



# Oracle Maestro

## **User's guide**

# Table of Contents

Foreword	0
<b>I Welcome to Oracle Maestro!</b>	<b>1</b>
1 System Requirements .....	3
2 Installation .....	4
3 How can I purchase Oracle 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 .....	15
First time started .....	16
Tabbed MDI .....	17
Switching between windows .....	18
4 Shortcut keys .....	20
<b>III Databases and Database Profiles</b>	<b>21</b>
1 Creating Database Profiles .....	23
Setting connection properties .....	23
Setting profile options .....	23
2 Editing Database Profile .....	26
Editing connection properties .....	26
Setting profile options .....	27
Setting default directories .....	29
Editing obligatory scripts to execute .....	30
Setting log options .....	30
Statistics .....	31
3 Database Editor .....	33
<b>IV Database Object Management</b>	<b>35</b>
1 Create Objects .....	36
Create Database Object Dialog .....	36
Overview of Create Objects Wizards .....	37
Setting object name .....	38
Viewing common information .....	38
2 Edit Objects .....	39
Overview of Object Editors .....	39
Permissions of the Object .....	40
Object grants .....	41
Object Dependencies .....	42

SQL Definition .....	43
Parameter Editor .....	44
Executing functions and procedures .....	44
<b>Modify Object Properties</b> .....	<b>45</b>
<b>Describe Objects</b> .....	<b>45</b>
<b>3 Duplicate Objects</b> .....	<b>46</b>
<b>Duplicate Object Wizard</b> .....	<b>46</b>
Selecting source and destination databases .....	46
Selecting object to duplicate .....	47
Modifying new object definition .....	48
<b>Duplicate Selected Object</b> .....	<b>48</b>
<b>Copy, Paste and Drag-n-Drop features</b> .....	<b>49</b>
<b>4 Browse Objects</b> .....	<b>51</b>
<b>Database Explorer</b> .....	<b>51</b>
Filtering explorer content .....	54
<b>Object Browser</b> .....	<b>54</b>
<b>Object Manager</b> .....	<b>56</b>
<b>Filter Builder Dialog</b> .....	<b>57</b>

## V Database Objects 59

<b>1 Schemas</b> .....	<b>60</b>
<b>Schema Editor</b> .....	<b>61</b>
<b>2 Tables</b> .....	<b>63</b>
<b>Create Table Wizard</b> .....	<b>64</b>
Storage option properties .....	67
<b>Table Editor</b> .....	<b>69</b>
Editing table properties .....	69
Managing table data .....	72
Gathering statistic properties .....	74
<b>Fields</b> .....	<b>75</b>
LOB properties .....	78
<b>Indexes</b> .....	<b>79</b>
<b>Foreign Keys</b> .....	<b>82</b>
<b>Checks</b> .....	<b>85</b>
<b>Triggers</b> .....	<b>87</b>
Create Trigger Wizard .....	88
Trigger Editor .....	90
<b>Foreign Key References</b> .....	<b>92</b>
<b>3 Views</b> .....	<b>95</b>
<b>Create View Wizard</b> .....	<b>96</b>
<b>View Editor</b> .....	<b>101</b>
Editing view properties .....	102
Viewing data .....	103
<b>4 Materialized Views</b> .....	<b>105</b>
<b>Create Materialized Views Wizard</b> .....	<b>106</b>
<b>Materialized Views Editor</b> .....	<b>110</b>
<b>5 Procedures</b> .....	<b>113</b>
<b>Create Procedure Wizard</b> .....	<b>114</b>
Specifying procedure options .....	115
Managing parameters .....	116
Specifying procedure definition .....	117

<b>Procedure Editor</b> .....	<b>118</b>
Editing properties .....	119
Viewing procedure results .....	120
<b>6 Functions</b> .....	<b>122</b>
<b>Create Function Wizard</b> .....	<b>123</b>
<b>Function Editor</b> .....	<b>125</b>
Editing properties .....	125
Viewing Function results .....	127
<b>7 Synonyms</b> .....	<b>129</b>
<b>Create Synonym Wizard</b> .....	<b>130</b>
<b>Synonym Editor</b> .....	<b>131</b>
Editing synonym properties .....	131
<b>8 Schema Triggers</b> .....	<b>133</b>
<b>Create Schema Triggers Wizard</b> .....	<b>134</b>
Specifying schema trigger properties .....	135
Specifying trigger definition .....	138
<b>Schema Triggers Editor</b> .....	<b>138</b>
Editing trigger properties .....	139
<b>9 Database links</b> .....	<b>142</b>
<b>Create Database Link Wizard</b> .....	<b>143</b>
Specifying link properties .....	144
<b>Database Link Editor</b> .....	<b>145</b>
Editing link properties .....	146
<b>10 Packages</b> .....	<b>147</b>
<b>Create Package Wizard</b> .....	<b>148</b>
Specifying package properties .....	149
Adding methods .....	150
Create Package Method Wizard .....	151
Specifying package definition .....	152
<b>Package Editor</b> .....	<b>153</b>
<b>11 Package bodies</b> .....	<b>156</b>
<b>Create Package Body Wizard</b> .....	<b>157</b>
<b>Package Body Editor</b> .....	<b>158</b>
<b>12 Sequences</b> .....	<b>160</b>
<b>Create Sequence Wizard</b> .....	<b>161</b>
<b>Sequence Editor</b> .....	<b>162</b>
<b>13 Types</b> .....	<b>165</b>
<b>Create Type Wizard</b> .....	<b>166</b>
Specifying type properties .....	166
Setting type subitems .....	167
Specifying type definition .....	168
<b>Type Editor</b> .....	<b>169</b>
Editing type options .....	170
<b>Type bodies</b> .....	<b>171</b>
Create Type Body Wizard .....	172
Specifying body definition .....	173
Type Body Editor .....	174
Editing type body properties .....	175
<b>14 Users</b> .....	<b>178</b>
<b>Create User Wizard</b> .....	<b>178</b>
<b>User Editor</b> .....	<b>179</b>

Editing user properties.....	179
<b>15 User profiles .....</b>	<b>182</b>
<b>Create User Profile Wizard .....</b>	<b>183</b>
Specifying profile properties.....	184
<b>User Profile Editor .....</b>	<b>186</b>
Editing profile properties.....	186
Managing users.....	188
<b>16 Roles .....</b>	<b>190</b>
<b>Create Role Wizard .....</b>	<b>191</b>
<b>Role Editor .....</b>	<b>192</b>
<b>17 Collections .....</b>	<b>193</b>
<b>Create Collection Wizard .....</b>	<b>194</b>
Setting collection properties.....	195
<b>Collection Editor .....</b>	<b>196</b>
Editing collection properties.....	196
<b>18 Java sources .....</b>	<b>198</b>
<b>Create Java Source Wizard .....</b>	<b>199</b>
Specifying java source properties.....	200
Specifying java source definition.....	200
<b>Java Source Editor .....</b>	<b>201</b>
Editing java source properties.....	202
<b>19 Jobs .....</b>	<b>204</b>
<b>Create Job Wizard .....</b>	<b>205</b>
Specifying job properties.....	205
Specifying job definition.....	206
<b>Job Editor .....</b>	<b>207</b>
Editing job properties.....	208
<b>20 Queues .....</b>	<b>210</b>
<b>Create Queue Wizard .....</b>	<b>211</b>
Specifying queue properties.....	211
<b>Queue Editor .....</b>	<b>212</b>
Editing queue properties.....	213
<b>21 Queue tables .....</b>	<b>215</b>
<b>Create Queue Table Wizard .....</b>	<b>216</b>
Specifying queue table properties.....	217
<b>Queue Table Editor .....</b>	<b>218</b>
Editing queue table properties.....	218
<b>22 Clusters .....</b>	<b>221</b>
<b>Create Cluster Wizard .....</b>	<b>222</b>
Specifying cluster properties.....	223
Adding cluster subitems.....	224
<b>Cluster Editor .....</b>	<b>225</b>
Editing cluster properties.....	226
<b>23 Libraries .....</b>	<b>228</b>
<b>Create Library Wizard .....</b>	<b>229</b>
Specifying library properties.....	229
<b>Library Editor .....</b>	<b>230</b>
Editing library properties.....	231
<b>24 Directories .....</b>	<b>233</b>
<b>Create Directory Wizard .....</b>	<b>234</b>

Specifying directory properties .....	235
<b>Directory Editor</b> .....	<b>236</b>
Editing directory properties .....	237
<b>25 Database triggers</b> .....	<b>239</b>
<b>Create Database Trigger Wizard</b> .....	<b>240</b>
<b>Database Trigger Editor</b> .....	<b>242</b>
<b>26 Public Synonyms</b> .....	<b>244</b>
<b>Create Public Synonym Wizard</b> .....	<b>245</b>
Specifying synonym properties .....	246
<b>Public Synonym Editor</b> .....	<b>247</b>
Editing synonym properties .....	248
<b>27 Public database links</b> .....	<b>250</b>
<b>Create Public Database Link Wizard</b> .....	<b>251</b>
Specifying link properties .....	252
<b>Public Database Link Editor</b> .....	<b>253</b>
Editing link properties .....	253
<b>28 Tablespaces</b> .....	<b>255</b>
<b>Create Tablespace Wizard</b> .....	<b>256</b>
Specifying tablespace properties .....	257
Adding tablespace files .....	258
<b>Tablespace Editor</b> .....	<b>259</b>
Editing tablespace properties .....	260

## VI Queries

**262**

<b>1 SQL Editor</b> .....	<b>264</b>
SQL Formatter .....	265
Executing query .....	266
Query Parameters .....	268
<b>2 Visual Query Builder</b> .....	<b>269</b>
Creating query diagram .....	270
Working with editor area .....	275
Executing query .....	276

## VII Data Management

**278**

<b>1 Data View</b> .....	<b>279</b>
Working with data grid .....	280
Working with info cards .....	285
Data input form .....	286
Data filtering .....	287
<b>2 BLOB Editor</b> .....	<b>290</b>
Editing as image .....	290
Editing as hexadecimal dump .....	291
Editing as plain text .....	291
Editing as HTML .....	292
Editing as PDF document .....	293
<b>3 Export Data Wizard</b> .....	<b>295</b>
Setting destination file name and format .....	295
Setting header and footer .....	296
Selecting fields for export .....	297
Adjusting data formats .....	297

Setting format-specific options .....	298
Setting common export options .....	301
4 Get SQL Dump .....	302
Selecting fields .....	302
Specifying dump options .....	303
5 Import Data Wizard .....	305
Setting source file name and format .....	306
Setting the accordance between source and target columns .....	308
Map builder .....	309
Data formats .....	310
Customizing common options .....	311
<b>VIII Database Tools</b>	<b>313</b>
1 Script Runner .....	315
2 SQL Script Editor .....	316
3 Generate Database Report .....	318
Selecting reporting elements and setting other report options .....	318
Reporting objects and editing styles options .....	318
Editing database report style .....	319
4 Database Export .....	320
5 BLOB Viewer .....	321
Viewing as hexadecimal dump .....	321
Viewing as plain text .....	322
Viewing as image .....	323
Viewing as HTML .....	324
Viewing as PDF .....	325
6 Diagram Viewer .....	327
Customizing diagram properties .....	328
Exporting diagram image .....	329
7 Data Analysis .....	331
Input SELECT query .....	332
Managing report data .....	333
8 Report Designer .....	336
Designer Tools and Objects .....	338
Object Inspector .....	340
9 Schema Designer .....	343
Designer Navigation Bar .....	345
Schema Designer Toolbox .....	345
10 PL/SQL Debugger .....	348
Debugging process .....	349
Debug information .....	351
11 Session Browser .....	353
12 Dependency tracker .....	354
13 SQL Generator .....	355
14 DML procedures generation .....	356
15 Generation of updatable views .....	358
16 Split table .....	359

<b>17 Nullable Column Checker .....</b>	<b>361</b>
<b>18 Dialogs .....</b>	<b>363</b>
Find Text dialog .....	363
Replace Text dialog .....	364

## **IX Options 367**

<b>1 Application .....</b>	<b>368</b>
<b>Preferences .....</b>	<b>368</b>
<b>Confirmations .....</b>	<b>369</b>
<b>Directories .....</b>	<b>371</b>
<b>Tools .....</b>	<b>371</b>
Explorer .....	373
Object Manager .....	374
SQL Editor .....	374
SQL Script Editor .....	375
Query Builder .....	376
Colors .....	378
BLOB Viewer .....	378
Data Export .....	379
Database Designer .....	380
<b>Object Editors .....</b>	<b>381</b>
Table .....	383
<b>Data Grid .....</b>	<b>383</b>
Options .....	384
Colors .....	386
Formats .....	386
Filter .....	388
<b>2 Editors &amp; Viewers .....</b>	<b>390</b>
<b>General .....</b>	<b>390</b>
<b>Display .....</b>	<b>391</b>
<b>SQL highlight .....</b>	<b>392</b>
<b>XML highlight .....</b>	<b>393</b>
<b>PHP highlight .....</b>	<b>394</b>
<b>Code Insight .....</b>	<b>395</b>
<b>Code Folding .....</b>	<b>396</b>
<b>3 Appearance .....</b>	<b>398</b>
<b>Bars and menus .....</b>	<b>398</b>
<b>Trees and lists .....</b>	<b>399</b>
<b>Edit controls .....</b>	<b>400</b>
<b>Check boxes .....</b>	<b>401</b>
<b>Buttons .....</b>	<b>402</b>
<b>Page controls .....</b>	<b>403</b>
<b>Group boxes .....</b>	<b>404</b>
<b>Splitters .....</b>	<b>405</b>
<b>4 Export Settings .....</b>	<b>407</b>
Specifying destination file .....	407
Selecting setting categories .....	407
Selecting database profiles .....	408
Saving settings .....	408

**Index**

**410**

# 1 Welcome to Oracle Maestro!

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

## **Basic Oracle Maestro features**

### **Support of the latest Oracle features**

Use Oracle Maestro to work with Oracle 8.x - 12.x. Among other features and objects implemented in the latest versions of the server, our software supports functions, procedures, sequences, packages, types, queues, 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>[21]</sup> for details.

### **Powerful database object management**

Oracle 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](#)<sup>[56]</sup>, sort, group and filter the database objects within [Object Browser](#)<sup>[54]</sup>, 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>[59]</sup> section.

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

Oracle 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>[63]</sup> section to learn more.

### **Building and executing queries**

Oracle 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>[262]</sup> section.

### **Powerful data management tools**

Oracle 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>[278]</sup> to learn the details.

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

Oracle 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, PL/SQL Debugger, 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>[313]</sup> section.

### **Security management**

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

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

In Oracle 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>[367]</sup> dialog description.

## 1.1 System Requirements

### Client environment

- Pentium PC or higher;
- Windows NT4/2000/XP/Vista/Windows 7/Windows 8/Windows 10/Windows 11;
- 512 MB RAM (1 GB recommended);
- 25 MB of free hard disk space;
- SVGA-compatible video adapter;
- Oracle client software.

### Server environment

- Oracle 8.x - 12.x.

## 1.2 Installation

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

- download the Oracle 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 Oracle Maestro shortcut in the corresponding program group of the Windows Start menu after the installation is completed.

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

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

You can also use the full distribution package to upgrade your current version of Oracle 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 Oracle Maestro?

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

You can select licensing options and register Oracle 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.

Oracle 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 export data from Oracle tables and queries to most popular formats, import data into the tables, generate SQL dump of selected tables, and export/import BLOB fields from/to files.



**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.



**Database Converter** is a user friendly tool to migrate any local or remote ADO-compatible database to Oracle.

Such tools transfer database schema and data and are equipped with native support for the most popular database servers.



**Data Sync** is a powerful and easy-to-use tool for database contents comparison and synchronization.

Such tools can be useful for database administrators, developers and testers that need a quick, easy and reliable way to compare and synchronize their data.



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 Oracle Maestro news at <http://www.sqlmaestro.com/products/oracle/maestro/news/>

## 2 Getting Started

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

### How to get started:

- [Connect to a database with Oracle Maestro](#)<sup>[13]</sup>
- [Explaining user interface](#)<sup>[15]</sup>
- [How Oracle Maestro looks when you start it for the first time](#)<sup>[16]</sup>
- [Shortcut keys](#)<sup>[20]</sup>

### Learning more:

- ❑ Study the [Overview of Database Object Management](#)<sup>[35]</sup> section for detailed instructions on using Oracle Maestro.
- ❑ See [Database Tools](#)<sup>[313]</sup> and [Queries](#)<sup>[262]</sup> sections for instructions on more advanced procedures!
- ❑ Find out more about [Working with Data in Oracle Maestro](#)<sup>[278]</sup>.
- ❑ Customize the way Oracle Maestro works, see [Program Options](#)<sup>[367]</sup> for full details.

## 2.1 Connect to a database

To manage an existing database with Oracle Maestro, you have to [create the according database profile](#)<sup>[23]</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>[26]</sup>).

When the profile is created you can connect to the database. To do so, select the database in the [Explorer tree](#)<sup>[51]</sup>, 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

To connect to an Oracle database with Oracle Maestro, specify the following connection options:

### User name

Use the field to specify the username to be used to connect to Oracle.

### Password

Enter the password for the user account on server.

### Database name

An entry from [TNSNames.ora](https://www.oracle.com/technetwork/database/enterprise-articles/tnsnamesora-111623.html).

### Connect mode

Allows you to connect with required administrative privileges (SYSDBA or SYSOPER). [More information](#).

Check [Use Operating system authentication](#) to allow Oracle to pass control of user authentication to the operating system. The technology works as follows:

- First, create an OS user (if it doesn't exist).
- Check a value of the Oracle `OS_AUTHENT_PREFIX` initialization parameter. Current value of this parameter can be retrieved using the following query:

```
SELECT VALUE FROM V$PARAMETER  
WHERE NAME = 'os_authent_prefix'
```

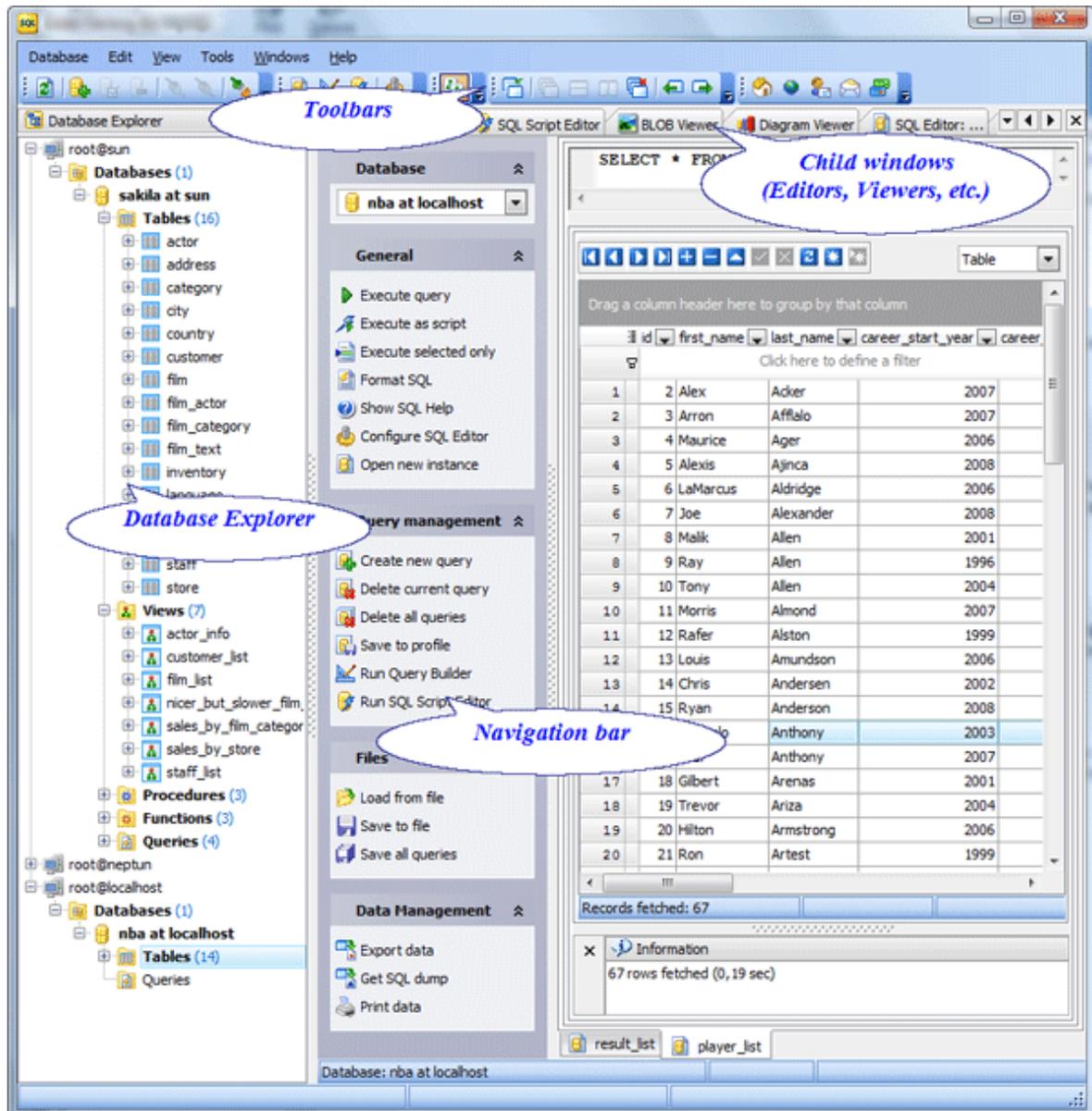
The default value is OPS\$. The initialization parameter can be modified with the ALTER SYSTEM command.

- Create a database user. The user must use the external identification and its name must be the prefix value concatenated to the OS username (on Windows platforms you would expect an Oracle username of "OPS\$DOMAIN\MY\_USER" for the user "my\_user").

## 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 Oracle Maestro's user interface. For detailed descriptions, see below.



### Database Explorer

The [Database Explorer](#)<sup>[51]</sup> occupies the left side of Oracle Maestro main window. It represents all the connected databases objects [including system objects](#)<sup>[27]</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](#)<sup>[54]</sup>

## Editors and Viewers

According to the MDI style implementation the Oracle 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>[18]</sup>, [Tabbed MDI](#)<sup>[17]</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>[20]</sup>

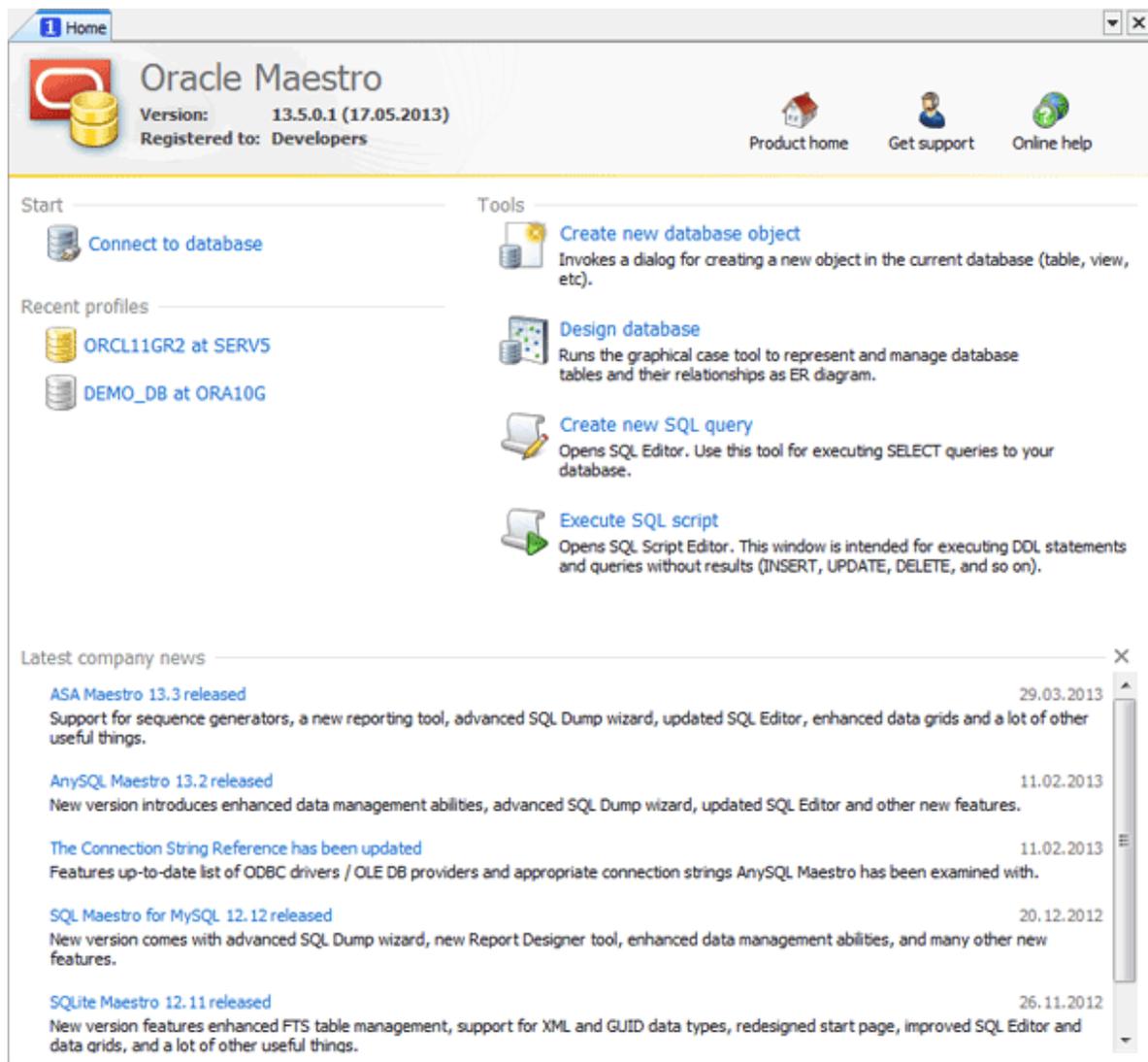
## Toolbars

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 Oracle 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>[23]</sup>

The window provides you with quick access to the [Create Database Object](#)<sup>[36]</sup> dialog, [Schema Designer](#)<sup>[34]</sup>, [SQL Editor](#)<sup>[26]</sup>, and [SQL Script Editor](#)<sup>[31]</sup> recently used database profiles. At the bottom of the page the latest company news and current discount programs are represented.

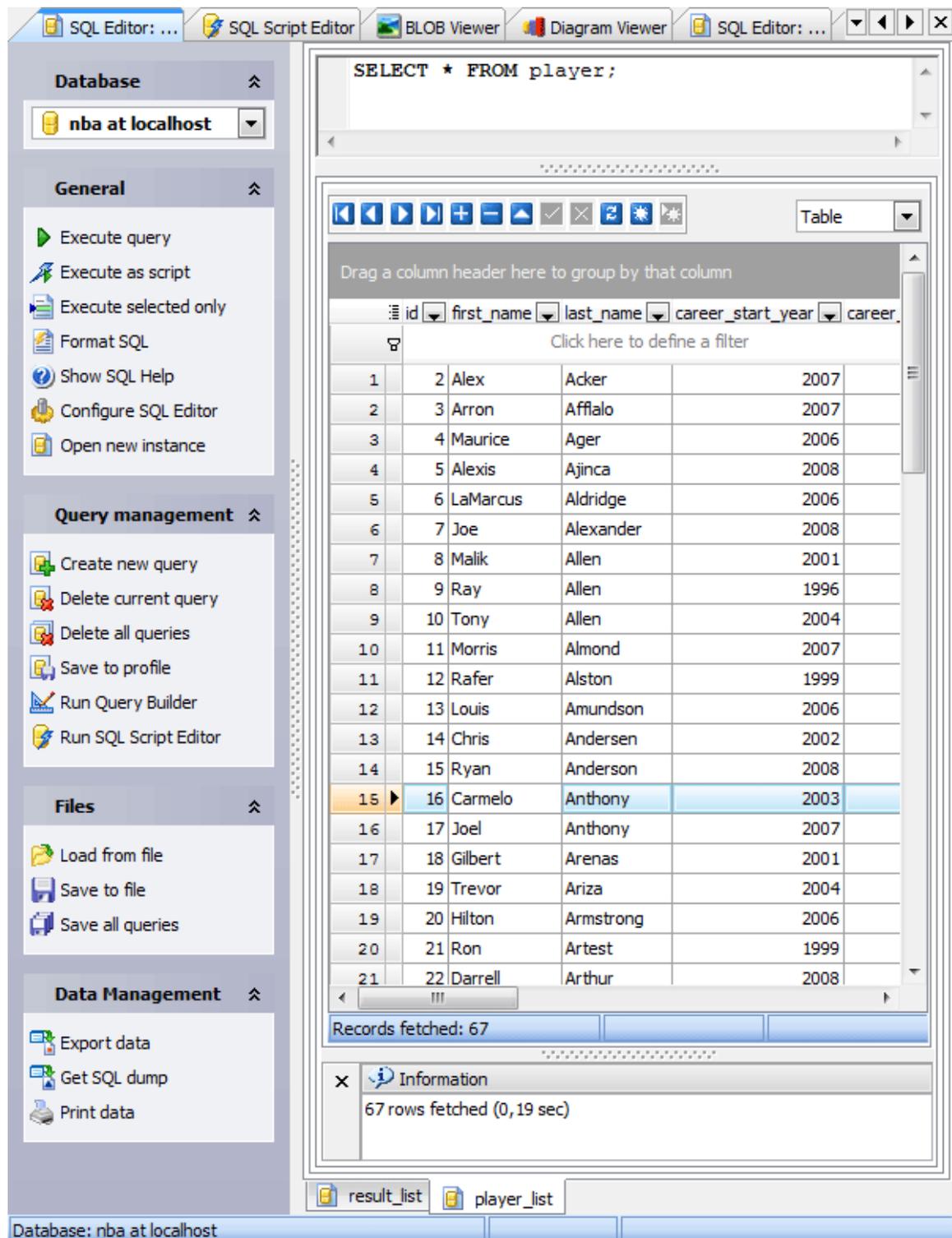


### 2.3.2 Tabbed MDI

Oracle 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.

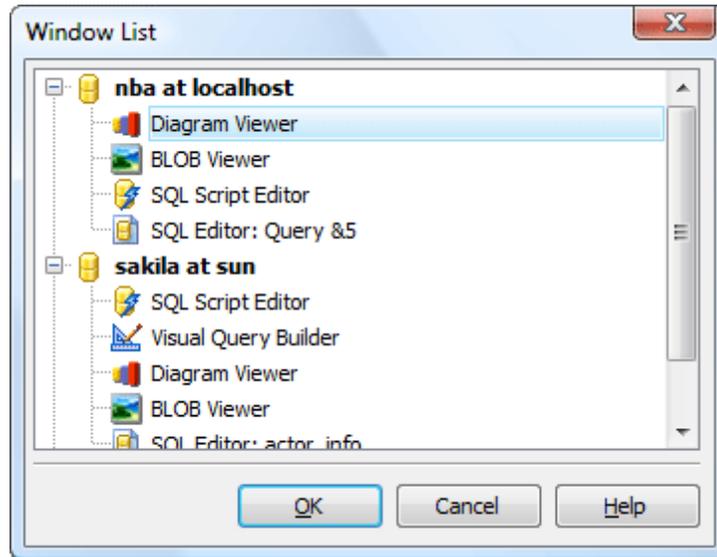


### 2.3.3 Switching between windows

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 Oracle Maestro.

<b>Interface</b> <sup>15</sup>	
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>
Oracle 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> <sup>264</sup>	
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> <sup>21</sup>	
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

Oracle 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 Oracle Maestro startup, login prompt before connection, etc.). To start working with databases in Oracle Maestro, you should create database profile(s) first.

Use the following links for details:

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

In Oracle Maestro database profiles are created within [Create Database Profiles Wizard](#)<sup>[23]</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>[23]</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>[23]</sup>
- [Specify database profile options](#)<sup>[23]</sup>

**See also:** [Edit Database Profile Dialog](#)<sup>[26]</sup>

### 3.1.1 Setting connection properties

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

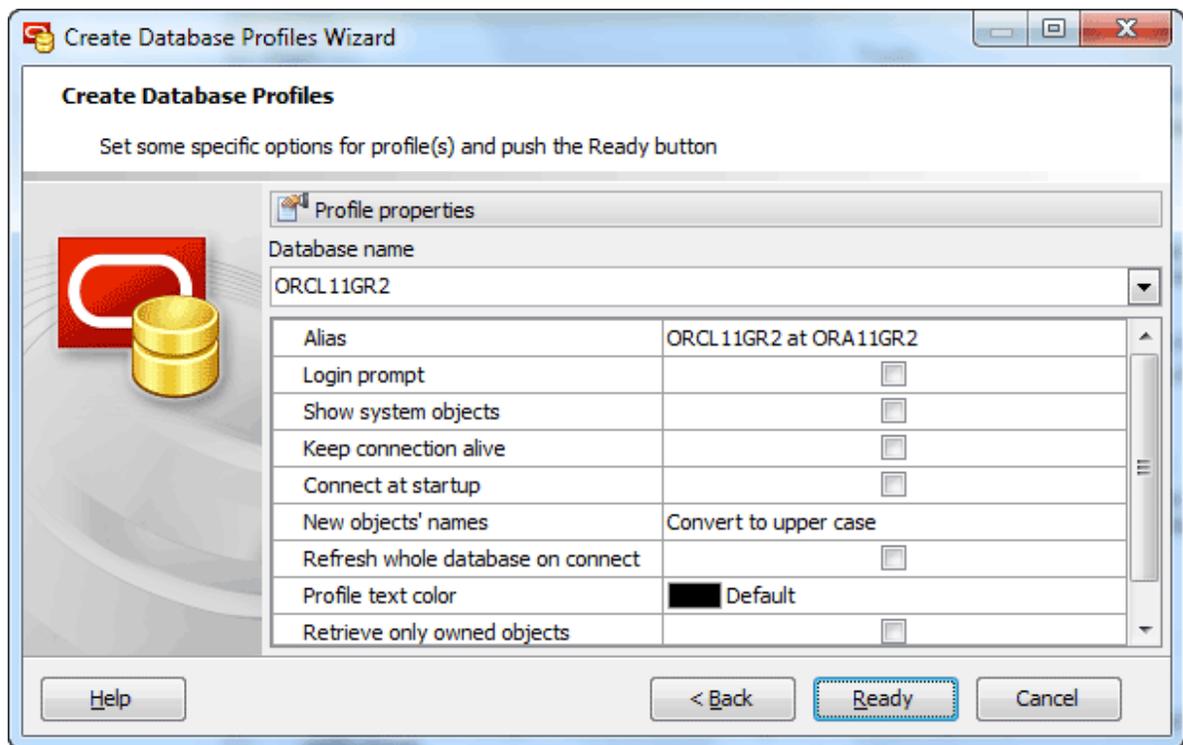


- Hide already registered databases

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

### 3.1.2 Setting profile options

To create a new profile, set the database name you want to connect.



[Login prompt before connection](#)

Use the option to enable Oracle 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.

[Retrieve only owned objects](#)

Use the option to decrease a quantity of the objects represented in [Explorer tree](#). If it's enabled the tree displays only the objects which owner is the current user.

[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.

Click the [Ready](#) button when done to start working with the selected databases in Oracle Maestro.

## 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.

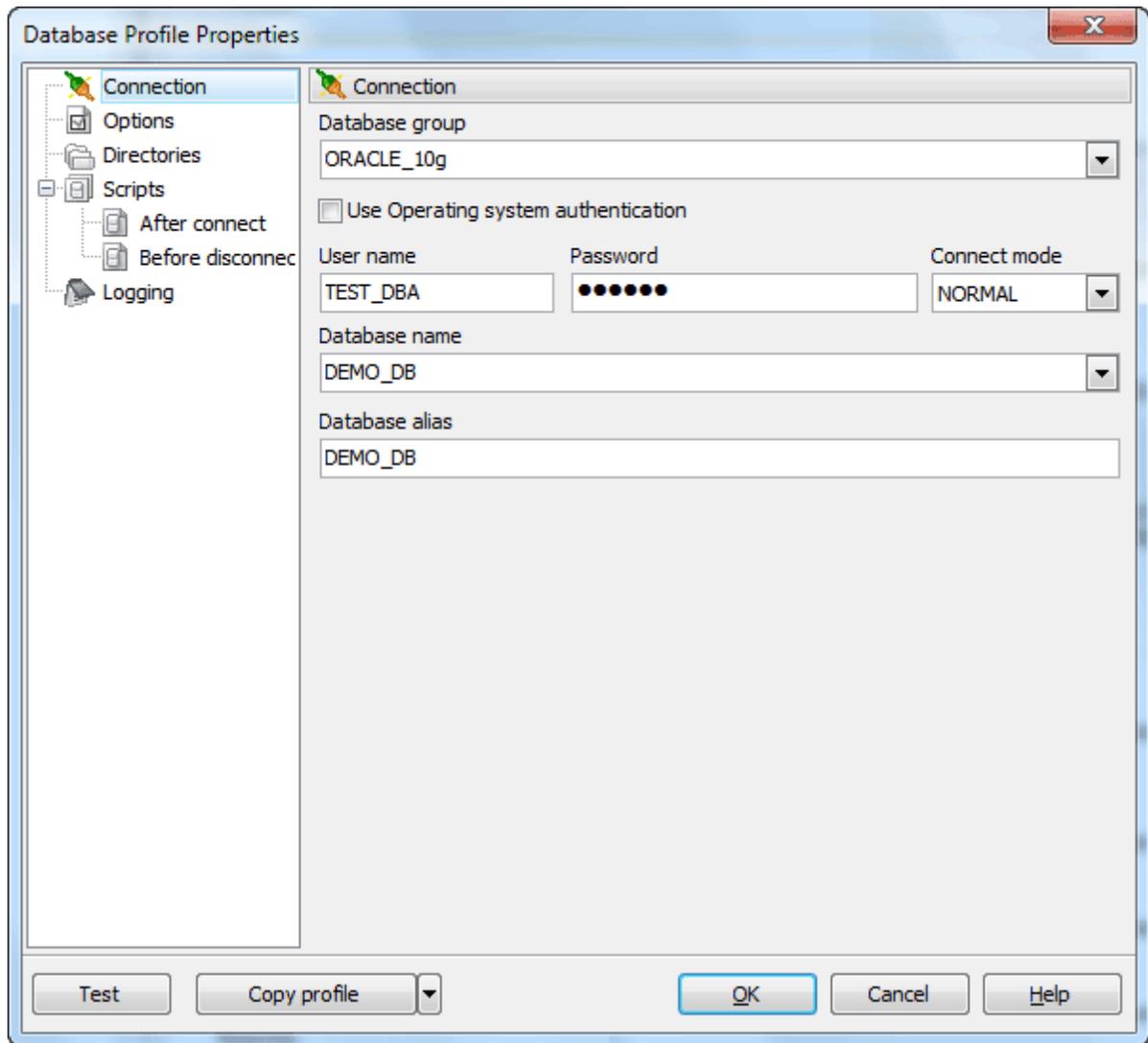
Instead of manual profile options editing you can copy all the options from the another existing profile with the [Copy profile](#) button.

- [Editing database connection properties](#) <sup>[26]</sup>
- [Settings database options](#) <sup>[27]</sup>
- [Setting default directories for database tools](#) <sup>[29]</sup>
- [Editing obligatory scripts to execute](#) <sup>[30]</sup>
- [Setting log options and file names](#) <sup>[30]</sup>

**See also:** [Create Database Profile Wizard](#) <sup>[23]</sup>

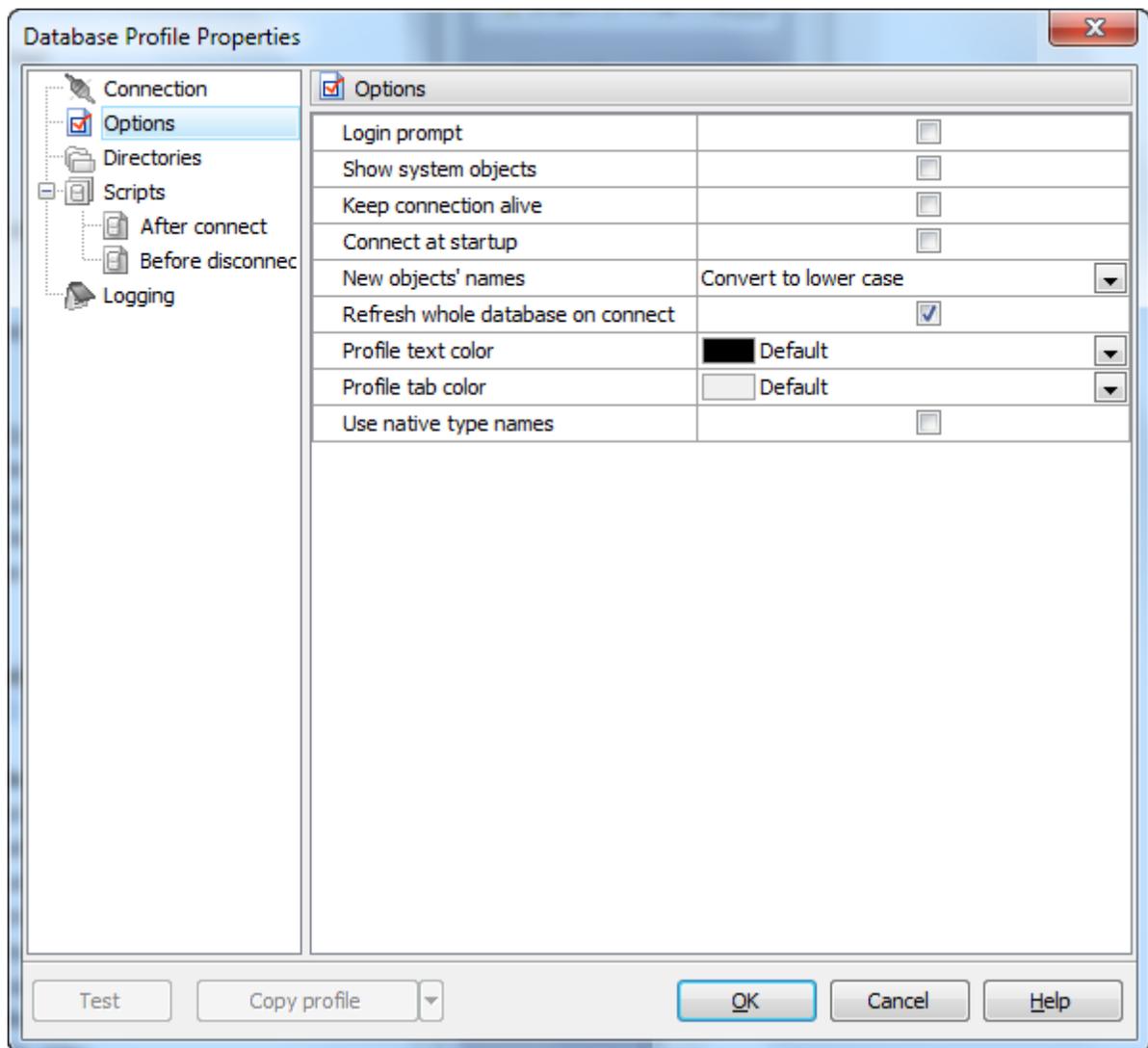
### 3.2.1 Editing connection properties

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 Oracle 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.

Retrieve only owned objects

Use the option to decrease a quantity of the objects represented in Explorer tree. If it's enabled the tree displays only the objects which owner is the current user.

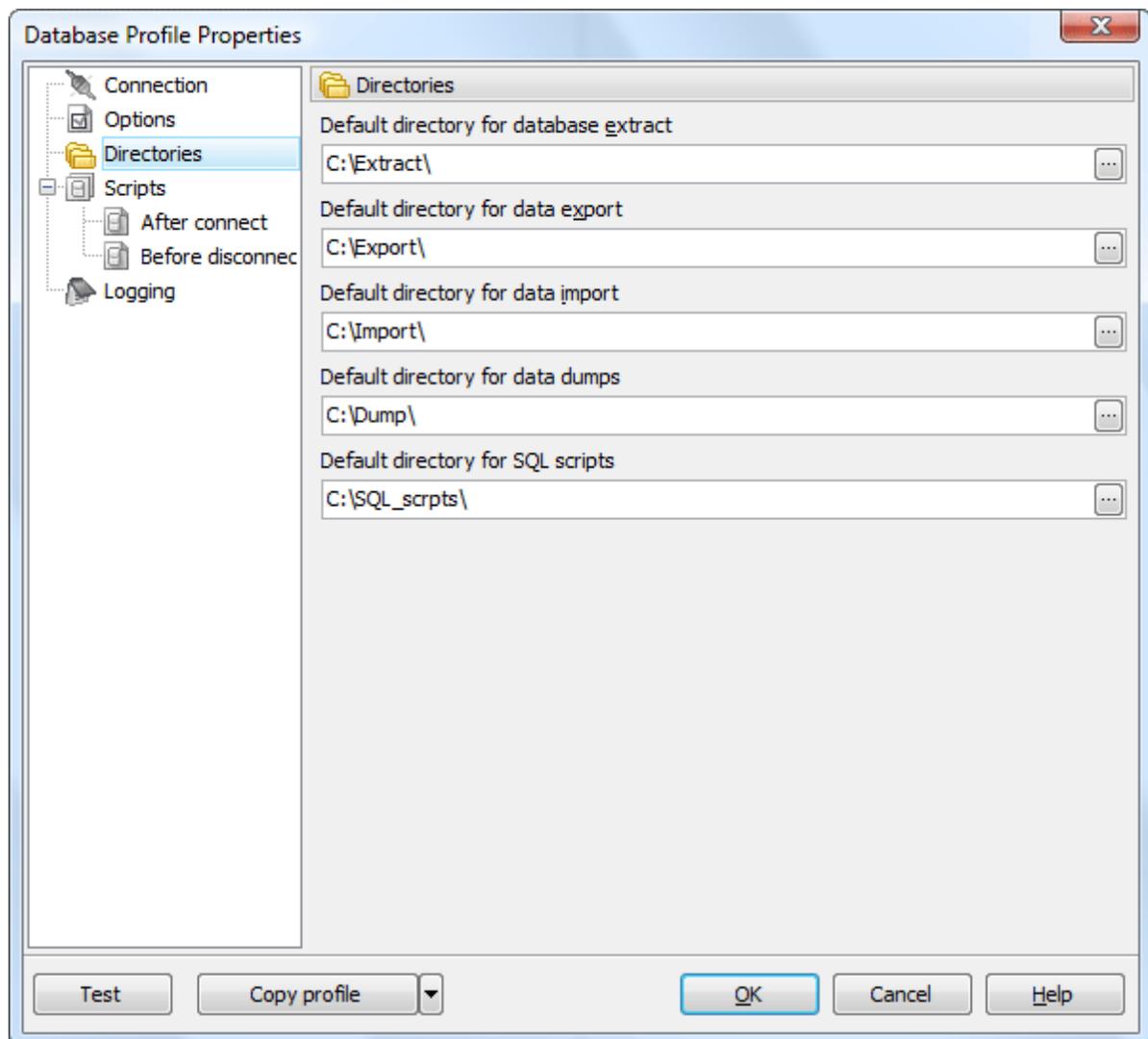
Refresh whole database on connect

Use the option along with the [Show empty schemas](#)<sup>373</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.

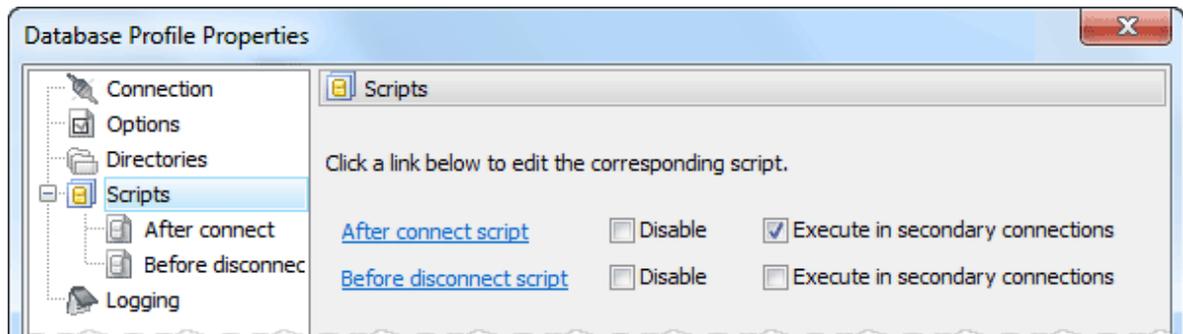
### 3.2.3 Setting default directories

Use the tab to specify the default directories respectively for database extract, data export, data import, data load, table backup/restore and data dump.

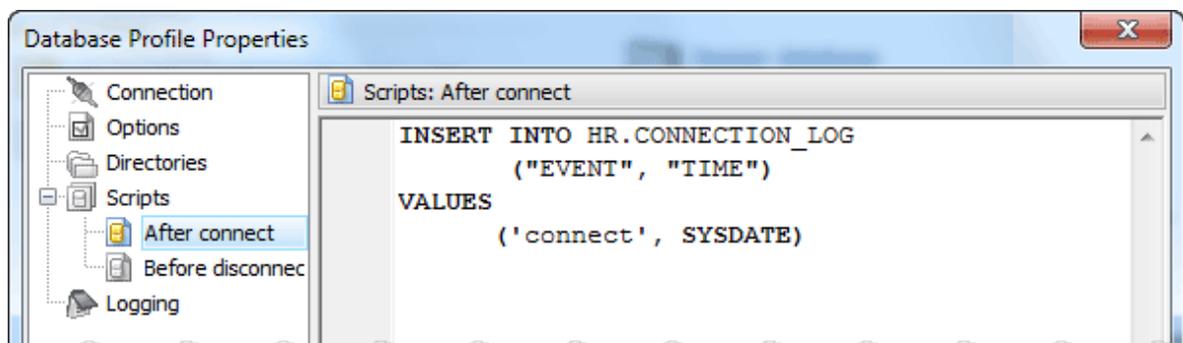


### 3.2.4 Editing obligatory scripts to execute

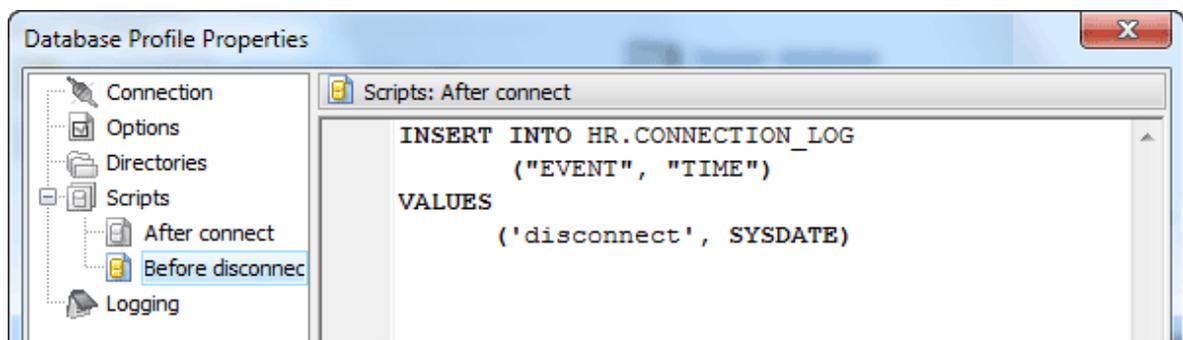
Use the tab to specify the obligatory scripts to execute in all database connections established by the software (on executing queries, browsing objects data, etc.). There is a possibility to enable/disable a written script.



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

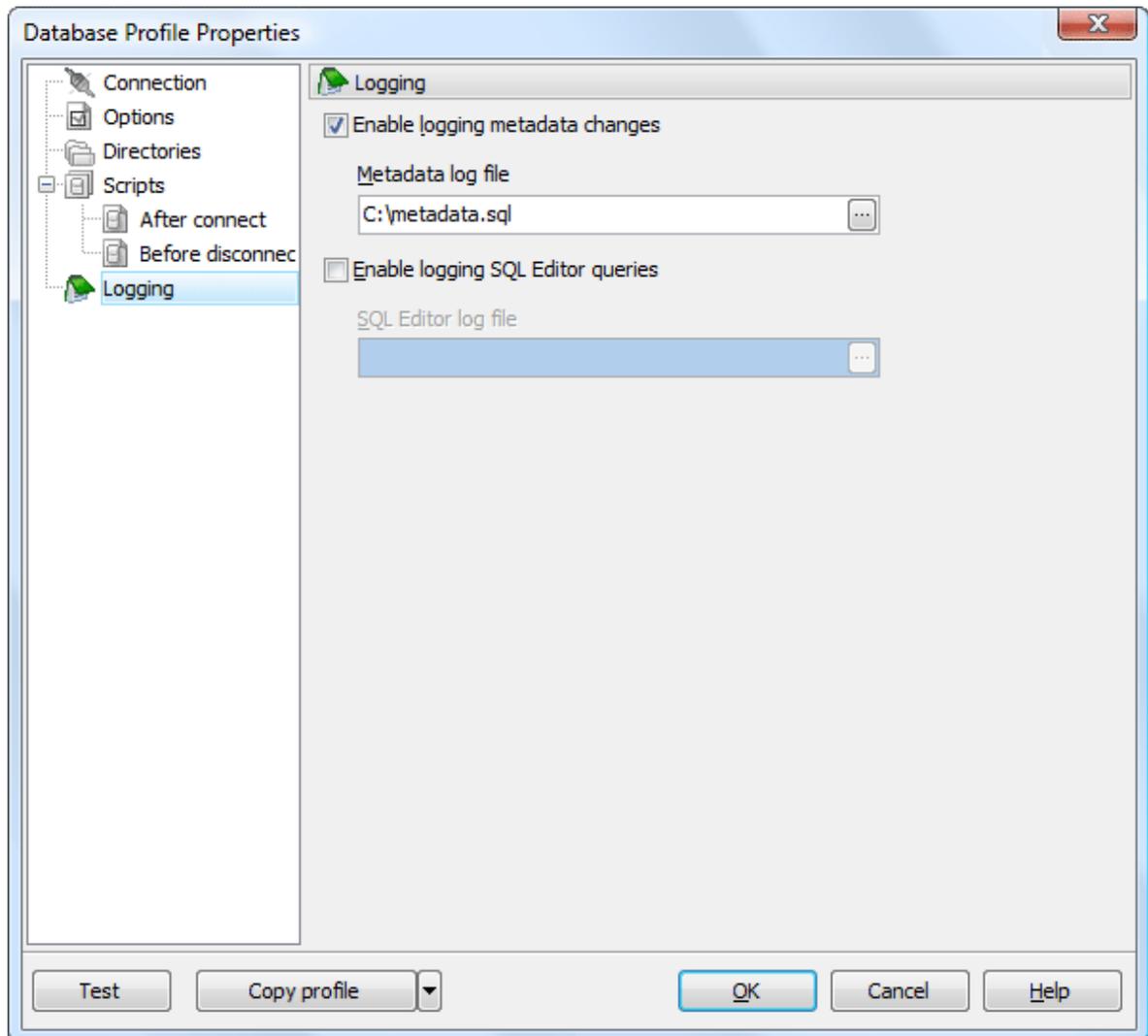


The next screen represents the example of an obligatory script to execute before Oracle Maestro will disconnect from the database. The script writes a disconnect time to the log table.



### 3.2.5 Setting log options

Enable/disable metadata changes logging and SQL query logging and specify the corresponding log file names if necessary.



### 3.2.6 Statistics

This tab allows you to view usage statistics for the current profile. Click the **Reset Statistics** button to clear all the displayed values.

**Statistics**

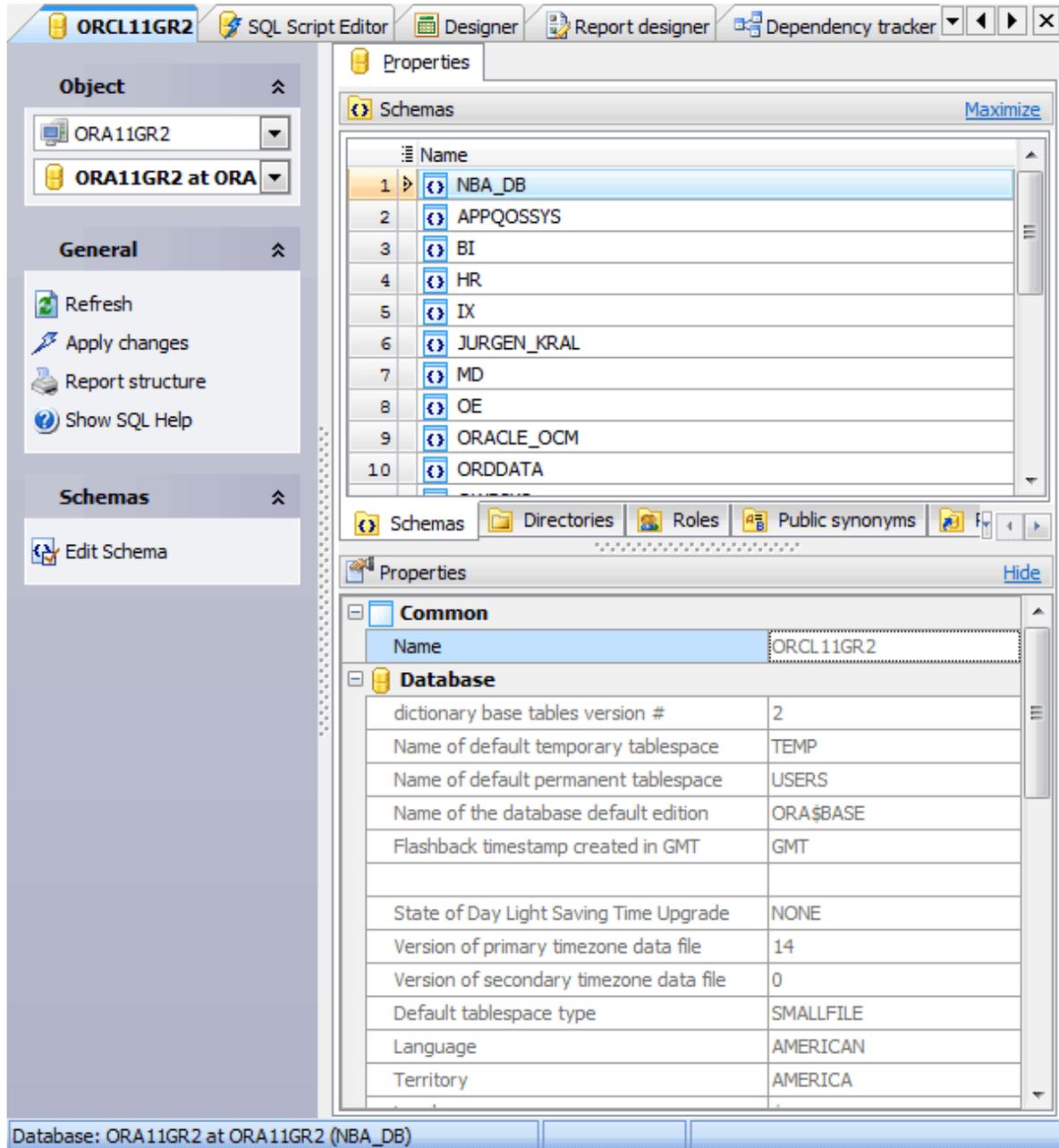
Statistics	
Creation time	N/A
Last modification time	N/A
Number of connections	6
Last connection time	18.08.2017 16:14:16
Total uptime	2:03:51:22

Reset statistics

### 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 Database Object Management

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

- [Browse Database Objects](#) <sup>51</sup>
- [Create New Objects](#) <sup>36</sup>
- [Edit Existing Objects](#) <sup>39</sup>
- [Duplicate Objects](#) <sup>46</sup>

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

## 4.1 Create Objects

Oracle Maestro provides a number of [Create Object Wizards](#)<sup>[37]</sup> to accomplish the most facile Oracle object creation.

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

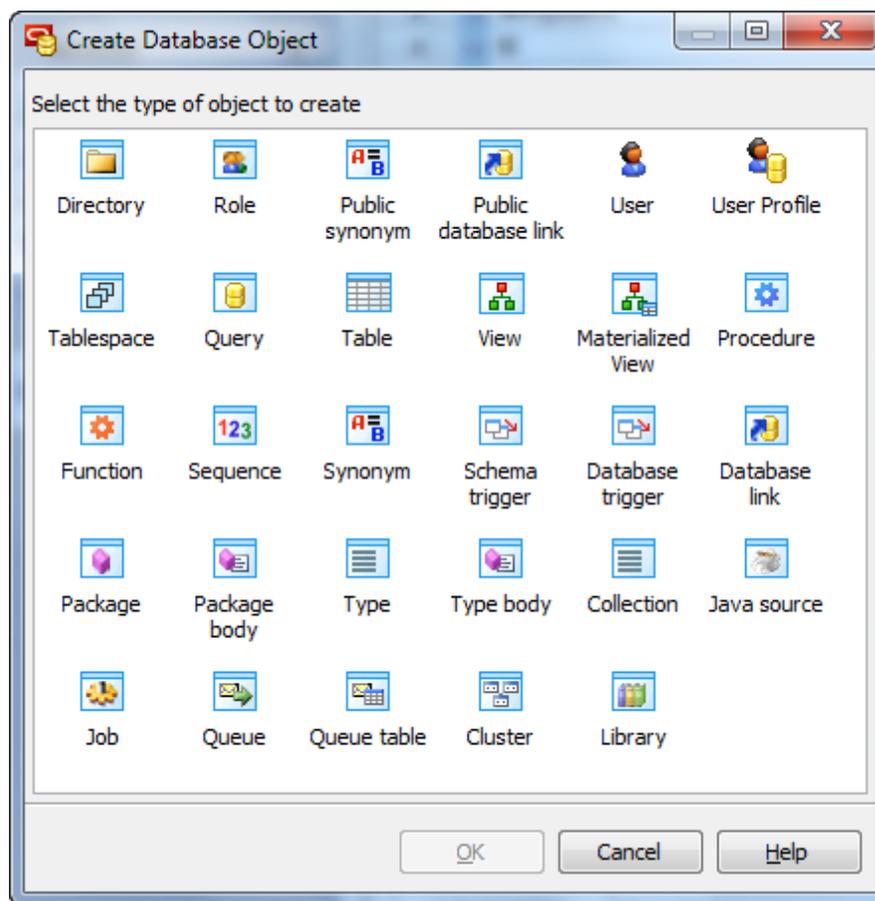
- select the **Object | Create Database Object...** main menu item;
- select the necessary icon (table, query, view, trigger, etc.) in the [Create Database Object](#)<sup>[38]</sup> dialog

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 Oracle 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>[38]</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:

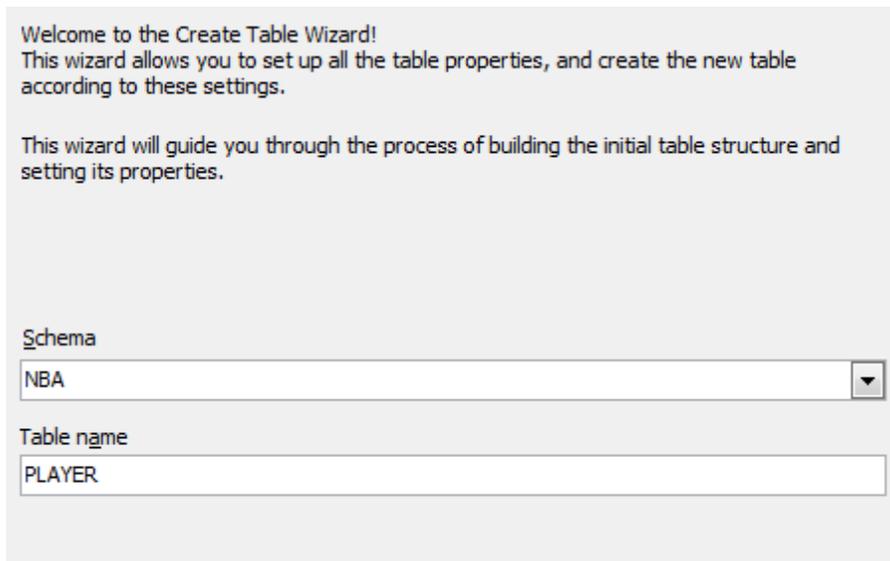
- [Create Table Wizard](#)<sup>[64]</sup>
- [Create View Wizard](#)<sup>[96]</sup>
- [Create Function Wizard](#)<sup>[123]</sup>
- [Create Sequence Wizard](#)<sup>[161]</sup>
- [Create Type Wizard](#)<sup>[166]</sup>
- [Creation of a New Query](#)<sup>[262]</sup>
- [Create User Wizard](#)<sup>[178]</sup>
- [Create Role Wizard](#)<sup>[191]</sup>
- [Create Procedure Wizard](#)<sup>[114]</sup>
- [Create Synonym Wizard](#)<sup>[130]</sup>
- [Create Directory Wizard](#)<sup>[234]</sup>
- [Create Database Trigger Wizard](#)<sup>[240]</sup>
- [Create Public Synonym Wizard](#)<sup>[245]</sup>
- [Create Database Link Wizard](#)<sup>[143]</sup>
- [Create Tablespace Wizard](#)<sup>[258]</sup>
- [Create User Profile Wizard](#)<sup>[183]</sup>
- [Create Materialized View Wizard](#)<sup>[106]</sup>
- [Create Schema Trigger Wizard](#)<sup>[134]</sup>
- [Create Public Database Link Wizard](#)<sup>[251]</sup>
- [Create Package Wizard](#)<sup>[148]</sup>
- [Create Package Body Wizard](#)<sup>[157]</sup>
- [Create Type Body Wizard](#)<sup>[172]</sup>
- [Create Java Source Wizard](#)<sup>[199]</sup>
- [Create Job Wizard](#)<sup>[205]</sup>
- [Create Queue Wizard](#)<sup>[211]</sup>
- [Create Queue Table Wizard](#)<sup>[216]</sup>
- [Create Cluster Wizard](#)<sup>[222]</sup>
- [Create Collection Wizard](#)<sup>[194]</sup>

- [Create Library Wizard](#)<sup>229</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 Oracle 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.

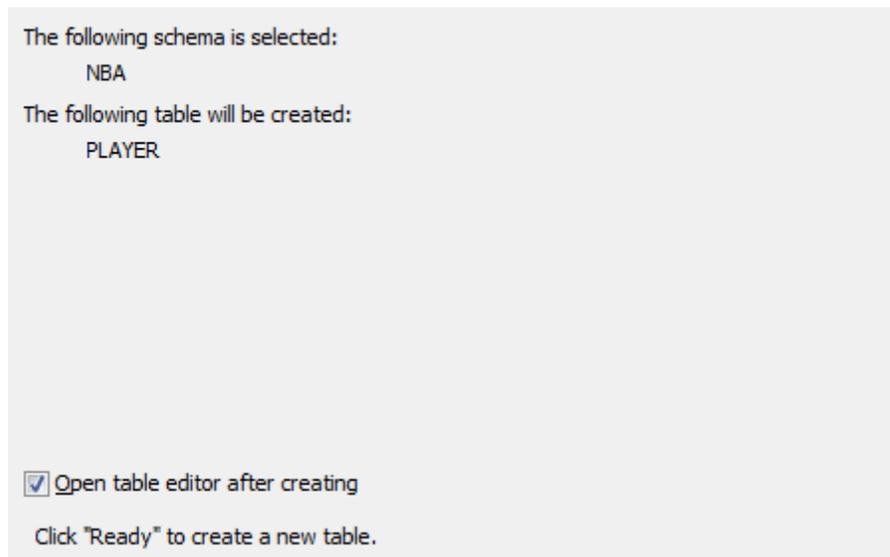
This wizard will guide you through the process of building the initial table structure and setting its properties.

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

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

- edit object comment with the [Describe Object](#)<sup>[46]</sup> dialog;
- briefly view and modify [object properties](#)<sup>[45]</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 Oracle Maestro tools for working with existing objects. The proper editor can be opened automatically after the object is created. You can also open the necessary object editor with the corresponding items of popup menus of the [Explorer Tree](#)<sup>[51]</sup>, [Object Manager](#)<sup>[36]</sup> or [Object Browser](#)<sup>[54]</sup>.

The editors consist of a several tabs. Some tabs are similar for all editors. This part purpose is to formulate the basic principles of all [Object Editors in Oracle Maestro](#).

- To edit object options such as name, owner, etc. use the [Properties](#) tab. To understand an option, see the appropriate topic of the corresponding [Object Editor](#) manual section and Oracle documentation.

This tab also allows you to manage objects belonging to the selected one. To reset any tab to default settings, open it when holding the **Ctrl** key.

- 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>[41]</sup> tab.
- Object correlation with another Oracle objects is represented on [Dependencies](#)<sup>[42]</sup>.
- 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>[43]</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>[44]</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:**

- [Table Editor](#) [69]
- [View Editor](#) [101]
- [Function Editor](#) [125]
- [Sequence Editor](#) [162]
- [Type Editor](#) [169]
- [User Editor](#) [179]
- [Role Editor](#) [192]
- [Procedure Editor](#) [118]
- [Synonym Editor](#) [131]
- [Directory Editor](#) [236]
- [Database Trigger Editor](#) [242]
- [Public Synonym Editor](#) [247]
- [Database Link Editor](#) [145]
- [Tablespace Editor](#) [259]
- [User Profile Editor](#) [186]
- [Materialized View Editor](#) [110]
- [Schema Trigger Editor](#) [138]
- [Public Database Link Editor](#) [253]
- [Package Editor](#) [153]
- [Package Body Editor](#) [158]
- [Type Body Editor](#) [174]
- [Java Source Editor](#) [201]
- [Job Editor](#) [207]
- [Queue Editor](#) [212]
- [Queue Table Editor](#) [218]
- [Cluster Editor](#) [225]
- [Collection Editor](#) [196]
- [Library Editor](#) [230]

#### 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 ( *schemas, tables, views, procedures, functions, synonyms, types, queues, tablespaces, etc.*) to one or more users.

Grantee	Create	Select	Insert	Update	Delete	Drop
<b>Users</b>						
accountants@%		●	●	●		
admin@%	●	●		●	●	●
alex@%		●				
clerks@MERCURY	●	●				
developers@URAN	●	●	●			●
john@%		●				
john@localhost		●				
managers@%		●				
mary@%		●				
root@%		●				
root@localhost		●				

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

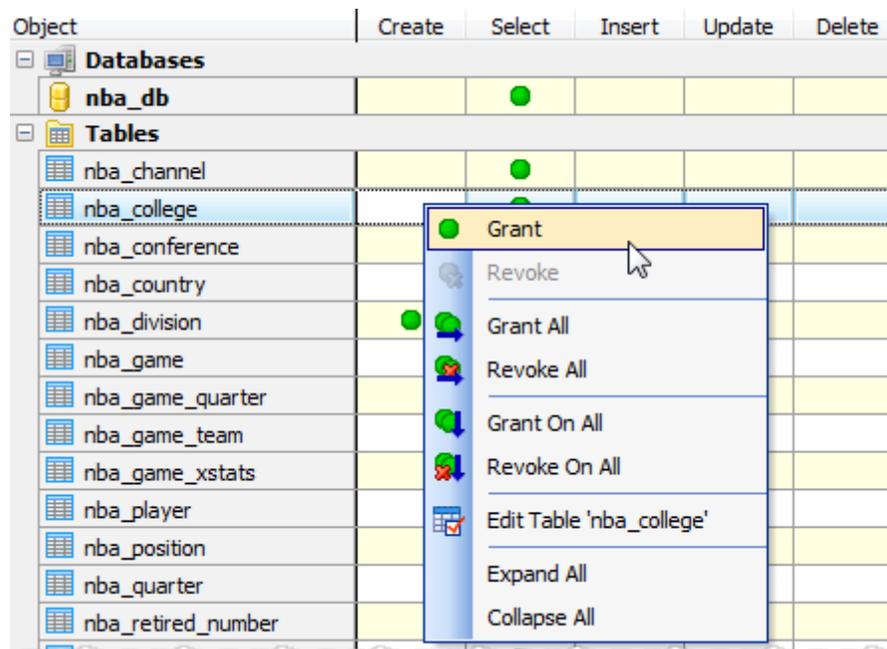
**See also:** [Users](#)<sup>[178]</sup>, and [Roles](#)<sup>[190]</sup>

#### 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.

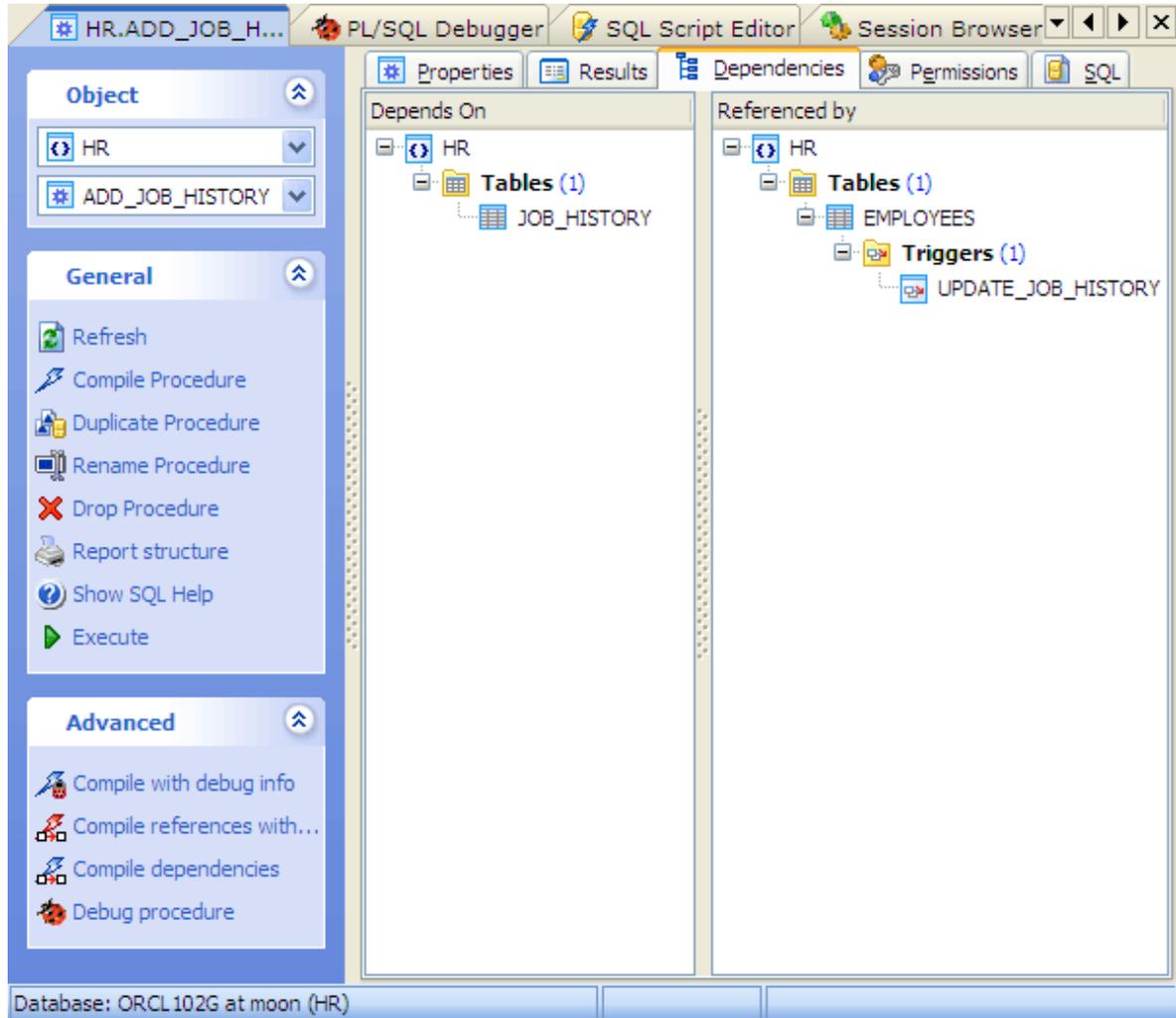
#### 4.2.1.3 Object Dependencies

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>390</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)

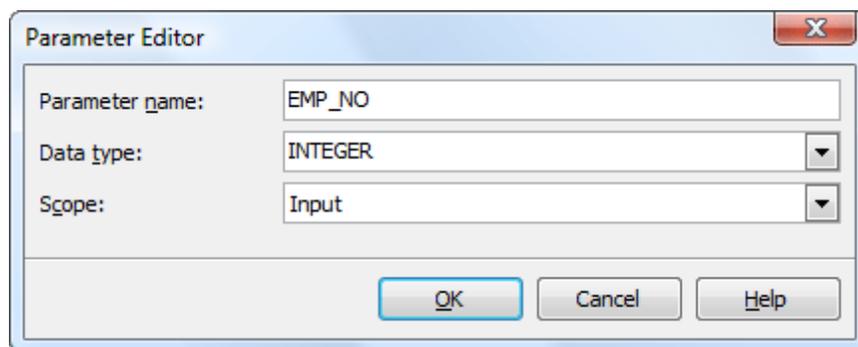
In [collapsed mode](#) the text is hidden behind one text line denoting the first line of the collapsed region.

[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](#).

Oracle supports the [scope](#) function parameters (*Input* is the default value).



#### 4.2.1.6 Executing functions and procedures

[Procedure 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 [procedure](#) has parameters, Oracle 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		
History			
Date and time	@conference_id	@division_id	@team_id
▶ 18.02.2010 17:44:44	2	2	1
18.02.2010 17:44:19	1	1	4
18.02.2010 17:42:41	1	3	2
18.02.2010 17:42:18	1	2	7

Oracle 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 [Procedure 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 Oracle 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

Oracle 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>[46]</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>[48]</sup> you can find some additional info.
3. By **Drag-n-Drop**<sup>[49]</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 Oracle 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.

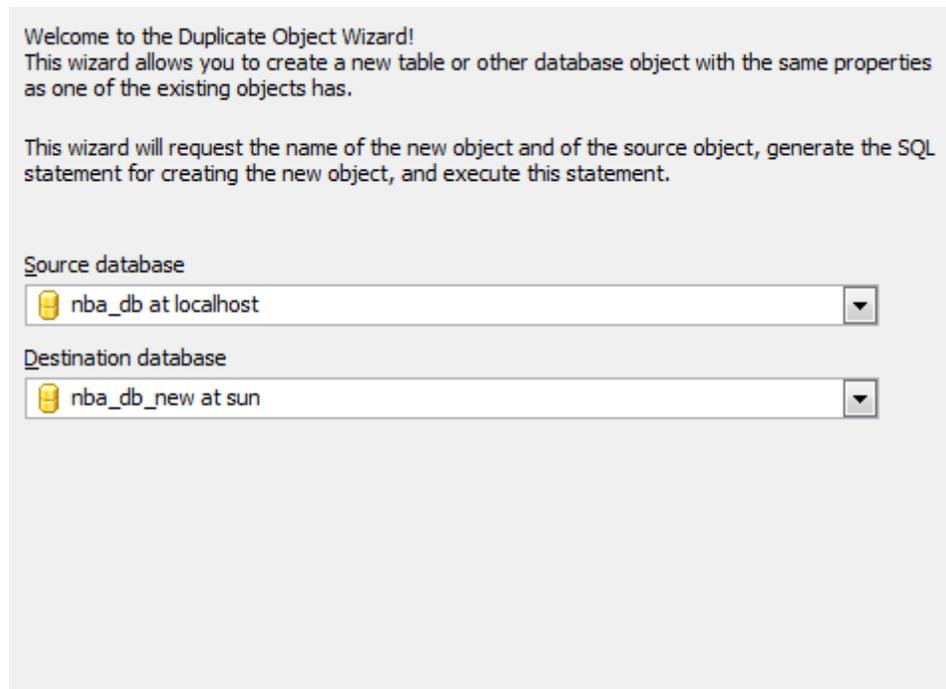
- [Selecting source and destination databases](#)<sup>[46]</sup>
- [Selecting object to duplicate](#)<sup>[47]</sup>
- [Modifying definition of a new object](#)<sup>[48]</sup>

**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.

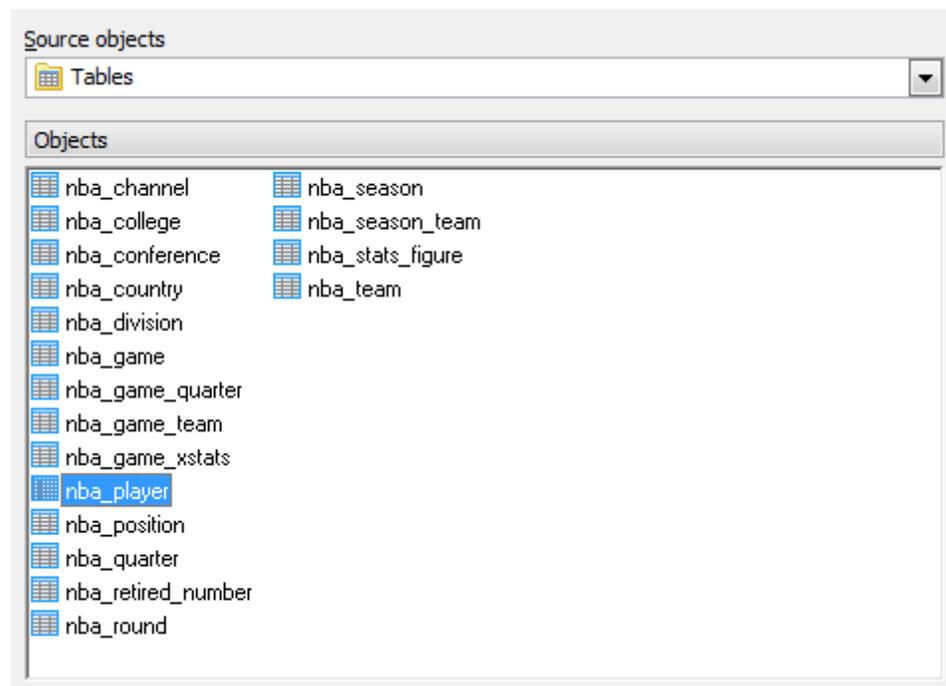
You should connect to the destination database beforehand (see [Database Management](#)<sup>[21]</sup>).



#### 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.



### 4.3.1.3 Modifying new object definition

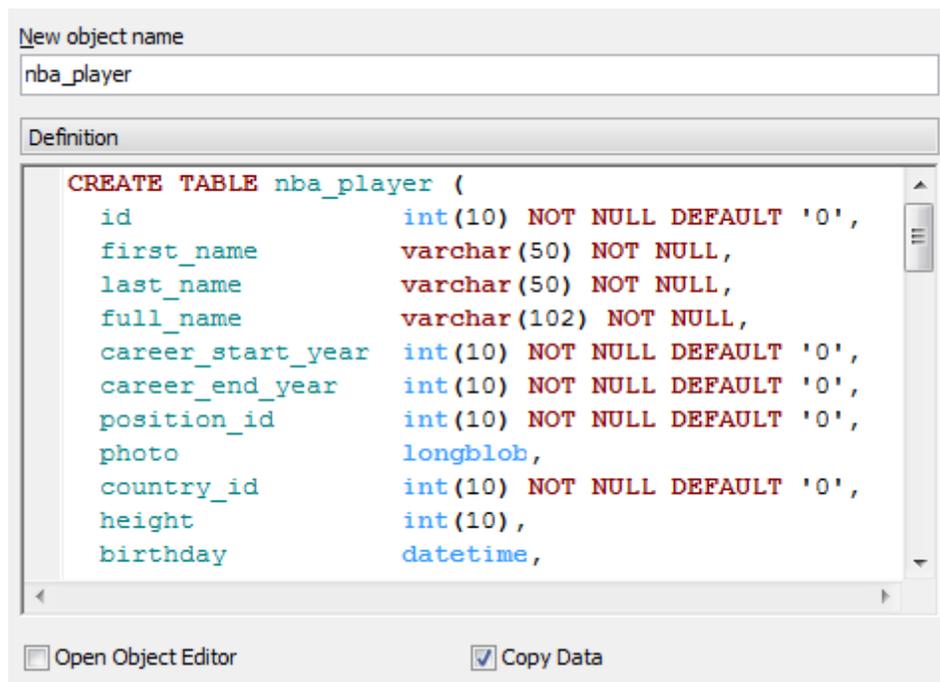
The last wizard step allows you to edit the new object definition directly.

Use this step to edit the name of object been creating ([New Object Name](#)). By default Oracle 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 Oracle 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

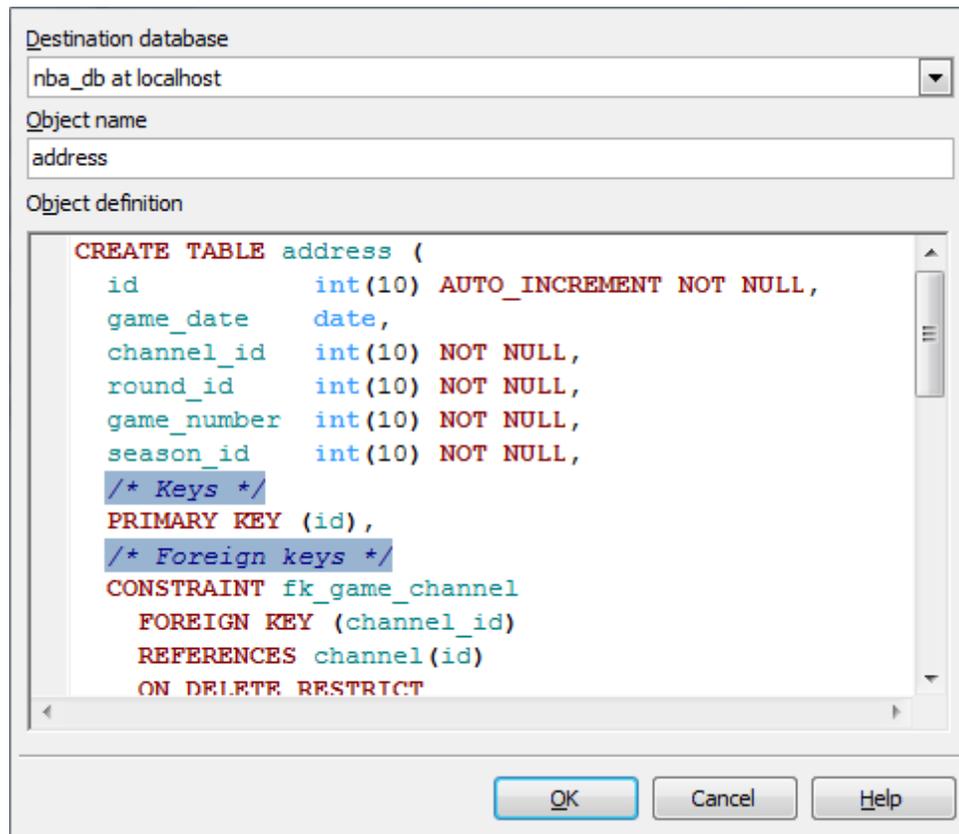
```
CREATE TABLE nba_player (  
  id          int(10) NOT NULL DEFAULT '0',  
  first_name  varchar(50) NOT NULL,  
  last_name   varchar(50) NOT NULL,  
  full_name   varchar(102) NOT NULL,  
  career_start_year int(10) NOT NULL DEFAULT '0',  
  career_end_year int(10) NOT NULL DEFAULT '0',  
  position_id 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 Oracle 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

Oracle 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

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

- [Database Explorer](#)<sup>[51]</sup>: objects are represented as a hierarchy (grouped by kind and listed under the according Oracle database node, provided with subobjects if exist)
- [Object Browser](#)<sup>[54]</sup>: an extension of explorer with ability to sort, group, filter and multiple select objects.
- [Object Manager](#)<sup>[56]</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 Oracle 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 Oracle Maestro main window. All the objects at the Explorer tree are grouped by kind and listed under the according Oracle 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.

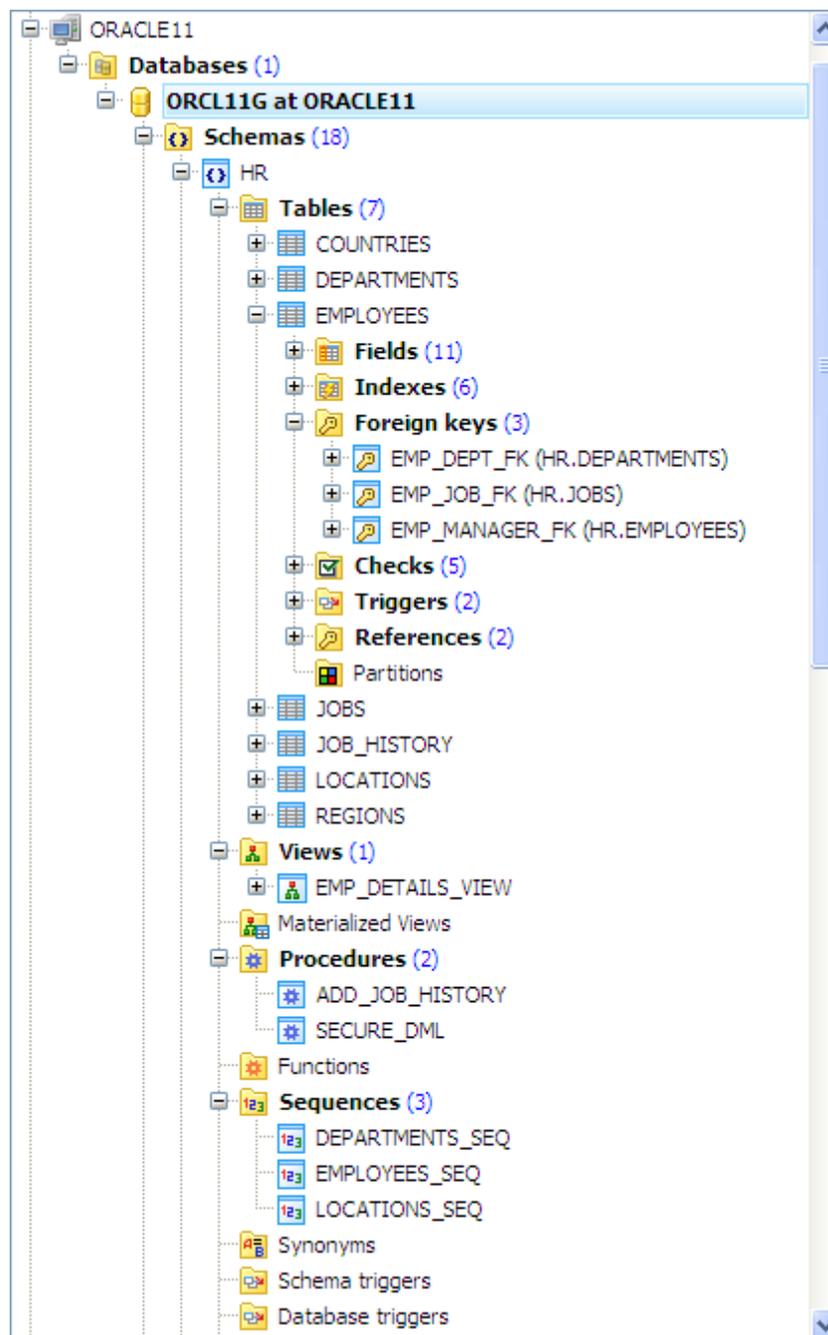
**See also:** [Object Manager](#)<sup>[56]</sup>, [Object Browser](#)<sup>[54]</sup>

**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>[373]</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>[51]</sup>
- [How can I disconnect from a database?](#)<sup>[53]</sup>
- [What operations can I accomplish upon database objects within the Explorer Tree?](#)<sup>[53]</sup>
- [Can I copy a database object from one database to another?](#)<sup>[53]</sup>
- [Can I filter Explorer content?](#)<sup>[54]</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](#) <sup>[23]</sup> (the Create Database Profiles... item);
- rename currently selected database profile (the Rename Database Profile... item);
- [edit currently selected database profile](#) <sup>[23]</sup> (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?**

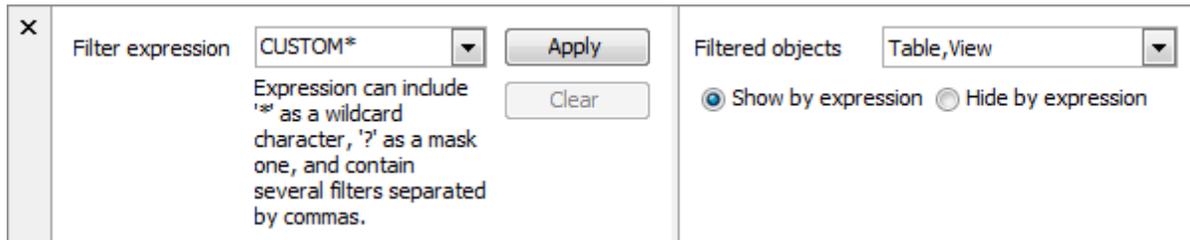
Database Explorer provides you with an ability of copying database objects from one database to another. To perform this operation, you should connect to both the source and the destination databases first. After the connection is established, simply drag and drop an object to copy from the source database 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

Oracle 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.



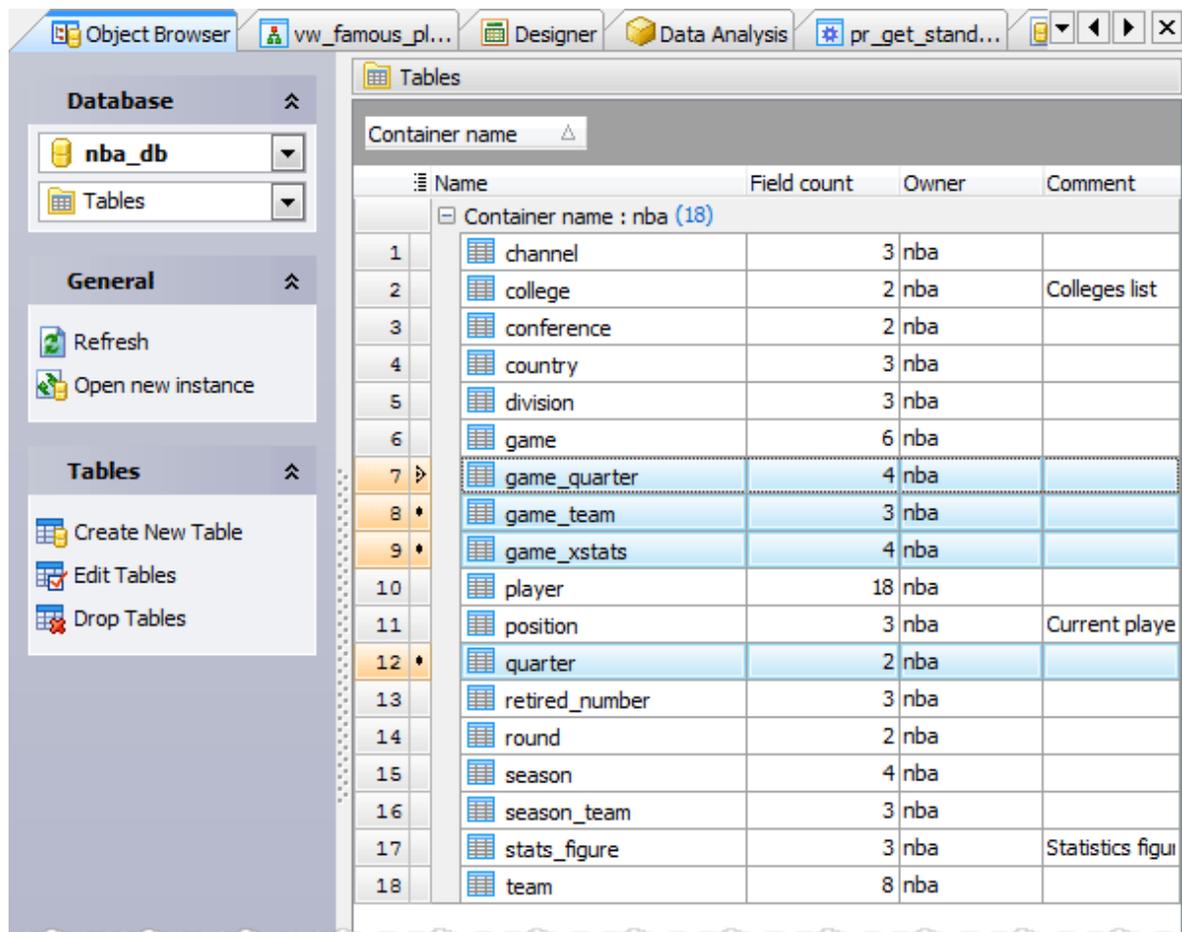
- Specify the Filter expression. The expression can contain any part of object name combined with an asterisk ('\*') as a wildcard character and a question-mark ('?') as a mask character.
- Define the Filtered objects, object types for filtering in the explorer tree.
- Check the according radio button (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](#)<sup>[53]</sup> and [Object Browser](#)<sup>[54]</sup> as well.

#### 4.4.2 Object Browser

[Object Browser](#) is a tool for operating on database objects designed as an extension of [Database Explorer](#)<sup>[51]</sup> 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 Oracle 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

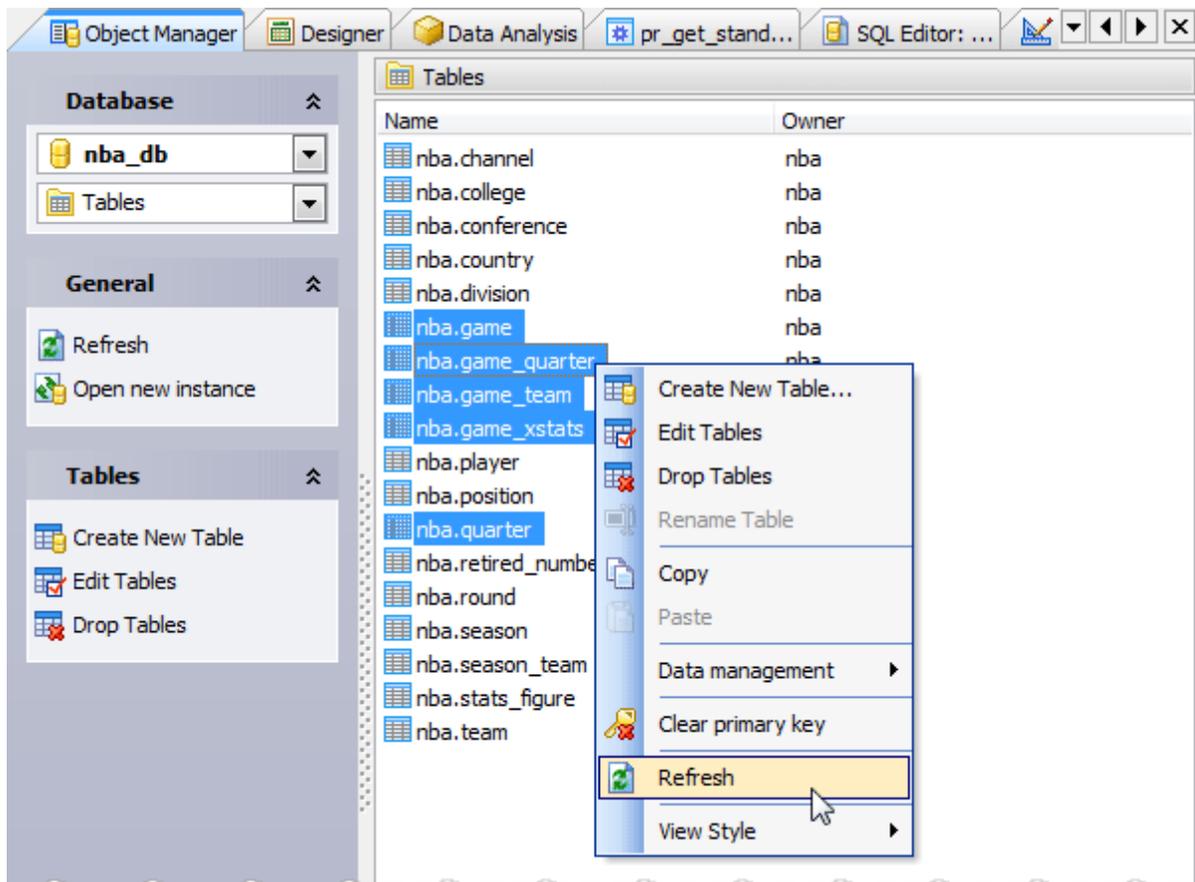
After you set a filter, the filtering panel becomes visible at the bottom of the grid where you can see the active filtering condition and easily enable or disable it by clicking the check box on the left. Using this panel you can also customize your filter in a more complicated way by clicking the [Customize](#)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

**See also:** [Object Manager](#)<sup>[56]</sup>, [Data View](#)<sup>[279]</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>[51]</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](#) <sup>54</sup>

## 4.4.4 Filter Builder Dialog

FilterBuilderDialog 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 Oracle Maestro. To work with database objects you should [connect to the database](#) first.

- [Tables](#) <sup>63</sup>
- [Views](#) <sup>95</sup>
- [Functions](#) <sup>122</sup>
- [Sequences](#) <sup>160</sup>
- [Types](#) <sup>165</sup>
- [Users](#) <sup>178</sup>
- [Roles](#) <sup>190</sup>
- [Procedures](#) <sup>113</sup>
- [Synonyms](#) <sup>129</sup>
- [Directories](#) <sup>233</sup>
- [Database Triggers](#) <sup>239</sup>
- [Public Synonyms](#) <sup>244</sup>
- [Database Links](#) <sup>142</sup>
- [Tablespaces](#) <sup>255</sup>
- [User Profiles](#) <sup>182</sup>
- [Materialized Views](#) <sup>105</sup>
- [Schema Triggers](#) <sup>133</sup>
- [Public Database Links](#) <sup>250</sup>
- [Packages](#) <sup>147</sup>
- [Package Bodies](#) <sup>156</sup>
- [Type Bodies](#) <sup>171</sup>
- [Java Sources](#) <sup>198</sup>
- [Jobs](#) <sup>204</sup>
- [Queues](#) <sup>210</sup>
- [Queue Tables](#) <sup>215</sup>
- [Clusters](#) <sup>221</sup>
- [Collections](#) <sup>193</sup>
- [Libraries](#) <sup>228</sup>

## 5.1 Schemas

A schema is a collection of database objects. A schema is owned by a database user and has the same name as that user. Schema objects are the logical structures that directly refer to the database's data.

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

Schemas are edited within [Schema Editor](#)<sup>[61]</sup>. 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

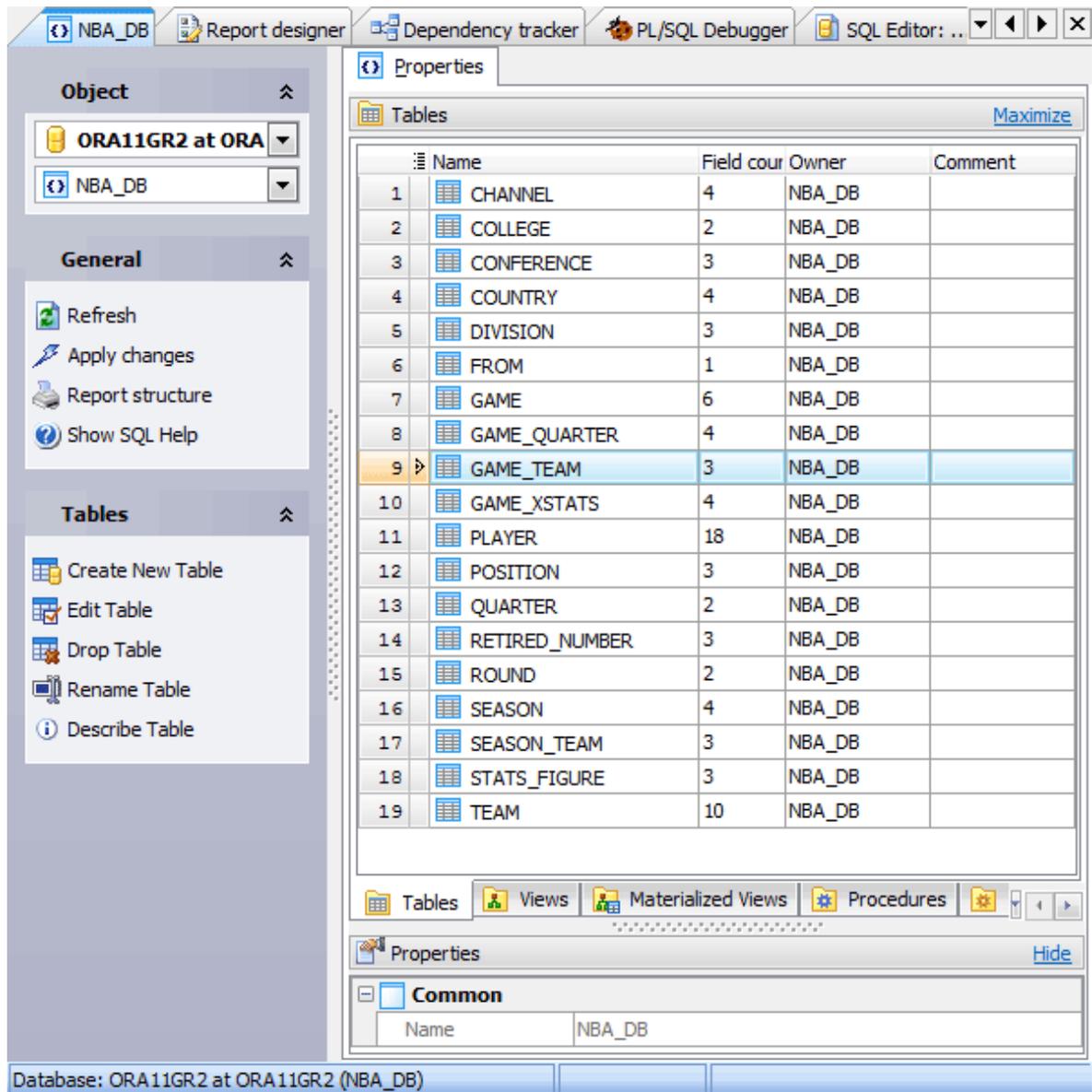
- open the database in [Database Editor](#) and the [Schemas](#) tab there;
- select the schema to drop;
- press the **Delete** key or select the [Drop Schema](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#))

and confirm dropping in the dialog window.

## 5.1.1 Schema Editor

[Schema Editor](#) allows you to browse schema content, manage users permissions on the schema objects, and see the SQL definition of this schema.

The basic principles of Object Editors in Oracle Maestro are explained in a [separate topic](#) [30]. Below you will find a description of editor tabs that are unique for the current object.



The [Properties](#) tab allows you to view schema options and to browse schema content divided into groups according to their types (*tables, views, functions, etc.*). The popup menu of each tab allows you to create new, edit, copy or drop the appropriate schema object. The grid allows you to operate with several objects at a time. For this purpose select 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.

#### Owner

This field allows you to view the schema owner.

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

Oracle 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>[64]</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>[65]</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>[72]</sup>, generate [simple SQL statements](#)<sup>[355]</sup>, [CRUD procedures](#)<sup>[356]</sup> to work with this table, and [split the table](#)<sup>[359]</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. To create a relational table in your own schema, you must have the CREATE TABLE system privilege. To create a table in another user's schema, you must have the CREATE ANY TABLE system privilege. Also, the owner of the schema to contain the table must have either space quota on the tablespace to contain the table or the UNLIMITED TABLESPACE system privilege.

The basic principles of Create Object Wizards in Oracle Maestro are explained in a [separate topic](#)<sup>[37]</sup>. Below you will find a description of wizard steps that are unique for the current object.

#### [Name](#)

The name of the table being created as it was specified at the previous step.

#### [Owner](#)

You can specify here the name of the Oracle server user that will own the new table, or leave this field blank to use the default user (namely, the user executing the command).

#### [Comment](#)

Set the optional text to describe the new table.

### Storage option

Specify here the physical attributes and tablespace storage for the new table. For more information see [Storage option properties](#) <sup>(67)</sup>.

### Organization (STANDARD, INDEX-ORGANIZED, CLUSTERED)

The clause lets you specify the order in which the data rows of the table are stored.

#### Has cache

For data that is accessed frequently, this clause indicates that the blocks retrieved for this table are placed at the most recently used end of the least recently used (LRU) list in the buffer cache when a full table scan is performed. This attribute is useful for small lookup tables. You cannot check it for an index-organized table. However, index-organized tables implicitly provide [Has cache](#) behavior.

#### Has monitoring

In the releases up to 10.1, you could use these clauses to start or stop the collection of modification statistics on this table. Since Oracle 10.1 these clauses have been deprecated.

#### Has logging

Specify whether the creation of the table and of any indexes required because of constraints, partition, or LOB storage characteristics will be logged in the redo log file (on) or not (off). The logging attribute of the table is independent of that of its indexes.

### Degree

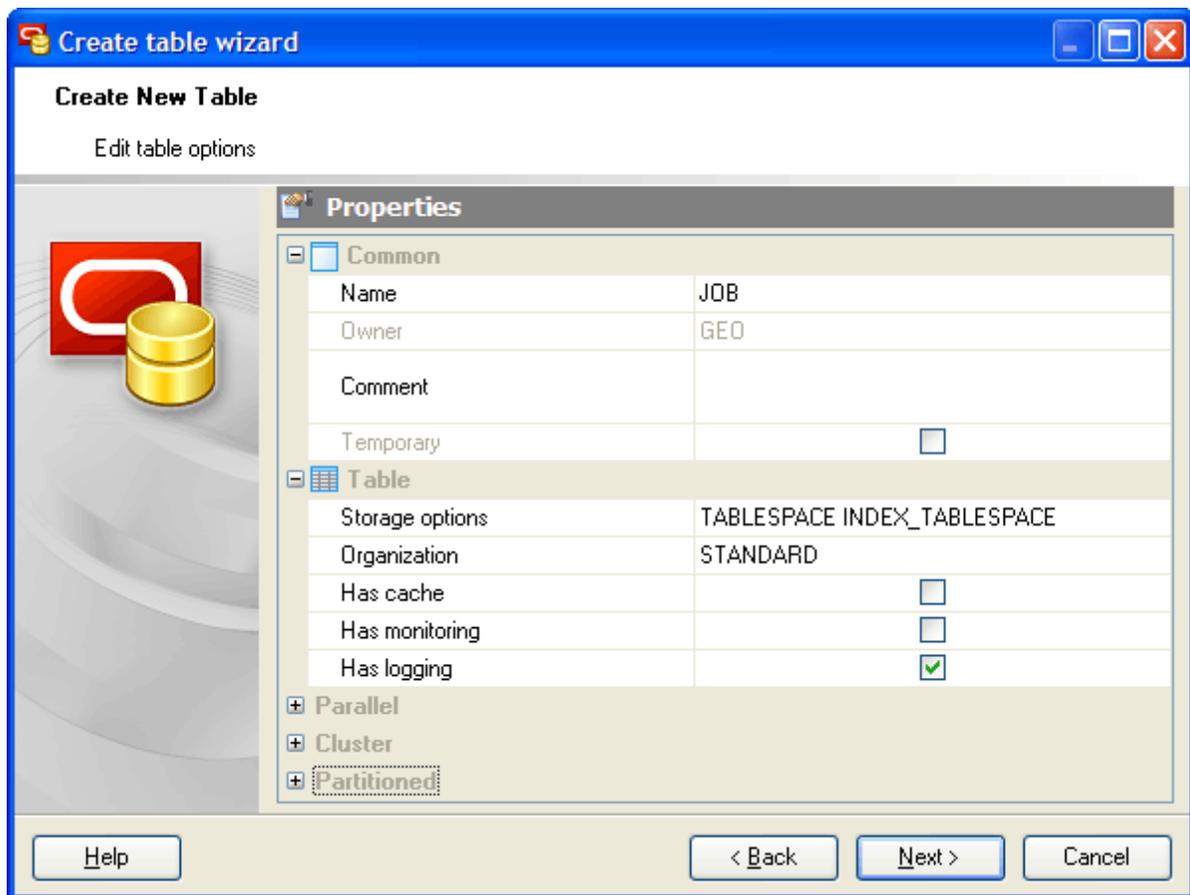
Specify here the degree of parallelism, which is the number of parallel threads used in the parallel operation. Each parallel thread may use one or two parallel execution servers. Normally Oracle calculates the optimum degree of parallelism, so it is not necessary for you to specify integer.

### Cluster name

Specify the cluster to contain data of the table being creating.

#### Row movement

The option allows you to restrict the row movement in the newly table. You cannot specify this clause for a nonpartitioned index-organized table.



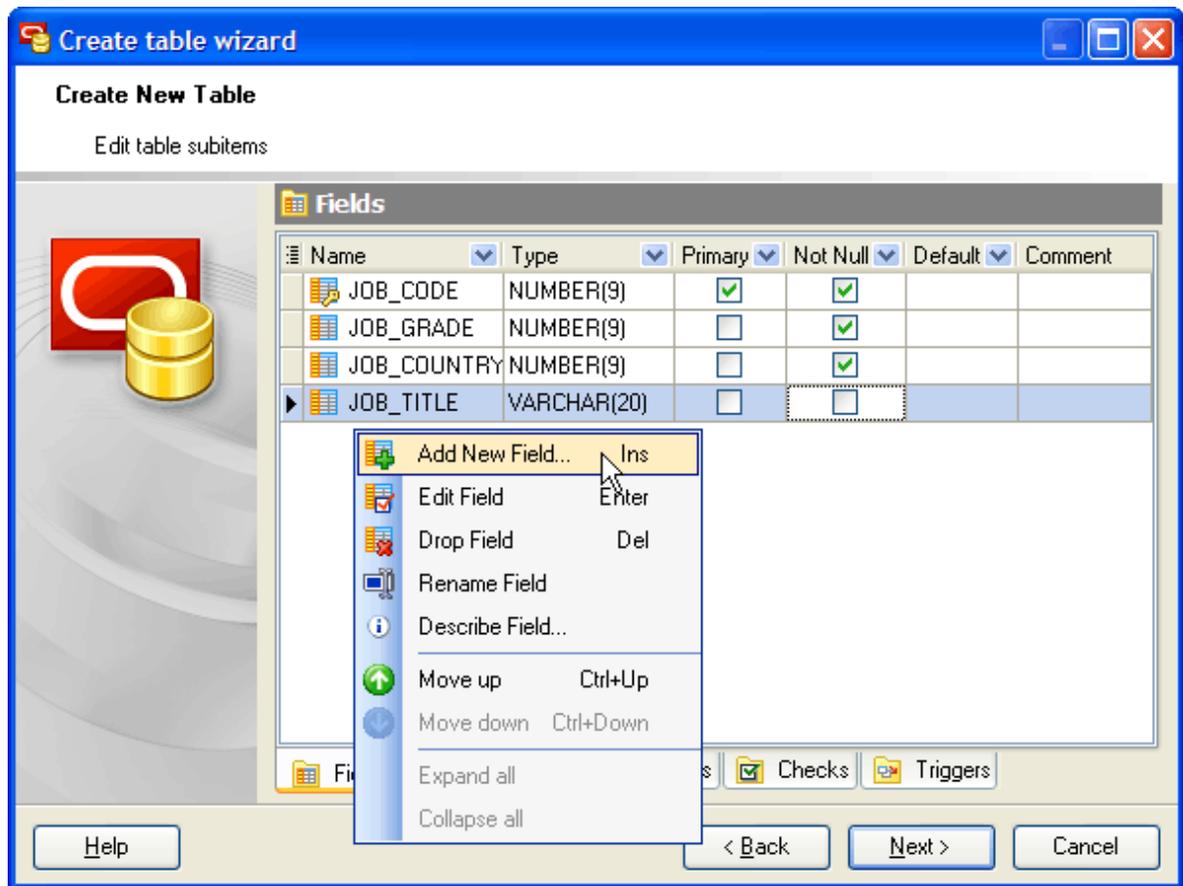
### Adding table subitems

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>[75]</sup>, [foreign\\_key](#)<sup>[82]</sup>, [check](#)<sup>[85]</sup>, [trigger](#)<sup>[88]</sup>, and [index](#)<sup>[79]</sup>, follow the according link.

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

**Note**, that the absolute maximum number of columns in a table is 1000.



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.

### 5.2.1.1 Storage option properties

#### Tablespace Name

Define the tablespace name in which Oracle Database creates the table, object table OID index, partition, LOB data segment, LOB index segment, or index-organized table overflow data segment. If you omit the field, then the database creates that item in the default tablespace of the owner of the schema containing the table.

#### Free (%)

Specify a whole number representing the percentage of space in each data block of the database object reserved for future updates to rows of the object. The value of PCTFREE must be a value from 0 to 99. A value of 0 means that the entire block can be filled by inserts of new rows. The default value is 10. This value reserves 10% of each block for updates to existing rows and allows inserts of new rows to fill a maximum of 90% of each block.

#### Used (%)

Specify a whole number representing the minimum percentage of used space that Oracle maintains for each data block of the database object. A block becomes a candidate for row insertion when its used space falls below PCTUSED. PCTUSED is specified as a

positive integer from 0 to 99 and defaults to 40.

#### Init trans

Specify the initial number of concurrent transaction entries allocated within each data block allocated to the database object. This value can range from 1 to 255 and defaults to 1. In general, you should not change the `Init trans` value from its default.

#### Max trans

The parameter determine the maximum number of concurrent update transactions allowed for each data block in the segment. Since Oracle 10.1 this parameter has been deprecated. In latest versions Oracle automatically allows up to 255 concurrent update transactions for any data block, depending on the available space in the block.

#### Increase (%)

Specify the percent by which the third and subsequent extents grow over the preceding extent. The default value is 50, meaning that each subsequent extent is 50% larger than the preceding extent. The minimum value is 0, meaning all extents after the first are the same size. The maximum value depends on your operating system.

#### Initial extent

Specify in bytes the size of the first extent of the object. Oracle allocates space for this extent when you create the schema object. Use K or M to specify this size in kilobytes or megabytes.

#### Next extent

Specify in bytes the size of the next extent to be allocated to the object. Use K or M to specify the size in kilobytes or megabytes. The default value is the size of 5 data blocks. The minimum value is the size of 1 data block. The maximum value depends on your operating system.

#### Min extents

Specify the total number of extents to allocate when the object is created. This parameter lets you allocate a large amount of space when you create an object, even if the space available is not contiguous. The default and minimum value is 1, meaning that Oracle allocates only the initial extent, except for rollback segments, for which the default and minimum value is 2. The maximum value depends on your operating system.

#### Max extents

Specify the total number of extents, including the first, that Oracle can allocate for the object. The minimum value is 1 except for rollback segments, which always have a minimum of 2. The default value depends on your data block size.

#### Buffer pool

Use the clause lets to specify a default buffer pool or cache for a schema object. All blocks for the object are stored in the specified cache.

**Note:** You cannot specify this clause for a cluster table. However, you can specify it for a cluster.

#### Freelist groups

Specify the number of groups of free lists for the database object you are creating. The default and minimum value for this parameter is 1. Oracle uses the instance number of Real Application Clusters instances to map each instance to one free list group.

### Freelists

For objects other than tablespaces and rollback segments, specify the number of free lists for each of the free list groups for the table, partition, cluster, or index. The default and minimum value for this parameter is 1, meaning that each free list group contains one free list. The maximum value of this parameter depends on the data block size. If you specify a [Freelists](#) value that is too large, then Oracle returns an error indicating the maximum value.

## 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 Oracle Maestro are explained in a [separate topic](#) <sup>[39]</sup>. Below you will find a description of editor tabs that are unique for the current object.

- [Editing table properties](#) <sup>[69]</sup>
- [Viewing table data](#) <sup>[72]</sup>
- [Gathering statistic properties](#) <sup>[74]</sup>

To compile all the table dependencies as well as to compile all the table references with debug information at once, use the popup menu of the table node at the Explorer tree.

### 5.2.2.1 Editing table properties

The [Properties](#) section allows you to view general table properties and also to modify the table name a comment for the table.

The screenshot displays the Oracle SQL Developer interface for the 'EMPLOYEES@HR' table. The 'Fields' tab is selected, showing a table with the following columns:

Name	Type	Not Null	Default
EMPLOYEE_ID	NUMBER	<input checked="" type="checkbox"/>	
FIRST_NAME	VARCHAR2	<input type="checkbox"/>	
LAST_NAME	VARCHAR2	<input checked="" type="checkbox"/>	
EMAIL	VARCHAR2	<input checked="" type="checkbox"/>	
PHONE_NUMBER	VARCHAR2	<input type="checkbox"/>	
HIRE_DATE	DATE	<input checked="" type="checkbox"/>	
JOB_ID	VARCHAR2	<input checked="" type="checkbox"/>	
SALARY	NUMBER	<input type="checkbox"/>	
COMMISSION_PCT	NUMBER	<input type="checkbox"/>	
MANAGER_ID	NUMBER	<input type="checkbox"/>	
DEPARTMENT_ID	NUMBER	<input type="checkbox"/>	

The 'Properties' tab is also visible, showing the following properties:

Property	Value
Name	EMPLOYEES
Comment	employees table. Contains 107 rows. References with departments,
Created	18.11.2008 15:21:39
Last DDL time	18.11.2008 15:23:03
Time stamp	2008-11-18:15:21:39
Temporary	<input type="checkbox"/>
<b>Table</b>	
Storage options	TABLESPACE EXAMPLE
Organization	STANDARD
Has cache	<input type="checkbox"/>
Has compression	<input type="checkbox"/>
Has monitoring	<input checked="" type="checkbox"/>
Has logging	<input type="checkbox"/>
<b>Parallel</b>	
Instances	1
Degree	1

### 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>[75]</sup>, [Foreign Keys](#)<sup>[82]</sup>, [Checks](#)<sup>[86]</sup>, [Triggers](#)<sup>[90]</sup>, and [Indexes](#)<sup>[79]</sup>.

Use the [Name](#) field to rename the table.

**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 Oracle server.

#### Owner

You can view the name of the table [Owner](#).

#### Comment

This field contains a comment to the table.

#### Storage option

Edit here the physical attributes and tablespace storage for the new table. For more information see [Storage option properties](#)<sup>[67]</sup>.

#### Organization (STANDARD, INDEX-ORGANIZED, CLUSTERED)

The field contains a sort of the order in which the data rows of the table are stored.

#### Has cache

For data that is accessed frequently, this clause indicates that the blocks retrieved for this table are placed at the most recently used end of the least recently used (LRU) list in the buffer cache when a full table scan is performed. This attribute is useful for small lookup tables. You cannot check it for an index-organized table. However, index-organized tables implicitly provide [Has cache](#) behavior.

#### Has monitoring

In the releases up to 10.1, you could use these clauses to start or stop the collection of modification statistics on this table. Since Oracle 10.1 these clauses have been deprecated.

#### Has logging

Specify whether the creation of the table and of any indexes required because of constraints, partition, or LOB storage characteristics will be logged in the redo log file (on) or not (off). The logging attribute of the table is independent of that of its indexes.

#### Degree

Specify here the degree of parallelism, which is the number of parallel threads used in the parallel operation. Each parallel thread may use one or two parallel execution servers. Normally Oracle calculates the optimum degree of parallelism, so it is not necessary for you to specify integer.

#### Row movement

The option allows you to restrict the row movement in the newly table. You cannot specify this clause for a nonpartitioned index-organized table.

The editor also displays some additional information for clustered, index organized and partitioned tables and other info including statistic data.

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>[279]</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>[290]</sup>.

#### Lookup editors

Lookup editor displays the content of parent table's columns within the drop-down window. Oracle 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		address_id	address	district	city_id	postal_code
1	5	5	1913 Hanoi Way	Nagasaki	463	35200
2	6	6	1121 Loja Avenue	California	449	17886
2	7	7	692 Joliet Street	Attika	38	83579
1	8	8	1566 Inegl Manor	Mandalay	349	53561
2	9	9	53 Idfu Parkway	Nantou	361	42399
1	10	10	1795 Santiago de Compostela Way	Texas	295	18743
2	11	11	900 Santiago de Compostela Parkway	Central Serbia	280	93896
2	12	12	478 Joliet Way	Hamilton	200	77948
1						
2						
1	21	DONNA	THOMPSON	DONNA.THOMPSON@sakilacustomer.org		

#### 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																																																																													
<div style="border: 1px solid gray; padding: 5px;"> <p>1 SalesOrderDetail (ProductID)</p> <table border="1"> <thead> <tr> <th></th> <th>SalesOrderID</th> <th>SalesOrderDetailID</th> <th>OrderQty</th> <th>ProductID</th> <th>UnitPrice</th> <th>UnitPriceDisc</th> </tr> </thead> <tbody> <tr> <td></td> <td colspan="6" style="text-align: center;">Click here to define a filter</td> </tr> <tr> <td>1</td> <td>71938</td> <td>113283</td> <td>1</td> <td>712</td> <td>5,394</td> <td></td> </tr> <tr> <td>2</td> <td>71897</td> <td>112902</td> <td>4</td> <td>712</td> <td>5,394</td> <td></td> </tr> <tr> <td>3</td> <td>71858</td> <td>112375</td> <td>3</td> <td>712</td> <td>5,394</td> <td></td> </tr> <tr> <td>4</td> <td>71902</td> <td>112962</td> <td>3</td> <td>712</td> <td>5,394</td> <td></td> </tr> <tr> <td>5</td> <td>71797</td> <td>111053</td> <td>6</td> <td>712</td> <td>5,394</td> <td></td> </tr> <tr> <td>6</td> <td>71816</td> <td>111457</td> <td>4</td> <td>712</td> <td>5,394</td> <td></td> </tr> <tr> <td>7</td> <td>71784</td> <td>110761</td> <td>10</td> <td>712</td> <td>5,394</td> <td></td> </tr> <tr> <td>8</td> <td>71783</td> <td>110748</td> <td>11</td> <td>712</td> <td>5,2142</td> <td></td> </tr> <tr> <td>9</td> <td>71782</td> <td>110670</td> <td>10</td> <td>712</td> <td>5,394</td> <td></td> </tr> </tbody> </table> </div>									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	
	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

Oracle 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

Oracle 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 file 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:

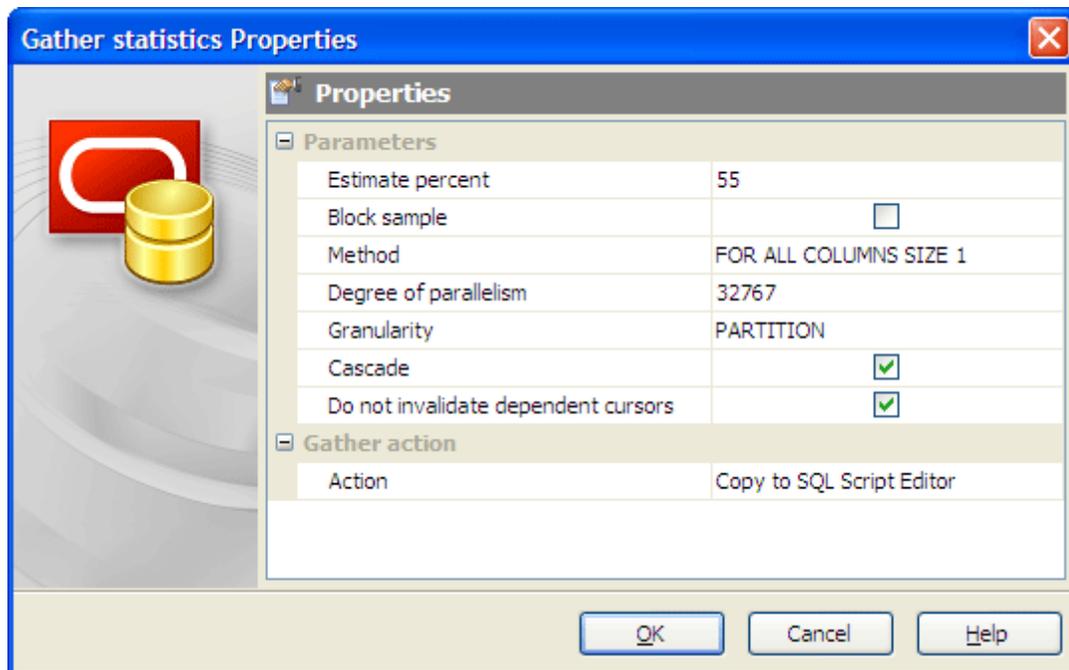
Suppose we have a table 'employee' with Non-Blob data as follows:

```
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.

### 5.2.2.3 Gathering statistic properties

Starting with version 7.10 Oracle Maestro supports gathering statistics with DBMS\_STATS package. To open [Gather statistics Properties](#) window, use the [Gather statistic](#) link on the [Navigation bar](#) of the table editor. Process parameters are described below.



#### Estimate percent

Percentage of rows to estimate (NULL means compute) The valid range is [0.000001,100]. Use the constant DBMS\_STATS.AUTO\_SAMPLE\_SIZE to have Oracle determine the appropriate sample size for good statistics. This is the default. The default value can be changed using the SET\_PARAM Procedure.

#### Block sample

Whether or not to use random block sampling instead of random row sampling. Random block sampling is more efficient, but if the data is not randomly distributed on disk, then the sample values may be somewhat correlated.

#### Method

Accepts:

- FOR ALL [INDEXED | HIDDEN] COLUMNS [size\_clause]
- FOR COLUMNS [size\_clause] column|attribute [size\_clause] [,column|attribute [size\_clause]...]

size\_clause is defined as size\_clause := SIZE {integer | REPEAT | AUTO | SKEWONLY}

- integer : Number of histogram buckets. Must be in the range [1,254].

- REPEAT : Collects histograms only on the columns that already have histograms.

- AUTO : Oracle determines the columns to collect histograms based on data distribution and the workload of the columns.

- SKEWONLY : Oracle determines the columns to collect histograms based on the data

distribution of the columns.  
The default is FOR ALL COLUMNS SIZE AUTO.

#### Degree of parallelism

The default for degree is NULL. The default value can be changed using the SET\_PARAM Procedure. NULL means use the table default value specified by the DEGREE clause in the CREATE TABLE or ALTER TABLE statement.

#### Granularity

Granularity of statistics to collect (only pertinent if the table is partitioned).

ALL gathers all (subpartition, partition, and global) statistics  
DEFAULT gathers global and partition-level statistics. This option is obsolete, and while currently supported, it is included in the documentation for legacy reasons only  
GLOBAL gathers global statistics  
PARTITION gathers partition-level statistics  
SUBPARTITION gathers subpartition-level statistics

#### Cascade

Gather statistics on the indexes for this table. Index statistics gathering is not parallelized. Using this option is equivalent to running the GATHER\_INDEX\_STATS Procedure on each of the table's indexes.

#### Do not invalidate dependent cursors

Does not invalidate the dependent cursors if set to TRUE. The procedure invalidates the dependent cursors immediately if set to FALSE.

#### Action (Execute immediately, Copy to SQL Script Editor, Create job)

After all the parameters of the statistic collection were set you can use them in different ways: execute immediately, copy to SQL Script Editor for the further work, or create job to automate the statistic gathering process.

### 5.2.3 Fields

Table columns are created and edited 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 screenshot shows the 'Field Editor' dialog box with the following details:

- Field name:** CURR\_RETIREMENT
- Field type:** NUMBER
- Status:** Computed
- Parameters:**
  - Size: 0
  - Precision: 0
  - Unlimited size:
- Field flags:**
  - Not null:
  - Primary key:
  - Autoincrement:
  - Configure autoincrement: [Configure autoincrement](#)
- Formula (11g+):** "SALARY\*.0005\*YEARS\_OF\_SERVICE"
- Default value:** (empty)
- LOB properties:** (empty)
- Comment:** The column derives the current value of the employees' retirement benefit, which is a formula based on the employee's years of service, the total

To specify the [Data Type](#), select 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 Oracle 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.

### Unlimited size

Check the option to take away size limitation of the column values.

### 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](#).

**LOB properties** (Available if BLOB, CLOB or NCLOB was selected as field type)

Use the clause to specify the LOB storage characteristics for a newly added LOB column, partition, or subpartition, or when you are converting a LONG column into a LOB column.

**For details see:** [LOB properties](#).

#### Default value

Within the box you can assign a default value for the field column. The action is optional. If the default value was specified during the new row created and no values is specified for some of the columns, the columns will be filled with their respective default values.

The **Comment** box allows you to set optional text describing the field.

### 5.2.3.1 LOB properties

#### Tablespace

The only parameter of LOB parameters you can specify for a hash partition or hash subpartition. It allows you to define storage characteristics of the field.

#### Storage in row

Specify whether the LOB value is to be stored in the row (inline) or outside of the row (out of line). The LOB locator is always stored inline regardless of where the LOB value is stored. If checked the clause specifies that the LOB value is stored inline if its length is less than approximately 4000 bytes minus system control information. Otherwise, the LOB value is stored out of line regardless of the length of the LOB value.

#### Chunk size

Specify the number of bytes to be allocated for LOB manipulation. If integer is not a multiple of the database block size, then Oracle Database rounds up (in bytes) to the next multiple. For example, if the database block size is 2048 and integer is 2050, then the database allocates 4096 bytes (2 blocks). The maximum value is 32768 (32 K), which is the largest Oracle Database block size allowed. The default size is one Oracle Database block.

#### Use retention

Use this clause to indicate that Oracle Database should retain old versions of this LOB column. You can specify the retention parameter only if the database is running in automatic undo mode.

#### Old versions space

Specify the maximum percentage of overall LOB storage space used for maintaining old versions of the LOB. The default value is 10, meaning that older versions of the LOB data are not overwritten until they consume 10% of the overall LOB storage space.

#### Free pools

Specify the number of groups of free lists for the LOB segment. Normally integer will be

the number of instances in a Real Application Clusters environment or 1 for a single-instance database. You can specify this parameter only if the database is running in automatic undo mode.

#### Cache

Use the CACHE clauses to indicate how Oracle Database should store blocks in the buffer cache.

#### Logging

The clause lets you specify whether creation of a database object will be logged in the redo log file or not.

## 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?**

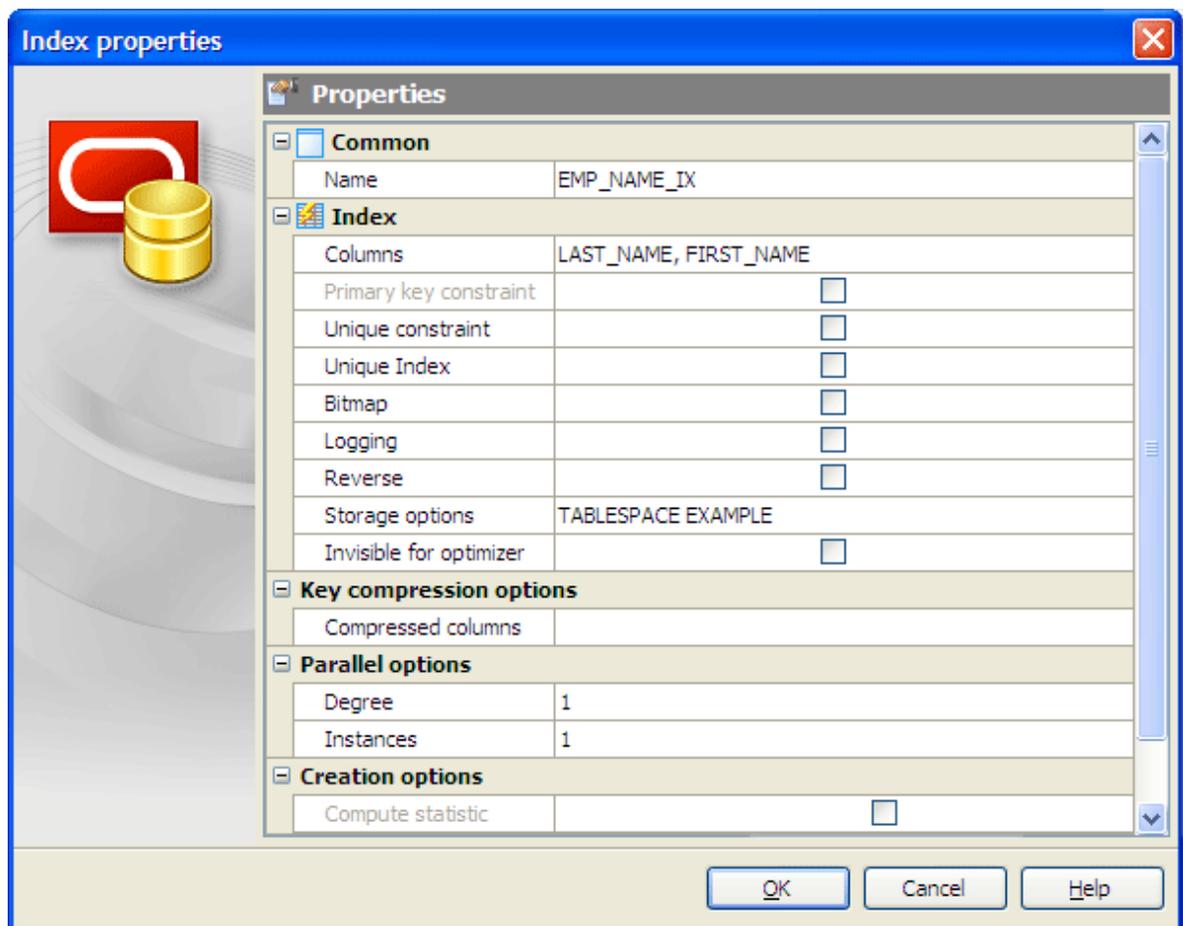
To drop the table index:

- 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 constraint**

Check the option to permit no duplicate values. A unique column must also define the NOT NULL attribute. A table can have one or more unique keys.

**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.

**Bitmap**

Use the checkbox to set storing rowids associated with a key value as a bitmap.

**Logging**

Specify whether the creation of the index will be logged (LOGGING) or not logged

(NOLOGGING) in the redo log file. This setting also determines whether subsequent Direct Loader (SQL\*Loader) and direct-path INSERT operations against the index are logged or not logged. LOGGING is the default.

#### Reverse

Specify the clause to store the bytes of the index block in reverse order, excluding the rowid.

You can also define [Storage options](#) here.

#### Compressed columns

Specify compression value to enable key compression, which eliminates repeated occurrence of key column values and may substantially reduce storage. Use integer to specify the prefix length (number of prefix columns to compress).

Use the [Parallel options](#) to parallelize creation of the domain index. For a nonpartitioned domain index, Oracle Database passes the explicit or default degree of parallelism to the ODCIIndexCreate cartridge routine, which in turn establishes parallelism for the index.

## 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** from the respective drop-down list.

- **NO ACTION** Produce an error indicating that the deletion or update will create a foreign key constraint violation. If the constraint is deferred this error will be produced at constraint check time if there still exist any referencing rows. This is the default action.
- **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.

**Enabled**

When checked, foreign key is enforced.

**Validated**

The behavior of the clause always depends on whether the constraint is enabled or disabled, either explicitly or by default.

- Enabled validated key specifies that all old and new data also complies with the constraint. An enabled validated constraint guarantees that all data is and will continue to be valid.
- Enabled novalidated key ensures that all new DML operations on the constrained data comply with the constraint. This clause does not ensure that existing data in the table complies with the constraint and therefore does not require a table lock.
- Disabled validated key disables the constraint and drops the index on the

constraint, but keeps the constraint valid. This feature is most useful in data warehousing situations, because it lets you load large amounts of data while also saving space by not having an index.

- Disabled novalidated key signifies that Oracle makes no effort to maintain the constraint (because it is disabled) and cannot guarantee that the constraint is true (because it is not being validated).

#### Deffered

Check the option to indicate that Oracle should check this constraint at the end of subsequent transactions. Otherwise Oracle should check this constraint at the end of each subsequent SQL statement.

## 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.

The [Check Properties](#) editor allows you to add a new check constraint or edit an existing one. This dialog can be invoked from [Table Editor](#) , or via the popup menu of the corresponding nodes of the explorer tree.

### How can I add a new check?

Checks are created within [Check Properties](#) . 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](#)  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?

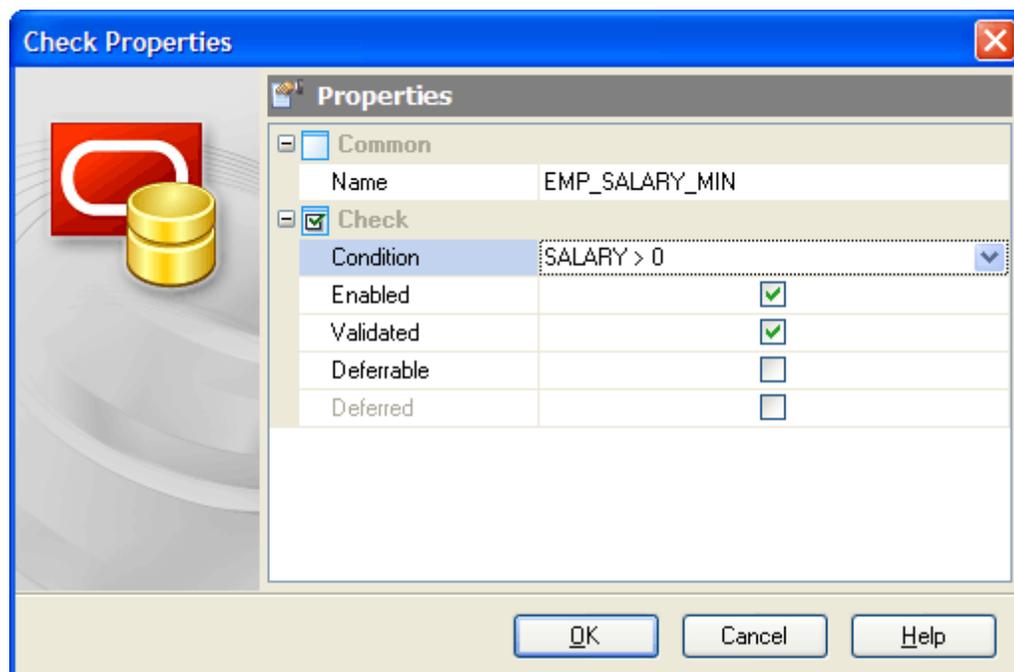
To drop the check:

- 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**

The name of the check constraint.

### 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.

#### Enabled

When checked, the check is enforced.

#### Validated

The behavior of the clause always depends on whether the constraint is enabled or disabled, either explicitly or by default.

- Enabled validated constraint specifies that all old and new data also complies with the constraint. An enabled validated constraint guarantees that all data is and will continue to be valid.
- Enabled novalidated constraint ensures that all new DML operations on the constrained data comply with the constraint. This clause does not ensure that existing data in the table complies with the constraint and therefore does not require a table lock.
- Disabled validated constraint disables the constraint and drops the index on the constraint, but keeps the constraint valid. This feature is most useful in data warehousing situations, because it lets you load large amounts of data while also saving space by not having an index.
- Disabled novalidated constraint signifies that Oracle makes no effort to maintain the constraint (because it is disabled) and cannot guarantee that the constraint is true (because it is not being validated).

#### Defferable

The option indicates whether or not, in subsequent transactions, constraint checking can be deferred until the end of the transaction using the SET CONSTRAINT(S) statement.

#### Deffered

Check the option to indicate that Oracle should check this constraint at the end of subsequent transactions. Otherwise Oracle should check this constraint at the end of each subsequent SQL statement.

## 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>88</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>[90]</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 Oracle Maestro are explained in a

[separate topic](#)<sup>37</sup>. Below you will find a description of wizard steps that are unique for the current object.

### Specifying trigger properties

To define a new trigger, you need to set its [Name](#), .

#### Owner

Here the trigger creator is displayed.

#### Enabled

Check the option to enforce the trigger.

#### When fire (Before, After, Instead of)

Specifies the table operation that causes the trigger to fire.

#### When condition

Specify the trigger condition, which is a SQL condition that must be satisfied for the database to fire the trigger.

#### Nested table columns

Specify the column of a view upon which the trigger is being defined. Such a trigger will fire only if the DML operates on the elements of the nested table.

#### For each row

Specify the option to designate the trigger as a row trigger. Oracle Database fires a row trigger once for each row that is affected by the triggering statement and meets the optional trigger constraint defined in the WHEN condition.

#### Trigger events

Check necessary boxes to specify DML statements that can cause the trigger to fire. Oracle Database fires the trigger in the existing user transaction.

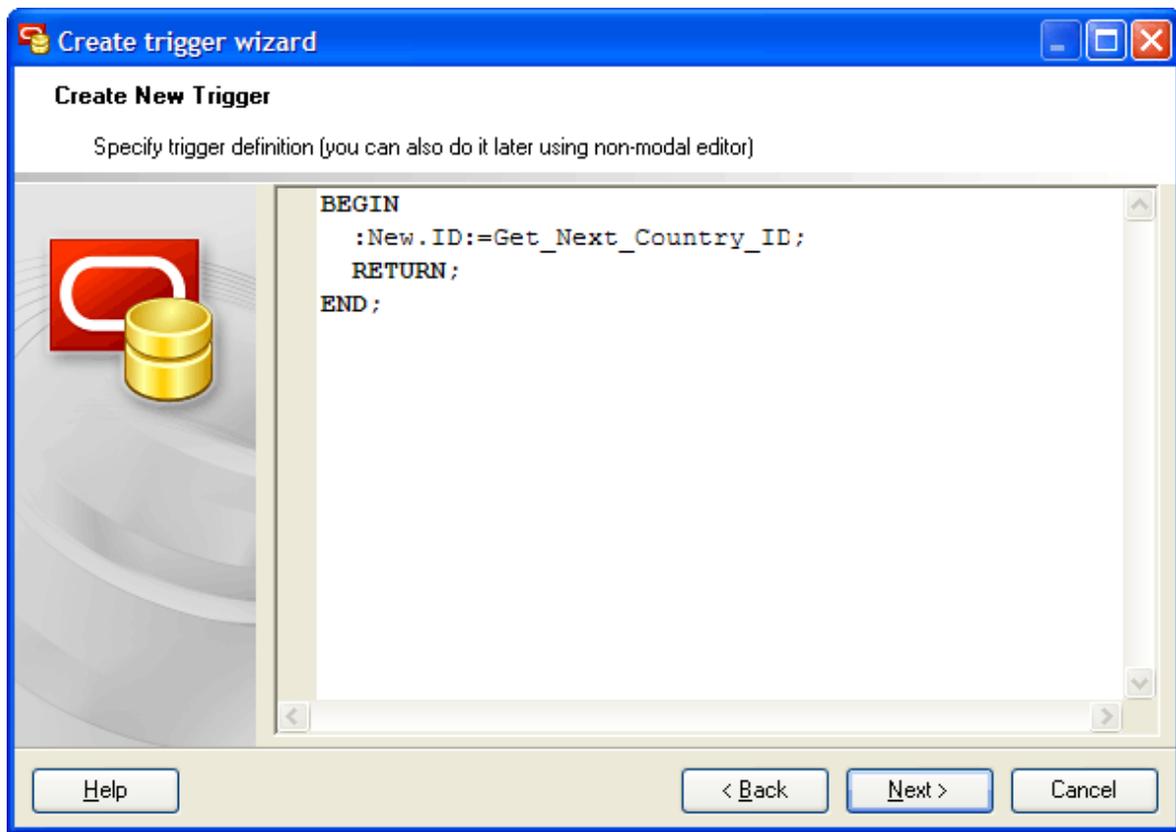
#### Update columns (Available if the trigger fires on update)

Specify the columns, the database to fire the trigger whenever an UPDATE statement changes a value in one of the columns.

Use [Referencing properties](#) to set the correlation names in the PL/SQL block and WHEN condition of a row trigger to refer specifically to respective (new, old, parent) value of the current row.

### Specifying trigger definition

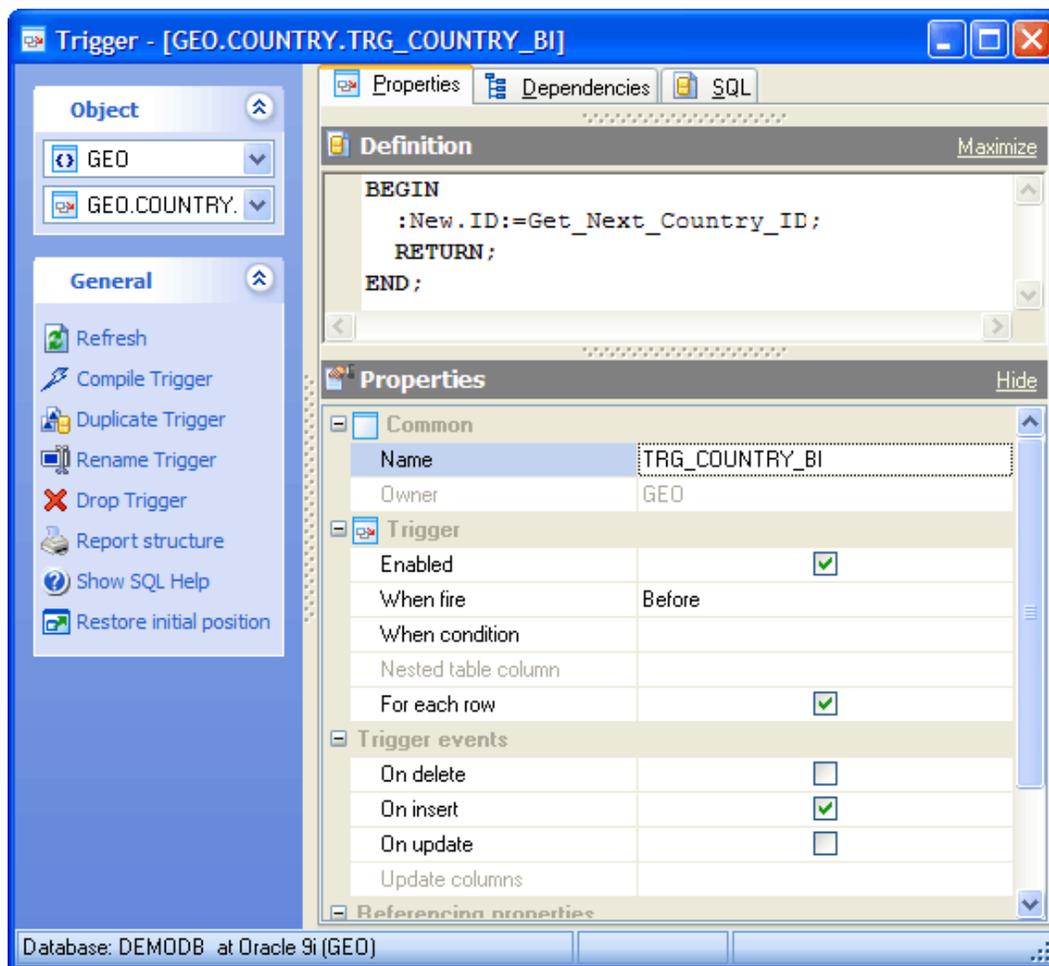
Here you can specify the trigger [definition](#). Specify the trigger steps to be executed when the trigger fires. The step is optional: you can do it later using a non-modal editor.



### 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 Oracle Maestro are explained in a [separate topic](#) [3]. Below you will find a description of editor tabs that are unique for the current object.



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

The **Errors** tab displays all the necessary information about object errors. If an error has occurred during the object compilation it appears in the list with some additional properties: **Order** (one after another), **Line** and **Position** (object definition location the error was found out), **Error** (corresponding PL/SQL exception).

### 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 Oracle server.

**Enabled**

Check the option to enforce the trigger.

#### When fire (Before, After, Instead of)

Specifies the table operation that causes the trigger to fire.

#### When condition

Specify the trigger condition, which is a SQL condition that must be satisfied for the database to fire the trigger.

#### Nested table columns

The field represents the column of a view upon which the trigger is being defined. Such a trigger will fire only if the DML operates on the elements of the nested table.

#### For each row

Specify the option to designate the trigger as a row trigger. Oracle Database fires a row trigger once for each row that is affected by the triggering statement and meets the optional trigger constraint defined in the WHEN condition.

#### Trigger events

Check necessary boxes to specify DML statements that can cause the trigger to fire. Oracle Database fires the trigger in the existing user transaction.

#### Update columns (Available if the trigger fires on update)

Specify the columns, the database to fire the trigger whenever an UPDATE statement changes a value in one of the columns.

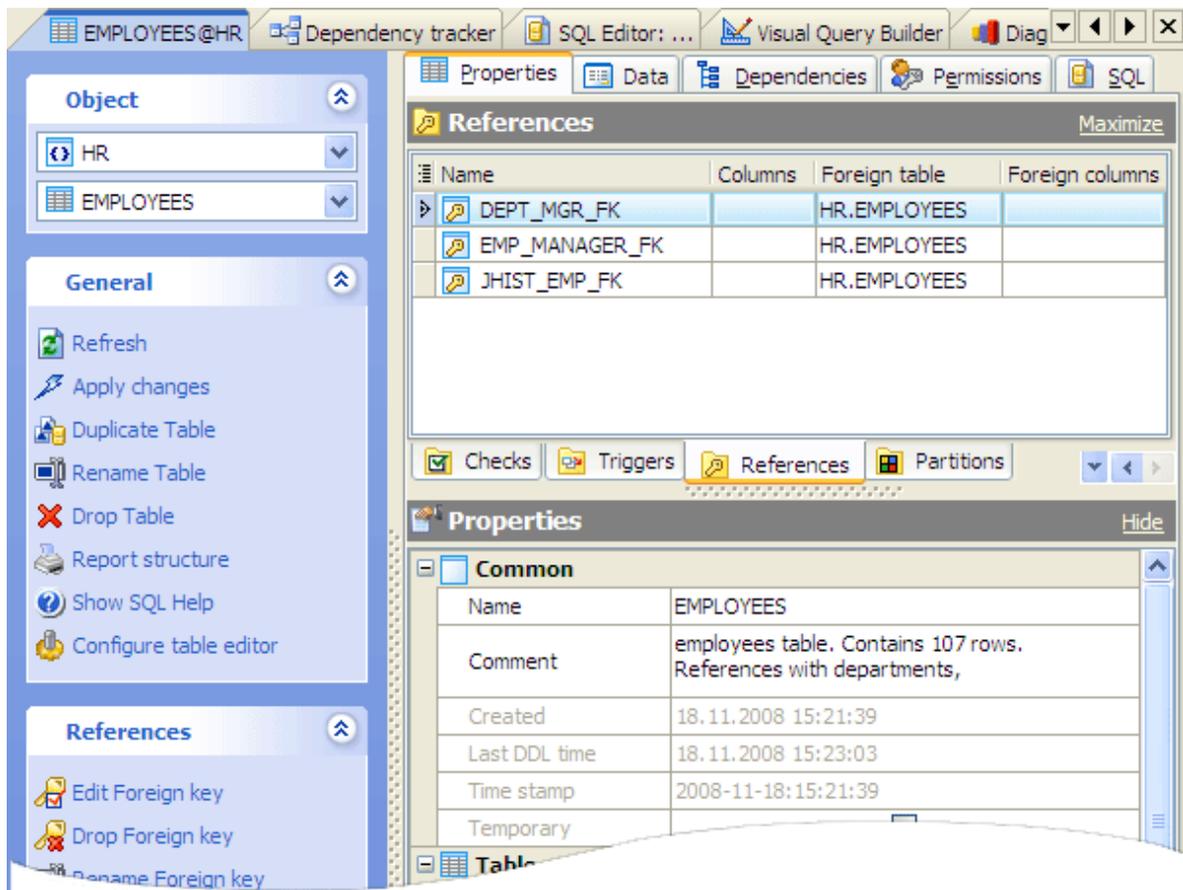
Use [Referencing properties](#) to set the correlation names in the PL/SQL block and WHEN condition of a row trigger to refer specifically to respective (new, old, parent) value of the current row.

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>[96]</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>[107]</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.

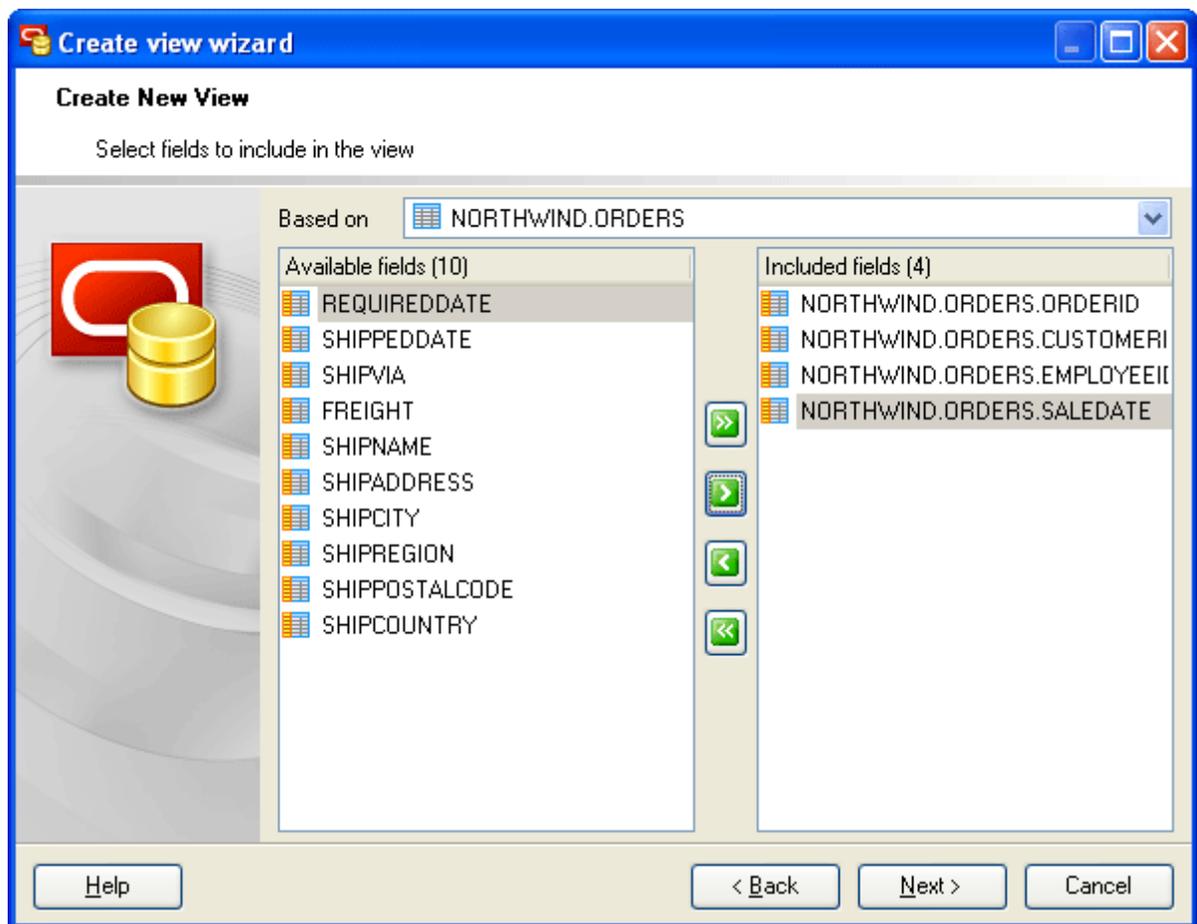
### 5.3.1 **Create View Wizard**

[Create View Wizard](#) guides you through the process of creating a new view. See [How To Create View](#)<sup>[36]</sup> to learn how to run this wizard.

The basic principles of Create Object Wizards in Oracle Maestro are explained in a [separate topic](#)<sup>[37]</sup>. Below you will find a description of wizard steps that are unique for 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.

#### Comment

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

#### Force view

Specify the option if you want to create the view regardless of whether the base tables of the view or the referenced object types exist or the owner of the schema containing the view has privileges on them. These conditions must be true before any SELECT, INSERT, UPDATE, or DELETE statements can be issued against the view.

#### With Check Option

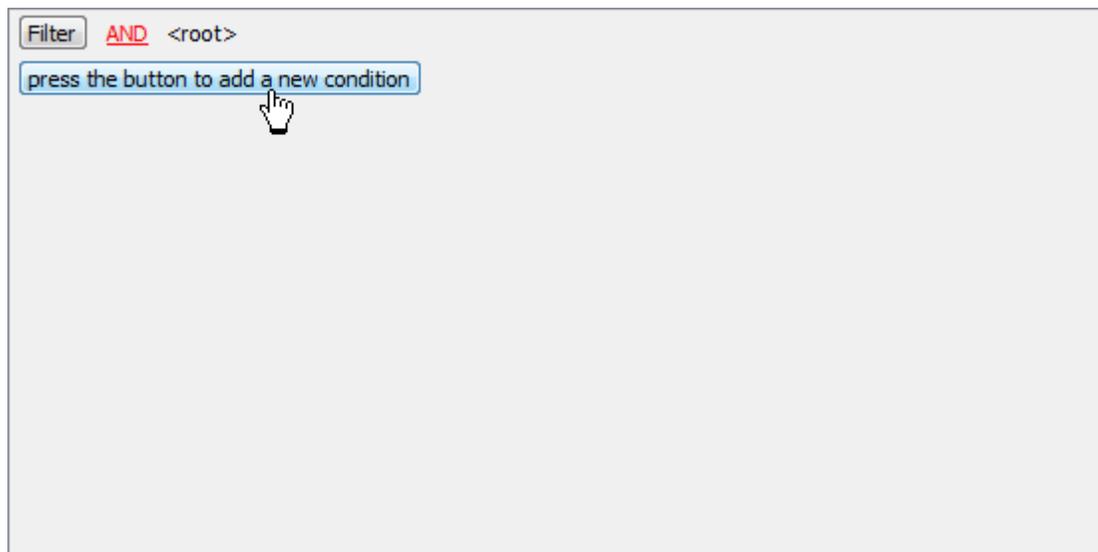
Forces all data modification statements executed against the view to follow the criteria set within the select statement. When a row is modified through a view, the **With Check** option ensures the data remaining visible through the view after the modification is committed. To check the option the owner of the view table must have been granted the INSERT, UPDATE, or DELETE privilege for the view table.

### Specifying the WHERE condition

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

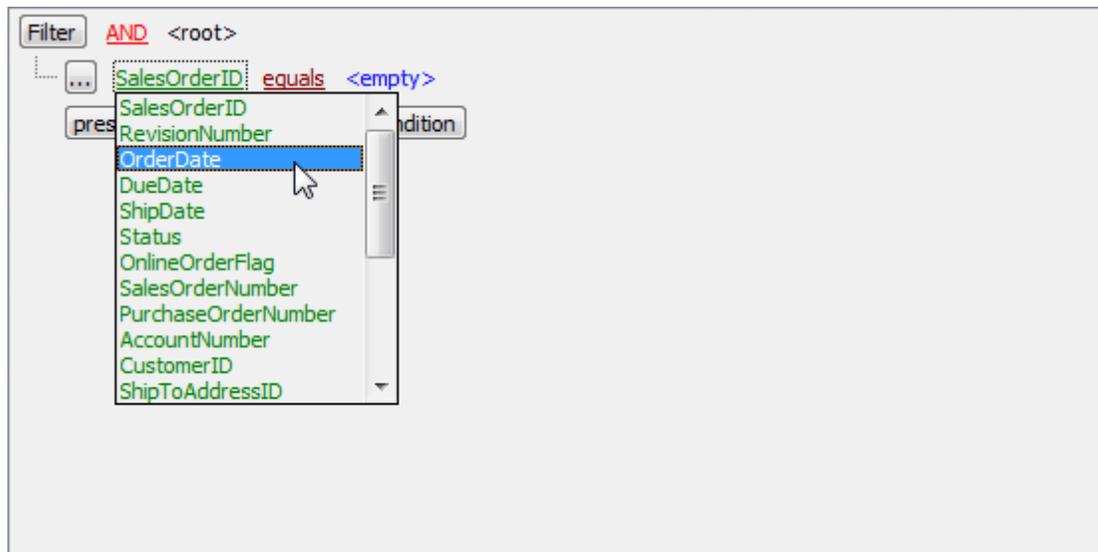
#### ■ Adding a new condition to the filter

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.



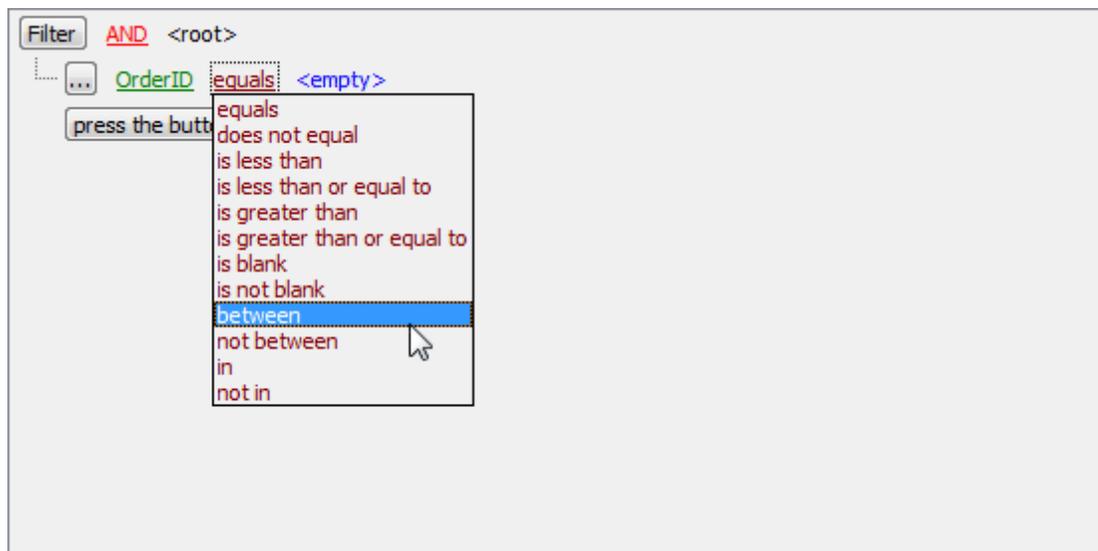
#### ■ Setting a filter criteria in the condition

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



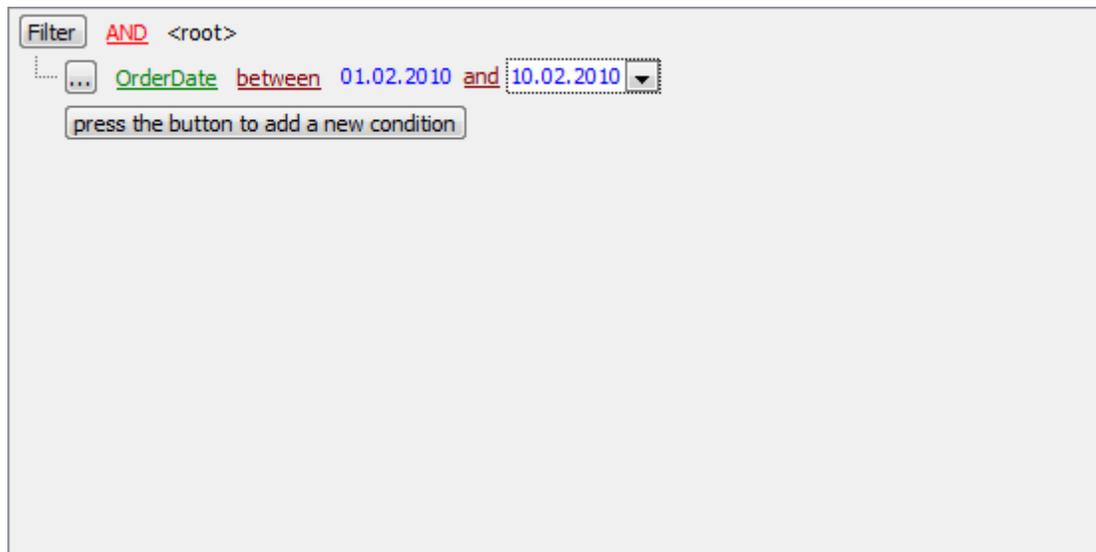
#### ■ Setting an operator in the condition

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.

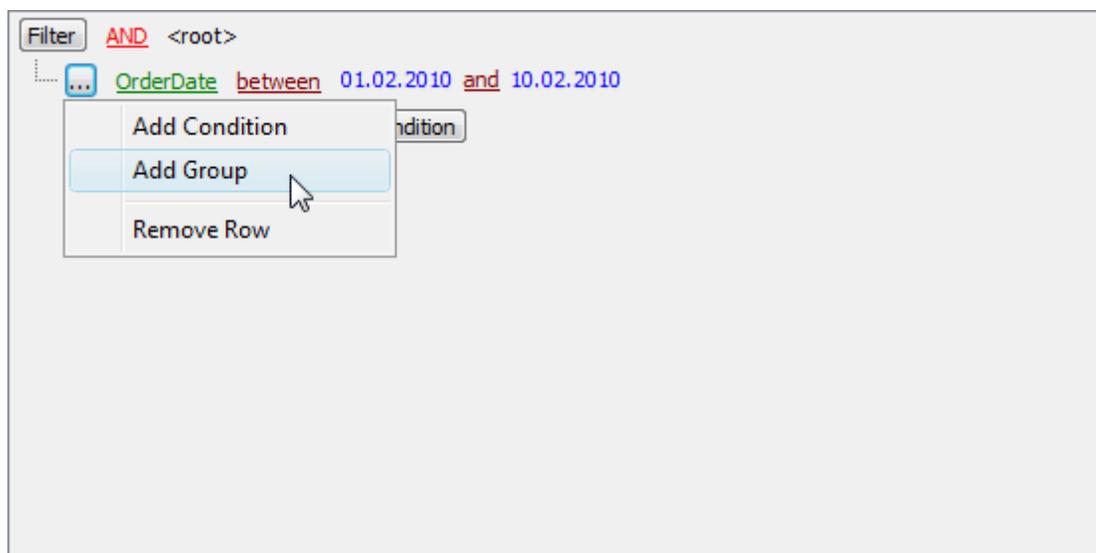


Now use the [Apply](#) button to see the filter result.

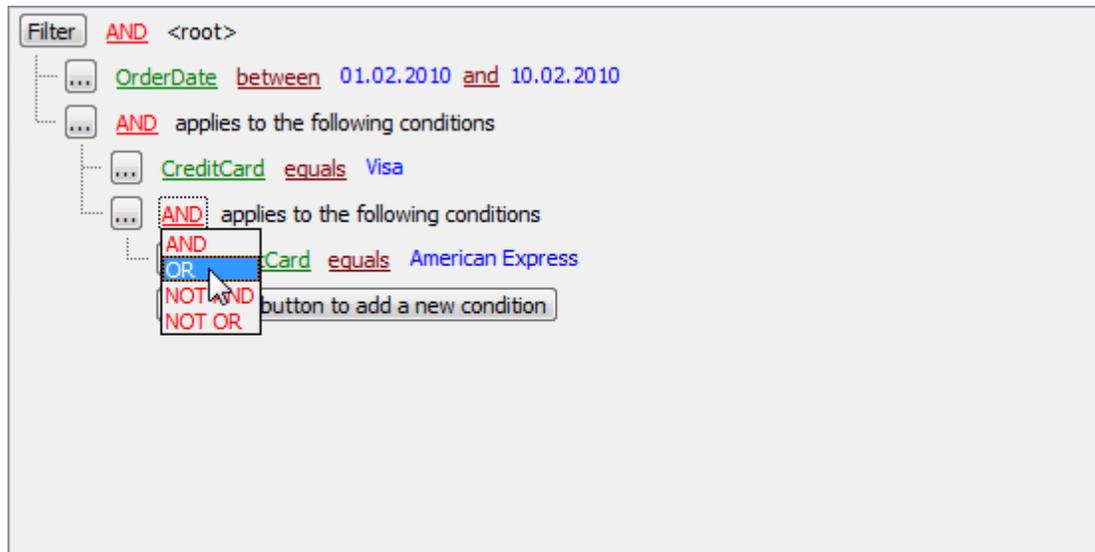
You can add additional conditions to the same root level to be combined by the AND operator.

#### ■ Adding a new group

Suppose we need to select orders made between 01.02.2010 and 10.02.2010 and paid 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.



### Adding view subitems

On this step of the wizard you can specify triggers of the new view.  
To add a new object:

- Press **Insert** or use pop-up menu to open the [Create Trigger Wizard](#)<sup>[88]</sup>;
- Specify new object properties.

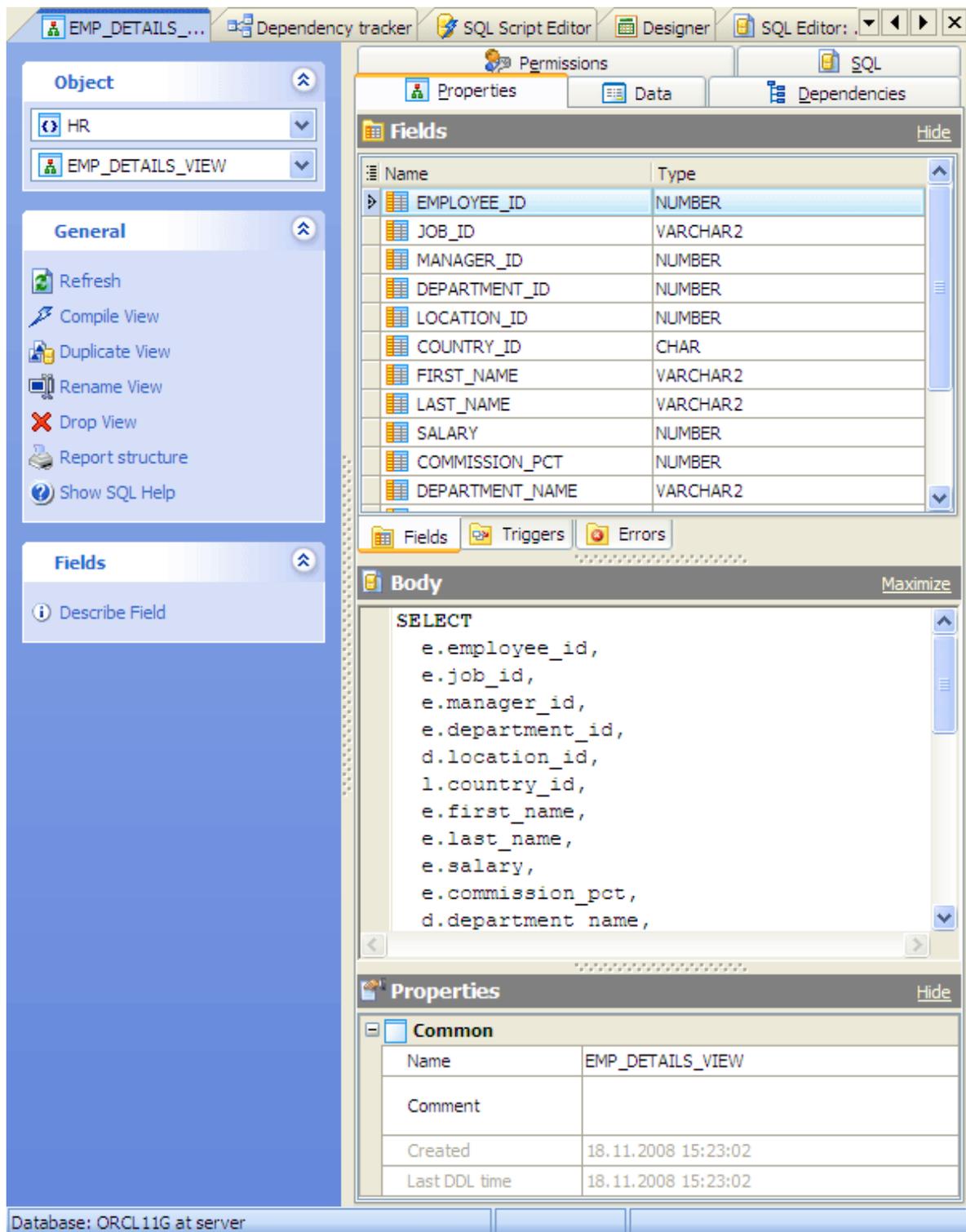
## 5.3.2 View Editor

**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 Oracle Maestro are explained in a [separate topic](#)<sup>[39]</sup>. Below you will find a description of editor tabs that are unique for the current object.

- [Editing view properties](#)<sup>[102]</sup>
- [Viewing data](#)<sup>[103]</sup>

**See also:** [Create View Wizard](#)<sup>[96]</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, the view owner and the comment for the view.

### 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](#), [Triggers](#)

The **Errors** tab displays all the necessary information about object errors. If an error have occurred during the object compilation it appears in the list with some additional properties: **Order** (one after another), **Line** and **Position** (object definition location the error was found out), **Error** (corresponding PL/SQL exception).

### 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 Oracle server.

### Created

The field displays the date the object was created.

### Last DDL time

Use the field to find the date when the last data definition language (DDL) operation was performed on the current object. The Last DDL time can help you to find if any changes to the object definitions have been made on or after a specific time.

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.

The screenshot displays the SQL Developer interface with the EMP\_DETAILS\_VIEW table selected. The left sidebar shows the Object Explorer with 'HR' and 'EMP\_DETAILS\_VIEW' selected, and the General tab active. The main window shows the table data with columns EMPLOYEE\_ID, JOB\_ID, FIRST\_NAME, LAST\_NAME, and MA. The data is grouped by DEPARTMENT\_ID (60, 70, 80). The status bar at the bottom indicates 'Records fetched: 106/106' and 'Database: ORCL11G at server'.

DEPARTMENT_ID	EMPLOYEE_ID	JOB_ID	FIRST_NAME	LAST_NAME	MA
60	103	IT_PROG	Alexander	Hunold	
60	104	IT_PROG	Bruce	Ernst	
60	105	IT_PROG	David	Austin	
60	106	IT_PROG	Valli	Pataballa	
60	107	IT_PROG	Diana	Lorentz	
70	204	PR_REP	Hermann	Baer	
80	145	SA_MAN	John	Russell	
80	146	SA_MAN	Karen	Partners	
80	147	SA_MAN	Alberto	Errazuriz	
80	148	SA_MAN	Gerald	Cambrault	
80	149	SA_MAN	Eleni	Zlotkey	
80	150	SA_REP	Peter	Tucker	
80	151	SA_REP	David	Bernstein	
80	152	SA_REP	Peter	Hall	
80	153	SA_REP	Christopher	Olsen	
80	154	SA_REP	Nanette	Cambrault	
80	155	SA_REP	Oliver	Tuvault	
80	156	SA_REP	Janette	King	
80	157	SA_REP	Patrick	Sully	
80	158	SA_REP	Allan	McEwen	
80	159	SA_REP	Lindsey	Smith	
80	160	SA_REP	Louise	Doran	
80	161	SA_REP	Sarath	Sewall	
80	162	SA_REP	Clara	Vishney	
80	163	SA_REP	Danielle	Greene	
80	164	SA_REP	Mattea	Marvins	
80	165	SA_REP	David	Lee	

## 5.4 Materialized Views

A materialized view is a database object that contains the results of a query. The FROM clause of the query can name tables, views, and other materialized views. Collectively these objects are called master tables (a replication term) or detail tables (a data warehousing term). This reference uses "master tables" for consistency. The databases containing the master tables are called the master databases.

### ■ How can I create a new materialized view?

New materialized views are created within [Create Materialized view Wizard](#)<sup>[106]</sup>. In order to run the wizard you should either

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

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

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

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

Materialized views can be edited within [Materialized view Editor](#). In order to run the editor you should either

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

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

- select the materialized view to rename in the explorer tree;
- select the [Rename Materialized view](#) item from the popup menu

or

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

#### ■ **How can I drop a materialized view?**

To drop a materialized view:

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

or

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

and confirm dropping in the dialog window.

### 5.4.1 Create Materialized Views Wizard

[Create Materialized View Wizard](#) guides you through the process of creating a new view. See [How To Create Materialized View](#) <sup>[105]</sup> to learn how to run this wizard.

The basic principles of Create Object Wizards in Oracle Maestro are explained in a [separate topic](#) <sup>[37]</sup>. Below you will find a description of wizard steps that are unique for the current object.

#### **Selecting fields**

To specify fields to be used in the view, select a table or a view from the [Based on](#) drop-down menu and 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. If column names are not provided, they are taken from the output column names of the query.

#### **Specifying options**

Use this step to set [Owner](#), and [Comment](#) for the new materialized view. If not specified, default values are consulted.

#### **Build mode (IMMEDIATE, DEFERRED)**

Specify [IMMEDIATE](#) to indicate that the materialized view is to be populated immediately. Otherwise, the materialized view is to be populated by the next refresh operation. The first (deferred) refresh must always be a complete refresh.

**Can be refreshed**

Use the option to permit the materialized view refresh.

**Refresh method (FORCE, FAST, COMPLETE, NEVER)**

Specify **FORCE** to indicate that when a refresh occurs, Oracle Database will perform a fast refresh if one is possible or a complete refresh otherwise. If you do not specify a refresh method (**FAST**, **COMPLETE**, or **FORCE**), then **FORCE** is the default. Specify **FAST** to indicate the incremental refresh method, which performs the refresh according to the changes that have occurred to the master tables. Specify **COMPLETE** to indicate the complete refresh method, which is implemented by executing the defining query of the materialized view. If you request a complete refresh, then Oracle Database performs a complete refresh even if a fast refresh is possible.

**Refresh mode (ON DEMAND, ON COMMIT, NEVER)**

Specify **ON DEMAND** to indicate that the materialized view will be refreshed on demand by calling one of the three DBMS\_MVIEW refresh procedures. Specify **ON COMMIT** to indicate that a fast refresh is to occur whenever the database commits a transaction that operates on a master table of the materialized view. This clause may increase the time taken to complete the commit, because the database performs the refresh operation as part of the commit process. The clause is not supported for materialized views containing object types.

**With (PRIMARY KEY, ROWID)**

Specify **PRIMARY KEY** to create a primary key materialized view. This is the default and should be used in all cases except those described for **ROWID**. Primary key materialized views allow materialized view master tables to be reorganized without affecting the eligibility of the materialized view for fast refresh. The master table must contain an enabled primary key constraint. Specify **ROWID** to create a rowid materialized view. Rowid materialized views are useful if the materialized view does not include all primary key columns of the master tables.

### **Specifying WHERE condition**

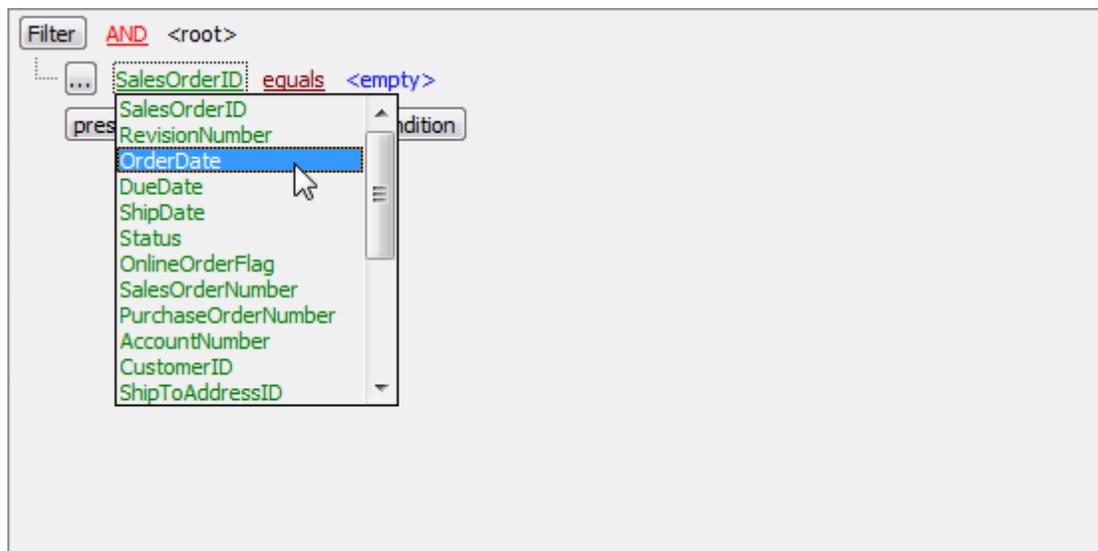
**■ Adding a new condition to the filter**

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.



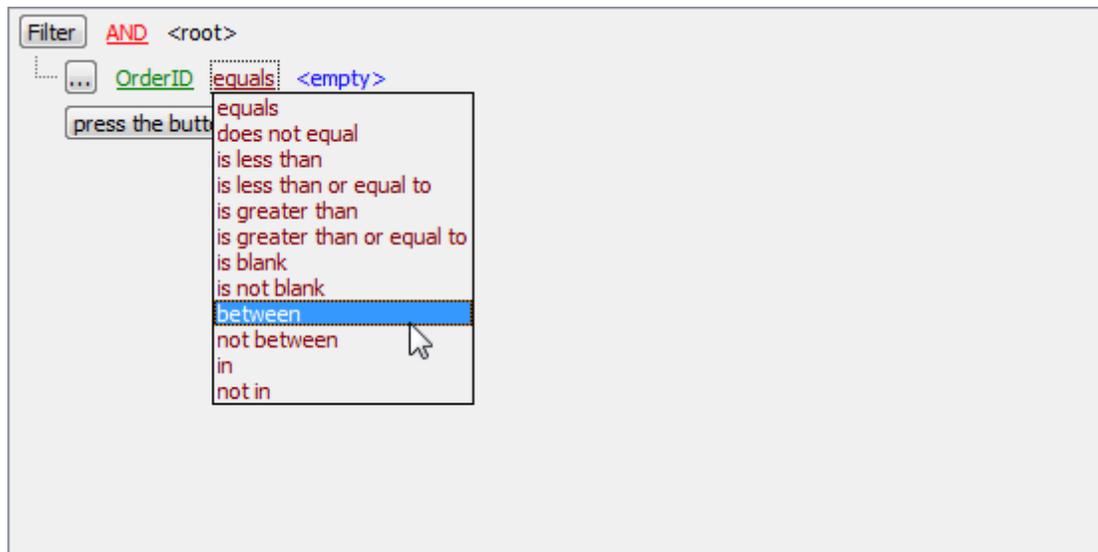
#### ■ Setting a filter criteria in the condition

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.



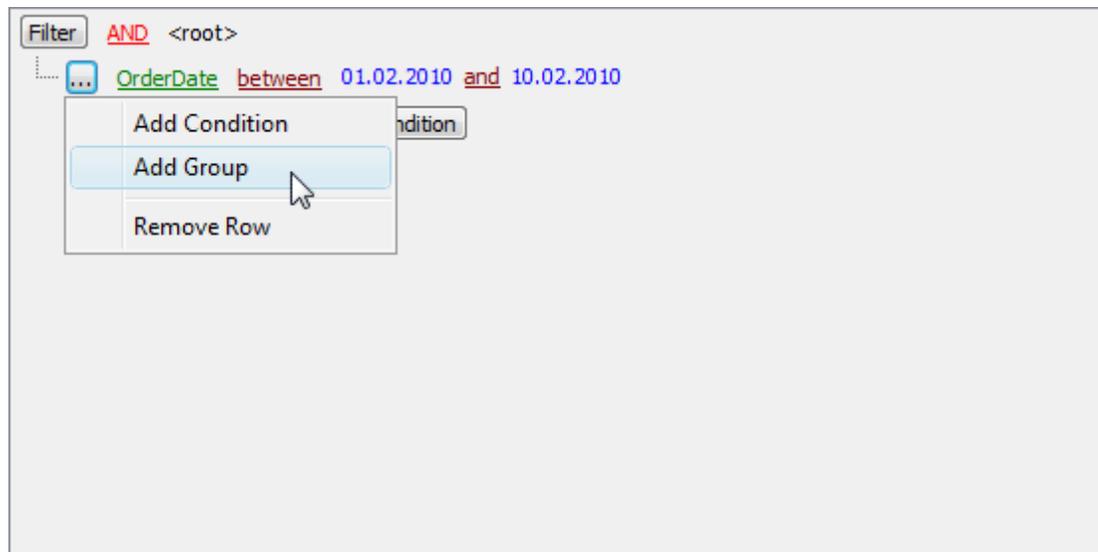
Now use the [Apply](#) button to see the filter result.

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

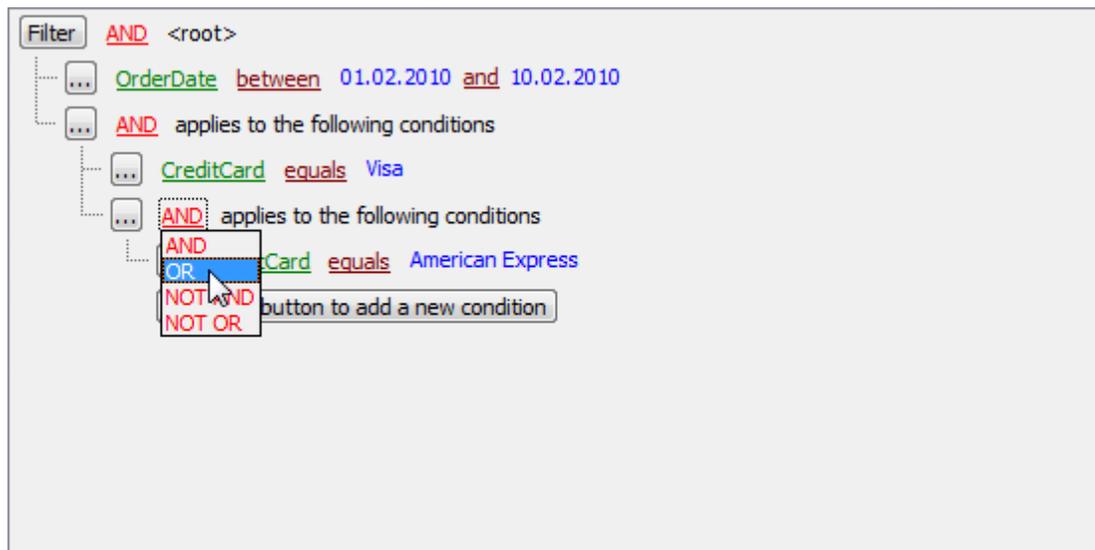
#### ■ Adding a new group

Suppose we need to select orders made between 01.02.2010 and 10.02.2010 and paid 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.



**See also:** [Materialized View Editor](#)<sup>[110]</sup>

## 5.4.2 Materialized Views Editor

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

The basic principles of Object Editors in Oracle Maestro are explained in a [separate topic](#)<sup>[39]</sup>. Below you will find a description of editor tabs that are unique for the current object.

The [Properties](#) tab allows you to rename the view, change the definition, the owner and the comment of the view.

The [Errors](#) tab displays all the necessary information about object errors. If an error have occurred during the object compilation it appears in the list with some additional properties: [Order](#) (one after another), [Line](#) and [Position](#) (object definition location the error was found out), [Error](#) (corresponding PL/SQL exception).

The [Fields](#) tab represents fields included in the materialized view. Use grid's popup menu to describe or rename fields.

The [Body](#) area contains the query used to populate the view. To change the query, modify the SQL statement and use the [Compile materialized view](#) at the [Navigation bar](#).

#### [Created](#)

The field displays the date the object was created.

#### [Last DDL time](#)

Use the field to find the date when the last data definition language (DDL) operation was performed on the current object. The Last DDL time can help you to find if any changes to the object definitions have been made on or after a specific time.

#### [Build mode \(IMMEDIATE, DEFFERED\)](#)

Specify [IMMEDIATE](#) to indicate that the materialized view is to be populated immediately. Otherwise, the materialized view is to be populated by the next refresh operation. The first (deferred) refresh must always be a complete refresh.

#### [Can be refreshed](#)

Use the option to permit the materialized view refresh.

#### [Refresh method \(FORCE, FAST, COMPLETE, NEVER\)](#)

Specify [FORCE](#) to indicate that when a refresh occurs, Oracle Database will perform a fast refresh if one is possible or a complete refresh otherwise. If you do not specify a refresh method ([FAST](#), [COMPLETE](#), or [FORCE](#)), then [FORCE](#) is the default. Specify [FAST](#) to indicate the incremental refresh method, which performs the refresh according to the changes that have occurred to the master tables. Specify [COMPLETE](#) to indicate the complete refresh method, which is implemented by executing the defining query of the materialized view. If you request a complete refresh, then Oracle Database performs a complete refresh even if a fast refresh is possible.

#### [Refresh mode \(ON DEMAND, ON COMMIT, NEVER\)](#)

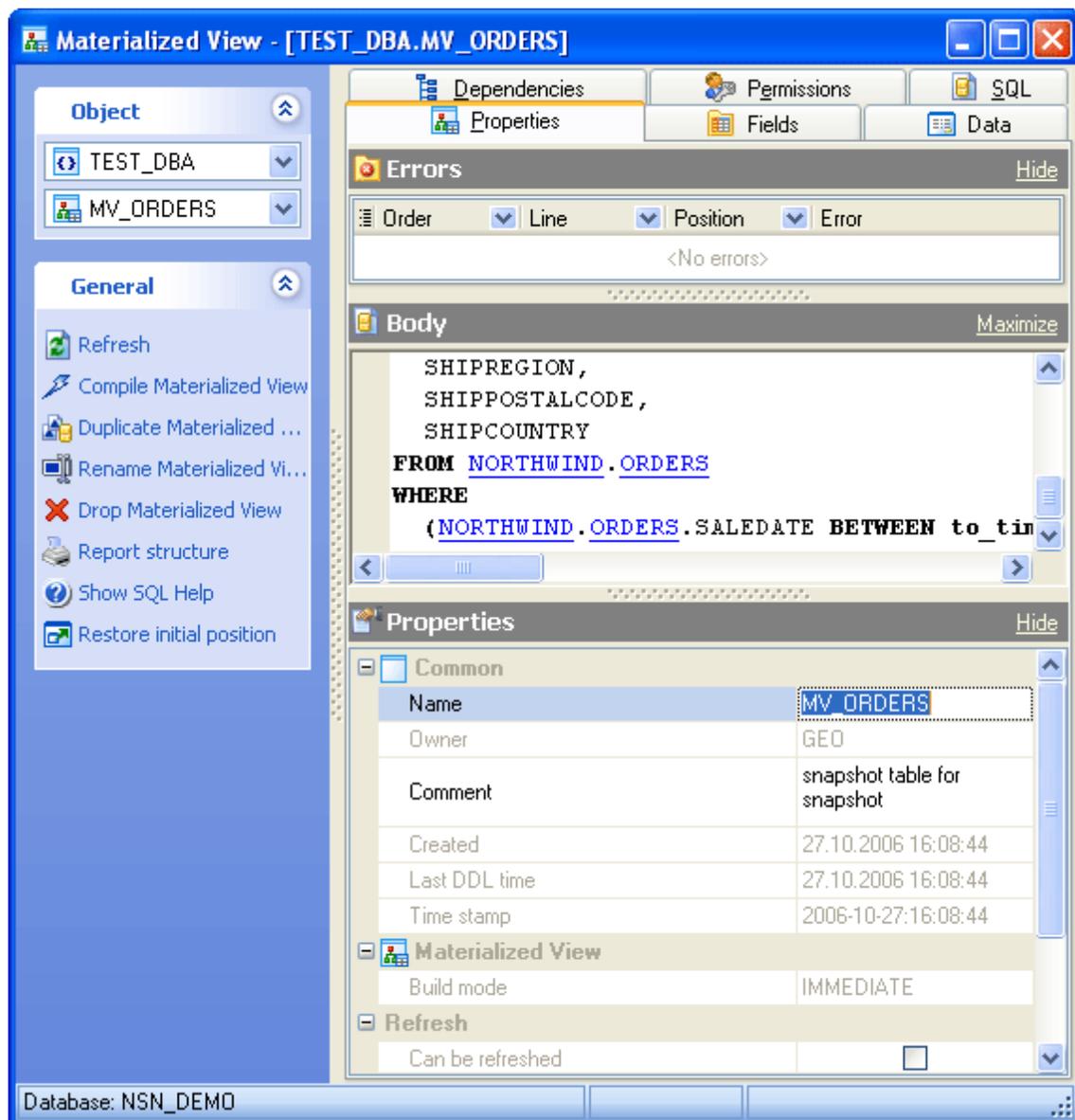
Specify [ON DEMAND](#) to indicate that the materialized view will be refreshed on demand by calling one of the three DBMS\_MVIEW refresh procedures. Specify [ON COMMIT](#) to indicate that a fast refresh is to occur whenever the database commits a transaction that operates on a master table of the materialized view. This clause may increase the time taken to complete the commit, because the database performs the refresh operation as part of the commit process. The clause is not supported for materialized views containing object types.

#### [With \(PRIMARY KEY, ROWID\)](#)

Specify [PRIMARY KEY](#) to create a primary key materialized view. This is the default and should be used in all cases except those described for [ROWID](#). Primary key materialized views allow materialized view master tables to be reorganized without affecting the

eligibility of the materialized view for fast refresh. The master table must contain an enabled primary key constraint. Specify `ROWID` to create a rowid materialized view. Rowid materialized views are useful if the materialized view does not include all primary key columns of the master tables.

The **Data** tab displays current view data represented as a grid (see [Data View](#)<sup>[279]</sup> 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](#)<sup>[290]</sup> to view and edit the BLOB fields.



**See also:** [Create Materialized View Wizard](#)<sup>[106]</sup>

## 5.5 Procedures

A procedure is a group of PL/SQL statements that you can call by name. A call specification (sometimes called call spec) declares a Java method or a third-generation language (3GL) routine so that it can be called from SQL and PL/SQL. The call spec tells Oracle Database which Java method to invoke when a call is made. It also tells the database what type conversions to make for the arguments and return value.

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

New procedures are created within [Create Procedure Wizard](#)<sup>[114]</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>[118]</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.

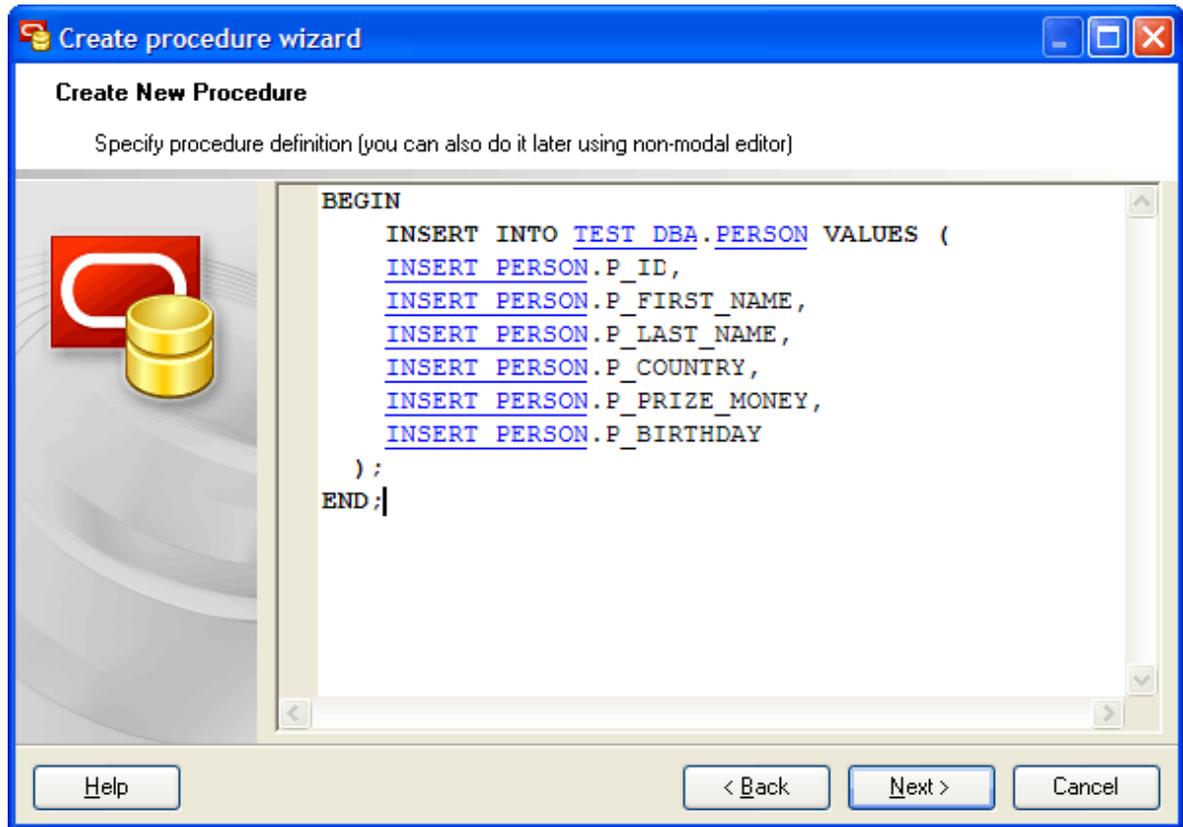
## 5.5.1 Create Procedure Wizard

[Create Procedure Wizard](#) guides you through the process of creating a new procedure. See [How To Create Procedure](#)<sup>[113]</sup> for instructions on running this wizard.

The basic principles of Create Object Wizards in Oracle Maestro are explained in a [separate topic](#)<sup>[37]</sup>. Below you will find a description of wizard steps that are unique for the current object.

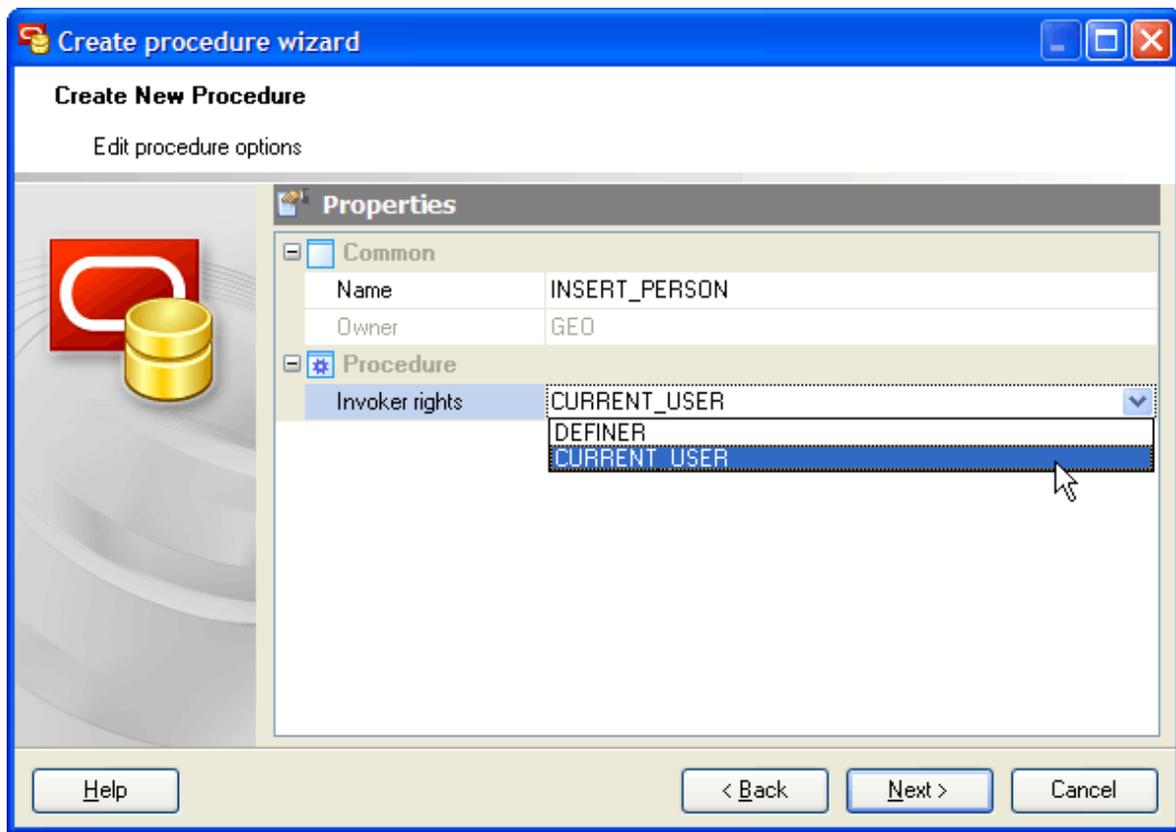
- [Specifying procedure options](#)<sup>[115]</sup>
- [Managing parameters of a new procedure](#)<sup>[116]</sup>
- [Specifying procedure definition](#)<sup>[117]</sup>

**See also:** [Procedure Editor](#)<sup>[118]</sup>



#### 5.5.1.1 Specifying procedure options

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



#### Name

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

#### Owner

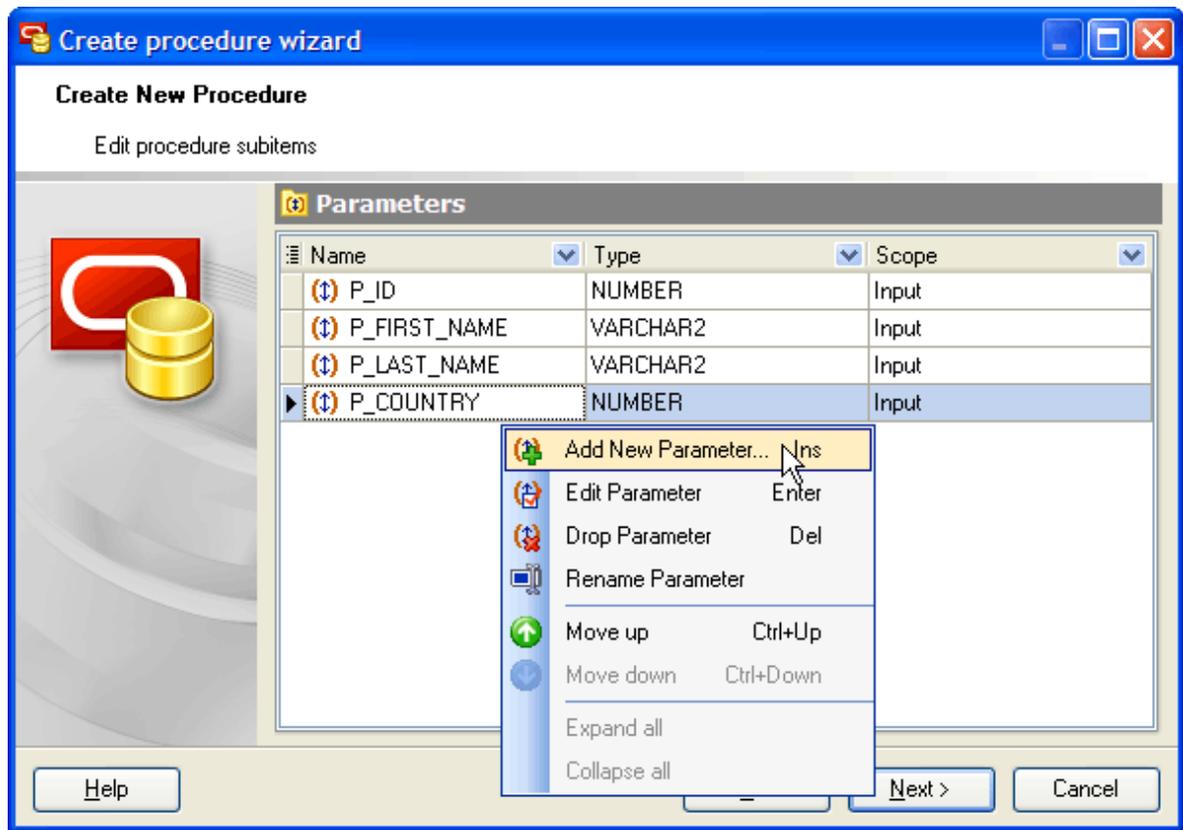
Defines the owner for the procedure.

#### Invoker rights (DEFINER, CURRENT USER)

Specify **DEFINER** to indicate that the procedure executes with the privileges of the owner of the schema in which the procedure resides, and that external names resolve in the schema where the procedure resides. Specify **CURRENT USER** to indicate that the procedure executes with the privileges of **CURRENT USER**. This clause also specifies that external names in queries, DML operations, and dynamic SQL statements resolve in the schema of **CURRENT USER**. External names in all other statements resolve in the schema in which the procedure resides.

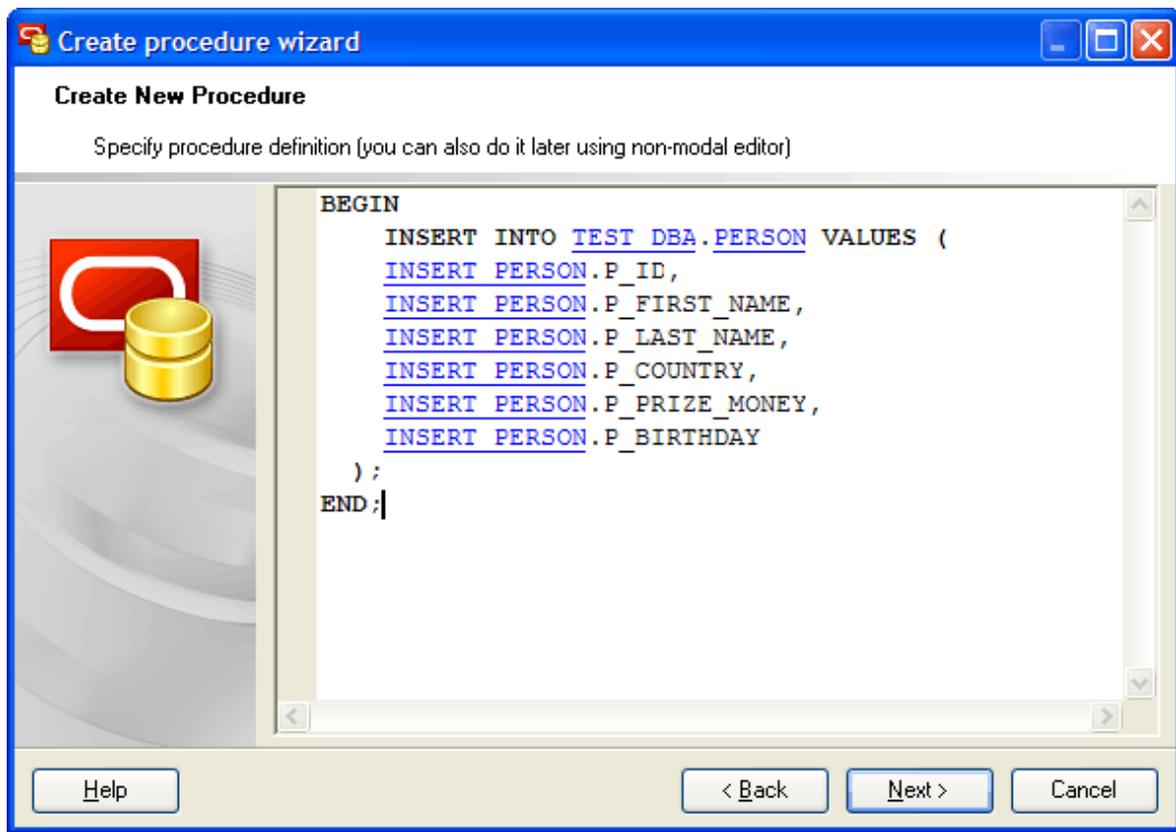
#### 5.5.1.2 Managing parameters

Use the pop-up menu or press **Insert** to add a new parameter and set its properties in [Parameter Editor](#)<sup>44</sup>. Press **Enter** or use the appropriate pop-up menu item to edit the selected parameter, or the **Delete** to delete one.



### 5.5.1.3 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.



## 5.5.2 Procedure Editor

[Procedure Editor](#) allows you to execute the existing procedures or edit their 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>[113]</sup> for details).

The basic principles of Object Editors in Oracle Maestro are explained in a [separate topic](#)<sup>[39]</sup>. Below you will find a description of editor tabs that are unique for the current object.

- [Editing procedure properties](#)<sup>[119]</sup>
- [Viewing procedure results](#)<sup>[120]</sup>

**See also:** [Create Procedure Wizard](#)<sup>[114]</sup>

The screenshot shows the Oracle Maestro SQL Editor interface. The main window displays the SQL Editor for the procedure `ADD_JOB_HISTORY`. The **Parameters** tab is active, showing a table of parameters:

Name	Type	Scope
P_EMP_ID	NUMBER	Input
P_START_DATE	DATE	Input
P_END_DATE	DATE	Input
P_JOB_ID	VARCHAR2	Input
P_DEPARTMENT_ID	NUMBER	Input

The **Definition** tab shows the following SQL code:

```

BEGIN
  INSERT INTO job_history (employee_id,
                          start_date,
                          end_date,
                          job_id,
                          department_id)
  VALUES (p_emp_id,
          p_start_date,
          p_end_date,
          p_job_id,
          p_department_id);
END add job history;

```

The **Properties** tab shows the following details:

Common	
Name	ADD_JOB_HISTORY
Created	18.11.2008 15:23:03
Last DDL time	18.11.2008 15:23:03
Time stamp	2008-11-18:15:23:03

Procedure	
Invoker rights	DEFINER

The status bar at the bottom indicates the database is ORCL11G at server and the procedure is Modified.

### 5.5.2.1 Editing properties

The **Parameters** tab contains the list of the current procedure parameters with its options. Here you can view the **Name** and the **Type** of each parameter of the procedure, and the parameter's **Scope**

The **Errors** tab displays all the necessary information about object errors. If an error have occurred during the object compilation it appears in the list with some additional properties: **Order** (one after another), **Line** and **Position** (object definition location the error was found out), **Error** (corresponding PL/SQL exception).

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 procedure.

#### 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 Oracle server.

#### Owner

There is the owner for the procedure.

#### Created

The field displays the date the object was created.

#### Last DDL time

Use the field to find the date when the last data definition language (DDL) operation was performed on the current object. The Last DDL time can help you to find if any changes to the object definitions have been made on or after a specific time.

#### Invoker rights (DEFINER, CURRENT USER)

Specify **DEFINER** to indicate that the procedure executes with the privileges of the owner of the schema in which the procedure resides, and that external names resolve in the schema where the procedure resides. Specify **CURRENT USER** to indicate that the procedure executes with the privileges of **CURRENT USER**. This clause also specifies that external names in queries, DML operations, and dynamic SQL statements resolve in the schema of **CURRENT USER**. External names in all other statements resolve in the schema in which the procedure resides.

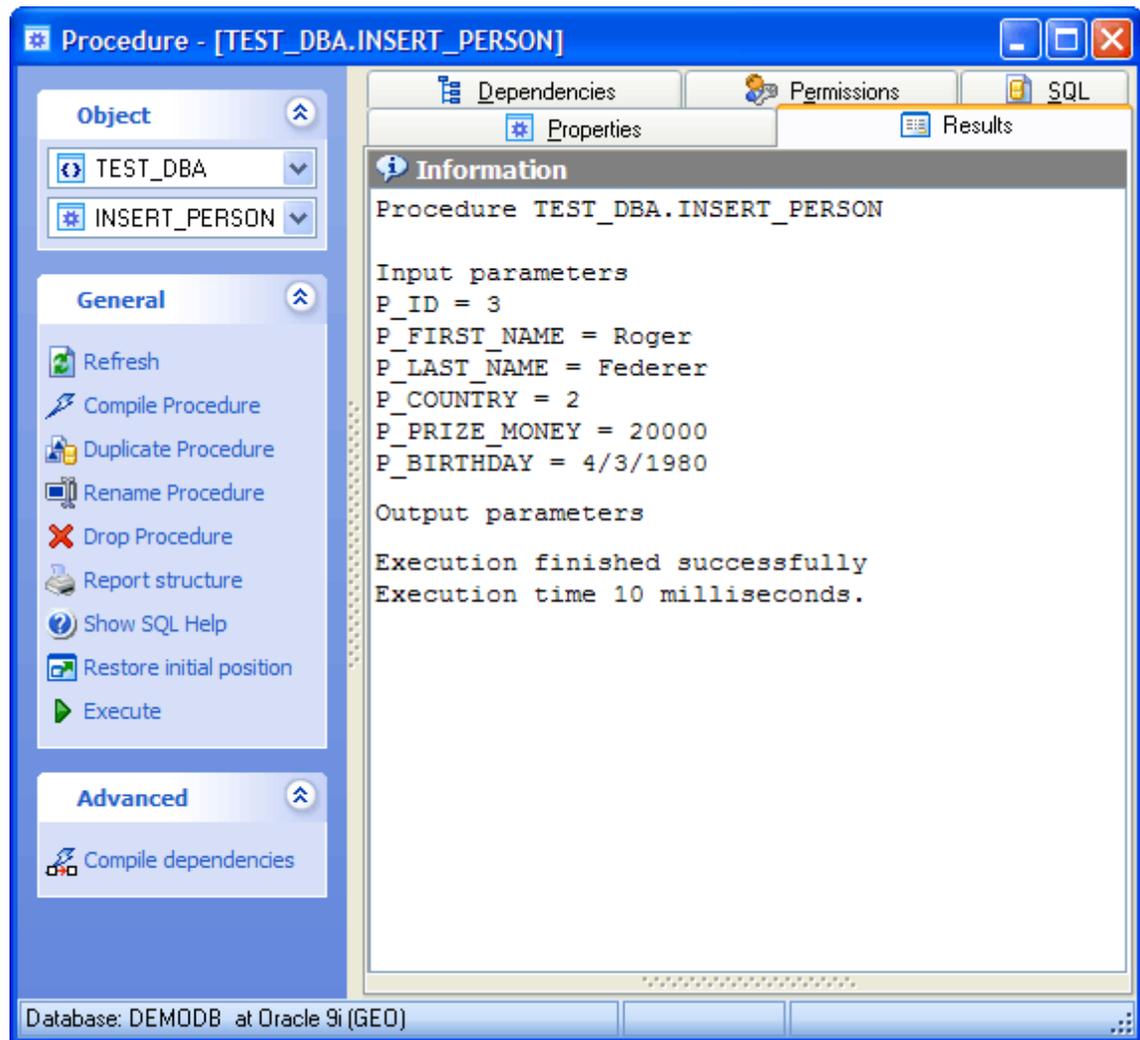
To apply the changes, select the **Apply Changes** item in the **Navigation bar** or use **Ctrl+F9** or **Ctrl+F7** shortcut keys.

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.

### 5.5.2.2 Viewing procedure results

The **Results** tab represents the result of the successfully executed procedure, if it returns a data that is represented as a grid (see Data View for details). Use grid's popup menu to export data, get SQL dump. In result with BLOB fields you can also call the **BLOB Editor** to view the BLOB fields. The data in result set is always read-only.

See also: [Executing procedure](#)<sup>44</sup>



## 5.6 Functions

A stored function (also called a user function or user defined function) is a set of PL/SQL statements you can call by name. Stored functions are very similar to procedures, except that a function returns a value to the environment in which it is called. User functions can be used as part of a SQL expression. 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>[123]</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 [Functions](#) 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>[125]</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 [Functions](#) 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](#))

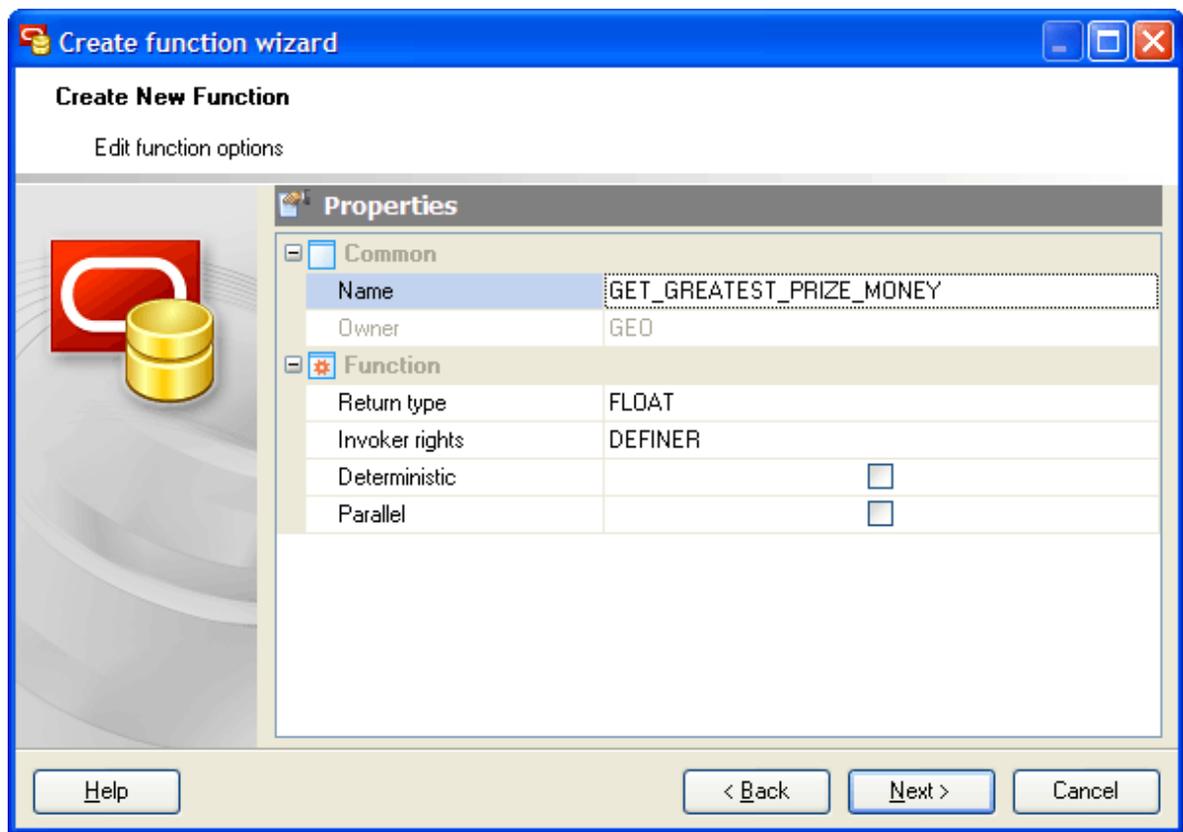
and confirm dropping in the dialog window.

## 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>[122]</sup> for instructions on running this wizard.

The basic principles of Create Object Wizards in Oracle Maestro are explained in a [separate topic](#)<sup>[37]</sup>. Below you will find a description of wizard steps that are unique for 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.

#### Return type

Defines the data type of the function result.

#### Invoker rights (DEFINER, CURRENT USER)

Specify **DEFINER** to indicate that the procedure executes with the privileges of the owner of the schema in which the procedure resides, and that external names resolve in the schema where the procedure resides. Specify **CURRENT USER** to indicate that the procedure executes with the privileges of **CURRENT USER**. This clause also specifies that external names in queries, DML operations, and dynamic SQL statements resolve in the schema of **CURRENT USER**. External names in all other statements resolve in the schema in which the procedure resides.

#### 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

#### Parallel

The option indicates that the function can be executed from a parallel execution server

of a parallel query operation. The function should not use session state, such as package variables, as those variables are not necessarily shared among the parallel execution servers.

### 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>[44]</sup>. Use the [Edit](#) and [Delete](#) items to manage Function parameters.

### Specifying Function Definition

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

## 5.6.2 Function Editor

[Function Editor](#) allows you to execute the existing Function, and edit its definition (Function name, parameter list, etc.). 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](#)).

The basic principles of Object Editors in Oracle Maestro are explained in a [separate topic](#)<sup>[39]</sup>. Below you will find a description of editor tabs that are unique for the current object.

- [Editing Function properties](#)<sup>[125]</sup>
- [Viewing Function results](#)<sup>[127]</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 view its [Scope](#)

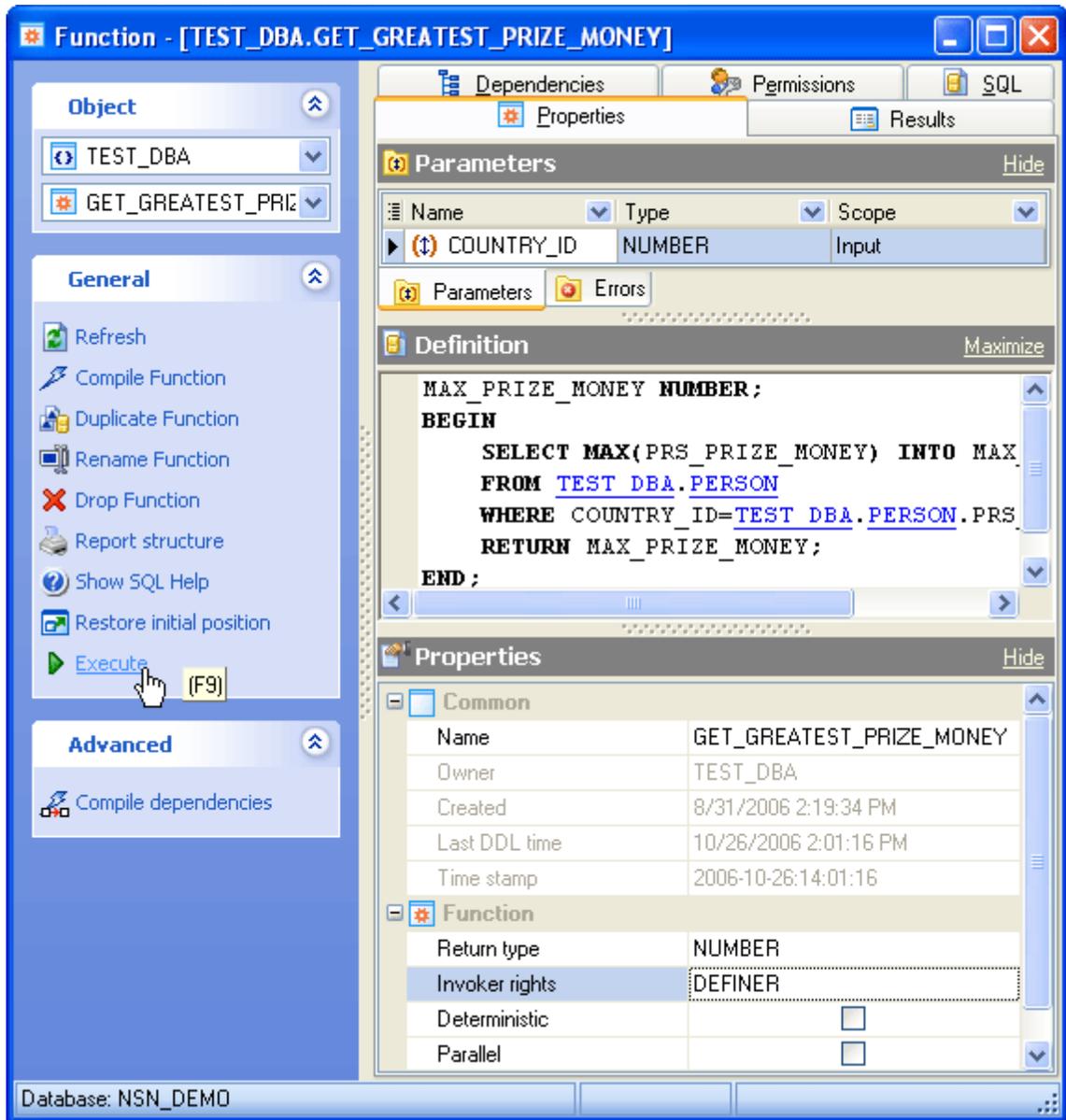
Parameters can be edited within the [Parameter Editor](#) dialog window. In order to open the dialog you should

- 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 [Errors](#) tab displays all the necessary information about object errors. If an error have occurred during the object compilation it appears in the list with some additional properties: [Order](#) (one after another), [Line](#) and [Position](#) (object definition location the error was found out), [Error](#) (corresponding PL/SQL exception).

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.

#### Created

The field displays the date the object was created.

#### Last DDL time

Use the field to find the date when the last data definition language (DDL) operation was

performed on the current object. The Last DDL time can help you to find if any changes to the object definitions have been made on or after a specific time.

#### Return type

The field defines the data type of the function result.

#### Invoker rights (DEFINER, CURRENT USER)

Specify **DEFINER** to indicate that the procedure executes with the privileges of the owner of the schema in which the procedure resides, and that external names resolve in the schema where the procedure resides. Specify **CURRENT USER** to indicate that the procedure executes with the privileges of **CURRENT USER**. This clause also specifies that external names in queries, DML operations, and dynamic SQL statements resolve in the schema of **CURRENT USER**. External names in all other statements resolve in the schema in which the procedure resides.

#### 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

#### Parallel

The option indicates that the function can be executed from a parallel execution server of a parallel query operation. The function should not use session state, such as package variables, as those variables are not necessarily shared among the parallel execution servers.

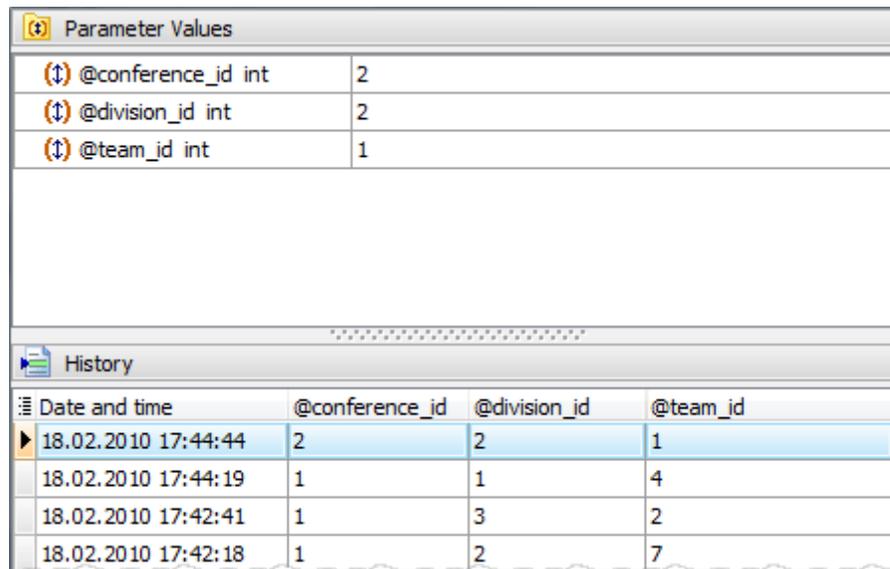
To apply the changes, select the **Apply Changes** item in the **Navigation bar** or use **Ctrl+F9** or **Ctrl+F7** shortcut keys.

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

**Procedure 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 procedure has parameters, Oracle 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.



The screenshot shows two dialog boxes from Oracle Maestro. The top dialog, titled "Parameter Values", contains three rows of parameter information:

Parameter	Value
@conference_id int	2
@division_id int	2
@team_id int	1

The bottom dialog, titled "History", contains a table with four columns: "Date and time", "@conference\_id", "@division\_id", and "@team\_id". It lists four historical entries:

Date and time	@conference_id	@division_id	@team_id
18.02.2010 17:44:44	2	2	1
18.02.2010 17:44:19	1	1	4
18.02.2010 17:42:41	1	3	2
18.02.2010 17:42:18	1	2	7

Oracle 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 [Procedure 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](#).

## 5.7 Synonyms

A [synonym](#) is an alternative name for a schema-scoped object. You can use a single-part name to reference a base object by using a synonym instead of using a two-part, three-part, or four-part name to reference the base object. Synonyms provide both data independence and location transparency. Synonyms permit applications to function without modification regardless of which user owns the table or view and regardless of which database holds the table or view. However, synonyms are not a substitute for privileges on database objects. Appropriate privileges must be granted to a user before the user can use the synonym.

### ■ How can I create a new synonym?

New synonyms are created within Create Synonym Wizard. In order to run the wizard you should either

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

To create a new synonym with the same properties as one of the existing synonyms has:

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

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

Synonyms can be edited within Synonym Editor. In order to run the editor you should either

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

You can change the name of the synonym using the [Rename](#)

[Synonym](#) dialog. To open the dialog you should either

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

or

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

#### ■ **How can I drop a synonym?**

To drop a synonym:

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

or

- open the schema in [Schema Editor](#) and the [Synonyms](#) tab there;
- select the synonym to drop;
- press the **Delete** key or select the [Drop Synonym](#) 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 Synonym Wizard

New synonyms are created within Create Synonym Wizard. In order to run the wizard you should either

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

or

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

or

- open the schema in [Schema Editor](#) and the [Synonyms](#) tab there;
- press the **Insert** key or select the [Create New Synonym...](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#)).

To create a new synonym with the same properties as one of the existing synonyms has:

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

The basic principles of Create Object Wizards in Oracle Maestro are explained in a [separate topic](#)<sup>[37]</sup>. Below you will find a description of wizard steps that are unique for the current object.

Owner

Select the owner for the synonym.

#### Database link

You can specify a complete or partial database link to create a synonym for a schema object on a remote database where the object is located. If you specify the link and omit schema, then the synonym refers to an object in the schema specified by the database link.

#### Object

Specify the base object that the synonym references.

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

## 5.7.2 Synonym Editor

Synonyms can be edited within Synonym Editor. In order to run the editor you should either

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

or

- open the schema in [Schema Editor](#) and the [Synonyms](#) tab there;
- select the synonym to edit;
- press the **Enter** key or select the [Edit Synonym](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#)).

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

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

or

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

The basic principles of Object Editors in Oracle Maestro are explained in a [separate topic](#)<sup>[39]</sup>. Below you will find a description of editor tabs that are unique for the current object.

**See also:** [Synonym properties](#)<sup>[131]</sup>, [Create Synonym Wizard](#)<sup>[130]</sup>

### 5.7.2.1 Editing synonym properties

#### Synonym Properties

##### Name

You can edit the synonym name here.

### Owner

Here you can view the owner for the synonym.

### Created

The field displays the date the object was created.

### Last DDL time

Use the field to find the date when the last data definition language (DDL) operation was performed on the current object. The Last DDL time can help you to find if any changes to the object definitions have been made on or after a specific time.

### Database link

You can specify a complete or partial database link to create a synonym for a schema object on a remote database where the object is located. If you specify the link and omit schema, then the synonym refers to an object in the schema specified by the database link.

### Object

The field represents the base object that the synonym references.

## Synonym Data

The **Data** tab displays the Alias data as a grid (see [Data View](#)<sup>[279]</sup> 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.

## 5.8 Schema Triggers

The schema trigger fires whenever any user connected as schema initiates the triggering event.

### ■ How can I create a new schema trigger?

New schema triggers are created within [Create Schema Trigger Wizard](#)<sup>[134]</sup>. In order to run the wizard you should either

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

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

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

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

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

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

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

- select the schema trigger to rename in the explorer tree;

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

#### ■ **How can I drop a schema trigger?**

To drop a schema trigger:

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

and confirm dropping in the dialog window.

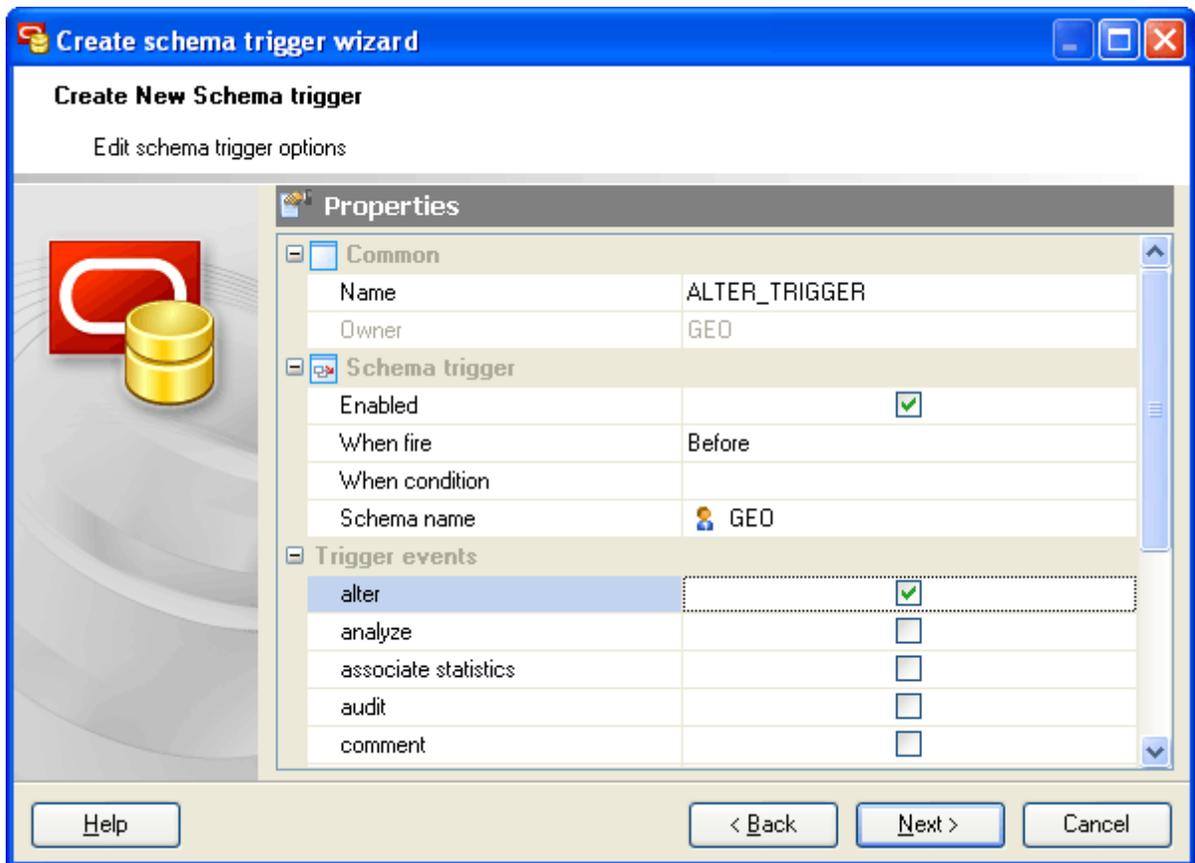
### 5.8.1 Create Schema Triggers Wizard

[Create Schema Trigger Wizard](#) guides you through the process of creating a new schema trigger. See [How To Create Schema Trigger](#)<sup>[133]</sup> for instructions on running this wizard.

The basic principles of Create Object Wizards in Oracle Maestro are explained in a [separate topic](#)<sup>[37]</sup>. Below you will find a description of wizard steps that are unique for the current object.

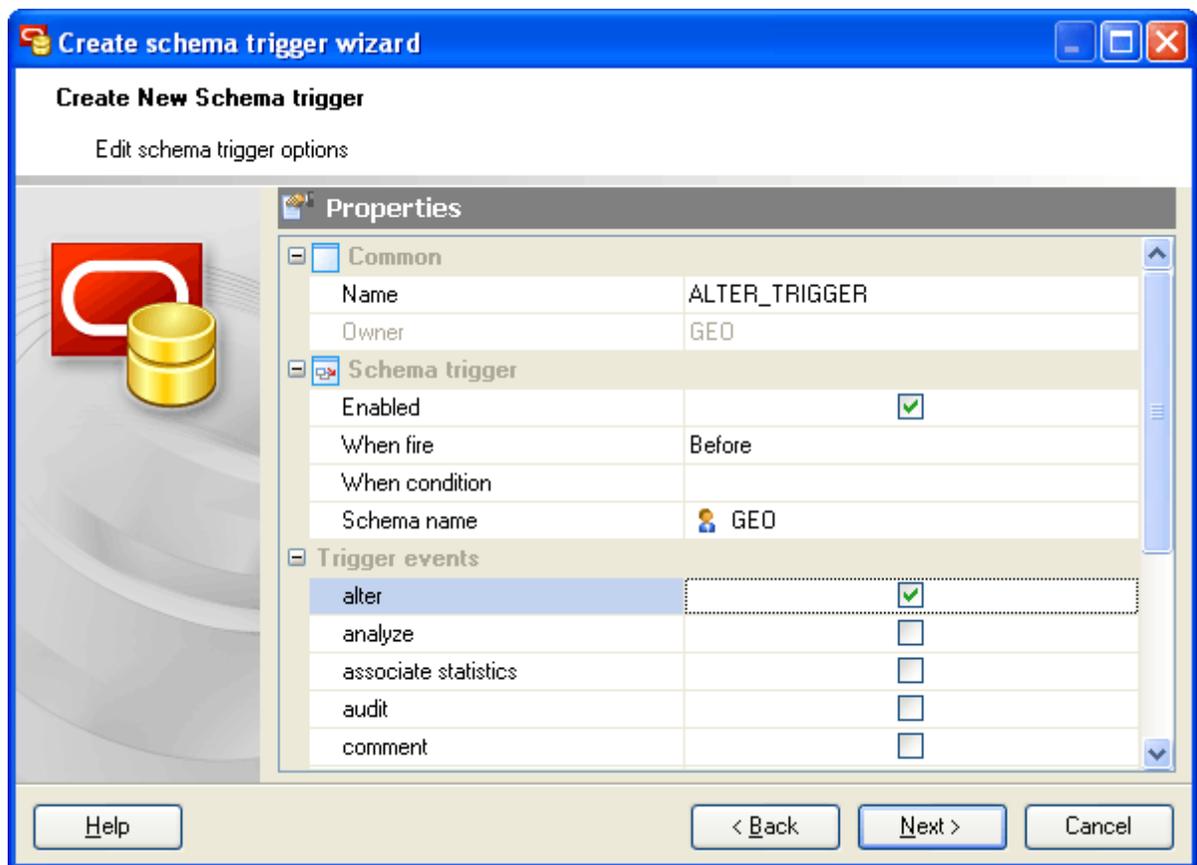
- [Specifying schema trigger properties](#)<sup>[135]</sup>
- [Specifying trigger definition](#)<sup>[136]</sup>

**See also:** [Schema Trigger Editor](#)<sup>[138]</sup>



#### 5.8.1.1 Specifying schema trigger properties

The wizard step was supplied to define common schema trigger properties. The detailed description of the properties you can find below.



#### Name

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

#### Owner

The field displays the owner of the new trigger.

#### Enabled

Check the option to enable the trigger.

#### When fire (Before, After)

Specify **Before** to cause the database to fire the trigger before executing the triggering event. For row triggers, the trigger is fired before each affected row is changed. Specify **After** to cause the database to fire the trigger after executing the triggering event. For row triggers, the trigger is fired after each affected row is changed.

#### When condition

Specify the trigger condition, which is a SQL condition that must be satisfied for the database to fire the trigger.

Specify the **Schema name** to define the trigger on the current schema. The trigger fires whenever any user connected as schema initiates the triggering event.

#### alter

Specify **alter** to fire the trigger whenever an ALTER statement modifies a database

object in the data dictionary.

[analyze](#)

Specify [analyze](#) to fire the trigger whenever the database collects or deletes statistics or validates the structure of a database object.

[associate statistic](#)

Specify ASSOCIATE STATISTICS to fire the trigger whenever the database associates a statistics type with a database object.

[audit](#)

Specify [audit](#) to fire the trigger whenever the database tracks the occurrence of a SQL statement or tracks operations on a schema object.

[comment](#)

Specify [comment](#) to fire the trigger whenever a comment on a database object is added to the data dictionary.

[create](#)

Specify [create](#) to fire the trigger whenever a CREATE statement adds a new database object to the data dictionary.

[diassociate statistic](#)

Specify [diassociate statistic](#) to fire the trigger whenever the database disassociates a statistics type from a database object.

[drop](#)

Specify [drop](#) to fire the trigger whenever a DROP statement removes a database object from the data dictionary.

[grant](#)

Specify [grant](#) to fire the trigger whenever a user grants system privileges or roles or object privileges to another user or to a role.

[no audit](#)

Specify [no audit](#) to fire the trigger whenever a [no audit](#) statement instructs the database to stop tracking a SQL statement or operations on a schema object.

[rename](#)

Specify [rename](#) to fire the trigger whenever a [rename](#) statement changes the name of a database object.

[revoke](#)

Specify [revoke](#) to fire the trigger whenever a [revoke](#) statement removes system privileges or roles or object privileges from a user or role.

[truncate](#)

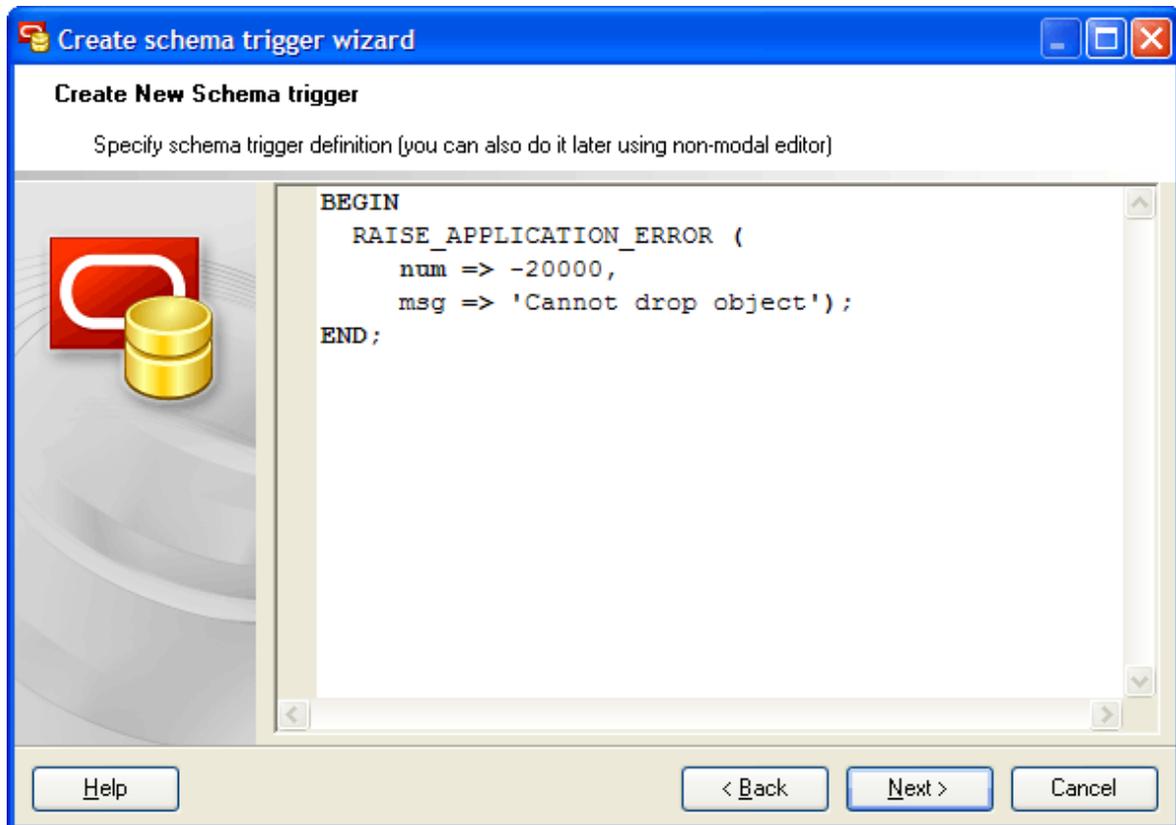
Specify [truncate](#) to fire the trigger whenever a [truncate](#) statement removes the rows from a table or cluster and resets its storage characteristics.

[DDL](#)

Specify [DDL](#) to fire the trigger whenever any of the preceding [DDL](#) statements is issued.

### 5.8.1.2 Specifying trigger definition

Here you can specify the system trigger [definition](#). Specify the trigger steps to be executed when the trigger fires. You can also do it later using non-modal editor.



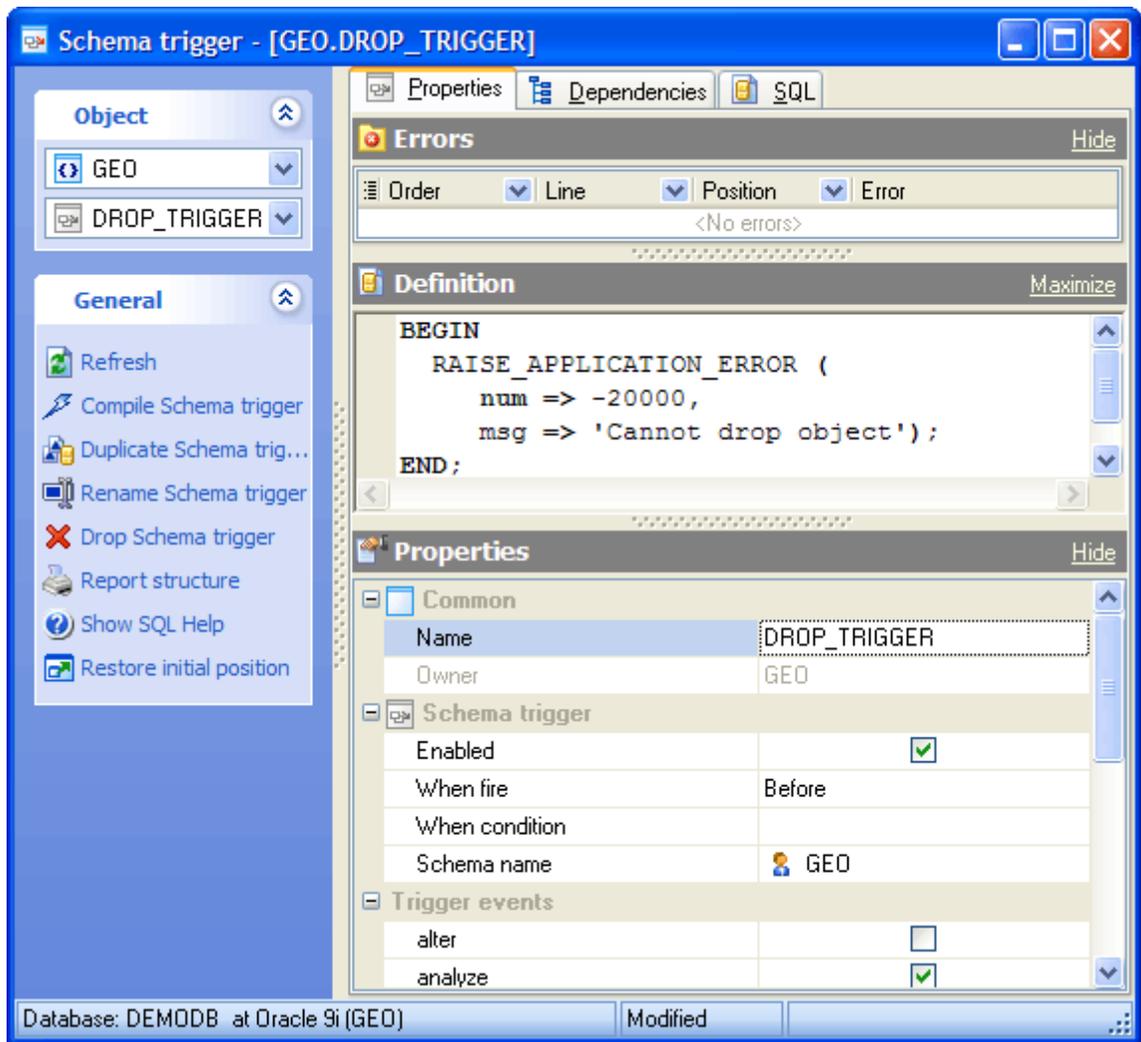
## 5.8.2 Schema Triggers Editor

[Trigger Editor](#) can be opened automatically after the trigger is created and is available on editing the trigger (see [Editing Schema Triggers](#)<sup>[133]</sup> for details).

The basic principles of Object Editors in Oracle Maestro are explained in a [separate topic](#)<sup>[39]</sup>. Below you will find a description of editor tabs that are unique for the current object.

For more information about the editor's parts and their elements see [Schema trigger properties](#)<sup>[139]</sup>.

**See also:** [Create Schema Trigger Wizard](#)<sup>[134]</sup>



### 5.8.2.1 Editing trigger properties

Schema Trigger Editor provides you with an ability to edit trigger properties. The Properties tab allows you to change the trigger name, the trigger events.

The Errors tab displays all the necessary information about object errors. If an error have occurred during the object compilation it appears in the list with some additional properties: **Order** (one after another), **Line** and **Position** (object definition location the error was found out), **Error** (corresponding PL/SQL exception).

The Definition field contains the definition of the schema trigger.

#### 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 Oracle server.

### Owner

There is the owner for the trigger.

### Enabled

Check the option to enable the trigger.

### When fire (Before, After)

Specify **Before** to cause the database to fire the trigger before executing the triggering event. For row triggers, the trigger is fired before each affected row is changed. Specify **After** to cause the database to fire the trigger after executing the triggering event. For row triggers, the trigger is fired after each affected row is changed.

### When condition

Specify the trigger condition, which is a SQL condition that must be satisfied for the database to fire the trigger.

Specify the **Schema name** to define the trigger on the current schema. The trigger fires whenever any user connected as schema initiates the triggering event.

### alter

Specify **alter** to fire the trigger whenever an ALTER statement modifies a database object in the data dictionary.

### analyze

Specify **analyze** to fire the trigger whenever the database collects or deletes statistics or validates the structure of a database object.

### associate statistic

Specify ASSOCIATE STATISTICS to fire the trigger whenever the database associates a statistics type with a database object.

### audit

Specify **audit** to fire the trigger whenever the database tracks the occurrence of a SQL statement or tracks operations on a schema object.

### comment

Specify **comment** to fire the trigger whenever a comment on a database object is added to the data dictionary.

### create

Specify **create** to fire the trigger whenever a CREATE statement adds a new database object to the data dictionary.

### diassociate statistic

Specify **diassociate statistic** to fire the trigger whenever the database disassociates a statistics type from a database object.

### drop

Specify **drop** to fire the trigger whenever a DROP statement removes a database object from the data dictionary.

### grant

Specify **grant** to fire the trigger whenever a user grants system privileges or roles or

object privileges to another user or to a role.

**no audit**

Specify **no audit** to fire the trigger whenever a **no audit** statement instructs the database to stop tracking a SQL statement or operations on a schema object.

**rename**

Specify **rename** to fire the trigger whenever a **rename** statement changes the name of a database object.

**revoke**

Specify **revoke** to fire the trigger whenever a **revoke** statement removes system privileges or roles or object privileges from a user or role.

**truncate**

Specify **truncate** to fire the trigger whenever a **truncate** statement removes the rows from a table or cluster and resets its storage characteristics.

**DDL**

Specify **DDL** to fire the trigger whenever any of the preceding **DDL** statements is issued. 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.

## 5.9 Database links

A database link is a schema object in one database that enables you to access objects on another database. The other database need not be an Oracle Database system.

Once you have created a database link, you can use it to refer to tables and views on the other database. In SQL statements, you can refer to a table or view on the other database by appending @dblink to the table or view name. You can query a table or view on the other database with the SELECT statement. You can also access remote tables and views using any INSERT, UPDATE, DELETE, or LOCK TABLE statement.

### ■ How can I create a new database link?

New database links are created within [Create Database Link Wizard](#)<sup>143</sup>. In order to run the wizard you should either

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

To create a new database link with the same properties as one of the existing database link has:

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

### ■ How can I edit an existing database link?

Database links can be edited within [Database link Editor](#)<sup>145</sup>. In order to run the editor you should either

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

link of the [Navigation Bar](#)).

You can change the name of the database link using the [Rename Database Link](#) dialog. To open the dialog you should either

- select the database link to rename in the explorer tree;
- select the [Rename Database Link](#) item from the popup menu

or

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

#### ■ **How can I drop a database link?**

To drop a database link:

- select the database link to drop in the explorer tree;
- select the [Drop Database Link](#) item from the popup menu

or

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

and confirm dropping in the dialog window.

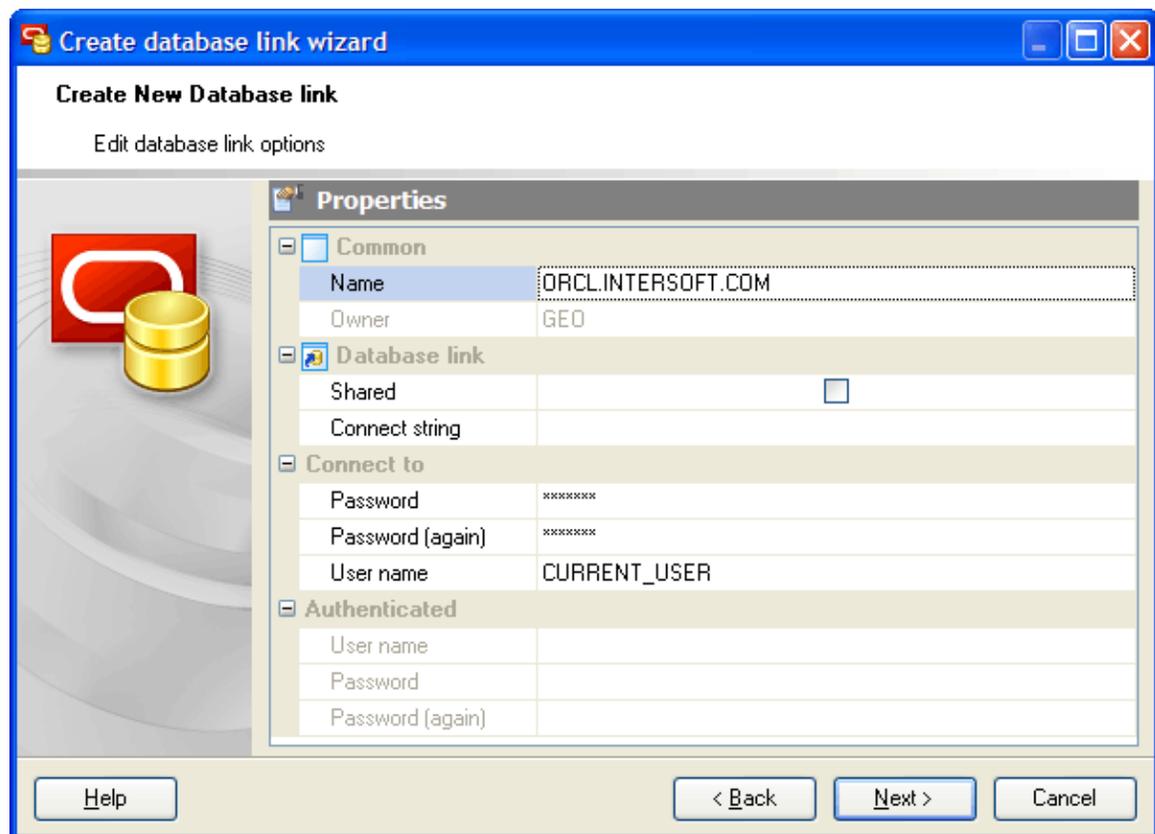
### 5.9.1 Create Database Link Wizard

[Create Database Link Wizard](#) guides you through the process of creating a new link. See [How To Create Database Link](#)<sup>[142]</sup> for instructions on running this wizard.

The basic principles of Create Object Wizards in Oracle Maestro are explained in a [separate topic](#)<sup>[37]</sup>. Below you will find a description of wizard steps that are unique for the current object.

- [Specifying link properties](#)<sup>[144]</sup>

**See also:** [Database Link Editor](#)<sup>[145]</sup>



### 5.9.1.1 Specifying link properties

The wizard step was supplied to define common database link properties. The detailed description of the properties you can find below.

#### Name

The field represents the new database link name as it was set on the previous wizard step.

#### Owner

The field displays the owner of the new database link.

#### Shared

Specify SHARED to use a single network connection to create a public database link that can be shared among multiple users.

#### Connect string

Specify the [service name](#) of a remote database. If you specify only the database name, then Oracle Database implicitly appends the database domain to the connect string to create a complete service name. Therefore, if the database domain of the remote database is different from that of the current database, then you must specify the complete service name

#### Connect to (Password, User name)

The [Connect to](#) clause lets you enable a connection to the remote database. Specify `CURRENT_USER` to create a current user database link. The current user must be a

global user with a valid account on the remote database. If the database link is used directly, that is, not from within a stored object, then the current user is the same as the connected user. When executing a stored object (such as a procedure, view, or trigger) that initiates a database link, `CURRENT_USER` is the username that owns the stored object, and not the username that called the object. For example, if the database link appears inside procedure `scott.p` (created by `scott`), and user `jane` calls procedure `scott.p`, the current user is `scott`.

#### Autenticated (Password, User name)

Specify the `User name` and `Password` on the target instance. This clause authenticates the user to the remote server and is required for security. The specified username and password must be a valid username and password on the remote instance. The username and password are used only for authentication. No other operations are performed on behalf of this user.

You must specify this clause when using the `Shared` clause.

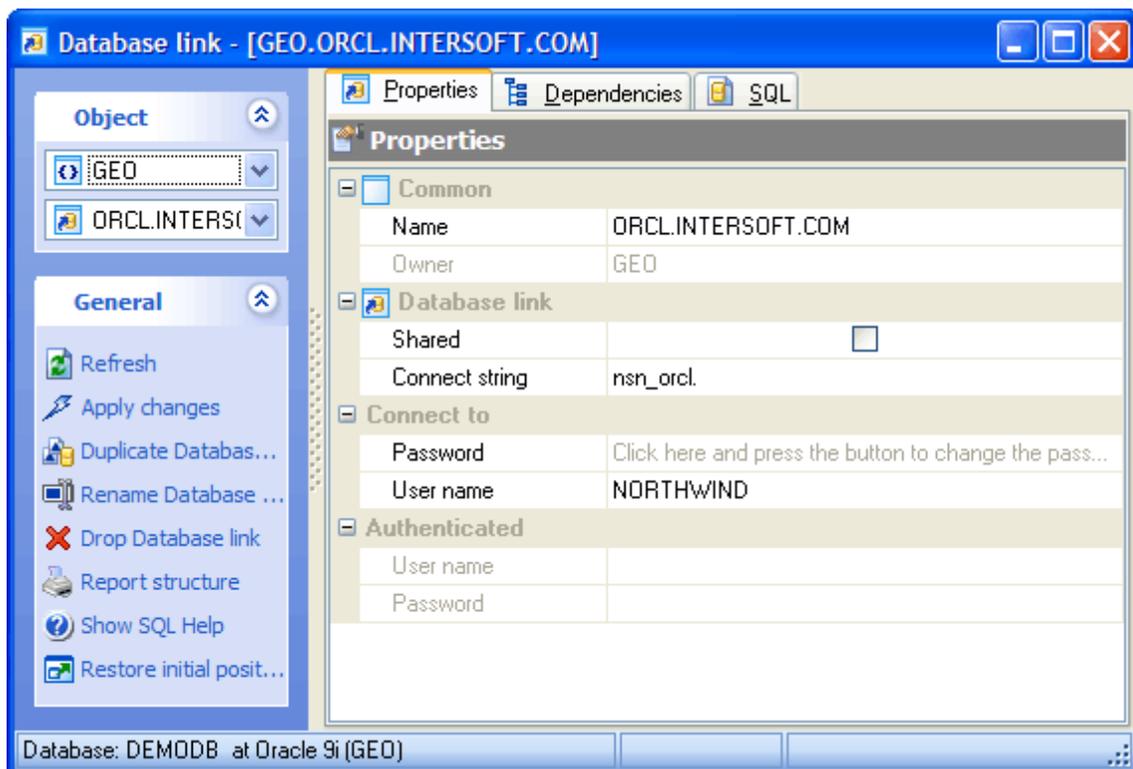
## 5.9.2 Database Link Editor

`Database Link Editor` can be opened automatically after the link is created and is available on editing (see [Editing Database links](#)<sup>[142]</sup> for details).

The basic principles of Object Editors in Oracle Maestro are explained in a [separate topic](#)<sup>[39]</sup>. Below you will find a description of editor tabs that are unique for the current object.

For more information about the editor's parts and their elements see [Database link properties](#)<sup>[146]</sup>.

**See also:** [Create Database Link Wizard](#)<sup>[143]</sup>



### 5.9.2.1 Editing link properties

[Database Link Editor](#) provides you with an ability to edit link properties. The detailed description of the properties you can find below.

#### Name

The field represents the new database link name as it was set on the previous wizard step.

#### Owner

The field displays the owner of the new database link.

#### Shared

Specify SHARED to use a single network connection to create a public database link that can be shared among multiple users.

#### Connect string

Specify the [service name](#) of a remote database. If you specify only the database name, then Oracle Database implicitly appends the database domain to the connect string to create a complete service name. Therefore, if the database domain of the remote database is different from that of the current database, then you must specify the complete service name

#### Connect to (Password, User name)

The [Connect to](#) clause lets you enable a connection to the remote database. Specify [CURRENT\\_USER](#) to create a current user database link. The current user must be a global user with a valid account on the remote database. If the database link is used directly, that is, not from within a stored object, then the current user is the same as the connected user. When executing a stored object (such as a procedure, view, or trigger) that initiates a database link, [CURRENT\\_USER](#) is the username that owns the stored object, and not the username that called the object. For example, if the database link appears inside procedure `scott.p` (created by `scott`), and user `jane` calls procedure `scott.p`, the current user is `scott`.

#### Authenticated (Password, User name)

Specify the [User name](#) and [Password](#) on the target instance. This clause authenticates the user to the remote server and is required for security. The specified username and password must be a valid username and password on the remote instance. The username and password are used only for authentication. No other operations are performed on behalf of this user.

You must specify this clause when using the [Shared](#) clause.

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 Packages

Package is an encapsulated collection of related procedures, functions, and other program objects stored together in the database. The package specification declares these objects. The package body, specified subsequently, defines these objects.

### ■ How can I create a new package?

New packages are created within [Create Package Wizard](#)<sup>[148]</sup>. In order to run the wizard you should either

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

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

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

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

Packages can be edited within [Package Editor](#)<sup>[153]</sup>. In order to run the editor you should either

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

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

- select the package to rename in the explorer tree;
  - select the [Rename Package](#) item from the popup menu
- or
- open the schema in [Schema Editor](#) and the [Packages](#) tab there;

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

#### ■ How can I drop a package?

To drop a package:

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

or

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

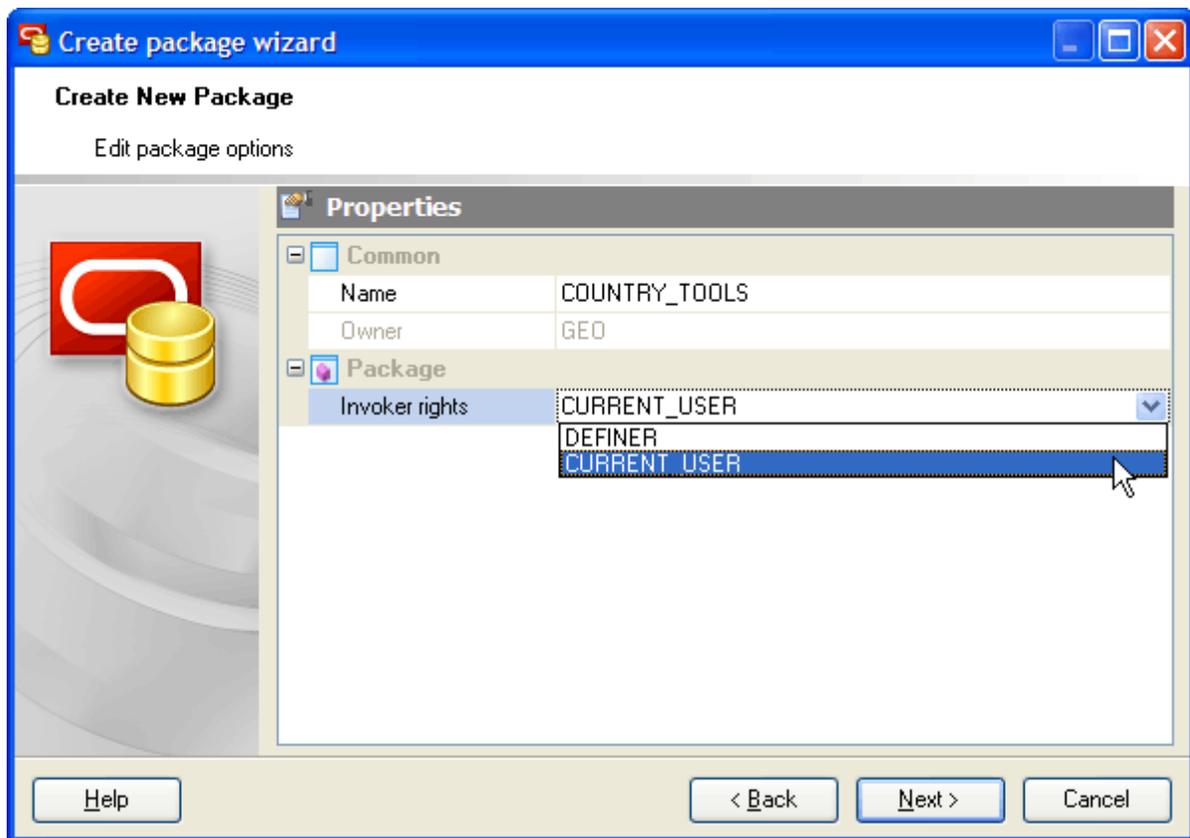
and confirm dropping in the dialog window.

### 5.10.1 Create Package Wizard

[Create Package Wizard](#) guides you through the process of creating a new database package. See [How To Create Package](#)<sup>[147]</sup> for instructions on running this wizard.

The basic principles of Create Object Wizards in Oracle Maestro are explained in a [separate topic](#)<sup>[37]</sup>. Below you will find a description of wizard steps that are unique for the current object.

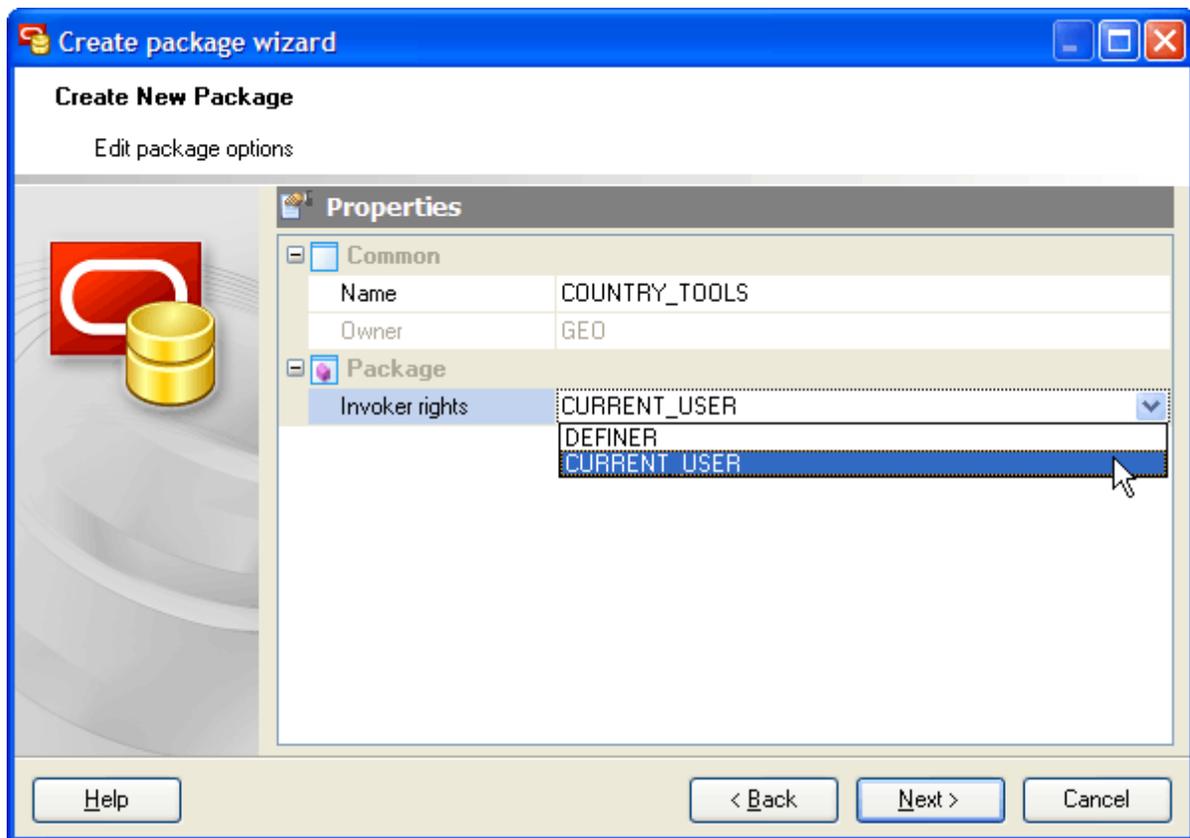
- [Specifying package properties](#)<sup>[149]</sup>
- [Adding methods](#)<sup>[150]</sup>
- [Specifying package definition](#)<sup>[152]</sup>



See also: [Package Editor](#) <sup>153</sup>

### 5.10.1.1 Specifying package properties

The wizard step was supplied to define common package properties. The detailed description of the properties you can find below.



#### Name

The field represents the new database link name as it was set on the previous wizard step.

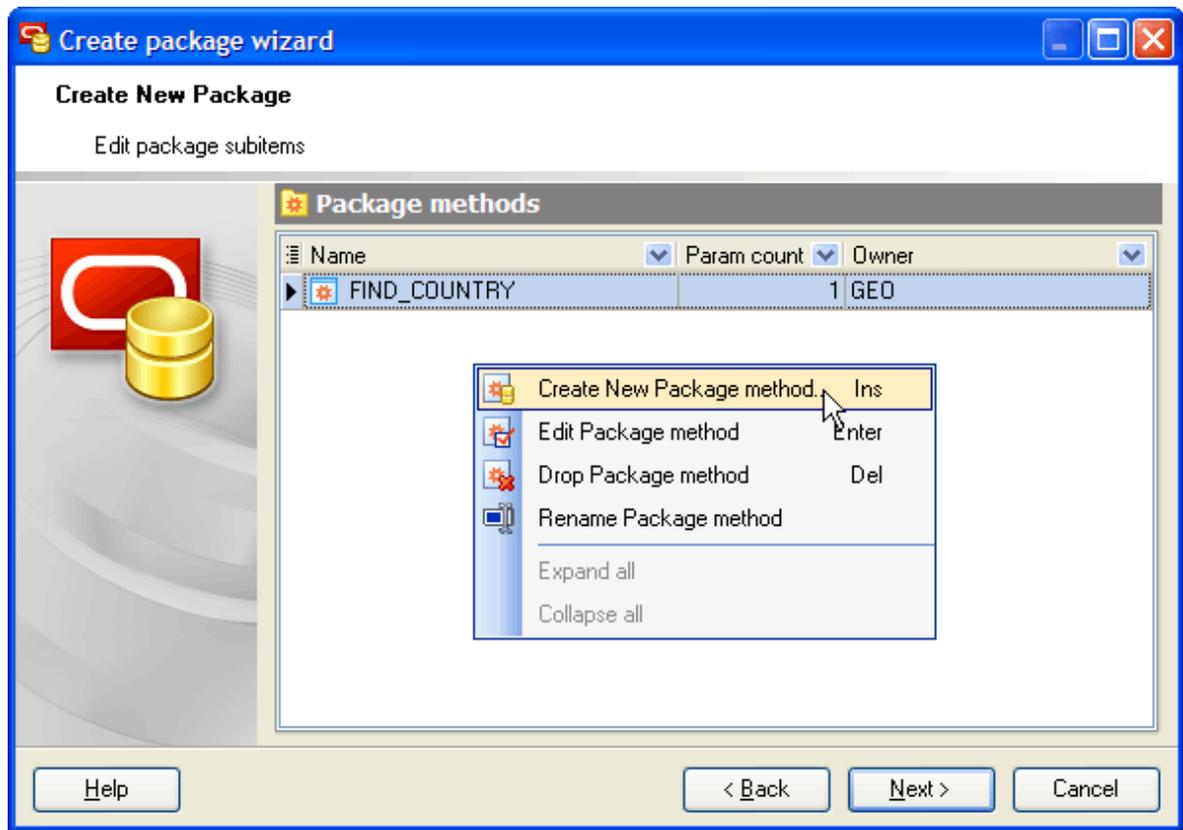
#### Owner

The field displays the owner of the new package.

The **Invoker Rights (Definer, Current\_user)** lets you specify whether the functions and procedures in the package execute with the privileges and in the schema of the user who owns the package or with the privileges and in the schema of **Current\_user**. This specification applies to the corresponding package body as well. This clause also determines how Oracle Database resolves external names in queries, DML operations, and dynamic SQL statements in the package.

#### 5.10.1.2 Adding methods

Use popup menu **Add New Method...** item to add a new method (function or procedure) and set its properties in [Create Package Method Wizard](#)<sup>[15]</sup>. Use the **Edit** and **Delete** items to manage prepared package methods.



#### 5.10.1.2.1 Create Package Method Wizard

The wizard is provided to supply a possibility to create functions and procedures during the package creation.

The first wizard step allows you to specify new function (procedure) properties.

##### Name

Specify a name for the function.

##### Owner

The field displays the owner of the new function.

##### Function

Check the option to create a new package function. Otherwise a procedure will be created.

##### Return type (Available only for functions)

Defines the data type of the function result.

##### Invoker rights (DEFINER, CURRENT USER)

Specify **DEFINER** to indicate that the procedure executes with the privileges of the owner of the schema in which the procedure resides, and that external names resolve in the schema where the procedure resides. Specify **CURRENT USER** to indicate that the procedure executes with the privileges of **CURRENT USER**. This clause also specifies that external names in queries, DML operations, and dynamic SQL statements resolve in the schema of **CURRENT USER**. External names in all other statements resolve in the schema

in which the procedure resides.

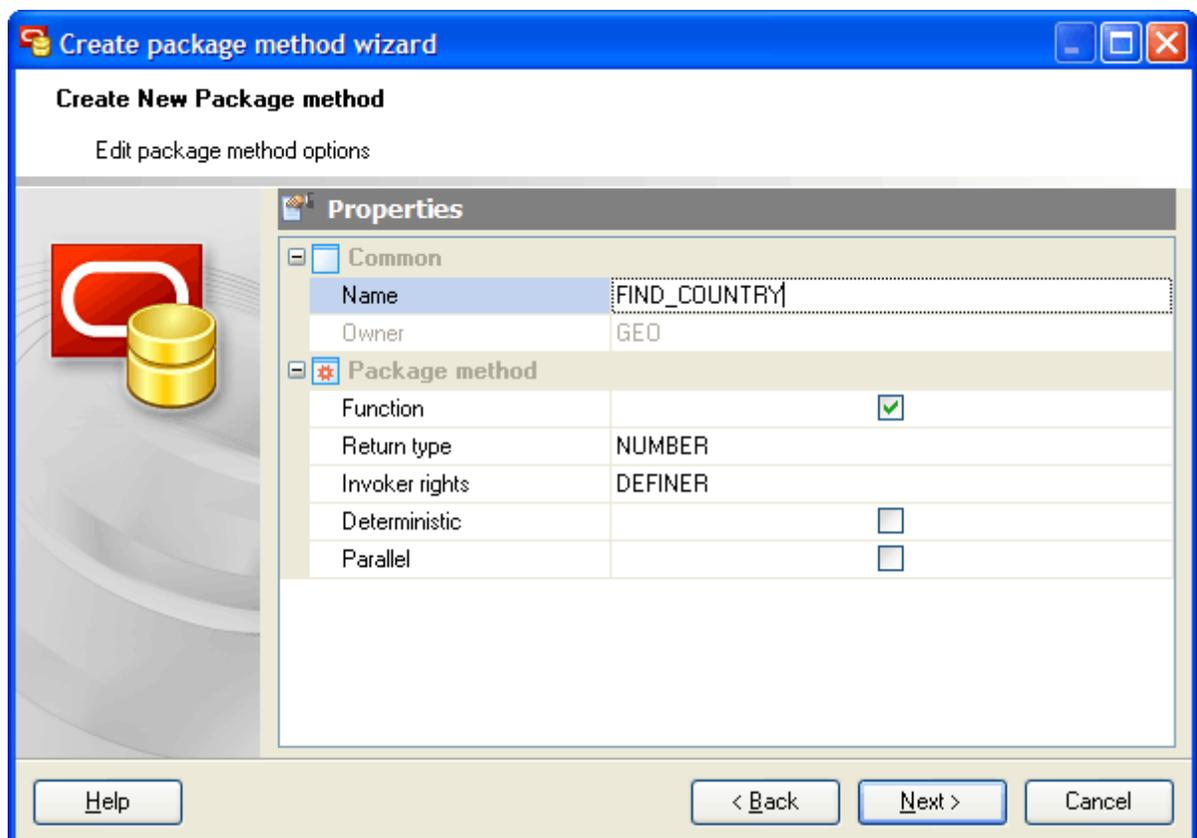
**Deterministic**

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

**Parallel**

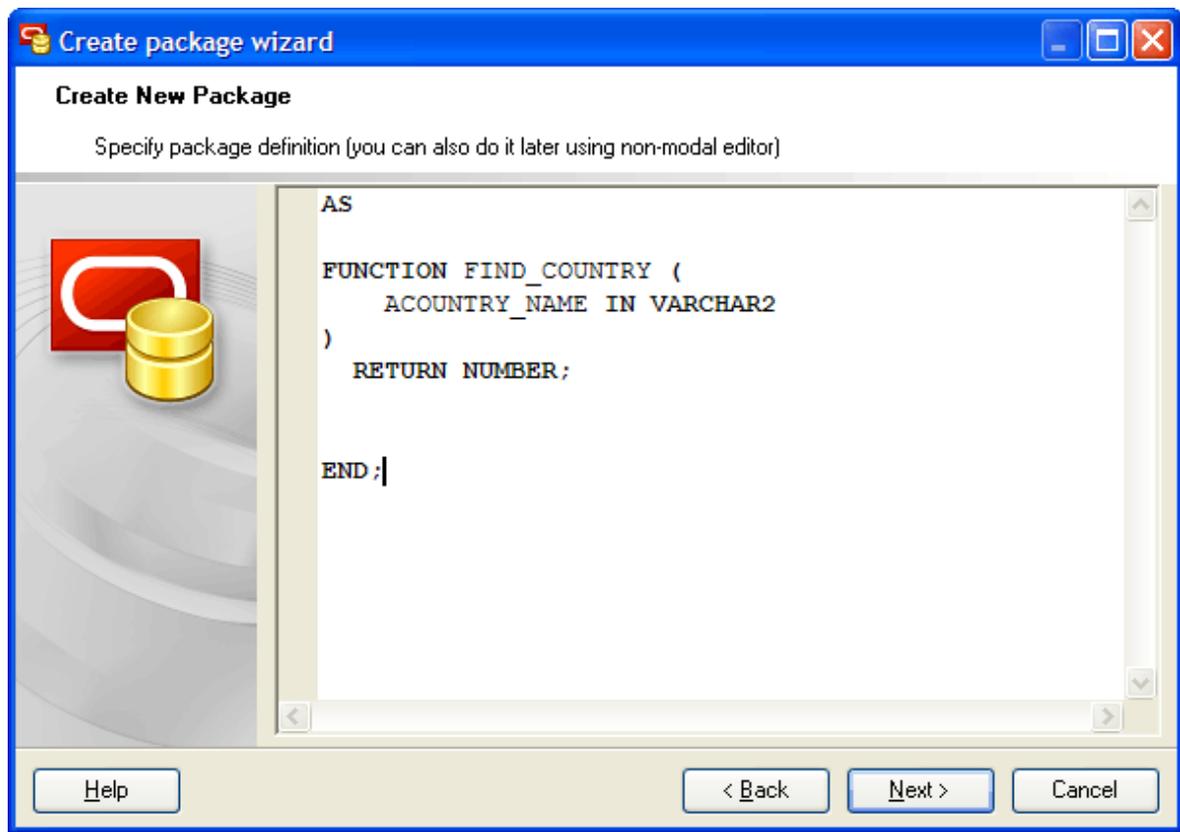
The option indicates that the function can be executed from a parallel execution server of a parallel query operation. The function should not use session state, such as package variables, as those variables are not necessarily shared among the parallel execution servers.

The next wizard step allows to define function/procedure parameters. Use popup menu **Add New Parameter...** item to add a new parameter and set its properties in **Parameter Editor**. Use the **Edit** and **Delete** items to manage method parameters.



### 5.10.1.3 Specifying package definition

Here you can specify the package definition. You can also do it later using non-modal editor.

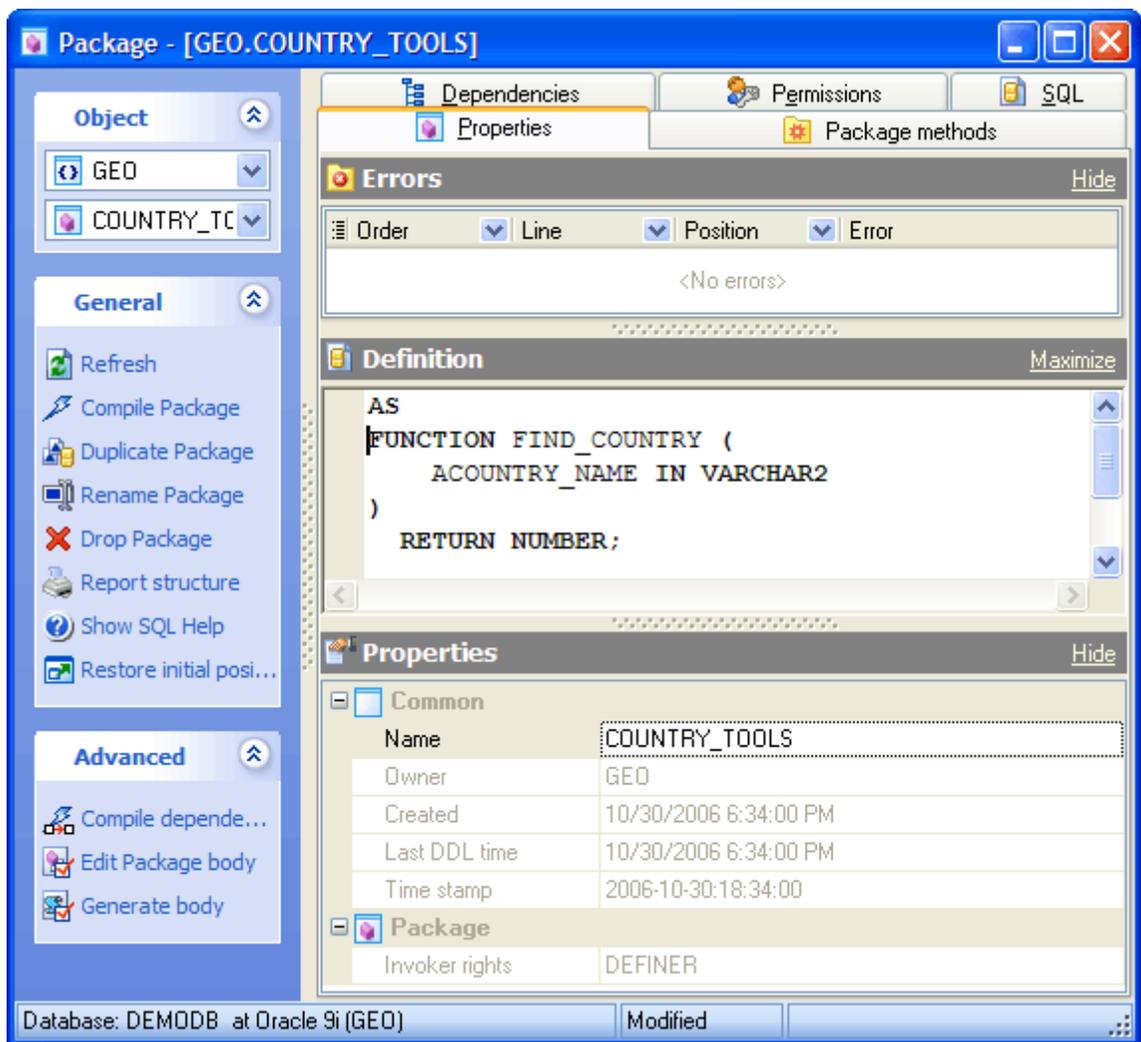


### 5.10.2 Package Editor

[Package Editor](#) can be opened automatically after the database package is created and is available on editing (see [Editing Packages](#)<sup>[147]</sup> for details).

The basic principles of Object Editors in Oracle Maestro are explained in a [separate topic](#)<sup>[39]</sup>. Below you will find a description of editor tabs that are unique for the current object.

**See also:** [Create Package Wizard](#)<sup>[148]</sup>



The **Errors** tab displays all the necessary information about object errors. If an error have occurred during the object compilation it appears in the list with some additional properties: **Order** (one after another), **Line** and **Position** (object definition location the error was found out), **Error** (corresponding PL/SQL exception).

#### Definition

The area contains the package definition which is available to editing.

#### Name

Use the field to change the package name.

You can also view here the package **Owner** and package **Invoker rights** and some statistic option.

#### Created

The field displays the date the object was created.

#### Last DDL time

Use the field to find the date when the last data definition language (DDL) operation was performed on the current object. The Last DDL time can help you to find if any changes to the object definitions have been made on or after a specific time.

The [Package methods](#) tab allows you to manage the package functions and procedures. The editor allows you to modify package definition and a comment to the package. 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 Package bodies

Package is an encapsulated collection of related procedures, functions, and other program objects stored together in the database. The package specification declares these objects. The package body, specified subsequently, defines these objects.

### ■ How can I create a package body?

New package bodies are created within [Create Package Body Wizard](#) <sup>[157]</sup>. In order to run the wizard you should either

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

To create a new package body with the same properties as one of the existing package body has:

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

### ■ How can I edit an existing package body?

Package bodies can be edited within [Package body Editor](#) <sup>[158]</sup>. In order to run the editor you should either

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

You can change the name of the package body using the [Rename Package Body](#) dialog. To open the dialog you should either

- select the package body to rename in the explorer tree;
- select the [Rename Package Body](#) item from the popup menu

or

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

#### ■ **How can I drop a package body?**

To drop a package body:

- select the package body to drop in the explorer tree;
- select the [Drop Package Body](#) item from the popup menu

or

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

and confirm dropping in the dialog window.

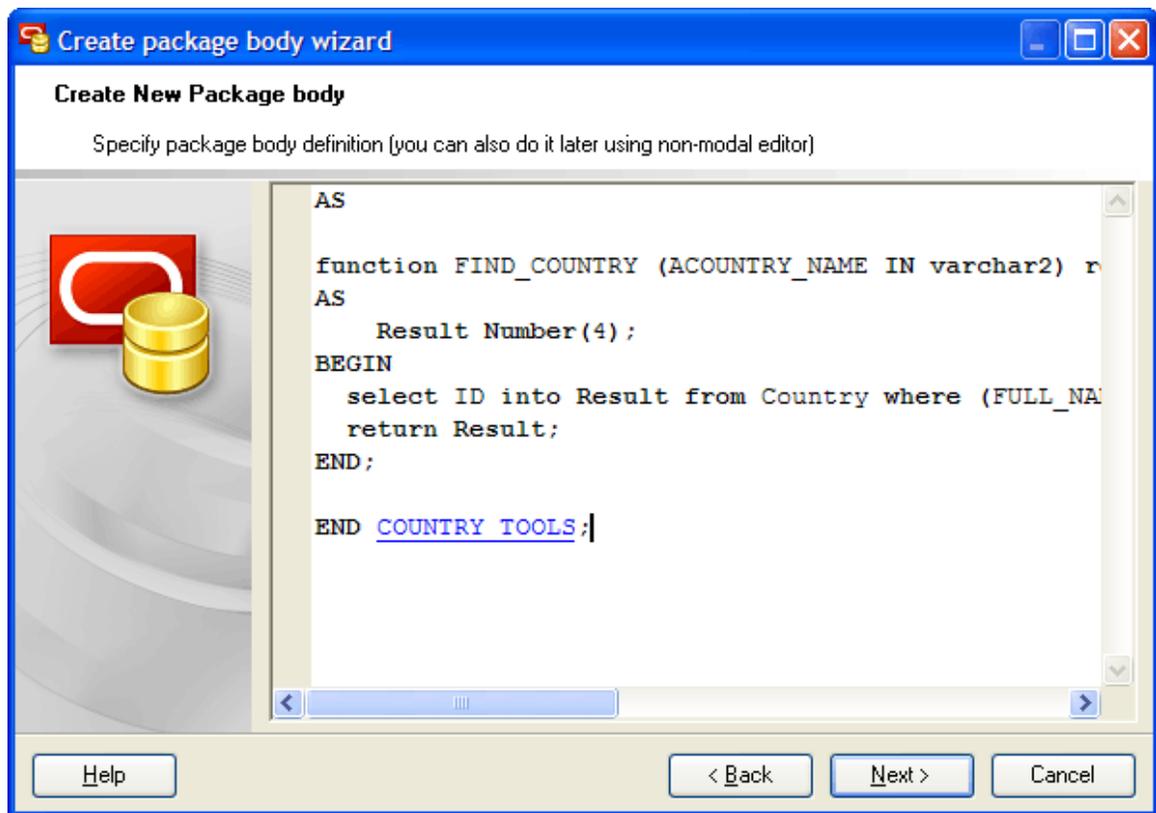
### 5.11.1 Create Package Body Wizard

[Create Package Body Wizard](#) guides you through the process of creating a new database package body. See [How To Create Package Body](#)<sup>[156]</sup> for instructions on running this wizard.

The basic principles of Create Object Wizards in Oracle Maestro are explained in a [separate topic](#)<sup>[37]</sup>. Below you will find a description of wizard steps that are unique for the current object.

On this step you can specify the package body [definition](#). You can also do it later using non-modal editor.

**See also:** [Package Body Editor](#)<sup>[158]</sup>

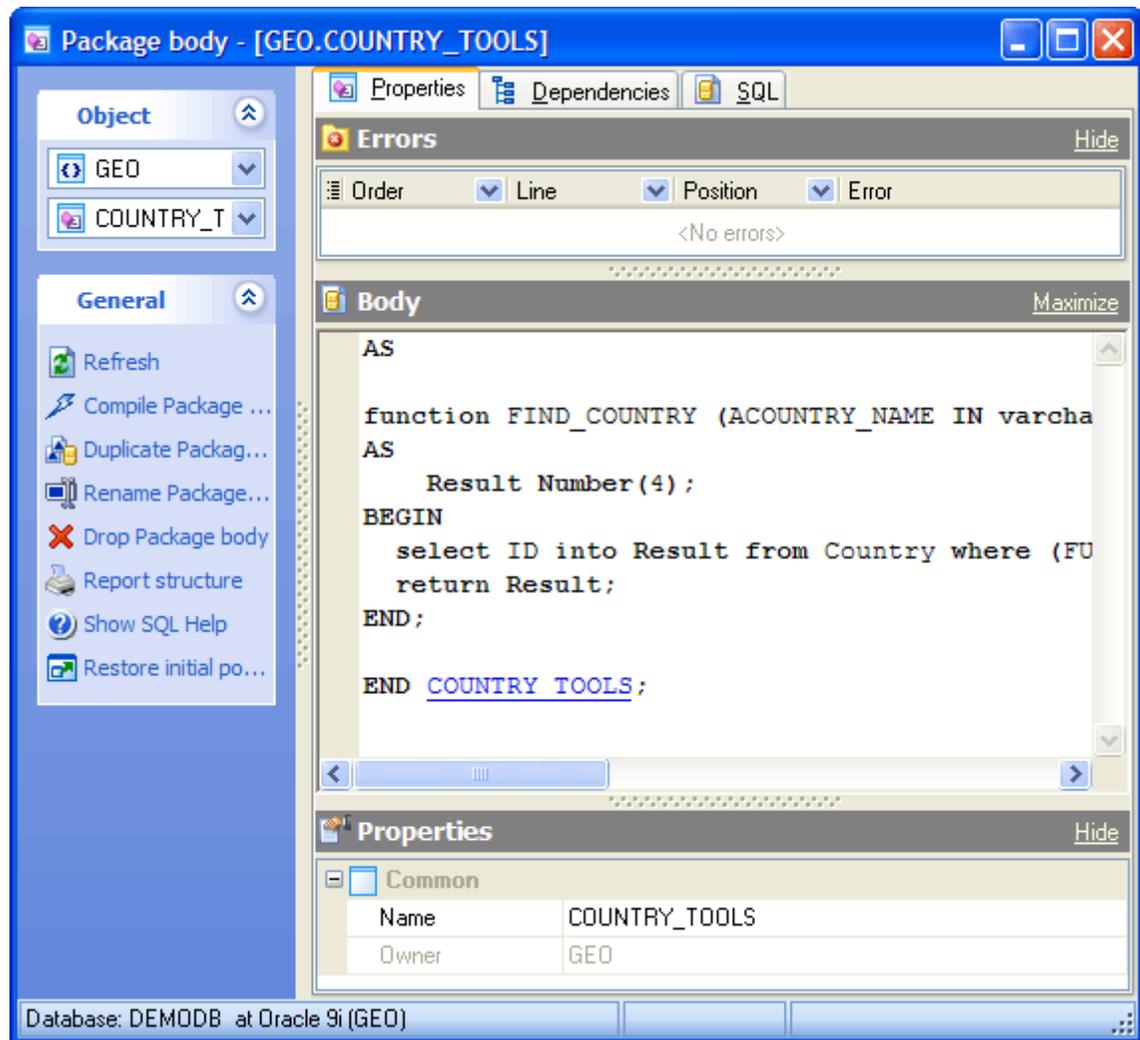


### 5.11.2 Package Body Editor

Package Body Editor can be opened automatically after the body is created and is available on editing (see [Editing Package bodies](#)<sup>[156]</sup> for details).

The basic principles of Object Editors in Oracle Maestro are explained in a [separate topic](#)<sup>[39]</sup>. Below you will find a description of editor tabs that are unique for the current object.

**See also:** [Create Package Body Wizard](#)<sup>[157]</sup>



The **Errors** tab displays all the necessary information about object errors. If an error have occurred during the object compilation it appears in the list with some additional properties: **Order** (one after another), **Line** and **Position** (object definition location the error was found out), **Error** (corresponding PL/SQL exception).

#### Body

The area contains the package body definition which is available to editing.

#### Name

The field allows to rename the package body.

#### Owner

Here you can find the package body owner.

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.

## 5.12 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>161</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>162</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.

## 5.12.1 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 Oracle Maestro are explained in a [separate topic](#)<sup>[37]</sup>. Below you will find a description of wizard steps that are unique for 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.

### 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

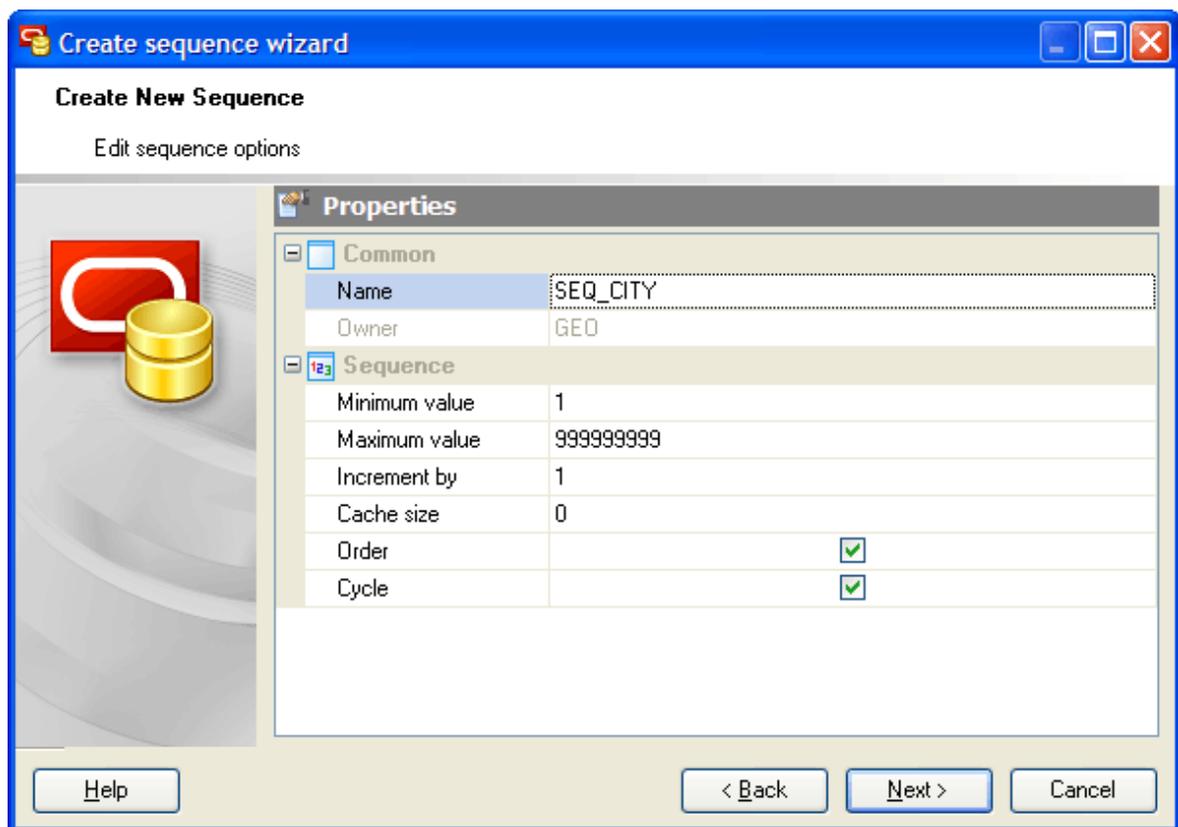
Specify how many sequence numbers are to be preallocated and stored in memory for faster access. The minimum value is 1 (only one value can be generated at a time, i.e., no cache), and this is also the default.

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

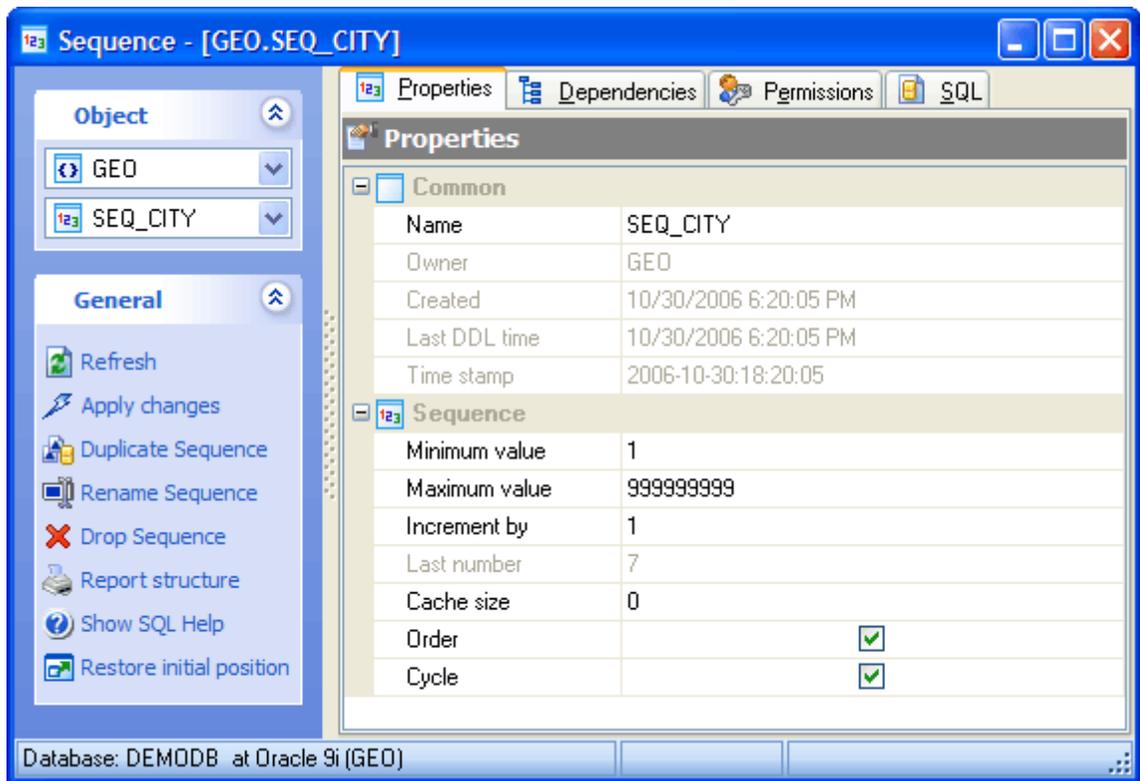
The CYCLE option allows the sequence to wrap around when the *maxvalue* or *minvalue* has been reached by an ascending or descending sequence respectively. If the limit is reached, the next number generated will be the minvalue or maxvalue, respectively.



### 5.12.2 Sequence Editor

Use [Sequence Editor](#) to change properties of existing sequences. The editor can be opened automatically after [the sequence is created](#)<sup>[16]</sup> or from the Explorer Tree and Object Manager.

The basic principles of Object Editors in Oracle Maestro are explained in a [separate topic](#)<sup>[3]</sup>. Below you will find a description of editor tabs that are unique for the current object.



#### Name

Here you can rename the sequence.

#### Owner

Shows the owner of the sequence.

#### Created

The field displays the date the object was created.

#### Last DDL time

Use the field to find the date when the last data definition language (DDL) operation was performed on the current object. The Last DDL time can help you to find if any changes to the object definitions have been made on or after a specific time.

#### 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.

## 5.13 Types

User-defined datatypes use Oracle built-in datatypes and other user-defined datatypes as the building blocks of object types that model the structure and behavior of data in applications. The sections that follow describe the various categories of user-defined types. The [Type body](#)<sup>[171]</sup> contains the code for the methods that implement the type.

### ■ How can I create a new type?

New types are created within [Create Type Wizard](#)<sup>[166]</sup>. In order to run the wizard you should either

- select the [Object | Create Database Object...](#) main menu item;
  - select the [Type](#) icon in the [Create Database Object](#) dialog
- or
- select the [Types](#) list or any object from that list in the explorer tree;
  - select the [Create New Type...](#) item from the popup menu
- or
- open the schema in [Schema Editor](#) and the [Types](#) tab there;
  - press the **Insert** key or select the [Create New 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 type?

Types can be edited within [Type Editor](#)<sup>[169]</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 Type ...](#) item from the popup menu
- or
- open the schema in [Schema Editor](#)<sup>[61]</sup> and the [Types](#) tab there;
  - select the type to edit;
  - press the **Enter** key or select the [Edit Type](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#)).

### ■ How can I drop a type?

To drop a type:

- select the type to drop in the explorer tree;

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

and confirm dropping in the dialog window.

### 5.13.1 Create Type Wizard

**Create Type Wizard** guides you through the process of creating a new schema type.

The basic principles of Create Object Wizards in Oracle Maestro are explained in a [separate topic](#)<sup>[37]</sup>. Below you will find a description of wizard steps that are unique for the current object.

- [Specifying type properties](#)<sup>[166]</sup>
- [Editing type subitems](#)<sup>[167]</sup>
- [Specifying type definition](#)<sup>[168]</sup>

#### 5.13.1.1 Specifying type properties

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

The screenshot shows the 'Create type wizard' dialog box with the title 'Create New Type' and subtitle 'Edit type options'. On the left is a graphic of a red link and gold coins. The main area is a 'Properties' table:

Properties	
[-] Common	
Name	EMPLOYEE_T
Owner	GEO
[-] Type	
Kind	Simple
Super type	
Instantiable	<input checked="" type="checkbox"/>
Final	<input type="checkbox"/>
Invoker rights	DEFINER
Type OID	

At the bottom are buttons: Help, < Back, Next >, and Cancel.

Owner

Defines the owner of the new type.

#### Instantiable

Use the `Instantiable` clause to indicate whether any object instances of this type can be constructed. Check to set that object instances of this type can be constructed. Otherwise, no default or user-defined constructor exists for this object type. You must specify these keywords for any type with Noninstantiable methods and for any type that has no attributes, either inherited or specified in this statement.

#### Final

At the top level of the syntax, these clauses specify the inheritance attributes of the type. Use the `Final` clause to indicate whether any further subtypes can be created for this type. Check the option if no further subtypes can be created for this type.

#### Invoker rights

The option lets you to specify whether the member functions and procedures of the object type execute with the privileges and in the schema of the user who owns the object type or with the privileges and in the schema of `CURRENT_USER`. This specification applies to the corresponding type body as well.

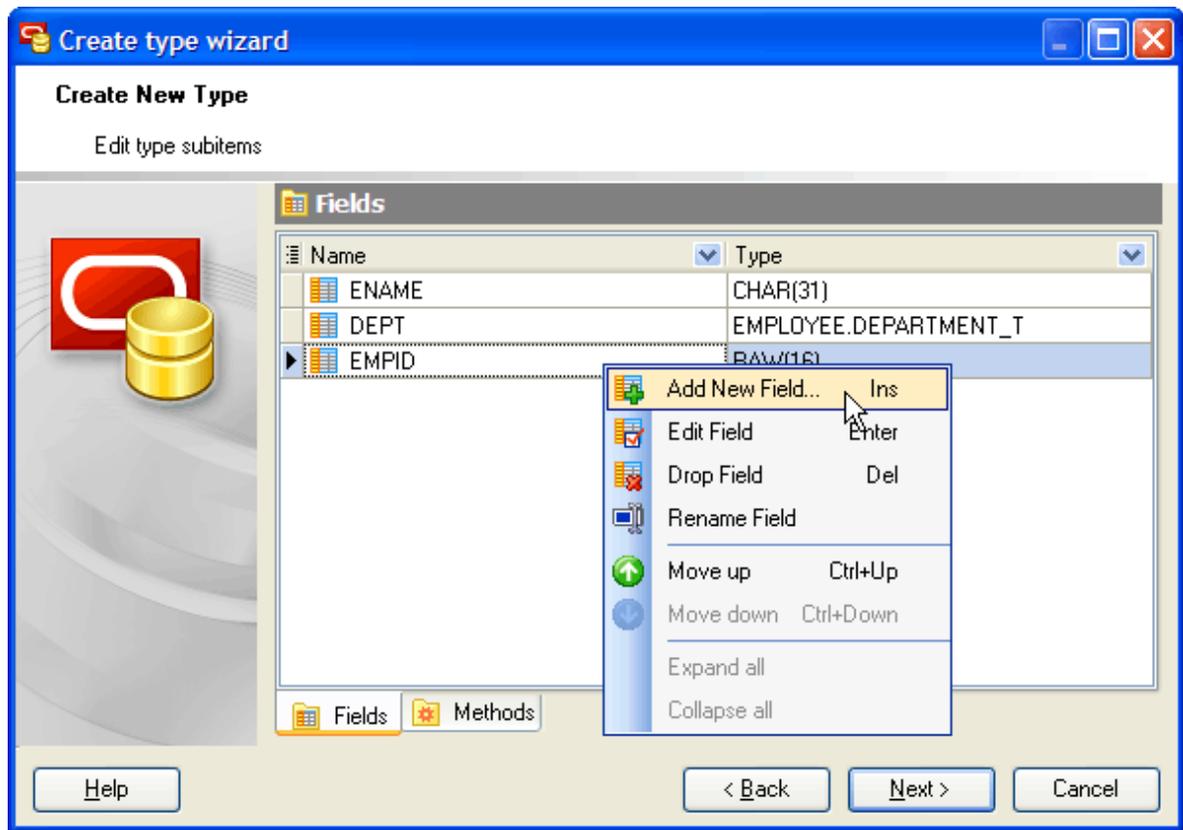
This clause also determines how Oracle Database resolves external names in queries, DML operations, and dynamic SQL statements in the member functions and procedures of the type.

#### Type OID

The `OID` clause is useful for establishing type equivalence of identical objects in more than one database.

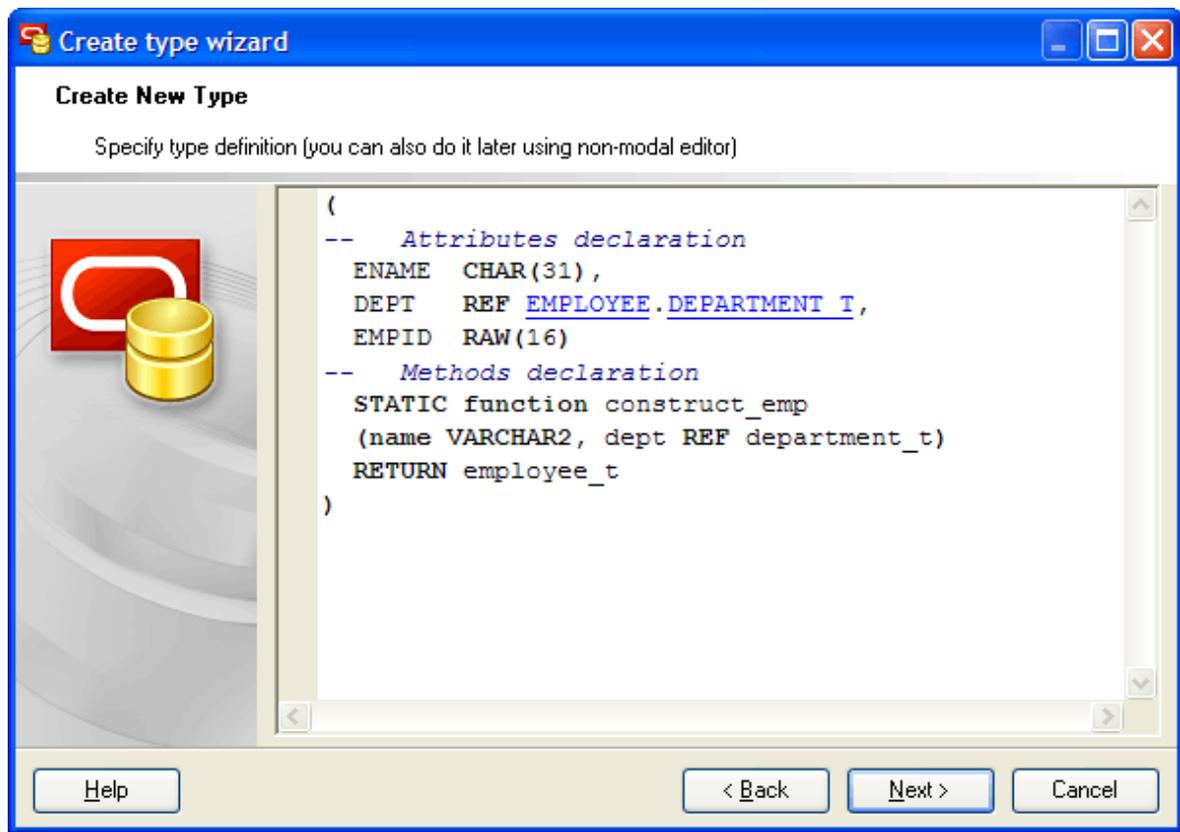
### 5.13.1.2 Setting type subitems

The wizard step allows to define type fields and methods. Just open the corresponding tab and manage corresponding objects within the popup menu.



### 5.13.1.3 Specifying type definition

Here you can specify the type definition. You can also do it later using non-modal editor.



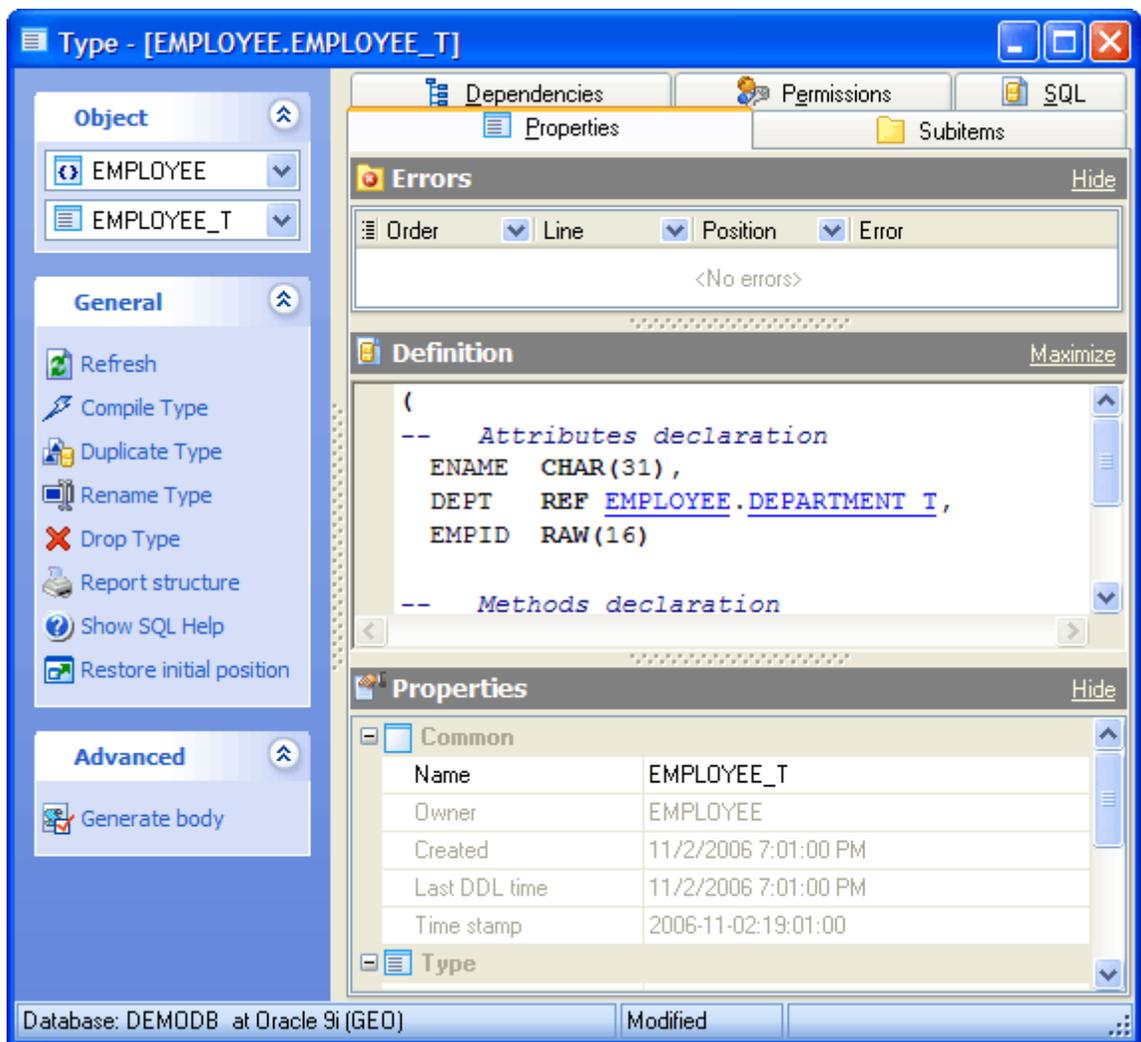
### 5.13.2 Type Editor

Type Editor is the basic Oracle Maestro tool for working with existing types. You can open a type in Type Editor from the Explorer Tree or Object Manager.

The basic principles of Object Editors in Oracle Maestro are explained in a [separate topic](#)<sup>[39]</sup>. Below you will find a description of editor tabs that are unique for the current object.

The description of type options you can find at the [corresponding topic](#)<sup>[170]</sup>

**See also:** [Create Type Wizard](#)<sup>[166]</sup>



### 5.13.2.1 Editing type options

Type Editor provides you with an ability to edit type properties. The **Properties** tab allows you to change the type name, the type owner, type definition, and other type properties. You can also find the OID there.

The **Errors** tab displays all the necessary information about object errors. If an error have occurred during the object compilation it appears in the list with some additional properties: **Order** (one after another), **Line** and **Position** (object definition location the error was found out), **Error** (corresponding PL/SQL exception).

**Subitems** tab allows to manage type fields and methods. Just open the corresponding grid and using popup menu add/edit/drop the necessary objects.

#### Name

Here you can change the type name.

#### Owner

The field contains the owner of the type.

#### Created

The field displays the date the object was created.

#### Last DDL time

Use the field to find the date when the last data definition language (DDL) operation was performed on the current object. The Last DDL time can help you to find if any changes to the object definitions have been made on or after a specific time.

#### Instantiable

Use the [Instantiable](#) clause to indicate whether any object instances of this type can be constructed. Check to set that object instances of this type can be constructed. Otherwise, no default or user-defined constructor exists for this object type. You must specify these keywords for any type with Noninstantiable methods and for any type that has no attributes, either inherited or specified in this statement.

#### Final

At the top level of the syntax, these clauses specify the inheritance attributes of the type. Use the [Final](#) clause to indicate whether any further subtypes can be created for this type. Check the option if no further subtypes can be created for this type.

#### Invoker rights

The option means whether the member functions and procedures of the object type execute with the privileges and in the schema of the user who owns the object type or with the privileges and in the schema of CURRENT\_USER. This clause also determines how Oracle Database resolves external names in queries, DML operations, and dynamic SQL statements in the member functions and procedures of the type.

#### Type OID

The OID clause is useful for establishing type equivalence of identical objects in more than one database.

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.

### 5.13.3 Type bodies

Type body contains the member methods defined in the object type specification

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

New Type bodies are created within [Create Type body Wizard](#)<sup>[172]</sup>. In order to run the wizard you should either

- select the [Object | Create Database Object...](#) main menu item;
  - select the [Type body](#) icon in the [Create Database Object](#) dialog
- or
- select the [Type bodies](#) list or any object from that list in the explorer tree;
  - select the [Create New Type body...](#) item from the popup menu
- or
- open the schema in [Schema Editor](#) and the [Type bodies](#) tab there;
  - press the **Insert** key or select the [Create New Type body](#) item

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

To create a new type body with the same properties as one of the existing type body has:

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

#### ■ **How can I edit an existing type body?**

Type bodies can be edited within [Type Body Editor](#)<sup>[174]</sup>. In order to run the editor you should either

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

or

- open the schema in [Schema Editor](#)<sup>[61]</sup> and the [Type bodies](#) tab there;
- select the type body to edit;
- press the **Enter** key or select the [Edit Type body](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#)).

#### ■ **How can I drop a type body?**

To drop a type body:

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

or

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

and confirm dropping in the dialog window.

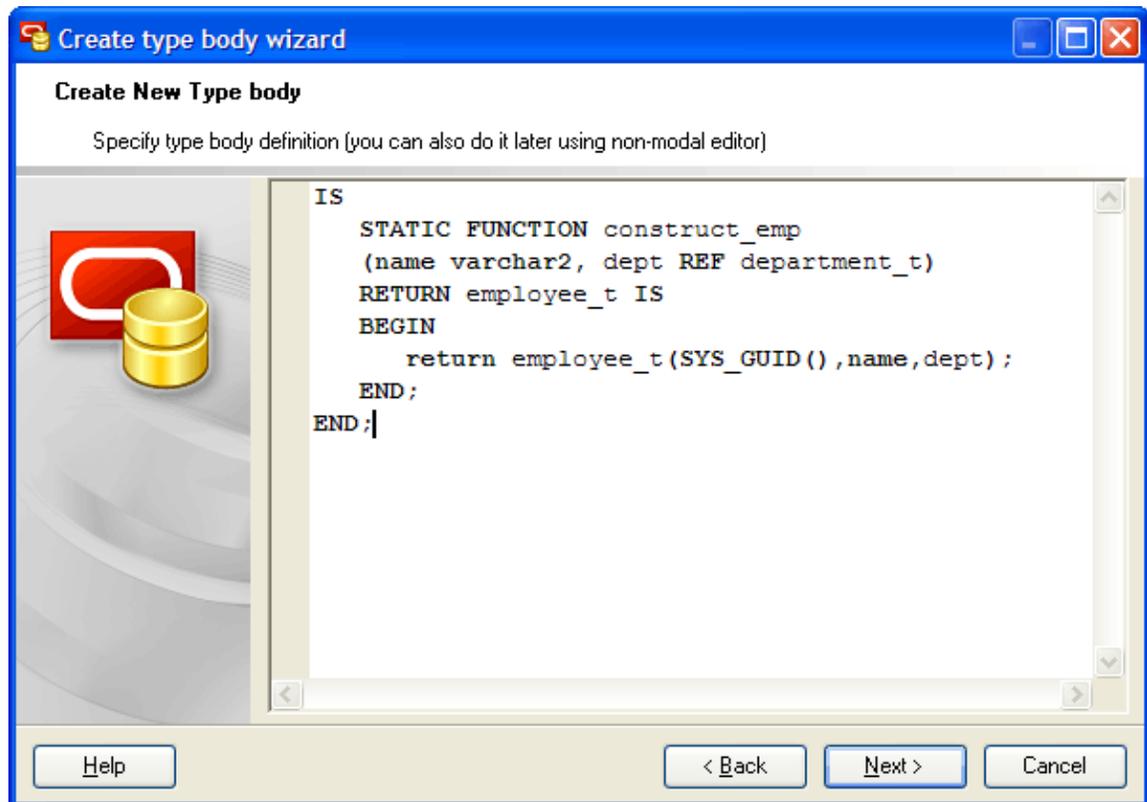
### 5.13.3.1 Create Type Body Wizard

[Create Type Body Wizard](#) guides you through the process of creating a new database type body. See [How To Create Type Body](#)<sup>[171]</sup> for instructions on running this wizard.

The basic principles of Create Object Wizards in Oracle Maestro are explained in a [separate topic](#)<sup>[37]</sup>. Below you will find a description of wizard steps that are unique for the current object.

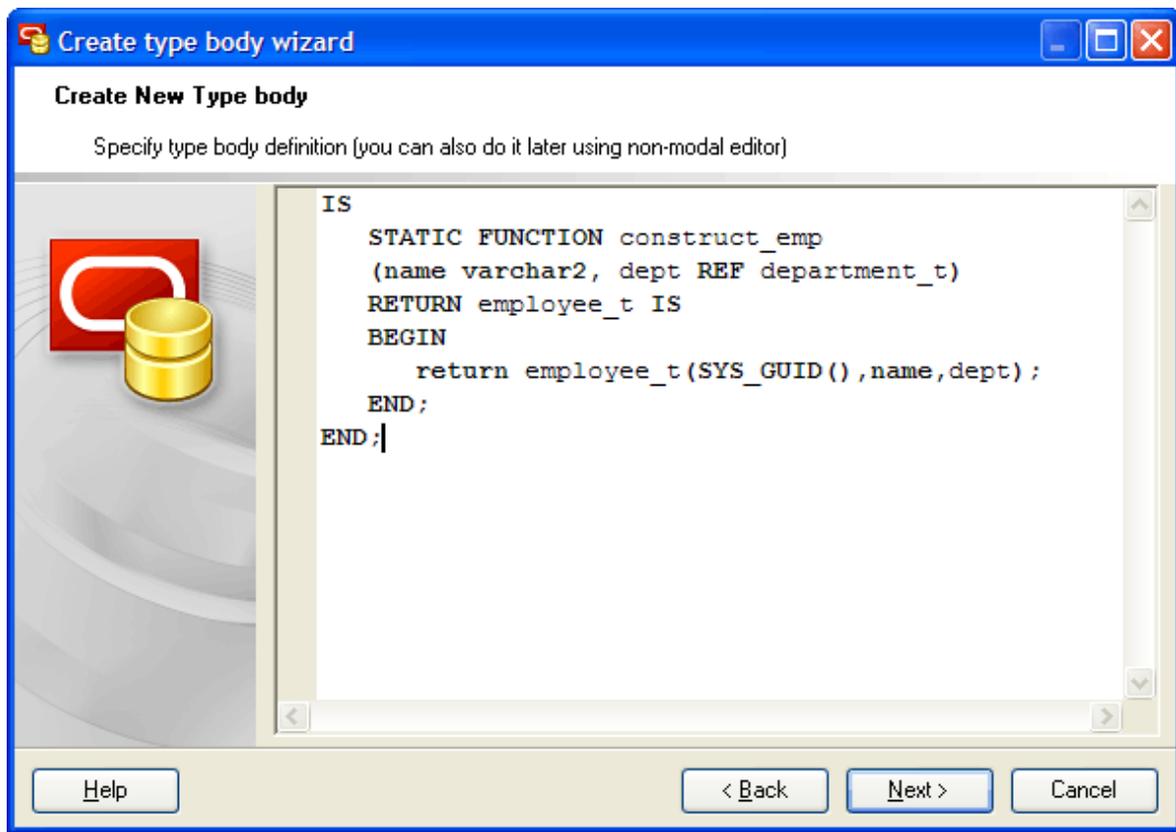
- [Specifying body definition](#)<sup>[152]</sup>

See also: [Type Body Editor](#)<sup>[174]</sup>



#### 5.13.3.1.1 Specifying body definition

Here you can specify the type body definition. You can also do it later using non-modal editor.



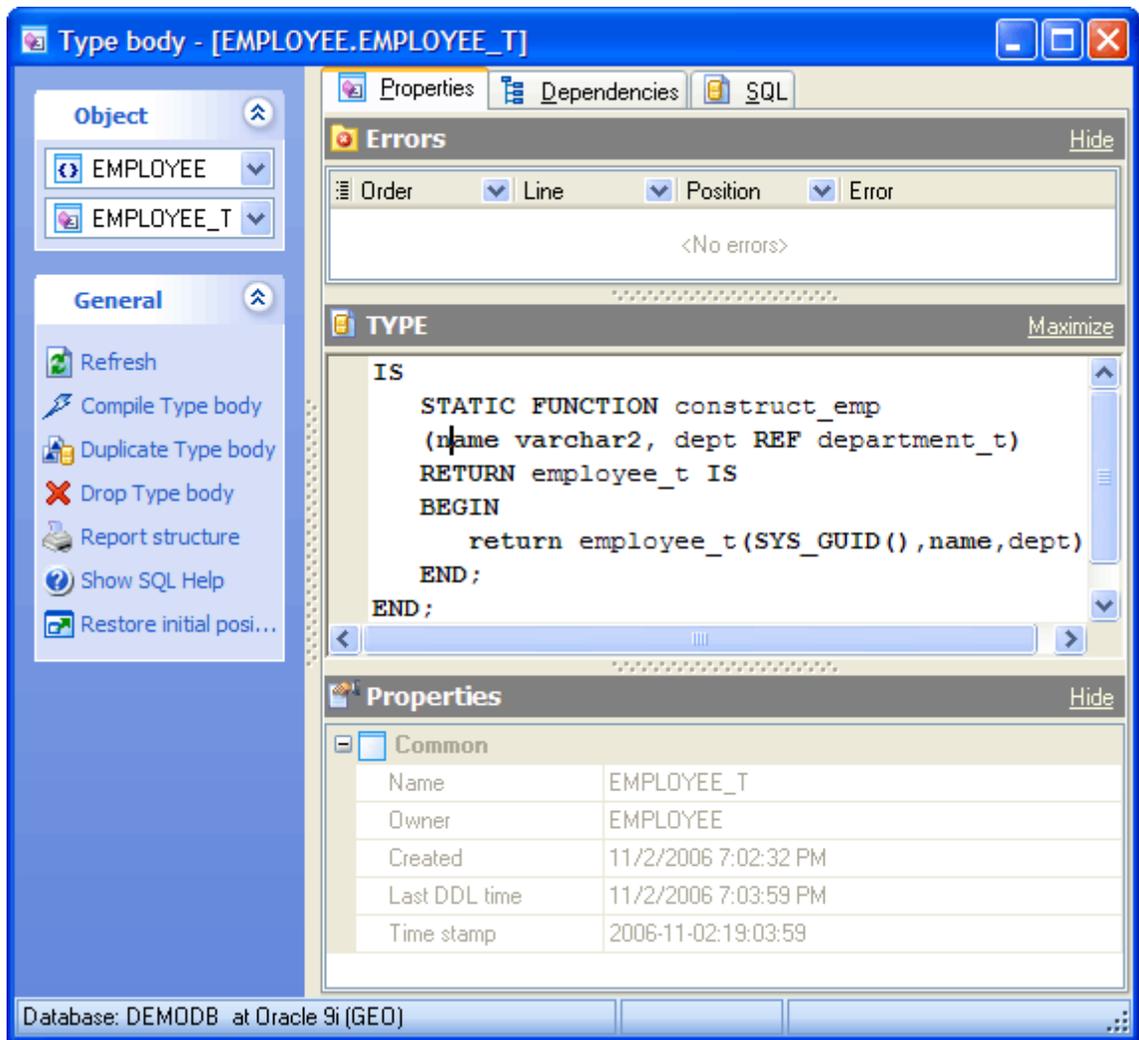
### 5.13.3.2 Type Body Editor

Type Body Editor can be opened automatically after the body is created and is available on editing (see [Editing Type bodies](#)<sup>[171]</sup> for details).

The basic principles of Object Editors in Oracle Maestro are explained in a [separate topic](#)<sup>[39]</sup>. Below you will find a description of editor tabs that are unique for the current object.

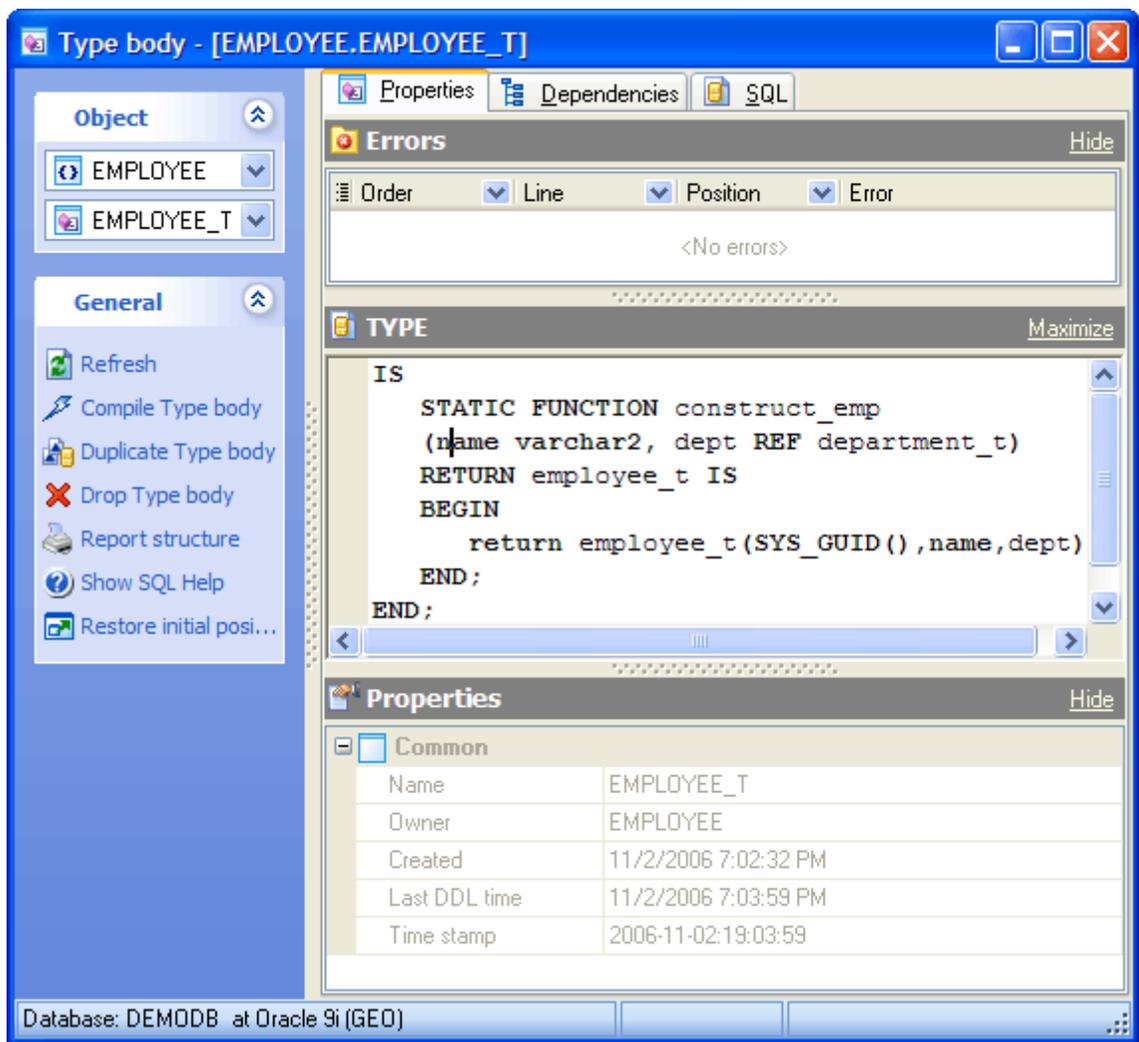
- [Editing type body properties](#)<sup>[175]</sup>

**See also:** [Create Type Body Wizard](#)<sup>[172]</sup>



#### 5.13.3.2.1 Editing type body properties

Type Body Editor provides you with an ability to edit body properties. The detailed description of the properties you can find below.



The **Errors** tab displays all the necessary information about object errors. If an error have occurred during the object compilation it appears in the list with some additional properties: **Order** (one after another), **Line** and **Position** (object definition location the error was found out), **Error** (corresponding PL/SQL exception).

#### Type

The area contains the type definition which is available for editing.

#### Name

The field represents the new package body name.

#### Owner

The field contains the owner of the type body.

#### Created

The field displays the date the object was created.

#### Last DDL time

Use the field to find the date when the last data definition language (DDL) operation was performed on the current object. The Last DDL time can help you to find if any changes to the object definitions have been made on or after a specific time.

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.

## 5.14 Users

A database user is an account through which you can log in to the database, and to establish the means by which Oracle permits access by the user.

### 5.14.1 Create User Wizard

[Create User Wizard](#) guides you through the process of creating a new database user. See [How To Create User](#)<sup>[178]</sup> to learn how to run the wizard.

The basic principles of Create Object Wizards in Oracle Maestro are explained in a [separate topic](#)<sup>[37]</sup>. Below you will find a description of wizard steps that are unique for the current object.

#### Name

The field contains the new user name as it was set on the previous step.

#### Identified type (Externally, By password, Globally)

The clause lets you indicate how Oracle Database authenticates the user. Select [By password](#) clause to create a local user and indicate that the user must specify password to log on to the database. Specify [Externally](#) to create an external user. Such a user must be authenticated by an external service, such as an operating system or a third-party service. In this case, Oracle Database relies on the login authentication of the operating system to ensure that a specific operating system user has access to a specific database user. Choose [Globally](#) to create a global user. Such a user must be authenticated by the enterprise directory service.

#### Global name

The field is available if Globally was set as Identified type. The name string can take one of two forms:

- The X.509 name at the enterprise directory service that identifies this user. It should be of the form CN=username,other\_attributes, where other\_attributes is the rest of the user's distinguished name (DN) in the directory.
- A null string ( ' ') indicating that the enterprise directory service will map authenticated global users to the appropriate database schema with the appropriate roles.

#### Profile

Specify the profile you want to assign to the user. The profile limits the amount of database resources the user can use. If you omit this clause, then Oracle Database assigns the DEFAULT profile to the user.

#### Default tablespace

Specify the default tablespace for objects that the user creates. If you omit this clause, then the user's objects are stored in the database default tablespace. If no default tablespace has been specified for the database, then the user's objects are stored in the SYSTEM tablespace.

#### Temporary tablespace

Specify the tablespace or tablespace group for the user's temporary segments. If you omit this clause, then the user's temporary segments are stored in the database default temporary tablespace or, if none has been specified, in the SYSTEM tablespace.

#### Is password expired

Specify the option if you want the user's password to expire. This setting forces the user or the DBA to change the password before the user can log in to the database.

#### Is account locked

Check the box to lock the user's account and disable access. Specify ACCOUNT UNLOCK to unlock the user's account and enable access to the account.

#### Password for Authentication

Passwords can contain only single-byte characters from your database character set regardless of whether the character set also contains multibyte characters.

#### Specifying user's role membership

This wizard step allows you to specify role's privileges, quotas and system privileges for the new user. For this purpose go to the appropriate tab and check necessary boxes.

#### Quotas

Use this clause to allow the user to allocate up to integer bytes of space in the tablespace. This quota is the maximum space in the tablespace the user can allocate.

#### Tablespace

Select from drop-down list the tablespace for the quotation.

#### Quota (Unlimited, 100M, 200M, 400M)

Set here the maximum space in the tablespace the user can allocate. [Unlimited](#) lets the user allocate space in the tablespace without bound.

**Note:** To create a user you must have the CREATE USER system privilege.

**See also:** [User Editor](#)<sup>[179]</sup>

## 5.14.2 User Editor

[User Editor](#) allows you to edit user properties, permissions, grants, user quotas and manage user's objects (see [How to edit database user](#)<sup>[178]</sup> for details).

The basic principles of Object Editors in Oracle Maestro are explained in a [separate topic](#)<sup>[39]</sup>. Below you will find a description of editor tabs that are unique for the current object.

This editor consists of several tabs: the [Properties](#)<sup>[179]</sup> tab allows to browse user options and the [Objects](#) tab allows to browse database objects owned by the user.

**See also:** [Create User Wizard](#)<sup>[178]</sup>

### 5.14.2.1 Editing user properties

The [Properties](#) tab allows you to change the user name, the user password, the role list the user belongs to, user quotas and user's system privileges.

#### Name

The field allows you to rename the database user.

### Created

The field displays the date the object was created.

### Last DDL time

Use the field to find the date when the last data definition language (DDL) operation was performed on the current object. The Last DDL time can help you to find if any changes to the object definitions have been made on or after a specific time.

### Identified type (Externally, By password, Globally)

The clause lets you indicate how Oracle Database authenticates the user. Select **By password** clause to create a local user and indicate that the user must specify password to log on to the database. Specify **Externally** to create an external user. Such a user must be authenticated by an external service, such as an operating system or a third-party service. In this case, Oracle Database relies on the login authentication of the operating system to ensure that a specific operating system user has access to a specific database user. Choose **Globally** to create a global user. Such a user must be authenticated by the enterprise directory service.

### Global name

The field is available if Globally was set as Identified type. The name string can take one of two forms:

- The X.509 name at the enterprise directory service that identifies this user. It should be of the form CN=username,other\_attributes, where other\_attributes is the rest of the user's distinguished name (DN) in the directory.
- A null string ( ' ') indicating that the enterprise directory service will map authenticated global users to the appropriate database schema with the appropriate roles.

### Profile

Specify the profile you want to assign to the user. The profile limits the amount of database resources the user can use. If you omit this clause, then Oracle Database assigns the DEFAULT profile to the user.

### Default tablespace

Specify the default tablespace for objects that the user creates. If you omit this clause, then the user's objects are stored in the database default tablespace. If no default tablespace has been specified for the database, then the user's objects are stored in the SYSTEM tablespace.

### Temporary tablespace

Specify the tablespace or tablespace group for the user's temporary segments. If you omit this clause, then the user's temporary segments are stored in the database default temporary tablespace or, if none has been specified, in the SYSTEM tablespace.

### Is password expired

Specify the option if you want the user's password to expire. This setting forces the user or the DBA to change the password before the user can log in to the database.

### Is account locked

Check the box to lock the user's account and disable access. Specify ACCOUNT UNLOCK to unlock the user's account and enable access to the account.

### Password for Authentication

Passwords can contain only single-byte characters from your database character set regardless of whether the character set also contains multibyte characters.

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.

## 5.15 User profiles

User profile is a set of limits on database resources. If you assign the profile to a user, then that user cannot exceed these limits. Use profiles to limit the database resources available to a user for a single call or a single session.

### ■ How can I create a new user profile?

New user profile are created within [Create User Profile Wizard](#).<sup>[183]</sup> In order to run the wizard you should either

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

To create a new user profile with the same properties as one of the existing user profile has:

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

### ■ How can I edit an existing user profile?

User profiles can be edited within [User Profile Editor](#).<sup>[186]</sup> In order to run the editor you should either

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

You can change the name of the user profile using the [Rename User Profile](#) dialog. To open the dialog you should either

- select the [user profile](#) to rename in the explorer tree;
- select the [Rename User Profile](#) item from the popup menu

or

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

#### ■ **How can I drop a user profile?**

To drop a user profile:

- select the [user profile](#) to drop in the explorer tree;
- select the [Drop User Profile](#) item from the popup menu

or

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

and confirm dropping in the dialog window.

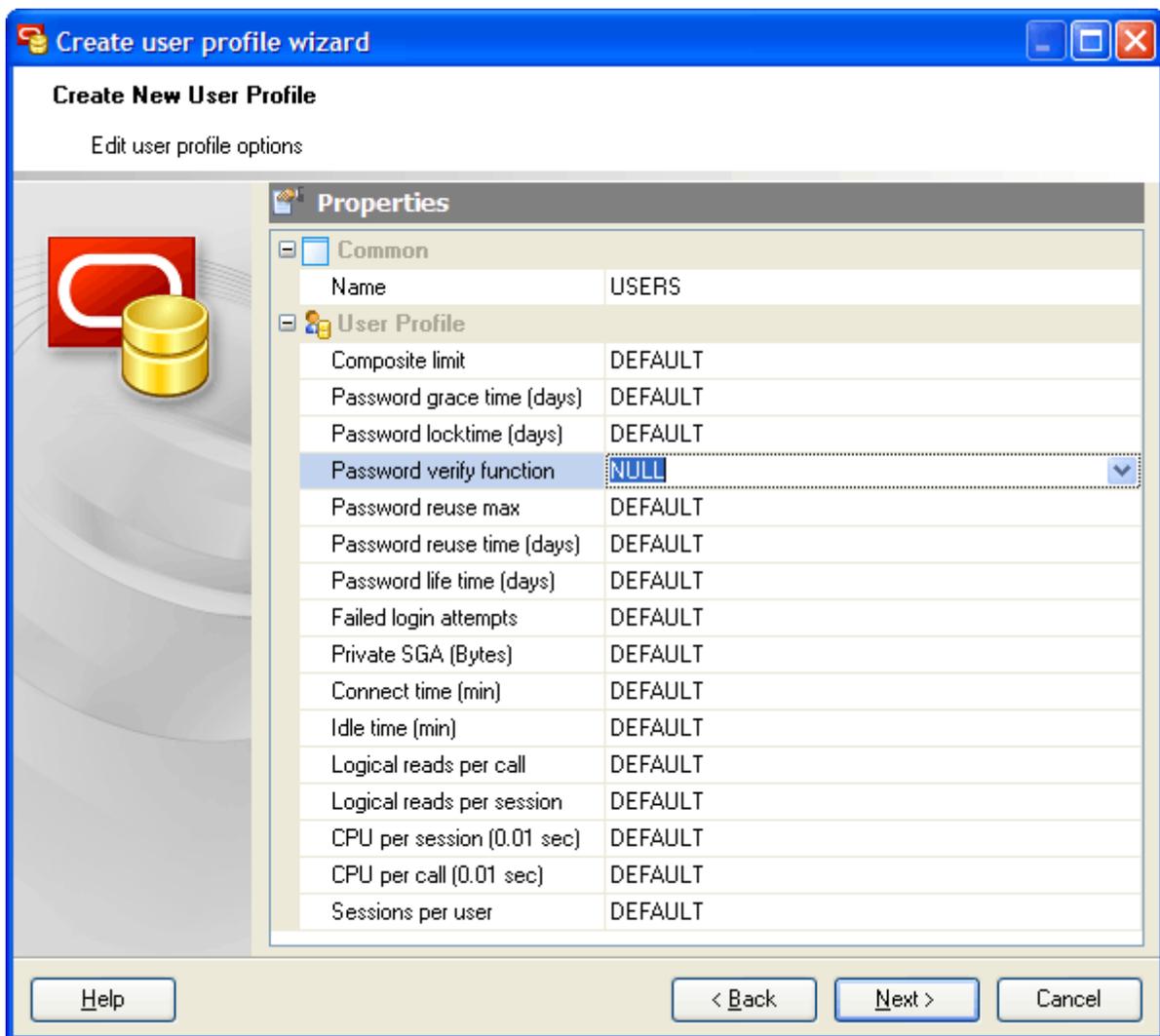
### 5.15.1 Create User Profile Wizard

[Create User Profile Wizard](#) guides you through the process of creating a new database user profile. See [How To Create User Profile](#)<sup>[182]</sup> for instructions on running this wizard.

The basic principles of Create Object Wizards in Oracle Maestro are explained in a [separate topic](#)<sup>[37]</sup>. Below you will find a description of wizard steps that are unique for the current object.

- [Specifying profile properties](#)<sup>[184]</sup>

**See also:** [User Profile Editor](#)<sup>[186]</sup>



### 5.15.1.1 Specifying profile properties

The wizard step was supplied to define common user profile properties. The detailed description of the properties you can find below.

#### Name

The field contains the editable new user profile name as it was set on the previous wizard step.

#### Composite limit

Specify the total resource cost for a session, expressed in service units. Oracle Database calculates the total service units as a weighted sum of CPU per session, Connect time, Logical reads per session, and Private SGA.

#### Password grace time (days)

Specify the number of days after the grace period begins during which a warning is issued and login is allowed. If the password is not changed during the grace period, the password expires.

#### Password lock time (days)

Specify the number of days an account will be locked after the specified number of consecutive failed login attempts.

#### Password verify function

The clause lets a PL/SQL password complexity verification script be passed as an argument to the CREATE PROFILE statement. Oracle Database provides a default script, but you can create your own routine or use third-party software instead.

#### Password reuse max

The clause specifies the number of password changes required before the current password can be reused. For these parameter to have any effect, you must specify an integer for both of them.

#### Password reuse time (days)

The option specifies the number of days before which a password cannot be reused.

#### Password life time (days)

Specify the number of days the same password can be used for authentication. If you also set a value for [Password grace time](#), the password expires if it is not changed within the grace period, and further connections are rejected. If you do not set a value for [Password grace time](#), its default of [Unlimited](#) will cause the database to issue a warning but let the user continue to connect indefinitely.

#### Failed login attempts

Specify the number of failed attempts to log in to the user account before the account is locked.

#### Private SGA (Bytes)

Specify the amount of private space a session can allocate in the shared pool of the system global area (SGA), expressed in bytes.

#### Connect time (min)

Specify the total elapsed time limit for a session, expressed in minutes.

#### Idle time (min)

Specify the permitted periods of continuous inactive time during a session, expressed in minutes. Long-running queries and other operations are not subject to this limit.

#### Logical reads per call

Specify the permitted the number of data blocks read for a call to process a SQL statement (a parse, execute, or fetch).

#### Logical reads per session

Specify the permitted number of data blocks read in a session, including blocks read from memory and disk.

#### CPU per session (0.01 sec)

Specify the CPU time limit for a session, expressed in hundredth of seconds.

#### CPU per call (0.01 sec)

Specify the CPU time limit for a call (a parse, execute, or fetch), expressed in hundredths of seconds.

Sessions per user

Specify the number of concurrent sessions to which you want to limit the user.

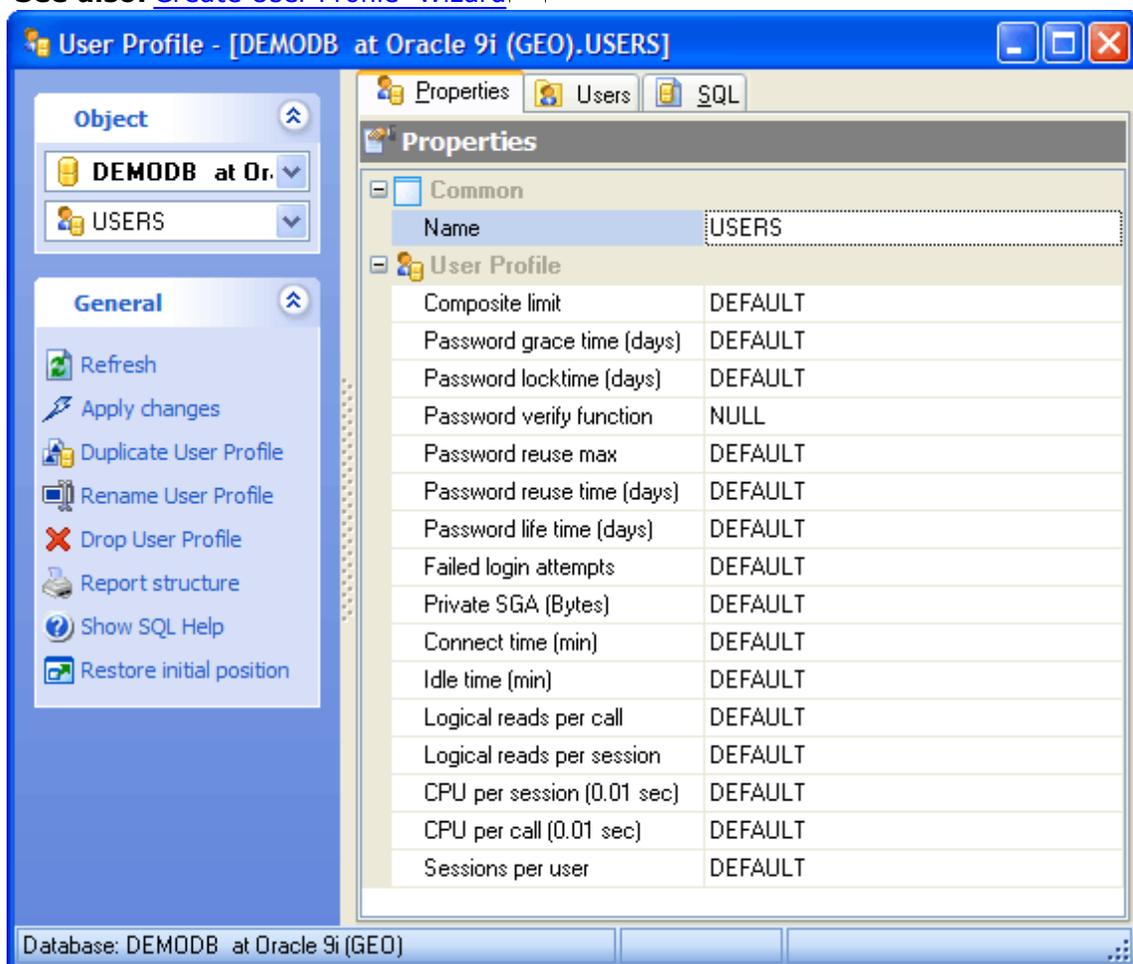
## 5.15.2 User Profile Editor

User Profile Editor can be opened automatically after the user is created and is available on editing (see [Editing User Profiles](#) <sup>182</sup> for details).

The basic principles of Object Editors in Oracle Maestro are explained in a [separate topic](#) <sup>39</sup>. Below you will find a description of editor tabs that are unique for the current object.

- [Editing profile properties](#) <sup>186</sup>
- [Managing users](#) <sup>188</sup>

**See also:** [Create User Profile Wizard](#) <sup>183</sup>



### 5.15.2.1 Editing profile properties

User Profile Editor provides you with an ability to edit profile properties. The Properties tab allows you to change the user password and session options.

Name

The field contains the editable new user profile name as it was set on the previous

wizard step.

#### Composite limit

Specify the total resource cost for a session, expressed in service units. Oracle Database calculates the total service units as a weighted sum of [CPU per session](#), [Connect time](#), [Logical reads per session](#), and [Private SGA](#).

#### Password grace time (days)

Specify the number of days after the grace period begins during which a warning is issued and login is allowed. If the password is not changed during the grace period, the password expires.

#### Password lock time (days)

Specify the number of days an account will be locked after the specified number of consecutive failed login attempts.

#### Password verify function

The clause lets a PL/SQL password complexity verification script be passed as an argument to the CREATE PROFILE statement. Oracle Database provides a default script, but you can create your own routine or use third-party software instead.

#### Password reuse max

The clause specifies the number of password changes required before the current password can be reused. For these parameter to have any effect, you must specify an integer for both of them.

#### Password reuse time (days)

The option specifies the number of days before which a password cannot be reused.

#### Password life time (days)

Specify the number of days the same password can be used for authentication. If you also set a value for [Password grace time](#), the password expires if it is not changed within the grace period, and further connections are rejected. If you do not set a value for [Password grace time](#), its default of [Unlimited](#) will cause the database to issue a warning but let the user continue to connect indefinitely.

#### Failed login attempts

Specify the number of failed attempts to log in to the user account before the account is locked.

#### Private SGA (Bytes)

Specify the amount of private space a session can allocate in the shared pool of the system global area (SGA), expressed in bytes.

#### Connect time (min)

Specify the total elapsed time limit for a session, expressed in minutes.

#### Idle time (min)

Specify the permitted periods of continuous inactive time during a session, expressed in minutes. Long-running queries and other operations are not subject to this limit.

#### Logical reads per call

Specify the permitted the number of data blocks read for a call to process a SQL

statement (a parse, execute, or fetch).

#### Logical reads per session

Specify the permitted number of data blocks read in a session, including blocks read from memory and disk.

#### CPU per session (0.01 sec)

Specify the CPU time limit for a session, expressed in hundredth of seconds.

#### CPU per call (0.01 sec)

Specify the CPU time limit for a call (a parse, execute, or fetch), expressed in hundredths of seconds.

#### Sessions per user

Specify the number of concurrent sessions to which you want to limit the user.

#### Created

The field displays the date the object was created.

#### Last DDL time

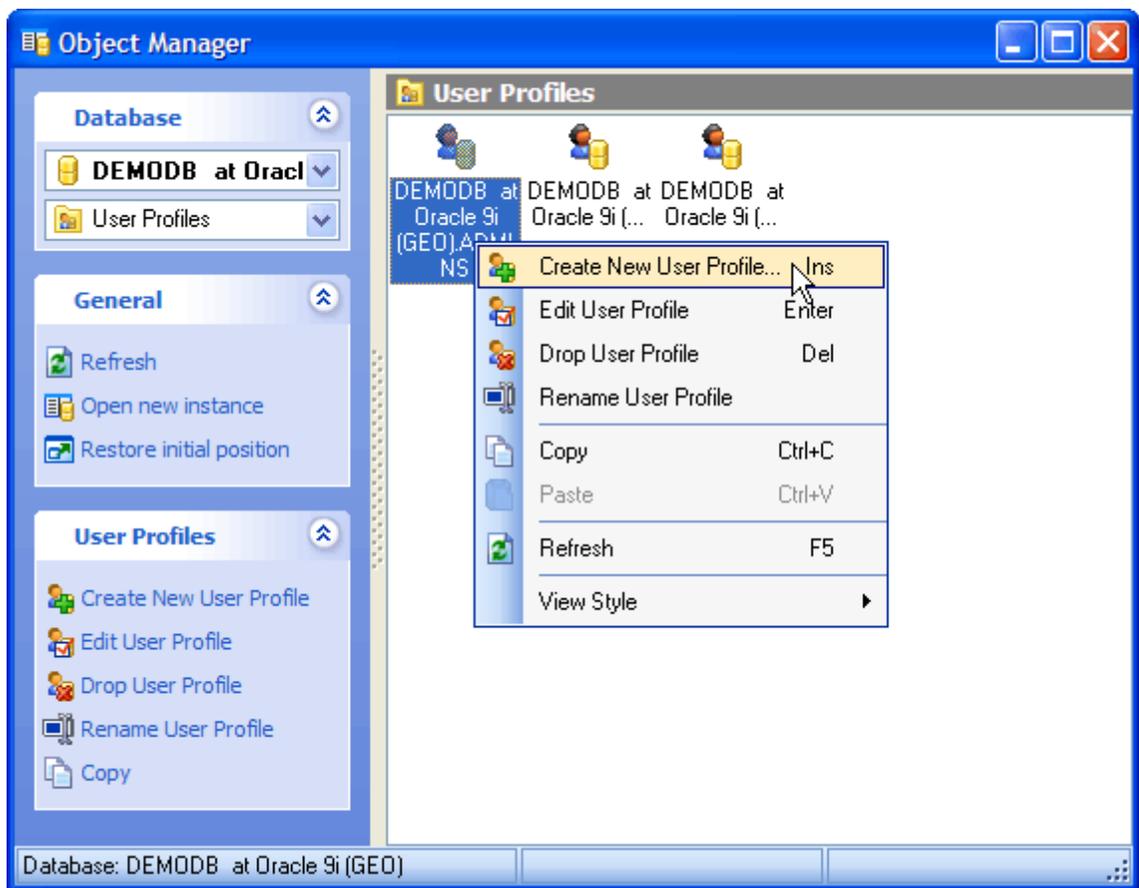
Use the field to find the date when the last data definition language (DDL) operation was performed on the current object. The Last DDL time can help you to find if any changes to the object definitions have been made on or after a specific time.

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.

### 5.15.2.2 Managing users

The [Users](#) tab allows you to view all users which are belonged to the current profile. Using pop-up menu you can drop necessary user or open its editor.



## 5.16 Roles

Using roles can simplify security administration in databases with a large number of users or with a complex security system. Role is a set of privileges that can be granted to users or to other roles. You can use roles to administer database privileges. You can add privileges to a role and then grant the role to a user. The user can then enable the role and exercise the privileges granted by the role.

### ■ How can I create a new role?

New roles are created within [Create Role Wizard](#)<sup>[191]</sup>. In order to run the wizard you should either

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

To create a new role with the same properties as one of the existing roles has:

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

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

Roles can be edited within [Role Editor](#)<sup>[192]</sup>. In order to run the editor you should either

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

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

- select the role to rename in the explorer tree;

- select the [Rename Role](#) item from the popup menu

or

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

#### ■ **How can I drop a role?**

To drop a role (note that you can drop database roles only):

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

or

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

and confirm dropping in the dialog window.

### 5.16.1 Create Role Wizard

[Create Role Wizard](#) guides you through the process of creating a new database role. See [How To Create role](#)<sup>[190]</sup> to learn how to run this wizard.

The basic principles of Create Object Wizards in Oracle Maestro are explained in a [separate topic](#)<sup>[37]</sup>. Below you will find a description of wizard steps that are unique for the current object.

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

#### Name

The field contains new role name as it was set on the previous step.

#### Identity type (By password, Externally, Globally, Using package, Not identified)

The role option indicates the specified method a user must be authorized by.

- The [By password](#) clause lets you create a local role and indicates that the user must specify the password to the database when enabling the role. The password can contain only single-byte characters from your database character set regardless of whether this character set also contains multibyte characters.
- Specify [Externally](#) to create an external role. An external user must be authorized by an external service, such as an operating system or third-party service, before enabling the role.
- Specify [Globally](#) to create a global role. A global user must be authorized to use the role by the enterprise directory service before the role is enabled, or at login.
- The [Using package](#) clause lets you create an application role, which is a role that can be enabled only by applications using an authorized package.
- Specify [Not identified](#) to indicate that this role is authorized by the database and that no password is required to enable the role.

#### Package name

Set the package for Using package roles. If you do not specify schema, then the database assumes the package is in your own schema.

#### Password

Specifies the password that database users will use to activate the role.

#### Managing role members

The wizard step allows to define users and roles to be the new role members, to add the role being created to another database roles, and to grant new role system privileges. Just open the corresponding tab and check the appropriate boxes.

### 5.16.2 Role Editor

[Role Editor](#) allows you to edit role properties and permissions. It can be open automatically after the role is created and is available on editing (see [How to edit role](#) for details).

The basic principles of Object Editors in Oracle Maestro are explained in a [separate topic](#). Below you will find a description of editor tabs that are unique for the current object.

#### Name

The field allows you to view and modify the role name.

#### Identity type (By password, Externally, Globally, Using package, Not identified)

The role option indicates the specified method a user must be authorized by.

- The [By password](#) clause lets you create a local role and indicates that the user must specify the password to the database when enabling the role. The password can contain only single-byte characters from your database character set regardless of whether this character set also contains multibyte characters.
- Specify [Externally](#) to create an external role. An external user must be authorized by an external service, such as an operating system or third-party service, before enabling the role.
- Specify [Globally](#) to create a global role. A global user must be authorized to use the role by the enterprise directory service before the role is enabled, or at login.
- The [Using package](#) clause lets you create an application role, which is a role that can be enabled only by applications using an authorized package.
- Specify [Not identified](#) to indicate that this role is authorized by the database and that no password is required to enable the role.

#### Package name

Set the package for Using package roles. If you do not specify schema, then the database assumes the package is in your own schema.

#### Password

Specifies the password that database users will use to activate the application role.

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.

## 5.17 Collections

**Collection** is an ordered group of elements, all of the same type (for example, the grades for a class of students). Each element has a unique subscript that determines its position in the collection. PL/SQL offers three kinds of collections: associative arrays, nested tables, and varrays (short for variable-size arrays). Nested tables extend the functionality of associative arrays (formerly called "PL/SQL tables" or "index-by tables").

Collections work like the arrays found in most third-generation programming languages. Collections can have only one dimension. Most collections are indexed by integers, although associative arrays can also be indexed by strings. To model multi-dimensional arrays, you can declare collections whose items are other collections.

Nested tables and varrays can store instances of an object type and, conversely, can be attributes of an object type. Collections can also be passed as parameters. You can use them to move columns of data into and out of database tables or between client-side applications and stored subprograms.

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

New collections are created within [Create Collection Wizard](#)<sup>[194]</sup>. In order to run the wizard you should either

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

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

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

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

Collections can be edited within [Collection Editor](#)<sup>[196]</sup>. In order to run the editor you should either

- select the **collection** for editing in the explorer tree (type the first letters of the collection name for quick search);
  - select the **Edit Collection ...** item from the popup menu
- or

- open the schema in [Schema Editor](#)<sup>[61]</sup> and the [Collection](#) tab there;
- select the collection to edit;
- press the **Enter** key or select the [Edit Collection](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#)).

#### ■ **How can I drop a collection?**

To drop a collection:

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

or

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

and confirm dropping in the dialog window.

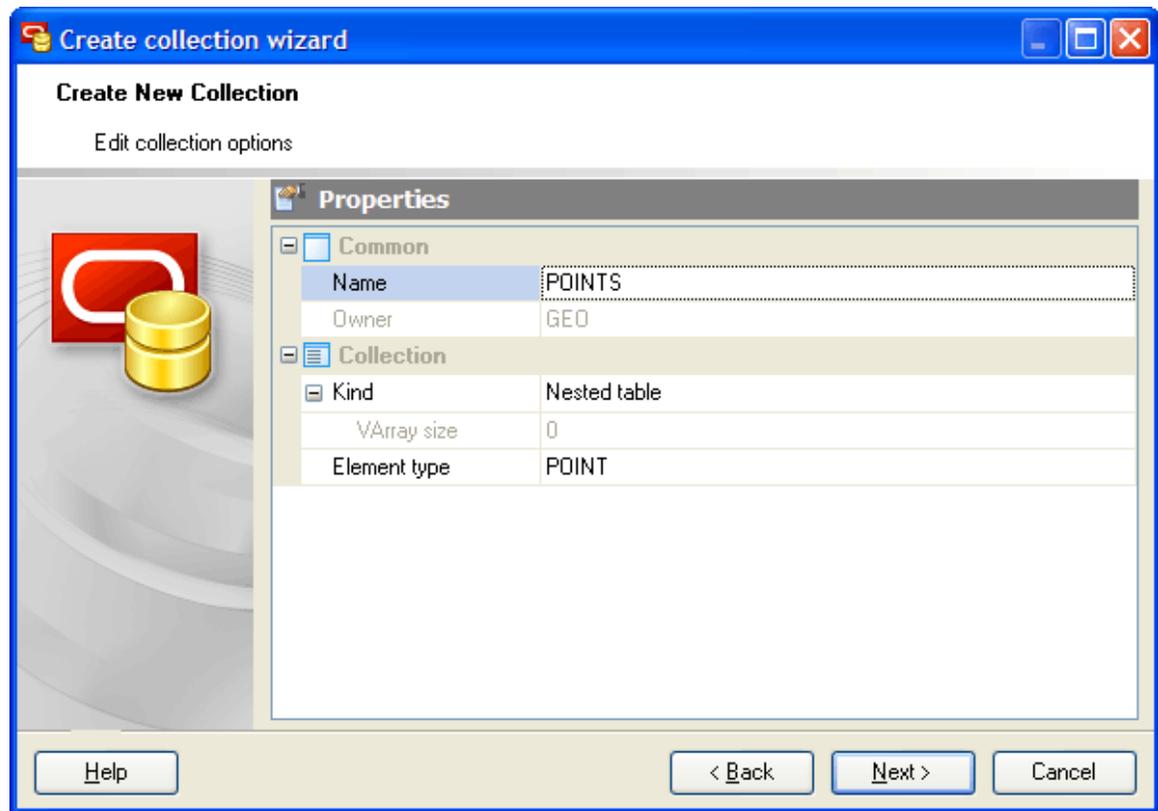
### 5.17.1 Create Collection Wizard

[Create Collection Wizard](#) guides you through the process of creating a new database collection. See [How To Create Collection](#)<sup>[193]</sup> for instructions on running this wizard.

The basic principles of Create Object Wizards in Oracle Maestro are explained in a [separate topic](#)<sup>[37]</sup>. Below you will find a description of wizard steps that are unique for the current object.

- [Setting collection properties](#)<sup>[195]</sup>

**See also:** [Collection Editor](#)<sup>[196]</sup>



### 5.17.1.1 Setting collection properties

This step is provided for specifying the new collection properties in the most convenient way.

#### Name

The field represents the new collection name as it was set on the previous wizard step.

#### Owner

The field displays the owner of the collection.

#### Kind (Varying array, Nested table)

**Nested table** hold an arbitrary number of elements. They use sequential numbers as subscripts. You can define equivalent SQL types, allowing nested tables to be stored in database tables and manipulated through SQL.

**Varying array** (short for variable-size arrays) hold a fixed number of elements (although you can change the number of elements at runtime). They use sequential numbers as subscripts. You can define equivalent SQL types, allowing varrays to be stored in database tables. They can be stored and retrieved through SQL, but with less flexibility than nested tables.

#### VArray Size

A varray has a maximum size, which you specify in its type definition. Its index has a fixed lower bound of 1 and an extensible upper bound. For example, the current upper bound for varray Grades is 7, but you can increase its upper bound to maximum of 10. A varray can contain a varying number of elements, from zero (when empty) to the

maximum specified in its type definition.

#### Element type

Any PL/SQL datatype except `BINARY_INTEGER`, `BOOLEAN`, `LONG`, `LONG RAW`, `NATURAL`, `NATURALN`, `PLS_INTEGER`, `POSITIVE`, `POSITIVEN`, `REF CURSOR`, `SIGNTYPE`, or `STRING`. Also, with varrays, `element_type` cannot be `BLOB`, `CLOB`, or an object type with `BLOB` or `CLOB` attributes.

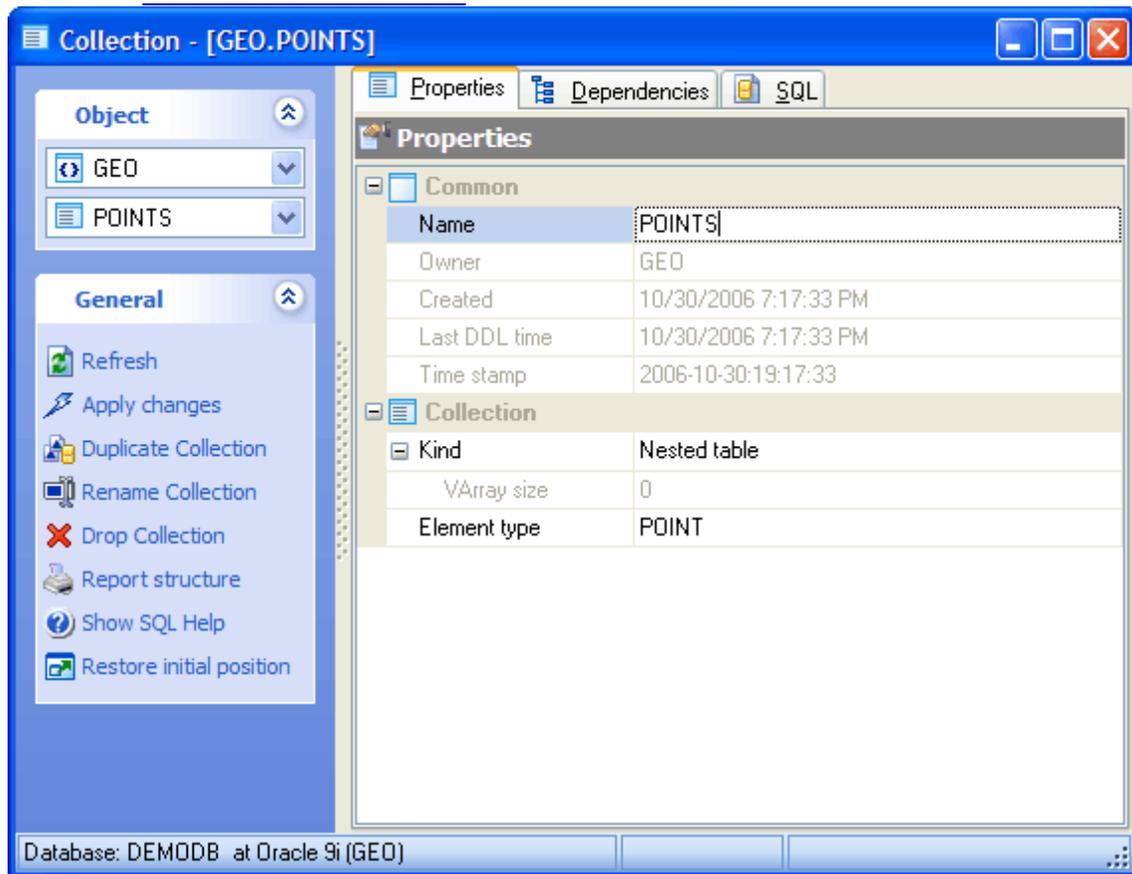
### 5.17.2 Collection Editor

Collection Editor can be opened automatically after the collection is created and is available on editing (see [Editing Collections](#)<sup>[193]</sup> for details).

The basic principles of Object Editors in Oracle Maestro are explained in a [separate topic](#)<sup>[39]</sup>. Below you will find a description of editor tabs that are unique for the current object.

- [Editing collection properties](#)<sup>[196]</sup>

**See also:** [Create Collection Wizard](#)<sup>[194]</sup>



#### 5.17.2.1 Editing collection properties

Collection Editor provides you with an ability to edit collection properties. The **Properties** tab allows you to change the collection name, the collection type.

##### Name

Use the field to rename the collection.

**Owner**

The field displays the owner of the collection.

**Created**

The field displays the date the object was created.

**Last DDL time**

Use the field to find the date when the last data definition language (DDL) operation was performed on the current object. The Last DDL time can help you to find if any changes to the object definitions have been made on or after a specific time.

**Kind (Varying array, Nested table)**

**Nested table** hold an arbitrary number of elements. They use sequential numbers as subscripts. You can define equivalent SQL types, allowing nested tables to be stored in database tables and manipulated through SQL.

**Varying array** (short for variable-size arrays) hold a fixed number of elements (although you can change the number of elements at runtime). They use sequential numbers as subscripts. You can define equivalent SQL types, allowing varrays to be stored in database tables. They can be stored and retrieved through SQL, but with less flexibility than nested tables.

**VArray Size**

A varray has a maximum size, which you specify in its type definition. Its index has a fixed lower bound of 1 and an extensible upper bound. For example, the current upper bound for varray Grades is 7, but you can increase its upper bound to maximum of 10. A varray can contain a varying number of elements, from zero (when empty) to the maximum specified in its type definition.

**Element type**

Any PL/SQL datatype except `BINARY_INTEGER`, `BOOLEAN`, `LONG`, `LONG RAW`, `NATURAL`, `NATURALN`, `PLS_INTEGER`, `POSITIVE`, `POSITIVEN`, `REF CURSOR`, `SIGNTYPE`, or `STRING`. Also, with varrays, `element_type` cannot be `BLOB`, `CLOB`, or an object type with `BLOB` or `CLOB` attributes.

## 5.18 Java sources

### ■ How can I create a new java source?

New java sources are created within [Create Java Source Wizard](#)<sup>[199]</sup>. In order to run the wizard you should either

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

To create a new java source with the same properties as one of the existing java source has:

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

### ■ How can I edit an existing java source?

Java sources can be edited within [Java Source Editor](#).<sup>[201]</sup> In order to run the editor you should either

- select the [java source](#) for editing in the explorer tree (type the first letters of the java source name for quick search);
  - select the [Edit Java Source...](#) item from the popup menu
- or
- open the schema in [Schema Editor](#)<sup>[61]</sup> and the [Java Source](#) tab there;
  - select the java source to edit;
  - press the **Enter** key or select the [Edit Java Source](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#)).

### ■ How can I drop a java source?

To drop a java source:

- select the java source to drop in the explorer tree;
- select the [Drop Java Source](#) item from the popup menu

or

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

and confirm dropping in the dialog window.

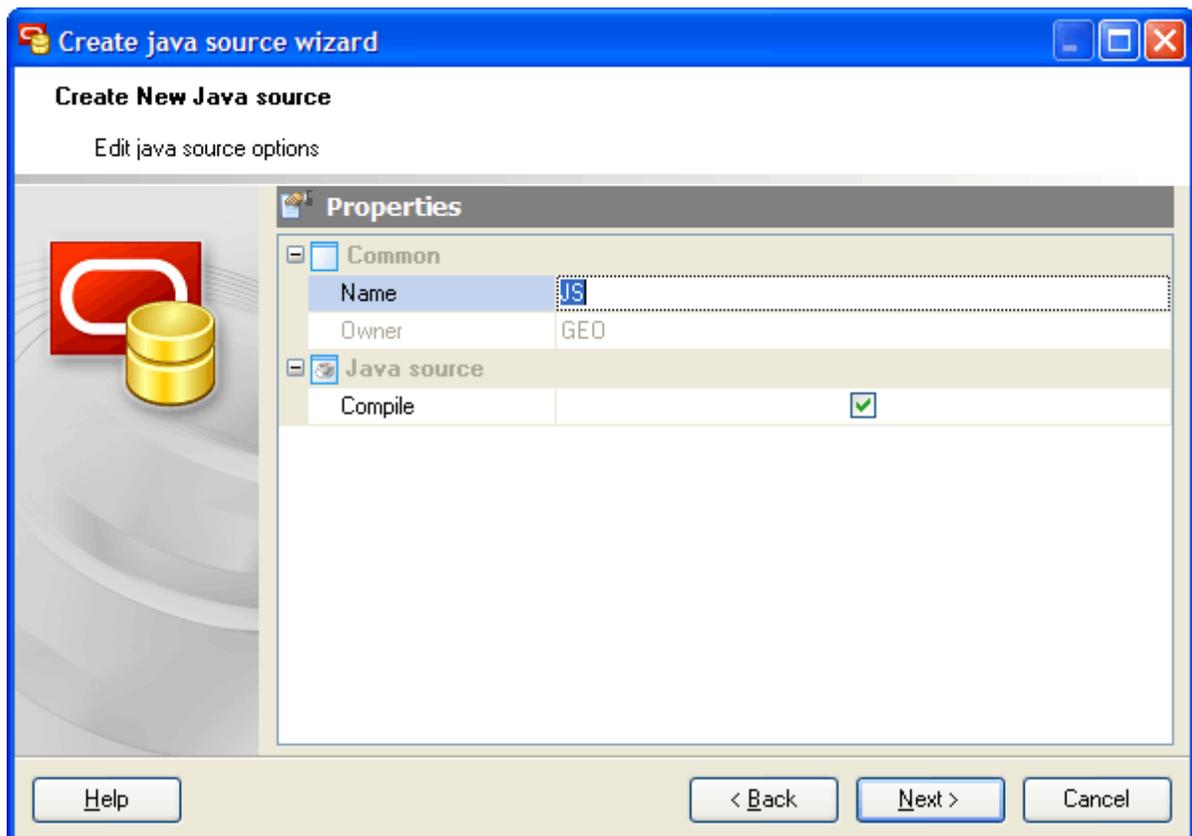
### 5.18.1 Create Java Source Wizard

[Create Java Source Wizard](#) guides you through the process of creating a new java source. See [How To Create Java Source](#)<sup>[198]</sup> for instructions on running this wizard.

The basic principles of Create Object Wizards in Oracle Maestro are explained in a [separate topic](#)<sup>[37]</sup>. Below you will find a description of wizard steps that are unique for the current object.

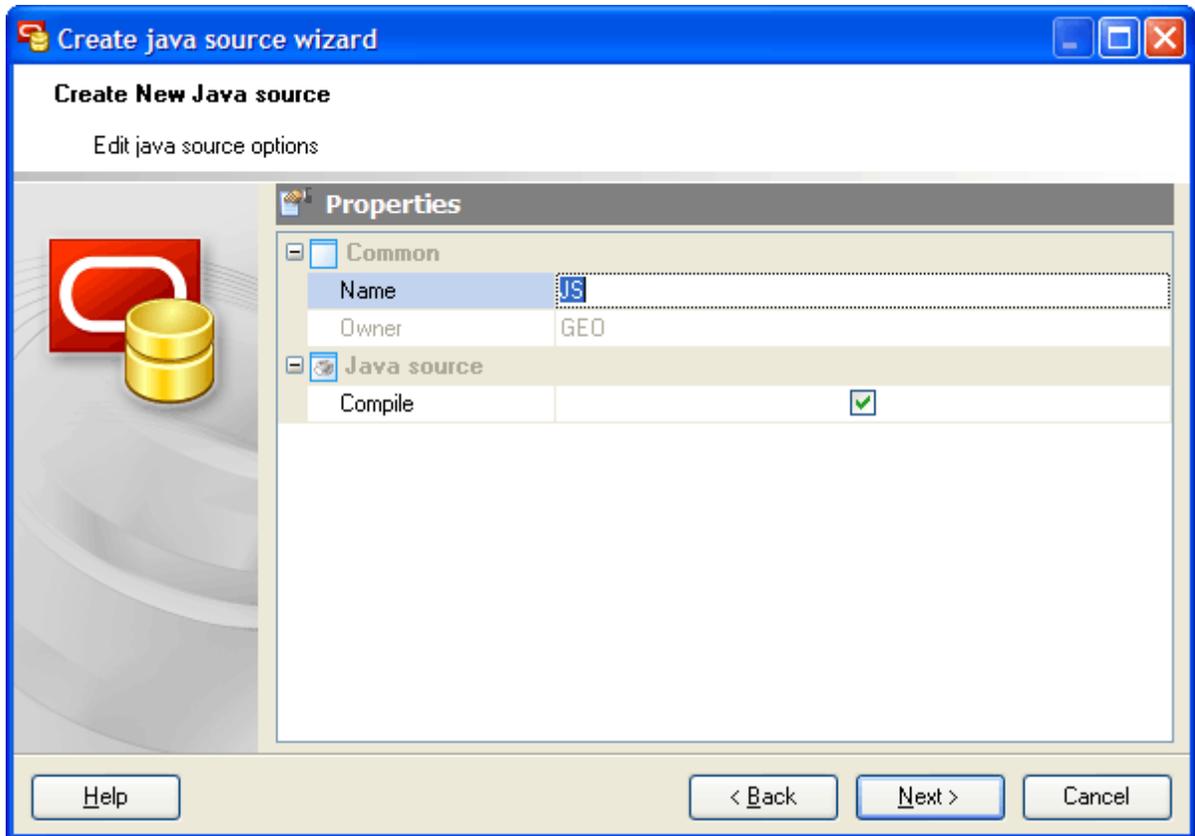
- [Specifying java source properties](#)<sup>[200]</sup>
- [Specifying source definition](#)<sup>[138]</sup>

**See also:** [Java Source Editor](#)<sup>[201]</sup>



### 5.18.1.1 Specifying java source properties

The wizard step was supplied to define common java source properties. The detailed description of the properties you can find below.



#### Name

The field represents the new java source name as it was set on the previous wizard step.

#### Owner

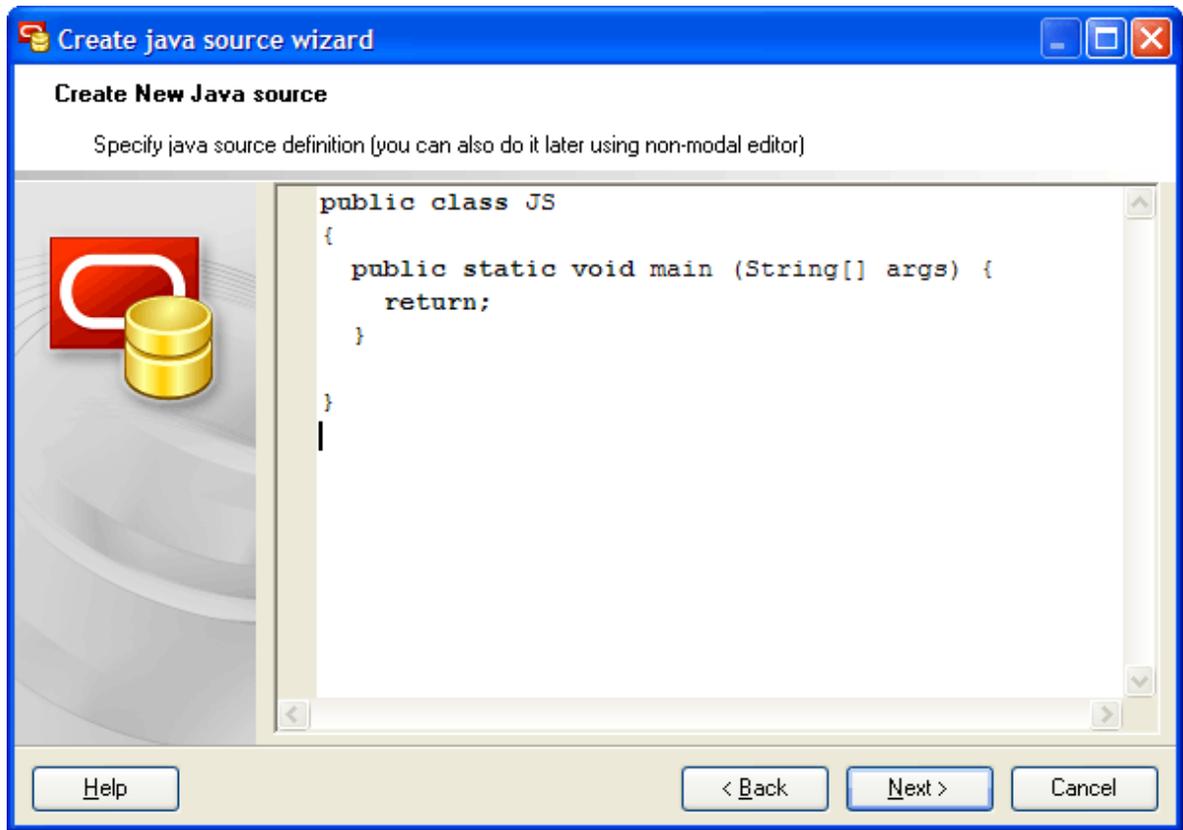
The field displays the owner of the java source.

#### Is compile

The option specifies that Oracle Database should attempt to resolve the creating Java source succeeds. The source compilation will occur.

### 5.18.1.2 Specifying java source definition

Here you can specify the java source [definition](#). You can also do it later using non-modal editor.



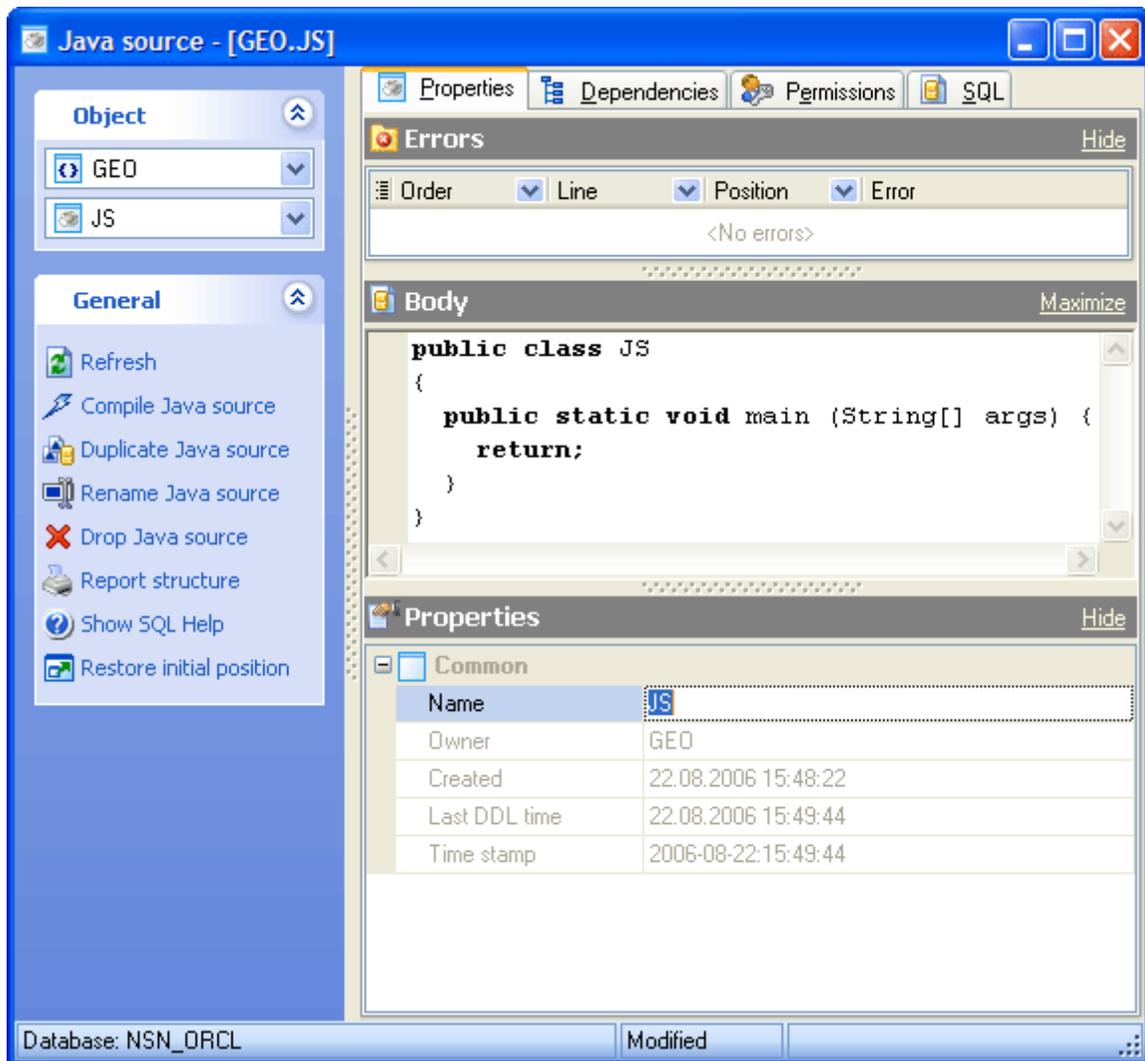
### 5.18.2 Java Source Editor

Java Source Editor can be opened automatically after the java source is created and is available on editing (see [Editing Java Sources](#)<sup>[198]</sup> for details).

The basic principles of Object Editors in Oracle Maestro are explained in a [separate topic](#)<sup>[39]</sup>. Below you will find a description of editor tabs that are unique for the current object.

- [Editing java source properties](#)<sup>[202]</sup>

**See also:** [Create Java Source Wizard](#)<sup>[199]</sup>



### 5.18.2.1 Editing java source properties

Java Source Editor provides you with an ability to edit java source properties.

The **Errors** tab displays all the necessary information about object errors. If an error have occurred during the object compilation it appears in the list with some additional properties: **Order** (one after another), **Line** and **Position** (object definition location the error was found out), **Error** (corresponding PL/SQL exception).

#### Body

Use the area to edit java source definition.

#### Name

Here you can view and change the java source 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 Oracle server.

#### Owner

There is the owner for the trigger. <%OWNER%

**Is compile**

The option specifies that Oracle Database should attempt to resolve the creating Java source succeeds. The source compilation will occur.

**Created**

The field displays the date the object was created.

**Last DDL time**

Use the field to find the date when the last data definition language (DDL) operation was performed on the current object. The Last DDL time can help you to find if any changes to the object definitions have been made on or after a specific time.

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.

## 5.19 Jobs

Using [job](#) objects you can automate administrative tasks and run them on a recurring basis.

### ■ How can I create a new job?

New jobs are created within [Create Job Wizard](#)<sup>[206]</sup>. In order to run the wizard you should either

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

or

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

or

- open the schema in [Schema Editor](#) and the [Jobs](#) tab there;
- press the **Insert** key or select the [Create New Job](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#)).

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

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

### ■ How can I edit an existing job?

Existing jobs are edited within [Job Editor](#)<sup>[207]</sup>. In order to run the editor you should either

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

or

- open the schema in [Schema Editor](#)<sup>[61]</sup> and the [Job](#) tab there;
- select the job to edit;
- press the **Enter** key or select the [Edit Job](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#)).

### ■ How can I drop a job?

To drop a job:

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

or

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

and confirm dropping in the dialog window.

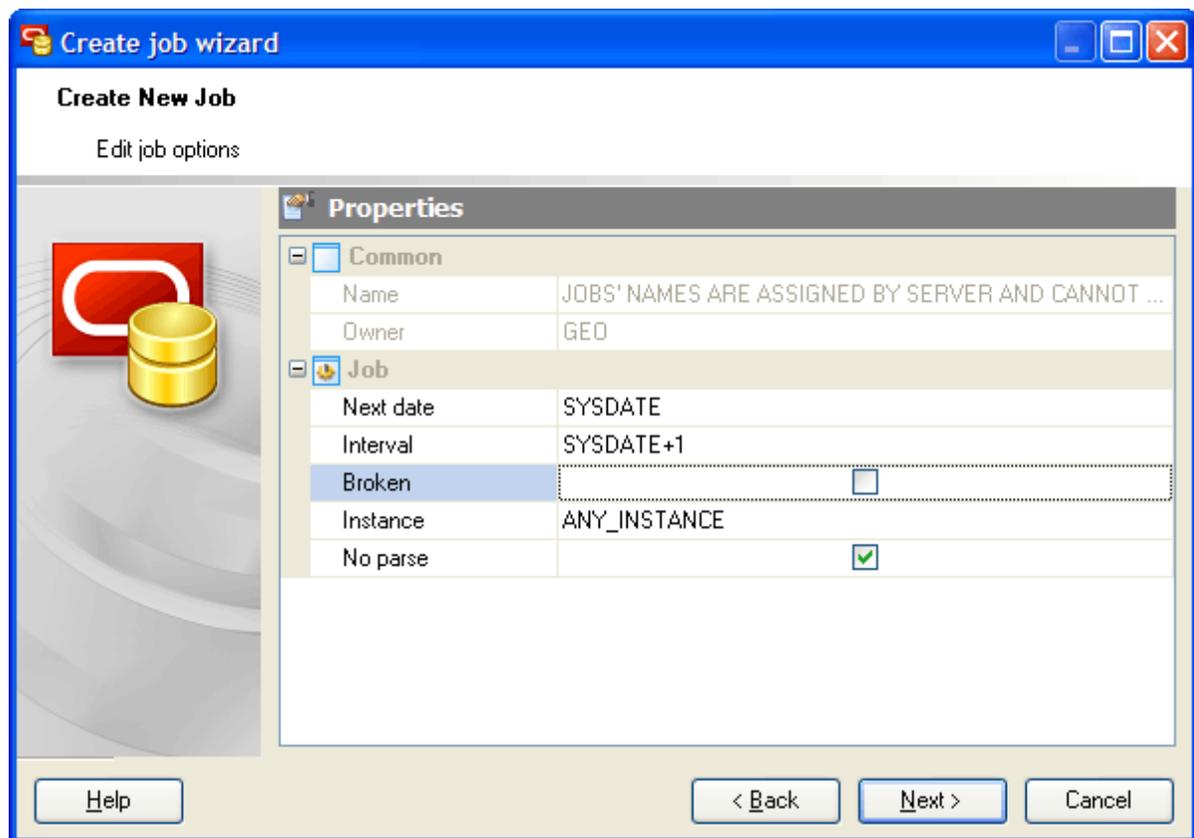
## 5.19.1 Create Job Wizard

[Create Job Wizard](#) guides you through the process of creating a new job. See [How To Create Job](#) <sup>[204]</sup> for instructions on running this wizard.

The basic principles of Create Object Wizards in Oracle Maestro are explained in a [separate topic](#) <sup>[37]</sup>. Below you will find a description of wizard steps that are unique for the current object.

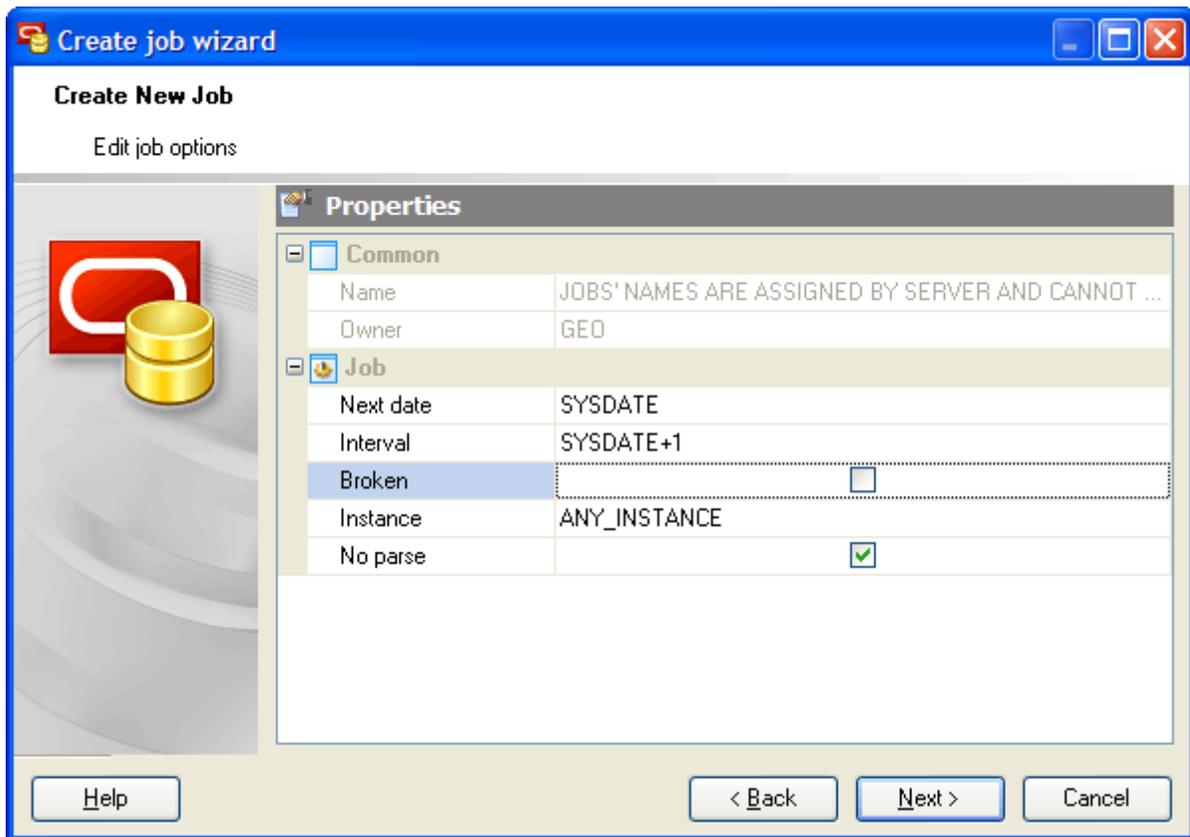
- [Specifying job properties](#) <sup>[205]</sup>
- [Specifying job definition](#) <sup>[206]</sup>

**See also:** [Job Editor](#) <sup>[207]</sup>



### 5.19.1.1 Specifying job properties

The wizard step was supplied to define common job properties. The detailed description of the properties you can find below.



#### Name

The field isn't available to editing because job names are assigned by server and cannot be change.

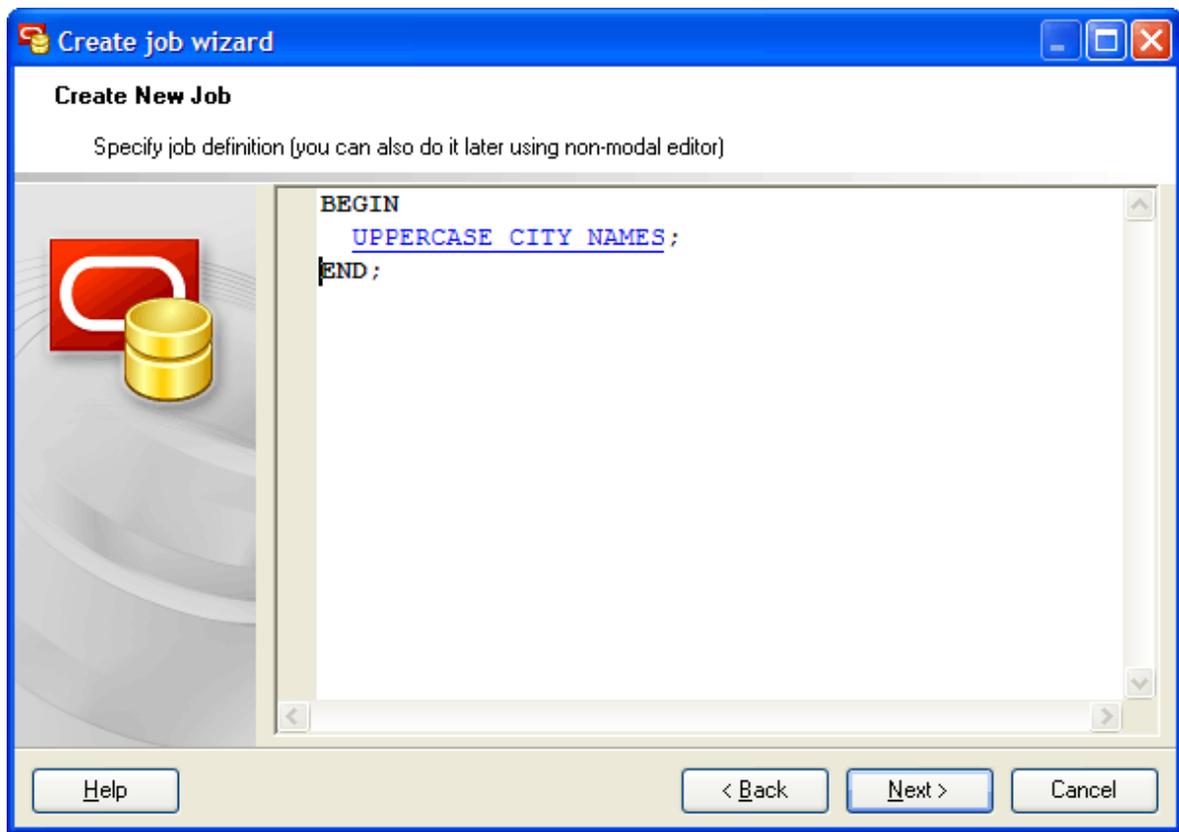
#### Owner

The field displays the owner of the job.

Use the following options to specify the job schedule: [Next day](#), [Interval](#), [Is broken](#), [Instance](#), [Is no parse](#).

#### 5.19.1.2 Specifying job definition

Here you can specify the job [definition](#). You can also do it later using non-modal editor.



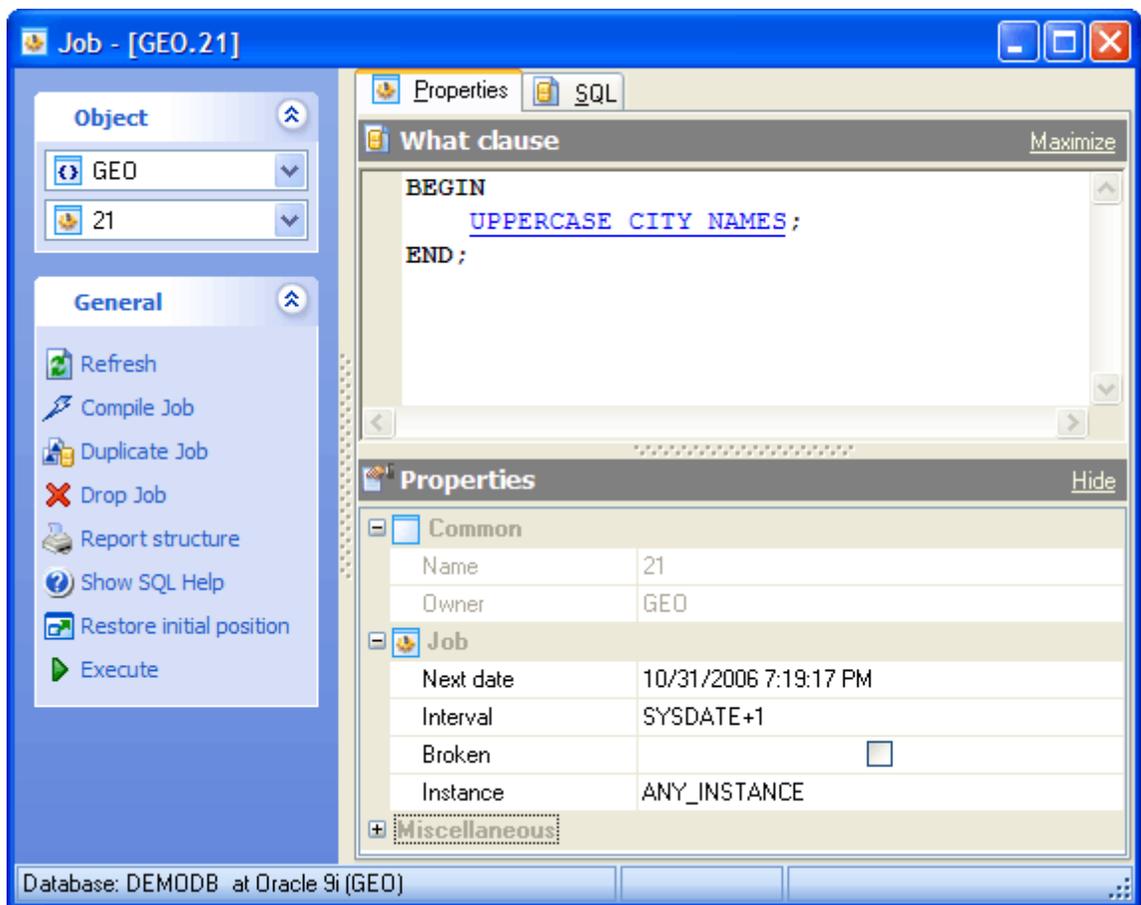
### 5.19.2 Job Editor

Job Editor can be opened automatically after the job is created and is available on editing (see [Editing Jobs](#)<sup>[204]</sup> for details).

The basic principles of Object Editors in Oracle Maestro are explained in a [separate topic](#)<sup>[39]</sup>. Below you will find a description of editor tabs that are unique for the current object.

- [Editing job properties](#)<sup>[208]</sup>

**See also:** [Create Job Wizard](#)<sup>[205]</sup>



### 5.19.2.1 Editing job properties

Job Editor provides you with an ability to edit job properties.

#### What clause

Use the area to change job definition.

#### Owner

There is the owner for the job. <%OWNER%

#### Created

The field displays the date the object was created.

#### Last DDL time

Use the field to find the date when the last data definition language (DDL) operation was performed on the current object. The Last DDL time can help you to find if any changes to the object definitions have been made on or after a specific time.

Use the following options to specify the job schedule: [Next day](#), [Interval](#), [Is broken](#), [Instance](#).

You can also find some statistic information for the job in [Miscellaneous](#) list.

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.

## 5.20 Queues

### ■ How can I create a new queue?

New queues are created within [Create Queue Wizard](#)<sup>[21]</sup>. In order to run the wizard you should either

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

or

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

or

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

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

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

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

Queues can be edited within [Queue Editor](#). In order to run the editor you should either

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

or

- open the schema in [Schema Editor](#)<sup>[61]</sup> and the [Queue](#) tab there;
- select the queue to edit;
- press the **Enter** key or select the [Edit Queue](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#)).

### ■ How can I drop an queue?

To drop a queue:

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

or

- open the schema in [Schema Editor](#) and the [Queue](#) tab there;

- select the queue to drop;
- press the **Delete** key or select the [Drop Queue](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#))

and confirm dropping in the dialog window.

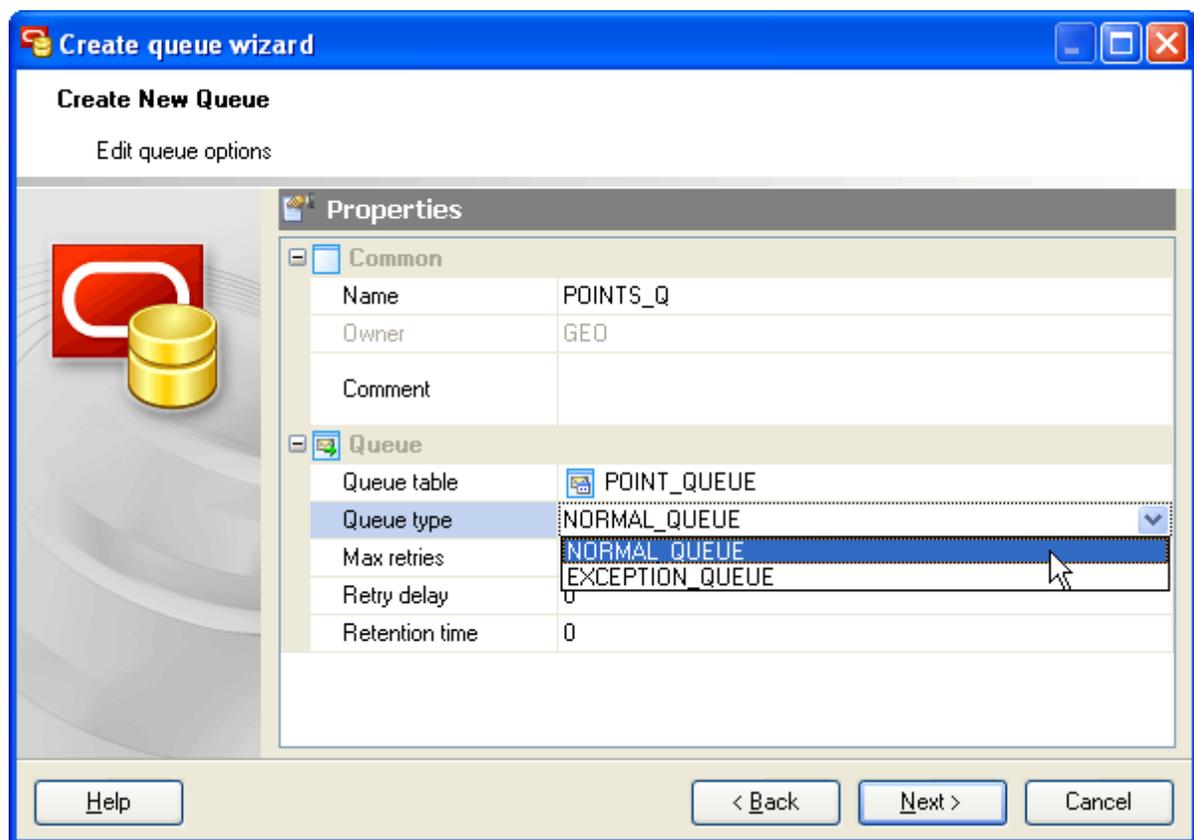
## 5.20.1 Create Queue Wizard

Create Queue Wizard guides you through the process of creating a new database queue. See [How To Create Queue](#)<sup>[210]</sup> for instructions on running this wizard.

The basic principles of Create Object Wizards in Oracle Maestro are explained in a [separate topic](#)<sup>[37]</sup>. Below you will find a description of wizard steps that are unique for the current object.

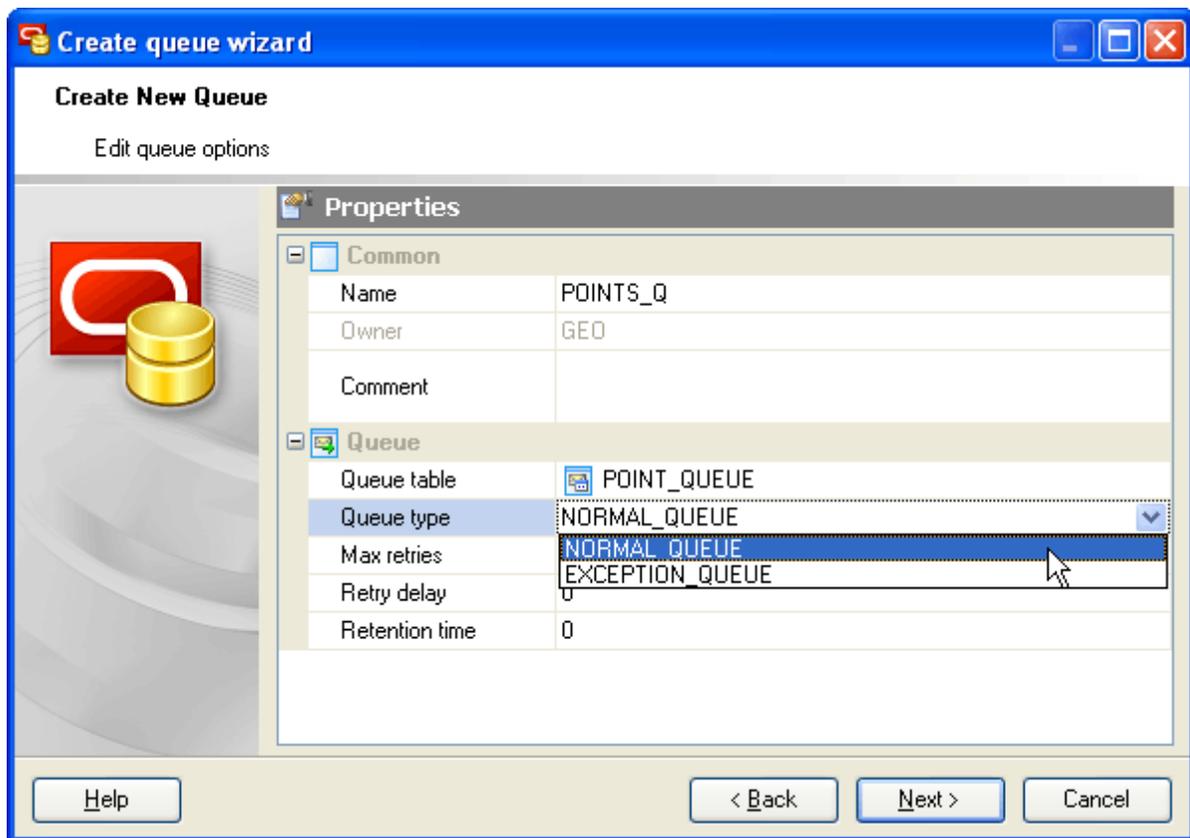
- [Specifying queue properties](#)<sup>[211]</sup>

**See also:** [Queue Editor](#)<sup>[212]</sup>



### 5.20.1.1 Specifying queue properties

The wizard step was supplied to define common queue properties. The detailed description of the properties you can find below.



#### Name

The field represents the new queue name as it was set on the previous wizard step.

#### Owner

The field displays the owner of the queue.

#### Comment

Use the field to describe the queue or leave it blank.

#### Queue table

Set the queue table for the queue.

#### Queue type (Normal\_queue, Exception\_queue)

Specify whether the queue contains exceptions or not.

## 5.20.2 Queue Editor

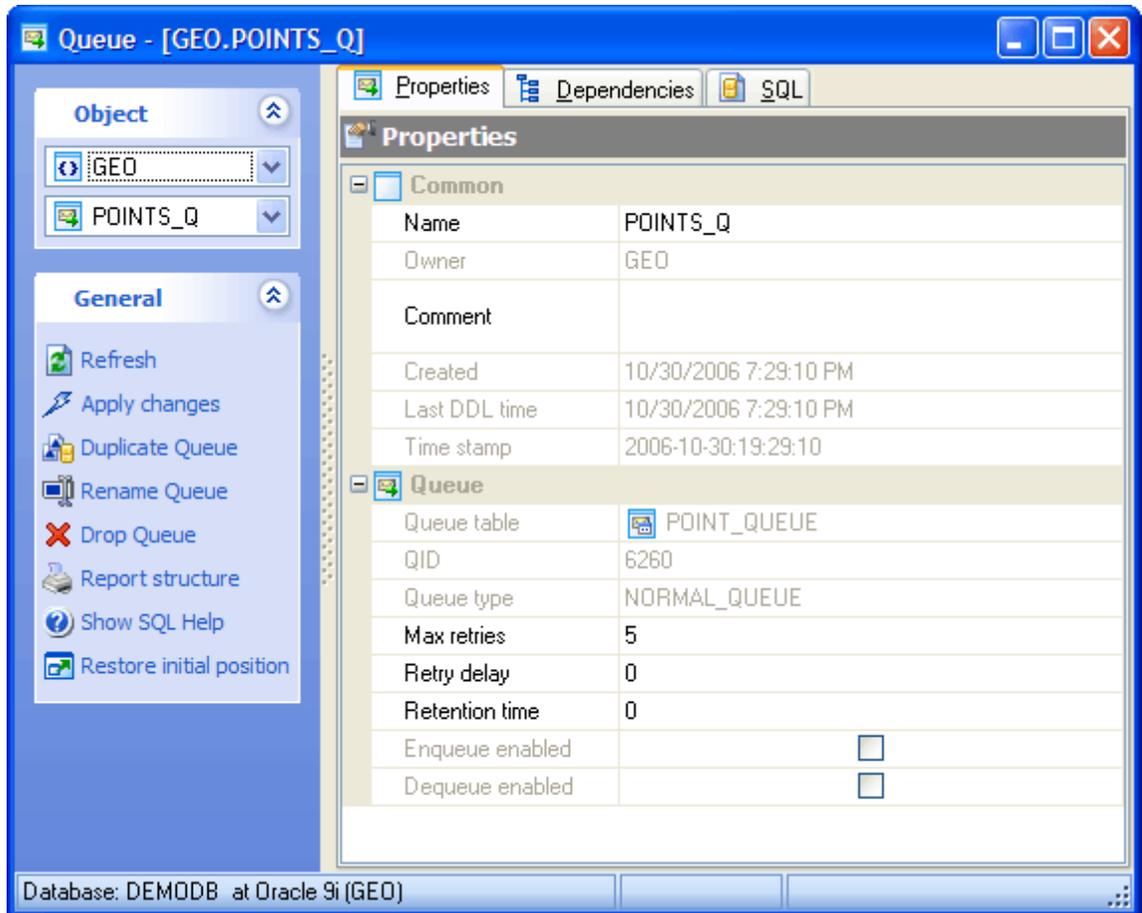
[Queue Editor](#) can be opened automatically after the queue is created and is available on editing (see [Editing Queues](#)<sup>[210]</sup> for details).

The basic principles of Object Editors in Oracle Maestro are explained in a [separate topic](#)<sup>[30]</sup>. Below you will find a description of editor tabs that are unique for the current

object.

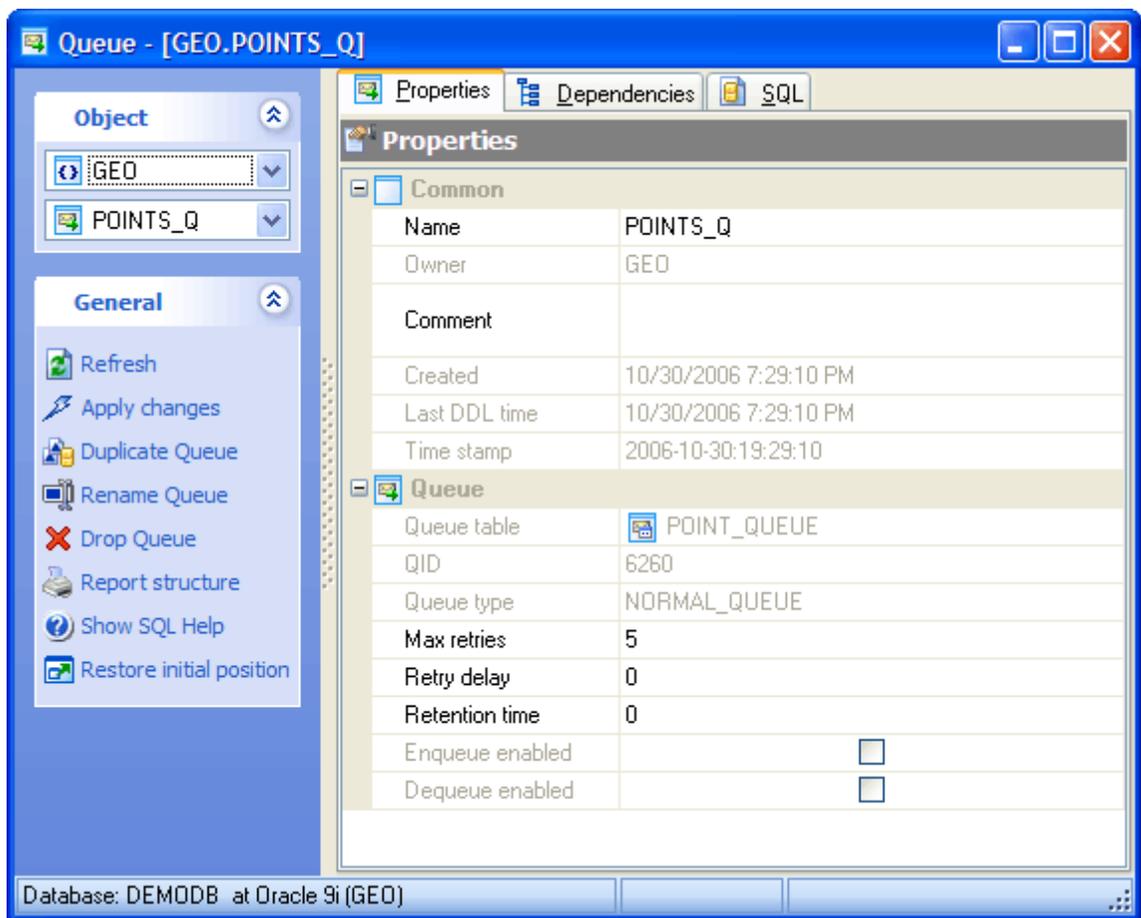
- [Editing queue properties](#)<sup>[213]</sup>

**See also:** [Create Queue Wizard](#)<sup>[214]</sup>



### 5.20.2.1 Editing queue properties

Queue Editor provides you with an ability to view queue properties.



#### Name

Here you can view and change the queue name.

#### Owner

There is the owner for the queue. <%OWNER%

#### Comment

Use the field to describe the queue or leave it blank.

#### Queue table

Use the field to find the queue table of the queue.

#### Created

The field displays the date the object was created.

#### Last DDL time

Use the field to find the date when the last data definition language (DDL) operation was performed on the current object. The Last DDL time can help you to find if any changes to the object definitions have been made on or after a specific time.

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.

## 5.21 Queue tables

### ■ How can I create a new queue table?

New queue tables are created within [Create Queue Table Wizard](#).<sup>[216]</sup> In order to run the wizard you should either

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

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

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

### ■ How can I edit an existing queue table?

Queue tables can be edited within [Queue Table Editor](#).<sup>[218]</sup> In order to run the editor you should either

- select the [queue table](#) for editing in the explorer tree (type the first letters of the queue table name for quick search);
  - select the [Edit Queue Table...](#) item from the popup menu
- or
- open the schema in [Schema Editor](#)<sup>[61]</sup> and the [Queue Table](#) tab there;
  - select the queue table to edit;
  - press the **Enter** key or select the [Edit Queue Table](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#)).

### ■ How can I drop a queue table?

To drop a queue table:

- select the [queue table](#) to drop in the explorer tree;
- select the [Drop Queue Table](#) item from the popup menu

or

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

and confirm dropping in the dialog window.

### 5.21.1 Create Queue Table Wizard

[Create Queue Table Wizard](#) guides you through the process of creating a new queue table. See [How To Create Queue Table](#)<sup>[215]</sup> for instructions on running this wizard.

The basic principles of Create Object Wizards in Oracle Maestro are explained in a [separate topic](#)<sup>[37]</sup>. Below you will find a description of wizard steps that are unique for the current object.

- [Specifying queue table properties](#)<sup>[217]</sup>

**See also:** [Queue Table Editor](#)<sup>[218]</sup>

Common	
Name	POINT_QUEUE
Owner	GEO
Comment	

Queue table	
Storage options	TABLESPACE USERS INITRANS 1
Payload type	POINT
Sort order	ENQUEUE_TIME
Has multiple consumers	<input type="checkbox"/>
Message grouping	NONE
Compatible	8.0
Owner instance	1
Primary instance	0
Secondary instance	0
Secure	<input type="checkbox"/>

### 5.21.1.1 Specifying queue table properties

The wizard step was supplied to define common queue table properties. The detailed description of the properties you can find below.

Properties	
Common	
Name	POINT_QUEUE
Owner	GEO
Comment	
Queue table	
Storage options	TABLESPACE USERS INITRANS 1
Payload type	POINT
Sort order	ENQUEUE_TIME
Has multiple consumers	<input type="checkbox"/>
Message grouping	NONE
Compatible	8.0
Owner instance	1
Primary instance	0
Secondary instance	0
Secure	<input type="checkbox"/>

#### Name

The field represents the new queue table name as it was set on the previous wizard step.

#### Owner

The field displays the owner of the queue table.

#### Comment

Use the field to describe the queue table or leave it blank.

#### Storage options

Specify the [Storage option properties](#) for the queue table.

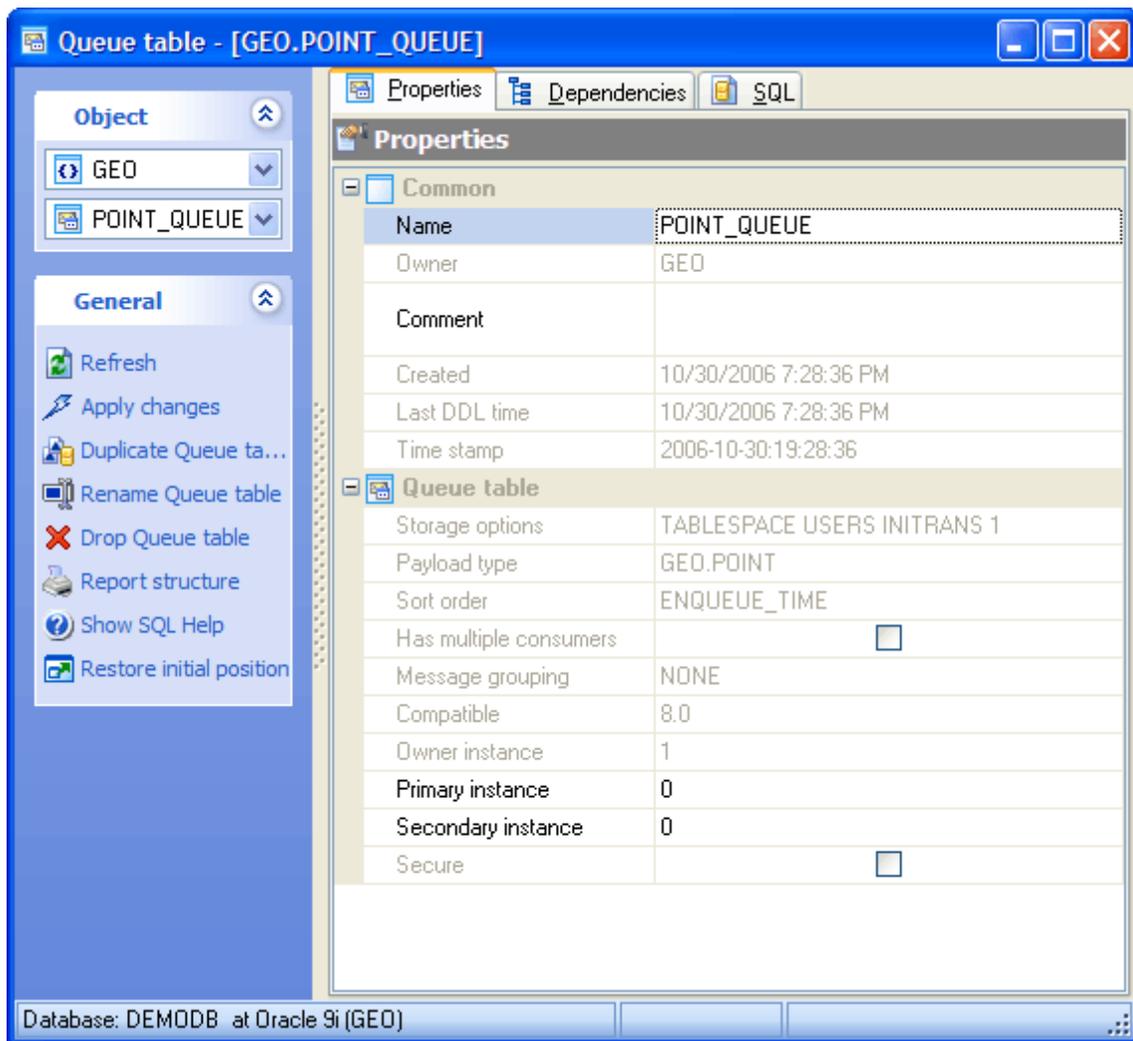
## 5.21.2 Queue Table Editor

Queue Table Editor can be opened automatically after the queue table is created and is available on editing (see [Editing Queue Tables](#)<sup>[215]</sup> for details).

The basic principles of Object Editors in Oracle Maestro are explained in a [separate topic](#)<sup>[209]</sup>. Below you will find a description of editor tabs that are unique for the current object.

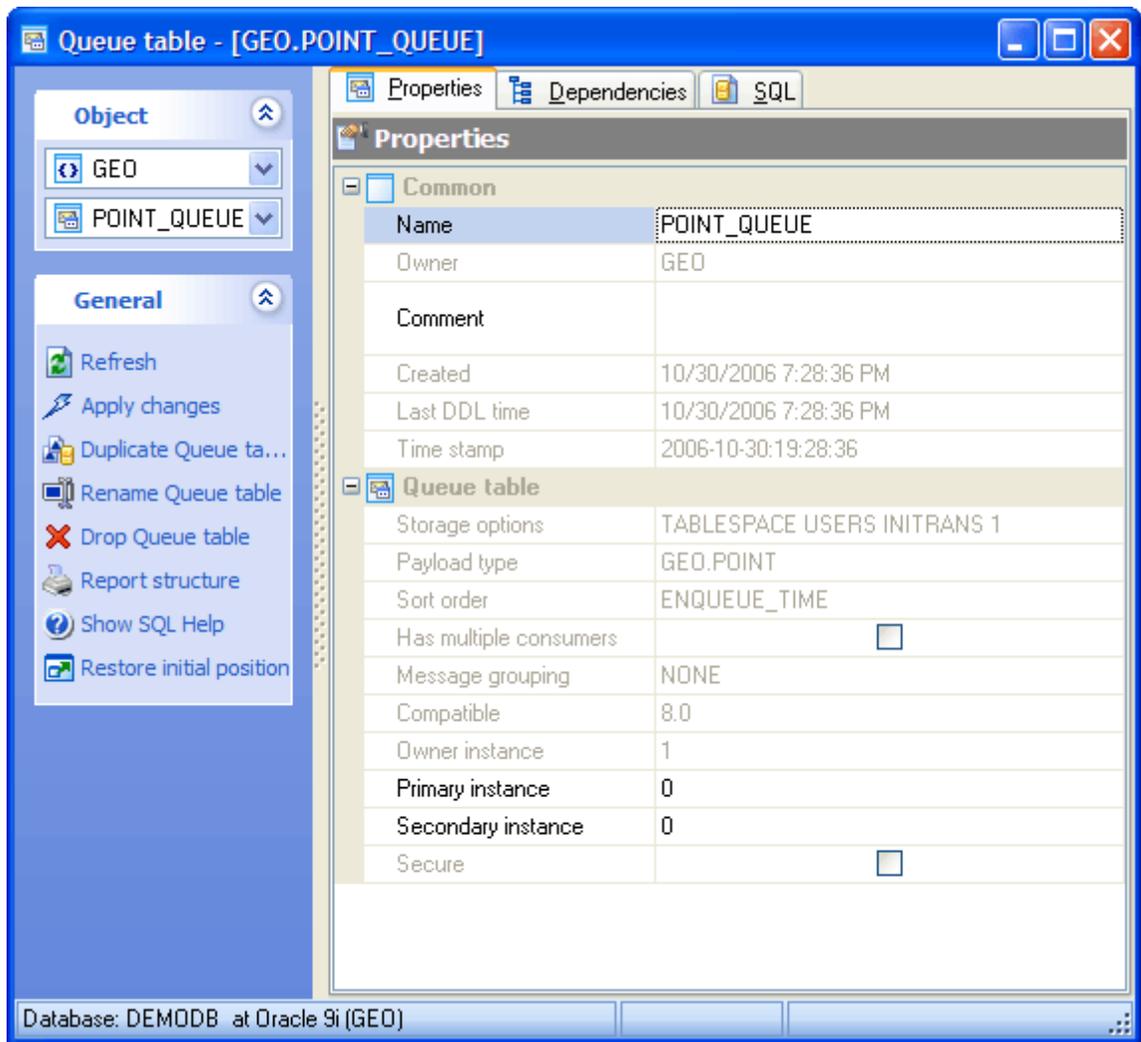
- [Editing queue table properties](#)<sup>[218]</sup>

**See also:** [Create Queue Table Wizard](#)<sup>[216]</sup>



### 5.21.2.1 Editing queue table properties

Queue Table Editor provides you with an ability to view and also edit some its properties in the handiest way.



#### Name

Here you can view and change the queue table name.

#### Owner

There is the owner for the queue table. <%OWNER%

#### Comment

Use the field to describe the queue table or leave it blank.

#### Created

The field displays the date the object was created.

#### Last DDL time

Use the field to find the date when the last data definition language (DDL) operation was performed on the current object. The Last DDL time can help you to find if any changes to the object definitions have been made on or after a specific time.

#### Storage options

The field displays the tablespace the queue table is physically located.

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.

## 5.22 Clusters

A cluster is a schema object that contains data from one or more tables, all of which have one or more columns in common. Oracle Database stores together all the rows from all the tables that share the same cluster key.

### ■ How can I create a new cluster?

New clusters are created within [Create cluster Wizard](#).<sup>[222]</sup> In order to run the wizard you should either

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

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

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

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

Clusters can be edited within [Cluster Editor](#).<sup>[225]</sup> In order to run the editor you should either

- select the **cluster** for editing in the explorer tree (type the first letters of the cluster name for quick search);
  - select the **Edit Cluster...** item from the popup menu
- or
- open the schema in [Schema Editor](#)<sup>[61]</sup> and the **Cluster** tab there;
  - select the cluster to edit;
  - press the **Enter** key or select the **Edit Cluster** item from the popup menu (alternatively, you may use the corresponding link of the **Navigation Bar**).

### ■ How can I drop a cluster?

To drop a cluster:

- select the **Cluster** to drop in the explorer tree;

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

and confirm dropping in the dialog window.

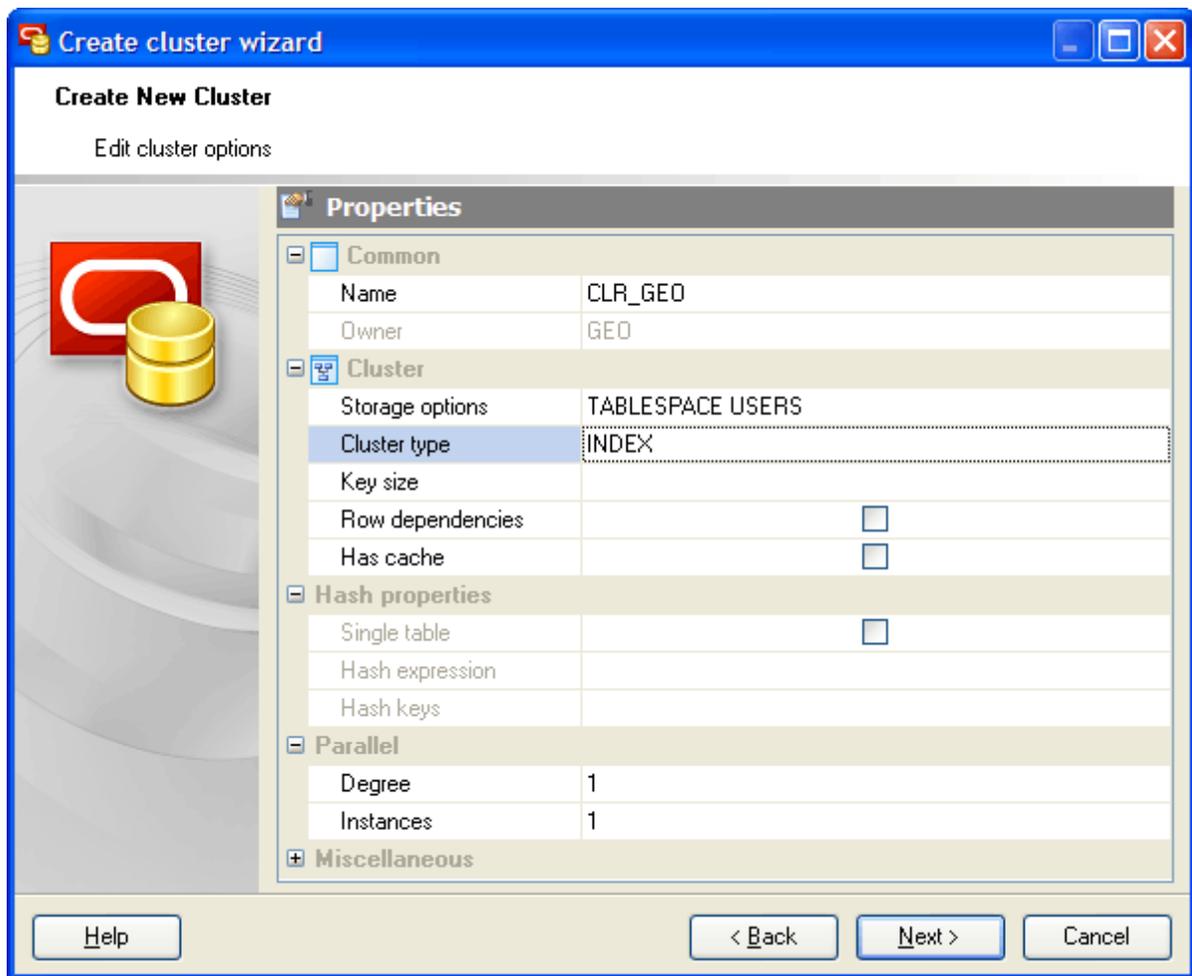
### 5.22.1 Create Cluster Wizard

[Create Cluster Wizard](#) guides you through the process of creating a new cluster. See [How To Create Cluster](#)<sup>[221]</sup> for instructions on running this wizard.

The basic principles of Create Object Wizards in Oracle Maestro are explained in a [separate topic](#)<sup>[37]</sup>. Below you will find a description of wizard steps that are unique for the current object.

- [Specifying cluster properties](#)<sup>[223]</sup>
- [Editing cluster subitems](#)<sup>[224]</sup>

**See also:** [Cluster Editor](#)<sup>[225]</sup>



### 5.22.1.1 Specifying cluster properties

The wizard step was supplied to define common cluster properties. The detailed description of the properties you can find below.

#### Name

The field represents the new cluster name as it was set on the previous wizard step.

#### Owner

The field displays the owner of the cluster.

#### Storage options

Specify the tablespace in which the cluster is to be created. For more information see [Storage option properties](#).

#### Cluster type (Default, Index, Hash)

Specify **Index** to create an indexed cluster. In an indexed cluster, Oracle Database stores together rows having the same cluster key value. Each distinct cluster key value is stored only once in each data block, regardless of the number of tables and rows in which it occurs. In a **Hash** cluster, Oracle Database stores together rows that have the same hash key value.

### Key size

Specify the number of hash values for the hash cluster. The hash value for a row is the value returned by the hash function of the cluster. Oracle Database rounds up the `keys` value to the nearest prime number to obtain the actual number of hash values. The minimum value for this parameter is 2. If you omit both the `INDEX` clause and the `keys` parameter, the database creates an indexed cluster by default.

### Row dependencies

Specify the option if you want to enable row-level dependency tracking. This setting is useful primarily to allow for parallel propagation in replication environments. It increases the size of each row by 6 bytes.

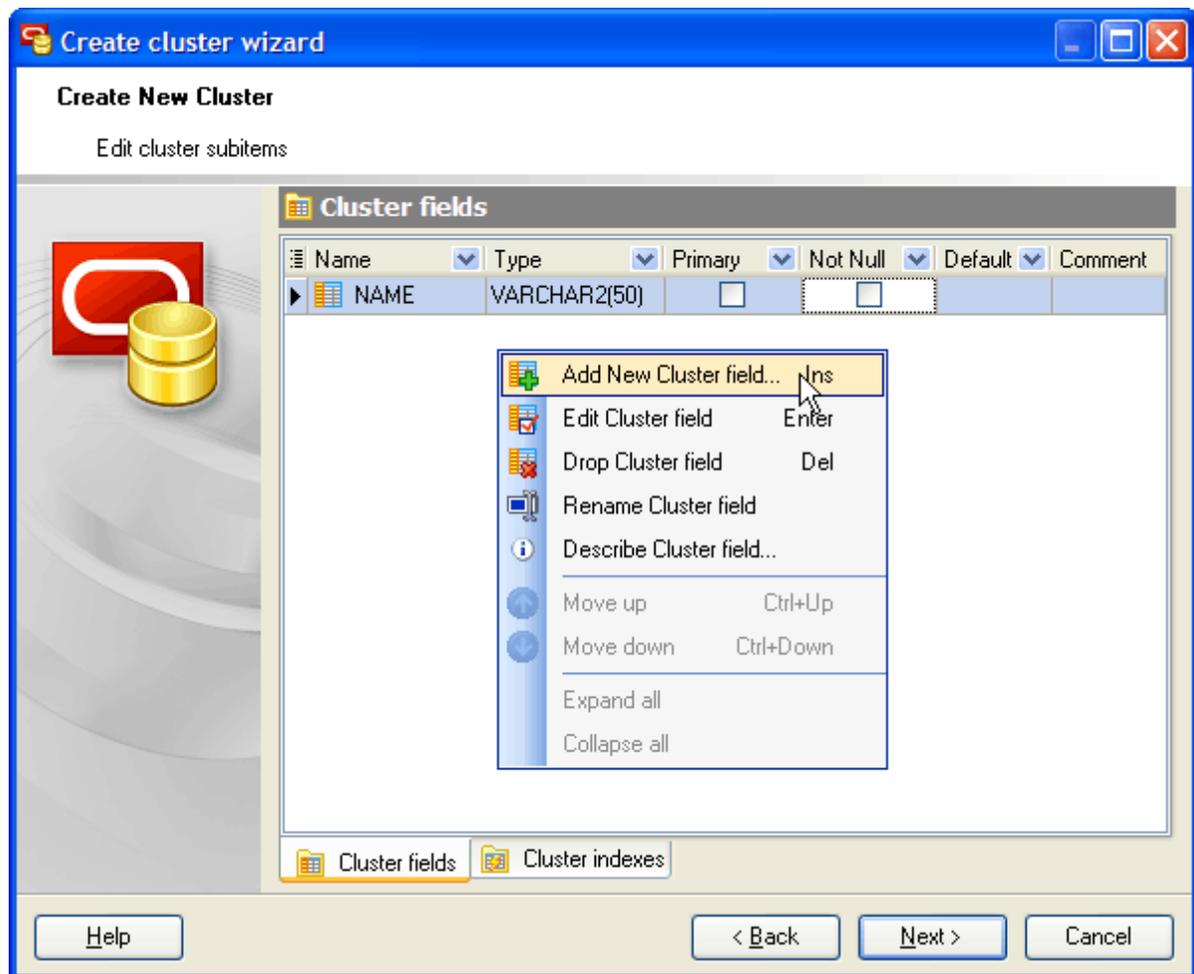
### Has cache

Check the option if you want the blocks retrieved for this cluster to be placed at the most recently used end of the least recently used (LRU) list in the buffer cache when a full table scan is performed. This clause is useful for small lookup tables.

You can also set `Parallel` properties.

#### 5.22.1.2 Adding cluster subitems

The wizard step allows you to create all the cluster subitems (cluster fields, cluster indexes) during the cluster creation. Just select the necessary tab and use popup menu to create new, edit or drop the selected cluster subitems.



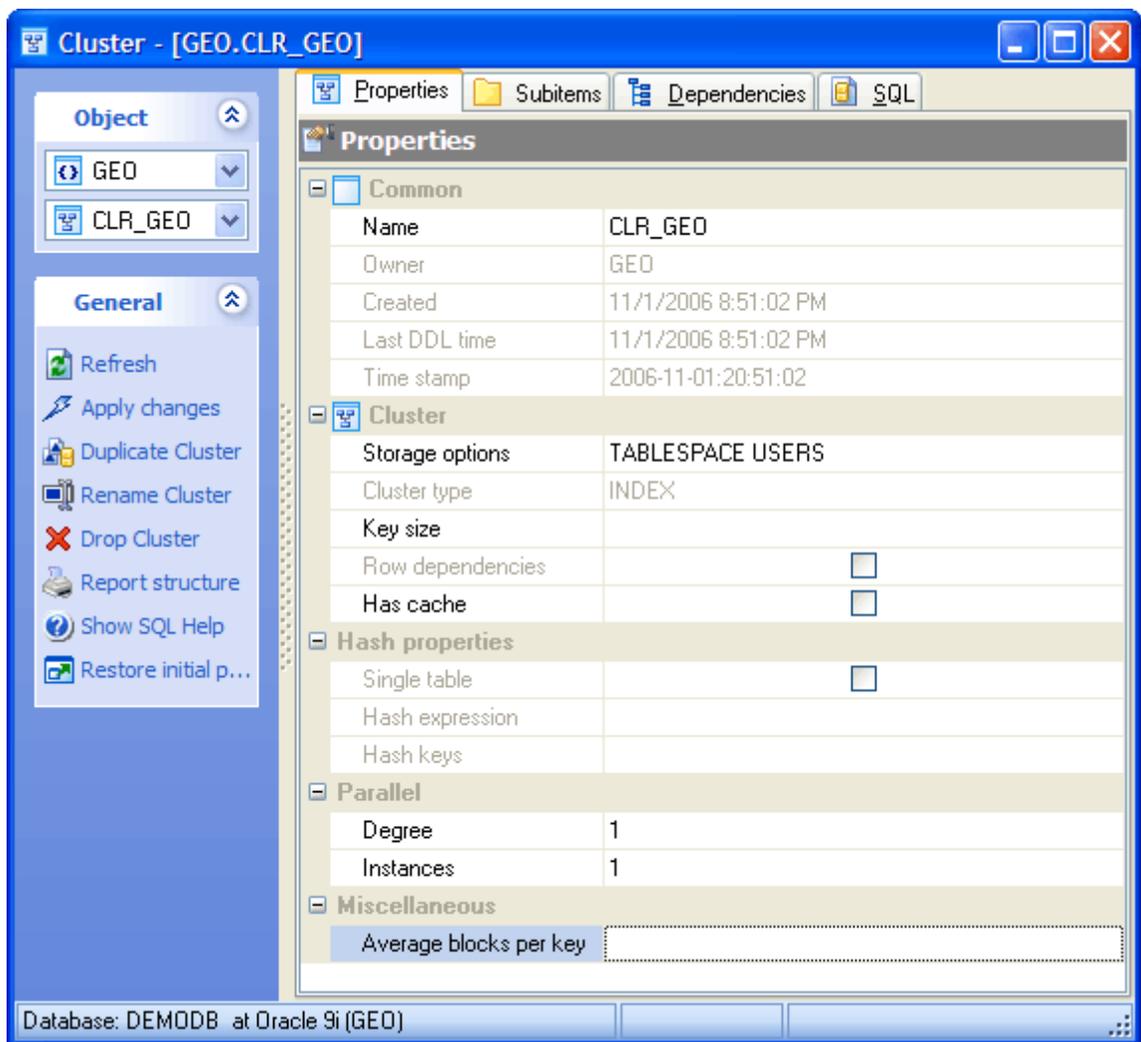
### 5.22.2 Cluster Editor

Cluster Editor can be opened automatically after the cluster is created and is available on editing (see [Editing Clusters](#)<sup>[221]</sup> for details).

The basic principles of Object Editors in Oracle Maestro are explained in a [separate topic](#)<sup>[39]</sup>. Below you will find a description of editor tabs that are unique for the current object.

- [Editing cluster properties](#)<sup>[226]</sup>

**See also:** [Create Cluster Wizard](#)<sup>[222]</sup>



### 5.22.2.1 Editing cluster properties

Cluster Editor provides you with an ability to edit cluster properties in the easiest way.

Every Subitems tab is intended for managing cluster fields, tables, indexes. Each object can be opened in its editor. Use grid's popup menu to create new, edit or drop the selected cluster 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.

#### Name

Here you can view and change the cluster name.

#### Owner

The field displays the owner of the cluster.

#### Created

The field displays the date the object was created.

#### Last DDL time

Use the field to find the date when the last data definition language (DDL) operation was performed on the current object. The Last DDL time can help you to find if any changes to the object definitions have been made on or after a specific time.

#### Storage options

Here you can find the tablespace in which the cluster is physically located.

#### Cluster type (Default, Index, Hash)

Specify [Index](#) to create an indexed cluster. In an indexed cluster, Oracle Database stores together rows having the same cluster key value. Each distinct cluster key value is stored only once in each data block, regardless of the number of tables and rows in which it occurs. In a [Hash](#) cluster, Oracle Database stores together rows that have the same hash key value.

#### Key size

Specify the number of hash values for the hash cluster. The hash value for a row is the value returned by the hash function of the cluster. Oracle Database rounds up the [keys](#) value to the nearest prime number to obtain the actual number of hash values. The minimum value for this parameter is 2. If you omit both the [INDEX](#) clause and the [keys](#) parameter, the database creates an indexed cluster by default.

#### Row dependencies

Specify the option if you want to enable row-level dependency tracking. This setting is useful primarily to allow for parallel propagation in replication environments. It increases the size of each row by 6 bytes.

#### Has cache

Check the option if you want the blocks retrieved for this cluster to be placed at the most recently used end of the least recently used (LRU) list in the buffer cache when a full table scan is performed. This clause is useful for small lookup tables.

You can also set [Parallel](#) properties and view some statistic options in [Miscellaneous](#) tab. 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.

## 5.23 Libraries

Library is a schema object associated with an operating-system shared library.

### ■ How can I create a new library?

New libraries are created within [Create Library Wizard](#).<sup>[229]</sup> In order to run the wizard you should either

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

or

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

or

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

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

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

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

Libraries can be edited within [Library Editor](#).<sup>[230]</sup> In order to run the editor you should either

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

or

- open the schema in [Schema Editor](#)<sup>[61]</sup> and the [Library](#) tab there;
- select the library to edit;
- press the **Enter** key or select the [Edit Library](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#)).

### ■ How can I drop a library?

To drop a library:

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

or

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

and confirm dropping in the dialog window.

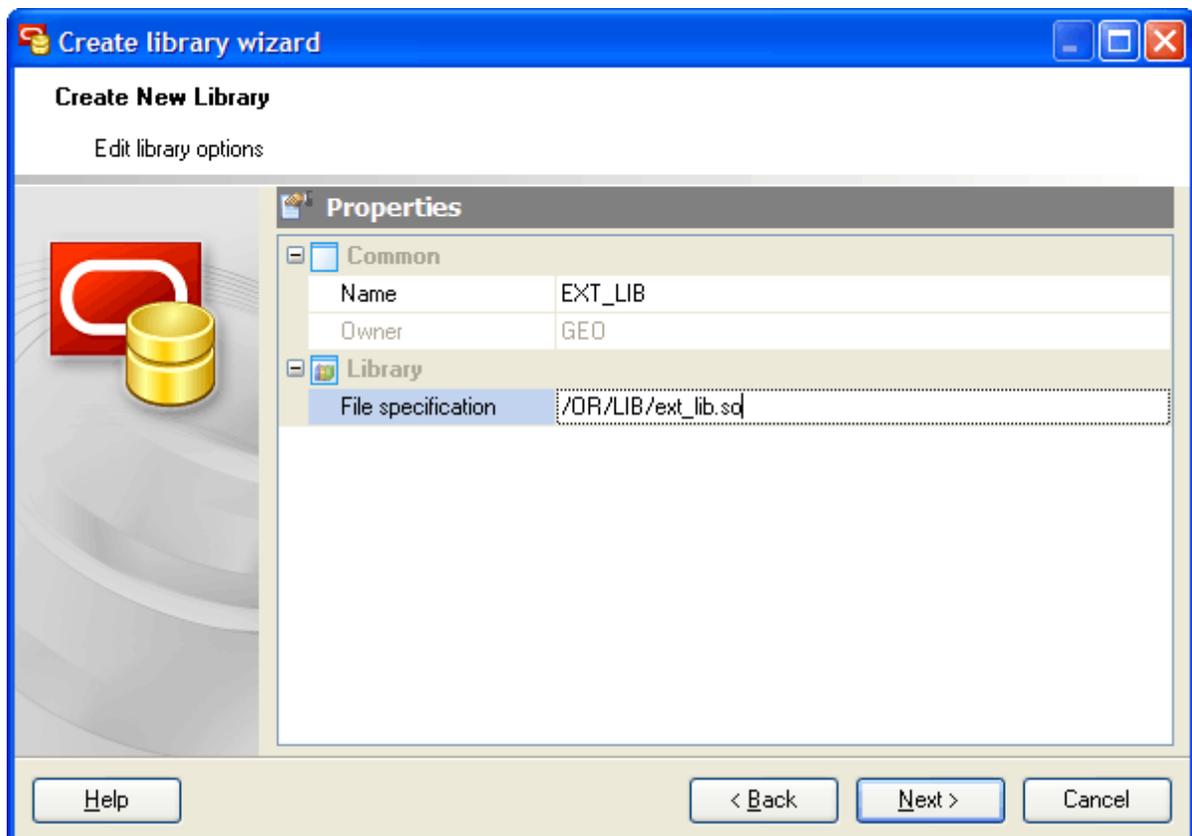
### 5.23.1 Create Library Wizard

[Create Library Wizard](#) guides you through the process of creating a new library. See [How To Create Library](#)<sup>[228]</sup> for instructions on running this wizard.

The basic principles of Create Object Wizards in Oracle Maestro are explained in a [separate topic](#)<sup>[37]</sup>. Below you will find a description of wizard steps that are unique for the current object.

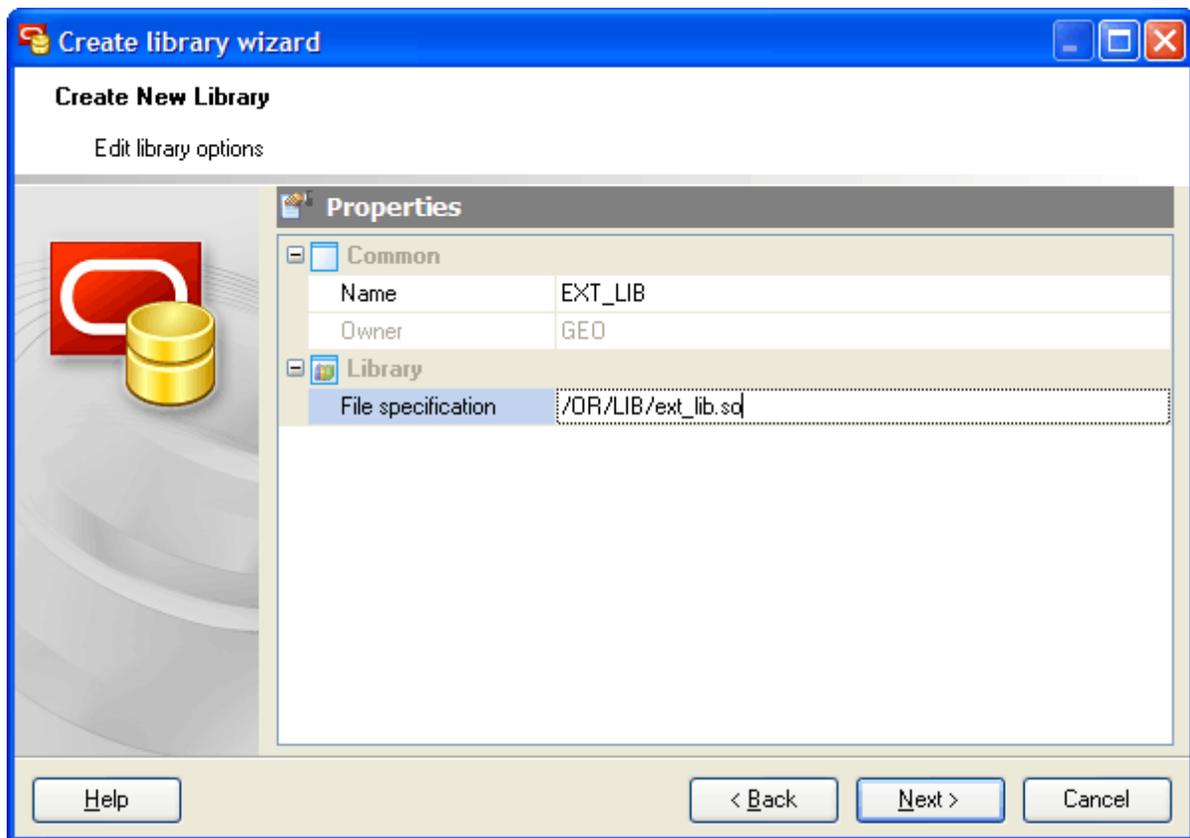
- [Specifying library properties](#)<sup>[229]</sup>

**See also:** [Library Editor](#)<sup>[230]</sup>



#### 5.23.1.1 Specifying library properties

The wizard step was supplied to define common library properties. The detailed description of the properties you can find below.



#### Name

The field represents the new library name as it was set on the previous wizard step.

#### Owner

The field displays the owner of the library.

#### File specification

Specify a string literal, enclosed in single quotation marks. This string should be the path or filename your operating system recognizes as naming the shared library.

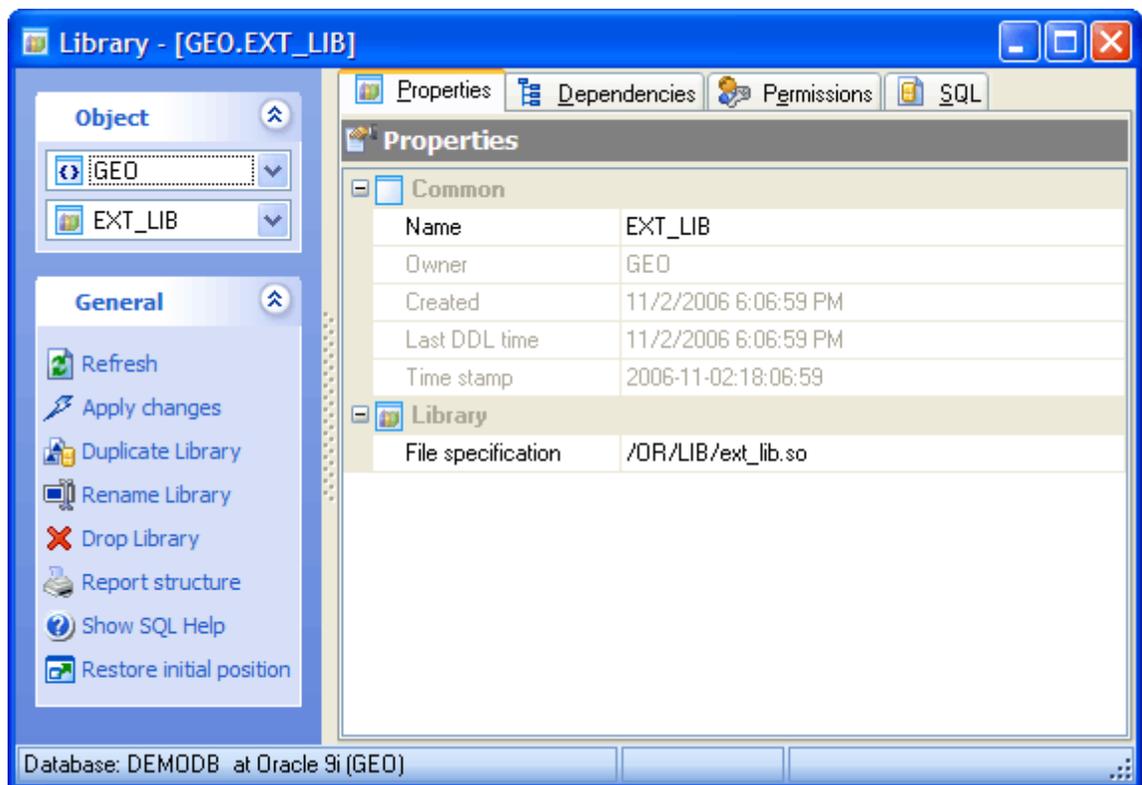
### 5.23.2 Library Editor

[Library Editor](#) can be opened automatically after the library is created and is available on editing (see [Editing Libraries](#)<sup>[228]</sup> for details).

The basic principles of Object Editors in Oracle Maestro are explained in a [separate topic](#)<sup>[229]</sup>. Below you will find a description of editor tabs that are unique for the current object.

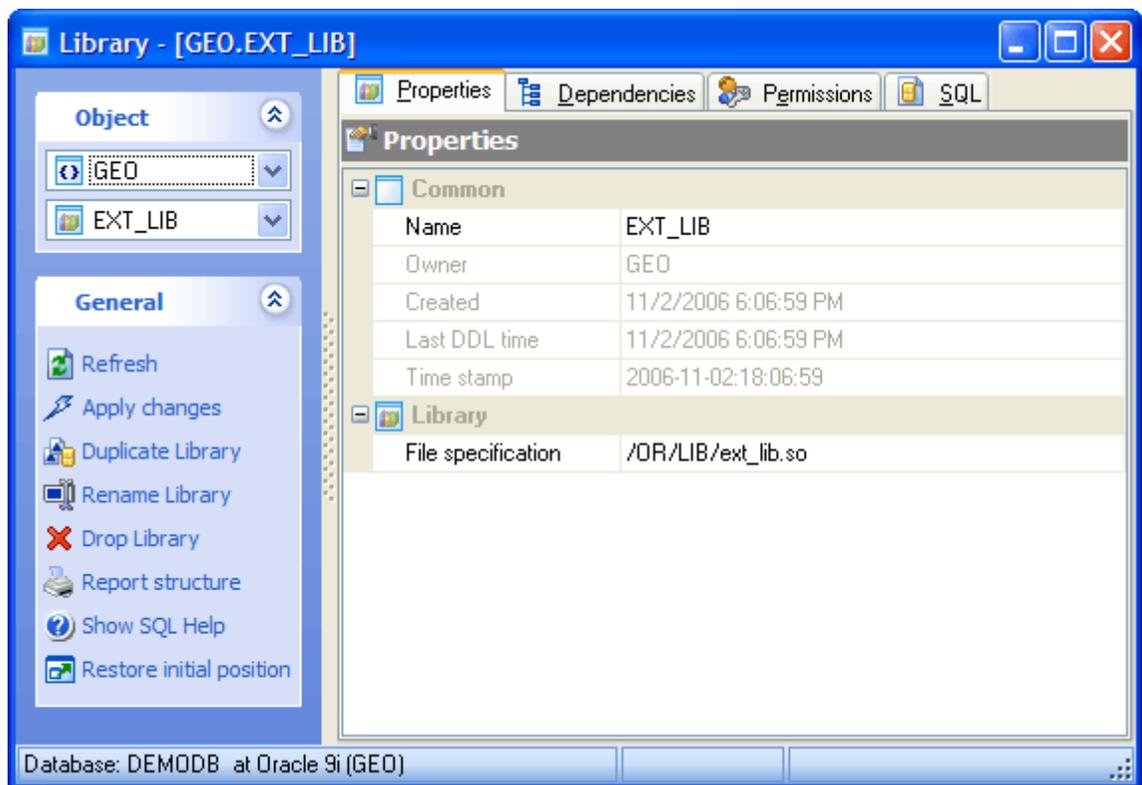
- [Editing library properties](#)<sup>[231]</sup>

**See also:** [Create Library Wizard](#)<sup>[229]</sup>



### 5.23.2.1 Editing library properties

Library Editor provides you with an ability to edit its properties in several clicks.



#### Name

Here you can view and change the library name.

#### Owner

The field displays the owner of the library.

#### Created

The field displays the date the object was created.

#### Last DDL time

Use the field to find the date when the last data definition language (DDL) operation was performed on the current object. The Last DDL time can help you to find if any changes to the object definitions have been made on or after a specific time.

#### File specification

You can change here a string literal, enclosed in single quotation marks. This string should be the path or filename your operating system recognizes as naming the shared library.

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.

## 5.24 Directories

A directory object specifies an alias for a directory on the server file system where external binary file LOBs (BFILEs) and external table data are located. You can use directory names when referring to BFILEs in your PL/SQL code and OCI calls, rather than hard coding the operating system path name, for management flexibility.

All directories are created in a single namespace and are not owned by an individual schema. You can secure access to the BFILEs stored within the directory structure by granting object privileges on the directories to specific users.

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

New directories are created within [Create Directory Wizard](#).<sup>[234]</sup> In order to run the wizard you should either

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

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

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

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

Directories can be edited within [Directory Editor](#).<sup>[236]</sup> In order to run the editor you should either

- select the [directory](#) for editing in the explorer tree (type the first letters of the directory name for quick search);
  - select the [Edit Directory...](#) item from the popup menu
- or
- open the schema in [Schema Editor](#).<sup>[61]</sup> and the [Directory](#) tab there;
  - select the directory to edit;
  - press the **Enter** key or select the [Edit Directory](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#)).

### ■ How can I drop a directory?

To drop a directory:

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

or

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

and confirm dropping in the dialog window.

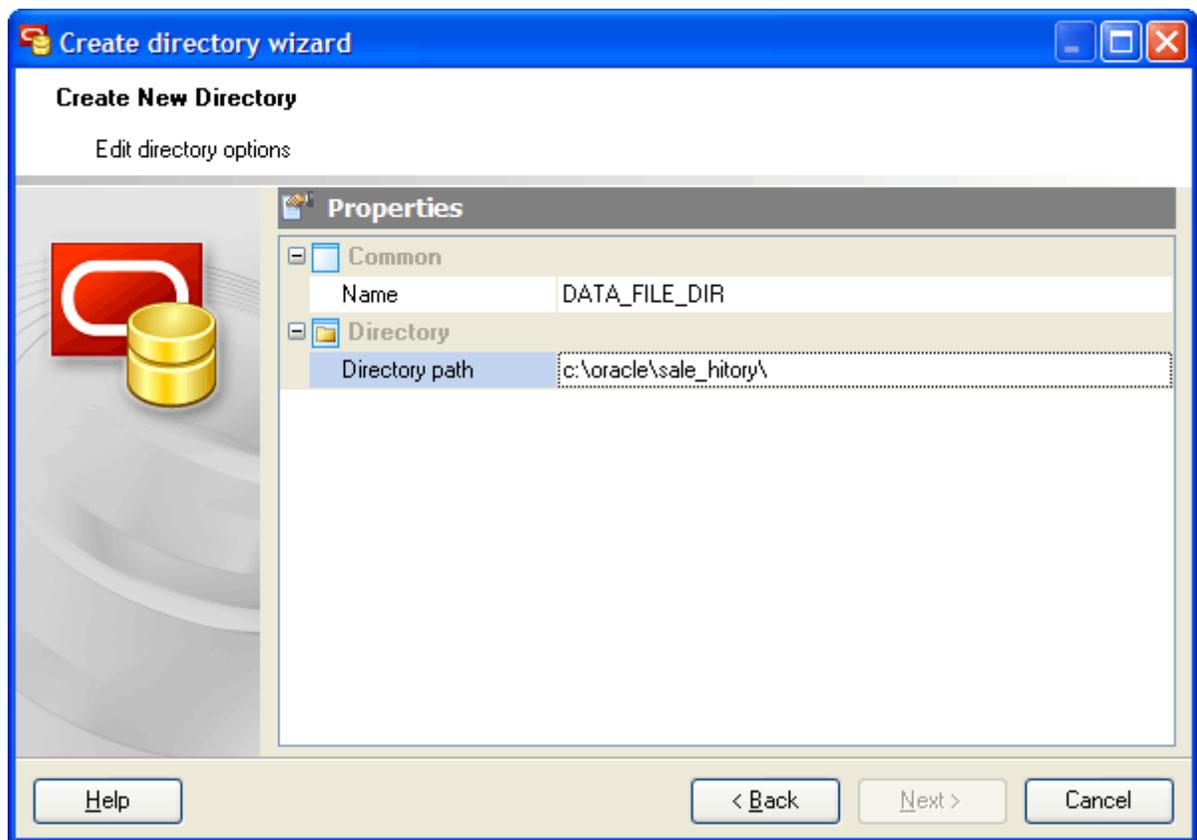
## 5.24.1 Create Directory Wizard

[Create Directory Wizard](#) guides you through the process of creating a new directory. See [How To Create Directory](#)<sup>[233]</sup> for instructions on running this wizard.

The basic principles of Create Object Wizards in Oracle Maestro are explained in a [separate topic](#)<sup>[37]</sup>. Below you will find a description of wizard steps that are unique for the current object.

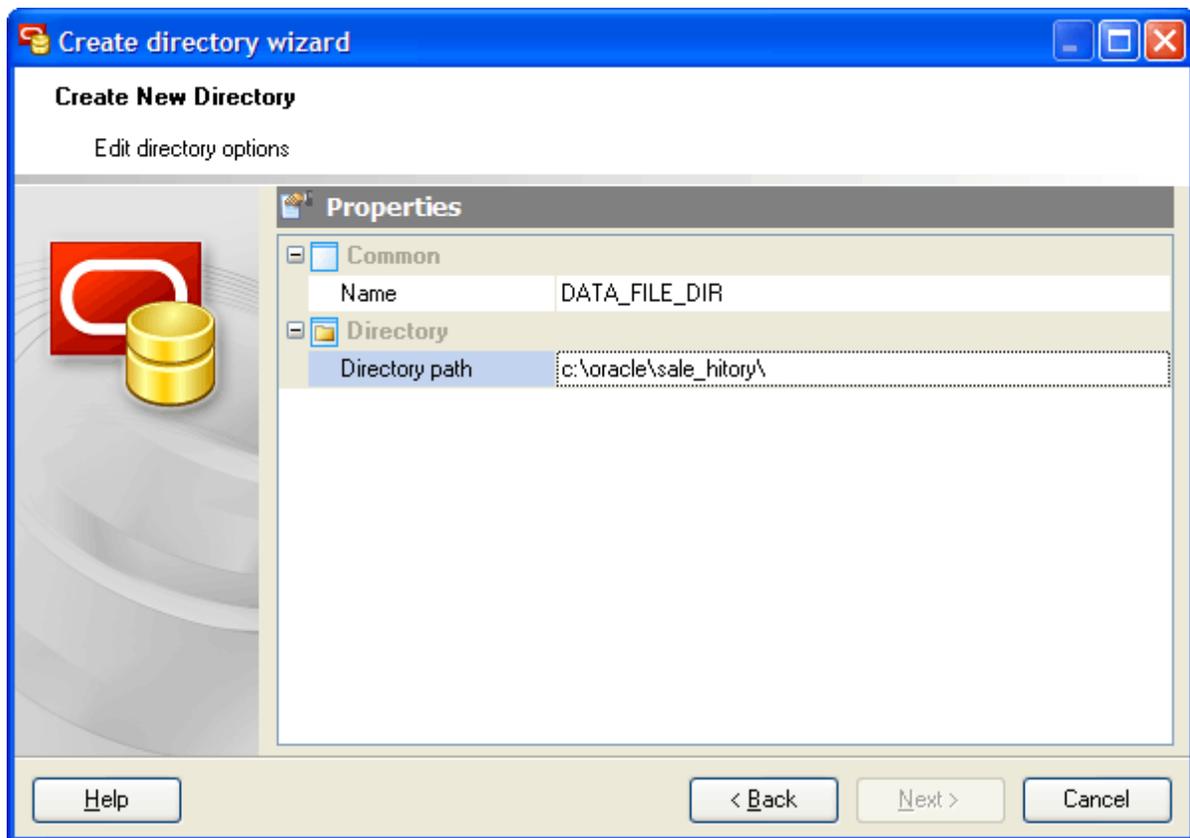
- [Specifying directory properties](#)<sup>[235]</sup>

**See also:** [Directory Editor](#)<sup>[236]</sup>



#### 5.24.1.1 Specifying directory properties

The wizard step was supplied to define common directory properties. The detailed description of the properties you can find below.



#### Name

The field represents the new directory name as it was set on the previous wizard step. The maximum length of directory is 30 bytes. You cannot qualify a directory object with a schema name.

**Note:** Oracle Database does not verify that the directory you specify actually exists. Therefore, take care that you specify a valid directory in your operating system. In addition, if your operating system uses case-sensitive path names, be sure you specify the directory in the correct format. You need not include a trailing slash at the end of the path name.

#### Directory path

Specify the full path name of the operating system directory of the server where the files are located. The single quotes are required, with the result that the path name is case sensitive.

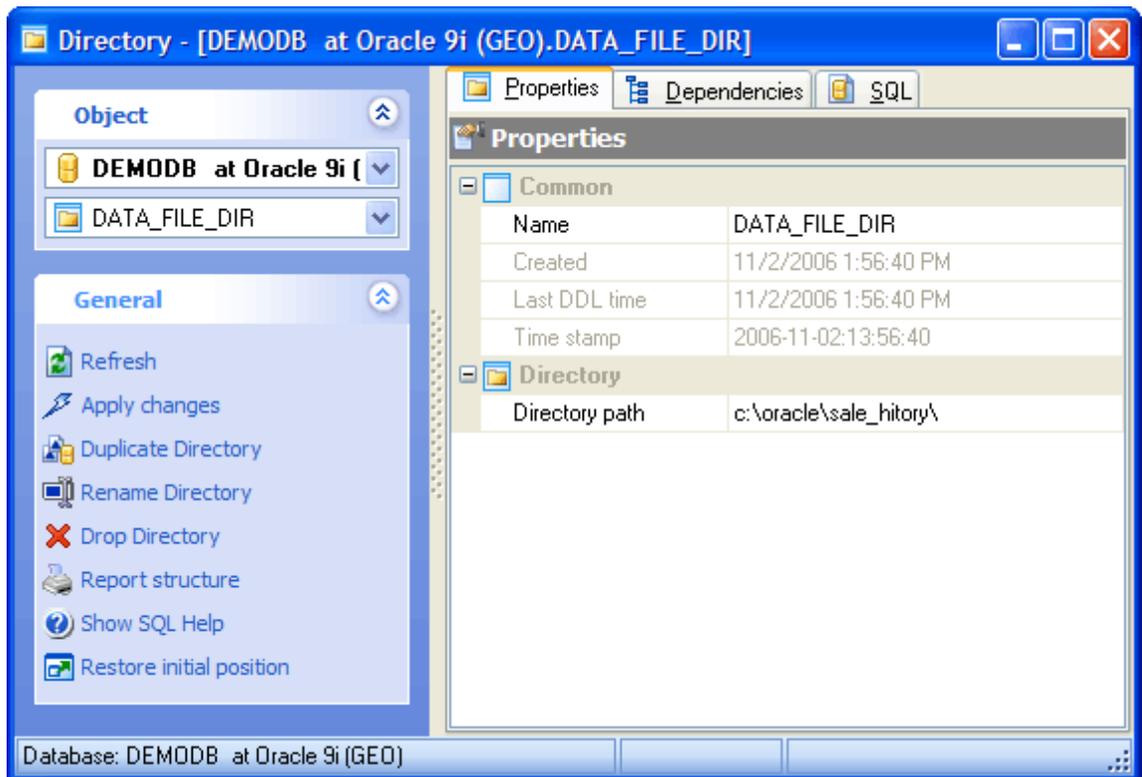
### 5.24.2 Directory Editor

[Directory Editor](#) can be opened automatically after the directory is created and is available on editing (see [Editing Directories](#)<sup>[233]</sup> for details).

The basic principles of Object Editors in Oracle Maestro are explained in a [separate topic](#)<sup>[33]</sup>. Below you will find a description of editor tabs that are unique for the current object.

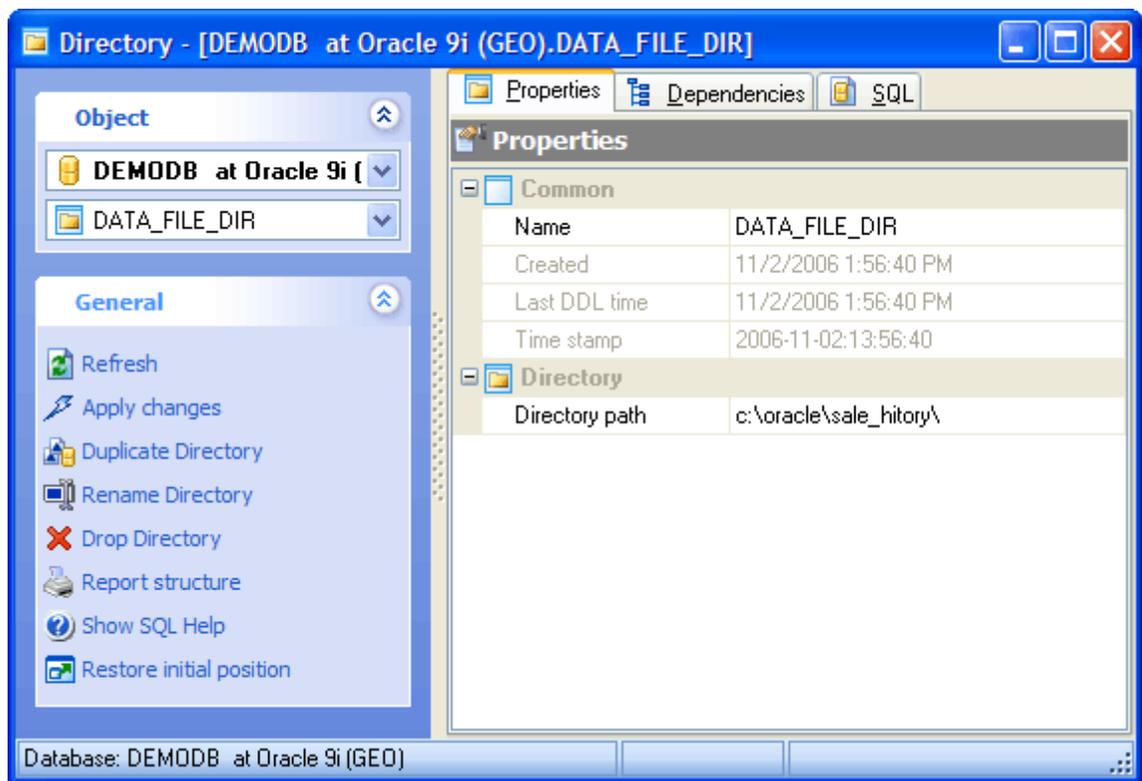
- [Editing directory properties](#)<sup>[237]</sup>

See also: [Create Directory Wizard](#)<sup>[234]</sup>



#### 5.24.2.1 Editing directory properties

Directory Editor provides you with an ability to edit directory properties in a couple clicks.



#### Name

Here you can view and change the directory name.

#### Created

The field displays the date the object was created.

#### Last DDL time

Use the field to find the date when the last data definition language (DDL) operation was performed on the current object. The Last DDL time can help you to find if any changes to the object definitions have been made on or after a specific time.

#### Directory path

Use the field to edit the full path name of the operating system directory of the server where the files are located. The single quotes are required, with the result that the path name is case sensitive.

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.

## 5.25 Database triggers

A database trigger is a stored PL/SQL block associated with the database. Oracle Database automatically executes a trigger when specified conditions occur.

### ■ How can I create a new database trigger?

New database triggers are created within [Create Database Trigger Wizard](#).<sup>[240]</sup> In order to run the wizard you should either

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

To create a new database trigger with the same properties as one of the existing database trigger has:

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

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

Database triggers can be edited within [Database Trigger Editor](#).<sup>[242]</sup> In order to run the editor you should either

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

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

- select the database trigger to rename in the explorer tree;

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

#### ■ **How can I drop a database trigger?**

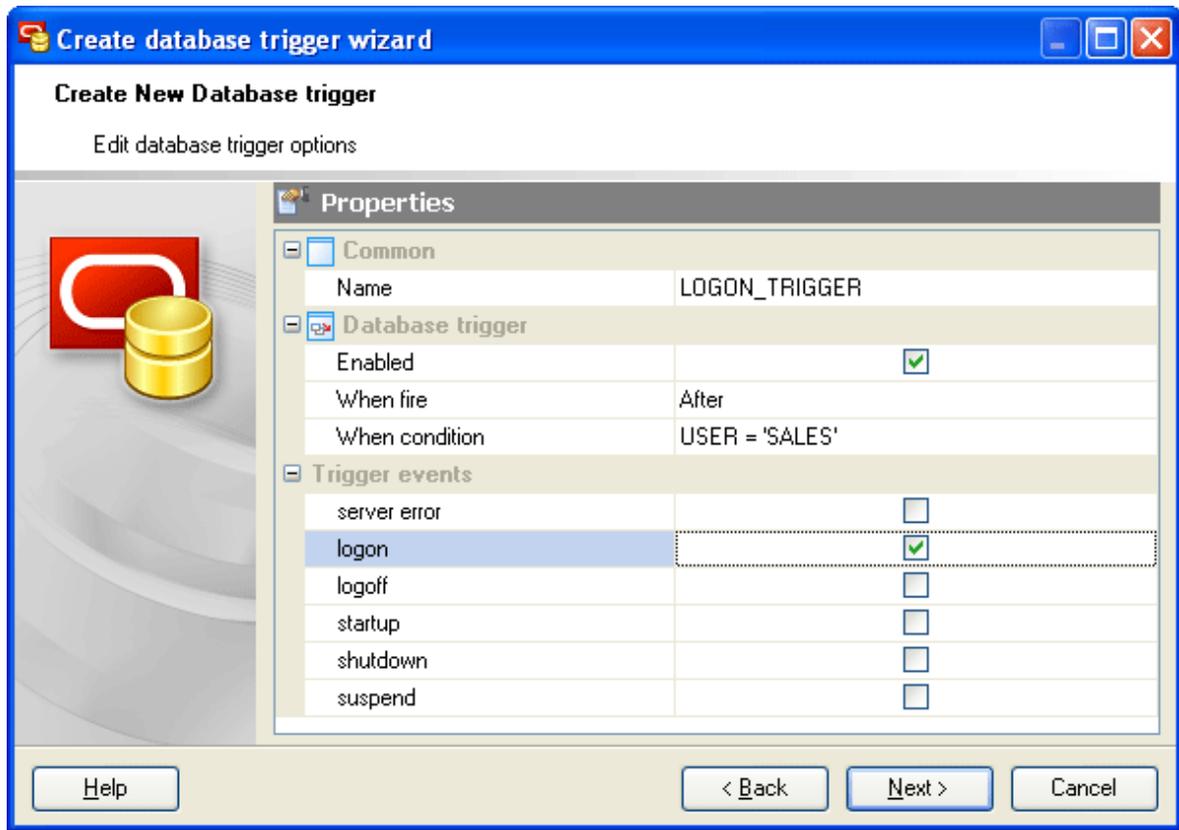
To drop a database trigger:

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

and confirm dropping in the dialog window.

### 5.25.1 Create Database Trigger Wizard

[Create Database Trigger Wizard](#) guides you through the process of creating a new trigger. See [How To Create Database Trigger](#)<sup>[239]</sup> for instructions on running this wizard.



The basic principles of Create Object Wizards in Oracle Maestro are explained in a [separate topic](#)<sup>37</sup>. Below you will find a description of wizard steps that are unique for the current object.

The wizard step was supplied to define common database trigger properties.

#### Name

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

#### Is enabled

Check the option to enable the trigger.

#### When fire (Before, After)

Specify **Before** to cause the database to fire the trigger before executing the triggering event. For row triggers, the trigger is fired before each affected row is changed. Specify **After** to cause the database to fire the trigger after executing the triggering event. For row triggers, the trigger is fired after each affected row is changed.

#### When condition

Specify the trigger condition, which is a SQL condition that must be satisfied for the database to fire the trigger.

#### Server error

Check the box to fire the trigger whenever a server error message is logged.

#### Logon

Specify the option to fire the trigger whenever a client application logs onto the database.

#### Logoff

Use the option to fire the trigger whenever a client application logs off the database.

#### Startup

Specify the clause to fire the trigger whenever the database is opened.

#### Shutdown

Specify the option to fire the trigger whenever an instance of the database is shut down.

#### Suspend

Use the clause to fire the trigger whenever a server error causes a transaction to be suspended.

#### Specifying trigger definition

Use this step to specify the trigger steps to be executed when the trigger fires. You can also do it later using non-modal editor.

### 5.25.2 Database Trigger Editor

[Database Trigger Editor](#) can be opened automatically after the trigger is created and is available on editing (see [Editing Triggers](#)<sup>[239]</sup> for details).

The basic principles of Object Editors in Oracle Maestro are explained in a [separate topic](#)<sup>[39]</sup>. Below you will find a description of editor tabs that are unique for the current object.

[Database Trigger Editor](#) provides you with an ability to edit trigger properties. The [Properties](#) tab allows you to change the trigger name, the trigger definition, and the trigger events.

The [Errors](#) tab displays all the necessary information about object errors. If an error have occurred during the object compilation it appears in the list with some additional properties: [Order](#) (one after another), [Line](#) and [Position](#) (object definition location the error was found out), [Error](#) (corresponding PL/SQL exception).

#### Name

Use the field to rename the database trigger.

#### Is enabled

Check the option to enable the trigger.

#### When fire (Before, After)

Specify [Before](#) to cause the database to fire the trigger before executing the triggering event. For row triggers, the trigger is fired before each affected row is changed. Specify [After](#) to cause the database to fire the trigger after executing the triggering event. For row triggers, the trigger is fired after each affected row is changed.

#### When condition

Specify the trigger condition, which is a SQL condition that must be satisfied for the

database to fire the trigger.

#### Server error

Check the box to fire the trigger whenever a server error message is logged.

#### Logon

Specify the option to fire the trigger whenever a client application logs onto the database.

#### Logoff

Use the option to fire the trigger whenever a client application logs off the database.

#### Startup

Specify the clause to fire the trigger whenever the database is opened.

#### Shutdown

Specify the option to fire the trigger whenever an instance of the database is shut down.

#### Suspend

Use the clause to fire the trigger whenever a server error causes a transaction to be suspended.

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.

## 5.26 Public Synonyms

Synonym is an alternative name for a table, view, sequence, procedure, stored function, package, materialized view, Java class schema object, user-defined object type, or another synonym.

Public synonyms are accessible to all users. However each user must have appropriate privileges on the underlying object in order to use the synonym.

When resolving references to an object, Oracle Database uses a public synonym only if the object is not prefaced by a schema and is not followed by a database link.

### ■ How can I create a new synonym?

New public synonyms are created within [Create Public Synonym Wizard](#)<sup>[245]</sup>.<sup>[240]</sup> In order to run the wizard you should either

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

or

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

or

- open the schema in [Schema Editor](#) and the [Public Synonyms](#) tab there;
- press the **Insert** key or select the [Create New Public Synonym](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#)).

To create a new public synonym with the same properties as one of the existing public synonym has:

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

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

Public synonyms can be edited within [Public Synonym Editor](#)<sup>[247]</sup>. In order to run the editor you should either

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

or

- open the database in [Database Editor](#) and the [Public Synonyms](#) tab there;
- select the public synonym to edit;
- press the **Enter** key or select the [Edit Public Synonym](#) item from

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

You can change the name of the public synonym using the [Rename Public Synonym](#) dialog. To open the dialog you should either

- select the public synonym to rename in the explorer tree;
  - select the [Rename Public Synonym](#) item from the popup menu
- or
- open the database in [Database Editor](#) and the [Public Synonyms](#) tab there;
  - select the public synonym to rename;
  - select the [Rename Public Synonym](#) item from the popup menu (alternatively, you may use the corresponding link of the [Navigation Bar](#)).

#### ■ **How can I drop a synonym?**

To drop a public synonym:

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

and confirm dropping in the dialog window.

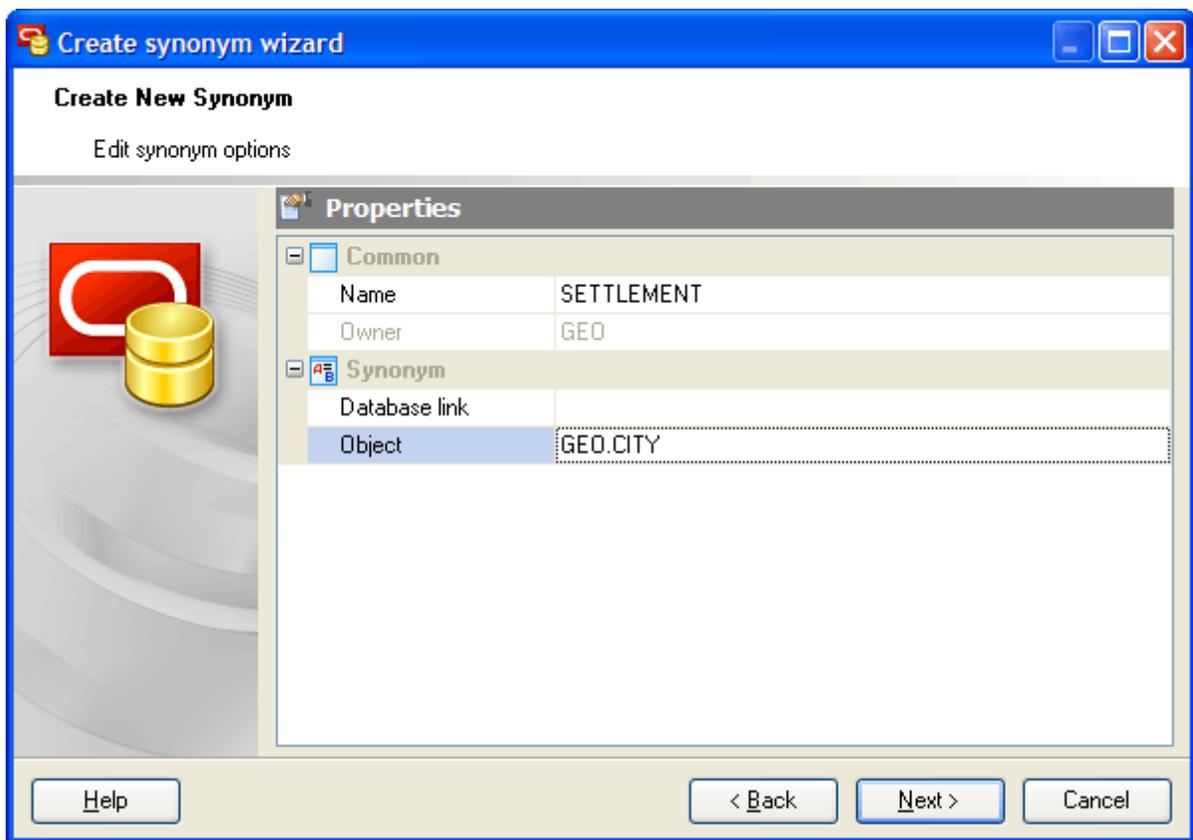
### 5.26.1 Create Public Synonym Wizard

[Create Public Synonym Wizard](#) guides you through the process of creating a new synonym. See [How To Create Public Synonym](#)<sup>[244]</sup> for instructions on running this wizard.

The basic principles of Create Object Wizards in Oracle Maestro are explained in a [separate topic](#)<sup>[37]</sup>. Below you will find a description of wizard steps that are unique for the current object.

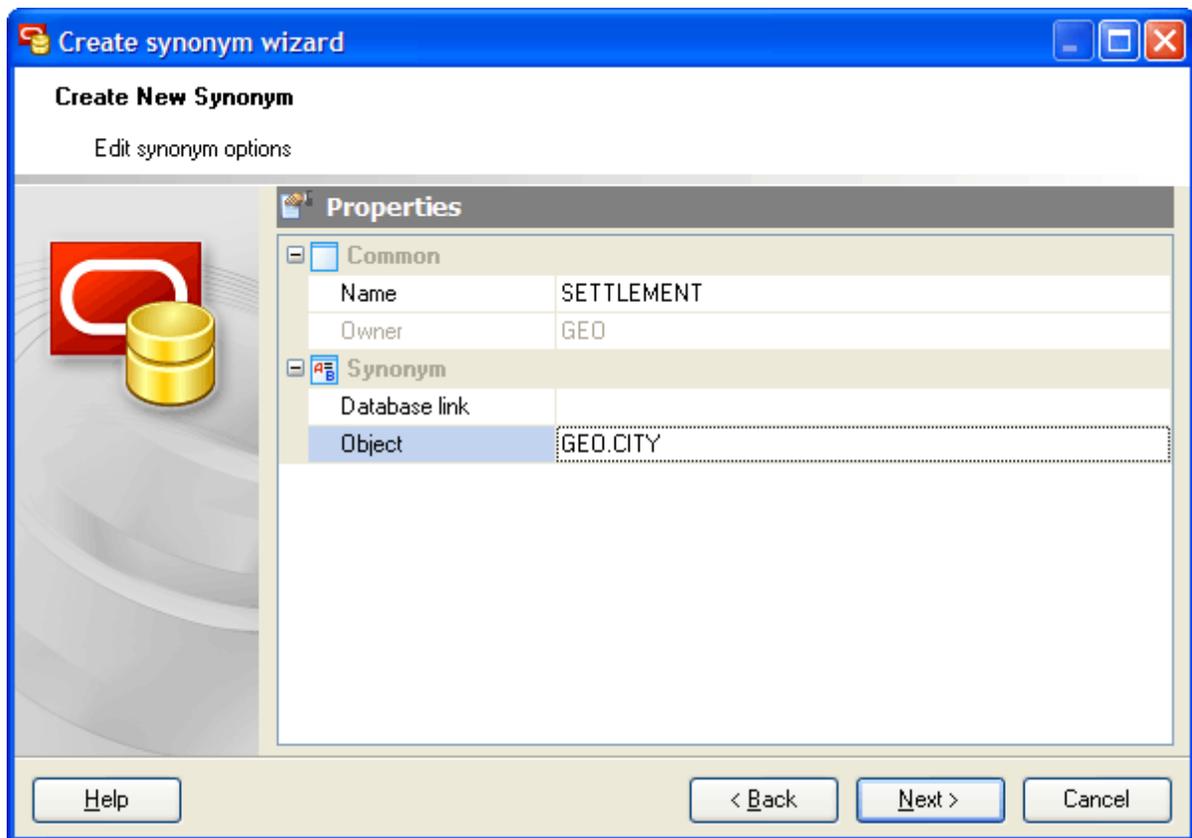
- [Specifying synonym properties](#)<sup>[246]</sup>

**See also:** [Public Synonym Editor](#)<sup>[247]</sup>



#### 5.26.1.1 Specifying synonym properties

The wizard step was supplied to define common public synonym properties. The detailed description of the properties you can find below.



#### Name

The field represents the new public synonym name as it was set on the previous wizard step.

#### Owner

Here you can see the owner of the new public synonym.

#### Database link

You can specify a complete or partial database link to create a synonym for a schema object on a remote database where the object is located.

#### Object

Select the object for the synonym from a drop-down list.

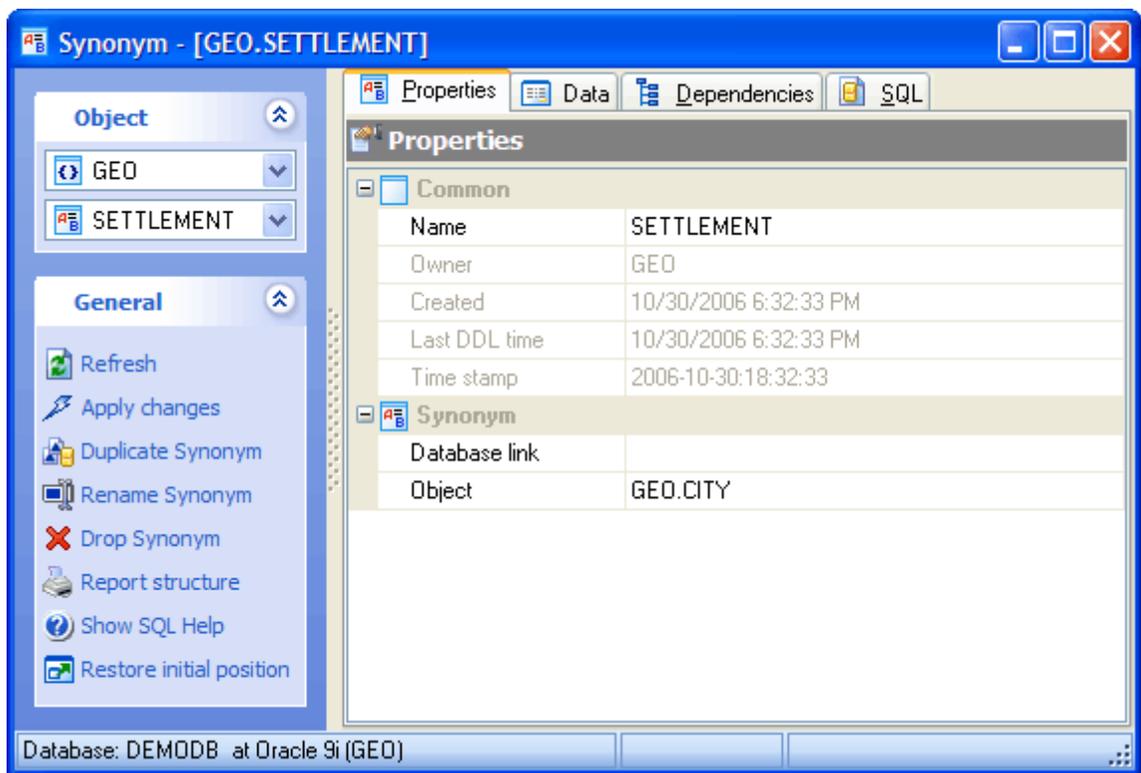
## 5.26.2 Public Synonym Editor

[Public Synonym Editor](#) can be opened automatically after the synonym is created and is available on editing (see [Editing Public Synonyms](#)<sup>[244]</sup> for details).

The basic principles of Object Editors in Oracle Maestro are explained in a [separate topic](#)<sup>[33]</sup>. Below you will find a description of editor tabs that are unique for the current object.

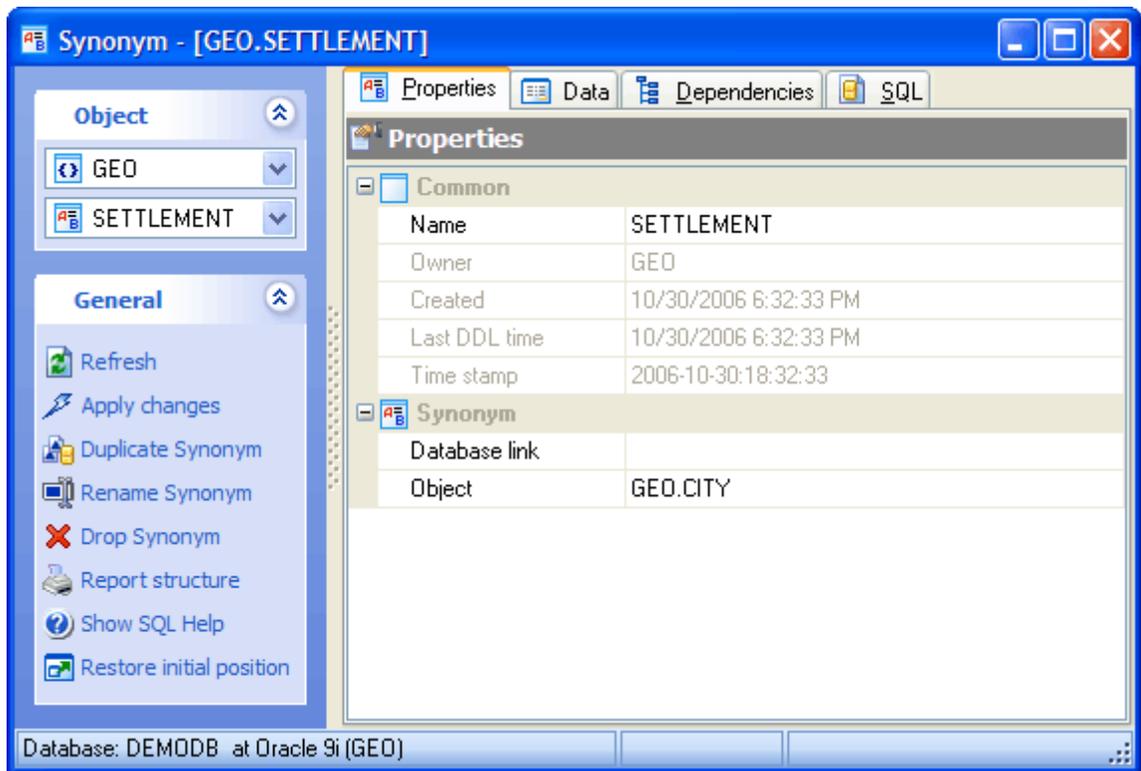
- [Editing synonym properties](#)<sup>[245]</sup>

**See also:** [Create Public Synonym Wizard](#)<sup>[245]</sup>



#### 5.26.2.1 Editing synonym properties

Public Synonym Editor provides you with an ability to edit synonym properties fast and easy.



#### Name

Here you can view and change the public synonym name.

#### Owner

The field displays the owner of the public synonym.

#### Created

The field displays the date the object was created.

#### Last DDL time

Use the field to find the date when the last data definition language (DDL) operation was performed on the current object. The Last DDL time can help you to find if any changes to the object definitions have been made on or after a specific time.

#### Database link

Here you can edit a synonym database link on a remote database where the referencing object is located.

#### Object

Select the object for the synonym from a drop-down list.

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.

## 5.27 Public database links

A database link is a schema object in one database that enables you to access objects on another database. The other database need not be an Oracle Database system. A public database link is available to all users.

### ■ How can I create a new database link?

New public database links are created within [Create Public Database Link Wizard](#)<sup>[251]</sup>,<sup>[240]</sup> In order to run the wizard you should either

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

To create a new public database link with the same properties as one of the existing public database link has:

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

### ■ How can I edit an existing database link?

Public database links can be edited within [Public Database Link Editor](#)<sup>[253]</sup>. In order to run the editor you should either

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

You can change the name of the public database link using the [Rename Public Database Link](#) dialog. To open the dialog you should

either

- select the [Public Database Link](#) to rename in the explorer tree;
- select the [Rename Public Database Link](#) item from the popup menu

or

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

#### ■ **How can I drop a database link?**

To drop a public database link:

- select the [public database link](#) to drop in the explorer tree;
- select the [Drop Public Database Link](#) item from the popup menu

or

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

and confirm dropping in the dialog window.

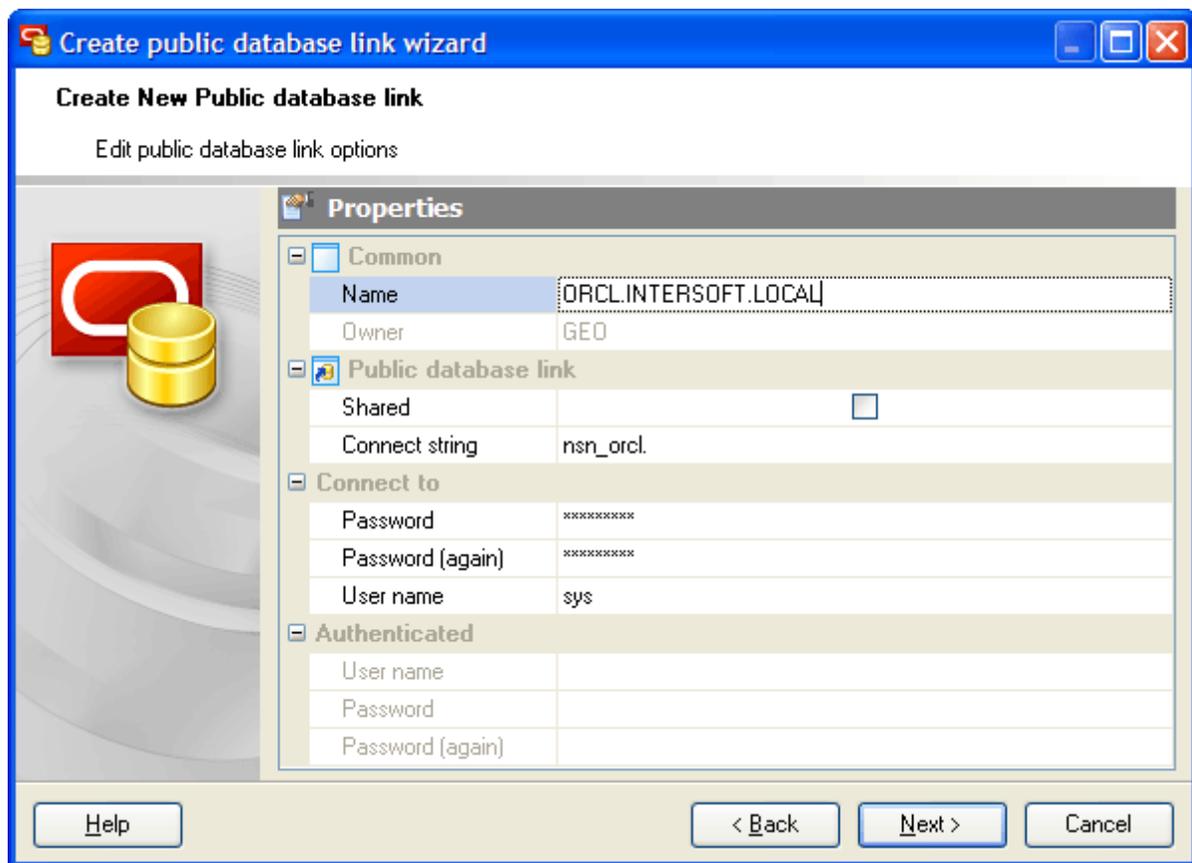
### 5.27.1 Create Public Database Link Wizard

[Create Public Database Link Wizard](#) guides you through the process of creating a new public database link. See [How To Create Public Database Link](#)<sup>[250]</sup> for instructions on running this wizard.

The basic principles of Create Object Wizards in Oracle Maestro are explained in a [separate topic](#)<sup>[37]</sup>. Below you will find a description of wizard steps that are unique for the current object.

- [Specifying link properties](#)<sup>[252]</sup>

**See also:** [Public Database Link Editor](#)<sup>[253]</sup>



### 5.27.1.1 Specifying link properties

The wizard step was supplied to define common public link properties. The detailed description of the properties you can find below.

#### Name

The field represents the new public database link name as it was set on the previous wizard step.

#### Owner

Here you can see the owner of the new public database link.

#### Shared

Check the option to use a single network connection to create a public database link that can be shared among multiple users.

#### Connect string

Specify the service name of a remote database. If you specify only the database name, then Oracle Database implicitly appends the database domain to the connect string to create a complete service name. Therefore, if the database domain of the remote database is different from that of the current database, then you must specify the complete service name.

You also need to specify the **User name** and **Password** used to connect to the remote database

**Authenticated (User name, Password)** (Available for shared public database links)

Specify the username and password on the target instance. This clause authenticates the user to the remote server and is required for security. The specified username and password must be a valid username and password on the remote instance. The username and password are used only for authentication. No other operations are performed on behalf of this user.

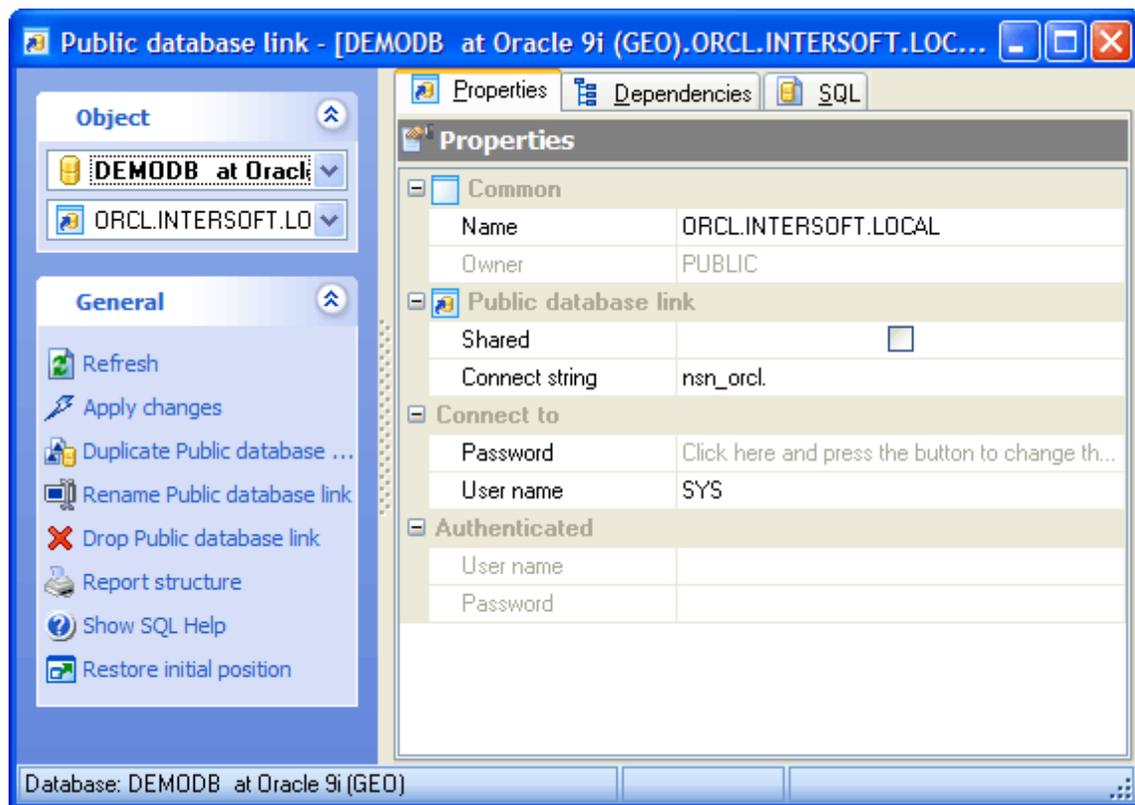
## 5.27.2 Public Database Link Editor

Public Database Link Editor can be opened automatically after the link is created and is available on editing (see [Editing Public Database Links](#)<sup>[250]</sup> for details).

The basic principles of Object Editors in Oracle Maestro are explained in a [separate topic](#)<sup>[39]</sup>. Below you will find a description of editor tabs that are unique for the current object.

- [Editing link properties](#)<sup>[253]</sup>

**See also:** [Create Public Database Link Wizard](#)<sup>[251]</sup>



### 5.27.2.1 Editing link properties

Public Database Link Editor provides you with an ability to edit link properties as easy as it can be.

#### Name

Here you can view and change the public database link name.

#### Owner

Here you can see the owner of the public database link.

### Created

The field displays the date the object was created.

### Last DDL time

Use the field to find the date when the last data definition language (DDL) operation was performed on the current object. The Last DDL time can help you to find if any changes to the object definitions have been made on or after a specific time.

### Shared

Check the option to use a single network connection to create a public database link that can be shared among multiple users.

### Connect string

Specify the service name of a remote database. If you specify only the database name, then Oracle Database implicitly appends the database domain to the connect string to create a complete service name. Therefore, if the database domain of the remote database is different from that of the current database, then you must specify the complete service name.

You also need to specify the [User name](#) and [Password](#) used to connect to the remote database

### Authenticated (User name, Password) (Available for shared public database links)

Specify the username and password on the target instance. This clause authenticates the user to the remote server and is required for security. The specified username and password must be a valid username and password on the remote instance. The username and password are used only for authentication. No other operations are performed on behalf of this user.

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.

## 5.28 Tablespaces

Tablespace is an allocation of space in the database that can contain schema objects.

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

New tablespaces are created within [Create Tablespace Wizard](#)<sup>[256]</sup>. In order to run the wizard you should either

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

To create a new tablespace with the same properties as one of the existing tablespaces has:

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

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

Tablespaces can be edited within [Tablespace Editor](#)<sup>[259]</sup>. In order to run 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 database in [Database 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 may use the corresponding link of the [Navigation Bar](#)).

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

- select the [tablespace](#) to rename in the explorer tree;
  - select the [Rename Tablespace](#) item from the popup menu
- or
- open the database in [Database Editor](#) and the [Tablespaces](#) tab

there;

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

#### ■ **How can I drop a tablespace?**

To drop a tablespace:

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

or

- open the schema in [Schema 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.

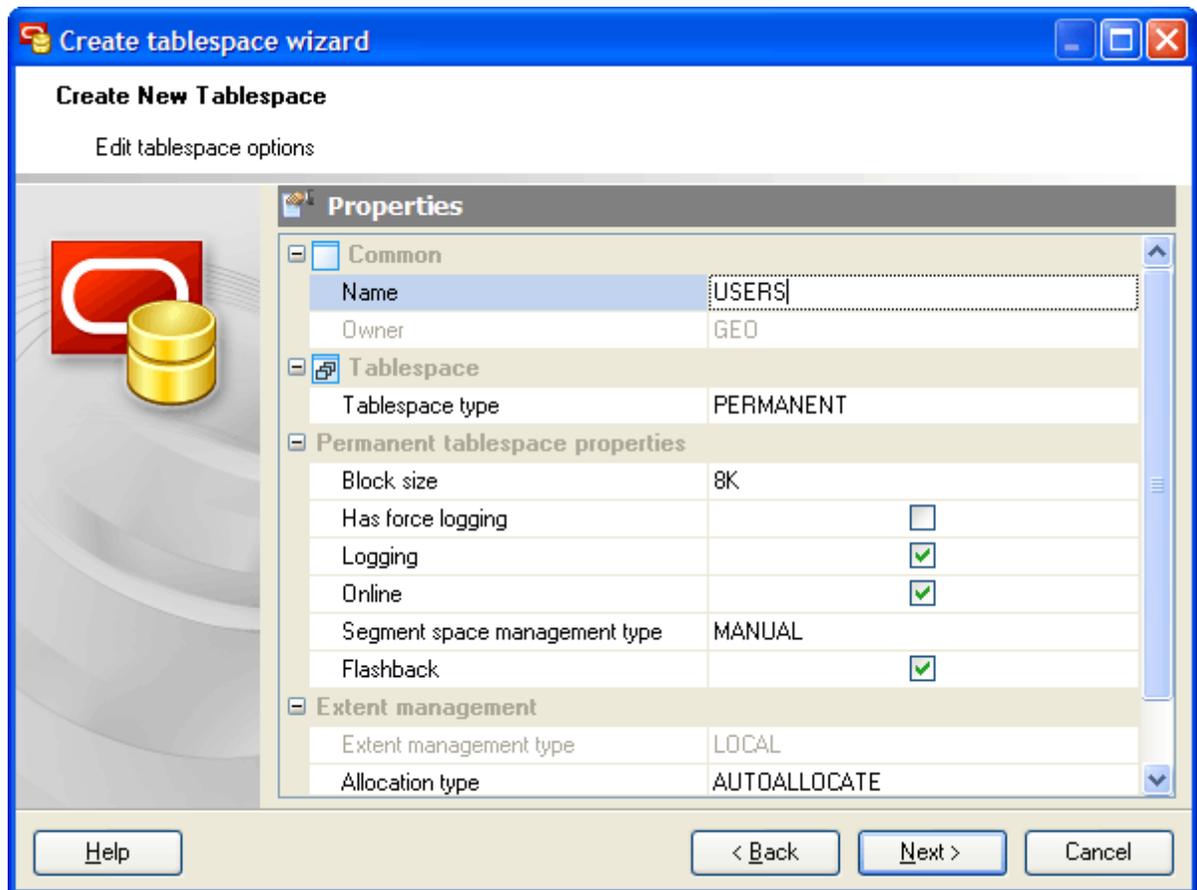
### 5.28.1 Create Tablespace Wizard

[Create Tablespace Wizard](#) guides you through the process of creating a new tablespace. See [How To Create Tablespace](#)<sup>[256]</sup> for instructions on running this wizard.

The basic principles of Create Object Wizards in Oracle Maestro are explained in a [separate topic](#)<sup>[37]</sup>. Below you will find a description of wizard steps that are unique for the current object.

- [Specifying tablespace properties](#)<sup>[257]</sup>
- [Adding tablespace files](#)<sup>[258]</sup>

**See also:** [Tablespace Editor](#)<sup>[259]</sup>



### 5.28.1.1 Specifying tablespace properties

The wizard step was supplied to define common tablespace properties. The detailed description of the properties you can find below.

#### Name

The field represents the new tablespace name as it was set on the previous wizard step.

#### Owner

Here you can see the owner of the new tablespace.

#### Tablespace type (Permanent, Temporary, Undo)

Select **Permanent** to create a tablespace containing persistent schema objects. Objects in permanent tablespaces are stored in datafiles.

Choose **Temporary** to create a tablespace containing schema objects only for the duration of a session. Objects in temporary tablespaces are stored in tempfiles.

Use **Undo** clause to create a undo tablespace which is a type of permanent tablespace used by Oracle Database to manage undo data if you are running your database in automatic undo management mode. Oracle strongly recommends that you use automatic undo management mode rather than using rollback segments for undo.

#### Block size

Use the clause to specify a nonstandard block size for the tablespace.

**Has force logging**

Use this clause to put the tablespace into **force logging** mode. Oracle Database will log all changes to all objects in the tablespace except changes to temporary segments, overriding any **nologging** setting for individual objects. The database must be open and in READ WRITE mode.

This setting does not exclude the **nologging** attribute. That is, you can specify both **force logging** and **nologging**. In this case, **nologging** is the default logging mode for objects subsequently created in the tablespace, but the database ignores this default as long as the tablespace or the database is in **force logging** mode. If you subsequently take the tablespace out of **force logging** mode, then the **nologging** default is once again enforced. You cannot specify **force logging** for an undo or temporary tablespace.

**Logging**

Specify the default logging attributes of all tables, indexes, materialized views, materialized view logs, and partitions within the tablespace. This clause is not valid for a temporary or undo tablespace

**Online**

Check the option to make the tablespace available immediately after creation to users who have been granted access to the tablespace.

**Segment space management type (Manual, Auto)**

The option is relevant only for permanent, locally managed tablespaces. It lets you specify whether Oracle Database should track the used and free space in the segments in the tablespace using free lists or bitmaps. This clause is not valid for a temporary tablespace.

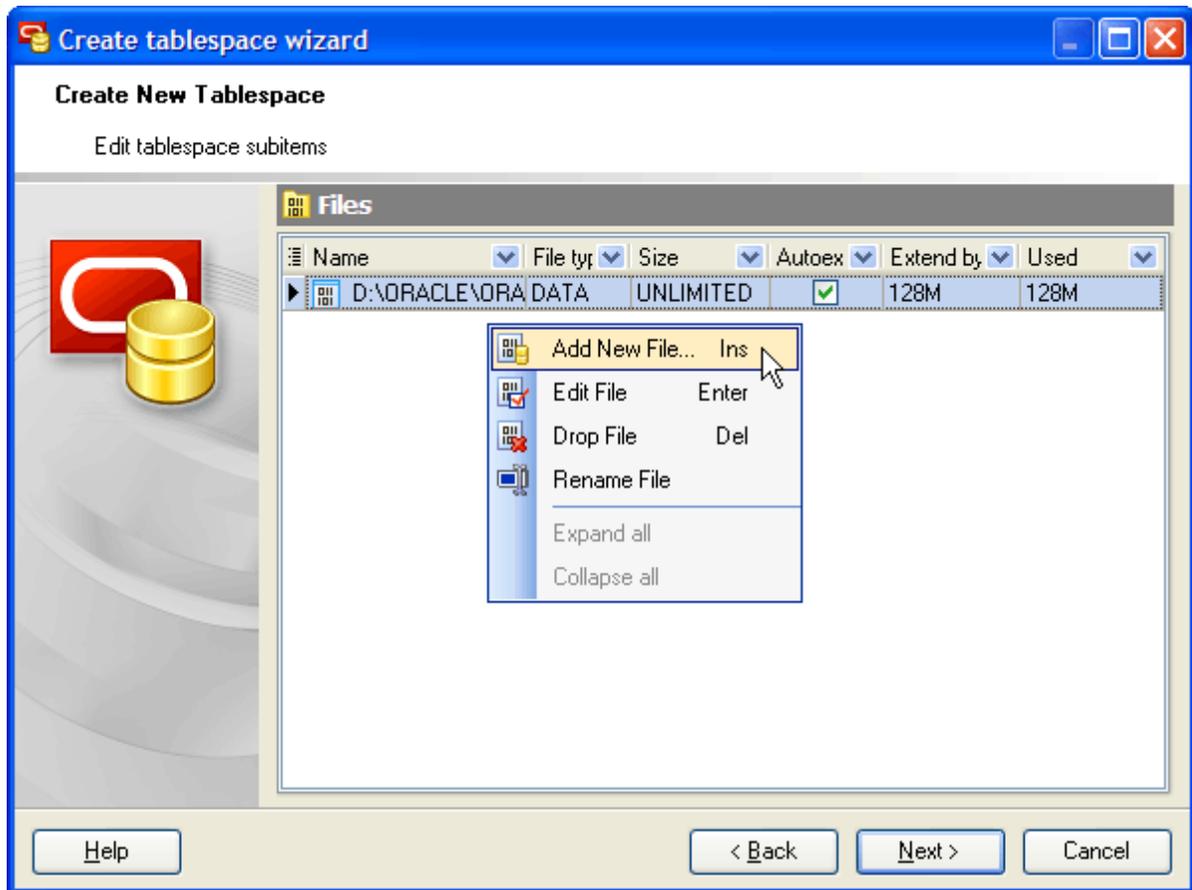
**Flashback** (This clause is not valid for **temporary** or **undo** tablespaces)

Check the option to put the tablespace in **flashback** mode. Oracle Database will save Flashback log data for this tablespace and the tablespace can participate in a **flashback database** operation.

You can also use the wizard to specify **Extent management** properties.

### 5.28.1.2 Adding tablespace files

The wizard step allows you to create tablespace files during the tablespace creation. Just press **Insert** or use popup menu to create new, edit or drop the selected file.



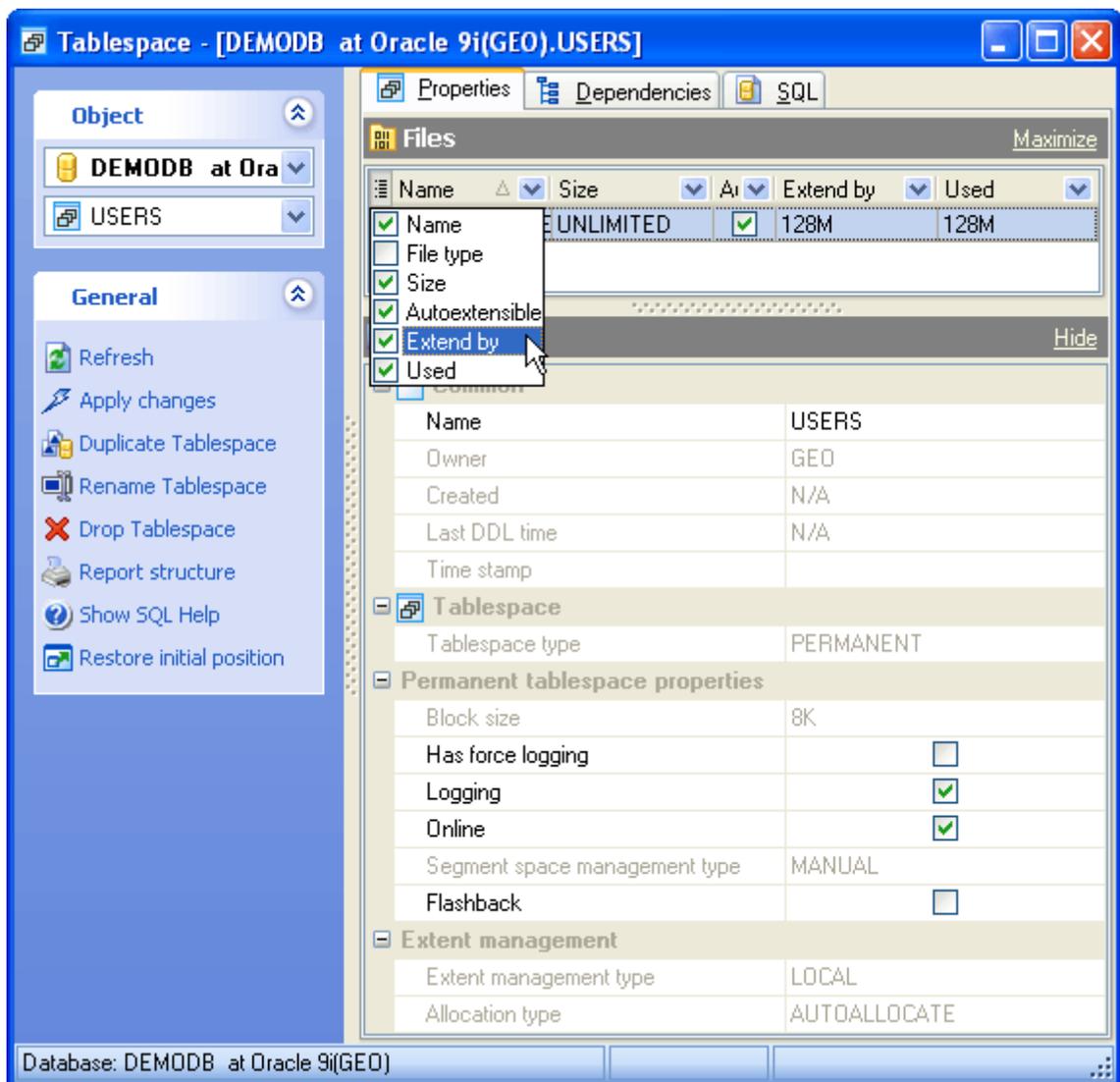
### 5.28.2 Tablespace Editor

Tablespace Editor can be opened automatically after the tablespace is created and is available on editing (see [Editing Tablespaces](#)<sup>[256]</sup> for details).

The basic principles of Object Editors in Oracle Maestro are explained in a [separate topic](#)<sup>[39]</sup>. Below you will find a description of editor tabs that are unique for the current object.

- [Editing tablespace properties](#)<sup>[260]</sup>

**See also:** [Create Tablespace Wizard](#)<sup>[256]</sup>



### 5.28.2.1 Editing tablespace properties

Tablespace Editor consists of Files, Quotas, Segments and Properties tabs:

#### Files

The tab contains tablespace files. Use popup menu of the grid to add a new and edit existing files. Here you can view and change the size of tablespace file and the tablespace name.

#### Created

The field displays the date the object was created.

#### Last DDL time

Use the field to find the date when the last data definition language (DDL) operation was performed on the current object. The Last DDL time can help you to find if any changes to the object definitions have been made on or after a specific time.

Here you can also find the [Tablespace type](#), [Block size](#), and tablespace [Extent management](#) properties

**Has force logging**

Use this clause to put the tablespace into **force logging** mode. Oracle Database will log all changes to all objects in the tablespace except changes to temporary segments, overriding any **nologging** setting for individual objects. The database must be open and in READ WRITE mode.

This setting does not exclude the **nologging** attribute. That is, you can specify both **force logging** and **nologging**. In this case, **nologging** is the default logging mode for objects subsequently created in the tablespace, but the database ignores this default as long as the tablespace or the database is in **force logging** mode. If you subsequently take the tablespace out of **force logging** mode, then the **nologging** default is once again enforced. You cannot specify **force logging** for an undo or temporary tablespace.

 **Logging**

Specify the default logging attributes of all tables, indexes, materialized views, materialized view logs, and partitions within the tablespace. This clause is not valid for a temporary or undo tablespace

 **Online**

Check the option to make the tablespace available immediately after creation to users who have been granted access to the tablespace.

**Segment space management type (Manual, Auto)**

The option is relevant only for permanent, locally managed tablespaces. It lets you specify whether Oracle Database should track the used and free space in the segments in the tablespace using free lists or bitmaps. This clause is not valid for a temporary tablespace.

 **Flashback** (This clause is not valid for **temporary** or **undo** tablespaces)

Check the option to put the tablespace in **flashback** mode. Oracle Database will save Flashback log data for this tablespace and the tablespace can participate in a **flashback database** operation.

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 Queries

Oracle Maestro provides several tools for working with SQL queries:

- [SQL Editor](#)<sup>[264]</sup> for editing the query text directly and executing SELECT queries;
- [Visual Query Builder](#)<sup>[269]</sup> for building SELECT, INSERT, UPDATE and DELETE queries visually;
- [SQL Script Editor](#)<sup>[316]</sup> for executing SQL scripts.

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](#).

Oracle 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.

## 6.1 SQL Editor

[SQL Editor](#) is a tool for creating and executing SELECT 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>[262]</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>[374]</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>[266]</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>[265]</sup> is also at your disposal.

SQL Editor allows you to use [Visual Query Builder](#)<sup>[269]</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

Oracle 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>[394]</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.

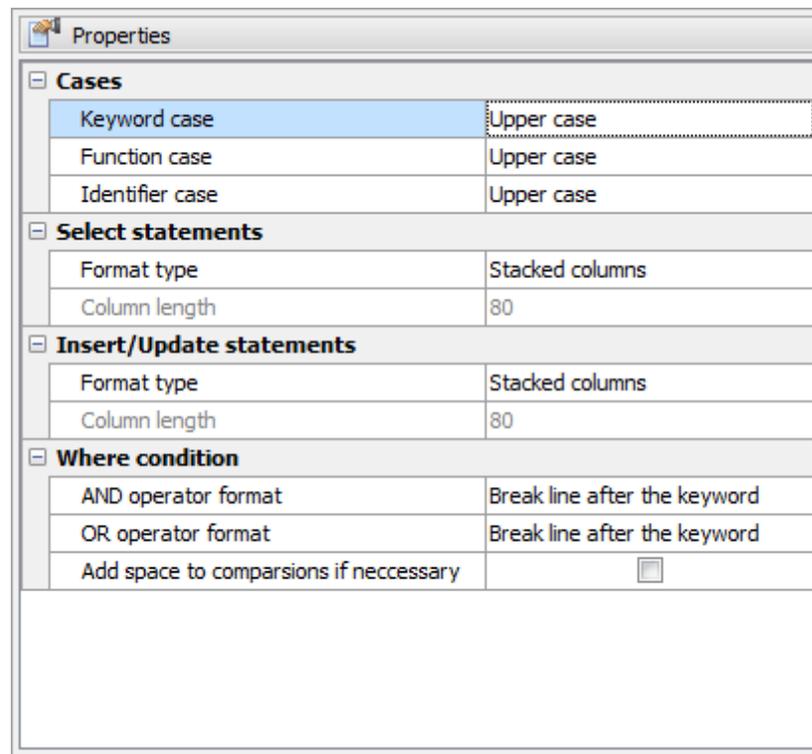
**See also:** [Visual Query Builder](#)<sup>[269]</sup>, SQL Script Editor, [SQL Editor Options](#)<sup>[374]</sup>

### 6.1.1 SQL Formatter

Oracle 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.



## 6.1.2 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>278</sup> section.

The screenshot displays the Oracle Maestro SQL Editor interface. The left sidebar contains navigation and management options for the database 'sdb\_demo at sun'. The main window shows a SQL query and its results.

**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

**SQL Query:**

```

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:**

	GAME_DATE	HOME_TEAM	HOME_TEAM_SCORE
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

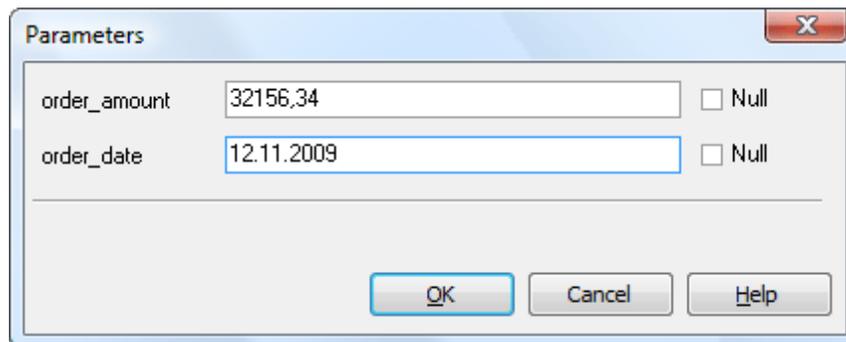
**Information:** 1315 rows fetched (0,56 sec)

Database: sdb\_demo at sun

### 6.1.3 Query Parameters

Both [SQL Editor](#)<sup>[264]</sup> and [Visual Query Builder](#)<sup>[269]</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>[371]</sup> tab of Oracle Maestro Options.

## 6.2 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.

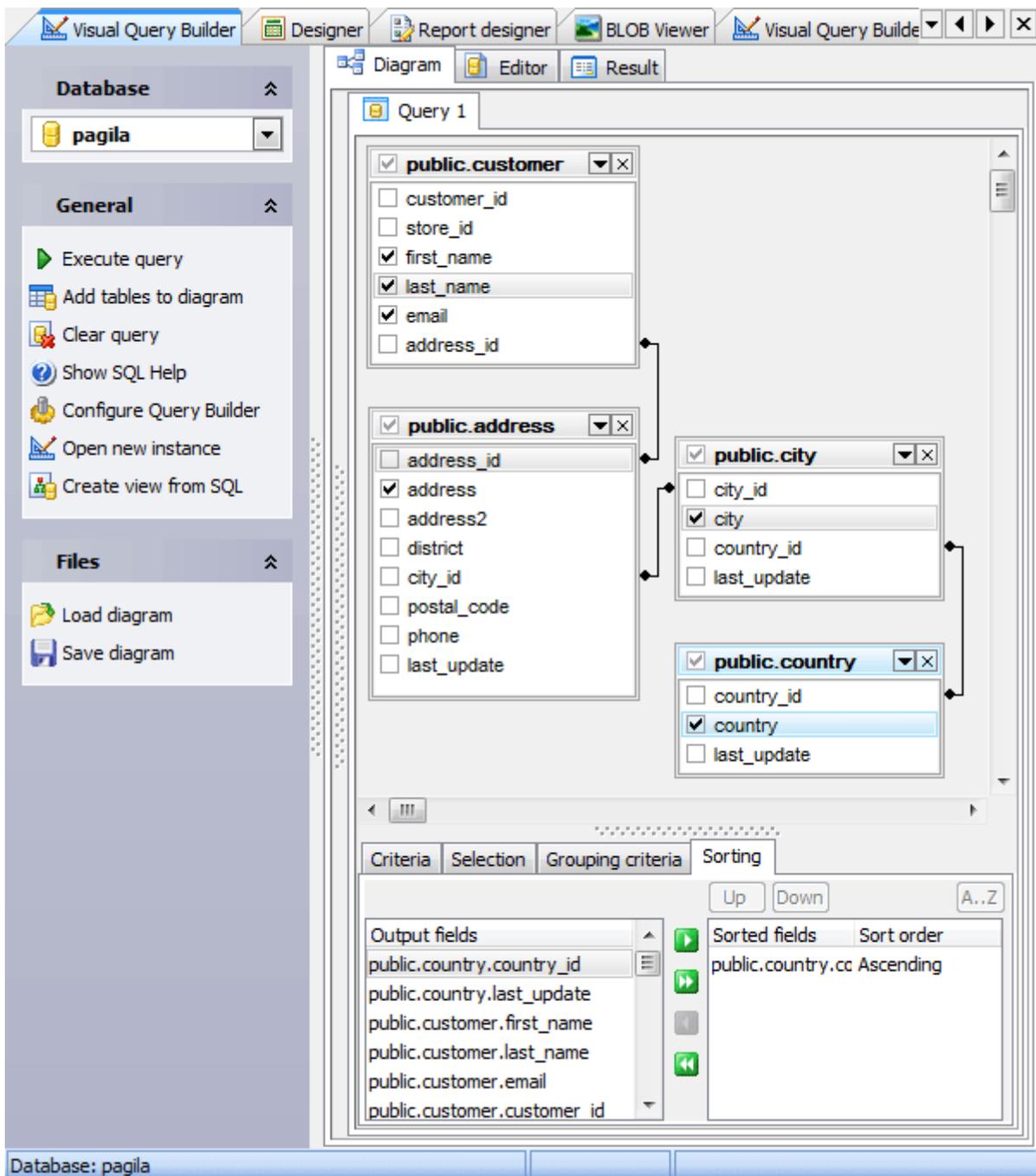
The most popular query management actions (creating, editing, deleting) are covered by the corresponding [topic](#)<sup>[262]</sup>.

Builder consists of 3 tabs:

- [Diagram](#)<sup>[270]</sup> - to create a query from a graphical interface,
- [Editor](#)<sup>[275]</sup> - to modify the query text before its executing,
- [Result](#)<sup>[276]</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>[96]</sup>.

**See also:** [SQL Editor](#)<sup>[264]</sup>, [Visual Query Builder Options](#)<sup>[376]</sup>, [Query Parameters](#)<sup>[268]</sup>



### 6.2.1 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.

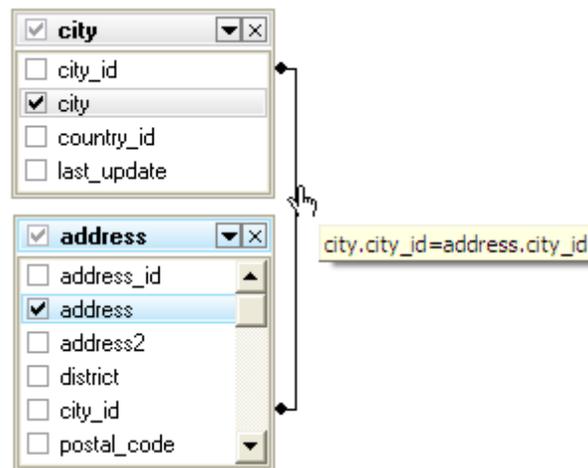


To include all the table fields, tick off the option box to the left of the table caption. In case none fields is included, the SQL statement is generated as `SELECT * FROM <Table_Name>`, i.e. all the fields are selected.

To remove the fields from the query, uncheck the corresponding boxes.

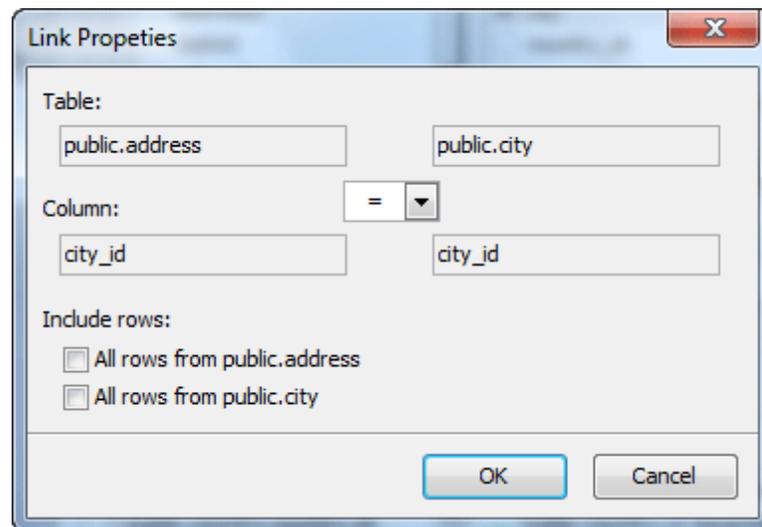
#### ■ **Join tables if necessary**

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.

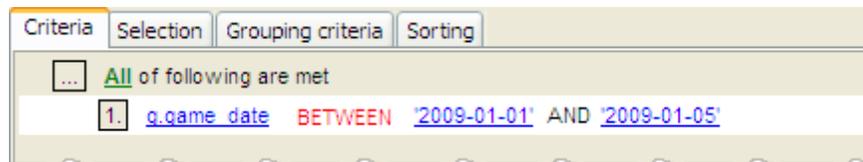
To delete all the links of an object, click the '-' button next to the object alias. To insert a point to the link line, select the [Insert Point](#) item from the popup menu, and the new point will appear. Using the point you can move the link line. It does not cause any changes in the query but makes 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.

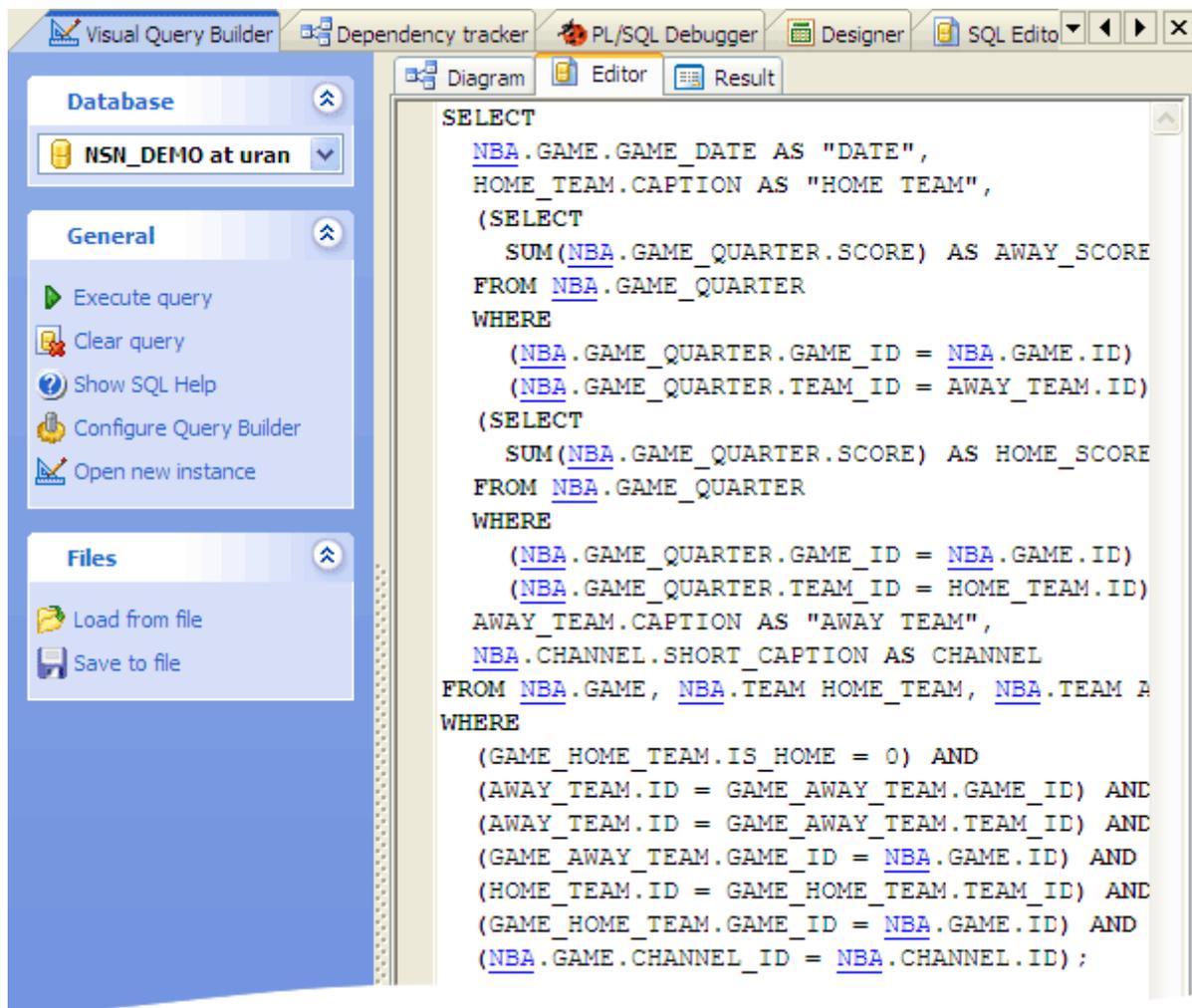
**Note:** The column names from the first `SELECT` statement are used as the column names for the results returned.

Selected columns listed in corresponding positions of each `SELECT` statement should have the same data type.

### 6.2.2 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](#).



### 6.2.3 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.

The screenshot displays the Oracle Maestro interface with the following components:

- 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
- Query Result:**

	* 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
- Status:** Records fetched: 599
- Information:** 599 rows fetched (0,19 sec)

## 7 Data Management

Query results and table data are displayed on the [Data](#)<sup>[72]</sup> or [Result](#)<sup>[266]</sup> tabs of [Table Editor](#)<sup>[68]</sup>, [SQL Editor](#)<sup>[264]</sup> or [Visual Query Builder](#)<sup>[269]</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>[279]</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>[295]</sup> allows you to export data to various formats, including MS Excel, MS Access, RTF, HTML, PDF and more.
- [Get SQL Dump](#)<sup>[302]</sup> exports data to the SQL script as a number of INSERT statements.
- [Import Data](#)<sup>[305]</sup> provides you with possibility to import data from MS Excel, MS Access, DBF, XML, TXT, and CSV.
- [Edit BLOB](#)<sup>[290]</sup> allows you to view and edit the content of BLOB and TEXT fields.

## 7.1 Data View

Oracle Maestro represents all data (stored in tables and views, results of queries and procedures) in [grid](#)<sup>[280]</sup> or in [info cards](#)<sup>[285]</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. To reset grid to default settings, open the Data tab when holding the **Ctrl** key.

	CUST_NO	CUSTOMER	CONTACT_FIRST	CONTACT_LAST	PHONE_NO	AD
1	1001	Signature Design	Dale J.	Little	(619) 530-2710	15
2	1002	Dallas Technologies	Glen	Brown	(214) 960-2233	P.
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	33
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

Records fetched: 15/15      LIMIT 1000 OFFSET 0

### 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>[286]</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>[383]</sup>.

### Navigation bar

The Data management group of the Navigation bar allows to invoke [Data Input Form](#)<sup>[286]</sup>, [SQL Editor](#)<sup>[264]</sup> with SELECT query, [Data Export](#)<sup>[295]</sup>, and [Data Import](#)<sup>[305]</sup> modules using corresponding links, also get [SQLdump](#)<sup>[302]</sup> of the current data set and print current data with enabled preview in WYSIWYG mode.

**See also:** [Table Editor](#)<sup>[69]</sup>, [SQL Editor](#)<sup>[264]</sup>, [Visual Query Builder](#)<sup>[269]</sup>

### 7.1.1 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)

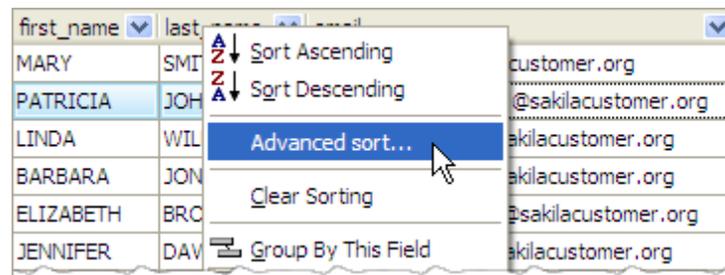
By default, the grid automatically switches to the simple mode for queries returning more than 5000 records (the number can be customized in the [Options](#) dialog).

The following abilities are not available in the simple grid mode:

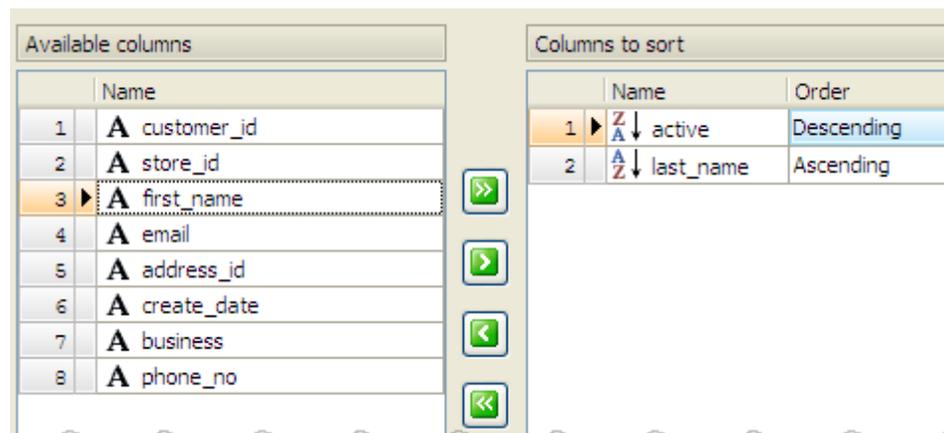
■ **Sorting data (only in the full grid mode)**

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>[287]</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.

city_id	address	last_update	phone
<input type="checkbox"/>	address_id	MySakila Drive	15.02.2006 4:45:30
<input checked="" type="checkbox"/>	address	MySQL Boulevard	15.02.2006 4:45:30
<input type="checkbox"/>	address2	Workhaven Lane	15.02.2006 4:45:30 14033335568
<input type="checkbox"/>	district		
<input checked="" type="checkbox"/>	city	11 Lillydale Drive	15.02.2006 4:45:30 6172235589
<input type="checkbox"/>	postal_code	13 Hanoi Way	15.02.2006 4:45:30 28303384290
<input checked="" type="checkbox"/>	phone	21 Loja Avenue	15.02.2006 4:45:30 838635286649
<input checked="" type="checkbox"/>	last_update	2 Joliet Street	15.02.2006 4:45:30 448477190408

#### ■ Columns reordering

To reorder columns, use drag-n-drop.

### ■ Grouping records

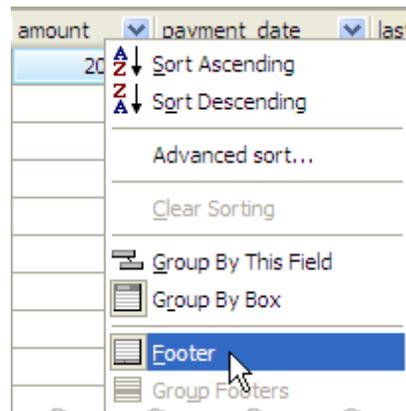
You can group grid data by any of the columns by dragging the column header to the destination area. Now all the records are displayed as subnodes to the grouping row value as shown in the picture. To reverse grouping, just drag the column name from the upper area back.

The screenshot displays the Oracle Maestro interface with a data grid grouped by 'round' and 'date'. The columns are ID, team1ID, team2ID, score1, score2, refereeID, and comments. The status bar shows 'Records fetched: 380' and an information popup shows '380 rows fetched ( 0,64 sec)'.

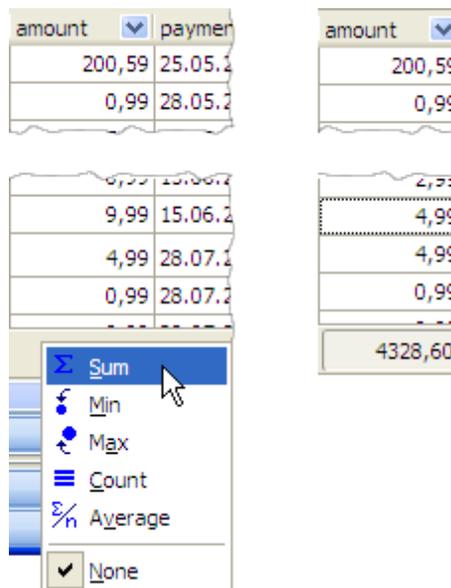
ID	team1ID	team2ID	score1	score2	refereeID	comments
round : 1						
round : 2						
round : 3						
date : 24.08.2004						
date : 25.08.2004						
24	6	2	3	0	8	Jeffers brukte 34 minutter pa a vinne
22	1	4	3	0	17	I den hundrede kampen i alle konkurra
29	16	5	1	2	19	Southamptons slapp inn mal i sin 11. 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						

### ■ 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.



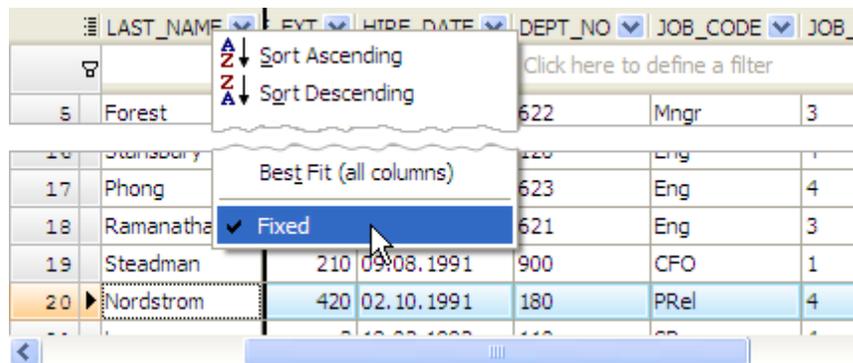
For grouped data use **Group Footers**.

### ■ 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.

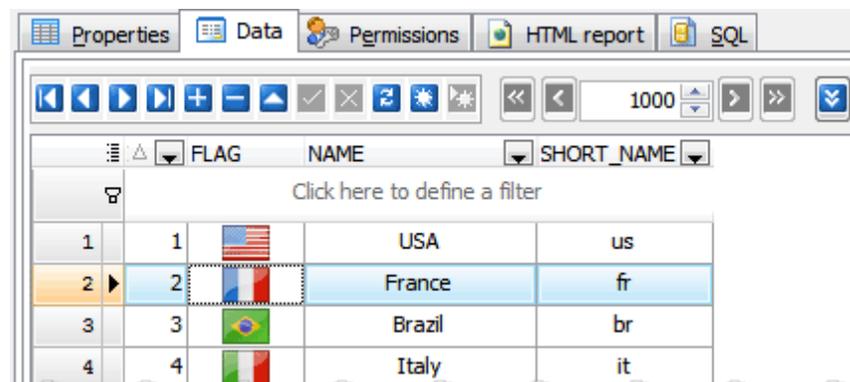


### ■ Row numbering

There is also a possibility to display row numbers in grids. You can [adjust](#)<sup>[384]</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>[290]</sup> (see below).

### ■ Working with BLOBs

To [edit a BLOB field](#)<sup>[290]</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).

## 7.1.2 Working with info cards

**Info cards** correspond to the records. You can [filter records by custom conditions](#)<sup>[287]</sup> and edit data directly in info cards or with [Data Input Form](#)<sup>[286]</sup>.

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

### 7.1.3 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 pop-up menu or **Ctrl+Alt+D** shortcut.

language_id	* name	* last_update
1	English	15.02.2006 10
2	Italian	15.02.2006 10
3	Japanese	15.02.2006 10
4	Mandarin	15.02.2006 10
5	French	15.02.2006 10
6	German	15.02.2006 10

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.

Controls containing values of primary and foreign key columns are marked with the 'gold key' and 'silver key' images accordingly. Controls containing values of required (NOT NULL) columns are marked with a red asterisk.

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.

### 7.1.4 Data filtering

Oracle 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**

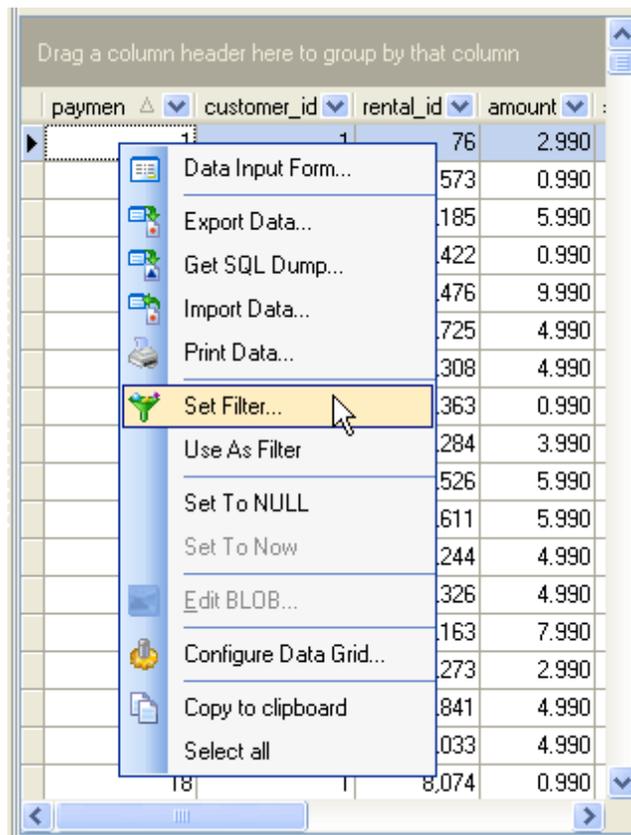
Use the drop-down button in the column caption area to filter records by the selected column value(s) or enter the filter condition directly in [the filter row](#)<sup>388</sup>.

#### 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>[388]</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>[264]</sup> and save as a query. You can also use [Generate query](#) link on the Navigation bar.

**See also:** [Data View](#)<sup>[279]</sup>, [Table Editor](#)<sup>[68]</sup>, [SQL Editor](#)<sup>[264]</sup>, [Visual Query Builder](#)<sup>[269]</sup>

## 7.2 BLOB Editor

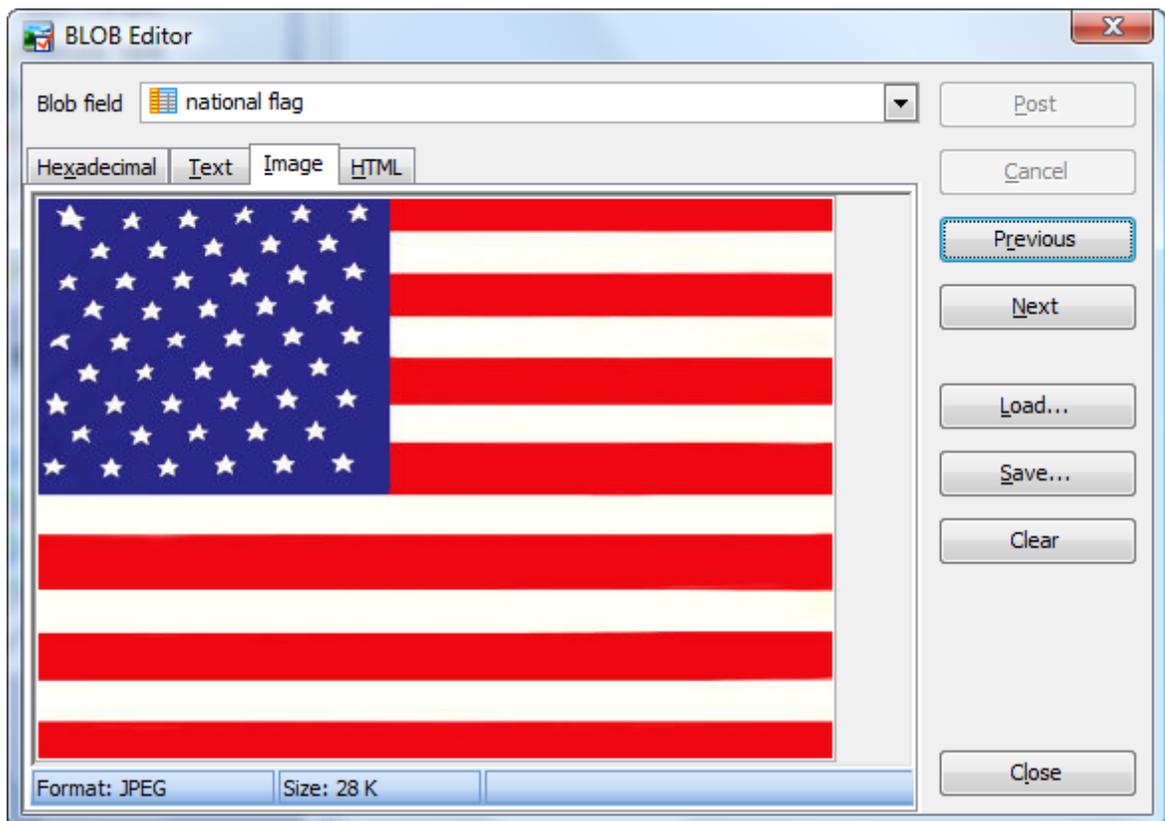
BLOB Editor is a tool to view and edit BLOB data in the following ways: [hexadecimal dump](#)<sup>[291]</sup>, [plain text](#)<sup>[291]</sup>, [graphical image](#)<sup>[290]</sup>, [HTML page](#)<sup>[292]</sup>, or [PDF document](#)<sup>[293]</sup>. BLOB Editor is invoked from [data grid](#)<sup>[279]</sup> of any [table editor](#)<sup>[69]</sup> or the result tab of [SQL Editor](#)<sup>[264]</sup> and [Visual Query Builder](#)<sup>[269]</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>[321]</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.

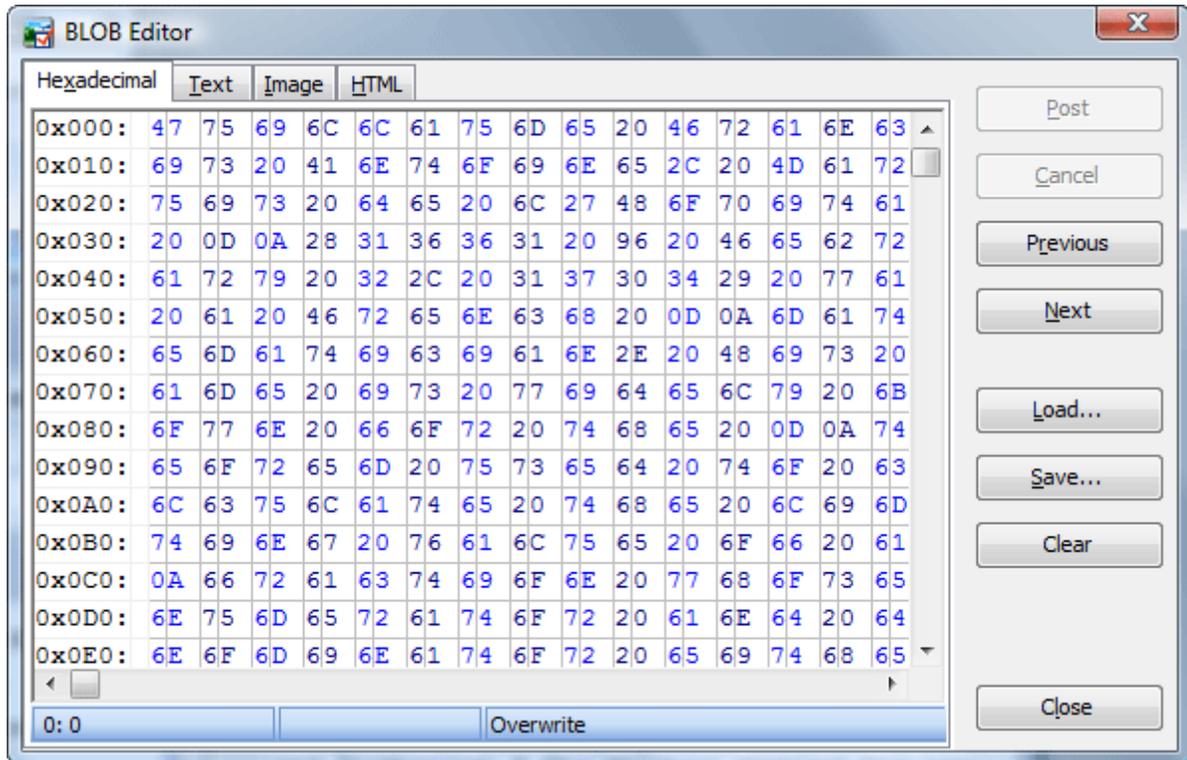
### 7.2.1 Editing as image

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.



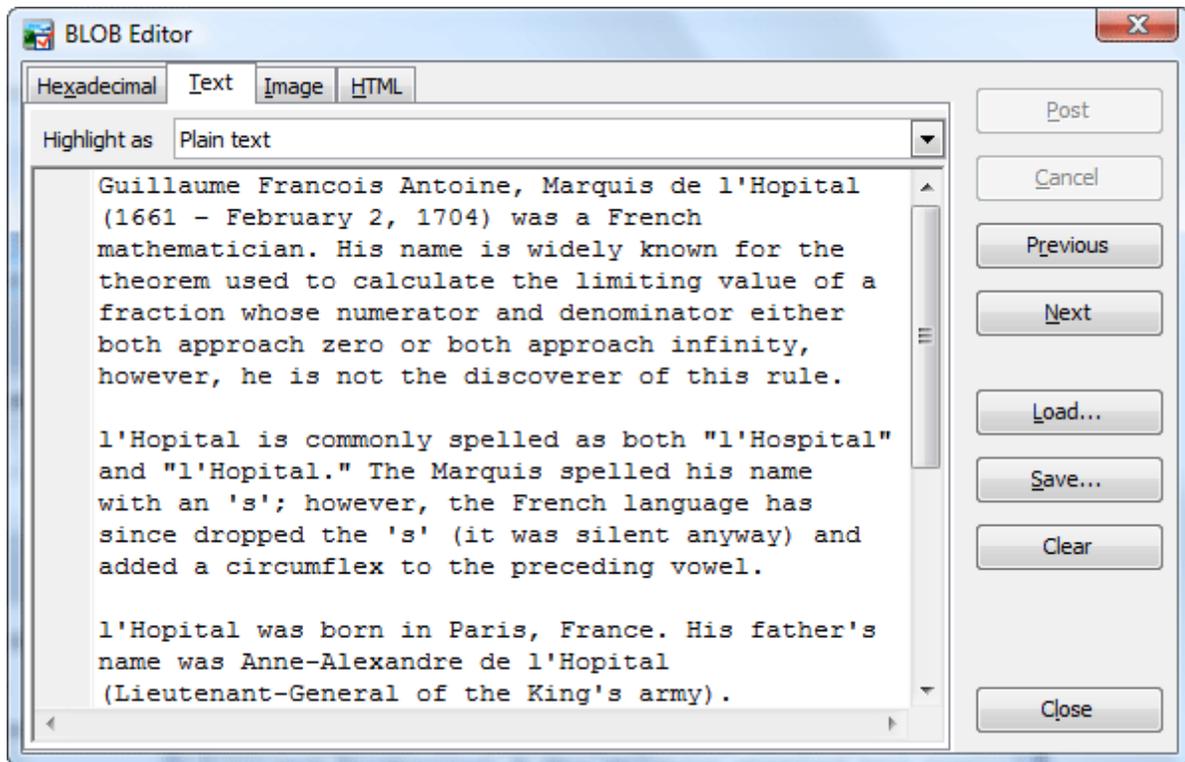
## 7.2.2 Editing as hexadecimal dump

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.



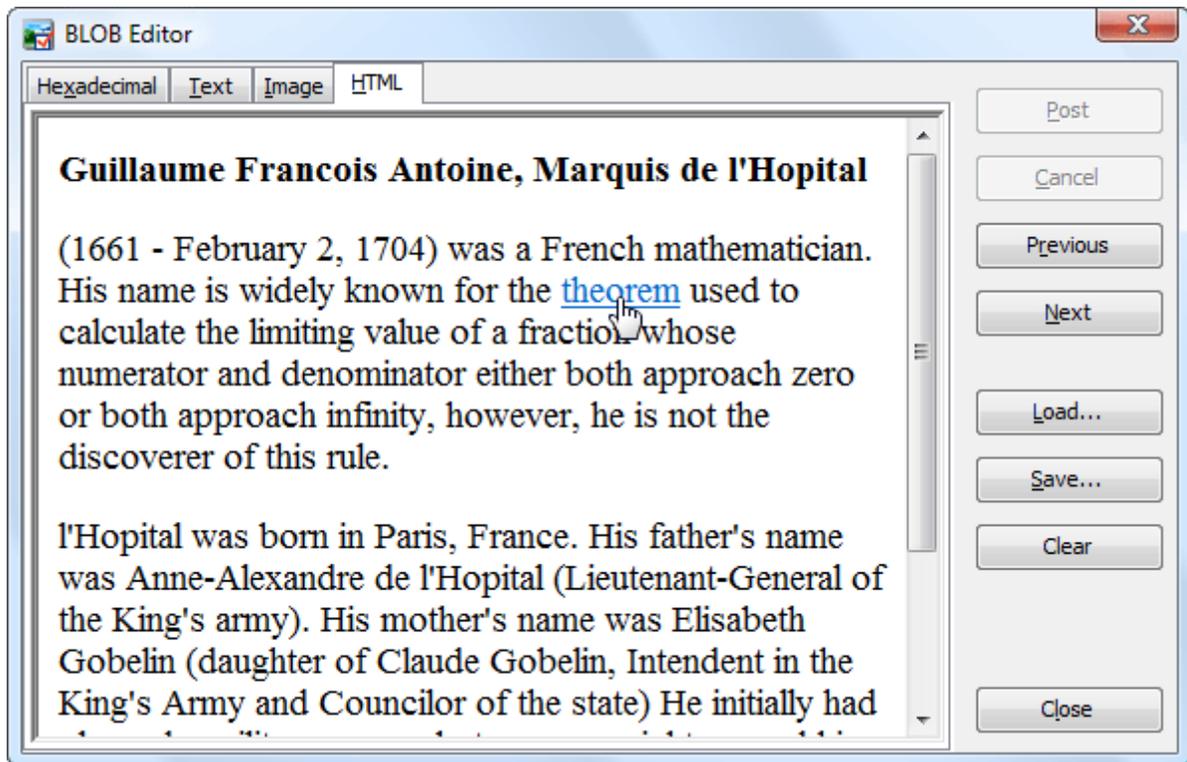
## 7.2.3 Editing as plain text

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.



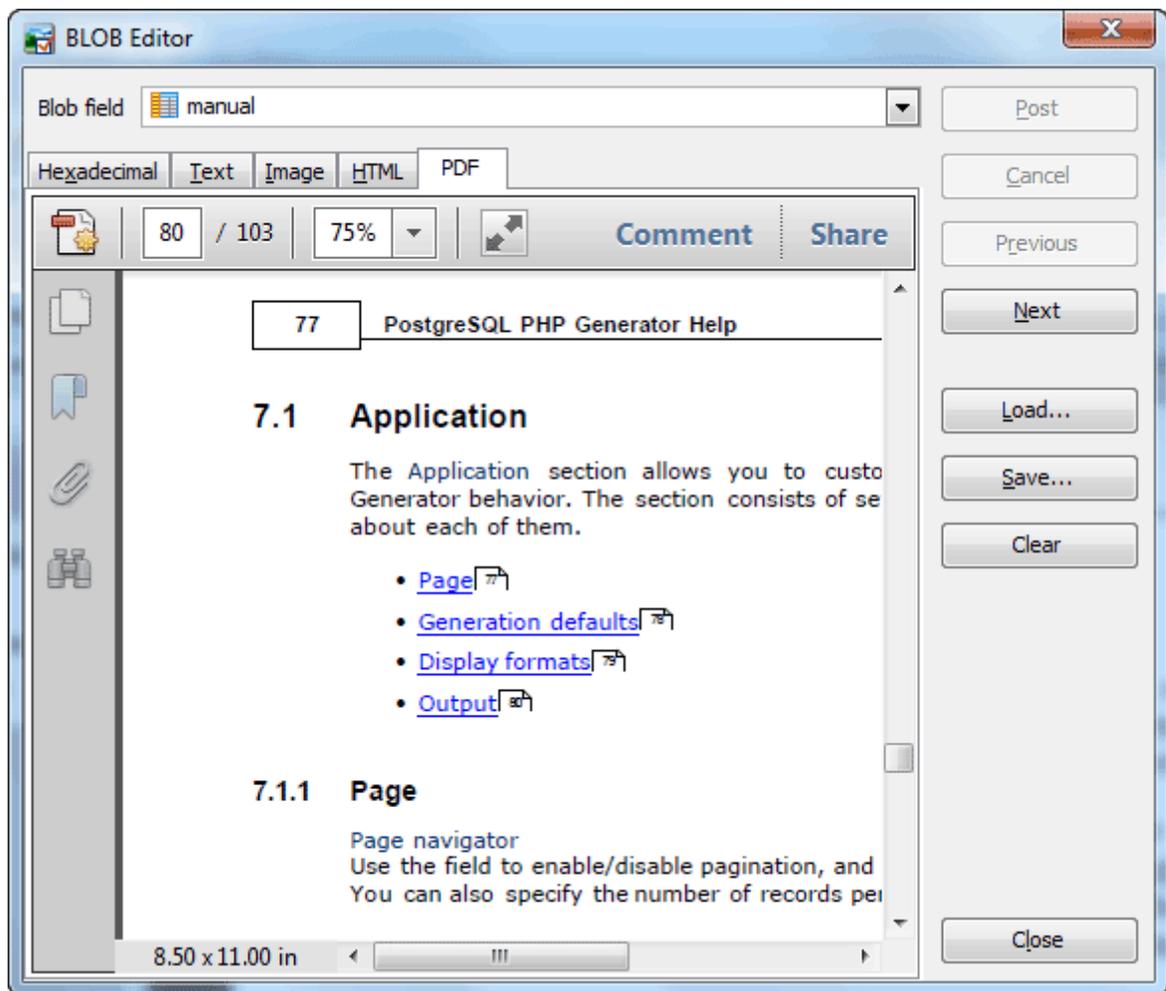
#### 7.2.4 Editing as HTML

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.



### 7.2.5 Editing as PDF document

The PDF panel presents field data as PDF document. To accomplish common operations with data, use the Adobe Reader toolbar.



## 7.3 Export Data Wizard

Data Export wizard is a tool to save data from Oracle 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>[295]</sup> of the destination file;
- Specify such additional options of the result file as [header and footer](#) <sup>[296]</sup>, [formats applied to exported data](#) <sup>[297]</sup> and [some format-specific options](#) <sup>[298]</sup>;
- [Select columns](#) <sup>[297]</sup> you want to include into result files;
- [Specify other export options](#) <sup>[301]</sup>.

**See also:** Get SQLDump, [Import Data Wizard](#) <sup>[305]</sup>

### 7.3.1 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.

The file name may contain current timestamp with the `%ts:TIMESTAMP_FORMAT%` string. Examples of valid log file names:

```
dbname_export_%ts:yyyy_mm_dd%.log  
export_%ts:yyyy_mm_dd_hh_mm%.log  
%ts:yyyy_mm_dd_hh_mm_ss%.log
```

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

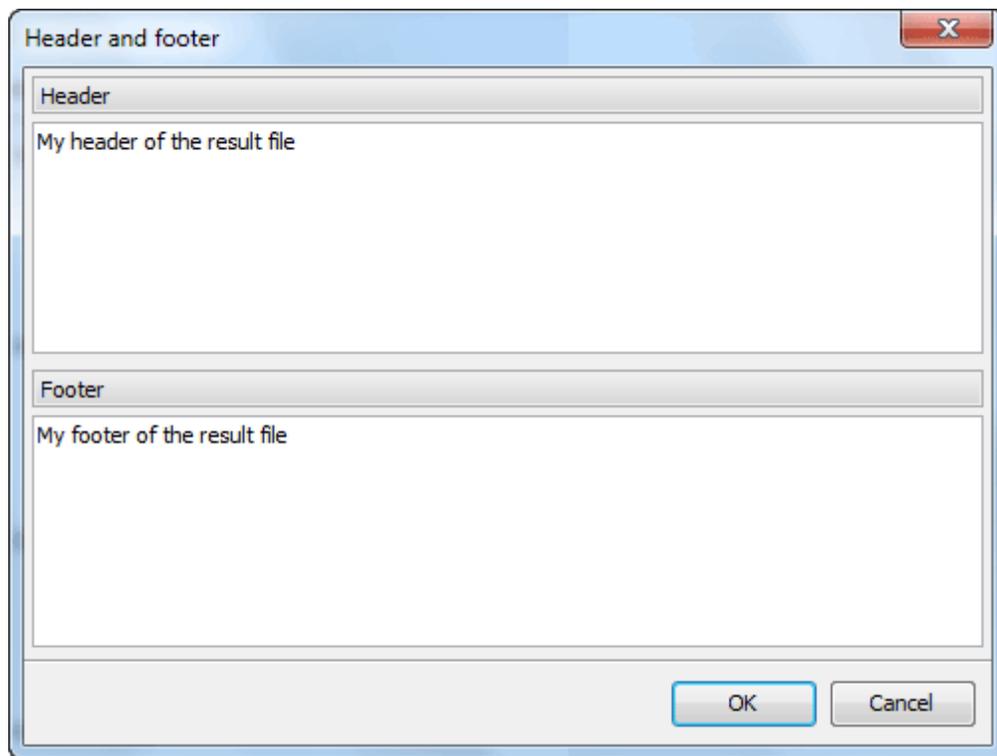
Select or enter the result file name and specify the encoding if necessary. To add current timestamp to the result file name, use the %ts:TIMESTAMP\_FORMAT% string (for example, %ts:yyyy\_mm\_dd%). Hint: you can set default directory for data export in the Edit Database Profile dialog.

File name: C:\Data\Excel\Customers.xls

Encoding: ANSI

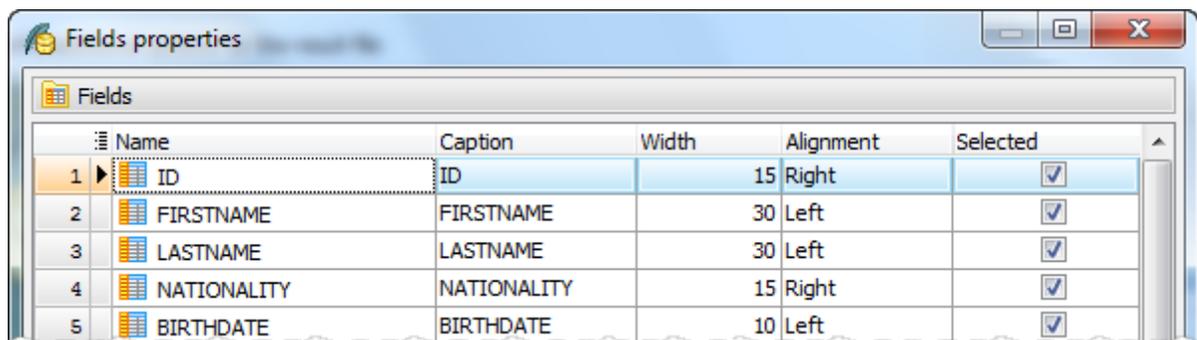
### 7.3.2 Setting header and footer

To specify the result file's header and footer, double click the corresponding button and complete fields of the [Header and Footer](#) window.



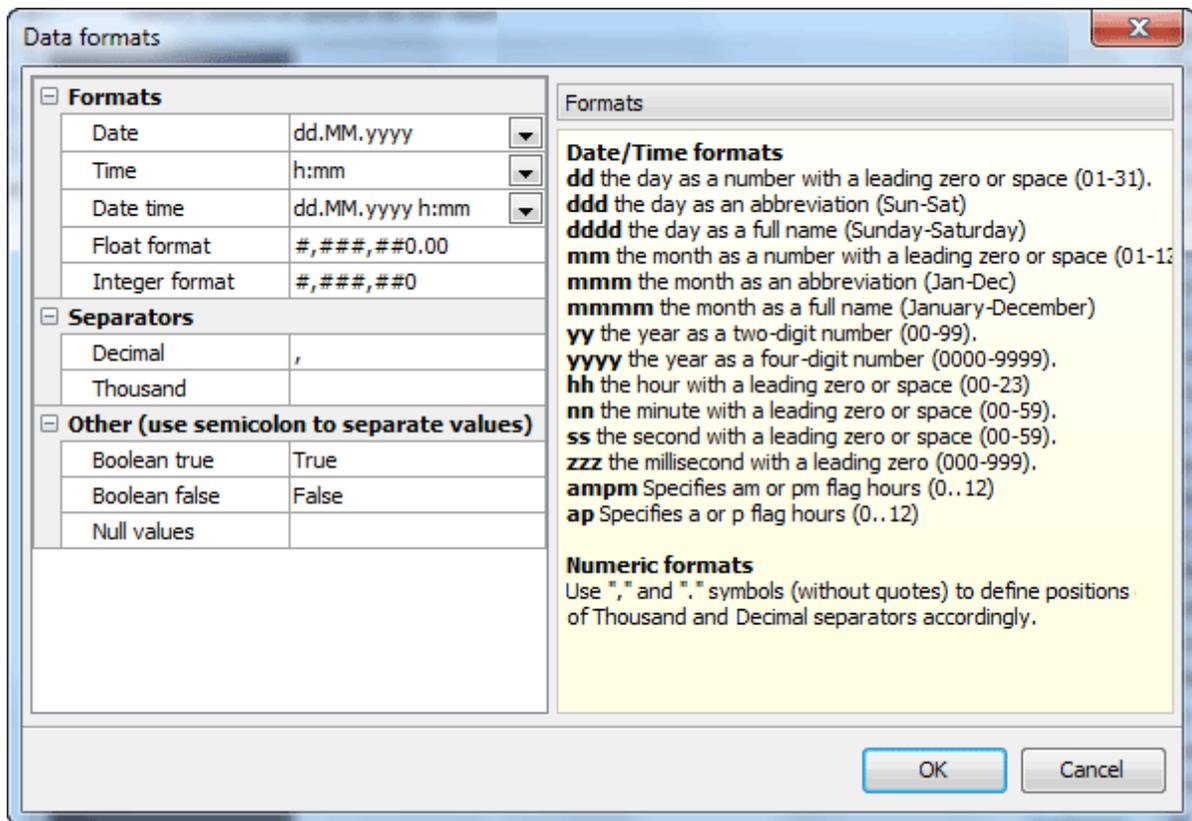
### 7.3.3 Selecting fields for export

Uncheck the Selected box to exclude the corresponding field from the export, specify a Caption to be used for the result column, and also width, and alignment for output columns (when applicable).



### 7.3.4 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.



### 7.3.5 Setting format-specific options

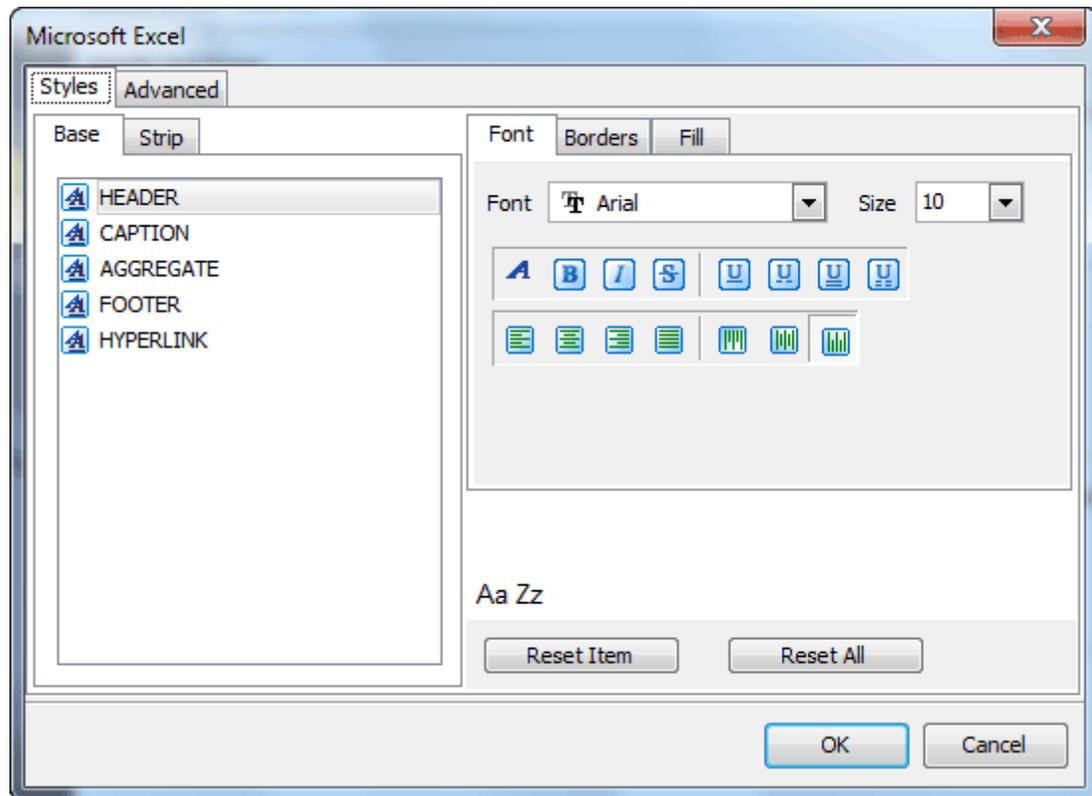
Each format supposes corresponding additional export options. Use the wizard option to adjust export properties depending on the target file format you have selected earlier. The following formats are at your disposal: [Microsoft Excel](#)<sup>[298]</sup>, Microsoft Excel 2007, [CSV](#)<sup>[300]</sup>, [Text](#)<sup>[300]</sup>, [HTML](#)<sup>[299]</sup>, [XML](#)<sup>[300]</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.



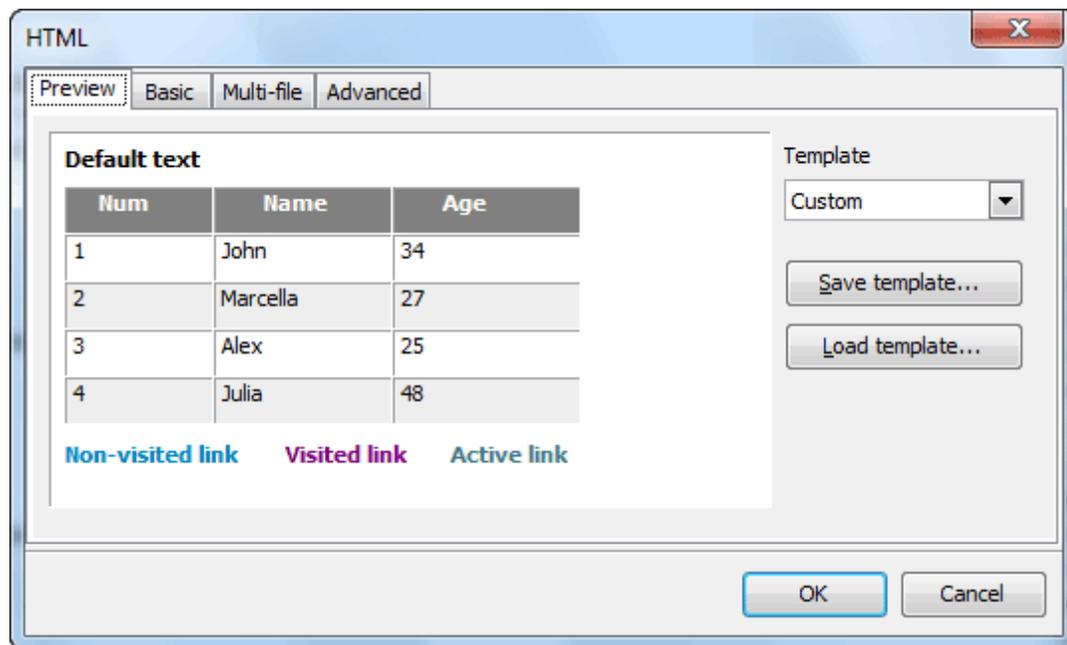
## HTML

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.

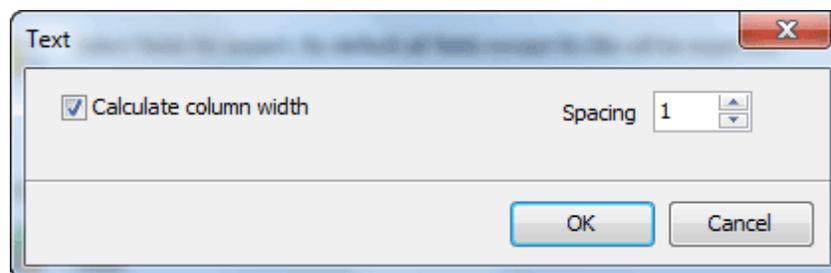
The **Multi-file** tab provides you with a possibility to split target HTML file into several separated files. This tab allows you to specify the record count for a single file, set an option to generate an index HTML file, and add an ability of navigation between each other to each of exported files.

The **Advanced** tab contains such HTML options as default font, background, cell padding and spacing, etc.



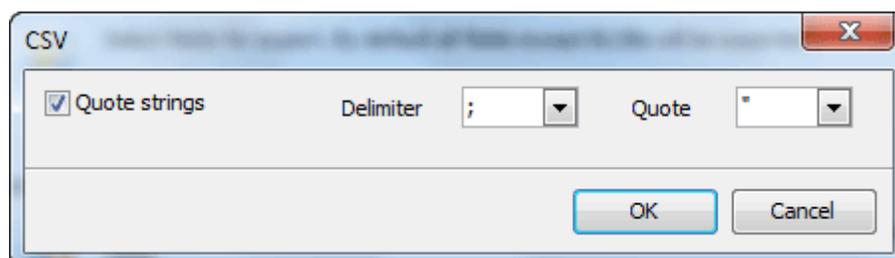
#### Text files

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.



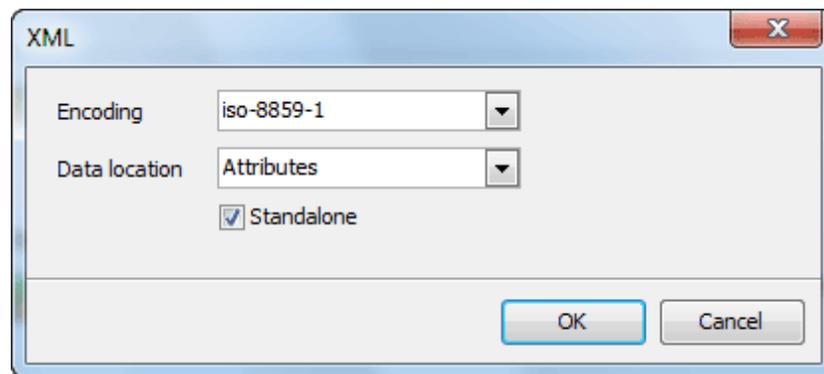
#### CSV files

You can specify column separator and optional values quote character for the target file on this step.



#### XML documents

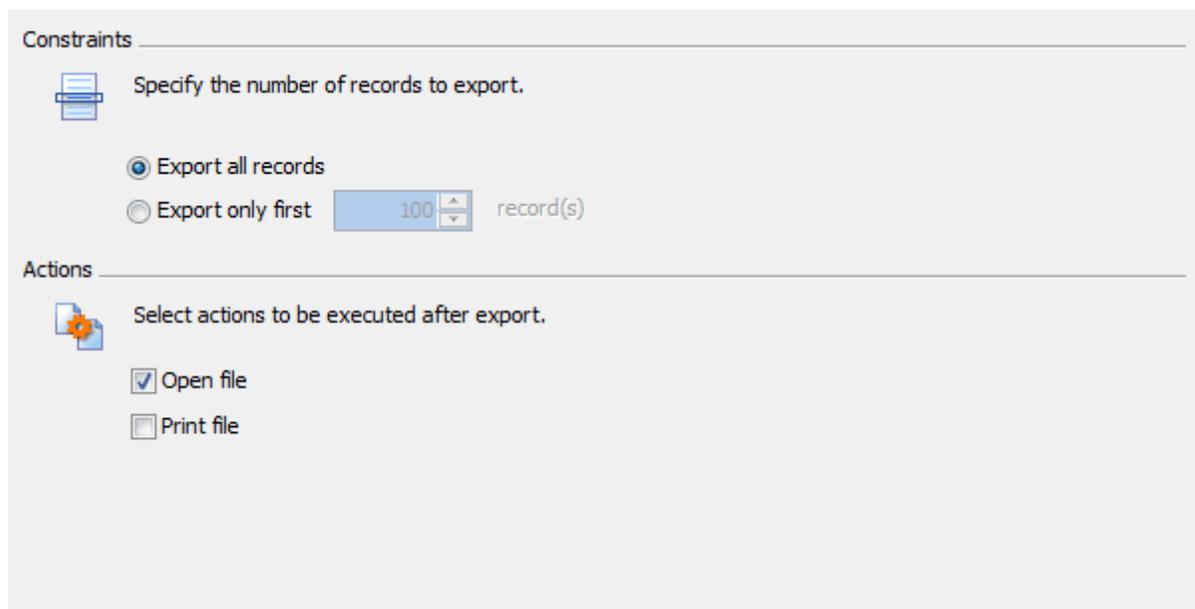
Specify XML document encoding in the **Encoding** edit box and set the Standalone option on if you wish the target document to be standalone.



### 7.3.6 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.



## 7.4 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>[302]</sup>
- [Specifying dump options](#) <sup>[303]</sup>

**See also:** [Export Data Wizard](#) <sup>[295]</sup>, [SQL Script Editor](#) <sup>[316]</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 (MySQL) [v]  
Native (MySQL)  
PostgreSQL  
SQL Server  
Oracle  
Firebird  
SQLite

Output

Send to script editor  
 Save to file

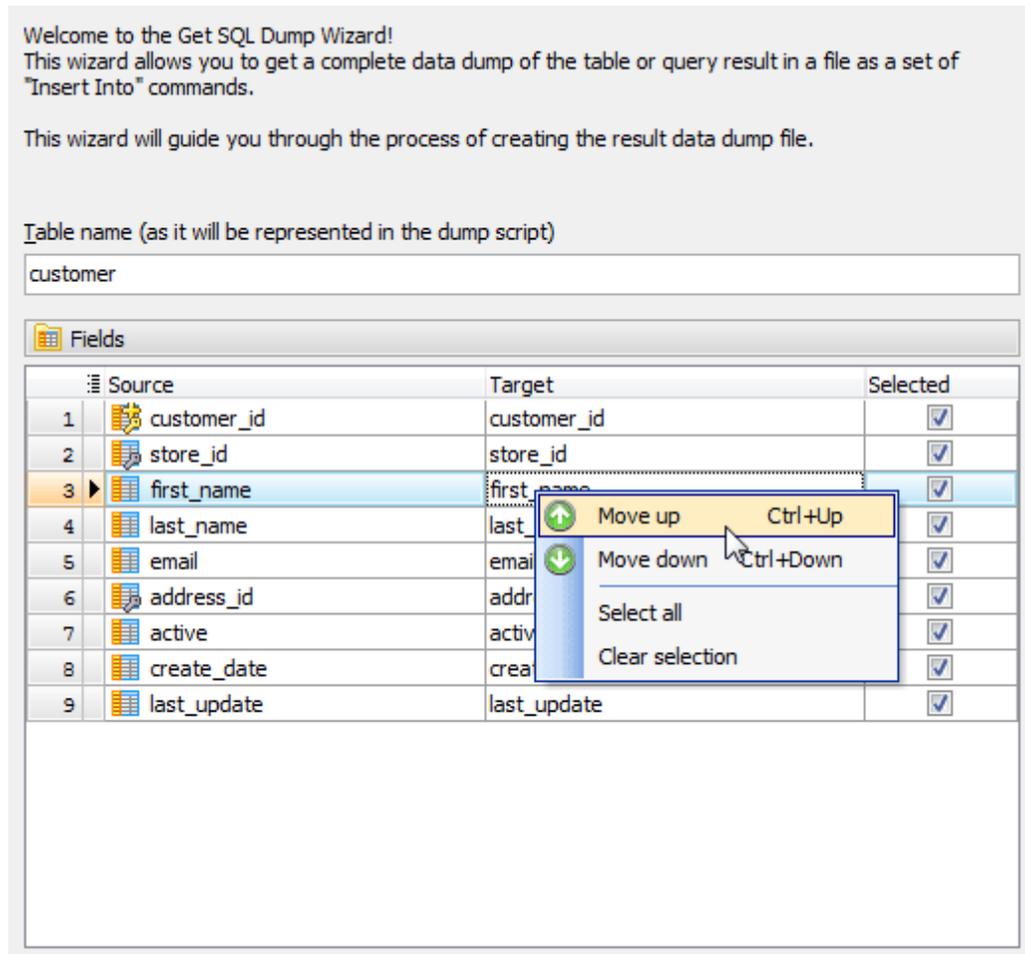
File name: C:\Users\marina\Documents\customer.sql  
Encoding: ANSI

Click "Ready" to dump your data.

### 7.4.1 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. *Text*, *GUID*, *Date*, *Time*, and *DateTime* columns are included in the result *INSERT* statements according to the **Storage Options** of the [Database Profile](#)<sup>[27]</sup>.



## 7.4.2 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>[316]</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

Native (MySQL)  
PostgreSQL  
**SQL Server**  
Oracle  
Firebird  
SQLite

Output   Send to script editor  
 Save to file

File name  Encoding

Click "Ready" to dump your data.

## 7.5 Import Data Wizard

[Import Data Wizard](#) provides you with a graphical user interface to import data from the most popular files formats into existing Oracle 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>[306]</sup> of the input data and the source file name;
- [Map source file columns and target table fields](#) <sup>[306]</sup>;
- [Specify other import options](#) <sup>[311]</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.

File name	Password	Encoding
D:\Data\Excel\employee.xls		ANSI
Connection string	Identifier quote characters	
	None (table_name)	
Data source	Data location	Delimiter
Employee_list	Attributes	
		Quote

See also: [Export Data Wizard](#)

## 7.5.1 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:

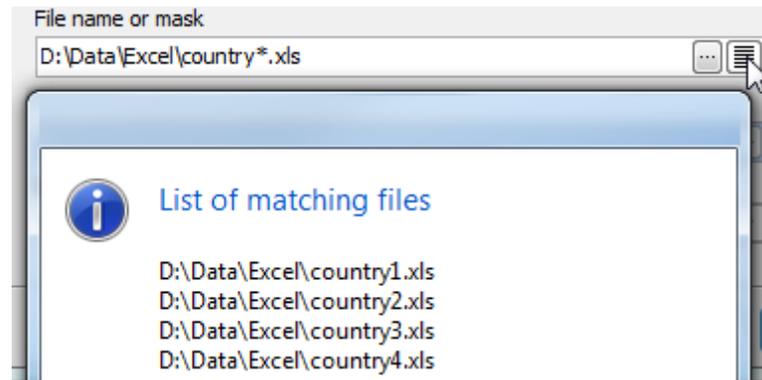
*D:\Data\Excel\country1.xls*

*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>
```

## 7.5.2 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>[309]</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>[310]</sup> used for the input file.

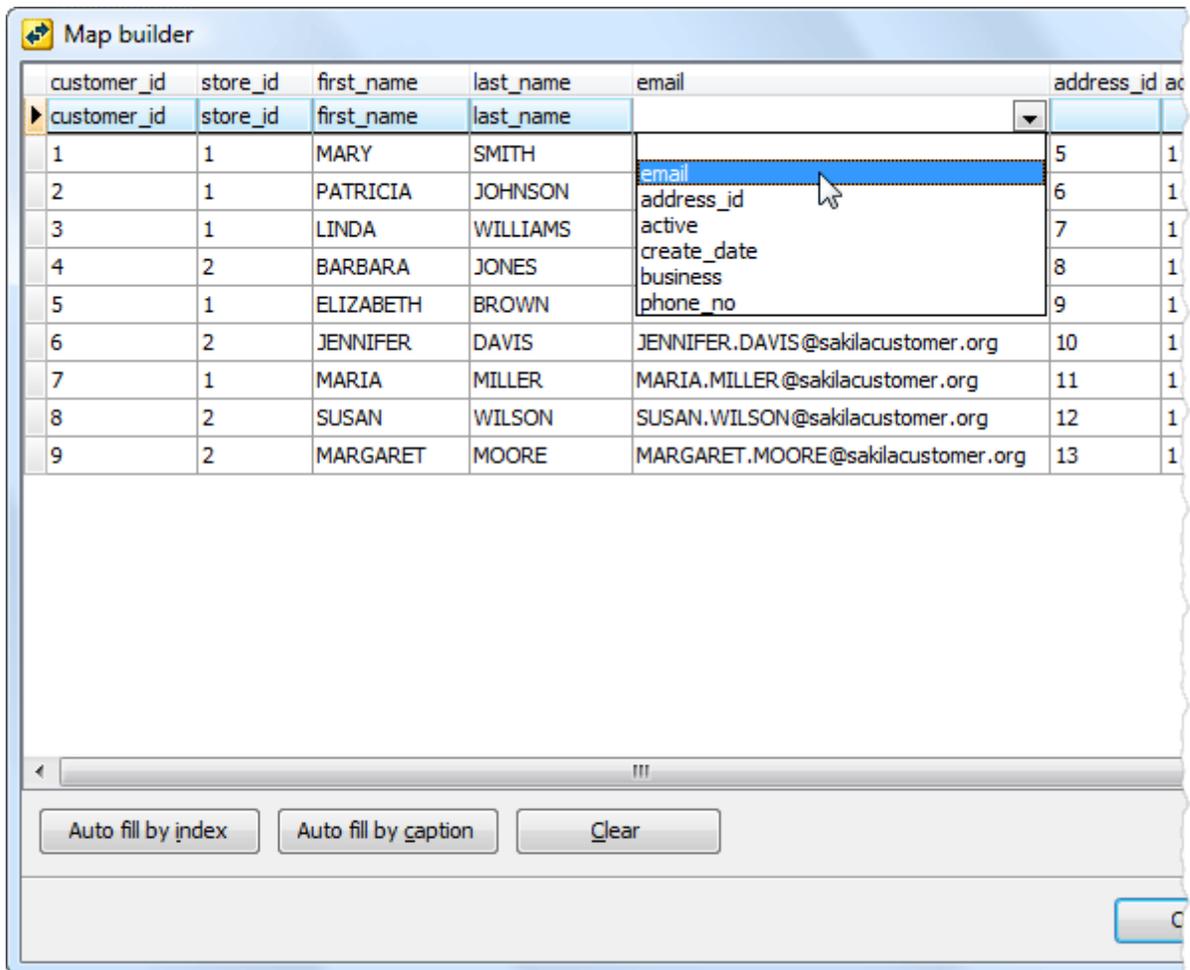
To exclude the first file row, use the [File contains column header](#) checkbox.

Columns				
	Target field	Source column	Replacements	Empty values interpretation
1	film_id	A		
2	title	B		As Null
3	description	C		As Null
4	release_year	D		
5	language_id	E		
6	original_language_id	F		
7	rental_duration	G		
8	rental_rate	H		
9	length	I		
10	replacement_cost	J		
11	rating	K		As Null
12	last_update	L		
13	special_features	M		As Null
14	fulltext			As Null

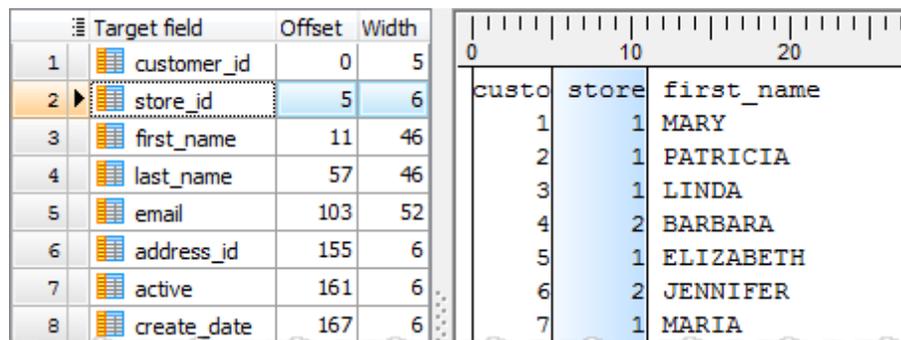
File contains column headers

### 7.5.2.1 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.



### 7.5.2.2 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	mmmm 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.

<input type="checkbox"/> <b>Formats</b>		<b>Date time formats</b> <b>dd</b> the day as a number with a leading zero or space (01-31). <b>ddd</b> the day as an abbreviation (Sun-Sat) <b>dddd</b> the day as a full name (Sunday-Saturday) <b>mm</b> the month as a number with a leading zero or space (01-12). <b>mmm</b> the month as an abbreviation (Jan-Dec) <b>mmmm</b> the month as a full name (January-December) <b>yy</b> the year as a two-digit number (00-99). <b>yyyy</b> the year as a four-digit number (0000-9999). <b>hh</b> the hour with a leading zero or space (00-23) <b>nn</b> the minute with a leading zero or space (00-59). <b>ss</b> the second with a leading zero or space (00-59). <b>zzz</b> the millisecond with a leading zero (000-999). <b>ampm</b> Specifies am or pm flag hours (0..12) <b>ap</b> Specifies a or p flag hours (0..12)
Date		
Time		
Date time		
<input type="checkbox"/> <b>Separators</b>		
Decimal	,	
Thousand	#160	
<input type="checkbox"/> <b>Other (use semicolon to separate values)</b>		
Boolean true	True	
Boolean false	False	
Null values	;NULL	

### 7.5.3 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 neseccary 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.

### Import mode

If the [Update existing records](#) option is turned ON, the records will be either updated or inserted: an UPDATE will be performed when a target row exists in the table and an INSERT is performed when the target row does not exist.

Import Data wizard supports [SQL\\*Loader](#) to insert data to the table. This feature can speed up the import process up to 10 times so it is recommended to use it always if possible. Uncheck this option to use INSERT statements instead.

## 8 Database Tools

**Oracle Maestro** provides a number of powerful tools for working with databases.

The following tools are available:

- [SQL Editor](#) <sup>[264]</sup>  
Creates and executes SQL queries.
- [Visual Query Builder](#) <sup>[269]</sup>  
Builds queries visually.
- [Script Runner](#) <sup>[315]</sup>  
Executes SQL scripts to the database.
- [SQL Script Editor](#) <sup>[316]</sup>  
Allows to edit and execute SQL scripts.
- [Generate Database Report Wizard](#) <sup>[318]</sup>  
Generates the database HTML or PDF report for structure of selected object in a whole or partially.
- [BLOB Viewer](#) <sup>[321]</sup>  
Displays a content of BLOB fields in different representations.
- [Diagram Viewer](#) <sup>[327]</sup>  
Represents data from a table or a query as a diagram in various ways.
- [Data Analysis](#) <sup>[331]</sup>  
Allows to slice and dice information efficiently according your business rules.
- [Report Designer](#) <sup>[336]</sup>  
Prepares data for reading, viewing, and printing in a polished look.
- [Schema Designer](#) <sup>[343]</sup>  
Allows to represent database tables and relationships as ER diagrams.
- [PL/SQL Debugger](#) <sup>[348]</sup>  
An excellent tool to debug PL/SQL code such as procedures and functions (both stand-alone and packaged) using traditional debugging features.
- [Session Browser](#) <sup>[353]</sup>  
A very useful feature for DBAs to monitor the users' activity.
- [Database Export](#) <sup>[320]</sup>  
Allows you to choose export mode (database, schemas, tablespaces or tables), select objects to export and specify a lot of additional options to get the database dump according to your needs.
- [SQL Generator](#) <sup>[355]</sup>  
Provides you with a set of simple SQL statements.

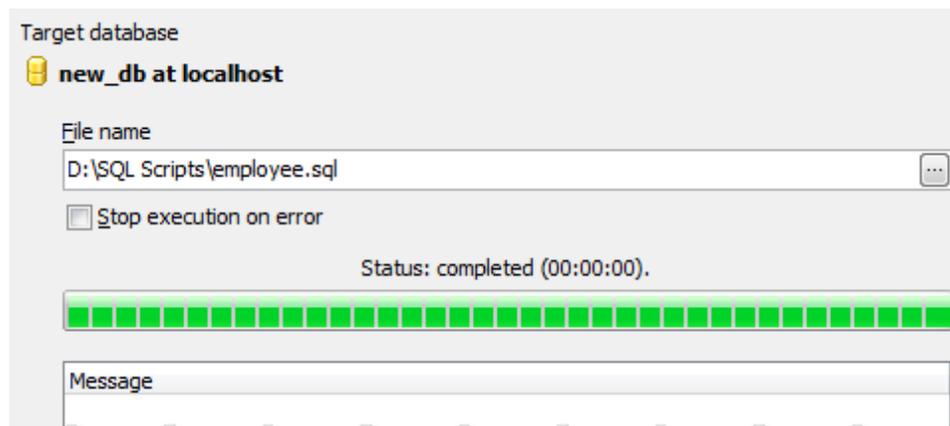
- 
- Simple tools for [DML procedures](#) and [Updatable views](#) generation allow to create a bunch of CRUD procedures automatically.

## 8.1 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>[316]</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.



## 8.2 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>[264]</sup>. If you have a script that is ready to use, execute it with [Script Runner](#)<sup>[315]</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>[315]</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>[375]</sup> tab.

The screenshot displays the Oracle Maestro SQL Script Editor interface. The left sidebar shows the database 'sakila at localhost' and a tree view of the database schema including tables like ACTOR, ADDRESS, CITY, COUNTRY, and customer. The main editor area contains the following SQL script:

```

CREATE TABLE ACTOR (
  ACTOR_ID      INTEGER      PRIMARY KEY,
  FIRST_NAME    VARCHAR(45)  NOT NULL,
  LAST_NAME     VARCHAR(45)  NOT NULL,
);

CREATE TABLE COUNTRY (
  COUNTRY_ID    INTEGER      NOT NULL,
  COUNTRY      VARCHAR(50)  NOT NULL,
  /* Keys */
  PRIMARY KEY (COUNTRY_ID)
);

CREATE TABLE CITY (
  CITY_ID        INTEGER      PRIMARY KEY,
  CITY           VARCHAR(50)  NOT NULL,
  COUNTRY_ID     INTEGER      NOT NULL,
  /* Foreign keys */
  CONSTRAINT FK_CITY_COUNTRY
    FOREIGN KEY (COUNTRY_ID)
      REFERENCES COUNTRY(COUNTRY_ID)
);

CITY_ID        INTEGER      NOT NULL,
POSTAL_CODE    VARCHAR(10),
PHONE          VARCHAR(20) NOT NULL,
/* Keys */
PRIMARY KEY (ADDRESS_ID),
/* Foreign keys */
CONSTRAINT FK_ADDRESS_CITY
  FOREIGN KEY (CITY_ID)
    REFERENCES CITY(CITY_ID)
  ON DELETE RESTRICT
  ON UPDATE CASCADE
);

CREATE INDEX IDX_FK_CITY_ID
ON ADDRESS
(CITY_ID);

CREATE TABLE customer (

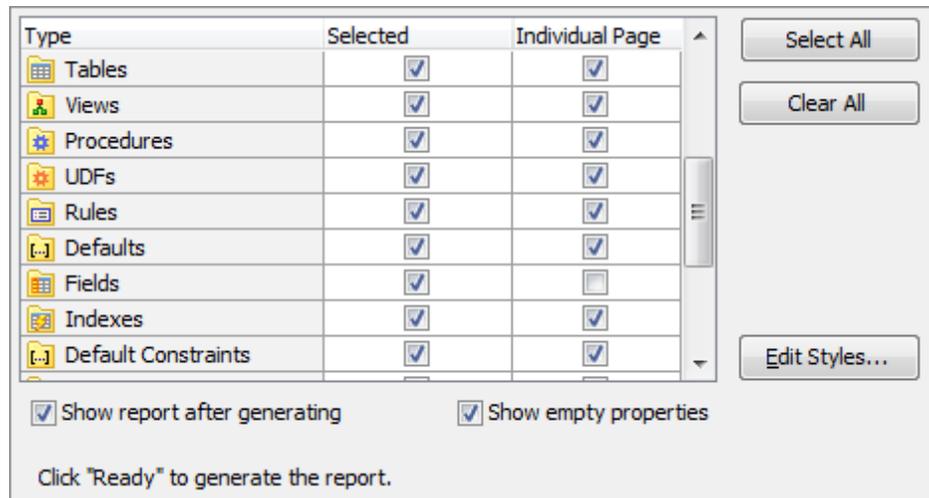
```

The definition for the CITY table is highlighted in yellow in the original image. The status bar at the bottom indicates '44: 1'.

## 8.3 Generate Database Report

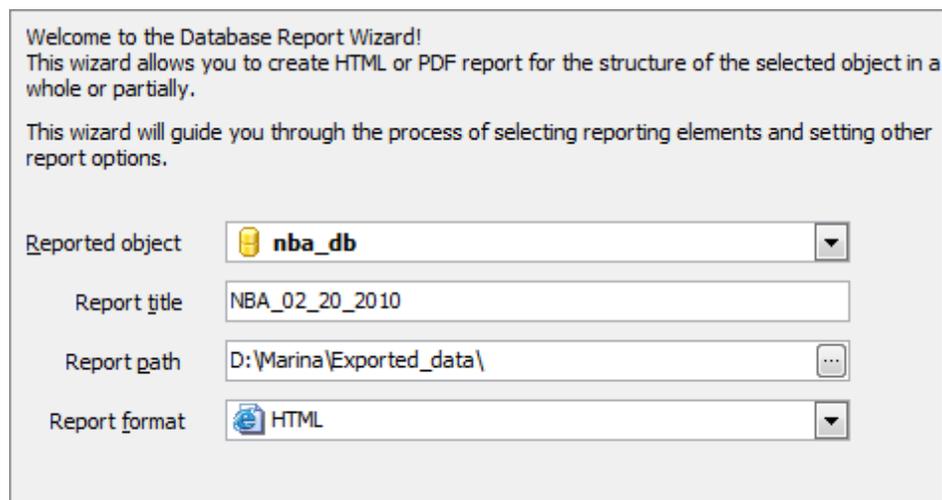
With the help of [Generate Database Report Wizard](#) you can create HTML or PDF report for the structure of the selected object in the whole or partially. To run this wizard select the [Tools | Generate Database Report](#) main menu item.

- [Selecting reporting elements and setting other report options](#)<sup>[318]</sup>
- [Specifying reporting objects and editing styles](#)<sup>[318]</sup>



### 8.3.1 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.



### 8.3.2 Reporting objects and editing styles options

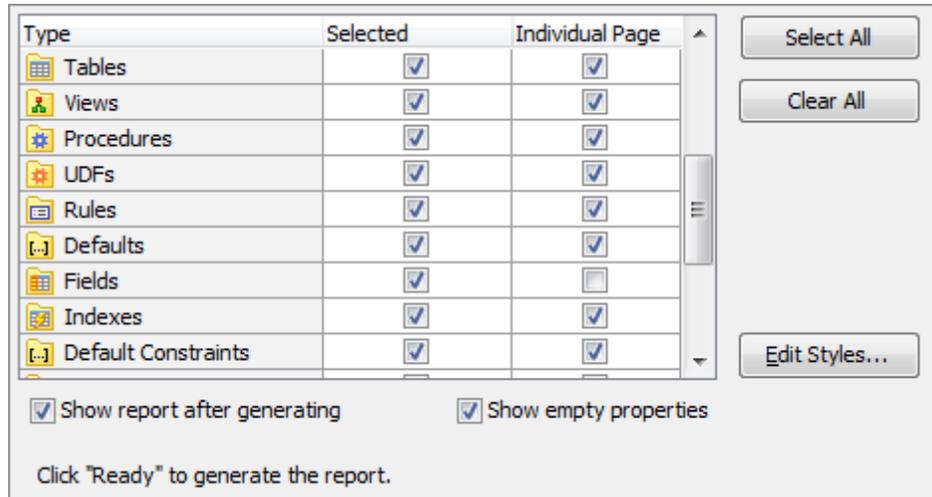
This step allows you to select the essential objects to report and to specify the output format and style using [Report Style Editor](#)<sup>[319]</sup>.

- Show report after generating

If checked, opens the result files in the associated program after making the report.

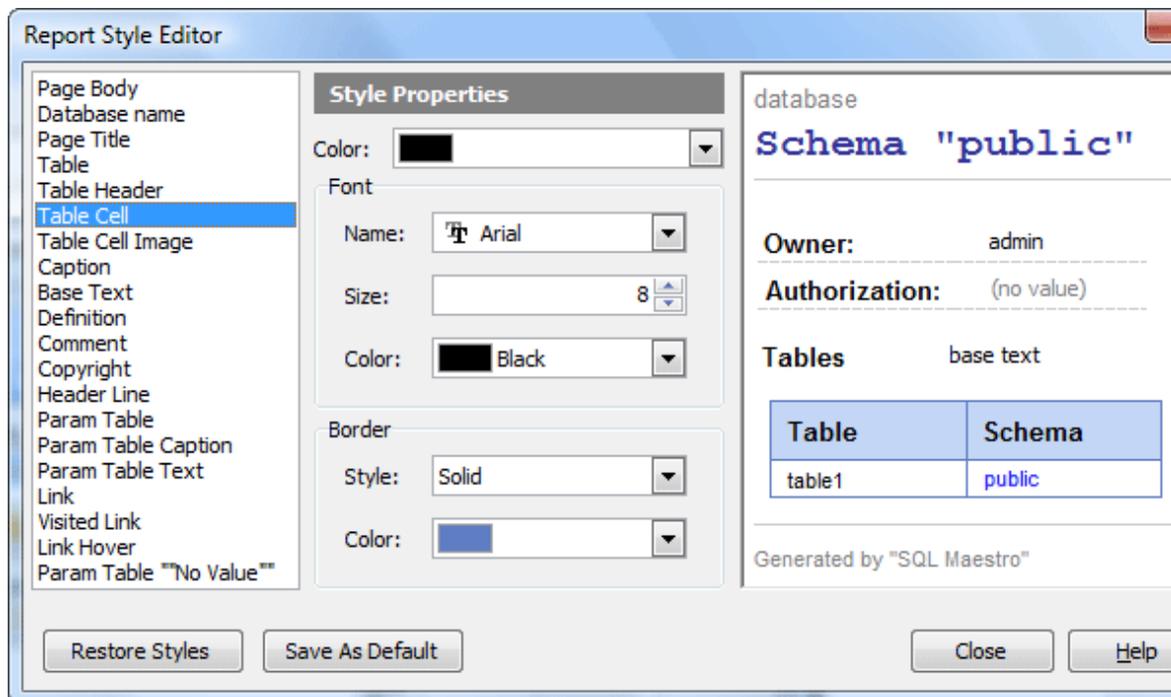
Show empty properties

If checked, allows you to report objects even if they are empty.



### 8.3.3 Editing database report style

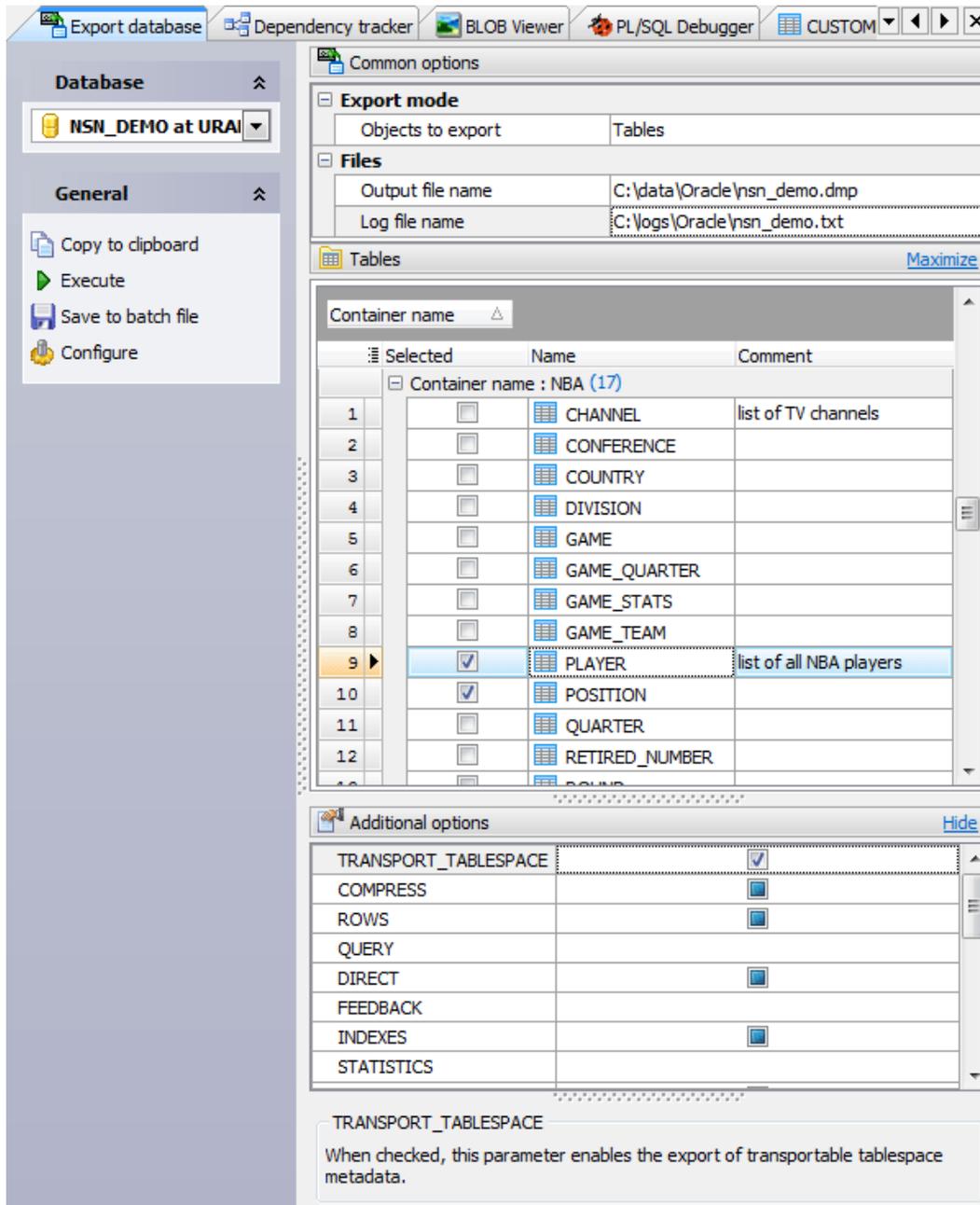
Using Report Style Editor you can specify style properties of a report including font size, color and name for different elements.



## 8.4 Database Export

Since last version Oracle Maestro provides you with [Database Export](#) tool, which represents a GUI wrapper for standard EXP utility. The tool allows you to choose export mode (database, schemas, tablespaces or tables), select objects to export and specify a lot of additional options to get the database dump according to your needs. It is also possible to save all the settings as a batch file for future executing. The [Database Export](#) window also can be invoked from the [Tools](#) menu.

**Note:** you should have installed the EXP utility to use this functionality.



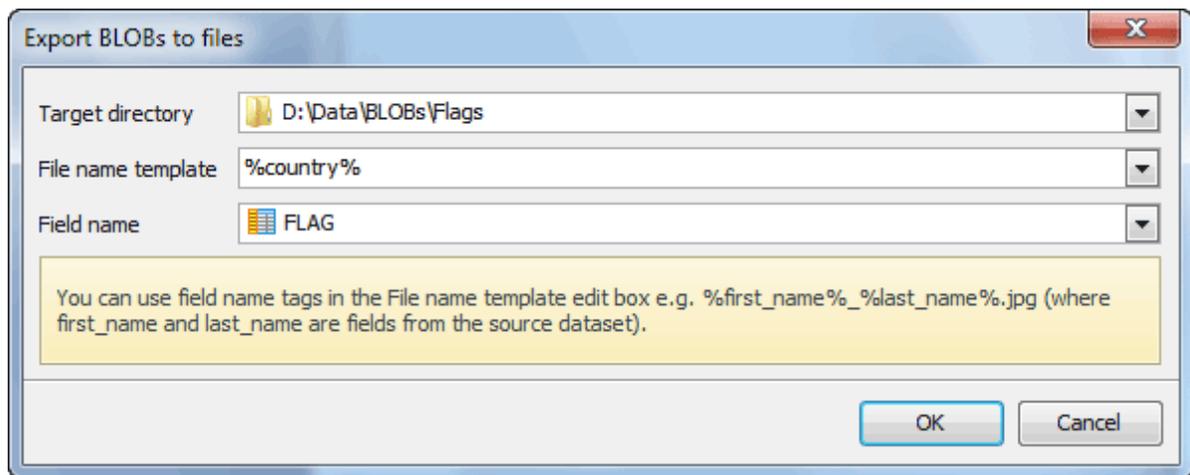
## 8.5 BLOB Viewer

BLOB Viewer allows you to view the content of the BLOB fields in various representations.

- [Viewing BLOB field as hexadecimal dump](#)<sup>[321]</sup>
- [Viewing BLOB field as plain text](#)<sup>[322]</sup>
- [Viewing BLOB field as graphical image](#)<sup>[323]</sup>
- [Viewing BLOB field as HTML](#)<sup>[324]</sup>
- [Viewing BLOB field as PDF](#)<sup>[325]</sup>

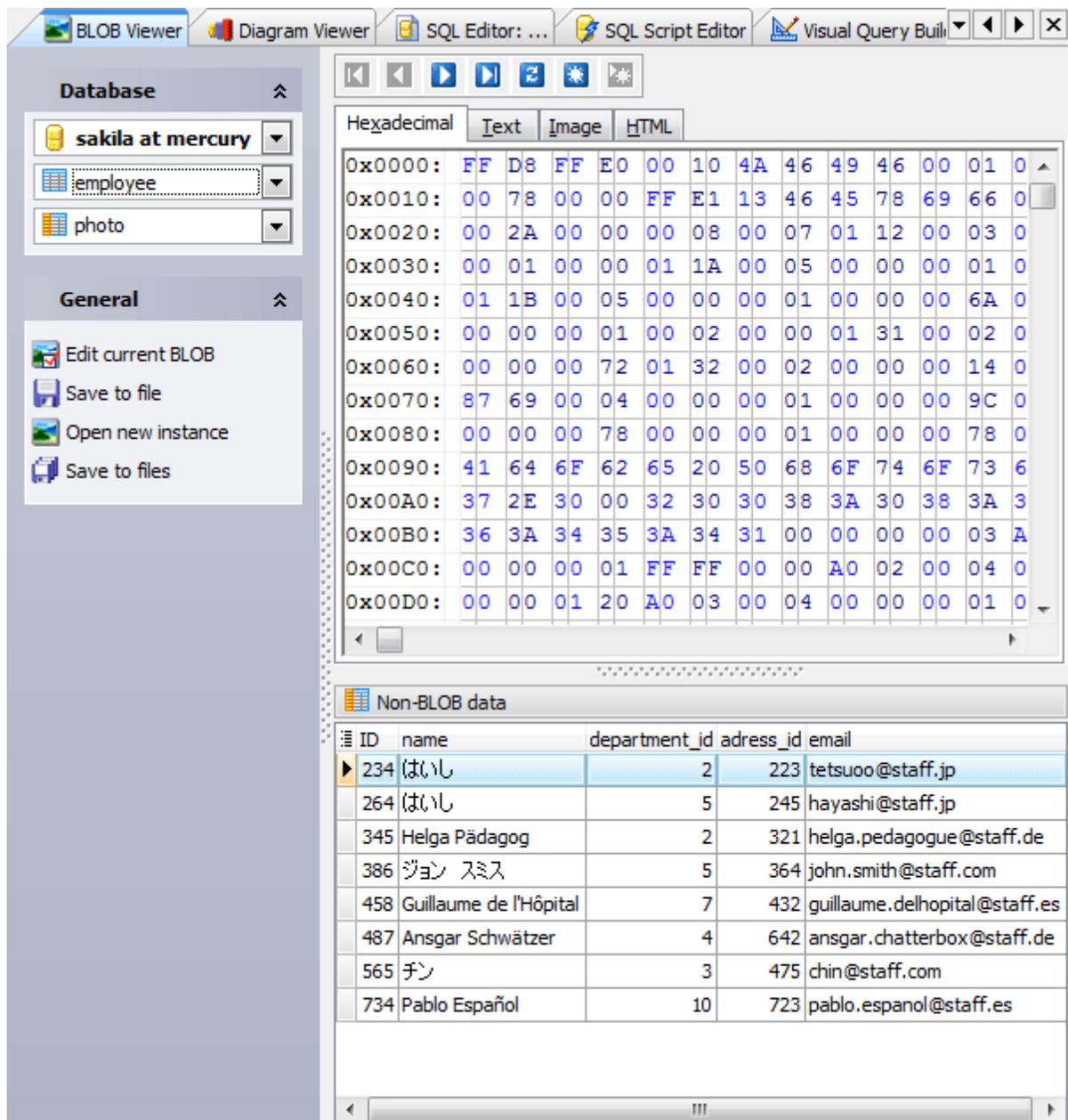
**See also:** [BLOB Editor](#)<sup>[290]</sup>

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.



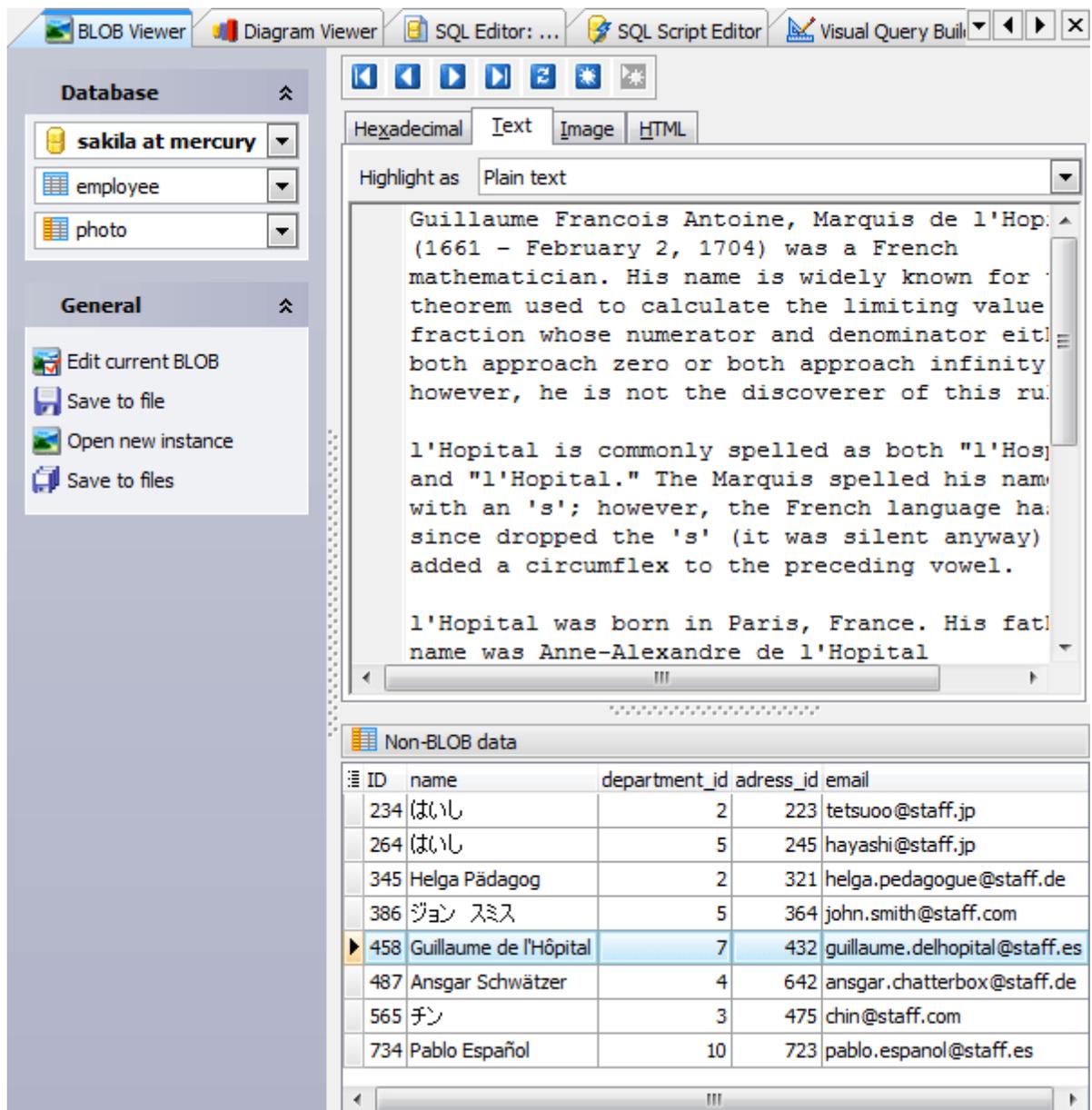
### 8.5.1 Viewing as hexadecimal dump

The **Hexadecimal** panel allows you to view data in hexadecimal mode.



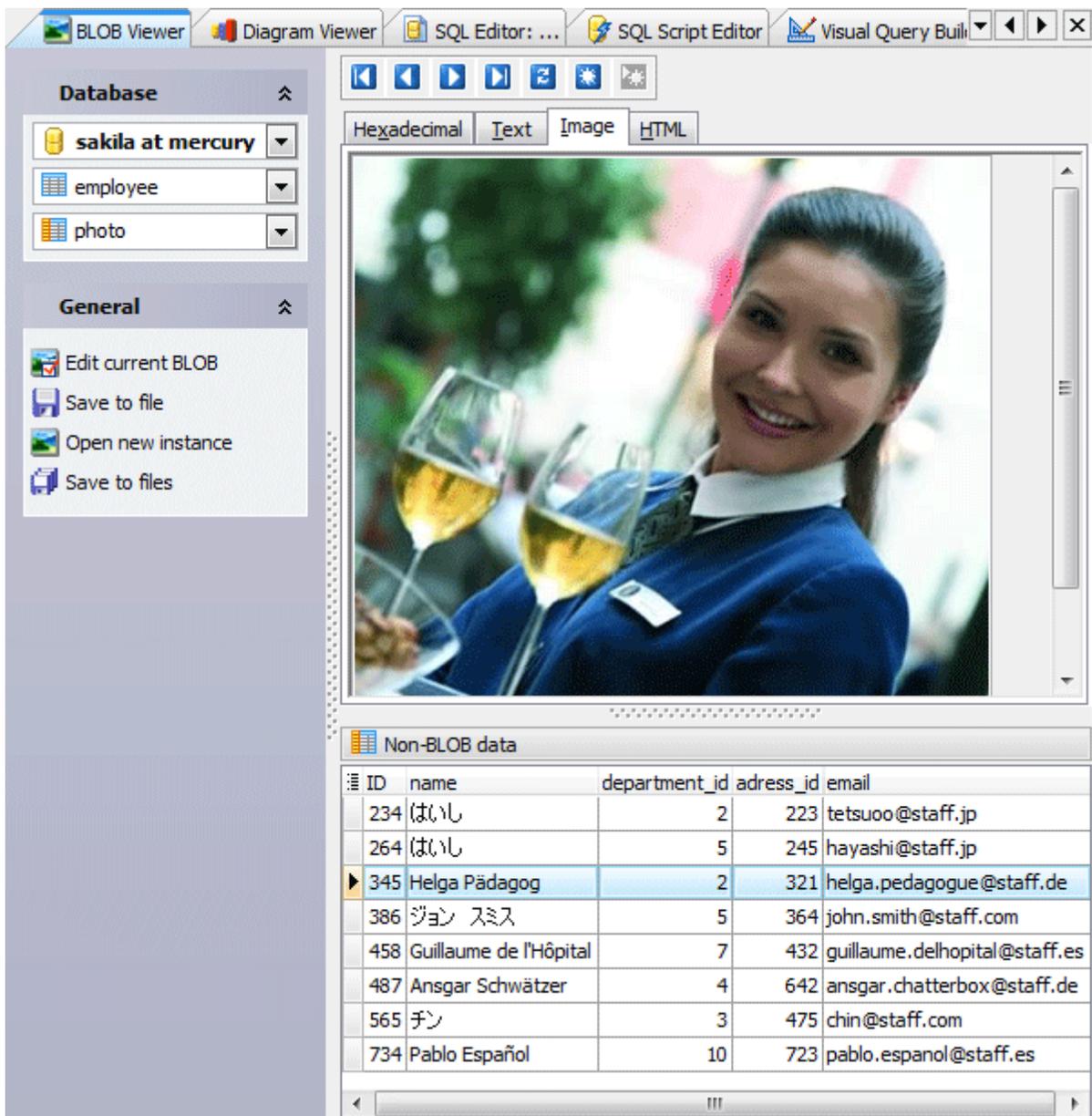
## 8.5.2 Viewing as plain text

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.



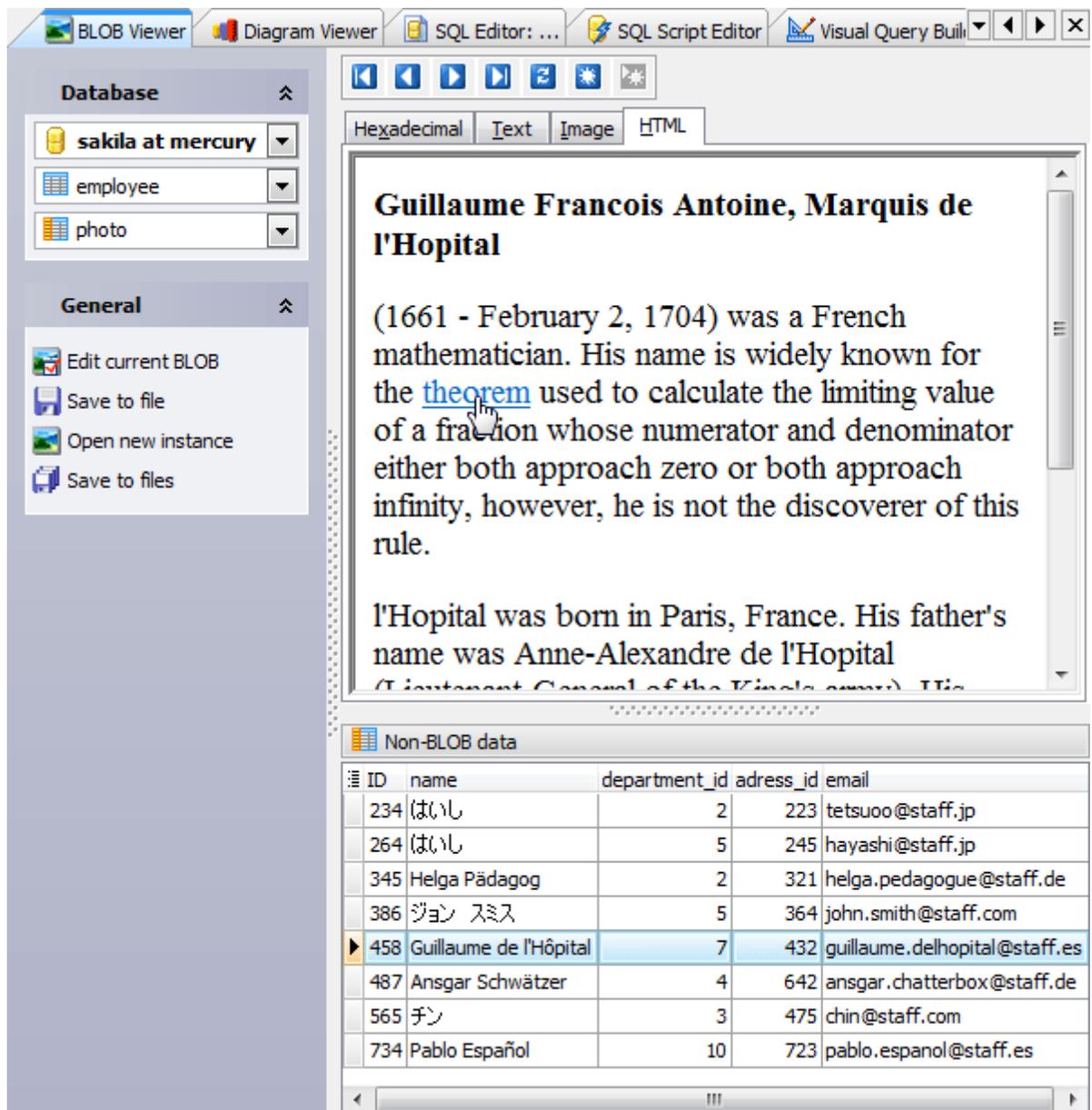
### 8.5.3 Viewing as image

The [Image](#) panel displays field data as image.



## 8.5.4 Viewing as HTML

The [HTML](#) panel displays field data as HTML.



### 8.5.5 Viewing as PDF

The PDF panel allows you to browse PDF data stored in the database.

The screenshot shows a database tool interface with the following components:

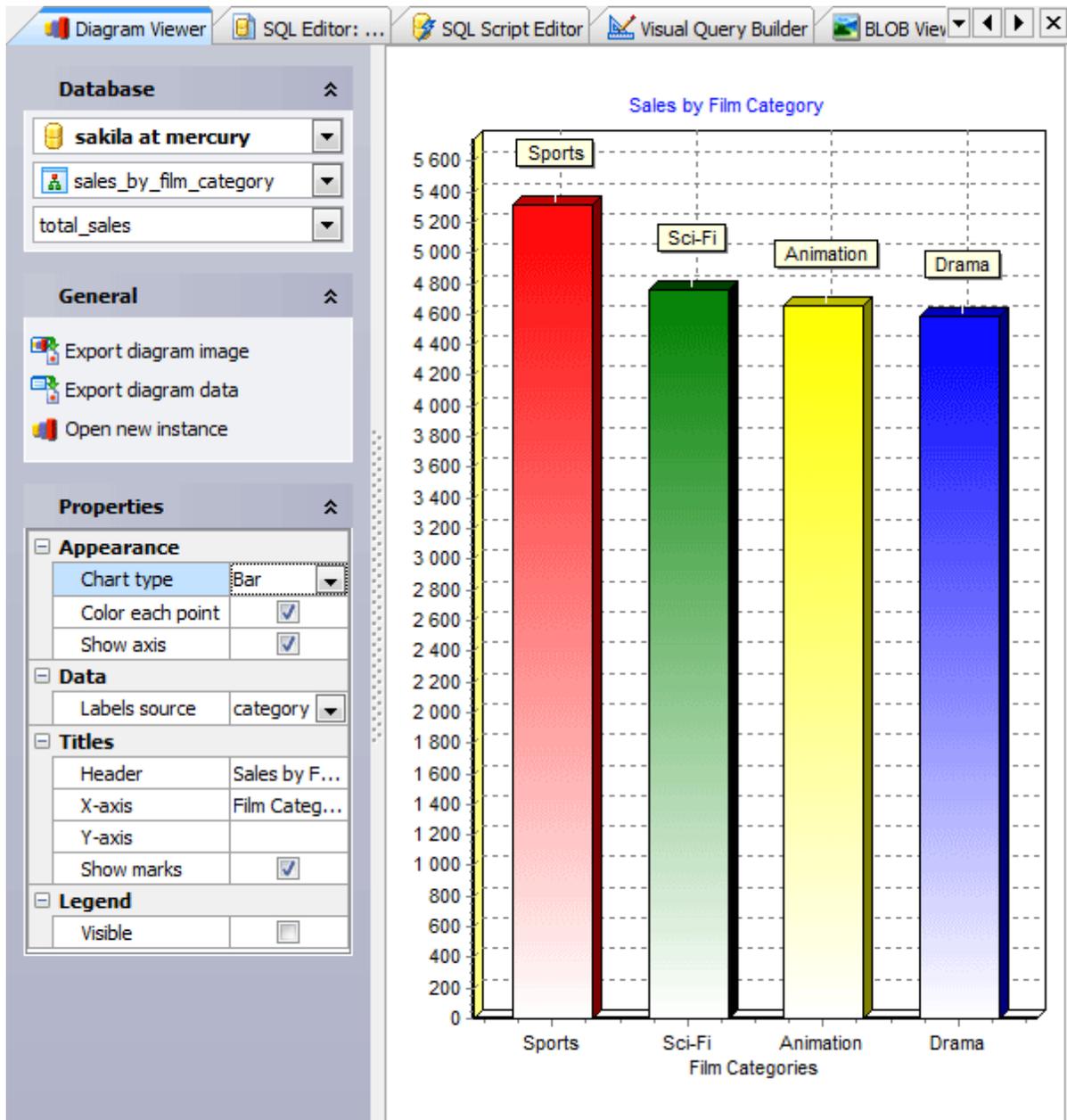
- Top Bar:** Contains tabs for BLOB Viewer, SQL Script Editor, Data Analysis, Visual Query Builder, and Designer.
- Database Sidebar:**
  - Database:** test\_utf8 at d
  - public.software**
  - manual**
  - General:**
    - Edit current BLOB
    - Save to file
    - Open new instance
    - Save to files
- Main Content Area:**
  - Navigation: 18 / 103, 75% zoom, Comment, Share.
  - Content: PostgreSQL PHP Generator Help, section 2 Getting started.
    - Connection properties:** Set the [connection parameters](#) for the c with.
    - Script connection properties:** Specify here connection parameters for Postgre example, if your webserver and PostgreSQL se Host as localhost.
    - Projects:** When working with a project, all the session and may be edited if necessary. To run a w Project on the first wizard step and enter projects are also available from this popup [Projects](#).
  - Inset window: Connection properties
    - I can connect to the server directly or via SSH tunneling
    - [Configure SSH options](#)
- Bottom Panel:** Non-BLOB data table.

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

## 8.6 Diagram Viewer

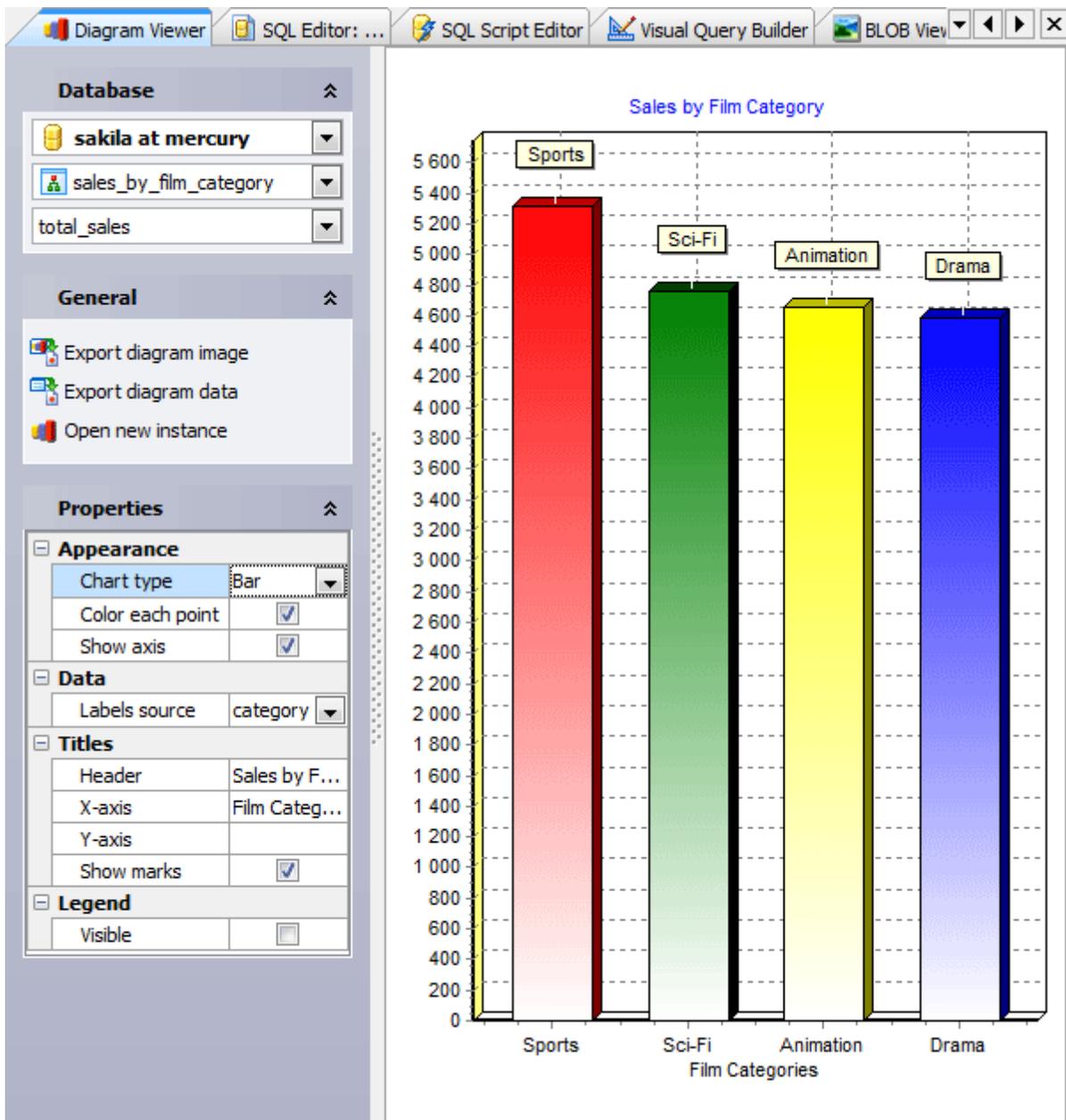
Diagram Viewer is a tool for representing data from a table or a query as a diagram in various ways. This means you can build a diagram represented as bars, lines, areas, points or pies, colored or not, with axis visible or not; specify axis labels source, diagram header and more. The Diagram Viewer also has the [Export diagram image](#)<sup>329</sup> and the Export diagram data features implemented, with a lot of formats supported.

- [Customizing diagram options](#)<sup>328</sup>
- [Exporting diagram as a graphical image](#)<sup>329</sup>



## 8.6.1 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

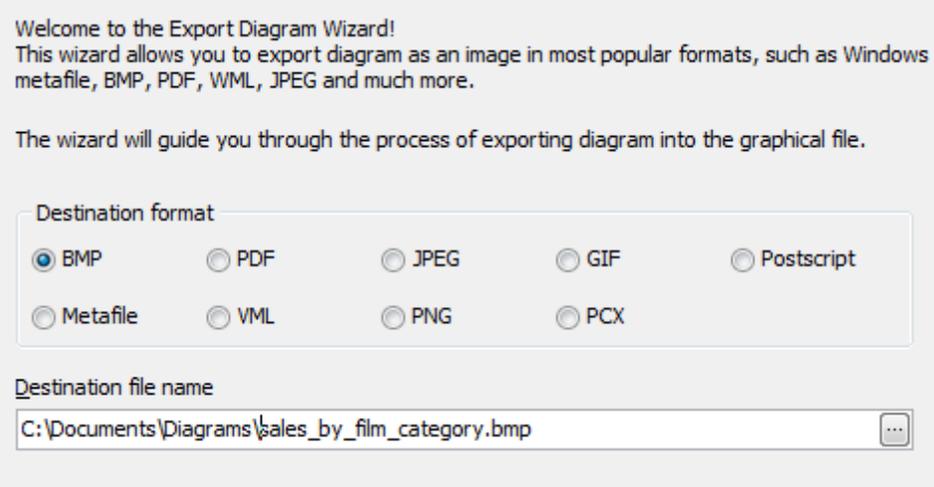
### Legend

The only **Visible** property of this subgroup specifies whether the legend rectangle should be represented on the right side of the diagram or not.

## 8.6.2 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.

## 8.7 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:

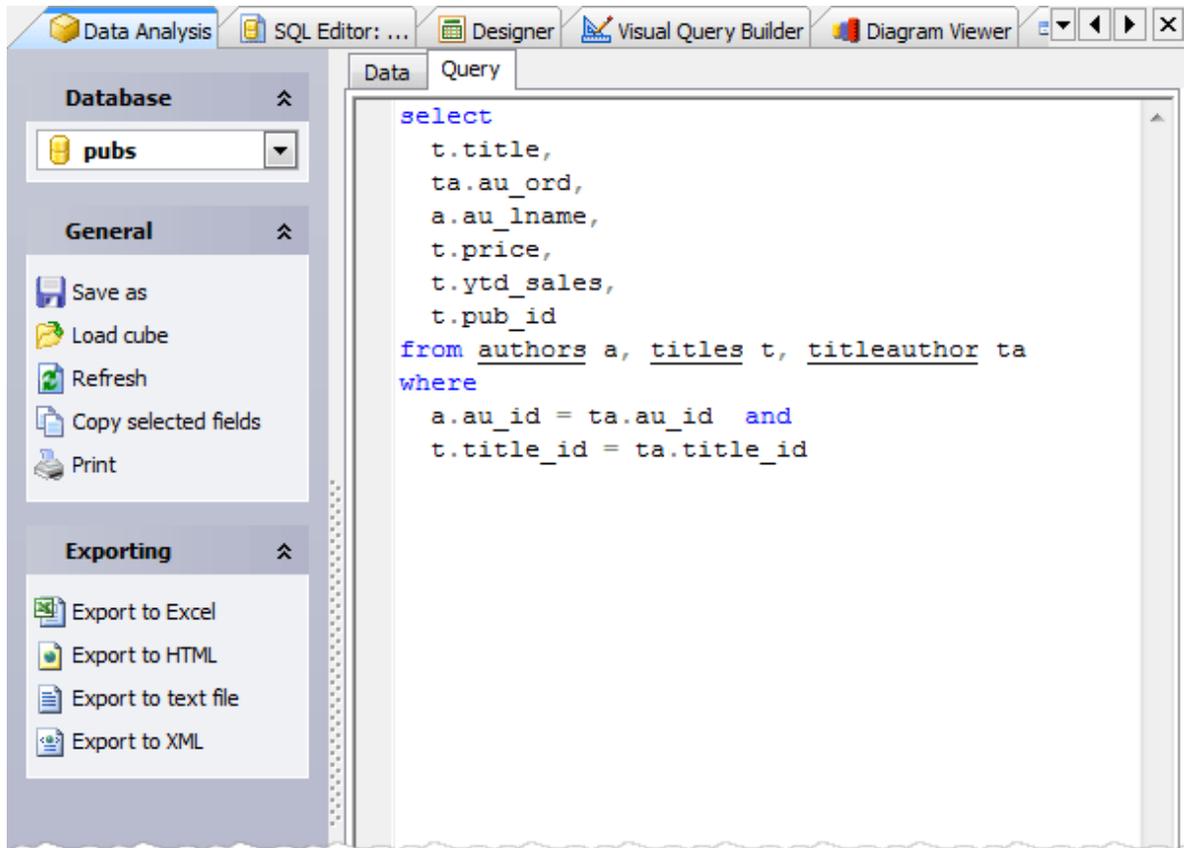
- [Input SELECT query](#)<sup>332</sup> in the **Query** window or load it from the .cub file.
- [Manage report data](#)<sup>333</sup> in the **Data** window.

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.

au		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"/> Blotchet-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 Vallev			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. Anvone?		14,99		14,99
<input type="checkbox"/> Hunter	Secrets of Silicon Vallev			20,00	20,00
<input type="checkbox"/> Karsen	Computer Phobic AND Non-Phobic Individuals: Beh		21,59		21,59
<input type="checkbox"/> Lockslev	Emotional Security: A New Alorithm	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. Anvone?		14,99		14,99
O'Leary 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 Anger 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. Anvone?		14,99		14,99
<input type="checkbox"/> del Castillo	Silicon Vallev Gastronomic Treats		19,99		19,99
Grand Total		59,87	147,02	146,82	353,71

### 8.7.1 Input SELECT query

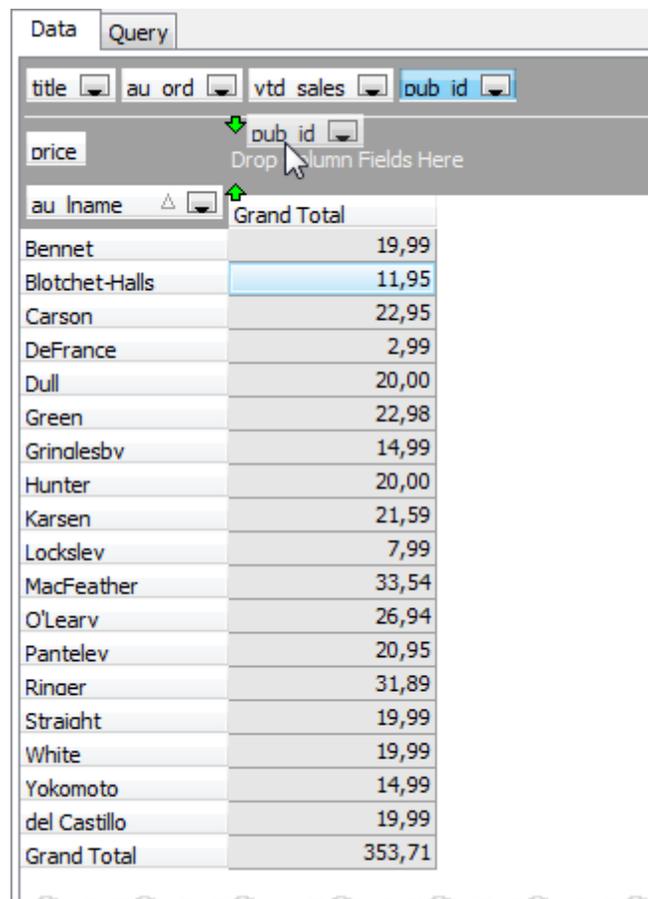
To get an [OLAP cube](#)<sup>333</sup>, enter SELECT query as a snowflake schema, represented by centralized fact tables (with numeric data) which are connected to multiple dimensions (the numeric data to be categorized by). Input the query text in the SQL Editor area directly or use "drag-n-drop" operation [SQL Editor](#) or Visual Query Builder areas and the Query tab of Data Analysis.



## 8.7.2 Managing report data

The [Data](#) tab allows you to manipulate the created OLAP cube appearance. At the beginning all the [query](#)<sup>332</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.

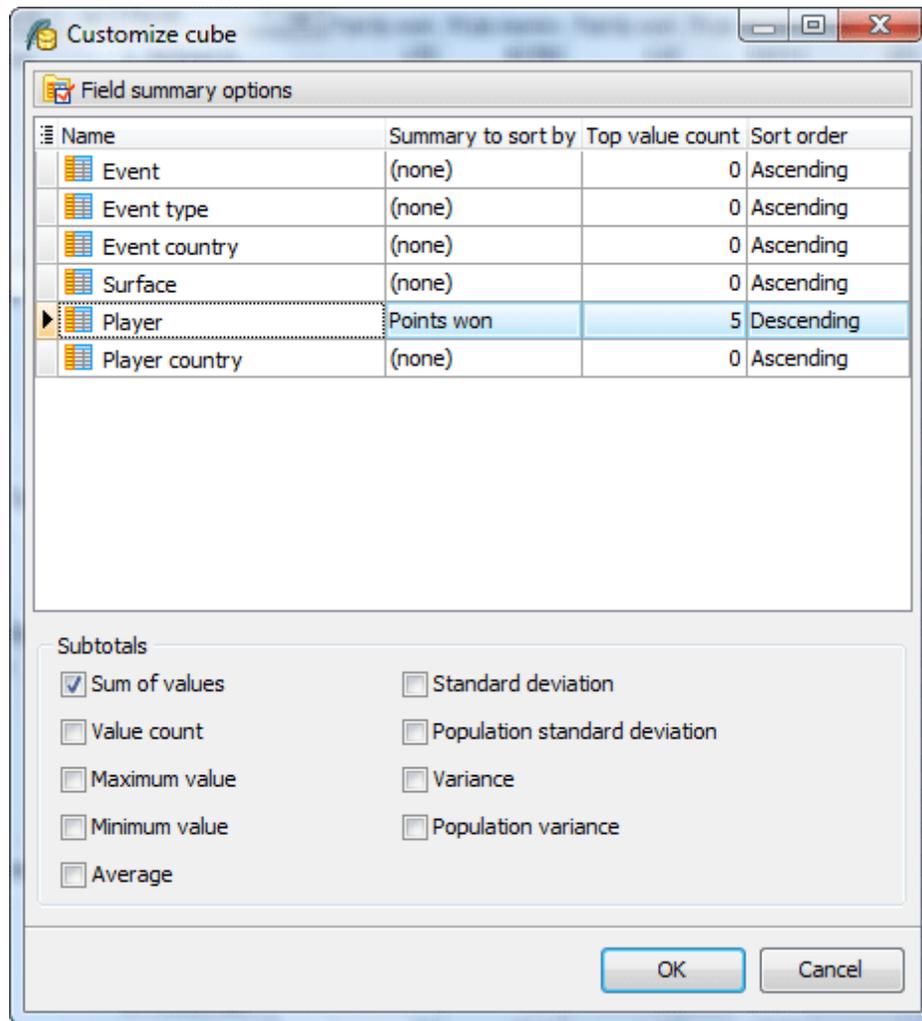
**Note:** Use for the Data Fields area only numerical columns.



The screenshot shows a query window with a grid of data. The columns are: title, au\_ord, vtd\_sales, pub\_id, price, au\_lname, and Grand Total. The data is sorted by au\_lname. The Grand Total row is highlighted in blue.

title	au_ord	vtd_sales	pub_id	price	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'Learn	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.



## 8.8 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>[338]</sup> for the data to be used in the report;
- [add all necessary objects](#)<sup>[338]</sup> to the report template;
- set the objects' format within the [Inspector window](#)<sup>[340]</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

First name	Last name	E-mail
Mike	Hillyer	Mike.Hillyer@sakilastaff.com
Payment		
Date	Amount	
04.26.2012	2.99	
04.27.2012	0.99	
04.23.2012	5.99	
04.26.2012	4.99	
04.23.2012	4.99	
04.24.2012	3.99	
04.23.2012	5.99	
04.28.2012	4.99	

## 8.8.1 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.

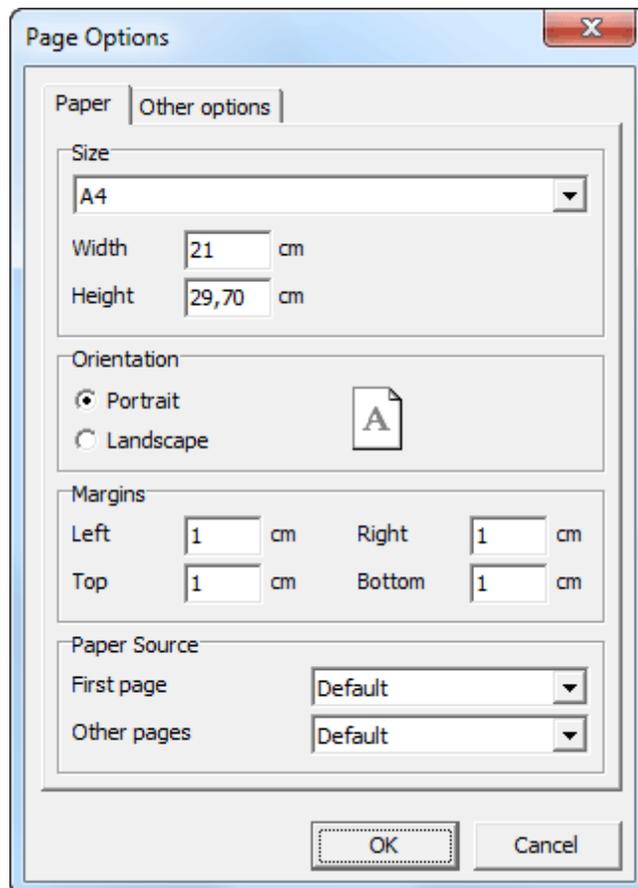
**Subreport object** allows inserting an additional report design page inside the basic one.

**System text** displays service information (date, time, page number, etc), as well as aggregate values.

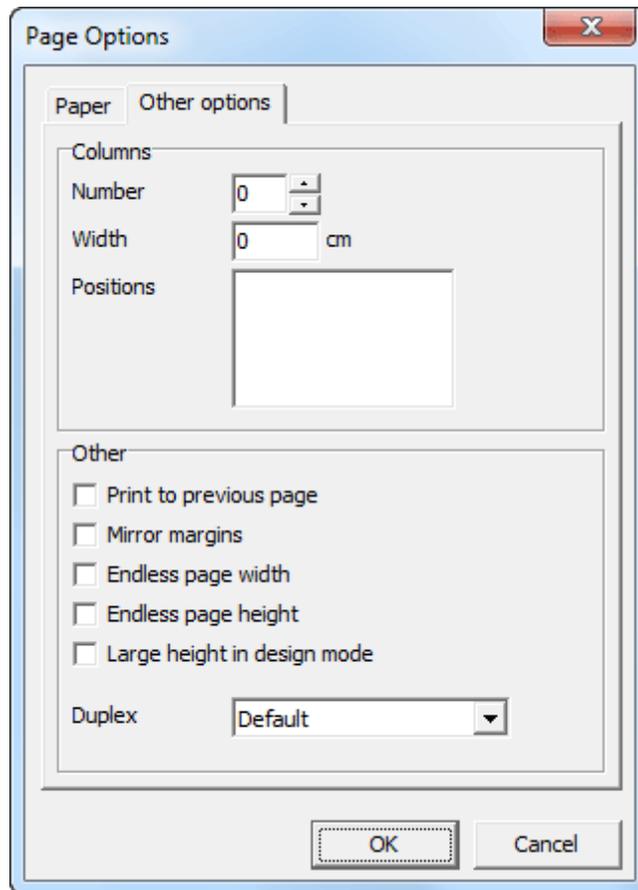
[Draw object](#) represents different geometrical figures (diagonal line, rectangle, rounded rectangle, ellipse, triangle, and diamond).

### 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.

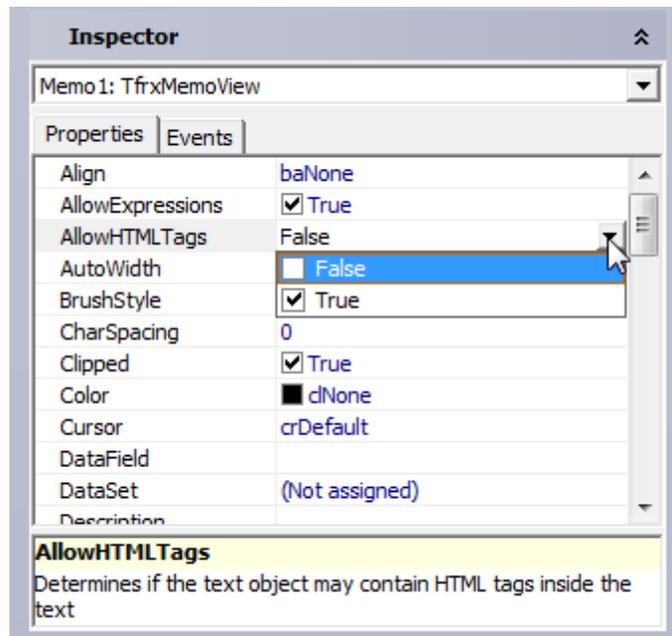
The [Mirror margins](#) option switches right and left margins of page for even pages during previewing or printing a report.

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.

## 8.8.2 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).

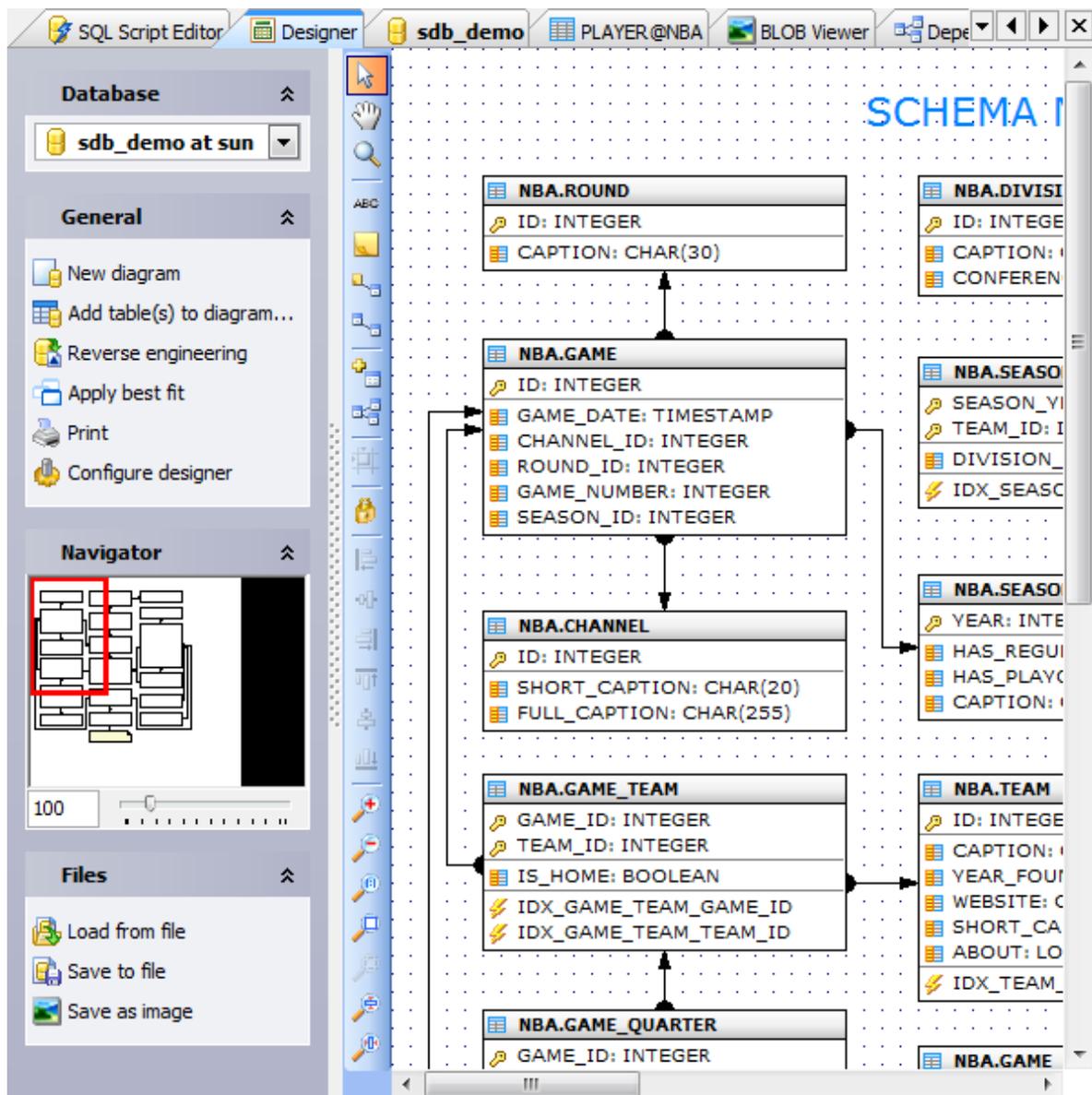
**Rotation** - specify the angle of the text rotation in the field.

**WordWrap** - if it is disabled, the long lines will be simply cut off.

## 8.9 Schema Designer

Schema Designer allows you to create physical Entity Relationship Diagram that will represent objects in your Oracle 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>[344]</sup>
- [How can I add a relationship between tables?](#)<sup>[344]</sup>
- [How can I delete a diagram object?](#)<sup>[344]</sup>
- [How can I work with diagram objects?](#)<sup>[345]</sup>

**See also:** [Designer Navigation bar](#)<sup>[345]</sup>, [Schema Designer Toolbox](#)<sup>[345]</sup>.

### Adding a table

You can add an existing Oracle table to the diagram using popup menu in the working area, or with the corresponding link on the [Navigation bar](#)<sup>[345]</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>[82]</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>[345]</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>[82]</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>[75]</sup>, [Index Editor](#)<sup>[79]</sup>, [Create Trigger Wizard](#)<sup>[88]</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.

## **8.9.1 Designer Navigation Bar**

The [Navigation Bar](#) of [Schema Designer](#) provides you the following opportunities:

Use the [Database](#) drop-down list to move around your Oracle 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>[380]</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).

## **8.9.2 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.

## 8.10 PL/SQL Debugger

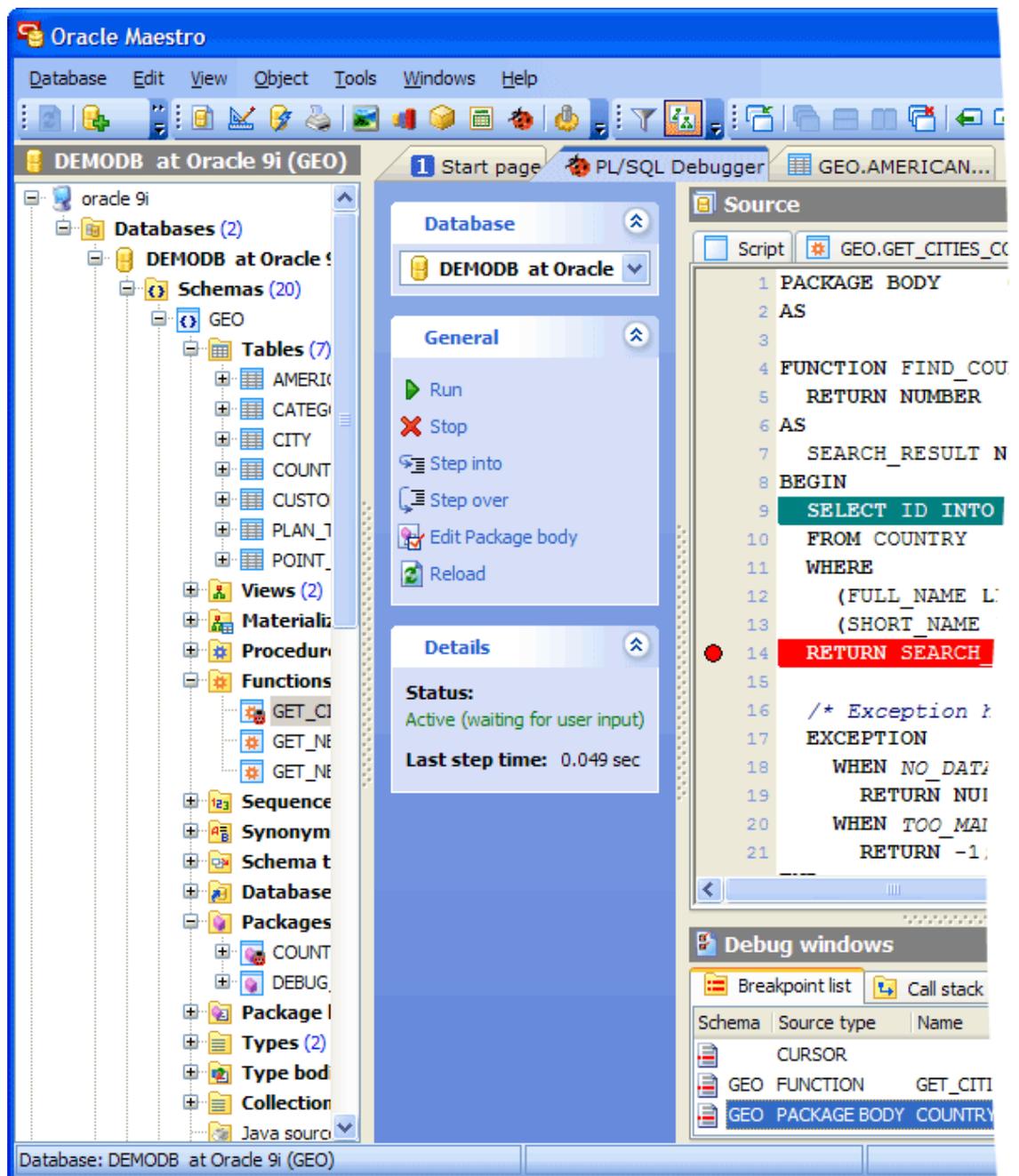
The [PL/SQL Debugger](#) tool allows you to debug [PL/SQL](#) code such as procedures and functions (both stand-alone and packaged) using traditional debugging features such as setting breakpoints, viewing variable values, and examining the call stack.

To start working with the debugger, choose the [Tools | PL/SQL Debugger](#) main menu item. You can also use the corresponding link at the [Navigation bar](#) of the [Procedure](#)<sup>[118]</sup>/[Function Editor](#)<sup>[125]</sup> or select the appropriate command from the object pop-up menu in the [Explorer](#) tree.

**Note:** To debug a function/procedure with nested ones, compile them with [debug information](#) in advance. Procedures/functions/packages that were compiled in this way have differing object icon in the [Explorer](#) tree.

**Note:** You can also debug an anonymous [PL/SQL](#) block. Just open [PL/SQL](#) Debugger and paste/enter the script.

- [Debugging process](#)<sup>[349]</sup>
- [Debug information](#)<sup>[351]</sup>



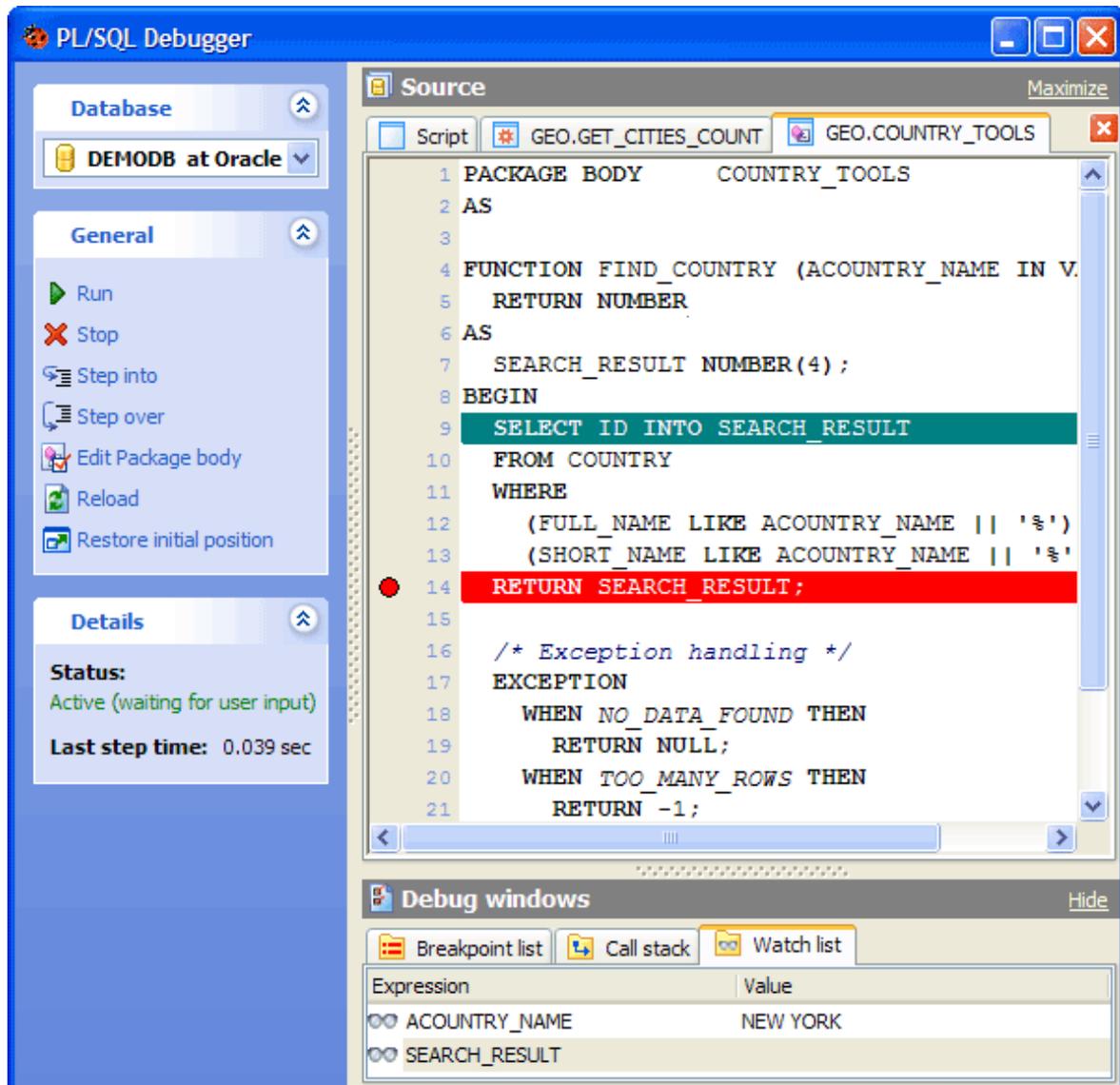
### 8.10.1 Debugging process

For those programming errors that are really hard to track, Oracle Maestro provides an integrated PL/SQL debugger. The main principles of the debugging process within the PL/SQL Debugger are similar to the same tools methods.

The PL/SQL Debugger window consists of three parts: Source, Debug window, and Navigation bar.

The Source tab contains script editors for each program unit, so that you can easily switch between them to view the source, set/remove breakpoints, and so on. The Debug window is provided to represent the debugging process characteristics such as Breakpoints information, Call Stack, and Watch List.

- [Controlling execution](#) <sup>350</sup>
- [Using breakpoints](#) <sup>351</sup>
- [Debug information](#) <sup>351</sup>



### Controlling execution

You can set the procedure parameters before the debugging process.

To start the debugger, use the **Run (F9)** link at the Navigation bar.

After starting the debugger, execution will pause on the first script statement. After this, you can control execution with the buttons in the Navigation bar:

Use the [Run \(F9\)](#) link at the [Navigation bar](#) to run the script until completion, a breakpoint or an exception.

To move to the next line, click the [Step Over](#) icon or use **F8**. The nested procedure will be executed, but you will not step into the source.

[Step into \(F7\)](#) allows you to open the nested procedure as the new [Source](#) instance to debug it line-by-line.

**Note:** The nested procedure body is available to debug if it was [compiled with debug information](#).

#### **Stop (Ctrl + F2)**

Use the link to abort the debugging process. To make changes in the procedure body, use the corresponding [Procedure Editor](#).

### **Using Breakpoints**

Breakpoints can be used to halt program execution on a certain line in your [PL/SQL](#) code. When execution halts, you can view variables, step through the code, and so on.

To set a breakpoint, just click on the grey leftmost edge of the code window opposite to a necessary line. To delete a breakpoint, simply click on the breakpoint mark again.

It's also possible to disable/enable a breakpoint by clicking the breakpoint indicator with pressed **Ctrl**.

## **8.10.2 Debug information**

### [Breakpoint list](#)

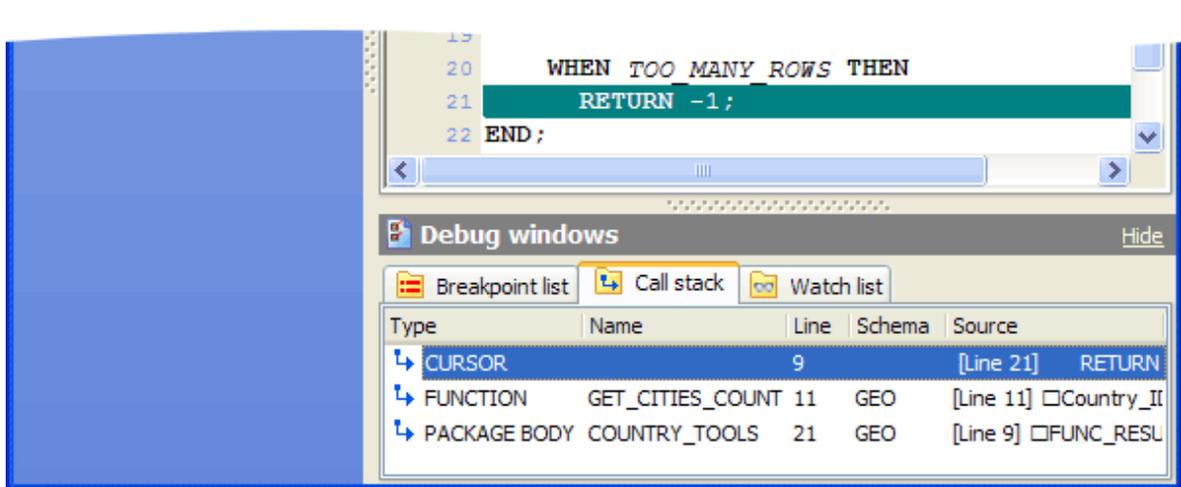
The tab contains all the information about the session breakpoints.

### [Viewing variable values](#)

If you want to view variables values, you need to create a watch with the [Watch list](#) tab pop-up menu. Just select [Add Watch](#) and enter the variable name.

### [Viewing the call stack](#)

Whenever the debugger is paused, you can use the [Call stack](#) window to see the procedure flow as a stack of method calls that got you to the current line. It is automatically updated after each debug step.



## 8.11 Session Browser

**Session Browser** is very useful tool for DBAs who want to monitor the users' activity (in fact, there are potentially thousands of sessions in a database at any one time). You can view details for each session (such as login, connect time, database name, client host, last SQL statement executed and more) as well as group and filter sessions.

To access the **Session Browser** window, select the corresponding item from the **Tools** menu.

The screenshot displays the Oracle Maestro Session Browser interface. On the left, the 'Database' dropdown is set to 'NSN\_DEMO at Oracle 9i'. Below it, the 'General' pane contains 'Refresh', 'Kill', and 'Immediate kill' buttons. The main 'Sessions' table lists several active sessions:

ID	Termination	User	Logon time	Client application
10	SATURN	EMPLOYEE	08.05.2007 10:23:17	OradeCodeFactory.ε
14	ORACLE	SYS	08.05.2007 11:39:24	OradeMaestro.exe
15	SATURN	EMPLOYEE	08.05.2007 10:22:43	OradeCodeFactory.ε
17	NEPTUN	TEST_DBA	08.05.2007 11:48:55	OradeMaestro.exe
18	SATURN	EMPLOYEE	08.05.2007 10:23:17	OradeCodeFactory.ε
20	ORACLE	EMPLOYEE	08.05.2007 11:27:11	OradeCodeFactory.ε

The 'Properties' pane for session ID 10 shows the following details:

- Common:** Name: 10 EMPLOYEE@SATURN
- User session:** ID: 10, Login: EMPLOYEE, Connect time: 08.05.2007 10:23:17, Database name: NSN\_DEMO, Application module, Session type: USER
- Client:** (Details are partially obscured)

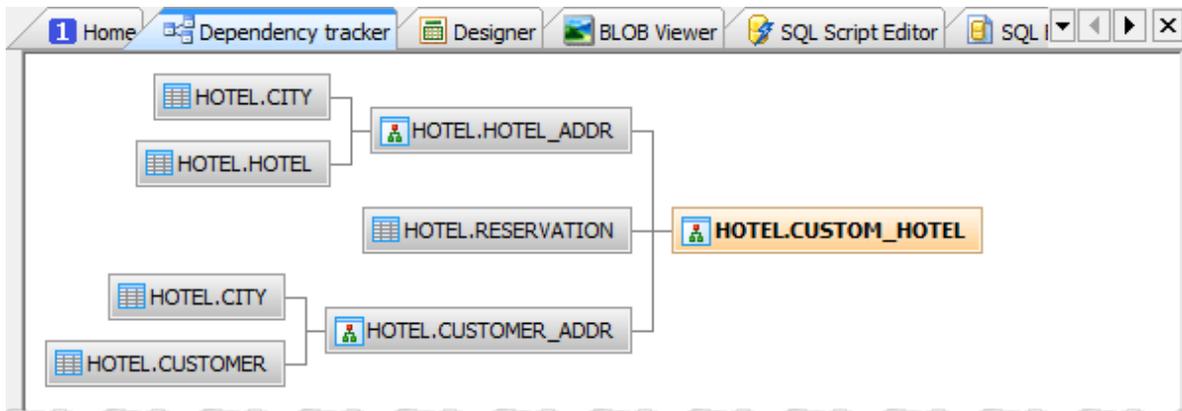
At the bottom, there are buttons for 'Session' and 'Last SQL'.

## 8.12 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.

This tool allows you to see the way any database object is involved in the net of scheme dependencies. The selected object is displayed in a highlighted rectangle in the center of the working area. The right side of this area represents objects depending on the selected one. The left side represents objects on which the selected object depends.

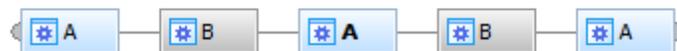
To highlight all occurrences of an object in the diagram, click the appropriate rectangle. Double click an object to change the tracker focus to this object. Right click a rectangle to display a popup menu with common operations related to the appropriate object.



The recursive dependencies are marked with a semicircle. This means marked object depends on itself directly or via other objects.

### Example

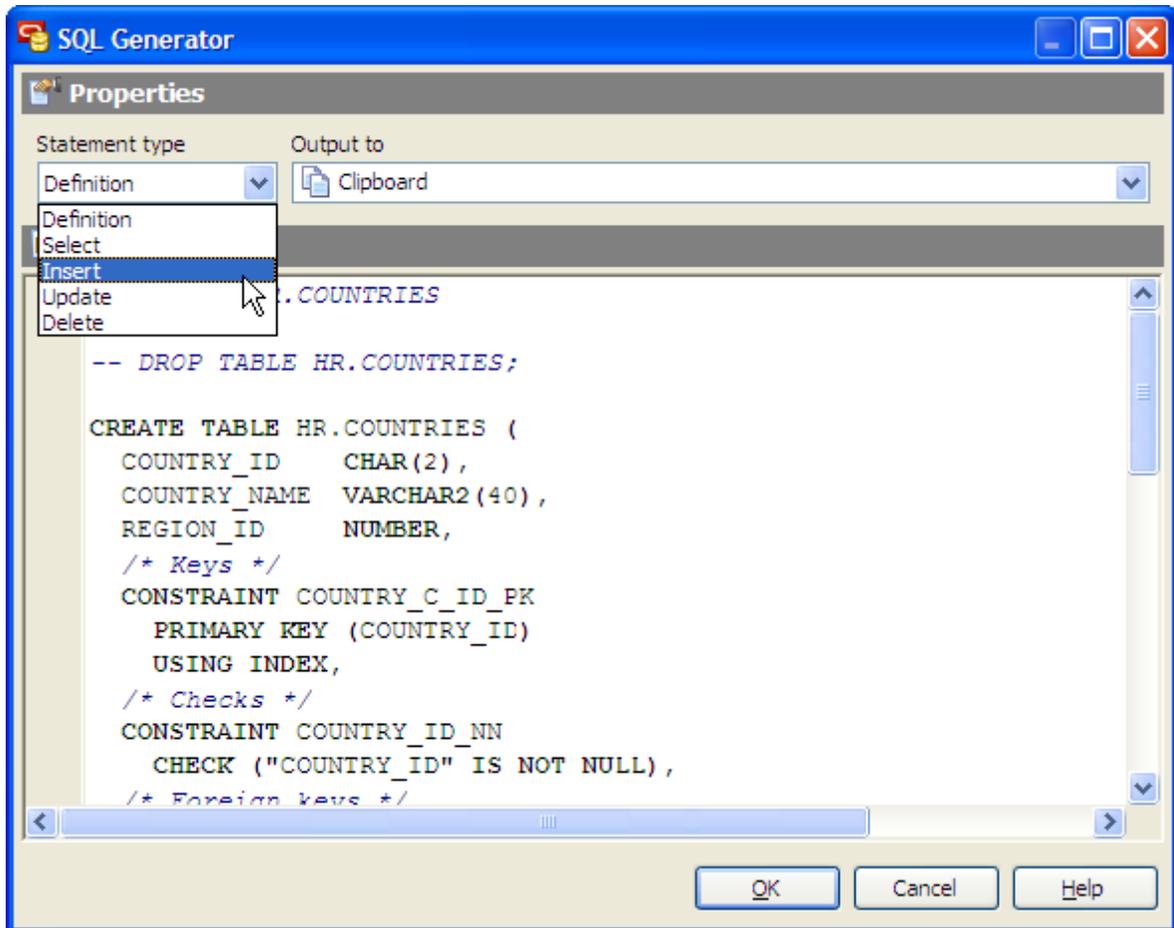
Suppose we have two procedures A and B calling each other. The tracker displays these dependencies in the following way:



## 8.13 SQL Generator

Among other features Oracle 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.



## 8.14 DML procedures generation

Oracle 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 Oracle 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

## 8.15 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

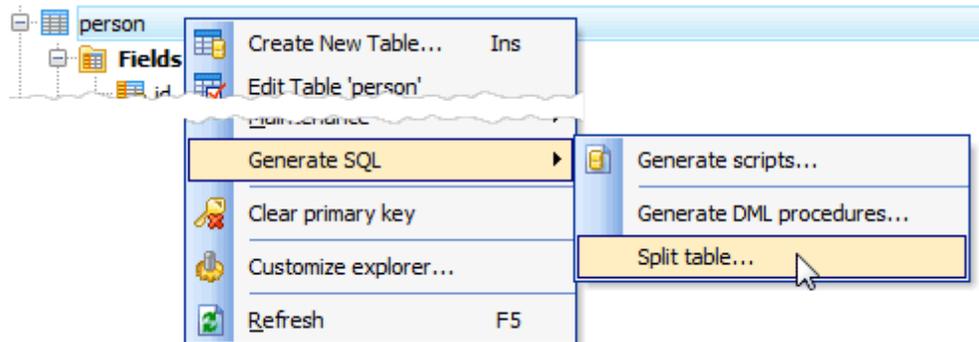
Abilities	
Insert	<input checked="" type="checkbox"/>
Update	<input checked="" type="checkbox"/>
Delete	<input checked="" type="checkbox"/>

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

## 8.16 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. Oracle 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	city

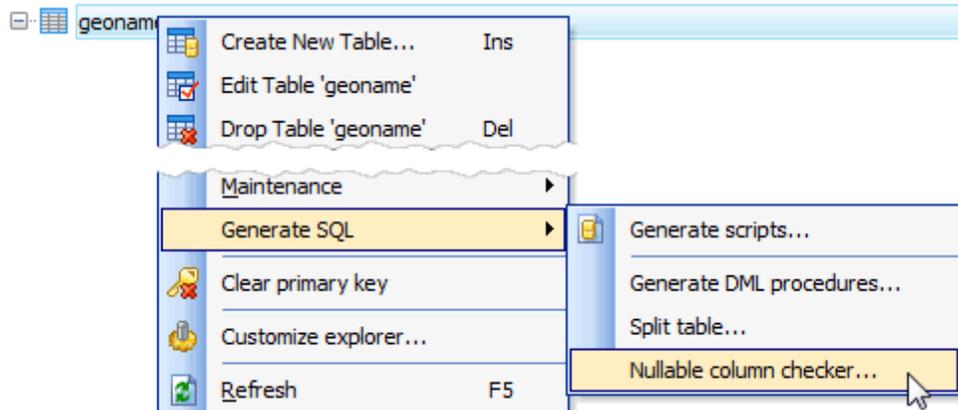
Then click Ready and get the following tables:

	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

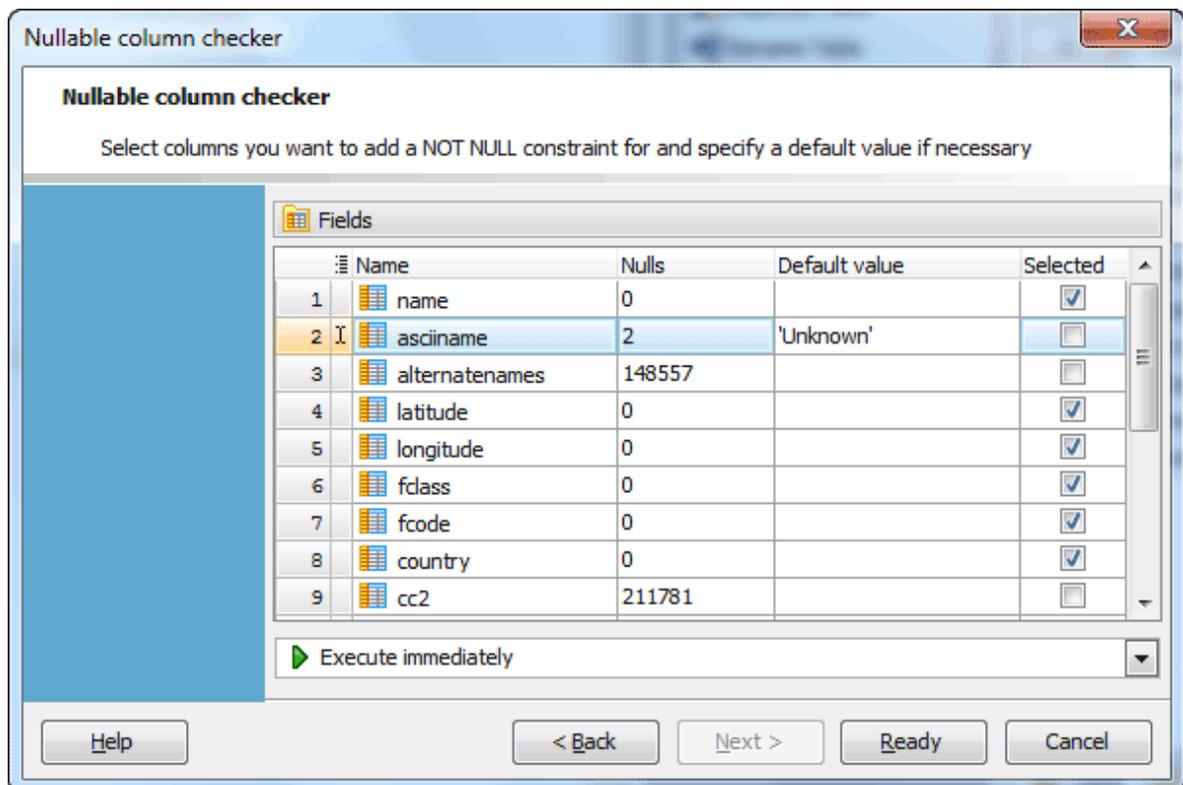
	id	city
1	1	Boston
2	2	New York

## 8.17 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.



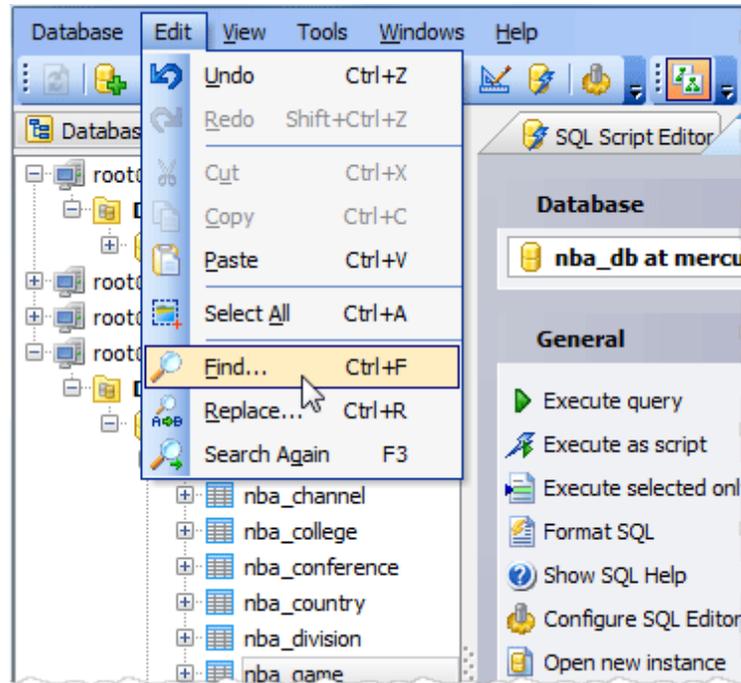
To get SQL scripts enforcing NOT NULL constraints to columns of an existing table, select the necessary columns, specify the default values to be used instead of existing columns NULLs and select the action to perform after the generation. The created scripts can be copied to Clipboard, saved to a file, sent to [SQL\\_Script\\_Editor](#)<sup>[316]</sup> or executed immediately.



## 8.18 Dialogs

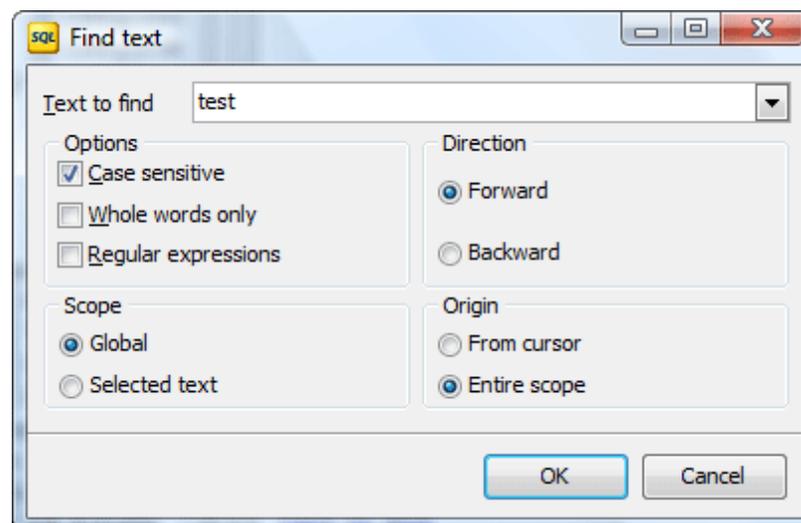
Oracle Maestro provides two dialogs for searching and replacing text in the editor areas of the database tools. Both of them are available through the popup menu of the editor area.

- [Find Text dialog](#) <sup>363</sup>
- [Replace Text dialog](#) <sup>364</sup>



### 8.18.1 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

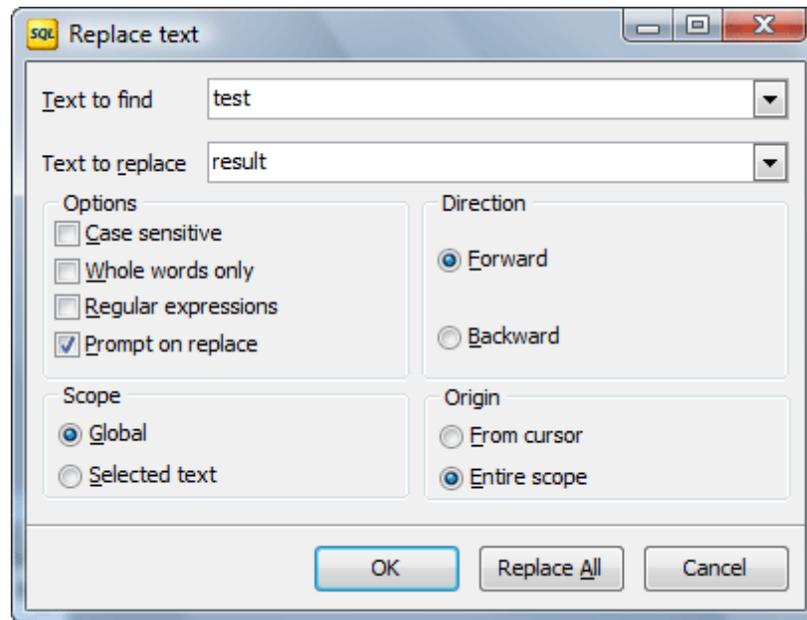
The search starts at the cursor's current position, and then proceeds either forward to the end of the scope, or backward to the beginning of the scope depending on the **Direction** setting. **From Cursor** is the default setting.

### 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.

## 8.18.2 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.

## 9 Options

Oracle 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>[368]</sup>  
General Oracle Maestro options: environment style, confirmations, window restrictions, explorer tree, [SQL Editor](#), [Visual Query Builder](#), etc.
- [Editors & Viewers](#)<sup>[390]</sup>  
Customizing of all the SQL editors - [SQL Editor](#), [SQL Script Editor](#), etc.
- [Appearance](#)<sup>[398]</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>[407]</sup> for details).

It is a good idea to check through these settings before you start working with Oracle Maestro. You may be surprised at all the things you can adjust and configure!

## 9.1 Application

The **Application** section allows you to customize common rules of Oracle Maestro behavior. The section consists of several tab; follow the links to find out more about each of them.

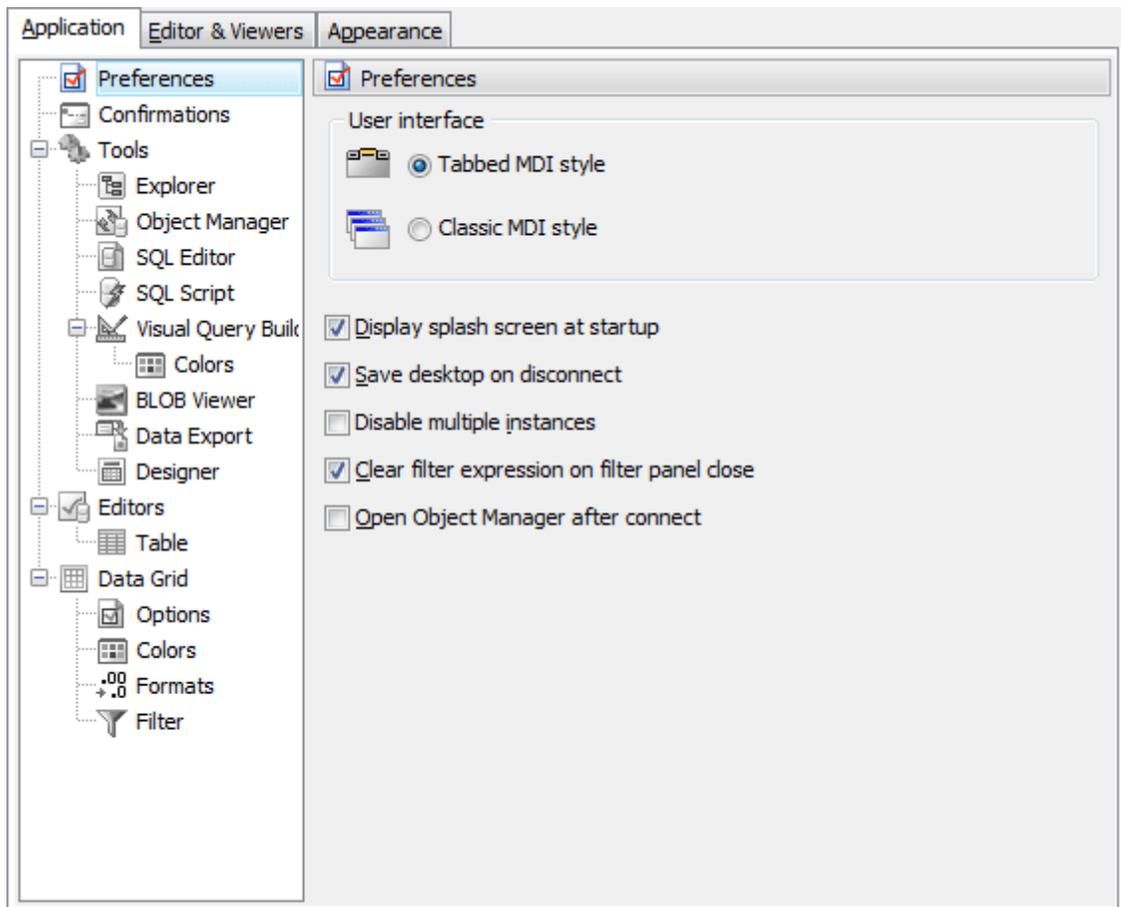
- [Preferences](#)<sup>[368]</sup>
- [Confirmations](#)<sup>[369]</sup>
- [Directories](#)<sup>[371]</sup>
- [Tools](#)<sup>[371]</sup>
  - [Explorer](#)<sup>[373]</sup>
  - [Object Manager](#)<sup>[374]</sup>
  - [SQL Editor](#)<sup>[374]</sup>
  - [SQL Script Editor](#)<sup>[375]</sup>
  - [Query Builder](#)<sup>[376]</sup>
  - [BLOB Viewer](#)<sup>[378]</sup>
  - [Export data](#)<sup>[379]</sup>
  - [Database Designer](#)<sup>[380]</sup>
- [Object Editors](#)<sup>[381]</sup>
  - [Table](#)<sup>[383]</sup>
  - [Data Grid](#)<sup>[383]</sup>
    - [Colors](#)<sup>[386]</sup>
    - [Formats](#)<sup>[386]</sup>
    - [Filter](#)<sup>[388]</sup>

### 9.1.1 Preferences

User interface area allow you to select your favorite UI style according to your preferences.

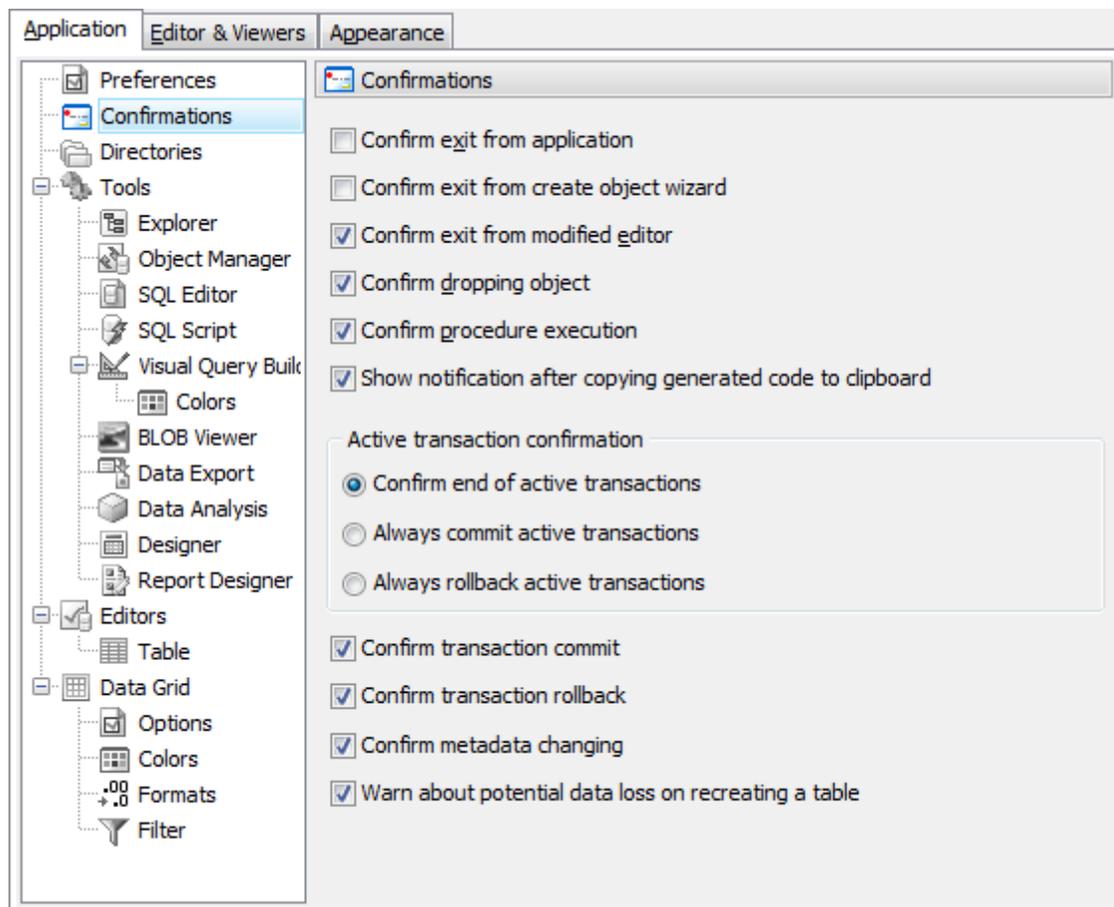
- [Display splash screen at startup](#)  
Displays the splash screen on Oracle Maestro startup.
- [Save desktop on disconnect](#)  
Saves all the database windows and their positions on disconnecting from the database.
- [Disable multiple instances](#)  
Prohibits running multiple instances of Oracle 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.



## 9.1.2 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 Oracle Maestro will commit or rollback transactions without asking.

**Confirm metadata changing**

If this option is checked, the program requires confirmation for changing metadata.

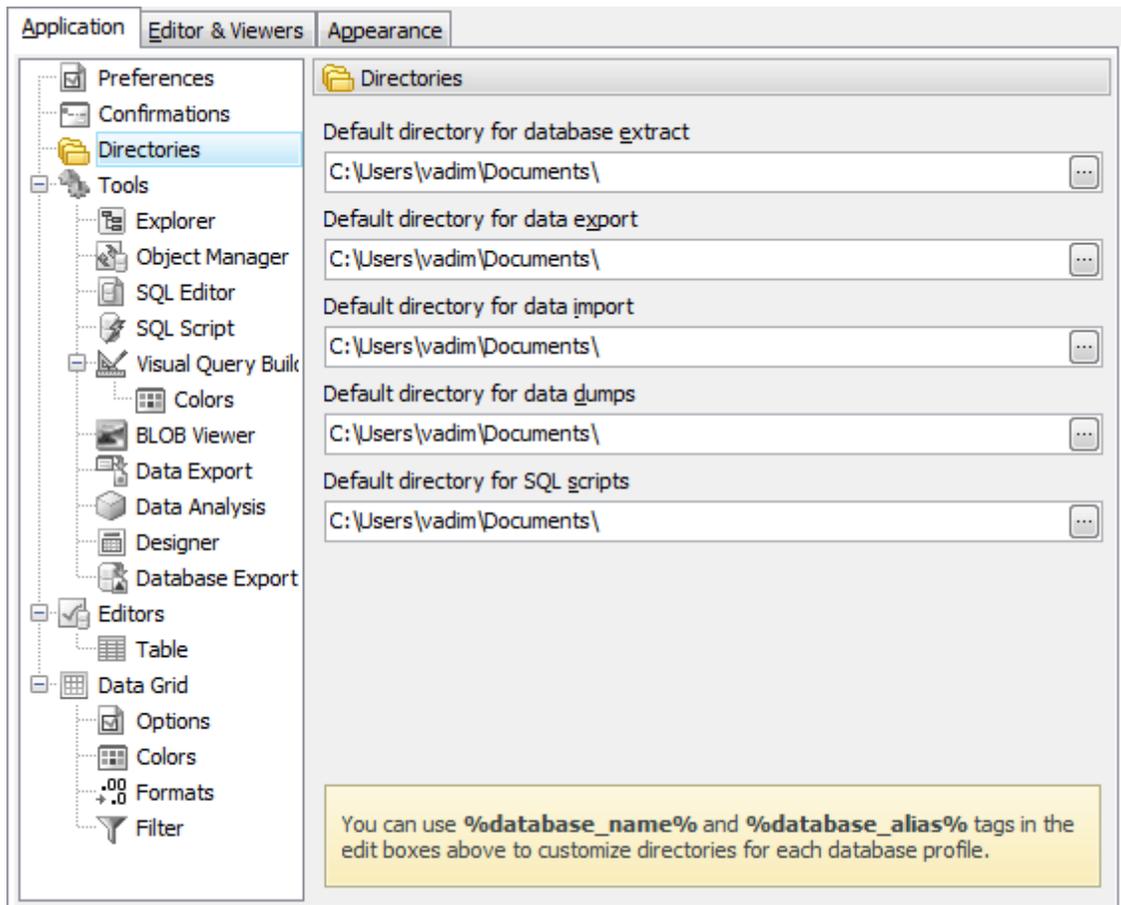
### 9.1.3 Directories

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:**

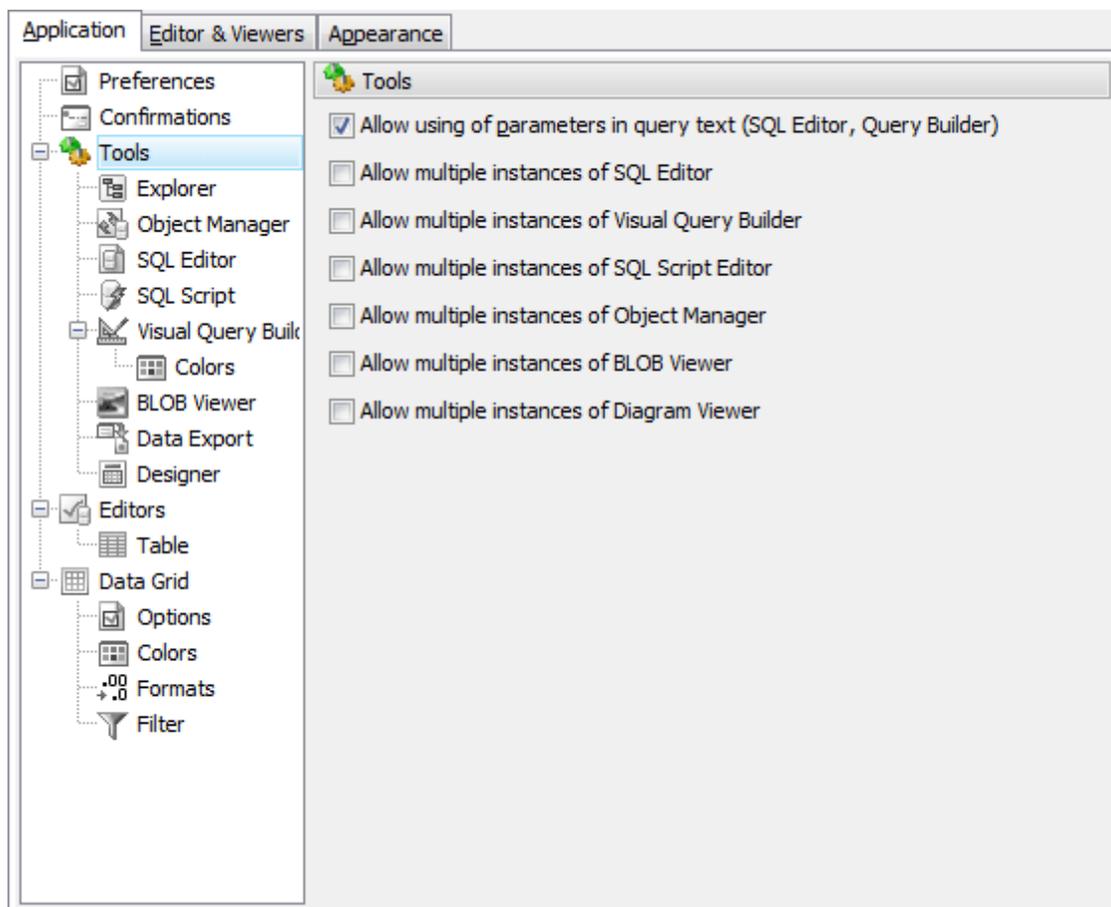
To store all SQL scripts in folders sorted by databases in the "C:\SQL Scripts\" directory, specify the default directory for SQL scripts as follows:

```
C:\SQL Scripts\%database_name%
```



### 9.1.4 Tools

Below you will find a detailed description of the following tools options.

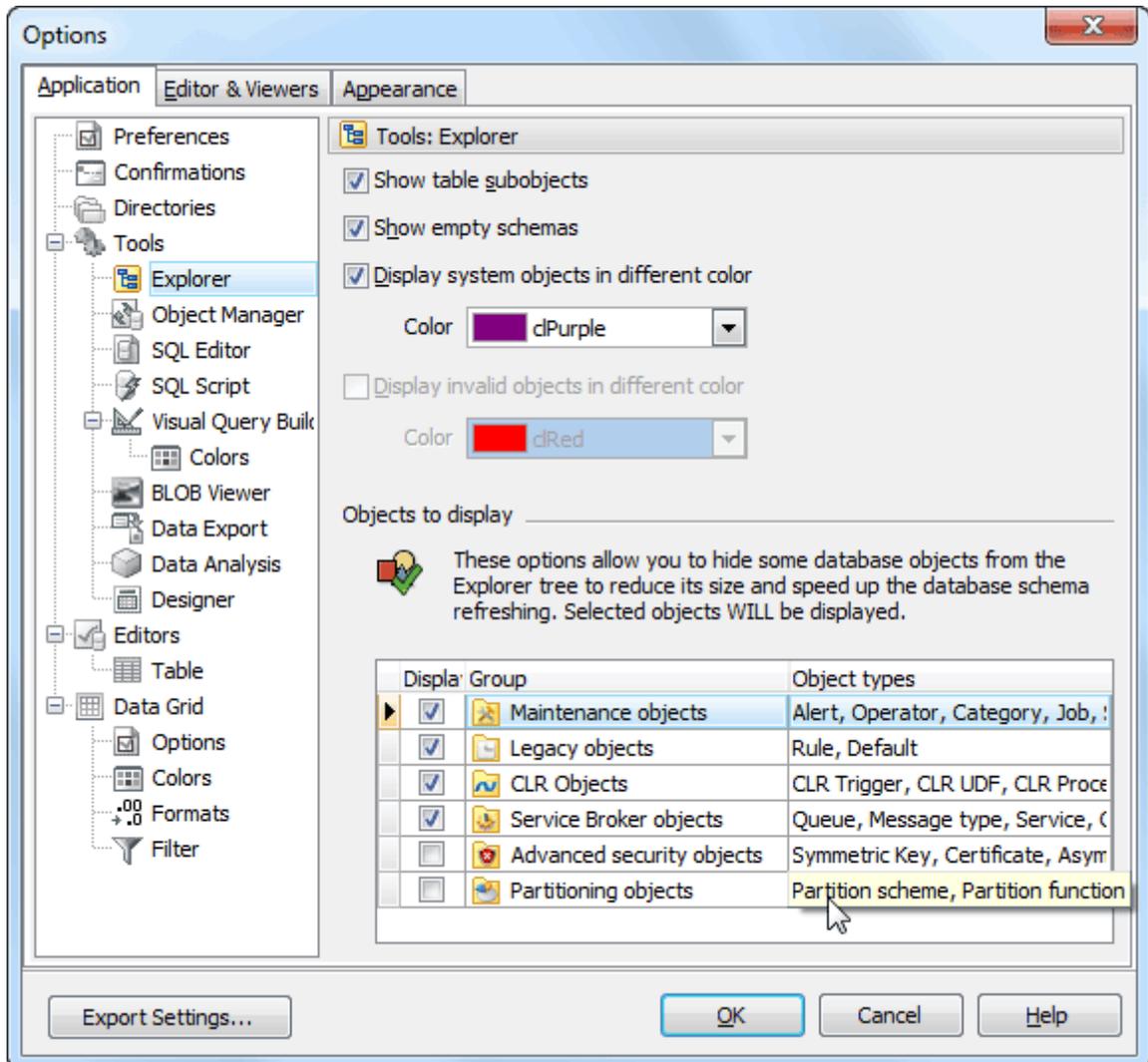


- [Allow using of parameters in query text](#)  
Check this option to be able to use query parameters in [SQL Editor](#)<sup>[264]</sup> and [Visual Query Builder](#)<sup>[269]</sup>.
- [Allow multiple instances of SQL Editor](#)  
Check this option to be able to use multiple instances of [SQL Editor](#)<sup>[264]</sup> simultaneously.
- [Allow multiple instances of Visual Query Builder](#)  
Check this option to be able to use multiple instances of [Visual Query Builder](#)<sup>[269]</sup> simultaneously.
- [Allow multiple instances of SQL Script Editor](#)  
Check this option to be able to use multiple instances of [SQL Script Editor](#)<sup>[316]</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>[321]</sup> simultaneously.
- [Allow multiple instances of Diagram Viewer](#)  
Check this option to be able to use multiple instances of [Diagram Viewer](#)<sup>[327]</sup>

simultaneously.

### 9.1.4.1 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.
- Show empty schemas**  
Shows/hides empty schemas in the explorer tree.

**Note:** To hide empty schemas, check the option along with the [Refresh\\_hole\\_database\\_on\\_connect](#)<sup>[23]</sup> profile option and reconnect to the database.

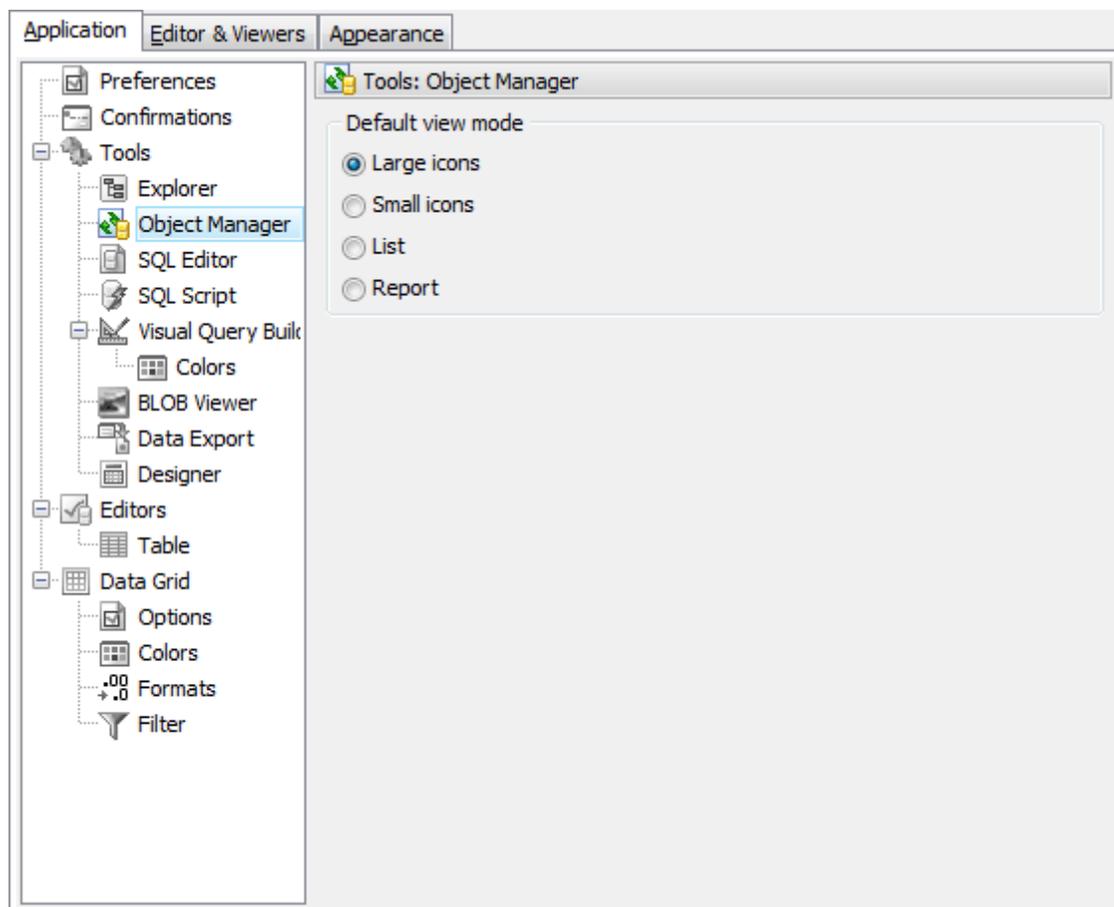
Display system objects in different color  
Represents all system objects in selected color.

Display invalid objects in different color  
Represents all invalid 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.

#### 9.1.4.2 Object Manager

Below you will find a detailed decryption of the following [Object Manager](#) options.

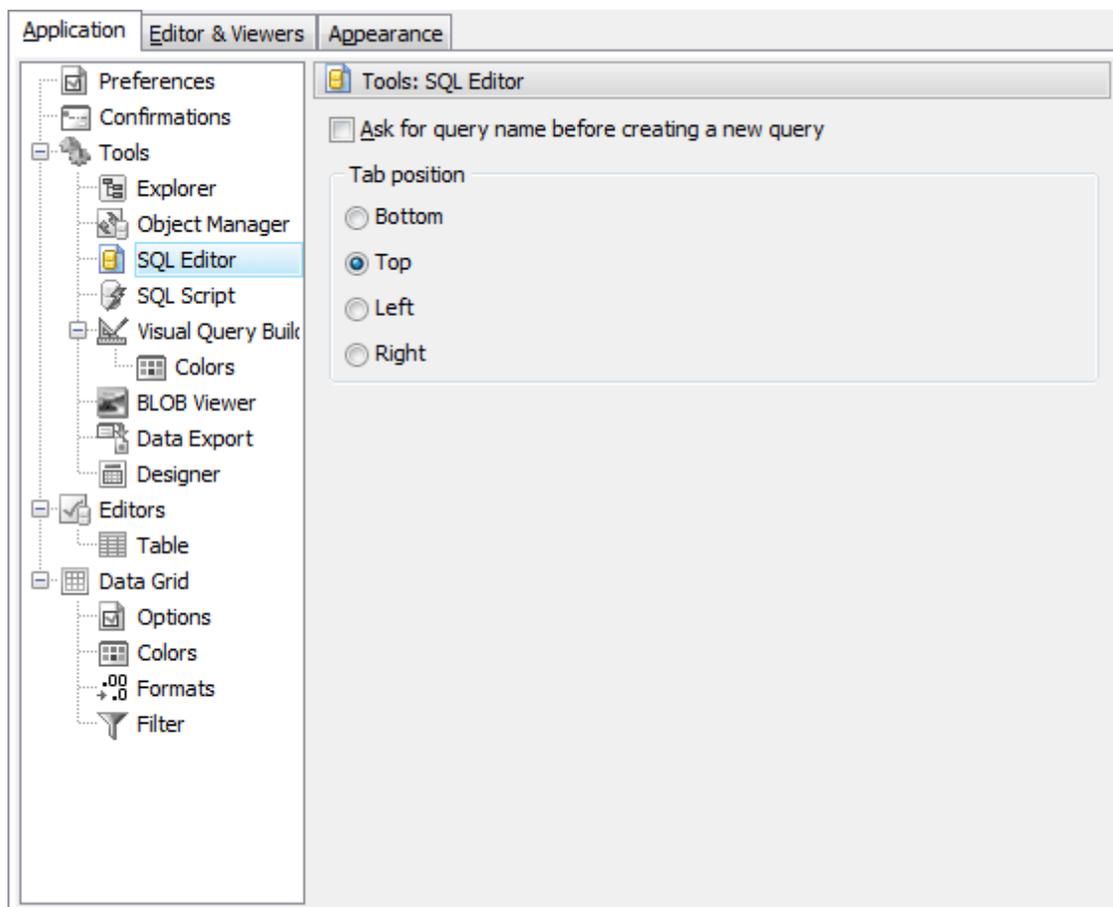


##### Default view mode

Defines which view mode (*large icons, small icons, list or report*) is applied to Object Manager by default.

#### 9.1.4.3 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>264</sup> asks for a query name each time you create a new query.

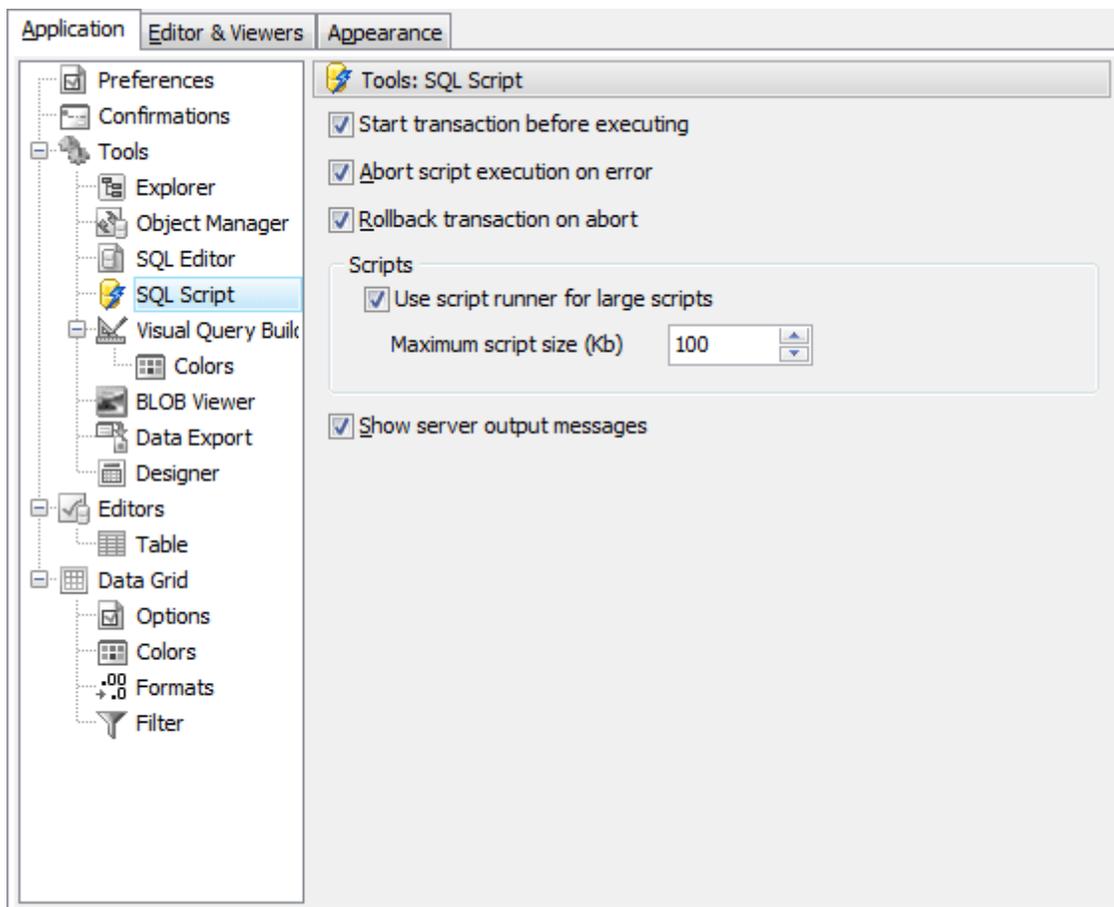
**Auto commit**

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.

#### 9.1.4.4 SQL Script Editor

Below you will find a detailed description 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**

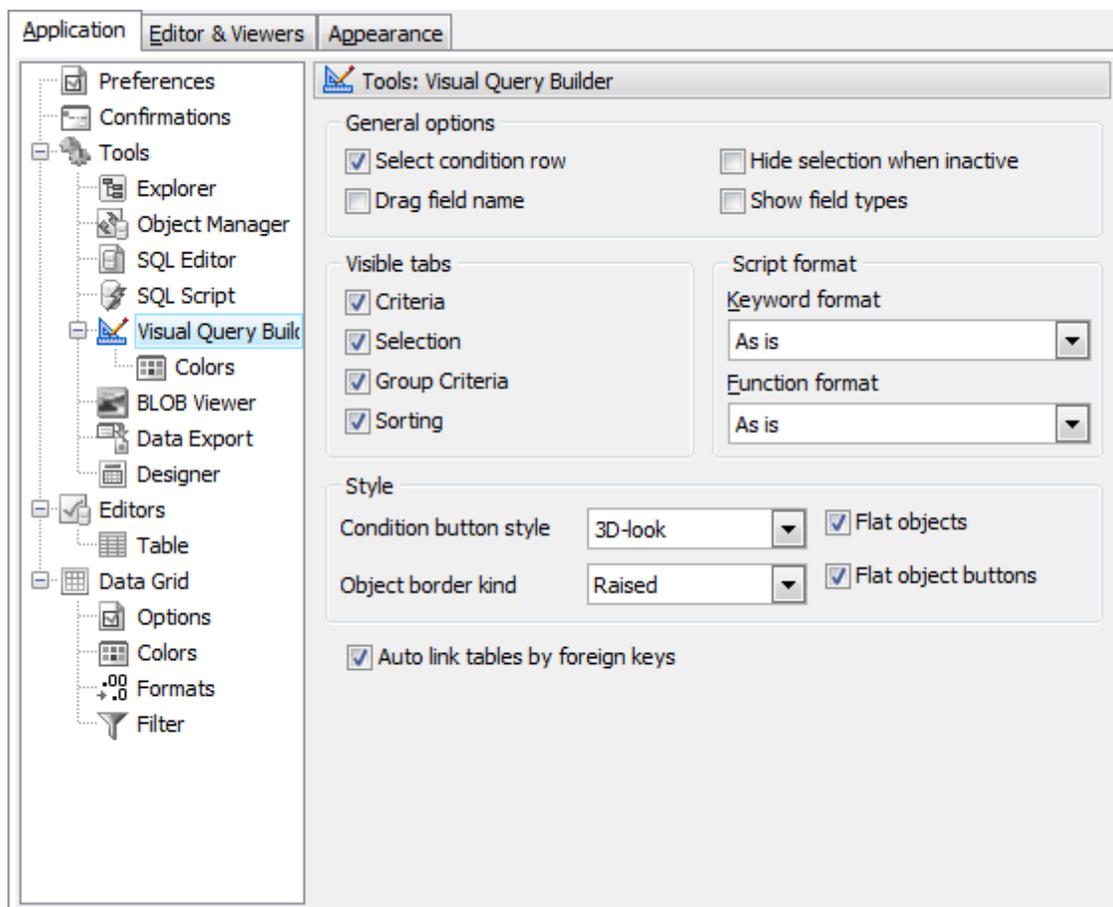
Check the box to execute large script in the fastest way. You can change the maximum size of a script to execute without script runner.

**Show server output messages**

Turn the option ON to see warning messages generated by the server.

#### 9.1.4.5 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>[269]</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

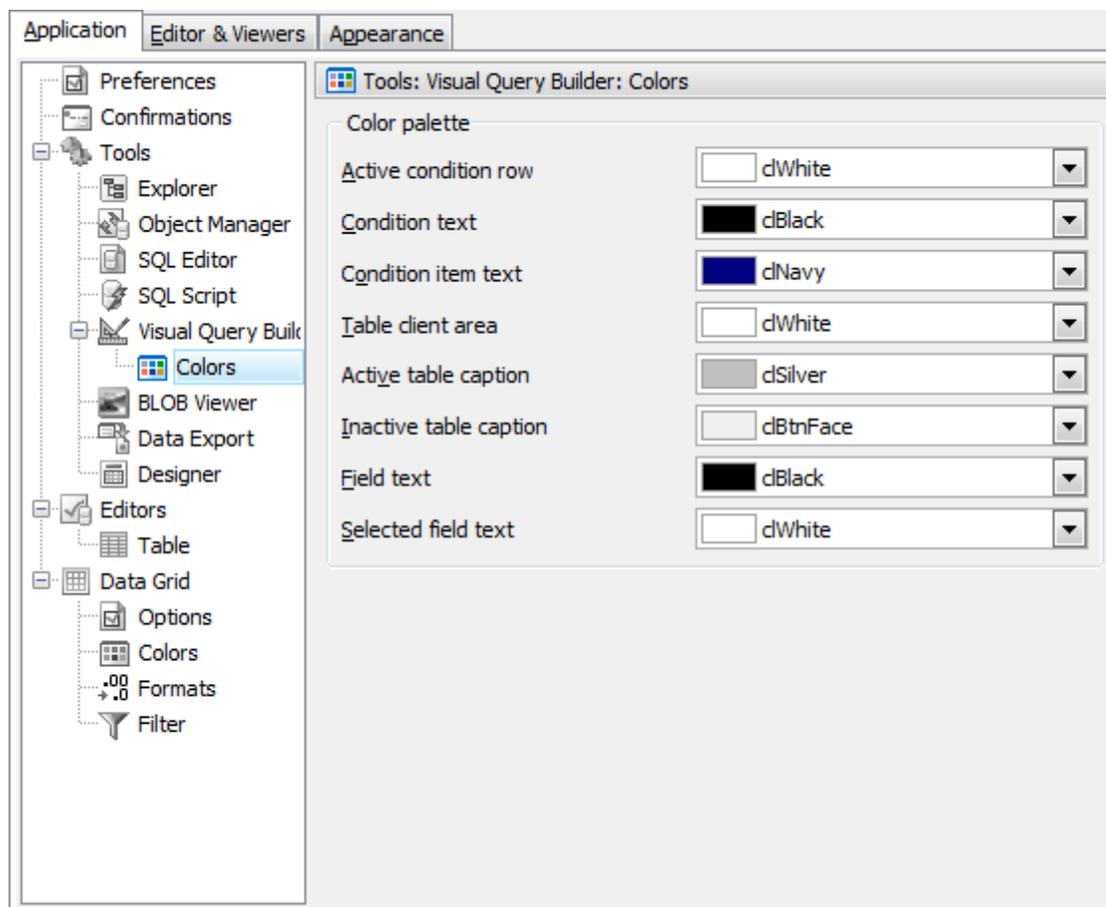
If tables that have foreign key reference are placed in **Query Builder**, in diagram they will be auto linked.END

## 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.

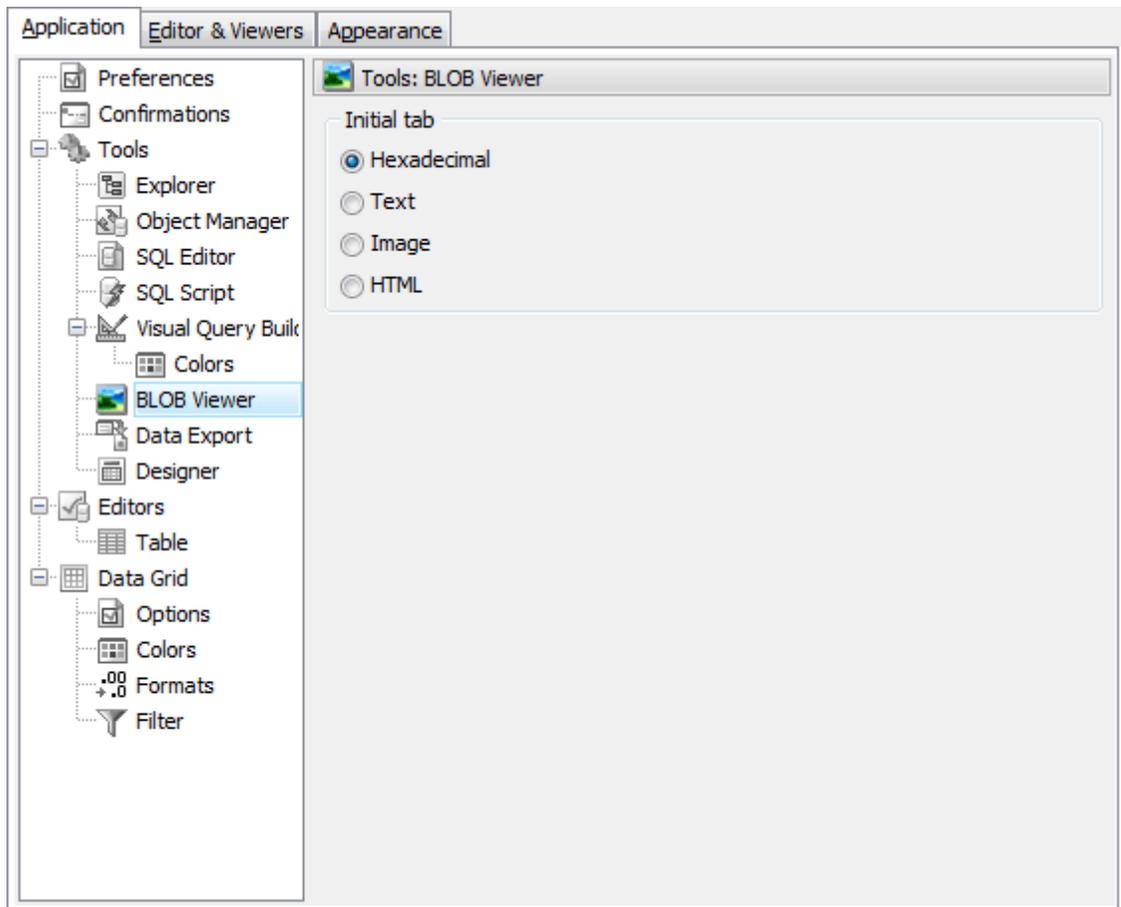
### 9.1.4.5.1 Colors

The tab is provided to editing of the **Query Builder** color schema. Customize colors for all editor element according to your preferences.



### 9.1.4.6 BLOB Viewer

Below you will find a detailed decryption of the following [BLOB Viewer](#)<sup>[32]</sup> options.



#### Initial tab

Specifies which tab of **BLOB Viewer** should be active when it is initially opened.

#### 9.1.4.7 Data Export

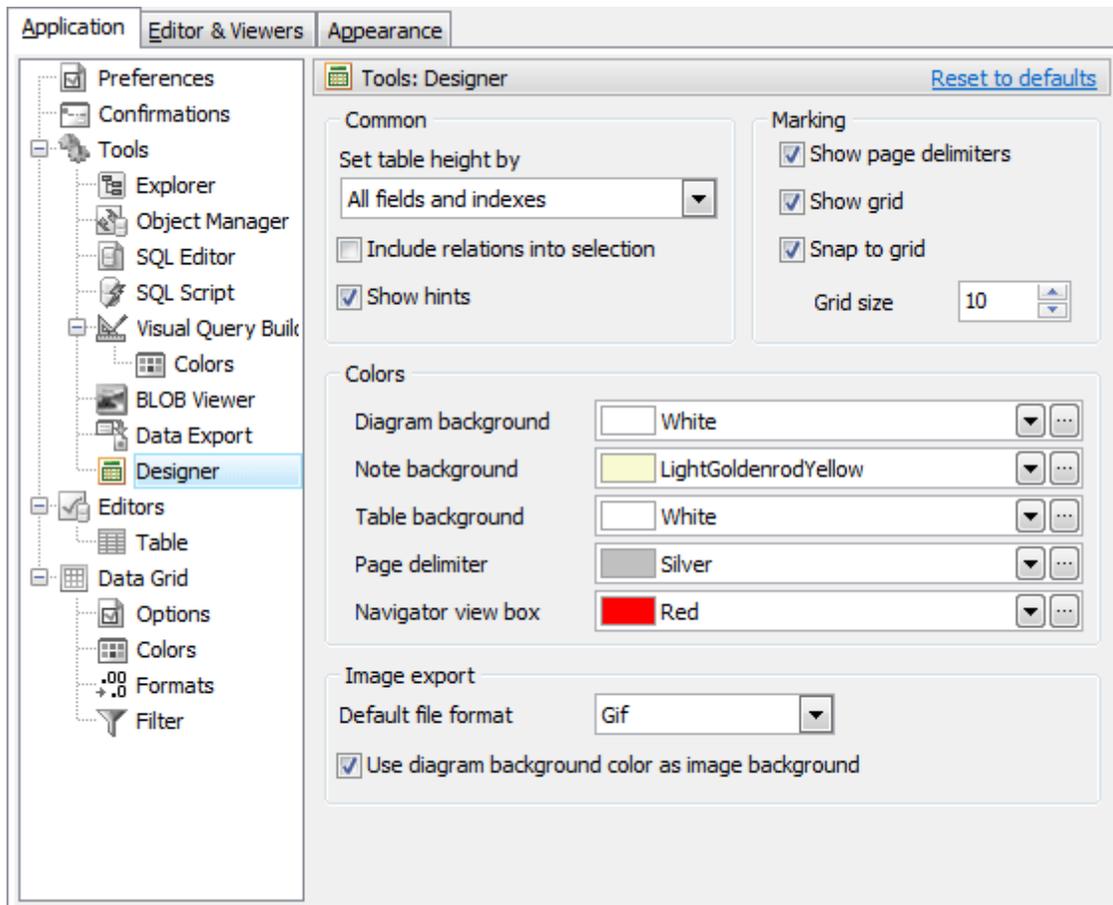
This window allows you to customize formats applied to exported data. Edit the format masks to adjust the result format in the way you need.

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:

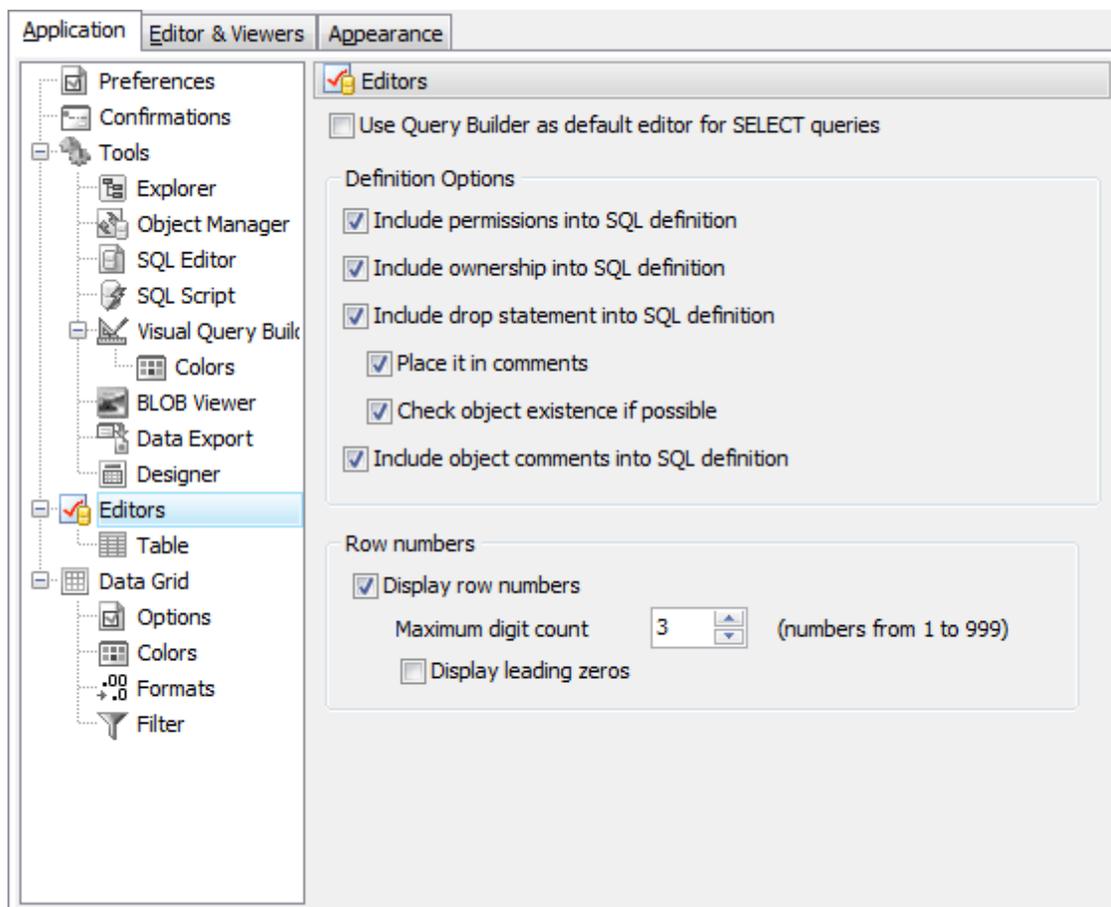
dd	day of the month, represented by 1 or 2 symbols. For example, the first day of month is 1
DD	day of the month, represented only by 2 symbols. For example, the first day of month is 01
mm	minutes





### 9.1.5 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>[269]</sup> instead of [SQL Editor](#)<sup>[264]</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' you numbering column will be of the form: '001, 002, 003, ...'.

### 9.1.5.1 Table

#### Initial tab

Specifies which tab of [Table Editor](#)<sup>[69]</sup> should be active when it is initially opened.

[Retrieve record count before loading data in the data grid](#)

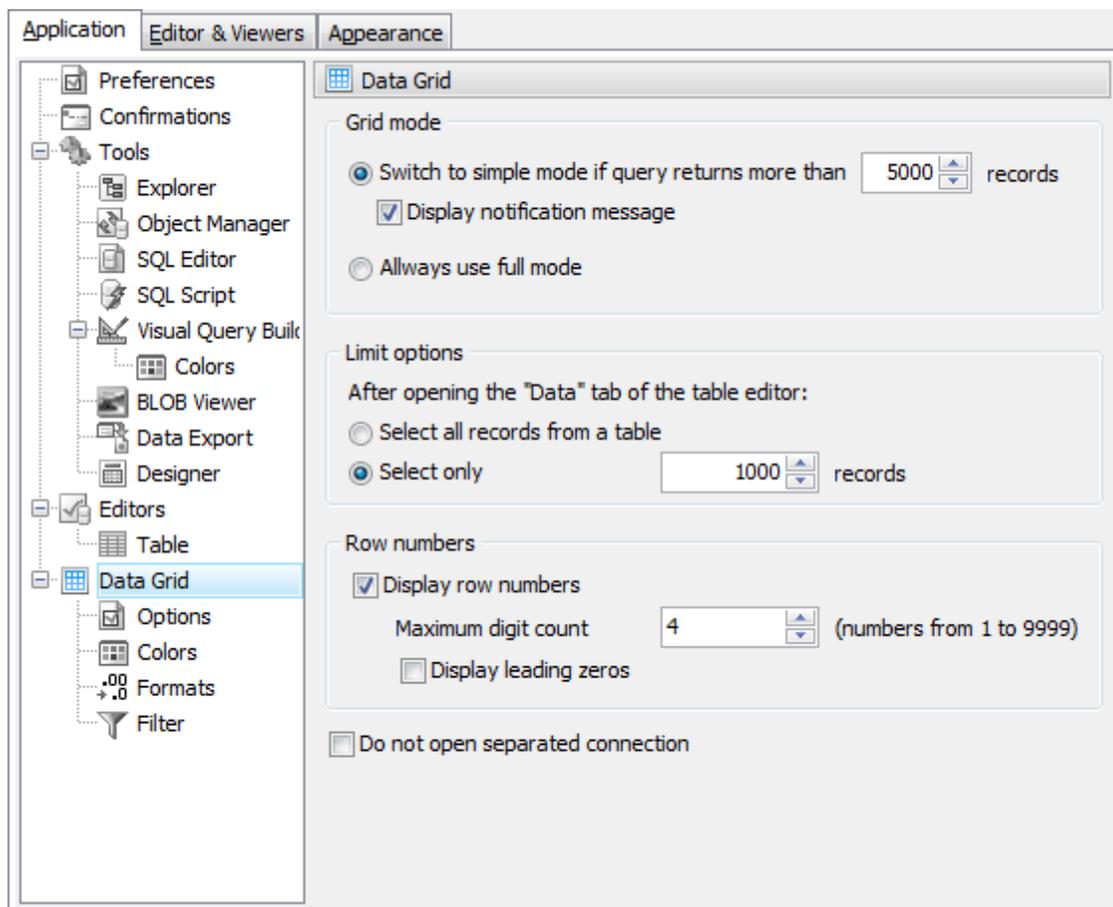
With this option enabled all the SELECT COUNT(\*)... query is executed before loading data in the data grid.

#### Default field type

Specifies the field type appearing in [Field Editor](#)<sup>[70]</sup> by default.

### 9.1.6 Data Grid

Below you will find a detailed description of the following data grid options.



Oracle Maestro provides you with [two grid modes](#)<sup>[280]</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).

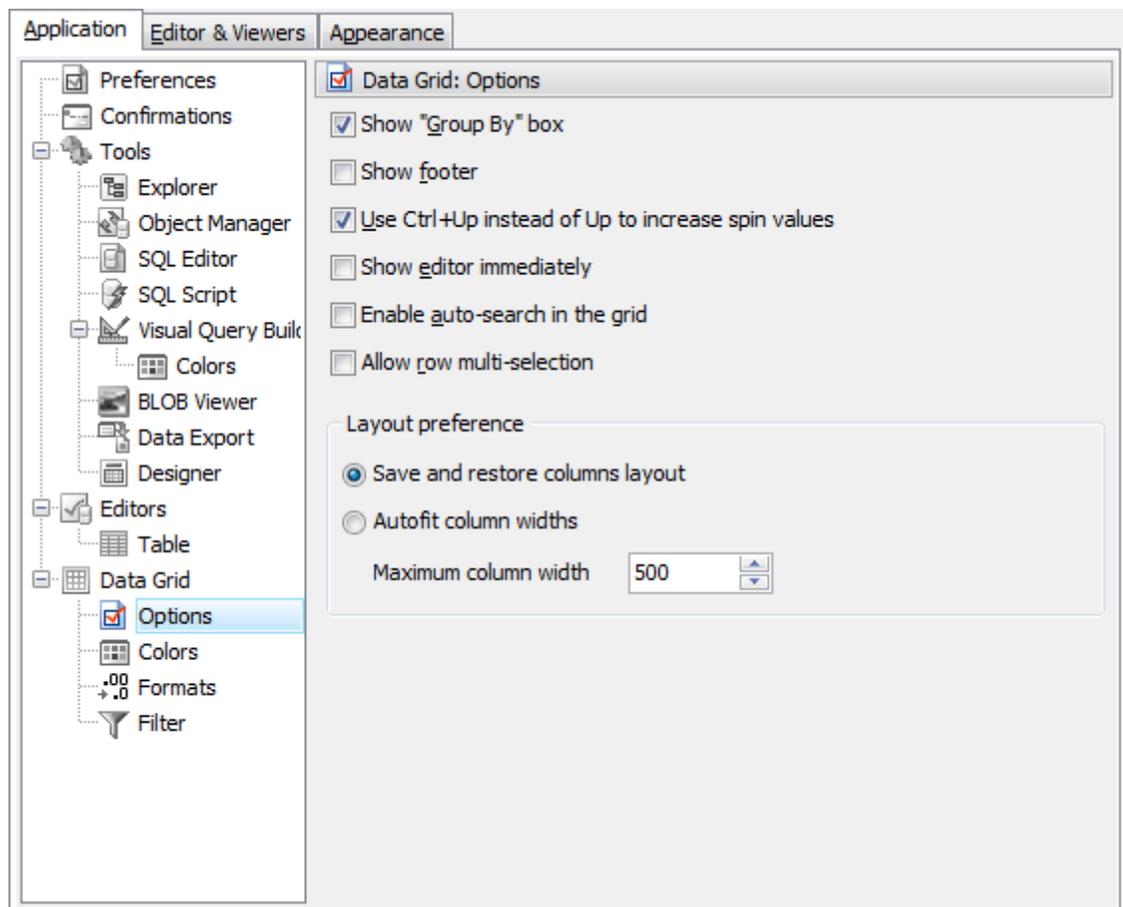
For uniformity you can use the [Display leading zeros](#) option. With this option enabled and maximum digit count '3' you numbering column will be of the form: '001, 002, 003, ...'.

#### [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.

### 9.1.6.1 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

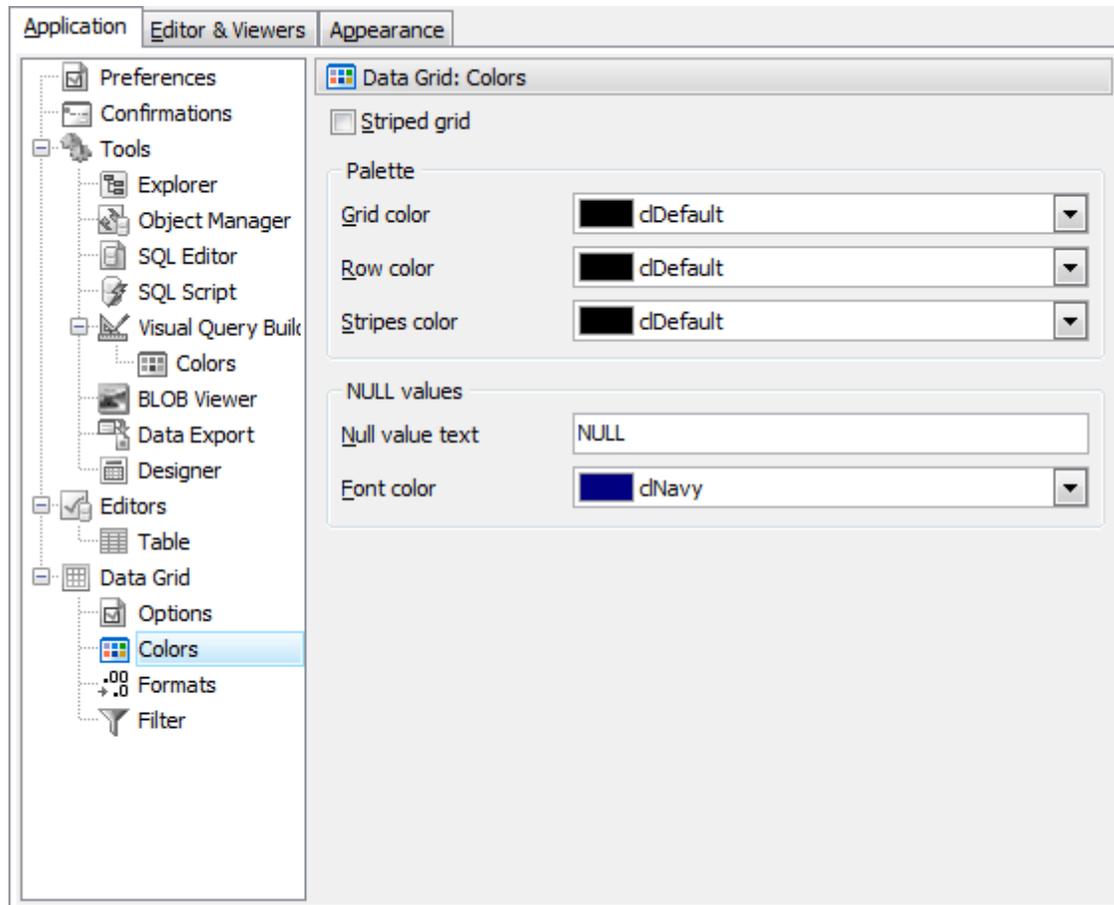
Allows you to select multiple records using the Ctrl and Shift keys.

**Layout preference**

Select whether Oracle Maestro should remember the column positions for the grids or fit them automatically.

### 9.1.6.2 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**

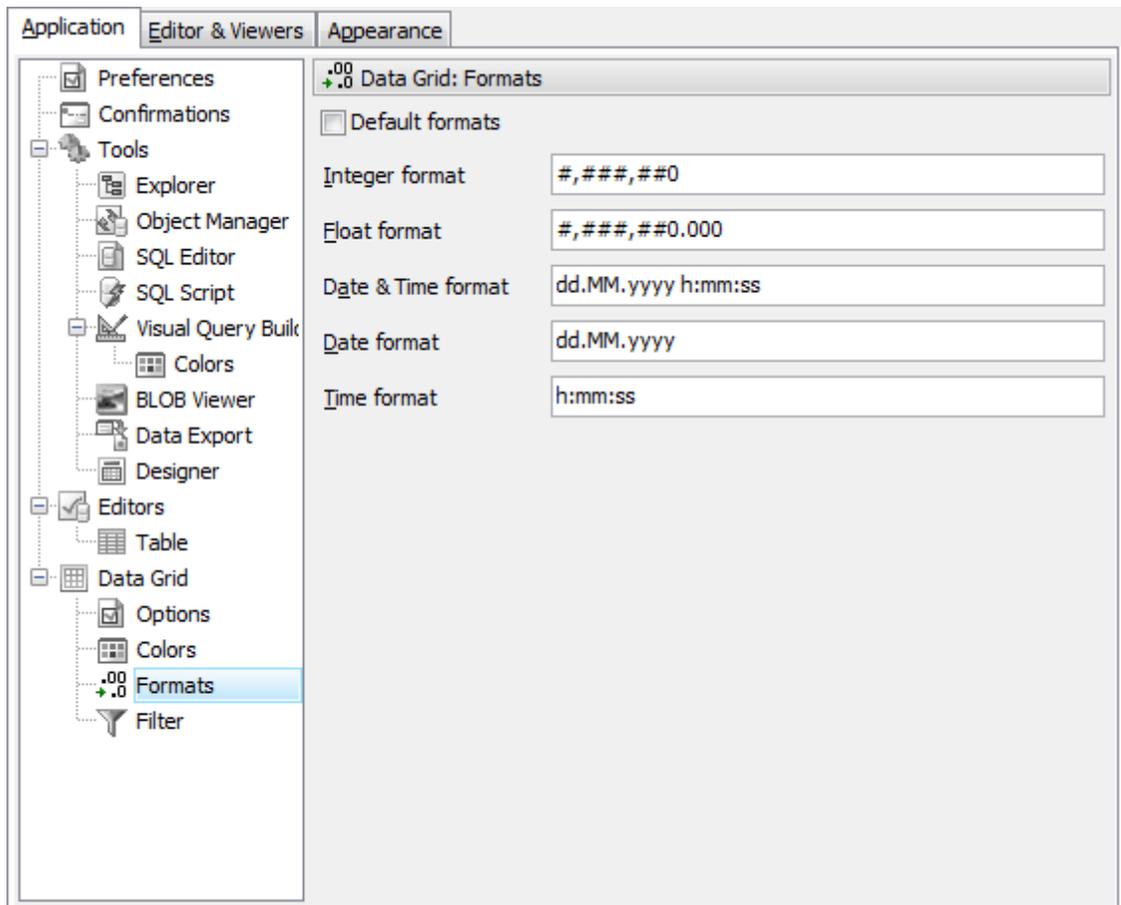
Defines the color of the odd rows if the [Striped Grid](#) option is on.

**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.

### 9.1.6.3 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).

## 9.2 Editors & Viewers

The [Editors & Viewers](#) section allows you to set the parameters of viewing and editing the SQL statements within Oracle Maestro.

- [General](#) <sup>[390]</sup>
- [Display](#) <sup>[391]</sup>
- [SQL highlight](#) <sup>[392]</sup>
- [PHP highlight](#) <sup>[394]</sup>
- [XML highlight](#) <sup>[393]</sup>
- [Code Insight](#) <sup>[395]</sup>
- [Code Folding](#) <sup>[396]</sup>

**See also:** [SQL Editor](#) <sup>[264]</sup>, [SQL Script Editor](#) <sup>[316]</sup>, [Visual Query Builder](#) <sup>[269]</sup>, [Table Editor](#) <sup>[69]</sup>.

### 9.2.1 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 editor window.

[Show special chars](#)

If this option is checked, special chars (like line breaks) are displayed in the editor window.

[Use smart tabs](#)

With this option on the number of tab stops is calculated automatically, depending on the previous line tab.

[Convert tabs to spaces](#)

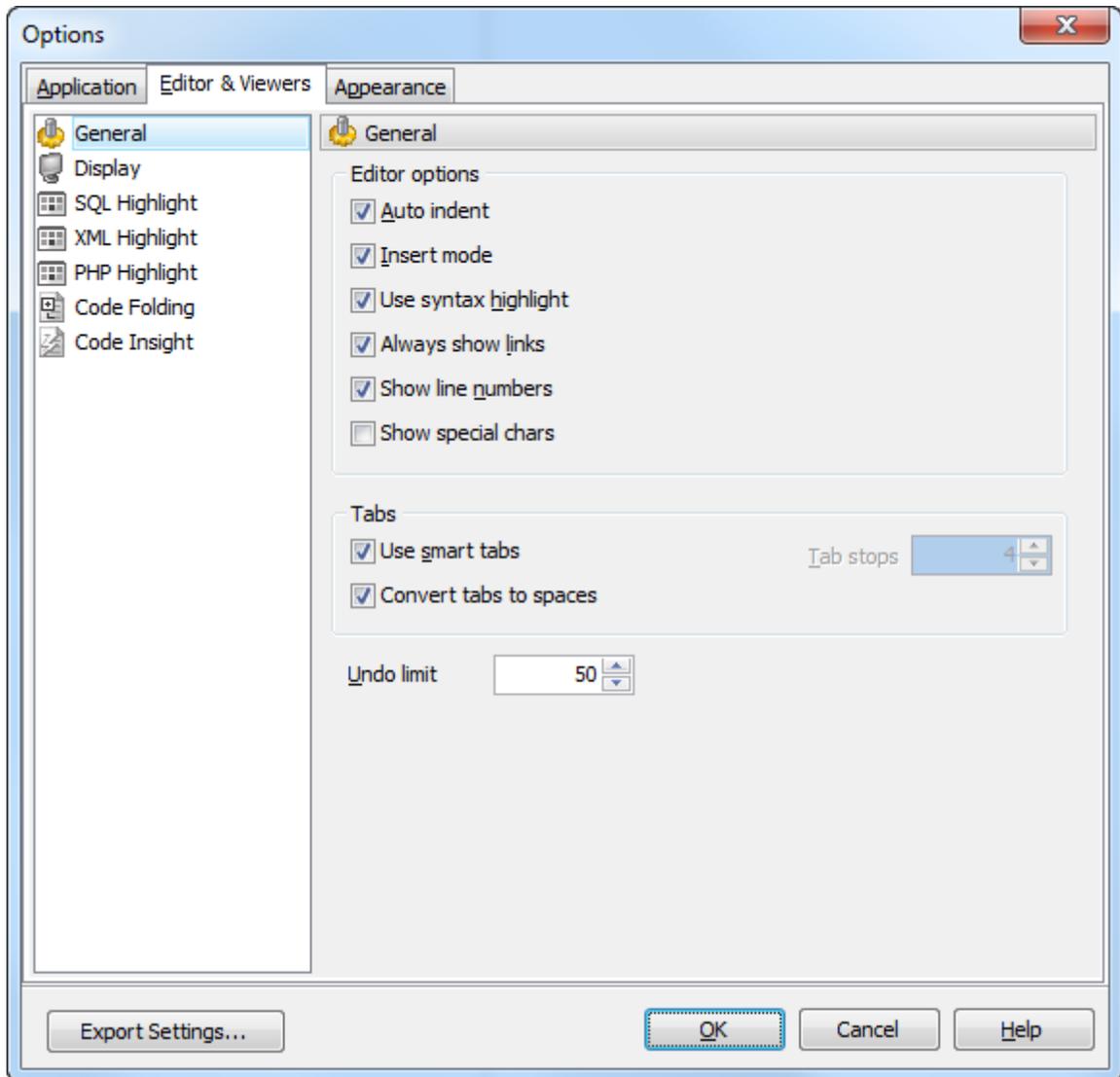
If this option is checked, each time you press the Tab key, the appropriate number of spaces will be added to the edited text.

[Tab Stops](#)

Defines the tab length, used when editing text.

### Undo Limit

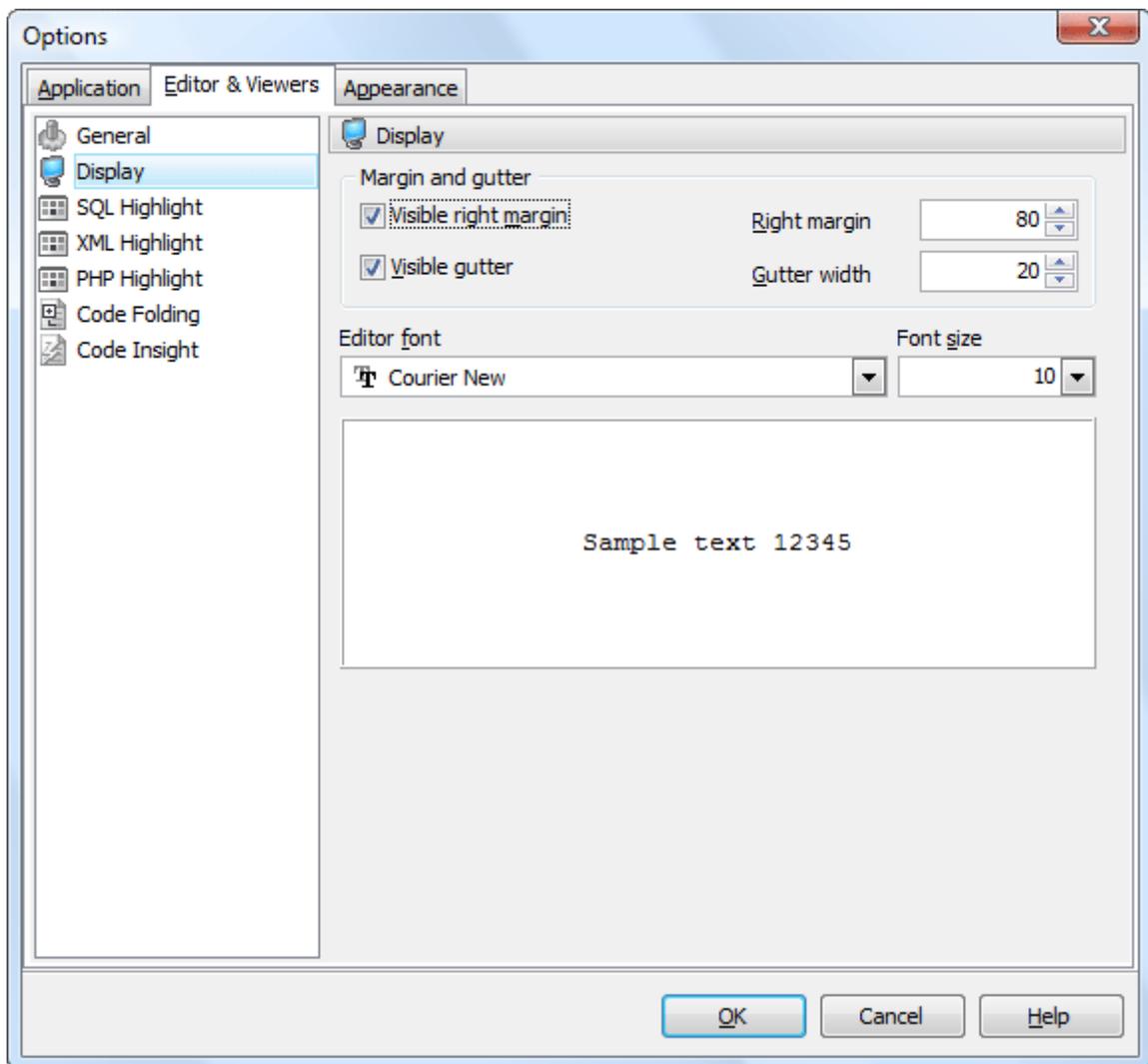
Defines the maximum number of changes possible to be undone.



## 9.2.2 Display

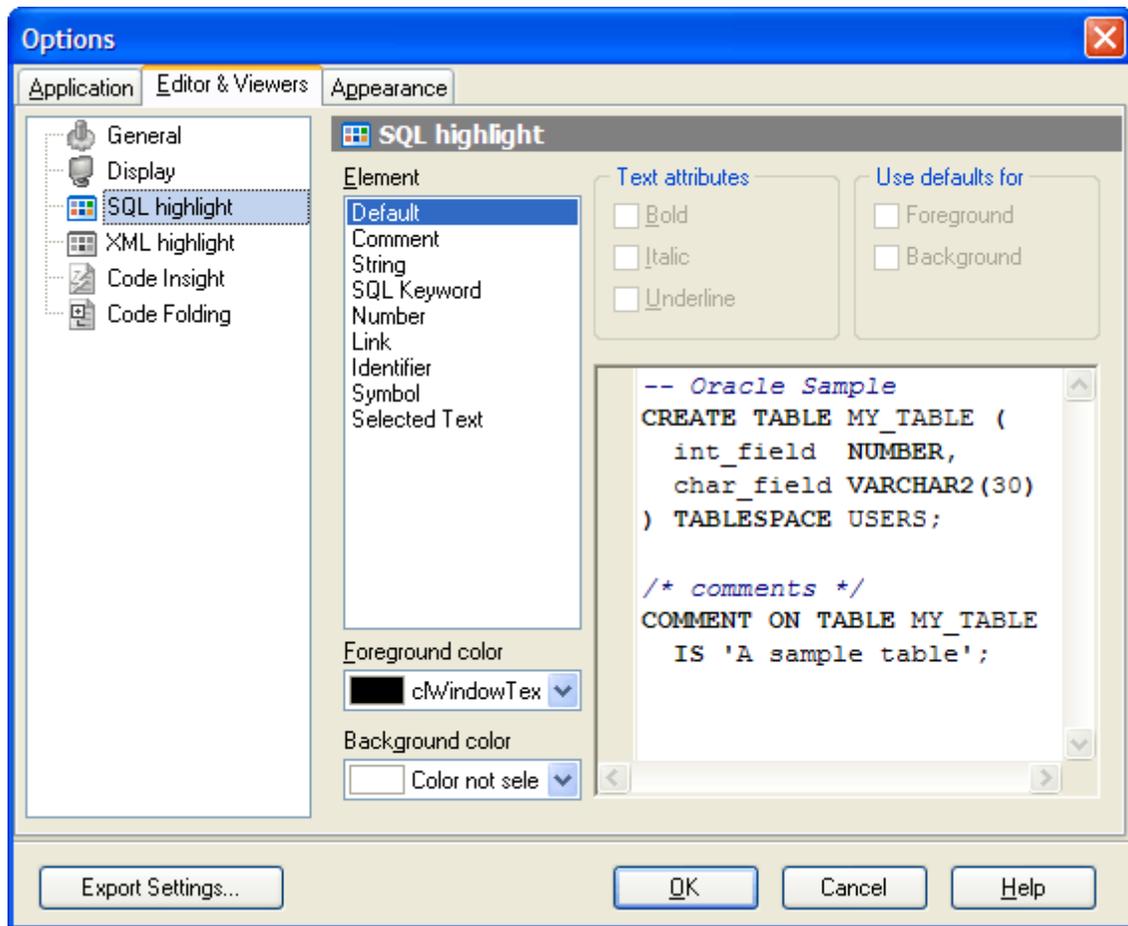
You can disable/enable the right text margin and the gutter of the editor area, set the position of the right text margin as [Right margin](#), and the [Gutter width](#).

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.



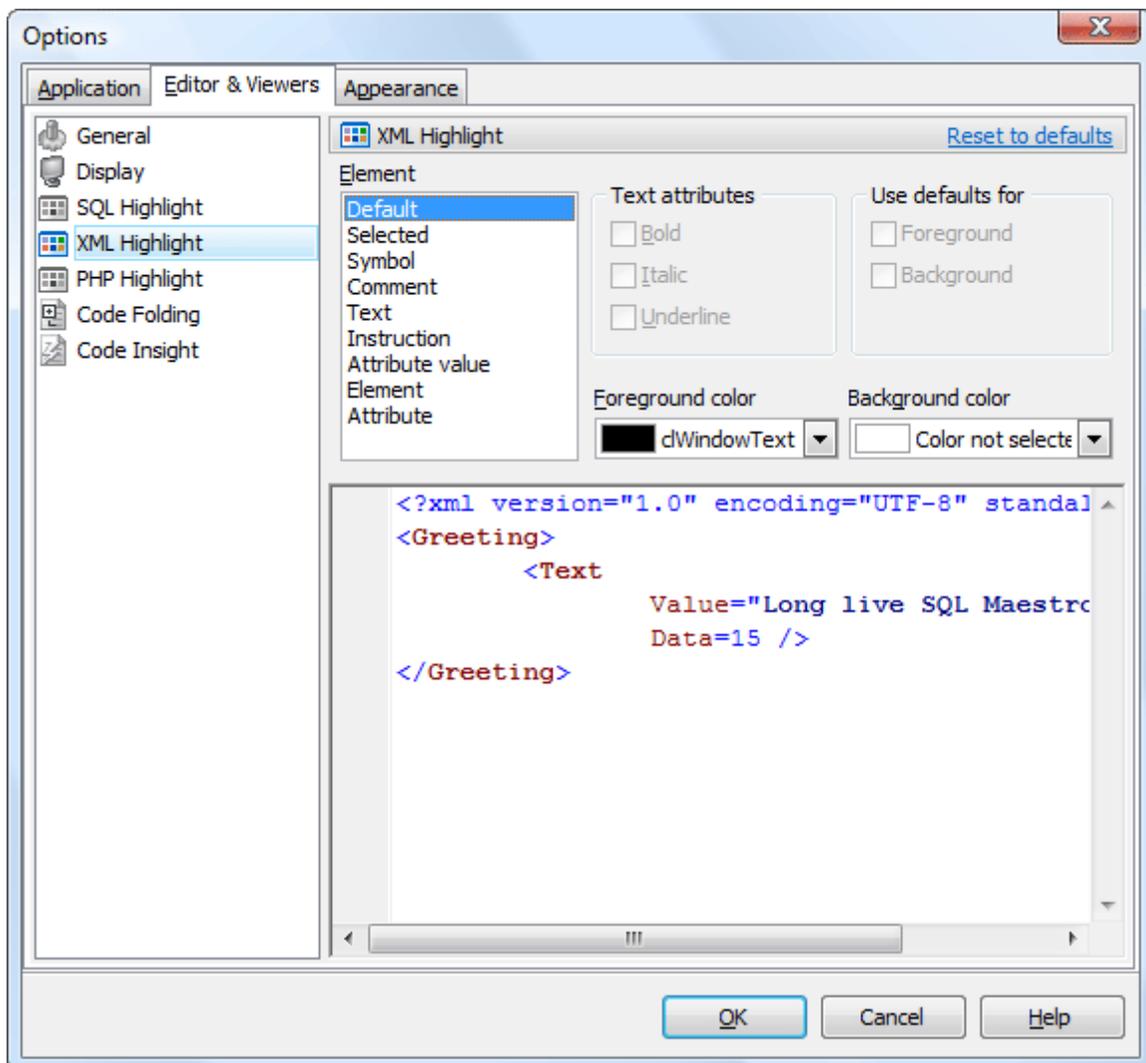
### 9.2.3 SQL highlight

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.



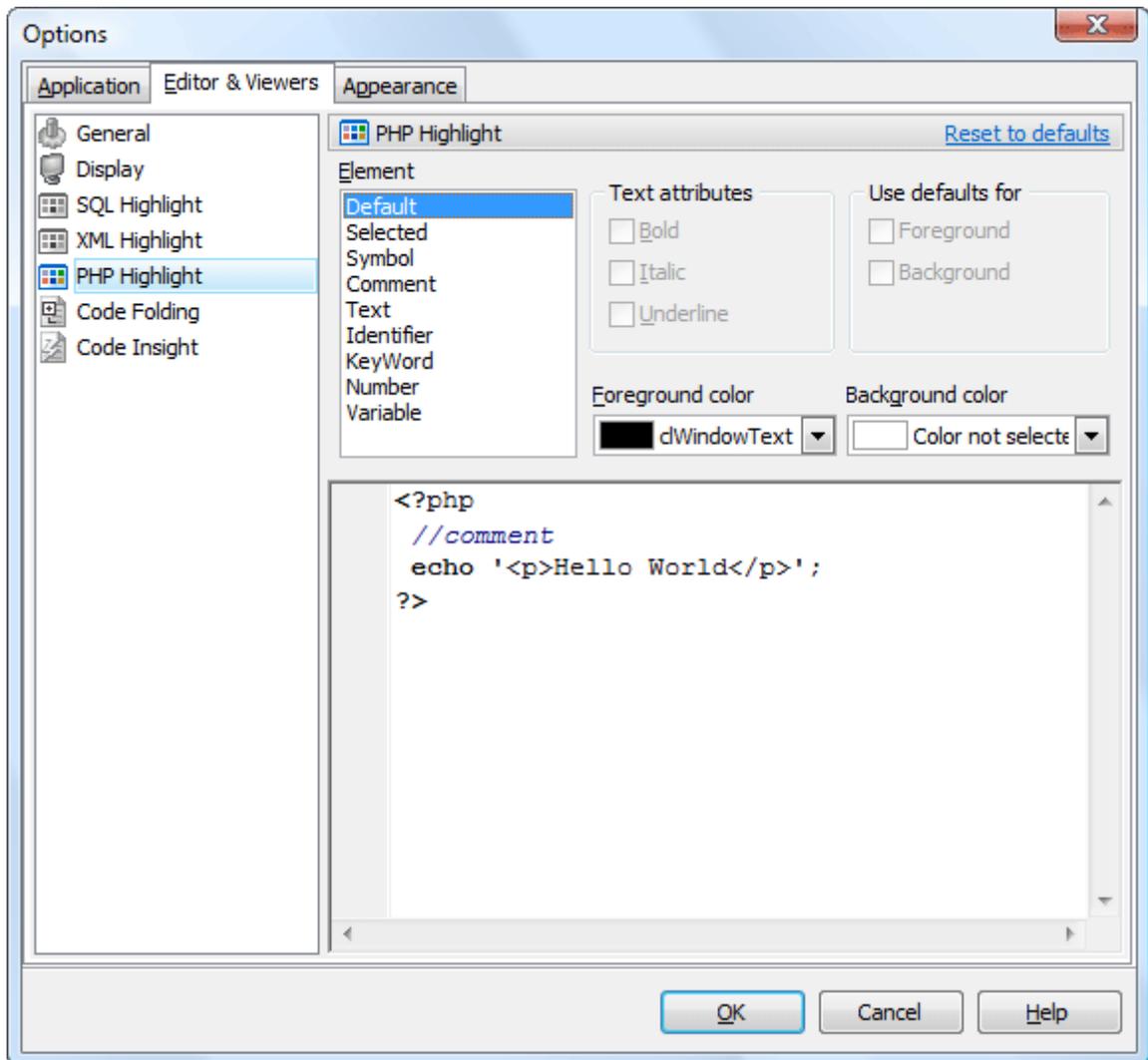
### 9.2.4 XML highlight

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.



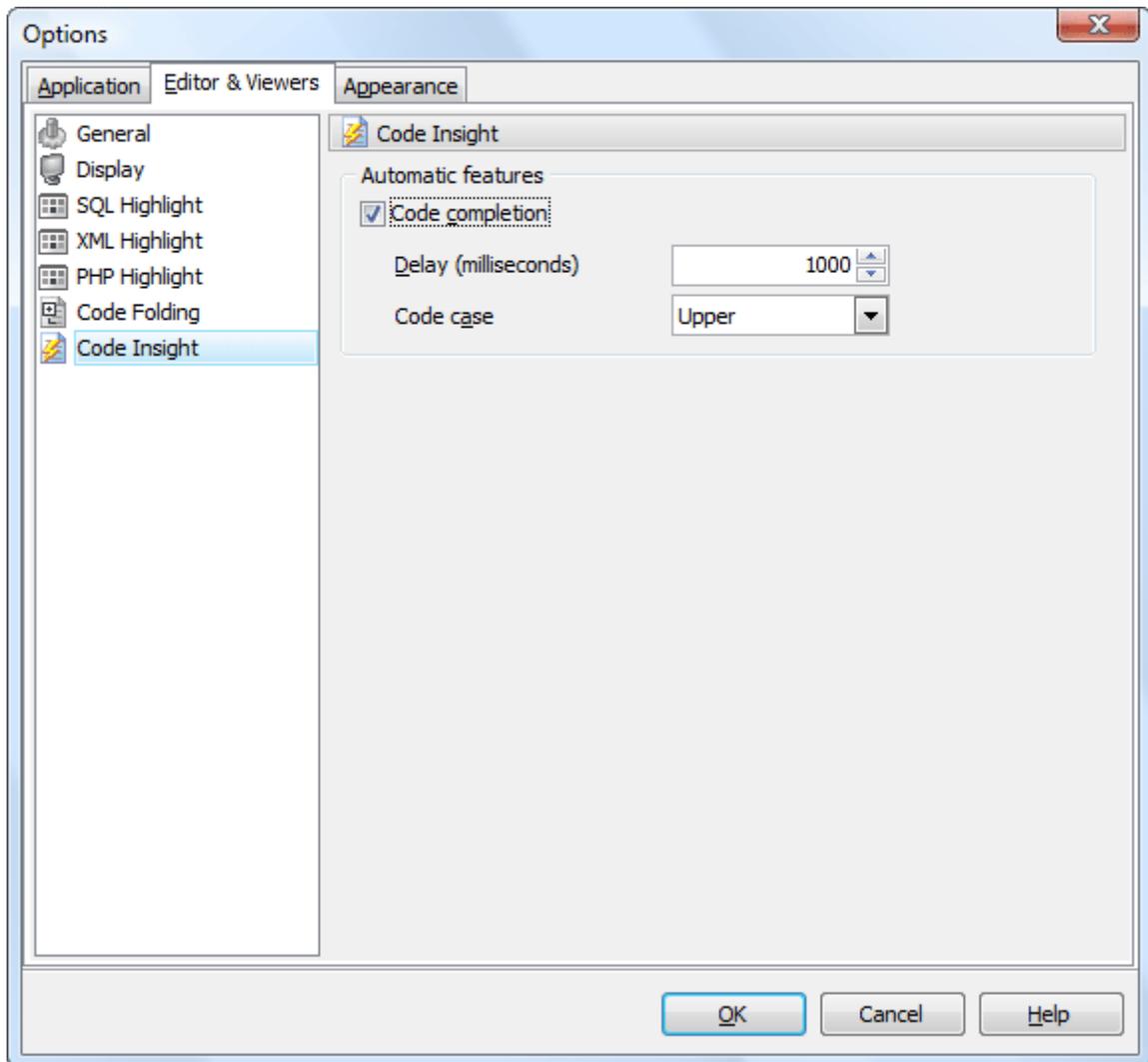
## 9.2.5 PHP highlight

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.



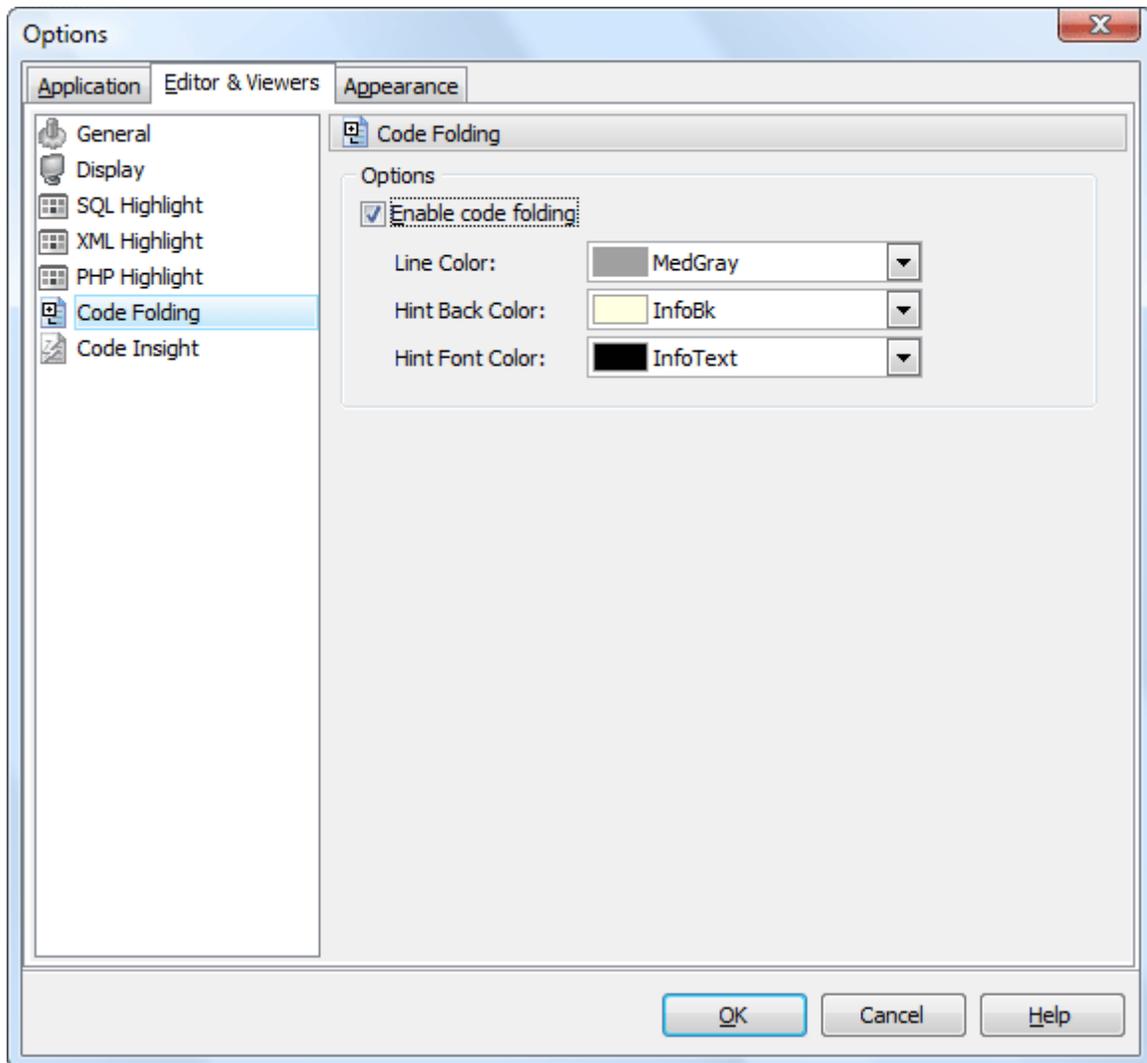
## 9.2.6 Code Insight

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.



## 9.2.7 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.

## 9.3 Appearance

The [Appearance](#) section allows you to customize the application interface style to your preferences.

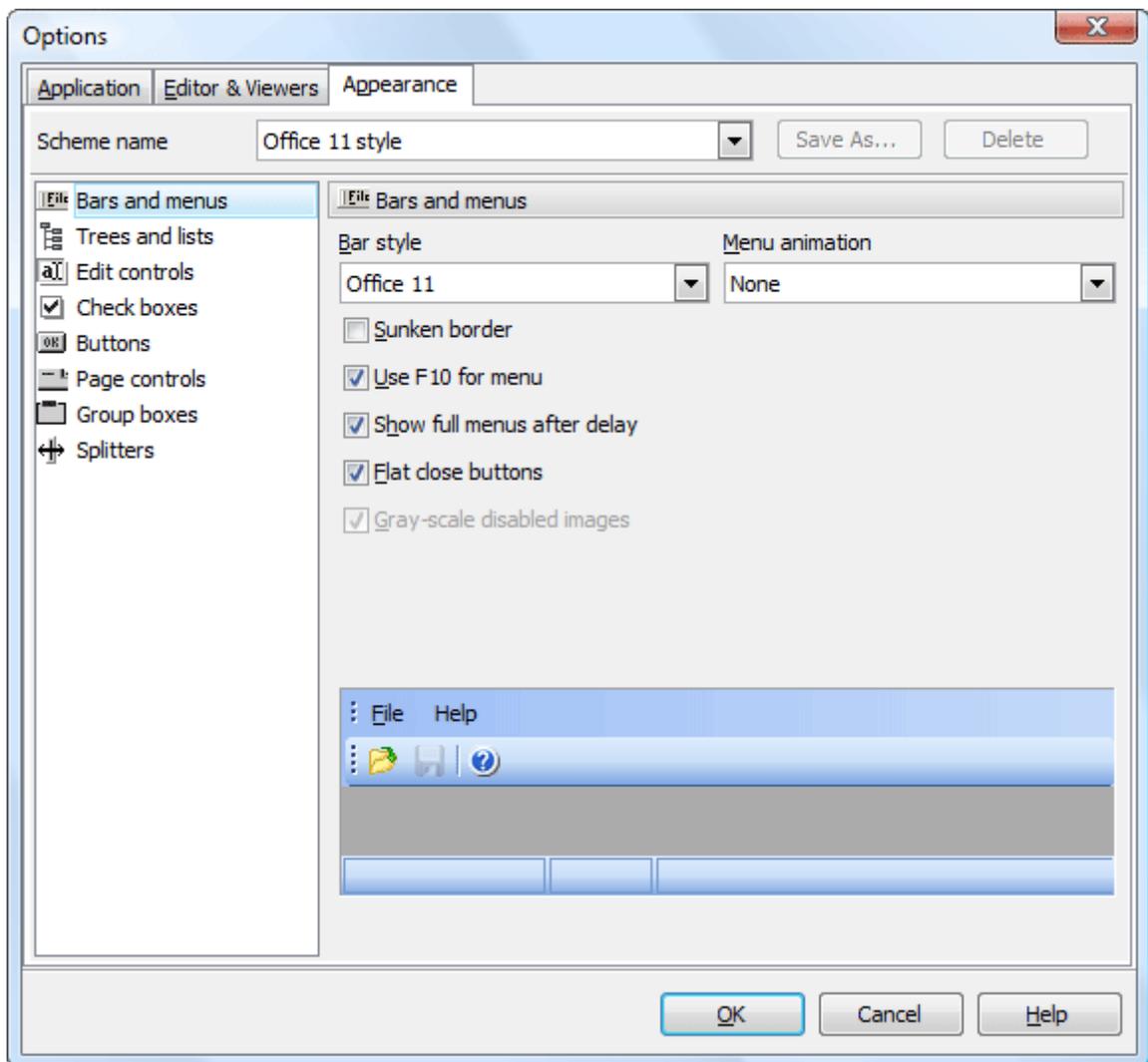
Use the [Scheme name](#) box to select the interface scheme you prefer: *Office XP style*, *Windows XP native style*, etc. You can create your own interface schemes by customizing any visual options ([Bars and menus](#), [Trees and lists](#), [Edit controls](#), [Check boxes](#), [Buttons](#), etc.) and clicking the [Save As](#) button. All the customized options are displayed on the sample panel.

- [Bars and menus](#) <sup>[398]</sup>
- [Trees and lists](#) <sup>[399]</sup>
- [Edit controls](#) <sup>[400]</sup>
- [Check boxes](#) <sup>[401]</sup>
- [Buttons](#) <sup>[402]</sup>
- [Page controls](#) <sup>[403]</sup>
- [Group boxes](#) <sup>[404]</sup>
- [Splitters](#) <sup>[405]</sup>

### 9.3.1 Bars and menus

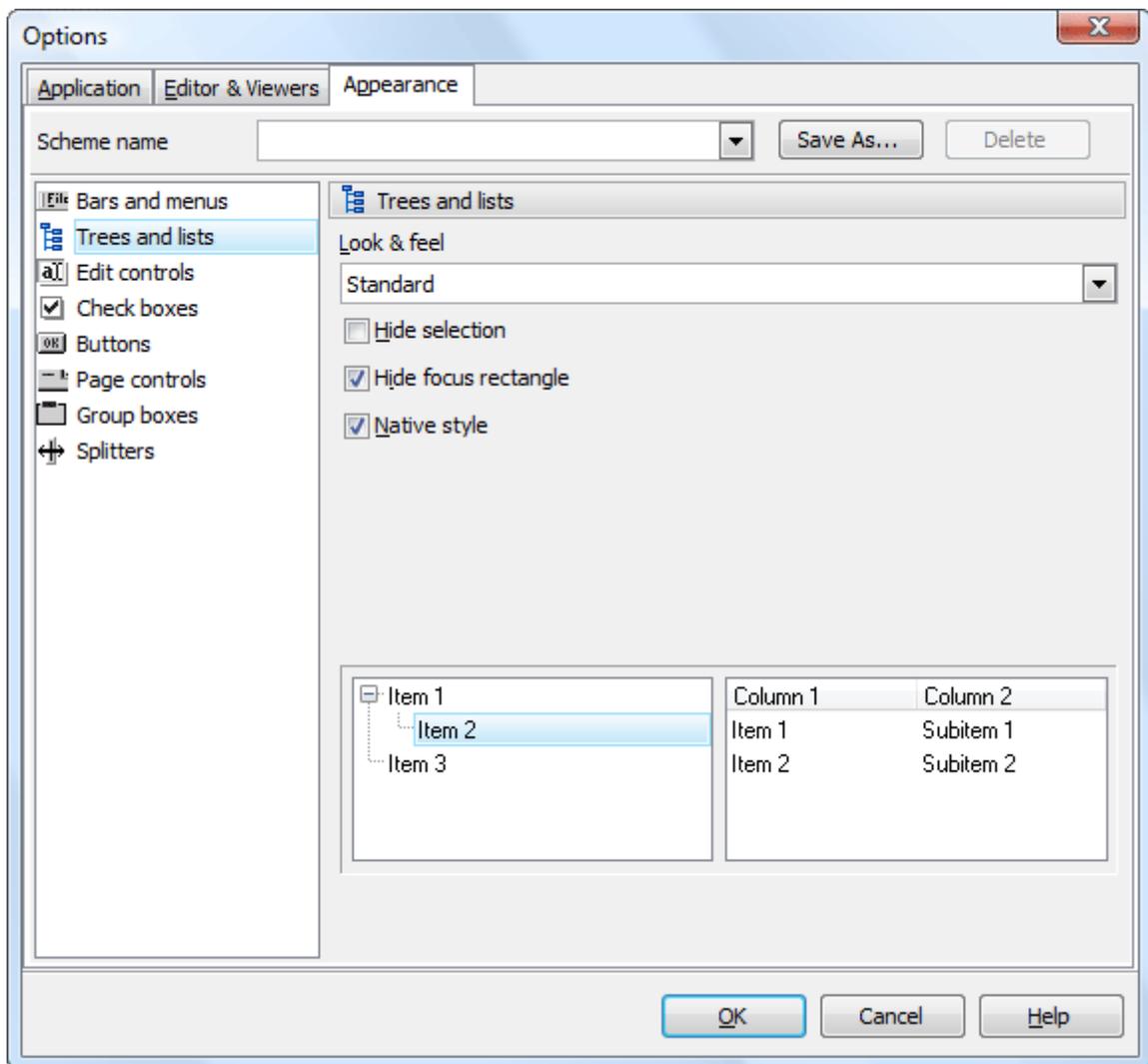
Use the [Bars and menus](#) item to customize Oracle Maestro toolbars style and menu animation.

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.



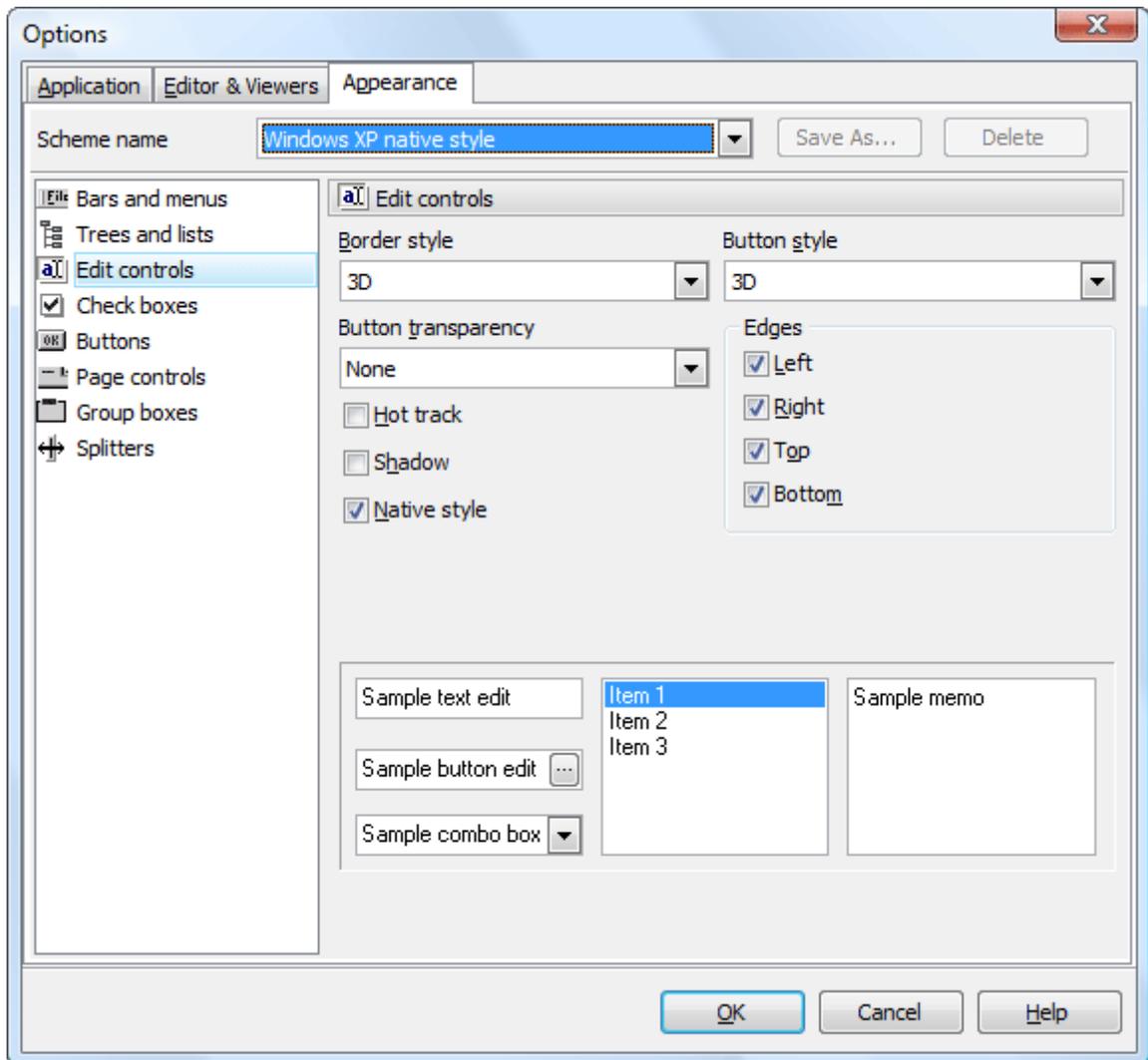
### 9.3.2 Trees and lists

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.



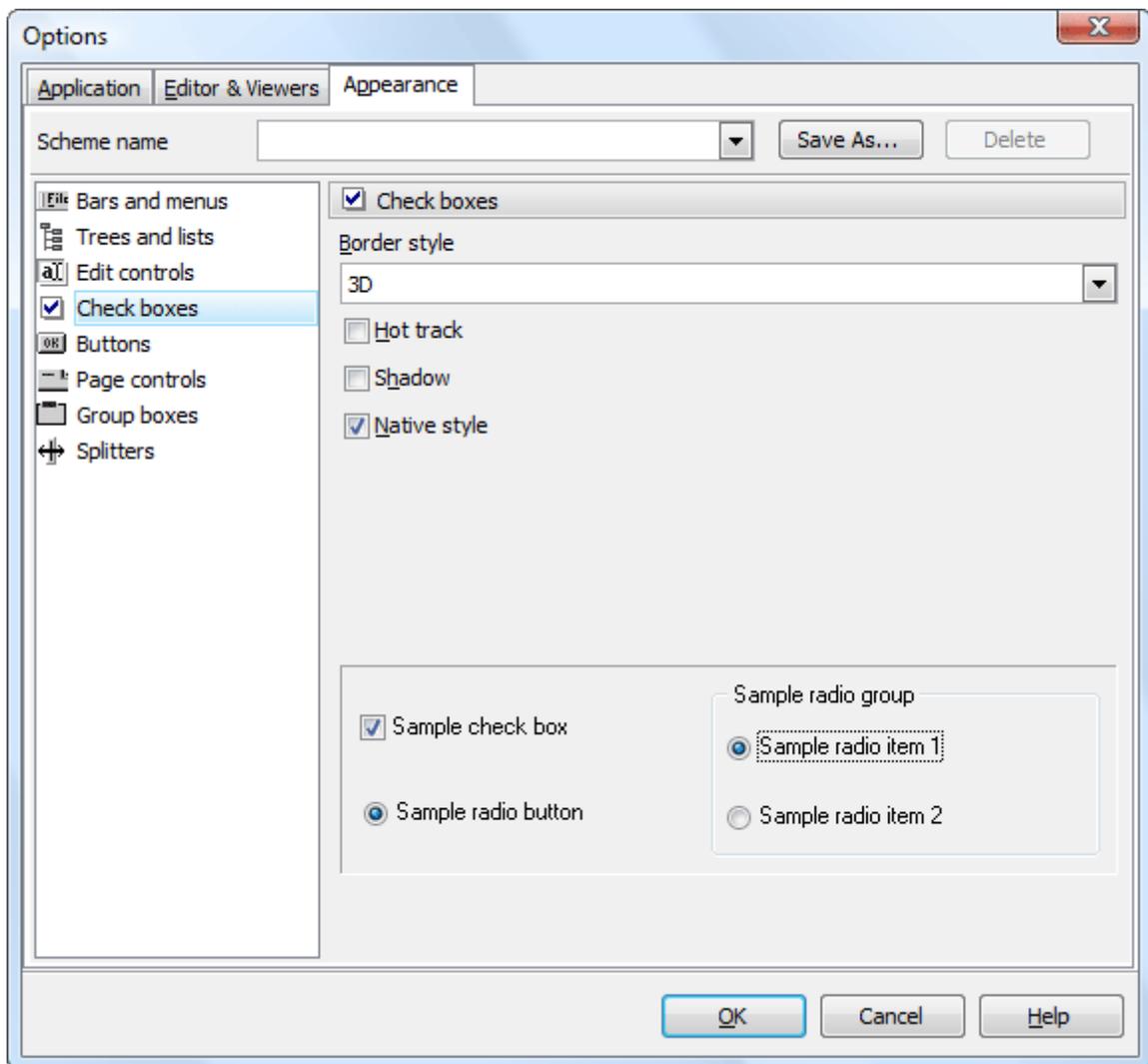
### 9.3.3 Edit controls

Use the [Edit controls](#) item to customize the appearance of different Oracle 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.



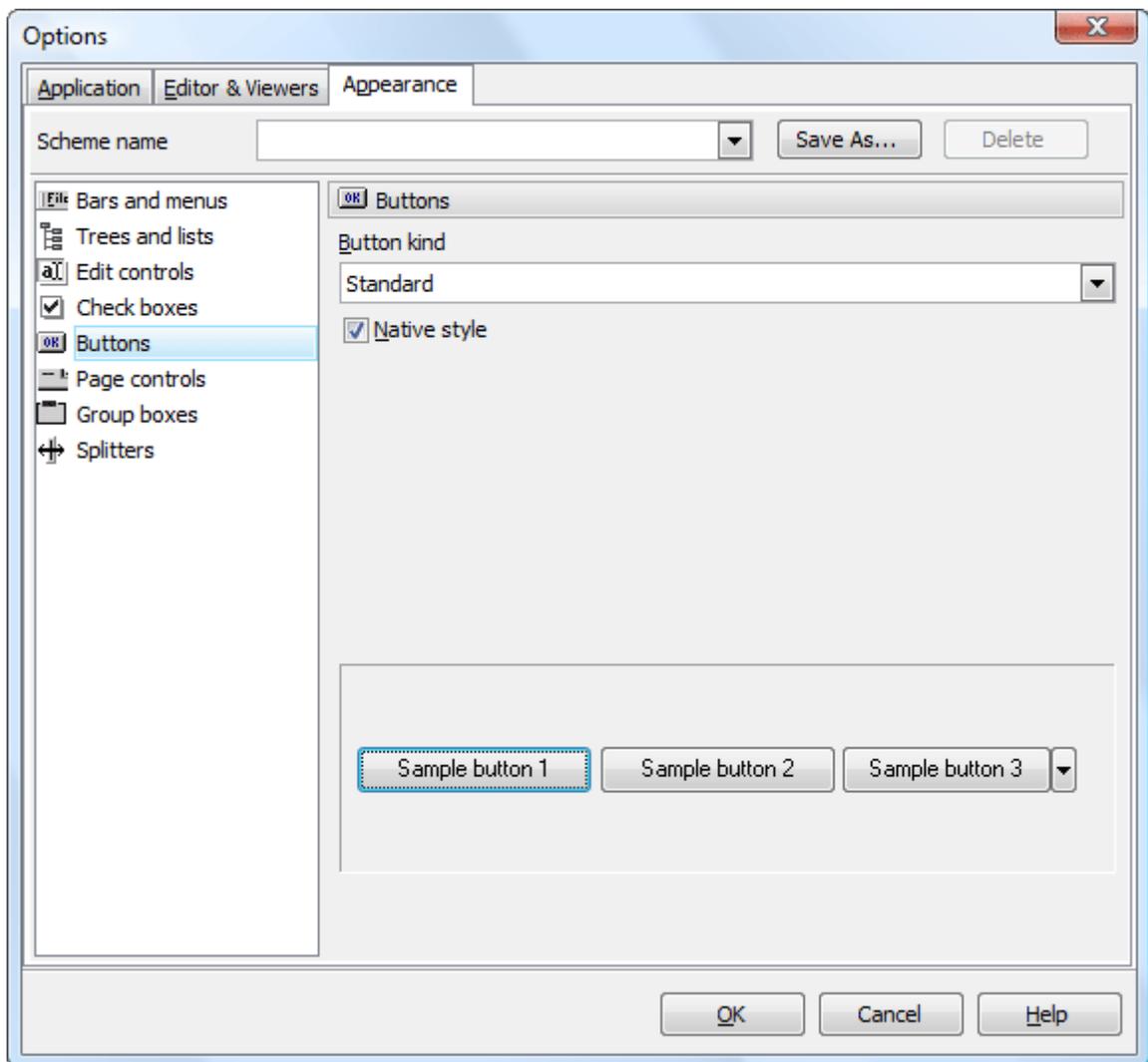
### 9.3.4 Check boxes

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.



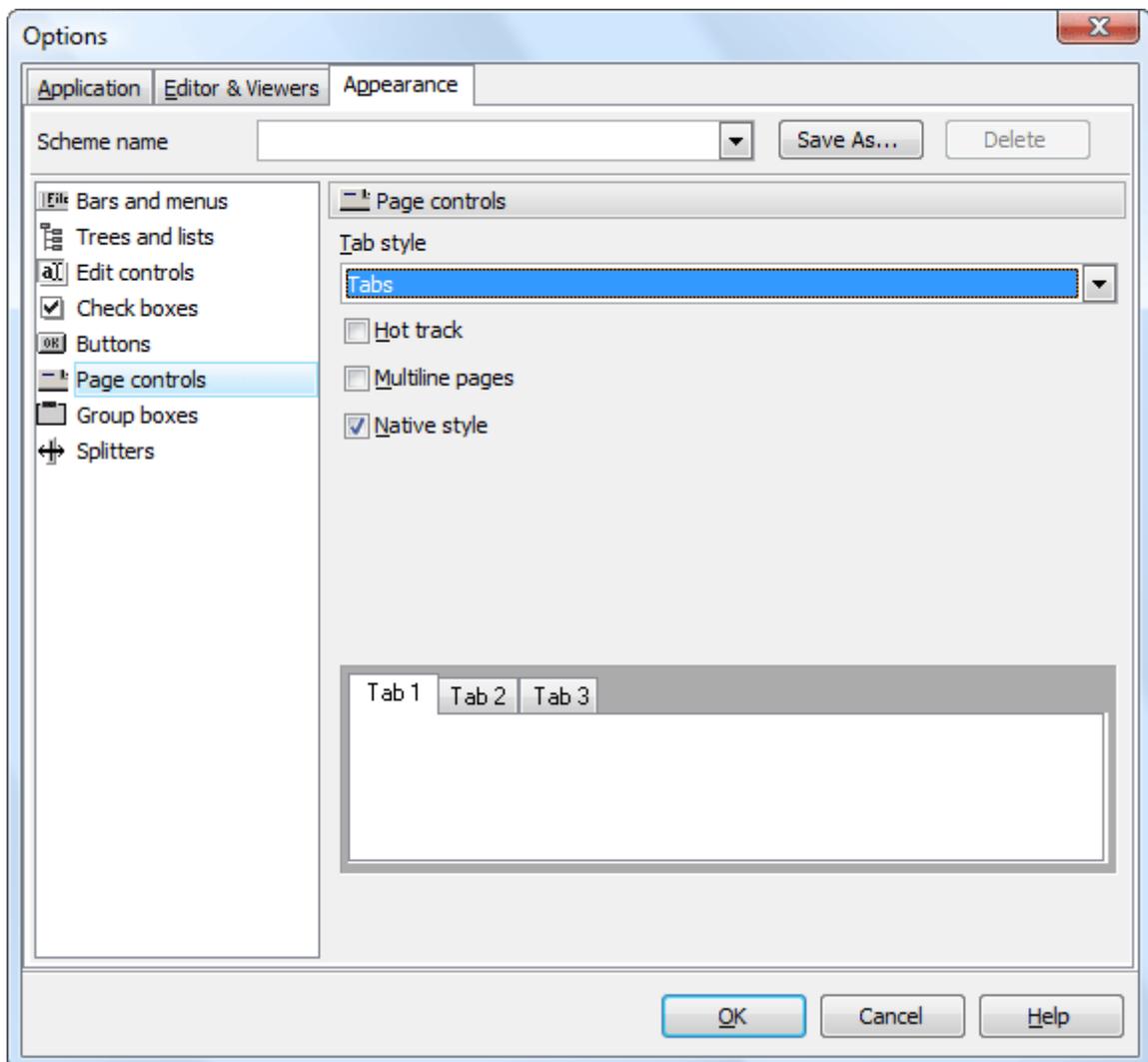
### 9.3.5 Buttons

Use the [Buttons](#) item to customize Oracle Maestro buttons. The tab allows you to adjust the appearance of buttons and define sample buttons as well.



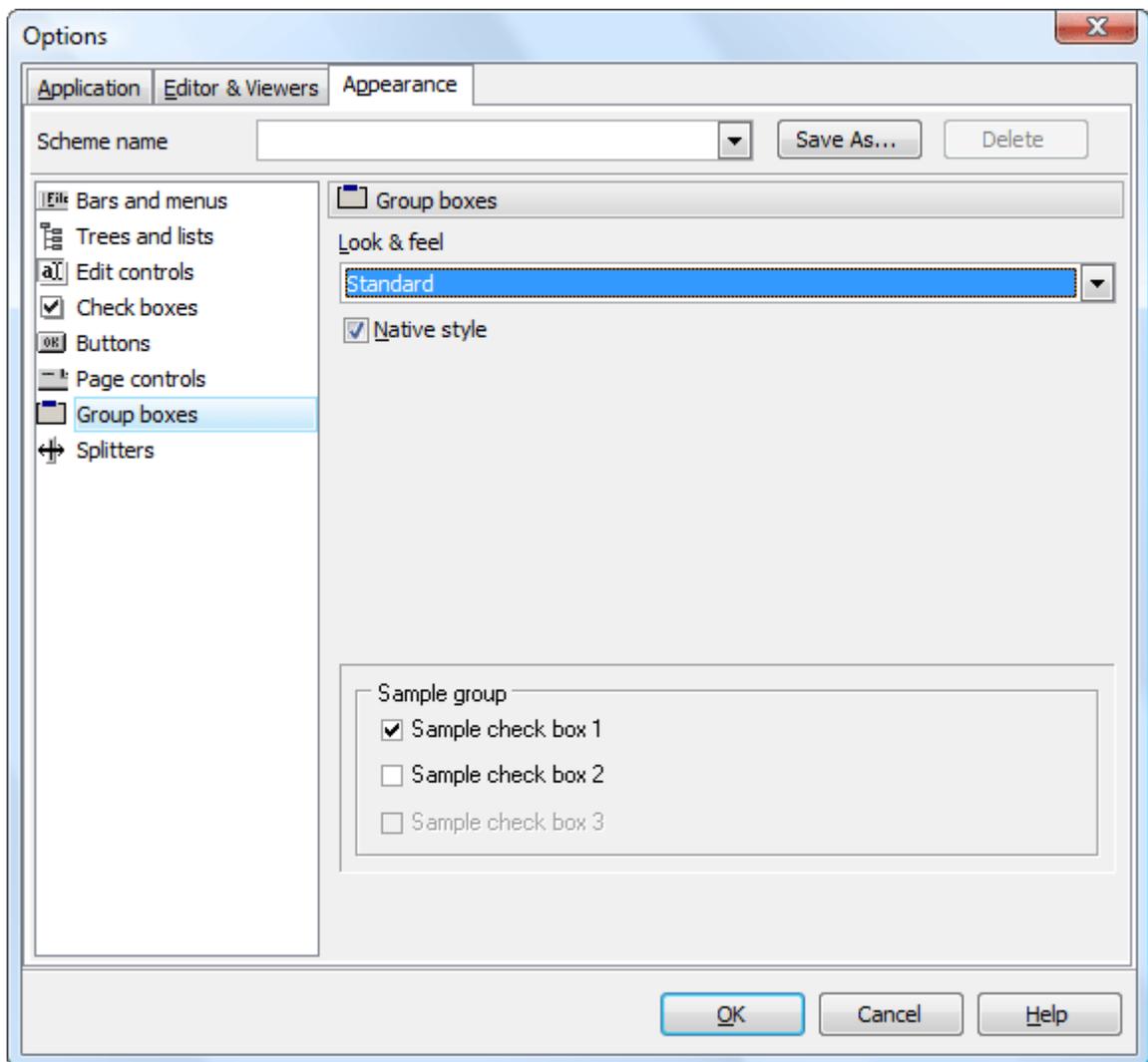
### 9.3.6 Page controls

The [Page controls](#) item allows you to customize the style of all Oracle Maestro page controls. The tab allows you to select tab styles, enable/disable hot track, multi-line pages and native style.



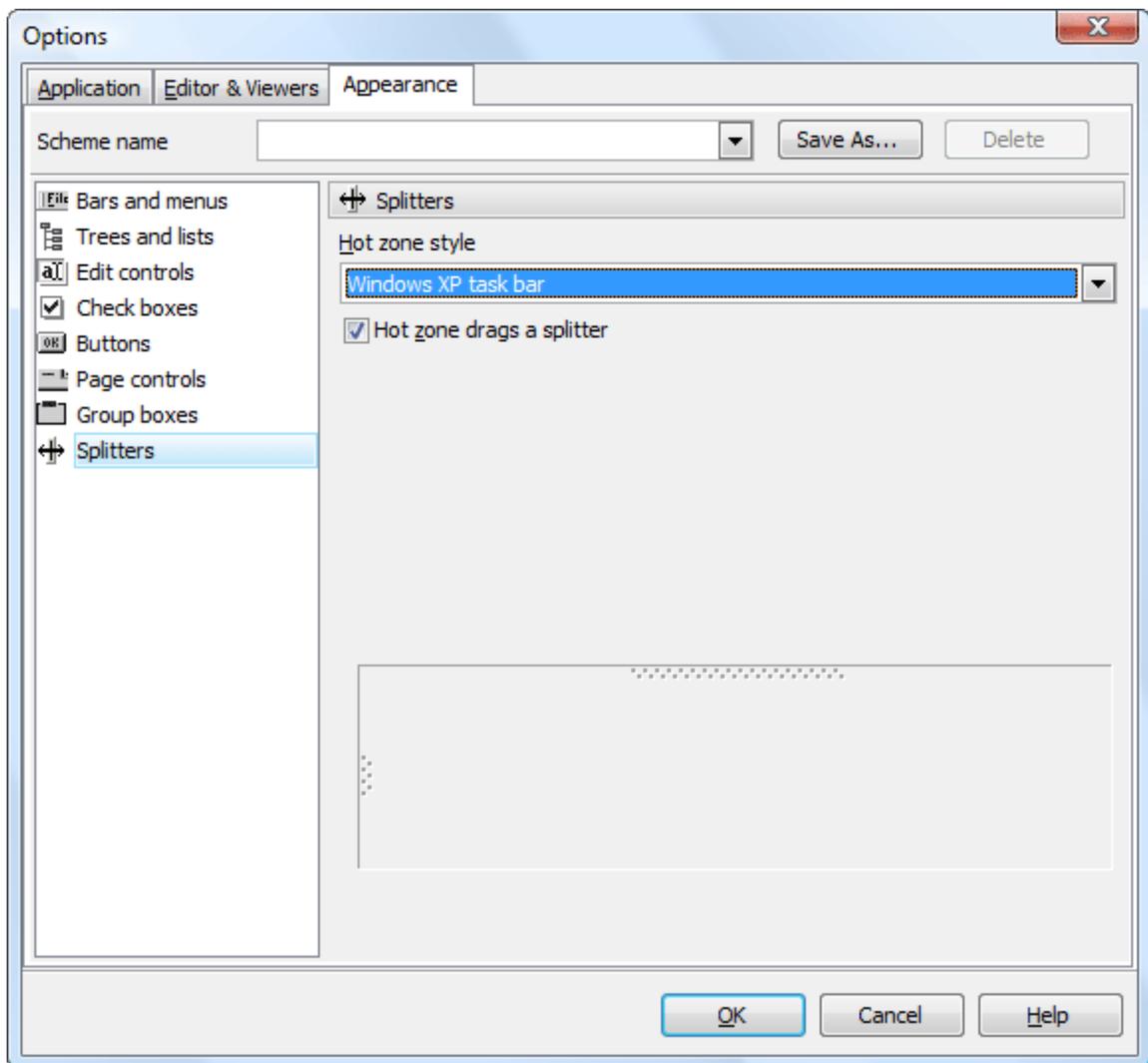
### 9.3.7 Group boxes

Use the [Group boxes](#) item to customize all Oracle Maestro group boxes according to your preferences. Use tab to apply styles for group boxes, enable/disable native style and define samples.



### 9.3.8 Splitters

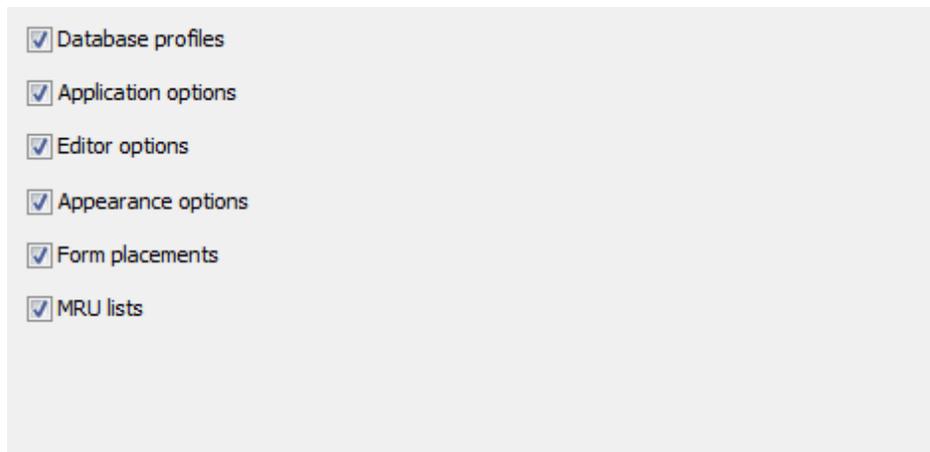
Use the [Splitters](#) item to customize all Oracle Maestro splitters according to your preferences. Use the tab to select hot zone style (*Windows XP task bar, Media Player 8, Media Player 9, Simple or none*) and specify the [Hot zone drags a splitter](#) option.



## 9.4 Export Settings

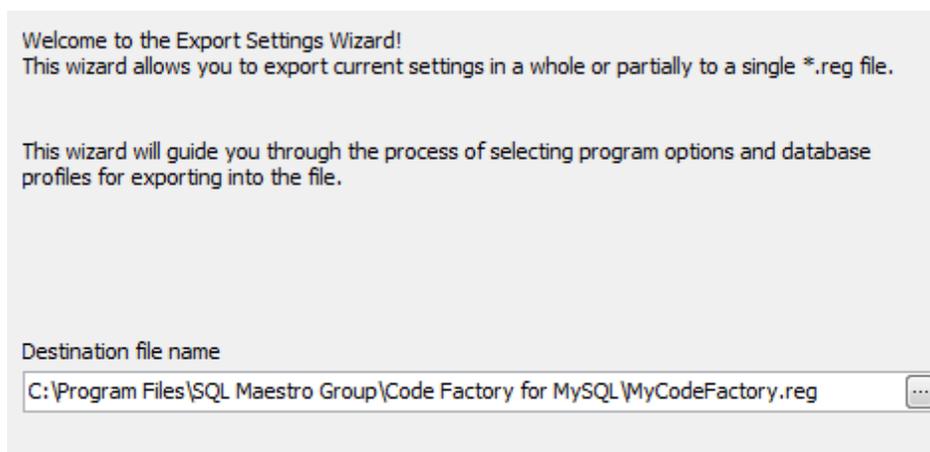
Export Settings Wizard allows you to export all or partial Oracle Maestro settings to single \*.reg file which can be applied to the application of Oracle Maestro installed on another machine or used to backup previous settings. To run the wizard, select the Tools | Options main menu item and click Export Settings in the [Options](#)<sup>[367]</sup> dialog.

- [Specifying destination file to save settings to](#)<sup>[407]</sup>
- [Specifying settings categories to save](#)<sup>[407]</sup>
- [Select database profiles to save](#)<sup>[408]</sup>
- [Saving settings](#)<sup>[408]</sup>



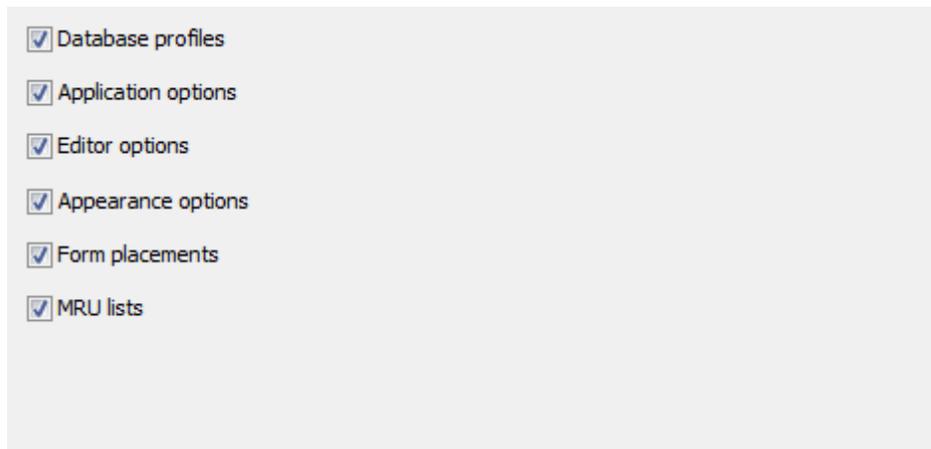
### 9.4.1 Specifying destination file

Specify a \*.reg file to extract Oracle Maestro setting to.



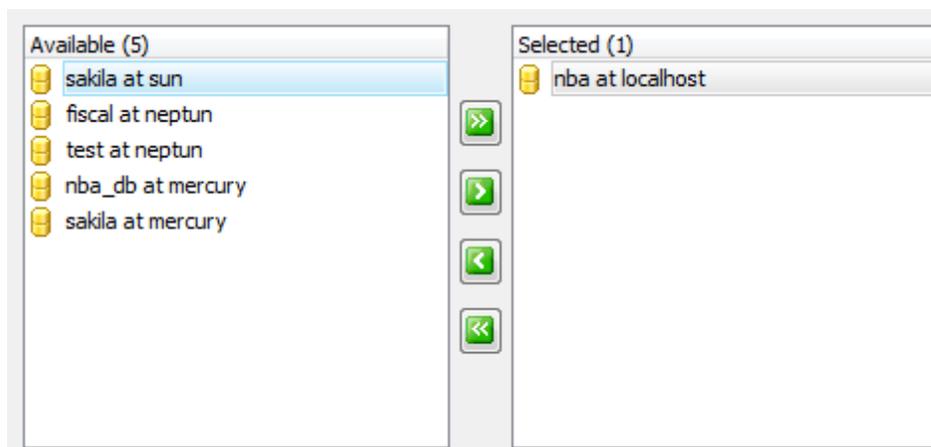
### 9.4.2 Selecting setting categories

The options of this step specify the information saved to the result file, e.g. Database profiles, [Application options](#)<sup>[368]</sup>, etc.



### 9.4.3 Selecting database profiles

Select database profiles to save their settings by moving them from the [Available Databases](#) list to the [Selected Databases](#) one.



### 9.4.4 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 -

- Appearance Options
  - Bar and menus 398
  - Buttons 402
  - Check boxes 401
  - Edit controls 400
  - Group boxes 404
  - Page controls 403
  - Splitters 405
  - Trees and lists 399

## - B -

- BLOB Editor 290
  - Hexadecimal mode 291
  - HTML mode 292
  - Image mode 290
  - Navigating with the BLOB editor 290
  - PDF mode 293
  - Plain text mode 291
- BLOB Viewer 321
  - Hexadecimal mode 321
  - HTML mode 324
  - Image mode 323
  - PDF mode 325
  - Plain text mode 322

## - C -

- Checks 85
- Cluster Editor 225
  - Editing cluster properties 226
- Clusters 221
  - Cluster Editor 225
  - Create Cluster Wizard 222
- Collection Editor 196
  - Editing collection properties 196
- Collections 193
  - Collection Editor 196
  - Create Collection Wizard 194
- Create Cluster Wizard 222
  - Adding cluster subitems 224

- Specifying cluster properties 223
- Create Collection Wizard 194
  - Setting collection properties 195
- Create Database link Wizard 143
  - Specifying database link properties 144
- Create Database Profiles Wizard 23
  - Setting connection properties 23
  - Setting profile options 23
- Create Database trigger Wizard 240
- Create Directory Wizard 234
  - Specifying directory properties 235
- Create Function Wizard 123
- Create Java source Wizard 199
  - Specifying java source definition 200
  - Specifying java source properties 200
- Create Job Wizard 205
  - Specifying job definition 206
  - Specifying job properties 205
- Create Library Wizard 229
  - Specifying library properties 229
- Create Materialized View Wizard 106
- Create object wizard - basic principles 37
  - Setting object name 38
  - Viewing common information 38
- Create Package Wizard 148
  - Adding methods 150
  - Create Package Method Wizard 151
  - Specifying package definition 152
  - Specifying package properties 149
- Create Procedure Wizard 114
  - Managing parameters 116
  - Specifying procedure definition 117
  - Specifying procedure options 115
- Create Public database link Wizard 251
  - Specifying public database link properties 252
- Create Public synonym Wizard 245
  - Specifying public synonym properties 246
- Create Queue table Wizard 216
  - Specifying queue table properties 217
- Create Queue Wizard 211
  - Specifying queue properties 211
- Create Role Wizard 191
- Create Schema Trigger Wizard
  - Specifying schema trigger properties 135
  - Specifying trigger definition 138
- Create SchemaTrigger Wizard 134
- Create Sequence Wizard 161
- Create Synonym Wizard 130

Create Table Wizard 64  
 Create Tablespace Wizard 256  
   Adding tablespace files 258  
   Specifying tablespace properties 257  
 Create Trigger Wizard 88  
 Create Type body Wizard 172  
   Specifying type body definition 173  
 Create Type Wizard 166  
   Editing type subitems 167  
   Specifying type definition 168  
   Specifying type properties 166  
 Create updatable views 358  
 Create User profile Wizard 183  
   Specifying user profile properties 184  
 Create User Wizard 178  
 Create View Wizard 96  
 CRUD Procedures Generation 356

## - D -

Data Analysis 331  
   Data 333  
   Query 332  
 Data Management 278  
   BLOB editor 290  
   Data View 279  
   Export Data Wizard 295  
   Get SQL Dump Wizard 302  
   Import Data Wizard 305  
 Data View  
   Editing data in dialog 286  
   Lookup editor 286  
   Viewing data 279  
   Working with data grid 280  
   Working with info cards 285  
 Database Editor 33  
 Database Explorer  
   Filtering explorer content 54  
 Database link Editor 145  
   Editing link properties 146  
 Database links 142  
   Create Database link Wizard 143  
   Database link Editor 145  
 Database Management 21  
   Creating a database profile 23  
   Database editor 33  
   Editing a database profile 26  
 Database Object  
   Functions 122  
   Materialized Views 105  
   Procedures 113  
   Schemas 60  
   Tables 63  
   Views 95  
 Database Objects 59  
   Changing object properties 45  
   Copy, Paste and Drag-n-Drop features 49  
   Create object wizard 37  
   Creating a database object 36  
   Describing object 45  
   Duplicate Object wizard 46  
   Duplicating a database object 46  
   Duplicating selected object 48  
   Managing database objects 56  
   Object Editor 39  
   Queries 262  
   Roles 190  
   User profiles 182  
   Users 178  
   Viewing database objects 54  
 Database Profile Editor 26  
   Connection properties 26  
   Database options 27  
   Default directories 29  
   Editing obligatory scripts to execute 30  
   Setting log options 30  
   Statistics 31  
 Database trigger Editor  
   Editing database trigger properties 242  
 Database triggers 239  
   Create Database trigger Wizard 240  
 Diagram Viewer 327  
   Exporting diagram image 329  
   Selecting fields 328  
 Directories 233  
   Create Directory Wizard 234  
   Directory Editor 236  
 Directory Editor 236  
   Editing directory properties 237  
 DML Procedures Generation 356  
 Duplicate Object Wizard 46  
   Modifying new object's definition 48  
   Selecting object to duplicate 47  
   Selecting source and destination databases 46  
 Duplication a database object 46  
   Duplicate Object Wizard 46

- Duplication a database object 46
  - Duplicating selected objects 48

## - E -

### Editor & Viewer Options

- Code Folding 396
- Code Insight 395
- Display 391
- General 390
- PHP highlight 394
- SQL highlight 392
- XML highlight 393

### EULA 6

### Export Data Wizard 295

- Adjusting data formats 297
- Selecting fields for export 297
- Setting common export options 301
- Setting destination file name 295
- Setting format-specific options 298
- Setting header and footer 296

### Export Settings Wizard

- Saving settings 408
- Selecting database profiles 408
- Selecting setting categories 407
- Specifying destination file 407

## - F -

### Field Editor 75

### Field Properties

- LOB properties 78

### Filter Panel 54

### Find Text Dialog 363

### Foreign Key References 92

### Foreign Keys 82

### Function Editor 125

- Editing function properties 125
- Viewing function result 127

### Functions 122

- Create Function Wizard 123
- Function Editor 125

## - G -

### Generate Database Report Wizard 318

- Editing report style 319

- Selecting reporting elements 318
- Setting report paths 318
- Get SQL Dump Wizard 302
  - Selecting fields 302
  - Specifying dump options 303
- Getting Started 12
  - Explaining user interface 15
  - First time started 16
  - Switching between windows 18
  - Working with databases 13

## - I -

### Import Data Wizard 305

- Customizing common options 311
- Data Format 310
- Map builder 309
- Setting fields correspondence 308
- Setting source file name 306

### Indexes 79

### Installation instructions 4

## - J -

### Java source Editor 201

- Editing java source properties 202

### Java sources 198

- Create Java source Wizard 199
- Java source Editor 201

### Job Editor 207

- Editing job properties 208

### Jobs 204

- Create Job Wizard 205
- Job Editor 207

## - L -

### Libraries 228

- Create Library Wizard 229
- Library Editor 230

### Library Editor 230

- Editing library properties 231

### License Agreement 6

## - M -

### Materialized View Editor 110

Materialized Views 105  
 Create Materialized View Wizard 106  
 Materialized View Editor 110

## - O -

Object Browser 54  
 Object editor - basic principles 39  
 Function executing 44, 127  
 Object dependencies 42  
 Object grants 41  
 Parameter Editor 44  
 Permissions of an object 40  
 Results tab 44, 127  
 SQL definition 43  
 Object Management 35  
 Changing object properties 45  
 Copy, Paste and Drag-n-Drop features 49  
 Creating a database object 36  
 Describing object 45  
 Duplicating a database object 46  
 Managing clusters 221  
 Managing collections 193  
 Managing Database links 142  
 Managing database triggers 239  
 Managing directories 233  
 Managing functions 122  
 Managing java sources 198  
 Managing jobs 204  
 Managing libraries 228  
 Managing materialized views 105  
 Managing objects 56  
 Managing Package bodies 156  
 Managing Packages 147  
 Managing procedures 113  
 Managing public database links 250  
 Managing public synonyms 244  
 Managing Queries 262  
 Managing queue tables 215  
 Managing queues 210  
 Managing roles 190  
 Managing Schema triggers 133  
 Managing sequences 160  
 Managing synonyms 129  
 Managing tables 63  
 Managing tablespaces 255  
 Managing Type bodies 171  
 Managing types 165

Managing user profiles 182  
 Managing users 178  
 Managing views 95  
 Schemas 60  
 Viewing objects 54  
 Wizards and Editors 39  
 Object Manager 56  
 Options 367  
 Appearance 398  
 Application 368  
 Application confirmations 369  
 Application preferences 368  
 BLOB Viewer 378  
 Data Grid 383  
 Designer 380  
 Editor & Viewers 390  
 Editors 381  
 Explorer 373  
 Export 379  
 Export Settings 407  
 Object Manager 374  
 Query Builder 376  
 SQL Editor 374  
 SQL Script Editor 375  
 Tools 371  
 Oracle Maestro 1  
 Getting started 12  
 Installation 4  
 License agreement 6  
 Registration 5  
 System requirements 3

## - P -

Package bodies 156  
 Create Package body Wizard 157  
 Package body Editor 158  
 Package body Editor 158  
 Package body Wizard 157  
 Package Editor 153  
 Packages 147  
 Create Package Wizard 148  
 Package Editor 153  
 PL/SQL Debugger 348  
 Debug information 351  
 Debugging process 349  
 Procedure Editor 118  
 Editing properties 119

Procedure Editor 118  
     Viewing procedure results 120  
 Procedures 113  
     Create Procedure Wizard 114  
     Procedure Editor 118  
 Public database link Editor 253  
     Editing public database link properties 253  
 Public database links 250  
     Create Public database link Wizard 251  
     Public database link Editor 253  
 Public synonym Editor 247  
     Editing public synonym properties 248  
 Public synonyms 244  
     Create Public synonym Wizard 245  
     Public synonym Editor 247  
 Purchase Oracle Maestro 5

## - Q -

Queries 262  
     Query Parameters 268  
     SQL Editor 264  
     Visual Query Builder 269  
 Query Parameters 268  
 Queue Editor 212  
     Editing queue properties 213  
 Queue table Editor 218  
     Editing queue table properties 218  
 Queue tables 215  
     Create Queue table Wizard 216  
     Queue table Editor 218  
 Queues 210  
     Create Queue Wizard 211  
     Queue Editor 212

## - R -

Registration 5  
 Replace Text Dialog 364  
 Report Designer 336  
     Object Inspector 340  
     Toolbox 338  
 Role Editor  
     Editing role properties 192  
 Roles 190  
     Create Role Wizard 191

## - S -

Schema Designer 343  
     Navigation bar 345  
     Toolbox 345  
 Schema Editor 61  
 Schema Objects  
     Database links 142  
     Package bodies 156  
     Packages 147  
     Schema triggers 133  
     Sequences 160  
     Synonyms 129  
     Type bodies 171  
     Types 165  
 Schema Trigger Editor 138  
     Editing trigger properties 139  
 Schema Triggers  
     Create SchemaTrigger Wizard 134  
     Schema Trigger Editor 138  
 Schemas 60  
     Schema editor 61  
 SchemaTriggers 133  
 Sequence Editor 162  
 Sequences 160  
     Create Sequence Wizard 161  
     Sequence Editor 162  
 Server Objects  
     Changing object properties 45  
     Clusters 221  
     Collections 193  
     Copy, Paste and Drag-n-Drop features 49  
     Create object wizard 37  
     Database triggers 239  
     Describing object 45  
     Directories 233  
     Java sources 198  
     Jobs 204  
     Libraries 228  
     Object Editor 39  
     Public database links 250  
     Public synonyms 244  
     Queue tables 215  
     Queues 210  
     Tablespaces 255  
 Split table 359  
 SQL Editor 264

SQL Editor 264  
     Executing query 266  
 SQL Script Editor 316  
 Synonym Editor 131  
     Editing synonym properties 131  
 Synonyms 129  
     Create Synonym Wizard 130  
     Synonym Editor 131  
 System requirements 3

## - T -

Tabbed MDI 15  
 Table Editor 69  
     Editing table properties 69  
     Gathering statistic properties 74  
     Master-detail data view 72  
     Viewing table data 72  
 Tables 63  
     Checks 85  
     Create Table Wizard 64  
     Fields 75  
     Foreign key references 92  
     Foreign keys 82  
     Indexes 79  
     Table editor 69  
     Triggers 87  
 Tablespace Editor 259  
     Editing tablespace properties 260  
 Tablespaces 255  
     Create Tablespace Wizard 256  
     Tablespace Editor 259  
 Tools 313  
     BLOB Viewer 321  
     Data Analysis 331  
     Database Designer 343  
     Dependency Tracker 354  
     Diagram Viewer 327  
     Dialogs 363  
     Generate Database Report Wizard 318  
     PL/SQL Debugger 348  
     Process Browser 353  
     Report Designer 336  
     Schema Designer 343  
     Script Runner 315  
     SQL Generator 355  
     SQL Script Editor 316  
 Trigger Editor 90

Triggers 87  
     Create Trigger Wizard 88  
     Trigger Editor 90  
 Type bodies 171  
     Create Type body Wizard 172  
     Type body Editor 174  
 Type body Editor 174  
     Editing type body properties 175  
 Type Editor 169  
     Editing type properties 170  
 Types 165  
     Create Type Wizard 166  
     Type Editor 169

## - U -

User Editor 179  
     Editing user properties 179  
 User interface 15  
 User profile Editor 186  
     Editing user profile properties 186  
     Managing user profiles 188  
 User profiles 182  
     Create User profile Wizard 183  
     User profile Editor 186  
 Users 178  
     Create User Wizard 178  
     User Editor 179

## - V -

View Editor 101  
     Editing view properties 102  
     Viewing data 103  
 Views 95  
     Create View Wizard 96  
     View Editor 101  
 Visual Query Builder 269  
     Executing a query 276  
     Working with editor area 275

## - W -

WHERE condition 96, 106  
 Window List 18  
 Wizards and Editors 39  
     Create object wizard - basic principles 37

Wizards and Editors	39
Object Editor	39