

MySQL Workbench

MySQL Workbench

Abstract

This manual documents the MySQL Workbench SE version 5.2 and the MySQL Workbench OSS version 5.2.

If you have not yet installed MySQL Workbench OSS please download your free copy from the [download site](#). MySQL Workbench OSS is available for Windows, Mac OS X, and Linux.

Document generated on: 2010-03-24 (revision: 19659)

Copyright © 2006, 2010, Oracle and/or its affiliates. All rights reserved.

This documentation is NOT distributed under a GPL license. Use of this documentation is subject to the following terms: You may create a printed copy of this documentation solely for your own personal use. Conversion to other formats is allowed as long as the actual content is not altered or edited in any way. You shall not publish or distribute this documentation in any form or on any media, except if you distribute the documentation in a manner similar to how Sun disseminates it (that is, electronically for download on a Web site with the software) or on a CD-ROM or similar medium, provided however that the documentation is disseminated together with the software on the same medium. Any other use, such as any dissemination of printed copies or use of this documentation, in whole or in part, in another publication, requires the prior written consent from an authorized representative of Sun Microsystems, Inc. Sun Microsystems, Inc. and MySQL AB reserve any and all rights to this documentation not expressly granted above.

For more information on the terms of this license, for details on how the MySQL documentation is built and produced, or if you are interested in doing a translation, please contact <http://www.mysql.com/company/contact/>.

If you want help with using MySQL, please visit either the [MySQL Forums](#) or [MySQL Mailing Lists](#) where you can discuss your issues with other MySQL users.

For additional documentation on MySQL products, including translations of the documentation into other languages, and downloadable versions in variety of formats, including HTML, CHM, and PDF formats, see [MySQL Documentation Library](#).

Table of Contents

1. MySQL Workbench Introduction	1
2. MySQL Workbench Editions	2
3. Installing and Launching MySQL Workbench	3
3.1. Hardware Requirements	3
3.2. Software Requirements	3
3.3. Starting MySQL Workbench	4
3.3.1. Installing MySQL Workbench on Windows	4
3.3.2. Launching MySQL Workbench on Windows	4
3.3.3. Uninstalling MySQL Workbench on Windows	5
3.3.4. Installing MySQL Workbench on Linux	5
3.3.5. Launching MySQL Workbench on Linux	6
3.3.6. Uninstalling MySQL Workbench on Linux	6
3.3.7. Installing MySQL Workbench on Mac OS X	6
3.3.8. Launching MySQL Workbench on Mac OS X	7
3.3.9. Uninstalling MySQL Workbench on Mac OS X	7
3.4. Activation Procedure (Commercial Version)	7
4. Getting Started Tutorial	8
4.1. Creating a Model	8
4.2. Adding Data to Your Database	13
5. The Home Screen	17
5.1. Workbench Central	17
5.2. Workspace	17
5.3. Workbench Application Minimum Window Size	18
5.4. Workbench Preferences	18
5.4.1. The General Tab	19
5.4.2. The Model Tab	19
5.4.3. The MySQL Tab	20
5.4.4. The Diagram Tab	20
5.4.5. The Appearance Tab	21
6. SQL Development	22
6.1. Manage Connections	23
6.1.1. Standard TCP/IP Connection	24
6.1.2. Local Socket/Pipe Connection	25
6.1.3. Standard TCP/IP over SSH Connection	25
6.2. SQL Editor	26
6.2.1. Main Menu	27
6.2.2. Toolbar	28
6.2.3. SQL Statements Panel	28
6.2.4. Main Tabsheets	29
6.2.5. Sidebar	31
7. Data Modeling	34
7.1. Modeling Menus	34
7.1.1. The File Menu	35
7.1.2. The Edit Menu	35
7.1.3. The View Menu	38
7.1.4. The Arrange Menu	38
7.1.5. The Model Menu	39
7.1.6. The Database Menu	41
7.1.7. The Plugins Menu	41
7.1.8. The Scripting Menu	41
7.1.9. The Community Menu	41
7.1.10. The Help Menu	41
7.2. The Toolbar	42
7.2.1. Tool-specific Toolbar Items	43
7.3. Model Editor	43
7.3.1. EER Diagrams	44
7.3.2. The Physical Schemata	44
7.3.3. Schema Privileges	44
7.3.4. SQL Scripts and Model Notes	46
7.3.5. The Model Navigator Palette	46
7.3.6. The Catalog and Layers Palettes	46
7.3.7. The Properties and History Palettes	47
7.4. EER Diagram Editor	48
7.4.1. The Vertical Toolbar	48

7.5. Working with Models	51
7.5.1. Creating Tables	51
7.5.2. Creating Foreign Key Relationships	58
7.5.3. Creating Views	61
7.5.4. Creating Routines and Routine Groups	62
7.5.5. Creating Layers	65
7.5.6. Creating Notes	66
7.5.7. Creating Text Objects	67
7.5.8. Creating Images	68
7.5.9. Reverse Engineering	68
7.5.10. Forward Engineering	78
7.6. Modeling Tutorials	94
7.6.1. Importing a Data Definition SQL Script	94
7.6.2. Using the Default Schema	95
7.6.3. Basic Modeling	96
7.6.4. Documenting the <i>sakila</i> Database	97
7.7. Printing	99
7.7.1. Printing Options	99
7.8. MySQL Workbench Schema Validation Plugins (Commercial Version)	99
7.8.1. General Validation	99
7.8.2. MySQL-Specific Validation	100
7.9. Customizing DBDoc Model Reporting Templates	100
7.9.1. Supported Template Markers	103
7.9.2. Creating a custom template	105
8. Server Administration	108
8.1. Server Instances	108
8.1.1. Creating and Managing Server Instances	109
8.2. Server Administration and Configuration	112
8.2.1. Startup Tab	113
8.2.2. Configuration tab	114
8.2.3. Accounts tab	115
8.2.4. Connections tab	117
8.2.5. Variables tab	117
8.2.6. Data Dump tab	118
8.2.7. Logs tab	120
9. The Workbench Scripting Shell	122
9.1. Introduction	122
9.2. Exploring the Workbench Scripting Shell	122
9.3. The Shell	123
9.4. The Globals, Modules, and Classes Tabs	125
10. Plugins	126
10.1. The <i>catalog_utils.grt.lua</i> Module	126
10.1.1. Using the GRT Shell	126
10.2. Accessing Plugins	126
11. Keyboard Shortcuts	128
12. MySQL Workbench FAQ	131
A. MySQL Workbench Change History	133
A.1. Changes in Release 5.2	133
A.1.1. Changes in MySQL Workbench 5.2.17 (Not yet released beta)	133
A.1.2. Changes in MySQL Workbench 5.2.16 (17th February 2010 beta)	134
A.1.3. Changes in MySQL Workbench 5.2.15 (28th January 2010 beta)	138
A.1.4. Changes in MySQL Workbench 5.2.14 (21st January 2010 beta)	139
A.1.5. Changes in MySQL Workbench 5.2.13 (Not released beta)	139
A.1.6. Changes in MySQL Workbench 5.2.12 (Not released beta)	140
A.1.7. Changes in MySQL Workbench 5.2.11 (18th December 2009 beta)	143
A.1.8. Changes in MySQL Workbench 5.2.10 (1st December 2009 beta)	146
A.1.9. Changes in MySQL Workbench 5.2.9 (Internal release only beta)	147
A.1.10. Changes in MySQL Workbench 5.2.8 (18th November 2009 beta)	149
A.1.11. Changes in MySQL Workbench 5.2.7 (Internal release only alpha)	150
A.1.12. Changes in MySQL Workbench 5.2.6 (21st October 2009 alpha)	151
A.1.13. Changes in MySQL Workbench 5.2.5 (Internal Release Only alpha)	152
A.1.14. Changes in MySQL Workbench 5.2.4 (7th October 2009 alpha)	152
A.1.15. Changes in MySQL Workbench 5.2.3 (15th September 2009 alpha)	153
A.1.16. Changes in MySQL Workbench 5.2.2 (27th July 2009 alpha)	155
A.1.17. Changes in MySQL Workbench 5.2.1 (22th May 2009 alpha)	155
A.1.18. Changes in MySQL Workbench 5.2.0 (30th April 2009 alpha)	155
A.2. Changes in Release 5.1	156
A.2.1. Changes in MySQL Workbench 5.1.19 (Not yet released GA)	156
A.2.2. Changes in MySQL Workbench 5.1.18 (3rd September 2009 GA)	156
A.2.3. Changes in MySQL Workbench 5.1.17 (14th August 2009 beta)	158

A.2.4. Changes in MySQL Workbench 5.1.16 (30th June 2009 beta)	162
A.2.5. Changes in MySQL Workbench 5.1.15 (26th June 2009 beta)	163
A.2.6. Changes in MySQL Workbench 5.1.14 (19th June 2009 beta)	164
A.2.7. Changes in MySQL Workbench 5.1.13 (12th June 2009 beta)	164
A.2.8. Changes in MySQL Workbench 5.1.12 (27th April 2009 beta)	168
A.2.9. Changes in MySQL Workbench 5.1.11 (Not yet released beta)	168
A.2.10. Changes in MySQL Workbench 5.1.10 (10th April 2009 beta)	168
A.2.11. Changes in MySQL Workbench 5.1.9 (Not yet released beta)	169
A.2.12. Changes in MySQL Workbench 5.1.8 (Not yet released)	170
A.2.13. Changes in MySQL Workbench 5.1.7 (Not yet released)	170
A.2.14. Changes in MySQL Workbench 5.1.6 (Not yet released)	170
A.2.15. Changes in MySQL Workbench 5.1.4 (Not yet released)	171
A.3. Changes in Release 5.0	171
A.3.1. Changes in MySQL Workbench 5.0.30 (18th February 2009)	171
A.3.2. Changes in MySQL Workbench 5.0.29 (12th December 2008)	172
A.3.3. Changes in MySQL Workbench 5.0.28 (6th December 2008)	172
A.3.4. Changes in MySQL Workbench 5.0.27 (7th November 2008)	173
A.3.5. Changes in MySQL Workbench 5.0.26 (16th October 2008)	174
A.3.6. Changes in MySQL Workbench 5.0.25 (12th September 2008)	174
A.3.7. Changes in MySQL Workbench 5.0.24 (12th August 2008)	175
A.3.8. Changes in MySQL Workbench 5.0.23 (25th June 2008)	176
A.3.9. Changes in MySQL Workbench 5.0.22 (27th May 2008)	177
A.3.10. Changes in MySQL Workbench 5.0.21 (27th April 2008)	178
A.3.11. Changes in MySQL Workbench 5.0.20 (26th April 2008)	178
A.3.12. Changes in MySQL Workbench 5.0.19 (15th April 2008)	178
A.3.13. Changes in MySQL Workbench 5.0.18rc (not released)	178
A.3.14. Changes in MySQL Workbench 5.0.17rc (07th April 2008)	179
A.3.15. Changes in MySQL Workbench 5.0.16rc (26th March 2008)	180
A.3.16. Changes in MySQL Workbench 5.0.15rc (17th March 2008)	181
A.3.17. Changes in MySQL Workbench 5.0.14abeta (28th February 2008)	183
A.3.18. Changes in MySQL Workbench 5.0.14beta (25th February 2008)	183

List of Figures

3.1. MySQL Workbench Mac OS X Installation Screen	7
4.1. Getting Started Tutorial - Home Screen	8
4.2. Getting Started Tutorial - New Schema	8
4.3. Getting Started Tutorial - Columns	9
4.4. Getting Started Tutorial - EER Diagram	10
4.5. Getting Started Tutorial - Manage Connections	11
4.6. Getting Started Tutorial - Review Script	12
4.7. Getting Started Tutorial - Edit Table Data	13
4.8. Getting Started Tutorial - Edit Data	14
4.9. Getting Started Tutorial - Results	15
5.1. The Home screen	17
5.2. The Preferences Dialog Box	18
6.1. Manage DB Connections - Dialog	23
6.2. Manage DB Connections - Advanced Tab	24
6.3. Manage DB Connections - Socket/Pipe Parameters	25
6.4. Manage DB Connections - SSH Parameters	26
6.5. SQL Editor	26
6.6. SQL Editor - Toolbar	28
6.7. SQL Editor - Statements Panel	29
6.8. SQL Editor - Main Tabsheets	29
6.9. SQL Editor - Results Tabsheets	30
6.10. SQL Editor - Results Tabsheets Navigation Controls	30
6.11. SQL Editor - Connection Information Palette	31
6.12. SQL Editor - Default Schema Listbox	31
6.13. SQL Editor - Schemata Explorer	32
6.14. SQL Editor - Snippets Palette	32
7.1. The Find Window	36
7.2. The MySQL Model Page	43
7.3. Roles and Privileges	45
7.4. Role Editor	45
7.5. The Model Navigator Palette	46
7.6. The Vertical Toolbar	48
7.7. A Table on an EER Diagram	52
7.8. The Table Editor	53
7.9. The Columns Tab	54
7.10. The Indexes Tab	55
7.11. The Relationship Connector	59
7.12. The Split Connector	60
7.13. The Layer Object	66
7.14. Reverse Engineer Database Wizard	69
7.15. Connect to DBMS	70
7.16. Select Schemata	71
7.17. Fetch Object Info	72
7.18. Select Objects	73
7.19. Show Filter	74
7.20. Progress	75
7.21. Results	76
7.22. Message Log	77
7.23. SQL Export Options	78
7.24. SQL Object Export Filter	79
7.25. Review Generated Script	80
7.26. Options	81
7.27. Script	82
7.28. Catalog Validation	83
7.29. Select Objects to Forward Engineer	84
7.30. Set parameters for connecting to a DBMS	85
7.31. Set parameters for connecting to a DBMS	86
7.32. Model and Database Differences	87
7.33. Controlling Synchronization Direction	88
7.34. Update Model Button	89
7.35. Ignore Button	90
7.36. Update Source Button	91
7.37. Click arrows to change direction of synchronization	91
7.38. Catalog Sources	92

7.39. Catalog Diff Report	93
7.40. Adding Tables to the Canvas	95
7.41. The Default Schema	96
7.42. Adding an EER Diagram	96
7.43. The <code>sakila</code> EER Diagram	98
8.1. Manage Server Instances Dialog	110
8.2. Server Instances Wizard	112
8.3. MySQL Workbench - Admin page	113
8.4. Administrator - Startup tab	114
8.5. Administrator - Configuration tab	115
8.6. Administrator - Accounts tab	115
8.7. Administrator - Connections tab	117
8.8. Administrator - Variables tab	117
8.9. Administrator - Export to Disk	118
8.10. Administrator - Import from Disk	119
8.11. Administrator - Advanced Options	120
8.12. Administrator - Logs tab	121
9.1. The Workbench Scripting Shell	122
9.2. The Workbench Scripting Shell Panels	123

Chapter 1. MySQL Workbench Introduction

MySQL Workbench provides a graphical tool for working with MySQL Servers and databases. MySQL Workbench supports MySQL Server versions 5.0 and above. It does not support MySQL Server versions 4.x. MySQL Workbench provides three main areas of functionality:

- Database Connections
- Server Instances
- Database Models

This section provides a brief overview of this functionality.

1. **Database Connections** - Allows you to create and manage connections to database servers. As well as allowing you to configure connection parameters, MySQL Workbench provides the capability to execute SQL queries on the database connections using the built-in SQL Editor. This functionality replaces that previously provided by the Query Browser stand-alone application.
2. **Server Instances** - Allows you to create and administer server instances. This functionality replaces that previously provided by the MySQL Administrator stand-alone application.
3. **Database Models** - Allows you to create models of your database schema graphically, reverse and forward engineer between a schema and a live database, and edit all aspects of your database using the comprehensive Table Editor. The Table Editor provides easy-to-use facilities for editing Tables, Columns, Indexes, Triggers, Partitioning, Options, Inserts and Privileges, Routines and Views.

MySQL Workbench is available in two editions. The Community Edition and the Standard Edition. The Community Edition is available free of charge. The Standard Edition provides additional Enterprise features, such as database documentation generation, at a low cost.

Chapter 2. MySQL Workbench Editions

The Community Edition (OSS)

The Community Edition is the foundation of all MySQL Workbench editions—versions that are currently available or those that will become available in the future. All editions of MySQL Workbench are based on the Community Edition and all future improvements to the base framework and feature set will be included in this version. The Community Edition is a full feature product that puts a powerful database management tool into the hands of the MySQL community.

The Standard Edition

The Standard Edition is a commercial extension that builds on top of the OSS Edition and adds modules and plugins, allowing for an optimized work flow. The highlights of this edition are the added schema object privilege system, schema validation plugins, model reporting, and online printing. If you use the MySQL Workbench in a professional environment upgrading to the commercial edition can greatly improve your work flow.

A comparison of edition features can be found at [MySQL Workbench Developer Central](#).

Chapter 3. Installing and Launching MySQL Workbench

MySQL Workbench is available for the following platforms:

- Windows
- Linux
- Mac OS X

Binary distributions of MySQL Workbench are available for the above platforms. Source code distributions are also available as a [tar.gz](#) package, or an RPM package.

The following sections explain the installation process for each of these platforms.

3.1. Hardware Requirements

MySQL Workbench requires a current system to run smoothly. The minimum hardware requirements are:

- CPU: Intel Core or Xeon 3GHz (or Dual Core 2GHz) or equal AMD CPU
- Cores: Single (Dual/Quad Core is recommended)
- RAM: 4 GB (6 GB recommended)
- Graphic Accelerators: nVidia or ATI with support of OpenGL 1.5 or higher
- Display Resolution: 1280×1024 is recommended, 1024×768 is minimum.

3.2. Software Requirements

The following operating systems are officially supported:

- Windows 7 (64-bit, Professional level or higher)
- Mac OS X 10.6.1+
- Ubuntu 9.10 (64bit)
- Ubuntu 8.04 (32bit/64bit)

For convenience the following builds are also available:

- Windows XP SP3, Vista
- Mac OSX (10.5 and 10.6) Intel
- Ubuntu 8.04 (i386/x64)
- Ubuntu 9.04 (i386/x64)
- Fedora 11 (i386/x64)

MySQL Workbench also has the following general requirements:

1. The Microsoft .NET 3.5 Framework.
2. Cairo 1.6.0 or later
3. glib-2.10

4. libxml-2.6
5. libsigc++ 2.0
6. pcre
7. libzip

Note

For convenience the Windows libraries are available as the download “Dependencies for Compiling in Windows”.

Note

On start up, the application checks the OpenGL version and selects between software and hardware rendering. To determine the rendering method that is being used, open the [HELP](#) menu and choose the [SYSTEM INFO](#) submenu.

3.3. Starting MySQL Workbench

The procedure for launching MySQL Workbench depends on the platform. Generally, there are two ways to launch MySQL Workbench from the command line and from the graphical user interface of the host operating system. Using the command-line launching facility is useful when you want to customize some aspects of the way MySQL Workbench operates. Launching MySQL Workbench for each of the supported platforms is described in the following sections.

3.3.1. Installing MySQL Workbench on Windows

MySQL Workbench may be installed using the Windows installer file or it may be installed manually from a ZIP file.

Installing MySQL Workbench Using the Installer

MySQL Workbench can be installed using the Windows Installer ([.msi](#)) installation package. The MSI package bears the name [mysql-workbench-version-win32.msi](#), where *version* indicates the MySQL Workbench version number.

Important

Installing MySQL Workbench using the installer requires either Administrator or Power User privileges. If you are using the ZIP file without an installer, you do not need Administrator or Power User privileges.

Improving the MySQL Installation Wizard depends on the support and feedback of users. If you find that the MySQL Installation Wizard is lacking some feature important to you, or if you discover a bug, please report it in our bugs database. To do this use the [REPORT A BUG](#) option under the [HELP](#) menu.

1. To install MySQL Workbench, right click on the MSI file and select the [INSTALL](#) option from the pop-up menu, or simply double-click the file.
2. In the **SETUP TYPE** window you may choose a [Complete](#) or [Custom](#) installation. To use all features of MySQL Workbench choose the [Complete](#) option.
3. Unless you choose otherwise, MySQL Workbench is installed in [C:\%PROGRAMFILES%\MySQL\MySQL Workbench 5.1 edition_type\](#), where [%PROGRAMFILES%](#) is the default directory for programs for your locale. The [%PROGRAMFILES%](#) directory may be [C:\Program Files](#) or [C:\programme](#).

Installing from the ZIP File

If you are having problems running the installer, as an alternative, you can download a ZIP file without an installer. That file is called [mysql-workbench-version-win32.zip](#). Using a ZIP utility, unpack it to the directory of your choice. You may also want to create a shortcut on your desktop or the quick launch bar.

To install using the ZIP file, download the ZIP file to a convenient location and decompress the file. You can place the resulting directory anywhere on your system. You do not need to install or configure the application before using it.

3.3.2. Launching MySQL Workbench on Windows

To start MySQL Workbench on Windows select [START](#), [PROGRAMS](#), [MYSQL](#) and then select MySQL Workbench.

You may also start MySQL Workbench from the command line. To view the available command-line options, issue the command

`MySQLWorkbench -help | more` from the MySQL Workbench installation directory. You will see the following output:

```
MySQL Workbench 5.1.12 OSS. (C) 2006-2009 by Sun Microsystems.  
All rights reserved.  
Usage: MySQLWorkbench [options] [model file]  
Options  
-help (-h) ..... Print this output  
-open filename .. Open the given filename at startup  
-nologo ..... Do not display the splash screen  
-verbose (-v) ... Print verbose output in the GRT Shell  
-version ..... Print the version information  
-grtversion ..... Print the GRT version information  
-swrendering .... Force the canvas to use software rendering instead of OpenGL  
-log ..... Instruction to save messages (other debug info) to file
```

The MySQL Workbench version number is displayed followed by a usage message and then the options. Use the `-swrendering` option if your video card does not support OpenGL 1.5. The `-version` option can be used to display the MySQL Workbench version number. The `-grtversion` can be used to display the GRT shell version number. The other options are self-explanatory.

Note

When using command-line options that display output to a console window, namely `-help` and `-version`, be sure that you pipe the output through the `more` command otherwise nothing will be displayed.

3.3.3. Uninstalling MySQL Workbench on Windows

The method for uninstalling MySQL Workbench will depend on how you install MySQL Workbench in the first place.

Removing MySQL Workbench when installed Using the Installer

1. To uninstall MySQL Workbench, open the **CONTROL PANEL** and Choose **ADD OR REMOVE PROGRAMS**. Find the MySQL Workbench entry and choose the REMOVE button. Doing this will remove MySQL Workbench.
2. Any modules added to the `C:\Program Files\MySQL\MySQL Workbench version\modules` directory will **not** be deleted.

Note

It is not possible to remove MySQL Workbench from the command line if you have installed MySQL Workbench using the installer. Although you can manually remove some of the components There is no command-line option for removing MySQL Workbench.

Removing the MySQL Workbench directory manually will not remove all the files belonging to MySQL Workbench.

When installed from a ZIP file

If you installed MySQL Workbench using a ZIP file, to remove MySQL Workbench you can just delete the MySQL Workbench directory.

Note

If you installed any additional modules within the `modules` directory and you want to keep them, make sure you copy those modules to a different directory before deleting the MySQL Workbench directory.

3.3.4. Installing MySQL Workbench on Linux

There are several binary distributions of MySQL Workbench available for Linux. These include:

- Fedora 10 amd64 (RPM)
- Ubuntu 8.04 i386 (DEB)
- Ubuntu 8.10 amd64 (DEB)

In addition to the binary distributions, it is also possible to download the MySQL Workbench source code as a `tar.gz` or RPM package.

Check the MySQL Workbench [download page](#) for the latest packages.

The procedure for installing on Linux depends on which Linux distribution you are using.

Installing DEB packages

On Ubuntu, and other systems that use the Debian package scheme, you can install MySQL Workbench using a command such as:

```
shell> sudo dpkg -i package.deb
```

Note that `package.deb` will be the MySQL Workbench package, for example, `mysql-workbench-oss-version_i386.deb`, where `version` is the MySQL Workbench version number.

Note

You may be warned that certain libraries are not available, depending on what you already have installed. Install the required libraries and then install the MySQL Workbench package again.

Installing RPM packages

On RedHat-based systems, and other systems using the RPM package format, MySQL Workbench can be installed by a command such as:

```
shell> sudo rpm -i package.rpm
```

Again, note that `package.rpm` will be the MySQL Workbench package, for example, `mysql-workbench-oss-version-ifc10.x86_64.rpm`, and `version` is the MySQL Workbench version number.

3.3.5. Launching MySQL Workbench on Linux

Once MySQL Workbench has been installed it can be launched by selecting APPLICATIONS, PROGRAMMING, MySQL Workbench from the main menu.

MySQL Workbench can also be launched from the command line on Linux. Type the command:

```
shell> /usr/bin/mysql-workbench --help
```

This will display the available command-line options:

```
mysql-workbench [<options>] [<model file>]
Options:
  --force-sw-render      Force Xlib rendering
  --force-opengl-render  Force OpenGL rendering
  --help, -h              Show command line options and exit
```

3.3.6. Uninstalling MySQL Workbench on Linux

The procedure for uninstalling MySQL Workbench on Linux depends on the packe you are using.

Uninstalling DEB packages

For Debian packages the command is:

```
shell> sudo dpkg -r mysql-workbench-oss
```

This does not remove the configuration files. If you wish to also remove the configuration files use:

```
shell> sudo dpkg --purge mysql-workbench-oss
```

Uninstalling RPM packages

To uninstall RPM packages use:

```
shell> sudo rpm -e mysql-workbench-oss
```

This does not remove the configuration files.

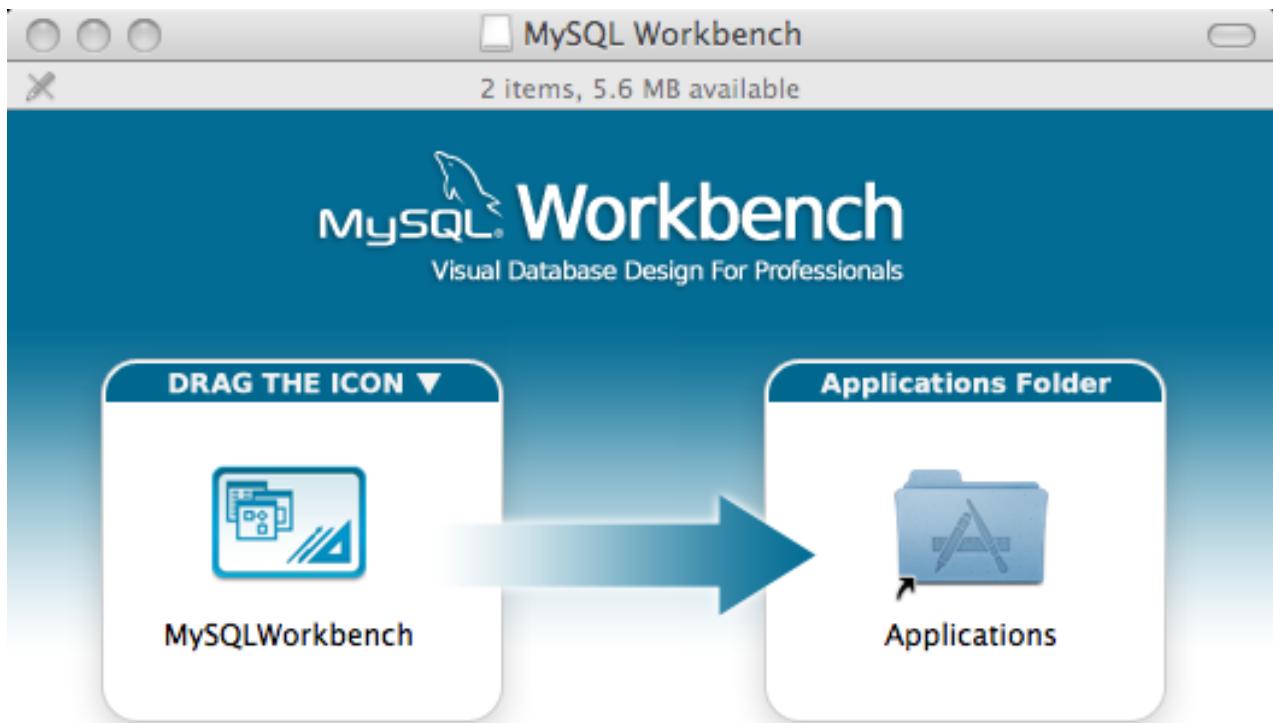
3.3.7. Installing MySQL Workbench on Mac OS X

MySQL Workbench is available for Mac OS X and is distributed as a DMG file. The file is named

`version-osx10.5-i686.dmg`, where `version` is the MySQL Workbench version.

To install MySQL Workbench on Mac OS X, simply download the file. Double-click the downloaded file. You will be presented with the installation screen:

Figure 3.1. MySQL Workbench Mac OS X Installation Screen



To install the applications drag the **MySQL Workbench icon** to your Applications folder.
You may then "Eject" and throw away this disk image.



Drag the MySQL Workbench icon onto the Application icon as instructed. MySQL Workbench is now installed.

You can now launch MySQL Workbench from the Applications folder.

3.3.8. Launching MySQL Workbench on Mac OS X

To launch MySQL Workbench on Mac OS X, simply open the Applications folder in the Finder, then double-click on MySQL Workbench.

It is also possible to start MySQL Workbench from the command line:

```
shell> open MySQLWorkbench.app <model file>
```

A model file must be specified. There are no command-line options with the Mac OS X version of MySQL Workbench.

3.3.9. Uninstalling MySQL Workbench on Mac OS X

To uninstall MySQL Workbench for Mac OS X, simply locate MySQL Workbench in the Applications folder, right-click, and select MOVE TO TRASH. The application is uninstalled.

3.4. Activation Procedure (Commercial Version)

Chapter 4. Getting Started Tutorial

This tutorial provides a quick hands-on introduction to using MySQL Workbench for beginners. If you have used MySQL Workbench before you can safely skip this tutorial.

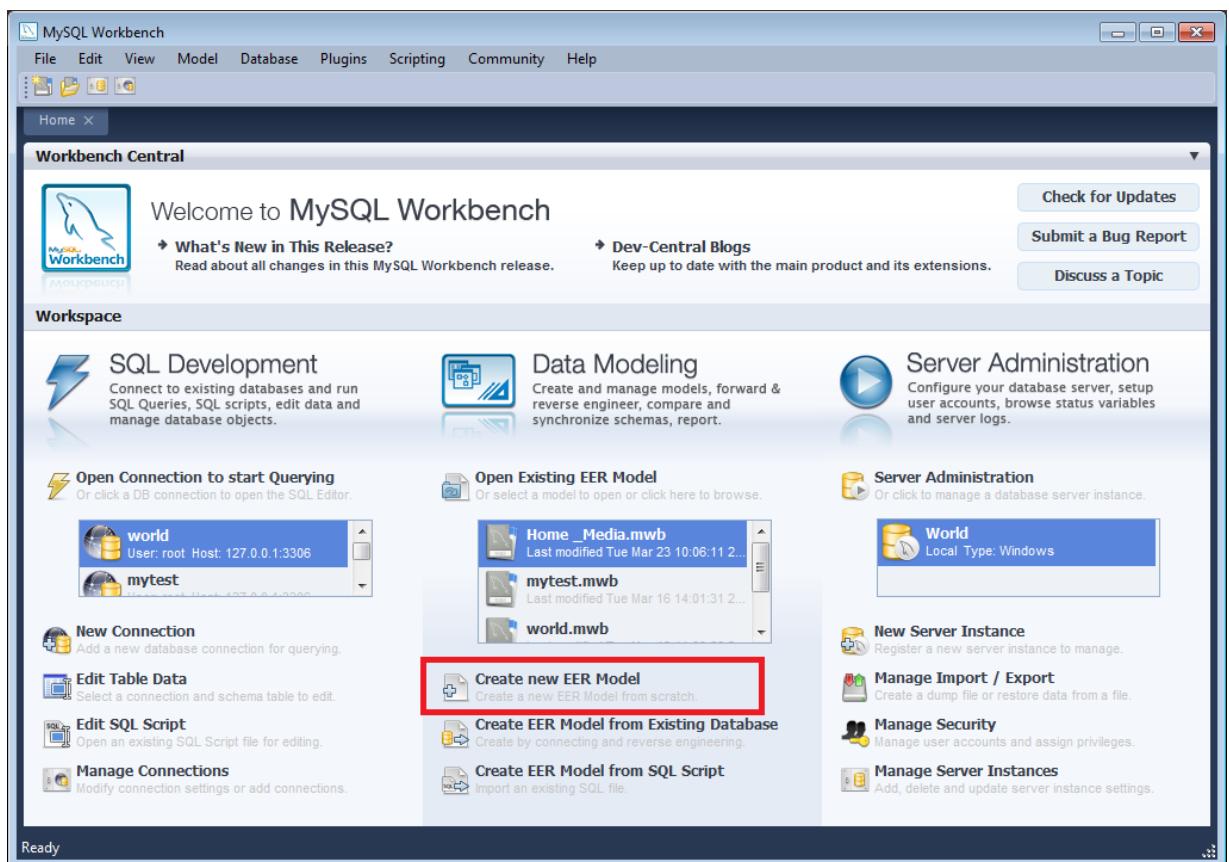
In order to complete this tutorial you will need to have a locally installed MySQL Server. If you only have access to a remote MySQL server you will need to enter appropriate connection parameters when required. This tutorial requires MySQL Workbench version 5.2.16 or above. You also need a basic understanding of MySQL concepts.

4.1. Creating a Model

In this section you will learn how to create a new database model, create a table, create an EER Diagram of your model, and then forward engineer your model to the live database server.

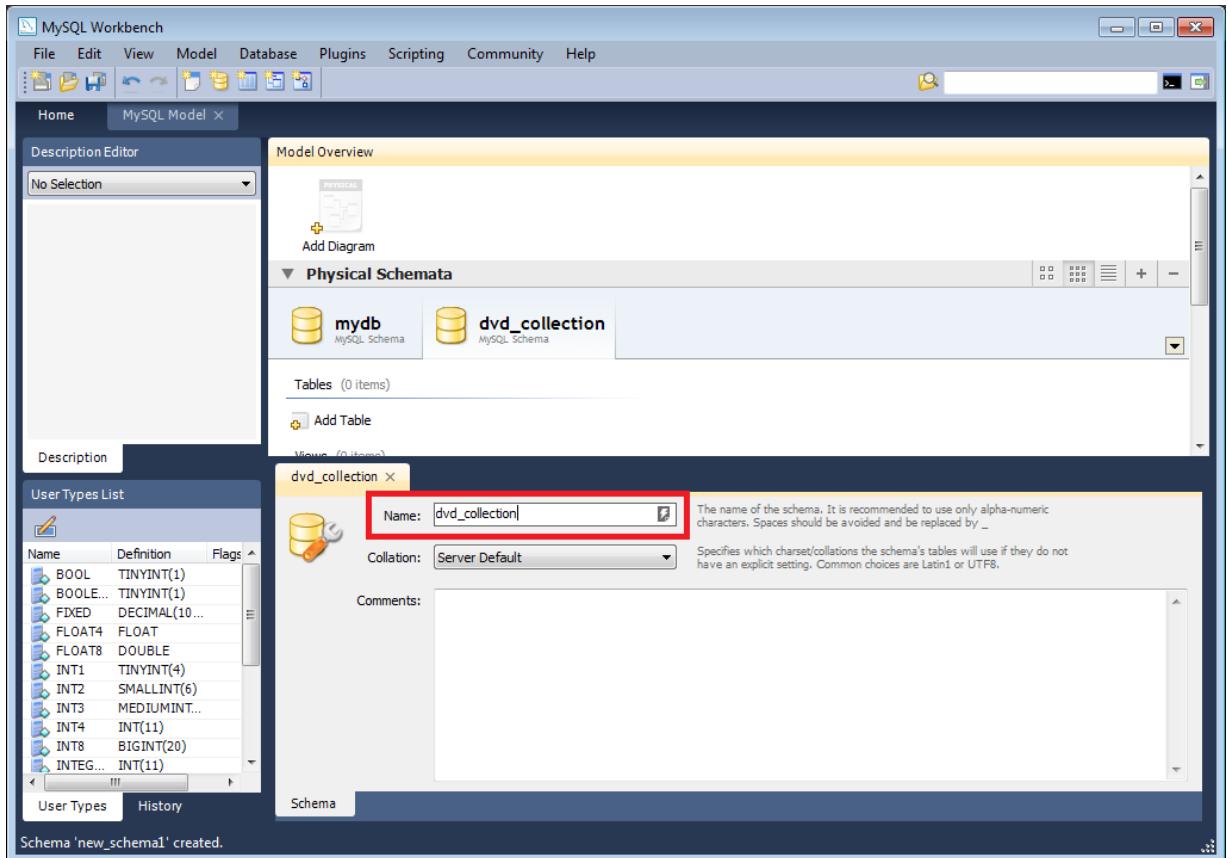
1. Start MySQL Workbench. On the Home screen select **CREATE NEW EER MODEL**. A model can contain multiple schemata. Note that when you create a new model, it contains the `mydb` schema by default. You can change the name of this schema to serve your own purposes, or simply delete it.

Figure 4.1. Getting Started Tutorial - Home Screen



2. On the Physical Schemata toolbar, click the button + to add a new schema. This will create a new schema and display a tabsheet for the schema. In the tabsheet, change the name of the schema to “dvd_collection”, by typing into the field called **NAME**. Ensure that this change is reflected on the Physical Schemata tab. Now you are ready to add a table to your schema. If at this stage you receive a message dialog asking to rename all schema occurrences, you can click YES to apply your name change.

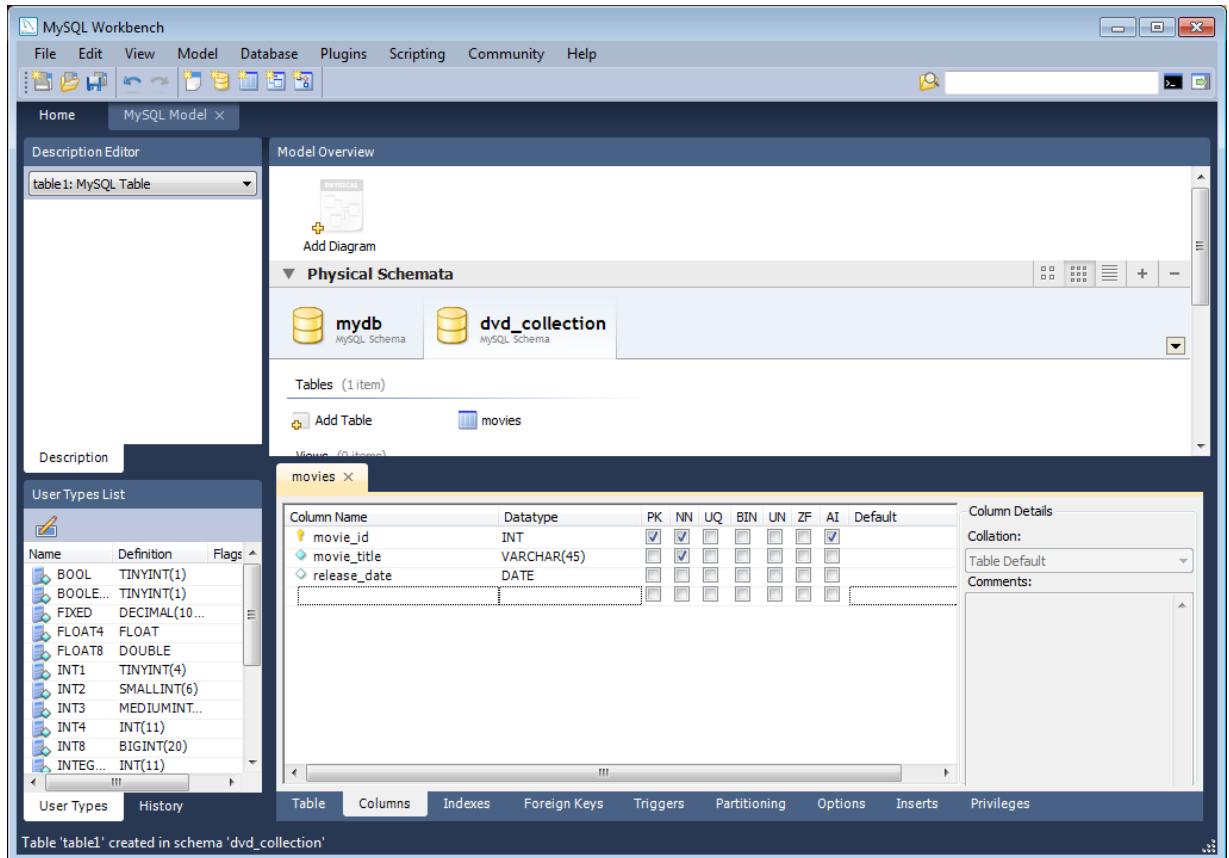
Figure 4.2. Getting Started Tutorial - New Schema



3. In the Physical Schemata section double-click **ADD TABLE**.
4. Double click **TABLE1** to launch the table editor (you may not have to do this as the table editor will automatically load at this point if you are using later versions of MySQL Workbench). In the table editor, change the name of the table to “movies” and press **Enter**. The table editor will then switch from the **TABLE** tab to the **COLUMNS** tab, to allow you to enter details of your table columns.
5. Change the name of the first column to “movie_id”. Select a datatype of **INT**. You will then make this column have the following properties: primary key, not null, autoincrement. To do this click the **PK**, **NN**, and **AI** checkboxes.
6. Add two further columns:

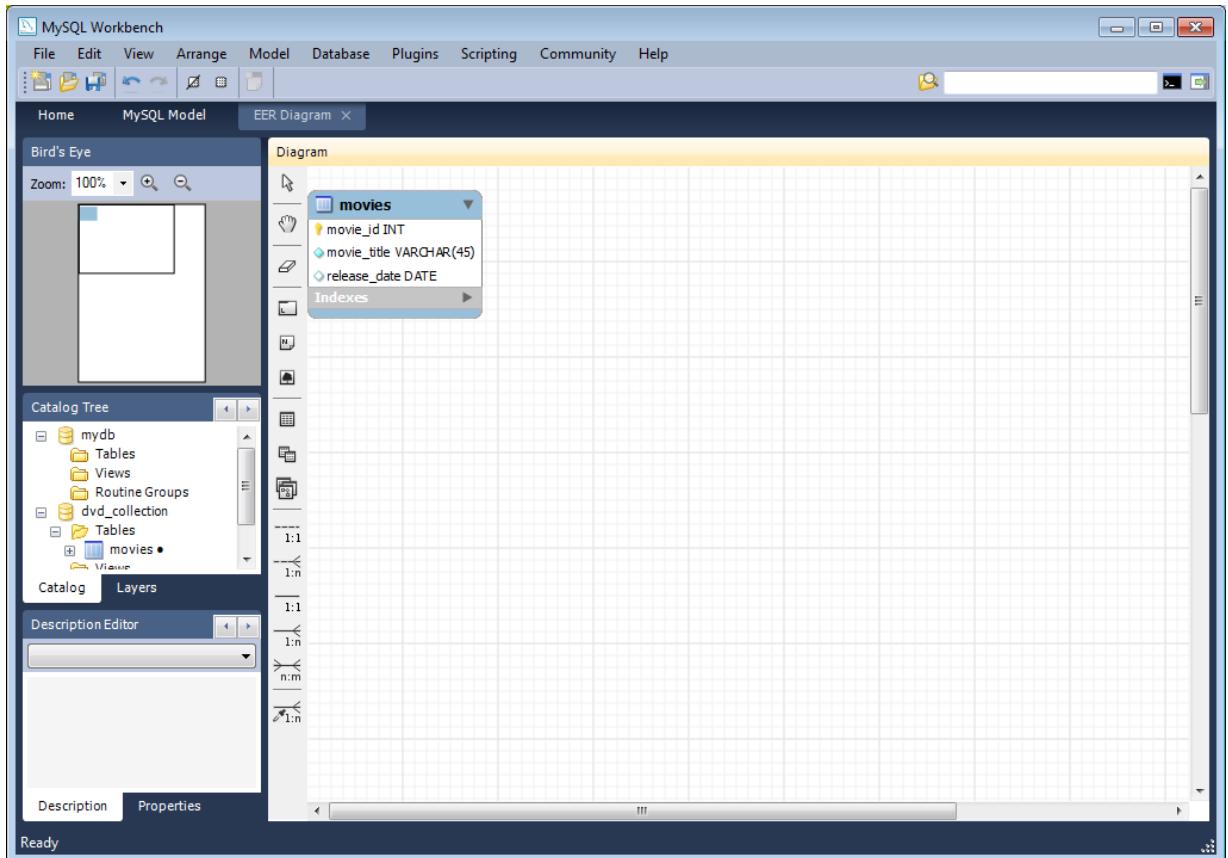
Column Name	Data Type	Column Properties
movie_title	VARCHAR(45)	NN
release_date	DATE (YYYY-MM-DD)	None.

Figure 4.3. Getting Started Tutorial - Columns



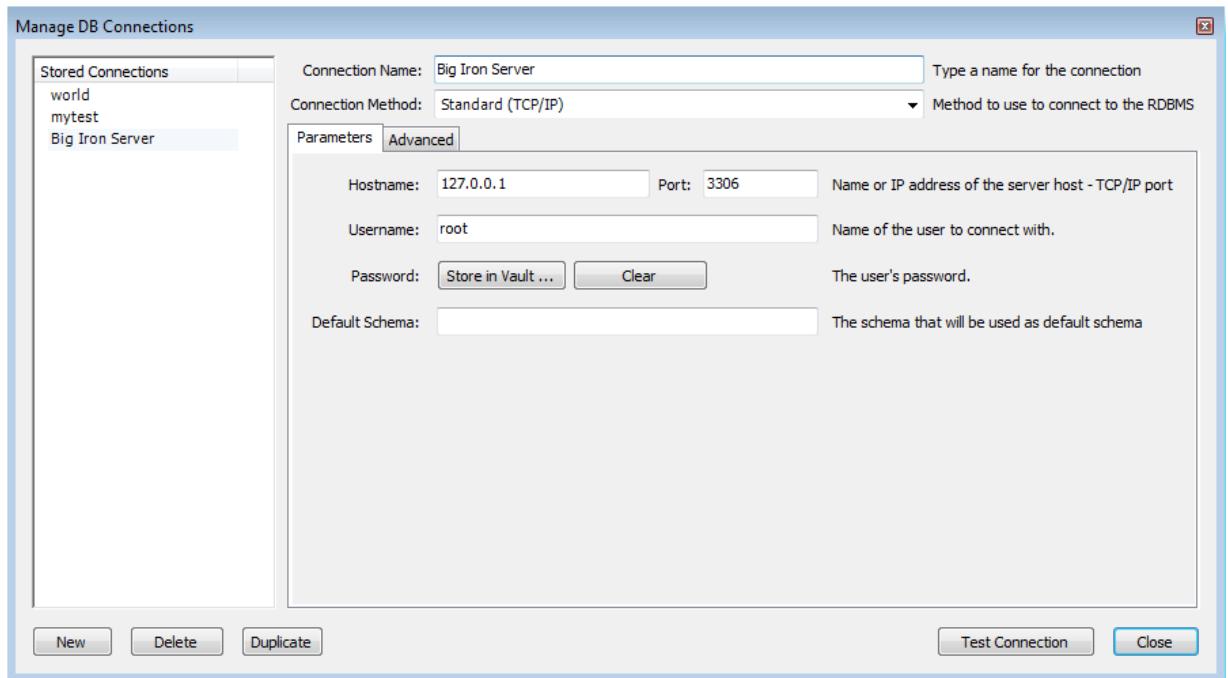
7. Now you can obtain a visual representation of this schema so far. From the main menu select MODEL, CREATE DIAGRAM FROM CATALOG OBJECTS. The EER Diagram will be created and displayed.

Figure 4.4. Getting Started Tutorial - EER Diagram



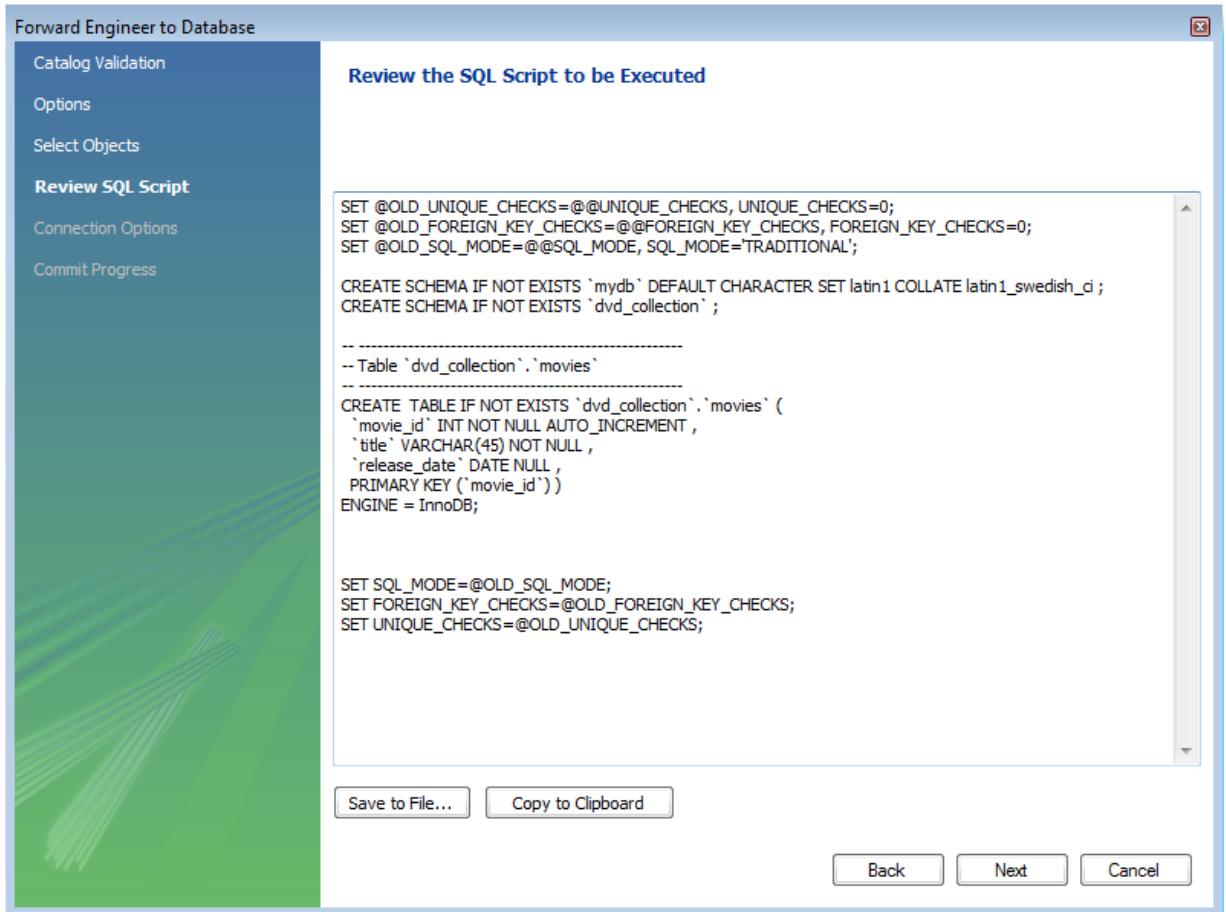
8. Now, in the table editor, change the name of the column “movie_title” to “title”. Note that the EER Diagram is automatically updated to reflect this change.
9. At this point you can save your model. Click the main toolbar button **SAVE MODEL TO CURRENT FILE**. In this case you have not yet saved this file so you will be prompted to enter a model file name. For this tutorial enter “Home_Media”. The Home_Media model may contain further schemata in addition to **dvd_collection**, such as **cd_collection**. Click **SAVE** to save the model.
10. You can synchronize your model with the live database server. First you need to tell MySQL Workbench how to connect to the live server. From the main menu select **DATABASE, MANAGE CONNECTIONS...**
11. In the **MANAGE DB CONNECTIONS** dialog click **NEW**.
12. Enter “Big Iron Server” for the connection name. This allows us to identify which server this connection corresponds to, although it is possible to create multiple connections to the same server.
13. Enter the username for the account you will use to connect to the server.
14. Click on the **STORE IN VAULT...** button and enter the password for the username you entered in the previous step. You can optionally ignore this step, and you will be prompted for this password whenever MySQL Workbench connects to the server.
15. Click **TEST CONNECTION** to test your connection parameters. If everything is OK at this point you can click **CLOSE**.

Figure 4.5. Getting Started Tutorial - Manage Connections



16. You are now ready to forward engineer your model to the live server. From the main menu select [DATABASE](#), [FORWARD ENGINEER....](#). The **FORWARD ENGINEER TO DATABASE** wizard will be displayed.
17. The first page of the wizard is the Catalog Validation page. Click the RUN VALIDATIONS button to validate the Catalog. If everything is in order the wizard will report that validation finished successfully. Click **NEXT** to continue.
18. The Options page of the wizard shows various advanced options. For this tutorial you can ignore these and simply click **NEXT**.
19. On the next page you can select the object you want to export to the live server. In this case we only have a table, so no other objects need to be selected. Click **NEXT**.
20. The next screen, Review SQL Script, displays the script that will be run on the live server to create your schema. Review the script to make sure that you understand the operations that will be carried out. Click **NEXT**.

Figure 4.6. Getting Started Tutorial - Review Script



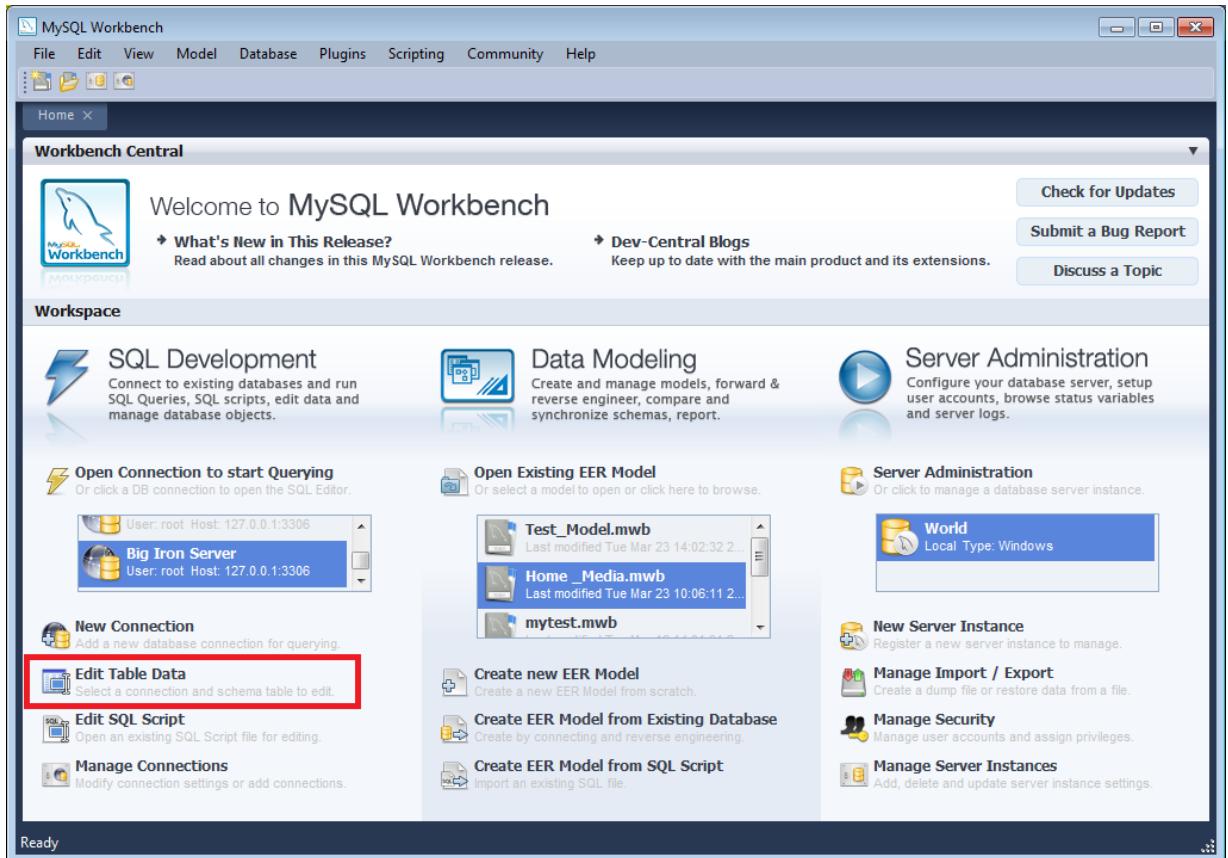
21. Select the connection you created earlier, “Big Iron Server”. Click EXECUTE. Check the messages for any errors, and then click CLOSE to exit the wizard.
22. Ensure that the script ran without error on the server and then click CLOSE. As a simple test that the script worked launch the MySQL Command Line Client. Enter `SHOW DATABASES;` and identify your schema. Enter `USE dvd_collection;`, to select your schema. Now enter `SHOW TABLES;`. Enter `SELECT * FROM movies;`, this will return the empty set as you have not yet entered any data into your database. Note that it is possible to use MySQL Workbench to carry out such checks, and you will see how to do this later, but the MySQL Command Line Client has been used here as you have probably used this previously.
23. Ensure that your model is saved. Click **SAVE MODEL TO CURRENT FILE** on the main toolbar.

4.2. Adding Data to Your Database

In the previous section you created a model, schema, and table. You also forward engineered your model to the live server. In this section you will see how you can use MySQL Workbench to add data into your database on the live server.

1. On the Home screen click the link **EDIT TABLE DATA** in the SQL Development area of the Workspace. This launches **EDIT TABLE DATA** wizard.

Figure 4.7. Getting Started Tutorial - Edit Table Data



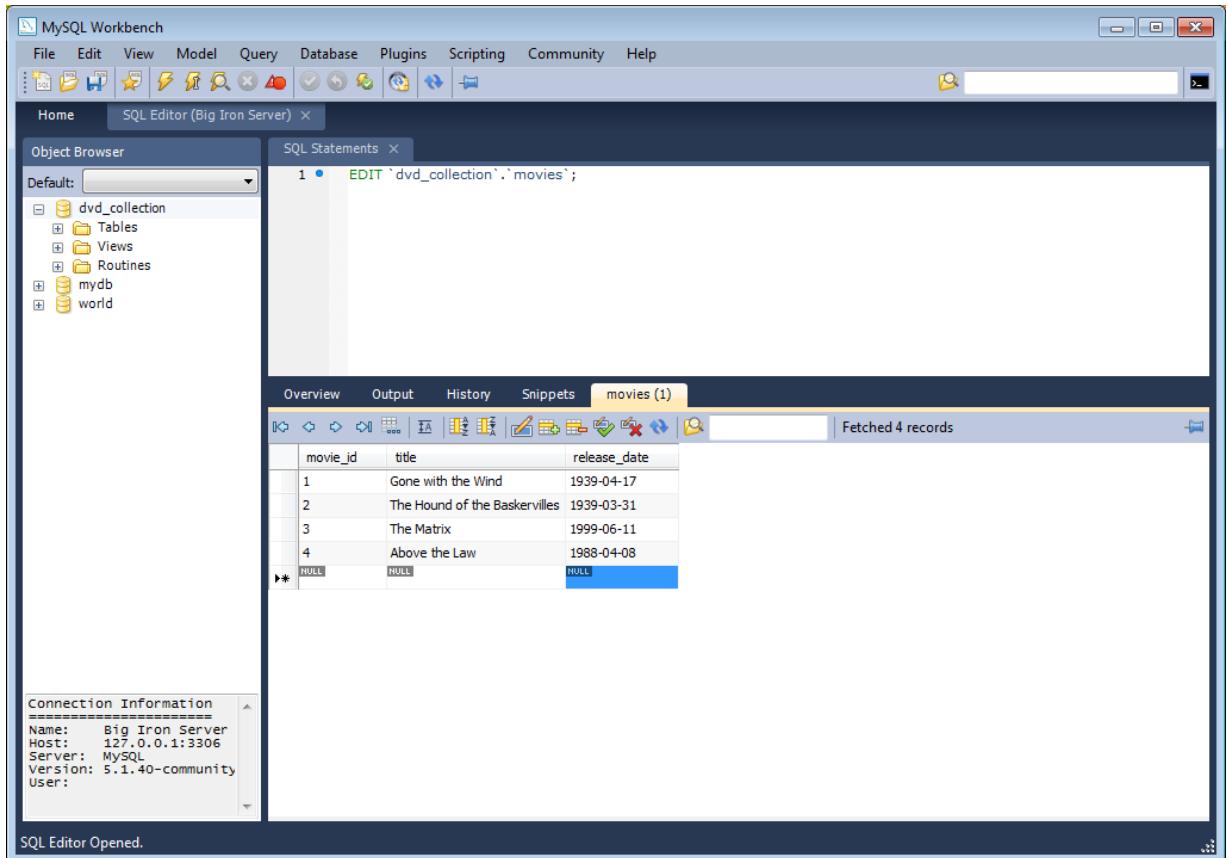
2. In the wizard select the “Big Iron Server” connection from the stored connection drop down listbox. Click NEXT.
3. Select the schema, [dvd_collection](#). Select the table to edit, [movies](#). Click FINISH.
4. You will see a data grid. This is where you can enter the data for your database. Remember that the `movie_id` was set to be autoincrement, so you do not need to enter values directly for this column. In the data grid enter the following movie information:

title	release_date
Gone with the Wind	1939-04-17
The Hound of the Baskervilles	1939-03-31
The Matrix	1999-06-11
Above the Law	1988-04-08

Note: do not modify any values in the `movie_id` column.

5. Now click the APPLY CHANGES TO DATA SOURCE button in the toolbar located in the bottom right corner. A list of SQL statements will be displayed. Confirm that you understand the operations to be carried out. Click APPLY SQL to apply these changes to the live server.
6. Confirm that the script was executed correctly and then click FINISH.
7. View the data grid again and observe that the autoincrement values have been generated.

Figure 4.8. Getting Started Tutorial - Edit Data



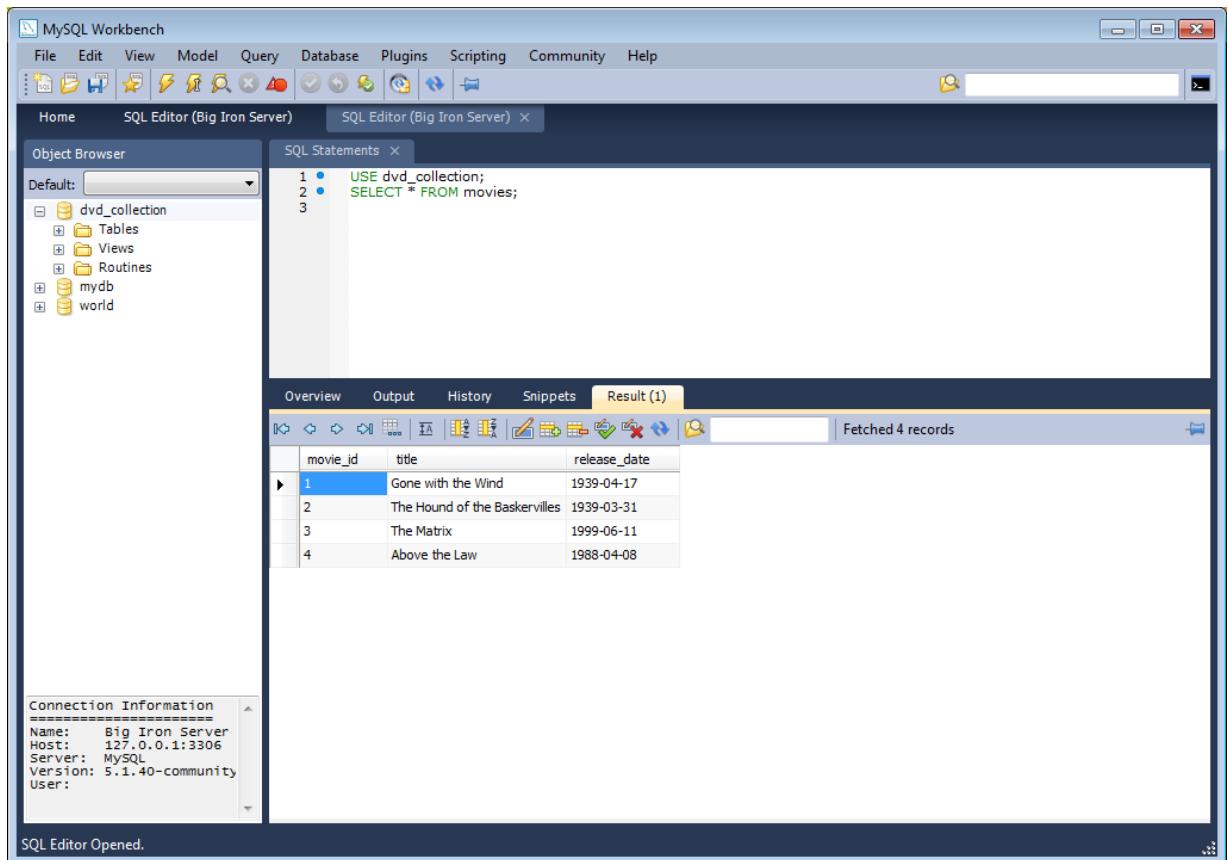
8. Now you will check that the data really has been applied to the live server. Launch the MySQL Command Line Client. Enter `SELECT * FROM movies;` to see the data just entered.
9. You can also carry out a similar check from within MySQL Workbench. Click on the Home screen tab.
10. Click the link **OPEN CONNECTION TO START QUERYING** in the SQL Development section of the Workspace. This will launch the **CONNECT TO DATABASE** dialog. Select “Big Iron Server” from the drop down listbox. Click OK.
11. A new SQL Editor tab will be displayed. In the SQL Statements area enter the following code:

```

USE dvd_collection;
SELECT * FROM movies;
  
```

12. Now click the EXECUTE SQL SCRIPT IN CONNECTED SERVER toolbar button - this resembles a small lightning bolt. The SQL Editor will display a new Result tab contain the result of executing the SQL statements.

Figure 4.9. Getting Started Tutorial - Results



In this section of the tutorial you have learnt how to add data to your database, and also how to execute SQL statements using MySQL Workbench.

Chapter 5. The Home Screen

When MySQL Workbench is first started you will be presented with the **HOME** screen. There are two main sections of the **HOME** screen:

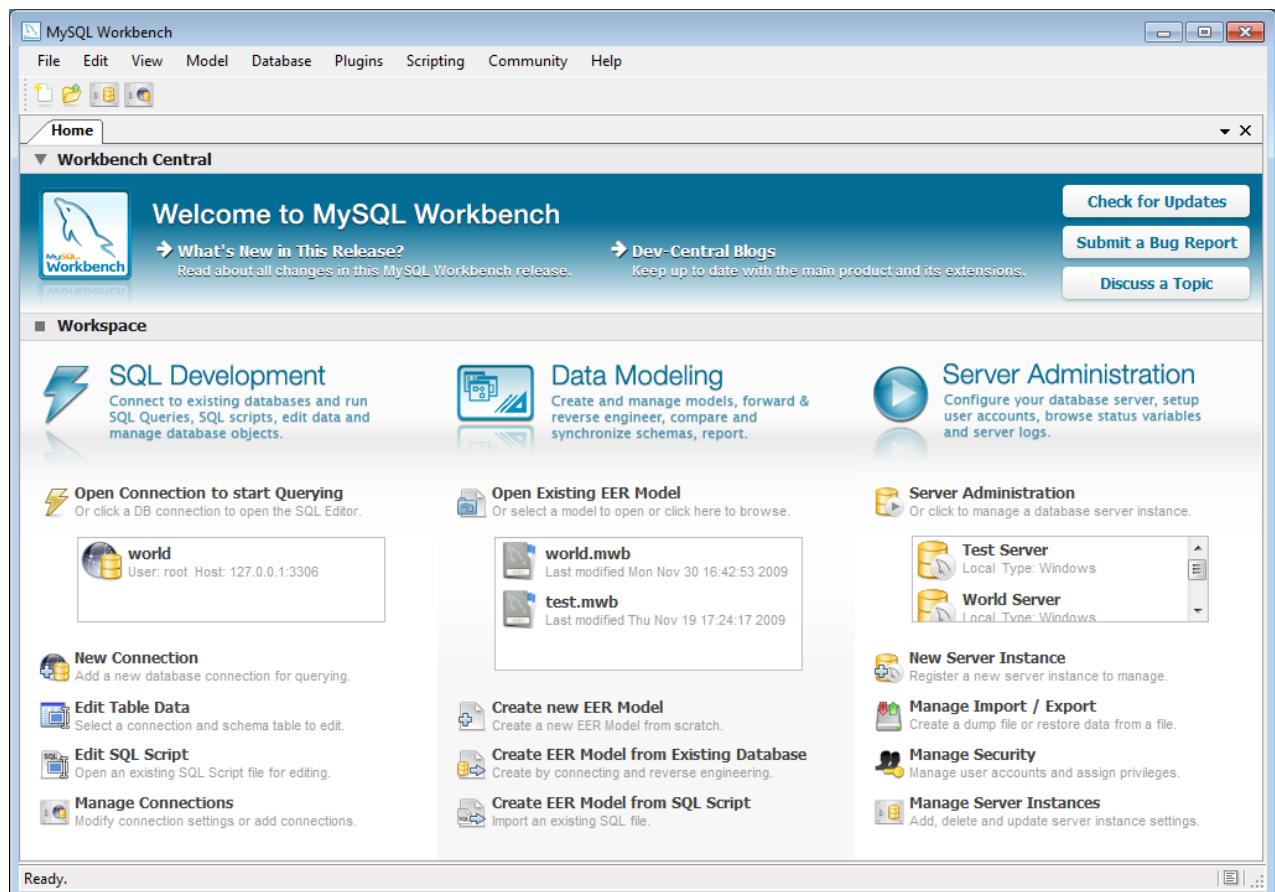
- Workbench Central
- Workspace

Note

MySQL Workbench 5.2 introduced the **HOME** screen. MySQL Workbench 5.1 simply displays the MySQL Model workspace rather than the Home screen. Note that 5.1 does not support the SQL Editor and Server Administration functionality of 5.2.

These can be seen in the following screenshot:

Figure 5.1. The Home screen



5.1. Workbench Central

Workbench Central allows you to keep up to date with MySQL Workbench news, events and resources. You can read the developer blogs, find out what's new in the release, access the forums, check for updates and file a bug report.

5.2. Workspace

The Workspace is designed to allow you to quickly get to the task you would like to carry out. For convenience it is divided into three main areas, in alignment with MySQL Workbench functionality:

1. SQL Development
2. Data Modelling
3. Server Administration

For further information on SQL Development see [Chapter 6, SQL Development](#).

For further information on Data Modeling see [Chapter 7, Data Modeling](#).

For further information on Server Administration see [Chapter 8, Server Administration](#).

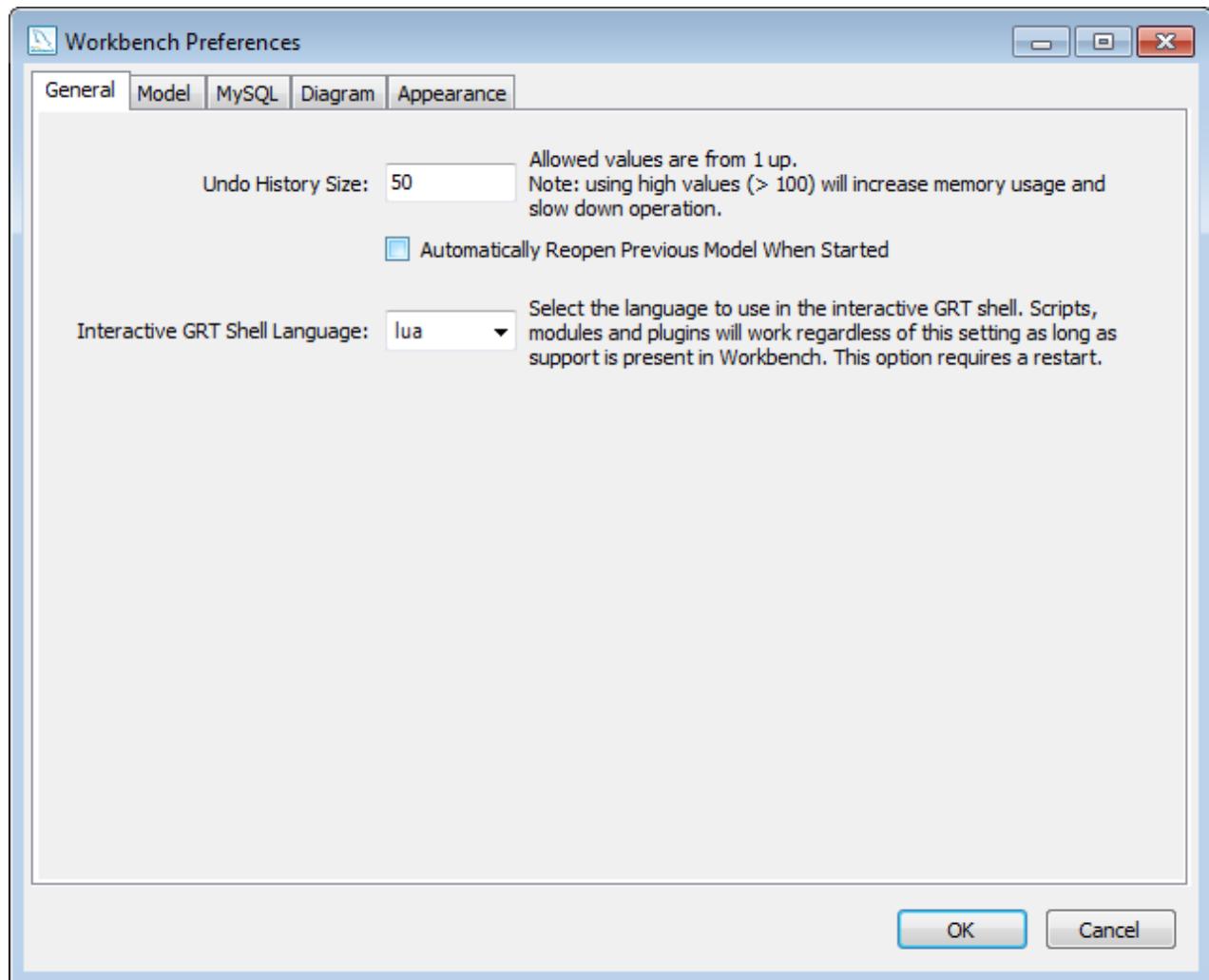
5.3. Workbench Application Minimum Window Size

From version 5.2.10, the MySQL Workbench application features a fixed minimum window size of 1024x768. You will not be able to manually reduce the size of the application to less than this resolution.

5.4. Workbench Preferences

The **PREFERENCES** menu sets MySQL Workbench defaults. Choosing the **PREFERENCES** menu item opens the following dialog box:

Figure 5.2. The *Preferences* Dialog Box



The following list describes the dialog box tabs:

- GENERAL – The delete and undo history options
- MODEL – Default object names
- MYSQL –
- DIAGRAM – EER diagram settings
- APPEARANCE – Change colours and fonts used by various Workbench components

A more detailed discussion of these options follows.

5.4.1. The General Tab

Under the GENERAL option limit the number of undos by setting a value for the **UNDO HISTORY SIZE** text box. For an unlimited number of undos set the value to **0**.

If you want the previously worked on model to automatically be reopened when you start Workbench check the **AUTOMATICALLY REOPEN PREVIOUS MODEL WHEN STARTED** checkbox.

You can select the language to be used in the GRT shell by choosing a language from the drop down listbox **INTERACTIVE GRT SHELL LANGUAGE**. Currently there is a choice between Lua and Python.

5.4.2. The Model Tab

Use the **WHEN DELETING PHYSICAL MODEL FIGURES IN DIAGRAM** frame to determine the behavior when deleting objects from the EER diagram canvas. Choose Ask and whenever you delete an object you will be asked whether you wish to remove the object from an EER diagram only or also from the catalog. The Keep Database Object in Catalog is the safest option. You also have the option of deleting the object from both the EER diagram and the catalog.

Note

If you choose the Ask option a confirmation dialog box will only open when you are deleting an object from an EER Diagram. When deleting in the MySQL Model view there is **no** confirmation dialog window and the delete action always removes the object from the catalog.

There are a variety of ways of deleting an object from an EER canvas; using the eraser tool, choosing a pop-up menu option, using the delete key, and by choosing the delete option under the EDIT menu. In each case, the action performed by the delete key is determined by the option chosen from the **WHEN DELETING PHYSICAL MODEL FIGURES IN DIAGRAM** frame.

Use the model tab to set the default value for various object names and the primary key data type. A listing of those items with their default values follows:

- Primary Key Column Name – id%table%
- Primary Key Column Type – INT
- Column Name – %table%col
- Column Type – VARCHAR(45)
- Foreign Key Name – fk%stable_%dtable%
- Foreign Key Column Name – %table%_column%
- ON UPDATE – NO ACTION
- ON DELETE – NO ACTION
- Associative Table Name – %stable%_has_%dtable%

The Primary Key Column Name is the default primary key column name when using the table editor. Likewise with the default primary key data type, and column name and column type. The remaining items are the default names used when using the relationship tools on an EER diagram.

Items enclosed by percentage signs are variables. Their meanings are as follows:

- %table% – the table associated with the object

- `%column%` – the column associated with the object
- `%stable%` – the source table
- `%dtable%` – the destination table

Legitimate values for the foreign key delete or update rules are:

- `RESTRICT`
- `CASCADE`
- `SET NULL`
- `NO ACTION` (default)

For more information about these actions see [Section 7.5.1.3.5, “The Foreign Keys Tab”](#).

5.4.3. The MySQL Tab

This allows you to set the default table storage engine, the default SQL Mode, and the utility to be used for MySQL dumps.

The document property `SqlMode` defines `SQL_MODE` for all operations affecting SQL parsing at the document scope. The purpose of this option is to preserve the consistency of SQL statements within the document.

The property has the following function:

1. Sets the `SQL_MODE` DBMS session variable to the value stored in the `SqlMode` property of the document when performing reverse engineering, forward engineering or synchronization operations.
2. Honors the `SQL_MODE` values defined in `SqlMode` so that SQL parsing is correct.

Only a subset of all possible `SQL_MODE` values affect the MySQL Workbench SQL parser. These values are: `ANSI_QUOTES`, `HIGH_NOT_PRECEDENCE`, `IGNORE_SPACE`, `NO_BACKSLASH_ESCAPES`, `PIPES_AS_CONCAT`. Other values do not affect the MySQL Workbench SQL parser and are ignored.

If the value of `SqlMode` is not set then the default value of the `SQL_MODE` session variable defined by the server stays unchanged during operations with the server. However, the MySQL Workbench SQL parser will behave as if `SQL_MODE` is also not set. This may potentially lead to inconsistencies in SQL syntax stored in the document. If you choose to not set the `SqlMode` property, ensure that the default `SQL_MODE` variable defined by the server does not contain any values from the following list: `ANSI_QUOTES`, `HIGH_NOT_PRECEDENCE`, `IGNORE_SPACE`, `NO_BACKSLASH_ESCAPES`, `PIPES_AS_CONCAT`.

The `SqlMode` property is defined in two locations: globally and at document scope. Every document upon its creation copies the value of the global property into the property defined for the document. The property value defined at document scope always has higher priority over the one defined globally.

The **PATH TO MYSQLDUMP TOOL** textfield allows you to enter the name of the MySQL Dump tool you wish to use. Leave this textfield blank to use the default Workbench dump tool.

5.4.4. The Diagram Tab

Use this tab to determine display settings for an EER diagram.

Select whether to expand new objects by checking the **EXPAND NEW OBJECTS** check box and select whether to draw line crossings by checking the **DRAW LINE CROSSINGS** check box.

From this tab you can also set the maximum number of characters for:

- Column Names
- Column Types
- Routine Names

Note that this changes the display properties only, not the objects themselves.

5.4.5. The Appearance Tab

Use this tab to set the available colors for the objects that appear on an EER diagram canvas. You can also add colors if you wish.

Changes made here affect the drop down list box of colors that appears on the toolbar when adding objects to an EER diagram canvas. For a discussion of using this list box see [Section 7.2.1, “Tool-specific Toolbar Items”](#).

You can also use this tab to set the font face, font size, and the font style for the following list of items:

- Editor
- Layer Title
- Text Figure Text
- Text Figure Title
- Connection Caption
- Routine Group Figure Item
- Routine Group Figure Title
- Table Figure Items
- Table Figure Section
- Table Figure Title
- View Figure Title

Choose from the drop down list of fonts, font sizes and styles.

Chapter 6. SQL Development

This facility in MySQL Workbench provides the functionality that was formerly available in MySQL Query Browser.

MySQL Workbench now provides extensive facilities for working directly with SQL code. Before working directly with a live server a connection must be created. Once a connection is established it is the possible to execute SQL code directly on the server and manipulate the server using SQL code.

The starting point for embarking on SQL Development work is the SQL Development area of the Home screen, which has the following action items:

1. Open Connection to start Querying
2. Open Connection to start Querying (icon)
3. New Connection
4. Edit Table Data
5. Edit SQL Script
6. Manage Connections

Open Connection to start Querying

Clicking on this action item launches the Connect to Database Wizard. From this wizard you can select a predefined connection. A new SQL Editor tab is launched where you

To read about the SQL Editor see [Section 6.2, “SQL Editor”](#).

Open Connection to start Querying (icon)

If you already have created a connection to a database it will appear in this panel as an icon. Double clicking the icon will directly launch a SQL Editor tab, and connect you to the database as defined by the connection.

New Connection

Clicking the New Connection action item launches the Manage DB Connections wizard. This wizard allows you to create a new connection. Note the wizard when launched from here does not display existing connections, it only allows you to create a new connection.

To read more about creating and managing connections see [Section 6.1, “Manage Connections”](#).

Edit Table Data

This action item allows you to edit table data. When clicked the Edit Table Data wizard is launched. This is a two stage wizard. The first stage allows you to select a Stored Connection. The second stage allows you to select the Schema and Table you want to edit. Once the wizard is completed a SQL Editor tab is launched which displays a data grid that allows you to interactively edit table data as required.

Edit SQL Script

Clicking this action item launched the Edit SQL Script wizard. This is a two stage wizard. The first stage allows you to select a Stored Connection. The second stage allows you to select a SQL Script file, and optionally have the script executed after it is opened. Once the wizard is completed a SQL Editor tab will be launched, with the script displayed. If you optionally selected to run the script, the script will run and the results will be displayed.

Manage Connections

Clicking this action item launches the Manage DB Connections wizard. This wizard also displays Stored Connections, which can be selected to change as required. New connections can also be created from this wizard.

To read more about managing connections see [Section 6.1, “Manage Connections”](#).

The SQL Development activities can therefore be grouped into two broad categories:

1. Creating and managing connections. This functionality is primarily managed through the Manage DB Connections dialog.

2. Editing, running, and manipulating databases through SQL code. This functionality is associated with what is termed the *SQL Editor*.

Each of these is described in more detail in the following sections.

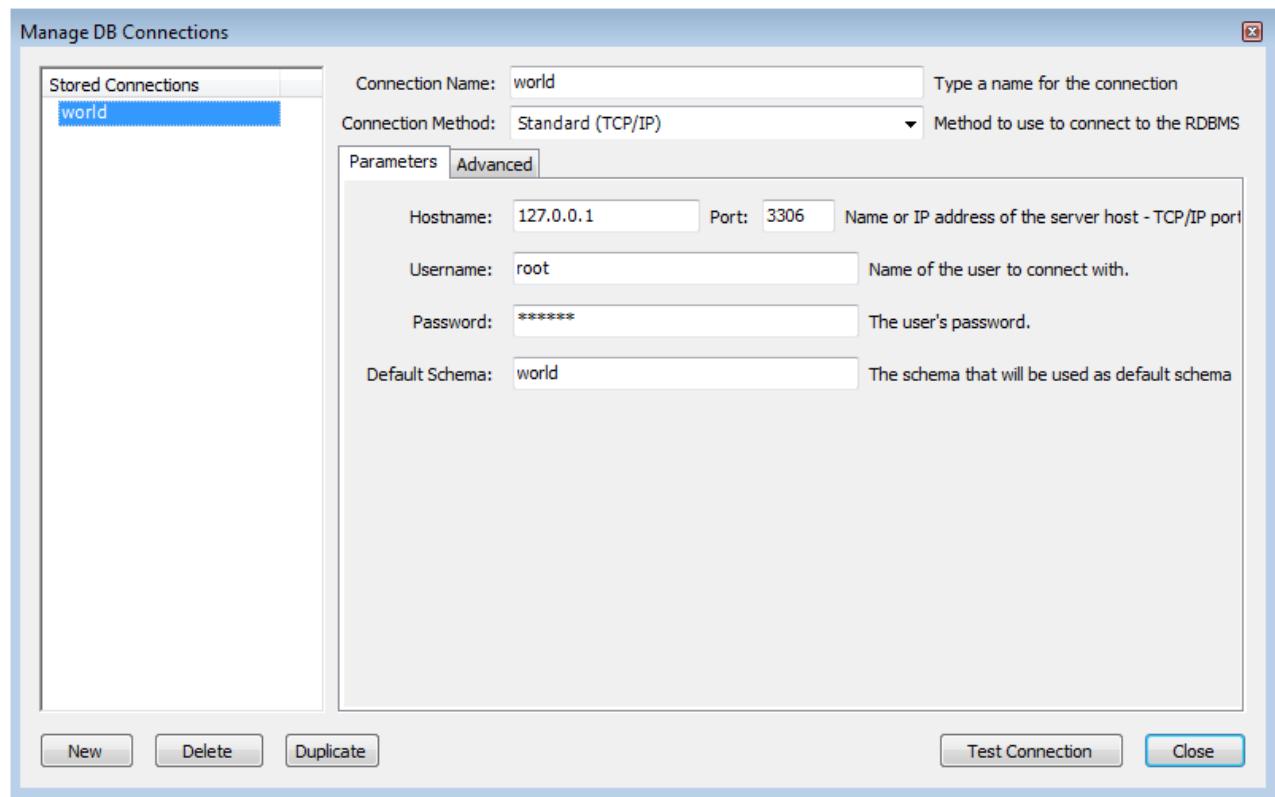
6.1. Manage Connections

MySQL Workbench provides a tool, the Manage DB Connections dialog, for creating and managing connections to servers. The connections created can then be used from the wizards that need to connect to a server, for example the wizard used to reverse engineer a live database. However, it is still possible to set connection parameters from these wizards if required, without invoking the Manage DB Connections dialog directly.

The Manage DB Connections dialog is invoked by selecting **DATABASE**, **MANAGE CONNECTIONS** from the main menu. It can also be invoked from any of the wizards requiring access to a live database. This is achieved by using the **MANAGE STORED CONNECTIONS** item, found in the wizard's **STORED CONNECTION** drop down list box.

Once the Manage DB Connections dialog is launched, you are presented with a dialog that allows you to create or delete connections:

Figure 6.1. Manage DB Connections - Dialog



Click **NEW** to create a new connection. Once created the connection can be selected from the **STORED CONNECTIONS** list. You can then set various parameters for the connection, including the following:

- **CONNECTION NAME:** The name to use to refer to this connection. This connection can then be selected from a dropdown list box in other wizards requiring a connection.
- **CONNECTION METHOD:** The methods available are Standard TCP/IP, Local Socket/Pipe, and Standard TCP/IP over SSH.

Once you have selected a connection method, the textfields available in the **PARAMETERS** tab and the **ADVANCED** tab of the dialog will change accordingly. More details of these options and parameters are available in the following sections.

Once all parameters have been set as required you can click the **TEST CONNECTION** button to test the connection to the live server. Once you are satisfied that the connection works as expected you can then close the wizard by clicking the **CLOSE** button. You can

then use the stored connection from any of the wizards requiring connection to a live server.

You can also duplicate an existing connection using the DUPLICATE button.

6.1.1. Standard TCP/IP Connection

This section discusses the settings on the **PARAMETERS** and **ADVANCED** tabs for the Standard TCP/IP connection type.

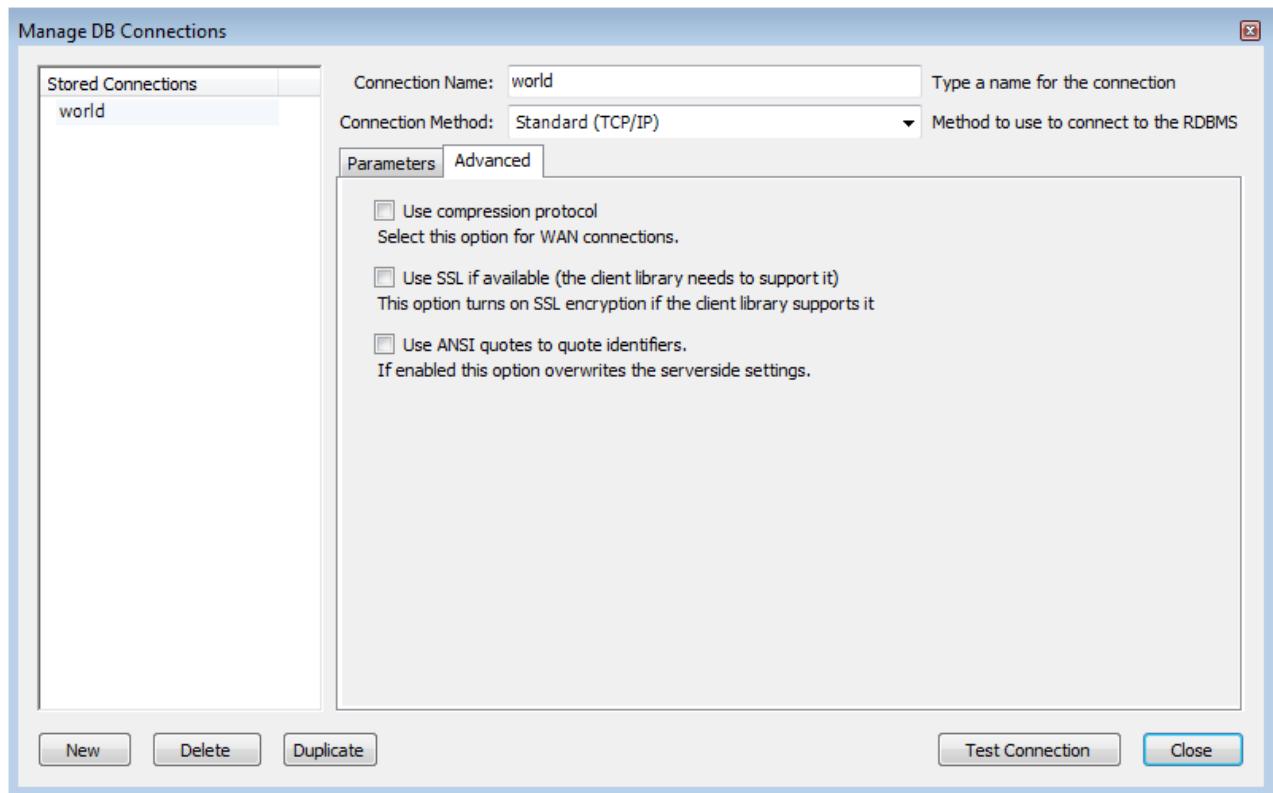
Parameters tab

- **HOSTNAME:** The host name or IP address of the MySQL server.
- **USERNAME:** User account to use for the connection.
- **PASSWORD:** Optional password for the account used. If you do not enter a password here you will be prompted to enter the password for the account to be used when MySQL Workbench attempts to establish the connection.
- **PORT:** The TCP/IP port on which the MySQL server is listening (the default is 3306).
- **DEFAULT SCHEMA:** When the connection to the server is established this is the schema that will be connected to by default. This becomes the default schema for use in other parts of MySQL Workbench.

Advanced tab

There are also more parameters that can be set for the connection via the **ADVANCED** tab:

Figure 6.2. Manage DB Connections - Advanced Tab



The advanced options include checkboxes for:

- **USE COMPRESSION PROTOCOL:** If checked, the communication between the application and the MySQL server will be compressed, which may increase transfer rates. This corresponds to starting a MySQL command-line tool with the `--compress` option.
- **USE SSL IF AVAILABLE:** This option turns on SSL encryption. The client library needs to support this option. Note: this feature is currently not supported.

- **USE ANSI QUOTES TO QUOTE IDENTIFIERS:** Treat “” as an identifier quote character (like the “” quote character) and not as a string quote character. You can still use “” to quote identifiers with this mode enabled. With this option enabled, you cannot use double quotes to quote literal strings, because it is interpreted as an identifier. Note: if this option is selected, it overrides the server setting.

6.1.2. Local Socket/Pipe Connection

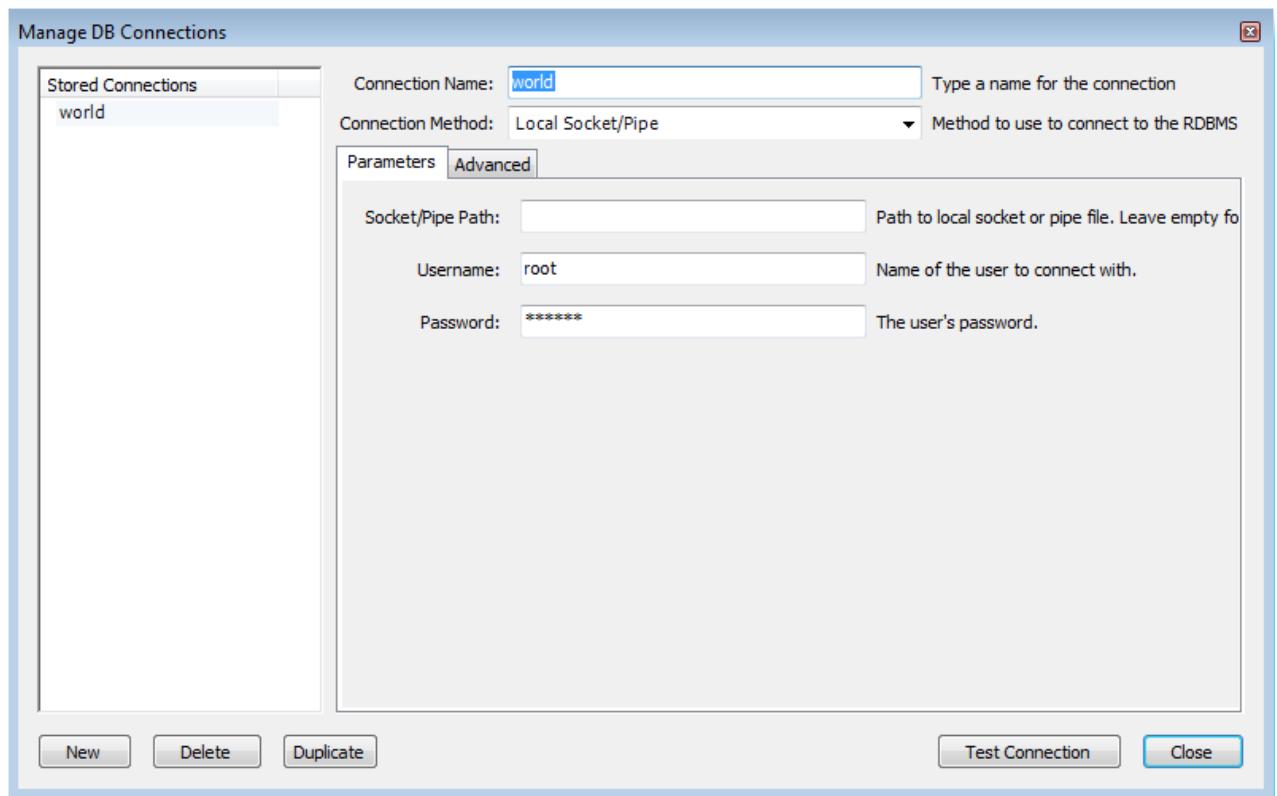
This connection type allows MySQL Workbench to connect to MySQL Server using a socket or pipe.

Parameters

The unique textfield here is **SOCKET/PIPE PATH**. The name of the socket or pipe is entered here. If the textfield is left blank the default socket or pipe name is used. The default pipe name on Microsoft Windows is [MySQL](#). On UNIX the default socket name is [/tmp/mysql.sock](#).

This option can be seen in the following screenshot:

Figure 6.3. Manage DB Connections - Socket/Pipe Parameters



Advanced

The only option available in this tab is **USE ANSI QUOTES TO QUOTE IDENTIFIERS**. This option was discussed in [Section 6.1.1, "Standard TCP/IP Connection"](#).

6.1.3. Standard TCP/IP over SSH Connection

The connection type allows MySQL Workbench to connect to MySQL Server using TCP/IP over an SSH tunnel.

Parameters

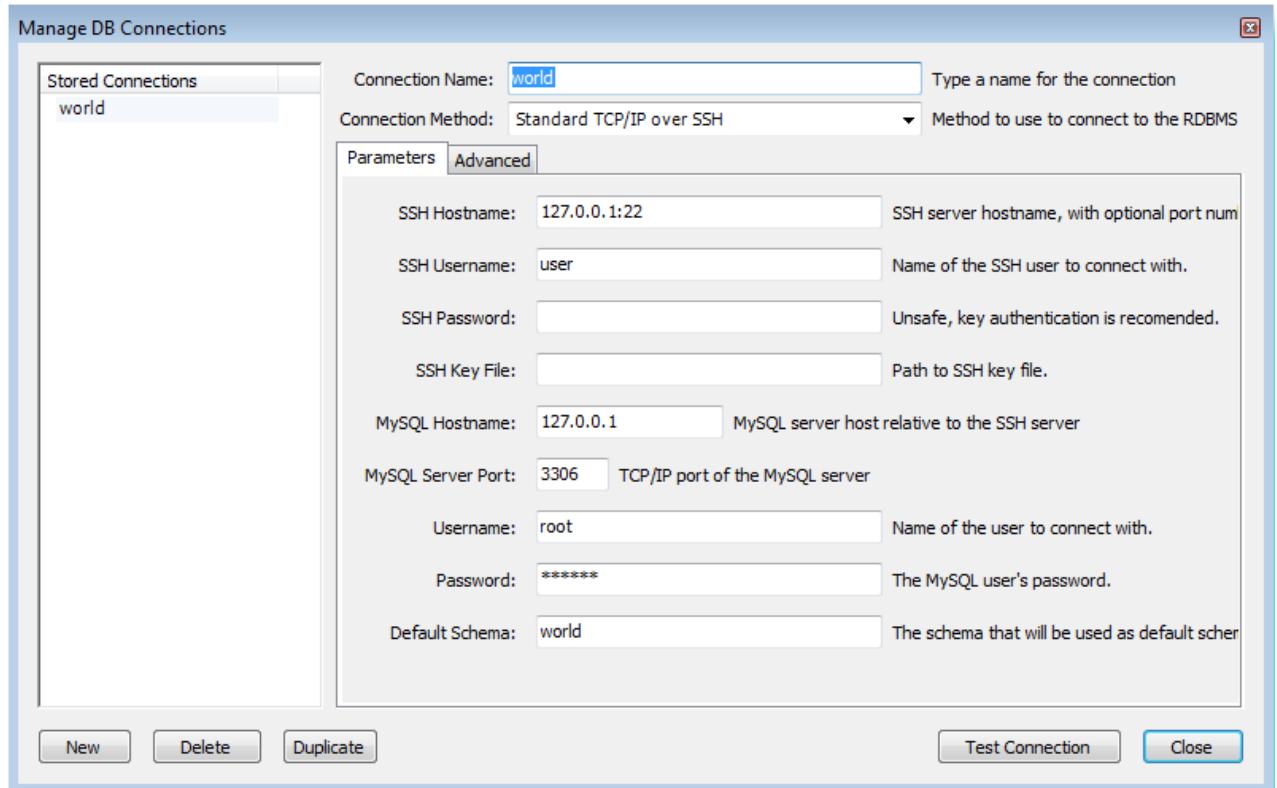
In addition to a number of parameters that are in common with Standard TCP/IP connections, this connection method features a number of specialized parameters. These are listed here:

- **SSH Hostname** - this is the name of the SSH server. An optional port number can also be provided.

- **SSH Username** - this is the name of the SSH username to connect with.
- **SSH Password** - the SSH password. It is recommended that an SSH key file is also used.
- **SSH Key File** - a path to the SSH key file. Note, only key files in OpenSSH format are currently supported.

These are shown in the following screenshot:

Figure 6.4. Manage DB Connections - SSH Parameters



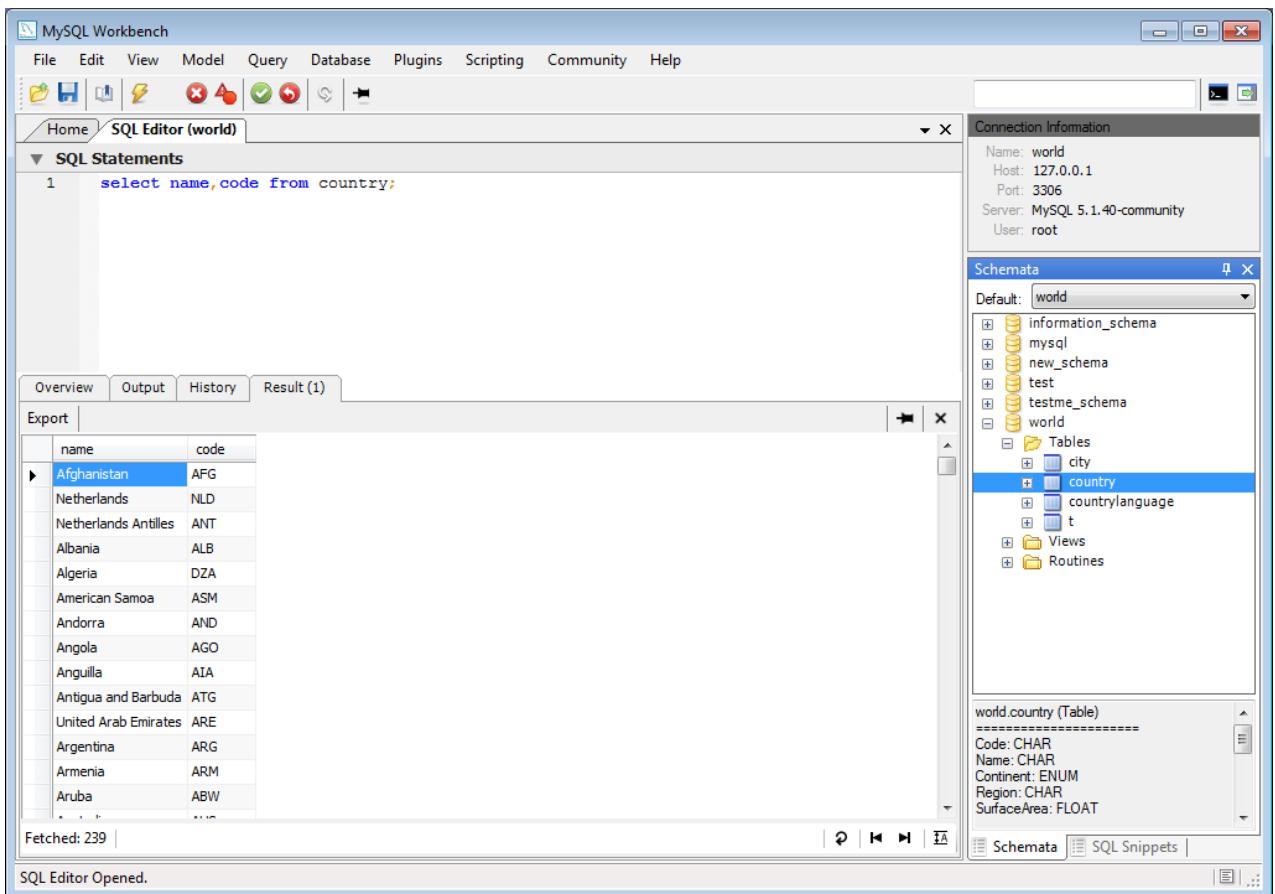
Advanced

The options here are the same as for the Standard TCP/IP connection, details of which can be found in [Section 6.1.1, “Standard TCP/IP Connection”](#).

6.2. SQL Editor

MySQL Workbench 5.2 introduced the SQL Editor facility. The SQL Editor can be launched via various action items on the Home screen. It can also be launched by selecting **DATABASE**, **QUERY DATABASE** from the main menu, or by using the keyboard shortcut **Ctrl+U** on Windows, or **Cmd+U** on Mac OS X. At this point you will be asked to select either a stored connection or enter the details for a new connection. Once a connection has been made to the server a new tab called **SQL EDITOR (SCHEMA)** is displayed:

Figure 6.5. SQL Editor



The main elements of the SQL Editor user interface are:

- The Main Menu
- Toolbar
- Statements Panel
- Main Tabsheets
- Sidebar

Each of these are described in more detail in the following sections.

6.2.1. Main Menu

When a SQL Editor tab is selected the most important item on the main menu bar is QUERY. The Query item features the following menu items:

- EXECUTE - causes the script in the SQL Statements area to be executed.
- STOP - stop execution of the currently running script.
- RECONNECT - reconnect to the server.
- NEW TAB - Creates a duplicate of the current SQL Editor tab.
- COMMIT TRANSACTION - commits a database transaction.
- ROLLBACK TRANSACTION - rolls back a database transaction.
- REFRESH - synchronizes with the live server and refreshes views such as the live Overview tabsheet.

- COMMIT RESULT EDITS - commits any changes you have made to the server.
- DISCARD RESULT EDITS - discards any changes you have made.
- EXPORT RESULTS - This allows you to export result sets to a file. Selecting this option displays the **EXPORT QUERY RESULTS TO FILE** dialog. The dialog allows you to select which result set you wish to export, the file format (CSV, HTML, XML) and the name and location of the output file. Then click EXPORT to export the data.

6.2.2. Toolbar

The toolbar features ten buttons, as shown in the following screenshot:

Figure 6.6. SQL Editor - Toolbar



From left to right these are:

- **Open a SQL Script File** - Clicking this button allows any saved SQL script to be loaded ready for execution. The script will be displayed in the **SQL STATEMENTS** area.
- **Save SQL Script to File** - Clicking this button allows the currently loaded SQL script to be saved to a file specified by the user.
- **Save SQL to Snippets List** - SQL code snippets saved here can be given a name. They will appear in the **SQL SNIPPETS** palette in the SQL Editor sidebar.
- **Execute SQL Script in Connected Server** - Executes the currently loaded SQL script. Results are displayed in one or more Results tabs.
- **Explain selected SQL** - generates a results set tab containing the result of running an **EXPLAIN** on the selected SQL.
- **Stop the query being executed** - halts execution of the currently executing SQL script. This restarts the connection to the database server.
- **Toggle whether execution of SQL script should continue after failed statements** - If the red 'breakpoint' circle is displayed the script will finish on a statement that fails. If the button is depressed so that the green arrow is displayed, then execution will continue past the failed code, possibly generating additional result sets. Any error generated from attempting to execute the faulty statement will be recorded in either case in the Output tabsheet.
- **Commit** - commits a transaction.
- **Rollback** - rolls back a transaction.
- **Toggle Auto-Commit Mode** - if selected, transactions will automatically be committed.
- **Reconnect to DBMS** - re-establishes the database connection.
- **Refresh state of database structures** - refreshes the view of Schemata, Tables, Views and Routines that appears in the Live Overview Schema tabsheet. For example, if a SQL script creates a new table, it will not appear in the Overview tab until the refresh toolbar button is pressed.
- **Toggle whether query result tabs should be kept between queries by default** - normally when a script is executed any results generated from previous executions of the script are lost, and the new results displayed in the results tab. If this toggle button is pressed, so that the pin appears inserted, results will be retained between executions. Each execution of the script will create a new Results tab containing the result set.

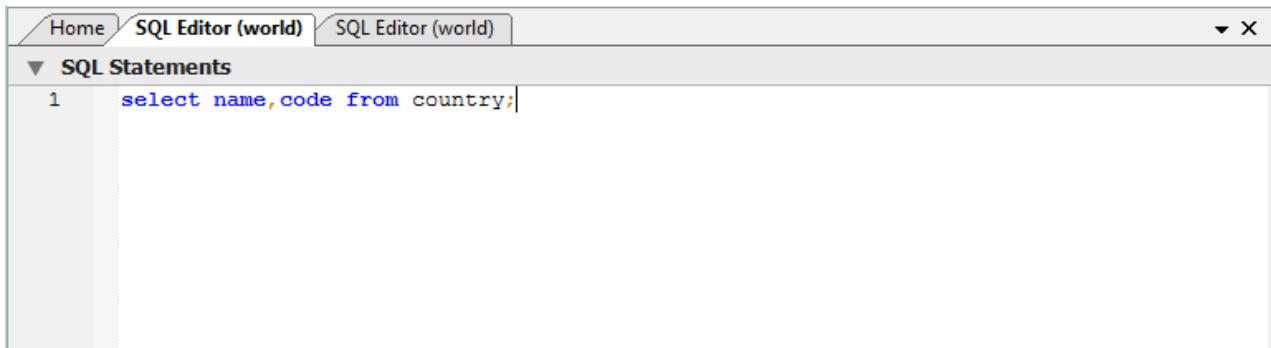
6.2.3. SQL Statements Panel

In this area you can enter SQL Statements directly.

The statements entered can be saved to a file for later use. At any point you can also execute the statements you have entered.

To save a snippet of code entered into the SQL Statements panel, click the SAVE SQL TO SNIPPETS LIST, enter a name, and click OK. The snippet can be inserted into the SQL Statements panel at any time by double-clicking on the named snippet in the SQL Snippets Palette.

Figure 6.7. SQL Editor - Statements Panel



Note

There is a quick way to enter the names of tables, views and columns. Simply double-click the item in the Schemata Palette and the name of the item clicked will be inserted into the SQL Statements panel.

6.2.4. Main Tabsheets

The main tabsheets area contains several tabs:

- Live Schema Overview Tabsheet
- Output Tabsheet
- History Tabsheet
- Results Tabsheets
- Live Editing Tabsheet

Figure 6.8. SQL Editor - Main Tabsheets



Each of these is described in more detail in the following sections.

6.2.4.1. Live Schema Overview Tabsheet

This tabsheet provides an overview of the schema. The schema objects Tables, Views and Routines are displayed for the current schema.

6.2.4.2. Output Tabsheet

The Output tabsheet displays a summary of the communication between the script and the server. The messages displayed can be information or errors. Each message displays the time, the action that was carried out, and the response from the server. This is useful for troubleshooting scripts.

6.2.4.3. History Tabsheet

The History tabsheet provides a history of SQL operations carried out. Both the time of the SQL operation and the SQL code itself is recorded. To view the SQL executed click on the time, and the SQL code executed will be displayed in the SQL column.

6.2.4.4. Results Tabsheets

The results area of the screen shows the results from any queries that have been executed. If the script contains multiple queries then multiple result tabs will be generated, one for each query that returns results.

Figure 6.9. SQL Editor - Results Tabsheets



A screenshot of the SQL Editor interface showing the 'Results' tabsheet. The tabs at the top are 'Overview', 'Output', 'History', and 'Result (1)', with 'Result (1)' being the active tab. Below the tabs is an 'Export' button. The main area displays a table with two columns: 'name' and 'code'. The first row, 'Afghanistan' with code 'AFG', is selected and highlighted in blue. The table contains 239 rows of data. At the bottom left, it says 'Fetched: 239'. On the right side, there are standard window controls (minimize, maximize, close) and navigation icons.

	name	code
▶	Afghanistan	AFG
	Netherlands	NLD
	Netherlands Antilles	ANT
	Albania	ALB
	Algeria	DZA
	American Samoa	ASM
	Andorra	AND
	Angola	AGO
	Anguilla	AIA
	Antigua and Barbuda	ATG
	United Arab Emirates	ARE
	Argentina	ARG
	Armenia	ARM
	Aruba	ABW
...

Fetched: 239 |

Controls are provided to allow you to easily move over the results. These are shown in the following screenshot:

Figure 6.10. SQL Editor - Results Tabsheets Navigation Controls



There are four controls available. The controls from left to right are:

1. **Refresh Data from Data Source** - Refreshes the current result set from the data source.
2. **Go to first record** - Highlights the first record in the current result set.
3. **Go to last record** - Highlights the last record in the current result set.
4. **Toggle wrapping of cell contents** - The data in the cell can either be truncated or wrapped. This button allows you to toggle between these options.

6.2.4.5. Live Editing Tabsheets

It is possible to edit data in real time using the Live Editing tabsheets. In the Overview tab, if a table is double-clicked, a live editing tab will be launched, allowing you to edit the data maintained in that table. Field data can be edited by clicking on a field and entering the required data, or editing existing data. In addition to the controls offered by the Results tabsheet, the Live Editor tab features two additional controls:

1. **Apply Changes to Data Source** - Applies any changes that may have been made to the data fields to the live server.
2. **Discard any Changes** - Discards any changes that may have been made to the data fields.

These additional controls allow you to apply or discard any changes you may have made to the field data. If changes are applied, the data will then be synchronized with the live server. If changes are discarded the live server will not be affected.

Note

It is possible to enter a function, or other expression, into a field. If doing so, the prefix `\func` should be used, to prevent MySQL Workbench from escaping quotes. For example, if entering the expression `md5('fred')` MySQL Workbench would generate the code `md5('fred\')`. To prevent this enter the expression as `\func md5('fred')`. This will ensure that the quoting is not escaped.

See also [Section 7.5.1.3.9, “The Inserts Tab”](#).

6.2.5. Sidebar

The Sidebar contains several panels. These are:

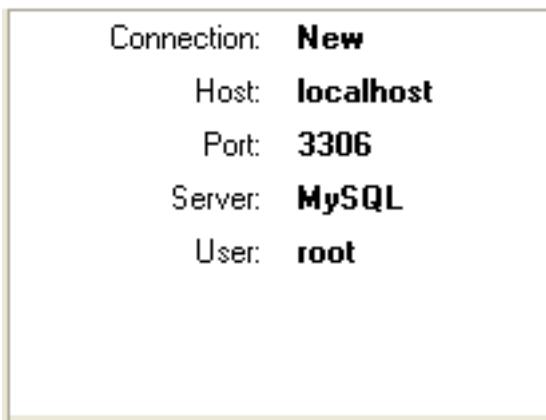
- Connection Information Palette
- Schemata Palette
- SQL Snippets Palette

Each of these is described in more detail in the following sections.

6.2.5.1. Connection Information Palette

This section provides a summary of the current connection to the server.

Figure 6.11. SQL Editor - Connection Information Palette



6.2.5.2. Schemata Palette

The Schemata Palette contains a drop down listbox and a schemata explorer control.

Default Schema Listbox

The drop down listbox lists the schema that are available on the currently connected server. It is possible to select a schema to become the currently active schema using this facility.

This selector executes a `USE DB` statement. Once set, subsequent statements without schema qualifiers will be executed against this default schema. Note that this will only be set for the query session. If you wish to set a default schema for multiple MySQL Workbench sessions, you will need to set the default schema for the stored connection. To do this from the Home screen click **MANAGE CONNECTIONS**, then in the **MANAGE DB CONNECTION** dialog set the desired default schema on the **PARAMETERS** tab.

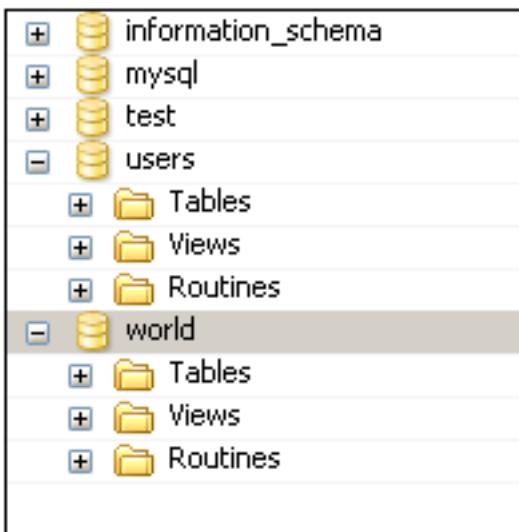
Figure 6.12. SQL Editor - Default Schema Listbox



Schemata Explorer

This area allows you to explore the schemata available on the currently connected server.

Figure 6.13. SQL Editor - Schemata Explorer



A useful feature that was introduced in MySQL Workbench 5.2.9 is the ability to rapidly enter the names of columns, tables and views into the SQL Statement area. Double-clicking views, tables, and column names in the schemata explorer will insert the corresponding name into the SQL Statements area. This reduces typing significantly when entering SQL statements containing several tables, columns or views.

The Schemata Palette also features a context menu which can be displayed by right-clicking on, for example, a table. Right-clicking on a table would display the following menu items:

- VIEW DATA - pulls table data from the live server into a Results tabsheet.
- EDIT DATA - pulls table data from the live server into a named tabsheet, and allows editing. Data can be saved directly to the live server.
- COPY NAME TO CLIPBOARD - Will copy the qualified name of the table to the clipboard.
- ALTER TABLE - displays the table editor loaded with the details of the table clicked on.
- DROP TABLE - drops a table. All data will be lost if this operation is carried out.
- REFRESH ALL - refreshes all schemata in the explorer by resynching with the server.

Right-clicking on an empty area inside the schemata palette displays the following menu options:

- CREATE SCHEMA - this allows you to create a new schema on the connected server. You can apply your changes to synchronize with the live server by clicking the APPLY button.
- REFRESH ALL - Simply synchronizes with the live server so that information with the schemata explorer is updated.

6.2.5.3. Snippets Palette

When you enter SQL code into the SQL Statements area, you can use the Save SQL to Snippets List toolbar button to save your SQL code. At the time of saving you will be asked to specify a name. These named snippets can be viewed from the SQL Snippets Palette. You can load any SQL snippet into the SQL Statements area by double-clicking the desired snippet in the Snippets Palette.

Figure 6.14. SQL Editor - Snippets Palette

SQL Snippets	
#	SQL
A	TestSnippet select Name, Population from Country ;
B	Tony Snippet select Name, Code from Country;
C	AnotherSnippet select * from Country;
D	A Snippet For All Seasons select Name, Population from Country where

Chapter 7. Data Modeling

MySQL Workbench provides extensive capabilities for creating and manipulating database models. Some of these capabilities are listed here:

- Create and manipulate a model graphically.
- Reverse engineer a live database to a model.
- Forward engineer a model to a script or live database.
- Create and edit tables and insert data.

This is not an exhaustive list. These, and additional data modeling capabilities, are discussed in the following sections.

The Home screen is the typical starting point for work with data modeling. In the Data Modeling section of the Workspace you can use the action items there to create and manage models, forward and reverse engineer, and compare and synchronize schemata. These action items are listed below:

1. Open an Existing EER Model
2. Open an Existing EER Model (icon)
3. Create new EER Model
4. Create EER Model from Existing Database
5. Create EER Model from SQL Script

Open an Existing EER Model

Clicking this action item launches a file browser - you can then select the model file you wish to load. A new MySQL Model tab will then be created, and your model displayed.

Open an Existing EER Model (icon)

If you have already created one or more model files you can simply double-click the item of the model you wish to load. A new MySQL Model tab will be created, and your model displayed.

Create new EER Model

Clicking this action item will launch a new MySQL Model tab, with a blank model ready for you to work on.

Create EER Model from Existing Database

The purpose of this action item is to allow you to create an EER Model from an existing live database. Clicking this action item launches the Reverse Engineer Database. This is a multi-stage wizard that allows you to select a connection to a live server, and select the schema and objects you wish to reverse engineer into your new model. This is a convenient way to see how an existing database is structured.

For further information on reverse engineering see [Section 7.5.9.2, “Reverse Engineering a Live Database”](#).

Create EER Model from SQL Script

The purpose of this action item is to allow you to create a model from a SQL Create script. Such a script may have been created by hand or may be as a result of reverse engineering an existing database to generate the script, which may then be modified according to requirements. Clicking this action item launches the Reverse Engineer SQL Script wizard. This is a multi-stage wizard that allows you to select the script you want to create your model from.

For further information see [Section 7.5.9.1, “Reverse Engineering Using a Create Script”](#).

7.1. Modeling Menus

Some menu options are not available in the OSS version of this application, and are only available in the Standard Edition. This is indicated where applicable.

7.1.1. The File Menu

Use this menu item to open a project, begin a new project, or save a project. Choosing NEW MODEL opens the default schema, mydb. Choosing OPEN MODEL opens a file dialog box with the default file type set to MySQL Workbench Models (MWB). To display a list of recently opened MWB files, choose the OPEN RECENT menu option. The keyboard command to create a new project is **Ctrl N** and the command to open an existing project is **Ctrl O**.

To close the currently active MySQL Model or EER Diagram tab, use the CLOSE TAB option. You can also do this from the keyboard by pressing **Ctrl W**. To reopen the MySQL Model tab, see [Section 7.1.3, “The View Menu”](#). To reopen an EER Diagram tab double click the EER Diagram icon in the EER Diagrams section of the MySQL Model page.

Use the SAVE MODEL or SAVE MODEL AS menu options to save a model. When you save a model its name appears in the title bar of the application. If you have made changes to a project and have not saved those changes, an asterisk appears in the title bar following the model name. When you save a model it is saved as a MySQL Workbench file with the extension mwb.

Use the IMPORT menu option to import a MySQL data definition (DDL) script file, one created by issuing the command `mysql -dump --no-data`, for example. If the script does not contain a `CREATE db_name;` statement, the schema objects will be copied to the default schema, mydb. If the script creates a database, a new tab bearing the database name is added to the Physical Schemata section of the MySQL Model page. If the script contains data, it will be ignored. Importing a DDL script is discussed in detail in [Section 7.5.9.1, “Reverse Engineering Using a Create Script”](#).

Under the Import menu option you can also import DBDesigner4 files.

There are variety of options under the EXPORT menu item. You may generate the SQL statements necessary to create a new database or alter an existing one. These menu items are discussed in detail in [Section 7.5.10.1, “Forward Engineering Using SQL Scripts”](#).

Using the EXPORT menu item you can also export an EER diagram as a PNG, SVG, PDF or Postscript file. For an example of a PNG file see [Figure 7.43, “The sakila EER Diagram”](#).

The PAGE SETUP menu item allows you to set the paper size, orientation and margins for printing purposes.

The print options are only enabled if the EER DIAGRAMS tab is selected. You have the choice of printing your model directly to your printer, printing it as a PDF file, or creating a PostScript file. For more information see [Section 7.7, “Printing”](#). Note: the printing options are only available in commercial versions of MySQL Workbench.

Use the DOCUMENT PROPERTIES menu option to set the following properties of your project:

- Name – Defaults to MySQL Model
- Version – The project version number.
- Author – The project author.
- Project – The project name.
- Created – Not editable, determined by the MWB file attributes.
- Last Changed – Not editable, determined by the MWB file attributes.
- Description – A description of your project.

7.1.2. The Edit Menu

Under this menu item find the options for cutting, copying, and pasting. These actions can also be performed using the **Ctrl X**, **Ctrl C**, and **Ctrl V** key combinations. Undo a deletion using the UNDO DELETE 'OBJECT NAME' option. The **Ctrl Z** key combination can also be used to undo an operation. It is also possible to carry out a REDO operation using either the menu item, or the key combination **Ctrl Y**.

Also find a DELETE 'OBJECT NAME' menu item for removing the currently selected object. The text description for this menu item changes to reflect the name of the currently selected object. The keyboard command for this action is **Ctrl Delete**. You can also right click an object and choose the delete option from the pop-up menu.

The DELETE 'OBJECT NAME' menu item behaves differently depending upon circumstances. For instance, if an EER DIAGRAM is active and a table on the canvas is the currently selected object, a dialog box may open asking whether you want to remove the table from the canvas only or from the database as well. For setting the default behavior when deleting from an EER Diagram see [Section 5.4.2, “The Model Tab”](#).

Warning

If the [MySQL Model](#) page is active, the selected object will be deleted from the catalog and there will be *no confirmation dialog box*.

Choose [EDIT SELECTED](#) to edit the currently selected object. You can also perform edits in a new window by selecting [EDIT SELECTED IN NEW WINDOW](#). The keyboard shortcut for [EDIT SELECTED](#) is **Ctrl E** and **Ctrl Shift E** for [EDIT SELECTED IN NEW WINDOW](#).

The [SELECT](#) option has the following submenus:

- [SELECT ALL](#) (Keyboard shortcut, **Ctrl A**) – Select all the objects on the active EER diagram.
- [SIMILAR FIGURES](#) (Objects of the same type) – Use this option to find objects similar to the currently selected object.
- [CONNECTED FIGURES](#) – Use this option to find all the objects connected to the currently selected object.

These menu items are only active when an **EER DIAGRAM** tab is selected. The [SIMILAR FIGURES](#) and the [CONNECTED FIGURES](#) menu options are disabled if no object is currently selected on an EER diagram.

When multiple objects have been selected using one of these menu options, you can navigate between selected items by choosing the [GO TO NEXT SELECTED](#) or [GO TO PREVIOUS SELECTED](#) menu options.

Selecting items changes some of the [EDIT](#) menu options. If only one object is selected, that object's name appears after the [CUT](#), [COPY](#) and [DELETE](#) menu options. If more than one object is selected, these menu items show the number of objects selected.

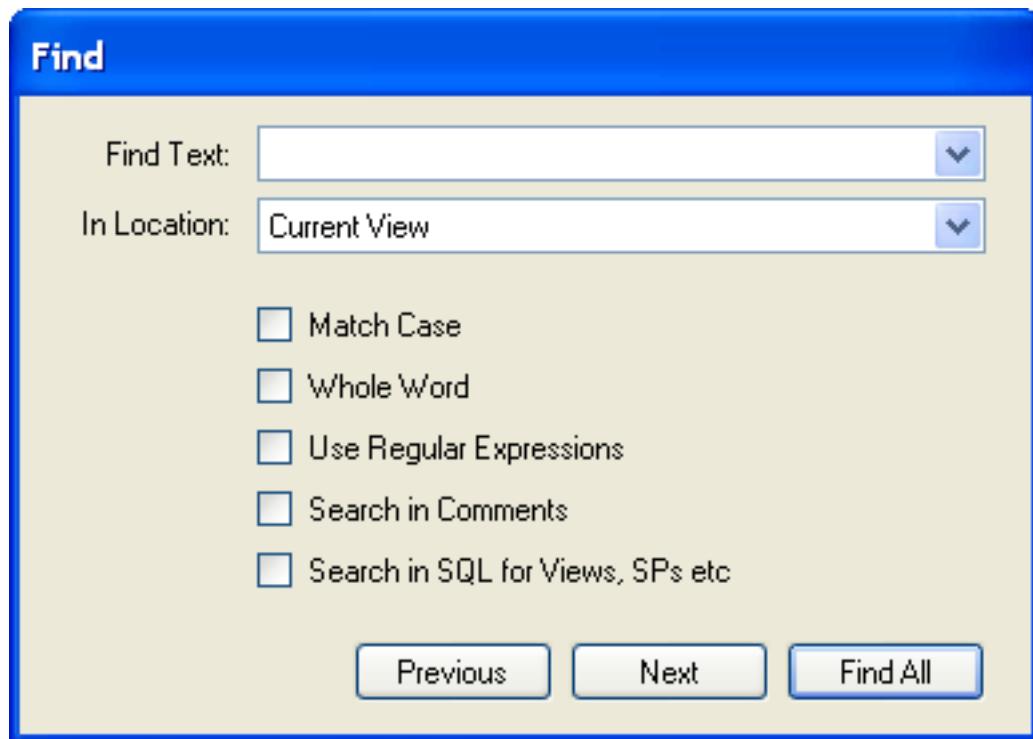
7.1.2.1. Find Dialog Window

The [FIND](#) menu item displays a sub-menu with the following menu items:

- [FIND](#) - takes you to the toolbar search box. You can look for objects in the current view. Find can locate objects in the Model view, the EER Diagram view, and also in the Catalog palette.
- [FIND NEXT](#) - finds the next occurrence of the object.
- [FIND PREVIOUS](#) - finds the previous occurrence of the object.
- [SEARCH AND REPLACE](#) - displays the Search and Replace dialog. This is currently only for use with the SQL Editor, to allow you to quickly search and replace script code items.

The Standard Edition of MySQL Workbench includes a more advanced Find facility:

Figure 7.1. The Find Window



You can search the following locations:

- Entire Model – Search the entire model.
- Current View – Search the current view only. This may be the [MySQL Model](#) page.
- All Views – Search the [MySQL Model Page](#) and all EER diagrams.
- Database Objects – Search database objects only.
- Selected Figures – Search the currently selected objects. This feature only works for EER diagrams.

Enter the text you wish to search for in the **FIND TEXT** drop down list box. You may also select any or all of the following check boxes:

- Match Case
- Whole Word
- Use Regular Expression
- Search in Comments
- Search in SQL for Views, SPs etc.

Any text you enter into the **FIND TEXT** drop down list box is retained for the duration of your session. Use the **NEXT** or **PREVIOUS** buttons to find occurrences of your search criterion.

Clicking the **FIND ALL** button opens a **FIND RESULTS** window anchored at the bottom of the application. If you wish, you may undock this window as you would any other.

Use this window to navigate to objects. For example, double clicking the **Description** of an object located on an EER diagram navigates to the specific diagram and selects the object. Notice that the properties of the object are displayed in the **Properties** palette.

The **Find** dialog window can also be opened using the **Ctrl F** key combination. Use **Ctrl G** to find the next occurrence and **Ctrl Shift G** to find a previous occurrence. Close the **Find** dialog window by clicking the X in the top right corner or by pressing the **Esc** key.

7.1.2.2. Workbench Preferences

This menu option allows you to set global preferences for the MySQL Workbench application.

For further information see [Section 5.4, “Workbench Preferences”](#).

7.1.3. The View Menu

The Options available under this menu item are:

- [HOME](#) – Selects the Home screen.
- [WINDOWS](#)
 - [MODEL NAVIGATOR](#) – Open the [Model Navigator](#) palette
 - [CATALOG](#) – Open the [Catalog](#) palette
 - [LAYERS](#) – Open the [Layers](#) palette
 - [USER DATATYPES](#) – Open the [User Datatypes](#) palette
 - [OBJECT DESCRIPTION](#) – Open the [Description](#) palette
 - [OBJECT PROPERTIES](#) – Open the [Properties](#) palette
 - [UNDO HISTORY](#) – Open the [History](#) palette

These menu options provide a means for opening the windows associated with these options.

- [ADVANCED](#)
 - [OUTPUT WINDOW](#) – Use this option to display the console output. The keyboard shortcut for this menu item is **Ctrl F2**.
 - [GRT SHELL](#) – Open the GRT shell. For more information about the GRT shell see [Chapter 9, The Workbench Scripting Shell](#). The keyboard shortcut for opening the GRT shell is **Ctrl F3**.
- [RESET WINDOW LAYOUT](#) – Reset all windows to their default layout.
- [ZOOM 100%](#) – The default level of detail of an EER diagram.
- [ZOOM IN](#) – Zoom in on an EER diagram.
- [ZOOM OUT](#) – Zoom out from an EER diagram.

The ability to zoom in on an EER diagram is also available using the slider tool in the [Model Navigator](#) palette. See [Section 7.3.5, “The Model Navigator Palette”](#).

- [SET MARKER](#) – Use this option to bookmark an object. From the keyboard select the object you wish to bookmark and use the key combination **Ctrl Shift** and the number of the marker (1 through 9). You may create up to nine markers.
- [GO TO MARKER](#) – Return to a marker. From the keyboard use the **Ctrl** key and the number of the marker.

7.1.4. The Arrange Menu

The [Arrange](#) menu option applies only to objects on an EER diagram canvas and is only visible if an EER diagram view is active. The options under this menu item are as follows:

- [ALIGN TO GRID](#) – Align items on the canvas to the grid lines.
- [BRING TO FRONT](#) – Use this option to bring objects to the foreground.
- [SEND TO BACK](#) – Use this option to move objects to the background.
- [CENTER DIAGRAM CONTENTS](#) – Use this option to center objects on the canvas.
- [AUTOLAYOUT](#) – Use this option to automatically arrange objects on the canvas.
- [RESET OBJECT SIZE](#) – This option expands an object on an EER diagram. For example, if a table has a long column name that is

not fully displayed, using this menu option will expand the table making the column visible. This menu item is not enabled until an object is selected.

- **EXPAND ALL** – Use this option to expand all objects on an EER diagram. This option will display a table's columns if the object notation supports expansion. Some object notations, such as **Classic**, do not allow for expansion or contraction. Indexes will not automatically be expanded unless they were previously expanded and have been collapsed using the **COLLAPSE ALL** menu option.
- **COLLAPSE ALL** – Undo the operation performed by **EXPAND ALL**.

7.1.5. The Model Menu

The menu options available under the **Model** menu item are as follows:

- **ADD DIAGRAM** – Create a new EER Diagram. The keyboard shortcut is **Ctrl T**.
- **CREATE DIAGRAM FROM CATALOG OBJECTS** – Create an EER diagram from all the objects in the catalog.
- **DBDOC – MODEL REPORTING...** – For information on using this menu option see [Section 7.1.5.1, “The DBDoc Model Reporting Dialog Window \(Commercial Version\)”. Commercial version only.](#)
- **USER DEFINED TYPES** – Choosing this menu option presents you with a dialog box, allowing you to add and delete user defined data types.
- **OBJECT NOTATION** – The items available under this option are discussed in [Section 7.1.5.3, “The Object Notation Menu Options”](#).
- **RELATIONSHIP NOTATION** – The items available under this option are discussed in [Section 7.1.5.4, “The Relationship Notation Menu Option”](#).
- **DIAGRAM PROPERTIES AND SIZE** – Choosing this menu option opens a diagram size dialog box. Use this dialog box to adjust the width or height of the canvas. The unit of measure is pages; the default value is two.

When you have tables with numerous columns, use this menu option to increase the size of the EER.

- **VALIDATION** – The items available under this option are discussed in [Section 7.1.5.2, “The Validation Menu Options \(Commercial Version\)”. Commercial version only.](#)
- **MODEL OPTIONS** – Set options at the model level. These options should not be confused with the options that are set globally for the Workbench application, and which are now referred to as Workbench Preferences. The available model options are a subset of the Workbench Preferences options.

For more information on Workbench Preferences see [Section 5.4.2, “The Model Tab”](#).

7.1.5.1. The DBDoc Model Reporting Dialog Window (Commercial Version)

This dialog window is found by navigating to the **MODEL** menu item and choosing the **DBDOC - MODEL REPORTING ...** option.

Note

The **DBDOC - MODEL REPORTING ...** option is not available in the MySQL Workbench OSS version.

Use this dialog window to set the options for creating documentation of your database models.

You can learn about this menu item in more detail in the following section [The DBDoc Model Reporting Dialog Window](#).

7.1.5.2. The Validation Menu Options (Commercial Version)

Under the **MODEL** menu option there are two validation options, **VALIDATION** and **VALIDATION (MySQL)**. Use these options for general validation and MySQL-specific validation of the objects and relationships defined in your model.

Note

These options are not available in the MySQL Workbench OSS version.

Under the **VALIDATION** option the menu items are:

- VALIDATE ALL – Perform all the validation options available
- EMPTY CONTENT VALIDATION – Check for objects with no content, for example a table with no columns
- TABLE EFFICIENCY VALIDATION – Check the efficiency of tables, for example a table with no primary key defined
- DUPLICATE IDENTIFIERS VALIDATION – Check for duplicate identifiers, for example two tables with the same name
- CONSISTENCY VALIDATION – Check for consistent naming conventions
- LOGIC VALIDATION – Check, for example, that a foreign key does not reference a nonprimary key column in the source table

Under the VALIDATION (MySQL) option the menu items are:

- VALIDATE ALL – Perform all the validation options available
- INTEGRITY VALIDATION – Check for invalid references, for example, a table name longer than the maximum allowed
- SYNTAX VALIDATION – Check for correct SQL syntax
- DUPLICATE IDENTIFIERS VALIDATION (ADDITIONS) – Check for objects with the same name

For detailed information about validation see [Section 7.8, “MySQL Workbench Schema Validation Plugins \(Commercial Version\)”](#).

7.1.5.3. The Object Notation Menu Options

The options under the OBJECT NOTATION menu apply exclusively to an EER diagram. They are grayed out if an EER diagram tab is not selected.

The menu options are as follows:

- WORKBENCH (DEFAULT) – Display table columns, indexes, and triggers.
- WORKBENCH (SIMPLIFIED) – Show only a table's columns.
- WORKBENCH (PKS AND FKs ONLY) – Show only columns that are primary and foreign keys.
- CLASSIC – Similar to the [Workbench \(Simplified\)](#) style showing only the table's columns.
- IDEF1X – The ICAM DEFinition language information modeling style.

The object notation style that you choose persists for the duration of your MySQL Workbench session and is saved along with your model. When MySQL Workbench is restarted, the object notation reverts to the default.

Note

If you plan to export or print an EER diagram be sure to decide on a notation style first. Changing notation styles after objects have been placed on a diagram can significantly change the appearance of the diagram.

7.1.5.4. The Relationship Notation Menu Option

The options under the RELATIONSHIP NOTATION menu apply exclusively to an EER diagram. They are grayed out if an EER diagram tab is not selected. The menu options are as follows:

- CROW'S FOOT (IE) – The default modeling style. For an example see [Figure 7.40, “Adding Tables to the Canvas”](#).
- CLASSIC – Uses a diamond shape to indicate cardinality.
- CONNECT TO COLUMNS
- UML – Universal Modeling Language style.
- IDEF1X – The ICAM DEFinition language information modeling method

To view the different styles, set up a relationship between two or more tables and choose the different menu options

The relationship notation style that you choose persists for the duration of your MySQL Workbench session and is saved along with your model. When MySQL Workbench is restarted, the relationship notation reverts to the default, the [Crow's Foot](#) style.

Note

If you plan to export or print an EER diagram be sure to decide on a notation style first. Changing notation styles after objects have been placed on a diagram can significantly change the appearance of the diagram.

7.1.6. The Database Menu

There are several options under the [DATABASE](#) menu option:

- [QUERY DATABASE](#) – Launches the SQL Editor, which allows you to create SQL code and execute it on a live server. For more information see [Section 6.2, “SQL Editor”](#).
- [MANAGE CONNECTIONS](#) – Launches the Manage DB Connections dialog, which allows you to create and manage multiple connections. For more information see [Section 6.1, “Manage Connections”](#)
- [REVERSE ENGINEER](#) – Create a model from an existing database. For more information, see [Section 7.5.9.2, “Reverse Engineering a Live Database”](#).
- [FORWARD ENGINEER](#) – Create a database from a model. For more information, see [Section 7.5.10.2, “Forward Engineering to a Live Server”](#).
- [SYNCHRONIZE MODEL](#) – Synchronize your database model with an existing database. For more information, see [Section 7.5.10.3, “Database Synchronization”](#).
- [GENERATE CATALOG DIFF REPORT](#) – Compare your schema model with a live database or a script file. [Section 7.5.10.4, “Creating a Catalog Diff Report”](#).

7.1.7. The Plugins Menu

The [PLUGINS](#) menu option lists any plugins that you may have installed. For more information about this menu option see [Chapter 10, *Plugins*](#).

7.1.8. The Scripting Menu

This menu currently has two menu items:

The [RUN WORKBENCH SCRIPT](#) menu option.

The [INSTALL PLUGIN/MODULE FILE](#) menu option.

7.1.9. The Community Menu

The [COMMUNITY](#) menu option offers the following choices:

- [WORKBENCH BLOG](#)
- [FAQs ABOUT WORKBENCH](#)
- [LEARN HOW TO CODE FOR WORKBENCH](#)
- [DISCUSS WORKBENCH TOPICS](#)
- [CONTRIBUTE TO WORKBENCH](#)

Use these menu options to go online and learn more about MySQL Workbench.

7.1.10. The Help Menu

The [HELP](#) menu option offers the following choices:

- [HELP INDEX](#) – opens a windows showing the MySQL Workbench documentation. Read, search, or print the documentation

from this window.

- [CHECK FOR UPDATES](#) – open your default browser on the MySQL Workbench website and check for a newer version
- [UPDATE](#) – update to the latest version
- [ADDONS](#) – install or uninstall MySQL Workbench addons
- [MYSQL.COM WEBSITE](#) – open your default browser on the MySQL website home page
- [WORKBENCH PRODUCT PAGE](#) – open your default browser on the MySQL Workbench product page
- [SYSTEM INFO](#) – information about your system, useful when reporting a bug. For more information see [Section 7.1.10.1, “System Info”](#)
- [REPORT A BUG](#) – open your default browser on the MySQL bug report page
- [VIEW REPORTED BUGS](#) – open your default browser to see a list of current bugs
- [ABOUT WORKBENCH](#) – show the MySQL Workbench [About](#) window

Use these menu options to go online and learn more about MySQL Workbench.

7.1.10.1. System Info

Use the [SYSTEM INFO](#) menu option to determine information about your system. This option is especially useful for determining your rendering mode. Sample output follows.

```
read_mysql_cfg_file C:\Program Files\MySQL\MySQL Server 5.1\my.ini
[('tmp_table_size', '9M'),
('myisam_sort_buffer_size', '18M'),
('table_cache', '256'),
('read_rnd_buffer_size', '256K'),
('port', '3306'), ('max_connections', '100'),
('innodb_buffer_pool_size', '18M'),
('myisam_max_sort_file_size', '100G'),
('sql-mode', '"STRICT_TRANS_TABLES,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION"'),
('basedir', '"C:/Program Files/MySQL/MySQL Server 5.1/"'),
('default-character-set', 'latin1'),
('datadir', '"C:/ProgramData/MySQL/MySQL Server 5.1/Data/"'),
('innodb_log_buffer_size', '1M'),
('innodb_log_file_size', '10M'),
('innodb_thread_concurrency', '8'),
('read_buffer_size', '64K'),
('innodb_additional_mem_pool_size', '2M'),
('thread_cache_size', '8'),
('innodb_flush_log_at_trx_commit', '1'),
('query_cache_size', '0'),
('sort_buffer_size', '256K'),
('default-storage-engine', 'INNODB'),
('key_buffer_size', '11M')
MySQL Workbench OSS for Windows version 5.2.8
Cairo Version: 1.8.6
Rendering Mode: GDI requested (create a diagram to confirm)
OpenGL Driver Version: Not Detected
OS: unknown
CPU: Intel(R) Core(TM)2 Duo CPU      T9300 @ 2.50GHz, 1.0 GB RAM
Video adapter info:
Adapter type: VirtualBox Graphics Adapter
Chip Type: VBOX
BIOS String: Version 0xB0C2 or later
Video Memory: 12288 KB
```

7.2. The Toolbar

The MySQL Workbench toolbar is found immediately below the menu bar. The following tools always appear on the toolbar:

- The new document icon – Click this icon to create a new document
- The folder icon – Click this icon to open a MySQL Workbench file (MWB)
- The save icon – Click this icon to save the current MySQL Workbench project
- The right and left arrows – Click the left arrow to perform an “Undo” operation. Click the right arrow to perform a “Redo” operation.

Other tools appear on the toolbar depending upon the context.

When an EER diagram canvas is selected, the following icons appear to the right of the arrow icons:

- The toggle grid icon – Used for turning the grid on and off
- The grid icon – Used for aligning objects on the canvas with the grid

7.2.1. Tool-specific Toolbar Items

The toolbar also changes depending upon which tool from the vertical toolbar is active. These tools are discussed in [Section 7.4.1, “The Vertical Toolbar”](#).

If the [Table](#) tool is active, drop down list boxes of schemata, engine types and collations appear on the toolbar. There is also a color choice drop down list box.

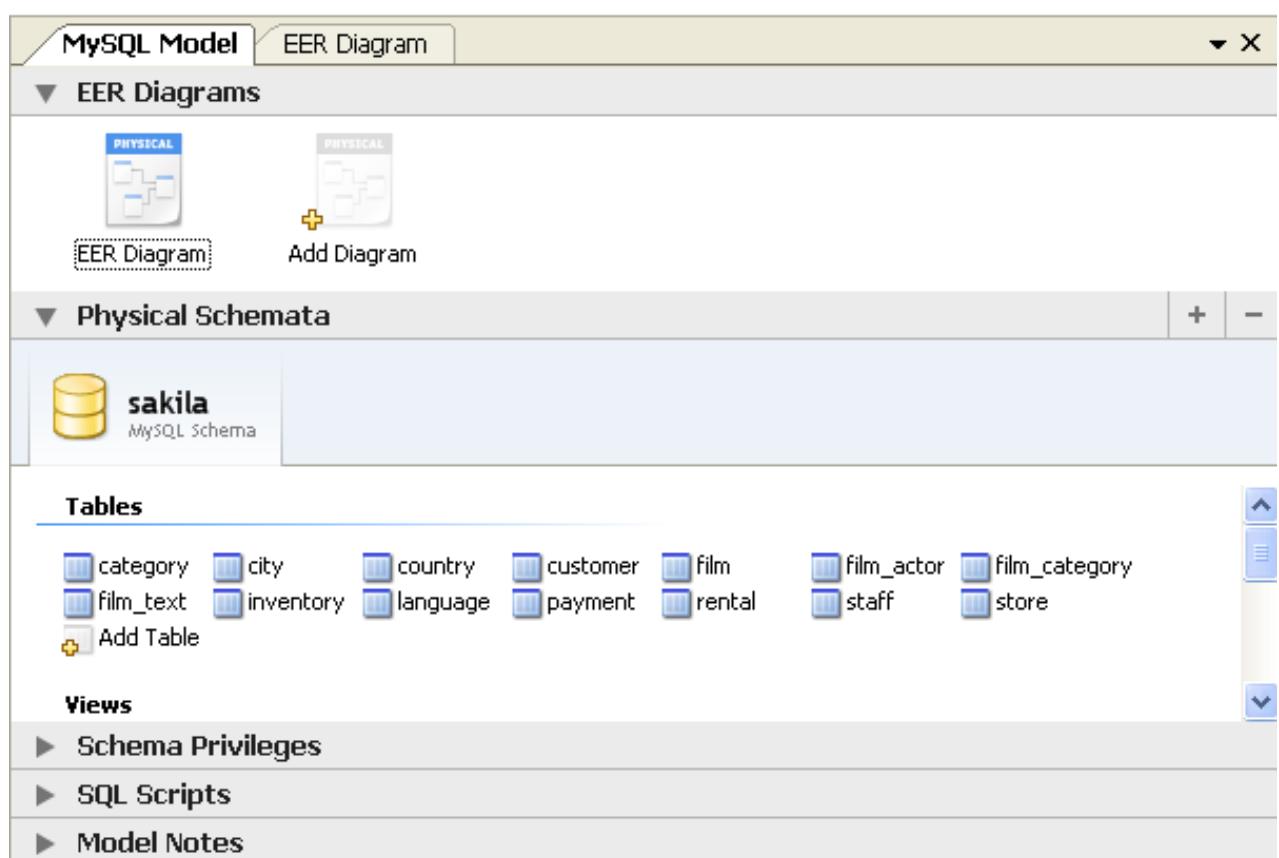
The color choice drop down list box is also displayed when the [Note](#) or [Layer](#) tools are active.

When the [View](#) or [Routine Group](#) tools are active, the schemata list box and the a color choice drop list box are displayed.

7.3. Model Editor

MySQL Workbench opens on the [MySQL Model](#) page with the [MySQL Model](#) tab selected.

Figure 7.2. The MySQL Model Page



The sections within the MySQL Model page are:

- EER Diagrams
- Physical Schemata
- Schema Privileges (Commercial version)

- SQL Scripts
- Model Notes

For each of these sections objects can be added to a project by clicking the appropriate add object icon. You may also rename, edit, cut, copy, or delete objects on this page by right clicking. Doing this opens a pop-up menu.

The sections within the MySQL Model page are discussed in what follows.

7.3.1. EER Diagrams

Use the [Add Diagram](#) icon in this area to create EER diagrams. When you add an EER diagram a new tab appears below the toolbar. Use this tab to navigate to the newly created EER diagram. EER Diagrams are discussed in depth in [Section 7.4, “EER Diagram Editor”](#).

7.3.2. The Physical Schemata

The [Physical Schemata](#) panel of the [MySQL Model](#) page shows the active schemata and the objects that they contain.

Expand and contract the [Physical Schemata](#) section by double clicking the arrow on the left of the [Physical Schemata](#) title bar. When the [Physical Schemata](#) section is expanded, all the schemata that are currently loaded are displayed.

Each schema shows as a tab; a specific schema is selected by clicking its tab. When MySQL Workbench is first opened a default schema, `mydb` is selected. You can start working with this schema or you can load a new MySQL Workbench Models (MWB) file.

There are a variety of ways to add schema to the [Physical Schemata](#) panel. You can open an MWB file, reverse engineer a MySQL create script, or, if you are using a commercial version of MySQL Workbench, you can reverse engineer a database by connecting to a MySQL server.

You can also add a new schema by clicking the + button on the top right of the [Physical Schemata](#) panel. To remove a schema, click its tab and use the - button found to the immediate left of the + button. To the left of these buttons are three buttons that control the way database object icons are displayed. The left-most button displays database objects as large icons, the next button to the right displays small icons in multiple rows, and the last button displays small icons in a single list.

To expand the Physical Schemata window move the mouse pointer anywhere over the gray area that defines the lower edge of the [Physical Schemata](#) panel. Hold down the right mouse button and move the mouse to adjust the size of the window.

7.3.2.1. Schema Objects

The [Physical Schemata](#) panel is divided up into the following sections:

- Tables
- Views
- Routines
- Routine Groups

Each section contains the specified database objects and an icon used for creating additional objects.

Any objects added to the [Physical Schemata](#) show up in the [Catalog](#) palette docked on the right side of the application. Any database objects added to an EER diagram canvas also show up in the [Physical Schemata](#) section. For information about adding objects to an EER diagram canvas see [Section 7.4, “EER Diagram Editor”](#).

7.3.3. Schema Privileges

The [Privileges](#) panel of the [MySQL Model](#) page is used to create users for your schemata and also to define roles — making it much easier to assign the same rights to different users or different rights to different users.

The [Schema Privileges](#) panel is divided up into the following sections:

- Users
- Roles

The following image displays the **Schema Privileges** section of the **MySQL Model** tab.

Figure 7.3. Roles and Privileges

7.3.3.1. Adding Roles

To add a role double click the **Add Role** icon. Doing this creates a role with the default name `role1`. Right clicking a role opens a pop-up menu with the following options:

- RENAME – Rename the role
- CUT '`ROLE_NAME`' – Cut the role
- COPY '`ROLE_NAME`' – Copy the role
- EDIT ROLE ... – Open the role editor.
- EDIT IN NEW WINDOW – Open the role editor in a new editor window.
- DELETE '`ROLE_NAME`' – Remove the role

Use the **Rename** option to give a newly created role a name descriptive of its function.

All roles that have been defined are listed under **Roles** on the left side of the role editor. Double clicking a role object opens the role editor docked at the bottom of the screen.

Figure 7.4. Role Editor

Select the role that you wish to add objects to. You may drag and drop objects from the [Physical Schemata](#) or from the [Catalog](#) panel to the [Objects](#) section of the role editor. To assign privileges to a role select a role from the [Roles](#) section and then select an object in the [Objects](#) section. In the [Privileges](#) section check the rights you wish to assign to this role. For example, a [web_user](#) role might have only [SELECT](#) privileges and only for database objects exposed through a web interface. Creating roles can make the process of assigning rights to new users much easier.

7.3.3.2. Adding Users

To add a user double click the [Add User](#) icon. Doing this creates a user with the default name [user1](#). Double clicking this user opens the user editor docked at the bottom of the application.

In the [User Editor](#), set the user's name using the **NAME** text box and set the password using the **PASSWORD** text box. Assign one role or a number of roles to the user by selecting the desired roles from the text area on the right and then clicking the < button. Roles may be revoked by moving them in the opposite direction.

Right clicking a user opens a pop-up menu. These options function as described in [Section 7.3.3.1, “Adding Roles”](#).

7.3.4. SQL Scripts and Model Notes

The two remaining panels on the [MySQL Model](#) page are [SQL Scripts](#) panel and the [Model Notes](#) panel.

Use the [SQL Scripts](#) panel to load and modify SQL scripts. If you created your project from an SQL script and plan to create an [ALTER](#) script, you may want to add the original script here, since it will be needed in order to create an [ALTER](#) script. For more information, see [Section 7.5.10.1.2, “Altering a Schema”](#).

Use the [Model Notes](#) panel to write project notes. Any scripts or notes added will be saved with your project.

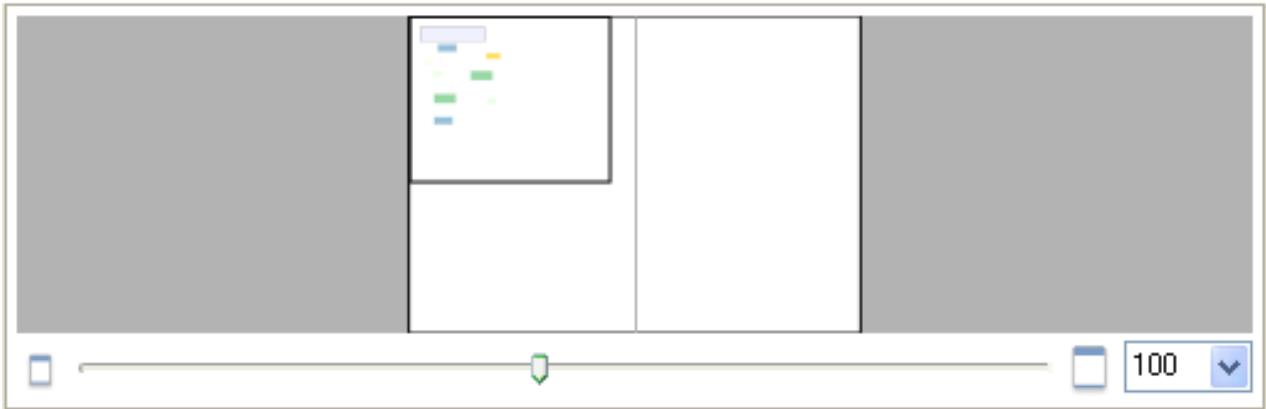
7.3.5. The Model Navigator Palette

Docked at the top right of the application is the [Model Navigator](#) palette. This palette gives you an overview of the objects placed on an EER diagram canvas and for this reason it is most useful when an EER diagram is active. Any objects that you have placed on the canvas should be visible in the navigator.

The [Model Navigator](#) palette shows the total area of an EER diagram. A black rectangular outline indicates the view port onto the visible area of the canvas. To change the view port of an EER diagram left click this black outline and drag it to the desired location. You can zoom in on selected areas of an EER diagram by using the slider tool at the bottom of this window. The dimensions of the view port change as you zoom in and out. If the slider tool has the focus you can also zoom using the arrow keys.

The default size of the [Model Navigator](#) is two pages. To change this use the [MODEL, DIAGRAM SIZE](#) menu option.

Figure 7.5. The Model Navigator Palette



7.3.6. The Catalog and Layers Palettes

By default these two palettes are docked on the right, in the middle of the application. You can select one or the other by clicking the tab at the bottom of this palette.

7.3.6.1. The Catalog Palette

The [Catalog](#) palette shows all the schemata that are present in the [Physical Schemata](#) section of the [MySQL Model](#) page.

Expand the view of the objects contained in a specific schema by clicking the + button to the left of the schema name. Doing this displays the following folder icons:

- Tables
- Views
- Routine Groups

Expand each of these in turn by clicking the + button to the left of the folder icon.

Selecting an object in this palette, displays its properties in the [Properties](#) palette.

This palette is principally used to drag and drop objects onto an EER diagram canvas.

You can autohide this window by clicking the push pin icon on the title bar. Doing this displays a vertical tab on the right side of the application. Mouse over this tab to view its contents. Autohiding a window when it is a tabbed window, autohides all the tabbed windows at that location.

7.3.6.2. The Layers Palette

This palette shows all the layers and figures that have been placed on an EER diagram. If a layer or figure is currently selected, an [X](#) appears beside the name of the object and its properties are displayed in the [Properties](#) palette. This can be especially useful in determining which objects are selected when you have selected multiple objects using the various options under the [SELECT](#) menu option. For more information on this topic see [Section 7.1.2, “The Edit Menu”](#).

Selecting an object in the [Layers](#) palette also adjusts the view port to the area of the canvas where the object is located.

You can autohide this window by clicking the push pin icon on the title bar. Doing this displays a vertical tab on the right side of the application. Mouse over this tab to view its contents. Autohiding a window when it is a tabbed window, autohides all the tabbed windows at that location.

7.3.6.2.1. Finding Invisible Objects Using the Layers Palette

In some circumstances you may want to make an object on an EER diagram invisible. To do this, select the object and, in the [Properties](#) palette, set the [visible](#) property to [False](#).

The [Layer](#) palette provides an easy way to locate an invisible object. Open the [Layers](#) palette and select the object by double clicking it. Once an object is selected you can reset the [visible](#) property from the [Properties](#) palette.

7.3.7. The Properties and History Palettes

By default these two palettes are docked on the right, at the bottom of the application. You can select one or the other by clicking the tab at the bottom of this palette.

7.3.7.1. The Properties Palette

The [Properties](#) palette is used to display and edit the properties of objects on an EER diagram. It is especially useful for editing display objects such as layers and notes.

All objects except connections have the following properties except as noted:

- [color](#) – The color accent of the object. The color of the object is displayed here as is its hexadecimal value. Change the color of the object by changing this value. Only characters that are legal for hexadecimal values may be entered. You can also change the color by clicking the ... button. This opens a color changer dialog box.
- [description](#) – Applicable to layers only. A means of documenting the purpose of a layer.
- [expanded](#) – This attribute applies to objects such as tables that can be expanded to show columns, indexes, and triggers.
- [height](#) – The height of the object. Depending upon the object, this property may be read only or read/write.
- [left](#) – The number of pixels from the object to the left side of the canvas.
- [locked](#) – Whether the object is locked or not. The value for this attribute is either [true](#) or [false](#).
- [manualSizing](#) – Whether the object has been manually sized or not. The value for this attribute is either [true](#) or [false](#).
- [name](#) – The name of the object.

- `top` – The number of pixels from the object to the top of the canvas.
- `visible` – This property controls whether an object shows up on the canvas or not. Use ‘`1`’ for true and ‘`0`’ for false. It is currently only used for relationships.
- `width` – The width of the object. Depending upon the object, this property may be read only or read/write.

In addition to the properties listed above, tables also have the following properties:

- `indexesExpanded` – This property determines whether indexes are displayed when a table is placed on the canvas. Use ‘`1`’ for true and ‘`0`’ for false.
- `triggersExpanded` – This property determines whether triggers are displayed when a table is placed on the canvas. Use ‘`1`’ for true and ‘`0`’ for false.

For a discussion of the properties of connections see [Section 7.5.2.3, “The Properties of a Connection”](#).

7.3.7.2. The History Palette

Use the `History` palette to review the actions that you have taken. Left clicking an entry opens a pop-up menu with the option, `COPY HISTORY ENTRIES TO CLIPBOARD`. Choose this option to select a single entry. You can select multiple contiguous entries by pressing the `Shift` key and clicking the entries you wish to copy. Select noncontiguous entries by using the `Ctrl` key.

Only actions that alter the MySQL model or change an EER diagram are captured by the `History` palette.

7.4. EER Diagram Editor

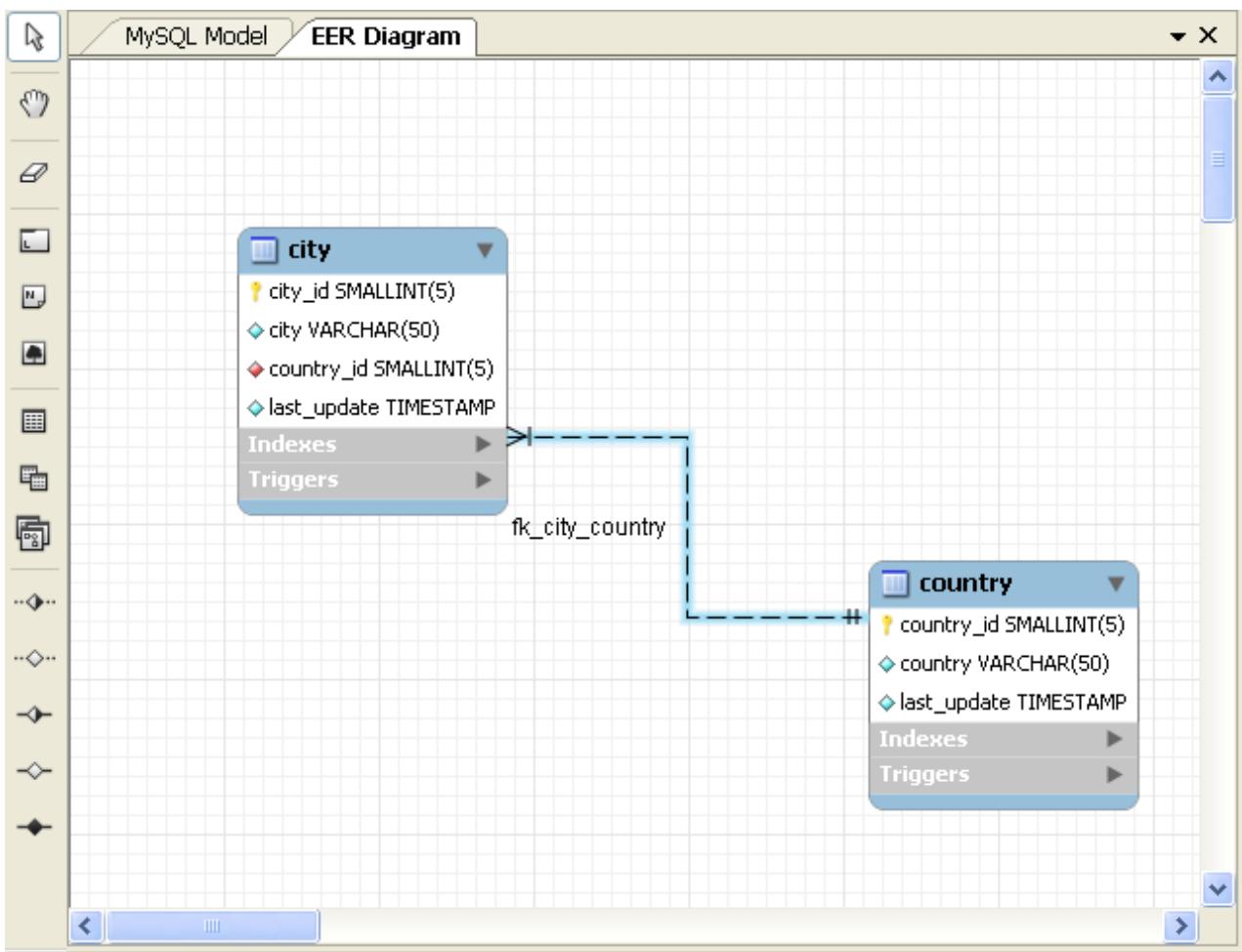
EER diagrams are created by double clicking the `Add Diagram` icon. You may create any number of EER diagrams just as you may create any number of physical schemata. Each EER diagram shows as a tab below the toolbar; a specific EER diagram is selected by clicking its tab.

Clicking an EER diagram tab navigates to the canvas used for graphically manipulating database objects. On the left side of this page is the `Vertical Toolbar`.

7.4.1. The Vertical Toolbar

The vertical toolbar shows on the left sidebar when an EER diagram tab is selected. The tools on this toolbar assist in creating EER diagrams.

Figure 7.6. The Vertical Toolbar



Clicking a tool changes the mouse pointer to a pointer that resembles the tool icon, indicating which tool is active. These tools can also be activated from the keyboard by pressing the key associated with the tool. Locating the mouse over a toolbar icon displays a description of the tool and its shortcut key.

A more detailed description of each of these tools follows.

7.4.1.1. The Standard Mouse Pointer

The standard mouse pointer, located at the top of the vertical toolbar, is the default mouse pointer for your operating system. Use this tool to revert to the standard mouse pointer after using other tools.

From the keyboard, use the **Esc** key to revert to the default pointer.

7.4.1.2. The Hand Tool

The hand tool is used to move the entire EER diagram. Left click on this tool and then left click anywhere on the EER diagram canvas holding down the mouse button. Moving the mouse changes the view port of the canvas.

To determine your position on the canvas look at the [Model Navigator](#) panel on the upper right. If the [Model Navigator](#) panel is not open, use the [VIEW](#), [WINDOWS](#), [MODEL NAVIGATOR](#) to open it.

From the keyboard, use the **H** key to activate this tool.

You can also change the view port of an EER diagram using the [Model Navigator](#) panel. To do this see [Section 7.3.5, “The Model Navigator Palette”](#).

7.4.1.3. The Eraser Tool

Use the eraser tool to delete objects from the EER Diagram canvas.

Change the mouse pointer to the eraser tool and click the object you wish to delete. Depending upon your settings, the delete dialog

box should open, asking you to confirm the type of deletion.

Note

The delete action of the [eraser](#) tool is controlled by the general option setting for deletion. Be sure that you understand the available options described in [Section 5.4.2, “The Model Tab”](#) before using the eraser tool.

From the keyboard, use the **D** key to activate this tool.

In addition to using the [eraser](#) tool, you can also delete an object by selecting it and pressing **Ctrl Delete** or right clicking it and choosing [DELETE](#) from the pop up menu.

7.4.1.4. The Layer Tool

The layer tool is the rectangular icon with a capital [L](#) in the lower left corner.

The layer tool is used to organize the objects on an EER Diagram canvas. It is useful for grouping together similar objects. You may, for instance, use it to group all your views together.

Click the layer tool and use it to draw a rectangle on the canvas. Change to the standard mouse pointer tool and pick up any objects you would like to place on the newly created layer.

To change the size of a layer, first select it by clicking on it. When a layer is selected small rectangles appear at each corner and in the middle of each side. Adjust the size by dragging any one of these rectangles.

You can also make changes to a layer by selecting the layer and changing properties in the **PROPERTIES** panel. Using the **PROPERTIES** panel is the only way to change the name of a layer.

From the keyboard, use the **L** key to activate this tool. For more information about layers see [Section 7.5.5, “Creating Layers”](#).

7.4.1.5. The Text Tool

The text tool is the square icon with a capital [N](#) in the top left corner. Use this tool to place text objects on the EER diagram canvas. Click the tool and then click the desired location on the canvas. Once a text object has been dropped on the canvas, the mouse pointer reverts to its default.

To add text to a text object, right click the text object and choose either of the pop-up menu options, [EDIT NOTE ...](#) or [EDIT IN NEW WINDOW ...](#).

You can manipulate the properties of a text object by selecting it and then changing its properties in the **Properties** panel.

From the keyboard, use the **N** key to activate this tool. For more information about text objects see [Section 7.5.7, “Creating Text Objects”](#).

7.4.1.6. The Image Tool

Use the image tool to place an image on the canvas. When this tool is selected and you click on the canvas, a dialog box opens allowing you to select the desired graphic file.

From the keyboard, use the **I** key to activate this tool. For more information about images see [Section 7.5.8, “Creating Images”](#).

7.4.1.7. The Table Tool

Use this tool to create a table on the EER Diagram canvas.

Clicking on the canvas, creates a table. To edit this table, right click it and choose [EDIT TABLE](#) or [EDIT IN NEW WINDOW](#) from the pop-up menu. You can also simply double-click the table to load it into the table editor.

From the keyboard, use the **T** key to activate this tool.

For more information about creating and editing tables see [Section 7.5.1.3, “The MySQL Table Editor”](#).

7.4.1.8. The View Tool

Use this tool to create a view on an EER Diagram canvas.

When this tool is activated, a schema drop-down box appears on the toolbar below the main menu, allowing you to associate the new view with a specific schema. You can also select a color for the object by choosing from the color drop down list box to the right of the schema list box.

After selecting this tool, clicking on the canvas creates a new view. To edit this view, right click it and choose [EDIT VIEW](#) or [EDIT IN NEW WINDOW ...](#) from the pop-up menu.

From the keyboard, use the **V** key to activate this tool.

For more information about creating and editing views see [Section 7.5.3, “Creating Views”](#).

7.4.1.9. The Routine Group Tool

Use this tool to create a routine group on the EER Diagram canvas.

When this tool is activated, a schema drop-down box appears on the toolbar below the main menu, allowing you to associate the routine group with a specific schema. You can also select a color for the routine group by choosing from the color drop down list box to the right of the schema list box.

After selecting this tool, clicking in the canvas creates a new group. To edit this view, right click it and choose [EDIT ROUTINE GROUP](#) or [EDIT IN NEW WINDOW ...](#) from the pop-up menu.

From the keyboard, use the **G** key to activate this tool.

For more information about creating and editing routine groups see [Section 7.5.4.2, “Routine Groups”](#).

7.4.1.10. The Relationship Tools

The five relationship tools are used to represent the following relationships:

- One-to-many nonidentifying relationships
- One-to-one nonidentifying relationships
- One-to-many identifying relationships
- One-to-one identifying relationships
- Many-to-many identifying relationships

These tools appear at the bottom of the vertical tool bar. Mouse over each tool to see a text hint that describes its function.

For more information about relationships see [Section 7.5.2, “Creating Foreign Key Relationships”](#).

7.5. Working with Models

7.5.1. Creating Tables

7.5.1.1. Adding Tables to the Physical Schemata

Double clicking the [Add table](#) icon in the [Physical Schemata](#) section of the [MySQL Model](#) page adds a table with the default name of `table1`. If a table with this name already exists, the new table is named `table2`.

Adding a new table automatically opens the table editor docked at the bottom of the application. Using the table editor is described in [Section 7.5.1.3, “The MySQL Table Editor”](#).

Right clicking a table opens a pop-up menu with the following options:

- [CUT '`TABLE_NAME`'](#)
- [COPY '`TABLE_NAME`'](#)
- [EDIT TABLE](#)
- [EDIT IN NEW WINDOW](#)
- [COPY SQL TO CLIPBOARD](#)
- [DELETE '`TABLE_NAME`'](#)

If the table editor is not open the EDIT TABLE ... option opens it. If it is already open, the selected table replaces the previous one. EDIT IN NEW WINDOW opens a new table editor tab.

The cut and copy options are useful for copying tables between different schemata.

Warning

Use the DELETE 'TABLE NAME' to remove a table from the database. There will be **no** confirmation dialog box.

Any tables added to the Physical Schemata also show up in the Catalog palette on the right side of the application. They may be added to an EER Diagram by dragging and dropping them from this palette.

7.5.1.2. Adding Tables to an EER Diagram

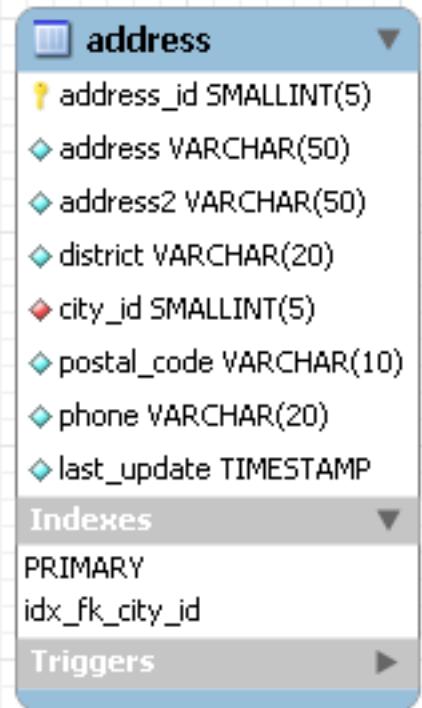
Tables can also be added to an EER Diagram using the table tool on the vertical toolbar. To do this make sure that the EER Diagram tab is selected, and right click the table icon on the vertical toolbar. The table icon is the rectangular tabular icon.

Clicking the mouse on this icon changes the mouse pointer to a table pointer. You can also change the mouse pointer to a table pointer by pressing the **T** key.

Choosing the table tool changes the contents of the toolbar that appears immediately below the menu bar. When the Tables pointer is active, this toolbar contains a drop down list box of schemata, a drop down list box of engines, a drop down list box of collations, and a drop down color chart. Use these list boxes to select the appropriate schema, engine, collation, and color accent for the new table. Make sure that you associate the new table with a database. The engine and collation of a table can easily be changed from the table editor and the color of your table can be changed later using the Properties palette. The Default Engine and Default Collation values refer to the database defaults.

Create a table by clicking anywhere on the EER Diagram canvas. Doing this creates a new table with the default name table1. To revert to the default mouse pointer, click the arrow icon at the top of the vertical toolbar.

Figure 7.7. A Table on an EER Diagram



As shown in the preceding diagram the primary key is indicated by a key icon and indexed fields are indicated by a different colored diamond icon. Click the arrow to the right of the table name to toggle the display of the fields. Toggle the display of indexes and triggers in the same way.

Right clicking a table opens a pop-up menu with the following options:

- CUT 'TABLE NAME'

- [COPY '*TABLE_NAME*'](#)
- [EDIT TABLE](#)
- [EDIT IN NEW WINDOW](#)
- [COPY SQL TO CLIPBOARD](#)
- [DELETE '*TABLE_NAME*'](#)

With the exception of the deletion option, these menu options function as described in [Section 7.5.1.1, “Adding Tables to the Physical Schemata”](#). The behavior of the delete option is determined by your MySQL Workbench options settings. For more information, see [Section 5.4.2, “The Model Tab”](#).

7.5.1.3. The MySQL Table Editor

The MySQL Table Editor is a component that enables the creation and modification of tables. Using the MySQL Table Editor you can add or modify a table's columns or indexes, change the engine, add foreign keys, or simply alter the table's name.

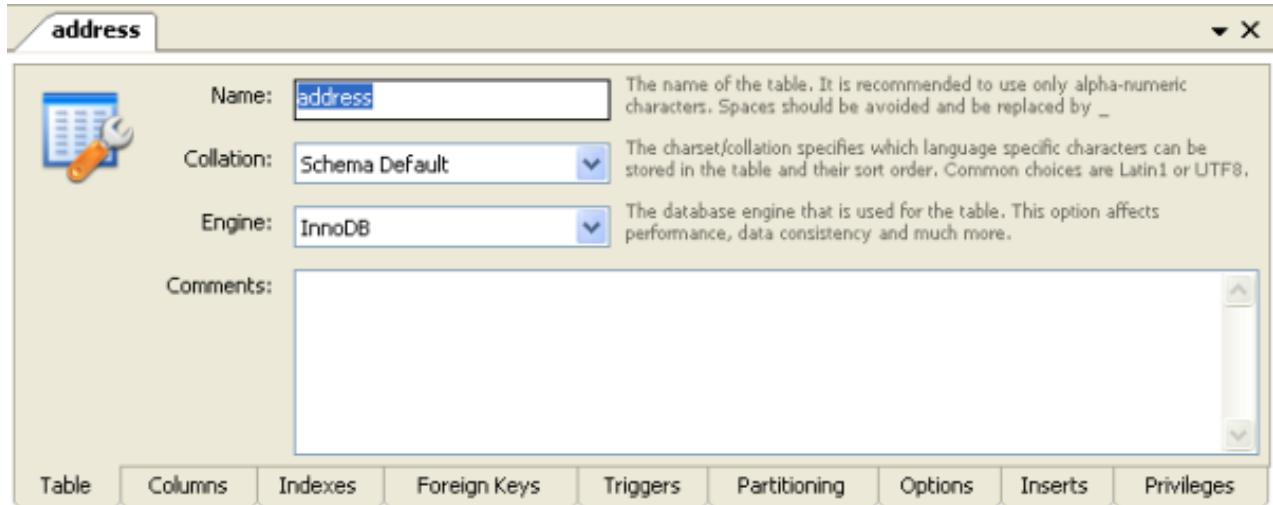
The MySQL Table Editor can be accessed from the MySQL Workbench by first selecting the **MySQL MODEL** tab and then double clicking a table in the **Physical Schemata** panel. You can also access it from an EER Diagram by double clicking a table object.

7.5.1.3.1. The Main Editor Window

Any number of tables may be edited in the MySQL Table Editor at any one time. Adding an additional table creates a new tab at the top of the editor. By default the MySQL Table Editor appears docked at the bottom of the application.

Double clicking the title bar undocks the editor. Do the same to redock it. The MySQL Table Editor is shown in the following figure.

Figure 7.8. The Table Editor



The MySQL Table Editor consists of a work space divided into the following tabs:

- **TABLE:** Use this tab to edit features that apply to the table as a whole
- **COLUMNS:** Use this tab to add or modify columns
- **INDEXES:** Use this tab to add or modify indexes
- **FOREIGN KEYS:** Use this tab to add or modify foreign keys
- **TRIGGERS:** Use this tab to add or modify triggers
- **PARTITIONING:** Use this tab to manage partitioning
- **OPTIONS:** Use this tab to add or modify various general, table and row level options

- **INSERTS:** Use this tab for writing INSERT statements
- **PRIVILEGES:** Use this tab to set privileges on the table

Each of these tabs is discussed in further detail in the following sections.

7.5.1.3.2. The Table Tab

Use this tab to edit the table name or add a comment to the table. Easily change the collation or the table engine using drop down list boxes.

7.5.1.3.3. The Columns Tab

The [Columns](#) tab is used to display and edit all the column information for a table. Using this tab, you can add, drop, and alter columns.

You can also use the column tab to change the name, data type, default value, and other properties of your table's columns.

Figure 7.9. The Columns Tab

To add a column simply click the [Column Name](#) field in an empty row and enter an appropriate value. Select a data type from the [DATATYPE](#) drop down list box. Select a column property checkbox as required according to the following list of column properties:

- **PK** - Primary key.
- **NN** - Not null.
- **UQ** - Unique.
- **BIN** - Binary.
- **UN** - Unsigned.
- **ZF** - Zero fill.
- **AI** - Autoincrement.

Right clicking a row under the [Column Name](#) column opens a pop-up window with the following options:

- [MOVE UP](#) – Move the selected column up.
- [MOVE DOWN](#) – Move the selected column down.
- [DELETE SELECTED COLUMNS](#) – Select multiple contiguous columns by right clicking and pressing the **Shift** key. Use the **Ctrl** key to select noncontiguous columns.
- [REFRESH GRID](#) – Update all information in the [Columns](#) tab.

To change the name, data type, default value, or comment of a column, double click on the value you wish to change. The content then becomes editable.

You can also add column comments to the [Column Comment](#) text area.

To the left of the column name is an icon that indicates whether the column is a member of the primary key. If the icon is a small

key, that column belongs to the primary key, otherwise the icon is a blue diamond. To add or remove a column from the primary key, double click on the icon. You can also add a primary key by checking the **PRIMARY KEY** checkbox in the **Column Details** section of the table editor.

If you wish to create a composite primary key you can select multiple columns and check the PK checkbox. However, there is an additional step that is required, you will need to click the Indexes tab, then in the Index Columns panel you need to set the desired order of the primary keys.

Note

When entering default values, in the case of **CHAR** and **VARCHAR** data types MySQL Workbench will attempt to automatically add quote marks, if the user does not start their entry with one. For other data types the user must manage quoting if required, as it will not be handled automatically by MySQL Workbench.

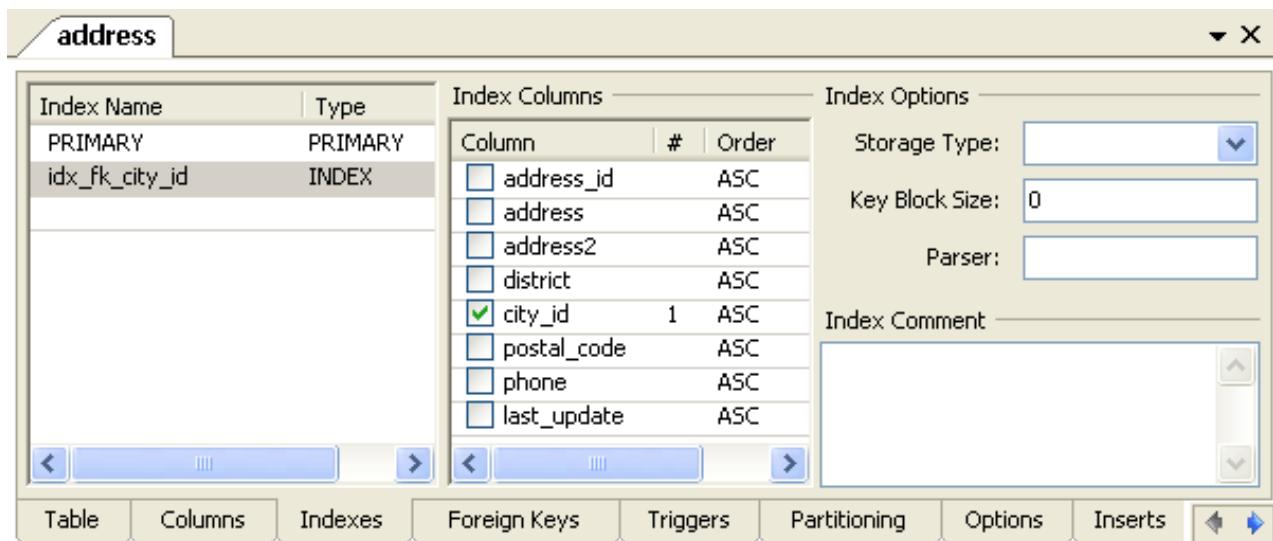
Caution

Care must be taken when entering a default value for non-numeric **ENUM** columns. When entering a non-numeric default value it will not be automatically quoted. You must manually add single quote characters for the default value. Note that MySQL Workbench will **not** prevent you from entering the default value without the single quotes. If a non-numerical default value is entered without quotes, this will lead to errors. For example, if the model is reverse engineered, the script will contain unquoted default values for **ENUM** columns and will fail if an attempt is made to run the script on MySQL Server.

7.5.1.3.4. The Indexes Tab

The **Indexes** tab holds all index information for your table. You can add, drop, and modify indexes using this tab.

Figure 7.10. The Indexes Tab



Select an index by right clicking it. Doing this displays information about the index in the **INDEX COLUMNS** section.

To add an index, click the last row in the index list. Enter a name for the index and select the index type from the drop down list box. Select the column or columns that you wish to index by checking the column name in the **INDEX COLUMNS** list. You can remove a column from the index by removing the check mark from the appropriate column.

You can also specify the order of an index by choosing **ASC** or **DESC** under the **Order** column. Create an index prefix by specifying a numeric value under the **Length** column. You cannot enter a prefix value for fields that have a data type that does not support prefixing.

To drop an index, right click the row of the index you wish to delete and then select the **DELETE SELECTED INDEXES** menu option.

7.5.1.3.5. The Foreign Keys Tab

The **Foreign Keys** tab is organized in much the same fashion as the **Indexes** tab and adding or editing a foreign key is similar to adding or editing an index.

To add a foreign key, click the last row in the **Foreign Key Name** list. Enter a name for the foreign key and select the column

or columns that you wish to index by checking the column name in the **COLUMN** list. You can remove a column from the index by removing the check mark from the appropriate column.

Under **FOREIGN KEY OPTIONS** choose an action for the update and delete events.

The options are:

- RESTRICT
- CASCADE
- SET NULL
- NO ACTION

To drop a foreign key, right click the row you wish to delete and then select the DELETE SELECTED FKs menu option.

To modify any of the properties of a foreign key, simply select it and make the desired changes.

7.5.1.3.6. The Triggers Tab

The **Triggers** tab opens a text area for editing an existing trigger or creating a new trigger. Create a trigger as you would from the command line.

7.5.1.3.7. The Partitioning Tab

If you wish to enable partitioning for your table check the **ENABLE PARTITIONING** check box. Doing this enables the partitioning options.

The **PARTITION BY** drop down list box displays the types of partitions you can create. These are:

- HASH
- LINEAR HASH
- KEY
- LINEAR KEY
- RANGE
- LIST

Use the **PARAMETERS** text box to define the parameter(s) that will be supplied to the partitioning function, an integer column value for example.

Choose the number of partitions from the **PARTITION COUNT** drop down list box. If you wish to manually configure your partitions check the **MANUAL** check box. Doing this enables entry of values into the partition configuration table. The entries in this table are:

- Partition
- Values
- Data Directory
- Index Directory
- Min Rows
- Max Rows
- Comment

Subpartitioning is not yet enabled. For more information about partitioning see [Partitioning](#).

7.5.1.3.8. The Options Tab

The **OPTIONS** tab allows you to set the general options and row options.

In the **GENERAL OPTIONS** frame, choose a pack keys option. The options are **Default**, **Pack None**, and **Pack All**. You may also encrypt the definition of a table. The **AUTO_INCREMENT** and delayed key update behaviors apply only to MyISAM tables.

To set the row format, choose the desired row format from the drop-down list. See [MyISAM Table Storage Formats](#) for more information about the different row formats that are available. This only applies to MyISAM tables.

These options are:

- Default
- Dynamic
- Fixed
- Compressed
- Redundant
- Compact

When you expect a table to be particularly large, use the **Avg. Row**, **Min. Rows**, and **Max. Rows** options to enable the MySQL server to better accommodate your data. See [CREATE TABLE Syntax](#) for more information on how to use these options.

The **Storage Options** section is used to configure a custom path to the table storage and data files. This option can help improve data integrity and server performance by locating different tables on different hard drives. This option is only available for MyISAM tables.

The **Merge Table** Options section is used to configure MERGE tables in MyISAM. To create a MERGE table, select **MERGE** as your storage engine and then specify the tables you wish to MERGE in the **UNION TABLES** dialog.

You can also specify the action the server should take when users attempt to perform INSERT statements on the merge table. See [The MERGE Storage Engine](#) for more information about MERGE tables. Again, this only applies to MyISAM tables. You may also select the **Merge Method** by selecting from the drop down list box.

7.5.1.3.9. The Inserts Tab

Use the **Inserts** tab to insert records into the table. Clicking the **OPEN EDITOR ...** button on the lower right hand side opens the **Standard Inserts** editor. Use this editor to add records to the table.

When you have finished adding records, press **OK**. The records you have added are displayed in the **Inserts** tab. Reopening the editor displays all the records shown in the **Inserts** tab. To edit a record simply click on the field you wish to change and enter the new data. To delete a record, click the button on the left beside the desired row and then press the **Delete** key. You can select and delete all records by clicking in the top left hand column of the editor and then pressing the **Delete** key. Your changes will not be applied until you press the **OK** button. To back out of an operation, press the **CANCEL** button.

Any records you add will be inserted when you forward engineer the database (if you choose the **Generate INSERT statements for tables** option).

Note when entering string values there is slightly different behavior between the 5.0, 5.1 and 5.2 versions of MySQL Workbench.

For 5.0 and 5.1 if a string is entered without leading and trailing quotes, the Inserts Editor adds quoting and escapes characters that require it. However, if quoted text is entered, the Inserts Editor carries out no further checks since it assumes a correctly escaped and quoted sequence has been entered.

5.2 features a new Inserts Editor. In this case the user enters the string without quoting or escaping and the Inserts Editor takes care of all quoting and escaping as required.

Note

It is possible to enter a function, or other expression, into a field. If doing so, the prefix `\func` should be used, to prevent MySQL Workbench from escaping quotes. For example, if entering the expression `md5('fred')` MySQL Workbench would generate the code `md5('fred')`. To prevent this enter the expression as `\func md5('fred')`. This will ensure that the quoting is not escaped.

7.5.1.3.10. The Privileges Tab

Use the **Privileges** tab to assign specific roles and privileges to a table. You may also assign privileges to a role using the role

editor. For a discussion of this topic see [Section 7.3.3.1, “Adding Roles”](#).

When this tab is first opened, all the roles that have been created are displayed in the list box on the right. Move the roles you wish to associate with this table to the **ROLES** list box on the left. Do this by selecting a role and then clicking the < button. Use the **Shift** key to select multiple contiguous roles and the **Ctrl** key to select noncontiguous roles.

To assign privileges to a role click on a role in the **ROLES** list box. Doing this displays all available privileges in the **ASSIGNED PRIVILEGES** list box. The privileges that display are:

- [ALL](#)
- [CREATE](#)
- [DROP](#)
- [GRANT OPTION](#)
- [REFERENCES](#)
- [ALTER](#)
- [DELETE](#)
- [INDEX](#)
- [INSERT](#)
- [SELECT](#)
- [UPDATE](#)
- [TRIGGER](#)

You can choose to assign all privileges to a specific user or any other privilege listed in the preceding. Privileges irrelevant to a specific table, the [FILE](#) privilege for example, are not shown.

If a role has already been granted privileges on a specific table, those privileges show as already checked in the **ASSIGNED PRIVILEGES** list box.

7.5.2. Creating Foreign Key Relationships

Foreign key constraints are supported for the [InnoDB](#) storage engine only. For other storage engines the foreign key syntax is correctly parsed but not implemented. For more information see [Foreign Keys](#).

Using MySQL Workbench you may add a foreign key from within the table editor or by using the relationship tools on the vertical toolbar of an EER Diagram. This section deals with adding a foreign key using the foreign key tools. To add a foreign key using the table editor see [Section 7.5.1.3.5, “The Foreign Keys Tab”](#).

Using the graphical tools to add foreign keys is most effective when you are building tables from the ground up. If you have imported a database using an SQL script and do not need to add fields to your tables you may find it more effective to define foreign keys using the table editor.

7.5.2.1. Adding Foreign Key Relationships Using an EER Diagram

There are five foreign key tools on the vertical toolbar on the left side of an EER Diagram. These tools are:

- The [one-to-many non-identifying relationship](#) tool
- The [one-to-one non-identifying relationship](#) tool
- The [one-to-many identifying relationship](#) tool
- The [one-to-one identifying relationship](#) tool
- The [many-to-many relationship](#) tool

An identifying relationship is one where the child table cannot be uniquely identified without its parent. Typically this occurs where an intermediary table is created to resolve a many-to-many relationship. In such cases, the primary key is usually a composite key made up of the primary keys from the two original tables. An identifying relationship is indicated by a solid line between the tables and a nonidentifying relationship is indicated by a broken line.

Create or drag and drop the tables that you wish to connect. Ensure that there is a primary key in the table that will be on the “one” side of the relationship. Click on the appropriate tool for the type of relationship you wish to create. If you are creating a one-to-many relationship first click on the table that is on the “many” side of the relationship and then on the table containing the referenced key.

Doing this creates a field in the table on the many side of the relationship. The default name of this field is `table_name_key_name` where the table name and the key name are both derived from the table containing the referenced key.

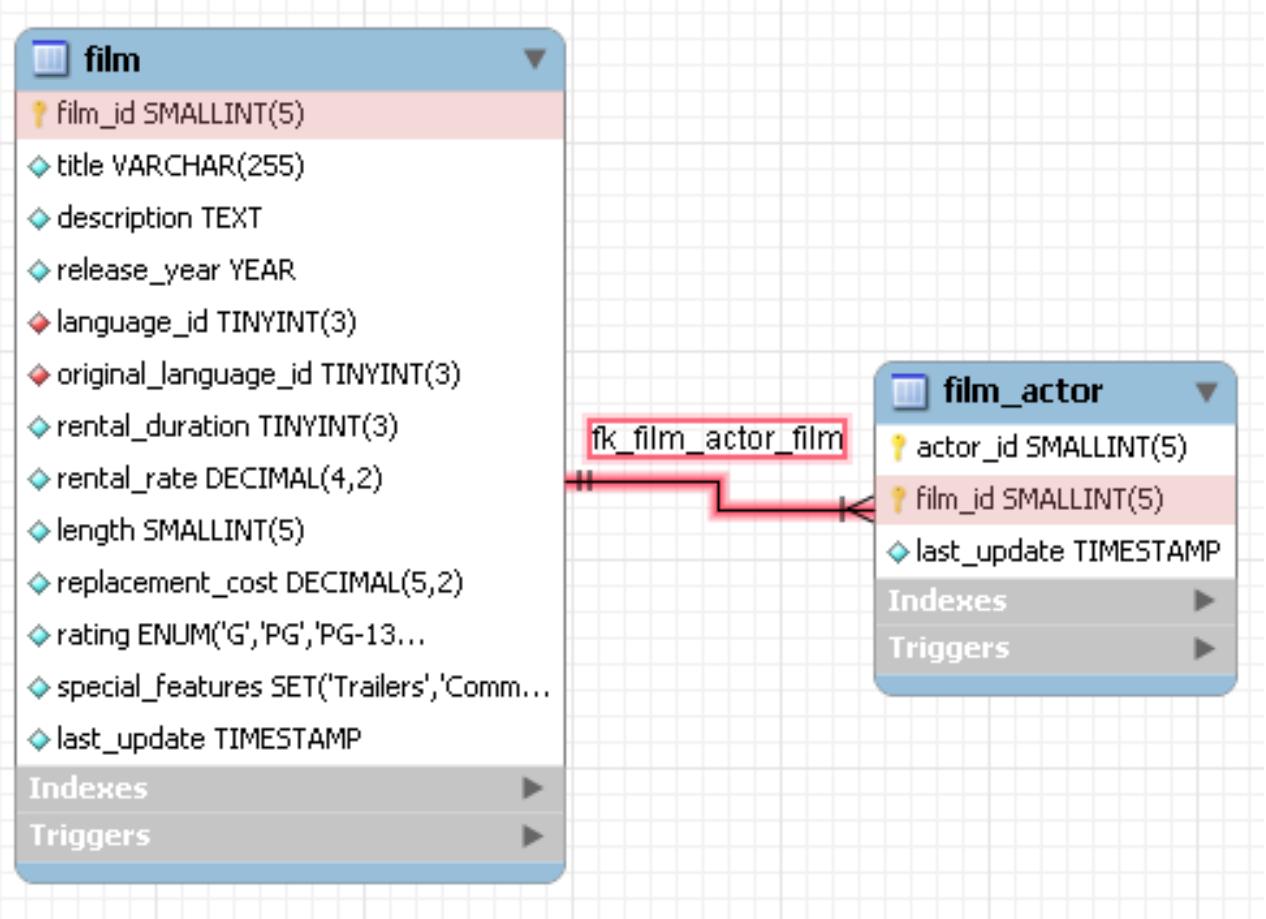
When the many-to-many tool is active, double clicking a table creates an associative table with a many-to-many relationship. For this tool to function there must be a primary key defined in the initial table.

Use the `MODEL`, `MENU OPTIONS` menu item to set a project-specific default name for the foreign key column (see [Section 7.1.5.4, “The Relationship Notation Menu Option”](#)). To change the global default see [Section 5.4.2, “The Model Tab”](#).

To edit the properties of a foreign key double click anywhere on the connection line that joins the two tables. Doing this opens the relationship editor.

Mousing over a relationship connector highlights the connector and the related keys as shown in the following figure.

Figure 7.11. The Relationship Connector



The `film` and the `film_actor` tables are related on the `film_id` field and these fields are highlighted in both tables. Since the `film_id` field is part of the primary key in the `film_actor` table, a solid line is used for the connector between the two tables.

If the placement of a connection's caption is not suitable, you can change its position by dragging it to a different location. If you have set a secondary caption, its position can also be changed. (For more information about secondary captions see [Section 7.5.2.3, “The Properties of a Connection”](#). Where the notation style allows, `Classic` for instance, the cardinality indicators can also be re-positioned.

The relationship notation style in [Figure 7.11, “The Relationship Connector”](#) is the default, crow's foot. If you are using a commercial version of MySQL Workbench you can change this. For more information, see [Section 7.1.5.4, “The Relationship Notation Menu Option”](#).

You can select multiple connections by holding down the **Ctrl** key as you click on a connection. This can be useful for highlighting specific relationships on an EER diagram.

7.5.2.2. The Relationship Editor

Double-clicking a relationship on the EER diagram canvas opens up the relationship editor. This has two tabs: **RELATIONSHIP**, and **FOREIGN KEY**.

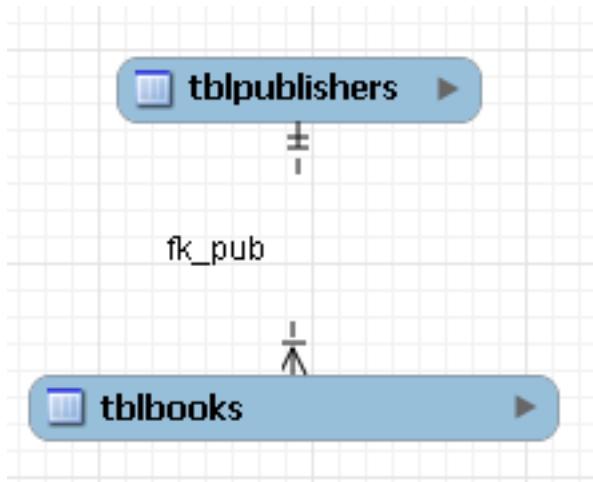
The Relationship tab

In the **RELATIONSHIP** tab you can set the caption of a relationship using the **CAPTION** text box. This name displays on the canvas and is also the name used for the constraint itself. The default value for this name is `fk_source_table_destination_table`. Use the **MODEL**, **MENU OPTIONS** menu item to set a project-specific default name for foreign keys. To change the global default see [Section 5.4.2, “The Model Tab”](#).

You can also add a secondary caption to a relationship and also a comment.

The **VISIBILITY SETTINGS** frame is used to determine how the relationship is displayed on the EER Diagram canvas. **Fully Visible** is the default but you can also choose to hide relationship lines or to use split lines. The split line style is pictured in the following:

Figure 7.12. The Split Connector



Note

A broken line connector is used to indicate a nonidentifying relationship. The split line style can be used with either an identifying relationship or a nonidentifying relationship. It is used for display purposes only and does not indicate anything about the nature of a relationship.

To set the notation of a relationship go to the **MODEL**, **RELATIONSHIP NOTATION** menu item. For more information, see [Section 7.1.5.4, “The Relationship Notation Menu Option”](#).

The Foreign Key tab

The **FOREIGN KEY** tab contains two frames: **REFERENCING TABLE** and **REFERENCED TABLE**.

The **MANDATORY** checkboxes are used to select whether the referencing table and the referenced table are mandatory. The default value for both of these constraints is `true`, which is indicated by the checkboxes being checked.

Between the frames there are a set of radio buttons that allow you to choose whether the relationship is one-to-one or one-to-many.

7.5.2.3. The Properties of a Connection

To select a connection, right click it. When a connection is selected it is highlighted and its properties are displayed in the properties palette. The properties of a connection are quite different from the properties of other objects. These properties are described in the following list:

- **caption** – The name of the object. By default this property is centered above the connection line. Its default value is the name

of the foreign key.

- `captionXOffs` – The “x” offset of the caption.
- `captionYOffs` – The “y” offset of the caption.
- `comment` – The comment associated with the relationship.
- `drawSplit` – Whether or not to show the relationship as a continuous line.
- `endCaptionXOffs` – The “x” termination point of the caption offset.
- `endCaptionYOffs` – The “y” termination point of the caption offset.
- `extraCaption` – A secondary caption. The default location for this extra caption is centered beneath the connection line.
- `extraCaptionXOffs` – The “x” offset of the secondary caption.
- `extraCaptionYOffs` – The “y” offset of the secondary caption.
- `mandatory` – Whether or not the entities are mandatory. For more information, see [Section 7.5.2.2, “The Relationship Editor”](#).
- `many` – False if the relationship is a one-to-one relationship.
- `middleSegmentOffset` – The offset of the middle section of the connector.
- `name` – The name used to identify the connection on the EER Diagram canvas. Note that this is **not** the name of the foreign key.
- `referredMandatory` – Whether or not the referred entity is mandatory
- `startCaptionXOffs` – The start of the “x” offset of the caption.
- `startCaptionYOffs` – The start of the “y” offset of the caption.

In most cases you can change the properties of a relationship using the relationship editor rather than the `Properties` palette.

If you make a relationship invisible by hiding it using the relationship editor's **VISIBILITY SETTINGS**, and then the relationship editor is closed, you will no longer be able to select the relationship in order to bring up its relationship editor. To make the relationship visible again you will need to expand the table object relating to the relationship in the **LAYERS** window and select the relationship object. Once selected, you can change the relationship's `visible` property to `True` in the corresponding **PROPERTIES** window. The relationship will then be visible in the **EER DIAGRAM** window.

7.5.3. Creating Views

You can add views to a database either from the `Physical Schemata` section of the `MySQL Model` page or from the EER Diagram.

7.5.3.1. Adding Views to the Physical Schemata

Double clicking the `Add View` icon in the `Physical Schemata` section of the `MySQL Model` page adds a view with the default name of `view1`. If a view with this name already exists, the new view is named `view2`.

Adding a new view automatically opens the view editor docked at the bottom of the application. Using the view editor is described in [Section 7.5.3.3, “The View Editor”](#).

Right clicking a table opens a pop-up menu with the following options:

- `CUT 'VIEW NAME'`
- `COPY 'VIEW NAME'`
- `EDIT VIEW...`
- `EDIT IN NEW WINDOW`
- `DELETE 'VIEW NAME'`

If the table editor is not open the EDIT VIEW ... option opens it. If it is already open, the selected table replaces the previous one. EDIT IN NEW WINDOW opens a new view editor tab.

The cut and copy options are useful for copying views between different schemata and COPY SQL TO CLIPBOARD copies the CREATE VIEW statement to the clipboard.

Warning

Use the DELETE 'VIEW NAME' to remove a view from the database. There will be **no** confirmation dialog box.

Any views added to the Physical Schemata also show up in the Catalog palette on the right side of the application. They may be added to an EER Diagram by dragging and dropping them from this palette.

7.5.3.2. Adding Views to an EER Diagram

Views can also be added to an EER Diagram using the View tool on the vertical toolbar. To do this make sure that the EER Diagram tab is selected, and right click the view icon on the vertical toolbar. The view icon is the two overlapping rectangles found below the table icon.

Clicking the mouse on this icon changes the mouse pointer to a view pointer. You can also change the mouse pointer to a view pointer by pressing the **V** key.

Choosing the View tool changes the contents of the toolbar that appears immediately below the menu bar. When the Views pointer is active, this toolbar contains a drop down list box of schemata and a drop down color chart. Use these list boxes to select the appropriate schema and color accent for the new view. Make sure that you associate the new view with a database. The color of your view can easily be changed later using the Properties palette.

Create a view by clicking anywhere on the EER Diagram canvas. This creates a new view with the default name view1. To revert to the default mouse pointer, click the arrow icon at the top of the vertical toolbar.

Right clicking a view opens a pop-up menu. With the exception of the delete option, these menu options function as described in [Section 7.5.3.1, “Adding Views to the Physical Schemata”](#). The behavior of the delete option is determined by your MySQL Workbench options settings. For more information, see [Section 5.4.2, “The Model Tab”](#).

7.5.3.3. The View Editor

You can invoke the view editor by double clicking a view object on the EER Diagram canvas or by double clicking a view in the Physical Schemata section on the MySQL Model page. Doing this opens the view editor docked at the bottom of the application. Double clicking the title bar undocks the editor. Do the same to redock it. Any number of views may be open at the same time. Each additional view appears as a tab at the top of the view editor,

There are two tabs at the bottom of the view editor, the **VIEW** and the **PRIVILEGES** tabs. Navigate between different tabs using the mouse or from the keyboard by pressing **Ctrl + Alt + Tab**.

7.5.3.3.1. The View Tab

From the View tab of the view editor you can perform the following tasks:

- Rename the view using the **NAME** text box.
- Enter the SQL to create a view using the **SQL** text area.
- Comment a view using the **COMMENTS** text area.

7.5.3.3.2. The Privileges Tab

The Privileges tab of the view editor functions in exactly the same way as the Privileges tab of the table editor. For more information, see [Section 7.5.1.3.10, “The Privileges Tab”](#).

7.5.3.3.3. Modifying a View using the Properties Palette

When you select a view on the EER Diagram canvas, its properties are displayed in the Properties palette. Most of the properties accessible from the Properties palette apply to the appearance of a view on the EER Diagram canvas.

For a list of the properties accessible through the Properties palette see [Section 7.3.7.1, “The Properties Palette”](#).

7.5.4. Creating Routines and Routine Groups

You can add Routine Groups to a database either from the **PHYSICAL SCHEMATA** section of the **MySQL Model** page or from an EER Diagram. Routines may only be added from the **PHYSICAL SCHEMATA** section of the **MySQL Model** page.

To view an existing schema, along with its Routines and Routine Groups, select **DATABASE**, **REVERSE ENGINEER...** from the main menu. After the schema has been added to the current model, you can see the schema objects on the **PHYSICAL SCHEMATA** panel on the **MySQL Model** page. The Routines and Routine Groups are listed there.

MySQL Workbench unifies both stored procedures and stored functions into one logical object called a Routine. Routine Groups are used to group routines that are related. You can decide how many Routine Groups you want to create and you can use the **ROUTINE GROUP EDITOR** to assign specific routines to a group, using a drag and drop interface.

When designing an EER Diagram you can place the Routine Groups on the canvas by dragging them from the **CATALOG PALETTE**. Placing individual routines on the diagram is not permitted, as it would clutter the canvas.

7.5.4.1. Routines

7.5.4.1.1. Adding Routines to the Physical Schemata

Double clicking the **Add Routine** icon in the **Physical Schemata** section of the **MySQL Model** page adds a routine with the default name of **routine1**. If a routine with this name already exists, the new routine is named **routine2**.

Adding a new routine automatically opens the routine editor docked at the bottom of the application. Using the routine editor is described in [Section 7.5.4.1.2, “The Routine Editor”](#).

Right clicking a routine opens a pop-up menu with the following options:

- [RENAME](#)
- [CUT '*ROUTINE_NAME*'](#)
- [COPY '*ROUTINE_NAME*'](#)
- [EDIT ROUTINE...](#)
- [EDIT IN NEW WINDOW](#)
- [COPY SQL TO CLIPBOARD](#)
- [DELETE '*ROUTINE_NAME*'](#)

The [EDIT ROUTINE ...](#) option opens the routine editor.

The cut and paste options are useful for copying routines between different schemata.

Note

Deleting the code for routine from the **ROUTINES** tab of the Routine Group Editor will result in removal of the routine object from the model.

Note

To remove a routine from a routine group use the controls on the **ROUTINE GROUP** tab of the Routine Group Editor.

The action of the delete option varies depending upon the way you have configured MySQL Workbench. For more information, see [Section 5.4.2, “The Model Tab”](#).

7.5.4.1.2. The Routine Editor

You can invoke the routine editor by double clicking a routine in the **Physical Schemata** section on the **MySQL Model** page. Doing this opens the routine editor docked at the bottom of the application. Double clicking the routine tab undocks the editor. Double click the title bar to redock it. Any number of routines may be open at the same time. Each additional routine appears as a tab at the top of the routine editor,

There are two tabs at the bottom of the routine editor, the **VIEW** and the **PRIVILEGES** tabs. Navigate between different tabs using the mouse or from the keyboard by pressing **Ctrl + Alt + Tab**.

7.5.4.1.2.1. The **Routine** Tab

From the [Routine](#) tab of the routine editor you can perform the following tasks:

- Rename the routine using the **NAME** text box.
- Enter the SQL to create a routine using the **SQL** text area.

7.5.4.1.2.2. The [Privileges](#) Tab

The [Privileges](#) tab of the routine editor functions in exactly the same way as the [Privileges](#) tab of the table editor. For more information, see [Section 7.5.1.3.10, “The Privileges Tab”](#).

Note

Privileges are only available in the Standard Edition of MySQL Workbench.

7.5.4.2. Routine Groups

7.5.4.2.1. Adding Routine Groups to the Physical Schemata

Double clicking the [Add Routine Group](#) icon in the [Physical Schemata](#) section of the [MySQL Model](#) page adds a routine with the default name of `routines1`. If a routine group with this name already exists, the new routine group is named `routines2`.

Adding a new routine group automatically opens the routine groups editor docked at the bottom of the application. Using the routine groups editor is described in [Section 7.5.4.2.3, “The Routine Group Editor”](#).

Right clicking a routine group opens a pop-up menu with the following options:

- [RENAME](#)
- [CUT '`ROUTINE GROUP NAME`'](#)
- [COPY '`ROUTINE GROUP NAME`'](#)
- [EDIT ROUTINE...](#)
- [EDIT IN NEW WINDOW](#)
- [COPY SQL TO CLIPBOARD](#)
- [DELETE '`ROUTINE GROUP NAME`'](#)

The [EDIT ROUTINE GROUP...](#) option opens the routine group editor. Using the routine group editor is described in [Section 7.5.4.2.3, “The Routine Group Editor”](#).

The cut and paste options are useful for copying routine groups between different schemata.

Deleting a routine group from the [MySQL Model](#) page removes the group but does not remove any routines contained in that group.

Any routine groups added to the [Physical Schemata](#) also show up in the [Catalog](#) palette on the right side of the application. They may be added to an EER Diagram by dragging and dropping them from this palette.

7.5.4.2.2. Adding Routine Groups to an EER Diagram

Routine groups can also be added to an EER Diagram using the [Routine Groups](#) tool on the vertical toolbar. To do this make sure that the [EER Diagram](#) tab is selected, and right click the routine groups icon on the vertical toolbar. The routine groups icon is immediately above the lowest toolbar separator.

Clicking the mouse on this icon changes the mouse pointer to a routine group pointer. You can also change the mouse pointer to a routine pointer by pressing the **G** key.

Choosing the [Routine Group](#) tool changes the contents of the toolbar that appears immediately below the menu bar. When the [Routine Groups](#) pointer is active, this toolbar contains a drop down list box of schemata and a drop down color chart. Use these list boxes to select the appropriate schema and color accent for the new routine group. Make sure that you associate the new routine group with a database. The color of your routine group can easily be changed later using the [Properties](#) palette.

Create a routine group by clicking anywhere on the EER Diagram canvas. This creates a new routine group with the default name `routines1`. To revert to the default mouse pointer, click the arrow icon at the top of the vertical toolbar.

Right clicking a routine group opens a pop-up menu. With the exception of the delete option and rename options these menu options function as described in [Section 7.5.4.2.1, “Adding Routine Groups to the Physical Schemata”](#). There is no rename option and the behavior of the delete option is determined by your MySQL Workbench options settings. For more information, see [Section 5.4.2, “The Model Tab”](#).

7.5.4.2.3. The Routine Group Editor

You can invoke the routine group editor by double clicking a routine group object on the EER Diagram canvas or by double clicking a routine group in the [Physical Schemata](#) section on the [MySQL Model](#) page. Doing this opens the routine group editor docked at the bottom of the application. Double clicking the title bar undocks the editor. Do the same to redock it. Any number of routine groups may be open at the same time. Each additional routine group appears as a tab at the top of the routine editor,

There are two tabs at the bottom of the routine editor, the **ROUTINE GROUP** and the **PRIVILEGES** tabs. Navigate between different tabs using the mouse or from the keyboard by pressing **Ctrl + Alt + Tab**.

7.5.4.2.3.1. The [Routine Groups](#) Tab

From the [Routine Groups](#) tab of the routine groups editor you can perform the following tasks:

- Rename the routine group using the **NAME** text box.
- Add routines to the group by dragging and dropping them.
- Add comments to the routine group.

7.5.4.2.3.2. The [Privileges](#) Tab

The [Privileges](#) tab of the routine group editor functions in exactly the same way as the [Privileges](#) tab of the table editor. For more information, see [Section 7.5.1.3.10, “The Privileges Tab”](#).

Note

Privileges are only available in the Standard Edition of MySQL Workbench.

7.5.4.2.3.3. Modifying a Routine Group Using the Properties Palette

When you select a routine group on the EER Diagram canvas, its properties are displayed in the [Properties](#) palette. All of the properties accessible from the [Properties](#) palette apply to the appearance of a routine group on the EER Diagram canvas.

For a list of the properties accessible through the [Properties](#) palette see [Section 7.3.7.1, “The Properties Palette”](#).

7.5.5. Creating Layers

You can add layers to a database only from an EER Diagram. Layers are used to help organize objects on the canvas. Typically, related objects are added to the same layer; for example, you may choose to add all your views to one layer.

7.5.5.1. Adding Layers to an EER Diagram

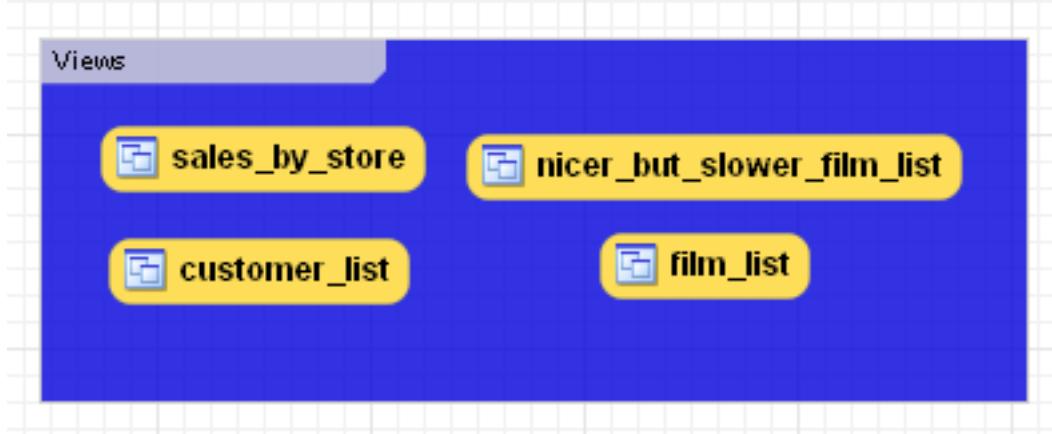
Layers are added to an EER Diagram using the [Layer](#) tool on the vertical toolbar. To do this select an [EER Diagram](#) tab and right click the layer icon on the vertical toolbar. The layer icon is the rectangle with an ‘L’ in the lower left corner and it is found below the eraser icon.

Clicking the mouse on this icon changes the mouse pointer to a layer pointer. You can also change the mouse pointer to a layer pointer by pressing the **L** key.

Choosing the [Layer](#) tool changes the contents of the toolbar that appears immediately below the menu bar. When the [Layers](#) pointer is active, this toolbar contains a drop down color chart. Use this list box to select the color accent for the new layer. The color of your layer can easily be changed later using the [Properties](#) palette.

Create a layer by clicking anywhere on the EER Diagram canvas and, holding the left mouse button down, draw a rectangle of a suitable size. This creates a new layer with the default name `layer1`. To revert to the default mouse pointer, click the arrow icon at the top of the vertical toolbar.

Find below an image of a layer containing a number of views:

Figure 7.13. The Layer Object

Use the `name` property of the [Properties](#) palette to change the name of a layer.

Right clicking a layer opens a pop-up menu with the following options:

- [CUT 'LAYER_NAME'](#)
- [COPY 'LAYER_NAME'](#)
- [DELETE 'LAYER_NAME'](#)

The cut and copy options are useful for copying layers between different schemata.

Since layers are not schema objects, no confirmation dialog box opens when you delete a layer regardless of how you have configured MySQL Workbench. Deleting a layer does **not** delete schema objects from the catalog.

7.5.5.1.1. Adding Objects to a Layer

Add an object to a layer by dragging and dropping it directly from the [Catalog](#) palette onto a layer. If you pick up an object from an EER diagram you need to press **Ctrl** as you drag it on to the layer, otherwise it will not be "locked" inside the layer.

Locking objects to a layer prevents their accidental removal. You cannot remove them simply by clicking and dragging; in order to remove an object, you also need to press the **Ctrl** key while dragging it.

As a visual cue that the object is being "locked", the outline of the layer is highlighted as the object is dragged over it.

If you drag a layer over a table object, the table object will automatically be added to the layer. This also works for multiple table objects.

Note that layers cannot be nested, that is, a layer cannot contain another layer object.

7.5.5.2. Modifying a Layer using the Properties Palette

When you select a layer on the EER Diagram canvas, its properties are displayed in the [Properties](#) palette. The properties accessible from the [Properties](#) palette apply to the appearance of a layer on the EER Diagram canvas.

In some circumstances you may want to make a layer invisible. To do this, select the layer and, in the [Properties](#) palette, set the `visible` property to `False`. To locate an invisible object, open the [Layers](#) palette and select the object by double clicking it. Once an object is selected you can reset the `visible` property from the [Properties](#) palette.

For a list of the properties accessible through the [Properties](#) palette see [Section 7.3.7.1, “The Properties Palette”](#). In addition to the properties listed there, a layer also has a `description` property. Use this property to document the purpose of the layer.

7.5.6. Creating Notes

You can add notes to a database only from the [Model Notes](#) section of the [MySQL Model](#) page. Notes are typically used to help document the design process.

7.5.6.1. Adding Notes

Double clicking the [Add Note](#) icon in the [Model Notes](#) section of the [MySQL Model](#) page adds a note with the default name of `note1`. If a note with this name already exists, the new note is named `note2`.

Adding a new note automatically opens the note editor docked at the bottom of the application. Using the note editor is described in [Section 7.5.6.2, “The Note Editor”](#).

Right clicking a note opens a pop-up menu with the following options:

- [RENAME](#)
- [CUT '`NOTE_NAME`'](#)
- [COPY '`NOTE_NAME`'](#)
- [DELETE '`NOTE_NAME`'](#)

The [EDIT NOTE ...](#) option opens the note editor. Using the note editor is described in [Section 7.5.6.2, “The Note Editor”](#).

The cut and copy options are useful for copying notes between different schemata.

Notes can only be added on the [MySQL Model](#) page.

7.5.6.2. The Note Editor

You can invoke the note editor by double clicking a note object in the [Model Note](#) section on the [MySQL Model](#) page. Doing this opens the note editor docked at the bottom of the application. Double clicking the note tab undocks the editor. Double click the title bar to redock it. Any number of notes may be open at the same time. Each additional note appears as a tab at the top of the note editor.

Using the editor you can change the name of a note or its contents.

7.5.7. Creating Text Objects

Text objects are applicable to an EER diagram only. They can be used for documentation purposes, for example, to explain a grouping of schema objects. They are also useful for creating titles for an EER diagram should you decide to export a diagram as a PDF or PNG file.

7.5.7.1. Adding Text Objects to an EER Diagram

Text objects can be added to an EER Diagram using the [Text Object](#) tool on the vertical toolbar. To do this make sure that the [EER Diagram](#) tab is selected, and right click the text object icon on the vertical toolbar. The text object icon is the rectangular icon found below the label icon.

Clicking the mouse on this icon changes the mouse pointer to a text object pointer. You can also change the mouse pointer to a text object pointer by pressing the **N** key.

Choosing the [Text Object](#) tool changes the contents of the toolbar that appears immediately below the menu bar. When the [Text Object](#) pointer is active, this toolbar contains a drop down color chart. Use this list box to select the color accent for the new text object. The color of your text object can easily be changed later using the [Properties](#) palette.

Create a text object by clicking anywhere on the EER Diagram canvas. This creates a new text object with the default name `text1`. To revert to the default mouse pointer, click the arrow icon at the top of the vertical toolbar.

Right clicking a text object opens a pop-up menu. These menu options are identical to the options for other objects. However, since a text object is not a database object, there is no confirmation dialog box when you delete a text object.

7.5.7.2. The Text Object Editor

You can invoke the text object editor by double clicking a text object on the EER Diagram canvas. Doing this opens the editor docked at the bottom of the application. Double clicking the text object table undocks the editor. Double click the title bar to redock it. Any number of text objects may be open at the same time. Each additional text objects appears as a tab at the top of the text editor.

Using the editor you can change the name of a text object or its contents.

7.5.7.2.1. Modifying a Text Object Using the **Properties** Palette

When you select a text object on the EER Diagram canvas, its properties are displayed in the **Properties** palette. Most of the properties accessible from the **Properties** palette apply to the appearance of a view on the EER Diagram canvas.

For a list of the properties accessible through the **Properties** palette see [Section 7.3.7.1, “The Properties Palette”](#).

There is no property in the **Properties** palette for changing the font used by a text object. To change the font used by a text object choose the **Appearance** tab of the Workbench Preferences dialog. For more information, see [Section 5.4.5, “The Appearance Tab”](#).

7.5.8. Creating Images

Images only exist on the EER Diagram canvas; you can only add them from the EER Diagram window.

7.5.8.1. Adding Images to an EER Diagram

Images can be added to an EER Diagram using the **Image** tool on the vertical toolbar. To add an image make sure that the **EER Diagram** tab is selected, and right click the image icon on the vertical toolbar. The image icon is the icon just above the table icon.

Clicking the mouse on this icon changes the mouse pointer to an image pointer. You can also change the mouse pointer to an image pointer by pressing the **I** key.

Create a image by clicking anywhere on the EER Diagram canvas. This opens a file open dialog box. Select the desired image, and close the dialog box to create an image on the canvas. To revert to the default mouse pointer, click the arrow icon at the top of the vertical toolbar.

Right clicking this object opens a pop-up menu with the following options:

- [CUT 'IMAGE'](#)
- [COPY 'IMAGE'](#)
- [EDIT IMAGE ...](#)
- [EDIT IN NEW WINDOW ...](#)
- [DELETE 'IMAGE'](#)

These menu options function in exactly the same way as they do for other objects on an EER diagram. However, images are not database objects so there is no confirmation dialog box when they are deleted.

7.5.8.2. The Image Editor

You can invoke the image editor by double clicking a image object on an EER Diagram canvas. Doing this opens the image editor docked at the bottom of the application. Double clicking the image editor tab undocks the editor. Double click the title bar to redock it. Any number of images may be open at the same time. Each additional image appears as a tab at the top of the image editor,

7.5.8.2.1. The **Image** Tab

From the **Image** tab of the image editor you can perform the following tasks:

- Rename the image using the **NAME** text box.
- Browse for an image using the **BROWSE** button.

7.5.9. Reverse Engineering

Using MySQL Workbench you can reverse engineer a database using a MySQL create script or you can connect to a live MySQL server and import a single database or a number of databases. Reverse engineering using a MySQL DDL script applies to all versions of MySQL Workbench; reverse engineering a database directly from a MySQL server applies to commercial versions of MySQL Workbench only.

7.5.9.1. Reverse Engineering Using a Create Script

Reverse engineering using a create script is done by using the **FILE**, **IMPORT**, **REVERSE ENGINEER MySQL CREATE SCRIPT ...** menu options. Doing this opens a file open dialog box with the default file type set to an SQL script file, a file with the extension **.sql**.

You can create a data definition (DDL) script by executing the **mysqldump db_name --no-data > script_file.sql** command. Using the **--no-data** option ensures that the script contains DDL statements only. However, if you are working with a script that also contains DML statements you need not remove them; they will be ignored.

Note

If you plan to redesign a database within MySQL Workbench and then export the changes, be sure to retain a copy of the original DDL script. You will need the original script in order to create an **ALTER** script. For more information, see [Section 7.5.10.1.2, “Altering a Schema”](#).

Use the **--databases** option with **mysqldump** if you wish to create the database as well as all its objects. If there is no **CREATE DATABASE db_name** statement in your script file, you must import the database objects into an existing schema or, if there is no schema, a new unnamed schema is created.

If your script creates a database, a new physical schemata tab is created on the [MySQL Model](#) page.

Any database objects may be imported from a script file in this fashion; tables, views, routines, and routine groups. Any indexes, keys, and constraints are also imported. Objects imported using an SQL script can be manipulated within MySQL Workbench in the same way that any other objects can.

Before exiting, be sure to save the schema. Choose the **FILE**, **SAVE** menu item and the reverse-engineered database will be saved as a MySQL Workbench file with the extension **mwb**.

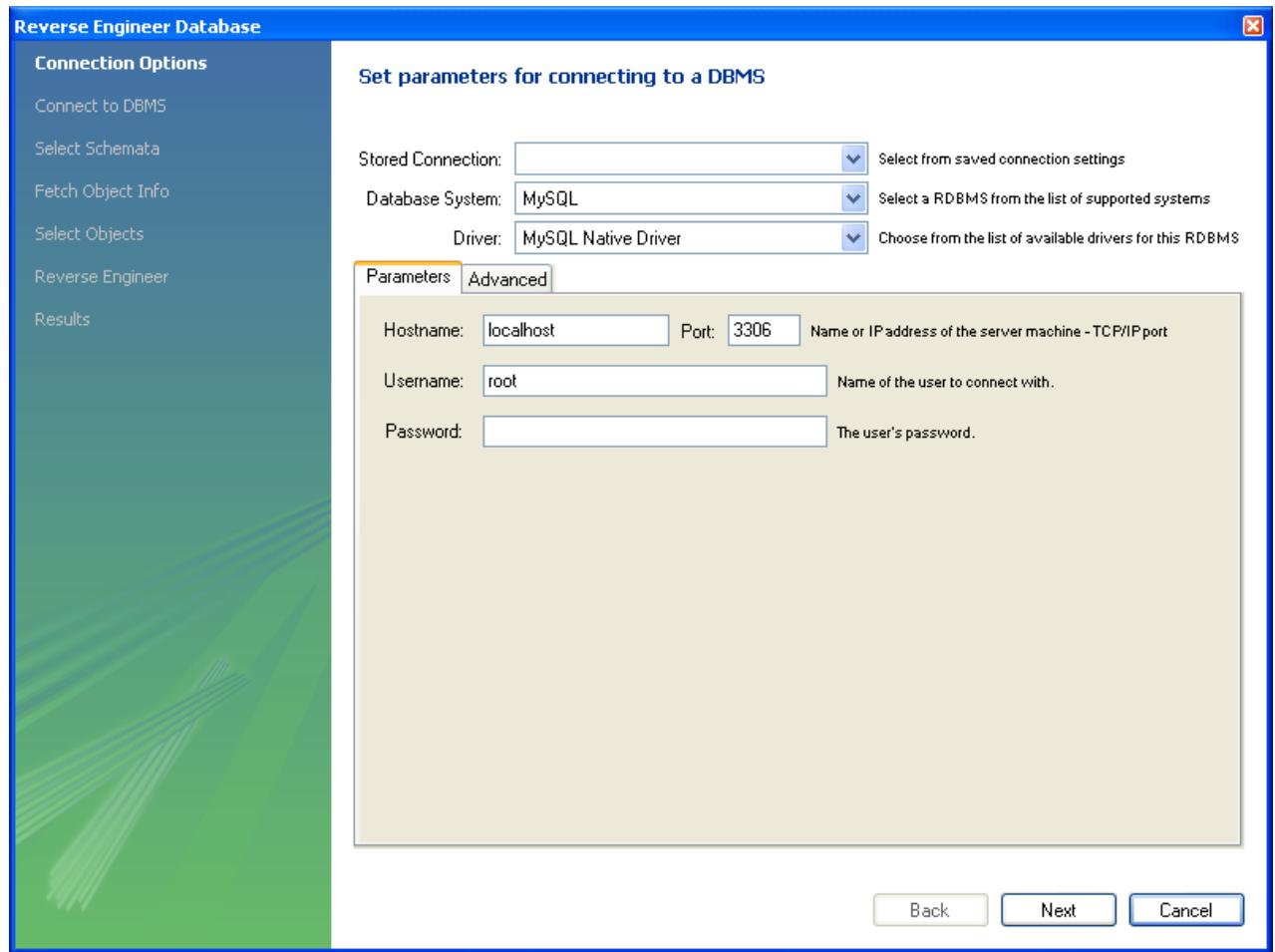
See [Section 7.6.1, “Importing a Data Definition SQL Script”](#) for a tutorial on reverse engineering the **sakila** database.

7.5.9.2. Reverse Engineering a Live Database

This section explains how to reverse engineer a live database using MySQL Workbench.

Select the **DATABASE**, **REVERSE ENGINEER ...** menu item from the main menu. Doing this opens the Reverse Engineer Database wizard.

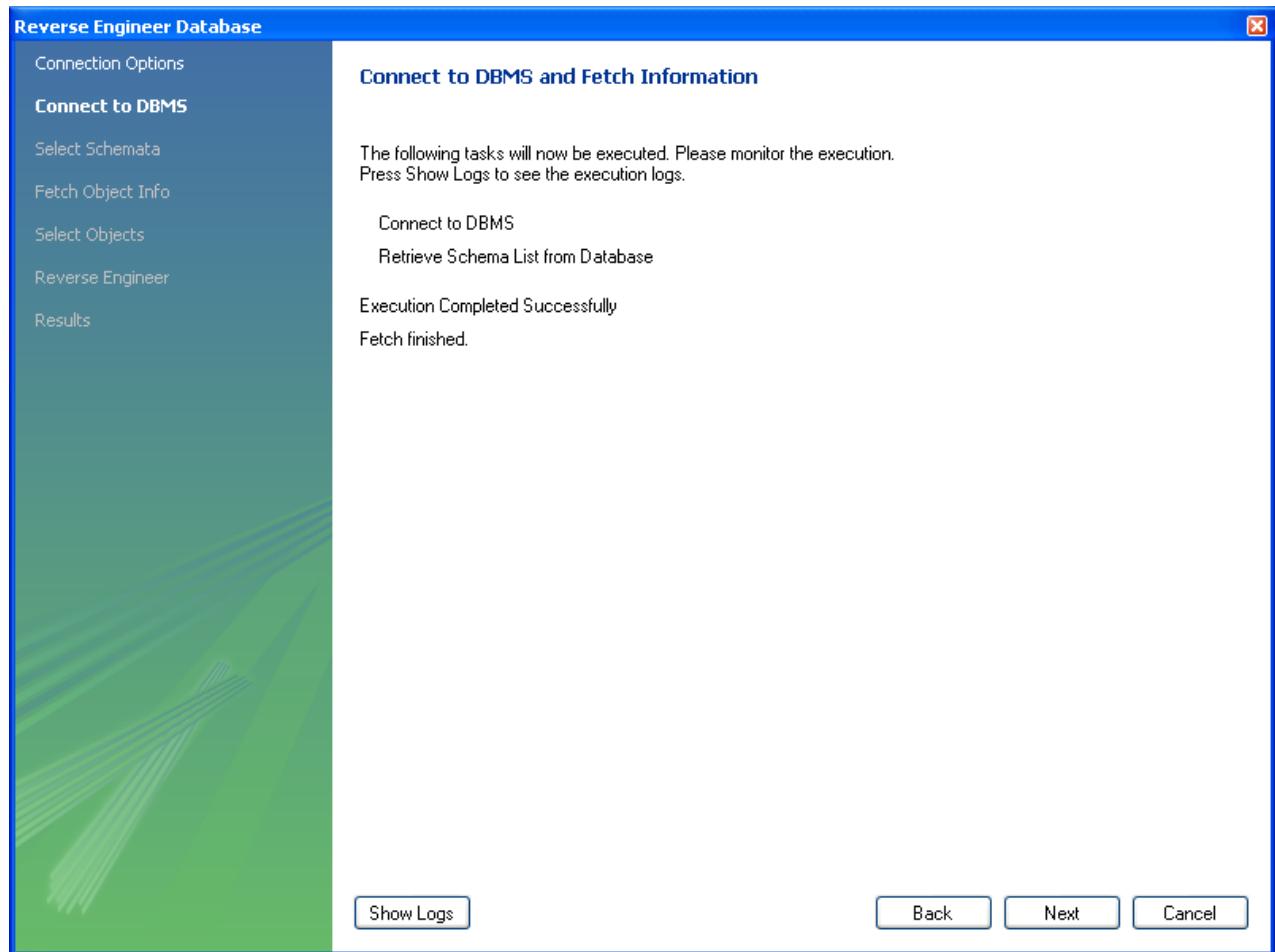
Figure 7.14. Reverse Engineer Database Wizard



The first page of the wizard allows you to set up a connection to the live database you wish to reverse engineer. This allows you to set up a new connection, or select a previously created connection. Typical information required for the connection includes host-name, username and password.

Once this information has been entered, or you have selected a stored connection, click the NEXT button to proceed to the next page:

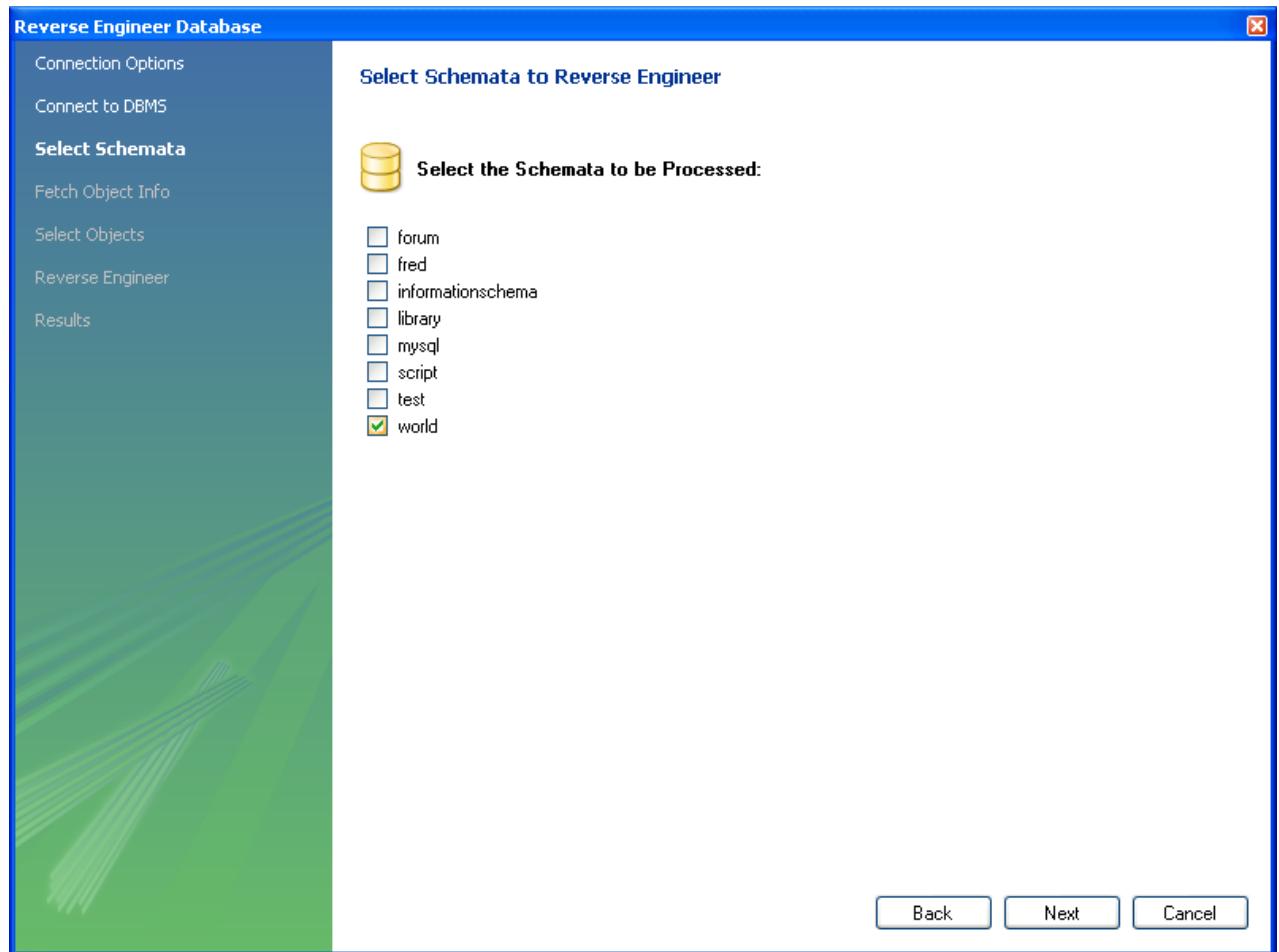
Figure 7.15. Connect to DBMS



Review the displayed information to make sure that the connection did not generate errors, then click NEXT.

On the next page you can select the database schema you want to connect to among those available on the server. Simply click the checkbox or checkboxes associated with the schema you wish to process:

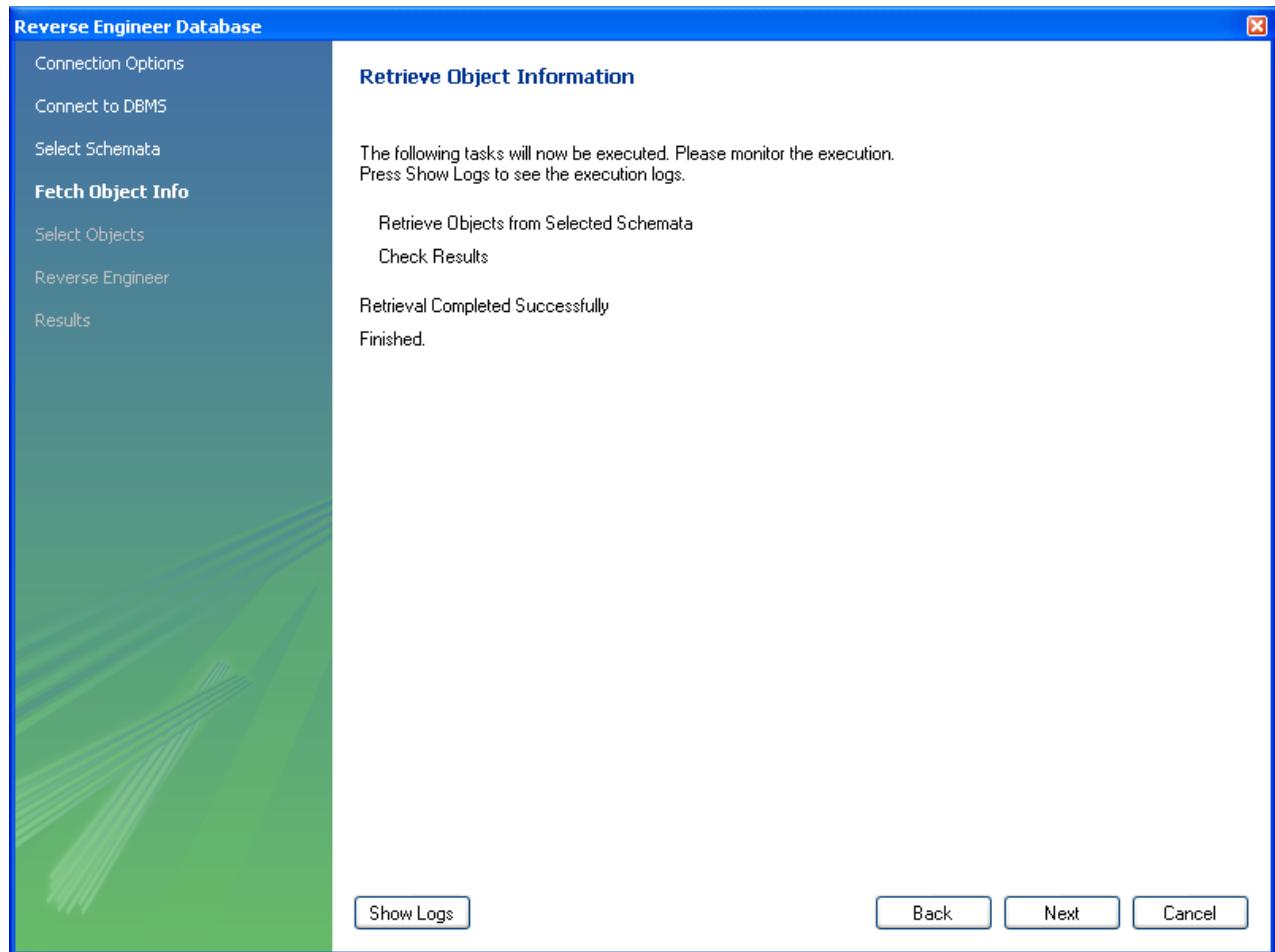
Figure 7.16. Select Schemata



Once you have selected the desired schema, click the NEXT button to continue.

The wizard then displays the tasks it carried out and summarizes the results of the operation:

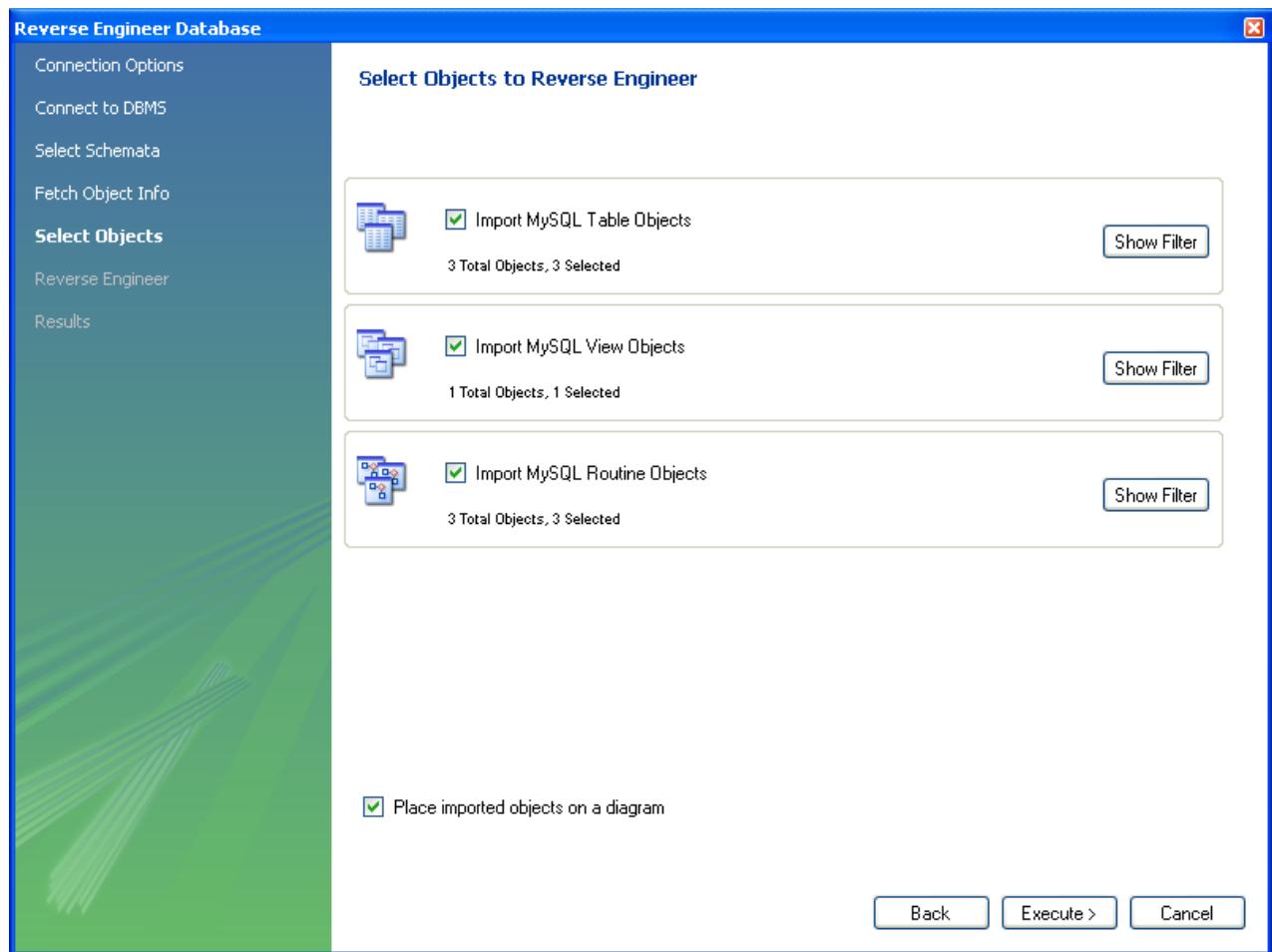
Figure 7.17. Fetch Object Info



Review the results before clicking NEXT to continue.

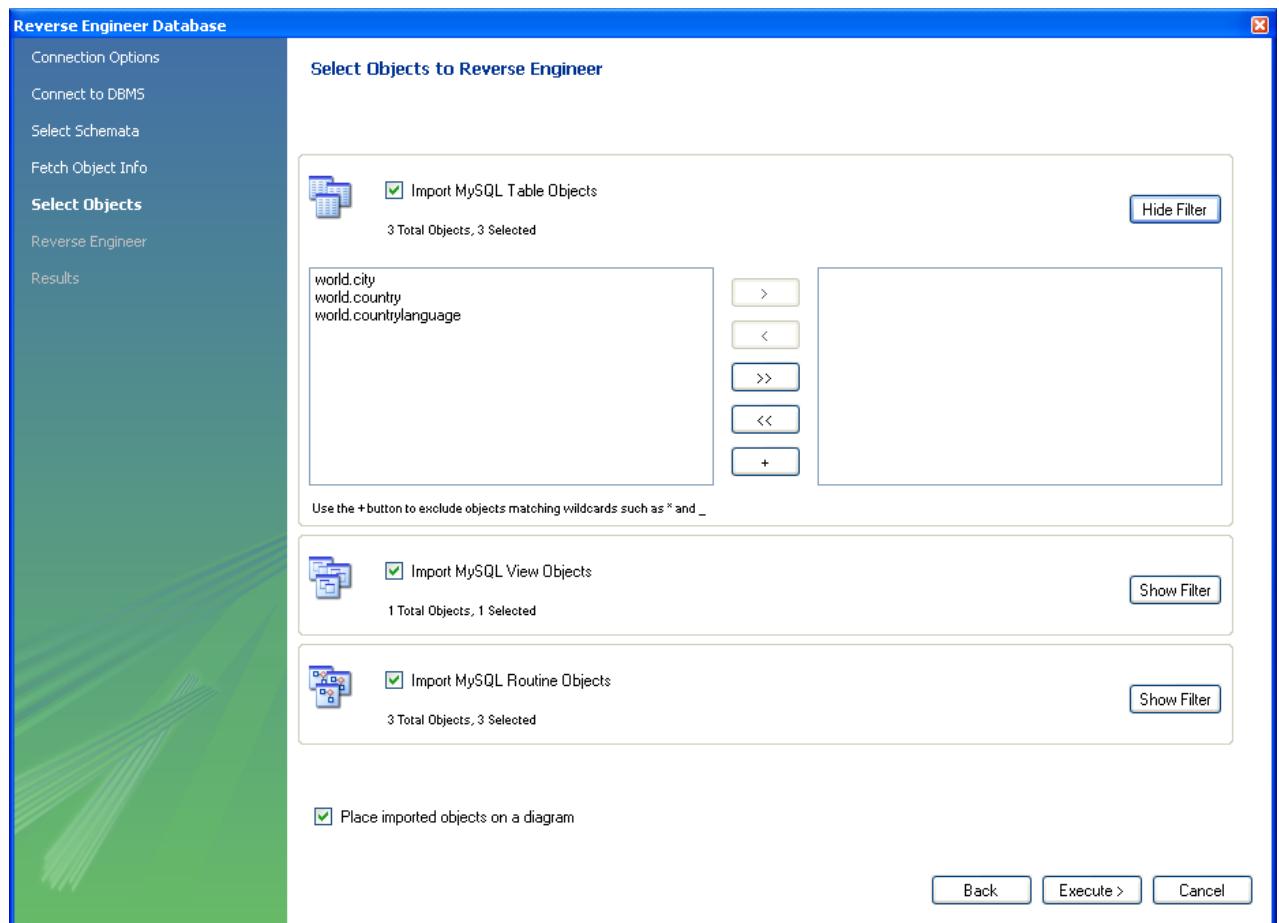
The next page is the [Select Objects](#) page. It is sectioned off by object type. This screen is of special interest if you do not wish to import all the objects from the existing database — this screen gives you the option of filtering which objects are imported. Each section has a SHOW FILTER button. Click this button if you do not want to import all the objects of a specific type.

Figure 7.18. Select Objects



For the **IMPORT MySQL TABLE OBJECTS** section, if you click on the SHOW FILTER button the following page is displayed:

Figure 7.19. Show Filter



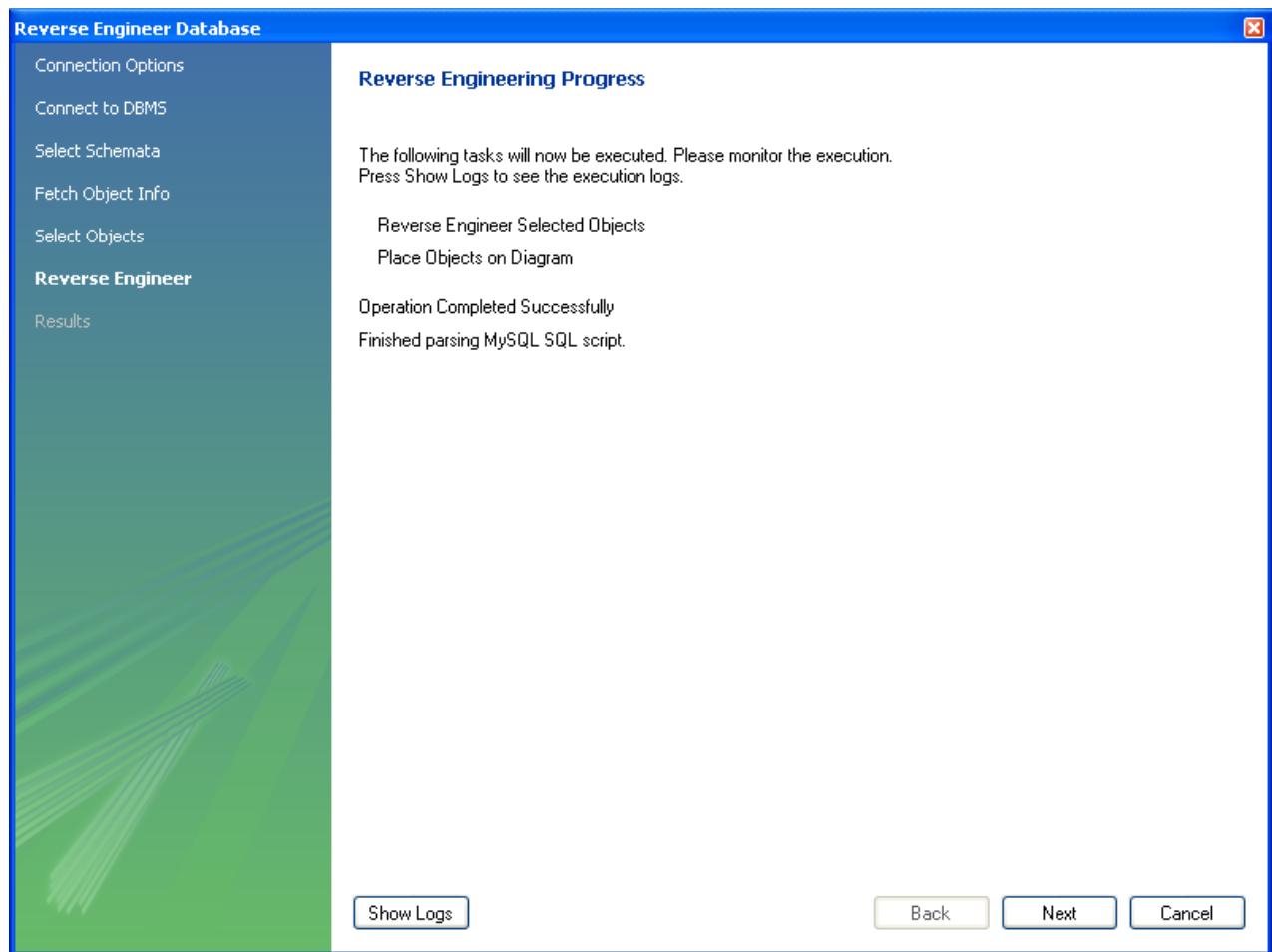
This allows you to select specific tables that you wish to import. Having selected the desired tables you can optionally hide the filter by clicking the HIDE FILTER button.

The other sections have similar filters available.

You can click NEXT to continue to the next page.

The wizard then displays the tasks that have been carried out and whether the operation was successful or not. If errors were generated then you can click on the SHOW LOGS button to see the nature of the errors.

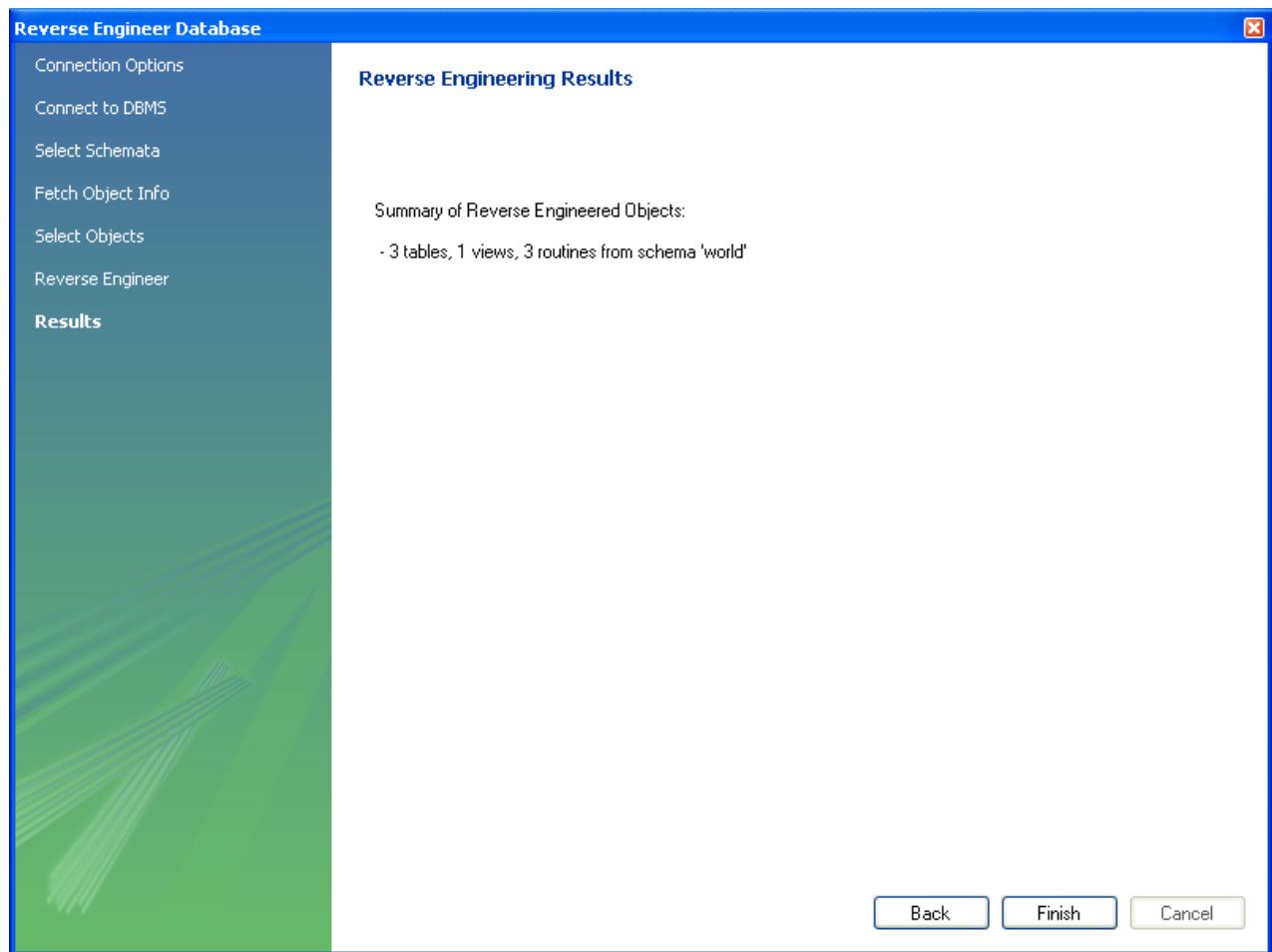
Figure 7.20. Progress



Click NEXT to continue to the next page.

The final screen of the wizard provides a summary of the reverse engineered objects:

Figure 7.21. Results



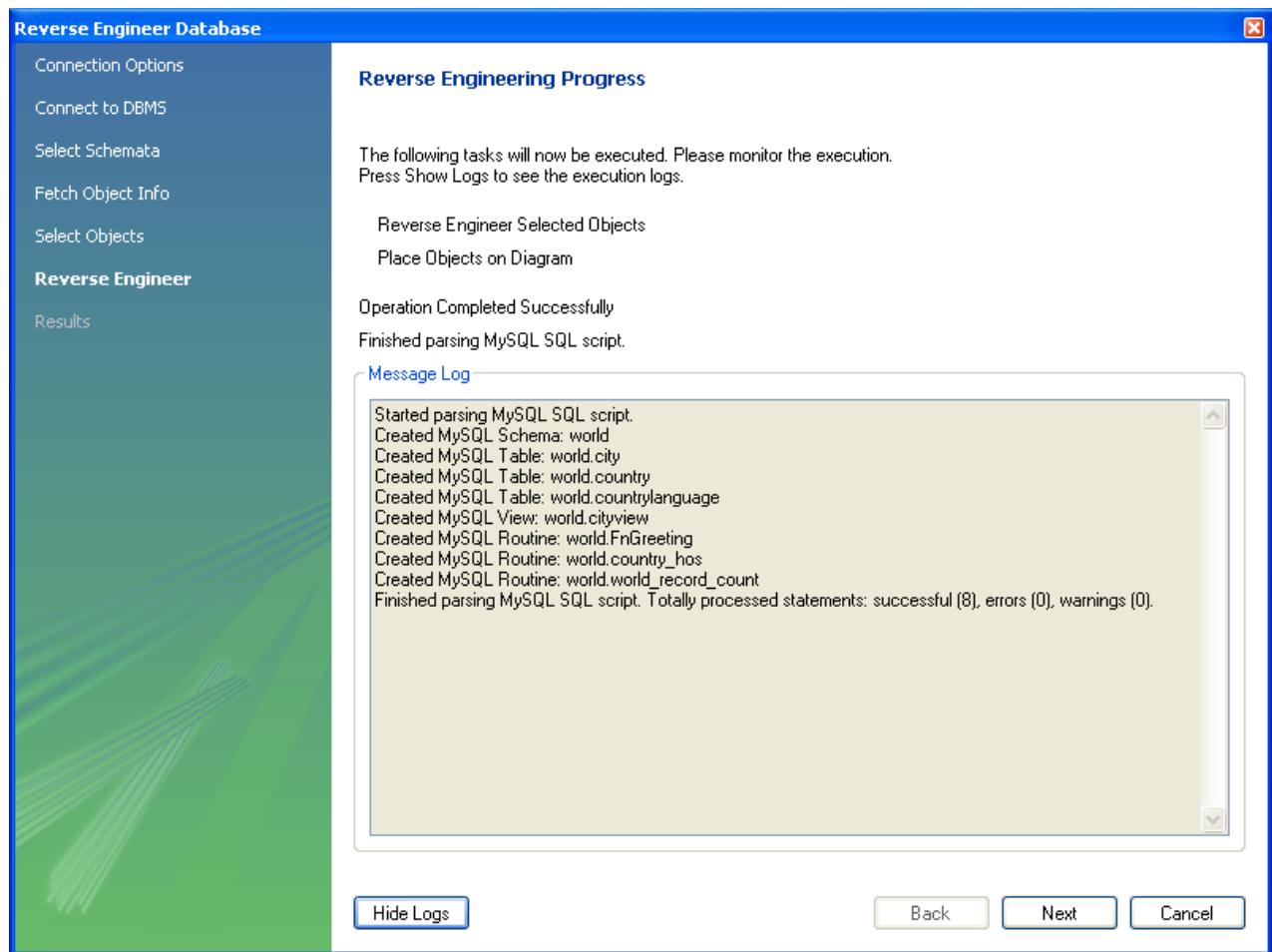
Click FINISH to exit the wizard.

Before leaving MySQL Workbench be sure to save the schema. Choose the FILE, SAVE menu item and the reverse-engineered database will be saved as a MySQL Workbench file with the extension mwb.

7.5.9.2.1. Errors During Reverse Engineering

During reverse engineering the application checks for tables and views that duplicate existing names and disallows duplicate names if necessary. If you attempt to import an object that duplicates the name of an existing object you will be notified with an error message. To see any errors that have occurred during reverse engineering you can click on the button SHOW LOGS. This will create a panel containing a list of messages, including any error messages than may have been generated. Click the HIDE LOGS button to close the panel.

Figure 7.22. Message Log



If you wish to import an object with the same name as an existing object, rename the existing object before reverse engineering.

If you import objects from more than one schema, there will be a tab in the [Physical Schemata](#) section of the [MySQL Model](#) page for each schema imported.

You cannot reverse engineer a live database that has the same name as an existing schema. If you wish to do this, first rename the existing schema.

7.5.10. Forward Engineering

It is possible to forward engineer a database using an SQL script or by connecting to a live database.

7.5.10.1. Forward Engineering Using SQL Scripts

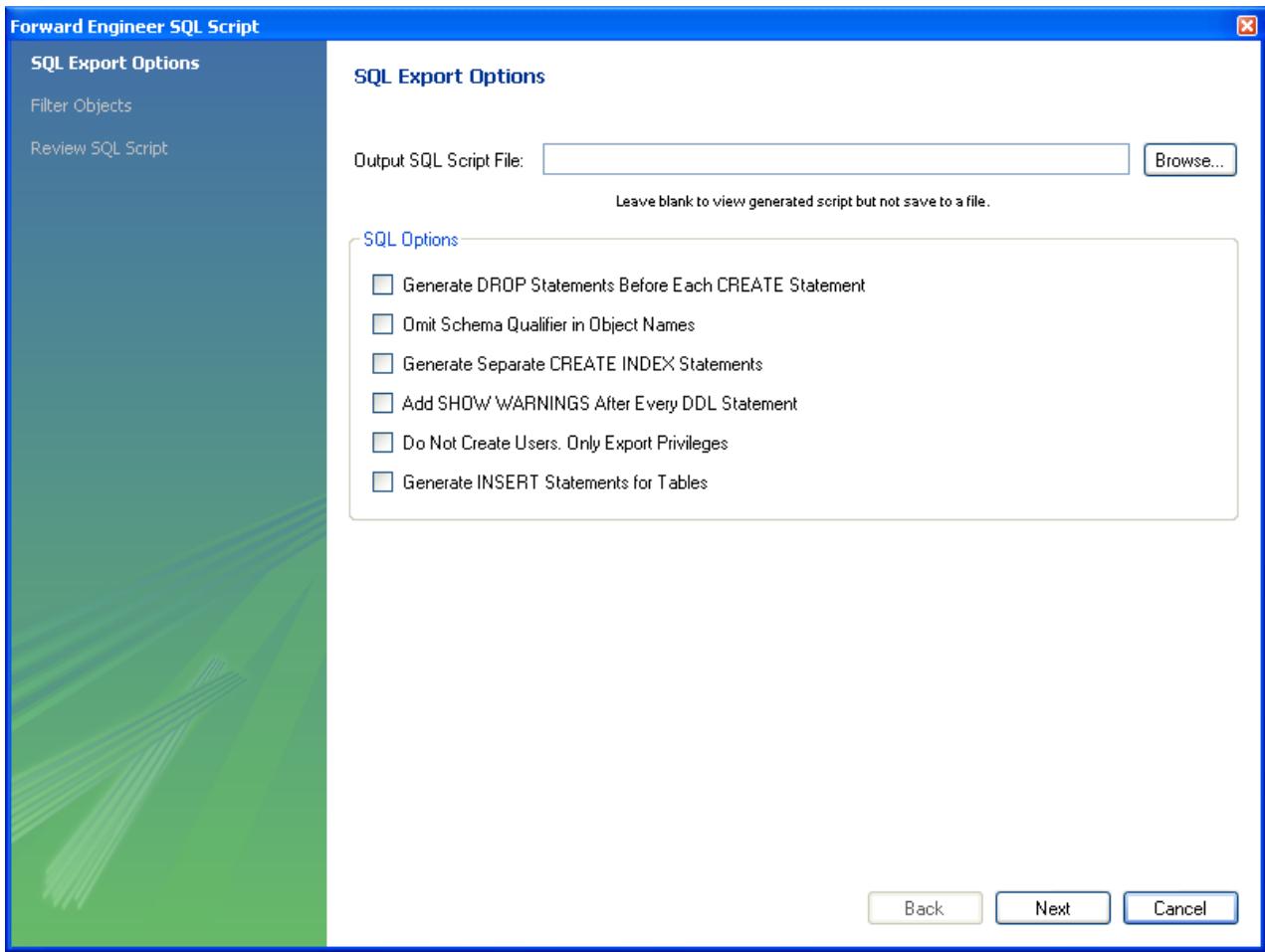
To create a script of your database model use the [EXPORT](#) option found under the [FILE](#) menu. You may export a script to alter an existing database or create a new database. The script to create a database is similar to the one created using the `mysqldump db_name` command.

If you choose to create a database, there are a number of export options that you may choose from.

7.5.10.1.1. Creating a Schema

Selecting [FILE](#), [EXPORT](#), [FORWARD ENGINEER SQL CREATE SCRIPT](#) will start the Forward Engineer SQL Script wizard. The first page of the wizard is:

Figure 7.23. SQL Export Options



The SQL Export Options displays the following facilities:

Output SQL Script File

You can enter the name of your output file by entering it into the **OUTPUT SQL SCRIPT FILE** text box, or by using the **BROWSE** button to select a file. If this text box is left blank you will be able to view the generated script, but it will not be saved to a file.

Generate DROP Statements Before Each CREATE Statement

Omit Schema Qualifier in Object Names

Generate Separate CREATE INDEX Statements

Choosing this option creates separate statements for index creation instead of creating indexes as part of a **CREATE TABLE** statement.

Add SHOW WARNINGS after every DDL statement

Do Not Create Users. Only Export Privileges

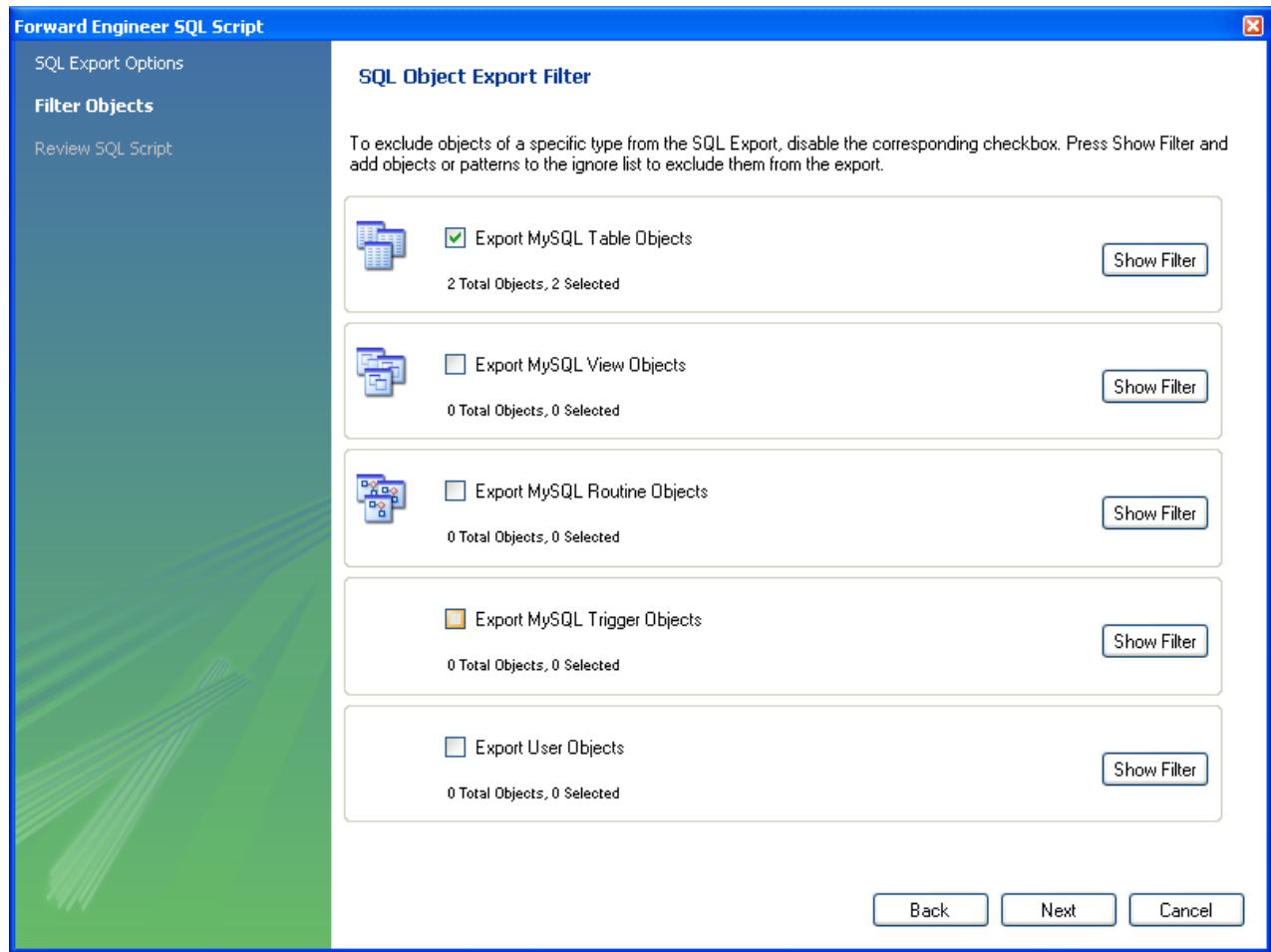
To update the privileges of existing users as opposed to creating new users, select this check box. Exporting privileges for nonexistent users will result in errors when you execute the **CREATE** script. Exporting users that already exist, will also result in an error.

Generate INSERT Statements for Tables

If you have added any records to a table using the **INSERT** tab of the MySQL Table Editor, choose this option. For more information about inserting records see [Section 7.5.1.3.9, “The Inserts Tab”](#).

Clicking **NEXT** takes you to the **SQL OBJECT EXPORT FILTER** page where you can select the objects you wish to export.

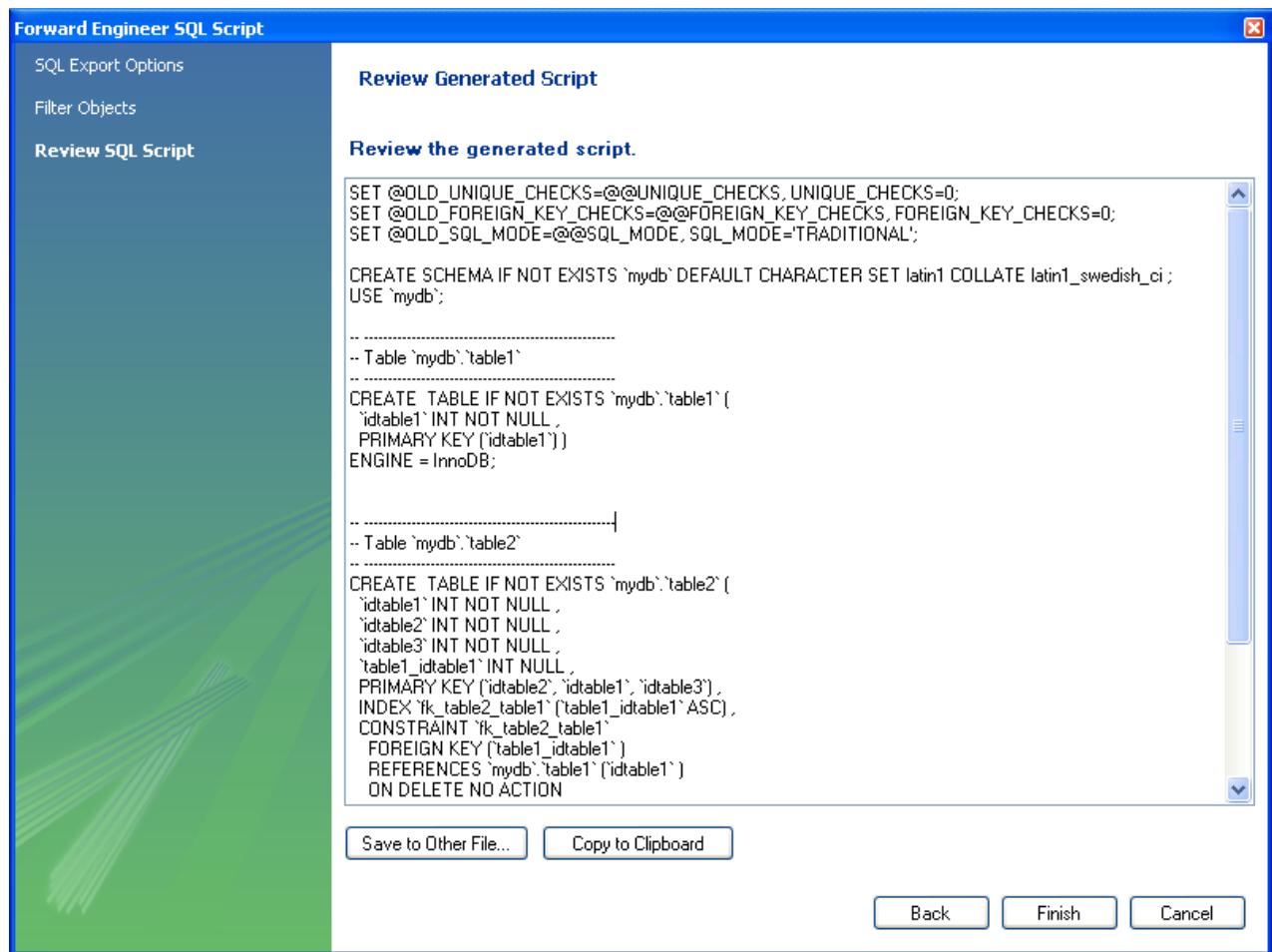
Figure 7.24. SQL Object Export Filter



Precise control over the objects to be exported can be fine tuned by clicking the SHOW FILTER button. Once the objects to be exported have been selected it is possible to reduce the expanded panel by clicking the same button, now labelled HIDE FILTER.

Having selected the objects you wish to export you can click the NEXT button to review the script that has been generated:

Figure 7.25. Review Generated Script



The FINISH button saves the script file and exits. You may return to the previous screen using the BACK button.

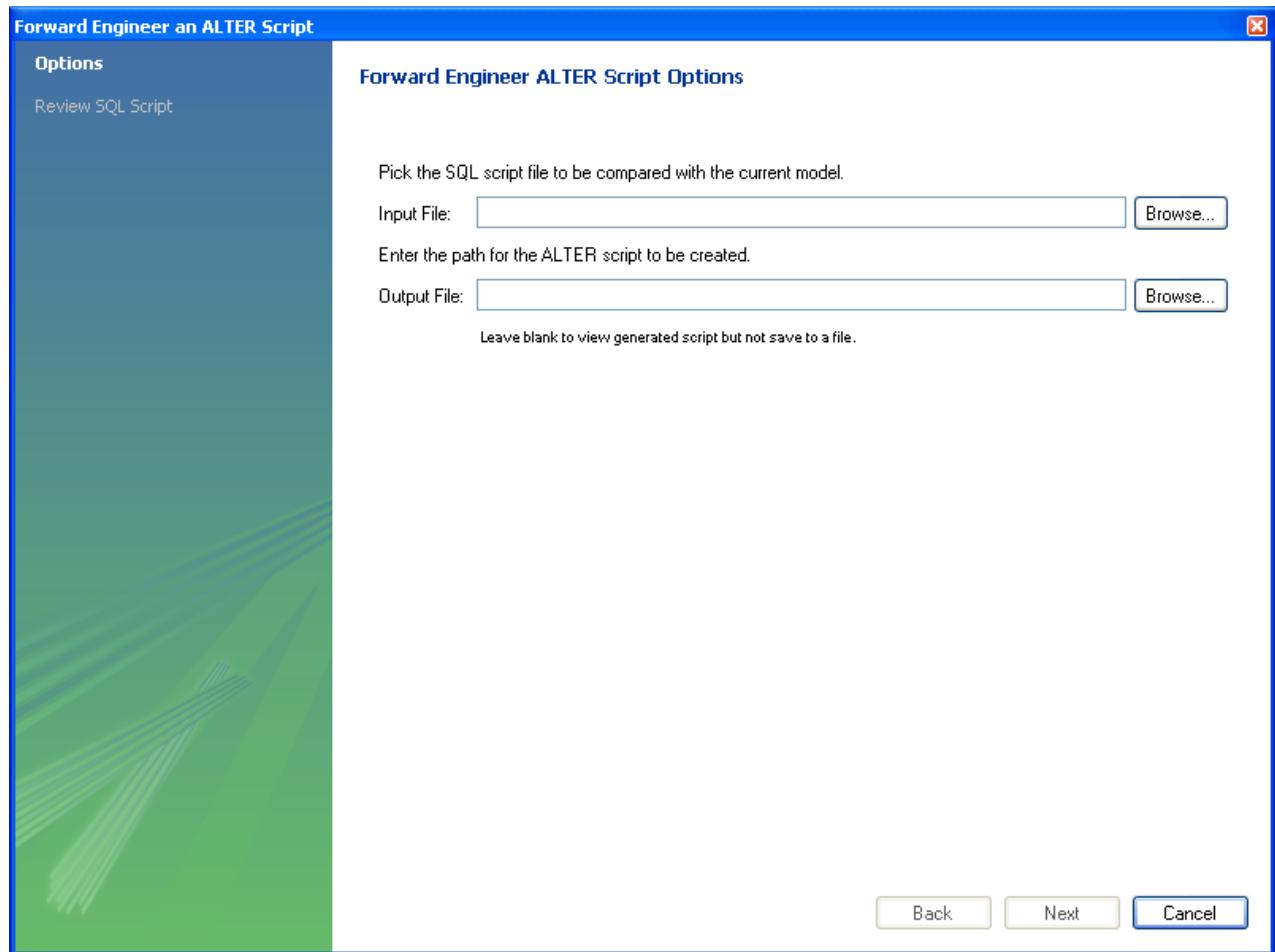
You can then use the saved script to create a database.

7.5.10.1.2. Altering a Schema

The menu option for altering a schema, FORWARD ENGINEER SQL ALTER SCRIPT ..., is used for updating a database that has been redesigned within MySQL Workbench. Typically, this option is used when the SQL script of a database has been imported into MySQL Workbench and changed, and then you want to create a script that can be run against a database to alter it to reflect the adjusted model. For instructions on importing a DDL script see [Section 7.5.9.1, “Reverse Engineering Using a Create Script”](#).

Select FILE, EXPORT, FORWARD ENGINEER SQL ALTER SCRIPT to start the Foward Engineer an ALTER Script wizard. You will be presented with the first page:

Figure 7.26. Options



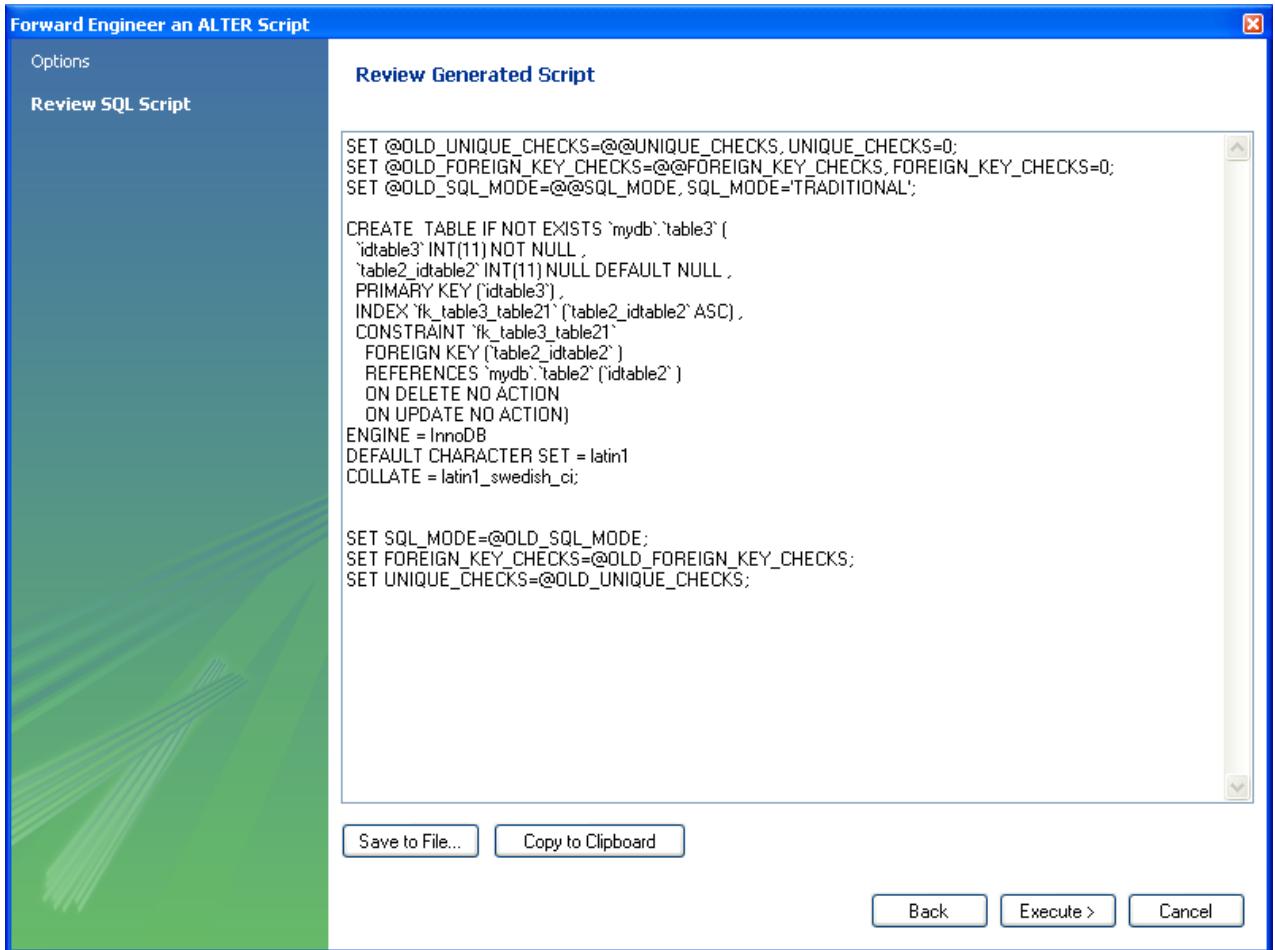
This first screen allows you to select a SQL script and compare it with the model currently in MySQL Workbench. The difference between the two models will be used to create an alter script that can be used to modify the target schema so that it matches the model held in MySQL Workbench. You can also simply view the script generated, rather than saving it to a file, by leaving the **OUTPUT FILE** text box empty.

Note

The script selected as the Input File must use full schema qualifiers, such as `schema_name.table_name`, otherwise MySQL Workbench will not be able to generate a useable alter script.

Pressing NEXT brings you to the **REVIEW SQL SCRIPT** screen.

Figure 7.27. Script



Here you can review and change the alter script that will be generated. Make any changes you wish and, if you are happy with the changes, save the [ALTER](#) script to file using the **SAVE TO FILE ...** button. You can also click the **EXECUTE** button to write the script to the previously specified output file.

The generated script can then be used to update the database.

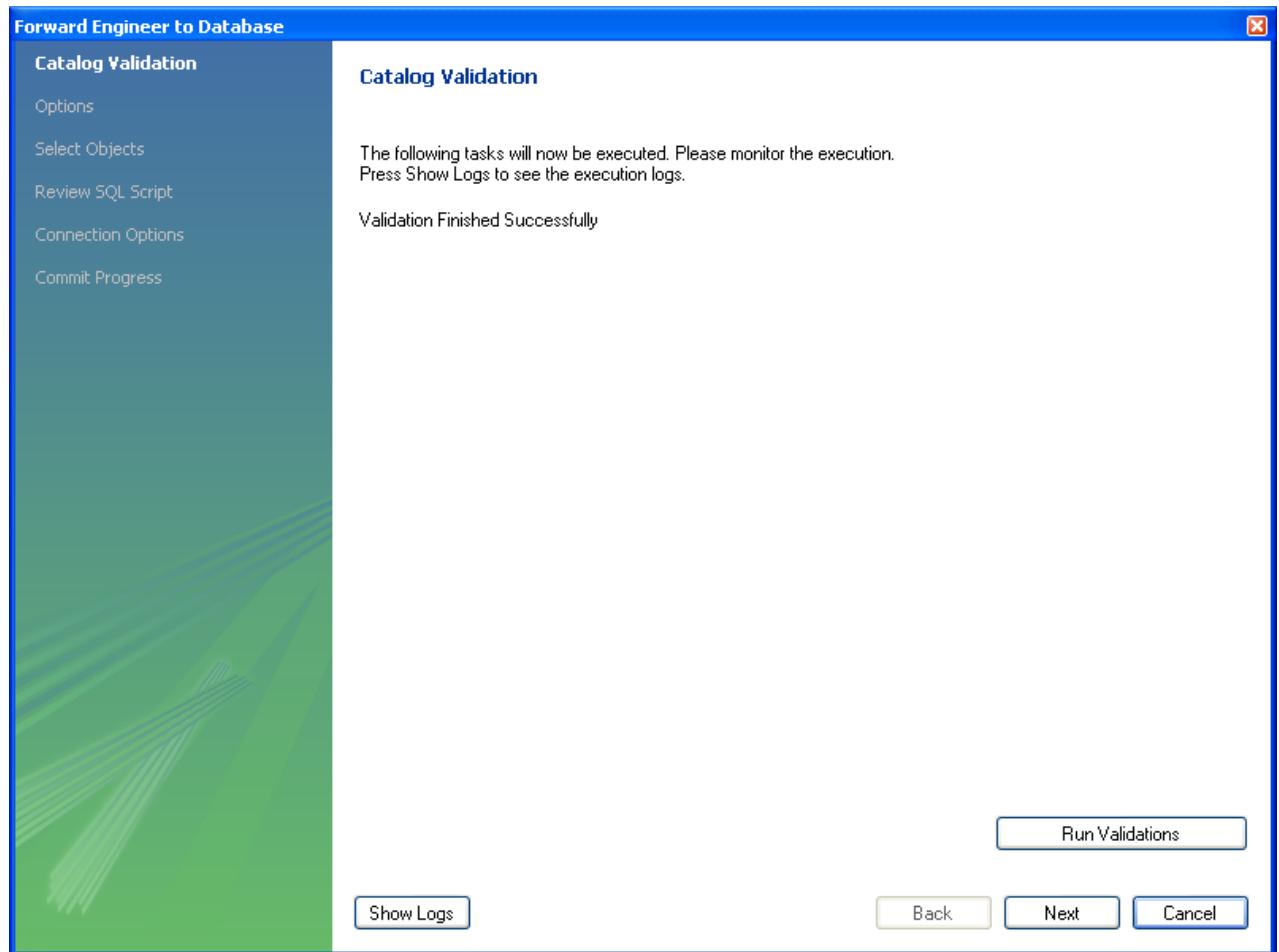
7.5.10.2. Forward Engineering to a Live Server

Use forward engineering to export your schema design to a MySQL server.

Select the schema that you wish to forward engineer and then choose **DATABASE, FORWARD ENGINEER ...** option from the main menu.

The first page to be displayed is Catalog Validation:

Figure 7.28. Catalog Validation



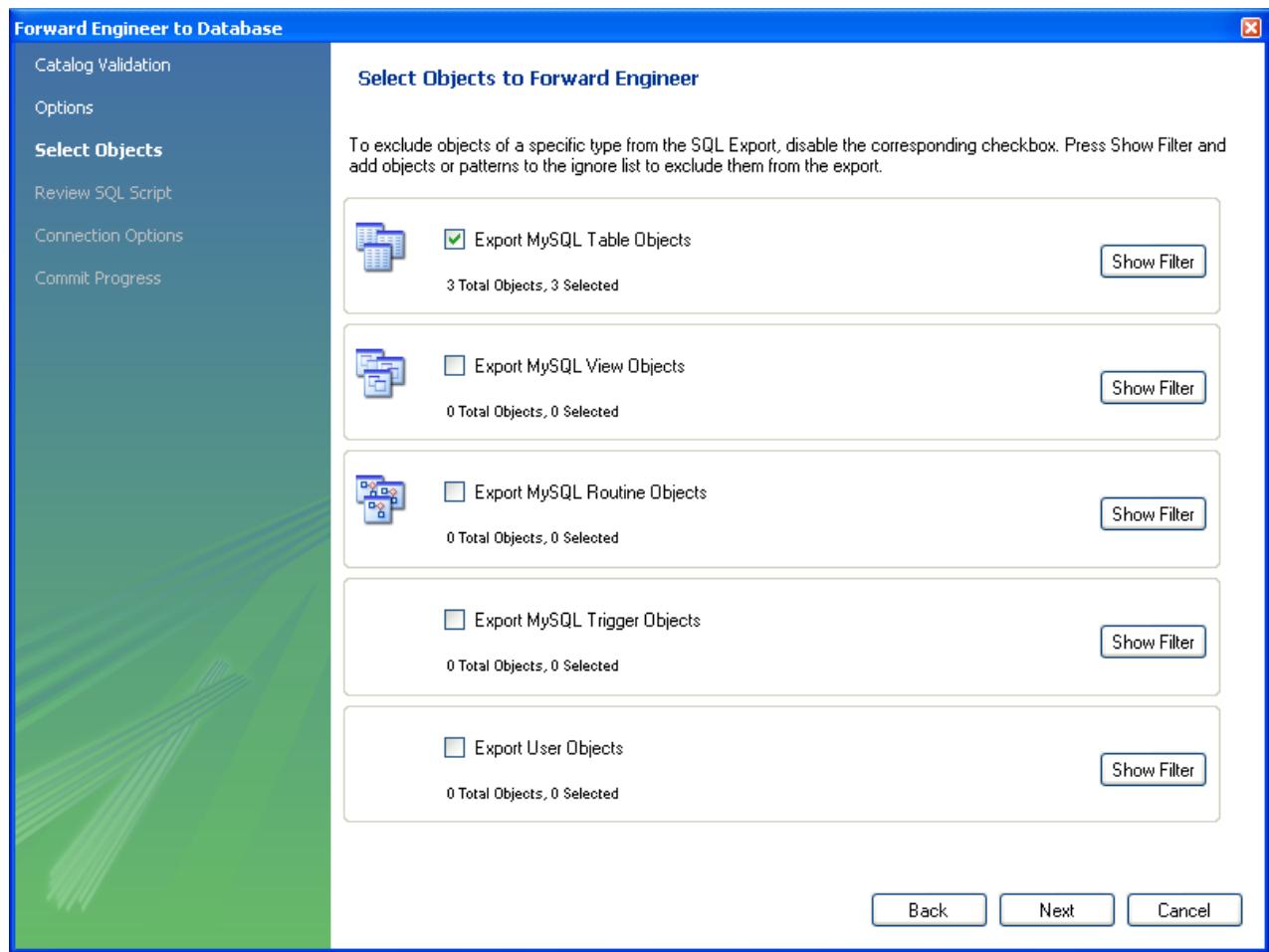
Click RUN VALIDATIONS to validate the catalog.

Click the NEXT to continue.

The next page allows you to set options for the database to be created. These options are as described in [Section 7.5.10.1.1, "Creating a Schema"](#). Select the required options and then click NEXT.

The next page allows you to select the objects to forward engineer:

Figure 7.29. Select Objects to Forward Engineer

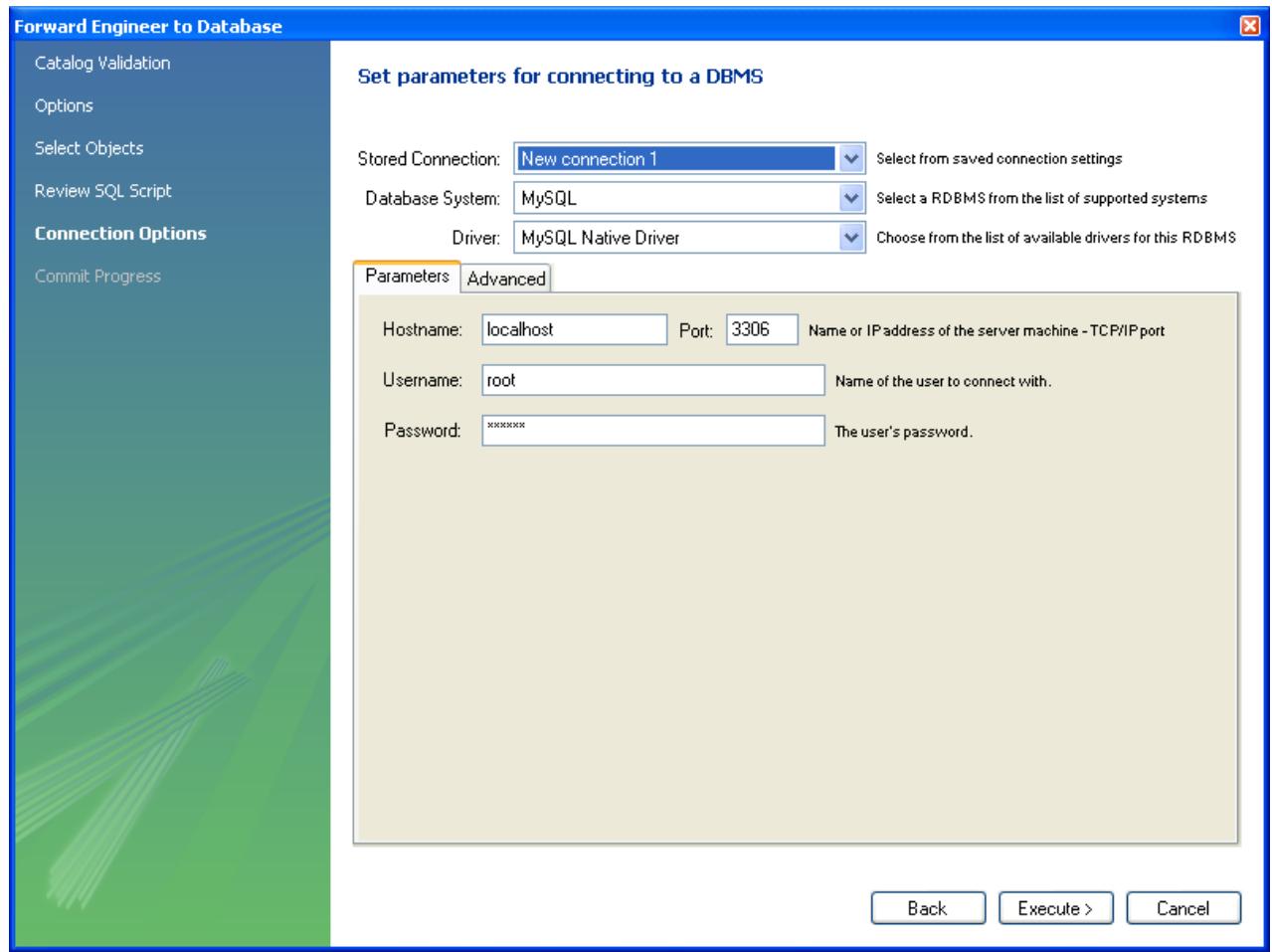


To select a subset of objects to forward engineer use the SHOW FILTER/HIDE FILTER button, and then select specific objects. Once you have selected your objects click NEXT to continue

On the [Review Script](#) page you may review and edit the SQL script that will be executed. Click NEXT to continue if you are satisfied with the generated script.

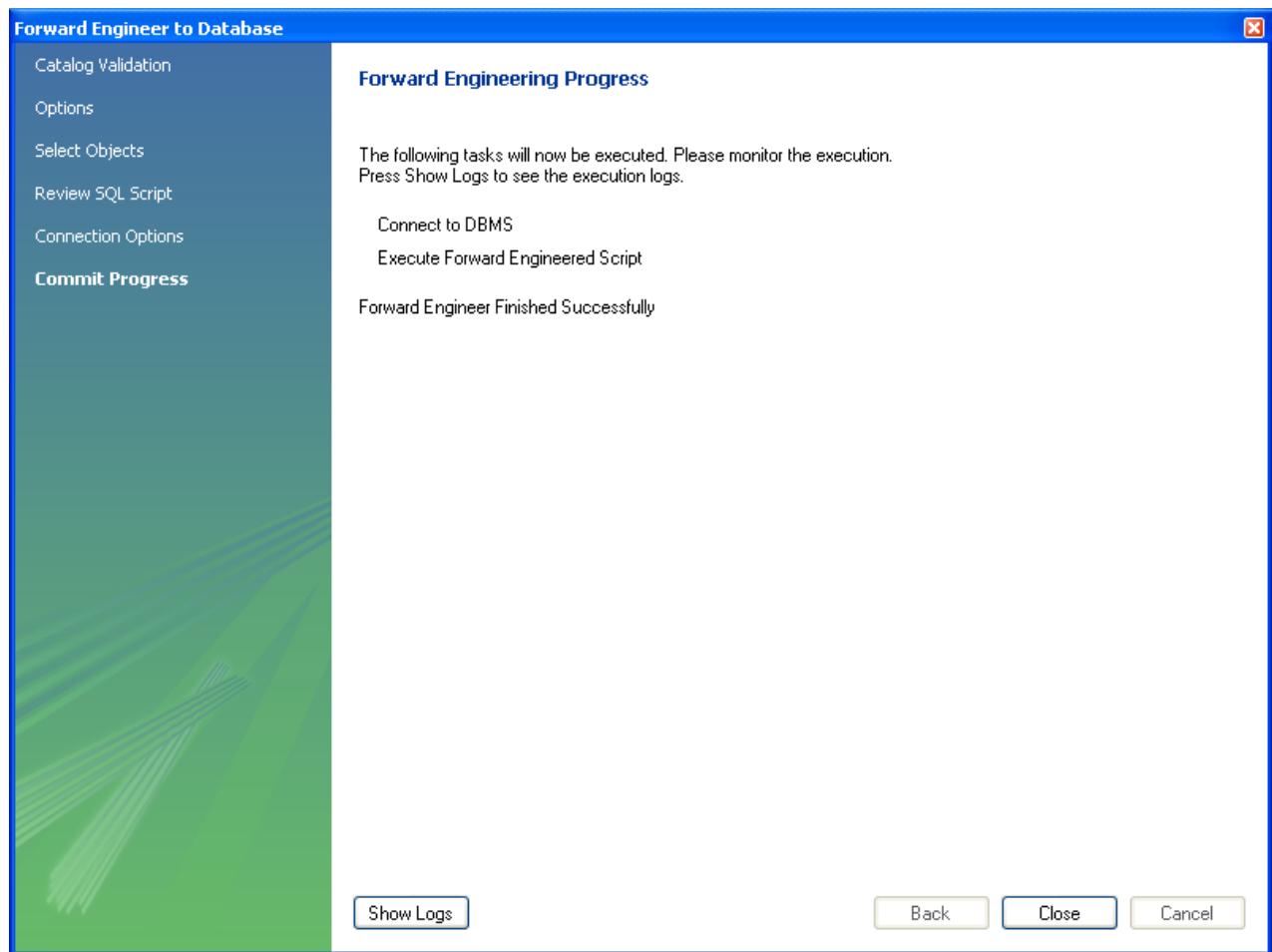
The next step of the process is to connect to a MySQL server in order to create the new database schema. This page allows you to use a previously stored connection, or enter the connection parameters:

Figure 7.30. Set parameters for connecting to a DBMS



Once the connection parameters have been set click EXECUTE. The next page of the wizard displays the results of the forward engineering process:

Figure 7.31. Set parameters for connecting to a DBMS



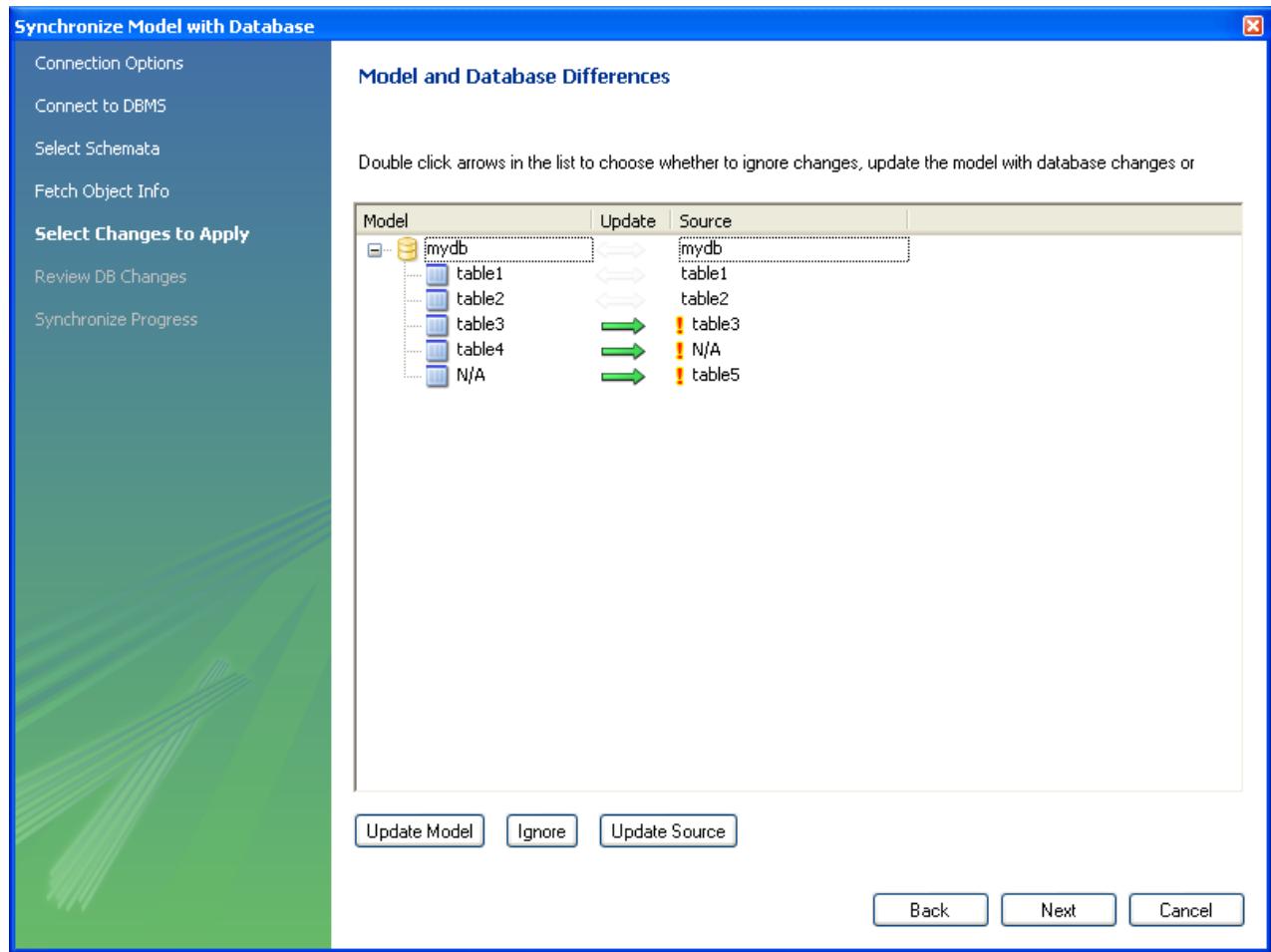
You can confirm the creation of the schema by connecting to the target MySQL server and issuing the `SHOW DATABASES;` command.

7.5.10.3. Database Synchronization

It is possible to synchronize a model in MySQL Workbench with a live database. By default, the synchronization process will change the live database to be the same as the model, but this is configurable during the synchronization process.

MySQL Workbench allows you control over the direction of synchronization, and the objects synchronized, in a completely flexible way. You can choose to synchronize only certain tables, allow synchronization to the live database only, allow synchronization from the live database to the model only, or a combination of directions. In effect you have complete control as to whether the synchronization is unidirectional or bidirectional, and which objects exactly are subject to synchronization. This is all controlled in the **SELECT CHANGES TO APPLY** page of the synchronization wizard:

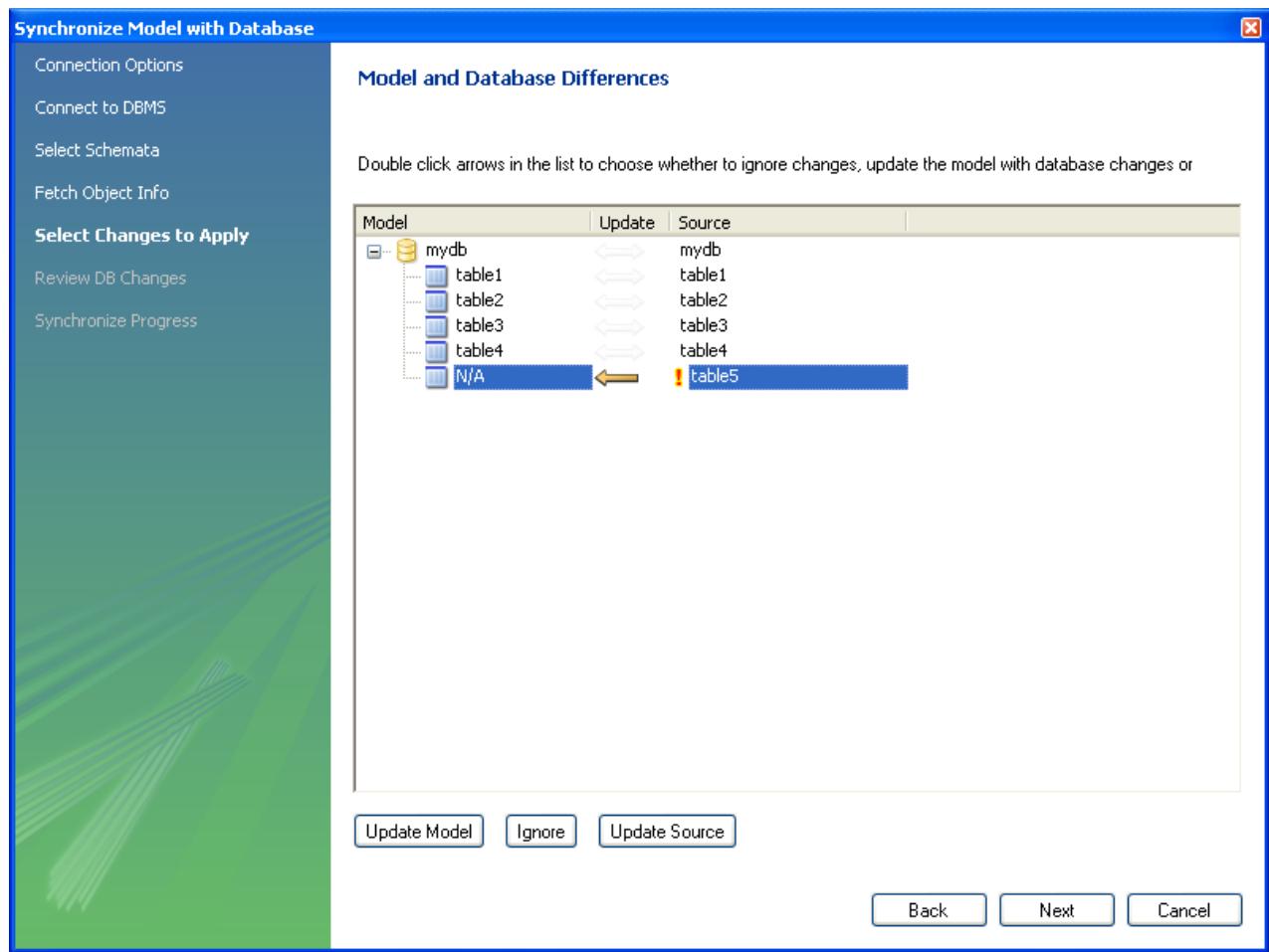
Figure 7.32. Model and Database Differences



In the above example the live database consists of `table1`, `table2` and `table3`. In MySQL Workbench an additional table, `table4`, has been created, along with a relationship between it and `table3`. Further, `table5` exists in the live database, but not in the model. The actions that are configured to occur would result in `table3` being altered (to include the relationship with `table4`), `table4` being created and `table5` being dropped, in the live database. It is possible to reconfigure this though.

The next example shows how the direction of synchronization can be changed:

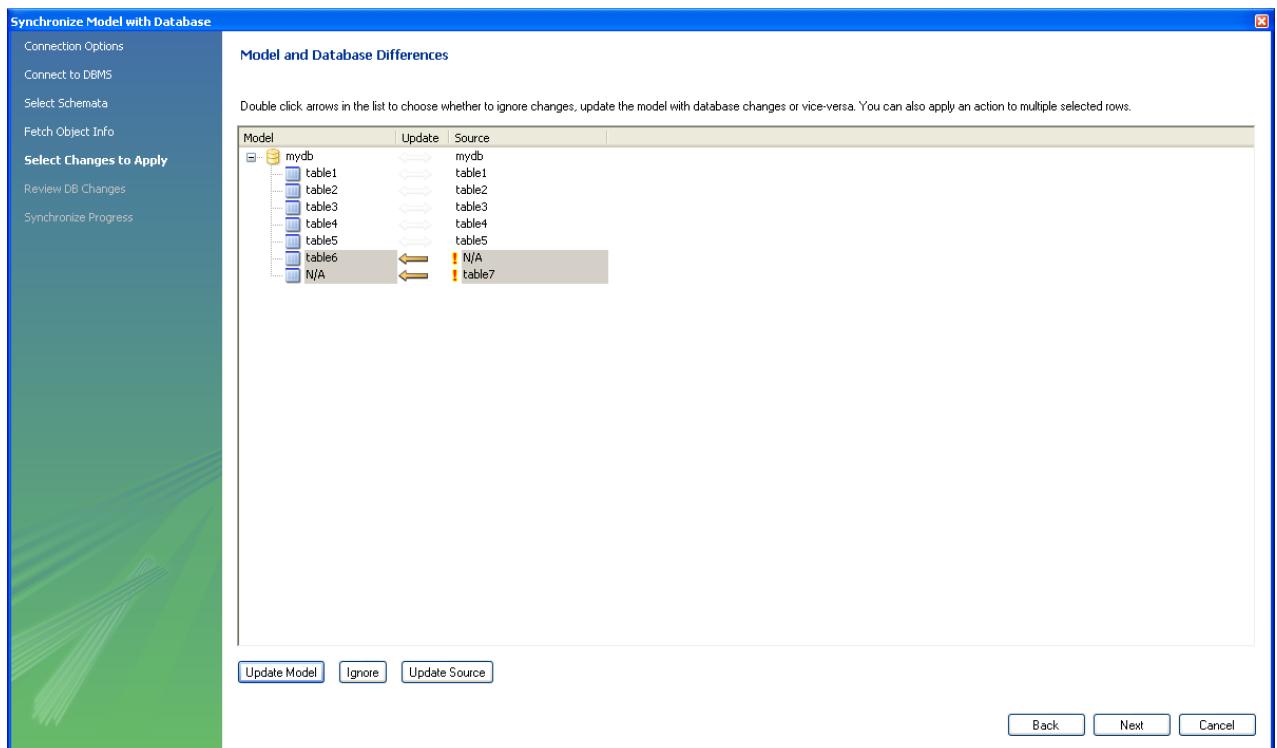
Figure 7.33. Controlling Synchronization Direction



In this case the synchronization direction has been changed so that rather than the default action of `table5` being dropped from the live database, it will be incorporated into the MySQL Workbench model.

For convenience the wizard provides three additional buttons to allow synchronization directions to be applied to a group of selected changes. The UPDATE MODEL button causes the selected changes to only be applied to the model itself:

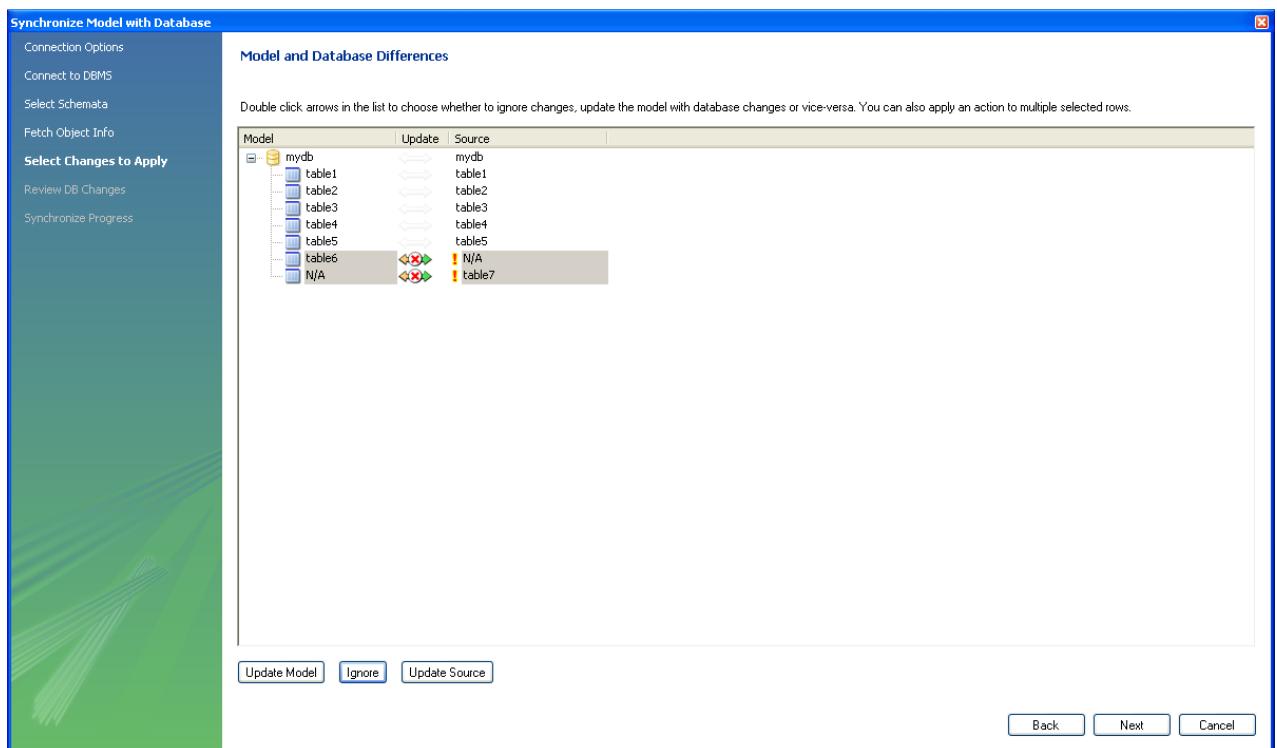
Figure 7.34. Update Model Button



In the above example **table7** would be added to the model.

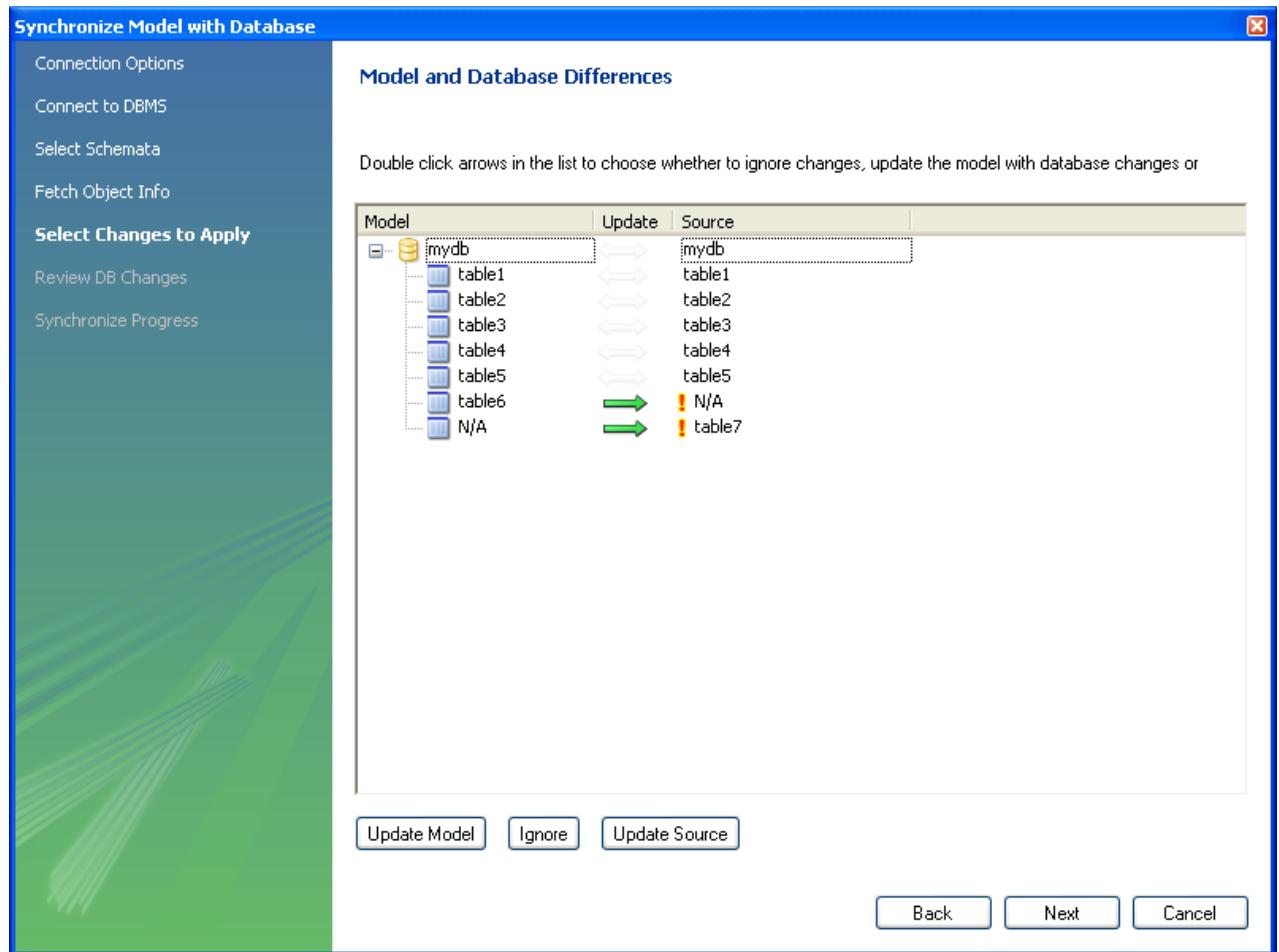
The IGNORE button causes the selected changes to be ignored, no synchronization will take place for those changes:

Figure 7.35. Ignore Button



In the above example no changes would take place.

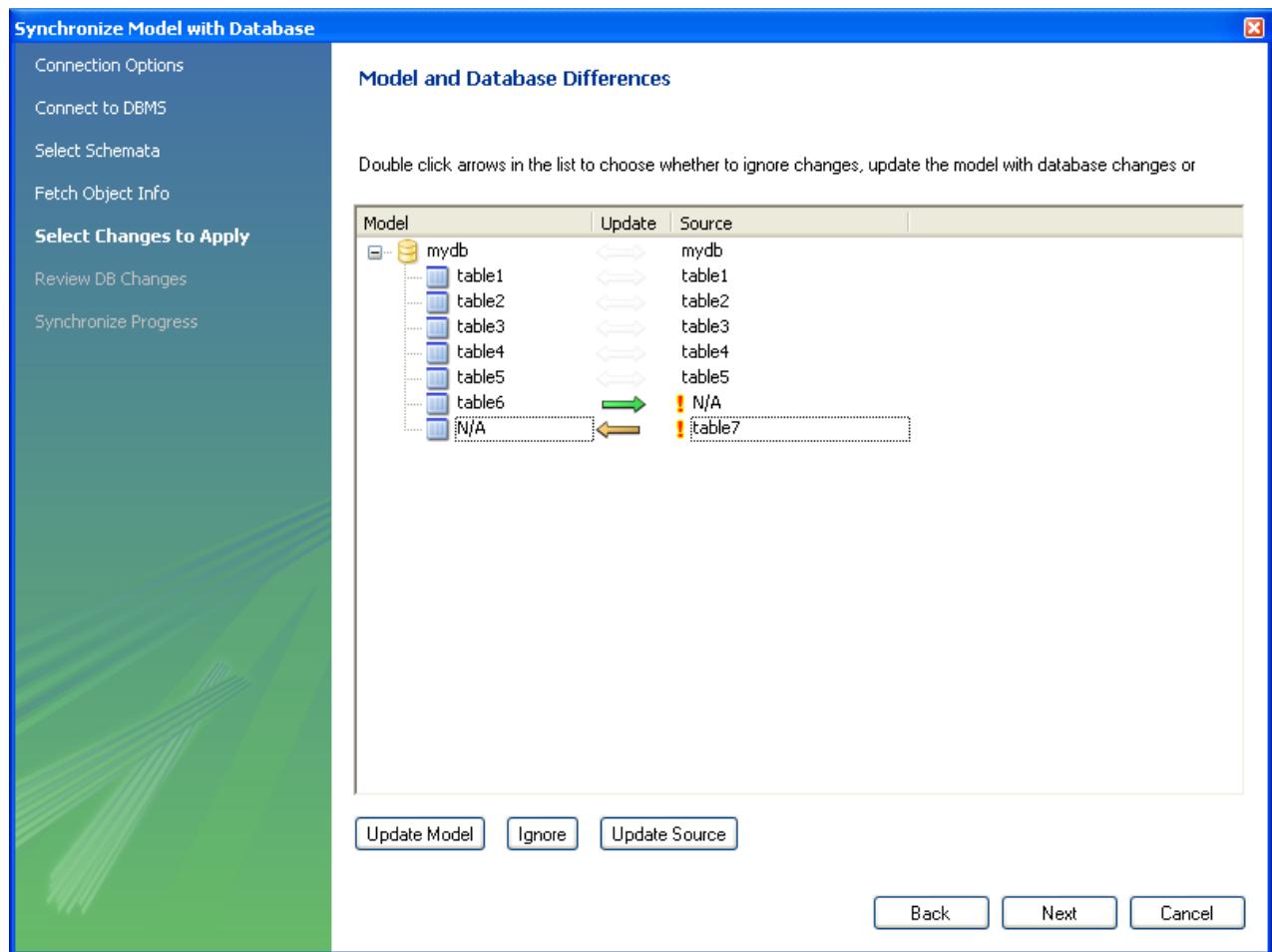
The UPDATE SOURCE button causes the selected changes to only update the live database:

Figure 7.36. Update Source Button

In this example `table6` would be added to the live database and `table7` would be dropped from the live database.

It is also possible to control individual changes by clicking on the arrows. Clicking an arrow causes it to change between the three available synchronization directions:

Figure 7.37. Click arrows to change direction of synchronization



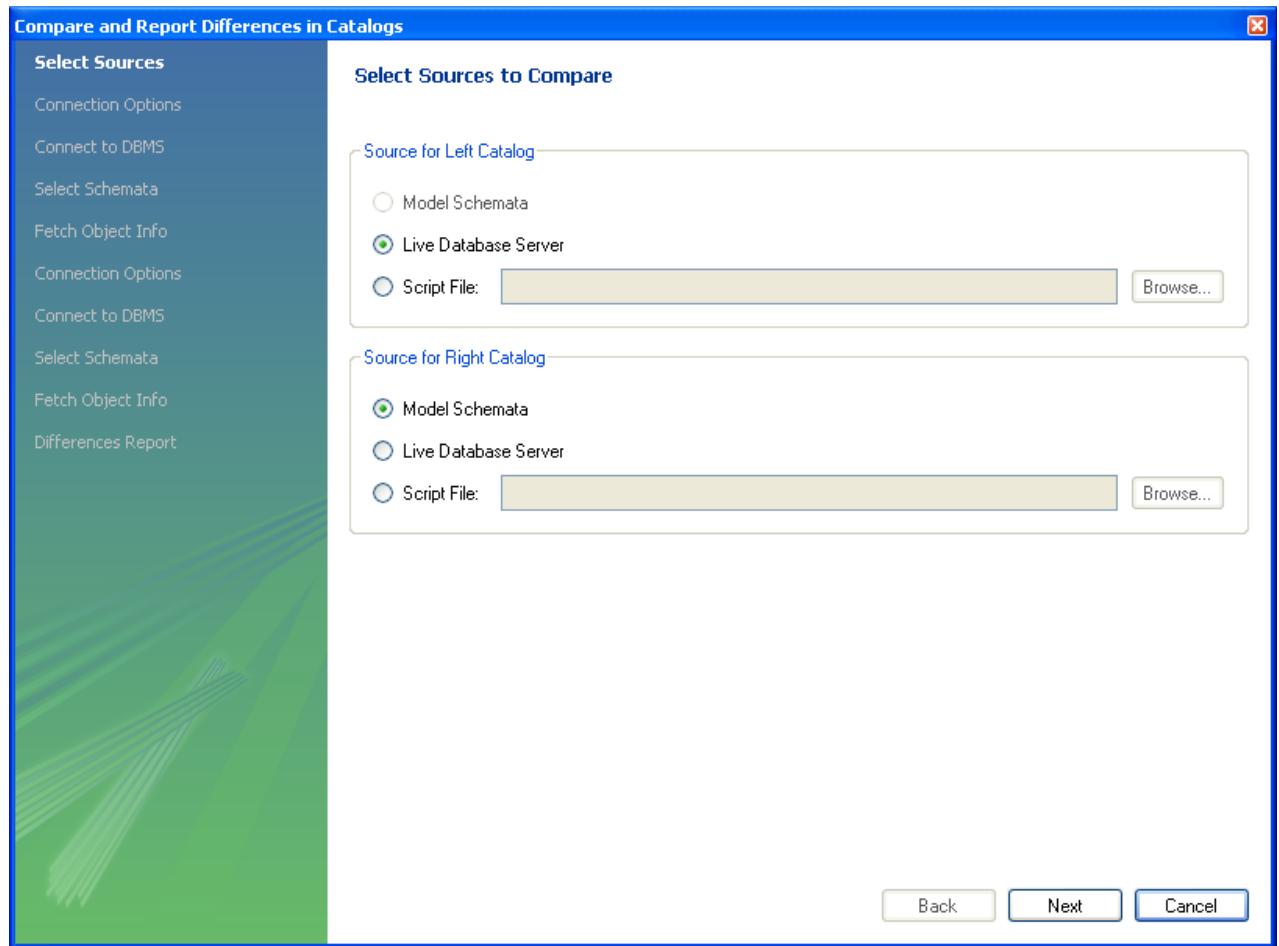
In the above example `table6` will be created in the live database, and `table7` will be created in the model.

7.5.10.4. Creating a Catalog Diff Report

This facility allows you to create a report detailing the differences between your MySQL Workbench model, and a live database or script. Select DATABASE, GENERATE CATALOG DIFF REPORT from the main menu to run the Compare and Report Differences in Catalogs wizard.

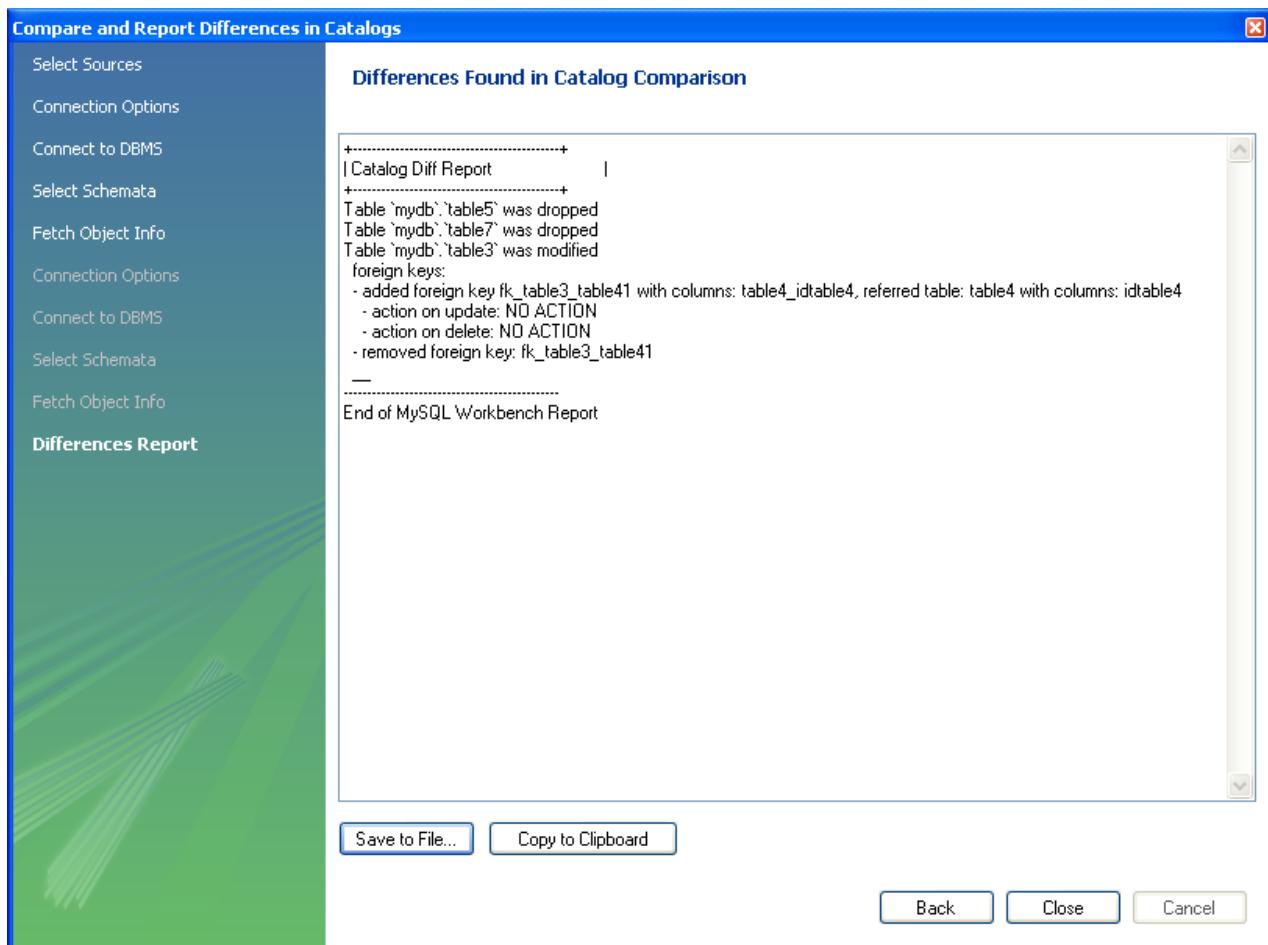
Having started the wizard the first step is to specify the catalogs you wish to compare. For example, you may simply wish to compare your live database against your current MySQL Workbench model:

Figure 7.38. Catalog Sources



You then proceed through the wizard, providing connection information if accessing a live database. The wizard will then produce a catalog diff report showing the differences between the compared catalogs:

Figure 7.39. Catalog Diff Report



7.6. Modeling Tutorials

This chapter contains three short tutorials intended to familiarize you with the basics of MySQL Workbench. These tutorials show how MySQL Workbench can be used both to design and to document databases.

Creating a database from scratch is the focus of [Section 7.6.2, “Using the Default Schema”](#) and exploring the graphic design capabilities of MySQL Workbench is touched upon in [Section 7.6.3, “Basic Modeling”](#). Both these tutorials show the database design capabilities of MySQL Workbench

Importing an SQL data definition script is probably the quickest way to familiarize yourself with MySQL Workbench — this tutorial makes use of the `sakila` database and emphasizes the use of MySQL Workbench as a documentation tool. Examples taken from the `sakila` database are used throughout the documentation so doing this tutorial can be very helpful in understanding MySQL Workbench.

7.6.1. Importing a Data Definition SQL Script

For this tutorial use the `sakila` database script found in the [Example Databases](#) section of the <http://dev.mysql.com/doc/> page.

After downloading the file, extract it to a convenient location. Open MySQL Workbench and find the [REVERSE ENGINEER MySQL CREATE SCRIPT](#) menu option by first selecting [FILE](#) and then [IMPORT](#). Find and import the `sakila-schema.sql` file. This is the script that contains the data definition statements for the `sakila` database. The file filter for the file open dialog window defaults to `*.sql` so you should only be able to view files with the `sql` extension.

If the file was successfully imported, the application's status bar reads, [Import MySQL Create Script done](#). To view the newly imported script, expand the [Physical Schemata](#) section by double clicking the arrow on the left of the [Physical Schemata](#) title bar. Select the tab labelled **SAKILA**.

Yo may also wish to remove the default schema tab, `mydb`. Do this by selecting this tab and then clicking the **-** button on the upper right in the **PHYSICAL SCHEMATA** panel.

To view all the objects in the `sakila` schema, you may need to expand the **PHYSICAL SCHEMATA** window. To do this move the

mouse pointer anywhere over the gray area that defines the lower edge of the **PHYSICAL SCHEMATA** window. Hold down the right mouse button and move the mouse to adjust the size of the window.

Once you've expanded the window, all the objects in the `sakila` database should be visible. Tables appear at the top followed by views and then routines. There are no routine groups in this schema, but you should see the **ROUTINE GROUPS** section and an [Add Group](#) icon.

For a complete description of importing a MySQL create script see [Section 7.5.9.1, “Reverse Engineering Using a Create Script”](#).

7.6.1.1. Adding an EER Diagram

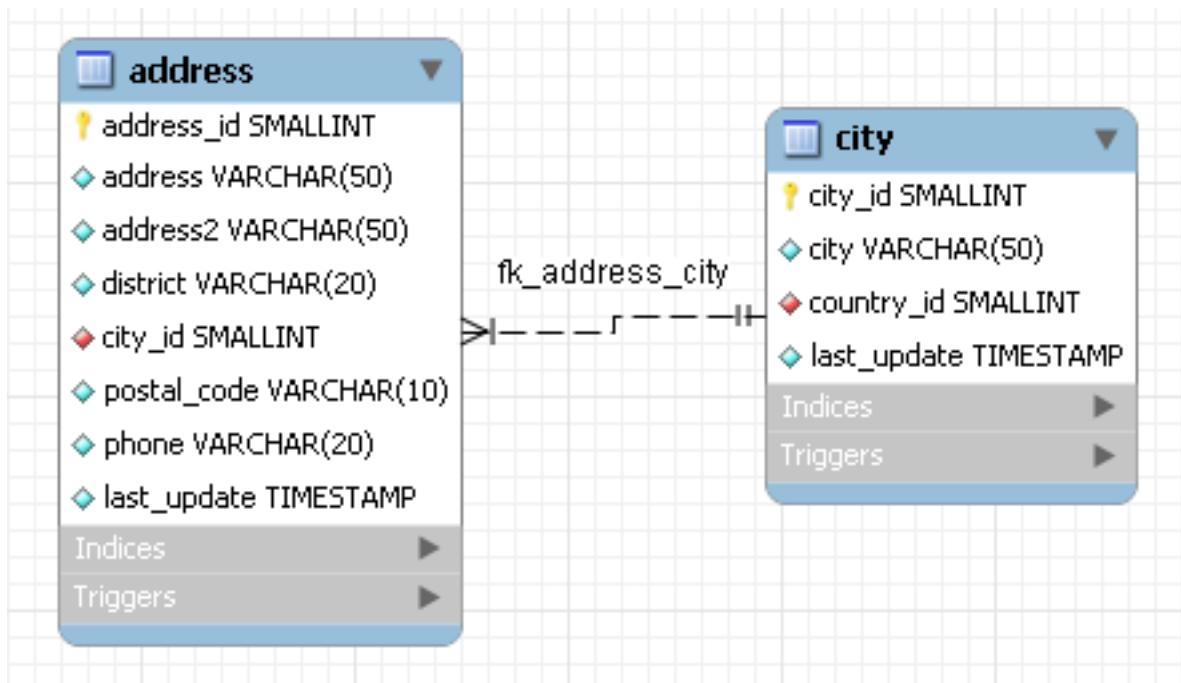
To create an EER diagram for the `sakila` database, first add an EER diagram by double clicking the [Add Diagram](#) icon in the **EER DIAGRAMS** panel. This should create and open a new [EER Diagram](#).

The [EER Diagram](#) canvas is where object modeling takes place. To add a table to the canvas, select the **CATALOG** tab in the middle panel on the right side of the application. This should display any schemata that appear in the **MySQL MODEL** tab.

Find the `sakila` schema and expand the view of its objects by clicking the + button to the left of the schema name. Expand the tables list in the same way.

You can add tables to the EER canvas by picking them up from the **CATALOG** panel and placing them on the canvas. Drop the `address` table and the `city` table onto the canvas.

Figure 7.40. Adding Tables to the Canvas



MySQL Workbench automatically discovers that `address.city_id` has been defined as a foreign key referencing the `city.city_id` field. Drop the `country` table onto the canvas and immediately you should see the relationship between the `country` table and the `city` table. (See [Figure 7.43, “The sakila EER Diagram”](#) to view a PNG file of all the relationships in the `sakila` database.)

Choose the **Properties** tab of the panel on the lower right and then click one of the tables on the canvas. This displays the properties of the table in the **Properties** window. While a table is selected you can use the **Properties** window to change a table's properties. For example, entering `#FF0000` for the color value will change the color accent to red.

Changing the color of a table is a good way to identify a table quickly — something that becomes more important as the number of tables increases. Changing the color of a table is also an easy way to identify a table in the **Model Navigator** panel. This panel, the uppermost panel on the left side of the screen, gives a bird's eye view of the entire EER canvas.

Save your changes to a **MySQL Workbench Models** file (mwb) by choosing [SAVE](#) from the **FILE** menu or by using the keyboard command **Ctrl S**.

7.6.2. Using the Default Schema

When you first open MySQL Workbench a default schema, `mydb` appears as the left-most tab of the **PHYSICAL SCHEMATA** section of MySQL Workbench. You can begin designing a database by using this default schema.

Figure 7.41. The Default Schema



To change the name of the default schema, double click the schema tab. This opens a schema editor window docked at the bottom of the application. To undock or redock this window, simply double click anywhere in the editor title bar.

To rename the schema, use the text box labeled **NAME**. Once you have renamed the schema a lightning bolt icon appears right aligned in the **NAME** text box, indicating that other changes are pending. Click in the **COMMENTS** text area and a dialog box opens asking if you wish to rename all schema occurrences. Clicking **YES** ensures that your changes are propagated throughout the application. Add comments to the database and change the collation if you wish. Close the schema editor by clicking the X button.

7.6.2.1. Creating a New Table

Create a new table by double clicking the **ADD TABLE** icon in the **Physical Schemata** panel. Doing this opens the table editor docked at the bottom of the application. If you wish, you can undock or dock this editor in exactly the same way as the schema editor window.

Use the first tab of the table editor to change the name, collation, and engine. You may also add a comment.

Add columns to the new table by selecting the **COLUMNS** tab. Use the default column name or enter a new name of your choosing. Use the **Tab** key to move to the next column and set the column's data type.

Altering the table by adding indexes or other features is also easily done using the table editor.

7.6.2.2. Creating Other Schema Objects

Additional objects such as views or routines can be added in the same way as tables.

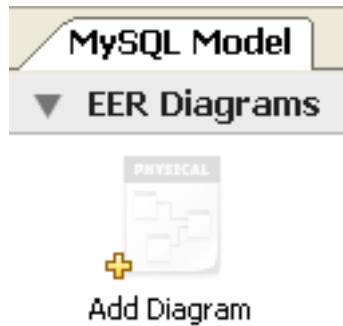
Any objects you have created can be found in the **CATALOG** palette on the right. To view these schema objects select the **CATALOG** tab in the middle palette on the right. View all the objects by clicking the + button to the left of the schema name.

Save your changes to a **MySQL Workbench Models** file (mwb) by choosing **SAVE** from the **FILE** menu or by using the keyboard command **Ctrl S**.

7.6.3. Basic Modeling

On the **MySQL Model** page double click the **ADD DIAGRAM** icon. This creates and opens a new **EER Diagram**.

Figure 7.42. Adding an EER Diagram



From an EER diagram page you can graphically design a database.

7.6.3.1. Adding a Table

The tools in the vertical toolbar on the left of the **EER DIAGRAM** tab are used for designing an EER diagram. Start by creating a table using the table tool. The table tool is the rectangular grid in the middle of the vertical toolbar. Mousing over it shows the message, **Place a New Table (T)**.

Clicking on this tool changes the mouse pointer to a hand with a rectangular grid. Create a table on the canvas by clicking anywhere on the **EER Diagram** grid.

Right click the table and choose **EDIT IN NEW WINDOW** from the pop-up menu. This opens the table editor, docked at the bottom of the application.

The table name defaults to **table1**. Change the name by entering **invoice** into the **NAME:** text box. Notice that the name of the tab in the table editor and the name of the table on the canvas, both change to this new value.

Pressing **Tab** or **Enter** while the cursor is in the table name text box, selects the **COLUMNS** tab of the table editor and creates a default column named, **idinvoice**.

Pressing **Tab** or **Enter** again sets the focus on the **Datatype** drop-down list box with **INT** selected. Notice that a field has been added to the table on the EER canvas.

Pressing **Tab** yet again and the focus shifts to adding a second column. Add a **Description** and a **Customer_id** column. When you are finished, close the table editor, by clicking the **X** button on the top left of the table editor.

7.6.3.2. Create a Foreign Key

Select the table tool again and place another table on the canvas. Name this table **invoice_item**. Next click on the **1:n Non-Identifying Relationship** tool.

First click on the **invoice_item** table; notice that a red border indicates that this table is selected. Next click on the **invoice** table. Doing this creates a foreign key in the **invoice_item** table, the table on the “many” side of the relationship. This relationship between the two tables is shown graphically in crow’s foot notation.

Revert to the default mouse pointer by clicking the arrow at the top of the vertical toolbar. Click on the **invoice_item** table and select the **FOREIGN KEYS** tab.

Click in the **FOREIGN KEY NAME** text box. The referenced table should show in the **REFERENCED TABLE** column and the appropriate column in the **REFERENCED COLUMN** column.

To delete the relationship between two tables, click on the line joining the tables and then press **Ctrl Delete**.

Experiment with the other tools on the vertical toolbar. Delete a relationship by selecting the eraser tool and clicking on the line joining two tables. Create a view, add a text object, or add a layer.

Save your changes to a **MySQL Workbench Models** file (MWB) by choosing **SAVE** from the **FILE** menu or by using the keyboard command **Ctrl S**.

7.6.4. Documenting the **sakila** Database

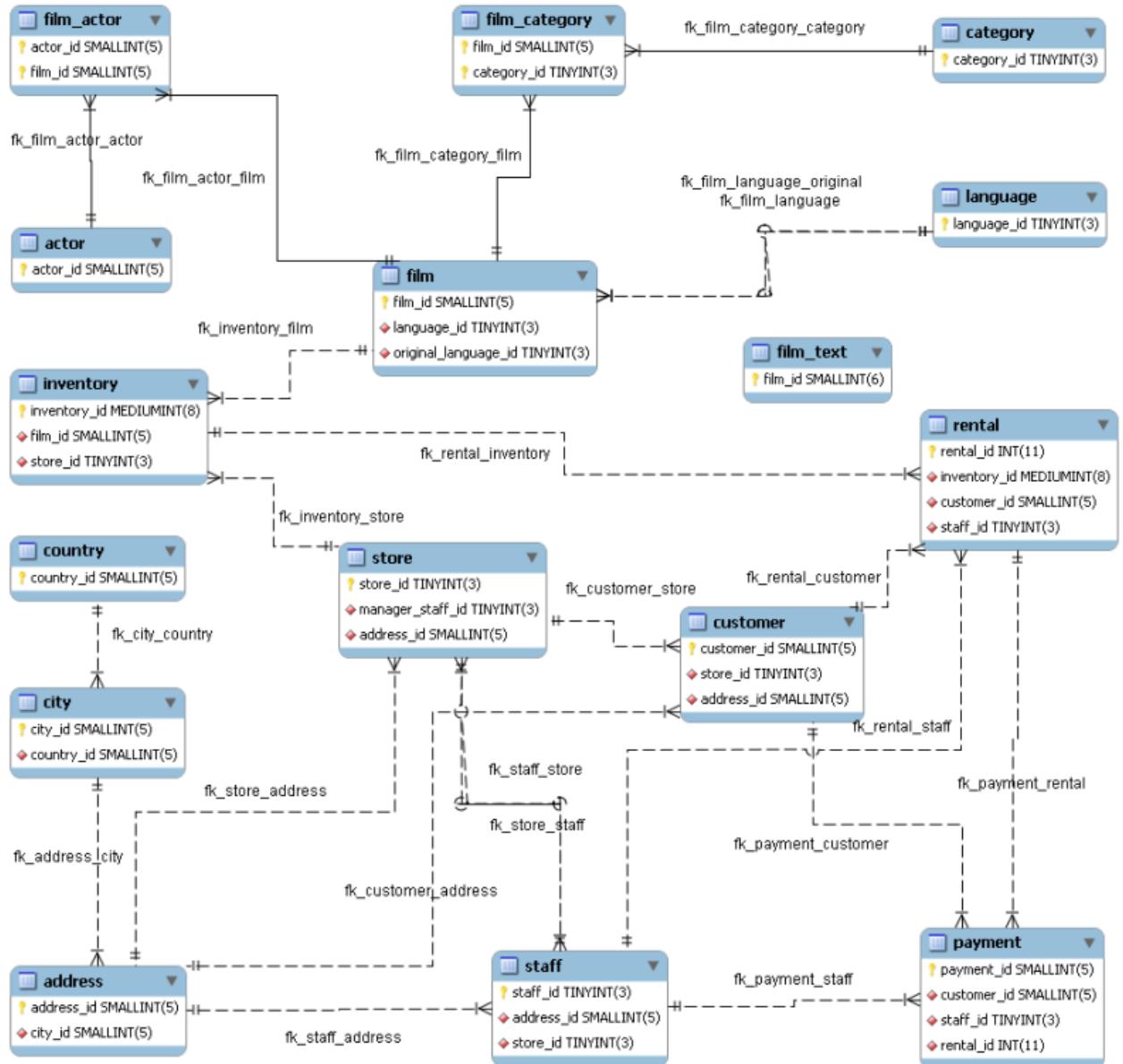
This chapter highlights the capabilities of MySQL Workbench as a documentation tool using the **sakila** database as an example. This is a sample database provided by MySQL and found in the **Example Databases** section of the <http://dev.mysql.com/doc> page. An EER diagram is an invaluable aid to a quick understanding of any database. There is no need to read through table definition statements; glancing at an EER diagram can immediately indicate that various tables are related.

You can also see how tables are related; what the foreign keys are and what the nature of the relationship is.

7.6.4.1. A PNG File of the `sakila` Database

Find below an EER diagram showing all the tables in the `sakila` database. This image was created using the menu options FILE, EXPORT, EXPORT AS PNG ...

Figure 7.43. The `sakila` EER Diagram



The object notation style used in Figure 7.43, “The `sakila` EER Diagram” is [Workbench \(PKs only\)](#). This notation only shows primary keys and no other columns so it is especially useful where space is at a premium. The relationship notation is the default, Crow’s Foot.

As the connection lines show, each table is related to at least one other table in the database (with the exception of the `film_text` table). Some tables have two foreign keys that relate to the same table. For example the `film` table has two foreign keys that relate to the `language` table, namely `fk_film_language_original` and `fk_film_language`. Where there is more than one relationship between two tables, the connection lines run concurrently.

Identifying and nonidentifying relationships are indicated by solid and broken lines respectively. For example, the foreign key `category_id` is part of the primary key in the `film_category` table so its relationship to the `category` table is drawn with

a solid line. On the other hand, in the `city` table, the foreign key, `country_id`, is not part of the primary key so the connection uses a broken line.

7.7. Printing

The printing options are used to create printouts of your EER Diagrams and are found under the FILE menu. For creating *documentations* of your models see [Section 7.1.5.1, “The DBDoc Model Reporting Dialog Window \(Commercial Version\)”](#).

7.7.1. Printing Options

The printing menu options are grayed if an EER Diagram is not active. The menu options are as follows:

- PAGE SETUP ...

Use this option to choose the paper size, orientation, and margins.

- PRINT

Use this option to send your EER Diagram directly to the printer. This option generates a preview before printing. From the preview you can adjust the scale of the view and also choose a multi-page view. Clicking the printer icon at the top left of this window, prints the currently selected EER Diagram. Close the print preview window if you need to adjust the placement of objects on the EER Diagram canvas.

- PRINT TO PDF ...

Use this option to create a PDF file of your EER Diagram.

- PRINT TO PS ...

Use this option to create a PostScript file of your EER Diagram.

7.8. MySQL Workbench Schema Validation Plugins (Commercial Version)

MySQL Workbench provides validation modules so that you can test your models before implementing them.

The validation plugins are accessed from the MODEL menu option. One plugin performs general validation for any Relational Database Management System (RDMS) and the other is MySQL-specific. Beneath these menu items are a number of specific validation tests. Running any one of these tests opens an output window docked at the bottom of the application. Warning messages are displayed on the left side of this window and the tests performed are displayed on the right.

The tasks performed by the validation modules are outlined in what follows.

7.8.1. General Validation

The types of validation and examples that violate validation are listed in what follows:

- **Empty Content validation**

- A table with no columns
- A routine or view with no SQL code defined
- A routine group containing no routines
- A table, view, or routine not referenced by at least one role
- A user with no privileges
- Objects such as tables that do not appear on at least one EER Diagram

- **Table Efficiency Validation**

- A table with no primary key
- A primary key that does not use an integer-based data type

- A foreign key that refers to a column with a different data type
- **Duplicated Identifiers Validation**
 - Duplicate object names
 - Duplicate role or user names
 - Duplicate index or routine names
- **Consistency Validation**
 - Use of the same column with columns of differing data types
- **Logic Validation**
 - A foreign key that refers to a column other than the primary key in the source table
 - Any object that is object is either read- or write-only by role definition
 - Placeholder objects left over from reverse engineering

7.8.2. MySQL-Specific Validation

The types of MySQL-specific validation and examples that violate validation are listed in the following.

- **Integrity Violation**
 - An object name longer than the maximum allowed
 - A foreign key defined for an engine type that doesn't support foreign keys (not yet implemented)
 - A view or routine that references a nonexistent table (not yet implemented)
 - A default value that does not match a column's data type
 - An invalid partitioning scheme
- **Syntax Violation**
 - A routine, trigger, or view with incorrect SQL syntax
 - A reserved keyword used as an identifier
 - Use of an invalid character

7.9. Customizing DBDoc Model Reporting Templates

This document aims to provide an overview of creating and modifying DBDoc Model Reporting templates, as used by MySQL Workbench.

The MySQL Workbench DBDoc Model Reporting system is based on the [Google Template System](#). This document does not attempt to explain the Google Template System in detail. The Google document [How To Use the Google Template System](#) provides a useful overview of how the Google Template System works.

The templates employed by the DBDoc Model Reporting system are text files that contain [Markers](#). These text files are processed by the template system built into MySQL Workbench, and the markers replaced by actual data. The output files are then generated. It is these output files, typically HTML or text, that are then viewed by the user.

Markers can be of six types:

1. Template Include
2. Comment
3. Set delimiter

-
4. Pragma
 5. Variable
 6. Section start and Section end

The last two are the most commonly used in MySQL Workbench templates and these important markers will be briefly described in the following sections.

1. Variables

The use of variables in the templates is straightforward. Any variables denoted by markers in the template file, will be replaced by their corresponding data, prior to the output file being generated. The mapping between variables and their corresponding data is stored by MySQL Workbench in what is known as a Data Dictionary. In the data dictionary the variable name is the *key* and the variable's corresponding data is the *value*. The data dictionaries are built by MySQL Workbench and filled with the data contained in the model being processed.

By way of example, the following code snippet shows part of a template file:

```
Total number of Schemata: {{SCHEMA_COUNT}}
```

In the generated output file the variable {{SCHEMA_COUNT}} will be replaced by the number of schemata in the model:

```
Total number of Schemata: 2
```

A variable can appear as many times as required in the template file.

2. Sections

Sections are used to perform iteration in the templates. When MySQL Workbench exchanges the variables in a section for data it will do so iteratively, using all data in the data dictionary in which the variable is defined. MySQL Workbench builds the data dictionaries according to the model currently being processed.

Again, this is best illustrated by example:

```
{{#SCHEMATA}}
Schema: {{SCHEMA_NAME}}
{{/SCHEMATA}}
```

In the previous code snippet the section start is indicated by the {{#SCHEMATA}} marker. The end of the section is indicated by the {{/SCHEMATA}} marker. When the template is processed, MySQL Workbench will note the section and iterate the section until the variable data for {{SCHEMA_NAME}} in the corresponding data dictionary is exhausted. For example, if the model being processed contains two schemata, the output for the section might resemble the following:

```
Schema: Airlines
Schema: Airports
```

That is, the model contains two schemata, Airlines and Airports.

Data Dictionaries

It is important to understand the relationship between sections and data dictionaries in more detail. In a data dictionary the *key* for a variable is the variable name, a marker. The variable *value* is the variable's data. The entry for a section in a data dictionary is different. For a section entry in a data dictionary, the key is the section name, the marker. However, the value associated with the key is a list of data dictionaries. In MySQL Workbench each section is usually associated with a data dictionary. You can think of a section as *activating* its associated dictionary (or dictionaries).

When a template is processed, data dictionaries are loaded in a hierarchical pattern, forming a tree of data dictionaries. This is illustrated by the following table:

Data Dictionary	Loads Data Dictionary
MAIN	SCHEMATA
SCHEMATA	TABLES, COLUMNS (Detailed is true), FOREIGN_KEYS (Detailed is true), INDICES (Detailed is true)
TABLES	REL_LISTING, INDICES_LISTING, COLUMNS_LISTING, TABLE_COMMENT_LISTING, DDL_LISTING

COLUMNS_LISTING	COLUMNS (Detailed is false)
REL_LISTING	REL (Detailed is false)
INDICES_LISTING	INDICES (Detailed is false)

The root of the tree is the *main* dictionary. Additional dictionaries are then loaded from the root to form the dictionary tree.

Note

If a template has no sections in it, then any variables used in the template will be looked up in the main dictionary. If a variable is not found in the main dictionary (which can be thought of as associated with the default, or main, section) then no data will be generated in the output file for that marker.

Evaluation of variables

The tree structure of the data dictionaries is important when it comes to evaluation of variables. As variables are defined in data dictionaries, their associated value only has meaning when that particular data dictionary is active, and that means when the section associated with that data dictionary is active. When a variable lookup occurs, the system will check the data dictionary associated with the current section. If the variable value can be found there the replacement is made. However, if the variable's value is not found in the current data dictionary then the parent data dictionary will be checked for the variable's value and so on up the tree until the main data dictionary, or root, is reached.

This can best be illustrated by an example. Assume we want to display the names of all columns in a model. Consider the following template as an attempt to achieve this:

```
Report
-----
Column Name: {{COLUMN_NAME}}
```

This template will produce no output, even for a model that contains many columns. In this example the only data dictionary active is the main dictionary. `COLUMN_NAME` however is stored in the `COLUMNS` data dictionary, which is associated with the `COLUMNS` section.

With this knowledge the template can be improved as follows:

```
Report
-----
{{#COLUMNS}}
Column Name: {{COLUMN_NAME}}
{{/COLUMNS}}
```

This still does not produce output. Referring to the table [Data Dictionary Hierarchy Tree](#) explains why. The `COLUMNS` data dictionary has the parent dictionary `COLUMNS_LISTING`. `COLUMNS_LISTING` has the parent `TABLES`, which has the parent `SCHEMATA`, whose parent is the main dictionary. Remember in order for a dictionary to be involved in variable lookup, its associated section must currently be active.

So in order to achieve the desired output we would need the template to be something like the following:

```
Report
-----
{{#SCHEMATA}}
{{#TABLES}}
{{#COLUMNS_LISTING}}
{{#COLUMNS}}
Column Name: {{COLUMN_NAME}}
{{/COLUMNS}}
{{/COLUMNS_LISTING}}
{{/TABLES}}
{{/SCHEMATA}}
```

The following template is the same, but with explanatory comments added:

```
Report
-----
{{! Main dictionary active}}
{{#SCHEMATA}} {{! SCHEMATA dictionary active}}
{{#TABLES}} {{! TABLES dictionary active}}
{{#COLUMNS_LISTING}} {{! COLUMNS_LISTING dictionary active}}
{{#COLUMNS}} {{! COLUMNS dictionary active}}
Column Name: {{COLUMN_NAME}} {{! COLUMN_NAME variable is looked-up, and found, in COLUMNS data dictionary}}
```

```
{{/COLUMNS}}
{/COLUMNS_LISTING}}
{/TABLES}}
{/SCHEMATA}}
```

Imagine now that for each column name displayed you also wanted to display its corresponding schema name, the template would look like this:

```
Report
-----
{{#SCHEMATA}}
{{#TABLES}}
{{#COLUMNS_LISTING}}
{{#COLUMNS}}
Schema Name: {{SCHEMA_NAME}} Column Name: {{COLUMN_NAME}}
{{/COLUMNS}}
{{/COLUMNS_LISTING}}
{{/TABLES}}
{{/SCHEMATA}}
```

When variable lookup is performed for `SCHEMA_NAME` the `COLUMNS` dictionary will be checked. As the variable is not found there the parent dictionary will be checked, `COLUMNS_LISTING`, and so on until the variable is eventually found, where it is held, in the `SCHEMATA` dictionary.

If there are multiple schemata in the model the outer section will be iterated over a matching number of times, and `SCHEMA_NAME` will accordingly have the correct value on each iteration.

It's important to always consider which dictionary needs to be active (and which parents) for a variable to be evaluated correctly. In the following section you will find a table that helps you identify section requirements.

7.9.1. Supported Template Markers

A list of supported markers follows. These markers can be used in any template, including custom templates.

Marker text	Type	Data Dictionary defined in (if variable) or parent dictionary (if section)	Corresponding data
TITLE	Variable	MAIN	Title of the report
GENERATED	Variable	MAIN	Date and time that the report was generated
STYLE_NAME	Variable	MAIN	The name of the style selected in MySQL Workbench. This is typically used to load the corresponding CSS file, depending on the name of the style selected in MySQL Workbench.
SCHEMA_COUNT	Variable	MAIN	The total number of schemata in the model
PROJECT_TITLE	Variable	MAIN	Project title as set for the model in <u>DOCUMENT PROPERTIES</u> .
PROJECT_NAME	Variable	MAIN	Project name as set for the model in <u>DOCUMENT PROPERTIES</u> .
PROJECT_AUTHOR	Variable	MAIN	Project author as set for the model in <u>DOCUMENT PROPERTIES</u> .
PROJECT_VERSION	Variable	MAIN	Project version as set for the model in <u>DOCUMENT PROPERTIES</u> .
PROJECT_DESCRIPTION	Variable	MAIN	Project description as set for the model in <u>DOCUMENT PROPERTIES</u> .
PROJECT_CREATED	Variable	MAIN	Automatically set for the model project, but as displayed in <u>DOCUMENT PROPERTIES</u> .
PROJECT_CHANGED	Variable	MAIN	Automatically set for the model project, but as displayed in <u>DOCUMENT PROPERTIES</u> .
TOTAL_TABLE_COUNT	Variable	MAIN	Total number of tables in all schemata contained in the model.
TOTAL_COLUMN_COUNT	Variable	MAIN	Total number of columns in all tables in all schemata in the model.

TOTAL_INDEX_COUNT	Variable	MAIN	Total number of indexes in the model.
TOTAL_FK_COUNT	Variable	MAIN	Total number of foreign keys in the model.
SCHEMATA	Section	MAIN	Used to mark the start and end of a SCHEMATA section. The SCHEMATA data dictionary becomes active in this section.
SCHEMA_NAME	Variable	SCHEMATA	The name of a schema in the model
SCHEMA_ID	Variable	SCHEMATA	The ID of the schema
TABLE_COUNT	Variable	SCHEMATA	Total total number of tables in the current schema
COLUMN_COUNT	Variable	SCHEMATA	The total number of columns in the current schema
INDICES_COUNT	Variable	SCHEMATA	The total number of indexes in the current schema
FOR_EIGN_KEYS_COUNT	Variable	SCHEMATA	The total number of foreign keys in the current schema
TABLES	Section	SCHEMATA	Marks the start and end of a TABLES section. The TABLES data dictionary becomes active in this section.
TABLE_NAME	Variable	TABLES	The table name
TABLE_ID	Variable	TABLES	The table ID
COLUMNS_LISTING	Section	TABLES	Marks the start and end of a COLUMNS_LISTING section. The COLUMNS_LISTING data dictionary becomes active in this section.
COLUMNS	Section	COLUMNS_LISTING	Marks the start and end of a COLUMNS section. The COLUMNS data dictionary becomes active in this section.
COLUMN_KEY	Variable	COLUMNS	Whether the column is a primary key or not
COLUMN_NAME	Variable	COLUMNS	The name of the column
COLUMN_DATATYPE	Variable	COLUMNS	The data type of the column
COLUMN_NOTNULL	Variable	COLUMNS	Whether or not the column allows NULLs
COLUMN_DEFAULT_VALUE	Variable	COLUMNS	The default value of entries in this column
COLUMN_COMMENT	Variable	COLUMNS	Any comment for this column
COLUMN_ID	Variable	COLUMNS	The ID of the column
COLUMN_KEY_PART	Variable	COLUMNS (if detailed)	The type of column key
COLUMN_NULLABLE	Variable	COLUMNS (if detailed)	Can the column contain NULLs
COLUMN_AUTO_INC	Variable	COLUMNS (if detailed)	Does the column auto-increment
COLUMN_CHARSET	Variable	COLUMNS (if detailed)	The character set used by the column
COLUMN_COLLATION	Variable	COLUMNS (if detailed)	The collation used by the column
COLUMN_IS_USERTYPE	Variable	COLUMNS (if detailed)	Whether or not the column is a user type
INDICES_LISTING	Section	TABLES	Marks the start and end of an INDICES_LISTING section. The INDICES_LISTING data dictionary becomes active in this section.
INDICES	Section	INDICES_LISTING	Marks the start and end of an INDICES section. The INDICES data dictionary becomes active in this section.
INDEX_NAME	Variable	INDICES	The name of the index
INDEX_PRIMARY	Variable	INDICES	Is this a primary index
INDEX_UNIQUE	Variable	INDICES	Is this a unique index
INDEX_TYPE	Variable	INDICES	The type of index e.g. PRIMARY
INDEX_KIND	Variable	INDICES	The kind of index

INDEX_COMMENT	Variable	INDICES	A comment for the index
INDEX_ID	Variable	INDICES	The ID of the index
INDEX_COLUMNS	Section	INDICES	Marks the start and end of an INDEX_COLUMNS section. The INDEX_COLUMNS data dictionary becomes active in this section.
INDEX_COLUMN_NAME	Variable	INDEX_COLUMNS	The index column name
INDEX_COLUMN_ORDER	Variable	INDEX_COLUMNS	The index column order e.g. ascending, descending
INDEX_COLUMN_COMMENT	Variable	INDEX_COLUMNS	A comment for the index column
INDEX_KEY_BLOCK_SIZE	Variable	INDEX_COLUMNS (if detailed)	Index key block size
REL_LISTING	Section	TABLES	Marks the start and end of a REL_LISTING section. The REL_LISTING data dictionary becomes active in this section.
REL	Section	REL_LISTING	Marks the start and end of a REL section. The REL data dictionary becomes active in this section.
REL_NAME	Variable	REL, FOREIGN_KEYS	Relationship name
REL_TYPE	Variable	REL, FOREIGN_KEYS	Relationship type
REL_PARENTTABLE	Variable	REL, FOREIGN_KEYS	Parent table of relationship
REL_CHILDTABLE	Variable	REL, FOREIGN_KEYS	Child table of relationship
REL_CARD	Variable	REL, FOREIGN_KEYS	Relationship card
FOREIGN_KEY_ID	Variable	REL	Foreign key ID
FOREIGN_KEYS	Section	SCHEMATA	Marks the start and end of a FOREIGN_KEYS section. The FOREIGN_KEYS data dictionary becomes active in this section.
FK_DELETE_RULE	Variable	FOREIGN_KEYS	Foreign key delete rule
FK_UPDATE_RULE	Variable	FOREIGN_KEYS	Foreign key update rule
FK_MANDATORY	Variable	FOREIGN_KEYS	Foreign key mandatory
TABLE_COMMENT	Section	TABLES	Marks the start and end of a TABLE_COMMENT_LISTING section. The TABLE_COMMENT_LISTING data dictionary becomes active in this section.
DDL_LISTING	Section	TABLES	Marks the start and end of a DDL_LISTING section. The DDL_LISTING data dictionary becomes active in this section.
DDL_SCRIPT	Variable	DDL_LISTING	Display the DDL script of the currently active entity e.g. SCHEMATA, TABLES

Using the table

The table shows which variables are defined in which sections. The variable should be used in its correct section, otherwise its value will not be displayed.

Note

It should be remembered though that the data dictionaries used to perform the lookup form a hierarchical tree, so it is possible to use a variable defined in a parent section, in a child section.

7.9.2. Creating a custom template

In the simplest case a template consists of two files. A template file, which has a `.tpl` extension, and a special file `info.xml`. The `info.xml` file has important metadata about the template. A third file is optional, that is the preview image file. This preview file provides a thumbnail image illustrating the appearance of the generated report.

One of the easiest ways to create a custom template is to make a copy of any existing template.

For example, you make a custom template based on the `Text_Basic`. The following procedure demonstrates this.

1. First you need to make a copy of the template on which you are going to base your custom template. To do this navigate to the folder where the templates are stored. Assuming MySQL Workbench has been installed into the default location on Windows, this would be `C:\Program Files\MySQL\MySQL Workbench 5.0\SE\modules\data\wb_model_reporting`.
2. Then make a copy of the template folder you wish to base your new template on. In this case a copy of the `Text_Basic.tpl` folder is made. The copy can be given any suitable name, for example, `Custom_Basic.tpl`.
3. Now the `info.xml` file needs to be edited, to reflect your custom template. The unedited file in this case is shown here:

```
<?xml version="1.0"?>
<data>
  <value type="object" struct-name="workbench.model.reporting.TemplateInfo"
    id="{BD6879ED-814C-4CA3-A869-9864F83B88DF}" struct-checksum="0xb46b524d">
    <value type="string" key="description">A basic TEXT report listing schemata and objects.</value>
    <value type="string" key="name">HTML Basic Frame Report</value>
    <value type="list" content-type="object"
      content-struct-name="workbench.model.reporting.TemplateStyleInfo"
      key="styles">
      <value type="object" struct-name="workbench.model.reporting.TemplateStyleInfo"
        id="{7550655C-CD4B-4EB1-8FAB-AAEE49B2261E}" struct-checksum="0xab08451b">
        <value type="string" key="description">Designed to be viewed with a fixed sized font.</value>
        <value type="string" key="name">Fixed Size Font</value>
        <value type="string" key="previewImageFileName">preview_basic.png</value>
        <value type="string" key="styleTagValue">fixed</value>
      </value>
    </value>
    <value type="string" key="mainFileName">report.txt</value>
  </value>
</data>
```

Two objects are defined in the file. The `TemplateInfo` object and the `TemplateStyleInfo` object. These objects contain information about the template that will be displayed in the DBDoc Model Reporting wizard main screen.

4. The first thing you need to change are the object GUIDs that are used in the file. In this example there are two that need replacing:

```
id="{BD6879ED-814C-4CA3-A869-9864F83B88DF}"
...
id="{7550655C-CD4B-4EB1-8FAB-AAEE49B2261E}"
```

Generate two new GUIDS. This can be done using any suitable command-line tool. There are also free online tools that can be used to generate GUIDs. The `info.xml` file should then be edited accordingly.

5. Edit the textual information for the `TemplateInfo` and `TemplateStyleInfo` objects to reflect the purpose of the custom template.
6. The modified file will now look something like the following:

```
<?xml version="1.0"?>
<data>
  <value type="object" struct-name="workbench.model.reporting.TemplateInfo"
    id="{cac9ba3f-ee2a-49f0-b5f6-32580fab1640}" struct-checksum="0xb46b524d">
    <value type="string"
      key="description">Custom basic TEXT report listing schemata and objects.</value>
    <value type="string" key="name">Custom Basic text report</value>
    <value type="list" content-type="object"
      content-struct-name="workbench.model.reporting.TemplateStyleInfo" key="styles">
      <value type="object"
        struct-name="workbench.model.reporting.TemplateStyleInfo"
        id="{39e3b767-a832-4016-8753-b4cb93aa2dd6}" struct-checksum="0xab08451b">
        <value type="string" key="description">Designed to be viewed with a fixed sized font.</value>
        <value type="string" key="name">Fixed Size Font</value>
        <value type="string" key="previewImageFileName">preview_basic.png</value>
        <value type="string" key="styleTagValue">fixed</value>
      </value>
    </value>
    <value type="string" key="mainFileName">custom_report.txt</value>
  </value>
</data>
```

7. The next step is to create the new template file. Again this may best be achieved, depending on your requirements, by editing

an existing template. In this example the template file `report.txt.tpl` is shown here:

```
+-----+  
| MySQL Workbench Report |  
+-----+  
  
Total number of Schemata: {{SCHEMA_COUNT}}  
======{{#SCHEMATA}}  
{{SCHEMA_NR}}. Schema: {{SCHEMA_NAME}}  
-----  
## Tables {{TABLE_COUNT}} ##  
{{#TABLES}}{{TABLE_NR_FMT}}. Table: {{TABLE_NAME}}  
{{#COLUMNS_LISTING}}## Columns ##  
Key Column Name Datatype Not Null Default Comment  
{{#COLUMNS}}{{COLUMN_KEY}}{{COLUMN_NAME}}{{COLUMN_DATATYPE}} »  
{{COLUMN_NOTNULL}}{{COLUMN_DEFAULTVALUE}}{{COLUMN_COMMENT}}  
{{/COLUMNS}}{{/COLUMNS_LISTING}}  
{{#INDICES_LISTING}}## Indices ##  
Index Name Columns Primary Unique Type Kind Comment  
{{#INDICES}}{{INDEX_NAME}}{{#INDICES_COLUMNS}}{{INDEX_COLUMN_NAME}} »  
{{INDEX_COLUMN_ORDER}}{{INDEX_COLUMN_COMMENT}}{{/INDICES_COLUMNS}} »  
{{INDEX_PRIMARY}}{{INDEX_UNIQUE}}{{INDEX_TYPE}}{{INDEX_KIND}}{{INDEX_COMMENT}}  
{{/INDICES}}{{/INDICES_LISTING}}  
{{#REL_LISTING}}## Relationships ##  
Relationship Name Relationship Type Parent Table Child Table Cardinality  
{{#REL}}{{REL_NAME}}{{REL_TYPE}}{{REL_PARENTTABLE}}{{REL_CHILDTABLE}}{{REL_CARD}}  
{{/REL}}{{/REL_LISTING}}  
-----  
{{/TABLES}}  
{{/SCHEMATA}}  
=====  
End of MySQL Workbench Report
```

This template shows details for all schemata in the model.

8. The above template file can be edited in any way you like, with new markers being added, and existing markers being removed as required. For the custom template example you might want to create a much simpler template. Such as the one following:

```
+-----+  
| MySQL Workbench Custom Report |  
+-----+  
  
Total number of Schemata: {{SCHEMA_COUNT}}  
======{{#SCHEMATA}}  
Schema Name: {{SCHEMA_NAME}}  
-----  
## Tables {{TABLE_COUNT}} ##  
{{#TABLES}}  
Table Name: {{TABLE_NAME}}  
{{/TABLES}}  
{{/SCHEMATA}}  
  
Report Generated On: {{GENERATED}}  
=====  
End of MySQL Workbench Custom Report
```

This simplified report just lists the schemata and the tables in a model. The date and time the report was generated will also be displayed as a result of the use of the `{{GENERATED}}` variable.

9. The custom template can then be tested. Start MySQL Workbench, load the model to generate the report for, select the **MODEL, DBDOC - MODEL REPORTING** menu item. Then select the new custom template from the list of available templates, select an output directory and then click FINISH to generate the report. Finally, navigate to the output directory to view the finished report.

Chapter 8. Server Administration

Since version 5.2.6 MySQL Workbench has included functionality for managing server instances. A server instance is created to provide a way of connecting to a server to be managed. The first step then is to create a server instance if none exists, or to work with an existing server instance.

MySQL Workbench also provides functionality to administer and configure a server via these server instances. Thus, the Server Administrator functionality can be broadly grouped into two main areas:

1. Creating and managing server instances
2. Administration and configuration functions via a server instance

Each of these broad areas is each discussed in more detail in the following sections.

8.1. Server Instances

In the Workspace section of the Home screen is an area for Server Administration tasks. This section of the Workspace has the following action items:

1. Server Administration
2. Server Administration (icon)
3. New Server Instance
4. Manage Data Import/Export
5. Manage Security
6. Manage Server Instances

Server Administration

The purpose of this action item is to allow you to quickly connect to a predefined server instance, and carry out administration functions on the associated server. Clicking this item launches the Security Manager dialog, from which you can select the server instance you wish to connect to. A new Admin tab will be launched, which displays the Server Status and Configuration.

Server Administration (icon)

If you have already created server instances, then you can most quickly launch these by clicking on the icon for the Server Instance you wish to access. A new Admin tab will be launched, which displays Server Status and Configuration.

New Server Instance

This action item allows you to create a new server instance. A server instance is primarily a combination of connection and configuration details for a specific server that you wish to manage. When you click this item a wizard is launched that allows you to specify the connection and various other configuration parameters. After completion of the wizard, a new Admin tab is launched, which displays Server Status and Configuration.

Manage Data Import/Export

The purpose of this action item is to allow you to create a dump file from a database, or restore data from a file to a live database. Clicking this item launches the Import/Export MySQL Data wizard. This allows you to select a server instance to connect to.

Manage Security

The purpose of this action item is to take you quickly to the screen that allows you to manage user accounts. It simply launches an Admin page and locates you on the Accounts tab.

Manage Server Instances

Clicking this action item launches the Manage Server Instances dialog. Within this dialog you can change the configuration of existing server instances, or create a new server instance.

This topic is discussed in more depth in the section [Section 8.1.1, “Creating and Managing Server Instances”](#).

8.1.1. Creating and Managing Server Instances

Server instances can be created and managed from the **HOME** page. As mentioned in the previous section, there are several ways new server instances can be created:

1. By clicking the New Server Instance action item from the Server Administration section of the Home screen. This launches the **CREATE A NEW SERVER INSTANCE** wizard.
2. By clicking the Manage Server Instances action item from the Server Administration section of the Home screen. This launches the **MANAGE SERVER INSTANCES** dialog, from within which a new server instance can be created.

Each of these two options is now described in turn.

8.1.1.1. New Server Instance Wizard

Clicking the New Server Instance action item launches the **CREATE A NEW SERVER INSTANCE** wizard. The wizard provides a step-by-step approach to creating a new server instance. This is most suitable for beginners. Once some familiarity is achieved with the various settings and parameters required, a new instance can also be quickly created from the Manage Server Instances dialog discussed later.

The steps presented in the wizard are as follows:

1. Specify host machine
2. Operating system
3. Host SSH Connection
4. Test Settings
5. Database Connection
6. Test DB Connection
7. MySQL Config File
8. Specify Commands
9. Complete Setup
10. Results

Specify host machine

On this page you can select **LOCALHOST** if you intend to manage a server on your local machine. Or you can select **REMOTE HOST**. In this latter case you will need to provide the IP address or the network name of the remote server. Click **NEXT** to continue.

Operating system

On this page you simply select the operating system that applies to your server. You should also ensure that the option you select from the drop-down listbox also reflects your MySQL Server version and installation type, for example, Windows (MySQL 5.1 Installer Package).

Host SSH Connection

If you specified a Remote Host on the Specify Host Machine page, you will be presented with the Host SSH Connection page. This page allows you to enable SSH for the login to the server instance. This facility allows you to create a secure connection to remotely administer and configure the server instance. You need to enter the username and password of the account that will be used to log in to the server for administration and configuration activities. If you do not enter a password, you will be prompted for the password when the connection is established by MySQL Workbench. You can optionally specify the path to your SSH key for use with the server, rather than enter a username and password.

Note

This connection is to allow remote administration and configuration of the MySQL Server itself. It is not the same as the connection used to connect to a server for general database manipulation.

Note

If managing a remote server, you will need to use an SSH connection type if you wish to start or stop the server or edit its configuration file. Other administrative functions do not require an SSH connection type.

Test Settings

On the next page your settings will be tested. The wizard will report back the results of attempting to connect to the server. If an error occurs you will be directed to view the logs, which can be done by clicking the SHOW LOGS button.

Database Connection

This page allows you to select a connection to a specific database. The settings entered previously have been concerned with the connection to the server required for administrative purposes. This page is concerned with connection to a specific database. You can either launch the Manage DB Connections dialog or select a pre-existing connection from a drop-down listbox. The former is most useful if you have not created any connections. If you do need to create a connection at this point then refer to [Section 6.1, "Manage Connections"](#). Once a connection has been selected click NEXT to continue.

Test DB Connection

On this page your database connection will be tested and results displayed. If an error occurred you will be directed to view the logs. This can be achieved by clicking the SHOW LOGS button.

MySQL Config File

In order for MySQL Server configuration information to be fetched and displayed in MySQL Workbench the location of the MySQL Server configuration file needs to be known. The wizard will be able to determine the most likely location of the configuration file, based on the selection made on the Operating System page of the wizard. However, it is possible to test that this information is correct by clicking the CHECK PATH and CHECK SECTION buttons. The wizard will then report if the configuration file and server configuration section can in fact be accessed. It is also possible to manually enter the location of the configuration file, and the section pertaining to MySQL Server data, but again these manually entered values should be tested using the buttons provided. Click the NEXT button to continue.

Specify Commands

This page allows you to set the commands required to start, stop and check the status of the running server instance. These fields will have defaults set based on the option selected on the Operating System page of the wizard. It is possible to customize the commands if required, but the defaults will be suitable in most cases. Click NEXT to continue.

Complete Setup

On this page you finally assign a name to the server instance. This name is used in various parts of the GUI to allow you to recall this instance. After setting a suitable name, click NEXT to continue.

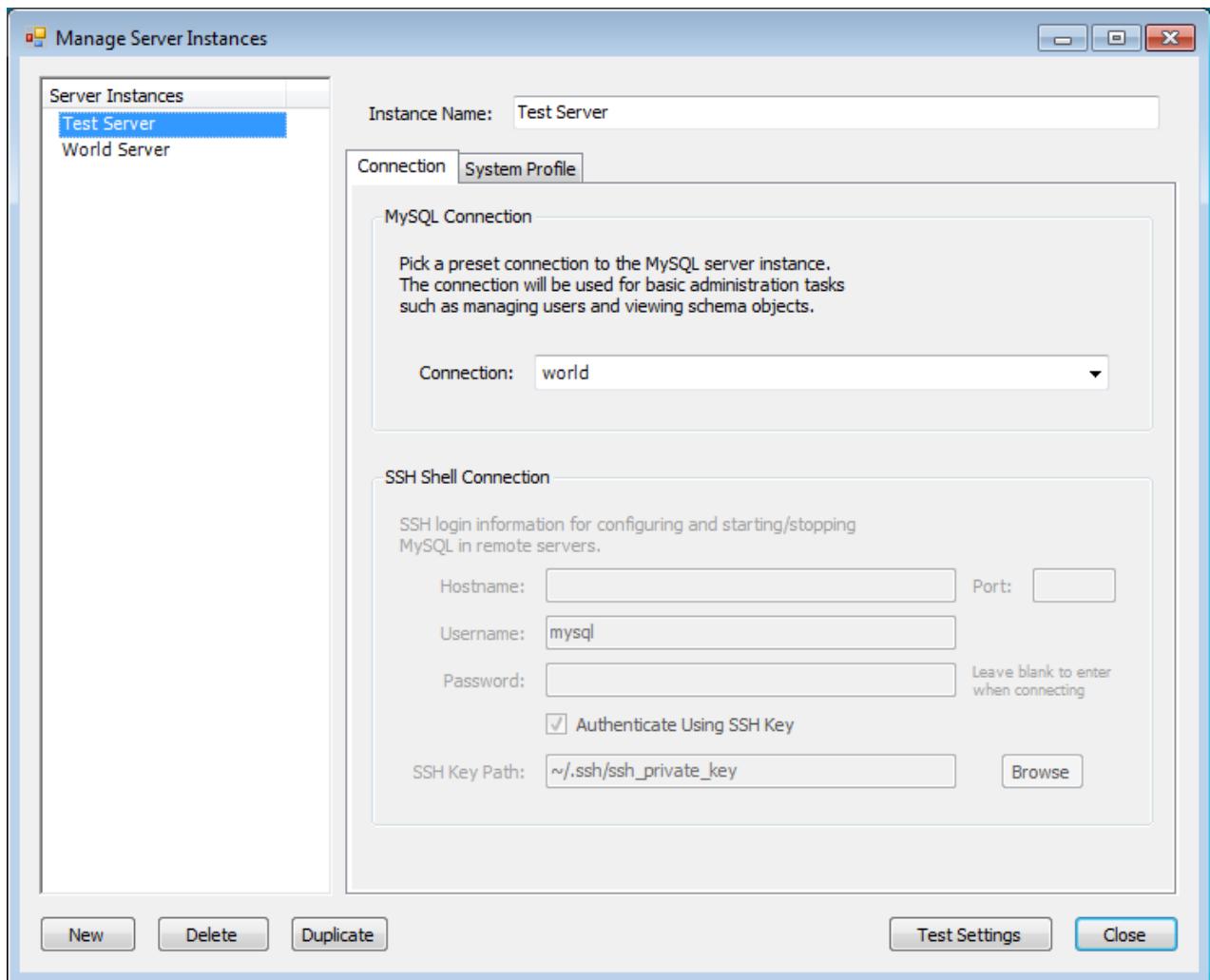
Results

This is the final page of the wizard. It simply summarizes the settings you have selected. If everything looks good click FINISH to create the server instance and exit the wizard.

8.1.1.2. Manage Server Instances Dialog

The Manage Server Instances dialog allows you to create, delete and manage server instances. The CONNECTION tab of the wizard allows you to select a predefined connection to connect with a server to be managed. In addition, it is also possible to connect to a remote server using an SSH connection.

Figure 8.1. Manage Server Instances Dialog



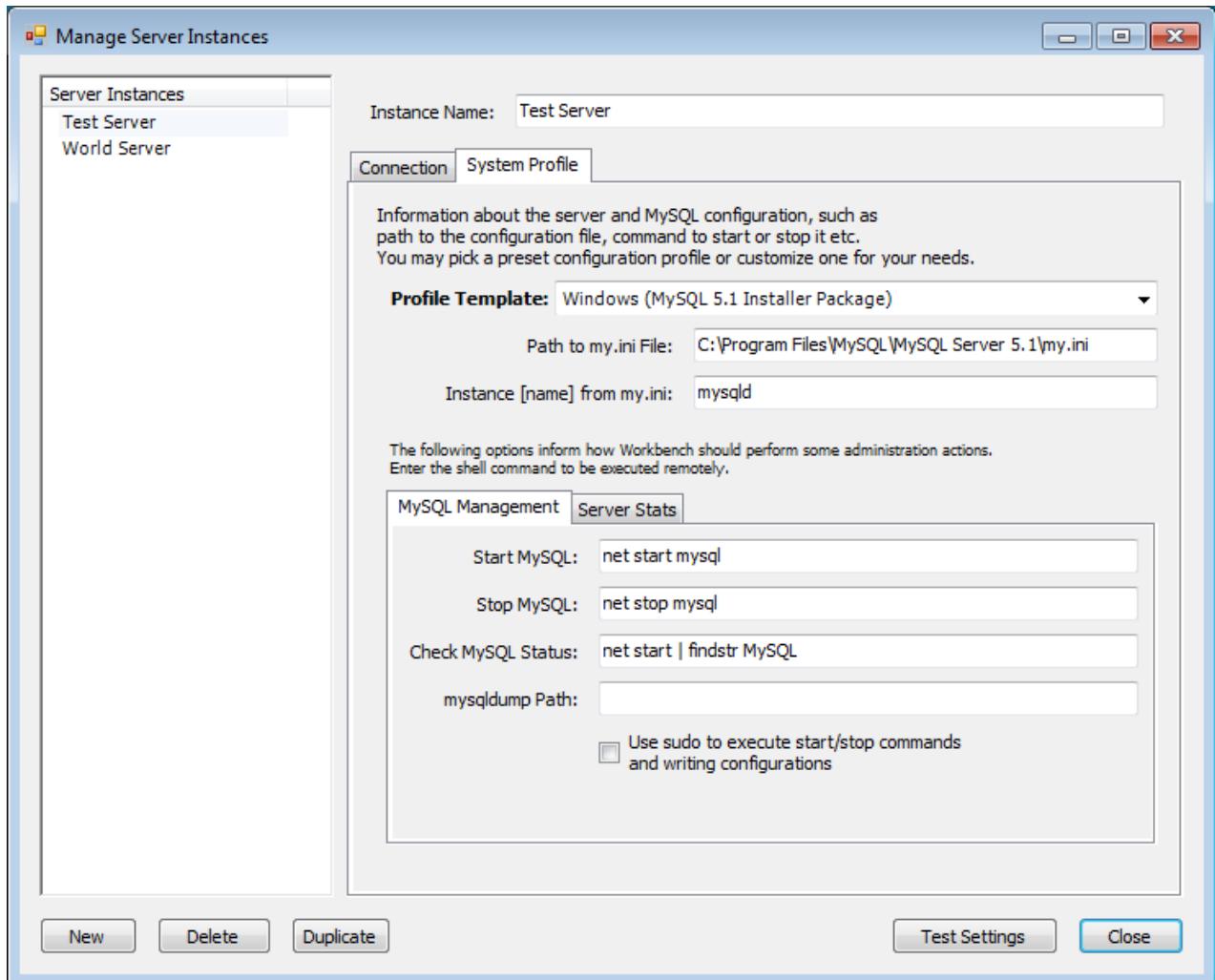
The **SYSTEM PROFILE** tab of the wizard allows you to specify server specific information. This is primarily achieved through selecting a Profile Template. A Profile Template contains standard information used in managing the server instance. The following Profile Templates are available:

- Fedora Linux (MySQL Package)
- Fedora Linux (Vendor Package)
- FreeBSD (MySQL Package)
- Generic Linux (MySQL tar package)
- Mac OS X (MySQL Package)
- OpenSolaris (MySQL Package)
- RHEL (MySQL Package)
- SLES (MySQL Package)
- Ubuntu Linux (MySQL Package)
- Ubuntu Linux (Vendor Package)
- Windows (MySQL 5.0 Installer Package)
- Windows (MySQL 5.1 Installer Package)
- Windows (MySQL zip package)

- Custom

Once a profile is selected a number of default parameters will be set, including commands used to start and stop MySQL, commands to check server status and the location of the `my.ini` configuration file.

Figure 8.2. Server Instances Wizard



After an instance has been created it can be launched by double-clicking its icon in the **SERVER ADMINISTRATION** panel of the **HOME** page. This creates an Admin page. The Admin page has two main panels, the **SERVER STATUS** panel, and the **CONFIGURATION** panel. The **CONFIGURATION** panel features multiple tabs: **STARTUP**, **CONFIGURATION**, **ACCOUNTS**, **CONNECTIONS**, **VARIABLES**, **DATA DUMP**, and **LOGS**.

8.2. Server Administration and Configuration

The functionality included in MySQL Workbench for administering servers is similar to that formerly provided by MySQL Administrator. The Administrator functionality in MySQL Workbench is grouped into several tabs:

- **Startup** - Allows you to start and stop the MySQL server, and view the startup message log.
- **Configuration** - Allows you to view and edit the MySQL Configuration file (`my.ini`) using GUI controls.
- **Accounts** - Allows you to create user accounts and assign roles and privileges.
- **Connections** - Displays connections to MySQL Server.
- **Variables** - Displays server and status variables.

- **Data Dump** - Import and export of data.
- **Logs** - Displays server log file entries.

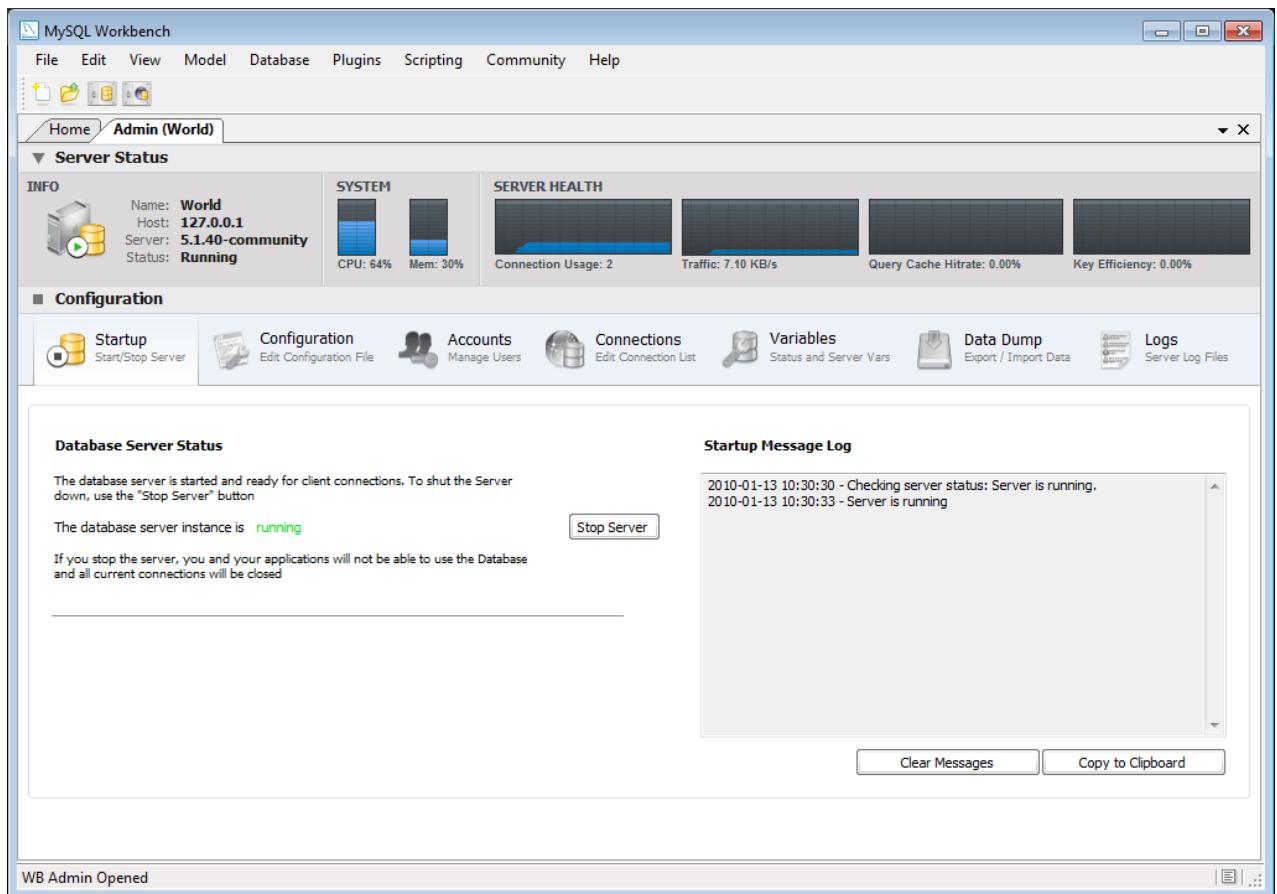
The Administrator also displays system and server status. System status displayed includes:

- CPU utilization
- Memory usage
- Connection Health

For server health the following are displayed:

- Connection Usage
- Traffic
- Query Cache Hit Rate
- Key Efficiency

Figure 8.3. MySQL Workbench - Admin page

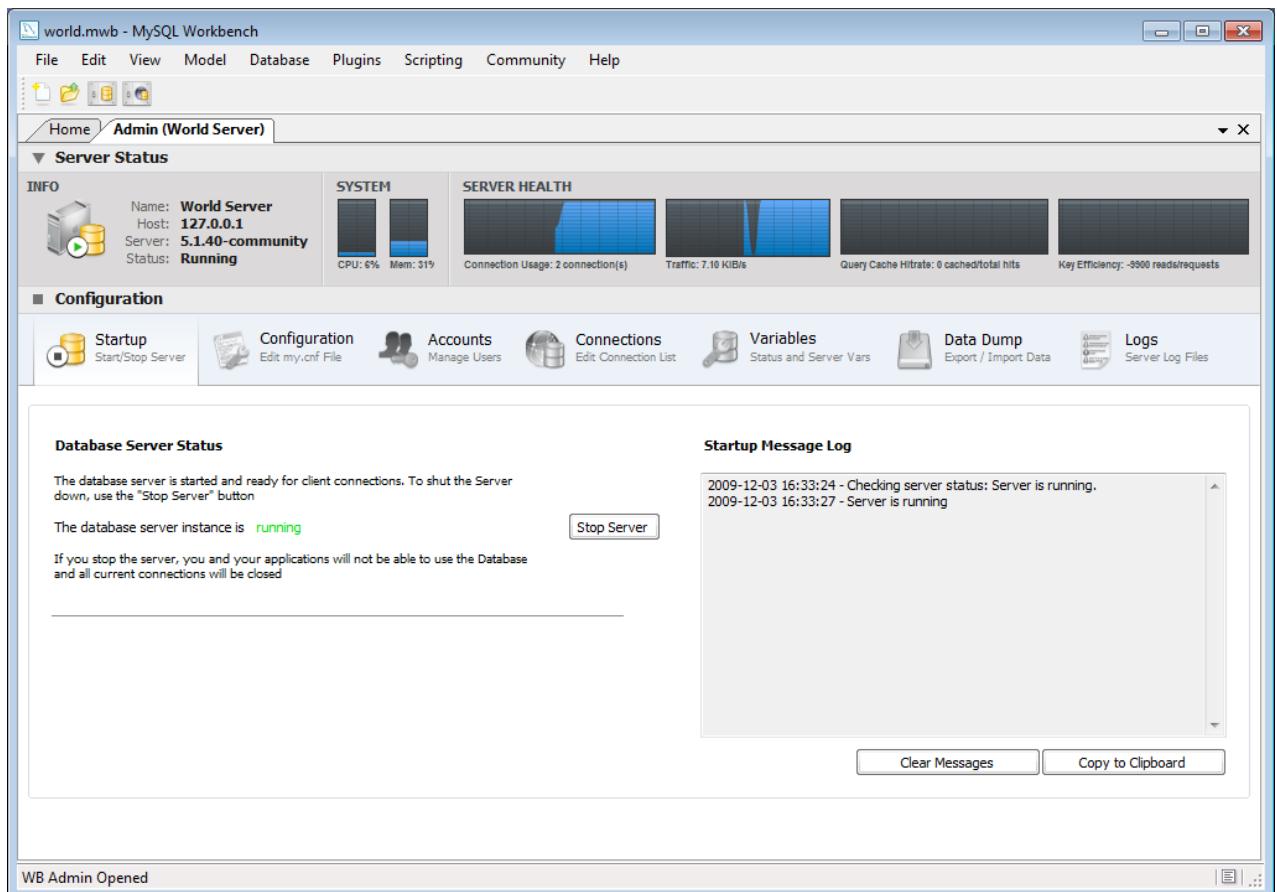


8.2.1. Startup Tab

The Startup tab has several purposes:

- Displaying database server status.
- Start up and shut down the server.
- Displaying the Startup Message log.
- The ability to select whether the server starts up when the system starts up.

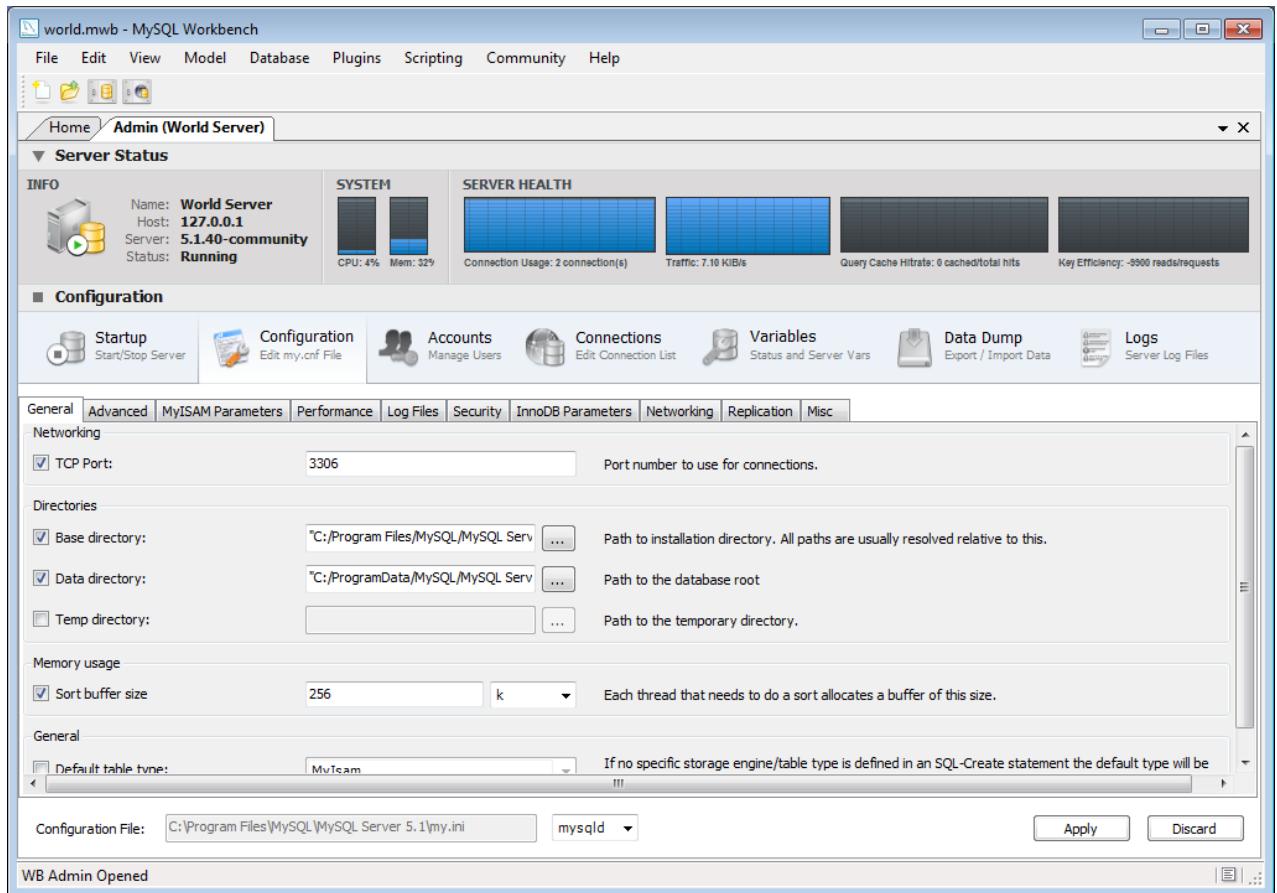
Figure 8.4. Administrator - Startup tab



8.2.2. Configuration tab

The configuration tab allows you to edit the `my.ini` configuration file through selecting checkboxes and other GUI controls. This tab also features a number of sub-tabs, which provide access to various sub-sections within the configuration file. The sub-tabs are:

- General
- MyISAM Parameters
- InnoDB Parameters
- Performance
- Log Files
- Replication
- Networking
- Security
- Advanced

Figure 8.5. Administrator - Configuration tab

8.2.3. Accounts tab

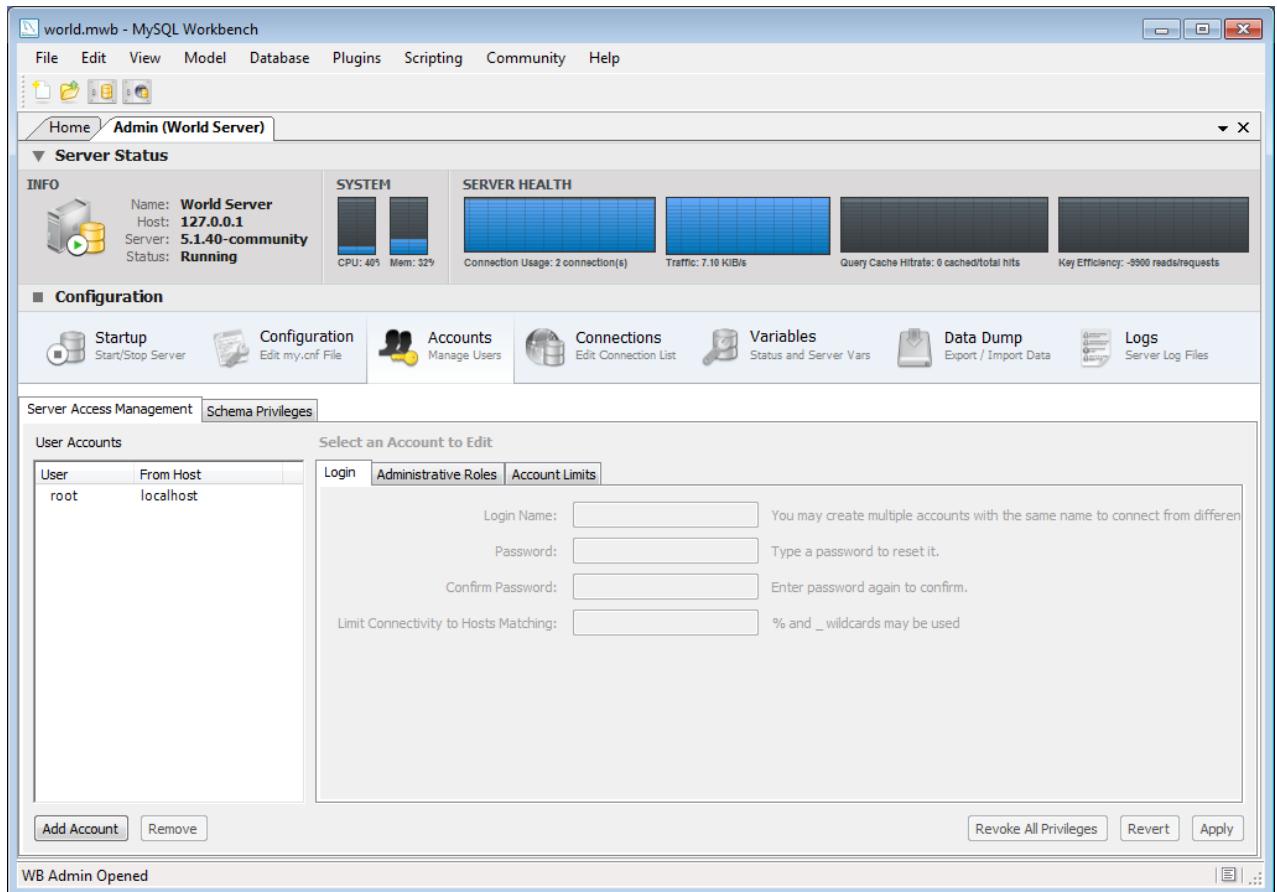
The Accounts tab has two sub-tabs:

- Server Access Management
- Schema Privileges

SERVER ACCESS MANAGEMENT allows you to list existing user accounts. You can also add and delete accounts. You can allocate administrative roles and also set account limits.

SCHEMA PRIVILEGES allows you to set specific privileges on a user basis.

Figure 8.6. Administrator - Accounts tab



Note

In the current version of MySQL Workbench it is not possible to manage privileges below the schema level. For example, it is not possible to view or manage grants at the table, column, or procedure level. Support for this feature is however planned for MySQL Workbench 6.0.

8.2.3.1. Administrative Roles

To aid in the assignment of privileges to MySQL Server users, MySQL Workbench introduces the concept of Administrative Roles. Roles are a quick way of granting a number of privileges to a user, based on the work the user needs to carry out on the server. It is also possible to assign multiple roles to a user. To assign roles, click on the User Account you wish to modify, and then click on the **ADMINISTRATIVE ROLES** tab. Then click the checkboxes according to the roles you wish to allocate to the user. Note once you select a role to a user you will see the accumulated privileges in the **GLOBAL PRIVILEGES ASSIGNED TO USER** panel. For example, if you select the role **BackupAdmin** the privileges granted would include **EVENT, LOCK TABLES, SELECT, SHOW DATABASES**. Then if you additionally select the role of **ReplicationAdmin**, the list of privileges will be expanded to also include **REPLICATION CLIENT, REPLICATION SLAVE** and **SUPER**.

The roles available are:

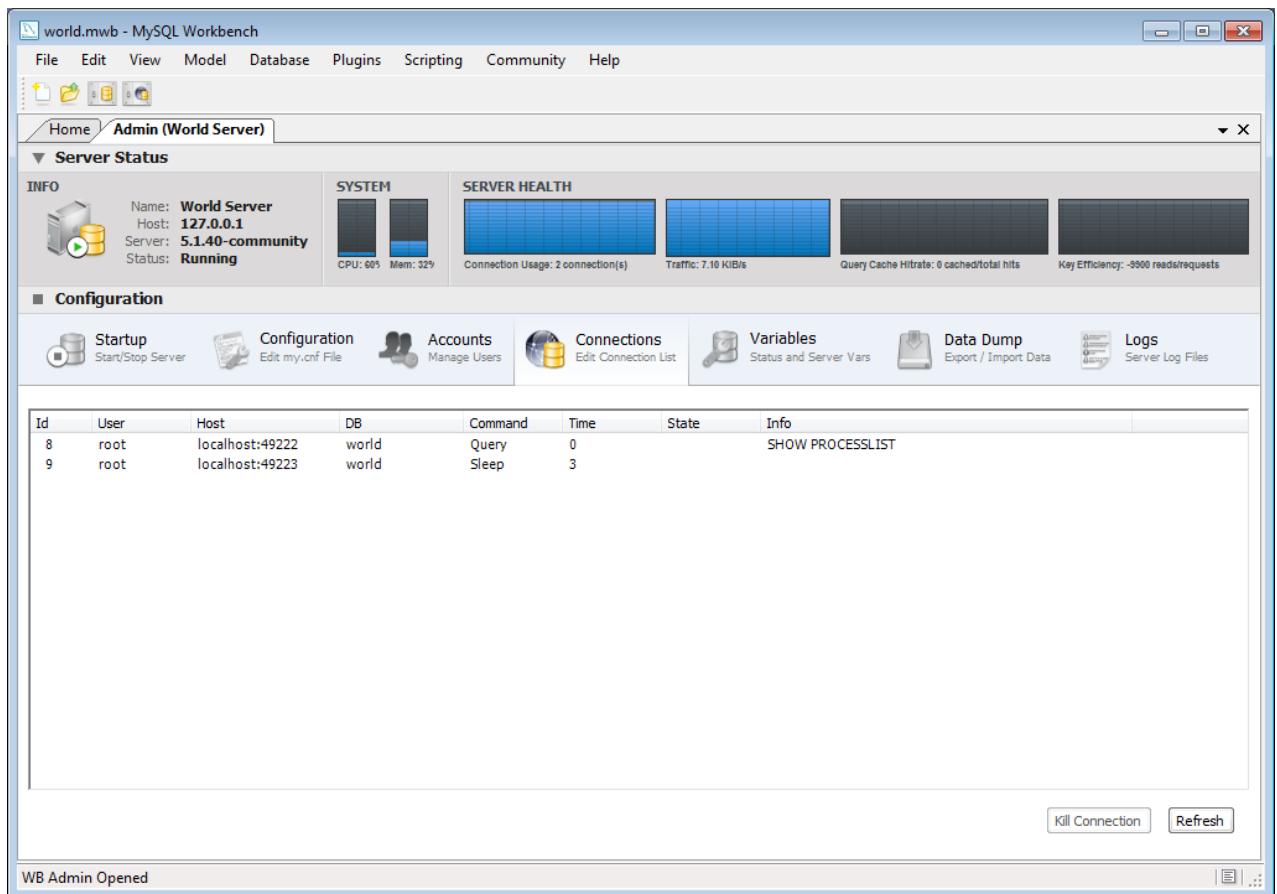
- **DBA** - Grants all privileges
- **MaintenanceAdmin** - Grants privileges to maintain server
- **ProcessAdmin** - Grants privileges to monitor and kill user processes
- **UserAdmin** - Grants privileges to create users and reset passwords
- **SecurityAdmin** - Grants privileges to manage logins and grant and revoke server
- **MonitorAdmin** - Grants privileges to monitor server
- **DBManager** - Grants privileges to manage databases
- **DBDesigner** - Grants privileges to create and reverse engineer any database schema

- **ReplicationAdmin** - Grants privileges to set up and manage replication
- **BackupAdmin** - Grants privileges required to backup databases

8.2.4. Connections tab

This tab lists all current connections to the monitored server.

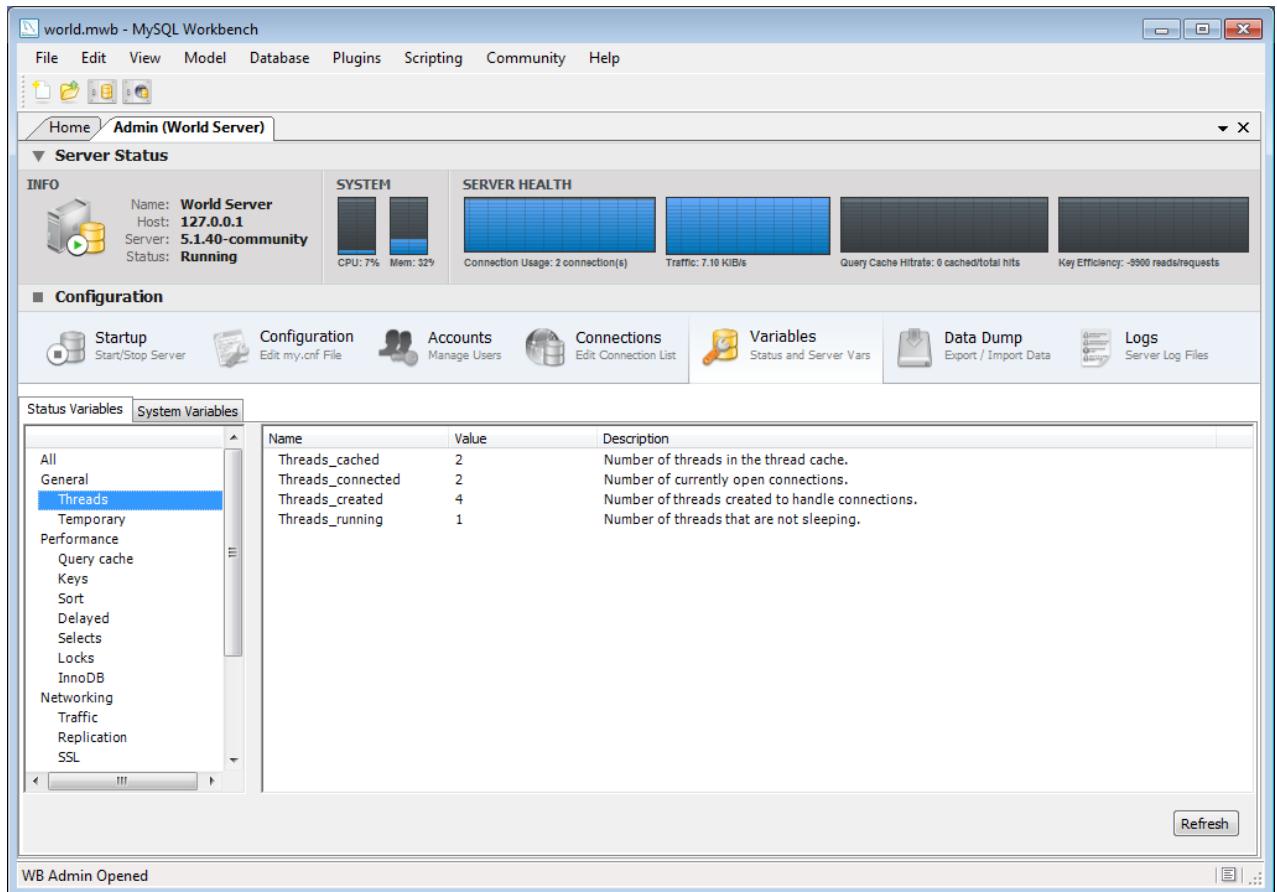
Figure 8.7. Administrator - Connections tab



8.2.5. Variables tab

The Variables tab displays a list of all server and status variables.

Figure 8.8. Administrator - Variables tab



8.2.6. Data Dump tab

The Import/Export Server Data tab allows you to create a dump file, or restore data from a dump file. Clicking the **IMPORT/EXPORT SERVER DATA** action item launches a new Admin screen, at the Data Dump tab.

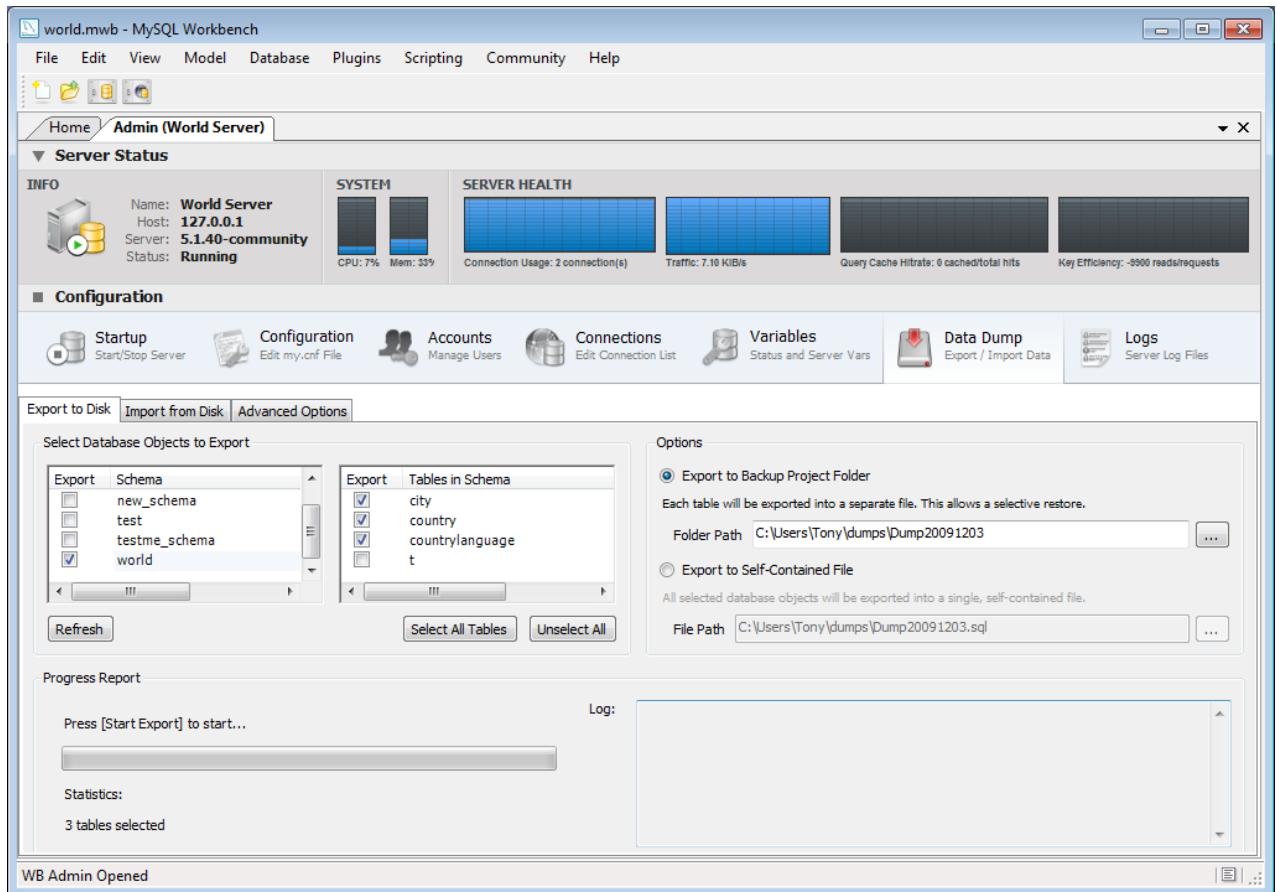
Within the Data Dump tab are three further tabbed windows:

- Export to Disk
- Import from Disk
- Advanced Options

8.2.6.1. Export to Disk

This tab allows you to select the schema and tables you wish to export. You also have the option to export tables to their own files, or all tables to a single file. Exporting tables to individual files allows you to restore on a per-table basis.

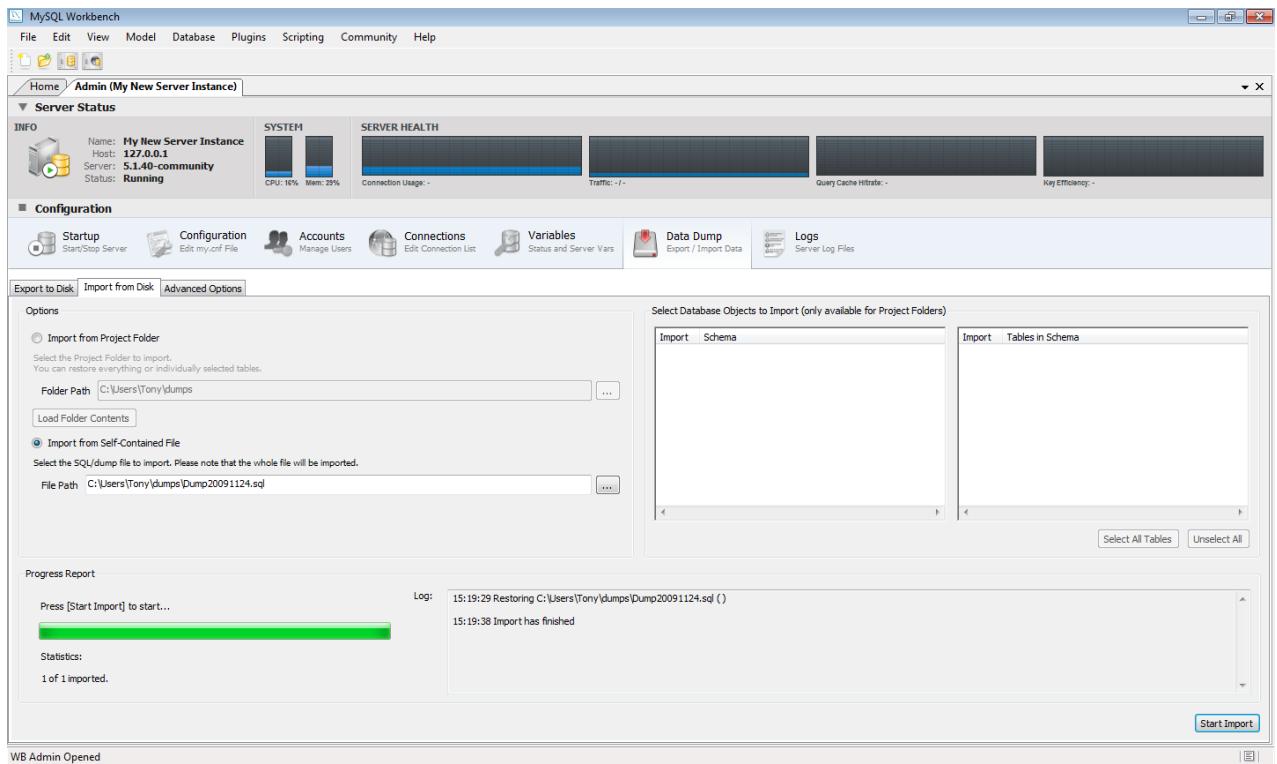
Figure 8.9. Administrator - Export to Disk



8.2.6.2. Import from Disk

This tab allows you to import a previously exported project. You can select to import a project where tables were stored in individual files, in which case you will also be able to select which of these tables you wish to import. You can also simply import a project saved to a single file.

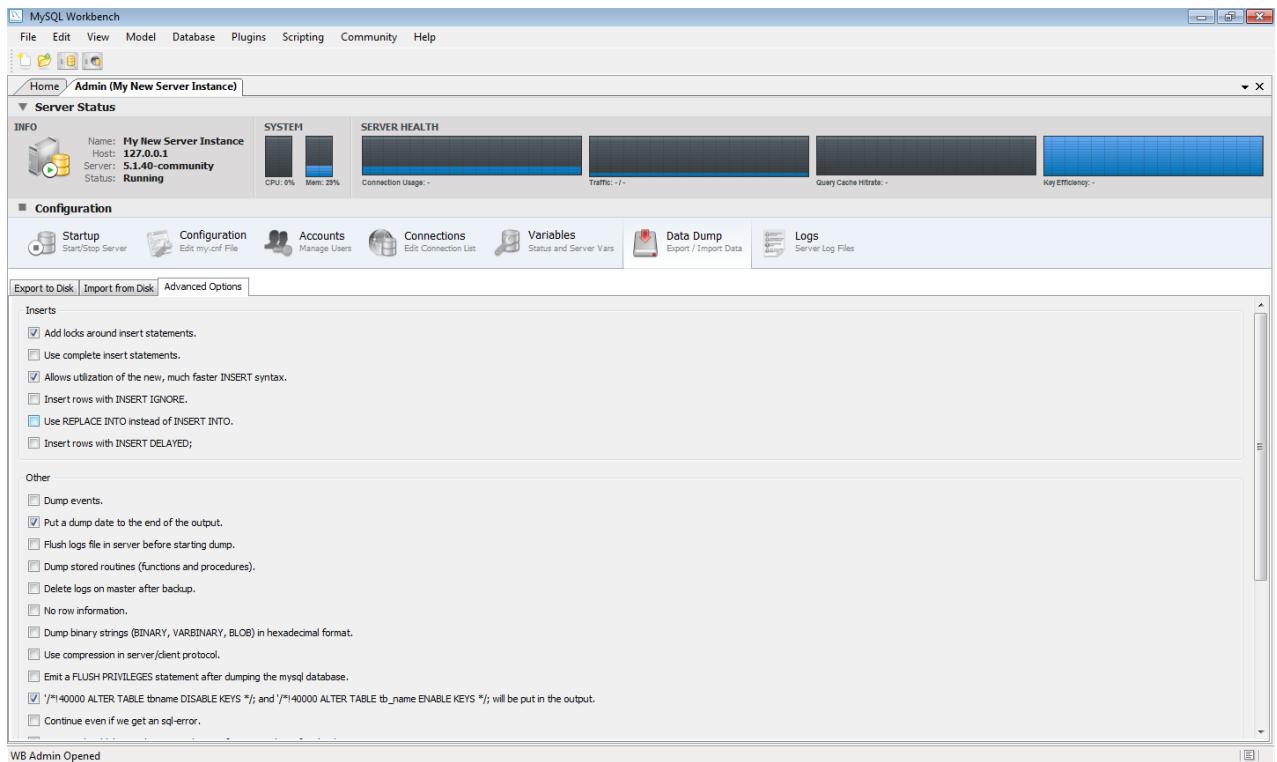
Figure 8.10. Administrator - Import from Disk



8.2.6.3. Advanced Options

This contains a number of options to allow you to control the export process. These options control the SQL generated.

Figure 8.11. Administrator - Advanced Options



8.2.7. Logs tab

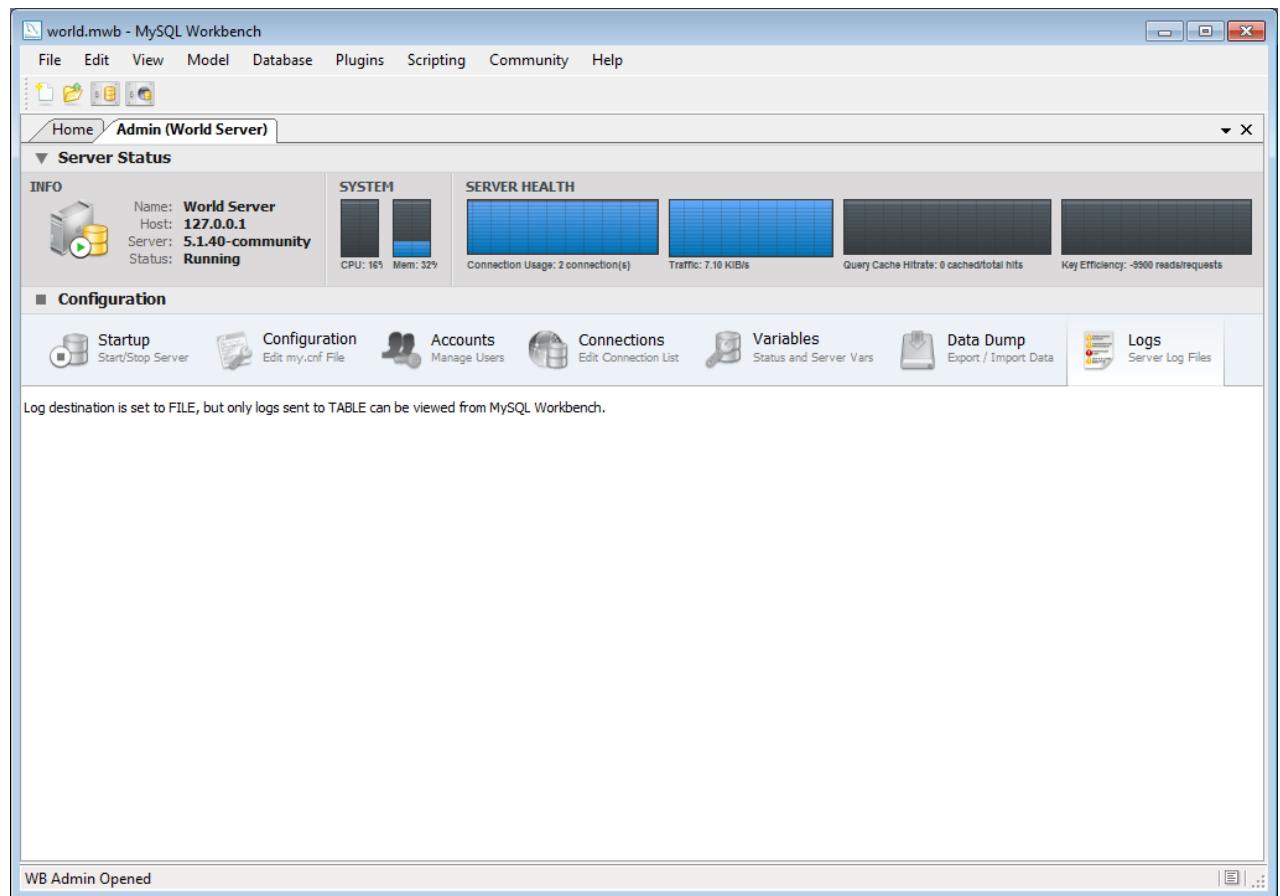
The Logs tab features two sub-tabs:

- General
- Slow Query Log

The **GENERAL** tab shows entries from the server's General log file.

The **SLOW QUERY LOG** tab displays entries from the Slow Query Log file.

Figure 8.12. Administrator - Logs tab



Chapter 9. The Workbench Scripting Shell

9.1. Introduction

The Workbench Scripting Shell provides a means for expanding MySQL Workbench. Through the use of the scripting shell, MySQL Workbench can support new behavior and data sources using code written in languages such as Lua and Python.

The scripting shell is not only useful for expanding MySQL Workbench. By using a script file from within the scripting shell you can perform repetitive tasks programmatically from the command line.

The default development language is [Lua](#), a lightweight scripting language expressly designed for extending applications. For more information about this language see [lua.org](#).

The Python language is also supported, further details of this language can be found from the official [Python site](#).

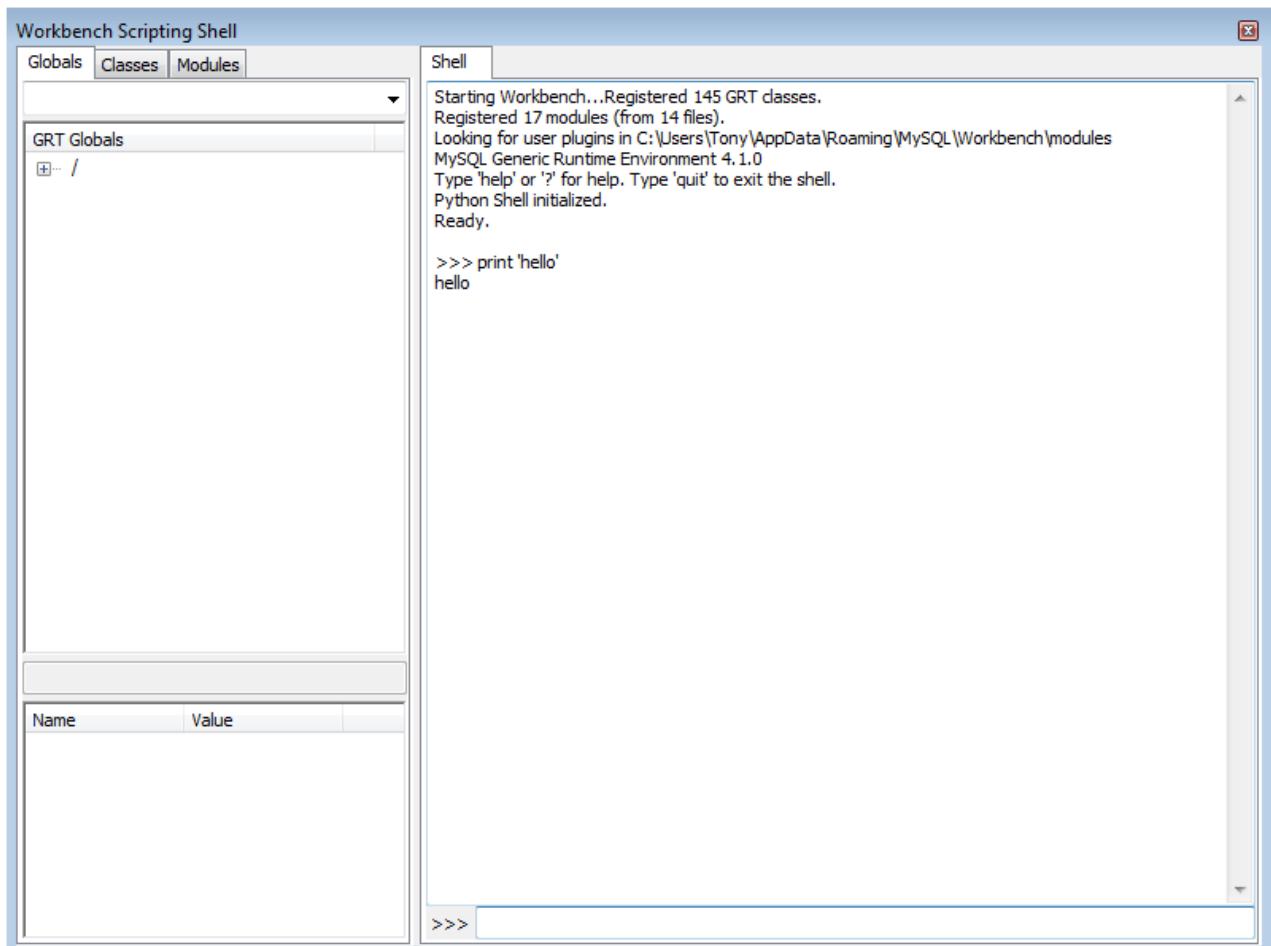
The programming language to be used in Workbench Scripting Shell can be selected from the General tab of the Workbench Preferences dialog. The Workbench Preferences dialog can be displayed using the main menu item [EDIT, PREFERENCES](#).

9.2. Exploring the Workbench Scripting Shell

To open the Workbench Scripting Shell, select [VIEW, ADVANCED, GRT SHELL](#) from the main menu. You can also open the Workbench Scripting Shell using the **Ctrl F3** key combination on Windows and Linux, **cmd F3** on Mac OS X, or by clicking the shell button above the EER diagram navigator. The Workbench Scripting Shell will then open in a new dialog.

The following screenshot shows the Workbench Scripting Shell dialog:

Figure 9.1. The Workbench Scripting Shell



9.3. The Shell

The Workbench Scripting Shell is primarily used for running Lua or Python scripts or typing commands in these languages directly. However, you can also access the Workbench Scripting Shell Scripting Library functions and global functions and objects. To see the available commands type “?”. You can also cut and paste text to and from the shell window.

For information on running script files, type `? run` at the Workbench Scripting Shell prompt. The following message is displayed:

```
Shell Command - shell.run
-----
Load and execute a lua script file.

run filename

Parameters:
filename      File that should be loaded and executed.

Examples:
run scripts/test.lua
Runs the script scripts/test.lua.
```

Opening the Workbench Scripting Shell opens additional windows docked on the right side of MySQL Workbench:

Figure 9.2. The Workbench Scripting Shell Panels

The Workbench Model interface displays a tree structure:

- User Scripts
- User Modules
- User Libraries
- Standard Modules
 - catalog_utils.grt.lua
 - table_utils.grt.lua
 - tools.grt.lua
 - wb_utils_grt.py
 - wb_utils_grt.pyc

GRT Tree

GRT Globals Tree

The GRT Tree window shows a file structure:

- /
- [wb]
- doc
 - logicalModel
 - diagrams
 - customData
 - markers
 - options
 - owner
 - logicalModel
 - overviewPanels

GRT Inspector

Member	Value
currentDiagram	<>
customData	<dict>
diagrams	<list>
markers	<list>
name	
options	<dict>
owner	<object>

Discussion of these additional windows follows.

9.4. The Globals, Modules, and Classes Tabs

The Workbench Scripting Shell features the **GLOBALS**, **CLASSES** and **MODULES** tabs, in addition to the main **SHELL** tab.

The Globals Tab

When the Globals tab is selected, the **GRT TREE** window is found on the left side of the screen. At the top of the window is a drop down list box that contains **GRT GLOBALS TREE** by default.

Selecting an object in the GRT tree displays its properties in the **GRT INSPECTOR** window found immediately below the **GRT TREE** window.

The GRT Inspector

The GRT inspector displays the properties of the object that is currently selected in the [GRT Global Tree](#).

The Classes Tab

A [class](#) is a user-defined data type formed by combining primitive data types. This tab shows the definitions of the classes used by the objects in the [Modules](#) tab.

When the **CLASSES** tab is selected, the dropdown listbox lists the following items:

- Group by Name – group by the object name
- Group by Hierarchy – group by inheritance
- Group by Package – group by functionality

The default view for this tab is by name; this view simply shows all the different objects arranged alphabetically. Click the + icon or double click a package to show the properties of the struct.

If you switch to the hierarchical view you will see [GRT Object](#) — the parent object from which all other objects are derived.

The Modules Tab

The **MODULES** tab allows you to browse the MySQL Workbench installed modules and their functions.

Chapter 10. Plugins

Using plugins you can write your own modules to perform a variety of tasks. To do this you need to know the Lua scripting language and be familiar with the various objects used by MySQL Workbench. For more information on this topic see [Chapter 9, The Workbench Scripting Shell](#).

10.1. The `catalog_utils.grt.lua` Module

The best way to learn about plugins is to examine one. An example Lua plugin is located in the `C:\Program Files\MySQL\MySQL Workbench version\modules` directory. In that directory you can find the `catalog_utils.grt.lua` Lua file. This is the plugin for copying SQL to the clipboard.

Any plugin must contain at least three functions:

- `getModuleInfo()` – This function tells the GRT that the file is a module.
- `getPluginInfo()` – This function tells MySQL Workbench that the module is a function and registers that plugin.
- The function or functions that perform the work of the module.

Open the `catalog_utils.grt.lua` file in a text editor, and you can see that it has the required `objectPluginInput()` and `getPluginInfo()` functions, a helper function called `objectPluginInput()` and the function that performs the work of this module, `copySQLToClipboard()`.

10.1.1. Using the GRT Shell

You can use the GRT shell to help understand the `catalog_utils.grt.lua` module. Open the GRT shell by pressing **Ctrl F3** or by selecting the GRT SHELL menu option found under the VIEW, ADVANCED menu options.

Notice that the `getPluginInfo` function makes use of the `grtv` module by invoking its `getObj` function. The `grtv` module is a built-in Lua module that assists with working from the GRT shell. To find out more about the `grtv` module, type `? grtv` in the GRT shell console window. You should see something similar to the following:

```
GRT Value Management Library - grtv
-----
A library that contains functions to work with GRT values.

clearList           child             fromXml
getContent-Type   getKey            getListItemByObjName
getn               getGlobal          insert
load               newDict            newList
newObj              remove             save
setGlobal           toLua             toXml
typeOf
```

Typing `help grtv.<command>` displays help on a specific command. For example, entering `help grtv newObj` displays the following output.

```
GRT Value Management Library - grtv.newObj
-----
Creates a new GRT object initialized with the optional given values. All simple values ("int", "string", "real") are initialized and all lists and dicts are created.
Object references are left null.

grtv newObj (structName[, initValuesDict])

Parameters:
structName      Struct name of the object to create
initValuesDict   A dictionary containing initial values for object fields.

Examples:
rdbmsMgmt= grtv newObj("db.mgmt.Management",
{name="rdbmsManagement", owner=app})
Create a new object from the struct "db.mgmt.Management"
```

The `grtv newObj` function is used by both the `objectPluginInput` and the `getPluginInfo` functions of the `catalog_utils.grt.lua` module. A `structName` is the first parameter for the `grtv newObj` function. In the case of the `objectPluginInput` function in `catalog_utils.grt.lua` module, the `structName` is `app.PluginObjectInput`. To find out more about this struct, locate it in the `Structs` window.

10.2. Accessing Plugins

Plugins can be accessed by calling them from the GRT Shell. For more information on this topic see [Section 9.3, “The Shell”](#). You can also add entries to the `main_menu.xml` file, found in the `C:\Program Files\MySQL\MySQL Workbench version\data` directory.

To determine how to add entries to the `main_menu.xml` file, open it in a text editor and search for `Plugins Menu`. Performing this search takes you to the top level `Plugins` entry. Beneath this entry find a sample submenu entry that has been commented out. This entry should look something like the following:

```
<value type="object" struct-name="app.MenuItem" id="com.mysql.wb.menu.plugins.plugin_name">
  <link type="object" key="owner" struct-name="app.MenuItem">com.mysql.wb.menu.plugins</link>
  <value type="string" key="name">menu_name</value>
  <value type="string" key="caption">menu_caption</value>
  <value type="string" key="itemType">cascade</value>
  <value type="list" key="subItems" content-type="object" content-struct-name="app.MenuItem">
    </value>
```

Use this entry as a guideline for creating submenus under the `PLUGINS` top level menu entry.

Warning

Ensure that any XML added to the `main_menu.xml` file is well-formed, otherwise MySQL Workbench may crash on startup.

Chapter 11. Keyboard Shortcuts

On Mac OS X modifier key is **cmd**, on other platforms it is **Ctrl**.

File Menu

Function	Context	Keyboard Shortcut
New Model	All	Modifier+N
Open Model	All	Modifier+O
Open SQL Script	SQL Editor	Modifier+Shift+O
Close Tab	All	Modifier+W
Save Model	Model	Modifier+S
Save Script	SQL Editor	Modifier+S
Save Model As	Model	Modifier+Shift+S
Save Script As	SQL Editor	Modifier+Shift+S
Forward Engineer SQL CREATE Script	Model	Modifier+Shift+G
Forward Engineer SQL ALTER Script	Model	Modifier+Alt+Y
Synchronize With SQL CREATE Script	Model	Modifier+Shift+Y
Print	EER Diagram mode only	Modifier+P
Exit	All	Modifier+Q

Edit Menu

Function	Context	Keyboard Shortcut
Undo	Model, EER Diagram	Modifier+Z
Redo	Model, EER Diagram	Modifier+Y, Modifier+Shift+Z (Mac OS X)
Cut	All	Modifier+X
Copy	All	Modifier+C
Paste	All	Modifier+V
Delete	All	Modifier+Delete, Command+BackSpace (Mac OS X)
Edit Selected	Model, EER Diagram	Modifier+E
Edit Selected in New Window	Model, EER Diagram	Modifier+Shift+E
Select All	EER Diagram	Modifier+A
Find	All	Modifier+F
Find Advanced	All	Modifier+Alternate+F
Find Next	All	F3
Find Previous	All	Shift+F3
Search and Replace	All	Modifier+Shift+F

View Menu

Function	Context	Keyboard Shortcut
Output Window	All	Modifier+F2, Modifier+Option+2 (Mac OS X)
Set Marker n	EER Diagram	Modifier+Shift+n (n is integer 1..9)
Go to Marker n	EER Diagram	Modifier+n (n is integer 1..9)

Arrange Menu

Function	Context	Keyboard Shortcut
Bring to Front	EER Diagram	Modifier+Shift+F

Send to Back	EER Diagram	Modifier+Shift+B
--------------	-------------	------------------

Model Menu

Function	Context	Keyboard Shortcut
Add Diagram	Model, EER Diagram	Modifier+T
Validate All	Model, EER Diagram	Modifier+Alt+V
Validate All (MySQL)	Model, EER Diagram	Modifier+Alt+B
Model Options	Model, EER Diagram	Command+Alternate+, (Shorcut only available on Mac OS X)

Query Menu

Function	Context	Keyboard Shortcut
Execute statement	SQL Editor	Modifier+Return
Execute statements	SQL Editor	Modifier+Shift+Return
New Tab	SQL Editor	Modifier+T

Database Menu

Function	Context	Keyboard Shortcut
Query Database	All	Modifier+U
Reverse Engineer	Model, EER Diagram	Modifier+R
Forward Engineer	Model, EER Diagram	Modifier+G
Synchronize Model	Model, EER Diagram	Modifier+Y

Scripting menu

Function	Context	Keyboard Shortcut
Scripting Shell	All	Modifier+F3, Modifier+Option+3 (on Mac OS X)
Run Workbench Script File	All	Modifier+Shift+R

Help Menu

Function	Context	Keyboard Shortcut
Help Index	All	F1, Command+Option+question (on Mac OS X)

EER Diagram Mode

In the EER Diagram view, there are a number of other keyboard shortcuts available.

Function	Keyboard Shortcut
Selection tool	Escape
Hand tool	H
Delete tool	D
Layer tool	L
Note tool	N
Image tool	I
Table tool	T
View tool	V
Routine Group tool	G

Non-Identifying Relationship 1:1	1
Non-Identifying Relationship 1:n	2
Identifying Relationship 1:1	3
Identifying Relationship 1:n	4
Identifying Relationship n:m	5
Relationship Using Existing Columns	6

Chapter 12. MySQL Workbench FAQ

Frequently Asked Questions with answers.

Questions

- **12.1:** When a model is exported using the main menu item **FILE, EXPORT, FORWARD ENGINEER SQL CREATE SCRIPT**, some server variables are temporarily set to enable faster SQL import by the server. The statements added at the start of the code are:

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@SQL_MODE, SQL_MODE='TRADITIONAL';
```

These statements function as follows:

- `SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;` - determines if an InnoDB engine performs duplicate key checks. Import is much faster for large data sets if this check is not performed.
- `SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;` - determines if the server should check that a referenced table exists when defining a foreign key. Due to potential circular references, this check must be turned off for the duration of the import, in order to allow defining foreign keys.
- `SET @OLD_SQL_MODE=@SQL_MODE, SQL_MODE='TRADITIONAL';` - sets `SQL_MODE` to `TRADITIONAL`, causing the server to operate in a more restrictive mode.

These server variables are then reset at the end of the script using the following statements:

```
SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

- **12.2:** MySQL Workbench 5.0 appears to run slowly. How can I increase performance?

Questions and Answers

- 12.1: When a model is exported using the main menu item FILE, EXPORT, FORWARD ENGINEER SQL CREATE SCRIPT, some server variables are temporarily set to enable faster SQL import by the server. The statements added at the start of the code are:**

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@SQL_MODE, SQL_MODE='TRADITIONAL';
```

These statements function as follows:

- `SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;` - determines if an InnoDB engine performs duplicate key checks. Import is much faster for large data sets if this check is not performed.
- `SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;` - determines if the server should check that a referenced table exists when defining a foreign key. Due to potential circular references, this check must be turned off for the duration of the import, in order to allow defining foreign keys.
- `SET @OLD_SQL_MODE=@SQL_MODE, SQL_MODE='TRADITIONAL';` - sets `SQL_MODE` to `TRADITIONAL`, causing the server to operate in a more restrictive mode.

These server variables are then reset at the end of the script using the following statements:

```
SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

- 12.2: MySQL Workbench 5.0 appears to run slowly. How can I increase performance?**

Although graphics rendering may appear slow, there are several other reasons why performance may be less than expected. The following tips may offer improved performance:

- Upgrade to the latest version. MySQL Workbench 5.0 is still being continually maintained and some performance-related issues may have been resolved.

- Limit the number of steps to save in the **UNDO HISTORY** facility. Depending on the operations performed, having an infinite undo history can use a lot of memory after a few hours of work. In **TOOLS**, **OPTIONS**, **GENERAL**, enter a number in the range 10 to 20 into the **UNDO HISTORY SIZE** spinbox.
- Disable relationship line crossing rendering. In large diagrams, there may be a significant overhead when drawing these line crossings. In **TOOLS**, **OPTIONS**, **DIAGRAM**, uncheck the option named **DRAW LINE CROSSINGS**.
- Check your graphics card driver. The GDI rendering that is used in MySQL Workbench 5.0 is not inherently slow, as most video drivers support hardware acceleration for GDI functions. It can help if you have the latest native video drivers for your graphics card.
- Upgrade to MySQL Workbench 5.1. MySQL Workbench 5.1 has had many operations optimized. For example, opening an object editor, such as the table editor, is much faster, even with a large model loaded. However, these core optimizations will not be back-ported to 5.0.

Appendix A. MySQL Workbench Change History

The following sections outline the changes between versions for MySQL Workbench.

A.1. Changes in Release 5.2

A.1.1. Changes in MySQL Workbench 5.2.17 (Not yet released beta)

Seventh public Beta release. This section documents all changes and bug fixes that have been applied since the release of 5.2.16.

Functionality added or changed:

- If MySQL Workbench attempts to connect to MySQL Server version 4.x, an error message is generated to indicate this version is not supported. ([Bug#51455](#))
- A description column has been added to the snippets table. ([Bug#51010](#))
- Various improvements to the SQL Editor user interface, including additional context-sensitive menu items in the Object Explorer, additional buttons in the Create Table and Alter Table dialogs, and more clearly displayed error messages. ([Bug#50637](#), [Bug#49918](#))
- In the SQL Editor, the **COMMENTS** tab in the **CREATE VIEW** dialog has been removed. ([Bug#49270](#))

Bugs fixed:

- The New Server Instance wizard did not set the correct name for the startup script, when the installation type was set to Fedora Linux (Vendor Package). Testing the connection resulted in the error “Operation failed: /etc/init.d/mysql start is invalid”. ([Bug#51802](#))
- In the Object Explorer of the SQL Editor, right-clicking on a table displayed the menu option **SEND TO SQL EDITOR**. Editor was spelled incorrectly. ([Bug#51790](#))
- When opening the `ip2nation.sql` file MySQL Workbench crashed. ([Bug#51606](#), [Bug#51531](#))
- Exporting a result set to a CSV file resulted in a file with a trailing comma appended to each line including the heading. ([Bug#51594](#))
- In the **INSERTS** tab of the Table Editor, if a row was right-clicked and **DELETE SELECTED ROWS** selected, MySQL Workbench crashed. ([Bug#51584](#))
- On the MySQL Model tab, if a table was clicked on and edited using the context menu option **EDIT TABLE**, MySQL Workbench crashed. ([Bug#51410](#), [Bug#50936](#))
- If all screens and tabs were closed in MySQL Workbench and then **DATA**, **MANAGE CONNECTIONS** selected, then when the **MANAGE DB CONNECTIONS** dialog was closed an unknown exception occurred. ([Bug#51403](#))
- If a server instance was created with the New Instance wizard with SSH disabled, then the following exception occurred when attempting to administer the server instance:

```
Exception = System.Runtime.InteropServices.SEHException
Message = External component has thrown an exception.
FullText = System.Runtime.InteropServices.SEHException: External component has thrown an
exception.
   at MySQL.Forms.DrawBoxImpl.drawbox_mouse_click(Object sender, MouseEventArgs e)
   at System.Windows.Forms.Control.OnMouseClick(MouseEventArgs e)
   at System.Windows.Forms.Control.WmMouseUp(Message& m, MouseButtons button, Int32
clicks)
   at System.Windows.Forms.Control.WndProc(Message& m)
   at System.Windows.Forms.Control.ControlNativeWindow.OnMessage(Message& m)
   at System.Windows.Forms.Control.ControlNativeWindow.WndProc(Message& m)
   at System.Windows.Forms.NativeWindow.Callback(IntPtr hWnd, Int32 msg, IntPtr wparam,
IntPtr lparam)
```

([Bug#51368](#))

- After exporting a schema to disk, importing the schema did not restore the tables. ([Bug#51261](#))

- In the SQL Editor, if a foreign key name was changed using the Foreign Key tab or Indexes tab of the Alter Table dialog, the following error was generated:

```
Type mismatch: expected type string, but got list
```

(Bug#51192)

- When the MySQL Workbench source code was configured, the package `gnome-keyring-1` was not found. (Bug#51090)
- There were discrepancies between the list of pre-requisite packages given on the MySQL Workbench website and those listed in the `README` file in the MySQL Workbench distribution. (Bug#51085)
- An Out of Range exception occurred when switching from the Table Editor to the Home screen. (Bug#50980, Bug#51030)
- When a schema with Foreign keys and associated automatically generated foreign indexes was exported with the **SKIP CREATION OF FOREIGN KEYS** checkbox selected, the generated script still contained the indexes. (Bug#49987)
- The live data editor in SQL Editor did not allow the columns to be sorted in descending order by clicking the column heading. It only allowed column sorting in ascending order through clicking the column heading. (Bug#49302)
- In the SQL editor, if Alter Table was invoked for a table, and then the partitioning tab selected and partitioning enabled, it was possible to select a partition count of 0, which then generated an error if an attempt was made to apply changes. (Bug#49050)

A.1.2. Changes in MySQL Workbench 5.2.16 (17th February 2010 beta)

Sixth public Beta release. This section documents all changes and bug fixes that have been applied since the release of 5.2.15.

Functionality added or changed:

- In the SQL editor, the keyboard shortcut to run a single query has been changed to **Ctrl + Enter**. To run all queries the keyboard shortcut has been changed to **Ctrl + Shift + Enter**. (Bug#50747)
- The key sequence **Ctrl+Q** has been added as a shortcut for the main menu item `FILE, EXIT`. (Bug#50727)
- In the SQL Editor a comment has been added to the EDIT statement to clarify its functionality. (Bug#50705)
- The Manage DB Connections dialog now prevents removal of a connection used by a server instance. (Bug#50547)
- In the Logs tab of the Administrator it was not possible to view long queries, or copy their text. MySQL Workbench has been changed so that double-clicking on an entry in the Logs tab pops up a dialog with the complete text, and also provides a **COPY DETAIL** button to copy the text of the entry. (Bug#49442)
- The data dump facility, used for exporting data to disk, has been modified to allow the `--single-transaction` option to be specified.

If `--single-transaction` is enabled, the table selection will be restricted the following ways:

1. If a single schema is selected, it is possible to select/deselect its tables as required.
2. If more than one schema is selected, all tables from these schemas must be selected, because `mysqldump --databases` will be used in this case.

(Bug#49220)

- When a model with multiple EER Diagrams was opened, all EER Diagrams would be displayed in tabs. This happened whether or not a EER Diagram had been displayed in a tab prior to saving the model.

MySQL Workbench has been changed so that the EER Diagram tab state is saved, so that when a model is opened, only those EER Diagram tabs that were open on save are restored. This prevents unnecessary cluttering of the interface. (Bug#44454, Bug#50732)

Bugs fixed:

- **Security Fix:** Passwords were stored in plain text format in the file `server_instances.xml`.

To improve security MySQL Workbench has been changed in the following ways:

1. The password is no longer stored in the connection XML file.
2. When a connection is opened, a password request dialog is displayed and the password requested. The password can optionally be stored in the system keychain/vault/keyring.
3. When editing a connection profile, you can also store the password in the system keychain.
4. For compatibility, when MySQL Workbench starts it will look for passwords stored in the XML file. If any password is found, it will be removed from the XML file and automatically stored in the keychain. When MySQL Workbench exits, the connections file will be free from passwords.

([Bug#50194](#))

- When using the **SERVER ADMINISTRATION** link, or **MANAGE SECURITY** link, on the **HOME** screen, to log in to a remote server through an SSH connection, if the password dialog was closed, and cancellation of the SSH connection dialog acknowledged, then the following exception was generated:

```
Exception = System.Runtime.InteropServices.SEHException
Message = External component has thrown an exception.
FullText = System.Runtime.InteropServices.SEHException: External component has thrown an
exception.
    at MySQL.Forms.DrawBoxImpl.drawbox_mouse_click(Object sender, MouseEventArgs e)
...
```

([Bug#51088](#))

- In the **CREATE A NEW SERVER INSTANCE** wizard, on the last page of the wizard if the BACK button was pressed and then the NEXT button pressed, an error was generated stating the server instance already existed. ([Bug#51060](#))
- Using FILE, EXPORT, FORWARD ENGINEER ALTER SCRIPT to export a model resulted in MySQL Workbench generating the exception AccessViolationException. ([Bug#51053](#))
- In the SQL Editor, if using the Alter Table dialog, attempting to apply changes after renaming an index resulted in the following error:

```
SQL Error 1091: Can't DROP 'username_foo'; check that column/key exists
```

([Bug#50970](#))

- In the **SYSTEM PROFILE** tab of the **MANAGE SERVER INSTANCES** dialog, if the **PATH TO CONFIGURATION FILE** textfield was set using the browse button, ..., then the value in the textfield appeared to be set correctly to the selected file. However, if the dialog was closed and reopened, the new path was not displayed in the text field. If the path was manually entered, rather than using the browse button, the textfield would display the correct path even if the dialog was closed and reopened. ([Bug#50965](#))
- In the SQL Editor, if the Edit Data dialog was invoked for a table, and some data edited, MySQL Workbench crashed when the **APPLY SQL** button was clicked. ([Bug#50920](#))
- An exported script containing triggers and views resulted in the error **#1046 - No database selected** when run on the MySQL Server. This was due to a missing **USE DATABASE** statement in the generated script. ([Bug#50900](#))
- The second invocation of FILE, PRINT caused MySQL Workbench to crash. ([Bug#50885](#))
- The SQL code editors used in the modeling functionality within MySQL Workbench, for example in the Triggers tab of the Table Editor, failed to identify and highlight SQL code errors. ([Bug#50835](#))
- In the SQL Editor the text “Parsing SQL ...” displayed in the status bar remained once the parsing was complete.

MySQL Workbench has been changed to display “No errors found” once parsing has successfully completed. ([Bug#50833](#))

- If a MySQL server was set to accept named pipe connections only, and then a server instance created in MySQL Workbench using a named pipe connection, it was not possible to subsequently connect to the server with MySQL Workbench. ([Bug#50830](#))
- Expanded Schemata Palettes in the SQL Editor would collapse when switching between SQL Editor tabs. ([Bug#50815](#))
- In the SQL Editor the toolbar button to execute SQL statements was missing.

MySQL Workbench also now includes a new toolbar button to execute a single statement. ([Bug#50791](#))

- In the MySQL Model tab, if the Table Editor was launched and then the Home screen tab clicked, MySQL Workbench displayed the MySQL Workbench Unexpected Error dialog. ([Bug#50768](#))

- On the MySQL Model page using Roles and Users to grant privileges to a schema resulted in errors when the model was forward engineered, and the resultant script applied against a MySQL server. The script failed due to invalid SQL syntax, as the SQL created did not correctly apply the privileges to all schema objects. ([Bug#50762](#))
- In the Data Dump facility of the Administrator, attempting to export a schema to disk failed if a table name contained a space. ([Bug#50728](#))
- In the Workbench Scripting Shell dialog, clicking on an item in the value inspector panel (lower left corner) caused an exception:

```
System.Runtime.InteropServices.SEHException: External component has thrown an
exception.
  at MySQL.Grt.TreeModel.expand_node(NodeId node)
  at MySQL.Grt.TreeModel.TreeViewExpanding(Object sender, TreeViewAdvEventArgs e)
  at System.EventHandler`1.Invoke(Object sender, TEventArgs e)
  at Aga.Controls.Tree.TreeViewAdv.OnExpanding(TreeNodeAdv node)
  ...
...
```

([Bug#50683](#))

- In the Data Dump facility of the Administrator it was not possible to import a file where the path contained a space. ([Bug#50609](#), [Bug#50007](#))
- In the SQL Editor a situation occurred where all results tabs could not be closed. Further, results tabs were incorrectly created with the same label. ([Bug#50334](#), [Bug#50865](#))
- In the Manage DB Connections dialog it was not possible to select a default schema for the Socket/Pipe connection type, as this facility was not provided by the dialog user interface. ([Bug#50283](#))
- It was possible to connect to a database using a password containing a space. However, when this connection was used to attempt a backup the operation failed, due to the password not being correctly quoted. ([Bug#50213](#))
- The messages generated by `mysqldump` when a data export operation failed were difficult to interpret, for example:

```
Operation failed with exitcode 2
```

([Bug#50137](#))

- In the SQL Editor, when editing table data, if an operation failed when changes were applied, the error message was not clearly visible. ([Bug#50112](#))
- In the SQL Editor, if a `LONGTEXT` field was being edited in the table data live editor, when the changes were applied an error was generated.

MySQL Workbench has also been changed so that large text values need to be edited in an external editor and then pasted into the grid cell. ([Bug#50111](#), [Bug#50692](#), [Bug#50948](#), [Bug#50814](#))

- When synchronizing a schema with a live server, the scale of columns with type `DECIMAL` was erroneously set to zero. For example, a `DECIMAL(17, 5)` was found to be set to `DECIMAL (17, 0)` in the generated script. ([Bug#50110](#))
- When a non-SSH server instance was created, and a connection made to a remote server, only the Data Dump facility of the Administrator was available. The same problem occurred if SSH-based administration was deselected for the server instance. ([Bug#50098](#))
- In the SQL Editor, if a database was selected in the Overview tab and then either the `DROP SCHEMA...` context menu item was selected, or the drop schema toolbar button clicked, the `DROP DATABASE` dialog was displayed. However, if the dialog close button was then clicked to cancel the dialog, the database was still dropped, instead of being unaffected by the cancelling of the dialog. ([Bug#50072](#), [Bug#50960](#))
- In the **CREATE A NEW SERVER INSTANCE** wizard, if an operating system of type **WINDOWS (MySQL 5.1 INSTALLER PACKAGE)** was selected, then the following error was generated on the **TEST SETTINGS** page:

```
Error: File C:\Program Files\MySQL\MySQL Server 5.1\my.ini doesn't exist
```

This was because the file was actually stored in `C:\Programas\MySQL\MySQL Server 5.1\my.ini`.

Similar bugs where the configuration file could not be found were reported on both English and non-English systems. ([Bug#50050](#), [Bug#50635](#), [Bug#50966](#), [Bug#50873](#), [Bug#51008](#))

- If a table was dropped from a live database, and then the model synchronized, the dropped table would be detected, but the table was not dropped from the model concerned. ([Bug#50000](#))

- For a default MySQL Server installation, no `my.ini` or `my.cnf` file is created. This proved problematic when creating a server instance in MySQL Workbench, as the **CREATE A NEW SERVER INSTANCE** wizard expected a configuration file to be specified. If the path to the configuration file was left blank, a model error dialog was displayed by the wizard. If alternatively, one of the standard locations for the configuration file was entered, problems arose when an attempt was made to subsequently change configuration values in the configuration section of the Admin screen. The problems included MySQL Workbench hanging, and repetitive requests for a 'super user' password. ([Bug#49766](#), [Bug#50317](#))
- In the SQL Editor, in the add routine dialog, if the template was used, and a simple `SELECT 1;` statement added to the template, then when changes were applied the following error was generated:

```
ERROR 1064: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for
SQL Statement:
CREATE PROCEDURE `world`.`new_routine` ()
BEGIN
SELECT 1
```

([Bug#49710](#))

- In the **FORWARD ENGINEER SQL SCRIPT** wizard, selecting the checkbox **OMIT SCHEMA QUALIFIER IN OBJECT NAMES** caused the `CREATE SCHEMA` statement to be removed. `DROP SCHEMA` statements were also removed, even if the **GENERATE DROP SCHEMA** checkbox was selected.

MySQL Workbench has been changed so that if the **OMIT SCHEMA QUALIFIER IN OBJECT NAMES** checkbox is selected, it is possible to optionally select the checkbox **INSERT USE STATEMENTS**, to enable or disable the use of `USE` statements. ([Bug#49682](#))

- If the Data Dump facility in Administrator was used to export to a self-contained file, then when that file was imported an unhandled exception was generated. ([Bug#49529](#))
- If the MySQL Server was stopped then attempting to access the Logs, Accounts, Connections, Variables or Data Dump tabs in the Administrator resulted in unhandled exceptions. ([Bug#49439](#))
- MySQL Workbench did not support SSH keys that required a passphrase to be entered. ([Bug#49418](#))
- When using an SSH connection to manage a remote server, MySQL Workbench repeatedly prompted the user to enter a password, even though a key file had been specified. ([Bug#49307](#))
- In the SQL Editor, when using the live editor, if `QUERY`, `EXPORT RESULTS` was selected from the main menu and a CSV output format chosen, then the exported data was found to be in tab delimited format, rather than CSV format. ([Bug#49303](#))
- When MySQL Workbench was connected to a remote server using an SSH connection, and a data export operation performed, then MySQL Workbench actually attempted to perform the data export on a local server instance, rather than on the connected remote server. If the remote and local server had a common user account, then a backup could be performed that appeared to complete successfully, but which contained data from the local server rather than the connected remote server. ([Bug#49295](#))
- In the Alter Table dialog of the SQL Editor, support for triggers appeared to be provided, but underlying functionality was missing. ([Bug#49287](#))
- In the SQL Editor, if Alter Table was invoked, and the collation for a column changed, then after clicking **APPLY**, no changes were detected. ([Bug#49277](#))
- The descriptions used for options in the **ADVANCED OPTIONS** tab in the Data Dump section of the administrator were lacking clarity and in some cases dated. ([Bug#49224](#))
- The import log contained messages with redundant parentheses. ([Bug#49218](#))
- There was no facility to cancel the Export to Disk process, once started, in the Administrator. Further, the **START EXPORT** button erroneously remained enabled during the export process, allowing the user to start new export processes, resulting in errors. ([Bug#49115](#))
- In the Data Dump facility of the Administrator, if a schema was selected, but its corresponding checkbox not selected, and then several of its tables selected for export, the following error occurred when the **START EXPORT** button was clicked:

```
Nothing to do, no schemas or tables selected.
```

([Bug#49110](#))

- In the **CREATE A NEW SERVER INSTANCE** wizard, using the default SSH Key Path, `~/.ssh/id_rsa`, resulted in a 'file not found' error when testing the connection. However, if the path was entered as `/home/username/.ssh/id_rsa` the connection test was successful. The same problem was also present in the **MANAGE DB CONNECTIONS** dialog. ([Bug#49090](#))

- The first connection created with New Connection did not appear immediately in the list of available connections. ([Bug#49079](#), [Bug#49801](#))
- Numerous variables were missing from the DDL section of the **STATUS VARIABLES** tab in the administrator. ([Bug#49073](#), [Bug#49077](#))
- In the configuration file editor it was possible to select a directory, such as **TEMP DIRECTORY** on the General tab, without specifying a corresponding path. ([Bug#49035](#))
- In the General tab of the configuration file editor, the option **DEFAULT TABLE TYPE** was present, even though it was deprecated in MySQL Server 5.0. Further, selecting this option and applying changes did not change the server configuration file. ([Bug#49006](#))
- If the command for checking server status was changed in a server instance, the change did not take effect unless MySQL Workbench was restarted. ([Bug#48992](#))
- MySQL Workbench did not correctly detect the status of MySQL Servers where multiple servers were running on the same host, and so displayed incorrect server status on the Admin screen. ([Bug#48975](#))
- In the **FORWARD ENGINEER SQL SCRIPT** wizard, selection of the **SKIP CREATION OF FOREIGN KEYS** checkbox was not reflected in the generated script. ([Bug#47969](#))
- When using MySQL Workbench to create an [ALTER](#) script, the generated script did not reflect columns where only the case of the column name had changed. ([Bug#45556](#))
- If a schema that contained tables with no engine defined was reverse engineered, and then the engine type was changed in MySQL Workbench, then when the model was exported the ALTER script did not contain code to change the engine of the table. ([Bug#45110](#))

A.1.3. Changes in MySQL Workbench 5.2.15 (28th January 2010 beta)

Fifth public Beta release. This section documents all changes and bug fixes that have been applied since the release of 5.2.14.

Functionality added or changed:

- The GRT Shell exhibited various issues and could generate exceptions in some circumstances.

The GRT Shell has been updated and is now implemented as a stand-alone dialog known as the Workbench Scripting Shell. ([Bug#49298](#))

Bugs fixed:

- Double clicking on any main tab, just below the main menu toolbar, caused MySQL Workbench to generate an exception. ([Bug#50562](#))
- If text was typed into the search bar in the SQL Editor, and the sidebar button clicked twice, MySQL Workbench crashed. This only happened on Mac OS X. ([Bug#50560](#))
- In the **COLUMNS** tab of the Table Editor, attempting to change the column datatype using the drop-down list caused MySQL Workbench to crash. ([Bug#50546](#), [Bug#50598](#), [Bug#50527](#))
- The Portrait and Landscape icons were missing from the Page Setup dialog. ([Bug#50529](#))
- It was not possible to print EER Diagrams or schemata. The [FILE](#), [PRINT](#) option was grayed out, and [Ctrl+P](#) did not have any effect. ([Bug#50528](#))
- On an EER Diagram, when a relationship was placed using the toolbar button **PLACE A RELATIONSHIP USING EXISTING COLUMNS**, if the relationship was subsequently checked in the Foreign Keys tab of the Table Editor, it was found to contain incorrect values for foreign key names, for example, fk_%dcolumn%1. It was apparent that the placeholder had not had its value correctly substituted. ([Bug#50492](#))
- MySQL Workbench reported the remote server as being down, in the Database Server Status section of the Administrator, even though the server was in fact running, and queries could be successfully run against the database using MySQL Workbench. ([Bug#50453](#))
- When the menu item [FILE](#), [PAGE SETUP](#) was selected from the main menu, MySQL Workbench crashed. ([Bug#50315](#))

- When performing a data dump in the Administrator, the operation failed with an exit code 7.

MySQL Workbench has been changed so that it will generate an error if the `mysqldump` executable cannot be found. ([Bug#50184](#))

- Client-side sorting always sorted on an alpha basis, regardless of data type. This meant numerical values were not sorted into the order expected. ([Bug#50158](#))
- With an EER Diagram open, the FILE, PAGE SETUP menu item was unavailable. ([Bug#49863](#))
- When working through the Manage DB Connections wizard, the prompt to **ENTER SSH PASSWORD** appeared as a sheet behind the modal dialog box for the wizard. In some situations, it was not possible to see the **ENTER SSH PASSWORD** sheet, and it was not possible to click any buttons on, or close, the modal wizard.

This meant the user had to move the modal dialog box in order to see the **ENTER SSH PASSWORD** sheet, but it was not obvious that this was possible, because the window decorations indicated that the modal dialog could not be focused.

MySQL Workbench has now been changed so that a dialog is used for Manage DB Connections when creating a new connection, rather than a wizard. ([Bug#49810](#))

- The **SYNCHRONIZE MODEL WITH DATABASE** wizard generated a script that erroneously dropped schemata that had been selected to be ignored. ([Bug#49587](#))
- On the **GENERAL** tab of the configuration file editor, if a value was set for **TEMP DIRECTORY**, and the changes applied, then the value set was not displayed in the preview, implying it would not be set in the server configuration file. ([Bug#49423](#))
- When a MySQL Workbench unhandled exception occurred and the exception dialog was displayed, if the user clicked **QUIT**, and there were unsaved changes, a new dialog was displayed, warning of unsaved changes. If **CANCEL** was clicked on this dialog, MySQL Workbench would exit, which was not the expected behavior, as changes would then be lost. If **YES** was clicked on this dialog, to save changes, then MySQL Workbench crashed. Overall, the behavior of the dialogs was confusing.

A new error dialog has been introduced that changes the handling. It presents the user options to go to the bug report page, copy debug information to the clipboard and to close the dialog. ([Bug#49304](#))

- In the Administrator, when an attempt was made to import multiple tables from the same project folder, MySQL Workbench only imported the first table and then stopped, reporting the import process as finished. ([Bug#49217](#))
- In the SQL Editor, if an attempt was made to change the number of partitions in a table, using Alter Table, the SQL code produced did not contain the necessary `ALTER TABLE` statement to effect this change. ([Bug#49054](#))
- If the MySQL Server was stopped outside of MySQL Workbench then the server status displayed in the Administrator did not update correctly until the Startup tab was clicked. The log file did however correctly note the change in status. ([Bug#48966](#))
- When exporting a model to a single file using FILE, EXPORT, EXPORT AS SINGLE PAGE PDF or EXPORT AS SINGLE PAGE POSTSCRIPT FILE, the table positions contained in the file were incorrect. ([Bug#47384](#))
- If a model contained a table which used a user defined type for a column, then when the model was forward engineered an erroneous `COLUMN CHANGE` statement was generated for the column. This only happened for user defined types without additional arguments. For example, a user defined type using `INTEGER(11)` would not create a `COLUMN CHANGE` statement, but using `INTEGER` would. ([Bug#45834](#))

A.1.4. Changes in MySQL Workbench 5.2.14 (21st January 2010 beta)

Fourth public Beta release. This section documents all changes and bug fixes that have been applied since the release of 5.2.13.

Bugs fixed:

- If a multiline configuration option in the MySQL server configuration file was removed using the configuration editor of the Administrator tab, then MySQL Workbench generated an error. ([Bug#50470](#))
- In the SQL Editor, the Alter Table dialog created incorrect DDL for changes to the partitioning. ([Bug#49055](#))
- In the Physical Schemata section of the MySQL Model tab, the tables were not displayed in alphabetical order. ([Bug#47143](#))

A.1.5. Changes in MySQL Workbench 5.2.13 (Not released beta)

This section documents all changes and bug fixes that have been applied since the release of 5.2.12.

Bugs fixed:

- Selecting the HELP, CHECK FOR UPDATES menu item opened the web browser and displayed an error:

```
Not Found  
The requested URL /version-check.php was not found on this server.  
Apache Server at wb.mysql.com Port 80
```

([Bug#50415](#))

- When building MySQL Workbench 5.2.11 from source on CentOS 5.4, the following error occurred:

```
python_context.cpp:388: error: invalid conversion from 'const char*' to 'char*'  
python_context.cpp:388: error: initializing argument 1 of 'PyObject*  
PyImport_ImportModule(char*)'  
make[4]: *** [python_context.lo] Error 1  
make[4]: Leaving directory `/home/install/mysql-workbench-oss-5.2.11/library/grt/src'  
make[3]: *** [all-recursive] Error 1  
make[3]: Leaving directory `/home/install/mysql-workbench-oss-5.2.11/library/grt'  
make[2]: *** [all-recursive] Error 1  
make[2]: Leaving directory `/home/install/mysql-workbench-oss-5.2.11/library'  
make[1]: *** [all-recursive] Error 1  
make[1]: Leaving directory `/home/install/mysql-workbench-oss-5.2.11'  
make: *** [all] Error 2
```

Note, from MySQL Workbench 5.2.13, binary and source packages for CentOS 5.4 will be available. Due to the dependence of MySQL Workbench on certain packages that are older in CentOS 5.4 than in other operating systems, MySQL Workbench has some minor features, such as tool tips, disabled. ([Bug#50360](#))

- In the SQL Editor, if a routine was edited, and the changes applied, then the changes were automatically reverted if the apply failed. This meant that any changes the user had made had to be typed in again. ([Bug#49790](#))
- In the SQL Editor, if an object was created or renamed, that change was not reflected in the user interface. This change fixes the problem for Linux and Mac OS X. See also [Bug#50424](#) for the Windows platform. ([Bug#49454](#))
- Invalid values were written to the MySQL server configuration file by the configuration editor. MySQL Workbench also added new entries to the configuration file, rather than update the existing entries. As a result of this the server failed to start. ([Bug#49420](#))
- If the server configuration file contained more than one entry for `replicate-do-db`, then the configuration editor on the Admin tab only displayed the second entry, rather than both. ([Bug#49299](#))
- In the configuration editor in the Admin tab, the listboxes for displaying units, such as for the **SORT BUFFER SIZE** option, did not show 'G', which represents Gigabytes. This bug was fixed in 5.2.10.

It was subsequently decided to standardize all units around uppercase characters. In particular the 'K' was changed to 'K'. This change was made in version 5.2.13. ([Bug#49013](#))

A.1.6. Changes in MySQL Workbench 5.2.12 (Not released beta)

This section documents all changes and bug fixes that have been applied since the release of 5.2.11.

Functionality added or changed:

- In the SQL Editor, if a table was edited with the Edit Data facility, and table values changed, no warning was given if an attempt was made to close the tab without having first applied changes using the apply button.
MySQL Workbench has been changed so that if the user attempts to close a live editing tab, without having applied any changes made, then a warning dialog is displayed. ([Bug#49925](#))
- MySQL Workbench has been changed to allow a connection to be created where the password does not have to be stored. In this case, MySQL Workbench prompts the user to enter the correct password when the connection is established. ([Bug#49409](#))
- Undo operations are no longer tracked in the editors that work with live databases, such as table editors within the SQL Editor. ([Bug#49284](#))

Bugs fixed:

- **Security Fix:** The password for the connected MySQL Server was exposed by the SQL Administrator in MySQL Workbench. The password was displayed in plain text form in the Startup Message Log on the Startup tab of the Admin page. ([Bug#43287](#))
- In SQL Editor, `SELECT` statements that called functions, for example `SELECT md5('abcd')`, did not give the expected result. ([Bug#50248](#))
- In the Table Editor, the drop down listbox for selecting the table engine type contained a reference to `calpont`. This should now be `infinidb`. ([Bug#50159](#))
- In Mac OS X, the context-sensitive menu was missing from the live editing tab of the SQL Editor. Right-clicking in the editing tab simply displayed a Mac OS X menu, rather than the context-sensitive menu to allow actions such as copying, deletion and setting selections to NULL. ([Bug#50114](#))
- Custom Python modules failed to load and were not displayed in the MySQL Workbench Plugins menu.

In MySQL Workbench 5.2 the syntax used has changed from `@wbplugin` to `@ModuleInfo.Plugin`. ([Bug#50108](#))

- On the Model page, if some text was entered into a note in Model Notes, then the model saved and closed, on reloading the note would be empty.

This also happened for a script added in SQL Scripts. After the model was saved and reloaded, the script would be empty. ([Bug#50108](#))

- The script code generated from a model for routines did not quote schema names in `USE` statements. ([Bug#50051](#))
- Lower panels such as the Table Editor remained open, even when the user returned to the Home screen. ([Bug#49926](#))
- In the SQL Editor, when using the `QUERY, EXPORT RESULTS...` menu item, the file extension was omitted when selection of an existing output file was made. ([Bug#49870](#))
- After selecting Edit Table Data from the Home screen, and launching the Edit Table Data wizard, MySQL Workbench crashed if a connection was selected and then the NEXT button clicked. ([Bug#49864](#))
- In the Manage DB Connections dialog, when attempting to create a Standard TCP/IP over SSH connection, errors occurred when testing the connection in the case where a key file had been specified. If the path to the key file was quoted the error was:

```
Cannot open SSH Tunnel:  ERROR Invalid request
```

If the same path was not quoted the error generated was:

```
Could not connect SSH tunnel:  ERROR [Errno 22] invalid mode ('r') or filename:  
'D:\\Documents and Settings\\tf221715\\.ssh\\id_dsa'
```

It should be noted that currently only key files using the OpenSSH format are supported. ([Bug#49812](#))

- In the SQL Editor, if a Stored Procedure was executed and then immediately executed again, the following error was generated:

```
Error code 2014  
Commands out of sync; you can't run this command now
```

This problem only affected MySQL Workbench running on the Mac OS X platform. ([Bug#49553](#))

- Right-clicking on a Routine Group and then selecting `COPY SQL TO CLIPBOARD` did not work. No code was copied to the clipboard. ([Bug#49440](#))
- Opening a model file, closing it and quickly reopening it caused MySQL Workbench to hang. ([Bug#49428](#))
- Clicking the TEST CONNECTION button in the Manage DB Connections dialog caused MySQL Workbench to lock up. The error generated on the console was:

```
18743 INFO Connecting to SSH server at 127.0.0.1:22...  
Unhandled exception in thread started by <bound method Tunnel._threadloop of  
<__main__.Tunnel instance at 0x9acea8>>
```

This happened for the connection method Standard TCP/IP over SSH. ([Bug#49419](#))

- Two SSH tunnels were opened for a single connection to an SSH server. The user was also prompted for a password when a new connection was established, even when the existing tunnel was reused. ([Bug#49412](#))

- SSH Tunnels opened by MySQL Workbench were never closed. SSH Tunnels created when clicking TEST CONNECTION in the **MANAGE DB CONNECTIONS** dialog, were not closed. This was also the case when the SQL Editor and Server Administration tabs were closed. ([Bug#49411](#))
- In the Partitioning tab of the alter table dialog it was possible to enter a number greater than 10 into the Partition Count text-field. However, when these changes were applied the DDL generated substituted any value entered that was greater than 10 with the value 2. ([Bug#49380](#))
- MySQL Workbench generated an unhandled exception when trying to add a new column using the Alter Table facility in the SQL Editor. After the `VARCHAR(45)` column was added, without selecting any column checkboxes, the exception occurred when the APPLY button was clicked. ([Bug#49364](#))
- If a database was reverse engineered from a source, and a table renamed, then although this change was detected by MySQL Workbench, no DDL code was generated if an attempt was then made to synchronize this change with the source. ([Bug#49313](#))
- In the SQL Editor, when a schema was dropped, it was still displayed in the Overview tab and in the schemata explorer of the side panel. Further, MySQL Workbench did not provide any confirmation that the schema had been dropped successfully or otherwise. ([Bug#49282](#))
- The Alter Schema dialog in the SQL Editor allowed the schema name to be changed via the **NAME** textfield, even though doing this is not supported at the server level. ([Bug#49281](#))
- In the SQL Editor, if an item in the History tab was right-clicked, and the menu option APPEND SELECTED ITEMS TO SQL SCRIPT selected, then MySQL Workbench generated an unhandled exception:

```
glibmm-ERROR **:  
unhandled exception (type std::exception) in signal handler:  
what: vector::_M_range_check
```

([Bug#49245](#))

- In the **CREATE A NEW SERVER INSTANCE** wizard, if the checkbox **ENABLE SSH LOGIN BASED ADMINISTRATION** was cleared, the wizard still prompted the user to enter an SSH password when the **NEXT** button was clicked. ([Bug#49226](#))
- In the Server Status section of the Admin page, the **SYSTEM** graphic showed CPU usage at 50%, while the corresponding text label showed CPU usage at 0%. The graphic and label remained out of synchronization for around 15 seconds. ([Bug#49212](#))
- In the SQL Editor if an `EDIT table` operation was being carried out, it appeared that data in the table was not updated, and also a row already inserted would be erroneously deleted when attempting to add a new row of data, when changes were applied.

MySQL Workbench has been changed so that if an error occurs during changes being applied, an error dialog is displayed showing any errors that occurred. Also, the status text is now updated to say “Apply Failed”, rather than “Changes Applied”. ([Bug#49179](#))

- The Manage DB Connections dialog did not prevent duplicate connection names or an empty string as a valid connection name. ([Bug#49150](#))
- The toolbar buttons on the Home screen did not reflect the order of the action items in the Workspace. The Open Server Profile button and Open DB Connection Manager buttons needed to exchange positions. ([Bug#49149](#))
- In the Logs tab of the Administrator, the Oldest and Newest buttons appeared to have reversed functionality. Oldest displayed the most recent timestamps and Newest displayed the oldest. ([Bug#49065](#))
- In the Manage DB Connections dialog, many textfield labels appeared truncated. ([Bug#49052](#), [Bug#49378](#))
- The ID for a connection was displayed in the Connections tab of the Admin page, even after the corresponding administration session had been closed. ([Bug#48989](#))
- The Connections drop-down listbox in the Manage Server Instances dialog did not display the username for socket and pipe connections. For example it displayed `localhost - @:0 <Local Socket/Pipe>`, rather than `localhost - root:@:0 <Local Socket/Pipe>`. ([Bug#48969](#))
- Various errors occurred when attempting to modify the server configuration using the Configuration tab of the Administrator:

```
Unhandled exception: global name 'tempfile' is not defined
```

```
Unhandled exception: [Errno 22] Invalid argument:  
'C:\\users\\\\tax\\\\appdata\\\\local\\\\temp\\\\tmpo47ttn'
```

Also, if the Logs tab was selected in order to view the server log files the following error occurred:

```
Unhandled exception: Error executing 'Select @@  
log_output': Unknown system variable 'log_output'
```

([Bug#48906](#))

- In the **FORWARD ENGINEER AN ALTER SCRIPT** wizard, the textfields and associated buttons for selecting input and output files were not positioned correctly within the window, when the selected filename was long. Although the files could be selected when the wizard was first launched, if the wizard was launched subsequently, the text fields would be filled with the previously selected filenames, causing the textfields and Browse buttons to be located beyond the borders of the window.

A similar problem also affected the Forward Engineer SQL Script wizard. ([Bug#48222](#))

- In the Partitioning tab of the Alter Schema dialog, it was not possible to create a single partition. If a partition count of 1 was specified this was actually set to 2 in the generated DDL. ([Bug#48114](#))
- When performing a Synchronize Model operation, all FK relationships in the EER diagram that had the setting `draw split` were redrawn as `fully visible`. ([Bug#47767](#))
- When running MySQL Workbench from the command line, it could not open model files where the path contained a space. ([Bug#46297](#))

A.1.7. Changes in MySQL Workbench 5.2.11 (18th December 2009 beta)

Third public Beta release. This section documents all changes and bug fixes that have been applied since the release of 5.2.10.

Functionality added or changed:

- MySQL Workbench had confusing and erroneous behavior when attempting to handle multiple model tabs. It has now been changed so that if a model is currently loaded, and a new model or saved model needs to be loaded, the current model will be closed first before then opening the new model. ([Bug#49422](#))
- In the configuration editor, it was not possible to find, and therefore set, the configuration variable “old”.

This has now been added to the General tab in the configuration editor of the Admin tab. ([Bug#49039](#))

- It was impossible to use a function call as a field value in an editable result set, such as for a result set in the Query Editor or in the Inserts tab in the Table Editor. This was because MySQL Workbench automatically escaped string parameters passed to the function call. For example, if an attempt was made to enter into a field a function such as `md5('fred')`, MySQL Workbench would generate the SQL code `md5('fred')`.

MySQL Workbench has now been changed so that it is possible to enter a function, or any other expression, into a field using the `\func` prefix. For example, `\func md5('fred')` can be entered. MySQL Workbench will now ensure that the string 'fred' is not escaped. ([Bug#38600](#))

Bugs fixed:

- **Security Fix:** When using the Data Dump facility in the Admin screen, the full `mysqldump` command, including the password used, was written to the logs. ([Bug#49294](#))
- In the SQL Editor the SQL Statements area could only display ten lines of code and was not resizeable. ([Bug#49788](#))
- In an EER diagram, if the model was synchronized with a live database, then any foreign key relationship lines that were set to hidden became visible. ([Bug#49631](#))
- In the Home screen Workspace, in the central panels listing connections, models and server instances, the items in the panels appeared to be rendered as links. However, clicking on these “links” had no effect, and the items could only be loaded by double-clicking.

MySQL Workbench has been changed to remove the link effect, and these items can only be loaded by double-clicking. ([Bug#49623](#))

- When MySQL Workbench silently reconnected to a server after a communication failure, it failed to reinitialize the connection correctly. As a result of this failure Autocommit was silently enabled. If a user was working with transactional tables, the commit and rollback toolbar buttons appeared to work, although they had no effect. ([Bug#49462](#))

- If a connection failed MySQL Workbench reconnected silently. This caused problems with transactions. If changes were made to a table, before a `COMMIT`, and the connection lost, then MySQL Workbench would reconnect silently and allow the user to `COMMIT`. However, it did not warn that this `COMMIT` was on a new connection, and that the `COMMIT` would have no effect. ([Bug#49461](#))
- In the Advanced tab of the configuration editor, the option `DELAY KEY WRITE` had a file selector button associated with it. This should have been a drop down list box offering the values `ON`, `OFF`, `ALL`. ([Bug#49424](#))
- When using the **CREATE A NEW SERVER INSTANCE WIZARD** the panel used to enter the SSH password was hidden by the wizard. ([Bug#49416](#))
- If the Home screen was closed and then an attempt made to reopen it from the `VIEW`, `HOME` main menu option, then MySQL Workbench crashed. ([Bug#49388](#))
- MySQL Workbench did not have the ability to toggle Autocommit mode, or a facility to explicitly start a transaction, in the SQL Editor. This was in contrast to Query Browser, which did support such facilities. ([Bug#49384](#))
- When using the Reverse Engineer SQL Script to import the Sakila script file, `sakila-data.sql`, MySQL Workbench occasionally crashed. ([Bug#49381](#))
- In the History tab of the SQL Editor, it was possible to simultaneously select multiple entries in the Time panel. However, this did not seem to serve any useful purpose as the content of the SQL column only displayed the code for a single entry. ([Bug#49375](#))
- In the Schemata Palette of the SQL Editor the default schema drop down selector did not work if the schema name contained a '.' character. ([Bug#49373](#))
- When using the SQL Snippets palette, right-clicking on a snippet, and then selecting the menu item `INSERT TEXT INTO SQL AREA`, replaced all text in the SQL Statements area. This happened after another snippet had previously been inserted, as the default state was to leave all code in the SQL Statements area selected. ([Bug#49370](#))
- The `AUTHORS` file in the MySQL Workbench distribution was empty. ([Bug#49341](#))
- In the Overview tab of the SQL Editor the drop database button on the toolbar did not work correctly. If pressed the dialog presented did not contain the name of the current database in its message, and generated text such as:


Note the empty string where the database name should have been. Further, the text on the dialog button `DROP` was also missing the database name:



([Bug#49330](#))

- The path to the an external `mysqldump` tool set in the MySQL tab of the Preferences dialog was ignored by MySQL Workbench. This prevented the Data Dump facility in the Admin screen from working correctly, as the required tool could not be found. ([Bug#49319](#))
- The live data editor of the SQL Editor, which was launched by double-clicking on a table in the Overview tab, did not behave correctly. If a column value was changed, and the cursor remained in the edited cell, then if the `APPLY MADE CHANGES TO DATA SOURCE` toolbar button was clicked, the contents of the cell reverted to its value prior to editing. However, other cell values that had been edited were correctly saved. ([Bug#49301](#))
- In the SQL Editor, when editing live table data, there was no right-click menu item to delete a row. ([Bug#49300](#))
- When importing data via the Data Dump tab of the Admin screen, MySQL Workbench did not notify the user of failed imports, because it could not detect failed imports. ([Bug#49297](#))
- When a export was performed via the Data Dump tab of the Admin screen, and the export failed, the resultant SQL file was not deleted. ([Bug#49296](#))
- In the Overview tab of the SQL Editor there was no scrolling facility available. This meant that if the model contained a large number of schema objects the panel area was quickly filled, and it was not possible to view all of the objects without the ability to scroll. ([Bug#49290](#))
- In the SQL Editor, using the Alter Routine wizard to generate a script to modify an existing routine in the live database results in errors, due to incorrect script code being emitted. For example, the statement `DELIMITER $$` was missing from the start of the script. ([Bug#49289](#))

- When an attempt was made to load the `sakila.sql` script file in the SQL Scripts section of the Model tab, MySQL Workbench crashed with the following error:

```
glibmm-ERROR **:  
unhandled exception (type std::exception) in signal handler:  
what: File '/home/kolbe/Downloads/sakila-db/sakila-data.sql' contains invalid UTF-8  
data.
```

([Bug#49242](#))

- In the Overview tab of the SQL Editor, the toolbar buttons to add and drop a schema did not function. Clicking the buttons appeared to have no effect. ([Bug#49240](#))
- In the Overview tab of SQL Editor, the toolbar buttons representing large icons, small icons, list, add, and delete did not function correctly. ([Bug#49239](#))
- The Configuration tab on the Admin screen had a text label “Edit my.cnf File”. This text was not appropriate as on Windows the configuration file is called `my.ini`. ([Bug#49237](#))
- In the configuration editor, on the Admin tab, the operation of the Apply and Cancel buttons did not work correctly. If an option was selected, and then Cancel selected in the view changes dialog, and then Apply clicked again, the selected option was incorrectly listed twice in the configuration file. ([Bug#49236](#))
- In the **CREATE NEW SERVER INSTANCE** wizard, on the MySQL Config File page, clicking the CHECK PATH button generated an exception. ([Bug#49228](#))
- Even though the server had been stopped, the Server Status graphical panel displayed values for active connections, traffic and key efficiency, giving the impression that the server was still active. ([Bug#49225](#))
- The server health graphs Connection Usage and Traffic, in the Server Status panel of Admin tab, appeared to indicate the server was operating at 100% capacity, even when this was not the case.

MySQL Workbench has been changed to use variable scaling, rather than linear scaling, for these graphs. ([Bug#49214](#))

- On the Admin screen the labels for the monitoring graphs were difficult to read. Further, the value for Traffic was sometimes displayed as a negative number. ([Bug#49211](#))
- In the **LOG FILES** tab of the configuration editor, selected from the Admin tab, the option **WRITE LOGS TO** did not work correctly. If either the **FILES** or **TABLES** options were selected from the option list, the MySQL Server would not subsequently start. This was because the options should have been **FILE** and **TABLE**. ([Bug#49123](#))
- When using the Data Dump facility on the Admin screen, the export process appeared to hang when exporting with the **EXPORT TO BACKUP PROJECT FOLDER** radio button selected. ([Bug#49113](#))
- When using the Create Table wizard in the SQL Editor, the resulting dialog contained certain options that could not be deselected once selected. For example, the Merge Method option featured a drop down list with three options: Prevent Inserts, First Table, Last Table. Note that once one of these options was selected, there was no way to clear this selection, as there was no facility to select anything other than one of these three options. ([Bug#49048](#))
- In the **GENERAL** tab of the configuration editor, changing the data directory using the file chooser control led to an invalid directory being introduced into the MySQL configuration file. This prevented the MySQL Server from starting. ([Bug#49036](#))
- In the General tab of the configuration editor in the Admin screen, the option **Default table type** provided an incomplete, and incorrectly capitalized, list of storage engines/table types. ([Bug#49010](#))
- In the Configuration tab of the Admin screen, any changes made and applied were not reflected in the configuration file of a local MySQL Server installation. ([Bug#49008](#))
- In the Server Status section of the Admin screen, the values for CPU status appeared to be erroneously multiplied by 100, and the Memory status appeared to be continually 0. ([Bug#48994](#))
- In the Document Properties dialog the created date and last changed date values were reversed. ([Bug#48104](#))
- When a schema containing tables with foreign key indexes was synchronized with a live server, MySQL Workbench attempted to erroneously drop and recreate at least one of the foreign key indexes. ([Bug#47766](#))
- After synchronizing a model with a live database, and saving the resulting modified model, MySQL Workbench crashed. On restarting and attempting to reopen the model file, MySQL Workbench generated the following error:

```
Error unserializing GRT data inserting null value to not null list
```

(Bug#47518)

- In an EER diagram, if relationship links were laid out as desired, and then a synchronization with the live server carried out, the relationship links were repositioned by MySQL Workbench. (Bug#47234)
- If a relationship link was selected in the EER Diagram view, and the menu item PLUGINS, OBJECTS, COPY SQL TO CLIPBOARD was selected, an error was generated.

Note that in version 5.2.11 this menu option is correctly disabled for these objects. (Bug#39556)

A.1.8. Changes in MySQL Workbench 5.2.10 (1st December 2009 beta)

Second public Beta release. This section documents all changes and bug fixes that have been applied since the release of 5.2.9.

Functionality added or changed:

- MySQL Workbench now has the command line option `--version`, which is used to display the version of the application, when launching the application from the command line. (Bug#49136)
- The HELP, WORKBENCH PRODUCT PAGE menu item launched the default web browser, but displayed an outdated product page. This was due to a web server configuration issue. MySQL Workbench now displays the correct, up-to-date product page. (Bug#49066)

Bugs fixed:

- When using ALTER TABLE in the SQL Editor, if a column name was changed, and the **APPLY** button clicked, the **APPLY OBJECT CHANGES** wizard was launched. The **REVIEW CHANGES** page indicated that the script that would be applied would drop the altered column then add a new column with the new name. This led to column data being lost, as the script should instead have used CHANGE COLUMN, to change the name of the column. (Bug#49286)
- When using ALTER TABLE in the SQL Editor, if a table name was changed, and the **APPLY** button clicked, the **APPLY OBJECT CHANGES** wizard was launched. The **REVIEW CHANGES** page indicated that the script that would be applied would incorrectly create a new table as a duplicate of the table being renamed, rather than use ALTER to rename the table. (Bug#49275)
- In the configuration editor, in the Admin tab, the DISCARD button simply unchecked all options, rather than just those that had been checked during the current editing session. This made the button appear to have a “clear all” function, rather than the expected “revert changes” function. (Bug#49234)
- When using the Forward Engineer SQL CREATE Script wizard, if the checkbox **OMIT SCHEMA QUALIFIER IN OBJECT NAMES** was selected, then the script code for views was omitted. (Bug#49153)
- In the **ADMINISTRATIVE ROLES** tab of **SERVER ACCESS MANAGEMENT**, selection of Roles did not work correctly. For example, selecting the **DBA** checkbox, and then deselecting it, caused all roles to be deselected. Further, using the REVERT button resulted in the selected user disappearing from the **USER ACCOUNTS** panel. (Bug#49071)
- MySQL Workbench did not handle signed and unsigned integers correctly. For example, if performing a synchronization between a model and a live database where the only difference was a column was declared to be of type INTEGER in one case and UNSIGNED INTEGER in the other, the difference would not be detected and the ALTER script would imply the databases were the same. (Bug#49063)
- A MySQL configuration file caused MySQL Workbench to crash, when an attempt was made to view it in the configuration editor of the Admin tab. The error generated was:

```
Unhandled exception: 'bool' object has no attribute 'strip'
```

(Bug#49060, Bug#49602)

- In the SQL Editor, two buttons on the toolbar had no tooltips. These were the green check or tick mark, and the red back arrow buttons. Further, they did not seem to become enabled or disabled according to the context, making it difficult to determine their intended function. (Bug#49059)
- On the **MySQL CONFIG FILE** page of the New Server Profile wizard, clicking **CHECK SECTION** generated an exception if there were options in the MySQL configuration file that did not have values assigned:

```
Check if mysqld section exists in /tmp/my.cnf
** Message: function call error
```

```
Traceback (most recent call last):
  File "/usr/lib/mysql-workbench/modules/wb_admin_grt.py", line 292, in
testInstanceSettingByName
    parser.read([config_file])
  File "/usr/lib/python2.5/ConfigParser.py", line 267, in read
    self._read(fp, filename)
  File "/usr/lib/python2.5/ConfigParser.py", line 490, in _read
    raise e
ConfigParser.ParsingError: File contains parsing errors: /tmp/my.cnf
[line  2]: 'log-bin\n'
glibmm-ERROR **:
unhandled exception (type std::exception) in signal handler:
what: error calling WbAdmin.testInstanceSettingByName: see output for details
```

([Bug#49057](#), [Bug#47954](#))

- On the Admin screen, the server configuration option **KEY BUFFER**, was located under the **GENERAL** tab, rather than the **MYISAM PARAMETERS** tab, even though it was a MyISAM-only option. ([Bug#49017](#))
- In the configuration editor in the Admin tab, the listboxes for displaying units, such as for the **SORT BUFFER SIZE** option, did not show 'G', which represents Gigabytes. This bug was fixed in 5.2.10.

It was subsequently decided to standardize all units around uppercase characters. In particular the 'k' was changed to 'K'. This change was made in version 5.2.13. ([Bug#49013](#))

- In the SQL Editor, the database explorer of the Schemata palette in the side panel did not work correctly. Incorrect behavior included random collapsing of expanded databases and occasional crashing. ([Bug#48981](#))
- When using the data modeler, MySQL Workbench extended vertically to the maximum size of the screen, overlapping the dock, and thereby making it impossible to use. ([Bug#48976](#))
- In the **MANAGE SERVER INSTANCES** dialog, if the **SERVER INSTANCES** list pane was empty, then on creating a new Server Instance, it was not possible to change the instance's name. The instance had to be created, then the dialog closed. On reopening the dialog the instance could be renamed. ([Bug#48967](#))
- The Generate Catalog Diff Report feature did not allow the comparison of imported scripts, as selecting the radio button for this option had no effect. ([Bug#47230](#))
- If a collation was changed for a table in the Table Editor, there was no facility to then set this back to the schema default.

A new entry has been now been added to the collation selection listbox - **SCHEMA DEFAULT**. ([Bug#46513](#))

A.1.9. Changes in MySQL Workbench 5.2.9 (Internal release only beta)

Internal release. This section documents all changes and bug fixes that have been applied since the release of 5.2.8.

Functionality added or changed:

- If, while editing data in the Inserts Editor, the **ESC** key was accidentally pressed, the Inserts Editor would close without warning and all data entered to that point would be lost.

The improved Inserts Editor does not display this characteristic. Pressing the **ESC** key will have no effect. ([Bug#48452](#))

Bugs fixed:

- MySQL Workbench crashed when the **IMPORT/EXPORT SERVER DATA** Action Item on the Home screen was clicked. ([Bug#49064](#))
- Clicking on the Action Item, **MANAGE SECURITY**, on the Home screen generated the following exception:

```
Traceback (most recent call last):
  File "C:\Program Files\MySQL\MySQL Workbench 5.2 OSS\modules\wb_admin_grt.py", line
199, in openSecurityManager
    tab.wait_server_check(4)
  File "C:\Program Files\MySQL\MySQL Workbench 5.2 OSS\modules\wb_admin_grt.py", line 95,
in wait_server_check
    while tab.configuration.last_is_running_check is None and time.time() - t < timeout:
NameError: global name 'tab' is not defined
```

Further, if this Action Item was clicked again then MySQL Workbench crashed. ([Bug#49061](#))

- The Forward Engineer SQL Script failed to generate SQL code when the checkbox **GENERATE INSERT STATEMENTS FOR TABLES** was selected, and the tables contained rows. ([Bug#49046](#))
- In the **OVERVIEW** tab of the SQL Editor, representing a “live” view of the database currently connected to, if an attempt was made to edit a view, the resulting script generated unnecessary, and in fact dangerous, `DROP TABLE` statements. This had the potential side-effect that a table that coincidentally had the same name as the view, would be dropped with ensuing data loss. ([Bug#49041](#))
- In the **CONFIGURATION** tab of the Server Administrator screen, if log file configuration changes were applied, MySQL Workbench hung if the password request dialog was cancelled. ([Bug#49037](#))
- The Workbench Configuration editor interface contained a backtick character rather than a single apostrophe. ([Bug#49014](#))
- In the **ADVANCED** tab of the **ADMIN** screen, the checkbox **SQL MODE** had an erroneous file chooser button associated with it. The button should not have been there. ([Bug#49012](#))
- MySQL Workbench did not parse the following entry in the `my.ini` file correctly:

```
default-storage-engine=INNODB
```

This resulted in the default storage engine not being detected correctly by MySQL Workbench, although this entry was correctly recognized by the MySQL server. However, MySQL Workbench did parse the following entry correctly, due to the correct capitalization being used:

```
default-storage-engine=InnoDB
```

([Bug#49007](#))

- After creating a new Server Instance from the **HOME** screen, and then clicking on the **LOGS** tab in the Admin screen, the following exception was generated:

```
Unhandled Exception: Error executing 'SELECT @@log_output':  
Unknown system variable 'log_output';
```

([Bug#49004](#))

- Clicking on the Action Item **MANAGE SECURITY** on the Home screen resulted in MySQL Workbench crashing. ([Bug#48990](#))
- Any **TEXT** columns in the Table Editor, or **TEXT** columns in the SQL Editor results tabsheet, were displayed as **BOOLEAN** values, either 1 or 0. ([Bug#48982](#))
- MySQL Workbench crashed on opening a model file. This appeared to be due to MySQL Workbench causing corruption in the model file. ([Bug#48891](#))
- In the EER Diagram view, the TOGGLE GRID and ALIGN OBJECTS TO GRID toolbar buttons were not rendered correctly when in the selected state. ([Bug#48822](#))
- The Inserts Editor did not display columns of type **ENUM**. ([Bug#48288](#))
- On the **FILTER OBJECTS** page of the Forward Engineer SQL Script wizard, clicking SHOW FILTER, selecting multiple objects from the left panel, and then clicking >, would result in only the first of the selected objects being moved to the right hand panel. ([Bug#48116](#))
- Working through the **SYNCHRONIZE MODEL WITH DATABASE** wizard did not result in the live database being updated. ([Bug#47953](#))
- Using UTF-8 accented characters in the Stored Routine DDL editor caused the DDL parser to raise a syntax error and refuse to save the routine. ([Bug#47730](#))
- The Forward Engineer SQL Script wizard did not generate the export script. On the **REVIEW SQL SCRIPT** page of the wizard the review panel was blank. Further, clicking COPY TO CLIPBOARD generated the following exception:

```
Unknown Exception caught in c:\documents and settings\mysqldev\my documents\visual  
studio 2008\projects\workbench52\backend\windows\wbprivate.wr\src\Wb.h at line 1085.
```

([Bug#47482](#))

- When using the Forward Engineer SQL Script wizard, the script generated for a model containing multiple schemata, only included the `CREATE SCHEMA` statement for the first schema. ([Bug#47202](#))
- Running the **FORWARD ENGINEER AN ALTER SCRIPT** wizard resulted in an ALTER script that dictated changes were required to tables, even in the case where no such changes were necessary. ([Bug#47063](#))
- Attempting to use [DATABASE, GENERATE CATALOG DIFF REPORT](#) resulted in a Segmentation Fault. ([Bug#46810](#))
- The Forward Engineer SQL Script wizard did not generate correct delimiter syntax for routines. For example, it generated the following code:

```
DELIMITER //
//
CREATE PROCEDURE `sakila`.`film_not_in_stock`(IN p_film_id INT, IN p_store_id INT, OUT
p_film_count INT)
READS SQL DATA
BEGIN
    SELECT inventory_id
    FROM inventory
    WHERE film_id = p_film_id
    AND store_id = p_store_id
    AND NOT inventory_in_stock(inventory_id);
    SELECT FOUND_ROWS() INTO p_film_count;
END //
//
```

This resulted in errors when an attempt was made to run the generated script on MySQL Server. ([Bug#46505](#))

A.1.10. Changes in MySQL Workbench 5.2.8 (18th November 2009 beta)

First Beta release of 5.2. This section documents all changes and bug fixes that have been applied since the release of 5.2.7.

Bugs fixed:

- When a table was edited in the EER Diagram view, and **Ctrl+S** was pressed to save the model, the model file was not saved. ([Bug#48682](#))
- If an model object, such as a schema or table, was deleted or its creation was undone using the undo feature, while the object editor was open, then a crash occurred if the editor was then subsequently closed. ([Bug#48664](#))
- When running MySQL Workbench the following error occurred when an attempt was made to change the Windows screen resolution:

```
cairo error: out of memory
```

([Bug#48520](#))

- MySQL Workbench did not start correctly. On start up it generated the following error:

```
Microsoft Visual C++ Runtime Library
Runtime Error!
Program C:\Pro...
This application has requested the Runtime to terminate it in an unusual way.
Please contact the application's support team for more information.
```

([Bug#48389](#))

- Once a default value had been set for a column in the Table Editor, it was not possible to remove it. ([Bug#47085](#))
- The View text editor was overly aggressive in trimming excess whitespace from View definitions. If there was some hesitation in typing after having typed one or more spaces, the editor would trim whitespace back to the last non-space character typed. ([Bug#46894](#))
- In the EER Diagram view, the auto-resizing of tables did not work correctly, tables were too small to allow all columns to be visible. ([Bug#46806](#))
- When using the Fedora 10 RPM installation packages on Fedora 11, opening a database connection gave the following error:

```
Cannot Connect to Database Server
```

```
Connection 'antonia' could not be established: Database driver: Failed to open library '/usr/lib/mysql-workbench/mysqlcppconn.so'. Check settings.
```

The MySQL Workbench libraries were installed in `/usr/lib64/`, not `/usr/lib/`. ([Bug#46428](#))

- In the Manage DB Connections dialog, the text fields **USERNAME**, **PASSWORD**, and **DEFAULT SCHEMA** accepted text, but as the text was being entered only the top half of the characters typed was displayed. However, once the text had been submitted, the characters were displayed correctly. ([Bug#45106](#))
- Printing of an EER Diagram did not work correctly if a table vertically spanned multiple pages. In this case pages would be printed up to and including the page that contained the first part of the table that spanned multiple pages, but the pages containing the remaining parts of the table would not be printed. This problem typically occurred when a table had more fields than could comfortably fit on a single page. ([Bug#33919](#))

A.1.11. Changes in MySQL Workbench 5.2.7 (Internal release only alpha)

This section documents all changes and bug fixes that have been applied since the release of 5.2.6.

Functionality added or changed:

- In the Administrator tab of MySQL Workbench the Data Dump (Export/Import Data) feature now uses the `--comments` parameter when executing an import or export operation. ([Bug#23002](#))

Bugs fixed:

- In the **MANAGE SERVER INSTANCES** wizard, MySQL Workbench crashed if the NEW button was clicked. ([Bug#48347](#))
- On Ubuntu Linux, MySQL Workbench did not warn if a new model was to be saved as a file that already existed on the drive. ([Bug#48345](#))
- The `Use compression protocol` option was not working correctly when set in the **MANAGE DB CONNECTIONS** wizard.

When the checkbox **USE COMPRESSION PROTOCOL** was selected in the Advanced tab of the **MANAGE DB CONNECTIONS** wizard, the following error was generated when the connection was tested:

```
No bool value passed for CLIENT_COMPRESS
```

The same error occurred even if the **USE COMPRESSION PROTOCOL** checkbox was subsequently cleared. ([Bug#48202](#))

- When editing a stored routine or trigger in the Table Editor, the cursor would relocate to the top of the editing window if a key was not pressed for a few seconds.

Note

On Windows this issue was fixed in 5.1.18. On Mac OS X this issue was fixed in 5.2.4. On Linux this issue was fixed in 5.2.7.

([Bug#48156](#))

- Attempting to export the result of an SQL query to CSV using `QUERY`, `EXPORT RESULTS...` gave the error:

```
Failed to open template file: `/usr/share/mysql-workbench/modules/data/sqlide/CSV.tpl`
```

This occurred on a Fedora 11 AMD64 install, from RPM. The requested file was found in a different directory, `/usr/share/mysql-workbench/sqlide/`. The error did not occur after the file was moved to the required location. ([Bug#48156](#))

- In the Columns tab of the Table Editor, the tool tip for the AI column did not work. If the cursor was placed over the column no tool tip text was displayed. ([Bug#48129](#))
- If an EER Diagram tab was closed, it was not possible to reopen it by clicking on its icon on the model page. ([Bug#47545](#))
- In the GRT shell the `help` command, which is equivalent to `?`, crashed when no arguments were specified. ([Bug#47503](#))
- The script generated by `FILE`, `EXPORT`, `FORWARD ENGINEER SQL CREATE SCRIPT` dropped a schema if it existed and then re-

created it. This resulted in loss of existing data. ([Bug#47468](#))

- When an attempt was made to add a database connection the following error was generated:

```
Test connect failed
Couldn't load library libmysqlclient_r.so: libmysqlclient_r.so: cannot open shared object
file: No such file or directory
```

This only happened on Linux. ([Bug#47238](#), [Bug#48507](#))

- In the Inserts tab of the Table Editor, values entered for a `BIT` column were not saved. If values were entered and then the changes applied, the values in the `BIT` column disappeared. ([Bug#47100](#))
- When a model was loaded and an EER Diagram opened, the model navigator did not display the model correctly. ([Bug#46970](#))
- In the Table Editor, when trying to modify a column default value, errors occurred.
 - On a `TIMESTAMP` column it was not possible to enter a valid value of the format yyyy-mm-dd hh:mm:ss. This caused an error if forward engineering was attempted.
 - A `NULL` default value for a column could not be deleted.

([Bug#46878](#))

- In the `INSERTS` tab of the Table Editor, if a string that contained spaces was added as an insert, then the string was truncated to the text before the first space. ([Bug#46624](#))
- In the Reverse Engineer Database wizard it was not possible to select only one table to reverse engineer, and then proceed to the next step. ([Bug#45881](#))
- When running a plugin that called a function such as the following, the document's status was set to "unsaved", and an asterisk was displayed in the title bar, as soon as the file selection dialog opened. This occurred even for a new document, or an unmodified document.

```
function rfstest(obj)
    local path
    path = Workbench:requestFileSave("Caption", "SQL Files (*.sql)|*.sql")
end
```

([Bug#44813](#))

- Forward Engineer SQL CREATE Script generated schema qualifiers for Stored Procedure names regardless of the setting of the **OMIT SCHEMATA QUALIFIERS FROM OUTPUT** checkbox. This resulted in the generated script containing Stored Procedure names prefixed by the schema name. ([Bug#43276](#))
- In the Table Editor, the keyboard shortcuts such as `Ctrl+X`, `Ctrl+V` and `Ctrl+C`, affected the main document, rather than being confined to within the Table Editor. For example, if `Ctrl+C` was used to copy a column, this erroneously resulted in a duplicate table being created in the main model. ([Bug#42626](#))
- If a collation was set as the schema default, this was not correctly handled when the model was synchronized with a live database.

If a collation was set as a schema default, and then the model forward engineered to a database, then the collation was correctly set. If however, the model was then synchronized with the live database, MySQL Workbench attempted to alter the collations to `utf8_general_ci`. ([Bug#38807](#))

A.1.12. Changes in MySQL Workbench 5.2.6 (21st October 2009 alpha)

This section documents all changes and bug fixes that have been applied since the release of 5.2.5.

Bugs fixed:

- If a foreign key relationship was altered to point to a column in a different table, the foreign key was updated correctly but the EER diagram was not redrawn to reflect the new relationship. ([Bug#47807](#))
- After using the Forward Engineer to Database wizard, MySQL Workbench would crash if an attempt was subsequently made to exit the application. ([Bug#47276](#))
- When editing a trigger, the trigger editor would automatically insert a delimiter during a pause in typing, and also relocate the

cursor. ([Bug#45929](#))

A.1.13. Changes in MySQL Workbench 5.2.5 (Internal Release Only alpha)

This section documents all changes and bug fixes that have been applied since the release of 5.2.4.

Bugs fixed:

- The following exception was generated when closing the **RELATIONSHIP** tab for an EER Diagram:

```
System.NullReferenceException: Referência de objeto não definida para uma instância de
um objeto.
  em MySQL.GUI.Workbench.MainForm.mainDockPanel_ActiveDocumentClosing(Object sender,
ActiveDocumentClosingEventArgs e)
  em
WeifenLuo.WinFormsUI.Docking.DockPanel.OnActiveDocumentClosing(ActiveDocumentClosingEventArgs e)
  em WeifenLuo.WinFormsUI.Docking.DockPane.CloseContent(IDockContent content)
  em WeifenLuo.WinFormsUI.Docking.DockPane.CloseActiveContent()
  em WeifenLuo.WinFormsUI.Docking.VS2005DockPaneStrip.Close_Click(Object sender,
EventArgs e)
  em System.Windows.Forms.Control.OnClick(EventArgs e)
  em System.Windows.Forms.Control.WmMouseUp(Message& m, MouseButtons button, Int32
clicks)
  em System.Windows.Forms.Control.WndProc(Message& m)
  em System.Windows.Forms.Control.ControlNativeWindow.OnMessage(Message& m)
  em System.Windows.Forms.Control.ControlNativeWindow.WndProc(Message& m)
  em System.Windows.Forms.NativeWindow.Callback(IntPtr hWnd, Int32 msg, IntPtr wparam,
IntPtr lparam)
```

([Bug#47958](#))

- When attempting to run MySQL Workbench, the splash screen appeared and then the application immediately crashed, generating the “Send error report to Microsoft” dialog. On debugging it was found that MySQL Workbench was throwing a [Type-LoadException](#):

```
Could not load type 'WeifenLuo.WinFormsUI.Docking.ActiveDocumentClosingEventArgs' from
assembly 'WeifenLuo.WinFormsUI.Docking, Version=2.2.3428.40956, Culture=neutral,
PublicKeyToken=null'.
```

On inspection of the referenced assembly it was confirmed that there was no type [ActiveDocumentClosingEventArgs](#) defined. ([Bug#47534](#), [Bug#47480](#))

A.1.14. Changes in MySQL Workbench 5.2.4 (7th October 2009 alpha)

This section documents all changes and bug fixes that have been applied since the release of 5.2.3.

Functionality added or changed:

- Some keyboard shortcuts have been changed.

On Microsoft Windows and Linux **Ctrl+Y** is now used for Redo. Mac OS X continues to use **Cmd+Shift+Z** as its default for Redo. For Synchronize Model, **Ctrl+Alt+Y** is used on Windows and Linux, and **Cmd+Ctrl+Y** is used on Mac OS X. ([Bug#46285](#))

Bugs fixed:

- When editing a stored routine or trigger in the Table Editor, the cursor would relocate to the top of the editing window if a key was not pressed for a few seconds.

Note

On Windows this issue was fixed in 5.1.18. On Mac OS X this issue was fixed in 5.2.4. On Linux this issue was fixed in 5.2.7.

([Bug#48156](#))

- If a new empty EER diagram was created, then MODEL, RELATIONSHIP NOTATION, CONNECT TO COLUMNS was selected, MySQL Workbench generated the following messages:

```
** Message: item_activated: 0x28d5cf0 ->
'plugin:wb.view.setRelationshipNotation:fromcolumn'
** Message: unhandled message 4: wb.view.setRelationshipNotation finished in 0.00s
```

These messages were not generated if the EER diagram contained at least one table. ([Bug#47565](#))

- When a model was synchronized, the following incorrect SQL was generated:

```
CREATE TABLE IF NOT EXISTS `synthescom`.`adx_clienti` (
  `id` INT(11) UNSIGNED NOT NULL AUTO_INCREMENT ,
  `citta` VARCHAR(100) NULL DEFAULT NULL ,
  `provincia` VARCHAR(100) NULL DEFAULT NULL ,
  PRIMARY KEY (`id`)
)
ENGINE = MyISAM
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_general_ci;
```

Note that the closing parenthesis, which should have been located after the PRIMARY KEY statement, is missing. This resulted in the following error being generated:

```
ERROR: Error 1064: You have an error in your SQL syntax; check the manual that corresponds
to your MySQL server version for the right syntax to use near 'ENGINE = MyISAM
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_general_ci' at line 13
```

([Bug#47407](#))

- When a table was opened for editing in the Query Editor, the DECIMAL column values appeared in the result set as integer values, disregarding precision and scale settings. The fractional part of the number was discarded. It was also not possible to add the fractional part of the number during editing. ([Bug#47405](#))
- In the EER Diagram, Layer names were not rendered. This only happened with OpenGL rendering (only on Microsoft Windows). ([Bug#47385](#))
- When the QUERY DATABASE menu option was selected, the explorer on the right-hand side displayed Tables, Views and Routines, but when the Routines folder was expanded the routines contained in the model were missing. ([Bug#47088](#))
- Clicking the TEST CONNECTION button in DB Connection Editor wizard did not appear to have any effect. ([Bug#47083](#))
- In the Query Editor, when an attempt was made to expand a schema and select a table, the following exception was generated:

```
***** Exception Text *****
System.AccessViolationException: Attempted to read or write protected memory. This is
often an indication that other memory is corrupt.
  at MySQL.Grt.TreeModel.expand_node(NodeId node)
  at MySQL.Grt.TreeModel.TreeViewExpanding(Object sender, TreeViewAdvEventArgs e)
  at System.EventHandler`1.Invoke(Object sender, TEventArgs e)
  at Aga.Controls.Tree.TreeViewAdv.OnExpanding(TreeNodeAdv node)
  at Aga.Controls.Tree.TreeViewAdv.SetIsExpanded(TreeNodeAdv node, Boolean value)
  at Aga.Controls.Tree.TreeViewAdv.SetIsExpanded(ExpandArgs eargs)
  at Aga.Controls.Tree.TreeViewAdv.SetIsExpanded(TreeNodeAdv node, Boolean value, Boolean
ignoreChildren)
...
```

([Bug#47044](#))

- On Mac OS X using **cmd + backspace** to delete a column in the table editor did not work. ([Bug#46613](#))
- MySQL Workbench EER Diagram view did not allow the creation of a foreign key constraint on a primary key. ([Bug#39546](#))

A.1.15. Changes in MySQL Workbench 5.2.3 (15th September 2009 alpha)

This section documents all changes and bug fixes that have been applied since the release of 5.2.2.

Functionality added or changed:

- In the Inserts tab of the Table Editor, if a value was entered with quoting then any characters in the string requiring escaping were not escaped. However, if the same string was entered without quoting then the string would be escaped correctly by MySQL Workbench.

For 5.0 and 5.1 this is expected behavior. If a value is entered without leading and trailing quotes, the Inserts Editor adds quoting and escapes characters that require it. However, if quoted text is entered, the Inserts Editor carries out no further checks since it assumes a correctly escaped and quoted sequence has been entered.

5.2 features a new Inserts Editor. In this case the user enters the string without quoting or escaping and the Inserts Editor takes care of all quoting and escaping as required. ([Bug#38906](#))

Bugs fixed:

- In the Forward Engineer SQL CREATE script wizard, in the dialog asking for confirmation to overwrite an existing file, clicking the CANCEL button led to the wizard proceeding to the next step, while clicking the REPLACE button led to cancellation of the action. The functionality of the buttons appeared to be reversed. ([Bug#47257](#))
- When adding inserts with a `NULL` value in a column, MySQL Workbench incorrectly added single quotes to the NULL value.

Note

Note, when entering a `NULL` value right-click the button next to the value and select **SET SELECTION TO NULL**, otherwise `NULL` will be interpreted as a string literal.

([Bug#47122](#))

- MySQL Workbench crashed on launch on Mac OS X 10.6.0. ([Bug#46953](#))
- If the Default Storage Engine was selected as MyISAM in the Model Options dialog, then when a new table was created in the EER Diagram it was found to have a storage engine type of InnoDB. ([Bug#46752](#))
- When a model containing invalid/broken foreign key definitions was loaded into MySQL Workbench, the errors were detected and repaired.

However, when the model was saved and reloaded it contained changed data types. All `INTEGER` columns were changed to `INTEGER(11)`. Further, some `BOOLEAN` columns were changed to `BOOLEAN(1)`. ([Bug#46467](#))

- If the Table Editor had been invoked then the Text Boxes in the TOOLS, PREFERENCES dialog became vertically misaligned. This only happened for MySQL Workbench running on Mac OS X. ([Bug#46255](#))
- When a connection was selected in the DB Connection Editor it appeared to flicker a number of times before MySQL Workbench crashed. ([Bug#46065](#))
- When a database with Stored Procedures was reverse-engineered into MySQL Workbench the Stored Procedures were not displayed in the Routine Editor, and so could not be edited. ([Bug#45704](#))
- When running MySQL Workbench on Mac OS X, if a table was added to the model diagram and deleted the application crashed. This happened for a new diagram from a schema imported from a live database. ([Bug#45692](#))
- During Forward Engineering, clicking the COPY TO CLIPBOARD button generated code that contained an extra newline per line. ([Bug#45579](#))
- Attempts to rebuild the MySQL Workbench source RPM failed. The RPM appeared to contain an unmodified template `mysql-workbench.spec.in` instead of the correct spec file with variable placeholders replaced by actual values.

When using the following command to rebuild the RPM:

```
shell> rpmbuild -ba --clean SPECS/mysql-workbench.spec.in
```

The following error was generated:

```
error: File /usr/src/packages/SOURCES/mysql-workbench-oss-@VERSION@.tar.gz:  
No such file or directory
```

Further, the spec file in the package had the incorrect suffix “.spec.in” instead of “.spec”. ([Bug#45515](#))

- In MySQL Workbench on Mac OS X, it was not possible to resize a Text Object on the EER Diagram canvas.
See also [Bug#39887](#). ([Bug#45472](#))
- It was not possible to assign columns to indexes, as the **COLUMN** checkboxes did not respond to mouse events. ([Bug#45260](#))
- MySQL Workbench generated a segmentation fault when clicking on either the **NN** or **AI** checkboxes on the **COLUMNS** tab of the Table Editor. ([Bug#45075](#))
- In the **COLUMNS** tab of the Table Editor, if the column was of type **ENUM** and was given a default value, then when an attempt was made to forward engineer the schema it would not validate. ([Bug#44368](#))
- In the Physical Schemata pane if more schema were added than could fit within the pane, then these schema could not be viewed, as there was no ability to scroll the pane. Further, double clicking the schema in the Catalog pane did not locate the schema in the Physical Schemata pane.

MySQL Workbench was changed so that a small down arrow button was added to the right side of the Physical Schemata tab bar. This provides a dropdown to select a specific schema tab. ([Bug#39735](#))

- When HELP, ABOUT was selected from the main menu, it was not possible to copy MySQL Workbench version information to the clipboard. ([Bug#39610](#))
- When a diagram was exported as PNG it used the height and width of the model as displayed in the EER Diagram. This resulted in the exported picture being too small or too large, rather than being scaled to a consistent size. ([Bug#36226](#))

A.1.16. Changes in MySQL Workbench 5.2.2 (27th July 2009 alpha)

This section documents all changes and bug fixes that have been applied since the release of 5.2.1.

Bugs fixed:

- The Synchronize Model with Database wizard contain a spelling mistake. The word “synchronization” was misspelled as “synchronizatiob”. ([Bug#45939](#))
- MySQL Workbench 5.1.16 failed to create INSERTs in the exported DDL for some tables. ([Bug#45920](#))
- MySQL Workbench crashed on startup on Mac OS X. ([Bug#45869](#))
- If MySQL Workbench gave the message that the MySQL Server was no longer available (because of timeout due to inactivity) then MySQL Workbench crashed when the user tried to reconnect. ([Bug#45123](#))
- The UML relationship notation was incorrect.

When setting a Foreign Key to NULLable, the table referenced by the Foreign Key should be marked as optional. However, MySQL Workbench marked the table containing the Foreign Key itself as optional. ([Bug#45069](#))

- If a user added a **LIMIT** clause to a query, the automatically added **LIMIT 0,1000** clause caused a syntax error. ([Bug#45051](#))
- The new SQL IDE, introduced in 5.2, did not support returning results from procedures. When any routine was called, there were no results displayed in the query editor window. However, the log contained the message “Response: OK”.

When another query was run on the same connection, the log message was:

```
Error Code: 2014 Commands out of sync; you can't run this command now
```

([Bug#44910](#))

A.1.17. Changes in MySQL Workbench 5.2.1 (22th May 2009 alpha)

This section documents all changes and bug fixes that have been applied since the release of 5.2.0.

A.1.18. Changes in MySQL Workbench 5.2.0 (30th April 2009 alpha)

This is the first alpha release of 5.2.0.

A.2. Changes in Release 5.1

A.2.1. Changes in MySQL Workbench 5.1.19 (Not yet released GA)

This section documents all changes and bug fixes that have been applied since the release of 5.1.18.

Functionality added or changed:

- In the **FORWARD ENGINEER SQL SCRIPT** wizard the checkbox **GENERATE DROP SCHEMA** now causes every `CREATE SCHEMA` statement to be prepended with `DROP SCHEMA IF EXISTS `schemaname``. ([Bug#46706](#))

Bugs fixed:

- The **OMIT SCHEMA QUALIFIER IN OBJECT NAMES** option in the **FORWARD ENGINEER SQL SCRIPT** wizard appeared to have no effect. After selecting this option and proceeding through the wizard, the generated script still contained schema qualifiers. ([Bug#46268](#))
- Synchronizing a model with a live database, without having made any changes to the model or the database, caused the model to appear as unsaved, indicating that unnecessary changes may have been made to the model. ([Bug#40914](#))
- When synchronizing with a live database, if the direction of synchronization was changed to inbound, then the script generated indicated that no inbound changes would be made. ([Bug#40648](#))
- MySQL Workbench EER Diagram view did not allow the creation of a foreign key constraint on a primary key. ([Bug#39546](#))

A.2.2. Changes in MySQL Workbench 5.1.18 (3rd September 2009 GA)

This section documents all changes and bug fixes that have been applied since the release of 5.1.17. This is the first GA level release of 5.1.

Functionality added or changed:

- The Column Editor has been updated to remove the feature whereby a column could be toggled to a primary key by double clicking it. Primary key status can now only be set by selecting the appropriate checkbox in the adjacent **COLUMN DETAILS** frame. ([Bug#46579](#))
- It was not possible to change column order by dragging and dropping columns in the **COLUMNS** tab of the Table Editor. MySQL Workbench has been changed to allow you to drag and drop a column to change the column order as required. ([Bug#40601](#))

Bugs fixed:

- When editing a stored routine or trigger in the Table Editor, the cursor would relocate to the top of the editing window if a key was not pressed for a few seconds.

Note

On Windows this issue was fixed in 5.1.18. On Mac OS X this issue was fixed in 5.2.4. On Linux this issue was fixed in 5.2.7.

([Bug#48156](#))

- MySQL Workbench generated an unhandled exception when using **FILE**, **EXPORT**, **FORWARD ENGINEER SQL CREATE SCRIPT**. The exception was generated if an attempt was made to generate a script so that it wrote over a script already created. ([Bug#47115](#))
- Clicking the **TEST CONNECTION** button in DB Connection Editor wizard did not appear to have any effect. ([Bug#47083](#))
- MySQL Workbench crashed on launch on Mac OS X 10.6.0. ([Bug#46953](#))
- When the datatype of a table column was changed in the Table Editor, it was not reflected in the EER Diagram, although the tooltip was updated correctly. This only happened in MySQL Workbench 5.1.17. ([Bug#46940](#))
- In the **COLUMNS** tab of the Table Editor, when the empty row at the end of the column list was right clicked, MySQL Work-

bench crashed with the following error:

```
** (mysql-workbench-bin:15234): WARNING **: /tmp/sakila.mwbd1/document.mwb.xml:26: link
'{591FC376-B82F-4F3D-B185-BA5C65B77080}' <object workbench.Workbench> key=owner could not
be resolved
** Message: unhandled message 4: wb.file.openRecentModel finished in 0.58s

glibmm-ERROR **:
unhandled exception (type std::exception) in signal handler:
what: Index out of range.

aborting...
Aborted
```

([Bug#46937](#))

- The DDL was generated without foreign keys, regardless of whether the **SKIP GENERATION OF FOREIGN KEYS** option was turned on. ([Bug#46875](#))
- In the Column tab of the Table Editor, if the **NN** checkbox was selected or deselected, the change was not immediately reflected in the EER Diagram, but was seen if some other operation caused the table object in the diagram to be refreshed. This also happened with the **AI** checkbox. ([Bug#46869](#))
- The auto-completion of column names in the Table Editor completed names unnecessarily, requiring characters in the name to be manually deleted. This only happened on the Linux version of MySQL Workbench. ([Bug#46847](#))
- The **Esc** key did not initiate the closing of an active dialog box. ([Bug#46829](#))
- Editing of Text Objects did not work correctly. When **Enter** was hit, line feed did not move editing to the next line, and introduced a box character. ([Bug#46789](#))
- The Forward Engineer SQL Script wizard generated invalid SQL code for a small model with foreign keys:

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL';

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

([Bug#46787](#))

- MySQL Workbench did not export Foreign Keys when using the **FILE**, **EXPORT**, **FORWARD ENGINEER SQL CREATE SCRIPT** wizard, even though the option **SKIP CREATION OF FOREIGN KEYS** checkbox was cleared. Further, MySQL Workbench crashed when a column with a Foreign Key was added to a table. ([Bug#46783](#))
- If the Default Storage Engine was selected as MyISAM in the Model Options dialog, then when a new table was created in the EER Diagram it was found to have a storage engine type of InnoDB. ([Bug#46752](#))
- When synchronizing with a live database, the script generated included drop statements that were placed in the wrong order, this led to the a schema being dropped after it was created. ([Bug#46740](#))
- When a N:M identifying relationship was created on a single table, a foreign key name collision occurred - both keys were given the same name. MySQL Workbench has been changed so that key names have a trailing number added to avoid conflicts. ([Bug#46363](#))
- MySQL Workbench generated an exception when the mouse wheel was used to move between tabs in the Table Editor:

```
(mysql-workbench-bin:4864): Gtk-CRITICAL **: gtk_tree_view_unref_tree_helper: assertion
`node != NULL' failed

glibmm-ERROR **:
unhandled exception (type std::exception) in signal handler:
what: invalid index

aborting...
Aborted
```

([Bug#46304](#))

- When an EER Diagram contained tables or views with underscores in their names, the context menu associated with them displayed their names incorrectly in the menu. The underscores where displayed as accelerator keys in the context menu. ([Bug#46302](#))
- The **USER DEFINED TYPES** dialog, launched from **MODEL**, **USER DEFINED TYPES...** on the main menu, did not allow user

defined types to be changed. Further, it appeared to be possible to edit the new user type in the **USER TYPES** side panel, but no changes made there were retained.

MySQL Workbench has been changed to allow User Types to be edited only from the **USER DEFINED TYPES** dialog, and correct operation of this dialog has been restored. ([Bug#45936](#))

- When editing a trigger, the trigger editor would automatically insert a delimiter during a pause in typing, and also relocate the cursor. ([Bug#45929](#))
- The DBDoc documentation generation system did not support facilities for Stored Routines, Views and Triggers. ([Bug#41589](#))
- After using the [HELP](#), [UPDATE...](#) to upgrade MySQL Workbench, the application failed to start correctly. After the splash screen was displayed MySQL Workbench displayed an error dialog - “MySQL Workbench has stopped working”. ([Bug#41460](#))
- In the Physical Schemata pane if more schema were added than could fit within the pane, then these schema could not be viewed, as there was no ability to scroll the pane. Further, double clicking the schema in the Catalog pane did not locate the schema in the Physical Schemata pane.

MySQL Workbench was changed so that a small down arrow button was added to the right side of the Physical Schemata tab bar. This provides a dropdown to select a specific schema tab. ([Bug#39735](#))

- When a diagram was exported as PNG it used the height and width of the model as displayed in the EER Diagram. This resulted in the exported picture being too small or too large, rather than being scaled to a consistent size. ([Bug#36226](#))
- Printing of an EER Diagram did not work correctly if a table vertically spanned multiple pages. In this case pages would be printed up to and including the page that contained the first part of the table that spanned multiple pages, but the pages containing the remaining parts of the table would not be printed. This problem typically occurred when a table had more fields than could comfortably fit on a single page. ([Bug#33919](#))

A.2.3. Changes in MySQL Workbench 5.1.17 (14th August 2009 beta)

This section documents all changes and bug fixes that have been applied since the release of 5.1.16.

Functionality added or changed:

- Bug reported: If code with an error was entered into the **TRIGGERS** tab of the Table Editor, the code disappeared when typing stopped.

What actually happened was when a table name was given that was not that of the table currently being edited, the trigger code was moved to the table specified.

MySQL Workbench has now been changed so that if a table name is specified other than other than that of the table being edited, then the table name is highlighted as a syntax error, rather than moving the trigger code to the table actually specified. ([Bug#46349](#))

- When a model with multiple EER Diagrams was opened, all EER Diagrams would be displayed in tabs. This happened whether or not a EER Diagram had been displayed in a tab prior to saving the model.

MySQL Workbench has been changed so that the EER Diagram tab state is saved, so that when a model is opened, only those EER Diagram tabs that were open on save are restored. This prevents unnecessary cluttering of the interface. ([Bug#44454](#), [Bug#50732](#))

- When a Text Object was placed on an EER Diagram it could not be resized and also did not appear to automatically resize correctly around any text typed in.

Text Objects on the EER Diagram canvas can now be resized by grabbing the sizing handles that appear when the object is selected. As with Table Objects, a manual modification of the object sets the [manualSizing](#) attribute of the Text Object to `true`. This property can also be set in the Properties pane. This means that these Text Objects automatically expand on entering more text, but the size isn't automatically reduced when text is removed or wrapped manually. By setting this property to `false`, the Text Object size is also automatically reduced to fit the containing text. ([Bug#39887](#))

- In the **FOREIGN KEYS** tab it was not possible to rename a foreign key by renaming the corresponding index entry.

MySQL Workbench has been changed so that it is possible to rename a foreign key by renaming its corresponding index. The foreign key name is now automatically updated when the index is renamed. The index name is also updated when the foreign key name is changed. ([Bug#39511](#))

- In MySQL Workbench it was not possible to generate an export script without schema information. MySQL Workbench has

been changed to include the option **OMIT SCHEMA QUALIFIER IN OBJECT NAMES**. This can be found in the Forward Engineer SQL Script and Forward Engineer to Database wizards.

Note that this does not change the script used in Procedures or Views. If you are using schema qualifiers in these locations you will have to remove them manually. ([Bug#34827](#))

- The menu item HELP, VIEW REPORTED BUGS has been added to the main menu. This opens the bugs listing for MySQL Workbench in the default browser. ([Bug#32813](#))

Bugs fixed:

- When trying to connect to a server from MySQL Workbench the following error was generated:

```
Connection 'Server' could not be established: No bool value passed for CLIENT_COMPRESS
```

This occurred when using connections created in MySQL Workbench 5.2.1 with 5.2.2. ([Bug#46635](#))

- On the EER Diagram, if an Undo operation was carried out after an Autosize operation, then table objects were reduced to their minimum size. ([Bug#46605](#))
- An Unknown Exception was caught after placing a relationship between two tables in the EER Diagram. The exception generated was:

```
Unknown Exception caught in: c:\documents and settings\mysqldev\my documents\visual studio 2008\projects\workbench\backend\windows\wbprivate.wr\src\Wb.h at line 1026
```

([Bug#46562](#))

- In the Columns tab of the Table Editor, if a default value was added for a column, it could not then be removed. ([Bug#46509](#))
- Calling the PLUGINS, CATALOG, DUMP ALL TABLE COLUMNS item from the main menu generated the following error:

```
Error executing plugin wb.catalog.util.dumpColumns: error calling PyWbUtils.printAllColumns: 'str' object is not callable
```

([Bug#46477](#))

- In the EER Diagram, if a layer was double-clicked a new tab was added to the Table Editor. However, if a relationship or table was double-clicked any existing tab would be reused, rather than a new tab being created. There appeared to be no consistent policy on how double-clicks should be handled. ([Bug#46466](#))
- The **INSERT** statements generated for columns of type **TEXT** by the Insert Editor were not correct, as the values were not quoted. Also, when the Insert Editor was closed and then reopened the last column entry was lost. ([Bug#46390](#))
- When an attempt was made to synchronize Stored Procedures from the source database to MySQL Workbench, the Stored Procedures were deleted from the source database. ([Bug#46346](#))
- If an attempt was made to load a new model file while a model file was already loaded then the following error dialog was generated:

```
OverviewBE::get_node: invalid node 1.1
```

([Bug#46292](#))

- When PLUGINS, CATALOG, GIVE A PREFIX TO ALL TABLES IN CATALOG, was selected from the main menu, MySQL Workbench crashed with the following error:

```
** (mysql-workbench-bin:5898): WARNING **:  
/home/miguel/.mysql/workbench/wb_options.xml:325: link  
'b7ee49b4-67f5-11de-9d1e-0800272fd858' <object GrtObject> key-owner could not be resolved  
** Message: item_activated: 0xadbb608 -> 'plugin:wb.file.newDocument'  
** Message: unhandled message 4: wb.file.newDocument finished in 0.34s  
** Message: item_activated: 0xaacbea8 -> 'plugin:wb.util.prefixTables'  
glibmm-ERROR **:  
unhandled exception (type std::exception) in signal handler:  
what: request_input_becb not implemented  
aborting...  
Aborted
```

This only happened when running on Linux. ([Bug#46280](#))

- If the Table Editor had been invoked then the Text Boxes in the **TOOLS, PREFERENCES** dialog became vertically misaligned. This only happened for MySQL Workbench running on Mac OS X. ([Bug#46255](#))
- When a SQL Script was added and then its name changed, the name would erroneously revert back to “Script”. ([Bug#46246](#))
- Reverse Engineering from a live database failed if the server had **SQL_MODE** set to **ANSI_QUOTES**. ([Bug#46185](#))
- In the **COLUMNS** tab of the Table Editor, selecting the **AI** checkbox before entering a name for the column caused MySQL Workbench to crash. ([Bug#46150](#))
- The Microsoft Windows version of MySQL Workbench leaked GDI objects, resulting in a drop in performance. ([Bug#46101](#))
- The text contained in a Text Object in an EER Diagram was syntax highlighted. This was not appropriate for a simple text note. ([Bug#46092](#))
- When using the Reverse Engineer Database wizard a Segmentation Fault was generated by MySQL Workbench on Ubuntu Linux:

```
(mysql-workbench-bin:22735): Gtk-CRITICAL **: gtk_tree_selection_get_selected: assertion `selection->type != GTK_SELECTION_MULTIPLE' failed
** Message: item_activated: 0x30d4920 -> 'plugin:db.plugin.database.rev_eng'
** Message: unhandled message 4: db.plugin.database.rev_eng finished in 0,00s
```

([Bug#46078](#))

- When writing code in the Routine Editor it appeared that the code was lost if the Routine Editor was closed. ([Bug#46049](#))
- MySQL Workbench crashed if a table was selected, right-clicked, and then the menu item **COPY SQL TO CLIPBOARD** chosen. MySQL Workbench also crashed with the same model if an attempt was made to export the schema using the Forward Engineer SQL CREATE Script wizard. ([Bug#46025](#))
- In the EER Diagram, when hovering the cursor over a column in a table, the hint box contained the text associated with the subsequent column, and the first column hint box contained the name of the table. ([Bug#45997](#))
- The Synchronize Model with Database wizard contain a spelling mistake. The word “synchronization” was misspelled as “synchronizatiob”. ([Bug#45939](#))
- If an attempt was made to copy and paste a trigger, the trigger code would revert to the source trigger, when the pasted trigger was edited.

MySQL Workbench has been changed so that triggers are associated with the table in which they are edited. If a trigger has the wrong schema or table pointed to by its **CREATE TRIGGER** statement, it will be highlighted as a syntax error. ([Bug#45931](#))

- MySQL Workbench 5.1.16 failed to create INSERTs in the exported DDL for some tables. ([Bug#45920](#))
- MySQL Workbench crashed on startup on Mac OS X. ([Bug#45869](#))
- When a model was synchronized with a live database, and only stored procedures needed to be synchronized, then the following error was generated:

```
ERROR: Error 1046: No database selected
```

([Bug#45867](#))

- In MySQL Workbench running on Mac OS X, if any changes were made in the **PAGE SETUP** dialog then the following error was generated:

```
builtin:wb.page_setup
Type mismatch: expected object of type app.PaperType, but got app.PaperType
```

Note, the **PAGE SETUP** dialog is accessed by selecting **FILE, PAGE SETUP...** from the main menu. ([Bug#45861](#))

- The Forward Engineering SQL ALTER Script wizard generated **DROP INDEX** and **ADD UNIQUE INDEX** statements for any unique index, even for one that had not been modified. These unnecessary statements had to be manually deleted from the script. ([Bug#45830](#))

- When compiling the MySQL Workbench source code with GNU C Compiler (GCC) version 4.4.0, the compilation failed, as the version of the Boost library used was not compatible with GCC 4.4.0.

The version of the Boost library required for compatibility with GCC 4.4.0 is 1.37 or later. ([Bug#45798](#))

- MySQL Workbench crashed and also lost procedure objects when attempting to synchronize with a live database. ([Bug#45773](#))
- When building MySQL Workbench on PPC/PPC64 on Linux, the build failed because “bswap32” was not defined.

```
In file included from src/template_string.cc:47:
src/base/arena.h: In member function 'void google::BaseArena::ReturnMemory(void*, size_t)':
src/base/arena.h:211: warning: comparison between signed and unsigned integer expressions
src/template_string.cc: In function 'uint32 UNALIGNED_LOAD32(const void*)':
src/template_string.cc:120: error: 'bswap32' was not declared in this scope
```

([Bug#45629](#))

- Relationship lines were not always drawn on the EER Diagram. ([Bug#45583](#))
- During Forward Engineering, clicking the COPY TO CLIPBOARD button generated code that contained an extra newline per line. ([Bug#45579](#))
- Attempts to rebuild the MySQL Workbench source RPM failed. The RPM appeared to contain an unmodified template `mysql-workbench.spec.in` instead of the correct spec file with variable placeholders replaced by actual values.

When using the following command to rebuild the RPM:

```
shell> rpmbuild -ba --clean SPECS/mysql-workbench.spec.in
```

The following error was generated:

```
error: File /usr/src/packages/SOURCES/mysql-workbench-oss-@VERSION@.tar.gz:
No such file or directory
```

Further, the spec file in the package had the incorrect suffix “.spec.in” instead of “.spec”. ([Bug#45515](#))

- MySQL Workbench 5.1.13 running on Ubuntu Linux crashed when it attempted to load a model file created using MySQL Workbench 5.0 on Windows XP. The same model file did load correctly using MySQL Workbench 5.1.12. ([Bug#45491](#))
 - In MySQL Workbench on Mac OS X, it was not possible to resize a Text Object on the EER Diagram canvas.
- See also [Bug#39887](#). ([Bug#45472](#))
- When synchronizing a model with a live database, clicking the UPDATE MODEL actually caused the server to be updated as if UPDATE SOURCE had been clicked. ([Bug#45456](#))
 - The settings for `ON UPDATE` and `ON DELETE` in Foreign Key/Relationship Defaults set in the **MODEL** tab of **TOOLS, PREFERENCES**, were not taken into account when new tables and relationships were created. ([Bug#45393](#), [Bug#45239](#))
 - The Inserts Editor did not have functionality to remove a row once added. This problem only occurred with MySQL Workbench running on Linux. ([Bug#44458](#))
 - In the **COLUMNS** tab of the Table Editor, if the column was of type `ENUM` and was given a default value, then when an attempt was made to forward engineer the schema it would not validate. ([Bug#44368](#))
 - In the Forward Engineer SQL Script wizard, on the **SQL OBJECT EXPORT FILTER** page, if BACK was clicked to go to the **SQL EXPORT OPTIONS** page, and then NEXT was clicked to return to the **SQL OBJECT EXPORT FILTER** page, then on that page all sections were duplicated. If this was repeated the objects were triplicated and so on. ([Bug#44317](#))
 - In the **COLUMNS** tab of the Table Editor, when setting a column datatype to `ENUM` it was not possible to choose Collation for the column details. Only the Table Default collation was available. ([Bug#43352](#))
 - In the **COLUMNS** tab of the Table Editor, a column name could be entered with leading or trailing spaces. This led to the following error when exporting the schema:

```
ERROR 1166 (42000): Incorrect column name 'name '
```

([Bug#43345](#))

- The script created by Forward Engineer SQL CREATE Script failed with an error if the model contained a view that referenced

a column defined in another view. ([Bug#43061](#))

- In the **COLUMNS** tab of the Table Editor, if a column is dragged and dropped to change its position, then after the move the column highlighted is not the one moved. ([Bug#42476](#))
- In the **COLUMNS** tab of the Table Editor, if a column was deleted, and it had inserts in the **INSERTS** tab, then in the **INSERTS** tab the last entry would be incorrectly deleted. ([Bug#41931](#))
- In the **FOREIGN KEYS** tab of the Table Editor, the values of the **ON UPDATE** and **ON DELETE** fields, in the **FOREIGN KEY OPTIONS** pane, did not always refresh for each Foreign Key constraint selected in the table on the left side.

For example, if the currently selected constraint was `SET NULL` for **ON DELETE**, and then a different selection was made for a constraint that has no action specified for **ON DELETE**, the field remained with the value from the previous selection, which gave a false indication that such action was specified in the definition of the newly selected constraint.

Note that incorrect field values were only displayed when the last selected constraint did not have an action defined. ([Bug#41887](#))

- When the flags of a User Defined Type were changed in the **USER TYPES** panel, the change was not updated in the EER Diagram, the Table Editor, or the exported SQL. ([Bug#41453](#))
- A model failed to synchronize with a live database, as a column of type `DOUBLE` was exported as a type `DOUBLE (256)`. ([Bug#41290](#))
- When a MySQL Workbench model was synchronized with a live database, a foreign key relationship, recently added to the database, was not reflected in the EER Diagram. Although the foreign key was created in the table object, the relationship connector widget was not drawn on the canvas. ([Bug#41219](#))
- In the **COLUMNS** tab of the Table Editor, when moving a column lower in the list of columns using drag and drop, the column was incorrectly placed one position below the point indicated by the position marker. ([Bug#40641](#))
- When CONNECT TO COLUMNS was selected for the Relationship Notation option, the line connecting a foreign key to its referenced table was drawn to the incorrect column. ([Bug#40627](#))
- When HELP, ABOUT was selected from the main menu, it was not possible to copy MySQL Workbench version information to the clipboard. ([Bug#39610](#))
- Foreign Key checks were not enabled before the standard inserts section of the script started. ([Bug#35180](#))

A.2.4. Changes in MySQL Workbench 5.1.16 (30th June 2009 beta)

This section documents all changes and bug fixes that have been applied since the release of 5.1.15. This release is the first GA build of 5.1.

Functionality added or changed:

- The MySQL Workbench TOOLS, OPTIONS... menu item has been renamed to PREFERENCES..., and the corresponding dialog now has the title **WORKBENCH PREFERENCES**. ([Bug#44462](#))
- When a relationship was edited in the EER diagram, a new tab was created in the Table Editor, rather than using any existing relationship tab. For example, if one relationship tab existed, and then another relationship was edited, a new tab would be created, rather than using the existing tab. ([Bug#39624](#))
- In an EER diagram, if a table contained enums with many values, the table was excessively wide, and had to be manually resized.

Version 5.1.16 now features a new preference setting which lets you specify the maximum length of a data type definition string, to be displayed in a table in an EER diagram. This is available for `ENUM` and `SET` types. Data type definitions that exceed these lengths are displayed as `SET/ENUM(. . .)` in the EER diagram.

The new setting can be found by selecting TOOLS, PREFERENCES from the main menu, and then selecting the **DIAGRAM** tab. The settings are then available in the **TABLES** panel. ([Bug#34919](#))

Bugs fixed:

- When an attempt was made to add a new table to a schema page other than the default **MySQL MODEL** page, then the following error was generated:

```
Unknown Exception caught in c:\documents and settings\mysqldev\my documents\visual studio  
2008\projects\workbench\backend\windows\wbprivate.wr\src\Wb.h at line 1026
```

([Bug#45821](#))

- When a model was exported using either **FILE**, **EXPORT**, **FORWARD ENGINEER SQL CREATE SCRIPT** or **DATABASE**, **FORWARD ENGINEER...**, MySQL Workbench crashed with a segmentation fault. This occurred after the objects to export were selected, it did not matter if one or all objects were selected. This occurred using MySQL Workbench 5.1.14 RC on Ubuntu Linux and Windows. ([Bug#45718](#))
- The data type **ENUM** did not accept parenthesis in its values. ([Bug#45607](#))
- In the **COLUMNS** tab of the Table Editor, when a column was right-clicked to display the context-sensitive menu, the **MOVE UP** and **MOVE DOWN** menu items were disabled. This meant that the order of the columns could not be changed in the normal manner. ([Bug#45590](#))
- Right-clicking in a blank area of the **USER TYPES** window generated an exception:

```
System.NullReferenceException: Object reference not set to an instance of an object.  
    at MySQL.GUI.Workbench.UserDatatypesForm.contextMenuStrip1_Opening(Object sender,  
CancelEventArgs e)  
    at System.Windows.Forms.ToolStripDropDown.OnOpening(CancelEventArgs e)  
...
```

([Bug#45490](#))

- When attempting to enter trigger code into the Triggers tab of the Table Editor, a timer appeared to periodically fire, which caused the delimiter to be added while typing.

This only happened when using the Mac OS X version of MySQL Workbench. ([Bug#44264](#))

- In the Forward Engineer SQL Script wizard, the location opened by the **BROWSE** button defaulted to the MySQL Workbench install directory. However, after an output script file was set the location opened by the **BROWSE** button returned to the default, rather than remembering the last location accessed. This required navigating to the correct location each time the script was re-generated. ([Bug#43837](#))
- MySQL Workbench displayed incorrect icons in the Table Editor. The icons for columns displayed in the table editor did not match those displayed in the EER Diagram. ([Bug#42794](#))
- In the Forward Engineer SQL CREATE Script wizard, the **GENERATE INSERT STATEMENTS FOR TABLES** option did not take into account which tables were excluded in the SQL Export Filters. All insert statements were generated, including those for tables that had been excluded in the export filters. ([Bug#40913](#))
- In the EER Diagram, right-clicking to edit a table sometimes failed with the following error:

```
Cannot execute db.mysql.plugin.edit.table  
Plugin requires unhandled argument type.
```

([Bug#39513](#))

- When a table was renamed, inserts that had been created for it disappeared. Further, when a table was updated in **PHYSICAL SCHEMATA** or in the **CATALOG** pane, inserts were not updated. ([Bug#38654](#))
- If font size was changed in the Appearance tab of the Workbench Preferences dialog, this had no immediate effect on the visual appearance of the EER Diagram. ([Bug#38198](#))
- In the **INDEXES** tab of the Table Editor, it was not possible to change the order number of a column for an index of type **PRIMARY** in the **INDEX COLUMNS** panel, using the drop down list box available in the # column. ([Bug#37273](#))
- When zooming an EER Diagram, the text in a table would sometimes extend beyond the edge of the table box. ([Bug#35407](#))

A.2.5. Changes in MySQL Workbench 5.1.15 (26th June 2009 beta)

This section documents all changes and bug fixes that have been applied since the release of 5.1.14. This release is RC3.

Functionality added or changed:

- A routine object was deleted if its code was removed from the routine group. This behavior has now been noted in the main

documentation. ([Bug#45738](#))

Bugs fixed:

- Selecting **DATABASE**, **GENERATE CATALOG DIFF REPORT** from the main menu caused MySQL Workbench to crash. ([Bug#45652](#))
- Generated INSERT statements did not have a database name.

The generated `INSERT` statements were at the end of the SQL output. However, there was no `USE database` statement before the group of inserts. The comment before the group of `INSERTS` did refer to the database table.

If you had multiple databases, this generated errors on import. ([Bug#45642](#))

- In the **FORWARD ENGINEER SQL SCRIPT** wizard, when the SHOW FILTER button was clicked, the left hand panel containing database and table names was too narrow, especially if a long database name or table name was used. This made it hard to determine which tables to select. ([Bug#45623](#))
- The **FILE**, **EXPORT**, **FORWARD ENGINEER SQL CREATE SCRIPT** wizard contained malfunctioning buttons. When prompted with the message “Do you want to overwrite an existing .sql file?”, the REPLACE and CANCEL buttons had the reverse effect. Clicking REPLACE cancelled the action, whereas clicking CANCEL proceeded to the next step. ([Bug#45619](#))
- In Ubuntu Linux right-clicking on any item in the **CATALOG**, **LAYERS**, **USER TYPES**, or **HISTORY** tabs, or other tabs typically docked in the right hand side of MySQL Workbench, did not display a context-sensitive menu. ([Bug#44298](#))
- MySQL Workbench **SYNCRONIZE MODEL** feature had unexpected behavior. For example, synchronizing the model with the live database would cause changes to the database, even if this was not desired. Also, if tables were dropped in the live database, and the model synchronized, the changes to the live database were not reflected in the model. Further, if a model was created and synchronized to an empty schema in the live database, when synchronization took place again, MySQL Workbench would indicate many changes were required, even if no changes had been made to the model or database. ([Bug#42149](#))
- Double-clicking a layer in the **LAYERS** tab did not select it in the EER Diagram canvas. However, other objects in the **LAYERS** tab could be selected on the EER Diagram canvas in this manner. ([Bug#34938](#))
- Right-clicking on a relationship in the **EER DIAGRAM** canvas, and selecting **EDIT IN NEW WINDOW...** from the context-sensitive menu, had no effect. ([Bug#34069](#))

A.2.6. Changes in MySQL Workbench 5.1.14 (19th June 2009 beta)

This section documents all changes and bug fixes that have been applied since the release of 5.1.13.

Bugs fixed:

- On Mac OS X, MySQL Workbench crashed when creating a new document after another one had been worked on.
If editing a document, and then selecting to create a new document and not saving the current document when prompted, MySQL Workbench crashed when the save dialog closed. ([Bug#45519](#))
- Printing an EER diagram crashed MySQL Workbench. ([Bug#45518](#))
- When right-clicking on an item that was not selected, such as a table or routine, in the **MySQL MODEL** tab, the item was selected without the other items being deselected.
The expected behavior for applications in both Windows and Linux is that right-clicking an item that has not been selected, should deselect all other items. Right-clicking an already selected item should not alter any of the selections. ([Bug#44268](#))
- When synchronizing a model with a database, rather than synchronizing only the selected tables, all tables were synchronized. ([Bug#43485](#))

A.2.7. Changes in MySQL Workbench 5.1.13 (12th June 2009 beta)

This section documents all changes and bug fixes that have been applied since the release of 5.1.12.

Bugs fixed:

- When using the Forward Engineer SQL Script wizard, the **SQL OBJECT EXPORT FILTER** step displayed duplicate items. ([Bug#45241](#))
- When using the Generate Catalog Diff Report wizard, the Stored Connection information was not automatically populated when a previously defined Stored Connection was selected. ([Bug#45234](#))
- Adding Insert records for a table worked correctly. However, there was no way to remove the records once the OPEN EDITOR facility has been used to add them. ([Bug#45233](#))
- On Mac OS X, using the Forward Engineer to Database, or Forward Engineer SQL Script wizards caused MySQL Workbench to generate the error message:

```
The application MySQLWorkbench quit unexpectedly after it was relaunched.
```

([Bug#45229](#))

- When defining a primary key with a user-defined data type, it was not possible to set the auto-increment (AI) attribute. Clicking the AI checkbox had no effect.

To circumvent this problem it was possible to temporarily change the data type of the column to a built-in data type such as `INT`, set the AI attribute and then change the data type back to the desired user-defined data type. The model could then be saved. However, if this model was reopened and FILE, EXPORT, FORWARD ENGINEER SQL CREATE SCRIPT used to generate a script, MySQL Workbench would crash. ([Bug#45165](#))

- MySQL Workbench did not limit foreign key name length. This meant that an exported SQL script was invalid and MySQL Server generated an error on attempting to import it:

```
1059 - ER_TOO_LONG_IDENT
```

([Bug#45139](#))

- On Mac OS X, if the mouse pointer was used to select an entity and then COMMAND+DELETE was pressed before the tooltip appeared, then MySQL Workbench crashed when the delete was confirmed. ([Bug#45042](#))
- Foreign key names must be unique. However, MySQL Workbench default generated foreign key names were sometimes identical, causing Forward Engineer SQL CREATE Script to fail. ([Bug#45027](#))
- Synchronization of the model with an external database failed to report errors. This meant there was the possibility of an incomplete synchronization, without the developer being informed. ([Bug#45024](#))
- Changing a relationship's visibility had no effect.

If a relationship was right-clicked in the EER diagram, EDIT RELATIONSHIP selected, and then the **RELATIONSHIP** tab selected, the visibility radio buttons were displayed. However, if the visibility settings were changed there was no effect. ([Bug#44988](#))

- On Ubuntu Linux, MySQL Workbench crashed with a segmentation fault when an attempt was made to save the model. ([Bug#44974](#))
- On the Linux version of MySQL Workbench it was not possible to rename a layer by double-clicking it in the **LAYERS** window (which is typically docked to the right hand side of the screen).

A layer editor has now been added to the Linux version of MySQL Workbench. This can be activated by double-clicking the layer's title. ([Bug#44972](#))

- It was not possible to rename a table created using **ADD TABLE**, by copying text into its text field. ([Bug#44913](#))
- In the **FLAGS** box, which is located in the **COLUMN DETAILS** panel in the **TABLE EDITOR**, the same flags were not listed as available for both the `INT` and `INTEGER` data types. `INT` has both `UNSIGNED` and `ZEROFILL` listed, but these were not listed for the column if it was of type `INTEGER`. Also, the **AI** checkbox was not selectable for columns defined as `INTEGER`. ([Bug#44872](#))
- Using the **FOREIGN KEYS** tab of the **TABLE EDITOR**, it was not possible to correctly create multiple foreign key relationships. ([Bug#44701](#))
- MySQL Workbench appeared not to display table rows in an EER diagram for tables that had more than 20 columns.

MySQL Workbench now includes the ability to manually resize the table, besides automatically limiting the number of columns to a user selectable value, by order of appearance, regardless of being key columns or not. ([Bug#44675](#))

- On Debian Linux (Lenny), after adding a second table and selecting AI on the first column element, MySQL Workbench gener-

ated a segmentation fault. ([Bug#44612](#))

- In the **WORKBENCH PREFERENCES** dialog, on the **DIAGRAM** tab, in the **TABLES** panel, the checkboxes **SHOW COLUMN TYPES** and **SHOW COLUMN FLAGS** did not work correctly. Selecting or deselecting **SHOW COLUMN TYPES** only had an effect when MySQL Workbench was restarted. Selecting **SHOW COLUMN FLAGS** had no effect at all, and the flags were never displayed. ([Bug#44586](#))
- When in the **COLUMNS** tab of the **TABLE EDITOR**, if an attempt was made to change a column data type to **INTEGER(n)** this would have no effect and the column data type would remain unchanged. However, if **INT(n)** was entered this was accepted and the data type changed accordingly. ([Bug#44552](#))
- When a new index was created in the **INDEXES** tab for a table that already had indexes, the previously displayed index field remained selected. This caused concern that the newly created index would be on the wrong field. ([Bug#44532](#))
- The Forward Engineer SQL Script wizard did not allow the SQL script text to be copied using the keyboard shortcut. Only the mouse could be used to select and copy the text. ([Bug#44531](#))
- On Ubuntu Linux, the following MySQL Workbench menu items did not work: **PAGE SETUP**, **PRINT PREVIEW** and **PRINT**.

If these menu items were selected the following console messages were generated:

```
** Message: item_activated: 0xaa44bf0 -> 'builtin:wb.page_setup'
** Message: show_progress_becb not implemented
** Message: unhandled message 4: wb.print.setup finished in 0,00s

** Message: show_progress_becb not implemented
create font Helvetica 0 0 11,000000 (200)** Message: item_activated: 0xa4c9800 ->
'plugin:wb.print.printPreview'
** Message: show_progress_becb not implemented

** (mysql-workbench-bin:11732): WARNING **: Could not open editor shared object
'/usr/lib/mysql-workbench/plugins/'
** Message: unhandled message 4: wb.print.printPreview finished in 0,00s

** Message: show_progress_becb not implemented
** Message: item_activated: 0xa4cec08 -> 'plugin:wb.print.print'
** Message: show_progress_becb not implemented
** Message: unhandled message 4: wb.print.print finished in 0,00s

** Message: show_progress_becb not implemented
create font Helvetica 0 0 11,000000 (300)
```

([Bug#44524](#))

- When a Routine Group was renamed, this was not immediately updated in the EER diagram. ([Bug#44503](#))
- When `Workbench:copyToClipboard()` was used from within a plugin, only garbage characters were copied to the Windows clipboard. This affected the operation of plugins.

When a plugin attempted to output the string “abcdefghijklmnopqrstuvwxyz0123456789!\$\$%&/()=” a runtime exception was generated. ([Bug#44461](#))

- The **FORWARD ENGINEER SQL SCRIPT** wizard would write over an existing file without prompting the user for confirmation. ([Bug#44437](#))
- On Mac OS X, an application window's red close button should contain a solid circle if the current document is unsaved. However, for MySQL Workbench this did not work correctly. Although the circle was initially shown, it was not displayed after subsequent edits to the model. ([Bug#44435](#))
- When attempting to delete an existing EER diagram, MySQL Workbench crashed.

The origin of the diagram seemed to have no bearing on whether the crash occurred or not, as it happened both with diagrams created from reverse engineering and those created independently. ([Bug#44407](#))

- Deleting or cutting a Relation from an EER diagram crashed MySQL Workbench. ([Bug#44340](#))
- When Relations were deleted via the Foreign Keys tab, MySQL Workbench intermittently generated an exception:

```
System.AccessViolationException: Attempted to read or write protected memory. This is
often an indication that other memory is corrupt.
  at System.Windows.Forms.Control.MarshaledInvoke(Control caller, Delegate method,
Object[] args, Boolean synchronous)
  at System.Windows.Forms.Control.Invoke(Delegate method, Object[] args)
  at System.Windows.Forms.Control.Invoke(Delegate method)
  at MySQL.GUI.Workbench.Plugins.ObjectEditorPlugin.RefreshFormDataInvoke()
  at MySQL.Grt.DelegateSlot0<void>.<void>.cpp_callback()
  at sigc.pointer_functor0<void>.(pointer_functor0<void>* )
  at sigc.adaptor_functor<sigc::pointer_functor0<void>>
>.(adaptor_functor<sigc::pointer_functor0<void> >* )
```

```
    at sigc.internal.slot_call0<sigc::pointer_functor0<void>,void>.call_it(slot_rep* rep)
    at MySQL.Grt.Db.TableEditorBE.remove_fk(NodeId fk)
    at
MySQL.GUI.Workbench.Plugins.DbMysqlTableEditor.deleteSelectedFKsToolStripMenuItem_Click(Ob
ject sender, EventArgs e)
    at System.Windows.Forms.ToolStripItem.RaiseEvent(Object key, EventArgs e)
    at System.Windows.Forms.ToolStripItem.OnClick(EventArgs e)
    at System.Windows.Forms.ToolStripItem.HandleClick(EventArgs e)
    at System.Windows.Forms.ToolStripItem.HandleMouseUp(MouseEventArgs e)
    at System.Windows.Forms.ToolStripItem.FireEventInteractive(EventArgs e,
ToolStripItemEventType met)
    at System.Windows.Forms.ToolStripItem.FireEvent(EventArgs e, ToolStripItemEventType
met)
    at System.Windows.Forms.ToolStrip.OnMouseUp(MouseEventArgs mea)
    at System.Windows.Forms.ToolStripDropDown.OnMouseUp(MouseEventArgs mea)
    at System.Windows.Forms.Control.WmMouseUp(Message& m, MouseButtons button, Int32
clicks)
    at System.Windows.Forms.Control.WndProc(Message& m)
    at System.Windows.Forms.ScrollableControl.WndProc(Message& m)
    at System.Windows.Forms.ToolStrip.WndProc(Message& m)
    at System.Windows.Forms.ToolStripDropDown.WndProc(Message& m)
    at System.Windows.Forms.Control.ControlNativeWindow.OnMessage(Message& m)
    at System.Windows.Forms.Control.ControlNativeWindow.WndProc(Message& m)
    at System.Windows.Forms.NativeWindow.Callback(IntPtr hWnd, Int32 msg, IntPtr wparam,
IntPtr lparam)
```

([Bug#44326](#))

- When exiting MySQL Workbench with unsaved changes, you are given the choices SAVE, DON'T SAVE and CANCEL. When CANCEL was clicked, MySQL Workbench exited without saving, rather than returning to the application without making any changes.

This only happened when clicking the main application Close button, in the upper right corner. It did not happen if the application was exited by selecting FILE, EXIT from the main menu. ([Bug#44267](#))

- The COPY SQL TO CLIPBOARD menu item, which was normally available when right-clicking on a table in MySQL Workbench, was not displayed. ([Bug#44254](#))
- If the order of columns was changed in the **COLUMNS** tab, this was not reflected in the EER diagram. ([Bug#44240](#))
- In the table editor the VARBINARY data type was not listed in the drop down listbox for columns. ([Bug#44023](#))
- In the table editor, when a column was added with a data type of TIMESTAMP, and the default value of 0 entered, MySQL Workbench erroneously added quotes around the 0. Any generated SQL script therefore contained errors, and would subsequently fail. ([Bug#44006](#))
- The Forward Engineer SQL Script wizard did not emit table INSERT statements in the generated script when requested to do so. ([Bug#43799](#), [Bug#44385](#))
- When synchronizing a model with a database, using the Synchronize Model with Database wizard, the data type translation resulted in errors. ([Bug#42728](#))
- Omit Schema Qualifiers did not omit schema for foreign keys.

When using FILE, EXPORT, FORWARD ENGINEER SQL CREATE SCRIPT and selecting the **OMIT SCHEMA QUALIFIERS** checkbox, the schema for foreign keys were not omitted. ([Bug#42328](#))

- When editing a column's type in the **COLUMNS** tab of the **TABLE EDITOR**, the drop down listbox for selecting the data type displayed BOOL. However, once that had been selected the data type was displayed as BOOLEAN.

Further, if a column data type was set by entering INTEGER in the **DATATYPE** column, it then changed to INT on pressing return. ([Bug#41934](#))

- MySQL Workbench frequently crashed with an unknown exception in Wb.h at line 1010.

This occurred while working on tables within EER diagrams on a design consisting of more than 190 tables and 20 EER diagrams. ([Bug#41325](#))

- Invisible characters in a reverse engineered script caused errors to be generated when the same script was forward engineered.

The problem was due to the reverse engineered script containing a mixture of \n, \r and \r\n line endings. These are now normalized to \n in the reverse engineering code. The validation process now also checks SQL code objects to ensure valid line endings. ([Bug#41254](#))

- When working in the **ROUTINES** tab of the **ROUTINE GROUP** editor, the **ROUTINE** tab did not appear to save changes to routine code.

If code was copied from an external application into the **ROUTINE** tab, then the model saved and MySQL Workbench exited, then on restarting MySQL Workbench, any changes to the routine code were lost. ([Bug#40885](#))

- When trying to synchronize a model to a database, tables defining some columns as **FLOAT** or **DOUBLE** prevented the synchronization SQL script from working.

The SQL generated used the syntax **FLOAT(256)** or **DOUBLE(256)** which was rejected by MySQL Server 5.0.51a.

Editing the generated script manually to use just **FLOAT** or **DOUBLE** worked. However, MySQL Workbench then assumed the target table was not correctly synchronized, and prompted the user accordingly. ([Bug#40169](#))

- When a relationship was fully visible in the EER diagram, if the user chose to have it drawn split, or hidden, MySQL Workbench behaved as expected. However, if the relationship was hidden, and the user chose to have it drawn split, or fully visible again, the relationship was not properly redrawn. The user had to move either of the tables involved in the relationship to have the relationship redrawn. ([Bug#40015](#))
- When editing a foreign key in the Foreign Keys tab of the Table Editor, and then removing the corresponding relationship from the EER Diagram canvas, the Foreign Keys tab remained open and populated, instead of being cleared and closed. ([Bug#39478](#))
- When tables with relationship connectors were dragged or copied from one EER diagram to another, the connectors were not always drawn. ([Bug#38545](#))

A.2.8. Changes in MySQL Workbench 5.1.12 (27th April 2009 beta)

This section documents all changes and bug fixes that have been applied since the release of 5.1.11.

Bugs fixed:

- In the **FOREIGN KEY** tab of the Relationship Editor, the two buttons labeled **EDIT TABLE...** were inactive. Clicking them had no effect. ([Bug#45391](#))
- When running on Ubuntu Linux the edit window in MySQL Workbench was not expanded horizontally by default.

When editing existing routines, the horizontal scroll-bar covered 100% of the visible area, no matter how long any of the rows are. This meant that if a row expanded outside the visible area, it was not possible to scroll sideways to see the rest of the row. All of the row was there, but the right part was not visible. ([Bug#44296](#))

- It was not possible to change the Default Collation for any schema. ([Bug#44220](#))

A.2.9. Changes in MySQL Workbench 5.1.11 (Not yet released beta)

This section documents all changes and bug fixes that have been applied since the release of 5.1.10.

Bugs fixed:

- MySQL Workbench crashed when attempting to delete an EER diagram from the **EER DIAGRAMS** section of the main project tab. ([Bug#44245](#))
- When using MySQL Workbench on Linux, if a new layer was created it was not possible to then change the layer's name in the Properties tab. ([Bug#44202](#))

A.2.10. Changes in MySQL Workbench 5.1.10 (10th April 2009 beta)

This section documents all changes and bug fixes that have been applied since the release of 5.1.9.

Bugs fixed:

- It was not possible to edit the cardinality of existing relations using the MAC OS X version of MySQL Workbench. Attempting to change the cardinality via the radio buttons had no effect. For example, it was not possible to change a relation from one-to-many to one-to-one. ([Bug#4043](#))
- When a nonstandard port was specified in **DATABASE**, **FORWARD ENGINEER...**, such as 3307, MySQL Workbench still attemp-

ted to connect to port 3306. ([Bug#44014](#))

- When a foreign key was created, a column was selected in the original table, but no column was selected in the foreign table. If the project was saved and an attempt was made to reload it the following error message was generated:

```
Unserializing GRT data - Inserting null value to not null list
```

This error prevented MySQL Workbench from opening the project file, resulting in all model data being inaccessible. ([Bug#43997](#))

- The name of the table displayed on the tab in the Table Editor did not update when the table name was changed in the editor. ([Bug#43960](#))
- Data entered using the Inserts tab in the Table Editor was not displayed in the Inserts tab, but was saved to the MWB file when the project was saved. The data could be viewed in MySQL Workbench running on platforms other than Mac OS X, but the Mac OS X version of MySQL Workbench did not display the data entered using the Inserts tab. ([Bug#43907](#))
- When a plugin was installed using the TOOLS, INSTALL PLUGIN/MODULE FILE... menu option, MySQL Workbench copied the selected plugin and reported:

```
'Copied module /Users/tilman/Desktop/SymfonyYmlExport.grt.lua to  
'/Users/tilman/Library/Application Support/MySQL/Workbench/modules/SymfonyYmlExport.grt.lua'  
Please restart Workbench for the change to take effect.'
```

When restarted, MySQL Workbench crashed. MySQL Workbench could only be fully restarted by first removing the installed file from the modules directory. ([Bug#43906](#))

- Soon after MySQL Workbench was launched error popup dialogs were generated with the message:

```
"AXDocument" attribute unsupported by <some component>
```

This occurred on the Mac OS X version of MySQL Workbench, when a third-party usage monitoring utility was running. This error was due to the fact that custom exception reporting was disabled by default. ([Bug#43872](#))

- An attempt to edit a table resulted in the following error:

```
The plugin db.mysql.editors.mwbplugin does not  
contain the published object DbMysqlTableEditor
```

This only happened if the project was opened immediately after MySQL Workbench was started. ([Bug#43863](#))

- If a MySQL Workbench project file was opened, all tabs closed, and then FILE, NEW selected, MySQL Workbench crashed with the following error message:

```
Unknown exception caught in c:\users\tax\documents\visual studio 2008\projects\wb  
5.1\workbench\backend\windows\wbpriate.wr\src\Wb.h at line 994.
```

([Bug#43850](#))

- If an attempt was made to use the undo feature after having first saved a project, MySQL Workbench crashed. ([Bug#43849](#))
- When creating a new 1:n link MySQL Workbench froze. ([Bug#43812](#))
- Using the Navigator to zoom in and out of an EER diagram view caused MySQL Workbench to crash. ([Bug#43782](#))
- The Many-to-Many Table generated by MySQL Workbench was of engine type Server Default. It should have been of type InnoDB. ([Bug#43776](#))
- On opening a diagram the file tables layout was broken. All tables were placed in the top left corner.

It appeared that MySQL Workbench stored the diagram layout in the MWB file correctly but it was not retrieved correctly on file load. ([Bug#43455](#))

- In the EER Diagram view, if the Tables folder was expanded in the Catalog Palette, and then a table dragged onto the EER Diagram, the expanded Tables folder immediately collapsed. ([Bug#41922](#))

A.2.11. Changes in MySQL Workbench 5.1.9 (Not yet released beta)

This section documents all changes and bug fixes that have been applied since the release of 5.1.8.

Bugs fixed:

- In Model View after performing EDIT, SELECT, SELECT ALL, it was not then possible to undo the operation. ([Bug#43225](#))
- When a script was processed by FILE, IMPORT, REVERSE ENGINEER MYSQL CREATE SCRIPT, columns with a boolean data type had their data type ignored. As a result, the type of the column in the Table Editor was empty. ([Bug#43094](#))

A.2.12. Changes in MySQL Workbench 5.1.8 (Not yet released)

This section documents all changes and bug fixes that have been applied since the release of 5.1.7.

Bugs fixed:

- In the Reverse Engineer Database wizard it was not possible to select only one table to reverse engineer, and then proceed to the next step. ([Bug#45881](#))
- Using the Inserts Editor caused MySQL Workbench to crash. The error message generated when used was:

```
** Message: =====
** Message: refresh_gui_becb unhandled refresh: 20 RefreshTimer
```

This was due to the fact that, in the 5.1.4 Alpha version of MySQL Workbench, the Inserts Editor had not been implemented. ([Bug#40671](#))

A.2.13. Changes in MySQL Workbench 5.1.7 (Not yet released)

This section documents all changes and bug fixes that have been applied since the release of 5.1.6.

Bugs fixed:

- MySQL Workbench 5.1.7 for MacOSX crashed on startup. The reason was that it was looking for `libmysqlclient.15`, which was not found in `/usr/local/mysql/lib`. The error generated was:

```
Process:      MySQLWorkbench [14915]
Path:         /Applications/MySQLWorkbench.app/Contents/MacOS/MySQLWorkbench
Identifier:   com.sun.MySQLWorkbench
Version:      ??? (??)
Code Type:    X86 (Native)
Parent Process: launchd [95]

Date/Time:    2009-02-02 18:53:52.120 +0100
OS Version:  Mac OS X 10.5.6 (9G55)
Report Version: 6

Exception Type: EXC_BREAKPOINT (SIGTRAP)
Exception Codes: 0x0000000000000002, 0x0000000000000000
Crashed Thread: 0

Dyld Error Message:
  Library not loaded: /usr/local/mysql/lib/libmysqlclient.15.dylib
  Referenced from: /Applications/MySQLWorkbench.app/Contents/MacOS/MySQLWorkbench
  Reason: image not found
```

Note that MySQL was installed, but the specific version of client library required was not present. ([Bug#42550](#))

A.2.14. Changes in MySQL Workbench 5.1.6 (Not yet released)

This section documents all changes and bug fixes that have been applied since the release of 5.1.5.

Functionality added or changed:

- MySQL Workbench has been changed so that layers and tables listed in the Layers palette are sorted in alphabetical order, making it easier to find the required object. ([Bug#39781](#))

Bugs fixed:

- A dialog displayed a message with a missing filename. The message displayed was:

```
Import of SQL script file '' has finished successfully.
```

Note the filename is missing from the message.

This dialog is located in the [FILE](#), [IMPORT](#), [REVERSE ENGINEER SQL CREATE SCRIPT](#) wizard. It is displayed on the page after importing the file, clicking NEXT and then EXECUTE. ([Bug#39922](#))

- The ADVANCED button displayed the text label &ADVANCED.

This button is located in the [FILE](#), [IMPORT](#), [REVERSE ENGINEER SQL CREATE SCRIPT](#) wizard. It is displayed on the page after importing the file. ([Bug#39921](#))

A.2.15. Changes in MySQL Workbench 5.1.4 (Not yet released)

This section documents all changes and bug fixes that have been applied since the release of 5.1.3.

Functionality added or changed:

- There was a problem where relationships that were hidden could then not be selected in order to bring up their relationship editor. Relationships can now be selected as objects in the Layer window. Once selected, the relationship's [visible](#) property can be set to [True](#) in the Properties window, thus making the relationship visible again. ([Bug#40167](#))

Bugs fixed:

- Loading a model using the Linux version of Workbench resulted in a crash. However, the model loaded correctly with the Windows versions of Workbench. ([Bug#39992](#))
- A model created using the Windows version of Workbench caused the Linux version of Workbench to crash on loading the model. ([Bug#39983](#))

A.3. Changes in Release 5.0

A.3.1. Changes in MySQL Workbench 5.0.30 (18th February 2009)

This section documents all changes and bug fixes that have been applied since the release of 5.0.29.

Bugs fixed:

- MySQL Workbench crashed when the mouse wheel was used. If you scrolled the [OPTIONS](#) tab of the [TABLE EDITOR](#), closed the [TABLE EDITOR](#) and then used the mouse wheel again on the [MYSQL MODEL](#) page, MySQL Workbench crashed. ([Bug#42847](#))
- Introducing a UserType into a model caused the [FILE](#), [EXPORT](#), [FORWARD ENGINEER SQL CREATE SCRIPT](#) wizard to crash. Further, performing a [PLUGINS](#), [OBJECTS](#), [COPY SQL TO CLIPBOARD](#) operation also caused MySQL Workbench to crash. ([Bug#42085](#))
- The Forward Engineer SQL CREATE Script wizard failed to generate a script correctly.

This happened when using the [FILE](#), [EXPORT](#), [FORWARD ENGINEER SQL CREATE SCRIPT](#) facility. If, in the wizard, **OBJECT OF TYPE MYSQL TABLE** was selected, and then all tables added to the **EXCLUSION MASKS** pane, before moving back the required table to the **OBJECTS TO PROCESS** pane, the script was generated for the entire database rather than the selected table. ([Bug#41475](#))

- When a diagram was renamed, the history displayed:

```
Rename 'new name' to 'new name'
```

It should have instead displayed:

```
Rename 'old name' to 'new name'
```

(Bug#41355)

- If a model contained a View that was using a Function, and an attempt was made to Synchronize the database, then an error was generated such as:

```
Error 1305: FUNCTION `bleble` does not exist
```

A similar error was also generated if the Forward Engineer SQL CREATE Script wizard was used. (Bug#40846)

- The viewport, which is the combobox in the top right corner of Workbench, did not scale to less than 40%. However, resizes above 40% worked fine. (Bug#39607)
- The Forward Engineer SQL ALTER Script wizard produced an erroneous script.

If Forward Engineer SQL CREATE Script was used to generate a script and this was then used as an input to Forward Engineer SQL ALTER Script, without having made any changes to the model, then an ALTER script with no changes should be produced. However, the ALTER script showed many changes, even though no changes had been made to the model. (Bug#37709)

A.3.2. Changes in MySQL Workbench 5.0.29 (12th December 2008)

This section documents all changes and bug fixes that have been applied since the release of 5.0.28.

Bugs fixed:

- Workbench crashed when objects other than tables were moved out of a layer. (Bug#41358)
- In the EER Diagram view an icon was not displayed for Not-NULL items. (Bug#41326)
- When a diagram was renamed, the label of the corresponding tab was not automatically updated. However, when the focus was changed, the text was correctly updated. (Bug#38867)
- The table figures in the Diagram view had insufficient information. They did not display information such as constraints or default values. (Bug#38553)
- When the grid was activated, dragged objects on layers were incorrectly placed with an offset of -1,-1. (Bug#35989)
- The last column in a table disappeared in the table editor, and it was not possible to add further columns. (Bug#35905)

A.3.3. Changes in MySQL Workbench 5.0.28 (6th December 2008)

This section documents all changes and bug fixes that have been applied since the release of 5.0.27.

Bugs fixed:

- If you attempted to select several tables in the table list of the **MySQL MODEL** view, and you accidentally included the ADD TABLE button in your selection, then a message box appeared warning of an unknown exception:

```
"Unknown Exception caught in: c:\documents and settings\mysqldev\my documents\visual studio 2005\projects\workbench\backend\windows\wb.wr\src\Wb.h at line 1010"
```

The program did not crash. Only the messagebox appeared. (Bug#41201)

- If two foreign keys were created in a table that referenced a second table and then an attempt was made to delete the relations and the referenced table, MySQL Workbench crashed. (Bug#41025)
- When clicking the + and - buttons in the **PHYSICAL SCHEMATA** pane of the **MySQL MODEL** tab, an **Unhandled Exception** was generated:

```
System.Runtime.InteropServices.SEHException: External component has thrown an exception.
```

(Bug#40971)

- The **REFERENCED COLUMN** pane of the **FOREIGN KEY** tab became cleared if the foreign key was renamed. Subsequently, attempting to choose a **REFERENCED COLUMN** did not display a link in the **EER DIAGRAM** view. In order to get foreign key relationships working again it was necessary to de-select the checkboxes from the **COLUMNS** pane, re-select them, and then select

the REFERENCED COLUMN pane. ([Bug#40649](#))

- When a table was renamed the inserted data was lost. ([Bug#40327](#))
- A complex EER diagram threw an exception whenever an action was performed on it. However, other diagrams in the same MWB file functioned correctly.

The exception generated was:

```
System.Runtime.InteropServices.SEHException: Un composant externe a levé une exception.  
  à wb.ModelViewForm.handle_mouse_button(ModelViewForm*, MouseButton, Boolean, Int32  
, Int32, EventState)  
  à MySQL.Workbench.ModelViewForm.OnMouseUp(MouseEventArgs e, Int32 X, Int32 Y, Keys  
keystate, MouseButtons buttons)  
  à MySQL.GUI.Workbench.ModelViewForm.CanvasPanel_MouseUp(Object sender, MouseEventArgs  
e)  
  à System.Windows.Forms.Control.OnMouseUp(MouseEventArgs e)  
  à MySQL.Utilities.WindowsCanvasViewerPanel.OnMouseUp(MouseEventArgs e)  
  à System.Windows.Forms.Control.WmMouseUp(Message& m, MouseButtons button, Int32  
clicks)  
  à System.Windows.Forms.Control.WndProc(Message& m)  
  à System.Windows.Forms.ScrollableControl.WndProc(Message& m)  
  à System.Windows.Forms.Control.ControlNativeWindow.OnMessage(Message& m)  
  à System.Windows.Forms.Control.ControlNativeWindow.WndProc(Message& m)  
  à System.Windows.Forms.NativeWindow.Callback(IntPtr hWnd, Int32 msg, IntPtr wparam,  
IntPtr lparam)
```

([Bug#39360](#))

A.3.4. Changes in MySQL Workbench 5.0.27 (7th November 2008)

This section documents all changes and bug fixes that have been applied since the release of 5.0.26.

Functionality added or changed:

- There was a problem where relationships that were hidden could then not be selected in order to bring up their relationship editor. Relationships can now be selected as objects in the Layer window. Once selected, the relationship's `visible` property can be set to `True` in the Properties window, thus making the relationship visible again. ([Bug#40167](#))

Bugs fixed:

- When a stored routine was edited, the edit cursor jumped back to the start of the page unless typing was constant. ([Bug#40426](#))
- When using the COPY INSERT TO CLIPBOARD menu item the generated SQL code was incorrect. The "S" was missing from "VALUES" and the data was not included. This resulted in SQL code such as:

```
INSERT INTO `table1` (`table1_id`, `descr`) VALUE ();
```

([Bug#40041](#))

- If a trigger was renamed, and the design then synched with a database instance, the generated SQL created a trigger with the new name and then dropped the trigger with the old name. This resulted in the following error:

```
Error 1235: This version of MySQL doesn't yet support  
'multiple triggers with the same action time and event for one table'
```

([Bug#39989](#))

- The COPY SQL TO CLIPBOARD action (right click menu on table) did not use Windows-compatible line endings. ([Bug#39476](#))
- When a column had a data type `BOOLEAN` and it was exported using FORWARD ENGINEER SQL ALTER, the exported type was `BOOLEAN(2)` instead of `BOOLEAN`. ([Bug#39257](#))
- Workbench application performance was poor, with slow loading times and excessive memory usage. ([Bug#38439](#))
- When a DBDesigner model with 333 tables was imported into Workbench the RAM usage went up to approximately 1GB. Workbench then crashed with the following exception:

```
Error creating cairo context: out of memory
```

(Bug#37178)

A.3.5. Changes in MySQL Workbench 5.0.26 (16th October 2008)

This section documents all changes and bug fixes that have been applied since the release of 5.0.25.

Bugs fixed:

- When attempting to export a model using the FILE, EXPORT, FORWARD ENGINEER SQL CREATE SCRIPT menu item, Workbench crashed on clicking the wizard's FINISH button. ([Bug#39578](#))
- The **COPY INSERT TO CLIPBOARD** action generated SQL with lower case keywords. This was not consistent with the behavior of the **COPY SQL TO CLIPBOARD** action. ([Bug#39477](#))
- Renaming a table and then selecting FORWARD ENGINEER SQL ALTER SCRIPT did not result in a RENAME statement. Instead, DROP and CREATE statements were generated. ([Bug#39256](#))
- The script generated by the EXPORT, FORWARD ENGINEER SQL CREATE SCRIPT menu item contained invalid statements when using two schemata. ([Bug#39211](#))
- Exported SQL code containing a trigger that called a procedure would fail when an INSERT activated the trigger. ([Bug#39088](#))

A.3.6. Changes in MySQL Workbench 5.0.25 (12th September 2008)

This section documents all changes and bug fixes that have been applied since the release of 5.0.24.

Bugs fixed:

- If the user closed all tabs and then quit, Workbench crashed. ([Bug#39346](#))
- Foreign keys referencing a deleted table were not removed. ([Bug#39150](#))
- FORWARD ENGINEER SQL CREATE SCRIPT and FORWARD ENGINEER SQL ALTER SCRIPT generated scripts that did not put index names in quotes. ([Bug#39140](#))
- When Workbench was started with the GRT Shell tab opened, the object tree in the **GRT TREE** pane was not displayed. ([Bug#39122](#))
- When triggers were exported with the **GENERATE DROP TABLES STATEMENTS** option checked, DROP TRIGGER IF EXISTS did not appear in the exported SQL. ([Bug#39119](#))
- The **TRIGGERS** tab would always enable Insert mode when opened. ([Bug#39118](#))
- In the **FOREIGN KEY** tab of the **TABLE EDITOR**, the dropdown menu that is displayed on clicking in the **REFERENCED TABLE** column, listed table names by creation date, rather than by sorted name. ([Bug#38944](#))
- If any DEFAULT properties were defined for a model, they appeared to be lost after saving the model and restarting Workbench. ([Bug#38825](#))
- When you loaded a UTF-8 encoded script file into Workbench, the embedded SQL editor replaced international characters with the ? symbol. ([Bug#38783](#))
- When creating Views and Routines, the entry in the **UNDO HISTORY** window showed "Parse MySQL View" instead of "View Created", and "Parse MySQL Routine" instead of "Routine Created".

When subsequently undoing this operation the correct text was displayed. Performing a redo then resulted in the incorrect text being displayed again.

Additionally, when undoing a Routine Group, the previous undo action in the history was incorrectly renamed and the last entry in the history was deleted. ([Bug#36047](#))

- In the **TABLE EDITOR** tab, wherever data could be entered, such as in the **FOREIGN KEY NAME** entry field, the default wrap protocol was to go to a new line. This resulted in text that was only partially visible. ([Bug#34510](#))
- The synchronization wizard could show a diff tree for schemata different from those that had been selected. ([Bug#32365](#))

A.3.7. Changes in MySQL Workbench 5.0.24 (12th August 2008)

This section documents all changes and bug fixes that have been applied since the release of 5.0.23.

Functionality added or changed:

- In the `MySQLGrtShell.exe` program the **VALUES** tab has been renamed to **GRT TREE**. However, the **GRT TREE** tab only shows a root node because there is no GRT Tree loaded when the `Shell` is started in standalone mode. ([Bug#35052](#))

Bugs fixed:

- Indexes listed when the **INDEX** tab was selected could not be deleted if the index type was `FOREIGN`. ([Bug#38639](#))
- When the menu item `MODEL`, `VALIDATION`, `VALIDATE ALL` was selected, and an error dialog subsequently displayed, the dialog error message had a missing dot separator between the database name and table name. ([Bug#38632](#))
- When a DBDesigner 4 model that contained duplicate relationships was imported into Workbench, and then exported, the resultant script would fail when executed on MySQL server. ([Bug#38488](#))
- It was not possible to synchronize a model to an external database, if the model contained triggers. ([Bug#38436](#))
- When resizing the comment column under **PHYSICAL SCHEMATA** view in column format, the column resize was reverted when switching between schemas. ([Bug#38431](#))
- An attempt to copy a table and then paste it into a new schema resulted in an `Unknown Exception` being generated. ([Bug#38429](#))
- If you created a new view with an `OR REPLACE` clause, the `FORWARD ENGINEER SQL CREATE SCRIPT` output contained the `OR REPLACE` clause twice. ([Bug#38337](#))
- When a DBDesigner 4 XML file was imported into Workbench the `INSERT` statements were incorrectly converted. ([Bug#38196](#))
- Importing a script that specified an incorrect data type required Workbench to close. ([Bug#38146](#))
- Workbench crashed when using the `MODEL`, `VALIDATION(MYSQL)`, `VALIDATE ALL` menu item on a model that contained a dangling foreign key index. ([Bug#38115](#))
- Foreign key options (`onDelete`, `onUpdate`) are not imported from DBDesigner schema. ([Bug#37794](#))
- In the `mysql-workbench-oss-5.0.23-win32-noinstall` version of Workbench the menu item `PLUGINS`, `OBJECTS`, `COPY SQL TO CLIPBOARD` did not work. ([Bug#37736](#))
- When synchronizing the database, table comments were not updated. However, column comments worked as expected. ([Bug#37686](#))
- Running `HELP`, `UPDATE...` crashes Workbench when the wizard comes to the point where it is trying to close Workbench. ([Bug#37665](#))
- `DATABASE`, `SYNCHRONIZE` did not update the model view when the table was changed in the database, until after Workbench was restarted. ([Bug#37634](#))
- `FORWARD ENGINEER SQL CREATE SCRIPT` did not reflect changes made to the model. ([Bug#37574](#))
- When using the `FORWARD ENGINEER SQL CREATE SCRIPT`, columns marked as `NOT NULL` were generated as `NOT NULL DEFAULT NULL`. ([Bug#37385](#))
- Errors were generated in SQL code during `FORWARD ENGINEER SCHEMA` for Inserts data in `TIMESTAMP` columns. ([Bug#37059](#))
- If a database was imported using `REVERSE ENGINEER SQL ALTER SCRIPT` and the database name changed in Workbench, the script then generated by `FORWARD ENGINEER SQL ALTER SCRIPT` was incorrect. ([Bug#36178](#))
- The auto-increment flag was not cleared internally for a column, when the type of that column was changed to one for which auto-increment is invalid; for example, `char`. When the model was exported using `EXPORT`, `FORWARD ENGINEER SQL CREATE SCRIPT`, the resulting script incorrectly retained the auto-increment flag for the changed column. ([Bug#36085](#))

A.3.8. Changes in MySQL Workbench 5.0.23 (25th June 2008)

This section documents all changes and bug fixes that have been applied since the release of 5.0.22.

Functionality added or changed:

- It was not clear how a stored connection profile could be edited and the changes saved. Tooltips have been added to the relevant buttons and the main documentation clarified. ([Bug#37061](#))

Bugs fixed:

- The FILE, EXPORT, FORWARD ENGINEER SQL CREATE SCRIPT menu item exports a script it is then unable to import using the FILE, EXPORT, REVERSE ENGINEER MYSQL CREATE SCRIPT menu item, as it incorrectly imports comments containing special characters. ([Bug#37563](#), [Bug#37562](#))
- Workbench was failing to correctly export Trigger DDLs. ([Bug#37432](#))
- Using GENERATE SCHEMA DIFF REPORT resulted in a crash. The crash was caused by improper handling of an invalid FK in a table. While this issue is correctly reported by a validation module, in Standard Edition GENERATE SCHEMA DIFF REPORT didn't handle that correctly. ([Bug#37393](#))
- When a new column was added to a table Inserts data was deleted. ([Bug#37192](#))
- Trying to edit a table in a new window displays an error message dialog:

```
plugin:wb.edit.editSelectedInNewWindow
Invalid plugin
Invalid plugin wb.edit.editSelectedInNewWindow
```

([Bug#37180](#))

- If you try to place a new image into an EER Diagram and select an invalid filetype, you get a error message dialog with the following text:

```
CAIRO ERROR: INVALID MATRIX (NOT INVERTIBLE)
```

If you then click OK to clear the dialog and then try to select PLACE A NEW TABLE, the error message dialog is displayed again. ([Bug#37079](#))

- The FORWARD ENGINEER wizard did not report connection status correctly. If invalid database credentials were entered, the wizard reported success, even though the connection failed. ([Bug#37060](#))
- Incorrect behavior when editing a table. When the COLUMNS tab is selected, if you want to delete multiple selected tables at once, Workbench removes the wrong columns. ([Bug#37045](#))
- The script generated by the FILE, EXPORT, FORWARD ENGINEER SQL ALTER SCRIPT menu item contains syntax errors. ([Bug#36889](#))
- The export filter did not properly filter tables. ([Bug#36739](#))
- Workbench generated incorrect syntax when attempting to synchronize with a live server. The resultant code was missing commas which resulted in a syntax error. ([Bug#36674](#))
- After reverse engineering an SQL create script and drawing some EER diagrams, a subsequent import of the same script destroys the EER diagrams. All tables in the catalog are updated, but the reference of the table in the diagram to the table in the catalog is lost. The tables in the diagram are still visible, but do not correspond to the table in the catalog.

After closing and re-opening the file, all diagrams are empty and it is impossible to delete the diagrams. However, in the overview in the upper right corner, the tables placed in the diagram are still visible. ([Bug#36381](#))

- Mouse wheel does not work when you double-click a table and select the OPTIONS tab. ([Bug#36374](#))
- When FILE, EXPORT, FORWARD ENGINEER SQL ALTER SCRIPT menu item is selected it causes an ALTER Script Generation (Script Synchronization) error. ([Bug#36355](#))
- The behavior of the SYNCHRONIZE wizard was inconsistent when cancelled and re-run. ([Bug#36177](#))
- Several windows and tabs have fields which are either not completely visible or are obscured by labels that overlap the field. ([Bug#36115](#))

- When creating a [Schema Diff Report](#) from the local model to a live database, the wizard crashed with an unhandled exception. ([Bug#35878](#))
- Collapsing of the EER Diagram section of the **MySQL MODEL** tab is not retained after program relaunch. ([Bug#35717](#))
- In the **MySQL MODEL** tab, in the summary line for **PHYSICAL SCHEMATA**, there are three icons, one for large icon view, one for small icon view, one for list view. Changing the view is not saved between application launches. ([Bug#35716](#))
- Performing a [DATABASE SYNCHRONIZATION](#) resulted in erroneous [ALTER](#) statements being generated. ([Bug#34812](#))
- Menu item was incorrectly named [GENERATE SCHEMA DIFF REPORT](#), when it should have been called [GENERATE CATALOG DIFF REPORT](#). ([Bug#34398](#))
- Workbench failed to restore window states, window positions and side-panel sizes from the previous execution of the application. ([Bug#32442](#))
- The [AUTO_INCREMENT](#) attribute is now ignored on import for column types that do not support it. ([Bug#31986](#))

A.3.9. Changes in MySQL Workbench 5.0.22 (27th May 2008)

This section documents all changes and bug fixes that have been applied since the release of 5.0.21.

Bugs fixed:

- Can not add values for [TIMESTAMP](#) columns in the [INSERTS](#) editor. ([Bug#37009](#))
- When columns are added to, or removed from a table, Workbench deletes all [INSERTS](#) data. ([Bug#37008](#))
- Trigger definition auto-formatting resulted in malformed code. ([Bug#36815](#), [Bug#37685](#))
- The script generated by the [FILE](#), [EXPORT](#), [FORWARD ENGINEER SQL](#) [CREATE SCRIPT](#) menu item contains a spurious quote mark. ([Bug#36753](#))
- For [CREATE TABLE](#) statements, [TIME](#) column default values were not quoted properly. ([Bug#36669](#))
- Print preview in landscape orientation did not work correctly. ([Bug#36647](#))
- When opening a model created with an earlier version of Workbench, the [INDEXES](#) tab displayed indexes of type [FOREIGN](#) as type [INDEX](#), and it was not possible to change them back to [FOREIGN](#). ([Bug#36453](#))
- If a table column definition allows [NULL](#) and has been set with a default of [NULL](#), integrity validation operations complained that the default value for the column is invalid. ([Bug#36397](#))
- After use of Control-X to cut text from a text-edit box and Control-Z to undo the operation, the canvas was updated correctly but not the text box. ([Bug#36358](#))
- Shifted content could not be scrolled or navigated. ([Bug#36328](#))
- The mousewheel scrolled the overview pane when it was open behind the insert-editor. ([Bug#36253](#))
- View renaming in overview did not work properly and has been disabled. ([Bug#36202](#))
- The Copy to SQL operation caused a crash. ([Bug#36184](#))
- Dragging objects out of a layer did not work properly. ([Bug#36053](#))
- The enabled/disabled status of items in the [EDIT](#) menu was not updated properly. ([Bug#35962](#))
- Relationships were drawn over tables. ([Bug#35867](#))
- The script generated by database synchronize contained errors. ([Bug#35644](#))
- Setting up foreign key relationships across multiple schemas did not work. ([Bug#34546](#))
- Scrollbars now appear correctly when editor windows are reduced in height. ([Bug#32454](#))
- Table partitioning information was not exported properly. ([Bug#32226](#))

A.3.10. Changes in MySQL Workbench 5.0.21 (27th April 2008)

This section documents all changes and bug fixes that have been applied since the release of 5.0.20.

Bugs fixed:

- The undo operation did not completely undo a relationship between two tables. It removed only the line drawn between two tables, but did not undo the fields and keys. ([Bug#36645](#))
- Double clicking a column-heading separator in Find results caused a crash. ([Bug#36266](#))
- The scripts generated by the FILE, EXPORT, FORWARD ENGINEER SQL ALTER SCRIPT and FILE, EXPORT, FORWARD ENGINEER SQL CREATE SCRIPT include unnecessary SQL code. ([Bug#36170](#))
- The COPY CONNECTION NN menu item on the context menu of a connection does not have a complementary PASTE CONNECTION menu item. The EDIT menu has a greyed-out PASTE CONNECTION menu item. ([Bug#36166](#))
- When a schema used InnoDB, and then was switched to use MyISAM, the script generated by **FORWARD ENGINEER SQL CREATE SCRIPT** still contained InnoDB-only syntax. ([Bug#35947](#))
- FORWARD ENGINEER wizard failed to create a table, but did not show any error messages. ([Bug#35874](#))
- Saving a file restores the column widths of the list view to default under **PHYSICAL SCHEMATA**. ([Bug#35718](#))
- When making a column a primary key and this column has NULL as default value, this default value is not changed. When the table gets synchronized back to the database Workbench creates a statement such as:

```
ALTER TABLE `test_defhan`.`table1` CHANGE COLUMN `id_table1` `id_table1` INT(11) NOT NULL  
DEFAULT NULL, ...
```

This leads to an error:

ERROR 1067: INVALID DEFAULT VALUE FOR 'ID_TABLE1' ([Bug#32972](#))

A.3.11. Changes in MySQL Workbench 5.0.20 (26th April 2008)

This section documents all changes and bug fixes that have been applied since the release of 5.0.19.

Bugs fixed:

- Re-creating a deleted relationship caused a crash. ([Bug#36385](#))
- The message log on the Forward Engineer Progress/Results Advanced dialog had no scroll bar. ([Bug#36192](#))

A.3.12. Changes in MySQL Workbench 5.0.19 (15th April 2008)

This section documents all changes and bug fixes that have been applied since the release of 5.0.18rc.

Bugs fixed:

- The HTML Basic Single Page DBDoc report from the MODEL -> DBDOC -> MODEL REPORTING menu option was missing the schema and table numbers. ([Bug#36060](#))
- Forward Engineer SQL CREATE Script wizard generated no output script. Further, no error or warning messages appeared to be generated that might explain this.

The error message in this case was displayed in the Advanced Log, which was not visible to the user. MySQL Workbench was changed so that the Advanced Log appeared to the user if it received an error message. ([Bug#34430](#))

A.3.13. Changes in MySQL Workbench 5.0.18rc (not released)

This section documents all changes and bug fixes that have been applied since the release of 5.0.17rc.

Functionality added or changed:

- Foreign key labels could not be hidden, and displayed labels were not centered. There are now options to hide all connection captions, and to center captions. ([Bug#30902](#))

Bugs fixed:

- In the table editor, setting the input focus by clicking the mouse did not work. ([Bug#35969](#))
- The Reference Column dropdown used during foreign key creation was slow to display. ([Bug#35948](#))
- In the table editors foreign key Tab, when a column for the foreign key is checked (right pane), the Referenced Column dropdown opens. Pressing Escape at this point caused a crash. ([Bug#35926](#))
- After changing the Row Format option, closing the table editor and opening a new document caused a crash. ([Bug#35925](#))
- If a table in an EER Diagram was double-clicked in an attempt to open it for editing in the Table Editor, the following error was generated:

```
Cannot load selected plugin(\db.mysql.editors.wbp.fe.dll::DbMysqlTableEditor)
```

([Bug#35897](#))

- Synchronizing the data model with a live database from the SQL Diff Tree dialog resulted in a crash. ([Bug#35884](#))
- Creating a Schema Diff Report from the local model to a live database caused a crash. ([Bug#35878](#))
- The Pack Keys option could not be saved. ([Bug#35872](#))
- Some menus or submenus had items enabled when the corresponding features were disabled. ([Bug#35870](#))
- The Connection Caption option did not work properly. ([Bug#35859](#))
- The status of a connection line in a table diagram was not updated when a foreign key relationship between tables was changed. ([Bug#35800](#))
- The FILE -> EXPORT -> EXPORT AS PNG menu item was enabled under some circumstances in which it should have been disabled. ([Bug#35746](#))
- Scrolling was slow for table models with large numbers of tables. ([Bug#35655](#))
- Pressing Ctrl-Z to undo the last change in an SQL Script text box deleted the entire script. ([Bug#35649](#))
- Workbench is unable to read files such as Workbench Model Files from a non-English directory. ([Bug#35547](#))
- Workbench allowed table comments to be entered longer than the maximum length of 60 characters. ([Bug#34507](#))
- A crash could occur during foreign key creation. ([Bug#33545](#))
- Autoplacing for display of complex schemas has been improved. ([Bug#32888](#))
- Typing `q` in the GRT Shell caused a crash. ([Bug#32755](#))

A.3.14. Changes in MySQL Workbench 5.0.17rc (07th April 2008)

This section documents all changes and bug fixes that have been applied since the release of 5.0.16rc.

Bugs fixed:

- Creating a new view and then deleting it caused a `System.AccessViolationException`. ([Bug#35840](#))
- Editing a stored procedure within Workbench could cause an exception. ([Bug#35828](#))
- The modified timestamp for an existing model was not correctly updated for all changes. ([Bug#35719](#))
- Identifiers for field names in DML SQL statements would not be quoted correctly, allowing for reserved words to be included in the SQL statements. ([Bug#35710](#))
- Workbench would crash repeatedly when drawing the diagram for a table where the referenced column in a foreign key rela-

tionship was blank. ([Bug#35677](#))

- Identifiers using uppercase characters for stored procedures would automatically be modified to lowercase. ([Bug#35650](#))
- When working with the **SQL SCRIPT** editor, it was not possible to select all the text in the display when using **Ctrl A**. ([Bug#35646](#))
- The **MODEL NAVIGATION** window could not be collapsed like other palettes. ([Bug#35642](#))
- Modifying the primary key index definition for within the table view would not update the entity relationship diagram. ([Bug#35639](#))
- When validating an existing model using the Forward Engineer Wizard, MySQL-specific validation would fail. ([Bug#35604](#))
- Deleting an existing layer on a diagram and then editing other objects on the same canvas could generate a number of exceptions, and could corrupt the Workbench file. ([Bug#35603](#))
- Switching to the Connect to Columns notation with an existing model would cause an exception. ([Bug#35601](#))
- Data in **BLOG** and **TEXT** columns defined using the **INSERTS** tab would not be quoted correctly in the resulting SQL. ([Bug#35525](#))
- Opening an existing Workbench model with an invalid foreign key definition would cause an exception. ([Bug#35501](#))
- Moving multiple tables on the same diagram, and then using Undo to revert the model to the original layout, only the first table selected be returned to its original position. ([Bug#35465](#))
- When adding a foreign key relationship within a catalog with an existing entity relationship diagram, the foreign key relationship is not added to the existing diagram. ([Bug#35429](#))
- The precise position of individual connections would not be retained when the schema was saved. ([Bug#35397](#))
- Opening a GRT shell while the table editor is open would raise an exception. ([Bug#35349](#))
- When modifying an existing foreign key relationship, the generated **ALTER** script did not reflect the modification. ([Bug#35265](#), [Bug#35830](#))
- When creating foreign key relationships that point to more than one table, the same foreign key identifier for the same table could be created. This would create invalid SQL code for creating the table. ([Bug#35262](#))
- When importing an existing DB Designer schema, Workbench could crash. ([Bug#35123](#))
- Setting up indexes in both the index and foreign key list views, the mouse pointer would disappear while the entry box was in use. ([Bug#35062](#))
- Double clicking the Catalog title bar undocked the GRT Tree window. ([Bug#34856](#))
- The font for views and routines was not monospace by default. ([Bug#34537](#))
- When using the Forward Engineer Wizard, if an error occurred, the dialog showing the error detail would be incomplete, and determining the reason for the error would be masked because the end of log message would be hidden. ([Bug#34509](#))
- When using the **HIDE MENU ITEMS NOT APPLICABLE TO THIS EDITION** option, a simplified version of the **FIND** dialog box was not available. ([Bug#34493](#))
- Editing the text of the Trigger portion of an existing schemata would introduce additional text into the Trigger definition. ([Bug#34397](#))
- Creating more than five stored procedures or views in a model would cause the dialog box for the operation to move to a different layer, making it inaccessible when using the mouse. ([Bug#34153](#))
- Selecting **EXPORT**, **FORWARD ENGINEER ALTER SCRIPT** from the **FILE** would open a **SQL SCRIPT SYNCRHONIZATION** dialog, rather than export dialog. ([Bug#34099](#))
- When moving more than layer in Model Navigator, only the first layer's position would be reflected correctly in the output. ([Bug#33627](#))

A.3.15. Changes in MySQL Workbench 5.0.16rc (26th March 2008)

This section documents all changes and bug fixes that have been applied since the release of 5.0.15rc.

Functionality added or changed:

- Options and configuration options that affect models can now be set on a model by model basis. Choose OPTIONS from the MODEL menu and choose the **DIAGRAM** tab. ([Bug#34610](#))

Bugs fixed:

- When double clicking on the row in a column as a primary key, the primary key property would be toggled. The editor will now allow you to edit the value when clicking on a data row on the table. ([Bug#35613](#))
- Opening the **INDEXES** portion of a table would generate a unhandled exception error. ([Bug#35598](#))
- When disabling global options on an individual model would fail to honor the model specific options would be ignored. ([Bug#35516](#))
- When placing a 1:n relation, an `index out of range` error could be raised. This could further result in `operation on NULL object: Invalid value` errors when trying to edit the relation. ([Bug#35447](#))
- Setting the value of a numeric column to a negative value was not supported. ([Bug#35442](#))
- Printing an HTML version of the schema would produce a fatal error. ([Bug#35400](#))
- The OK and CANCEL buttons for the **DIAGRAM SIZE** dialog would not be initialized properly. ([Bug#34808](#))
- When using print preview on a diagram, clicking the PRINT button would send a blank page to the printer. ([Bug#34630](#))
- When copying multiple table definitions from one schema to another, only the first table in the selection would be pasted into the new schema. ([Bug#34483](#))
- The **DRAW LINE CROSSING** option would fail to be recognized correctly. You can also now set this on an individual model basis using the OPTIONS optin in the MODEL menu. ([Bug#34248](#))
- Copying an existing module to the plugins directory would trigger a double registration of the modulem, and produce an error. ([Bug#34134](#))
- When exporting a diagram to PDF, some additional lines would be added to the generated PDF. ([Bug#33586](#))
- Placing an image on to the canvas could crash the application. For images larger than the canvas, the image is automatically reduced so that it is properly visible on the canvas for editing. ([Bug#33179](#))
- A 1:m relation in a diagram would fail to be generated properly when exported as a PDF. ([Bug#32882](#))
- The Undo and Redo options would not be applied properly when making modifications to partition definitions. ([Bug#32279](#))

A.3.16. Changes in MySQL Workbench 5.0.15rc (17th March 2008)

This section documents all changes and bug fixes that have been applied since the release of 5.0.14abeta.

Bugs fixed:

- Using **UNDO** on a relationship within a model would cause an exception. ([Bug#35243](#))
- A foreign key relationship to the source table (a reflexive relationship) gives a bad representation in the entity model diagram. ([Bug#35237](#), [Bug#34810](#))
- Generating an **ALTER SCRIPT** or using the synchronize functionality on a model with entity relationships, the relationship lines within the diagram would be generated twice. ([Bug#35213](#))
- Boolean values were unsupported when trying to insert values into a table, the `TRUE` would instead be replaced by a textual, quoted version '`TRUE`'. ([Bug#35205](#))
- Printing a model diagram to PDF or Postscript, results in a corrupt file PDF or Postscript file that does not match the model. ([Bug#35197](#))
- Deleting objects within the overview pane when the corresponding editor pane for those objects is open would cause a crash. ([Bug#35186](#))

- When entering data into the **DEFAULT** column of the table editor, the use of the **Return** key for saving the information about the default value was not supported. ([Bug#35127](#))
- There was a typographical error in the help message for the GRT command `cd`. The word `Absolute` was missing the final `e`. This has been corrected. ([Bug#35119](#))
- When changing the name within a foreign key relationship, the modified name is not reflected in the tables to which the foreign key is related. ([Bug#35093](#))
- Scrollbar navigation did not work after importing a DB Designer schema with a large canvas size. However, you could still navigate using the `Model Navigator` palette. ([Bug#34988](#))
- After importing a DB Designer schema, the following error occurred: "Cairo error: input string not valid UTF-8." ([Bug#34987](#))
- Creating a new file after changing an existing file with modifications could lead to the original being deleted without prompting to save the changes. ([Bug#34976](#))
- When saving an existing model, the **MySQL MODEL** overview panel would scroll to the top of the model definition. ([Bug#34975](#))
- Changing the `drawSplit` property of a connection from the `Properties` palette did not update the `Visibility` section of the connection editor. ([Bug#34934](#))
- Editing a primary key column within a model on Microsoft Vista could cause a crash. ([Bug#34922](#))
- On the `MySQL Model` page, when the large icons view was selected, the `Add Table` icon disappeared. ([Bug#34904](#))
- Incorrect `ALTER` statements are created during the synchronization process if you add foreign keys to an existing or imported model. ([Bug#34897](#))
- A new **GRT INSPECTOR** tab would be created every time the **GRT SHELL** was opened. In addition, manually closing the **GRT INSPECTOR** and **GRT SHELL** components would cause an exception. ([Bug#34857](#))
- Opening an existing MySQL Workbench file after associating the `.mwb` extension with the application leads to a crash when you open a MySQL Workbench file. ([Bug#34849](#))
- When editing a model, the windows and toolbars would realign themselves during selection. This was related to the configured font size and the DPI setting of the monitor, causing the application to redraw the windows to account for the configuration combination. ([Bug#34822](#))
- Attempting to move a table on an EER diagram after deleting a relationship, caused the application to crash. ([Bug#34816](#))
- The **NEW FILE** dialog is nonmodal, and could be hidden by other windows. The dialog is now always drawn on top of other windows. ([Bug#34784](#))
- Changing the column name of a table when you have pending inserts to the table did not change the column name in the corresponding `INSERT` statements. ([Bug#34500](#))
- The `Properties` palette was not cleared when a new project was started. It retained the properties of the last selected object. ([Bug#34433](#))
- Deleting an existing schema with an open table editor would not close the table editor window. The window is now closed when the schema is deleted. ([Bug#34345](#))
- Searching a project specifying `Entire Model` in the `In Location:` drop down list box did not return any results. This applied to the Standard Edition only. ([Bug#34170](#))
- When the page size was changed from A4 to B4 it was not possible to move objects on an EER diagram beyond the old page boundaries. ([Bug#34148](#))
- When editing comments, the **Return** key would move to the next column, which prevented the use of newlines within the comment information. Workbench now allows use of the **Return** key within the comment field. ([Bug#33980](#))
- Where relationship lines crossed, and one of the connectors was changed to `Hidden` or `Draw Split`, the semi circle that indicated the previous intersection was still shown on the remaining connector. ([Bug#33818](#))
- Editing an existing diagram could cause an unhandled exception on Windows Vista. ([Bug#33477](#))
- Identifiers (tables, column, index, triggers and other data types) could be created with names longer than the maximum support by MySQL Server. ([Bug#33265](#))

- The application crashed when attempting to export an SQL CREATE script. ([Bug#33263](#))
- Placing an object on the canvas of an EER diagram where you have reverse engineered an existing database, would lead to multiple copies of the object appearing on the diagram. ([Bug#32891](#))
- When scrolling through a schema, the tables in the schema were not redrawn correctly. ([Bug#32835](#))
- On an EER diagram you could not select a relationship if the connection line wasn't stepped. You can now select a connector even if it is not stepped. ([Bug#32734](#))
- Printing a model when there is no printer connected could result in an application exception. ([Bug#32320](#))

A.3.17. Changes in MySQL Workbench 5.0.14abeta (28th February 2008)

This unscheduled beta release fixes [Bug#34847](#) Other bug fixes that have been applied since the release of 5.0.14beta are also documented.

Bugs fixed:

- MWB files were not saved properly if Workbench crashed. Reopening such files caused Workbench to crash. ([Bug#34848](#))
See also [Bug#34847](#).
- Workbench models created in version 5.0.13 crashed when used with version 5.0.14. The unscheduled Beta release, 5.0.14a fixes this bug. ([Bug#34847](#))
- When clicking the BROWSE button in the image editor, the default file name was `openFileDialog1`. This now defaults to an empty string. ([Bug#34622](#))
- Repeatedly changing the object notation crashed Workbench. This is no longer repeatable. ([Bug#34499](#))
- Importing a DBDesigner file immediately threw an exception. This happened even when software rendering was used. DB-Designer files can now be imported without incident. ([Bug#33588](#))

A.3.18. Changes in MySQL Workbench 5.0.14beta (25th February 2008)

This section documents all changes and bug fixes that have been applied since the release of 5.0.13beta.

The following improvements have been added to this version of Workbench:

- EXPAND ALL and COLLAPSE ALL menu options have been added under the ARRANGE menu. The EXPAND ALL option expands all objects on an EER. This option will display a table's columns if the object notation supports expansion. Indexes will not automatically be expanded unless they were previously expanded and have been collapsed using the COLLAPSE ALL menu option. Some object notations, such as Classic, do not allow for expansion or contraction. COLLAPSE ALL undoes the operation performed by EXPAND ALL.
- A FIT OBJECTS TO CONTENTS option has been added under the ARRANGE menu option. This option expands an object on an EER diagram. For example, if a table has a long column name that is not fully displayed, using this menu option will expand the table making the column name visible.
- A SYSTEM INFORMATION menu option has been added to the HELP menu. This option displays information about your system that is useful when reporting a bug.
- An EXPORT AS SVG menu option has been added under the FILE, EXPORT menu option.
- Because of serious performance and display issues Workbench no longer uses Mesa. For those users who don't have native OpenGL support, Workbench now uses the Windows GDI API. The command line switch for using this mode is `-swrendering`. For more information about running Workbench from the command line see [Section 3.3.2, “Launching MySQL Workbench on Windows”](#).
- The GRT inspector has been improved to support new types. Namely:
 - text
 - longtext

- bool
- color
- file

This makes it much easier to change object properties manually. Multiple selection support has also been improved — you can easily change a value for several selected objects at once.

Bugs fixed:

- When exporting an SQL CREATE script it was possible to create two tables in the same schema with the same name. ([Bug#34668](#))
- After placing related tables on an EER diagram and then removing them using the UNDO menu option, the connection lines between related tables no longer showed up. ([Bug#34601](#))
- When choosing the EXPORT AS PNG menu option the file dialogue box file type was All Files instead of PNG. The same was true for EXPORT AS SINGLEPAGE PDF and EXPORT AS SINGLEPAGE PS. The default is now the appropriate file type. ([Bug#34548](#))
- If there was a relationship between table A and table B and also one between table B and table A, the connection lines appeared on top of each other. Connection lines now appear attached at the related columns. ([Bug#34543](#))
- When there were multiple tables with long identifiers the Physical Schemata section of the MySQL Model page was messy. Table names were obscured and sometimes overlapped. Also, the position of the Add Table icon was not optimal. Now the space between table names is adjusted to the largest entry and the Add Table icon is fixed in the upper left corner. ([Bug#34536](#))
- When returning to the SQL Export Filter page after using the BACK button, filters were no longer selected. Selections now persist. ([Bug#34503](#))
- The export filters were applied more than once when forward engineering an SQL CREATE script. This happened if you exported the script after using the BACK button on the SQL Export Filter page. ([Bug#34501](#))
- When the object notation was Workbench Classic the width of a table on an EER diagram could not be less than the widest column. If there was an enum column with many options, this made for a disproportionately wide table. Table width can now be less than the widest column. ([Bug#34496](#))
- When multiple objects on an EER diagram were selected and deleted, Workbench crashed. This happened when both connections and tables were selected. ([Bug#34434](#))
- Setting a column to AUTO_INCREMENT caused the application to crash. ([Bug#34418](#))
- It was reported that you could not add a primary key to a table imported from a MySQL CREATE script. This was not true but did highlight the fact that the method for adding a primary key was not obvious. Now, in addition to adding a primary key by double clicking the icon to the left of a column in the table editor view, you can also add a primary key by checking the PRIMARY KEY checkbox in the Column Details section of the table editor. ([Bug#34408](#))
- When using the menu option GENERATE SCHEMA DIFF REPORT an exception was thrown. A new tree-less version of the Diff report plugin resolves this problem. ([Bug#34396](#))
- Users failed to be created when exporting an SQL CREATE script. ([Bug#34342](#))
- When a table's expanded property was set to 0, the connection line between related tables, appeared at a diagonal orientation. A connection line is now docked on the sides of a table even when the expanded property is set to 0. ([Bug#34249](#))
- Copying a table from the MySQL Model page to an EER diagram canvas created a duplicate table with the same name as the original. This table did not show up in the Catalog palette or in the appropriate schema in the Physical Schemata section of the MySQL Model page. ([Bug#34230](#))
- Creating a new foreign key did not update an EER diagram. An EER diagram is now updated immediately. ([Bug#34206](#))
- When there were many tables on an EER diagram, constant screen refreshing made the application unusable. The performance of the software rendering mode has been improved. ([Bug#33646](#))
- A table with many columns did not display properly. When the table was expanded on an EER diagram it was impossible to scroll down and view all the columns. Improved rendering has helped solve this problem. However, for very large tables you may have to increase the size of an EER. To do this use the MODEL, DIAGRAM SIZE ... menu option. ([Bug#33367](#))

- When changing the foreign key column of a table on an EER diagram, the foreign key did not change color and Workbench crashed when attempting to save the MWB file. The application no longer crashes and the foreign key is updated. ([Bug#33139](#))
- It was not possible to resize a table that used the `Workbench (Default)` object notation. This was problematic for a number of reasons:
 - Long table names make the table very wide.
 - Column definitions that are long relative to the table name, are truncated.
 - Even if you trimmed column names using the **DIAGRAM** tab of the `Workbench Options` the names were sometimes truncated bled over the table border.

This has been corrected. ([Bug#32981](#))

- When there were two schemata and two EER diagrams tables did not show up on the EER diagram if tables from different schemata were added to different EER diagrams. This was caused by defective software rendering. ([Bug#32588](#))
- When forward engineering to a live database, objects not selected on the `Select Objects` page were still created. This applied to tables, routines, and users. ([Bug#32578](#))
- It was not possible to drag or resize tables on an EER diagram. Tables can now be manually resized. To revert a table to automatic sizing use the `Property` palette and set `manualSizing` to True. ([Bug#32549](#))
- The display turned black when the application was resized. This happened when viewing the `MySQL Model` page or when viewing an EER diagram. ([Bug#23959](#))