



PostgreSQL Code Factory

User's guide

Table of Contents

Foreword	0
I Welcome to PostgreSQL Code Factory!	1
1 System Requirements	2
2 Installation	3
3 How can I purchase PostgreSQL Code Factory?	4
4 License Agreement	5
5 About SQL Maestro Group	7
6 What's new	10
II Getting Started	11
1 Connect to a database	12
2 Connection parameters	13
3 Explaining user interface	17
First time started	18
Tabbed MDI	19
Switching between windows	20
4 Shortcut keys	22
III Databases and Database Profiles	23
1 Creating Database Profiles	25
Setting connection properties	25
Setting profile options	26
2 Editing Database Profile	28
Editing connection properties	28
Setting profile options	29
Setting default directories	30
Editing obligatory scripts to execute	31
Setting log options	33
Statistics	34
IV Browse Objects	36
1 Database Explorer	37
Filtering explorer content	39
2 Filter Builder Dialog	41
V Queries and Scripts	42
1 SQL Editor	44
SQL Formatter	45
Executing query	46
Query Parameters	48
2 Visual Query Builder	49

Creating query diagram	50
Working with editor area	55
Executing query	56
3 Script Runner	58
4 SQL Script Editor	59

VI Data management 61

1 Data View	62
Working with data grid	63
Working with info cards	68
Data input form	69
Data filtering	70
2 BLOB Editor	73
Editing as image	73
Editing as hexadecimal dump	74
Editing as plain text	74
Editing as HTML	75
Editing as PDF document	76
3 Export Data Wizard	78
Setting destination file name and format	78
Setting header and footer	79
Selecting fields for export	80
Adjusting data formats	80
Setting format-specific options	81
Setting common export options	84
4 Get SQL Dump	85
Selecting fields	85
Specifying dump options	86
5 Import Data Wizard	88
Setting source file name and format	89
Setting the accordance between source and target columns	91
Map builder	92
Data formats	93
Customizing common options	94

VII Database Tools 96

1 BLOB Viewer	97
Viewing as hexadecimal dump	97
Viewing as plain text	98
Viewing as image	99
Viewing as HTML	100
Viewing as PDF	101
2 Diagram Viewer	103
Customizing diagram properties	104
Exporting diagram image	105
3 SQL Generator	107
4 Dialogs	108
Find Text dialog	108
Replace Text dialog	109

VIII Options	112
1 Application	113
Preferences	113
Confirmations	114
Tools	115
Explorer	116
SQL Editor	117
SQL Script Editor	118
Query Builder	119
Colors	121
BLOB Viewer	121
Data Export	122
Data Grid	123
Options	124
Colors	126
Formats	126
Filter	128
2 Editors & Viewers	130
General	130
Display	131
SQL highlight	132
XML highlight	133
PHP highlight	134
Code Insight	135
Code Folding	136
3 Appearance	138
Bars and menus	138
Trees and lists	139
Edit controls	140
Check boxes	141
Buttons	142
Page controls	143
Group boxes	144
Splitters	145
4 Export Settings	147
Specifying destination file	147
Selecting setting categories	147
Selecting database profiles	148
Saving settings	148
Index	150

1 Welcome to PostgreSQL Code Factory!

PostgreSQL Code Factory is a premier PostgreSQL GUI tool aimed at the SQL queries and scripts development. It is a good choice for everyone who need build SQL statements and edit SQL scripts with a convenient easy-to-use interface. The software provides you with a convenient easy-to-use interface, so it really does not require a deep knowledge of PostgreSQL from its users.

Key features include:

- **Visual Query Builder:** PostgreSQL Code Factory provides you with the powerful tool intended for designing queries as visual diagrams. This tool does not require any knowledge of the SELECT statement syntax, it will form a query automatically, you just need to mark what information you want to retrieve;
- **Handy SQL Editor** with code folding and syntax highlighting to prevent mistakes in syntax at once. Also it is possible to separate SQL scripts into regions that can be individually collapsed or expanded;
- **Simultaneous executing** of several queries with multi-threading in order to continue your work with the software while the query is executing;
- **Advanced data management:** viewing, editing, grouping, sorting and filtering abilities to analyze the data in the most convenient way;
- **Data export** to as many as 14 file formats including Excel, RTF and HTML;
- **Data import** from Excel, CSV, text files and more;
- **Powerful BLOB Viewer/Editor:** with PostgreSQL Code Factory you can view or edit BLOB data in the following ways: hexadecimal dump, plain text, graphical image or HTML page. A graphical representation of BLOB data supports five image formats: BMP, Windows metafile, JPEG, GIF and PNG.

The application also provides you with a powerful set of tools to edit and execute SQL scripts, build visual diagrams for numeric data, customize user interface according to your needs and much more.

With all these features our software will be an everyday assistant in your work with PostgreSQL database server!

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.

Server environment

- PostgreSQL from 7.3 to 16.

1.2 Installation

To install **PostgreSQL Code Factory** on your PC:

- download the PostgreSQL Code Factory 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 PostgreSQL Code Factory shortcut in the corresponding program group of the Windows Start menu after the installation is completed.

1.3 How can I purchase PostgreSQL Code Factory?

Thank you for your interest in purchasing **PostgreSQL Code Factory**!

You can select licensing options and register PostgreSQL Code Factory 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.

PostgreSQL Code Factory 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.

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.

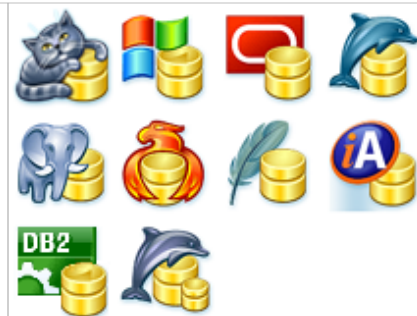
The slogan of our company is **The Shortest Path to SQL**. It is aimed to denote that we set to create easy-to-use products meant for those who appreciate comfort, friendly program interface and support when working with SQL servers.

- 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 PostgreSQL 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 PostgreSQL .

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

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 PostgreSQL Code Factory news at <http://www.sqlmaestro.com/products/postgresql/codefactory/news/>

2 Getting Started

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

How to get started:

- [Connect to a database with PostgreSQL Code Factory](#)^[12]
- [Explaining user interface](#)^[17]
- [How PostgreSQL Code Factory looks when you start it for the first time](#)^[18]
- [Shortcut keys](#)^[22]

Learning more:

- ❑ See [Database Tools](#)^[96] section for instructions on more advanced procedures!
- ❑ Find out more about [Working with Data in PostgreSQL Code Factory](#)^[61].
- ❑ Customize the way PostgreSQL Code Factory works, see [Program Options](#)^[112] for full details.

2.1 Connect to a database

To manage an existing database with PostgreSQL Code Factory, you have to [create the according database profile](#)^[28] 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](#)^[28]).

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

How can I disconnect from a database?

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

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

See also: [Connection parameters](#)^[13]

2.2 Connection parameters

PostgreSQL Code Factory allows you to connect to PostgreSQL directly, or via Secure SHell (SSH) tunnel, or HTTP tunnel.

- **Direct connection**

It is the most natural and the most preferable connection mode. Use it each time it is possible.

- **SSH tunnel connection**

If your PostgreSQL server does not allow direct connections from your remote workstations, you can establish connection to an allowed intermediate SSH server and forward all PostgreSQL commands through the [Secure SHell \(SSH\) tunnel](#).

- **HTTP tunnel connection**

[HTTP tunneling](#) is a technique used in conditions of restricted network connectivity including firewalled networks, networks behind proxy servers, and NATs. It is the slowest way and is recommended to use if the others are impossible.

Irrespectively of a connection mode you should specify common credentials as follows:

Host

The host name of the PostgreSQL server.

Port number

The TCP/IP port to use. Default PostgreSQL port is 5432.

User name

The username used to connect to PostgreSQL.

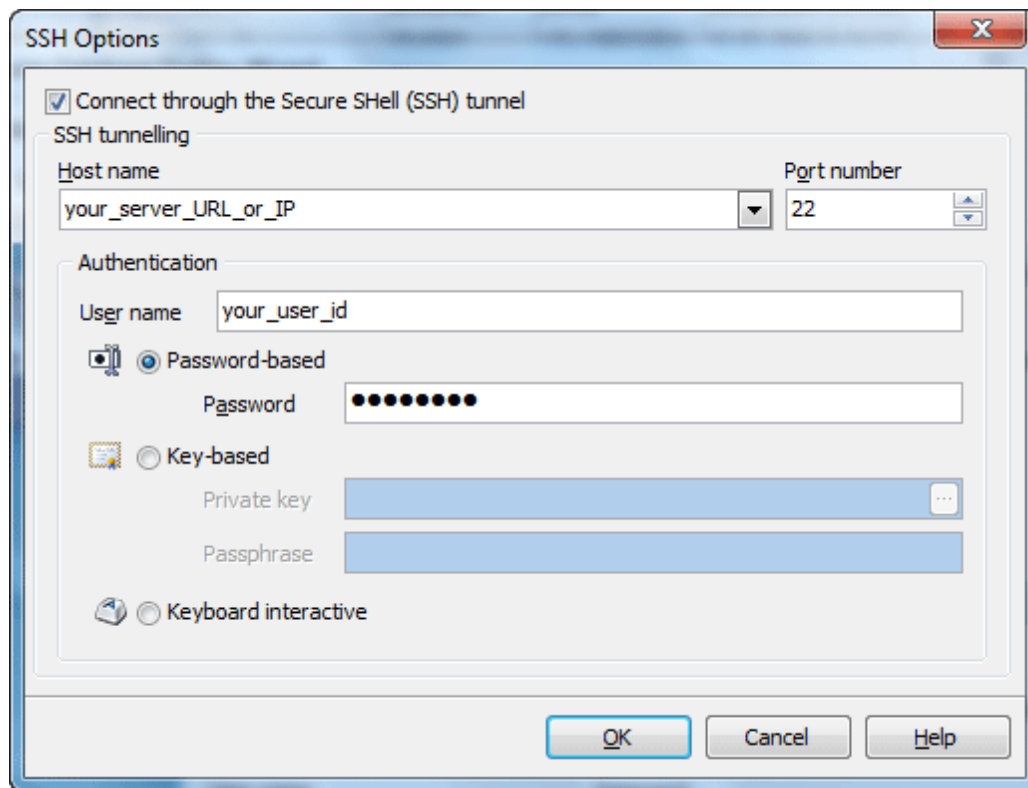
Password

The password for the user account on server.

📖 **More about SSH tunnel connection**

To establish connection to intermediate SSH server and forward all PostgreSQL commands through the secure tunnel, you need to:

1. Check [I can connect to the server directly or via SSH tunneling](#).
2. Follow the [Configure SSH options](#) link to open the [SSH Options](#) window.



3. Check [Connect through the Secure Shell \(SSH\) tunnel](#) and complete the following fields:

Host name

Specify the host name or IP of your site. Note, that PostgreSQL host name always should be set relatively to the SSH server. For example, if both of PostgreSQL and SSH servers are located on the same computer, you should specify localhost as Host name instead of server's external host name or IP address.

Port number

Enter the port number for the SSH server.

4. Enter valid [User name](#) for the remote server and select the [Authentication](#) method and set corresponding credentials.

Password-based

Set the [password](#) corresponding to the specified user.

Key-based

Specify the path to the [Private key](#) file with the corresponding [Passphrase](#) to log in to the remote server. PostgreSQL Code Factory accepts keys in **ssh.com** or **OpenSSH** formats. To convert a private key from PuTTY's format to one of the formats supported by our software, [use the PuTTYgen utility](#) that can be freely downloaded from the [PuTTY website](#).

Keyboard interactive

Keyboard authentication is the advanced form of password authentication, aimed specifically at the human operator as a client. During keyboard authentication zero or more prompts (questions) is presented to the user. The user should give the answer to each prompt (question). The number and contents of the questions are virtually not limited, so certain types of automated logins are also possible.

More about connection via HTTP tunnel

To connect to a remote server using an HTTP tunnel, you need to:

1. Upload the connection PHP script to your website. The script is named *pgsql_tunnel.php* and can be found under the installation folder, usually *C:\Program Files\SQL Maestro Group\PostgreSQL Code Factory*.
2. Select the [I have to use HTTP tunneling](#) radio button.
3. Enter the connection PHP script URL, e.g. *www.yoursite.com/files/pgsql_tunnel.php*. You can test the connection before the profile is created. Just use [Test script using default browser](#) to open connection script in your browser, enter all the required connection parameters and click the [Test connection](#) button.

Connection Script

Fields marked by * are required.

Host/Server name (or IP) *:	<input type="text" value="neptun"/>
User *:	<input type="text" value="postgres"/>
Password:	<input type="password" value="••••••••"/>
Port (if not 5432):	<input type="text" value="5433"/>
Database *:	<input type="text" value="adventure"/> <input type="button" value="v"/>
	<input type="button" value="Get Database List"/>
	<input type="button" value="Test Connection"/> <input type="button" value="ShowTables"/>

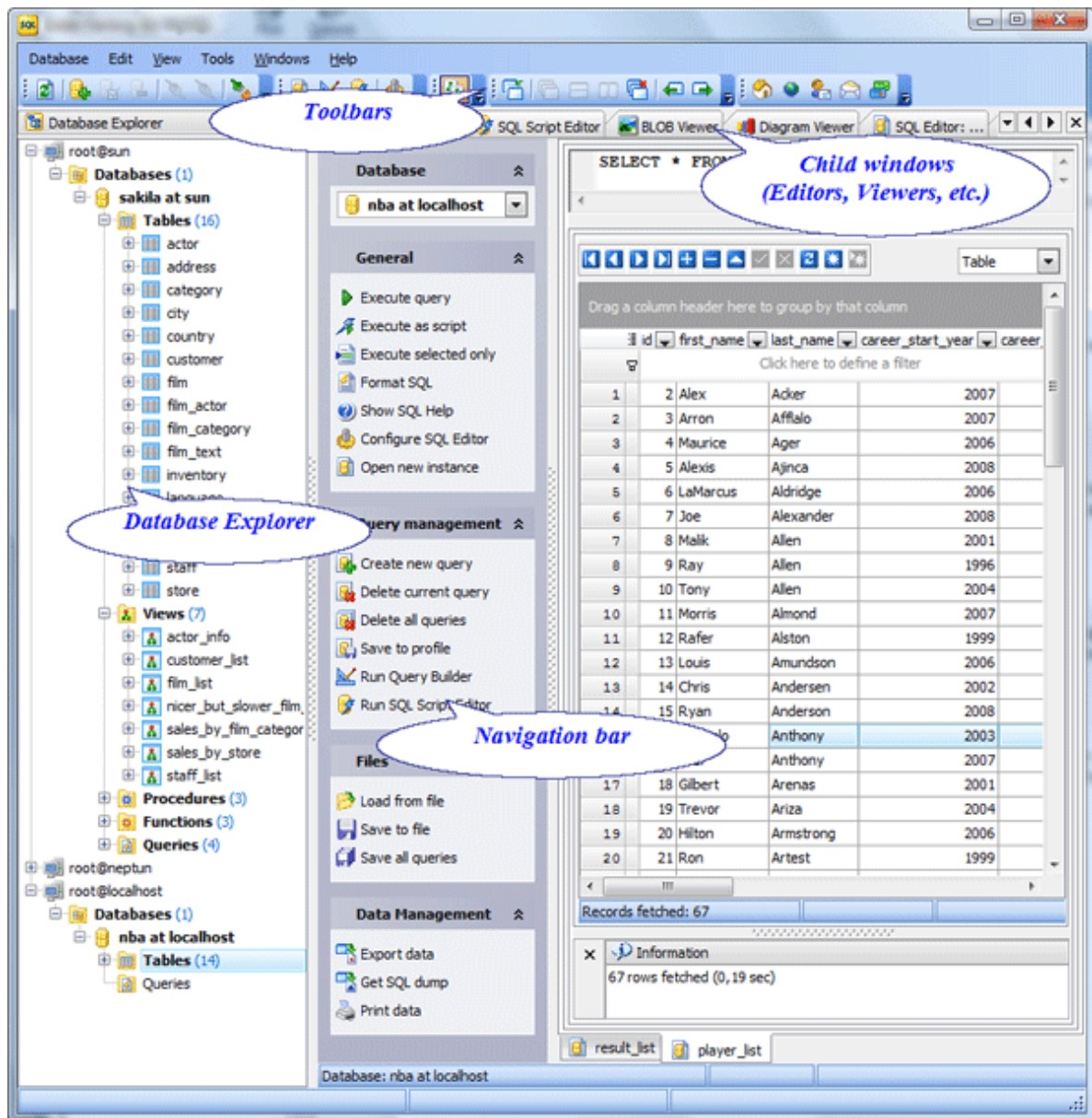
© 2002-2008 SQL Maestro Group

4. In case using of a proxy server use [Configure tunnelling options](#) to open the [HTTP tunnelling options](#) window and specify your [proxy server](#) connection parameters and [HTTP authentication](#).

Note: You are actually connecting to your database through the PHP script on the server, so in most cases the host/server name is "localhost" unless the target database server is not installed on the same computer as the Web server.

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](#).



Database Explorer

The [Database Explorer](#)^[37] occupies the left side of PostgreSQL Code Factory main window. It represents all objects of the connected database [including system objects](#)^[29].

The explorer provides the fastest way to reach object SQL definitions.

See also: [Filtering explorer content](#)^[39]

Editors and Viewers

According to the MDI style implementation the PostgreSQL Code Factory 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](#)^[20], [Tabbed MDI](#)^[19]

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.

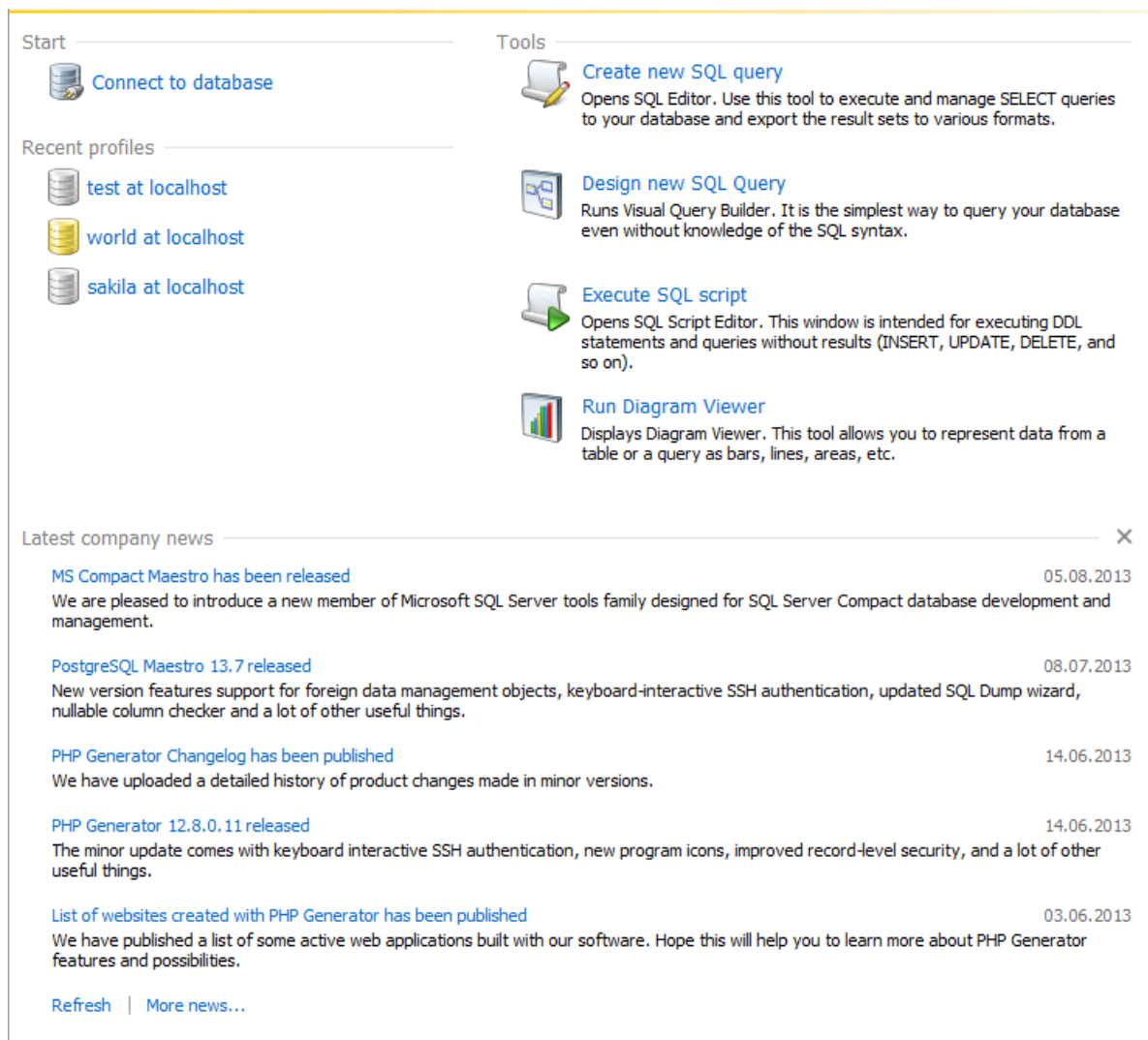
See also: [Shortcut keys](#)^[22]

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 PostgreSQL Code Factory 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](#)^[25]

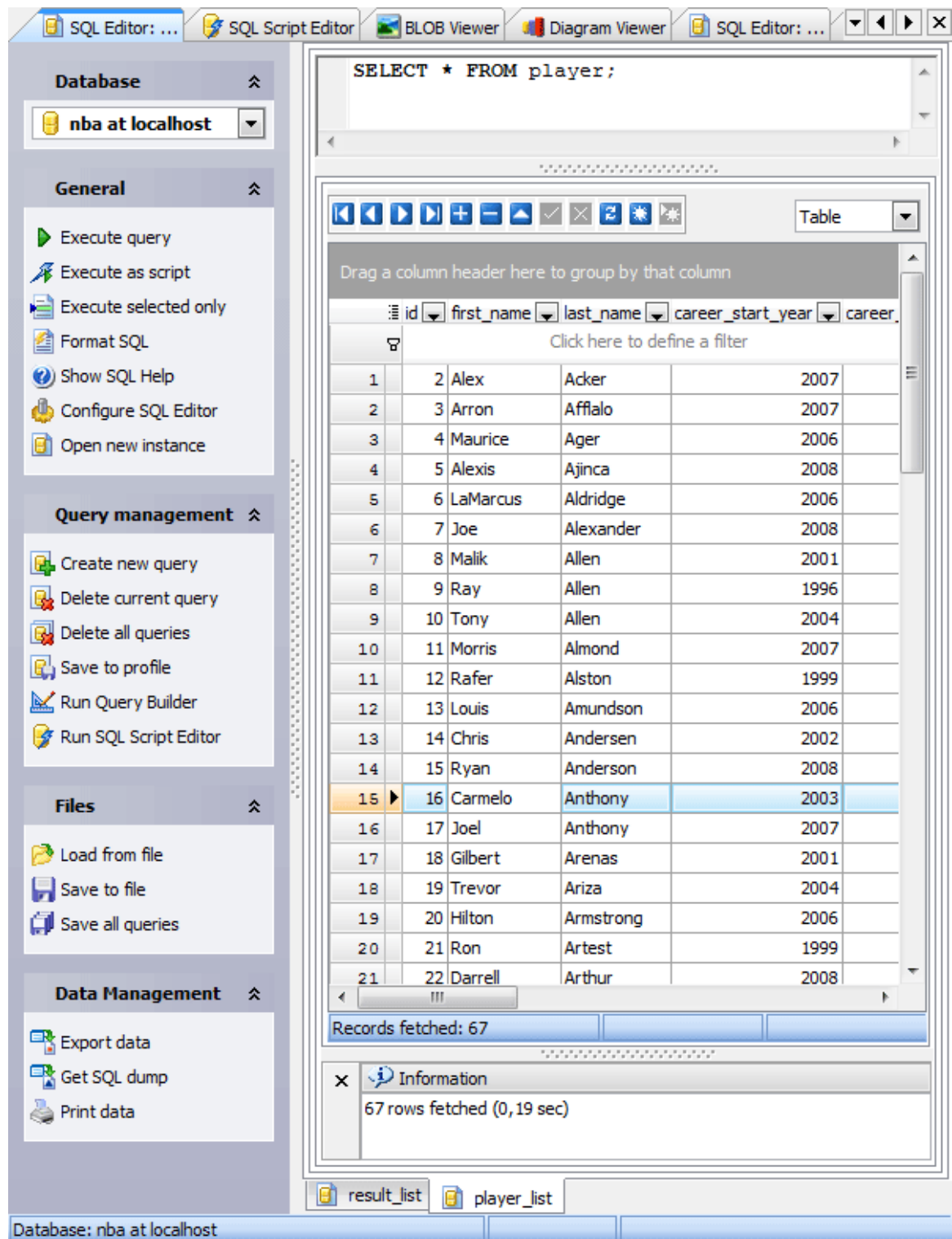


2.3.2 Tabbed MDI

PostgreSQL Code Factory 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.

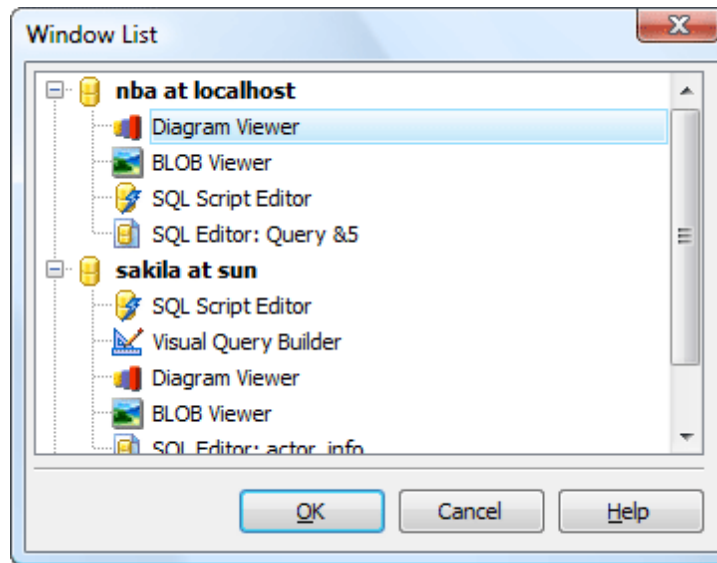


2.3.3 Switching between windows

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 PostgreSQL Code Factory.

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

3 Databases and Database Profiles

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

Use the following links for details:

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

In PostgreSQL Code Factory database profiles are created within [Create Database Profiles Wizard](#)^[25]. 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](#)^[26] 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

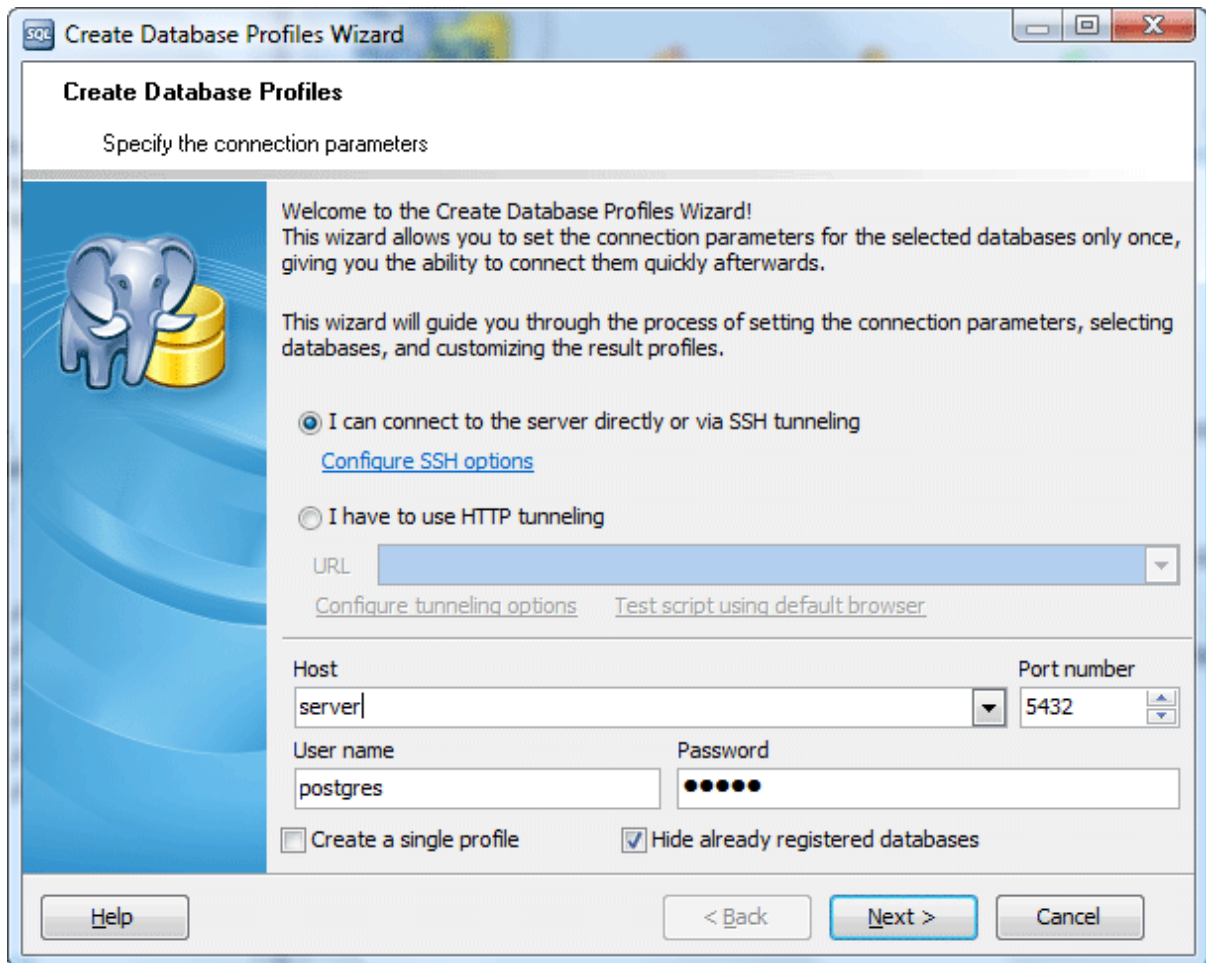
Create Database Profiles Wizard allows you to create a single database profile or several profiles from one host. To run the wizard, select the **Database | Create Database Profiles...** main menu item, or press the **Shift+Ctrl+P** hot keys combination. You can also use the **Create Database Profiles**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

- [Set connection properties](#)^[25]
- [Specify database profile options](#)^[26]

See also: [Edit Database Profile Dialog](#)^[28]

3.1.1 Setting connection properties

Specify PostgreSQL connection properties to be used on further connections.



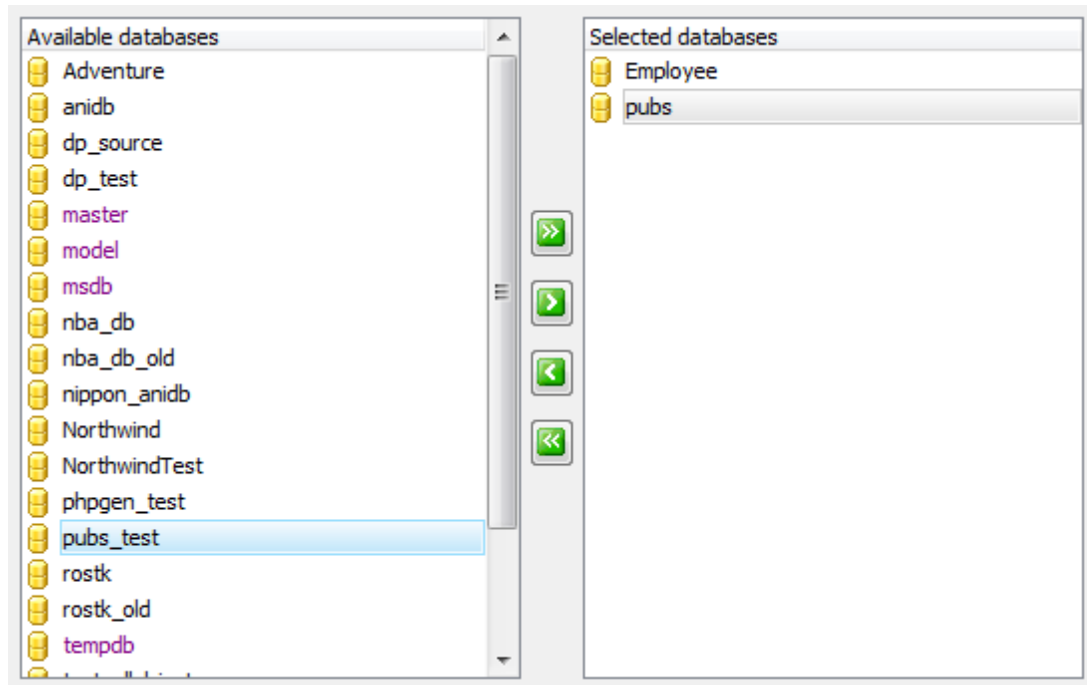
☒ Check the [Create a single profile](#) option to set the database name manually and create a single profile for this database.

☒ [Hide already registered databases](#)

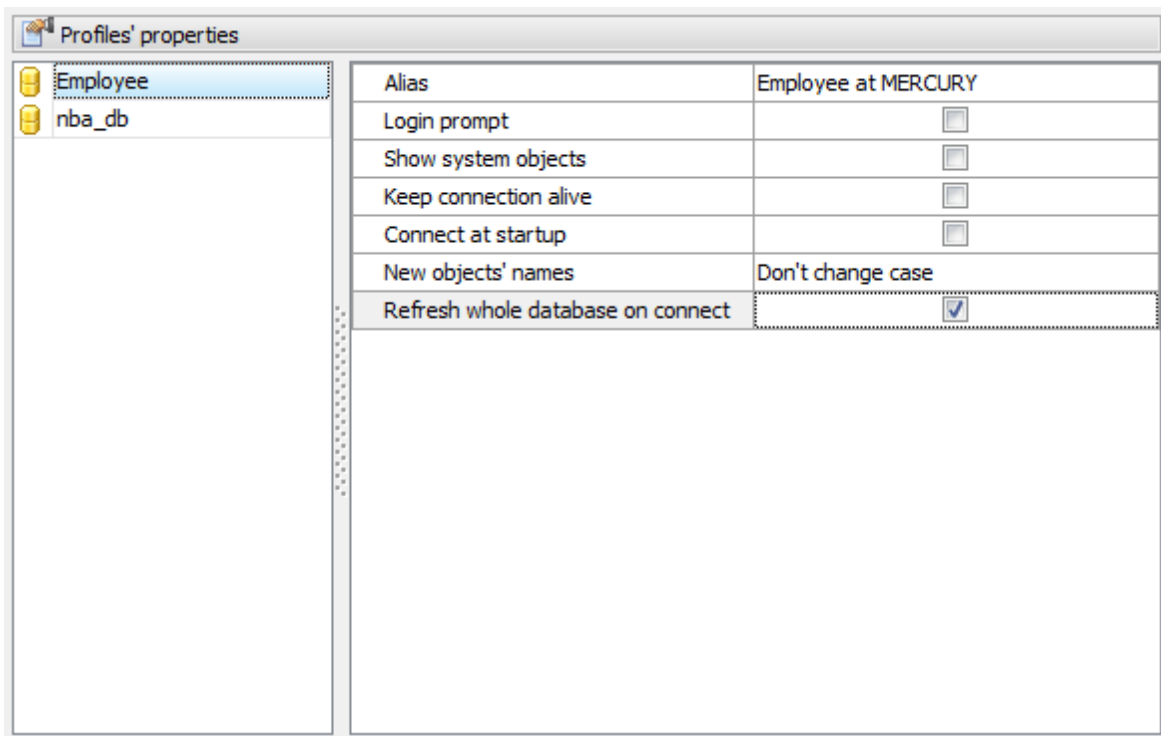
Check the box to shorten the databases list on the next wizard step.

3.1.2 Setting profile options

To create a new profile, select databases to be registered by moving them from the [Available databases](#) list to the [Selected databases](#) list. This step is available if the [Create a single profile](#) option is unchecked on the previous wizard step.



You can select several databases to set options for all the selected databases at once (except the alias which should be unique for each individual database).



☒ [Login prompt before connection](#)

Use the option to enable PostgreSQL Code Factory to prompt for user name and password every time you connect to the database.

☒ [Show system objects](#)

Check the box to make system objects visible.

☒ [Keep connection alive](#)

Check the box for pinging server before each query execution.

☒ [Connect at startup](#)

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

[New objects' names \(Don't change case, Convert to upper case, Convert to lower case\)](#)

The option allows you to specify the newly created objects case.

☒ [Refresh whole database on connect](#)

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

[Profile text color](#)

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

Click the [Ready](#) button when done to start working with the selected databases in PostgreSQL Code Factory.

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](#) ^[28]
- [Settings database options](#) ^[29]
- [Setting default directories for database tools](#) ^[30]
- [Editing obligatory scripts to execute](#) ^[31]
- [Setting log options and file names](#) ^[33]

See also: [Create Database Profile Wizard](#) ^[25]

3.2.1 Editing connection properties

The tab allows you to change connection properties 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.

Connection

☒ I can connect to the server directly or via SSH tunneling
[Configure SSH options](#)

☐ I have to use HTTP tunneling
URL:
[Configure proxy settings](#) [Test script using default browser](#)

Host: Port number:

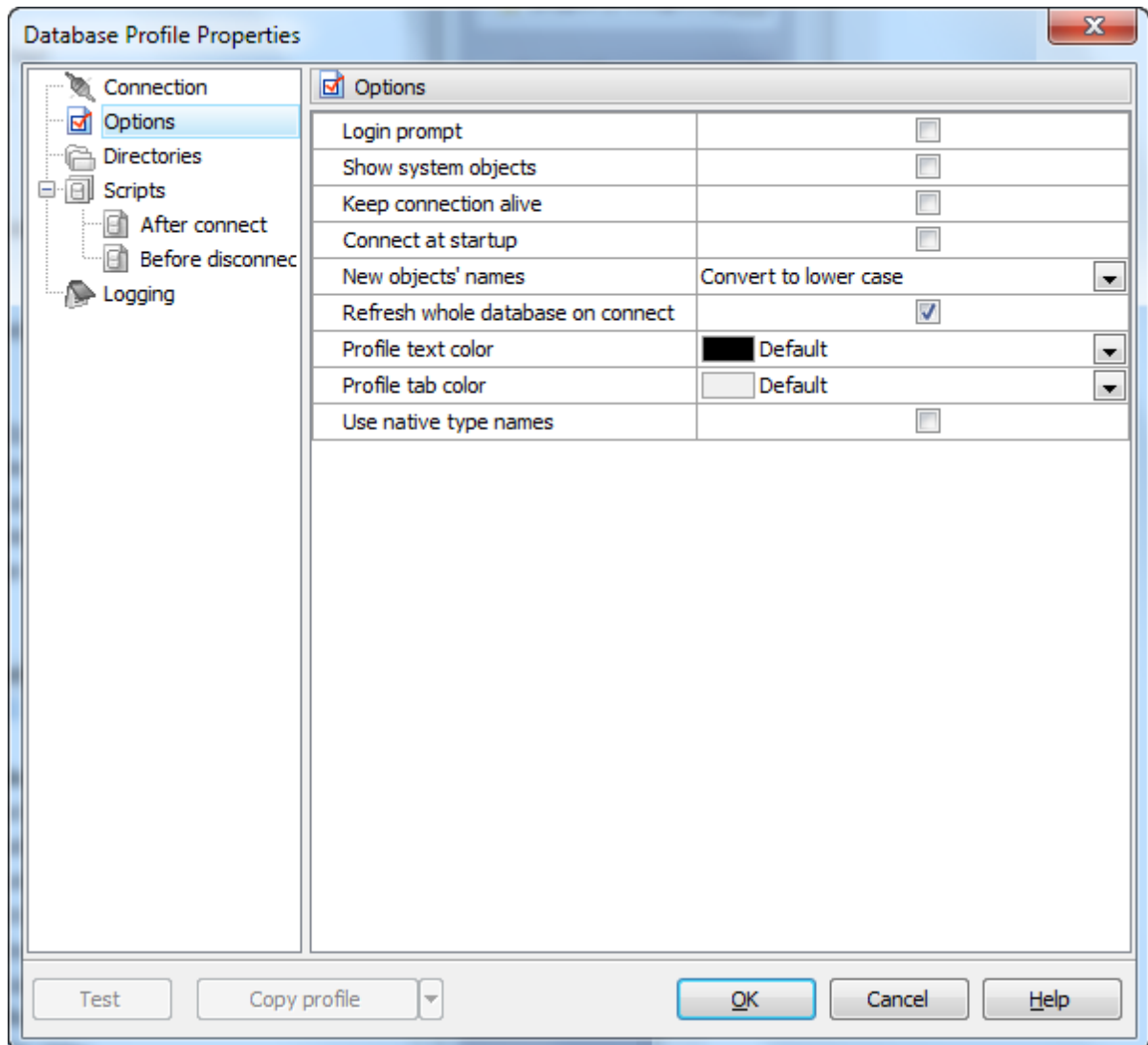
User name: Password:

Database name:

Database alias:

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 PostgreSQL Code Factory 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 connection to the profile database is automatically established at the application startup.

New objects' names (Don't change case, Convert to upper case, Convert to lower case)
Use the option to change the case for newly created objects.

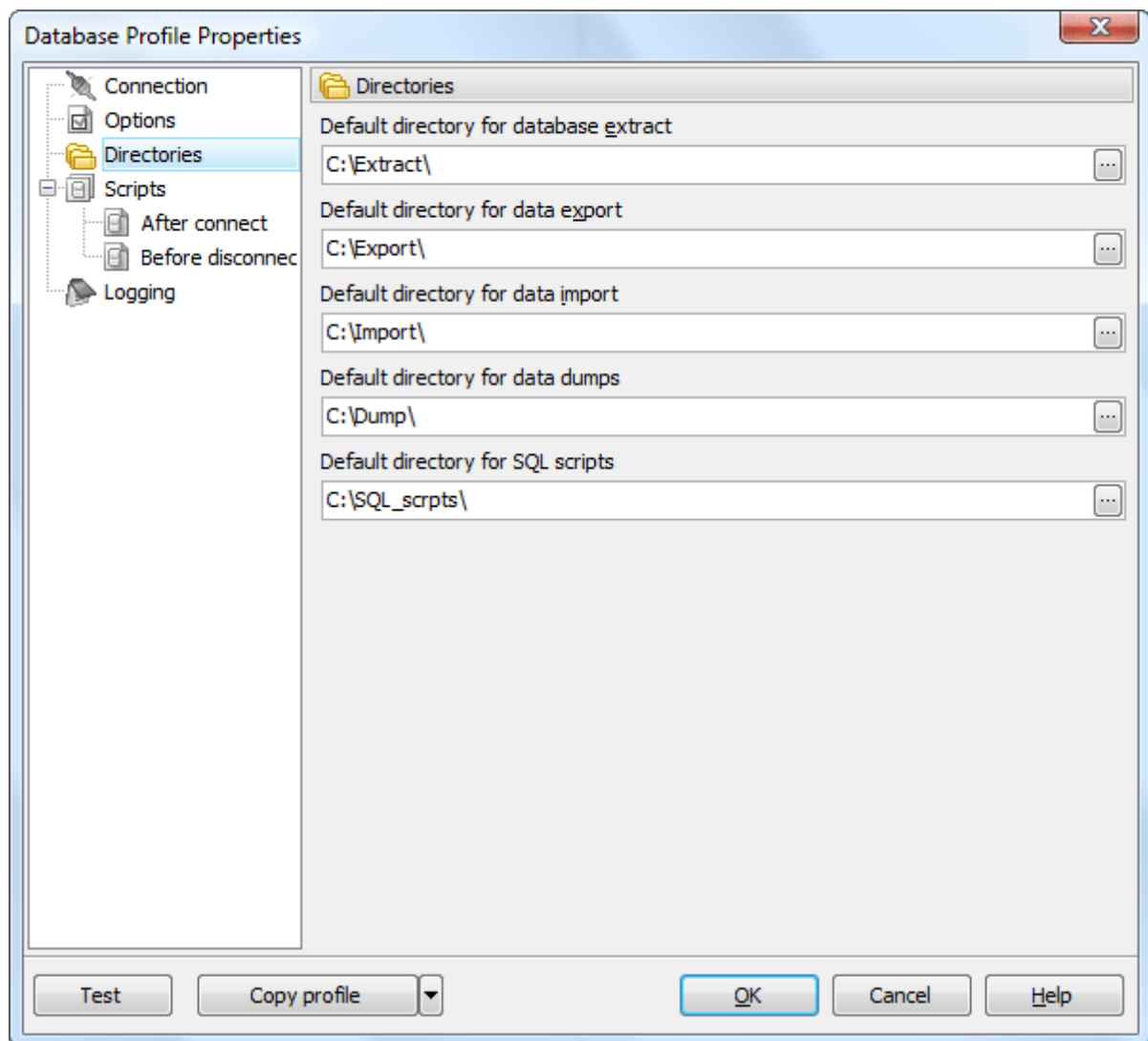
☒ Refresh whole database on connect

Use the option along with the [Show empty schemas](#)¹¹⁶ explorer options to hide/show empty schemas in the explorer tree.

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

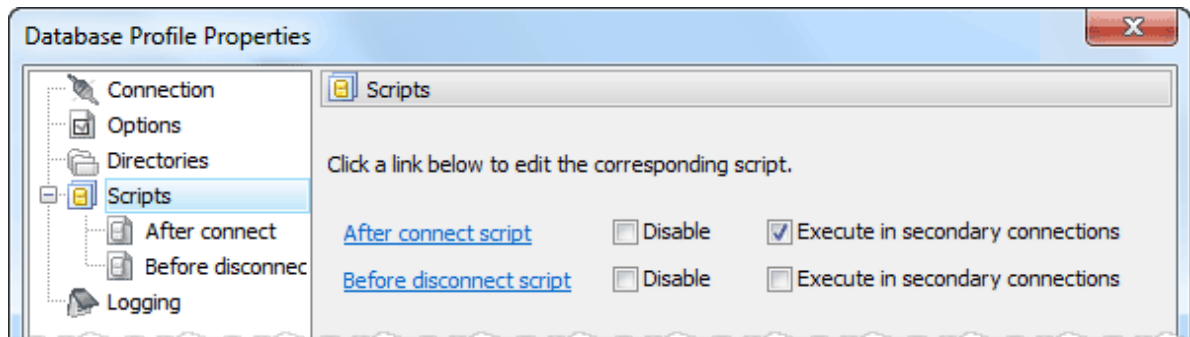
3.2.3 Setting default directories

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

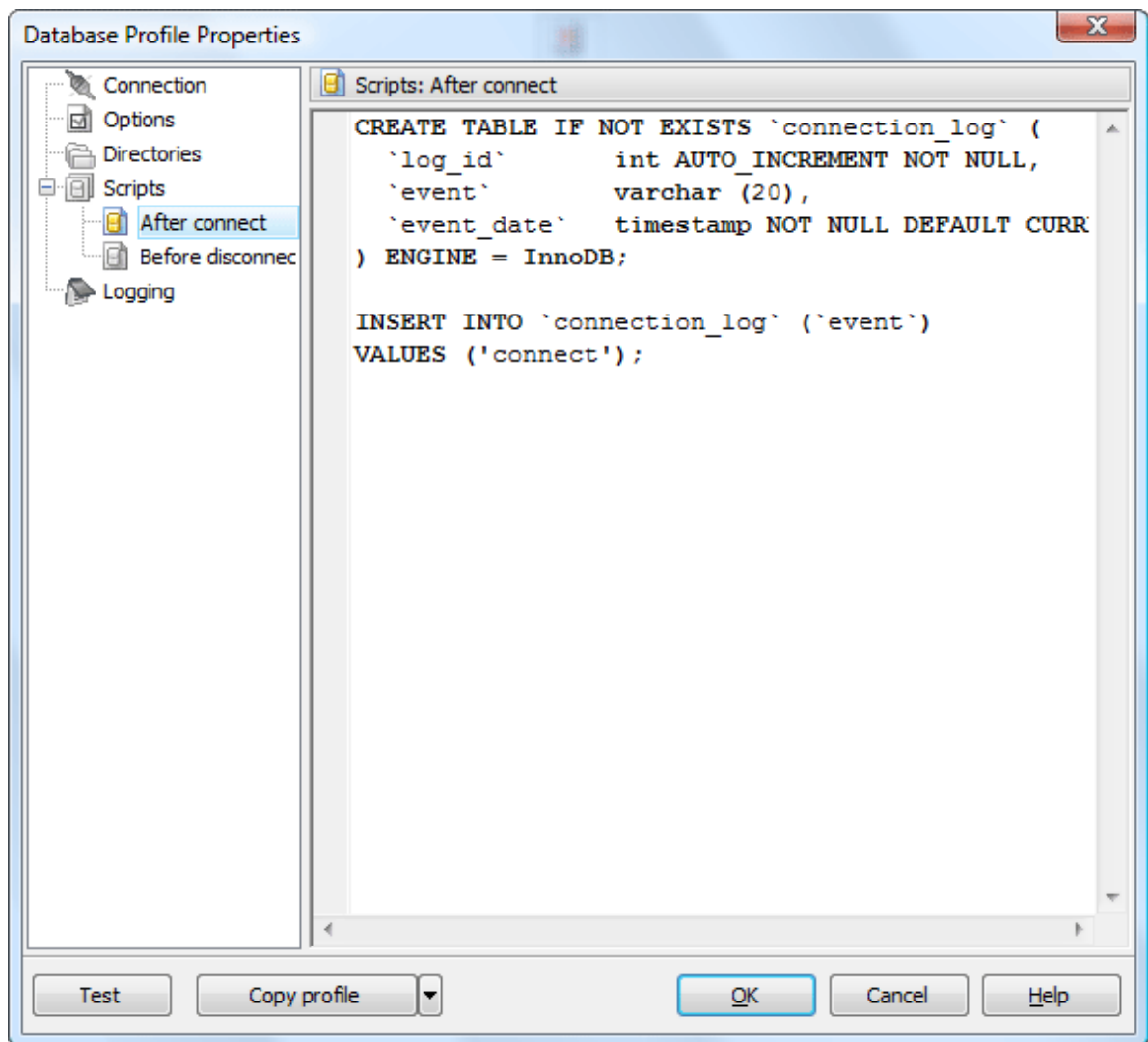


3.2.4 Editing obligatory scripts to execute

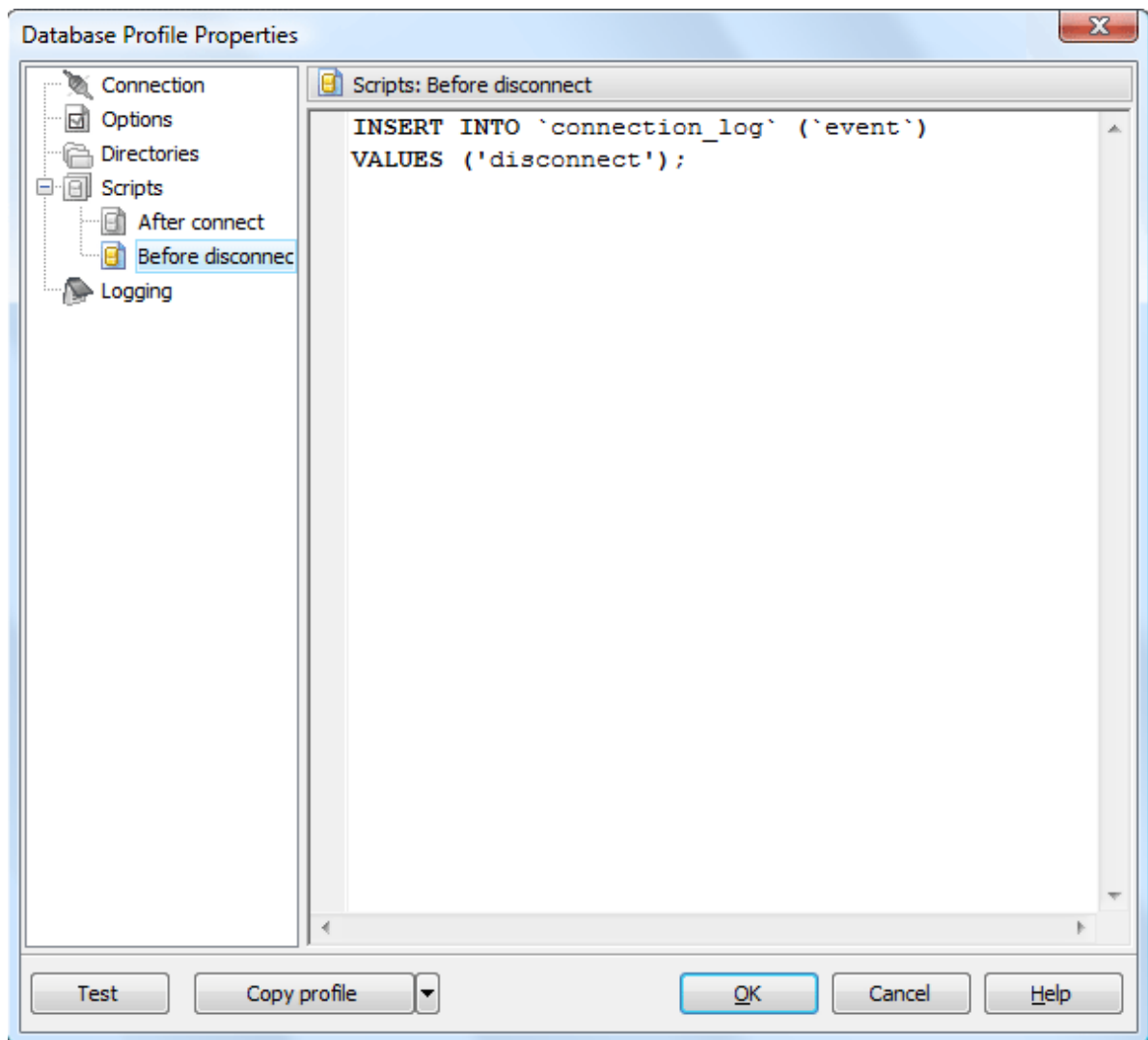
Use the tab to specify the obligatory scripts to execute in all database connections established by the software (on executing queries, browsing objects data, etc.). There is a possibility to enable/disable a written script.



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

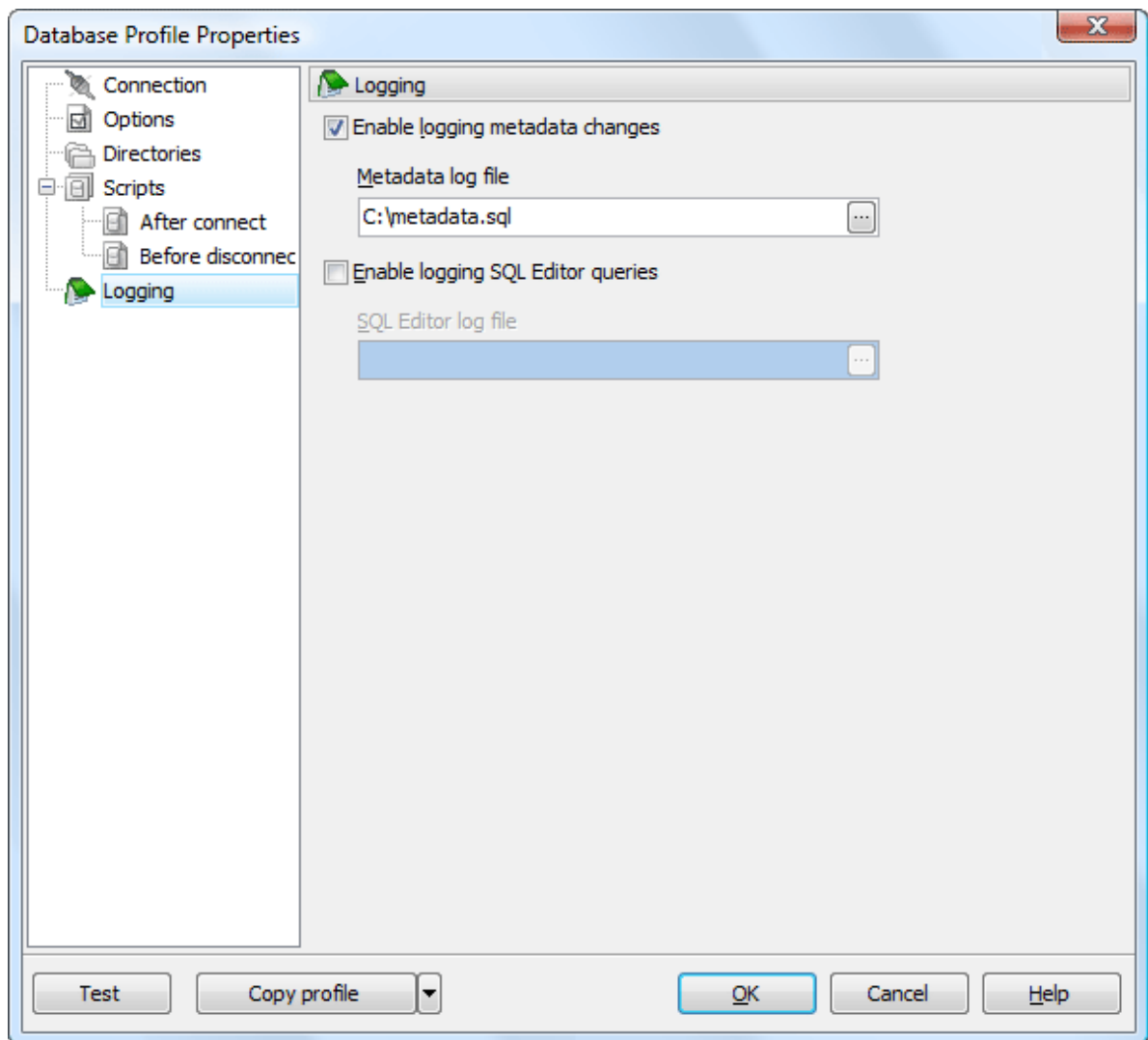


The next screen represents the example of an obligatory script to execute before PostgreSQL Code Factory will disconnect from the database. The script writes a disconnect time to the log table.




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

4 Browse Objects

PostgreSQL Code Factory allows to browse objects stored in a Remote Server database within Database Explorer. It represents objects grouped by kind and listed under the according PostgreSQL servers/database node, provided with subobjects if exist. It's possible to [reduce](#)^[39] the number of represented objects in the explorer tree and also to [hide/display](#)^[116] table subobjects, represent system objects in different color, etc.

4.1 Database Explorer

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

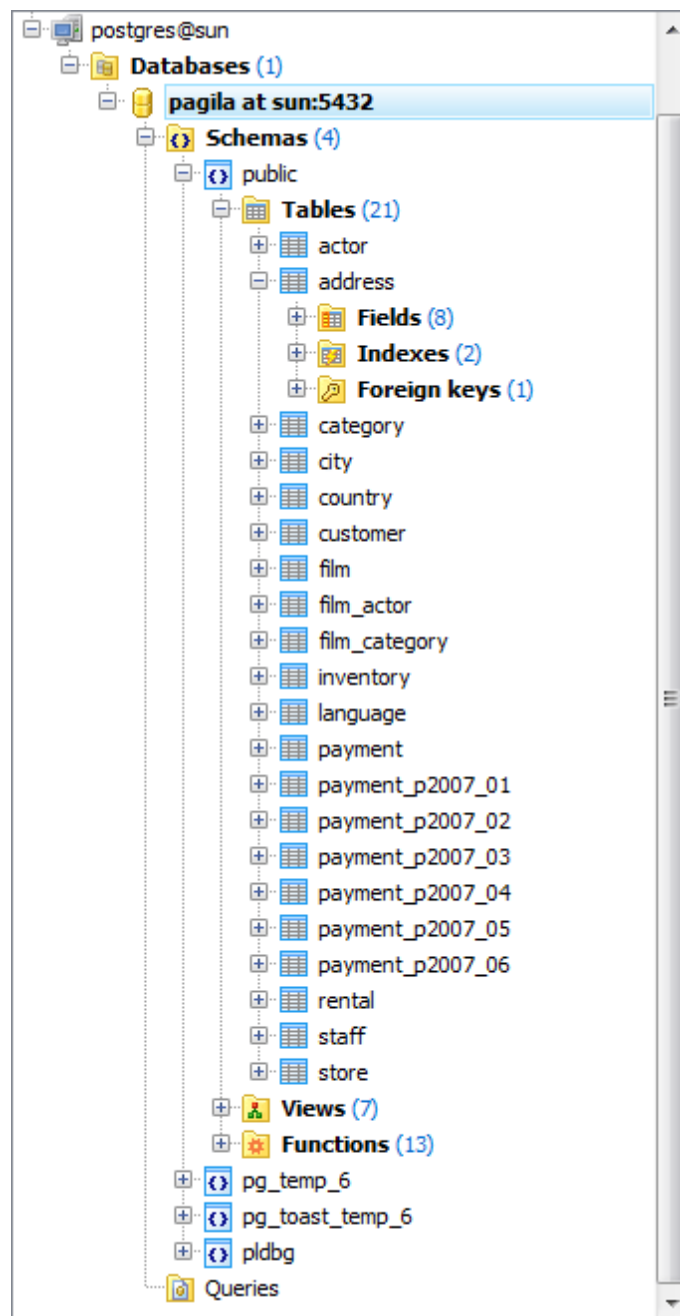
To start working with a database you need to create its profile first. The conception of database profiles gives you an opportunity to connect to databases in one touch and work with the selected databases only.

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](#)^[116] 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?](#)^[38]
- [How can I connect to a database?](#)^[37]
- [How can I disconnect from a database?](#)^[39]



Operations upon database profiles in the Explorer Tree

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

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

- Database Profile item);
- remove all profiles of selected server (the Remove all Profiles item of Databases node's popup menu).

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

How can I connect to a database?

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

How can I disconnect from a database?

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

Operations upon database objects

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

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

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

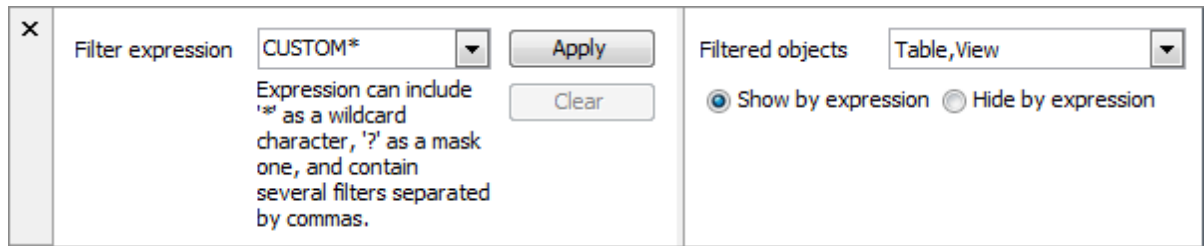
Database Explorer provides you with an ability of copying database objects from one database to another. To perform this operation, you should connect to both the source and the destination databases first. After the connection is established, simply drag and drop an object to copy from the source database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

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

4.1.1 Filtering explorer content

PostgreSQL Code Factory 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.



The screenshot shows a dialog box titled 'Filter Panel' with a close button (X) in the top-left corner. The dialog is divided into two main sections. The left section is for defining the filter expression, featuring a dropdown menu currently set to 'CUSTOM*', an 'Apply' button, and a 'Clear' button. Below the dropdown, a text box explains: 'Expression can include '*' as a wildcard character, '?' as a mask one, and contain several filters separated by commas.' The right section is for selecting filtered objects, with a dropdown menu set to 'Table, View' and two radio buttons: 'Show by expression' (which is selected) and 'Hide by expression'.

- 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.

4.2 Filter Builder Dialog

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

5 Queries and Scripts

PostgreSQL Code Factory provides several tools for working with SQL queries and scripts:

- [SQL Editor](#)^[44] for editing the query text directly and executing SELECT queries;
- [Visual Query Builder](#)^[49] for building SELECT, INSERT, UPDATE and DELETE queries visually.

Both SQL Editor and Visual Query Builder supports [parameters in queries](#)^[48]

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, 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](#).

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.

5.1 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](#)^[42].

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](#)^[117] 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](#)^[46].

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](#)^[45] is also at your disposal.

SQL Editor allows you to use [Visual Query Builder](#)^[49] 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

PostgreSQL Code Factory 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](#)^[134].

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.

Transaction management

SQL Editor supports the explicit transaction management. You can execute queries either in [autocommit mode](#) (default behavior) or [manage transactions manually](#). In the second case you have to issue the *BEGIN* statement to start a transaction and explicitly end the transaction by *COMMIT* or *ROLLBACK* statements (it is also possible to use the corresponding links at the editor's navigation bar).

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.

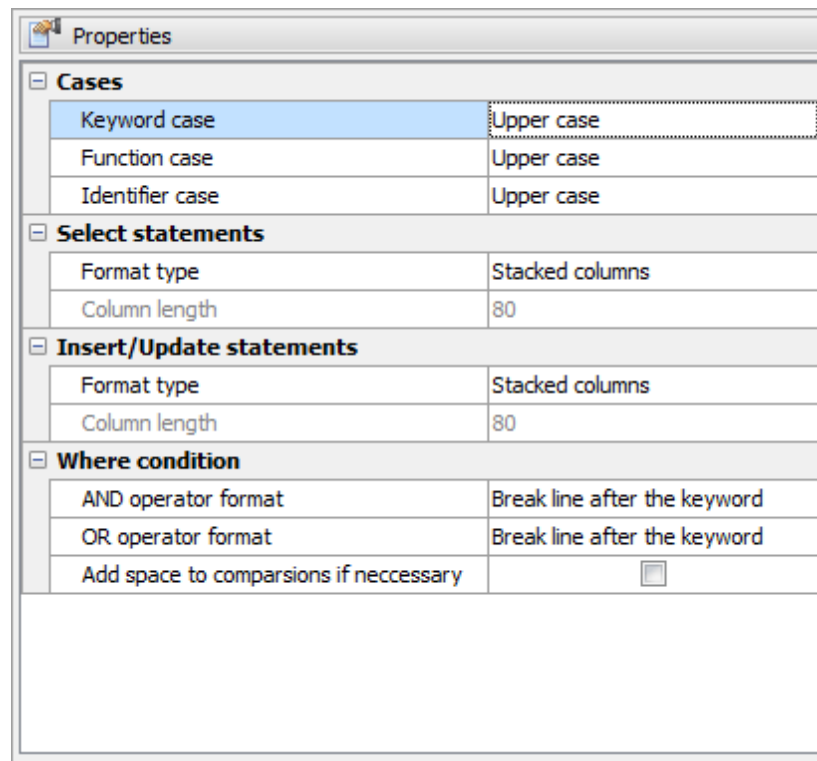
See also: [Visual Query Builder](#)^[49], SQL Script Editor, [SQL Editor Options](#)^[117]

5.1.1 SQL Formatter

PostgreSQL Code Factory 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.



5.1.2 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](#) ^[61] section.

The screenshot displays the SQL Maestro Group application interface. The top toolbar includes icons for SQL Editor, Visual Query Builder, Diagram Viewer, actor, BLOB Viewer, and Data Analysis. The left sidebar contains several expandable sections: Database (showing 'sakila at localhost'), General (with options like 'Execute query', 'Execute as script', etc.), Query management (with options like 'Create new query', 'Delete current query', etc.), Files (with options like 'Load from file', 'Save to file', etc.), and Data Management (with options like 'Export data', 'Get SQL dump', etc.).

The main window is titled 'CUSTOMER_CONTACT_INFO' and 'Query 1'. It contains the following SQL query:

```
SELECT
    CU.CUSTOMER_ID AS ID,
    CU.FIRST_NAME FIRST_NAME,
    CU.LAST_NAME AS LAST_NAME,
    A.PHONE AS PHONE,
    A.POSTAL_CODE AS ZIP_CODE,
    CU.EMAIL AS EMAIL,
    CN.COUNTRY
FROM CUSTOMER CU
JOIN ADDRESS A ON
    CU.ADDRESS_ID = A.ADDRESS_ID
INNER JOIN CITY C ON
    C.CITY_ID = A.CITY_ID
```

Below the query editor is a toolbar with navigation icons. The results pane shows a table with columns: ID, FIRST_NAME, LAST_NAME, PHONE, ZIP_CODE, and EMAIL. The data is grouped by country, with the 'United States' group expanded, showing 36 records. The first few rows of the 'United States' group are:

ID	FIRST_NAME	LAST_NAME	PHONE	ZIP_CODE	EMAIL
479	ZACHARY	HITE	191958435142	88749	ZACHARY.HITE@SACRAMENTO.COM
305	RICHARD	MCCRARY	262088367001	42141	RICHARD.MCCRARY@SACRAMENTO.COM
96	DIANA	ALEXANDER	6171054059	30695	DIANA.ALEXANDER@SACRAMENTO.COM
330	SCOTT	SHELLEY	165450987037	91590	SCOTT.SHELLEY@SACRAMENTO.COM
537	CLINTON	BUFORD	484500282381	79814	CLINTON.BUFORD@SACRAMENTO.COM
212	WILMA	RICHARDS	168758068397	25053	WILMA.RICHARDS@SACRAMENTO.COM
149	VALERIE	BLACK	885899703621	25545	VALERIE.BLACK@SACRAMENTO.COM
526	KARL	SEAL	214756839122	31342	KARL.SEAL@SACRAMENTO.COM
14	BETTY	WHITE	517338314235	16266	BETTY.WHITE@SACRAMENTO.COM

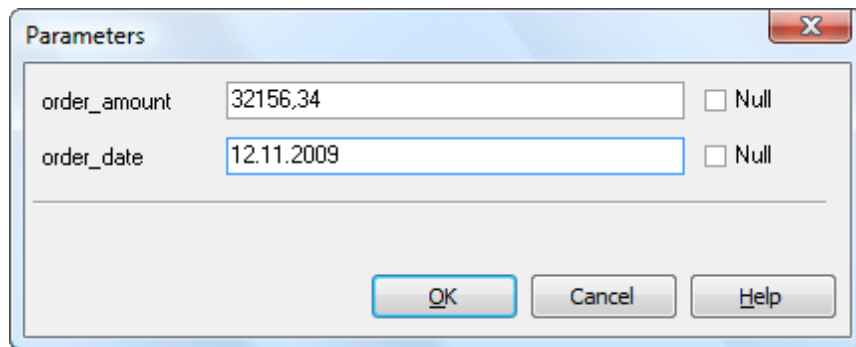
At the bottom of the results pane, it states 'Records fetched: 584'. An information box at the bottom indicates '584 rows fetched (0.20 sec)'.

The status bar at the bottom of the application shows 'Database: sakila at localhost'.

5.1.3 Query Parameters

Both [SQL Editor](#)^[44] and [Visual Query Builder](#)^[49] 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](#)^[115] tab of PostgreSQL Code Factory Options.

5.2 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.

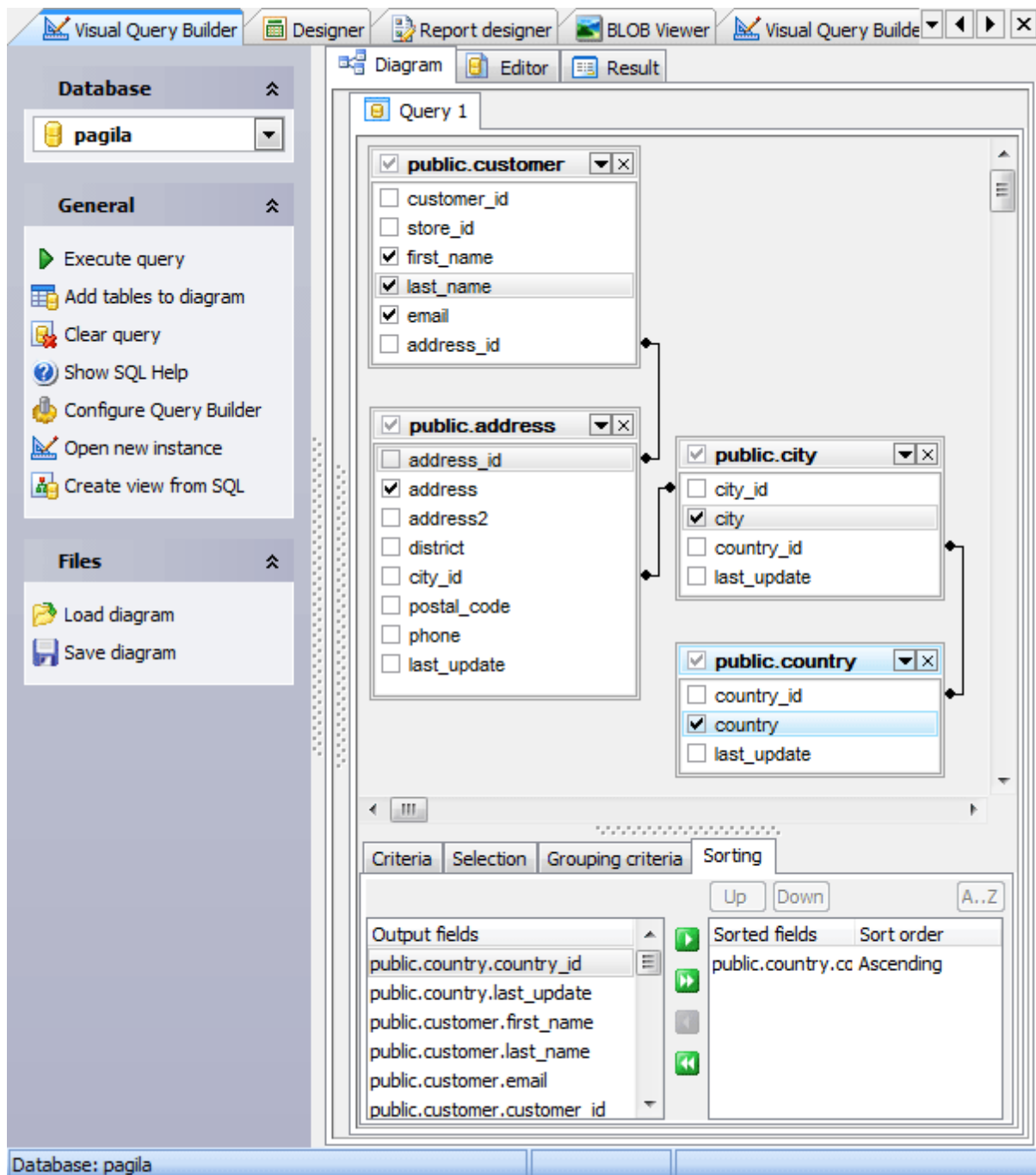
The most popular query management actions (creating, editing, deleting) are covered by the corresponding [topic](#)^[42].

Builder consists of 3 tabs:

- [Diagram](#)^[50] - to create a query from a graphical interface,
- [Editor](#)^[56] - to modify the query text before its executing,
- [Result](#)^[56] (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.

See also: [SQL Editor](#)^[44], [Visual Query Builder Options](#)^[119], [Query Parameters](#)^[48]



5.2.1 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.**

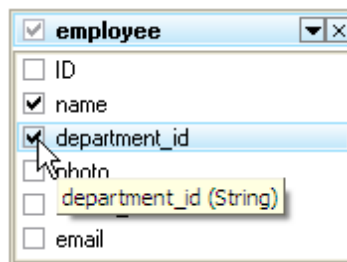
Use the [Add Table\(s\)](#) link of the area popup menu and select tables from the opened window (Use **Ctrl** or **Shift** pressed to select several tables).

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.

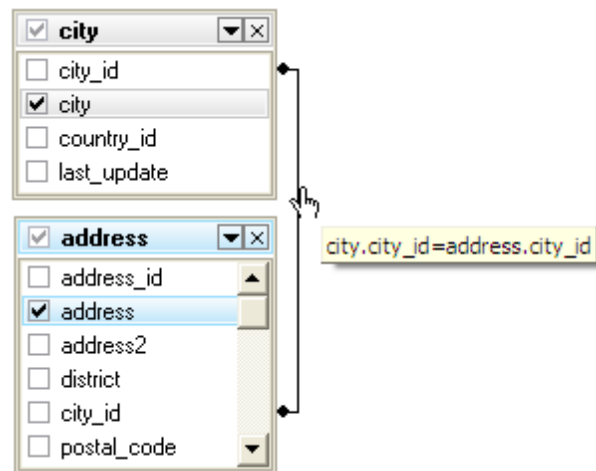


To include all the table fields, tick off the option box to the left of the table caption. In case none fields is included, the SQL statement is generated as `SELECT * FROM <Table_Name>`, i.e. all the fields are selected.

To remove the fields from the query, uncheck the corresponding boxes.

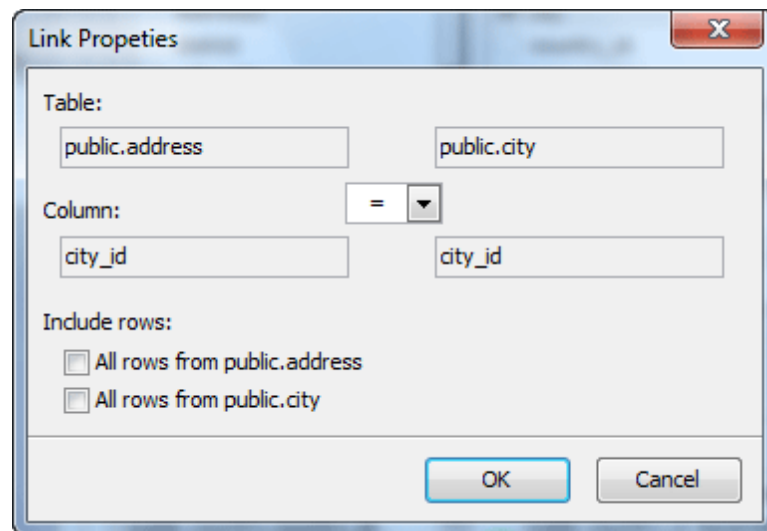
■ **Join tables if necessary**

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.

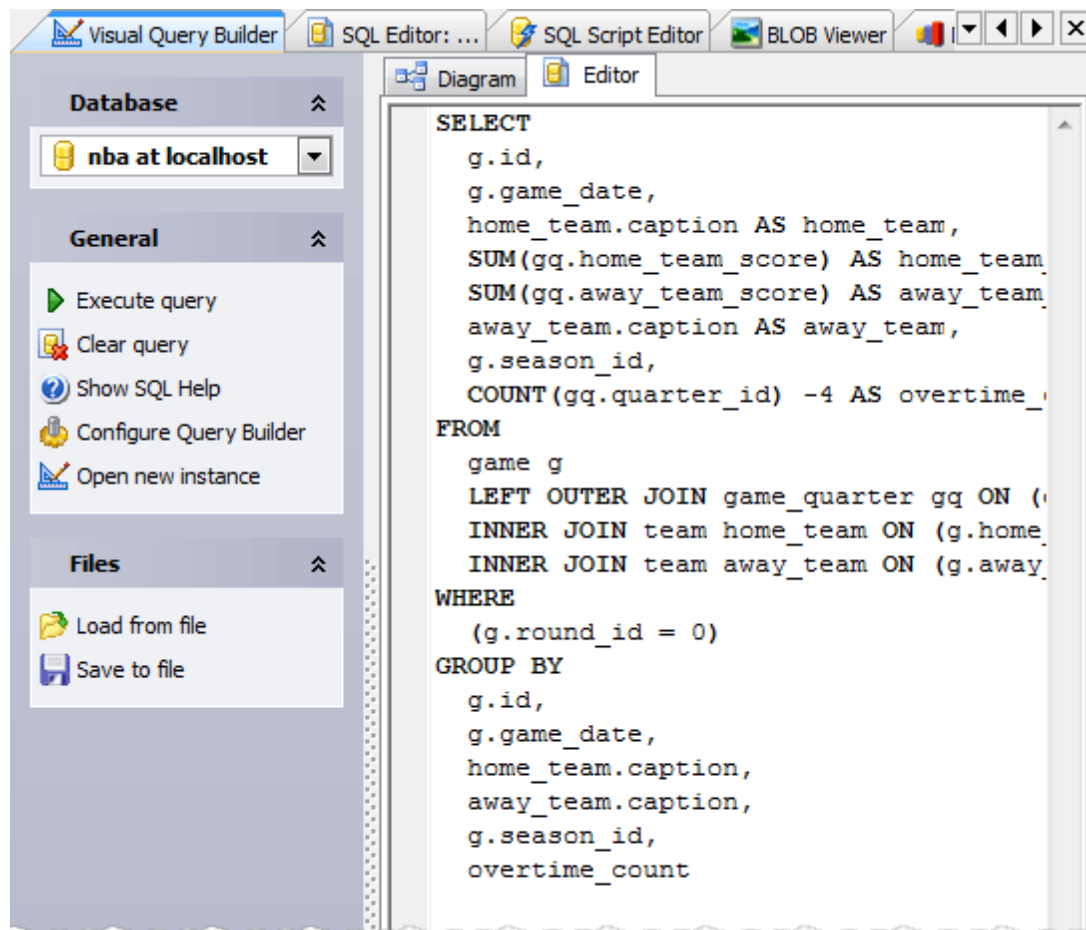
Note: The column names from the first `SELECT` statement are used as the column names for the results returned.

Selected columns listed in corresponding positions of each `SELECT` statement should have the same data type.

5.2.2 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](#).



5.2.3 Executing query

To execute the query select the [Execute](#) item in the navigation bar. After that the [Result](#) tab is displayed. This page contains the result data returned by the query, as a grid (see [Data View](#) for details). The popup menu of this tab and the items of the navigation bar allow you to export data and get SQL dump.

Visual Query Builder Designer Report designer BLOB Viewer Visual Query Builder

Database: pagila

General

- Show SQL Help
- Configure Query Builder
- Open new instance
- Create view from SQL

Data Management

- Export data
- Get SQL dump
- Print data

Diagram Editor Result

Table

* country

	* first_name	* last_name	email
	NULL	NULL	NULL
country : United Kingdom (9)			
1	ANNE	POWELL	ANNE.POWELL@sakilacustomer.org
2	APRIL	BURNS	APRIL.BURNS@sakilacustomer.org
3	ARMANDO	GRUBER	ARMANDO.GRUBER@sakilacustomer.org
4	CECIL	VINES	CECIL.VINES@sakilacustomer.org
5	DAN	PAINE	DAN.PAINE@sakilacustomer.org
6	GILBERT	SLEDGE	GILBERT.SLEDGE@sakilacustomer.org
7	MARSHALL	THORN	MARSHALL.THORN@sakilacustomer.org
8	MATTIE	HOFFMAN	MATTIE.HOFFMAN@sakilacustomer.org
9	SANDRA	MARTIN	SANDRA.MARTIN@sakilacustomer.org
country : United States (36)			
1	ALICE	STEWART	ALICE.STEWART@sakilacustomer.org
2	ANA	BRADLEY	ANA.BRADLEY@sakilacustomer.org
3	ASHLEY	RICHARDSON	ASHLEY.RICHARDSON@sakilacustomer.org
4	BETTY	WHITE	BETTY.WHITE@sakilacustomer.org
5	BILL	GAVIN	BILL.GAVIN@sakilacustomer.org
6	BRANDY	GRAVES	BRANDY.GRAVES@sakilacustomer.org
7	BRYAN	HARDISON	BRYAN.HARDISON@sakilacustomer.org
8	CAROLE	BARNETT	CAROLE.BARNETT@sakilacustomer.org
9	CAROLINE	BOWMAN	CAROLINE.BOWMAN@sakilacustomer.org
10	CASSANDRA	WALTERS	CASSANDRA.WALTERS@sakilacustomer.org
11	CLINTON	BUFORD	CLINTON.BUFORD@sakilacustomer.org
12	DIANA	ALEXANDER	DIANA.ALEXANDER@sakilacustomer.org
13	EVA	RAMOS	EVA.RAMOS@sakilacustomer.org
14	IAN	STILL	IAN.STILL@sakilacustomer.org


Records fetched: 599

Information

599 rows fetched (0,19 sec)

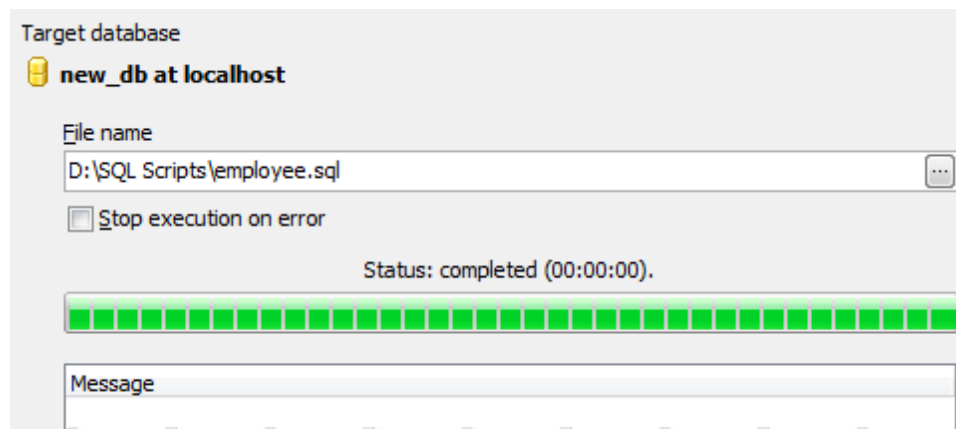
Database: pagila

5.3 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** .

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.



5.4 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](#)⁴⁴. 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.

The screenshot shows the PostgreSQL Code Factory Help interface. The sidebar on the left contains several sections:

- Database:** sakila at localhost
- General:** Execute script, Execute selected only, Execute script from file
- Script management:** Create new script, Configure SQL Script Editor, Open new instance
- Files:** Load script from file, Save current changes, Save script as new file
- Explorer:** Databases, Users, UDFs, Remote servers, **Tables (5)** (expanded), Views, Procedures, Functions, Events, **Indexes (2)**

The main area displays the following SQL code:

```
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 ( ...
```

A tooltip is visible over the CITY table definition, showing the same code block.

At the bottom left, the status bar shows "44: 1".

6 Data management

Query results are displayed on the **Result** tabs of [SQL Editor](#)^[44] or [Visual Query Builder](#)^[49].

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](#)^[62] 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](#)^[78] allows you to export data to various formats, including MS Excel, MS Access, RTF, HTML, PDF and more.
- [Get SQL Dump](#)^[85] exports data to the SQL script as a number of INSERT statements.
- [Import Data](#)^[88] provides you with possibility to import data from MS Excel, MS Access, DBF, XML, TXT, and CSV.
- [Edit BLOB](#)^[73] allows you to view and edit the content of BLOB and TEXT fields.

6.1 Data View

PostgreSQL Code Factory represents all data (stored in tables and views, results of queries and procedures) in [grid](#)^[68] or in [info cards](#)^[68]. 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	ADDRESS
1	1001	Signature Design	Dale J.	Little	(619) 530-2710	151
2	1002	Dallas Technologies	Glen	Brown	(214) 960-2233	P. 4
3	1003	Buttle, Griffith and Co.	James	Buttle	(617) 488-1864	230
4	1004	Central Bank	Elizabeth	Brocket	61 211 99 88	66
5	1005	DT Systems, LTD.	Tai	Wu	(852) 850 43 98	400
6	1006	DataServe International	Tomas	Bright	(613) 229 3323	200
7	1007	Mrs. Beauvais	NULL	Mrs. Beauvais	NULL	P.C
8	1008	Anini Vacation Rentals	Leilani	Briggs	(808) 835-7605	330
9	1009	Max	Max	NULL	22 01 23	1 E
10	1010	MPM Corporation	Miwako	Miyamoto	3 880 77 19	2-6
11	1011	Dynamic Intelligence Corp	Victor	Granges	01 221 16 50	Flo
12	1012	3D-Pad Corp.	Michelle	Roche	1 43 60 61	22
13	1013	Lorenzi Export, Ltd.	Andreas	Lorenzi	02 404 6284	Via

Navigation buttons

Both data representations are equipped with navigation buttons. They are represented at the top of the data tab and allow you to navigate between records and to accomplish common operations:

- To add a new record, use the *Plus* button or the **Insert** shortcut.
- To delete a new record, use the *Minus* button or the **Delete** shortcut.
- To edit an existing record, push the corresponding button or invoke the [Data Input Form](#)^[69] 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](#)^[123].

Navigation bar

The Data management group of the Navigation bar allows to invoke [Data Input Form](#)^[69], [SQL Editor](#)^[44] with SELECT query, [Data Export](#)^[78], and [Data Import](#)^[88] modules using corresponding links, also get [SQLdump](#)^[85] of the current data set and print current data with enabled preview in WYSIWYG mode.

See also: [SQL Editor](#)^[44], [Visual Query Builder](#)^[49]

6.1.1 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)

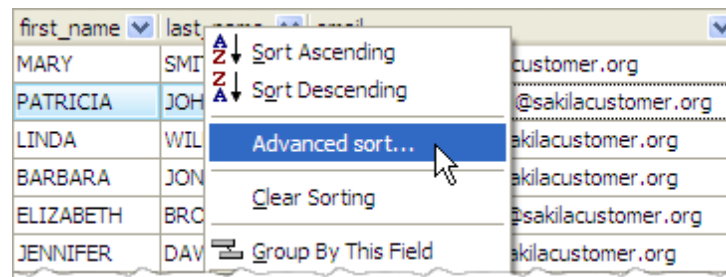
By default, the grid automatically switches to the simple mode for queries returning more than 5000 records (the number can be customized in the [Options](#) ¹²³ dialog).

The following abilities are not available in the simple grid mode:

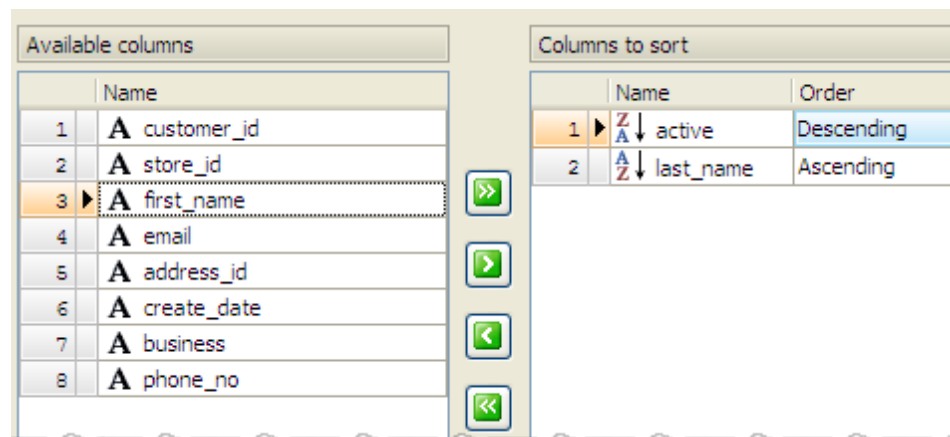
■ **Sorting data (only in the full grid mode)**

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.



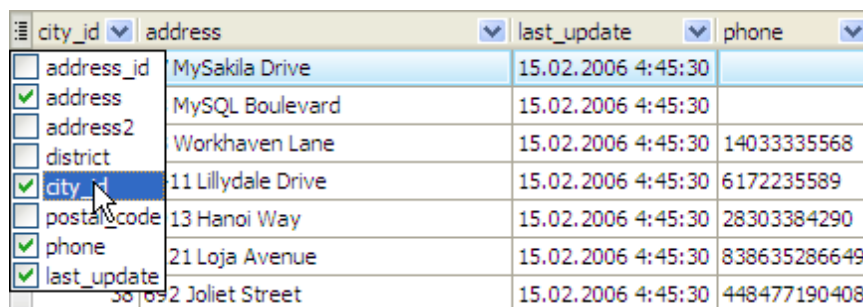
To cancel the sorting order, press **Ctrl** and click on the sorted column caption.

■ Filtering represented records (only in the full grid mode)

There are several ways to filter data represented in grid. See [the corresponding topic](#) to find out their descriptions.

■ Hiding selected columns

You can show/hide columns using a button in the left top corner of the grid. Just check/uncheck the column in the drop-down list.



■ Columns reordering

To reorder columns, use drag-n-drop.

■ Grouping records

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

The screenshot shows the PostgreSQL Code Factory interface with a data grid. The grid is grouped by 'round' and 'date'. The 'round' column is expanded to show 'round : 3', which is further expanded to show 'date : 25.08.2004'. The data rows are grouped under these dates. The bottom status bar shows 'Records fetched: 380' and an 'Information' popup shows '380 rows fetched (0,64 sec)'.

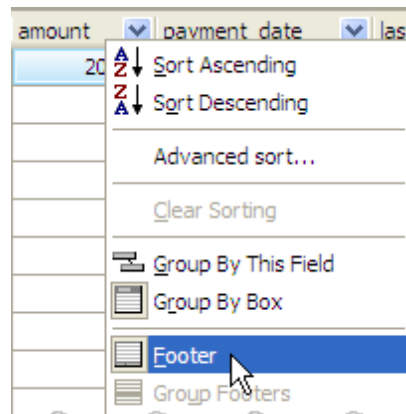
ID	team1ID	team2ID	score1	score2	refereeID	comments
round : 1						
round : 2						
round : 3						
date : 24.08.2004						
date : 25.08.2004						
24	6	2	3	0	8	Jeffers brukte 34 minutter på å vinne
22	1	4	3	0	17	I den hundrede kampen i alle konkurranse
29	16	5	1	2	19	Southamptons slapp inn mål i sin 11. li
27	9	13	0	2	0	Fulhams første tap denne sesongen,
28	14	18	2	2	18	Newcastle skuffer i årets Premier League
26	19	17	1	1	12	Det ble uavgjort på Hawthornes etter
date : 30.08.2004						
31	12	8	0	0	1	Igjen skuffet Manchester United mot
date : 14.12.2004						
round : 4						
round : 5						
date : 11.09.2004						
date : 12.09.2004						
date : 13.09.2004						
round : 6						

Records fetched: 380

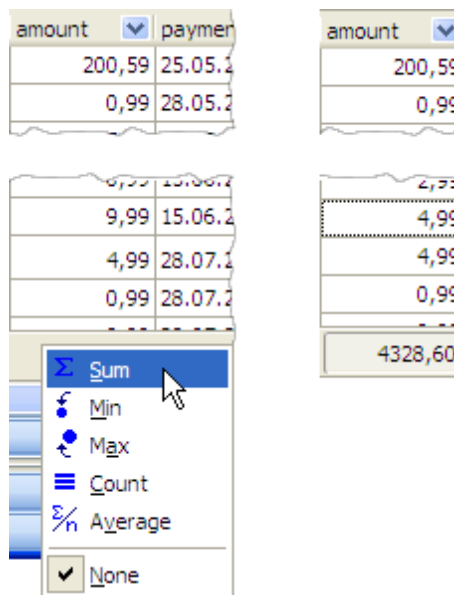
Information
380 rows fetched (0,64 sec)

■ Using aggregate functions

To get a sum of column values, a min or a max value, an average column value or an amount of records, use [Data Grid Footer](#). Select the [Footer](#) item at the grid caption's popup menu.



It will be shown at the bottom of the grid. The popup menu of the footer allows you to get an aggregate function result calculated with the corresponding column values.



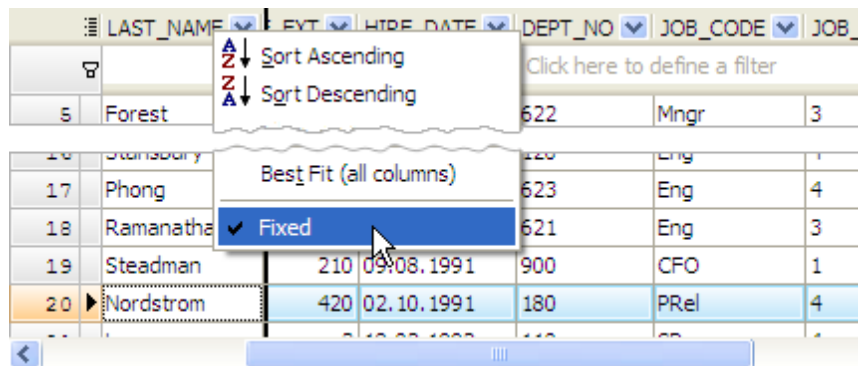
For grouped data use [Group Footers](#).

■ Data alignment

The grid's header popup menu allows to align column data. Use the [Alignment](#) link and select the alignment type.

■ Fixing columns

You can fix grid columns to view them permanently when working with other grid data. To fix a column, choose the corresponding item from the grid's header popup menu.

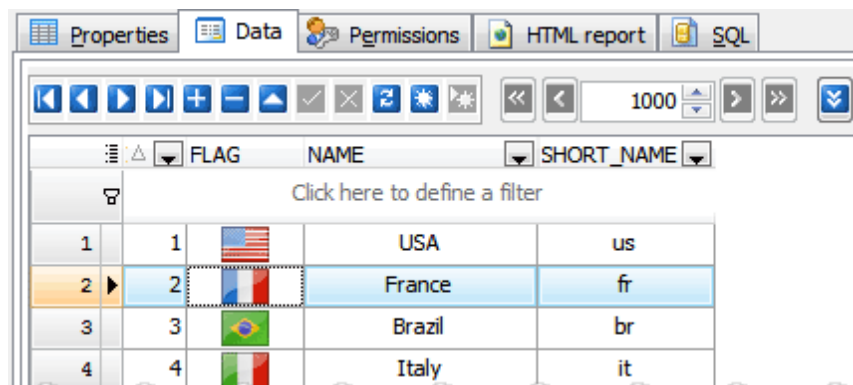


■ Row numbering

There is also a possibility to display row numbers in grids. You can [adjust](#)^[124] 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](#)^[73] (see below).

■ Working with BLOBs

To [edit a BLOB field](#)^[73], 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).

6.1.2 Working with info cards

Info cards correspond to the records. You can [filter records by custom conditions](#)^[70] and edit data directly in info cards or with [Data Input Form](#)^[69].

id:	
first_name:	Gilbert
last_name:	Arenas
career_start_year:	2001
career_end_year:	0
position_id:	6
photo:	
country_id:	1
height:	193
birthday:	06.01.1982
weight:	97,5
college_id:	15
current_team_id:	12
current_number:	0

id:	
first_name:	Hilton
last_name:	Armstrong
career_start_year:	2006
career_end_year:	0
position_id:	11
photo:	
country_id:	1
height:	211
birthday:	11.11.1984
weight:	106,6
college_id:	7
current_team_id:	27
current_number:	12

id:	
first_name:	Darrell
last_name:	Arthur
career_start_year:	
career_end_year:	
position_id:	
photo:	
country_id:	
height:	
birthday:	25.03.198
weight:	
college_id:	
current_team_id:	
current_number:	

id:	
first_name:	Trevor
last_name:	Ariza
career_start_year:	2004
career_end_year:	0
position_id:	10
photo:	
country_id:	1
height:	203
birthday:	30.06.1985
weight:	95,3
college_id:	2
current_team_id:	5
current_number:	3

id:	
first_name:	Ron
last_name:	Artest
career_start_year:	1999
career_end_year:	0
position_id:	10
photo:	
country_id:	1
height:	201
birthday:	13.11.1979
weight:	117,9
college_id:	16
current_team_id:	22
current_number:	96

id:	
first_name:	D.J.
last_name:	Augustin
career_start_year:	
career_end_year:	
position_id:	
photo:	
country_id:	
height:	
birthday:	10.11.198
weight:	
college_id:	
current_team_id:	
current_number:	

Records fetched: 67

6.1.3 Data input form

Use [Data Input Form](#) to add new records or edit existing ones. To invoke the dialog, use the corresponding link from the popup menu or **Ctrl+Alt+D** shortcut.

The screenshot shows a 'Data Input Form' dialog box with the following fields and values:

- payment_id: 5
- customer_id: 1
- staff_id: 2
- rental_id: 1,476
- amount: 9.990
- payment_date: 9/22/2006
- last_update: (calendar open for September 2006, 22nd selected)

Buttons on the right: Insert, Post, Cancel, Previous, Next, Close.

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.

6.1.4 Data filtering

PostgreSQL Code Factory 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).

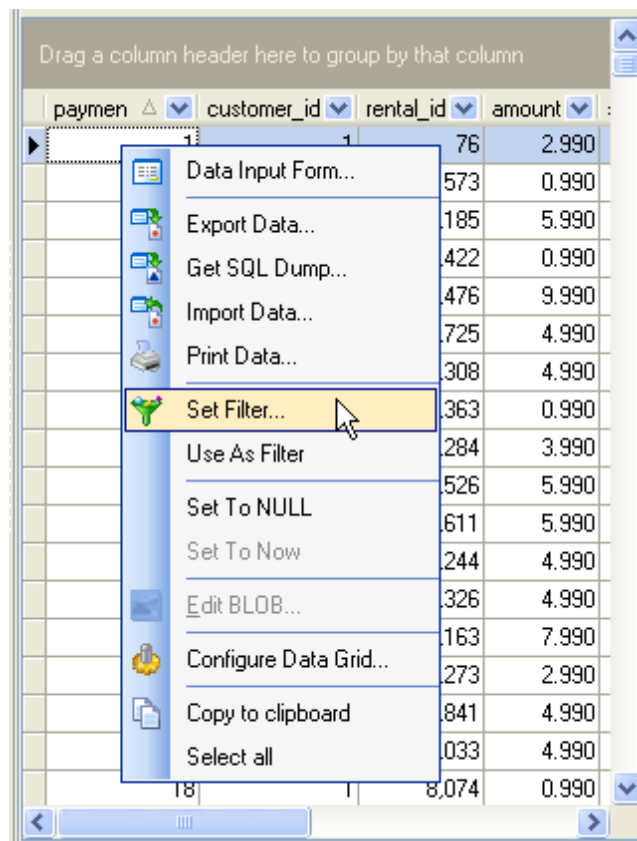
- **Filter by two operators**

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.

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](#)⁴⁴ and save as a query. You can also use [Generate query](#) link on the Navigation bar.

See also: [Data View](#)⁶², [SQL Editor](#)⁴⁴, [Visual Query Builder](#)⁴⁹

6.2 BLOB Editor

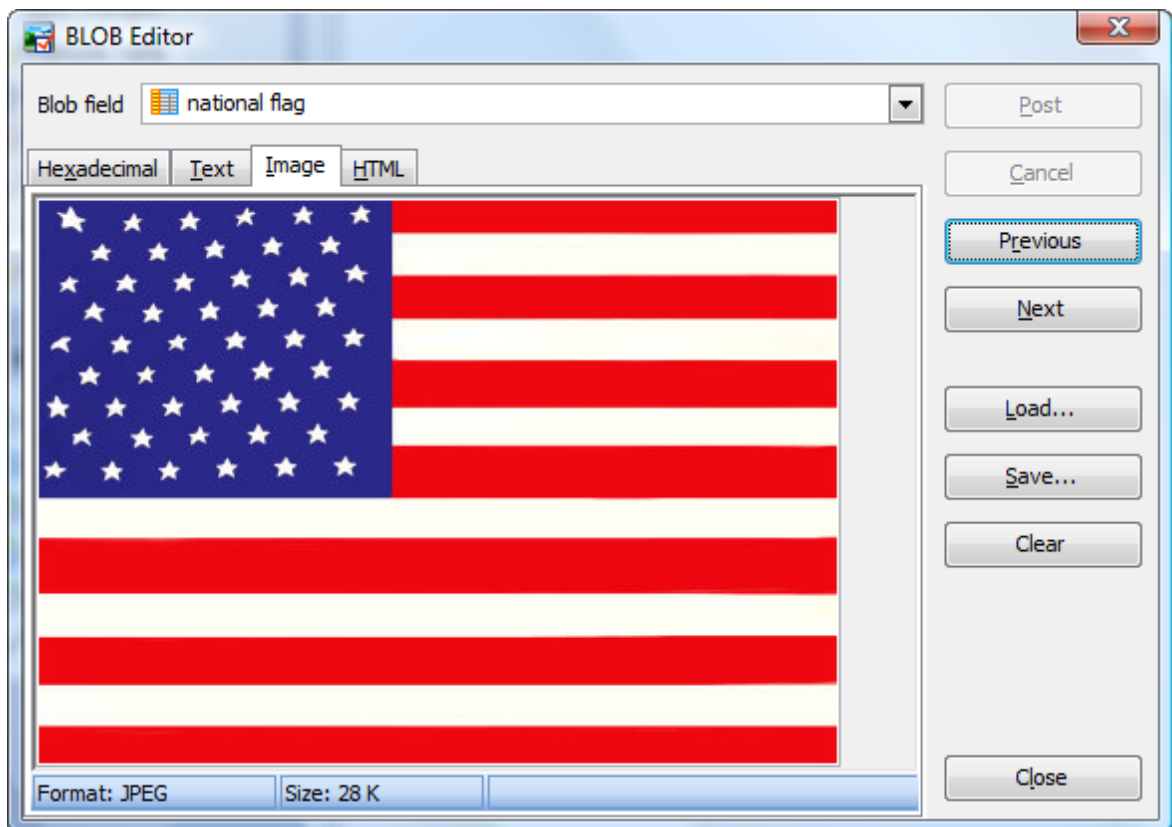
BLOB Editor is a tool to view and edit BLOB data in the following ways: [hexadecimal dump](#)^[74], [plain text](#)^[74], [graphical image](#)^[73], [HTML page](#)^[75], or [PDF document](#)^[76]. BLOB Editor is invoked from the result tab of [SQL Editor](#)^[44] and [Visual Query Builder](#)^[49] by 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](#)^[97] 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.

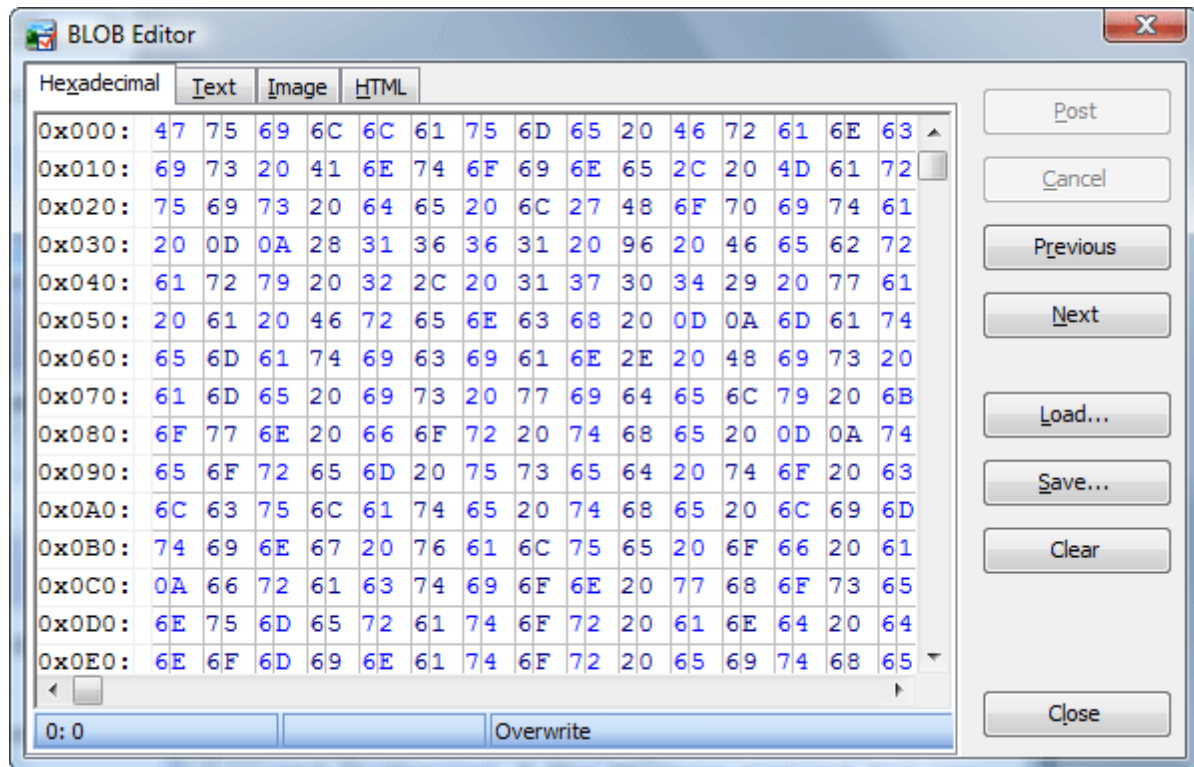
6.2.1 Editing as image

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.



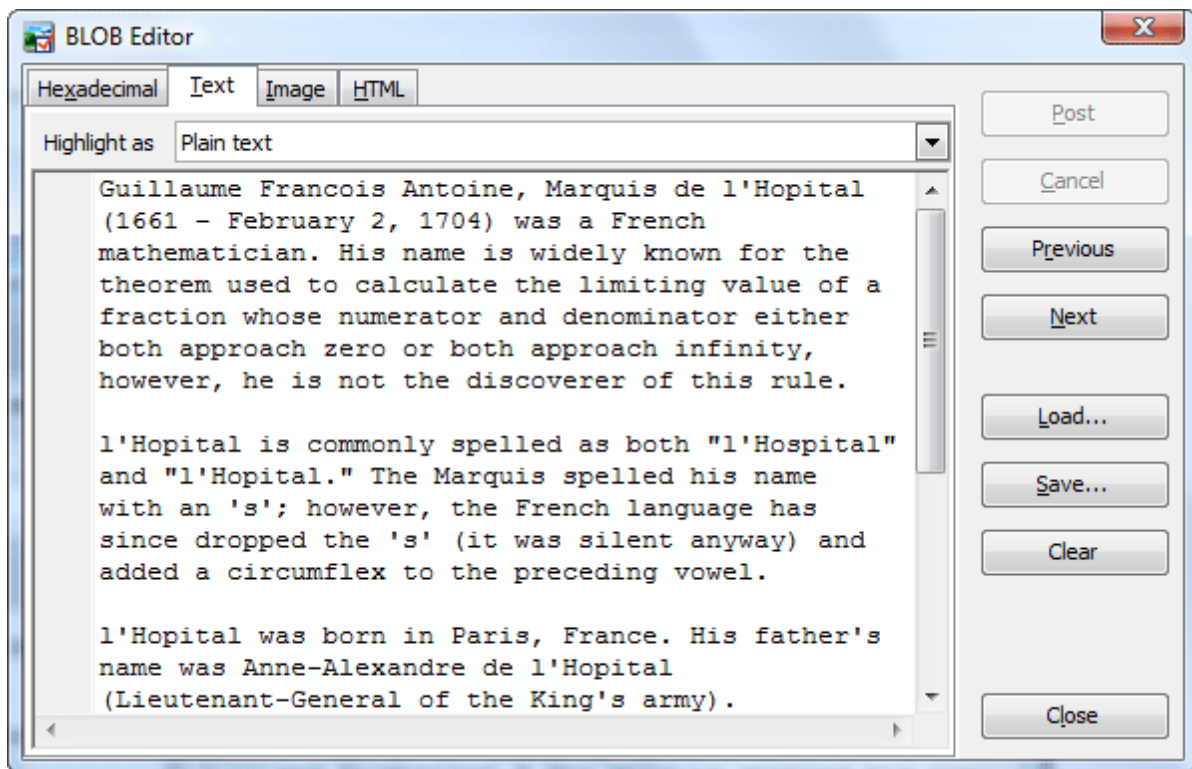
6.2.2 Editing as hexadecimal dump

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.



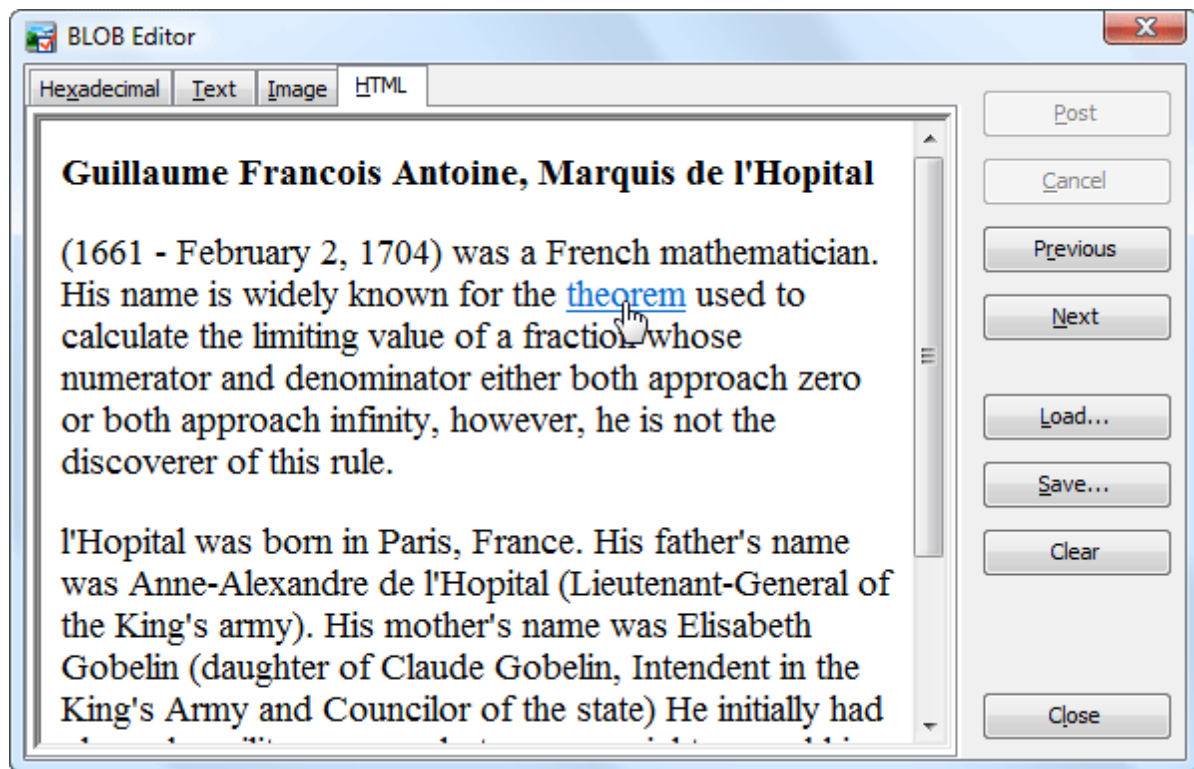
6.2.3 Editing as plain text

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.



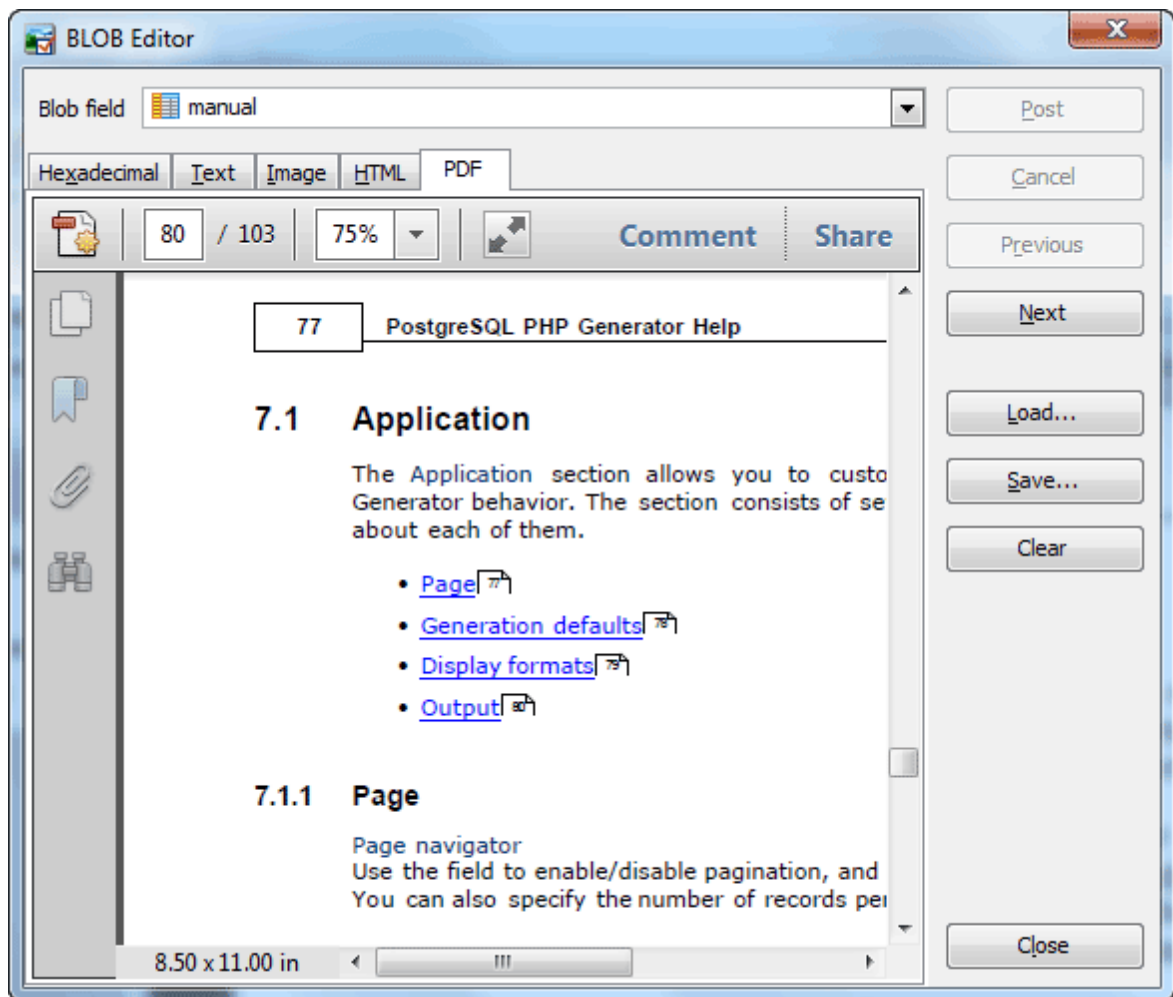
6.2.4 Editing as HTML

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.



6.2.5 Editing as PDF document

The [PDF](#) panel presents field data as PDF document. To accomplish common operations with data, use the Adobe Reader toolbar.



6.3 Export Data Wizard

Data Export wizard is a tool to save data from PostgreSQL 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

- 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](#)^[78] of the destination file;
- Specify such additional options of the result file as [header and footer](#)^[79], [formats applied to exported data](#)^[80] and [some format-specific options](#)^[81];
- [Select columns](#)^[80] you want to include into result files;
- [Specify other export options](#)^[84].

See also: Get SQL Dump, [Import Data Wizard](#)^[88]

6.3.1 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

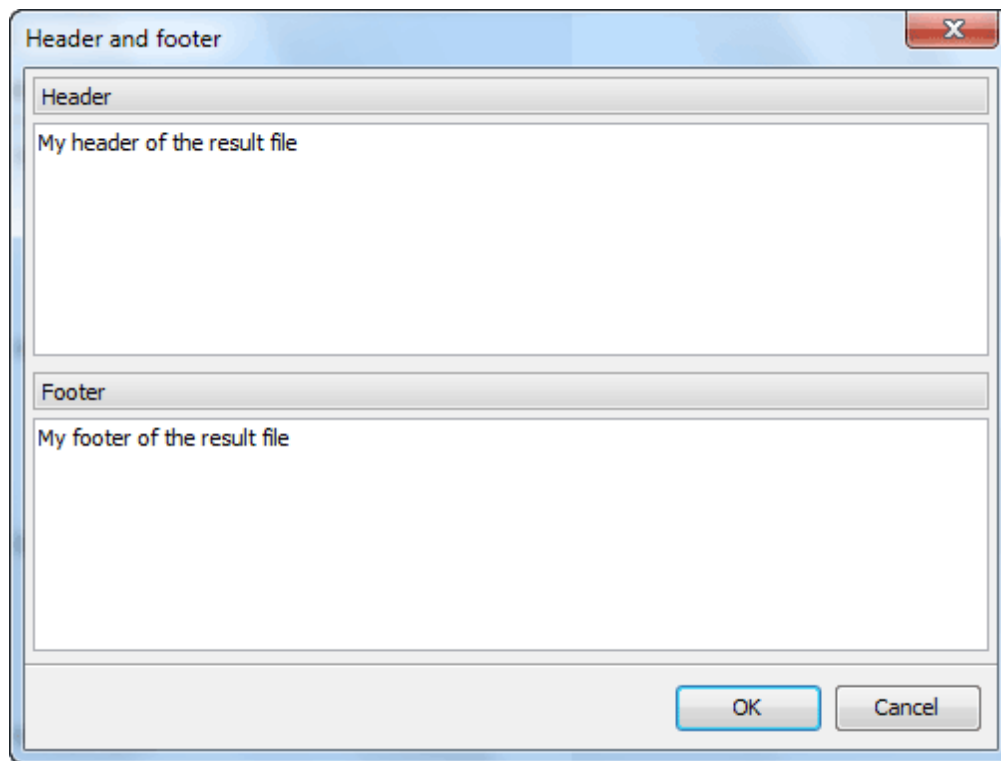
C:\Data\Excel\Customers.xls

Encoding

ANSI

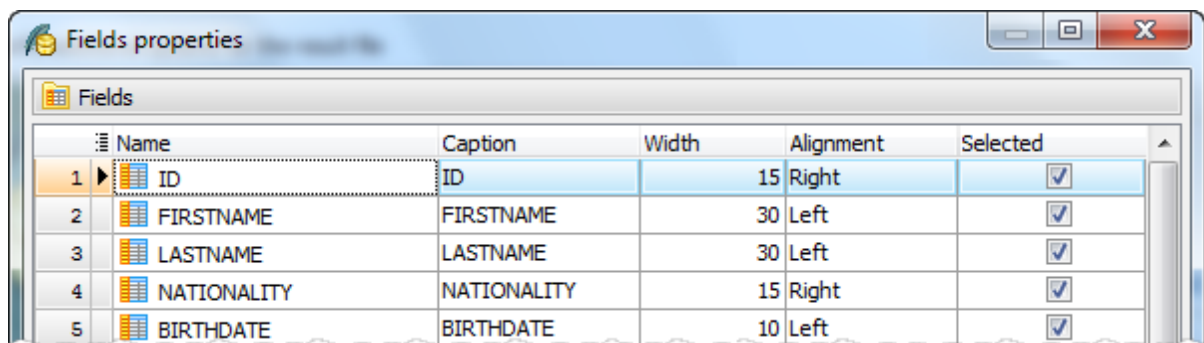
6.3.2 Setting header and footer

To specify the result file's header and footer, double click the corresponding button and complete fields of the [Header and Footer](#) window.



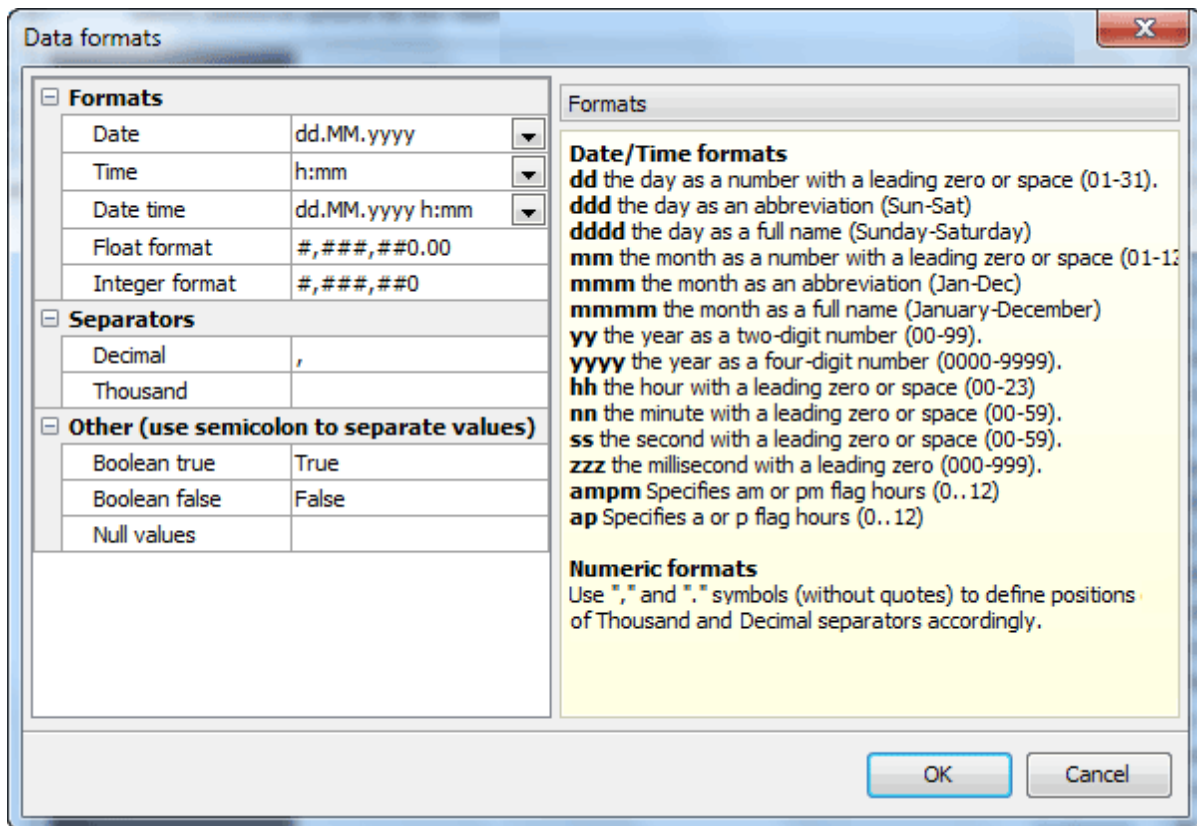
6.3.3 Selecting fields for export

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).



6.3.4 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.



6.3.5 Setting format-specific options

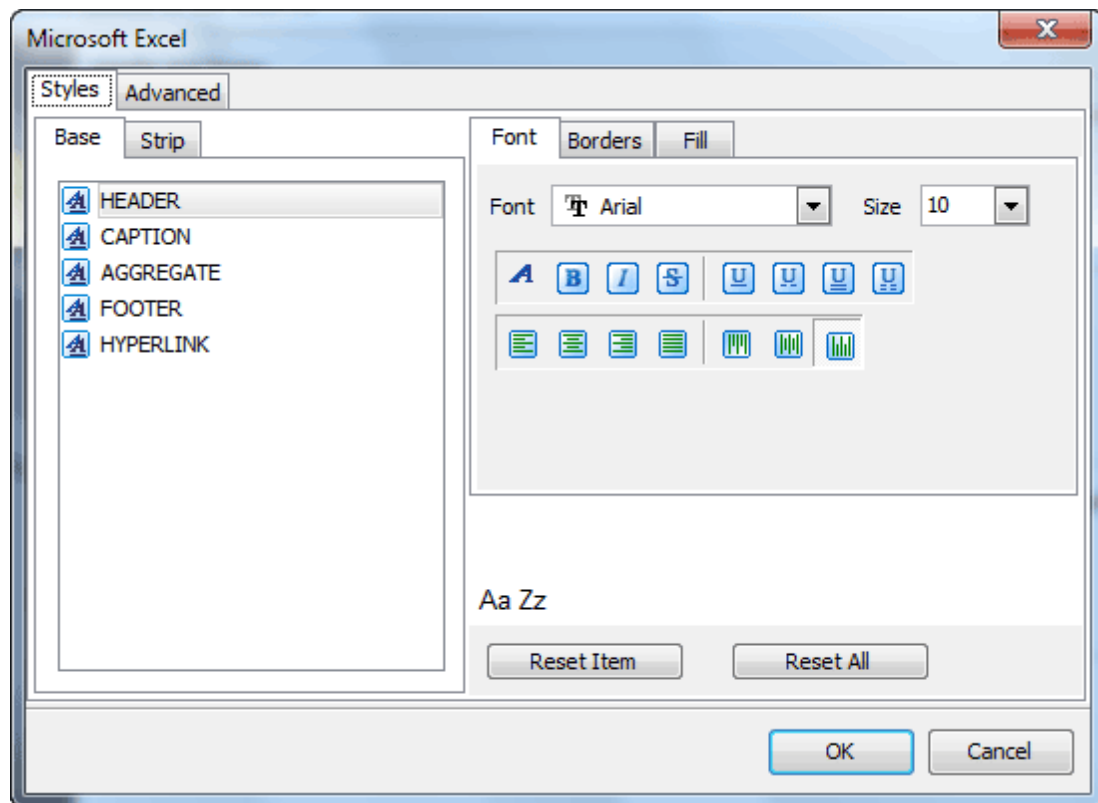
Each format supposes corresponding additional export options. Use the wizard option to adjust export properties depending on the target file format you have selected earlier. The following formats are at your disposal: [Microsoft Excel](#)^[81], Microsoft Excel 2007, [CSV](#)^[82], [Text](#)^[83], [HTML](#)^[82], [XML](#)^[83], 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.



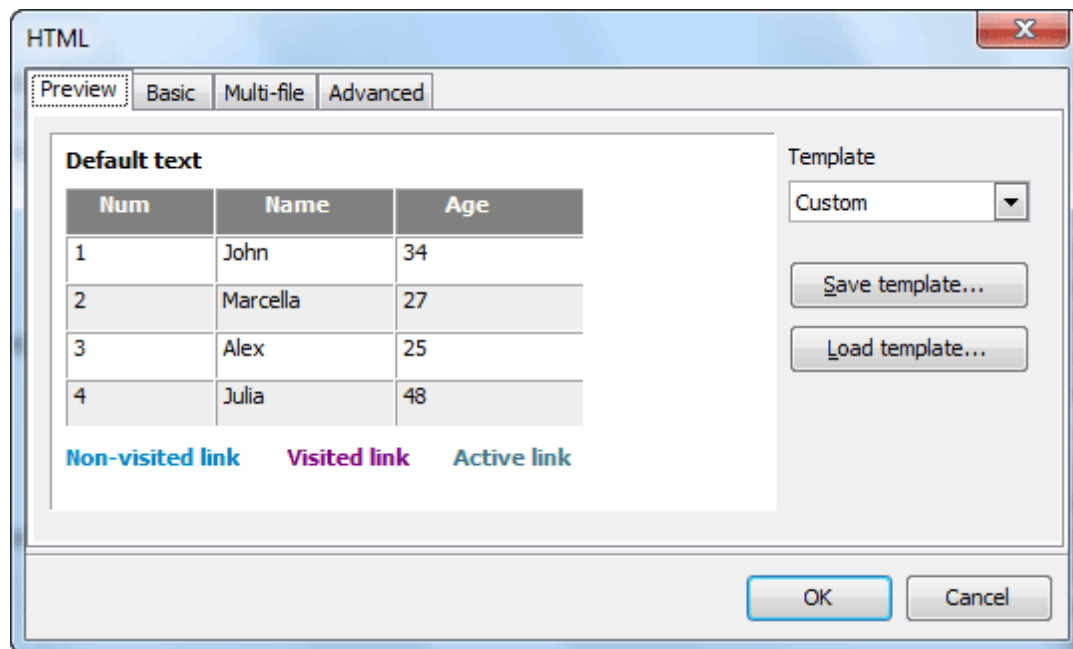
HTML

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.

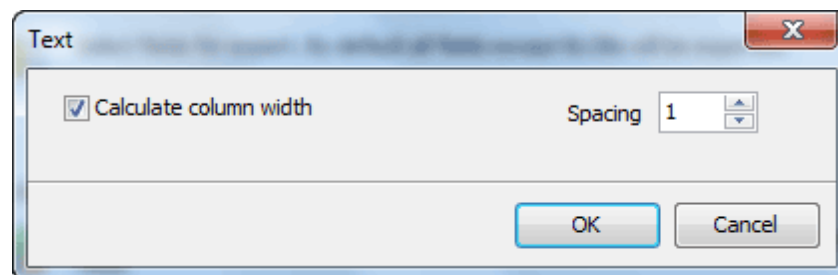
The [Multi-file](#) tab provides you with a possibility to split target HTML file into several separated files. This tab allows you to specify the record count for a single file, set an option to generate an index HTML file, and add an ability of navigation between each other to each of exported files.

The [Advanced](#) tab contains such HTML options as default font, background, cell padding and spacing, etc.



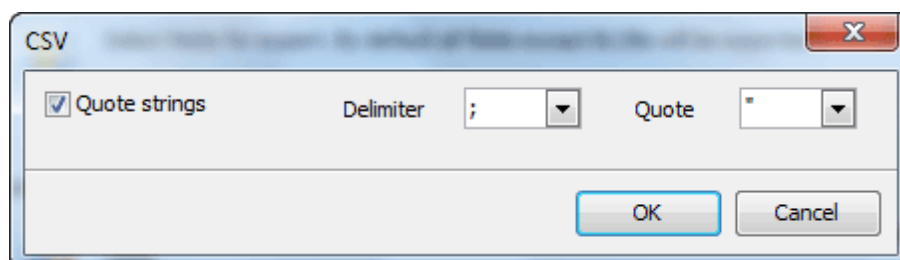
Text files

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.



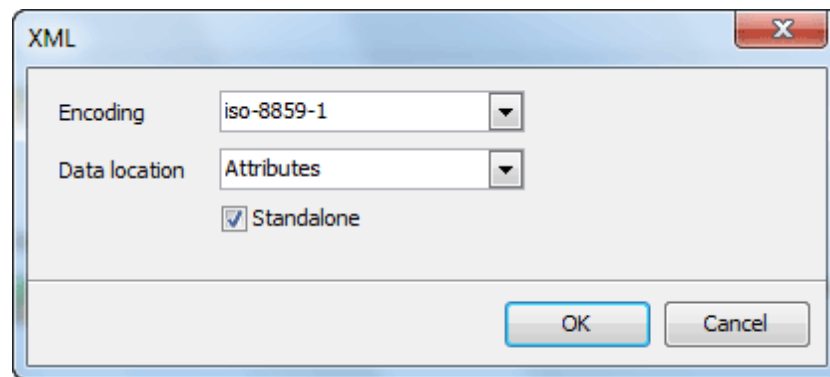
CSV files

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.



6.3.6 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.

A screenshot of a dialog box with two sections: 'Constraints' and 'Actions'. The 'Constraints' section has a document icon and the text 'Specify the number of records to export.' It contains two radio buttons: 'Export all records' (selected) and 'Export only first' (unselected). Next to 'Export only first' is a numeric spinner set to '100' followed by the text 'record(s)'. The 'Actions' section has a gear icon and the text 'Select actions to be executed after export.' It contains two checkboxes: 'Open file' (checked) and 'Print file' (unchecked).

6.4 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](#).

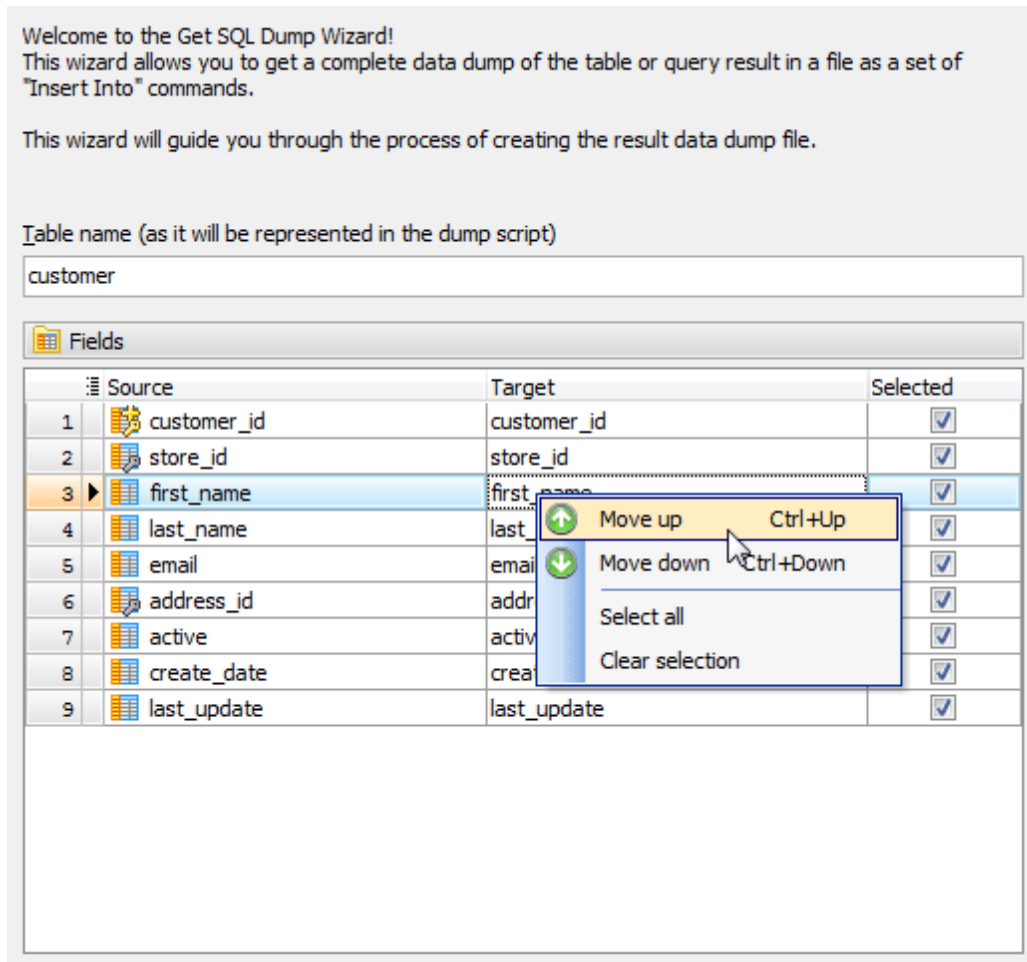
- [Selecting fields to include in the result INSERT statement](#) ⁸⁶
- [Specifying dump options](#) ⁸⁶

See also: [Export Data Wizard](#) ⁷⁸, [SQL Script Editor](#) ⁵⁹

6.4.1 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](#)^[29].



6.4.2 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](#)^[50] 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

  dump

Output

☒ Send to script editor

☐ Save to file

File name ...

Encoding

Click "Ready" to dump your data.

6.5 Import Data Wizard

[Import Data Wizard](#) provides you with a graphical user interface to import data from the most popular files formats into existing PostgreSQL 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

- select the node of the table for importing at the [Explorer tree](#);
- select the [Data Management](#) group of the node's popup menu;
- use the Import [Data](#) item.

To import data,

- [Set the format](#)^[88] of the input data and the source file name;
- [Map source file columns and target table fields](#)^[91];
- [Specify other import options](#)^[94].

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](#)

6.5.1 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:

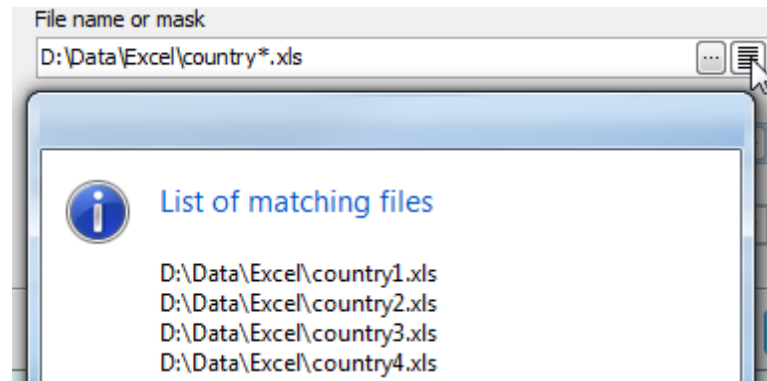
D:\Data\Excel\country1.xls

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>
```


6.5.2 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](#)  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](#)  used for the input file.

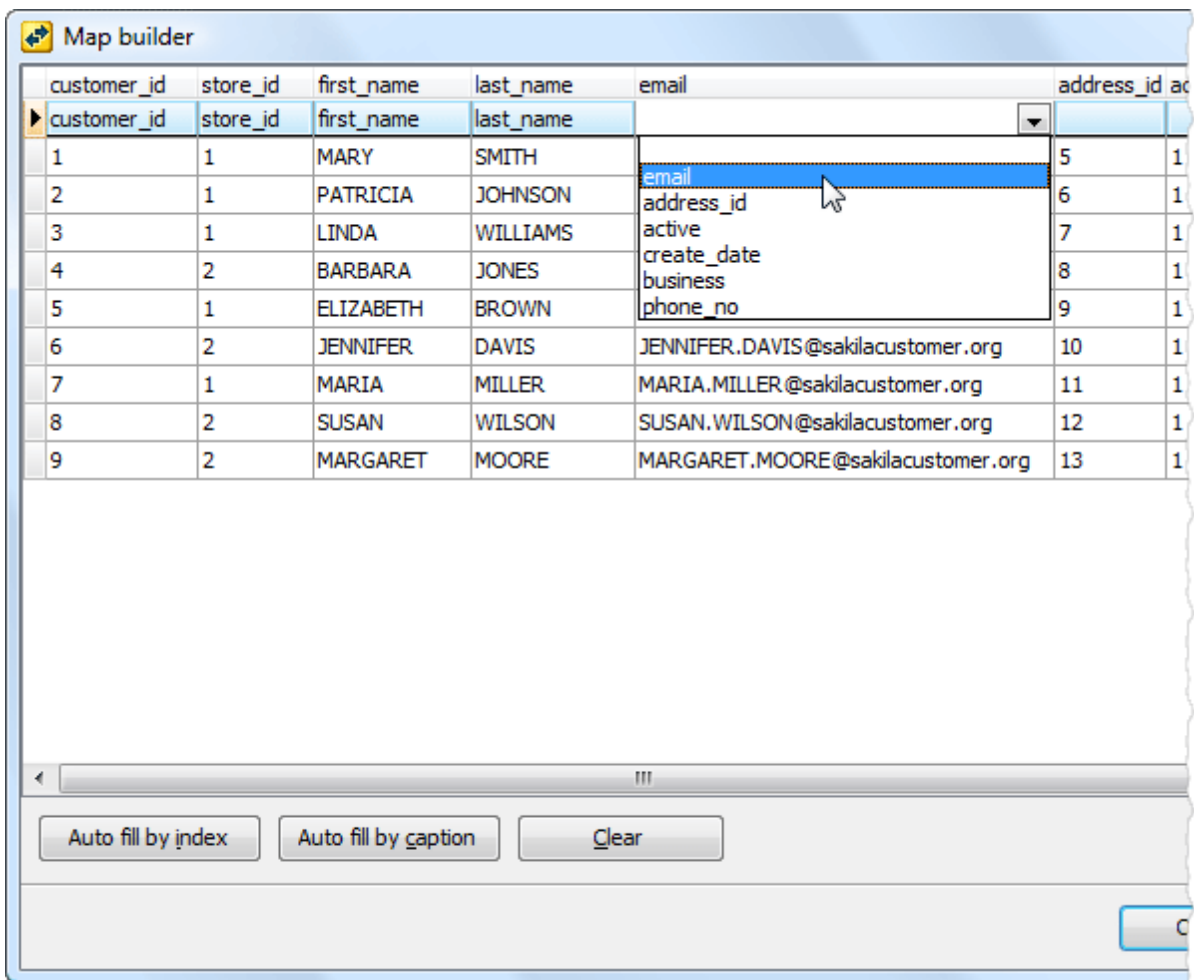
To exclude the first file row, use the [File contains column header](#) checkbox.

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

6.5.2.1 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.

	Target field	Offset	Width
1	customer_id	0	5
2	store_id	5	6
3	first_name	11	46
4	last_name	57	46
5	email	103	52
6	address_id	155	6
7	active	161	6
8	create_date	167	6

	custo	store	first_name
1	1		MARY
2	1		PATRICIA
3	1		LINDA
4	2		BARBARA
5	1		ELIZABETH
6	2		JENNIFER
7	1		MARIA

6.5.2.2 Data formats

Use the window fields to indicate format masks of the source data imported to the table. It allows the application to import data correctly.

The components of the date time format mask are represented at the window. Compose

your date, time, and date time format mask of this components and separators. The following table contains some types of input fields and suggests masks to import them.

To import these input data correctly	Use these format masks
June 29	mmm dd
Jun 29, 2009	mmm dd, yyyy
Tue Jun 14 16:50:49	ddd mmm 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.

<div> <div>Formats</div> <table> <tr><td>Date</td><td></td></tr> <tr><td>Time</td><td></td></tr> <tr><td>Date time</td><td></td></tr> </table> <div> <div>Separators</div> <table> <tr><td>Decimal</td><td>,</td></tr> <tr><td>Thousand</td><td>#160</td></tr> </table> <div> <div>Other (use semicolon to separate values)</div> <table> <tr><td>Boolean true</td><td>True</td></tr> <tr><td>Boolean false</td><td>False</td></tr> <tr><td>Null values</td><td>;NULL</td></tr> </table> </div> </div> </div>		Date		Time		Date time		Decimal	,	Thousand	#160	Boolean true	True	Boolean false	False	Null values	;NULL	<div>Date time formats</div> <div> dd the day as a number with a leading zero or space (01-31). ddd the day as an abbreviation (Sun-Sat) dddd the day as a full name (Sunday-Saturday) mm the month as a number with a leading zero or space (01-12). mmm the month as an abbreviation (Jan-Dec) mmm the month as a full name (January-December) yy the year as a two-digit number (00-99). yyyy the year as a four-digit number (0000-9999). hh the hour with a leading zero or space (00-23) nn the minute with a leading zero or space (00-59). ss the second with a leading zero or space (00-59). zzz the millisecond with a leading zero (000-999). ampm Specifies am or pm flag hours (0..12) ap Specifies a or p flag hours (0..12) </div>
Date																		
Time																		
Date time																		
Decimal	,																	
Thousand	#160																	
Boolean true	True																	
Boolean false	False																	
Null values	;NULL																	

6.5.3 Customizing common options

On the wizard step you can set the number of records to import, whether the tool import all table records or only the specified number. In the second case you can set the number of records to skip.

Logging

This options group let you to manage logging of the import process.

Scripts

There are many cases where the import process is necessary to correct with additional scripts. So to disable table indexes before the importing, specify the corresponding scripts to be executed before and after the process.

The typical example of usage of the [Before each table](#) and [After each table](#) scripts is the import data to autoincrement columns of several tables. In this case it's necessary to set the corresponding scripts:

```
SET IDENTITY_INSERT %table_name% ON
```

and

```
SET IDENTITY_INSERT %table_name% OFF
```

to be executed before and after import data to each table correspondingly.

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 the [COPY FROM](#) command 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.

7 Database Tools

PostgreSQL Code Factory provides a number of powerful tools for working with databases.

The following tools are available:

- [SQL Editor](#) ⁴⁴
Creates and executes SQL queries.
- [Visual Query Builder](#) ⁴⁹
Builds queries visually.
- [SQL Script Editor](#) ⁵⁹
Executes SQL scripts to the database.
- [BLOB Viewer](#) ⁹⁷
Displays a content of BLOB fields in different representations.
- [Diagram Viewer](#) ¹⁰³
Represents data from a table or a query as a diagram in various ways.
- [SQL Generator](#) ¹⁰⁷
Provides you with a set of simple SQL statements.

7.1 BLOB Viewer

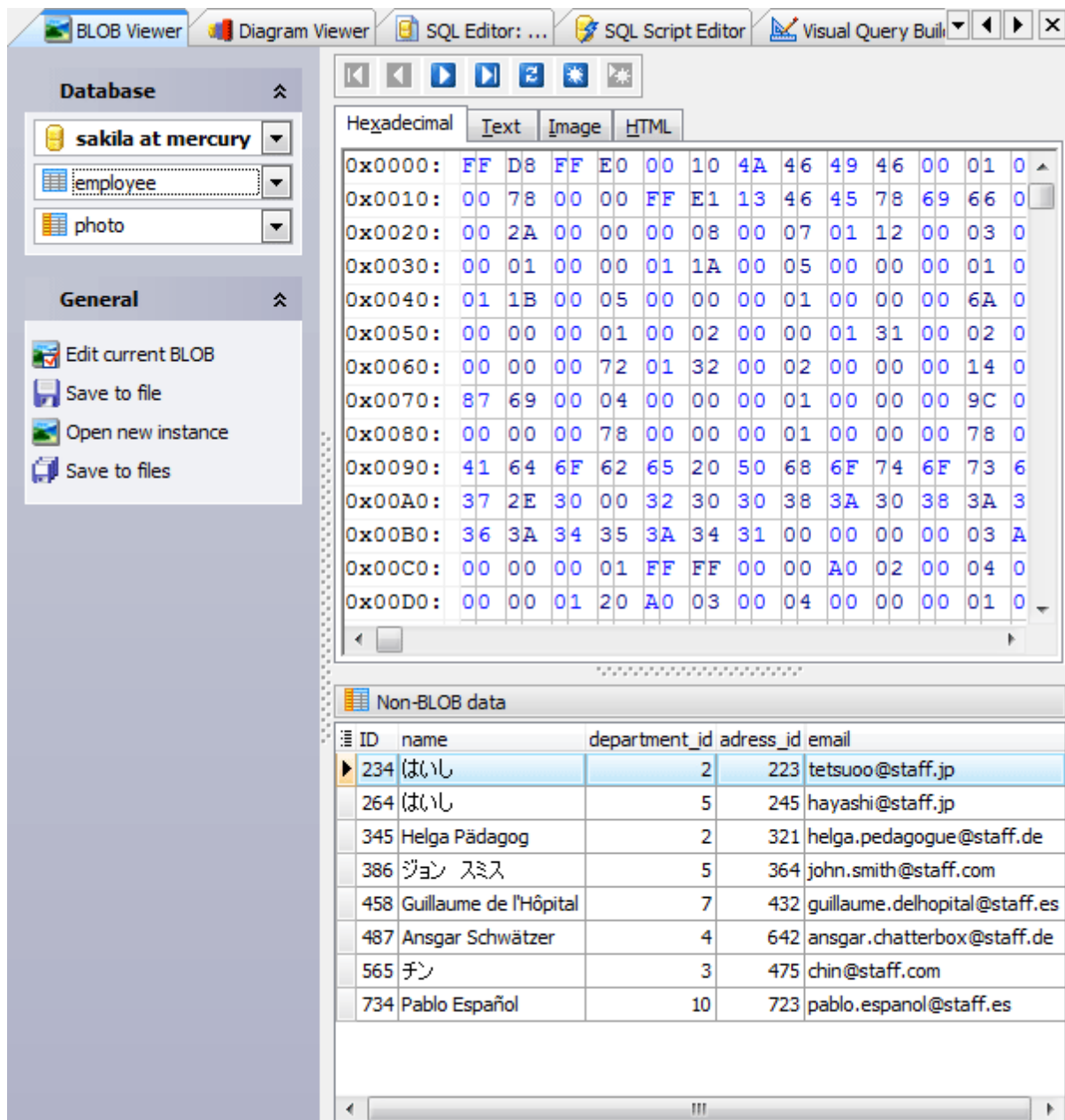
BLOB Viewer allows you to view the content of the BLOB fields in various representations.

- [Viewing BLOB field as hexadecimal dump](#)^[97]
- [Viewing BLOB field as plain text](#)^[98]
- [Viewing BLOB field as graphical image](#)^[99]
- [Viewing BLOB field as HTML](#)^[100]
- [Viewing BLOB field as PDF](#)^[101]

See also: [BLOB Editor](#)^[73]

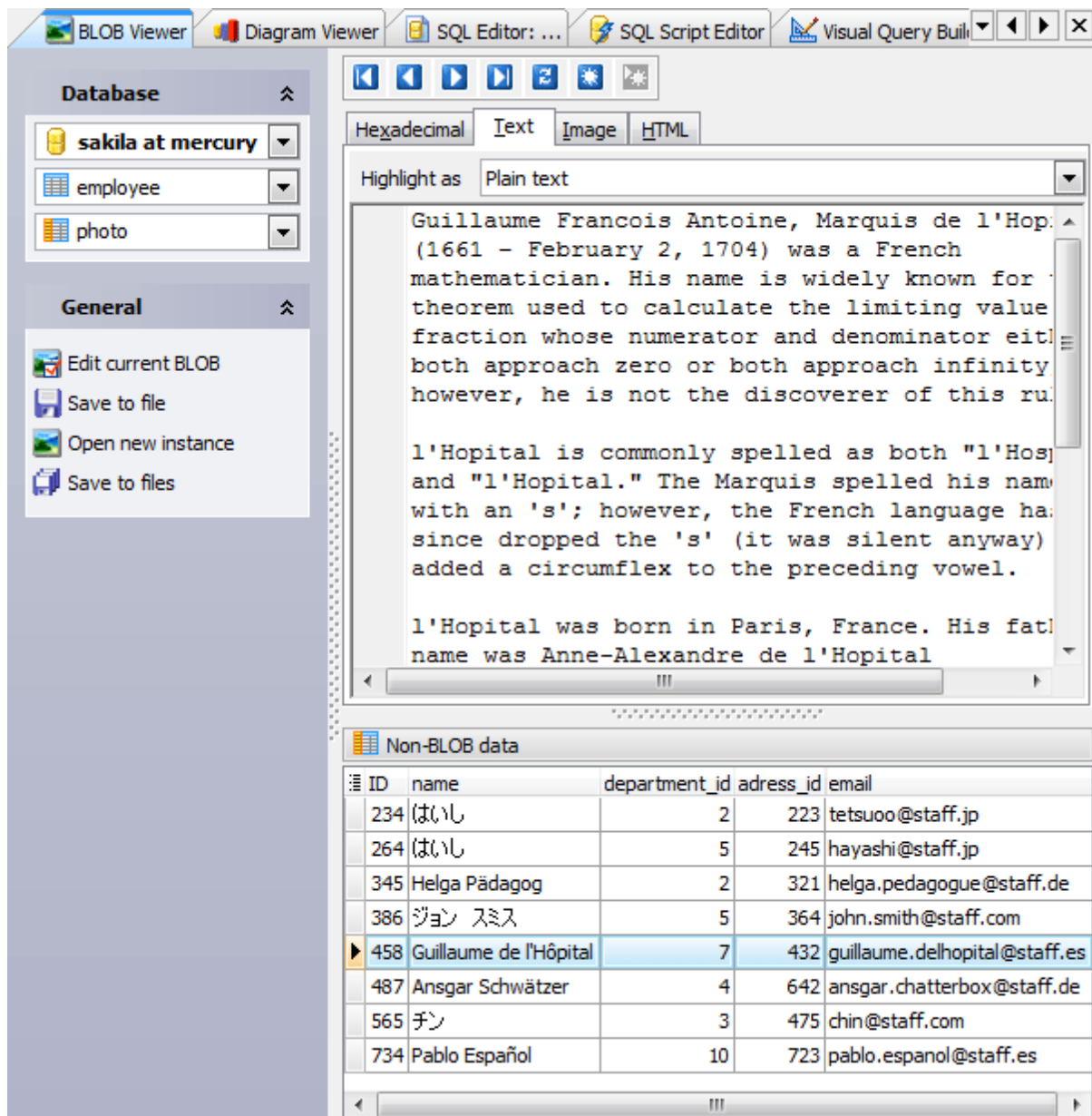
7.1.1 Viewing as hexadecimal dump

The [Hexadecimal](#) panel allows you to view data in hexadecimal mode.



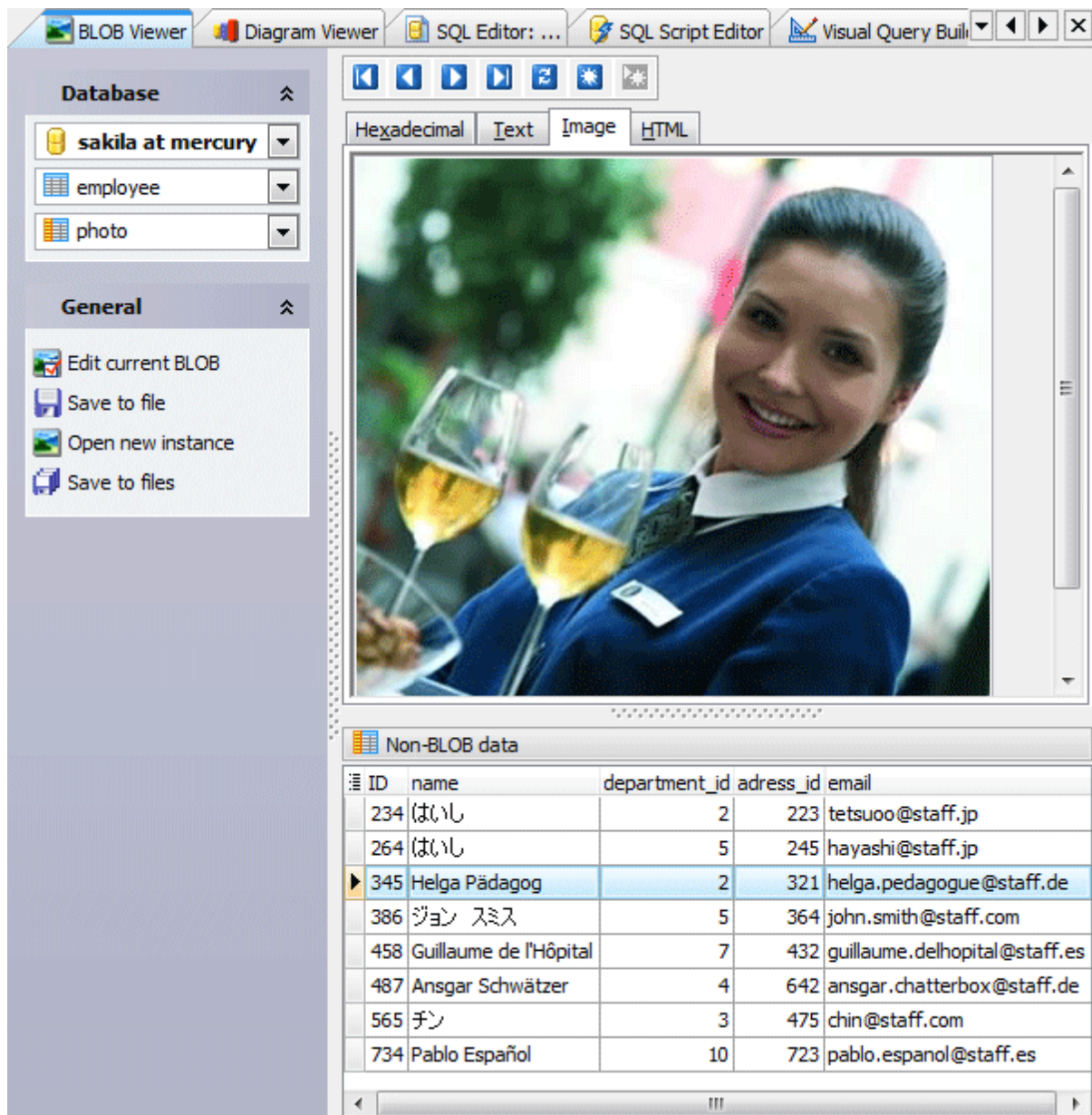
7.1.2 Viewing as plain text

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.



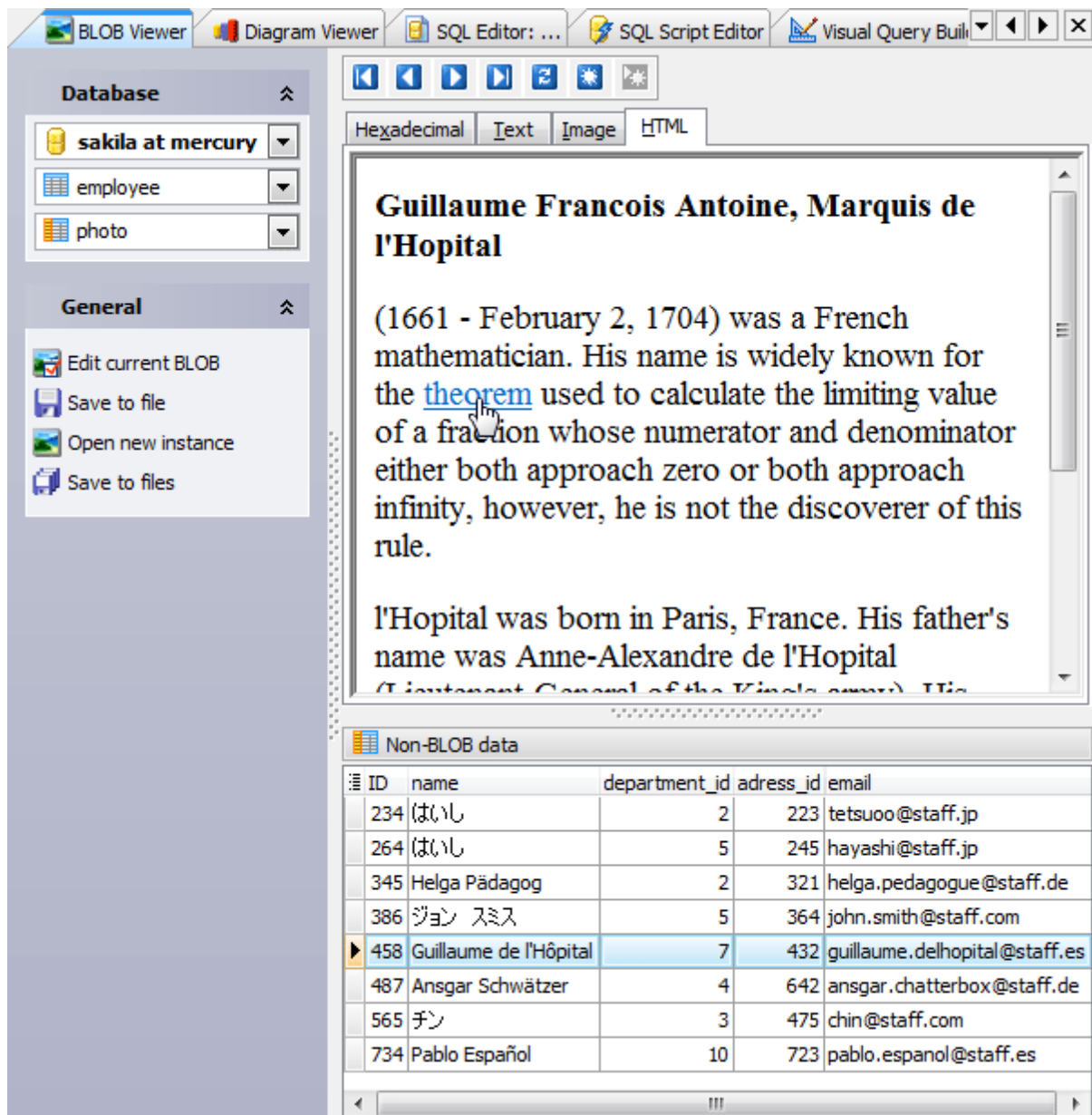
7.1.3 Viewing as image

The [Image](#) panel displays field data as image.



7.1.4 Viewing as HTML

The [HTML](#) panel displays field data as HTML.



7.1.5 Viewing as PDF

The PDF panel allows you to browse PDF data stored in the database.

Database

test_utf8 at d

public.software

manual

General

Edit current BLOB

Save to file

Open new instance

Save to files

Hexadecimal Text Image HTML PDF

18 / 103 75%

Comment Share

15 PostgreSQL PHP Generator Help

2 Getting started

Connection properties
Set the [connection parameters](#) for the connection with.

Script connection properties
Specify here connection parameters for PostgreSQL. For example, if your webserver and PostgreSQL server are on the same host as localhost.

Projects
When working with a project, all the session variables can be edited if necessary. To run a web application, select a Project on the first wizard step and enter the project name. Projects are also available from this popup menu.

Connection properties

☒ I can connect to the server directly or via SSH tunneling

[Configure SSH options](#)

8.50 x 11.00 in

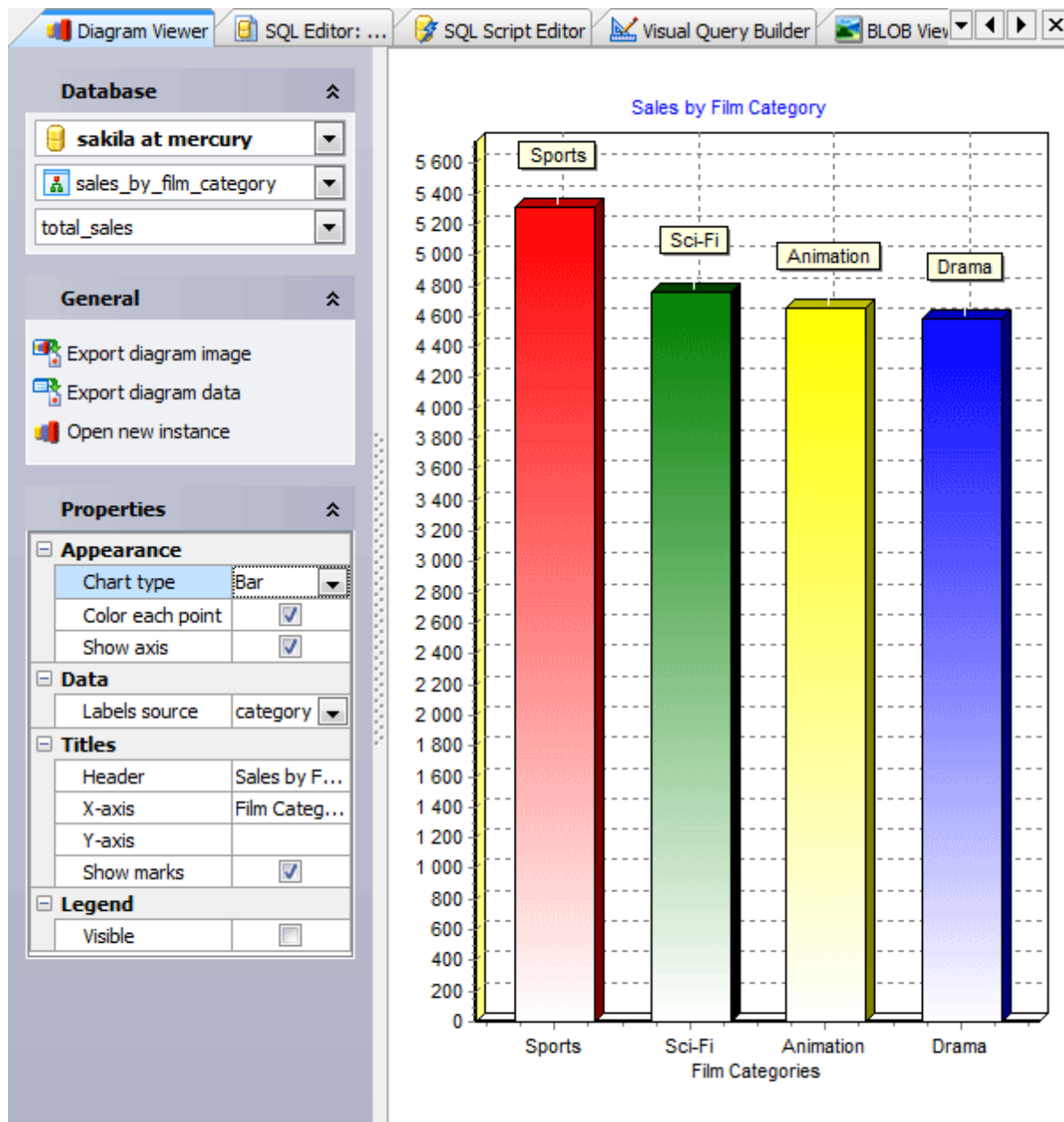
Non-BLOB data

id	full_name	description_id
1	PostgreSQL PHP Generator	1
2	Code Factory for MySQL	3
3	SQLite DataWizard	2
4	MS SQL Maestro	4

7.2 Diagram Viewer

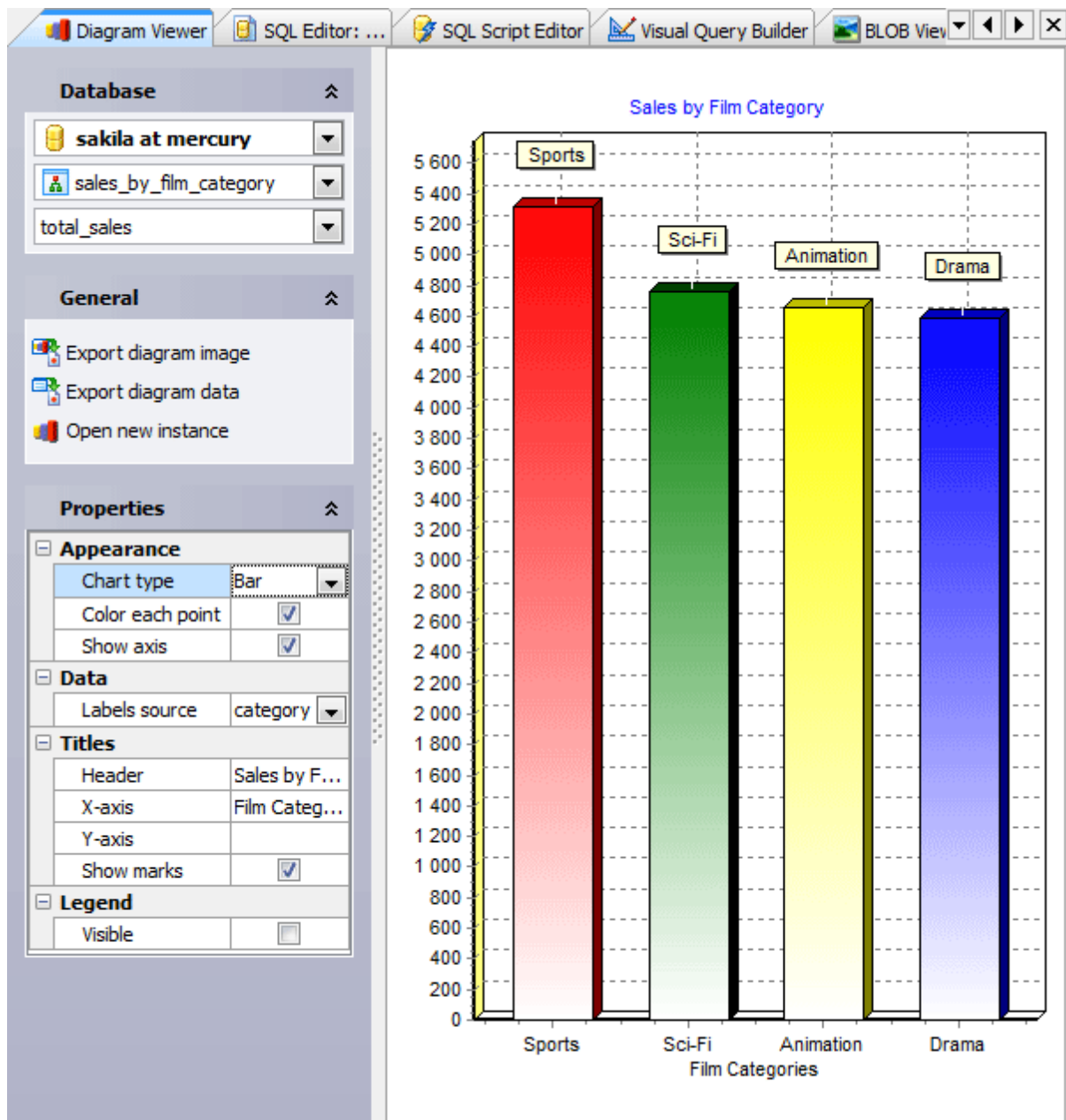
Diagram Viewer is a tool for representing data from a table or a query as a diagram in various ways. This means you can build a diagram represented as bars, lines, areas, points or pies, colored or not, with axis visible or not; specify axis labels source, diagram header and more. The **Diagram Viewer** also has the [Export diagram image](#)^[105] and the [Export diagram data](#) features implemented, with a lot of formats supported.

- [Customizing diagram options](#)^[104]
- [Exporting diagram as a graphical image](#)^[105]



7.2.1 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

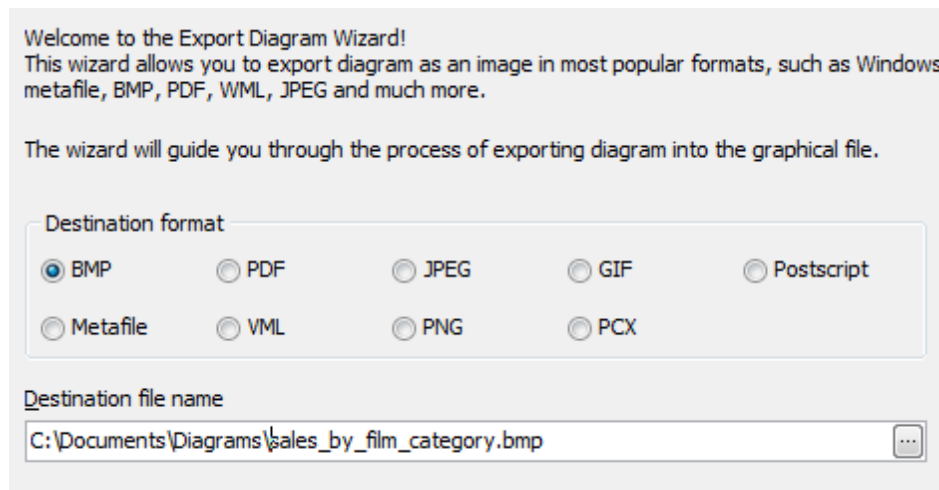
Legend

The only **Visible** property of this subgroup specifies whether the legend rectangle should be represented on the right side of the diagram or not.

7.2.2 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.



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	<input type="text" value="386"/>	Height	<input type="text" value="611"/>
-------	----------------------------------	--------	----------------------------------

☒ Keep aspect ratio

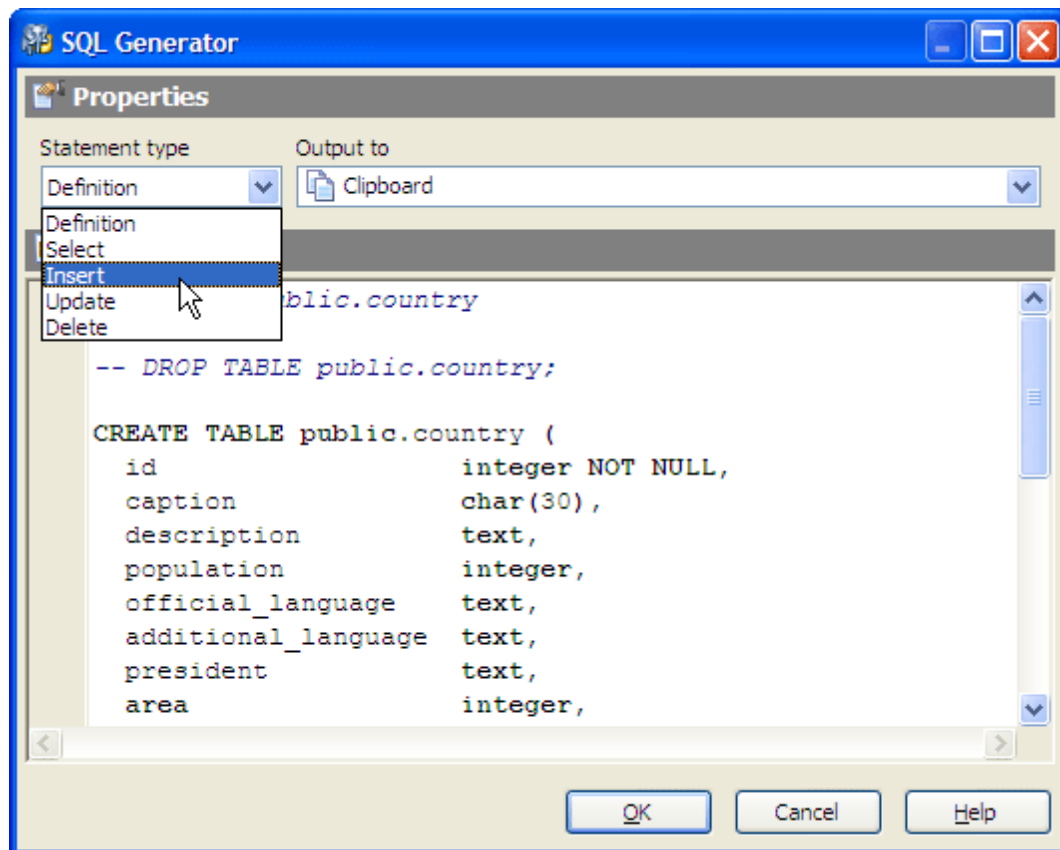
☐ Open exported diagram in associated program

Click "Ready" to export the diagram.

7.3 SQL Generator

Among other features PostgreSQL Code Factory 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).

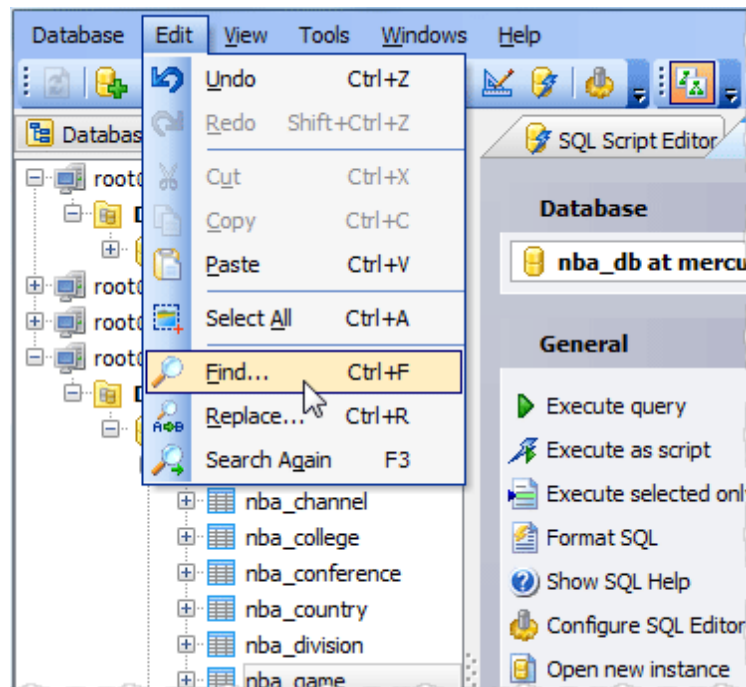
The SQL Generator window can be invoked from the Explorer tree.



7.4 Dialogs

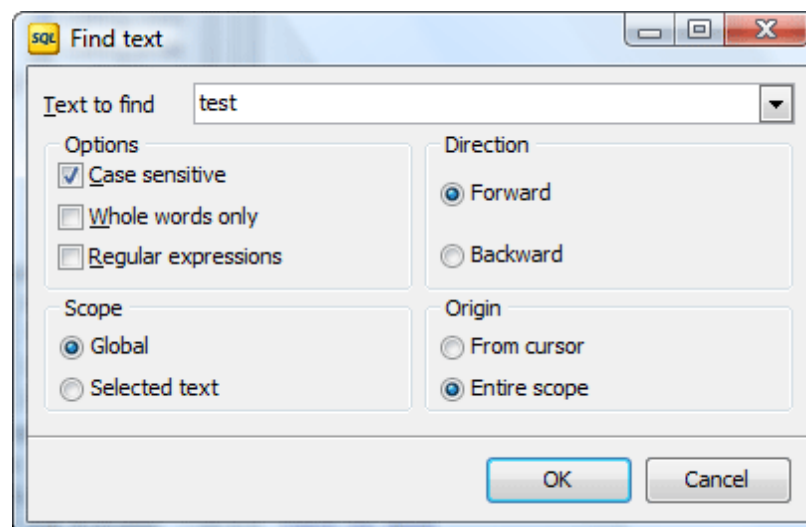
PostgreSQL Code Factory provides two dialogs for searching and replacing text in the editor areas of the database tools. Both of them are available through the popup menu of the editor area.

- [Find Text dialog](#) ^[108]
- [Replace Text dialog](#) ^[109]



7.4.1 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

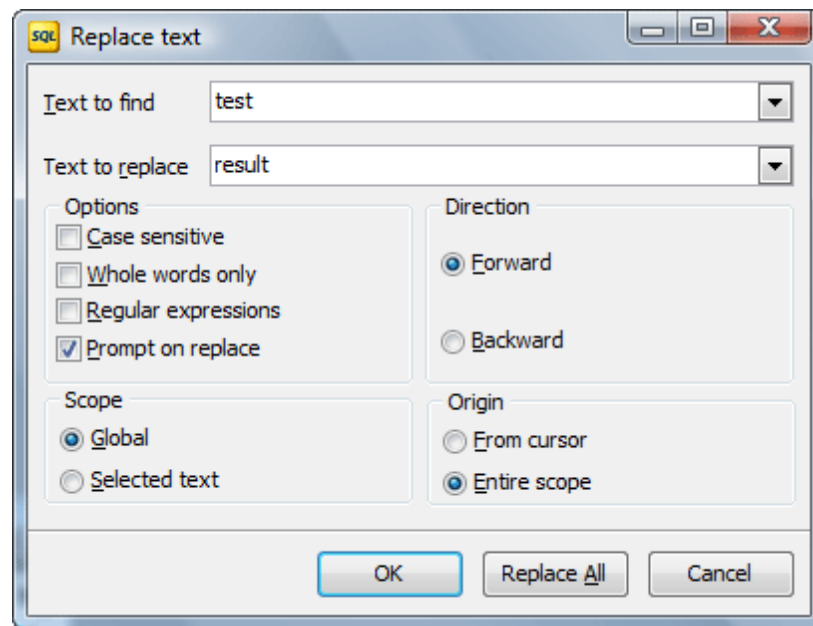
The search starts at the cursor's current position, and then proceeds either forward to the end of the scope, or backward to the beginning of the scope depending on the [Direction](#) setting. [From Cursor](#) is the default setting.

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.

7.4.2 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.

8 Options

PostgreSQL Code Factory 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](#)^[113]
General PostgreSQL Code Factory options: environment style, confirmations, window restrictions, explorer tree, [SQL Editor](#), [Visual Query Builder](#), etc.
- [Editors & Viewers](#)^[130]
Customizing of all the SQL editors - [SQL Editor](#), [SQL Script Editor](#), etc.
- [Appearance](#)^[138]
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](#)^[147] for details).

It is a good idea to check through these settings before you start working with PostgreSQL Code Factory. You may be surprised at all the things you can adjust and configure!

8.1 Application

The [Application](#) section allows you to customize common rules of PostgreSQL Code Factory behavior. The section consists of several tabs; follow the links to find out more about each of them.

- [Preferences](#) ^[113]
- [Confirmations](#) ^[114]
- [Tools](#) ^[115]
 - [Explorer](#) ^[116]
 - [SQL Editor](#) ^[117]
 - [SQL Script Editor](#) ^[118]
 - [Query Builder](#) ^[119]
 - [BLOB Viewer](#) ^[121]
 - [Export data](#) ^[122]
- [Data Grid](#) ^[123]
 - [Colors](#) ^[126]
 - [Formats](#) ^[126]

8.1.1 Preferences

User interface area allows you to select your favorite UI style according to your preferences.

☒ [Display splash screen at startup](#)

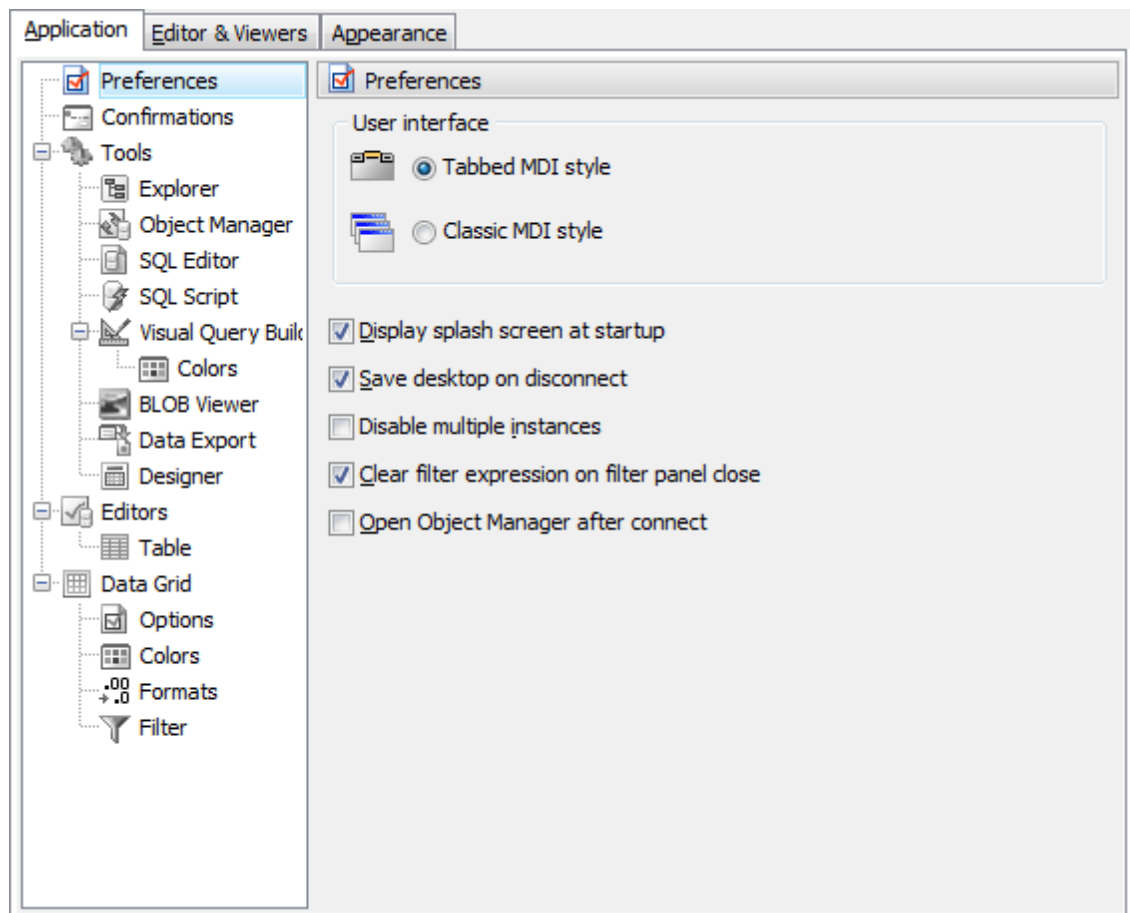
Displays the splash screen on PostgreSQL Code Factory startup.

☒ [Save desktop on disconnect](#)

Saves all the database windows and their positions on disconnecting from the database.

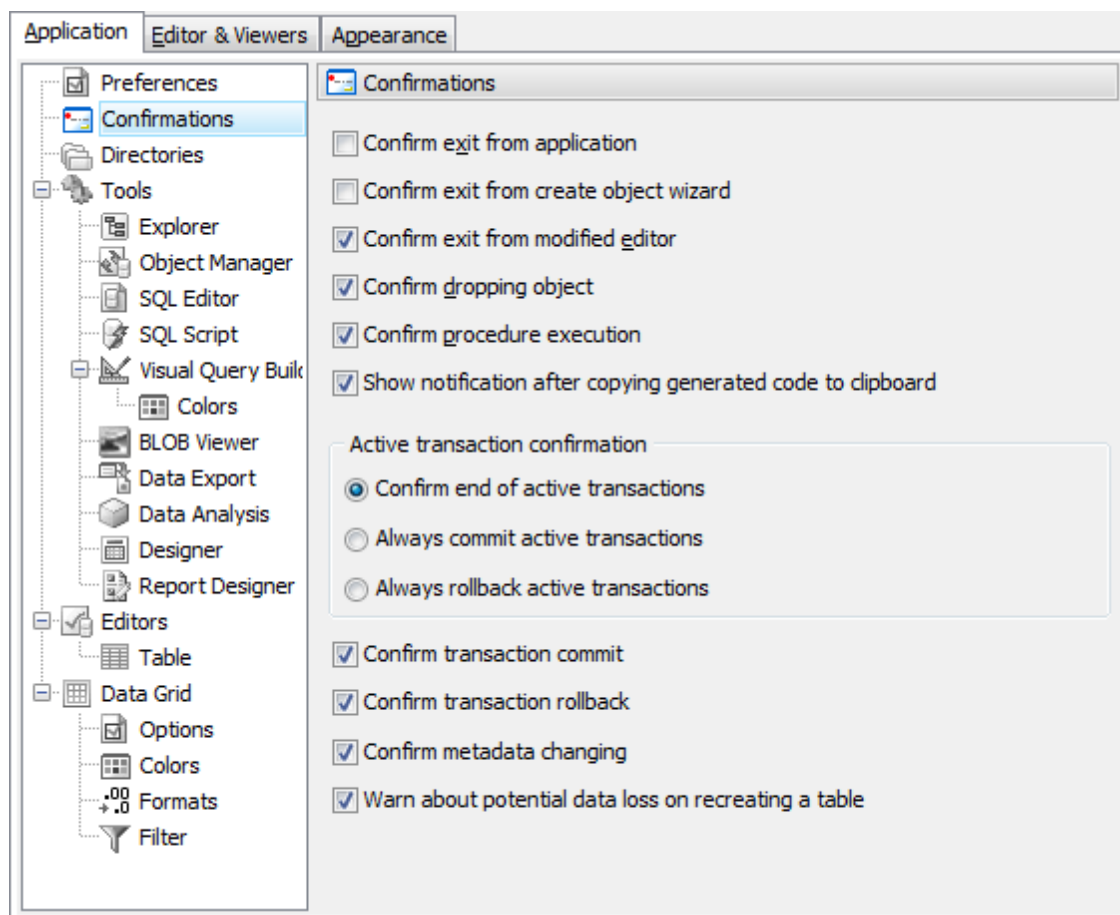
☒ [Disable multiple instances](#)

Prohibits running multiple instances of PostgreSQL Code Factory.



8.1.2 Confirmations

Use this tab to manage application confirmations.



☒ **Confirm exit from application**

If this option is checked, the program requires confirmation when you want to exit <% PRODUCT_NAME%>.

☒ **Transaction confirmation**

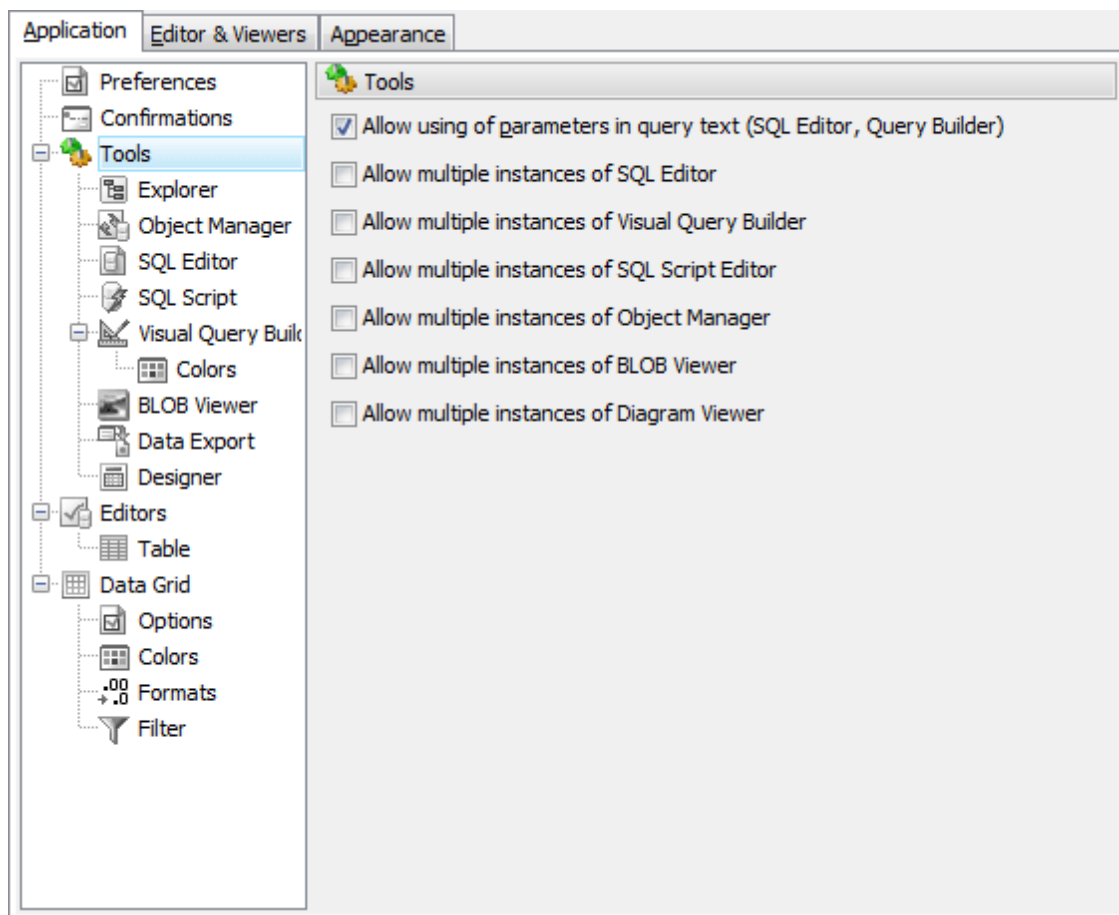
Select whether you will be prompted to commit or rollback active transaction or PostgreSQL Code Factory will commit or rollback transactions without asking.

☒ **Confirm metadata changing**

If this option is checked, the program requires confirmation for changing metadata.

8.1.3 Tools

Below you will find a detailed decryption of the following tools options.



☒ **Allow using of parameters in query text**

Check this option to be able to use query parameters in [SQL Editor](#)^[44] and [Visual Query Builder](#)^[49].

☒ **Allow multiple instances of SQL Editor**

Check this option to be able to use multiple instances of [SQL Editor](#)^[44] simultaneously.

☒ **Allow multiple instances of Visual Query Builder**

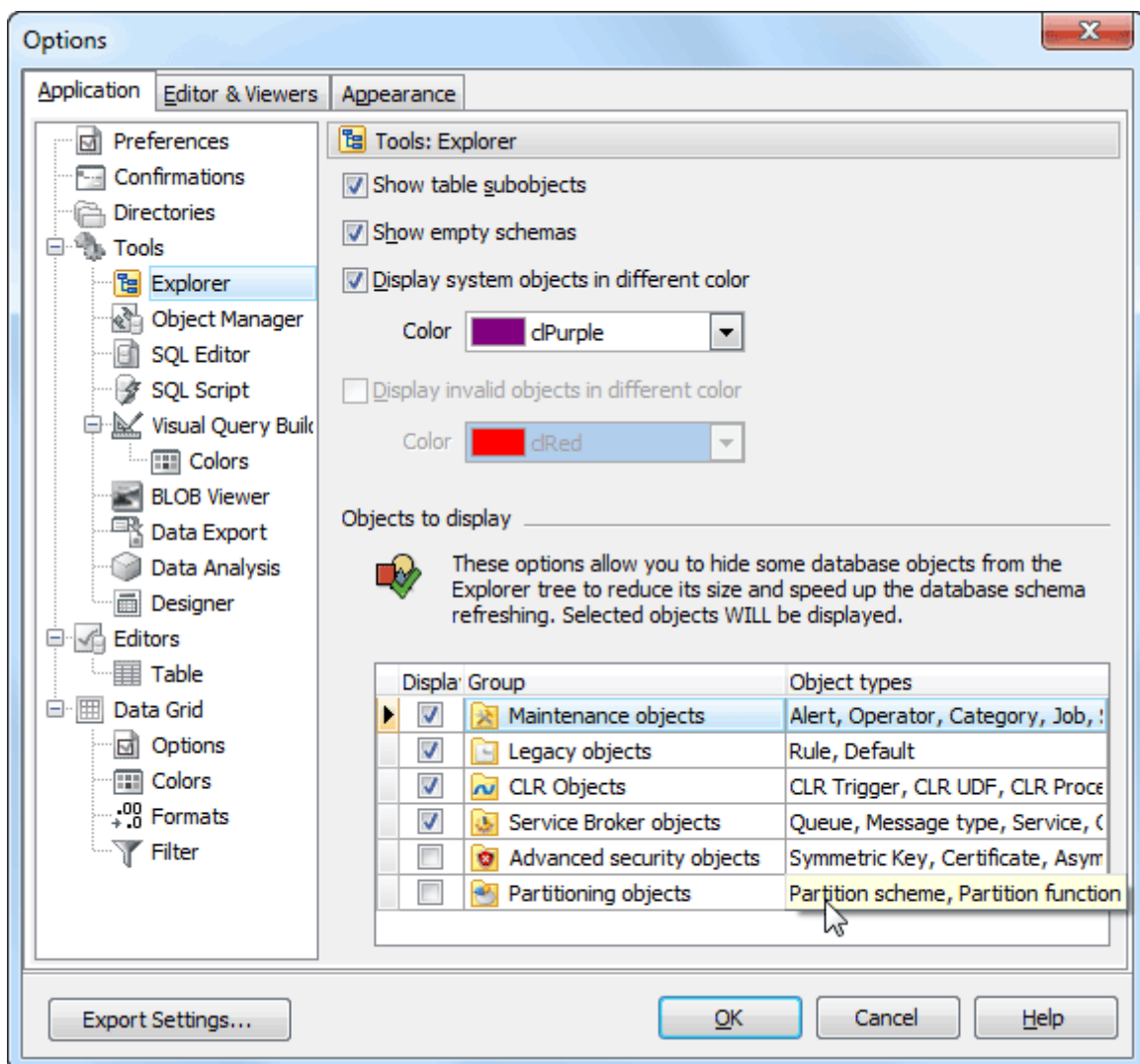
Check this option to be able to use multiple instances of [Visual Query Builder](#)^[49] simultaneously.

☒ **Allow multiple instances of SQL Script Editor**

Check this option to be able to use multiple instances of [SQL Script Editor](#)^[59] simultaneously.

8.1.3.1 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.

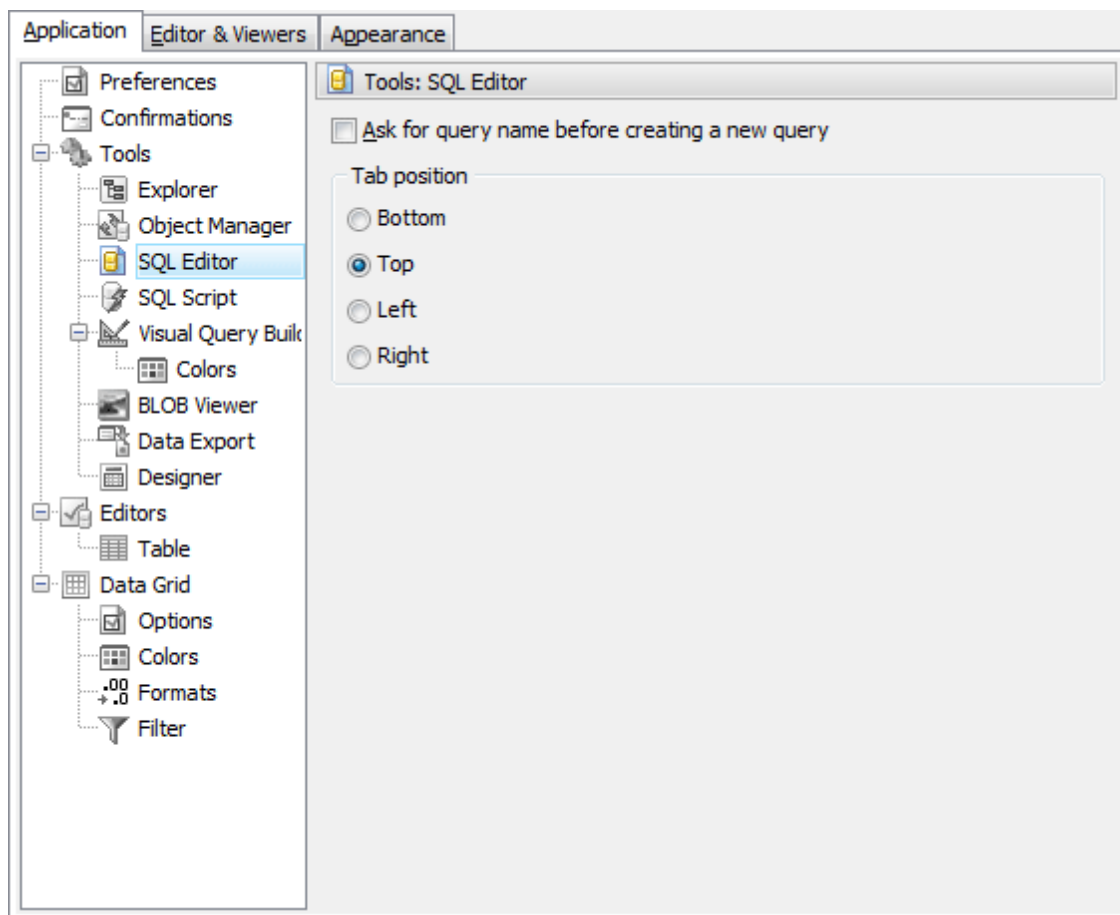
☒ **Display system objects in different color**

Represents all system objects in selected color.

You can also exclude/include rarely used objects from/to the Explorer tree. Manage object groups to be displayed at Explorer with corresponding checkboxes.

8.1.3.2 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](#) asks for a query name each time you create a new query.

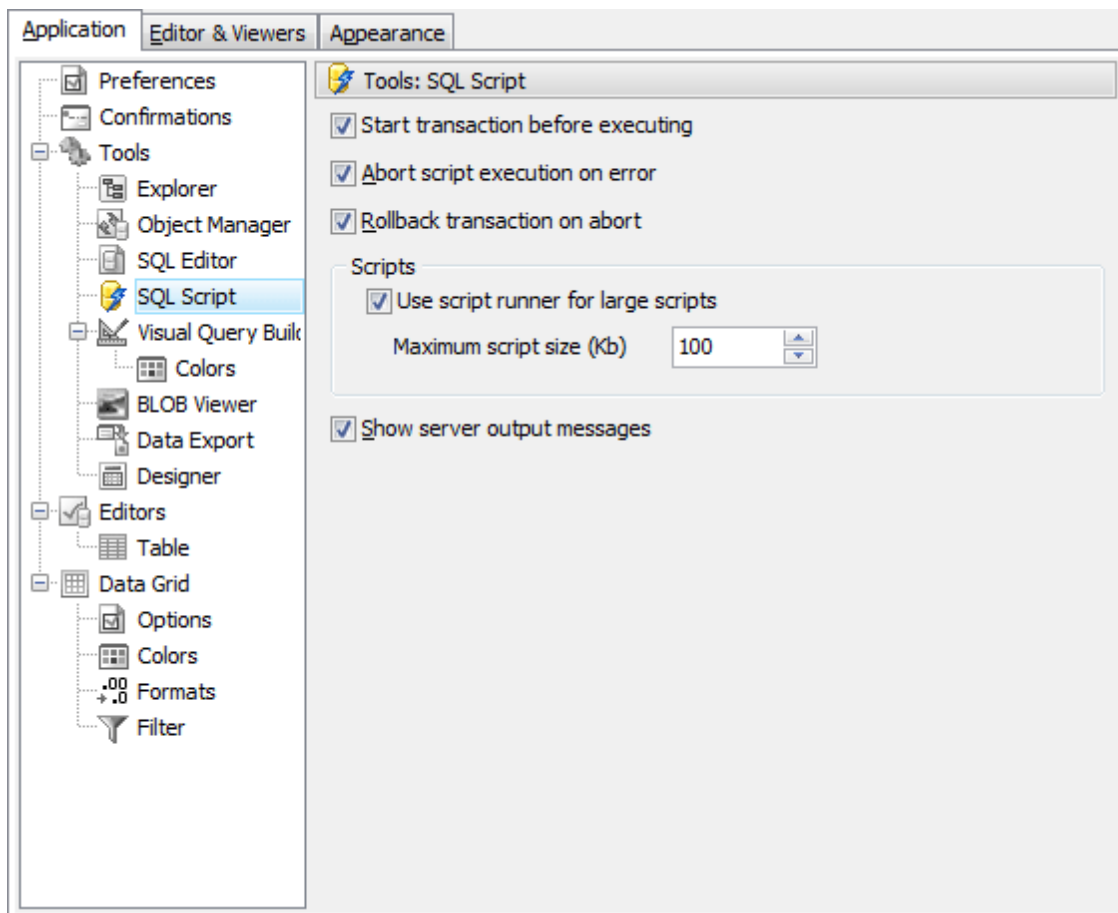
☒ **Auto commit**

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.

8.1.3.3 SQL Script Editor

Below you will find a detailed decryption of the following [SQL Script Editor](#) options.



☒ **Abort script execution on error**

If this option is checked, script execution aborts when an error occurs.

☒ **Rollback transaction on abort**

This option evokes automatic rollback on script execution abort.

☒ **Use script runner for large scripts**

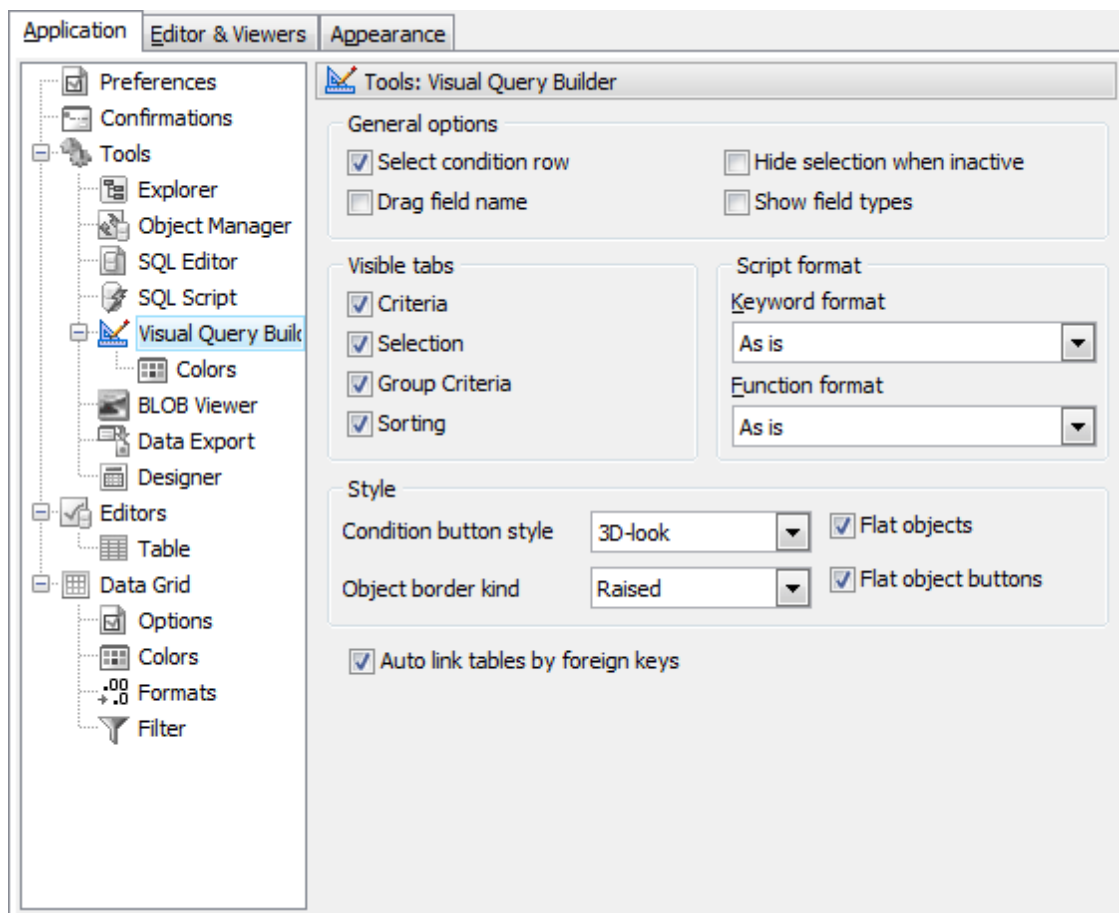
Check the box to execute large script in the fastest way. You can change the maximum size of a script to execute without script runner.

☒ **Show server output messages**

Turn the option ON to see warning messages generated by the server.

8.1.3.4 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](#).

☒ **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

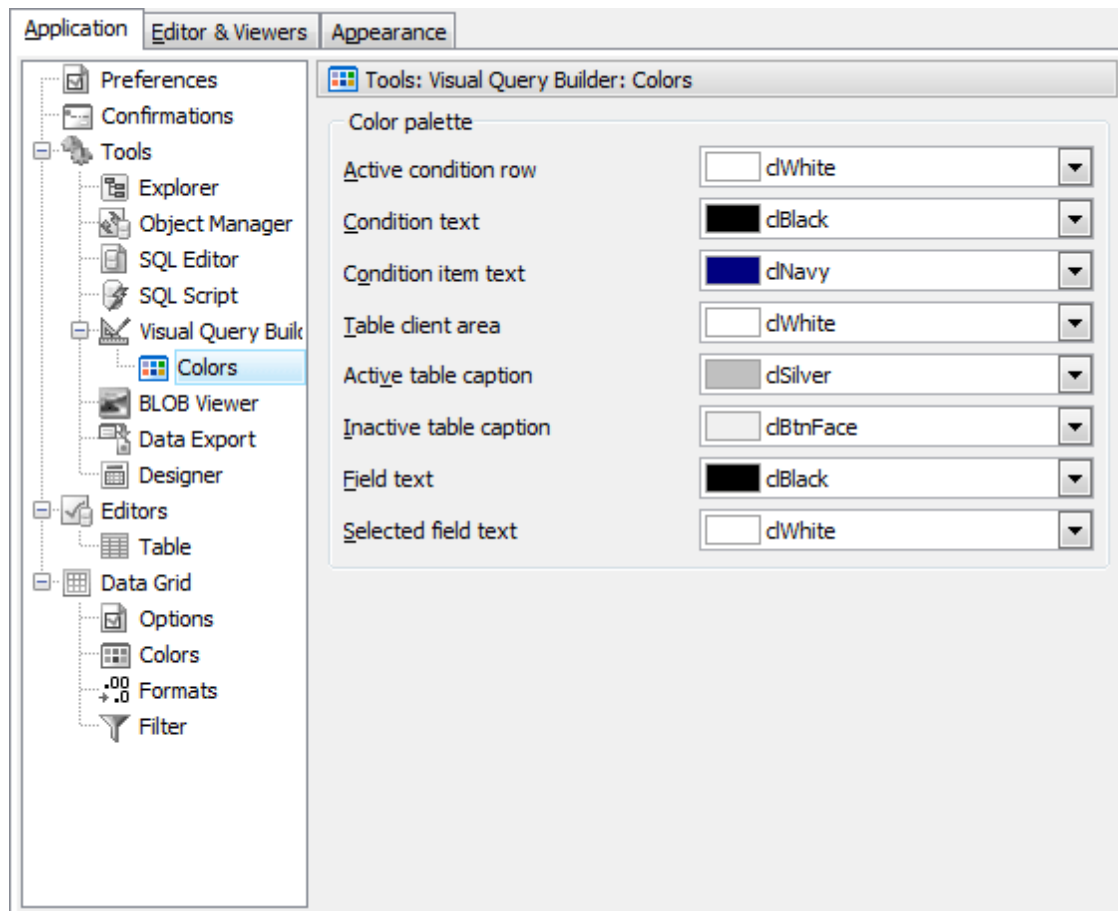
If tables that have foreign key reference are placed in [Query Builder](#), in diagram they will be auto linked.END

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.

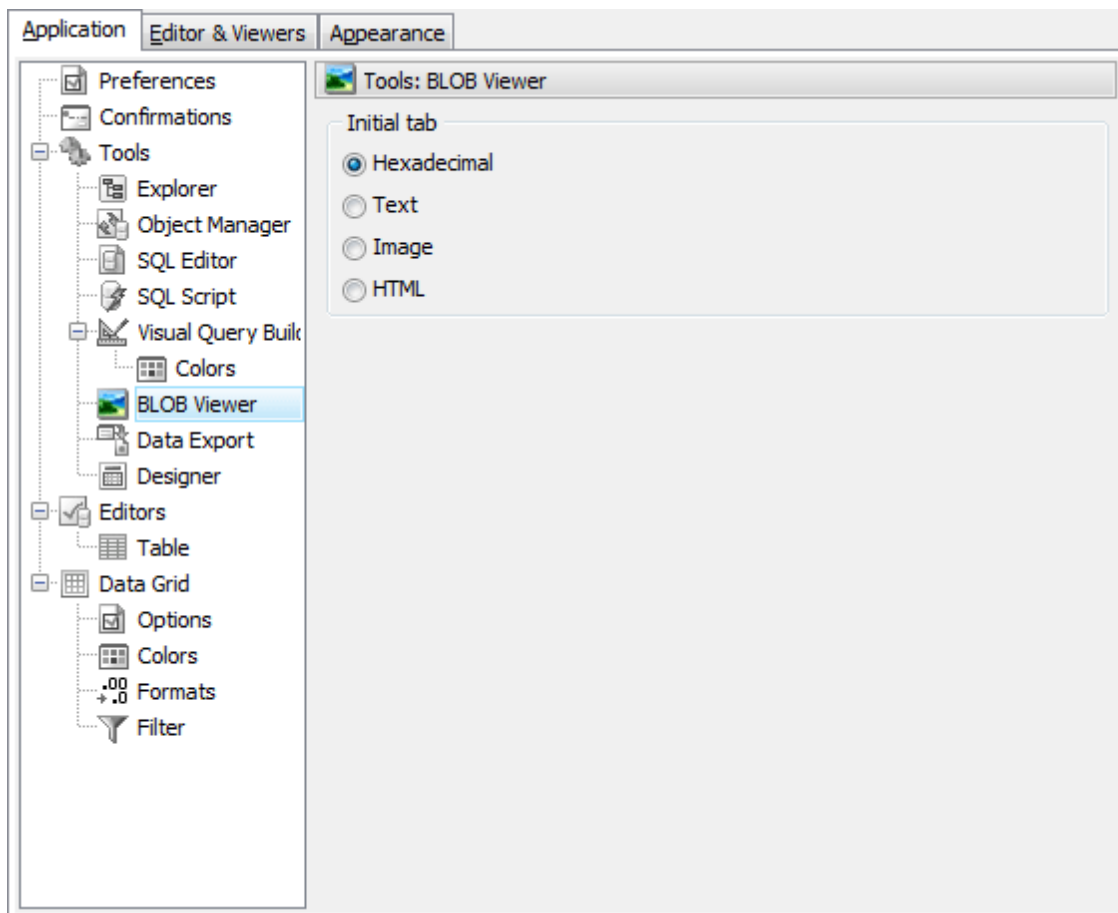
8.1.3.4.1 Colors

The tab is provided to editing of the Query Builder color schema. Customize colors for all editor element according to your preferences.



8.1.3.5 BLOB Viewer

Below you will find a detailed decryption of the following [BLOB Viewer](#) ⁹⁷ options.



Initial tab

Specifies which tab of **BLOB Viewer** should be active when it is initially opened.

8.1.3.6 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

PostgreSQL Code Factory provides you with [two grid modes](#) of viewing data:

- Full grid mode allows you to group, filter and sort data in a usual way.
- Simple mode is provided for working with large records number. For data fetching speed-up, filtering, sorting, and grouping features are not enabled in this mode.

You can use [notification message](#) to indicate simple mode.

Set the number of records to switch to simple mode automatically or select [Always use full mode](#).

[Limit options](#)

Allows you either to select all records from table after opening the Data tab, or select only specified number of rows on one page with an ability to rotate pages and view all data.

[Row numbers](#)

This options group allows you to manage grid rows numbering.

To enable/disable the numbering, use [Display row numbers](#) checkbox. You can set the number columns width with [Maximum digit count](#). (I.e. for the value '3' the max column number will be 999).

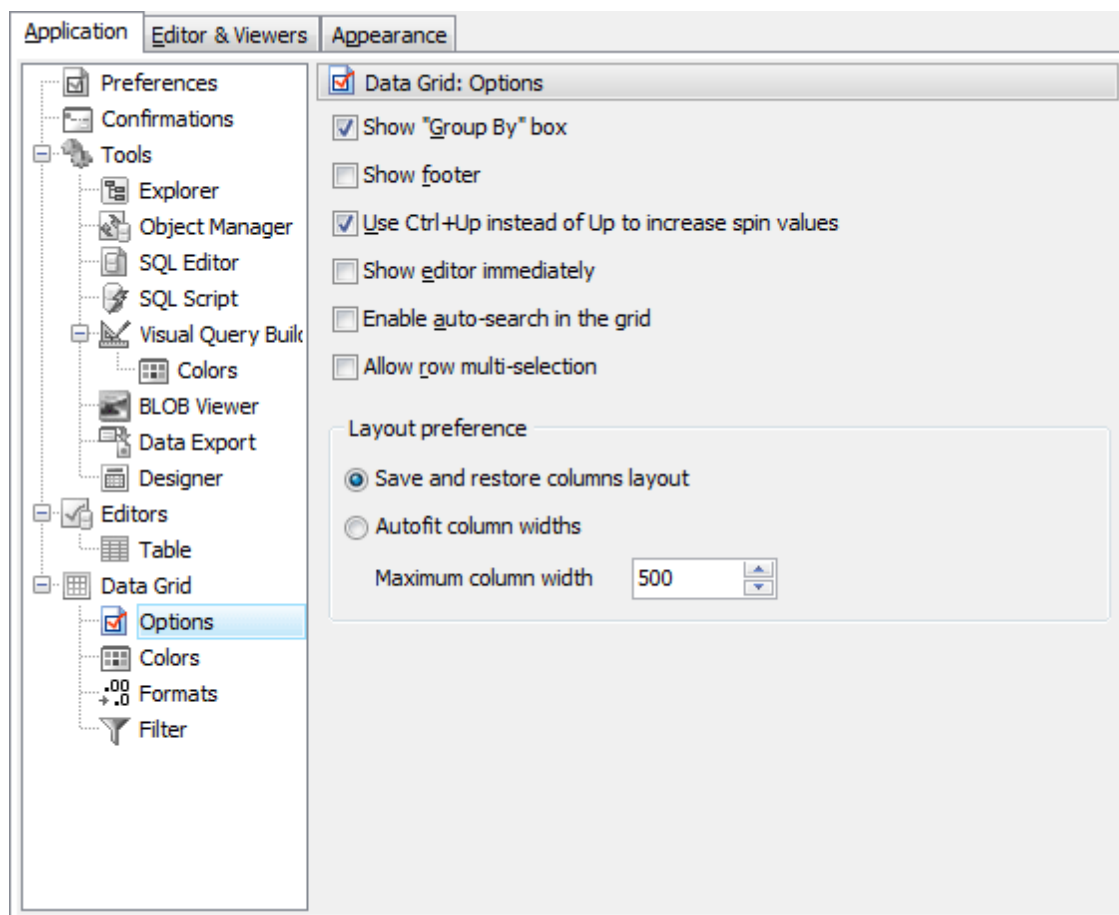
For uniformity you can use the [Display leading zeros](#) option. With this option enabled and maximum digit count '3' your numbering column will be of the form: '001, 002, 003, ...'.

☒ [Do not open separated connection](#)

With this option enabled a new connections for fetching data is not opened. This gives you an ability to work with data a little bit faster, because time for opening a new connections is not demanded.

8.1.4.1 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**

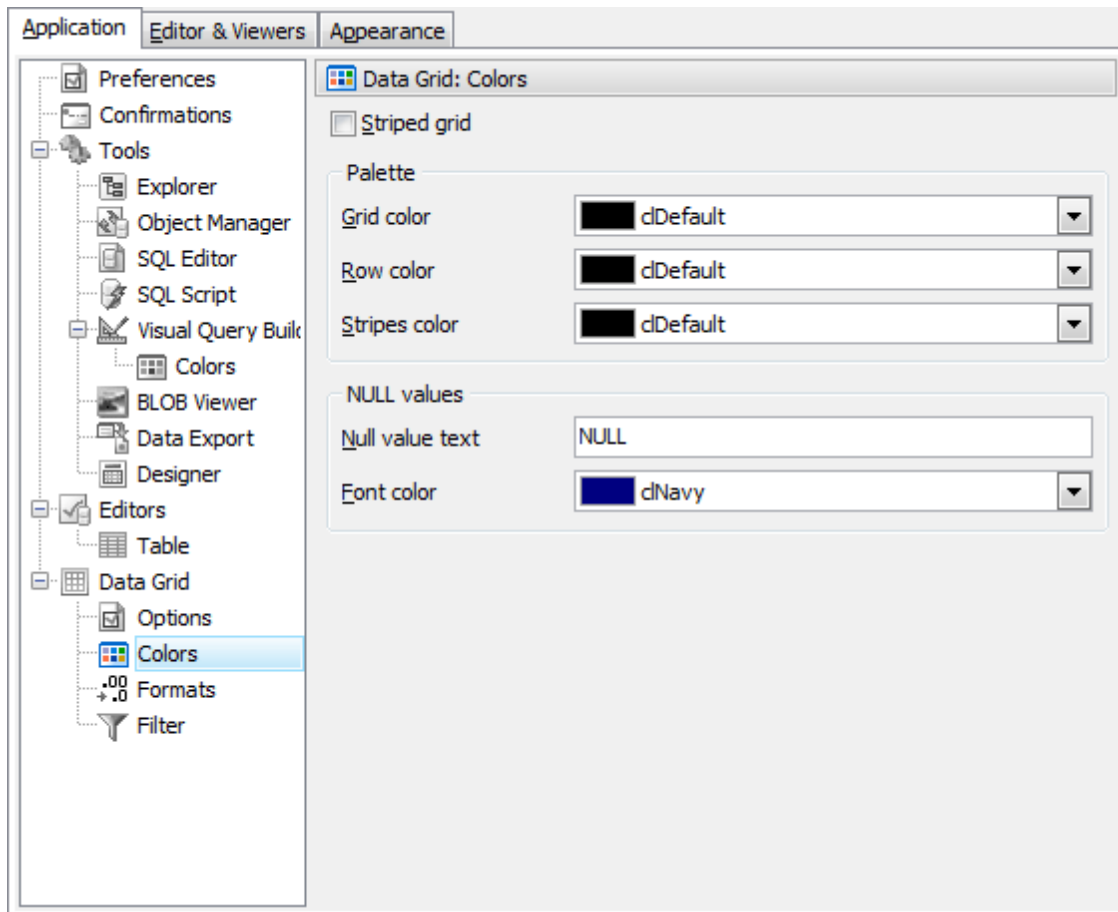
Allows you to select multiple records using the Ctrl and Shift keys.

Layout preference

Select whether PostgreSQL Code Factory should remember the column positions for the grids or fit them automatically.

8.1.4.2 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

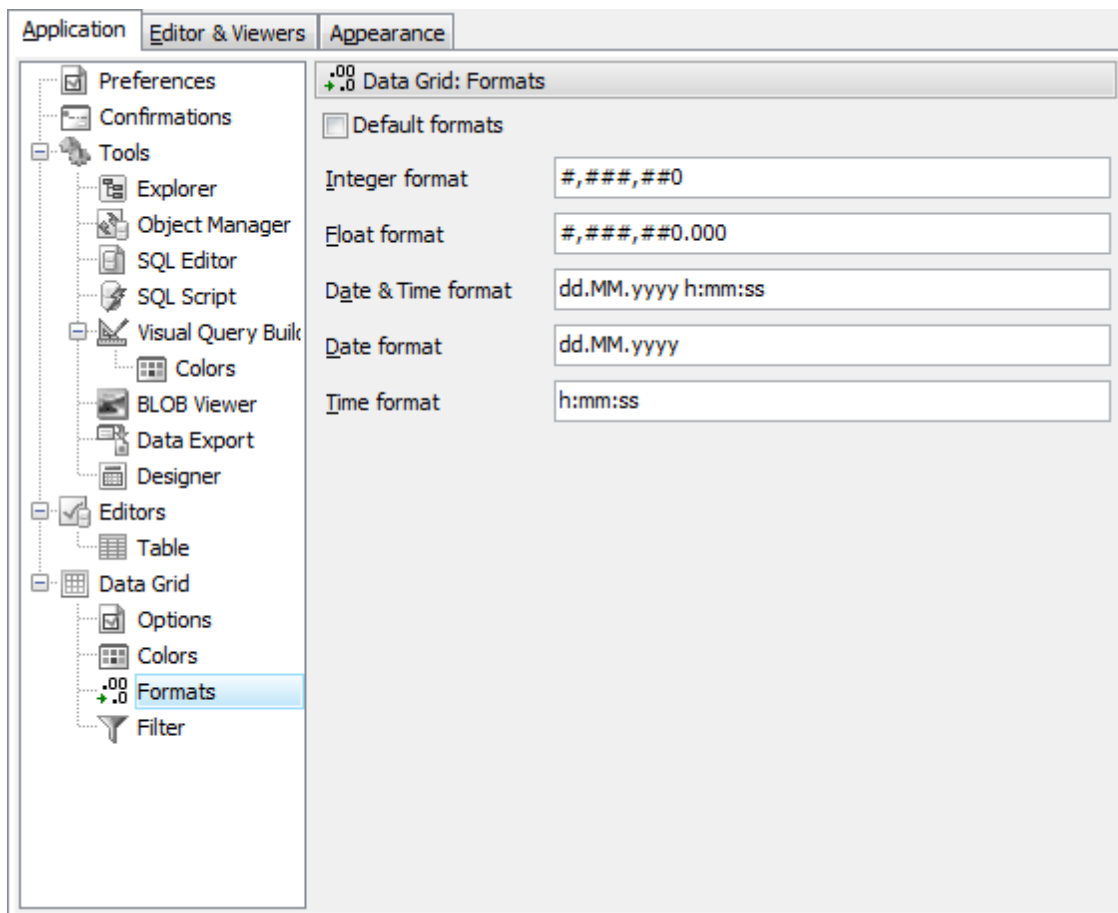
Defines the color of the odd rows if the [Striped Grid](#) option is on.

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.

8.1.4.3 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

(".", ",", ":") symbols can be placed as separators in format masks for *integer*, *float*, *date*, *time*, *date time*, *currency* types.

These options allows you to customize data filtering in grids.



Change the **Position of filter panel** and customize timestamp data filtering: check the **Use relative dates in filters** box to include in column popup filter such options as "Yesterday",

"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).

8.2 Editors & Viewers

The [Editors & Viewers](#) section allows you to set the parameters of viewing and editing the SQL statements within PostgreSQL Code Factory.

- [General](#) ^[130]
- [Display](#) ^[131]
- [SQL highlight](#) ^[132]
- [PHP highlight](#) ^[134]
- [XML highlight](#) ^[133]
- [Code Insight](#) ^[135]
- [Code Folding](#) ^[136]

See also: [SQL Editor](#) ^[44], [SQL Script Editor](#) ^[59], [Visual Query Builder](#) ^[49].

8.2.1 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](#)

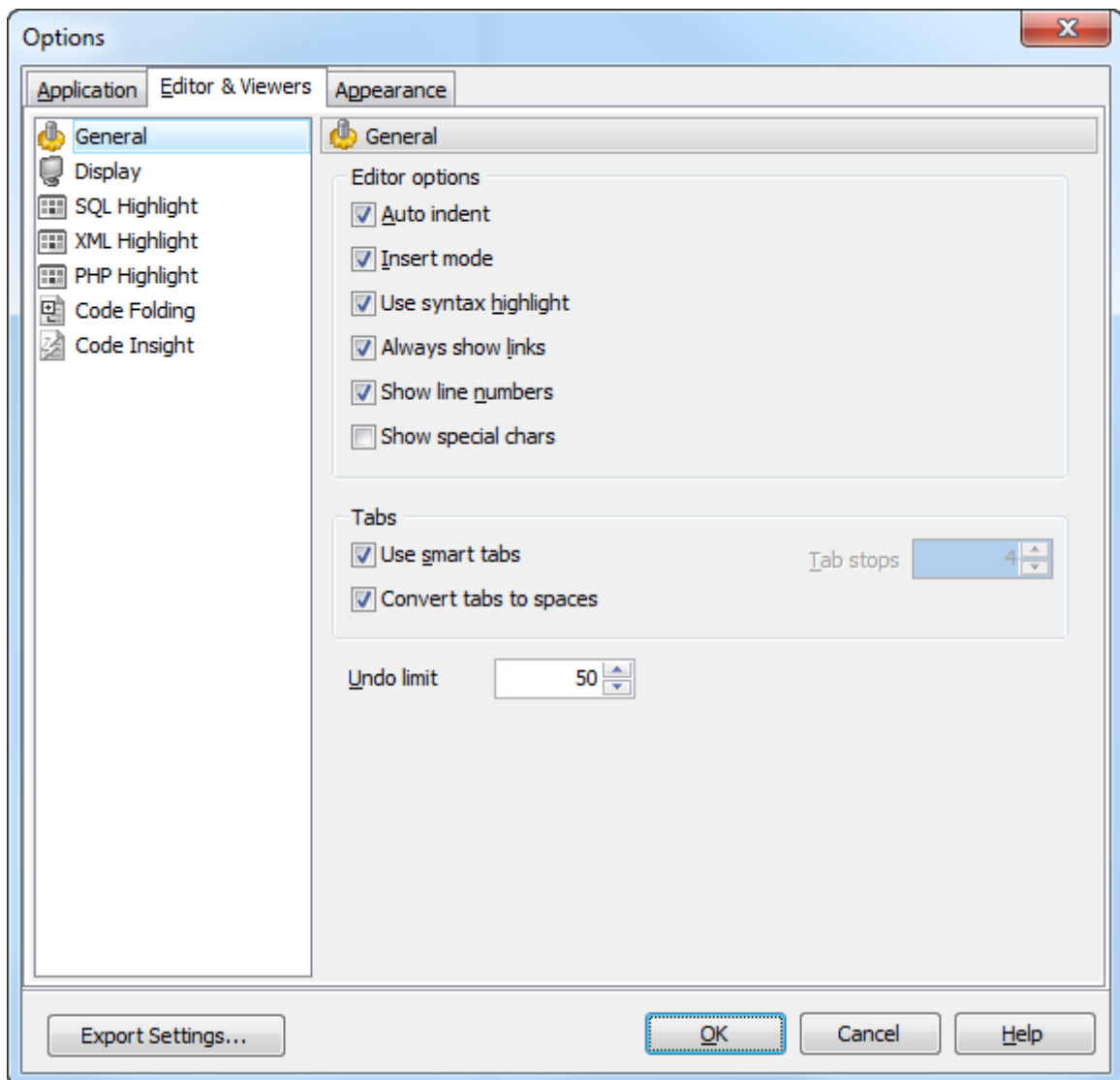
If this option is checked, each time you press the Tab key, the appropriate number of spaces will be added to the edited text.

[Tab Stops](#)

Defines the tab length, used when editing text.

Undo Limit

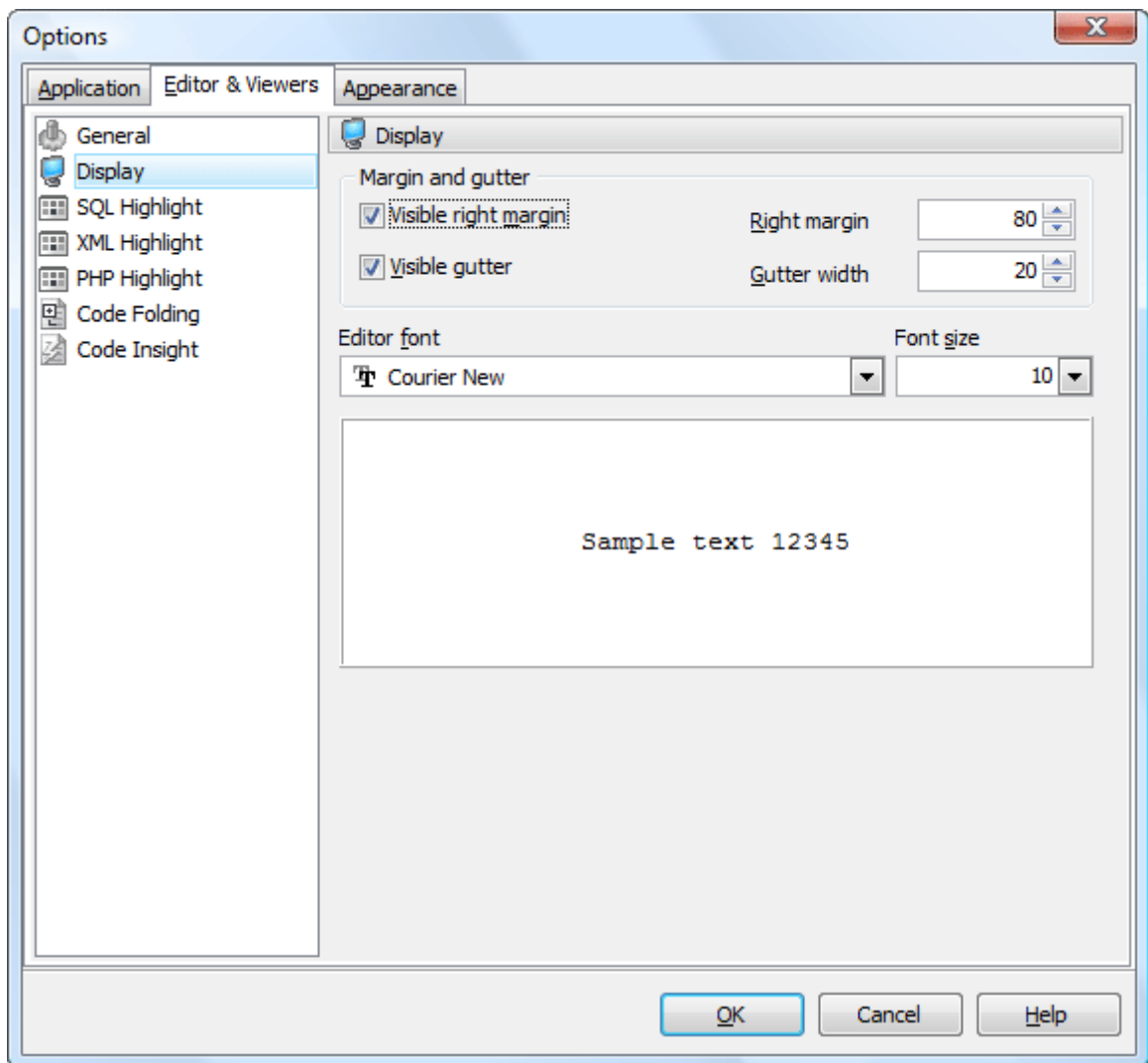
Defines the maximum number of changes possible to be undone.



8.2.2 Display

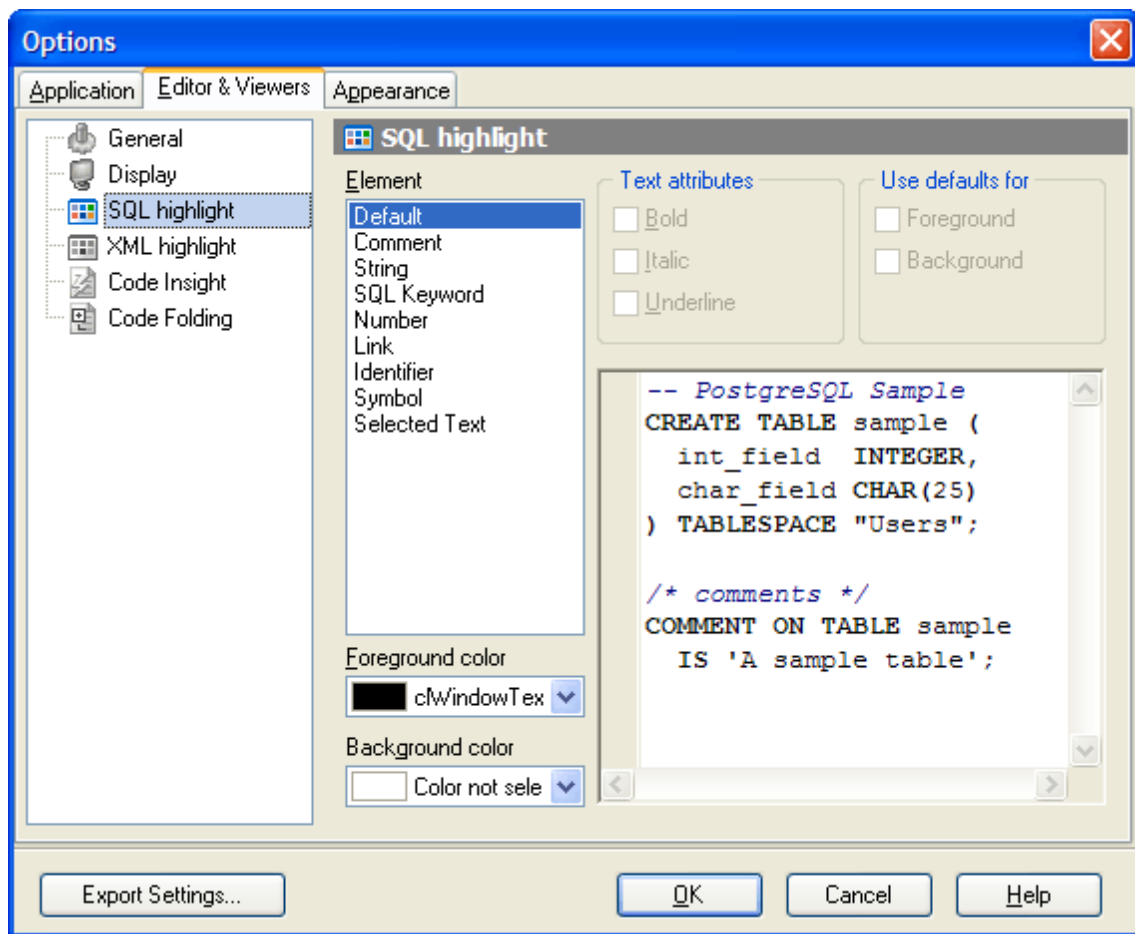
You can disable/enable the right text margin and the gutter of the editor area, set the position of the right text margin as [Right margin](#), and [the Gutter width](#).

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.



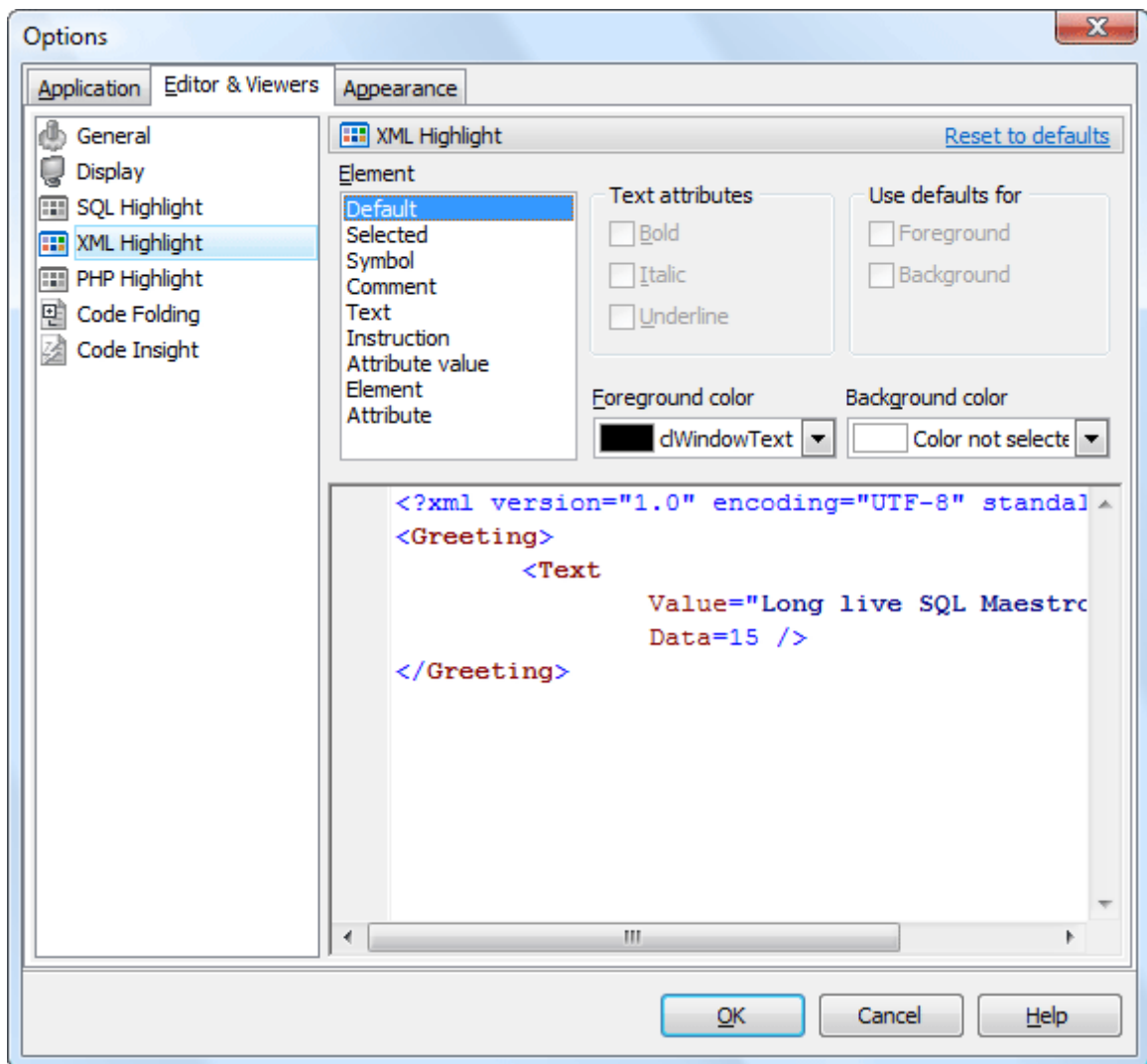
8.2.3 SQL highlight

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.



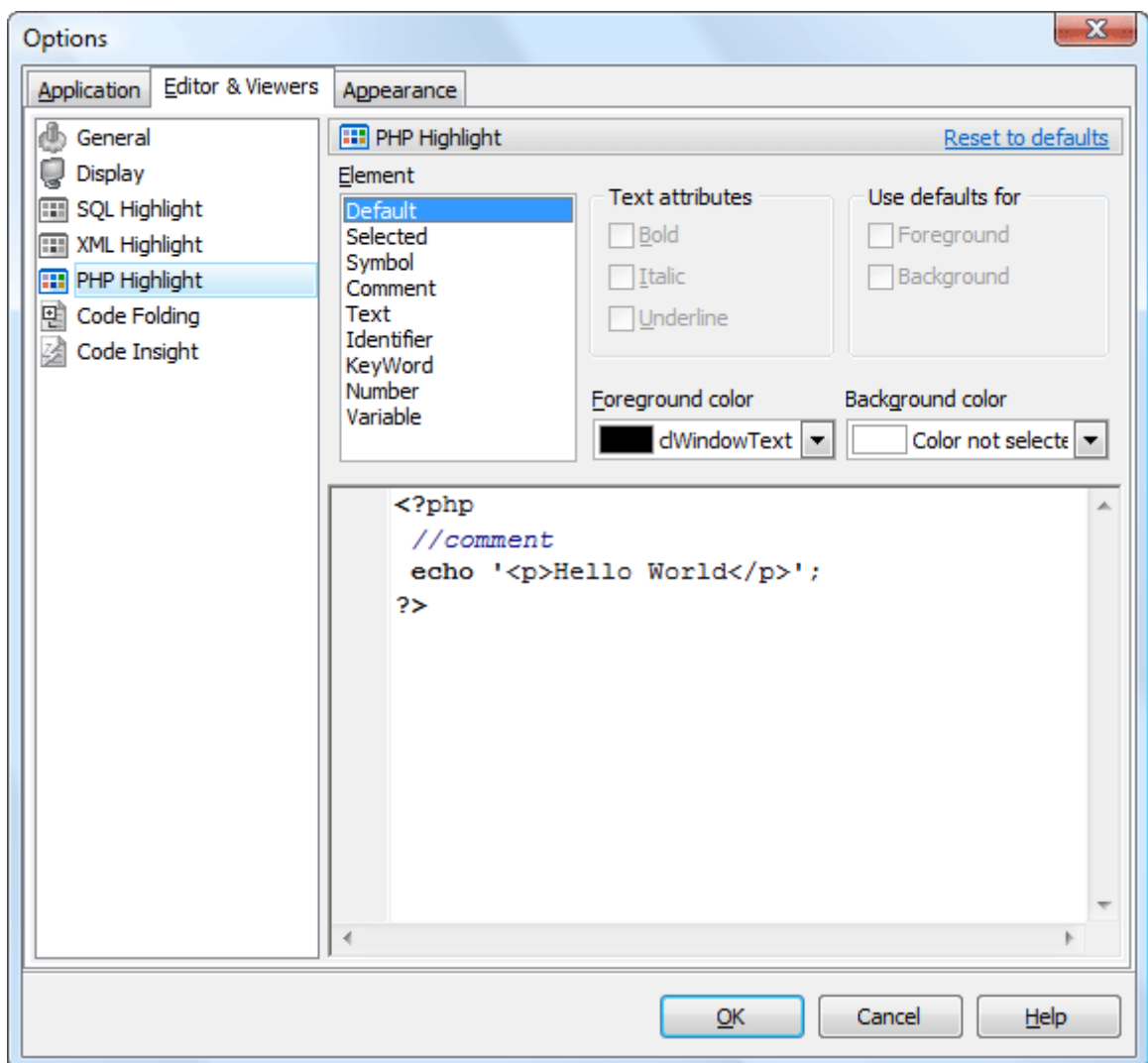
8.2.4 XML highlight

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.



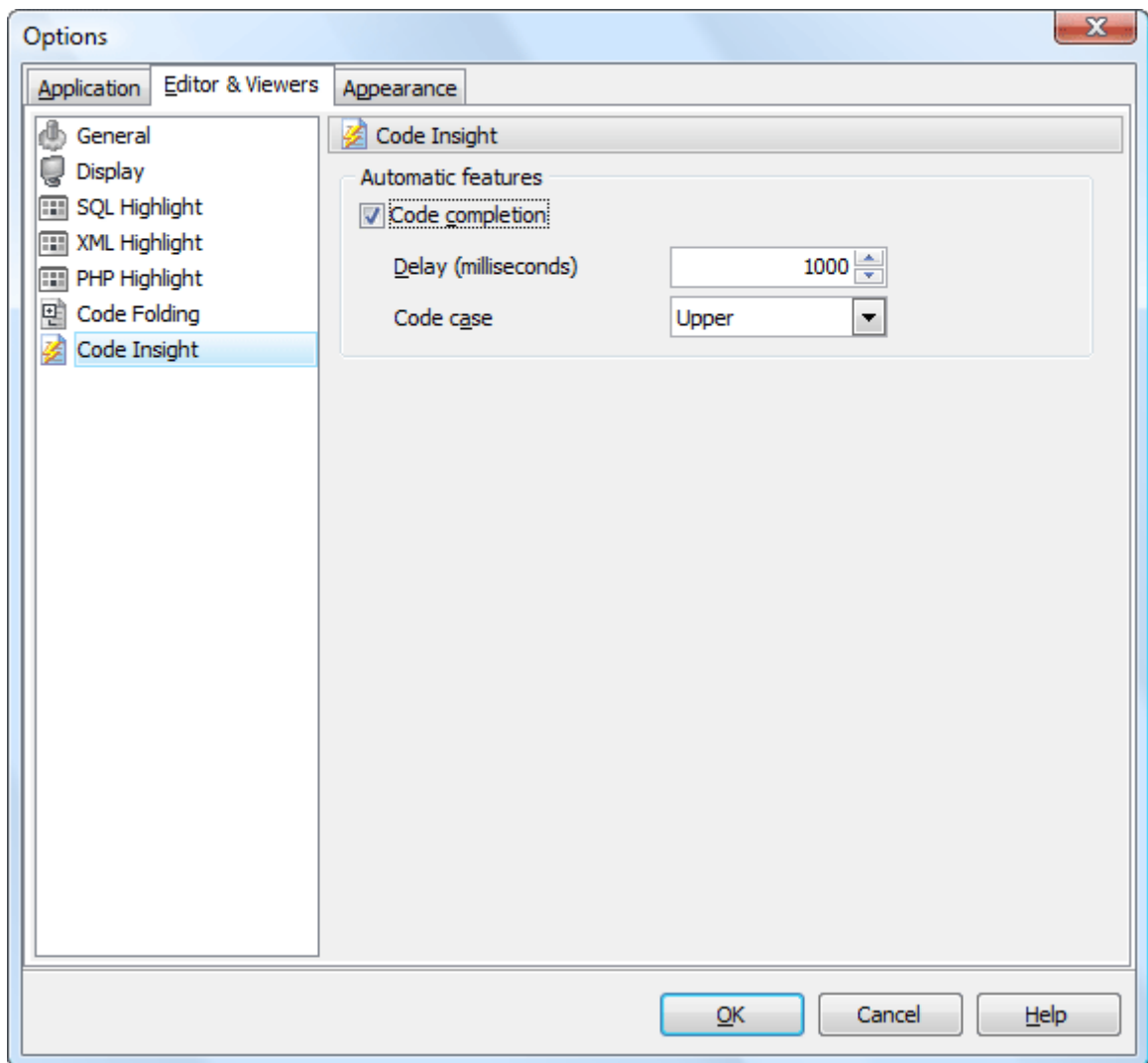
8.2.5 PHP highlight

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.



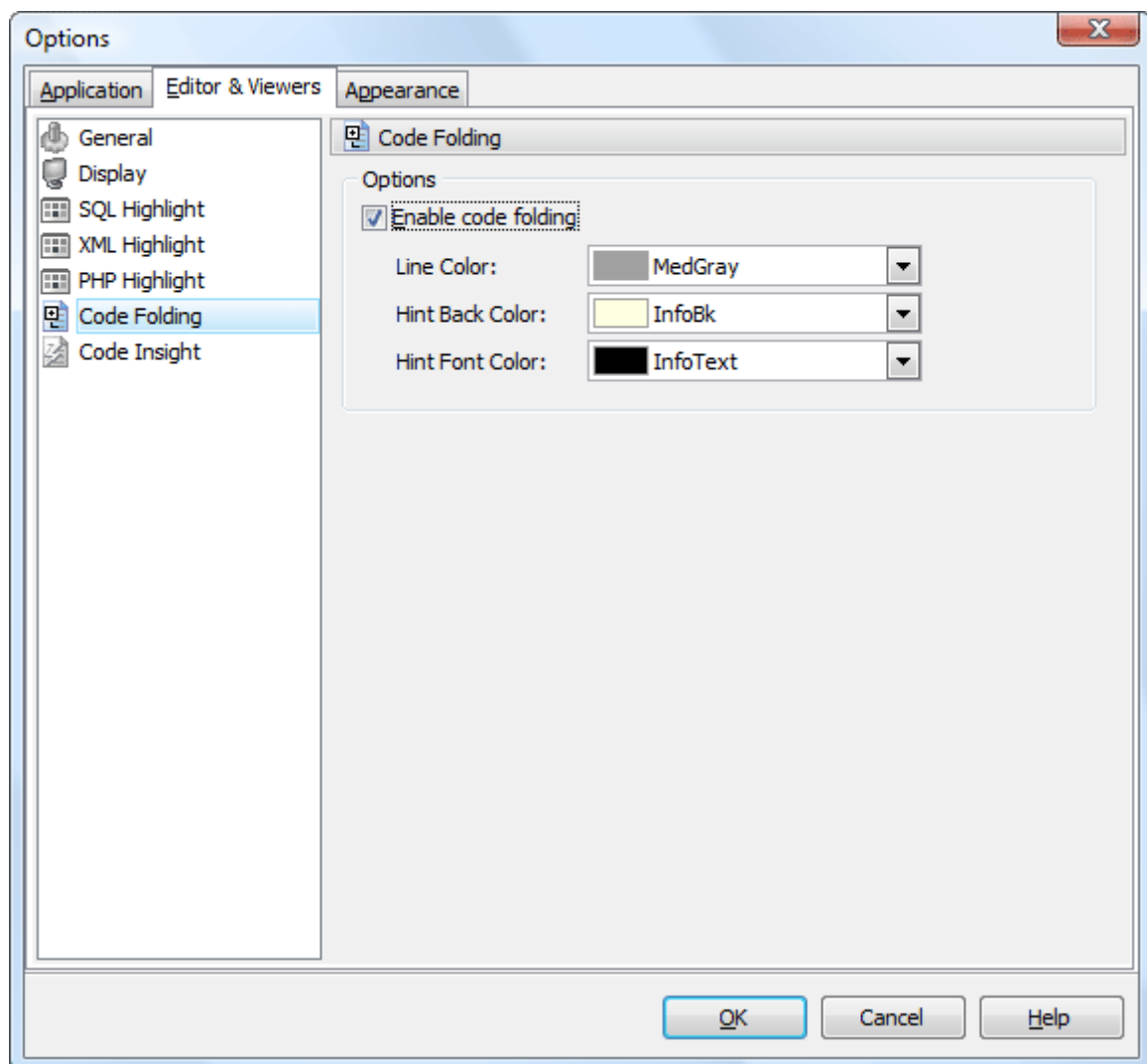
8.2.6 Code Insight

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.



8.2.7 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.

8.3 Appearance

The [Appearance](#) section allows you to customize the application interface style to your preferences.

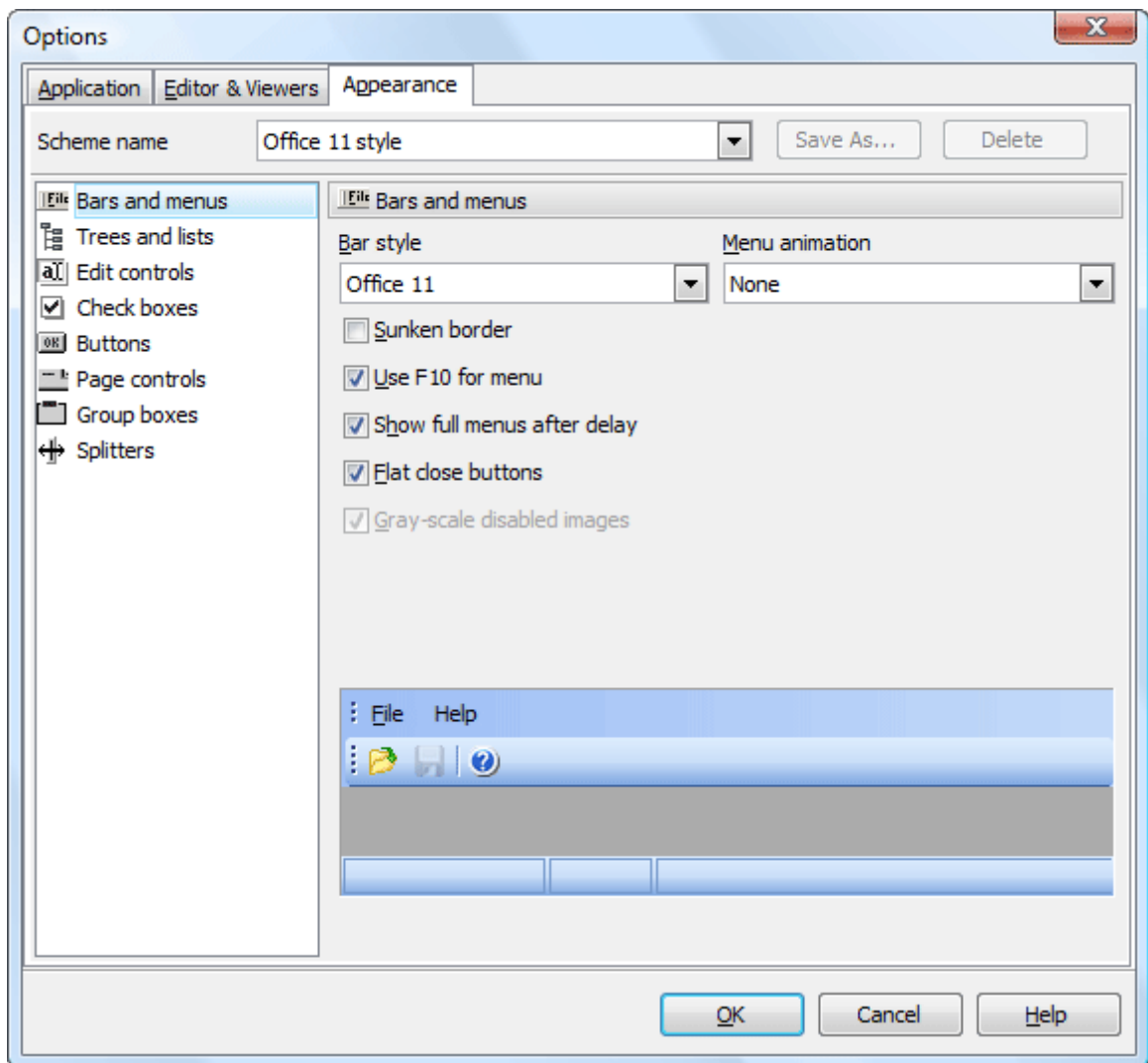
Use the [Scheme name](#) box to select the interface scheme you prefer: *Office XP style*, *Windows XP native style*, etc. You can create your own interface schemes by customizing any visual options ([Bars and menus](#), [Trees and lists](#), [Edit controls](#), [Check boxes](#), [Buttons](#), etc.) and clicking the [Save As](#) button. All the customized options are displayed on the sample panel.

- [Bars and menus](#) ¹³⁸
- [Trees and lists](#) ¹³⁹
- [Edit controls](#) ¹⁴⁰
- [Check boxes](#) ¹⁴¹
- [Buttons](#) ¹⁴²
- [Page controls](#) ¹⁴³
- [Group boxes](#) ¹⁴⁴
- [Splitters](#) ¹⁴⁵

8.3.1 Bars and menus

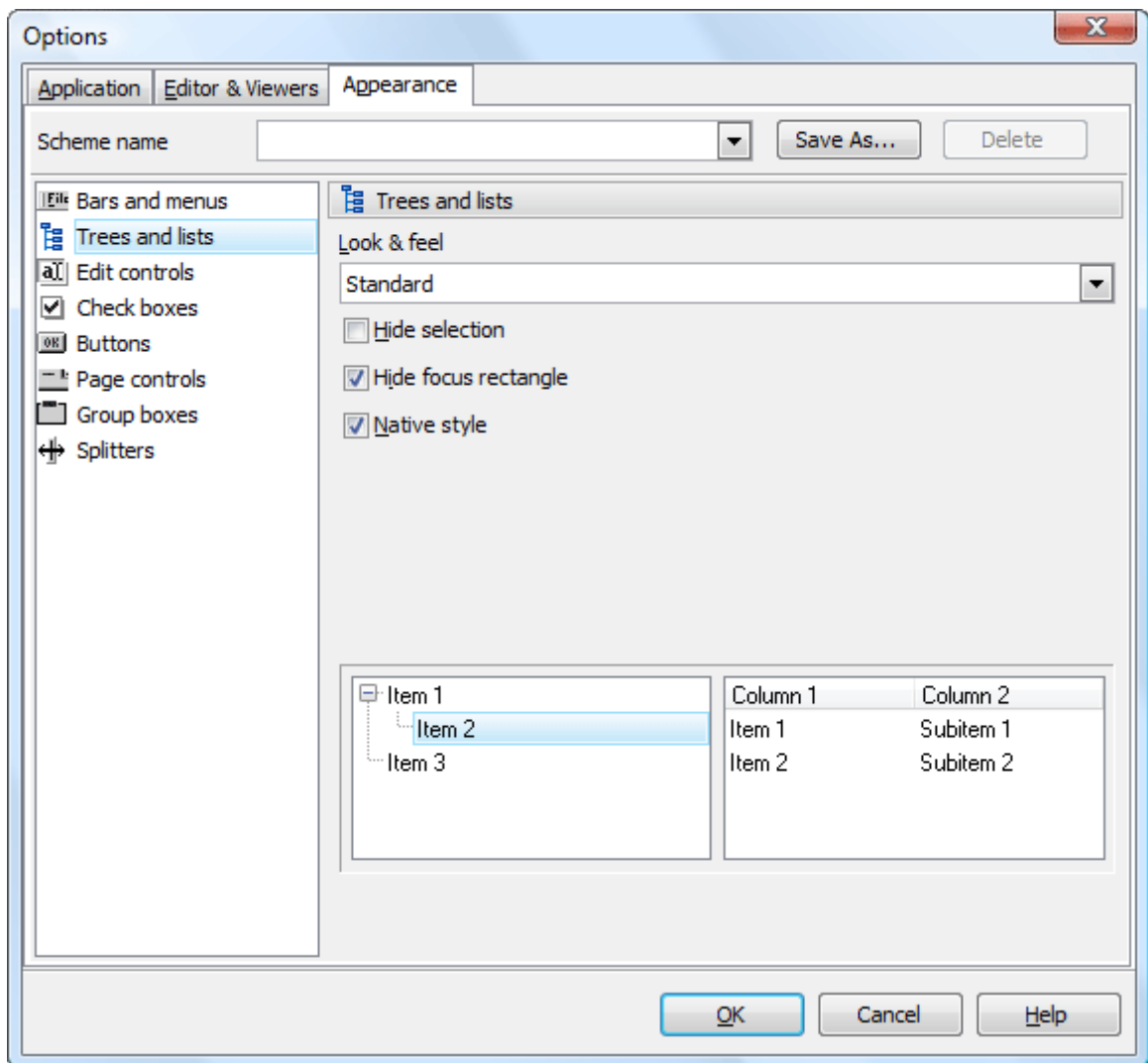
Use the [Bars and menus](#) item to customize PostgreSQL Code Factory toolbars style and menus animation.

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.



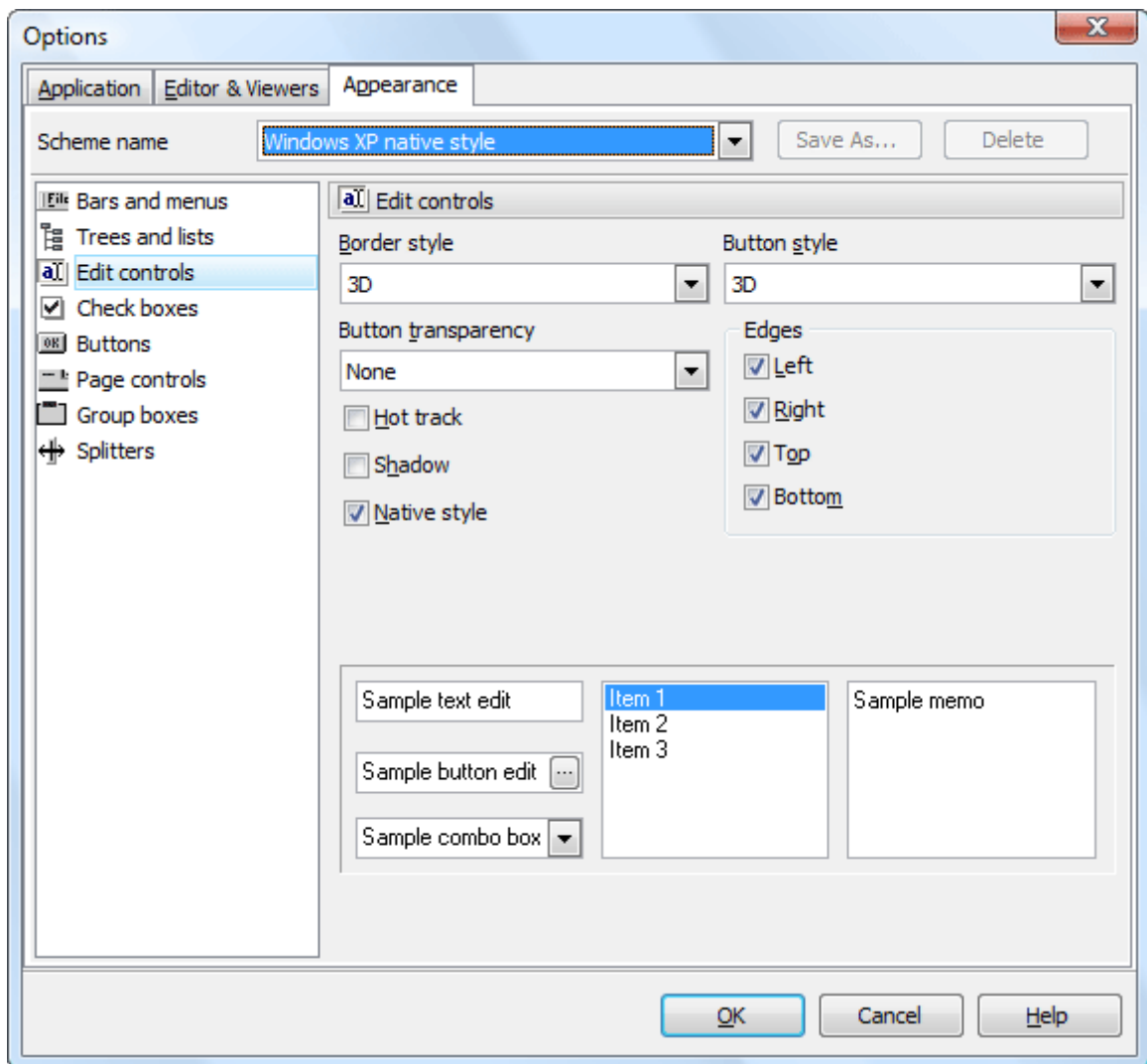
8.3.2 Trees and lists

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.



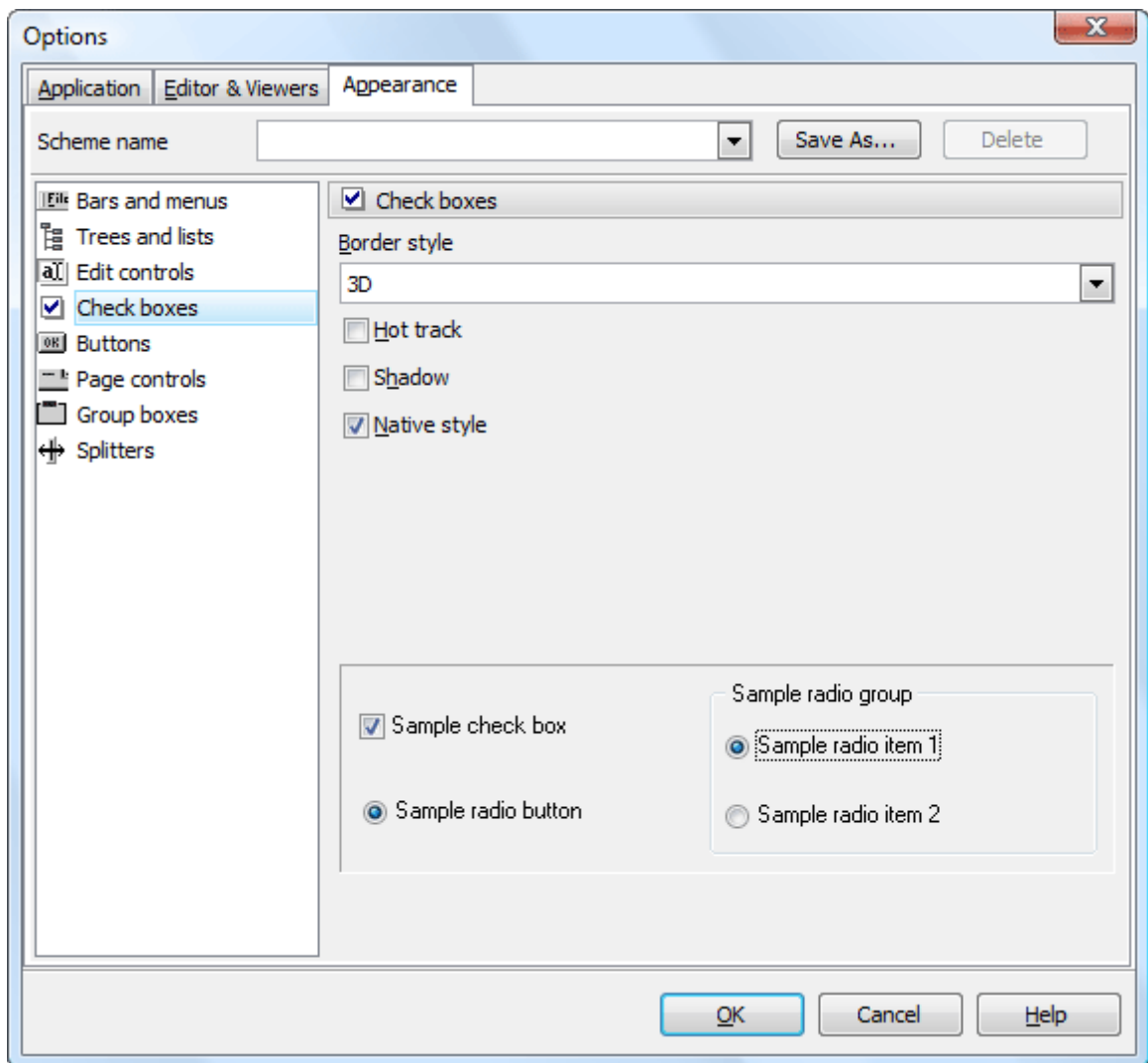
8.3.3 Edit controls

Use the [Edit controls](#) item to customize the appearance of different PostgreSQL Code Factory 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.



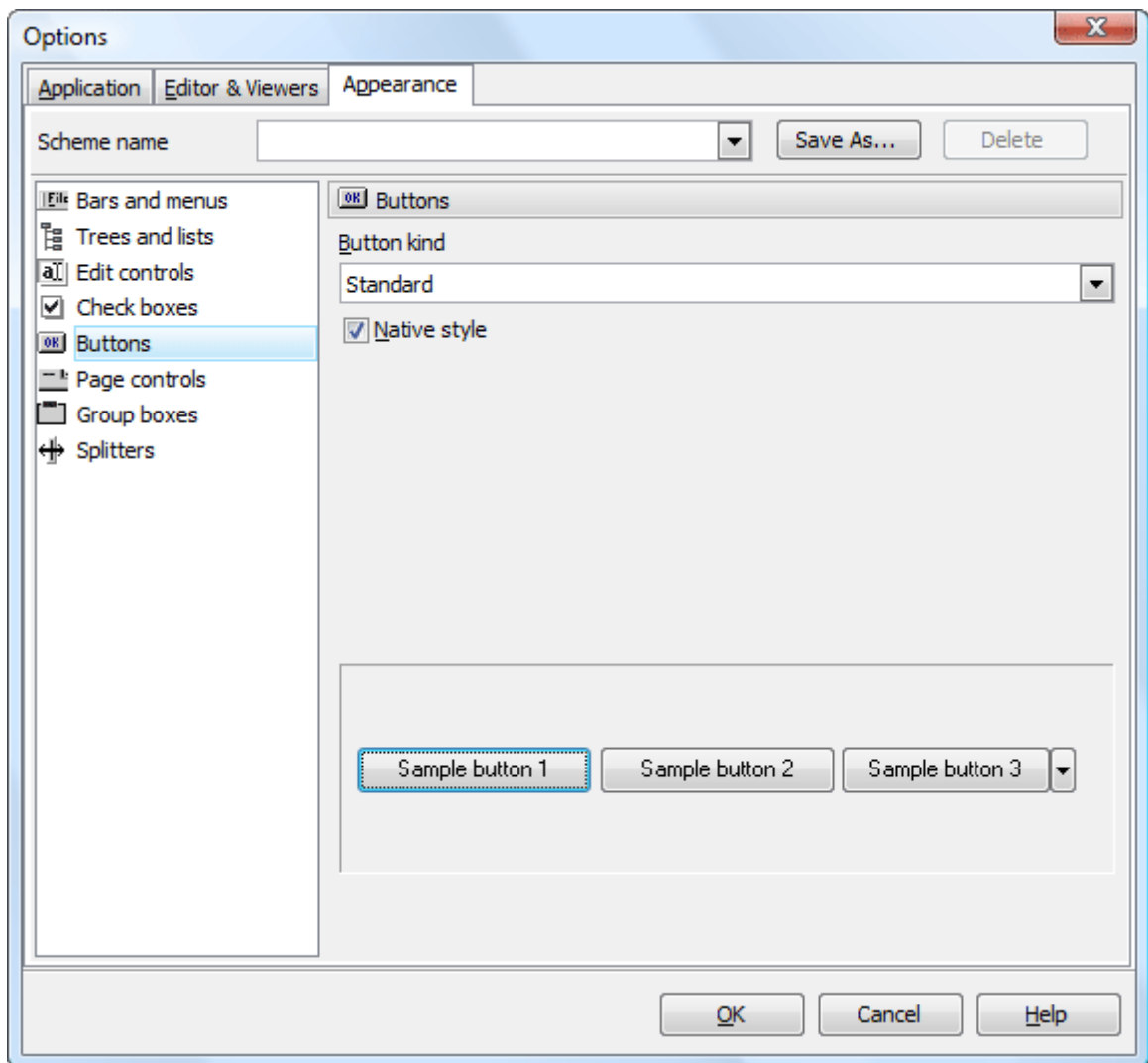
8.3.4 Check boxes

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.



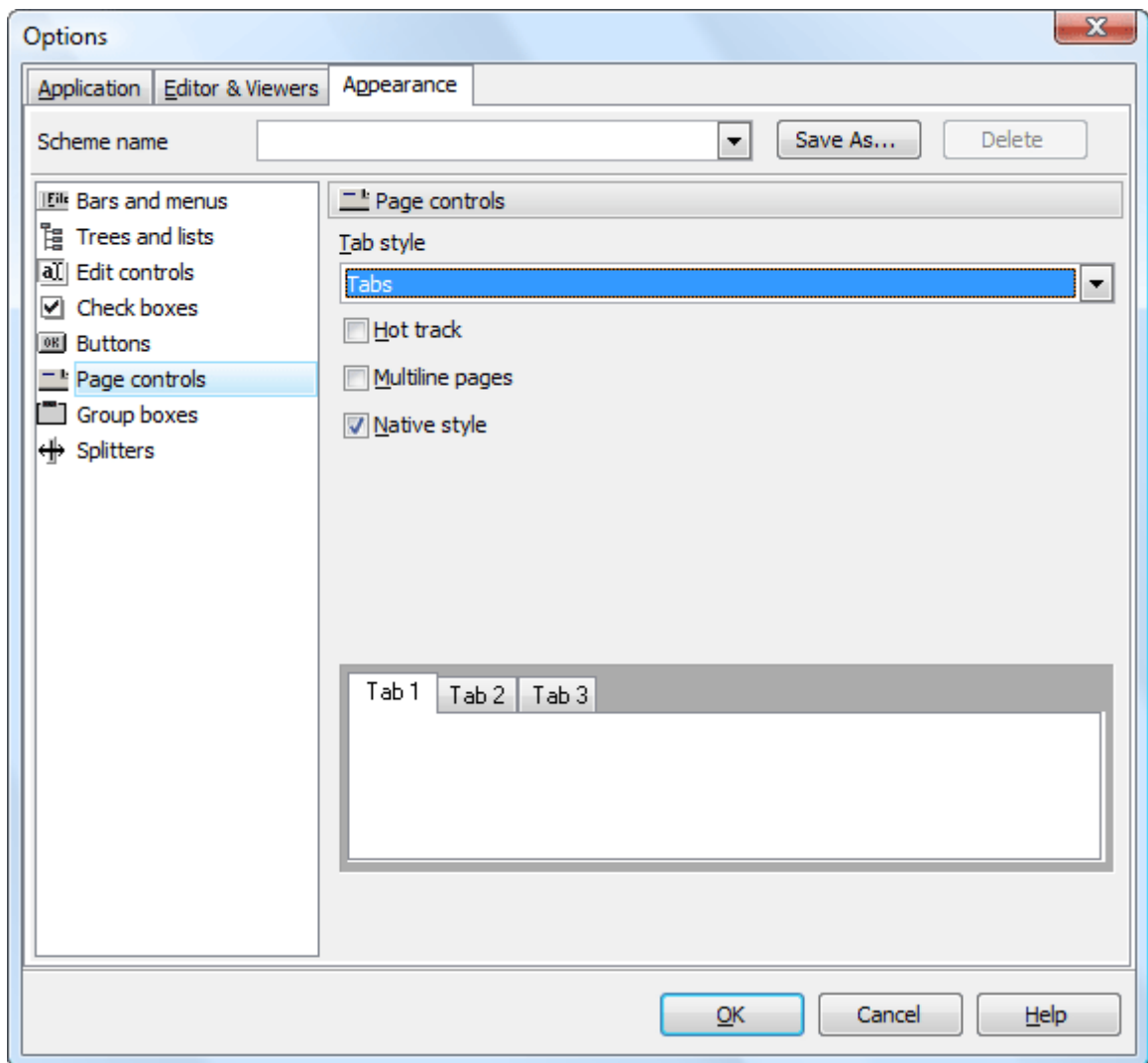
8.3.5 Buttons

Use the [Buttons](#) item to customize PostgreSQL Code Factory buttons. The tab allows you to adjust the appearance of buttons and define sample buttons as well.



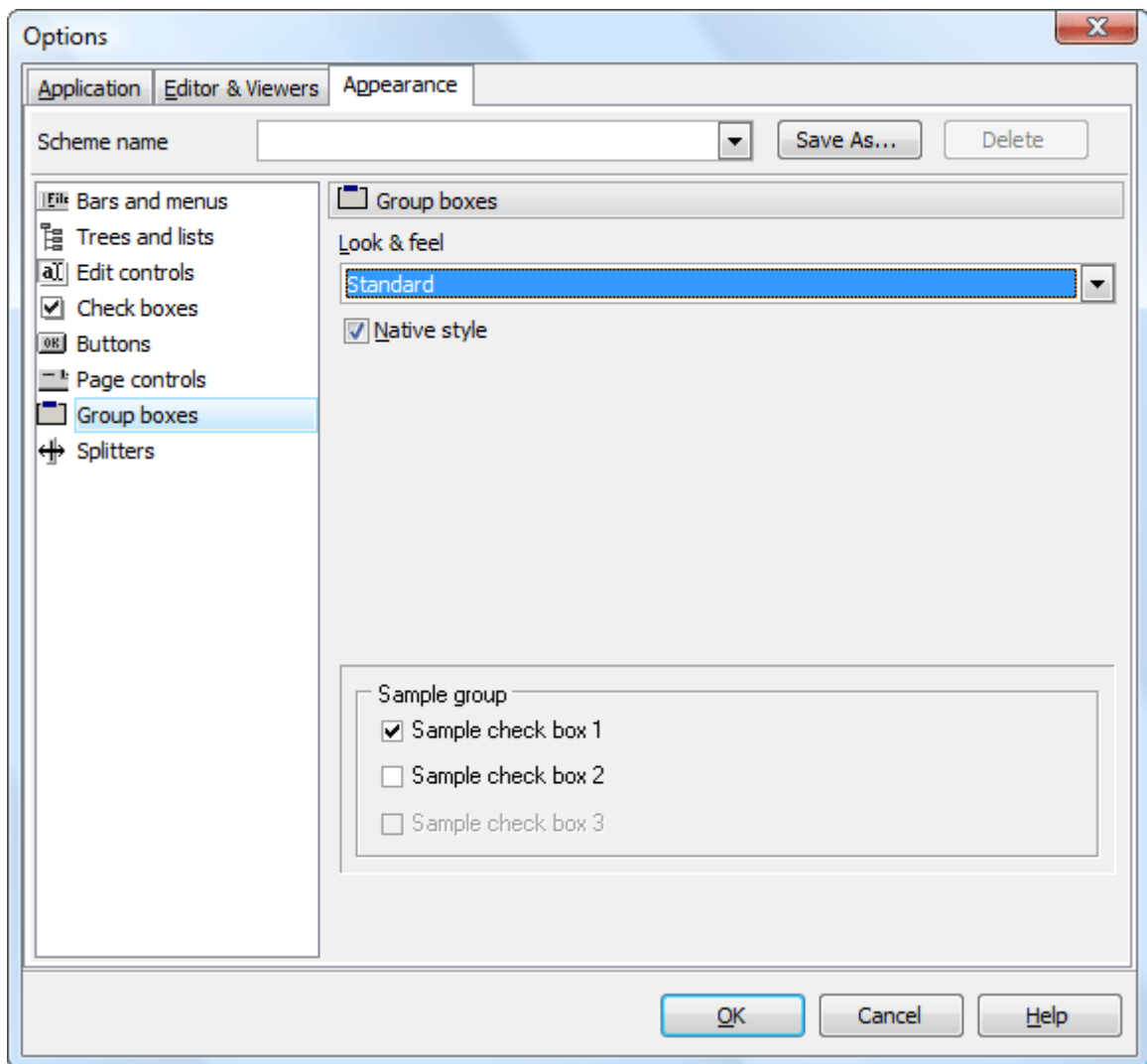
8.3.6 Page controls

The [Page controls](#) item allows you to customize the style of all PostgreSQL Code Factory page controls. The tab allows you to select tab styles, enable/disable hot track, multi-line pages and native style.



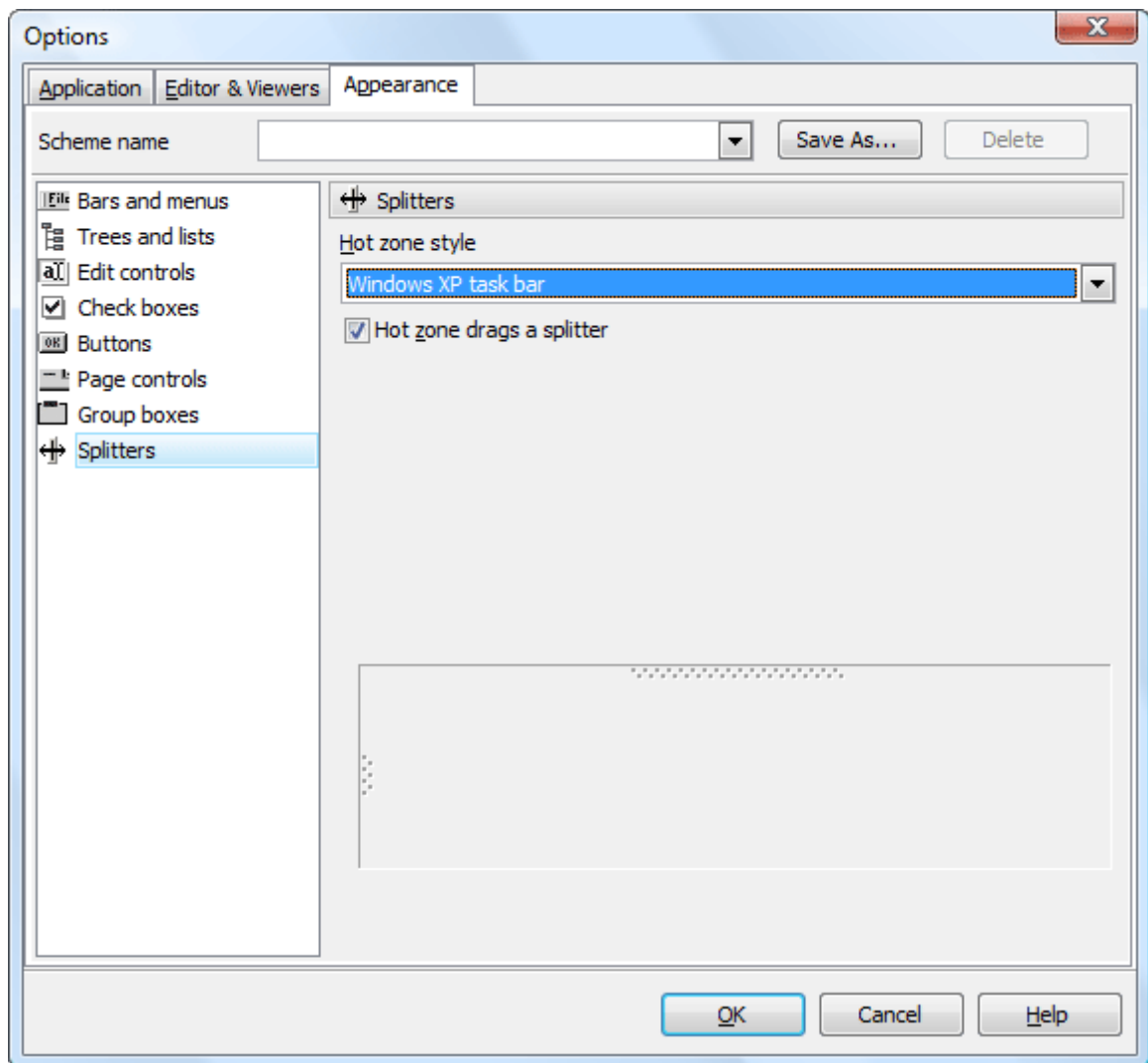
8.3.7 Group boxes

Use the [Group boxes](#) item to customize all PostgreSQL Code Factory group boxes according to your preferences. Use tab to apply styles for group boxes, enable/disable native style and define samples.



8.3.8 Splitters

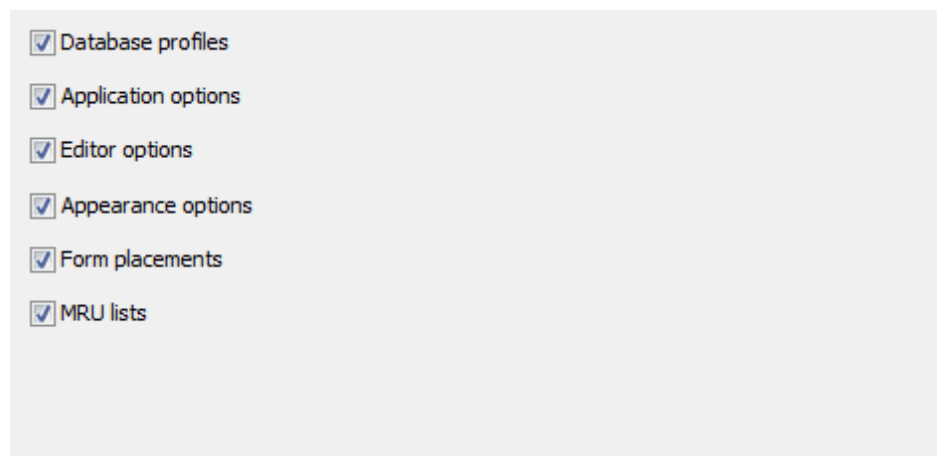
Use the [Splitters](#) item to customize all PostgreSQL Code Factory splitters according to your preferences. Use the tab to select hot zone style (*Windows XP task bar*, *Media Player 8*, *Media Player 9*, *Simple* or *none*) and specify the [Hot zone drags a splitter](#) option.



8.4 Export Settings

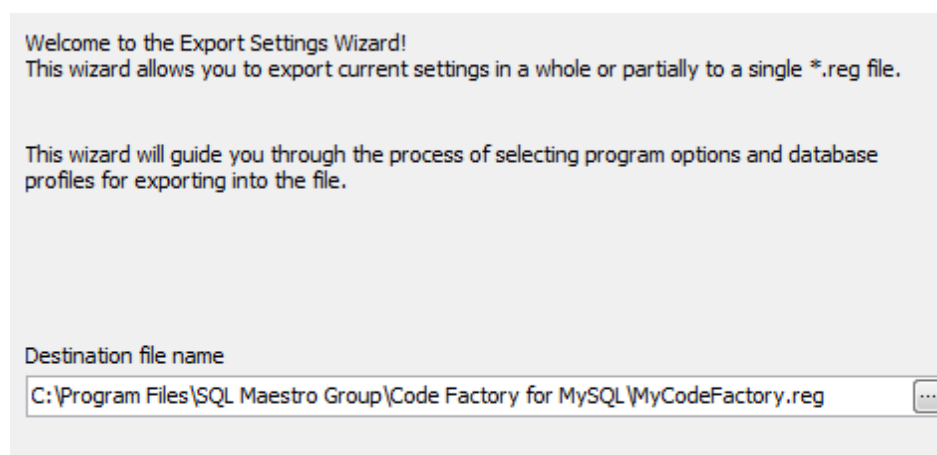
Export Settings Wizard allows you to export all or partial PostgreSQL Code Factory settings to single *.reg file which can be applied to the application of PostgreSQL Code Factory installed on another machine or used to backup previous settings. To run the wizard, select the Tools | Options main menu item and click Export Settings in the [Options](#)^[112] dialog.

- [Specifying destination file to save settings to](#)^[147]
- [Specifying settings categories to save](#)^[147]
- [Select database profiles to save](#)^[148]
- [Saving settings](#)^[148]



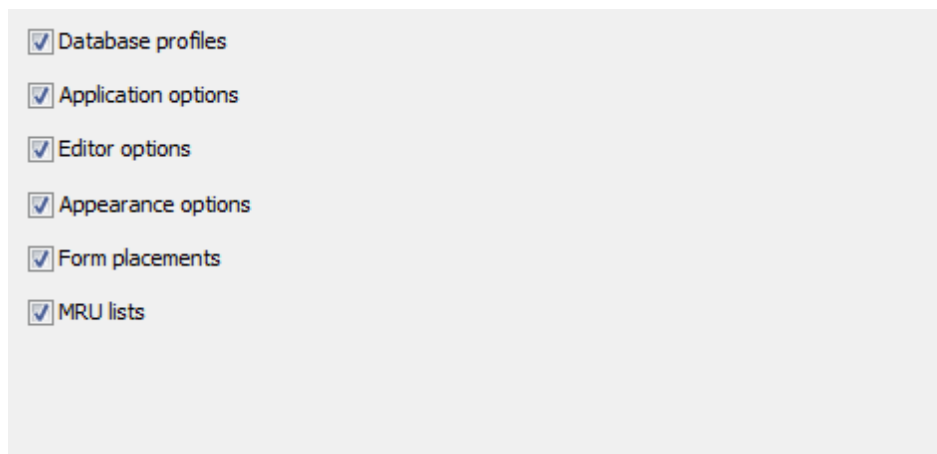
8.4.1 Specifying destination file

Specify a *.reg file to extract PostgreSQL Code Factory setting to.



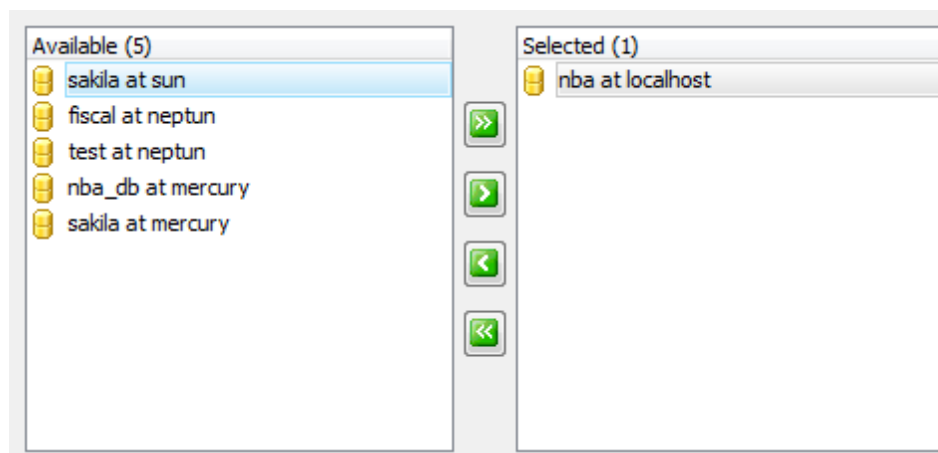
8.4.2 Selecting setting categories

The options of this step specify the information saved to the result file, e.g. Database profiles, [Application options](#)^[113], etc.



8.4.3 Selecting database profiles

Select database profiles to save their settings by moving them from the [Available Databases](#) list to the [Selected Databases](#) one.



8.4.4 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 138
 - Buttons 142
 - Check boxes 141
 - Edit controls 140
 - Group boxes 144
 - Page controls 143
 - Splitters 145
 - Trees and lists 139

- B -

- BLOB Editor 73
 - Hexadecimal mode 74
 - HTML mode 75
 - Image mode 73
 - Navigating with the BLOB editor 73
 - PDF mode 76
 - Plain text mode 74
- BLOB Viewer 97
 - Hexadecimal mode 97
 - HTML mode 100
 - Image mode 99
 - PDF mode 101
 - Plain text mode 98

- C -

- Create Database Profiles Wizard 25
 - Setting connection properties 25
 - Setting profile options 26

- D -

- Data Management 61
 - BLOB editor 73
 - Data View 62
 - Export Data Wizard 78
 - Get SQL Dump Wizard 85
 - Import Data Wizard 88

- Data View
 - Editing data in dialog 69
 - Lookup editor 69
 - Viewing data 62
 - Working with data grid 63
 - Working with info cards 68
- Database Explorer
 - Filtering explorer content 39
- Database Management
 - Creating a database profile 25
 - Editing a database profile 28
- Database Objects
 - Queries 42
- Database Profile Editor 28
 - Connection properties 28
 - Database options 29
 - Default directories 30
 - Editing obligatory scripts to execute 31
 - Setting log options 33
 - Statistics 34
- Diagram Viewer 103
 - Exporting diagram image 105
 - Selecting fields 104

- E -

- Editor & Viewer Options
 - Code Folding 136
 - Code Insight 135
 - Display 131
 - General 130
 - PHP highlight 134
 - SQL highlight 132
 - XML highlight 133
- EULA 5
- Export Data Wizard 78
 - Adjusting data formats 80
 - Selecting fields for export 80
 - Setting common export options 84
 - Setting destination file name 78
 - Setting format-specific options 81
 - Setting header and footer 79
- Export Settings Wizard
 - Saving settings 148
 - Selecting database profiles 148
 - Selecting setting categories 147
 - Specifying destination file 147

- F -

Filter Panel 39
Find Text Dialog 108

- G -

Get SQL Dump Wizard 85
 Selecting fields 85
 Specifying dump options 86
Getting Started 11
 Explaining user interface 17
 First time started 18
 Switching between windows 20
 Working with databases 12

- I -

Import Data Wizard 88
 Customizing common options 94
 Data Format 93
 Map builder 92
 Setting fields correspondence 91
 Setting source file name 89
Installation instructions 3

- L -

License Agreement 5

- O -

Object Management
 Managing Queries 42
Options 112
 Appearance 138
 Application 113
 Application confirmations 114
 Application preferences 113
 BLOB Viewer 121
 Data Grid 123
 Editor & Viewers 130
 Explorer 116
 Export 122
 Export Settings 147

Query Builder 119
SQL Editor 117
SQL Script Editor 118
Tools 115

- P -

PostgreSQL Code Factory 1
 Getting started 11
 Installation 3
 License agreement 5
 Registration 4
 System requirements 2
Purchase PostgreSQL Code Factory 4

- Q -

Queries 42
 Query Parameters 48
 SQL Editor 44
 Visual Query Builder 49
Query Parameters 48

- R -

Registration 4
Replace Text Dialog 109

- S -

SQL Editor 44
 Executing query 46
SQL Script Editor 59
System requirements 2

- T -

Tabbed MDI 17
Tools 96
 BLOB Viewer 97
 Diagram Viewer 103
 Dialogs 108
 Script Runner 58
 SQL Generator 107
 SQL Script Editor 59

- U -

User interface 17

- V -

Visual Query Builder 49

 Executing a query 56

 Working with editor area 55

- W -

Window List 20