup>
- [What operations can I accomplish upon database objects within the Explorer Tree?](#)<sup>[52]</sup>
- [Can I copy a database object from one database to another?](#)<sup>[52]</sup>
- [Can I filter Explorer content?](#)<sup>[53]</sup>



### Operations upon database profiles in the Explorer Tree

Using popup menu of the Explorer area you can realize the following operations:

- [create new database profiles](#)  (the Create Database Profiles... item);
- rename currently selected database profile (the Rename Database Profile... item);
- [edit currently selected database profile](#)  (the Edit Database Profile... item);
- reorder existing database profiles (the Reorder Databases...item of Databases node's popup menu or using drag-n-drop);
- reorder servers (the Reorder Servers...item of a server's popup menu);
- remove currently selected database profile from the explorer tree (the Remove Database Profile item);
- remove all profiles of selected server (the Remove all Profiles item of Databases node's popup menu).

In addition to these operations, Database Explorer gives you an ability to reorder existing profiles by performing drag-and-drop operations within the explorer tree.

### How can I connect to a database?

You can establish connection to a database in Database Explorer by selecting the database profile and double-clicking it or pressing the Enter key (alternatively, you may use the Shift+Ctrl+C hot key combination). The same operation is also available through the Connect to Database item from the explorer popup menu, or through the Database | Connect to Database main menu item.

### How can I disconnect from a database?

You can abort connection from a database in Database Explorer by selecting the database profile and pressing the Shift+Ctrl+D hot key combination. The same operation is also available through the Disconnect from Database item from the explorer popup menu, or through the Database | Disconnect from Database main menu item.

### Operations upon database objects

Database Explorer allows you to perform the following operations with database objects using its popup menu (note that the popup menu contains object-specific items only when some database object is currently selected in the explorer tree):

- create a new database object (the Create New Object... item);
- edit currently selected database object (using the Edit Object... item, pressing the Enter key or double-clicking the database object);
- drop the selected object from the database (the Drop Object... item);
- rename the selected database object (the Rename Object... item);
- edit the database object properties (the Object properties ... item);
- duplicate the selected object (the Duplicate Object... item).
- run the Object Browser tool (the Browse ... item).

### Can I copy a database object from one database to another?

Database Explorer provides you with an ability of copying database objects from one database to another. To perform this operation, you should connect to both the source

and the destination databases first. After the connection is established, simply drag and drop an object to copy from the source database to the corresponding node (Tables, Queries, etc.) of the destination database.

**Note:** You also can use the Edit | Copy and the Edit | Paste main menu items to copy/paste a database object using Windows clipboard (alternatively, you may use the Ctrl+C/Ctrl+V hot keys combinations respectively).

#### 4.4.1.1 Filtering explorer content

DB2 Maestro allows you to reduce the number of represented objects in the explorer tree. To hide seldom usable objects, filter your explorer content.

Filter Panel is available through the View | Show Filter Panel main menu item.

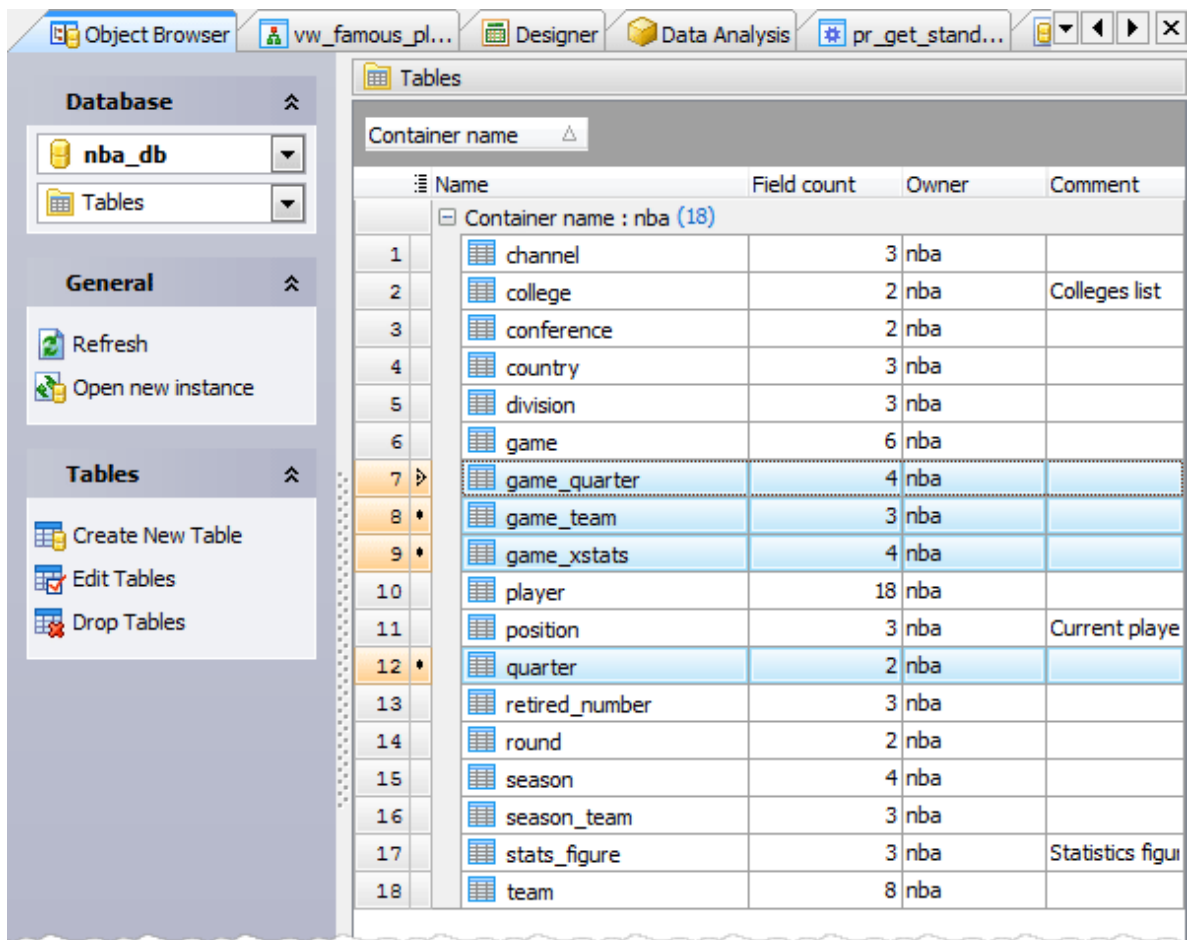
- Specify the Filter expression. The expression can contain any part of object name combined with an asterisk ('\*') as a wildcard character and a question-mark ('?') as a mask character.
- Define the Filtered objects, object types for filtering in the explorer tree.
- Check the according radio button (Show by expression, Hide by expression) to define whether database objects will be shown or hidden in accordance with the filter expression.
- Click Apply button.

**Note:** A filter expression, if applied to the content of Database Explorer, is applied to the content of [Object Manager](#) and [Object Browser](#) as well.

#### 4.4.2 Object Browser

[Object Browser](#) is a tool for operating on database objects designed as an extension of [Database Explorer](#) with ability to *sort*, *group* and *filter* the database objects. It also provides such operation as multiple selecting of objects (for *copying*, *dropping*, etc.) and the ability of using drag-and-drop operations between [Object Browser](#) and [Database Explorer](#). To open [Object Browser](#) select the [Object | Object Browser](#) main menu item.

**Note:** At least one connection to a database should be established to make [Object Browser](#) available.



### Sorting database objects

**Object Browser** represents database objects in a grid. The object kind to display is defined on the top of the **Navigation bar**. The columns correspond to the objects properties and rows correspond to the objects. Click the column caption to sort objects by the values of this column in the ascending or descending mode. The navigation buttons allow you to open current object editor, create new or drop the existing one.

As **Object Manager** the browser allows you to operate on several objects at a time. You have an opportunity to select a batch of objects and after the object group is selected, you can operate on it (e.g. *drop several objects at once*) as if it were a single object.

The unique feature of the DB2 Maestro is an opportunity of drag-and-drop operations between **Object Browser** and **SQL Editor**, **SQL Script Editor**. After the action objects are represented in **SQL Editor** as SQL queries (if they contain data) or as their full name in the database otherwise. **SQL Script Editor** displays the objects as SQL definition.

### Grouping database objects

You can group grid objects by any of the columns by dragging the column header to the destination area. Now all the records are displayed as subnodes to the grouping row value as shown in the picture. To reverse grouping, just drag the column name from the upper area back.

### Filtering database objects

You can filter objects in the grid using one of the following methods:

- use the drop-down button in the column caption area to filter objects by the value of the selected column
- click the drop-down button in the column caption area, then select the [Custom](#) item and build a simple filter within the dialog in the following way: select a logical operator for checking the column values (like is less than, is greater than, etc) and set the value to be checked by this operator in the neighboring box; then set the second condition if necessary in the following way and set the relation between these two conditions, whether both of them should be matched or just one of them; use the '\_' character to represent any single symbol in the condition and the '%' character to represent any series of symbols in the condition

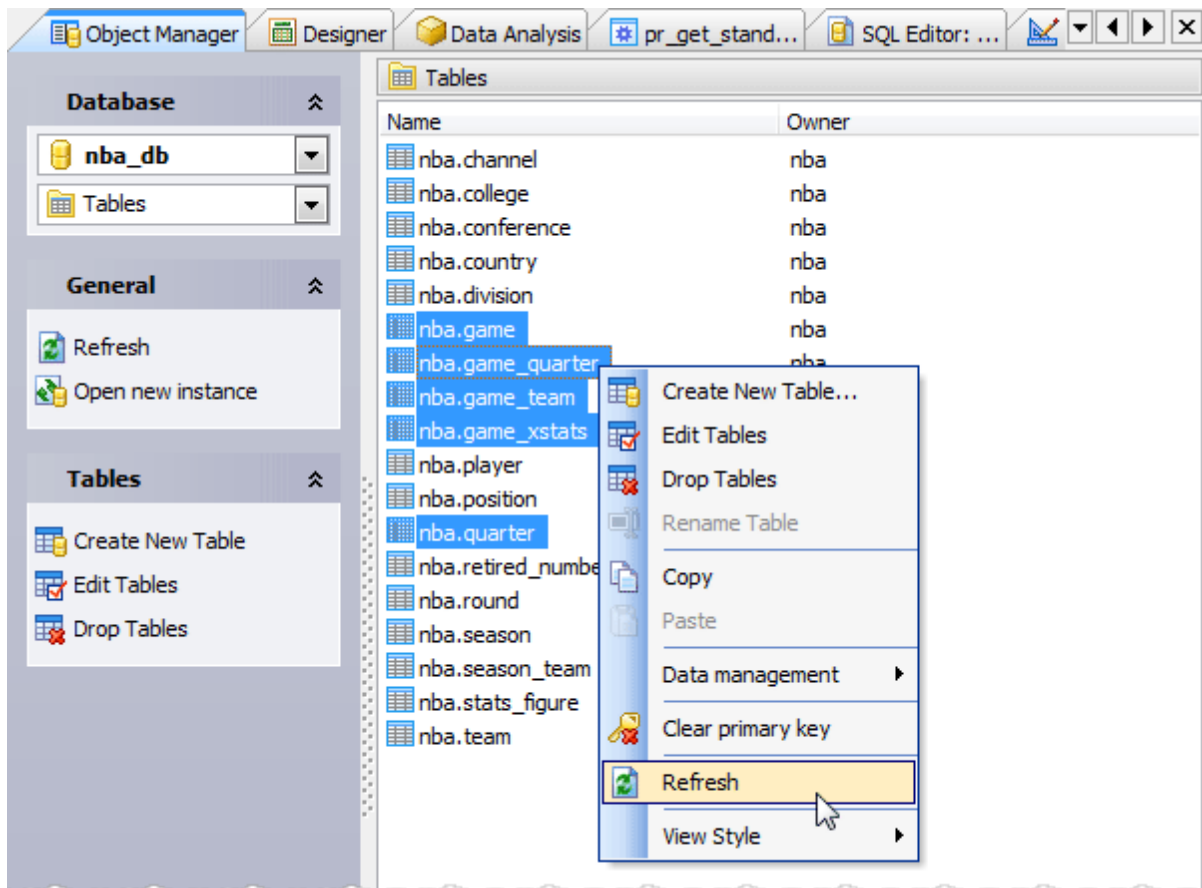
After you set a filter, the filtering panel becomes visible at the bottom of the grid where you can see the active filtering condition and easily enable or disable it by clicking the check box on the left. Using this panel you can also customize your filter in a more complicated way by clicking the [Customize](#) button and building your filter within the Filter Builder dialog.

**See also:** [Object Manager](#)<sup>[55]</sup>, [Data View](#)<sup>[212]</sup>

#### 4.4.3 Object Manager

[Object Manager](#) is a tool for operating on database objects designed as an extension of the [Database Explorer](#)<sup>[55]</sup> with advanced features, such as multiple selecting of objects (for *copying*, *dropping*, etc.) and the ability of using drag-and-drop operations between [Object Manager](#) and [Database Explorer](#) as well as between two instances of the [Object Manager](#). To open [Object Manager](#) select the [Object | Object Manager](#) main menu item.

**Note:** At least one connection to a database should be established to make [Object Manager](#) available.



### Using popup menu

The popup menu of **Object Manager** may have different content depending on the current selection. The common menu items allow you to switch the object list view between four standard modes (*large icons, small icons, list and report*), refresh the current view, and select all the objects in the view. If none of objects are currently selected, other menu items are unavailable to use, except of the one for creating a new object. If one or more objects are selected, clipboard operations (such as copy and paste) become available as well as the items for editing and dropping selected object(s). If the current object type of the Object Manager is "Tables", the *Empty Table(s)* menu item is also available.

### Multiple selecting of database objects

**Object Manager** allows you to operate on several objects at a time. You have an opportunity to select a batch of objects and after the object group is selected, you can operate on it (e.g. *drop several objects at once*) as if it were a single object.

**See also:** [Object Browser](#)

## 4.4.4 Filter Builder Dialog

Filter Builder Dialog allows to limit represented objects according to specified conditions. It may be useful for filtering records in data grids of Table Editors, SQL Editor or Visual Query Builder as well as to filter database objects in Object Browser, and on setting a condition on anew view creating. All these cases are similar, see how it works on the

following example.

## 5 Database Objects

The following list contains database objects supported by DB2 Maestro. To work with database objects you should [connect to the database](#) first.


- [Schemas](#)
- [Tables](#)
- [Views](#)
- [Functions](#)
- [Distinct Types](#)
- [Sequences](#)
- [Procedures](#)
- [External Functions and Procedures](#)
- [Structured Types](#)
- [Materialized Query Tables](#)

## 5.1 Schemas

A [schema](#) is a collection of named objects; it provides a way to group those objects logically. A schema is also a name qualifier; it provides a way to use the same natural name for several objects, and to prevent ambiguous references to those objects; for example, the schema names 'INTERNAL' and 'EXTERNAL' make it easy to distinguish two different SALES tables (INTERNAL.SALES, EXTERNAL.SALES).

Schemas also enable multiple applications to store data in a single database without encountering namespace collisions.

### ■ How can I add a new schema?

New schemas are created within [Create Schema Wizard](#). In order to run the wizard you should either

- select the [Object | Create Database Object...](#) main menu item;
  - select the [Schema](#) icon in the [Create Database Object](#) dialog
- or
- select the [Schemas](#) list or any object from that list in the explorer tree;
  - select the [Create New Schema...](#) item from the popup menu
- or
- open the database in [Database Editor](#) and the [Schemas](#) tab there;
  - press the **Insert** key or select the [Create New Schema](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#)).

To create a new schema with the same properties as one of the existing schemas has:

- select the [Object | Duplicate Database Object...](#) main menu item;
- follow the instructions of [Duplicate Object Wizard](#).

### ■ How can I edit an existing schema?

Schemas are edited within [Schema Editor](#). In order to run the editor you should either

- select the schema for editing in the explorer tree (type the first letters of the schema name for quick search);
  - select the [Edit Schema...](#) item from the popup menu
- or
- open the database in [Database Editor](#) and the [Schemas](#) tab there;
  - select the schema to edit;
  - press the **Enter** key or select the [Edit Schema](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#)).

You can change the name of the schema using the [Rename Schema](#) dialog. To open the dialog you should either

- select the schema to rename in the explorer tree;
- select the [Rename Schema](#) item from the popup menu

or

- open the database in [Database Editor](#) and the [Schemas](#) tab there;
- select the schema to rename;
- select the [Rename Schema](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#)).

#### ■ How can I drop an existing schema?

To drop a schema:

- select the schema to drop in the explorer tree;
- select the [Drop Schema](#) item from popup menu

or

- open the database in [Database Editor](#) and the [Schemas](#) tab there;
- select the schema to drop;
- press the **Delete** key or select the [Drop Schema](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#))

and confirm dropping in the dialog window.

### 5.1.1 Create Schema Wizard

[Create Schema Wizard](#) guides you through the process of creating a new database schema.

The basic principles of Create Object Wizards in DB2 Maestro are covered by the corresponding [topic](#)<sup>[36]</sup>. See below to find the description of wizard steps that are unique to the current object.

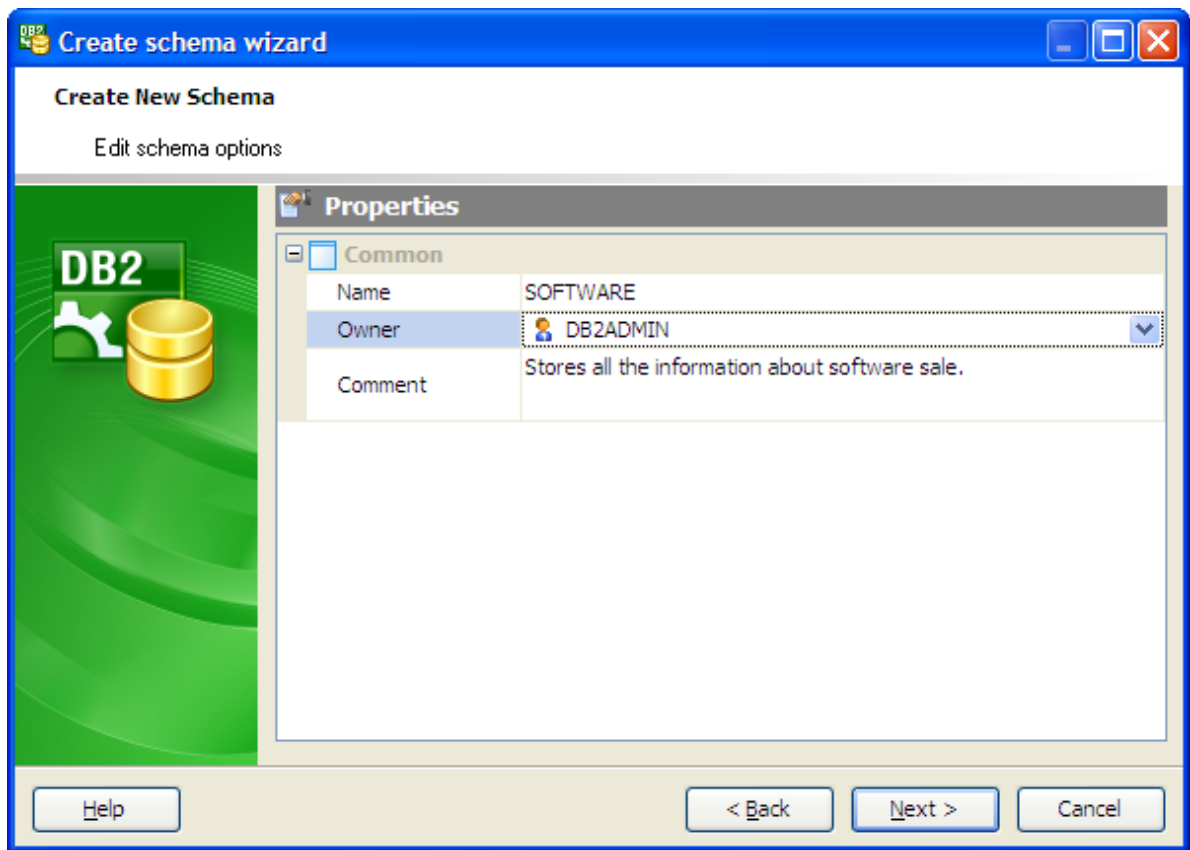
#### Schema options

##### Owner

Use the field to specify the owner of the new schema. The default owner is the user who have created the schema. By default, only the owner of an object can perform various operations with the object. In order to allow other users to operate it, privileges must be granted. (However, users that have the superuser attribute can always access any object.)

##### Comment

The box allows you to set optional text describing the new schema.

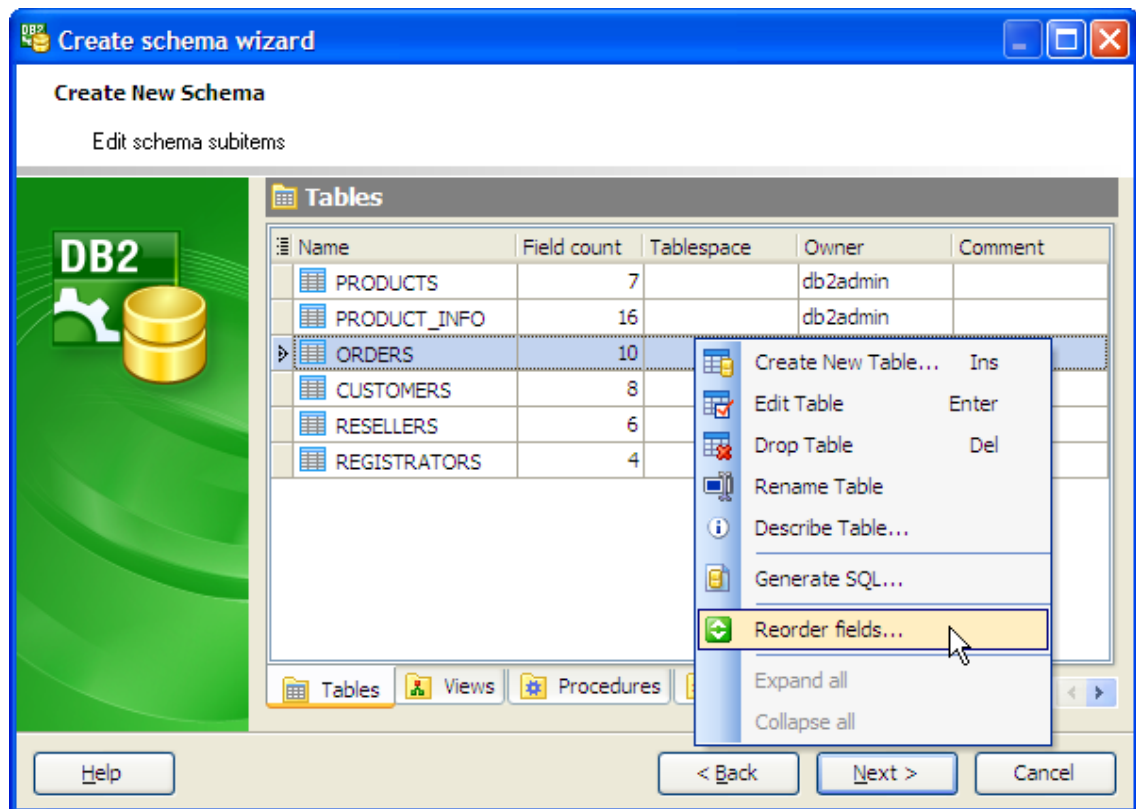


### Adding schema subitems

This wizard step allows to specify the new schema content. To add an object:

- Choose the necessary tab ([Tables](#) - to manage schema tables, [Views](#) - to manage schema views, and so on);
- Follow the [Create Object Wizard](#) link of the tab's pop-up menu;
- Complete the corresponding create object wizard. To find out how to add a new [table](#)<sup>[66]</sup>, [procedure](#)<sup>[102]</sup>, [function](#)<sup>[116]</sup>, [domain](#)<sup>[124]</sup>, [Distinct Type](#)<sup>[124]</sup>, [sequence](#)<sup>[137]</sup>, [alias](#)<sup>[129]</sup>, or [view](#)<sup>[92]</sup>, read the corresponding topics.

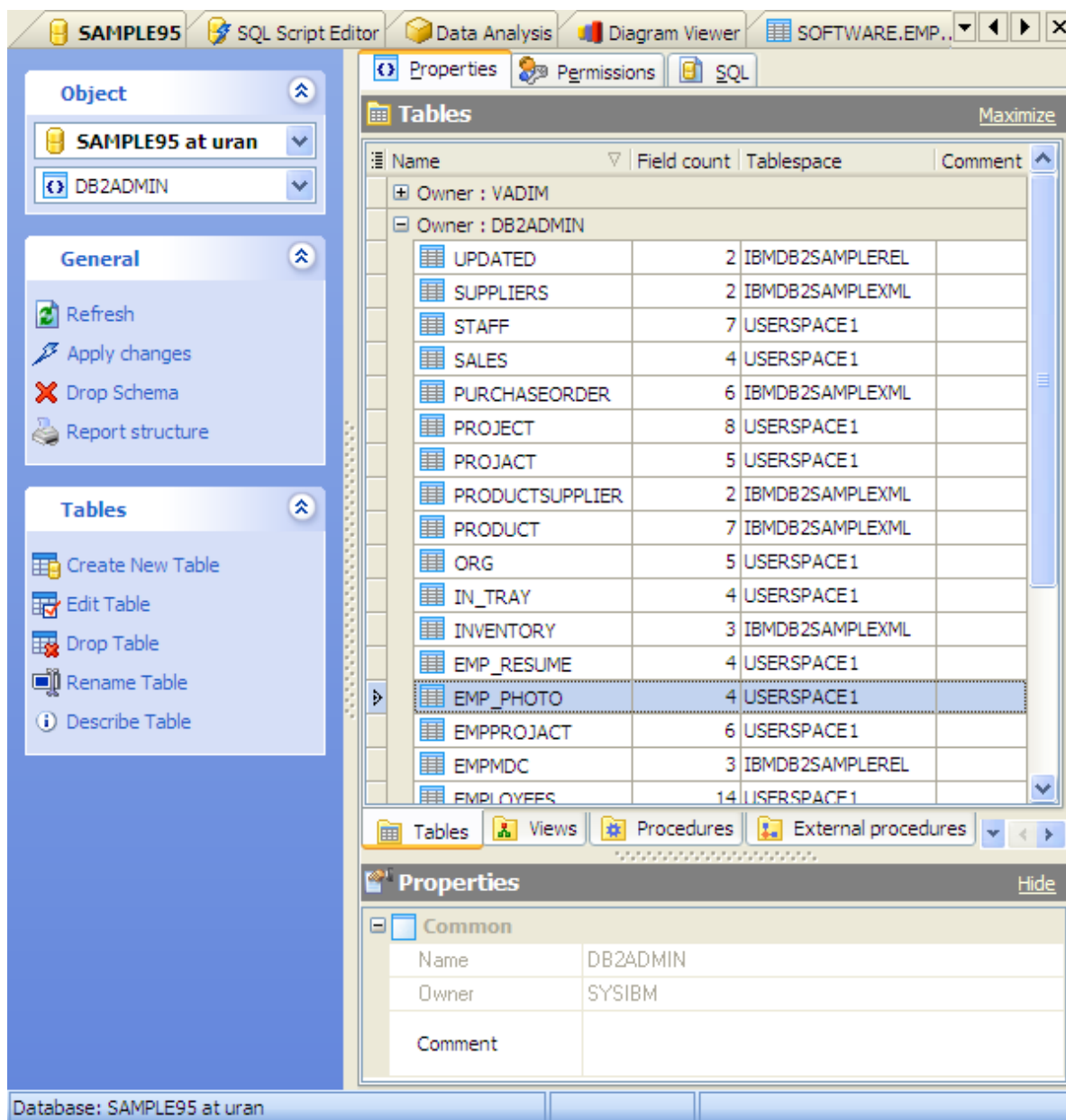
The popup menu of each tabs allow you to reorder specified objects.



### 5.1.2 Schema Editor

[Schema Editor](#) allows you to manage schema subobjects, grant privileges, view dependencies with other database objects, and also represents the SQL definition and HTML report of the schema. A schema is opened in [Schema Editor](#) from the [Explorer Tree](#) or [Object Manager](#) or after the schema creation.

The basic principles of Object Editors in DB2 Maestro are considered in the corresponding [topic](#)<sup>[39]</sup>. See below to find the description of editor tabs that are unique to the current object.



The **Properties** tab allows you to view the general schema options and to browse all objects corresponding to the schema.

At the top of the tab there are several tabs storing information about the schema subitems (e.g. *tables*, *views*, *functions*, etc.). The popup menu of each tab allows you to create new, edit, copy or drop the selected schema object. The grid allows you to operate with several objects at a time. To do it, select schema objects with the **Shift** or the **Ctrl** key pressed. After a group of objects is selected you can operate with them, e.g. *delete several objects* at once, as if it is a single object.

#### Name

Here you can view the schema name.

To apply the changes, select the [Apply Changes](#) item in the [Navigation bar](#) or use **Ctrl+F9** or **Ctrl+F7** shortcut keys.

It is also possible to modify object properties without opening the object editor: use the [Object Properties](#) item of the popup menu of the selected object from the explorer tree.

## 5.2 Tables

DB2 Maestro allows you to manipulate tables with easy: add new tables to the database, modify existing ones, browse table options and data. The sections below describe each of these actions in detail.

### ■ How can I add a new table?

New tables are created within [Create Table Wizard](#)<sup>[66]</sup>. In order to run the wizard you should either

- select the [Object | Create Database Object...](#) main menu item;
- select the [Table](#) icon in the [Create Database Object](#) dialog

or

- select the [Tables](#) list or any object from that list in the explorer tree;
- select the [Create New Table...](#) item from the popup menu.

To create a new table with the same properties as one of the existing tables has:

- select the [Object | Duplicate Database Object...](#) main menu item;
- follow the instructions of [Duplicate Object Wizard](#).

### ■ How can I work with an existing table?

Tables can be edited within [Table Editor](#)<sup>[68]</sup>. In order to run the editor you should either

- select the table for editing in the explorer tree (type the first letters of the table name for quick search);
- select the [Edit Table...](#) item from the popup menu

or

- open [Schema \(Database\) Editor](#) and the [Tables](#) tab there;
- select the table to edit;
- press the **Enter** key or select the [Edit Table](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#)).

You can also view and edit table properties without launching [Table Editor](#):

- select the table for editing in the explorer tree (type the first letters of the table name for quick search);
- select the [Table Properties...](#) item from the popup menu;
- edit table properties within the [Table Properties](#) dialog.

You can change the name of the table using the [Rename Table](#) dialog. To open the dialog you should either

- select the table to rename in the explorer tree;

- select the [Rename Table](#) item from the popup menu
- or
- open [Schema \(Database\) Editor](#) and the [Tables](#) tab there;
  - select the table to rename;
  - select the [Rename Table](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#)).

#### ■ How can I drop the existing table?

To drop a table:

- select the table to drop in the explorer tree;
  - select the [Drop Table](#) item from the popup menu
- or
- open [Schema \(Database\) Editor](#) and the [Tables](#) tab there;
  - select the table to drop;
  - press the **Delete** key or select the [Drop Table](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#))

and confirm dropping in the dialog window.

Table Editor allows you to work with table data including [master-detail\\_views](#)<sup>[70]</sup>, generate [simple SQL statements](#)<sup>[280]</sup>, [CRUD procedures](#)<sup>[281]</sup> to work with this table, and [split the table](#)<sup>[284]</sup> into two separate tables.

## 5.2.1 Create Table Wizard

[Create Table Wizard](#) guides you through the process of creating a new database table. The basic principles of Create Object Wizards in DB2 Maestro are covered by the corresponding [topic](#)<sup>[36]</sup>. See below to find the description of wizard steps that are unique to the current object.

**Table options** (All fields at this wizard step are optional, i.e. it is not obligatory for you to fill them.)

The field [Name](#) contains the name of the table being created as it was set on the previous step.

#### [Owner](#)

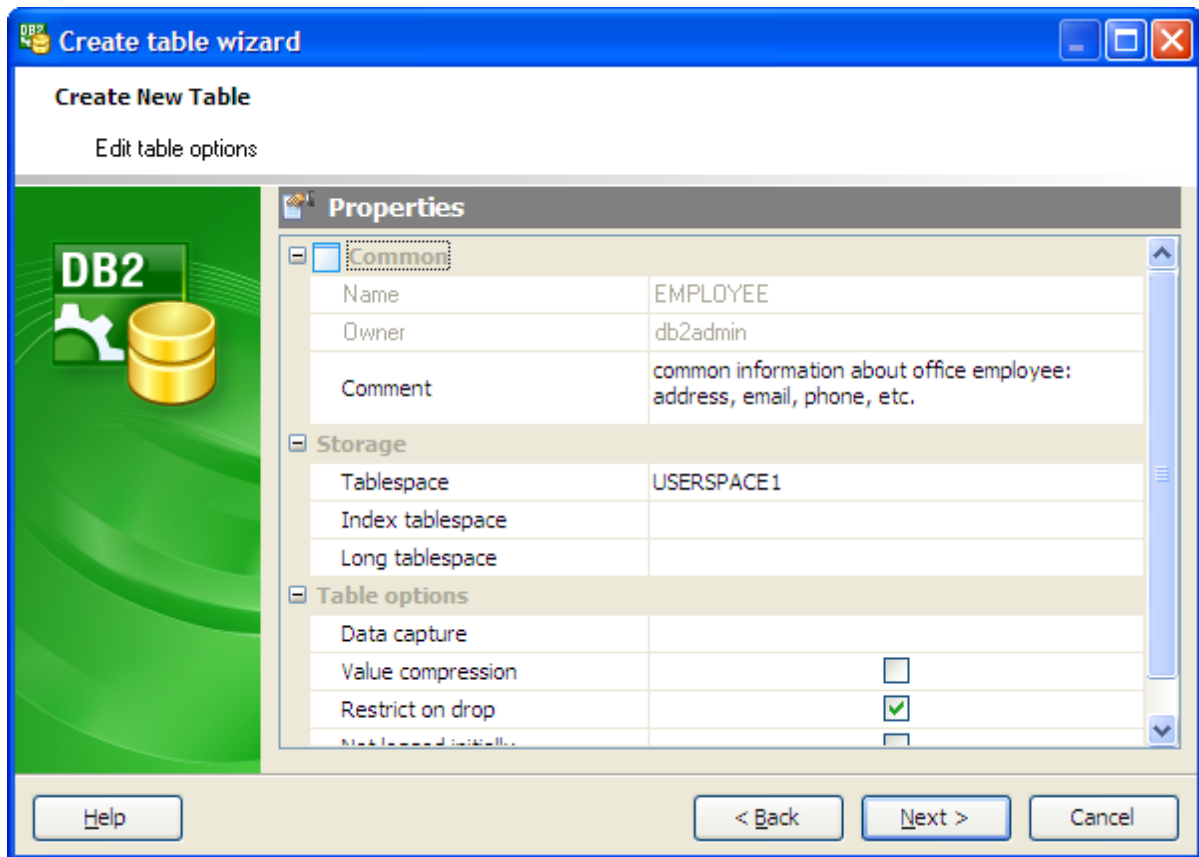
You can view here the name of the DB2 server user that will own the new table, or leave this field blank to use the default user (namely, the user executing the command). By default, only the owner of an object can perform various operations with the object. In order to allow other users to operate it, privileges must be granted. (However, users that have the superuser attribute can always access any object.)

#### [Comment](#)

Set the optional text to describe the new table.

This step also allows to set table storage options (tablespaces) and others. For more information use DB2 manuals at

<http://www-1.ibm.com/support/docview.wss?rs=71&uid=swg27009552>.

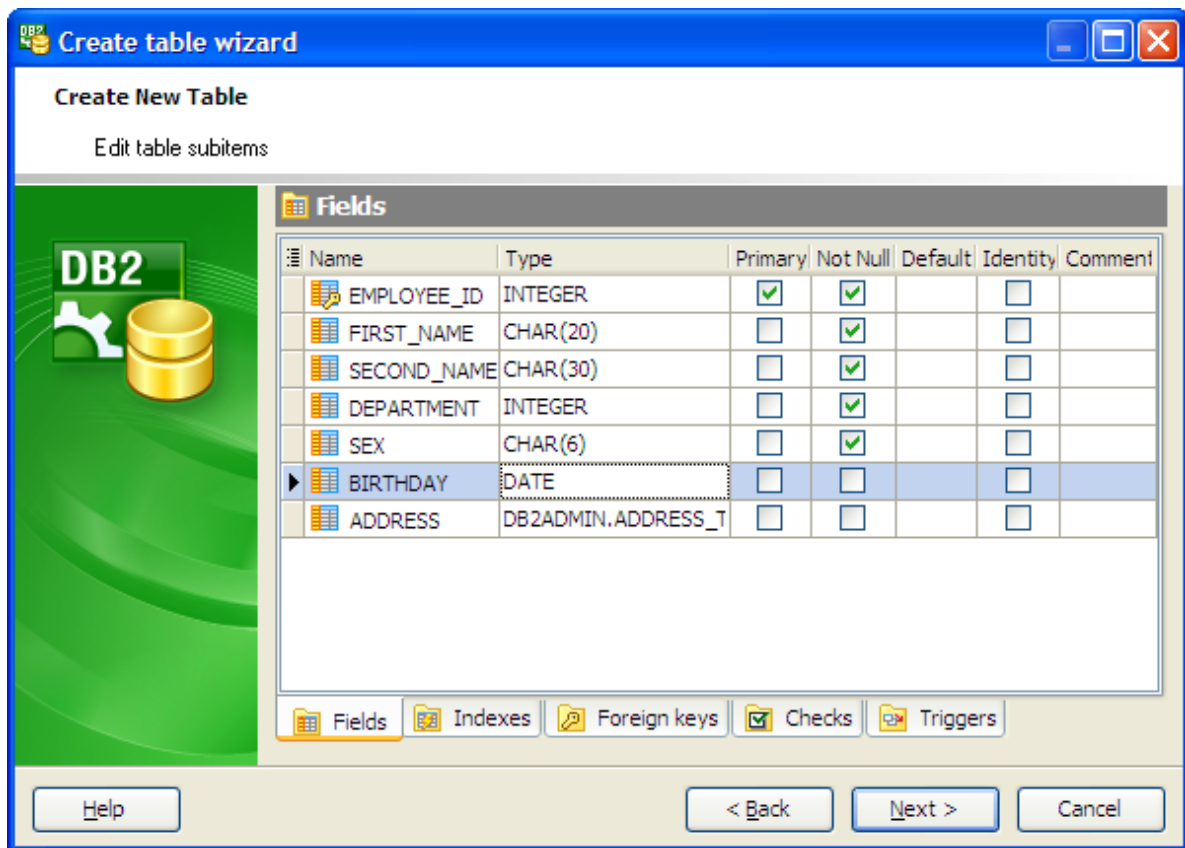


### Adding table subitems

On this step of the wizard you can fulfill the new table with fields, indexes, and foreign keys. To add a new object:

- Choose the necessary page (**Fields** - to add table fields, **Indexes** - table indexes, and so on);
- Follow the corresponding link of the tab's pop-up menu;
- Specify properties of the new object. To find the description of [field](#)<sup>[72]</sup>, [foreign key](#)<sup>[78]</sup>, [check](#)<sup>[81]</sup>, [trigger](#)<sup>[84]</sup>, and [index](#)<sup>[76]</sup>, follow the according link.

The popup menu of each tab allows to edit, drop, reorder, and rename specified objects, etc.



Click [Add All](#) or [Add](#) to include table(s) to table definition. Use the [Remove](#) or [Remove All](#) items to exclude table(s) from the list.

## 5.2.2 Table Editor

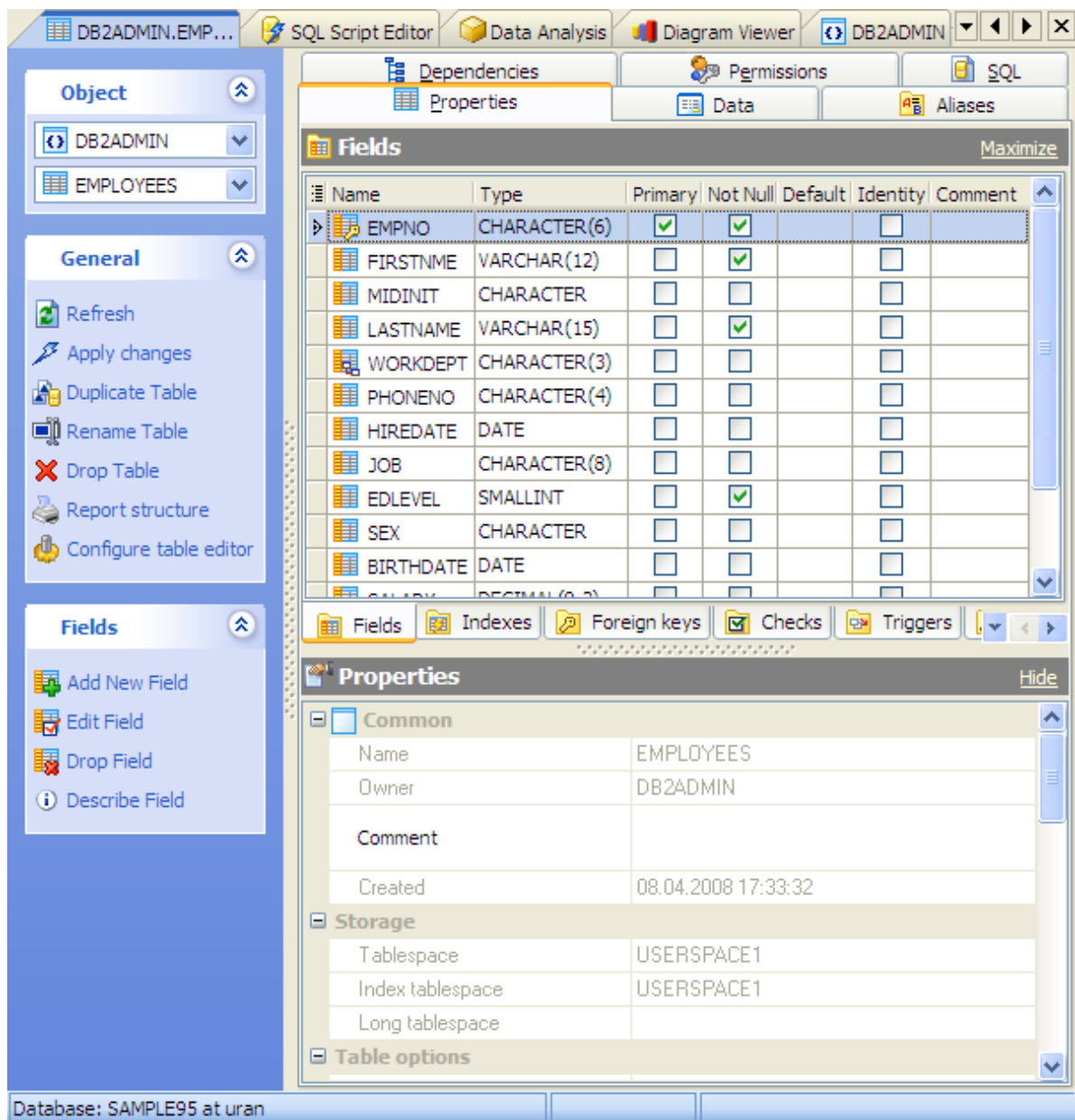
[Table Editor](#) allows you to create, edit and drop table fields, indexes, manage table data and other table subobjects. It can be opened automatically after the table is created and is available on editing the table. To open [Table Editor](#), double-click the corresponding node at the [Explorer Tree](#) or [Object Manager](#).

The basic principles of Object Editors in DB2 Maestro are considered in the corresponding [topic](#). See below to find the description of editor tabs that are unique to the current object.

- [Editing table properties](#)
- [Viewing table data](#)

### 5.2.2.1 Editing table properties

The [Properties](#) section allows you to view general table properties and also to modify .



### Subitems

Every tab is intended for work with defined **objects** (*fields, indexes, etc.*). To modify any object, double click it or use grid's popup menu. The menu also allows you to add new, rename, describe, copy/paste, and drop selected objects. To operate with several objects at a time, select them with the **Shift** or the **Ctrl** key pressed. After a group of objects is selected you can operate with it, e.g. *delete several objects at once*, as if it is a single object.

**See also:** [Fields](#)<sup>[72]</sup>, [Triggers](#)<sup>[86]</sup>, and [Indexes](#)<sup>[76]</sup>.

The tab also allows to change table storage options (tablespaces) and others. For more information use DB2 manuals at <http://www-1.ibm.com/support/docview.wss?>

[rs=71&uid=swg27009552.](#)

To apply the changes, select the [Apply Changes](#) item in the [Navigation bar](#) or use **Ctrl+F9** or **Ctrl+F7** shortcut keys.

It is also possible to modify object properties without opening the object editor: use the [Object Properties](#) item of the popup menu of the selected object from the explorer tree.

### 5.2.2.2 Managing table data

The [Data](#) tab displays the table data as a grid or as info cards (see [Data View](#)<sup>[212]</sup> for details). To edit/add a table record, use [Data Input Form](#) or type the new data directly in the grid (card). To export/import/get SQL dump of the table data, invoke corresponding modules from the grid's popup menu. To view and edit the content of BLOB columns, run [BLOB Editor](#)<sup>[223]</sup>.

#### Lookup editors

Lookup editor displays the content of parent table's columns within the drop-down window. DB2 Maestro enables a lookup editor for a column linked by a foreign key with a single column from another table. To get the corresponding data, double click the field or use **F2** shortcut and press **Alt+Down Arrow Key**.

1	7	LINDA	WILLIAMS	LINDA.WILLIAMS@sakilacustomer.org	1																																													
2	8	BARBARA	JONES	BARBARA.JONES@sakilacustomer.org	1																																													
1	9	ELIZABETH	BROWN	ELIZABETH.BROWN@sakilacustomer.org	1																																													
2	<table><thead><tr><th>address_id</th><th>address</th><th>district</th><th>city_id</th><th>postal_code</th></tr></thead><tbody><tr><td>5</td><td>1913 Hanoi Way</td><td>Nagasaki</td><td>463</td><td>35200</td></tr><tr><td>6</td><td>1121 Loja Avenue</td><td>California</td><td>449</td><td>17886</td></tr><tr><td>7</td><td>692 Joliet Street</td><td>Attika</td><td>38</td><td>83579</td></tr><tr><td>8</td><td>1566 Inegl Manor</td><td>Mandalay</td><td>349</td><td>53561</td></tr><tr><td>9</td><td>53 Idfu Parkway</td><td>Nantou</td><td>361</td><td>42399</td></tr><tr><td>10</td><td>1795 Santiago de Compostela Way</td><td>Texas</td><td>295</td><td>18743</td></tr><tr><td>11</td><td>900 Santiago de Compostela Parkway</td><td>Central Serbia</td><td>280</td><td>93896</td></tr><tr><td>12</td><td>478 Joliet Way</td><td>Hamilton</td><td>200</td><td>77948</td></tr></tbody></table>					address_id	address	district	city_id	postal_code	5	1913 Hanoi Way	Nagasaki	463	35200	6	1121 Loja Avenue	California	449	17886	7	692 Joliet Street	Attika	38	83579	8	1566 Inegl Manor	Mandalay	349	53561	9	53 Idfu Parkway	Nantou	361	42399	10	1795 Santiago de Compostela Way	Texas	295	18743	11	900 Santiago de Compostela Parkway	Central Serbia	280	93896	12	478 Joliet Way	Hamilton	200	77948
address_id	address	district	city_id	postal_code																																														
5	1913 Hanoi Way	Nagasaki	463	35200																																														
6	1121 Loja Avenue	California	449	17886																																														
7	692 Joliet Street	Attika	38	83579																																														
8	1566 Inegl Manor	Mandalay	349	53561																																														
9	53 Idfu Parkway	Nantou	361	42399																																														
10	1795 Santiago de Compostela Way	Texas	295	18743																																														
11	900 Santiago de Compostela Parkway	Central Serbia	280	93896																																														
12	478 Joliet Way	Hamilton	200	77948																																														
1																																																		
2																																																		
1	21	DONNA	THOMPSON	DONNA.THOMPSON@sakilacustomer.org	1																																													

#### Master-Detail Data View

To get data in the [master-detail](#) view mode (multiple detail pages are displayed for a single master row), use the [Show/Hide details](#) link at the editor's navigation bar. This mode allows you add/edit/delete data of detail pages. To open/close the appropriate detail page click the +/- icon or use +/- shortcuts.

6	+	710	Mountain Bike Socks, L	SO-B909-L	White	3,3963	9,5
7	+	711	Sport-100 Helmet, Blue	HL-U509-B	Blue	13,0863	34,99
8	-	712	AWC Logo Cap	CA-1098	Multi	6,9223	8,99
1 SalesOrderDetail (ProductID)							
SalesOrderID SalesOrderDetailID OrderQty ProductID UnitPrice UnitPriceDisc							
Click here to define a filter							
1		71938	113283	1	712	5,394	
2		71897	112902	4	712	5,394	
3		71858	112375	3	712	5,394	
4		71902	112962	3	712	5,394	
5		71797	111053	6	712	5,394	
6		71816	111457	4	712	5,394	
7		71784	110761	10	712	5,394	
8		71783	110748	11	712	5,2142	
9		71782	110670	10	712	5,394	
9	+	713	Long-Sleeve Logo Jersey, S	LJ-0192-S	Multi	38,4923	49,99
10	+	714	Long-Sleeve Logo Jersey, M	LJ-0192-M	Multi	38,4923	49,99
11	+	715	Long-Sleeve Logo Jersey, L	LJ-0192-L	Multi	38,4923	49,99

### Import from Clipboard

DB2 Maestro supports data import from clipboard. It is supposed that columns within the data block are separated by the tabulation symbol, records are separated by newlines and the first line of the data block contains column headers.

Example:

ColHeader1 ColHeader2

R1C1 R1C2

R2C1 R2C2

The same data format is supported by a lot of other applications, so the ability allows you to copy data from MS Excel, another table or view, or even from a data set from a different DBMS especially if it is opened with an appropriate our product.

### Uploading files as BLOBs

DB2 Maestro allows you to upload files as BLOBs into a table. For this purpose the file names must contain the information on the record they need to be placed to: the files need to be named in the same manner and include content of one or several table columns that can uniquely identify each row. To import files, specify the file name template using filed name tags (i.e. %id%, %user%, where 'id' and 'user' are the fact table columns). You can also set the default file to be uploaded to NULL fields.

Example:

Suppose we have a table 'employee' with Non-Blob data as follows:

Id User

1 Max

2 July

And we need to import the 1.jpg and 2.jpg files to a BLOB column of the table. The files are stored in the "D:\Images" directory. In this case we need to specify the "D:\Images\%Id%.jpg" file name template.

### 5.2.3 Fields

Table columns are created and edited in DB2 Maestro within the [Field Editor](#).

#### ■ How to add a new column to a table?

To add a new table column, you should either:

- open the table in [Table Editor](#) and the [Fields](#) tab there;
- press the **Insert** key or select the [Add New Field...](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#))


or

- select the table in the explorer tree and use the [Create New Field](#) popup menu item

or

- select the table [Fields](#) node or any field within the table in the explorer tree and use the [Add New Field...](#) popup menu item.

#### ■ How to edit an existing table field?

Table fields are edited within the [Field Editor](#)  dialog window. In order to open the dialog you should either

- open the table in [Table Editor](#) and the [Fields](#) tab there;
- press the **Enter** key or select the [Edit Field](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#))

or

- select the field to edit in the explorer tree and use the [Edit Field](#) popup menu item.

You can change the name of the field using the [Rename Field](#) dialog. To open the dialog you should either

- select the field to rename in the explorer tree;
- select the [Rename Field](#) item from the popup menu

or

- open the table in [Table Editor](#) and the [Fields](#) tab there;
- select the field to rename;
- select the [Rename Field](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#)).

#### ■ How to drop an existing table field?

To drop the table field:

- select the field to drop in the explorer tree;
  - select the [Drop Field](#) item from the popup menu
- or
- open the table in Table Editor and the [Fields](#) tab there;
  - press the **Delete** key or select the [Drop Field](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#))
- and confirm dropping in the dialog window.

The image shows the 'Field Editor' dialog box in DB2 Maestro. The 'Field name' is 'EMPLOYEE\_ID' and the 'Field type' is 'INTEGER'. Under 'Parameters', 'Size' and 'Precision' are both set to 0, and 'LOB Unit' is 'Bytes'. There are checkboxes for 'Bit data', 'Compact', and 'Logged', all of which are unchecked. In the 'Value generation' section, 'Primary key' and 'Not null' are checked, while 'Unique' and 'Compress system default' are unchecked. There are input fields for 'Default' and 'Formula', both of which are empty. The 'Identity' checkbox is checked, and there is a link to 'Options'. The 'Referenced' checkbox is unchecked, and there is a button to select a table. The 'Comment' field contains the text 'Primary key for EMPLOYEE table'. At the bottom are 'OK', 'Cancel', and 'Help' buttons.

Specify the [Field Name](#) and specify the [Field Data Type](#) by selecting it from the drop-down list.

**Note:** the name of the object must be unique among all the object names in the table. You can use any identifier that is allowed by DB2 server.

#### Parameters

Use the [Size](#) edit box to define the length of the field value for integer, float, char and other data types and use [Precision](#) to define the precision of the field value, e.g. for *float* data type.

### Precision

The first integer is the precision of the number; that is, the total number of digits; it may range from 1 to 31.

### BIT data

Specifies that the contents of the column are to be treated as bit (binary) data. During data exchange with other systems, code page conversions are not performed. Comparisons are done in binary, irrespective of the database collating sequence.

### Compact

Specifies that the values in the LOB column should take up minimal disk space (free any extra disk pages in the last group used by the LOB value), rather than leave any leftover space at the end of the LOB storage area that might facilitate subsequent append operations. Note that storing data in this way may cause a performance penalty in any append (length-increasing) operations on the column.

### Logged

Specifies that changes made to the column are to be written to the log. The data in such columns is then recoverable with database utilities (such as RESTORE DATABASE).

LOBs greater than 1 gigabyte cannot be logged and LOBs greater than 10 megabytes should probably not be logged.

### Field flags

#### ☒ Not Null

Forbids the NULL values for the field.

#### ☒ Unique

Includes the field into the unique key (index).

#### ☒ Primary Key

With this option checked the field becomes the only field with a primary key. If you check this field, you will not be able to set this attribute for any other field in the table. Hence if you want to create a compound primary key, do not check this field but create a primary key through the Indexes tab of [Table Editor](#) or the appropriate step of [Create Table Wizard](#).

To set the default value for the field use the [Default value](#) box. This value is always inserted to the table in this field when the field value is not specified in the insert query.

**Note:** To set a new primary key you need to define a field default value

### Formula


Indicates that DB2 will always generate a value for the column when a row is inserted into the table or whenever the result value of the generation-expression may change. The result of the expression is stored in the table. It is the recommended value unless you are using data propagation, or doing unload and reload operations.

### Referenced

This provides a shorthand method of defining a foreign key composed of a single column. Thus, if a references-clause is specified in the definition of column C, the effect is the

same as if that references-clause were specified as part of a FOREIGN KEY clause in which C is the only identified column.

The [Comment](#) box allows you to set optional text describing the field.

For field identity properties see [Setting field identity](#) .

#### 5.2.3.1 Setting field identity

Check the [Identity](#) option to indicate that the new column is an identity column.

##### [Start value](#)

Specifies the first value for the identity column. This value can be any positive or negative value that could be assigned to this column as long as there are no non-zero digits to the right of the decimal point. The default is 1.

##### [Increment by](#)

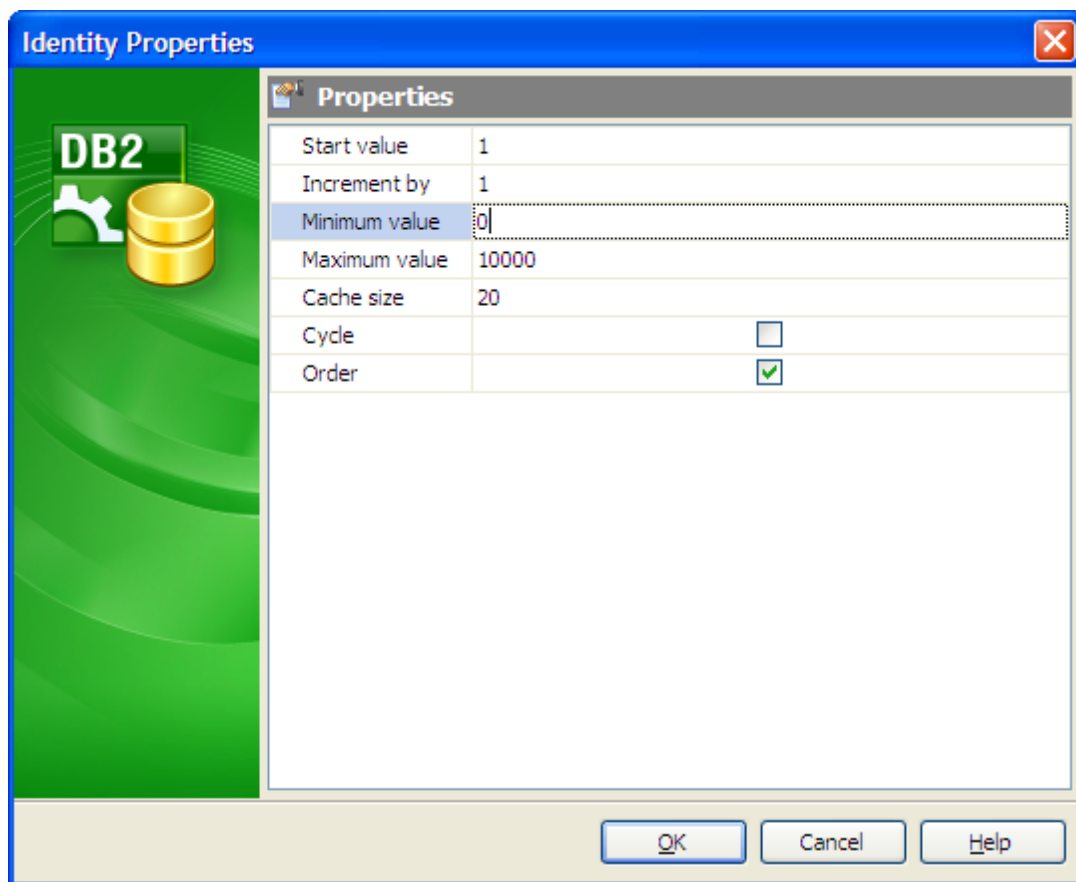
Specifies the interval between consecutive values of the identity column. This value can be any positive or negative value that could be assigned to this column. This value cannot be zero and cannot exceed the value of a large integer constant, provided that there are no non-zero digits to the right of the decimal point. If this value is negative, then the sequence of values for this identity column descends. If this value is positive, then the sequence of values for this identity column ascends. The default is 1.

##### [Cache size](#)

Specifies how many values of the identity sequence that DB2 pre-allocates and keeps in memory. Pre-allocating and storing values in the cache reduces logging when values are generated for the identity column. If a new value is needed for the identity column and there are none available in the cache, then the allocation of the value involves waiting for the log. However, when a new value is needed for the identity column and there is an unused value in the cache, the allocation of that identity value can be made quicker by not performing the logging.

You can also use the window to set [Minimum value](#), [Maximum value](#), [Cycle](#), and [Order](#).

**Note:** A table can only have a single IDENTITY column.



## 5.2.4 Indexes

[Indexes](#) are primarily used to enhance database performance (though inappropriate use may result in slower performance). The key field(s) for the index are specified as column names. Multiple fields can be specified if the index method supports multicolumn indexes.

### ■ How can I create a table index?

Table indexes are created within the [Index Properties](#) dialog window. In order to open the dialog you should either

- open the table in [Table Editor](#) and the [Indexes](#) tab there;
- press the **Insert** key or select the [Add New Index...](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#))

or

- select the table in the explorer tree and use the [Create New Index](#) popup menu item

or

- select the table [Indexes](#) node or any index within the table in the explorer tree and use the [Add New Index...](#) popup menu item.

### ■ How can I edit an existing index?

Table indexes are edited within the [Index Properties](#) dialog window. In order to open the dialog you should either

- open the table in [Table Editor](#) and the [Indexes](#) tab there;
- press the **Enter** key or select the [Edit Index](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#))

or

- select the index to edit in the explorer tree and use the [Edit Index](#) popup menu item.

You can change the name of the index using the [Rename Index](#) dialog. To open the dialog you should either

- select the index to rename in the explorer tree;
- select the [Rename Index](#) item from the popup menu

or

- open the table in [Table Editor](#) and the [Indexes](#) tab there;
- select the index to rename;
- select the [Rename Index](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#)).

### ■ How can I drop a table index?

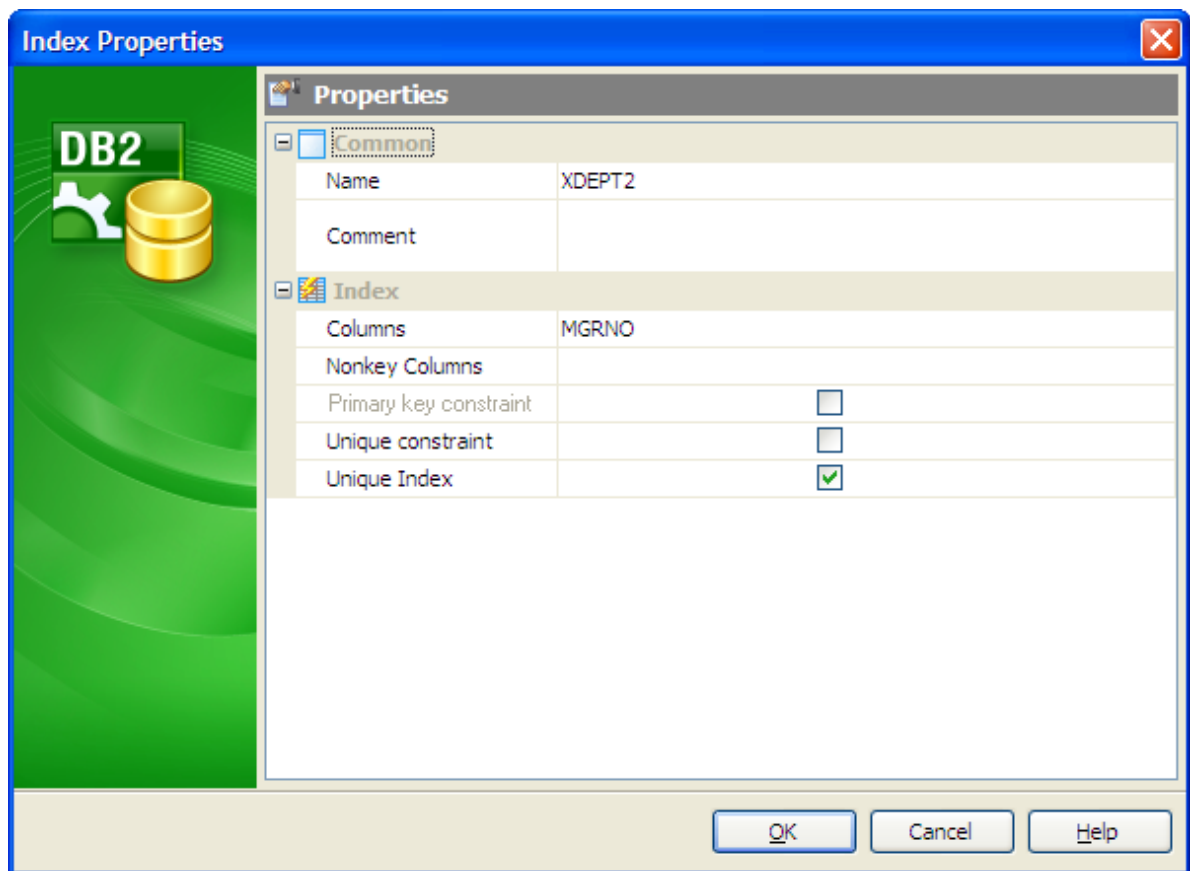
To drop the table index:

- select the index to drop in the explorer tree;
- select the [Drop Index](#) item from the popup menu

or

- open the table in [Table Editor](#) and the [Indexes](#) tab there;
- press the **Delete** key or select the [Drop Index](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#))

and confirm dropping in the dialog window.



Use the **Columns** drop-down list to select a key field(s) for the index.

☒ **Primary key constraint**

With this option checked this field becomes a compound primary key. It is useful in case the table has more than one primary key.

☒ **Unique Index**


If checked, creates a unique index for the table, i.e. the database system ensures that no two rows of the specified table have the same values in the indexed columns. In this way, if two rows both contain the NULL value for all columns of an index, the two index values are not considered to be identical. If at least one column does not contain the NULL value, two rows that have the same value in all non-NULL columns are considered to be identical.

### 5.2.5 Foreign Keys

A foreign key is a field (or collection of fields) in one table that uniquely identifies a row of another table. In other words, a foreign key is a column or a combination of columns that is used to establish and enforce a link between the data in two tables.

**Note:** To create a foreign key constraint, it is necessary to have this privilege for both the referencing and referenced tables.

■ **How can I add a new foreign key?**

Foreign keys are created within the [Foreign Key Properties](#)  dialog window. In order to open the dialog you should either

- open the table in [Table Editor](#) and the [Foreign Keys](#) tab there;
- press the **Insert** key or select the [Add New Foreign Key...](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#))


or

- select the table in the explorer tree and use the [Create New Foreign Key](#) popup menu item

or

- select the table [Foreign Keys](#) node or any foreign key within the table in the explorer tree and use the [Add New Foreign Key...](#) popup menu item.

#### **How can I edit an existing foreign key?**

Foreign Keys are edited within the [Foreign Key Properties](#)  dialog window. In order to open the dialog you should either

- open the table in [Table Editor](#) and the [Foreign Keys](#) tab there;
- press the **Enter** key or select the [Edit Foreign Key](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#))

or

- select the foreign key to edit in the explorer tree and use the [Edit Foreign Key](#) popup menu item.

You can change the name of the foreign key using the [Rename Foreign Key](#) dialog. To open the dialog you should either

- select the foreign key to rename in the explorer tree;
- select the [Rename Foreign Key](#) item from the popup menu

or

- open the table in [Table Editor](#) and the [Foreign Keys](#) tab there;
- select the foreign key to rename;
- select the [Rename Foreign Key](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#)).

#### **How can I drop a foreign key?**

To drop the foreign key:

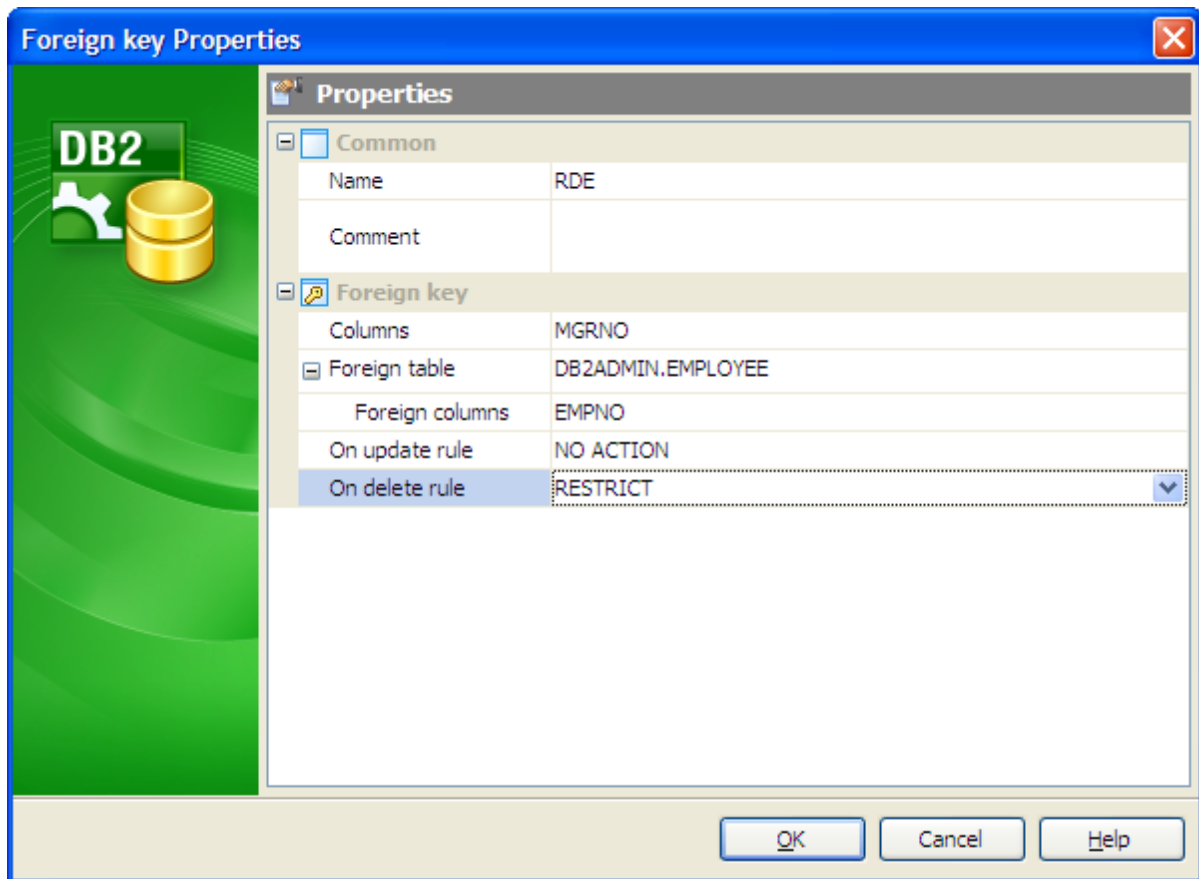
- select the foreign key to drop in the explorer tree;
- select the [Drop Foreign Key](#) item from the popup menu

or

- open the table in [Table Editor](#) and the [Foreign Keys](#) tab there;
- press the **Delete** key or select the [Drop Foreign Key](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#))

and confirm dropping in the dialog window.

Set the Foreign Key [Name](#), select [Columns](#) from the [Available Fields](#) list to include into the foreign key, select the Foreign Table [Name](#) from the drop-down list and its fields from the list to include, set other foreign key properties and apply the changes by clicking the [OK](#) button.



All the fields which are included into the Foreign Key must be included into indexes as well. See [Indexes](#) for details.

Set rule [ON DELETE](#).

- *CASCADE* Delete any rows referencing the deleted row, or update the value of the referencing column to the new value of the referenced column, respectively.
- *SET NULL* Set the referencing column(s) to null.
- *SET DEFAULT* Set the referencing column(s) to their default values.

## 5.2.6 Checks

A [check](#) constraint is the most generic constraint type. It allows you to specify that the value in a certain column must satisfy a Boolean (truth-value) expression.

[Check Properties](#) allows you to add a new check constraint or edit an existing one. It is available from [Table Editor](#)<sup>[68]</sup>, or from the corresponding nodes of the explorer tree.

### ■ How can I add a new check?

Checks are created within [Check Properties](#)<sup>[81]</sup>. In order to run the wizard you should either

- open the table in [Table Editor](#) and the [Checks](#) tab there;
- press the **Insert** key or select the [Add New Check...](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#))

or

- select the table in the explorer tree and use the [Create New Check...](#) popup menu item

or

- select the table [Checks](#) node or any check within the table in the explorer tree and use the [Add New Check...](#) popup menu item.

### ■ How can I edit an existing check?

Checks are edited within the [Check Properties](#)<sup>[81]</sup> dialog window. In order to open the dialog you should either

- open the table in [Table Editor](#) the [Checks](#) tab there;
- press the **Enter** key or select the [Edit Check](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#))

or

- select the check to edit in the explorer tree and use the [Edit Check](#) popup menu item.

You can change the name of the check using the [Rename Check](#) dialog. To open the dialog you should either

- select the check to rename in the explorer tree;
- select the [Rename Check](#) item from the popup menu

or

- open the table in [Table Editor](#) and the [Checks](#) tab there;
- select the check to rename;
- select the [Rename Check](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#)).

### ■ How can I drop a check?

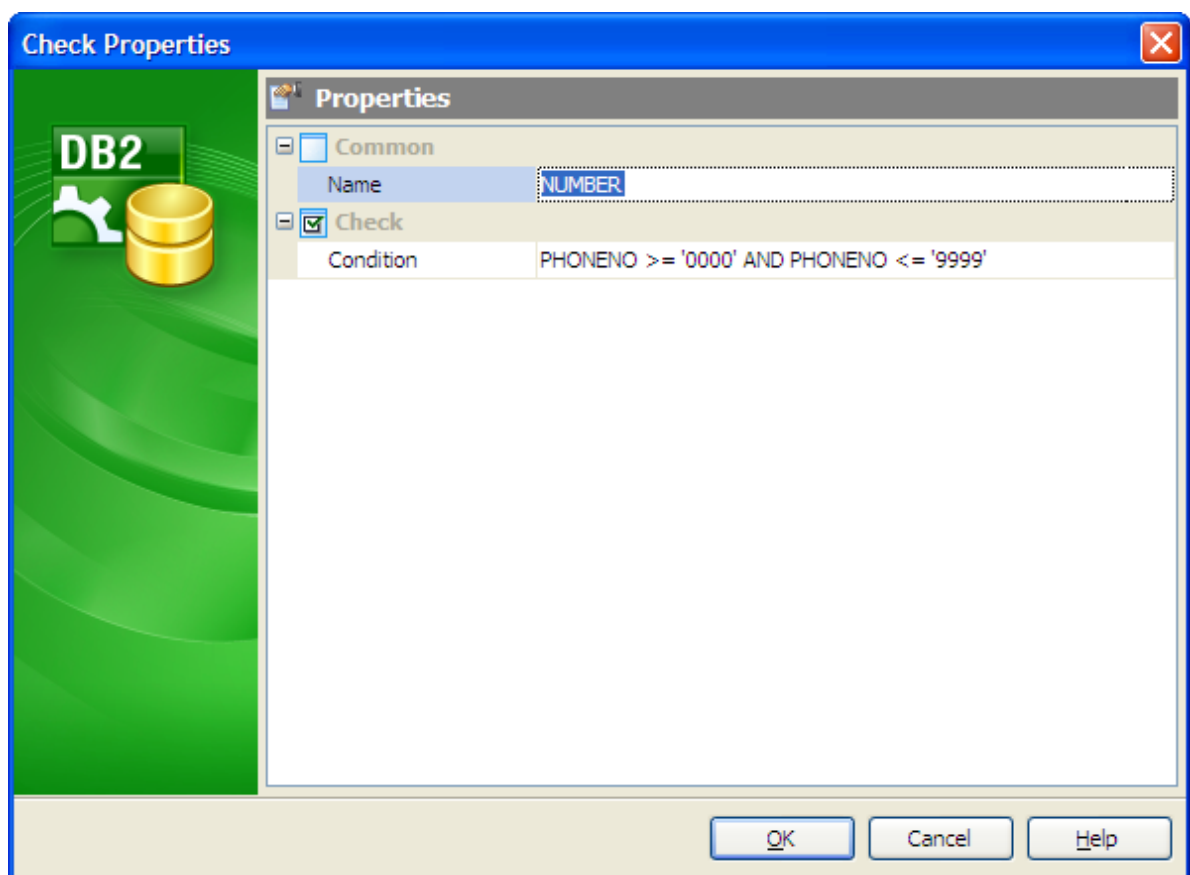
To drop the check:

- select the check to drop in the explorer tree;
- select the [Drop Check](#) item from the popup menu

or

- open the table in [Table Editor](#) and the [Checks](#) tab there;
- press the **Delete** key or select the [Drop Check](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#))

and confirm dropping in the dialog window.



#### Name

Here you can set the check name.

#### Condition

Specify an expression producing a Boolean result which new or updated rows must satisfy for an insert or update operation to succeed. Expressions evaluating to [True](#) or [Unknown](#) succeed. In case any row of an insert or update operation produce a [FALSE](#) result an error exception is raised and the insert or update does not alter the database.

## 5.2.7 Triggers

A **trigger** is a specification that the database should automatically execute a particular function whenever a certain type of operation is performed. A trigger can be defined to execute before or after an INSERT, UPDATE, or DELETE operation, either once per modified row, or once per SQL statement. If a trigger event occurs, the trigger fires.

### ■ How can I add a new trigger?

Triggers are created within [Create Trigger Wizard](#)<sup>[84]</sup>. In order to run the wizard you should either

- open the table in [Table Editor](#) and the [Triggers](#) tab there;
  - press the **Insert** key or select the [Add New Trigger...](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#))
- or
- select the table in the explorer tree and use the [Create New Trigger...](#) popup menu item
- or
- select the table [Triggers](#) node or any trigger within the table in the explorer tree and use the [Add New Trigger...](#) popup menu item.

### ■ How can I edit an existing trigger?

Triggers are edited within the [Trigger Editor](#)<sup>[86]</sup> dialog window. In order to open the dialog you should either

- open the table in [Table Editor](#) and the [Triggers](#) tab there;
  - press the **Enter** key or select the [Edit Trigger](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#))
- or
- select the trigger to edit in the explorer tree and use the [Edit Trigger](#) popup menu item.

You can change the name of the trigger using the [Rename Trigger](#) dialog. To open the dialog you should either

- select the trigger to rename in the explorer tree;
  - select the [Rename Trigger](#) item from the popup menu
- or
- open the table in [Table Editor](#) and the [Triggers](#) tab there;
  - select the trigger to rename;
  - select the [Rename Trigger](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#)).

### ■ How can I drop a trigger?

To drop the trigger:

- select the trigger to drop in the explorer tree;
- select the [Drop Trigger](#) item from the popup menu

or

- open the table in [Table Editor](#) and the [Triggers](#) tab there;
- press the **Delete** key or select the [Drop Trigger](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#))

and confirm dropping in the dialog window.

#### 5.2.7.1 Create Trigger Wizard

[Create Trigger Wizard](#) guides you through the process of creating of a new table trigger.

The basic principles of Create Object Wizards in DB2 Maestro are covered by the corresponding [topic](#)<sup>[36]</sup>. See below to find the description of wizard steps that are unique to the current object.

#### Specifying trigger properties

##### Name

The field represents the editable new trigger name as it was set on the previous step.

##### Comment

This field contains a comment to the table trigger.

##### Type (*Before, After*)

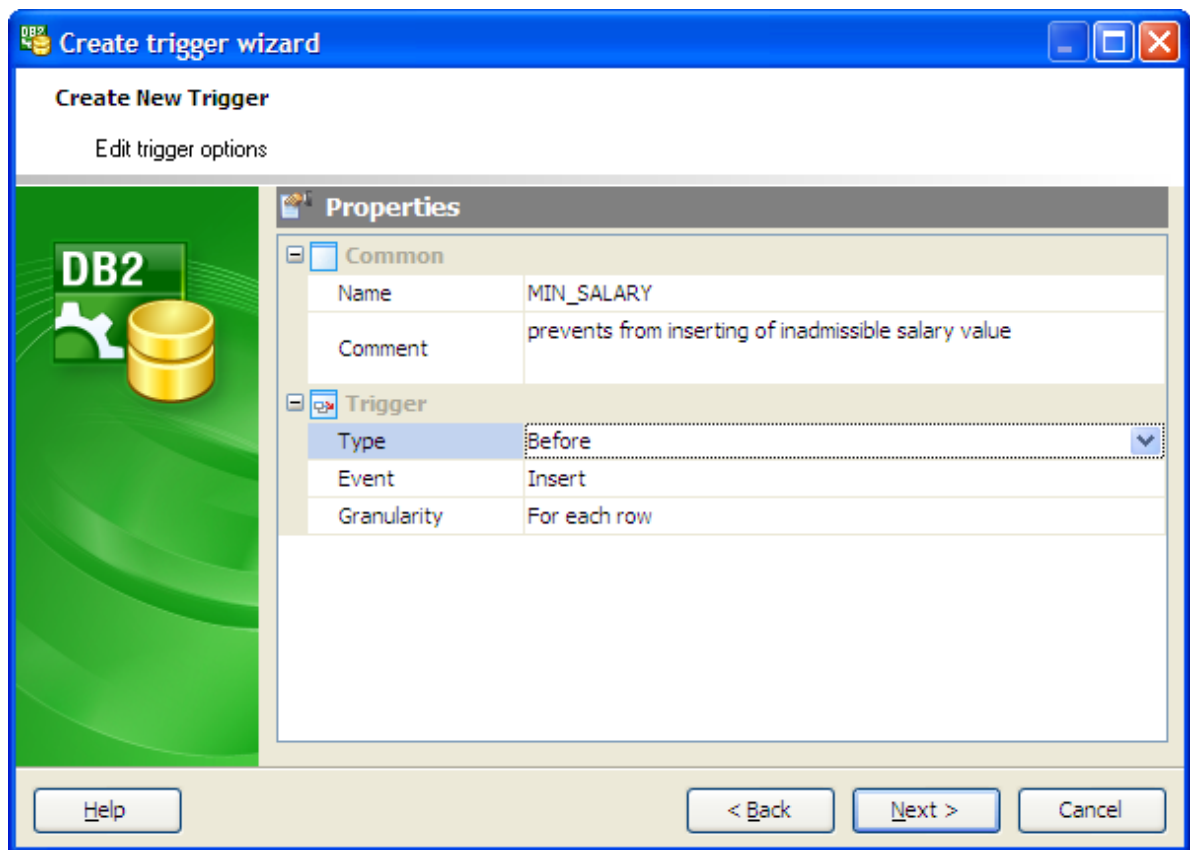
Determines whether the function is called before or after the event.

##### Event

One of *Insert*, *Update*, or *Delete*; this specifies the event that will fire the trigger.

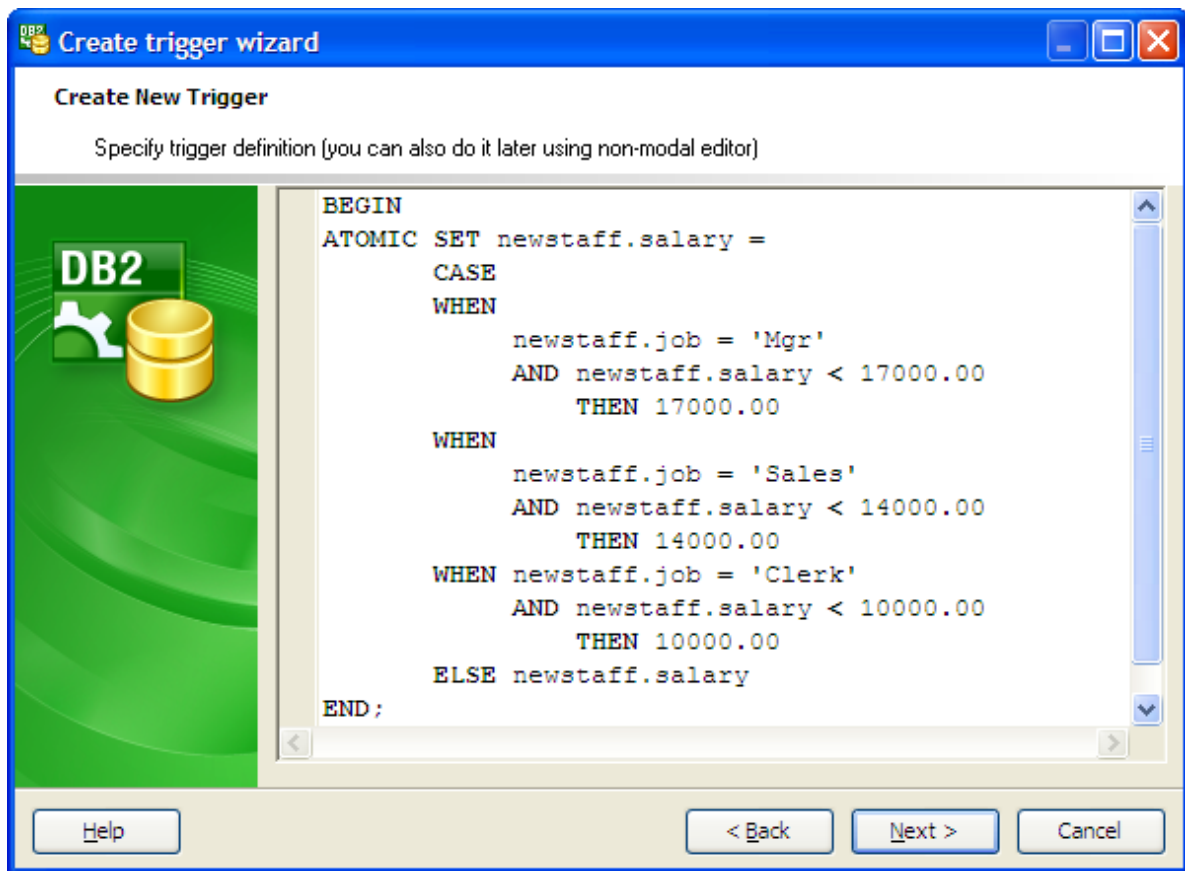
##### For Each (*Row, Statement*)

This specifies whether the trigger procedure should be fired once for every row affected by the trigger event, or just once per SQL statement.



### Specifying trigger definition

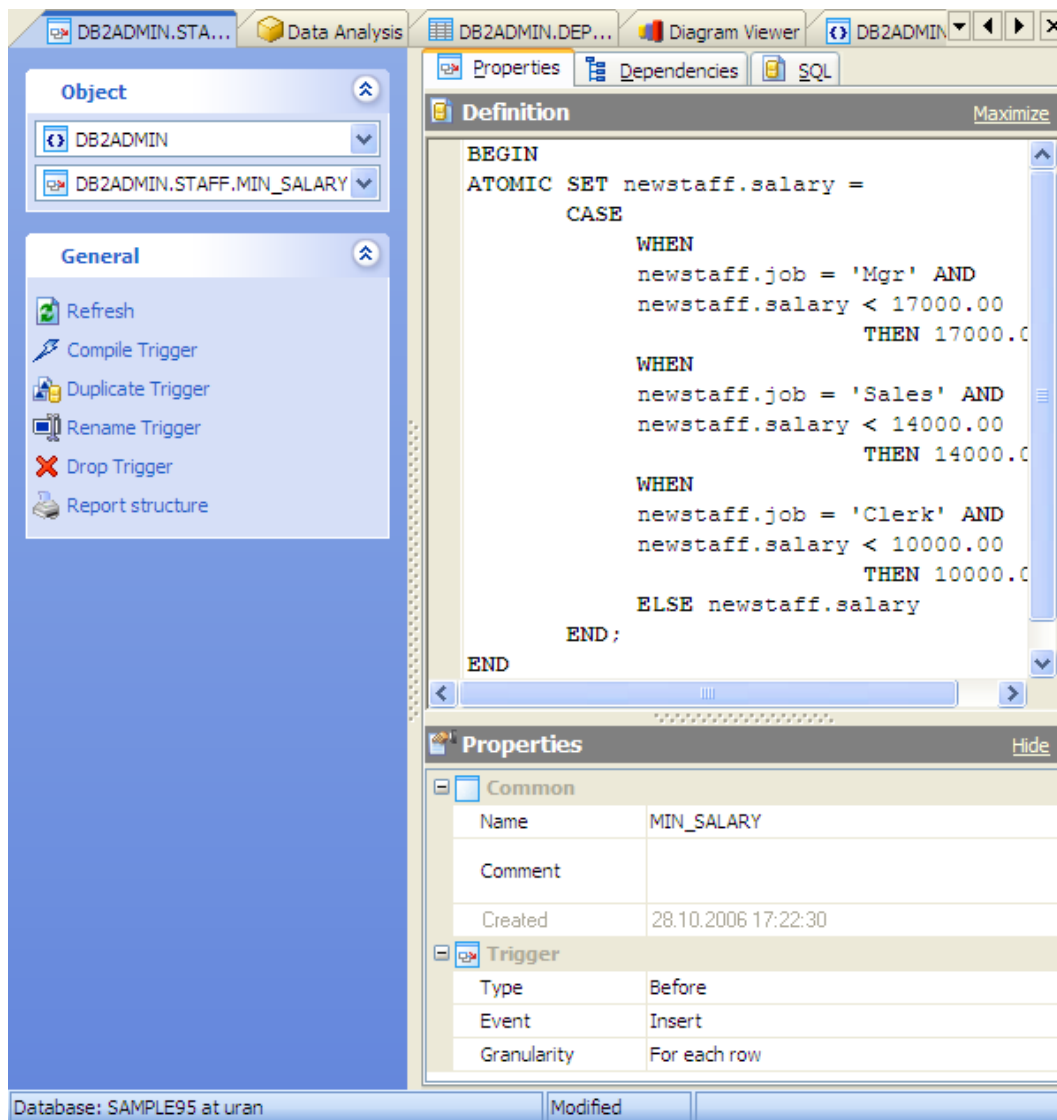
Here you can specify the trigger [definition](#). Specify the trigger steps to be executed when the trigger fires. The step is optional: you can do it later using a non-modal editor.



### 5.2.7.2 Trigger Editor

[Trigger Editor](#) can be opened automatically after the trigger is created and is available on editing the trigger.

The basic principles of Object Editors in DB2 Maestro are considered in the corresponding [topic](#)<sup>[39]</sup>. See below to find the description of editor tabs that are unique to the current object.



The main tab of the editor consists of several parts: trigger definition, and trigger properties.

### Definition

Defines the trigger conditions and actions.

### Properties

#### Name

Here you can view and change the trigger name.

**Note:** the name of the object must be unique among all the object names in its container. Moreover, all the objects that are source of data need unique names among themselves. You can use any identifier that is allowed by DB2 server.

#### Comment

This field contains a comment to the trigger.

#### [For Each \(Row, Statement\)](#)

This specifies whether the trigger procedure should be fired once for every row affected by the trigger event, or just once per SQL statement.

#### [Type\(Before, After\)](#)

Determines whether the function is called before or after the event.


#### [Event \(Insert, Update, or Delete\)](#)

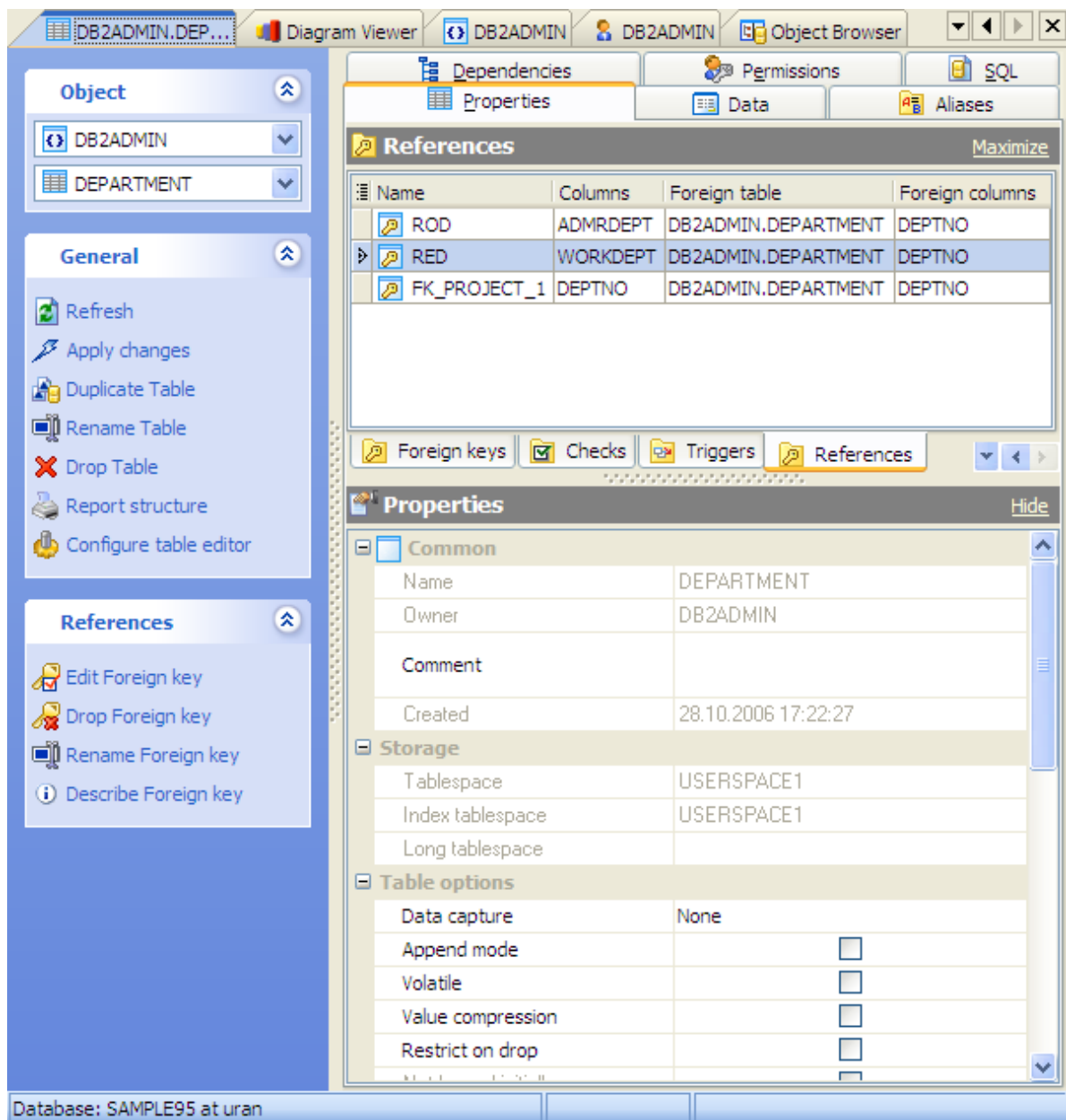
Specifies the event that will fire the trigger.

To apply the changes, select the [Apply Changes](#) item in the [Navigation bar](#) or use **Ctrl+F9** or **Ctrl+F7** shortcut keys.

It is also possible to modify object properties without opening the object editor: use the [Object Properties](#) item of the popup menu of the selected object from the explorer tree.

### 5.2.8 Foreign Key References

A foreign key specifies that the values in a column (or a group of columns) must match the values appearing in some row of another table. The [Foreign Key References](#) tab allows you to manage foreign keys created in other tables and reference for columns of the current one. Table objects are managed within the corresponding tab of [Table Editor](#) . Unlike *tables* or *views*, Foreign Key References are actually not database objects. These are only references to foreign keys. They are designed specially for easy foreign keys management.



See also: [Foreign Keys](#)

### How can I add a new foreign key reference?

Table foreign key references are edited within the [Foreign Key Editor](#) dialog window. In order to open the dialog you should either

- open the table in **Table Editor**;
- open the **Subitems** item and the **Foreign Keys References** tab there;
- press **Enter** key or select the **Edit Foreign Key** item from the popup menu

or

- select the foreign key to edit in the appropriate table group of

the explorer tree and use the [Edit Foreign Key](#) popup menu item.

#### ■ How can I edit an existing foreign key reference?

Table foreign key references are edited within the [Foreign Key Editor](#) dialog window. In order to open the dialog you should either

- open the table in [Table Editor](#);
- open the [Subitems](#) item and the [Foreign Keys References](#) tab there;
- press **Enter** key or select the [Edit Foreign Key](#) item from the popup menu

or

- select the foreign key to edit in the appropriate table group of the explorer tree and use the [Edit Foreign Key](#) popup menu item.

#### ■ How can I drop a foreign key reference?

To drop the foreign key reference:

- open the table in [Table Editor](#);
- open the [Subitems](#) item and the [Foreign Keys References](#) tab there;
- press **Delete** key or select the [Drop Foreign Key](#) item from the popup menu;

or

- select the foreign key to drop in the appropriate table group of the explorer tree and use the [Drop Foreign Key](#) popup menu item.

and confirm dropping in the dialog window.

## 5.3 Views

**Views** are useful for allowing users to access a set of relations (tables) as if it were a single table, and limiting their access to just that. Views can also be used to restrict access to rows (a subset of a particular table).

### ■ How can I create a new view?

New views are created within [Create View Wizard](#)<sup>97</sup>. In order to run the wizard you should either

- select the **Object | Create Database Object...** main menu item;
  - select the **View** icon in the **Create Database Object** dialog
- or
- select the **Views** list or any object from that list in the explorer tree;
  - select the **Create New View...** item from the popup menu
- or
- open **Schema (Database) Editor** and the **Views** tab there;
  - press the **Insert** key or select the **Create New View** item from the popup menu (alternatively, you may use the corresponding link of the **Navigation Bar**).

To create a new view with the same properties as one of the existing views has:

- select the **Object | Duplicate Database Object...** main menu item.
- follow the instructions of **Duplicate Object Wizard**.

### ■ How can I edit an existing view definition?

Views can be edited within [View Editor](#)<sup>97</sup>. In order to run the editor you should either

- select the view for editing in the explorer tree (type the first letters of the view name for quick search);
  - select the **Edit View...** item from the popup menu
- or
- open **Schema (Database) Editor** and the **Views** tab there;
  - select the view to edit;
  - press the **Enter** key or select the **Edit View** item from the popup menu (alternatively, you may use the corresponding link of the **Navigation Bar**).

You can change the name of the view using the **Rename View** dialog. To open the dialog you should either

- select the view to rename in the explorer tree;
  - select the **Rename View** item from the popup menu
- or
- open **Schema (Database) Editor** and the **Views** tab there;

- select the view to rename;
- select the [Rename View](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#)).

#### ■ How can I drop a view?

To drop a view:

- select the view to drop in the explorer tree;
  - select the [Drop View](#) item from the popup menu
- or
- open [Schema \(Database\) Editor](#) and the [Views](#) tab there;
  - select the view to drop;
  - press the **Delete** key or select the [Drop View](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#))

and confirm dropping in the dialog window.

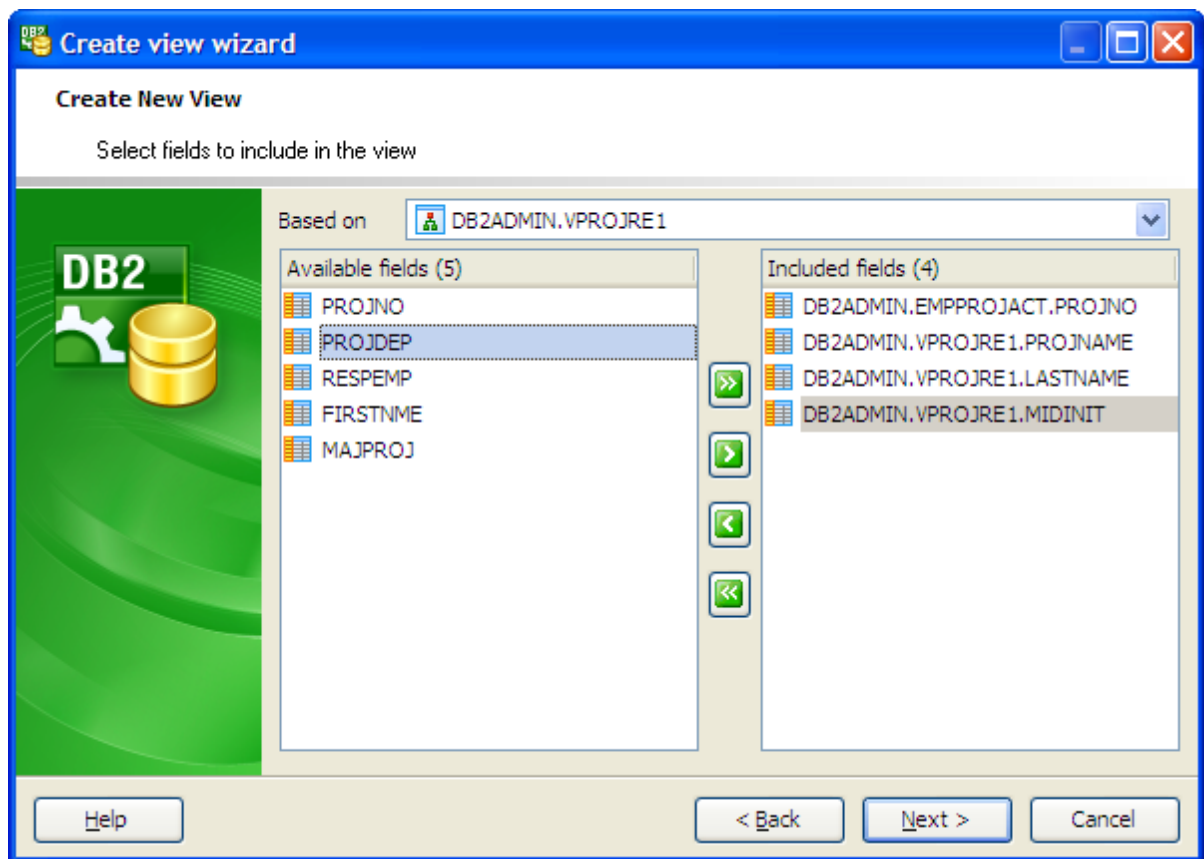
### 5.3.1 Create View Wizard

[Create View Wizard](#) guides you through the process of creating a new view. See [How To Create View](#)<sup>[91]</sup> to learn how to run this wizard.

The basic principles of Create Object Wizards in DB2 Maestro are covered by the corresponding [topic](#)<sup>[36]</sup>. See below to find the description of wizard steps that are unique to the current object.

#### Selecting fields for a new view

Select a table or a view from the [Based on](#) drop-down menu. Then specify which fields will be used in the new view. Use [Add All](#) or [Add Selected](#) buttons to include field(s) into view definition. Use the [Remove Selected](#) or [Remove All](#) items to exclude field(s) from the view's field list. Click the [Next](#) button to proceed.



### Specifying view options

#### Name

You may specify here the name of the view being created.

#### Owner

Defines the owner of the new view. By default, only the owner of an object can perform various operations with the object. In order to allow other users to operate it, privileges must be granted. (However, users that have the superuser attribute can always access any object.)

#### Comment

The box allows you to set optional text describing the view.

#### Check Option

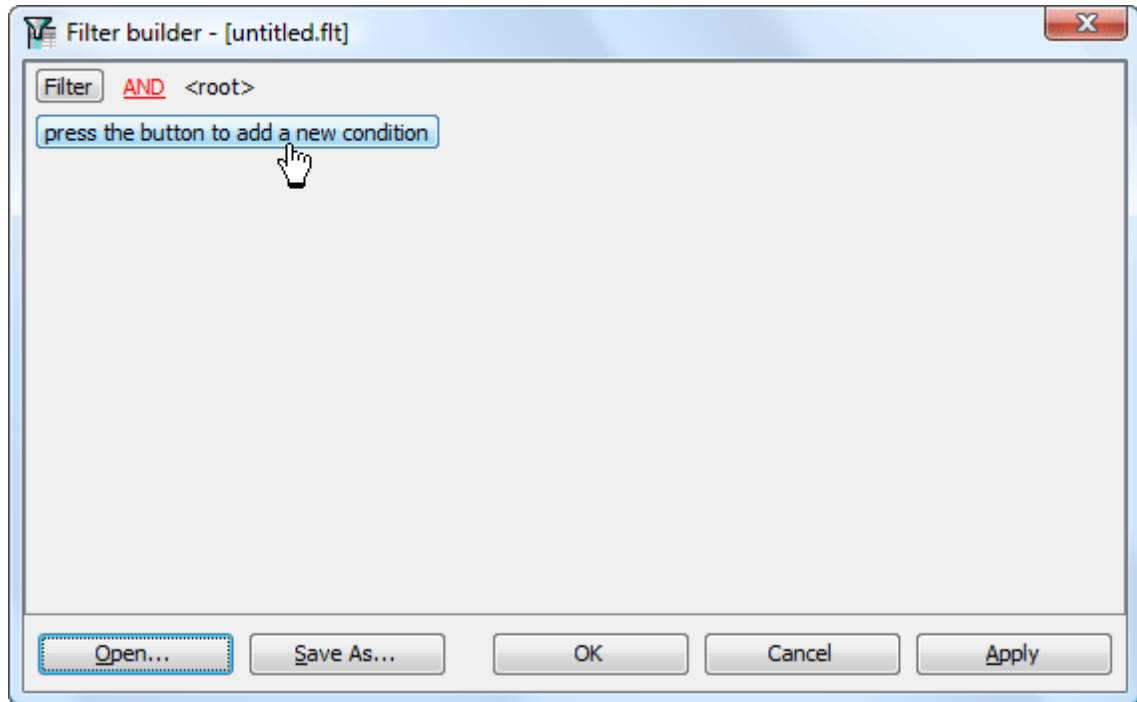
Check the option for an updatable view to prevent inserts or updates to rows except those for which the WHERE clause in the select statement is true. *Cascaded check option* and *Local check option* allow you to determine the scope of check testing when the view is defined in terms of another view. The *Local* restricts the *Check option* only to the view being defined. *Cascaded* causes the checks for underlying views to be evaluated as well.

### Specifying the WHERE condition

DB2 Maestro provides the [Filter Builder](#) dialog to facilitate a creating of the WHERE condition.

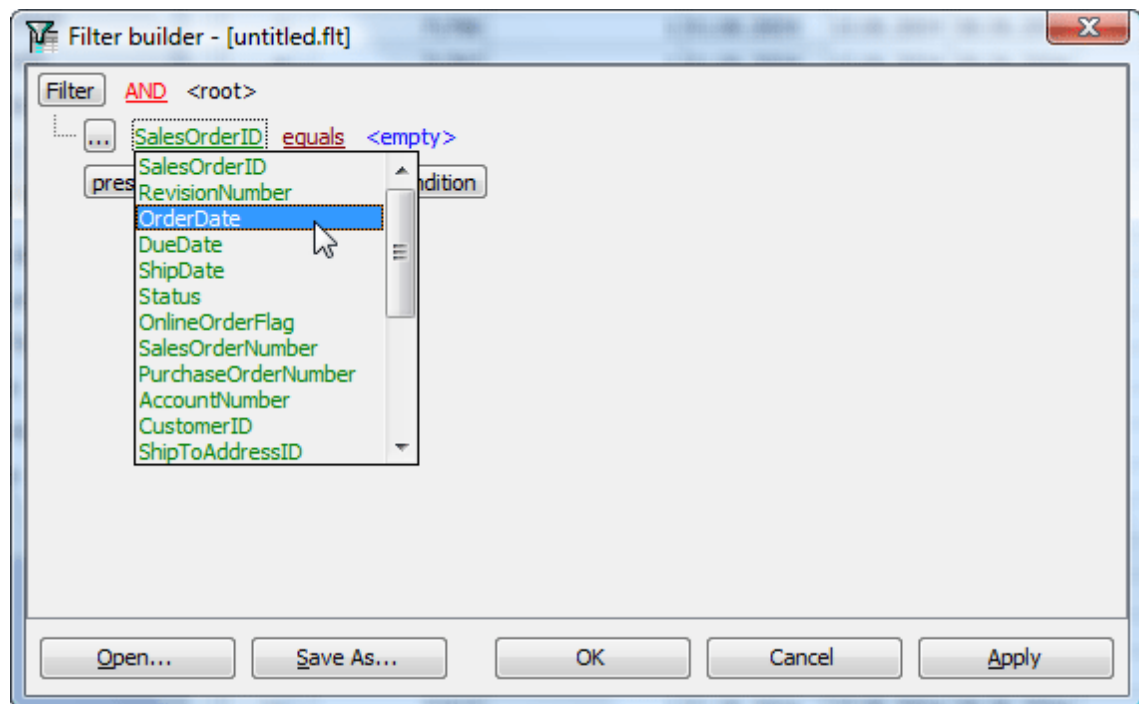
### ■ Adding a new condition to the filter

Suppose we need to select orders from the sample table *Orders* made between 01.02.2010 and 10.02.2010. These criteria are applied to the *OrderDate* column. Press the button to add this condition. Alternatively, you can use the [Filter](#) button and select the [Add Condition](#) option from the drop-down menu.



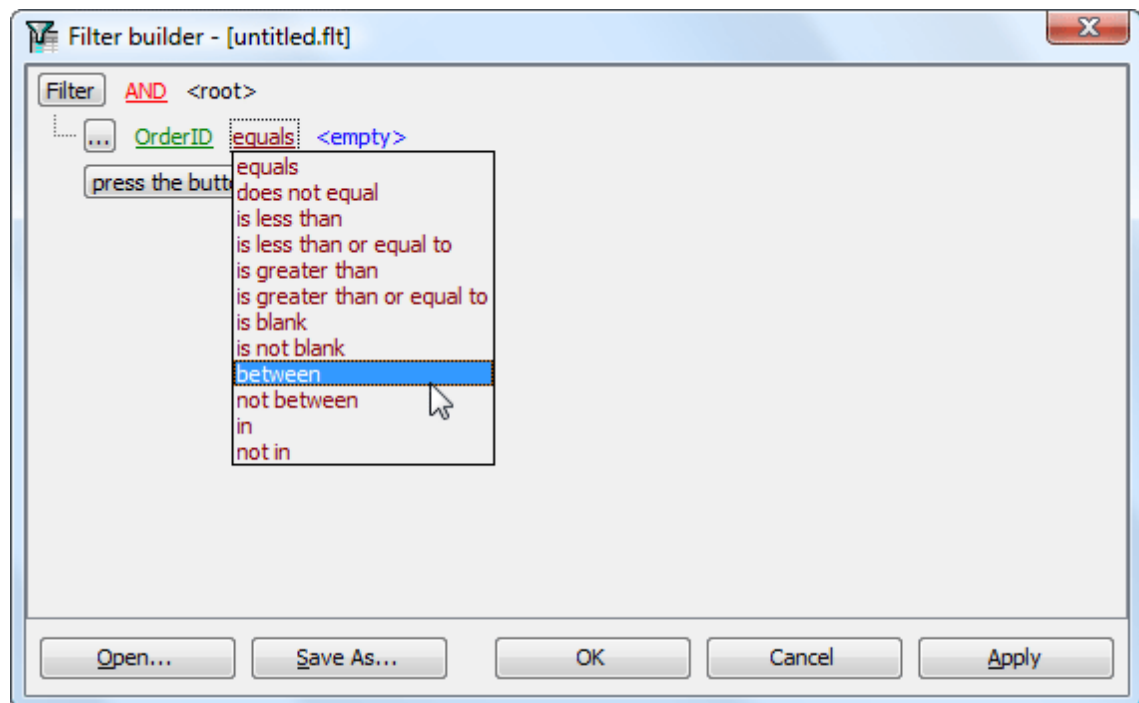
### ■ Setting a filter criteria in the condition

Select the *OrderDate* column in the drop-down list of the available columns.



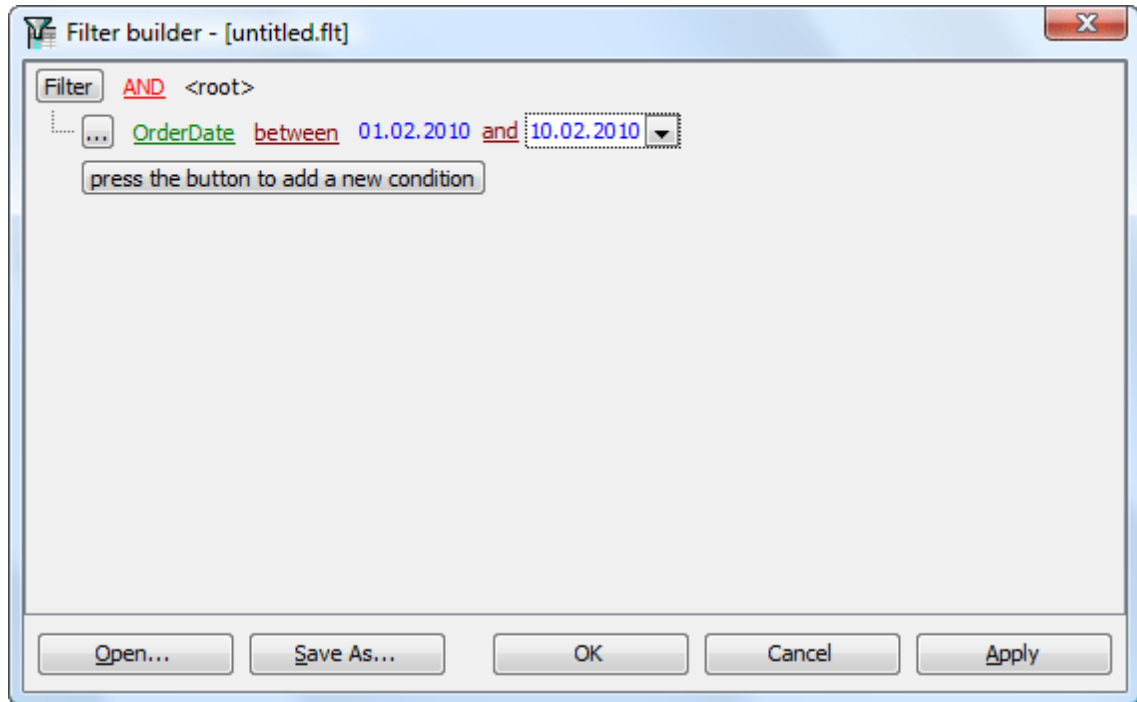
#### ■ Setting an operator in the condition

Set the proper operator. In our example it is BETWEEN.



### ■ Setting criteria values in the condition

Next, you need to specify the range values for the selected operator. The editor used in value boxes is determined by the editor type assigned to the corresponding column.

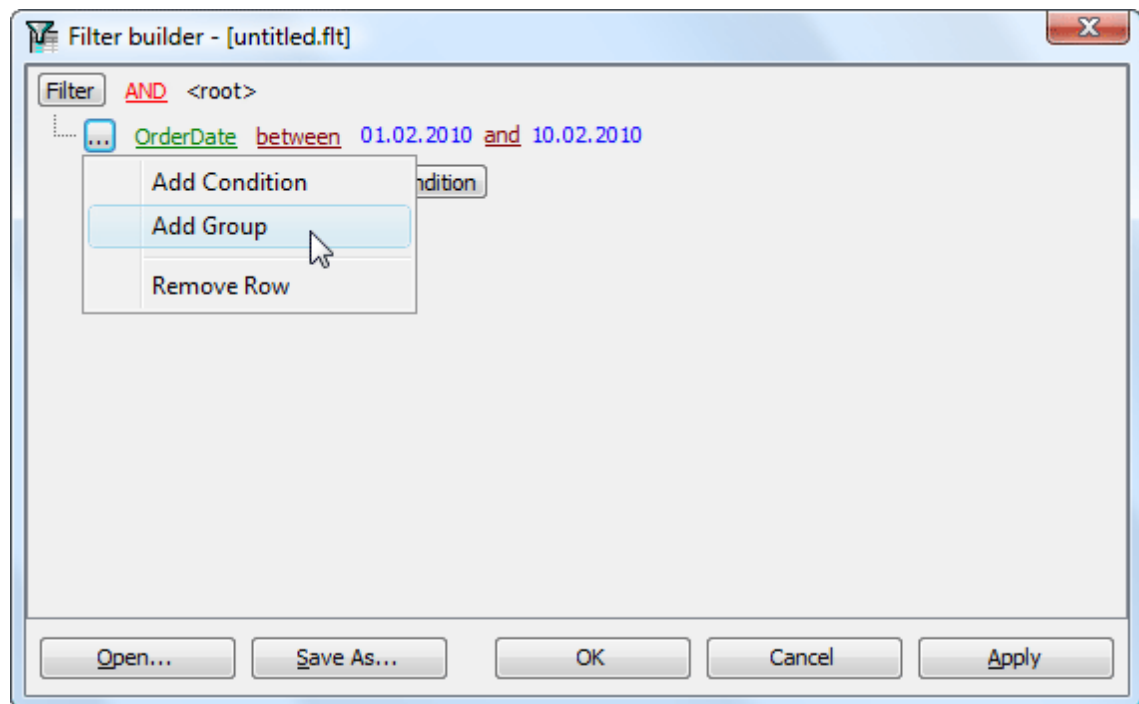


Now use the [Apply](#) button to see the filter result.

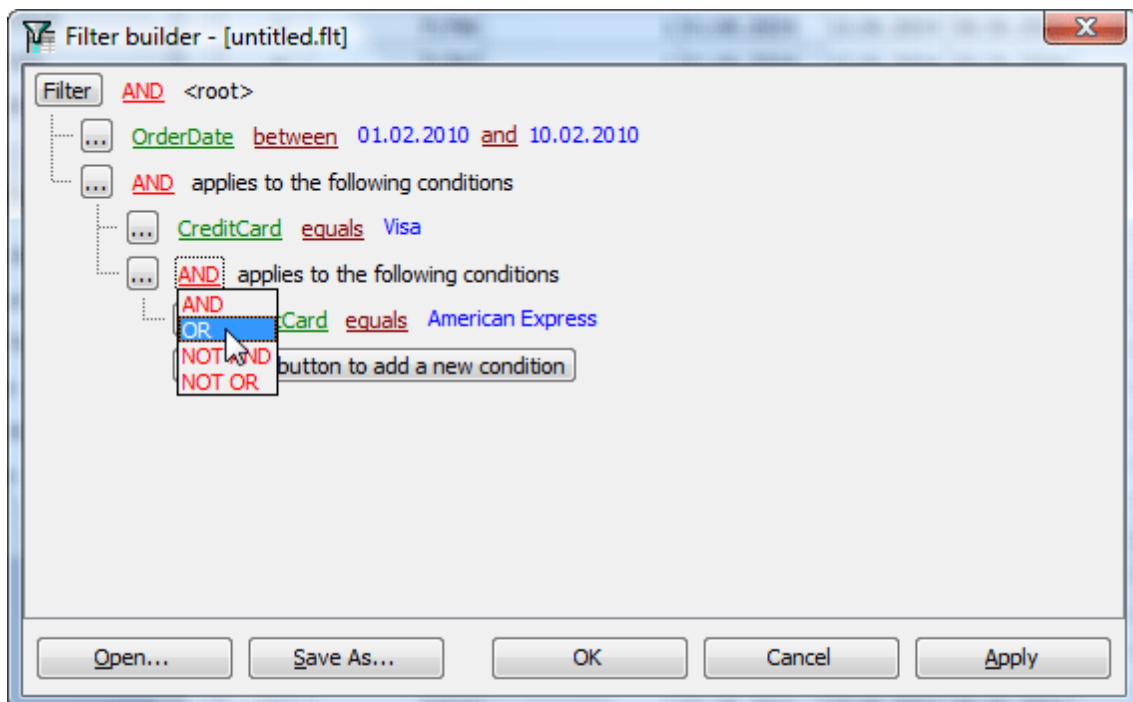
You can add additional conditions to the same root level to be combined by the AND operator.

### ■ Adding a new group

Suppose we need to select orders made between 01.02.2010 and 10.02.2010 and payed via 'Visa' or 'American Express'. This is a complex filter condition combining two simple conditions with the OR operator. Conditions from the same root level are combined by the AND operator. To add a condition combined with the previous one with the OR (NOT AND, NOT OR) operator, use a new group of conditions.



The next screen represents the finished filter conditions for this example.



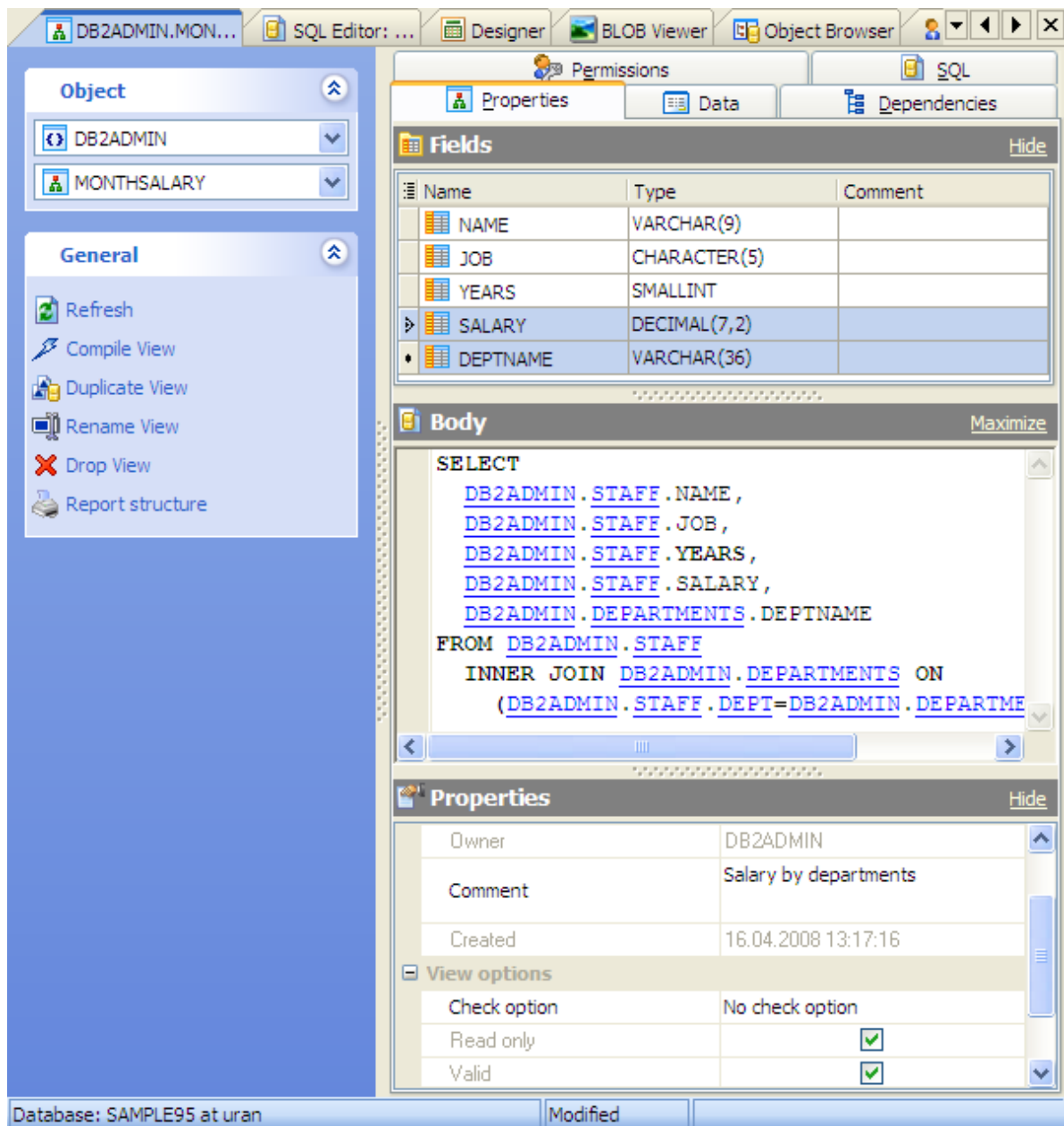
### 5.3.2 View Editor

[View Editor](#) allows you to edit the existing view definition (view name and the SELECT statement it implements).

The basic principles of Object Editors in DB2 Maestro are considered in the corresponding [topic](#)<sup>[39]</sup>. See below to find the description of editor tabs that are unique to the current object.

- [Editing view properties](#)<sup>[98]</sup>
- [Viewing data](#)<sup>[99]</sup>

**See also:** [Create View Wizard](#)<sup>[92]</sup>



### 5.3.2.1 Editing view properties

View Editor provides you with an ability to edit view properties. The **Properties** tab allows you to change the view name, view definition.

### Subitems

Every tab is intended for managing some view [subitems](#) (e.g. *fields*). Each object can be opened in its editor. Use grid's popup menu to create new, edit or drop the selected view subitems. Using the popup menu you can also copy the selected objects to clipboard or paste previously copied objects.

You can operate on several objects at a time. For this you have to select view objects with the **Shift** or the **Ctrl** key pressed. After a group of objects is selected you can operate on it, e.g. *delete several objects at once*, as if it were a single object.

**See also:** [Fields](#) 

### Body

You can edit the view definition in this box.

Use the [Name](#) field to specify the view name.

**Note:** the name of the object must be unique among all the object names in its container. Moreover, all the objects that are source of data need unique names among themselves. You can use any identifier that is allowed by DB2 server.

To apply the changes, select the [Apply Changes](#) item in the [Navigation bar](#) or use **Ctrl+F9** or **Ctrl+F7** shortcut keys.

It is also possible to modify object properties without opening the object editor: use the [Object Properties](#) item of the popup menu of the selected object from the explorer tree.

#### 5.3.2.2 Viewing data

The [Data](#) tab displays the data represented in the view as a grid (see Data View for details). The popup menu of this tab and the Data Management navigation bar allow you to export data, get SQL dump, set the value of the selected record to *Null* or to *Now* (for [Date](#) values). In tables with BLOB fields you can also call BLOB Editor to view and edit the BLOB fields.

**Object**

- DB2ADMIN
- MONTHSALARY

**General**

- Refresh
- Compile View
- Duplicate View
- Rename View
- Drop View
- Report structure

**Data Management**

- Generate query
- Data input form
- Export data
- Get SQL dump
- Import data
- Print data

**Permissions**

**Properties**

**Data**

**Dependencies**

Table

JOB: DEPTNAME

NAME	YEARS	SALARY
Click here to define a filter		
+ JOB : Clerk		
- JOB : Mgr		
+ DEPTNAME : SUPPORT SERVICES		
- DEPTNAME : SPIFFY COMPUTER SERVICE DIV.		
Molinare	7	82959,2
Lu	10	90010
Daniels	5	79260,25
Jones	12	81234
- DEPTNAME : PLANNING		
Hanes	10	80659,8
+ DEPTNAME : OPERATIONS		
- DEPTNAME : MANUFACTURING SYSTEMS		
Plotz	7	78352,8
- DEPTNAME : INFORMATION CENTER		
Sanders	7	98357,5
+ DEPTNAME : DEVELOPMENT CENTER		
+ DEPTNAME : ADMINISTRATION SYSTEMS		
- JOB : Sales		

Records fetched: 35/35

Database: SAMPLE95 at uran Modified

## 5.4 Procedures

A procedure is an application program that you can call by name. The argument or arguments of a procedure are individual scalar values, which can be of different types and can have different meanings. The arguments can be used to pass values into the procedure, receive return values from the procedure, or both.

### ■ How can I create a new procedure?

New procedures are created within [Create Procedure Wizard](#)<sup>[102]</sup>. In order to run the wizard you should either

- select the [Object | Create Database Object...](#) main menu item;
  - select the [Procedure](#) icon in the [Create Database Object](#) dialog
- or
- select the [Procedures](#) list or any object from that list in the explorer tree;
  - select the [Create New Procedure...](#) item from the popup menu
- or
- open [Schema Editor](#) and the [Procedures](#) tab there;
  - press the **Insert** key or select the [Create New Procedure](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#)).

To create a new procedure with the same properties as one of the existing procedures has:

- select the [Object | Duplicate Database Object...](#) main menu item;
- follow the instructions of [Duplicate Object Wizard](#).

### ■ How can I edit an existing procedure definition?

Procedures can be edited within [Procedure Editor](#)<sup>[106]</sup>. In order to run the editor you should either

- select the procedure for editing in the explorer tree (type the first letters of the procedure name for quick search);
  - select the [Edit Procedure...](#) item from the popup menu
- or
- open [Schema Editor](#) and the [Procedures](#) tab there;
  - select the procedure to edit;
  - press the **Enter** key or select the [Edit Procedure](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#)).

You can change the name of the procedure using the [Rename Procedure](#) dialog. To open the dialog you should either

- select the procedure to rename in the explorer tree;
  - select the [Rename Procedure](#) item from the popup menu
- or
- open [Schema Editor](#) and the [Procedures](#) tab there;

- select the procedure to rename;
- select the [Rename Procedure](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#)).

#### ■ How can I execute a procedure?

To execute a procedure:

- select the procedure in the explorer tree (type the first letters of the procedure name for quick search);
- select the [Edit Procedure...](#) item from the popup menu;
- execute the procedure using the [Execute](#) link of the [Navigation Bar](#)

or

open [Schema Editor](#) and the [Procedures](#) tab there;

- select the procedure to execute;
- press the **Enter** key or select the [Edit Procedure](#) item from the popup menu, or use the corresponding link of the [Navigation Bar](#);
- execute the procedure using the [Execute](#) link of the [Navigation bar](#).

#### ■ How can I drop a procedure?

To drop a procedure:

- select the procedure to drop in the explorer tree;
- select the [Drop Procedure](#) item from the popup menu

or

- open [Schema Editor](#) and the [Procedures](#) tab there;
- select the procedure to drop;
- press the **Delete** key or select the [Drop Procedure](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#))

and confirm dropping in the dialog window.

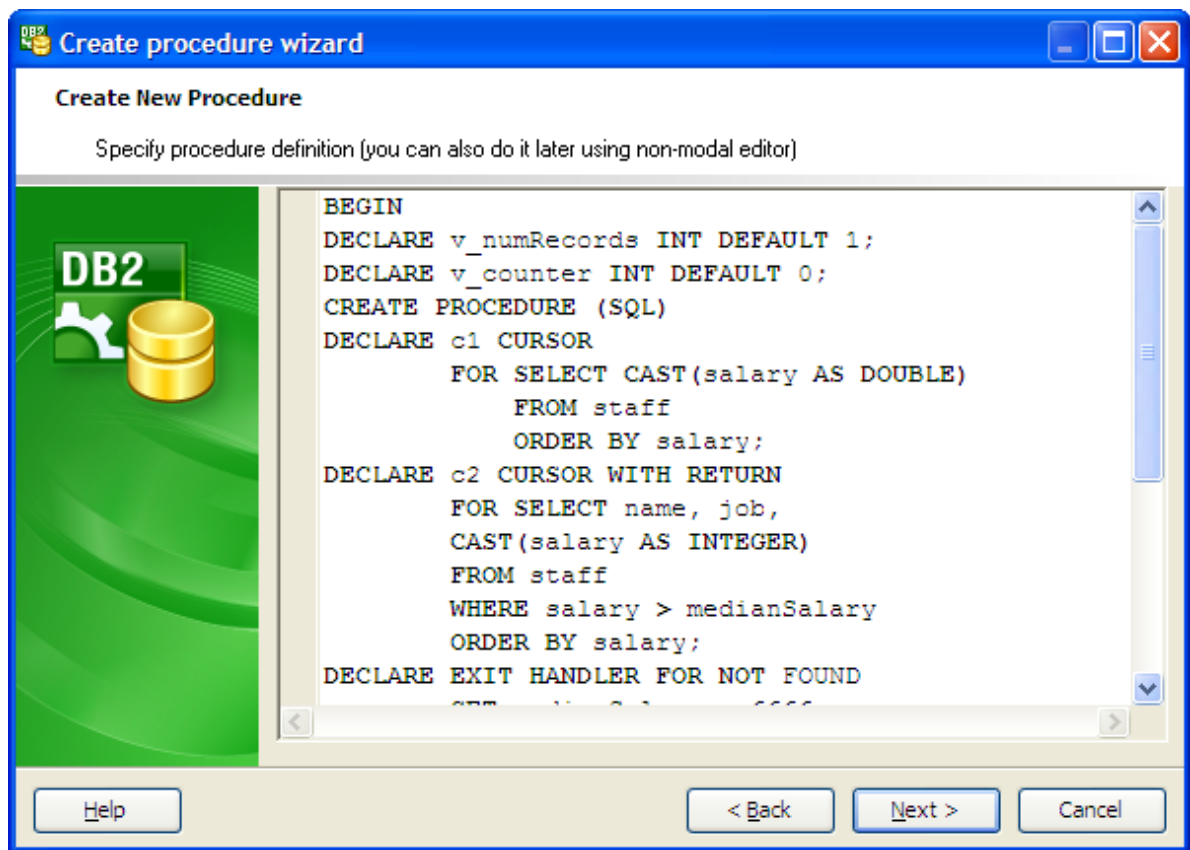
## 5.4.1 Create Procedure Wizard

[Create Procedure Wizard](#) guides you through the process of creating a new procedure. See [How To Create Procedure](#)<sup>[101]</sup> for instructions on running this wizard.

The basic principles of Create Object Wizards in DB2 Maestro are covered by the corresponding [topic](#)<sup>[36]</sup>. See below to find the description of wizard steps that are unique to the current object.

- [Specifying procedure options](#)<sup>[103]</sup>
- [Managing parameters of a new procedure](#)<sup>[105]</sup>
- [Specifying procedure definition](#)<sup>[105]</sup>

**See also:** [Procedure Editor](#)<sup>[106]</sup>



#### 5.4.1.1 Specifying procedure options

Specify procedure options according to your needs. The detailed description is given below.

**Create procedure wizard**

**Create New Procedure**

Edit procedure options

**Properties**

- ☐ **Common**

Name	DEPT_MEDIAN_SALARY
Owner	db2admin
Comment	SQL procedure that returns the median staff salary. Return a result set containing the name, position, and salary of all
Specific name	
- ☒ **Procedure**

Data access	Contains SQL
Dynamic result sets	1
Deterministic	<input checked="" type="checkbox"/>

Help    < Back    Next >    Cancel

#### Name

The new procedure name as it was set on the previous step.

#### Owner

Defines the owner for the procedure. By default, only the owner of an object can perform various operations with the object. In order to allow other users to operate it, privileges must be granted. (However, users that have the superuser attribute can always access any object.)

#### Comment

Specify a comment for the procedure.

#### ☒ Deterministic

Check the option if the procedure always produces the same result for the same input parameters, and uncheck otherwise.

#### Specific name

Provides a unique name for the instance of the procedure that is being defined. This specific name can be used when dropping the procedure or commenting on the procedure. It can never be used to invoke the procedure.

#### Data access (Contains SQL, Reads SQL data, Modifies SQL data)

Indicates the level of data access for SQL statements included in the procedure.

*Contains SQL* indicates that SQL statements that neither read nor modify SQL data can be

executed by the procedure. Statements that are not supported in procedures might return a different error.

*Reads SQL data* Indicates that some SQL statements that do not modify SQL data can be included in the procedure.

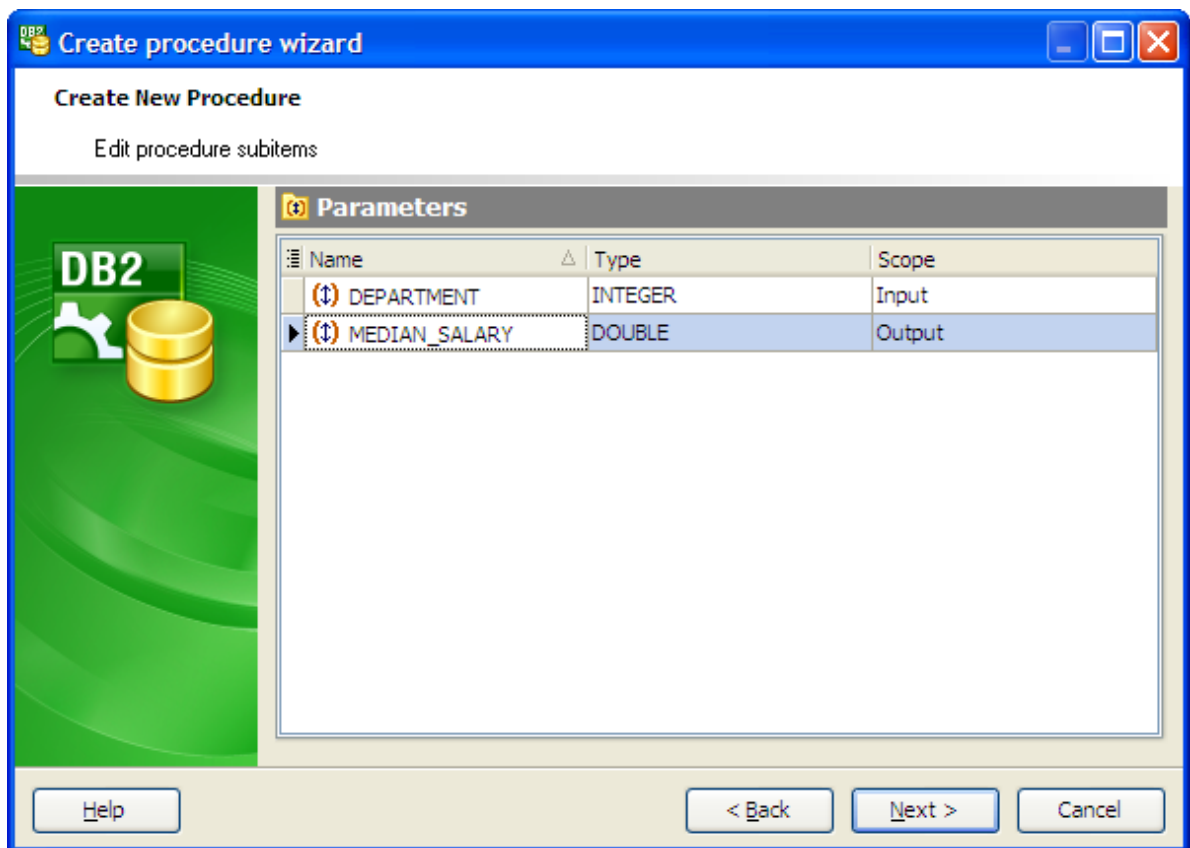
*Modifies SQL data* indicates that the procedure can execute any SQL statement except statements that are not supported in procedures.

#### Dynamic result sets

Indicates the estimated upper bound of returned result sets for the procedure.

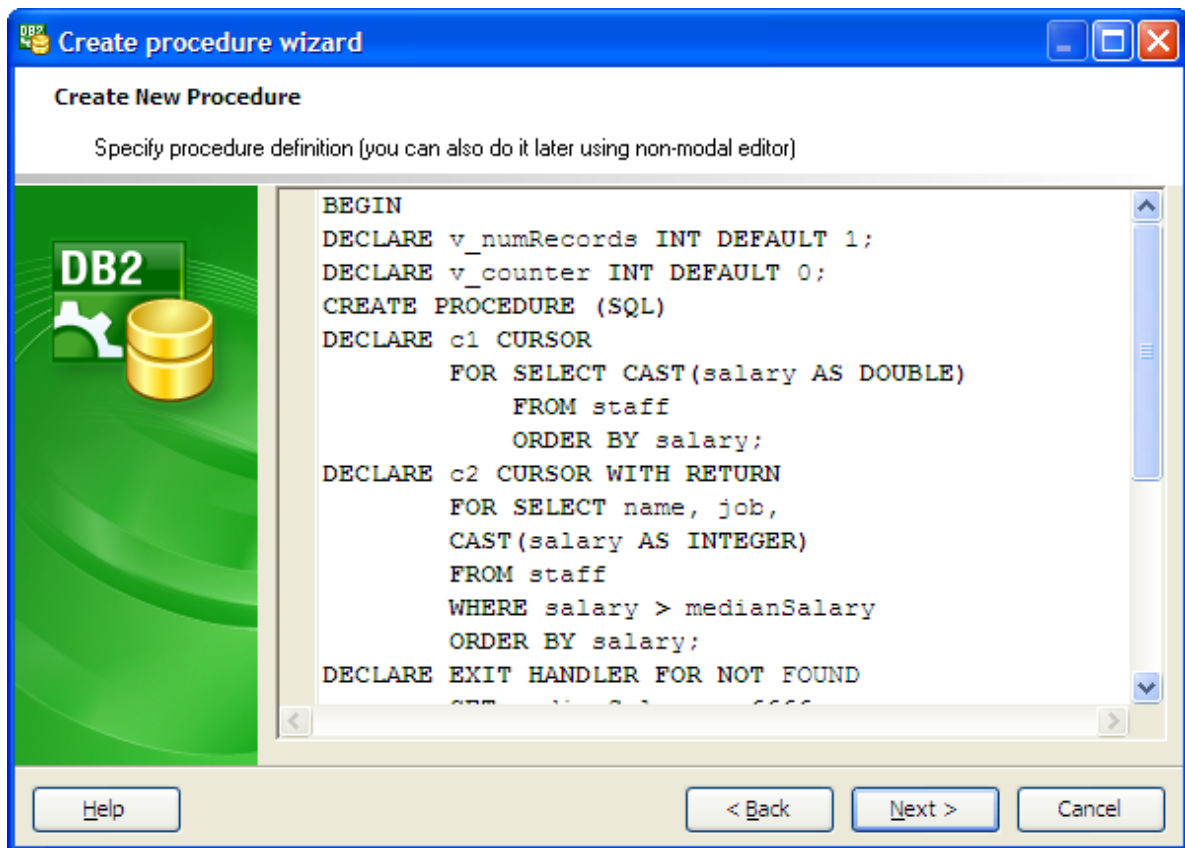
#### 5.4.1.2 Managing parameters

Use the pop-up menu or press **Insert** to add a new parameter and set its properties in [Parameter Editor](#). Press **Enter** or use the appropriate pop-up menu item to edit the selected parameter, or the **Delete** to delete one.



#### 5.4.1.3 Specifying procedure definition

At this step you can specify the SQL definition for the new procedure. The step is optional: you can do it later using a non-modal editor.



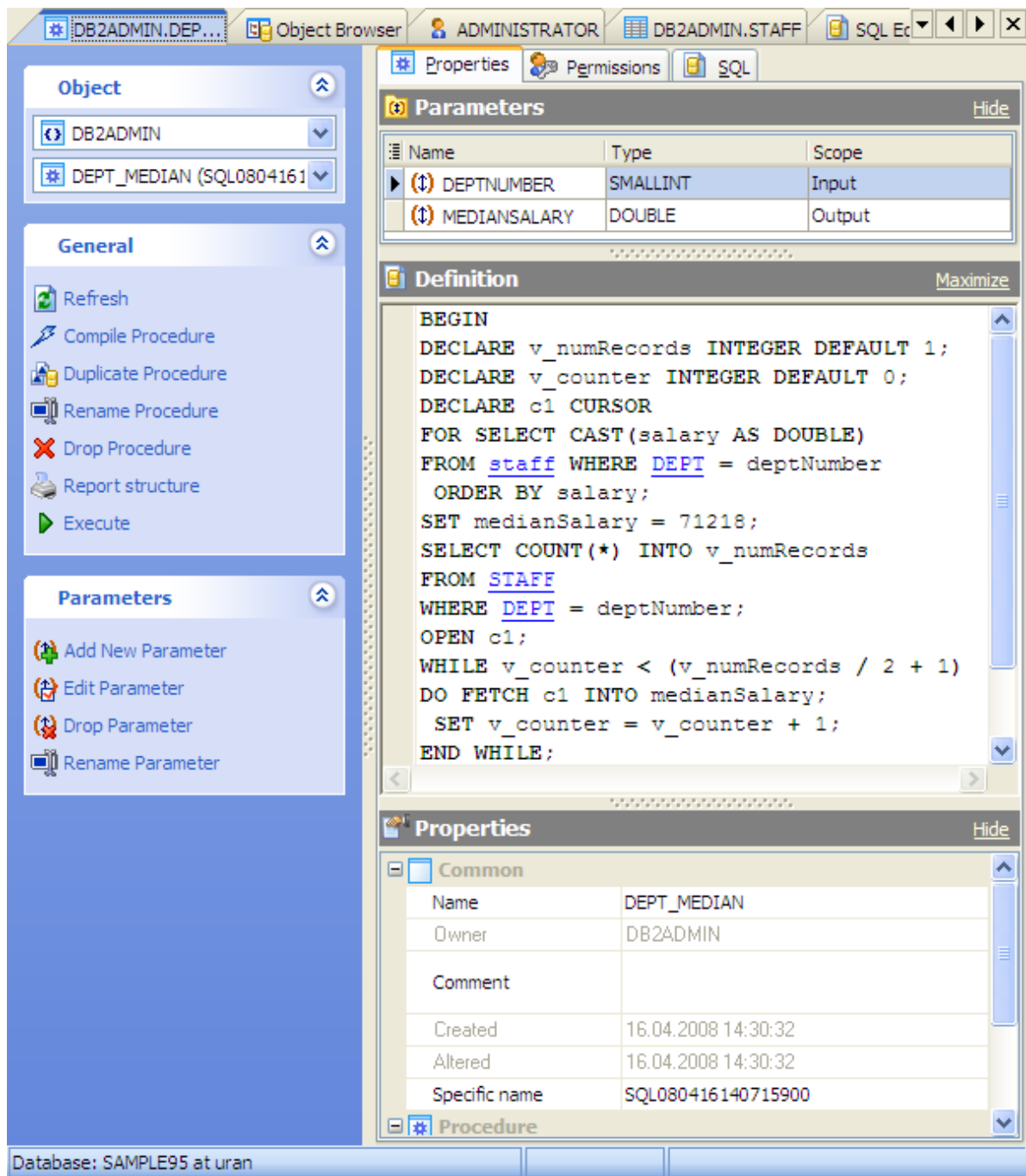
### 5.4.2 Procedure Editor

**Procedure Editor** allows you to edit its definition (*procedure name, parameter list, procedure body, etc.*). It opens when you create a new procedure or edit the existing one (see [How to edit procedure](#)<sup>[101]</sup> for details).

The basic principles of Object Editors in DB2 Maestro are considered in the corresponding [topic](#)<sup>[39]</sup>. See below to find the description of editor tabs that are unique to the current object.

- [Editing procedure properties](#)<sup>[107]</sup>

**See also:** [Create Procedure Wizard](#)<sup>[102]</sup>



#### 5.4.2.1 Editing properties

Parameters can be edited within the [Parameter Editor](#) dialog window. In order to open the dialog you should

- open the object in its editor and the [Parameters](#) tab there;
- select the parameter to edit;
- press the **Enter** key or select the [Edit Parameter](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#)).

### Name

Defines the procedure name.

**Note:** the name of the object must be unique among all the object names in its container. Moreover, all the objects that are source of data need unique names among themselves. You can use any identifier that is allowed by DB2 server.

### Owner

There is the owner for the procedure. By default, only the owner of an object can perform various operations with the object. In order to allow other users to operate it, privileges must be granted. (However, users that have the superuser attribute can always access any object.)

### Comment

This field contains a comment to the procedure.

### Deterministic

Check the option if the procedure always produces the same result for the same input parameters, and uncheck otherwise.

### Specific name

Provides a unique name for the instance of the procedure that is being defined. This specific name can be used when dropping the procedure or commenting on the procedure. It can never be used to invoke the procedure.

### Data access (*Contains SQL, Reads SQL data, Modifies SQL data*)

Indicates the level of data access for SQL statements included in the procedure.

*Contains SQL* indicates that SQL statements that neither read nor modify SQL data can be executed by the procedure. Statements that are not supported in procedures might return a different error.

*Reads SQL data* Indicates that some SQL statements that do not modify SQL data can be included in the procedure.

*Modifies SQL data* indicates that the procedure can execute any SQL statement except statements that are not supported in procedures.

### Dynamic result sets

Indicates the estimated upper bound of returned result sets for the procedure.

To apply the changes, select the [Apply Changes](#) item in the [Navigation bar](#) or use **Ctrl+F9** or **Ctrl+F7** shortcut keys.

It is also possible to modify object properties without opening the object editor: use the [Object Properties](#) item of the popup menu of the selected object from the explorer tree.

## 5.5 External Procedures and Functions

An [external procedure/function](#) is a user-defined function which is determined to the database with a reference to an object code library, and a function within that library that will be executed when the function is invoked. External functions cannot be column functions.

### ■ How can I create a new external procedure?

New External Procedures are created within [Create External Procedure Wizard](#)<sup>[110]</sup>. In order to run the wizard you should either

- select the [Object | Create Database Object...](#) main menu item;
  - select the External Procedure icon in the Create Database Object dialog
- or
- select the [External Procedures](#) list or any object from that list in the explorer tree;
  - select the [Create New External Procedure...](#) item from the popup menu
- or
- open the database in [Database Editor](#) and the [External Procedures](#) tab there;
  - press the **Insert** key or select the [Create New External Procedure...](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#)).

To create a new External Procedure with the same properties as one of the existing External Procedure has:

- select the [Object | Duplicate Database Object...](#) main menu item;
- follow the instructions of [Duplicate Object Wizard](#).

### ■ How can I edit an external procedure definition?

External Procedures can be edited within [External Procedure Editor](#)<sup>[112]</sup>. In order to run the editor you should either

- select the External Procedure for editing in the explorer tree (type the first letters of the External Procedure name for quick search);
  - select the [Edit External Procedure ...](#) item from the popup menu
- or
- open the database in [Database Editor](#) and the [External Procedures](#) tab there;
  - select the External Procedure to edit;
  - press the **Enter** key or select the [Edit External Procedure](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#)).

You can change the name of the External Procedure using the

[Rename External Procedure](#) dialog. To open the dialog you should either

- select the External Procedure to rename in the explorer tree;
- select the [Rename External Procedure](#) item from the popup menu

or

- open the database in [Database Editor](#) and the [External Procedures](#) tab there;
- select the External Procedure to rename;
- select the [Rename External Procedure](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#)).

#### ■ **How can I drop an external procedure?**

To drop a External Procedure:

- select the External Procedure to drop in the explorer tree;
- select the [Drop External Procedure](#) item from the popup menu

or

- open the database in [Database Editor](#) and the [External Procedures](#) tab there;
- select the External Procedure to drop;
- press the **Delete** key or select the [Drop External Procedure](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#))

and confirm dropping in the dialog window.

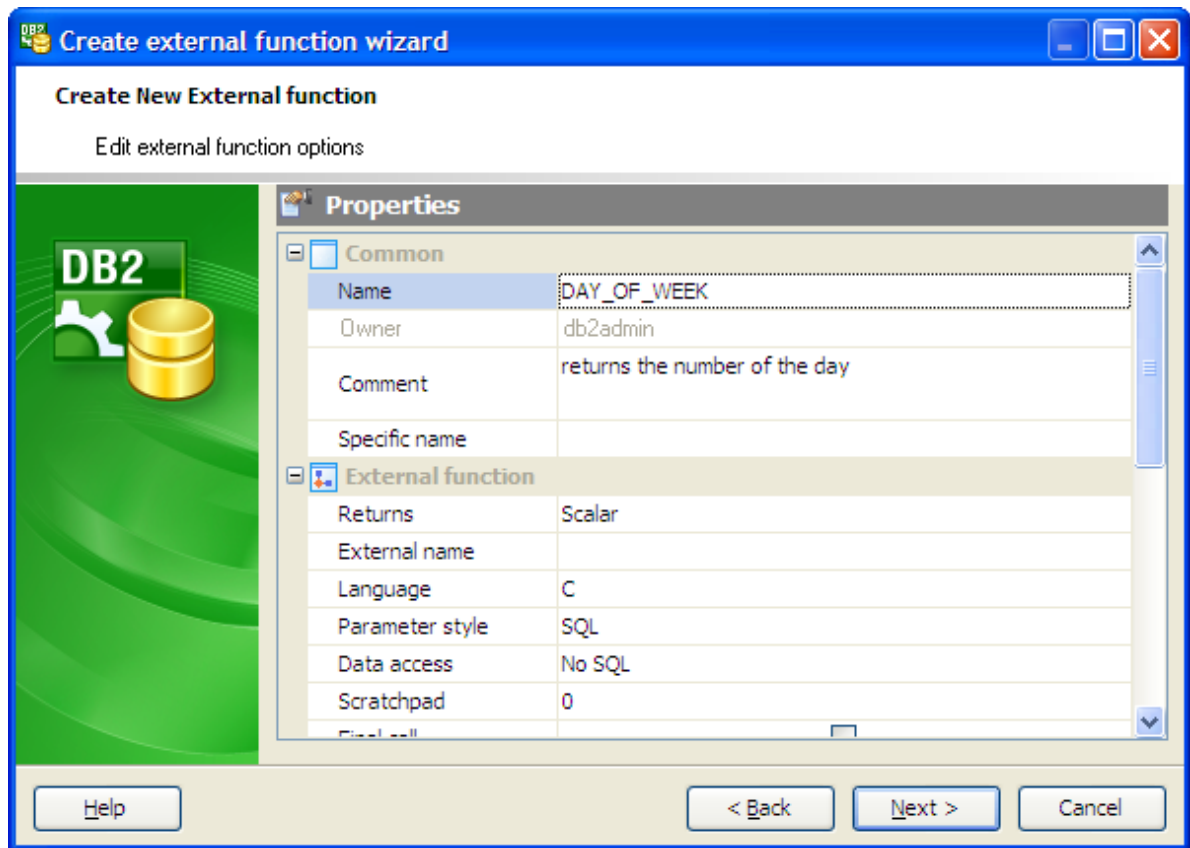
### 5.5.1 Create External Procedure Wizard

[Create External Procedure/Function Wizard](#) guides you through the process of creating a new procedure/function. See [How To Create Procedure](#)<sup>[109]</sup> for instructions on running this wizard.

The basic principles of Create Object Wizards in DB2 Maestro are covered by the corresponding [topic](#)<sup>[36]</sup>. See below to find the description of wizard steps that are unique to the current object.

**See also:** [External Procedure/Function Editor](#)<sup>[112]</sup>

First wizard step allows to set procedure options according to your needs.



#### Name

The new procedure name as it was set on the previous step.

#### Owner

Defines the owner for the procedure. By default, only the owner of an object can perform various operations with the object. In order to allow other users to operate it, privileges must be granted. (However, users that have the superuser attribute can always access any object.)

#### Comment

Specify a comment for the procedure.

#### Parameter style

This clause is used to specify the conventions used for passing parameters to and returning the value from procedures.

#### Specific name

Provides a unique name for the instance of the procedure that is being defined. This specific name can be used when dropping the procedure or commenting on the procedure. It can never be used to invoke the procedure.

#### Data access (Contains SQL, Reads SQL data, Modifies SQL data)

Indicates the level of data access for SQL statements included in the procedure.

*Contains SQL* indicates that SQL statements that neither read nor modify SQL data can be

executed by the procedure. Statements that are not supported in procedures might return a different error.

*Reads SQL data* Indicates that some SQL statements that do not modify SQL data can be included in the procedure.

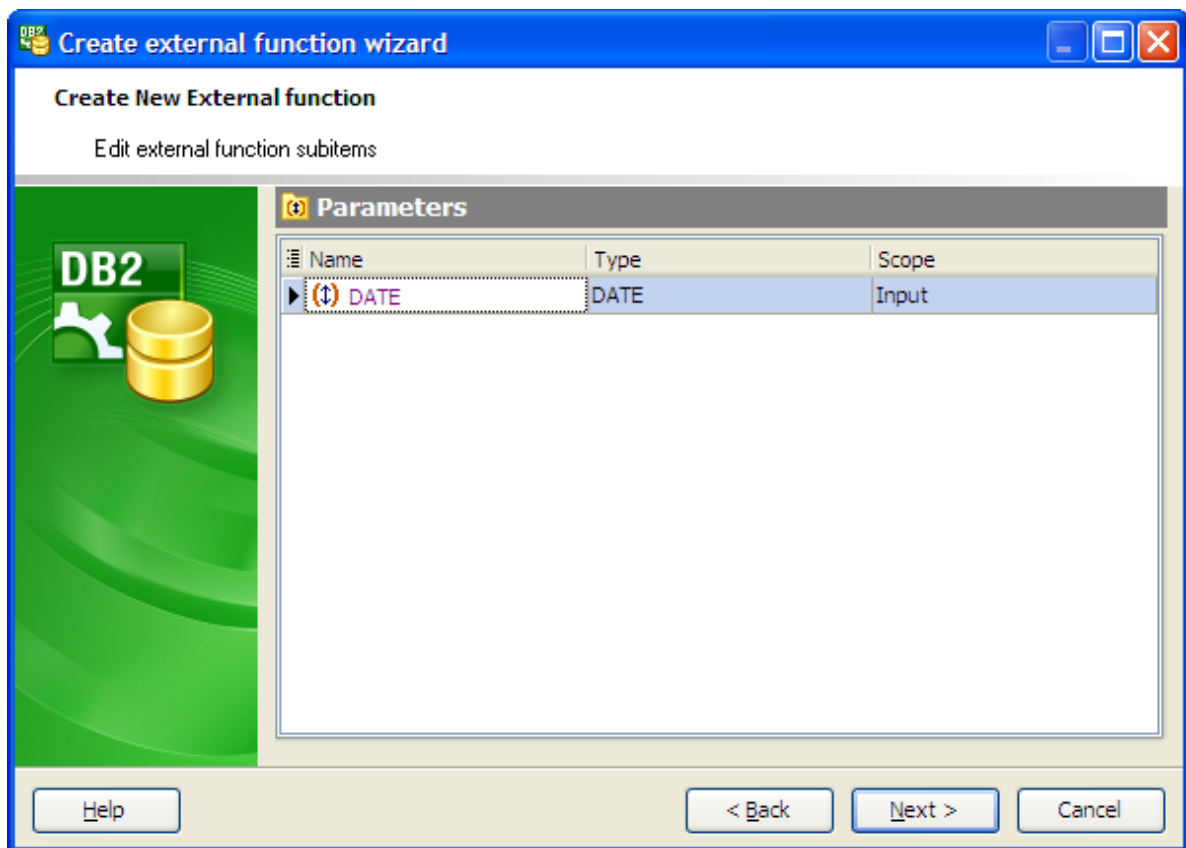
*Modifies SQL data* indicates that the procedure can execute any SQL statement except statements that are not supported in procedures.

#### Dynamic result sets

Indicates the estimated upper bound of returned result sets for the procedure.

The description of the other External Function properties you can find in DB2 SQL Reference Volume 2.

On the second step you can easily specify the procedure parameters. Use the pop-up menu or press **Insert** to add a new parameter and set its properties in [Parameter Editor](#) [43]. Press **Enter** or use the appropriate pop-up menu item to edit the selected parameter, or the **Delete** to delete one.

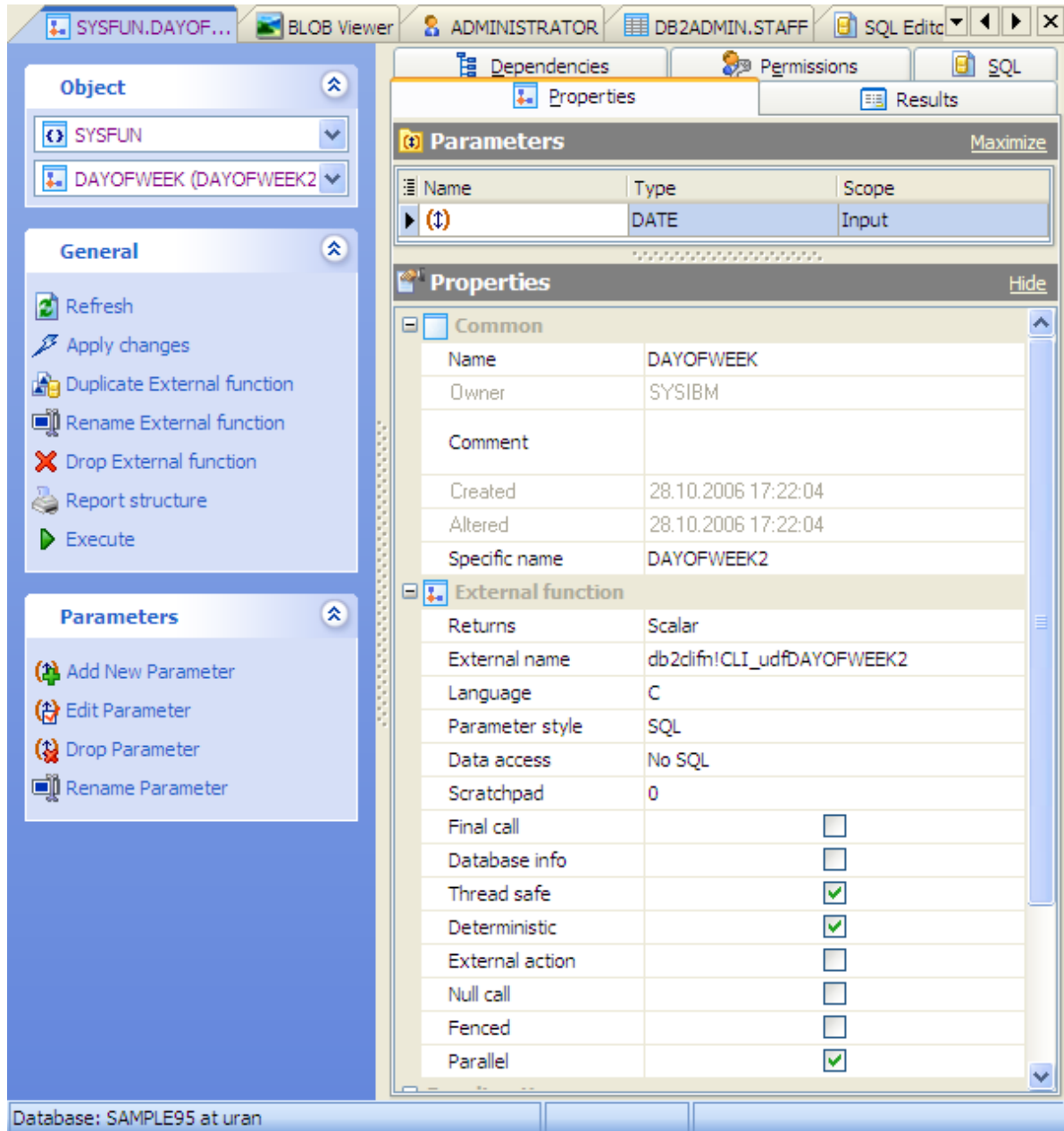


## 5.5.2 External Procedure Editor

[External Procedure/Function Editor](#) allows you to edit its definition (*procedure name*, *parameter list*, *procedure body*, etc.). It opens when you create a new procedure or edit the existing one (see [How to edit procedure](#) [109] for details).

The basic principles of Object Editors in DB2 Maestro are considered in the corresponding [topic](#)<sup>[39]</sup>. See below to find the description of editor tabs that are unique to the current object.

**See also:** [Create External Procedure Wizard](#)<sup>[110]</sup>



Name

The new procedure name as it was set on the previous step.

Owner

Defines the owner for the procedure. By default, only the owner of an object can perform various operations with the object. In order to allow other users to operate it, privileges must be granted. (However, users that have the superuser attribute can always access any object.)

#### Comment

Specify a comment for the procedure.

#### Parameter style

This clause is used to specify the conventions used for passing parameters to and returning the value from procedures.

#### Specific name

Provides a unique name for the instance of the procedure that is being defined. This specific name can be used when dropping the procedure or commenting on the procedure. It can never be used to invoke the procedure.

#### Data access (*Contains SQL, Reads SQL data, Modifies SQL data*)

Indicates the level of data access for SQL statements included in the procedure.

*Contains SQL* indicates that SQL statements that neither read nor modify SQL data can be executed by the procedure. Statements that are not supported in procedures might return a different error.

*Reads SQL data* Indicates that some SQL statements that do not modify SQL data can be included in the procedure.

*Modifies SQL data* indicates that the procedure can execute any SQL statement except statements that are not supported in procedures.

#### Dynamic result sets

Indicates the estimated upper bound of returned result sets for the procedure.

The description of the other External Function properties you can find in DB2 SQL Reference Volume 2.

## 5.6 Functions

A function is an operation that is denoted by a function name followed by a pair of parentheses enclosing the specification of arguments (there may be no arguments). SQL Anywhere allows to define user-specific database functions. In an SQL statement, you can then use these user-defined database functions in the same way as any other predefined functions.

### ■ How can I create a new Function?

New Functions are created within [Create\\_Function\\_Wizard](#)<sup>[116]</sup>. In order to run the wizard you should either

- select the **Object | Create Database Object...** main menu item;
  - select the **Function** icon in the **Create Database Object** dialog
- or
- select the **Functions** list or any object from that list in the explorer tree;
  - select the **Create New Function...** item from the popup menu
- or
- open and the **Functions** tab there;
  - press the **Insert** key or select the **Create New Function** item from the popup menu (alternatively, you may use the corresponding link of the **Navigation Bar**).

To create a new Function with the same properties as one of the existing Functions has:

- select the **Object | Duplicate Database Object...** main menu item;
- follow the instructions of **Duplicate Object Wizard**.

### ■ How can I edit an existing Function definition?

Functions can be edited within [FunctionEditor](#)<sup>[119]</sup>. In order to open the editor you should either

- select the Function for editing in the explorer tree (type the first letters of the Function name for quick search);
  - select the **Edit Function** item from the popup menu
- or
- open and the **Functions** tab there;
  - select the Function to edit;
  - press the **Enter** key or select the **Edit Function** item from the popup menu (alternatively, you may use the corresponding link of the **Navigation Bar**).

You can change the name of the Function using the **Rename Function** dialog:

- select the Function to rename in the explorer tree;
- select the **Rename Function** item from the popup menu.

### ■ How can I execute a Function?

To execute the Function:

- select the Function in the explorer tree (type the first letters of the Function name for quick search);
- select the [Edit Function...](#) item from the popup menu;
- execute the Function using the [Execute](#) link of the [Navigation Bar](#)

or

- open [Schema Editor](#) and the [Functions](#) tab there;
- select the Function to execute;
- press the **Enter** key or select the [Edit Function](#) item from the popup menu, or use the corresponding link of the [Navigation Bar](#);
- execute the Function using the [Execute](#) link of the [Navigation bar](#).

### ■ How can I drop a Function?

To drop a Function:

- select the Function to drop in the explorer tree;
- select the [Drop Function](#) item from the popup menu

or

- open [Schema Editor](#) and the [Functions](#) tab there;
- select the Function to drop;
- press the **Delete** key or select the [Drop Function](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#))

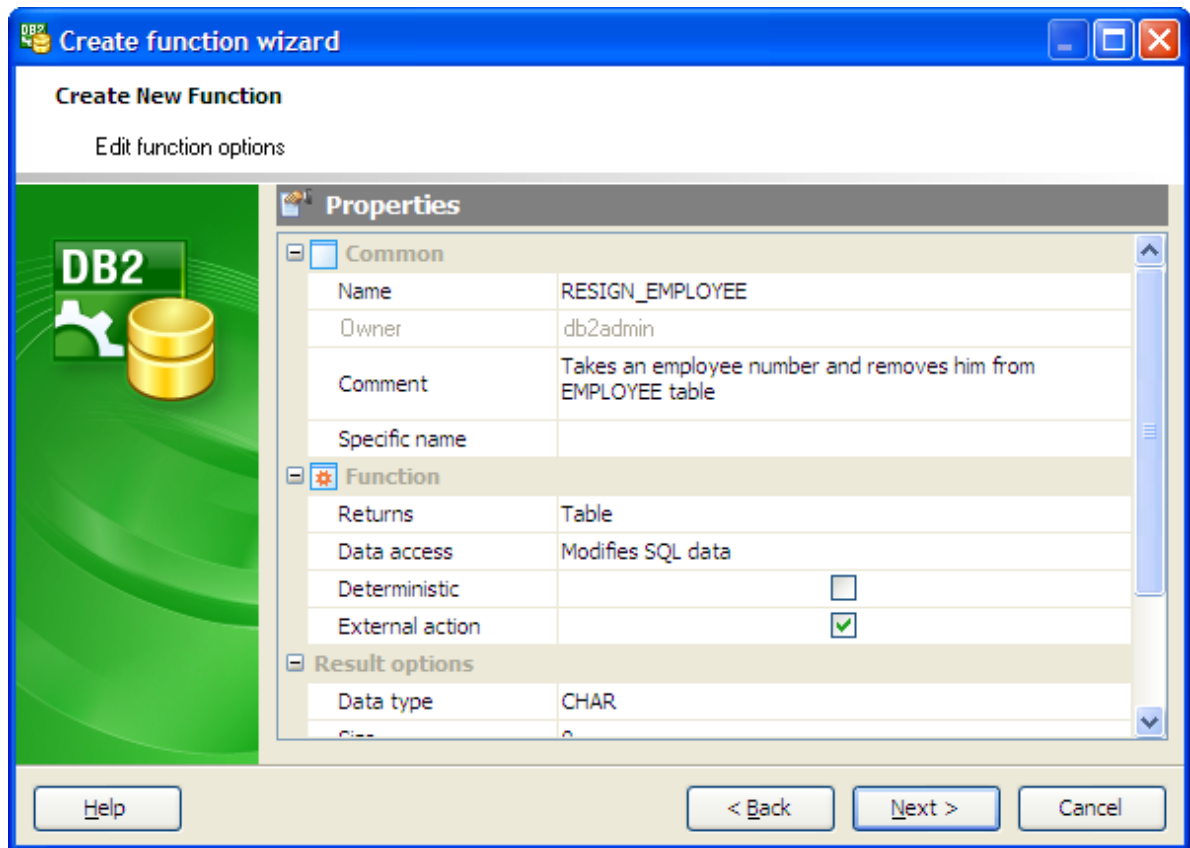
and confirm dropping in the dialog window.

## 5.6.1 Create Function Wizard

[Create Function Wizard](#) guides you through the process of creating a new Function. See [How To create Function](#)<sup>[115]</sup> for instructions on running this wizard.

The basic principles of Create Object Wizards in DB2 Maestro are covered by the corresponding [topic](#)<sup>[36]</sup>. See below to find the description of wizard steps that are unique to the current object.

Specify the properties for the new Function according to your needs. The detailed description is given below.



## Specifying Function properties

### Name

Specify a name for the function.

### Owner

The field displays the owner of the new function. By default, only the owner of an object can perform various operations with the object. In order to allow other users to operate it, privileges must be granted. (However, users that have the superuser attribute can always access any object.)

### Comment

Supply a comment to the Function if necessary.

### ☒ Deterministic

Check the box to indicate that the function returns the same result value whenever it is called with the same values for its arguments

### Specific name

Provides a unique name for the instance of the procedure that is being defined. This specific name can be used when dropping the procedure or commenting on the procedure. It can never be used to invoke the procedure.

### Data access (Contains SQL, Reads SQL data, Modifies SQL data)

Indicates the level of data access for SQL statements included in the procedure.

*Contains SQL* indicates that SQL statements that neither read nor modify SQL data can be executed by the procedure. Statements that are not supported in procedures might return a different error.

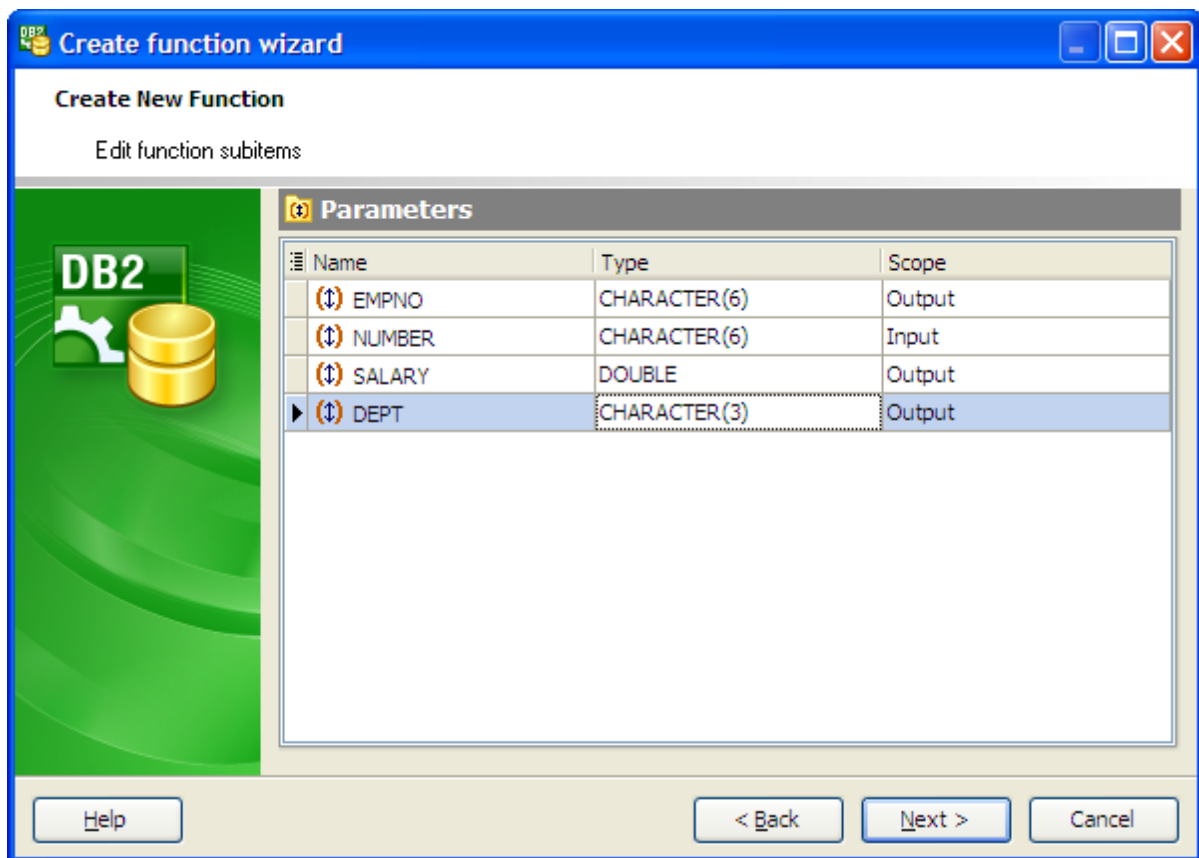
*Reads SQL data* Indicates that some SQL statements that do not modify SQL data can be included in the procedure.

*Modifies SQL data* indicates that the procedure can execute any SQL statement except statements that are not supported in procedures.

#### Dynamic result sets

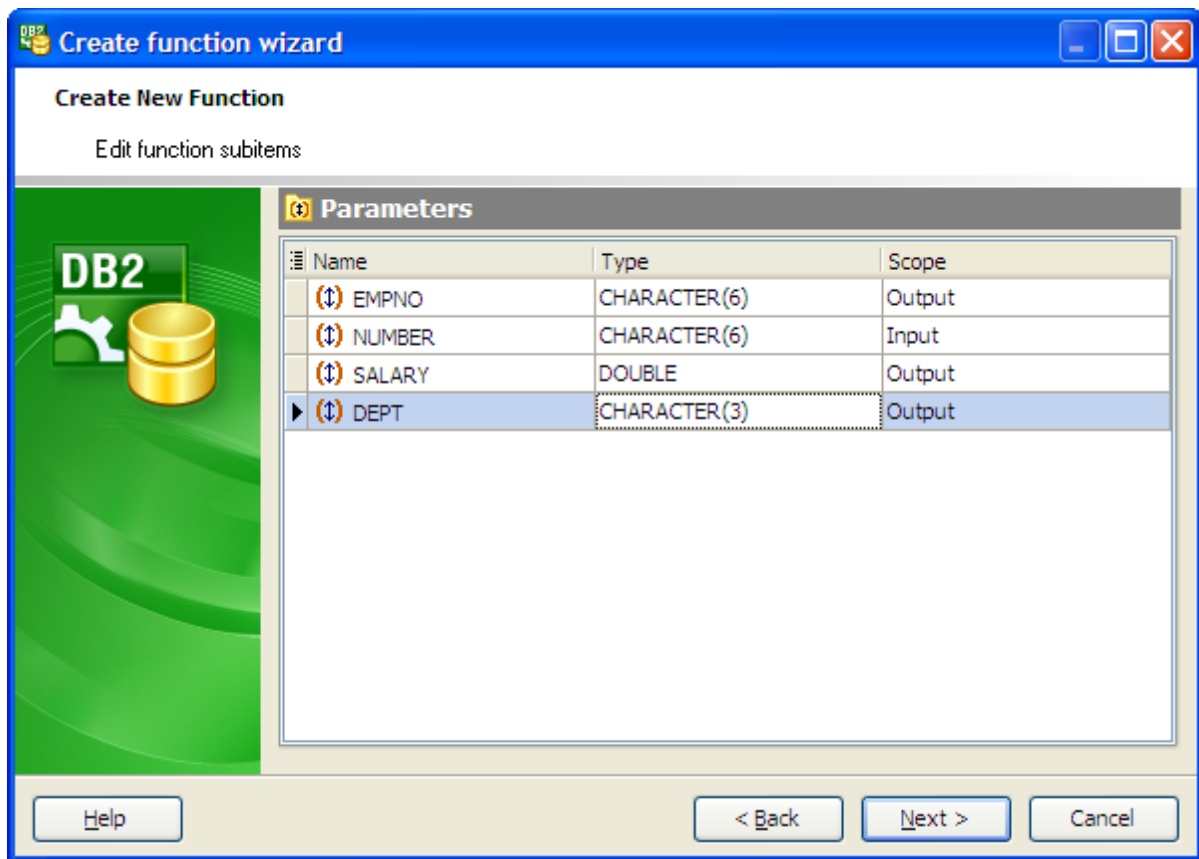
Indicates the estimated upper bound of returned result sets for the procedure.

The description of the other Function properties you can find in DB2 SQL Reference Volume 2.



#### Managing parameters of a new Function

Use popup menu [Add New Parameter...](#) item to add a new parameter and set its properties in [Parameter Editor](#)<sup>43</sup>. Use the [Edit](#) and [Delete](#) items to manage Function parameters.



## 5.6.2 Function Editor

**Function Editor** allows you to execute the existing Function or edit its definition (Function name, parameter list, etc.). It is opened when you create a new Function or edit the existing one (see [How to edit Function](#)<sup>[115]</sup> for details).

The basic principles of Object Editors in DB2 Maestro are considered in the corresponding [topic](#)<sup>[39]</sup>. See below to find the description of editor tabs that are unique to the current object.

- [Editing Function properties](#)<sup>[119]</sup>
- [Viewing Function results](#)<sup>[121]</sup>

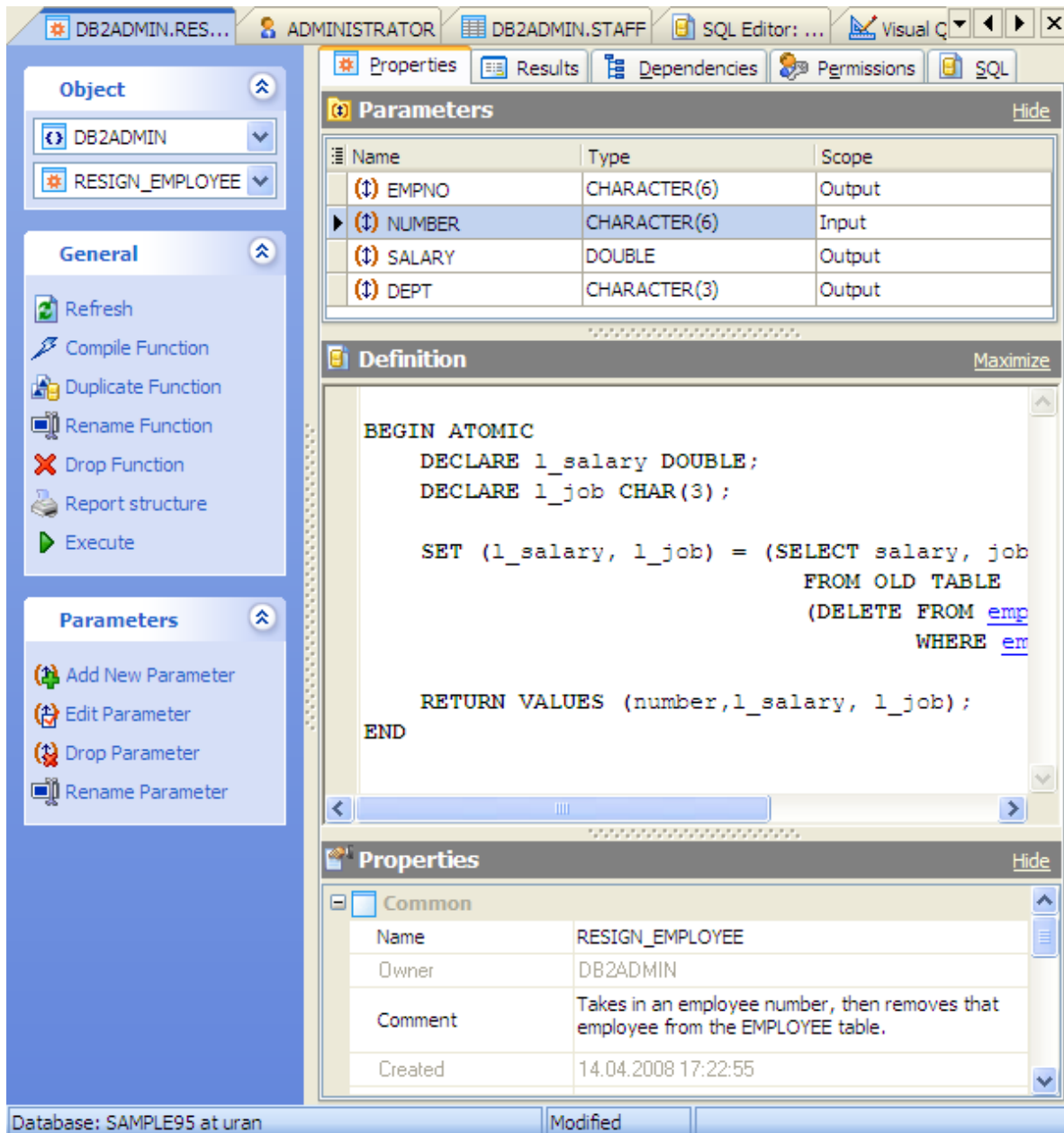
### 5.6.2.1 Editing properties

The **Parameters** tab contains the list of the current Function parameters with its options. Here you can view the **Name** and the **Type** of each Function parameter and also supply a **Comment** for the parameter.

Parameters can be edited within the **Parameter Editor** dialog window. In order to open the dialog you should

- open the object in its editor and the **Parameters** tab there;
- select the parameter to edit;

- press the **Enter** key or select the **Edit Parameter** item from the popup menu (alternatively, you may use the corresponding link of the **Navigation Bar**).  
The **Definition** field contains the definition of the Function. Specify a string constant defining the Function here; the meaning depends on the language. It may be an internal Function name, the path to an object file, an SQL command or text in a procedural language.



### Name

You can edit the Function name here. The name of the Function must be unique among all the Function names in the database.

### Owner

The field contains the owner of the Function. By default, only the owner of an object can perform various operations with the object. In order to allow other users to operate it, privileges must be granted. (However, users that have the superuser attribute can always access any object.)

#### Comment

Specify a comment to the Function if necessary.

#### ☒ Deterministic

Check the box to indicate that the function returns the same result value whenever it is called with the same values for its arguments

#### Specific name

Provides a unique name for the instance of the procedure that is being defined. This specific name can be used when dropping the procedure or commenting on the procedure. It can never be used to invoke the procedure.

#### Data access (*Contains SQL, Reads SQL data, Modifies SQL data*)

Indicates the level of data access for SQL statements included in the procedure.

*Contains SQL* indicates that SQL statements that neither read nor modify SQL data can be executed by the procedure. Statements that are not supported in procedures might return a different error.

*Reads SQL data* Indicates that some SQL statements that do not modify SQL data can be included in the procedure.

*Modifies SQL data* indicates that the procedure can execute any SQL statement except statements that are not supported in procedures.

#### Dynamic result sets

Indicates the estimated upper bound of returned result sets for the procedure.

The description of the other External Function properties you can find in DB2 SQL Reference Volume 2.

To apply the changes, select the [Apply Changes](#) item in the [Navigation bar](#) or use **Ctrl+F9** or **Ctrl+F7** shortcut keys.

It is also possible to modify object properties without opening the object editor: use the [Object Properties](#) item of the popup menu of the selected object from the explorer tree.

### 5.6.2.2 Viewing Function results

The [Results](#) tab represents the result of the successfully executed Function.

**See also:** [Executing functions](#) 

The screenshot displays the SQL Maestro Group application window. The interface includes a top toolbar with icons for Properties, Results, Dependencies, Permissions, and SQL. Below the toolbar, the Results tab is active, showing a query result table. The table has three columns: EMPNO, SALARY, and DEPT. The first row of data shows EMPNO 200210, SALARY NULL, and DEPT NULL. The interface also features a left sidebar with sections for Object (DB2ADMIN, RESIGN\_EMPLOYEE), General (Refresh, Compile Function, Duplicate Function, Rename Function, Drop Function, Report structure, Execute), and Data Management (Export data, Get SQL dump, Print data). At the bottom, a status bar indicates the database is SAMPLE95 at uran. An Information panel at the bottom right shows '1 rows fetched ( 0,27 sec)'.

Database: SAMPLE95 at uran

EMPNO	SALARY	DEPT
200210	NULL	NULL

Records fetched: 1

Information  
1 rows fetched ( 0,27 sec)

## 5.7 Distinct Types

### ■ How can I create a new Distinct Type?

New Distinct Types are created within [Create Distinct Type Wizard](#) <sup>[124]</sup>. In order to run the wizard you should either

- select the [Object | Create Database Object...](#) main menu item;
- select the [Distinct Type](#) icon in the [Create Database Object](#) dialog

or

- select the [Distinct Types](#) list or any object from that list in the explorer tree;
- select the [Create New Distinct Type...](#) item from the popup menu

or

- open and the [Distinct Types](#) tab there;
- press the **Insert** key or select the [Create New Distinct Type](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#)).

To create a new Distinct Type with the same properties as one of the existing Distinct Types has:

- select the [Object | Duplicate Database Object...](#) main menu item;
- follow the instructions of [Duplicate Object Wizard](#).

### ■ How can I edit an existing Distinct Type?

Distinct Types can be edited within [Distinct Type Editor](#) <sup>[125]</sup>. In order to run the editor you should either

- select the Distinct Type for editing in the explorer tree (type the first letters of the Distinct Type name for quick search);
- select the [Edit Distinct Type...](#) item from the popup menu

or

- open and the [Distinct Types](#) tab there;
- select the Distinct Type to edit;
- press the **Enter** key or select the [Edit Distinct Type](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#)).

You can change the name of the Distinct Type using the [Rename Distinct Type](#) dialog. To open the dialog you should either

- select the Distinct Type to rename in the explorer tree;
- select the [Rename Distinct Type](#) item from the popup menu

or

- open [Schema Editor](#) and the [Distinct Types](#) tab there;
- select the Distinct Type to rename;
- select the [Rename Distinct Type](#) item from the popup menu

(alternatively, you may use the corresponding link of the [Navigation Bar](#)).

### ■ How can I drop a Distinct Type?

To drop a Distinct Type:

- select the Distinct Type to drop in the explorer tree;
  - select the [Drop Distinct Type](#) item from the popup menu
- or
- open [Schema Editor](#) and the [Distinct Types](#) tab there;
  - select the Distinct Type to drop;
  - press the **Delete** key or select the [Drop Distinct Type](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#))

and confirm dropping in the dialog window.

## 5.7.1 Create Distinct Type Wizard

[Create Distinct Type Wizard](#) guides you through the process of creating a new Distinct Type. See [How To Create Distinct Type](#)<sup>[123]</sup> to learn how to run this wizard.

The basic principles of Create Object Wizards in DB2 Maestro are covered by the corresponding [topic](#)<sup>[36]</sup>. See below to find the description of wizard steps that are unique to the current object.

- [Specifying Distinct Type Options](#)<sup>[124]</sup>

**See also:** [Distinct Type Editor](#)<sup>[125]</sup>

### 5.7.1.1 Specifying Distinct Type properties

Specify the properties for the new Distinct Type according to your needs. The detailed description is given below.

**Create distinct type wizard**

**Create New Distinct type**

Edit distinct type options

**Properties**

☐ **Common**

Name	T_PHONE_NUMBER
Owner	db2admin
Comment	

☒ **Distinct type**

Source type name	NUMERIC
Size	9
Precision	1
LOB unit	Bytes
Bit data	<input type="checkbox"/>

Help < Back Next > Cancel

#### Comment

Supply a comment to the Distinct Type if necessary.

#### Source type name

Specifies the data type used as the basis for the internal representation of the distinct type. The source data type cannot be of type XML.

#### Precision

The first integer is the precision of the number; that is, the total number of digits; it may range from 1 to 31.

#### BIT data

Specifies that the contents of the column are to be treated as bit (binary) data. During data exchange with other systems, code page conversions are not performed. Comparisons are done in binary, irrespective of the database collating sequence.

LOBs greater than 1 gigabyte cannot be logged and LOBs greater than 10 megabytes should probably not be logged.

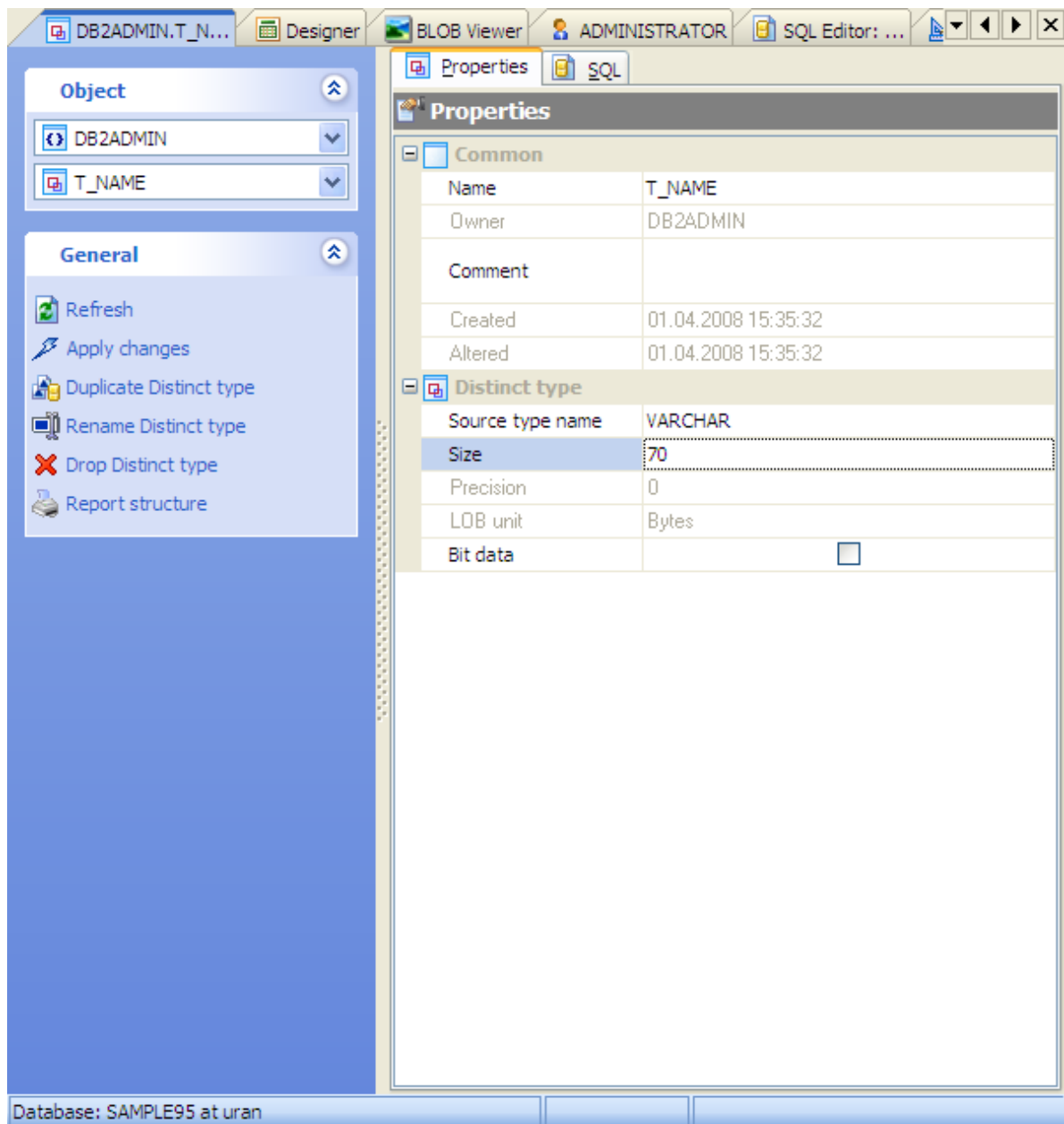
## 5.7.2 Distinct Type Editor

**Distinct Type Editor** is opened automatically after a new Distinct Type is created and is available on editing the existing one (see [Edit Distinct Type](#)<sup>[123]</sup> for details).

The basic principles of Object Editors in DB2 Maestro are considered in the corresponding [topic](#)<sup>[39]</sup>. See below to find the description of editor tabs that are unique to the current object.

For more information about the editor's parts and their element see [Distinct Type properties](#)<sup>[127]</sup>.

**See also:** [Create Distinct Type Wizard](#)<sup>[124]</sup>



### 5.7.2.1 Editing properties

[Distinct Type Editor](#) provides you with an ability to edit Distinct Type properties. The [Properties](#) tab allows you to change the Distinct Type name, the Distinct Type owner, etc.

#### [Name](#)

Here you can view and edit the Distinct Type name. The name of the Distinct Type must be unique among all the Distinct Type names in the schema.

#### [Comment](#)

This field stores a comment to the Distinct Type.

#### [Source type name](#)

Specifies the data type used as the basis for the internal representation of the distinct type. The source data type cannot be of type XML.

#### [Precision](#)

The first integer is the precision of the number; that is, the total number of digits; it may range from 1 to 31.

#### [BIT data](#)

Specifies that the contents of the column are to be treated as bit (binary) data. During data exchange with other systems, code page conversions are not performed. Comparisons are done in binary, irrespective of the database collating sequence.

LOBs greater than 1 gigabyte cannot be logged and LOBs greater than 10 megabytes should probably not be logged.

To apply the changes, select the [Apply Changes](#) item in the [Navigation bar](#) or use **Ctrl+F9** or **Ctrl+F7** shortcut keys.

It is also possible to modify object properties without opening the object editor: use the [Object Properties](#) item of the popup menu of the selected object from the explorer tree.

## 5.8 Aliases

An [alias](#) is an alternative name for a table or a view. It can be used to reference a table or a view if an existing table or view can be referenced. An alias cannot be used in all contexts; for example, it cannot be used in the check condition of a check constraint. An alias cannot reference a declared temporary table.

Like tables or views, an alias can be created, dropped, and have comments associated with it. However, unlike tables, aliases can refer to each other in a process called chaining. Aliases are publicly referenced names, so no special authority or privilege is required to use them. Access to the table or the view referred to by an alias, however, does require the authorization associated with these objects.

### ■ How can I create a new alias?

New aliases are created within [Create Alias Wizard](#)<sup>[129]</sup>. In order to run the wizard you should either

- select the [Object | Create Database Object...](#) main menu item;
  - select the [Alias](#) icon in the [Create Database Object](#) dialog
- or
- select the [Aliases](#) list or any object from that list in the explorer tree;
  - select the [Create New Alias...](#) item from the popup menu
- or
- open the schema in [Schema Editor](#) and the [Aliases](#) tab there;
  - press the **Insert** key or select the [Create New Alias...](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#)).

To create a new alias with the same properties as one of the existing aliases has:

- select the [Object | Duplicate Database Object...](#) main menu item;
- follow the instructions of [Duplicate Object Wizard](#).

### ■ How can I edit an existing alias?

Aliases can be edited within [Alias Editor](#)<sup>[131]</sup>. In order to run the editor you should either

- select the alias for editing in the explorer tree (type the first letters of the alias name for quick search);
  - select the [Edit Alias ...](#) item from the popup menu
- or
- open the schema in [Schema Editor](#) and the [Aliases](#) tab there;
  - select the alias to edit;
  - press the **Enter** key or select the [Edit Alias](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#)).

You can change the name of the alias using the [Rename Alias](#) dialog. To open the dialog you should either

- select the alias to rename in the explorer tree;
- select the [Rename Alias](#) item from the popup menu

or

- open the schema in [Schema Editor](#) and the [Aliases](#) tab there;
- select the alias to rename;
- select the [Rename Alias](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#)).

#### ■ **How can I drop an alias?**

To drop a alias:

- select the alias to drop in the explorer tree;
- select the [Drop Alias](#) item from the popup menu

or

- open the schema in [Schema Editor](#) and the [Aliases](#) tab there;
- select the alias to drop;
- press the **Delete** key or select the [Drop Alias](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#))

and confirm dropping in the dialog window.

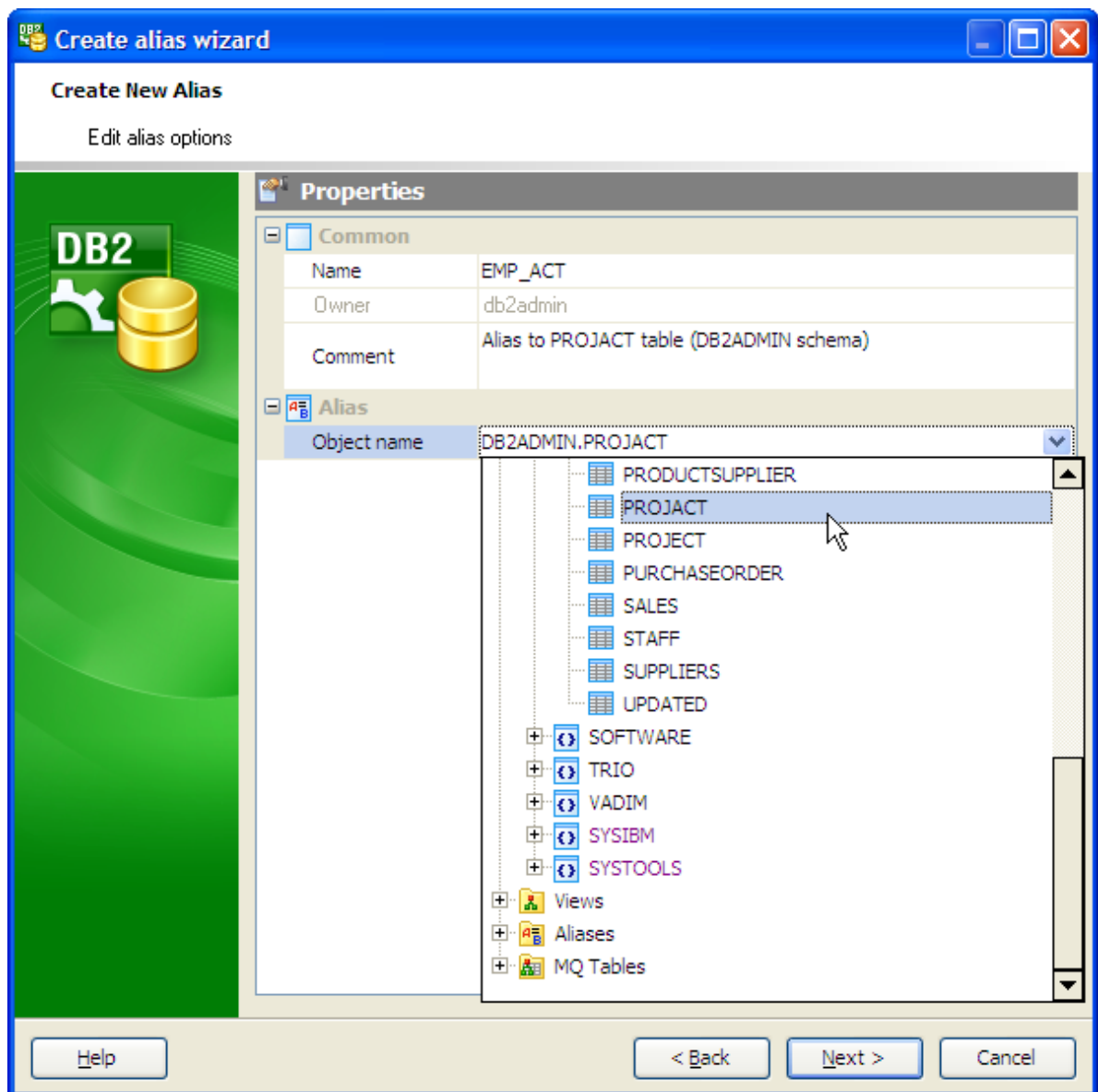
## 5.8.1 Create Alias Wizard

[Create Alias Wizard](#) guides you through the process of creating a new Alias. See [How To Create Alias](#)<sup>[128]</sup> for instructions on running this wizard.

The basic principles of Create Object Wizards in DB2 Maestro are covered by the corresponding [topic](#)<sup>[36]</sup>. See below to find the description of wizard steps that are unique to the current object.

- [Specifying Alias options](#)<sup>[130]</sup>

**See also:** [Alias Editor](#)<sup>[131]</sup>



#### 5.8.1.1 Specifying alias options

Specify alias options according to your needs.

##### Name

Specify a name for the alias. The name must not identify a table, view, nickname, or alias that exists in the current database. If a two-part name is specified, the schema name cannot begin with 'SYS'. The rules for defining an alias name are the same as those used for defining a table name.

##### Comment

Specify a comment to the alias or leave the field blank.

##### Object name

Specify the base object that the alias references. Select from drop-down list a table, view, nickname, or alias for which alias-name is defined. If another alias name is supplied

(alias-name2), then it must not be the same as the new alias-name being defined (in its fully-qualified form). The table-name cannot be a declared temporary table.

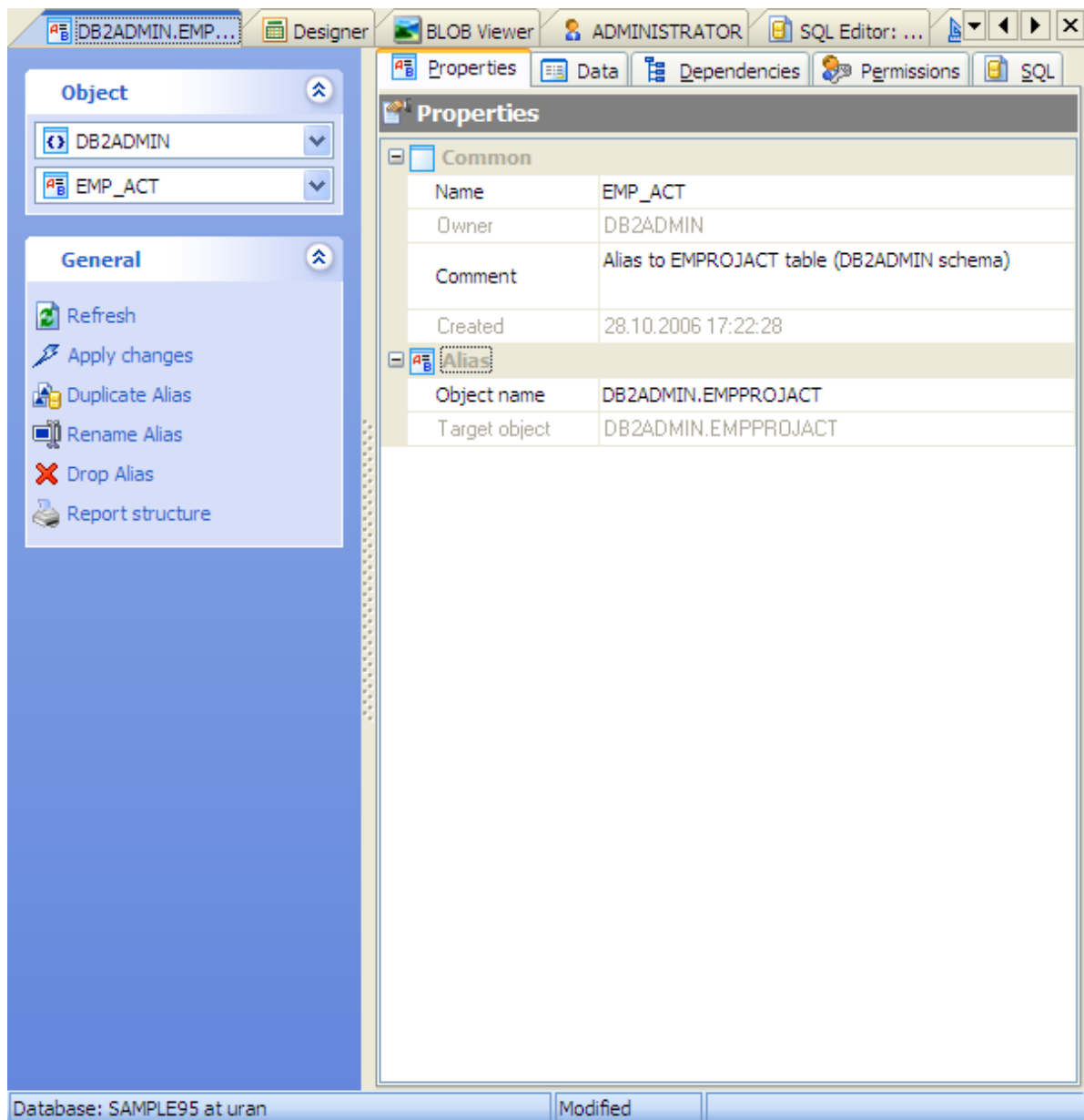
## 5.8.2 Alias Editor

[Alias Editor](#) allows you edit Alias properties. It opens when you create a new Alias or edit the existing one (see [How to edit Alias](#)<sup>[128]</sup> for details).

The basic principles of Object Editors in DB2 Maestro are considered in the corresponding [topic](#)<sup>[39]</sup>. See below to find the description of editor tabs that are unique to the current object.

- [Editing Alias properties](#)<sup>[132]</sup>
- [Managing Alias data](#)<sup>[133]</sup>

**See also:** [Create Alias Wizard](#)<sup>[129]</sup>



#### 5.8.2.1 Editing alias properties

Specify alias options according to your needs. The detailed description is given below.

##### Name

You can edit the alias name here.

##### Owner

Here you can view the owner for the alias. By default, only the owner of an object can perform various operations with the object. In order to allow other users to operate it, privileges must be granted. (However, users that have the superuser attribute can always access any object.)

##### Comment

Specify a comment to the alias.

[Create Date](#)

Stores the date when the alias was created.

[Modify Date](#)

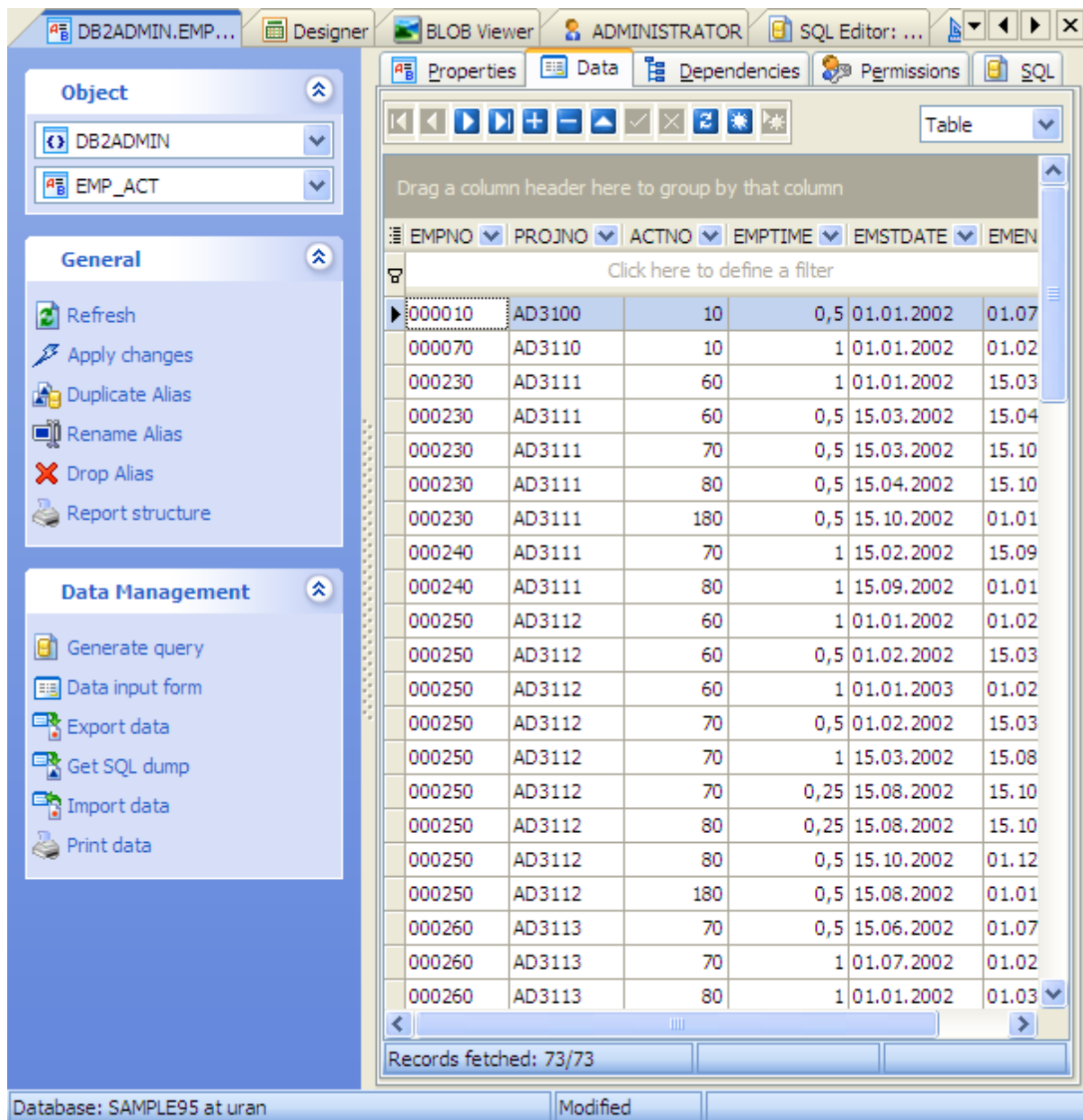
Stores the date when the alias was last modified.

[Object name](#)

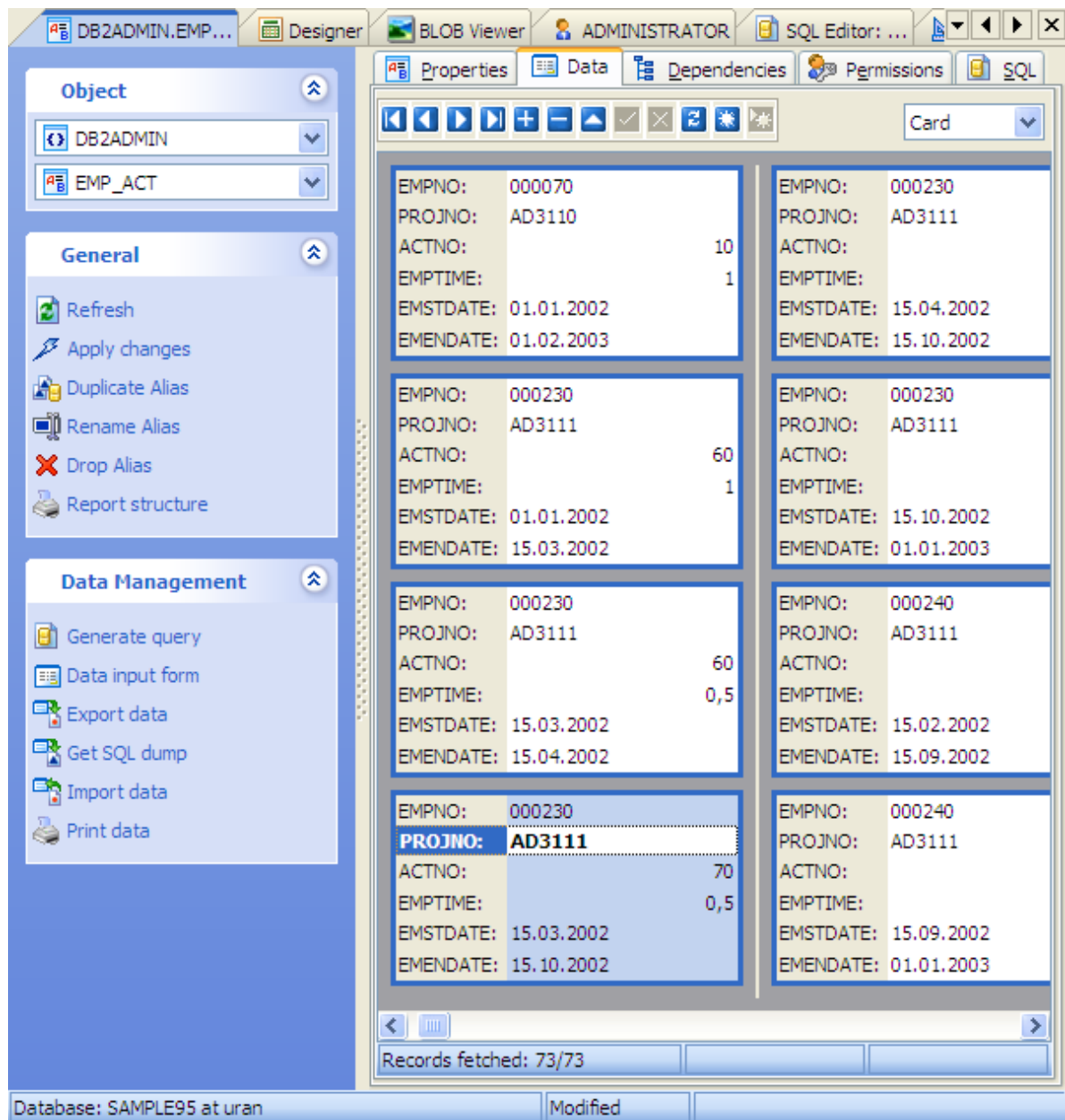
Here you can change the base object that the alias references. Just select from drop-down list a new table, view, nickname, or alias for which alias-name is defined.

#### 5.8.2.2 Managing alias data

The [Data](#) tab displays the Alias data as a grid (see Data View for details). Use grid's popup menu to open [Data Input Form](#), to invoke the [Export Data](#), and Get [SQL Dump](#) modules, to set the value of the selected record to *NULL* or to *Now* (for *Date* values).



For your convenience it was implemented two modes of viewing data: as table and as info cards.



## 5.9 Sequences

Sequence is a database object from which multiple users may generate unique integers. You can use sequences to automatically generate primary key values.

When a sequence number is generated, the sequence is incremented, independent of the transaction committing or rolling back. If two users concurrently increment the same sequence, then the sequence numbers each user acquires may have gaps, because sequence numbers are being generated by the other user. One user can never acquire the sequence number generated by another user. Once a sequence value is generated by one user, that user can continue to access that value regardless of whether the sequence is incremented by another user.

### ■ How can I create a new sequence?

New sequences are created within [Create Sequence Wizard](#)<sup>137</sup>. In order to run the wizard you should either

- select the [Object | Create Database Object...](#) main menu item;
  - select the [Sequence](#) icon in the Create Database Object dialog
- or
- select the [Sequences](#) list or any object from that list in the explorer tree;
  - select the [Create New Sequence...](#) item from the popup menu
- or
- open the schema in [Schema Editor](#) and the [Sequences](#) tab there;
  - press the **Insert** key or select the [Create New Sequence](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#)).

To create a new sequence with the same properties as one of the existing sequences has:

- select the [Object | Duplicate Database Object...](#) main menu item;
- follow the instructions of [Duplicate Object Wizard](#).

### ■ How can I edit an existing sequence?

Sequences can be edited within [Sequence Editor](#)<sup>138</sup>. In order to run the editor you should either

- select the sequence for editing in the explorer tree (type the first letters of the sequence name for quick search);
  - select the [Edit Sequence ...](#) item from the popup menu
- or
- open the schema in [Schema Editor](#) and the [Sequences](#) tab there;
  - select the sequence to edit;
  - press the **Enter** key or select the [Edit Sequence](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#)).

### ■ How can I drop a sequence

To drop a sequence:

- select the sequence to drop in the explorer tree;
  - select the [Drop Sequence](#) item from the popup menu
- or
- open the schema in [Schema Editor](#) and the [Sequences](#) tab there;
  - select the sequence to drop;
  - press the **Delete** key or select the [Drop Sequence](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#))

and confirm dropping in the dialog window.

## 5.9.1 Create Sequence Wizard

Sequences may be created with [Create Sequence Wizard](#). Just specify the wizard options according to your needs.

The basic principles of Create Object Wizards in DB2 Maestro are covered by the corresponding [topic](#)<sup>[36]</sup>. See below to find the description of wizard steps that are unique to the current object.

### Name

The field allows you to specify the new sequence name being set on the previous wizard step.

### Owner

Defines the owner of the new sequence.[Start Value](#)

Specify the starting sequence value here. The default starting value is *minvalue* for ascending sequences and *maxvalue* for descending ones.

### Increment by

Specify which value is added to the current sequence value to create a new value. A positive value will make an ascending sequence, a negative one a descending sequence. The default value is 1.

### Maximum Value

Determine the maximum value for the sequence. If this clause is not supplied or NO MAXVALUE is specified, then default values will be used. The defaults are  $2^{63}-1$  and  $-1$  for ascending and descending sequences, respectively.

### Minimum Value

Determine the minimum value a sequence can generate. If this clause is not supplied or NO MINVALUE is specified, then defaults will be used. The defaults are 1 and  $-2^{63}+1$  for ascending and descending sequences, respectively.

### Cashe size

Specify how many sequence numbers are to be preallocated and stored in memory for faster access. The minimum value is 1 (only one value can be generated at a time, i.e.,

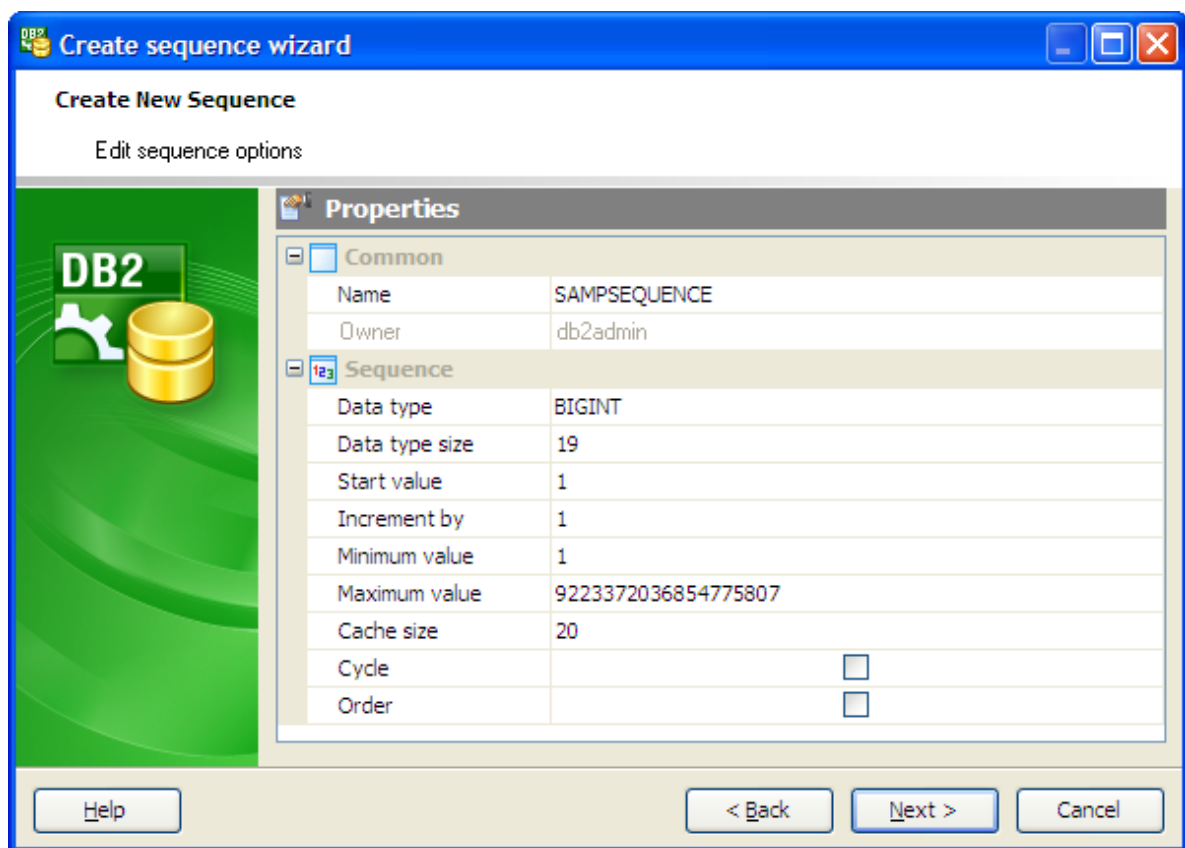
no cache), and this is also the default.

☒ **Order**

Specify the option to guarantee that sequence numbers are generated in order of request. This clause is useful if you are using the sequence numbers as timestamps. Guaranteeing order is usually not important for sequences used to generate primary keys.

☒ **Cycle**

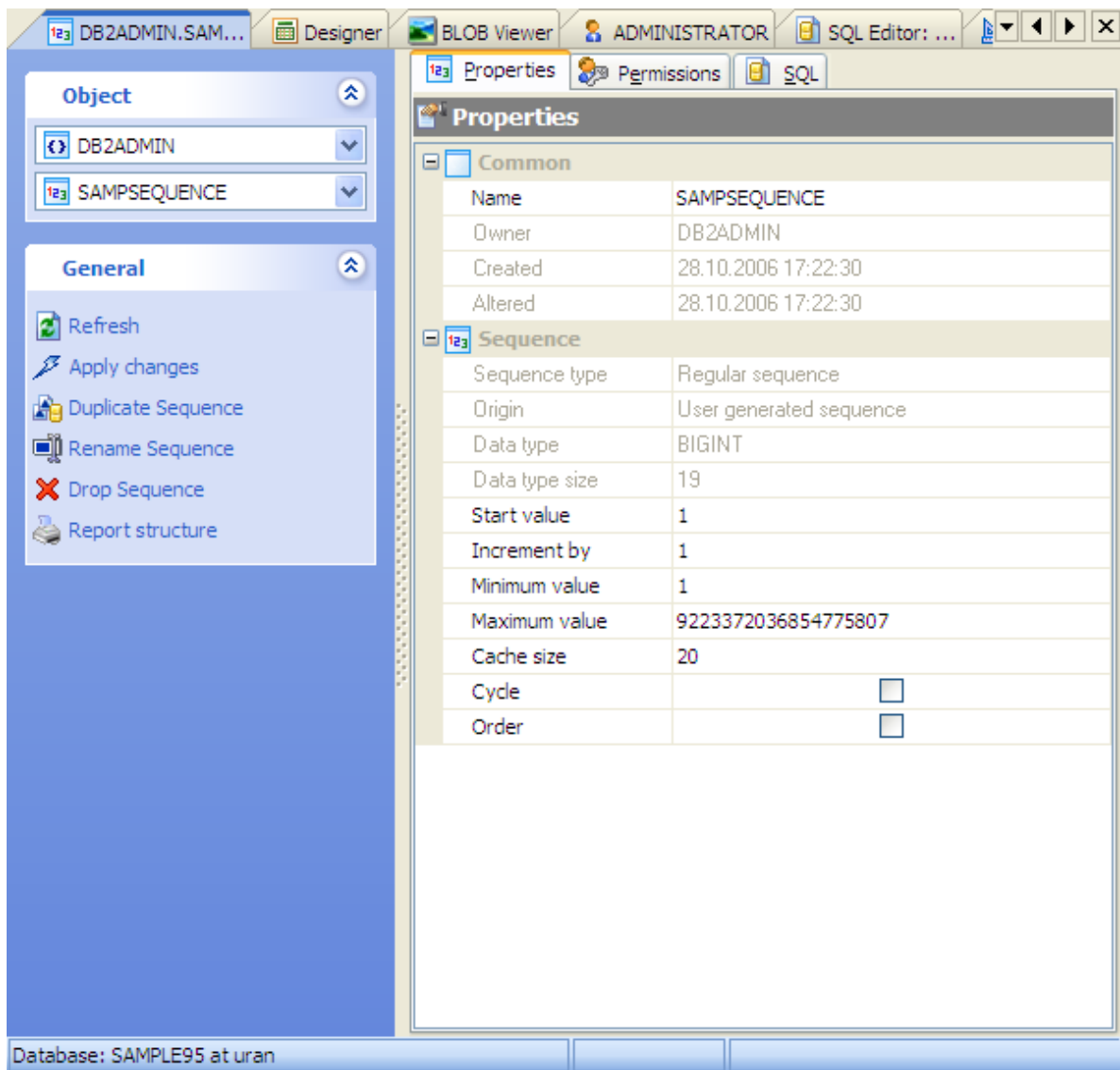
The CYCLE option allows the sequence to wrap around when the *maxvalue* or *minvalue* has been reached by an ascending or descending sequence respectively. If the limit is reached, the next number generated will be the minvalue or maxvalue, respectively.



## 5.9.2 Sequence Editor

Use [Sequence Editor](#) to change properties of existing sequences. The editor can be opened automatically after [the sequence is created](#)<sup>[137]</sup> or from the [Explorer Tree](#) and [Object Manager](#).

The basic principles of Object Editors in DB2 Maestro are considered in the corresponding [topic](#)<sup>[38]</sup>. See below to find the description of editor tabs that are unique to the current object.



### Name

Here you can rename the sequence.

### Owner

Shows the owner of the sequence. By default, only the owner of an object can perform various operations with the object. In order to allow other users to operate it, privileges must be granted. (However, users that have the superuser attribute can always access any object.) **Start Value** contains the starting sequence value here.

### Increment by

Here you can edit the value which is added to the current sequence value to create a new value.

The **Maximum Value** and the **Minimum Value** contain the maximum and the minimum values for the sequence.

### Cashe size

Specify how many sequence numbers are to be preallocated and stored in memory for faster access.

☒ **Order**

Specify the option to guarantee that sequence numbers are generated in order of request. This clause is useful if you are using the sequence numbers as timestamps. Guaranteeing order is usually not important for sequences used to generate primary keys.

☒ **Cycle**

The checkbox represents whether the sequence is cycle.

To apply the changes, select the [Apply Changes](#) item in the [Navigation bar](#) or use **Ctrl+F9** or **Ctrl+F7** shortcut keys.

It is also possible to modify object properties without opening the object editor: use the [Object Properties](#) item of the popup menu of the selected object from the explorer tree.

## 5.10 Structured Types

A [structured type](#) is a user-defined data type that contains one or more named attributes. Each attribute has a name and a data type of its own. Attributes are properties that describe an instance of a type. A structured type can serve as the type of a table, in which each column of the table derives its name and data type from one of the attributes of the structured type.

### ■ How can I create a new structured type?

New Structured types are created within [Create Structured Type Wizard](#)<sup>[142]</sup>. In order to run the wizard you should either

- select the [Object | Create Database Object...](#) main menu item;
- select the [Structured Type](#) icon in the [Create Database Object](#) dialog

or

- select the [Structured Types](#) list or any object from that list in the explorer tree;
- select the [Create New Structured Type...](#) item from the popup menu

or

- open the schema in [Schema Editor](#) and the [Structured Types](#) tab there;
- press the **Insert** key or select the [Create New Structured Type](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#)).

To create a new type with the same properties as one of the existing types has:

- select the [Object | Duplicate Database Object...](#) main menu item;
- follow the instructions of [Duplicate Object Wizard](#).

### ■ How can I edit an existing structured type?

Structured Types can be edited within [Structured Type Editor](#)<sup>[146]</sup>. In order to run the editor you should either

- select the type for editing in the explorer tree (type the first letters of the type name for quick search);
- select the [Edit Structured Type ...](#) item from the popup menu

or

- open the schema in [Schema Editor](#)<sup>[62]</sup> and the [Structured Types](#) tab there;
- select the type to edit;
- press the **Enter** key or select the [Edit Structured Type](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#)).

### ■ How can I drop a structured type?

To drop a structured type:

- select the type to drop in the explorer tree;
  - select the [Drop Structured Type](#) item from the popup menu
- or
- open the schema in [Schema Editor](#) and the [Structured Types](#) tab there;
  - select the type to drop;
  - press the **Delete** key or select the [Drop Structured Type](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#))

and confirm dropping in the dialog window.

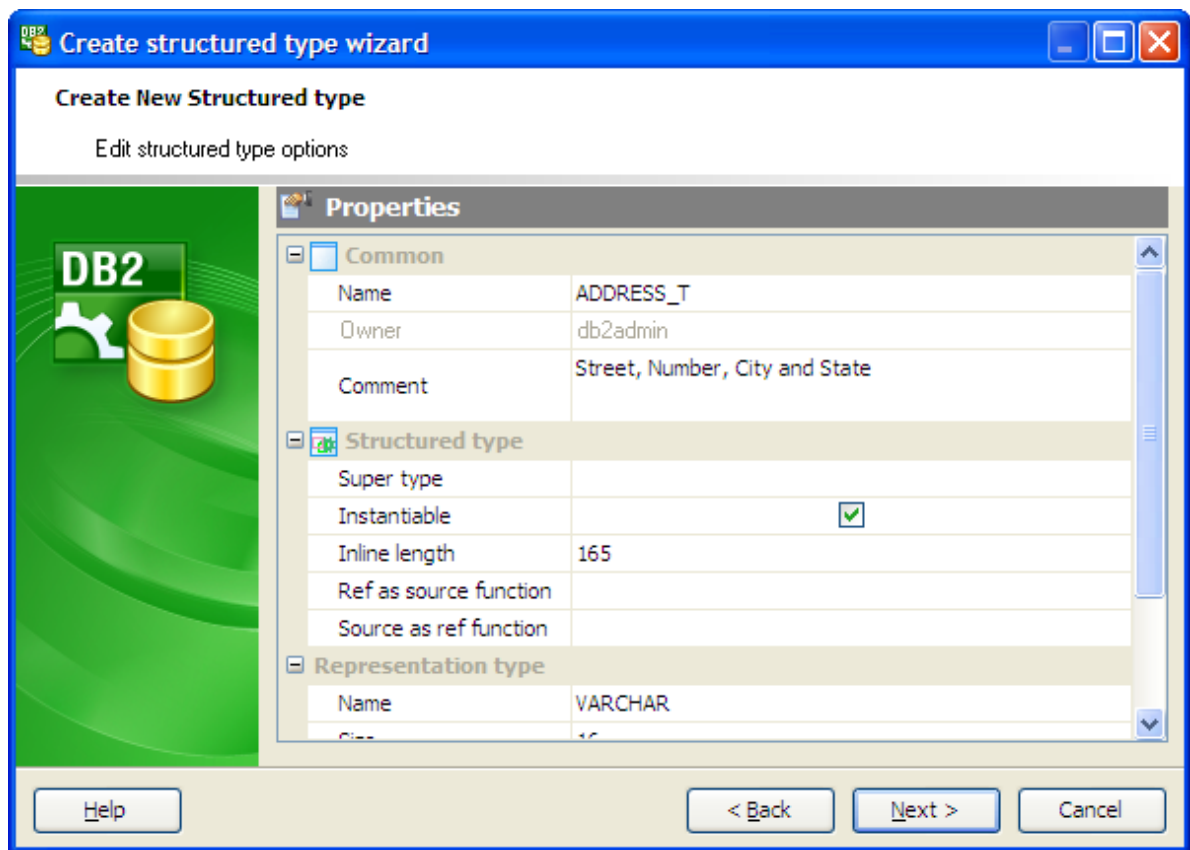
## 5.10.1 Create Structured Type Wizard

[Create Structured Type Wizard](#) guides you through the process of creating a new user-defined type. See [How To Create Structured Type](#)<sup>[141]</sup> for instructions on running this wizard.

The basic principles of Create Object Wizards in DB2 Maestro are covered by the corresponding [topic](#)<sup>[36]</sup>. See below to find the description of wizard steps that are unique to the current object.

- [Specifying type properties](#)<sup>[143]</sup>
- [Setting type subitems](#)<sup>[145]</sup>

**See also:** [Structured Type Editor](#)<sup>[146]</sup>



#### 5.10.1.1 Specifying type properties

The wizard step was supplied to define common structured type properties. The detailed description of the properties you can find below.

**Create structured type wizard**

**Create New Structured type**

Edit structured type options

**Properties**

<b>Common</b>	
Name	ADDRESS_T
Owner	db2admin
Comment	Street, Number, City and State
<b>Structured type</b>	
Super type	
Instantiable	<input checked="" type="checkbox"/>
Inline length	165
Ref as source function	
Source as ref function	
<b>Representation type</b>	
Name	VARCHAR
Size	16

Help < Back Next > Cancel

#### Name

The field represents the new type name as it was set on the previous wizard step. The name, including the implicit or explicit qualifier, must not identify any other type (built-in, structured, or distinct) already described in the catalog. The unqualified name must not be the same as the name of a built-in data type, *BINARY*, *VARBINARY*, or *BOOLEAN*.

#### Owner

Here you can see the owner of the new structured type. By default, only the owner of an object can perform various operations with the object. In order to allow other users to operate it, privileges must be granted. (However, users that have the superuser attribute can always access any object.)

#### Comment

Use the field to describe the structured type or leave it blank.

#### Super type

You can specify here that this structured type is a subtype under the specified supertype-name. The supertype-name must identify an existing structured type. If supertype-name is specified without a schema name, the type is resolved by searching the schemas on the SQL path. This field is not required.

#### Instantiable

Determines whether an instance of the structured type can be created. Implications of not instantiable structured types are:

- no constructor function is generated for a non-instantiable type

- a non-instantiable type cannot be used as the type of a table or view
- a non-instantiable type can be used as the type of a column (only null values or instances of instantiable subtypes can be inserted into the column).

#### Inline length

This option indicates the maximum size (in bytes) of a structured type column instance to store inline with the rest of the values in the row of a table. Instances of a structured type or its subtypes, that are larger than the specified inline length, are stored separately from the base table row, similar to the way that LOB values are handled.

#### Ref as source function

Defines the name of the system-generated function that casts a reference type value for this structured type to the data type rep-type. A schema name must not be specified as part of the function. The cast function is created in the same schema as the structured type. If the clause is not specified, the default value is rep-type (the name of the representation type).

#### Source as ref function

Defines the name of the system-generated function that casts a value with the data type rep-type to the reference type of this structured type. A schema name must not be specified as part of the function. The cast function is created in the same schema as the structured type. If the clause is not specified, the default value is type-name (the name of the structured type).

#### Representation type properties:

##### Source type name

Specifies the data type used as the basis for the internal representation of the distinct type. The source data type cannot be of type XML.

##### Precision

The first integer is the precision of the number; that is, the total number of digits; it may range from 1 to 31.

##### BIT data

Specifies that the contents of the column are to be treated as bit (binary) data. During data exchange with other systems, code page conversions are not performed. Comparisons are done in binary, irrespective of the database collating sequence.

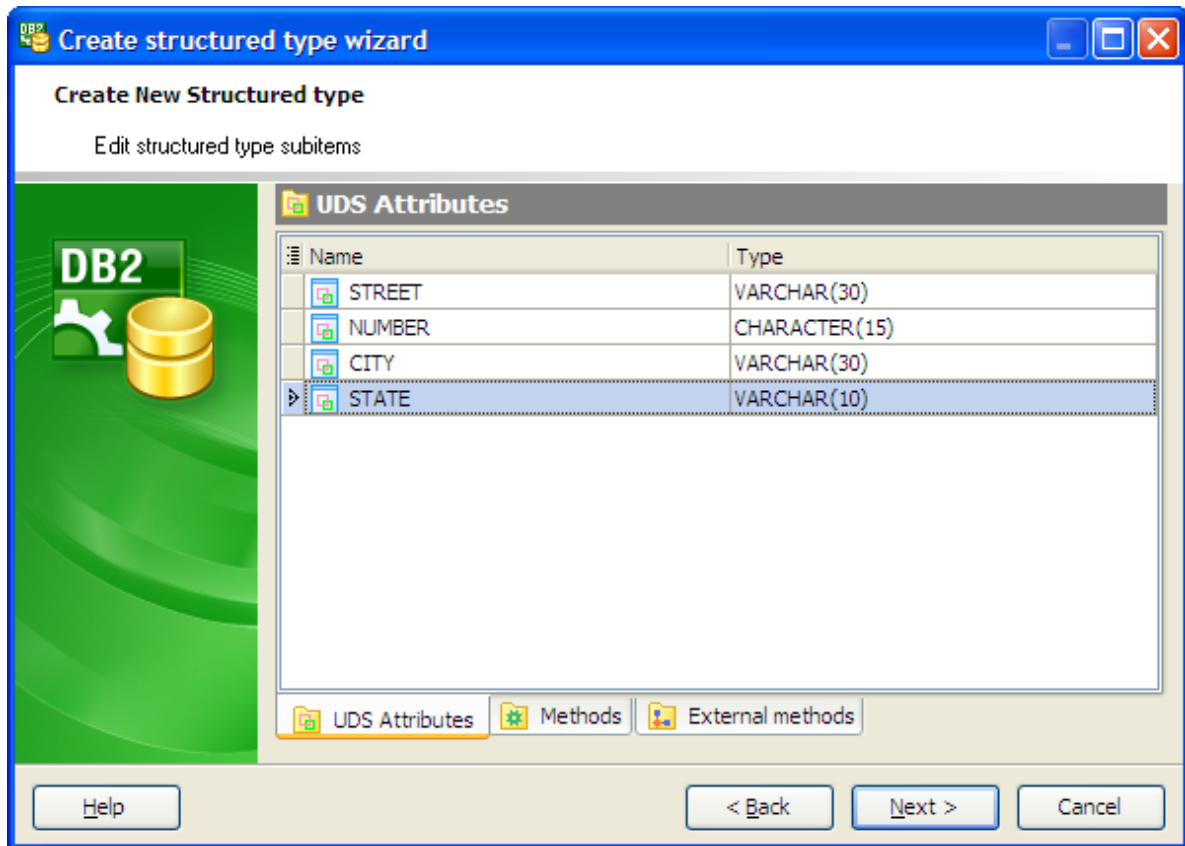
LOBs greater than 1 gigabyte cannot be logged and LOBs greater than 10 megabytes should probably not be logged.

### 5.10.1.2 Setting type subitems

The wizard step allows to define type UDS Attributes, Methods and External methods. Just open the corresponding tab and manage corresponding objects within the popup menu.

For corresponding information see:

- [UDS Attributes](#)<sup>[150]</sup>,
- [Methods](#)<sup>[151]</sup>,
- [Extended methods](#)<sup>[154]</sup>.



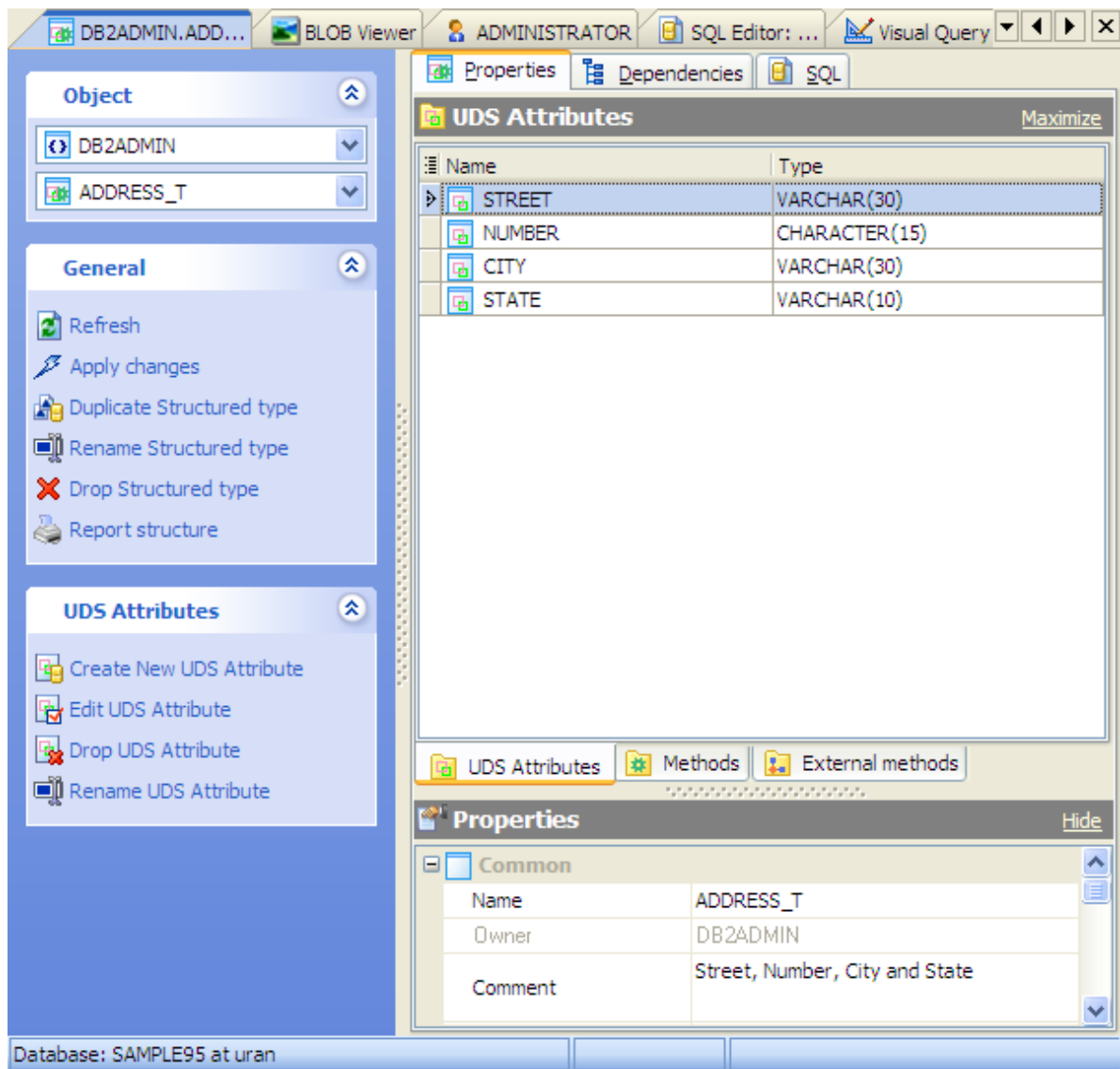
### 5.10.2 Structured Type Editor

[Structured Type Editor](#) can be opened automatically after the type is created and is available on editing (see [Editing Structured Type](#)<sup>[141]</sup> for details).

The basic principles of Object Editors in DB2 Maestro are considered in the corresponding [topic](#)<sup>[39]</sup>. See below to find the description of editor tabs that are unique to the current object.

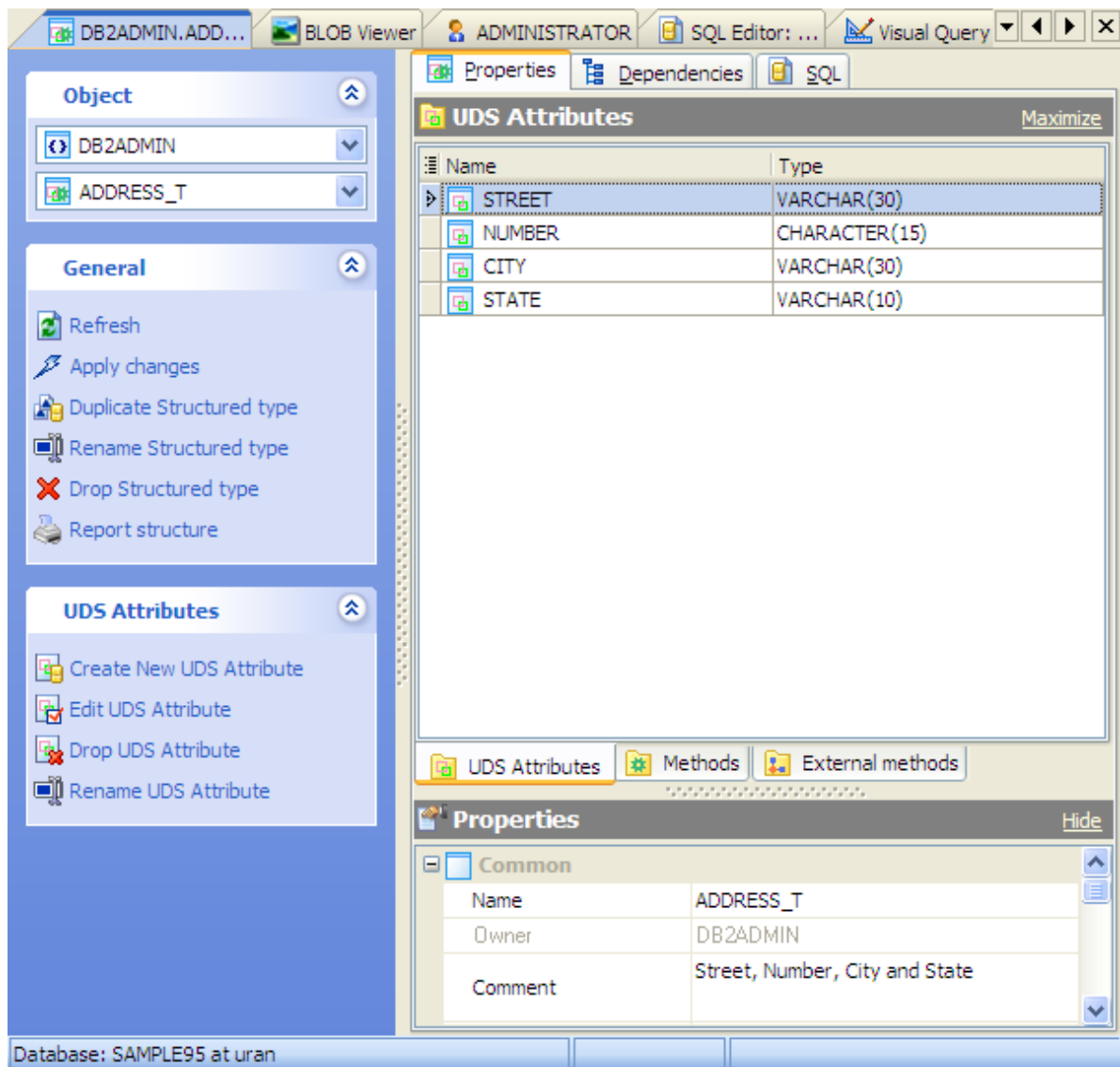
- [Editing type properties](#)<sup>[147]</sup>

**See also:** [Create Structured Type Wizard](#)<sup>[142]</sup>



#### 5.10.2.1 Editing type properties

Public Synonym Editor provides you with an ability to edit synonym properties fast and easy.



### Name

The field represents the new type name as it was set on the previous wizard step. The name, including the implicit or explicit qualifier, must not identify any other type (built-in, structured, or distinct) already described in the catalog. The unqualified name must not be the same as the name of a built-in data type, *BINARY*, *VARBINARY*, or *BOOLEAN*.

### Owner

Here you can see the owner of the new structured type. By default, only the owner of an object can perform various operations with the object. In order to allow other users to operate it, privileges must be granted. (However, users that have the superuser attribute can always access any object.)

### Comment

Use the field to describe the structured type or leave it blank.

### Super type

You can specify here that this structured type is a subtype under the specified supertype-name. The supertype-name must identify an existing structured type. If supertype-name is specified without a schema name, the type is resolved by searching the schemas on the SQL path. This field is not required.

#### Instantiable

Determines whether an instance of the structured type can be created. Implications of not instantiable structured types are:

- no constructor function is generated for a non-instantiable type
- a non-instantiable type cannot be used as the type of a table or view
- a non-instantiable type can be used as the type of a column (only null values or instances of instantiable subtypes can be inserted into the column).

#### Inline length

This option indicates the maximum size (in bytes) of a structured type column instance to store inline with the rest of the values in the row of a table. Instances of a structured type or its subtypes, that are larger than the specified inline length, are stored separately from the base table row, similar to the way that LOB values are handled.

#### Ref as source function

Defines the name of the system-generated function that casts a reference type value for this structured type to the data type rep-type. A schema name must not be specified as part of the function. The cast function is created in the same schema as the structured type. If the clause is not specified, the default value is rep-type (the name of the representation type).

#### Source as ref function

Defines the name of the system-generated function that casts a value with the data type rep-type to the reference type of this structured type. A schema name must not be specified as part of the function. The cast function is created in the same schema as the structured type. If the clause is not specified, the default value is type-name (the name of the structured type).

#### Representation type properties:

##### Source type name

Specifies the data type used as the basis for the internal representation of the distinct type. The source data type cannot be of type XML.

##### Precision

The first integer is the precision of the number; that is, the total number of digits; it may range from 1 to 31.

##### BIT data

Specifies that the contents of the column are to be treated as bit (binary) data. During data exchange with other systems, code page conversions are not performed. Comparisons are done in binary, irrespective of the database collating sequence.

LOBs greater than 1 gigabyte cannot be logged and LOBs greater than 10 megabytes should probably not be logged.

To apply the changes, select the [Apply Changes](#) item in the [Navigation bar](#) or use **Ctrl+F9** or **Ctrl+F7** shortcut keys.

It is also possible to modify object properties without opening the object editor: use the [Object Properties](#) item of the popup menu of the selected object from the explorer tree.

#### 5.10.2.2 UDS Attributes

Set UDS Attributes within the Properties window. Note, the attribute-name cannot be the same as any other attribute of this structured type or any supertype of this structured type.

Properties	
Common	
Name	STREET
UDS Attribute	
Data type	VARCHAR
Size	30
Precision	0
LOB Unit	Bytes
Bit data	<input type="checkbox"/>
Logged	<input type="checkbox"/>
Compact	<input type="checkbox"/>

##### Data type

The data type of the attribute. It is one of the standard data types, other than LONG VARCHAR, LONG VARGRAPHIC, XML, or a distinct type based on LONG VARCHAR or LONG VARGRAPHIC.

Use the [Size](#) edit box to define the length of the field value for integer, float, char and other data types and use Precision to define the precision of the field value, e.g. for float data type.

Precision

##### Precision

The first integer is the precision of the number; that is, the total number of digits; it

may range from 1 to 31.

#### LOB Unit

Specifies the options associated with LOB types.

#### Bit data

Specifies that the contents of the column are to be treated as bit (binary) data. During data exchange with other systems, code page conversions are not performed. Comparisons are done in binary, irrespective of the database collating sequence.

#### Logged

Specifies that changes made to the column are to be written to the log. The data in such columns is then recoverable with database utilities (such as RESTORE DATABASE).

#### Compact

Specifies that the values in the LOB column should take up minimal disk space (free any extra disk pages in the last group used by the LOB value), rather than leave any leftover space at the end of the LOB storage area that might facilitate subsequent append operations. Note that storing data in this way may cause a performance penalty in any append (length-increasing) operations on the column.

### 5.10.2.3 Methods

Use the wizard to define a method of the new structured type.

#### Name

It must be an unqualified SQL identifier. A number of names used as keywords in predicates are reserved for system use, and cannot be used as a method-name. The names are SOME, ANY, ALL, NOT, AND, OR, BETWEEN, NULL, LIKE, EXISTS, IN, UNIQUE, OVERLAPS, SIMILAR, MATCH, and the comparison operators. In general, the same name can be used for more than one method if there is some difference in their signatures.

**Create method wizard**

Create New Method

Edit method options

**Properties**

**Common**

Name	SAMEZIP
Owner	db2admin
Specific name	

**Method**

Data access	Contains SQL
Deterministic	<input checked="" type="checkbox"/>
External action	<input type="checkbox"/>

**Result options**

Data type	INTEGER
Size	0
Precision	0
LOB unit	
Bit data	<input type="checkbox"/>

Help < Back Next > Cancel

#### Data Access

Indicates what type of SQL statements can be executed. Because the SQL statement supported is the RETURN statement, the distinction has to do with whether or not the expression is a subquery.

#### Deterministic

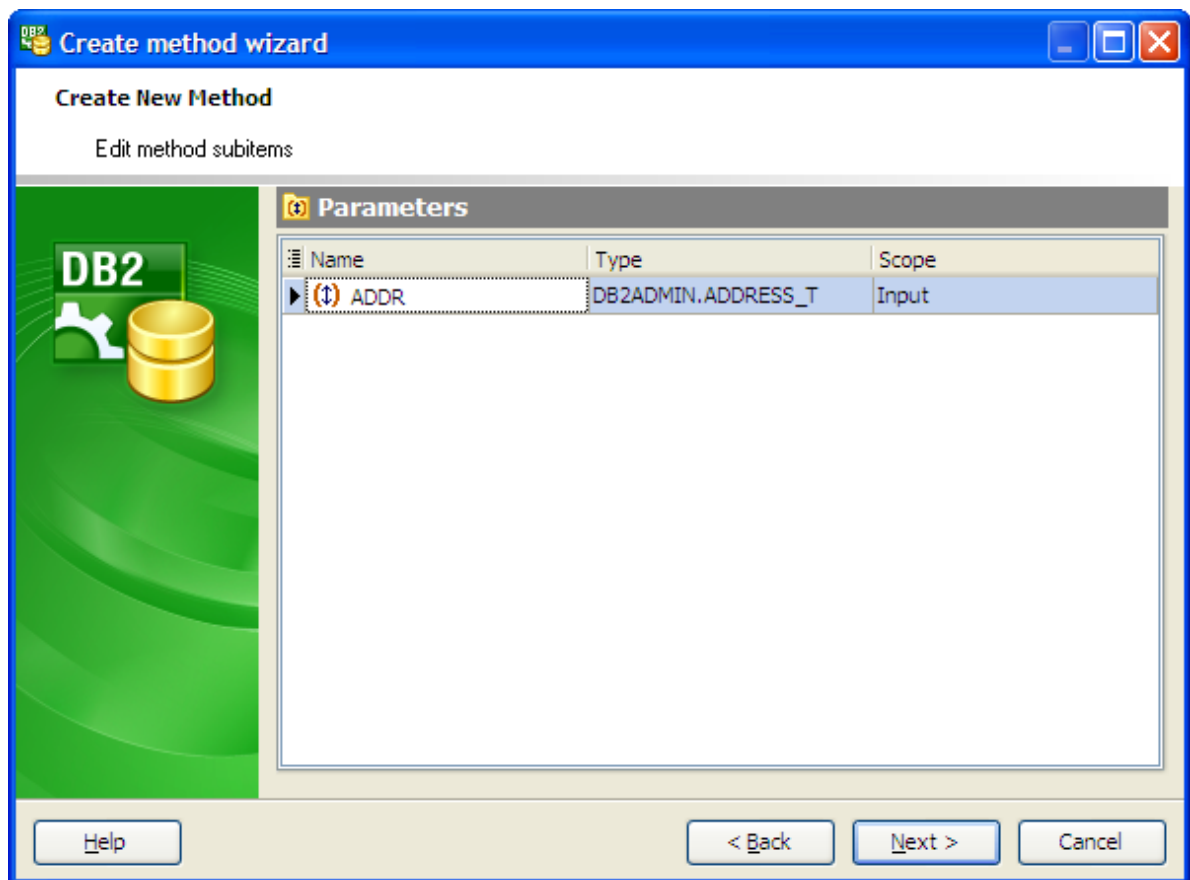
Check the option to specify whether the method always returns the same results for given argument values (ON) or whether the method depends on some state values that affect the results (OFF). That is, a deterministic method must always return the same result from successive invocations with identical inputs.

#### External action

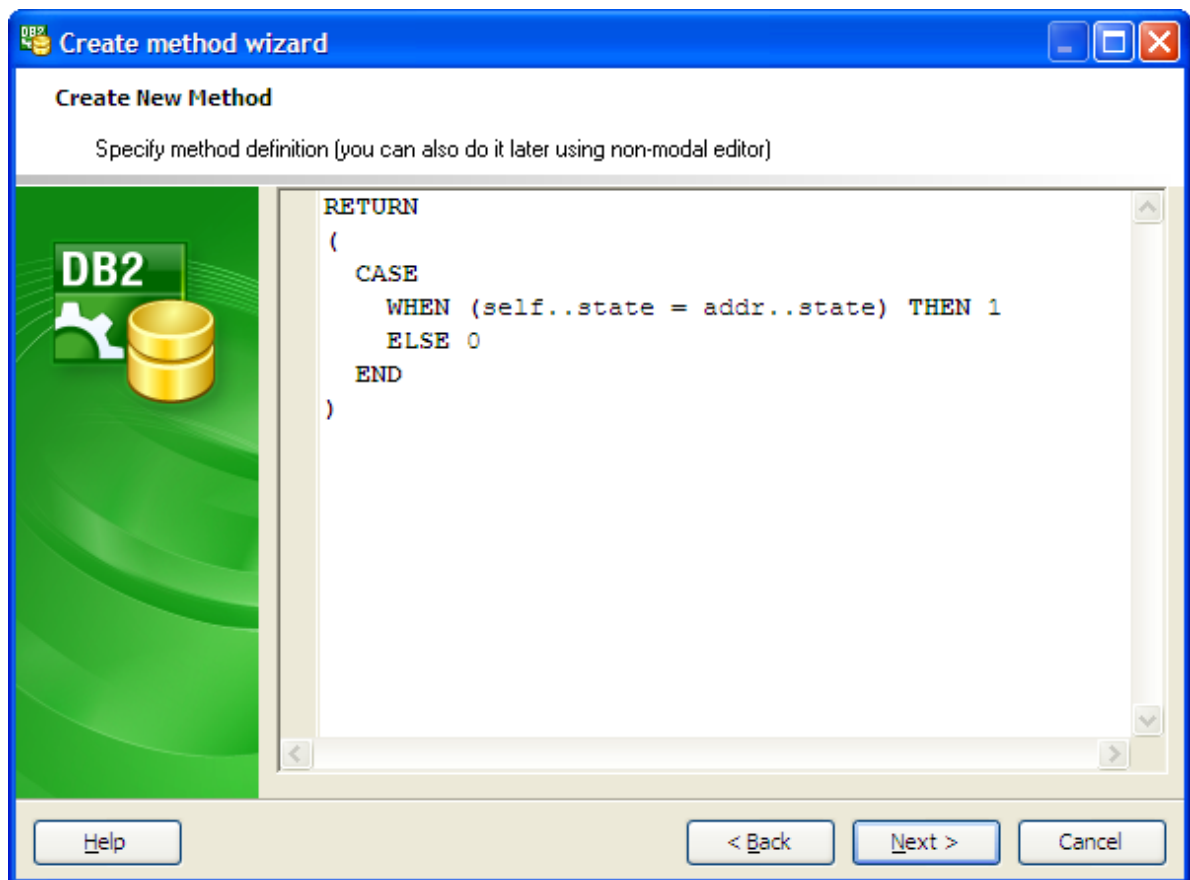
This optional clause specifies whether or not the method takes some action that changes the state of an object not managed by the database manager.

You can also set here method [Result options](#) (Data type, size, Precision, LOB unit, Bit data)

Use popup menu [Add New Parameter...](#) item to add a new parameter and set its properties. Use the [Edit](#) and [Delete](#) items to manage method parameters.

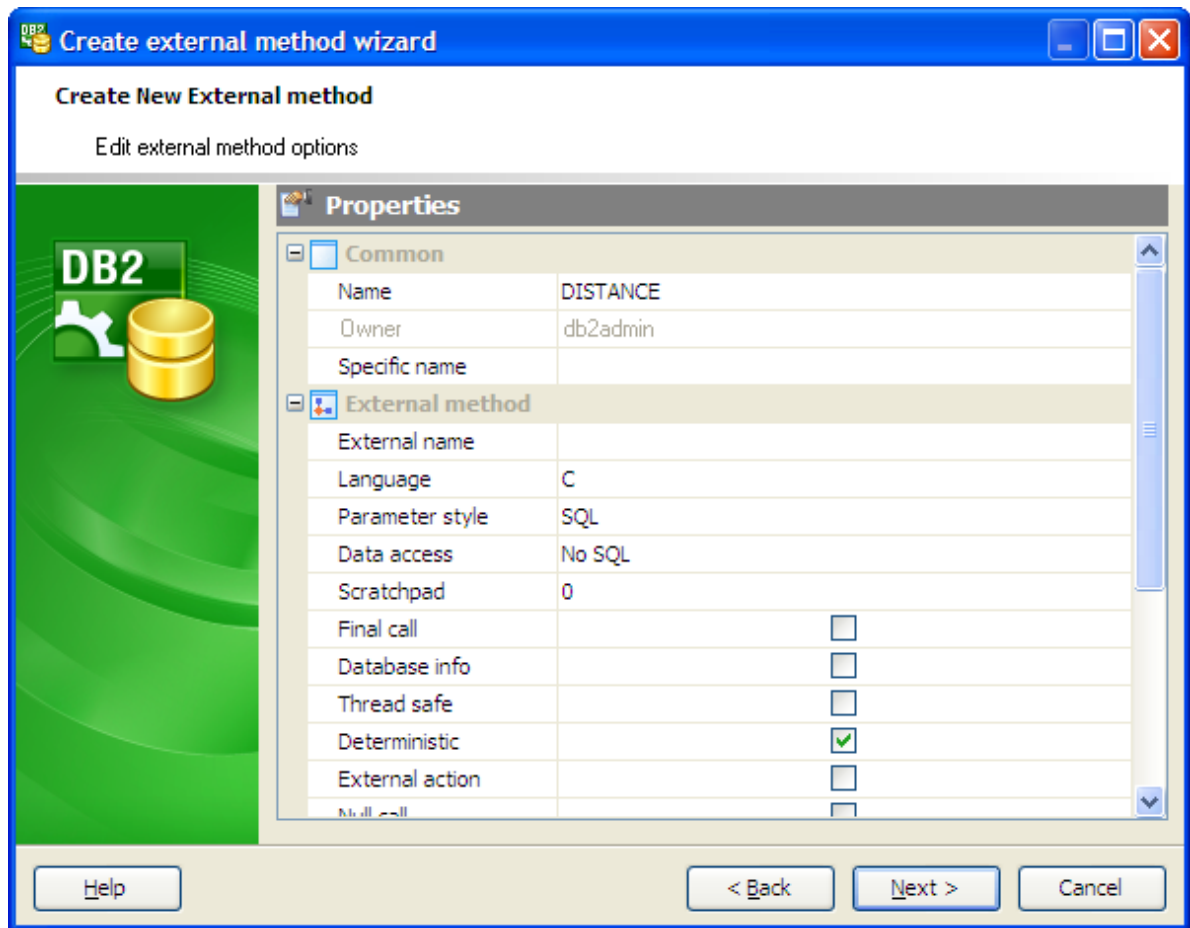


At this step you can specify the SQL definition for the new structured type method. The step is optional: you can do it later using a non-modal editor.



#### 5.10.2.4 Extended methods

Use External method wizard to create a new one. Set the new method **Name** and (if necessary) **Specific name** to provide a unique name for the instance of the method.



### Language

This mandatory clause is used to specify the language interface convention to which the user-defined method body is written.

### Parameter style

This clause is used to specify the conventions used for passing parameters to and returning the value from methods.

### Data access

Indicates whether the method issues any SQL statements and, if so, what type.

### Scratchpad

This optional clause may be used to specify whether a scratchpad is to be provided for an external method. It is strongly recommended that methods be re-entrant, so a scratchpad provides a means for the method to ?save state? from one call to the next.

### Final call

This optional clause specifies whether a final call is to be made to an external method. The purpose of such a final call is to enable the external method to free any system resources it has acquired. It can be useful in conjunction with the SCRATCHPAD keyword in situations where the external method acquires system resources such as memory and anchors them in the scratchpad.

The description of the other External method properties you can find in DB2 SQL Reference Volume 2.

The next wizard step allows you to specify parameters of the new external method. Use popup menu [Add New Parameter...](#) item to add a new parameter and set its properties.

## 5.11 MQ Tables

MQ Table is a materialized query table based on a query.

### ■ How can I create a new materialized query table?

New MQ tables are created within [Create MQ Table Wizard](#)<sup>158</sup>. In order to run the wizard you should either

- select the [Object | Create Database Object...](#) main menu item;
  - select the [MQ Table](#) icon in the [Create Database Object](#) dialog
- or
- select the [MQ Tables](#) list or any object from that list in the explorer tree;
  - select the [Create New MQ Table...](#) item from the popup menu
- or
- open the schema in [Schema Editor](#) and the [MQ Tables](#) tab there;
  - press the **Insert** key or select the [Create New MQ Table...](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#)).

To create a new MQ Table with the same properties as one of the existing MQ Tables has:

- select the [Object | Duplicate Database Object...](#) main menu item;
- follow the instructions of [Duplicate Object Wizard](#).

### ■ How can I edit an existing materialized query table?

MQ Tables can be edited within [MQ Table Editor](#)<sup>164</sup>. In order to run the editor you should either

- select the MQ Table for editing in the explorer tree (type the first letters of the MQ Table name for quick search);
  - select the [Edit MQ Table ...](#) item from the popup menu
- or
- open the schema in [Schema Editor](#) and the [MQ Tables](#) tab there;
  - select the MQ Table to edit;
  - press the **Enter** key or select the [Edit MQ Table](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#)).

You can change the name of the MQ Table using the [Rename MQ Table](#) dialog. To open the dialog you should either

- select the MQ Table to rename in the explorer tree;
  - select the [Rename MQ Table](#) item from the popup menu
- or
- open the schema in [Schema Editor](#) and the [Aliases](#) tab there;
  - select the MQ Table to rename;
  - select the [Rename MQ Table](#) item from the popup menu

(alternatively, you may use the corresponding link of the [Navigation Bar](#)).

#### ■ How can I drop a materialized query table?

To drop a MQ Table:

- select the MQ Table to drop in the explorer tree;
- select the [Drop MQ Table](#) item from the popup menu

or

- open the schema in [Schema Editor](#) and the [MQ Tables](#) tab there;
- select the MQ Table to drop;
- press the **Delete** key or select the [Drop MQ Table](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#))

and confirm dropping in the dialog window.

### 5.11.1 Create MQ Table Wizard

[Create MQ Table Wizard](#) guides you through the process of creating a new materialized-query table. See [How To Create MQ Table](#)<sup>[157]</sup> for instructions on running this wizard.

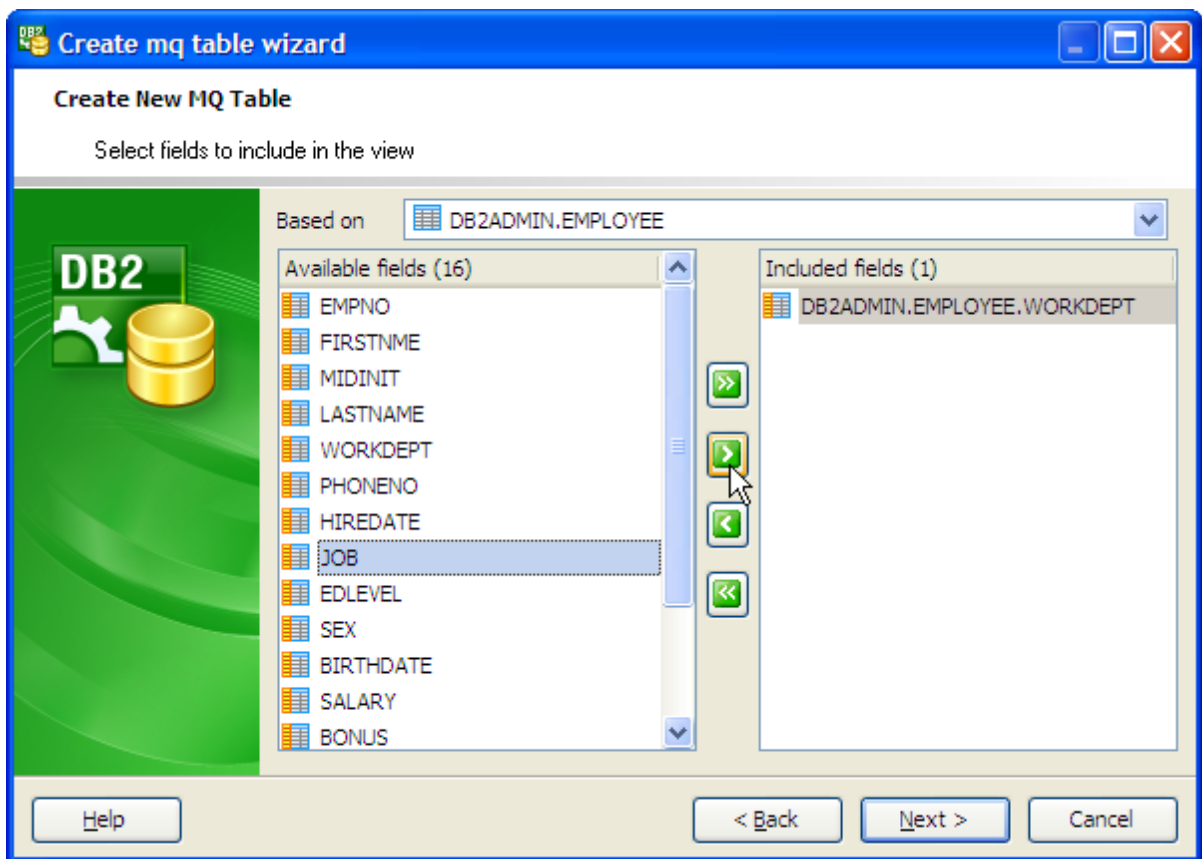
The basic principles of Create Object Wizards in DB2 Maestro are covered by the corresponding [topic](#)<sup>[36]</sup>. See below to find the description of wizard steps that are unique to the current object.

- [Selecting fields](#)<sup>[158]</sup>
- [Specifying table properties](#)<sup>[159]</sup>
- [Specifying the WHERE condition](#)<sup>[161]</sup>

**See also:** [MQ Table Editor](#)<sup>[164]</sup>

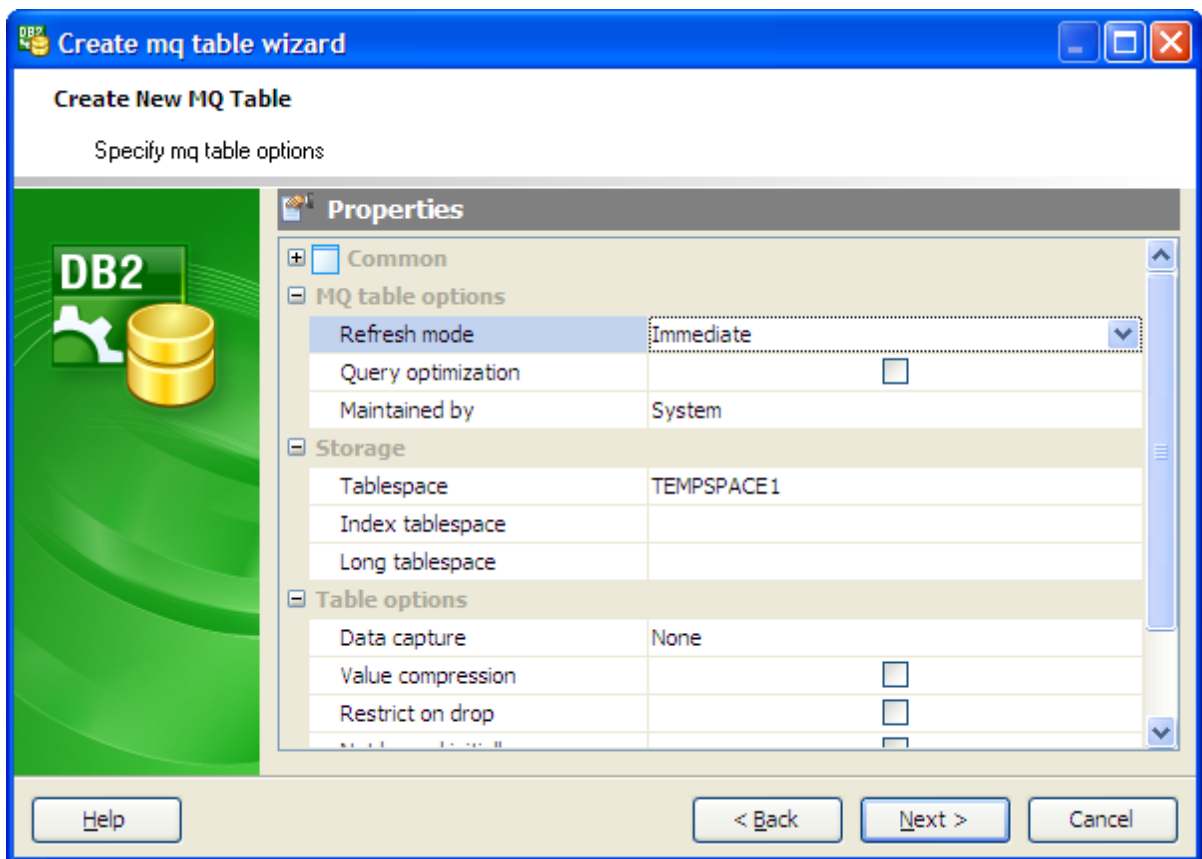
#### 5.11.1.1 Selecting fields

Select a table or a view from the [Based on](#) drop-down menu. Then specify which fields will be used in the new MQ Table. Use [Add All](#) or [Add Selected](#) buttons to include field(s) into view definition. Use the [Remove Selected](#) or [Remove All](#) items to exclude field(s) from the view's field list. Click the [Next](#) button to proceed.



#### 5.11.1.2 Specifying table properties

The wizard step was supplied to define common MQ table properties. The detailed description of the properties you can find below.



#### Name

The field represents the new MQ Table name as it was set on the previous wizard step.

#### Owner

Here you can see the owner of the materialized query table. By default, only the owner of an object can perform various operations with the object. In order to allow other users to operate it, privileges must be granted. (However, users that have the superuser attribute can always access any object.)

#### Refresh mode (Immediate, Deferred, Once)

Indicates how the data in the table is maintained.

#### Query optimization

Check the option to allow the materialized query table can be used for query optimization under appropriate circumstances.

#### Maintained by

Specifies whether the data in the materialized query table is maintained by the system, user, or replication tool. The default is SYSTEM.

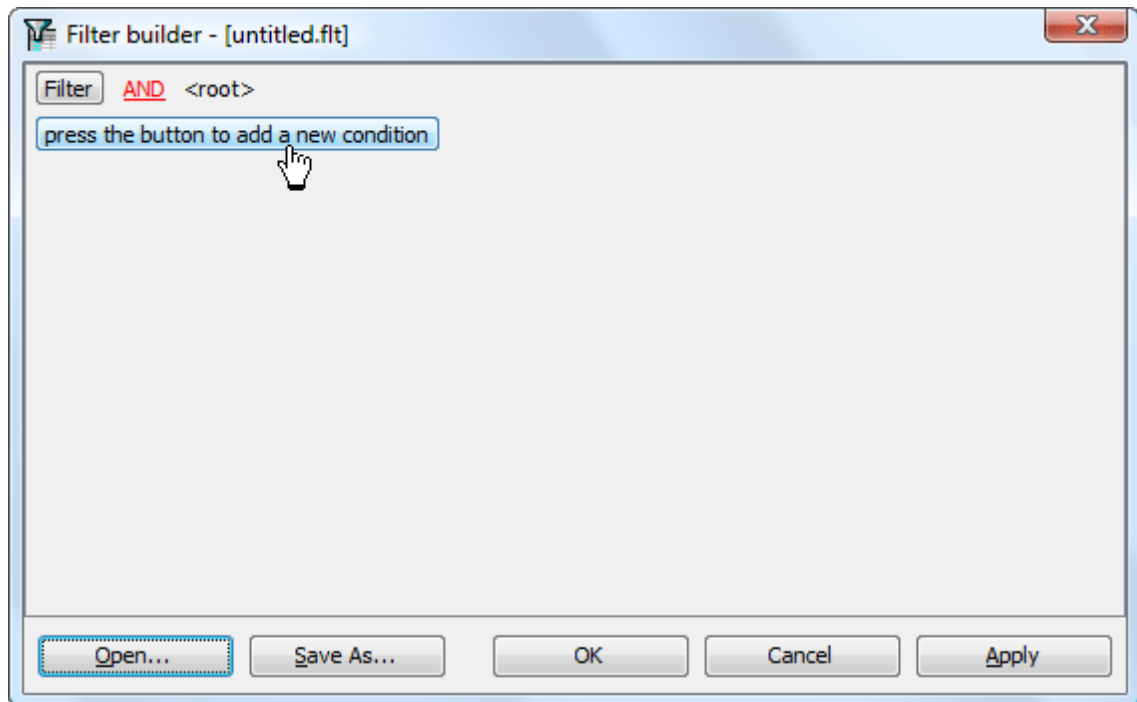
You can specify here also [Storage options](#) and [Table options](#) (data capture, value compression, restrict on drop, etc.) for materialized query table.

The full description of the materialized query table properties you can find in DB2 SQL Reference Volume 2.

### 5.11.1.3 Specifying WHERE condition

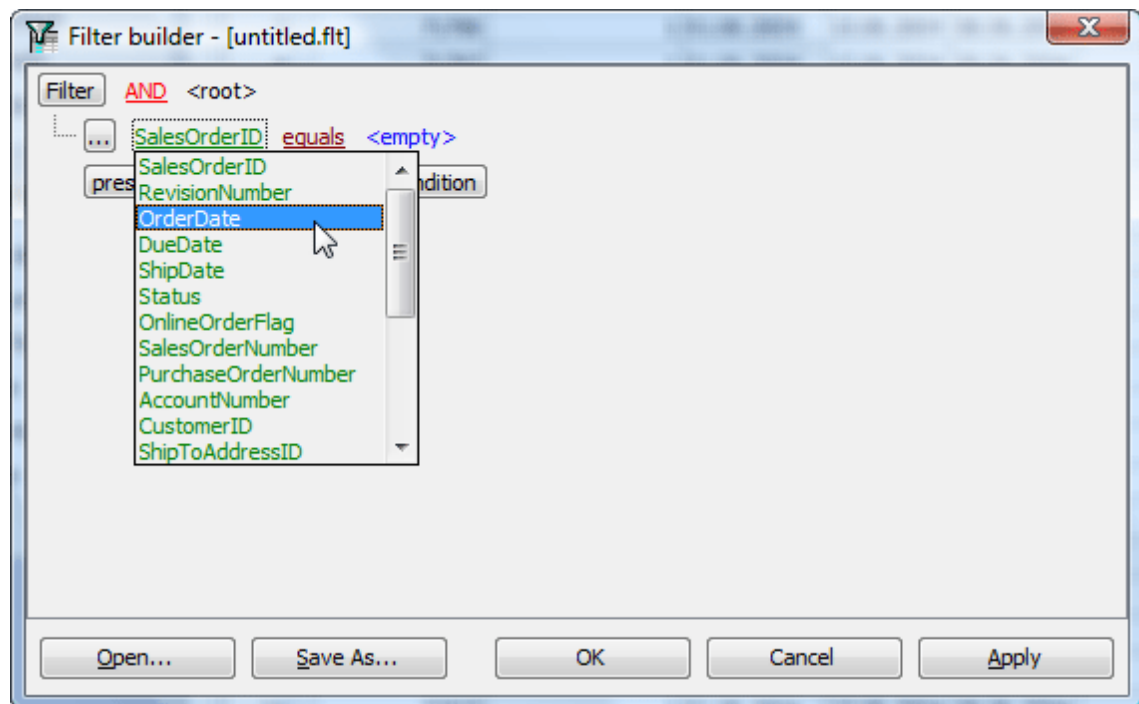
#### ■ Adding a new condition to the filter

Suppose we need to select orders from the sample table *Orders* made between 01.02.2010 and 10.02.2010. These criteria are applied to the *OrderDate* column. Press the button to add this condition. Alternatively, you can use the [Filter](#) button and select the [Add Condition](#) option from the drop-down menu.



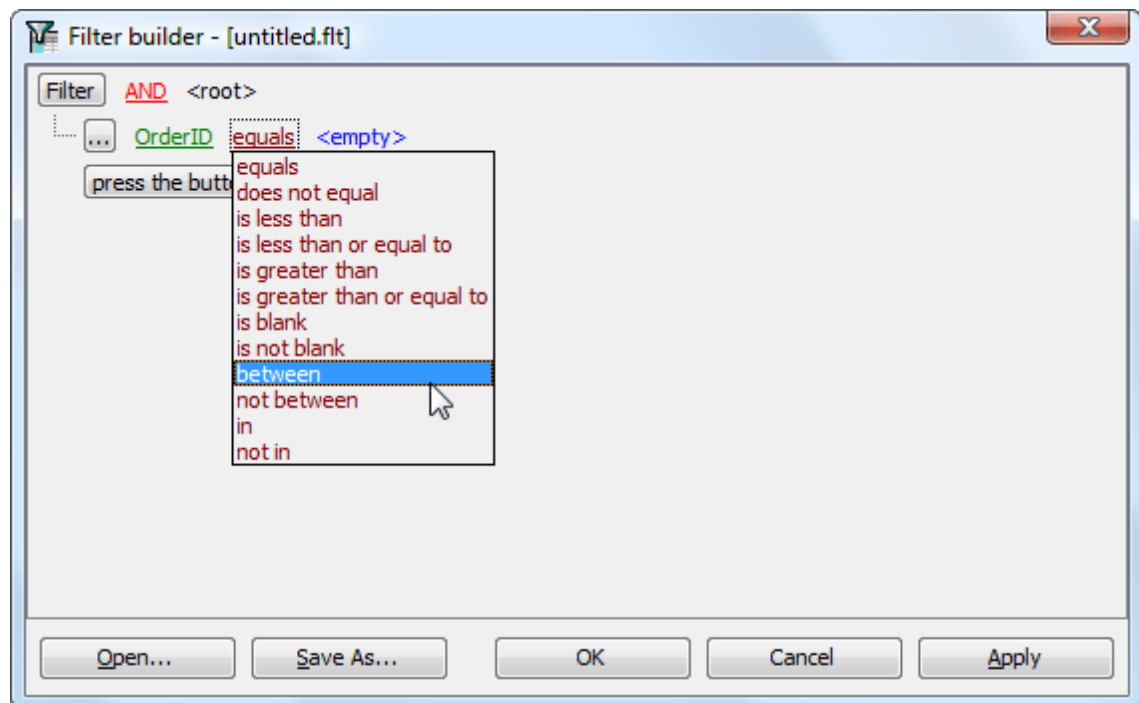
#### ■ Setting a filter criteria in the condition

Select the *OrderDate* column in the drop-down list of the available columns.



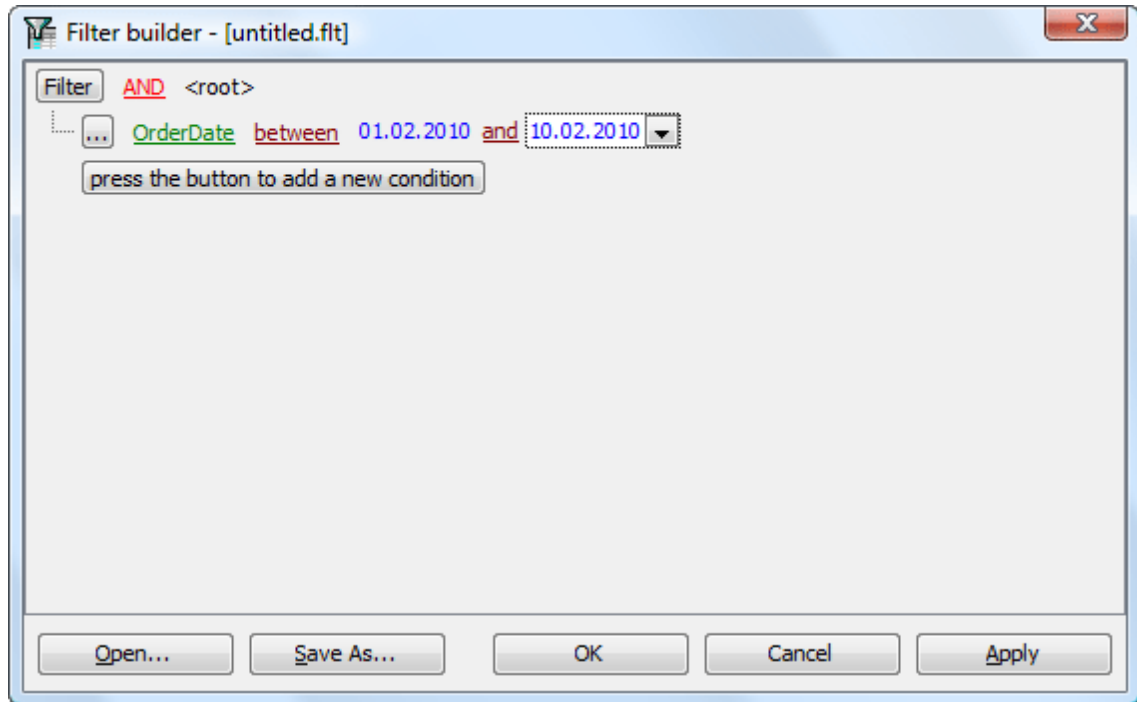
#### ■ Setting an operator in the condition

Set the proper operator. In our example it is BETWEEN.



### ■ Setting criteria values in the condition

Next, you need to specify the range values for the selected operator. The editor used in value boxes is determined by the editor type assigned to the corresponding column.

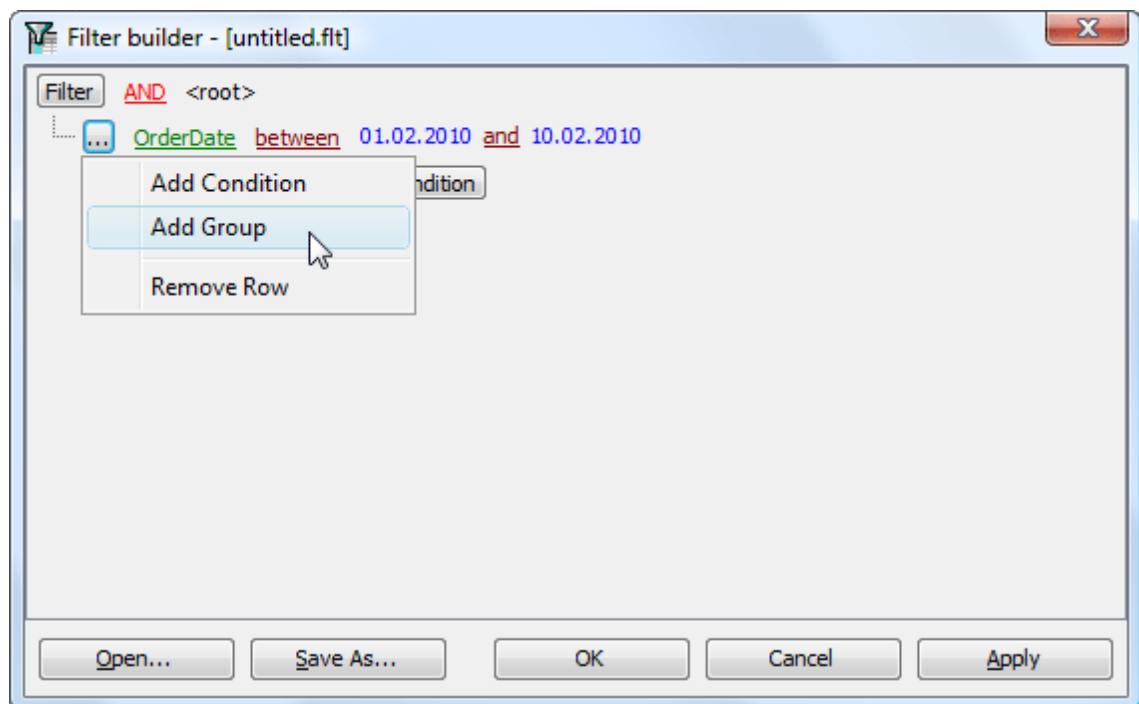


Now use the [Apply](#) button to see the filter result.

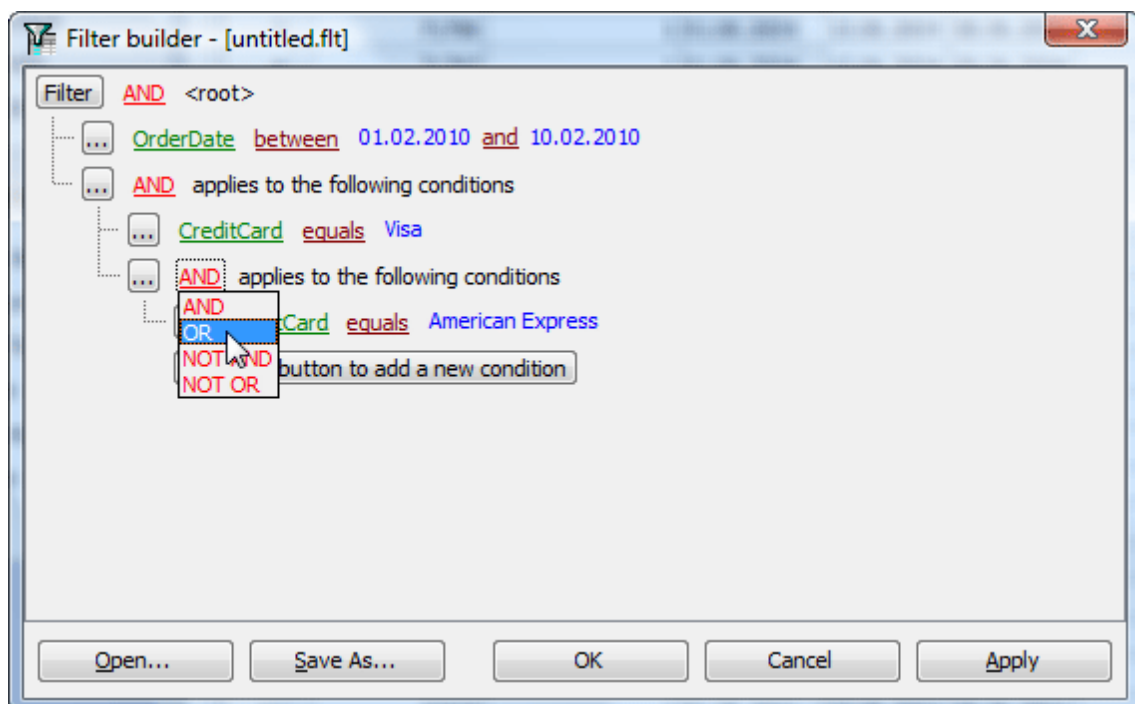
You can add additional conditions to the same root level to be combined by the AND operator.

### ■ Adding a new group

Suppose we need to select orders made between 01.02.2010 and 10.02.2010 and payed via 'Visa' or 'American Express'. This is a complex filter condition combining two simple conditions with the OR operator. Conditions from the same root level are combined by the AND operator. To add a condition combined with the previous one with the OR (NOT AND, NOT OR) operator, use a new group of conditions.



The next screen represents the finished filter conditions for this example.



### 5.11.2 MQ Table Editor

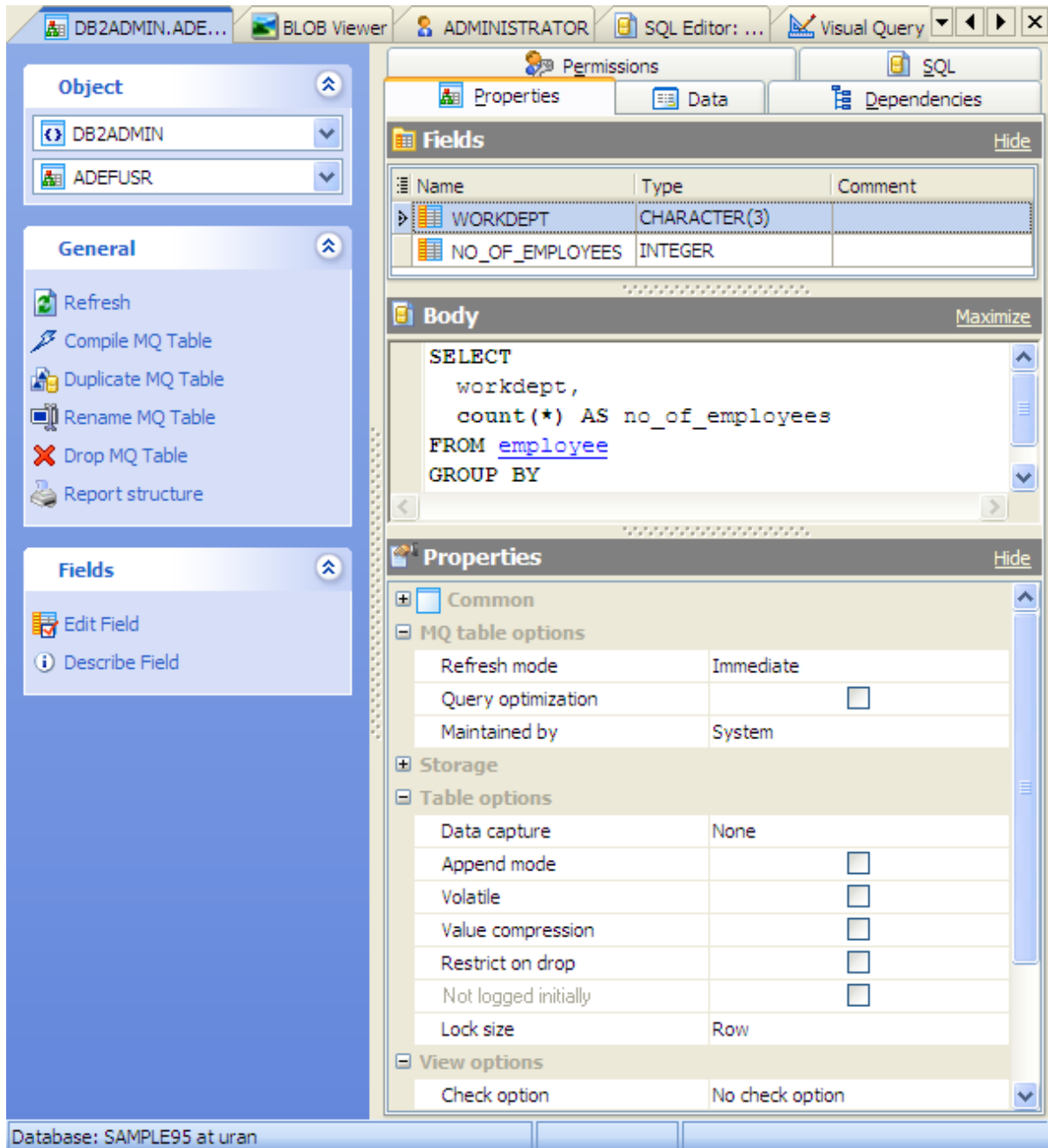
[MQ Table Editor](#) can be opened automatically after the materialized-query table is created and is available on editing (see [Editing MQ Table](#)<sup>[157]</sup> for details).

The basic principles of Object Editors in DB2 Maestro are considered in the corresponding

[topic](#)<sup>[39]</sup>. See below to find the description of editor tabs that are unique to the current object.

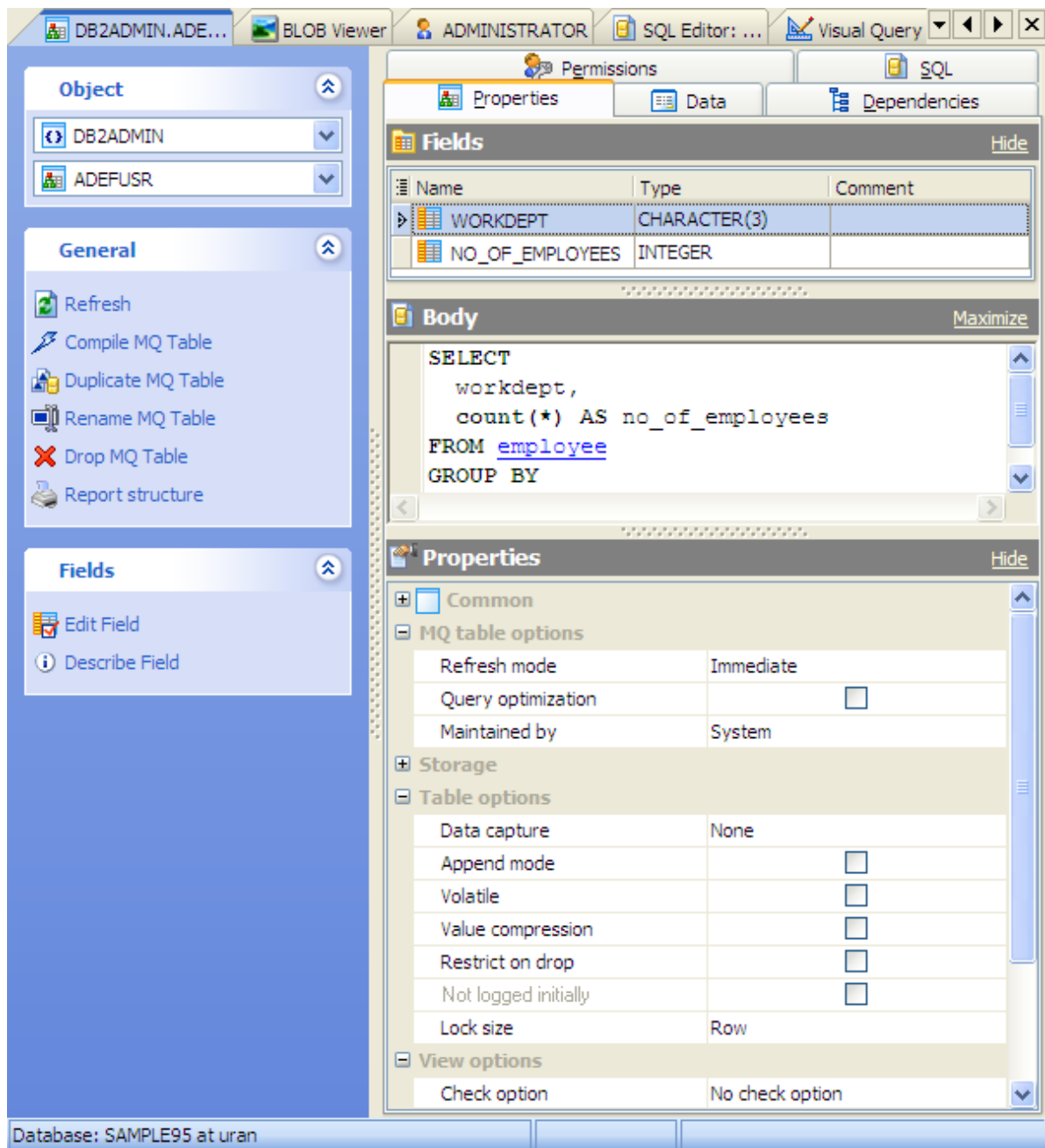
- [Editing table properties](#)<sup>[165]</sup>
- [Viewing table data](#)<sup>[167]</sup>

**See also:** [Create MQ Table Wizard](#)<sup>[158]</sup>



#### 5.11.2.1 Editing table properties

MQ Table Editor provides you with an ability to edit its properties fast and easy.



### Name

The field represents the new MQ Table name as it was set on the previous wizard step.

### Owner

Here you can see the owner of the materialized query table. By default, only the owner of an object can perform various operations with the object. In order to allow other users to operate it, privileges must be granted. (However, users that have the superuser attribute can always access any object.)

### Refresh mode (Immediate, Deferred, Once)

Indicates how the data in the table is maintained.

### Query optimization

Check the option to allow the materialized query table can be used for query optimization under appropriate circumstances.

### Maintained by

Specifies whether the data in the materialized query table is maintained by the system, user, or replication tool. The default is SYSTEM.

You can specify here also [Storage options](#) and [Table options](#) (data capture, value compression, restrict on drop, etc.) for materialized query table.

The full description of the materialized query table properties you can find in DB2 SQL Reference Volume 2.

To apply the changes, select the [Apply Changes](#) item in the [Navigation bar](#) or use **Ctrl+F9** or **Ctrl+F7** shortcut keys.

It is also possible to modify object properties without opening the object editor: use the [Object Properties](#) item of the popup menu of the selected object from the explorer tree.

#### 5.11.2.2 Viewing table data

The [Data](#) tab displays the data represented in the view as a grid (see Data View for details). The popup menu of this tab and the Data Management navigation bar allow you to export data, get SQL dump, set the value of the selected record to *Null* or to *Now* (for [Date](#) values). In tables with BLOB fields you can also call BLOB Editor to view and edit the BLOB fields.

The screenshot displays the SQL Maestro Group interface. The left sidebar contains a tree view with 'Object' (DB2ADMIN, ADEFUSR) and 'General' (Refresh, Compile MQ Table, Duplicate MQ Table, Rename MQ Table, Drop MQ Table, Report structure) sections. The 'Data Management' section includes Generate query, Data input form, Export data, Get SQL dump, Import data, and Print data. The main window shows the 'Data' tab for a table, with a toolbar for navigation and filtering. The table data is as follows:

WORKDEPT	NO_OF_EMPLOYEES
A00	5
B01	1
C01	4
D11	11
D21	7
E01	1
E11	7
E21	6

Records fetched: 8/8

Database: SAMPLE95 at uran

## 6 Server Objects

With DB2 Maestro you can accomplish all the popular actions of server administration. Note that before working with server objects you should connect to any server database first (see [Database Management](#)<sup>[23]</sup>).

The following list contains the most common server objects supported by DB2 Maestro.

- [User](#)<sup>[173]</sup>;
- [Groups](#)<sup>[180]</sup>;
- [Tablespaces](#)<sup>[187]</sup>;
- [Boofer Pools](#)<sup>[192]</sup>.

Below you can find some common ways of server object management.

### ■ Creating of a new server object

New server objects are created within the appropriate [Create Object Wizard](#). In order to run the wizard you should either

- select the corresponding object list (such as ) or any object from that list and then use the [Create New...](#) item from the popup menu

or

- open the server in [Server Editor](#)<sup>[171]</sup> and the necessary objects' tab there and press **Insert** or select the [Create New...](#) item from the popup menu (Alternatively, use the corresponding link of the [Navigation Bar](#)).

### ■ Editing of an existing server object

Server objects are edited within the corresponding [Object Editor](#). In order to open the editor you should either

- select the server object for editing in the explorer tree (type the first letters of the object name for quick search);
- select the [Edit Object](#) item from the popup menu

or

- open the server in [Server Editor](#) and the corresponding objects' tab there;
- select the server object to edit;
- press the **Enter** key or select the [Edit Object](#) item from the popup menu (alternatively, you can use the corresponding link of the [Navigation Bar](#)).

### ■ Dropping of a server object

To drop the existing server object:

- select the server object to drop in the explorer tree;

- select the [Drop Object](#) item from the popup menu
- or
- open the server in [Server Editor](#) and the appropriate objects' tab there;
  - select the server object to drop;
  - press the **Delete** key or select the [Drop Object](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#))

and confirm dropping in the dialog window.

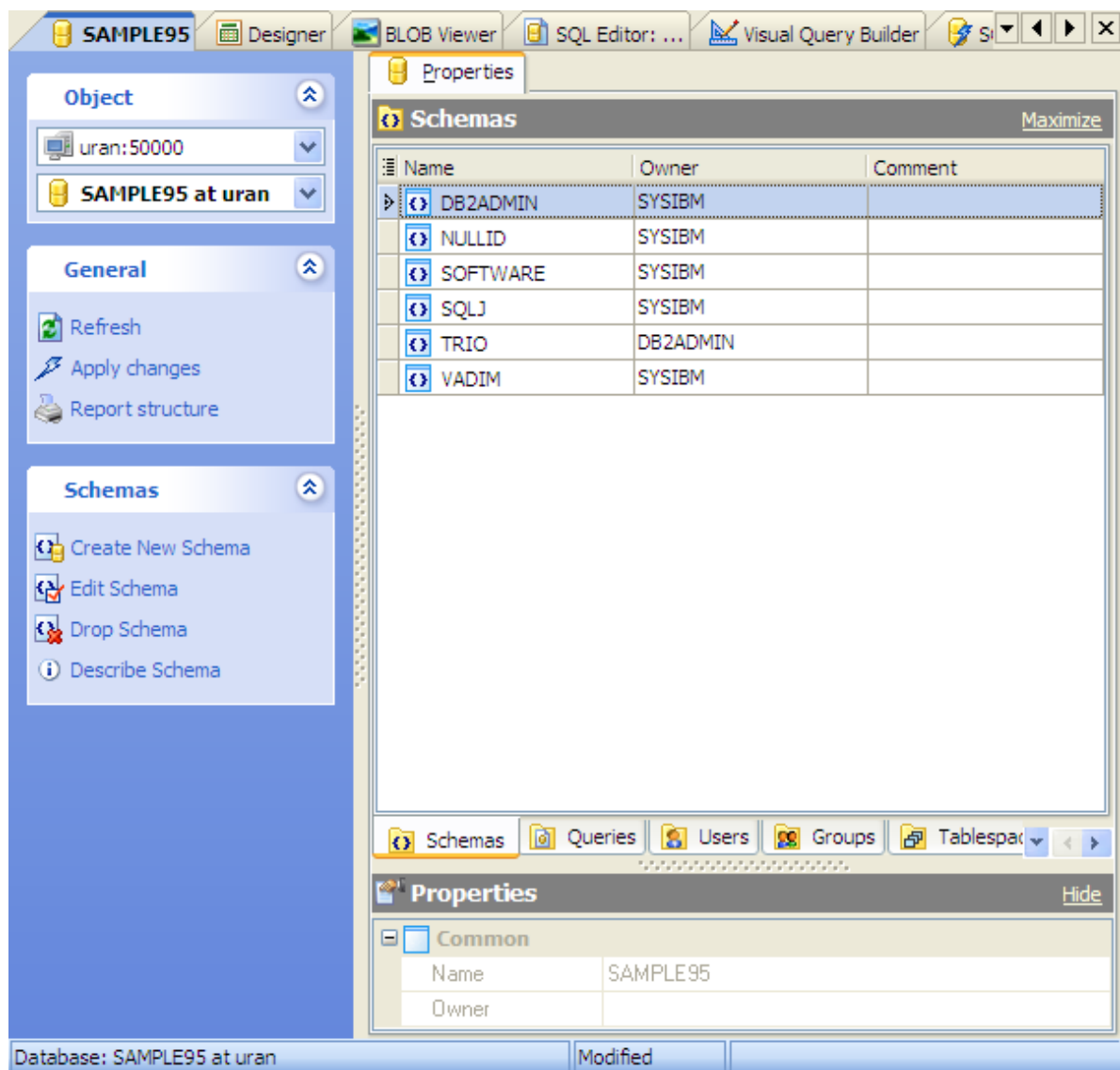
## 6.1 Server Editor

Server Editor allows you to look through all the server objects. Only connected databases are available for examination.

To run [Server Editor](#), select the essential server from the [Explorer Tree](#) and press **Enter**, or use [popup menu](#).

Server Editor consists of several tabs according different kinds of server-scope objects. Every tab is intended for managing server objects (e.g. *databases*, *users*, *groups*, *tablespaces*, etc.). Any object can be opened in its editor. The popup menu allows you to create new, edit or drop the selected server object. Using the popup menu you can also create a copy of the object.

You can operate on several objects at a time. For this you have to select database objects with the **Shift** or the **Ctrl** key pressed. After the group of objects is selected, you can operate on it, e.g. *delete several objects* at once, as it were a single object.



## 6.2 Users

On operating systems where users and groups exist with the same name, you should specify whether you are granting the privilege to the user or group.

Users (identified by an authorization ID) can successfully execute SQL or XQuery statements only if they have the authority to perform the specified function. To create a table, a user must be authorized to create tables; to alter a table, a user must be authorized to alter the table; and so forth.

### ■ How can I create a new user?

New users are created within [Create User Wizard](#)<sup>[173]</sup>. In order to run the wizard you should open the server in [Server Editor](#) and the [Users](#) tab there and press **Insert** or select the [Create New User](#) item from the popup menu (Alternatively, use the corresponding link of the [Navigation Bar](#)).

### ■ How can I edit an existing user?

Users are edited within [User Editor](#)<sup>[176]</sup>. In order to run the editor you should

- select the user for editing in the explorer tree (type the first letters of the user name for quick search);
- select the [Edit User](#) item from the popup menu.

### ■ How can I drop a user?

To drop the existing user:

- select the user to drop in the explorer tree;
- select the [Drop User](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#))

and confirm dropping in the dialog window.

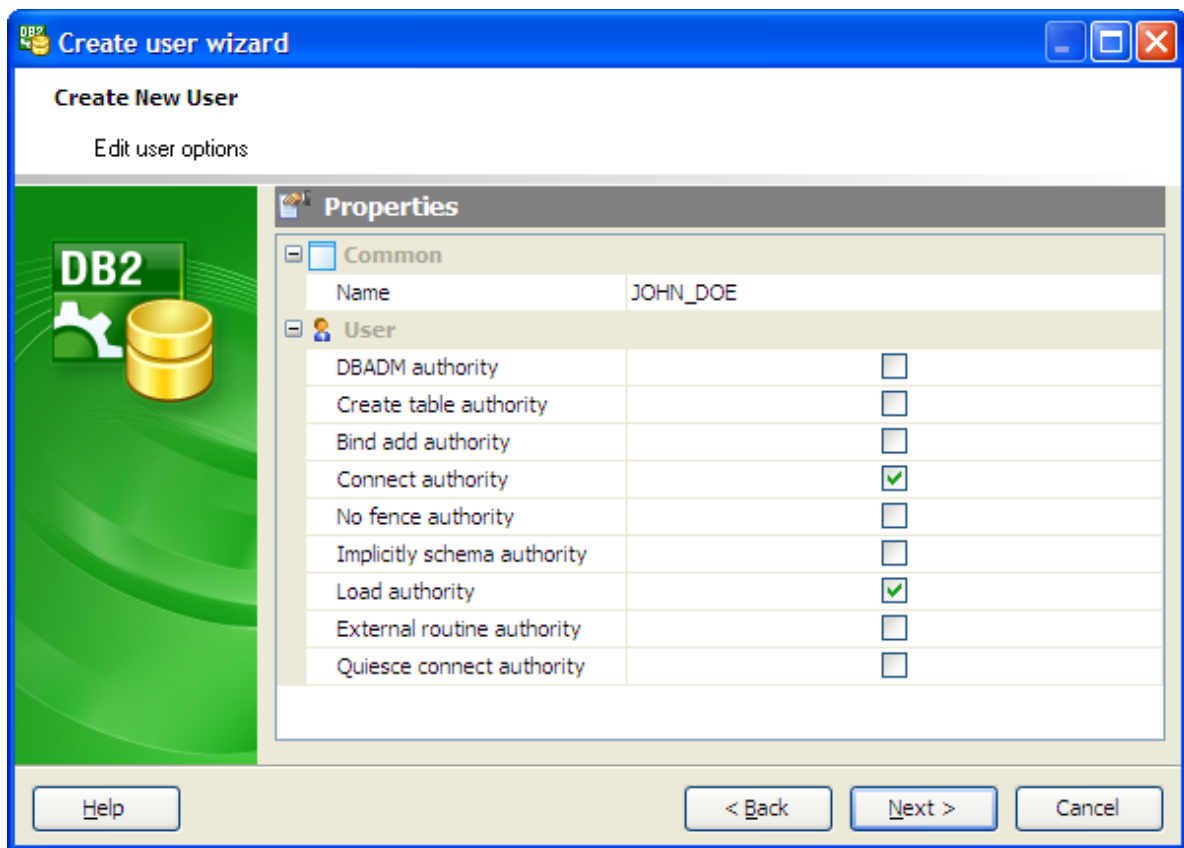
### 6.2.1 Create User Wizard

[Create User Wizard](#) guides you through the process of creating a new user. See [How To Create User](#)<sup>[173]</sup> to learn how to run this wizard.

The basic principles of Create Object Wizards in DB2 Maestro are covered by the corresponding [topic](#)<sup>[36]</sup>. See below to find the description of wizard steps that are unique to the current object.

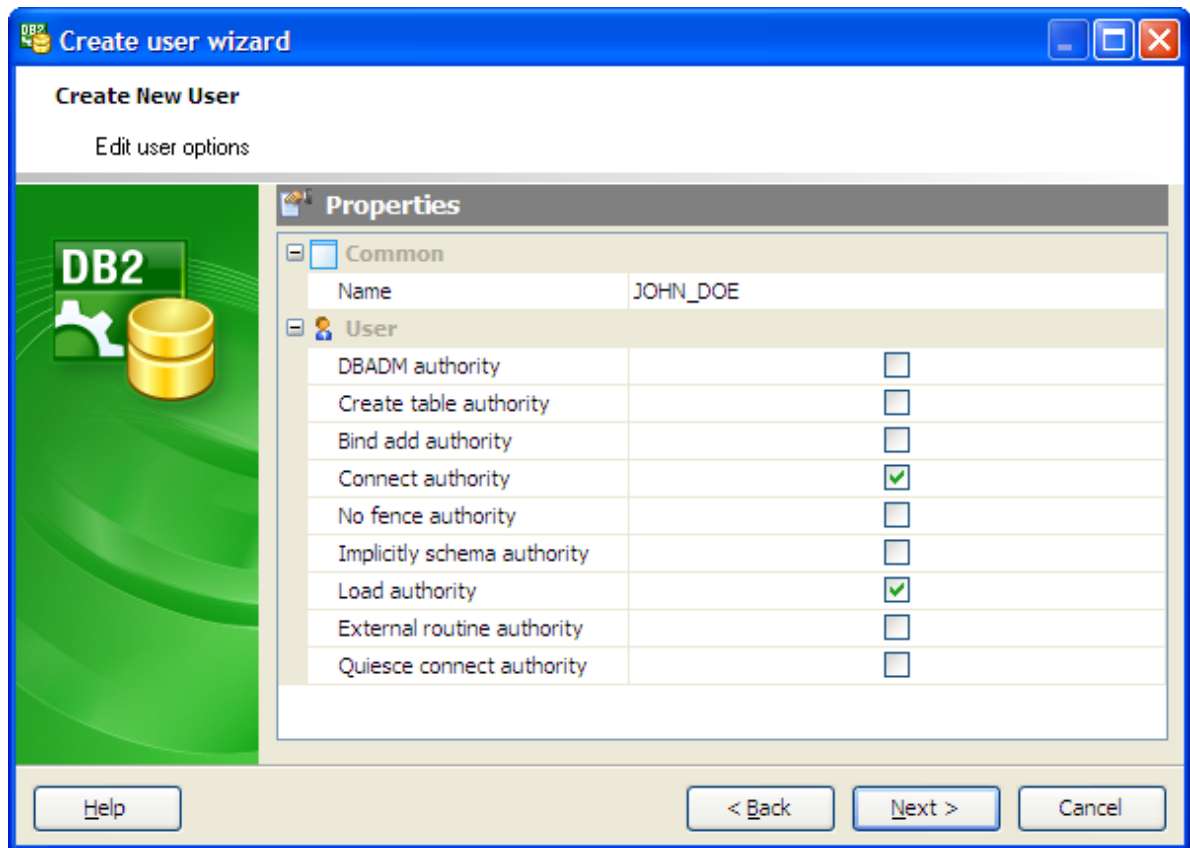
- [Specifying user options](#)<sup>[174]</sup>

**See also:** [User Editor](#)<sup>[176]</sup>



#### 6.2.1.1 Specifying user options

Specify user options according to your needs. The detailed description is given below.



#### DBADM authority

Grants the database administrator authority and all other database authorities except for security administrator authority (SECADM). A database administrator holds nearly all privileges on nearly all objects in the database. The only exceptions are those privileges that are part of the security administrator authority. A database administrator can grant any privilege that is part of database administrator authority to others. All database authorities except for SECADM are implicitly and automatically granted to an authorization-name that is granted [DBADM](#) authority.

#### Create table authority

Grants the authority to create base tables. The creator of a base table automatically has the CONTROL privilege on that table. The creator retains this privilege even if the [Create table authority](#) authority is subsequently revoked.

#### Bind add authority

Grants the authority to create packages. The creator of a package automatically has the CONTROL privilege on that package and retains this privilege even if the [Bind add authority](#) authority is subsequently revoked.

#### Connect authority

Grants the authority to access the database.

#### No fence authority

Grants the authority to register routines that execute in the database manager's process. Care must be taken that routines so registered will not have adverse side effects. (For more information, see the description of the FENCED clause on the CREATE

or ALTER routine statements.) Once a routine has been registered as not fenced, it continues to run in this manner, even if [No fence authority](#) is subsequently revoked. [External routine authority](#) is automatically granted to an authorization-name that is granted [No fence authority](#) authority

#### [Implicitly schema authority](#)

Grants the authority to implicitly create a schema.

#### [Load authority](#)

Grants the authority to load in this database. This authority gives a user the right to use the LOAD utility in this database. SYSADM and [DBADM authority](#) also have this authority by default. However, if a user only has LOAD authority (not SYSADM or [DBADM authority](#)), the user is also required to have table-level privileges. In addition to LOAD privilege, the user is required to have:

- INSERT privilege on the table for LOAD with mode INSERT, TERMINATE (to terminate a previous LOAD INSERT), or RESTART (to restart a previous LOAD INSERT)
- INSERT and DELETE privilege on the table for LOAD with mode REPLACE, TERMINATE (to terminate a previous LOAD REPLACE), or RESTART (to restart a previous LOAD REPLACE)
- INSERT privilege on the exception table, if such a table is used as part of LOAD

#### [External routine authority](#)

Grants the authority to register external routines. Care must be taken that routines so registered will not have adverse side effects.

#### [Quiesce connect authority](#)

Grants the authority to access the database while it is quiesced.

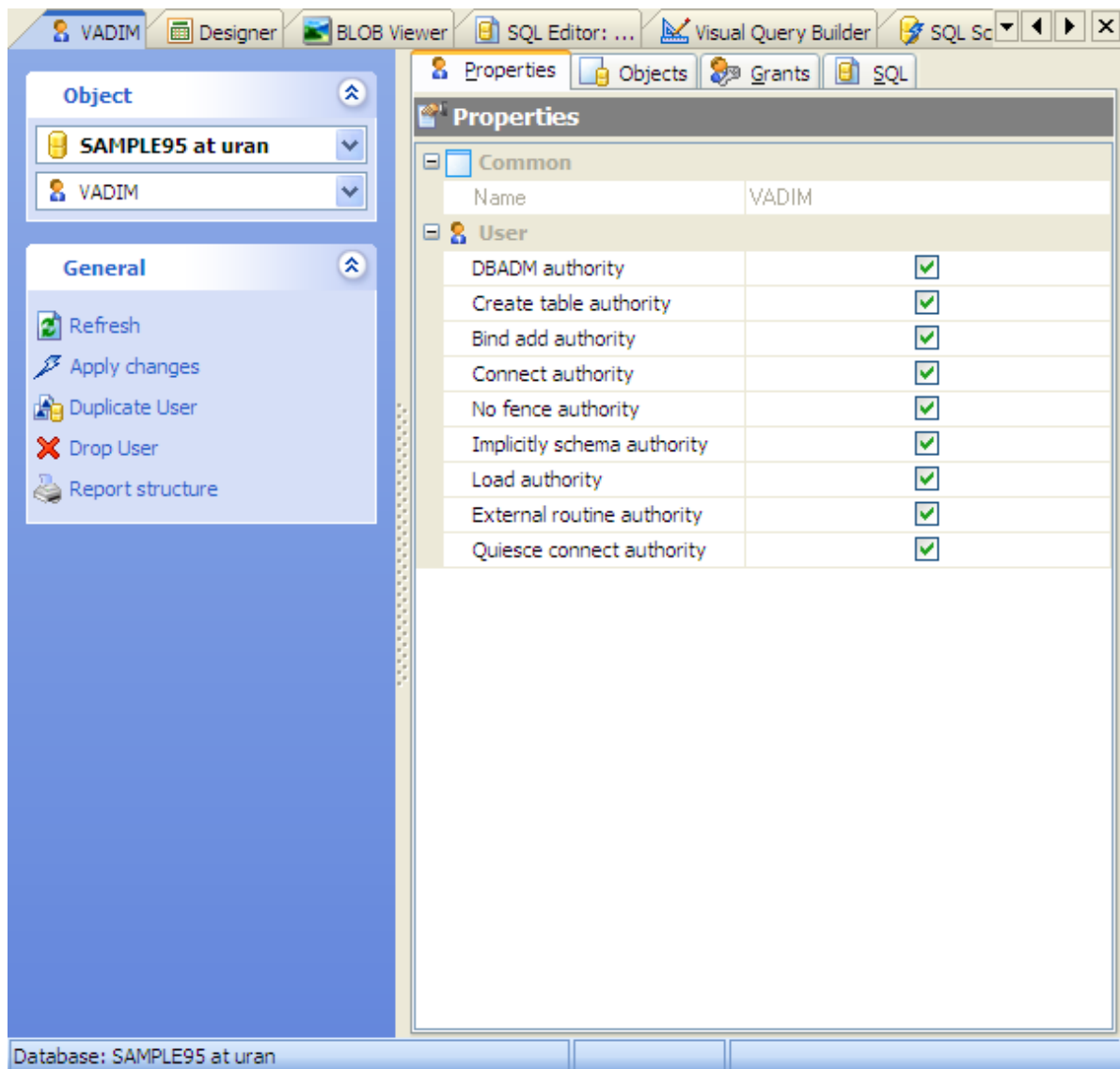
## 6.2.2 User Editor

[User Editor](#) is the basic DB2 Maestro tool for working with existing users. It can be opened automatically after the user is created and is available on editing the user (see [How to edit user](#)<sup>[173]</sup> for details).

The basic principles of Object Editors in DB2 Maestro are considered in the corresponding [topic](#)<sup>[39]</sup>. See below to find the description of editor tabs that are unique to the current object.

- [Editing user properties](#)<sup>[177]</sup>
- [Managing user objects](#)<sup>[179]</sup>

**See also:** [Create User Wizard](#)<sup>[173]</sup>



### 6.2.2.1 Editing user properties

User Editor provides you with an ability to edit user properties. The **Properties** tab allows you to change the user password, the user grants.

#### Name

Here you can view the user name. **DBADM authority**

Grants the database administrator authority and all other database authorities except for security administrator authority (SECADM). A database administrator holds nearly all privileges on nearly all objects in the database. The only exceptions are those privileges that are part of the security administrator authority. A database administrator can grant any privilege that is part of database administrator authority to others. All database authorities except for SECADM are implicitly and automatically granted to an authorization-name that is granted **DBADM** authority.

#### Create table authority

Grants the authority to create base tables. The creator of a base table automatically

has the CONTROL privilege on that table. The creator retains this privilege even if the [Create table authority](#) authority is subsequently revoked.

#### [Bind add authority](#)

Grants the authority to create packages. The creator of a package automatically has the CONTROL privilege on that package and retains this privilege even if the [Bind add authority](#) authority is subsequently revoked.

#### [Connect authority](#)

Grants the authority to access the database.

#### [No fence authority](#)

Grants the authority to register routines that execute in the database manager's process. Care must be taken that routines so registered will not have adverse side effects. (For more information, see the description of the FENCED clause on the CREATE or ALTER routine statements.) Once a routine has been registered as not fenced, it continues to run in this manner, even if [No fence authority](#) is subsequently revoked. [External routine authority](#) is automatically granted to an authorization-name that is granted [No fence authority](#) authority

#### [Implicitly schema authority](#)

Grants the authority to implicitly create a schema.

#### [Load authority](#)

Grants the authority to load in this database. This authority gives a user the right to use the LOAD utility in this database. SYSADM and [DBADM authority](#) also have this authority by default. However, if a user only has LOAD authority (not SYSADM or [DBADM authority](#)), the user is also required to have table-level privileges. In addition to LOAD privilege, the user is required to have:

- INSERT privilege on the table for LOAD with mode INSERT, TERMINATE (to terminate a previous LOAD INSERT), or RESTART (to restart a previous LOAD INSERT)
- INSERT and DELETE privilege on the table for LOAD with mode REPLACE, TERMINATE (to terminate a previous LOAD REPLACE), or RESTART (to restart a previous LOAD REPLACE)
- INSERT privilege on the exception table, if such a table is used as part of LOAD

#### [External routine authority](#)

Grants the authority to register external routines. Care must be taken that routines so registered will not have adverse side effects.

#### [Quiesce connect authority](#)

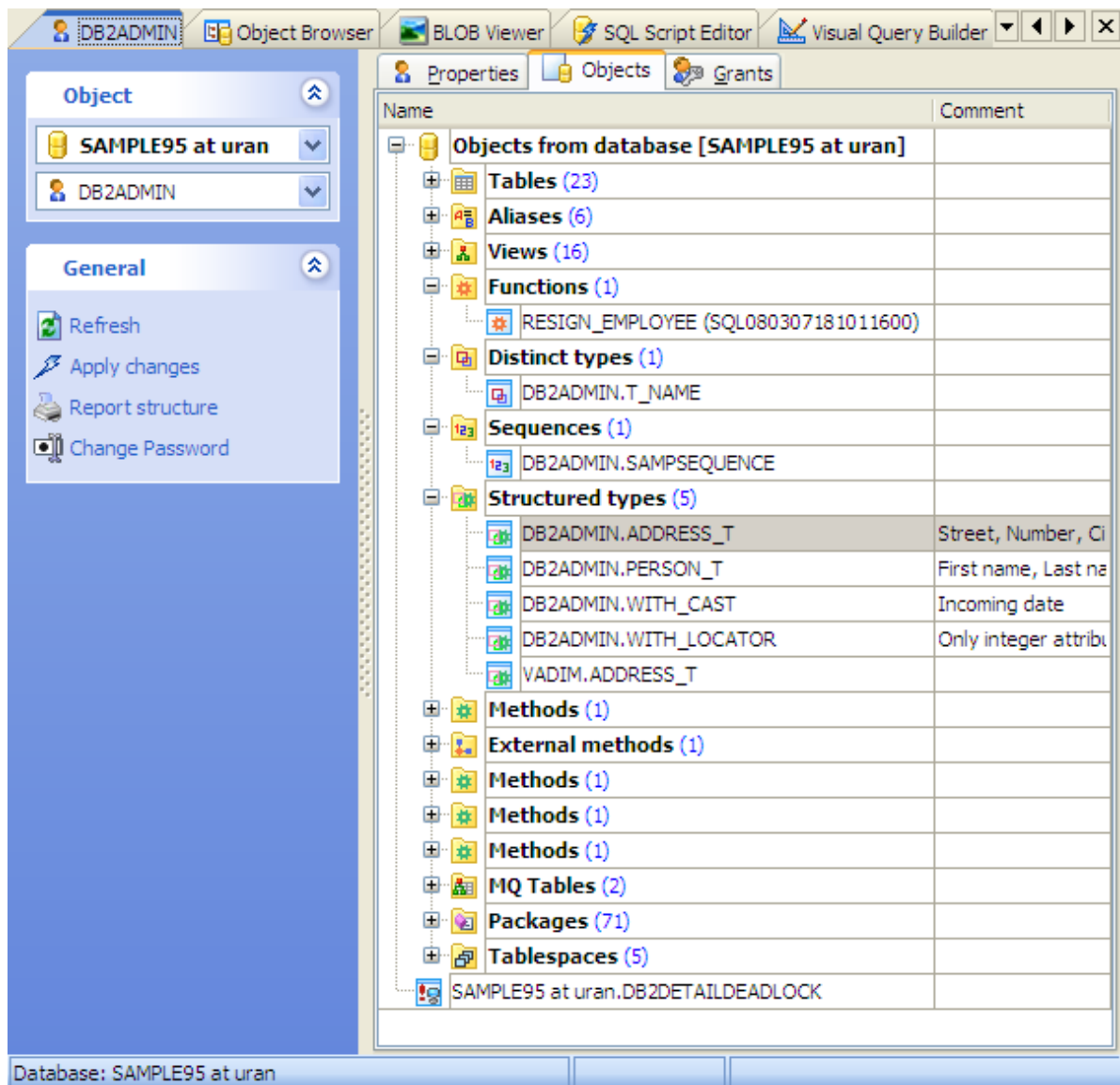
Grants the authority to access the database while it is quiesced.

To apply the changes, select the [Apply Changes](#) item in the [Navigation bar](#) or use **Ctrl+F9** or **Ctrl+F7** shortcut keys.

It is also possible to modify object properties without opening the object editor: use the [Object Properties](#) item of the popup menu of the selected object from the explorer tree.

### 6.2.2.2 Managing user objects

Objects that are having this user as an owner are managed within the **Objects** tab of **User Editor**<sup>[176]</sup>. Double click a row to open the necessary object editor. Use grid's popup menu items to edit, drop or rename object.



## 6.3 Groups

On operating systems where users and groups exist with the same name, you should specify whether you are granting the privilege to the user or group.

### ■ How can I create a new group?

New groups are created within [Create Group Wizard](#)<sup>[180]</sup>. In order to run this wizard you should select the [Groups](#) list or any object from that list and then use the [Create New Group](#) item from the popup menu.

### ■ How can I edit an existing group?

Groups are edited within [Group Editor](#)<sup>[183]</sup>. In order to open the editor you should

- select the group for editing in the explorer tree (type the first letters of the group name for quick search);
- select the [Edit Group](#) item from the popup menu.

### ■ How can I drop a group?

To drop the existing group:

- select the group to drop in the explorer tree;
- select the [Drop Group](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#))

and confirm dropping in the dialog window.

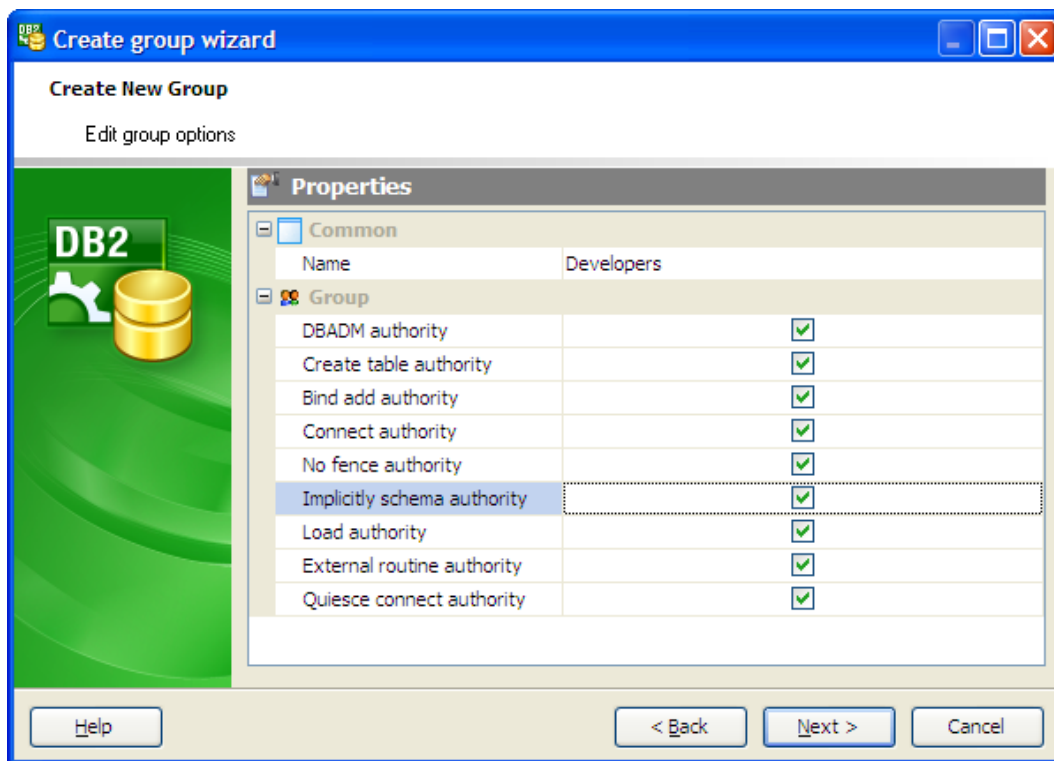
### 6.3.1 Create Group Wizard

[Create Group Wizard](#) guides you through the process of creating a new database group. See [How To Create Group](#)<sup>[180]</sup> to learn how to run this wizard.

The basic principles of Create Object Wizards in DB2 Maestro are covered by the corresponding [topic](#)<sup>[36]</sup>. See below to find the description of wizard steps that are unique to the current object.

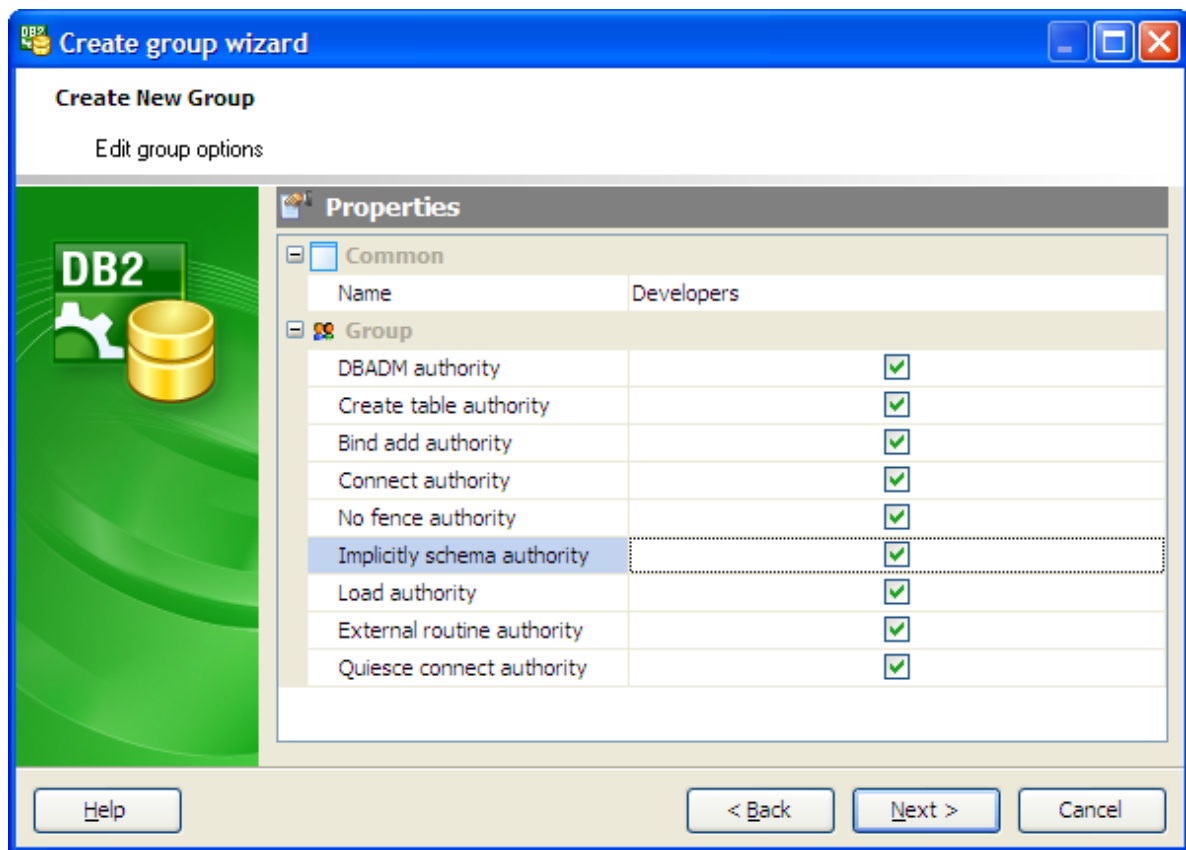
- [Specifying group options](#)<sup>[181]</sup>

**See also:** [Group Editor](#)<sup>[183]</sup>



#### 6.3.1.1 Specifying group options

Specify group options according to your needs. The detailed description is given below.



#### DBADM authority

Grants the database administrator authority and all other database authorities except for security administrator authority (SECADM). A database administrator holds nearly all privileges on nearly all objects in the database. The only exceptions are those privileges that are part of the security administrator authority. A database administrator can grant any privilege that is part of database administrator authority to others. All database authorities except for SECADM are implicitly and automatically granted to an authorization-name that is granted [DBADM](#) authority.

#### Create table authority

Grants the authority to create base tables. The creator of a base table automatically has the CONTROL privilege on that table. The creator retains this privilege even if the [Create table authority](#) authority is subsequently revoked.

#### Bind add authority

Grants the authority to create packages. The creator of a package automatically has the CONTROL privilege on that package and retains this privilege even if the [Bind add authority](#) authority is subsequently revoked.

#### Connect authority

Grants the authority to access the database.

#### No fence authority

Grants the authority to register routines that execute in the database manager's process. Care must be taken that routines so registered will not have adverse side

effects. (For more information, see the description of the `FENCED` clause on the `CREATE` or `ALTER` routine statements.) Once a routine has been registered as not fenced, it continues to run in this manner, even if [No fence authority](#) is subsequently revoked. [External routine authority](#) is automatically granted to an authorization-name that is granted [No fence authority](#) authority

#### [Implicitly schema authority](#)

Grants the authority to implicitly create a schema.

#### [Load authority](#)

Grants the authority to load in this database. This authority gives a user the right to use the `LOAD` utility in this database. `SYSADM` and [DBADM authority](#) also have this authority by default. However, if a user only has `LOAD` authority (not `SYSADM` or [DBADM authority](#)), the user is also required to have table-level privileges. In addition to `LOAD` privilege, the user is required to have:

- `INSERT` privilege on the table for `LOAD` with mode `INSERT`, `TERMINATE` (to terminate a previous `LOAD INSERT`), or `RESTART` (to restart a previous `LOAD INSERT`)
- `INSERT` and `DELETE` privilege on the table for `LOAD` with mode `REPLACE`, `TERMINATE` (to terminate a previous `LOAD REPLACE`), or `RESTART` (to restart a previous `LOAD REPLACE`)
- `INSERT` privilege on the exception table, if such a table is used as part of `LOAD`

#### [External routine authority](#)

Grants the authority to register external routines. Care must be taken that routines so registered will not have adverse side effects.

#### [Quiesce connect authority](#)

Grants the authority to access the database while it is quiesced.

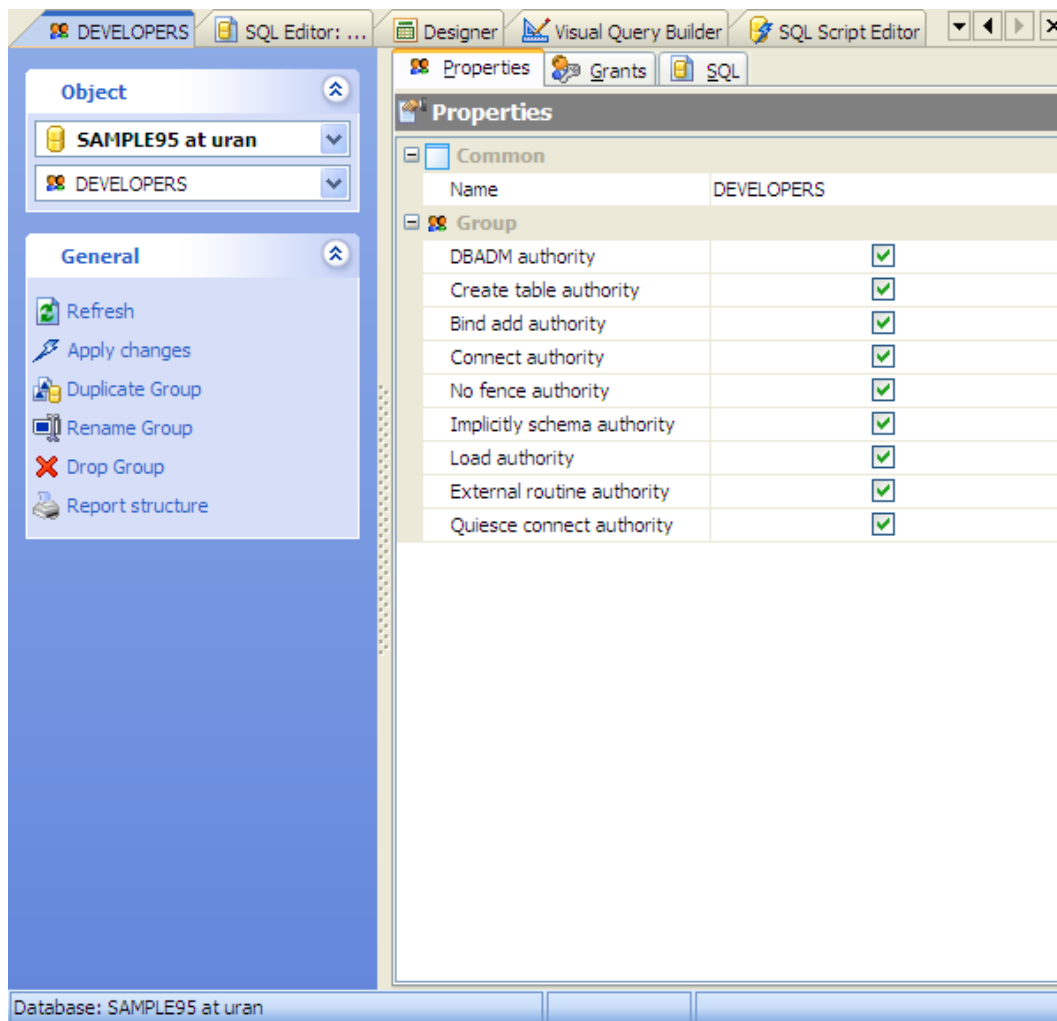
## 6.3.2 Group Editor

[Group Editor](#) is the basic DB2 Maestro tool for working with existing groups. It can be open automatically after the group is created and is available on editing the group (see [How to edit group](#)<sup>[180]</sup> for details).

You can open a group in [Group Editor](#) from the [Explorer Tree](#) or [Object Manager](#).

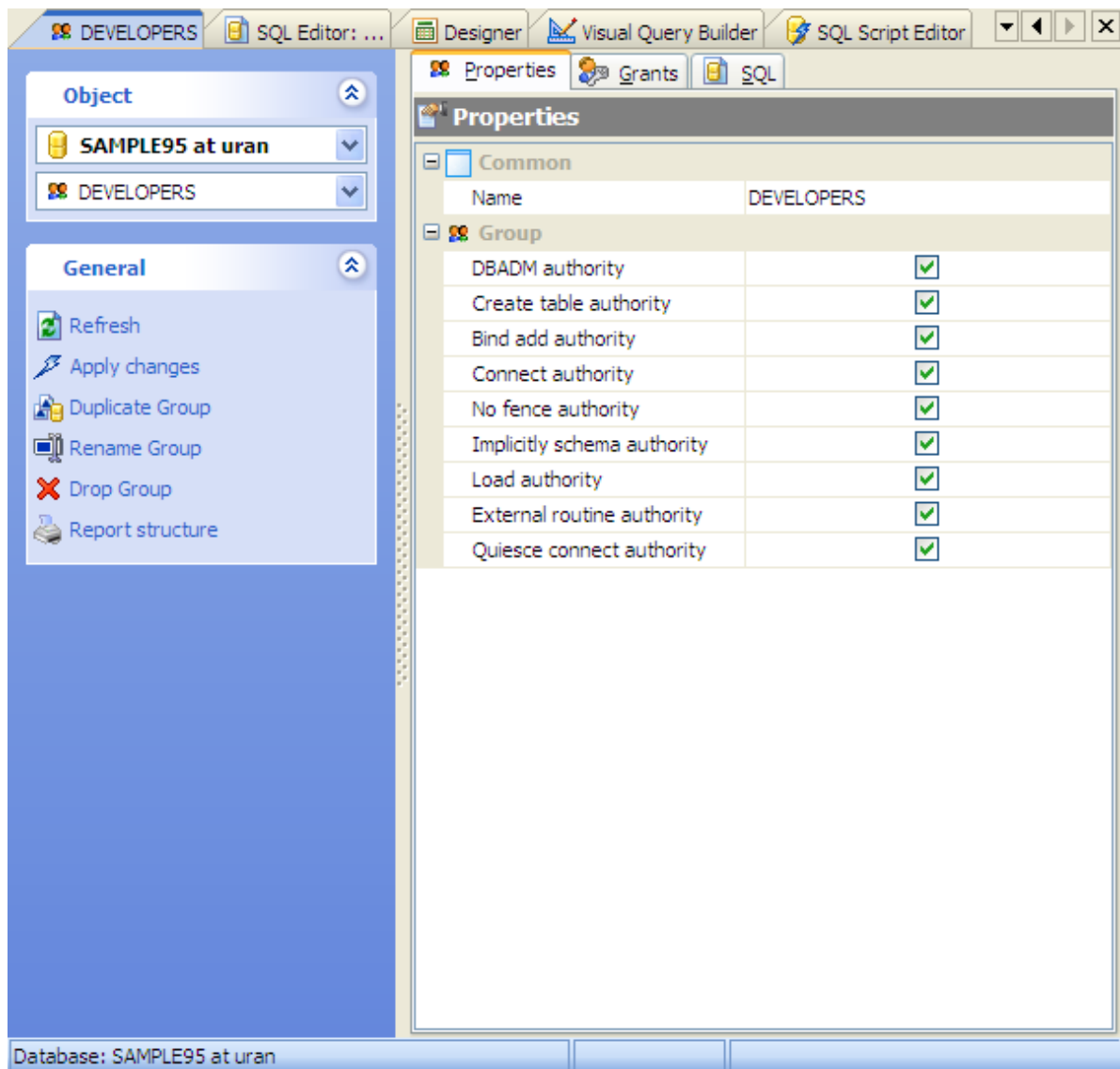
The basic principles of Object Editors in DB2 Maestro are considered in the corresponding [topic](#)<sup>[39]</sup>. See below to find the description of editor tabs that are unique to the current object.

- [Editing group properties](#)<sup>[184]</sup>



#### 6.3.2.1 Editing group properties

Group Editor provides you with an ability to edit group properties. The Properties tab allows you to change the group name, manage the group user.



### Name

Here you can view the group name. [DBADM authority](#)

Grants the database administrator authority and all other database authorities except for security administrator authority (SECADM). A database administrator holds nearly all privileges on nearly all objects in the database. The only exceptions are those privileges that are part of the security administrator authority. A database administrator can grant any privilege that is part of database administrator authority to others. All database authorities except for SECADM are implicitly and automatically granted to an authorization-name that is granted [DBADM authority](#).

### Create table authority

Grants the authority to create base tables. The creator of a base table automatically has the CONTROL privilege on that table. The creator retains this privilege even if the [Create table authority](#) authority is subsequently revoked.

### Bind add authority

Grants the authority to create packages. The creator of a package automatically has the CONTROL privilege on that package and retains this privilege even if the [Bind add authority](#) authority is subsequently revoked.

#### [Connect authority](#)

Grants the authority to access the database.

#### [No fence authority](#)

Grants the authority to register routines that execute in the database manager's process. Care must be taken that routines so registered will not have adverse side effects. (For more information, see the description of the FENCED clause on the CREATE or ALTER routine statements.) Once a routine has been registered as not fenced, it continues to run in this manner, even if [No fence authority](#) is subsequently revoked. [External routine authority](#) is automatically granted to an authorization-name that is granted [No fence authority](#) authority

#### [Implicitly schema authority](#)

Grants the authority to implicitly create a schema.

#### [Load authority](#)

Grants the authority to load in this database. This authority gives a user the right to use the LOAD utility in this database. SYSADM and [DBADM authority](#) also have this authority by default. However, if a user only has LOAD authority (not SYSADM or [DBADM authority](#)), the user is also required to have table-level privileges. In addition to LOAD privilege, the user is required to have:

- INSERT privilege on the table for LOAD with mode INSERT, TERMINATE (to terminate a previous LOAD INSERT), or RESTART (to restart a previous LOAD INSERT)
- INSERT and DELETE privilege on the table for LOAD with mode REPLACE, TERMINATE (to terminate a previous LOAD REPLACE), or RESTART (to restart a previous LOAD REPLACE)
- INSERT privilege on the exception table, if such a table is used as part of LOAD

#### [External routine authority](#)

Grants the authority to register external routines. Care must be taken that routines so registered will not have adverse side effects.

#### [Quiesce connect authority](#)

Grants the authority to access the database while it is quiesced.

To apply the changes, select the [Apply Changes](#) item in the [Navigation bar](#) or use **Ctrl+F9** or **Ctrl+F7** shortcut keys.

It is also possible to modify object properties without opening the object editor: use the [Object Properties](#) item of the popup menu of the selected object from the explorer tree.

## 6.4 Tablespaces

Storage structures contain database objects. The basic storage structure is the [tablespace](#); it contains tables, indexes, large objects, and data defined with a LONG data type.

There are two types of table spaces:

- Database managed space (DMS) A table space that is managed by the database manager.
- System managed space (SMS) A table space that is managed by the operating system.

All table spaces consist of containers. A container describes where objects are stored. A subdirectory in a file system is an example of a container.

### ■ How can I create a new tablespace?

New tablespaces are created within [Create Tablespace Wizard](#)<sup>[188]</sup>. In order to run this wizard you should either

- select the [Tablespaces](#) list or any object from that list and then use the [Create New Tablespace](#) item from the popup menu
- or
- open the server in [Server Editor](#) and the [Tablespaces](#) tab there and press **Insert** or select the [Create New Tablespace](#) item from the popup menu (Alternatively, use the corresponding link of the [Navigation Bar](#)).

### ■ How can I edit an existing tablespace?

Tablespaces are edited within [Tablespace Editor](#)<sup>[189]</sup>. In order to open the editor you should either

- select the tablespace for editing in the explorer tree (type the first letters of the tablespace name for quick search);
  - select the [Edit Tablespace](#) item from the popup menu
- or
- open the server in [Server Editor](#) and the [Tablespaces](#) tab there;
  - select the tablespace to edit;
  - press the **Enter** key or select the [Edit Tablespace](#) item from the popup menu (alternatively, you can use the corresponding link of the [Navigation Bar](#)).

### ■ How can I drop a tablespace?

To drop the existing tablespace:

- select the tablespace to drop in the explorer tree;
  - select the [Drop Tablespace](#) item from the popup menu
- or

- open the server in [Server Editor](#) and the [Tablespaces](#) tab there;
- select the tablespace to drop;
- press the **Delete** key or select the [Drop Tablespace](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#))

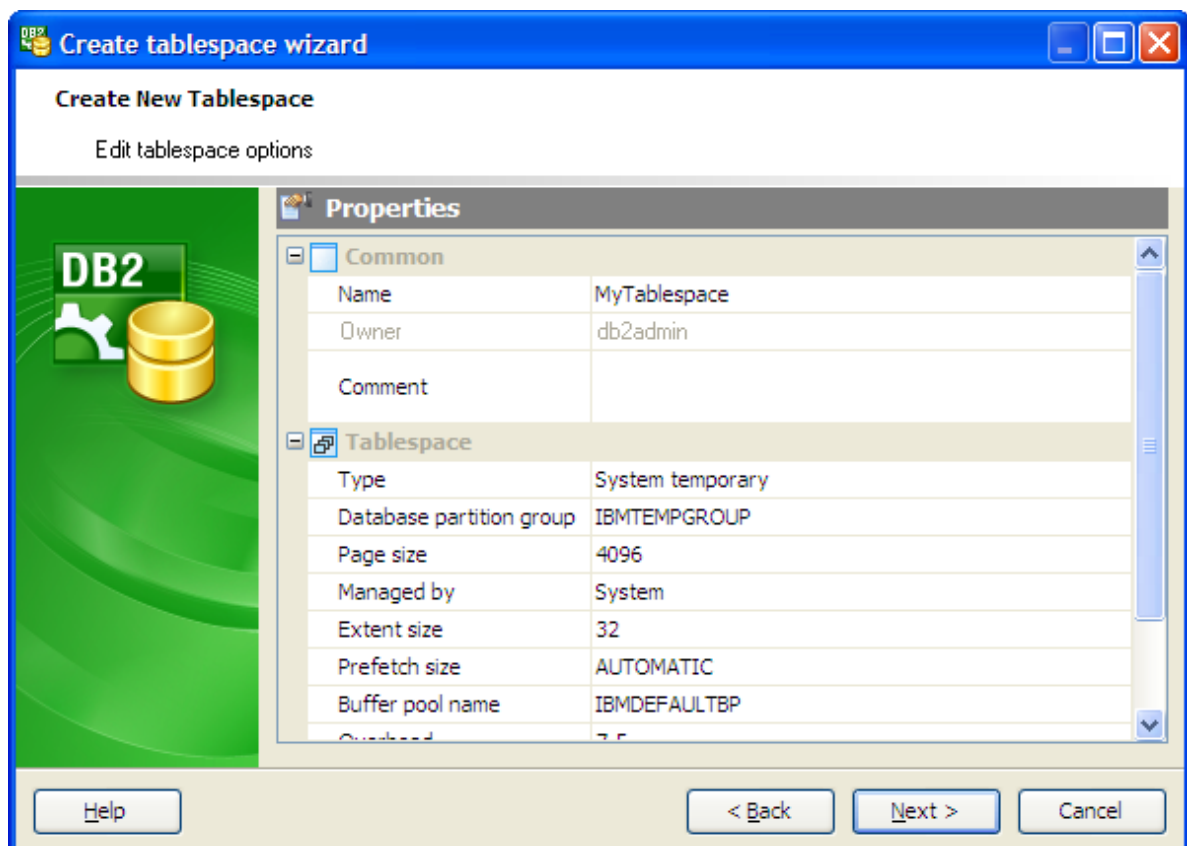
and confirm dropping in the dialog window.

### 6.4.1 Create Tablespace Wizard

[Create Tablespace Wizard](#) guides you through the process of creating a new server tablespace.

The basic principles of Create Object Wizards in DB2 Maestro are covered by the corresponding [topic](#)<sup>[36]</sup>. See below to find the description of wizard steps that are unique to the current object.

Specify tablespace options according to your needs. The detailed description is given below.



**Type** (*LARGE, REGULAR, SYSTEM TEMPORARY, or USER TEMPORARY*)

Specifies the type of table space that is to be created. If no type is specified, the default is determined by the MANAGED BY clause.

**Database partition group**

Specifies the database partition group for the table space. The database partition group must exist.

#### Page size

Defines the size of pages used for the table space. The valid values for integer without the suffix K are 4 096, 8 192, 16 384, or 32 768. The valid values for integer with the suffix K are 4, 8, 16, or 32. Any number of spaces is allowed between integer and K, including no space.

#### Managed by (SYSTEM,DATABASE)

*SYSTEM* specifies that the table space is to be an SMS table space. When the type of table space is not specified, the default behavior is to create a regular table space.

*DATABASE* specifies that the table space is to be a DMS table space. When the type of table space is not specified, the default behavior is to create a large table space.

#### Extent size

Specifies the number of pages that will be written to a container before skipping to the next container. The extent size value can also be specified as an integer value followed by K (for kilobytes) or M (for megabytes). If specified in this way, the floor of the number of bytes divided by the page size is used to determine the value for the extent size. The database manager cycles repeatedly through the containers as data is stored.

#### Prefetch size

Specifies to read in data needed by a query prior to it being referenced by the query, so that the query need not wait for I/O to be performed.

#### Buffer pool name

The name of the buffer pool used for tables in this table space.

#### Overhead

Specifies the I/O controller overhead and disk seek and latency time. This value is used to determine the cost of I/O during query optimization.

#### Transferrate

Specifies the time to read one page into memory. This value is used to determine the cost of I/O during query optimization.

#### Drop recovery

Indicates whether dropped tables in the specified table space can be recovered.

## 6.4.2 Tablespace Editor

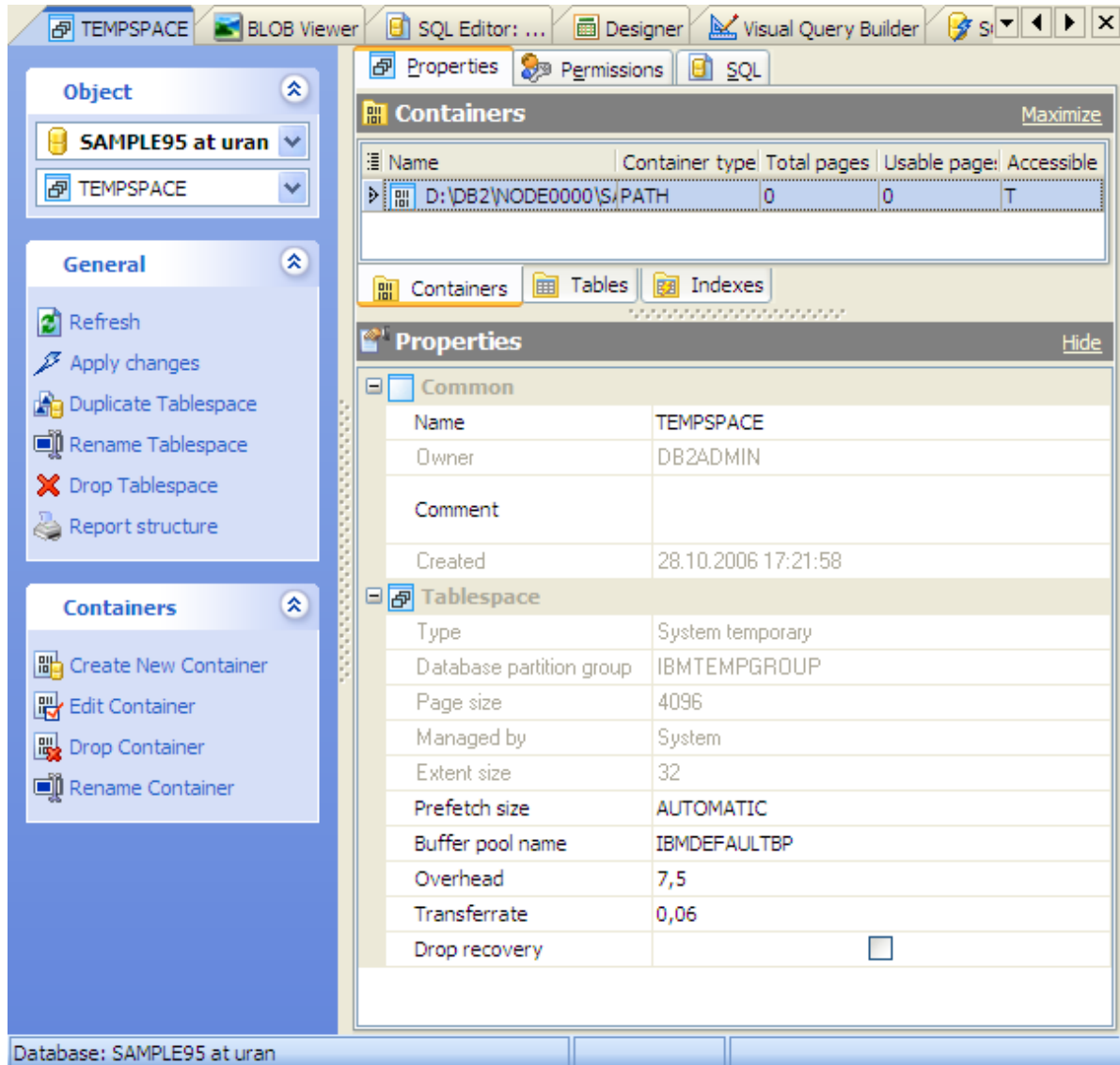
[Tablespace Editor](#) is the basic DB2 Maestro tool for working with existing tablespaces. It can be open automatically after the tablespace is created and is available on editing the tablespace (see [How to edit tablespace](#)<sup>[187]</sup> for details).

You can open a tablespace in [Tablespace Editor](#) from the [Explorer Tree](#) or [Object Manager](#).

The basic principles of Object Editors in DB2 Maestro are considered in the corresponding [topic](#)<sup>[38]</sup>. See below to find the description of editor tabs that are unique to the current object.

## Tablespace properties<sup>[190]</sup>

See also: [Create Tablespace Wizard](#)<sup>[188]</sup>



### 6.4.2.1 Editing tablespace properties

Tablespace Editor provides you with an ability to edit tablespaces.

**Type** (*LARGE, REGULAR, SYSTEM TEMPORARY, or USER TEMPORARY*)

Specifies the type of table space that is to be created. If no type is specified, the default is determined by the MANAGED BY clause.

**Database partition group**

Specifies the database partition group for the table space. The database partition group must exist.

#### Page size

Defines the size of pages used for the table space. The valid values for integer without the suffix K are 4 096, 8 192, 16 384, or 32 768. The valid values for integer with the suffix K are 4, 8, 16, or 32. Any number of spaces is allowed between integer and K, including no space.

#### Managed by (SYSTEM,DATABASE)

*SYSTEM* specifies that the table space is to be an SMS table space. When the type of table space is not specified, the default behavior is to create a regular table space.

*DATABASE* specifies that the table space is to be a DMS table space. When the type of table space is not specified, the default behavior is to create a large table space.

#### Extent size

Specifies the number of pages that will be written to a container before skipping to the next container. The extent size value can also be specified as an integer value followed by K (for kilobytes) or M (for megabytes). If specified in this way, the floor of the number of bytes divided by the page size is used to determine the value for the extent size. The database manager cycles repeatedly through the containers as data is stored.

#### Prefetch size

Specifies to read in data needed by a query prior to it being referenced by the query, so that the query need not wait for I/O to be performed.

#### Buffer pool name

The name of the buffer pool used for tables in this table space.

#### Overhead

Specifies the I/O controller overhead and disk seek and latency time. This value is used to determine the cost of I/O during query optimization.

#### Transferrate

Specifies the time to read one page into memory. This value is used to determine the cost of I/O during query optimization.

#### Drop recovery

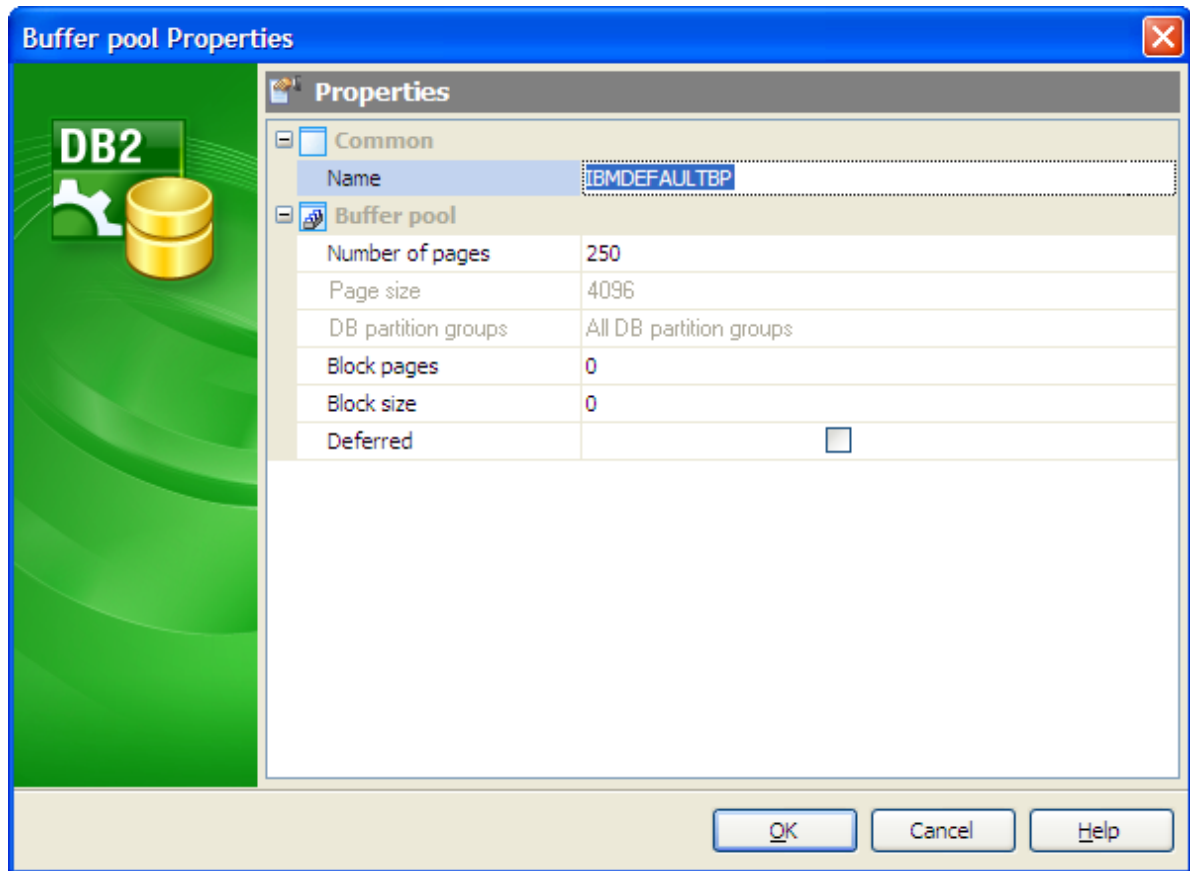
Indicates whether dropped tables in the specified table space can be recovered.

To apply the changes, select the [Apply Changes](#) item in the [Navigation bar](#) or use **Ctrl+F9** or **Ctrl+F7** shortcut keys.

It is also possible to modify object properties without opening the object editor: use the [Object Properties](#) item of the popup menu of the selected object from the explorer tree.

## 6.5 Boofer Pools

A [buffer pool](#) is associated with a specific table space, thereby allowing control over which data will share the same memory areas for data buffering.



### Number of pages

The number of pages for the new buffer pool.

### Page size

Specifies the size of the buffer pool. For a partitioned database, this will be the default size for all database partitions on which the buffer pool exists. The default is 1000 pages.

### DB partition groups

Identifies the database partition group or groups to which the buffer pool definition applies.

### Block pages

Specifies the number of pages that should exist in the block-based area. The number of pages must not be greater than 98 percent of the number of pages for the buffer pool.

### Block size

Specifies the number of pages in a block. The block size must be a value between 2 and 256 (SQLSTATE 54053). The default value is 32.

#### Deferred

If checked the buffer pool will be created when the database is deactivated (all applications need to be disconnected from the database). Reserved memory space is not needed; DB2 will allocate the required memory from the system.

## 7 Queries

DB2 Maestro provides several tools for working with SQL queries:

- [SQL Editor](#)<sup>[196]</sup> for editing the query text directly and executing SELECT queries;
- [Visual Query Builder](#)<sup>[202]</sup> for building SELECT, INSERT, UPDATE and DELETE queries visually;
- [SQL Script Editor](#)<sup>[248]</sup> for executing SQL scripts.

Both SQL Editor and Visual Query Builder supports [parameters in queries](#)<sup>[201]</sup>

Save frequently used queries to profiles and manage them in the same way as if they were database objects. This means that you can view queries in the explorer tree, in [Object Manager](#) and [Object Browser](#), use them in [BLOB Viewer](#) and [Diagram Viewer](#), perform drag-and-drop operation upon them, and copy them to clipboard like you copy an object.

### **How can I create a new SQL query?**

New queries can be created either in [SQL Editor](#) or in [Visual Query Builder](#).

To create a new query in [SQL Editor](#):

- select the [Tools | SQL Editor](#) main menu item;
- select the [Create New Query](#) item from the navigation bar;
- edit the query text on the [Editor](#) tab of [SQL Editor](#).

To create a new query in [Query Builder](#):

- select the [Tools | Visual Query Builder](#) main menu item;
- build the query on the [Diagram](#) tab of [Visual Query Builder](#).

DB2 Maestro also provides you with SQL Generator, a tool to create simple SQL statements.

### **How can I save a query to a file/profile?**

To save an existing query from the editor:

- to save the query to profile, use the [Save to profile](#) link from the [Navigation bar](#).
- to save the current query to an \*.sql file, select the [Save to file](#) item from the [Navigation bar](#);
- to save all the opened queries to one file, select the [Save all queries](#) item from the [Navigation bar](#);
- to save the designed diagram, select the [Save diagram](#) item from the [Navigation bar](#) of the [Diagram](#) tab of [Visual Query Builder](#).

### **How can I edit an existing SQL query?**

Queries can be opened either in [SQL Editor](#) or in [Visual Query Builder](#).

You can open the query directly from the Explorer tree with a double click or using popup menu. By default it will be opened in [SQL Editor](#).

To edit a query from file, open [SQL Editor](#) (the [Tools | SQL Editor](#) main menu item) and use [Load From File](#) from the [Navigation Bar](#) of [SQL Editor](#) to load a query from an \*.sql file.

To edit a query in [Query Builder](#), open the builder (the [Tools | Visual Query Builder](#) main menu item) and then perform one of the following operations:

- to edit a query from a profile, drag it from the [Explorer](#) and drop on the [Editor](#) tab;
- to load a previously saved diagram, use the [Load Diagram](#) item from the [Navigation Bar](#);
- to load a query from an \*.sql file, open the [Editor](#) tab and select the [Load query](#) item from the [Navigation Bar](#) .

On the [Query Builder](#) opening the [Diagram](#) tab contains the last edited query.

### **How can I execute an SQL query?**

To execute a query:

- create a new query or open the existing one;
- select the [Execute Query](#) item from the navigation bar of [SQL Editor](#) or [Visual Query Builder](#) respectively;
- view/edit the returned data on the [Result](#) tab.

## 7.1 SQL Editor

[SQL Editor](#) is the basic DB2 Maestro tool for creating and executing queries. It allows you to create and edit SQL text for the query, prepare and execute queries, and view the results of execution. To open [SQL Editor](#), select the [Tools | SQL Editor](#) main menu item.

The most popular query management actions (creating, editing, deleting) are covered by the corresponding [topic](#)<sup>[194]</sup>.

To use the editor for working with several queries, open new query tab with the [Create new query](#) link on the Navigation bar. With the tabs' popup menu you can create a new query, close existing one, save the query to profile, etc even if editor's navigation bar is closed. Queries' tabs [can be](#)<sup>[299]</sup> displayed at the all sides of the editor (bottom, top, left or right).

For more information about query executing and working with query result see the [corresponding topic](#)<sup>[199]</sup>.

### Working with query text

The [popup menu](#) of the editing area provides you with standard operations for working with text such as *Cut* (**Ctrl+X**), *Copy* (**Ctrl+C**), *Paste* (**Ctrl+V**), *Undo* (**Ctrl+Z**), *Redo* (**Shift+Ctrl+Z**) along with a possibility to convert selected text to different cases (*lower*, *UPPER*, and *NameCase*).

You can also comment/uncomment selected text (**Shift+Ctrl+.** and **Shift+Ctrl+,** shortcuts respectively). If no text is selected, the whole line will be commented. By the way, it is not necessary to select commented text to uncomment it, just press **Shift+Ctrl+.** having the cursor inside the commented text. Both kinds of comments (single-line and multi-line) are supported. [SQL Formatter](#)<sup>[198]</sup> is also at your disposal.

SQL Editor allows you to use [Visual Query Builder](#)<sup>[202]</sup> modal instance to design query visually and load the result query text directly in the editor area. For this purpose use the [Design query](#) link of the editor area's popup menu.

### Code completion

DB2 Maestro provides you with code completion (as on the screen below) to select from a list of tables, columns, views, or other objects without having to manually enter the object's name in the editor. You can activate the completion list by pressing the **Ctrl+Space** key combination.

### Syntax highlighting

Database objects are highlighted in the text. You can open the proper object editor by clicking the object name in the text with the **Ctrl** key pressed or with the [Find Object](#) link on the [Navigation bar](#). To adjust the highlighting settings, use [SQL highlight options](#)<sup>[319]</sup>.

### Line modification markers

Lines of code that have been edited during the current session are indicated with a yellow line in the left margin of the editor. When you save the file, the yellow

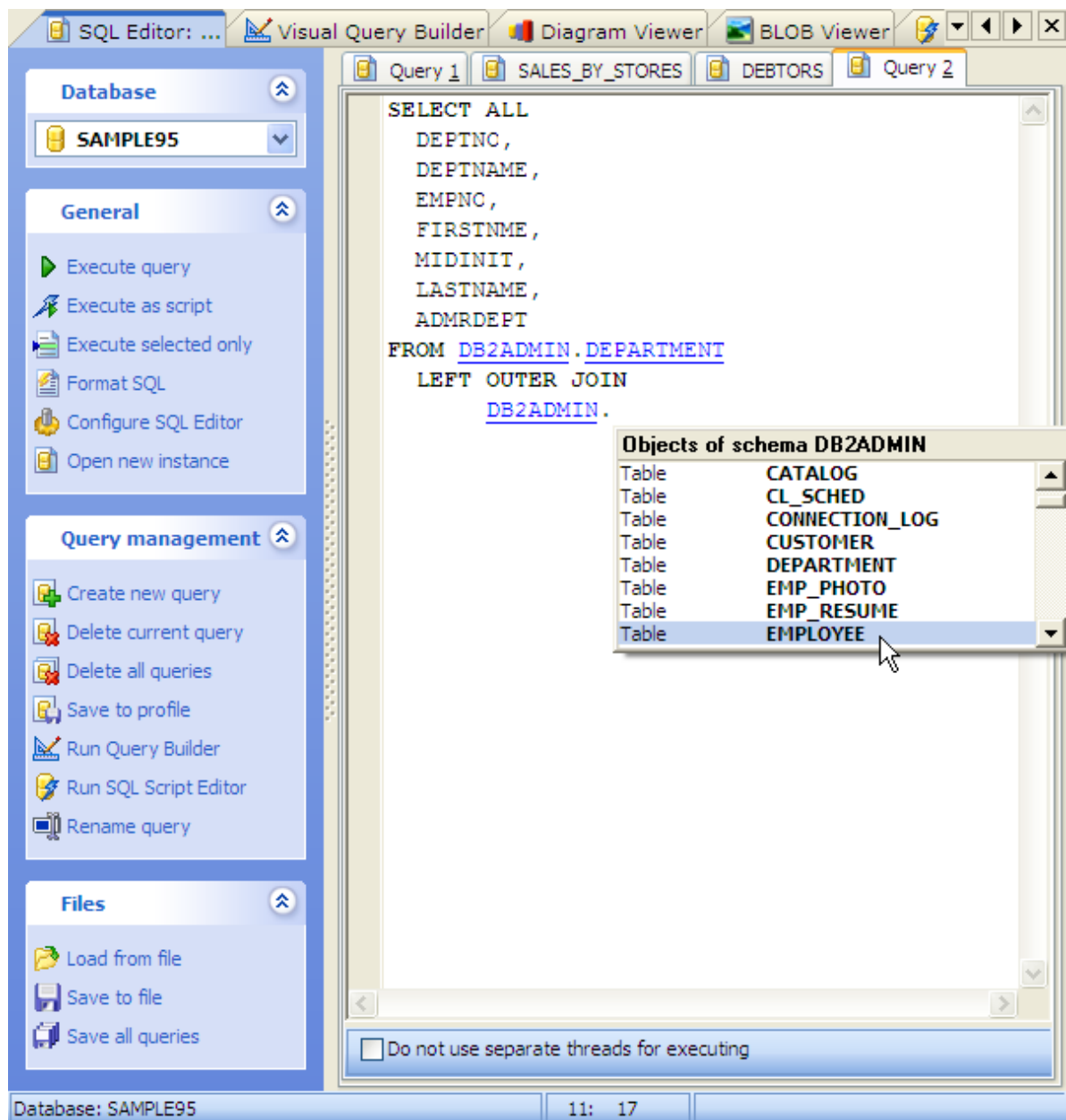
markers turn green. Thus at any time, yellow markers show changed but unsaved lines of code, and green markers show changes in this session that have been saved.

#### **Find and replace text**

Use find and replace to search for, and optionally, replace text in the [SQL Editor](#). To open [Find text/Replace text](#) window, use [Edit | Find/Replace](#) main menu item, corresponding link of popup menu, or **Ctrl+F/Ctrl+H** shortcut. You can also use the [Search again](#) link to apply recent Find text dialog.

#### **Managing the query text**

To load query from .sql file, use the corresponding link on the Navigation bar. You can also find there links allowing you to save query text to file, export the contents of the editor to RTF and HTML formats (to file or to clipboard), copy the selected text from to clipboard as a ready-to-use string written in one of the following programming languages: C#, C++, Delphi (Object Pascal), and Java, and also print/preview the contents of the editor.



**See also:** [Visual Query Builder](#)<sup>[202]</sup>, SQL Script Editor, [SQL Editor Options](#)<sup>[299]</sup>

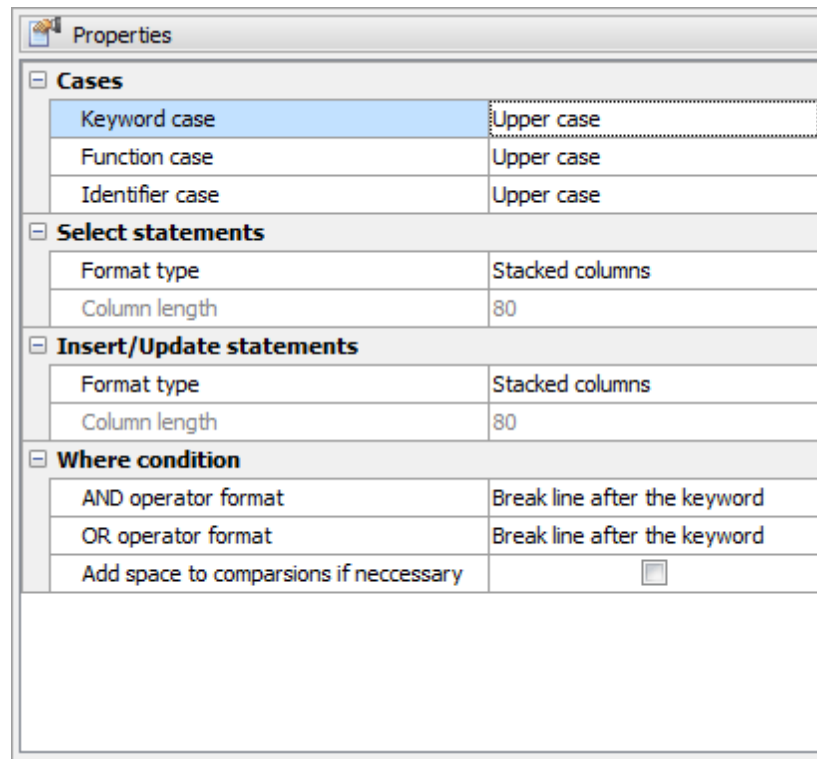
### 7.1.1 SQL Formatter

DB2 Maestro provides you with [SQL Formatter](#) for DML statements (*SELECT*, *INSERT*, *UPDATE* and *DELETE*). It can be invoked through the [Format SQL](#) link on the SQL Editor's navigation bar (**Ctrl+Alt+D** shortcut).

The following options allows you to tune up SQL scripts according to your preferences.

- Cases (for keywords, functions, and identifiers);
- Format type and column length for *INSERT/UPDATE*, and *SELECT* statements;

- *AND* and *OR* operators format.



## 7.1.2 Executing query

SQL Editor provides you with several variants of the query executing.

- To execute all statements of the text area with result data, click the [Execute query](#) item of the Navigation bar or use **F5**, **F8**, or **F9** shortcuts. Statements of each tab of SQL Editor are executed together in a separate thread in order to continue your work with the software while the query is executing.
- You can also [execute query as script](#) (**Shift+F5**, **Shift+F8**, **Shift+F9**). In this case the query does not return data.
- To execute only a selected part of the query text, use [Execute selected only](#) or the **Alt+F5**, **Alt+F8**, **Alt+F9** shortcuts.
- There is also a possibility to execute a statement at the cursor position. For this purpose, use the [Execute at cursor](#) link at the Navigation bar or use the **Ctrl+F5**, **Ctrl+F8**, or **Ctrl+F9** shortcuts.

If the query text is correct, the query is executed, and if the query statement is supposed to return data (e.g. SELECT statement), the [Result](#) tab opens with the data returned by the query. If an error occurs while executing the query, execution stop is stopped and the appropriate error message is displayed in the Information tab.

The [Result](#) area displays the result data in grid. All principles of working with data you can find in [Data Management](#) <sup>211</sup> section.

SQL Editor: ... BLOB Viewer Dependency tracker Visual Query Builder GAME@

**Database**

sdb\_demo at sun

**General**

- Execute query
- Execute as script
- Execute selected only
- Format SQL
- Show SQL Help
- Configure SQL Editor
- Open new instance

**Query management**

- Create new query
- Delete current query
- Delete all queries
- Save to profile
- Run Query Builder
- Run SQL Script Editor

**Files**

- Load from file
- Save to file
- Save all queries

**Data management**

- Export data
- Get SQL dump
- Print data

```

SELECT
  NBA.GAME.GAME_DATE,
  HOME_TEAM.CAPTION AS HOME_TEAM,
  (SELECT
    SUM(NBA.GAME_QUARTER.SCORE) AS FIELD_1
  FROM NBA.GAME_QUARTER
  WHERE
    (NBA.GAME_QUARTER.GAME_ID = NBA.GAME.ID)
    (NBA.GAME_QUARTER.TEAM_ID = HOME_TEAM.ID)
  (SELECT
    SUM(NBA.GAME_QUARTER.SCORE) AS FIELD_2
  FROM NBA.GAME_QUARTER
  WHERE
    (NBA.GAME_QUARTER.GAME_ID = NBA.GAME.ID)
    (NBA.GAME_QUARTER.TEAM_ID = HOME_TEAM.ID)
  )
FROM NBA.GAME

```

Result 1 Result 2

Table

Drag a column header here to group by that column

	GAME_DATE	HOME_TEAM	HOME_TEAM_SCORE
	Click here to define a filter		
1	28.10.2008	Boston Celtics	
2	28.10.2008	Chicago Bulls	1
3	28.10.2008	Los Angeles Lakers	
4	29.10.2008	Philadelphia 76ers	
5	29.10.2008	Orlando Magic	
6	29.10.2008	Washington Wizards	
7	29.10.2008	New York Knicks	1
8	29.10.2008	Detroit Pistons	1
9	29.10.2008	Minnesota Timberwolves	
10	29.10.2008	Oklahoma City Thunder	
11	29.10.2008	San Antonio Spurs	

Records fetched: 1315

Information  
1315 rows fetched (0,56 sec)

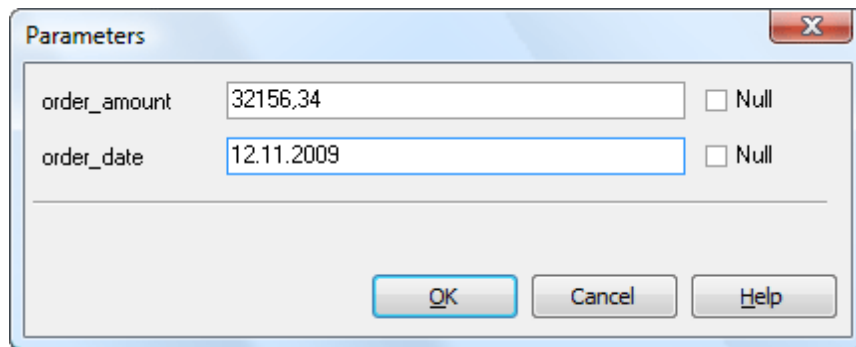
RESULT\_LIST Query 5 Query 1

Database: sdb\_demo at sun

### 7.1.3 Query Parameters

Both [SQL Editor](#)<sup>[196]</sup> and [Visual Query Builder](#)<sup>[202]</sup> admit to using parameters inside the query text. A parameter is a kind of variable. Its value can be specified just before the query execution in the [Parameters](#) window. In the query text the parameter should appear as an identifier with a colon (':') at its beginning, e.g. `:param1`.

The [Parameters](#) dialog is used to specify the query parameters as well as values of the input parameters of procedures or functions before the execution. Enter parameter values and click the **OK** button to apply the values and execute the query or use the **Cancel** button to abort the execution.



**Note:** To allow use parameters in query text, check the corresponding option at the [Tools](#)<sup>[296]</sup> tab of DB2 Maestro Options.

## 7.2 Visual Query Builder

[Visual Query Builder](#) is provided for building data manipulation statements visually. It allows you to create and edit queries without knowledge of SQL, prepare and execute queries, and view the results of the execution. Builder can produce *INSERT*, *UPDATE* and *DELETE* statements as well as the *SELECT* statements containing subqueries and/or *UNIONS*. One instance of the builder can be used only for one query at a time. To open [Visual Query Builder](#), select the [Tools | Query Builder](#) main menu item.

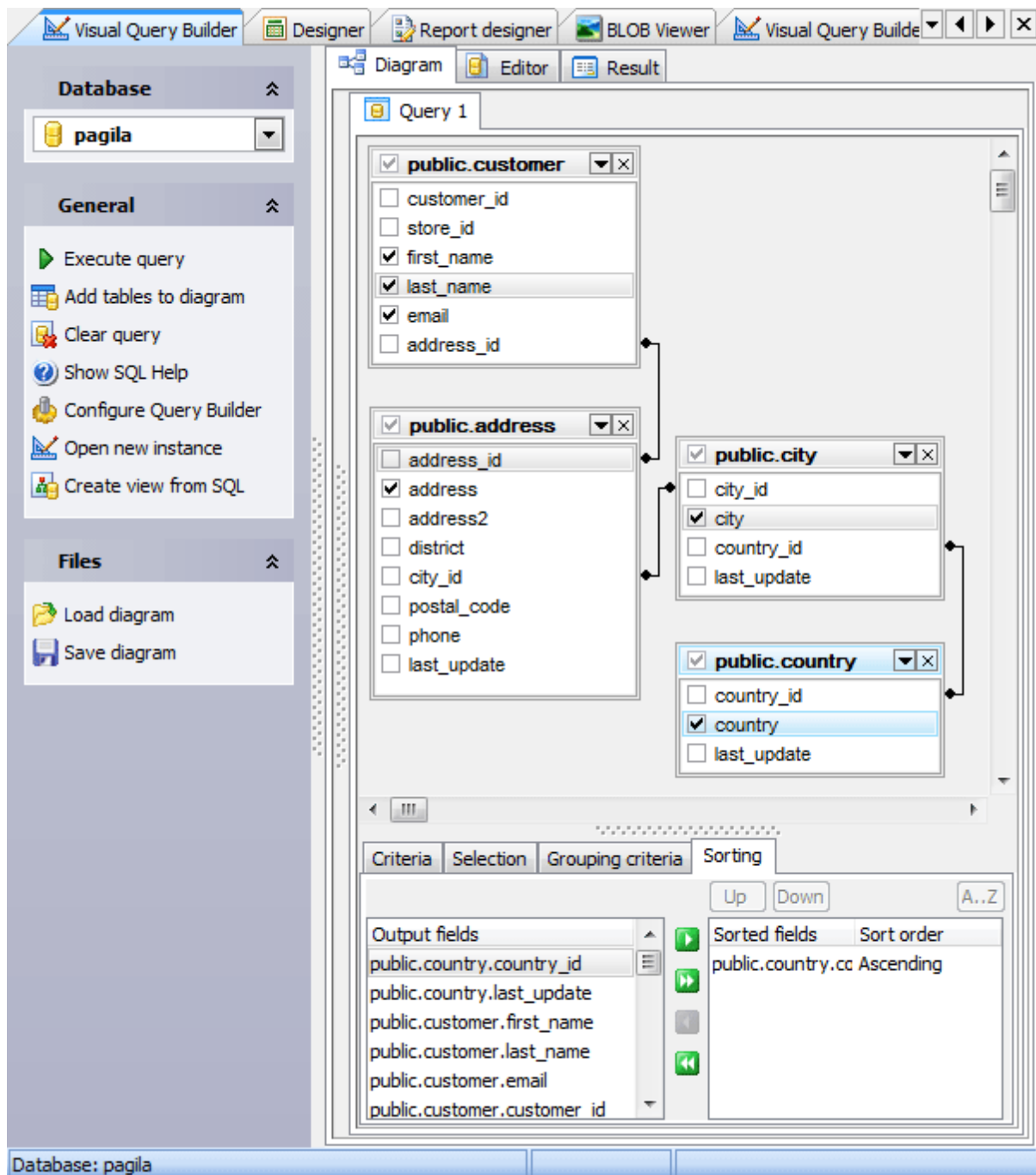
The most popular query management actions (creating, editing, deleting) are covered by the corresponding [topic](#)<sup>[194]</sup>.

Builder consists of 3 tabs:

- [Diagram](#)<sup>[203]</sup> - to create a query from a graphical interface,
- [Editor](#)<sup>[208]</sup> - to modify the query text before its executing,
- [Result](#)<sup>[209]</sup> (appears after the query executing) - for working with data the query returns.

The builder also allows you to create a view based on the prepared query. For this purpose after the query creating and possibly testing use the Create view from SQL link at the Navigation bar to invoke the corresponding window, and specify [view properties](#)<sup>[92]</sup>.

**See also:** [SQL Editor](#)<sup>[196]</sup>, [Visual Query Builder Options](#)<sup>[301]</sup>, [Query Parameters](#)<sup>[201]</sup>



### 7.2.1 Creating query diagram

The **Diagram** tab is the main area of Visual Query Builder. Using its graphical interface you can select tables and views, join or select columns, and add conditions to the statement.

The **Query Explorer** field occupies the left side of Visual Query Builder main window. All the queries included in the result query (unions, subqueries) are represented at the Query Explorer for prompt access. They are grouped by kind and listed under the according node.

Below step-by-step description of query diagram creating.

- **Select the statement type** from the drop-down list at the top of the [Diagram](#) tab (*SELECT, INSERT, UPDATE, DELETE*).

■ **Add required tables to the Diagram area.**

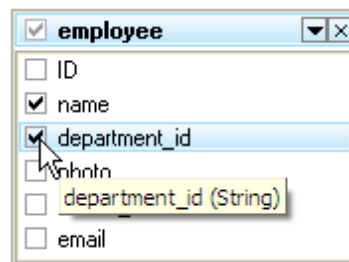
Use the [Add Table\(s\)](#) link of the area popup menu and select tables from the opened window (Use **Ctrl** or **Shift** pressed to select several tables).

To add only one table, simply drag it from the [Database Explorer](#) or from [Object Manager/Browser](#) to the [Diagram](#) area.

To remove the object, close its window or select the object and press the **Delete** key.

■ **Pick up columns with data to output**

To include a table field to the query, tick off the option box to the left of the field name in the list or double-click it to see the blue icon next to the field name.

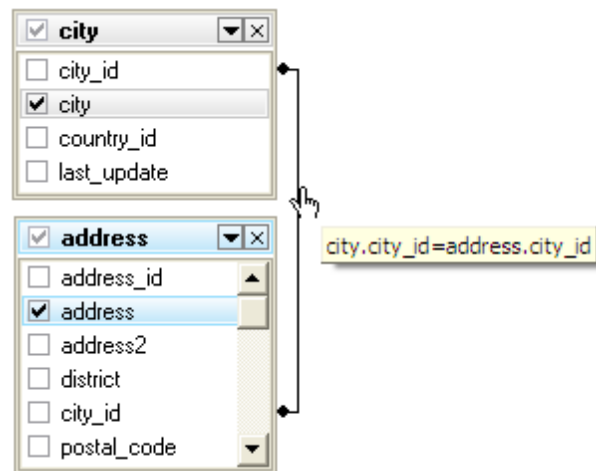


To include all the table fields, tick off the option box to the left of the table caption. In case none fields is included, the SQL statement is generated as `SELECT * FROM <Table_Name>`, i.e. all the fields are selected.

To remove the fields from the query, uncheck the corresponding boxes.

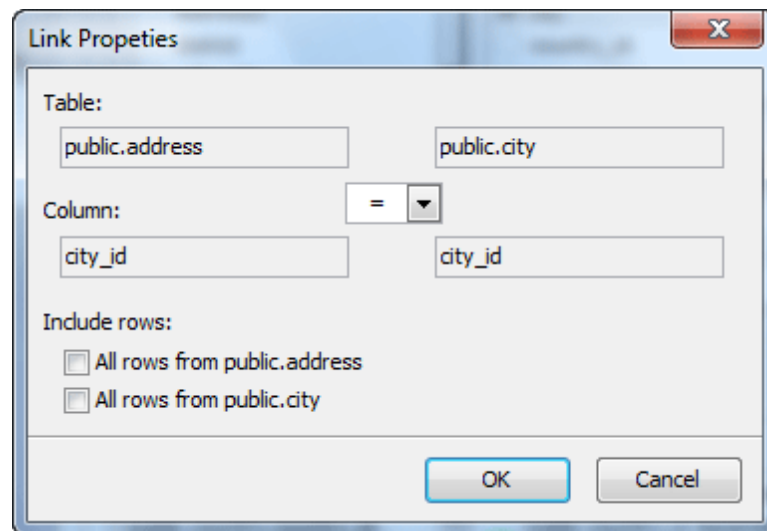
■ **Join tables if necessary**

Visual Query Builder supports *INNER JOIN*, *LEFT OUTER JOIN*, and *RIGHT OUTER JOIN*. To associate database objects by two fields, drag a field from the first object's field list to a field from another object's field list. This will set a link between these objects by the selected fields. After you finish dragging, a line will appear between the linked fields. By default *INNER JOIN* syntax will be used.



You can view the properties of the object association from the query tab directly. Just set the cursor to the link line. A hint containing the association condition will appear.

To edit the properties, select the [Properties](#) item from the popup menu. A dialog window will appear, there you can change the association condition by selecting it from the list (`=`, `>`, `<`, `>=`, `<=`, `<>`). To create *LEFT OUTER JOIN* / *RIGHT OUTER JOIN* statements, check [All rows from first\\_table](#)/[All rows from second\\_table](#) from the window.



To remove a link between objects, select the [Delete Link](#) item from the popup menu.

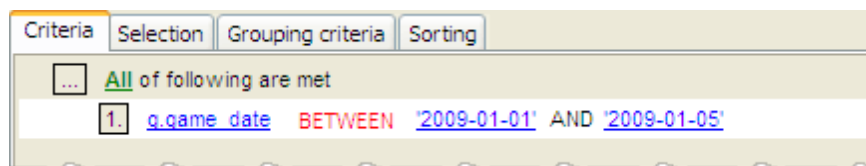
To delete all the links of an object, click the '-' button next to the object alias. To insert a point to the link line, select the [Insert Point](#) item from the popup menu, and the new point will appear. Using the point you can move the link line. It does not cause any changes in the query but makes the diagram performance vivid and the visual building more obvious.

#### Specify WHERE condition

**Criteria** tab allows you to set the selection conditions. To add a condition, click the button on the left and select the [Add condition](#) item in the popup menu. Edit the condition by clicking its parts and setting their values. Clicking the button to the left of the condition string activates the popup menu which allows you to add a new condition of the same enclosure level, add a new enclosure level, delete the current condition, open or close the condition if it is composite.

A simple condition string contains three fields: an argument, a condition and a second argument (if required for the condition). Clicking each field allows you to set its value. Clicking the argument field make it possible to edit the argument as a text field. You can set a table name or a definite value in this field. The popup menu of the field in the editing mode which contains the [Insert Field](#) function (also called by the **Shift+Enter** hot keys combination).

This function allows you to choose a field from the list of all the table fields available in the query. The popup menu of the condition field allows you to specify the condition you need. The way of proceeding the condition is set in the upper string of the area (*All, Any, None, or Not all* of the following are possible variants). Click the underlined word to modify it.



#### ■ Create subquery if necessary

You can add one or more subqueries to further limit the tables and records returned from a *SELECT* statement when setting a *WHERE* condition in the query builder. To add subquery:

- open [Criteria](#) tab;
  - click the button on the left and select the [Add condition](#) item in the popup menu;
  - right click on an argument field and use the [Insert query](#) link of the popup menu;
  - build the subquery in the new query tab that have appeared in the [Diagram](#) area,
- or
- open [Selection](#) tab;
  - use the [Insert query](#) link of the popup menu;
  - build the subquery in the new query tab that have appeared in the [Diagram](#) area.

#### ■ Use column aliases

You can set/edit the object alias directly from the query tab by double-clicking the object caption.

Criteria Selection Grouping criteria Sorting			
<input type="checkbox"/> Select only unique records			
Source field name	Name of output field	Aggregate	Grouping
nba.game.game_date	Game_date		
home_team.caption	caption		
(SELECT SUM(nba.game_quarter.score)	Home_team_score		
(SELECT SUM(nba.game_quarter.score)	Away_team_score		
away_team.caption	caption		
nba.channel.short_caption	short_caption		

In case the alias is used as the expression's column name use the [Selection](#) tab displays the output fields of the query. It allows you to edit the names of the query or CASE output fields, set their displaying order and set the aggregate functions (*SUM*, *MIN*, *MAX*, *AVG*, etc.) for each field.

<i>AVG</i>	Returns the average of the values in a group
<i>BIT_AND</i>	Returns the bitwise AND of all bits in the expression.
<i>BIT_OR</i>	Returns the bitwise OR of all bits in the expression.
<i>COUNT</i>	Returns the total number of items in a column. This function does not ignore NULL values when calculating results.
<i>GROUP_CONCAT</i>	Returns a string result with the concatenated non-NULL values from a group.
<i>MAX</i>	Returns the maximum value for the column.
<i>MIN</i>	Returns the minimum value for the column.
<i>STD</i>	Returns the population standard deviation of the expression.
<i>STDDEV</i>	Returns the sample standard deviation of a numeric expression evaluated over a set.
<i>SUM</i>	Returns the sum of all the values in the expression.
<i>VARIANCE</i>	Returns the population standard variance of the expression.

To remove the field from the list, select the [Delete current row](#) item from the popup menu of the field row.

To modify the input query field, double-click it and then type the field name or select one from the drop-down list.

To modify the output query field name, double-click it and enter the field name.

#### ■ **DISTINCT option**

To specify removal of duplicate rows from the result set, open the [Selection](#) tab and check the [Select only unique records](#) box.

#### ■ **Add HAVING statement**

Set the conditions to be included into the HAVING statement within the [Grouping Criteria](#) tab. They are set in the same way as the *WHERE* conditions. To set the aggregate function for the field, double-click the field row in the [Aggregate](#) column and then type the function name or select one from the drop-down list.

#### ■ **ORDER BY clause**

Set the way of sorting the query records within the [Sorting](#) tab. The field list on the left represents all the output query fields; the list on the right contains fields by which the query records will be sorted. To move the field from one list to another, drag the selected field or use the [Add](#) and [Remove](#) buttons. To change the sorting order, select a field in the right list and move it using the [Up](#) and [Down](#) buttons.

To change the sorting direction, select a field in the right list and switch the direction (*Ascending, Descending*) using the [A..Z/Z..A](#) button.

#### ■ **Create UNIONS**

To combine the result from multiple `SELECT` statements into a single result set, use the [Add union](#) link of the Query Explorer popup menu.

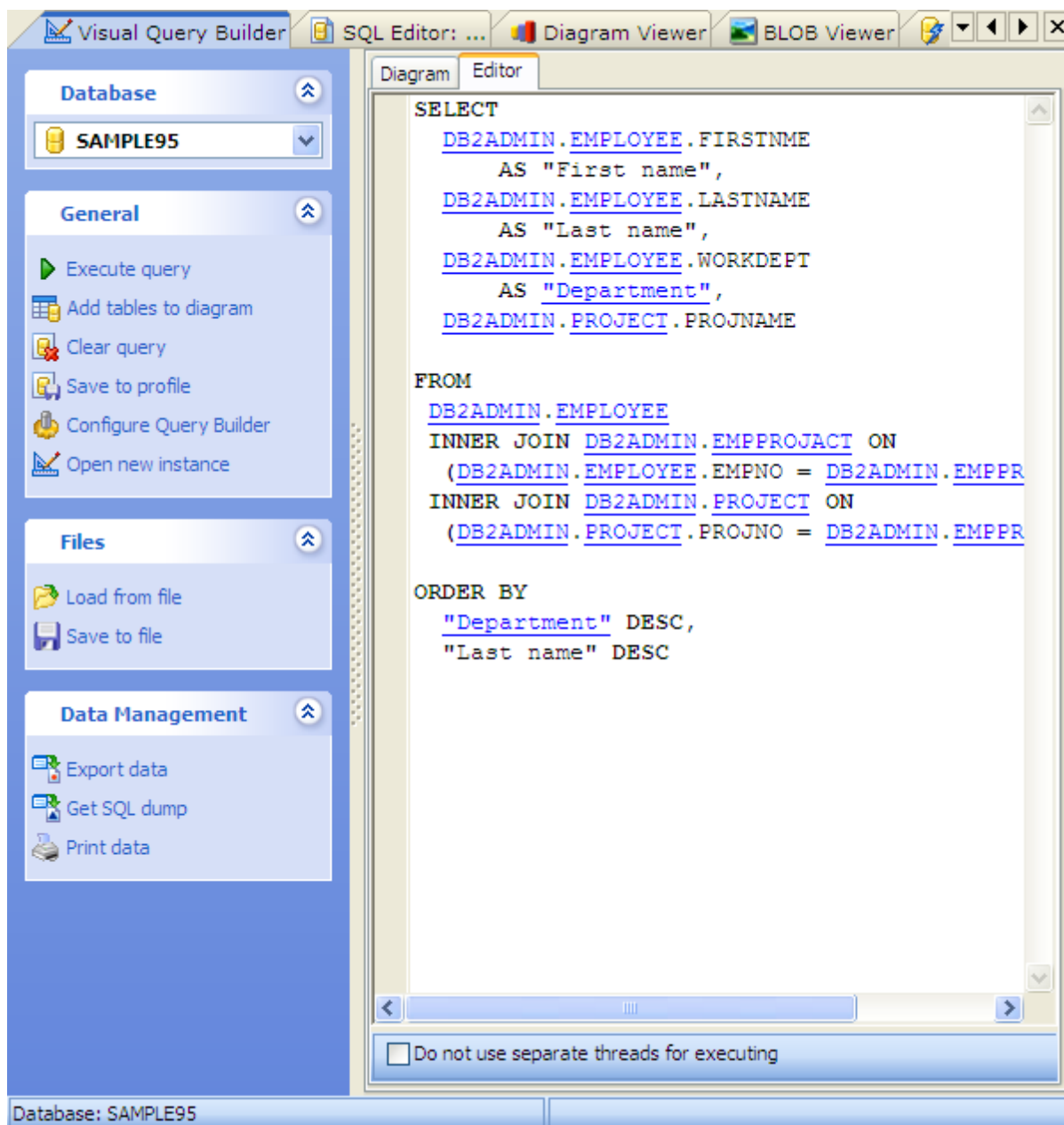
**Note:** The column names from the first `SELECT` statement are used as the column names for the results returned.

Selected columns listed in corresponding positions of each `SELECT` statement should have the same data type.

### 7.2.2 Working with editor area

In the [Editor](#) area the query text is automatically generated while you are building the query.

You can edit this text according to the rules of SQL, and all the changes will be displayed on the [Diagram](#) page of [Visual Query Builder](#).



### 7.2.3 Executing query

To execute the query select the **Execute** item in the navigation bar. After that the **Result** tab is displayed. This page contains the result data returned by the query, as a grid (see Data View for details). The popup menu of this tab and the items of the navigation bar allow you to export data and get SQL dump.

Visual Query Builder Designer Report designer BLOB Viewer Visual Query Builder

Database **pagila**

General

- Show SQL Help
- Configure Query Builder
- Open new instance
- Create view from SQL

Data Management

- Export data
- Get SQL dump
- Print data

Diagram Editor Result

Table

\* country

	* first_name	* last_name	email
	NULL	NULL	NULL
country : United Kingdom (9)			
1	ANNE	POWELL	ANNE.POWELL@sakilacustomer.org
2	APRIL	BURNS	APRIL.BURNS@sakilacustomer.org
3	ARMANDO	GRUBER	ARMANDO.GRUBER@sakilacustomer.org
4	CECIL	VINES	CECIL.VINES@sakilacustomer.org
5	DAN	PAINE	DAN.PAINE@sakilacustomer.org
6	GILBERT	SLEDGE	GILBERT.SLEDGE@sakilacustomer.org
7	MARSHALL	THORN	MARSHALL.THORN@sakilacustomer.org
8	MATTIE	HOFFMAN	MATTIE.HOFFMAN@sakilacustomer.org
9	SANDRA	MARTIN	SANDRA.MARTIN@sakilacustomer.org
country : United States (36)			
1	ALICE	STEWART	ALICE.STEWART@sakilacustomer.org
2	ANA	BRADLEY	ANA.BRADLEY@sakilacustomer.org
3	ASHLEY	RICHARDSON	ASHLEY.RICHARDSON@sakilacustomer.org
4	BETTY	WHITE	BETTY.WHITE@sakilacustomer.org
5	BILL	GAVIN	BILL.GAVIN@sakilacustomer.org
6	BRANDY	GRAVES	BRANDY.GRAVES@sakilacustomer.org
7	BRYAN	HARDISON	BRYAN.HARDISON@sakilacustomer.org
8	CAROLE	BARNETT	CAROLE.BARNETT@sakilacustomer.org
9	CAROLINE	BOWMAN	CAROLINE.BOWMAN@sakilacustomer.org
10	CASSANDRA	WALTERS	CASSANDRA.WALTERS@sakilacustomer.org
11	CLINTON	BUFORD	CLINTON.BUFORD@sakilacustomer.org
12	DIANA	ALEXANDER	DIANA.ALEXANDER@sakilacustomer.org
13	EVA	RAMOS	EVA.RAMOS@sakilacustomer.org
14	IAN	STILL	IAN.STILL@sakilacustomer.org

Records fetched: 599

Information

599 rows fetched (0,19 sec)

Database: pagila

## 8 Working with data in DB2 Maestro

Query results and table data are displayed on the [Data](#)<sup>[70]</sup> or [Result](#)<sup>[199]</sup> tabs of [Table Editor](#)<sup>[68]</sup>, [SQL Editor](#)<sup>[196]</sup> or [Visual Query Builder](#)<sup>[202]</sup>.

Data are displayed as a grid (or as info cards) which provide a lot of useful features such as editing, grouping, sorting, filtering, etc. See [Data View](#)<sup>[212]</sup> for details.

Navigation bars of these tabs as well as popup menus of their working areas places at your disposal the following functions for managing data:

- [Export Data](#)<sup>[228]</sup> allows you to export data to various formats, including MS Excel, MS Access, RTF, HTML, PDF and more.
- [Get SQL Dump](#)<sup>[235]</sup> exports data to the SQL script as a number of INSERT statements.
- [Import Data](#)<sup>[238]</sup> provides you with possibility to import data from MS Excel, MS Access, DBF, XML, TXT, and CSV.
- [Edit BLOB](#)<sup>[223]</sup> allows you to view and edit the content of BLOB and TEXT fields.

## 8.1 Data View

DB2 Maestro represents all data (stored in tables and views, results of queries and procedures) in [grid](#)<sup>[213]</sup> or in [info cards](#)<sup>[218]</sup>. By default, data is displayed in a grid - tabular view of data. To change the type of the data representation, use the drop-down list at the top of the tab. Both of the data representations support UNICODE/UTF-8 data. The status bar displays the number of records in the current data set.

	CUST_NO	CUSTOMER	CONTACT_FIRST	CONTACT_LAST	PHONE_NO	ADDRESS
1	1001	Signature Design	Dale J.	Little	(619) 530-2710	150
2	1002	Dallas Technologies	Glen	Brown	(214) 960-2233	P. 4
3	1003	Buttle, Griffith and Co.	James	Buttle	(617) 488-1864	230
4	1004	Central Bank	Elizabeth	Brocket	61 211 99 88	66
5	1005	DT Systems, LTD.	Tai	Wu	(852) 850 43 98	400
6	1006	DataServe International	Tomas	Bright	(613) 229 3323	200
7	1007	Mrs. Beauvais	NULL	Mrs. Beauvais	NULL	P.C
8	1008	Anini Vacation Rentals	Leilani	Briggs	(808) 835-7605	330
9	1009	Max	Max	NULL	22 01 23	1 E
10	1010	MPM Corporation	Miwako	Miyamoto	3 880 77 19	2-6
11	1011	Dynamic Intelligence Corp	Victor	Granges	01 221 16 50	Flo
12	1012	3D-Pad Corp.	Michelle	Roche	1 43 60 61	22
13	1013	Lorenzi Export, Ltd.	Andreas	Lorenzi	02 404 6284	Via

### Navigation buttons

Both data representations are equipped with navigation buttons. They are represented at the top of the data tab and allow you to navigate between records and to accomplish common operations:

- To add a new record, use the *Plus* button or the **Insert** shortcut.
- To delete a new record, use the *Minus* button or the **Delete** shortcut.
- To edit an existing record, push the corresponding button or invoke the [Data Input Form](#)<sup>[219]</sup> using popup menu of the necessary record, with **Ctrl+Alt+D** shortcut, or with the corresponding link at the Navigation bar. To edit a field value, click it and enter the new one inline.

The pagination option allows you to limit the number of browsed records. By default, the number of records represented in grid at once is 1000. To change the number of records represented in the current grid, enter the necessary value in the pagination bar. To specify the default one or to disable pagination, use the [data grid option](#)<sup>[308]</sup>.

### Navigation bar

The Data management group of the Navigation bar allows to invoke [Data Input Form](#)<sup>[219]</sup>,

[SQL Editor](#)<sup>[196]</sup> with SELECT query, [Data Export](#)<sup>[228]</sup>, and [Data Import](#)<sup>[238]</sup> modules using corresponding links, also get [SQLdump](#)<sup>[235]</sup> of the current data set and print current data with enabled preview in WYSIWYG mode.

**See also:** [Table Editor](#)<sup>[68]</sup>, [SQL Editor](#)<sup>[196]</sup>, [Visual Query Builder](#)<sup>[202]</sup>

### 8.1.1 Working with data grid

Our software offers two grid modes:

- the full grid mode is a fully functional data representation equipped with abilities to filter and to sort data;
- the simple grid mode is provided for working with large number of records. For speed-up data fetching, filtering and sorting abilities are not enabled in this mode. The notification bar at the top of the grid (see the picture below) announces that the grid has been switched to the simple mode.

Result 1 Result 2 Result 3

Table

The grid has been switched to the simple mode because of the query returned more than 4000 rows (you can customize this number in the [Options](#) dialog). Filtering, sorting and grouping features are not enabled in this mode.

Other actions:  
[Switch to full mode now](#) | [Always use full mode](#) | [Dismiss this message](#)

CNO	TITLE	FIRSTNAME	NAME	ZIP	ADDRESS
3000	Mrs	Jenny	Porter	10580	1340 N. Ash Street, #3
3100	Mr	Peter	Brown	48226	1001 34th St., APT.3
3200	Company	NULL	Datasoft	90018	486 Maple St.
3300	Mrs	Rose	Brian	75243	500 Yellowstone Drive, #2
3400	Mrs	Mary	Griffith	20005	3401 Elder Lane
3500	Mr	Martin	Randolph	60615	340 MAIN STREET, #7
3600	Mrs	Sally	Smith	75243	250 Curtis Street
3700	Mr	Mike	Jackson	45211	133 BROADWAY APT. 1
3800	Mrs	Rita	Doe	97213	2000 Humboldt St., #6
3900	Mr	George	Howe	75243	111 B Parkway, #23
4000	Mr	Frank	Miller	95054	27 5th St., 76
4100	Mrs	Susan	Baker	90018	200 MAIN STREET, #94
4200	Mr	Joseph	Peters	92714	700 S. Ash St., APT.12
4300	Company	NULL	TOOLware	20019	410 Mariposa St., #10
4400	Mr	Antony	Jenkins	20903	55 A Parkway, #15
4401	Company	NULL	MagicStrawberry	78146	76 Highland Road, #120
4402	Company	NULL	OrangeHand	78609	212 Oak Avenue, #30

Records fetched: 4495

**Information**  
 4495 rows fetched ( 2,00 sec)

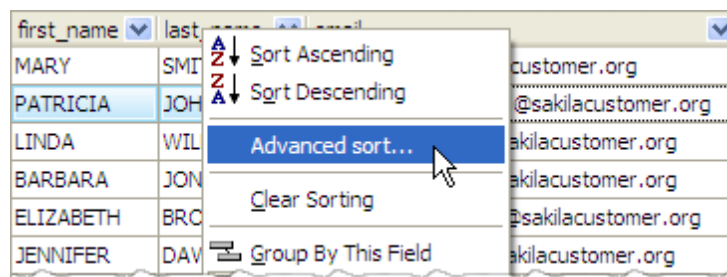
By default, the grid automatically switches to the simple mode for queries returning more than 5000 records (the number can be customized in the [Options](#) dialog).

The following abilities are not available in the simple grid mode:

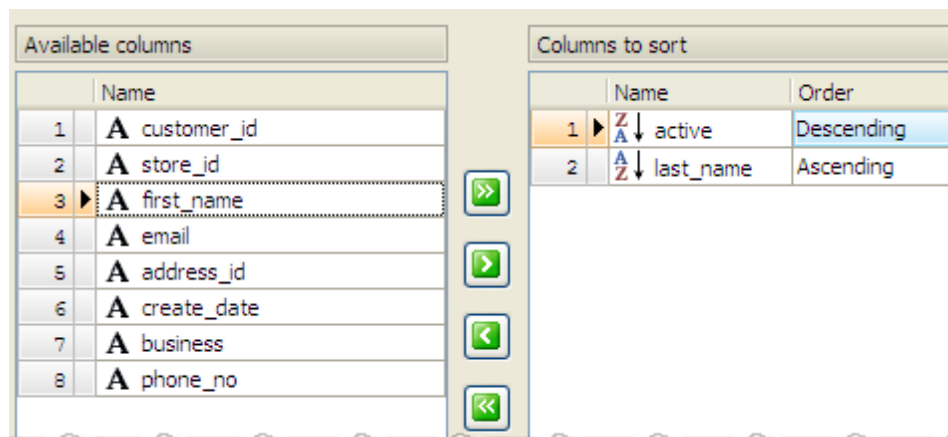
#### ■ **Sorting data (only in the full grid mode)**

Click the column caption to sort data by the values of this column. To select sort order (ascending or descending), use popup menu of the column caption.

To sort data on a combination of grid columns, use the [Advanced sort...](#) link of the popup menu of the grid's header. The [Advanced sorting](#) window will be shown.



Select there the columns you want to sort from the [Available columns](#) list in the order of priority. Specify the sort order if necessary and click OK.



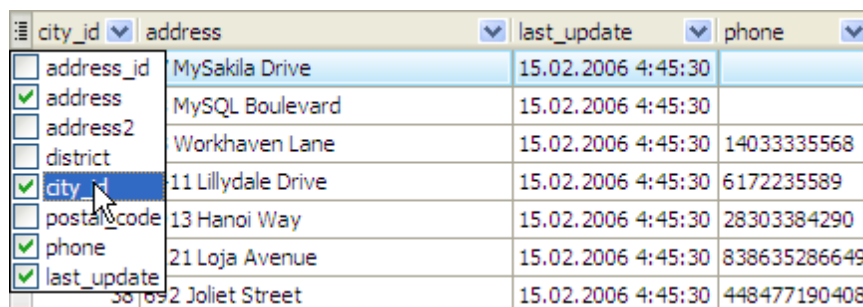
To cancel the sorting order, press **Ctrl** and click on the sorted column caption.

#### ■ Filtering represented records (only in the full grid mode)

There are several ways to filter data represented in grid. See [the corresponding topic](#)<sup>[220]</sup> to find out their descriptions.

#### ■ Hiding selected columns

You can show/hide columns using a button in the left top corner of the grid. Just check/uncheck the column in the drop-down list.



#### ■ Columns reordering

To reorder columns, use drag-n-drop.

### ■ Grouping records

You can group grid data by any of the columns by dragging the column header to the destination area. Now all the records are displayed as subnodes to the grouping row value as shown in the picture. To reverse grouping, just drag the column name from the upper area back.

The screenshot shows the DB2 Maestro interface with a data grid. The grid is grouped by 'round' and 'date'. The columns are ID, team1ID, team2ID, score1, score2, refereeID, and comments. The data is displayed in a hierarchical view with expandable nodes for each round and date. The bottom status bar shows 'Records fetched: 380' and an information popup indicates '380 rows fetched ( 0,64 sec)'.

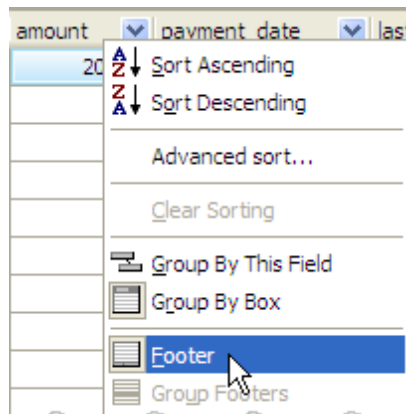
ID	team1ID	team2ID	score1	score2	refereeID	comments
round : 1						
round : 2						
round : 3						
date : 24.08.2004						
date : 25.08.2004						
24	6	2	3	0	8	Jeffers brukte 34 minutter pa a vinne
22	1	4	3	0	17	I den hundrede kampen i alle konkurra
29	16	5	1	2	19	Southamptons slapp inn mal i sin 11. li
27	9	13	0	2	0	Fulhams forste tap denne sesongen,
28	14	18	2	2	18	Newcastle skuffer i arets Premier Leag
26	19	17	1	1	12	Det ble uavgjort pa Hawthornes etter
date : 30.08.2004						
31	12	8	0	0	1	Igjen skuffet Manchester United mot
date : 14.12.2004						
round : 4						
round : 5						
date : 11.09.2004						
date : 12.09.2004						
date : 13.09.2004						
round : 6						

Records fetched: 380

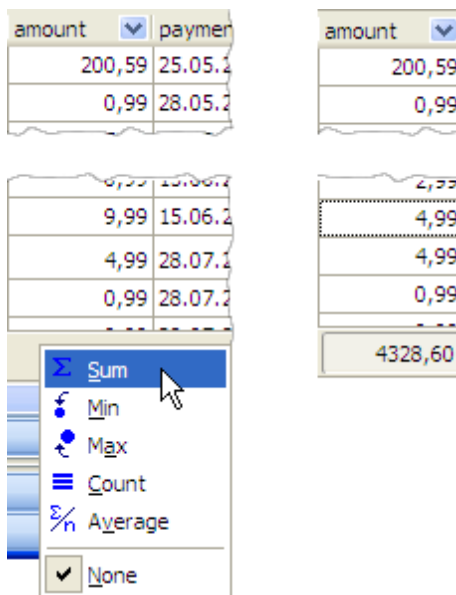
Information  
380 rows fetched ( 0,64 sec)

### ■ Using aggregate functions

To get a sum of column values, a min or a max value, an average column value or an amount of records, use **Data Grid Footer**. Select the **Footer** item at the grid caption's popup menu.



It will be shown at the bottom of the grid. The popup menu of the footer allows you to get an aggregate function result calculated with the corresponding column values.



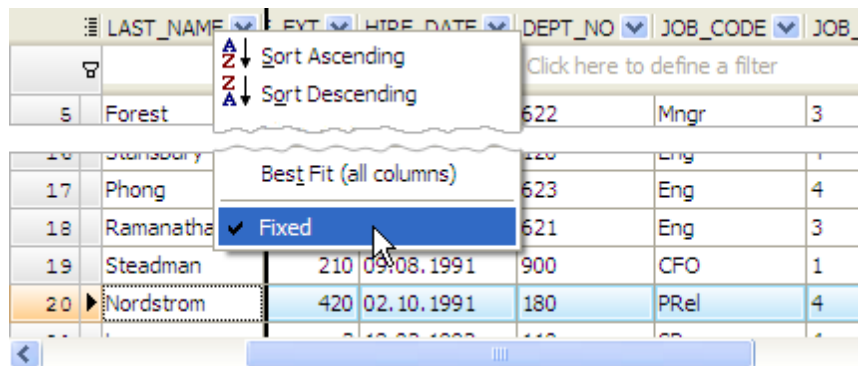
For grouped data use [Group Footers](#).

### ■ Data alignment

The grid's header popup menu allows to align column data. Use the [Alignment](#) link and select the alignment type.

### ■ Fixing columns

You can fix grid columns to view them permanently when working with other grid data. To fix a column, choose the corresponding item from the grid's header popup menu.

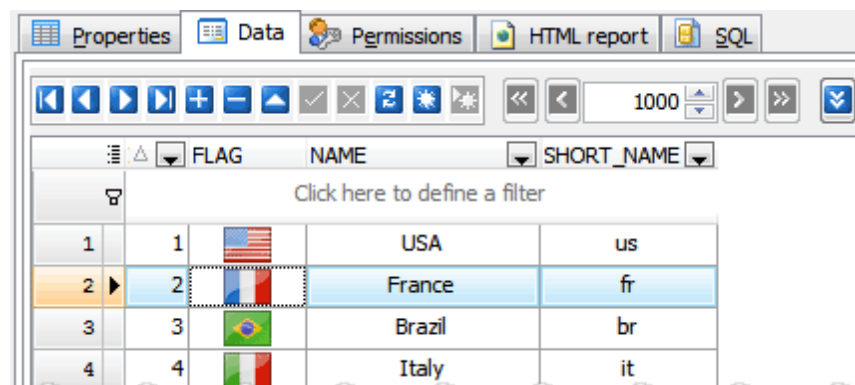


### ■ Row numbering

There is also a possibility to display row numbers in grids. You can [adjust](#)<sup>[309]</sup> the corresponding column to yours liking.

### ■ Inline images

It is possible to display images directly in the grid as on the picture below.



To enable/disable this view mode, open the *Enable inline images* window using the *Manage inline images* item of the column popup menu. The window options allow to set or change the image fitting and specify the row height. To add new images or change existing ones, use [BLOB Editor](#)<sup>[223]</sup> (see below).

### ■ Working with BLOBs

To [edit a BLOB field](#)<sup>[223]</sup>, double click the field, or use the corresponding popup menu item. There are also possibilities to export all BLOBs stored in the table column to files and import BLOBs from a directory to the table columns. In this case you need to set the Target directory, specify the template to be used for file names and the column BLOBs to be exported from (imported to).

## 8.1.2 Working with info cards

**Info cards** correspond to the records. You can [filter records by custom conditions](#)<sup>[220]</sup> and edit data directly in info cards or with [Data Input Form](#)<sup>[219]</sup>.

The screenshot displays the 'Data Input Form' in DB2 Maestro, showing a grid of six player records. Each record is presented as a form with fields for various attributes. The records are for Gilbert Arenas, Hilton Armstrong, Darrell Arthur, Trevor Ariza, Ron Artest, and D.J. Augustin. The interface includes a toolbar at the top with navigation and action icons, and a status bar at the bottom indicating 'Records fetched: 67'.

id:	first_name:	last_name:	career_start_year:	career_end_year:	position_id:	photo:	country_id:	height:	birthday:	weight:	college_id:	current_team_id:	current_number:
18	Gilbert	Arenas	2001	0	6		1	193	06.01.1982	97,5	15	12	0
20	Hilton	Armstrong	2006	0	11		1	211	11.11.1984	106,6	7	27	12
	Darrell	Arthur							25.03.198				
19	Trevor	Ariza	2004	0	10		1	203	30.06.1985	95,3	2	5	3
21	Ron	Artest	1999	0	10		1	201	13.11.1979	117,9	16	22	96
	D.J.	Augustin							10.11.198				

Records fetched: 67

### 8.1.3 Data input form

Use [Data Input Form](#) to add new records or edit existing ones. To invoke the dialog, use the corresponding link from the popup menu or **Ctrl+Alt+D** shortcut.

address_id	address	district
4	1411 Lillydale Drive	QLD
5	1913 Hanoi Way	Nagasaki
6	1121 Loja Avenue	California
7	692 Joliet Street	Attika
8	1566 Inegl Manor	Mandalay
9	53 Idfu Parkway	Nantou
10	1795 Santiago de Compostela Way	Texas

The dialog's fields contain the values of the current grid row. Use the [Insert](#) button to enter values of a new record and the [Post](#) button to update the current row. The [Cancel](#) button reverts all the field values within a form to their initial values (or to the last posted values). The [Previous](#) and [Next](#) buttons allow you to switch between grid records without closing the dialog.

There are possibilities to use lookup editors on working with columns linked with foreign keys, a calendar for *timestamp* columns and a calculator for *decimal* ones.

## 8.1.4 Data filtering

DB2 Maestro support filtering records by the following methods:

### Filter by a column value

Select the [Use as Filter](#) item from the field popup menu to filter records by the current column value.

### Filter by several column values

Use the drop-down button in the column caption area to filter records by the selected column value(s) or enter the filter condition directly in [the filter row](#) <sup>313</sup>.

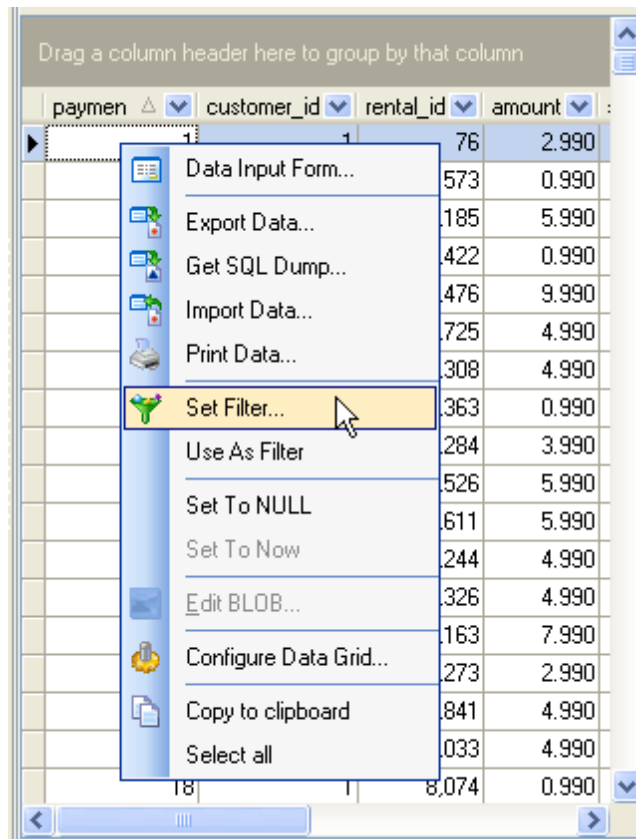
### Filter by two operators

Invoke simple filter dialog using the [Custom](#) item of the column caption area drop-

down list. Select a logical operator for checking the column values (like "is less than", "is greater than", etc.) and set the value to be checked by this operator in the next box; then set the second condition if necessary in the following way and set the relation between these two conditions, whether both of them should be matched or just one of them; use the '\_' character to represent any single symbol in the condition and the '%' character to represent any series of symbols in the condition.

### **Filter by any custom criteria**

To filter data according to more difficult custom conditions, use the Filter Builder dialog. To invoke the dialog, use the [Set Filter](#) link of popup menu or click the [Customize](#) button on the [Filter](#) panel. This panel is visible if any filtering is already applied to the grid (you can use column header menu or grid menu for quick filtering).



The dialog also allows to save filter criteria to an external file for future use.

After you set a filter, the filtering panel becomes visible at the top/bottom of the grid where you can see the active filtering condition and easily enable or disable it by clicking the check box on the left. To customize the filtering process, use [filter options](#)<sup>313</sup>.

The [Copy current filter as SQL condition to clipboard](#) feature is useful in case the same compound filter is applied several times. Just once apply the filter, copy to clipboard as SQL condition, paste to [SQL Editor](#)<sup>196</sup> and save as a query. You can also use [Generate query](#) link on the Navigation bar.

**See also:** [Data View](#)<sup>[212]</sup>, [Table Editor](#)<sup>[68]</sup>, [SQL Editor](#)<sup>[196]</sup>, [Visual Query Builder](#)<sup>[202]</sup>

## 8.2 BLOB Editor

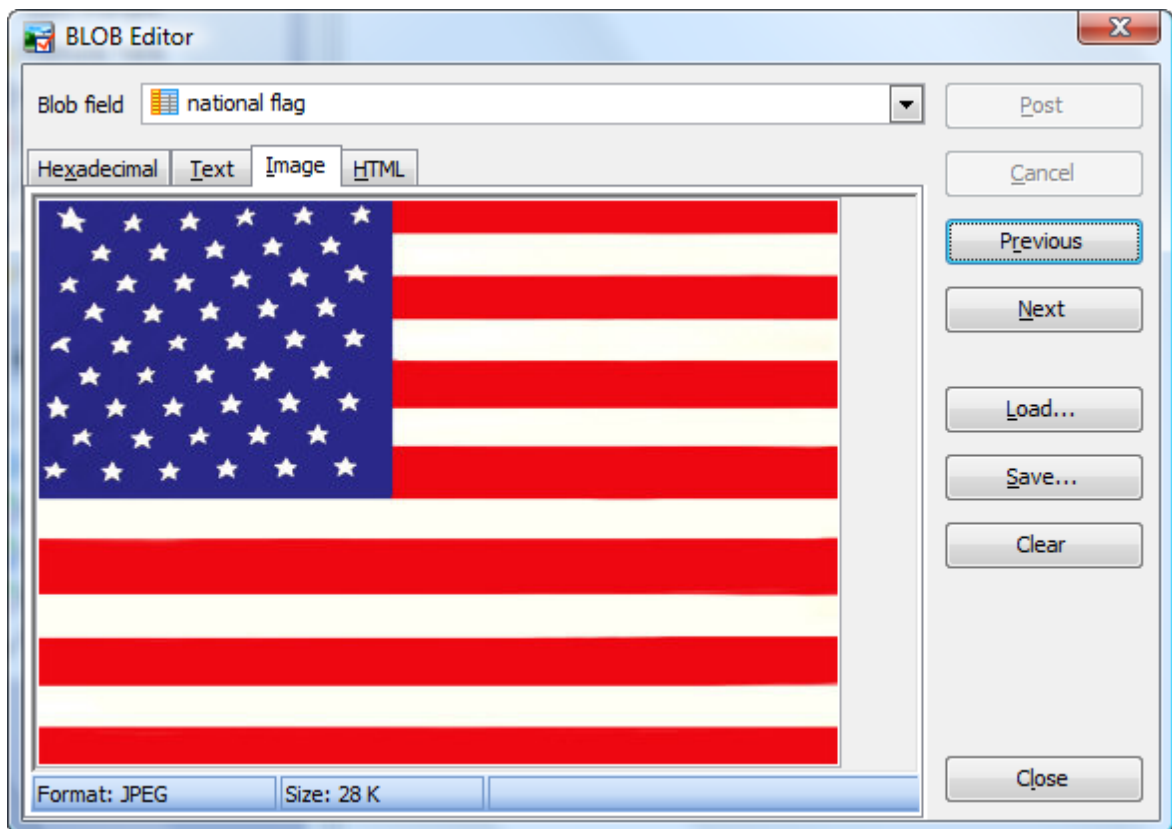
BLOB Editor is a tool to view and edit BLOB data in the following ways: [hexadecimal dump](#)<sup>[224]</sup>, [plain text](#)<sup>[224]</sup>, [graphical image](#)<sup>[223]</sup>, [HTML page](#)<sup>[225]</sup>, or [PDF document](#)<sup>[226]</sup>. BLOB Editor is invoked from [data grid](#)<sup>[212]</sup> of any [table editor](#)<sup>[68]</sup> or the result tab of [SQL Editor](#)<sup>[196]</sup> and [Visual Query Builder](#)<sup>[202]</sup> by double clicking of the BLOB field to be edited or with the [Edit BLOB](#) link of the field's popup menu. The editor also can be called from [BLOB Viewer](#)<sup>[252]</sup> with the [Edit current BLOB](#) button.

With BLOB Editor you can work with all BLOB columns of the grid. To switch between columns, select the necessary one from the [BLOB field](#) list.

[BLOB Editor](#) allows you to navigate between the grid records using the [Previous](#) and [Next](#) buttons. You can load the new BLOB content and save or clear it using corresponding buttons. After changes are made, click the [Post](#) button to apply the changes or the [Cancel](#) button to discard them.

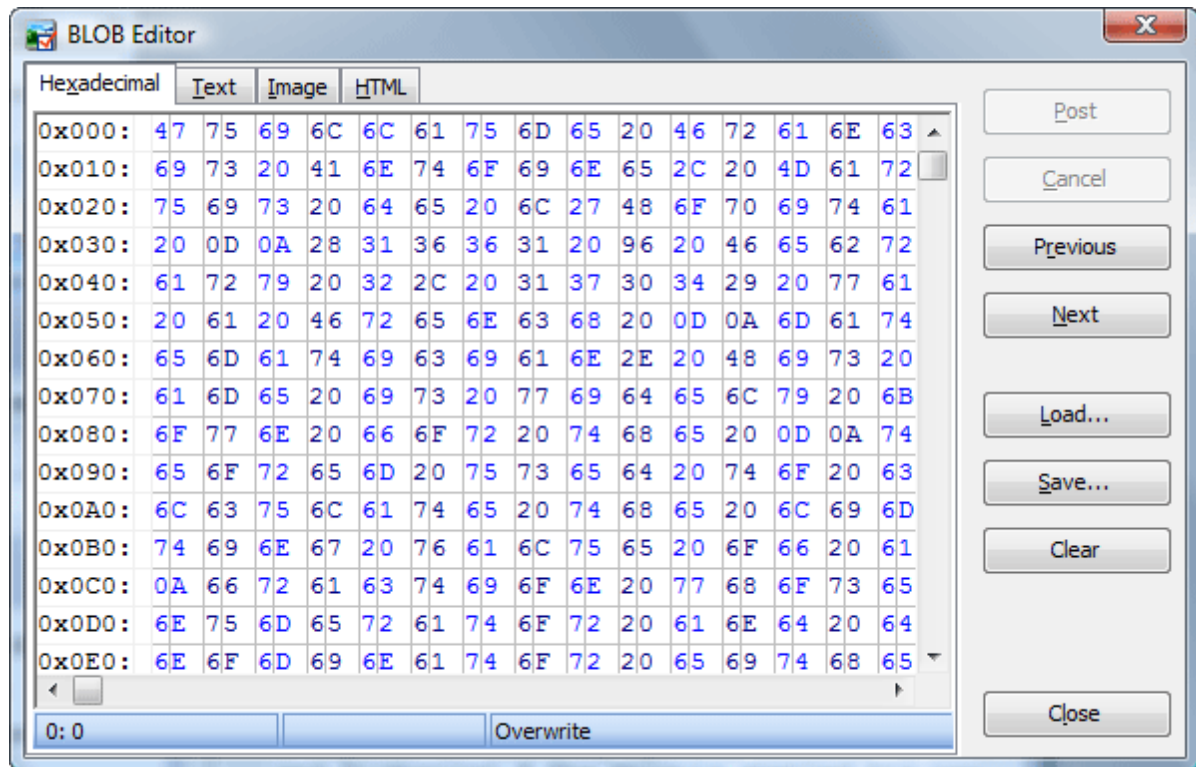
### 8.2.1 Editing as image

The [Image](#) panel of BLOB Editor displays field data as graphical image. Use the [Save](#) and [Load](#) buttons to save the image to a file or load an image from a file. A graphical representation of BLOB data supports five image formats: BMP, Windows metafile, JPEG, GIF and PNG.



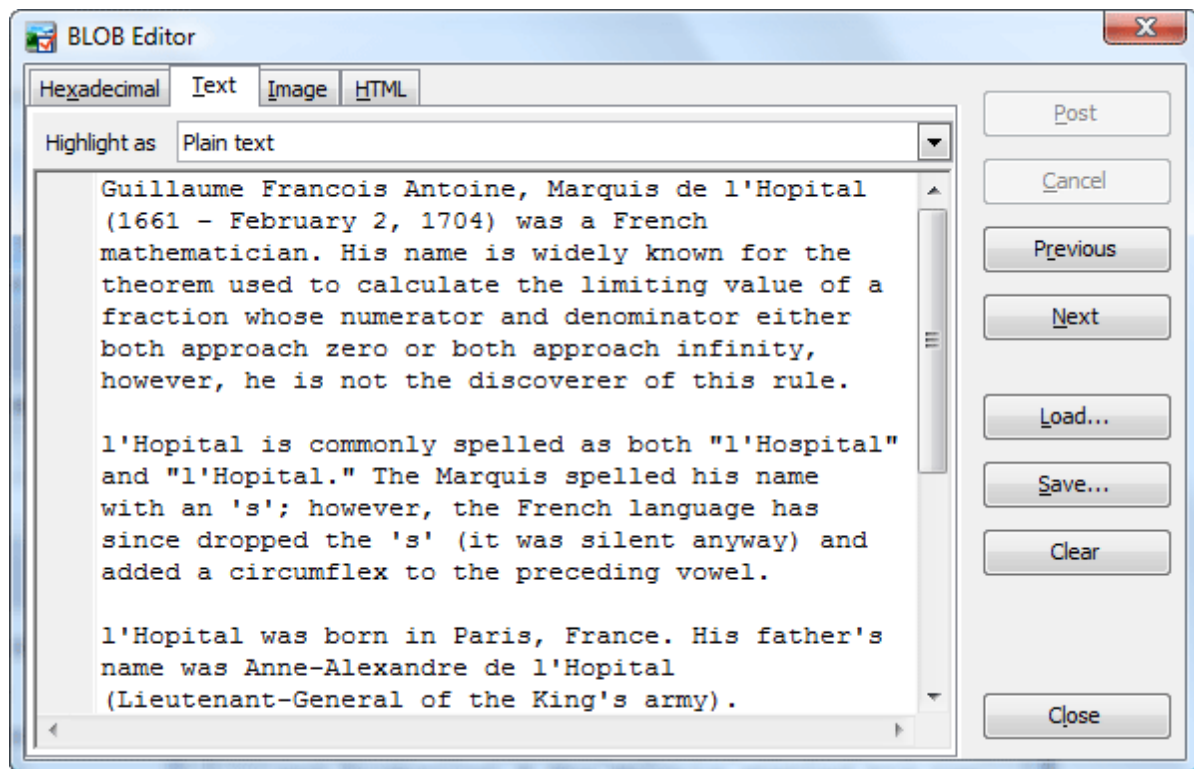
## 8.2.2 Editing as hexadecimal dump

The [Hexadecimal](#) panel allows you to edit data in hexadecimal mode. To load/save a hexadecimal dump from/to a file, use the corresponding buttons. Use the Insert key to switch between Insert and Overwrite modes.



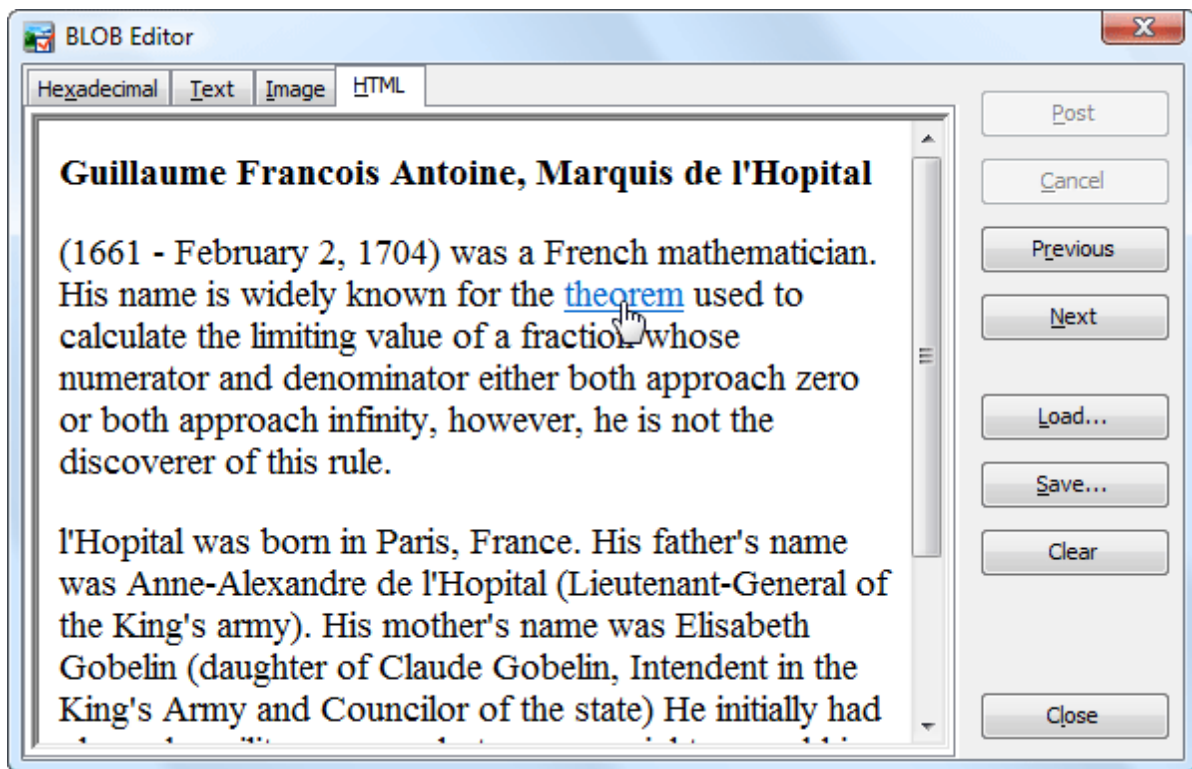
## 8.2.3 Editing as plain text

The [Text](#) panel allows you to edit data as a simple text. Several types of text highlighting are available (*Plain text*, *SQL*, *XML*, *Java*, *VBScript*, *JScript*, *Cmd batch*, *PHP*, *CSS*, *UnixShell Script*, *INI*, and *HTML*). The popup menu of the panel allows you to invoke [Find Text](#), [Replace Text](#) and [Go to line](#) dialogs.



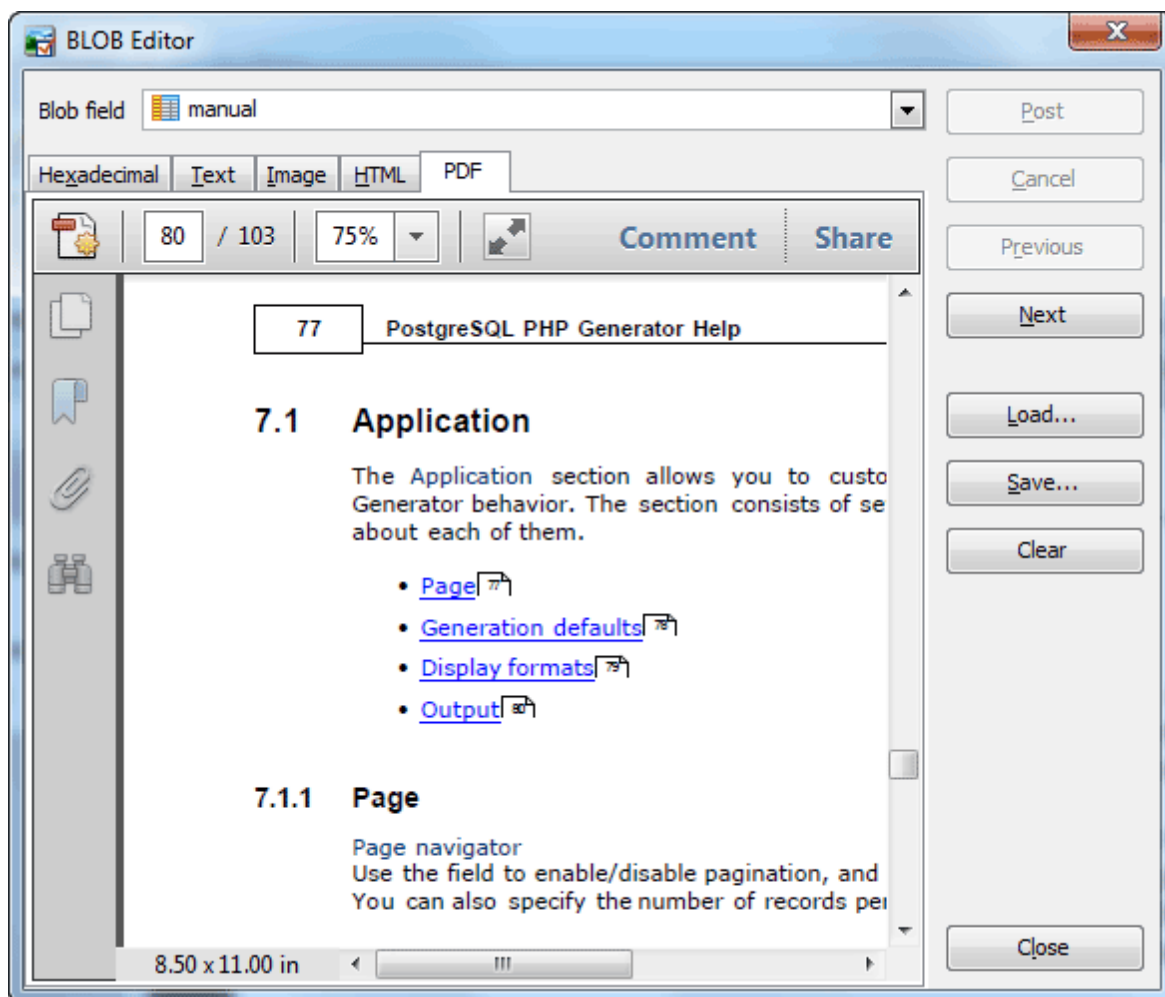
#### 8.2.4 Editing as HTML

The [HTML](#) panel presents field data as HTML. You can load a new content of the current field from a .html file or type it manually within the [Text](#) tab of the editor.



### 8.2.5 Editing as PDF document

The PDF panel presents field data as PDF document. To accomplish common operations with data, use the Adobe Reader toolbar.



## 8.3 Export Data Wizard

Data Export wizard is a tool to save data from DB2 tables, views, and queries to the most popular formats. It allows you to fully customize output files including header and footer, fonts, colors, and data formats.

Export Data tool supports:

- Microsoft Office Excel 97-2003, 2007
- CSV
- HTML
- XML
- Text
- Microsoft Office Word 97-2003, 2007
- Microsoft Office Access
- OpenDocument Spreadsheet
- OpenDocument Text
- DBF
- PDF
- RTF
- DIF
- SYLK
- LaTeX.

In order to run the wizard you should either

- open the table in [Table Editor](#);
- go on to the [Data](#) tab

or

- open and execute the query in [SQL Editor](#) or [Query Builder](#);
- proceed to the [Result](#) tab

and select the [Export Data](#) item from the [Navigation Bar](#).

To export your data,


- [Set the format and the name](#)<sup>[228]</sup> of the destination file;
- Specify such additional options of the result file as [header and footer](#)<sup>[229]</sup>, [formats applied to exported data](#)<sup>[231]</sup> and [some format-specific options](#)<sup>[231]</sup>;
- [Select columns](#)<sup>[230]</sup> you want to include into result files;
- [Specify other export options](#)<sup>[234]</sup>.








**See also:** Get SQL Dump, [Import Data Wizard](#)<sup>[238]</sup>

### 8.3.1 Setting destination file name and format

Select one of the available destination formats and set the name for the result file. The file name extension in the [Destination file name](#) box varies according to the selected export type.


Destination format \_\_\_\_\_

 Select one of the available destination formats.

-  ☒ Microsoft Office Excel 97 - 2003
-  ☐ Microsoft Office Excel 2007 - 2010
-  ☐ Delimiter-separated values (CSV, DSV, TSV)
-  ☐ Text file (Fixed-width columns)
-  ☐ HTML
-  ☐ XML
-  ☐ Other

Microsoft Office Word 97 - 2003 ▼

Destination file \_\_\_\_\_

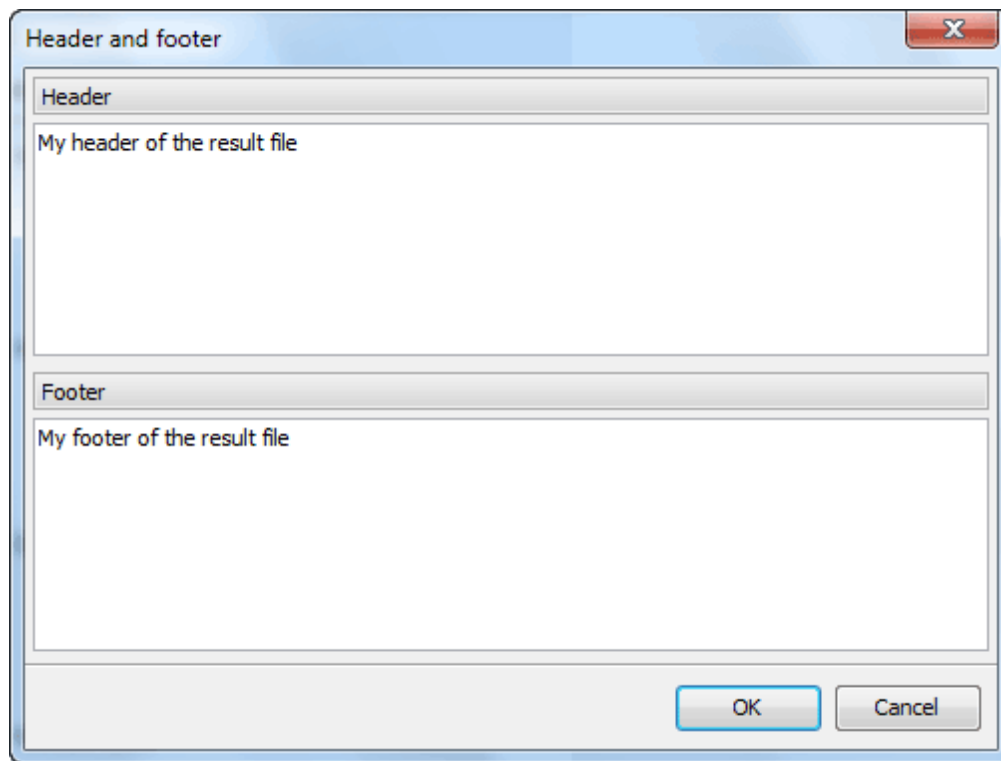
 Select or enter the result file name and specify the encoding if necessary. Hint: you can set default directory for data export in the Edit Database Profile dialog.

File name Encoding

C:\Data\Excel\Customers.xls ... ANSI ▼

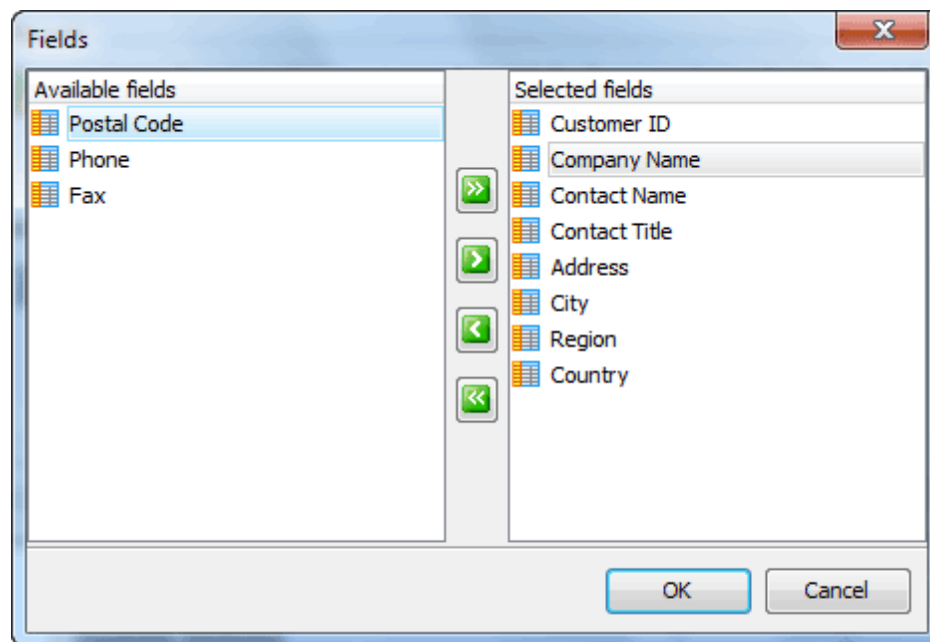
### 8.3.2 Setting header and footer

To specify the result file's header and footer, double click the corresponding button and complete fields of the [Header and Footer](#) window.



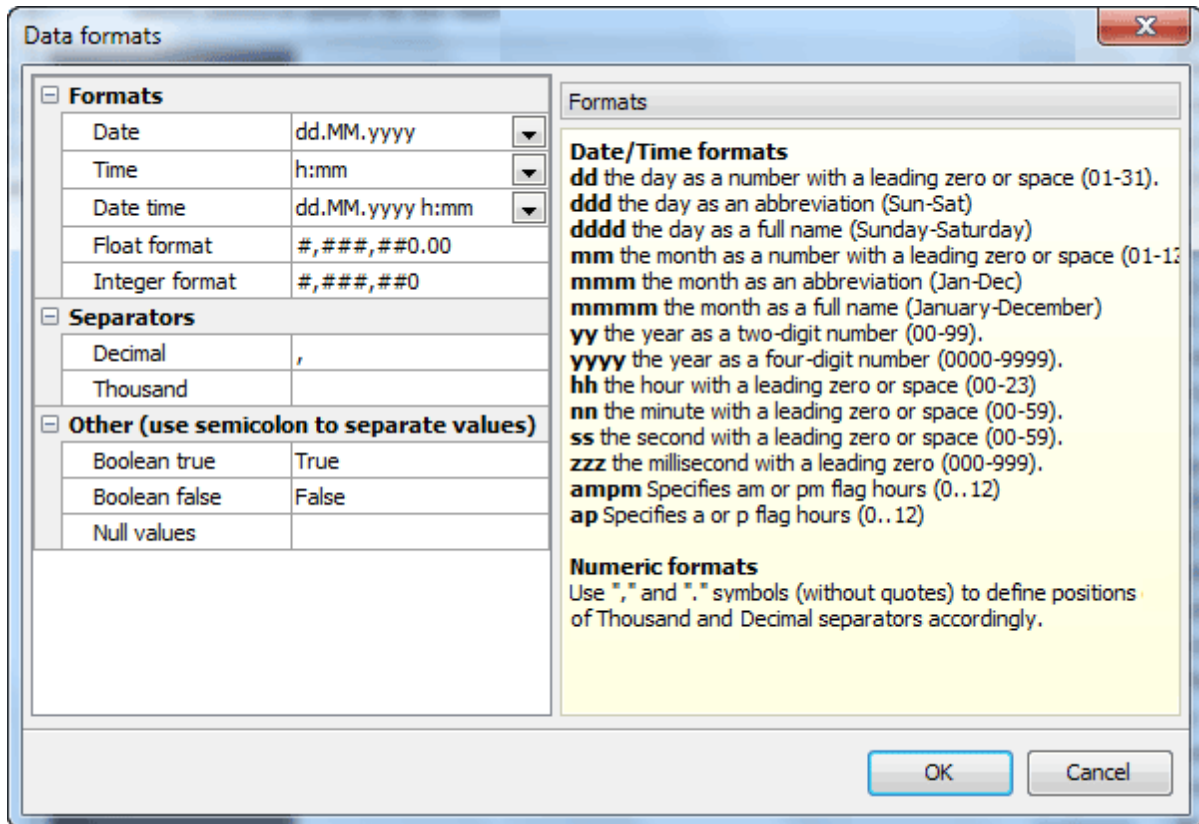
### 8.3.3 Selecting fields for export

Select table fields for export. If you leave all the fields in the [Available Fields](#) list, all fields will be exported except BLOBs.



### 8.3.4 Adjusting data formats

This step allows you to customize formats applied to exported data. Edit the format masks to adjust the result format in the way you need.



### 8.3.5 Setting format-specific options

Each format supposes corresponding additional export options. Use the wizard option to adjust export properties depending on the target file format you have selected earlier. The following formats are at your disposal: [Microsoft Excel](#)<sup>[231]</sup>, Microsoft Excel 2007, [CSV](#)<sup>[233]</sup>, [Text](#)<sup>[233]</sup>, [HTML](#)<sup>[232]</sup>, [XML](#)<sup>[233]</sup>, Microsoft Word, Microsoft Word 2007, Microsoft Access, OpenDocument Spreadsheet, OpenDocument Text, DBF, PDF, RTF, DIF, SYLK, and LaTeX.

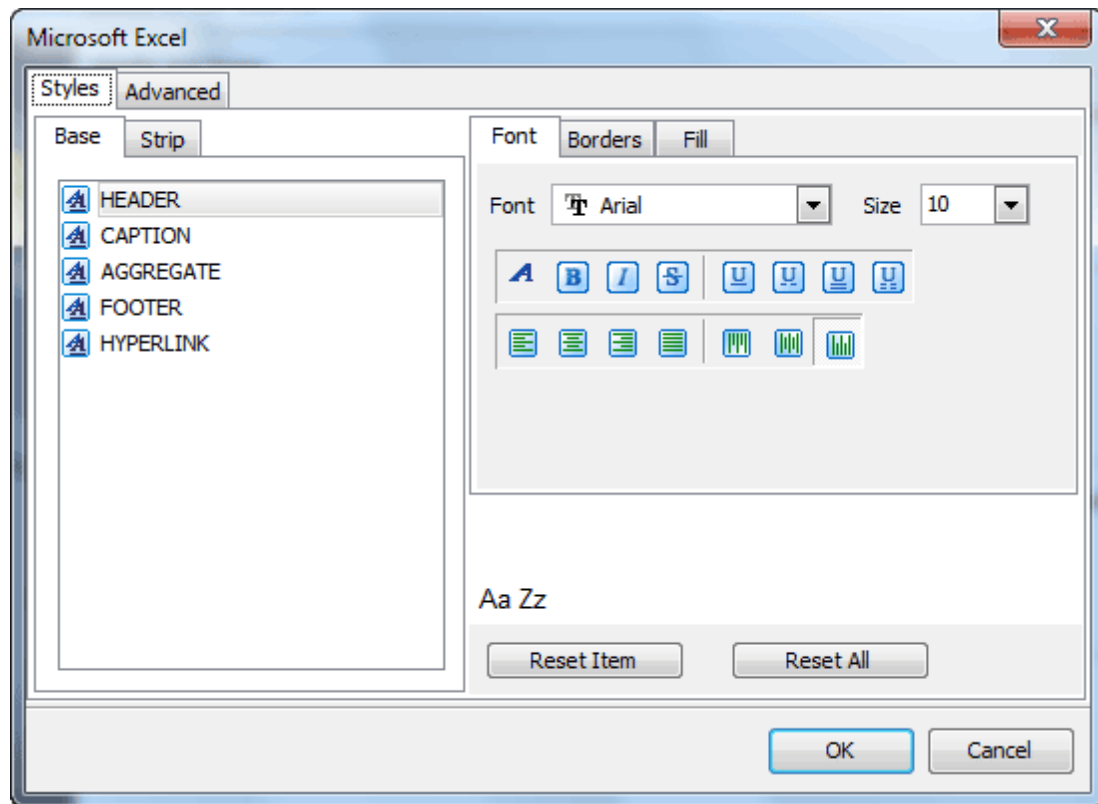
#### Microsoft Excel

The **Data Format** tab contains general options, which allow you to adjust the format for each kind of Excel cells. This means that you can specify such parameters as font, borders, filling color and method, etc. for each entity (such as data field, header, footer, caption, data, hyperlink and so on) separately. Also it is possible to create styles to make target Excel file be striped by columns or rows (the **Styles** tab).

The **Extensions** tab provides a possibility to add hyperlinks and notes to any cell of target file. Click the **Plus** button to add a new hyperlink or note to target Excel sheet and adjust its parameters. Click the **Minus** button to delete added hyperlink or note.

The **Advanced** tab allows you to define page header, page footer and title for target

Excel sheet.



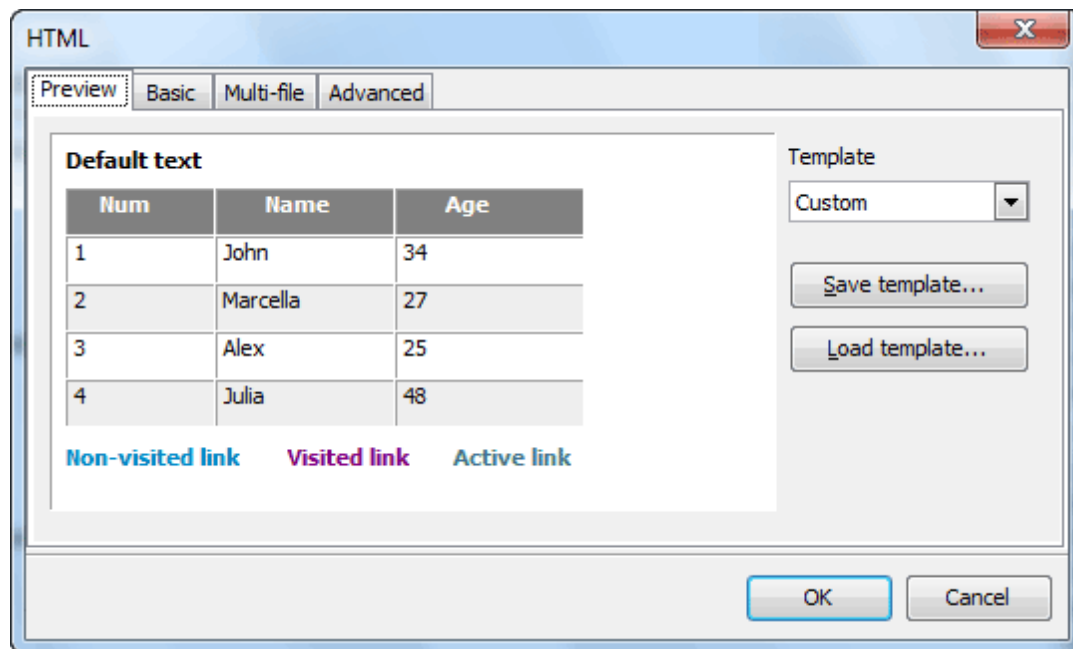
### HTML

The [Preview](#) tab allows you to select the style of HTML file from a number of built-in templates provided by the [Templates](#) combo box. You can choose any of these templates, customize it by clicking on objects in the preview panel, and save it as a custom template using the [Save template](#) button. Use the [Load template](#) button to load previously saved custom templates from hard disk.

The [Basic](#) tab allows you to specify basic parameters of target HTML file, such as its title, cascade style sheet options, etc.

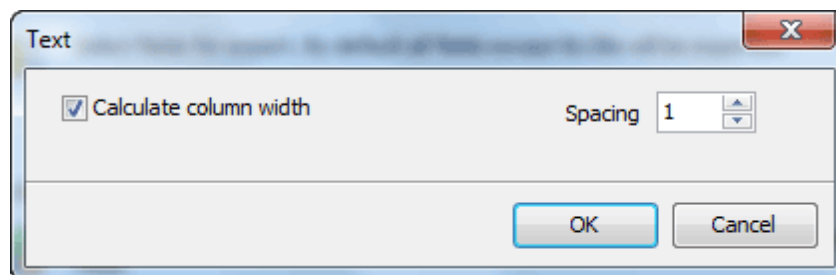
The [Multi-file](#) tab provides you with a possibility to split target HTML file into several separated files. This tab allows you to specify the record count for a single file, set an option to generate an index HTML file, and add an ability of navigation between each other to each of exported files.

The [Advanced](#) tab contains such HTML options as default font, background, cell padding and spacing, etc.



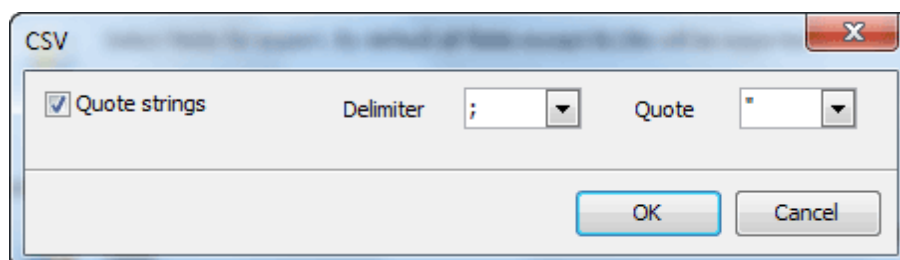
#### Text files

Set the [Calculate](#) column width options on if you want each column of target file to be adjusted to the maximum number of characters in it. The [Spacing](#) option specifies the number of spaces between columns in the target file.



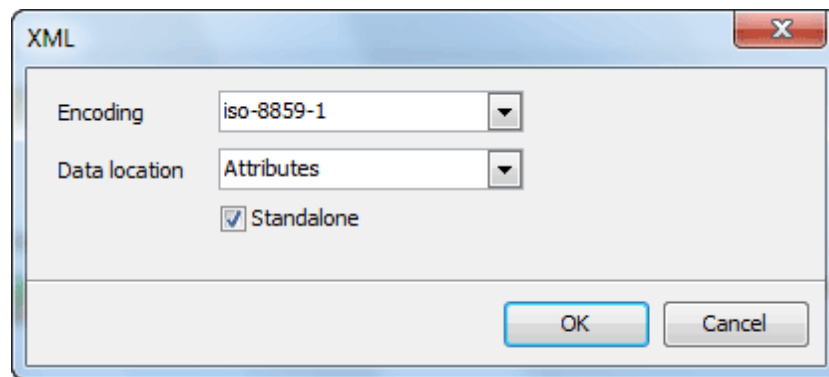
#### CSV files

You can specify column separator and optional values quote character for the target file on this step.



#### XML documents

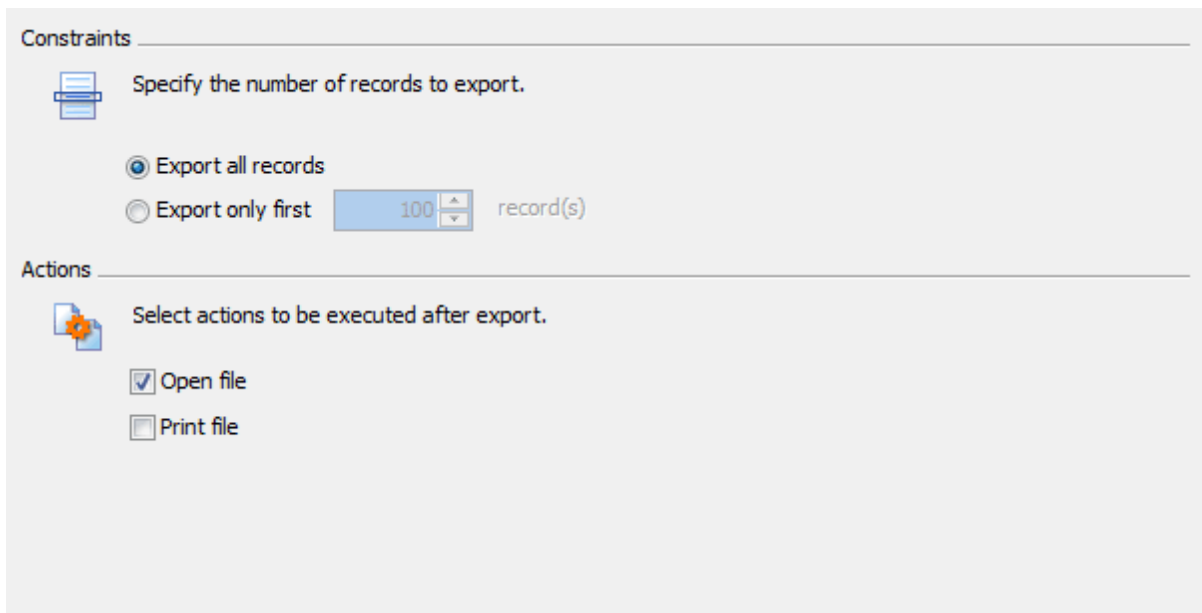
Specify XML document encoding in the [Encoding](#) edit box and set the Standalone option on if you wish the target document to be standalone.



### 8.3.6 Setting common export options

Use this step to specify options to be applied to all exported data:

- Select the number of records to be exported from each table: a fixed number or all records.
- Specify actions to be executed after the export. To open the result files in the associated program (MS Excel, Notepad, default browser, etc), check the [Open file](#) box. To send the result files to the default printer, use the [Print file](#) checkbox.



## 8.4 Get SQL Dump

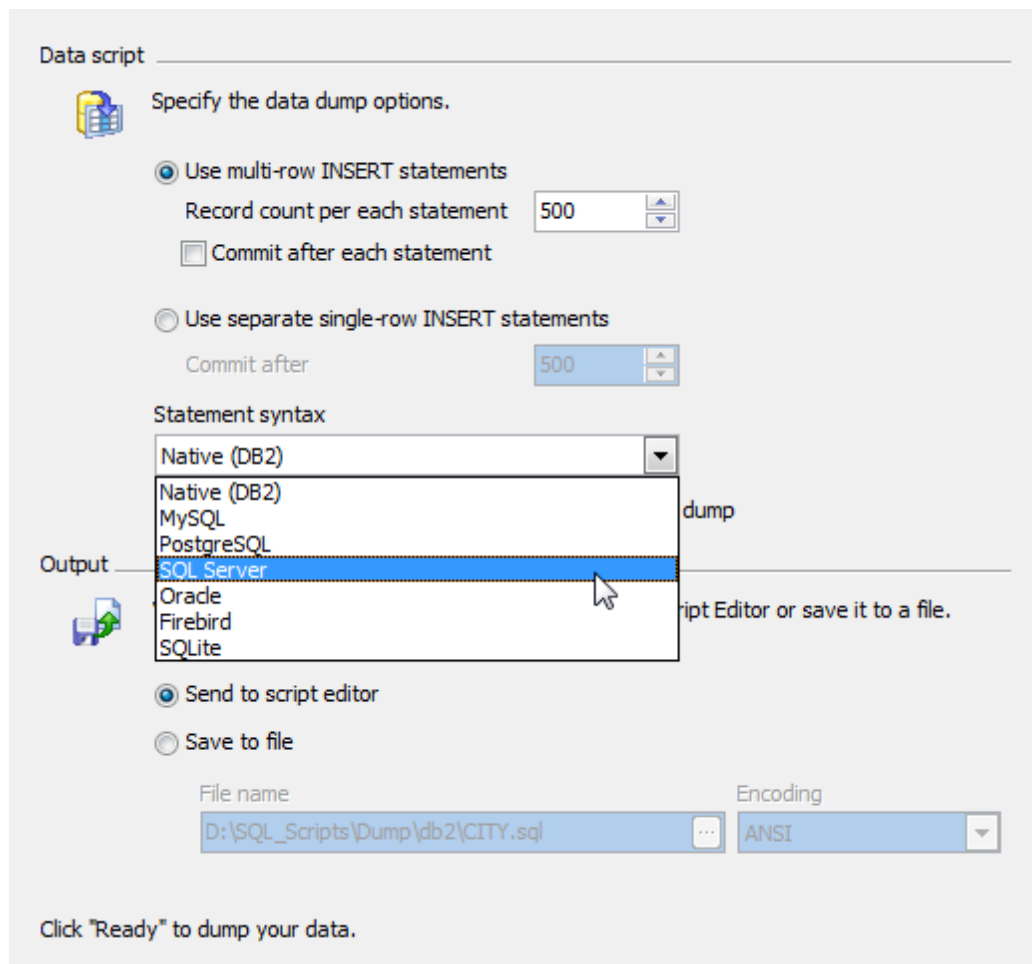
[Get SQL Dump Wizard](#) allows you to export data from a table or a query result to the SQL script as a number of INSERT statements.

In order to get a SQL dump from a table or a query:

- open the table in [Table Editor](#) or open and execute query in [SQL Editor](#) or [Query Builder](#);
- open the [Data](#) tab or the [Result](#) tab respectively;
- use the [Get SQL Dump](#) item of the [Navigation Bar](#).

- [Selecting fields to include in the result INSERT statement](#)<sup>[235]</sup>
- [Specifying dump options](#)<sup>[236]</sup>

**See also:** [Export Data Wizard](#)<sup>[228]</sup>, [SQL Script Editor](#)<sup>[248]</sup>



Data script

Specify the data dump options.

☒ Use multi-row INSERT statements  
Record count per each statement: 500  
☐ Commit after each statement

☐ Use separate single-row INSERT statements  
Commit after: 500

Statement syntax

Native (DB2)  
Native (DB2)  
MySQL  
PostgreSQL  
SQL Server  
Oracle  
Firebird  
SQLite

Output

☒ Send to script editor  
☐ Save to file

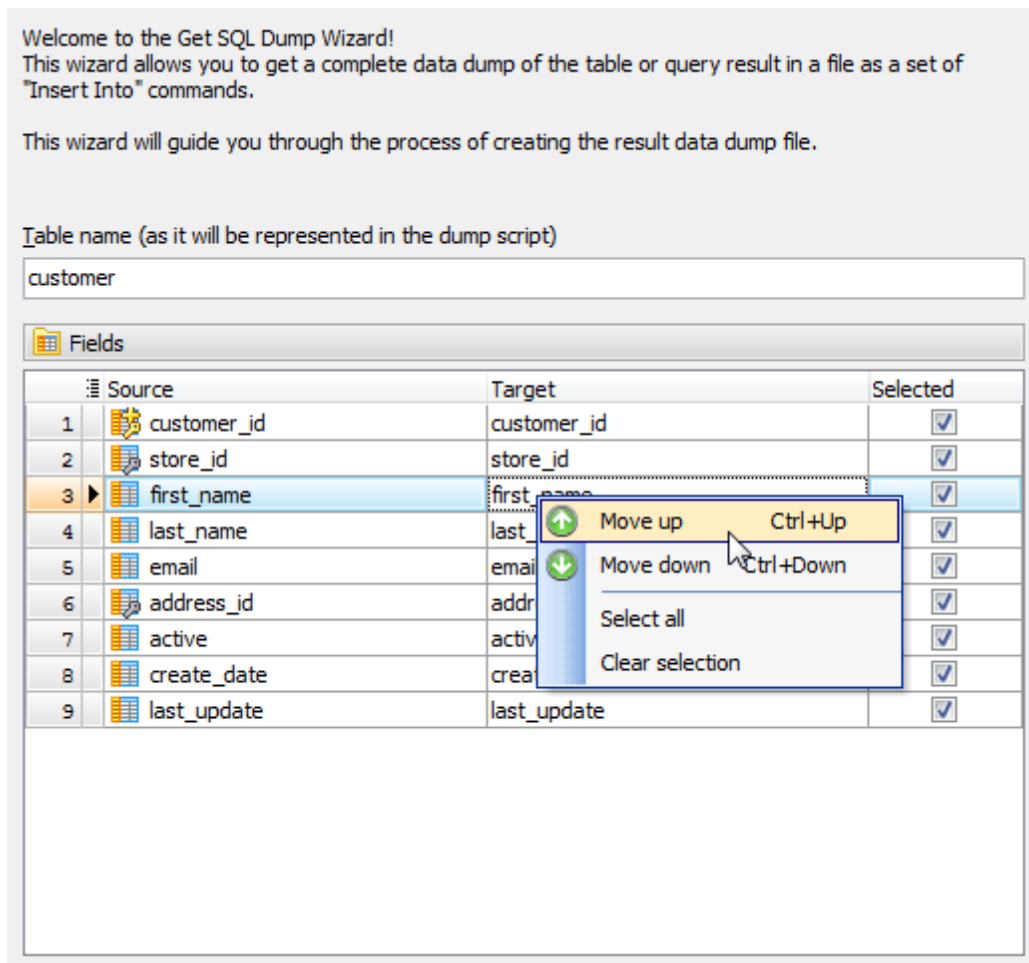
File name: D:\SQL\_Scripts\Dump\db2\CITY.sql  
Encoding: ANSI

Click "Ready" to dump your data.

### 8.4.1 Selecting fields

The first wizard step allows you to specify the table name as it will be included in the result script.

You can also select the fields to be included in the result *INSERT* statement. All the table fields are included into the [Selected fields](#) list by default. If you do not want some fields to be exported, move them back to the [Available fields](#) list.



### 8.4.2 Specifying dump options


Select the data dump mode to be used ([Multi-row INSERT statements](#) or [separate single-row INSERT statements](#)) and specify commits' frequency.

To [add the "CREATE TABLE" to the top of the dump](#), check the corresponding box.

Get SQL Dump Wizard allows you to send the result script to [SQL Script Editor](#)<sup>248</sup> or to save it to a specified file. Select the [Send to script editor](#) option to load the result to the editor. To save the result to the file, enter the script file name (\*.sql).

Click the [Ready](#) button to start the process.

Data script



 Specify the data dump options.

☒ Use multi-row INSERT statements  
Record count per each statement

☐ Commit after each statement

☐ Use separate single-row INSERT statements  
Commit after

Statement syntax

  Output

Native (DB2)  
Native (DB2)  
MySQL  
PostgreSQL  
SQL Server  
Oracle  
Firebird  
SQLite

☒ Send to script editor

☐ Save to file

File name  Encoding

Click "Ready" to dump your data.

## 8.5 Import Data Wizard

[Import Data Wizard](#) provides you with a graphical user interface to import data from the most popular files formats into existing DB2 tables. It allows you to adjust data formats, empty target tables, execute custom SQL scripts, etc.

Import Data tool supports:

- Microsoft Office Excel 95-2003
- Microsoft Office Excel 2007
- Microsoft Office Access
- Microsoft Office Access 2007
- Delimiter-separated values (CSV, DSV, TSV)
- DBF
- Text files
- XML
- ODBC data sources (any database accessible via an ODBC driver or OLE DB provider, such as SQL Server, MySQL, Oracle, MS Access, Sybase, DB2, PostgreSQL, etc.)

In order to run the wizard you should

- open the table in [Table Editor](#);
- go on to the [Data](#) tab;
- select the [Import Data](#) item from the [Navigation Bar](#).

To import data,

- [Set the format](#)<sup>[239]</sup> of the input data and the source file name;
- [Map source file columns and target table fields](#)<sup>[241]</sup>;
- [Specify other import options](#)<sup>[241]</sup>.

**Source format**

Select one of the available source formats.

- ☒ Microsoft Office Excel 97 - 2003
- ☐ Microsoft Office Excel 2007
- ☐ Microsoft Office Access
- ☐ Microsoft Office Access 2007
- ☐ Delimiter-separated values (CSV, DSV, TSV)
- ☐ Text file (Fixed-width columns)
- ☐ DBF
- ☐ XML
- ☐ ODBC data source

**Source file**

Select or enter the source file name and specify the encoding if necessary.

<b>File name</b>	<b>Password</b>	<b>Encoding</b>
D:\Data\Excel\employee.xls		ANSI
<b>Connection string</b>	<b>Identifier quote characters</b>	
	None (table_name)	
<b>Data source</b>	<b>Data location</b>	<b>Delimiter</b>
Employee_list	Attributes	
		<b>Quote</b>

**See also:** [Export Data Wizard](#)

## 8.5.1 Setting source file name and format

1. Select the format of the source file.
2. Specify the file you want to import. The file name extension in the **File name** box varies according to the selected import type. The wizard allows you to import data from several files at a time.

To import data from multiple files with the same structure, set the mask of the file names to the corresponding field. To see the list of matching files, use with the button on the right.

### Example 1:

Suppose, you need to import data from the following tables:

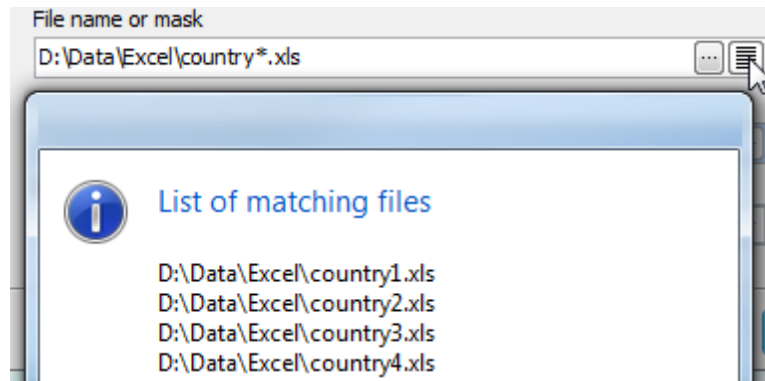
*D:\Data\Excel\country1.xls*

*D:\Data\Excel\country2.xls*

*D:\Data\Excel\country3.xls*

*D:\Data\Excel\country4.xls*

The mask for these file names is *D:\Data\Excel\country\*.xls*.



3. For ODBC data sources specify the [connection string](#) to be used to connect to the data source.
4. Select the data source to import: a table of MS Access database or a spreadsheet of MS Excel.
5. Enter the password to the database (MS Access).
6. For CSV file set the delimiter and quote characters.
7. Select source file [Encoding](#).
8. For .XML files, define the [XPath](#) to the data to be imported to the selected table and select whether data is stored in Attributes or in Subnodes.

### Example 2:

To import data from the following .xml file, use XPath=/*Employees/Employee* and Data location=*Subnodes*

```
<?xml version="1.0" encoding="utf-8"?>
<Employees>
  <Employee>
    <ID>1</ID>
    <FirstName>Klaus</FirstName>
    <LastName>Salchner</LastName>
    <PhoneNumber>410-727-5112</PhoneNumber>
  </Employee>
  <Employee>
    <ID>2</ID>
    <FirstName>Peter</FirstName>
    <LastName>Pan</LastName>
    <PhoneNumber>604-111-1111</PhoneNumber>
  </Employee>
</Employees>
```

### Example 3:

To import data from the .xml file below, use XPath=*DATAPACKET/Data/Item* and Data location=*Attributes*

```
<?xml version="1.0"?>
```

```
<DATAPACKETVersion="2.0">
<Data>
  <Item ID="1" FirstName="Klaus" LastName="Salchner" PhoneNumber="410-727-
5112" />
  <Item ID="2" FirstName="Peter" LastName="Pan" PhoneNumber="604-111-1111" />
</Data>
</DATAPACKET>
```

## 8.5.2 Setting the accordance between source and target columns

The wizard provides you with several ways to map input data to the target table columns.

- You can map columns automatically by order with the [Auto Fill](#) and [Auto fill all maps](#) buttons.
- You can do it manually using the drop-down list of [Source column](#) fields.
- To map columns visually, open [Map builder](#)<sup>[242]</sup> with the [Build map](#) link.

It's useful to save a specified map to a file for further using it in the next wizard sessions. To save a map, use the [More...](#) button and follow the [Save map](#) link.

To see the 100 first rows of input file or output table, use the [More...](#) button and follow the [View source data](#) or [Preview results](#) links respectively.

You can also specify [Replacements](#) to be applied to the selected column before the import and [data format masks](#)<sup>[244]</sup> used for the input file.

To exclude the first file row, use the [File contains column header](#) checkbox.

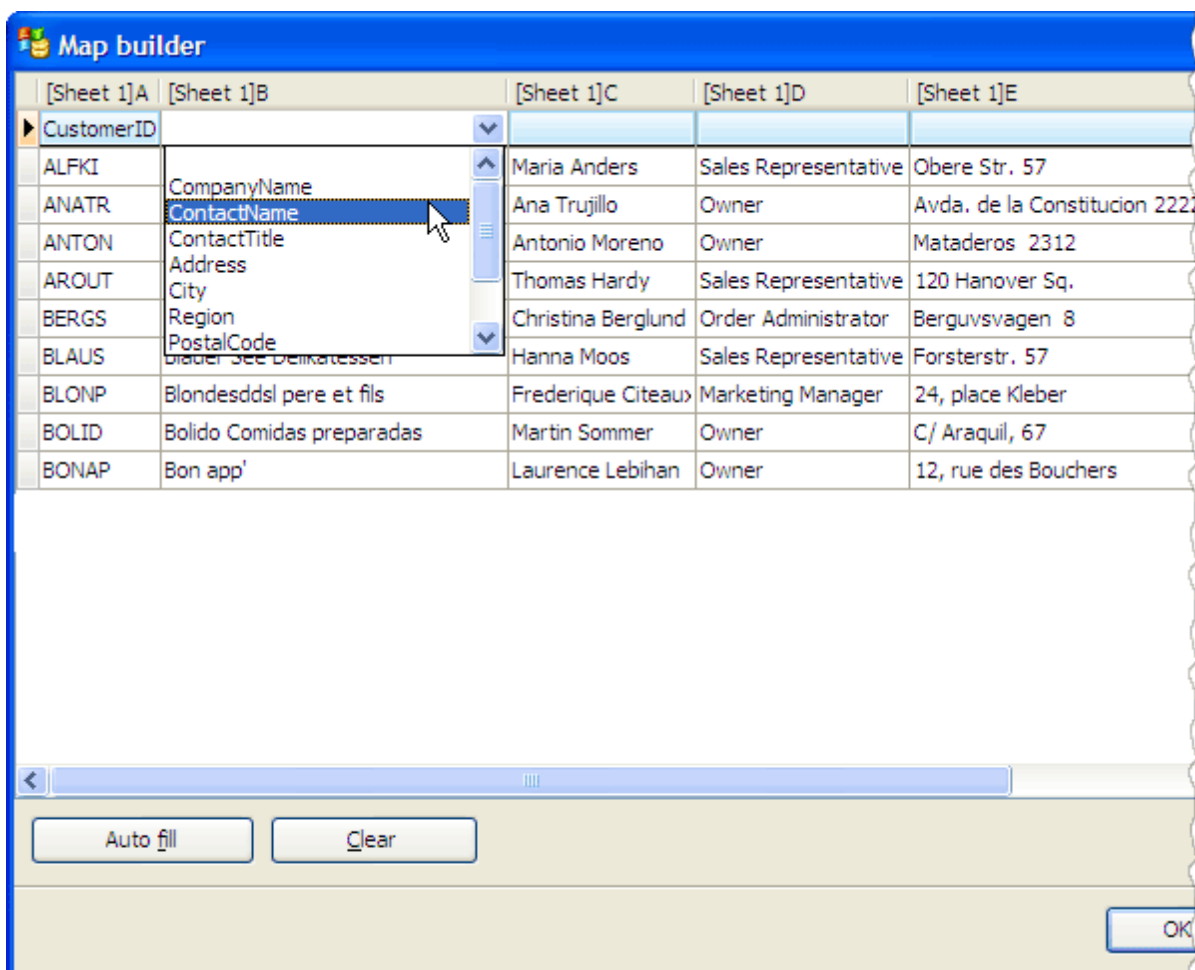
Columns

	Target field	Source column	Replacements
1	CustomerID	CustomerID	
2	CompanyName	CompanyName	
3	ContactName	ContactName	
4	ContactTitle	ContactTitle	
5	Address	Address	
6	City	City	
7	Region	Region	
8	PostalCode	PostalCode	
9	Country	Country	
10	Phone	Phone	
11	Fax	Fax	

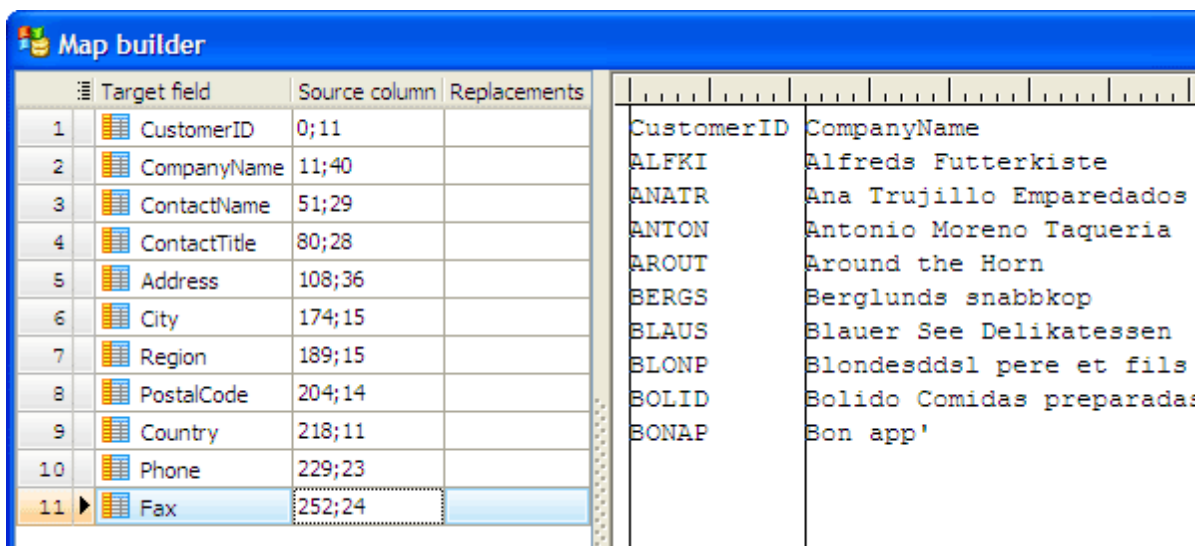
Auto fill Clear Build map... Define data formats... More...

### 8.5.2.1 Map builder

To specify the accordance between source and target columns visually, use popup menu of the upper row to map source file columns to target table fields.



For text files define columns bounds first. To add a bound, double-click near the column data in the builder area. To map a column to a target table field, select the field in the Target field list and then click between the bounds.



### 8.5.2.2 Data formats

Use the window fields to indicate format masks of the source data imported to the table. It allows the application to import data correctly.

The components of the date time format mask are represented at the window. Compose your date, time, and date time format mask of this components and separators. The following table contains some types of input fields and suggests masks to import them.

To import these input data correctly	Use these format masks
June 29	mmm dd
Jun 29, 2009	mmm dd, yyyy
Tue Jun 14 16:50:49	dddmmm dd hh:nn:ss
01/15/09 08:26 AM	mm/dd/yy h:nn ampm

You can also set decimal and thousand separators, and custom NULL, TRUE and FALSE values. If you have several values to be imported to NULL(TRUE, FALSE) value, use semicolons to separate them.

### 8.5.3 Customizing common options

On the wizard step you can set the number of records to import, whether the tool import all table records or only the specified number. In the second case you can set the number of records to skip.

#### Logging

This options group let you to manage logging of the import process.

#### Scripts

There are many cases where the import process is necessary to correct with additional

scripts. So to disable table indexes before the importing, specify the corresponding scripts to be executed before and after the process. The typical example of usage of the [Before each table](#) and [After each table](#) scripts is the import data to autoincrement columns of several tables. In this case it's necessary to set the corresponding scripts:

*SET IDENTITY\_INSERT %table\_name% ON*

and

*SET IDENTITY\_INSERT %table\_name% OFF*

to be executed before and after import data to each table correspondingly.

The screenshot shows the 'Records' section with a document icon and the text 'Specify the number of records to import.' Below this are three radio buttons: 'Import all records' (selected), 'Import only' (with a spinner box set to 100), and 'Skip' (with a spinner box set to 0). The 'Actions' section has a folder icon and the text 'You can empty the target table as well as execute SQL scripts before and/or after importing.' It includes an unchecked checkbox 'Empty target before importing' and a 'Customize scripts...' button. The 'Logging' section has a notepad icon and the text 'Enable this feature to get a detailed description of all the actions occurred during the import process.' It includes a checked checkbox 'Enable logging', a 'Log file name' label, a text box containing 'D:\Logs\Import\error.log', and a checked checkbox 'View the log file when a mistake occurs'.

**Records**

Specify the number of records to import.

☒ Import all records

☐ Import only 100 record(s)

Skip 0 record(s)

**Actions**

You can empty the target table as well as execute SQL scripts before and/or after importing.

☐ Empty target before importing

Customize scripts...

**Logging**

Enable this feature to get a detailed description of all the actions occurred during the import process.

☒ Enable logging

Log file name

D:\Logs\Import\error.log

☒ View the log file when a mistake occurs

## 9 Database Tools

**DB2 Maestro** provides a number of powerful tools for working with databases.

The following tools are available:

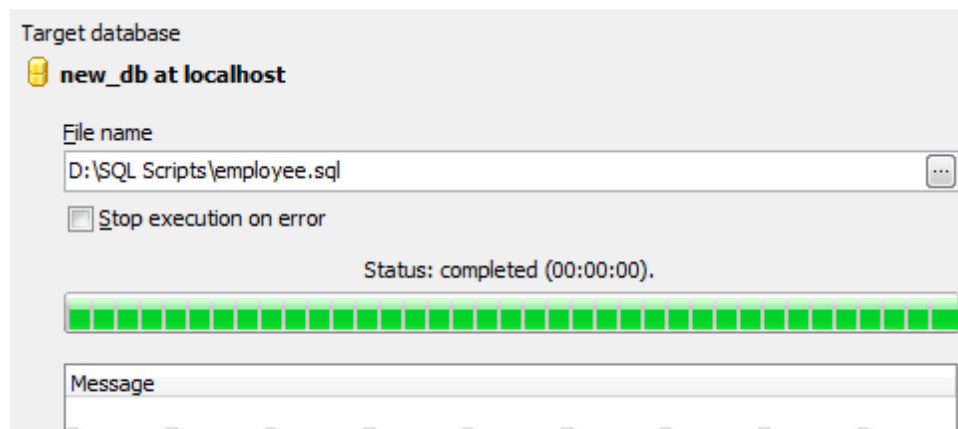
- **[SQL Editor](#)**<sup>[196]</sup>  
Creates and executes SQL queries.
- **[Visual Query Builder](#)**<sup>[202]</sup>  
Builds queries visually.
- **[Script Runner](#)**<sup>[247]</sup>  
Executes SQL scripts to the database.
- **[SQL Script Editor](#)**<sup>[248]</sup>  
Allows to edit and execute SQL scripts.
- **[Generate Database Report Wizard](#)**<sup>[250]</sup>  
Generates the database HTML or PDF report for structure of selected object in a whole or partially.
- **[BLOB Viewer](#)**<sup>[252]</sup>  
Displays a content of BLOB fields in different representations.
- **[Diagram Viewer](#)**<sup>[258]</sup>  
Represents data from a table or a query as a diagram in various ways.
- **[Data Analysis](#)**<sup>[262]</sup>  
Allows to slice and dice information efficiently according your business rules.
- **[Report Designer](#)**<sup>[267]</sup>  
Prepares data for reading, viewing, and printing in a polished look.
- **[Schema Designer](#)**<sup>[274]</sup>  
Allows to represent database tables and relationships as ER diagrams.
- **[SQL Generator](#)**<sup>[280]</sup>  
Provides you with a set of simple SQL statements.
- Simple tools for **[DML procedures](#)**<sup>[281]</sup> and **[Updatable views](#)**<sup>[283]</sup> generation allow to create a bunch of CRUD procedures automatically.

## 9.1 Script Runner

**Script Runner** is designed for executing of SQL scripts that don't require modifications. The window can be invoked from the **Tools** menu or with the **Execute script from file** link of **SQL Script Editor**<sup>[248]</sup>.

Script Runner allows to execute .sql files as well as archived scripts directly from .zip files. In case archived files this tool unpacks zip archives to temporary files by itself for further executing. The tool neither starts any implicit transactions before executing the script nor issues COMMIT or ROLLBACK commands after the executing.

To execute a script with Script Runner, set the file name and the **Stop execution on error** option value. This option allows to view all the execution errors (OFF). The specified script will be executed immediately on the database which name is represented at the top of the window.

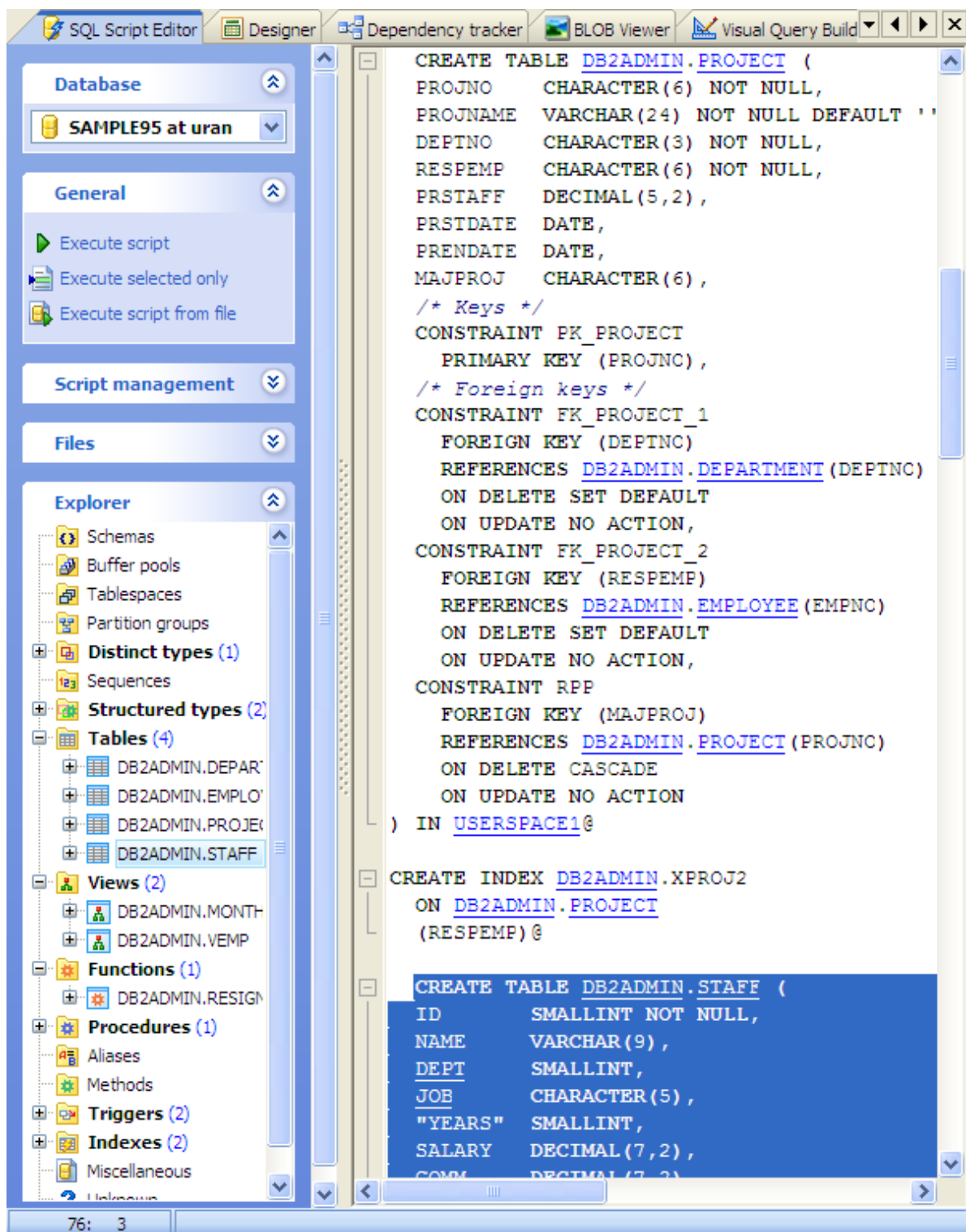


## 9.2 SQL Script Editor

[SQL Script Editor](#) is designed for SQL scripts editing and executing. The editor does not display results of SELECT queries. To work with such queries' data, use [SQL Editor](#)<sup>[196]</sup>. If you have a script that is ready to use, execute it with [Script Runner](#)<sup>[247]</sup>. To open [SQL Script Editor](#), select the [Tools | SQL Script Editor](#) main menu item.

To work with a script within SQL Script Editor, load it from an `.sql` file or type it in the editor area directly. To prevent mistakes in SQL syntax, the editor supports syntax highlighting, code completion and divides the script text into logical parts that can be individually collapsed or expanded (code folding). All the logical parts are represented at the [Explorer](#) at the [Navigation bar](#). It allows you to transfer to the proper script fragment quickly by clicking the corresponding node in the tree.

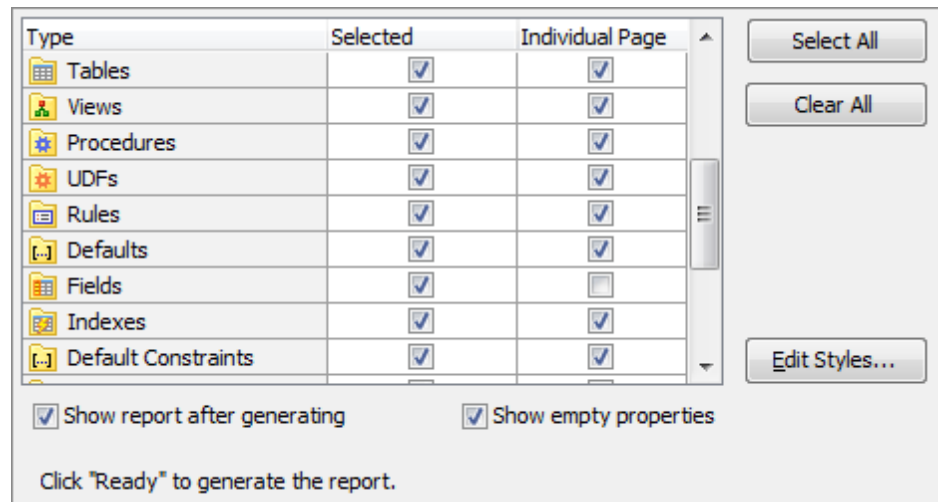
SQL Script Editor allows you to execute the whole SQL script or only its selected part. To make the executing of a large script much faster, execute the script directly from a file with [Script Runner](#)<sup>[247]</sup>. By default, if a user opens a file larger than 100K, SQL Script Editor will suggest him to execute the script file without opening it in the editor. This file size may be changed at the editor's [options](#)<sup>[300]</sup> tab.



## 9.3 Generate Database Report

With the help of [Generate Database Report Wizard](#) you can create HTML or PDF report for the structure of the selected object in the whole or partially. To run this wizard select the [Tools | Generate Database Report](#) main menu item.

- [Selecting reporting elements and setting other report options](#)<sup>[250]</sup>
- [Specifying reporting objects and editing styles](#)<sup>[250]</sup>



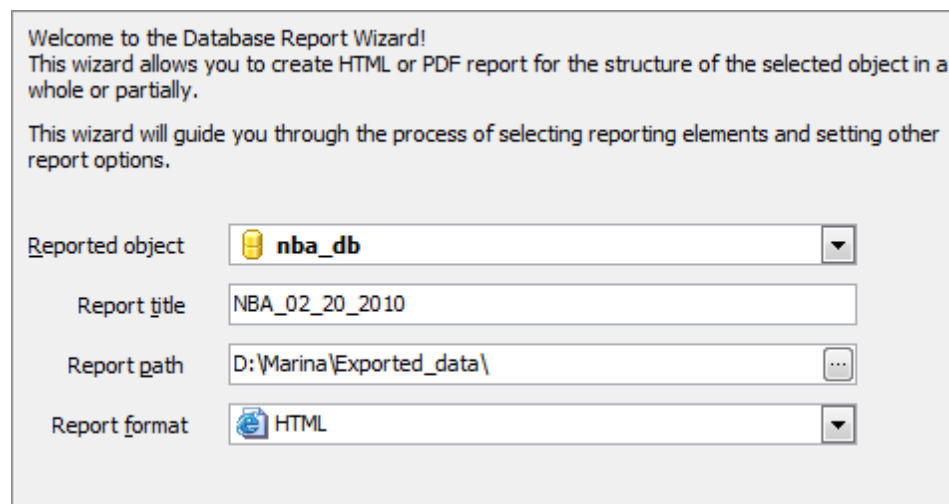
Type	Selected	Individual Page
Tables	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Views	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Procedures	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
UDFs	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Rules	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Defaults	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Fields	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Indexes	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Default Constraints	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

☒ Show report after generating
 ☒ Show empty properties

Click "Ready" to generate the report.

### 9.3.1 Selecting reporting elements and setting other report options

Select the [report object](#) and the [report format](#) items, set the [report title](#) and the [report path](#) options in the respective boxes.



Welcome to the Database Report Wizard!

This wizard allows you to create HTML or PDF report for the structure of the selected object in a whole or partially.

This wizard will guide you through the process of selecting reporting elements and setting other report options.

Reported object: nba\_db

Report title: NBA\_02\_20\_2010

Report path: D:\Marina\Exported\_data\ ...

Report format: HTML

### 9.3.2 Reporting objects and editing styles options

This step allows you to select the essential objects to report and to specify the output format and style using [Report Style Editor](#)<sup>[251]</sup>.

- ☒ Show report after generating

If checked, opens the result files in the associated program after making the report.

☒ **Show empty properties**

If checked, allows you to report objects even if they are empty.

Type	Selected	Individual Page
Tables	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Views	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Procedures	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
UDFs	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Rules	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Defaults	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Fields	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Indexes	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Default Constraints	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

☒ Show report after generating
 ☒ Show empty properties

Click "Ready" to generate the report.

Select All  
 Clear All  
 Edit Styles...

### 9.3.3 Editing database report style

Using [Report Style Editor](#) you can specify style properties of a report including font size, color and name for different elements.

**Report Style Editor**

Page Body  
 Database name  
 Page Title  
 Table  
 Table Header  
 Table Cell  
 Table Cell Image  
 Caption  
 Base Text  
 Definition  
 Comment  
 Copyright  
 Header Line  
 Param Table  
 Param Table Caption  
 Param Table Text  
 Link  
 Visited Link  
 Link Hover  
 Param Table "No Value"

**Style Properties**

Color:  

Font

Name: A Arial

Size: 8

Color:   Black

Border

Style: Solid

Color:

database

**Schema "public"**

Owner: admin

Authorization: (no value)

Tables base text

Table	Schema
table1	public

Generated by "SQL Maestro"

Restore Styles   Save As Default   Close   Help

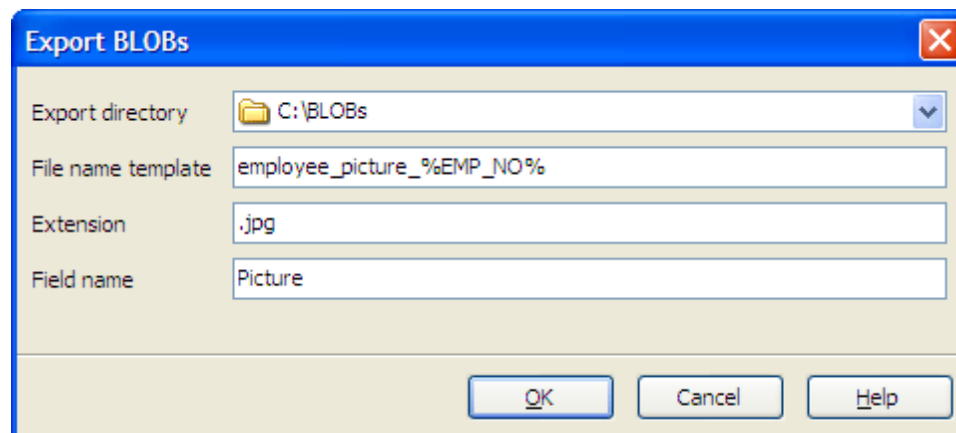
## 9.4 BLOB Viewer

BLOB Viewer allows you to view the content of the BLOB fields in various representations.

- [Viewing BLOB field as hexadecimal dump](#)<sup>[252]</sup>
- [Viewing BLOB field as plain text](#)<sup>[253]</sup>
- [Viewing BLOB field as graphical image](#)<sup>[254]</sup>
- [Viewing BLOB field as HTML](#)<sup>[255]</sup>
- [Viewing BLOB field as PDF](#)<sup>[256]</sup>

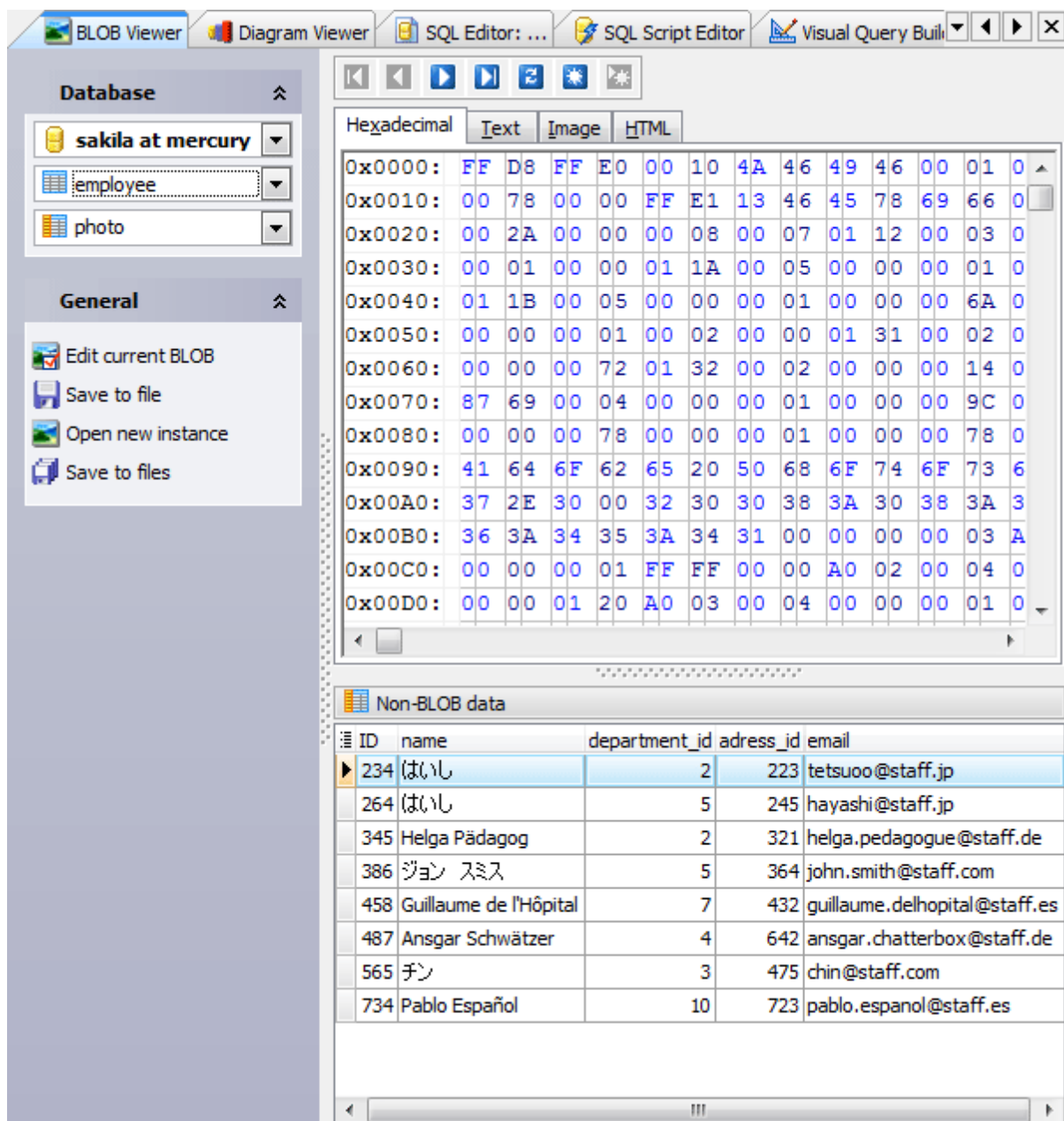
**See also:** [BLOB Editor](#)<sup>[223]</sup>

BLOB Viewer also allows you to save all BLOBs from a table or view to a given directory. Just click [Save to files](#) on the [Navigation bar](#) and fill all fields in the [Export BLOBs](#) window shown below. You can use table columns' names enclosed in % as a file name template.



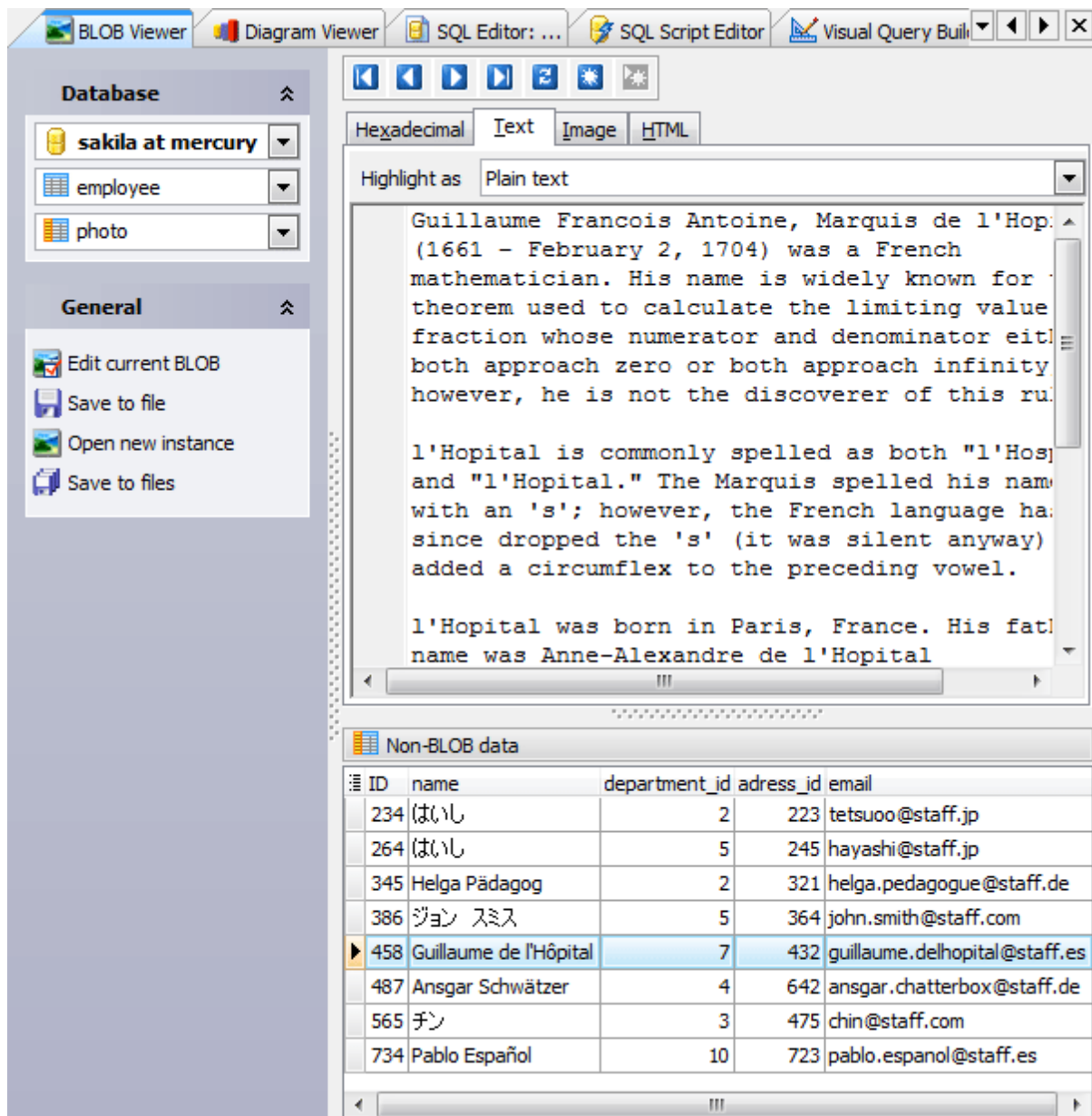
### 9.4.1 Viewing as hexadecimal dump

The [Hexadecimal](#) panel allows you to view data in hexadecimal mode.



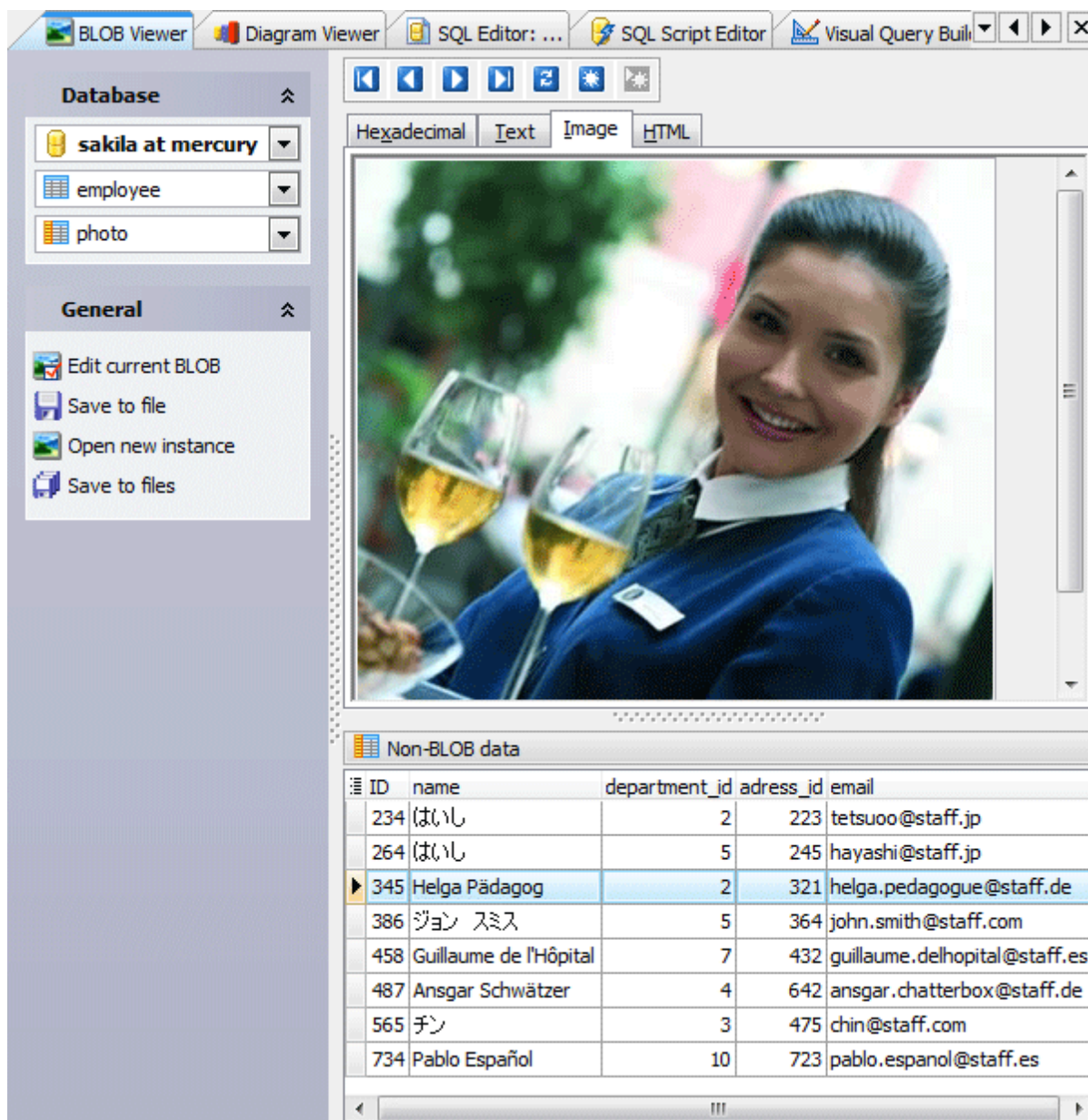
### 9.4.2 Viewing as plain text

The **Text** panel allows you to view data as simple text. For your convenience several types of text highlighting are available (*Plain text*, *HTML*, *JScript*, *CSS*, *PHP*, *XML*, *SQL*, and *SQLite DDL*). The popup menu of the panel provides you to **Find** or **Replace** a necessary text fragment.



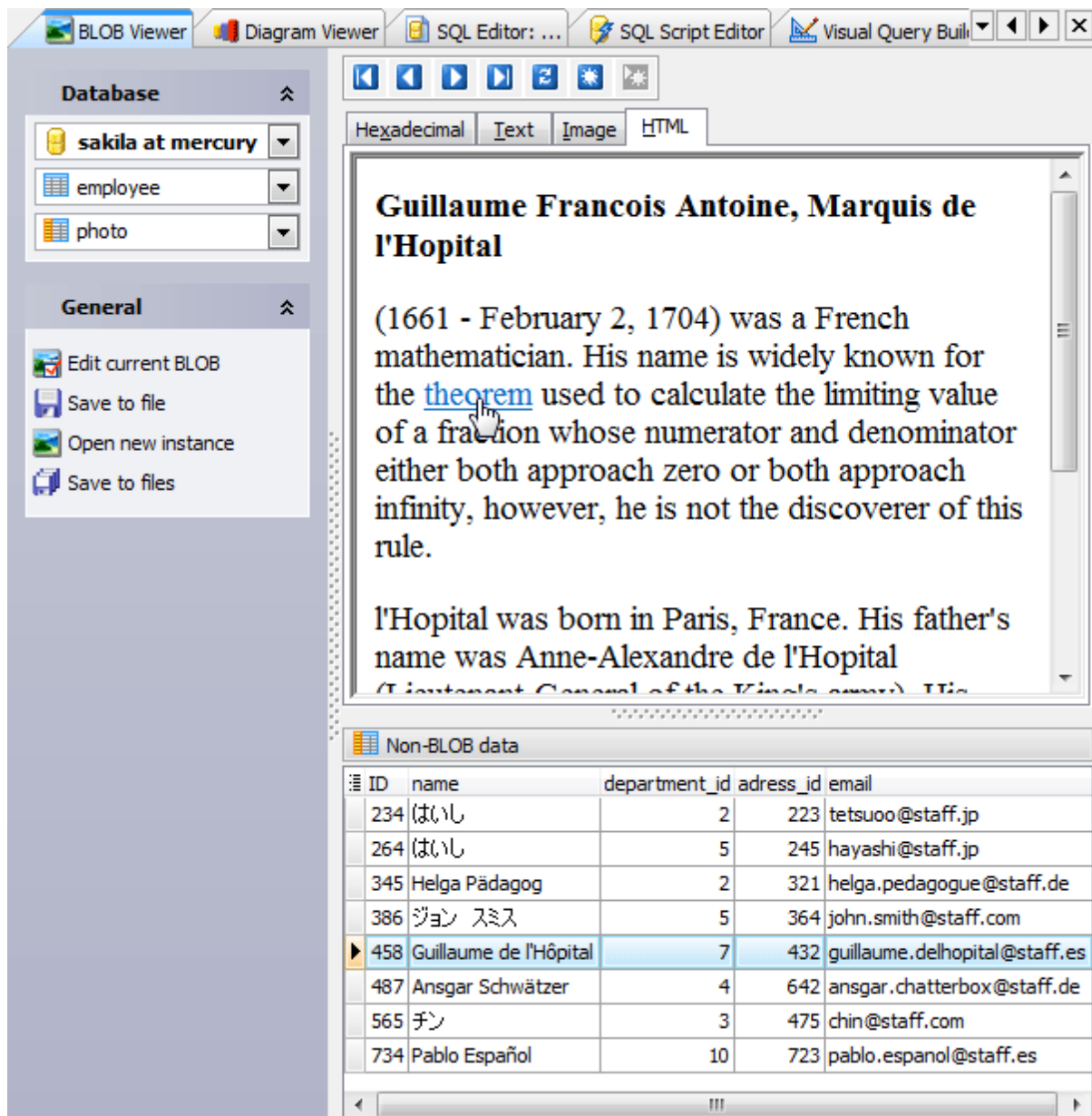
### 9.4.3 Viewing as image

The [Image](#) panel displays field data as image.



#### 9.4.4 Viewing as HTML

The [HTML](#) panel displays field data as HTML.



### 9.4.5 Viewing as PDF

The [PDF](#) panel allows you to browse PDF data stored in the database.

The screenshot displays the DB2 Maestro application window. The top menu bar includes options like BLOB Viewer, SQL Script Editor, Data Analysis, Visual Query Builder, and Designer. On the left, a sidebar shows the 'Database' section with 'test\_utf8 at d' selected, and the 'General' section with options like 'Edit current BLOB', 'Save to file', 'Open new instance', and 'Save to files'. The main area shows a document titled 'PostgreSQL PHP Generator Help' with a table of contents and a list of projects. A 'Connection properties' dialog box is open, showing options for connecting to the server.

**Database**

- test\_utf8 at d
- public.software
- manual

**General**

- Edit current BLOB
- Save to file
- Open new instance
- Save to files

**PostgreSQL PHP Generator Help**

## 2 Getting started

**Connection properties**  
Set the [connection parameters](#) for the connection with.

**Script connection properties**  
Specify here connection parameters for PostgreSQL. For example, if your webserver and PostgreSQL server are on the same host as localhost.

**Projects**  
When working with a project, all the session variables can be edited if necessary. To run a wizard, select a Project on the first wizard step and enter the project name. Projects are also available from this popup menu.

**Connection properties**

☒ I can connect to the server directly or via SSH tunneling  
[Configure SSH options](#)

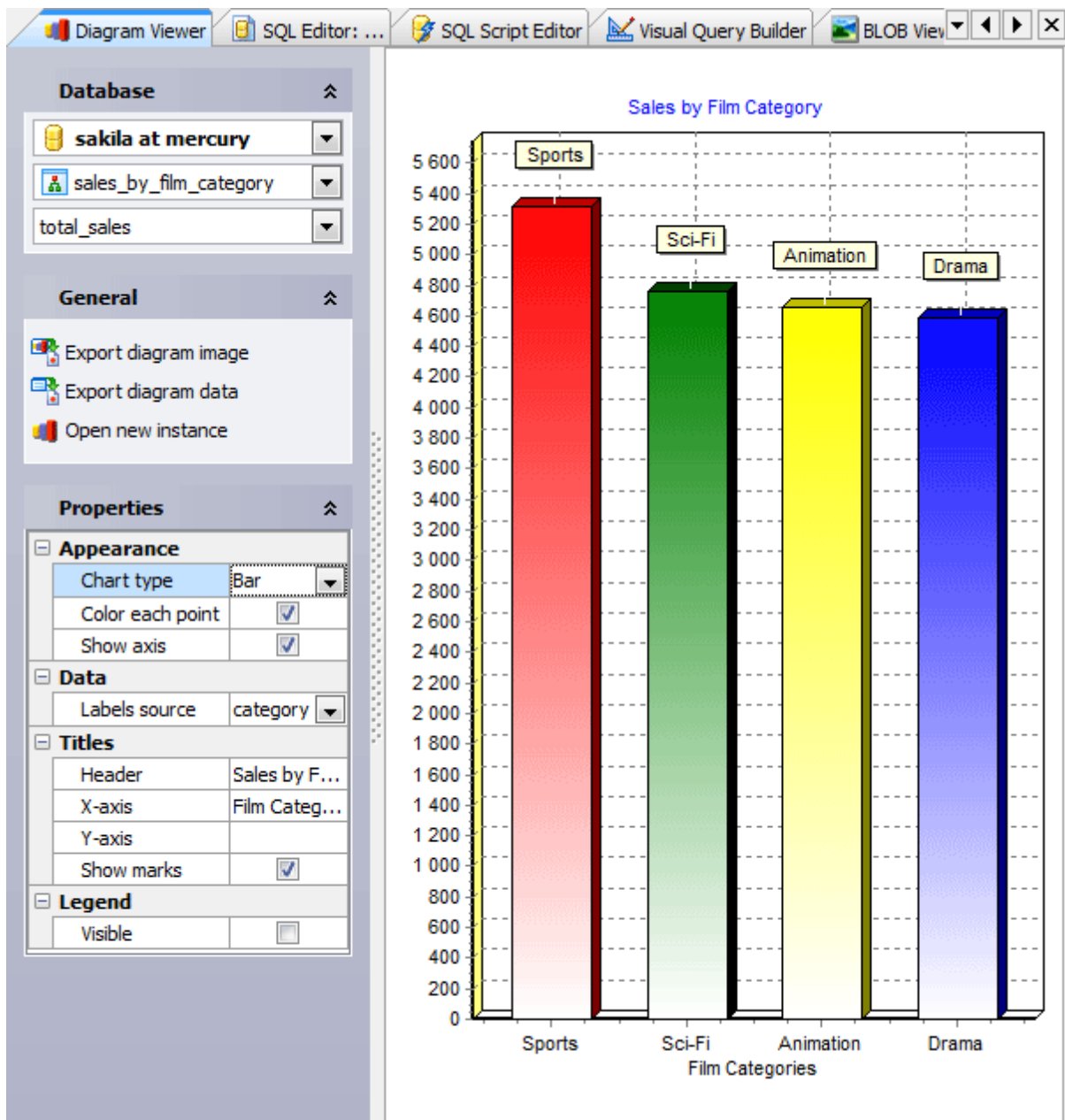
**Non-BLOB data**

id	full_name	description_id
1	PostgreSQL PHP Generator	1
2	Code Factory for MySQL	3
3	SQLite DataWizard	2
4	MS SQL Maestro	4

## 9.5 Diagram Viewer

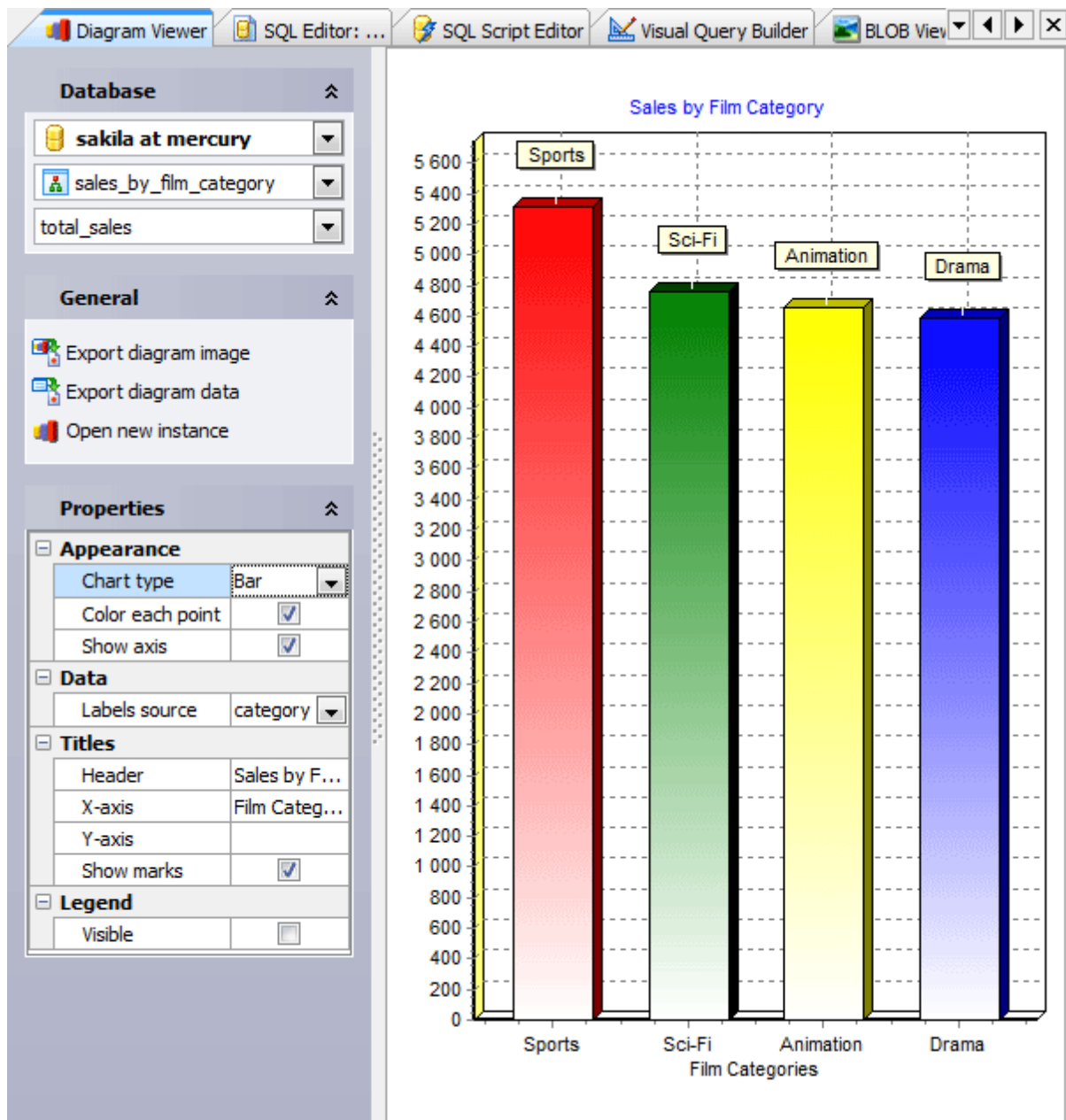
**Diagram Viewer** is a tool for representing data from a table or a query as a diagram in various ways. This means you can build a diagram represented as bars, lines, areas, points or pies, colored or not, with axis visible or not; specify axis labels source, diagram header and more. The **Diagram Viewer** also has the [Export diagram image](#)<sup>[260]</sup> and the [Export diagram data](#) features implemented, with a lot of formats supported.

- [Customizing diagram options](#)<sup>[259]</sup>
- [Exporting diagram as a graphical image](#)<sup>[260]</sup>



### 9.5.1 Customizing diagram properties

To build a diagram in [Diagram Viewer](#), you should select the source field(s) to be represented in the diagram first. Only numeric types of fields can be used in the diagram, and each selected field corresponds to a separated diagram series. Fields are selected by checking items in the third combo box from the top in the [Database](#) group of the [Navigation Bar](#). If the combo box is empty then either data source is not yet selected or it contains no numeric fields.



[Diagram Viewer](#) provides a special control for customizing the diagram properties. This control is located in the Properties group of the [Navigation Bar](#) and consists of four separate subgroups:

### Appearance

Contains properties responsible for major diagram appearance:

- **Chart type** - defines a way of how the diagram will be represented: as bars, lines, areas, points, pies, or fast lines
- **Color each points** - if checked, each bar, point, line or sector of the diagram has an individual color; if not checked, all the points are colored red
- **Show axis** - defines if the diagram has the axis and background grid or not

### Data

Contains the **Labels source** property which allows you to specify the field for X-axis labels as well as for diagram pointmarks .

### Titles

Contains properties for defining titles for different parts of the diagram:

- **Header** - defines the title appeared on the top of the diagram
- **X-axis** and **Y-axis** - define the titles for diagram axis
- **Show marks** - defines if the diagram point marks are visible or not

### Legend

The only **Visible** property of this subgroup specifies whether the legend rectangle should be represented on the right side of the diagram or not.

## 9.5.2 Exporting diagram image

**Diagram Viewer** provides an ability to export current diagram to a file as graphical image. This ability is constituted in **Export Diagram Wizard** which can be invoked by the **Export diagram image** item of the **Navigation Bar**.

Select the desired graphical format in the **Destination format** radio group and specify the file name in the **Destination file name** box.

Welcome to the Export Diagram Wizard!  
This wizard allows you to export diagram as an image in most popular formats, such as Windows metafile, BMP, PDF, WML, JPEG and much more.  
The wizard will guide you through the process of exporting diagram into the graphical file.

**Destination format**

☒ BMP    ☐ PDF    ☐ JPEG    ☐ GIF    ☐ Postscript  
☐ Metafile    ☐ VML    ☐ PNG    ☐ PCX

**Destination file name**

C:\Documents\Diagrams\sales\_by\_film\_category.bmp

Set the destination width and height by the corresponding spin edits. Check or uncheck the **Keep aspect ratio** option to keep the image ratio for exported image or not. Check the **Open exported image in associated program** option to view the image after the export is done.

Image size

Width  Height

☒ Keep aspect ratio

☐ Open exported diagram in associated program

Click "Ready" to export the diagram.

## 9.6 Data Analysis

**Data Analysis** is a tool to define a multidimensional model with analytic calculations to analyze information also called OLAP cube. Such cubes could effectively be re-oriented. So the tool allows you to view data in various ways, such as displaying all the cities down the page and all the products across a page and then immediately view it in another way. Because this re-orientation involves re-summarizing very large amounts of data, this new view of the data has to be generated efficiently to avoid wasting the analyst's time, i.e. within seconds, rather than the hours a relational database and conventional report-writer might have taken. It allows you to focus on business rules rather than creating dozens and dozens of reports. To run Data Analysis, choose [Tools | Data Analysis](#) main menu item.

To get an OLAP cube:

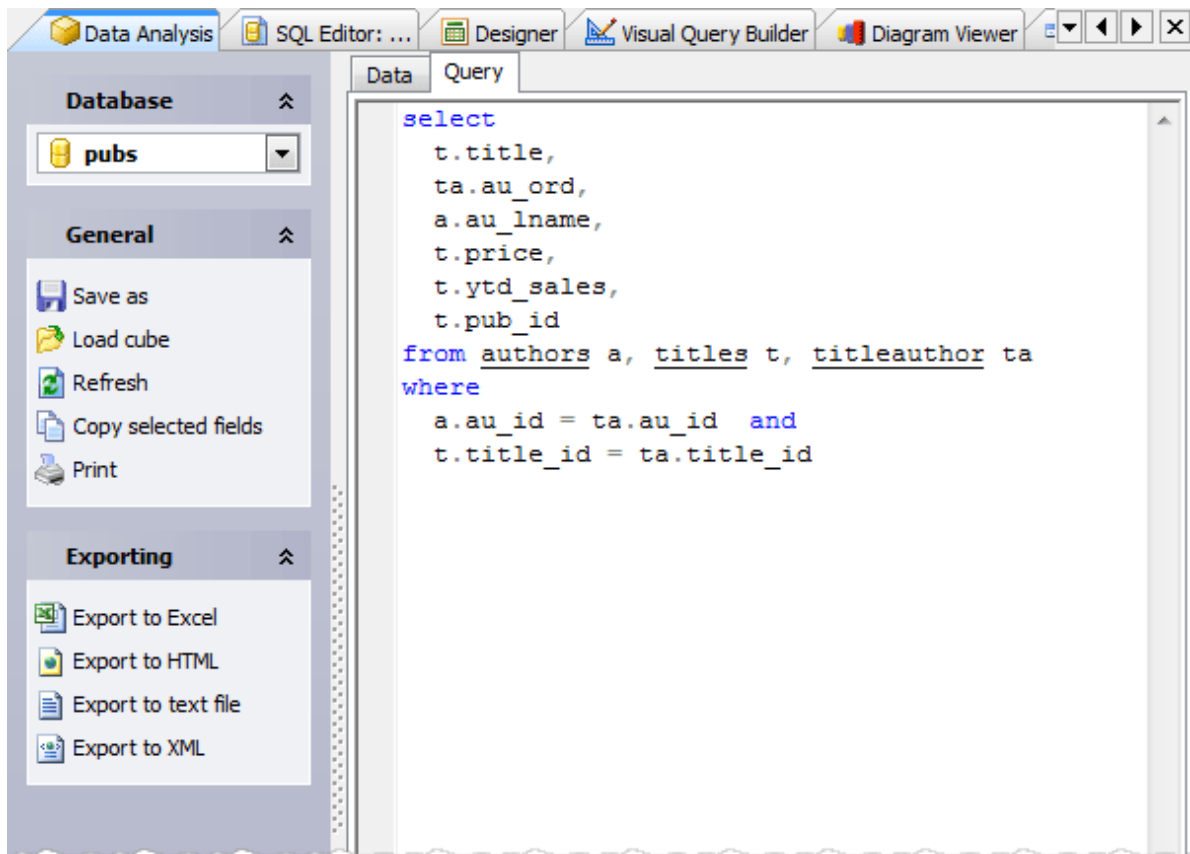
- [Input SELECT query](#)  in the [Query](#) window or load it from the .cub file.
- [Manage report data](#)  in the [Data](#) window.

The Data Analysis tool allows you to save the created OLAP cube to .cub file, print it, or export it to Excel, HTML, XML, and text file.

Data		Query			
au ord		vtd sales			
price		pub id			
au lname	title	0736	0877	1389	Grand Total
<input type="checkbox"/> Bennet	The Busv Executive's Database Guide			19,99	19,99
<input type="checkbox"/> Blotchett-Halls	Fiftv Years in Buckingham Palace Kitchens		11,95		11,95
<input type="checkbox"/> Carson	But Is It User Friendly?			22,95	22,95
<input type="checkbox"/> DeFrance	The Gourmet Microwave		2,99		2,99
<input type="checkbox"/> Dull	Secrets of Silicon Valley			20,00	20,00
<input type="checkbox"/> Green	The Busv Executive's Database Guide			19,99	19,99
	You Can Combat Computer Stress!	2,99			2,99
Green Total		2,99		19,99	22,98
<input type="checkbox"/> Grindlesbv	Sushi. Anyvone?		14,99		14,99
<input type="checkbox"/> Hunter	Secrets of Silicon Valley			20,00	20,00
<input type="checkbox"/> Karsen	Computer Phobic AND Non-Phobic Individuals: Beh		21,59		21,59
<input type="checkbox"/> Locksley	Emotional Security: A New Aloorithm	7,99			7,99
	Net Etiquette				
Lockslev Total		7,99			7,99
<input type="checkbox"/> MacFeather	Computer Phobic AND Non-Phobic Individuals: Beh		21,59		21,59
	Cooking with Computers: Surreptitious Balance Sh			11,95	11,95
MacFeather Total			21,59	11,95	33,54
<input type="checkbox"/> O'Leary	Cooking with Computers: Surreptitious Balance Sh			11,95	11,95
	Sushi. Anyvone?		14,99		14,99
O'Learv Total			14,99	11,95	26,94
<input type="checkbox"/> Pantelev	Onions. Leeks. and Garlic: Cooking Secrets of the		20,95		20,95
<input type="checkbox"/> Ringer	Is Ander the Enemy?	21,90			21,90
	Life Without Fear	7,00			7,00
	The Gourmet Microwave		2,99		2,99
Ringer Total		28,90	2,99		31,89
<input type="checkbox"/> Straight	Straight Talk About Computers			19,99	19,99
<input type="checkbox"/> White	Prolonned Data Deprivation: Four Case Studies	19,99			19,99
<input type="checkbox"/> Yokomoto	Sushi. Anyvone?		14,99		14,99
<input type="checkbox"/> del Castillo	Silicon Valley Gastronomic Treats		19,99		19,99
Grand Total		59,87	147,02	146,82	353,71

### 9.6.1 Input SELECT query

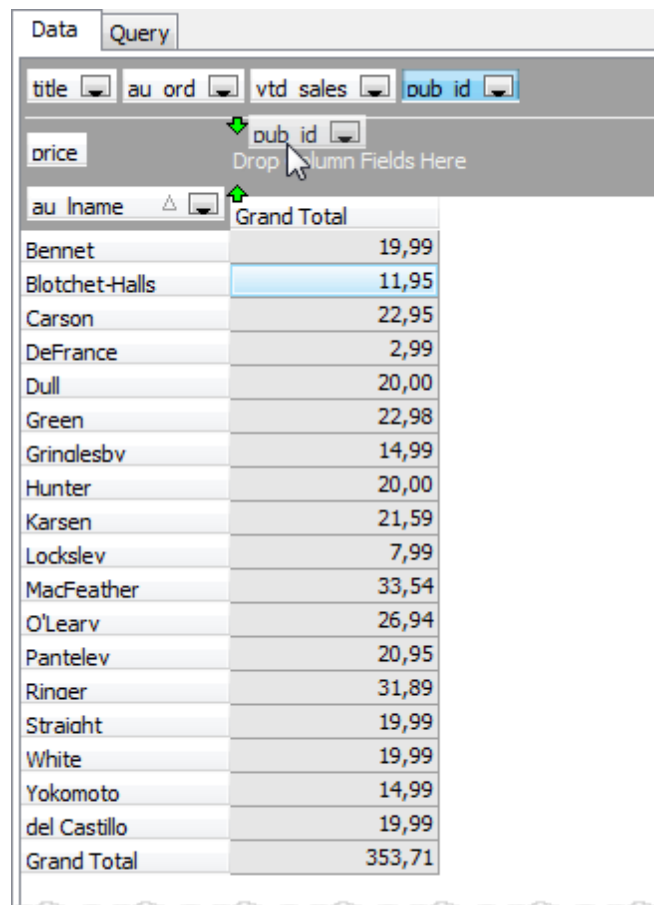
To get an [OLAP cube](#)<sup>[264]</sup>, enter SELECT query as a snowflake schema, represented by centralized fact tables (with numeric data) which are connected to multiple dimensions (the numeric data to be categorized by). Input the query text in the SQL Editor area directly or use "drag-n-drop" operation [SQL Editor](#) or Visual Query Builder areas and the Query tab of Data Analysis.



### 9.6.2 Managing report data

The [Data](#) tab allows you to manipulate the created OLAP cube appearance. At the beginning all the [query](#)<sup>[263]</sup> columns are arranged at the top of the tab. Put them according to your business rules: drag numeric columns to be filtered and summarized corresponding to the chosen columns and rows to the [Data Fields](#) area; place necessary columns to [Column Fields](#) / [Row Fields](#) areas respectively.

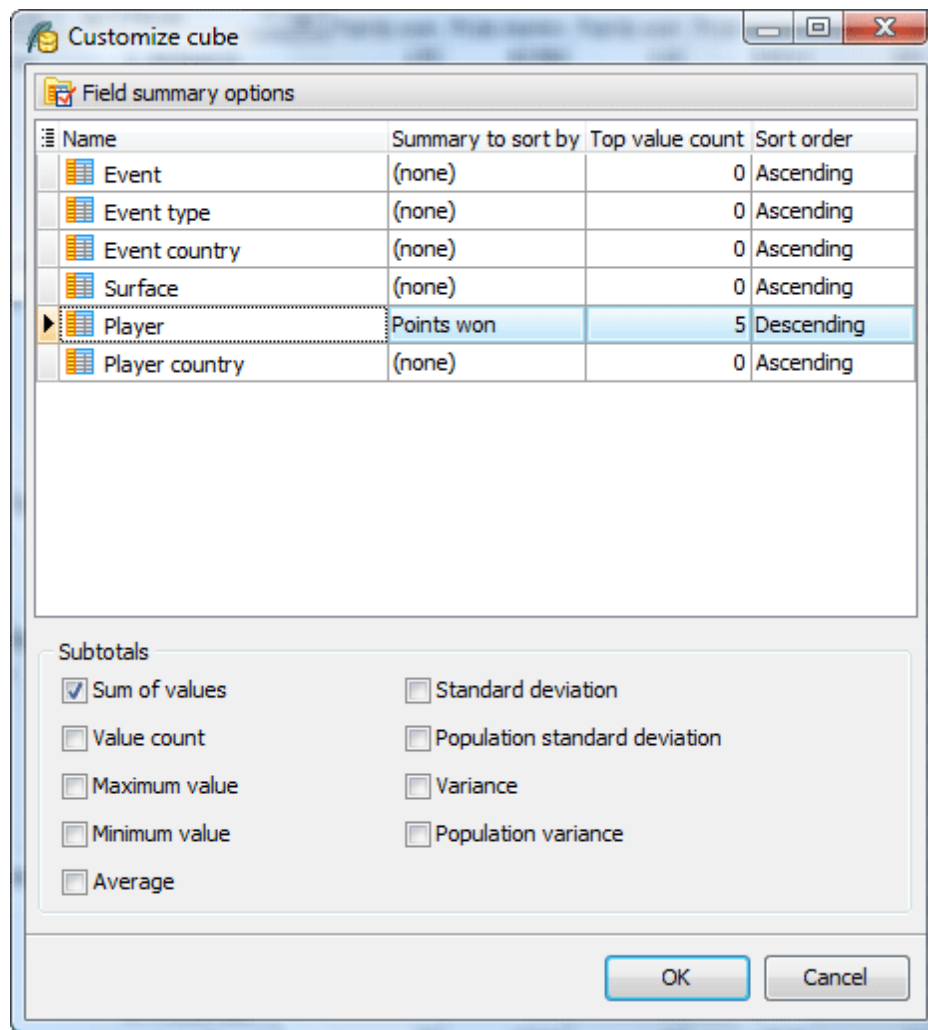
**Note:** Use for the Data Fields area only numerical columns.



The screenshot shows the 'Query' tab of the DB2 Maestro interface. At the top, there are input fields for 'title', 'au ord', 'vtd sales', and 'pub id'. Below these, a 'price' field is visible. A green arrow indicates the 'pub id' field is being dragged into the 'Drop Column Fields Here' area. The main data grid has two columns: 'au lname' and 'Grand Total'. The grid contains 20 rows of author names and their corresponding prices, followed by a 'Grand Total' row.

au lname	Grand Total
Bennet	19,99
Blotchet-Halls	11,95
Carson	22,95
DeFrance	2,99
Dull	20,00
Green	22,98
Grindlesbv	14,99
Hunter	20,00
Karsen	21,59
Lockslev	7,99
MacFeather	33,54
O'Learv	26,94
Pantelev	20,95
Ringer	31,89
Straight	19,99
White	19,99
Yokomoto	14,99
del Castillo	19,99
Grand Total	353,71

To set the aggregates calculated on the numeric columns, use the [Customize cub](#) window opened with the corresponding link at the Navigation bar. The window provides you also with an ability to specify columns the summary to be sorted by, the sort order and the max number of records represented in grid.



## 9.7 Report Designer

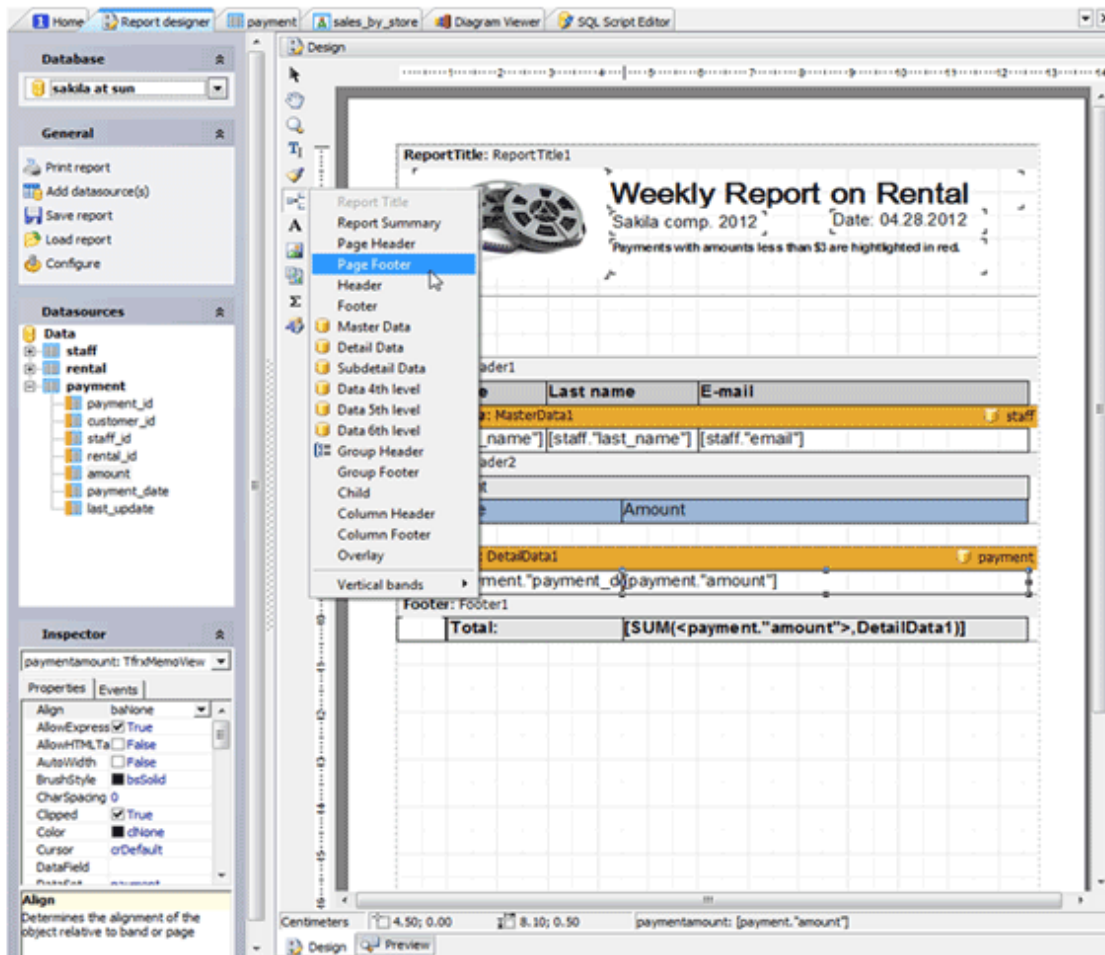
**Report Designer** allows you to create database reports, define reports appearance in your own style, equip it with master-detail data views, aggregate functions, and images and control the result with the ability of simultaneous previewing. To run Report Designer, choose [Tools | Report Designer](#) main menu item.

To create a report, you need to:

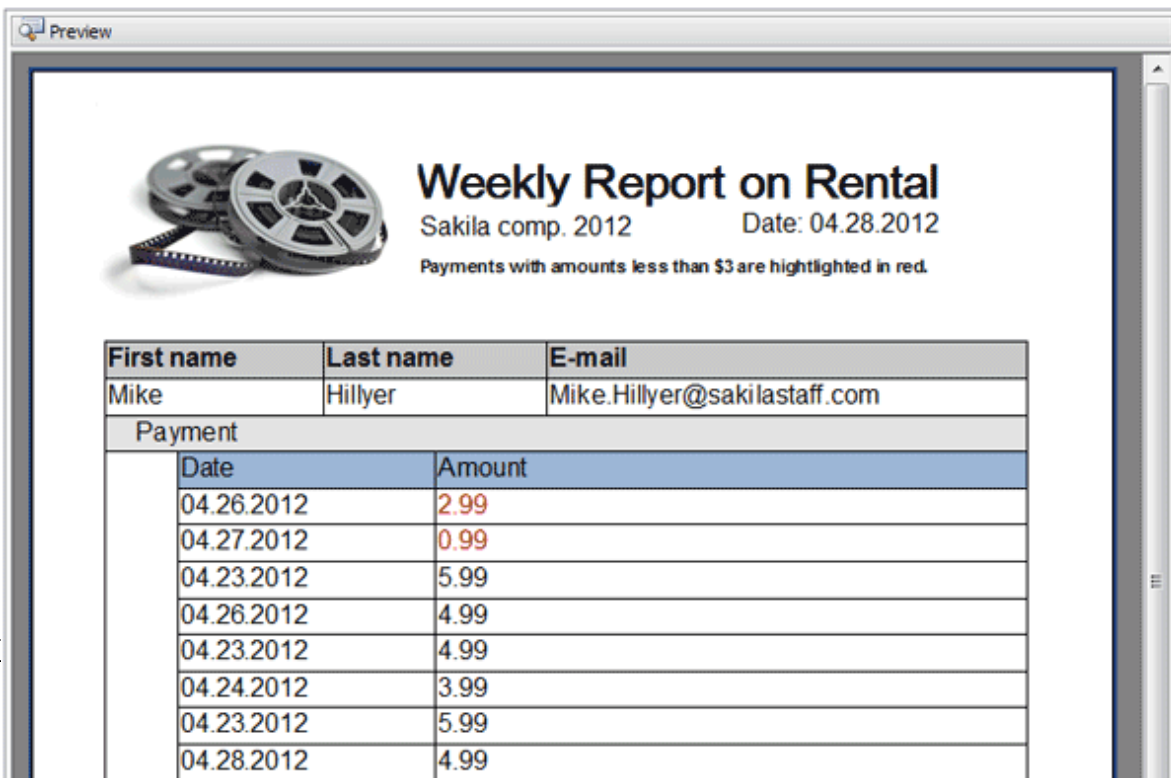
- [specify data sources](#)<sup>[269]</sup> for the data to be used in the report;
- [add all necessary objects](#)<sup>[269]</sup> to the report template;
- set the objects' format within the [Inspector window](#)<sup>[271]</sup>.

The prepared report pages are available immediately at the [Preview](#) window where you can browse it, save to it an .smr file, or print.

## Report Designer in Action



## Report Preview



### 9.7.1 Designer Tools and Objects

A blank report is presented as a paper page. At any place on the page, a user is able to add objects, which can display different information (such as text and/or graphics), as well as to define report's appearance. There is a possibility to use rulers and a grid with a specified size in the Design tab. To enable/disable these options, follow the [Configure](#) link at the [Navigation Bar](#) and check the corresponding boxes.

#### Datasources

To use content of a table (view) column data in a report,

- check the necessary database is selected as Database at the Navigation Bar;
- drag the table which data to be used in the report to the Datasources pane at the Navigation Bar;
- drag the necessary column from this pane and drop it to the necessary location on a report page.

#### Designer tools:

##### Select tool

The standard tool to select objects, modify their sizes, etc.

##### Hand tool

The tool allows dragging a report page.

##### Zoom tool

When the button is pressed, clicking on the left button doubles the zoom (adds 100%), while clicking the right one zooms out by 100%. When holding the left mouse button while dragging, the selected area would be zoomed.

##### Edit text tool

Clicking on the text object allows editing its contents right on the report page. If you hold the left mouse button when moving the cursor, the text object appears in the selected place, and then its editor launches.

##### Copy format tool

The button becomes enabled when the text object is selected. When clicking on the text object with the left button, it copies formatting, which has the previously selected text object, into the object.

#### Available objects:

[Band objects](#) allow to specify where, when, and how to display data and information in reports. Bands are used for logically placing the objects it contains at a location on the output page. Insert Band adds an area with definite behaviour according to its type such basic bands as Header, Footer, Title, and Summary, and databands whose allow to print data from database tables such as Master Data, Detail Data, etc.

[Text object](#) displays one or several text lines within the rectangular area.

[Picture object](#) displays a graphic file in BMP, JPEG, ICO, WMF, or EMF format.

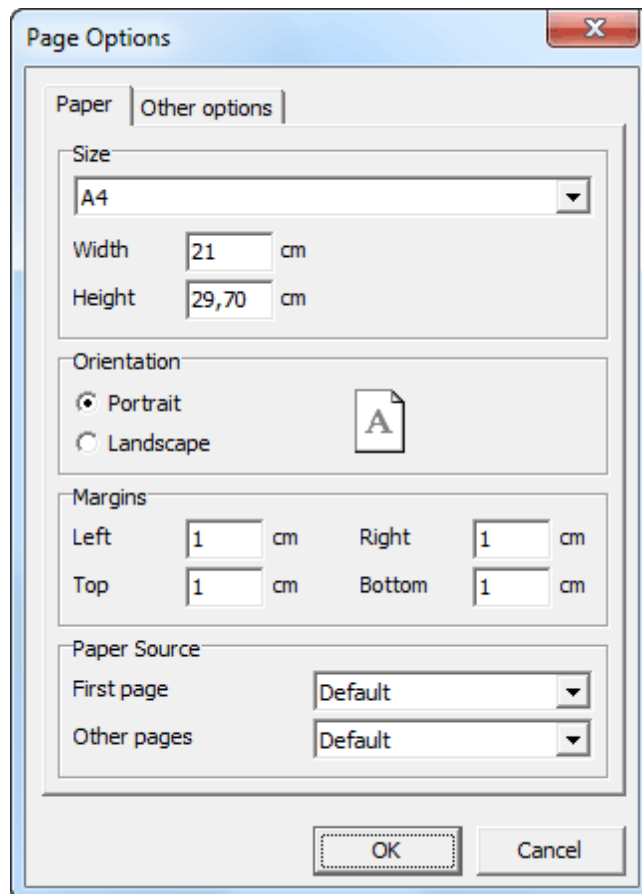
[Subreport object](#) allows inserting an additional report design page inside the basic one.

[System text](#) displays service information (date, time, page number, etc), as well as aggregate values.

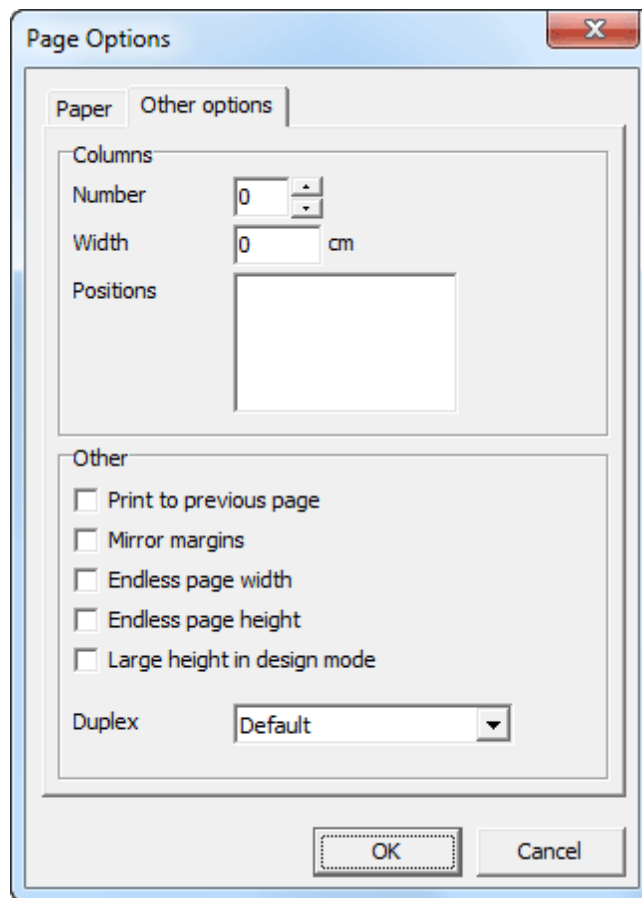
[Draw object](#) represents different geometrical figures (diagonal line, rectangle, rounded rectangle, ellipse, triangle, and diamond).

### Page options

This dialogue allows you to set the page settings of the prepared. To invoke the window, use the [Edit...](#) link of the page blank space popup menu. The dialogue has two pages: [Paper](#) and [Other options](#). On the [Paper](#) page, you can select size and alignment of paper, as well as set margins. In [Paper source](#) drop-down lists you can select a printer tray for the first page and the rest of the report pages.



On [Other Options](#) you can set the number of columns for multi-column reports' printing. The current settings are displayed in the designer.



The [Print to previous page](#) flag allows you to print pages, beginning from blank space of the previous page. This option can be used in case when a report template consists of several pages or when printing batch (composite) reports.

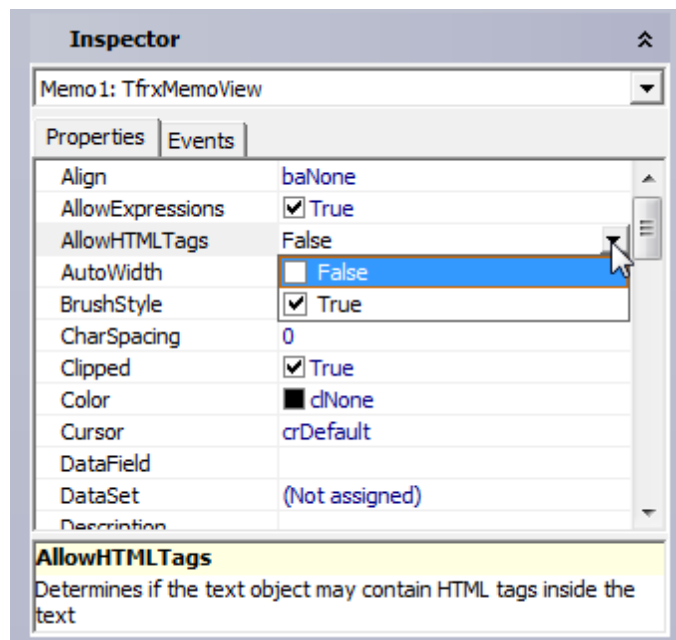
The [Mirror margins](#) option switches right and left margins of page for even pages during previewing or printing a report.

The [Endless page width & height](#) option increases page's sizes depending on number of data records on the page (when running a report). In this case you will see one big page in the preview window instead of several standard size pages.

The [Large height in design mode](#) option increases page's height several times more. This feature can be useful if many bands are located in the page, and must be used when working with the overlay band. This only effects the page height in design mode.

## 9.7.2 Object Inspector

[Object Inspector](#) pane allows you to specify the appearance of each report object in detail. To setup object properties, select it at the Design area or select it from the popup menu at the top of the pane. Now all the properties of the object are available for editing. The most of properties are provided by a set of available values. The description of the selected option is displayed at the bottom of the pane.



Below you can find a brief description of several options.

**Align** - set here the align option of the object according to the list.

**AllowExpressions** - enables the ability to display not only a static text, but expressions as well.

**AllowHTMLTags** - Enables using some simple HTML tags inside the text of an object. This option is disabled by default. Here is the list of supported tags:

<b> - bold text;

<i> - text in italic;

<u> - underlined text;

<sub> - subscript;

<sup> - superscript;

<font color> - font color;

<nowrap> - text which does not get broken up when using **WordWrap**, but gets transported wholly.

**Font:** there are abilities to specify the charset, font color, font name, and font size, and also set the bold, italic, underline, strike out attributes.

**Frame:** You can set as the color, the style and the shadow for all the frame, as well as for each frame line.

**BrushStyle** - type of object filling.

**CharSpacing** - space between symbols in pixels.

**GapX, GapY** - text indents from object's left and top boundaries (in pixels).

**LineSpacing** - space between lines (in pixels).

**ParagraphGap** – the first paragraph line indent (in pixels).

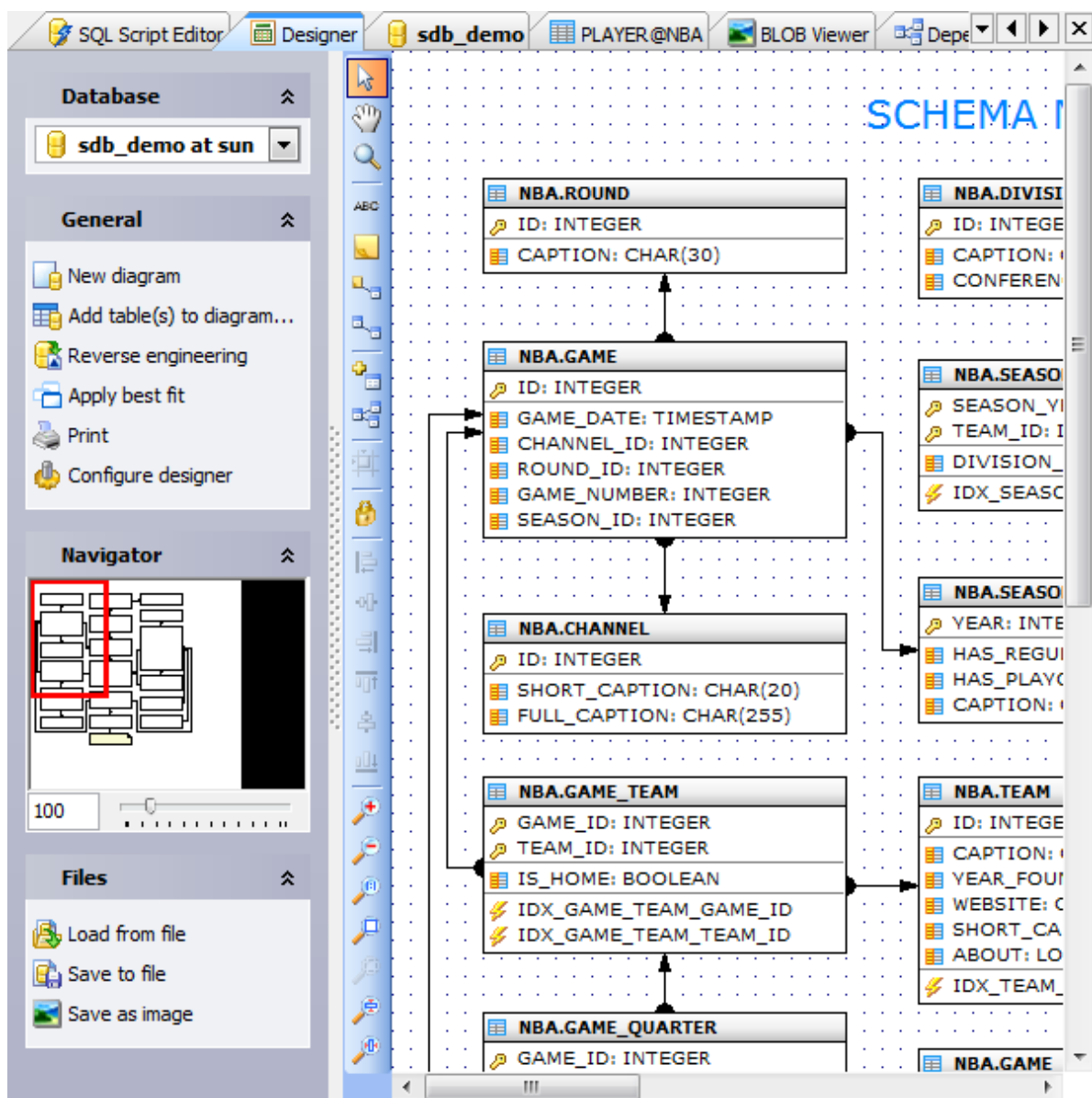
**Rotation** - specify the angle of the text rotation in the field.

**WordWrap** - if it is disabled, the long lines will be simply cut off.

## 9.8 Schema Designer

**Schema Designer** allows you to create physical Entity Relationship Diagram that will represent objects in your DB2 database. A diagram represents the tables of your database and the relationships between them. The tool is intended for reverse engineering and database modification in an easy and powerful way. It helps you to simplify database maintenance.

A diagram of your database can help you define operational aspects of your application logic that you might otherwise overlook. Also, a well-defined data diagram that accurately represents your tasks can be helpful in orienting employees to goals and operations. The data diagram can also serve as an invaluable communications tool for both internal and external constituents.



Below you can find answers for the following questions:

- [How can I add a table to a diagram?](#)<sup>[275]</sup>
- [How can I add a relationship between tables?](#)<sup>[275]</sup>
- [How can I delete a diagram object?](#)<sup>[275]</sup>
- [How can I work with diagram objects?](#)<sup>[276]</sup>

**See also:** [Designer Navigation bar](#)<sup>[276]</sup>, [Schema Designer Toolbox](#)<sup>[276]</sup>.

### Adding a table

You can add an existing DB2 table to the diagram using popup menu in the working area, or with the corresponding link on the [Navigation bar](#)<sup>[276]</sup>.

To create a new table, use the appropriate item of the popup menu in the working area. The table will be created in the current database.

Tables also may be dragged on the diagram from [Explorer](#) and the similar to [Explorer](#) tools like [Object Manager](#) and [Object Browser](#).

Moreover, [Designer](#) provides you with a possibility to represent all the tables and relationships existing in the database automatically (the [Reverse engineering](#) link of the [Navigation bar](#)). At that the database contents will be represented on a diagram in the most compact and vivid manner.

All the diagram objects are available for editing. Just double click the object (table or relationship) to view/edit its properties within the corresponding editor.

### Adding a relationship between tables

At adding to a diagram tables that reference on each other, the relationships between tables are represented automatically. The [Schema Designer](#) tool also allows you to add new foreign keys to the diagram tables. Thereto you can do the following.

Select a table (child table)

- Use the [Create new...](#) item of the popup menu to launch [Foreign Key Properties](#)<sup>[78]</sup> window.
- Specify there properties of the relationship been created.

Moreover you can add a reference graphically:

- Choose the [Create relation](#) tool on the [Toolbox](#)<sup>[276]</sup>. Your mouse cursor will change its appearance.
- Then click on the table (child table) that will have foreign key and then click on the second table (parent table) whose primary key will be referred by the new foreign key.
- Specify properties of the relationship been created in the [Foreign Key Properties](#)<sup>[78]</sup> window.

With the [Create new...](#) item of the popup menu you can also add a new field, an index, a trigger, etc. to the selected table. For more information about object properties see: [Field Editor](#)<sup>[72]</sup>, [Index Editor](#)<sup>[76]</sup>, [Create Trigger Wizard](#)<sup>[84]</sup>.

### Deleting of the diagram objects

To hide a table (several tables) or a relationship between tables, select the objects and click [Remove selection](#) link of the popup menu or [Navigation bar](#). You can also use the **Del**

key for this purpose.

It's also possible to physically delete a table/foreign key from the database: just select the object to delete and use the appropriate item of the popup menu.

### Editing of a diagram appearance

Movement of a table/several tables along the diagram is realized with dragging or pressing **Ctrl**+arrows. You can use **Shift**+arrows to change width/height of table/several tables representation.

[Designer](#) also allows you to edit shape of the line representing foreign key relations/ logical relations. In order to break the line you should

- Select the relationship.
- Press **Ctrl** and click on the necessary line section to create a new node.
- Position the node by dragging.

You can also delete a node on the line. Thereto

- Select the relationship.
- Press **Alt** and click the node to delete. In that case the near nodes will be united by a straight line.

## 9.8.1 Designer Navigation Bar

The [Navigation Bar](#) of [Schema Designer](#) provides you the following opportunities:

Use the [Database](#) drop-down list to move around your DB2 databases.

There are also links for adding a [New diagram](#) or an existing [table to diagram](#) quickly.

### [Reverse engineering](#)

The link provides you to create a new diagram with all the database tables and

### [Apply best fit](#)

Use the link to dispose tables on the diagram in the most clear manner.

### [Remove selection](#)

The link cancels current object selection.

Use [Print](#) to see the print preview of the diagram.

Certainly, it's possible to customize [Schema Designer](#) with [Configure designer](#). For more information see [Schema Designer Customization](#)<sup>[305]</sup>.

The [Navigator](#) part allows you to adjust the scale of the diagram and the position of the visible part.

Besides the [Navigation bar](#) allows you to [Load](#) a diagram from file, [Save to file](#), and [Save as image](#) (Bitmap, GIF and JPEG formats are supported).

## 9.8.2 Schema Designer Toolbox

The toolbox is located on the left side of the [Schema Designer's](#) working area.



### Move

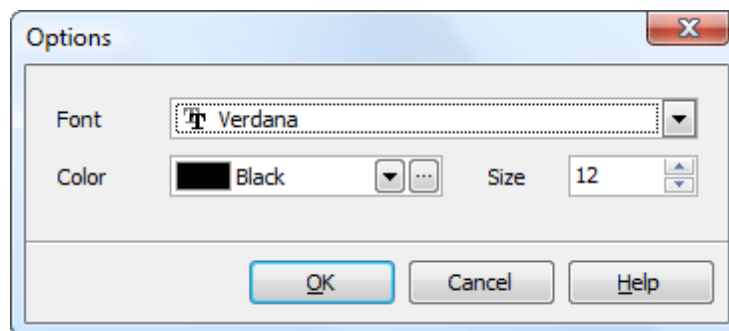
The tool is intended for selection of diagram objects. Use the tool then click anywhere inside of the object. Double click opens the corresponding [Object Editor](#).

To select multiple objects, use the tool then click and drag a selection rubber-band so that the rubber-band box encompasses the objects you want to select, and then release the mouse button.

To add objects to the list of already selected objects again, use the Move tool then click anywhere inside of the object holding the **Shift** button. To quick launch of the tool, use **M** shortcut.



Use [Create text box](#) to add [title and comments](#) on your diagram. Click on the necessary place and double-click on the appeared box to enter a text. You can also tune up the text font, color and size with [Text options](#) of the box popup menu. To quick launch of the tool, use **XX** shortcut.



Moreover you can add notes and also links between them and diagram elements using



[Create note](#) and [Create link to note](#) links. To quick launch of the tools, use **N** and **L** shortcuts accordingly.



### Lock



The tool to locking/unlocking diagram objects. This feature prevents your diagram from unforeseen changes: when the diagram is locked, you can neither move/resize/delete existing objects nor add new ones.



[Hand](#) tool moves a diagram within its window. To quick launch of the tool, use **H** shortcut.



[Zoom](#) magnifies and reduces the view of a diagram. To zoom out, hold the Alt key. To quick launch of the tool, use **Z** shortcut.

There are also tools allowing to  [Create table](#) and  [Create relation](#) directly from the [Designer](#). To quick launch of the tool, use **T** and **R** shortcut.

Below you can find toolset for aligning the selected objects by left and right edges, by horizontal and vertical centers, tops and bottoms.

Click the [Zoom in](#) button in the options bar to magnify to the next preset percentage.

When the image has reached its maximum magnification level, the command is dimmed.

Click the [Zoom out](#) button in the options bar to reduce to the previous preset percentage. When the image has reached its maximum reduction level, the command is dimmed.

Click the [Zoom 1:1](#) button to display a diagram at 100%.

Pay attention to the [Fit diagram](#) function, that pick-up properly scaling factor to display your diagram fully. For your convenience the [Fit selected](#), [Fit height](#), and [Fit width](#) were added.

## 9.9 Dependency tracker

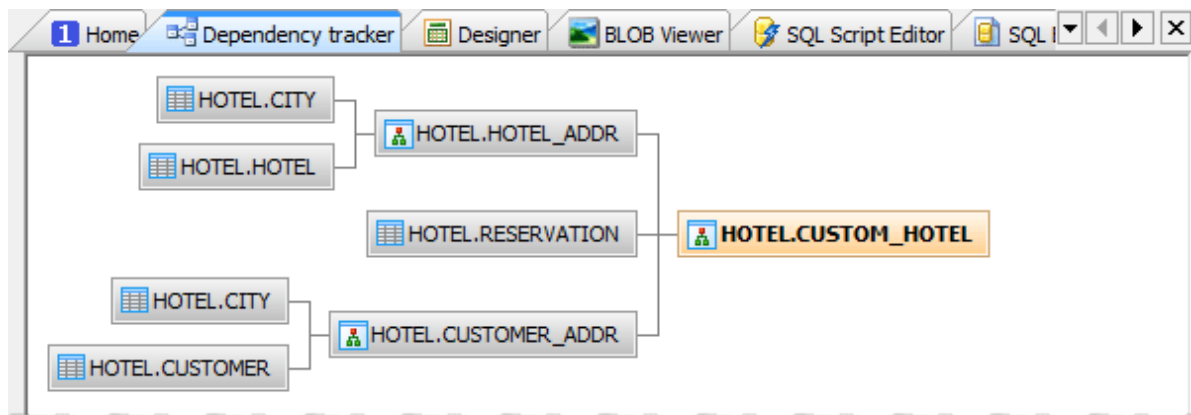
**Dependency tracker** is a tool to browse all-level dependencies of a schema object (table, view, function, etc). To display dependencies of an object, drag and drop it from the Explorer tree (or Object Manager, Object Browser) to the tracker's working area.

The selected object is represented in the center of the diagram as a pink rectangle.

Objects, depending of the selected one, are represented on the right side of the pink rectangle.

Objects, the selected one depends of, are represented on the left side of the pink rectangle.

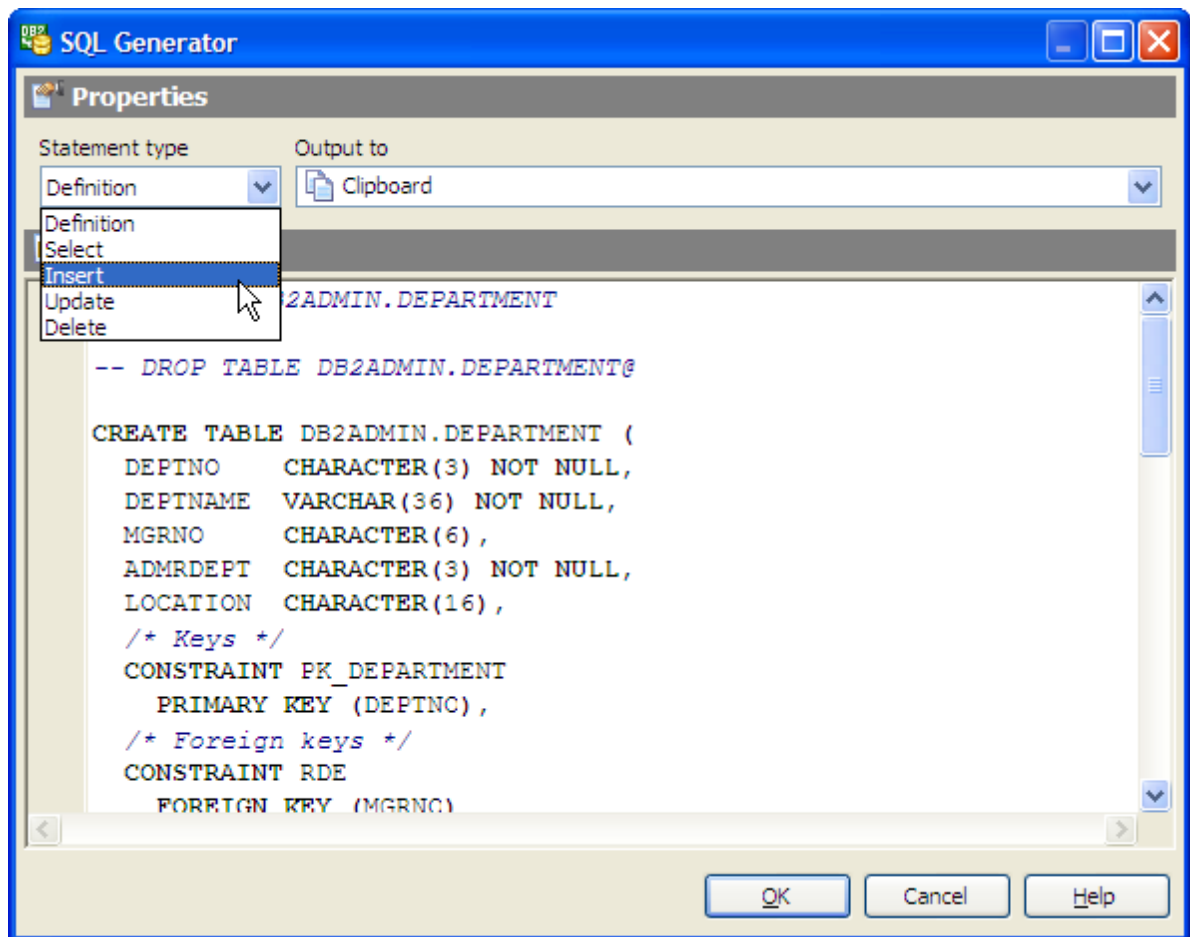
Objects that are represented on the diagram more than once are marked by semicircle. To find all inclusions of such object in the diagram, click any its rectangle. All the places the objects has on the diagram will be colored in pink.



## 9.10 SQL Generator

Among other features DB2 Maestro provides you with SQL Generator, a tool to create simple SQL statements. Just choose a database object, select statement type (Definition, Select, Insert, Update, or Delete) and the destination device (Clipboard, File, SQL Editor, SQL Script Editor).

The SQL Generator window can be invoked from the Explorer tree.



## 9.11 DML procedures generation

DB2 Maestro allows you to create DML (also known as CRUD) procedures automatically. CRUD is an acronym for the four essential database operations: Create, Read, Update, and Delete. The application designer has many choices for accomplishing the CRUD operations but the most efficient choice in terms of DB2 performance is to create a set of stored procedures to perform the operations.

### **The reasons for using DML Procedures instead of allowing ad hoc SQL statements are:**

- The best possible performance

After the first use of each stored procedure, the plan for executing the procedure is cached in the server's procedure cache. For subsequent invocations of the stored procedure, the plan is reused. This avoids the parsing and optimization steps with their overhead.

- Removing of the SQL code from the other layers of the application

By removing the SQL statements from the application code, all the SQL can be kept in the database.

- Preventing of SQL injection attacks


Anytime a client application uses string concatenation to create SQL statements, there is a possibility of a SQL injection attack. In short, these attacks involve clever entry of SQL in the data entry fields of an application in such a way that the SQL statements executed are different from the ones intended by the programmer. They require that the application developer is careless about not cleaning any user input to prevent the attack.

- Preventing of casual table browsing and modifications

If an application uses ad hoc SQL statements, the users of the application must have the required permissions on the database tables. Once they are given permission on the tables, they can work with them in any application that can read and manipulate the data such as Excel, Word and various report writers. Casual examination of the data and even updates that bypass the application's business rules become possible. Stored procedures have long been used to prevent casual browsing and updates. This is implemented by granting permission to execute the CRUD stored procedures to the users and revoking permission to access the tables directly.

### **To generate DML procedure,**

- select the [Object | Generate DML procedures...](#) main menu item (to create procedures for several tables) or use the corresponding popup menu item of the table's node at the Explorer tree (to create procedures for one table).
- Specify tables the procedures will be created for (in case of several tables).
- Uncheck the operations the procedures will not be created for. By default the procedures are generated for inserting, reading, updating, and deleting of table data.
- Adjust templates of procedures names.
- Select the action to perform after the generation. The created definitions can be copied to Clipboard, saved to a file, sent to SQL Script Editor or executed immediately.

 Options


Procedures to create


Select procedure	<input checked="" type="checkbox"/>
Update procedure	<input checked="" type="checkbox"/>
Insert procedure	<input checked="" type="checkbox"/>
Delete procedure	<input checked="" type="checkbox"/>

Naming

Select procedure name	sp_sel_%TableName%
Update procedure name	sp_upd_%TableName%
Insert procedure name	sp_ins_%TableName%
Delete procedure name	sp_del_%TableName%
Parameter name	p_%ColumnName%

Action to perform after generation

 Execute immediately



## 9.12 Generation of updatable views

To generate updatable view,

- select the **Object | Generate updatable views...** main menu item (to create views for several tables) or use the corresponding popup menu item of the table's node at the Explorer tree (to create a view for one table).
- Specify tables the views will be created for (in case of several tables).
- Specify the abilities to be available on working with the view data. By default the views are generated for inserting, updating, and deleting of table data.
- Adjust the name templates of views and corresponding triggers.
- Select the action to perform after the generation. The created definitions can be copied to Clipboard, saved to a file, sent to SQL Script Editor or executed immediately.

**Options**

Abilities	
Insert	<input checked="" type="checkbox"/>
Update	<input checked="" type="checkbox"/>
Delete	<input checked="" type="checkbox"/>

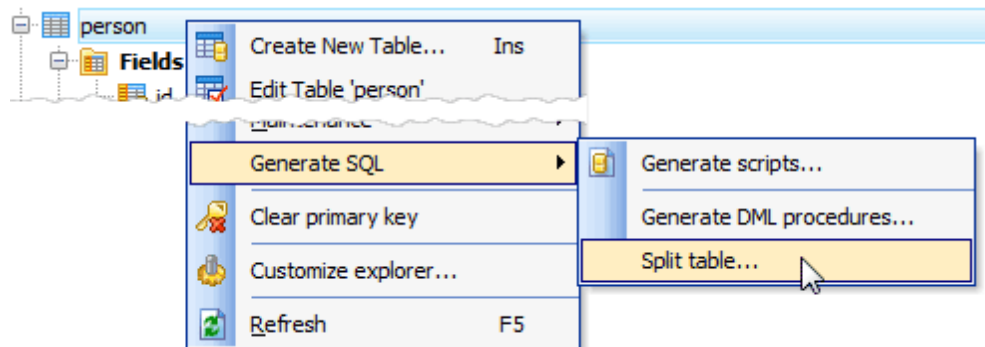
Naming	
View name	V_ %TableName%
Insert trigger name	TR_BI_ %TableName%
Update trigger name	TR_BU_ %TableName%
Delete trigger name	TR_BD_ %TableName%

Action to perform after generation

Execute immediately

## 9.13 Split table

It's not an uncommon situation when new requirements arise, or when you need to enforce referential integrity on a set of columns, and the best decision is to split a table into two separate tables. DB2 Maestro provides you with [Split Table Wizard](#), a simple tool to generate a bunch of SQL scripts to modify the primary table, to create a secondary table with a primary key, and to transfer data from the primary table to the secondary one without duplicating of data. To invoke the wizard, follow the corresponding link of the [Generate SQL](#) section of popup menu of the selected table at the Explorer tree.



Let's see the wizard in action on the example of a table with the following SQL definition:

```
CREATE TABLE person (
  id          integer NOT NULL,
  city        varchar(30) NOT NULL,
  full_name   varchar(30) NOT NULL,
  /* Keys */
  CONSTRAINT person_pkey
    PRIMARY KEY (id)
);
```

The table stores sample data:

	id	city	full_name
1	1	New York	John Smith
2	2	Boston	Mary Doe
3	3	Boston	Jason Lee
4	4	New York	Deisy O'Connor

To enforce the referential integrity, we specify 'city' as secondary table:

Primary table

public

person

Secondary table

public

city

The primary table must contain now only 'id' and 'full\_name' columns. The field 'city\_id' will be added to the table automatically.

Primary table fields

Name	
1	id
2	full_name

Secondary table fields

Name	
1	city

>

Now we have to specify what kind of primary key to be created for the secondary table: surrogate or natural. We create the 'city' table with a surrogate primary key.

☒ Use surrogate primary key

Surrogate key field name

id

☐ Use natural primary key

Secondary table key fields

Primary		Name	
1	<input type="checkbox"/>	city	<input checked="" type="checkbox"/>

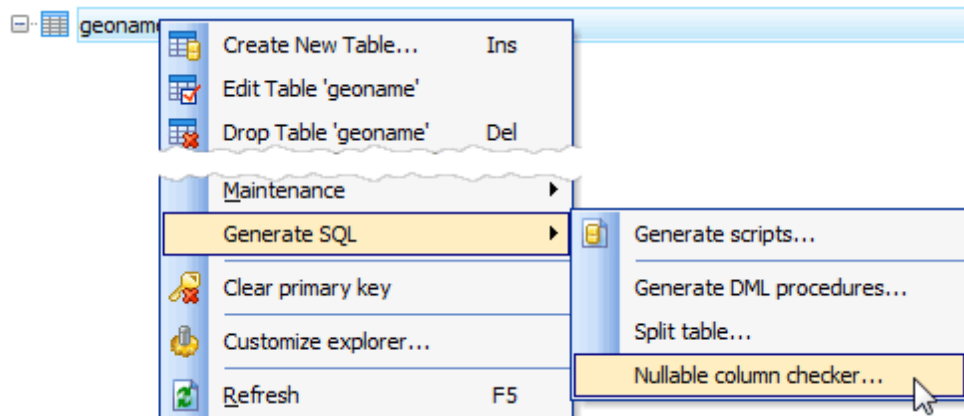
Then click Ready and get the following tables:

	id	full_name	city_id
1	1	John Smith	2
2	2	Mary Doe	1
3	3	Jason Lee	1
4	4	Deisy O'Connor	2

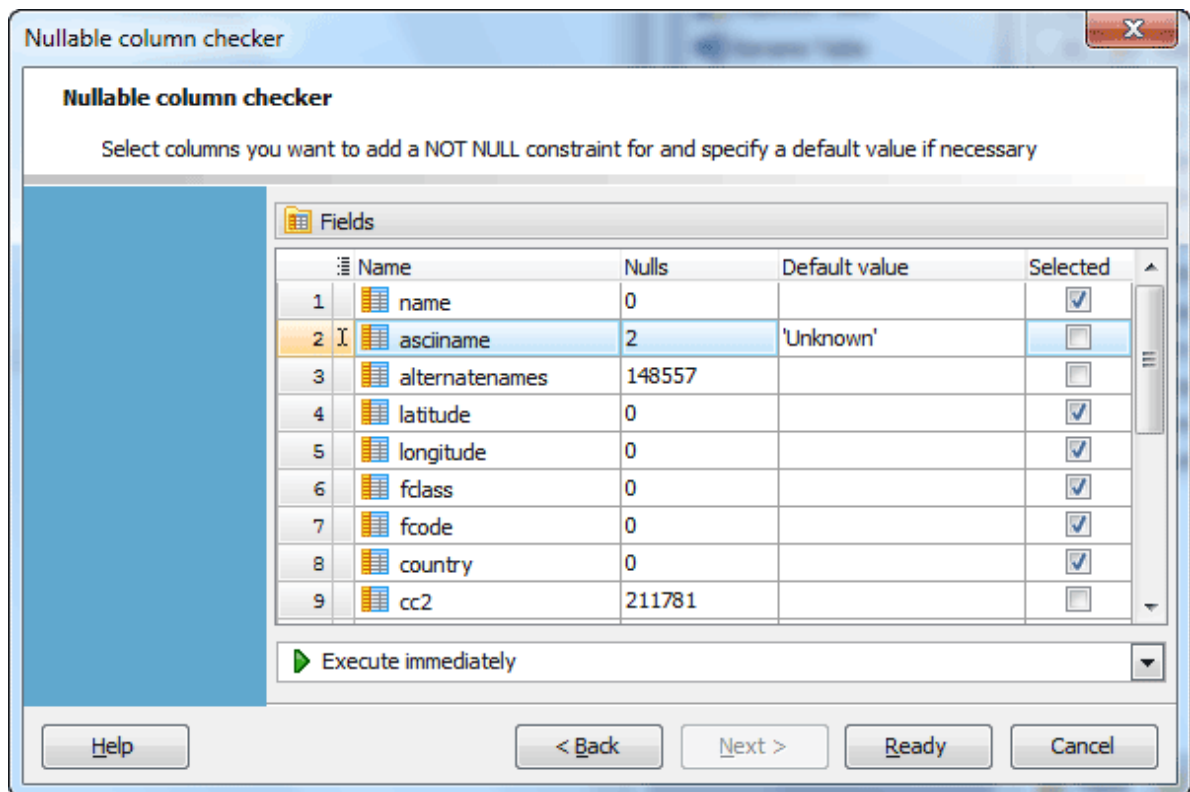
	id	city
1	1	Boston
2	2	New York

## 9.14 Nullable Column Checker

[Nullable Column Checker](#) allows you to refactor your database schema by enforcing NOT NULL constraints to all necessary table columns. It suggests candidates for NOT NULL columns among columns of the selected table and generates SQL script to replace all NULL values of selected columns with specified default values and to add the NOT NULL constraint to these columns. To invoke the wizard, follow the corresponding link of the [Generate SQL](#) section of popup menu of the selected table at the Explorer tree.



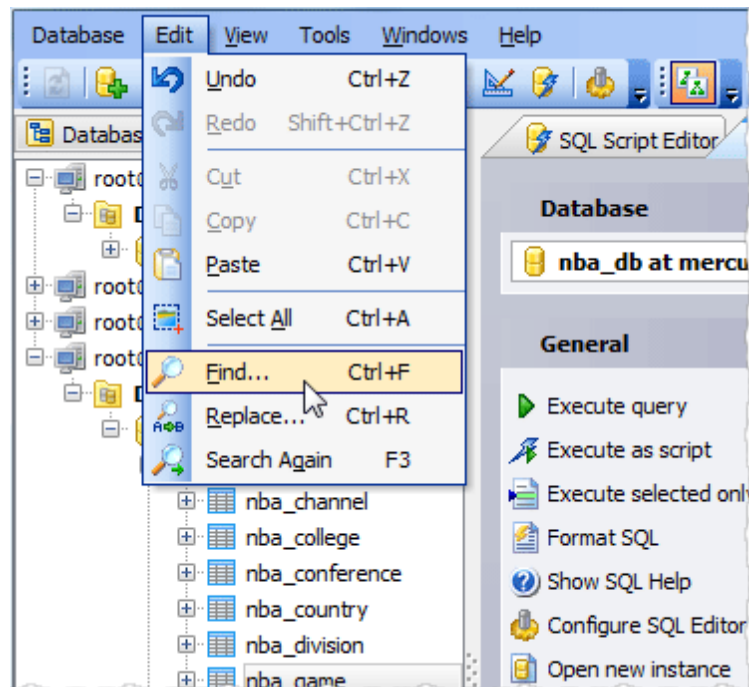
To get SQL scripts enforcing NOT NULL constraints to columns of an existing table, select the necessary columns, specify the default values to be used instead of existing columns NULLs and select the action to perform after the generation. The created scripts can be copied to Clipboard, saved to a file, sent to [SQL Script Editor](#)<sup>[248]</sup> or executed immediately.



## 9.15 Dialogs

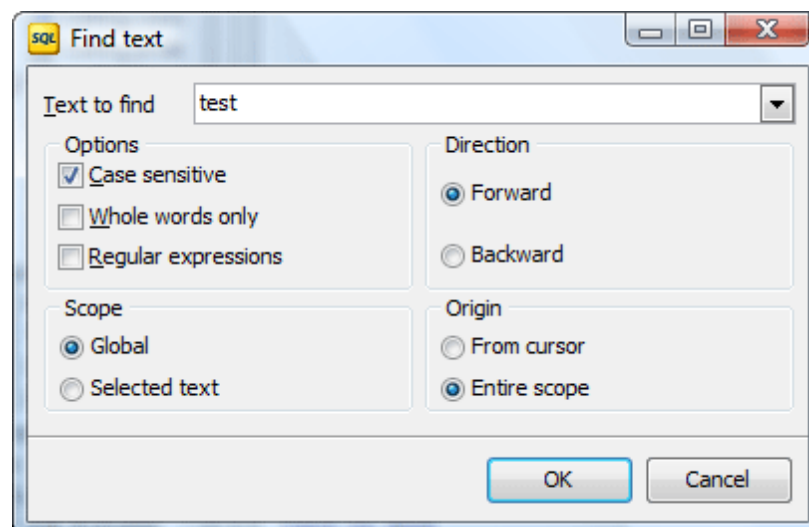
DB2 Maestro provides two dialogs for searching and replacing text in the editor areas of the database tools. Both of them are available through the popup menu of the editor area.

- [Find Text dialog](#)<sup>[288]</sup>
- [Replace Text dialog](#)<sup>[289]</sup>



### 9.15.1 Find Text dialog

The Find Text dialog is provided for quick search for certain text.



#### Text to find

Enter a search string or click the down arrow next to the input box to select from a list of previously entered search strings.

#### ☒ Case sensitive

Differentiates uppercase from lowercase when performing a search.

#### ☒ Whole words only

Searches for words only. (With this option off, the search string might be found within longer words.)

#### ☒ Regular expressions

Recognizes regular expressions in the search string.

#### Forward

Searches from the current position to the end of the file. [Forward](#) is the default.

#### Backward

Searches from the current position to the beginning of the file.

#### Global

Searches the entire file, in the direction specified by the [Direction](#) setting. Global is the default scope.

#### Selected text

Searches within the selected text only, in the direction specified by the [Direction](#) setting. You can use the mouse or block commands to select a block of text.

#### From cursor

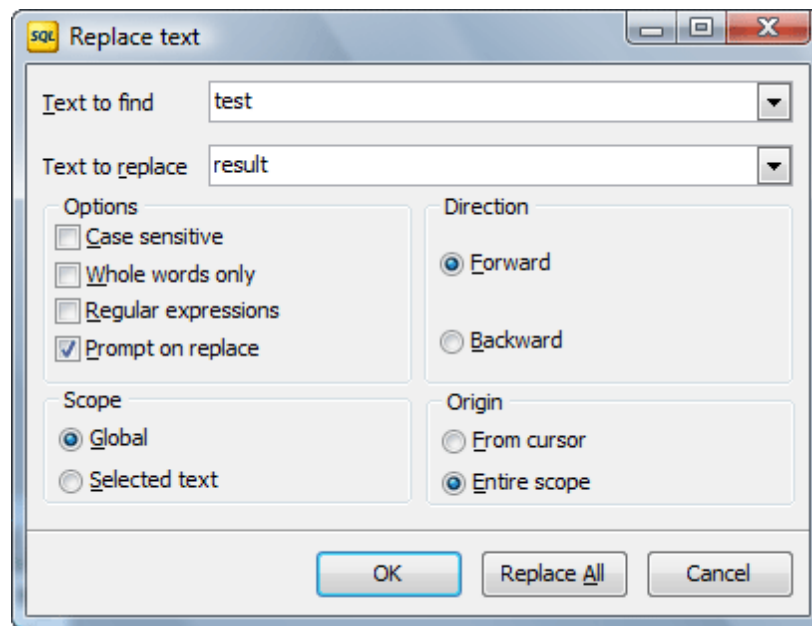
The search starts at the cursor's current position, and then proceeds either forward to the end of the scope, or backward to the beginning of the scope depending on the [Direction](#) setting. [From Cursor](#) is the default setting.

#### Entire scope

The search covers either the entire block of selected text or the entire file (no matter where the cursor is), depending upon the [Scope](#) options.

## 9.15.2 Replace Text dialog

The [Replace Text](#) dialog is provided for searching and replacing text in the editor window.



#### Text to find

Enter a search string. To select from a list of previously entered search strings, click the down arrow next to the input box.

#### Text to replace

Enter the replacement string. To select from a list of previously entered search strings, click the down arrow next to the input box. To replace the text with nothing, leave this input box blank.

#### ☒ Case sensitive

Differentiates uppercase from lowercase when performing a search.

#### ☒ Whole words only

Searches for words only. (With this option off, the search string might be found within longer words.)

#### ☒ Regular expressions

Recognizes specific regular expressions in the search string.

#### ☒ Prompt on replace

Prompts you before replacing each occurrence of the search string. When Prompt on replace is off, the editor automatically replaces the search string.

#### Forward

Searches from the current cursor position, to the end of the file. **Forward** is the default Direction setting.

#### Backward

Searches from the current cursor position, to the beginning of the file.

#### Global

Searches the entire file, in the direction specified by the Direction setting. **Global** is the

default scope.

#### [From cursor](#)

The search starts at the cursor's current position, and proceeds either forward to the end of the scope, or backward to the beginning of the scope depending on the Direction setting. [From cursor](#) is the default Origin setting.

#### [Entire scope](#)

The search covers either the entire block of selected text or the entire file (no matter where the cursor is in the file), depending upon the Scope options.

#### [Replace All](#)

Click Replace all to replace every occurrence of the search string. If you check [Prompt on replace](#), the [Confirm dialog](#) box appears on each occurrence of the search string.

## 10 Options

DB2 Maestro allows you to customize the way it works within the [Options](#) dialog. To open the dialog, select the [Tools | Options](#) main menu item.

The window allows you to customize the options grouped by the following sections:

- [Application](#)<sup>[293]</sup>  
General DB2 Maestro options: environment style, confirmations, window restrictions, explorer tree, [SQL Editor](#), [Visual Query Builder](#), etc.
- [Editors & Viewers](#)<sup>[315]</sup>  
Customizing of all the SQL editors - [SQL Editor](#), [SQL Script Editor](#), etc.
- [Appearance](#)<sup>[323]</sup>  
Customizing program interface - bars, trees, menus, etc.

Besides, the [Options](#) dialog allows you to export all program settings to a \*.reg file for future use, e.g. on another PC (see [Export Settings](#)<sup>[332]</sup> for details).

It is a good idea to check through these settings before you start working with DB2 Maestro. You may be surprised at all the things you can adjust and configure!

## 10.1 Application

The [Application](#) section allows you to customize common rules of DB2 Maestro behavior. The section consists of several tab; follow the links to find out more about each of them.

- [Preferences](#) <sup>[293]</sup>
- [Confirmations](#) <sup>[294]</sup>
- [Directories](#) <sup>[296]</sup>
- [Tools](#) <sup>[296]</sup>
  - [Explorer](#) <sup>[298]</sup>
  - [Object Manager](#) <sup>[299]</sup>
  - [SQL Editor](#) <sup>[299]</sup>
  - [SQL Script Editor](#) <sup>[300]</sup>
  - [Query Builder](#) <sup>[301]</sup>
  - [BLOB Viewer](#) <sup>[303]</sup>
  - [Export data](#) <sup>[304]</sup>
  - [Database Designer](#) <sup>[305]</sup>
- [Object Editors](#) <sup>[306]</sup>
  - [Table](#) <sup>[308]</sup>
- [Data Grid](#) <sup>[308]</sup>
  - [Colors](#) <sup>[311]</sup>
  - [Formats](#) <sup>[311]</sup>
  - [Filter](#) <sup>[313]</sup>

### 10.1.1 Preferences

User interface area allow you to select your favorite UI style according to your preferences.

☒ [Display splash screen at startup](#)

Displays the splash screen on DB2 Maestro startup.

☒ [Save desktop on disconnect](#)

Saves all the database windows and their positions on disconnecting from the database.

☒ [Disable multiple instances](#)

Prohibits running multiple instances of DB2 Maestro.

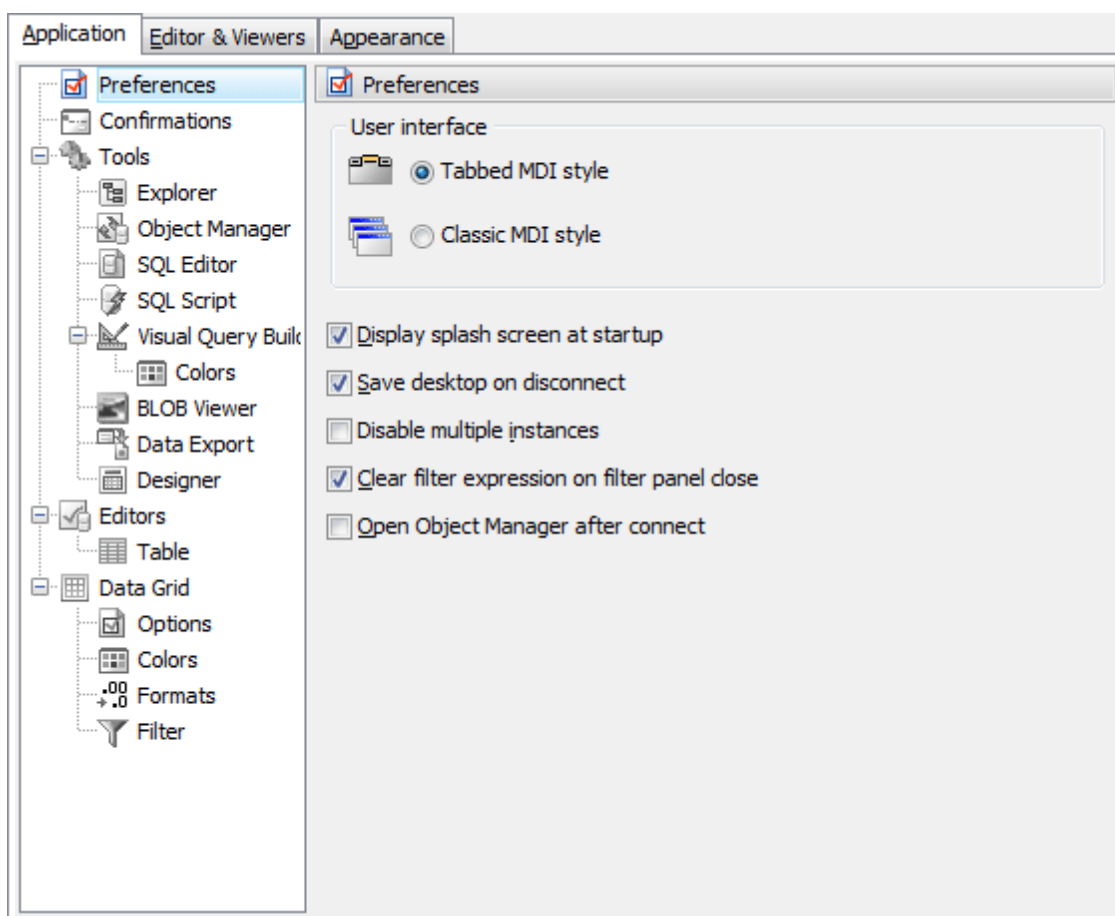
☒ [Open Object Manager after connect](#)

Opens the Object Manager window after connection is established.

☒ [Clear filter expression on filter panel close](#)

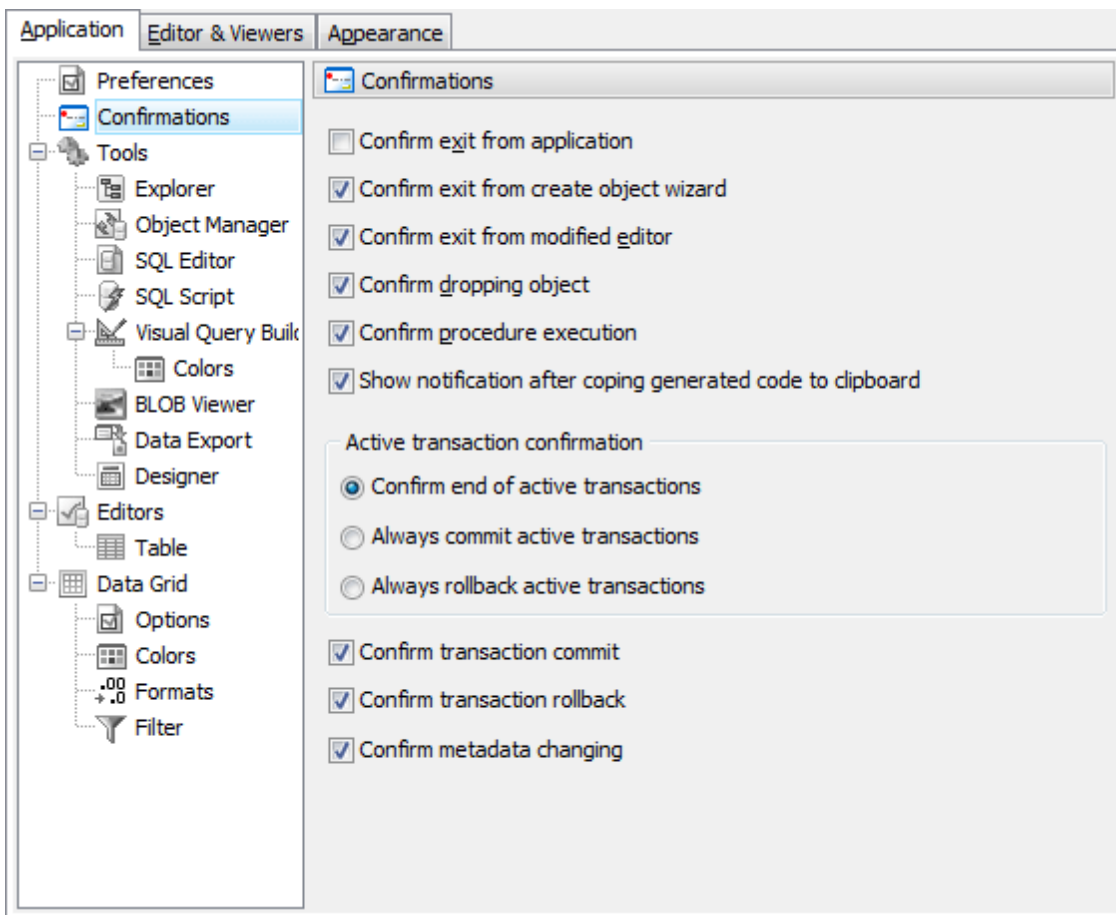
Clears the filter applied to the explorer tree and all the instances of Object Manager

after the filter panel is closed.



### 10.1.2 Confirmations

Use this tab to manage application confirmations.



☒ **Confirm exit from Create Object Wizard**

If this option is checked, the program requires confirmation each time you want to exit the Create Object Wizard.

☒ **Confirm exit from modified editor**

If this option is checked, the program asks you to confirm exit from the editor, if you have made any changes.

☒ **Confirm dropping object**

If this option is checked, the program requires confirmation for dropping database object.

☒ **Confirm exit from application**

If this option is checked, the program requires confirmation when you want to exit <% PRODUCT\_NAME%>.

☒ **Transaction confirmation**

Select whether you will be prompted to commit or rollback active transaction or DB2 Maestro will commit or rollback transactions without asking.

☒ **Confirm metadata changing**

If this option is checked, the program requires confirmation for changing metadata.

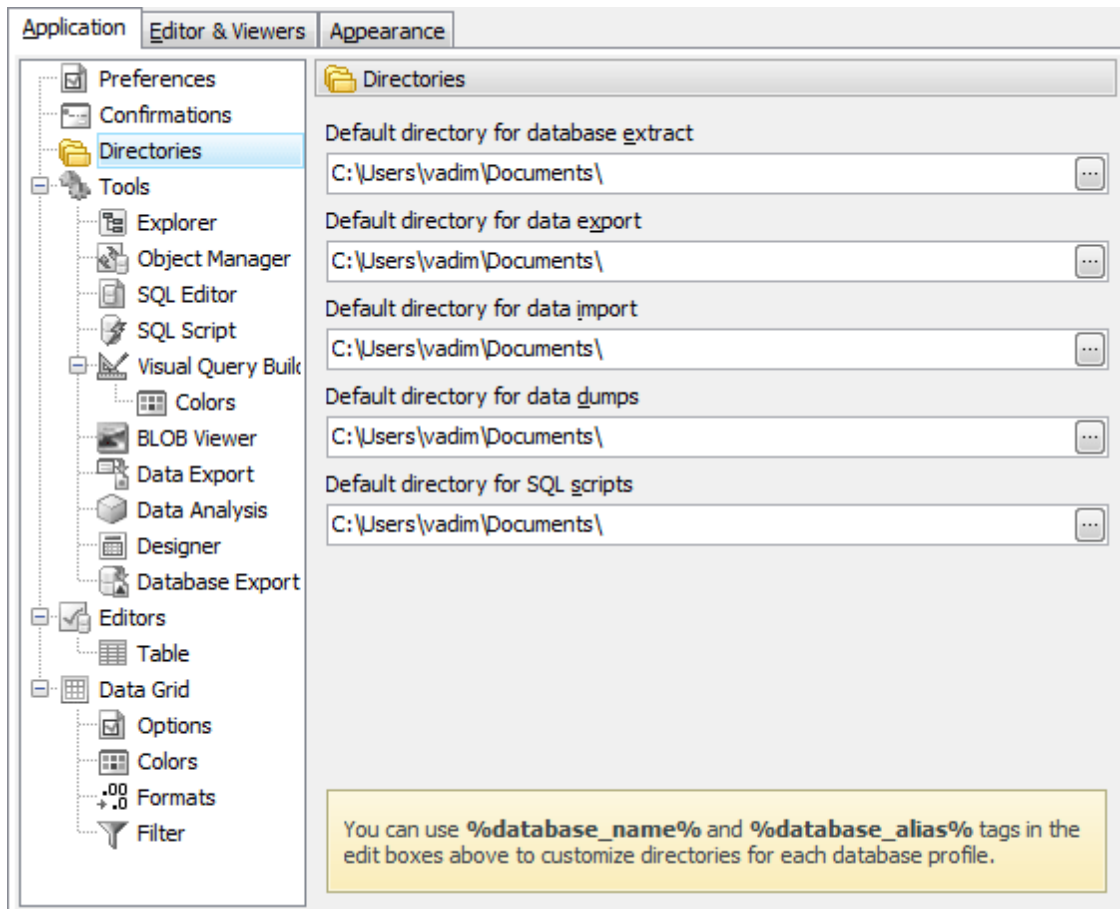
### 10.1.3 Directories

The tab allows you to specify default directories to be used on database profiles creating. You can use such variables as %database\_name%, %database\_alias%, and %user\_name%.

**Example:**

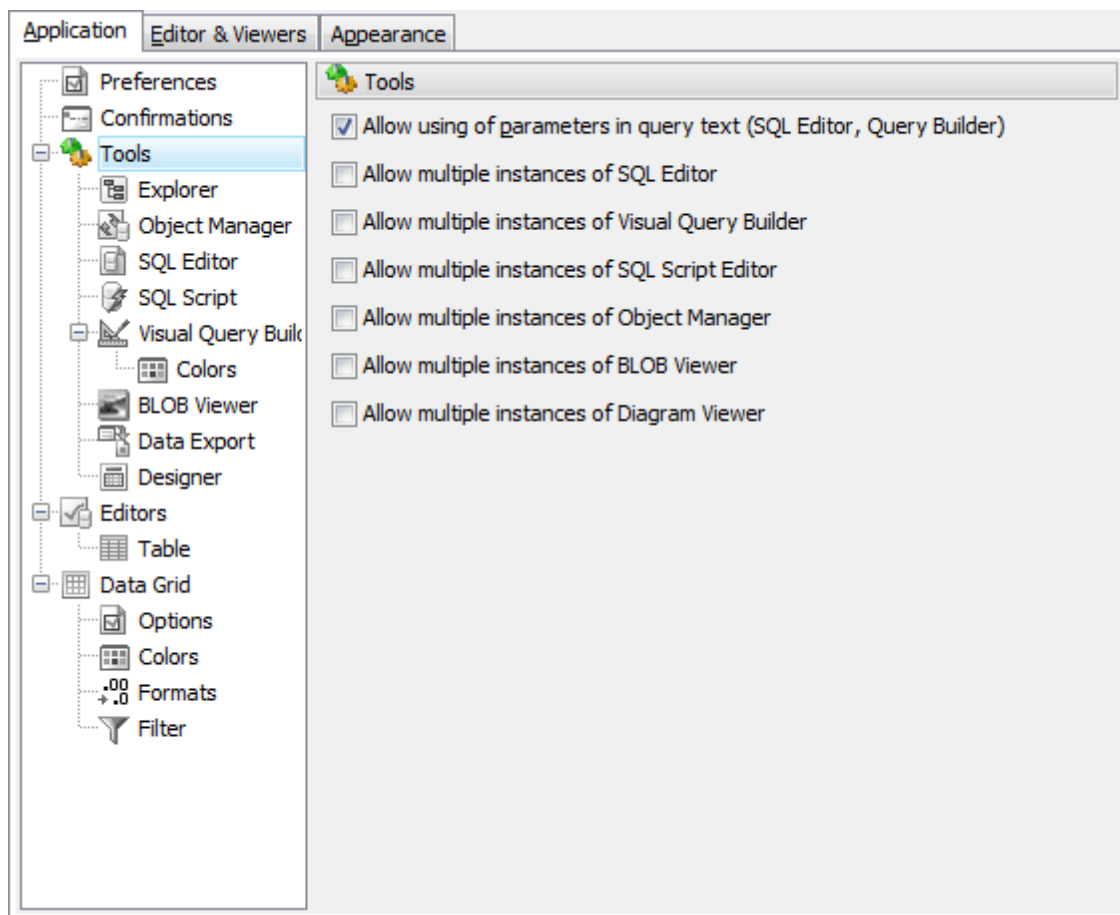
To store all SQL scripts in folders sorted by databases in the "C:\SQL Scripts\" directory, specify the default directory for SQL scripts as follows:

*C:\SQL Scripts\%database\_name%\*



### 10.1.4 Tools

Below you will find a detailed decryption of the following tools options.

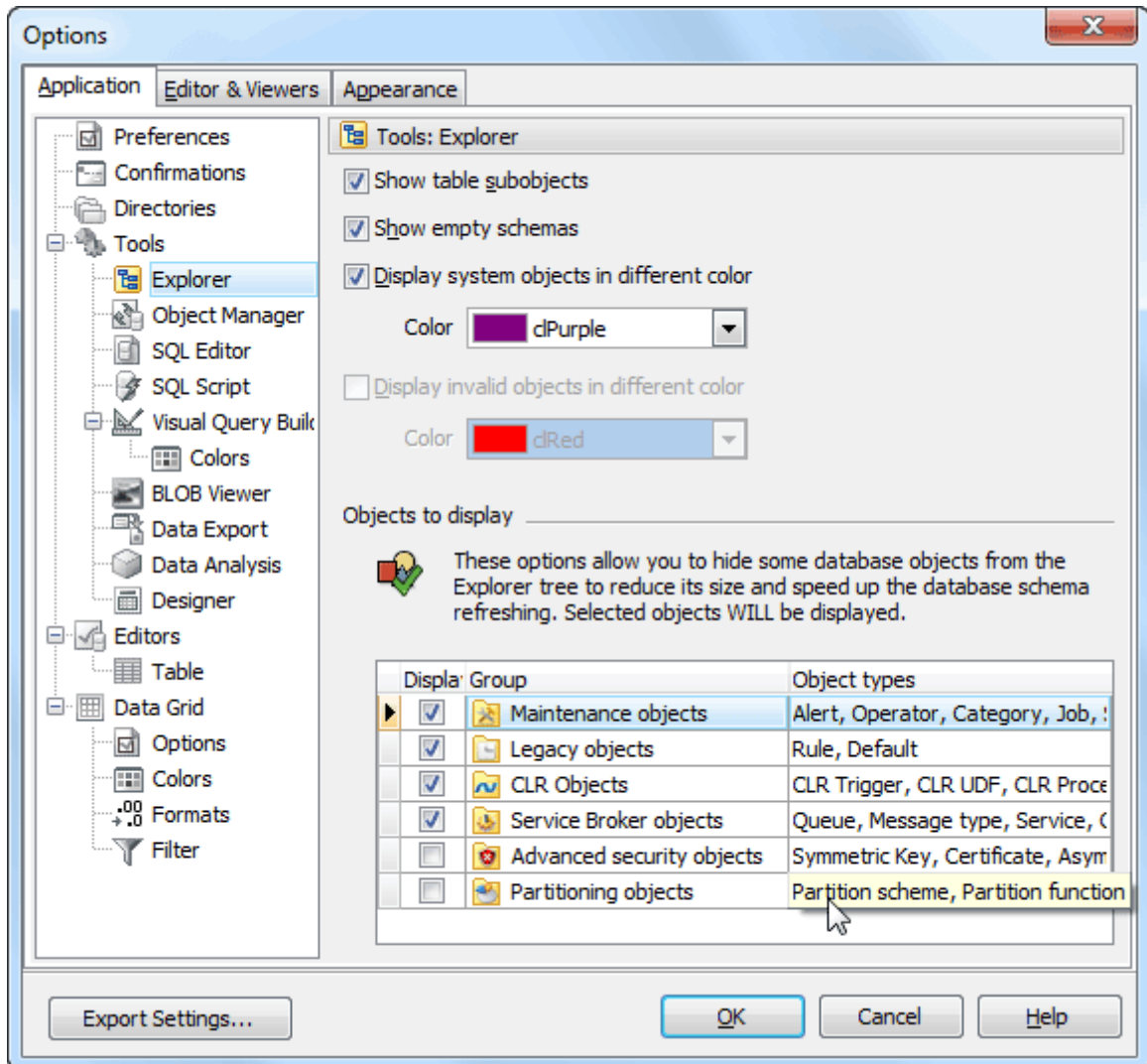


- ☒ **Allow using of parameters in query text**  
Check this option to be able to use query parameters in [SQL Editor](#)<sup>[196]</sup> and [Visual Query Builder](#)<sup>[202]</sup>.
- ☒ **Allow multiple instances of SQL Editor**  
Check this option to be able to use multiple instances of [SQL Editor](#)<sup>[196]</sup> simultaneously.
- ☒ **Allow multiple instances of Visual Query Builder**  
Check this option to be able to use multiple instances of [Visual Query Builder](#)<sup>[202]</sup> simultaneously.
- ☒ **Allow multiple instances of SQL Script Editor**  
Check this option to be able to use multiple instances of [SQL Script Editor](#)<sup>[248]</sup> simultaneously.
- ☒ **Allow multiple instances of Object Manager**  
Check this option to be able to use multiple instances of Object Manager simultaneously.
- ☒ **Allow multiple instances of BLOB Viewer**  
Check this option to be able to use multiple instances of [BLOB Viewer](#)<sup>[252]</sup> simultaneously.
- ☒ **Allow multiple instances of Diagram Viewer**  
Check this option to be able to use multiple instances of [Diagram Viewer](#)<sup>[258]</sup>

simultaneously.

#### 10.1.4.1 Explorer

Below you will find a detailed decryption of the following explorer options.



☒ **Show table subobjects**

Shows/hides table subobjects (fields and indexes) in the explorer tree.

☒ **Sort profiles by aliases**

Sorts profile aliases alphabetically in the explorer tree.

☒ **Expand the "Tables" node after connection**

Shows all database tables in the explorer tree after connecting to the database.

☒ **Expand the "Queries" node after connection**

Shows all database queries in the explorer tree after connecting to the database.

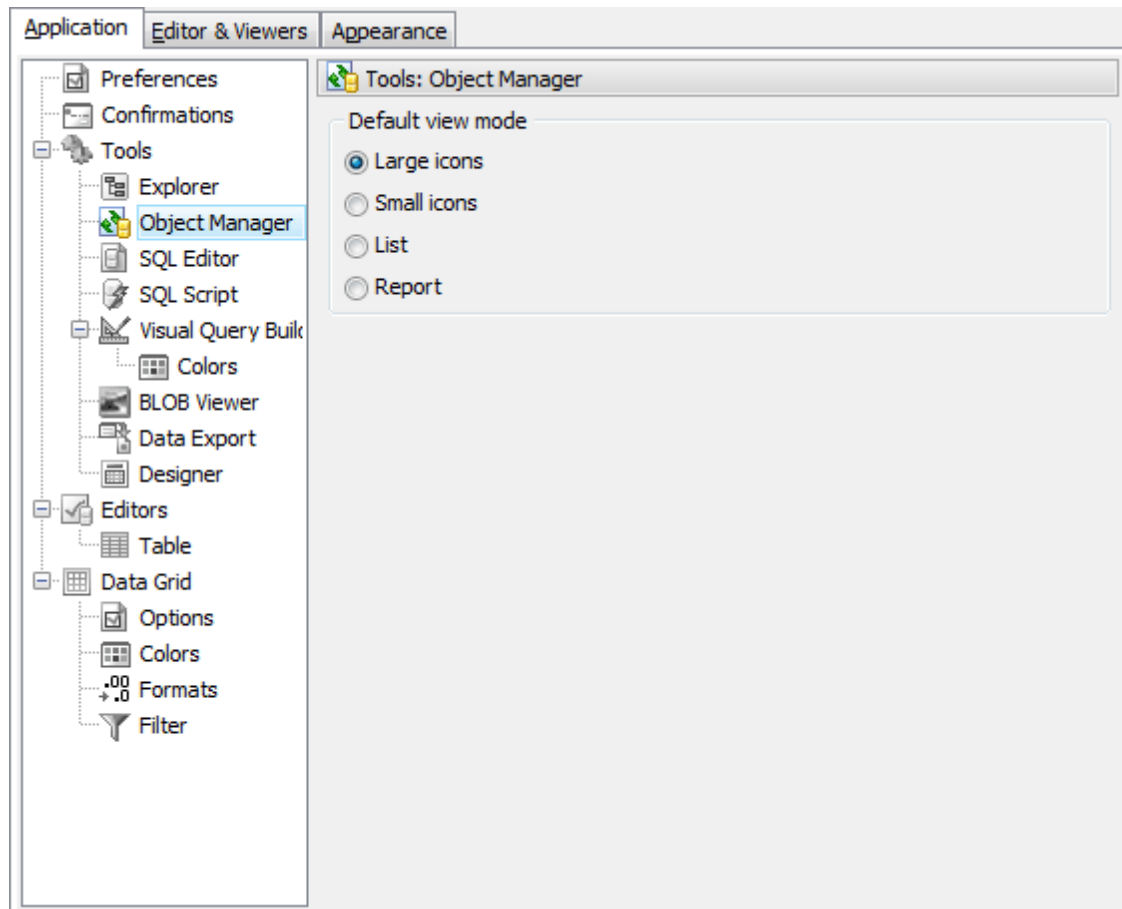
☒ **Display system objects in different color**

Represents all system objects in selected color.

You can also exclude/include rarely used objects from/to the Explorer tree. Manage object groups to be displayed at Explorer with corresponding checkboxes.

#### 10.1.4.2 Object Manager

Below you will find a detailed decryption of the following [Object Manager](#) options.

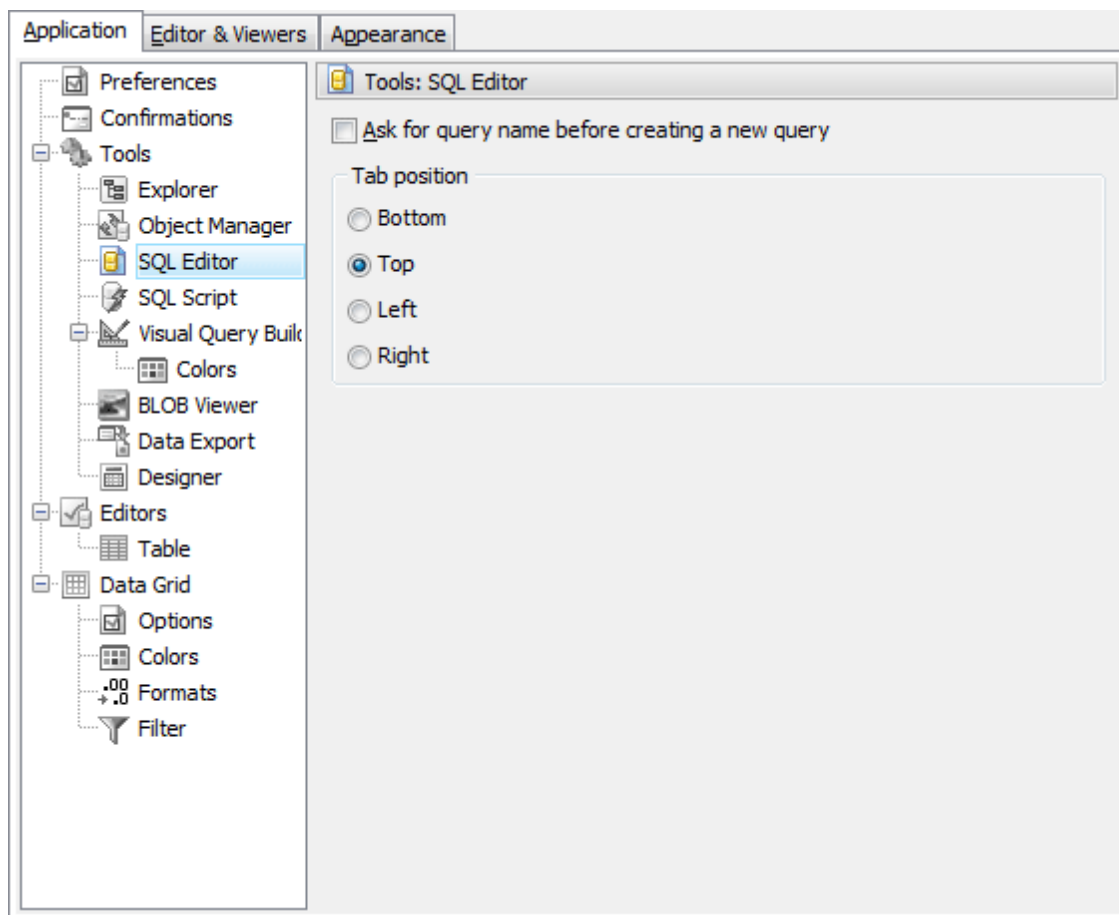


##### Default view mode

Defines which view mode (*large icons*, *small icons*, *list* or *report*) is applied to Object Manager by default.

#### 10.1.4.3 SQL Editor

Below you will find a detailed decryption of the following [SQL Editor](#) options.



☒ **Ask for query name before creating a new query**

If this option is checked, [SQL Editor](#)<sup>196</sup> asks for a query name each time you create a new query.

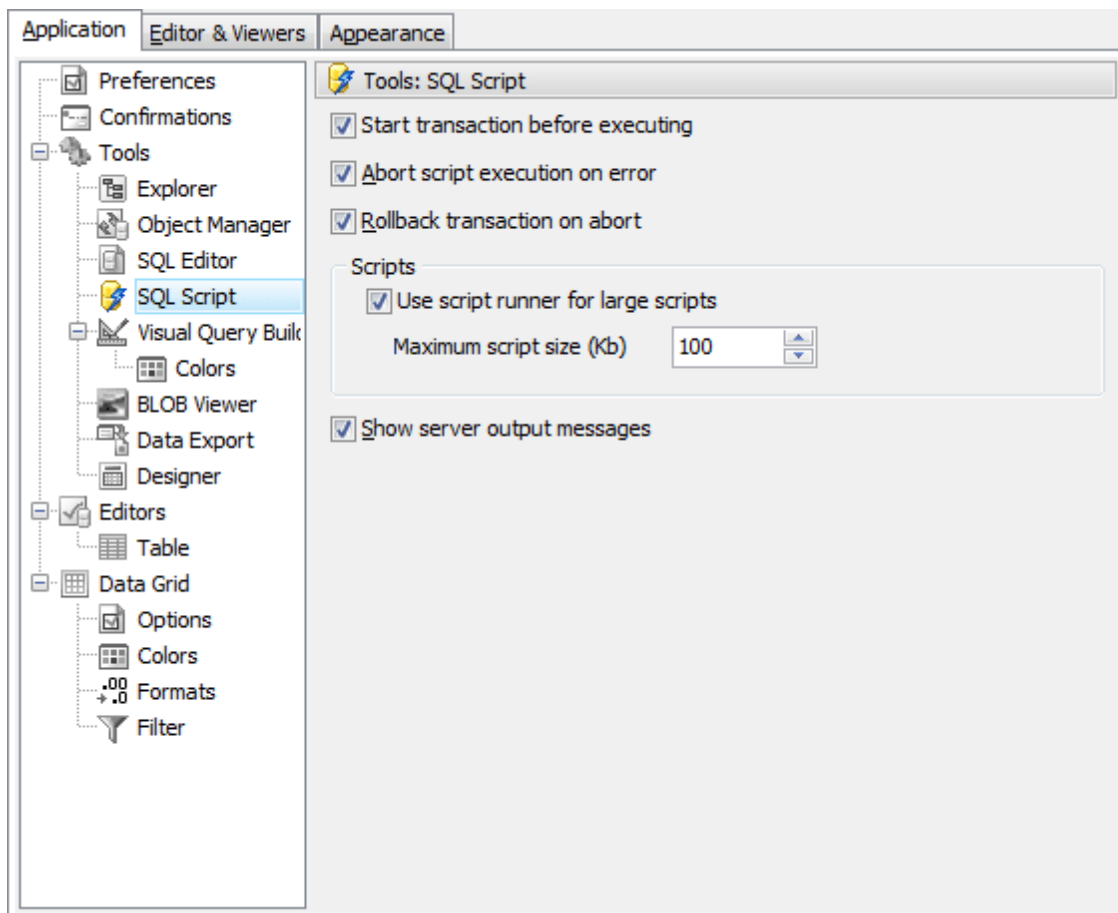
☒ **Auto commit**

Check the option to execute queries in autocommit mode (default value) or leave it blank to manage transactions manually.

You can also select [position](#) of query tabs.

#### 10.1.4.4 SQL Script Editor

Below you will find a detailed decryption of the following [SQL Script Editor](#) options.



☒ **Abort script execution on error**

If this option is checked, script execution aborts when an error occurs.

☒ **Rollback transaction on abort**

This option evokes automatic rollback on script execution abort.

☒ **Use script runner for large scripts**

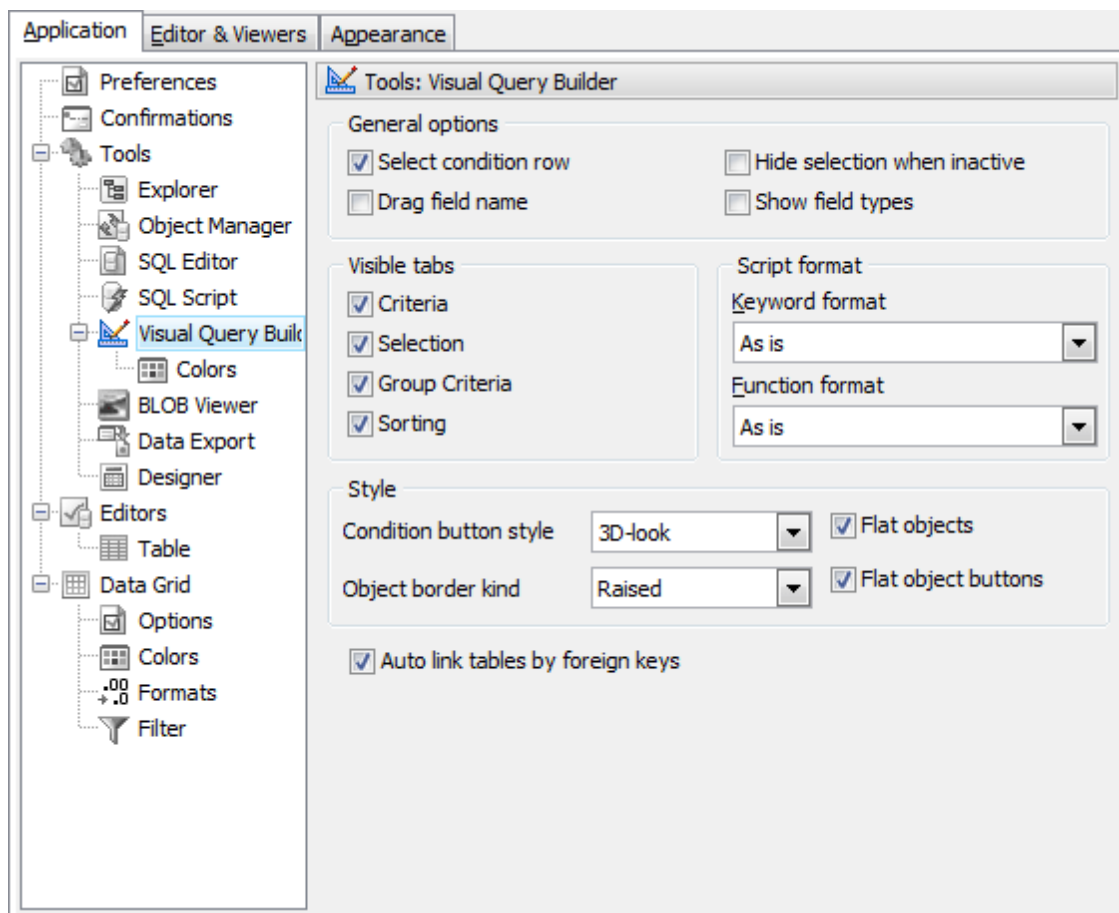
Check the box to execute large script in the fastest way. You can change the maximum size of a script to execute without script runner.

☒ **Show server output messages**

Turn the option ON to see warning messages generated by the server.

#### 10.1.4.5 Query Builder

Below you will find a detailed decryption of the following [Query Builder](#) options.



☒ **Select condition row**

Displays the selected condition in different row on the **Criteria** and **Grouping Criteria** tabs of [Visual Query Builder](#) <sup>[202]</sup>.

☒ **Drag field name**

Displays the dragged field name in the **Builder** area.

☒ **Hide selection when inactive**

Hides the selection when the query builder is inactive.

☒ **Show field types**

Displays the field type next to the field in the table box.

### Visible tabs

These options specify which the query builder tabs are available and which are not. Check them to make the appropriate tabs visible.

### Script format

These options specify the case formatting of keywords and functions in query text on the **Edit** tab. **As is** saves the original case, **Uppercase** sets all the keywords/functions to upper case, **Lowercase** sets all the keywords/functions to lower case, and **First upper** sets the first letters of all keywords/functions to upper case.

## Style

These options specify how different the [Query Builder](#) objects look like - 3D, flat, etc.

### ☒ Auto link tables by foreign keys

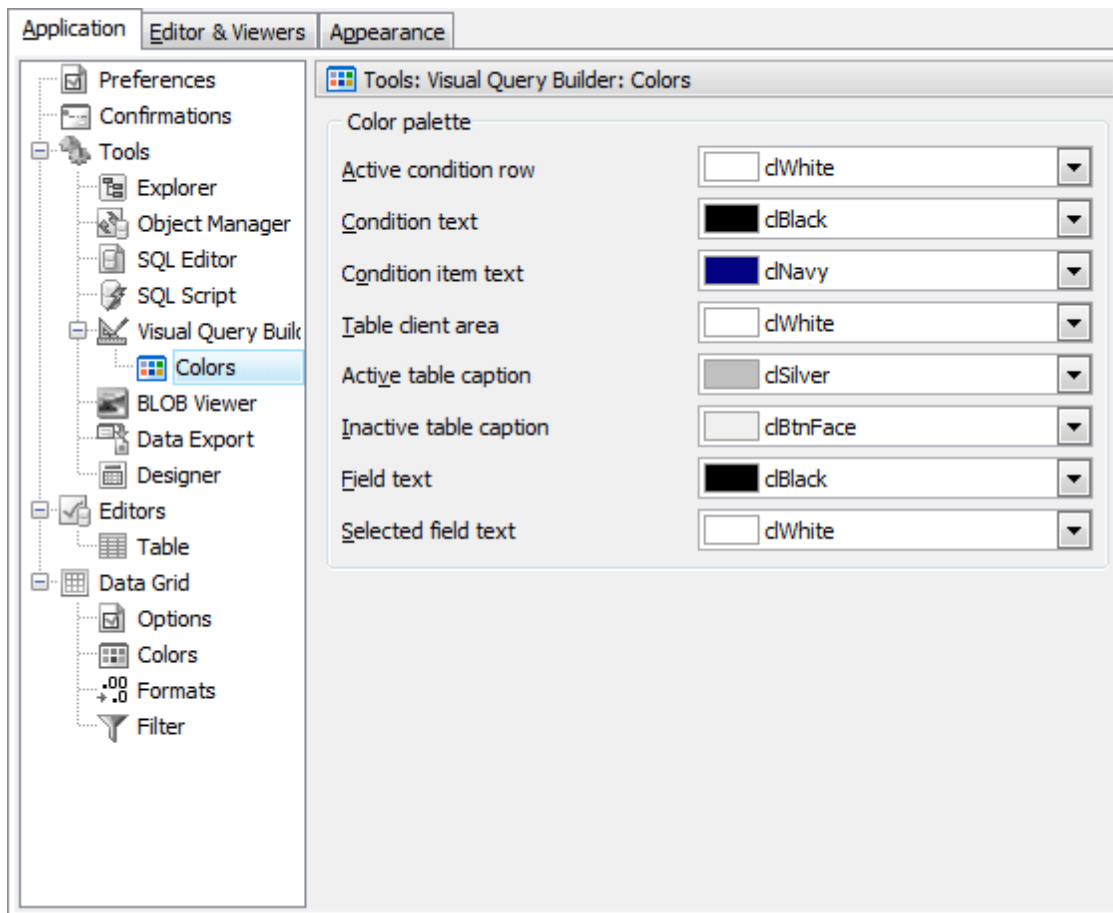
If tables that have foreign key reference are placed in [Query Builder](#), in diagram they will be auto linked.END

## Colors

These options define colors of the different [Query Builder](#) elements: condition row, active caption, table client area, etc. Click an item to select a color for the appropriate Query Builder element.

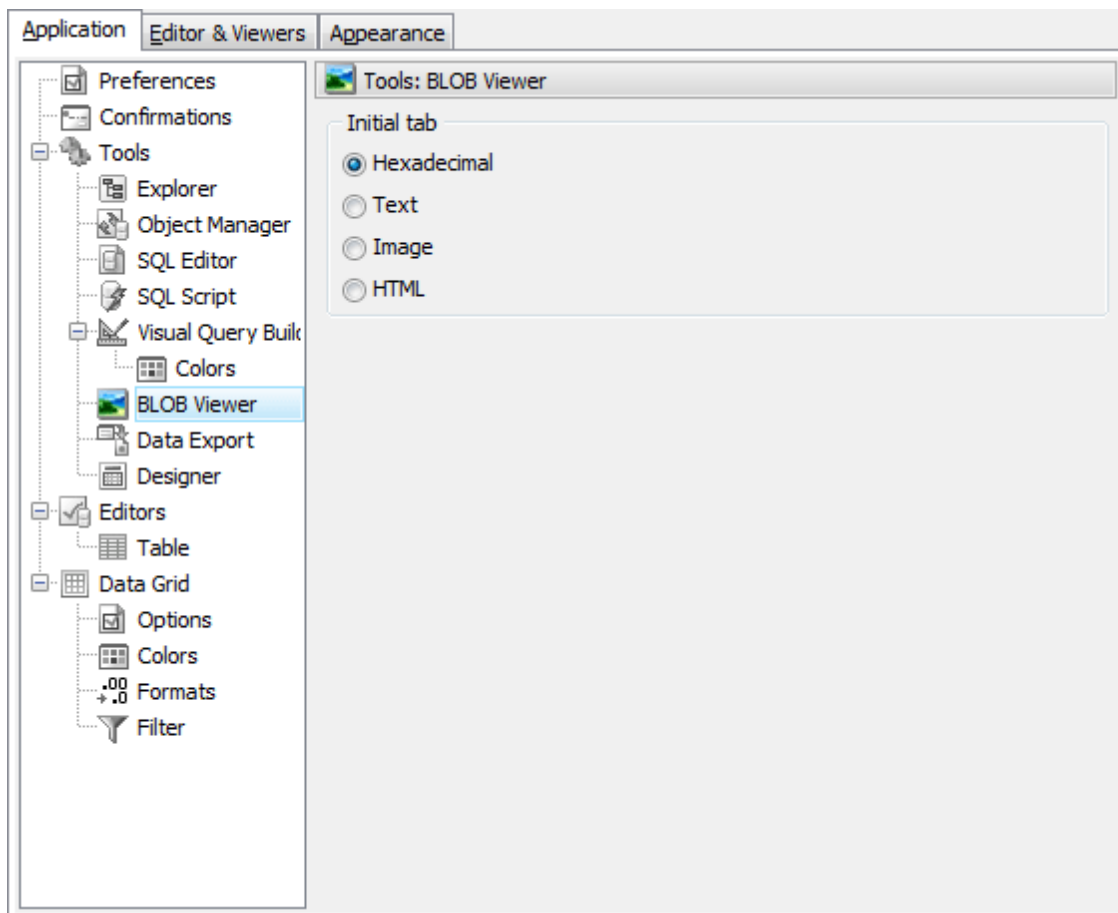
### 10.1.4.5.1 Colors

The tab is provided to editing of the Query Builder color schema. Customize colors for all editor element according to your preferences.



### 10.1.4.6 BLOB Viewer

Below you will find a detailed decryption of the following [BLOB Viewer](#)<sup>252</sup> options.



#### Initial tab

Specifies which tab of **BLOB Viewer** should be active when it is initially opened.

#### 10.1.4.7 Data Export

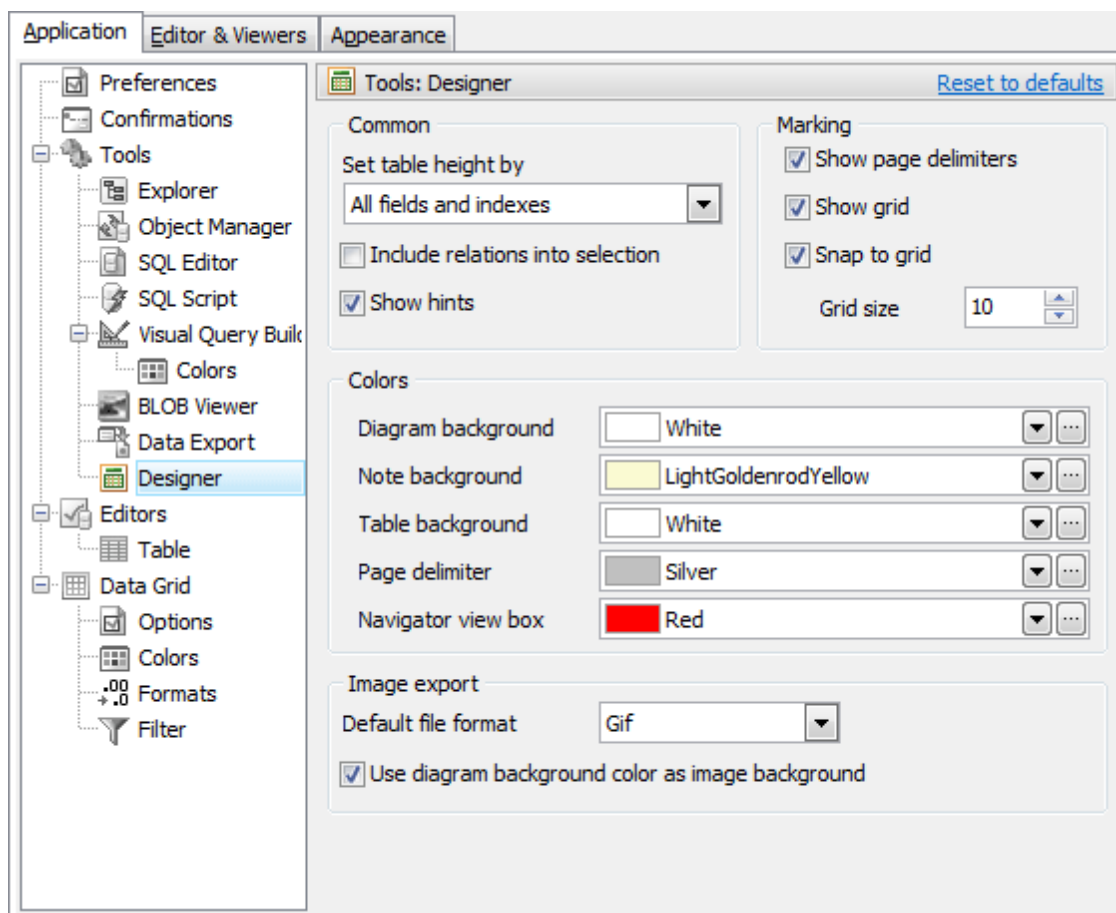
This window allows you to customize formats applied to exported data. Edit the format masks to adjust the result format in the way you need.

In *numeric* formats using digit placeholder (`#` or `0`) you can specify the format of number. For example, integer 1234567890 with `#####0` integer format is represented like 1 234 567 890. The locations of the leftmost '0' before the decimal point in the format string and the rightmost '0' after the decimal point in the format string determine the range of digits that are always present in the output string.

Conversion and their description for *date*, *time* and *date time* format:

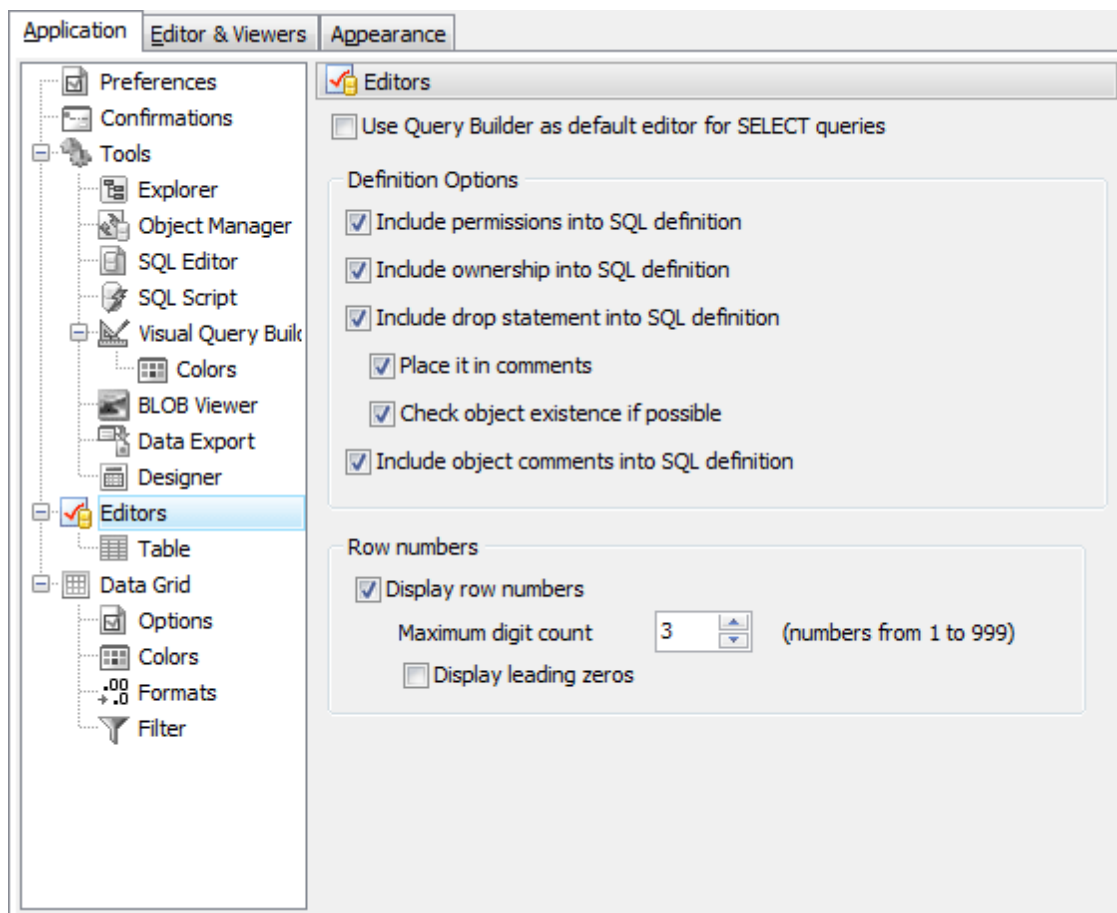
dd	day of the month, represented by 1 or 2 symbols. For example, the first day of month is 1
DD	day of the month, represented only by 2 symbols. For example, the first day of month is 01
mm	minutes





### 10.1.5 Object Editors

Below you will find a detailed description of the following object editors options.



☒ **Open each object editor in a new window**

With this option checked a new child window opens each time you open an object for editing, otherwise the edited object is being changed in the existing object editor (except the existing object editor is in modified state).

☒ **Use Query Builder as default editor for SELECT queries**

With this option enabled all the SELECT queries will be opened in [Visual Query Builder](#)<sup>[202]</sup> instead of [SQL Editor](#)<sup>[196]</sup>.

☒ **Include drop statement**

If checked, the SQL definition includes the drop statement.

☒ **Place it in comments**

With this option drop statement will be placed in comments of the SQL definition.

☒ **Include object comments into SQL definition**

With this option enabled comments that are specified for the object and object subitems are placed in SQL definition.

**Row numbers**

This options group allows you to manage the row numbering of the subobjects lists such as fields, indexes, parameters and so on.

To enable/disable the numbering, use [Display row numbers](#) checkbox. You can set the

number columns width with [Maximum digit count](#). (I.e. for the value '3' the max column number will be 999).

For uniformity you can use the [Display leading zeros](#) option. With this option enabled and maximum digit count '3' your numbering column will be of the form: '001, 002, 003, ...'.

### 10.1.5.1 Table

#### Initial tab

Specifies which tab of [Table Editor](#)<sup>[68]</sup> should be active when it is initially opened.

#### ☒ Retrieve record count before loading data in the data grid

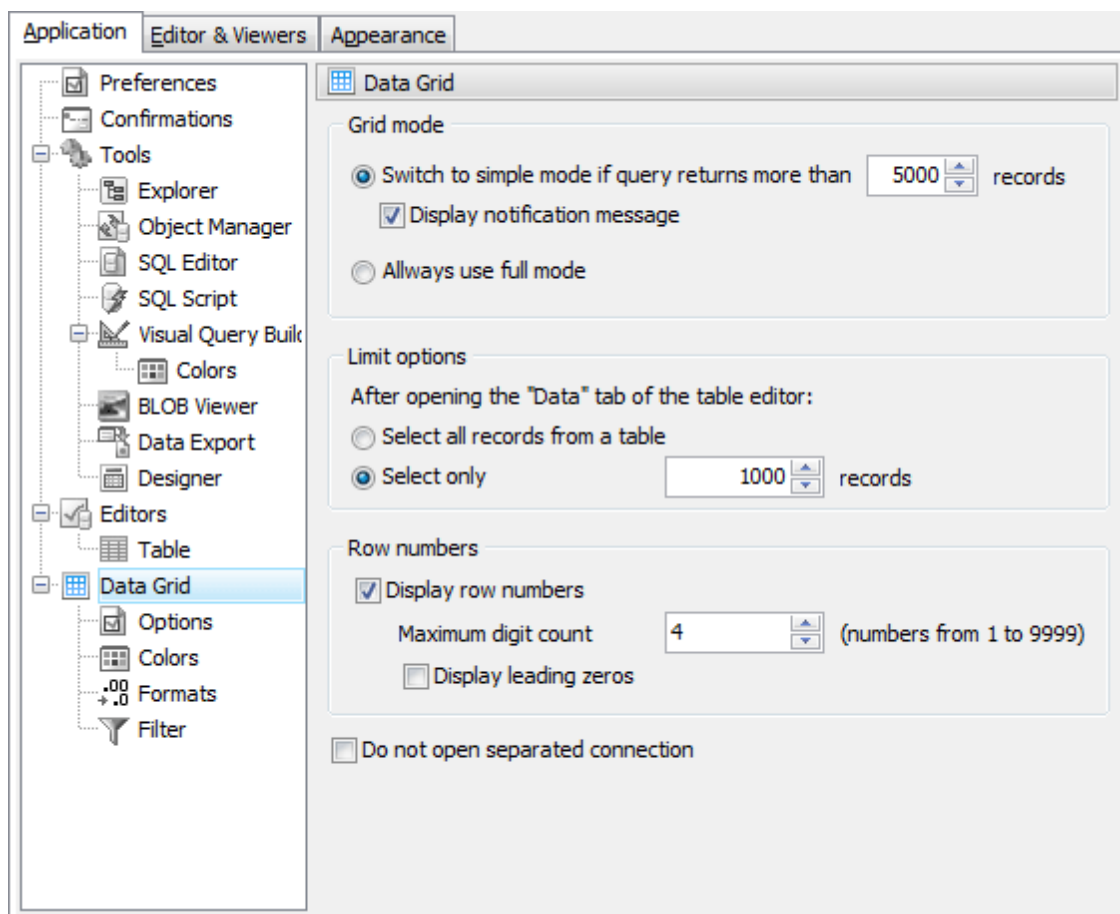
With this option enabled all the `SELECT COUNT(*)...` query is executed before loading data in the data grid.

#### Default field type

Specifies the field type appearing in [Field Editor](#)<sup>[72]</sup> by default.

### 10.1.6 Data Grid

Below you will find a detailed description of the following data grid options.



DB2 Maestro provides you with [two grid modes](#)<sup>[213]</sup> of viewing data:

- Full grid mode allows you to group, filter and sort data in a usual way.
- Simple mode is provided for working with large records number. For data fetching speed-up, filtering, sorting, and grouping features are not enabled in this mode.

You can use [notification message](#) to indicate simple mode.

Set the number of records to switch to simple mode automatically or select [Always use full mode](#).

#### [Limit options](#)

Allows you either to select all records from table after opening the Data tab, or select only specified number of rows on one page with an ability to rotate pages and view all data.

#### [Row numbers](#)

This options group allows you to manage grid rows numbering.

To enable/disable the numbering, use [Display row numbers](#) checkbox. You can set the number columns width with [Maximum digit count](#). (I.e. for the value '3' the max column number will be 999).

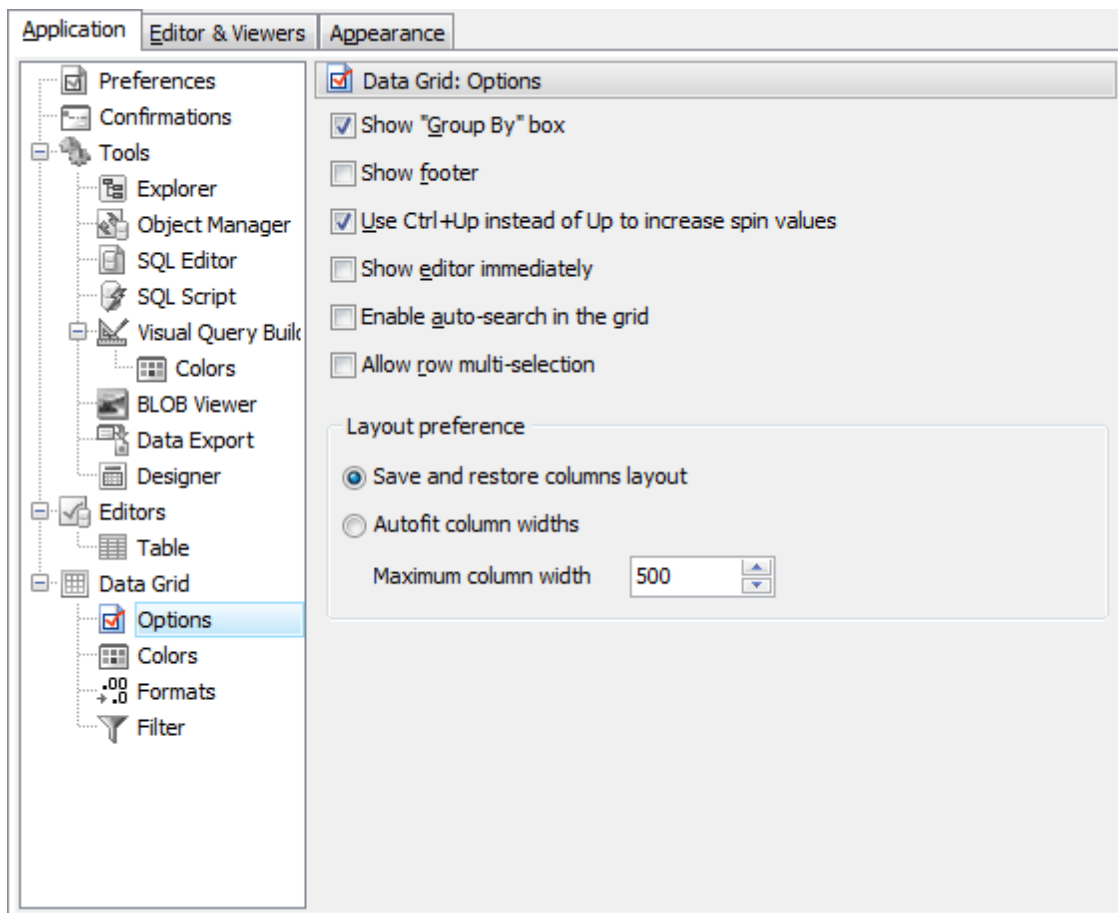
For uniformity you can use the [Display leading zeros](#) option. With this option enabled and maximum digit count '3' your numbering column will be of the form: '001, 002, 003, ...'.

#### ☒ [Do not open separated connection](#)

With this option enabled a new connections for fetching data is not opened. This gives you an ability to work with data a little bit faster, because time for opening a new connections is not demanded.

### 10.1.6.1 Options

Below you will find a detailed decryption of the data grid options.



☒ **Show "Group By" box**

Shows the box on the top of the grid view for grouping data by fields.

☒ **Show footer**

Shows the footer on the bottom of the grid view.

☒ **Use Ctrl+Up instead of Up to increase spin values**

Allows you to use Ctrl+Up and Ctrl+Down key combinations for editing the spin for numeric fields.

☒ **Show editor immediately**

Allows editing the cell value right after the cell is clicked.

☒ **Enable auto-search in the grid**

Allows you to search records in the grid by the first letters.

☒ **Allow row multi-selection**

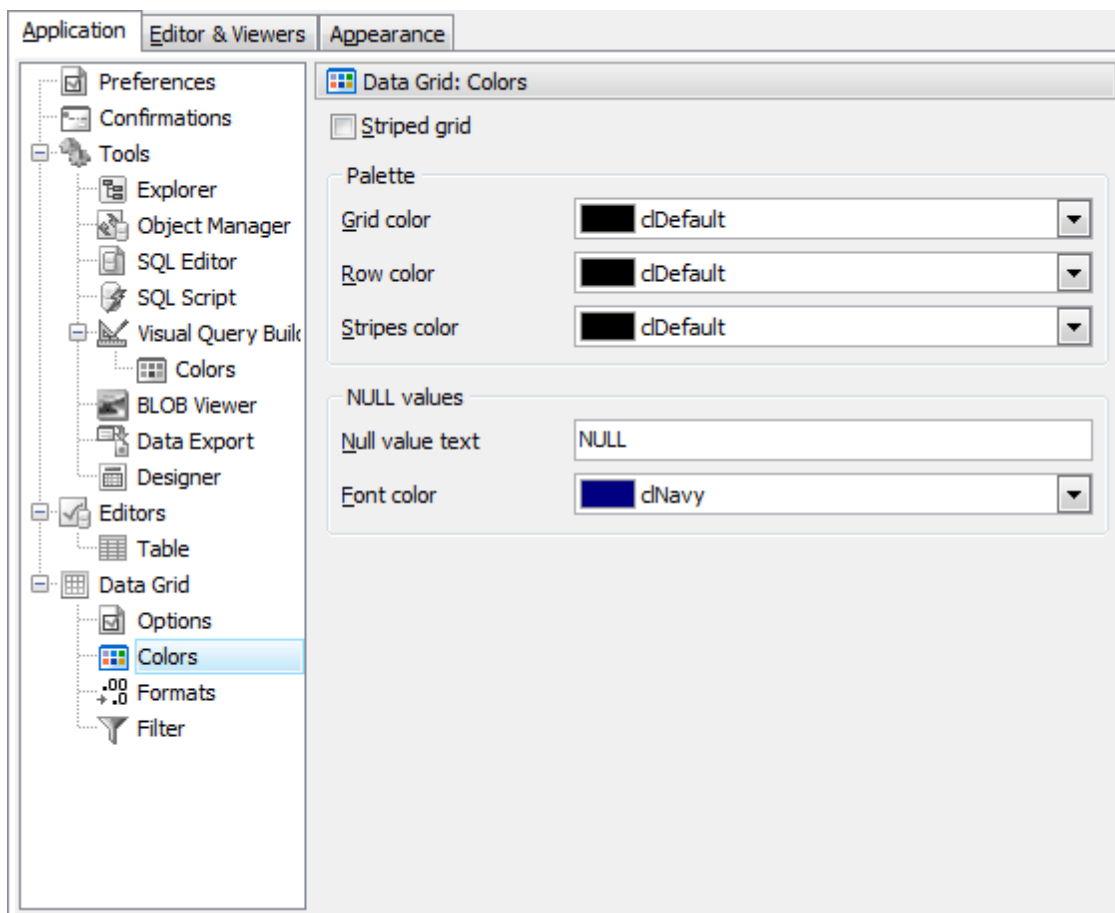
Allows you to select multiple records using the Ctrl and Shift keys.

**Layout preference**

Select whether DB2 Maestro should remember the column positions for the grids or fit them automatically.

### 10.1.6.2 Colors

Below you will find a detailed decryption of the following colors options.



☒ **Striped grid**

Displays the odd grid rows in a different color defined by the [Stripes color](#) option.

**Grid color**

Defines the background color of the data grid.

**Row color**

Defines the color of the selected row in the data grid.

**Stripes color**

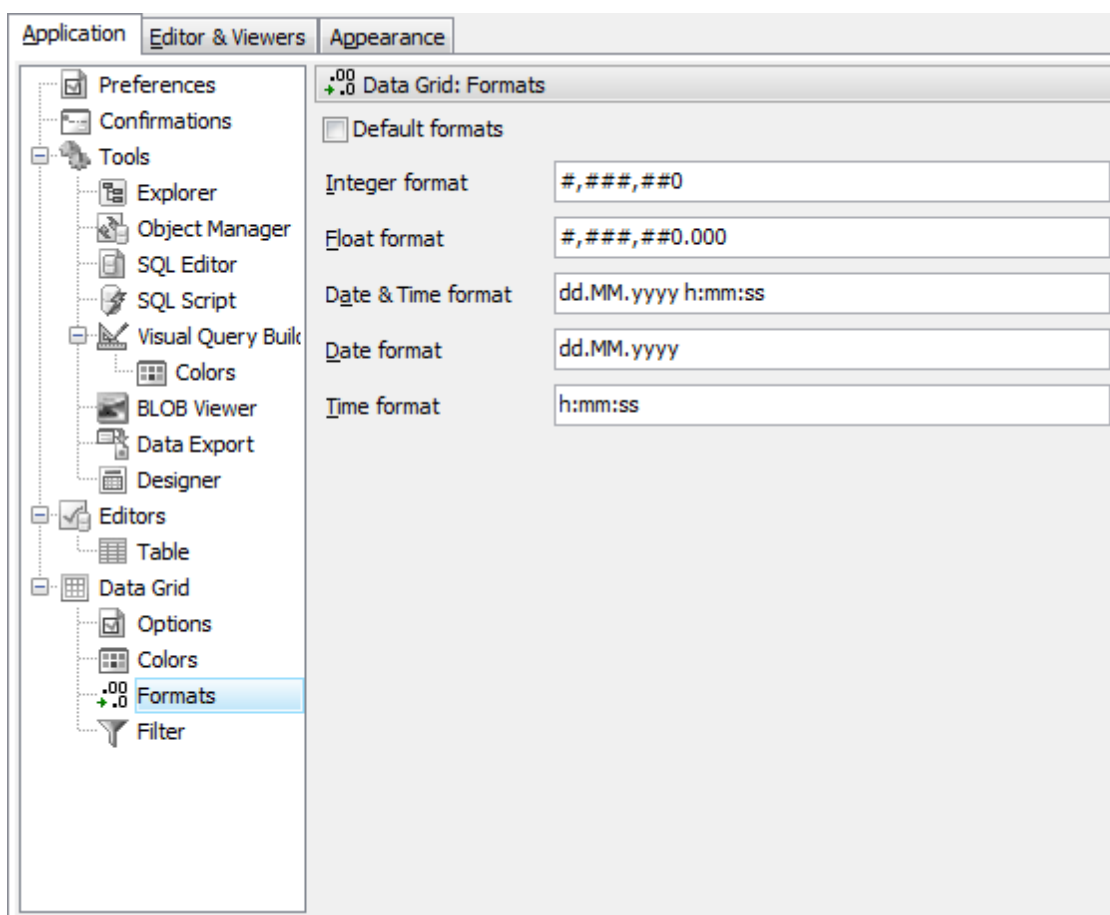
Defines the color of the odd rows if the [Striped Grid](#) option is on.

**Null values**

Use [Null value text](#) to define the text that stand for the NULL value and use [Font color](#) to set the color for displaying this text.

### 10.1.6.3 Formats

Below you will find a detailed decryption of the following formats options.



This window allows you to customize formats applied to data in grid. Edit the format masks to adjust the result format in the way you need.

In *numeric* formats using digit placeholder (# or 0) you can specify the format of number. For example, integer 1234567890 with # ### ##0 integer format is represented like 1 234 567 890. The locations of the leftmost '0' before the decimal point in the format string and the rightmost '0' after the decimal point in the format string determine the range of digits that are always present in the output string.

Conversion and their description for *date*, *time* and *date time* format:

dd	day of the month, represented by 1 or 2 symbols. For example, the first day of month is 1
DD	day of the month, represented only by 2 symbols. For example, the first day of month is 01
mm	minutes
MM	month
yy	year, represented by 2 symbols. For example, 2006 year will be 06



---

"Today", "Tomorrow", "Last 30 day", "Last week", "Next week", and others; check the [Ignore time part](#) box to neglect time part of timestamp data under the filtering.

By default filter buttons are shown at all columns header, to [show filter button only in selected column](#), check the corresponding option.

You can specify the case sensitivity of the grid filter with the [Case insensitive](#) checkbox (ON by default).

## 10.2 Editors & Viewers

The [Editors & Viewers](#) section allows you to set the parameters of viewing and editing the SQL statements within DB2 Maestro.

- [General](#) <sup>[315]</sup>
- [Display](#) <sup>[316]</sup>
- [SQL highlight](#) <sup>[317]</sup>
- [PHP highlight](#) <sup>[319]</sup>
- [XML highlight](#) <sup>[318]</sup>
- [Code Insight](#) <sup>[320]</sup>
- [Code Folding](#) <sup>[321]</sup>

**See also:** [SQL Editor](#) <sup>[196]</sup>, [SQL Script Editor](#) <sup>[248]</sup>, [Visual Query Builder](#) <sup>[202]</sup>, [Table Editor](#) <sup>[68]</sup>.

### 10.2.1 General

If the [Auto indent](#) option is checked, each new indentation is the same as the previous when editing SQL text.

☒ [Insert mode](#)

If this option is checked, insert symbols mode is default on.

☒ [Use syntax highlight](#)

Enables syntax highlight in the object editor window.

☒ [Always show links](#)

If this option is checked, hyperlinks are displayed in the editor window. To open a link click it with the **Ctrl** button pressed.

☒ [Show line numbers](#)

If this option is checked, line numbers are displayed in the SQL text editor window.

☒ [Use smart tabs](#)

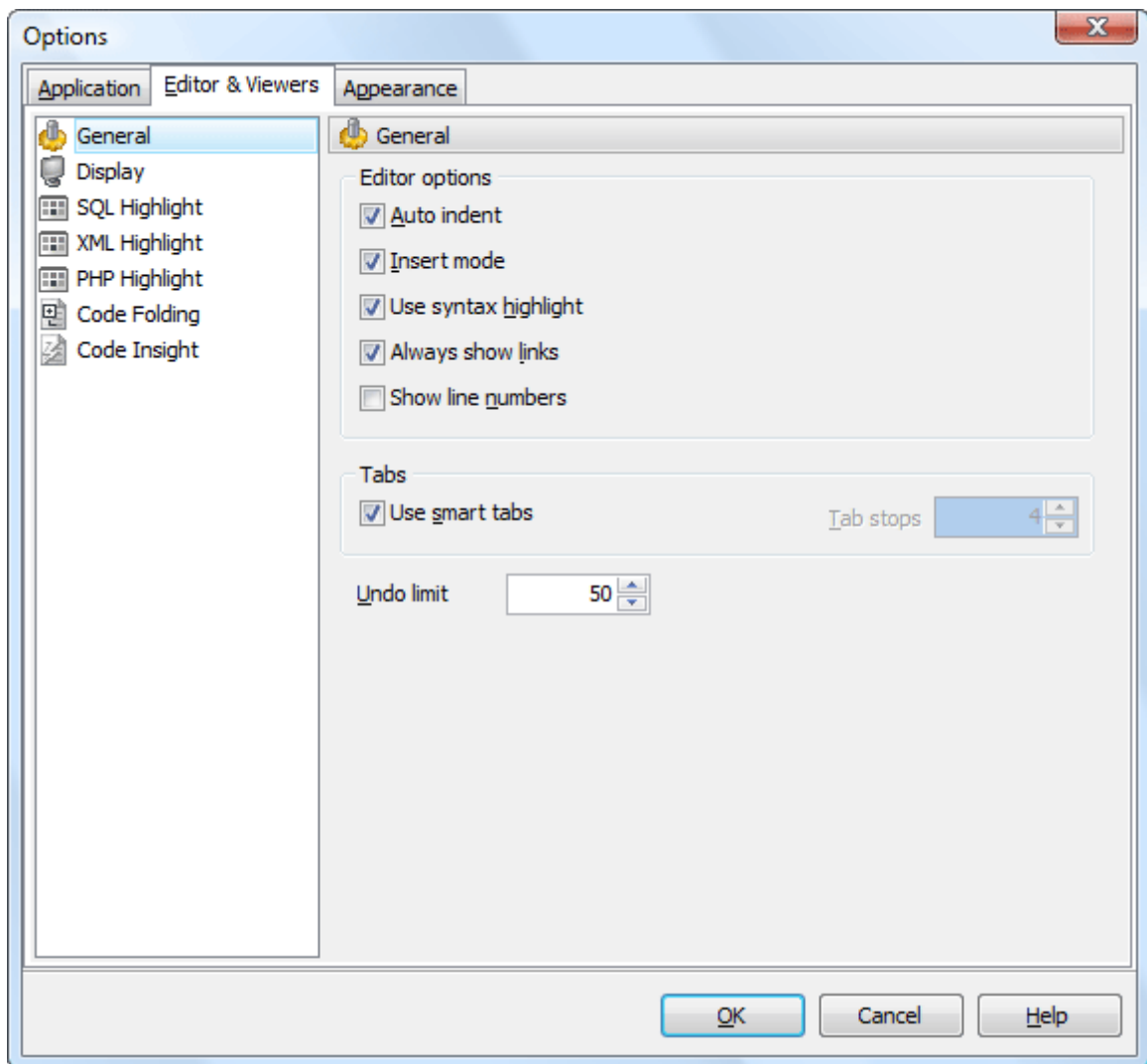
With this option on the number of tab stops is calculated automatically, depending on the previous line tab.

[Tab Stops](#)

Defines the tab length, used when editing text.

[Undo Limit](#)

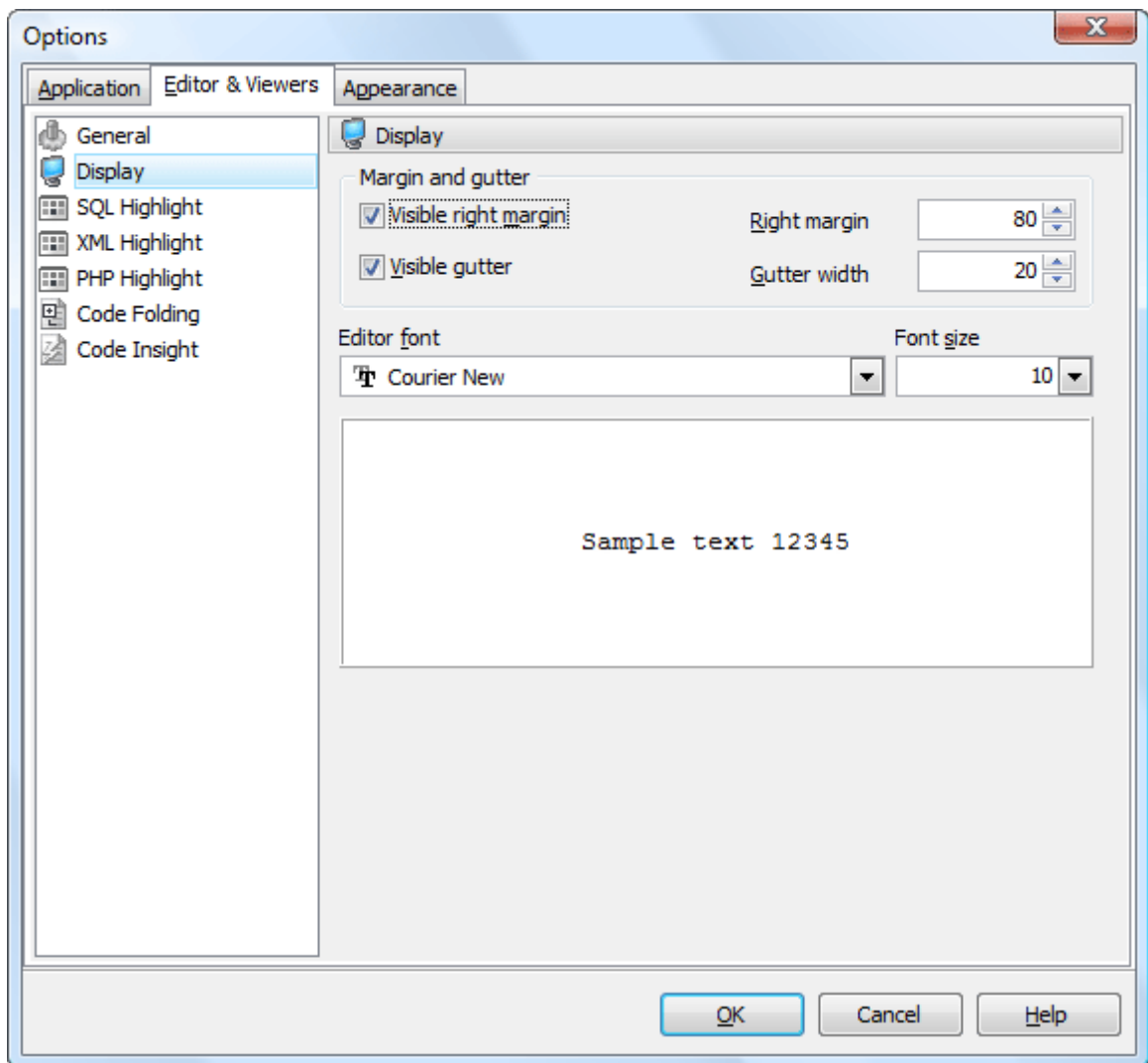
Defines the maximum number of changes possible to be undone.



### 10.2.2 Display

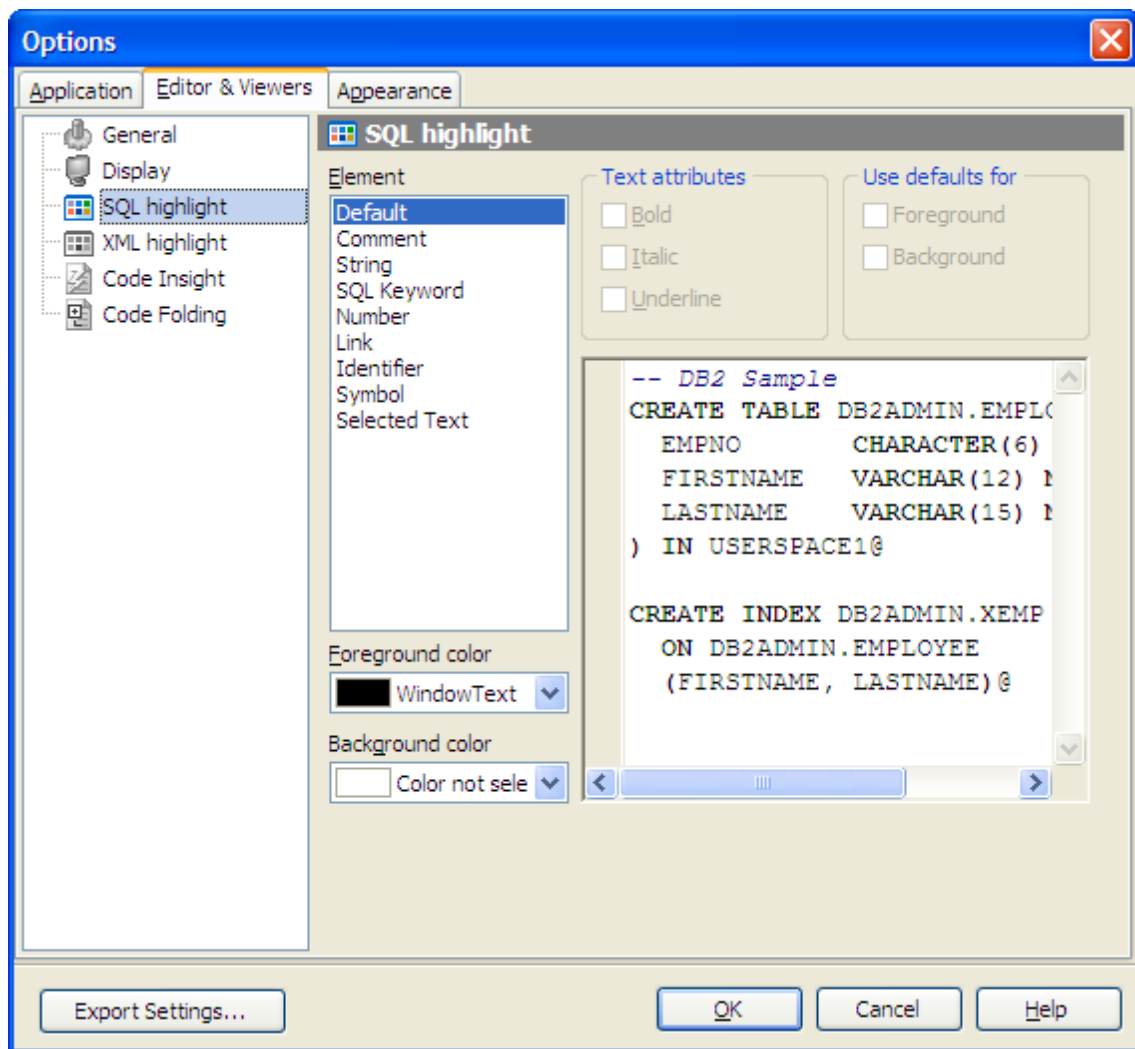
You can disable/enable the right text margin and the gutter of the editor area, set the position of the right text margin as [Right margin](#), and [the Gutter width](#).

Use the [Editor font](#) and [Font size](#) to define the font used in all program editors and viewers. The panel below displays the sample of the selected font.



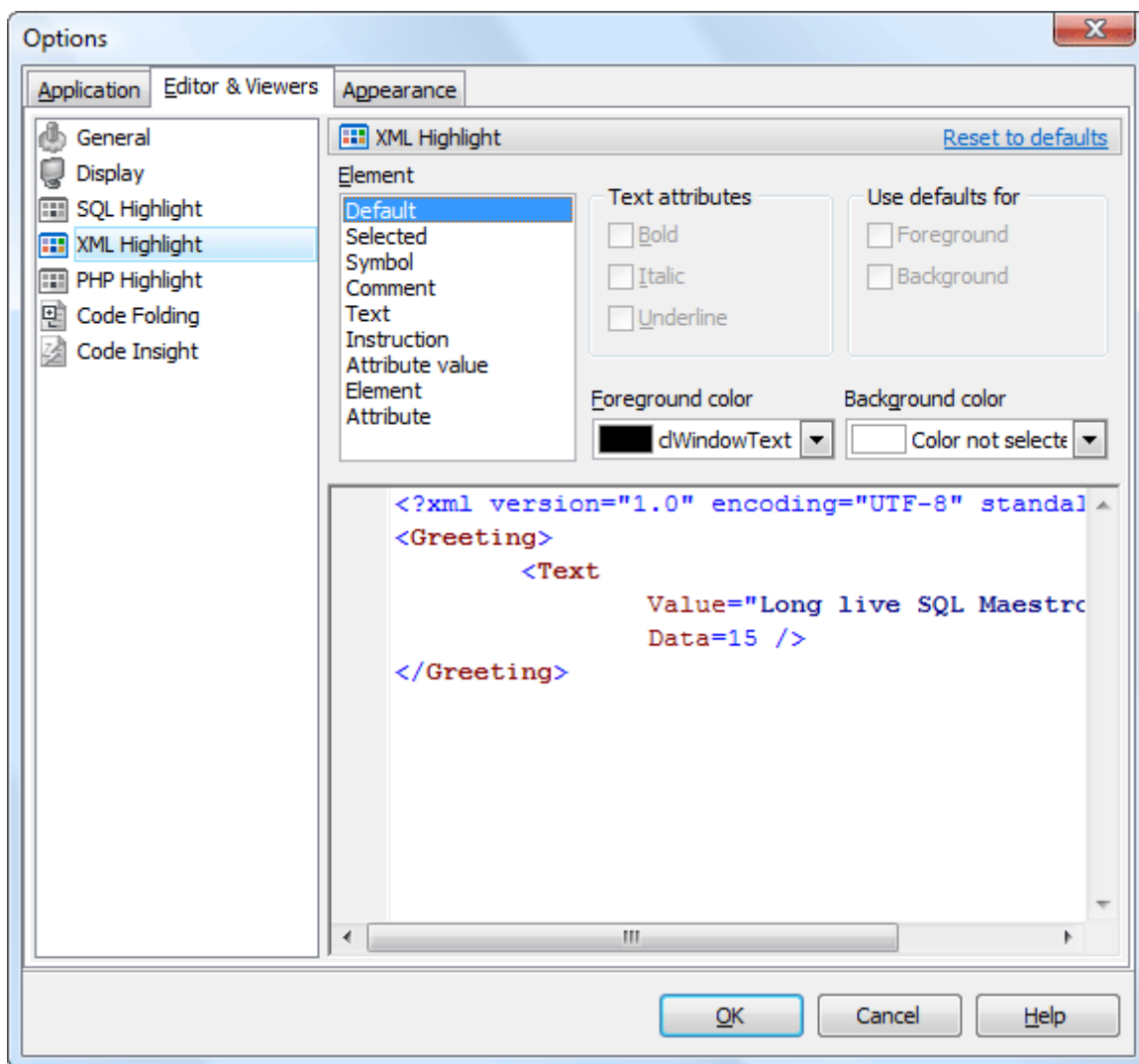
### 10.2.3 SQL highlight

Use the [SQL highlight](#) item to customize syntax highlight in all SQL editors and viewers, e.g. in [SQL Editor](#), [Query Builder](#), [Table Editor](#) and others. Select the text element from the list, e.g. *comment* or *SQL keyword* and adjust its foreground color, background color and text attributes according to your preferences.



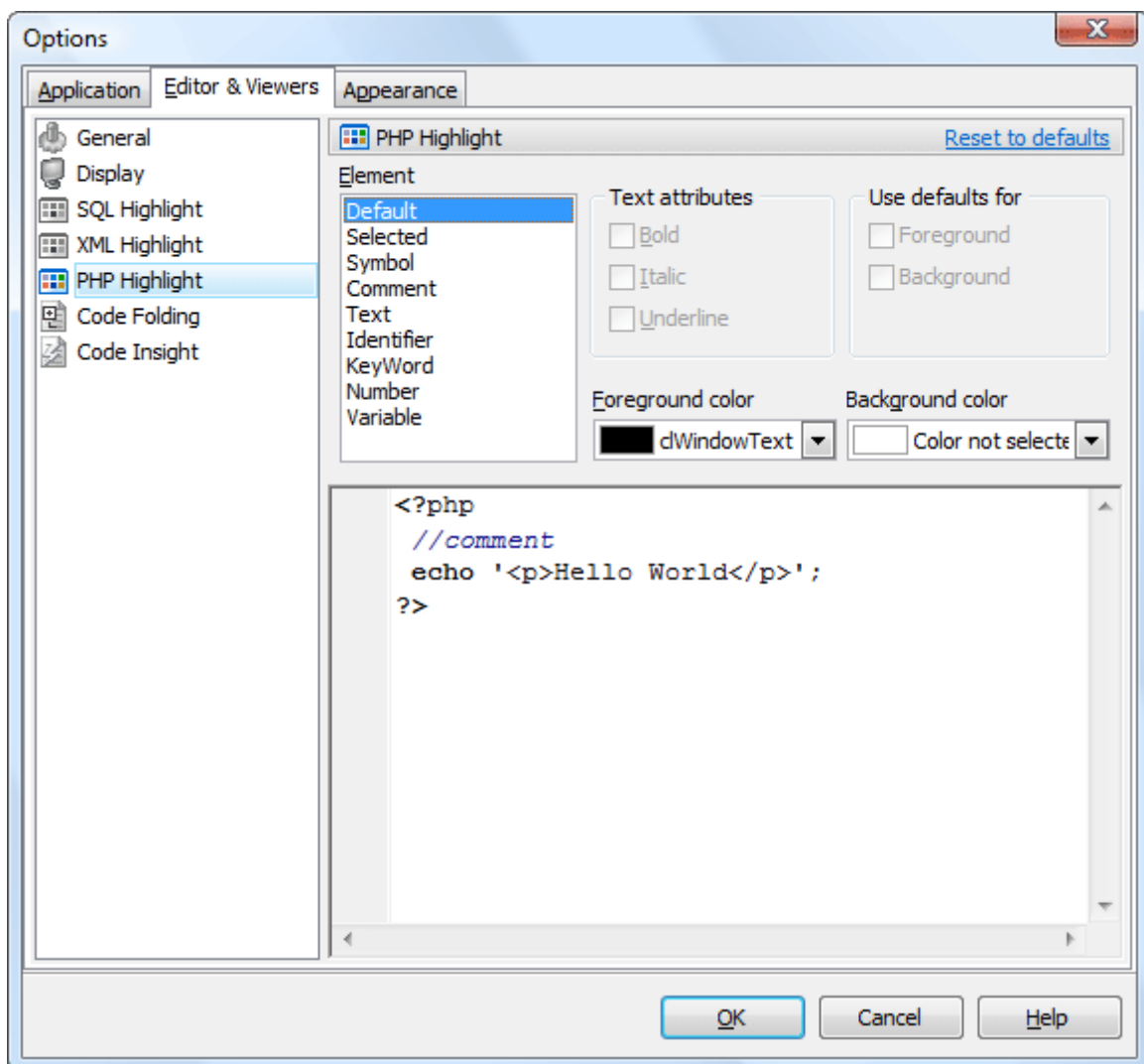
#### 10.2.4 XML highlight

Use the [XML highlight](#) item to customize XML syntax highlight for the text representation of BLOBs in [BLOB Viewer/Editor](#). Select the text element from the list, e.g. attribute or attribute value and adjust its foreground color, background color and text attributes according to your wishes.



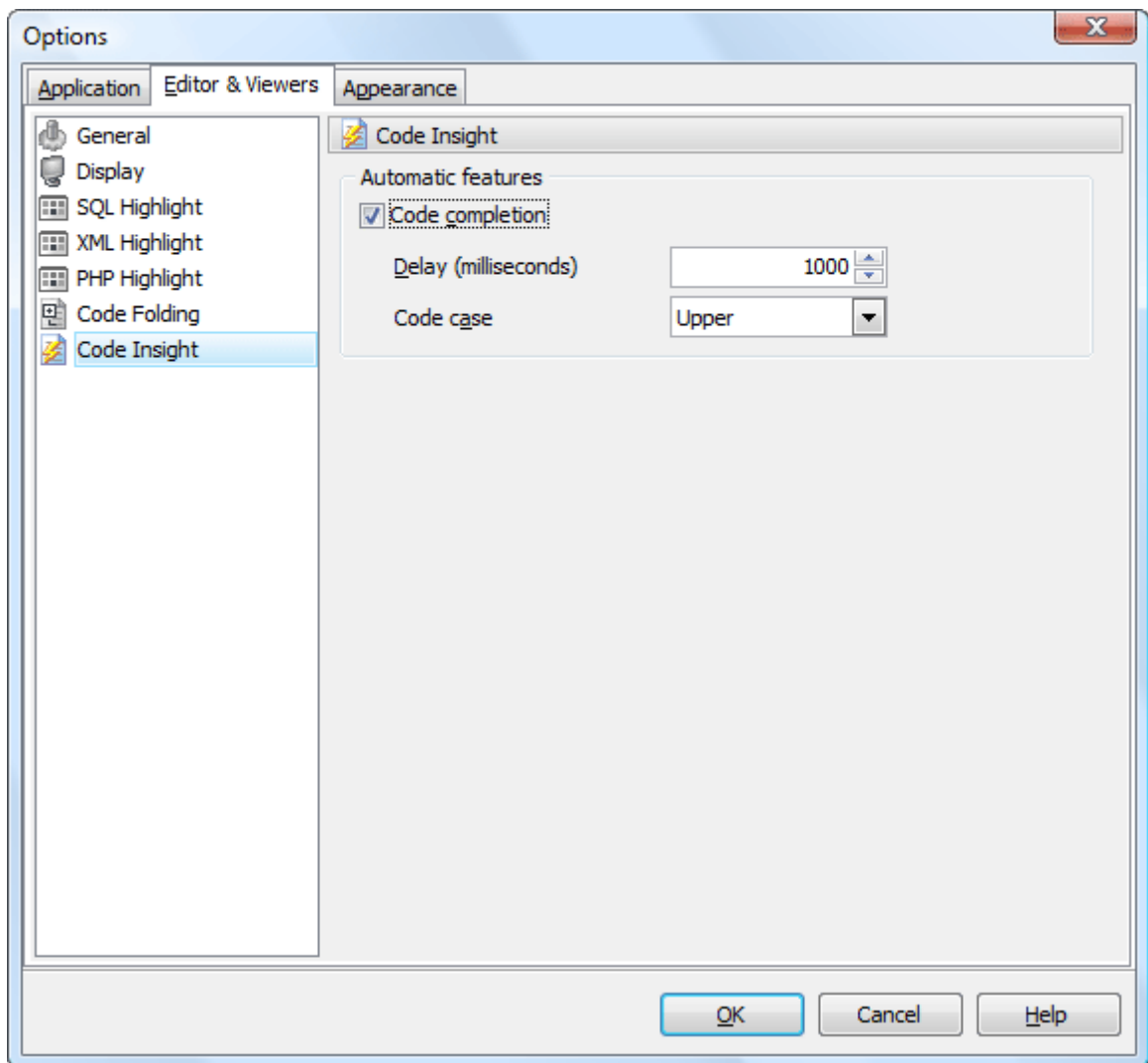
### 10.2.5 PHP highlight

Use the **PHP highlight** item to customize PHP syntax highlight for the text representation of BLOBs in **BLOB Viewer/Editor**. Select the text element from the list (e.g. Keyword, Comment, Identifier), and adjust its foreground color, background color and text attributes according to your wishes.



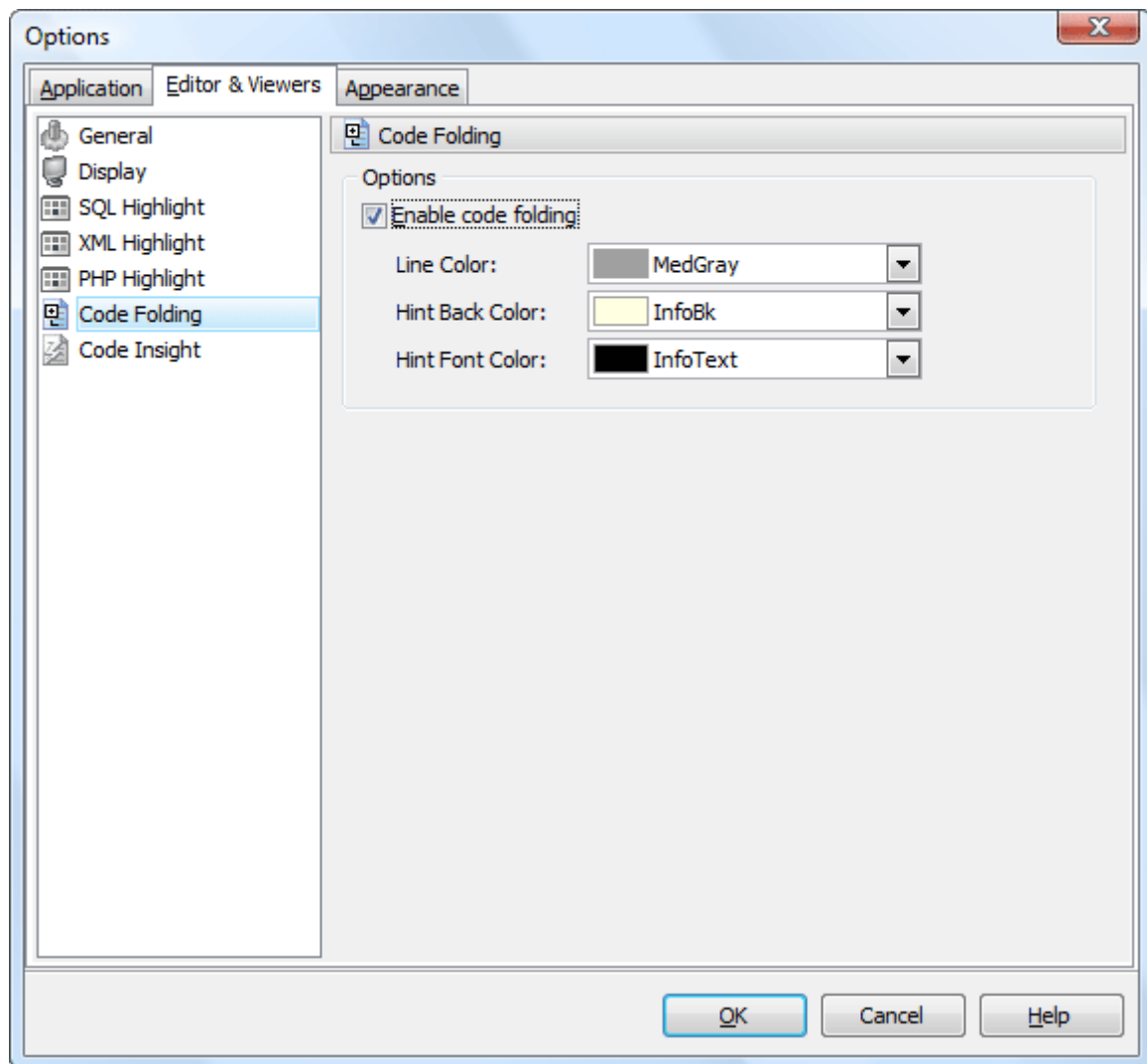
### 10.2.6 Code Insight

You can disable/enable the code completion with the corresponding option and also set the time it appears as *Delay*, and case of words inserted automatically.



### 10.2.7 Code Folding

The [Code Folding](#) item group makes it possible both to view the whole text and to divide it into logical parts (regions). Each part can be collapsed and extended. In extended mode the whole text is displayed (set by default), in collapsed mode the text is hidden behind one text line denoting the first line of the collapsed region.



You can enable/disable code folding in SQL editors and viewers and customize the colors of its items.

## 10.3 Appearance

The [Appearance](#) section allows you to customize the application interface style to your preferences.

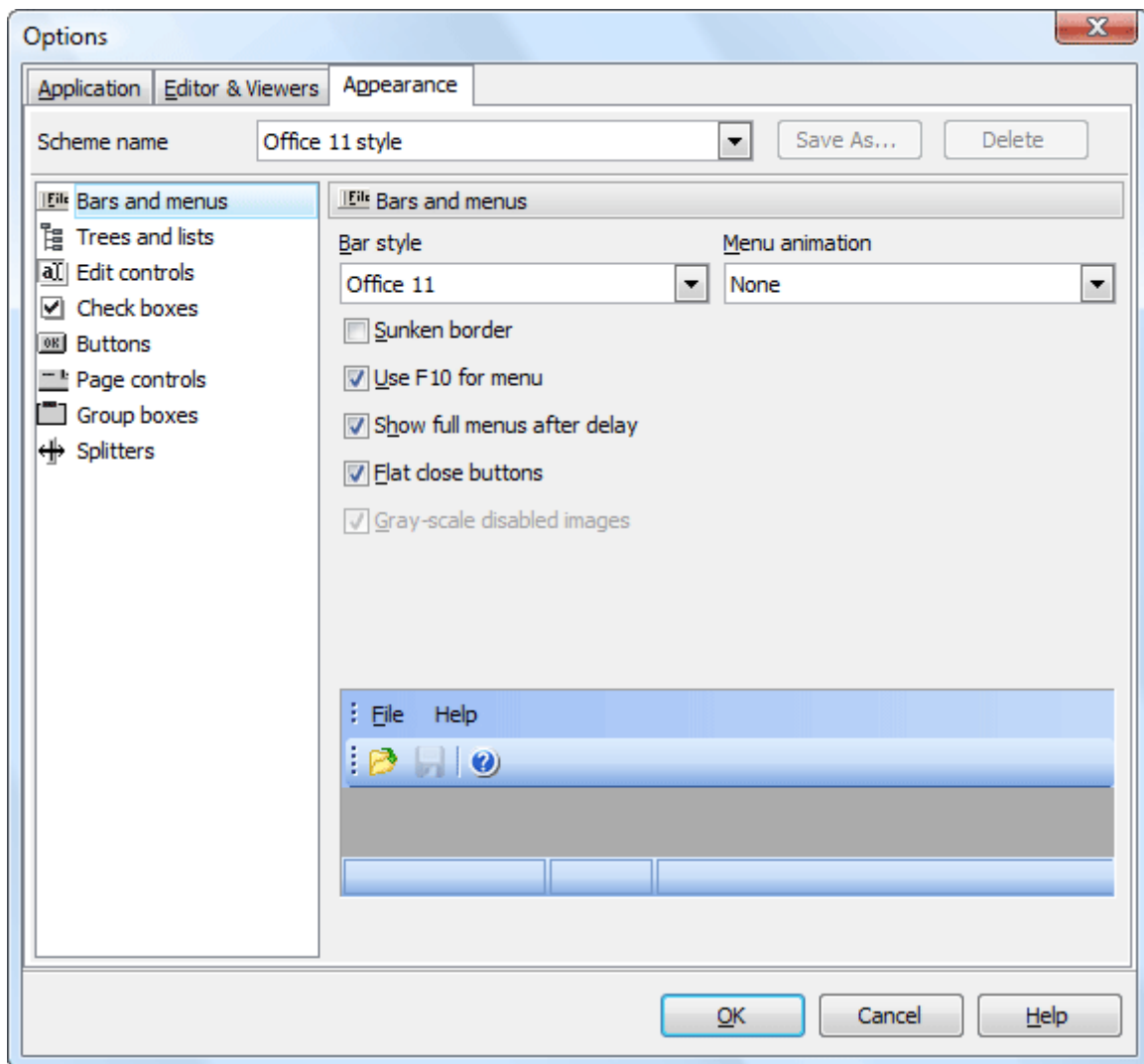
Use the [Scheme name](#) box to select the interface scheme you prefer: *Office XP style*, *Windows XP native style*, etc. You can create your own interface schemes by customizing any visual options ([Bars and menus](#), [Trees and lists](#), [Edit controls](#), [Check boxes](#), [Buttons](#), etc.) and clicking the [Save As](#) button. All the customized options are displayed on the sample panel.

- [Bars and menus](#) 
- [Trees and lists](#) 
- [Edit controls](#) 
- [Check boxes](#) 
- [Buttons](#) 
- [Page controls](#) 
- [Group boxes](#) 
- [Splitters](#) 

### 10.3.1 Bars and menus

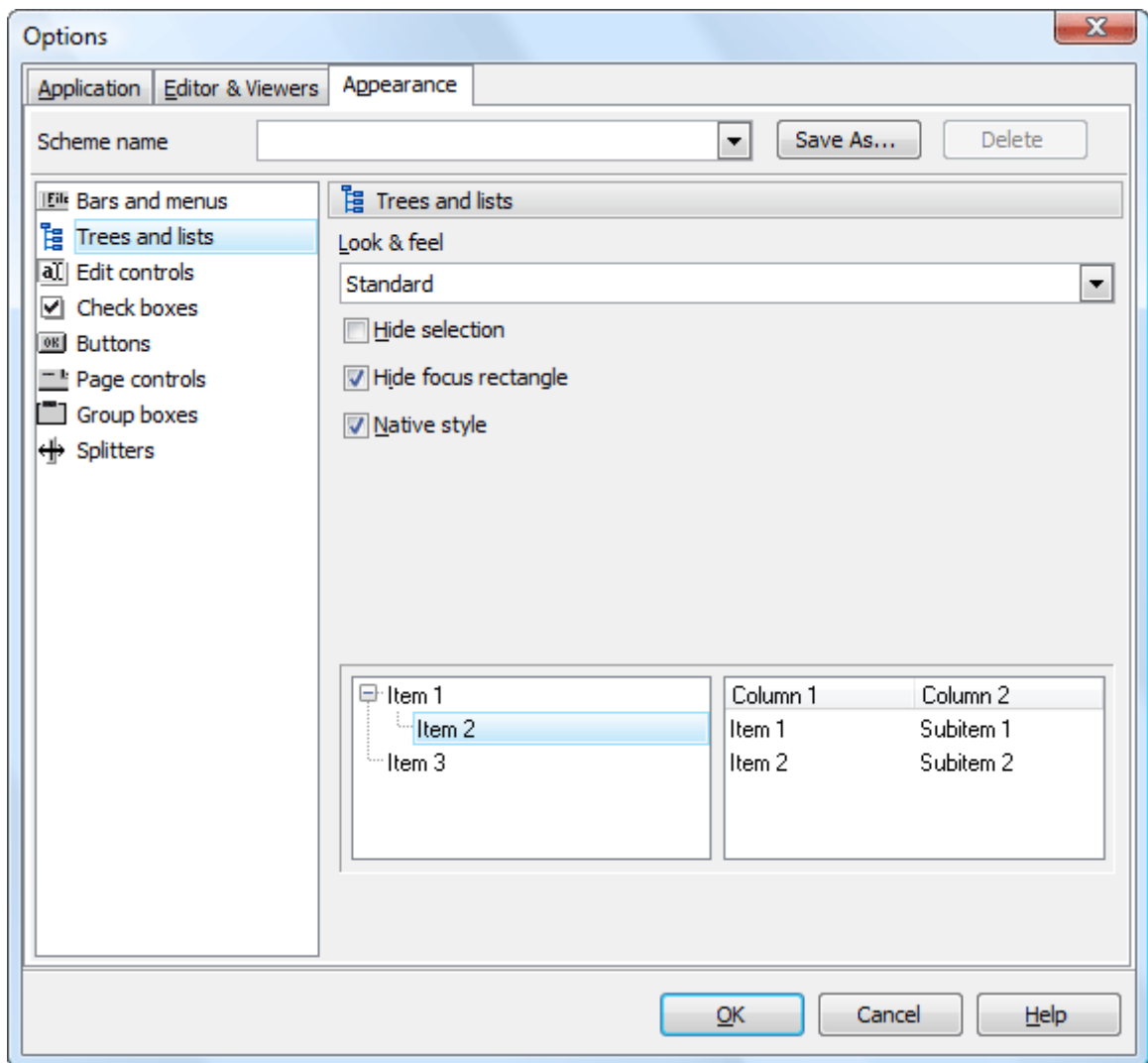
Use the [Bars and menus](#) item to customize DB2 Maestro toolbars style and menu animation.

The item allows you to select Bar style and menu animation from the corresponding drop-down lists and to enable or disable such options as sunken border, F10 key for opening menu, viewing full menus after delay, flat close buttons, gray-scale images.



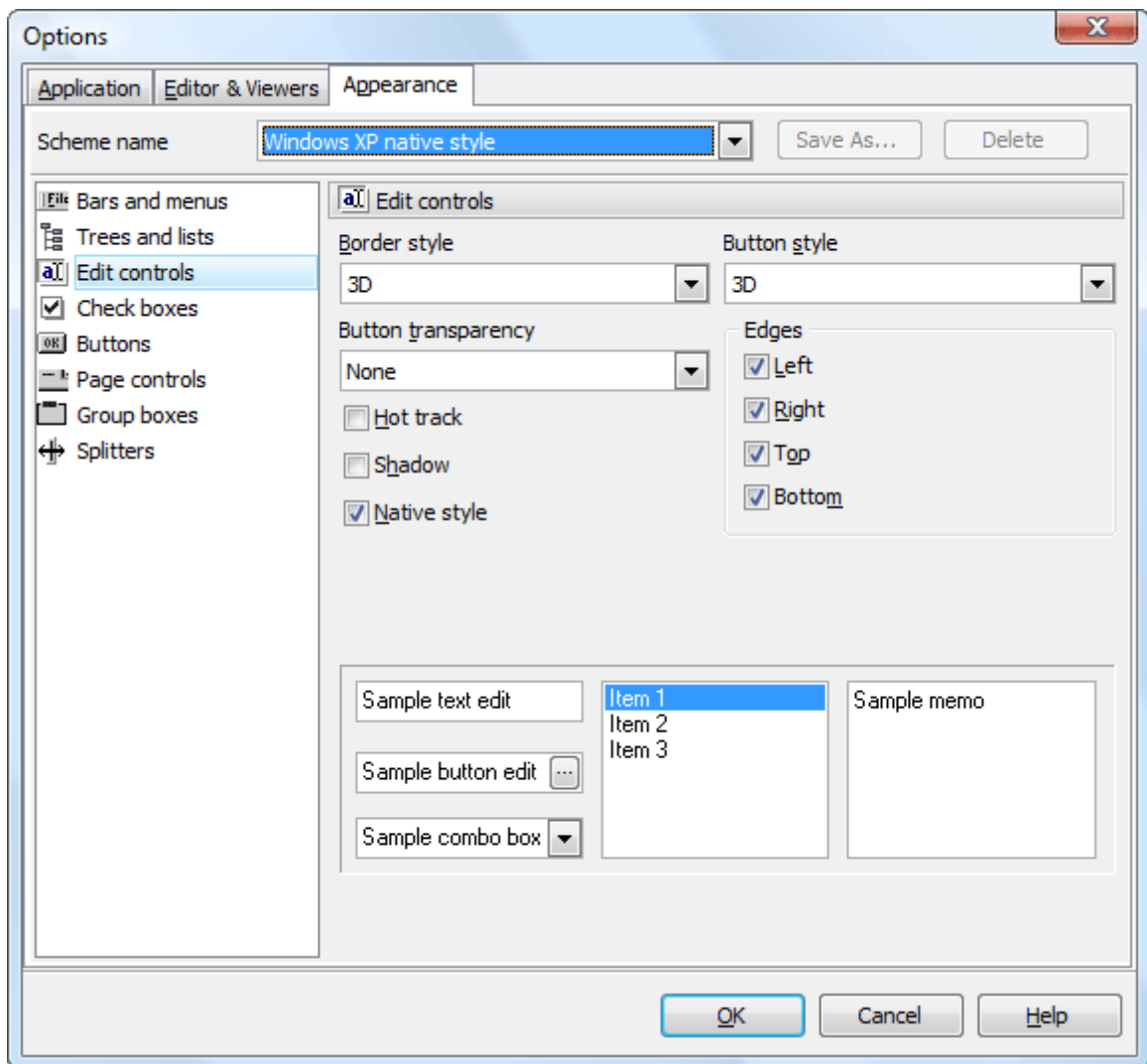
### 10.3.2 Trees and lists

Use the **Trees and lists** item to select various tree view options. Use the item to select *standard*, *flat* or *ultraflat* styles, check or uncheck the *hide selection*, *hide focus rectangle* and *native style* options.



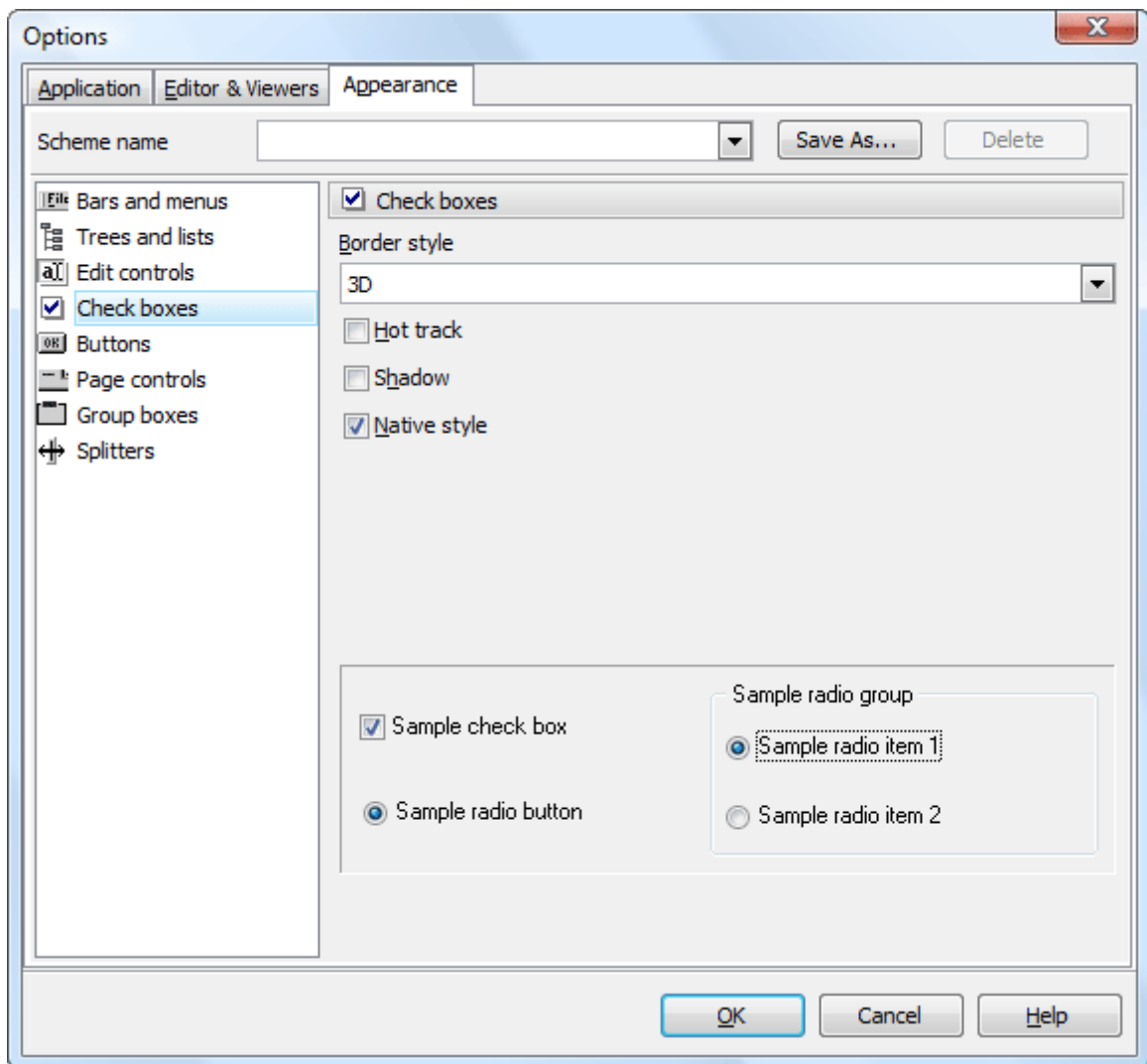
### 10.3.3 Edit controls

Use the [Edit controls](#) item to customize the appearance of different DB2 Maestro edit controls. The tab allows you to select the edit controls border style, button style and transparency, enable/disable hot tracks, shadows, native style and customize edges. It is also possible to define samples for the text edit, button edit and combo box controls.



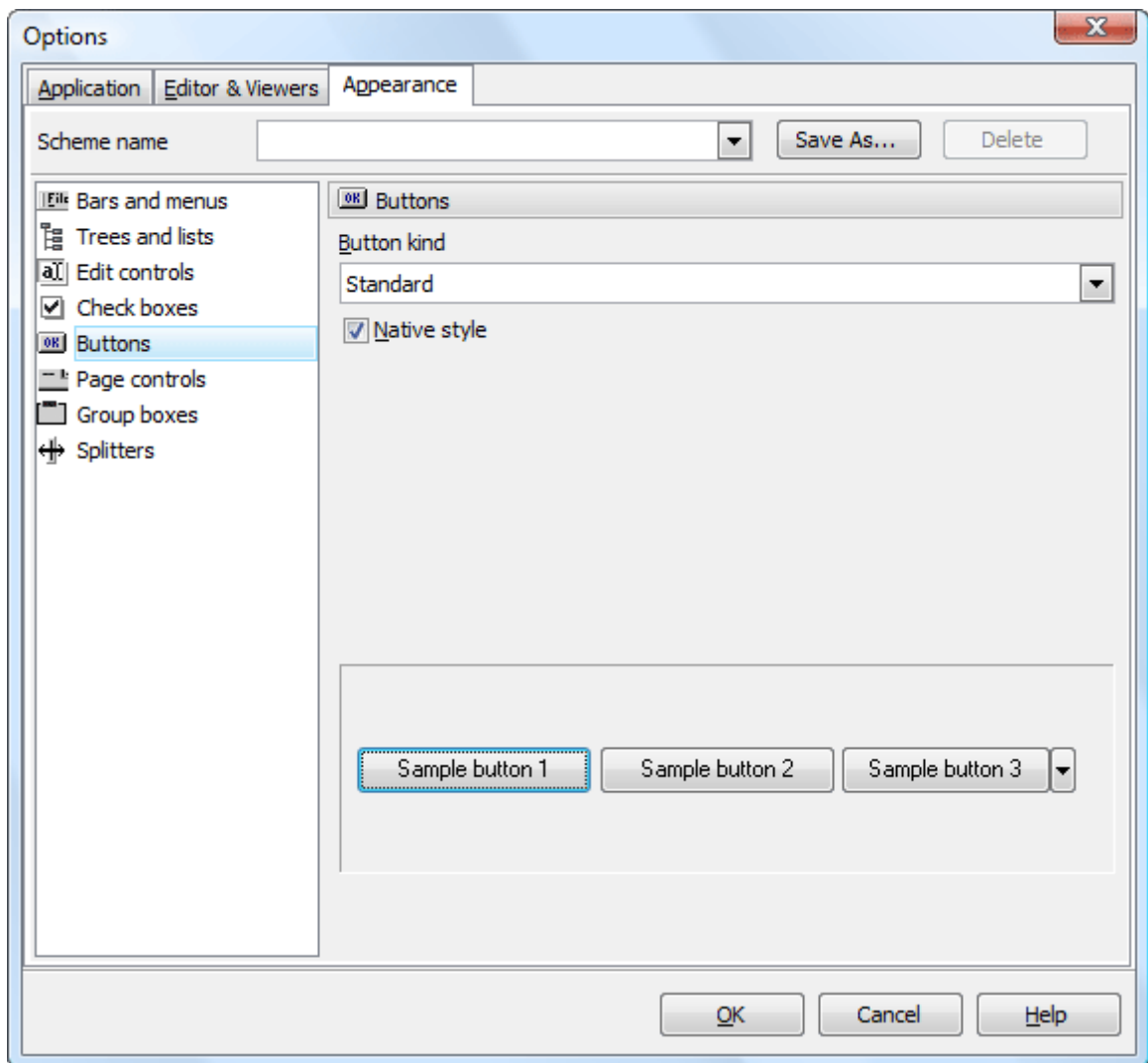
### 10.3.4 Check boxes

The [Check boxes](#) item allows you to customize the appearance of check boxes and radio buttons. The tab allows you to customize the appearance of check boxes: set border style, enable/disable hot tracks, shadows, native style. It is also possible to define samples for check boxes and radio buttons.



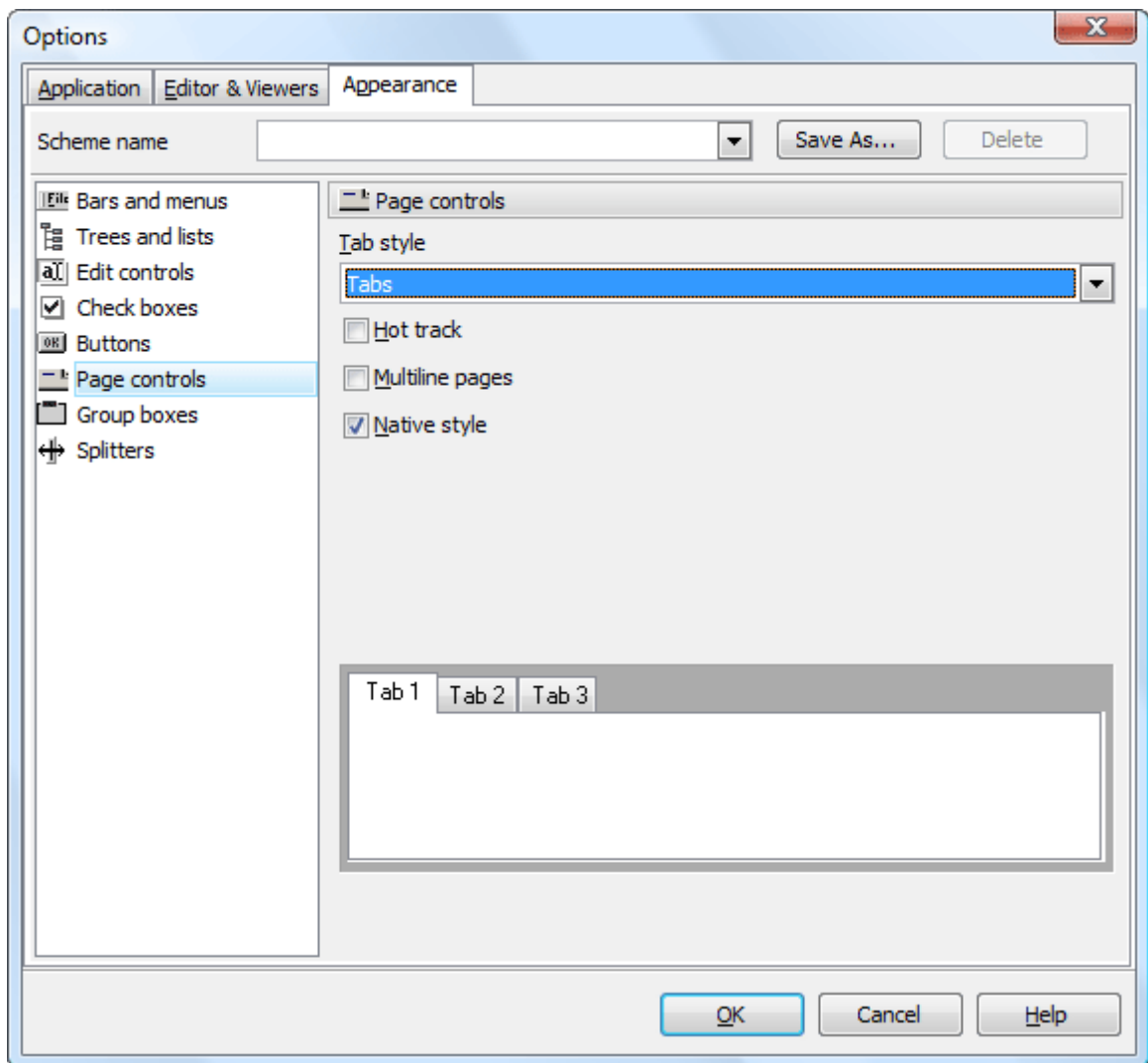
### 10.3.5 Buttons

Use the [Buttons](#) item to customize DB2 Maestro buttons. The tab allows you to adjust the appearance of buttons and define sample buttons as well.



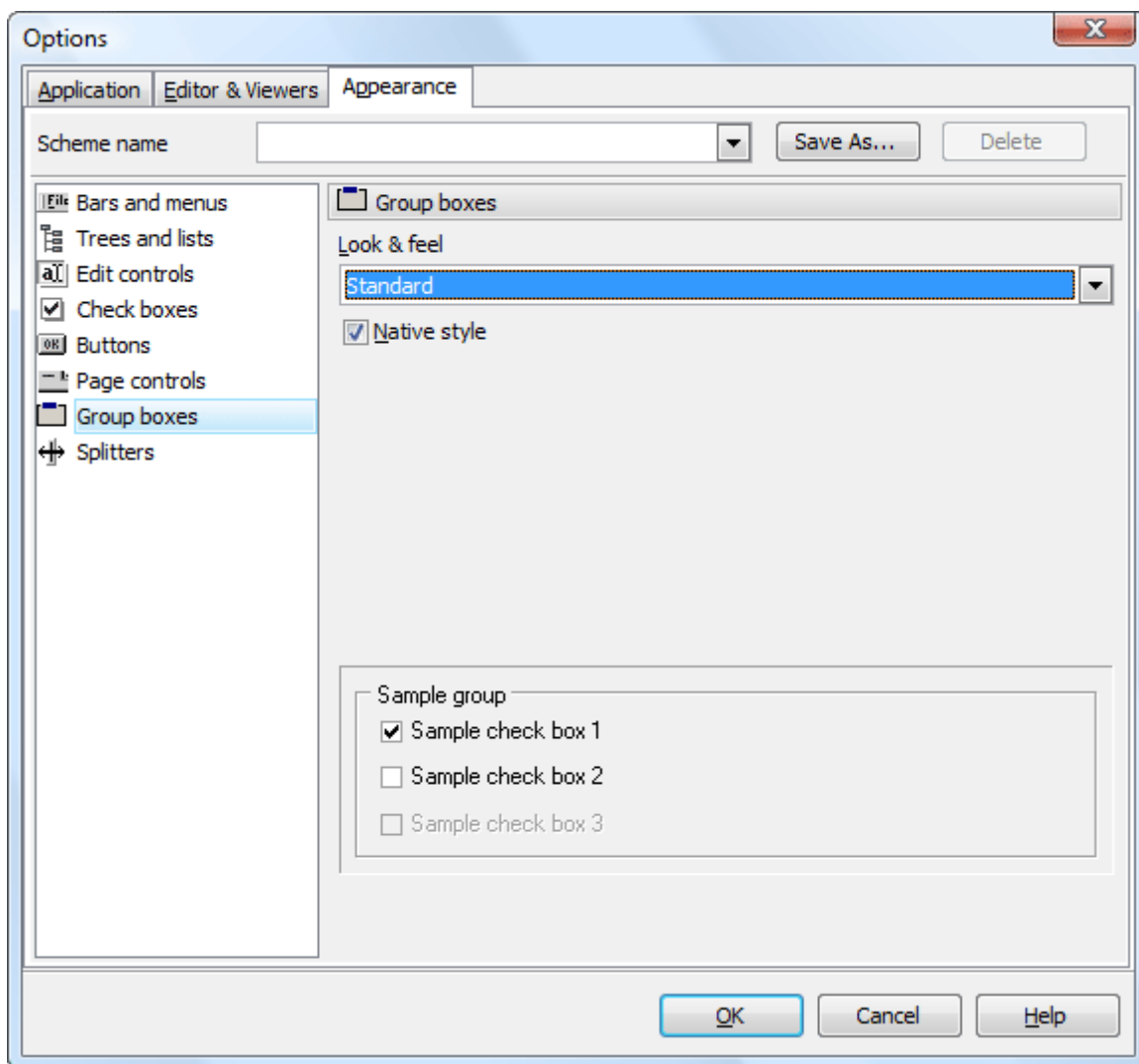
### 10.3.6 Page controls

The [Page controls](#) item allows you to customize the style of all DB2 Maestro page controls. The tab allows you to select tab styles, enable/disable hot track, multi-line pages and native style.



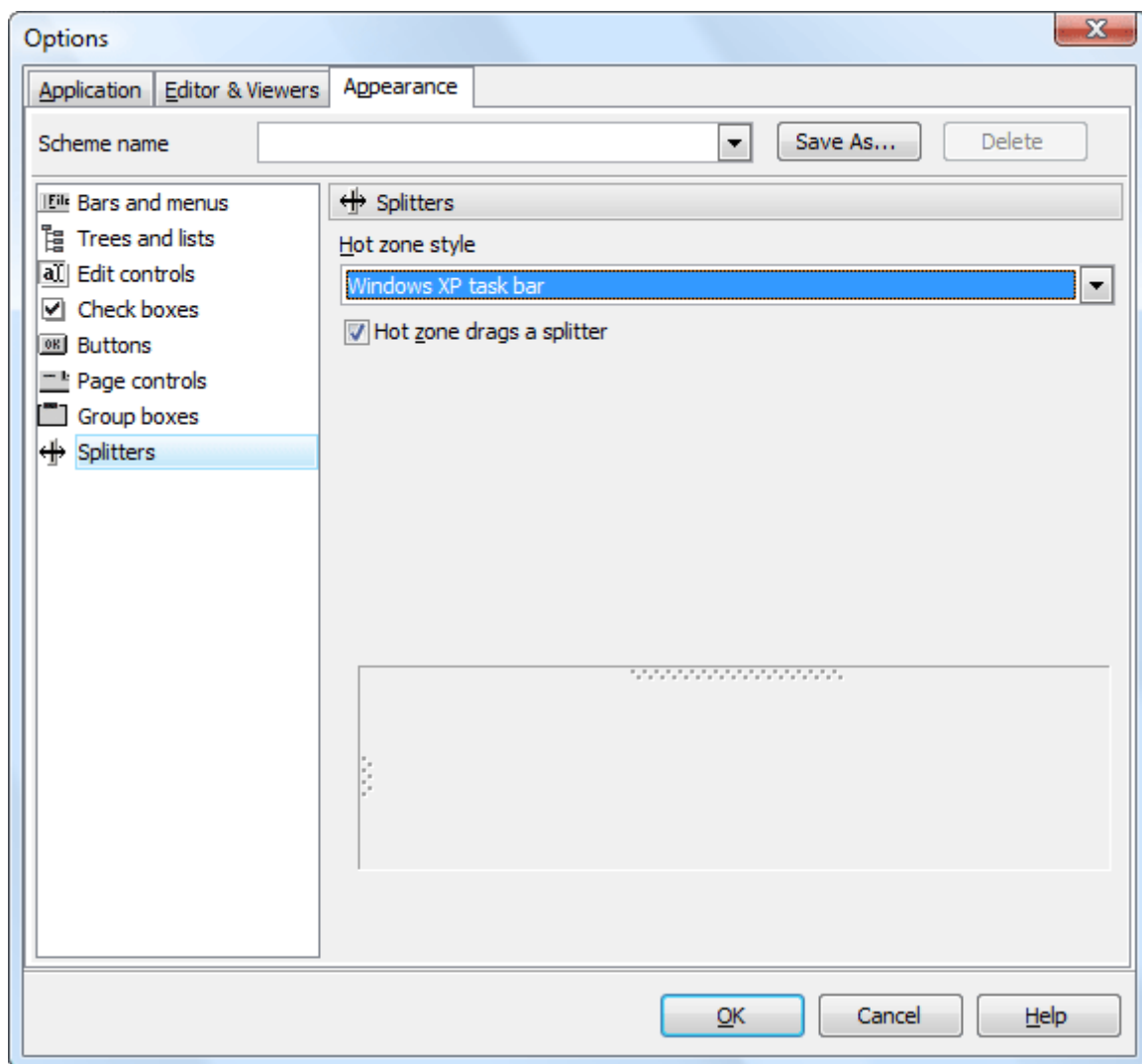
### 10.3.7 Group boxes

Use the [Group boxes](#) item to customize all DB2 Maestro group boxes according to your preferences. Use tab to apply styles for group boxes, enable/disable native style and define samples.



### 10.3.8 Splitters

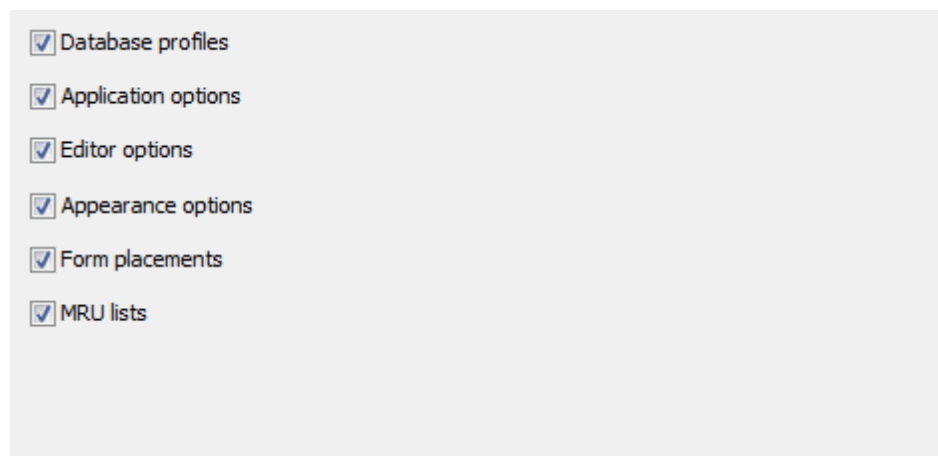
Use the [Splitters](#) item to customize all DB2 Maestro splitters according to your preferences. Use the tab to select hot zone style (*Windows XP task bar*, *Media Player 8*, *Media Player 9*, *Simple* or *none*) and specify the [Hot zone drags a splitter](#) option.



## 10.4 Export Settings

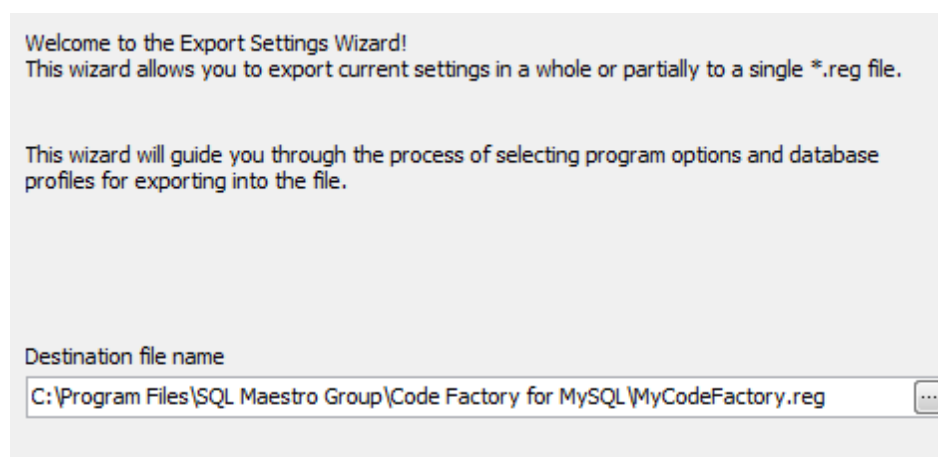
Export Settings Wizard allows you to export all or partial DB2 Maestro settings to single \*.reg file which can be applied to the application of DB2 Maestro installed on another machine or used to backup previous settings. To run the wizard, select the Tools | Options main menu item and click Export Settings in the [Options](#)<sup>[292]</sup> dialog.

- [Specifying destination file to save settings to](#)<sup>[332]</sup>
- [Specifying settings categories to save](#)<sup>[332]</sup>
- [Select database profiles to save](#)<sup>[333]</sup>
- [Saving settings](#)<sup>[333]</sup>



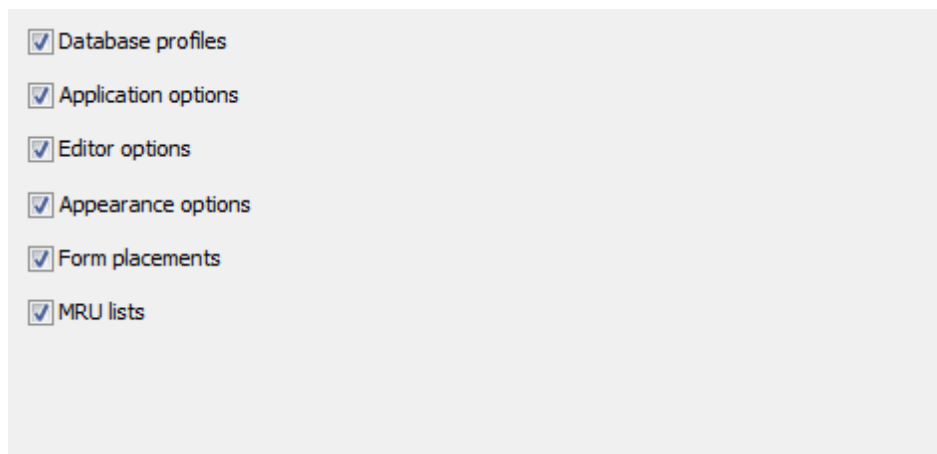
### 10.4.1 Specifying destination file

Specify a \*.reg file to extract DB2 Maestro setting to.



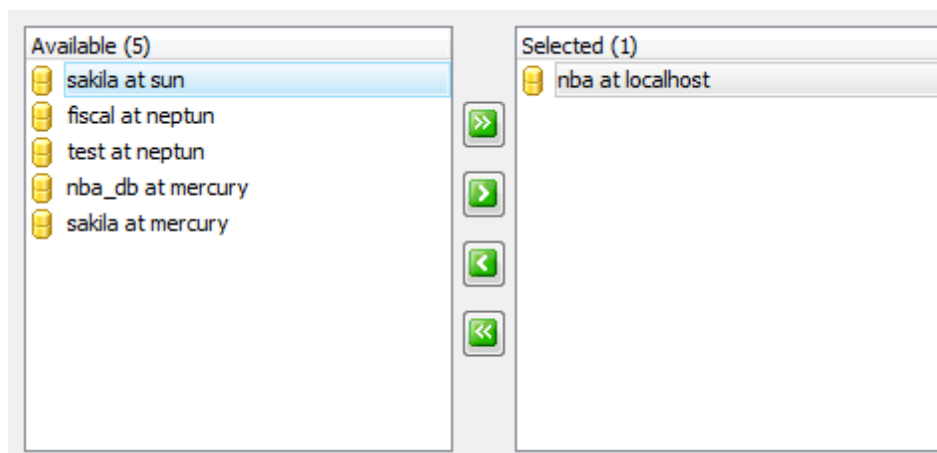
### 10.4.2 Selecting setting categories

The options of this step specify the information saved to the result file, e.g. Database profiles, [Application options](#)<sup>[293]</sup>, etc.



### 10.4.3 Selecting database profiles

Select database profiles to save their settings by moving them from the [Available Databases](#) list to the [Selected Databases](#) one.



### 10.4.4 Saving settings

Click the [Ready](#) button to start the extracting. The process log is displayed in the [Export log](#) box.

## Export log

The command(s) completed successfully.  
Exporting editor options...  
The command(s) completed successfully.  
Exporting appearance options...  
The command(s) completed successfully.  
Exporting form placements...  
The command(s) completed successfully.  
Exporting MRU lists...  
The command(s) completed successfully.



Click "Ready" to export settings.

# Index

## - A -

- Alias Editor 131
  - Editing alias properties 132
  - Managing alias data 133
- Aliases 128
  - Alias Editor 131
  - Create Alias Wizard 129
- Appearance Options
  - Bar and menus 323
  - Buttons 327
  - Check boxes 326
  - Edit controls 325
  - Group boxes 329
  - Page controls 328
  - Splitters 330
  - Trees and lists 324

## - B -

- BLOB Editor 223
  - Hexadecimal mode 224
  - HTML mode 225
  - Image mode 223
  - Navigating withing the BLOB editor 223
  - PDF mode 226
  - Plain text mode 224
- BLOB Viewer 252
  - Hexadecimal mode 252
  - HTML mode 255
  - Image mode 254
  - PDF mode 256
  - Plain text mode 253

## - C -

- Checks 81
- Create Alias Wizard 129
  - Specifying alias options 130
- Create Database Profiles Wizard 25
  - Setting connection properties 25
  - Setting profile options 25
- Create Domain Wizard 124

- Specifying domain properties 124
- Create External Procedure Wizard 110
- Create Function Wizard 116
- Create Group Wizard 180
  - Specifying group properties 181
- Create MQ Table Wizard 158
  - Selecting fields 158
  - Specifying MQ table properties 159
- Create object wizard - basic principles 36
  - Setting object name 37
  - Viewing common information 38
- Create Procedure Wizard 102
  - Managing parameters 105
  - Specifying procedure definition 105
  - Specifying procedure options 103
- Create Schema Wizard 60
- Create Sequence Wizard 137
- Create Structured type Wizard 142
  - Specifying structured type properties 143
  - Specifying structured type subitems 145
- Create Table Wizard 66
- Create Tablespace Wizard 188
- Create Trigger Wizard 84
- Create updatable views 283
- Create User Wizard 173
  - Specifying user properties 174
- Create View Wizard 92
- CRUD Procedures Generation 281

## - D -

- Data Analysis 262
  - Data 264
  - Query 263
- Data Management 211
  - BLOB editor 223
  - Data View 212
  - Export Data Wizard 228
  - Get SQL Dump Wizard 235
  - Import Data Wizard 238
- Data View
  - Editing data in dialog 219
  - Lookup editor 219
  - Viewing data 212
  - Working with data grid 213
  - Working with info cards 218
- Database Editor 34
- Database Explorer

- Database Explorer
  - Filtering explorer content 53
- Database Management 23
  - Creating a database profile 25
  - Database editor 34
  - Editing a database profile 27
- Database Object
  - External Functions 109
  - External Procedures 109
  - Functions 115
  - Procedures 101
  - Schemas 59
  - Tables 65
  - Views 91
- Database Objects 58
  - Changing object properties 44
  - Copy, Paste and Drag-n-Drop features 48
  - Create object wizard 36
  - Creating a database object 36
  - Describing object 44
  - Duplicate Object wizard 45
  - Duplicating a database object 45
  - Duplicating selected object 47
  - Managing database objects 55
  - Object Editor 39
  - Queries 194
  - Viewing database objects 53
- Database Profile Editor 27
  - Connection properties 27
  - Database options 28
  - Default directories 30
  - Editing obligatory scripts to execute 30
  - Setting log options 32
- DB2 Maestro 1
  - Getting started 12
  - Installation 4
  - License agreement 6
  - Registration 5
  - System requirements 3
- Diagram Viewer 258
  - Exporting diagram image 260
  - Selecting fields 259
- DML Procedures Generation 281
- Domain Editor 126
  - Editing domain properties 127
- Domains 123
  - Create Domain Wizard 124
  - Domain Editor 126
- Duplicate Object Wizard 45
  - Modifying new object's definition 47
  - Selecting object to duplicate 46
  - Selecting source and destination databases 45
- Duplicating a database object 45
  - Duplicate Object Wizard 45
  - Duplicating selected objects 47
- E -**
- Editor & Viewer Options
  - Code Folding 321
  - Code Insight 320
  - Display 316
  - General 315
  - PHP highlight 319
  - SQL highlight 317
  - XML highlight 318
- EULA 6
- Export Data Wizard 228
  - Adjusting data formats 231
  - Selecting fields for export 230
  - Setting common export options 234
  - Setting destination file name 228
  - Setting format-specific options 231
  - Setting header and footer 229
- Export Settings Wizard
  - Saving settings 333
  - Selecting database profiles 333
  - Selecting setting categories 332
  - Specifying destination file 332
- External Functions 109
- External Procedure Editor 112
- External Procedures 109
  - Create External Procedure Wizard 110
  - External Procedure Editor 112
- F -**
- Field Editor 72
  - Setting field identity 75
- Filter Panel 53
- Find Text Dialog 288
- Foreign Key References 88
- Foreign Keys 78
- Function Editor 119
  - Editing function properties 119

Function Editor 119  
     Viewing function result 121  
 Functions 115  
     Create Function Wizard 116  
     Function Editor 119

## - G -

Generate Database Report Wizard 250  
     Editing report style 251  
     Selecting reporting elements 250  
     Setting report paths 250  
 Get SQL Dump Wizard 235  
     Selecting fields 235  
     Specifying dump options 236  
 Getting Started 12  
     Explaining user interface 16  
     First time started 17  
     Switching between windows 20  
     Working with databases 13  
 Group Editor 183  
     Editing group properties 184  
 Groups 180  
     Create Group Wizard 180  
     Group Editor 183

## - I -

Import Data Wizard 238  
     Customizing common options 244  
     Data Format 244  
     Map builder 242  
     Setting fields correspondence 241  
     Setting source file name 239  
 Indexes 76  
 Installation instructions 4

## - L -

License Agreement 6

## - M -

MQ Table Editor 164  
     Editing MQ table properties 165  
     Viewing MQ table data 167  
 MQ Tables

Create MQ Table Wizard 158  
 MQ Table Editor 164

## - O -

Object Browser 53  
 Object editor - basic principles 39  
     Function executing 43  
     Object dependencies 41  
     Object grants 40  
     Parameter Editor 43  
     Permissions of an object 40  
     Results tab 43  
     SQL definition 42  
 Object Management 35  
     Changing object properties 44  
     Copy, Paste and Drag-n-Drop features 48  
     Creating a database object 36  
     Describing object 44  
     Duplicating a database object 45  
     Managing aliases 128  
     Managing domains 123  
     Managing external functions 109  
     Managing external procedures 109  
     Managing functions 115  
     Managing groups 180  
     Managing objects 55  
     Managing procedures 101  
     Managing Queries 194  
     Managing sequences 136  
     Managing structured types 141  
     Managing tables 65  
     Managing tablespaces 187  
     Managing users 173  
     Managing views 91  
     Schemas 59  
     Viewing objects 53  
     Wizards and Editors 39  
 Object Manager 55  
 Options 292  
     Appearance 323  
     Application 293  
     Application confirmations 294  
     Application preferences 293  
     BLOB Viewer 303  
     Data Grid 308  
     Designer 305  
     Editor & Viewers 315

- Options 292
  - Editors 306
  - Explorer 298
  - Export 304
  - Export Settings 332
  - Object Manager 299
  - Query Builder 301
  - SQL Editor 299
  - SQL Script Editor 300
  - Tools 296

## - P -

- Procedure Editor 106
  - Editing properties 107
- Procedures 101
  - Create Procedure Wizard 102
  - Procedure Editor 106
- Purchase DB2 Maestro 5

## - Q -

- Queries 194
  - Query Parameters 201
  - SQL Editor 196
  - Visual Query Builder 202
- Query Parameters 201

## - R -

- Registration 5
- Replace Text Dialog 289
- Report Designer 267
  - Object Inspector 271
  - Toolbox 269

## - S -

- Schema Designer 274
  - Navigation bar 276
  - Toolbox 276
- Schema Editor 62
- Schema Objects
  - Aliases 128
  - Domains 123
  - Sequences 136
- Schemas 59

- Create Schema Wizard 60
  - Schema editor 62
- Sequence Editor 138
- Sequences 136
  - Create Sequence Wizard 137
  - Sequence Editor 138
- Server Editor 171
- Server Management 171
- Server Objects 169
  - Changing object properties 44
  - Copy, Paste and Drag-n-Drop features 48
  - Create object wizard 36
  - Describing object 44
  - Groups 180
  - Object Editor 39
  - Structured types 141
  - Tablespaces 187
  - Users 173
- Split table 284
- SQL Editor 196
  - Executing query 199
- SQL Script Editor 248
- Structured Type Editor 146
  - Editing structured type properties 147
- Structured Types 141
  - Create Structured type Wizard 142
  - Extended Methods 154
  - Methods 151
  - Structured Type Editor 146
  - UDS Attributes 150
- System requirements 3

## - T -

- Tabbed MDI 16
- Table Editor 68
  - Editing table properties 68
  - Master-detail data view 70
  - Viewing table data 70
- Tables 65
  - Checks 81
  - Create Table Wizard 66
  - Fields 72
  - Foreign key references 88
  - Foreign keys 78
  - Indexes 76
  - Table editor 68
  - Triggers 83

- Tablespace Editor 189
  - Editing tablespace properties 190
- Tablespaces 187
  - Create Tablespace Wizard 188
  - Tablespace Editor 189
- Tools 246
  - BLOB Viewer 252
  - Data Analysis 262
  - Database Designer 274
  - Dependency Tracker 279
  - Diagram Viewer 258
  - Dialogs 288
  - Generate Database Report Wizard 250
  - Report Designer 267
  - Schema Designer 274
  - Script Runner 247
  - SQL Generator 280
  - SQL Script Editor 248
- Trigger Editor 86
- Triggers 83
  - Create Trigger Wizard 84
  - Trigger Editor 86

## - U -

- User Editor 176
  - Editing user properties 177
  - Managing user objects 179
- User interface 16
- Users 173
  - Create User Wizard 173
  - User Editor 176

## - V -

- View Editor 97
  - Editing view properties 98
  - Viewing data 99
- Views 91
  - Create View Wizard 92
  - View Editor 97
- Visual Query Builder 202
  - Executing a query 209
  - Working with editor area 208

## - W -

- WHERE condition 92, 161
- Window List 20
- Wizards and Editors 39
  - Create object wizard - basic principles 36
  - Object Editor 39