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: 2010-11-04 (revision: 23566)

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

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. MySQL is a trademark of Oracle Corporation and/or its affiliates, and shall not be used without Oracle's express written authorization. Other names may be trademarks of their respective owners.

This software and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

This documentation is in prerelease status and is intended for demonstration and preliminary use only. It may not be specific to the hardware on which you are using the software. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to this documentation and will not be responsible for any loss, costs, or damages incurred due to the use of this documentation.

The information contained in this document is for informational sharing purposes only and should be considered in your capacity as a customer advisory board member or pursuant to your beta trial agreement only. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle.

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this material is subject to the terms and conditions of your Oracle Software License and Service Agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced, or distributed to anyone outside Oracle without prior written consent of Oracle or as specifically provided below. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

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 Oracle 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 a authorized representative of Oracle. Oracle and/or its affiliates 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 visit 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 and PDF formats, see the MySQL Documentation Library.

Table of Contents

1. MySQL Workbench Introduction	1
2. MySQL Workbench Editions	2
3. Installing and Launching MySQL Workbench	
3.1. Hardware Requirements	
3.2. Software Requirements	
3.3. Starting MySQL Workbench	
3.3.1. Installing MySQL Workbench on Windows	
3.3.2. Launching MySQL Workbench on Windows	5
3.3.3. Uninstalling MySQL Workbench on Windows	5
3.3.4. Installing MySQL Workbench on Linux	6
3.3.5. Launching MySQL Workbench on Linux	6
3.3.6. Uninstalling MySQL Workbench on Linux	7
3.3.7. Installing MySOL Workbench on Mac OS X	7
3.3.8. Launching MySQL Workbench on Mac OS X	
3.3.9. Uninstalling MySOL Workbench on Mac OS X	
3.4. Activation Procedure (Commercial Version)	
4. Getting Started Tutorial	
41 Administering a MySQL Server	9
4.2 Creating a Model	18
4.3 Adding Data to Your Database	10 24
5 The Home Screen	27 28
5.1 Workbanch Control	20 28
5.1. Workdenene	20
5.2. Workspace	
5.5. Workbench Application Minimum Window Size	
5.4. workeench Preterences	
5.4.1. The General Tab	
5.4.2. The Administrator 1ab	
5.4.3. The SQL Editor Tab	
5.4.4. The Model Tab	
5.4.5. The MySQL Tab	
5.4.6. The Diagram Tab	
5.4.7. The Appearance Tab	
6. SQL Development	35
6.1. Open Connection to start Querying	
6.2. New Connection	35
6.3. Edit Table Data	35
6.4. Edit SQL Script	35
6.5. Manage Connections	
6.6. Manage DB Connections Dialog	
6.6.1. Standard TCP/IP Connection	
6.6.2. Local Socket/Pipe Connection	
6.6.3. Standard TCP/IP over SSH Connection	
6.7. SOL Editor	
6.7.1. Main Menu	
6.7.2. Toolbar	
673 SQL Query Panel	42
674 Main Tabsheets	42
675 Sidebar	45
7 Data Modeling	
7. One an Existing EER Model	رہ۔ 40
7.1 Open an Dabung EER Indeel 7.2 Create new EEP Model	رہ 10
7.2. Create HEW black induction Existing Database	
7.5. Create EER Model from SOL Script	4949 50
7.4. Cleate EEK Moutel Hom SQL Schpt	
/.J. MODEL EGILIOT	
7.5.1. Modeling Menus	
7.5.2. The loolbar	
7.5.5. EEK Diagrams	
7.5.4. The Physical Schemata	
(.5.5. Schema Privileges	60

7.5.6. SQL Scripts and Model Notes	61
7.5.7. The History Palette	61
7.5.8. The Model Navigator Panel	62
7.5.9. The Catalog Tree Palette	62
7.5.10. The Layers Palette	63
7.5.11. The Properties Palette	63
7.6. EER Diagram Editor	64
7.6.1. The Vertical Toolbar	64
7.7. Working with Models	67
7.7.1. Creating Tables	67
7.7.2. Creating Foreign Key Relationships	75
7.7.3. Creating Views	78
7.7.4. Creating Routines and Routine Groups	80
7.7.5. Creating Layers	83
7.7.6. Creating Notes	84
7.7.7. Creating Text Objects	85
7.7.8. Creating Images	85
7.7.9. Reverse Engineering	86
7.1.10. Forward Engineering	95
7.8. Modeling 1 utorials	114
7.8.1. Importing a Data Definition SQL Script	114
7.0.2. Using the Default Schema	110
7.8.4 Designmenting	11/
7.0 Printing	120
7.9. Finding	120
7.10 MySOI Workbanch Schama Validation Plugins (Commercial Version)	120
7 10.1 General Validation	120
7 10 2 MySOL Specific Validation	120
7.11. Customizing DBDoc Model Reporting Templates	121
7111 Supported Template Markers	121
7 11.2 Creating a custom template	127
8 Server Administration	130
8.1. Server Administration	130
8.2. New Server Instance	130
8.3. Manage Data Import/Export	130
8.4. Manage Security	131
8.5. Manage Server Instances	131
8.6. Creating and Managing Server Instances	131
8.6.1. New Server Instance Wizard	131
8.6.2. Manage Server Instances Dialog	133
8.7. Server Administration and Configuration	135
8.7.1. Startup Tab	137
8.7.2. Configuration tab	137
8.7.3. Accounts tab	138
8.7.4. Connections tab	140
8.7.5. Variables tab	140
8.7.6. Data Dump tab	141
8././. Logs tab	144
9. Extending Workbench	146
9.1. GRT and Workbench Data Organization	146
9.2. Modules	14/
9.3. Plugins	148
9.4. Autilig a OOI to a Flugili usilig MFOIIIIS	14ð 170
9.5. The workbench Scripting Shell	149
9.5.1. Exploring the workdenen scripting shell	149
9.5.2. The Sheh window	151
9.6 Tutorial: Writing Plugins	151
10. Keyboard Shortcuts	153
11. MySOL Workbench FAO	156
A. MySOL Workbench Change History	158
A.1. Changes in Release 5.2	158
A.1.1. Changes in MySQL Workbench 5.2.30 (Not yet released GA)	158
A.1.2. Changes in MySQL Workbench 5.2.29 (12 October 2010 GA)	158

A.1.3. Changes in MySQL Workbench 5.2.28 (19 September 2010 GA)	159
A.1.4. Changes in MySQL Workbench 5.2.27 (01 September 2010 GA)	161
A.1.5. Changes in MySQL Workbench 5.2.26 (06 August 2010 GA)	164
A.1.6. Changes in MySQL Workbench 5.2.25 (30 June 2010 GA)	165
A.1.7. Changes in MySQL Workbench 5.2.24 (21 June 2010 RC)	167
A.1.8. Changes in MySQL Workbench 5.2.23 (Internal release only RC)	167
A.1.9. Changes in MySQL Workbench 5.2.22 (02 June 2010 RC)	170
A.1.10. Changes in MySQL Workbench 5.2.21 (12 May 2010 RC)	174
A.1.11. Changes in MySQL Workbench 5.2.20 (27 April 2010 beta)	177
A.1.12. Changes in MySQL Workbench 5.2.19 (16 April 2010 beta)	179
A.1.13. Changes in MySOL Workbench 5.2.18 (13 April 2010 beta)	181
A.1.14. Changes in MySOL Workbench 5.2.17 (02 April 2010 beta)	182
A.1.15. Changes in MySOL Workbench 5.2.16 (17 February 2010 beta)	186
A 1.16. Changes in MySQL Workbench 5.2.15 (28 January 2010 beta)	190
A 1.17. Changes in MySQL Workbench 5.2.14 (21 January 2010 beta)	191
A 1.18 Changes in MySQL Workbench 5.2.13 (Not released beta)	192
A 119 Changes in MySQL Workbench 5.2.12 (Not released beta)	192
A 1 20 Changes in MySQL Workbench 5 2 11 (18 December 2009 heta)	195
A 1.20. Changes in MySQL Workbench 5.2.10 (0) December 2009 beta)	198
A 1.21 Changes in MySQL Workbench 5.2.9 (Internal release only beta)	200
A 1.22. Changes in MySQL Workbane 5.2.8 (18 Navambar 2009 bata)	200
A 1.23. Changes in MySQL Workbanch 5.2.7 (To rovember 2009 Octa)	202
A 1.24. Changes in MySQL Workbench 5.2.7 (Internal release only alpha)	203
A 1.25. Changes in MySQL workbench 5.25. (21 October 2009 apria)	204
A 1.20. Changes in MySQL workbench 5.2.5 (internal Release Only alpha)	205
A.1.27. Changes in MySQL workbench 5.2.4 (07 October 2009 alpha)	205
A.1.28. Changes in MySQL workbench 5.2.3 (15 September 2009 alpha)	207
A.1.29. Changes in MySQL Workbench 5.2.2 (27 July 2009 alpha)	208
A.1.30. Changes in MySQL Workbench 5.2.1 (22 May 2009 alpha)	209
A.1.51. Changes in MySQL Workbench 5.2.0 (30 April 2009 alpha)	209
A.2. Changes in Release 5.1	209
A.2.1. Changes in MySQL Workbench 5.1.19 (06 September 2010 GA)	209
A.2.2. Changes in MySQL Workbench 5.1.18 (03 September 2009 GA)	209
A.2.3. Changes in MySQL Workbench 5.1.17 (14 August 2009 beta)	211
A.2.4. Changes in MySQL Workbench 5.1.16 (30 June 2009 beta)	216
A.2.5. Changes in MySQL Workbench 5.1.15 (26 June 2009 beta)	217
A.2.6. Changes in MySQL Workbench 5.1.14 (19 June 2009 beta)	218
A.2.7. Changes in MySQL Workbench 5.1.13 (12 June 2009 beta)	218
A.2.8. Changes in MySQL Workbench 5.1.12 (27 April 2009 beta)	222
A.2.9. Changes in MySQL Workbench 5.1.11 (Not yet released beta)	222
A.2.10. Changes in MySQL Workbench 5.1.10 (10 April 2009 beta)	222
A.2.11. Changes in MySQL Workbench 5.1.9 (Not yet released beta)	224
	224
A.2.12. Changes in MySQL Workbench 5.1.8 (Not yet released)	224
A.2.12. Changes in MySQL Workbench 5.1.8 (Not yet released)	224 224 224
A.2.12. Changes in MySQL Workbench 5.1.8 (Not yet released) A.2.13. Changes in MySQL Workbench 5.1.7 (Not yet released) A.2.14. Changes in MySQL Workbench 5.1.6 (Not yet released)	224 224 224 225
A.2.12. Changes in MySQL Workbench 5.1.8 (Not yet released) A.2.13. Changes in MySQL Workbench 5.1.7 (Not yet released) A.2.14. Changes in MySQL Workbench 5.1.6 (Not yet released) A.2.15. Changes in MySQL Workbench 5.1.4 (Not yet released)	224 224 224 225 225
 A.2.12. Changes in MySQL Workbench 5.1.8 (Not yet released) A.2.13. Changes in MySQL Workbench 5.1.7 (Not yet released) A.2.14. Changes in MySQL Workbench 5.1.6 (Not yet released) A.2.15. Changes in MySQL Workbench 5.1.4 (Not yet released) A.3. Changes in Release 5.0 	224 224 224 225 225 225 225
 A.2.12. Changes in MySQL Workbench 5.1.8 (Not yet released) A.2.13. Changes in MySQL Workbench 5.1.7 (Not yet released) A.2.14. Changes in MySQL Workbench 5.1.6 (Not yet released) A.2.15. Changes in MySQL Workbench 5.1.4 (Not yet released) A.3. Changes in Release 5.0 A.3.1. Changes in MySQL Workbench 5.0.30 (18 February 2009) 	224 224 225 225 225 225 225
 A.2.12. Changes in MySQL Workbench 5.1.8 (Not yet released) A.2.13. Changes in MySQL Workbench 5.1.7 (Not yet released) A.2.14. Changes in MySQL Workbench 5.1.6 (Not yet released) A.2.15. Changes in MySQL Workbench 5.1.4 (Not yet released) A.3. Changes in Release 5.0 A.3.1. Changes in MySQL Workbench 5.0.30 (18 February 2009) A.3.2. Changes in MySQL Workbench 5.0.29 (12 December 2008) 	224 224 224 225 225 225 225 225 225 226
 A.2.12. Changes in MySQL Workbench 5.1.8 (Not yet released) A.2.13. Changes in MySQL Workbench 5.1.7 (Not yet released) A.2.14. Changes in MySQL Workbench 5.1.6 (Not yet released) A.2.15. Changes in MySQL Workbench 5.1.4 (Not yet released) A.3. Changes in Release 5.0 A.3.1. Changes in MySQL Workbench 5.0.30 (18 February 2009) A.3.2. Changes in MySQL Workbench 5.0.29 (12 December 2008) A.3.3. Changes in MySQL Workbench 5.0.28 (06 December 2008) 	224 224 224 225 225 225 225 225 226 227
 A.2.12. Changes in MySQL Workbench 5.1.8 (Not yet released) A.2.13. Changes in MySQL Workbench 5.1.7 (Not yet released) A.2.14. Changes in MySQL Workbench 5.1.6 (Not yet released) A.2.15. Changes in MySQL Workbench 5.1.4 (Not yet released) A.3. Changes in Release 5.0 A.3.1. Changes in MySQL Workbench 5.0.30 (18 February 2009) A.3.2. Changes in MySQL Workbench 5.0.29 (12 December 2008) A.3.3. Changes in MySQL Workbench 5.0.28 (06 December 2008) A.3.4. Changes in MySQL Workbench 5.0.27 (07 November 2008) 	224 224 225 225 225 225 225 225 226 227 227
 A.2.12. Changes in MySQL Workbench 5.1.8 (Not yet released) A.2.13. Changes in MySQL Workbench 5.1.7 (Not yet released) A.2.14. Changes in MySQL Workbench 5.1.6 (Not yet released) A.2.15. Changes in MySQL Workbench 5.1.4 (Not yet released) A.3. Changes in Release 5.0 A.3.1. Changes in MySQL Workbench 5.0.30 (18 February 2009) A.3.2. Changes in MySQL Workbench 5.0.29 (12 December 2008) A.3.3. Changes in MySQL Workbench 5.0.28 (06 December 2008) A.3.4. Changes in MySQL Workbench 5.0.27 (07 November 2008) A.3.5. Changes in MySQL Workbench 5.0.26 (16 October 2008) 	224 224 225 225 225 225 225 226 227 227 227 228
 A.2.12. Changes in MySQL Workbench 5.1.8 (Not yet released) A.2.13. Changes in MySQL Workbench 5.1.7 (Not yet released) A.2.14. Changes in MySQL Workbench 5.1.6 (Not yet released) A.2.15. Changes in MySQL Workbench 5.1.4 (Not yet released) A.3.1. Changes in MySQL Workbench 5.0.30 (18 February 2009) A.3.2. Changes in MySQL Workbench 5.0.29 (12 December 2008) A.3.3. Changes in MySQL Workbench 5.0.27 (07 November 2008) A.3.5. Changes in MySQL Workbench 5.0.26 (16 October 2008) A.3.6. Changes in MySQL Workbench 5.0.25 (12 September 2008) 	224 224 225 225 225 225 225 225 226 227 227 227 228 228
 A.2.12. Changes in MySQL Workbench 5.1.8 (Not yet released) A.2.13. Changes in MySQL Workbench 5.1.7 (Not yet released) A.2.14. Changes in MySQL Workbench 5.1.6 (Not yet released) A.2.15. Changes in MySQL Workbench 5.1.4 (Not yet released) A.3.1. Changes in MySQL Workbench 5.0.30 (18 February 2009) A.3.2. Changes in MySQL Workbench 5.0.29 (12 December 2008) A.3.3. Changes in MySQL Workbench 5.0.27 (07 November 2008) A.3.4. Changes in MySQL Workbench 5.0.26 (16 October 2008) A.3.5. Changes in MySQL Workbench 5.0.25 (12 September 2008) A.3.7. Changes in MySQL Workbench 5.0.24 (12 August 2008) 	224 224 225 225 225 225 225 226 227 227 227 228 228 228 229
 A.2.12. Changes in MySQL Workbench 5.1.8 (Not yet released) A.2.13. Changes in MySQL Workbench 5.1.7 (Not yet released) A.2.14. Changes in MySQL Workbench 5.1.6 (Not yet released) A.2.15. Changes in MySQL Workbench 5.1.4 (Not yet released) A.3.1. Changes in MySQL Workbench 5.0.30 (18 February 2009) A.3.2. Changes in MySQL Workbench 5.0.29 (12 December 2008) A.3.3. Changes in MySQL Workbench 5.0.28 (06 December 2008) A.3.4. Changes in MySQL Workbench 5.0.27 (07 November 2008) A.3.5. Changes in MySQL Workbench 5.0.26 (16 October 2008) A.3.6. Changes in MySQL Workbench 5.0.25 (12 September 2008) A.3.7. Changes in MySQL Workbench 5.0.24 (12 August 2008) A.3.8. Changes in MySQL Workbench 5.0.23 (25 June 2008) 	224 224 225 225 225 225 225 226 227 227 228 228 228 229 230
 A.2.12. Changes in MySQL Workbench 5.1.8 (Not yet released) A.2.13. Changes in MySQL Workbench 5.1.7 (Not yet released) A.2.14. Changes in MySQL Workbench 5.1.6 (Not yet released) A.2.15. Changes in MySQL Workbench 5.1.4 (Not yet released) A.3.1. Changes in MySQL Workbench 5.0.30 (18 February 2009) A.3.2. Changes in MySQL Workbench 5.0.29 (12 December 2008) A.3.3. Changes in MySQL Workbench 5.0.27 (07 November 2008) A.3.4. Changes in MySQL Workbench 5.0.26 (16 October 2008) A.3.5. Changes in MySQL Workbench 5.0.25 (12 September 2008) A.3.7. Changes in MySQL Workbench 5.0.24 (12 August 2008) A.3.8. Changes in MySQL Workbench 5.0.23 (25 June 2008) A.3.9. Changes in MySQL Workbench 5.0.22 (27 May 2008) 	224 224 225 225 225 225 225 225 226 227 227 228 228 228 229 230 232
 A.2.12. Changes in MySQL Workbench 5.1.8 (Not yet released) A.2.13. Changes in MySQL Workbench 5.1.7 (Not yet released) A.2.14. Changes in MySQL Workbench 5.1.6 (Not yet released) A.2.15. Changes in MySQL Workbench 5.1.4 (Not yet released) A.3.1. Changes in MySQL Workbench 5.0.30 (18 February 2009) A.3.2. Changes in MySQL Workbench 5.0.29 (12 December 2008) A.3.3. Changes in MySQL Workbench 5.0.27 (07 November 2008) A.3.4. Changes in MySQL Workbench 5.0.26 (16 October 2008) A.3.5. Changes in MySQL Workbench 5.0.25 (12 September 2008) A.3.7. Changes in MySQL Workbench 5.0.24 (12 August 2008) A.3.8. Changes in MySQL Workbench 5.0.23 (25 June 2008) A.3.9. Changes in MySQL Workbench 5.0.21 (27 April 2008) 	224 224 225 225 225 225 225 225 225 226 227 227 228 228 229 230 232 232
 A.2.12. Changes in MySQL Workbench 5.1.8 (Not yet released) A.2.13. Changes in MySQL Workbench 5.1.7 (Not yet released) A.2.14. Changes in MySQL Workbench 5.1.6 (Not yet released) A.2.15. Changes in MySQL Workbench 5.1.4 (Not yet released) A.3.1. Changes in MySQL Workbench 5.0.30 (18 February 2009) A.3.2. Changes in MySQL Workbench 5.0.29 (12 December 2008) A.3.3. Changes in MySQL Workbench 5.0.29 (12 December 2008) A.3.4. Changes in MySQL Workbench 5.0.27 (07 November 2008) A.3.5. Changes in MySQL Workbench 5.0.26 (16 October 2008) A.3.6. Changes in MySQL Workbench 5.0.27 (12 September 2008) A.3.7. Changes in MySQL Workbench 5.0.23 (25 June 2008) A.3.8. Changes in MySQL Workbench 5.0.22 (27 May 2008) A.3.10. Changes in MySQL Workbench 5.0.21 (27 April 2008) A.3.11. Changes in MySQL Workbench 5.0.20 (26 April 2008) 	224 224 225 225 225 225 225 225 226 227 227 227 228 228 229 230 232 232 233
 A.2.12. Changes in MySQL Workbench 5.1.8 (Not yet released) A.2.13. Changes in MySQL Workbench 5.1.7 (Not yet released) A.2.14. Changes in MySQL Workbench 5.1.6 (Not yet released) A.2.15. Changes in MySQL Workbench 5.1.4 (Not yet released) A.3.1. Changes in MySQL Workbench 5.0.30 (18 February 2009) A.3.2. Changes in MySQL Workbench 5.0.29 (12 December 2008) A.3.3. Changes in MySQL Workbench 5.0.28 (06 December 2008) A.3.4. Changes in MySQL Workbench 5.0.27 (07 November 2008) A.3.5. Changes in MySQL Workbench 5.0.26 (16 October 2008) A.3.6. Changes in MySQL Workbench 5.0.27 (12 September 2008) A.3.7. Changes in MySQL Workbench 5.0.24 (12 August 2008) A.3.8. Changes in MySQL Workbench 5.0.23 (25 June 2008) A.3.9. Changes in MySQL Workbench 5.0.21 (27 May 2008) A.3.10. Changes in MySQL Workbench 5.0.21 (27 April 2008) A.3.11. Changes in MySQL Workbench 5.0.20 (26 April 2008) A.3.12. Changes in MySQL Workbench 5.0.21 (15 April 2008) 	224 224 225 225 225 225 225 225 225 225
 A.2.12. Changes in MySQL Workbench 5.1.8 (Not yet released) A.2.13. Changes in MySQL Workbench 5.1.7 (Not yet released) A.2.14. Changes in MySQL Workbench 5.1.6 (Not yet released) A.2.15. Changes in MySQL Workbench 5.1.4 (Not yet released) A.3.1. Changes in MySQL Workbench 5.0.30 (18 February 2009) A.3.2. Changes in MySQL Workbench 5.0.29 (12 December 2008) A.3.3. Changes in MySQL Workbench 5.0.28 (06 December 2008) A.3.4. Changes in MySQL Workbench 5.0.27 (07 November 2008) A.3.5. Changes in MySQL Workbench 5.0.26 (16 October 2008) A.3.6. Changes in MySQL Workbench 5.0.26 (12 September 2008) A.3.7. Changes in MySQL Workbench 5.0.23 (25 June 2008) A.3.8. Changes in MySQL Workbench 5.0.21 (27 May 2008) A.3.9. Changes in MySQL Workbench 5.0.21 (27 April 2008) A.3.10. Changes in MySQL Workbench 5.0.20 (26 April 2008) A.3.11. Changes in MySQL Workbench 5.0.20 (26 April 2008) A.3.13. Changes in MySQL Workbench 5.0.20 (26 April 2008) A.3.13. Changes in MySQL Workbench 5.0.21 (27 April 2008) A.3.13. Changes in MySQL Workbench 5.0.20 (26 April 2008) A.3.13. Changes in MySQL Workbench 5.0.20 (26 April 2008) A.3.13. Changes in MySQL Workbench 5.0.21 (27 April 2008) A.3.13. Changes in MySQL Workbench 5.0.21 (27 April 2008) 	224 224 225 225 225 225 225 225 225 225
 A.2.12. Changes in MySQL Workbench 5.1.8 (Not yet released) A.2.13. Changes in MySQL Workbench 5.1.7 (Not yet released) A.2.14. Changes in MySQL Workbench 5.1.6 (Not yet released) A.2.15. Changes in MySQL Workbench 5.1.4 (Not yet released) A.3.1. Changes in MySQL Workbench 5.0.30 (18 February 2009) A.3.2. Changes in MySQL Workbench 5.0.29 (12 December 2008) A.3.3. Changes in MySQL Workbench 5.0.28 (06 December 2008) A.3.4. Changes in MySQL Workbench 5.0.27 (07 November 2008) A.3.5. Changes in MySQL Workbench 5.0.26 (16 October 2008) A.3.6. Changes in MySQL Workbench 5.0.26 (16 October 2008) A.3.7. Changes in MySQL Workbench 5.0.23 (25 Iune 2008) A.3.8. Changes in MySQL Workbench 5.0.24 (12 August 2008) A.3.9. Changes in MySQL Workbench 5.0.21 (27 May 2008) A.3.10. Changes in MySQL Workbench 5.0.21 (27 April 2008) A.3.11. Changes in MySQL Workbench 5.0.21 (26 April 2008) A.3.12. Changes in MySQL Workbench 5.0.21 (27 April 2008) A.3.13. Changes in MySQL Workbench 5.0.21 (27 April 2008) A.3.14. Changes in MySQL Workbench 5.0.21 (27 April 2008) A.3.13. Changes in MySQL Workbench 5.0.21 (27 April 2008) A.3.14. Changes in MySQL Workbench 5.0.17rc (07 April 2008) 	224 224 225 225 225 225 225 225 225 225
 A.2.12. Changes in MySQL Workbench 5.1.8 (Not yet released) A.2.13. Changes in MySQL Workbench 5.1.7 (Not yet released) A.2.14. Changes in MySQL Workbench 5.1.6 (Not yet released) A.2.15. Changes in MySQL Workbench 5.1.4 (Not yet released) A.3.10. Changes in MySQL Workbench 5.0.30 (18 February 2009) A.3.2. Changes in MySQL Workbench 5.0.29 (12 December 2008) A.3.3. Changes in MySQL Workbench 5.0.29 (06 December 2008) A.3.4. Changes in MySQL Workbench 5.0.27 (07 November 2008) A.3.5. Changes in MySQL Workbench 5.0.26 (16 October 2008) A.3.6. Changes in MySQL Workbench 5.0.26 (12 September 2008) A.3.7. Changes in MySQL Workbench 5.0.26 (12 September 2008) A.3.8. Changes in MySQL Workbench 5.0.22 (27 May 2008) A.3.9. Changes in MySQL Workbench 5.0.21 (27 April 2008) A.3.10. Changes in MySQL Workbench 5.0.21 (27 April 2008) A.3.11. Changes in MySQL Workbench 5.0.20 (26 April 2008) A.3.12. Changes in MySQL Workbench 5.0.18rc (not released) A.3.13. Changes in MySQL Workbench 5.0.19 (15 April 2008) A.3.14. Changes in MySQL Workbench 5.0.17rc (07 April 2008) A.3.15. Changes in MySQL Workbench 5.0.16rc (26 March 2008) 	224 224 225 225 225 225 225 226 227 227 227 227 228 228 229 230 232 233 233 233 233 234 235
 A.2.12. Changes in MySQL Workbench 5.1.8 (Not yet released) A.2.13. Changes in MySQL Workbench 5.1.7 (Not yet released) A.2.14. Changes in MySQL Workbench 5.1.6 (Not yet released) A.2.15. Changes in MySQL Workbench 5.1.4 (Not yet released) A.3.1. Changes in MySQL Workbench 5.0.20 (18 February 2009) A.3.2. Changes in MySQL Workbench 5.0.29 (12 December 2008) A.3.3. Changes in MySQL Workbench 5.0.28 (06 December 2008) A.3.4. Changes in MySQL Workbench 5.0.27 (07 November 2008) A.3.5. Changes in MySQL Workbench 5.0.26 (16 October 2008) A.3.6. Changes in MySQL Workbench 5.0.26 (12 September 2008) A.3.7. Changes in MySQL Workbench 5.0.23 (25 June 2008) A.3.8. Changes in MySQL Workbench 5.0.23 (25 June 2008) A.3.9. Changes in MySQL Workbench 5.0.21 (27 May 2008) A.3.10. Changes in MySQL Workbench 5.0.21 (27 April 2008) A.3.11. Changes in MySQL Workbench 5.0.20 (26 April 2008) A.3.12. Changes in MySQL Workbench 5.0.20 (26 April 2008) A.3.13. Changes in MySQL Workbench 5.0.20 (26 April 2008) A.3.14. Changes in MySQL Workbench 5.0.18rc (not released) A.3.15. Changes in MySQL Workbench 5.0.18rc (07 April 2008) A.3.14. Changes in MySQL Workbench 5.0.18rc (07 April 2008) A.3.15. Changes in MySQL Workbench 5.0.18rc (07 April 2008) A.3.16. Changes in MySQL Workbench 5.0.18rc (17 April 2008) A.3.16. Changes in MySQL Workbench 5.0.18rc (17 April 2008) A.3.16. Changes in MySQL Workbench 5.0.18rc (17 April 2008) A.3.16. Changes in MySQL Workbench 5.0.18rc (17 April 2008) A.3.16. Changes in MySQL Workbench 5.0.18rc (17 April 2008) A.3.16. Changes in MySQL Workbench 5.0.18rc (17 April 2008) A.3.16. Changes in MySQL Workbench 5.0.18rc (17 April 2008) A.3.16. Changes in MySQL Workbench 5.0.18rc (17 April 2008) 	224 224 225 225 225 225 225 226 227 227 228 228 229 230 232 233 233 233 233 234 235 236
 A.2.12. Changes in MySQL Workbench 5.1.8 (Not yet released) A.2.13. Changes in MySQL Workbench 5.1.7 (Not yet released) A.2.14. Changes in MySQL Workbench 5.1.6 (Not yet released) A.2.15. Changes in MySQL Workbench 5.1.4 (Not yet released) A.3.1. Changes in MySQL Workbench 5.0.30 (18 February 2009) A.3.2. Changes in MySQL Workbench 5.0.29 (12 December 2008) A.3.3. Changes in MySQL Workbench 5.0.28 (06 December 2008) A.3.4. Changes in MySQL Workbench 5.0.27 (07 November 2008) A.3.5. Changes in MySQL Workbench 5.0.26 (16 October 2008) A.3.6. Changes in MySQL Workbench 5.0.26 (16 October 2008) A.3.7. Changes in MySQL Workbench 5.0.26 (12 September 2008) A.3.8. Changes in MySQL Workbench 5.0.26 (12 September 2008) A.3.9. Changes in MySQL Workbench 5.0.21 (27 May 2008) A.3.9. Changes in MySQL Workbench 5.0.21 (27 May 2008) A.3.9. Changes in MySQL Workbench 5.0.22 (27 May 2008) A.3.10. Changes in MySQL Workbench 5.0.20 (26 April 2008) A.3.11. Changes in MySQL Workbench 5.0.21 (27 April 2008) A.3.12. Changes in MySQL Workbench 5.0.21 (27 April 2008) A.3.13. Changes in MySQL Workbench 5.0.21 (27 April 2008) A.3.14. Changes in MySQL Workbench 5.0.19 (15 April 2008) A.3.15. Changes in MySQL Workbench 5.0.17rc (07 April 2008) A.3.16. Changes in MySQL Workbench 5.0.17rc (17 April 2008) A.3.16. Changes in MySQL Workbench 5.0.15rc (17 March 2008) A.3.16. Changes in MySQL Workbench 5.0.15rc (17 March 2008) A.3.17. Changes in MySQL Workbench 5.0.15rc (17 March 2008) A.3.17. Changes in MySQL Workbench 5.0.14abeta (28 February 2008) 	224 224 225 225 225 225 225 226 227 227 227 228 228 229 230 232 233 233 233 233 233 233 234 235 236 238
 A.2.12. Changes in MySQL Workbench 5.1.8 (Not yet released) A.2.13. Changes in MySQL Workbench 5.1.7 (Not yet released) A.2.14. Changes in MySQL Workbench 5.1.6 (Not yet released) A.2.15. Changes in MySQL Workbench 5.1.4 (Not yet released) A.3.1. Changes in MySQL Workbench 5.0.30 (18 February 2009) A.3.2. Changes in MySQL Workbench 5.0.29 (12 December 2008) A.3.3. Changes in MySQL Workbench 5.0.28 (06 December 2008) A.3.4. Changes in MySQL Workbench 5.0.27 (07 November 2008) A.3.5. Changes in MySQL Workbench 5.0.26 (16 October 2008) A.3.6. Changes in MySQL Workbench 5.0.26 (16 October 2008) A.3.7. Changes in MySQL Workbench 5.0.23 (25 June 2008) A.3.8. Changes in MySQL Workbench 5.0.23 (25 June 2008) A.3.9. Changes in MySQL Workbench 5.0.21 (27 May 2008) A.3.10. Changes in MySQL Workbench 5.0.20 (26 April 2008) A.3.11. Changes in MySQL Workbench 5.0.20 (26 April 2008) A.3.12. Changes in MySQL Workbench 5.0.18rc (not released) A.3.13. Changes in MySQL Workbench 5.0.14rc (17 April 2008) A.3.14. Changes in MySQL Workbench 5.0.14rc (17 April 2008) A.3.15. Changes in MySQL Workbench 5.0.14rc (17 April 2008) A.3.16. Changes in MySQL Workbench 5.0.14rc (17 April 2008) A.3.17. Changes in MySQL Workbench 5.0.14rc (17 April 2008) A.3.16. Changes in MySQL Workbench 5.0.14rc (17 April 2008) A.3.16. Changes in MySQL Workbench 5.0.14rc (17 March 2008) A.3.16. Changes in MySQL Workbench 5.0.14rc (17 March 2008) A.3.17. Changes in MySQL Workbench 5.0.14re (28 February 2008) A.3.18. Changes in MySQL Workbench 5.0.14re (25 February 2008) 	224 224 225 225 225 225 225 226 227 228 227 228 229 230 232 233 233 233 233 233 233 233 233

B.1. NET Flat TabControl License	41
B.2. Boost Library License	41
B.3. cURL (libcurl) License	41
B 4. Python License 24	42

List of Figures

5.1. MySQL Workbench Mac OS A Instantion Screen	7
4.1. Getting Started Tutorial - Home Screen	9
4.2. Getting Started Tutorial - Specify Host Machine	9
4.3. Getting Started Tutorial - Database Connection	10
4.4. Getting Started Tutorial - Connection Test	11
4.5. Getting Started Tutorial - Operating System	12
4.6 Getting Started Tutorial - Test Host Settings	
4.7 Getting Started Tutorial - Review Settings	14
4.8 Catting Started Tutorial - Instance Name	15
4.9 Getting Started Tutorial - Home Screen Instance	16
4.10 Certing Started Tutorial Admin Storetun	10
4.10. Getting Stated Tutorial Admin Statup	1/
4.11. Getting statied Tutorial - none Screen	10
4.12. Getting Started Tutorial - New Schema	19
4.13. Getting Started Tutorial - Columns	20
4.14. Getting Started Tutorial - EER Diagram	21
4.15. Getting Started Tutorial - Manage Connections	22
4.16. Getting Started Tutorial - Review Script	23
4.17. Getting Started Tutorial - Edit Table Data	24
4.18. Getting Started Tutorial - Edit Data	25
4.19. Getting Started Tutorial - Results	26
5.1. The Home screen	28
5.2. The Preferences Dialog Box	29
6.1 Manage DB Connections - Dialog	36
6.2 Manage DB Connections - Advanced Tab	
6.2 Manage DB Connections - Advanced Tab	
6.4 Manage DB Connections - SOLU Desmantance	
0.4. Manage DD Connections - SSH Parameters	39
0.5 SQL Editor	40
6.6. SQL Editor - Toolbar	41
6.7. SQL Editor - SQL Query Panel	42
6.8. SQL Editor - Main Tabsheets	43
6.9. SQL Editor - Snippets Palette	43
6.10. SQL Editor - Results Tabsheets	44
6.11. SQL Editor - Results Tabsheets Navigation Controls	44
6.12. SQL Editor - Live Editing Tabsheet Navigation Controls	45
6.13. SQL Editor - Connection Information Palette	46
6.14. SOL Editor - Default Schema Listbox	46
6.15. SOL Editor - Schemata Explorer	46
71 The MySQL Model page	50
	50
7.2 The Eind Window	- - /
7.2. The Find Window	52
7.2. The Find Window	52
7.2. The Find Window	52 60 61
 7.2. The Find Window	
7.2. The Find Window	52 60 61 62 64
 7.2. The Find Window	52 60 61 62 64 68
7.2. The Find Window 7.3. Roles and Privileges 7.4. Role Editor 7.5. The Model Navigator Palette 7.6. The Vertical Toolbar 7.7. A Table on an EER Diagram 7.8. The Table Editor	
 7.2 The Find Window	52 60 61 62 64 68 69 70
 7.2 The Find Window	52 60 61 62 64 68 69 70 71
 7.2. The Find Window	52 60 61 62 64 68 69 70 71 76
 7.2. The Find Window	52 60 61 62 64 68 69 70 71 76 77
 7.2. The Find Window	52 60 61 62 64 68 69 70 71 76 77 83
 7.2. The Find Window	52 60 61 62 64 68 69 70 71 76 77 83 87
7.2. The Find Window 7.3. Roles and Privileges 7.4. Role Editor 7.5. The Model Navigator Palette 7.6. The Vertical Toolbar 7.7. A Table on an EER Diagram 7.8. The Table Editor 7.9. The Columns Tab 7.10. The Indexes Tab 7.11. The Relationship Connector 7.12. The Split Connector 7.13. The Layer Object 7.14. Reverse Engineer Database Wizard 7.15. Connect to DBMS	52 60 61 62 64 68 69 70 71 76 77 83 87 88
7.2. The Find Window 7.3. Roles and Privileges 7.4. Role Editor 7.5. The Model Navigator Palette 7.6. The Vertical Toolbar 7.7. A Table on an EER Diagram 7.8. The Table Editor 7.9. The Columns Tab 7.10. The Indexes Tab 7.11. The Relationship Connector 7.12. The Split Connector 7.13. The Layer Object 7.14. Reverse Engineer Database Wizard 7.15. Connect to DBMS 7.16. Select Schemata	52 60 61 62 64 68 69 70 71 76 77 83 87 88 88
 7.2 The Find Window	52 60 61 62 64 68 69 70 71 76 77 83 87 88 88 88
7.2 The Find Window 7.3 Roles and Privileges 7.4 Role Editor 7.5 The Model Navigator Palette 7.6 The Vertical Toolbar 7.7 A Table on an EER Diagram 7.8 The Table Editor 7.9 The Columns Tab 7.10 The Indexes Tab 7.11 The Relationship Connector 7.12 The Split Connector 7.13 The Layer Object 7.14 Reverse Engineer Database Wizard 7.15 Connect to DBMS 7.16 Select Schemata 7.17. Fetch Object Info 7.18 Select Objects	52 60 61 62 64 68 69 70 71 76 77 83 87 88 88 88 89
 7.2 The Find Window 7.3 Roles and Privileges 7.4 Role Editor 7.5 The Model Navigator Palette 7.6 The Vertical Toolbar 7.7 A Table on an EER Diagram 7.8 The Table Editor 7.9 The Columns Tab 7.10 The Indexes Tab 7.11 The Relationship Connector 7.12 The Split Connector 7.13 The Layer Object 7.14. Reverse Engineer Database Wizard 7.15 Connect to DBMS 7.16 Select Schemata 7.17 Fetch Object Info 7.18 Select Objects 7.10 Show Eiter 	52 60 61 62 64 68 70 71 76 77 83 87 88 88 89 90
 7.2. The Find Window 7.3. Roles and Privileges 7.4. Role Editor 7.5. The Model Navigator Palette 7.6. The Vertical Toolbar 7.7. A Table on an EER Diagram 7.8. The Table Editor 7.9. The Columns Tab 7.10. The Indexes Tab 7.11. The Relationship Connector 7.12. The Split Connector 7.13. The Layer Object 7.14. Reverse Engineer Database Wizard 7.15. Connect to DBMS 7.16. Select Schemata 7.17. Fetch Object Info 7.18. Select Objects 7.19. Show Filter 7.20. Proceese 	
7.2. The Find Window. 7.3. Roles and Privileges 7.4. Role Editor 7.5. The Model Navigator Palette 7.6. The Vertical Toolbar 7.7. A Table on an EER Diagram 7.8. The Table Editor 7.9. The Columns Tab 7.10. The Indexes Tab 7.10. The Indexes Tab 7.11. The Relationship Connector 7.12. The Split Connector 7.13. The Layer Object 7.14. Reverse Engineer Database Wizard 7.15. Connect to DBMS 7.16. Select Schemata 7.17. Fetch Object Info 7.18. Select Objects 7.19. Show Filter 7.20. Progress 7.21. Desetter 7.21.	
7.2. The Find Window 7.3. Roles and Privileges 7.4. Role Editor 7.5. The Model Navigator Palette 7.6. The Vertical Toolbar 7.7. A Table on an EER Diagram 7.8. The Table Editor 7.9. The Columns Tab 7.10. The Indexes Tab 7.11. The Relationship Connector 7.12. The Split Connector 7.13. The Layer Object 7.14. Reverse Engineer Database Wizard 7.15. Connect to DBMS 7.16. Select Schemata 7.17. Fetch Object Info 7.18. Select Objects 7.19. Show Filter 7.20. Progress 7.21. Results	

7.23. SQL Export Options	96
7.24. SQL Object Export Filter	97
7.25. Review Generated Script	97
7.26. Options	98
7.27. Script	99
7.28. Catalog Validation	. 100
7.29. Options	. 101
7.30. Select Objects to Forward Engineer	. 102
7.31. Review Script	. 103
7.32. Set parameters for connecting to a DBMS	. 104
7.33. Set parameters for connecting to a DBMS	. 105
7.34. Model and Database Differences	. 106
7.35. Controlling Synchronization Direction	. 107
7.36. Update Model Button	. 108
7.37. Ignore Button	. 109
7.38. Update Source Button	. 110
7.39. Click arrows to change direction of synchronization	. 111
7.40. Catalog Sources	. 112
7.41. Catalog Diff Report	. 113
7.42. Adding Tables to the Canvas	. 115
7.43. The Default Schema	. 116
7.44. Adding an EER Diagram	. 117
7.45. The sakila EER Diagram	. 118
8.1. Manage Server Instances Dialog	. 133
8.2. Manage Server Instances Dialog	. 134
8.3. MySQL Workbench - Admin page	. 136
8.4. Administrator - Startup tab	. 137
8.5. Administrator - Configuration tab	. 138
8.6. Administrator - Accounts tab	. 139
8.7. Administrator - Connections tab	. 140
8.8. Administrator - Variables tab	. 141
8.9. Administrator - Export to Disk	. 141
8.10. Administrator - Import from Disk	. 142
8.11. Administrator - Advanced Options	. 143
8.12. Administrator - Logs tab	. 144
9.1. The Workbench Scripting Shell	. 149

Chapter 1. MySQL Workbench Introduction

MySQL Workbench provides a graphical tool for working with MySQL Servers and databases. MySQL Workbench fully supports MySQL Server versions 5.1 and above. It is also compatible with MySQL Server 5.0, but not every feature of 5.0 may be supported. It does not support MySQL Server versions 4.x.

MySQL Workbench provides three main areas of functionality:

- SQL Development
- Data Modeling
- Server Administration

This section provides a brief overview of this functionality.

- 1. **SQL Development**: Enables you to create and manage connections to database servers. As well as allowing you 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.
- Data Modeling: Enables 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.
- 3. Server Administration: Enables you to create and administer server instances. This functionality replaces that previously provided by the MySQL Administrator stand-alone application.

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 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:

- MySQL Specific Schema Validation
- Model Validation
- General Schema Validation
- DBDoc

DBDoc provides the following features:

- Document complex database schemata
- Document all SQL object types
- Document output available in different file formats

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 <u>HELP</u> menu and choose the <u>SYSTEM INFO</u> 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.

In addition to platform-specific command line options, MySQL Workbench has the following command line options:

- --admin instance Launch MySQL Workbench and load the server instance specified.
- --query connection Launch MySQL Workbench and load the connection specified.
- --model modelfile Launch MySQL Workbench and load the model specified.
- --script script Launch MySQL Workbench and run the script specified.
- --run code Launch MySQL Workbench and run the code snippet specified.
- --quit-when-done quits MySQL Workbench after --script or --run finishes.

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 <u>REPORT A</u> <u>BUG</u> option under the <u>HELP</u> menu.

1. To install MySQL Workbench, right-click 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 you 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 <u>-help</u> and <u>-version</u>, 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.

Rmoving 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-lfcl0.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 <u>APPLICATIONS</u>, <u>PROGRAMMING</u>, <u>MYSQL WORKBENCH</u> 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 mysql-workbench-oss-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.

MySQL Workbench

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 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.

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 <u>MOVE TO TRASH</u>. 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.

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. This tutorial demonstrates the procedures on Microsoft Windows, they are, however, the same for all supported platforms.

4.1. Administering a MySQL Server

In this section you will see how you can use MySQL Workbench to connect to a server in order to carry out administrative functions, such as starting and stopping the server.

1. Launch MySQL Workbench. You will be presented with the Home screen:

Figure 4.1. Getting Started Tutorial - Home Screen

S MySQL Workbench		
File Edit View Database Plugins Scripting Co	mmunity Help	
Home ×		
Workbench Central		•
Welcome to MySQL W • What's New in This Release? Read about all changes in this MySQL	Vorkbench Workbench release.	WySQL Bug Reporter Workbench Team Blog
Workspace		
SQL Development Connect to existing databases and run SQL Queries, SQL scripts, edit data and manage database objects.	Create and manage models, forward & reverse engineer, compare and synchronize schemas, report.	Server Administration Configure your database server, setup user accounts, browse status variables and server logs.
Open Connection to Start Querying Or click a DB connection to open the SQL Editor.	Open Existing EER Model Or select a model to open or click here to browse.	Server Administration Or click to manage a database server instance.
New Connection Add a new database connection for querying.		Register a new server instance to manage.
Edit Table Data Select a connection and schema table to edit.	Create New EER Model Create a new EER Model from scratch.	Create a dump file or restore data from a file.
Edit SQL Script Open an existing SQL Script file for editing.	Create EER Model From Existing Database Create by connecting and reverse engineering.	Manage Security Manage user accounts and assign privileges.
Manage Connections Modify connection settings or add connections.	Create EER Model From SQL Script Import an existing SQL file.	Add, delete and update server instance settings.
		B ,

- In order to administer your MySQL Server you need to first create a Server Instance. This contains information about the target server, including how to connect to it. From the Home screen of MySQL Workbench, click NEW SERVER INSTANCE. The CREATE NEW SERVER INSTANCE PROFILE wizard will be displayed.
- 3. In this tutorial we will connect to a locally installed server, so click NEXT.

Figure 4.2. Getting Started Tutorial - Specify Host Machine

Create New Server Instance Profile		
Specify Host Machine	Specify the Host Machine the Database Server is running on	
Database Connection		
Test DB Connection	This wizard will guide you to create of a Server Profile to manage a MySQL server.	
Host SSH Connection	To fully support management of a remote MySQL server, a SSH daemon must be running in the target machine. The SSH login is used to start, stop and configure MySQL.	
Operating System	You may create a Profile without SSH if you do not need that functionality.	
Test Settings		
Review Parameters	If your database server is running on the same machine as this application select localhost. Otherwise please specify the TCP/IP address or the network name of the remote machine. You may also pick an	
MySQL Config File	existing database connection.	
Specify Commands		
Complete Setup	Kemole nosi	
	Address:	
	Take Parameters from Existing Database Connection	
94/11	Back Next Cancel)
		1

4. Next you will set up a connection, or select an existing connection to use to connect to the server. Assuming you have not already created a connection, you can use the default values here, although if your MySQL Server has a password set for root, you can set it here by clicking on Store in Vault. This allows you to connect to the server without needing to enter a password each time. It is also possible to use another account to connect to the server by setting the username and password here, if required.

Figure 4.3. Getting Started Tutorial - Database Connection

Create New Server Instance Profile			E
Specify Host Machine	Set the Databas	e Connection values	
Database Connection			
Test DB Connection	Connection Name:	localhost	Type a name for the connection
Host SSH Connection	Connection Method:	Standard (TCP/IP)	 Method to use to connect to the RDBMS
Operating System	Parameters Advan	nced	
Test Settings	Hostname:	localhost Port: 3306	Name or IP address of the server host - TCP
Review Parameters			
MySQL Config File	Username:	root	Name of the user to connect with.
Specify Commands	Password:	Store in Vault Clear	The user's password.
Complete Setup	Default Schema:		The schema that will be used as default sche
99111			
1919			Back Next Cancel

You can now click NEXT.

5. The connection will now be tested. You should see that the connection was successful. If not click BACK and check that you have entered the information required.

Figure 4.4. Getting Started Tutorial - Connection Test

Create New Server Instance Profile		
Specify Host Machine	Testing the Database Connection	
Database Connection		
Test DB Connection	The database connection information is being tested. This might take a few moments depending on your network	
Host SSH Connection	connection.	
Operating System		
Test Settings		
	♂ Get Server OS: Windows	
Review Parameters	Database connection tested successfully.	
MySQL Config File		
Specify Commands		
Complete Setup		
24/11	Show Logs Back Next Cancel]

If everything tested correctly, click NEXT.

6. On this screen you will set the operating system and installation type. In this case the installation is Microsoft Windows, and the installation type is MySQL 5.1 x86 Installer Package. Setting these options allows MySQL Workbench to determine location of configuration files, and the correct start up and shut down commands to use for the server.

Figure 4.5. Getting Started Tutorial - Operating System

Create New Server Instance Profile		
Specify Host Machine	Specify the operating	system of the machine
Database Connection		
Test DB Connection		
Host SSH Connection	Please select the operating If you are unsure about th	system that is running on the target machine and the type of database installation. e type of database installation select the (Vendor Package) variant. If your specific
Operating System	operating system is not in	his list, select a related variant.
Test Settings	Operating System:	Windows
Review Parameters	MySQL Installation Type:	Windows (MySQL 5.1 x86 Installer Package) 🗸
MySQL Config File		
Specify Commands		
Complete Setup		
<i>¶ </i>		
		Back Next Cancel

Once you have set the operating system and installation type, click NEXT.

7. The wizard will now check that it is able to access the start up and shut down commands, and access the MySQL Server configuration file.

Figure 4.6. Getting Started Tutorial - Test Host Settings

Create New Server Instance Profile		
Specify Host Machine	Testing Host Machine Settings	
Database Connection		
Test DB Connection	The connection to the host machine is being tested. This might take a few moments depending on your network	
Host SSH Connection	connection.	
Operating System	 Connect to host machine 	
Test Settings	Check location of start/stop commands	
Daview Darameters		
	Testing host machine settings is done.	
MySQL Config File		
Specify Commands		
Complete Setup		
99111		_
	Show Logs Back Next Cancel	

Check that everything is in order and then click NEXT.

8. You now have a chance to review the configuration settings so far. The information displayed varies slightly depending on platform, connection method and installation type:

Figure 4.7. Getting Started Tutorial - Review Settings



Click NEXT.

9. Finally you can give the server instance a suitable name. This will be used to select this particular instance from a list of available instances.

Figure 4.8. Getting Started Tutorial - Instance Name

Create New Server Instance Profile		
Specify Host Machine	Create the Instance	e Profile
Database Connection		
Test DB Connection	Please enter a name for	this server instance and click Next. A new Server Instance Profile entry will be created for
Host SSH Connection	managing this MySQL ser	ver.
Operating System	Server Instance Name:	mysqld@localhost
Test Settings		
Review Parameters		
MySQL Config File		
Specify Commands		
Complete Setup		
94711		Rack Finish Canad

Having set the desired name, you can click FINISH to complete the server instance creation process.

10. You will now be returned to the Home screen. You will see the new server instance you created, along with the new connection you created as part of the above procedure.

Figure 4.9. Getting Started Tutorial - Home Screen Instance

MySQL Workbench				
File Edit View Database Plugins Scripting Cor	nmunity Help			
Home ×				
Workbench Central				•
Welcome to MySQL Welcom		MySQL Doc Library	Workbench Toam Planet MySQL	Workbench
Workbench		Library Reporter	reambiog	Torums
Workspace				
SQL Development Connect to existing databases and run SQL Queries, SQL scripts, edit data and manage database objects.	Create and manage mode reverse engineer, compar synchronize schemas, rep	ls, forward & e and hort.	Server Admini Configure your database user accounts, browse st and server logs.	stration server, setup atus variables
Open Connection to Start Querying Or click a DB connection to open the SQL Editor.	Open Existing EER Model Or select a model to open or click here	to browse.	erver Administration click to manage a database se	rver instance.
User: root Host: localhost:3306			mysqld@localhost Local Type: Windows	
New Connection Add a new database connection for querying.		Re	ew Server Instance gister a new server instance t	o manage.
Edit Table Data Select a connection and schema table to edit.	Create New EER Model Create a new EER Model from scratch.	. M	anage Import / Export eate a dump file or restore data	from a file.
Edit SQL Script	Create EER Model From Existin	g Database Mineering.	anage Security mage user accounts and assig	n privileges.
Manage Connections Modify connection settings or add connections.	Create EER Model From SQL So Import an existing SQL file.	cript M	anage Server Instances Id, delete and update server ins	stance settings.
New server instance created.				8

You are now ready to test your new server instance.

11. From the Home screen, double-click the Server Instance you created. The Administrator will open on the STARTUP configuration page.

Figure 4.10. Getting Started Tutorial - Admin Startup

MySQL Workbench	
File Edit View Database Plugins Scripting Community Help	
Home Admin (mysqld@localhost) ×	
Server Status	•
INFO SYSTEM SERVER HEALTH	
Name: mysql@localhost Host: localhost Server: 5.1.40-community Status: Running CPU: 7% Mem: 38% Connection Usage: 3 Traffic: 7.16 KB/s	Query Cache Hitrate: 0.00% Key Efficiency: 0.00%
Configuration	
Startup Start/Stop Server Configuration File Accounts Manage Users Connections Edit Connection List Variables Start/Stop Server	Vars Data Dump Logs Server Log Files
Database Server Status Startup Message Log	
The database server is started and ready for client connections. To shut the Server 2010-09-14 16:03:19 - Ch down, use the "Stop Server" button 2010-09-14 16:03:20 - Ser	ecked server status: Server is running.
The database server instance is running Stop Server	
If you stop the server, you and your applications will not be able to use the Database and all current connections will be closed	
	-
Refresh Status	Clear Messages Copy to Clipboard
WB Admin Opened	 目 ;

- 12. Click the STOP SERVER button. The message window will show that the server has stopped.
- 13. Click the START SERVER button to resume the server. The message window will confirm that the server is running.

You have now seen how to create a server instance to allow you to manage a MySQL server.

For further information see Chapter 8, Server Administration.

4.2. 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 4.11. Getting Started Tutorial - Home Screen

MySQL Workbench			- • •
File Edit View Model Database Plugins Scrip	ting Community Help		
Home ×			
Workbench Central			
Welcome to MvSQL W	/orkbench		Check for Updates
* What's New in This Release?	Dev-Central Blogs		Submit a Bug Report
Workbench Read about all changes in this MySQL Morkbeuch Morkbeuch	Workbench release. Keep up to date with the main	product and its extensions.	Discuss a Topic
Workspace			
SQL Development Connect to existing databases and run SQL Queries, SQL scripts, edit data and manage database objects.	Create and manage models, forward & reverse engineer, compare and synchronize schemas, report.	Server Ac Configure your d user accounts, bi and server logs.	Iministration latabase server, setup rowse status variables
Popen Connection to start Querying Or click a DB connection to open the SQL Editor.	Open Existing EER Model Or select a model to open or click here to browse.	Server Administrati	on abase server instance.
world User: root Host: 127.0.0.1:3306	Home_Media.mwb Last modified Tue Mar 23 10:06:11 2 mytest.mwb Last modified Tue Mar 16 14:01:31 2	Korld Local Type: Wi	ndows
New Connection Add a new database connection for querying.	world.mwb	New Server Instance Register a new server in	e istance to manage.
Edit Table Data Select a connection and schema table to edit.	Create new EER Model Create a new EER Model from scratch.	Manage Import / E Create a dump file or res	xport tore data from a file.
Edit SQL Script	Create EER Model from Existing Database	Manage Security Manage user accounts a	and assign privileges.
Manage Connections Modify connection settings or add connections.	Create EER Model from SQL Script Import an existing SQL file.	Add, delete and update s	t ances server instance settings.
Ready			

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 4.12. Getting Started Tutorial - New Schema

NySQL Workbench		
File Edit View Model Dat	abase Plugins Scripting Community Help	
		N
Home MySQL Model ×		
Description Editor	Model Overview	
No Selection 👻		Î
	Add Diagram Physical Schemata	··· ··· = + -
	mydb MySQL-Schema	
	Tables (0 items)	
Description	16 aug (0, itawa)	Ŧ
User Types List		
	Name: dvd_collection C The name of the schema. It is recommended to use only alp characters. Spaces should be avoided and be replaced by _	ha-numeric
Name Definition Flags ^	Collation: Server Default Specifies which charset/collations the schema's tables will use have an explicit setting. Common choices are Latin or UTF8	e if they do not I.
BOOLE TINYINT(1) FIXED DECIMAL(10 =	Comments:	*
FLOAT4 FLOAT FLOAT8 DOUBLE INT1 TINYINT(4) INT2 SMALLINT(6) INT3 MEDIUMINT INT4 INT(11)		
INT8 BIGINT(20)	Scham .	-
Schema 'new_schema1' created.		

- 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 data type 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 4.13. Getting Started Tutorial - Columns

MySQL Workbench			
File Edit View Model Data	abase Plugins Scripting Community Help		
🖻 🖉 🖨 🕾 🕾 🔁 🕲 🛄		🔒	>_
Home MySQL Model ×			
Description Editor	Model Overview		
table 1: MySQL Table	Add Diagram Physical Schemata		▲ ₩ ₩ ₩ ≡ + -
	MySQL Schema dvd_collection		
	Tables (1 item)		-
Description User Types List	movies ×		
Name Definition Flags BOOL TINYINT(1) BOOLE TINYINT(1) FIXED DECIMAL(10 FLOAT4 FLOAT FLOAT4 FLOAT FLOAT8 DOUBLE INT1 TINYINT(4) INT2 SMALLINT(6) INT3 MEDIUMINT INT4 INT(11) INT8 BIGINT(20) INTE5 INT(11)	Column Name Datatype	PK NN UQ BIN UN ZF AI Default	Collation: Table Default Comments:
User Types History	Table Columns Indexes Foreign Keys	Triggers Partitioning Options Inserts	Privileges
Table 'table1' created in schema 'dvd_o	collection'		.:

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

Figure 4.14. Getting Started Tutorial - EER Diagram

MySQL Workbench			- • •
File Edit View Arrange M	odel Database Plugins Scripting Community Help	R	
Home MySQL Model E	ER Diagram X		
Bird's Eye	Diagram		
Zoom: 100% • 🔍 🔍	Image: Second state state Image: Second state state Image: Second state Image: Second state Image: Second state Image: Second state		E
Catalog Tree			
Description Properties	۲ III III III III III III III III III I		Þ
Ready			

- 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 <u>DATABASE</u>, <u>MANAGE CONNECTIONS...</u>.
- 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 4.15. Getting Started Tutorial - Manage Connections

		r			
tored Connections	Connection Name:	Big Iron Server			Type a name for the connection
world	Connection Method:	Standard (TCP/IP)		-	Method to use to connect to the RDBM
Big Iron Server	Parameters Advan	ced			
	Hostname:	127.0.0.1	Port: 3306	Name or IP add	dress of the server host - TCP/IP port
	Username:	root		Name of the us	ser to connect with.
	Password:	Store in Vault Clear	-	The user's pass	sword.
	Default Schema:			The schema the	at will be used as default schema

- 16. You are now ready to forward engineer your model to the live server. From the main menu select <u>DATABASE</u>, <u>FORWARD</u> <u>ENGINEER...</u>. 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 validaton 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 4.16. Getting Started Tutorial - Review Script

Forward Engineer to Database		
Catalog Validation	Review the SQL Script to be Executed	
Options		
Select Objects		
Review SQL Script Connection Options Commit Progress	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'; CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET latin 1 COLLATE latin 1_swedish_ci ; CREATE SCHEMA IF NOT EXISTS `dvd_collection` ; 	*
	<pre>vacchar(F3)TENULL; release_date`DATENULL; PRIMARY KEY ('movie_id')) ENGINE = InnoDB; SET SQL_MODE=@OLD_SQL_MODE; SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS; SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;</pre>	
	Save to File Copy to Clipboard Back Next Cancel	-

- 21. Select the connection you created earlier, "Big Iron Server". Click EXECUTE. Check the messages for any erros, 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.

4.3. 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 4.17. Getting Started Tutorial - Edit Table Data

MySQL Workbench			
File Edit View Model Database Plugins Script	ing Community Help		
Home ×			
Workbench Central			T
Welcome to MySQL W	orkbench		Check for Updates
* What's New in This Release?	* Dev-Central Blogs		Submit a Bug Report
Workbench Read about all changes in this MySQL V Morkpeucp Morkpeucp	Norkbench release. Keep up to date with the main p	product and its extensions.	Discuss a Topic
Workspace			
SQL Development Connect to existing databases and run SQL Queries, SQL scripts, edit data and manage database objects.	Create and manage models, forward & reverse engineer, compare and synchronize schemas, report.	Server Ac Configure your d user accounts, bi and server logs.	Iministration latabase server, setup rowse status variables
Open Connection to start Querying Or click a DB connection to open the SQL Editor.	Open Existing EER Model Or select a model to open or click here to browse.	Server Administrati	on abase server instance.
User: root Host: 127.0.0.1:3306 Big Iron Server User: root User: root Host: 127.0.0.1:3306	Test_Model.mwb Image: Constraint of the standard standar	Korld Local Type: Wi	ndows
New Connection Add a new database connection for querying.	mytest.mwb	New Server Instance Register a new server in	e istance to manage.
Edit Table Data Select a connection and schema table to edit.	Create new EER Model Create a new EER Model from scratch.	Manage Import / E	xport tore data from a file.
Edit SQL Script	Create EER Model from Existing Database Create by connecting and reverse engineering.	Manage Security Manage user accounts a	and assign privileges.
Manage Connections Modify connection settings or add connections.	Create EER Model from SQL Script	Add, delete and update s	cances server instance settings.
Ready			.::

- 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 4.18. Getting Started Tutorial - Edit Data

NySQL Workbench					
File Edit View Model Quer	y Database	Plugins Scripting Comr	nunity Help		
🗈 🗗 🐺 🦻 🖗 🔕 🗸	& ک 🛇 🖢	🙆 🚸 🖨		<u> </u>	
Home SQL Editor (Big Iron Serv	ver) ×				
Object Prowner	SOL Statements				
Object Browser		T `dvd_collection` `movies	`.		
Default:	1 - 101	ava_conection . movies	1		
G dvd_collection					
Routines					
🕀 📴 mydb					
🕀 📴 world					
	0	Nutant Uliatary Calara	(1)		
	Overview C	Jutput History Snippe	ts movies (1)		
	K & & A	🏭 🏛 比 👯 🛃 🖦	🖦 🗇 🐐 🚸	Fetched 4 records	-
	movie_id	title	release_date		
	1	Gone with the Wind	1939-04-17		
	2	The Hound of the Baskervilles	1939-03-31		
	3	The Matrix	1999-06-11		
	4	Above the Law	1988-04-08		
	▶*	NULL	NULL		
Connection Information					
Name: Big Iron Server					
Host: 127.0.0.1:3306					
Version: 5.1.40-community					
user.					
÷				 	

- 8. Now you will check that the data really has been applied to the live server. Launch the MySQL Command Line Client. Enter SE-LECT * 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 4.19. Getting Started Tutorial - Results
File Edit View Model Query Database Plugins Scripting Community Help	
	>
Home SQL Editor (Big Iron Server) SQL Editor (Big Iron Server) ×	
Object Browser SQL Statements ×	
Default: USE dvd_collection; 2 SELECT * FROM movies; 3 SELECT * FROM movies; 3 Routines Provide Notes Provide Output Victor Seineste Provide Output	
Overview Output History Snippets Result (1)	
III I I III III III III III III III II	-
movie_id title release_date	
▶ 1 Gone with the Wind 1939-04-17	
2 The Hound of the Baskervilles 1939-03-31	
3 The Matrix 1999-06-11	
4 Above the Law 1988-04-08	
Connection Information Name: Big Iron Server Host: 127.0.0.1:3306 Server: MySQL Version: 5.1.40-community User:	

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 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:





5.1. Workbench Central

Workbench Central enables 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.

Workbench Central includes the following facilities:

- What's new: A list of bug fixes and changes.
- MySQL Doc Library: Built in documentation.
- MySQL Bug Reporter: Links to the MySQL bug system, where you can report bugs.
- MySQL Team Blog: Currently links to Workbench team blog.
- Workbench Team Blog: Links to Workbench team blog.
- MySQL Newsletter: Currently links to the MySQL Developer forum.

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 <u>PREFERENCES</u> menu sets MySQL Workbench defaults. Choosing the <u>PREFERENCES</u> menu item opens the following dialog box:

Figure 5.2. The Preferences Dialog Box

Norkbenc	h Preferen	ices								(x
General Adn	ninistrator	SQL Editor	Model	MySQL	Diagram	Appearance					
			Auto	matically	Reopen Pre	evious Model W	/hen Started				
			Place	Sidebar	on the Righ	it Side					
			Force	e use of s	oftware ba	sed rendering	for EER diagra	ams			
	Undo H	listory Size:	50	Allowe Note: operat	d values ar using high v tion.	re from 1 up. values (> 100)	will increase r	memory usag	ge and slow d	own	
Interactive	e GRT Shel	Language:	pyth 🖣	Select will wo	the langua rk regardle	ge to use in th ss of this settir	e interactive (ng. This optior	GRT shell. So n requires a l	ripts, module restart.	s and plugins	
									ОК	Cancel	

The following list describes the dialog box tabs:

- <u>GENERAL</u>: The delete and undo history options
- <u>ADMINISTRATOR</u>: Configuration for tools used by the Administrator functionality
- <u>SQL EDITOR</u>: Configuration of the SQL Editor
- <u>MODEL</u>: Default object names
- <u>MySQL</u>:
- **DIAGRAM**: EER diagram settings
- <u>APPEARANCE</u>: Change colors and fonts used by various Workbench components

A more detailed discussion of these options follows.

5.4.1. The General Tab

The General tab allows you to set the following options:

- Automatically Reopen Previous Model When Started check this if you want the model you previously worked on to be automatically reopened when you start MySQL Workbench.
- Place Sidebar on the Right Side by default the Sidebar is placed on the left-hand side of the MySQL Workbench application. Select this option to place it on the right-hand side.
- Force use of software rendering for EER diagrams MySQL Workbench will use OpenGL for rendering when available. However, due to faulty drivers, problems do occasionally occur. These issues can be resolved by selecting the software rendering option here.
- Undo History Size you can limit the size of the undo history here. Set this value to 0 to have an unlimited undo history.
- Auto-save model interval An open model that has not been saved will automatically be saved after this period. On loading a model file, MySQL Workbench will notify the user if the file was not previously saved correctly, due to a crash or power failure. MySQL Workbench can then attempt to recover the last auto-saved version. Note that for automatic recovery to be available for a new file, it will have to have been saved at least once by the user.
- Interactive GRT Shell Language you can select the language to be used in the GRT shell by chosing a language from the drop down listbox INTERACTIVE GRT SHELL LANGUAGE. Currently there is a choice between Lua and Python. Python is the recommended option.

5.4.2. The Administrator Tab

This section provides configuration options that affect the Administrator functionality in MySQL Workbench.

Enables you to set paths to the mysqldump tool, and the mysql tool. If these are left blank the defaults will be used. This panel also enables you to set the directory for export dump files.

5.4.3. The SQL Editor Tab

This section provides configuration options that affect the SQL Editor functionality in MySQL Workbench.

There are three main groups of parameters that can be set here:

- SQL properties
- Query Editor
- Query Results

SQL Properties

SQL properties that can be set include the SQL_MODE, case sensitivity of identifiers, and the SQL delimiter used (by default this is \$\$).

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.

Query Editor

The query editor properties that can be set include the following:

- SHOW LIVE SCHEMA OVERVIEW This option allows a simplification of the user interface by removing the Overview tab from the SQL Editor. This is also extremely useful if schemata have a large number of tables, or there is a large number of schemata in a model. In each of these cases load times would be greatly increased as the tables and schemata are enumerated and drawn.
- SHOW SCHEMA CONTENTS IN SCHEMA TREE enumerating, populating and drawing large numbers of items can significantly increase loading times. For this reason this facility can be switched off for models containing large numbers of schemata and tables.
- SHOW METADATA SCHEMATA by default metadata schemata are not displayed. If required to view, for example to troubleshoot or check metadata information, they can be displayed by selecting this option.
- CONTINUE ON SQL SCRIPT ERROR should an error occur while executing a script, this option will allow you to continue executing the remainder of the script.
- FORBID UPDATE AND DELETE STATEMENTS WITHOUT A WHERE CLAUSE this option enables the SQL_SAFE_UPDATES option for the sesson, preventing UPDATE and DELETE statements from being executed if a WHERE clause is not present. This can avoid potentially dangerous situations where a command could accidentally update or delete all rows in a table.
- MAX SYNTAX ERROR COUNT large complex scripts can contain many errors. Further, a syntax error early on can lead to many subsequent syntax errors. For these reasons it is possible to limit the number of errors displayed using this option, the default being 100 error messages.
- **PROGRESS STATUS UPDATE INTERVAL** When executing long running queries over a slow connection you would need to increase this value, to prevent excess load on the connection.
- **DBMS CONNECTION KEEP-ALIVE INTERVAL** When executing long running queries over a slow connection you would need to increase this value to prevent the connection being lost.

Query Results

- LIMIT ROWS queries can sometimes result in an excessive number of rows being returned as a result. This can heavily load the connection, and take time to display in MySQL Workbench. To prevent this you can set a more moderate value here.
- LIMIT ROWS COUNT specify the maximum number of result rows to return.
- MAX. FIELD VALUE LENGTH TO DISPLAY to avoid display problems due to excessive field length, it is possible to set the maximum field length to display (in bytes).
- TREAT BINARY/VARBINARY AS NON-BINARY CHARACTER STRING Binary byte string values are not displayed by default in the results grid, but are instead marked as BLOB values. These can then be viewed or edited with the BLOB editor. Non-binary character string values are displayed in the results grid, and can be edited in the grid cell or using the BLOB editor. Note that if this option is turned on, data truncation may result. This is because binary byte string values may contain null bytes as part of their valid data. For non-binary character strings a null byte terminates the string.
- ENABLE DATA CHANGES COMMIT WIZARD In the SQL Editor, when editing table data and then clicking the APPLYING CHANGES TO DATA button, a wizard is launched to step you through applying you changes. This gives you a chance to review the SQL that will be applied to the live server to make the requested changes. If the option is deselected, then the changes will simply be applied to the server, without the wizard being displayed, and without a chance to review the changes that will be made.

5.4.4. The Model Tab

This section provides configuration options that affect the Modeling functionality in MySQL Workbench.

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 <u>EDIT</u> 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 7.7.1.3.5, "The Foreign Keys Tab".

5.4.5. The MySQL Tab

This enables you to set the default table storage engine.

5.4.6. 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.

5.4.7. 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 7.5.2.1, "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.

Note

Note that on Windows, the default font for the editor only supports latin-1 characters. If you need to use characters not supported by the latin-1 character set, you will need to change the font here.

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

Each of these action items is described in the following sections.

6.1. Open Connection to start Querying

Clicking 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 more about the SQL Editor, see Section 6.7, "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.

To read more about the SQL Editor see Section 6.7, "SQL Editor".

6.2. New Connection

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

To read more about creating and managing connections see Section 6.6, "Manage DB Connections Dialog".

6.3. Edit Table Data

This action item enables you to edit table data. When clicked the Edit Table Data wizard is launched. This is a two stage wizard. The first stage enables you to select a Stored Connection. The second stage enables 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 enables you to interactively edit table data as required.

To read more about the SQL Editor see Section 6.7, "SQL Editor".

6.4. Edit SQL Script

Clicking this action item launched the Edit SQL Script wizard. This is a two stage wizard. The first stage enables you to select a Stored Connection. The second stage enables 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.

6.5. 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 6.6, "Manage DB Connections Dialog".

6.6. Manage DB Connections Dialog

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 <u>DATABASE</u>, <u>MANAGE CONNECTIONS</u> 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 enables you to create or delete connections:

Manage DB Connections			
Stored Connections world mytest Big Iron Server	Connection Name: Connection Method: Parameters Advan	world Standard (TCP/IP) ced	Type a name for the connection • Method to use to connect to the RDBMS
localhost	Hostname: Username:	127.0.0.1 Port: 3306	Name or IP address of the server host - TCP/IP port Name of the user to connect with.
	Password:	Store in Vault Clear	The user's password.
	Default Schema:	world	The schema that will be used as default schema
New Delete Dup	Nove Up	Move Down	Test Connection Close

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 listbox 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.

6.6.1. 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. MySQL Workbench also has the ability to store this password in a vault. On Linux, the vault is only implemented using the Gnome keyring facility even on systems based on KDE.
- 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 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 using the ADVANCED tab:

Figure 6.2. Manage DB Connections - Advanced Tab

stored connections Type a name for the connection world mytest Big Iron Server Advanced localhost Use compression protocol Select this option for WAN connections. Use SSL if available (client library support needed) This option turns on SSL encryption if the client library supports Use ANSI quotes to quote identifiers. If enabled this option overwrites the serverside settings.	New J. Commentioner	Connection Names World	Turns a same for the connection
Wond mytest Big Iron Server localhost Connection Method: Standard (TCP/IP) Method to use to connect to the R Parameters Advanced Use compression protocol Select this option for WAN connections. Use SSL if available (client library support needed) This option turns on SSL encryption if the client library supports Use ANSI quotes to quote identifiers. If enabled this option overwrites the serverside settings. 	stored Connections	Connection Name: world	Type a name for the connection
Big Ton Server localhost Parameters Advanced Use compression protocol Select this option for WAN connections. Use SSL if available (client library support needed) This option turns on SSL encryption if the client library supports Use ANSI quotes to quote identifiers. If enabled this option overwrites the serverside settings.	mytest	Connection Method: Standard (TCP/IP)	 Method to use to connect to the RDBM
Iocalhost Use compression protocol Select this option for WAN connections. Use SSL if available (dient library support needed) This option turns on SSL encryption if the dient library supports Use ANSI quotes to quote identifiers. If enabled this option overwrites the serverside settings.	Big Iron Server	Parameters Advanced	
 Use ANSI quotes to quote identifiers. If enabled this option overwrites the serverside settings. 	localhost	Use compression protocol	Select this option for WAN connections.
Use ANSI quotes to quote identifiers. If enabled this option overwrites the serverside settings.			mis option tams on second yption in the ciencilorary supports it
		Use ANSI quotes to quote identifiers.	If enabled this option overwrites the serverside settings.

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 quotation marks to quote literal strings, because it is interpreted as an identifier. Note: if this option is selected, it overrides the server setting.

6.6.2. 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.

Ths option can be seen in the following screenshot:

Manage DB Connections			
Stored Connections world	Connection Name: Connection Method:	world Local Socket/Pipe	Type a name for the connection • Method to use to connect to the RDBMS
Big Iron Server localhost	Parameters Advance	ed	Path to local cocket or nine file. Leave empty for default
	Username:	root	Name of the user to connect with.
	Password:	Store in Vault Clear	The user's password.
	Default Schema:	world	The schema that will be used as default schema
New Delete Dup	licate Move Up	Move Down	Test Connection Close

Figure 6.3. Manage DB Connections - Socket/Pipe Parameters

Advanced

The only option available in this tab is **USE ANSI QUOUTES TO QUOTE IDENTIFIERS**. This option was discussed in Section 6.6.1, "Standard TCP/IP Connection".

6.6.3. Standard TCP/IP over SSH Connection

The connection type allows MySQL Workbench to connection 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

Manage DB Connections				X
Stored Connections	Connection Name: wo	vild		Type a name for the connection
world	Connection Method: St	andard TCP/IP over SSH	•	Method to use to connect to the RDBMS
Big Iron Server	Parameters Advanced			
localhost	SSH Hostname:	127.0.0.1:22	SSH serve	r hostname, with optional port number
	SSH Username:	user	Name of t	ne SSH user to connect with.
	SSH Password:	Store in Vault Clear	Password	for the SSH tunnel.
	SSH Key File:		Path to SS	H key file.
	MySQL Hostname:	127.0.0.1	MySQL ser	ever host relative to the SSH server
	MySQL Server Port:	3306	TCP/IP po	rt of the MySQL server
	Username:	root	Name of t	ne user to connect with.
	Password:	Store in Vault Clear	The MySQ	L user's password.
	Default Schema:	world	The schem	a that will be used as default schema
New Delete Dup	Nove Up	love Down		Test Connection Close

Advanced

The options here are the same as for the Standard TCP/IP connection, details of which can be found in Section 6.6.1, "Standard TCP/IP Connection".

6.7. SQL Editor

MySQL Workbench 5.2 introduced the SQL Editor facility. The SQL Editor can be launched using various action items on the Home screen. It can also be launched by selecting <u>DATABASE</u>, <u>QUERY DATABASE</u> 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
- SQL Query Panel
- Main Tabsheets (Overview, Output, History, Snippets, Results)
- Sidebar

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

6.7.1. Main Menu

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

- EXECUTE (ALL OR SELECTION): Causes all statements in the SQL Query area to be executed, or only the statements selected.
- <u>EXECUTE CURRENT STATEMENT</u>: Causes the current SQL statement to be executed.

- <u>EXPLAIN (ALL OR SELECTION)</u>: Describes all statements, or the selected statement.
- <u>EXPLAIN CURRENT STATEMENT</u>: Describes the current statement.
- STOP: Stop execution of the currently running script.
- <u>RECONNECT TO SERVER</u>: Reconnect to the MySQL server.
- <u>NEW TAB</u>: Creates a duplicate of the current SQL Editor tab.
- <u>COMMIT TRANSACTION</u>: Commits a database transaction.
- <u>ROLLBACK TRANSACTION</u>: Rolls back a database transaction.
- <u>REFRESH</u>: Synchronizes with the live server and refreshes views such as the live Overview tabsheet.
- <u>COMMIT RESULT EDITS</u>: Commits any changes you have made to the server.
- <u>DISCARD RESULT EDITS</u>: Discards any changes you have made.
- <u>EXPORT RESULTS</u>: This enables you to export result sets to a file. Selecting this option displays the **EXPORT QUERY RESULTS TO FILE** dialog. The dialog enables 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.

6.7.2. Toolbar

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

Figure 6.6. SQL Editor - Toolbar



From left to right these are:

- Create a New SQL Script File: Creates a new SQL Script tab where SQL code can be entered.
- **Open a SQL Script File**: Cicking this button allows any saved SQL script to be loaded ready for execution. The script will be displayed in the **SQL QUERY** 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.
- Execute Current SQL Statement in Connected Server: Executes the current SQL statement. Results are displayed in one or more Results tabs.
- Explain (All or Selection): Explain SQL statements, or the currently selected one.
- 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.

6.7.3. SQL Query 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 Query panel, click the SAVE SQL TO SNIPPETS LIST, enter a name, and click OK. The snippet can be inserted into the SQL Query panel at any time by double-clicking the named snippet in the SQL Snippets Palette.

Figure 6.7. SQL Editor - SQL Query 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 Query panel.

6.7.4. Main Tabsheets

The main tabsheets area contains several tabs:

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





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

6.7.4.1. 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.

6.7.4.2. 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.

6.7.4.3. 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 the time, and the SQL code executed will be displayed in the SQL column.

6.7.4.4. Snippets Tabsheet

When you enter SQL code into the SQL Query 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 Tab. You can load any SQL snippet into the SQL Query area by double-clicking the desired snippet in the Snippets Palette.

٥v	erview Output His	story Snippets	Result (1)
☆ :	≿ 🖏 🗞 Snippet Co	llection: User Snippet:	s ·
SQL A B C D	Snippets Test Snippet select Name, Population fit Tony Snippet select Name, Code from (C Another Snippet select * from Country; A Snippet For All Seasons select Name, Population f	rom Country; rom	<pre>select Name, Population from Country where Continent = "Asia"; select * from Country; select Name, Code from Country; select Name, Population from Country;</pre>
			4

Figure 6.9. SQL Editor - Snippets Palette

6.7.4.5. 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 6.10. SQ	Editor - Results	Tabsheets
-----------------	------------------	-----------

0	verview Output	History	Snippets	Result (1)	×		
ĸÞ			🖌 🖶 🖦	🤣 🔦 🚸		Fetched 239 records	
	name	code					-
۱.	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					-
							· ·

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

Figure 6.11. SQL Editor - Results Tabsheets Navigation Controls

😣 🗇 🗘 🖽 🔢 🇱 🗱 🖓 🍪	Fetched 239 records	
-------------------	---------------------	--

There are a number of controls available. The controls from left to right are:

- 1. Move to first row: Highlights the first record in the current result set.
- 2. Move to previous row: Highlights the previous record.
- 3. Move to next row: Highlights the next record.
- 4. **Move to last row**: Highlights the last record in the current result set.
- 5. **Toggle wrapping of cell contents**: The data in the cell can either be truncated or wrapped. This button enables you to toggle between these options.
- 6. Sort Ascending: Sorts selected column in ascending order.
- 7. Sort Descending: Sorts column in descending order.
- 8. Export record set to an external file: Outputs record set to a CSV, HTML, or XML file as required.
- 9. Refresh Data from Data Source: Refreshes the current result set from the data source.
- 10. Search for substring within data: Search data for the string entered in the search box.

6.7.4.6. 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 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 some additional controls. These controls are highlighted in the following screenshot:

0	verview	Output History	Snippets	country (1) ×					
⊳	\$\phi\$	M 🏗 🎼 👫 🛛	H 🖌 🖦 🖷	⊨ 🍫 🛸 💊 😂		Fetched 239 rec	ords		-
	Code	Name 🔺	Continent	Region	SurfaceArea	IndepYear	Population	LifeExpectancy	
۶.	AFG	Afghanistan	Asia	Southern and Central Asia	652090.00	1919	22720000	45.9	5
	ALB	Albania	Europe	Southern Europe	28748.00	1912	3401200	71.6	3
	DZA	Algeria	Africa	Northern Africa	2381741.00	1962	31471000	69.7	4
	ASM	American Samoa	Oceania	Polynesia	199.00	NULL	68000	75.1	3
	AND	Andorra	Europe	Southern Europe	468.00	1278	78000	83.5	1
	AGO	Angola	Africa	Central Africa	1246700.00	1975	12878000	38.3	e
	AIA	Anguilla	North America	Caribbean	96.00	NULL	8000	76.1	e
	ATA	Antarctica	Antarctica	Antarctica	13120000.00	NULL	0	NULL	C
	ATG	Antigua and Barbuda	North America	Caribbean	442.00	1981	68000	70.5	e
	ARG	Argentina	South America	South America	2780400.00	1816	37032000	75.1	3
	ARM	Armenia	Asia	Middle East	29800.00	1991	3520000	66.4	1
•	ADIA	A	III	Carle and Carle	102.00	NULL	102000	70.4	+
Rol	back comple	ete							\$

Figure 6.12. SQL Editor - Live Editing Tabsheet Navigation Controls

From left to right the highlighted controls are:

- 1. Edit current row: Enters edit mode for the currently selected record.
- 2. **Insert new row**: Inserts a new row and enables you to enter data. Your changes will not be reflected on the live server until you click APPLY CHANGES TO DATA.
- 3. **Delete selected rows**: Removes the selected rows. Your changes will not be reflected on the live server until you click APPLY CHANGES TO DATA.
- 4. Apply changes to data: Applies any changes that may have been made to the data fields to the live server.
- 5. **Discard changes to data**: Discards any changes that may have been made to the data fields, and does not apply them to the live server.

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 quotation marks. 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 7.7.1.3.9, "The Inserts Tab".

6.7.5. Sidebar

The Sidebar contains several panels. These are:

- Connection Information Panel
- Object Browser

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

6.7.5.1. Connection Information Panel

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

Figure 6.13. SQL Editor - Connection Information Palette

```
Connection Information

Name: world

Host: 127.0.0.1:3306

Server: MySQL

Version: 5.1.40-community

User:
```

6.7.5.2. Object Browser

The Object Browser 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 CONNEC-TIONS, then in the MANAGE DB CONNECTION dialog set the default schema on the PARAMETERS tab.

Figure 6.14. SQL Editor - Default Schema Listbox

Object Browser					
Default:	world 👻				
÷ 00 (dvd_collection mydb test				

Schemata Explorer

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

Figure 6.15. 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 Query area. This reduces typing significantly when entering SQL statements containing several tables, columns or views.

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

- <u>SELECT ROWS LIMIT 1000</u>: Pulls up to 1000 records of table data from the live server into a Results tabsheet.
- EDIT TABLE DATA: Pulls table data from the live server into a named tabsheet, and allows editing. Data can be saved directly to the live server.
- <u>COPY TO CLIPBOARD</u>: There are various sub-menus:
 - Name (short): Copies the table name to the clipboard.
 - Name (long): Copies the table name to the clipboard in the form `schema`.`table`.
 - Column Names: Copies the column names to the clipboard in the form `table`.`column1`, `table`.`column2`,....
 - Select All Statement: Copies a SELECT all columns statement to the clipboard in the form:

```
SELECT
`table`.`column1`,
`table`.`column2`,
...
FROM `schema`.`table`;
```

- Insert Statement: Copies an INSERT all columns statement to the clipboard.
- Update Statement: Copies an UPDATE all columns statement to the clipboard.

- Delete Statement: Copies a DELETE statement to the clipboard in the form DELETE FROM `world`.`country` WHERE <where_condition>;.
- <u>SEND TO SQL EDITOR</u>: Provides similar functionality to Copy to Clipboard. However, this options inserts the SQL code directly into the SQL Query panel, where it can be edited further as required.
- <u>ALTER TABLE</u>: Displays the table editor loaded with the details of the table clicked on.
- <u>CREATE TABLE</u>: Launches a dialog to allow you to create a new table.
- <u>DROP TABLE</u>: Drops a table. All data will be lost if this operation is carried out.
- <u>REFRESH ALL</u>: Refreshes all schemata in the explorer by resynching with the server.

Right-clicking an empty area inside the object browser displays the following menu options:

- <u>CREATE SCHEMA</u>: This enables 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.
- <u>REFRESH ALL</u>: Simply synchronizes with the live server so that information with the schemata explorer is updated.

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

These action items are described in the following sections.

7.1. 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.

You can read more about modeling in the section Section 7.5, "Model Editor".

7.2. 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.

You can read more about modeling in the section Section 7.5, "Model Editor".

7.3. 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 enables 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 7.7.9.2, "Reverse Engineering a Live Database".

7.4. 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 enables you to select the script you want to create your model from.

For further information see Section 7.7.9.1, "Reverse Engineering Using a Create Script".

7.5. Model Editor

When the Model Editor is executed from the Home Screen, the MySQL Model page is displayed. The MySQL Model page has three main panels: Description Editor, User Types List/History panel, and the main panel - the Model Overview panel. The Description Editor and User Types List/History panel are contained within the Sidebar. The Sidebar is located on the left by default, but can be relocated to the right using a setting in the Workbench Preferences dialog.

MySQL Workbench		
File Edit View Model Dat	abase Plugins Scripting Community Help	
: 🔁 🖉 🖨 🗠 🗠 🔁 🛅	🛅 🛅 🔰 🤒 🔛	N
Home MySQL Model (world.m	iwb) ×	
Description Editor	Model Overview	
country: MySQL Table	Add Diagram EER Diagram EER Diagram 1 EER Diagram 2	
	Physical Schemata	··· ··· + -
	mydb MySQL Schema mySQL Schema mySQL Schema mySQL Schema mySQL Schema	
	Tables (3 items)	
Description	Views (1item)	
User Types List	Add View 🔄 view 1	
	Routines (0 items)	
Name Definition Flags A	👸 Add Routine	
BOOLE TINYINT(1)	Routine Groups (0 items)	
FLOAT4 FLOAT	Add Group	
INT1 TINYINT(4)	Schema Privileges	
INT2 SMALLINT(6)	► SQL Scripts	
INT4 INT(11)	Model Notes	
INT8 BIGINT(20)		
User Types History		
Ready		 E ;

Figure 7.1. The MySQL Model page

The sections within the Model Overview panel are:

- EER Diagrams
- · Physical Schemata
- Schema Privileges

- 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 the following sections.

7.5.1. 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.

7.5.1.1. The File Menu

Use this menu item to open a project, begin a new project, or save a project. Choosing <u>NEW MODEL</u> opens the default schema, mydb. Choosing <u>OPEN MODEL</u> 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 <u>OPEN RECENT</u> 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 <u>CLOSE TAB</u> option. You can also do this from the keyboard by pressing **Ctrl W**. To reopen the MySQL Model tab, see Section 7.5.1.3, "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 <u>SAVE MODEL</u> or <u>SAVE MODEL AS</u> 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 <u>IMPORT</u> menu option to import a MySQL data definition (DDL) script file, one created by issuing the command mysqldump --no-data, for example. If the script does not contain a CREATE <u>db_name</u>; 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 7.7.9.1, "Reverse Engineering Using a Create Script".

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

There are variety of options under the <u>EXPORT</u> 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 7.7.10.1, "Forward Engineering Using SQL Scripts".

Using the <u>EXPORT</u> 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 7.45, "The sakila EER Diagram".

The PAGE SETUP menu item enables 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 7.9, "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.

7.5.1.2. 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 <u>UNDO DELETE 'OBJECT_NAME</u>' option. The **Ctrl Z** key combination can also be used to undo an operation. It is also possible to carry out a <u>REDO</u> operation using either the menu item, or the key combination **Ctrl Y**.

Also find a <u>DELETE 'OBJECT NAME</u>' 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 <u>DELETE 'OBJECT NAME'</u> 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 5.4.4, "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 <u>EDIT SELECTED</u> to edit the currently selected object. You can also perform edits in a new window by selecting <u>EDIT SELECTED</u> in <u>New WINDOW</u>. The keyboard shortcut for <u>EDIT SELECTED</u> is **Ctrl E** and **Ctrl Shift E** for <u>EDIT SELECTED</u> IN <u>New WINDOW</u>.

The <u>SELECT</u> option has the following submenus:

- SELECT ALL (Keyboard shortcut, Ctrl A): Select all the objects on the active EER diagram.
- <u>SIMILAR FIGURES</u> (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 <u>SIMILAR FIGURES</u> and the <u>CONNECTED FIGURES</u> 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 <u>GO TO NEXT SELECTED</u> or <u>GO TO PREVIOUS SELECTED</u> menu options.

Selecting items changes some of the <u>EDIT</u> menu options. If only one object is selected, that object's name appears after the <u>CUT</u>, <u>COPY</u> and <u>DELETE</u> menu options. If more than one object is selected, these menu items show the number of objects selected.

7.5.1.2.1. Find Dialog Window

The <u>FIND</u> menu item displays a sub-menu with the following menu items:

- <u>FIND</u>: 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.
- <u>FIND NEXT</u>: Finds the next occurrence of the object.
- FIND PREVIOUS: Finds the previous occurrence of the object.
- <u>SEARCH AND REPLACE</u>: 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 7.2. The Find Window

Find		
Find Text:	✓	
In Location:	Current View	
	Match Case	
	Whole Word	
	Search in Comments	
	Search in SQL for Views, SPs etc	
	Previous Next Find All)

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.

7.5.1.2.2. Workbench Preferences

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

For further information see Section 5.4, "Workbench Preferences".

7.5.1.3. The View Menu

The Options available under this menu item are:

- <u>HOME</u>: Selects the Home screen.
- <u>WINDOWS</u>
 - MODEL NAVIGATOR: Open the Model Navigator palette
 - <u>CATALOG</u>: Open the Catalog palette
 - LAYERS: Open the Layers palette
 - USER DATATYPES: Open the User Datatypes palette
 - <u>OBJECT DESCRIPTION</u>: Open the Description palette
 - <u>OBJECT PROPERTIES</u>: Open the Properties palette
 - <u>UNDO HISTORY</u>: Open the History palette These menu options provide a means for opening the windows associated with these options.
- <u>ADVANCED</u>
 - <u>OUTPUT WINDOW</u>: Use this option to display the console output. The keyboard shortcut for this menu item is Ctrl F2.
 - <u>GRT SHELL</u>: Open the GRT shell. For more information about the GRT shell see Section 9.5, "The Workbench Scripting Shell". The keyboard shortcut for opening the GRT shell is **Ctrl F3**.
- <u>RESET WINDOW LAYOUT</u>: Reset all windows to their default layout.
- <u>ZOOM 100%</u>: The default level of detail of an EER diagram.
- <u>ZOOM IN</u>: Zoom in on an EER diagram.
- <u>ZOOM OUT</u>: 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 7.5.8, "The Model Navigator Panel".

- <u>SET MARKER</u>: 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.

7.5.1.4. 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:

- <u>ALIGN TO GRID</u>: Align items on the canvas to the grid lines.
- <u>BRING TO FRONT</u>: Use this option to bring objects to the foreground.
- <u>SEND TO BACK</u>: Use this option to move objects to the background.

- <u>CENTER DIAGRAM CONTENTS</u>: Use this option to center objects on the canvas.
- <u>AUTOLAYOUT</u>: Use this option to automatically arrange objects on the canvas.
- <u>RESET OBJECT SIZE</u>: 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.
- <u>EXPAND ALL</u>: 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 <u>COLLAPSE ALL</u> menu option.
- <u>COLLAPSE ALL</u>: Undo the operation performed by <u>EXPAND ALL</u>.

7.5.1.5. 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 7.5.1.5.1, "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 7.5.1.5.3, "The Object Notation Menu Options".
- **RELATIONSHIP NOTATION**: The items available under this option are discussed in Section 7.5.1.5.4, "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 7.5.1.5.2, "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 5.4.4, "The Model Tab".

7.5.1.5.1. 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 about this menu item in more detail in the following section The DBDoc Model Reporting Dialog Window.

7.5.1.5.2. The Validation Menu Options (Commercial Version)

Under the <u>MODEL</u> menu option there are two validation options, <u>VALIDATION</u> and <u>VALIDATION</u> (<u>MYSQL</u>). 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 <u>VALIDATION</u> option the menu items are:

- <u>VALIDATE ALL</u>: Perform all the validation options available
- EMPTY CONTENT VALIDATION: Check for objects with no content, for example a table with no columns
- <u>TABLE EFFICIENCY VALIDATION</u>: Check the efficiency of tables, for example a table with no primary key defined
- <u>DUPLICATE IDENTIFIERS VALIDATION</u>: Check for duplicate identifiers, for example two tables with the same name
- <u>CONSISTENCY VALIDATION</u>: 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 <u>VALIDATION (MYSQL</u>) option the menu items are:

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

For detailed information about validation see Section 7.10, "MySQL Workbench Schema Validation Plugins (Commercial Version)".

7.5.1.5.3. The Object Notation Menu Options

The options under the <u>OBJECT NOTATION</u> 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:

- <u>WORKBENCH (DEFAULT)</u>: Display table columns, indexes, and triggers.
- <u>WORKBENCH (SIMPLIFIED)</u>: Show only a table's columns.
- <u>WORKBENCH (PKS AND FKS ONLY)</u>: Show only columns that are primary and foreign keys.
- <u>CLASSIC</u>: Similar to the Workbench (Simplified) style showing only the table's columns.
- <u>IDEF1X</u>: 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.

7.5.1.5.4. The Relationship Notation Menu Option

The options under the <u>RELATIONSHIP NOTATION</u> 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 7.42, "Adding Tables to the Canvas".
- <u>CLASSIC</u>: Uses a diamond shape to indicate cardinality.
- <u>CONNECT TO COLUMNS</u>
- <u>UML</u>: Universal Modeling Language style.
- <u>IDEF1X</u>: 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.

7.5.1.6. The Database Menu

There are several options under the <u>DATABASE</u> menu option:

- <u>OUERY DATABASE</u>: Launche the SQL Editor, which enables you to create SQL code and execute it on a live server. For more information see Section 6.7, "SQL Editor".
- <u>MANAGE CONNECTIONS</u>: Launche the Manage DB Connections dialog, which enables you to create and manage multiple connections. For more information see Section 6.6, "Manage DB Connections Dialog"
- <u>REVERSE ENGINEER</u>: Create a model from an existing database. For more information, see Section 7.7.9.2, "Reverse Engineering a Live Database".
- FORWARD ENGINEER: Create a database from a model. For more information, see Section 7.7.10.2, "Forward Engineering to a Live Server".
- <u>SYNCHRONIZE MODEL</u>: Synchronize your database model with an existing database. For more information, see Section 7.7.10.3, "Database Synchronization".
- <u>GENERATE CATALOG DIFF REPORT</u>: Compare your schema model with a live database or a script file. Section 7.7.10.4, "Creating a Catalog Diff Report".

7.5.1.7. The Plugins Menu

The <u>PLUGINS</u> menu option lists any plugins that you may have installed. For more information about this menu option see Section 9.3, "Plugins".

7.5.1.8. The Scripting Menu

This menu currently has three items:

- <u>SCRIPTING SHELL</u>: Launches the MySQL Workbench Scripting Shell.
- <u>RUN WORKBENCH SCRIPT</u>: Executes the specified script.
- <u>INSTALL PLUGIN/MODULE FILE</u>: Enables you to load and install a plugin/module file.

7.5.1.9. The Community Menu

The <u>COMMUNITY</u> menu option offers the following choices:

- <u>WORKBENCH BLOG</u>
- FAQs About Workbench
- LEARN HOW TO CODE FOR WORKBENCH
- DISCUSS WORKBENCH TOPICS
- <u>CONTRIBUTE TO WORKBENCH</u>

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

7.5.1.10. The Help Menu

The <u>HELP</u> menu option offers the following choices:

- <u>HELP INDEX</u>: Open a window 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.
- <u>UPDATE</u>: Update to the latest version.
- <u>MYSQL.COM WEBSITE</u>: Open your default browser on the MySQL website home page.
- <u>WORKBENCH PRODUCT PAGE</u>: Open your default browser on the MySQL Workbench product page.
- <u>SYSTEM INFO</u>: Information about your system, useful when reporting a bug. For more information, see Section 7.5.1.10.1, "System Info".
- <u>REPORT A BUG</u>: Open your default browser on the MySQL bug report page.
- <u>VIEW REPORTED BUGS</u>: Open your default browser to see a list of current bugs.
- <u>ABOUT WORKBENCH</u> : Show the MySQL Workbench About window.

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

7.5.1.10.1. System Info

Use the <u>SYSTEM INFO</u> 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_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', 'latinl'),
('datadir', '"C'/ProgramData/MySQL/MySQL Server 5.1/Data/"'),
('innob_log_buffer_size', '10M'),
('innodb_log_buffer_size', '10M'),
('innodb_log_buffer_size', '10M'),
('innodb_dfer_size', '10M'),
('innodb_dfer_size', '64K'),
('innodb_flush_log_at_trx_commit', '1'),
('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
```

7.5.2. 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

7.5.2.1. Tool-specific Toolbar Items

The toolbar also changes depending upon which tool from the vertical toolbar is active. These tools are discussed in Section 7.6.1, "The Vertical Toolbar".

If the Table tool is active, drop down list boxes of schemata, engine types and collations appear on the toolbar. The table properties can then be modified using the Properties Editor.

When an object is selected the object's properties, such as color, can be changed in the Properties Editor.

7.5.3. 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 7.6, "EER Diagram Editor".

7.5.4. 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.

7.5.4.1. 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 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 7.6, "EER Diagram Editor".

7.5.5. Schema Privileges

The Privileges panel of the MySQL Model page is used to create users for your schemata and also to define roles ----.

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



7.5.5.1. 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:

- <u>CUT 'ROLE_NAME</u>': Cut the role
- <u>COPY 'ROLE_NAME</u>': Copy the role
- <u>EDIT ROLE ...</u>: Open the role editor.
- <u>EDIT IN NEW WINDOW</u>: Open the role editor in a new editor window.
- <u>DELETE 'ROLE NAME</u>': Remove the role

• <u>COPY SQL TO CLIPBOARD</u>: Currently not implemented.

To rename a role, simply click the role name and you will then be able to edit the text.

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

accounting			▼ X
Roles	Objects	Privileges	
accounting	tblorders	 GRANT OPTION REFERENCES ALTER DELETE INDEX ✓ INSERT ✓ SELECT ✓ UPDATE TRIGGER 	
< >	Dupa phiosts from the catalog tree	to accion privilages	~
Roles	Prorag objects from the catalog tree	to assign privileges.	

Select the role that you wish to add objects to. You may drag and drop objects from the Physical Schemata 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 SE-LECT 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.

7.5.5.2. 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 7.5.5.1, "Adding Roles".

7.5.6. 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 to create an ALTER script. For more information, see Section 7.7.10.1.2, "Altering a Schema".

Use the Model Notes panel to write project notes. Any scripts or notes added will be saved with your project.

7.5.7. 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, <u>COPY</u> <u>HISTORY ENTRIES TO CLIPBOARD</u>. 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.

7.5.8. The Model Navigator Panel

Docked at the top left of the application is the **MODEL NAVIGATOR**, or **BIRD'S EYE** panel. This panel 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 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



7.5.9. The Catalog Tree Palette

The Catalog Tree 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, which can be found in the lower left corner of the screen.

The Catalog Tree palette is primarily used to drag and drop objects onto an EER diagram canvas.

Note

On Linux, there is a quirk in the GTK tree control, where a simple click always generates a new selection. If you want to drag multiple objects from the Catalog Tree to the EER diagram canvas, you need to perform the operation as follows:

1. Click first item in tree.
- 2. Hold **shift** and click last item and *do not release the shift key*.
- 3. Keep the **shift** key depressed and commence the dragging operation.
- 4. Release the **shift** key before you release the mouse button to successfully drop selected objects onto the canvas.

This also applies to use of the Ctrl key when selecting multiple non-adjacent elements in the Catalog Tree.

You can toggle the sidebar on and off using the TOGGLE SIDEBAR button, which is located in the top right of the application.

7.5.10. 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 <u>SELECT</u> menu option. For more information on this topic see Section 7.5.1.2, "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.

7.5.10.1. 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 object, such as a relationship, that has been set to hidden. Open the Layers palette and select the object by double-clicking it. You can then edit the object and change its visibility setting to Fully Visible.

7.5.11. 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 7.7.2.3, "The Properties of a Connection".

7.6. 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.

7.6.1. 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.

7.6.1.1. 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.

7.6.1.2. 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 <u>VIEW</u>, <u>WINDOWS</u>, <u>MODEL NAVIGATOR</u> 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 7.5.8, "The Model Navigator Panel".

7.6.1.3. 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 5.4.4, "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 <u>DELETE</u> from the pop up menu.

7.6.1.4. 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 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 7.7.5, "Creating Layers".

7.6.1.5. 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, <u>EDIT NOTE</u> ... or <u>EDIT IN NEW</u> <u>WINDOW</u>

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 7.7.7, "Creating Text Objects".

7.6.1.6. The Image Tool

Use the image tool to place an image on the canvas. When this tool is selected and you click 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 7.7.8, "Creating Images".

7.6.1.7. The Table Tool

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

Clicking the canvas, creates a table. To edit this table, right-click it and choose <u>EDIT TABLE</u> or <u>EDIT IN NEW WINDOW</u> 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 7.7.1.3, "The MySQL Table Editor".

7.6.1.8. 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 the canvas creates a new view. To edit this view, right-click it and choose <u>EDIT VIEW</u> or <u>EDIT IN NEW</u> <u>WINDOW</u>... 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 7.7.3, "Creating Views".

7.6.1.9. 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 the canvas creates a new group. To edit this view, right-click it and choose <u>EDIT ROUTINE GROUP</u> or <u>EDIT IN NEW WINDOW</u> ... 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 7.7.4.2, "Routine Groups".

7.6.1.10. 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 7.7.2, "Creating Foreign Key Relationships".

7.7. Working with Models

7.7.1. Creating Tables

7.7.1.1. 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 7.7.1.3, "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
- <u>COPY SQL TO CLIPBOARD</u>
- <u>COPY INSERT TO CLIPBOARD</u>
- <u>Delete 'table name'</u>

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

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

Warning

Use the <u>DELETE 'TABLE NAME</u>' 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.

7.7.1.2. 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 \mathbf{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

🔲 address 🔹 🔻
<pre>? address_id SMALLINT(5)</pre>
♦ address VARCHAR(50)
♦ address2 VARCHAR(50)
♦ district VARCHAR(20)
♦ city_id SMALLINT(5)
♦ postal_code VARCHAR(10)
♦ phone VARCHAR(20)
♦ last_update TIMESTAMP
Indexes 🛛 🔻
PRIMARY
idx_fk_city_id
Triggers 🕨

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:

- <u>Cut 'table name'</u>
- <u>COPY 'TABLE_NAME'</u>
- <u>EDIT TABLE</u>
- EDIT IN <u>New Window</u>
- <u>COPY SQL TO CLIPBOARD</u>
- <u>COPY INSERT TO CLIPBOARD</u>
- <u>Delete 'table name'</u>

With the exception of the deletion option, these menu options function as described in Section 7.7.1.1, "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 5.4.4, "The Model Tab".

7.7.1.3. 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.

7.7.1.3.1. The Main Editor Window

Any number of tables may be edited in the MySQL Table Editor at any one time. Adding another 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.

The MySQL Table Editor is shown in the following figure.



country ×	(
	Name:	country	The name of the table. It is recommended to use only alpha-numeric characters. Spaces should be avoided and be replaced by _
	Collation:	latin1 - default collation	The charset/collation specifies which language specific characters can be stored in the table and their sort order. Common choices are Latin1 or UTF8.
	Engine:	MyISAM	The database engine that is used for the table. This option affects performance, data consistency and much more.
	Comments:		A
			v
Table	Columns	Indexes Foreign Keys Trigg	ers Partitioning Options Inserts Privileges

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

- TABLE: Use this table 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.

7.7.1.3.2. 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.

7.7.1.3.3. 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

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default		Column Details
SurfaceArea	FLOAT		1						'0.00'		Collation:
IndepYear	SMALLINT(6)								NULL		Table Default
Population	INT(11)		1						'0'	_	Comments:
LifeExpectancy	FLOAT								NULL		
GNP	FLOAT								NULL		
GNPOId	FLOAT								NULL		
LocalName	CHAR(45)		1							=	
GovernmentForm	CHAR(45)		1							_	
HeadOfState	CHAR(60)								NULL		
🛇 Capital	INT(11)								NULL		
Code2	CHAR(2)		1								
										-	

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:

- <u>MOVE UP</u>: Move the selected column up.
- <u>MOVE DOWN</u>: Move the selected column down.
- <u>DELETE SELECTED COLUMNS</u>: Select multiple contiguous columns by right-clicking and pressing the **Shift** key. Use the **Ctrl** key to select noncontiguous columns.
- <u>**REFRESH</u>**: Update all information in the Columns tab.</u>
- <u>CLEAR DEFAULT</u>: Clear the assigned default value.
- <u>DEFAULT NULL</u>: Set the column default value to NULL.

• <u>DEFAULT</u>: Set the column default value to ".

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

You can also add column comments to the Column Comment text area. It is also possible to set the column collation, using the listbox in the COLUMN DETAILS panel.

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 or a white diamond. A blue diamond indicates the column has NN set. To add or remove a column from the primary key, double-click 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 quotation 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 quotation marks. If a non-numeric default value is entered without quotation marks, 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.

7.7.1.3.4. The Indexes Tab

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

country 3	×										
Index Nam	e	Туре		Index Columns					Index Options —		
PRIMARY		PRIMARY		Column	#	Order	Length	*	Storage Type:		•
				🔽 Code	1	ASC			Key Black Cierce		_
				Name		ASC			Key Block Size:	U	
				Continent		ASC			Parser:		
				Region		ASC					
				SurfaceArea		ASC		Ξ	Index Comment -		
				IndepYear		ASC					
				Population		ASC					
				LifeExpectancy		ASC					
				GNP		ASC					
				GNPOId		ASC					
				LocalName		ASC					
				GovernmentForm	n	ASC		-			-
J				HaadOfStata		NSC.					
Table	Columns	Indexes	Foreign Ke	ys Triggers F	Partitioning	Optio	ons Ins	erts	Privileges		

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.

7.7.1.3.5. 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.

7.7.1.3.6. 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.

7.7.1.3.7. 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 also available. For more information about partitioning see Partitioning.

7.7.1.3.8. The Options Tab

The **OPTIONS** tab enables you to set several types of options. These are grouped into the following frames:

- General Options
- Row Options
- Storage Options
- Merge Table options

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

General 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.

Row Options

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.

Storage 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.

Merge Table Options

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.

7.7.1.3.9. The Inserts Tab

Use the Inserts tab to insert records into the table.

To edit a record simply click the field you wish to change and enter the new data. Right-clicking a row displays a menu with the following items:

- Set Field(s) to NULL:
- Delete Row(s):
- Copy Row Content: Copies the row to the clipboard. Strings are copied quoted, and NULLs are preserved.
- Copy Row Content (unquoted): Copies the row to the clipboard. Strings are not quoted and NULLs are copied as a space.
- Copy Field Content: Copies the value of the selected field to the clipboard. Strings are quoted.
- Copy Field Content (unquoted): Copies the value of the selected field to the clipboard. Strings are not quoted.

Note that the insert editor features a toolbar. This has the same functionality as explained in Section 6.7.4.5, "Results Tabsheets" and Section 6.7.4.6, "Live Editing Tabsheets". You can also hover the cursor over the toolbar to display tooltips.

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 that 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 quotation marks, 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 quotation marks. 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.

7.7.1.3.10. 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 7.5.5.1, "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 the role in the **ROLES** list box. Doing this displays all available privileges in the **ASSIGNED PRIV-ILEGES** 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 as listed previously. 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.

7.7.2. 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 7.7.1.3.5, "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.

7.7.2.1. Adding Foreign Key Relationships Using an EER Diagram

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

- The one-to-one non-identifying relationship tool
- The one-to-many non-identifying relationship tool
- The one-to-one identifying relationship tool
- The one-to-many identifying relationship tool
- The many-to-many identifying relationship tool
- The Place a Relationship Using Existing Columns 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 rela-

tionship, first click the table that is on the "many" side of the relationship, 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 <u>MODEL</u>, <u>MENU OPTIONS</u> menu item to set a project-specific default name for the foreign key column (see Section 7.5.1.5.4, "The Relationship Notation Menu Option"). To change the global default see Section 5.4.4, "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 7.7.2.3, "The Properties of a Connection". Where the notation style allows, Classic for instance, the cardinality indicators can also be repositioned.

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 7.5.1.5.4, "The Relationship Notation Menu Option".

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

7.7.2.2. 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 <u>MODEL</u>, <u>MENU OPTIONS</u> menu item to set a project-specific default name for foreign keys. To change the global default see Section 5.4.4, "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

🔲 tblpublishers 🕨	
ŧ	
fk_pub	
<u> </u>	
🔲 tblbooks	

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 <u>MODEL</u>, <u>RELATIONSHIP NOTATION</u> menu item. For more information, see Section 7.5.1.5.4, "The Relationship Notation Menu Option".

The Foreign Key tab

The FOREIGN KEY tab contains several frames: REFERENCING TABLE, CARDINALITY 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.

In the Cardinality frame there is a set of radio buttons that allow you to choose whether the relationship is one-to-one or one-to-many. There is also a checkbox that enables you to specify whether the relationship is an identifying relationship.

7.7.2.3. 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 7.7.2.2, "The Relationship Editor".
- many: False if the relationship is a one-to-one relationship.
- middleSegmentOffset: The offset of the middle section of the connector.
- modelOnly: when this is set the connection will not be propagated to the DDL. It is just a logical connection drawn on a diagram. This is used, for example, when drawing MyISAM tables with a visual relationship, but with no foreign keys.
- 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 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** palette and select the relationship object. Once selected, you can edit the object by right-clicking, and selecting <u>EDIT OBJECT</u>. You can then set the **VISIBILITY SETTINGS** to **FULLY VISIBLE**. The relationship will then be visible in the **EER DIAGRAM** window.

7.7.3. 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.

7.7.3.1. 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 7.7.3.3, "The View Editor".

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

- CUT 'VIEW NAME'
- <u>COPY 'VIEW_NAME'</u>
- <u>Paste</u>
- EDIT VIEW...
- EDIT IN NEW WINDOW
- <u>COPY SQL TO CLIPBOARD</u>
- <u>Delete 'view_name'</u>

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

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

Warning

Use the <u>DELETE 'VIEW_NAME'</u> 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 left side of the application. They may be added to an EER Diagram, when in EER Diagram view, by dragging and dropping them from this palette.

7.7.3.2. 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 left-click the view icon on the vertical toolbar. The view icon is the two overlapping rectangles found below the table icon.

Clicking 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 main 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 7.7.3.1, "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 5.4.4, "The Model Tab".

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

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.

The Comments Tab

This tab enables you to enter comments for a particular view.

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 7.7.1.3.10, "The Privileges Tab".

7.7.3.4. 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 7.5.11, "The Properties Palette".

7.7.4. 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 <u>DATABASE</u>, <u>REVERSE ENGINEER...</u> 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.

7.7.4.1. Routines

7.7.4.1.1. 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 7.7.4.1.2, "The Routine Editor".

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

- <u>Rename</u>
- <u>Cut 'routine name'</u>
- <u>COPY 'ROUTINE NAME'</u>
- <u>Paste</u>
- EDIT ROUTINE...

- <u>EDIT IN NEW WINDOW</u>
- <u>COPY SQL TO CLIPBOARD</u>
- <u>Delete 'routine name'</u>

The <u>EDIT ROUTINE</u> ... 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 5.4.4, "The Model Tab".

7.7.4.1.2. 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. 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 **ROUTINE** and the **PRIVILEGES** tabs. Navigate between different tabs using the mouse or from the keyboard by pressing Ctrl + Alt + Tab.

7.7.4.1.2.1. 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.

7.7.4.1.2.2. 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 7.7.1.3.10, "The Privileges Tab".

Note

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

7.7.4.2. Routine Groups

7.7.4.2.1. 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 7.7.4.2.3, "The Routine Group Editor".

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

• <u>Rename</u>

- <u>CUT 'ROUTINE GROUP NAME'</u>
- <u>COPY 'ROUTINE GROUP NAME'</u>
- EDIT ROUTINE...
- EDIT IN NEW WINDOW
- <u>COPY SQL TO CLIPBOARD</u>
- <u>Delete</u> '*routine* group name'

The <u>EDIT ROUTINE GROUP...</u> option opens the routine group editor. Using the routine group editor is described in Section 7.7.4.2.3, "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 Digram by dragging and dropping them from this palette.

7.7.4.2.2. 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 7.7.4.2.1, "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 5.4.4, "The Model Tab".

7.7.4.2.3. 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.

7.7.4.2.3.1. The Routine GroupsTab

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.

7.7.4.2.3.2. 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 7.7.1.3.10, "The Privileges Tab".

Note

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

7.7.4.2.3.3. 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 7.5.11, "The Properties Palette".

7.7.5. 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.

7.7.5.1. 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 rightclick 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 \mathbf{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:



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:

- <u>CUT 'LAYER NAME'.</u>
- <u>COPY 'LAYER_NAME'</u>
- <u>Delete 'layer_name'</u>

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.

7.7.5.1.1. 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.

7.7.5.2. 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 7.5.11, "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.

7.7.6. 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.

7.7.6.1. 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 7.7.6.2, "The Note Editor".

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

- <u>Rename</u>
- <u>Cut 'note name'</u>
- COPY 'NOTE NAME'
- <u>Delete 'note name'</u>

The EDIT NOTE ... option opens the note editor. Using the note editor is described in Section 7.7.6.2, "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.

7.7.6.2. 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.

7.7.7. 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.

7.7.7.1. 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.

7.7.7.2. 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.

7.7.7.2.1. 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 7.5.11, "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 5.4.7, "The Appearance Tab".

7.7.8. Creating Images

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

7.7.8.1. 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 '
- <u>COPY ' IMAGE '</u>
- <u>EDIT IMAGE ...</u>
- EDIT IN NEW WINDOW ...
- <u>Delete ' Image '</u>

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.

7.7.8.2. 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,

7.7.8.2.1. 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.

7.7.9. 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.

7.7.9.1. Reverse Engineering Using a Create Script

Reverse engineering using a create script is done by using the <u>FILE</u>, <u>IMPORT</u>, <u>REVERSE ENGINEER MYSQL CREATE SCRIPT</u>... 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 to create an ALTER script. For more information, see Section 7.7.10.1.2, "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 <u>FILE</u>, <u>SAVE</u> menu item and the reverse-engineered database will be saved as a MySQL Workbench file with the extension mwb.

See Section 7.8.1, "Importing a Data Definition SQL Script" for a tutorial on reverse engineering the sakila database.

7.7.9.2. 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

Reverse Engineer Database			
Connection Options	Set parameters t	for connecting to a DBMS	
Connect to DBMS		,	
Select Schemata	Stored Connection:	world	 Select from saved connection settings
Fetch Object Info	Connection Method:	Standard (TCP/IP)	 Method to use to connect to the RDBMS
Select Objects	Parameters Advan	iced	1
Reverse Engineer	Hostname:	127.0.0.1 Port: 3306	Name or IP address of the server host - TCP
Results	Username:	root	Name of the user to connect with.
	Password:	Store in Vault Clear	The user's password.
	Default Schema:	world	The schema that will be used as default sche
91111		_	
977			Back Next Cancel

The first page of the wizard enables you to set up a connection to the live database you wish to reverse engineer. This enables you to set up a new connection, or select a previously created connection. Typical information required for the connection includes hostname, 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:





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

Reverse Engineer Database	
Connection Options	Retrieve Object Information
Connect to DBMS	
Select Schemata	The following tasks will now be executed. Please monitor the execution.
Fetch Object Info	Press Show Logs to see the execution logs.
Select Objects	♂ Retrieve Objects from Selected Schemata
Reverse Engineer	
Results	Retrieval Completed Successfully
	Finished.
91111	
19/19	Show Logs Back Next Cancel

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 the SHOW FILTER button the following page is displayed:

Figure 7.19. Show Filter

Reverse Engineer Database		×
Connection Options	Select Objects to Reverse Engineer	
Connect to DBMS		
Select Schemata		
Fetch Object Info		
Select Objects	Here W Import MySQL Table Objects	
Reverse Engineer	3 Total Objects, 3 Selected	
Results	world.city > world.country > world.countrylanguage <	
	Place imported objects on a diagram Back Execute > Cancel	

This enables 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, such as MYSQL ROUTINE OBJECTS, have similar filters available.

You can click EXECUTE 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 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

Reverse Engineer Database	
Connection Options	Reverse Engineering Results
Connect to DBMS	
Select Schemata	
Fetch Object Info	Summary of Reverse Engineered Objects:
Select Objects	- 3 tables from schema 'world'
Reverse Engineer	
Results	
94/1/	Back Finish Cancel

Click FINISH to exit the wizard.

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

7.7.9.2.1. 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 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.

7.7.10. Forward Engineering

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

7.7.10.1. Forward Engineering Using SQL Scripts

To create a script of your database model use the <u>EXPORT</u> option found under the <u>FILE</u> 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 mysgldump <u>db_name</u> command.

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

7.7.10.1.1. 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:



Forward Engineer SQL Script							
SQL Export Options	SOL Export Options						
Filter Objects							
Review SQL Script	Output SQL Script File:						
	Leave blank to view generated script but not save to a file.						
	Omit Schema Qualifier in Object Names						
	Generate Separate CREATE INDEX Statements						
	Add SHOW WARNINGS After Every DDL Statement						
	Do Not Create Users. Only Export Privileges						
	Generate INSERT Statements for Tables						
79/11	Back Next Cancel						

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 7.7.1.3.9, "The Inserts Tab".

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

Forward Engineer SQL Script		X						
SQL Export Options Filter Objects	SQL Object Export Filter							
Review SQL Script	To exclude objects of a specific type from the SQL Export, disable the corresponding checkbox. Press Show Filter and add objects or patterns to the ignore list to exclude them from the export.							
	Export MySQL Table Objects 2 Total Objects, 2 Selected	Show Filter						
	Export MySQL View Objects 0 Total Objects, 0 Selected	Show Filter						
	0 Total Objects, 0 Selected	Show Filter						
	Export MySQL Trigger Objects 0 Total Objects, 0 Selected	Show Filter						
	O Total Objects, 0 Selected	Show Filter						
		Back Next Cancel						

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

Forward Engineer SQL Script		×
SQL Export Options	Review Generated Scrint	
Filter Objects		
Review SQL Script	Review the generated script.	
	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';	^
	CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET latin1 COLLATE latin1_swedish_ci ; USE `mydb`;	
	CREATE TABLE IF NOT EXISTS `mydb`.`table1` (`idtable1` INT NOT NULL , PRIMARY KEY ('idtable1`)) ENGINE = InnoDB;	
	idtable3`INT NOT NULL, 'table1_idtable1`INT NULL, PRIMARY KEY (idtable2', idtable1', idtable3'), INDEX 'ik_table2_table1'(itable1_idtable1' ASC), CONSTRAINT 'ik_table2_table1' FOREIGN KEY ('table1_idtable1') REFERENCES 'mydb','table1' (idtable1') ON DELETE NO ACTION	
	Save to Other File Copy to Clipboard	
474	Back Finish Can	cel

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.

7.7.10.1.2. Altering a Schema

The menu option for altering a schema, <u>FORWARD ENGINEER SQL ALTER SCRIPT</u>..., 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 7.7.9.1, "Reverse Engineering Using a Create Script".

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

Figure 7.26. Options
Forward Engineer an ALTER Script		×
Options	Forward Engineer ALTER Script Options	
Review SQL Script		
	Disk the COL second first the second state the second state	
	Pick the SQL script rile to be compared with the current model.	<u> </u>
	Input Hile:	Browse
	Enter the path for the ALTER script to be created.	
	Output File:	Browse
	Leave blank to view generated script but not save to a file.	
	Back Next	Cancel

This first screen enables 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

Forward Engineer an ALTER Script		X
Options	Review Generated Script	
Review SQL Script	······	
	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';	~
	CREATE TABLE IF NOT EXISTS 'mydb'.'table3' ('idtable3' INT(11) NOT NULL , 'table2_idtable2' INT(11) NULL DEFAULT NULL , PRIMARY KEY ('idtable3'), INDEX 'K_ table3_table21' ('table2_idtable2' ASC) , CONSTRAINT 'K_ table3_table21' ; FOREIGN KEY ('table2_idtable2') REFERENCES 'mydb'.'table2' ('idtable2') ON DELETE NO ACTION ON UPDATE NO ACTION ON UPDATE NO ACTION ON UPDATE NO ACTION ENGINE = InnoDB DEFAULT CHARACTER SET = latin1 COLLATE = latin1_swedish_ci; SET SQL_MODE=@OLD_SQL_MODE; SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS; SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;	
	Save to File Copy to Clipboard	*
91/11	Back Execute > Cancel	

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.

7.7.10.2. 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 (validation is only available in the Standard Edition):

Figure 7.28. Catalog Validation



Click RUN VALIDATIONS to validate the catalog.

Click the NEXT to continue.

The next page enables you to set options for the database to be created. These options are as described in Section 7.7.10.1.1, "Creating a Schema":

Figure 7.29. Options

Forward Engineer to Database	
Options	Set Options for Database to be Created
Select Objects	
Review SQL Script	Options
Connection Options	DROP Objects Before Each CREATE Object
Commit Progress	Generate DROP SCHEMA
	Skip creation of FOREIGN KEYS
	Omit Schema Qualifier in Object Names
	Generate USE statements
	Generate Separate CREATE INDEX Statements
	Add SHOW WARNINGS After Every DDL Statement
	Do Not Create Users. Only Create Privileges
	Generate INSERT Statements for Tables
94/1/	Back Next Cancel

Select the required options and then click NEXT.

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

Figure 7.30. 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:

Figure 7.31. Review Script

Forward Engineer to Database		
Options	Review the SQL Script to be Executed	
Select Objects		
Review SQL Script		
Connection Options		
Commit Progress	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';	Î
	CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET latin1 COLLATE latin1_swedish_ci ; CREATE SCHEMA IF NOT EXISTS `mysql` DEFAULT CHARACTER SET latin1 ; CREATE SCHEMA IF NOT EXISTS `test` DEFAULT CHARACTER SET latin1 ; CREATE SCHEMA IF NOT EXISTS `world` DEFAULT CHARACTER SET latin1 ;	
	Table `mysql`.`columns_priv`	
	CREATE TABLE IF NOT EXISTS `mysql`.`columns_priv` (`Host` CHAR(60) CHARACTER SET 'utf8' COLLATE 'utf8_bin' NOT NULL DEFAULT ", `Db` CHAR(64) CHARACTER SET 'utf8' COLLATE 'utf8_bin' NOT NULL DEFAULT ", `User` CHAR(16) CHARACTER SET 'utf8' COLLATE 'utf8_bin' NOT NULL DEFAULT ", `Table_name` CHAR(64) CHARACTER SET 'utf8' COLLATE 'utf8_bin' NOT NULL DEFAULT ", `Column_name` CHAR(64) CHARACTER SET 'utf8' COLLATE 'utf8_bin' NOT NULL DEFAULT ", `Column_name` CHAR(64) CHARACTER SET 'utf8' COLLATE 'utf8_bin' NOT NULL DEFAULT ", `Column_name` CHAR(64) CHARACTER SET 'utf8' COLLATE 'utf8_bin' NOT NULL DEFAULT ", `Column_name` CHAR(64) CHARACTER SET 'utf8' COLLATE 'utf8_bin' NOT NULL DEFAULT ", `Column_priv` SET('Select', 'Insert', 'Update', 'References') CHARACTER SET 'utf8' NOT NULL DEFAULT ", PRIMARY KEY (`Host`, `Db`, `User`, `Table_name`, `Column_name`)) ENGINE = MyISAM DEFAULT CHARACTER SET = utf8 COLLATE = utf8_bin COMMENT = 'Column privileges';	
		Ŧ
	Save to File Copy to Clipboard	
74/14	Back Next Cancel	

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 enables you to use a previously stored connection, or enter the connection parameters:

Figure 7.32. Set parameters for connecting to a DBMS

Forward Engineer to Database			
Options	Set parameters f	for connecting to a DBMS	
Select Objects		,	
Review SQL Script	Stored Connection:	world	✓ Select from saved connection settings
Connection Options	Connection Method:	Standard (TCP/IP)	← Method to use to connect to the RDBMS
Commit Progress	Parameters Advan	ced	
	Hostname:	127.0.0.1 Port: 3306	Name or IP address of the server host - TCP
	Username:	root	Name of the user to connect with.
	Password:	Store in Vault Clear	The user's password.
	Default Schema:	world	The schema that will be used as default sche
14711			Back Execute > Cancel

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

Figure 7.33. 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.

7.7.10.3. 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 enables 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 7.34. Model and Database Differences

Connection Options	Model and Database Dif					
	i louer and bacabase bri	terences				
Connect to DBMS		Ter en ces	,			
Select Schemata	Double click arrows in the list to	o choose v	whether to ignore changes	, update the model v	vith database char	nges or
Fetch Object Info						
Select Changes to Apply	Model	Update	Source			
Review DB Changes	table1	\rightarrow	table1			
Synchronize Progress	table3 table4	111	table2 table3 N/A table5			
	Update Model Ignore	Update	Source	Back	Next	Cancel

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 7.35. Controlling Synchronization Direction

Synchronize Model with Database				×
Connection Options	Model and Database Diff	erences		
Connect to DBMS				
Select Schemata	Double click arrows in the list to	choose whether to ignore change	s, update the model with database changes or	
Fetch Object Info				_
Select Changes to Apply	Model	Update Source		
Review DB Changes	table1	table1		
Synchronize Progress	1 table2 table3 table4 N/A	table2 table3 table4 table5		
	Update Model Ignore	Update Source	Back Next Cancel	

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 allows 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 7.36. Update Model Button

Synchronize Model with Database							×
Connection Options	Model and Database	Differences					
Connect to DBMS		, Differences	,				
Select Schemata	Double click arrows in the	list to choose v	whether to ignore changes	update the model with database changes o	r vice-versa. You can also apply an actio	on to multiple selected rows.	
Fetch Object Info	Model	Update	Source				
Select Changes to Apply	🖃 🤤 mydb		mydb tablet				
Review DB Changes	table2		table2				
Synchronize Progress		ĴĴĴ	table3 table4 table5				
			table7				
C// Den 1							
	Update Model Igno	re Update	Source				
1919						Back Next	Cancel

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 7.37. Ignore Button

Synchronize Model with Database							×
Connection Options	Model and Database	Differences					
Connect to DBMS	i ibaci ana bacabase	Differences	, ,				
Select Schemata	Double click arrows in the	list to choose v	whether to ignore changes, up	late the model with database changes or vice-versa	a. You can also apply an action	to multiple selected rows.	
Fetch Object Info	Model	Update	Source				
Select Changes to Apply	🖃 🤤 mydb		mydb hable 1				
Review DB Changes	table2		table2				
Synchronize Progress	table3 table4		table3 table4 table5				
	table6	(%)	N/A table7				
	Update Model Ignor	e Update	Source				
- 14/14						Back Next	Cancel

In the above example no changes would take place.

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

Figure 7.38. Update Source Button

Synchronize Model with Database					×
Connection Options	Model and Database Di	ifforoncos			
Connect to DBMS		interence.			
Select Schemata					
Fetch Object Info	Double click arrows in the list	to choose v	whether to ignore change	is, update the model with database changes or	
	Model	Update	Source		
Select Lhanges to Apply	🖃 🤤 mydb	$\langle \Rightarrow \rangle$	mydb		
Review DB Changes	table1		table1		
Synchronize Progress	table3		table3		
	table4		table4		
	table5	\Leftrightarrow	table5		
	I table6	\Rightarrow	! N/A table7		
		_			
		· · · ·			
	Update Model Ignore	Update	Source		
				Back Next Cancel	

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 the arrows. Clicking an arrow causes it to change between the three available synchronization directions:

Figure 7.39. Click arrows to change direction of synchronization

Synchronize Model with Database				×
Connection Options	Model and Database Diff	ferences		
Connect to DBMS		lefences		
Select Schemata	Double click arrows in the list to	choose whether to ignore	changes, update the model with database changes or	
Fetch Object Info				
Select Changes to Apply	Model	Update Source		
Review DB Changes	table1	table1		
Synchronize Progress	table2 table3 table4 table5 table6	table2 table3 table4 table5 N/A table7		
	Update Model Ignore	Update Source	Back Next Canc	el

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

7.7.10.4. Creating a Catalog Diff Report

This facility enables you to create a report detailing the differences between your MySQL Workbench model, and a live database or script. Select <u>DATABASE</u>, <u>GENERATE CATALOG DIFF REPORT</u> 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 7.40. Catalog Sources

Compare and Report Differences in	Catalogs E
Select Sources	Select Sources to Compare
Connection Options	
Connect to DBMS	Cource for Left Catalog
Select Schemata	O Model Schemata
Fetch Object Info	Ive Database Server
Connection Options	Script File:
Connect to DBMS	
Select Schemata	Source for Right Catalog
Fetch Object Info	Model Schemata
Differences Report	Live Database Server
	Script File: Browse
1911	Back Next Cancel

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:



Compare and Report Differences in	Catalogs	X
Select Sources	Differences Found in Catalog Comparison	
Connection Options		
Connect to DBMS	++ L Catalog Diff Benort	^
Select Schemata	Table `mudb` `table5` was dropped	
Fetch Object Info	Table 'mydb', 'table7' was dropped Table 'mydb', 'table7' was dropped Table 'mydb', 'table3' was modified	
Connection Options	foreign keys: - added foreign key fk_table3_table41 with columns: table4_idtable4, referred table: table4 with columns: idtable4	
Connect to DBMS	- action on update: NO ACTION - action on delete: NO ACTION	
Select Schemata	- removed foreign key: fk_table3_table41 	
Fetch Object Info	End of MySQL Workbench Report	
Differences Report	Save to File Copy to Clipboard	X
71/11	Back Close Cancel	

7.8. 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 7.8.2, "Using the Default Schema" and exploring the graphic design capabilities of MySQL Workbench is touched upon in Section 7.8.3, "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.

7.8.1. 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 <u>REVERSE ENGINEER MySQL</u> <u>CREATE SCRIPT</u> menu option by first selecting <u>FILE</u> and then <u>IMPORT</u>. Find and import the <u>sakila-schema.sql</u> file. This is the script that contains the data definition statements for the <u>sakila</u> database. The file filter for the file open dialog window defaults to *.sql so you should only be able to view files with the <u>sql</u> 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 7.7.9.1, "Reverse Engineering Using a Create Script".

7.8.1.1. 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 7.42. 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 7.45, "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 proper-

ties. 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 <u>SAVE</u> from the <u>FILE</u> menu or by using the keyboard command **Ctrl S**.

7.8.2. 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 7.43. 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 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.

7.8.2.1. 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.

7.8.2.2. 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 <u>SAVE</u> from the <u>FILE</u> menu or by using the keyboard command **Ctrl S**.

7.8.3. Basic Modeling

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

Figure 7.44. Adding an EER Diagram



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

7.8.3.1. 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 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 <u>EDIT IN NEW WINDOW</u> 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 table 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.

7.8.3.2. Create a Foreign Key

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

First click the invoice_item table; notice that a red border indicates that this table is selected. Next click 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 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 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 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 <u>SAVE</u> from the <u>FILE</u> menu or by using the keyboard command **Ctrl S**.

7.8.4. 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.

7.8.4.1. A PNG File of the sakila Database

Find below an EER digram showing all the tables in the sakila database. This image was created using the menu options <u>FILE</u>, <u>EXPORT</u>, <u>EXPORT</u> AS <u>PNG</u>

Figure 7.45. The sakila EER Diagram



The object notation style used in Figure 7.45, "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.

7.9. Printing

The printing options are used to create printouts of your EER Diagrams and are found under the <u>FILE</u> menu. For creating *documentation* of your models see Section 7.5.1.5.1, "The DBDoc Model Reporting Dialog Window (Commercial Version)".

7.9.1. Printing Options

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

• <u>PAGE SETUP</u>...

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

• <u>Print</u>

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.

• <u>PRINT TO PS ...</u>

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

7.10. 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 <u>MODEL</u> 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.

7.10.1. 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

7.10.2. 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

7.11. 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), IN- DICES (Detailed is true)
TABLES	REL_LISTING, INDICES_LISTING, COLUMNS_LISTING, TA- BLE_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 that for a dictionary to be involved in variable lookup, its associated section must currently be active.

So to achieve the desired output we would need the template to be something like the following:

Report

{#SCHEMATA}}
{#TABLES}}
{#TABLES}
{#COLUMNS_LISTING}}
Column Name: {{COLUMNS}}
{/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}}
Column Name: {{COLUMN_NAME}} {{! COLUMN_NAME variable is looked-up, and found, in COLUMNS data dictionary}}
{/COLUMNS}}
{/COLUMNS}}
{/COLUMNS}}
{/COLUMNS_}
{/CALHAN_}
```

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}}
{//COLUMNSLISTING}}
{//CALUMNS_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.

7.11.1. Supported Template Markers

A list of supported markers follows. These markers can be used in any template, including custom templates.

Marker text	Туре	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 Work- bench. This is typically used to load the correspond- ing CSS file, depending on the name of the style se- lected in MySQL Workbench.

Marker text	Туре	Data Dictionary defined in (if variable) or parent dictionary (if section)	Corresponding data
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 PROP-</u> <u>ERTIES</u> .
PROJECT_NAME	Variable	MAIN	Project name as set for the model in <u>DOCUMENT</u> <u>PROPERTIES</u> .
PROJECT_AUTHOR	Variable	MAIN	Project author as set for the model in <u>DOCUMENT</u> <u>PROPERTIES</u> .
PROJECT_VERSION	Variable	MAIN	Project version as set for the model in <u>DOCUMENT</u> <u>PROPERTIES</u> .
PROJECT_DESCRIPTI ON	Variable	MAIN	Project description as set for the model in <u>DOCU-</u> <u>MENT 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 dis- played in <u>DOCUMENT PROPERTIES</u> .
TOTAL_TABLE_COUN T	Variable	MAIN	Total number of tables in all schemata contained in the model.
TOTAL_COLUMN_CO UNT	Variable	MAIN	Total number of columns in all tables in all schemata in the model.
TOTAL_INDEX_COUN T	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_DEFAULTV ALUE	Variable	COLUMNS	The default value of entries in this column

Marker text	Туре	Data Dictionary defined in (if variable) or parent dictionary (if section)	Corresponding data
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_COLLATIO N	Variable	COLUMNS (if detailed)	The collation used by the column
COLUMN_IS_USERTY PE	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.
IN- DEX_COLUMN_NAME	Variable	INDEX_COLUMNS	The index column name
IN- DEX_COLUMN_ORDE R	Variable	INDEX_COLUMNS	The index column order e.g. ascending, descending
IN- DEX_COLUMN_COM MENT	Variable	INDEX_COLUMNS	A comment for the index column
IN- DEX_KEY_BLOCK_SI ZE	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

Marker text	Туре	Data Dictionary defined in (if variable) or parent dictionary (if section)	Corresponding data
FOREIGN_KEYS	Section	SCHEMATA	Marks the start and end of a FOREIGN_KEYS sec- tion. 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
TA- BLE_COMMENT_LISTI NG	Section	TABLES	Marks the start and end of a TA- BLE_COMMENT_LISTING section. The TA- BLE_COMMENT_LISTING data dictionary be- comes active in this section.
TABLE_COMMENT	Variable	TA- BLE_COMMENT_LISTI NG	A comment for the table
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 eg. 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.

7.11.2. 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:

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-A869-9864F83B88BDF}"
...
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>
<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">Struct-checksum="0xab08451b">
<value type="string" key="name">Custom Basic text report</value>
</value type="string" key="name">Custom Basic text report</value>
</value>
</value type="string" key="name">Custom TemplateStyleInfo"
id="{39e3b767-a832-4016-8753-b4cb93aa2dd6}" struct-checksum="0xab08451b">
</value>
</value type="string" key="name">Custom Text-checksum="0xab08451b">
</value>
</value type="string" key="name">Custom Text-checksum="0xab08451b">
</value>
</value type="string" key="name">Custom Text-checksum
```

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:

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 <u>MODEL</u>, <u>DBDOC - MODEL REPORTING</u> 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 using 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 using a server instance

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

Each of these action items is described in the following sections.

8.1. 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 Server Administration 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 the icon for the Server Instance you wish to access. A new Admin tab will be launched, which displays Server Status and Configuration.

For further details see the section Section 8.7, "Server Administration and Configuration".

8.2. New Server Instance

This action item enables 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.

For further details see the section Section 8.6.1, "New Server Instance Wizard".

8.3. 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 enables you to select a server instance to connect to.

For further details see the section Section 8.7.6, "Data Dump tab".

8.4. Manage Security

The purpose of this action item is to take you quickly to the screen that enables you to manage user accounts. It simply launches an Admin page and locates you on the Accounts tab.

For further details see the section Section 8.7.3, "Accounts tab".

8.5. 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 8.6, "Creating and Managing Server Instances".

8.6. 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.

8.6.1. New Server Instance Wizard

Clicking the New Server Instance action item launches the **CREATE A NEW SERVER INSTANCE** wizard. The wizard provides a stepby-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 enables you to enable SSH for the login to the server instance. This facility enables 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 enables you to select a connection to a specific database. The settings entered previously have been concerned with the connection to the server required for admnistrative 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 6.6, "Manage DB Connections Dialog". 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

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 enables you to set the commands required to start, stop and check the status of the running server instance. These felds 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.

8.6.2. Manage Server Instances Dialog

The Manage Server Instances dialog enables you to create, delete and manage server instances. The **CONNECTION** tab of the wizard enables 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

lanage Server Instances			
Server Instances	Instance Name: World		
Wond	Connection of the Cl		
	System Profile		
	MySQL Connection		
	Pick a preset connection to the MySQL server ins basic administration tasks such as managing user	tance. The connection will be used for s and viewing schema objects.	
	Connection: world - root@127.0.0.1:3306 <5	world - root@127.0.0.1:3306 <standard (tcp="" ip)=""> -</standard>	
		Edit Selected Manage Connections	
	MySQL Server Version:	Connect and Check	
	SSH Shell Connection		
	Enable SSH login based administration		
	SSH login based administration allows changing M it in remote machines.	ySQL configuration and starting/stopping	
	Hostname:	Port:	
	Username: mysql		
	Password: Store in Vault	Remove from Vault	
	↓ Authenticate Using SSH Ke	iy	
	SSH Key Path: ~/.ssh/ssh_private_key	Browse	
New Delete Dup	licate Move Up Move Down	Close	

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. Manage Server Instances Dialog
erver Instances	
World	Instance Name: World
	Connection System Profile
	Information about the server and MySQL configuration, such as path to the configuration file, command to start o stop it etc. You may pick a preset configuration profile or customize one for your needs.
	System Type: Windows
	Installation Type: Custom
	Configuration File: C:\Program Files\MySQL\MySQL Server 5.1\my.ini
	Instance Name: mysqld
	The following options specify how Workbench should perform certain administration actions. Commands given here are the same as if they would be run in a system shell. Commands which return a value might be combined with some predefined filters. See "Filters Help" vor more details.
	The following options specify how Workbench should perform certain administration actions. Commands given here are the same as if they would be run in a system shell. Commands which return a value might be combined with some predefined filters. See "Filters Help" vor more details.
	The following options specify how Workbench should perform certain administration actions. Commands given here are the same as if they would be run in a system shell. Commands which return a value might be combined with some predefined filters. See "Filters Help" vor more details. MySQL Management Server Stats Start MySQL : sc start mysql
	The following options specify how Workbench should perform certain administration actions. Commands given here are the same as if they would be run in a system shell. Commands which return a value might be combined with some predefined filters. See "Filters Help" vor more details. MySQL Management Server Stats Start MySQL: sc start mysql
	The following options specify how Workbench should perform certain administration actions. Commands given here are the same as if they would be run in a system shell. Commands which return a value might be combined with some predefined filters. See "Filters Help" vor more details. MySQL Management Server Stats Start MySQL: sc start mysql Stop MySQL: sc stop mysql
	The following options specify how Workbench should perform certain administration actions. Commands given here are the same as if they would be run in a system shell. Commands which return a value might be combined with some predefined filters. See "Filters Help" vor more details. MySQL Management Server Stats Start MySQL: sc start mysql Stop MySQL: sc stop mysql Image: Comparison of the store of the stor
	The following options specify how Workbench should perform certain administration actions. Commands given here are the same as if they would be run in a system shell. Commands which return a value might be combined with some predefined filters. See "Filters Help" vor more details. MySQL Management Server Stats Start MySQL: sc start mysql Stop MySQL: sc stop mysql Image: Check MySQL Status: sc query mysql findstr RUNNING
	The following options specify how Workbench should perform certain administration actions. Commands given here are the same as if they would be run in a system shell. Commands which return a value might be combined with some predefined filters. See "Filters Help" vor more details. MySQL Management Server Stats Start MySQL: sc start mysql Stop MySQL: sc stop mysql Image: Check MySQL Status: sc query mysql findstr RUNNING Image: Check MySQL Status: sc query mysql findstr rights to execute status commands
	The following options specify how Workbench should perform certain administration actions. Commands given here are the same as if they would be run in a system shell. Commands which return a value might be combined with some predefined filters. See "Filters Help" vor more details. MySQL Management Server Stats MySQL Management Server Stats Start MySQL: sc start mysql Stop MySQL: sc stop mysql Image: Check MySQL Status: sc query mysql findstr RUNNING Studo command: v
	The following options specify how Workbench should perform certain administration actions. Commands given here are the same as if they would be run in a system shell. Commands which return a value might be combined with some predefined filters. See "Filters Help" vor more details. MySQL Management Server Stats Start MySQL: sc start mysql Stop MySQL: sc stop mysql Image: Check MySQL Status: sc query mysql findstr RUNNING Check MySQL Status: sc query mysql findstr rights to execute status commands Sudo command: Image: Check status commands
	The following options specify how Workbench should perform certain administration actions. Commands given here are the same as if they would be run in a system shell. Commands which return a value might be combined with some predefined filters. See "Filters Help" vor more details. MySQL Management Server Stats Start MySQL: sc start mysql Stop MySQL: sc stop mysql Image: Check MySQL Status: sc query mysql findstr RUNNING Image: Check MySQL Status: sc query mysql findstr RUNNING Sudo command: Image: Check Status =
	The following options specify how Workbench should perform certain administration actions. Commands given here are the same as if they would be run in a system shell. Commands which return a value might be combined with some predefined filters. See "Filters Help" vor more details. MySQL Management Server Stats Start MySQL: sc start mysql Stop MySQL: sc stop mysql Image: Check MySQL Status: sc query mysql findstr RUNNING Check MySQL Status: sc query mysql findstr rights to execute status commands Sudo command: Image: Sudo commands

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**.

8.7. 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: Enables you to start and stop the MySQL server, and view the startup message log.
- Configuration: Enables you to view and edit the MySQL Configuration file (my.ini) using GUI controls.
- Accounts: Enables 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

MySQL Workbench	- • •
File Edit View Model Database Plugins Scripting Community Help	
Home MySQL Model (world.mwb) EER Diagram3 Admin (World) ×	
Server Status	•
INFO SYSTEM SERVER HEALTH	
Name: World Host: 127.0.0.1 Server: 5.1.40-community Status: Running CPU: 43% Mem: 41% Connection Usage: 2 Traffic: 7.14 KB/s Query Cache Hitrate: 0.00% Key Efficient	ncy: 0.00%
Configuration	
Startup Start/Stop Server Kark/Stop Server Vars Edit Configuration File A Accounts Manage Users Connection List Part Connection List Variables Status and Server Vars Export / Import Data	Logs Server Log Files
Database Server Status Startup Message Log	
The database server is started and ready for dient connections. To shut the Server 2010-04-07 16:40:55 - Checked server status: Server is running. 2010-04-07 16:40:56 - Server is running.	*
The database server instance is running Stop Server	
If you stop the server, you and your applications will not be able to use the Database	
and all current connections will be closed	
	-
	linhanud
Clear Messages Copy to C	lipboard
Ready	E .;;

8.7.1. 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

MySQL Workbench	
File Edit View Model Database Plugins Scripting Community Help	
Home MySQL Model (world.mwb) EER Diagram3 Admin (World) $ imes$	
Server Status	τ.
INFO SYSTEM SERVER HEALTH	
Name: World Host: 127.0.0.1 Server: 5.1.40-community Status: Running CPU: 43% Mem: 41% Connection Usage: 2	Traffic: 7.14 KB/s Query Cache Hitrate: 0.00% Key Efficiency: 0.00%
Configuration	
Startup Start/Stop Server Configuration File Accounts Connection Edit Configuration File Manage Users Connection	Ist Variables List Status and Server Vars Data Dump Export / Import Data Server Logs Server Log Files
Database Server Status	Startup Message Log
The database server is started and ready for client connections. To shut the Server down, use the "Stop Server" button	2010-04-07 16:40:55 - Checked server status: Server is running.
The database server instance is running Stop Server	
If you stop the server, you and your applications will not be able to use the Database	
and all current connections will be closed	
	_
	Clear Messages Copy to Clipboard
Ready	 [目]

8.7.2. Configuration tab

The configuration tab enables 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	- Config	guration	tab
-------------	---------------	----------	----------	-----

MySQL Workbench
File Edit View Model Database Plugins Scripting Community Help
Home Admin (World) ×
Server Status
INFO SYSTEM SERVER HEALTH
Host: 127.0.0.1 Server: 5.1.40-community Status: Running CPU: 54% Mem: 42% Connection Usage: 2 Traffic: 7.10 KB/s Query Cache Hitrate: 0.00% Key Efficiency: 0.00%
Configuration
Startup Start/Stop Server Vars Configuration File Accounts Manage Users Connections List Connection List Variables Data Dump Edit Configuration File Accounts Manage Users Connection List Variables Status and Server Vars Data Dump Export / Import Data
General Advanced MyISAM Parameters Performance Log Files Security InnoDB Parameters NDB Parameters Transactions Networking Replication Misc
Networking
TCP Port: 3306 Port number to use for connections.
Disable networking Don't allow connections via TCP/IP.
Enable named pipes: Enable support for named pipes.
Directories
Base directory: "C:/Program Files/MySQL/MySQL Serv Path to installation directory. All paths are usually resolved relative to this.
☑ Data directory: "C:/ProgramData/MySQL/MySQL Serv Path to the database root
Temp directory: Path to the temporary directory.
Memory usage
✓ Sort buffer size 256 K ✓ Each thread that needs to do a sort allocates a buffer of this size.
Configuration File: C:\Program Files\MySQL\MySQL Server 5.1\my.ini mysqld
WB Admin Opened

8.7.3. Accounts tab

The Accounts tab has two sub-tabs:

- Server Access Management
- Schema Privileges

SERVER ACCESS MANAGEMENT enables 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 enables you to set specific privileges on a user basis.

Figure 8.6. Administrator - Accounts tab

NySQL Workbench	
File Edit View Model Database	Plugins Scripting Community Help
1 🔁 🤌 📧 🖸	
Home Admin (World) ×	
Server Status	τ.
INFO Name: World Host: 127.0.01 Server: 5.1.40-community Status: Running	SYSTEM CPU: 32% Mem: 44% SERVER HEALTH Connection Usage: 2 Traffic: 7.10 KB/s Query Cache Hitrate: 0.00% Key Efficiency: 75.00%
Configuration	
Startup Start/Stop Server Start/Stop Server	ration File Accounts Manage Users Connections Edit Connection List Pariables Status and Server Vars Data Dump Export / Import Data Server Logs Server Log Files
Server Access Management Schema Privileges	
User Accounts	Details for Account root@localhost
User From Host	Login Administrative Roles Account Limits
newuser % root localhost	Login Name: root You may create multiple accounts with the same name to connect from differen
	Password: Type a password to reset it.
	Commin Password:
	Limit Connectivity to Hosts Matching: localhost % and _ wildcards may be used
Add Account Remove	Revoke All Privileges Revert Apply
WB Admin Opened	

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.

8.7.3.1. 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 the User Account you wish to modify, then click 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, REP-LICATION 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

8.7.4. Connections tab

This tab lists all current connections to the monitored server.

Figure 8.7. Administrator - Connections tab



8.7.5. Variables tab

The Variables tab displays a list of all server and status variables.



NySQL Workbench			
File Edit View Model D	atabase Plugins Scriptir	ng Community I	Help
1 🔁 📂 📧 🖸			
Home Admin (World) ×			
Server Status			τ.
INFO	SYSTEM	SERVER HEAL	тн
Name: World Host: 127.0.0.1 Server: 5.1.40-con Status: Running	nmunity CPU: 0% Mem: 4	41% Connection Usa	ge: 2 Traffic: 7.11 KB/s Query Cache Hitrate: 0.00% Key Efficiency: 75.00%
Configuration			
Startup Start/Stop Server	Configuration Edit Configuration File	Accounts Manage Users	Connections Edit Connection List Variables Status and Server Vars Data Dump Export / Import Data Construction Files
Status Variables System Variables			
	Name	Value	Description
	Aborted_clients	2	The number of connections aborted because the client died without closing the connection
	Aborted_connects	0	The number of failed attempts to connect to the MySQL server
All Search Results	Binlog_cache_use	0	The number of transactions that used the temporary hinary log cache.
Binlog	Bytes received	5626	The number of bytes received from all clients
Replication	Bytes_sent	1239962	The number of bytes sent to all clients
Ndb	Com_admin_commands	0	Count of admin commands
Server	Com_assign_to_keycadhe	0	Com_assign_to_keycadhe
Merge	Com_alter_db	0	Count of ALTER DATABASE statements
Innodb	Com_alter_db_upgrade	0	Com_alter_db_upgrade
	Com_alter_event	0	Count for ALIER EVENT statements
	Com_alter_runction	0	Com alter procedure
	Com alter server	õ	Com alter server
	Com_alter_table	0	Count of ALTER TABLE statements
	Com_alter_tablespace	0	Com_alter_tablespace
	Com_analyze	0	Count of ANALYZE statements
			Refresh
WB Admin Opened			

8.7.6. Data Dump tab

The Import/Export Server Data tab enables 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

8.7.6.1. Export to Disk

This tab enables 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 enables you to restore on a per-table basis.

Figure 8.9. Administrator - Export to Disk

MySQL Workbench		
File Edit View Model Database F	Plugins Scripting Community Help	
Home Admin (World) ×		
Server Status		▼.
INFO Name: World	SYSTEM SERVER HEALTH	
Host: 127.00.1 Host: 127.00.1 Server: 5.1.40-community Status: Running	CPU: 24% Mem: 42% Connection Usage: 2	Traffic: 7.11 KB/s Query Cache Hitrate: 0.00% Key Efficiency: 75.00%
Configuration		
Startup Start/Stop Server Start/Stop Server	tion File Accounts Manage Users Cor Edit	Connection List Variables Status and Server Vars Data Dump Export / Import Data Server Log Files
Export to Disk Import from Disk Advanced Expo	rt Options	
Select Database Objects to Export		Options
Export Schema	Export Tables in Schema	 Export to Backup Project Folder
dvd_collection mvdb	city country	Each table will be exported into a separate file. This allows a selective restore.
mysql	Countrylanguage	Folder Path C: \Users\Tony\Documents/dumps\Dump20100407
world		© Export to Self-Contained File
		All selected database objects will be exported into a single, self-contained file.
		File Path C:\Users\Tony\Documents/dumps\Dump20100407.sql
		Create Dump in a Single Transaction
4		Dump Views
Refresh	Select All Tables Unselect All	Dump stored routines (procedures and functions)
	. –	
Press [Start Export] to start	Log:	A
WB Admin Opened		,,, ,,, B

8.7.6.2. Import from Disk

This tab enables 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



8.7.6.3. 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

MySQL Workbench	×		
File Edit View Model Database Plugins Scripting Community Help			
Home Admin (World) ×			
Server Status	•		
INFO Name: World Host: 127.0.0.1 Server: 5.1.40-community Status: Running Status: Running Status: Running Status: Running Status: Running Status: Running Status: Running Status: Status: Stat			
Configuration			
Startup Start/Stop Server 🖗 Configuration File & Accounts Edit Configuration File & Manage Users & Connections Edit Connection List & Xartus and Server Vars & Data Dump Startus and Server Vars	Files		
Export to Disk Import from Disk Advanced Export Options			
Inserts			
In add-locks - Surround each table dump with LOCK TABLES and UNLOCK TABLES statements.			
Complete-insert - Use complete INSERT statements that include column names.			
☑ extended-insert - Use multiple-row INSERT syntax that include several VALUES lists.	Ξ		
insert-ignore - Write INSERT IGNORE statements rather than INSERT statements.			
replace - Write REPLACE statements rather than INSERT statements.			
delayed-insert - Write INSERT DELAYED statements rather than INSERT statements.			
Other			
events - Dump events from the dumped databases.			
☑ dump-date - Include dump date as "Dump completed on" comment ifcomments is given.			
In shi-logs - Flush the MySQL server log files before starting the dump.			
C delete-master-logs - On a master replication server, delete the binary logs after performing the dump operation.			
🔲 no-data - Do not dump table contents.	Ŧ		
- WB Admin Opened	8 ;		

8.7.7. 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. Extending Workbench

MySQL Workbench provides an extension and scripting system that allows the developer to extend MySQL Workbench according to requirements. While the core of MySQL Workbench is developed using C++, it is possible to harness this core functionality using both the Lua and Python scripting languages. MySQL Workbench also provides access to a cross-platform GUI library, mforms, which enables the creation of extensions that feature a graphical user interface.

The extension capabilities enable the following:

- Automate common tasks
- Extend the Workbench user-interface
- Create plugins (code which can be invoked from the Workbench menu system)
- · Manipulate schemata
- Create custom Workbench features

9.1. GRT and Workbench Data Organization

The GRT, or Generic RunTime, is the internal system used by Workbench to hold model document data. It is also the mechanism by which Workbench can interact with Modules and Plugins. Workbench model data, such as diagrams, schemata and tables, is stored in a hierarchy of objects that can be accessed by any plugin. The data is represented using standard data types: integers, doubles, strings, dicts, lists and objects.

The GRT can be accessed using external scripting languages such as Lua and Python. Awareness is required of how the GRT data types map into the scripting language. In Python, for example, the GRT data types integers, doubles and strings are seen as corresponding Python data types. Lists and dicts are kept in their internal representation, but can however generally be treated as Python lists and dicts, and accessed in the usual way. Objects contain data fields and methods, but the GRT only recognizes objects from a pre-registered class hierarchy.

It is possible to fully examine the classes contained within the GRT using the Workbench Scripting Shell. Note that dots in class names are changed to underscores in their Python counterparts. For example, db.mysql.Table becomes db_mysql_Table in Python.

The Application Objects Tree (GRT Tree)

As mentioned previously the Workbench document data is stored in an object hierarchy. This hierarchy is known as the GRT Tree. The GRT Tree can be accessed and modified from supported external scripting languages such as Python. Care should be taken when modifiying the GRT Tree, in case a mistake leads to corruption of the document. Backups should be made before manipulating the tree. Read-only access to the tree is the safest approach, and is sufficient in most cases.

The main nodes in the Application Object Tree

Node	Description
wb.registry	Application data such as plugin registry, list of editors, and options.
wb.customData	A generic dictionary for data you can use to store your own data. This dictionary is saved and reloaded with Workbench and is global (not document specific).
wb.options	Contains some default options that are used by Workbench
wb.rdbmsMgmt	Internal registry of supported RDBMS modules, known data types.
wb.doc	The currently loaded model document.
wb.doc.physicalModels[0]	The currently loaded model object, containing the database catalog and dia- grams.
wb.doc.physicalModels[0].catalog	The database catalog for the model. Contains the list of schemata.
wb.doc.physicalModels[0]catalog.schemata	List of schemata in the model. Individual schema can be accessed as a list: schemata[0], schemata[1]
wb.doc.physicalModels[0].catalog.schemata[0].table s (.views, .routines,)	Lists of tables, views, routines in the schema.

Node	Description
wb.doc.physicalModels[0].diagrams	List of EER diagrams in the model
wb.doc.physicalModels[0].diagrams[0].figures (.layers, .connections,)	List of figures, layers, connections (relationships) in the diagram.

9.2. Modules

In the GRT Modules are libraries containing a list of functions that are exported for use by code in other modules, scripts or Workbench itself. Modules can currently be written in C++, Lua or Python, but the data types used for arguments and the return value must be GRT types.

GRT modules are similar to Python modules. They are however imported from the built-in grt module, instead of directly from an external file. The list of modules loaded into the grt module is obtained grt.modules. Modules can be imported in Python using statements such as from grt.modules import WbModel.

To export functions as a module from Python code you need to carry out the following:

- 1. The source file must be located in the user modules folder. This path is displayed in the Workbench Scripting Shell with the label LOOKING FOR USER PLUGINS IN.... It is also possible to install the file using the main menu item <u>SCRIPTING</u>, <u>INSTALL</u> <u>PLUGIN/MODULE FILE</u>.
- 2. The source file name must have the extension _grt.py, for example, my_module_grt.py.
- 3. Some module metadata needs to be defined. This can be done using the DefineModule function from the wb module:

```
from wb import *
ModuleInfo = DefineModule(name='MyModule', author='Your Name', version='1.0')
```

4. Functions to be exported require their signature to be declared. This is achieved using the export decorator in the previously created ModuleInfo object:

```
@ModuleInfo.export(grt.INT, grt.STRING)
def checkString(s):
```

Note, for the export statement, the return type is listed first, followed by the input parameter types. These types are specified as GRT typenames. The typenames that can be used are as follows:

- grt.INT: Integer values. Also used for boolean values.
- grt.DOUBLE: Floating-point numeric values.
- grt.STRING: UTF-8 or ASCII string data.
- grt.DICT: A key/value dictionary item. Keys must be strings.
- grt.LIST: A list of other values. It is possible to specify the type of the contents as a tuple in the form (grt.LIST, <type-or-class>). For example, (grt.LIST, grt.STRING) for a list of strings. For a list of table objects the following would be specified: (grt.LIST, grt.classes.db_table).
- grt.OBJECT: An instance of a GRT object or a GRT class object, from grt.classes.

Note these types are defined in the grt module, which must first be imported before they can be used.

The following code snippet illustrates declaring a module that exports a single function:

```
from wb import *
import grt
ModuleInfo = DefineModule(name='MyModule', author="your name", version='1.0')
@ModuleInfo.export(grt.DOUBLE, grt.STRING, (grt.LIST, grt.DOUBLE))
def printListSum(message, doubleList):
```

sum = 0
for d in doubleList:
 sum = sum + d
print message, sum
return sum

9.3. Plugins

Plugins are special Modules that are exposed to the user through the Workbench GUI. This is typically done using the main menu, or the context-sensitive menu. Much of the MySQL Workbench functionality is implemented using plugins, for example, tabel, view, and routine editors are native C++ plugins, as are the forward and reverse engineering wizards. The Administrator facility in MySQL Workbench is implemented entirely as a plugin in Python.

A plugin can be a simple function that performs some action on an input, and ends without further interaction with the user. Examples of this include auto-arranging a diagram, or making batch changes to objects. To create a simple plugin, the function needs to be located in a module and then declare it as a plugin using the plugin decorator of the ModuleInfo object.

Plugins can have an indefinite runtime, such as when they are driven by the user through a graphical user interface. This is the case for the various object editors and wizards within MySQL Workbench. Although this latter type of plugin will need to be declared in the usual way, only the entry point of the plugin will need to be executed in the plugin function, as most of the additional functionality will be invoked as a result of the user interacting with the GUI.

The syntax for declaring a plugin is as follows:

@ModuleInfo.plugin(plugin_name, caption, [input], [groups], [pluginMenu])

These parameters are defined as follows:

- plugin_name: A unique name for the plugin. It may only contain alphanumeric characters, dots, and underscores.
- caption: A caption to use for the plugin in menus.
- input: An optional list of input arguments.
- groups: Optional list of groups the plugin belongs to. Recognized values are:
 - Overview/Utility: The <u>CONTEXT</u> menu in the Model Overview.
 - Model/Utility: The menu for diagram objects.
 - Menu/<category>: The <u>PLUGINS</u> menu in the main menu.
- pluginMenu: Optional name of a submenu in the Plugins menu where the plugin should appear. For example, <u>CATALOG</u>, <u>OBJECTS</u>, <u>UTHITIES</u>. This is equivalent to adding a <u>Menu/<category></u> in the groups list.

9.4. Adding a GUI to a Plugin using MForms

MySQL Workbench is implemented with a C++ core back-end, and a native front-end for each supported platform. Currently the frontend is implemented with Windows Forms on Microsoft Windows, GTK+ on Linux, and Cocoa on Mac OS X. This approach allows the application to have a native look and feel, while reducing the amount of work required to maintain the project. However, the GUI functionality required by MySQL Workbench can be met by a subset of graphical operations. These are implemented in a cross-platform GUI library, MForms. This further reduces the development effort, as plugin developers can use MForms, rather than having to write front-end specific code for each supported platform. This also helps consistency of operation across all platforms. MForms is coded in C++, but provides a Python interface. To use it the Python code needs to import the mforms module.

MForms Containers

Given problems of using an absolute co-ordinate system across different platforms, MForms employs containers that perform automatic layout. The basic containers that MForms provides include:

• **Form**: A top-level window which can contain a single control, usually another container. The window will be sized automatically to fit its contents, but can also be sized statically.

- **Box**: This is a container that can be filled with one or more controls in a vertical or horizontal layout. Each child control can be set to use either the minimum of required space, or fill the box in the direction of the layout. In the direction perpendicular to the layout, for example vertical in a horizontal layout, the smallest possible size that can accommodate all child controls will be employed. So, in this example, the smallest height possible to accommodate the controls would be used.
- **Table**: This is a container that can organize one or more controls in a grid. The number of rows and columns in the table, and the location of controls within the grid, can be set by the developer.
- ScrollView: This is a container that can contain a single child control, and will add scrollbars if the contents do not fit the available space.

9.5. The Workbench Scripting Shell

The Workbench Scripting Shell provides a means for entering and executing scripts. Through the use of the scripting shell, MySQL Workbench can support new behavior and data sources using code written in Lua and Python. The shell can also be used to explore the current Workbench GRT (Generic Runtime) facilities.

The scripting shell is not only useful for expanding MySQL Workbench. You can use a script file from the scripting shell command line to perform repetitive tasks programmatically.

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 <u>EDIT</u>, <u>PREFERENCES</u>.

9.5.1. Exploring the Workbench Scripting Shell

To open the Workbench Scripting Shell, select <u>SCRIPTING</u>, <u>SCRIPTING SHELL</u> 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

Workbench Scripting Shell		
Globals Classes Modules	Shell	
/	Starting WorkbenchRegistered 147 GRT dasses. Debug level - 0 Registered 18 modules (from 15 files). Looking for user plugins in C:\Users\Tony\AppData\Roaming\MySQL\Workbench\modules MySQL Generic Runtime Environment 4.1.0 Type 'help' or ?! for help. Type 'quit' to exit the shell. Python Shell initialized. Ready.	*
/wb/rdbmsMgmt		
Name Value		
<pre>datatypeG <list> o name o owner <> filt rdbms <list> filt rdbms <list> storedCon <list> filt storedInst filt storedInst filt storedInst </list></list></list></list></list></list></list></list></list></list></list></list></list></list></list></list></list></list></pre>		Ŧ

9.5.2. The Shell Window

The Workbench Scripting Shell is primarily used for running Lua or Python scripts or typing commands in these languages directly. However, you can also use it to 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.

While individual commands can be entered into the shell, it is also possible to run a longer script, stored in an external file, using the main menu option <u>SCRIPTING</u>, <u>RUN WORKBENCH SCRIPT FILE</u>. When scripts are run outside of the shell, to see the output use the main menu item <u>VIEW</u>, <u>OUTPUT</u>.

It is also possible to run script files directly from the shell. For details 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.
```

Within the Workbench Scripting Shell, on the left side panel, are three tabs: GLOBALS, CLASSES, and MODULES. Discussion of these additional tabs follows.

9.5.3. The Globals, Classes, and Modules Tabs

The Workbench Scripting Shell features the GLOBALS, CLASSES and MODULES tabs, in addition to the main SHELL tab.

The Globals Tab

At the top of the window is a drop down list box that is used to select the starting point, or root, of the GRT Globals tree displayed beneath it. By default this starting point is the root of the tree, that is, '/'. The GRT Globals tree can be expanded and collapsed as required. The GRT Globals tree is the structure in which MySQL Workbench stores document data. Clicking any item will result in its name and value being displayed in the panel below the tree.

The Classes Tab

A class is a user-defined data type formed by combining primitive data types: integers, doubles, strings, dicts, lists, and objects. This tab shows the definitions of the classes used by the objects in the Modules tab. Clicking a class causes a brief description of the class to be displayed in a panel below the classes explorer.

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 **GROUP 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 GrtObject: the parent object from which all other objects are derived.

The Modules Tab

The **MODULES** tab enables you to browse the MySQL Workbench installed modules and their functions. Clicking a module within the explorer causes its details to be displayed in a panel below the explorer. This facility is useful for exploring the available modules, and their supported functions. It is also a way to check if custom modules have been correctly installed.

9.6. Tutorial: Writing Plugins

The purpose of this tutorial is to show you how to extend MySQL Workbench by creating a plugin.

The sample plugin

EER Diagrams are useful for visualizing complex database schemata. They are often created for existing databases, to clarify their purpose or document them. MySQL Workbench provides facilities for reverse engineering existing databases, and then creating an EER Diagram automatically. In this case relationship lines between foreign keys in the table will automatically be drawn. This graphical representation makes the relationships between the tables much easier to understand. However, one of the most popular storage engines for MySQL, MyISAM, does not include support for foreign keys. This means that MyISAM tables that are reverse engineered will not automatically have the relationship lines drawn between tables, making the database harder to understand. The plugin that will be created in this tutorial gets around this problem by using the fact that a naming convention is very often used for foreign keys: table-name_primarykeyname. Using this convention, foreign keys can automatically be created after a database is reverse engineered, which will result in relationship lines being drawn in the EER diagram.

Algorithm

The basic algorithm for this task would be as follows:

```
for each table in the schema
  for each column in the table
    look for another table whose name and primary key name match the current column name
    if such a table is found, add a foreign key referencing it
```

As iterating the complete table list to find a match can be slow for models with a large number of tables, it is necessary to optimize by pre-computing all possible foreign key names in a given schema.

auto_create_fks(grt.root.wb.doc.physicalModels[0].catalog.schemata[0])

Creating a Plugin from a Script

To create a plugin from an arbitrary script, it is first necessary to make the file a module, and export the required function from it. It is then necessary to declare the module as a plugin, and specify the return type and input arguments.

```
from wb import *
import grt
ModuleInfo = DefineModule(name="AutoFK", author="John Doe", version="1.0")
@ModuleInfo.plugin("sample.createGuessedForeignKeys", caption="Create Foreign Keys from ColumnNames",
input=[wbinputs.objectOfClass("db.mysql.schema")], groups=["Overview/Utility"])
@ModuleInfo.export(grt.INT, grt.classes.db_mysql_Schema)
def auto_create_fks(schema):
...
```

With the addition of the above code, the auto_create_fks() function is exported and will be added to the schema context menu in the model overview. When invoked it will receive the currently selected schema as its input.

Chapter 10. 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	Modifer+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
Сору	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 19)
Go to Marker n	EER Diagram	Modifier+n (n is integer 19)

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	Н
Delete tool	D

Function	Keyboard Shortcut
Layer tool	L
Note tool	N
Image tool	Ι
Table tool	Т
View tool	V
Routine Group tool	G
Non-Identfying Relationship 1:1	1
Non-Identfying 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 11. MySQL Workbench FAQ

Frequently Asked Questions with answers.

Questions

11.1: When a model is exported using the main menu item <u>FILE, EXPORT, FORWARD ENGINEER SQL CREATE SCRIPT</u>, 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, 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;
```

- 11.2: MySQL Workbench 5.0 appears to run slowly. How can I increase performance?
- 11.3: I get errors when creating or placing objects on an EER Diagram. I am using OpenGL rendering, AMD processor, and ATI graphics hardware.

Questions and Answers

11.1: When a model is exported using the main menu item <u>FILE, EXPORT, FORWARD ENGINEER SQL CREATE SCRIPT</u>, 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, 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;
```

11.2: 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 <u>TOOLS</u>, <u>OPTIONS</u>, **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 <u>TOOLS</u>, <u>OPTIONS</u>, **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 backported to 5.0.

11.3: I get errors when creating or placing objects on an EER Diagram. I am using OpenGL rendering, AMD processor, and ATI graphics hardware.

To solve this problem renew the ATI drivers pack, which can be downloaded from the AMD website.

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 5.2.30 (Not yet released GA)

Fifth GA release of 5.2. This section documents all changes and bug fixes that have been applied since the release of 5.2.29.

Bugs fixed:

• In the **REVERSE ENGINEER SQL SCRIPT** wizard, on the **INPUT AND OPTIONS** page, the button to select the SQL script file had the text "Brow" instead of "Browse". (Bug#57929)

A.1.2. Changes in MySQL Workbench 5.2.29 (12 October 2010 GA)

Fourth GA release of 5.2. This section documents all changes and bug fixes that have been applied since the release of 5.2.28.

Functionality added or changed:

• An open model that has not been saved will automatically be saved after this period. On loading a model file, MySQL Workbench will notify the user if the file was not previously saved correctly, due to a crash or power failure. MySQL Workbench can then attempt to recover the last auto-saved version. Note that for automatic recovery to be available for a new file, it will have to have been saved at least once by the user. (Bug#56718)

Bugs fixed:

- Clicking on the INSERTS tab of the Table Editor resulted in the MYSQL WORKBENCH UNEXPECTED ERROR dialog being displayed. (Bug#57346)
- If an EER diagram was created, and then closed, attempting to reopen it would cause MySQL Workbench to display an unexpected error dialog. (Bug#57259)
- Attempting to open the Table Editor in the SQL Editor multiple times for a table that contained no primary key caused MySQL Workbench to crash. (Bug#57139)
- Changes made to a server instance were not saved. In particular changing installation type to "Custom" was not saved, and changes to the sudo command were also not saved. (Bug#57026)
- When attempting to open an EER Diagram by double-clicking it in the Model Overview, MySQL Workbench generated the following unexpected error:

Cannot access a disposed object. Object Name: ModelDiagramForm.

(Bug#56963)

- When synchronizing a model to a live database, if a database was deselected in the wizard, to prevent it from being part of the synchronization process, a DROP statement for that database was generated in the SQL code, causing that database to be dropped from the live database, with corresponding loss of all data. (Bug#56938)
- In the ALTER TABLE dialog of the SQL Editor, attempting to create a new index in the INDEX tab was not actioned when the APPLY button was clicked. The dialog reported that "no changes to object were detected". (Bug#56906)
- It was only possible to connect to a MySQL server for administration using an SSH connection. (Bug#56874)
- If a table containing a User Defined Type based on VARCHAR was forward engineered, the resulting SQL did not correctly quote the

VARCHAR value, resulting in a server error. For example, the following code was generated:

CREATE TABLE IF NOT EXISTS `x`.`tabley` (`customstring` VARCHAR(10)); INSERT INTO `x`.`tabley` (`id_customstring`) VALUES (example);

The correct **INSERT** statement should be:

INSERT INTO `x`.`tabley` (`id_customstring`) VALUES ('example');

(Bug#56710)

- In the Administrator, clicking on the DATA DUMP tab generated the error Error updating DB: float division and schemata were not displayed. (Bug#56658)
- In the SQL Editor, if some table data was edited, and then the connection to the server was lost, clicking the APPLY CHANGES TO DATA toolbar button caused MySQL Workbench to crash.

MySQL Workbench has been changed so that the **OUTPUT** tab now shows an error message when a connection is lost. Once the connection is back up, clicking APPLY CHANGES TO DATA will re-establish the connection and send the changes to the server. (Bug#56170)

- Screen areas in the Administrator did not redraw correctly, it was necessary to move the main window in order to force a redraw. (Bug#55490)
- Folding of iteration statement blocks, such as WHILE, REPEAT and LOOP, did not work correctly in the SQL Editor. The loop end constructs such as END WHILE, END REPEAT and END LOOP were not recognized as expected. (Bug#55358)
- If a foreign key was set to RESTRICT for On Delete and On Update on the Foreign Keys tab of the Table Editor, MySQL Workbench generated SQL code to drop and add this key on every subsequent synchronization attempt. (Bug#55155)

A.1.3. Changes in MySQL Workbench 5.2.28 (19 September 2010 GA)

Third GA release of 5.2. This section documents all changes and bug fixes that have been applied since the release of 5.2.27.

Functionality added or changed:

- The following command line options have been added to MySQL Workbench:
 - --admin instance Launch MySQL Workbench and load the server instance specified.
 - --query connection Launch MySQL Workbench and load the connection specified.
 - --model modelfile Launch MySQL Workbench and load the model specified.
 - --script script Launch MySQL Workbench and run the script specified.
 - --run code Launch MySQL Workbench and run the code snippet specified.
 - --quit-when-done quits MySQL Workbench after --script or --run finishes. (Bug#46340)

Bugs fixed:

• Importing an exported file from a network share resulted in the following error due to incorrect escaping of the path:

```
08:44:00 Restoring \\server\d$\file.sql
Running: mysql.exe --defaults-extra-file="c:\...\tmp3oc8ig" --host=localhost
--user=root --port=3306 --default-character-set=utf8 --comments <
"\\\server\\d\$\\file.sql"
The specified path is invalid.</pre>
```

(Bug#56735)

- In the SQL Editor, a crash occurred if <u>USER SNIPPETS</u> was selected after another snippet collection had been selected, and then the REPLACE ACTIVE SQL EDITOR CONTENTS WITH SELECTED SNIPPET button was clicked. (Bug#56626)
- When running a TRUNCATE command in the SQL editor with 'safe updates' enabled, the following error was produced:

```
Error Code: 1175
You are using safe update mode and you tried to update a table without a WHERE that uses
a KEY column
```

(Bug#56597)

- In the ACCOUNTS tab of the Administrator, an UNEXPECTED ERROR occurred if a foreign character was used in the LOGIN NAME field of the SERVER ACCESS MANAGEMENT tab. The same bug also affected the SCHEMA PRIVILEGES tab. (Bug#56534)
- The Table tab of the Table Editor contained a typo in the description for the NAME field. Replaced was spelled as 'raplaced'. (Bug#56518)
- While attempting to define a foreign key for a table with a composite primary key, MySQL Workbench crashed. This happened while trying to select the fields of the composite primary key to be associated with the referenced columns. (Bug#56496)
- If a foreign key was about to be renamed, and the corresponding text field had focus, MySQL Workbench crashed if the model was then closed. (Bug#56315)
- In the SQL Editor, a statement containing CREATE DATABASE resulted in an unexpected error. (Bug#56295)
- The mydb database was created by default when reverse engineering from an existing database or script. (Bug#56183)
- If a SQL script containing international characters was saved in the SQL Editor, and then restored, the last character in the restored file was missing. (Bug#56083)
- In MySQL Workbench list controls, when an item was selected, and another item right-clicked, the context-sensitive menu displayed options related to the first list item selected, rather than the current list item being right-clicked, and any action selected was carried out on the first item. (Bug#56077)
- The Workbench Scripting Shell dialog could be increased in size, but could not subsequently be reduced in size. (Bug#56049)
- There were two problems in the NEW_TABLE dialog of the SQL Editor:
 - 1. When APPLY was clicked, in the **APPLY SQL SCRIPT TO DATABASE** dialog, if APPLY SQL was clicked, the SQL would be applied without error. However, if the BACK button was clicked and the APPLY SQL button clicked again, an error would be generated. If this process was repeated, each time a slightly different error would be added to the list of errors displayed.
 - The CANCEL and FINISH buttons appeared to have the same functionality, and simply returned the user to the NEW_TABLE dialog.
 (Bug#56000)
- If a model contained two schemata, and an attempt was made to forward engineer only one of the schemata by use of the object filters, then data loss could result. This happened because a DROP statement was generated for both schemata in the model, rather than just the one being forward engineered. (Bug#55918)
- When OpenGL rendering was selected, the EER Diagram area was not redrawn due to faulty OpenGL drivers.

The work around was to launch MySQL Workbench using the -swrendering command line option.

MySQL Workbench has been changed to also include an option to select software rendering in the **WORKBENCH PREFERENCES** dialog. (Bug#55863)

• In the View Editor, if the name of a view was changed in the code editor to the name of a pre-existing view of that name, the contents of the pre-existing view were replaced by the contents of the new view, even though the new view was created with a different name. (Bug#55661)

- MySQL Workbench application would not start on Mac OS X. (Bug#55412)
- MySQL Workbench exhibited a variety of crashes if, in the Model View, EER diagrams were repeatedly created and deleted. MySQL Workbench also crashed if all objects in the diagram were copied and pasted to the same diagram several times, and then the diagram deleted. Unexpected errors generated included:
 - 1. Object is currently in use elsewhere
 - 2. Index is outside the bounds of the array
 - 3. Index must be within the bounds of the List

(Bug#55304)

• A plugin which used Mforms generated the following error when MySQL Workbench started:

.... line 16, in <module>
import mforms
ImportError: No module named mforms

(Bug#55243)

• If the MySQL server configuration file was modified using the Administrator to deactivate InnoDB, the MySQL server failed to restart, and generated the following error in the log:

[ERROR] C:\Program Files\MySQL\MySQL Server 5.1\bin\mysqld: Error while setting value 'FALSE' to 'innodb'

(Bug#55029)

- It was no longer possible to rename a schema by double-clicking on the schema tab in the Physical Schemata section of the Model View. (Bug#55010)
- If a view was created (for example view1) for an existing schema, with correct syntax, and then <u>MODEL</u> menu, VALIDATE (SQL) was selected, and either VALIDATE ALL or CHECK SYNTAX was selected, then the following error message was generated in the OUTPUT window:

Syntax error in view 'view1'. View code is 'CREATE VIEW'

(Bug#54969)

• On Microsoft Windows, MySQL Workbench crashed if the Trebuchet MS font was missing or corrupted.

MySQL Workbench has been changed to fall back to a secondary font if the Trebuchet MS font is missing or corrupted. (Bug#54953)

- If all objects were selected in the model view, and then the menu item EDIT SELECTED selected, MySQL Workbench generated an unexpected error. (Bug#54695)
- On Microsoft Windows, when saving a MySQL configuration file from within the Administrator, Windows displayed the UAC dialog twice.

MySQL Workbench has been changed so that Windows now only displays the UAC dialog once. (Bug#49674)

• The SQL Editor did not display the number of rows matched after a query.

MySQL Workbench has been updated to ensure this information is displayed. (Bug#48772)

A.1.4. Changes in MySQL Workbench 5.2.27 (01 September 2010 GA)

Third GA release of 5.2. This section documents all changes and bug fixes that have been applied since the release of 5.2.26.

Functionality added or changed:

- The Sychronize Model With Database wizard has been changed recently to show all schemata in a model, regardless of whether they exist in the live database or not. In addition, the following changes have been made:
 - 1. The wizard shows a list of schemata that exist in the model but are missing from the live server, with a note recommending the use of the Forward Engineering wizard.
 - 2. The wizard does not auto-select schemata names that are in the model but not in the live database.

(Bug#54748)

Bugs fixed:

• While attempting to restore a backup of selected tables, the following error was generated:

```
File "C:\Program Files\MySQL\MySQL Workbench 5.2 CE\modules\wb_admin_export.py", line
841, in start
    logmsg = "Restoring %s (%s)" % (schema, table)
NameError: global name 'schema' is not defined
```

(Bug#56020)

- Forward engineering a model to a live database crashed MySQL Workbench. (Bug#56009)
- Dump tab did not display the target folder before the dump was performed. (Bug#55945)
- MySQL Workbench crashed when using the foreign key editor. The Microsoft Visual C++ Runtime Library runtime error generated prior to the crash was:

R6025 - pure virtual function call

(Bug#55910)

• If the sidebar was placed on the right hand side, using the option in Workbench Preferences, then when **ADD DIAGRAM** was doubleclicked then the following Unexpected Error was generated:

SplitterDistance value is invalid (-52)

(Bug#55902)

- In the SQL Editor, executing statements designed for MySQL Server version 5.5 generated syntax highlighting errors. (Bug#55891)
- In the SQL Editor, executing statements designed for MySQL Server version 5.5 generated syntax errors. (Bug#55867)
- After dropping a schema from the list in the Object Browser, the schema that was dropped remained, but with the name of the schema that came after it. When the list of available schemata was refreshed, an unhandled exception was generated:

```
An unhandled exception has occurred:
Invalid node index
```

(Bug#55861)

• If "Dump views" was selected for a backup, the following error was generated:

Unhandled exception: global name 'views_by_schema' is not defined

It was necessary to restart the Administrator in order to enable the dump facility again. (Bug#55833)

 In the SQL Editor, right-clicking on a routine name in the Object Browser caused MySQL Workbench to exit unexpectedly. (Bug#55822)

- MySQL Workbench crashed when using the PARTITIONING tab of the Table Editor. (Bug#55761)
- If a table was copied in the EER Diagram view, and then the model synchronized with the live database, the synchronize functionality did not recognize the copied table as a new table, and overwrote the original table. (Bug#55566)
- MySQL Workbench crashed when using the **EXPORT RECORDSET TO AN EXTERNAL FILE** command button on the **RESULT** pane of the SQL Editor. (Bug#55554)
- Renaming a column with a foreign key resulted in an invalid script being generated on forward engineering or synchronization with a live database. (Bug#55502)
- On the second attempt to save a read-only model file, the following error was generated:

Couldn't backup existing file name-of-the-file.mwb: No such file or directory

(Bug#55439)

- When a table was being edited in the Table Editor, and its name changed, the change of name was not reflected on the Table Editor's table tab. (Bug#55338)
- It was not possible to add LOCK TABLES privilege when assigning a new role in the Model Overview, as that privilege was missing from the available options. (Bug#55186)
- The schema collation was not saved after being changed in the Table Editor. (Bug#55152)
- When an attempt was made to synchronize a model with a live database, no schemata were listed as available in the Synchronize Model with Database wizard. This happened when the name of the schema in the model and the schema on the live server were only differentiated by case. (Bug#55147)
- When using MySQL Workbench to report a bug, it was not possible to attach a file using the CHOOSE FILE button. When the button was clicked, it had no effect. (Bug#55112)
- It was not possible to dump tables or schemata without the LOCK TABLE privilege. The following error was generated:

mysqldump: Got error: 1044: Access denied for user 'wwroot'@'%' to database db' when doing LOCK TABLES

(Bug#55019)

Selecting <u>VALIDATE (SQL)</u> from the Model menu and then selecting the <u>VALIDATE ALL</u> option reported no errors. Selecting <u>FOR-WARD ENGINEER</u> from the <u>DATABASE</u> menu and selecting **RUN VALIDATIONS** also reported no errors. However, executing the SQL script caused the following error:

ERROR: Error 1005: Can't create table 'xxx' (errno:150)

(Bug#54985)

• If the <u>CHECK DUPLICATED IDENTIFIERS (ADDITIONS)</u> operation was selected from the <u>VALIDATION (SQL)</u> submenu of the <u>MODEL</u> menu then the following error message was displayed:

Type mismatch: expected object of type db.Schema, but got workbench.physical.Model

(Bug#54968)

- When using the **EDIT TABLE DATA** wizard (launched from Home screen), if a database with a single table was connected to, then after selecting the table the FINISH button remained grayed out. (Bug#54942)
- When started, the following warnings were generated in the Workbench Scripting Shell:

```
Starting Workbench...Registered 147 GRT classes.
WARNING: Could not load wb.mysql.validation.grt.dylib: Invalid module
/Applications/MySQLWorkbench.app/Contents/PlugIns/wb.mysql.validation.grt.dylib
```

```
WARNING: Could not load wb.validation.grt.dylib: Invalid module
/Applications/MySQLWorkbench.app/Contents/PlugIns/wb.validation.grt.dylib
```

(Bug#54929)

- When an attempt was made to synchronize a modified exported script with the current model, using <u>FILE</u>, <u>EXPORT</u>, <u>SYNCHRONIZE</u> <u>WITH SQL CREATE SCRIPT</u>, the model was not updated. (Bug#54501)
- When the ALTER TABLE dialog on an InnoDB table was displayed in an SQL Editor session, the FOREIGN KEY OPTIONS INDEX showed INVALID, when a foreign key in the FOREIGN KEYS tab was clicked. (Bug#54471)
- MySQL Workbench caused Windows XP to reboot if launched from <u>START</u>, <u>PROGRAMS</u>, <u>MySQL</u>, <u>MySQL WORKBENCH 5.2 OSS</u>.

If a MySQL Workbench model file (.mwb) was opened directly, an error dialog titled **MICROSOFT VISUAL C++ RUNTIME LIBRARY** was displayed with the message:

```
R6034
An application has made an attempt to load the C runtime library incorrectly. Please
contact the application's support team for more information.
```

The dialog's only option was OK. After clicking OK, two more instances of the same dialog were displayed. After clicking OK for all three, MySQL Workbench opened with the model file displayed.

On Windows 7 x64 the same run-time error (R6034) occurred. (Bug#52949)

• When a SQL Editor connection for a remote server was opened, the following error was generated:

OverviewBE::get_node:invalid node 0.0.0.

(Bug#52648)

A.1.5. Changes in MySQL Workbench 5.2.26 (06 August 2010 GA)

Second GA release of 5.2. This section documents all changes and bug fixes that have been applied since the release of 5.2.25.

Functionality added or changed:

• Added default value of CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP to context menu of column editor for TIMESTAMP columns. (Bug#54987)

Bugs fixed:

- On the Home screen, the saved connections previously listed in the Workspace were no longer visible. (Bug#55484)
- When a model was synchronized with a live database, CHAR(n) columns were always synchronized as CHAR(1). (Bug#55334)
- Importing a DBDesigner model caused MySQL Workbench to crash. (Bug#55300)
- In the SYNCHRONIZE MODEL WITH DATABASE wizard, the list of schemata was not drawn correctly on the SELECT SCHEMATA page, if BACK was clicked from the FETCH OBJECT INFO page. (Bug#55174)
- Complex queries with multiple newlines did not fit well in the ACTION column of the OUTPUT tab of the SQL Editor. This prevented any query error messages from being read. (Bug#55117)
- In the Reverse Engineer Database wizard, if a pattern mask was used to filter tables being imported, MySQL Workbench subsequently crashed. (Bug#55076)
- In the Administrator, if the MySQL server configuration file was modified to enable skip-innodb, and the server restarted, InnoDB was still active. (Bug#55030)
- On the SNIPPETS tab, the tooltip for the button INSERT SELECTED SNIPPET TO CURSOR POSITION IN THE ACTIVE SQL EDITOR contained the misspelling "seleced". (Bug#55026)
- Configuration variables in the my.cnf file that contained an underscore character were not displayed in the CONFIGURATION sec-

tion of the MySQL Workbench Administrator. (Bug#54933)

- MySQL Workbench crashed if minimized while loading a model. (Bug#54918)
- In the **OVERVIEW** tab of the SQL Editor, if there was a large number of schemata, and an attempt was made to use the arrow buttons in the interface to scroll through them, then MySQL Workbench crashed, usually on the last schema in the list. (Bug#54891)
- When using the SQL Editor and issuing a query containing SQL_CALC_FOUND_ROWS and a subsequent FOUND_ROWS() call, FOUND_ROWS() returned zero instead of the number of matching rows without a LIMIT applied. (Bug#54868)
- MySQL Workbench would not compile on RHEL 5. This was due to the calls to the function set_opacity in library/ form/gtk/src/lf_utilities.cpp (line 661) and library/forms/gtk/src/lf_popup.cpp (line 64). The set_opacity function was not available in GTK 2.10. (Bug#54844)
- Multi-statement syntax did not work in the SQL Editor. If the delimiter was set to a character other than the default semi-colon, then the semi-colon should still have been recognized as a valid delimiter for multi-statements, but in the SQL Editor it was not. (Bug#54831)
- In the SQL Editor, if a column's comment contained a Chinese word, then in the Object Browser, if the context-sensitive menu item <u>SEND TO SQL EDITOR</u>, <u>CREATE STATEMENT</u> was selected, incomplete script code was sent to the SQL Query tab. (Bug#54822)
- When a model was forward engineered to a database, insert statements associated with binary data were not present in the generated code. As a result, binary data was not forwarded to the database. (Bug#54680)
- In an EER Diagram the circles at each end of a relationship line were drawn in different sizes. (Bug#54663)
- When forward engineering a SQL ALTER script, the generated code attempted to recreate all foreign keys. (Bug#54363)
- In forward engineering a model to a database, the PRIMARY index was not reordered correctly when columns were reordered. (Bug#54176)
- When zooming in and out of the EER Diagram view, MySQL Workbench generated an unknown error. (Bug#52829, Bug#55390)
- When connecting to a server, MySQL Workbench prompted for a password, even if one had not been set for that account. (Bug#52826)
- After creating a server instance and double-clicking it on the Home screen, the Administrator took an excessively long time to load (over 10 minutes). (Bug#52530)
- MySQL Workbench crashed when executing LOAD DATA LOCAL INFILE as a query in the SQL Editor. (Bug#49694)
- If the Start Server button was clicked twice in MySQL Workbench the server would stop. (Bug#49588)
- In the list view of the physical schemata, it was possible to click the column headings of the lists, but this failed to sort the items by that column. (Bug#38863)

A.1.6. Changes in MySQL Workbench 5.2.25 (30 June 2010 GA)

First GA release of 5.2. This section documents all changes and bug fixes that have been applied since the release of 5.2.24.

Functionality added or changed:

 MySQL Workbench has been changed to support prompting the user for a password on attempting to connect, when using a TCP over SSH connection type, if a password was not previously provided during configuration of the connection. (Bug#47892)

Bugs fixed:

 Compiling MySQL Workbench from source code failed. The error was generated while building library/ grt/src/grtpp_grt.cpp:

```
Could not find file cairo_features.h
(included from cairo.h, which is included from
library/base/src/string_utilities.h, which is included in grtpp_grt.cpp)
```

(Bug#54833)

- On exit MySQL Workbench prompted the user to save changes for every query tab that had been opened in the SQL Editor, even where a table had simple been viewed, and no script code had been written by the user. (Bug#54754)
- When synchronizing a model with the live database, the generated SQL included erroneous DROP and ADD column statements, and also erroneous ADD and DROP index statements. (Bug#54740)
- Inserts data did not appear to be preserved between MySQL Workbench versions.

A model was created with MySQL Workbench 5.1.18 and data inserted using the **INSERT** tab. If the model was then saved and loaded into MySQL Workbench 5.2.22, and the model forward engineered to a SQL script, the generated script contained **INSERT** statements that inserted NULL instead of the data originally typed in. (Bug#54639)

- In the CREATE ROUTINE dialog of the SQL Editor, the text label associated with the NAME textbox mentioned "view", rather than "routine". (Bug#54566)
- If MySQL Workbench was minimized after a connection with a server was established, the following error was generated:

SplitterDistance must be between PanellMinSize and (Width - Panel2MinSize)

(Bug#54482)

- In the Data Dump section of the Administrator, using the **EXPORT TO BACKUP PROJECT FOLDER** option resulted in the triggers being exported to both the table files and the routines files. If an attempt was then made to import the exported files, an error resulted due to the duplicated triggers. (Bug#54426)
- When a .dbquery file, created with MySQL Query Browser, was opened in the SQL Editor, the Byte Order Mark (BOM) was not removed. This led to an erroneous character being prepended to the first statement, making it illegal. (Bug#54322)
- When forward engineering to a live database, the generated SQL statements contained an erroneous additional carriage return (^M) at the end of each line. However, this manifested in two ways. In the first case, in the **FORWARD ENGINEER TO DATABASE** wizard, if SAVE TO FILE was selected, a carriage-return character was appended to each line. In the second case, if COPY TO CLIPBOARD was selected, and the content subsequently pasted from the clipboard to an editor, then the extra carriage return characters was attached only to the DDL statements inside Create View and Create Trigger. (Bug#54272, Bug#54244)
- Printing or previewing an EER Diagram in Landscape/Tabloid produced blank output. (Bug#54240, Bug#54601)
- MySQL Workbench generated excessive diagnostic messages on start up.

MySQL Workbench has been changed so that by default diagnostic messages are not displayed on Linux, unless activated by starting MySQL Workbench with the --verbose option. (Bug#54021)

- In the **COLUMNS** tab of the Table Editor, the handling of column ordering appeared to have numerous problems. For example, dragging and dropping a column did not locate the column as expected. (Bug#53749)
- In the SQL Editor, if the ALTER ROUTINE dialog was maximized, the dialog buttons were no longer visible. (Bug#53734)
- In the Configuration tab of the Administrator, the parameter INNODB_SUPPORT_XA, on the InnoDB Parameters tab, has a corresponding checkbox. If this was selected and changes applied, the confirmation dialog indicated the following changes were made:

Added: [mysqld] innodb_support_xa = True

However, what was added to the my.cnf file was "innodb_support_xa". This meant there was no way to set this parameter, which defaults to True, to False using MySQL Workbench. (Bug#53449)

- In the SQL Editor, if a query was executed, and MySQL Workbench minimized, then on maximizing the application, the **RESULTS** tab window area was not correctly redrawn. (Bug#51991)
- When synchronizing a model that contained triggers, erroneous DROP trigger and CREATE trigger statements were generated, even when the triggers had not been changed. (Bug#51929)

- MySQL Workbench displayed instability after an SSH connection to the server was closed. This resulted in hanging or crashing behavior. (Bug#50554)
- When creating a table containing columns with a BOOLEAN data type, the model then failed to validate, even though BOOLEAN is a type offered on the drop down list of types available for columns. (Bug#49538)
- In the Inserts editor, the underscores in column headings were incorrectly displayed as accelerator/shortcut keys. (Bug#48831)
- The labels for the MANAGE DB CONNECTIONS dialog, such as username and password, were hidden when the dialog was opened at its default size. (Bug#47890)
- In the Modeler, when opening a new object such as a table, the currently open tab was reused, rather than a new tab being opened. This made it difficult to compare objects such as tables and routines. (Bug#45557)
- In the EER Diagram view, when a table was copied, it was found that the paste menu option was visible, but disabled, when an attempt was made to paste the table into a new diagram. (Bug#38432)

A.1.7. Changes in MySQL Workbench 5.2.24 (21 June 2010 RC)

Fourth release candidate. This section documents all changes and bug fixes that have been applied since the release of 5.2.23.

Functionality added or changed:

• A KILL QUERY button has been added to the CONNECTIONS tab of the Administrator. (Bug#54409)

Bugs fixed:

• In the Configuration tab of the Administrator, after changes were applied an error was generated:

```
Could not Save Configuration File
There was an error saving the configurationfile: exec_cmd() takes at least 3 arguments (2 given)
```

This happened while connecting to a FreeBSD 7.1 server over an SSH connection. (Bug#54647)

• In the SQL Editor, if a space was typed into the **SQL QUERY** tab, an external tab, such as the **OVERVIEW** tab or **OUTPUT** tab selected, and then <u>FILE</u>, <u>CLOSE TAB</u> selected, MySQL Workbench crashed.

Further, the **ADD TABLE** dialog had numerous issues, including inability to select PK and other checkboxes, failure to validate input, inability to apply changes, and deletion of user data in certain cases. (Bug#54124)

- The **CONNECTIONS** tab in the Administrator did not feature automatic refresh, this had to be manually triggered using the REFRESH button. Further, the KILL CONNECTION button was sometimes incorrectly disabled. (Bug#53703)
- If a foreign key relationship was created, and its modelOnly property set, then when the model was synchronized with a live database, the relationship was correctly excluded, however the DDL code for the foreign key indexes was still generated in the synchronization script. (Bug#53420)
- Synchronizing a model with a live database generated ALTER statements, even when there appeared to be no difference between the database and the model. (Bug#50938)
- In the SQL Editor, there was a blank space between the EXECUTE and STOP toolbar buttons. Hovering the mouse cursor over this blank space resulted in the tooltip "Explain selected SQL" being displayed, which appeared to indicate a missing icon. (Bug#49316)

A.1.8. Changes in MySQL Workbench 5.2.23 (Internal release only RC)

Third release candidate. This section documents all changes and bug fixes that have been applied since the release of 5.2.22.

Functionality added or changed:

- In the SQL Editor, when more result tabs that could fit into the current panel were generated, it was not obvious how to navigate to
 those result sets that could not currently be displayed. This could be achieved using the left and right arrow keys, but MySQL Workbench has now been changed to include a spin control to navigate the available tabs. (Bug#52998)
- The Overview tab in the SQL Editor has now been changed to support scrolling of content using the mouse wheel. (Bug#50782)

Bugs fixed:

- In the SQL Editor, when an attempt was made to execute a selection in the SQL Query tab, all text was executed, not just the selected text. (Bug#54536)
- When clicking on a table, MySQL Workbench generated the following error message:

```
MySQL Workbench has encountered a problem
Cannot access a disposed object
Object name: 'DockedWindowPlugin'.
```

(Bug#54498)

- In the SQL Editor, exporting a recordset failed if the target path contained an international character. (Bug#54479)
- In the Inserts tab of the Table Editor, adding a new row and applying changes did not have any effect. (Bug#54456)
- In the **GENERAL** tab of **CONFIGURATION** in the Administrator, when the **TEMP DIRECTORY** option was selected, and a directory selected using the file browser, MySQL Workbench inserted backslashes rather than forward slashes. If backslashes were used in this location in the configuration file, the server crashed on startup. (Bug#54445)
- In the SQL Editor, if a table in the Overview tab was double clicked, the query generated replaced code already typed into the SQL Query tab, resulting in the loss of that code. (Bug#54362)
- On synchronizing a model with a live database incorrect SQL code was generated:

ALTER TABLE `mydb`.`table1` AUTO_INCREMENT = ;;

(Bug#54319)

- In the SQL Editor, when exporting a resultset to a TAB delimited file, MySQL Workbench added a .CSV extension. (Bug#54302)
- In the SQL Editor, selecting <u>ALTER TABLE</u> from the context-sensitive menu had no effect. (Bug#54283)
- In the SQL Editor, editing a table with a row highlighted, right-clicking on a BLOB column and loading a value from a file resulted in the error: "An external component has triggered an exception". (Bug#54266)
- Clicking the EDIT TABLE DATA action item from the Home screen resulted in MySQL Workbench crashing. (Bug#54230)
- The forward engineering option **OMIT SCHEMA QUALIFIER IN OBJECT NAMES** was not correctly applied to some statements in Trigger code. In these cases the schema name remained in place. (Bug#54222)
- When clicking on external links in the embedded browser, Internet Explorer was launched, rather than the default browser. (Bug#54192)
- When entering binary data into the **INSERTS** tab for a table, the data was truncated at the first null byte (0x00). (Bug#54156)
- Select All (Ctrl+A) did not work when editing comments for a table or column. (Bug#54154)
- After a foreign key was created, and the referenced table or column changed, the change was not reflected in the foreign key, resulting in errors. (Bug#54134)
- MySQL Workbench crashed if in the SQL Editor a pinned result tab was refreshed. (Bug#54114)
- The integrated docs webserver listened on a fixed port 8811, and was not able to handle the situation where this port may be used by another service, such as another MySQL Workbench instance. (Bug#54104)
- In the SQL Editor, when attempting to create a new schema using <u>CREATE SCHEMA</u>, with a server collation of UTF8 DEFAULT

COLLATION, the generated SQL resulted in an error when applied. (Bug#54097)

- Performing an undo operation in SQL Editor actually resulted in an undo operation in the Modeler. (Bug#54091)
- MySQL Workbench suffered user interface issues when used on a KDE-based system using a dark theme. (Bug#54083)
- The HTTP server thread started by MySQL Workbench to provide documentation to the local web browser listened on all IP interfaces, even though it was only intended for local use. (Bug#54083)
- Export of database failed due to a faulty path in the command line:

15:42:52 Dumping rma3 (etat_produit)

```
Running: "mysqldump.exe" --defaults-extra-file=c:\docume-1\user\local
settings\temp\tmp9ypdh --no-create-info=FALSE --order-by-primary=FALSE --force=FALSE
--no-data=FALSE --tz-utc=TRUE --flush-privileges=FALSE --compress=FALSE --replace=FALSE
--host=127.0.0.1 --insert-ignore=FALSE --extended-insert=TRUE --user=root
--quote-names=TRUE --hex-blob=FALSE --complete-insert=FALSE --add-locks=TRUE --port=3306
--disable-keys=TRUE --delayed-insert=FALSE --delete-master-logs=FALSE --comments=TRUE
--default-character=set=utf8 --flush-logs=FALSE --dump-date=TRUE --allow-keywords=FALSE
--create-options=TRUE --events=FALSE 'rma3" "etat_produit"
Could not open required defaults file: c:\docume-1\user\local
Fatal error in defaults handling. Program aborted
mysqldump: Got error: 1045: Access denied for user 'root'@'localhost' (using password:
NO) when trying to connect
Operation failed with exitcode 2
```

(Bug#54067)

- In the Alter Table dialog of the SQL Editor, if an attempt was made to set the NN flag for a column that did not have it set, an error was generated when the changes are applied. (Bug#53918)
- When carrying out an Alter Routine operation in the SQL Editor, if a routine was modified, and a statement used that, although syntactically correct, was not permitted in a routine, then when changes were applied, all modifications were lost. (Bug#53887)
- An EER Diagram appeared to contain hidden tables. Although the diagram contained 12 tables, when a 'select all' was performed the user was informed that 14 tables were selected. It seemed that certain tables had been resized to be so small as to no longer be visible in the diagram. (Bug#53885)
- In the Layer Tree panel, if a layer was given a blank name, then MySQL Workbench subsequently crashed, rather than warning about the use of a blank name. (Bug#53807)
- Keyboard shortcuts such as Cmd+C, Cmd+V, and Cmd+Z were not useable consistently in all parts of MySQL Workbench. (Bug#53790)
- Columns defined as BIT appeared as requiring synchronization, even if no changes had been made. (Bug#53747)
- In the ALTER ROUTINE dialog of the SQL Editor, if the FIND AND REPLACE dialog was invoked with Ctrl+F, and Ctrl+V used to paste in some search text, the text was actually pasted into the routine code area. (Bug#53735)
- MySQL Workbench failed to build from source code on Debian Linux. The following error was generated:

(Bug#53668)

- In the EER Diagram view, it was not possible to delete a relationship by right-clicking on the relationship and then selecting <u>DELETE</u> from the context-sensitive menu. (Bug#53659)
- In the SQL Query tab of SQL Editor, tooltip error messages were still displayed, even after switching to another tab. (Bug#53634)
- In the **DIAGRAM** Size dialog, displayed using <u>MODEL</u>, <u>DIAGRAM PROPERTIES AND SIZE</u>, it was possible to set the size to 100 pages by 100 pages, but then click outside this area. This caused an MySQL Workbench Unexpected Error. (Bug#53559)
- The DOCUMENT PROPERTIES dialog was displayed across two screens in a multi-screen setup. (Bug#53262)
- After a default primary key name was set in the Workbench Preferences dialog, in the Table Editor the PK flag was not selected as expected, and the column flags could not be selected. (Bug#52578)
- When synchronizing with a live database, MySQL Workbench did not interpret whitespace in ENUMs correctly. For example, ENUM('image', 'video', 'swf') was seen as different from ENUM('image', 'video', 'swf'), causing the live server and model to appear unsynchronized. (Bug#49182)
- When editing a column in the table editor, pressing **CTRL** + **Z** to undo the edit resulted in a previous operation being undone instead. (Bug#46286)

A.1.9. Changes in MySQL Workbench 5.2.22 (02 June 2010 RC)

Second release candidate. This section documents all changes and bug fixes that have been applied since the release of 5.2.21.

Functionality added or changed:

• On Linux, MySQL Workbench automatically added a .sql extension, where an extension was not specified, to any script file saved from the SQL Editor.

MySQL Workbench has been changed so that on Linux a .sql extension is not automatically added. However, on Windows the file extension is added if one is not specified. (Bug#53731)

- MySQL Workbench has been changed so that only the Model Overview tab indicates that changes need to be saved, using an asterisk, rather than this being reflected on individual EER Diagrams. (Bug#53042)
- The SQL Editor now includes a new context menu item <u>SET AS DEFAULT SCHEMA</u>. This is applicable to schema nodes in the **OBJECT BROWSER** and the **OVERVIEW** panel of the Query Editor. Previously the only way to change the default schema was to select the schema from the schema selector drop-down listbox located at the top of the Object Browser. (Bug#51479)
- MySQL Workbench has been changed to support creating a new tab in tabbed interfaces by using Ctrl+T on Windows and Cmd+T on Mac OS X. (Bug#48798)
- MySQL Workbench now supports entering of expressions such as DEFAULT or CURRENT_TIMESTAMP into the inserts editor. The expression needs to be preceded by \func. (Bug#36206)

Bugs fixed:

- If a property value was changed in the **WORKBENCH PREFERENCES** dialog, and OK clicked, or a new tab clicked before the cursor was moved away from the value being changed, then the change was silently reverted. (Bug#54060)
- If the MANAGE SERVER INSTANCES dialog was launched, and either the NEW or DELETE button clicked, and then the dialog closed, MySQL Workbench crashed if the NEW SERVER INSTANCE action item was then clicked on the Home screen. (Bug#54028)
- Gnome keyring package was a dependency of MySQL Workbench. It was undesirable to require its installation on KDE-based systems. (Bug#54010)
- When MySQL Workbench was launched, an overlay was displayed. In the application bar the dialog prompting for the keyring password was waiting for input but it was not possible to enter the password as the overlay hid the dialog. (Bug#53974)
- On the Model Overview page, when a new user object was created with Add User in the Schema privileges section, it was not possible to change any of the user object's attributes. For example, if the name of the object was changed, and entered, it would then change back to its original setting. (Bug#53946)
- In the EER Diagram view heavy flicker resulted if the cursor was held over relationship lines or tables. (Bug#53941)
- When a table containing triggers was loaded into the ALTER TABLE dialog of SQL Editor, clicking on the TRIGGERS tab showed an empty trigger, rather than the correct trigger code. (Bug#53940)
- In the SQL Editor, performing a query on a remote server returned BLOB values for columns that were not of type BLOB.

Since binary byte strings tend to contain null bytes in their values, for safety reasons they were not displayed in the results grid. They could only be viewed or edited by means of the BLOB editor to avoid data truncation.

MySQL Workbench has been changed to include a new global option which has been added to the **SQL EDITOR** tab of the **WORK-BENCH PREFERENCES** dialog. The option is "Treat BINARY/VARBINARY as non-binary character string". By default it is not selected. (Bug#53920)

- Attempting to copy objects from an existing diagram to a new diagram resulted in the following error: OverviewBE::get_node: invalid node 1.0.0.22. (Bug#53914)
- In the SQL Editor, the context-sensitive menu options <u>COPY TO CLIPBOARD</u> and <u>SEND TO SQL EDITOR</u> both failed to work. This applied to both copying in the **OBJECT BROWSER** and in the **OVERVIEW** tab. (Bug#53903)
- In the SQL Editor, when deleting a snippet from the snippet list using the toolbar button, the first snippet was deleted, regardless of which snippet was selected. (Bug#53860)
- When forward engineering a schema to a database with the **DROP OBJECTS BEFORE EACH CREATE OBJECT** option selected, the DROP statement used a delimiter before it had been declared. (Bug#53853)
- In the SQL Editor, if code was typed into the SQL Query tab, and then the application exited, MySQL Workbench did not prompt the user regarding unsaved changes, and so any code typed into the SQL Query tab, and not saved, was lost. (Bug#53848)
- In the SQL Editor, when renaming views or routines using the live editor, the generated ALTER script contained only drop statements, CREATE statements being omitted. (Bug#53847)
- In the SQL Editor, when attempting to change an index from INDEX to UNIQUE using the ALTER TABLE dialog, no changes were applied, and the following error message was generated: No changes to object were detected. (Bug#53787)
- In the CREATE SCHEMA dialog of the SQL Editor, when a new schema name was entered, and changes applied, the schema name appeared to revert to the default name new_schema, rather than being changed to the newly specified name. This only happened on Mac OS X. (Bug#53764)
- In the SQL Editor, if two procedures were defined in a script, each proceded with a suitable DROP statement, then when the script was executed and the procedures already existed a procedure already defined error occurred. It appeared that the second DROP statement was being ignored. This happened if the line preceding the second DROP statement was terminated with a space, which caused the DROP statement to be ignored, resulting in the error. (Bug#53760)
- In the SQL Editor, with multiple tabs open, if the script in each tab was saved to a file in turn, the script was actually saved over the previously saved script, resulting in the loss of that script. (Bug#53733)
- If multiple SQL Query tabs were opened in the SQL Editor, then when a snippet was saved to the snippet list, the snippet saved would always be from the last opened SQL Query tab (the furthest to the right), rather than the currently selected tab. (Bug#53713)
- In the Columns tab of the Table Editor, if typing in lower case the data type of the column was not automatically completed.

MySQL Workbench has been changed so that it now automatically completes the data type regardless of whether the user types upper or lower case letters. (Bug#53710)

• Compiling MySQL Workbench from source code failed with the following errors:

```
...
-I../../library/dbc/src -I../../library/vsqlite++/include -D_FORTIFY_SOURCE=2 -02
-fno-strict-aliasing -fwrapv -march=x86-64 -gdwarf=2 -g2 -Wextra -Wall -Wno-unused
-Wno-deprecated -ggdb3 -DENABLE_DEBUG -MT canvas_floater.o -MD -MP -MF
.deps/canvas_floater.Tpo -c -o canvas_floater.o `test -f 'model/canvas_floater.cpp' ||
echo './'`model/canvas_floater.cpp
```

model/canvas_floater.cpp: In constructor 'wb::Floater::Floater(mdc::Layer*, const std::string&)': model/canvas_floater.cpp:65:59: error: cannot call constructor 'mdc::Color::Color' directly model/canvas_floater.cpp:65:59: error: for a function-style cast, remove the redundant '::Color' make[3]: **** [canvas_floater.o] Error 1 make[3]: Leaving directory `/home/users/Arvenil/rpm/BUILD/mysql-workbench-oss-5.2.21/backend/wbprivate' make[2]: *** [all-recursive] Error 1 make[2]: Leaving directory `/home/users/Arvenil/rpm/BUILD/mysql-workbench-oss-5.2.21/backend' make[1]: *** [all-recursive] Error 1 make[1]: *** [all-recursive] Error 1 make[1]: Leaving directory `/home/users/Arvenil/rpm/BUILD/mysql-workbench-oss-5.2.21/ make: *** [all] Error 2

(Bug#53687)

- In the DATA DUMP tab of the Administrator, the OPTIONS panel was not rendered correctly. (Bug#53681)
- In the SQL Editor, if a query was executed, and then interrupted, MySQL Workbench hung. (Bug#53652)
- If a query was typed in the SQL Query tab, and then **cmd** + **Enter** immediately pressed to execute the query, the query failed. This only happened on Mac OS X. (Bug#53650)
- In the CREATE NEW SERVER INSTANCE PROFILE wizard, on the OPERATING SYSTEM page the MYSQL INSTALLATION TYPE drop-down listbox was not populated correctly. Although all items were initially correctly displayed, if the BACK button was clicked, and then NEXT clicked to return to the page, all the items in the list were duplicated. Each time this process was repeated the original list items would again be duplicated. (Bug#53649)
- On the Home screen, the MySQL Team Blog and Workbench Team Blog icons linked to the same page. Further, the MySQL News Letter icon linked to the MySQL Developer forum. (Bug#53628)
- In the SSH Shell Connection panel of the Manage Server Instances dialog, if any port other than 22 was specified it was subsequently ignored. (Bug#53623)
- The MySQL configuration file was corrupted by the Administrator when changes were applied, if the configuration file was not terminated with a new line. (Bug#53620)
- In the SQL Editor, the toolbar items on the Snippets tab to replace and insert snippets had no effect when clicked. (Bug#53610)
- Specifying a new font for the SQL Editor's SQL Query tab contents had no affect on appearance. (Bug#53605)
- In the Data Dump section of the Administrator, when an export was generated the following error occurred:

<code>mysqldump: Error 2020: Got packet bigger than <code>'max_allowed_packet'</code> bytes when dumping table <code>'xxx'</code> at <code>row: 1</code></code>

This only happened for larger tables. (Bug#53599)

- MySQL Workbench crashed when attempting to connect to a Linux server using an SSH connection. (Bug#53476)
- When a model containing a stored procedure was forward engineered to a script, the script contained invalid SQL code. This
 happened when the DROP OBJECTS BEFORE EACH CREATE STATEMENT and EXPORT MYSQL ROUTINE OBJECTS options
 were selected. Problems that occured with the SQL code included a delimiter being used before its definition and also DROP statements not being terminated. (Bug#53467)
- MySQL Workbench crashed on selection of menu item <u>MODEL</u>, <u>CREATE DIAGRAM FROM CATALOG OBJECTS</u>. (Bug#53454)
- It was not possible to edit schema properties, such as name and default collation, in the schema editor. (Bug#53438)
- When a model containing a table using a 1:n relationship was synchronized with a live database, the relationship then failed to render in the EER Diagram. (Bug#53252)
- When a large script was loaded into the SQL Editor, an out of memory exception occurred.

Inefficient end of line conversion in MySQL Workbench was the cause of memory exhaustion. The SQL Editor has been enhanced

to support custom end of line formats and now preserves original line endings used in the file. End of line conversion is not used any more and loading of big files can be done faster and more efficiently. (Bug#53190)

- In the Manage DB Connections dialog, selecting the USE ANSI QUOTE TO QUOTE IDENTIFIERS had no effect with respect to subsequent queries using that connection. (Bug#53160)
- Delimiters were erroneously added to trigger code if the currently selected schema was changed. (Bug#53151)
- In the **TRIGGERS** tab of the **TABLE EDITOR**, if a trigger was written that contained a DROP TRIGGER statement, the statement was automatically removed. (Bug#53023)
- MySQL Workbench generated an exception when an attempt was made to change the configuration file in the SYSTEM PROFILE tab of the MANAGE SERVER INSTANCES dialog. (Bug#52947)
- In the ALTER TABLE dialog of the SQL Editor, when a comment was entered for a column in the COMMENTS tab, and the changes applied, the comment was not saved. (Bug#52893)
- Setting WB_NO_GNOME_KEYRING=1 on Kubuntu did not appear to disable the Gnome keyring, it was therefore impossible to connect without an active keyring daemon. (Bug#52866)
- When scrolling in an EER Diagram, MySQL Workbench repeatedly displayed an error dialog containing the message "OpenGL error: invalid value". (Bug#52840)
- In the DATA DUMP section of the Administrator, when a schema was dumped to multiple files, the stored routines were not exported. (Bug#52798)
- It was not possible to set ON UPDATE CURRENT_TIMESTAMP as the default value for a column. (Bug#52350)
- If a table was dragged from the catalog tree on to the EER Diagram, then the catalog tree collapsed. (Bug#51862)
- In the CONNECTIONS tab of the Administrator, queries containing a comment did not always display correctly in the INFO column. (Bug#51848)
- Output did not scale correctly when printing sizes other than A4. (Bug#51515)
- When synchronizing to a live database, triggers showed as requiring updating, even when the model and live database were identical. (Bug#51409)
- If a SQL file, using Windows file format, was loaded into the SQL Editor, it was displayed with additional line breaks, and the SQL code was not interpreted correctly. Further, if code written in the SQL editor was saved to a file and then opened in an editor on Windows, it contained additional line breaks. (Bug#51308)
- In the Connections tab of the Administrator facility, hostnames were sometimes not displayed correctly. This happened if the hostname contained a '-' character. (Bug#51119)
- In the SQL Editor, if the ADD TABLE facility was used to create a new table with the same name as an existing table, then after changes were applied, the content of both tables was the same as the existing table, and changes unique to the second table were lost. (Bug#51107)
- When an attempt was made to forward engineer a project which contained a view, it failed with the following error:

ERROR: Error 1046: No database selected

(Bug#51091)

- Inserts added to a model and saved on Windows were not visible when that model was loaded on Mac OS X. (Bug#51029)
- In the SQL Editor, carrying out a SELECT on a table containing more than 1000 columns resulted in the error 'Too many SQL variables'. (Bug#50952)
- In the Model Overview page, when small icons view was selected, if table names were truncated, there were no tooltips to enable visibility of the full name. The tooltips, however, were present for the large icons view, and the details view. (Bug#50739)
- MySQL Workbench reported a server as not running, even though the server was correctly running. MySQL Server concerned was running on Windows 7, 64-bit. (Bug#50585)

- In the EER Diagram view, it was not possible to paste a copied layer. The paste menu item was disabled in the context-sensitive menu and main menu, and also pressing **Ctrl+V** had no effect. (Bug#50253)
- In the EER Diagram view, when the diagram was displayed over multiple pages, the gray line dividing pages in the Navigator was not visible in the diagram itself. (Bug#50243)
- In the EER Diagram view, when zoomed out, a layer could be placed beyond the diagram's perimeter at the bottom right. Further, if a layer was placed beyond the perimeter at the top left, it snapped back within confines of the diagram correctly, but the navigator view was not updated accordingly. (Bug#50242)
- Zooming out from a diagram in EER Diagram view resulted in severe flicker. (Bug#50240)
- In SQL Editor, when using the ALTER TABLE facility to set a column to Auto Increment (AI), the dialog reported that "No changes to object were detected". (Bug#50214)
- If a Workbench model file was double clicked, MySQL Workbench crashed on launch. However, if MySQL Workbench was started using its application icon, and then the same model file loaded from within MySQL Workbench, no problems were encountered. (Bug#49941)
- When using **CREATE EER MODEL FROM SQL SCRIPT** from the Home screen, the specified engine type was not present in the resultant EER model. (Bug#49568)

A.1.10. Changes in MySQL Workbench 5.2.21 (12 May 2010 RC)

First release candidate. This section documents all changes and bug fixes that have been applied since the release of 5.2.20.

Functionality added or changed:

- MySQL Workbench now features a check that compares the version of mysqldump on the client and server machines. (Bug#52689)
- MySQL Workbench has been changed to include the following keyboard shortcuts and menu items:
 - Execute Current Statement, **Ctrl** + **Enter**
 - Execute (All or Selection), Ctrl + Shift + Enter
 - Explain Current Statement, Ctrl + Alt + Enter
 - Explain (All or Selection), Ctrl + Alt + Shift + Enter

The Stop toolbar button and menu item are also now automatically enabled and disabled. (Bug#50911)

Bugs fixed:

- After proceeding through the **EDIT TABLE DATA** wizard, launched by the **EDIT TABLE DATA** action item on the Home screen, and clicking FINISH to complete the wizard, MySQL Workbench crashed. (Bug#53511)
- When forward engineering a model to a database, not all tables were created on the live database. (Bug#53506)
- Russian characters were not displayed correctly in the SQL Query tab of the SQL Editor. (Bug#53408)
- It was not possible for the index name and foreign key name of a column to be given different names in MySQL Workbench.

MySQL Workbench has been changed so that renaming an index will not automatically rename a corresponding foreign key. However, renaming a foreign key will automatically rename the associated index only if the names still match at the time the foreign key is renamed. (Bug#53370)

- When using the EDIT SQL SCRIPT action item on the Home screen, the script failed to load and execute. (Bug#53359)
- If a socket/pipe connection method was used, and the password stored in the keychain, MySQL Workbench could not connect to the server. (Bug#53355)

- In the Mac OS X version of MySQL Workbench, the overlay text for NULL and BLOB fields was missing from the results grid in the SQL editor. This meant that those fields appeared empty if a query was executed that returned a BLOB, such as select shal('hello'); (Bug#53335)
- Changing the fonts in the WORKBENCH PREFERENCES dialog had no effect on the appearance of MySQL Workbench. (Bug#53321, Bug#49832)
- In the SQL Editor, if some script text was highlighted, and Ctrl-S was pressed to save the script, only the highlighted text was saved to the file. (Bug#53275)
- An error was generated when a server was selected from the Import/Export MySQL Data dialog, displayed after clicking the Manage Import/Export action item on the Home screen. (Bug#53271)
- In the modeler, if a table was double-clicked to open it in the Table Editor, and then, without closing the Table Editor, another table double-clicked to open it, the Table Editor was rendered as a blank panel. (Bug#53250, Bug#53247)
- MySQL Workbench crashed after using the EDIT SQL SCRIPT action item from the Home screen. In the EDIT SQL SCRIPT wizard, a connection was chosen that did not use a stored password, and then a script selected and the FINISH button clicked to exit the wizard. MySQL Workbench subsequently crashed when the requested password was entered into the dialog and the OK clicked. (Bug#53247)
- In the SQL Editor, if a BLOB field containing an image was loaded into the Viewer, the image was not displayed. (Bug#53245)
- In the ADMINISTRATOR tab of the WORKBENCH PREFERENCES dialog, the description to the right of the text field PATH TO MYSQL TOOL was incorrect. (Bug#53201)
- MySQL Workbench permitted the creation of invalid foreign keys that referenced a column that did not use a PRIMARY KEY or UNIQUE key declaration.

MySQL Workbench has been changed to filter the list of columns permitted for a referenced foreign key, so that only primary key and indexed columns are listed. (Bug#53136)

- In the Alter Table dialog of the SQL Editor, it was not possible to create new Foreign Keys. (Bug#53134)
- If a model was saved using a file name that contained non-English characters, it could not be reopened from the list of existing models on the Home screen. (Bug#53110)
- In the SQL Editor, if multiple Query tabs were opened so that they used more than the available screen width, it was not possible to move to the tabs that were not visible.

MySQL Workbench has been changed so that scroll buttons appear should the number of tabs exceed the screen width. It is then possible to access non-visible tabs using the scroll buttons. (Bug#53070)

- If a model was forward engineered to a database, renamed, and then forward engineered again, the SQL that was generated was erroneous. (Bug#53068)
- In the **INSERTS** tab of the Table Editor, if characters were inadvertently typed into an integer column, MySQL Workbench crashed when a new column was selected. (Bug#53066)
- If a table was copied in the modeler, the copy of the table was renamed. However, foreign key relationships contained within the table were not renamed. (Bug#53020)
- In the Alter Table dialog of the SQL Editor, if two triggers were created, the second with a bad command, and the error corrected, then when changes were applied, the generated code was incorrect, dropping both triggers. (Bug#52997)
- It was not possible to increase the font size of text in the SQL Editor.

MySQL Workbench has been changed so that fonts set in the Workbench Preferences dialog now also affect the SQL Editor. (Bug#52968)

- It was not possible to delete a trigger from the **TRIGGERS** tab in the **ALTER TABLE** dialog. Even though the trigger code was selected and deleted, and then APPLY clicked, the code that was generated did not drop the deleted trigger as expected. (Bug#52955)
- Synchronize to live database failed when the model contained triggers. (Bug#52939)
- In the SQL Editor, when the ALTER TABLE context-sensitive menu option was used on a table, and changes made, it was not pos-

sible to save them, as the dialog reported that no changes to the object had been detected. (Bug#52786)

- In the SQL Editor, if the table data was edited, and then an attempt made to apply the changes, the changes were not applied. (Bug#52705)
- Selecting <u>PLUGINS</u>, <u>UTILITIES</u>, <u>TEST FILTERS</u> caused MySQL Workbench to generate an unexpected error:

Error calling WbAdmin.openFilterDebugger

The Filters Debugger facility can now be accessed in the MANAGE SERVER INSTANCES dialog. It is located on the Server Stats subtab of the Server Profile tab. (Bug#52655)

- In the SQL Editor the Find and Replace facility did not appear to function. (Bug#52251)
- When the page size for an EER Diagram was changed in <u>FILE</u>, <u>PAGE SETUP</u>, this was not reflected in the model navigator. (Bug#52077)
- When editing table data in the SQL Editor, the table would refresh after data was edited in the first column. This meant it was necessary to then scroll to the bottom of the table data to continue editing the record. (Bug#51965)
- Any columns that contained a comment would be marked for change when synchronizing the model with the live database. (Bug#51765)
- If a diagram name contained an ampersand, the name was not displayed correctly in the Model Overview. (Bug#51617)
- The default Export Directory Path in the WORKBENCH PREFERENCES dialog was given as DriveLetter: \MyDocumentsPath/dumps. This mixed the '/' and \\' directory separators.

Further, the preferred Export Directory Path set in the Workbench Preferences dialog, was not reflected in the Export to Disk section of the Data Dump facility in the Administrator. (Bug#51519)

- In the Alter Table dialog of the SQL Editor, the Column and Referenced Column were not updated when a foreign key was created in the Foreign Keys tab. (Bug#51262)
- In the Alter Table dialog of the SQL Editor, there was no effective way of altering a foreign key. (Bug#51124)
- In trigger code, triggers using the alias 'new' were correctly colored by the syntax highlighter, but triggers using the alias 'old' were not colored. (Bug#51061)
- When an EER Diagram layout was set to span multiple pages, and then printed, the diagram was printed so small as to be illegible. (Bug#51032)
- When MySQL Workbench was maximized it did not fit correctly onto the screen, the title bar was not displayed. (Bug#50975)
- If the name of a table was changed in the ALTER TABLE dialog, and APPLY clicked, then the message "no changes to object were detected" was received, even though a change had been made. (Bug#50894)
- In the EER Diagram view, changing the page orientation to landscape using <u>FILE</u>, <u>PAGE SETUP</u> caused multiple problems in MySQL Workbench. The Navigator orientation did not change, the diagram did not print at the correct size, and MySQL Workbench eventually crashed. (Bug#50861)
- When the SQL Editor was launched, MySQL Workbench became overloaded, due to a large number of tables (10,000+) in the schema.

MySQL Workbench has been changed to include two new options in the **WORKBENCH PREFERENCES** dialog: **DISABLE LIVE SCHEMA OVERVIEW** and **SHOW SCHEMA NAMES ONLY IN CATALOG LIST**. This enables the application to be configured to handle schemata with large numbers of tables. (Bug#50701)

• The Relationship and Layer editors did not close if the object they were editing was deleted.

MySQL Workbench has been changed to support auto-closing of all open editors, should the object being edited be deleted. (Bug#50681)

• If a PDF is generated of the EER Diagram and the resultant PDF file viewed in Acrobat Reader, then if an attempt is made to reprint the diagram, overwriting the file that is currently opened in Acrobat Reader, MySQL Workbench crashes. (Bug#50408)

- In the USER DEFINED TYPES dialog, launched using the main menu item <u>MODEL</u>, <u>USER DEFINED TYPES</u>, selection of the UN-SIGNED or ZEROFILL checkboxes was not reflected in SQL code generated by the Forward Engineer SQL Script wizard. (Bug#50323)
- If some changes were made in the Table Editor, and the editor was still in edit mode, then the active changes in the Table Editor were not saved when the model was saved. (Bug#50006)
- MySQL Workbench crashed when using the COMPARE AND REPORT DIFFERENCES IN CATALOGS wizard. (Bug#49561)
- Find was always enabled, even in contexts where it had no useful functionality. (Bug#49293)
- MySQL Workbench permitted setting the auto-increment checkbox for multiple columns in the Table Editor. (Bug#49278)
- In the SYNCHRONIZE MODEL WITH DATABASE wizard, on the SELECT CHANGES TO APPLY page, if a schema was selected and then any of the buttons UPDATE MODEL, IGNORE, or UPDATE SOURCE was clicked, this change was not applied to child objects of the schema. (Bug#45454)

A.1.11. Changes in MySQL Workbench 5.2.20 (27 April 2010 beta)

Tenth public Beta release. This section documents all changes and bug fixes that have been applied since the release of 5.2.19.

Functionality added or changed:

- The SQL Statements tab, and the record set data editor tab, now indicate when they have unsaved data, by displaying an asterisk character. (Bug#52799)
- If MySQL Workbench was used on an external monitor, as part of an extended desktop, then when the computer was started without the external monitor, MySQL Workbench was not displayed on the primary display, and was therefore invisible.

MySQL Workbench has been changed, so that if used on an extended desktop, and then the computer is started without the external display, the application will locate itself so that half of the application window is displayed on the primary desktop, where it can be dragged and resized as required. (Bug#52703)

• The FK/index mapping implementation has been changed so that indexes created for foreign keys are no longer required to be marked as FOREIGN. They now are created with type INDEX and can be changed to UNIQUE or other types.

Although, the artificial FOREIGN index type has been removed, MySQL Workbench still automatically creates and maintains indexes for each foreign key by naming them after the keyname and keeping the names (FK to IDX) synchronized. (Bug#48728)

Bugs fixed:

- The MySQL Workbench configure script, configure.in, contained a construct incompatible with NetBSD. The script used test == instead of test =. (Bug#53175)
- In the SQL Editor, when commands were typed into the SQL Statements area, and executed, they appeared on the Output tab. However, when the number of statements appearing in the Output tab exceeded the number that could be displayed in the default area, a vertical scrollbar did not appear. This meant any further statements that were executed were effectively hidden, as it was not possible to scroll vertically to display them. (Bug#53156)
- If more than one SQL Editor tabs were opened, closing the last one caused MySQL Workbench to crash. (Bug#53061)
- If a schema was opened, an object editor, such as the Table Editor opened, and the schema then closed, MySQL Workbench crashed if the schema was immediately reopened. (Bug#53027)
- In the VARIABLES tab of the Administrator, variables with long descriptions were not displayed correctly, they appeared wrapped to a new line, and clipped by the height of the DESCRIPTION row. (Bug#53025)
- If SQL Editor was using a named pipe connection, and the SQL Editor tab was closed, then MySQL Workbench stopped responding to user input and had to be killed using the Task Manager. (Bug#53021)
- The HUD blocked access to other applications that were running. This was particularly a problem when Administrator or SQL Editor were launched from the Home screen, and took a long time to load. (Bug#53006)

• In the Export to Disk tab of the Administrator's Data Dump facility, selecting multiple schemata for export to a self-contained file resulted in the following exception:

unhandled exception: local variable 'tables' referenced before assignment.

(Bug#52981)

- When an EER Diagram was displayed the **PROPERTIES** tab was empty. Also, if a table in the EER Diagram was selected the **PROPERTIES** tab remained empty. (Bug#52971)
- In the SQL Editor, when viewing field data of type VARBINARY using the <u>OPEN VALUE IN VIEWER</u> context-sensitive menu item, only the data up to the first null (0x00) was displayed. (Bug#52954)
- In the results editor of the SQL Editor, deleting more than two hundred records resulted in the following error:

Attempting to Read or Write protected memory. This is often an indication that other memory is corrupt

This happened when using either the delete key or the DELETE ROWS context-sensitive menu option. (Bug#52951)

- When MySQL Workbench was sized to 1280 x 800, the START EXPORT button was not visible in the **EXPORT TO DISK** tab of the Data Dump facility. (Bug#52932)
- If multiple tables were selected at the same time in the Overview tab of the SQL Editor, MySQL Workbench crashed. (Bug#52922)
- MySQL Workbench failed to compile. This was due to a missing #include <stdarg.h> statement in the file library/ sql-parser/include/myx_sql_tree_item.h. (Bug#52919)
- When building MySQL Workbench, the build process failed if the --no-copy-dt-needed-entries linker option was specified (this happens by default when building on Fedora 13). (Bug#52918)
- Print to PDF, Print to PS File, and all export functions did not work correctly. For example, if Print to PDF was selected, MySQL Workbench would attempt to open a new document, and if the user proceeded, the current document would be closed.

Further, selecting the menu option <u>SAVE MODEL AS</u> resulted in the **OPEN DOCUMENT** dialog being displayed. If the user clicked YES the application became stuck in a loop, if No was clicked an error dialog was displayed. (Bug#52909)

- Script editors, such as the editor in the SQL Editor, that used the Scintilla component, did not have full integration with the Edit menu. Short cuts and menu commands for operations such as copy, paste, select all, and find were not implemented. (Bug#52834, Bug#51806)
- In the EER Diagram view, layer objects did not respond to edit commands (either double clicking or using the <u>EDIT LAYER</u> contextsensitive menu option). (Bug#52822, Bug#52823)
- While MySQL Workbench was starting up, if the Windows screensaver activated, this led to unpredictable behavior of MySQL Workbench. For example, the application could freeze, fail to redraw its main window, or display artifacts. On other occasions the application was not able to accept keyboard input, and had to be terminated using the Task Manager. (Bug#52780)
- When switching between Model Overview Page, and EER Diagram View, MySQL Workbench incorrectly rendered the EER Diagram View inside the Table Editor panel. (Bug#52778)
- The MySQL Workbench dependency on libmysql has changed to use version 16 of the library rather than 15. (Bug#52682)
- If a table comment contained a single quote character, then when an attempt was made to synchronize this with a live server, an error would result. This was because the code generated by the synchronization process did not escape single quotation marks in the table comments. (Bug#52608)
- In the SNIPPETS tab of the SQL Editor, there appeared to be a third column, with no heading or data, in the snippets list.

MySQL Workbench has been changed so that the second column fills the available space. (Bug#52559)

• The View Editor deleted text pasted into the editor by the user. This happened if, for example, a view was copied from the editor, a new view created and the copied text pasted into the new view. At this point the pasted text would be deleted by the auto-parsing facility of the View Editor. (Bug#52433)

- In the **REVERSE ENGINEER DATABASE** wizard, on the **CONNECTION OPTIONS** page, if the first empty connection was selected from the **STORED CONNECTION** drop-down list box, and then the **CONNECTION METHOD** changed, the fields on the **PARAMETERS** tab did not change accordingly. (Bug#51742)
- Each time an **ADMIN** tab was started an instance of cscript.exe was executed. However, when the **ADMIN** tab was closed the corresponding cscript.exe process was not terminated. This resulted in ever increasing numbers of cscript.exe processes, which consumed resources unnecessarily, and constituted a resource leak. (Bug#51601)
- When selecting Print Preview for an EER Diagram, the objects in the preview appeared primarily as black filled rectangles. (Bug#51513)
- In the **COLUMNS** tab of the Table Editor, the bottom button of the comments scrollbar was situated too low, making downward scrolling very difficult, as only part of the scrollbar button was exposed. (Bug#51496)
- In the **COLUMNS** tab of the Table Editor, if a comment was added to a column, or a column with a comment was selected, then when another table was double-clicked to load it into the Table Editor, the comment for the column previously selected was still displayed in the inactive **COMMENTS** area. (Bug#51495)
- If a SQL statement was selected in the SQL Statements area of the SQL Editor, and copied to the snippets list using the SAVE SQL TO SNIPPETS LIST toolbar button, then the statement was only partially saved, the beginning of the statement being missing. This only happened for lines after the first line. Also, the further down the copied statement was located, the less text was successfully copied to the snippets list. This resulted in statements further down the code being heavily truncated, with only the latter portion of the statement being saved. (Bug#51474)
- If a snippet was deleted from the Snippets tab in the SQL Editor, then after MySQL Workbench was restarted the deleted snippet would reappear, as if it had never been deleted. (Bug#51335, Bug#52558)
- In the **REVERSE ENGINEER DATABASE** wizard, on the **SELECT SCHEMATA** page, any schema name containing an underscore was displayed without the underscore character. Further, if the **Alt** key was pressed, the underscores appeared under the first character located after where the underscore should have been displayed, giving the character the appearance of being an accelerator key. (Bug#51141, Bug#52965)
- In the COLUMNS tab of the Table Editor, if a column was right-clicked on, and then <u>MOVE UP</u> selected, the column ordering was not updated within the COLUMNS tab, until the area was clicked again. (Bug#51139)
- The MySQL Workbench make targets, with the exception of make all, were broken by the file ext/ctemplate/Makefile. (Bug#51024)
- In the SQL Editor, when a new tab was created, the focus was on the Object Explorer, rather than in the SQL Statements area. This meant that the user had to press the Tab key, or click within the SQL Statements area before they could start typing SQL code. (Bug#50711)
- MySQL Workbench crashed when the root user, located in the Server Access Management tab of the Accounts facility in the Administrator, was clicked. (Bug#50703)
- SQL Editor syntax highlighting did not correctly recognize escaping of the single quote character ('). Queries such as SELECT '\' FROM DUAL; were therefore not highlighted correctly. (Bug#50324)
- When using the Forward Engineer SQL Script wizard, if an existing script file was selected to be overwritten, the wizard would not continue, the file had to be deleted first. (Bug#46920)

A.1.12. Changes in MySQL Workbench 5.2.19 (16 April 2010 beta)

Ninth public Beta release. This section documents all changes and bug fixes that have been applied since the release of 5.2.18.

Functionality added or changed:

• In the Configuration tab of the Administrator, the listbox showing the section in the configuration file has been changed to a readonly control. (Bug#52443)

Bugs fixed:

- In the SQL Editor, when altering a routine, if the Apply SQL Script button was clicked, then MySQL Workbench froze. (Bug#52853)
- When trying to open a saved model file, MySQL Workbench generated the following error:

error executing plugin wb.file.openModel: Internal error: wait box creation must be done on the main thread

(Bug#52851)

- Double-clicking on a model file (*.mwb) to open it caused MySQL Workbench to crash. (Bug#52838)
- When opening a connection to start querying from the Home screen, the HUD displayed correctly. However, an error occurred with the connection, and the error dialog was displayed beneath the HUD. This meant it was not possible to click the dialog, although pressing **ESC** cleared it. The error dialog should have been displayed in front of the HUD, making it clickable. (Bug#52812)
- MySQL Workbench crashed when carrying out most tasks in the SQL Editor including adding tables, adding columns, and altering tables. (Bug#52789)
- The MySQL Workbench preference to locate the sidebar on the right did not work. Further, the TOGGLE SIDEBAR button did not function. (Bug#52631, Bug#53072)
- Attempting to build MySQL Workbench using the LDFLAGS="-Wl, --as-needed" linker options failed. (Bug#52570)
- MySQL Workbench listed gtkmm 2.4 as a dependency in configure.in. However, it used features only available in later versions of this library. (Bug#52406)
- In the Object Browser of the SQL Editor, if a schema was dropped, the schema concerned was not removed from the Object Browser, but another schema was. However, if the connection was synchronized, the Object Browser displayed the correct information. (Bug#51919)
- In the Data Dump facility of the Administrator, if Export to Self-Contained file was selected and a destination file explicitly chosen, the selected schema was deselected on return from the file chooser. (Bug#51797)
- MySQL Workbench crashed when creating a foreign key relationship in the EER Diagram view. (Bug#51602)
- In the SQL Editor, the syntax highlighter coloured two dashes as a comment. This was incorrect, as two dashes should only be highlighted as a comment when followed by a space. (Bug#51596)
- The internal script MySQL Workbench used to add a new user to MySQL Server was incorrect. After MySQL Workbench was started, it was possible to create a new user, but attempts to create additional users resulted in an unhandled exception. (Bug#50947)
- In the Table Editor of the MySQL Model page, and the Alter Table dialog of the SQL Editor, the text label "comments:" was truncated to "commen". (Bug#50765)
- In the EER Diagram view, using the context-sensitive menu items <u>COPY TABLE</u>, or <u>EDIT IN NEW WINDOW</u> resulted in the following error:

```
plugin:wb.edit.editSelectedFigureInNewWindow
Cannot execute plugin
A model diagram must be selected.
```

(Bug#50649)

- MySQL Workbench appeared to perform a Save As operation rather than a Save operation when saving a modified script file in the SQL Editor. Also, there was no indication that the file being worked on had unsaved changes (this is normally indicated by '*' in the title bar). Finally, the filename was not displayed on the SQL Editor tab. (Bug#50055, Bug#51373)
- The FILE, OPEN RECENT menu item was not being populated with recently opened script files. (Bug#50053)
- When two routine groups were being edited, and in both cases the **ROUTINES** tab was selected to display the SQL code, switching between the routine groups did not result in the target **ROUTINES** tab contents being updated. For example, if routines1 was being displayed and the routines2 selected, the routines1 code continued to be displayed in the **ROUTINES** tab. (Bug#49432)
- In the SQL Editor, the Alter Schema dialog had a comments field. However, the comments entered were not used in the generated DDL code.

MySQL Workbench has been changed so this comments field no longer appears. (Bug#49280)

- In the EER Diagram view, when typing a new value into the Navigator Zoom input control, the zoom level of the diagram changed, as the new value was being typed. The diagram should only have been redrawn when the new value being typed in was fully entered by pressing the **Enter** key, instead of changing dynamically as the zoom level value was being typed. (Bug#48597)
- In the EER Diagram view, if the cardinality of a relationship was changed and then an attempt made to edit a second relationship, the cardinality of the first relationship was applied to the second relationship. (Bug#46906)
- The ASSIGNED PRIVILEGES list on the PRIVILEGES tab in the View Editor lacked default grant options such as SELECT, UPDATE, and DROP. (Bug#42157)

A.1.13. Changes in MySQL Workbench 5.2.18 (13 April 2010 beta)

Eighth public Beta release. This section documents all changes and bug fixes that have been applied since the release of 5.2.17.

Functionality added or changed:

 MySQL Workbench now features the ability to copy field data in the SQL Editor results tab in quoted or unquoted mode. (Bug#51041)

Bugs fixed:

- When using the Forward Engineer SQL Script wizard, the wizard did not terminate DROP procedure IF EXISTS statements with a semi-colon. This caused SQL syntax errors if the script was executed on a server. (Bug#52743)
- In the SQL Editor, if a routine was altered in the routine editor, it was possible to close the editor without the editor prompting you to save any changes made, and so changes were lost. (Bug#52728)
- The Triggers tab in the Alter Table dialog of the SQL Editor erased all typed code, when an attempt was made to copy and paste text within the tab. (Bug#52587)
- The Data Dump facility in the Administrator exported the entire schema into a single file when the **DUMP VIEWS** option was selected. Further, if tables were selected and the **DUMP STORED ROUTINES** option selected, the routines would be stored in their own file, but with the DROP PROCEDURE IF EXISTS statements missing. (Bug#52579)
- During the import procedure of the Data Dump facility in the Administrator, stored routines were not imported. (Bug#52577)
- The Connection Information panel in the SQL Editor did not display information for the User field. (Bug#52560)
- MySQL Workbench crashed when an attempt was made to create an EER Diagram from a database containing a large number of tables (1500+). (Bug#52500)
- MySQL Workbench crashed if No in the Delete Object dialog was clicked when attempting to delete a view from an EER Diagram. (Bug#52310)
- In the SQL Editor, if in the Alter Routine dialog, the user had a routine that contained an error, the routine would be lost if an attempt was made to close the Alter Routine dialog using the close button, and then click YES to apply changes. (Bug#51921)
- In the Alter Table dialog of the SQL Editor, in the columns or indexes tab, the first right-click in the dialog central area failed to display the context-sensitive menu. However, it was displayed on subsequent clicks. (Bug#51796)
- In the SQL Editor, it was not possible to export a result set, if the result set was not the first result set. For example, if there were two result sets, it would be possible to export the first one created, but not the second result set. (Bug#51595)
- In the Object Explorer of the Sidebar, if an existing table was right-clicked, the context-sensitive menu displayed the option <u>CREATE</u> <u>TABLE...</u> However, selecting this option appeared to have no effect. (Bug#51570)
- In the SQL Editor, in the results editor, it was not possible to edit or copy data containing multiple lines of text. Further, lines containing new lines were displayed as if the new lines did not exist.

MySQL Workbench now includes a blob editor. This can be accessed by right-clicking in the field to edit and selecting <u>OPEN</u> <u>VALUE IN EDITOR</u>. (Bug#51561)

- In the Alter Routine dialog of the SQL editor, there was inconsistency between the operation of the close dialog button ('X') on the top right of the window, and the CLOSE button on the bottom right. When using the close dialog button you were prompted to save changes, regardless of whether you actually changed the routine or not. When using the CLOSE button, the user was not prompted, even if the routine had changed. (Bug#51518)
- The Message Log text area in the Forward Engineer to Database wizard was in active, preventing scrolling of the message when an error occurred. (Bug#51417)
- In the SQL Editor, right-clicking a row in the results editor (after double clicking a table), and selecting delete from the contextsensitive menu, did not have any effect. (Bug#51361)
- The scrollable boxes on the Home screen did not respond to mouse wheel events. (Bug#51213)
- When editing a row in the SQL Editor, if a column was NOT NULL and had a default value, and a value was not entered for it, then after changing another column if changes were applied an error would be generated. This was because the editor attempted to set the NOT NULL column to NULL, rather than to its default value. (Bug#50781)
- The Log text area in the REVERSE ENGINEER SQL SCRIPT wizard was inactive, preventing the log message from being scrolled. (Bug#50758)
- The model Navigator did not redraw itself correctly. The Navigator worked correctly for the first model loaded into MySQL Workbench, but not for subsequently loaded models. (Bug#50580, Bug#50622)
- When editing a table in the SQL Editor, the context sensitive menu items did not affect the row under the cursor, but a previously selected row. This potentially led to the unintentioned deletion of a row. (Bug#50113)
- On the Linux platform, no context-sensitive menu was displayed when right-clicking on a schema tab in the Physical Schemata section of the MySQL Model page. (Bug#49429)
- In the EER Diagram view, when a table with a name containing underscores was right-clicked, the context-sensitive menu displayed menu items containing the table name without the underscores. (Bug#49314)
- In the MySQL Model page it was not possible to delete a schema by using the context sensitive menu in the Physical Schemata section. (Bug#48055)

A.1.14. Changes in MySQL Workbench 5.2.17 (02 April 2010 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:

- The Execute Current Statement command in the SQL Editor has been improved. The statement that contains the cursor will be executed. Further, if the cursor is outside of a statement, the last statement before the cursor is executed. (Bug#52302)
- 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, Bug#51844)
- MySQL Workbench has been changed so that it is possible to run the start, stop, and check status scripts executed from the Administrator, with administrator privileges. A checkbox has been provided to enable you the option of acquiring the administrator rights to execute the commands. This option is switched off by default for the check status command. (Bug#51276)
- A description column has been added to the snippets table. (Bug#51010)
- The WeifenLuo library was removed. This was the cause of difficult to trace application crashes. (Bug#50706)
- 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)
- The MySQL Workbench installer has been changed to include a LAUNCH NOW checkbox. By default this is selected, so MySQL

Workbench will be launched when the installer exits. (Bug#50387)

- In the SQL Editor, the COMMENTS tab in the CREATE VIEW dialog has been removed. (Bug#49270)
- The Output window only appeared in the Model context. MySQL Workbench has been changed so that the Output window now has its own tab on the main screen. (Bug#48988)

Bugs fixed:

• In the SQL Editor, the wrong table was displayed when using EDIT. This happened if two schemata contained tables with the same name. For example, if two schemata, schema1 and schema2 both contained a table t1, then if the following SQL was entered and executed, schema2.t1 would be displayed instead of schema1.t1:

use schemal; edit t1;

(Bug#52401, Bug#52692)

- If, in the EER Diagram view, an attempt was made to edit two tables, MySQL Workbench crashed. (Bug#52158)
- In the EER Diagram view, when the menu item <u>MODEL</u>, <u>DIAGRAM PROPERTIES AND SIZE</u> was selected from the main menu, MySQL Workbench crashed. (Bug#52065, Bug#52375)
- The Alter Table dialog in the SQL Editor displayed comments with a single quote character prepending it, and with the last character of the comment truncated. (Bug#51972, Bug#52297)
- MySQL Workbench crashed when synchronizing a model to a live server. (Bug#51892)
- MySQL Workbench crashed while attempting to edit code in the SQL Editor. (Bug#51815)
- 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 EDTOR. Editor was spelled incorrectly. (Bug#51790)
- When synchronizing a model containing a large number of Stored Routines, not all routines were synchronized with the live server. (Bug#51731)
- When synchronizing between two triggers, unexpected USE statements and the comment "-- Trigger DDL Statements" appeared intermittently. This caused a syntax error in the script. (Bug#51728)
- If a connection was opened for querying and then **Ctrl**+**W** quickly pressed to close the connection, MySQL Workbench would crash. (Bug#51685)
- 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)
- MySQL Workbench crashed if objects were sequentially selected in the EER Diagram view. (Bug#51573)
- In the SQL Editor, when a SQL query was entered that contained a large number of characters, it appeared full width in the action column of the Ouput tab, causing the Response column to be unacceptably narrow. (Bug#51550)
- In the Alter Table dialog of the SQL Editor, if a column was added to a table, and then an attempt made to alter that column's data type, then the wizard generated ADD COLUMN code, rather than CHANGE COLUMN. This resulted in the following error when an attempt was made to apply the changes:

ERROR 1060: Duplicate column name 'test_column'

(Bug#51516, Bug#51719)

- In the Alter Table dialog of the SQL Editor, when a table was edited and changes applied, if subsequent changes were made and applied an error would result, as MySQL Workbench attempted to use ADD COLUMN rather than CHANGE COLUMN. (Bug#51481)
- When MANAGE IMPORT / EXPORT was clicked on the Home screen, a server instance selected, and then OK clicked, MySQL Workbench displayed an error dialog with the following message:

MySQL Workbench has encountered a problem, External component has thrown an exception.

(Bug#51477, Bug#51665, Bug#51703, Bug#51733, Bug#51800, Bug#51870, Bug#51895, Bug#51963, Bug#51944, Bug#51999, Bug#52052, Bug#52262)

- Building MySQL Workbench from source failed if LDFLAGS="-W1, --as-needed" option was specified. (Bug#51469)
- If a long running SQL query was executed in the SQL Editor, and then the SQL Editor tab closed, MySQL Workbench crashed if a new connection was started from the Home screen. The following exception was generated:

```
Unknown Exception: caught in c:\documents and
settings\mysqldev\build\mysql-workbench-oss-5.2.16\backend\windows\wbprivate.wr\src\Wb.h
at line 1087
```

(Bug#51467)

- It was not possible to connect to the MySQL server with MySQL Workbench if using sockets. (Bug#51419, Bug#51460)
- When a model with a view was exported using <u>FILE, EXPORT, FORWARD ENGINEER SQL CREATE SCRIPT</u>, a semi-colon was added to the end of the <u>CREATE VIEW</u> statement, even if one was already present from the view code entered. (Bug#51416)
- On the MySQL Model tab, if a table was clicked on and edited using the context menu option <u>EDIT TABLE</u>, MySQL Workbench crashed. (Bug#51410, Bug#50936)
- If all screens and tabs were closed in MySQL Workbench and then <u>DATA</u>, <u>MANAGE CONNECTIONS</u> selected, then when the MAN-AGE 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.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, Bug#51476, Bug#51522, Bug#51483, Bug#51978, Bug#51810, Bug#51883, Bug#51803, Bug#52115, Bug#52163, Bug#51292, Bug#51668, Bug#51784, Bug#51789, Bug#51940, Bug#51947, Bug#52021, Bug#52028, Bug#52108, Bug#52240)

- After exporting a schema to disk, importing the schema did not restore the tables. (Bug#51261)
- In the Routine Editor, the editor added "//" as the last delimiter, even though "DELIMITER \$\$" was used at the beginning of the code. (Bug#51247)
- 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)
- On the Output tab of the SQL Editor, it was not possible to properly view the contents of the Action and Message columns, if the text exceeded the standard column width.

MySQL Workbench has now added a detail panel, so that long messages can be easily viewed. (Bug#51040)

- In the Administrator section of MySQL Workbench, if a new user was created, the password was not correctly applied for the account, resulting in the new user being denied access when an attempt was made to connect to a server with that user account. (Bug#50983, Bug#51464)
- An Out of Range exception occurred when switching from the Table Editor to the Home screen. (Bug#50980, Bug#51030)
- In the **OUTPUT** tab of the SQL Editor, if a query produced an error response, it was difficult to read the returned error message. The message could only be read using mouse-over in the **RESPONSE** column, and the error message text could not be copied.

MySQL Workbench has been changed so that the **RESPONSE** column is now renamed to **MESSAGE**. Further, the **MESSAGE** column has been widened, and new **ACTION** and **MESSAGE** detail panels have been added, making it easier to read error messages. (Bug#50629, Bug#50860)

- When using the main menu item <u>SCRIPTING</u>, <u>RUN WORKBENCH SCRIPT FILE</u>, the dialog appended ".lua" to the selected filename. This resulted in a failure to load the desired file. (Bug#50501)
- In the SQL Editor, field names containing an underscore were displayed incorrectly on the Results tab. Instead of the underscore being displayed in the correct location the following character appeared underlined. (Bug#50132)
- 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)
- In the SQL Editor, in a result set tab it was not possible to copy values from the result set.

MySQL Workbench has been changed to include the context sensitive menu items <u>COPY ROW VALUES</u> and <u>COPY FIELD CONTENT</u>. (Bug#49683, Bug#50170)

- The Table Editor, Routine Editor, and other object editors stayed open, even if the user switched from the MySQL Model page to the SQL Editor page, Home screen, or Administrator page, thereby causing confusion. (Bug#49367)
- In the live editing tab of the SQL Editor, if the ALT+TAB key combination was used, the ascending and descending sort order of the columns could no longer be changed. (Bug#49366)
- The live data editor in SQL Editor did not permit the columns to be sorted in descending order by clicking the column heading. It only permitted column sorting in ascending order through clicking the column heading. (Bug#49302)
- 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, Bug#51634)
- In the Alter Routine dialog of the SQL Editor, when a routine was changed, and the changes successfully applied using the Apply SQL Script to Database dialog, the Alter Routine dialog still prompted the user to apply changes, even though the changes had already been successfully applied. (Bug#49273)
- When switching from a tab such as the SQL Editor tab to the Home screen, the toolbar did not change accordingly. However, the buttons on the toolbar were still active, leading to the possibility of unintentional actions. (Bug#49147)
- 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)
- This bug occurred when using the MySQL Workbench Administrator with MySQL Server version 5.0. When selecting Enable General Log in the Administrator's configuration page, MySQL Workbench attempted to add the option general-log to the configuration file, even thought this is not supported by MySQL Server 5.0. (Bug#49011)

• In the Overview tab of the SQL Editor, if there were a large number of schemata, it was not possible to access certain schemata, as they did not all fit on the schema tab control.

MySQL Workbench has been changed so that there is now a small drop down control that enables you to select the required schema, even if it is not currently visible on the schema tab control. (Bug#48898, Bug#50169)

- In the EER Diagram view, when zooming in or out of the diagram the screen was not redrawn correctly. (Bug#48020, Bug#48032, Bug#34505)
- When using the option **OMIT SCHEMATA QUALIFIER IN OBJECT NAMES** in the Forward Engineer dialog, the schema name still appeared in DROP, CREATE SCHEMA, INSERT and USE statements. The schema name also appeared in the table comments. (Bug#46837, Bug#51411)

A.1.15. Changes in MySQL Workbench 5.2.16 (17 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 <u>FILE</u>, <u>EXIT</u>. (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 enable 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 mysgldump --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:

(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 <u>FILE, EXPORT, FORWARD ENGINEER ALTER SCRIPT</u> 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 <u>FILE</u>, <u>PRINT</u> 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.GrtTreeModel.TreeViewExpanding(Object sender, TreeViewAdvEventArgs e)
    at System.EventHandler`l.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 <u>DROP SCHEMA...</u> 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 th 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 <u>QUERY</u>, <u>EXPORT RESULTS</u> 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, enabling 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.1.16. Changes in MySQL Workbench 5.2.15 (28 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 data type 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 <u>FILE</u>, <u>PRINT</u> 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 numeric 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 to see the **ENTER SSH PASSWORD** sheet, but is 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 <u>FILE</u>, <u>EXPORT</u>, <u>EXPORT AS SINGLE PAGE PDF</u> or <u>EXPORT AS SINGLE PAGE POST</u><u>SCRIPT FILE</u>, 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 IN-TEGER would. (Bug#45834)

A.1.17. Changes in MySQL Workbench 5.2.14 (21 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 alphabetic order. (Bug#47143)

A.1.18. 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 <u>HELP</u>, <u>CHECK FOR UPDATES</u> 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[2]: 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/grt'
make[2]: *** [all-recursive] Error 1
make[1]: Leaving directory `/home/install/mysql-workbench-oss-5.2.11/library/grt'
make[1]: *** [all-recursive] Error 1
make[1]: 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/library'
make[1]: *** [all-recursive] Error 1
make[1]: 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/library'
make[1]: Eaving directory `/home/install/mysql-workbench-oss-5.2.11/library'
make[1]: Leaving directory `/home/install/mysql-workbench-oss-5.2.11/library'
make[1]: Leaving directory `/home/install/mysql-workbench-oss-5.2.11/library'
make[1]: Leaving directory `/home/install/mysql-workbench-oss-5.2.11/library'
make[1]: Leaving directory `/home/install/mysql-workbench-oss-5.2.11/library'
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 the 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.1.19. 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 permit 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 enable 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 <u>QUERY</u>, <u>EXPORT RESULTS...</u> 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 <u>COPY SQL TO CLIPBOARD</u> 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 textfield. 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 reversed 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 permitted the schema name to be changed using 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 <u>APPEND SELECTED ITEMS TO SQL SCRIPT</u> 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\\tmp047ttn'

Also, if the Logs tab was selected 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.1.20. Changes in MySQL Workbench 5.2.11 (18 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 enable 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 <u>VIEW</u>, <u>HOME</u> 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 <u>INSERT TEXT INTO SQL AREA</u>, 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:

Do you want to drop database `` from DB server ...

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:

DROP ``

(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 using 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 using the Data Dump tab of the Admin screen, and the export failed, the resultant SQL file was not deleted. (Bug#49296)
- 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 <u>PLUGINS</u>, <u>OBJECTS</u>, <u>COPY SQL TO CLIPBOARD</u> 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.1.21. Changes in MySQL Workbench 5.2.10 (01 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 <u>HELP</u>, <u>WORKBENCH PRODUCT PAGE</u> 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 <u>ALTER TABLE</u> 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 <u>ALTER TABLE</u> 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 permit 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)

• When a model containing stored routines was forward engineered to a script, the script contained an erroneous additional delimiter after each stored routine. This resulted in a "No query specified" error when an attempt was made to run the script on a server. (Bug#39929)

A.1.22. 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</pre>
```

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 <u>DATABASE</u>, <u>GENERATE CATALOG DIFF REPORT</u> 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.1.23. Changes in MySQL Workbench 5.2.8 (18 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 permit 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 comfort-ably fit on a single page. (Bug#33919)

A.1.24. 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 checbox 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 <u>QUERY</u>, <u>EXPORT RESULTS...</u> 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 <u>FILE, EXPORT, FORWARD ENGINEER SQL CREATE SCRIPT</u> dropped a schema if it existed and then recreated 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.
 - 1. 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.
 - 2. 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.1.25. Changes in MySQL Workbench 5.2.6 (21 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.1.26. 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(ActiveDocumentClosingEventA
rgs e)
em WeifenLuo.WinFormsUI.Docking.DockPanel.CloseContent(IDockContent content)
em WeifenLuo.WinFormsUI.Docking.DockPane.CloseContent(IDockContent content)
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 TypeLoadEx-ception:

```
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.1.27. Changes in MySQL Workbench 5.2.4 (07 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 <u>MODEL</u>, <u>RELATIONSHIP NOTATION</u>, <u>CONNECT TO COLUMNS</u> 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 <u>QUERY DATABASE</u> 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:

(Bug#47044)
- On Mac OS X using cmd + backspace to delete a column in the table editor did not work. (Bug#46613)
- In the SYNCHRONIZE MODEL WITH DATABASE dialog the text in the upper right area of the dialog did not wrap correctly, resulting in some text not being visible. (Bug#45455)
- MySQL Workbench EER Diagram view did not permit the creation of a foreign key constraint on a primary key. (Bug#39546)

A.1.28. Changes in MySQL Workbench 5.2.3 (15 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 quotation marks, 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 quotation marks 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 IN-TEGER(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 <u>TOOLS</u>, <u>PREFERENCES</u> 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 <u>HELP</u>, <u>ABOUT</u> 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.1.29. Changes in MySQL Workbench 5.2.2 (27 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 mispelled 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.1.30. Changes in MySQL Workbench 5.2.1 (22 May 2009 alpha)

This section documents all changes and bug fixes that have been applied since the release of 5.2.0.

A.1.31. Changes in MySQL Workbench 5.2.0 (30 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 (06 September 2010 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 permit the creation of a foreign key constraint on a primary key. (Bug#39546)

A.2.2. Changes in MySQL Workbench 5.1.18 (03 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 enable 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 <u>FILE</u>, <u>EXPORT</u>, <u>FORWARD ENGINEER SQL CREATE SCRIPT</u>. 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 data type of a table column was changed in the Table Editor, it was not reflected in the EER Diagram, although the tooltip was udpated 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 Workbench crashed with the following error:

```
** (mysql-workbench-bin:15234): WARNING **: /tmp/sakila.mwbdl/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 <u>FILE</u>, <u>EXPORT</u>, <u>FORWARD ENGINEER SQL CREATE SCRIPT</u> 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 permit 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 permit 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)
- In the SYNCHRONIZE MODEL WITH DATABASE dialog the text in the upper right area of the dialog did not wrap correctly, resulting in some text not being visible. (Bug#45455)
- The DBDoc documentation generation system did not support facilities for Stored Routines, Views and Triggers. (Bug#41589)
- After using the <u>HELP</u>, <u>UPDATE...</u> 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 comfort-ably fit on a single page. (Bug#33919)

A.2.3. Changes in MySQL Workbench 5.1.17 (14 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 <u>HELP</u>, <u>VIEW REPORTED BUGS</u> 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 <u>PLUGINS, CATALOG, DUMP ALL TABLE COLUMNS</u> 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 <u>PLUGINS, CATALOG, GIVE A PREFIX TO ALL TABLES IN CATALOG</u>, 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-9dle-0800272fd858' <object Grt0bject> key=owner could not be resolved
** Message: item_activated: 0xadbb608 -> 'plugin:wb.file.newDocument'
** 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 <u>TOOLS</u>, <u>PREFERENCES</u> 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 <u>COPY SQL TO CLIPBOARD</u> 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 mispelled 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 <u>TOOLS</u>, <u>PREFERENCES</u>, 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 data type 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 <u>CONNECT TO COLUMNS</u> 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 <u>HELP</u>, <u>ABOUT</u> 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 (30 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 <u>TOOLS</u>, <u>OPTIONS...</u> menu item has been renamed to <u>PREFERENCES...</u>, 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 <u>TOOLS</u>, <u>PREFERENCES</u> 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 <u>FILE, EXPORT, FORWARD ENGINEER SQL CREATE SCRIPT</u> or <u>DATABASE</u>, <u>FORWARD</u> <u>ENGINEER...</u>, 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 regenerated. (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 apearance 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 (26 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 <u>FILE, EXPORT, FORWARD ENGINEER SQL CREATE SCRIPT</u> 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 <u>SYNCRONIZE MODEL</u> 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 <u>EDIT IN NEW WINDOW...</u> from the context-sensitive menu, had no effect. (Bug#34069)

A.2.6. Changes in MySQL Workbench 5.1.14 (19 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 (12 June 2009 beta)

This section documents all changes and bug fixes that have been applied since the release of 5.1.12.

- 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 $\ensuremath{\mathtt{MySQLW}}\xspace$ where the set of the set of

(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 <u>FILE, EXPORT, FORWARD ENGINEER SQL CREATE SCRIPT</u> 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, <u>EDIT RELATIONSHIP</u> 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 generated 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 MySOL 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 permit 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: <u>PAGE SETUP</u>, <u>PRINT PREVIEW</u> and <u>PRINT</u>.

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)

- ٠ Deleting or cutting a Relation from an EER diagram crashed MySQL Workbench. (Bug#44340)
- When Relations were deleted using 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</br/>sigc::pointer_functor0</br/>void>,void>.call_it(slot_rep* rep) at MySQL.Grt.Db.TableEditorBE.remove_fk(NodeId fk) 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.ToolStripMenuItem.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 CAN-CEL 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 <u>FILE</u>, <u>EXIT</u> from the main menu. (Bug#44267)

- The <u>COPY SQL TO CLIPBOARD</u> 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 quotation marks 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#4385)
- 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 <u>FILE</u>, <u>EXPORT</u>, <u>FORWARD ENGINEER SQL CREATE SCRIPT</u> 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 line endings. These are now normalized to n in the reverse engneering 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 (27 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 (10 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 using 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#44043)
- When a nonstandard port was specified in <u>DATABASE</u>, <u>FORWARD ENGINEER...</u>, such as 3307, MySQL Workbench still attempted 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 <u>TOOLS</u>, <u>INSTALL PLUGIN/MODULE FILE</u>... 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 <u>FILE</u>, <u>NEW</u> 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\wbprivate.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 In-

noDB. (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 <u>FILE, IMPORT, REVERSE ENGINEER MYSQL CREATE SCRIPT</u>, 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:

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:

```
MySQLWorkbench [14915]
/Applications/MySQLWorkbench.app/Contents/MacOS/MySQLWorkbench
Process:
Path:
Identifier:
                     com.sun.MySQLWorkbench
Version:
                    ??? (???)
X86 (Native)
Code Type:
Parent Process:
                    launchd [95]
Date/Time:
                     2009-02-02 18:53:52.120 +0100
                    Mac OS X 10.5.6 (9G55)
OS Version:
Report Version:
Exception Type: EXC_BREAKPOINT (SIGTRAP)
Exception Codes: 0x00000000000002, 0x00000000000000
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 alphabetic 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 <u>FILE</u>, <u>IMPORT</u>, <u>REVERSE ENGINEER SQL CREATE SCRIPT</u> 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 <u>FILE</u>, <u>IMPORT</u>, <u>REVERSE ENGINEER SQL CREATE SCRIPT</u> 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 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 (18 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 <u>FILE, EXPORT, FORWARD ENGINEER SQL CREATE SCRIPT</u> wizard to crash. Further, peforming a <u>PLUGINS, OBJECTS, COPY SQL TO CLIPBOARD</u> 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 <u>FILE</u>, <u>EXPORT</u>, <u>FORWARD ENGINEER SQL CREATE SCRIPT</u> 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 (12 December 2008)

This section documents all changes and bug fixes that have been applied since the release of 5.0.28.

- 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 (06 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. 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)
      MySQL.Utilities.WindowsCanvasViewerPanel.OnMouseUp(MouseEventArgs e)
    à
    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)
      System.Windows.Forms.NativeWindow.Callback(IntPtr hWnd, Int32 msg, IntPtr wparam,
IntPtr lparam)
```



A.3.4. Changes in MySQL Workbench 5.0.27 (07 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 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 <u>COPY INSERT TO CLIPBOARD</u> 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 (16 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 <u>FILE, EXPORT, FORWARD ENGINEER SQL CREATE SCRIPT</u> 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 <u>FORWARD ENGINEER SQL ALTER SCRIPT</u> did not result in a RENAME statement. Instead, DROP and CREATE statements were generated. (Bug#39256)
- The script generated by the <u>EXPORT</u>, <u>FORWARD ENGINEER SQL CREATE SCRIPT</u> 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 (12 September 2008)

This section documents all changes and bug fixes that have been applied since the release of 5.0.24.

- If the user closed all tabs and then quit, Workbench crashed. (Bug#39346)
- Foreign keys referencing a deleted table were not removed. (Bug#39150)
- <u>FORWARD ENGINEER SQL CREATE SCRIPT</u> and <u>FORWARD ENGINEER SQL ALTER SCRIPT</u> generated scripts that did not put index names in quotation marks. (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 (12 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)

- Indexes listed when the INDEX tab was selected could not be deleted if the index type was FOREIGN. (Bug#38639)
- When the menu item <u>MODEL</u>, <u>VALIDATION</u>, <u>VALIDATE ALL</u> 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 <u>FORWARD ENGINEER SQL CREATE SCRIPT</u> 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 <u>MODEL</u>, <u>VALIDATION(MYSQL</u>), <u>VALIDATE ALL</u> 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 <u>PLUGINS</u>, <u>OBJECTS</u>, <u>COPY SQL TO CLIPBOARD</u> did not work. (Bug#37736)
- When synchronizing the database, table comments were not updated. However, column comments worked as expected. (Bug#37686)
- Running <u>HELP</u>, <u>UPDATE...</u> crashes Workbench when the wizard comes to the point where it is trying to close Workbench. (Bug#37665)
- <u>DATABASE</u>, <u>SYNCHRONIZE</u> 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 DE-FAULT 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 <u>REVERSE ENGINEER SQL ALTER SCRIPT</u> and the database name changed in Workbench, the script then generated by <u>FORWARD ENGINEER SQL ALTER SCRIPT</u> 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 autoincrement is invalid; for example, char. When the model was exported using <u>EXPORT</u>, <u>FORWARD ENGINEER SQL CREATE SCRIPT</u>, 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 (25 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)

- The <u>FILE, EXPORT, FORWARD ENGINEER SQL CREATE SCRIPT</u> menu item exports a script it is then unable to import using the <u>FILE, EXPORT, REVERSE ENGINEER MYSQL CREATE SCRIPT</u> 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 <u>GENERATE SCHEMA DIFF REPORT</u> 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 <u>GENERATE SCHEMA DIFF REPORT</u> 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 <u>PLACE A NEW TABLE</u>, the error message dialog is displayed again. (Bug#37079)

- The <u>FORWARD ENGINEER</u> 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 <u>FILE</u>, <u>EXPORT</u>, <u>FORWARD ENGINEER SQL ALTER SCRIPT</u> 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 <u>FILE, EXPORT, FORWARD ENGINEER SQL ALTER SCRIPT</u> menu item is selected it causes an ALTER Script Generation (Script Synchronization) error. (Bug#36355)
- The behavior of the <u>SYNCHRONIZE</u> 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 <u>DATABASE SYNCHRONIZATION</u> resulted in erroneous ALTER statements being generated. (Bug#34812)
- Menu item was incorrectly named <u>GENERATE SCHEMA DIFF REPORT</u>, when it should have been called <u>GENERATE CATALOG DIFF</u> <u>REPORT</u>. (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 (27 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 <u>FILE</u>, <u>EXPORT</u>, <u>FORWARD ENGINEER SQL CREATE SCRIPT</u> menu item contains a spurious quotation 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 permits 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 <u>EDIT</u> 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 (27 April 2008)

This section documents all changes and bug fixes that have been applied since the release of 5.0.20.

- 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 <u>FILE, EXPORT, FORWARD ENGINEER SQL ALTER SCRIPT</u> and <u>FILE, EXPORT, FORWARD ENGINEER</u> <u>SQL CREATE SCRIPT</u> include unnecessary SQL code. (Bug#36170)

- The <u>COPY CONNECTION NN</u> menu item on the context menu of a connection does not have a complementary <u>PASTE CONNECTION</u> menu item. The <u>EDIT</u> menu has a greyed-out <u>PASTE CONNECTION</u> menu item. (Bug#36166)
- When a schema used InnoDB, and then was switched to use MyISAM, the script generated by FORWARD ENGINEER SQL CRE-ATE 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`.`tablel` CHANGE COLUMN `id_tablel` `id_tablel` 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 (26 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 Progess/Results Advanced dialog had no scroll bar. (Bug#36192)

A.3.12. Changes in MySQL Workbench 5.0.19 (15 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 <u>MODEL</u> -> <u>DBDoc</u> -> <u>MODEL REPORTING</u> 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 <u>FILE</u> -> <u>EXPORT</u> -> <u>EXPORT</u> AS <u>PNG</u> 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 permitted 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 (07 April 2008)

This section documents all changes and bug fixes that have been applied since the release of 5.0.16rc.

- 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, permitting 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 relationship 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 dissappear 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 <u>EXPORT</u>, FORWARD ENGINEER ALTER SCRIPT from the <u>FILE</u> would open a **SQL SCRIPT SYNCRHRONIZATION** 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 (26 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 <u>OPTIONS</u> from the <u>MODEL</u> 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 permit you to edit the value when you click 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 <u>OPTIONS</u> optin in the <u>MODEL</u> 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 (17 March 2008)

This section documents all changes and bug fixes that have been applied since the release of 5.0.14abeta.

- 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 updated 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 sizer 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 permits 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 indic-

ated 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 digram 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 (28 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. DBDesigner files can now be imported without incident. (Bug#33588)

A.3.18. Changes in MySQL Workbench 5.0.14beta (25 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:

- <u>EXPAND ALL</u> and <u>COLLAPSE ALL</u> menu options have been added under the <u>ARRANGE</u> menu. The <u>EXPAND ALL</u> 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 <u>COLLAPSE ALL</u> menu option. Some object notations, such as <u>Classic</u>, do not permit expansion or contraction. <u>COLLAPSE ALL</u> undoes the operation performed by <u>EXPAND ALL</u>.
- A <u>FIT OBJECTS TO CONTENTS</u> option has been added under the <u>ARRANGE</u> 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 <u>SYSTEM INFORMATION</u> menu option has been added to the <u>HELP</u> menu. This option displays information about your system that is useful when reporting a big.
- 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.

- 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 <u>UNDO</u> menu option, the connection lines between related tables no longer showed up. (Bug#34601)
- When choosing the <u>EXPORT AS PNG</u> menu option the file dialogue box file type was All Filesinstead of PNG. The same was true for <u>EXPORT AS SINGLEPAGE PDF</u> and <u>EXPORT AS SINGLEPAGE PS</u>. 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 <u>GENERATE SCHEMA DIFF REPORT</u> 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 ori-

ginal. 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 <u>MODEL</u>, <u>DIAGRAM SIZE</u>... 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 manualSizingto 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)

Appendix B. Third Party Licenses

Use of any of this software is governed by the terms of the licenses that follow.

MySQL Workbench

- Section B.2, "Boost Library License"
- Section B.1, ".NET Flat TabControl License"
- Section B.3, "cURL (libcurl) License"
- Section B.4, "Python License"

B.1. .NET Flat TabControl License

The following software may be included in this product:

```
• .NET Flat TabControl
```

Use of any of this software is governed by the terms of the license below:

```
It is free. Public domain!
Oscar Londono
```

B.2. Boost Library License

The following software may be included in this product:

Boost C++ Libraries

Use of any of this software is governed by the terms of the license below:

Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

B.3. cURL (libcurl) License

The following software may be included in this product:

cURL (libcurl)

Use of any of this software is governed by the terms of the license below:

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1996 - 2009, Daniel Stenberg, <daniel@haxx.se>. All rights reserved.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABLLITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

B.4. Python License

The following software may be included in this product:

Python Programming Language

This is the official license for the Python 2.7 release:

A. HISTORY OF THE SOFTWARE

Python was created in the early 1990s by Guido van Rossum at Stichting Mathematisch Centrum (CWI, see http://www.cwi.nl) in the Netherlands as a successor of a language called ABC. Guido remains Python's principal author, although it includes many contributions from others.

In 1995, Guido continued his work on Python at the Corporation for National Research Initiatives (CNRI, see http://www.cnri.reston.va.us) in Reston, Virginia where he released several versions of the software.

In May 2000, Guido and the Python core development team moved to BeOpen.com to form the BeOpen PythonLabs team. In October of the same year, the PythonLabs team moved to Digital Creations (now Zope Corporation, see http://www.zope.com). In 2001, the Python Software Foundation (PSF, see http://www.python.org/psf/) was formed, a non-profit organization created specifically to own Python-related Intellectual Property. Zope Corporation is a sponsoring member of the PSF.

All Python releases are Open Source (see http://www.opensource.org for the Open Source Definition). Historically, most, but not all, Python releases have also been GPL-compatible; the table below summarizes the various releases.

Release	Derived from	Year	Owner	GPL- compatible?	(1)
0.9.0 thru 1.2 1.3 thru 1.5.2 1.6 2.0 1.6.1 2.1 2.0.1 2.2 2.1.2 2.1.3 2.2.1 2.2.2 2.3.2 2.3.2 2.3.3 2.3.1 2.3.2 2.3.3 2.3.4 2.3.5 2.4 2.4.2 2.4.2 2.4.3 2.5 2.5.1	$\begin{array}{c} 1.2\\ 1.5.2\\ 1.6\\ 2.0+1.6.1\\ 2.0+1.6.1\\ 2.1+2.0.1\\ 2.1.1\\ 2.1.1\\ 2.1.1\\ 2.1.2\\ 2.2\\ 2.2$	1991-1995 1995-1999 2000 2001 2001 2001 2001 2002 2002	CWI CNRI CNRI BeOpen.com CNRI PSF PSF PSF PSF PSF PSF PSF PSF PSF PSF	yes yes no yes (2) no yes yes yes yes yes yes yes yes yes yes	ye
2.5.2 2.5.1 2008 PSF	yes				
----------------------	-----				
2.5.3 2.5.2 2008 PSF	yes				
2.6 2.5 2008 PSF	yes				
2.6.1 2.6 2008 PSF	yes				
2.6.2 2.6.1 2009 PSF	yes				
2.6.3 2.6.2 2009 PSF	yes				
2.6.4 2.6.3 2010 PSF	yes				
2.7 2.6 2010 PSF	yes				

Footnotes:

- (1) GPL-compatible doesn't mean that we're distributing Python under the GPL. All Python licenses, unlike the GPL, let you distribute a modified version without making your changes open source. The GPL-compatible licenses make it possible to combine Python with other software that is released under the GPL; the others don't.
- (2) According to Richard Stallman, 1.6.1 is not GPL-compatible, because its license has a choice of law clause. According to CNRI, however, Stallman's lawyer has told CNRI's lawyer that 1.6.1 is "not incompatible" with the GPL.

Thanks to the many outside volunteers who have worked under Guido's direction to make these releases possible.

B. TERMS AND CONDITIONS FOR ACCESSING OR OTHERWISE USING PYTHON

PYTHON SOFTWARE FOUNDATION LICENSE VERSION 2

1. This LICENSE AGREEMENT is between the Python Software Foundation ("PSF"), and the Individual or Organization ("Licensee") accessing and otherwise using this software ("Python") in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, PSF hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python alone or in any derivative version, provided, however, that PSF's License Agreement and PSF's notice of copyright, i.e., "Copyright (c) 2001, 2002, 2003, 2004, 2005, 2006 Python Software Foundation; All Rights Reserved" are retained in Python alone or in any derivative version prepared by Licensee.

3. In the event Licensee prepares a derivative work that is based on or incorporates Python or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python.

4. PSF is making Python available to Licensee on an "AS IS" basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

 ${\bf 6}.$ This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between PSF and Licensee. This License Agreement does not grant permission to use PSF trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

 ${\bf 8}.$ By copying, installing or otherwise using Python, Licensee agrees to be bound by the terms and conditions of this License Agreement.

BEOPEN.COM LICENSE AGREEMENT FOR PYTHON 2.0

BEOPEN PYTHON OPEN SOURCE LICENSE AGREEMENT VERSION 1

1. This LICENSE AGREEMENT is between BeOpen.com ("BeOpen"), having an office at 160 Saratoga Avenue, Santa Clara, CA 95051, and the Individual or Organization ("Licensee") accessing and otherwise using this software in source or binary form and its associated documentation ("the Software").

2. Subject to the terms and conditions of this BeOpen Python License

Agreement, BeOpen hereby grants Licensee a non-exclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use the Software alone or in any derivative version, provided, however, that the BeOpen Python License is retained in the Software, alone or in any derivative version prepared by Licensee.

3. BeOpen is making the Software available to Licensee on an "AS IS" basis. BEOPEN MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, BEOPEN MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

4. BEOPEN SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF THE SOFTWARE FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THE SOFTWARE, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

 $5.\ {\rm This}\ {\rm License}\ {\rm Agreement}\ {\rm will}\ {\rm automatically}\ {\rm terminate}\ {\rm upon}\ {\rm a}\ {\rm material}\ {\rm breach}\ {\rm of}\ {\rm its}\ {\rm terms}\ {\rm and}\ {\rm conditions}\ .$

6. This License Agreement shall be governed by and interpreted in all respects by the law of the State of California, excluding conflict of law provisions. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between BeOpen and Licensee. This License Agreement does not grant permission to use BeOpen trademarks or trade names in a trademark sense to endorse or promote products or services of Licensee, or any third party. As an exception, the "BeOpen Python" logos available at http://www.pythonlabs.com/logos.html may be used according to the permissions granted on that web page.

7. By copying, installing or otherwise using the software, Licensee agrees to be bound by the terms and conditions of this License Agreement.

CNRI LICENSE AGREEMENT FOR PYTHON 1.6.1

1. This LICENSE AGREEMENT is between the Corporation for National Research Initiatives, having an office at 1895 Preston White Drive, Reston, VA 20191 ("CNRI"), and the Individual or Organization ("Licensee") accessing and otherwise using Python 1.6.1 software in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, CNRI hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python 1.6.1 alone or in any derivative version, provided, however, that CNRI's License Agreement and CNRI's notice of copyright, i.e., "Copyright (c) 1995-2001 Corporation for National Research Initiatives; All Rights Reserved" are retained in Python 1.6.1 alone or in any derivative version prepared by Licensee. Alternately, in lieu of CNRI's License Agreement, Licensee may substitute the following text (omitting the quotes): "Python 1.6.1 is made available subject to the terms and conditions in CNRI's License Agreement. This Agreement together with Python 1.6.1 may be located on the Internet using the following unique, persistent identifier (known as a handle): 1895.22/1013. This Agreement may also be obtained from a proxy server on the Internet using the following URL: http://hdl.handle.net/1895.22/1013".

3. In the event Licensee prepares a derivative work that is based on or incorporates Python 1.6.1 or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python 1.6.1.

4. CNRI is making Python 1.6.1 available to Licensee on an "AS IS" basis. CNRI MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, CNRI MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR FURPOSE OR THAT THE USE OF PYTHON 1.6.1 WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. CNRI SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON 1.6.1 FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON 1.6.1, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. This License Agreement shall be governed by the federal intellectual property law of the United States, including without limitation the federal copyright law, and, to the extent such U.S. federal law does not apply, by the law of the Commonwealth of Virginia, excluding Virginia's conflict of law provisions. Notwithstanding the foregoing, with regard to derivative works based on Python 1.6.1 that incorporate non-separable material that was previously distributed under the GNU General Public License (GPL), the law of the Commonwealth of Virginia shall govern this License Agreement only as to issues arising under or with respect to Paragraphs 4, 5, and 7 of this License Agreement. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between CNRI and Licensee. This License Agreement does not grant permission to use CNRI trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By clicking on the "ACCEPT" button where indicated, or by copying, installing or otherwise using Python 1.6.1, Licensee agrees to be bound by the terms and conditions of this License Agreement.

ACCEPT

CWI LICENSE AGREEMENT FOR PYTHON 0.9.0 THROUGH 1.2 $\,$

Copyright (c) 1991 - 1995, Stichting Mathematisch Centrum Amsterdam, The Netherlands. All rights reserved.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Stichting Mathematisch Centrum or CWI not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

STICHTING MATHEMATISCH CENTRUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL STICHTING MATHEMATISCH CENTRUM BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.