Microolap Database Designer for MySQL

User's guide

Chernogolovka, Russian Federation
2015

Contents

1. Introduction ... 1
2. Features ... 2
3. What's new in version 2.0 ... 6
4. Tutorials ... 8
 4.1. Creating Your First Diagram ... 8
 4.2. Using Domains ... 27
5. Interface User Guide ... 32
 5.1. Diagram Window ... 34
 5.2. Ribbon ... 34
 5.3. GUI Skins ... 35
 5.4. Docking Windows ... 38
 5.5. Keyboard Shortcuts ... 40
 5.6. Environment Options ... 42
 5.7. Minimap Navigator ... 46
 5.8. Text Editor ... 46
6. Diagram ... 48
 6.1. Notation ... 49
 6.2. Creating a New Diagram ... 50
 6.3. Open an Existing Diagram ... 50
 6.4. Saving a Diagram ... 50
 6.5. Selecting and Moving Objects ... 51
 6.6. Copying and Pasting Objects ... 52
 6.7. Using Snapping Grid ... 52
 6.8. Export to Graphics ... 53
 6.9. Diagram Display Preferences ... 54
 6.10. Diagram Properties ... 61
  6.10.1. "Default Database Options" tab ... 63
  6.10.2. "Pages" tab ... 64
  6.10.3. "Preview" tab ... 64
  6.10.4. "Notes" tab ... 65
  6.10.5. "Statistics" tab ... 65
 6.11. Zooming a Diagram ... 66
 6.12. Auto Layout Diagram ... 67
 6.13. Find Objects ... 67
7. Diagram Objects ... 69
 7.1. Database ... 70
  7.1.1. Database Editor ... 70
 7.2. Tables ... 71
  7.2.1. Creating a Table ... 72
  7.2.2. Table Editor ... 73

7.2.3. Formatting Table 76
7.2.4. SQL Table Definition 77
7.2.5. Partitions Editor 77
7.2.6. Tables in Tree View Window 79
7.2.7. Table Manager 80
7.3. Columns 82
7.3.1. Column Editor 83
7.3.2. Column Manager 85
7.4. Domains 87
7.4.1. Domain Manager 87
7.5. References and Foreign Keys 90
7.5.1. Creating a Reference 90
7.5.2. Creating a Many-to-Many Reference 91
7.5.3. Reference Editor 93
7.5.4. Reference Manager 96
7.5.5. Manual Reference Drawing 97
7.6. Indexes 99
7.6.1. Index Editor 100
7.6.2. Index Manager 102
7.7. Stored Procedures and Functions 103
7.7.1. Creating a Stored Procedure or Function 103
7.7.2. Stored Routine Editor 104
7.7.3. Stored Routine Manager 107
7.8. Triggers 108
7.8.1. Trigger Editor 109
7.9. Views 110
7.9.1. Creating a View 111
7.9.2. View Editor 111
7.9.3. View Manager 114
7.10. Events 115
7.10.1. Creating an Event 116
7.10.2. Event Editor 116
7.10.3. Event Manager 119
7.11. Object Groups 121
7.11.1. Creating an Object Group 123
7.11.2. Object Group Editor 123
7.11.3. Adding/Deleting objects to/from Group 125
7.11.4. Deleting Object Groups 127
7.12. Notes 127
7.13. Stamps 129
8. Diagram Functions 130
8.1. Merge Diagram 130

8.2. Check Diagrams ... 132
8.3. Compare Diagram ... 136
8.3.1. Compare Result dialog ... 138
8.4. Diagram Versioning (Revisions) ... 139
8.4.1. Creating Diagram Revision ... 140
8.4.2. Revision Manager ... 141
8.4.3. Comparing Diagram Revisions ... 141
8.5. Diagram Synchronization ... 142
9. Database Functions ... 144
9.1. Database Generation ... 144
9.2. Database Modification ... 154
10. Database Accessing Tools ... 157
10.1. Database Connection Manager ... 157
10.1.1. Connection Profile Editor ... 159
10.2. SQL Executor ... 162
10.3. Connect to a Database ... 163
10.4. Disconnect from a Database ... 164
11. Reverse Engineering and Import ... 164
11.1. Reverse Engineering MySQL Database ... 164
11.2. Import from Access Database ... 167
11.3. Universal Reverse Engineering (ADO, OLEDB, ODBC) ... 169
12. Reports ... 173
12.1. Create a Report ... 174
13. Printing a Diagram ... 175
13.1. Page Setup ... 176
13.2. Print Preview ... 180
14. How to ... 181
14.1. How to Connect to a Database ... 182
14.2. How to Disconnect from a Database ... 182
14.3. How to Create a New Diagram ... 183
14.4. How to Create a Table ... 183
14.5. How to Create a Reference and Foreign Key ... 184
14.6. How to Check a Diagram ... 185
14.7. How to Edit Columns ... 188
14.8. How to Edit Table Index ... 191
14.9. How to Generate a Database ... 192
14.10. How to Modify a Database ... 202
14.11. How to Import from a MySQL Database ... 205
14.12. How to Import from a Microsoft Access Database ... 207
14.13. How to Import from other Databases ... 210
14.14. How to Modify Multiple Tables ... 214
14.15. How to Execute an SQL Script ... 216

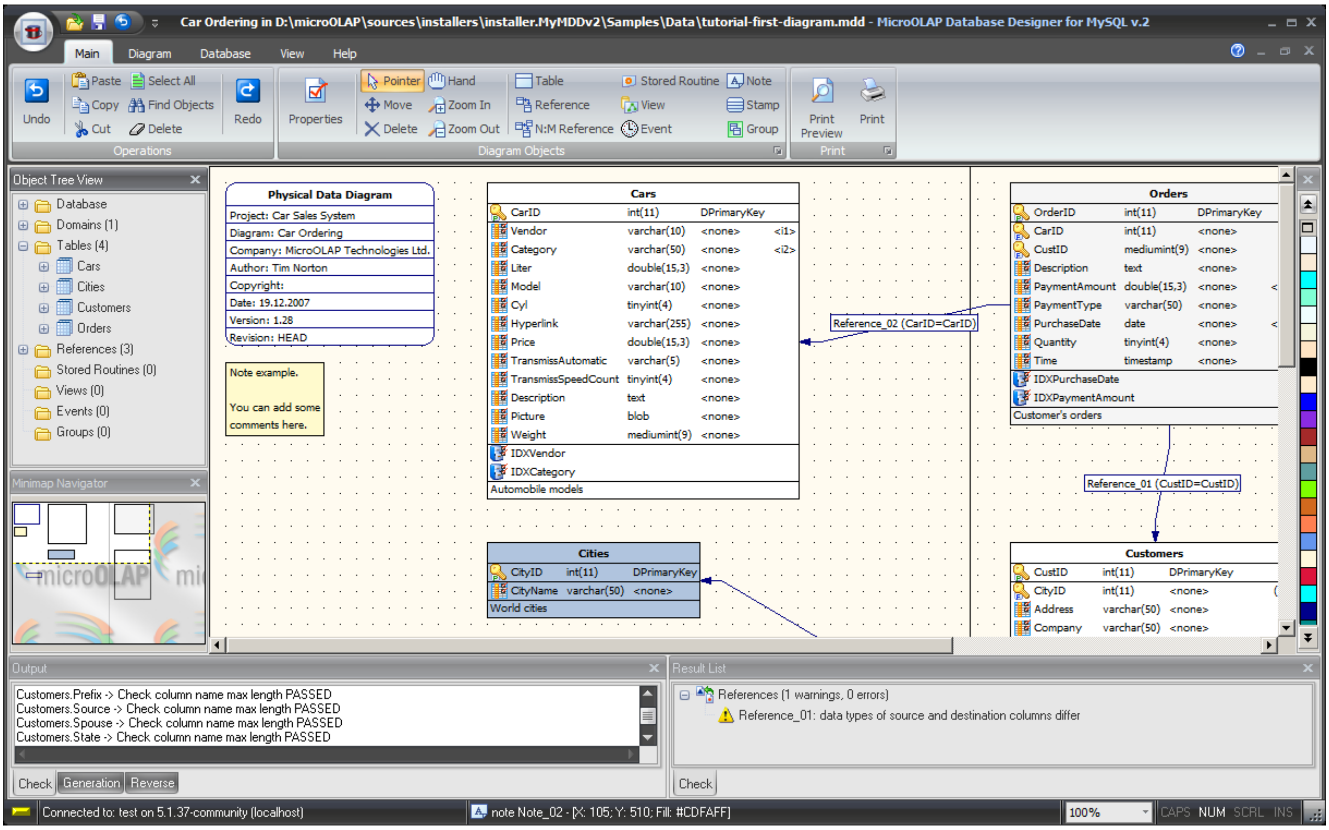
14.16. How to Merge Diagrams 218
14.17. How to Find Errors in a Diagram 220
14.18. How to Find Objects 224
14.19. How to Print a Diagram 225
14.20. How to Read a Diagram 226
14.21. How to View an SQL Table Definition 227
14.22. How to Export a Diagram to Graphics 228
14.23. How to Create a Domain 229
14.24. How to Insert a Comment 231
14.25. How to Change Table Formatting 233
15. Installation 234
16. Purchase 234
17. Support 234
18. License Agreement 234

1. Introduction

Welcome to Database Designer for MySQL

Database Designer for MySQL is a powerful solution for visual database development: creation, modification and reverse engineering. It greatly simplifies the process of database development and maintenance and makes it clear. By working with graphical representation of tables, columns, relations and other objects, you can build a database model that can be used to generate a physical database or to modify an existing one. The model can be automatically created from an existing database by reverse engineering.

Database Designer for MySQL is specially developed for MySQL database and takes into account its features, has build-in support of tables, triggers, references, stored procedures and functions, views, events, indexes, foreign keys and other database objects.



Physical Data Diagram
Project: Car Sales System
Diagram: Car Ordering
Company: MicroOLAP Technologies Ltd.
Author: Tim Norton
Copyright:
Date: 19.12.2007
Version: 1.28
Revision: HEAD

Note example.

You can add some comments here.

Cars			
CarID	int(11)	DPrimaryKey	
Vendor	varchar(10)	<none>	<i1>
Category	varchar(50)	<none>	<i2>
Liter	double(15,3)	<none>	
Model	varchar(10)	<none>	
Cyl	tinyint(4)	<none>	
Hyperlink	varchar(255)	<none>	
Price	double(15,3)	<none>	
TransmissAutomatic	varchar(5)	<none>	
TransmissSpeedCount	tinyint(4)	<none>	
Description	text	<none>	
Picture	blob	<none>	
Weight	mediumint(9)	<none>	
IDXVendor			
IDXCategory			
Automobile models			

Orders		
OrderID	int(11)	DPrimaryKey
CarID	int(11)	<none>
CustID	mediumint(9)	<none>
Description	text	<none>
PaymentAmount	double(15,3)	<none>
PaymentType	varchar(50)	<none>
PurchaseDate	date	<none>
Quantity	tinyint(4)	<none>
Time	timestamp	<none>
IDXPurchaseDate		
IDXPaymentAmount		
Customer's orders		

Reference_02 (CarID=CarID)

Reference_01 (CustID=CustID)

Cities		
CityID	int(11)	DPrimaryKey
CityName	varchar(50)	<none>
World cities		

Customers		
CustID	int(11)	DPrimaryKey
CityID	int(11)	<none>
Address	varchar(50)	<none>
Company	varchar(50)	<none>

```
Customers.Prefix -> Check column name max length PASSED
Customers.Source -> Check column name max length PASSED
Customers.Spouse -> Check column name max length PASSED
Customers.State -> Check column name max length PASSED
```

```
References (1 warnings, 0 errors)
  Reference_01: data types of source and destination columns differ
```

Connected to: test on 5.1.37-community (localhost)

note Note_02 - [X: 105; Y: 510; Fill: #CDFAFF]

Database Designer for MySQL allows you to construct your database in easy-to-understand visual environment by working with graphical representation of tables, columns, relations and so on. Finally you will get Physical Entity Relationship Model (ER Model) that can be exported in SQL script or directly built on working MySQL database server. After modification of model you can synchronize existing MySQL database with it.

The product allows to extract tables, attributes, relationships, indexes and other objects from existing databases including MySQL, Microsoft Access and Sybase ASE/ASA, Oracle, Informix, MSSQL, PostgreSQL, DB2, DBF and many others that accessible through OLEDB or ODBC.

You can generate reports with complete information about a developed model in printer-friendly HTML view.

Getting Started

To start using the product and find out more about it please refer to Creating Your First Diagram tutorial and short overview of Main Window Organization.

See also:
Tutorials: Tutorial - Creating Your First Diagram | Tutorial - Using Domains
Workspace: Interface User Guide
Diagram: Creating a New Diagram

2. Features

Database Designer for MySQL offers many features that allow you to be creative and productive in your work on database design and maintenance.

MySQL Native Support

Database Designer for MySQL provides native support for MySQL objects and data types. You can work with them in a very intuitive graphical interface. The central part of the application is a diagram that gives you a visual representation of the MySQL database.

The product supports all MySQL data types including TIMESTAMP, YEAR, TEXT, SET and other. Multiple table types and their features are also supported. You are able to manage table columns, indexes, references and other MySQL objects in the easiest way.

After you have completed designing the database diagram you can generate physical database directly on MySQL server and even modify the already existing local or remote database. Please, note that you can start working with the diagram by reverse engineering an existing MySQL database.

For further assistance please see the following topics:

Diagram overview

Main Window Organization

Database Generation

Reverse Engineering and Import overview.

Support of Modern MySQL Versions

Database Designer for MySQL supports 3.x, 4.x, 5.x and partially 6.x MySQL server versions.

Support of Stored Procedures, Functions, Views and Events

Database Designer for MySQL is the CASE tool, which supports MySQL 5.x+ innovative features such as stored procedures and functions, triggers, events and views.

For further assistance please see the following topics:

Stored Procedures and Functions

Creating a Stored Procedure

Views

Triggers

Events

Direct Connection to MySQL

Database Designer for MySQL has a unique ability to connect to MySQL databases directly without any slow and unreliable intermediates, such as ODBC or ADODB. It greatly improves the performance of such frequent operations as database synchronization and generation. Also this feature noticeably simplifies the process of setting up database connection: you don't have to tune any additional drivers such as MyODBC.

Direct connection to MySQL databases became possible due to the DAC for MySQL Delphi/C++Builder components from MicroOLAP Technologies.

Domains Support

The product has integrated Domains support for faster defining the required column properties. The domains are user-defined types, which contain all column information (data type, key options, dimensions and flags). This allows you to add and modify existing domain-based columns in multiple tables at once. Currently domains are not supported by MySQL, so it is a **Database Designer for MySQL** extension to the server.

For further assistance please see the following topics:

Using Domains tutorial

Domain Manager overview.

Easy Object Location and Modification

You can easily navigate through the diagram objects by using Object Tree View window. It contains combined information about all objects of your diagram and displays the diagram in a hierarchical tree structure. **Object Tree View** window shows you: **Domains**, **Tables**, **Columns**, **Indexes**, **References**, **Events**. To locate any diagram object quickly, right-click on it in the **Object Tree View** window and select '**Go to object**' from the context menu.

Also, **Database Designer for MySQL** provides you with very handy tools to help you modify significant parameters of multiple objects quickly:

Table Manager (Diagram | Table Manager menu item)

Column Manager (Diagram | Column Manager menu item)

Domain Manager (Diagram | Domain Manager menu item)

Index Manager (Diagram | Index Manager menu item)

Reference Manager (Diagram | Reference Manager menu item)

Stored Routine Manager (Diagram | Stored Routine Manager menu item)

View Manager (Diagram | View Manager menu item)

Event Manager (Diagram | Event Manager menu item).

Check Diagram Function

Database Designer for MySQL can check your diagram for consistency and find typical errors, which may occur during the process of database design and maintenance. In terms of convenience, this feature can be compared to Microsoft Word's spell checker. It saves your time greatly and gives you the opportunity to concentrate on the main development tasks.

For further assistance please see Check Diagram function.

Import From Multiple Data Sources

Database Designer for MySQL can generate your diagram from an existing database structure. It's possible by reverse engineering a MySQL database or importing a structure from other data sources.

Database Designer for MySQL allows you to import database objects from MySQL database, Microsoft Access and any ADO (OLEDB)-compatible datasources (Oracle, PostgreSQL, MS SQL and other)

For further assistance please see Reverse Engineering and Import Overview.

Report Generator

Database Designer for MySQL can generate comprehensive printable reports for you. This gives you the possibility of getting a hard copy of the report by printing it out. After that you can have your diagram approved by your colleagues and managers.

Generated reports contain information about all tables, their indexes, columns and references.

You can get reports in HTML format, please see Reports overview.

SQL Editor and Executor

With **Database Designer for MySQL** you do not need any external tools for executing simple SQL statements and queries. The **SQL Executor** tool allows you to modify the existing structure of a local or remote database and even view data by running **SELECT/INSERT/UPDATE/DELETE** SQL statements

For further assistance please see SQL Executor.

Full Customization of Diagram and Objects Appearance

The appearance of the diagram can be adjusted according to your requirements and wishes. You can set the colors of the object background, object lines and references. You can configure the representation of the table information: enable showing of column and indexes icons, displaying indexes, keys, foreign keys and other.

For further assistance please see Diagram Overview and Diagram Display Preferences.

Multi-document Interface

Database Designer for MySQL has Multi-Document Interface. This means that you can edit multiple documents simultaneously. You can place more than one diagram onto your work space, switch between them quickly, compare them, and drag objects from one to another

For further assistance please see Diagram Window.

Export of diagram to Graphical File

The product can export your diagram into multiple graphical formats: bitmap (**PNG, GIF, JPEG**) by using **File | Export | Image** menu item and vector (**EMF**) by using **File | Export | Vector Graphics** menu item. This allows you to use graphical representation of a diagram in your documents (i.e. use it in **Microsoft Word, LaTeX** and other), publish diagram to **Web sites**, create big **posters** with it.

For further assistance please see Export to Graphics.

Database Generation

When you have finished creating the database diagram, you get the possibility of generating SQL script for it (**Database | Generate Database**) and you can build the database on a local or remote MySQL server directly.

For further assistance please see Database Generation.

Database Modification

The Synchronize Database (**Database | Modify Database**) tool helps you not only generate a new database but, what is more important, synchronize the existing MySQL database with the diagram you are developing. **Database Designer for MySQL** will automatically generate all required SQL statements. Which means that you do not have to create complex and inconvenient ALTER TABLE statements manually; moreover it significantly reduces the complexity and improves the reliability of all database modification tasks. Whatever complex diagram you have, you can synchronize the database with it literally in two clicks

For further assistance please see Database Modification overview.

Database Connection Manager

You can store the connection profiles for your database within the **Database Connection Manager** and access any of them quickly. You do not have to remember multiple parameters of connections - just click on the connection profile and get access to the database

For further assistance please see Database Connection Manager overview.

See also:
Tutorials: Creating Your First Diagram, Using Domains

3. What's new in version 2.0

Version 2.0 of **Database Designer for MySQL** is new generation of this product. There are lot of new features, some old features were rewritten from scratch.

> **Please note:**
> Non-expired registration keys from 1.x version will work with version 2.0 too.

New Ribbon-based GUI

Graphical User's Interface (GUI) in version 2 is based on **Ribbon** (like that found in MS Office 2007). Ribbon is a set of **Tabs** with buttons, dropdown controls and other controls that provide you with access to the **Database Designer for MySQL** functions.

Please take a look at Ribbon topic for additional details.

GUI Skins

Now you can choose different GUI Skins for **Database Designer for MySQL** to appear according to your preferences.

There are several skin examples in GUI Skins topic.

New Compare Diagram tool

You can compare diagram with diagram, diagram with database and even database with another database now! You can also compare revisions of the same diagram (or even revisions of different diagrams if it makes any sense to you).

Details are in Compare Diagram topic.

Diagram versioning (diagram revisions)

Now you can create diagram snapshots called revisions. You can open previous revisions using new Revision Manager, save revisions to separate model file. Another useful feature is comparing diagram revisions using Compare Diagram tool to track changes of diagram between two revisions. You can also build SQL code according to compare results. This can be very useful for modifying real database after diagram modification.

Diagram versioning also is useful as backup solution when you want to save particular state of your diagram.

There is Diagram Versioning (Revisions) topic with more details on this.

New Modify Engine for Database Modification

Database Modification tool is completely rewritten in version 2.0. it is based on using Compare Diagram tool now. This allows to compare physical database with diagram more precisely, review all found differences and select ones you want to apply to database. Please take a look at Database Modification topic for details.

There is also new, more careful way of updating database structure according to diagram changes using one more new feature - Diagram Revisions. You can create revision of your diagram before changing it. And then apply only these changes to physical database rather then compare whole database structure to diagram.

There are additional details available in Comparing Diagram Revisions topic.

Diagram Synchronization with physical database changes

Version 2.0 brings one of long-awaited and most wanted features: Diagram Synchronization with physical database changes. **Database Designer for MySQL** compares physical database with your model, shows you list of differences and allows you to apply all or several of these differences to your model. For example you'd add column using "*ALTER TABLE tbl ADD ...*" query. With Diagram Synchronization you can get this new column in your database model.

This feature based on new Compare Diagram tool. This means that you can compare everything you want but not only physical database with your model. For example you can compare two revisions of *Model_A* and apply differences to *Model_B* and so on.

Additional details are available in Diagram Synchronization topic.

Manual drawing of references

One of the most wanted features. Now you can draw references manually point-by-point or let **Database Designer for MySQL** calculate them automatically as before. No more crossed references or hidden below table shape reference's labels!

Additional details are available in Manual Reference Drawing topic.

Ability to group objects on diagram

Starting from version 2.0 diagram objects can be united to object groups. Object Groups are the way to add more organization to your diagram. They allow to group several diagram objects to single object group giving it some name. All objects in this group will be moved with group. You can also delete whole group with or without all its objects.

Please take a look at Object Groups topic for additional details.

Unicode support

Full Unicode support for all editors and GUI is added. This means that you can use every character which your language supports including hieroglyph and other complicated character sets.

4. Tutorials

Here you can find a set of small tutorials for better understanding software features:

Creating Your First Diagram

Using Domains.

4.1. Creating Your First Diagram

Follow through the steps of this tutorial for a quick introduction to the process of modeling and modification databases with **Database Designer for MySQL**.

Table of Contents

Stage 1. Determining requirements

Stage 2. Creating a diagram

Stage 3. Creating a table

Stage 4. Setting table parameters

Stage 5. Defining table columns

Stage 6. Creating indexes

Stage 7. Creating and designing the rest of tables

Stage 8. Creating foreign keys

Stage 9. Checking the diagram

Stage 10. Generating the database

Stage 11. Altering the diagram

Stage 12. Synchronizing database with the diagram

Afterword

> **Please note:**
> The database diagram, which we develop in this section, is placed in the **tutorial-first-diagram.mdd** in the Samples directory in the program installation path.

Stage 1. Determining requirements

The aim of our tutorial is to develop the database for a simple car ordering system.

First, let's determine the requirements of our database:

- it should store primary information about cars we sell, such as price, vendor, model, total displacement, link to vendor's website, etc...
- we need information about our customers that have bought cars: customer name, company, phone, e-mail and other. Also geographical information is very important to us: one of functions of our exercise system is to generate sales statistics on the per-city basis. That is why we have to make effective queries to the database which contains conditions on cities. Note that we also must unambiguously identify cities. This leads to using to a customer city identifier instead of city name. This means that it is neccessary to store full information about cities separately.
- and of course, ordering information must be stored (i.e. an ordered car from list of available, a buyer from our list of customers, a payment amount, a payment type, quantity and so on).

So we should create and design the following tables:

- Cars for storing information about cars and theirs characteristics
- Customers to store customers information
- Cities to store information about cities where customers located
- Orders to store ordering information.

Now we'll examine the process of working with a table. And we'll do it on the **Cars** table example.

[<< Back to Table of Contents]

Stage 2. Creating a diagram

1. Launch the application using the icon in the '**Database Designer for MySQL**' program group in the start menu.

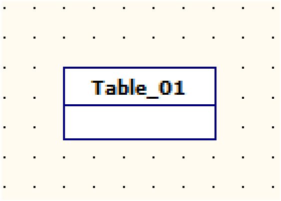
2. Select the **File | New** item on the main program menu to create a new database diagram.

3. You may save your diagram at any time by selecting **File | Save**.

<< Back to Table of Contents

Stage 3. Creating a table

To create a new table in a diagram, click on ▤ icon on the **Palette** toolbar. Your mouse cursor will change its appearance. Click on the diagram area to create a new table. An empty table will appear in the diagram:



Now you are ready to begin defining the fields and parameters for the **Cars** table.

<< Back to Table of Contents

Stage 4. Setting table parameters

To start designing the table double-click on its diagram representation. The program shows you the Table Properties Editor you see in the following picture.

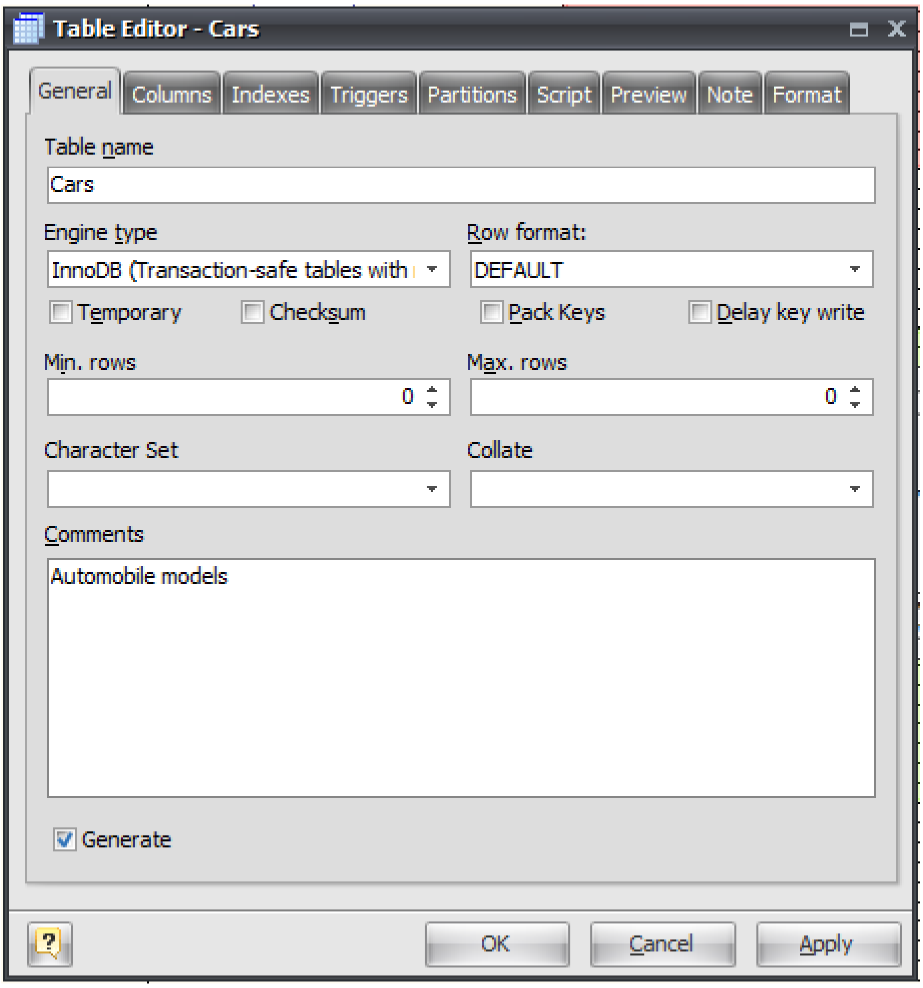


Table name: Cars

Engine type: InnoDB (Transaction-safe tables with

Row format: DEFAULT

□ Temporary □ Checksum □ Pack Keys □ Delay key write

Min. rows: 0

Max. rows: 0

Character Set:

Collate:

Comments: Automobile models

☑ Generate

As you can see, you can set general table parameters such as a table name, type and other.

Specify in this window name of the table - "**Cars**" and change the Engine type to **InnoDB**. The last is required because we need to use a table primary key in foreign keys of other tables, as you will see later.

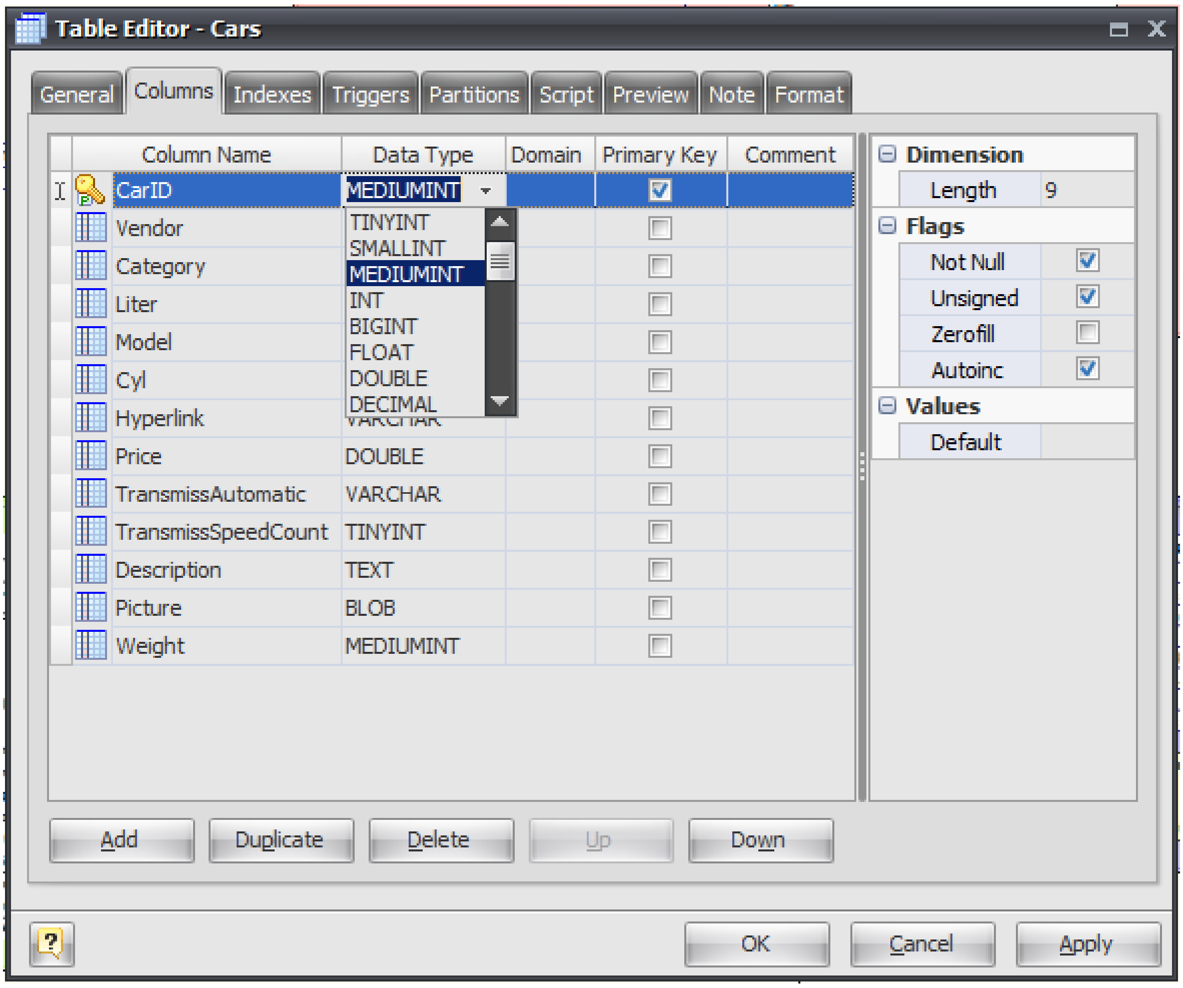
> **Please note:**
> You can read about InnoDB engine type in MySQL Manual: http://dev.mysql.com/doc/refman/5.1/en/innodb.html

<< Back to Table of Contents

Stage 5. Defining table columns

On this step we will create columns of the **Cars** table. As we saw in Stage 2 of this tutorial, we need to store general information about cars in this table.

Click on **Columns** tab of the **Table Properties Editor** that we've launched in the previous stage. The program displays the following window:



Column Name	Data Type	Domain	Primary Key	Comment
CarID	MEDIUMINT		☑	
Vendor			☐	
Category			☐	
Liter			☐	
Model			☐	
Cyl			☐	
Hyperlink	VARCHAR		☐	
Price	DOUBLE		☐	
TransmissAutomatic	VARCHAR		☐	
TransmissSpeedCount	TINYINT		☐	
Description	TEXT		☐	
Picture	BLOB		☐	
Weight	MEDIUMINT		☐	

Dimension	
Length	9
Flags	
Not Null	☑
Unsigned	☑
Zerofill	☐
Autoinc	☑
Values	
Default	

Now we're ready to begin defining fields for the **Cars** table.

First we'll create **CarID** column that will be used as the identifier of a car and as the primary key of **Cars** table.

Click the **Add** button to insert a new column to the table.

Be sure the insertion point is in the first row of the **Column Name** column, and then type in the name of the first field, **CarID**. Select next cell to use **Data Type** column. A button with a down arrow appears on the right side of the **Data Type** column. Here and elsewhere in **Database Designer for MySQL**, this type of button signifies the presence of a drop-down list. Click the down arrow or press Alt+Down arrow to open the list of data. In the **Data Type** column, you can either type in a valid value or select one from the list of values in the drop-down list. Select the appropriate data type for the column. In our case it is **INT**. In the **Comments** column you can enter a descriptive phrase for the column. Comments will help you to remember the destination of columns later and they will be displayed in reports.

Now we must set other column properties.

Database Designer for MySQL displays some property boxes in the field properties area in the left part of the **Table Properties** editor dialog. These boxes allow you to set properties - settings that determine how MySQL handles the column - and thereby customize a column. The properties displayed depend on the data type you selected; the properties appear with some default values in a place.

Since **CarID** is a surrogate primary key for the **Car** table, we must set **AUTO_INCREMENT** and **UNSIGNED** (to eliminate wasting of table space) properties for that. Click on the **Autoinc** and **Unsigned** checkboxes respectively.

Get back again into the column grid. Press **Ctrl + Down** or use once again the **Add** button - this creates a new table column. Enter **Vendor** as a new column name and then select **VARCHAR** as a data type. Set a property **Length** to 50 at the right dialog pane. You have created a new column and set parameters for it.

Then add several columns (and set parameters) for storing other car characteristics:

Category: VARCHAR. Length 10.

Model: VARCHAR. Length 30.

Liter: DOUBLE. Length 15. Decimals 3.

Cyl: TINYINT.

Hyperlink: VARCHAR. Length 255.

Price: DOUBLE. Length 15. Decimals 3.

TransmissAutomatic: VARCHAR. Length 5.

TransmissSpeedCount: TINYINT.

Description: TEXT.

Picture: BLOB.

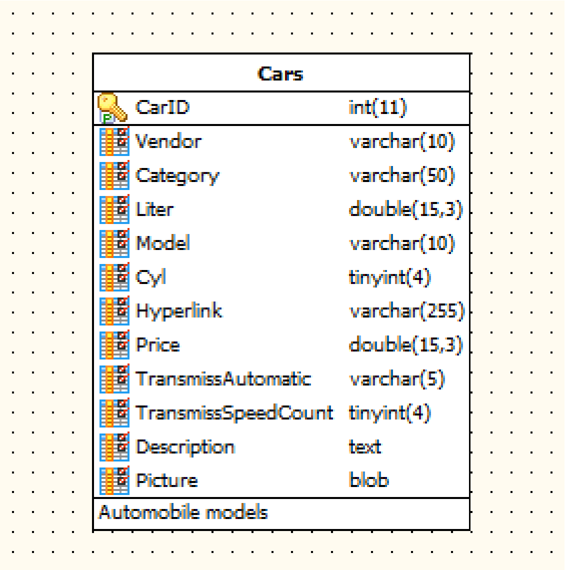
You can add any columns you prefer for describing cars.

After creating columns, click the **Apply** button and then select the **Preview** tab of the **Table Properties editor**. **Database Designer for MySQL** will show you the SQL script representing your table on current stage. You will see the following script:

```
DROP TABLE IF EXISTS `Cars`;

CREATE TABLE `Cars` (
  `CarID` int(11) UNSIGNED NOT NULL AUTO_INCREMENT,
  `Vendor` varchar(10),
  `Category` varchar(50),
  `Liter` double(15,3),
  `Model` varchar(10),
  `Cyl` tinyint(4) UNSIGNED DEFAULT '0',
  `Hyperlink` varchar(255),
  `Price` double(15,3),
  `TransmissAutomatic` varchar(5),
  `TransmissSpeedCount` tinyint(4) UNSIGNED DEFAULT '0',
  `Description` text,
  `Picture` blob,
  PRIMARY KEY(`CarID`)
)
ENGINE=INNODB
COMMENT = 'Automobile models';
```

In the diagram you can see following graphical representation of the table:



Cars	
CarID	int(11)
Vendor	varchar(10)
Category	varchar(50)
Liter	double(15,3)
Model	varchar(10)
Cyl	tinyint(4)
Hyperlink	varchar(255)
Price	double(15,3)
TransmissAutomatic	varchar(5)
TransmissSpeedCount	tinyint(4)
Description	text
Picture	blob
Automobile models	

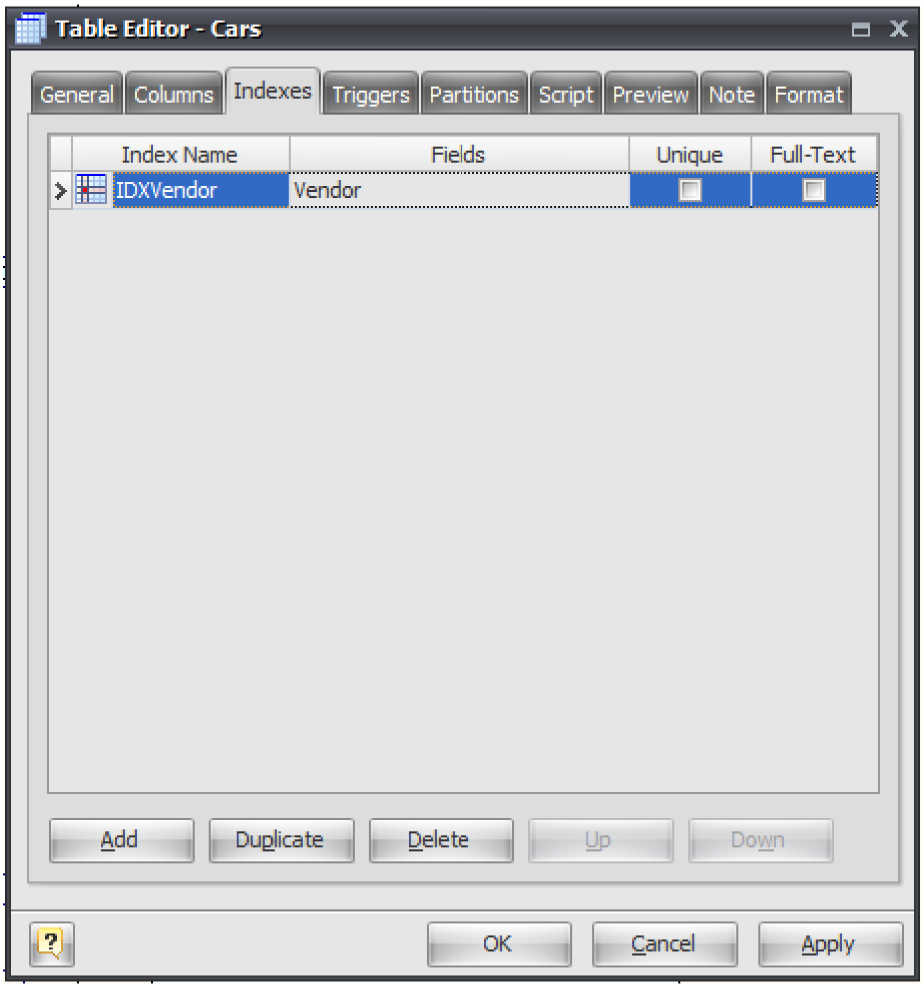
We have completed designing the table columns.

[<< Back to Table of Contents]

Stage 6. Creating indexes

Since we have to create queries to the **Cars** table with several conditions, it is important to optimize the speed of their execution. So we need to add indexes to the table. Determine columns what will be included into the conditions. Suppose it will be the **Vendor** and **Category** columns.

Click on the **Indexes** tab of the **Table Properties editor**. This dialog shows in the following picture:



Index Name	Fields	Unique	Full-Text
IDXVendor	Vendor	□	□

Let's create an index for the **Vendor** column. Click **Add** button to create a new table index.

Be sure the insertion point is in the first row of the **Index Name** column, and then type the index name, **IDXVendor**. *IDX* is the prefix intended to distinguish column names from index names. Click next cell to select fields that will be a part of index. Drop down menu with a list of table columns will appear. Click on the checkbox near the Vendor field. Please note, that you can create a multi-column index if your SELECT-queries conditions contain more than one column, but now we use only one column for simplicity. Then click the **OK** button on the drop-down menu to confirm your selection. You have created the index.

Do the same for the second index for the **Category** field.

After that, check the **Preview** tab of the **Table Properties editor**:

```
DROP TABLE IF EXISTS `Cars`;

CREATE TABLE `Cars` (
  `CarID` int(11) UNSIGNED NOT NULL AUTO_INCREMENT,
  `Vendor` varchar(10),
  `Category` varchar(50),
  `Liter` double(15,3),
  `Model` varchar(10),
  `Cyl` tinyint(4) UNSIGNED DEFAULT '0',
  `Hyperlink` varchar(255),
  `Price` double(15,3),
  `TransmissAutomatic` varchar(5),
  `TransmissSpeedCount` tinyint(4) UNSIGNED DEFAULT '0',
  `Description` text,
  `Picture` blob,
  PRIMARY KEY(`CarID`),
  INDEX `IDXVendor`(`Vendor`),
  INDEX `IDXCategory`(`Category`)
)
ENGINE=INNODB
COMMENT = 'Automobile models';
```

Pay attention to the last lines of the table body formatted bold.

> **Please note:**
> You don't have to create an index for the primary key column - **Database Designer for MySQL** already did it for you.
> You can adjust your table colors at Format tab of the Table Properties editor or with Color Palette

<< Back to Table of Contents

Stage 7. Creating and designing the rest of tables

Now we shall create the rest of the tables of our system database. They are: **Customers**, **Cities**, **Orders**.

> **Please note:**
> You can skip this laborious section and use the ready-to-use diagram **tutorial-first-diagram.mdd** bundled with the product in the **Samples** directory.

The process of table creating, defining columns and indexes is fully described in Stages 3 - 6 of this tutorial.

Please, create the following tables:

Table *Customers*

Columns:

CustID: MEDIUMINT UNSIGNED, AUTO_INCREMENT, PRIMARY KEY

Address: VARCHAR

Company: VARCHAR

Customer: VARCHAR

Description: TEXT

Email: VARCHAR

FaxPhone: VARCHAR

FirstName: VARCHAR

HomePhone: VARCHAR

LastName: VARCHAR

Occupation: VARCHAR

Prefix: VARCHAR

Source: VARCHAR

Spouse: VARCHAR

State: VARCHAR

Title: VARCHAR

ZipCode: VARCHAR.

Indexes:
for column **Email**.

Table *Cities*

Columns:
CityID: MEDIUMINT UNSIGNED, AUTO_INCREMENT, PRIMARY KEY

CityName: VARCHAR.

Table *Order*

Columns:
OrderID: MEDIUM INT UNSIGNED, AUTO_INCREMENT, PRIMARY KEY

Description: TEXT

PaymentAmount: DOUBLE

PaymentType: VARCHAR

PurchaseDate: DATE

Quantity: TINYINT

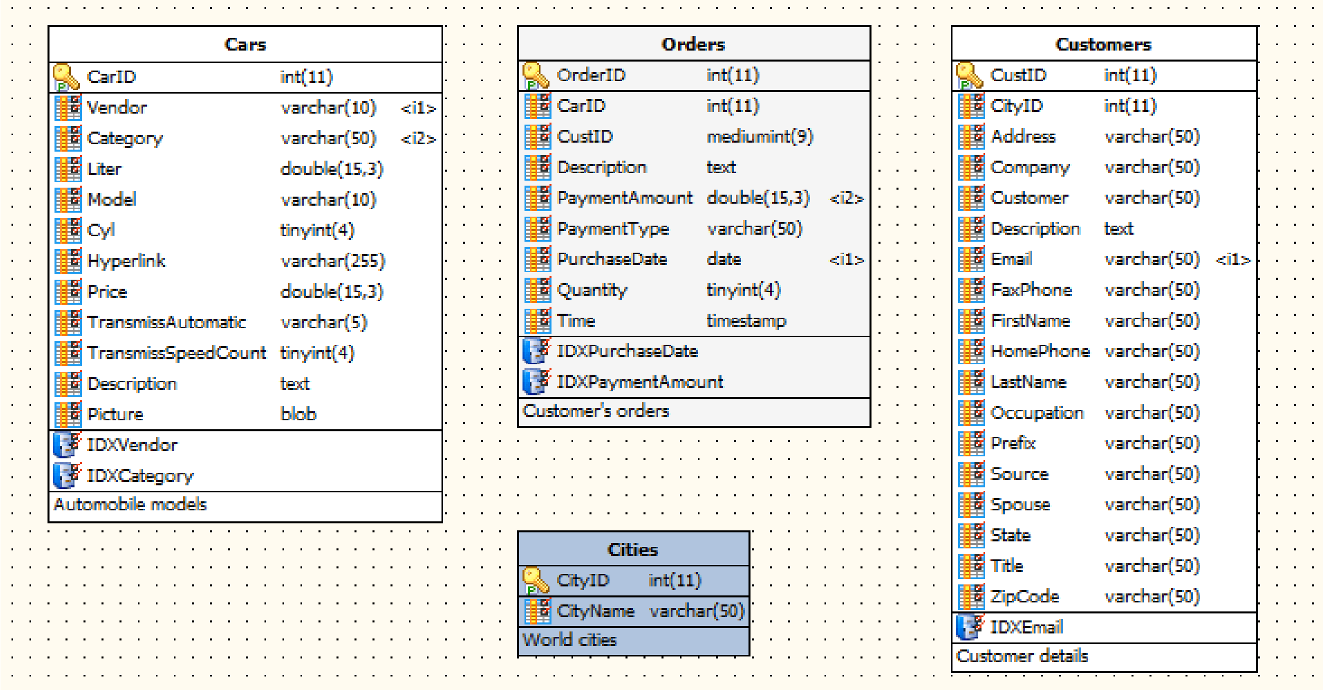
Time: TIMESTAMP.

Indexes:

for column **PaymentAmount**

for column **PurchaseDate**.

When you finish you should get the following diagram:



Cars		
CarID	int(11)	
Vendor	varchar(10)	<i1>
Category	varchar(50)	<i2>
Liter	double(15,3)	
Model	varchar(10)	
Cyl	tinyint(4)	
Hyperlink	varchar(255)	
Price	double(15,3)	
TransmissAutomatic	varchar(5)	
TransmissSpeedCount	tinyint(4)	
Description	text	
Picture	blob	
IDXVendor		
IDXCategory		
Automobile models		

Orders		
OrderID	int(11)	
CarID	int(11)	
CustID	mediumint(9)	
Description	text	
PaymentAmount	double(15,3)	<i2>
PaymentType	varchar(50)	
PurchaseDate	date	<i1>
Quantity	tinyint(4)	
Time	timestamp	
IDXPurchaseDate		
IDXPaymentAmount		
Customer's orders		

Cities	
CityID	int(11)
CityName	varchar(50)
World cities	

Customers		
CustID	int(11)	
CityID	int(11)	
Address	varchar(50)	
Company	varchar(50)	
Customer	varchar(50)	
Description	text	
Email	varchar(50)	<i1>
FaxPhone	varchar(50)	
FirstName	varchar(50)	
HomePhone	varchar(50)	
LastName	varchar(50)	
Occupation	varchar(50)	
Prefix	varchar(50)	
Source	varchar(50)	
Spouse	varchar(50)	
State	varchar(50)	
Title	varchar(50)	
ZipCode	varchar(50)	
IDXEmail		
Customer details		

[<< Back to Table of Contents]

Stage 8. Creating foreign keys

You have probably noticed that at the previous stage we didn't store customer and car information in the order table. And we didn't link customers with cities.

> **Please note:**
> You may skip the current stage and fill this the gap in our diagram by adding the column **CityID** to Customers table, and **CarID** and **CustID** columns to table **Orders**. But if you plan to use the MySQL foreign keys we recommend you to examine the current stage.

Database Designer for MySQL supports MySQL foreign keys. Foreign keys are responsible for data referential integrity in your database. Simply put, referential integrity means that when a record in a table refers to a corresponding record in another table, that corresponding record must exist.

As we know, the **Customers** table must be linked with the **Cities** table by the **CityID** column. When a new row is inserted in the **Customers** table, it must have a corresponding record in the **Cities** table. If you define the **CityID** column of the **Customers** table you enforce MySQL to produce such checks.

You can define a foreign key in any MySQL table type, including the default **MyISAM** table type, but they do not do anything - they are only used to enforce referential integrity in **InnoDB** or **Falcon** tables.

In order to create a foreign key in MySQL, you need the following:

- Both tables are to be **InnoDB** (or **Falcon**) tables
- The following syntax should be used: FORIGN_KEY (fk_fieldname) REFERENCES table_name (fieldname)
- The field being declared a foreign key requires to be declared as an index in the table definition.

> **Please note:**
> Since MySQL v4.1.2 indexes for foreign key columns are created automatically. So **Automatically create indexes for FK-columns** option in Diagram Properties dialog is disabled by default.

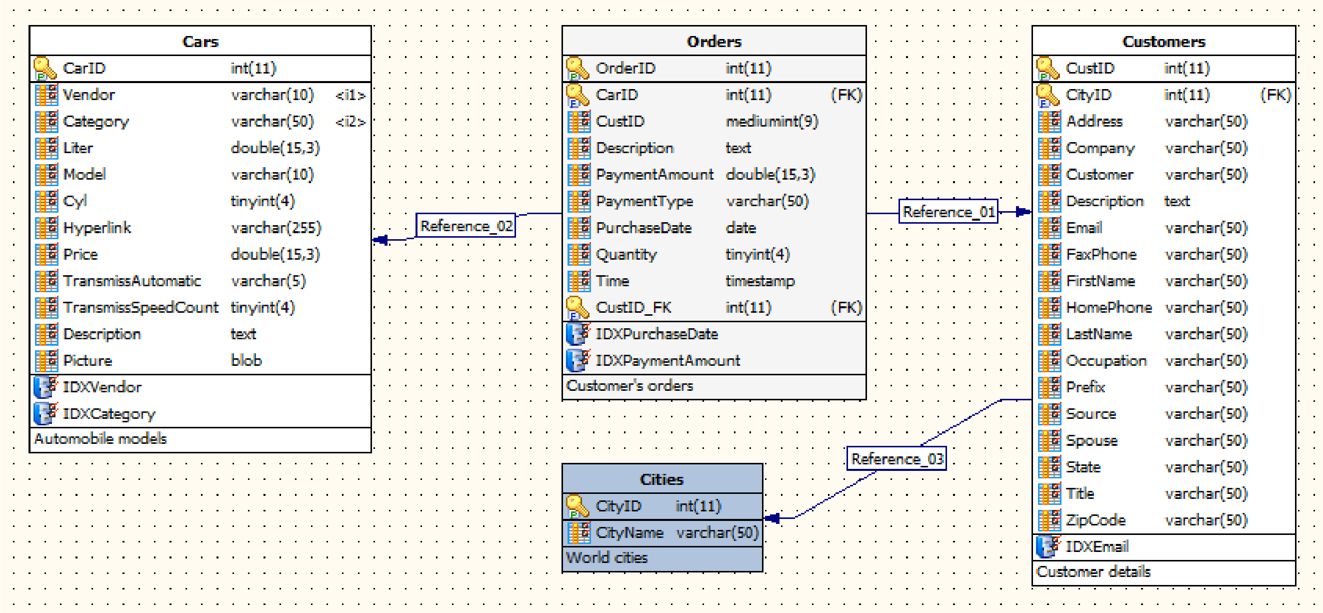
Database Designer for MySQL will help you to do it in one action by creating a reference between two tables.

To create a reference between tables, click on **Reference** () item on the **Palette** tab of Rebbon. Your mouse cursor will change its appearance. Then click on the table (child table) that will have foreign key and then click on the second table (parent table) whose primary key will be referred by the new foreign key.

Do it with the **Customers** and **Cities** tables respectively. Then with the **Orders** and **Customers** tables. And at the end with **Orders** and **Cars**.

Database Designer for MySQL has automatically created **CityID** column in **Customers** table, the same as the **CityID** in the **Cities** table and set the foreign key.

After creating all above references you will get the following diagram:



Cars		
CarID	int(11)	
Vendor	varchar(10)	<i1>
Category	varchar(50)	<i2>
Liter	double(15,3)	
Model	varchar(10)	
Cyl	tinyint(4)	
Hyperlink	varchar(255)	
Price	double(15,3)	
TransmissAutomatic	varchar(5)	
TransmissSpeedCount	tinyint(4)	
Description	text	
Picture	blob	
IDXVendor		
IDXCategory		
Automobile models		

Orders		
OrderID	int(11)	
CarID	int(11)	(FK)
CustID	mediumint(9)	
Description	text	
PaymentAmount	double(15,3)	
PaymentType	varchar(50)	
PurchaseDate	date	
Quantity	tinyint(4)	
Time	timestamp	
CustID_FK	int(11)	(FK)
IDXPurchaseDate		
IDXPaymentAmount		
Customer's orders		

Customers		
CustID	int(11)	
CityID	int(11)	(FK)
Address	varchar(50)	
Company	varchar(50)	
Customer	varchar(50)	
Description	text	
Email	varchar(50)	
FaxPhone	varchar(50)	
FirstName	varchar(50)	
HomePhone	varchar(50)	
LastName	varchar(50)	
Occupation	varchar(50)	
Prefix	varchar(50)	
Source	varchar(50)	
Spouse	varchar(50)	
State	varchar(50)	
Title	varchar(50)	
ZipCode	varchar(50)	
IDXEmail		
Customer details		

Cities	
CityID	int(11)
CityName	varchar(50)
World cities	

Now let's examine the **Preview** tab of the **Customers** table:

```
DROP TABLE IF EXISTS `Customers`;

CREATE TABLE `Customers` (
  `CustID` int(11) UNSIGNED NOT NULL AUTO_INCREMENT,
  `CityID` int(11) UNSIGNED NOT NULL DEFAULT '0',
  `Address` varchar(50),
  `Company` varchar(50),
  `Customer` varchar(50),
  `Description` text,
  `Email` varchar(50),
  `FaxPhone` varchar(50),
  `FirstName` varchar(50),
  `HomePhone` varchar(50),
  `LastName` varchar(50),
  `Occupation` varchar(50),
  `Prefix` varchar(50),
  `Source` varchar(50),
  `Spouse` varchar(50),
  `State` varchar(50),
  `Title` varchar(50),
  `ZipCode` varchar(50),
  PRIMARY KEY(`CustID`),
  INDEX `IDXEmail`(`Email`),
  CONSTRAINT `Reference_03` FOREIGN KEY (`CityID`)
    REFERENCES `Cities`(`CityID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
)
ENGINE=INNODB
COMMENT = 'Customer details';
```

You have created references between tables of our diagram and corresponding foreign keys.

Please note:
By default **Database Designer for MySQL** draws references automatically using shortest line

> between table shapes on diagram with reference's label placed in the center of this line. Sometimes this can lead to references crossing, or reference's label can fit below table shape. Starting from version 2.0 you can route references manually to avoid this. Take a look at Manual Reference Drawing topic.

<< Back to Table of Contents

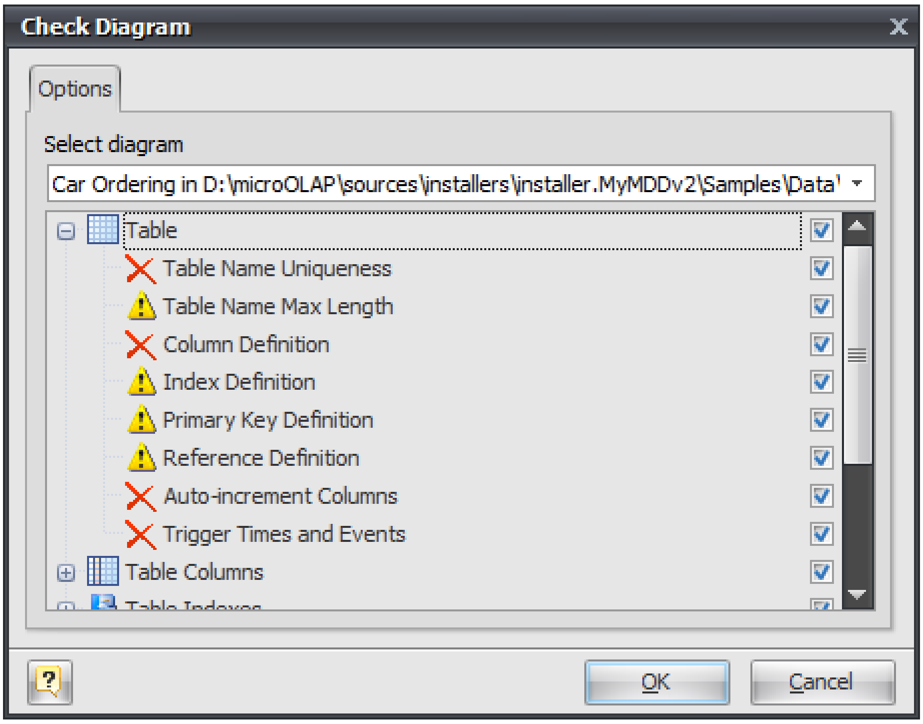
Stage 9. Checking the diagram

At this stage you will know how to find errors which usually occur during the process of designing a diagram.

Database Designer for MySQL can verify your diagram for consistency and find typical errors in database design and maintaining processes.

Select the **Diagram | Check Diagram** menu item or press **F4** to run the Check Diagram tool. The **Check Diagram** tool will verify your diagram for most frequent errors, such as table and column naming uniqueness, indexes availability and so on.

The **Check Diagram** dialog allows you to select the types of the errors you want to verify:



Check Diagram

Options

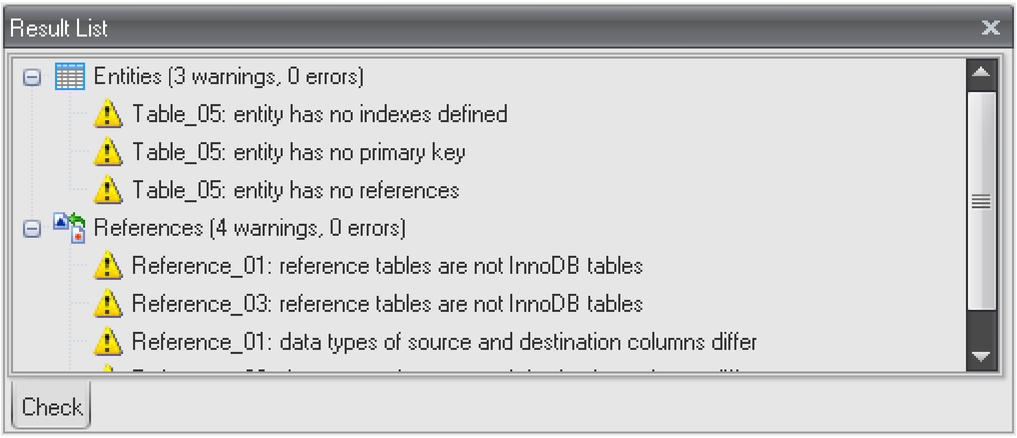
Select diagram

Car Ordering in D:\microOLAP\sources\installers\installer.MyMDDv2\Samples\Data'

- ☑ Table
  - ☑ Table Name Uniqueness
  - ☑ Table Name Max Length
  - ☑ Column Definition
  - ☑ Index Definition
  - ☑ Primary Key Definition
  - ☑ Reference Definition
  - ☑ Auto-increment Columns
  - ☑ Trigger Times and Events
- ☑ Table Columns
- ☑ Table Indexes

Click the **OK** button to execute the selected checks.

You can see the errors and warnings that were found in the diagram by the **Check Diagram** tool in the **Result List** docking window:



Entities (3 warnings, 0 errors)
- Table_05: entity has no indexes defined
- Table_05: entity has no primary key
- Table_05: entity has no references

References (4 warnings, 0 errors)
- Reference_01: reference tables are not InnoDB tables
- Reference_03: reference tables are not InnoDB tables
- Reference_01: data types of source and destination columns differ

This information will help you to correct errors, i.e. rename tables and columns, add appropriate indexes to tables and so on.

[<< Back to Table of Contents](#)

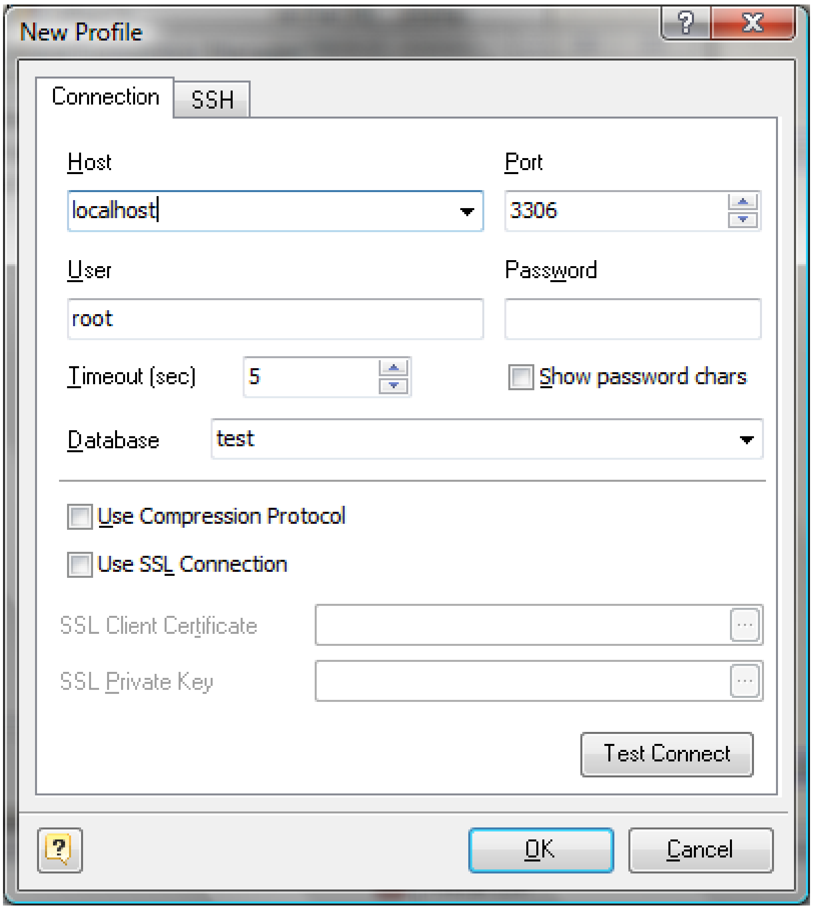
Stage 10. Generating the database

This stage will show you how to generate database in working MySQL server or save the SQL script that corresponds to your diagram.

When you finish creating the diagram, you can generate the database on MySQL server.

First, you need to establish a connection to your database. Select **Database | Connect** to start the Database Connection Manager. Press **Add** button to create new Connection Profile.

You will get the following dialog:



Host
localhost

Port
3306

User
root

Password

Timeout (sec) 5

□ Show password chars

Database test

□ Use Compression Protocol

□ Use SSL Connection

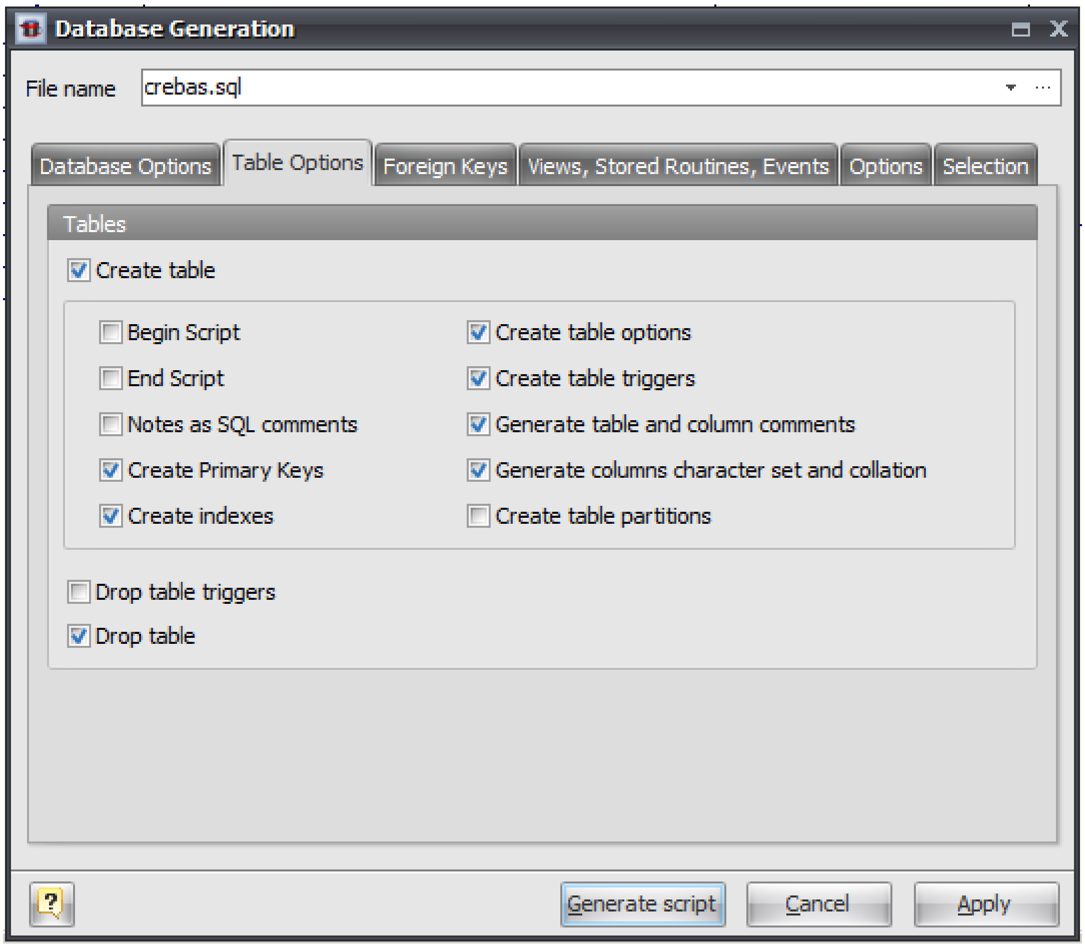
SSL Client Certificate

SSL Private Key

Enter the connection parameters: **Host**, **Port**, **User**, **Password**, and select a **Database** from the drop-down list. Then click the **OK** button to put the connection parameters to the list of available connections. After that click the **OK** button to establish connection to your database.

Now you are able to generate the database.

Select **Generate Database** ( ) item on **Database** tab of the Ribbon to start the Database Generation tool:



File name: crebas.sql

☑ Create table

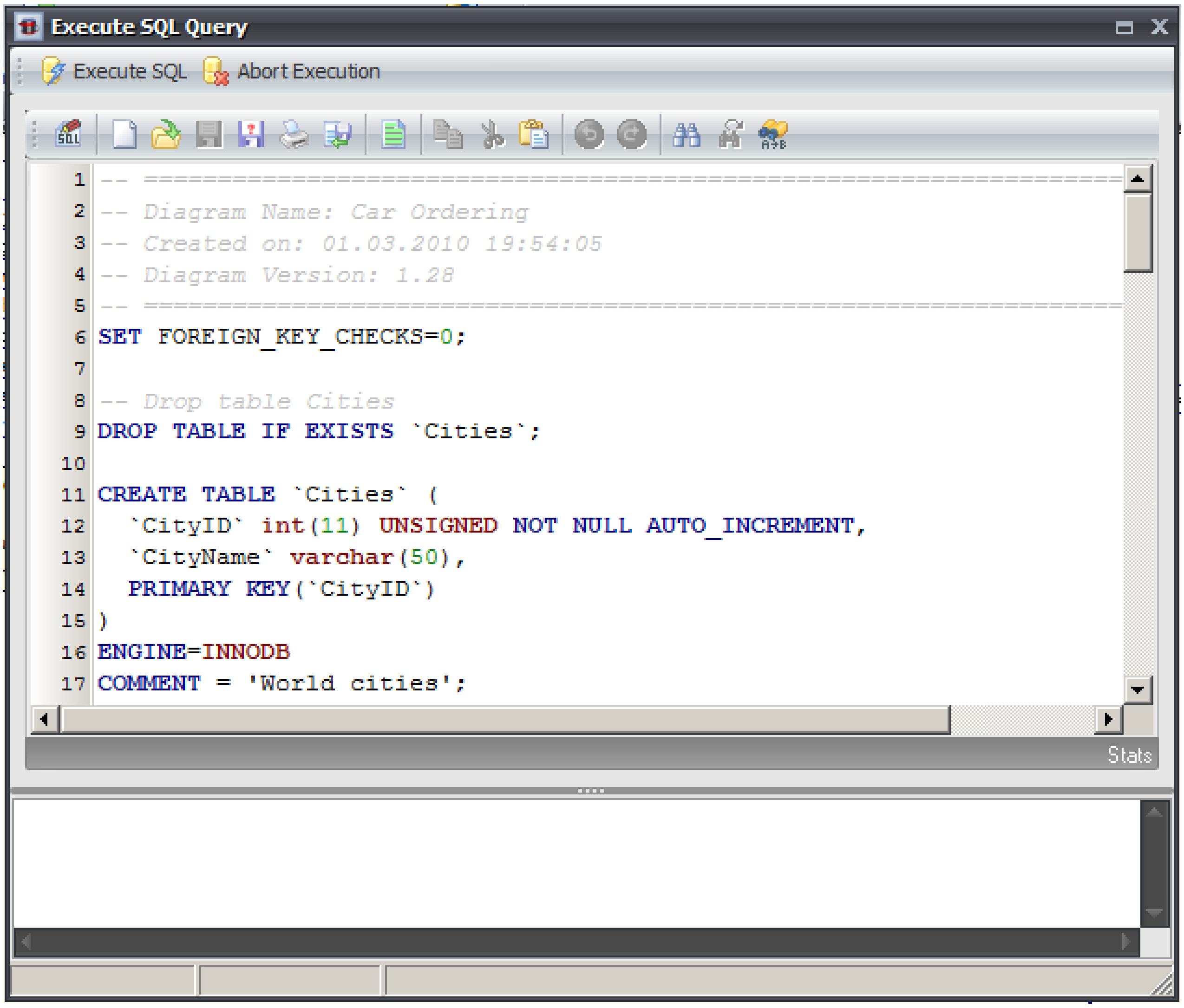
☐ Begin Script
☐ End Script
☐ Notes as SQL comments
☑ Create Primary Keys
☑ Create indexes
☑ Create table options
☑ Create table triggers
☑ Generate table and column comments
☑ Generate columns character set and collation
☐ Create table partitions

☐ Drop table triggers
☑ Drop table

This dialog allows you to setup the database generation options: select whether to generate tables, indexes, columns, enable table comments, etc. In the **Selection** tab of the dialog you can select tables for generating. To select a file for storing the generated script use the **File name** edit box.

Click the **Generate Script** button to generate consecution of SQL statements for creating your database.

After the process is finished, **Database Designer for MySQL** will display the generated SQL statements within the SQL Executor window. You can edit them according to your needs.



```
-- ============================================================
-- Diagram Name: Car Ordering
-- Created on: 01.03.2010 19:54:05
-- Diagram Version: 1.28
-- ============================================================
SET FOREIGN_KEY_CHECKS=0;

-- Drop table Cities
DROP TABLE IF EXISTS `Cities`;

CREATE TABLE `Cities` (
  `CityID` int(11) UNSIGNED NOT NULL AUTO_INCREMENT,
  `CityName` varchar(50),
  PRIMARY KEY(`CityID`)
)
ENGINE=INNODB
COMMENT = 'World cities';
```

To run the statements in the database click the **Execute SQL** button.

The process of statements executing will be shown in the bottom of the dialog.

You have generated your database from designed diagram.

<< Back to Table of Contents

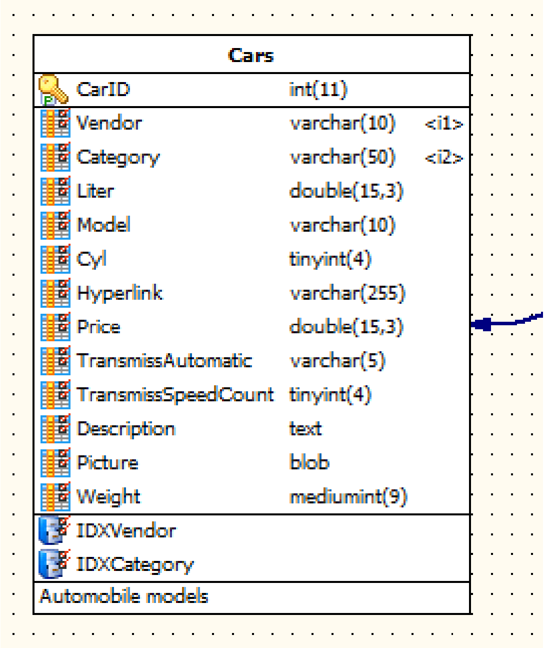
Stage 11. Altering the diagram

Let's start learning about one of the most powerful **Database Designer for MySQL** features - synchronizing the database with your diagram. Below we will make some changes to our diagram and then synchronize it with the database.

Let's assume, that we have to store a new car characteristic, e.g. weight. So, we need to add the **Weight** column to the **Cars** table.

Double-click on the **Cars** table to call the **Table Properties** editor. Then click the **Add** button to create a new column. Enter the name of the new column - **Weight** and attach MEDIUMINT data type to it. Click the **OK** to save our modifications.

You will get about this table on the diagram:



Cars		
CarID	int(11)	
Vendor	varchar(10)	<i1>
Category	varchar(50)	<i2>
Liter	double(15,3)	
Model	varchar(10)	
Cyl	tinyint(4)	
Hyperlink	varchar(255)	
Price	double(15,3)	
TransmissAutomatic	varchar(5)	
TransmissSpeedCount	tinyint(4)	
Description	text	
Picture	blob	
Weight	mediumint(9)	
IDXVendor		
IDXCategory		
Automobile models		

Now you can apply that changes to the database.

[<< Back to Table of Contents]

Stage 12. Synchronizing database with the diagram

At this stage you will learn how to apply changes in your diagram to the database.

> **Please note:**
> Database modification usually causes multiple complex statements for database structure modification. It is possible that some of them may execute incorrectly due to the database hight load. It's recommended to make a backup of your database before applying structure changes to it.

Click on the **Database | Modify Database** menu item to start the Database Modification tool.

In the **Options** tab you can set modification parameters: delete existing tables in the database, new table creation options, etc. The **Selection** tab allows you to select tables you want to modify.

Click **OK** to generate the database modification script.

Generated script will be shown in the the SQL Executor window and consist of required modification statements to bring the database to diagram state. You can edit the generated statements according to your special needs. After you've finished click on the **Execute SQL** button to apply modifications to the database.

As you can guess, it is hard work to create these modification statements manually: alter tables are not always easy and handy, often a simple change of a database structure may cause multiple SQL modification statements.

[<< Back to Table of Contents]

Afterword

We have examined most of the main features and tools of **Database Designer for MySQL**. Using **Database Designer for MySQL** you can perform many complex and frequent task for database design and maintenance in easy-to-use graphical environment. We wish you successful and productive work with **Database Designer for MySQL**!

<< Back to Table of Contents

See also:
Tutorial: Using Domains

4.2. Using Domains

This tutorial helps you to understand and start using one of the powerful features of the product -- domains.

Table of Contents

Stage 1. Examine tables and columns

Stage 2. Creating domains

Stage 3. Attaching domains to columns

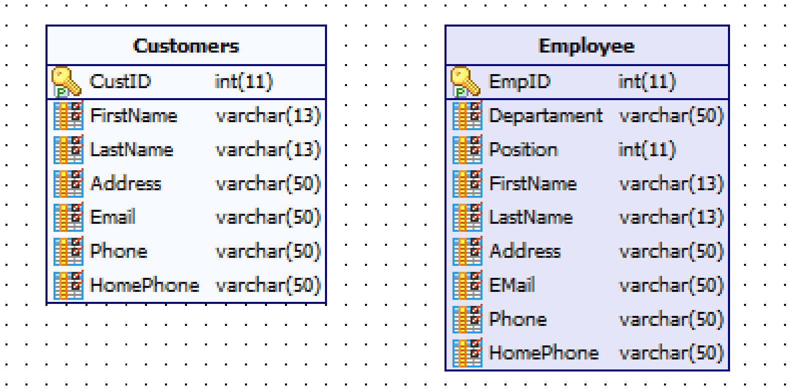
Stage 4. Modifying domains

Afterword

> **Please note:**
> The database diagram, which we develop in this section, is placed in the **tutorial-second-diagram.mdd** in the **Samples** directory in the program installation path.

Stage 1. Examine tables and columns

Let's look at a sample situation. We have the **Customers** and **Employee** tables:



Customers	
CustID	int(11)
FirstName	varchar(13)
LastName	varchar(13)
Address	varchar(50)
Email	varchar(50)
Phone	varchar(50)
HomePhone	varchar(50)

Employee	
EmpID	int(11)
Departament	varchar(50)
Position	int(11)
FirstName	varchar(13)
LastName	varchar(13)
Address	varchar(50)
EMail	varchar(50)
Phone	varchar(50)
HomePhone	varchar(50)

As you can note they have a few similar columns - for storing person names, e-mails, phones and so on.

Similar columns have the same data types because they are intended for storing similar data.

What will you do if you recognize that 13 chars are not enough for storing last and first person names in your system? And what if some of e-mail addresses do not fit into 20-chars field? Then you will have to change the incorrect column data types for each table.

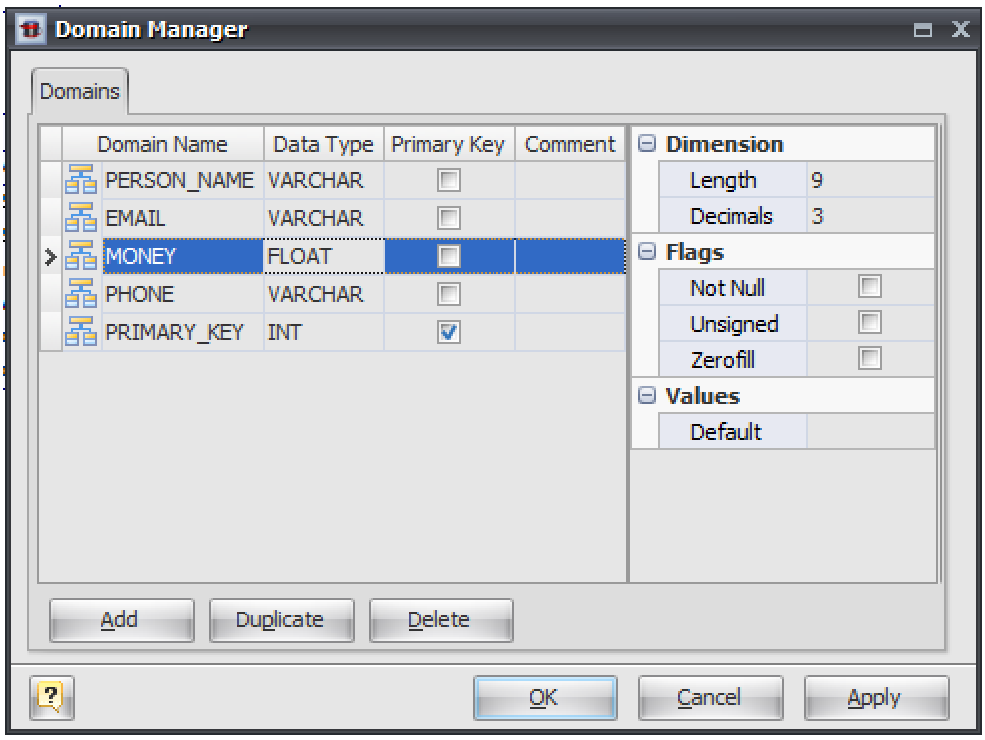
On the next stages we will show how to simplify this task and how to create columns with predefined parameters quickly.

[<< Back to Table of Contents]

Stage 2. Creating domains

The domains are user-defined types, which contain all column information (data type, key options, dimensions, and flags). They allow you to add and modify existing domain-based columns quickly. Currently domains are not supported by MySQL, so it is a **Database Designer for MySQL** extension to the server.

To create and manage domains select the **Diagram | Domain Manager...** menu item. You will see the Domain Manager dialog:



Domain Name	Data Type	Primary Key	Comment
PERSON_NAME	VARCHAR	□	
EMAIL	VARCHAR	□	
MONEY	FLOAT	□	
PHONE	VARCHAR	□	
PRIMARY_KEY	INT	☑	

Dimension	
Length	9
Decimals	3
Flags	
Not Null	□
Unsigned	□
Zerofill	□
Values	
Default	

The interface of this dialog is very similar to the **Columns** tab of the Table Properties editor.

Now you can create domains. Click the **Add** button to create a domain. Be sure the insertion point is in the first row of the **Domain Name** column, and then type in the name of the first field, **PERSON_NAME**. This domain will be used for columns that are intended for storing person names (last and first names) in our database. Select the next cell to use **Data Type** column. A button with a down arrow appears on the right side of the **Data Type** column. Select the appropriate data type - **VARCHAR**. Then set length in the right pane of the dialog to 13.

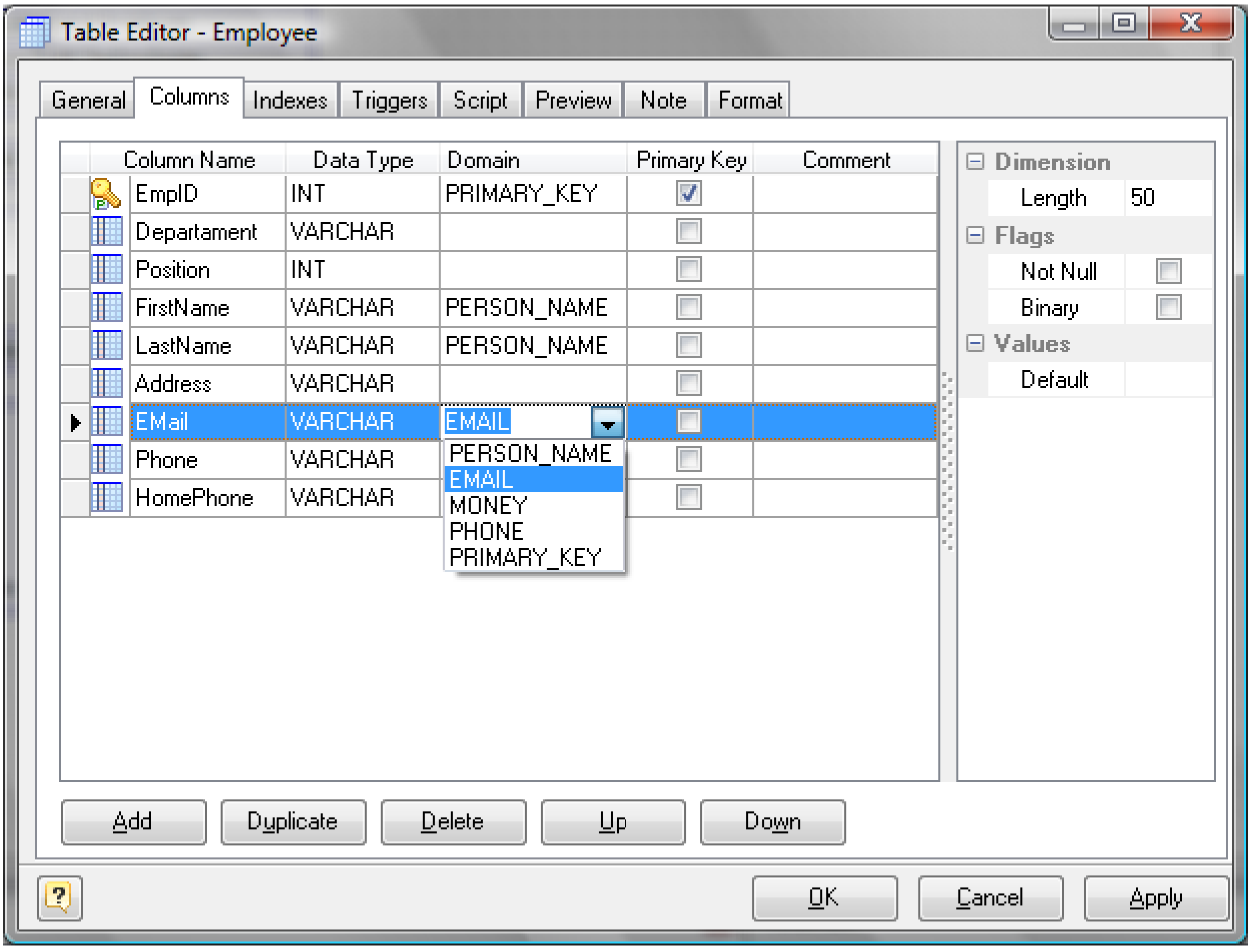
Add other neccessary domains for other types of the information used in your tables: **PRIMARY_KEY**, **EMAIL**, **PHONE**, etc.

<< Back to Table of Contents

Stage 3. Attaching domains to columns

Now we can attach domains to columns.

Double click on the **Employee** table and switch to **Columns** tab. You will see the following dialog:



Column Name	Data Type	Domain	Primary Key	Comment
EmpID	INT	PRIMARY_KEY	☑	
Departament	VARCHAR		□	
Position	INT		□	
FirstName	VARCHAR	PERSON_NAME	□	
LastName	VARCHAR	PERSON_NAME	□	
Address	VARCHAR		□	
EMail	VARCHAR	EMAIL	□	
Phone	VARCHAR		□	
HomePhone	VARCHAR		□	

Dimension
Length 50

Flags
Not Null □
Binary □

Values
Default

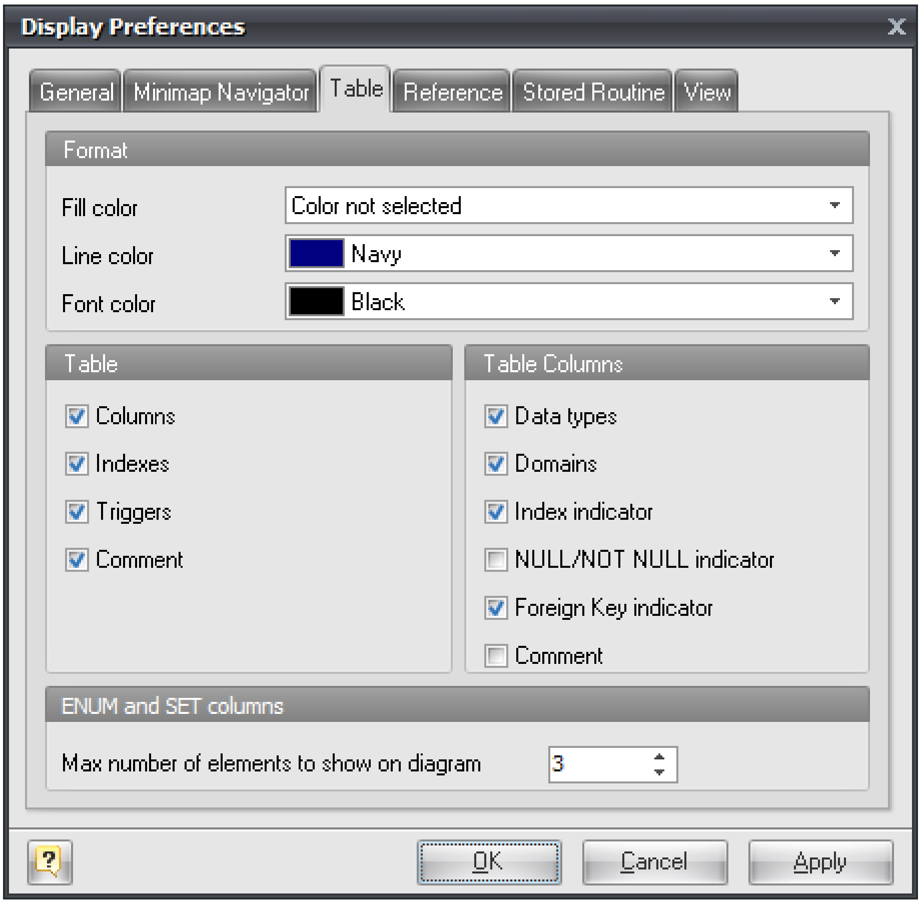
Click on the domain cell near the **FirstName** column to select the appropriate domain from the list of available diagram domains. Choose the **PERSON_NAME** domain. Now the **FirstName** column is domain-based, all changes in the **PERSON_NAME** domain will bring proper changes in the column. It is possible to consider domains as column templates. Please repeat these actions with the rest of the columns:

- attach to the FirstName and LastName columns the PERSON_NAME domain
- attach to the Email column the EMAIL domain
- attach to the EmpID column the PRIMARY_KEY domain
- and so on.

Click the **OK** button to commit your changes.

Do the same for the **Customers** table.

It is quite useful to show column domains within the table boxes on the diagram. Select the **Tools | Display Preferences** menu item to adjust the Diagram Display Preferences. Go to the **Table** tab and click on the **Domains** check box:



Format

Fill color: Color not selected

Line color: ■ Navy

Font color: ■ Black

Table

☑ Columns
☑ Indexes
☑ Triggers
☑ Comment

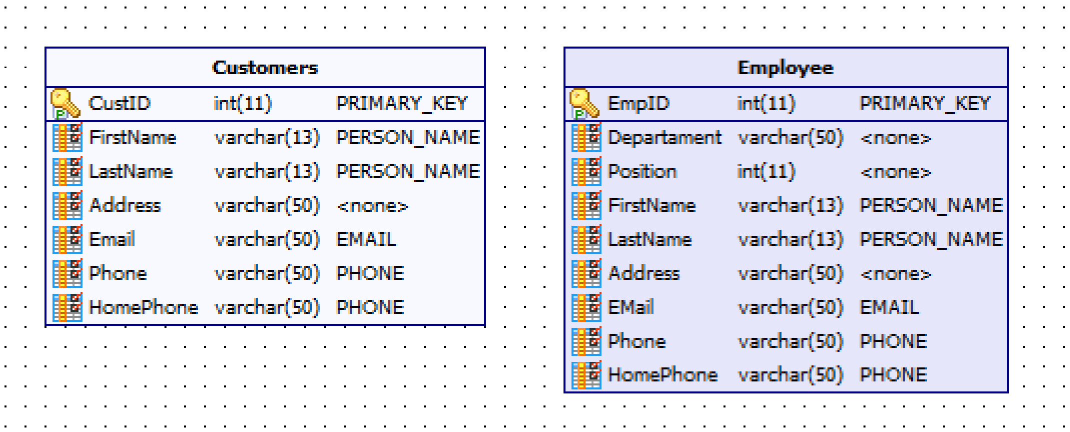
Table Columns

☑ Data types
☑ Domains
☑ Index indicator
☐ NULL/NOT NULL indicator
☑ Foreign Key indicator
☐ Comment

ENUM and SET columns

Max number of elements to show on diagram: 3

After that you will get the following diagram:



Customers		
CustID	int(11)	PRIMARY_KEY
FirstName	varchar(13)	PERSON_NAME
LastName	varchar(13)	PERSON_NAME
Address	varchar(50)	<none>
Email	varchar(50)	EMAIL
Phone	varchar(50)	PHONE
HomePhone	varchar(50)	PHONE

Employee		
EmpID	int(11)	PRIMARY_KEY
Departament	varchar(50)	<none>
Position	int(11)	<none>
FirstName	varchar(13)	PERSON_NAME
LastName	varchar(13)	PERSON_NAME
Address	varchar(50)	<none>
EMail	varchar(50)	EMAIL
Phone	varchar(50)	PHONE
HomePhone	varchar(50)	PHONE

Please note:
You can also attach domain to a column during the column creation.

[<< Back to Table of Contents]

Stage 4. Modifying domains

Suppose, that it is necessary to increase the quantity of chars for storing e-mail addresses.

Again, open the Domain Manager by clicking on the **Diagram | Domain Manager...** menu item. Click on the **EMAIL** domain in domains grid. And then increase the **Length** option (in the right side of dialog) to, say, 60. Click the **OK** button to store your changes.

Then examine your table columns, that are based on the **EMAIL** domain: their **Length** property are changed to the value, that was set for domain.

<< Back to Table of Contents

Afterword

We have learnt what domains are, how to use them in your diagram design and how they speed up your database maintenance process.

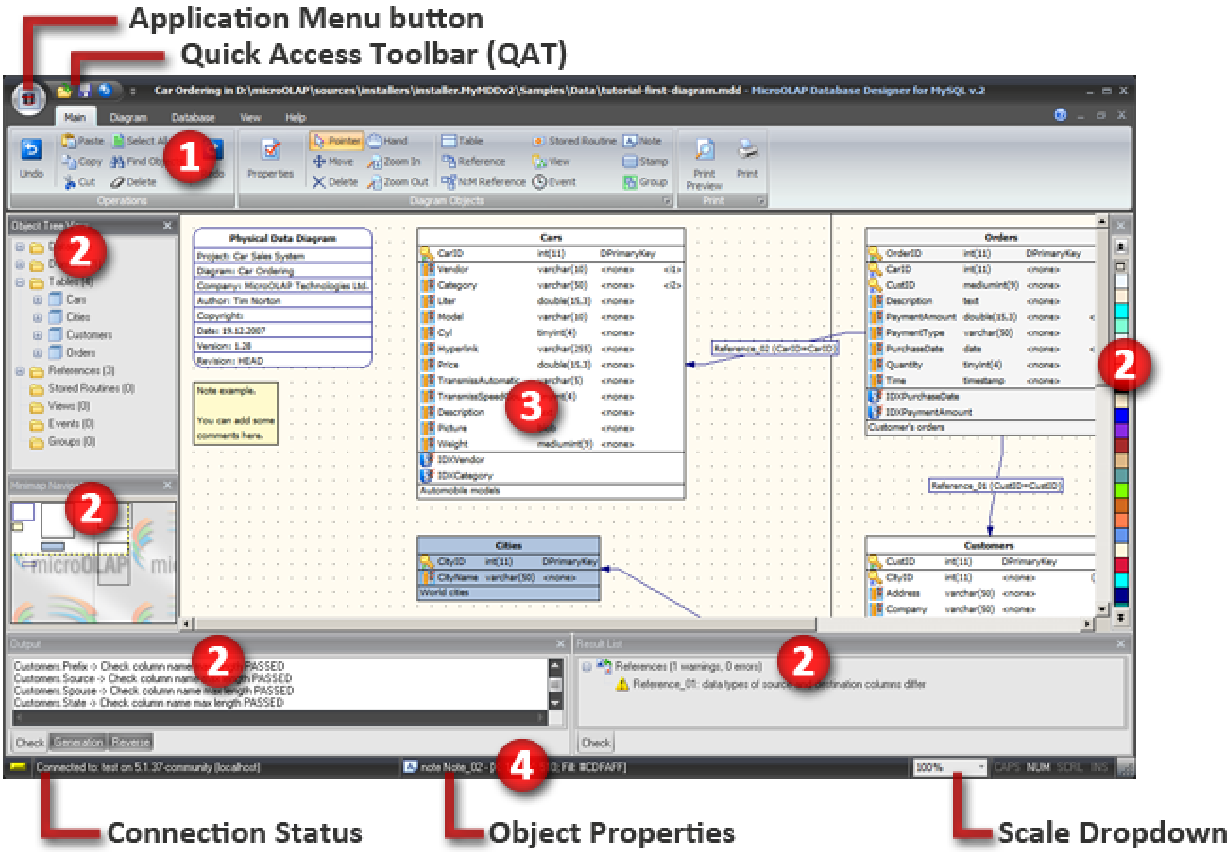
<< Back to Table of Contents

See also:
Tutorial: Creating Your First Diagram

5. Interface User Guide

The **Database Designer for MySQL** user interface has been designed to provide you with convenience and flexibility in performing database designing tasks. Version 2 of **Database Designer for MySQL** introduces Ribbon-based user interface (like that found in MS Office 2007) and different visual themes (Skins).

There are 4 main areas in the **Database Designer for MySQL** user interface. They are illustrated in the following screen shot:



1. Ribbon

All menu bars and toolbars from older versions are now grouped in a single **Ribbon**. **Ribbon** is a set of tabs with buttons, drop-down controls and so on that provide you with access to the **Database Designer for MySQL** functions. The behavior of **Ribbon** can be customized.

2. Docking Windows

Docking windows are portions of the **Database Designer for MySQL** interface containing specific functionality, such as the **Tree View Window**, **Result List**, **Output**, Minimap Navigator and **Color Palette**. Docking windows can be Docked, Floating and Tab-Docked.

3. Diagram Window

This area contains the database diagram, which is currently being edited. You can work with multiple diagrams at the same time.

4. Status bar

The **Status bar** shows you the information about current database connection, properties of the currently active diagram object, current scale and CAPS LOCK / NUMLOCK / SCROLL LOCK keyboard buttons state.

See also:
Diagram Objects: Diagram Overview
Diagram: Notation
Interface User Guide: Diagram Window | Ribbon | Docking Windows | Keyboard Shortcuts | Minimap Navigator | GUI Skins

5.1. Diagram Window

The **diagram window** is the main working area of the application. It displays the active diagram content. You can open multiple diagrams and switch between them using **Window** drop-down control on **View** tab of the Ribbon. Also you can dispose a number of diagrams at the working area. Please, note that more than one diagram can be active.

To change diagrams disposition in the working area automatically, use the following **View** tab of the Ribbon items:

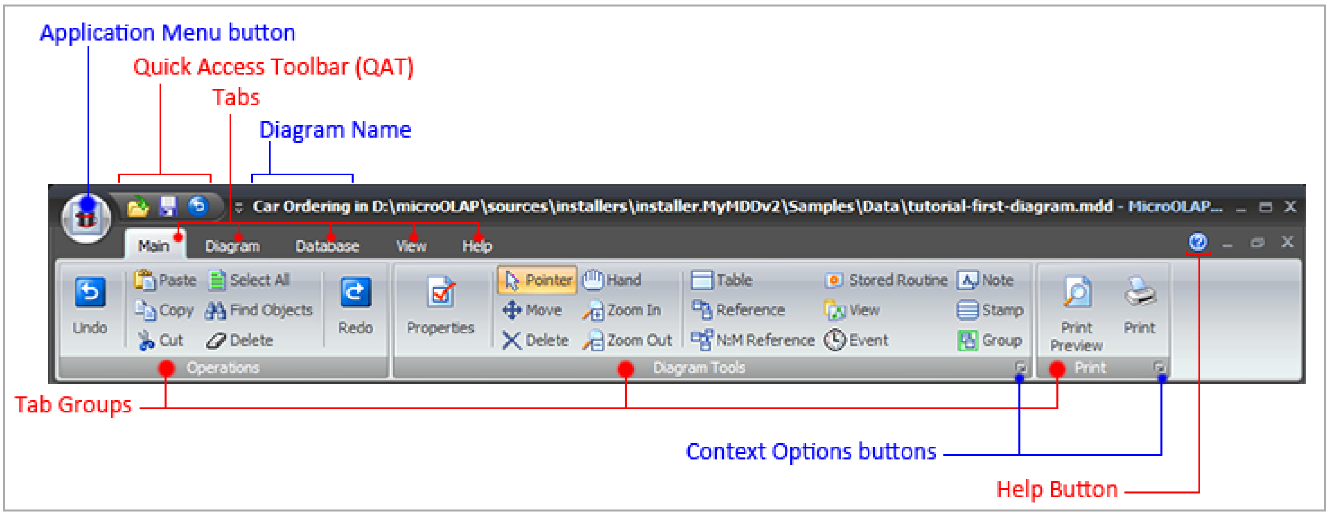
- Cascade
- Tile Horizontal
- Tile Vertical.

To close all opened diagrams, use the **Close All** (☒) item.

See also:
Diagram: Creating a New Diagram | Notation | Export to Graphics
Interface User Guide: Ribbon

5.2. Ribbon

Graphical User's Interface (GUI) in version 2 is based on **Ribbon** (like that found in MS Office 2007). Ribbon is a set of **Tabs** with buttons, drop-down controls and other controls that provide you with access to the **Database Designer for MySQL** functions. Controls on every **Tab** are grouped to **Tab Groups**.



Some **Tab Groups** have also **Context Options** buttons used to call options dialog specific to this **Tab Group**.

Application Button is used to open **Application Menu** with most important application actions such as opening-saving-creating new diagrams, printing and so on.

There is also **Quick Access Toolbar (QAT)** right next to **Application Button**. You can add most used actions to this special toolbar by using context-menu (right mouse button click) for every button on **Ribbon**. You can use QAT item's context menu (right mouse button click) to delete items from QAT. By default there are Open, Save and Undo commands added to QAT. You can also select whether QAT is placed below or above the **Ribbon**.

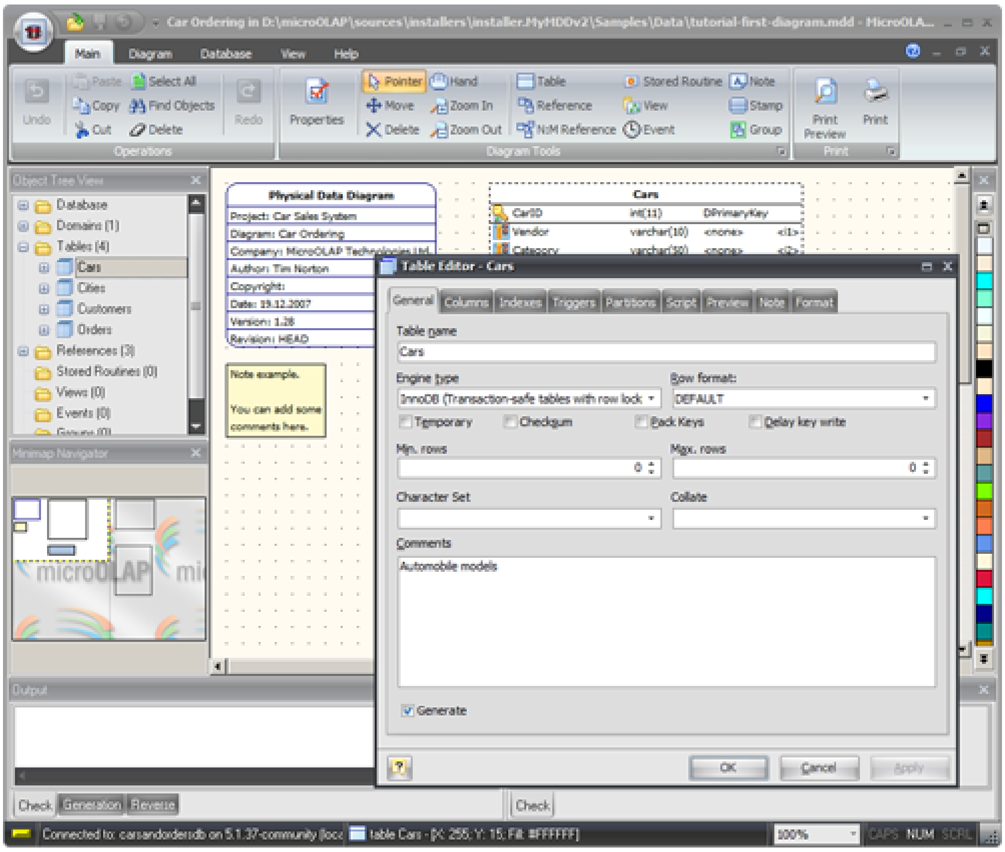
Whole **Ribbon** can be minimized if you don't have enough screen-space for diagram. You can minimize **Ribbon** by double-clicking left mouse button on Tab header or use "**Minimize the Ribbon**" context-menu item.

See also:
Workspace: Main Window Organization

5.3. GUI Skins

Starting from version 2.0 **Database Designer for MySQL** supports different visual appearance styles called skins. You can change current skin by using **Skins** drop-down control on **View** tab of the Ribbon.

Skin examples:



Car Ordering in D:\microOLAP\sources\installers\installer.MyMDDv2\Samples\Data\tutorial-first-diagram.mdd - MicroOLA...

Physical Data Diagram
Project: Car Sales System
Diagram: Car Ordering
Company: MicroOLAP Technologies Ltd.
Author: Tim Norton
Copyright:
Date: 19.12.2007
Version: 1.28
Revision: HEAD

Note example.

You can add some comments here.

Cars

CarID	int(11)	DPrimaryKey	
Vendor	varchar(10)	<none>	<I1>
Category	varchar(30)	<none>	<I2>

Table Editor - Cars

Table name: Cars

Engine type: InnoDB (Transaction-safe tables with row lock

Row format: DEFAULT

□ Temporary □ Checksum □ Pack Keys □ Delay key write

Min. rows: 0

Max. rows: 0

Character Set:

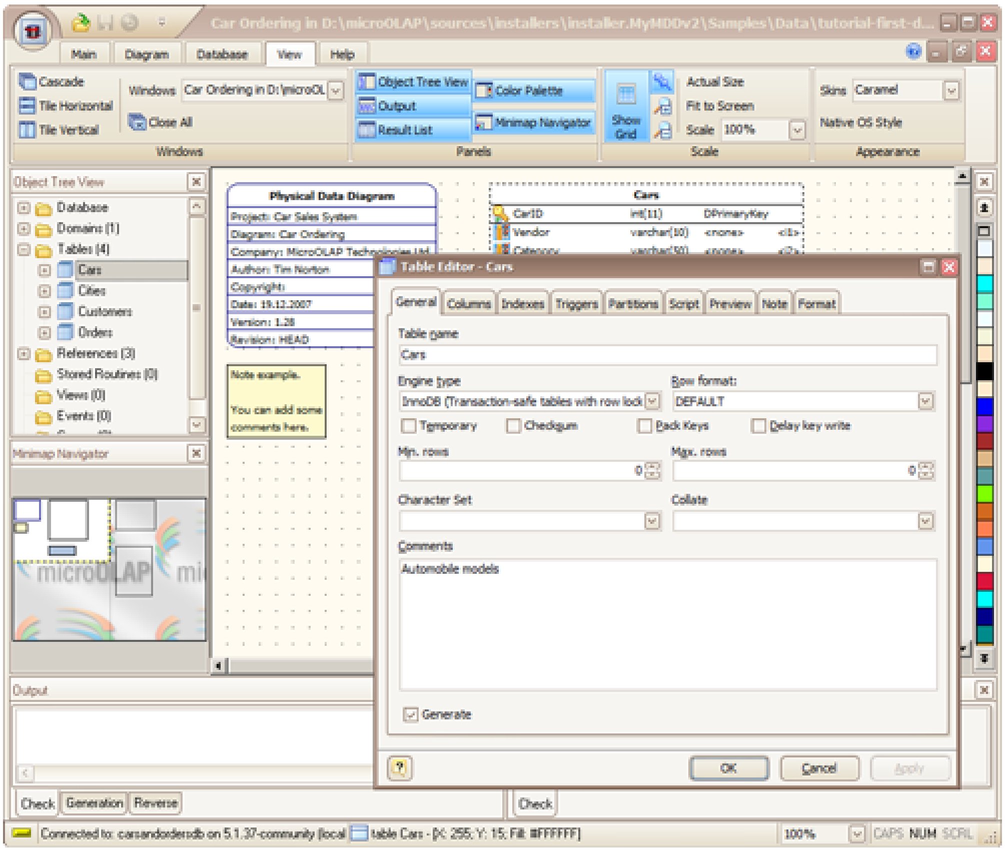
Collate:

Comments: Automobile models

☑ Generate

Connected to: carsandordersdb on 5.1.37-community (loca | table Cars - [X: 255; Y: 15; Fill: #FFFFFF] | 100%

Black



Windows: Car Ordering in D:\microOL

Scale: 100%

Skins: Caramel

Project: Car Sales System
Diagram: Car Ordering
Company: MicroOLAP Technologies Ltd
Author: Tim Norton
Copyright:
Date: 19.12.2007
Version: 1.28
Revision: HEAD

Cars			
CarID	int(11)	DPrimaryKey	
Vendor	varchar(10)	<none>	<i1>
Category	varchar(50)	<none>	<i2>

Table name: Cars

Engine type: InnoDB (Transaction-safe tables with row lock

Row format: DEFAULT

□ Temporary □ Checksum □ Pack Keys □ Delay key write

Min. rows: 0

Max. rows: 0

Character Set:

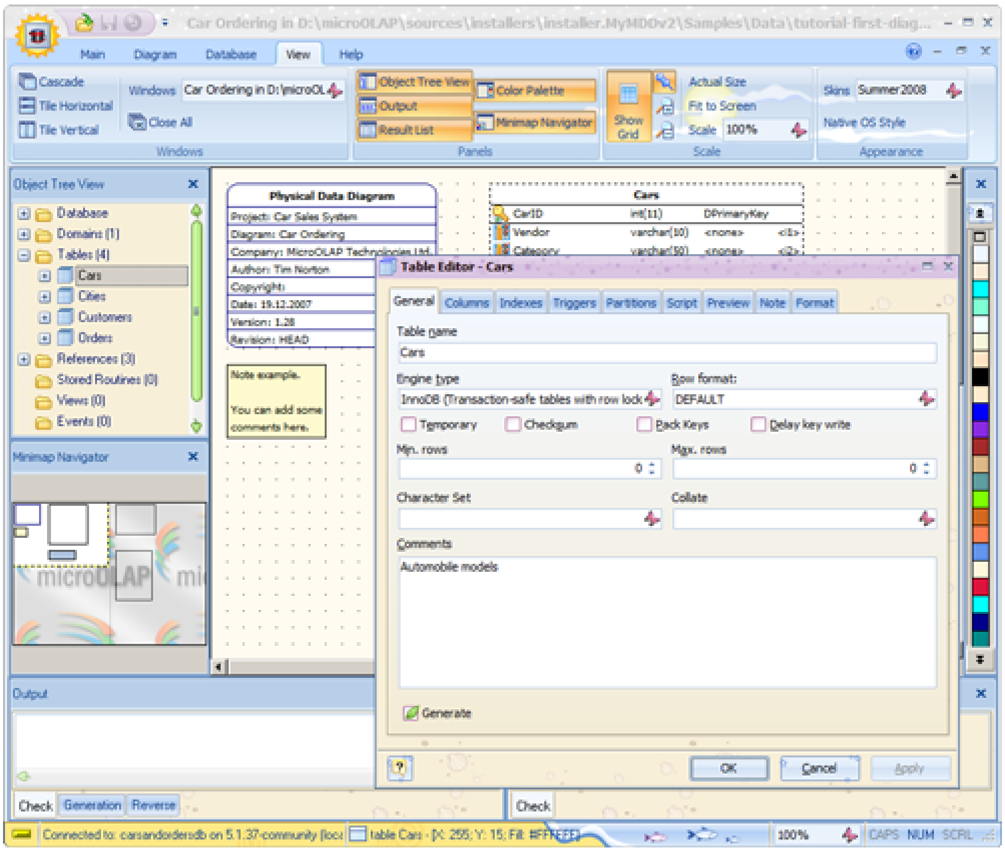
Collate:

Comments: Automobile models

☑ Generate

Connected to: carsandordersdb on 5.1.37-community (local | table Cars - (X: 255; Y: 15; Fill: #FFFFFF) | 100%

Caramel



Windows: Car Ordering in D:\microOL

Scale: 100%

Skins: Summer2008

Physical Data Diagram
Project: Car Sales System
Diagram: Car Ordering
Company: MicroOLAP Technologies Ltd.
Author: Tim Norton
Copyright:
Date: 19.12.2007
Version: 1.28
Revision: HEAD

Note example.

You can add some comments here.

Cars			
CarID	int(11)	DPrimaryKey	
Vendor	varchar(10)	<none>	<i1>
Category	varchar(50)	<none>	<i2>

Table name: Cars

Engine type: InnoDB (Transaction-safe tables with row lock

Row format: DEFAULT

□ Temporary □ Checksum □ Pack Keys □ Delay key write

Min. rows: 0

Max. rows: 0

Character Set:

Collate:

Comments: Automobile models

Connected to: carsandordersdb on 5.1.37-community (loca | table Cars - [X: 255; Y: 15; Fill: #FFFEFE] | 100% | CAPS NUM SCRL

Summer2008

You can also use **Native OS Style** button to disable skin at all and allow your GUI elements to be drawn by operating system.

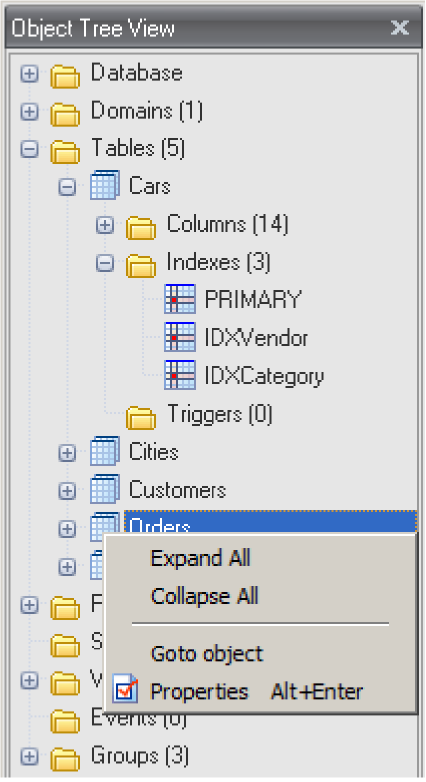
See also:
Interface User Guide: Ribbon

5.4. Docking Windows

Database Designer for MySQL user interface contains several **Docking Windows**. The product functionality is grouped in various docking windows.

In this topic you will find an overview of each **Docking Window**. Please, see the detailed description below.

Object Tree View Window



The **Diagram Tree View** contains an organized view of the diagram tables, columns, constraints, indexes, triggers, rules, stored procedures/functions, domains and references.

You can use the Tree View to:

Browse through the diagram objects

Search and locate the objects on the diagram

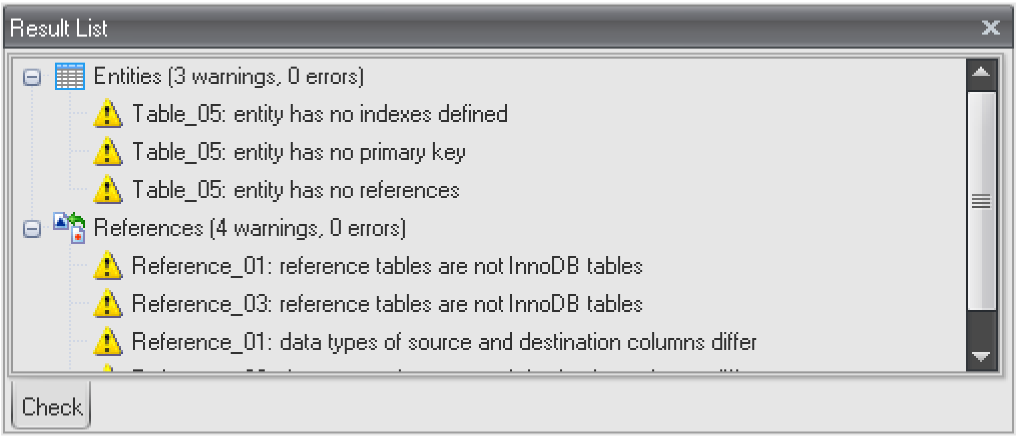
Call the appropriate properties editor for the object.

Output Window

Output window contains run-time information, which is generated by such tools as Check Diagram, Generate Database and Reverse Engineer database.

Output window consists of several tabs (Check, Generation, Reverse), each of which is intended for representing the information for an appropriate tool.

Result List Window



Entities (3 warnings, 0 errors)
- Table_05: entity has no indexes defined
- Table_05: entity has no primary key
- Table_05: entity has no references

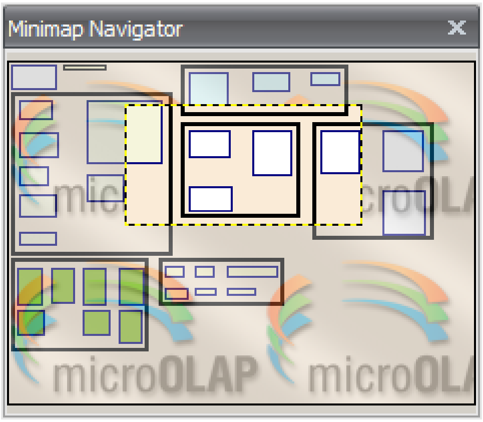
References (4 warnings, 0 errors)
- Reference_01: reference tables are not InnoDB tables
- Reference_03: reference tables are not InnoDB tables
- Reference_01: data types of source and destination columns differ

Result List window is intended to show the result of Check Diagram process.

Color Palette

Color Palette is intended to quickly set the color to visual objects. Left click will set fill color for an object, right click will set line color. If several objects selected, color setting will be applied to all of them.

Minimap Navigator



The Minimap Navigator is intended to help you move quickly around a drawing.

See also:
Interface User Guide: Main Window Organization | Minimap Navigator
Diagram Functions: Check Diagram
Database Functions: Database Generation | Database Modification

5.5. Keyboard Shortcuts

This topic describes keyboard shortcuts, that allow you to access **Database Designer for MySQL** functionality quickly.

Key	**Function**
Ctrl-N	Create a new diagram
Ctrl-O	Open an existing diagram
Ctrl-S	Save this diagram
Ctrl-P	Print this diagram
Ctrl-F4	Close this diagram
Alt-F4	Close the application
Ctrl-Z	Undo
Ctrl-Shift-Z	Redo
Ctrl-C	Clipboard Copy
Ctrl-X	Clipboard Cut
Ctrl-V	Clipboard Paste
Ctrl-F	Find diagram objects
Ctrl-A	Select all objects
Shift-F5	Redisplay diagram
F5	Set diagram actual size
F6	Zoom In diagram
F7	Zoom Out diagram
F4	Check Diagram

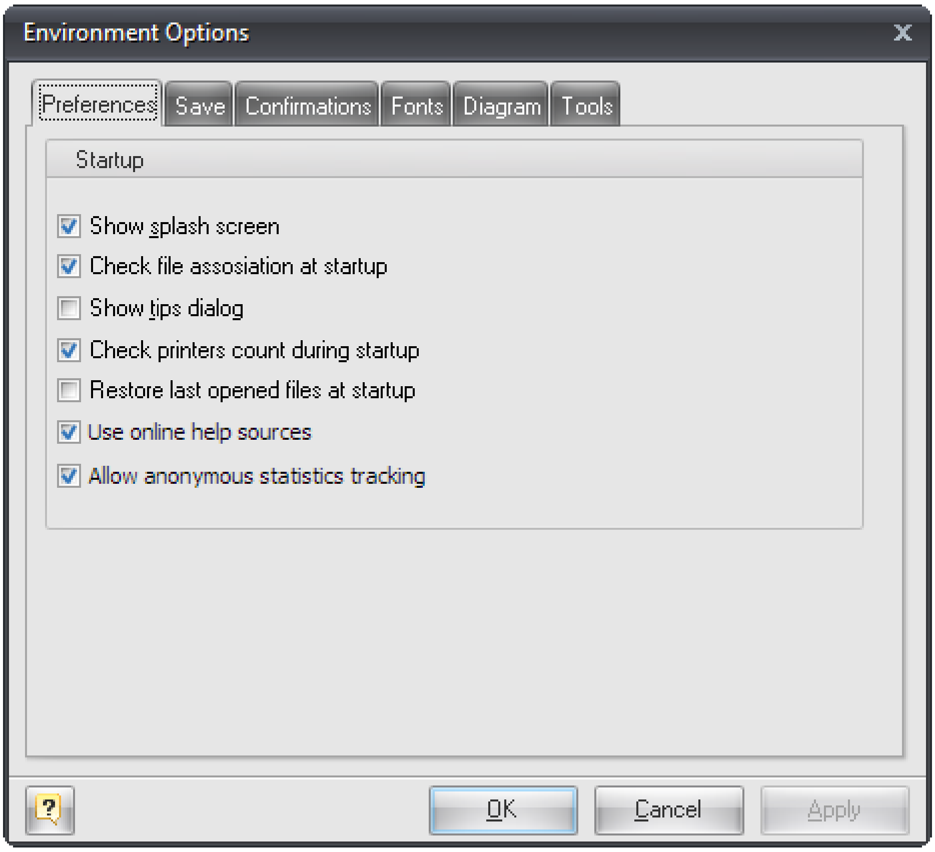
Key	Function
Shift-F6	Merge Diagram
Ctrl-G	Generate database
Ctrl-R	Reverse engineer MySQL database
Ctrl-M	Database Modification
Shift-Ctrl-E	Execute SQL statement
Shift-Ctrl-N	Connect to database
F1	Show help

5.6. Environment Options

Environment Options allow you to customize general **Database Designer for MySQL** options. To open the **Environment Options** dialog window use **Environment Options** (☑) button at the bottom of Main Application Menu.

The **Environment Options** dialog window consists of several tabs. Please, see the detailed description below:

Preferences



Startup

☑ Show splash screen
☑ Check file association at startup
☐ Show tips dialog
☑ Check printers count during startup
☐ Restore last opened files at startup
☑ Use online help sources
☑ Allow anonymous statistics tracking

In this tab you can adjust the standard behavior of the application on the startup or during the creation of new objects.

Show splash screen
This option enables you to view **Database Designer for MySQL** welcome screen at the application startup.

Check file association at startup
This option enables you to check if the diagram files are associated with **Database Designer for MySQL** at the program startup.

Show tips dialog box
This option enables you to view Tips Of the Day dialog window with useful recommendations on the application startup.

Check printers count during startup
Enables reading of the printer parameters at the application startup. If you want to check the parameters, the startup may slow down.

Restore last opened files at startup
This option enables the opening of model files which you were working with, when Designer was closed last time.

Use online help sources
Enable this option to use online manual opened in the preferred browser instead of local .CHM manual.

Allow anonymous statistics tracking

This option controls whether anonymous statistics module is enabled or not.

Save

In this tab you can change Designer behavior in Save operations.

Path to default location for Model files
Set this option to change default path for Save and Open commands.

Save backup files (.bak)
This option enables the creation of backup files for the diagram. We recommend to leave this option enabled.

Autosave model file
This option enables the sheduled model saving in the background.

Confirmations

This tab allows to enable or disable the confirmations, which pop up before some operations.

Show "Create new diagram" dialog box on reverse engineering
This option opens the dialog window that suggests you create a new diagram before Reverse Engineering and Import.

Show warning before database modification
This option opens the dialog window, that shows warning before Database Modification.

Show warning before deattaching a domain
This option opens the dialog window, that shows warning before modification of a domain-based table column.

Show warning before object deletion
This option opens the dialog window that shows warning when you're going to delete an object.

Display SQL Executor on View Data
This option opens SQL Executor with "SELECT * FROM <table name>" query when you use View Data feature from table's pop-up menu.

Fonts

In this tab you can change the diagram and application fonts.

Diagram Font
You can select the font that is used in the diagram. The option changes the font in all objects' captions as well as in the captions of all diagram tables, references, etc.

Application Font
You can select the font that will be used in the application. The option changes the font of dialog windows, menus and other objects.

Diagram

In this tab you can change the quantity of pages that will be used in new diagrams.

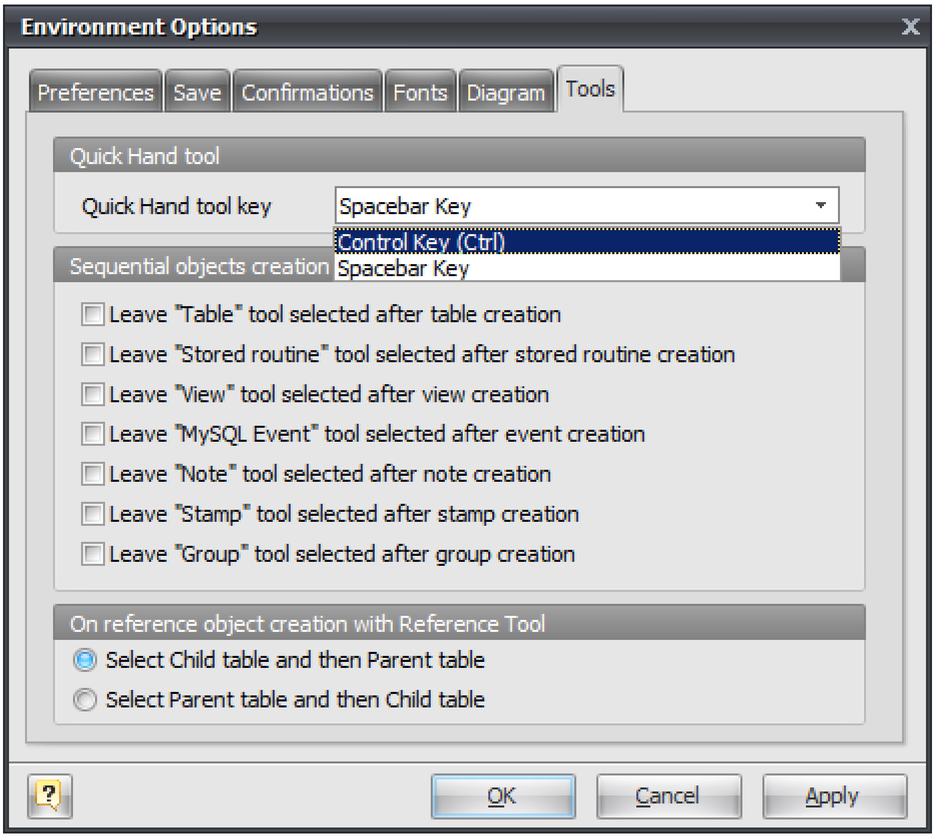
Horizontal Page Count
Horizontal size of the diagram, in pages.

Vertical Page Count
Vertical size of the diagram, in pages.

Enable undo
This option enables you to undo and redo functions of the application.

Undo limit
This option sets the limit of undo steps. The more the limit is set, the more memory is needed. The default value is set to 99.

Tools



Quick Hand tool

Quick Hand tool key: Spacebar Key
- Control Key (Ctrl)
- Spacebar Key

Sequential objects creation

□ Leave "Table" tool selected after table creation
□ Leave "Stored routine" tool selected after stored routine creation
□ Leave "View" tool selected after view creation
□ Leave "MySQL Event" tool selected after event creation
□ Leave "Note" tool selected after note creation
□ Leave "Stamp" tool selected after stamp creation
□ Leave "Group" tool selected after group creation

On reference object creation with Reference Tool

● Select Child table and then Parent table
○ Select Parent table and then Child table

In this tab you can change the behavior of tools used in designer.

Quick Hand tool key
This option sets hot key for enabling Hand tool while it is pressed. You may choose Spacebar or Control (Ctrl) key.

Sequenital objects creation group
With this option group you can manage a tool changing behavior. If checkbox is unmarked, then this tool will be set to the Pointer mode after an object is created, otherwise it will remain active.

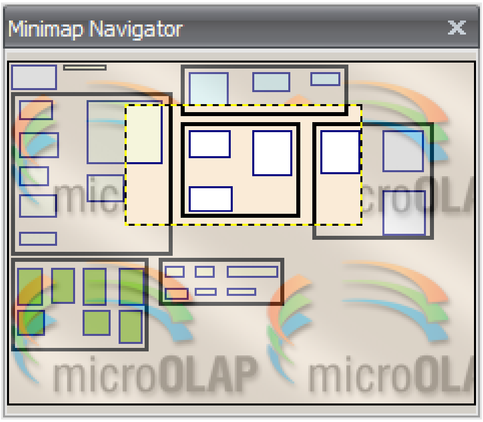
Highlight references connected to selected table
If this option is set references connected to selected table(s) will be drawn highlighted.

> **Please note:**
> Highlighted references are not selected! It is just a visualization enhancement, but not functional. To select references connected to table object you should right click on the table and then use **Select... -> Connected** references menu item.

See also:
Diagram Objects: Diagram Overview
Interface User Guide: Ribbon

5.7. Minimap Navigator

The **Minimap Navigator** is intended to help you move quickly around a drawing. It displays whole your diagram with all its objects and highlights currently visible area. You can click to some place on minimap to scroll your actual diagram to this area.



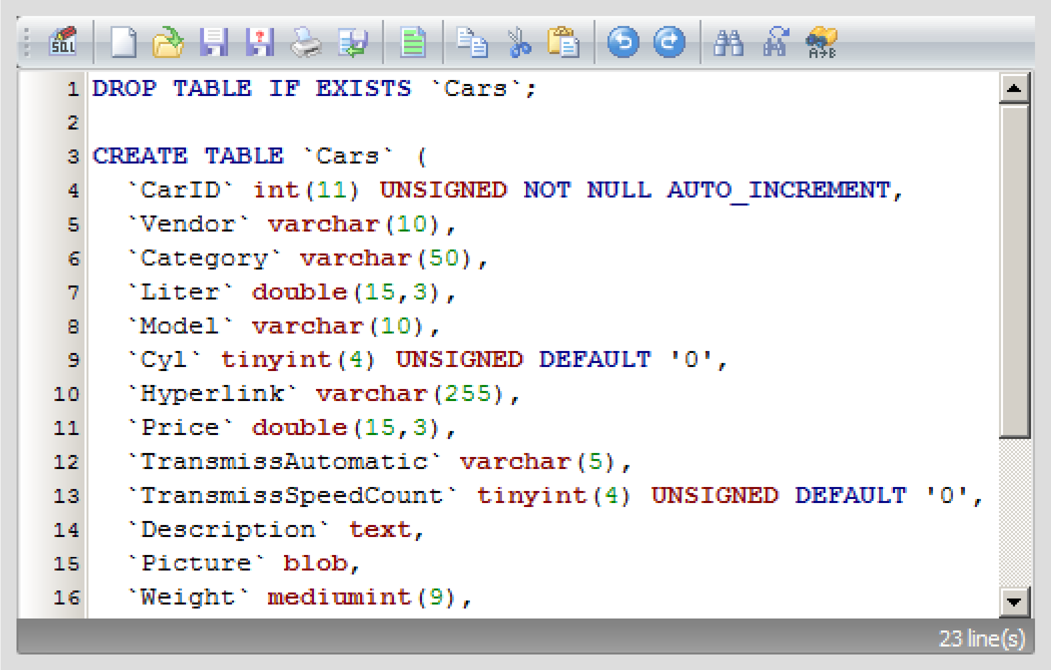
To show or hide **Minimap Navigator** use **Minimap Navigator** ( ) item on **View** tab of the Ribbon.

You can adjust **Minimap Navigator** appearance on **Minimap Navigator** tab of Diagram Display Preferences dialog.

See also:
Diagram: Diagram Display Preferences

5.8. Text Editor

Version 2.0 introduced unified **Text Editor** for all dialogs that require editing some text or SQL-script. For example, **Begin/End Script** tabs in Table Editor.



```
DROP TABLE IF EXISTS `Cars`;

CREATE TABLE `Cars` (
  `CarID` int(11) UNSIGNED NOT NULL AUTO_INCREMENT,
  `Vendor` varchar(10),
  `Category` varchar(50),
  `Liter` double(15,3),
  `Model` varchar(10),
  `Cyl` tinyint(4) UNSIGNED DEFAULT '0',
  `Hyperlink` varchar(255),
  `Price` double(15,3),
  `TransmissAutomatic` varchar(5),
  `TransmissSpeedCount` tinyint(4) UNSIGNED DEFAULT '0',
  `Description` text,
  `Picture` blob,
  `Weight` mediumint(9),
```

Toolbar buttons

- Advanced SQL Editor

- Clear editor contents

- Open file

- Save to File

- Save As

- Print text

- Insert from file

- Select All

- Copy to Clipboard

- Cut to Clipboard

- Paste from Clipboard

- Undo

- Redo

- Find Text

- Find Next

- Find and Replace text.

See also:
Diagram Objects: Table Editor

6. Diagram

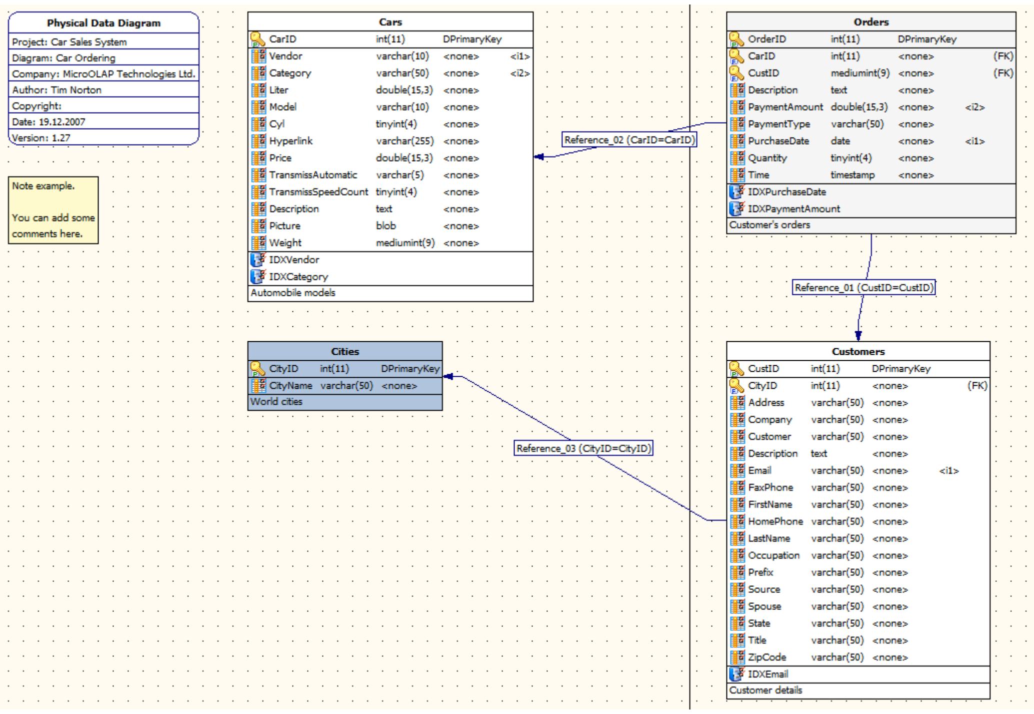
Database Designer for MySQL allows you to create physical *Entity Relationship Diagram*, that will represent object in your MySQL database.

A diagram represents the tables of your database and the relationships between them. At its core, a data diagram depicts the underlying structure of your database. It allows you to specify the data type to be used by each column in the table, and determine how tables will be stored in the database.

A diagram of your database can help you define operational aspects of your application logic that you might otherwise overlook. Also, a well-defined data diagram that accurately represents your tasks can be helpful in orienting employees to goals and operations. The data diagram can also serve as an invaluable communications tool for both internal and external constituents.

Moreover, the diagram you define does not only depict the underlying database structure, but with functionality of **Database Designer for MySQL** allows you to generate appropriate databases and even bring existing databases to the state of your diagram.

In the following picture you can see four tables: *Customers, Cities, Orders, Cars,* their columns and the references between them. Also you may see that there are indexes defined for table "*Customers*".



Physical Data Diagram
Project: Car Sales System
Diagram: Car Ordering
Company: MicroOLAP Technologies Ltd.
Author: Tim Norton
Copyright:
Date: 19.12.2007
Version: 1.27

Note example.

You can add some comments here.

Cars			
CarID	int(11)	DPrimaryKey	
Vendor	varchar(10)	<none>	<i1>
Category	varchar(50)	<none>	<i2>
Liter	double(15,3)	<none>	
Model	varchar(10)	<none>	
Cyl	tinyint(4)	<none>	
Hyperlink	varchar(255)	<none>	
Price	double(15,3)	<none>	
TransmissAutomatic	varchar(5)	<none>	
TransmissSpeedCount	tinyint(4)	<none>	
Description	text	<none>	
Picture	blob	<none>	
Weight	mediumint(9)	<none>	
IDXVendor			
IDXCategory			
Automobile models			

Orders				
OrderID	int(11)	DPrimaryKey		
CarID	int(11)	<none>		(FK)
CustID	mediumint(9)	<none>		(FK)
Description	text	<none>		
PaymentAmount	double(15,3)	<none>	<i2>	
PaymentType	varchar(50)	<none>		
PurchaseDate	date	<none>	<i1>	
Quantity	tinyint(4)	<none>		
Time	timestamp	<none>		
IDXPurchaseDate				
IDXPaymentAmount				
Customer's orders				

Cities		
CityID	int(11)	DPrimaryKey
CityName	varchar(50)	<none>
World cities		

Customers				
CustID	int(11)	DPrimaryKey		
CityID	int(11)	<none>		(FK)
Address	varchar(50)	<none>		
Company	varchar(50)	<none>		
Customer	varchar(50)	<none>		
Description	text	<none>		
Email	varchar(50)	<none>	<i1>	
FaxPhone	varchar(50)	<none>		
FirstName	varchar(50)	<none>		
HomePhone	varchar(50)	<none>		
LastName	varchar(50)	<none>		
Occupation	varchar(50)	<none>		
Prefix	varchar(50)	<none>		
Source	varchar(50)	<none>		
Spouse	varchar(50)	<none>		
State	varchar(50)	<none>		
Title	varchar(50)	<none>		
ZipCode	varchar(50)	<none>		
IDXEmail				
Customer details				

Reference_02 (CarID=CarID)
Reference_01 (CustID=CustID)
Reference_03 (CityID=CityID)

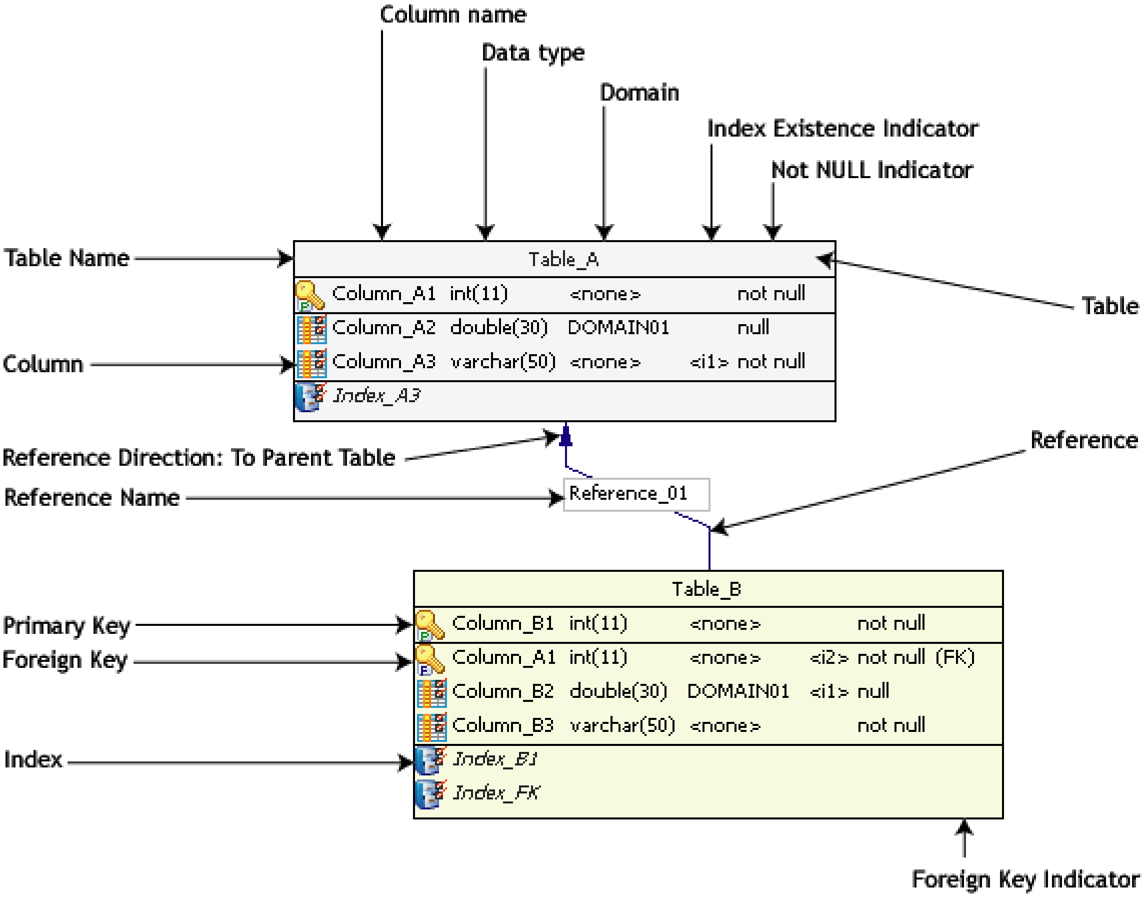
You can change the appearance of the diagram, control the number of the object attributes, which are displayed in the diagram, define the diagram grid displaying and behavior and so on in Diagram Display Preferences dialog.

See also:
Diagram: Notation | Creating a New Diagram | Diagram Display Preferences

6.1. Notation

Diagram notation

The following picture describes the diagram objects notation:



Table_A				
Column_A1	int(11)	<none>		not null
Column_A2	double(30)	DOMAIN01		null
Column_A3	varchar(50)	<none>	<i1>	not null
Index_A3				

Table_B				
Column_B1	int(11)	<none>		not null
Column_A1	int(11)	<none>	<i2>	not null (FK)
Column_B2	double(30)	DOMAIN01	<i1>	null
Column_B3	varchar(50)	<none>		not null
Index_B1				
Index_FK				

This way of diagram objects displaying is the most informative. You can change the displaying preferences according to your likes and dislikes. Please, see the Diagram Display Preferences section for more information.

See also:
Entity Relationship Diagram: Diagram Display Preferences

6.2. Creating a New Diagram

The first step to use **Database Designer for MySQL** is to create a new diagram.

To create a new diagram select the **File | New** menu item or press **Ctrl-N**. You can also use the **New diagram** button 🗋 on the **Standard** toolbar.

A new diagram will have a default name like "*Noname1*". You can change the name of your diagram in the Diagram Properties dialog window.

See also:
Diagram: Diagram Properties
Interface User Guide: Ribbon

6.3. Open an Existing Diagram

To open an existing diagram file, select the **File | Open** command from the program menu. You can also use the shortcut **Ctrl-O** or press the **File Open** button in the **Standard** toolbar. The standard **Open File** dialog box will then be displayed. Browse to the diagram file you want to open and double click on it. Also, you can open a multiple diagram at once. For that purpose, please, select the multiple diagram files you need in the **Open File** dialog window and click on the **Open** button.

Please note, that **Database Designer for MySQL** diagram files have the **.mdd** extension.

See also:
Interface User Guide: Ribbon

6.4. Saving a Diagram

You can save a diagram by choosing either the **Save** or **Save As** commands from the **File** menu. **Database Designer for MySQL** saves the diagram into a specially formatted **.mdd** file, which contains the diagram and its properties. You can also save the current diagram by clicking on the Save (💾) button in the **Standard** toolbar.

If you are saving a diagram for the first time, select the **Save** command, then type in a "file name" for your diagram and specify the location on your computer where you want to save it. Diagram file names must meet the requirements of a standard Windows filename.

If you are saving a diagram that was already saved, use the **Save** command to replace the file contents or the **Save As** command to save the diagram with a new name or to a location.

See also:
Interface User Guide: Ribbon

6.5. Selecting and Moving Objects

Once the object is in the diagram displaying area, you can select it to move it or make changes.

Selecting an object

To select an object, use the **Pointer** tool from the **Palette** toolbar, then click anywhere inside the object. The solid borders of the object will turn into the dashed ones.

Selecting multiple objects

To select multiple objects, use the **Pointer** tool from the **Palette** toolbar, then click the left mouse button and drag the selection rubber-band so that the rubber-band box encompasses the objects you want to select. Once all objects are within the rubber-band, release the mouse button.

To add objects to the list of the already selected ones, use the **Pointer** tool from the **Palette** toolbar, then click anywhere inside the object holding the **Shift** button. If the object has already been selected, it will be deselected.

> **Please note:**
> If you move mouse cursor from left to right when selecting only objects that are completely inside rubber-band will be selected. If you move mouse cursor from right to left than objects will be selected even if they partially covered by rubber-band.

Selecting all objects

You can also select all objects in the diagram. For that purpose, click on **Select All** from the right-click shortcut menu. Another way to select all objects is to use the **Edit | Select All** menu item or press **Ctrl-A**.

Deselecting objects

To deselect a selected object, click anywhere outside the selected object(s).

Moving objects

Once you have selected objects, you can change their location in the diagram. Just click within the selection with **Pointer** tool and drag the objects to the new location while holding down the left mouse button.

If **Snap to Grid** option set in Diagram Display Preferences them objects will snap to grid while moving.

See also:

Diagram: Using Snapping Grid | Diagram Display Preferences

6.6. Copying and Pasting Objects

Copying objects to the clipboard

When one or more objects are selected, you can copy those objects to the clipboard. To do that, press **Ctrl+C**, or choose the **Copy** item from the **Edit** menu. Alternatively, you can choose the **Copy** item from the right-click shortcut menu.

Pasting objects to the diagram

When you have copied the objects to the clipboard, you can paste them either into the current diagram or into another one.

To paste the objects from the clipboard into the diagram, press **Ctrl-V** or choose the **Paste** item from the **Edit** menu, you can also paste the objects by choosing the **Paste** item from the right-click shortcut menu.

Please note, that you can paste the objects not only into the current diagram, but into any other diagram. To do that, please, select the neccessary diagram from the list of the opened ones (this list can be found in the **Windows** menu), or open an existing diagram by selecting **File | Open** menu item. Certainly, you can also paste the objects into a new diagram.

See also:
Diagram: Diagram Properties
Interface User Guide: Ribbon

6.7. Using Snapping Grid

Very often it is neccessary to line up some tables and objects in a column. One of the ways to do so is to take advantage of the grid. If you click on the **Display Grid** button on the **Display** toolbar, you enable the displaying of the grid, that helps to justify the diagram objects.

The following picture illustrates such a grid:

If you enable the Snap to Grid function by clicking on the **Snap to Grid** button on the **Display** toolbar, then when you are moving a table or some other object, its upper left corner will "snap" to the nearest grid point. This feature will help you line up objects both horizontally and vertically.

Diagram objects moved into a diagram can be automatically snapped to the grid points even if the grid is not displayed on the diagram.

It is possible to change the interval between the grid points. Choose **Tools | Display Preferences** to define both the displaying of the diagram grid and behavior of the diagram, make sure that you stay in the **General** tab of the Diagram Display Preferences dialog window:

- select **Show grid** checkbox to display the grid on the diagram
- select **Snap to grid** checkbox to enable snapping feature
- modify value of **Size** field to change the interval between the grid points.

See also:
Entity Relationship Diagram: Diagram Display Preferences
Interface User Guide: Ribbon

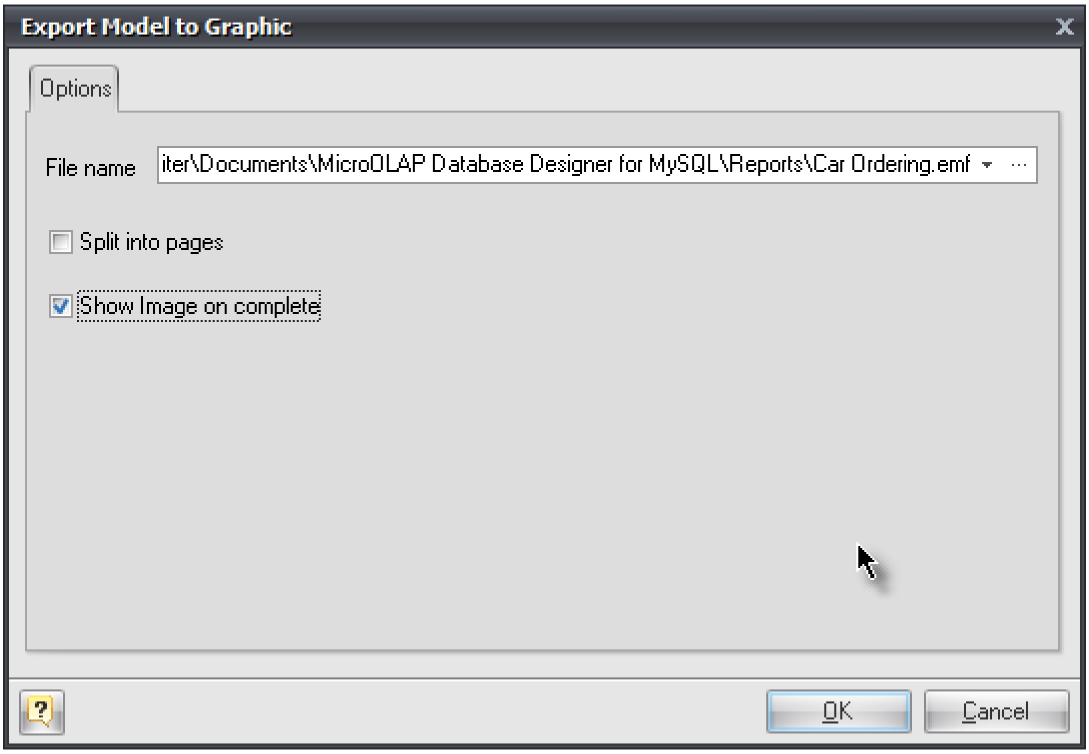
6.8. Export to Graphics

You can export your diagram to an image file and then use its graphical representation in external applications. For example, insert a diagram image into your program documentation (e.g. use MS Word), create big posters with your diagram, publish them to Web.

Database Designer for MySQL supports the following image types:

- Vector graphics: Windows Enhanced Metafile (EMF).
- Bitmap images: PNG, GIF, JPEG, BMP.

Use the **Application Menu | Export** menu item. Export Model to Graphic dialog will appear.



Export Model to Graphic

File name: iter\Documents\MicroOLAP Database Designer for MySQL\Reports\Car Ordering.emf

□ Split into pages

☑ Show Image on complete

Select file name and type in **Select image file** dialog showed by "**...**" button in **File name** editor.

You can also check **Split into pages** and/or **Show image on complete** options.

Click **OK** to save the file.

> **Please note:**
> Exporting a diagram to the vector graphics format allows you to freely scale a diagram of any size without any noticeable distortions. It enables you to create big posters with the diagram to demonstrate the data logic of your application.

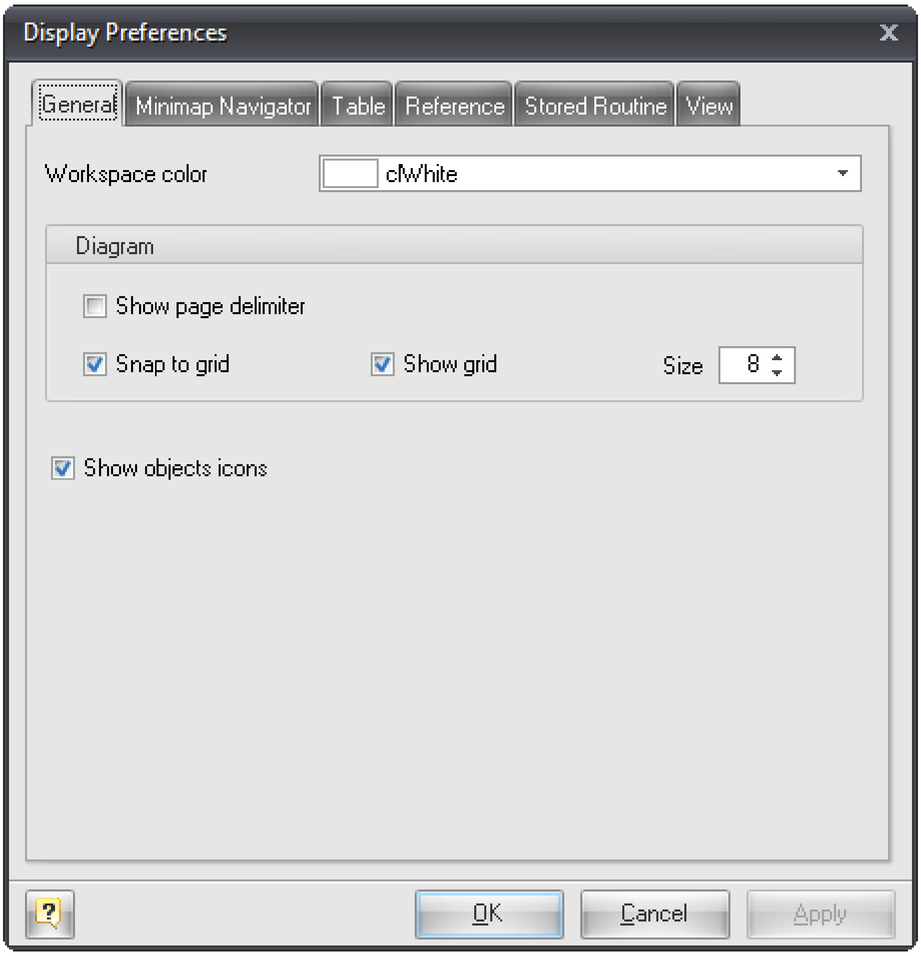
6.9. Diagram Display Preferences

Display Preferences editor allows you to change the appearance of the diagram, control the number of the object attributes, which are displayed in the diagram and define the diagram grid displaying and behavior.

To access the **Display Preferences** editor, use **Display Preferences** button on **Diagram** tab of ribbon or use diagram's context menu.

General

On this tab you can set the color of your diagram workspace, define the grid parameters and enable page delimiters displaying.



Workspace color: clWhite

Diagram

☐ Show page delimiter

☑ Snap to grid ☑ Show grid Size 8

☑ Show objects icons

Workspace Color
Use this option to set the color of the diagram background. Choose the color you need from this drop-down list.

Show Page Delimiter
Makes page delimiters visible in the diagram. This helps you to understand how the diagram will be disposed on paper. This is useful when you plan to maintain printed copy of your diagrams.

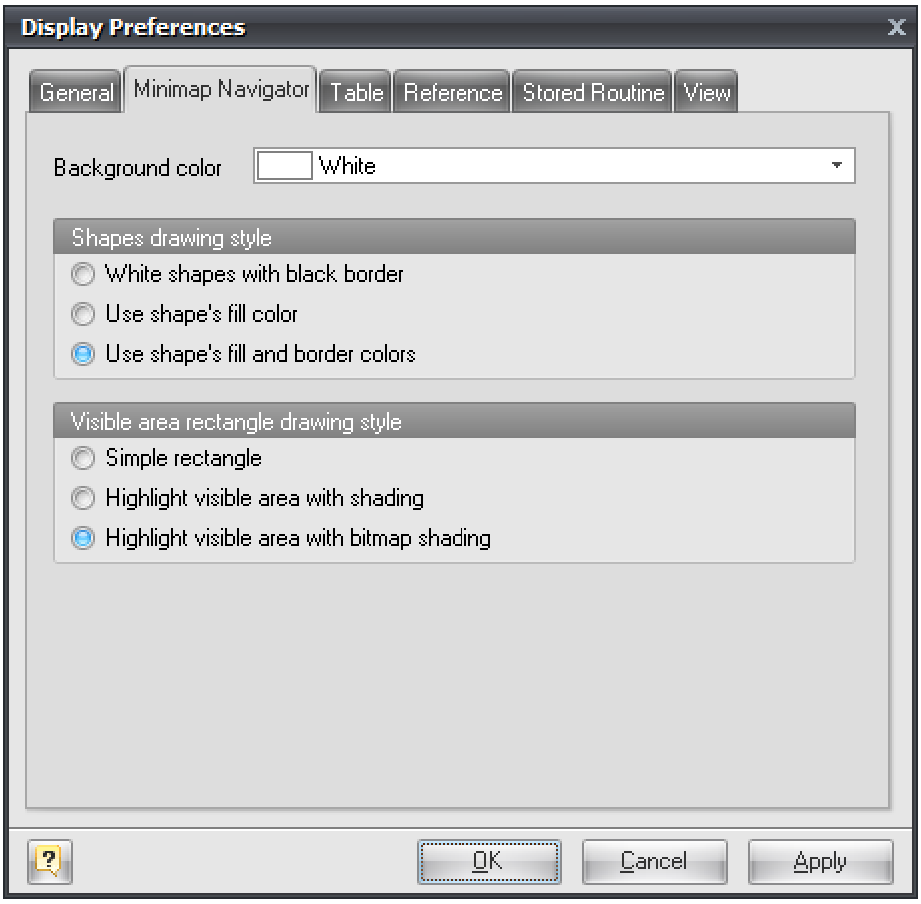
Snap to Grid
Enables snapping of the objects to the grid. If you enable this function, then when you move a table or some other object, its upper left corner will "snap" to the nearest grid point.

Show Grid
Shows the grid in your diagram.

Size
Use this option to change the interval between the grid points.

Minimap Navigator

On this tab you can setup appearance of Minimap Navigator. Options are self-explanatory.



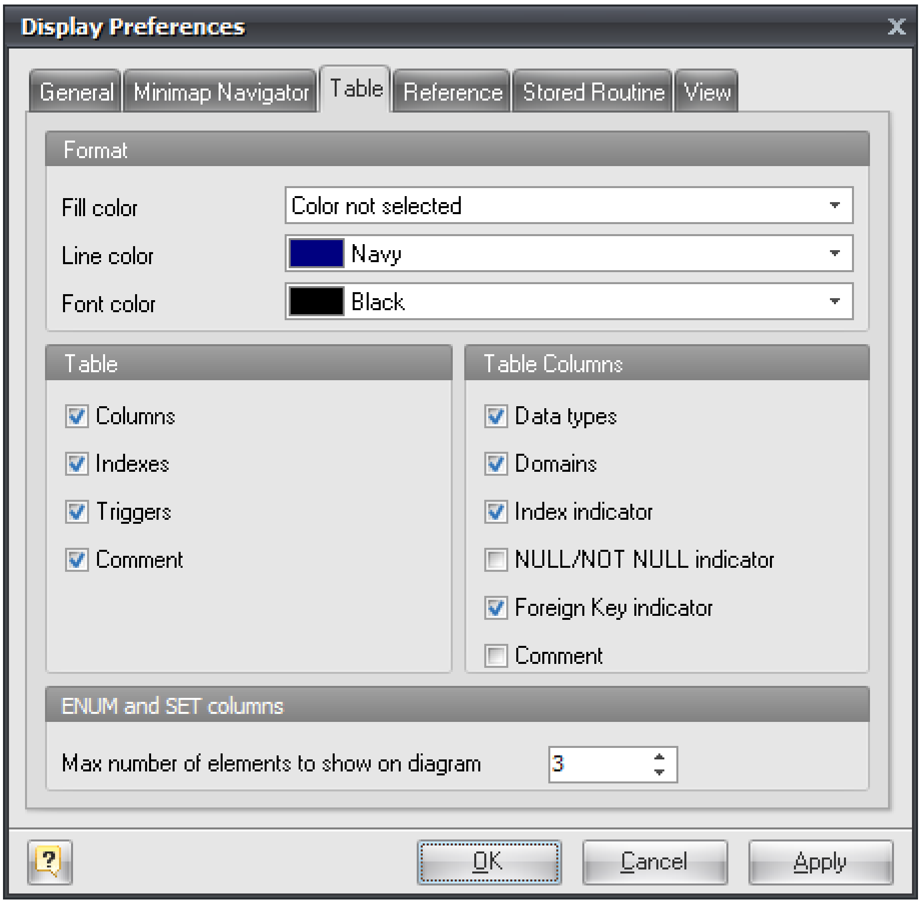
Background color: White

Shapes drawing style
○ White shapes with black border
○ Use shape's fill color
● Use shape's fill and border colors

Visible area rectangle drawing style
○ Simple rectangle
○ Highlight visible area with shading
● Highlight visible area with bitmap shading

Table

This tab allows you to define the table attributes to show in the diagram and set the default colors for the new tables. See Notation section to find out what the table parameters mean. Please note, that you can change some of the display parameters for a particular tables using **Format** tab in Table Editor.



Format

Fill color: Color not selected

Line color: Navy

Font color: Black

Table

☑ Columns

☑ Indexes

☑ Triggers

☑ Comment

Table Columns

☑ Data types

☑ Domains

☑ Index indicator

□ NULL/NOT NULL indicator

☑ Foreign Key indicator

□ Comment

ENUM and SET columns

Max number of elements to show on diagram: 3

Fill Color
Use this option to set the default color for a table background. Choose the color you need from this drop-down list.

Line Color
Use this option to set the border default color of a table. Choose the color you need from this drop-down list.

Font Color
Use this option to set the font default color used for a table displaying. Choose the color you need from this drop-down list.

Columns
Enables displaying table columns on the diagram.

Indexes
Enables displaying table indexes on the diagram.

Triggers
Enables displaying table triggers on the diagram.

Comment
Enables displaying table comments on the diagram.

Data types
Enables displaying a data type near a table column.

Domains
Show the domain indicator near a domain-based table column.

Index indicator
Shows the indexes for which this column belongs to.

NULL/NOT NULL indicator
Shows ("NOT NULL") indicator if column has NOT NULL status.

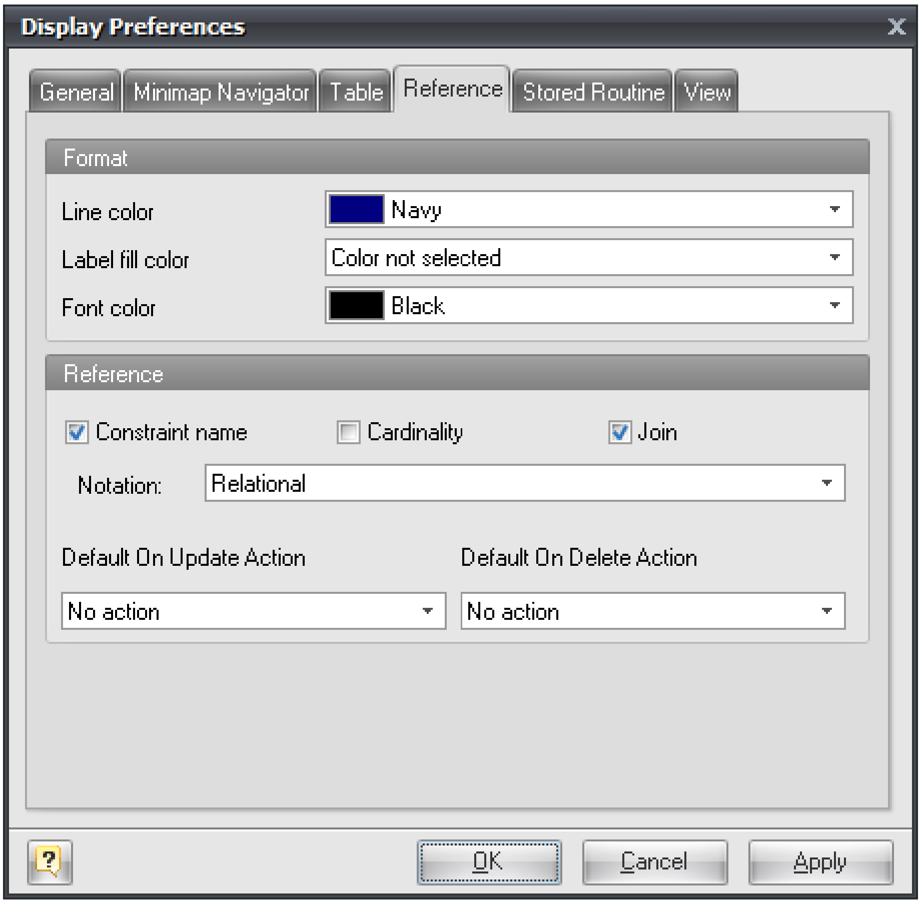
Foreign key indicator
Shows "FK" indicator if column has foreign key assigned with it.

Comment
Enables displaying column comments on the diagram.

Max number of elements to show on diagram
Sometimes ENUM or SET column can contain many values in its definition. This makes table shape very wide. You can restrict max number of displayed values using this parameter.

Reference

This tab allows you to set the reference display preferences. Note, that you can apply changes to a particular reference using Reference Editor.



Line color: Navy
Label fill color: Color not selected
Font color: Black

☑ Constraint name ☐ Cardinality ☑ Join

Notation: Relational

Default On Update Action: No action
Default On Delete Action: No action

Line Color
Use this option to set the diagram references default color. Choose the color you need from this

drop-down list.

Label Fill Color
Use this option to set the diagram reference labels default fill color. Choose the color you need from this drop-down list.

Font Color
Use this option to set the default color of reference label. Choose the color you need from this drop-down list.

Constraint Name
This option enables displaying of the reference name box in the middle of the reference.

Cardinality
This option enables displaying of the reference cardinality.

Join
This option enables displaying of the table columns used in the reference join.

Notation
Use this drop-down list to select the reference notation style:

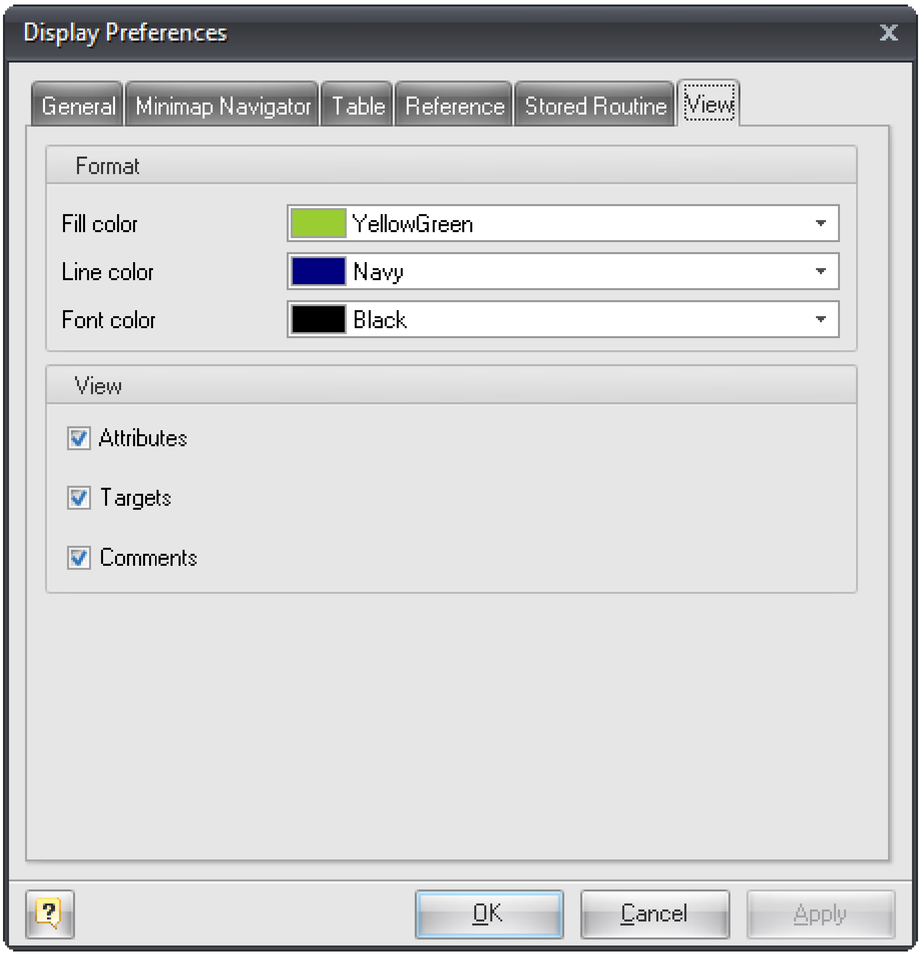
Relational. This notation shows a reading direction for references.
Conceptual (Cross Foot). This notation consists of splitting of a line into an approximation of the cardinality.

Default On Update Action
Choose On Update action that will be set for new references.

Default On Delete Action
Choose an On Delete action that will be set for new references.

Views

This tab allows you to set the display preferences of views. Note, that you can apply some changes to a particular view using View Editor.



Display Preferences

Format

Fill color: YellowGreen

Line color: Navy

Font color: Black

View

☑ Attributes

☑ Targets

☑ Comments

Fill Color
Use this option to set a default background color of a view symbol. Choose the color you need from this drop-down list.

Line Color
Use this option to set a default border color of a view. Choose the color you need from this drop-down list.

Font Color
Use this option to set the font default color used for a view displaying. Choose the color you need from this drop-down list.

Attributes
Enables displaying view attributes (columns) on the diagram.

Targets
Enables displaying view targets on the diagram.

Comment
Enables displaying view comments on the diagram.

Stored Routines

This tab allows you to set the default display preferences of stored procedures and functions. Note, that you can apply some changes to a particular stored procedure using Stored Routine Editor.

Fill Color
Use this option to set a default background color of a stored routine. Choose the color you need from this drop-down list.

Line Color
Use this option to set a default border color of a stored routine. Choose the color you need from this drop-down list.

Font Color
Use this option to set the font color used for a stored routine displaying. Choose the color you need from this drop-down list.

See also:
Interface User Guide: Minimap Navigator
Diagram: Using snapping grid | Notation
Diagram Objects: Table Editor | Reference Editor | Stored Routine Editor | View Editor

6.10. Diagram Properties

The **Diagram Properties** tool can be called be selecting **Diagram | Diagram Properties** from the ribbon-menu, or by clicking on the **Properties** menu item in diagram's popup menu.

Diagram Properties editor is a tool for setting basic diagram overview information and getting diagram numerical statistics. **Diagram Properties** editor also allows you to view diagram SQL preview.

General diagram properties

On **General** tab you can type in auxiliary information about the diagram creator, the diagram name and such. This information helps you to identify your diagram. It will be displayed on the Stamp diagram object and in the diagram reports.

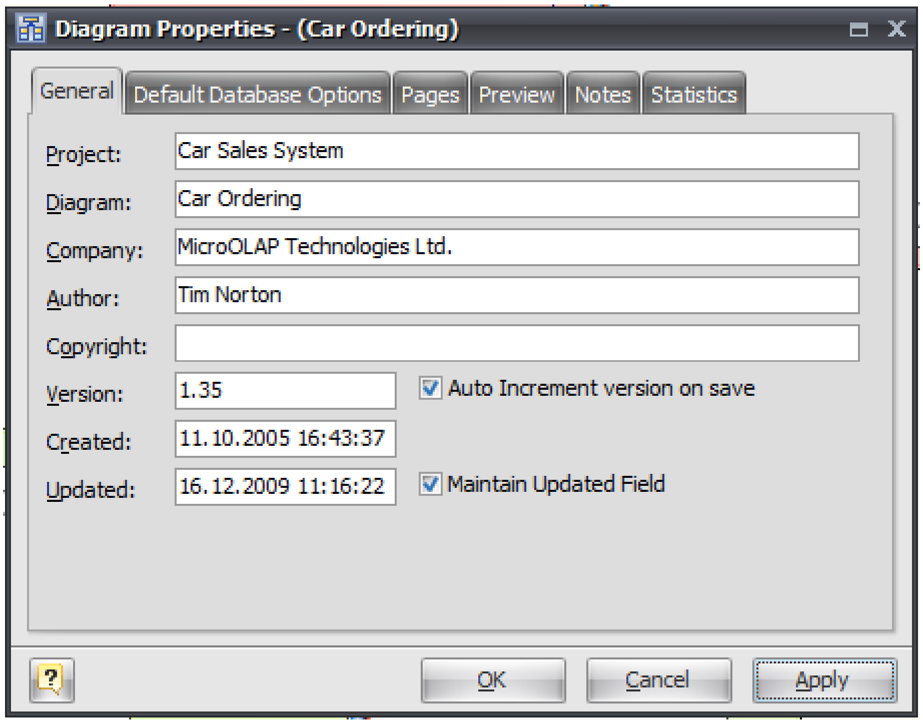


Diagram Properties - (Car Ordering)

Project: Car Sales System

Diagram: Car Ordering

Company: MicroOLAP Technologies Ltd.

Author: Tim Norton

Copyright:

Version: 1.35 ☑ Auto Increment version on save

Created: 11.10.2005 16:43:37

Updated: 16.12.2009 11:16:22 ☑ Maintain Updated Field

Project
Use this option to set a project name diagram belongs to. It may be a name of a system being developed. E.g. "Car Sales System", "Web Forum", etc.

Diagram
Use this option to set the diagram name. Note, that the diagram name will be displayed in the title of **Database Designer for MySQL** and in the list of the opened diagrams in the Windows dropdown at View tab of Ribbon. We kindly advise you to set a short and descriptive name.

Company
Use this field to set the name of the company that develops the diagram.

Author
Use this field to set the name of the person who develops the diagram.

Copyright
Use this field to set the copyright information of the model.

Version
Use this field to set the diagram version. This helps you to identify the state of the diagram.

Auto increment version on save
Set this option on to enable automatic increment of the minor part of the diagram version when saving the diagram. In other words, to change the version number from 1.0 to 1.1, 1.2, etc...

Created
This read-only field shows you the date and time when the diagram was created.

Updated
This read-only field shows you the date and time when the diagram was updated last.

Maintain Updated Field

Set this option to disable automatic updating of date and time in Updated field. This can be useful if you store your diagram in some source version control system and do not want **Database Designer for MySQL** update this field after every auto-save.

> **Please note:**
> You can place Stamp object to your diagram to display main diagram properties.

See also:
Diagram: "Default Database Options" tab | "Pages" tab | "Preview" tab | "Notes" tab | "Statistics" tab
Diagram Objects: Stamps

6.10.1. "Default Database Options" tab

This tab helps you to set the parameters, that will be applied as defaults for the newly created tables and generated SQL scripts.

To open this dialog window, please, select **Diagram | Diagram Properties** from the ribbon-menu, or click on the **Properties** menu item in diagram's popup menu. Then switch to the **Default Database Options** tab.

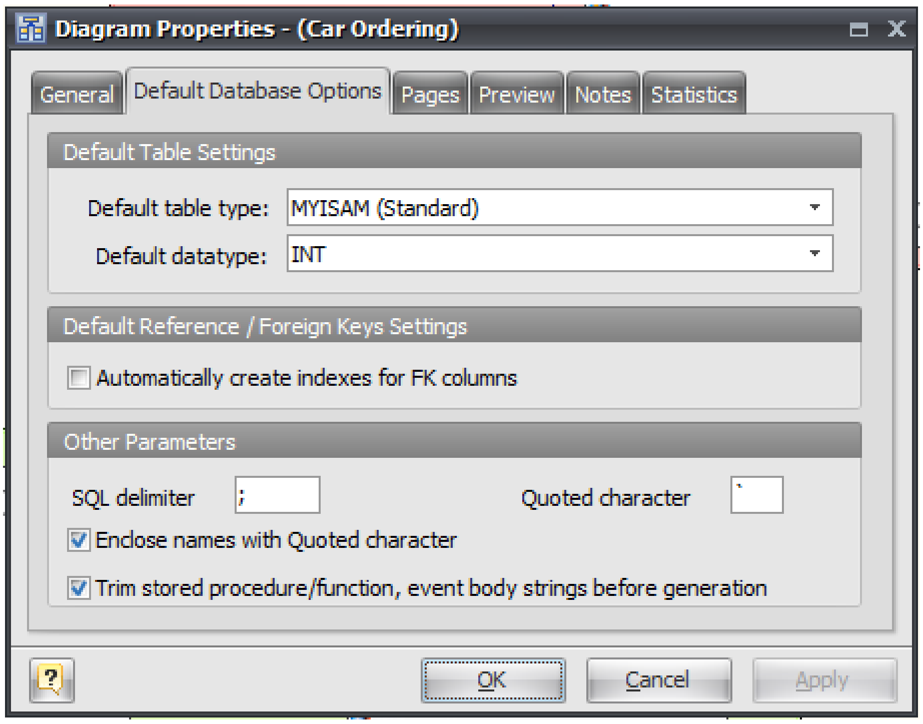


Diagram Properties - (Car Ordering)

Default Table Settings

Default table type: MYISAM (Standard)

Default datatype: INT

Default Reference / Foreign Keys Settings

□ Automatically create indexes for FK columns

Other Parameters

SQL delimiter: ;

Quoted character: `

☑ Enclose names with Quoted character

☑ Trim stored procedure/function, event body strings before generation

Default Table Type
Use this drop-down list to select MySQL table type from the MyISAM, InnoDB, Falcon etc. This option will be applied as a default table type for the newly created tables.

Default datatype
Use this drop-down list to select the data type from INT, VARCHAR, DECIMAL etc. This option will

be applied as a default data type for the newly created columns.

Automatically create indexes for FK columns
Check this option on if you want **Database Designer for MySQL** to create indexes on columns that will be foreign keys. Starting from MySQL 4.1 indexes for foreign key columns are automatically created by the server, so there is no standard need to create them manually.

SQL Delimiter
This option sets the delimiter that will be used on SQL script generation to separate SQL statements. Default MySQL delimiter symbol is ";". We strongly recommend do not change this property.

Quoted Character
This option sets the character that will be used on SQL script generation to quote table column names and values. Default MySQL quoted symbol is "`" (back-quote symbol). We strongly recommend do not change this property.

Enclose Names with Quoted character
This option forces quoting of the diagram objects names by a quote character, i.e. it makes `Column` and `Table` instead of Column and Table in SQL script.

Trim stored procedure/function, event body strings before generation
Set this option if you want to trim trailing and tailing white-spaces from stored routine or event bodies before generation SQL script.

See also:
Diagram: "General" tab | "Pages" tab | "Preview" tab | "Notes" tab | "Statistics" tab

6.10.2. "Pages" tab

This tab helps you to set the size of the diagram. You can change the number of pages in the diagram.

To open this dialog window, please, select **Diagram | Diagram Properties** from the ribbon-menu, or click on the **Properties** menu item in diagram's popup menu. Then switch to the **Pages** tab.

Horizontal Page Count
Horizontal size of the diagram, in pages.

Vertical Page Count
Vertical size of the diagram, in pages.

See also:
Diagram: "General" tab | "Default Database Options" tab | "Preview" tab | "Notes" tab | "Statistics" tab

6.10.3. "Preview" tab

This tab shows you SQL representation of your diagram. To open this dialog window, please, select **Diagram | Diagram Properties** from the ribbon-menu, or click on the **Properties** menu item in diagram's popup menu. Then switch to the **Preview** tab.

SQL-script is shown in Text Editor in read-only mode with syntax highlighting. So you can save script to file, copy to clipboard and so.

> **Please note:**
> We recommend to use this tab only during first stages of creating diagram just to preview database creation code, may be some tables creation code. When diagram becomes large opening this tab can take a lot of time. And you are not able to change any SQL-generation options or select which objects to generate. Also diagram is not checked before SQL-code generation.
>
> You need to use Generate Database dialog to generate physical database from diagram.

See also:
Diagram: "General" tab | "Default Database Options" tab | "Pages" tab | "Notes" tab | "Statistics" tab
Database Functions: Database Generation

6.10.4. "Notes" tab

This tab helps you to write descriptions and annotations for the diagram.

To open this dialog window, please, select **Diagram | Diagram Properties** from the ribbon-menu, or click on the **Properties** menu item in diagram's popup menu. Then switch to the **Notes** tab.

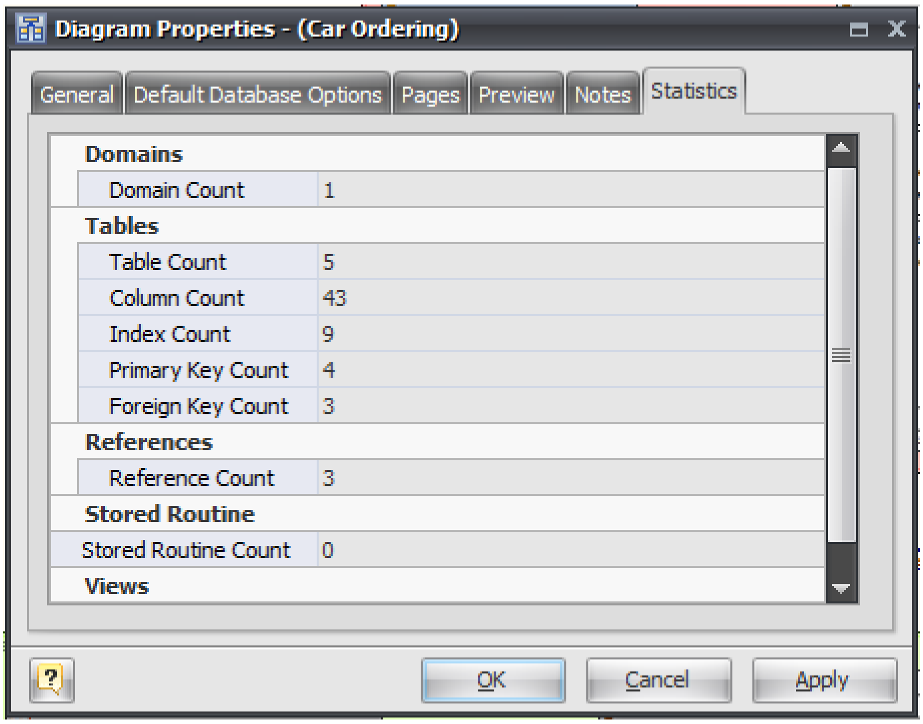
Use the bottom tabs to select between the description and annotation of the diagram.

See also:
Diagram: "General" tab | "Default Database Options" tab | "Pages" tab | "Preview" tab | "Statistics" tab

6.10.5. "Statistics" tab

This tab shows you the numerical statistics on the diagram objects.

To open this dialog window, please, select **Diagram | Diagram Properties** from the ribbon-menu, or click on the **Properties** menu item in diagram's popup menu. Then switch to the **Statistics** tab.



Domains	
Domain Count	1
Tables	
Table Count	5
Column Count	43
Index Count	9
Primary Key Count	4
Foreign Key Count	3
References	
Reference Count	3
Stored Routine	
Stored Routine Count	0
Views	

You can see how many domains, tables, columns and other objects your diagram contains.

See also:
Diagram: "General" tab | "Default Database Options" tab | "Pages" tab | "Preview" tab | "Notes" tab

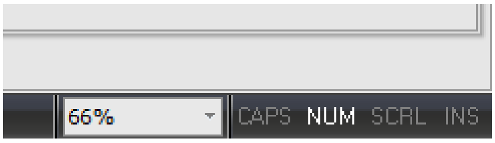
6.11. Zooming a Diagram

When a diagram is opened for the first time its zoom level is set to the normal size. You can change the zoom level of your database diagram.

There are several ways to zoom a diagram in the window:

- The most convenient variant is to use the **mouse wheel** holding **Ctrl** button. Turn up the mouse wheel to zoom in the diagram, to zoom out turn down the mouse wheel.

- In the **View** tab of ribbon you can find the **Scale** drop-down list with predefined scale factors. The same drop-down list is also placed in the status-bar.



- Press **F6** to zoom in the diagram, and **F7** to zoom out. Press **F5** to return to actual size (100%) of the diagram.

- The **Zoom-in** () and **Zoom-out** () tool buttons are placed in the **Diagram Objects** group on **Main** tab of ribbon.

- Select the **Zoom In** or **Zoom Out** menu items on **View** tab of ribbon to zoom in and to zoom out respectively.

Please, pay attention to the **Fit To Screen** function. It helps to display your diagram fully. You can call this function at the **View** tab of ribbon by clicking on the **Fit To Screen** button.

6.12. Auto Layout Diagram

The **Auto-Layout** function automatically rearranges symbols in the active diagram. The symbols are arranged in rows from left to right horizontally. This function is useful when you want to line up multiple tables in the diagram automatically.

To call this function click **Auto-Layout** button on **Diagram** tab of the Ribbon. Please note, that auto-layout operation is undoable in most cases.

See also:
Diagram: Using Snapping Grid | Zooming a Diagram

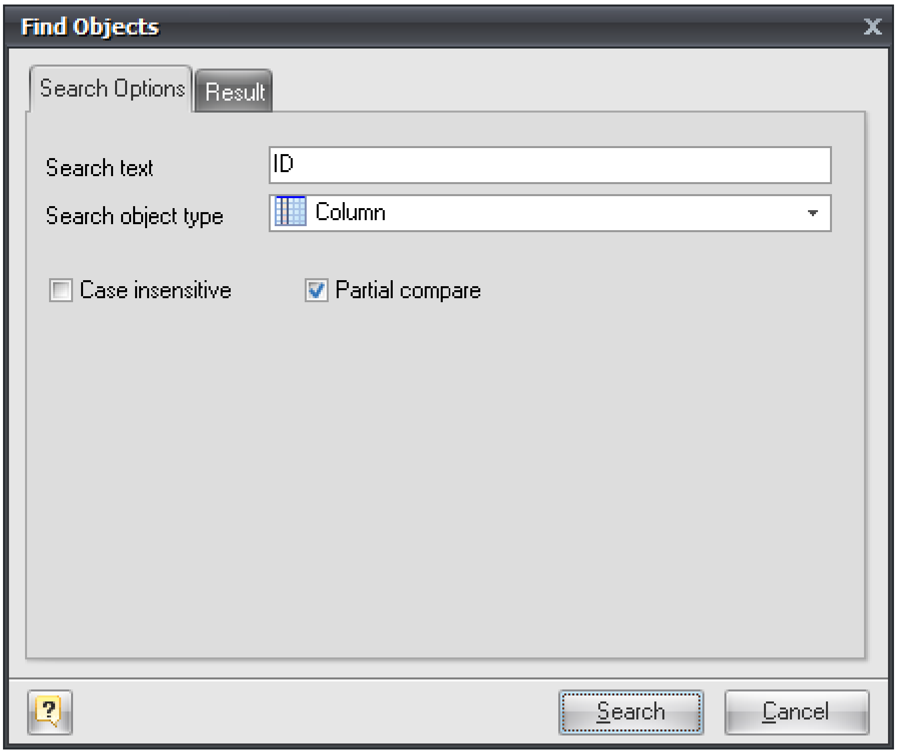
6.13. Find Objects

The **Find Objects** feature lets you search objects within the entire **Database Designer for MySQL** diagram so that you could locate the necessary objects in the diagram easily, find all the attributes related to a given object.

To call **Find Objects** dialog window, click the **Find Objects** () item on **Main** tab of Ribbon or press **Ctrl-F**.

Defining find parameters

You can define the search parameters in the **Find Objects** dialog window.



Search text: ID

Search object type: Column

□ Case insensitive ☑ Partial compare

Search text
This options allows you to define the text that you want to search in the objects and attributes names.

Search object type
You can define the object type that you want to search for. Also you can search for any type of objects. You can select the following object types:

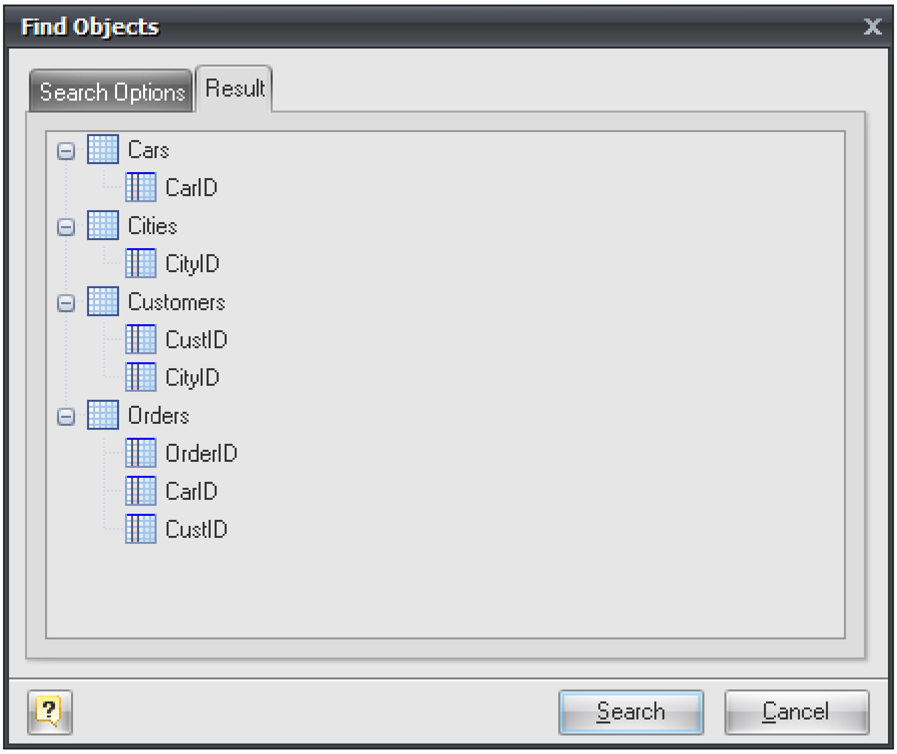
- Table
- Column
- Index
- Trigger
- Reference
- View
- Stored Routine
- Event.

Case insensitive
If you enable this option case matching will be off and it will not affect the results of your search.

Partial compare
If you enable this option you will find objects with partial name matching.

Using the Result List

The **Result** tab displays the result of the search in the result tree.



You can use the result tree to:

- learn to which objects the found objects (attributes) belong to;
- modify the found objects

Double click on the object in the result tree to call an appropriate editor for them.

7. Diagram Objects

Database Designer for MySQL allows to manage the following diagram objects:

Databases

Tables

Columns

Domains

References and Foreign Keys

Indexes

Stored Procedures and Functions

Triggers

Views

Events

[Object Groups](#) (not a database object)

[Notes](#) (not a database object)

[Stamps](#) (not a database object)

7.1. Database

Database is a **virtual** diagram object, that can be used during physical database generation process. You can set several basic parameters for the new database such as the database name, default encoding and so on.

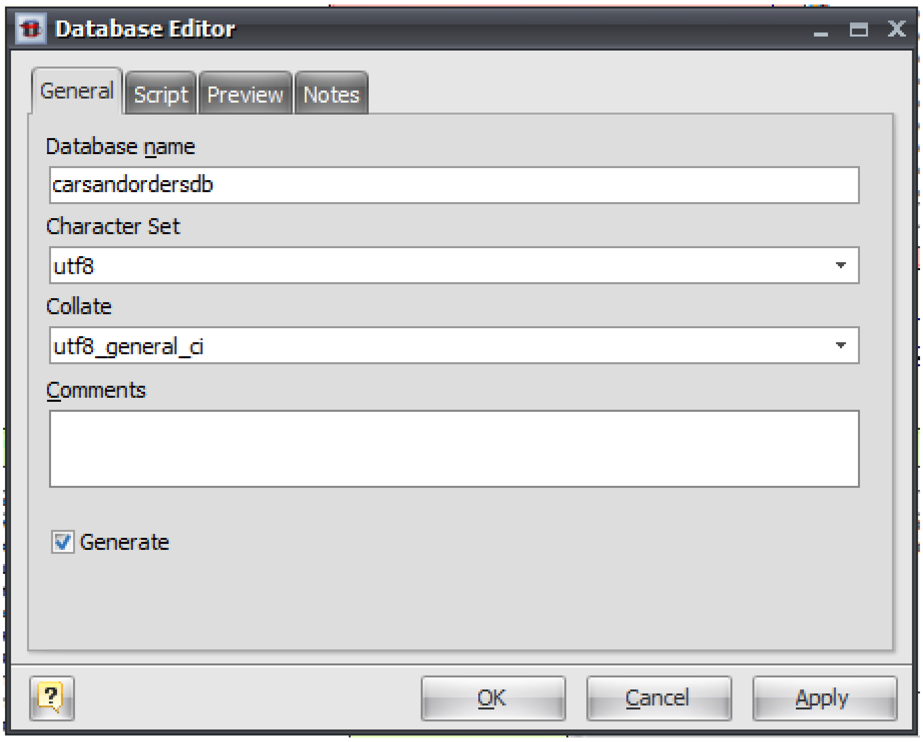
This object is especially interesting to those who want to create a new database. You may skip this section if you have an already created physical database and you only want to generate its structure.

To set up a database object use Database Editor.

See also:
Diagram Objects: Database Editor

7.1.1. Database Editor

Database Editor is helps you to set up a virtual diagram object - a database. This object can be used during database generation. To open **Database Editor** use **Database Editor** button on **Diagram** tab of ribbon.



Database name: carsandordersdb

Character Set: utf8

Collate: utf8_general_ci

Comments:

☑ Generate

Database Editor contains several tabs. Please, see the detailed description below.

General

This tab allows you to set the name of the database and other basic parameters.

Database Name
Use this field to set the database name.

Character Set
This option specifies the default database character set.

Collate
This option specifies the default database collation.

Comments
A comment for the table.

Generate
Set this option off to disable generation of the physical database. Other database objects will be generated in compliance with their own properties.

Script

This tab allows you to set SQL statements, which will be executed before (use **Begin** tab) and after (use **End** tab) generation of database.

Preview

The **Preview** tab displays the SQL statement, which will be executed during the database generation. This statement is made up according to the changes you have made using the previous tabs. Please note, that the text within the editor is read-only. The contents of this tab will be updated only when your press **Apply** button.

Notes

The **Notes** tab allows you to define the description and annotation for the edited database.

See also:
Diagram Objects: Database

7.2. Tables

Tables are the basic building blocks of a diagram. Every diagram is made up of one or more tables and various objects related to those tables.

Each table consists of a set of columns that contain the information about the type of data stored in the table. Each column must be assigned a name, data type, and length. A table can also have a set of foreign keys, indexes, triggers and partitions. Since columns, indexes, triggers and partitions are defined for a specific table, they are treated as properties of that table in a database diagram.

With **Database Designer for MySQL** you can:

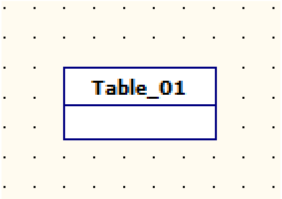
- Create new a table
- Modify the table parameters
- Modify multiple tables at once
- Manage table Columns
- Manage table Indexes
- Manage table Triggers
- Manage table Partitions

See also:
Diagram Objects: Creating a Table | Table Editor | Table Manager | Column Editor | Column Manager | Index Editor | Trigger Editor | Partitions Editor

7.2.1. Creating a Table

To create a new table in a diagram:

1. Click on the Table (▤) icon on the Main tab of ribbon. Your mouse cursor will change its appearance. Click on the diagram area to create a new table. Or right click on the empty space of diagram and choose Create Object | Table. An empty table will appear in the diagram:



2. (optional) Double click on the new table symbol in the diagram to display the Table Editor dialog window.

3. (optional) Enter the table name in the Table name field.

4. (optional) Click **OK** to save the changes.

New table will also appear on Minimap Navigator.

See also:
Interface User Guide: Minimap Navigator
Diagram Objects: Table Editor

7.2.2. Table Editor

The **Table Editor** dialog window is intended for editing the following properties of a diagram table: general and database-specific table options (table name, etc.); table columns, table indexes, table triggers, table partitions, display options and others.

To open **Table Editor**, simply double-click on a table in the diagram or select the **Properties** item from the table context menu.

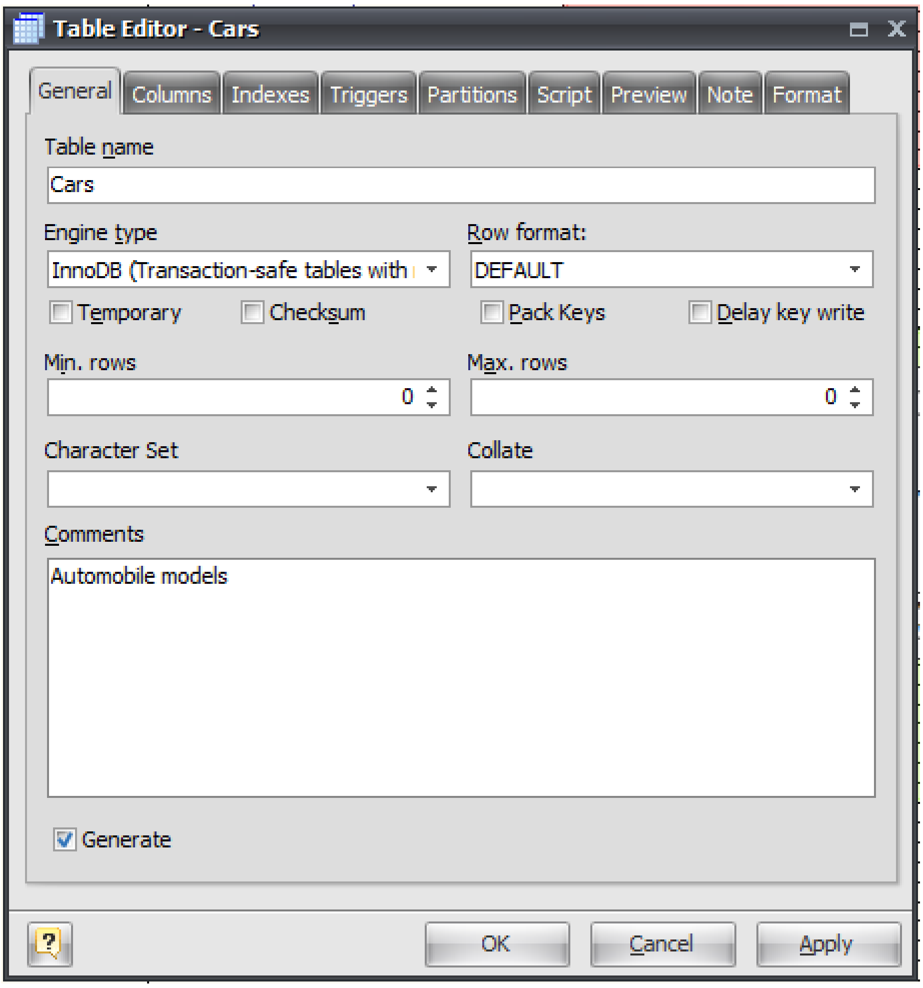


Table name: Cars

Engine type: InnoDB (Transaction-safe tables with

Row format: DEFAULT

□ Temporary □ Checksum □ Pack Keys □ Delay key write

Min. rows: 0

Max. rows: 0

Character Set:

Collate:

Comments: Automobile models

☑ Generate

Table Editor consists of several tabs. Please see the detailed description below.

General

This tab allows you to adjust the basic table properties, which are used for generating CREATE TABLE or ALTER TABLE statements for executing on the MySQL server.

Table name
The name of the table must be unique within the schema. To check your diagram for the uniqueness of table names use the Check Diagram tool.

Table Type

MySQL-specific type (storage engine) of the edited table. This type defines how MySQL stores and searches data in tables. Note, that to use MySQL foreign keys (diagram references) you should use the InnoDB table type.

> **Please note:**
> You can read about MySQL storage engines in MySQL Manual at http://dev.mysql.com/doc/refman/5.1/en/storage-engines.html

Row Format
This option defines how rows will be stored in the MySQL table.

Temporary
This option allows you to create a temporary table, i.e. a table which drops itself at the end of the session.

Checksum
This option works for MyISAM tables only. It maintains a checksum for all rows (makes table a little slower to update but makes it easier to find the corrupted tables).

Pack keys
This option works for MyISAM and ISAM tables only. It makes table indexes smaller. This usually makes updates slower and reads faster.

Delay key write
This option works for MyISAM tables only. It delays key table updates until the table is closed.

Min. rows
This option defines minimum number of rows you plan to store in the table. Leave zero for the default value.

Max. rows
This option defines maximum number of rows you plan to store in the table. Leave zero for the default value.

Character Set
This option specifies the table character set.

Collate
This option specifies the table collation.

Comments
MySQL comment for your table.

Generate Table
Set this option off to exclude the table from the default selection of generated tables in the Database Generation and Database Modification tools.

Columns

Use the **Columns** tab for adding, modifying, and deleting table columns. Please see the topic Column Editor for the detailed information.

Indexes

The **Indexes** tab is intended for managing table indexes. Please see the Index Editor topic for the detailed information.

Triggers

The **Triggers** tab is intended for managing table triggers. Please see the Trigger Editor topic for the detailed information.

Partitions

The **Partitions** tab is intended for managing table partitions. Please see the Partition Editor topic for the detailed information.

Script

This tab allows you to set SQL statements, which will be executed before (use **Begin** tab) and after (use **End** tab) generation of the table.

Preview

The **Preview** tab displays the SQL statement, which will be executed during the database generation. This statement is made up according to the changes you have made using previous tabs. Note, that the text within the editor is read-only.

Note

The **Note** tab allows you to define a description and an annotation for the edited table. This properties will not affect the physical MySQL database, but they can be useful for your diagram development.

Format

These options allow you to set table line and fill color for displaying on the diagram, different from the default table colors, which are defined within the Diagram Display Preferences dialog. Please see the Formatting Table topic for the detailed information.

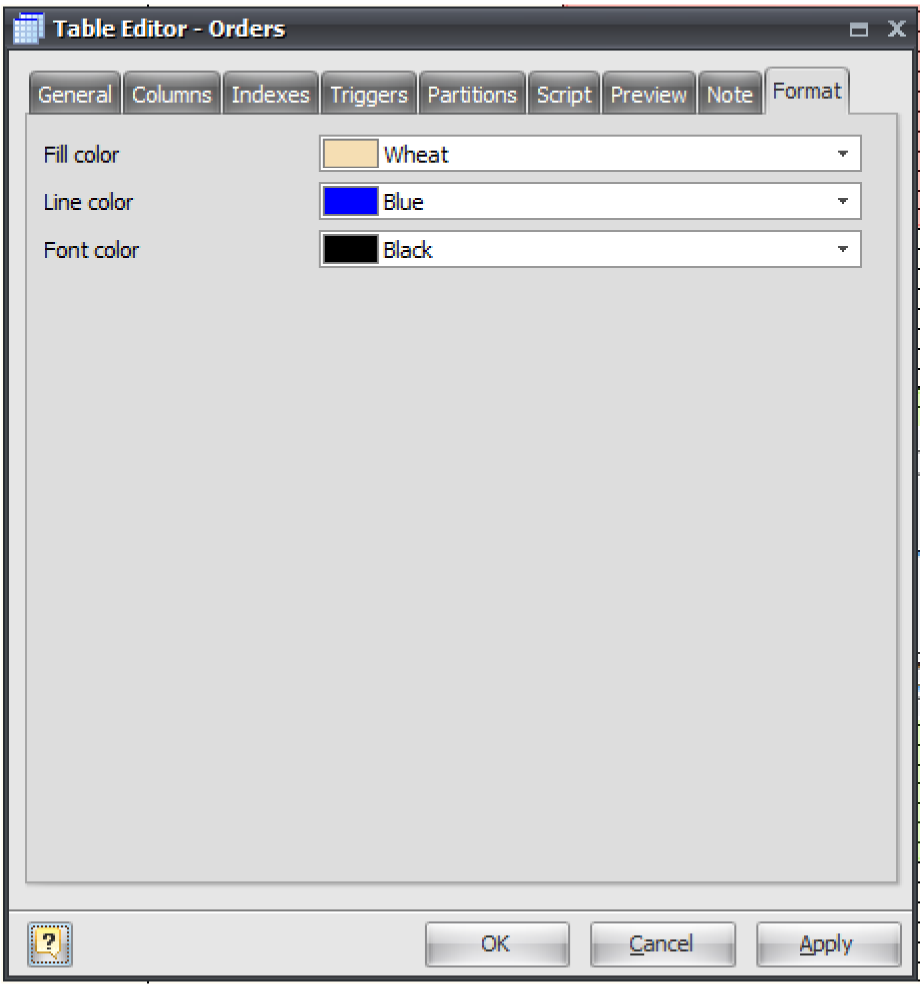
See also:

Diagram Objects: SQL Table Definition | Column Editor | Domain Manager | Index Editor | Trigger Editor | Partitions Editor | Formatting Table | Tables in Tree View Window | Table Manager
Database Functions: Database Generation | Database Modification
Diagram: Diagram Display Preferences

7.2.3. Formatting Table

You can set table line and fill color for displaying on the diagram, different from the default table colors, which are defined within the Diagram Display Preferences dialog.

Double click on the table symbol on diagram and Table Editor will appear. Go to the **Format** tab.



Fill color: Wheat

Line color: Blue

Font color: Black

Fill Color
Use this option to set a background color of the table. Choose the color you need from this drop-down list.

Line Color
Use this option to set a border color of the table. Choose the color you need from this drop-down list.

Font Color
Use this option to set a font color of the table. Choose the color you need from this drop-down list.

See also:
Diagram Objects: Table Editor

Diagram: Diagram Display Preferences

7.2.4. SQL Table Definition

Table SQL preview

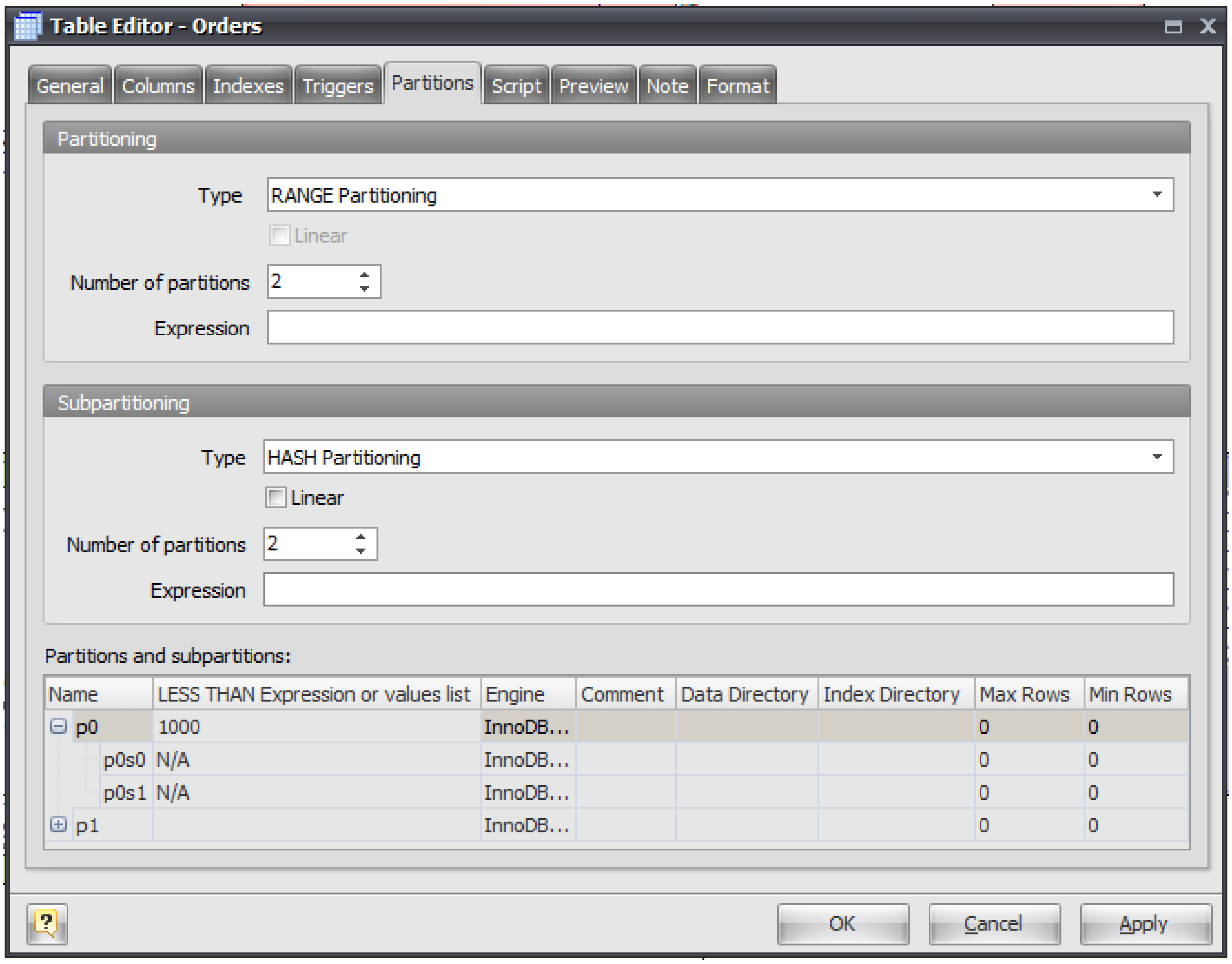
You can see the SQL representation of your table, it includes columns, indexes, foreign keys, etc. Actually this is SQL-code to be executed while generating database.

To see SQL representation of the table, double click on the table symbol and Table Editor will appear. Go to the **Preview** tab. SQL representation is shown using standard Text Editor. So you can save to file, copy to clipboard, print SQL-code and so on.

See also:
Diagram Objects: Table Editor

7.2.5. Partitions Editor

The **Partitions Editor** is placed within the Table Editor dialog. It allows you to set up table partitioning options, modify the list of table partitions and subpartitions as well as partition properties. Click the **Partitions** tab of the Table Editor to manage table partitions.



Partitioning

Type: RANGE Partitioning

□ Linear

Number of partitions: 2

Expression:

Subpartitioning

Type: HASH Partitioning

□ Linear

Number of partitions: 2

Expression:

Partitions and subpartitions:

Name	LESS THAN Expression or values list	Engine	Comment	Data Directory	Index Directory	Max Rows	Min Rows
⊟ p0	1000	InnoDB...				0	0
p0s0	N/A	InnoDB...				0	0
p0s1	N/A	InnoDB...				0	0
⊞ p1		InnoDB...				0	0

> **Please note:**
> Table partitioning was introduced with MySQL 5.1 version. This is quite complicated feature of MySQL with a lot of options, settings and restrictions. You can find more about partitioning in MySQL Manual at http://dev.mysql.com/doc/refman/5.1/en/partitioning.html

Partitions tab consists of following areas:

Partitioning
Allows to set up basic partitioning options

Subpartitioning
If table has two levels of partitioning (subpartitions) this options group allows to set up subpartitioning options.

Partitions and subpartitions
This table displays all table partitions and subpartitions according to settings in two previous areas. It allows you to define properties of particular partitions and subpartitions.

Partitioning

Type
Defines subpartitioning type

Linear
Allows to add LINEAR attribute for KEY and HASH sub-partitioning types

Number of partitions
Defines number of sub-partitions for every partition in a table

Expression
Allows to define *expr* part of *PARTITION BY RANGE|LIST|HASH|KEY (expr)* statement.

Subpartitioning

Type
Defines partitioning type

Linear
Allows to add LINEAR attribute for KEY and HASH partitioning types

Number of partitions
Defines number of partitions for table

Expression
Allows to define *expr* part of *SUBPARTITION BY RANGE|LIST|HASH|KEY (expr)* statement.

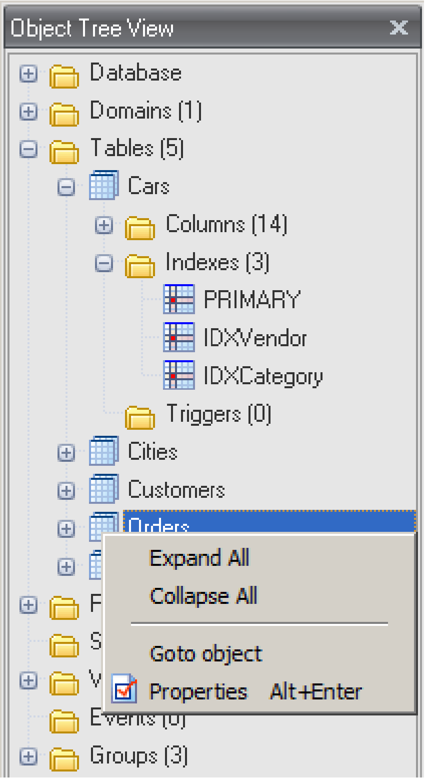
Partitions and subpartitions

This table allows you to set up options (such as name, storage engine or *expr* part of *LESS THAN(expr)* statement) for particular partition or subpartition.

See also:
Diagram Objects: Table Editor

7.2.6. Tables in Tree View Window

You can access to the diagram tables and browse through their objects by mean of the Object Tree View, that represent all the diagram objects, including tables, in a tree-like structure.



To expand table, click on the "**+**" sign near the table name, it will show you table columns and indexes. Click on the "**-**" sign to roll up the list.

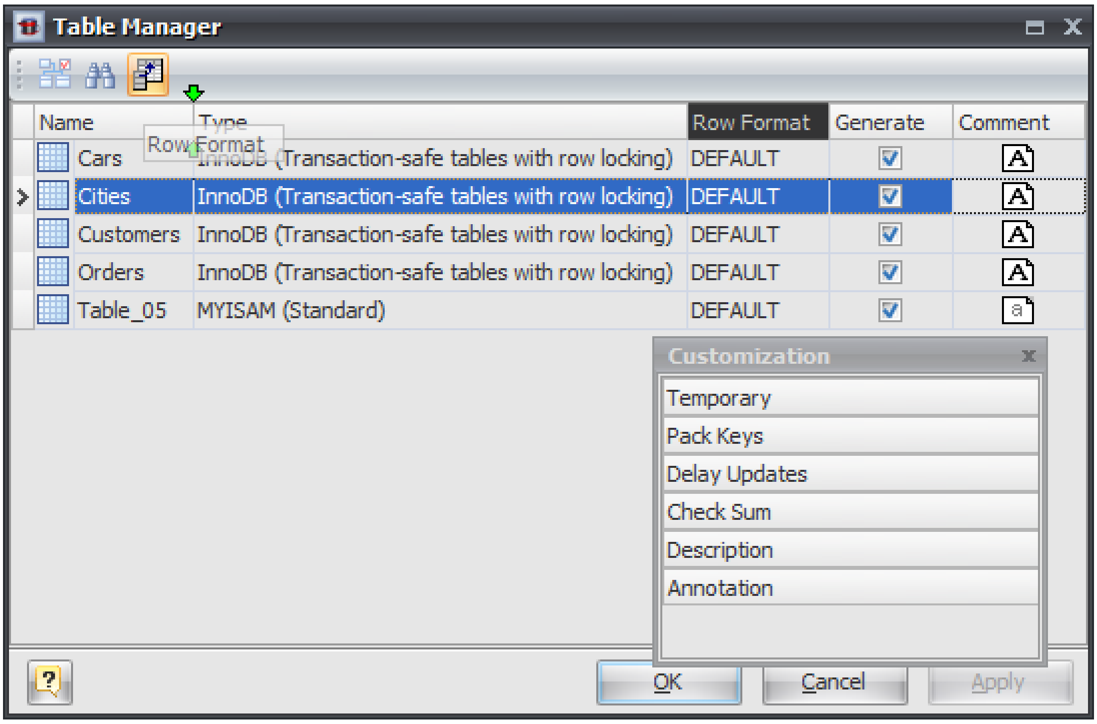
Examine the following table-related features in the Object Tree View:

- Double click on a table to call Table Editor for it. You can also use Properties menu item from context menu.
- Right-click on a table to see shortcut menu. Select the Goto object item to select the table on the diagram and scroll diagram to it.
- Double-click on a column to call the Column Editor.
- Double-click on an index to call the Index Editor.
- Double-click on an trigger to call the Trigger Editor.

See also:
Interface User Guide: Docking Windows
Diagram Objects: Column Editor | Table Editor | Index Editor

7.2.7. Table Manager

Database Designer for MySQL has a great feature that allows you to modify the basic parameters for multiple tables at once. This is the **Table Manager**. To run this tool use **Table Manager** item on **Diagram** tab of Ribbon or press **Ctrl+2**.



Name	Type	Row Format	Generate	Comment
Cars	InnoDB (Transaction-safe tables with row locking)	DEFAULT	☑	A
Cities	InnoDB (Transaction-safe tables with row locking)	DEFAULT	☑	A
Customers	InnoDB (Transaction-safe tables with row locking)	DEFAULT	☑	A
Orders	InnoDB (Transaction-safe tables with row locking)	DEFAULT	☑	A
Table_05	MYISAM (Standard)	DEFAULT	☑	a

Modifying parameter values

The grid in **Table Manager** represents the basic parameters of all diagram tables. The rows stand for the tables, and columns for the table parameters. This parameters can be changed by using Table Editor for each table one by one, but with **Table Manager** you can do it much more quickly.

Click on the cell to modify the parameter value. Depending on the parameter type, the activated in-place editor can be a text box, a drop-down list, a check box, etc. Please refer to the Table Editor topic to find out more about all table parameters.

To save your changes click the **OK** button. If you want to store changes and continue editing, click on the **Apply** button.

Customizing the parameter visibility

By default, there are only a few table parameters accessible in the grid, but you can make other table parameters visible as well. Click on the **Customize** button ( ) in the dialog toolbar for this purpose. The **Customize** tool window will appear.

The **Customize** tool window contains the list of table parameters, which are inaccessible in the main grid. To add a new column to the grid, select the required parameter in the list and drag to the new position in the grid. Two green arrows will indicate that dropping is allowed, and after you drop it, the new column will appear. To remove a column from the grid, drag the column header to the **Customize** list in the similar way.

Thus you can change the following tables parameters in the **Table Manager**:

Name

Type

Row Format

Generate

Comment

Temporary

Pack Keys

Check Sum

Description

Annotation.

Please refer to the Table Editor topic to find out more about this parameters.

Searching table in the list

If your diagram contains large number of tables, it may be important to search for a table in a most easy way. To find a table quickly by its name, click the **Find** button () at the dialog toolbar. The **Find Name** dialog will appear. Type in the name of table to find and click **OK**. If the table is found the grid cursor positions on the appropriate row.

Running the Table Editor

If you want to change some unavailable table parameters and attributes (such as columns and indexes), select the appropriate table in the dialog and click the **Edit** button () of the dialog toolbar. The Table Editor will appear, where you can edit all the parameters for the selected table.

See also:
Diagram Objects: Table Editor

7.3. Columns

In a table data is arranged into columns. A column stores data element, such as a person name, a price, or any similar type of information. When columns are created in a table, they are given a name that identifies their purpose and role, such as PersonName or Price. Usually, you must also specify additional properties, such as data type and how long the longest entry in the column will be, other properties can include the column is the table primary key.

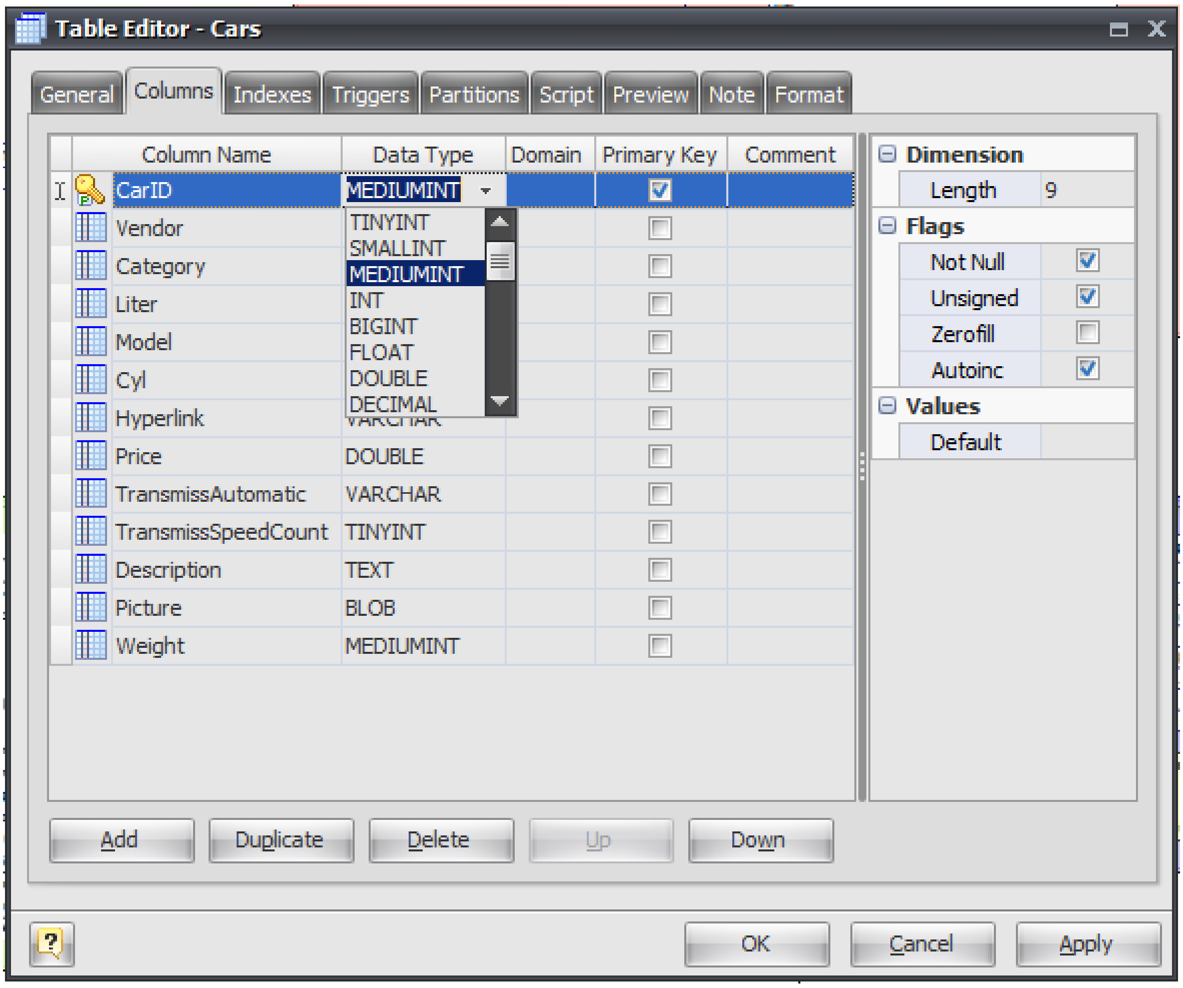
With **Database Designer for MySQL** you can:

- Edit table columns
- Edit multiple table columns at once.

See also:
Diagram Objects: Column Editor | Column Manager | Creating a Table

7.3.1. Column Editor

The **Column Editor** is placed within the Table Editor dialog. It allows you to modify the list of table columns as well as column properties. Click the **Columns** tab of the **Table Editor** to manage table columns.



Column Name	Data Type	Domain	Primary Key	Comment
CarID	MEDIUMINT		☑	
Vendor			☐	
Category			☐	
Liter			☐	
Model			☐	
Cyl			☐	
Hyperlink	VARCHAR		☐	
Price	DOUBLE		☐	
TransmissAutomatic	VARCHAR		☐	
TransmissSpeedCount	TINYINT		☐	
Description	TEXT		☐	
Picture	BLOB		☐	
Weight	MEDIUMINT		☐	

Dimension	
Length	9
Flags	
Not Null	☑
Unsigned	☑
Zerofill	☐
Autoinc	☑
Values	
Default	

The **Column Editor** consists of the following areas:

- Column List
- Properties Pane
- Button Pane.

Column List

The column list displays all the columns in the table and allows you to modify the following column properties:

Column name
The name of the column, which must be unique within the table;

Data type
The type of the column, which specifies data to store in the column;

Domain
You can specify a column domain, see the Domains topic for detailed information;

Primary key
Specify this option to include the field into the table primary key;

Comment
An arbitrary description for the column.

Properties Pane

The properties pane allows you to define the advanced properties of the column, selected in the **Column List**. The appearance of this pane changes according to the data type of the column. These properties are:

Length
This attribute defines the maximum allowed length of the stored values; it applies to all integer, decimal, and string types;

Decimals
This attribute defines the number of digits, which follow the decimal point;

Not null
This option indicates that the stored column value cannot be NULL;

Autoinc
This attribute makes the column value autoincrement, i.e. each new value is set automatically according to the previous value; it applies to all integer values;

Binary
This attribute indicates that the string value is stored in a binary form within the column;

Unsigned
This option indicates that the stored integer or decimal value must be above zero;

Zerofill
With this attribute all unspecified digits of the field value are filled as zeros according to the column length and decimals; it applies to all integer and decimal types;

Set
This attribute defines the list of permissible values for the field of the ENUM or SET type;

Default

This attribute defines the default value, which the column accepts if no other is specified.

Buttons Pane

The buttons under the list of columns allow you to perform the following actions:

Add
Add a new column with the default properties to the end of the list;

Duplicate
Add a new column with the same properties as the selected column to the end of the list;

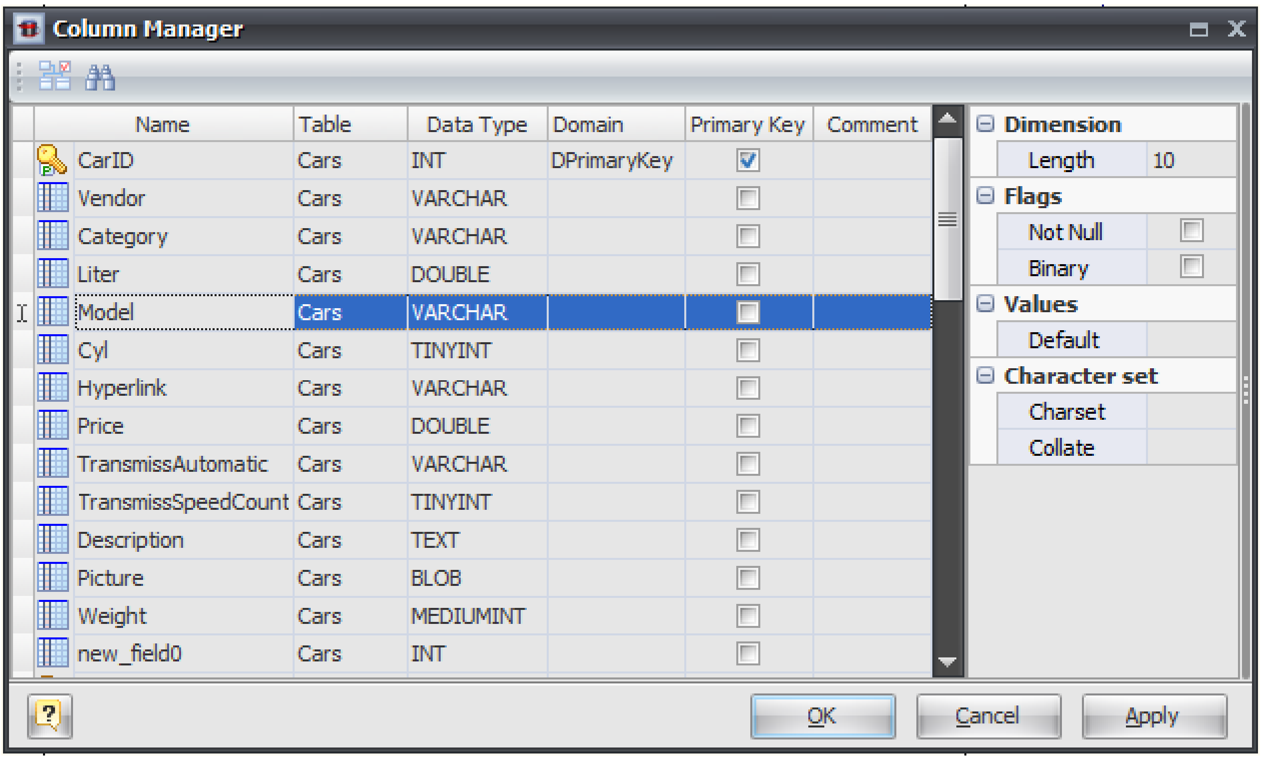
Delete
Remove the selected column from the list;

Up/Down
Move the selected column along the list.

See also:
Diagram Objects: Domains | Table Editor | Indexes | Triggers | Column Manager

7.3.2. Column Manager

The **Column Manager** allows you to view and modify the basic parameters of all the table columns within the diagram. To open the **Column Manager** use the **Column Manager** menu item on **Diagram** tab of Ribbon or press **Ctrl + 3**.



Name	Table	Data Type	Domain	Primary Key	Comment
CarID	Cars	INT	DPrimaryKey	☑	
Vendor	Cars	VARCHAR		□	
Category	Cars	VARCHAR		□	
Liter	Cars	DOUBLE		□	
Model	Cars	VARCHAR		□	
Cyl	Cars	TINYINT		□	
Hyperlink	Cars	VARCHAR		□	
Price	Cars	DOUBLE		□	
TransmissAutomatic	Cars	VARCHAR		□	
TransmissSpeedCount	Cars	TINYINT		□	
Description	Cars	TEXT		□	
Picture	Cars	BLOB		□	
Weight	Cars	MEDIUMINT		□	
new_field0	Cars	INT		□	

Dimension	
Length	10
Flags	
Not Null	□
Binary	□
Values	
Default	
Character set	
Charset	
Collate	

The **Column Manager** consists of the following areas:

- **Column Grid**
- **Properties Pane.**

Column Grid

The grid rows stand for the diagram columns, and the grid columns for the diagram column parameters. These parameters can be changed by using Column Editor for each table one by one, but with **Column Manager** you can do it much more quickly.

The grid allows you to modify the following column properties:

Name
The name of the column, which must be unique within the diagram.

Table
The name of the table which owns a column. This property can't be changed.

Data type
The data type of the column; it can be one of integer types, which differ in data size (TINYINT, SMALLINT, MEDIUMINT, INT, BIGINT), decimal type (FLOAT, DOUBLE, DECIMAL), a type for storing date and time values (DATE, DATETIME, TIMESTAMP, TIME, YEAR), string type (CHAR, VARCHAR), a large binary type (TINYBLOB, BLOB, MEDUIMBLOB, LONGBLOB), a multi-line text (TINYTEXT, TEXT, MEDIUMTEXT, LONGTEXT), enumeration type, which can contain only values from the Set attribute (ENUM), or a set of values from the Set attribute (SET).

Domain
You can specify a column domain, see the Domains topic for detailed information.

Primary key
Specify this option to include the field into the table primary key.

Comment
An arbitrary description for the column.

Properties Pane

The properties pane allows you to define the advanced properties of the column, selected in the **Column Grid**. Please refer to Column Editor topic to see the meaning of properties.

To save your changes click the **OK** button. If you want to store changes and continue editing, click on the **Apply** button.

Searching column in the list

If your diagram contains large number of table columns, it may be important to search for a column in a most easy way. To find a column quickly by its name, click the **Find** button (🔍) at the dialog toolbar. The **Find Name** dialog will appear. Type in the name of column to find and click **OK**. If the column is found the grid cursor positions on the appropriate row.

Running the Column Editor

If you want to change some parameters of the column parent (e.g. add or remove columns or indexes), select the appropriate column in the dialog and click the **Edit** button (▦) of the dialog toolbar. The Column Editor will appear, where you can edit all the parameters for the selected column and for its parent table.

See also:
Diagram Objects: Domains | Table Editor | Column Editor | Table Manager

7.4. Domains

A **domain** is a storage for some column attributes. Using domains you can create a column with all the required properties already defined (e.g. data type, length, precision, etc.), and modify a number of columns at once.

To create a domain and set all the domain properties use the Domain Manager. This tool allows you to add, modify, and remove domains.

After you have created diagram domains, it becomes much easier to add columns to the diagram tables - you can simply select the appropriate domain for the column within the Column Editor, and this assigns all domain attributes to the corresponding column attributes. It is also much more convenient to modify the domain-based columns than usual columns. Modifying any of the domain attribute updates all the columns, based on this domain.

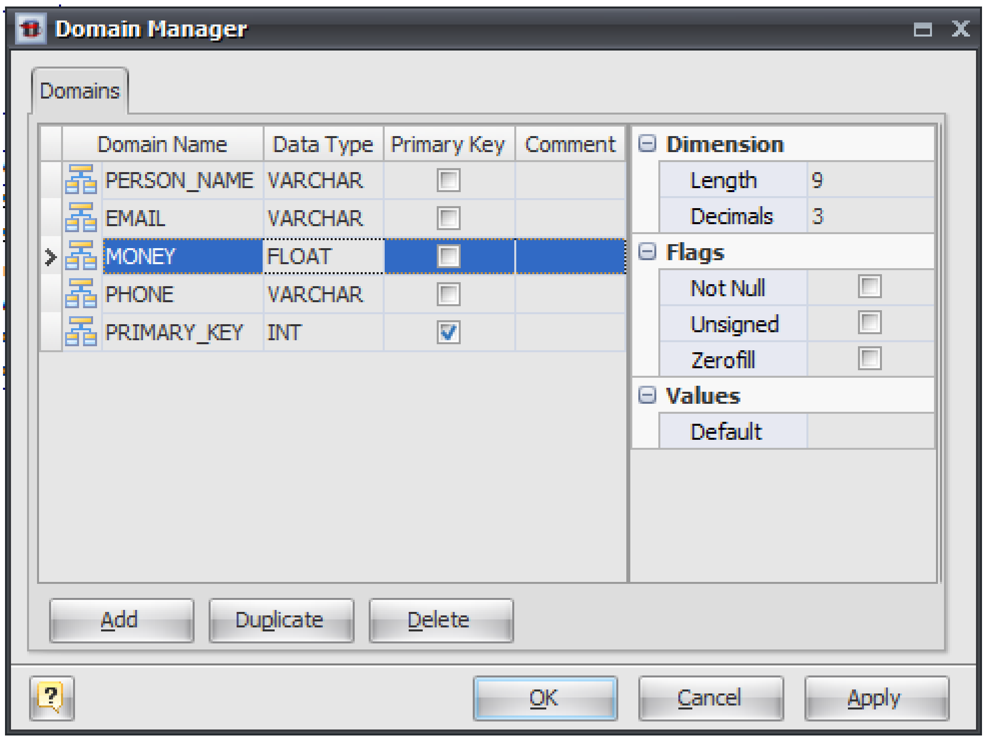
Deleting a domain does not delete the dependent columns. See the Domain Tutorial and the Domain Manager topics to learn more about working with diagram domains.

See also:
Diagram Objects: Column Editor | Domain Manager
Tutorials: Using Domains

7.4.1. Domain Manager

The **Domain Manager** is intended for managing diagram domains, which can be used for faster creating and modifying table columns.

To open the **Domain Manager** use the **Domain Manager** item on **Diagram** tab of Ribbon or press **Ctrl + 1**.



Domain Name	Data Type	Primary Key	Comment
PERSON_NAME	VARCHAR	□	
EMAIL	VARCHAR	□	
MONEY	FLOAT	□	
PHONE	VARCHAR	□	
PRIMARY_KEY	INT	☑	

Dimension	
Length	9
Decimals	3
Flags	
Not Null	□
Unsigned	□
Zerofill	□
Values	
Default	

The **Domain Manager** consists of the following areas:

- Domain List
- Properties Pane
- Button Pane.

Domain List

The domain list displays all the domains in the diagram and allows you to modify the following domain properties:

Domain name
The name of the domain, which must be unique within the diagram.

Data type
The type of the domain, which specifies data to store in the domain-based column; it can be one of integer types, which differ in data size (TINYINT, SMALLINT, MEDIUMINT, INT, BIGINT), decimal type (FLOAT, DOUBLE, DECIMAL), a type for storing date and time values (DATE, DATETIME, TIMESTAMP, TIME, YEAR), string type (CHAR, VARCHAR), a large binary type (TINYBLOB, BLOB, MEDUIMBLOB, LONGBLOB), a multi-line text (TINYTEXT, TEXT, MEDIUMTEXT, LONGTEXT), enumeration type, which can contain only values from the Set attribute (ENUM), or a set of values from the Set attribute (SET).

Primary key
Specify this option to include the domain-based columns into the table primary key.

Comment
An arbitrary description for the domain.

Properties Pane

The properties pane allows you to define the advanced properties of the domain, selected in the **Domain List**. The appearance of this pane changes according to the data type of the domain. These properties are:

Not null
This option indicates that a domain-based column value cannot be NULL;

Default
This attribute defines the default value, which the domain-column accepts if no other is specified;

Decimals
This attribute defines the number of digits, which follow the decimal point in the FLOAT, DOUBLE, and DECIMAL types.

Length
This attribute defines the maximum allowed length of the stored values; it applies to all integer, decimal, and string types.

Binary
This attribute indicates that the string value is stored in a binary form within the domain.

Unsigned
This option indicates that the stored integer or decimal value must be above zero.

Zerofill
With this attribute all unspecified digits of the domain-column value are filled as zeros according to the column size and precision; it applies to all integer and decimal types.

Unique
This option indicates that the domain-column value must be unique within the table, creating a unique index for the domain.

Autoinc
This attribute makes the domain-column value autoincrement, i.e. each new value is set automatically according to the previous value; it applies to all integer values.

Set
This attribute defines the list of permissible values for the domain of the ENUM or SET type.

Buttons Pane

The buttons under the list of domains allows you to perform the following actions:

Add
Add a new domain with the default properties to the end of the list;

Duplicate
Add a new domain with the same properties as the selected domain to the end of the list;

Delete

Remove the selected domain from the list.

See also:
Diagram Objects: Domains | Columns
Tutorials: Using Domains

7.5. References and Foreign Keys

Foreign key constraints are responsible for data referential integrity in your database. Simply put, referential integrity means that when a record in a table refers to a corresponding record in another table, that corresponding record must exist. So, a foreign key constraint specifies that the values in a column (or a group of columns) must match the values appearing in some row of another table.

A reference sets a foreign key constraint on the referencing table.

In order to create a foreign key in MySQL, you need the following:

- Both tables are to be InnoDB or Falcon tables
- The following syntax should be used: FOREIGN_KEY (fk_fieldname) REFERENCES table_name (fieldname)
- The field being declared a foreign key requires to be declared as an index in the table definition.

See also:
Diagram Objects: Creating a Reference | Creating a Many-to-Many Reference | Reference Editor | Reference Manager | Manual Reference Drawing

7.5.1. Creating a Reference

To create a reference between tables:

1. Click the **Reference** () item on the **Main** tab of Ribbon. Your mouse cursor will change its appearance.

2. Click on the table (referencing table or child table) that will have a foreign key.

3. Then click on the second table (referenced table or parent table) whose constraint (e.g. primary key) will be referenced by the new foreign key.

4. The **Joins** tab of the Reference Editor will be shown. You can choose columns of a referenced (parent) and referencing (child) table participated in the reference. The properties of auto column will be copied from the respective column of the referenced table.

Please note:

To create a self reference (that links columns at the same table), click the Reference () item on the Main tab of Ribbon. Then click the same table two times. To create N:M references, please refer to Creating a Many-to-Many Reference.

Please note:
By default **Database Designer for MySQL** draws references automatically using shortest line between table shapes on diagram with reference's label placed in the center of this line. Sometimes this can lead to references crossing, or reference's label can fit below table shape. Starting from version 2.0 you can route references manually to avoid this. Take a look at Manual Reference Drawing topic.

Reference Creation in Detail

On reference creation, **Database Designer for MySQL** performs the following actions:

1. Creates new column(s) in the referencing table, their parameters (name, data type) will be copied from the primary key constraint of the referenced table. If the referencing table already has column(s) analogous to the primary key(s) of the referenced table, this column(s) will be used as foreign-key column(s).

2. If there is no primary key or unique constraints in the referenced table, a standard primary key column will be created in the referenced table.

3. Creates foreign key constraint in the referencing table that refers to the referenced table primary key(s).

Please note:
Starting with MySQL 4.1.2 such foreign key constraints for references are created automatically by MySQL.So you can disable FK-indexes creation in "Default Database Options" tab of Diagram Properties dialog.

See the Diagram Display Preferences topic to find out more about default **References** options, actions, etc.

See the Notation topic to find out more about reference symbol on the diagram.

See also:
Diagram: Notation | "Default Database Options" tab | Diagram Properties
Diagram Objects: Reference Editor | Creating a Many-to-Many Reference

7.5.2. Creating a Many-to-Many Reference

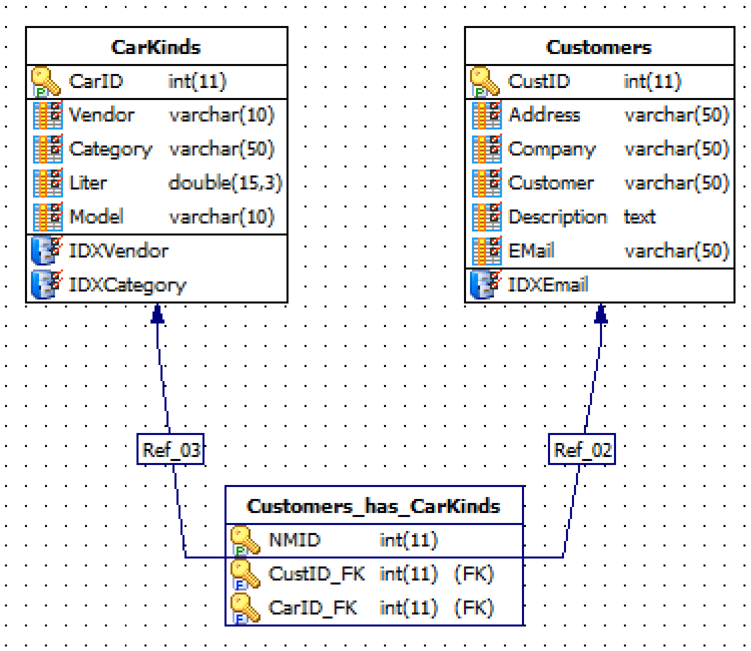
A *many-to-many* reference can relate one record in either table to many records in the other table.

The only way to create many-to-many (**N:M**) reference between two tables in MySQL is using an intermediate table, which records set association between primary keys of first table and primary keys of second table.

Database Designer for MySQL helps you set many-to-many reference between two tables easily:

- it automatically creates intermediate table with proper columns and indexes;
- and then creates regular references between columns of intermediate table and primary keys of tables being linked.

> **Warning:**
> Remember, that tables being linked must have primary key.



CarKinds

CarID	int(11)
Vendor	varchar(10)
Category	varchar(50)
Liter	double(15,3)
Model	varchar(10)
IDXVendor	
IDXCategory	

Customers

CustID	int(11)
Address	varchar(50)
Company	varchar(50)
Customer	varchar(50)
Description	text
EMail	varchar(50)
IDXEmail	

Customers_has_CarKinds

NMID	int(11)	
CustID_FK	int(11)	(FK)
CarID_FK	int(11)	(FK)

How to create...

To create a many-to-many reference between two tables, click the **N:M Reference** () item on the **Palette** tab of Ribbon. Your mouse cursor will change its appearance. Then consistently click two tables you want to link with many-to-many reference.

> **Please note:**
> By default **Database Designer for MySQL** draws references automatically using shortest line between table shapes on diagram with reference's label placed in the center of this line. Sometimes this can lead to references crossing, or reference's label can fit below table shape. Starting from version 2.0 you can route references manually to avoid this. Take a look at Manual Reference Drawing topic.

See also:
Diagram Objects: References and Foreign Keys | Creating a Reference

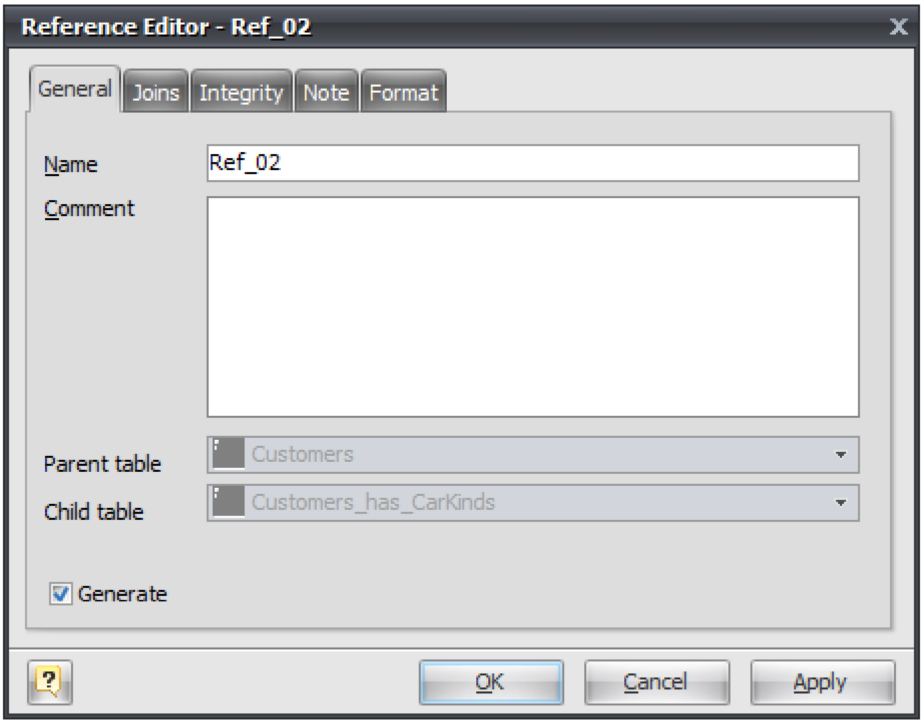
7.5.3. Reference Editor

Reference Editor is intended for editing the properties of a foreign key constraint, integrity rules, and choosing referenced columns.

To open the **Reference Editor**, simply double-click a reference on the diagram or select the **Properties** item from the reference context menu.

The dialog consists of several tabs, each of which is described in details below.

General



Reference Editor - Ref_02

Name: Ref_02

Comment:

Parent table: Customers

Child table: Customers_has_CarKinds

☑ Generate

This tab allows you to set up the basic reference properties.

Name
You can enter reference name, which will be displayed in the caption in the middle of the reference on the diagram. It is desirable, than name describe relationship role between tables.

Comment
The field to describe the reference role in full.

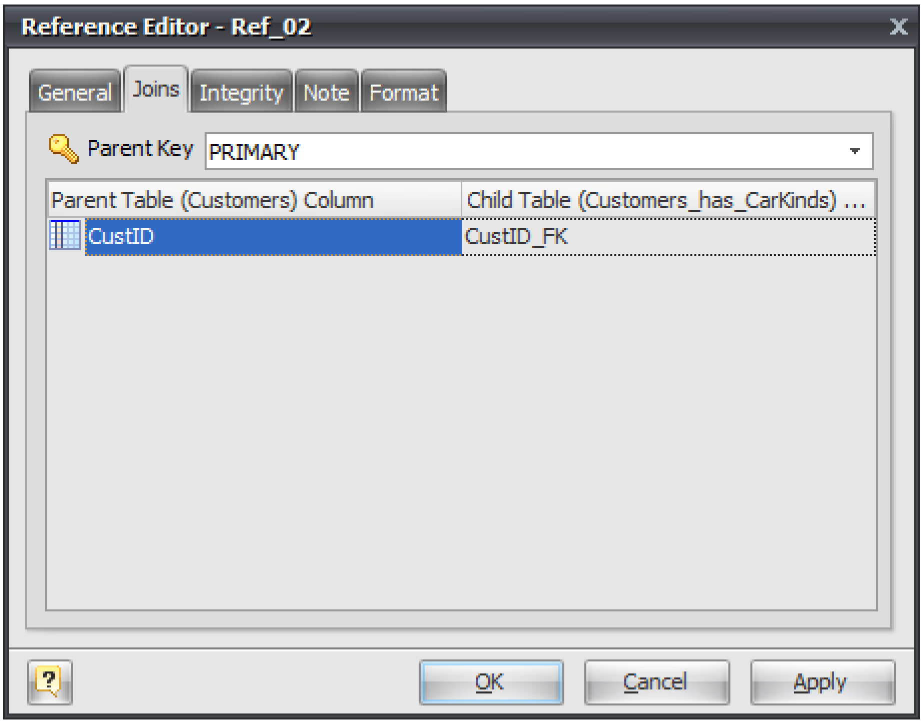
Parent table
Shows the referenced table's name.

Child table
Shows the referencing table's name.

Generate
Set this option off to exclude the table from the default selection of references in the Database Generation and Database Modification tools.

Joins

This tab allows to choose an active constraint of the referenced table and columns participated in the reference.



Parent Key: PRIMARY

Parent Table (Customers) Column	Child Table (Customers_has_CarKinds) ...
CustID	CustID_FK

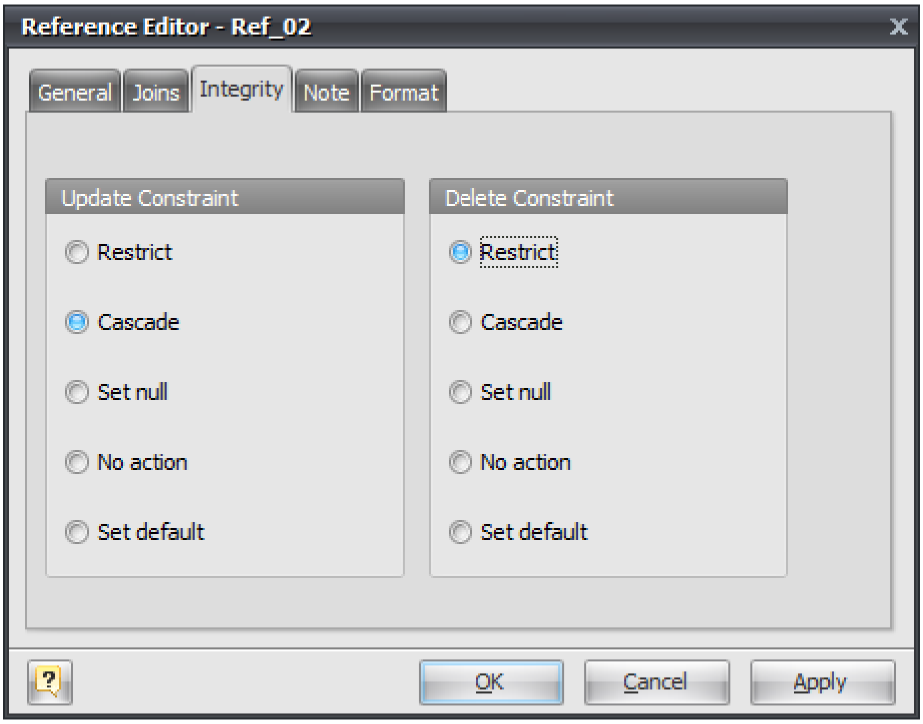
In the left side of the grid there is list of columns of *referenced* table. In the right side there is a list of corresponding foreign key columns of the *referencing* table.

To change assignment of foreign key column of the *referencing* table:

- Click on the **Child** table column cell, that corresponds to one of the columns of the referenced table.
- The drop-down menu with the list of the referencing table columns will appear.
- Then click on the required column of the referencing table.

Integrity

In this tab you can change integrity rules for foreign key constraint.



Update Constraint

○ Restrict
● Cascade
○ Set null
○ No action
○ Set default

Delete Constraint

● Restrict
○ Cascade
○ Set null
○ No action
○ Set default

When record in referenced table is changed (updated or deleted), it is possible to automatically modify associated records in the referencing table. This is the function of delete/update rules of the constraint.

For example, there are two tables: *Orders* and *Customers*. The *Orders* table has the foreign key column *CustID* which refer to the *Cusomers* table. You can delete a record from the *Customers* table and corresponding records from the *Orders* table, using one **DELETE** statement. It is possible because of cascade delete rule.

You can assign a rule both to update and delete event. In the left side there is list of update rules. In the right side there is list of delete rules. Click on the appropriate list item to change a rule.

Update constraint. Sets a rule that will be executed on update of a referenced table's record:

Restrict
Disallow update of the referenced table record if associated records in the child table exist.

Cascade
Update associated records in compliance with referenced table row update.

Set null
Set foreign key columns of associated records to null.

No acton
Do nothing with associated records.

Set default
Set foreign key columns of associated records to default column value. This default value can be set in the Table Columns Manager.

Delete constraint. Sets a rule that will be executed on delete of a referenced table's record:

Restrict
Disallow delete of referenced table record if associated records in the child table are exists.

Cascade
Delete all associated records.

Set null
Set foreign key columns of associated records to NULL.

No acton
Do nothing with associated records.

Set default
Set foreign key columns of associated records to default column value. This default value can be set in the Table Columns Manager.

Note

The **Note** tab allows you to define a description and an annotation for the edited reference. This properties will not affect the physical database, but they can be useful for your diagram development.

Format

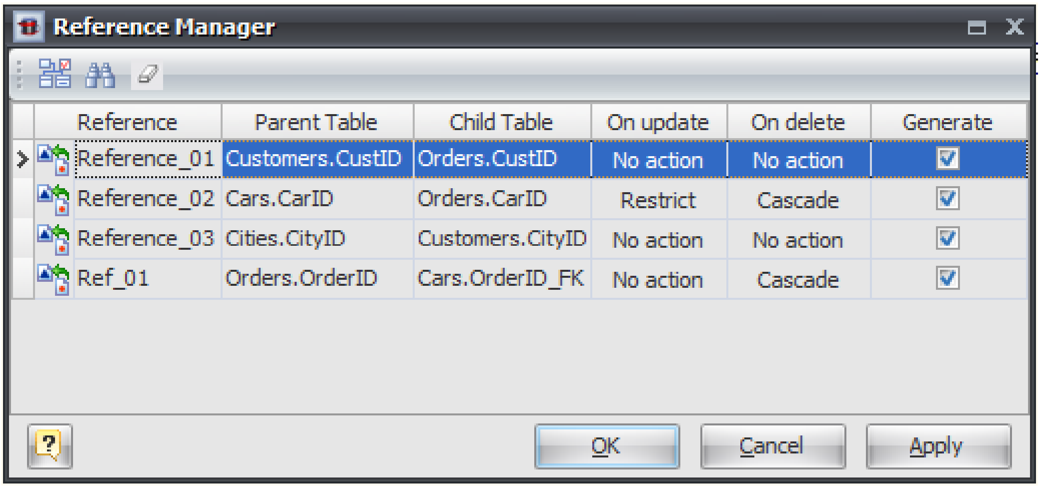
In this tab you can choose the colors which will be used for displaying edited reference on the diagram.

See also:
Diagram Objects: References and Foreign Keys | Creating a Reference | Column Editor | Column Manager
Database Functions: Database Generation | Database Modification

7.5.4. Reference Manager

Reference Manager allows you to view and modify basic parameters of table references within the diagram.

To open the Reference Manager use the **Reference Manager** () item on **Diagram** tab of Ribbon or press **Ctrl - 5**.



Reference	Parent Table	Child Table	On update	On delete	Generate
Reference_01	Customers.CustID	Orders.CustID	No action	No action	☑
Reference_02	Cars.CarID	Orders.CarID	Restrict	Cascade	☑
Reference_03	Cities.CityID	Customers.CityID	No action	No action	☑
Ref_01	Orders.OrderID	Cars.OrderID_FK	No action	Cascade	☑

The dialog contains a grid with the list of references that exist within the diagram. The grid shows reference names and their properties.

The grid allows you to modify the following properties:

Reference
The name of a reference;

On update
Sets a rule that will be executed on update of a referenced table's record;

On delete
Sets a rule that will be executed on delete of a referenced table's record;

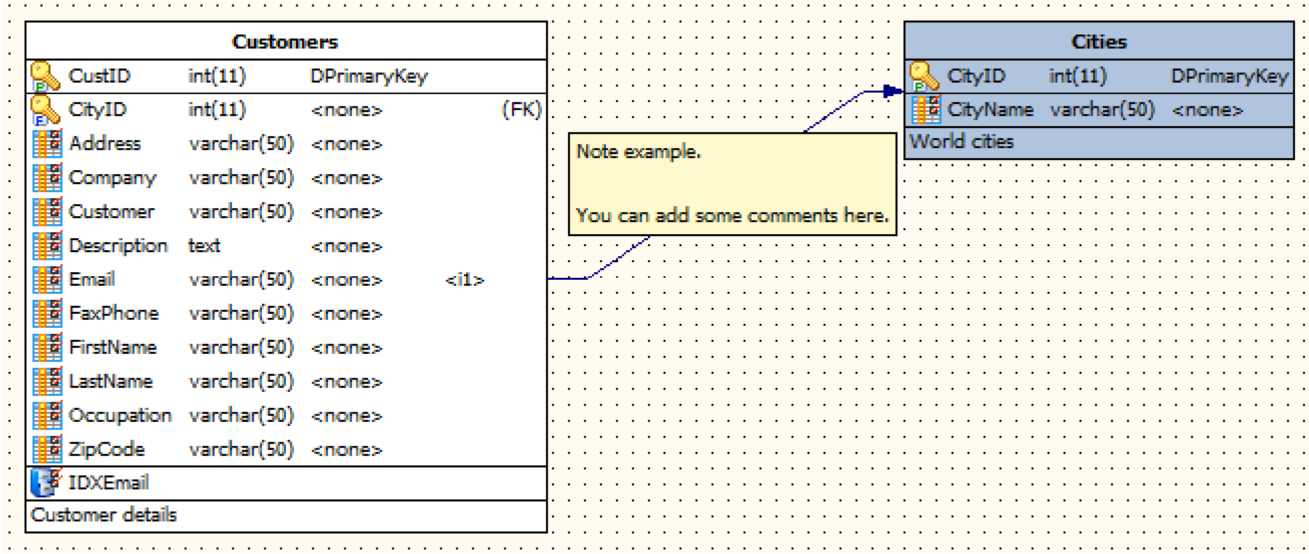
Generate
Includes the table to the default selection of references in the Database Generation and Database Modification tools.

The **Parent Table** and **Child Table** columns of the grid display relationships between columns of participated tables.

See also:
Diagram Objects: References and Foreign Keys | Creating a Reference

7.5.5. Manual Reference Drawing

By default **Database Designer for MySQL** routes references automatically using shortest line between two table shapes. Sometimes this is inconvenient because some other database objects can appear above reference or its label.



Customers			
CustID	int(11)	DPrimaryKey	
CityID	int(11)	<none>	(FK)
Address	varchar(50)	<none>	
Company	varchar(50)	<none>	
Customer	varchar(50)	<none>	
Description	text	<none>	
Email	varchar(50)	<none>	<i1>
FaxPhone	varchar(50)	<none>	
FirstName	varchar(50)	<none>	
LastName	varchar(50)	<none>	
Occupation	varchar(50)	<none>	
ZipCode	varchar(50)	<none>	
IDXEmail			
Customer details			

Cities		
CityID	int(11)	DPrimaryKey
CityName	varchar(50)	<none>
World cities		

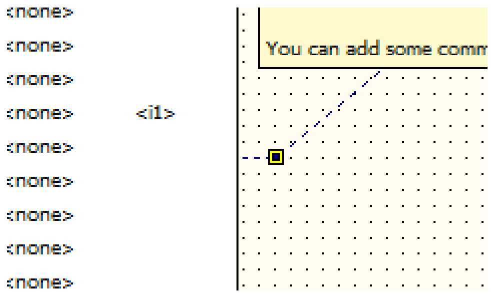
Note example.

You can add some comments here.

Starting with version 2.0 you can draw references manually, change label position, start and end tails position. You can add additional intermediate points to route your references in some other way.

Manual Start / End Tail Position

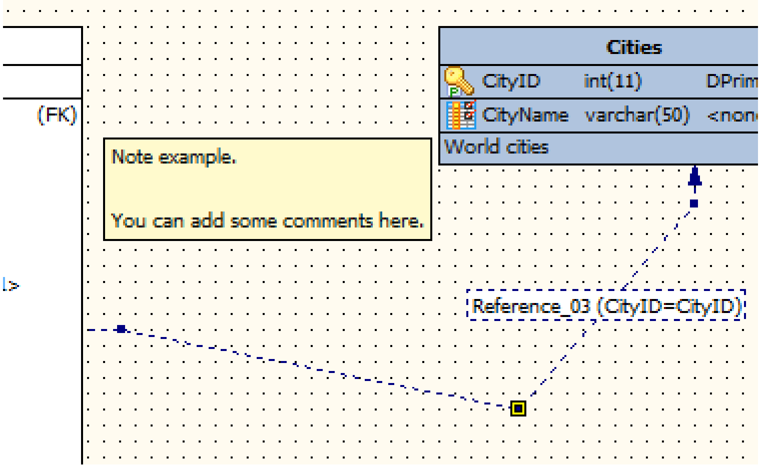
To allow changing of tails position use **Manual Start Tail Position** and **Manual End Tail Position** items from reference context menu (right mouse button click on reference). After enabling **Manual Start/End Tail Position** item small marker will appear at reference tail. You can click this marker by left mouse button and drag it to preferred location. Marker will be marked with black-yellow frame while moving.



Click **Manual Start/End Tail Position** item of context menu again to disable manual tail positioning and let **Database Designer for MySQL** calculate its position.

User Points

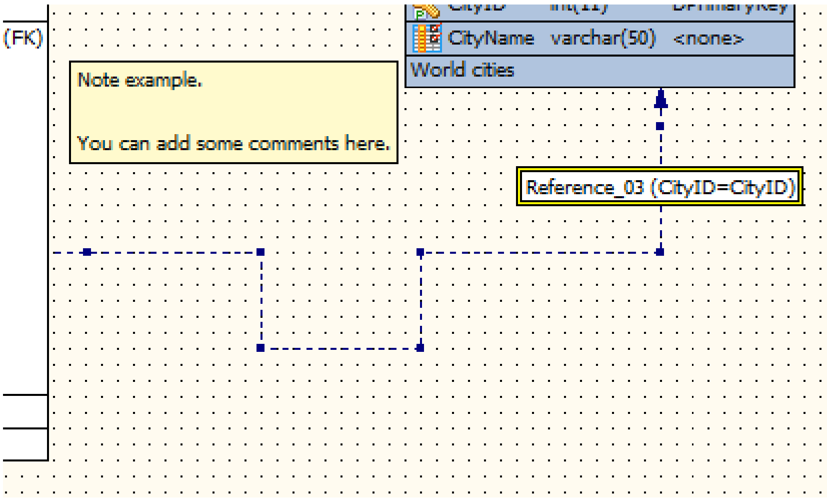
Use **Add Point** (➕) item of reference context-menu to add user point to reference. This points are displayed with small square markers when reference is selected. You can use left mouse button to drag such user point to any location you prefer. You can add such many user points as you need.


Cities
CityID int(11) DPrim
CityName varchar(50) <non
World cities
(FK)
Note example.
You can add some comments here.
Reference_03 (CityID=CityID)

You can delete user points by using **Delete Point** (✖) item of reference context-menu.

Reference Label Repositioning

Another new feature added in version 2.0 is an ability to move reference's label. Just click it by left mouse button and drag to any other location you prefer on reference. Label is marked with black-yellow frame while moving.


CityName varchar(50) <none>
World cities
(FK)
Note example.
You can add some comments here.
Reference_03 (CityID=CityID)

See also:
Diagram Objects: References and Foreign Keys | Creating a Reference

7.6. Indexes

In diagrams, you can create, edit, or delete table indexes, which gain fast access to specific information in a table. As a general rule, you should create an index on a table only if the data in the indexed columns will be queried frequently. Indexes take up disk space and slow the adding, deleting, and updating of rows. In most situations, the speed advantages of indexes for data retrieval greatly outweigh these disadvantages. However, if your application updates data very frequently or if you have disk space constraints, you might want to limit the number of indexes.

Before creating an index, you must determine what columns to use and what type of index to create.

You can create indexes based on a single column or on multiple columns in a database table. Multiple-column indexes enable you to distinguish between rows in which one column may have the same value. Indexes are also helpful if you often search or sort by two or more columns at a time. For example, if you often set criteria for last name and first name columns in the same query, it makes sense to create a multiple-column index on those two columns. To determine the usefulness of an index:

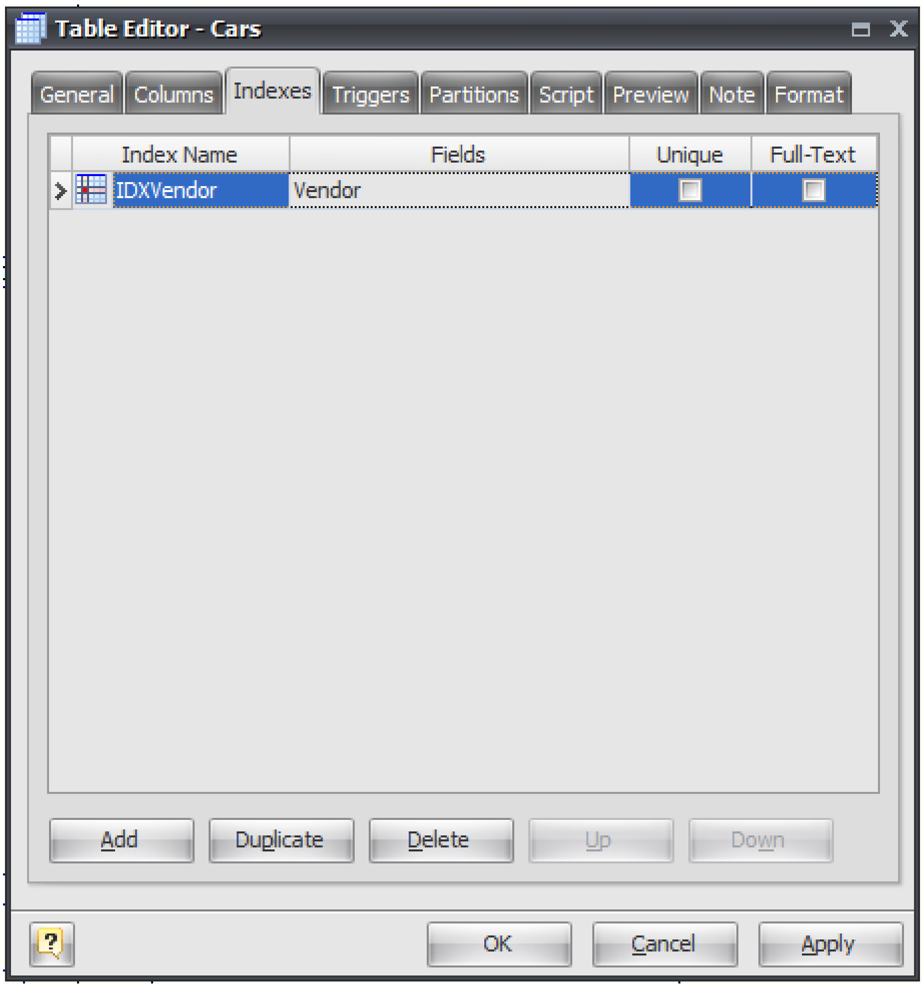
- Examine the WHERE and JOIN clauses of your queries. Each column included in either clause is a possible candidate for an index.
- Experiment with the new index to examine its effect on the performance of running queries.
- Consider the number of indexes already created on your table. It is best to avoid a large number of indexes on a single table.
- Examine the definitions of the indexes already created on your table. It is best to avoid overlapping indexes that contain shared columns.
- Examine the number of unique data values in a column and compare that number with the number of rows in the table. The result is the selectivity of that column, which can help you decide if a column is a candidate for an index and, if so, what type of index.

To modify table indexes, use Index Editor.

See also:
Diagram Objects: Index Editor | Index Manager

7.6.1. Index Editor

The **Index Editor** is placed within the Table Editor dialog. It allows you to modify the list of table indexes as well as index properties.



Index Name	Fields	Unique	Full-Text
IDXVendor	Vendor	□	□

The main element of the editor is the index list, which displays all indexes available within the table. The columns of the list allow you to modify the properties of the selected index.

These properties are:

Index Name
The name of the index, which must be unique within the table.

Unique
Defines the UNIQUE constraint for the selected columns. I.e. the combination of the included field values must be unique within the table.

Fields
The list of index attributes.

Full-text
This option can be applied to the index if all its fields are *CHAR*, *VARCHAR*, or *TEXT*; it allows you to perform relevance-based search within the included fields.

The buttons under the list of indexes allows you to perform the following actions:

Add
Add a new index with the default properties to the end of the list;

Duplicate
Add a new index with the same properties as the selected index to the end of the list;

Delete
Remove the selected index from the list;

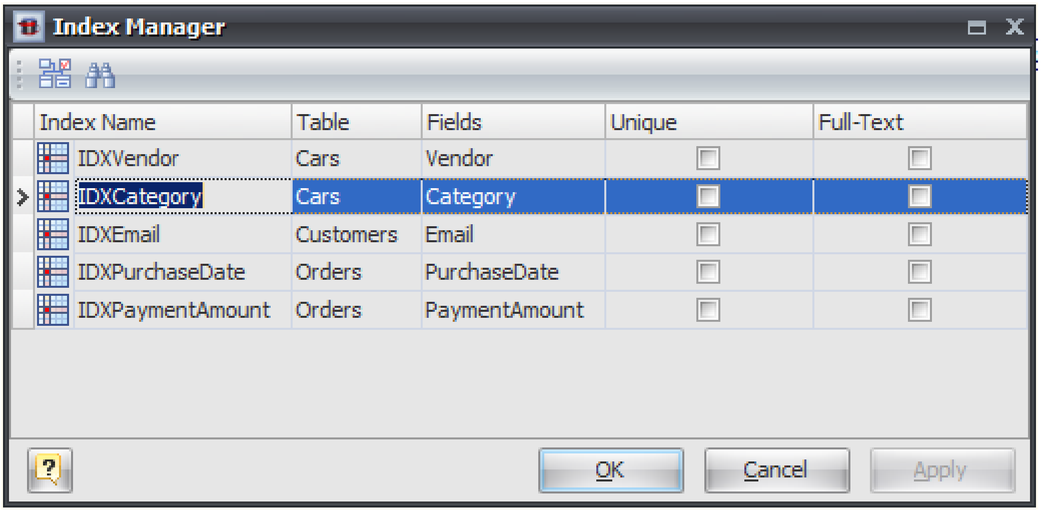
Up/Down
Move the selected index along the list.

See also:
Diagram Objects: Indexes | Index Manager

7.6.2. Index Manager

The **Index Manager** allows you to view and modify the basic parameters of all the table indexes within the diagram.

To open the **Index Manager** use the **Index Manager** item on **Diagram** tab of the Ribbon.



Index Name	Table	Fields	Unique	Full-Text
IDXVendor	Cars	Vendor	□	□
IDXCategory	Cars	Category	□	□
IDXEmail	Customers	Email	□	□
IDXPurchaseDate	Orders	PurchaseDate	□	□
IDXPaymentAmount	Orders	PaymentAmount	□	□

Modifying parameter values

This grid represents the basic parameters of all diagram indexes. The grid rows stand for the diagram indexes, and the grid columns for the index parameters. This parameters can be changed by using Index Editor for each table one by one, but with **Index Manager** you can do it much more quickly.

Click on the cell to modify the parameter value. Depending on the parameter type, the activated in-place editor can be a text box, a drop-down list, a check box, etc. Please refer to the Index Editor topic to find out more about all index parameters. Note, that the **Table** parameter indicates the name of the table a column belongs to, and cannot be changed.

To save your changes click the **OK** button. If you want to store changes and continue editing, click on the **Apply** button.

Searching index in the list

If your diagram contains large number of table indexes, it may be important to search for a index in a most easy way. To find a index quickly by its name, click the **Find** button () at the dialog toolbar. The **Find Name** dialog will appear. Type in the name of an index to find and click **OK**. If the index is found the grid cursor positions on the appropriate row.

Running the Index Editor

If you want to change some parameters of the index parent (e.g. add or remove columns or indexes), select the appropriate index in the dialog and click the **Edit** button () of the dialog toolbar. The Index Editor will appear, where you can edit all the parameters for the selected index and for its parent table.

See also:
Diagram Objects: Indexes | Index Editor

7.7. Stored Procedures and Functions

A stored procedure is a set of statements that can be stored in the server. Once this has been done, clients don't need to keep reissuing the individual statements but can refer to the stored procedure instead.

Database Designer for MySQL helps to build and manage stored procedures and functions in a powerful graphical environment.

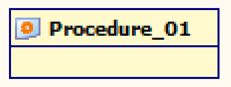
With **Database Designer for MySQL** you can:

- Create and Edit stored procedure or function;
- Manage all stored procedures and functions within the diagram.

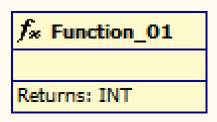
See also:
Diagram Objects: Creating a Stored Procedure or Function | Stored Routine Editor | Stored Routine Manager

7.7.1. Creating a Stored Procedure or Function

To create a new procedure (later on you can switch it to a stored function), click on the **Stored Procedure** () icon on the **Main** tab of the Ribbon. Your mouse cursor will change its appearance. Click on the diagram area to create a new stored procedure. A rectangle with the name of the stored procedure will appear in the diagram:



(Optional). To switch procedure to the function double click the routine rectangle. The Stored Routine Editor will be shown. Go to the **Definition** tab and select *Function* from the **Routine type** drop-down list. Click **OK** to save your changes.



To edit parameters of the stored procedures and functions use Stored Routine Editor.

See also:
Diagram Objects: Stored Procedures and Functions | Stored Routine Editor | Stored Routine Manager

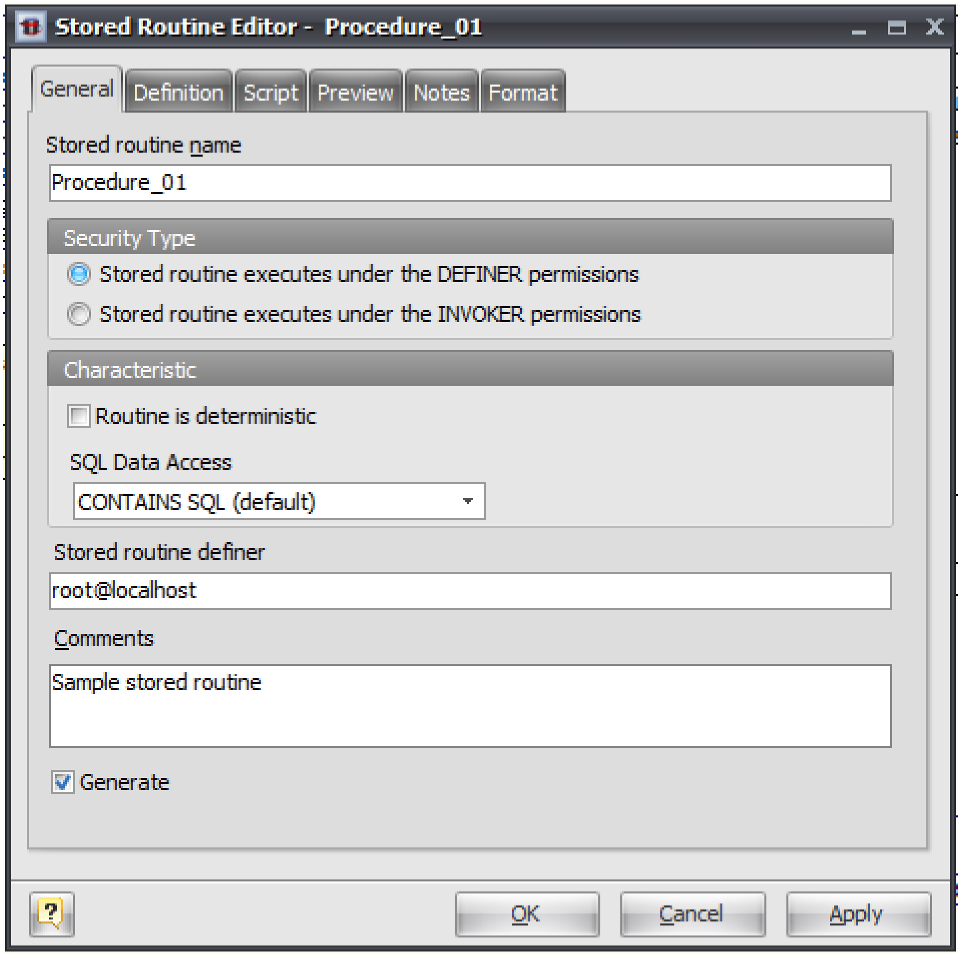
7.7.2. Stored Routine Editor

The **Stored Routine Editor** is provided for altering stored procedures/functions. All the parameters of stored procedure are valid for stored function, so you can switch between them easily.

To open the the **Stored Routine Editor**, simply double-click a stored procedure or function on the diagram or select the **Properties** item from the context menu.

The **Stored Routine Editor** consists of several tabs, each of which will be described below.

General



Stored routine name: Procedure_01

Security Type
- ● Stored routine executes under the DEFINER permissions
- ○ Stored routine executes under the INVOKER permissions

Characteristic
- □ Routine is deterministic
- SQL Data Access: CONTAINS SQL (default)

Stored routine definer: root@localhost

Comments: Sample stored routine

☑ Generate

This tab allows you to tune the basic properties of a stored procedure or function.

Stored Routine Name

The name of the stored routine must be unique within a schema. To check your diagram for the uniqueness of names use the Check Diagram tool.

Security Type

This option can be used to specify whether the routine should be executed using the permissions of the user who creates the routine or the user who invokes it. The default value is DEFINER. The creator or invoker must have permission to access the database with which the routine is associated.

Routine is deterministic

This option adds DETERMINISTIC clause to CREATE PROCEDURE statement. A procedure or function is considered “deterministic” if it always produces the same result for the same input parameters, and “not deterministic” otherwise.

SQL Data Access

This option provides information about the nature of data use by the routine. In MySQL, these characteristics are advisory only. The server does not use them to constrain what kinds of statements a routine will be allowed to execute.

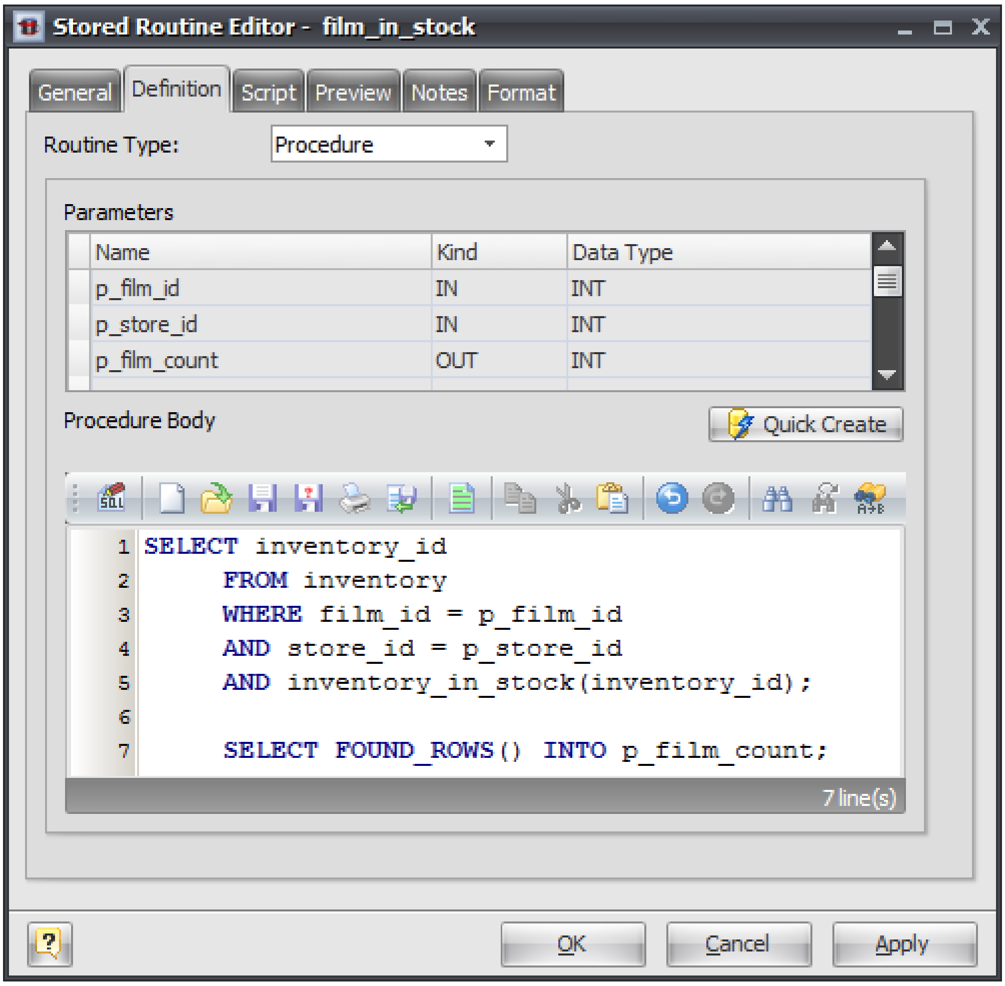
Stored routine definer

This optional parameter specifies the MySQL account to be used when checking access privileges at routine execution time for routines that have the DEFINER security type.

Comment
A native database comment for your routine.

Generate
Set this option off to exclude the routine from the default selection in the Database Generation and Database Modification tools.

Definition



Routine Type: Procedure

Parameters

Name	Kind	Data Type
p_film_id	IN	INT
p_store_id	IN	INT
p_film_count	OUT	INT

Procedure Body

```
SELECT inventory_id
     FROM inventory
     WHERE film_id = p_film_id
     AND store_id = p_store_id
     AND inventory_in_stock(inventory_id);

     SELECT FOUND_ROWS() INTO p_film_count;
```

7 line(s)

The **Definition** tab allows you to choose the routine type (stored procedure or function), determine the list of parameters, edit routine text.

Routine Type
This options helps you switch between stored procedure and function.

Parameters
This grid contains the list of all procedure parameters. To add a new parameter simply proceed to the first empty node and fill in the cells.

Returns
This option is available only if you choose Function in Routine Type drop-down list. It sets the type

of variable being returned by the function.

Routine Body
This text editor represents the routine text, which appears between BEGIN and END keywords in CREATE PROCEDURE statement.

Quick Create button
This button allows to execute CREATE PROCEDURE statement with current procedure body. This can be useful when you want to test procedure without closing editor.

Script

This tab allows you to set SQL statements, which will be executed before (use **Begin** tab) and after (use **End** tab) generation of the stored routine.

Preview

The **Preview** tab displays the SQL statement, which will be executed during the database generation. This statement is made up according to the changes you have made using previous tabs. Note, that the text within the editor is read-only.

Notes

The **Notes** tab allows you to define a description and an annotation for the edited stored routine. This properties will not affect the physical database, but they can be useful for your diagram development.

Format

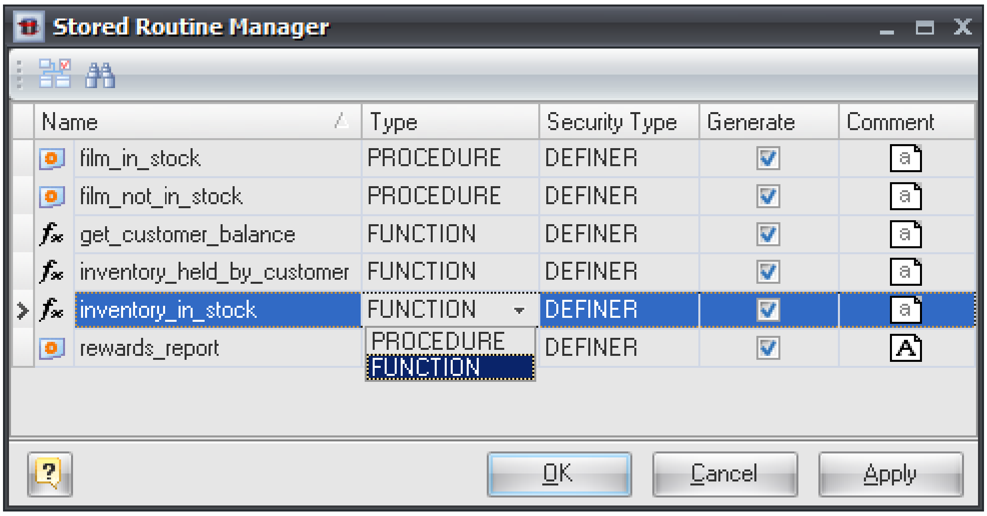
These options allow you to set the routine line and fill color for displaying on the diagram, different from the default colors.

See also:
Diagram Objects: Stored Procedures and Functions | Creating a Stored Procedure or Function | Stored Routine Manager

7.7.3. Stored Routine Manager

Stored Routine Manager allows you to view and modify the basic parameters of all stored procedures and functions within the diagram.

To open **Stored Routine Manager** use the **Stored Routine Manager** item on **Diagram** tab of the Ribbon or press **Ctrl + 6**.



Name	Type	Security Type	Generate	Comment
film_in_stock	PROCEDURE	DEFINER	☑	a
film_not_in_stock	PROCEDURE	DEFINER	☑	a
get_customer_balance	FUNCTION	DEFINER	☑	a
inventory_held_by_customer	FUNCTION	DEFINER	☑	a
inventory_in_stock	FUNCTION PROCEDURE FUNCTION	DEFINER	☑	a
rewards_report		DEFINER	☑	A

The main part of the dialog window is the stored routines grid. The grid rows stand for the stored procedures and functions, and the grid columns stand for their parameters.

These parameters can be changed by using Stored Routine Editor for each routine one by one, but with the manager you can do it much more quickly.

The grid allows you to modify the following properties:

Name
The name of the routine;

Security type
This option can be used to specify whether the routine should be executed using the permissions of the user who creates the routine or the user who invokes it. The default value is DEFINER. The creator or invoker must have permission to access the database with which the routine is associated;

Generate
Includes the routine to the default selection in the Database Generation and Database Modification tools;

Comment
An arbitrary description of the routine.

See also:
Diagram Objects: Stored Procedures and Functions | Creating a Stored Procedure or Function | Stored Routine Editor

7.8. Triggers

A trigger is a named database object that is associated with a table and that activates when a particular event occurs for the table. Activation of a trigger means execution of a set of commands attached to that trigger. For example, BEFORE trigger for INSERT statements could be used to check the values to be inserted into new rows.

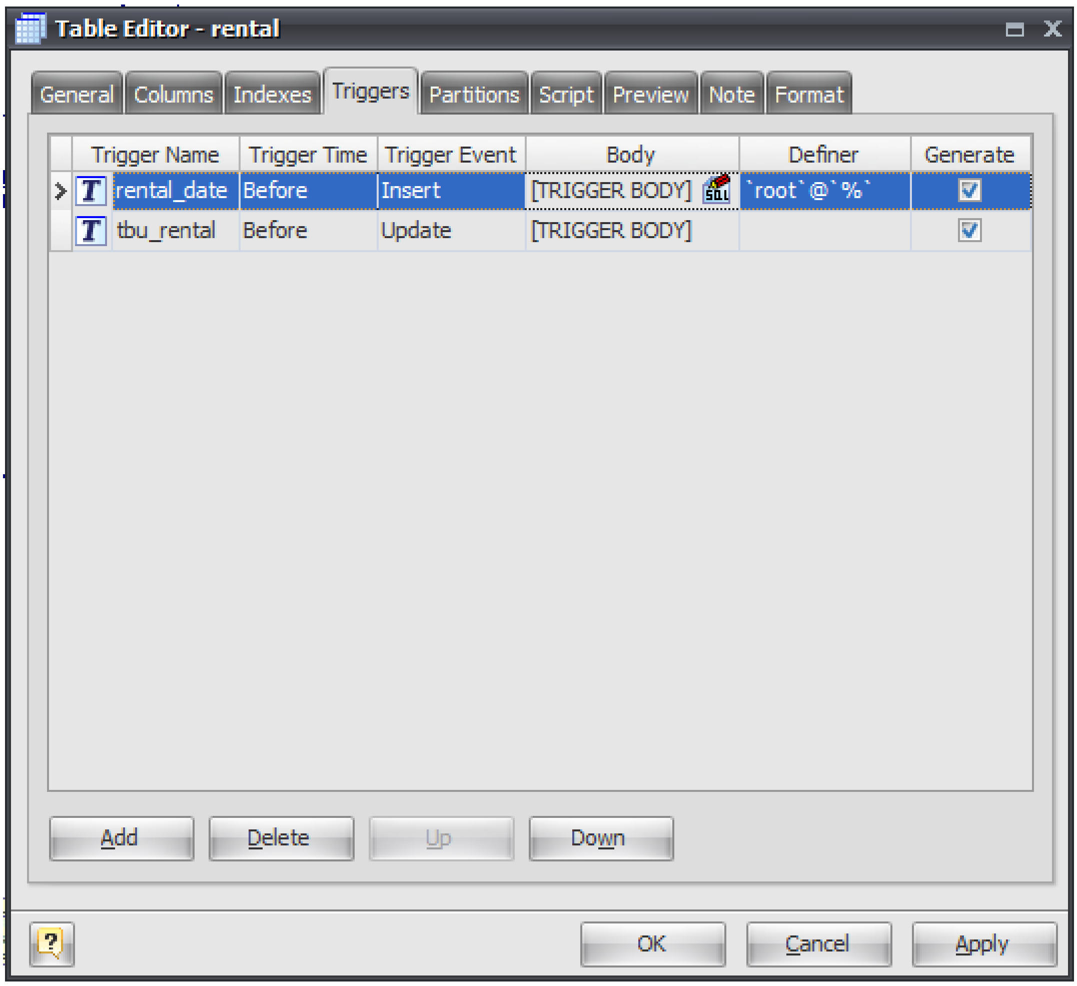
> **Please note:**
> You can read more about triggers in **Using Triggers** topic of MySQL Manual: http://dev.mysql.com/doc/refman/5.1/en/triggers.html

With **Database Designer for MySQL** you can edit table triggers.

See also:
Diagram Objects: Trigger Editor

7.8.1. Trigger Editor

Trigger Editor is placed within the Table Editor dialog on **Triggers** tab. It allows you to modify table triggers.



	Trigger Name	Trigger Time	Trigger Event	Body	Definer	Generate
>	rental_date	Before	Insert	[TRIGGER BODY]	`root`@`%`	☑
	tbu_rental	Before	Update	[TRIGGER BODY]		☑

The main element of the editor is the trigger list, which displays all the triggers in the table and their properties. These properties are as follows:

Trigger Name
The name of the trigger. It must be different from any other trigger name of the same table.

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

Trigger Event
Indicates the kind of statement that activates the trigger. It can be INSERT, UPDATE, or DELETE.

> **Please note:**
> In MySQL the combination of **Trigger Time** and **Trigger Event** must be unique within a table. I.e. there must be no triggers with the same Trigger Time and Trigger Event in the same table.

Body
Statements to execute when the trigger activates. Press SQL Editor (SQL) button to edit trigger body SQL-script.

Definer
The DEFINER clause specifies the MySQL account to be used when checking access privileges at trigger activation time. If a user value is given, it should be a MySQL account in *'user_name'@'host_name'* format (the same format used in the GRANT statement). The *user_name* and *host_name* values both are required. The definer can also be given as CURRENT_USER or CURRENT_USER(). The default DEFINER value is the user who executes the CREATE TRIGGER statement. (This is the same as DEFINER = CURRENT_USER).

Generate
Enables trigger creation during Database Generation and Database Modification.

The buttons under the list of triggers allow you to perform the following actions:

Add
Add a new trigger to the end of the list;

Delete
Remove the selected trigger from the list;

Up/Down
Move the selected trigger along the list.

See also:
Diagram Objects: Triggers | Table Editor

7.9. Views

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

You can think of a view as virtual table. It does not physically exist, but works like a real table. Usually it is created by a query joining one or more tables.

Please note:
You can read more about triggers in **Using Views** topic of MySQL Manual: http://dev.mysql.com/doc/refman/5.1/en/views.html

With **Database Designer for MySQL** you can:

Create view;

Edit a view;

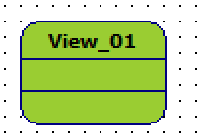
Manage all views within the diagram.

See also:
Diagram Objects: Creating a View | View Editor | View Manager

7.9.1. Creating a View

To create a new view please do the following:

1. Click on the Create View () icon on the Main tab of the Ribbon. Your mouse cursor will change its appearance. Click on the diagram area to create a new view. Or right click on the empty space of diagram and choose Create Object | View. A rectangle with the view name will appear on the diagram:

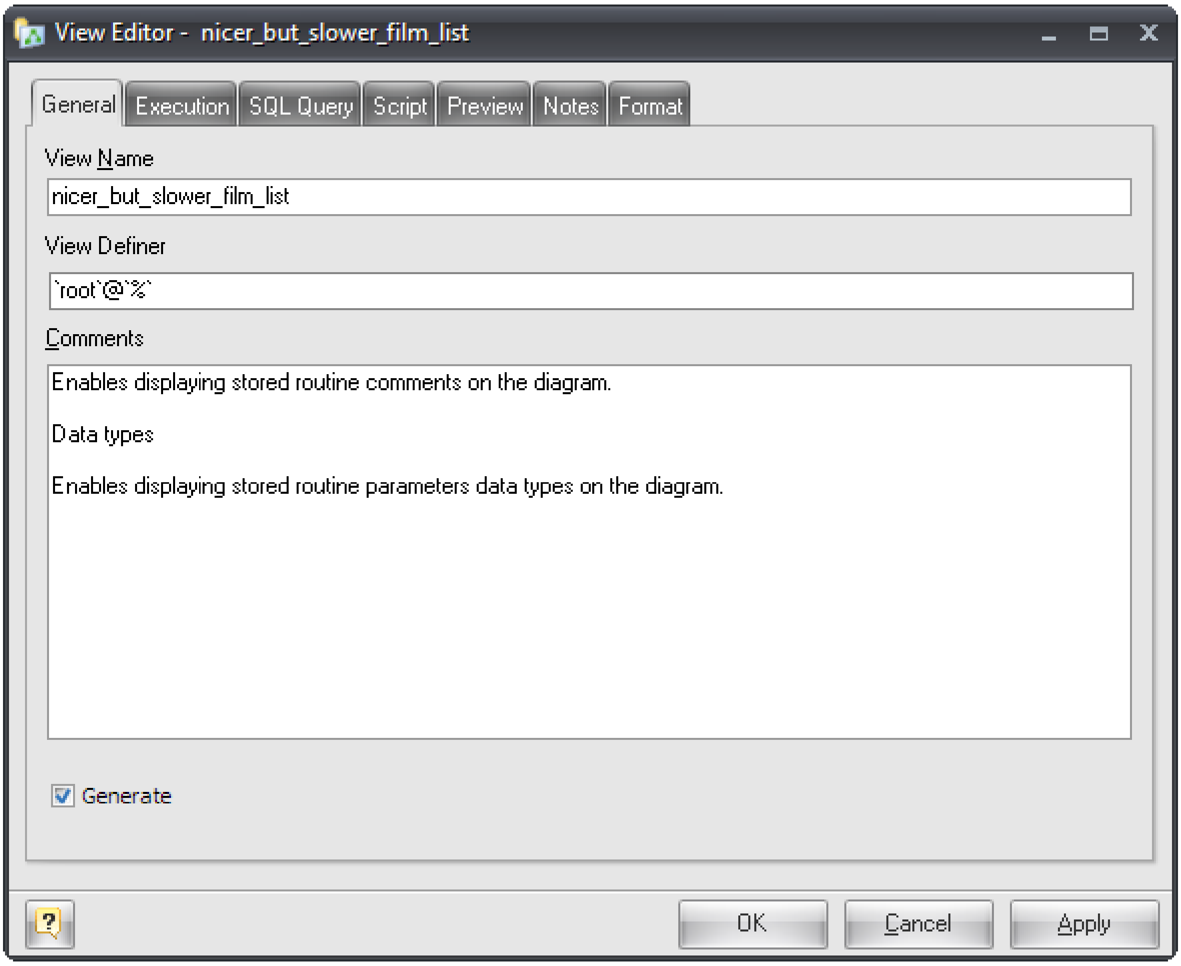


2. (optional) Double click on the new view symbol in the diagram to call the View Editor. You can set the name of the view and define its SQL query.

See also:
Diagram Objects: Views | View Editor

7.9.2. View Editor

The **View Editor** is provided for altering SQL views. To open the **View Editor**, simply double-click a view on the diagram or select the **Properties** item from the context menu.



View Name: nicer_but_slower_film_list

View Definer: `root`@`%`

Comments:

Enables displaying stored routine comments on the diagram.

Data types

Enables displaying stored routine parameters data types on the diagram.

☑ Generate

The **View Editor** contains several tabs, each of which will be described below.

General

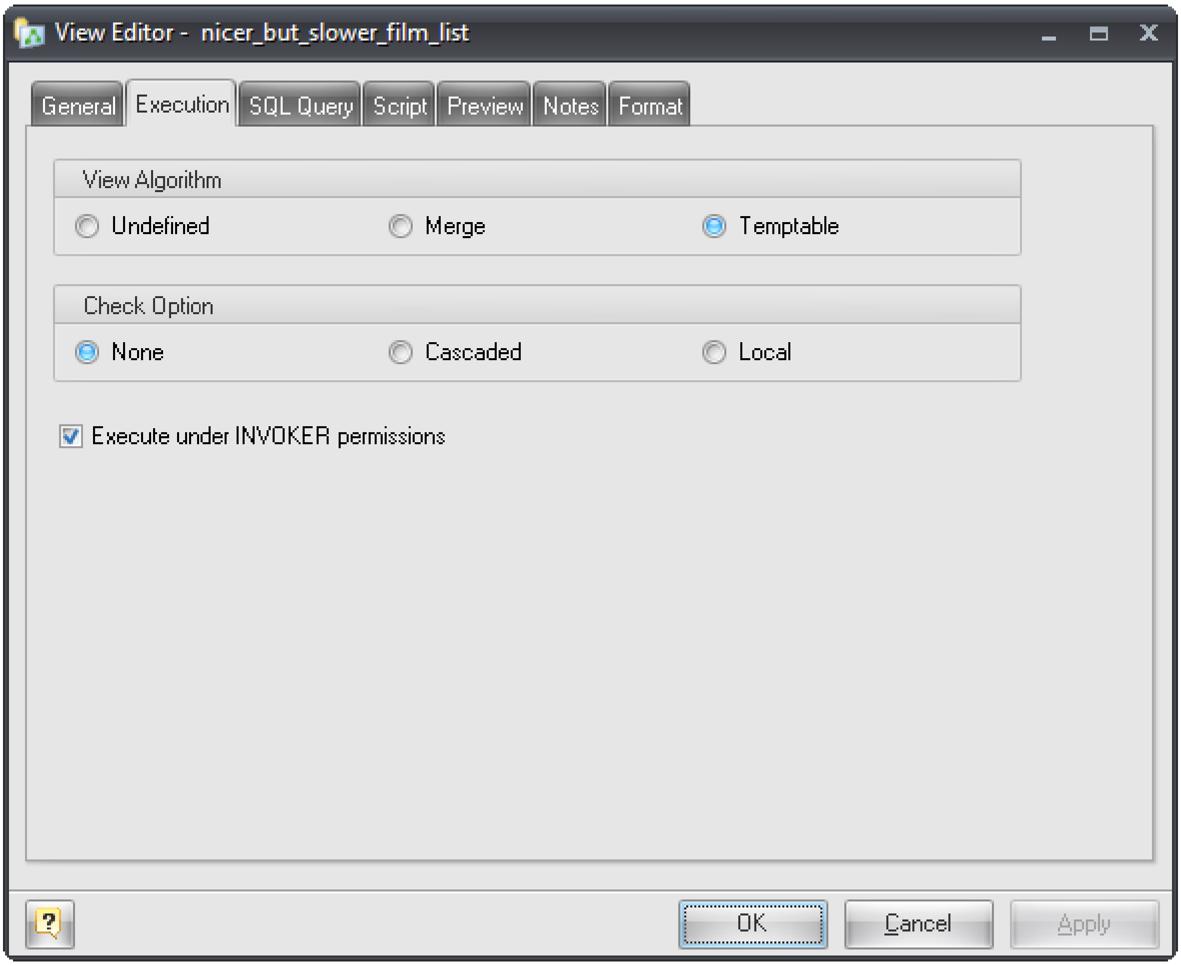
This tab allows you to set the name of the view and write comments for it. There are the following fields on the tab:

View Name
Sets the name of the view. To check your diagram for the uniqueness of names use the Check Diagram tool.

View Definer
The *DEFINER* clause determines which MySQL account to use when checking access privileges for the view when an SQL-statement is executed that references the view.

Comments
A native comment for the view.

Generate
Set this option off to exclude the view from the default selection in the Database Generation and Database Modification tools.



View Editor - nicer_but_slower_film_list

View Algorithm

○ Undefined ○ Merge ● Temptable

Check Option

● None ○ Cascaded ○ Local

☑ Execute under INVOKER permissions

Execution

The Execution tab provides control over options applied at view invocation time.

View Algorithm
It affects how MySQL processes the view. For *MERGE*, the text of a statement that refers to the view and the view definition are merged such that parts of the view definition replace corresponding parts of the statement. For *TEMPTABLE*, the results from the view are retrieved into a temporary table, which then is used to execute the statement. For *UNDEFINED*, MySQL chooses which algorithm to use. It prefers *MERGE* over *TEMPTABLE* if possible, because *MERGE* is usually more efficient and because a view cannot be updatable if a temporary table is used.

Check Option
For a view to be updatable, there must be a one-to-one relationship between the rows in the view and the rows in the underlying table. There are also certain other constructs that make a view nonupdatable. The *CHECK OPTION* clause can be given for an updatable view to prevent inserts or updates to rows except those for which the WHERE clause in the select_statement is true. In a *CHECK OPTION* clause for an updatable view, the *LOCAL* and *CASCADED* determine the scope of check testing when the view is defined in terms of another view. The *LOCAL* restricts the *CHECK OPTION* only to the view being defined. *CASCADED* causes the checks for underlying views to be evaluated as well. When neither keyword is given, the default is CASCADED.

Execute under INVOKER permissions
This option specifies the security context to be used when checking access privileges at view invocation time.

SQL Query

This tab allows to set the *SELECT* statement that provides the definition of the view.

A view can be created from many kinds of *SELECT* statements. For example, the *SELECT* can refer to a single table, a join of multiple tables, or a *UNION*. The *SELECT* need not even refer to any tables.

An example of *SELECT* statement, which could be used here:

```
SELECT product_id, product_name FROM Products
```

Script

This tab allows you to set SQL statements, which will be executed before (use **Begin** tab) and after (use **End** tab) generation of the view.

Preview

The **Preview** tab displays the SQL statement, which will be executed during the database generation. This statement is made up according to the changes you have made using previous tabs. Note, that the text within the editor is read-only. The content of this tab updates only then your press **Apply** button.

Notes

The **Notes** tab allows you to define a description and an annotation for the edited view. This properties will not affect the physical database, but they can be useful for your diagram development.

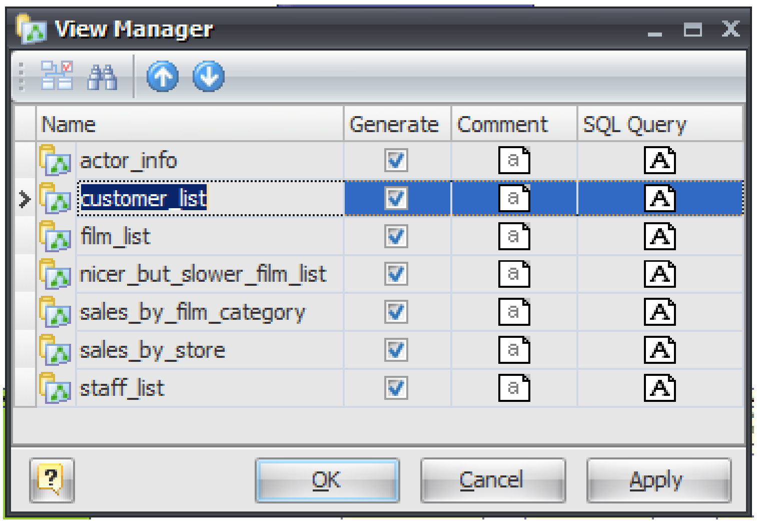
Format

These options allow you to set the view symbol line and fill color for displaying on the diagram, different from the default colors.

See also:
Diagram Objects: Views | Creating a View | View Manager

7.9.3. View Manager

The **View Manager** allows you to view and modify the basic parameters of all SQL views within the diagram. To open the **View Manager** use the **View Manager** ( ) item on **Diagram** tab of the Ribbon or press **Ctrl + 7**.



Name	Generate	Comment	SQL Query
actor_info	☑	a	A
customer_list	☑	a	A
film_list	☑	a	A
nicer_but_slower_film_list	☑	a	A
sales_by_film_category	☑	a	A
sales_by_store	☑	a	A
staff_list	☑	a	A

The main part of the dialog is the view grid. The grid rows stand for the views, and the grid columns for their parameters.

These parameters can be changed by using View Editor for each view one by one, but with the manager you can do it much more quickly.

The grid allows you to modify the following properties:

Name
The name of the view.

Generate
Includes the view to the default selection in the Database Generation and Database Modification tools.

Comment
An arbitrary description for the view.

SQL Query
SELECT statement that provides the definition of the view.

See also:
Diagram Objects: Views | Creating a View | View Editor

7.10. Events

MySQL Events are tasks that run according to a schedule. Therefore, sometimes they are referred as *scheduled* events. When you create an event, you are creating a named database object containing one or more SQL statements to be executed at one or more regular intervals, beginning and ending at a specific date and time. Conceptually, this is similar to the idea of the Unix **crontab** (also known as a *"cron job"*) or the **Windows Task Scheduler**.

> **Please note:**

> You can read more about MySQL events in **Using the Event Scheduler** topic of MySQL Manual: http://dev.mysql.com/doc/refman/5.1/en/events.html

With **Database Designer for MySQL** you can:

Create event;

Edit event;

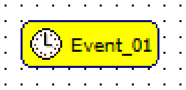
Manage all events within the diagram.

See also:
Diagram Objects: Creating an Event | Event Editor | Event Manager

7.10.1. Creating an Event

To create a new event please do the following:

1. Click on the Create Event (🕒) icon on the Main tab of the Ribbon. Your mouse cursor will change its appearance. Click on the diagram area to create a new event. Or right click on the empty space of diagram and choose Create Object | Event. A rectangle with the event name will appear on the diagram:



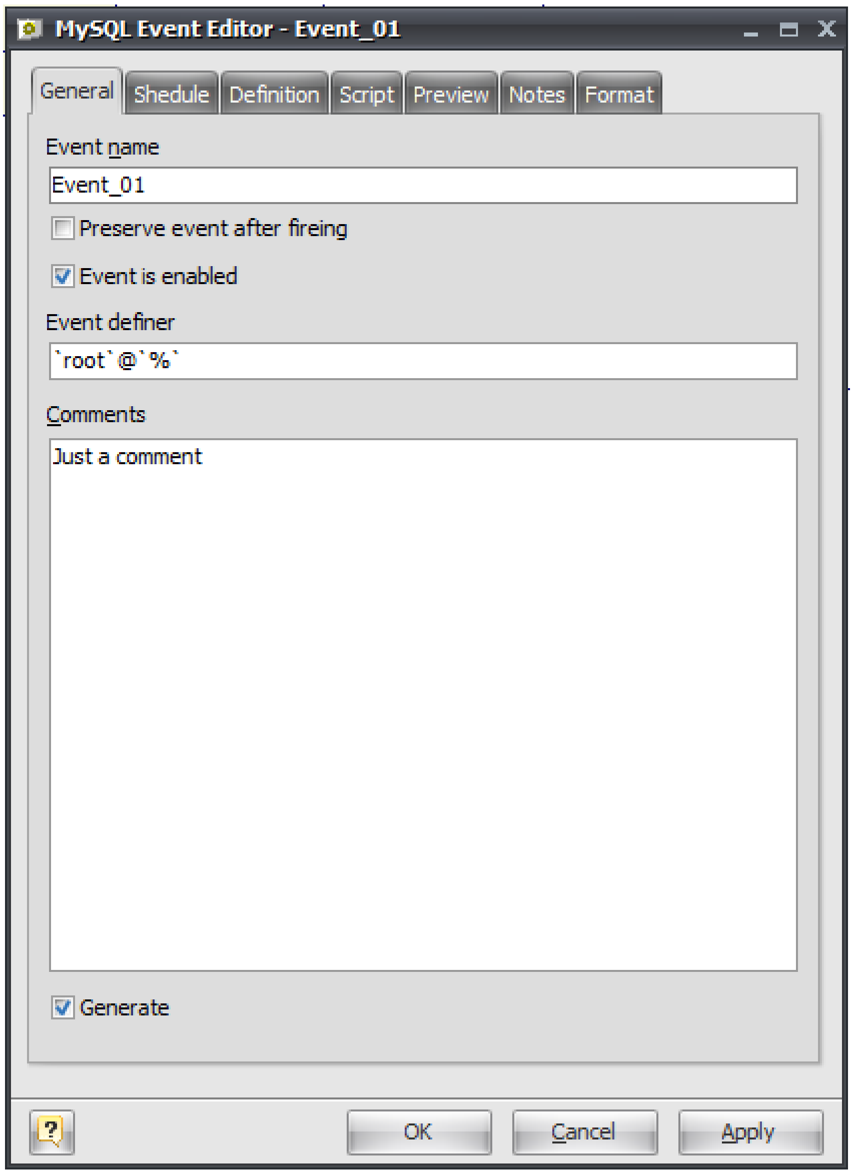
2. (optional) Double click on the new event symbol in the diagram to call the Event Editor. You can set the name of the event and define its schedule and SQL code.

See also:
Diagram Objects: Events | Event Editor | Event Manager

7.10.2. Event Editor

The **MySQL Event Editor** is provided for altering SQL views. To open the **MySQL Event Editor**, simply double-click an event on the diagram or select the **Properties** item from the context menu. The **MySQL Event Editor** contains several tabs, each of which will be described below.

General



Event name
Event_01

□ Preserve event after fireing

☑ Event is enabled

Event definer
`root`@`%`

Comments
Just a comment

☑ Generate

This tab allows you to set the name of the event and write comments for it. There are the following fields on the tab:

Event Name
Sets the name of the event. To check your diagram for the uniqueness of names use the Check Diagram tool.

Preserve event after fireing
Normally, once an event has expired, it is immediately dropped. You can override this behavior by checking this option. This will add ON COMPLETION PRESERVE clause to CREATE EVENT statement.

Event is enabled
You can create an event but keep it from being active turning off this option. The DISABLE keyword will be added to CREATE EVENT statement in this case.

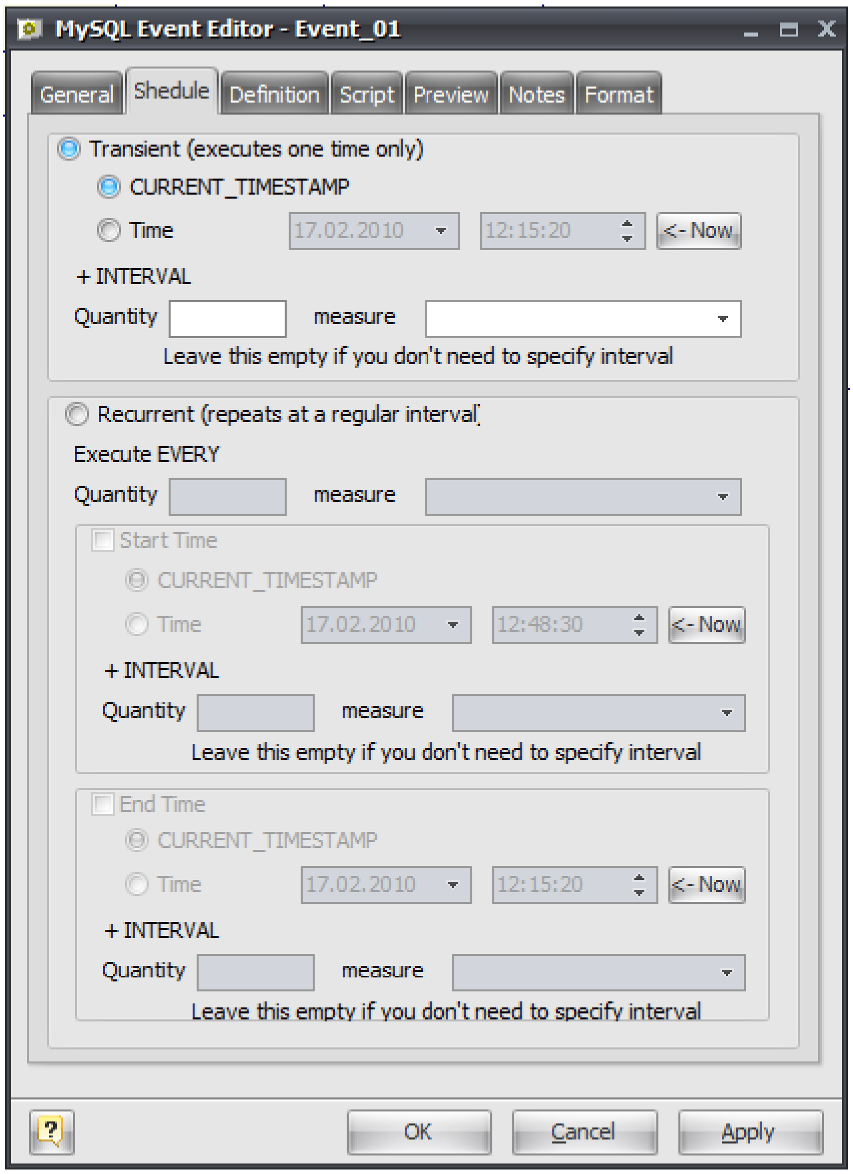
Event Definer

The DEFINER clause specifies the MySQL account to be used when checking access privileges at event execution time.

Comments
A native comment for the event.

Generate
Set this option off to exclude the event from the default selection in the Database Generation and Database Modification tools.

Schedule



MySQL Event Editor - Event_01

General | Shedule | Definition | Script | Preview | Notes | Format

● Transient (executes one time only)

● CURRENT_TIMESTAMP

○ Time 17.02.2010 12:15:20 <- Now

+ INTERVAL

Quantity measure

Leave this empty if you don't need to specify interval

○ Recurrent (repeats at a regular interval)

Execute EVERY

Quantity measure

□ Start Time

○ CURRENT_TIMESTAMP

○ Time 17.02.2010 12:48:30 <- Now

+ INTERVAL

Quantity measure

Leave this empty if you don't need to specify interval

□ End Time

○ CURRENT_TIMESTAMP

○ Time 17.02.2010 12:15:20 <- Now

+ INTERVAL

Quantity measure

Leave this empty if you don't need to specify interval

This tab allows you to set up event schedule. There are two types of events in MySQL:

Transient event
Event that executed one time only at specified date and time. You can set up exact execution date and time or use CURRENT_TIMESTAMP with some interval added.

Recurrent event
Event that executed periodically according to its schedule. You have to set up execution interval (Execute EVERY) for such events. You can also specify **Start Time** or **End Time** for such events.

> Please refer **CREATE EVENT Syntax** topic of MySQL Manual for more details about event schedule options: http://dev.mysql.com/doc/refman/5.1/en/create-event.html

Definition

This tab allows to set the SQL statement(s) that provides the body of event. This SQL code will be executed at specified time.

Quick Create button
This button allows to execute CREATE EVENT statement with current event body. This can be useful when you want to test event creation without closing editor.

Script

This tab allows you to set SQL statements, which will be executed before (use **Begin** tab) and after (use **End** tab) generation of the event.

Preview

The **Preview** tab displays the SQL statement, which will be executed during the database generation. This statement is made up according to the changes you have made using previous tabs. Note, that the text within the editor is read-only. The content of this tab updates only when your press **Apply** button.

Notes

The **Notes** tab allows you to define a description and an annotation for the edited event. This properties will not affect the physical database, but they can be useful for your diagram development.

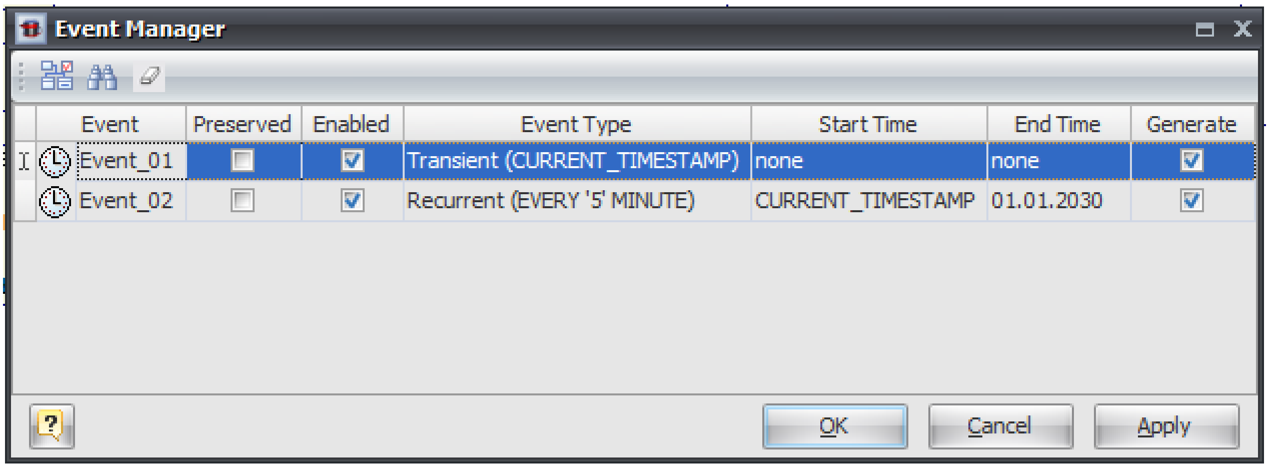
Format

These options allow you to set the event symbol line and fill color for displaying on the diagram, different from the default colors.

See also:
Diagram Objects: Events | Creating an Event | Event Manager

7.10.3. Event Manager

The **Event Manager** allows you to view and modify the basic parameters of all events within the diagram. To open the **Event Manager** use the **Event Manager** (🕒) item on **Diagram** tab of the Ribbon or press **Ctrl + 8**.



Event	Preserved	Enabled	Event Type	Start Time	End Time	Generate
Event_01	□	☑	Transient (CURRENT_TIMESTAMP)	none	none	☑
Event_02	□	☑	Recurrent (EVERY '5' MINUTE)	CURRENT_TIMESTAMP	01.01.2030	☑

The main part of the dialog is the event grid. The grid rows stand for the events, and the grid columns for their parameters.

These parameters can be changed by using Event Editor for each view one by one, but with the manager you can do it much more quickly.

The grid allows you to modify the following properties:

Event
The name of the event.

Preserved
Suppresses event deletion after event become expired.

Enabled
Enables or disables event right after creation.

Event Type
Displays event type (Transient or Recurrent). Read-only.

Start Time and End Time
Displays start and end execution times for recurrent events. Read-only.

Generate
Includes the event to the default selection in the Database Generation and Database Modification tools.

Searching an event in the list

If your diagram contains large number of events, it may be important to search for an event in a most easy way. To find an event quickly by its name, click the **Find** button (🔍) at the dialog toolbar. The **Find Name** dialog will appear. Type in the name of an event to find and click **OK**. If the event is found the grid cursor positions on the appropriate row.

Running the Event Editor

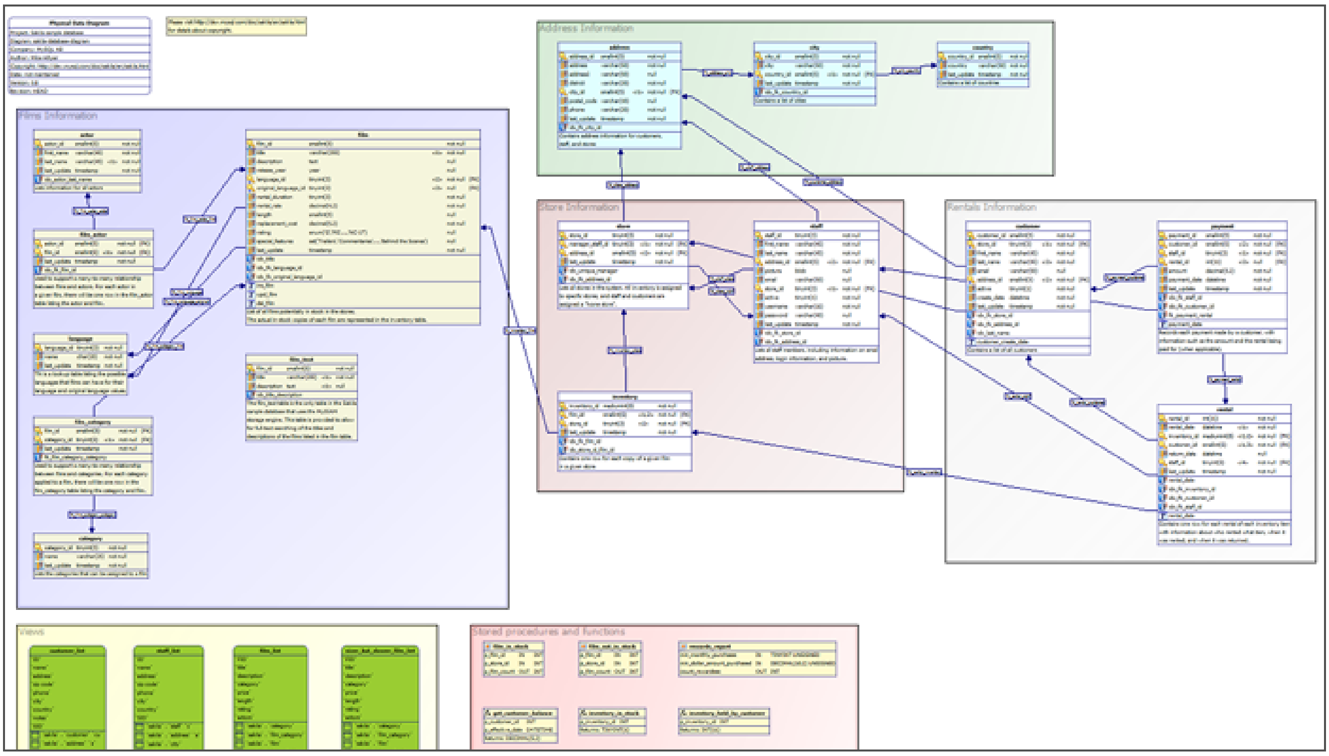
If you want to change some parameters of the event, select the appropriate event in the dialog and click the **Edit** button ( ) of the dialog toolbar. The Event Editor will appear, where you can edit all the parameters for the selected event.

See also:
Diagram Objects: Events | Creating an Event | Event Editor

7.11. Object Groups

Object Groups are the way to add more organization to your diagram. They allow to group several diagram objects to single **Object Group**. All objects in this group will be moved with group. You can also delete whole group with or without all its objects.

Here is an example of **Object Groups** usage. This is a sample diagram of "Sakila Database" available at http://dev.mysql.com/doc/sakila/en/sakila.html



> **Please note:**
> The "Sakila Database" diagram is placed in the sakila-database-diagram.mdd file shipped with the program.

There are 6 **Object Groups** created within this diagram to join tables and other objects into logical groups:

Films information
7 tables with different kind of information about films.

Address information
3 tables with information about stores and customers addresses.

Store information
3 tables with information about stores and staff.

Rentals information
3 tables with customers, rental orders and payment information.

Views
7 views with different kinds of queries.

Stored procedures and functions
6 stored procedures and functions to perform different actions with database information.

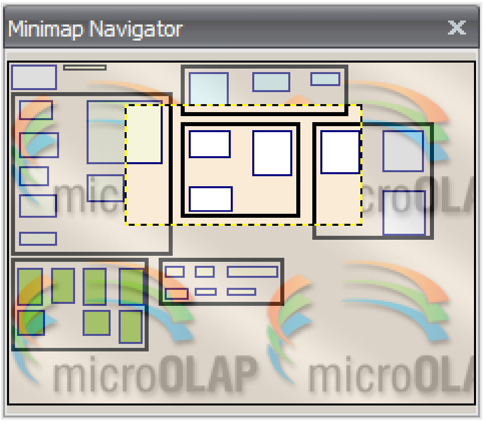
> **Please note:**
> **Object Groups** do not affect physical database structure. It is only a way to organize diagram object. **Object Groups** are not used when generating database structure.

Object Groups are displayed as semi-transparent rectangles on diagram. Group's name is displayed in left-upper corner of semi-transparent rectangle. You can adjust **Object Group** transparency and color with Object Group Editor.

Font size of **Object Group** name is independent on current diagram scale. So you can see **Object Group** name even when you use small scales (for example **33%**) to observe or move a lot of diagram objects simultaneously.

Object Group size rectangle on diagram is adjusted automatically to fit all objects attached to this group. You can also change **Object Group** size using resize markers, available when group is selected. You can also move group (with all its diagram objects) like any other object. And you can move diagram objects inside **Object Group** without any restrictions.

Object Groups are also displayed as bold frames on Minimap Navigator.



With **Database Designer for MySQL** you can:

Create Object Groups;

Edit Object Groups;

Add and Delete diagram objects to/from Groups;

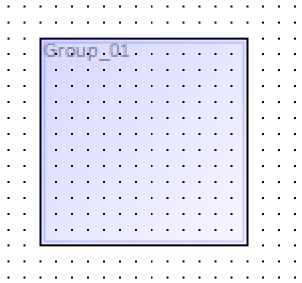
Delete Object Groups with or without containing objects.

See also:
Interface User Guide: Minimap Navigator
Diagram: Selecting and Moving Objects | Zooming a Diagram
Diagram Objects: Creating an Object Group | Object Group Editor | Adding/Deleting objects to/from Group | Deleting Object Groups

7.11.1. Creating an Object Group

To create a new **Object Group** please do the following:

1. Click on the Group (▣) icon on the **Main** tab of the Ribbon. Your mouse cursor will change its appearance. Click on the diagram area to create a new **Object Group**. Or right click on the empty space of diagram and choose **Create Object | Group**. A semi-transparent rectangle with the object group name will appear on the diagram:



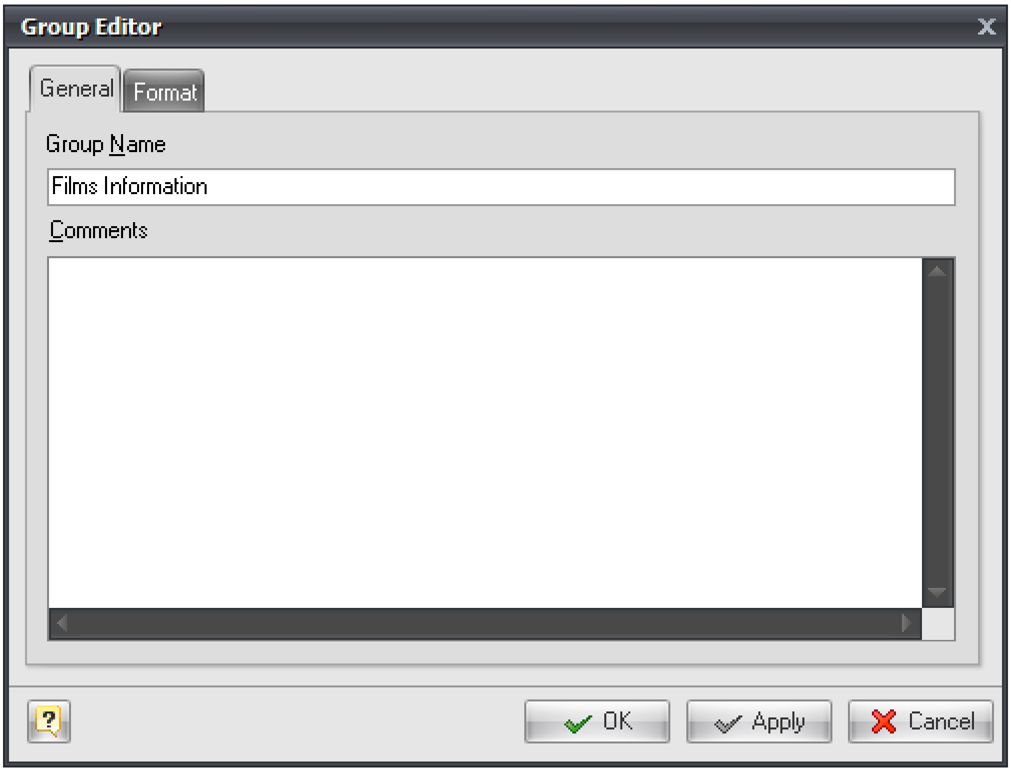
2. (optional) Double click on the new object group symbol in the diagram to call the Object Group Editor. You can set the name of the group and define its color and transparency.

3. (optional) Take a look at Adding/Deleting objects to/from Group to learn how can you add diagram objects to this group.

See also:
Diagram Objects: Object Groups | Object Group Editor | Adding/Deleting objects to/from Group

7.11.2. Object Group Editor

The **Group Editor** is provided for altering name and format of Object Group. To open the **Group Editor**, simply double-click an object group on the diagram or select the **Properties** item from the context menu.



Films Information

The **Group Editor** contains two tabs, each of which will be described below.

General

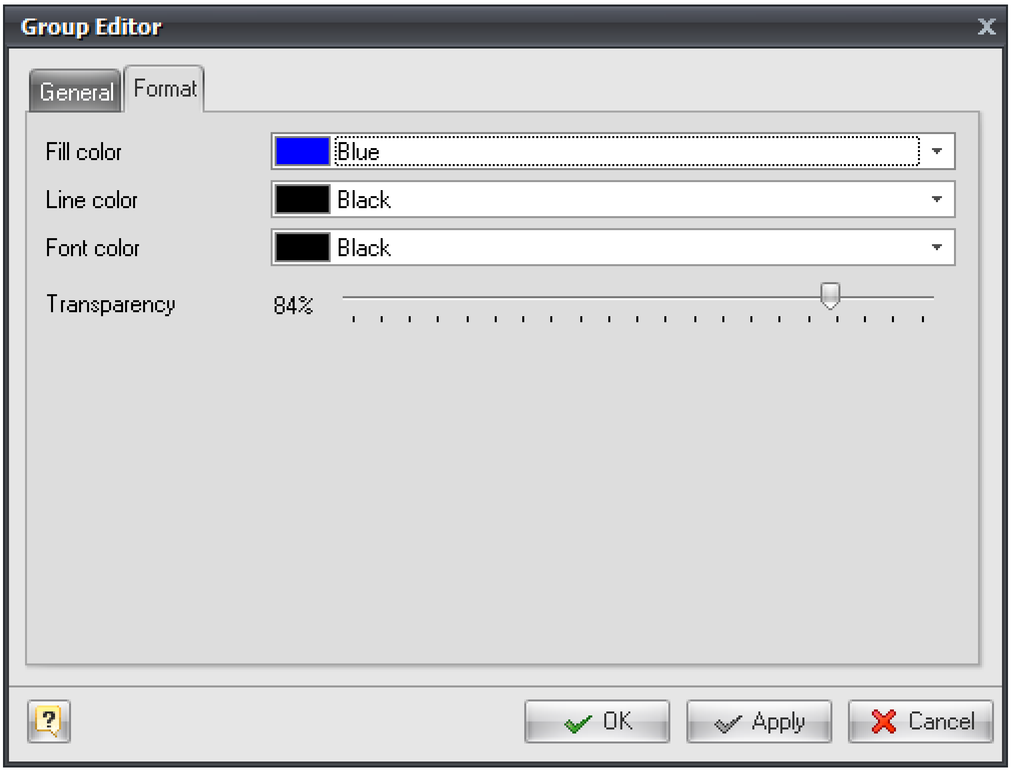
This tab allows you to set the name of the group and write comments for it. There are the following fields on the tab:

Group Name
Sets the name of the group.

Comments
A comment for the group. Group comments are not displayed at diagram.

> **Please note:**
> **Group Editor** has no **Generate** option like other object editors because **Object Groups** do not affect physical database structure and are not used during database generation.

Format



Fill color: Blue

Line color: Black

Font color: Black

Transparency: 84%

These options allow you to set the group's appearance on diagram:

Fill color

Line color

Font color

Transparency.

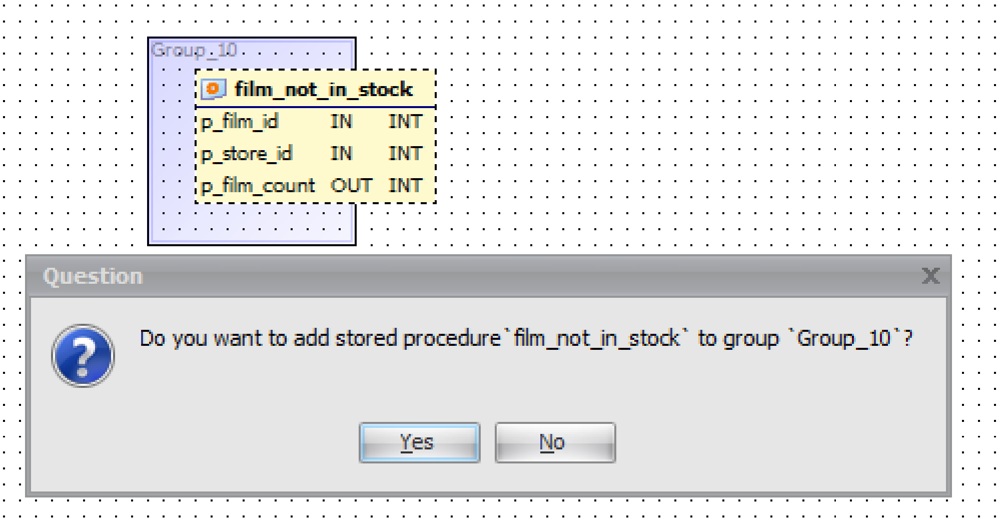
See also:
Diagram Objects: Object Groups | Creating an Object Group

7.11.3. Adding/Deleting objects to/from Group

Adding diagram objects to groups

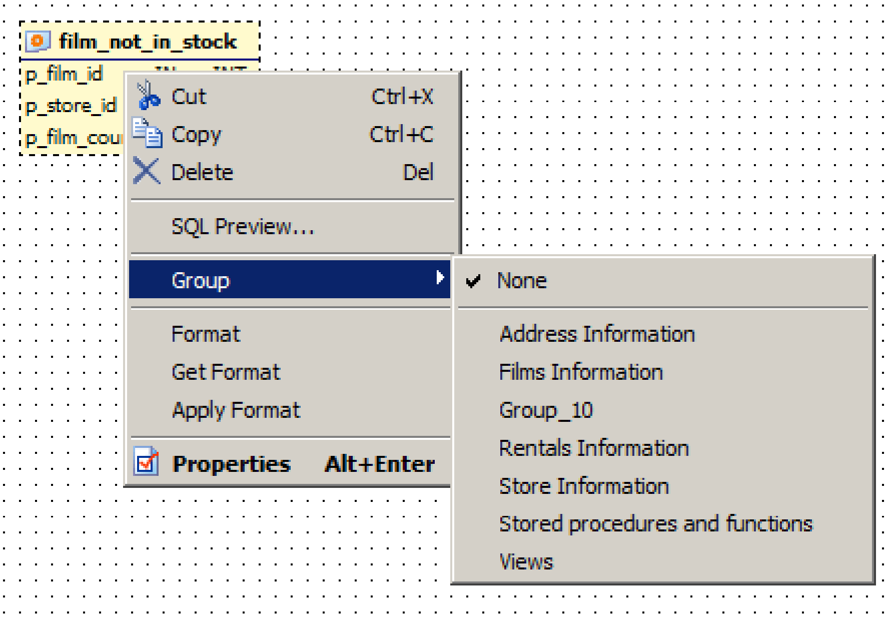
There are two ways to add diagram object to **Object Group**.

1. You can drag some object and drop it onto group symbol. **Database Designer for MySQL** will ask you "Do you want to add <object> to <group>?". After answering Yes this object will be attached to **Object Group**. **Object Group** size will be adjusted to fit object if needed.



Do you want to add stored procedure `film_not_in_stock` to group `Group_10`?

2. Another way to add object to **Object Group** is using object's context menu. Right-click on diagram object and use **Group** sub-menu to attach object to some group, detach from current group (**None** item) or to change current group. Current group object belongs to is marked with check mark in **Group** sub-menu. **Object Group** size will be adjusted to fit object if needed.



Please note:
Diagram objects that not belong to any **Object Group** can still be placed upon **Object Group** rectangle on diagram. Such objects can look like attached objects. But they will not be moved with group.

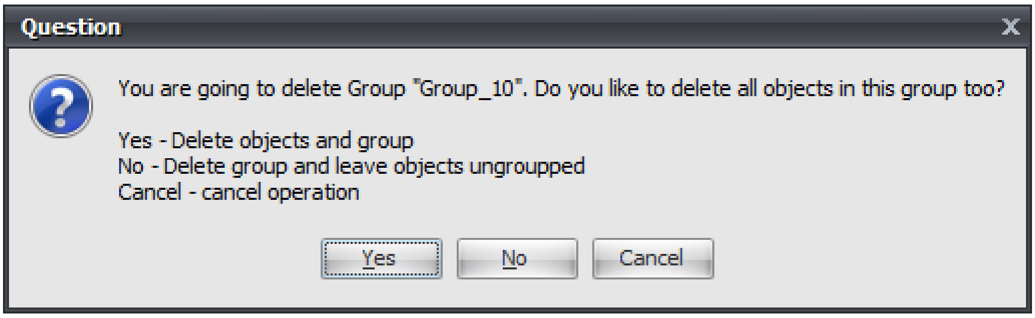
Deleting (detaching) diagram object from group

Once object is attached to some **Object Group** this object will be moved with group. And you can't just move object outside the group because group will enlarge size to fit moving object. So the only way to detach diagram object from **Object Group** is to use object's context menu. Right-click on diagram object and use **Group** | **None** item to detach object from group. Please note, that diagram object will remain at the same position after this operation. But you can move it outside the group rectangle because it is not attached to it.

See also:
Diagram Objects: Object Groups | Creating an Object Group | Object Group Editor | Deleting Object Groups

7.11.4. Deleting Object Groups

When you delete Object Group (using **Delete** key, using **Delete** ☒ tool or using group's context menu) **Database Designer for MySQL** asks you "Do you want to delete all objects in this group".



You are going to delete Group "Group_10". Do you like to delete all objects in this group too?

Yes - Delete objects and group
No - Delete group and leave objects ungroupped
Cancel - cancel operation

There are 3 possible answers:

Yes
Object Group will be deleted with all diagram objects belong to it.

No
Object Group will be deleted, but all its diagram objects will stay on diagram detached from group preserving their positions.

Cancel
Will close this dialog without any changes to diagram objects or **Object Groups**.

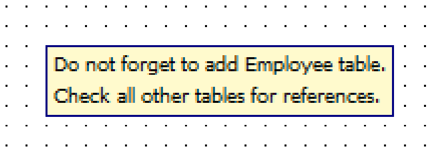
See also:
Diagram Objects: Object Groups | Creating an Object Group | Object Group Editor | Adding/Deleting objects to/from Group

7.12. Notes

Notes

You can type in the **Note box** any comment concerning your diagram. **Notes** are useful for writing various comments, such as object functionality or role description, things to do, etc in your diagram.

The following picture demonstrates a sample **Note**:



Do not forget to add Employee table.
Check all other tables for references.

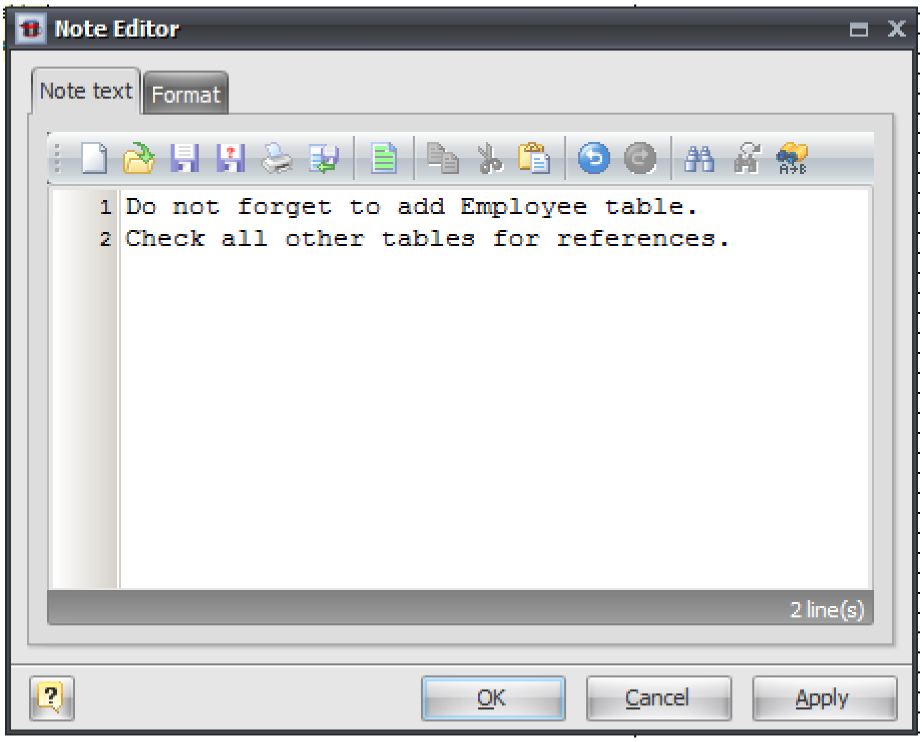
> **Please note:**
> **Note** is auxiliary object, it doesn't affect database generation.

Placing Note to the diagram

To place a note in the diagram, select the **Note** () item on **Main** tab of the Ribbon. Then click anywhere in the diagram. Certainly, you can place multiple notes in one diagram.

Modify the text in Note

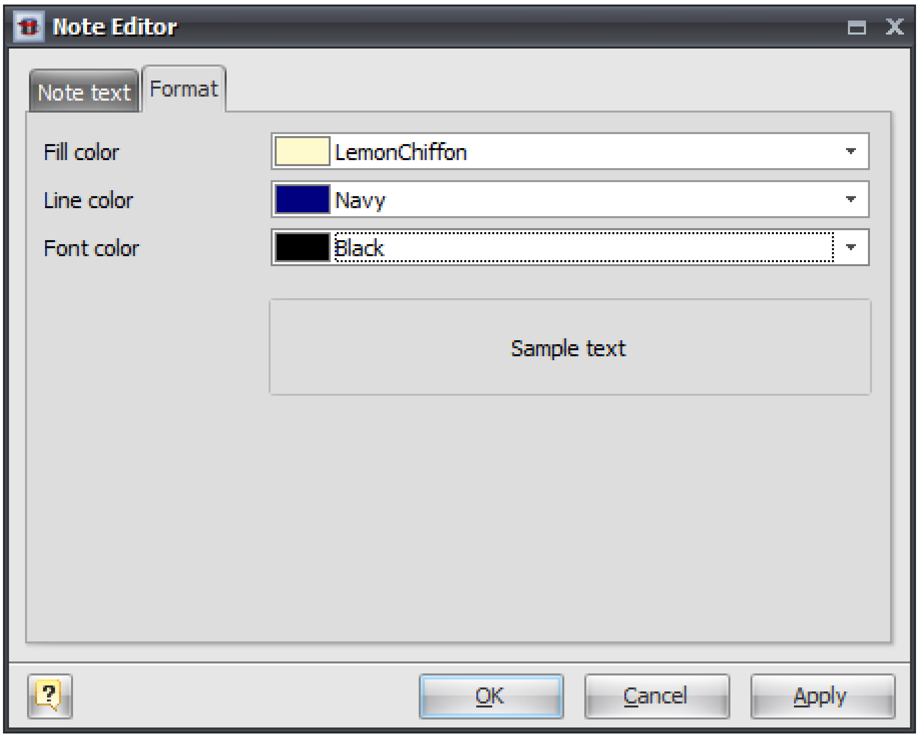
To modify the information, that is displayed in **Note**, double click on it. Then the **Note Editor** will appear:



```
Do not forget to add Employee table.
Check all other tables for references.
```

You can edit, save or load **Note** text. Press **OK** to apply your changes and close this dialog.

Formatting Note



Fill color: LemonChiffon

Line color: Navy

Font color: Black

Format tab of **Note Editor** allows you to adjust **Note** appearance:

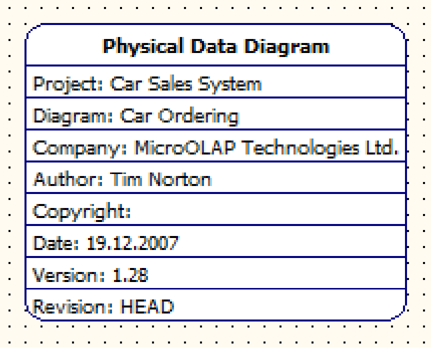
- **Fill color;**
- **Line color;**
- **Font color.**

7.13. Stamps

Stamps

A **Stamp box** displays essential information about the diagram such as: the name of the diagram itself, the author and version of the diagram and the date of modification. This information is retrieved from the diagram properties.

The following picture demonstrates sample **Stamp**:



Physical Data Diagram

Project: Car Sales System

Diagram: Car Ordering

Company: MicroOLAP Technologies Ltd.

Author: Tim Norton

Copyright:

Date: 19.12.2007

Version: 1.28

Revision: HEAD

> **Please note:**
> **Stamp** is auxiliary object, it doesn't affect database generation.

Placing Stamp to the diagram

To place **Stamp** on the diagram, select the **Stamp** (▤) item on **Main** tab of the Ribbon. Then click on any place of diagram. Note, that you can place multiple stamps on one diagram.

Modify Stamp information

To modify the information, that is displayed on the **Stamp**, double click on it or open Diagram Properties editor. Edit the appropriate fields to modify the stamp and diagram information.

See also:
Diagram: Diagram Properties

8. Diagram Functions

Database Designer for MySQL supports the following diagram functions:

Merge Diagram
Allows to merge content of two diagrams.

Check Diagram
Allows to check your database diagram for most typical errors and defects.

Compare Diagrams
Allows you to compare diagram with another diagram or physical database.

Diagram Versioning (Revisions)
Allows you to create "snapshots" of your diagram state after particular stages of diagram development. Later you can return your diagram to one of previous revisions, see changes between two revisions or even generate SQL-code for changing database structure from one diagram revision to another.

Diagram Synchronization
Allows to synchronize your diagram with changes in physical database. For example you changed several tables in your database directly using ALTER TABLE queries and you want to have this changes in your diagram.

8.1. Merge Diagram

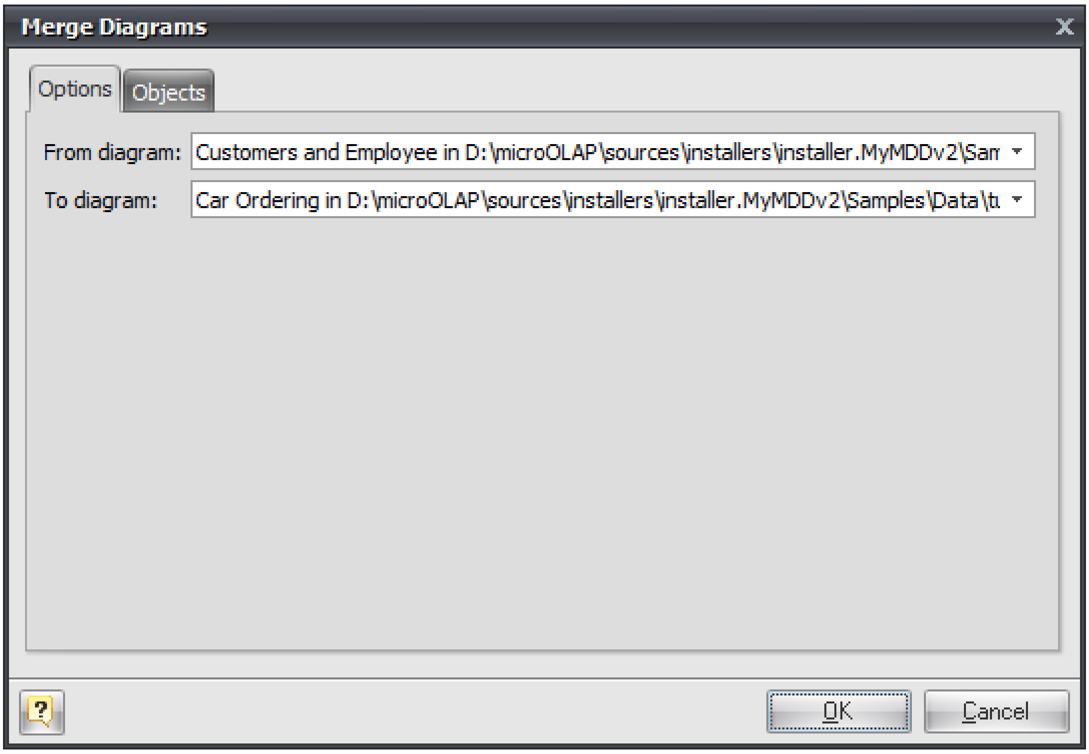
The **Merge Diagram** tool of the **Database Designer for MySQL** allows you to merge content of two diagrams. This allows you to create new cumulative diagram, which will include content of your two diagrams.

To merge two diagrams, please follow these steps:

1. Open two diagrams you want to merge. One of them will accumulate its own content and content of other diagram.

2. Start Merge Diagram tool by selecting the Merge Diagram () item on Diagram tab of the Ribbon.

3. Select diagrams you want to merge.



From diagram: Customers and Employee in D:\microOLAP\sources\installers\installer.MyMDDv2\Sam

To diagram: Car Ordering in D:\microOLAP\sources\installers\installer.MyMDDv2\Samples\Data\t

From diagram
Choose the source diagram from drop-down menu, which contains list of opened diagrams. Objects of this diagram will be added to the destination diagram.

To diagram
Choose the destination diagram from the drop-down menu, which contains the list of the opened diagrams. This diagram will contain its own objects and objects of source diagram.

4. Select the source diagram objects to merge. Go to the Objects tab.

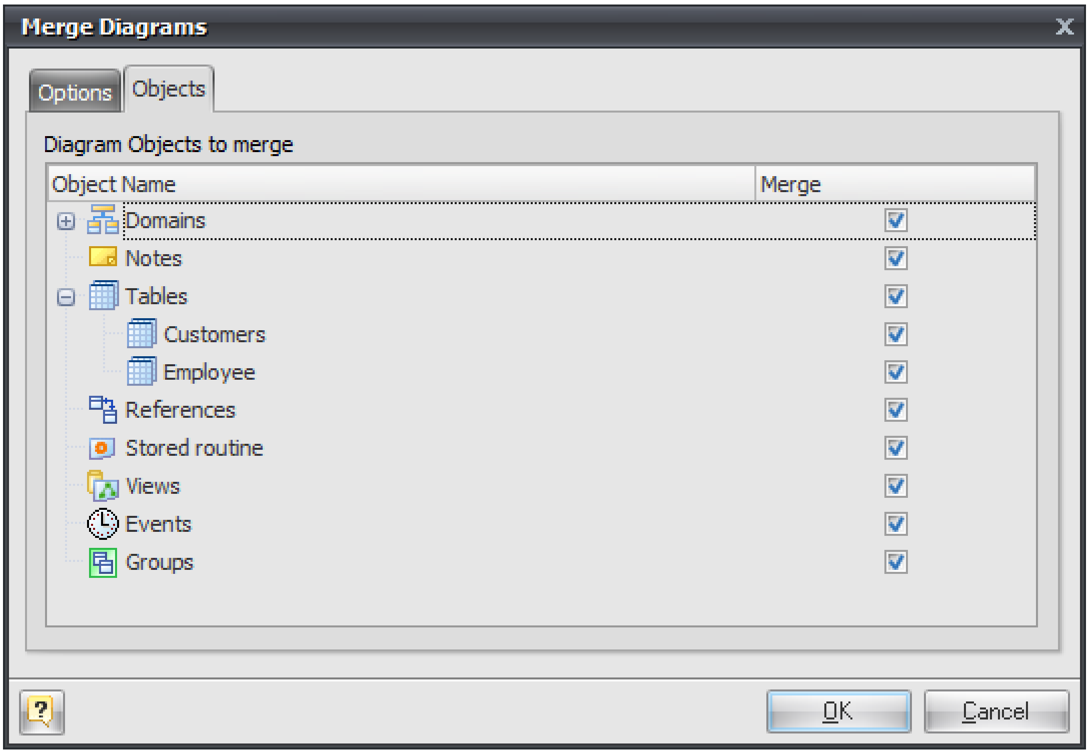


Diagram Objects to merge

Object Name	Merge
Domains	☑
Notes	☑
Tables	☑
Customers	☑
Employee	☑
References	☑
Stored routine	☑
Views	☑
Events	☑
Groups	☑

5. Click on the appropriate checkboxes in Merge column to select objects you want to add to the destination diagram. To select/deselect all objects of particular type, click on appropriate checkbox near them (by default all objects are already selected).

6. Click OK to add the selected objects of the source diagram to the destination diagram.

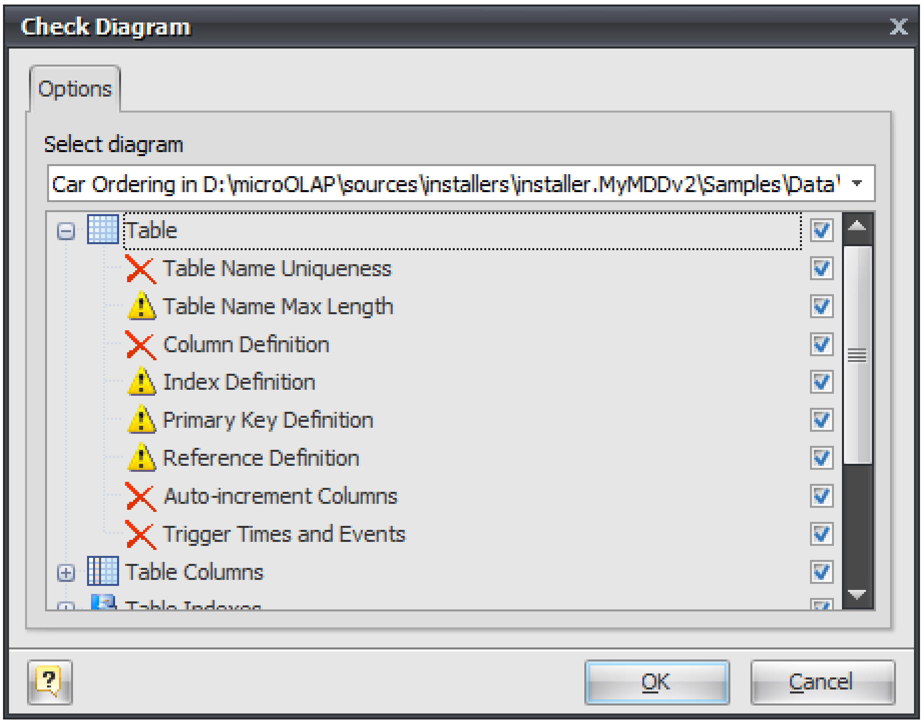
7. (optional) Save the resulted (destination) diagram to a new file.

See also:
Diagram: Open an Existing Diagram | Saving a Diagram

8.2. Check Diagrams

The **Check Diagram** tool of the **Database Designer for MySQL** allows you to check your database diagram for most typical errors and defects. The result of the check goes in a well structured form, using which you can easily bring your diagram to correspondence with the common standards of database modeling.

To open the **Check Diagram** dialog use the **Check Diagram** ( ) item on **Diagram** tab of the Ribbon or press **F4**.



Select diagram: Car Ordering in D:\microOLAP\sources\installers\installer.MyMDDv2\Samples\Data'

- ☑ Table
  - ✗ Table Name Uniqueness ☑
  - ⚠ Table Name Max Length ☑
  - ✗ Column Definition ☑
  - ⚠ Index Definition ☑
  - ⚠ Primary Key Definition ☑
  - ⚠ Reference Definition ☑
  - ✗ Auto-increment Columns ☑
  - ✗ Trigger Times and Events ☑
- ☑ Table Columns
- ☑ Table Indexes

Use the **Select diagram** drop-down list to select the diagram from the list of currently opened diagrams. Current opened diagram is selected by default.

The tree list below allows you to select what warnings and errors should be taken into account during the check. All warnings and errors are divided into categories, which correspond to the diagram objects. Remove selection from the warning/error or from the whole category to exclude it from the check.

These are the descriptions for all available warnings and errors.

Table

Error "Table Name Uniqueness"
Check the diagram for the uniqueness of each table name within a diagram.

Warning "Table Name Max Length"
MySQL allows only 64 characters in table names and cuts names if they are longer than this.

Error "Column Definition"
Check if each table within the diagram owns at least one column.

Warning "Index Definition"
Check if each table within the diagram owns at least one index.

Warning "Primary Key Definition"
Check if a primary key is defined for each table within the diagram.

Warning "Reference Definition"
Check if each table within the diagram is linked with other tables.

Error "Auto-increment columns"

Check if each table within the diagram has no more than one auto-increment column, as otherwise MySQL will not allow such table to be created.

Error "Trigger Times and Events"
Check if every trigger in each table within diagram has unique Time and Event combination.

Table Columns

Error "Column Name Uniqueness"
Check diagram tables for the uniqueness of each column name within the table.

Warning "Column Name Max Length"
MySQL allows only 64 characters in column names and cuts names if they are longer than this.

Warning "Auto-increment Column Definition"
Check if each auto-increment column within a table is a part of a primary key.

Table Indexes

Error "Index Name Uniqueness"
Check diagram tables for the uniqueness of each index name within the table.

Warning "Index Name Max Length"
MySQL allows only 64 characters in index names and cuts names if they are longer than this.

Warning "Duplicate Index Column"
Check if each table column is indexed only once.

References

Warning "Reference Table Type"
Check if both linked tables use storage engine that supports references (for example InnoDB or Falcon) for each diagram reference.

Error "Reference Column Data Types"
Check whether the linked columns are of the same data type for each diagram reference.

Error "Reference Column does not exists"
Check an existence of referenced column in parent table for each diagram reference.

Error "Reference Name uniqueness"
Check diagram references for uniqueness of each reference name.

Error "Check "SET NULL" FK-action for NOT NULL column"
MySQL does not allow SET NULL action for foreign keys that reference column marked with NOT NULL attribute.

Domains

Error "Domain Name Uniqueness"
Check diagram for the uniqueness of each domain name within a diagram.

Stored Routine

Error "Stored Routine Name Uniqueness"
Check diagram stored procedures and functions for the uniqueness of name within a schema.

Warning "Stored Routine Name Max Length"
MySQL allows only 64 characters in stored routine names and cuts names if they are longer than this

Views

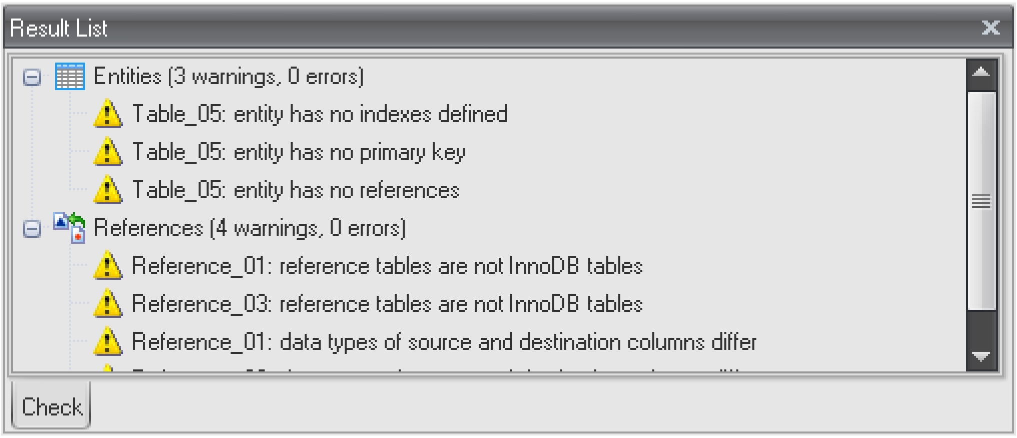
Error "View Name Uniqueness"
Check diagram stored procedures and functions for the uniqueness of name within a schema.

Warning "View Name Max Length"
MySQL allows only 63 characters in stored routine names and cuts names if they are longer than this.

Error "View was created on a not existing table"
Check existence of tables, which are used in the view.

Error "View was created on a not existing column"
check existence of table columns, which are used in the view.

After you click **OK** the check process will be displayed within the **Output** window and the result of the check will be displayed within the **Result** window in the same categorized view as described above.



Entities (3 warnings, 0 errors)
Table_05: entity has no indexes defined
Table_05: entity has no primary key
Table_05: entity has no references
References (4 warnings, 0 errors)
Reference_01: reference tables are not InnoDB tables
Reference_03: reference tables are not InnoDB tables
Reference_01: data types of source and destination columns differ

Double-click on a warning or an error in the list opens the editor window for the appropriate object (Table Editor, View Editor, Domain Manager and so on).

See also:
Interface User Guide: Docking Windows
Diagram Objects: Table Editor | Column Editor | Domain Manager | Reference Editor | Index Editor | Stored Routine Editor | Trigger Editor | View Editor

8.3. Compare Diagram

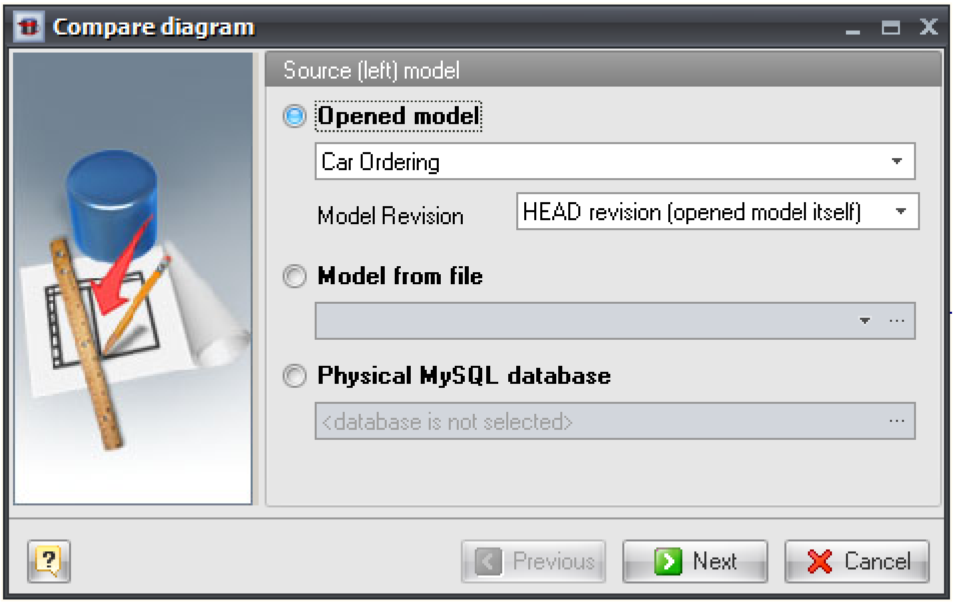
Compare Diagram tool is introduced in **Database Designer for MySQL v2**. It allows you to compare different diagrams, revisions of the same diagram, diagram with physical database and even one physical database to another. After comparing you can generate SQL-code that applies differences between diagrams to database or apply differences to opened diagram.

To open the **Compare Diagram** tool use the **Compare Diagram** ( ) item on **Diagram** tab of the Ribbon or press **Ctrl + F6**.

Compare Diagram tool compares two database models: *target model* is compared with *source model*. The result of comparing is an answer to question "What is changed in *target model* in compare with *source model*?". Database model can be opened diagram, diagram from file or model built by this tool from physical database.

Compare Diagram is Wizard-like tool. It collects all needed parameters at several steps and then performs comparing.

Step 1: Source (left) model



Compare diagram

Source (left) model

● Opened model

Car Ordering

Model Revision: HEAD revision (opened model itself)

○ Model from file

○ Physical MySQL database

<database is not selected>

You have to select *source model* at this step. There are 3 possible options:

Opened model
Use **Opened model** drop-down list to select one of diagrams currently opened in **Database Designer for MySQL**. Currently active diagram is selected by default in this list. This diagram can have some unsaved changes.
You can also select model revision from **Model Revision** drop-down list if you created revisions of your diagram. If you want to use current diagram state or you have no revisions of your diagram select "*HEAD revision*" option.

Model from file
This option allows you to select file with diagram that is not currently opened with **Database Designer for MySQL**.

Physical MySQL database
This option allows you to select some connection profile from Database Connection Manager. This physical database will be reverse engineered to temporary model. Then this model will be used as *source model* for comparing.

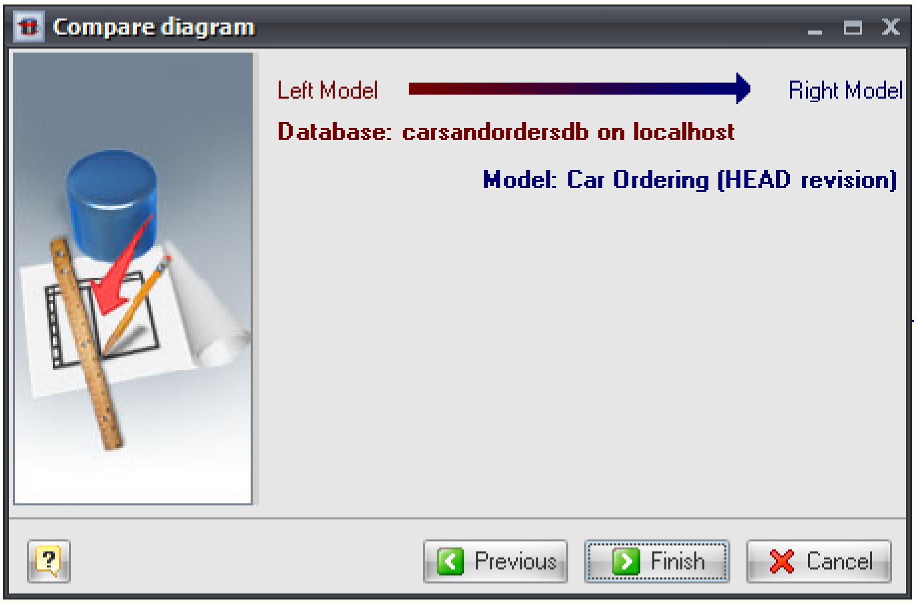
After selecting *source model* press **Next** button to move to Step 2.

Step 2: Target (right) model

This step allows to select *target model* for comparing. This step has the same options like first step. The only difference is that you are selecting *target mode* instead of *source model* here.

After selecting *target model* press **Next** button to move to Step 3.

Step 3: Review selected models



Database: carsandordersdb on localhost

Model: Car Ordering (HEAD revision)

This step just shows you both *source* and *target* models for you to be sure that you have selected proper models in proper order. If you sure with your selection you can press **Finish** to start compare process. Or you can use **Previous** button to return to model selection steps.

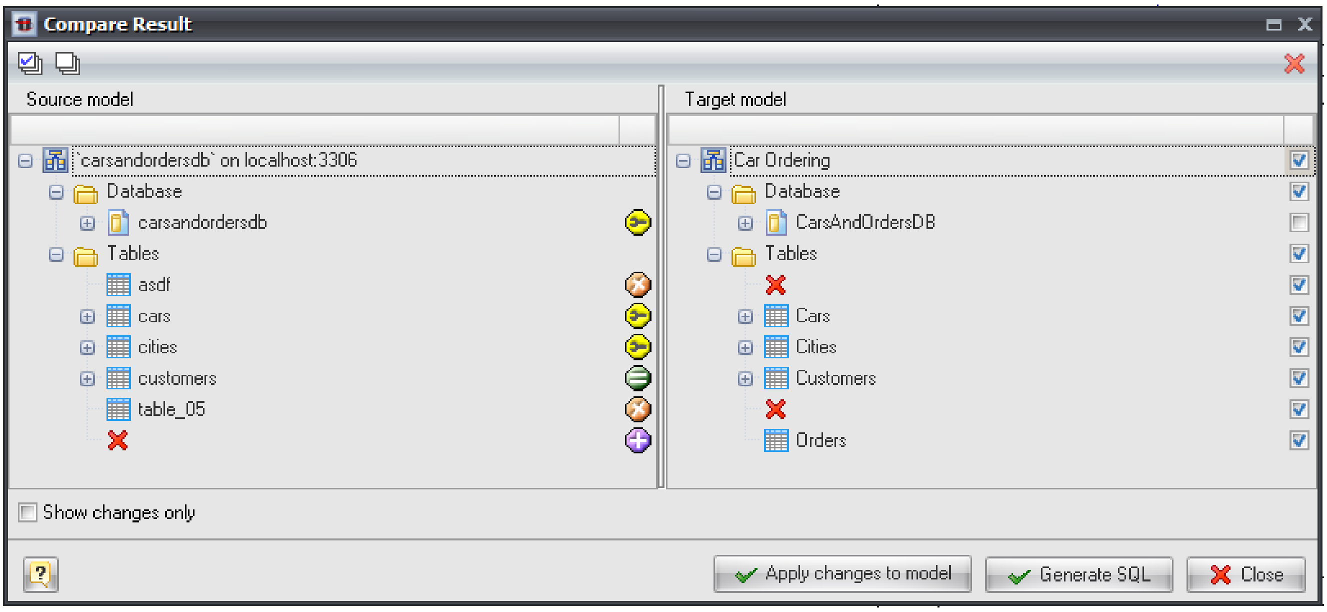
Database Designer for MySQL will reverse engineer physical databases if needed, compare models object by object and show Compare Result dialog with results of comparing.

See also:

Diagram Functions: Diagram Versioning (Revisions) | Compare Result dialog
Database Accessing Tools: Database Connection Manager

8.3.1. Compare Result dialog

Compare Result dialog is shown automatically after comparing diagrams or performing Database Modification. It shows results of comparing for every diagram object in two object trees: left tree show objects in *source model* and changes in this objects in right tree (*target model*).



Every object in left tree has one of following changes marks:

⊕ - **Object is added**. This means that database object does not exist in source model but exists in target model.

⊗ - **Object is deleted**. This means that database object exists in source model but do not exist in target model.

⊙ - **Object is modified**. This means that this object was found in both source and target models but one or more of its parameters or sub-objects are changed. Use plus mark near this object's name to see exact differences.

⊜ - **Objects are equal**. This means that object was found in both source and target models and there are no any differences between them.

Example: You can see on provided screen-shot that **Car Ordering** diagram has following changes in compare to physical database **`carsandordersdb`**:

- *asdf* and *table_05* were deleted (they exist in database, but there are no such tables in diagram)
- *Orders* table was added (there is no such table in database, but it is in diagram)
- *Cars* and *Cities* tables have some changes (table options and columns)

- *Customers* table has no any changes (table in database is equal to table in diagram).

You can use **Show changes only** checkbox to hide all unchanged objects from both trees. This can be very useful if you compare two models with many objects but with only several changes.

Compare Result dialog allows to perform following actions:

- **Apply changes to model**
  You can apply all found differences to diagram. That means that all added objects will be added to diagram, deleted objects will be deleted. All changes for each object will be applied to appropriate diagram object. This is useful if you need to synchronize changes made in physical database with your diagram. Take a look at Diagram Synchronization topic for additional details.

  You can use checkboxes in right tree to select particular changes for applying to diagram.

  After pressing **Apply changes to model** button you'll be prompted to select one of opened diagrams to apply changes. Do not forget to Save your diagram after applying changes. Or you can close diagram without saving changes if something goes wrong.

- **Generate SQL**
  This action allows to generate SQL-script for applying all found differences to physical database. There will be *CREATE <object>* statement generated for every added object, *DROP <object>* statement for every deleted object and so on. This is useful when you need to apply changes in your diagram to physical database. But there can be more difficult cases: for example comparing two physical databases and synchronizing their structure.

  You can use checkboxes in right tree to select particular changes for including in SQL-script.

  After pressing **Generate SQL** button **Modify Database Options** dialog will be shown. Take a look at Database Modification topic for additional details.

- **Close**
  Just closes this dialog without any changes to database or diagram.

See also:
Diagram: Saving a Diagram
Diagram Functions: Compare Diagram | Diagram Synchronization
Database Functions: Database Modification

8.4. Diagram Versioning (Revisions)

Since version 2.0 **Database Designer for MySQL** supports diagram versioning (revisions). Revision of diagram is diagram state saved in particular moment. In other words list of diagram revisions is a kind of diagram's changes history. Each revision has number, date and comment. With **Database Designer for MySQL** you can create revisions of your diagrams, return your diagram state to this revisions, save revisions to separate diagram file, compare revisions to review changes between them, generate SQL code to apply changes between revision to physical database. Revisions in **Database Designer for MySQL** has a lot of common with source version control systems such as Subversion, Git, CVS and others.

> **Please note:**
> Diagram revisions are saved to separate file with the same filename but with **.dmrev** extension added.

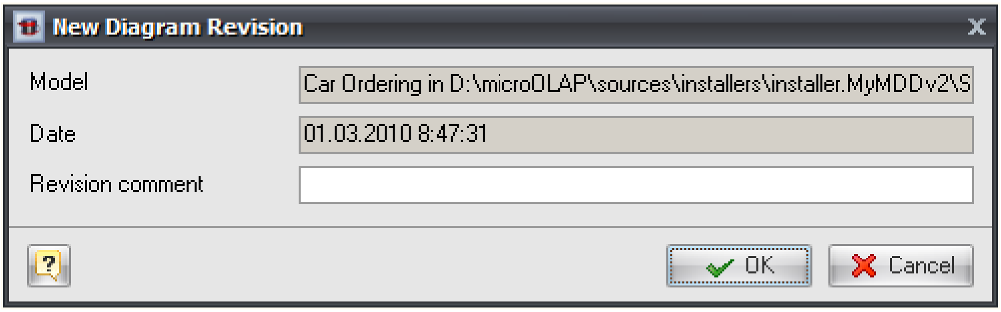
Database Designer for MySQL provides following tools to work with revisions:

- Create Diagram Revision tool.
- Revision Manager to manage revisions: view, delete, export to separate model file and so on.
- Compare Diagram tool can be used to compare revisions of the same diagram and to generate SQL code that applies changes between revisions to physical database.

See also:
Diagram Functions: Creating Diagram Revision | Revision Manager | Comparing Diagram Revisions | Compare Diagram

8.4.1. Creating Diagram Revision

To create new diagram revision use **Create Revision** ( ) item on **Diagram** tab of the Ribbon. The **New Diagram Revision** dialog will appear.



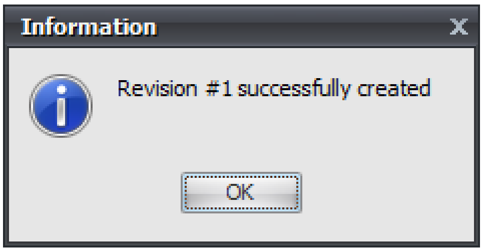
New Diagram Revision

Model: Car Ordering in D:\microOLAP\sources\installers\installer.MyMDDv2\S

Date: 01.03.2010 8:47:31

Revision comment:

This dialog shows you name and file name of current diagram, current date and time and allows you to enter comment for new revision. New revision is created after pressing **OK** button. After creating new revision **Database Designer for MySQL** shows information message box with revision number.



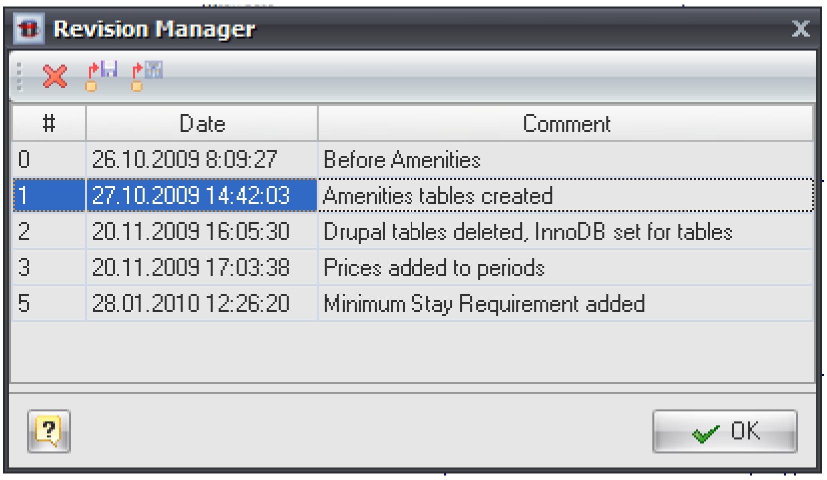
Information

Revision #1 successfully created

See also:
Diagram Functions: Diagram Versioning (Revisions) | Revision Manager | Comparing Diagram Revisions

8.4.2. Revision Manager

Revision Manager is used to manage all revision of current diagram. To open **Revision Manager** dialog use **Revision Manager** (▦) item on **Diagram** tab of the Ribbon.



#	Date	Comment
0	26.10.2009 8:09:27	Before Amenities
1	27.10.2009 14:42:03	Amenities tables created
2	20.11.2009 16:05:30	Drupal tables deleted, InnoDB set for tables
3	20.11.2009 17:03:38	Prices added to periods
5	28.01.2010 12:26:20	Minimum Stay Requirement added

Revision Manager allows you to perform following actions:

✕ **Delete revision**
Delete selected revision. For example some revisions are obsolete in your project. Or you made a revision at wrong point of development and do not need it.

Save model revision to file
Save selected revision to standalone file. You can use this action to restore diagram to some revision in the past: just save this revision to your current diagram file.

Open revision as temporary diagram
Open selected revision in new window to view diagram itself. It will be opened as unsaved diagram.

See also:
Diagram Functions: Diagram Versioning (Revisions) | Creating Diagram Revision | Comparing Diagram Revisions

8.4.3. Comparing Diagram Revisions

After creating several diagram revisions you may need to compare two revisions of diagram or some diagram revision with it current version to examine what is changed. There is no separate tool for comparing diagram revisions in **Database Designer for MySQL**. But you can compare diagram revisions using Compare Diagrams tool easily.

This can be useful if you generated physical database from some diagram revision in the past. Then several changes were made to your diagram and you want to apply this changes to physical database. Database Modification will compare diagram with physical database, and this can lead to direct database changes to be lost (for example database was modified directly by other developer). If you generate SQL-code only for changes between revisions in your diagram you will not affect any other database objects besides changed in your diagram between this revisions.

To compare two diagram revisions or some diagram revision with its current version follow this steps:

1. Open diagram with revisions you want to compare.

2. Open Compare Diagrams tool using the **Compare Diagram** ( ) item on **Diagram** tab of the Ribbon or press **Ctrl + F6**.

3. Currently opened model is selected by default at "Source (left) model" page. You need to select source diagram revision from **Model Revision** drop-down control. This is diagram revision you want to examine changes from. Then press **Next** button.

4. On "Target (right) model" page you need to select target diagram revision from the same drop-down control. This is diagram revision you want to compare source revision with. If you want compare source revision with current diagram version just left "HEAD revision" item selected in **Model Revision** drop-down control. Then press **Next** button.

5. Review selected diagram revisions at final wizard page and press **Finish** button.

6. Compare Result dialog will appear. Use "Show changes only" checkbox to hide unchanged items. There are all changes between selected revisions listed in this dialog.

7. Press **Generate SQL** button if you want to generate SQL-code to apply this changes to physical database. Or press **Close** button to close this dialog.

See also:
Diagram: Open an Existing Diagram
Diagram Functions: Compare Diagram | Diagram Versioning (Revisions) | Creating Diagram Revision | Revision Manager
Database Functions: Database Modification

8.5. Diagram Synchronization

Diagram Synchronization overview

Diagram Synchronization is modifying your diagram to lead to some physical database state. For example you have made some changes directly in your physical database using *ALTER TABLE* statements or with some third-party database administration tool. Now you want to make the same changes to your diagram, i.e. synchronize your diagram with physical database.

> **Please note:**
> **Diagram Synchronization** is opposite process to Database Modification.

That's how the **Diagram Synchronization** tool works:

1. Reverse engineers your existing MySQL database.

2. Compares your current database diagram with the result of reverse engineering.

3. Creates a list that contains differences between database objects and diagram objects. Then lets you to examine this list and to select changes you want to apply to diagram.

4. After analyzing the difference list, applies selected changes to opened diagram.

Database Designer for MySQL allows to synchronize your diagram with physical database using Compare Diagram tool.

Step by step guide

> **Please note:**
> **Diagram Synchronization** can perform a lot of changes to you diagram. So it is usually a good idea to backup your diagram before synchronizing it. You can just create new diagram revision before synchronizing diagram with database to have a way to rollback synchronization later.

To synchronize your diagram, please follow these steps.

1. Open diagram you want to synchronize with some database.

2. Start Compare Diagram tool by selecting the Compare Diagram ( ) item on Diagram tab of the Ribbon or press Ctrl + F6.

3. Currently opened model is selected by default at "Source (left) model" page. So press Next button.

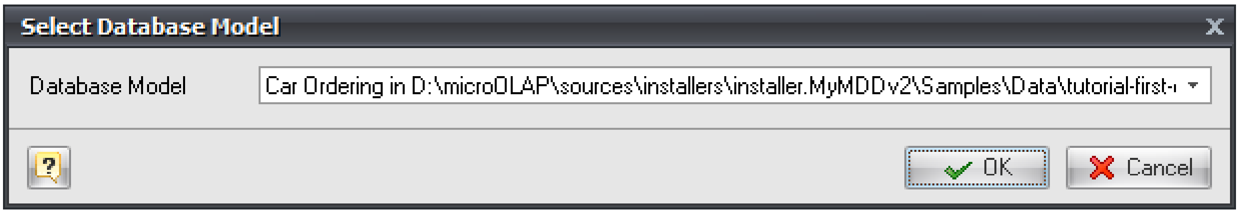
4. On "Target (right) model" page you need to select physical database you want to synchronize diagram with. Select Physical MySQL database option. Then use ... button to select Database Connection Profile. Press Next button.

5. Review selected diagram and Database Connection profile at final wizard page and press Finish button.

6. Compare Result dialog will appear. Use "Show changes only" checkbox to hide unchanged items. There are all changes between selected physical database and current diagram listed in this dialog. You can use checkboxes in right objects tree to select / unselect particular changes to be applied to diagram.

7. Press Apply changes to model button if you want to apply selected changes to diagram. Or press Close button to close this dialog.

8. After pressing Apply changes to model button Select Database Model dialog will appear with current diagram selected by default.



Select Database Model
Database Model
Car Ordering in D:\microOLAP\sources\installers\installer.MyMDDv2\Samples\Data\tutorial-first-
OK
Cancel

9. Ensure that proper diagram is selected in drop-down control and press OK button. Selected in Compare Result dialog changes will be applied to this diagram.

10. Review your diagram. If everything is OK you can save it.

See also:
Diagram: Open an Existing Diagram | Saving a Diagram
Diagram Functions: Compare Diagram | Compare Result dialog | Creating Diagram Revision
Database Functions: Database Modification

9. Database Functions

Database Designer for MySQL supports the following database related functions:

Database Generation
Allows to generate SQL script, that represents the diagram you developed and (optionally) executes it on the database server.

Database Modification
Allows to generate SQL script that leads your physical database to the current state of your diagram.

9.1. Database Generation

The **Database Generation** tool can generate SQL script, that represents the diagram you developed and (optionally) executes it on the database server.

You can generate database in two ways:

- Directly execute a generated script on a MySQL server using SQL Executor.
- Generate a script to be executed on MySQL server at a later time.

In both cases, the database generation commands are saved in a script file. You must always provide path to the script file.

Database Generation

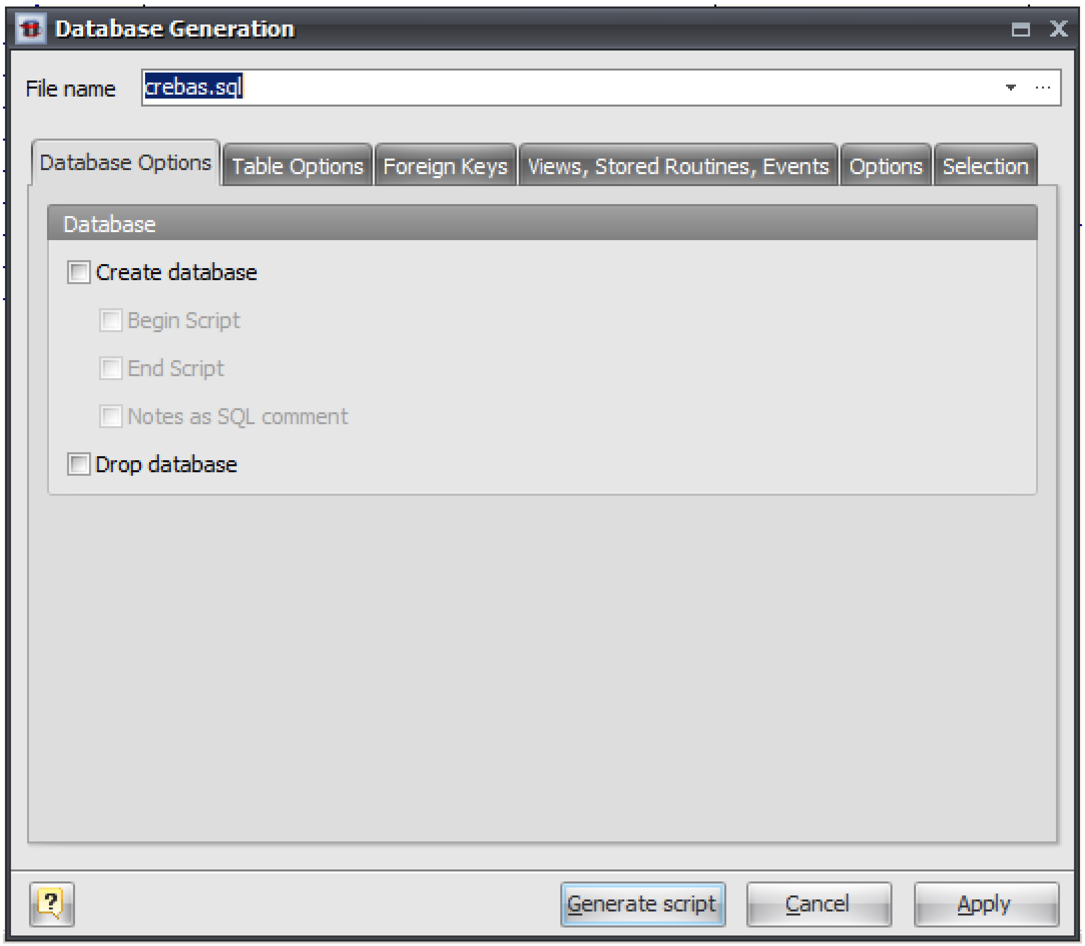
To generate database, start the **Database Generation** tool by using **Generate Database** ( ) item on **Database** tab of the Ribbon or pressing **Ctrl+G**. The following pictures demonstrate **Database Generation** tool interface.

- **File name field**
  This field allows you to set file, in which generated SQL statements will be stored. Click on the ... button near the field to browse to file on the file system.

Database Generation tool consist of several tabs, which contain SQL generation options. Let's explore them.

Database Options

In the **Database Options** tab of the **Database Generation** tool you can set database generation options.



Database Generation

File name: crebas.sql

Database

☐ Create database

☐ Begin Script

☐ End Script

☐ Notes as SQL comment

☐ Drop database

Create Database
This option enables generation of the database. Set this option on if you want to create a new physical database instead of using an existing one. The table structure (i.e. tables) will be generated in both cases.

Begin Script
This option enables inserting the begin script before the CREATE DATABASE statement. You can edit database begin script in Database Editor.

End Script
This option enables inserting the end script after the CREATE DATABASE statement. You can edit

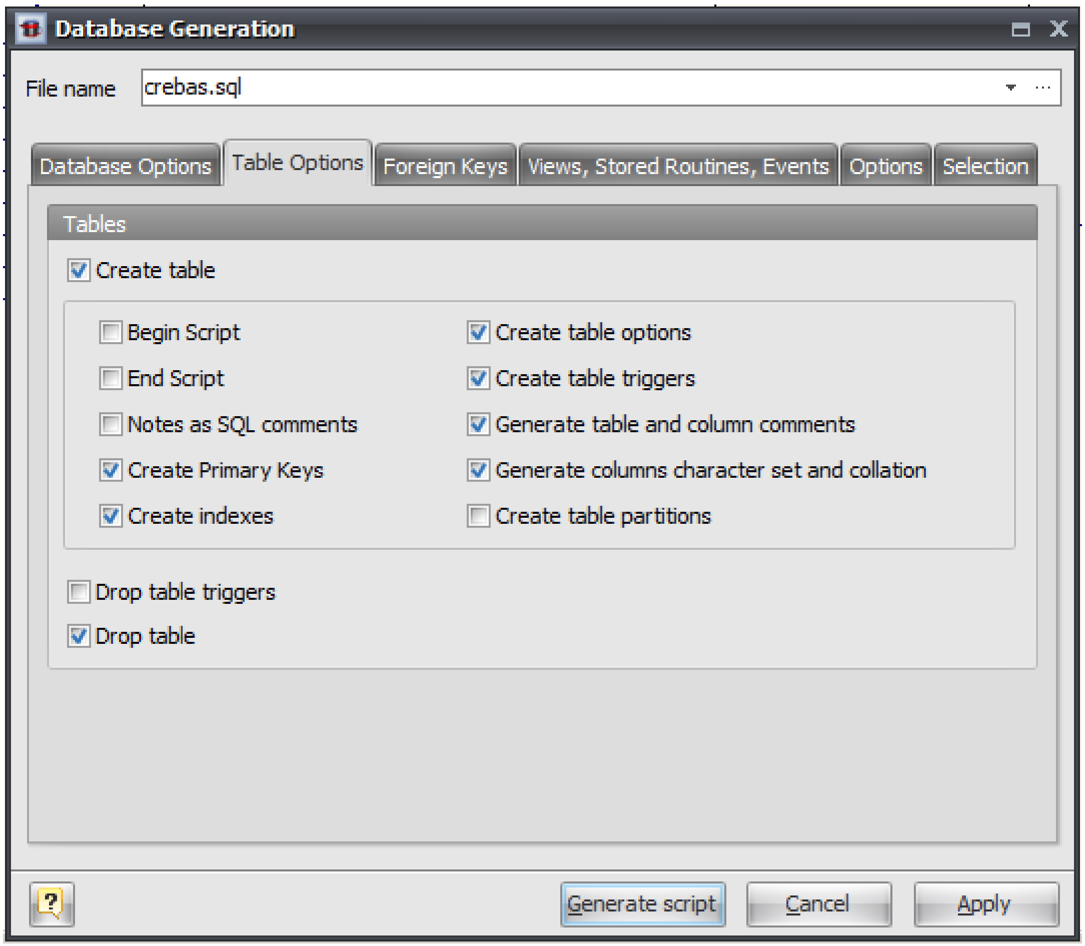
database end script in Database Editor.

Notes as SQL Comment
This option enables showing of database comments in SQL script.

Drop Database
This option enables dropping old database if it was already exist.

Table Options

In the **Table Options** tab of the **Database Generation** tool you can set tables generation options.



File name: crebas.sql

☑ Create table

☐ Begin Script
☐ End Script
☐ Notes as SQL comments
☑ Create Primary Keys
☑ Create indexes
☑ Create table options
☑ Create table triggers
☑ Generate table and column comments
☑ Generate columns character set and collation
☐ Create table partitions

☐ Drop table triggers
☑ Drop table

Create Tables
This option enables generation of tables.

Begin Script
This option enables inserting the begin script (it can be set using Table Editor) before the CREATE TABLE statement.

End Script
This option enables inserting the end script (it can be set using Table Editor) after the CREATE TABLE statement.

Notes as SQL Comments
This option enables showing of table notes in SQL script, which was set in the Table Editor. Notes

will be inserted to the SQL script as commented strings.

Create Primary Keys
This option enables generation of table primary keys.

Create Indexes
This option enables generation of indexes for the tables.

Create Table Options
This option enables setting of table options, such as ENGINE, ROW_FORMAT and others in the SQL script.

Create table triggers
Generate triggers set on table.

Generate table and column comments
This option adds COMMENT clause to column and table definitions if some comment set in Table Editor.

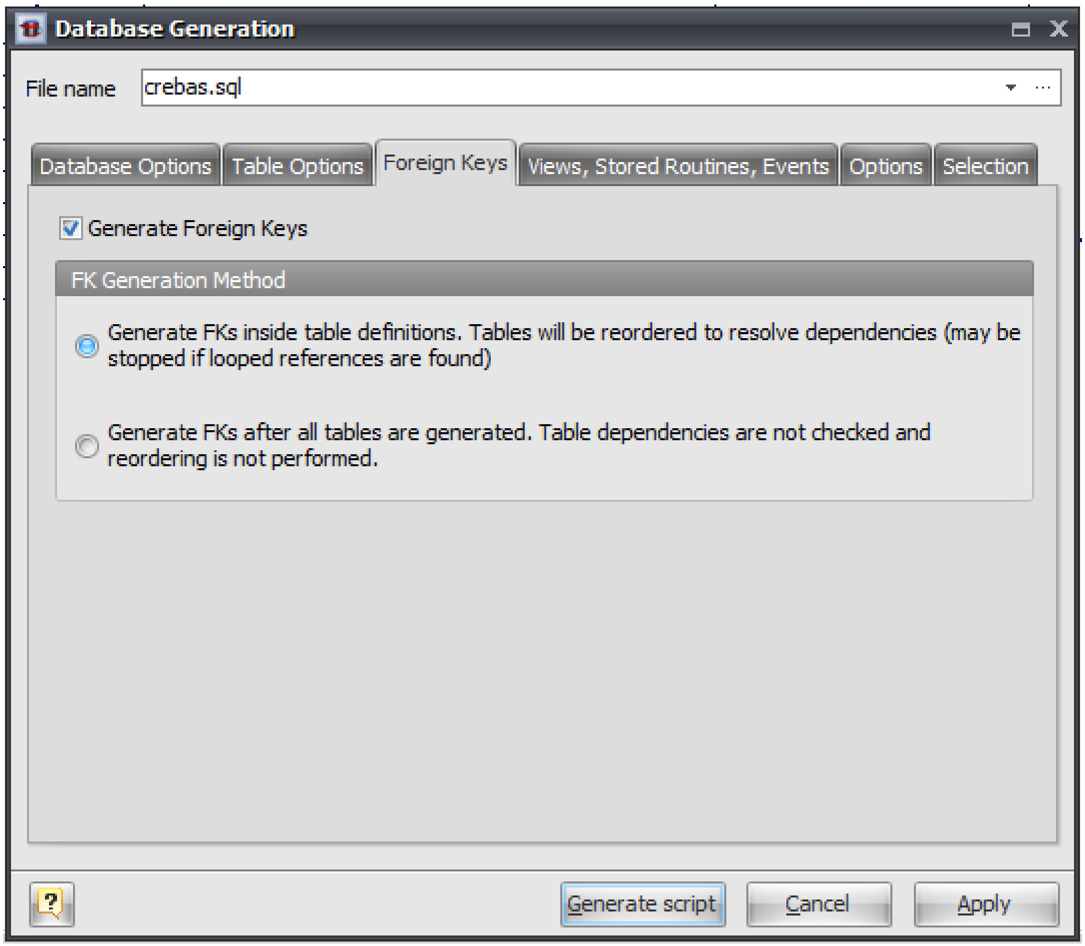
Create table partitions
Enable generation of partitions for tables.

Drop table triggers
This option generates DROP TRIGGER IF EXISTS statement before generating CREATE TRIGGER statement. There is no need to set this this option if Drop table option is set.

Drop table
This option enables dropping old tables if they were already exist. I.e. enables generating DROP TABLE IF EXISTS "<TABLE_NAME>" statement before CREATE TABLE statement.

Foreign Keys

Foreign Keys tab of **Database Generation** tool allows to set up foreign keys generation options.



File name: crebas.sql

☑ Generate Foreign Keys

● Generate FKs inside table definitions. Tables will be reordered to resolve dependencies (may be stopped if looped references are found)

○ Generate FKs after all tables are generated. Table dependencies are not checked and reordering is not performed.

Generate Foreign Keys

This option enables generation of table foreign keys presented as References on diagram.

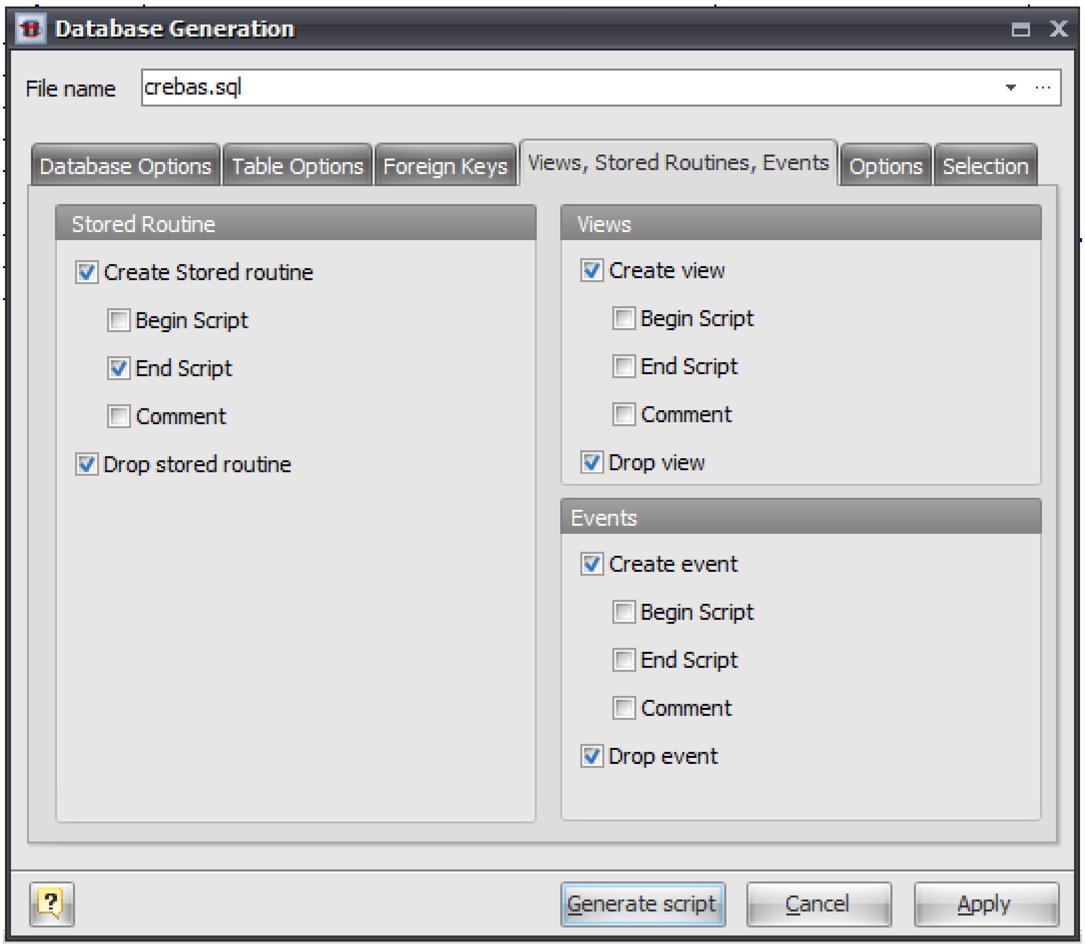
FK Generation Method

This option defines foregn keys generation method:

Generate FKs inside table definitions.
All FKs are generated as a part of table definition inside CREATE TABLE statement. This method requires reordering of diagram tables, because some tables may depend on other ones. Sometime there can be cycle references between tables that can not be generated properly using this method. That is why second method of FKs generation is preferred.

Generate FKs after all tables are generated.
When using this method all FKs will be generated using ALTER TABLE ... ADD CONSTRAINT statements. This method does not require tables reordering.

Views, Stored Routines, Events



File name: crebas.sql

Stored Routine

- ☑ Create Stored routine
  - □ Begin Script
  - ☑ End Script
  - □ Comment
- ☑ Drop stored routine

Views

- ☑ Create view
  - □ Begin Script
  - □ End Script
  - □ Comment
- ☑ Drop view

Events

- ☑ Create event
  - □ Begin Script
  - □ End Script
  - □ Comment
- ☑ Drop event

Stored Routines Group

Create Stored Routine
This option enables generation of stored routines (i.e. stored functions and procedures).

Begin Script
This option enables inserting the begin script before the SQL generation statement.

End Script
This option enables inserting the end script after the SQL generation statement.

Comment
This option enables showing of stored routines comments in SQL script.

Drop Stored Routine
This option enables dropping old stored routines if it were already exist.

Views Group

Create View
This option enables generation of database views.

Begin Script
This option enables inserting the begin script before the SQL generation statement.

End Script

This option enables inserting the end script after the SQL generation statement.

Comment
This option enables showing of view comments in SQL script.

Drop Views
This option enables dropping old views if it were already exist.

Events Group

Create View
This option enables generation of database events.

Begin Script
This option enables inserting the begin script before the events SQL generation statement.

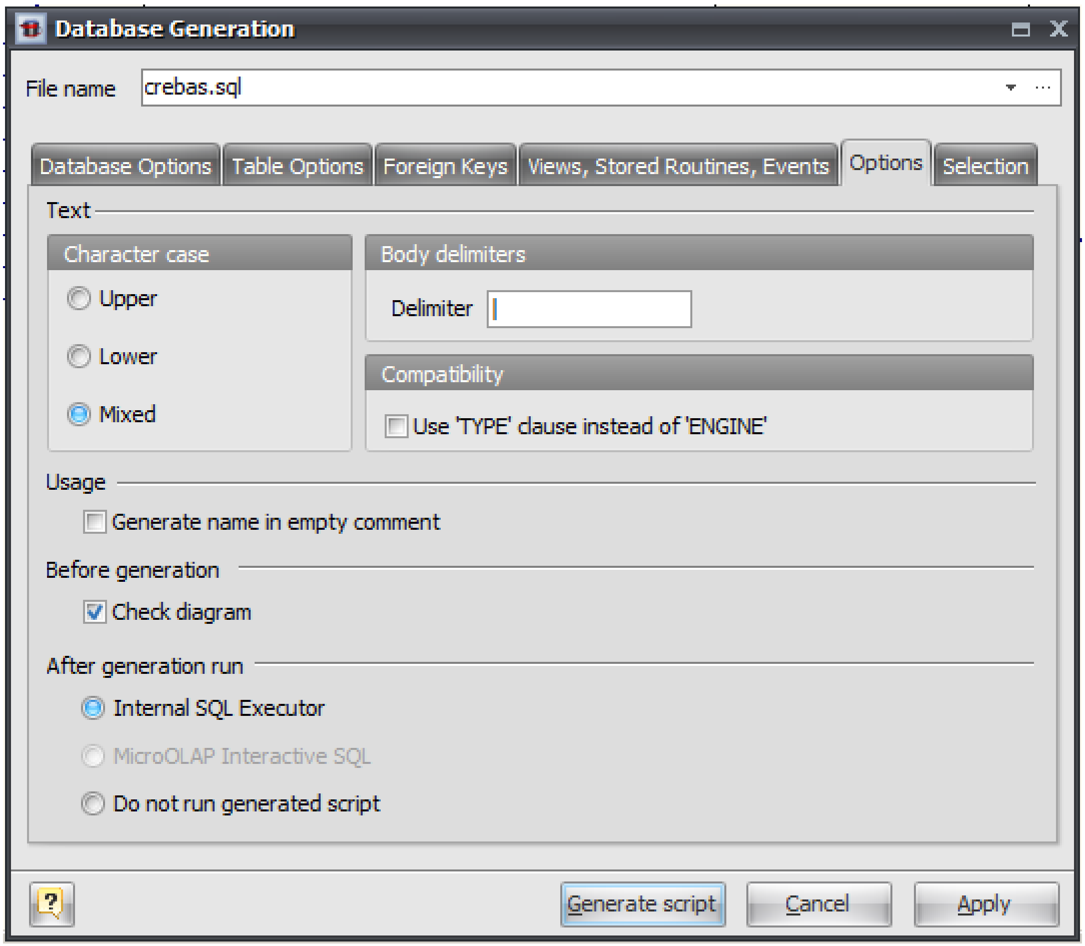
End Script
This option enables inserting the end script after the events SQL generation statement.

Comment
This option enables showing of event comments in SQL script.

Drop Views
This option enables dropping old events if it were already exist.

Options

This tab allows set generation-related options.



File name crebas.sql

Text

Character case

○ Upper

○ Lower

● Mixed

Body delimiters

Delimiter

Compatibility

□ Use 'TYPE' clause instead of 'ENGINE'

Usage

□ Generate name in empty comment

Before generation

☑ Check diagram

After generation run

● Internal SQL Executor

○ MicroOLAP Interactive SQL

○ Do not run generated script

Character case

Defines case of characters, which will be used in generated script. Use Mixed option to leave the characters unmodified.

Body Delimiter

Defines symbol or string to use as body delimiter with DELIMITER statement for multi-statement stored routines, events and triggers.

Use 'TYPE' clause instead of 'ENGINE'

This option enables generation of TYPE clause instead of ENGINE in CREATE TABLE statement. TYPE clause is deprecated since MySQL 5.1 so you probably do not need this option.

Generate name in empty comments

Write the name of an object if the comment field is empty.

Check diagram

Enables checking the diagram before generation.

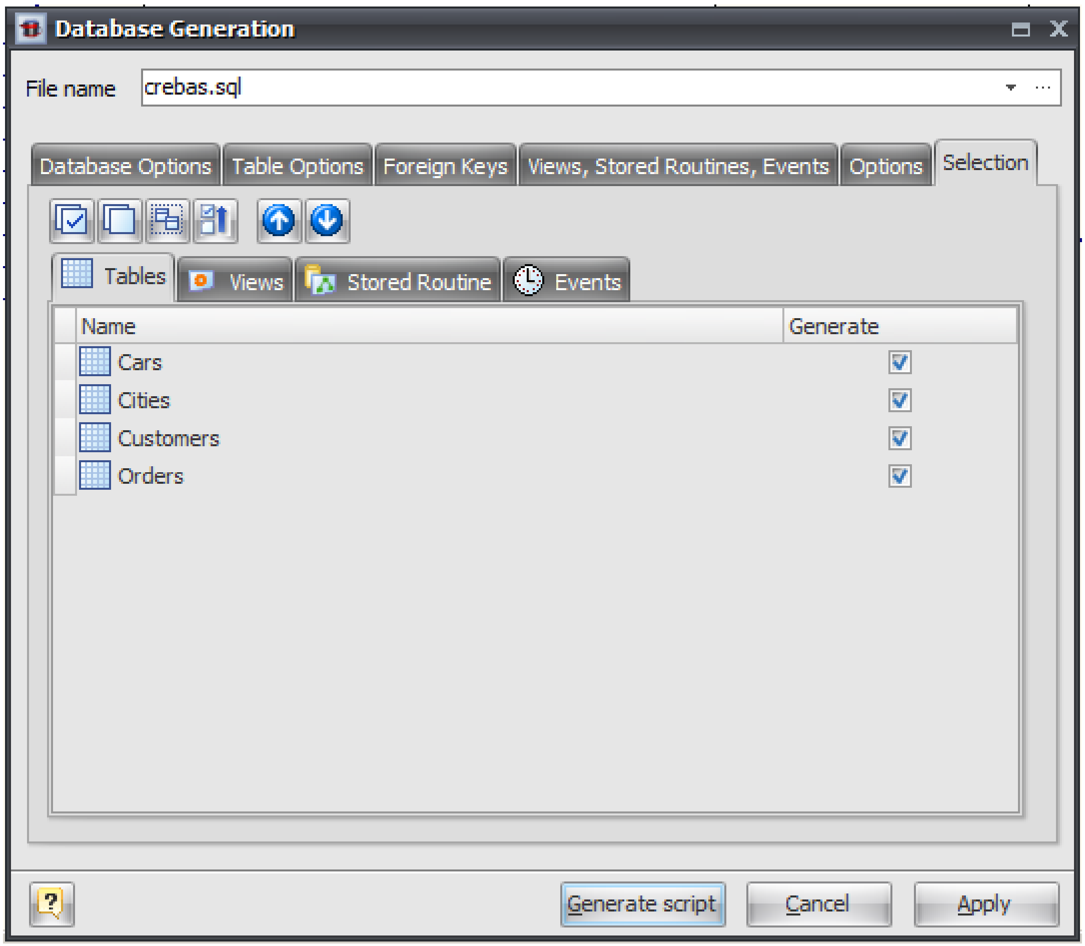
After generation, run

- Internal SQL Executor - Send the generated SQL statements into the internal SQL Executor.
- MicroOLAP Interactive SQL - This options active only if MicroOLAP Interactive SQL for MySQL is installed. Send the generated SQL statements into the advanced SQL editor MicroOLAP Interactive SQL for MySQL.
- Do not run generated script - Select this option if want just to generate SQL script without

running it. For example you can use this option if you want to execute SQL script on remote server.

Selecting objects to generate

You can select diagram objects you want to generate in SQL script or database. Use the **Selection** tab of the **Database Generation** tool for it.



File name crebas.sql

Name	Generate
Cars	☑
Cities	☑
Customers	☑
Orders	☑

There are several subtabs: **Tables**, **Stores Routines**, **Views** and **Views**. Each of which allows you to select appropriate diagram objects to generate.

To enable particular objects generation, click on the checkbox near it.

The default selection of objects to generate depends on their **Generate** property.

Pay attention to the buttons on the **Selection** tab:

Select All
Checks on all checkboxes.

Deselect All
Checks off all checkboxes.

Use graphical selection

Checks on checkboxes for objects, depending on diagram selection.

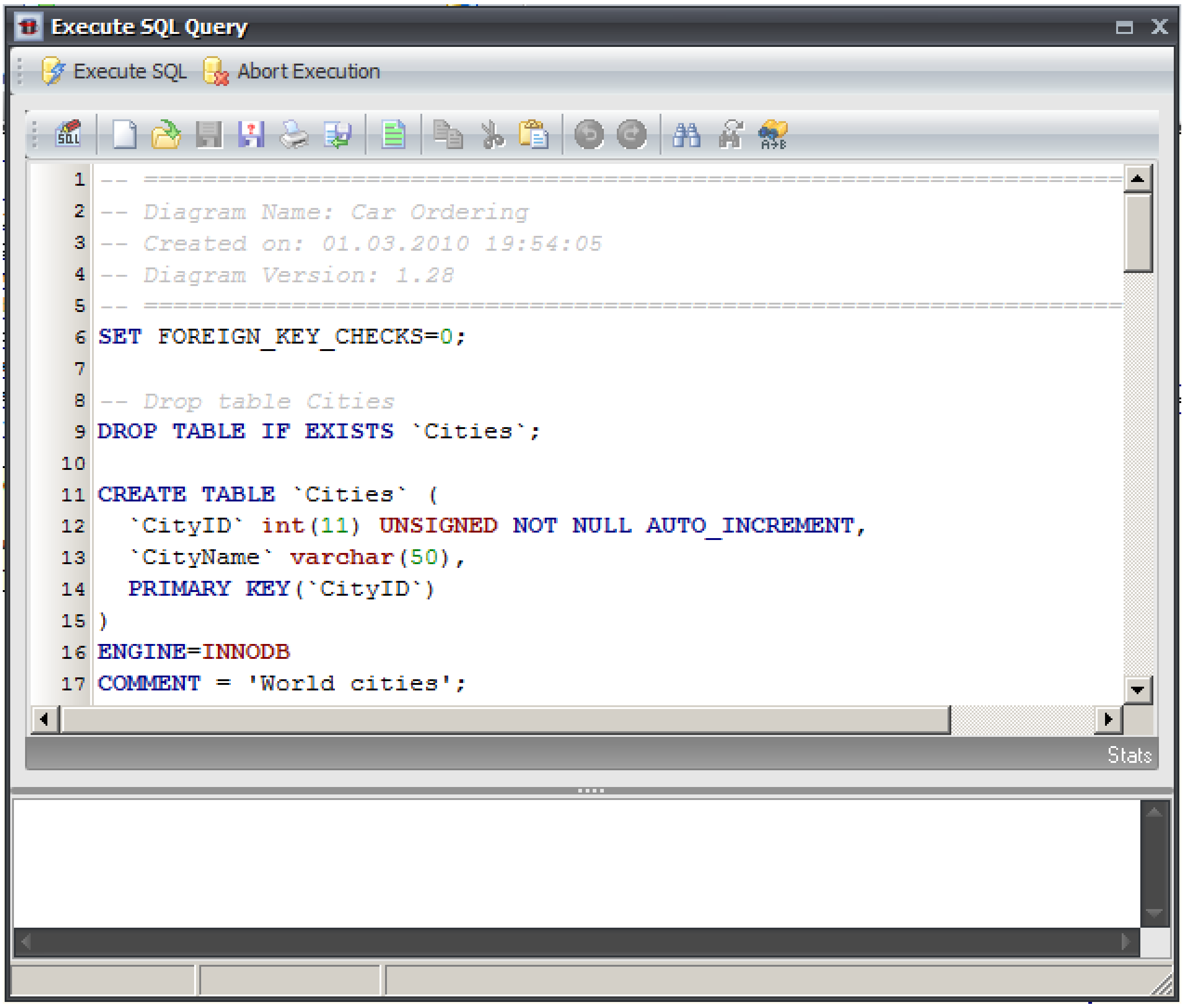
Move selected items to top
Reorders items in list for checked items to be at the beginning of list.

You can change the order of tables in which they will be placed in the generated SQL script. Use the buttons with arrows for this.

Generating, customizing and executing SQL

Click on the **Generate script** button on the **Database Generation** tool to generate SQL script. The generated SQL script will be stored in the file you have set.

If you selected **Internal SQL Executor** on **Options** tab, the SQL Executor with generated SQL statements will appear.



```
-- ====================================================================
-- Diagram Name: Car Ordering
-- Created on: 01.03.2010 19:54:05
-- Diagram Version: 1.28
-- ====================================================================
SET FOREIGN_KEY_CHECKS=0;

-- Drop table Cities
DROP TABLE IF EXISTS `Cities`;

CREATE TABLE `Cities` (
  `CityID` int(11) UNSIGNED NOT NULL AUTO_INCREMENT,
  `CityName` varchar(50),
  PRIMARY KEY(`CityID`)
)
ENGINE=INNODB
COMMENT = 'World cities';
```

You can easily customize statements for your needs. And then send them to the database server by clicking on the **Execute SQL** button.

Please, examine SQL Executor section to know more about it.

See also:
Database Accessing Tools: SQL Executor
Database Objects: Database Editor | Table Editor

9.2. Database Modification

Database Modification Overview

Once you have changed your diagram, it's usually necessary to apply these changes to your database. It's easy to do with the **Database Modification** tool. The **Database Modification** is generation of SQL script that leads your database to the current state of your diagram.

> **Please note:**
> Database modification usually causes multiple complex statements for database structure modification. It is possible that some of them may not execute correctly due to some physical reason. It's recommended to make a backup of your database before applying structure changes to database.

You can modify database in two ways:

- Directly execute a modification script on a MySQL server. Please examine Connect to a Database section to explore the database connection process;
- Generate a modification script for executing at a MySQL server some time later.

In both cases, the database modification commands are saved in a script file. You must always provide the path to the script file.

> **Please note:**
> Sometimes you need to apply changes in your diagram to physical database without leading whole database structure to current state of your diagram. For example if there are some objects in database that are not covered by your diagram or if you want to apply only particular set of diagram changes to database. You can do this using Diagram Revisions and Compare Diagram tool. Please take a look at Comparing Diagram Revisions topic for details.

That's how the **Database Modification** tool works:

1. Reverse engineers your existing MySQL database.

2. Compares the result with your current database diagram.

3. Creates a list that contains differences between database objects and diagram objects. Then lets you to examine this list and to select changes you want to apply to database.

4. After analyzing the difference list, creates necessary SQL statements, that modify database structure.

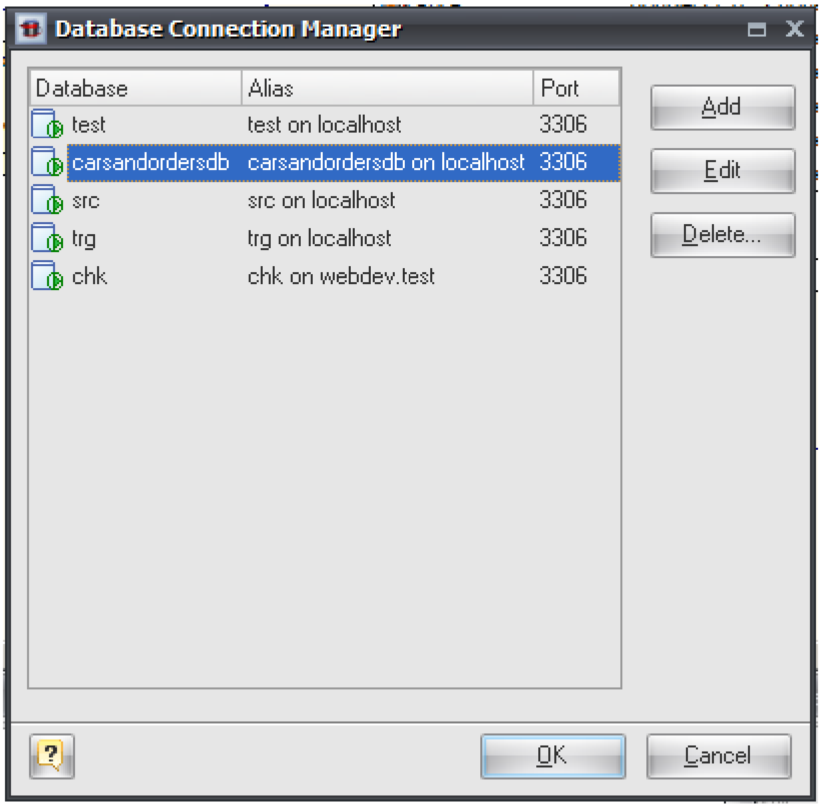
If some table is going to be modified, **Database Designer for MySQL** makes a backup copy of that table, so you can restore data and table structure later on if there will be some errors during the table structure alteration. Those backup tables have *_tmp_* string appended before their names. For example `*Cars*` table will have `*_tmp_Cars*` backup copy.

> **Please note:**
> Opposite process of applying physical database changes to diagram is described in Diagram Synchronization topic.

Step by step guide

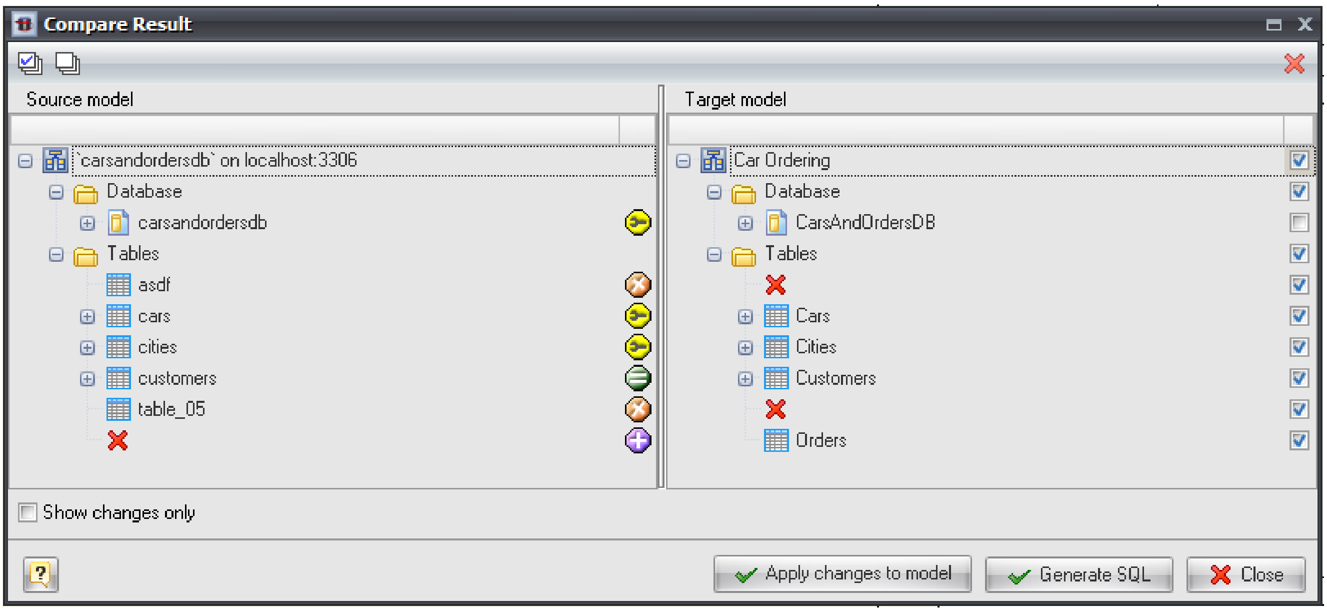
To modify your database, start **Database Modification** tool by selecting the **Modify Database** ( ) item on **Database** tab of the Ribbon or pressing **Ctrl+M**.

Database Designer for MySQL asks you with Database Connection Manager what database you want to modify.

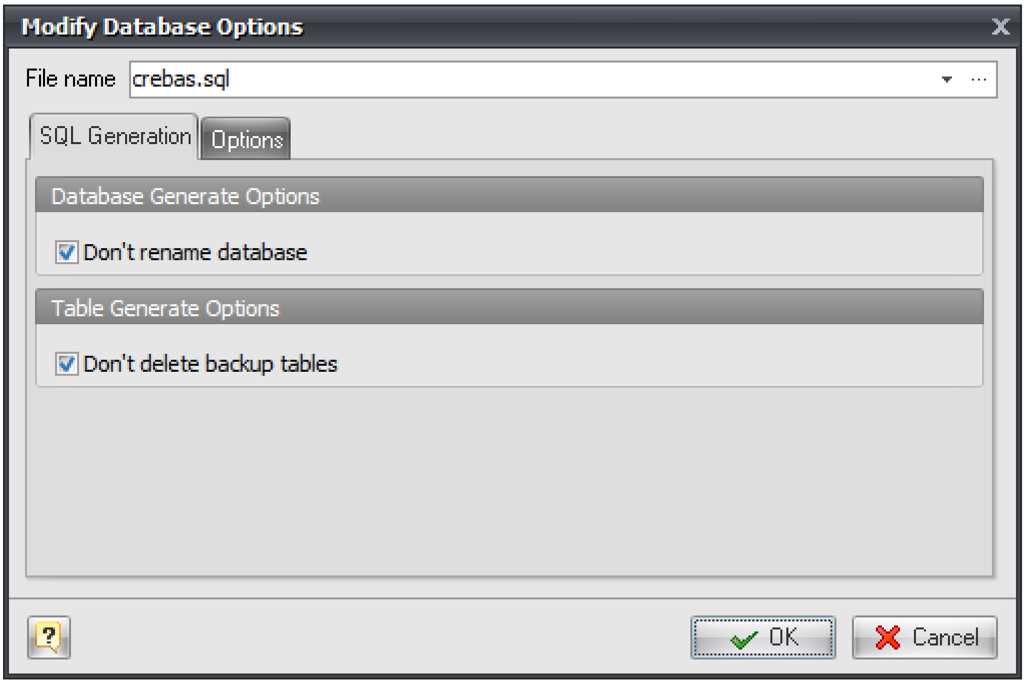


Database	Alias	Port
test	test on localhost	3306
carsandordersdb	carsandordersdb on localhost	3306
src	src on localhost	3306
trg	trg on localhost	3306
chk	chk on webdev.test	3306

After connecting to selected database **Database Designer for MySQL** performs Reverse Engineering of this database, compares it with diagram and shows Compare Result dialog with all differences found.



You can check or uncheck particular changes you want to generate SQL code for. Then press **Generate SQL** button to open **Modify Database Options** dialog.



File name: crebas.sql

☑ Don't rename database

☑ Don't delete backup tables

There are following options on **SQL Generation** tab:

Don't rename database
Check this option to prevent database renaming even if physical database name differs from one set in diagram.

Don't delete backup tables
If this option is set **Database Designer for MySQL** will not generate DROP TABLE statements for backup table copies (with _tmp_ prefix) after altering tables structure.

Settings on **Options** tab are the same as on **Options** tab of Database Generation tool.

Press **OK** button to generate SQL script. SQL Executor with generated SQL statements for database modifications will appear. You can easily customize statements according to your wants and wishes. And then send them to the database server by clicking on the **Execute SQL** button. Please, examine SQL Executor topic to know more about it.

See also:
Diagram Functions: Compare Diagram | Compare Result dialog | Diagram Versioning (Revisions) | Comparing Diagram Revisions | Diagram Synchronization
Database Functions: Database Generation
Database Accessing Tools: Database Connection Manager | Connect to a Database | SQL Executor

10. Database Accessing Tools

Database Designer for MySQL supports the following database accessing tools:

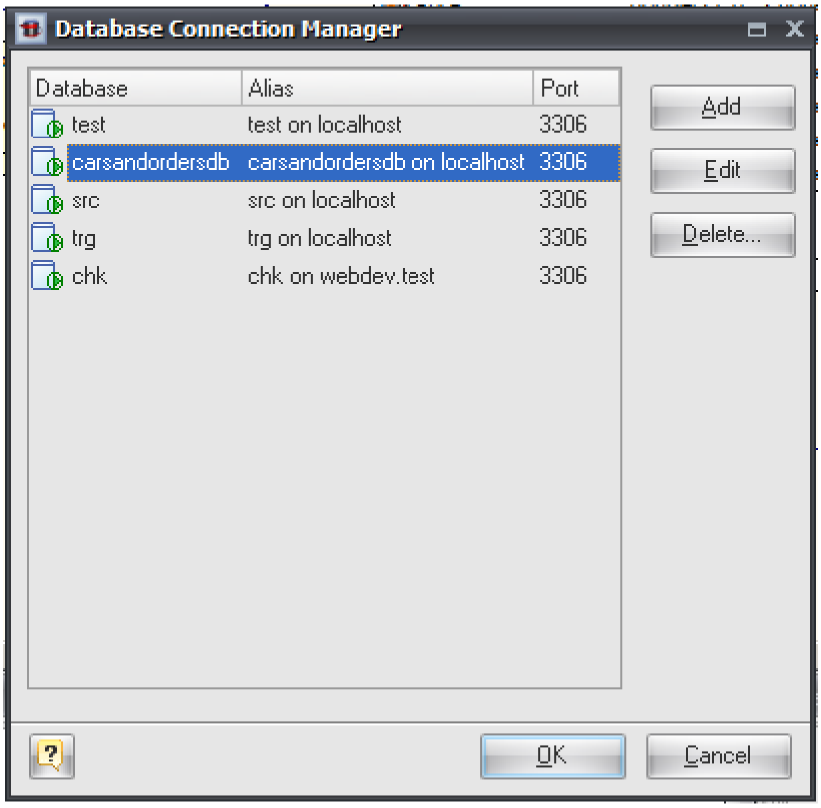
Database Connection Manager, which allows you to connect to a local or remote MySQL database for Database Generation, Reverse Engineering, Database Modification, or running SQL scripts.

SQL Executor, that allows to send SQL queries to the connected MySQL database and display the result.

10.1. Database Connection Manager

Database Connection Manager is a useful tool, which allows you to connect to a local or remote MySQL database for Database Generation, Reverse Engineering, Database Modification, Diagram Synchronization, or running SQL scripts. Use **Connect** ( ) item on **Database** tab of the Ribbon to open **Database Connection Manager**.

Database Connection Manager uses connection profiles, which allow you to set all the connection parameters for a specific database only once, and quickly connect to this database later.



Database	Alias	Port
test	test on localhost	3306
carsandordersdb	carsandordersdb on localhost	3306
src	src on localhost	3306
trg	trg on localhost	3306
chk	chk on webdev.test	3306

The dialog contains a list of available connection profiles. To connect to a database, select one of the defined profiles in the list, and click **OK**. Connection is assigned to your current diagram, so that each opened diagram could have its own connection.

To add a profile to the list, click the **Add** button and enter the connection properties using Connection Profile Editor launched.

Click **Edit** to change the profile connection properties.

To remove a profile from the list, click the **Delete** button.

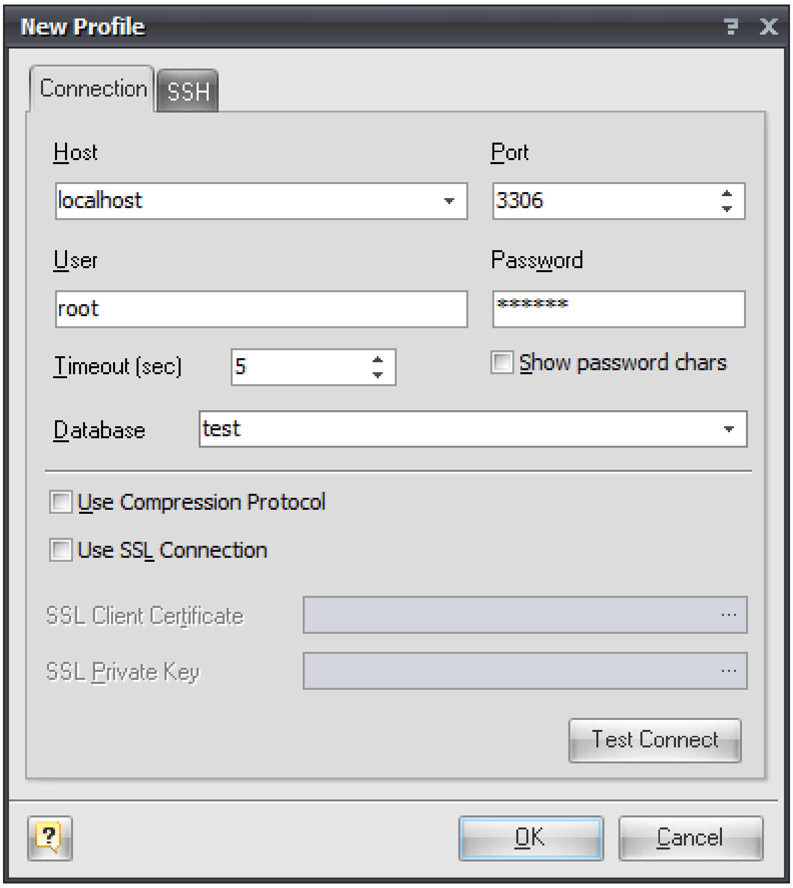
> **Please note:**
> Connection can be established with any MySQL server that works under any OS, including Windows, Linux, FreeBSD and others.

See also:
Database Accessing Tools: Connection Profile Editor | SQL Executor | Connect to a Database | Disconnect from a Database
Database Functions: Database Generation | Database Modification
Diagram Functions: Diagram Synchronization
Reverse Engineering and Import: Reverse Engineering and Import Overview | Reverse Engineering MySQL Database

10.1.1. Connection Profile Editor

The **Connection Profile Editor** allows you to set up the database connection parameters and other additional preferences. It can be launched during the process of a new profile creation and also can be used through the Database Connection Manager.

The **Profile Editor** consists of two tabs. Please see the detailed description below.



Host: localhost
Port: 3306
User: root
Password: *******
Timeout (sec): 5
□ Show password chars
Database: test
□ Use Compression Protocol
□ Use SSL Connection
SSL Client Certificate
SSL Private Key

Connection

The **Connection** tab allows you to define the MySQL server connection parameters and select the required database.

Host
Sets MySQL server address.

Port
Sets MySQL server port.

User
Sets MySQL server user login.

Password
Sets MySQL server user password.

Show password chars

This option enables showing real password chars instead of asterisks in the Password field.

Timeout
This option defines the time interval for the **Database Designer for MySQL** to connect to the MySQL database.

Database
Selects a database from the list of the ones, which are available on the server.

Use compression protocol
Compress traffic between server and client.

Use SSL Connection
Enable establishing secure SSL connection.

SSL Client Certificate
Allows to select the file containing the client certificate.

SSL Private key
Allows to select the file containing the client private key.

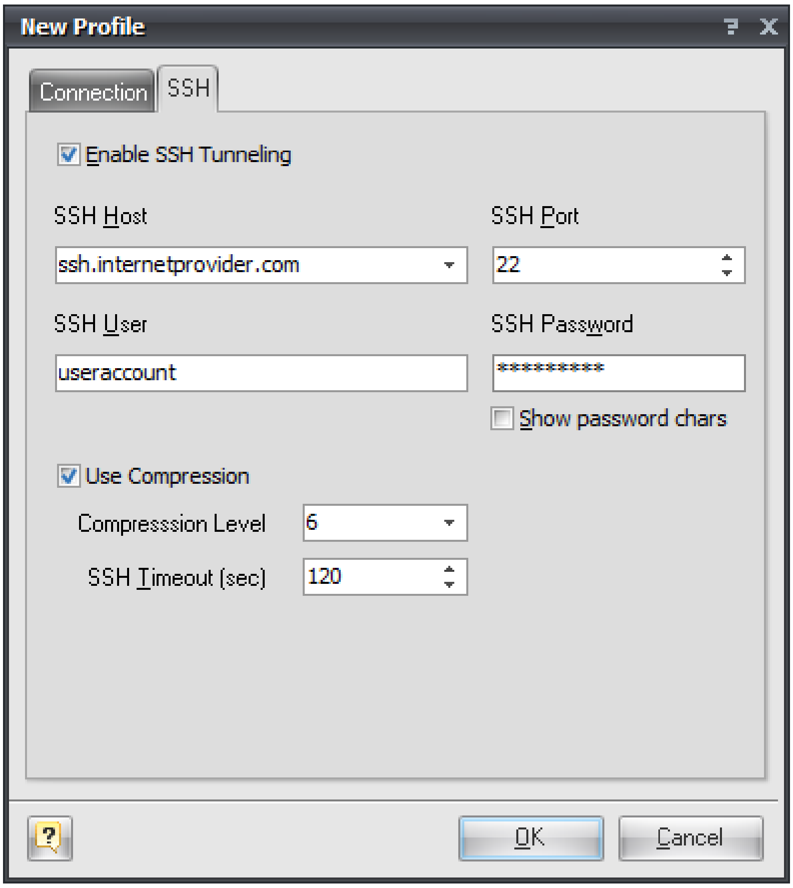
SSH

Connecting through SSH tunnel gives you two main advantages:

- You can connect to MySQL server even if the direct connections to it are not allowed. It's a typical situation on a shared hosting.
- The traffic between the server and **Database Designer for MySQL** is secured.

> **Please note:**
> Using SSH tunnel requires SSH service available on server you are connecting with.

If you set the SSH tunnel, this may request you to change the settings in the **Connection** tab: imagine that your current computer is the server you set the SSH tunnel with. So, perhaps you will have to change the value of the **Host** field to *localhost*.



☑ Enable SSH Tunneling

SSH Host: ssh.internetprovider.com

SSH Port: 22

SSH User: useraccount

SSH Password: **********

□ Show password chars

☑ Use Compression

Compression Level: 6

SSH Timeout (sec): 120

The **SSH** tab allows you to establish connection to MySQL using secure SSH tunneling. To connect to MySQL through SSH check the **Enable SSH Tunneling** option and specify the SSH connection parameters.

SSH Host
Sets SSH server address.

SSH Port
Sets SSH server port.

SSH User
Sets SSH server user login.

SSH Password
Sets SSH server user password.

Show password chars
This option enables showing real password chars in the SSH Password field instead of asterisks.

Use compression
Check this option to use the SSH compression, defined by the Compression Level option.

Compression Level
Set Compression Level to any value between 1 and 9 (1 for minimum compression, 9 for maximum compression). Typically, SSH clients use compression level 6 for optimum performance.

SSH Timeout

This option defines an interval of time that the **Database Designer for MySQL** will try to connect to the SSH host.

See also:
Database Accessing Tools: Database Connection Manager

10.2. SQL Executor

With **SQL Executor** you can send SQL queries to the connected MySQL database and display the result. The queries can contain any possible statements, e.g. UPDATE, DELETE, INSERT, SELECT statements etc. Note, that it is possible to run multiple SQL queries simultaneously.

The result of the SELECT-containing queries will be shown in the grid-based dialogues.

To execute SQL queries:

1. Connect to the database using Database Connection Manager
or
Use the already established database connection.

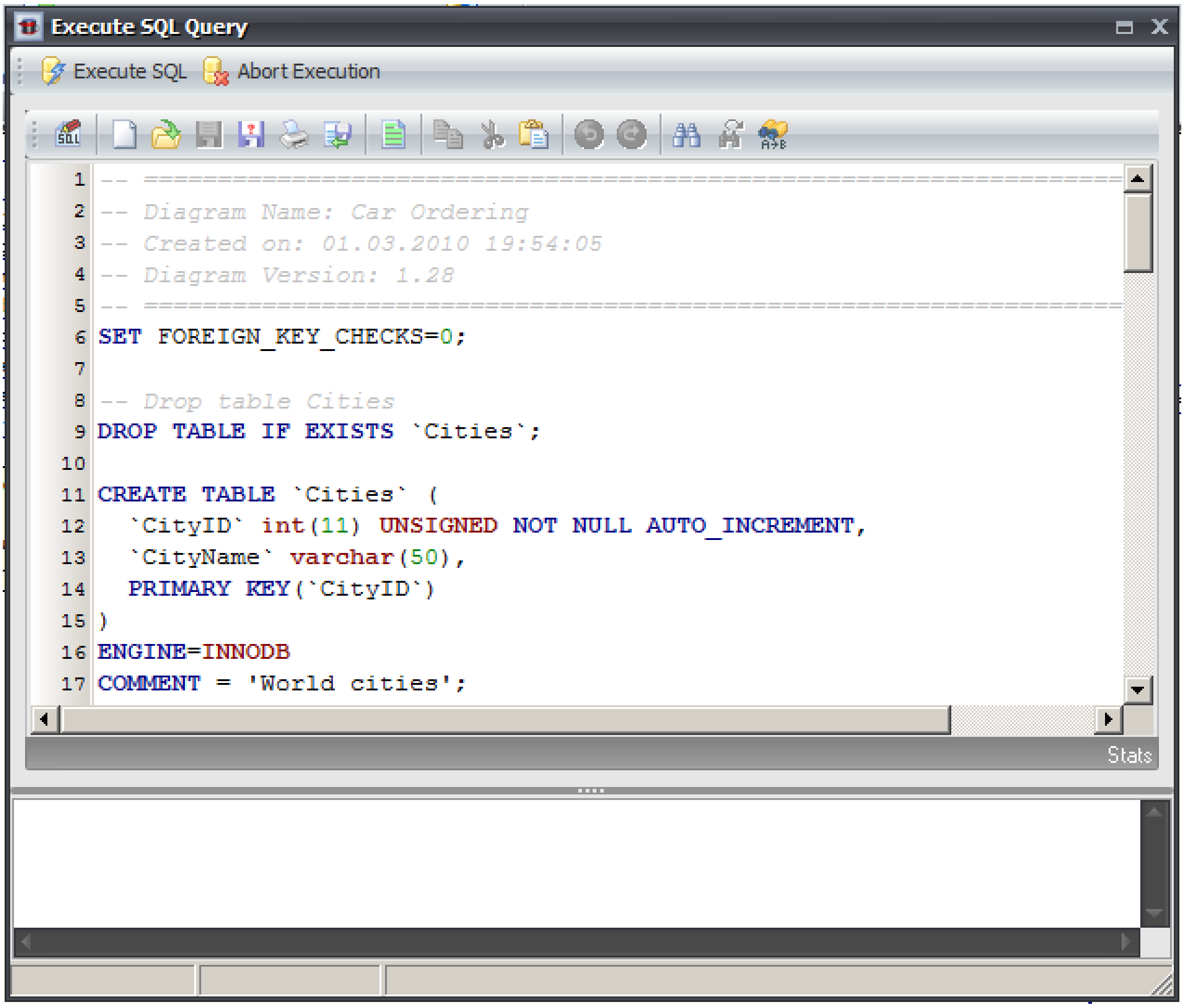
2. Call the **SQL Executor** by selecting **SQL Executor** ( ) item on **Database tab** of the Ribbon or pressing **Ctrl+Shift+E**.

3. Enter one or more SQL queries.

4. Press **F9** or click the **Execute SQL** ( ) button on the dialog window toolbar to execute your queries. A data grid window will be displayed for each SELECT-based query. To close the data grid window, click on the **Close** button. The execution status for each query will be displayed at the bottom of the **SQL Executor** dialog window.

5. You can save your queries into a file. Click on the **Save** ( ) button on the **SQL Executor** toolbar.

Please note, that **SQL Executor** dialogue window is used by Database Generation and Database Modification tools.



```
-- ======================================================================
-- Diagram Name: Car Ordering
-- Created on: 01.03.2010 19:54:05
-- Diagram Version: 1.28
-- ======================================================================
SET FOREIGN_KEY_CHECKS=0;

-- Drop table Cities
DROP TABLE IF EXISTS `Cities`;

CREATE TABLE `Cities` (
  `CityID` int(11) UNSIGNED NOT NULL AUTO_INCREMENT,
  `CityName` varchar(50),
  PRIMARY KEY(`CityID`)
)
ENGINE=INNODB
COMMENT = 'World cities';
```

See also:
Database Functions: Database Generation | Database Modification

10.3. Connect to a Database

To start working with a database using Generate Database, Database Modification, SQL Execute and Reverse Engineering functions, you have to connect to it first. To do so, please, follow these steps:

1. Select the **Connect** ( ) item on **Database tab** of the Ribbon, or press **Ctrl+Shift+N**. The Database Connection Manager will appear.

2. Select one of the defined database profiles in the list or add a new one.

3. Click **OK** to establish connection to the database.

See also:
Database Functions: Database Generation | Database Modification

Database Accessing Tools: Database Connection Manager | SQL Executor | Disconnect from a Database
Reverse Engineering and Import: Overview | Reverse Engineering MySQL Database

10.4. Disconnect from a Database

To break the current connection from the database, use the **Disconnect** ( ) item on **Database** tab of the Ribbon.

See also:
Database Accessing Tools: Database Connection Manager | SQL Executor | Connect to a Database

11. Reverse Engineering and Import

With **Database Designer for MySQL** you can extract database tables, attributes, relationships, indexes, triggers, views and other objects from existing databases.

Database Designer for MySQL supports a number of different communication methods: direct connection to MySQL database, ADO/ODBC support. This enables you import different objects from multiple databases.

Examine these import facilities:

- Reverse Engineering MySQL Database. You can connect to MySQL database directly and reverse engineer its objects.
- Import from Access Database. This tool allows you to import tables, attributes, relationships, indexes to your diagram from Microsoft Access database file.
- Universal Reverse Engineering. Use this tool to reverse engineer/import the structure of various databases. It allows you to reverse engineer the following databases: Sybase ASE and ASA, Oracle, Informix, MSSQL, DB2, DBF and many others that accessible by OLEDB/ADO or ODBC.

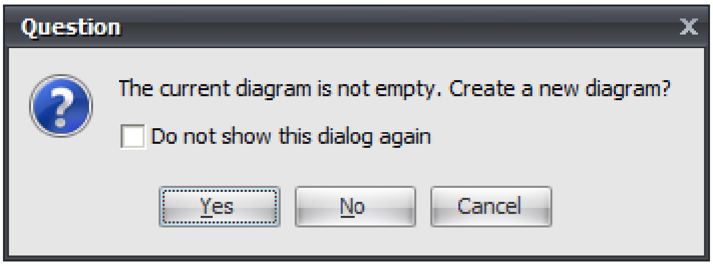
11.1. Reverse Engineering MySQL Database

With **Database Designer for MySQL** you can reverse engineer an existing MySQL database. This means that you can extract the database tables, attributes, relationships, indexes, views, stored procedures and other objects from the database to your diagram.

To reverse engineer MySQL database:

1. Select **Reverse Engineering** () item on **Database tab** of the Ribbon or press Ctrl+R.

2. If the currently opened diagram already contains some objects, a warning dialog box will appear:



The current diagram is not empty. Create a new diagram?

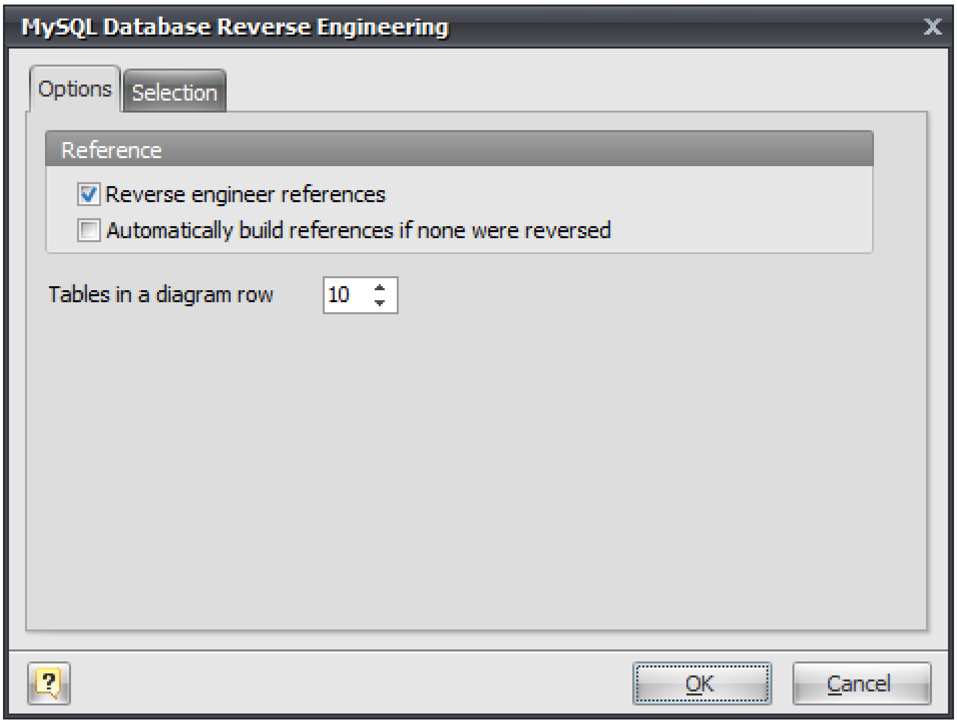
□ Do not show this dialog again

It requests you to create a new diagram to place the reversed objects into or use the currently opened diagram.

- Click **Yes** to create a new diagram or **No** to use the currently opened diagram. Click **Cancel** to cancel reverse engineering.
- Select **Do not show this dialog again** to disable future notifications.

3. If connection for the current diagram has not been established, Database Connection Manager will be shown. Select a profile from the list of the available ones or create a new profile to connect to the database you want to revere engineer.

4. The **MySQL Database Reverse Engineering tool** will be shown.



☑ Reverse engineer references

□ Automatically build references if none were reversed

Tables in a diagram row: 10

You can set the reverse engineering options in the **Options tab**.

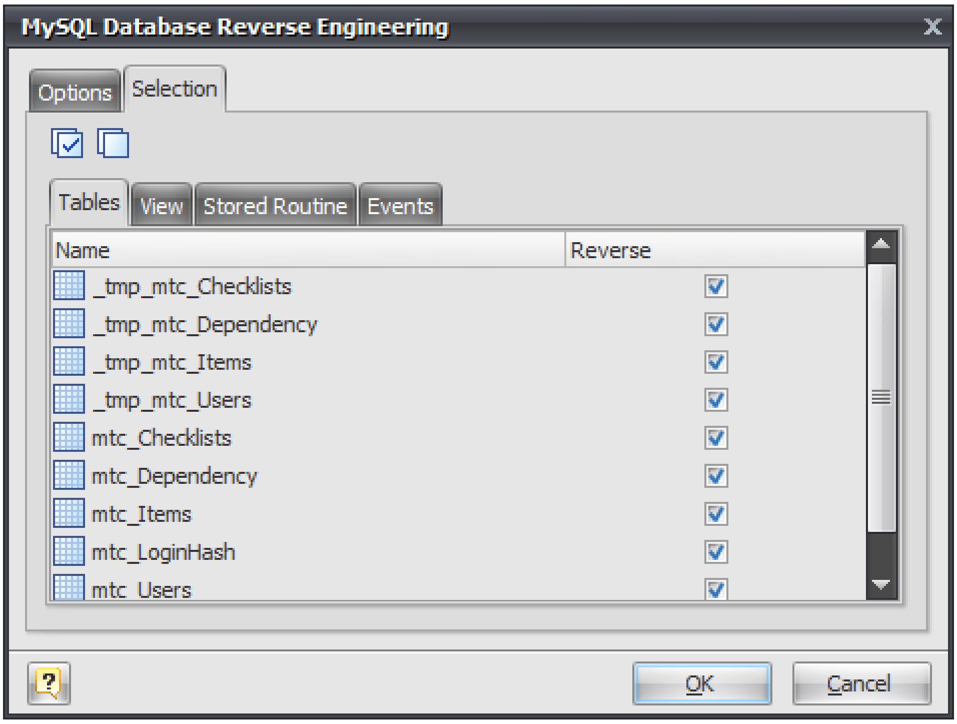
Reverse Engineer references
This option enables extracting foreign keys from the database and creating appropriate references in your diagram.

Automatically build references if none were reversed
If there are no physical references extracted, it is possible to build them from a logical structure of the database. Enabling this feature leads to automatic reconstruction of references. Such reconstruction works by the following scheme: each column of the table is being compared with all primary keys of other tables, and if the column name

and data type match one of the primary keys, a reference between the source column and the key column will be created. This option is available for modification only if Build references is checked.

Tables in a diagram row
This option determines how reversed tables will be disposed on the diagram. Reversed tables will be placed on the diagram in rows with equal distance, this option determines how many tables maximum there will be in one row.

5. In the **Selection tab** of the **MySQL Database Reverse Engineering dialog** you can choose the objects you want to reverse engineer.



MySQL Database Reverse Engineering

Options | Selection

Tables | View | Stored Routine | Events

Name	Reverse
_tmp_mtc_Checklists	☑
_tmp_mtc_Dependency	☑
_tmp_mtc_Items	☑
_tmp_mtc_Users	☑
mtc_Checklists	☑
mtc_Dependency	☑
mtc_Items	☑
mtc_LoginHash	☑
mtc_Users	☑

There are several subtabs: **Tables, Views, Stores Procedures, Events**. Each of them allows you to select appropriate database objects. Click on the checkbox near the object to enable its reverse engineering. Pay attention to the **Select All** (☑) and **Deselect All** (□) buttons on the tab, they allow you to select/deselect all objects in the list.

6. Click **OK** to start the database reverse engineering process. The **Output -> Reverse** docking window will display the state of the process.

7. The reversed database objects will be placed in your diagram.

See also:
Database Accessing Tools: Database Connection Manager
Reverse Engineering and Import: Reverse Engineering and Import Overview | Import from Access Database | Universal Reverse Engineering

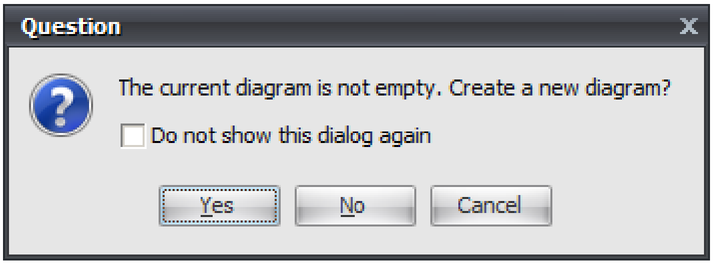
11.2. Import from Access Database

You can reverse engineer an existing **Microsoft Access** database. This means that you can extract the database tables, attributes, relationships, indexes and other objects from Microsoft Access database file to your diagram.

To reverse engineer **Microsoft Access** database:

1. Select **MS Access Database Reverse Engineering** () item from **Reverse Engineering** drop-down menu on **Database tab** of the Ribbon.

2. If the currently opened diagram already contains some objects, a warning dialog box will appear:



Question

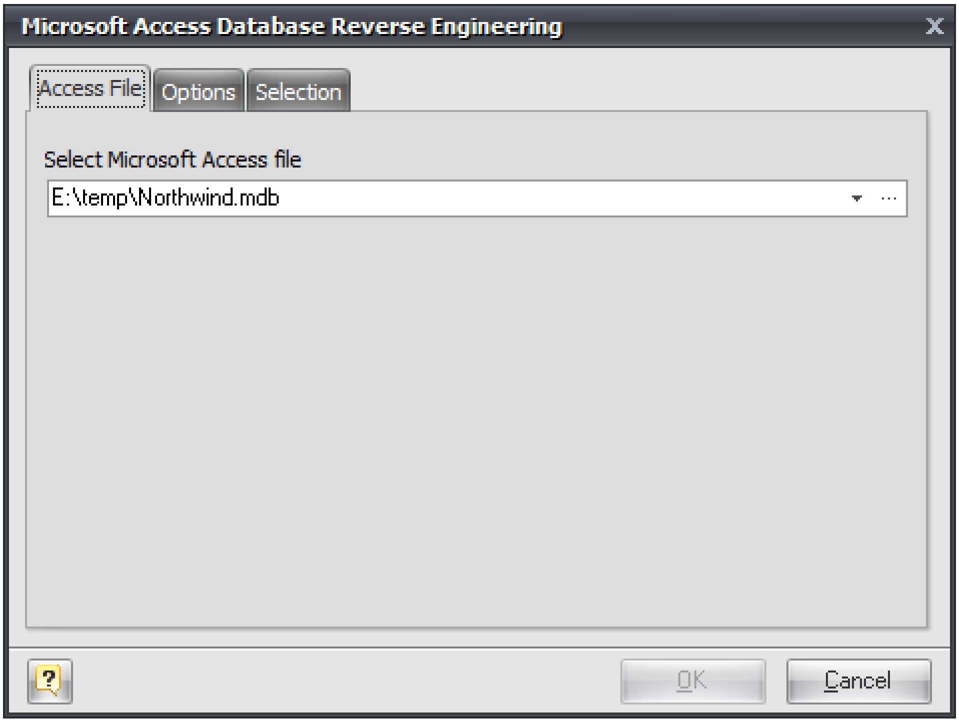
The current diagram is not empty. Create a new diagram?

□ Do not show this dialog again

It requests you to create a new diagram to place the reversed objects into or use the currently opened diagram.

Click **Yes** to create a new diagram, **No** to use the currently opened diagram. **Cancel** to cancel reverse engineering. Select **Do not show this dialog again** to disable future notifications.

3. The **Microsoft Access Reverse Engineering** tool will be shown.

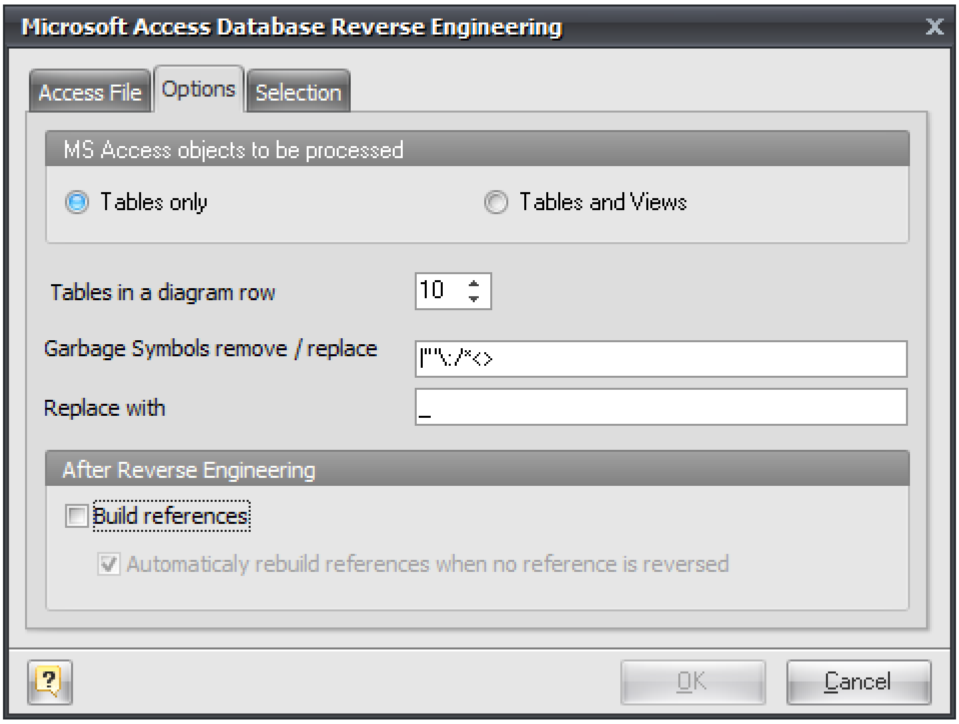


Microsoft Access Database Reverse Engineering

Select Microsoft Access file: E:\temp\Northwind.mdb

First, type in the full path to **Microsoft Access database** in the **Access File** tab or use "**...**"-button to browse file.

4. You can set the reverse engineering options in the **Options tab**.



MS Access objects to be processed

● Tables only ○ Tables and Views

Tables in a diagram row: 10

Garbage Symbols remove / replace: |'"\:/*<>

Replace with: _

After Reverse Engineering

☐ Build references

☑ Automaticaly rebuild references when no reference is reversed

Tables only
Reverse engineer only tables ignoring views.

Tables and Views
Reverse engineer both tables and views.

Tables in a diagram row
This option determines how reversed tables will be disposed on the diagram. Reversed tables will be placed on the diagram in rows with equal distance, this option determines how many tables maximum there will be in one row.

Garbage symbols remove/replace
Defines the symbols that will be replaced in the names of the objects.

Replace with
Defines the garbage replacement symbol.

Build references
This option enables extracting foreign keys from the database and creating appropriate references in your diagram.

Automatically rebuild references when no reference is reversed
If there are no physical references extracted, it is possible to build them from logical structure of the database.

Enabling this feature leads to automatic reconstruction of references. Such reconstruction works by the following scheme: each column of the table is being compared with all primary keys of other tables, and if the column name and data type match one of the primary keys, a reference between the source column and the key column will be created.

This option is available for modification only if Build references is checked.

5. In the **Selection tab** of the **Access Reverse Engineering** tool you can choose the tables you want to reverse engineer.

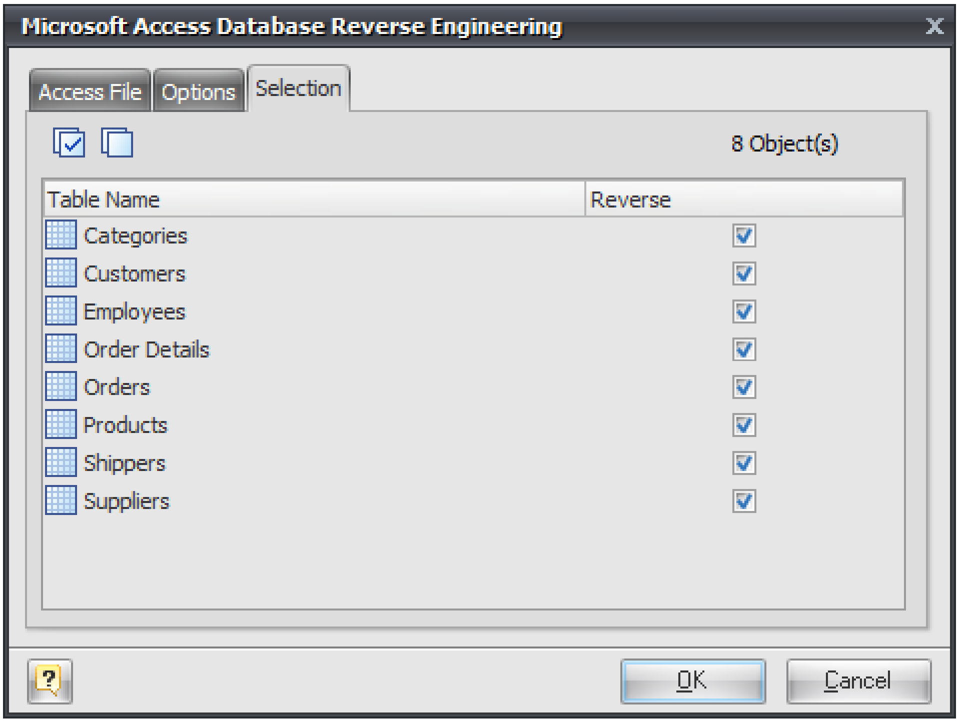


Table Name	Reverse
Categories	☑
Customers	☑
Employees	☑
Order Details	☑
Orders	☑
Products	☑
Shippers	☑
Suppliers	☑

Click on the checkbox near the table to enable its reverse engineering. Pay attention to the **Select All** (☑) and **Deselect All** (□) buttons on the tab, they allow you to select/deselect all tables in the list.

5. Click **OK** to start the database reverse engineering process. The **Output -> Reverse** docking window will display the state of the process.

6. The reversed database objects will be placed on your diagram.

See also:
Reverse Engineering and Import: Reverse Engineering and Import Overview | Reverse Engineering MySQL Database | Universal Reverse Engineering

11.3. Universal Reverse Engineering (ADO, OLEDB, ODBC)

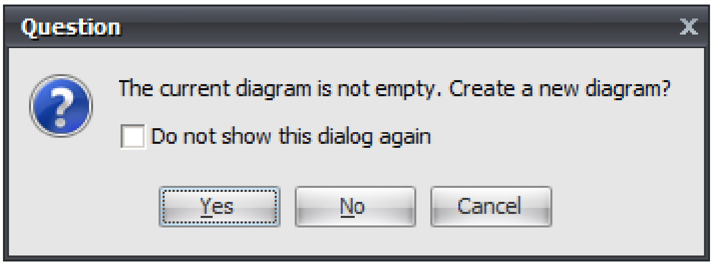
You can reverse engineer a number of databases, such as Sybase ASA and ASE, Oracle, PostgreSQL, Informix, MSSQL and others. This means that you can extract tables, attributes, relationships, indexes and other objects. It is possible with the **Universal Reverse Engineering** tool, which connects to a number of databases through **OLEDB** (**ADO**) or **ODBC** link. So, to reverse engineer a particular database, you need a corresponding **OLEDB** provider, or **ODBC** driver. In most cases, such libraries are installed in the system with the help of database client applications.

To reverse engineer a database through **OLEDB** or **ODBC** link:

1. Select **Universal Reverse Engineering** item from **Reverse Engineering** drop-down menu on

Database tab of the Ribbon.

2. If the currently opened diagram already contains some objects, the following dialog box will appear:



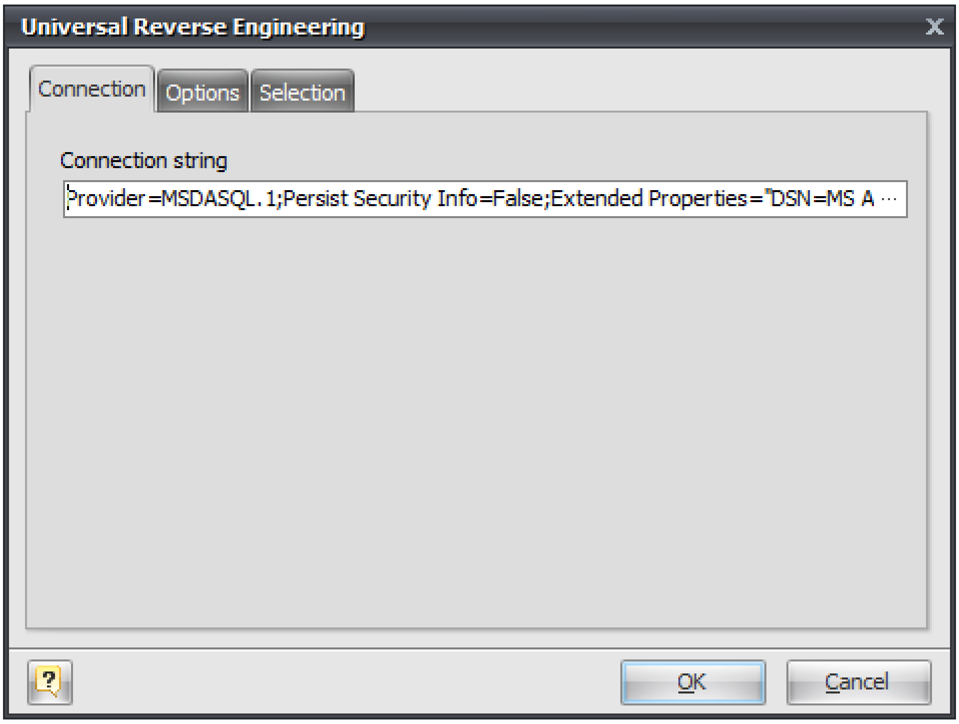
Question

The current diagram is not empty. Create a new diagram?

□ Do not show this dialog again

It requests you to create a new diagram to place the reversed objects into or use the currently opened diagram. Click **Yes** to create a new diagram, **No** to use the currently opened diagram, or **Cancel** to cancel reverse engineering. Select **Do not show this dialog again** to disable future notifications.

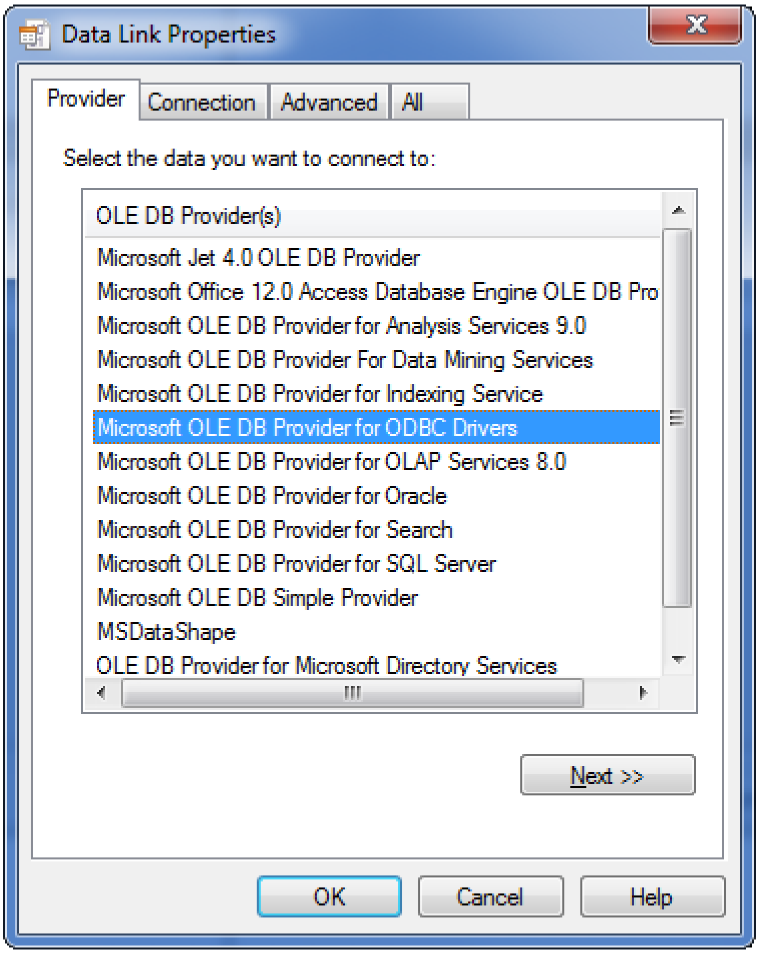
3. The **Universal Reverse Engineering** tool will be shown. First, provide full OLEDB connection string in the **Connection tab**.



Universal Reverse Engineering

Connection string: Provider=MSDASQL.1;Persist Security Info=False;Extended Properties="DSN=MS A

Click on the "**...**" button to call the standard system **Data Link Properties** dialog window, which helps you to build OLEDB connection string:

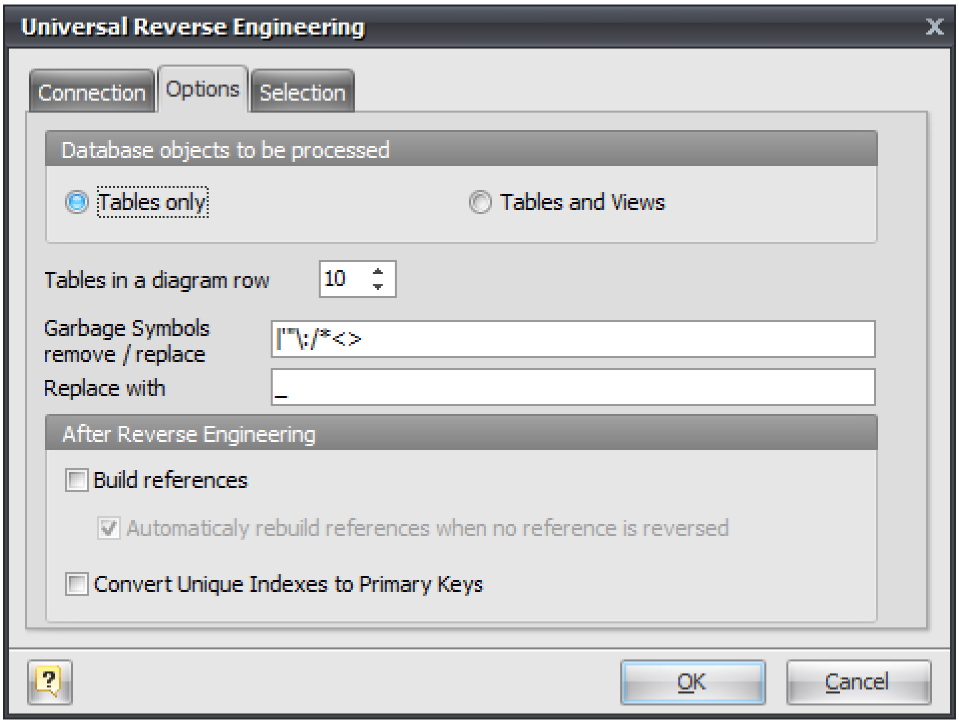


Select the data you want to connect to:

OLE DB Provider(s)
Microsoft Jet 4.0 OLE DB Provider
Microsoft Office 12.0 Access Database Engine OLE DB Pro
Microsoft OLE DB Provider for Analysis Services 9.0
Microsoft OLE DB Provider For Data Mining Services
Microsoft OLE DB Provider for Indexing Service
Microsoft OLE DB Provider for ODBC Drivers
Microsoft OLE DB Provider for OLAP Services 8.0
Microsoft OLE DB Provider for Oracle
Microsoft OLE DB Provider for Search
Microsoft OLE DB Provider for SQL Server
Microsoft OLE DB Simple Provider
MSDataShape
OLE DB Provider for Microsoft Directory Services

To use ODBC drivers, click on the **Microsoft OLE DB Provider for ODBC Drivers**. To use the native OLEDB provider, click on the appropriate item in the list. Browse through tabs to set other tabs and connection properties. Click on the **Help** button in the bottom of the dialog window to learn more about OLEDB connections. Click on the **OK** button to store the data link properties in the **Connection string** field of the **Universal Reverse Engineering** dialog.

4. You can set reverse engineering options in the **Options** tab:



Database objects to be processed

● Tables only ○ Tables and Views

Tables in a diagram row: 10

Garbage Symbols remove / replace: |'"\:/*<>

Replace with: _

After Reverse Engineering

□ Build references

☑ Automaticaly rebuild references when no reference is reversed

□ Convert Unique Indexes to Primary Keys

Tables only
Reverse engineer only tables ignoring views.

Tables and Views
Reverse engineer both tables and views.

Tables in a diagram row
This option determines how reversed tables will be disposed on the diagram. Reversed tables will be placed on the diagram in rows with equal distance, this option determines how many tables maximum there will be in one row.

Garbage symbols remove/replace
Defines the symbols that will be replaced in the names of the objects.

Replace with
Defines the garbage replacement symbol.

Build references
This option enables extracting foreign keys from the database and creating appropriate references in your diagram.

Automatically rebuild references when no reference is reversed
If there are no physical references extracted, it is possible to build them from logical structure of the database.

Enabling this feature leads to automatic reconstruction of references. Such reconstruction works by the following scheme: each column of the table is being compared with all primary keys of other tables, and if the column name and data type match one of the primary keys, a reference between the source column and the key column will be created.

This option is available for modification only if Build references is checked.

Convert Unique Indexes to Primary Keys
There no primary keys in some databases, but unique indexes used instead. You can use this optin to convert such unique indexes to primary keys.

5. In the **Selection tab** of the **Universal Reverse Engineering** tool you can choose the tables you want to reverse engineer.

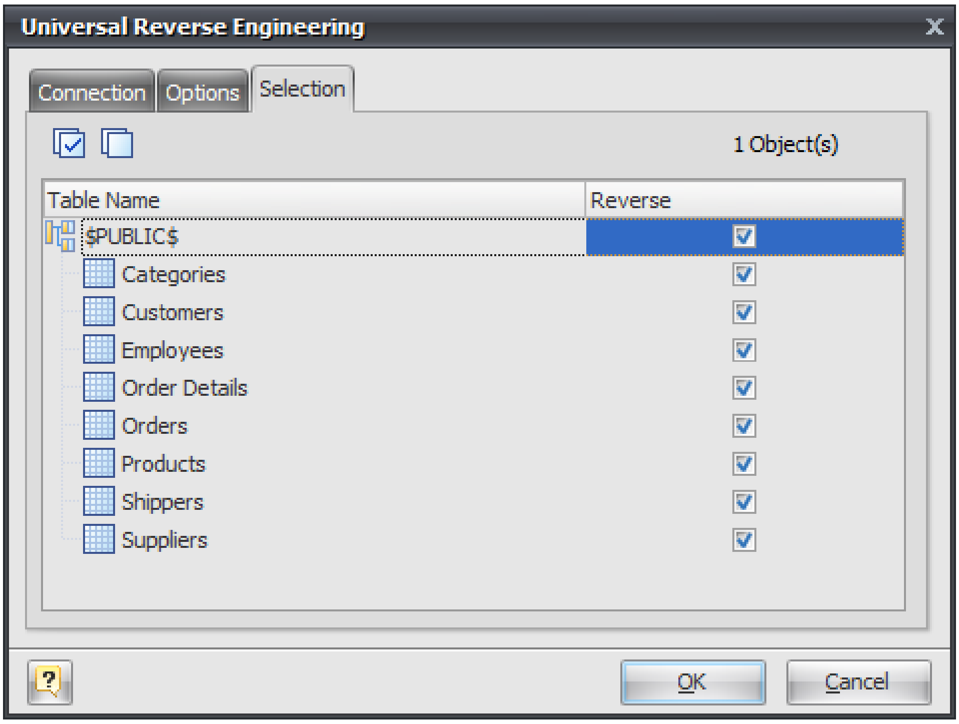


Table Name	Reverse
$PUBLIC$	☑
Categories	☑
Customers	☑
Employees	☑
Order Details	☑
Orders	☑
Products	☑
Shippers	☑
Suppliers	☑

Click on the checkbox near the table to enable its reverse engineering. Pay attention to the **Select All** (☑) and **Deselect All** (□) buttons on the tab, they allow you to select/deselect all tables in the list.

6. Click **OK** to start the database reverse engineering process. The **Output -> Reverse** docking window will display the state of the process.

7. The reversed database objects will be placed in your diagram.

See also:
Reverse Engineering and Import: Reverse Engineering and Import Overview | Reverse Engineering MySQL Database | Import from Access Database

12. Reports

Database Designer for MySQL can generate comprehensive printable reports for you. This gives you the possibility to get a hard copy of the report by printing it out. After that you get the opportunity to have it approved by your colleagues and managers. Generated reports contain the information about all tables, their indexes, columns and references, other database objects.

Please note:
Sometimes users ask for exporting reports to PDF format. **Database Designer for MySQL** does not support direct reports export to PDF format. But you can use free PDFCreator (http://sourceforge.net/projects/pdfcreator/) tool to convert our HTML-based report to PDF format.

See also:
Reports: Create a Report

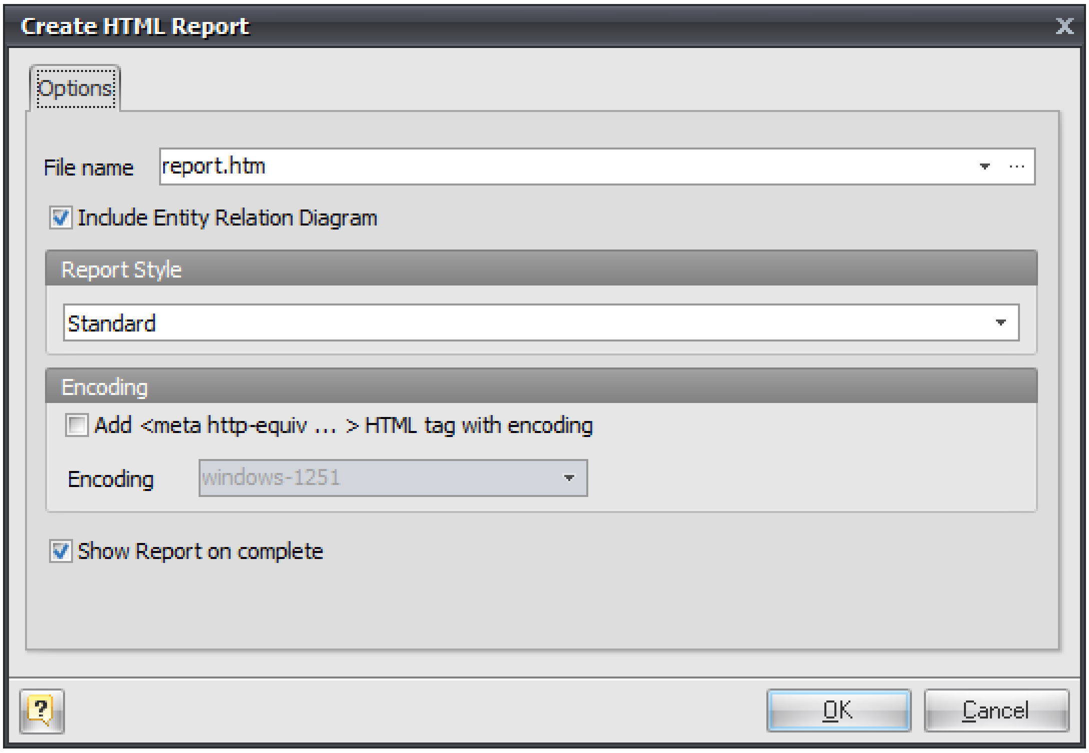
12.1. Create a Report

You can generate an HTML report, which will contain detailed information about the objects placed in the diagram, and diagram picture.

To generate an HTML report, follow these steps:

1. Select **Create Report** ( ) item on **Diagram tab** of Ribbon to call the **Create HTML Report** tool.

2. The report generation dialog window will appear.



Create HTML Report

Options

File name: report.htm

☑ Include Entity Relation Diagram

Report Style

Standard

Encoding

☐ Add <meta http-equiv ... > HTML tag with encoding

Encoding: windows-1251

☑ Show Report on complete

Please, examine the dialog window controls:

File name
Enter the name for the file, in which you want to store the contents of the generated report.

Include Entity Relation Diagram
Deselect this option if do not want to include image with whole diagram in report. This can be useful if you have really big diagram that can't fit to your page size. You can

always print diagram separately using Print Diagram tool.

Report style
Select the style of the report from this drop down menu. A style applies graphical environment to the report, such as font color, background, etc.

Show report on complete
Click on this option to open an HTML report in a default browser window after the generation.

Click **OK** to generate the report.

See also:
Reports: Overview
Printing a Diagram: Overview

13. Printing a Diagram

Printing a database diagram gives you a picture of your database structure to refer to or distribute.

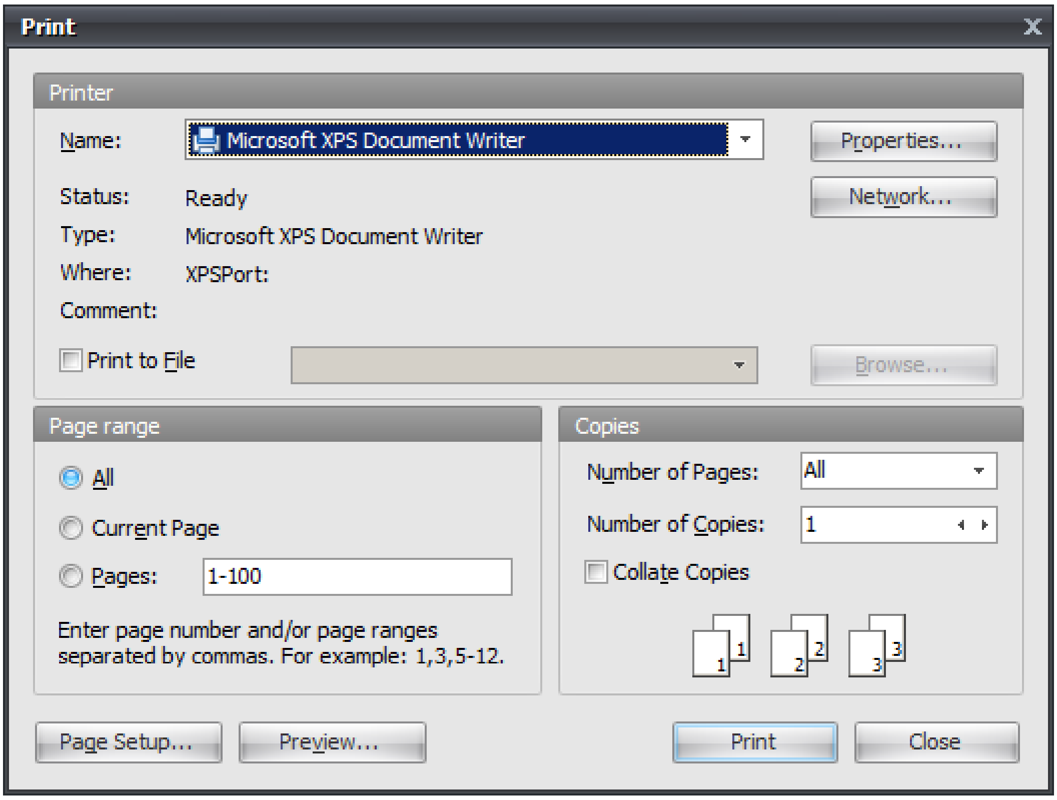
Before you start printing, arrange the objects in the database diagram to your satisfaction. You can change the shape, and position of the objects in the diagram without affecting their definitions in the database.

To change the layout of the diagram, move, size, and shape the objects. For example, you can use the mouse to move tables, or use the Auto Layout Diagram command to automatically reposition the objects.

You can change physical parameters of your paper, on which diagram will be printed out, specify margins, page headers and footers information using Page Setup dialog.

Before printing out a diagram, you can see how it will look paper, use the Print Preview tool for that.

To print out a diagram, select the **Print** (🖨) menu item from Application menu or press **Ctrl+P**.



Name: Microsoft XPS Document Writer

Status: Ready

Type: Microsoft XPS Document Writer

Where: XPSPort:

Comment:

□ Print to File

● All

○ Current Page

○ Pages: 1-100

Enter page number and/or page ranges separated by commas. For example: 1,3,5-12.

Number of Pages: All

Number of Copies: 1

□ Collate Copies

The **Print Setup** dialog is displayed. In the **Print Setup** dialog select the printer you want to print on, and then click the **OK** button to print a diagram.

See also:
Printing a Diagram: Page Setup | Print Preview

13.1. Page Setup

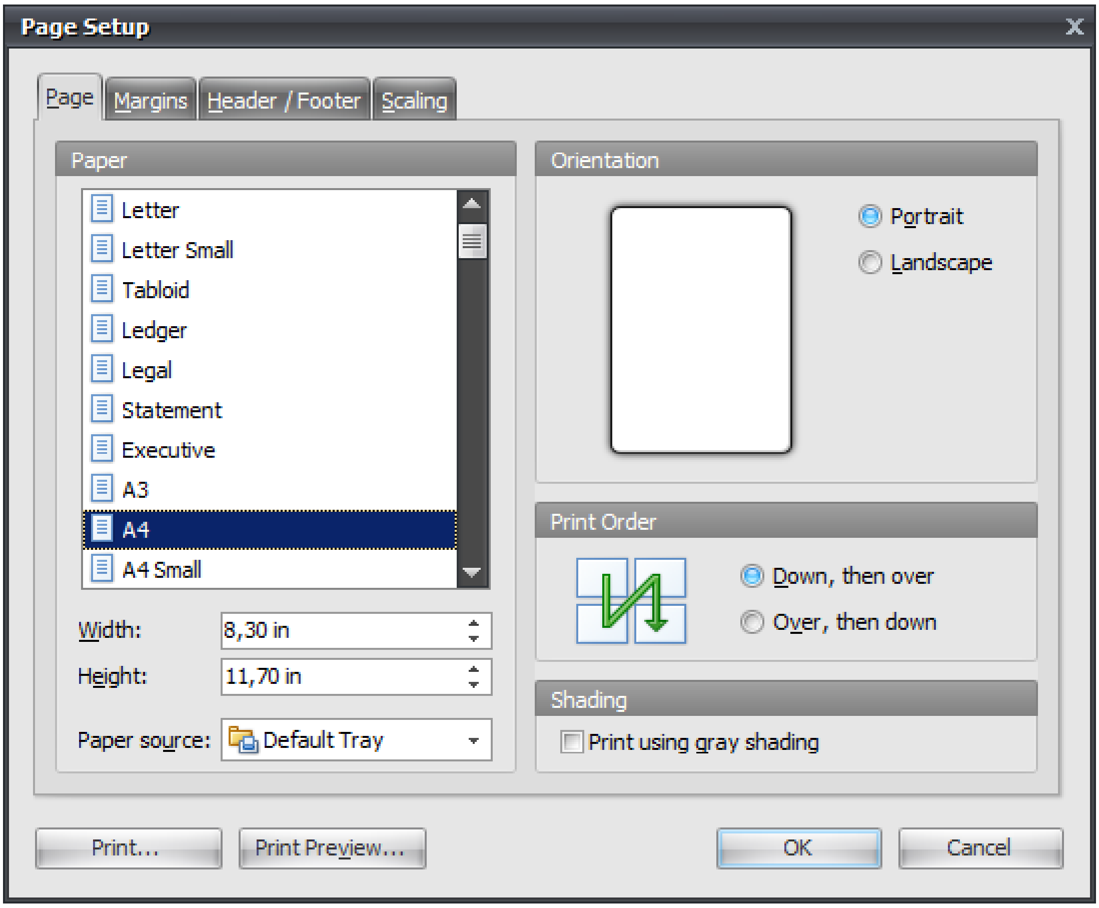
The **Page Setup** dialog window allows you to specify printing parameters, select the paper size and page orientation.

To call the **Page Setup** dialog window, select the **Page Setup** (🖨) menu item from Application Menu.

The **Page Setup** dialog window consists of several tabs, each of which is described in detail below.

Page

This tab allows you to specify page parameters.


Page Setup

Page | Margins | Header / Footer | Scaling

Paper

Letter
Letter Small
Tabloid
Ledger
Legal
Statement
Executive
A3
A4
A4 Small

Width: 8,30 in

Height: 11,70 in

Paper source: Default Tray

Orientation

● Portrait

○ Landscape

Print Order

● Down, then over

○ Over, then down

Shading

□ Print using gray shading

Paper Type
Allows you to choose one of the predefined paper types.

Dimension Width & Height
Allows you to specify non-standard dimensions for output.

Paper Source
Available paper sources (printer specific).

Orientation Portrait
Vertical page orientation.

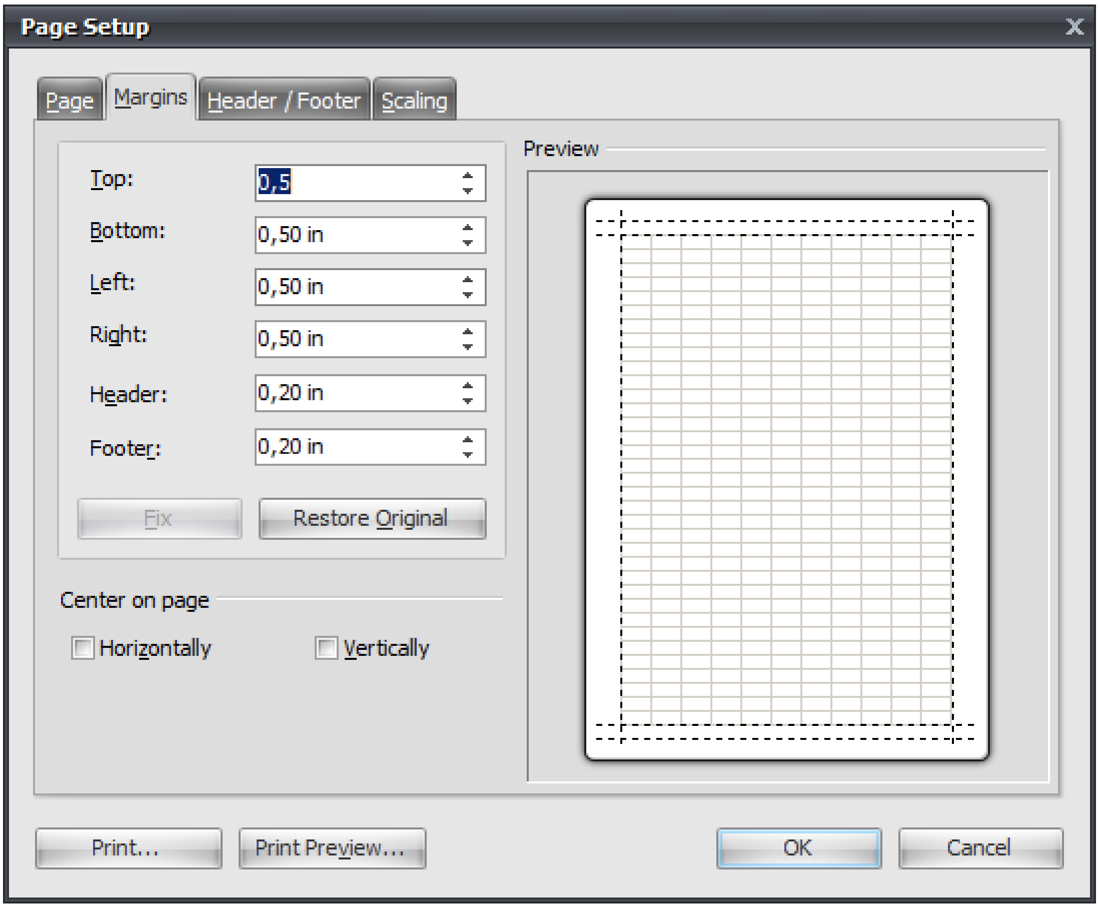
Orientation Landscape
Horizontal page orientation.

Print Order
Specifies the order in which pages will be printed.

Shading
Print using gray shading.

Margins

This tab allows you to set the margins and distances.



Top: 0,5

Bottom: 0,50 in

Left: 0,50 in

Right: 0,50 in

Header: 0,20 in

Footer: 0,20 in

Center on page

□ Horizontally □ Vertically

Top
Adjusts the amount of space that appears between the top of the page and the top of the header.

Header
Sets the height of the header.

Left
Adjusts the whitespace that appears between the left edge of the page and the left edge of the diagram.

Right
Adjusts the whitespace that appears between the right edge of the page and the right edge of the diagram.

Bottom
Adjusts the whitespace that appears between the bottom of the page and the bottom of the footer.

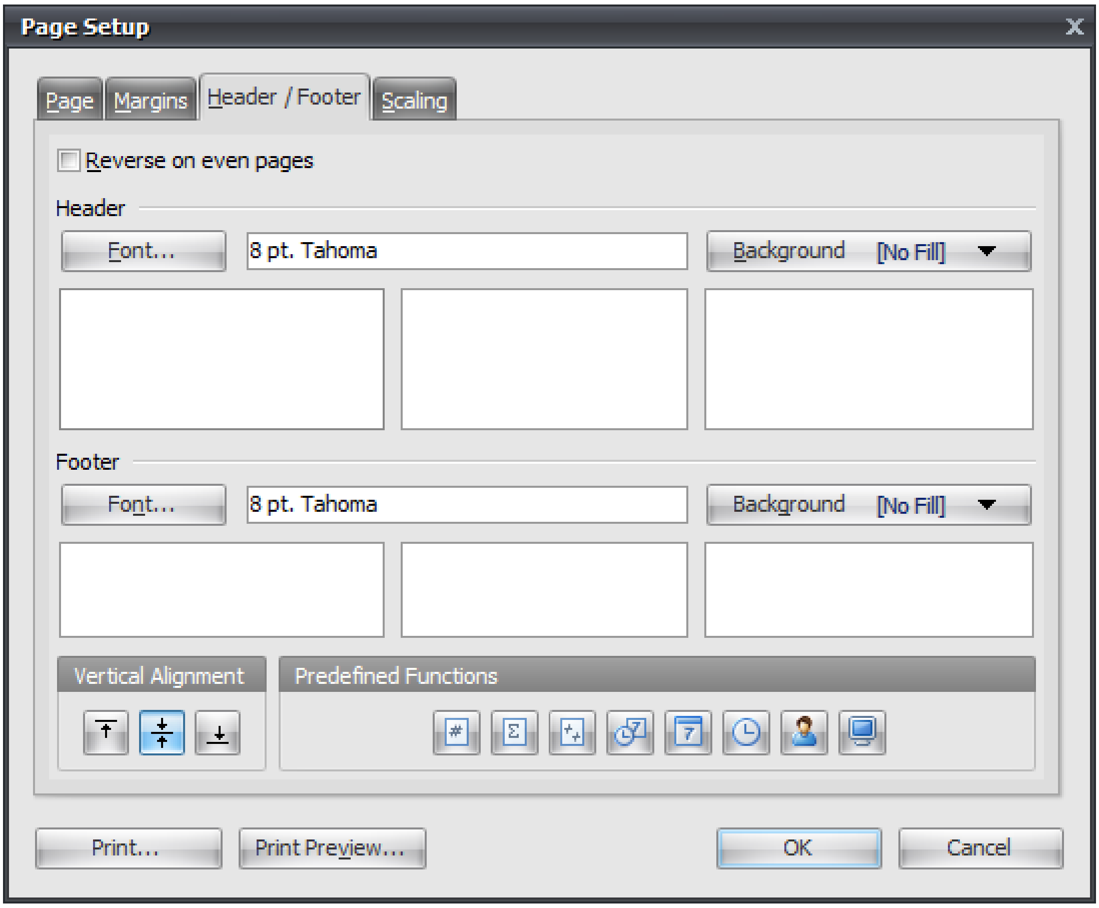
Footer
Sets the height of the footer.

Horizontally
Enable this option to align the content of the page to the center horizontally.

Vertically
Enable this option to align the content of the page to the center vertically.

Header/Footer

This tab allows you to specify the content of the header and footer, which will be displayed on each printed page.



Page Setup

Page | Margins | Header / Footer | Scaling

□ Reverse on even pages

Header

Font... | 8 pt. Tahoma | Background [No Fill]

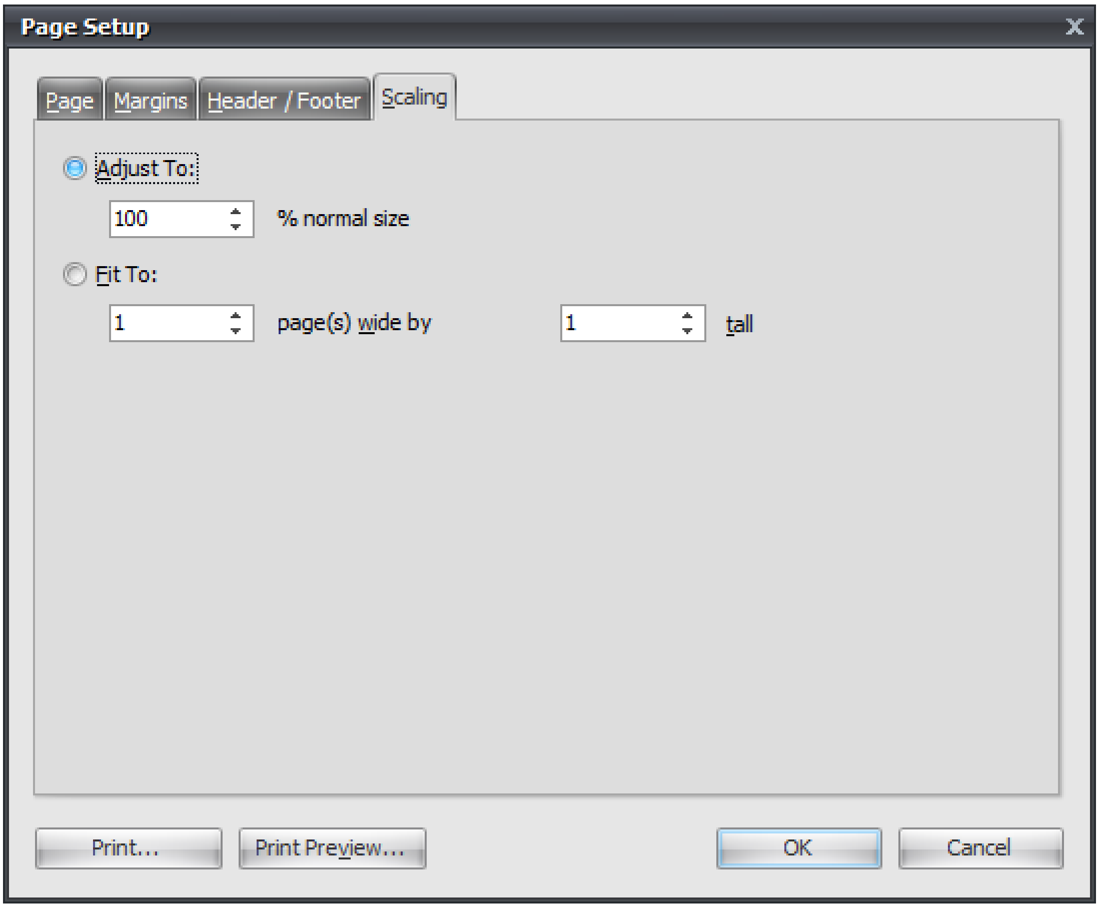
Footer

Font... | 8 pt. Tahoma | Background [No Fill]

You can create the content for each header and footer using a list of predefined functions.

Scaling

This tab allows you to specify scale factors for printing.



● Adjust To:
100 % normal size
○ Fit To:
1 page(s) wide by 1 tall

Adjust To:
Set this value to print source page with necessary percentage.

Fit To:
Set these values to print several source pages to one physical paper page.

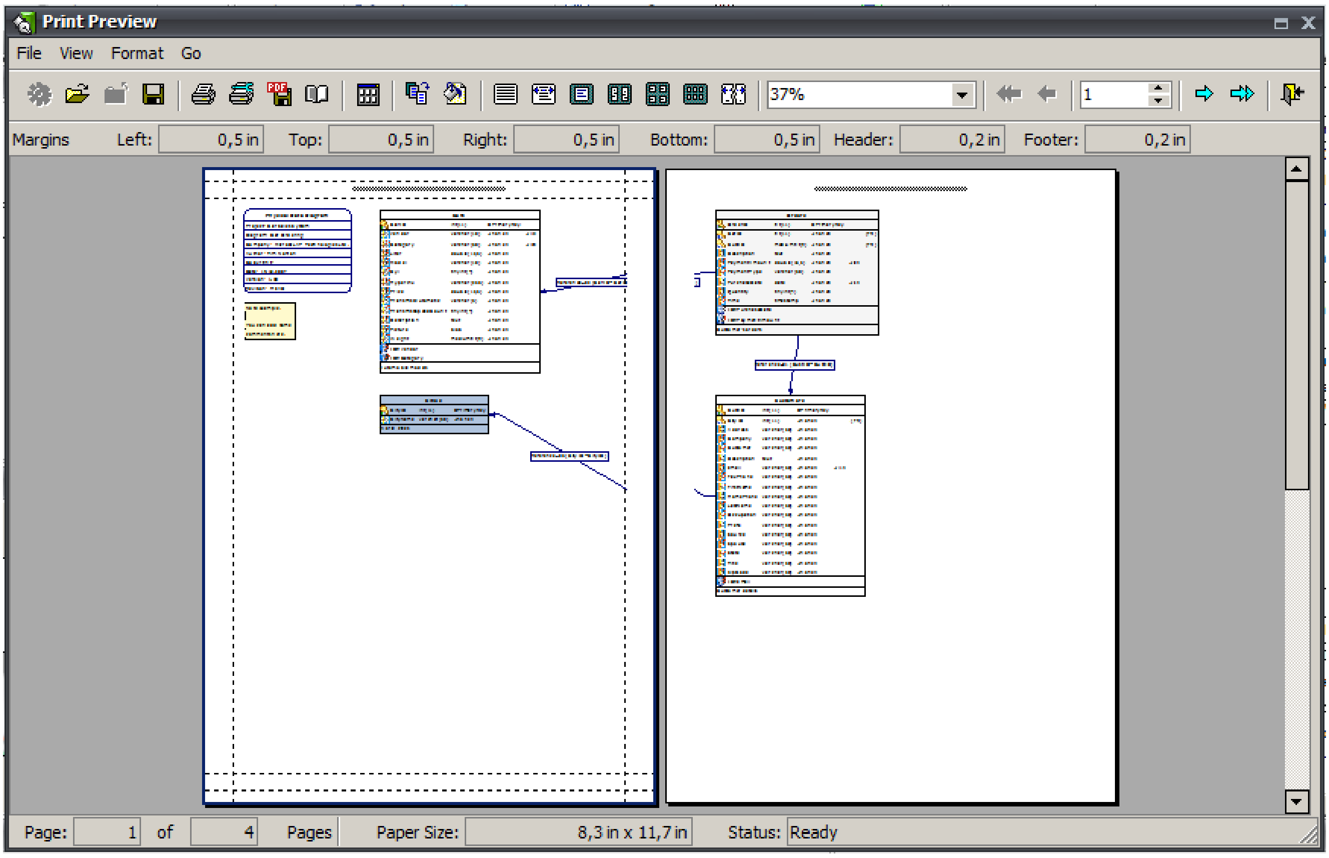
See also:
Printing a Diagram: Overview | Print Preview

13.2. Print Preview

It is often necessary to preview a diagram before printing it out. It is possible with the **Print Preview** window.

To call it, use the **Print Preview** ( ) menu item from Application Menu.

The following picture demonstrates how the **Print Preview** window looks like:



Margins Left: 0,5 in Top: 0,5 in Right: 0,5 in Bottom: 0,5 in Header: 0,2 in Footer: 0,2 in

Page: 1 of 4 Pages Paper Size: 8,3 in x 11,7 in Status: Ready

See also:
Printing a Diagram: Overview | Page Setup

14. How to...

How to

How to Connect to a Database

How to Disconnect from a Database

How to Create a New Diagram

How to Create a Table

How to Create a Reference and Foreign Key

How to Check a Diagram

How to Edit Columns

How to Edit Table Index

How to Generate a Database

How to Modify a Database

How to Import from a MySQL Database

How to Import from a Microsoft Access Database

How to Import from other Databases

How to Modify Multiple Tables

How to Execute an SQL Script

How to Merge Diagrams

How to Find Errors in a Diagram

How to Find Objects

How to Print a Diagram

How to Read a Diagram

How to View an SQL Table Definition

How to Export a Diagram to Graphics

How to Create a Domain

How to Insert a Comment

How to Change Table Formatting

> Please let us know if you need any other **How to...** topics.

14.1. How to Connect to a Database

To start working with a database using Generate Database, Database Modification, SQL Execute and Reverse Engineering functions, you have to connect to it first. To do so, please, follow these steps:

1. Select the **Connect** ( ) item on **Database tab** of the Ribbon, or press **Ctrl+Shift+N**. The Database Connection Manager will appear.

2. Select one of the defined database profiles in the list or add a new one.

3. Click **OK** to establish connection to the database.

See also:
Database Functions: Database Generation | Database Modification
Database Accessing Tools: Database Connection Manager | SQL Executor | Disconnect from a Database
Reverse Engineering and Import: Overview | Reverse Engineering MySQL Database

14.2. How to Disconnect from a Database

To break the current connection from the database, use the **Disconnect** ( ) item on **Database** tab of the Ribbon.

See also:
Database Accessing Tools: Database Connection Manager | SQL Executor | Connect to a Database

14.3. How to Create a New Diagram

The first step to use **Database Designer for MySQL** is to create a new diagram.

To create a new diagram select the **File | New** menu item or press **Ctrl-N**. You can also use the **New diagram** button on the **Standard** toolbar.

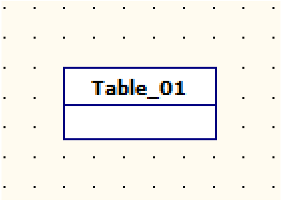
A new diagram will have a default name like "*Noname1*". You can change the name of your diagram in the Diagram Properties dialog window.

See also:
Diagram: Diagram Properties
Interface User Guide: Ribbon

14.4. How to Create a Table

To create a new table in a diagram:

1. Click on the Table ( ) icon on the Main tab of ribbon. Your mouse cursor will change its appearance. Click on the diagram area to create a new table. Or right click on the empty space of diagram and choose Create Object | Table. An empty table will appear in the diagram:



2. (optional) Double click on the new table symbol in the diagram to display the Table Editor dialog window.

3. (optional) Enter the table name in the Table name field.

4. (optional) Click **OK** to save the changes.

New table will also appear on Minimap Navigator.

See also:
Interface User Guide: Minimap Navigator
Diagram Objects: Table Editor

14.5. How to Create a Reference and Foreign Key

To create a reference between tables:

1. Click the **Reference** () item on the **Main** tab of Ribbon. Your mouse cursor will change its appearance.

2. Click on the table (referencing table or child table) that will have a foreign key.

3. Then click on the second table (referenced table or parent table) whose constraint (e.g. primary key) will be referenced by the new foreign key.

4. The **Joins** tab of the Reference Editor will be shown. You can choose columns of a referenced (parent) and referencing (child) table participated in the reference. The properties of auto column will be copied from the respective column of the referenced table.

> **Please note:**
> To create a self reference (that links columns at the same table), click the Reference () item on the Main tab of Ribbon. Then click the same table two times. To create N:M references, please refer to Creating a Many-to-Many Reference.

> **Please note:**
> By default **Database Designer for MySQL** draws references automatically using shortest line between table shapes on diagram with reference's label placed in the center of this line. Sometimes this can lead to references crossing, or reference's label can fit below table shape. Starting from version 2.0 you can route references manually to avoid this. Take a look at Manual Reference Drawing topic.

Reference Creation in Detail

On reference creation, **Database Designer for MySQL** performs the following actions:

1. Creates new column(s) in the referencing table, their parameters (name, data type) will be copied from the primary key constraint of the referenced table. If the referencing table already has column(s) analogous to the primary key(s) of the referenced table, this column(s) will be used as foreign-key column(s).

2. If there is no primary key or unique constraints in the referenced table, a standard primary key column will be created in the referenced table.

3. Creates foreign key constraint in the referencing table that refers to the referenced table primary key(s).

> **Please note:**
> Starting with MySQL 4.1.2 such foreign key constraints for references are created automatically by MySQL.So you can disable FK-indexes creation in "Default Database Options" tab of Diagram Properties dialog.

See the Diagram Display Preferences topic to find out more about default **References** options, actions, etc.

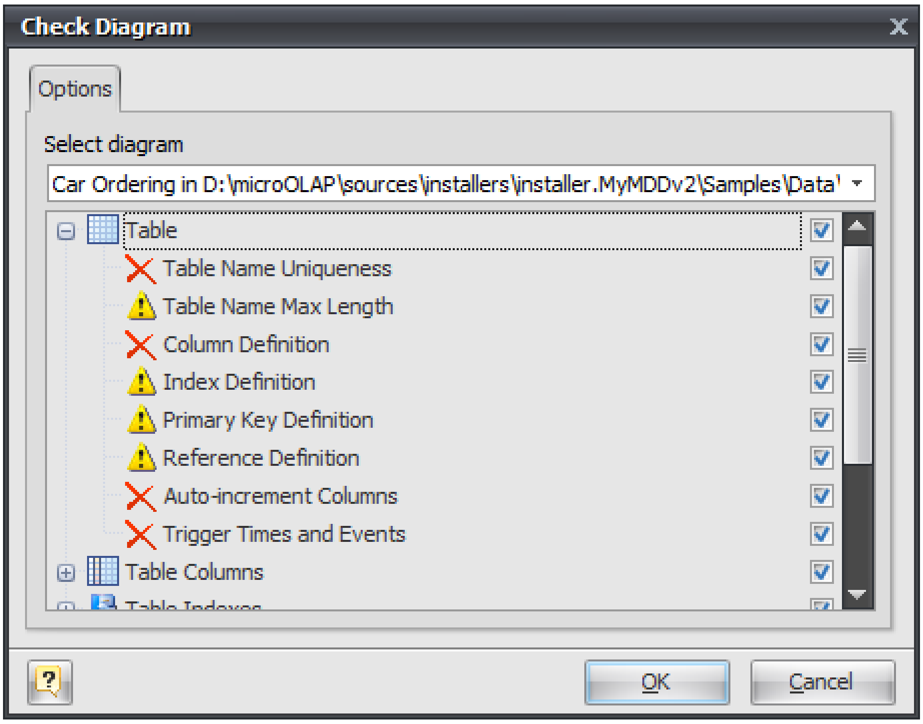
See the Notation topic to find out more about reference symbol on the diagram.

See also:
Diagram: Notation | "Default Database Options" tab | Diagram Properties
Diagram Objects: Reference Editor | Creating a Many-to-Many Reference

14.6. How to Check a Diagram

The **Check Diagram** tool of the **Database Designer for MySQL** allows you to check your database diagram for most typical errors and defects. The result of the check goes in a well structured form, using which you can easily bring your diagram to correspondence with the common standards of database modeling.

To open the **Check Diagram** dialog use the **Check Diagram** ( ) item on **Diagram** tab of the Ribbon or press **F4**.



Select diagram
Car Ordering in D:\microOLAP\sources\installers\installer.MyMDDv2\Samples\Data'

- Table ☑
  - Table Name Uniqueness ☑
  - Table Name Max Length ☑
  - Column Definition ☑
  - Index Definition ☑
  - Primary Key Definition ☑
  - Reference Definition ☑
  - Auto-increment Columns ☑
  - Trigger Times and Events ☑
- Table Columns ☑
- Table Indexes ☑

Use the **Select diagram** drop-down list to select the diagram from the list of currently opened diagrams. Current opened diagram is selected by default.

The tree list below allows you to select what warnings and errors should be taken into account during the check. All warnings and errors are divided into categories, which correspond to the diagram objects. Remove selection from the warning/error or from the whole category to exclude it from the check.

These are the descriptions for all available warnings and errors.

Table

Error "Table Name Uniqueness"
Check the diagram for the uniqueness of each table name within a diagram.

Warning "Table Name Max Length"
MySQL allows only 64 characters in table names and cuts names if they are longer than this.

Error "Column Definition"
Check if each table within the diagram owns at least one column.

Warning "Index Definition"
Check if each table within the diagram owns at least one index.

Warning "Primary Key Definition"
Check if a primary key is defined for each table within the diagram.

Warning "Reference Definition"
Check if each table within the diagram is linked with other tables.

Error "Auto-increment columns"
Check if each table within the diagram has no more than one auto-increment column, as otherwise MySQL will not allow such table to be created.

Error "Trigger Times and Events"
Check if every trigger in each table within diagram has unique Time and Event combination.

Table Columns

Error "Column Name Uniqueness"
Check diagram tables for the uniqueness of each column name within the table.

Warning "Column Name Max Length"
MySQL allows only 64 characters in column names and cuts names if they are longer than this.

Warning "Auto-increment Column Definition"
Check if each auto-increment column within a table is a part of a primary key.

Table Indexes

Error "Index Name Uniqueness"
Check diagram tables for the uniqueness of each index name within the table.

Warning "Index Name Max Length"
MySQL allows only 64 characters in index names and cuts names if they are longer than this.

Warning "Duplicate Index Column"
Check if each table column is indexed only once.

References

Warning "Reference Table Type"
Check if both linked tables use storage engine that supports references (for example InnoDB or Falcon) for each diagram reference.

Error "Reference Column Data Types"
Check whether the linked columns are of the same data type for each diagram reference.

Error "Reference Column does not exists"
Check an existence of referenced column in parent table for each diagram reference.

Error "Reference Name uniqueness"
Check diagram references for uniqueness of each reference name.

Error "Check "SET NULL" FK-action for NOT NULL column"
MySQL does not allow SET NULL action for foreign keys that reference column marked with NOT NULL attribute.

Domains

Error "Domain Name Uniqueness"
Check diagram for the uniqueness of each domain name within a diagram.

Stored Routine

Error "Stored Routine Name Uniqueness"
Check diagram stored procedures and functions for the uniqueness of name within a schema.

Warning "Stored Routine Name Max Length"
MySQL allows only 64 characters in stored routine names and cuts names if they are longer than this

Views

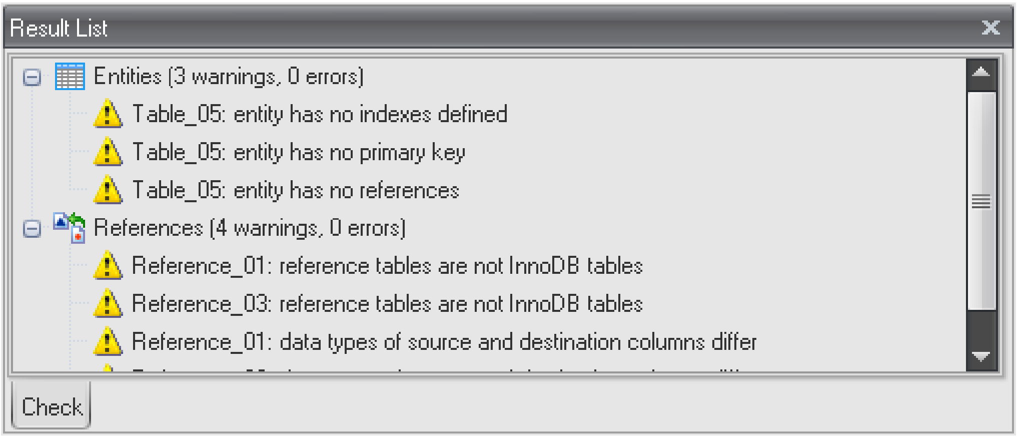
Error "View Name Uniqueness"
Check diagram stored procedures and functions for the uniqueness of name within a schema.

Warning "View Name Max Length"
MySQL allows only 63 characters in stored routine names and cuts names if they are longer than this.

Error "View was created on a not existing table"
Check existence of tables, which are used in the view.

Error "View was created on a not existing column"
check existence of table columns, which are used in the view.

After you click **OK** the check process will be displayed within the **Output** window and the result of the check will be displayed within the **Result** window in the same categorized view as described above.



Entities (3 warnings, 0 errors)
- Table_05: entity has no indexes defined
- Table_05: entity has no primary key
- Table_05: entity has no references

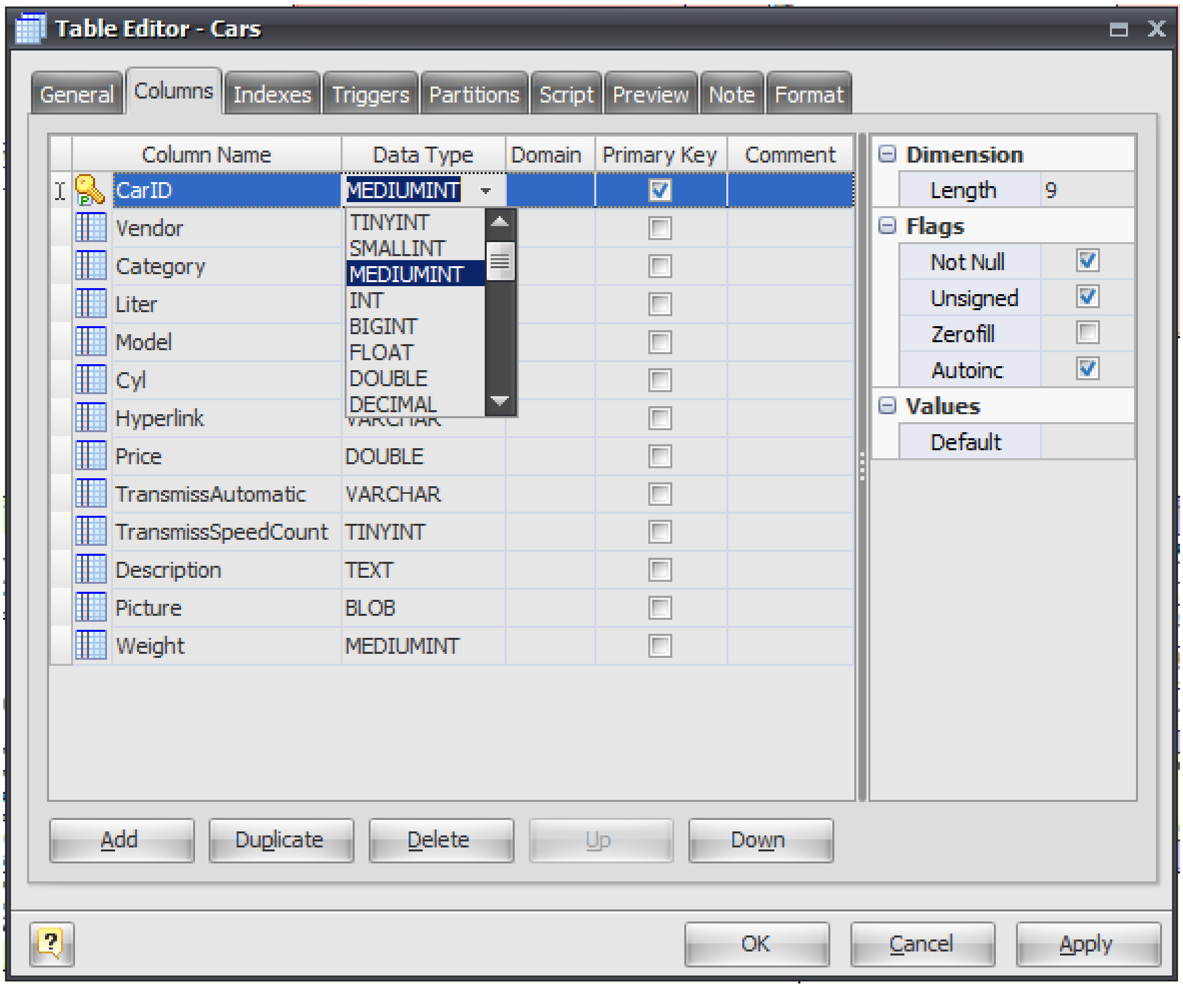
References (4 warnings, 0 errors)
- Reference_01: reference tables are not InnoDB tables
- Reference_03: reference tables are not InnoDB tables
- Reference_01: data types of source and destination columns differ

Double-click on a warning or an error in the list opens the editor window for the appropriate object (Table Editor, View Editor, Domain Manager and so on).

See also:
Interface User Guide: Docking Windows
Diagram Objects: Table Editor | Column Editor | Domain Manager | Reference Editor | Index Editor | Stored Routine Editor | Trigger Editor | View Editor

14.7. How to Edit Columns

The **Column Editor** is placed within the Table Editor dialog. It allows you to modify the list of table columns as well as column properties. Click the **Columns** tab of the **Table Editor** to manage table columns.



Column Name	Data Type	Domain	Primary Key	Comment
CarID	MEDIUMINT		☑	
Vendor			☐	
Category			☐	
Liter			☐	
Model			☐	
Cyl			☐	
Hyperlink	VARCHAR		☐	
Price	DOUBLE		☐	
TransmissAutomatic	VARCHAR		☐	
TransmissSpeedCount	TINYINT		☐	
Description	TEXT		☐	
Picture	BLOB		☐	
Weight	MEDIUMINT		☐	

Dimension	
Length	9
Flags	
Not Null	☑
Unsigned	☑
Zerofill	☐
Autoinc	☑
Values	
Default	

The **Column Editor** consists of the following areas:

- Column List
- Properties Pane
- Button Pane.

Column List

The column list displays all the columns in the table and allows you to modify the following column properties:

Column name
The name of the column, which must be unique within the table;

Data type
The type of the column, which specifies data to store in the column;

Domain
You can specify a column domain, see the Domains topic for detailed information;

Primary key
Specify this option to include the field into the table primary key;

Comment
An arbitrary description for the column.

Properties Pane

The properties pane allows you to define the advanced properties of the column, selected in the **Column List**. The appearance of this pane changes according to the data type of the column. These properties are:

Length
This attribute defines the maximum allowed length of the stored values; it applies to all integer, decimal, and string types;

Decimals
This attribute defines the number of digits, which follow the decimal point;

Not null
This option indicates that the stored column value cannot be NULL;

Autoinc
This attribute makes the column value autoincrement, i.e. each new value is set automatically according to the previous value; it applies to all integer values;

Binary
This attribute indicates that the string value is stored in a binary form within the column;

Unsigned
This option indicates that the stored integer or decimal value must be above zero;

Zerofill
With this attribute all unspecified digits of the field value are filled as zeros according to the column length and decimals; it applies to all integer and decimal types;

Set
This attribute defines the list of permissible values for the field of the ENUM or SET type;

Default
This attribute defines the default value, which the column accepts if no other is specified.

Buttons Pane

The buttons under the list of columns allow you to perform the following actions:

Add
Add a new column with the default properties to the end of the list;

Duplicate
Add a new column with the same properties as the selected column to the end of the list;

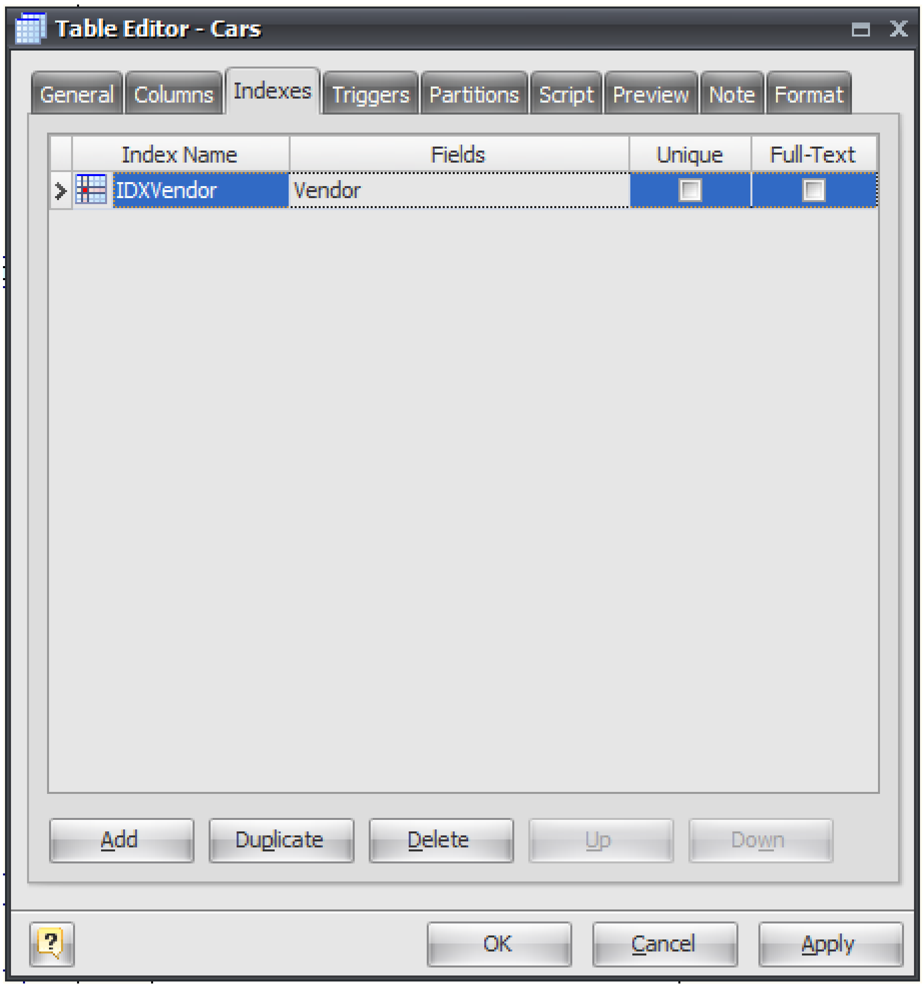
Delete
Remove the selected column from the list;

Up/Down
Move the selected column along the list.

See also:
Diagram Objects: Domains | Table Editor | Indexes | Triggers | Column Manager

14.8. How to Edit Table Index

The **Index Editor** is placed within the Table Editor dialog. It allows you to modify the list of table indexes as well as index properties.



Index Name	Fields	Unique	Full-Text
IDXVendor	Vendor	□	□

The main element of the editor is the index list, which displays all indexes available within the table. The columns of the list allow you to modify the properties of the selected index.

These properties are:

Index Name
The name of the index, which must be unique within the table.

Unique
Defines the UNIQUE constraint for the selected columns. I.e. the combination of the included field values must be unique within the table.

Fields
The list of index attributes.

Full-text
This option can be applied to the index if all its fields are *CHAR*, *VARCHAR*, or *TEXT*; it allows you to perform relevance-based search within the included fields.

The buttons under the list of indexes allows you to perform the following actions:

Add
Add a new index with the default properties to the end of the list;

Duplicate
Add a new index with the same properties as the selected index to the end of the list;

Delete
Remove the selected index from the list;

Up/Down
Move the selected index along the list.

See also:
Diagram Objects: Indexes | Index Manager

14.9. How to Generate a Database

The **Database Generation** tool can generate SQL script, that represents the diagram you developed and (optionally) executes it on the database server.

You can generate database in two ways:

- Directly execute a generated script on a MySQL server using SQL Executor.
- Generate a script to be executed on MySQL server at a later time.

In both cases, the database generation commands are saved in a script file. You must always provide path to the script file.

Database Generation

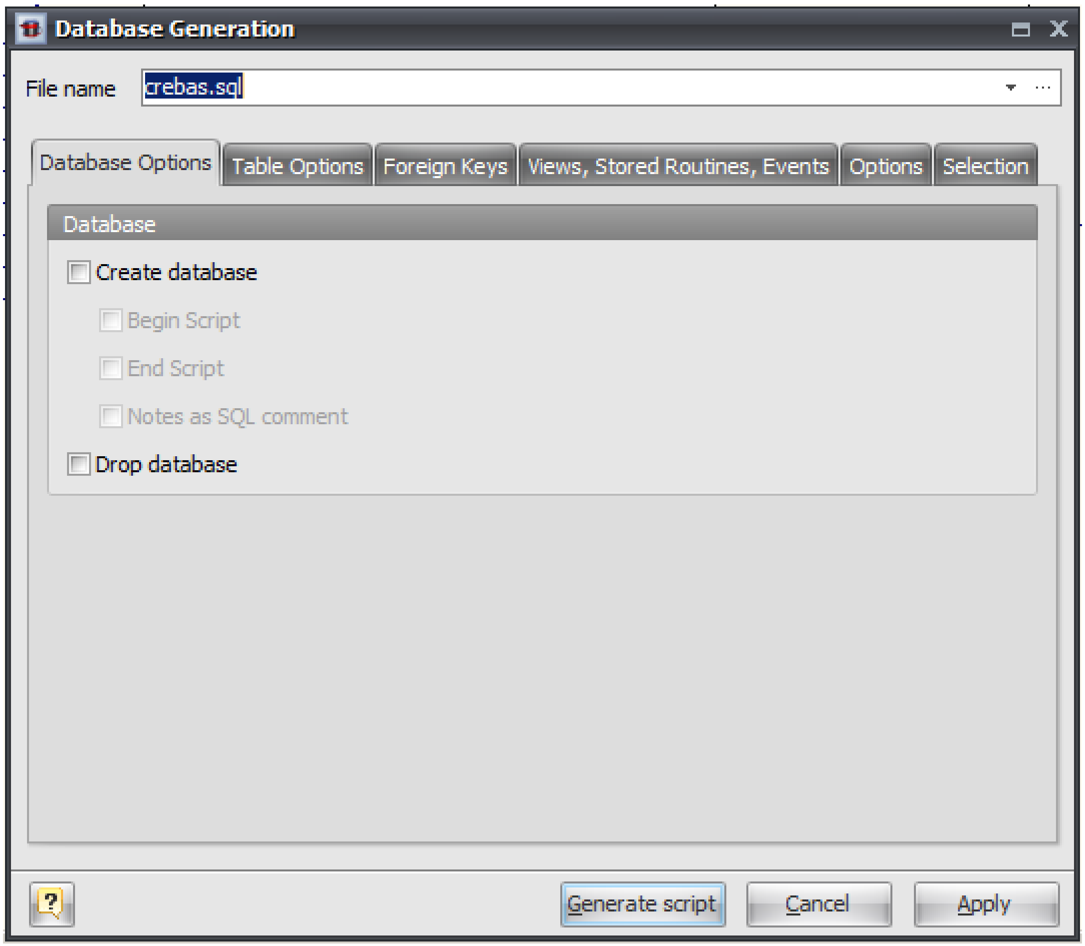
To generate database, start the **Database Generation** tool by using **Generate Database** ( ) item on **Database** tab of the Ribbon or pressing **Ctrl+G**. The following pictures demonstrate **Database Generation** tool interface.

- **File name field**
  This field allows you to set file, in which generated SQL statements will be stored. Click on the ... button near the field to browse to file on the file system.

Database Generation tool consist of several tabs, which contain SQL generation options. Let's explore them.

Database Options

In the **Database Options** tab of the **Database Generation** tool you can set database generation options.



File name crebas.sql

□ Create database

□ Begin Script

□ End Script

□ Notes as SQL comment

□ Drop database

Create Database

This option enables generation of the database. Set this option on if you want to create a new physical database instead of using an existing one. The table structure (i.e. tables) will be generated in both cases.

Begin Script

This option enables inserting the begin script before the CREATE DATABASE statement. You can edit database begin script in Database Editor.

End Script

This option enables inserting the end script after the CREATE DATABASE statement. You can edit database end script in Database Editor.

Notes as SQL Comment

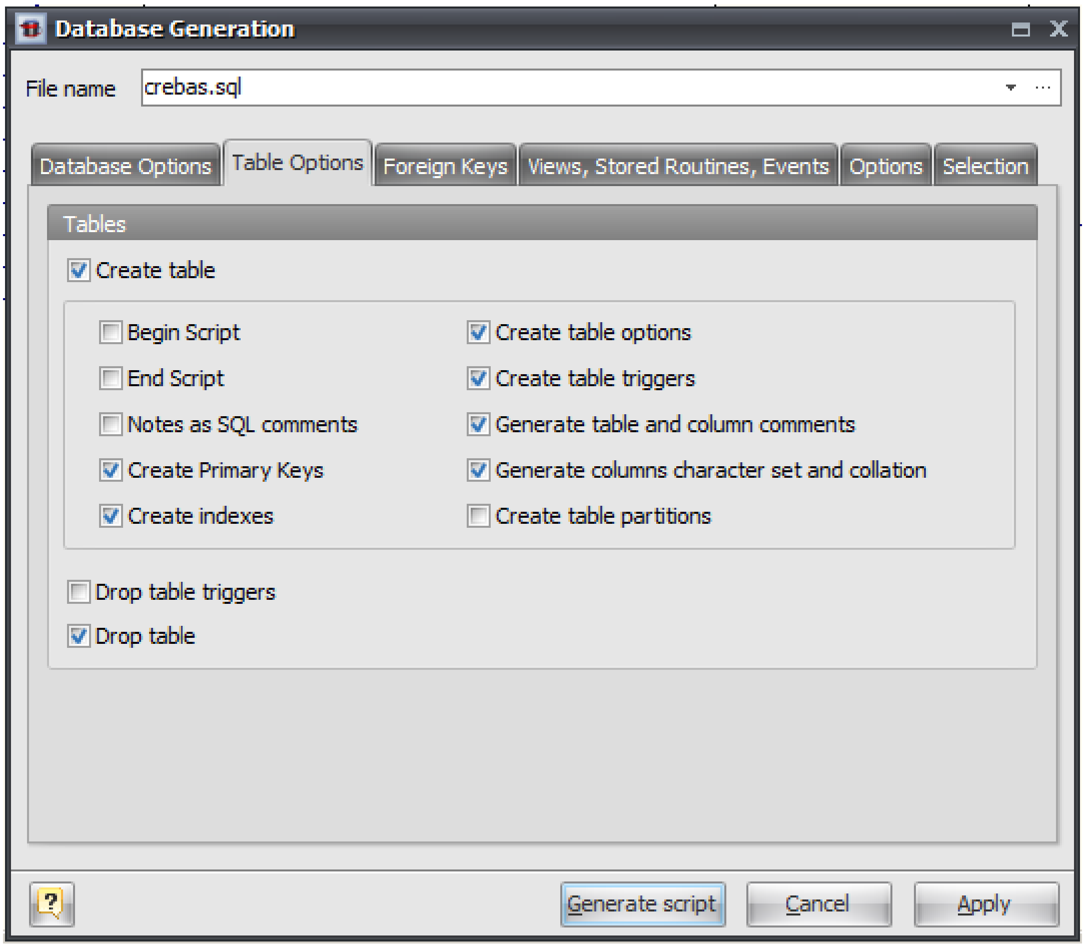
This option enables showing of database comments in SQL script.

Drop Database

This option enables dropping old database if it was already exist.

Table Options

In the **Table Options** tab of the **Database Generation** tool you can set tables generation options.



File name: crebas.sql

☑ Create table

☐ Begin Script
☐ End Script
☐ Notes as SQL comments
☑ Create Primary Keys
☑ Create indexes
☑ Create table options
☑ Create table triggers
☑ Generate table and column comments
☑ Generate columns character set and collation
☐ Create table partitions

☐ Drop table triggers
☑ Drop table

Create Tables
This option enables generation of tables.

Begin Script
This option enables inserting the begin script (it can be set using Table Editor) before the CREATE TABLE statement.

End Script
This option enables inserting the end script (it can be set using Table Editor) after the CREATE TABLE statement.

Notes as SQL Comments
This option enables showing of table notes in SQL script, which was set in the Table Editor. Notes will be inserted to the SQL script as commented strings.

Create Primary Keys
This option enables generation of table primary keys.

Create Indexes
This option enables generation of indexes for the tables.

Create Table Options
This option enables setting of table options, such as ENGINE, ROW_FORMAT and others in the SQL script.

Create table triggers
Generate triggers set on table.

Generate table and column comments
This option adds COMMENT clause to column and table definitions if some comment set in Table Editor.

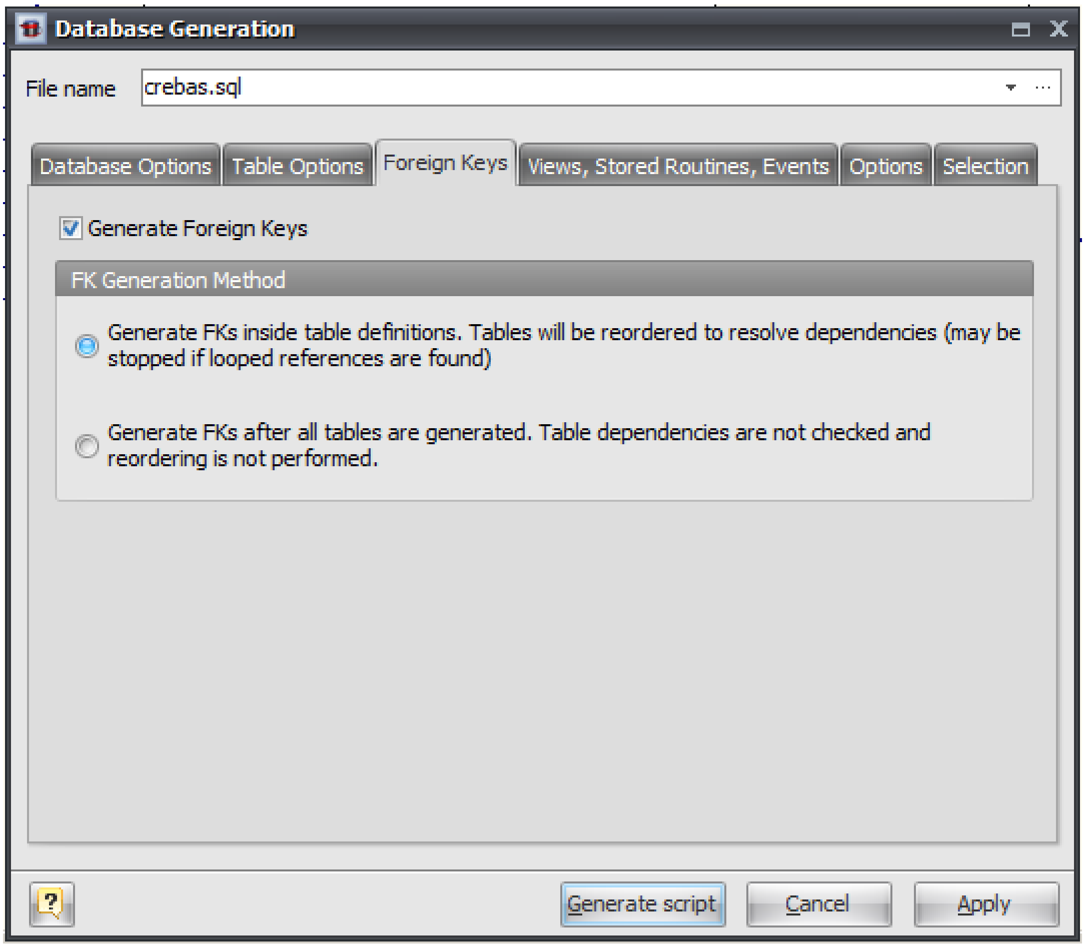
Create table partitions
Enable generation of partitions for tables.

Drop table triggers
This option generates DROP TRIGGER IF EXISTS statement before generating CREATE TRIGGER statement. There is no need to set this this option if Drop table option is set.

Drop table
This option enables dropping old tables if they were already exist. I.e. enables generating DROP TABLE IF EXISTS "<TABLE_NAME>" statement before CREATE TABLE statement.

Foreign Keys

Foreign Keys tab of **Database Generation** tool allows to set up foreign keys generation options.



File name: crebas.sql

☑ Generate Foreign Keys

FK Generation Method

● Generate FKs inside table definitions. Tables will be reordered to resolve dependencies (may be stopped if looped references are found)

○ Generate FKs after all tables are generated. Table dependencies are not checked and reordering is not performed.

Generate Foreign Keys

This option enables generation of table foreign keys presented as References on diagram.

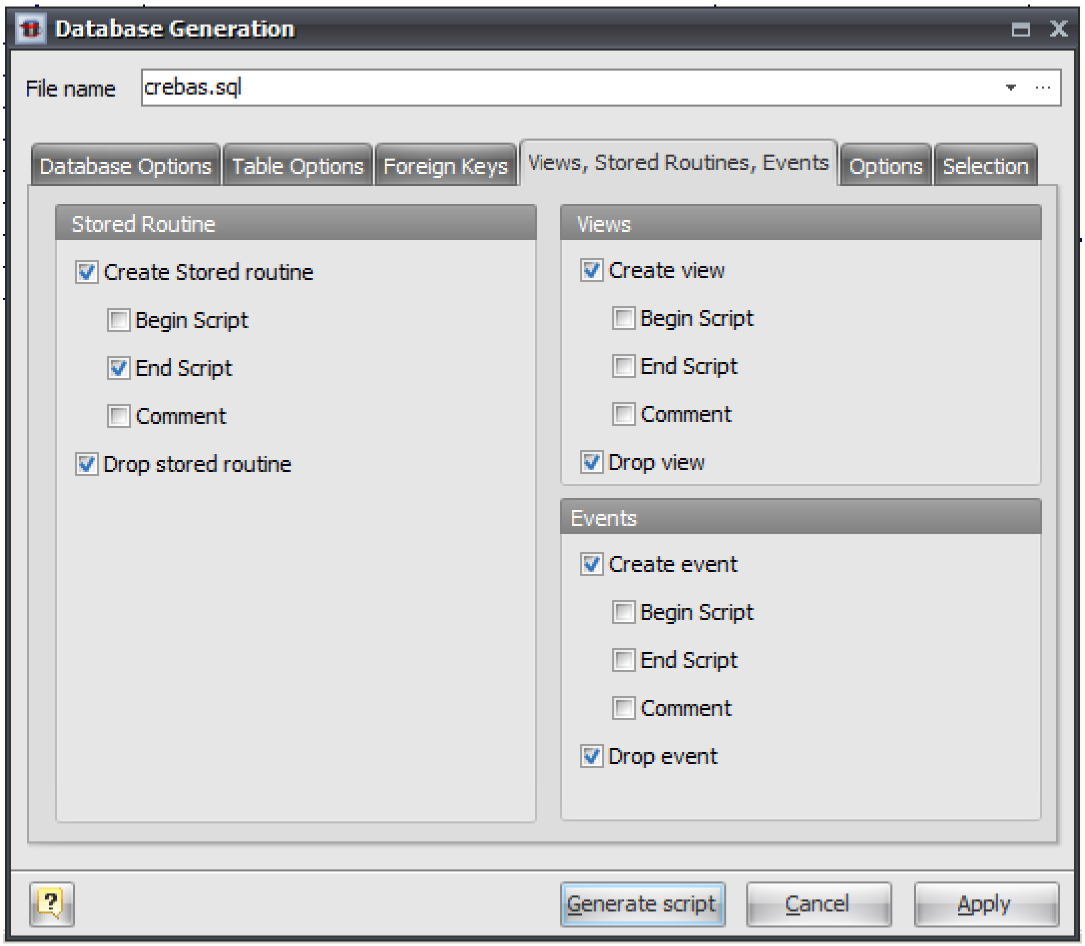
FK Generation Method

This option defines foregn keys generation method:

Generate FKs inside table definitions.
All FKs are generated as a part of table definition inside CREATE TABLE statement. This method requires reordering of diagram tables, because some tables may depend on other ones. Sometime there can be cycle references between tables that can not be generated properly using this method. That is why second method of FKs generation is preferred.

Generate FKs after all tables are generated.
When using this method all FKs will be generated using ALTER TABLE ... ADD CONSTRAINT statements. This method does not require tables reordering.

Views, Stored Routines, Events



File name: crebas.sql

Stored Routine

- ☑ Create Stored routine
  - ☐ Begin Script
  - ☑ End Script
  - ☐ Comment
- ☑ Drop stored routine

Views

- ☑ Create view
  - ☐ Begin Script
  - ☐ End Script
  - ☐ Comment
- ☑ Drop view

Events

- ☑ Create event
  - ☐ Begin Script
  - ☐ End Script
  - ☐ Comment
- ☑ Drop event

Stored Routines Group

Create Stored Routine
This option enables generation of stored routines (i.e. stored functions and procedures).

Begin Script
This option enables inserting the begin script before the SQL generation statement.

End Script
This option enables inserting the end script after the SQL generation statement.

Comment
This option enables showing of stored routines comments in SQL script.

Drop Stored Routine
This option enables dropping old stored routines if it were already exist.

Views Group

Create View
This option enables generation of database views.

Begin Script
This option enables inserting the begin script before the SQL generation statement.

End Script

This option enables inserting the end script after the SQL generation statement.

Comment
This option enables showing of view comments in SQL script.

Drop Views
This option enables dropping old views if it were already exist.

Events Group

Create View
This option enables generation of database events.

Begin Script
This option enables inserting the begin script before the events SQL generation statement.

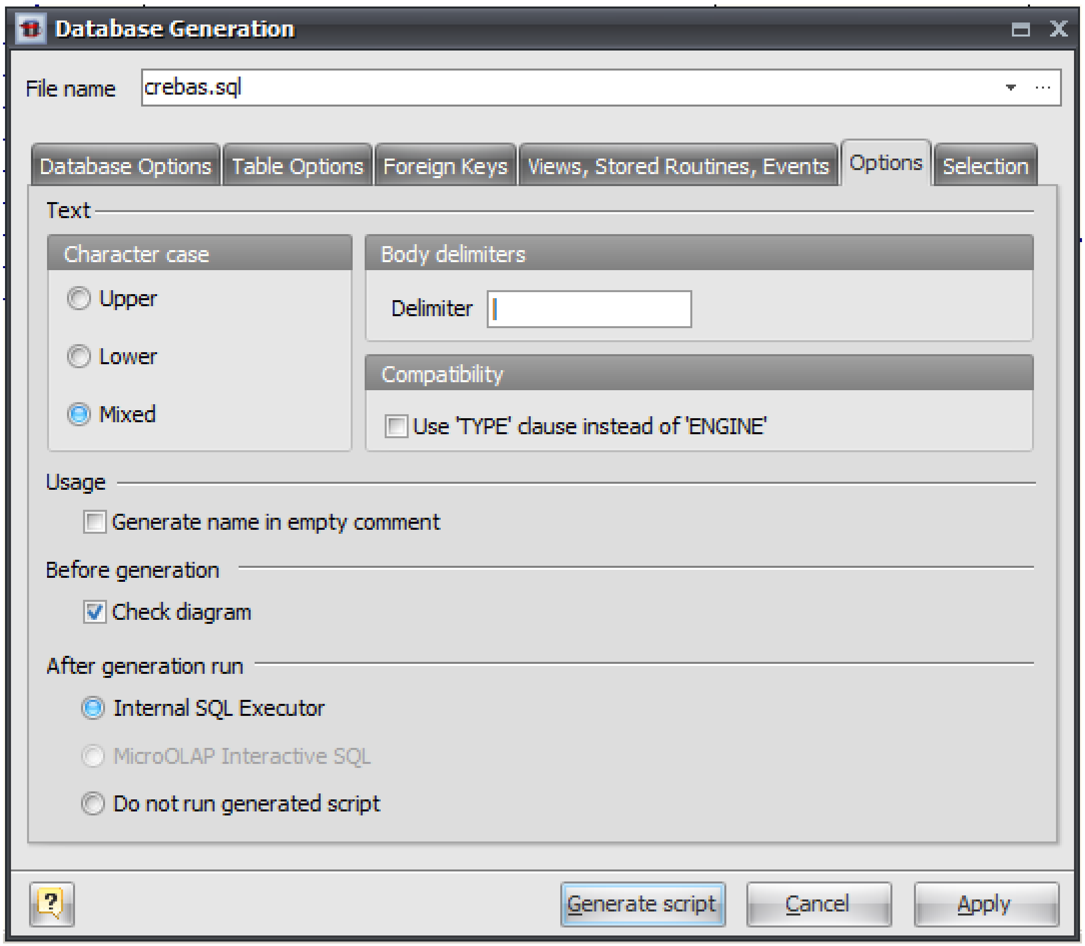
End Script
This option enables inserting the end script after the events SQL generation statement.

Comment
This option enables showing of event comments in SQL script.

Drop Views
This option enables dropping old events if it were already exist.

Options

This tab allows set generation-related options.



File name: crebas.sql

Text

Character case
- ○ Upper
- ○ Lower
- ● Mixed

Body delimiters
Delimiter: |

Compatibility
- □ Use 'TYPE' clause instead of 'ENGINE'

Usage
- □ Generate name in empty comment

Before generation
- ☑ Check diagram

After generation run
- ● Internal SQL Executor
- ○ MicroOLAP Interactive SQL
- ○ Do not run generated script

Character case

Defines case of characters, which will be used in generated script. Use Mixed option to leave the characters unmodified.

Body Delimiter

Defines symbol or string to use as body delimiter with DELIMITER statement for multi-statement stored routines, events and triggers.

Use 'TYPE' clause instead of 'ENGINE'

This option enables generation of TYPE clause instead of ENGINE in CREATE TABLE statement. TYPE clause is deprecated since MySQL 5.1 so you probably do not need this option.

Generate name in empty comments

Write the name of an object if the comment field is empty.

Check diagram

Enables checking the diagram before generation.

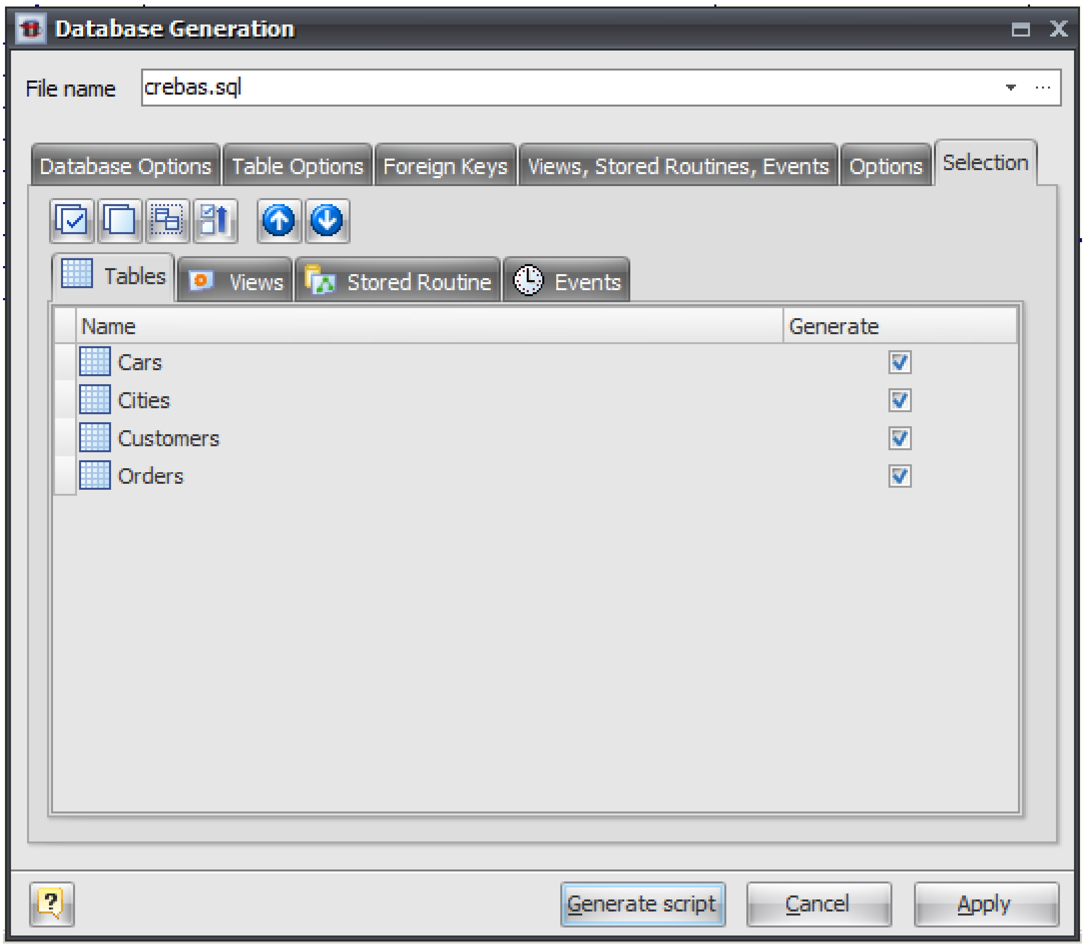
After generation, run

- Internal SQL Executor - Send the generated SQL statements into the internal SQL Executor.
- MicroOLAP Interactive SQL - This options active only if MicroOLAP Interactive SQL for MySQL is installed. Send the generated SQL statements into the advanced SQL editor MicroOLAP Interactive SQL for MySQL.
- Do not run generated script - Select this option if want just to generate SQL script without

running it. For example you can use this option if you want to execute SQL script on remote server.

Selecting objects to generate

You can select diagram objects you want to generate in SQL script or database. Use the **Selection** tab of the **Database Generation** tool for it.



File name: crebas.sql

Name	Generate
Cars	☑
Cities	☑
Customers	☑
Orders	☑

There are several subtabs: **Tables**, **Stores Routines**, **Views** and **Views**. Each of which allows you to select appropriate diagram objects to generate.

To enable particular objects generation, click on the checkbox near it.

The default selection of objects to generate depends on their **Generate** property.

Pay attention to the buttons on the **Selection** tab:

Select All
Checks on all checkboxes.

Deselect All
Checks off all checkboxes.

Use graphical selection

Checks on checkboxes for objects, depending on diagram selection.

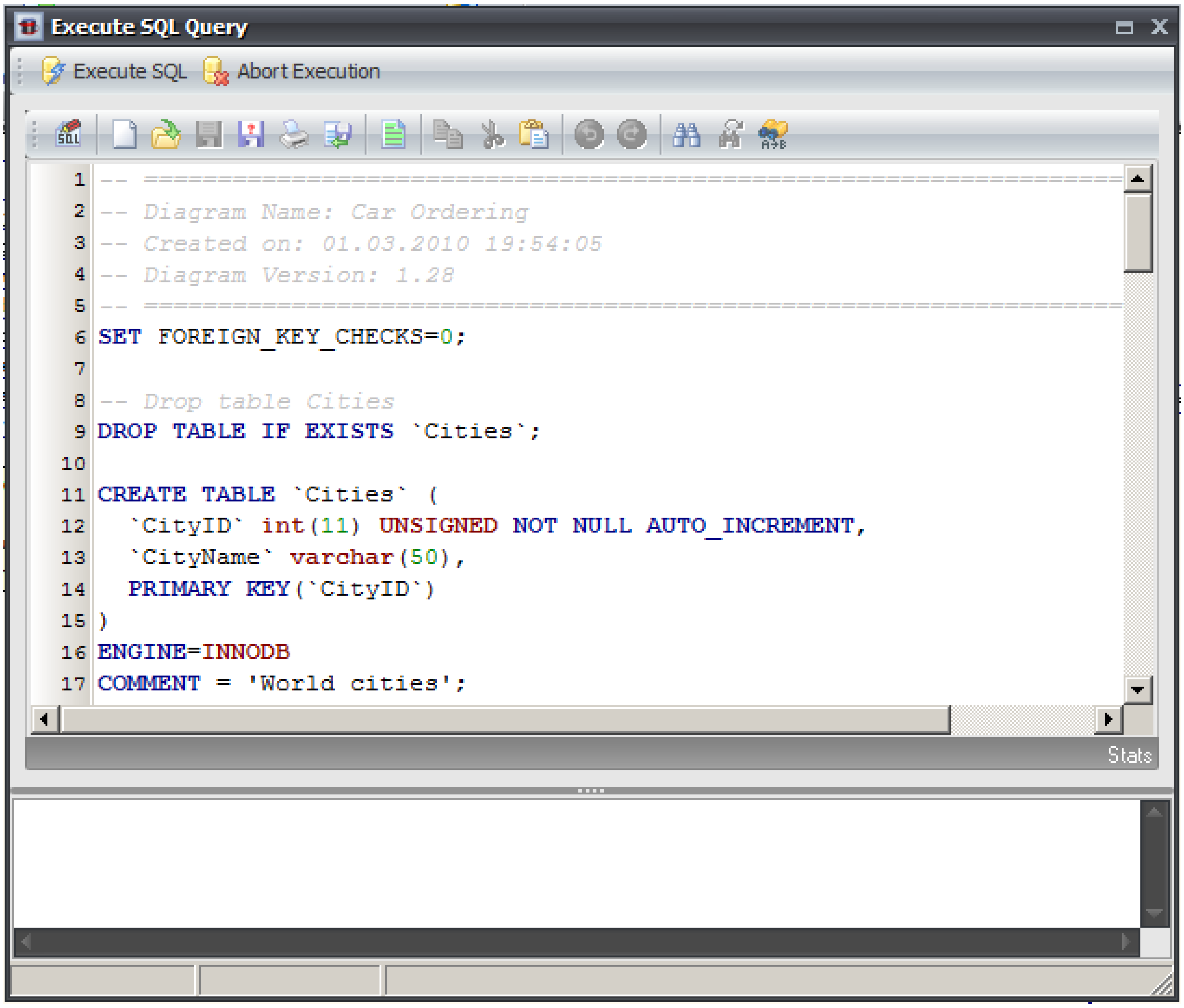
Move selected items to top
Reorders items in list for checked items to be at the beginning of list.

You can change the order of tables in which they will be placed in the generated SQL script. Use the buttons with arrows for this.

Generating, customizing and executing SQL

Click on the **Generate script** button on the **Database Generation** tool to generate SQL script. The generated SQL script will be stored in the file you have set.

If you selected **Internal SQL Executor** on **Options** tab, the SQL Executor with generated SQL statements will appear.



```
-- ==========================================================================
-- Diagram Name: Car Ordering
-- Created on: 01.03.2010 19:54:05
-- Diagram Version: 1.28
-- ==========================================================================
SET FOREIGN_KEY_CHECKS=0;

-- Drop table Cities
DROP TABLE IF EXISTS `Cities`;

CREATE TABLE `Cities` (
  `CityID` int(11) UNSIGNED NOT NULL AUTO_INCREMENT,
  `CityName` varchar(50),
  PRIMARY KEY(`CityID`)
)
ENGINE=INNODB
COMMENT = 'World cities';
```

You can easily customize statements for your needs. And then send them to the database server by clicking on the **Execute SQL** button.

Please, examine SQL Executor section to know more about it.

See also:
Database Accessing Tools: SQL Executor
Database Objects: Database Editor | Table Editor

14.10. How to Modify a Database

Database Modification Overview

Once you have changed your diagram, it's usually necessary to apply these changes to your database. It's easy to do with the **Database Modification** tool. The **Database Modification** is generation of SQL script that leads your database to the current state of your diagram.

> **Please note:**
> Database modification usually causes multiple complex statements for database structure modification. It is possible that some of them may not execute correctly due to some physical reason. It's recommended to make a backup of your database before applying structure changes to database.

You can modify database in two ways:

- Directly execute a modification script on a MySQL server. Please examine Connect to a Database section to explore the database connection process;
- Generate a modification script for executing at a MySQL server some time later.

In both cases, the database modification commands are saved in a script file. You must always provide the path to the script file.

> **Please note:**
> Sometimes you need to apply changes in your diagram to physical database without leading whole database structure to current state of your diagram. For example if there are some objects in database that are not covered by your diagram or if you want to apply only particular set of diagram changes to database. You can do this using Diagram Revisions and Compare Diagram tool. Please take a look at Comparing Diagram Revisions topic for details.

That's how the **Database Modification** tool works:

1. Reverse engineers your existing MySQL database.

2. Compares the result with your current database diagram.

3. Creates a list that contains differences between database objects and diagram objects. Then lets you to examine this list and to select changes you want to apply to database.

4. After analyzing the difference list, creates necessary SQL statements, that modify database structure.

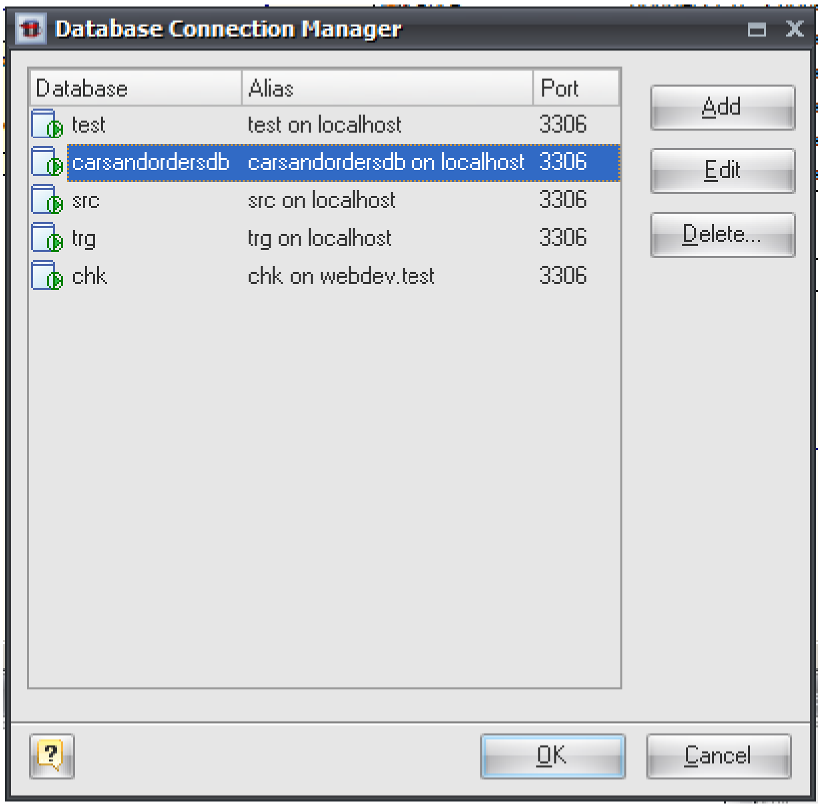
If some table is going to be modified, **Database Designer for MySQL** makes a backup copy of that table, so you can restore data and table structure later on if there will be some errors during the table structure alteration. Those backup tables have *_tmp_* string appended before their names. For example `*Cars*` table will have `*_tmp_Cars*` backup copy.

> **Please note:**
> Opposite process of applying physical database changes to diagram is described in Diagram Synchronization topic.

Step by step guide

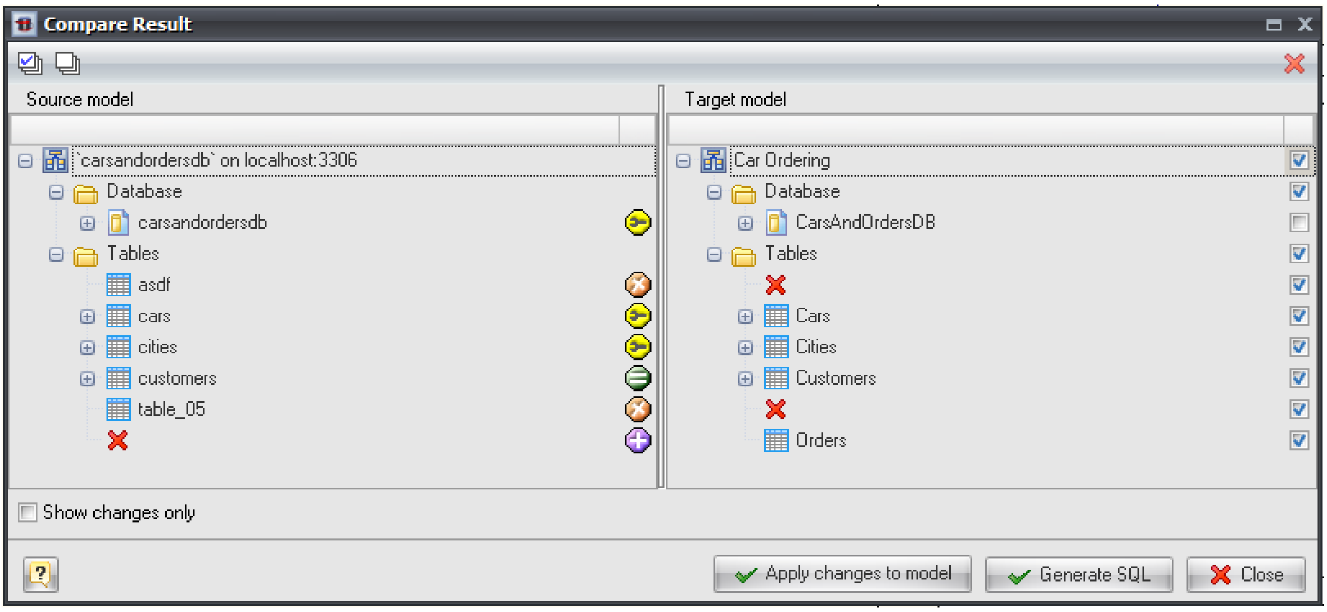
To modify your database, start **Database Modification** tool by selecting the **Modify Database** ( ) item on **Database** tab of the Ribbon or pressing **Ctrl+M**.

Database Designer for MySQL asks you with Database Connection Manager what database you want to modify.

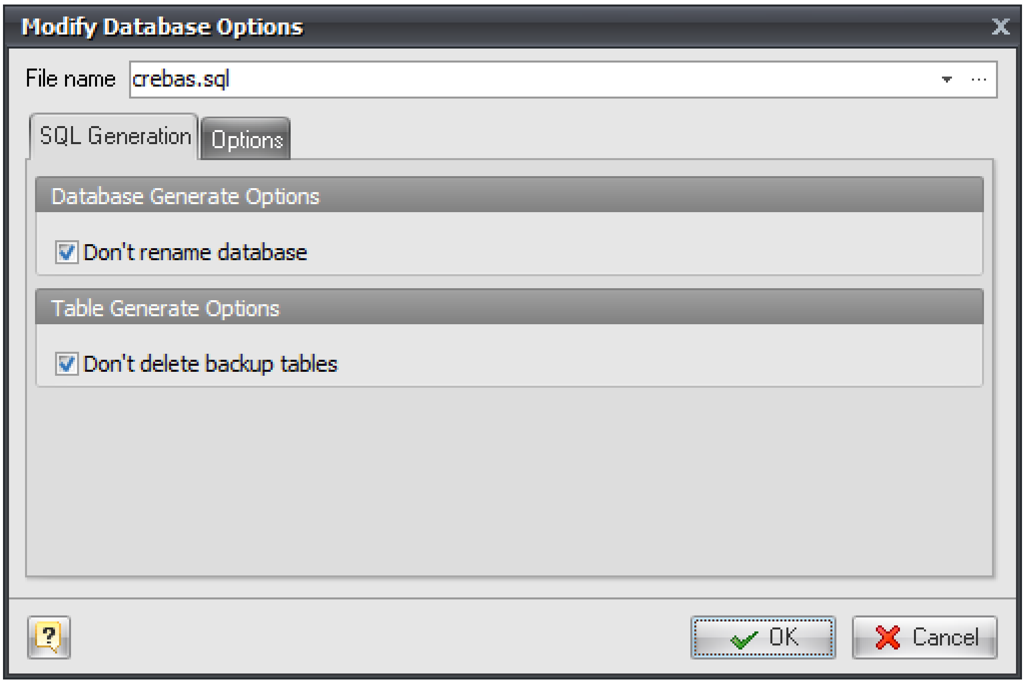


Database	Alias	Port
test	test on localhost	3306
carsandordersdb	carsandordersdb on localhost	3306
src	src on localhost	3306
trg	trg on localhost	3306
chk	chk on webdev.test	3306

After connecting to selected database **Database Designer for MySQL** performs Reverse Engineering of this database, compares it with diagram and shows Compare Result dialog with all differences found.



You can check or uncheck particular changes you want to generate SQL code for. Then press **Generate SQL** button to open **Modify Database Options** dialog.



File name crebas.sql

Database Generate Options

☑ Don't rename database

Table Generate Options

☑ Don't delete backup tables

There are following options on **SQL Generation** tab:

Don't rename database
Check this option to prevent database renaming even if physical database name differs from one set in diagram.

Don't delete backup tables
If this option is set **Database Designer for MySQL** will not generate DROP TABLE statements for backup table copies (with _tmp_ prefix) after altering tables structure.

Settings on **Options** tab are the same as on **Options** tab of Database Generation tool.

Press **OK** button to generate SQL script. SQL Executor with generated SQL statements for database modifications will appear. You can easily customize statements according to your wants and wishes. And then send them to the database server by clicking on the **Execute SQL** button. Please, examine SQL Executor topic to know more about it.

See also:
Diagram Functions: Compare Diagram | Compare Result dialog | Diagram Versioning (Revisions) | Comparing Diagram Revisions | Diagram Synchronization
Database Functions: Database Generation
Database Accessing Tools: Database Connection Manager | Connect to a Database | SQL Executor

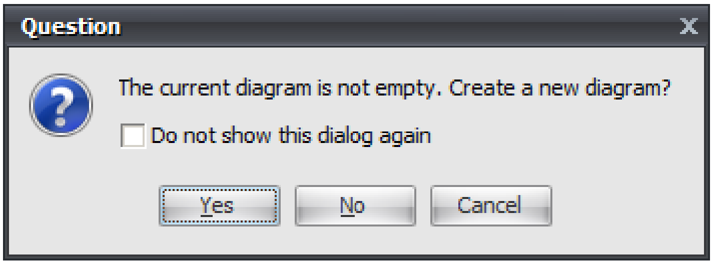
14.11. How to Import from a MySQL Database

With **Database Designer for MySQL** you can reverse engineer an existing MySQL database. This means that you can extract the database tables, attributes, relationships, indexes, views, stored procedures and other objects from the database to your diagram.

To reverse engineer MySQL database:

1. Select **Reverse Engineering** () item on **Database tab** of the Ribbon or press Ctrl+R.

2. If the currently opened diagram already contains some objects, a warning dialog box will appear:



Question

The current diagram is not empty. Create a new diagram?

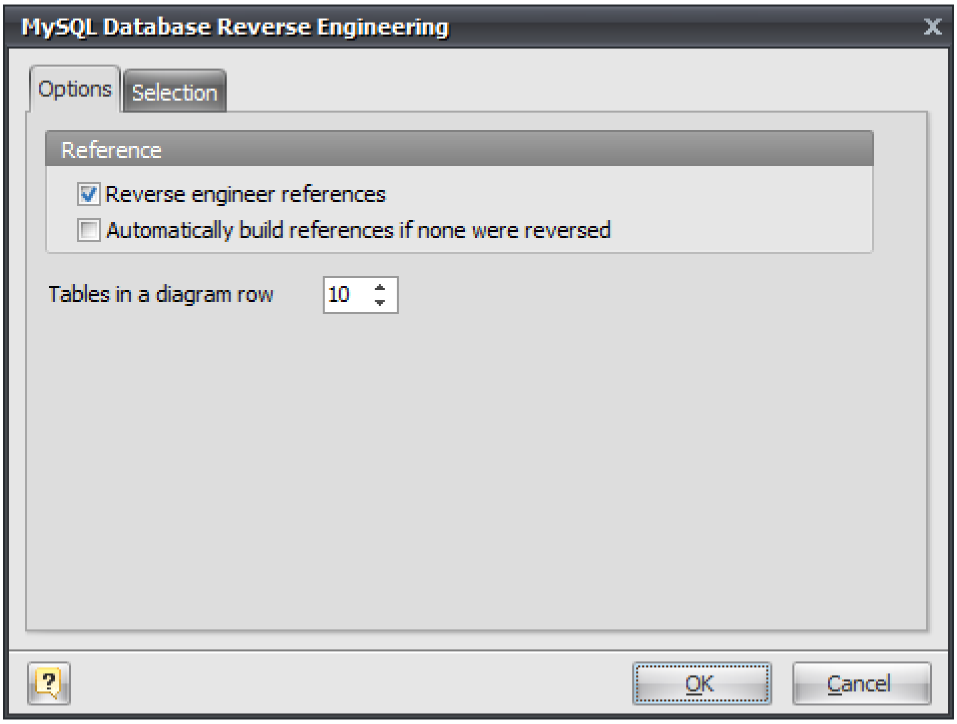
□ Do not show this dialog again

It requests you to create a new diagram to place the reversed objects into or use the currently opened diagram.

- Click **Yes** to create a new diagram or **No** to use the currently opened diagram. Click **Cancel** to cancel reverse engineering.
- Select **Do not show this dialog again** to disable future notifications.

3. If connection for the current diagram has not been established, Database Connection Manager will be shown. Select a profile from the list of the available ones or create a new profile to connect to the database you want to revere engineer.

4. The **MySQL Database Reverse Engineering tool** will be shown.



☑ Reverse engineer references

□ Automatically build references if none were reversed

Tables in a diagram row 10

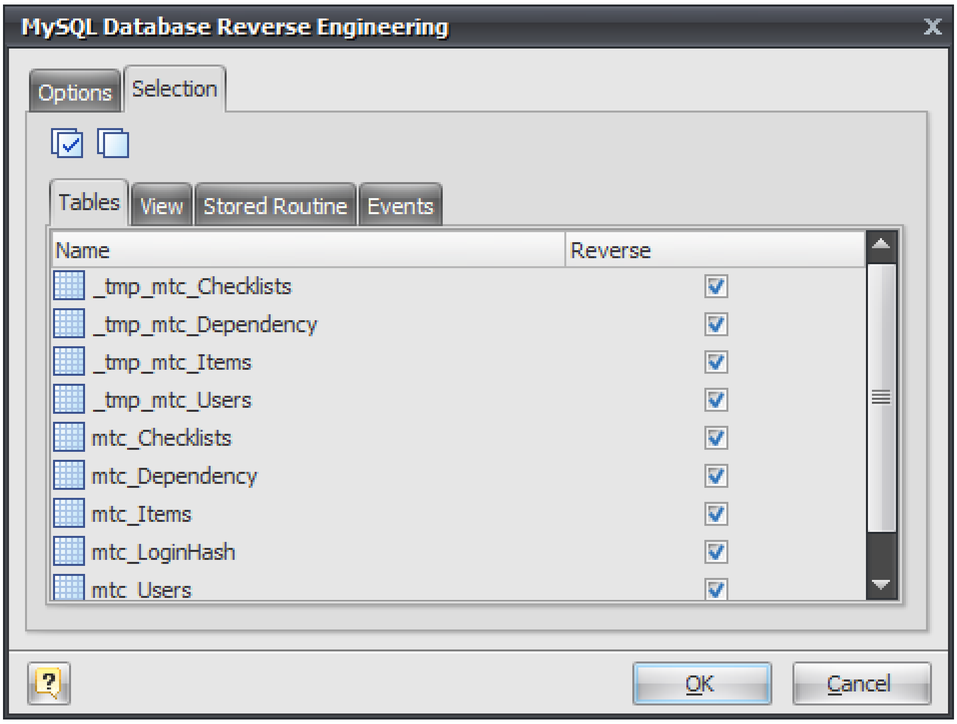
You can set the reverse engineering options in the **Options tab**.

Reverse Engineer references
This option enables extracting foreign keys from the database and creating appropriate references in your diagram.

Automatically build references if none were reversed
If there are no physical references extracted, it is possible to build them from a logical structure of the database. Enabling this feature leads to automatic reconstruction of references. Such reconstruction works by the following scheme: each column of the table is being compared with all primary keys of other tables, and if the column name and data type match one of the primary keys, a reference between the source column and the key column will be created. This option is available for modification only if Build references is checked.

Tables in a diagram row
This option determines how reversed tables will be disposed on the diagram. Reversed tables will be placed on the diagram in rows with equal distance, this option determines how many tables maximum there will be in one row.

5. In the **Selection tab** of the **MySQL Database Reverse Engineering dialog** you can choose the objects you want to reverse engineer.



Name	Reverse
_tmp_mtc_Checklists	☑
_tmp_mtc_Dependency	☑
_tmp_mtc_Items	☑
_tmp_mtc_Users	☑
mtc_Checklists	☑
mtc_Dependency	☑
mtc_Items	☑
mtc_LoginHash	☑
mtc_Users	☑

There are several subtabs: **Tables, Views, Stores Procedures, Events**. Each of them allows you to select appropriate database objects. Click on the checkbox near the object to enable its reverse engineering. Pay attention to the **Select All** (☑) and **Deselect All** (□) buttons on the tab, they allow you to select/deselect all objects in the list.

6. Click **OK** to start the database reverse engineering process. The **Output -> Reverse** docking window will display the state of the process.

7. The reversed database objects will be placed in your diagram.

See also:
Database Accessing Tools: Database Connection Manager
Reverse Engineering and Import: Reverse Engineering and Import Overview | Import from Access Database | Universal Reverse Engineering

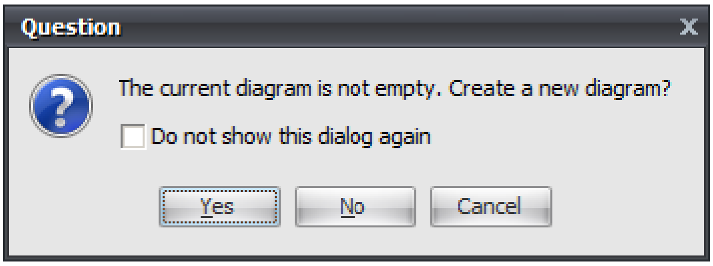
14.12. How to Import from a Microsoft Access Database

You can reverse engineer an existing **Microsoft Access** database. This means that you can extract the database tables, attributes, relationships, indexes and other objects from Microsoft Access database file to your diagram.

To reverse engineer **Microsoft Access** database:

1. Select **MS Access Database Reverse Engineering** () item from **Reverse Engineering** drop-down menu on **Database tab** of the Ribbon.

2. If the currently opened diagram already contains some objects, a warning dialog box will appear:



Question

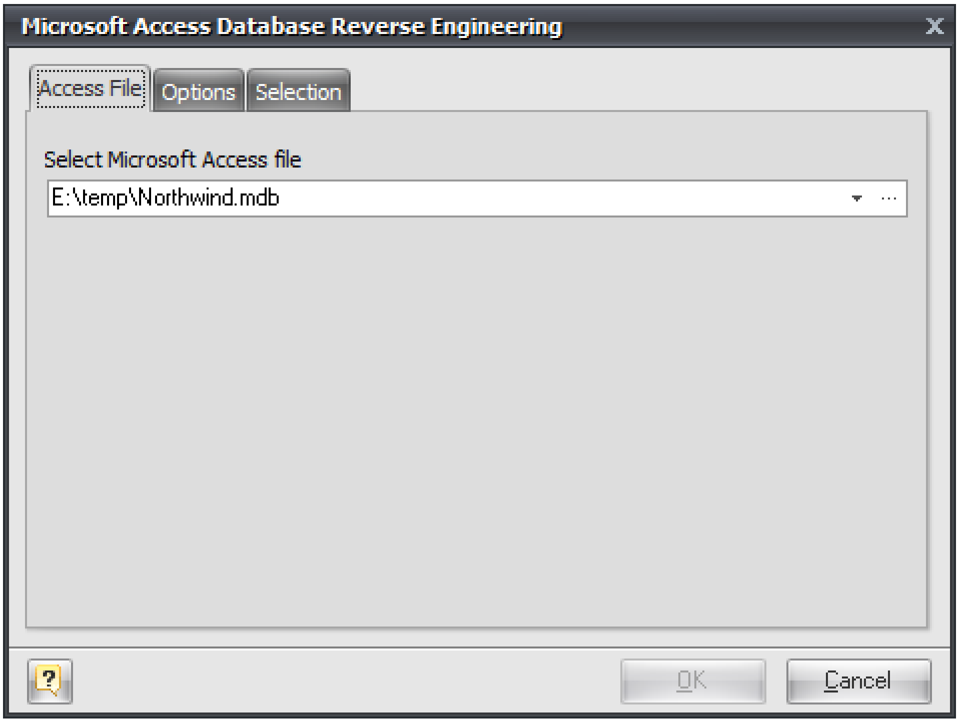
The current diagram is not empty. Create a new diagram?

□ Do not show this dialog again

It requests you to create a new diagram to place the reversed objects into or use the currently opened diagram.

Click **Yes** to create a new diagram, **No** to use the currently opened diagram. **Cancel** to cancel reverse engineering. Select **Do not show this dialog again** to disable future notifications.

3. The **Microsoft Access Reverse Engineering** tool will be shown.



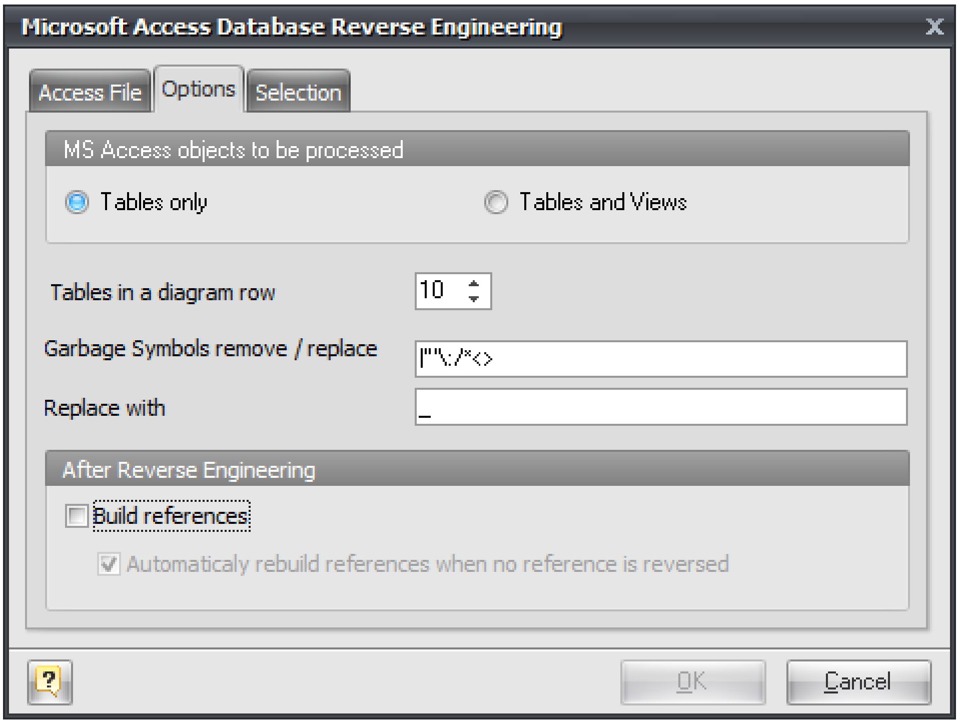
Microsoft Access Database Reverse Engineering

Select Microsoft Access file

E:\temp\Northwind.mdb

First, type in the full path to **Microsoft Access database** in the **Access File** tab or use "**...**"-button to browse file.

4. You can set the reverse engineering options in the **Options tab**.



Microsoft Access Database Reverse Engineering

MS Access objects to be processed

● Tables only ○ Tables and Views

Tables in a diagram row: 10

Garbage Symbols remove / replace: |"\:/*<>

Replace with: _

After Reverse Engineering

□ Build references

☑ Automaticaly rebuild references when no reference is reversed

Tables only

Reverse engineer only tables ignoring views.

Tables and Views

Reverse engineer both tables and views.

Tables in a diagram row

This option determines how reversed tables will be disposed on the diagram. Reversed tables will be placed on the diagram in rows with equal distance, this option determines how many tables maximum there will be in one row.

Garbage symbols remove/replace

Defines the symbols that will be replaced in the names of the objects.

Replace with

Defines the garbage replacement symbol.

Build references

This option enables extracting foreign keys from the database and creating appropriate references in your diagram.

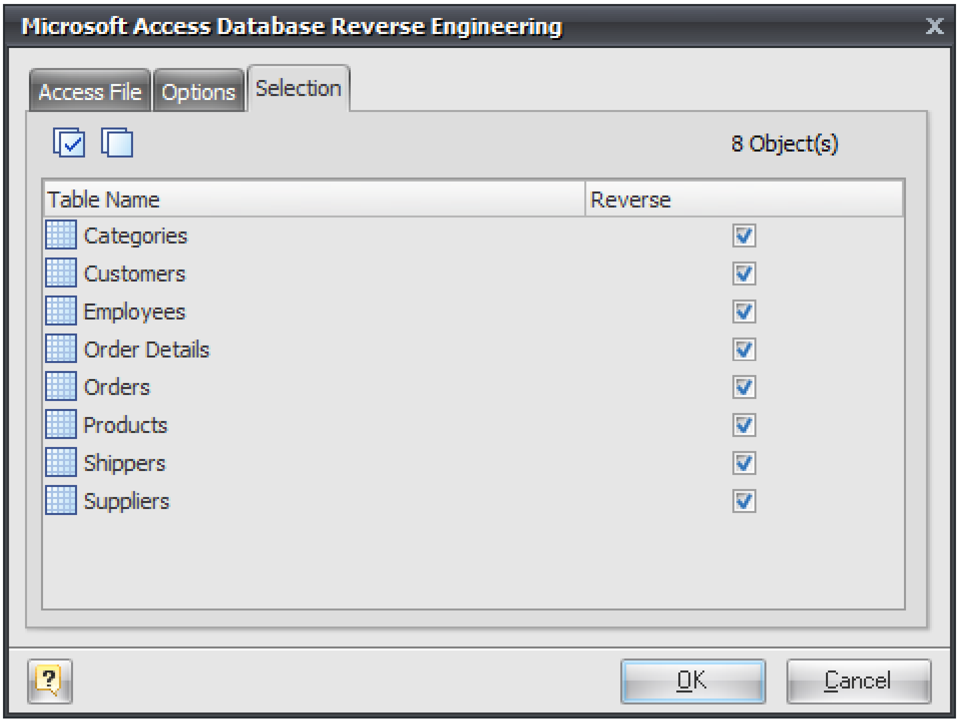
Automatically rebuild references when no reference is reversed

If there are no physical references extracted, it is possible to build them from logical structure of the database.

Enabling this feature leads to automatic reconstruction of references. Such reconstruction works by the following scheme: each column of the table is being compared with all primary keys of other tables, and if the column name and data type match one of the primary keys, a reference between the source column and the key column will be created.

This option is available for modification only if Build references is checked.

5. In the **Selection tab** of the **Access Reverse Engineering** tool you can choose the tables you want to reverse engineer.



8 Object(s)

Table Name	Reverse
Categories	☑
Customers	☑
Employees	☑
Order Details	☑
Orders	☑
Products	☑
Shippers	☑
Suppliers	☑

Click on the checkbox near the table to enable its reverse engineering. Pay attention to the **Select All** (☑) and **Deselect All** (□) buttons on the tab, they allow you to select/deselect all tables in the list.

5. Click **OK** to start the database reverse engineering process. The **Output -> Reverse** docking window will display the state of the process.

6. The reversed database objects will be placed on your diagram.

See also:
Reverse Engineering and Import: Reverse Engineering and Import Overview | Reverse Engineering MySQL Database | Universal Reverse Engineering

14.13. How to Import from other Databases

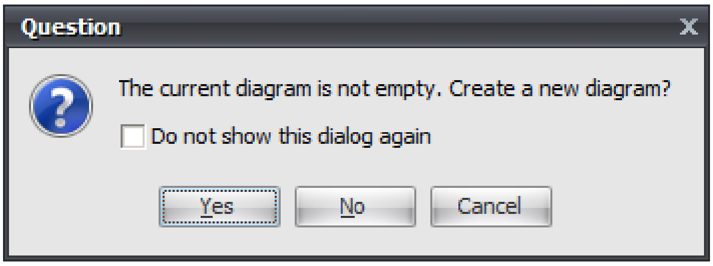
You can reverse engineer a number of databases, such as Sybase ASA and ASE, Oracle, PostgreSQL, Informix, MSSQL and others. This means that you can extract tables, attributes, relationships, indexes and other objects. It is possible with the **Universal Reverse Engineering** tool, which connects to a number of databases through **OLEDB** (**ADO**) or **ODBC** link. So, to reverse engineer a particular database, you need a corresponding **OLEDB** provider, or **ODBC** driver. In most cases, such libraries are installed in the system with the help of database client applications.

To reverse engineer a database through **OLEDB** or **ODBC** link:

1. Select **Universal Reverse Engineering** item from **Reverse Engineering** drop-down menu on

Database tab of the Ribbon.

2. If the currently opened diagram already contains some objects, the following dialog box will appear:



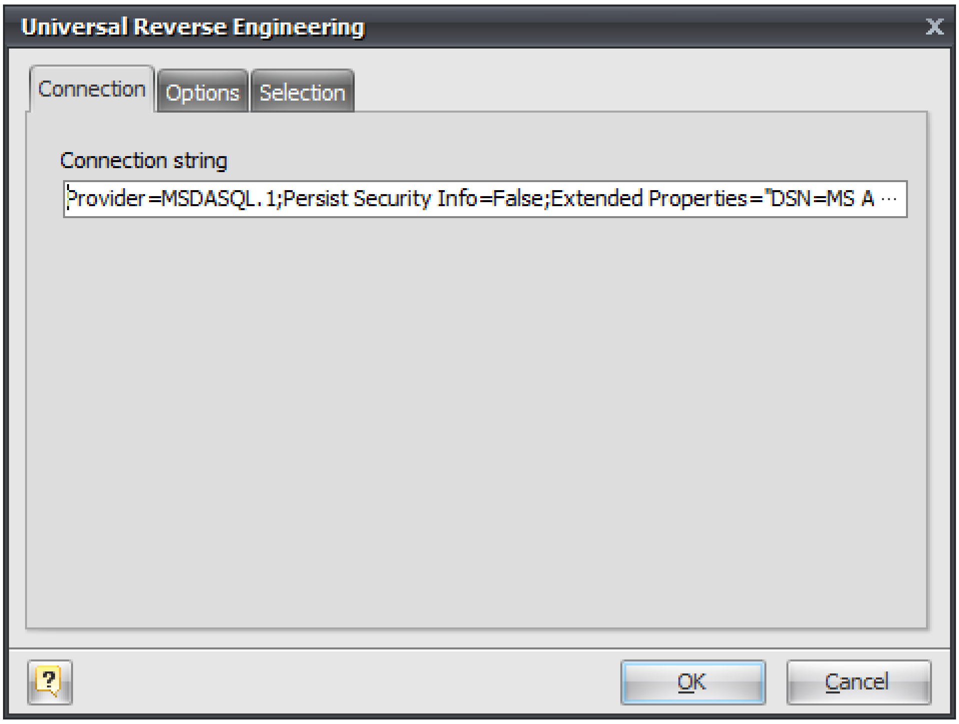
Question

The current diagram is not empty. Create a new diagram?

□ Do not show this dialog again

It requests you to create a new diagram to place the reversed objects into or use the currently opened diagram. Click **Yes** to create a new diagram, **No** to use the currently opened diagram, or **Cancel** to cancel reverse engineering. Select **Do not show this dialog again** to disable future notifications.

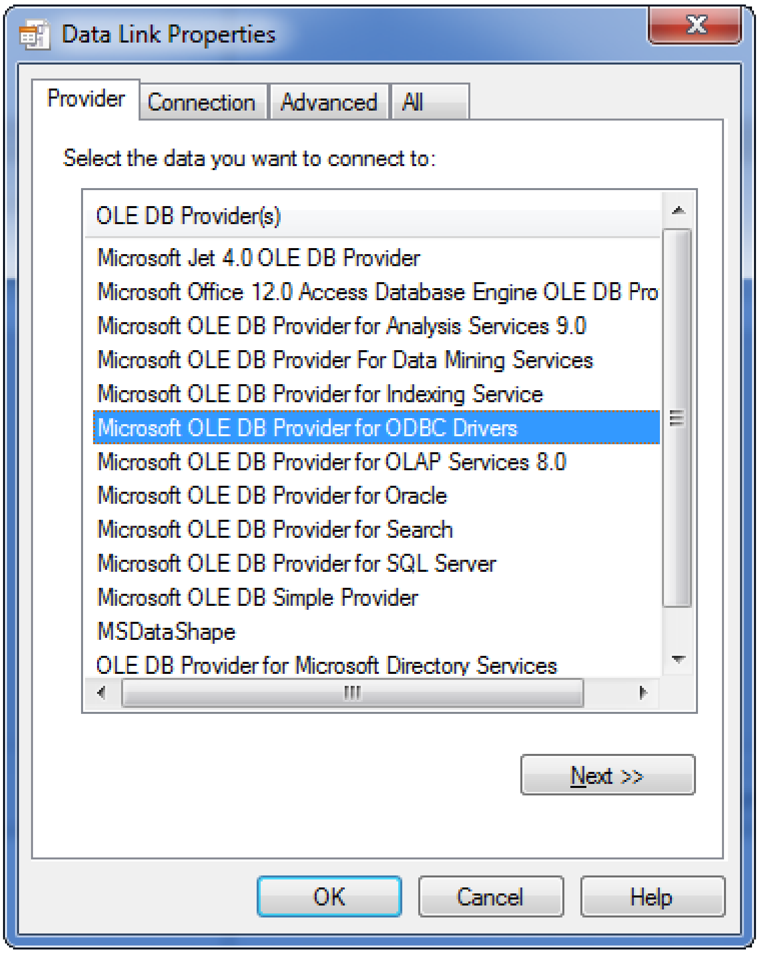
3. The **Universal Reverse Engineering** tool will be shown. First, provide full OLEDB connection string in the **Connection tab**.



Universal Reverse Engineering

Connection string: Provider=MSDASQL.1;Persist Security Info=False;Extended Properties="DSN=MS A

Click on the "**...**" button to call the standard system **Data Link Properties** dialog window, which helps you to build OLEDB connection string:

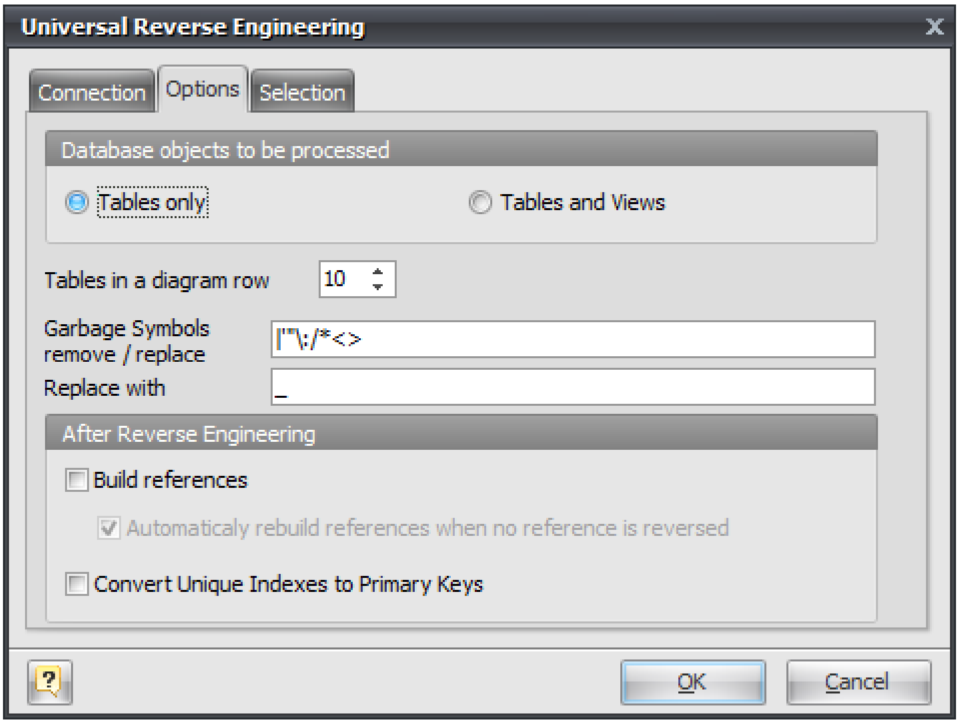


Select the data you want to connect to:

OLE DB Provider(s)
Microsoft Jet 4.0 OLE DB Provider
Microsoft Office 12.0 Access Database Engine OLE DB Pro
Microsoft OLE DB Provider for Analysis Services 9.0
Microsoft OLE DB Provider For Data Mining Services
Microsoft OLE DB Provider for Indexing Service
Microsoft OLE DB Provider for ODBC Drivers
Microsoft OLE DB Provider for OLAP Services 8.0
Microsoft OLE DB Provider for Oracle
Microsoft OLE DB Provider for Search
Microsoft OLE DB Provider for SQL Server
Microsoft OLE DB Simple Provider
MSDataShape
OLE DB Provider for Microsoft Directory Services

To use ODBC drivers, click on the **Microsoft OLE DB Provider for ODBC Drivers**. To use the native OLEDB provider, click on the appropriate item in the list. Browse through tabs to set other tabs and connection properties. Click on the **Help** button in the bottom of the dialog window to learn more about OLEDB connections. Click on the **OK** button to store the data link properties in the **Connection string** field of the **Universal Reverse Engineering** dialog.

4. You can set reverse engineering options in the **Options** tab:



Database objects to be processed

● Tables only ○ Tables and Views

Tables in a diagram row: 10

Garbage Symbols remove / replace: |'"\:/*<>

Replace with: _

After Reverse Engineering

□ Build references

☑ Automaticaly rebuild references when no reference is reversed

□ Convert Unique Indexes to Primary Keys

Tables only
Reverse engineer only tables ignoring views.

Tables and Views
Reverse engineer both tables and views.

Tables in a diagram row
This option determines how reversed tables will be disposed on the diagram. Reversed tables will be placed on the diagram in rows with equal distance, this option determines how many tables maximum there will be in one row.

Garbage symbols remove/replace
Defines the symbols that will be replaced in the names of the objects.

Replace with
Defines the garbage replacement symbol.

Build references
This option enables extracting foreign keys from the database and creating appropriate references in your diagram.

Automatically rebuild references when no reference is reversed
If there are no physical references extracted, it is possible to build them from logical structure of the database.

Enabling this feature leads to automatic reconstruction of references. Such reconstruction works by the following scheme: each column of the table is being compared with all primary keys of other tables, and if the column name and data type match one of the primary keys, a reference between the source column and the key column will be created.

This option is available for modification only if Build references is checked.

Convert Unique Indexes to Primary Keys
There no primary keys in some databases, but unique indexes used instead. You can use this optin to convert such unique indexes to primary keys.

5. In the **Selection tab** of the **Universal Reverse Engineering** tool you can choose the tables you want to reverse engineer.

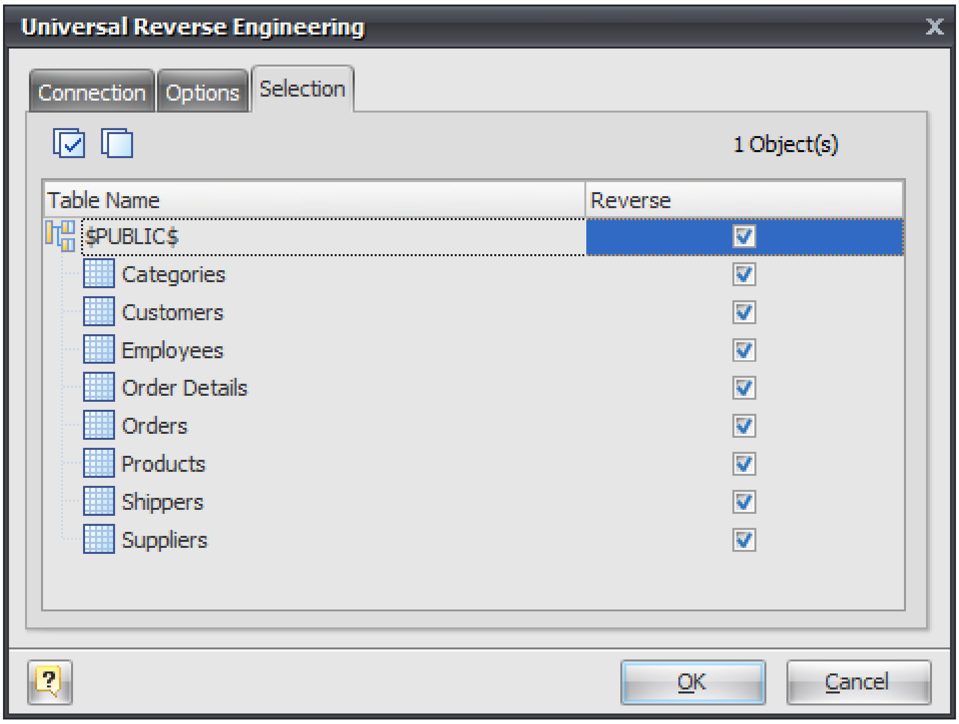


Table Name	Reverse
$PUBLIC$	☑
Categories	☑
Customers	☑
Employees	☑
Order Details	☑
Orders	☑
Products	☑
Shippers	☑
Suppliers	☑

Click on the checkbox near the table to enable its reverse engineering. Pay attention to the **Select All** (☑) and **Deselect All** (□) buttons on the tab, they allow you to select/deselect all tables in the list.

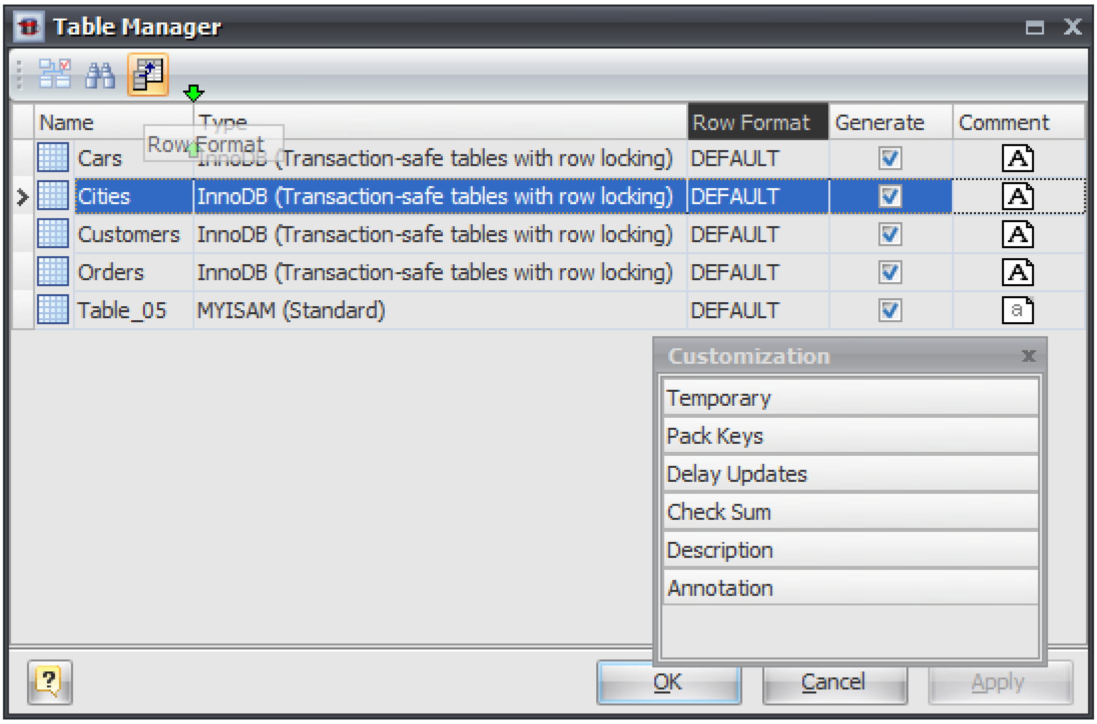
6. Click **OK** to start the database reverse engineering process. The **Output -> Reverse** docking window will display the state of the process.

7. The reversed database objects will be placed in your diagram.

See also:
Reverse Engineering and Import: Reverse Engineering and Import Overview | Reverse Engineering MySQL Database | Import from Access Database

14.14. How to Modify Multiple Tables

Database Designer for MySQL has a great feature that allows you to modify the basic parameters for multiple tables at once. This is the **Table Manager**. To run this tool use **Table Manager** item on **Diagram** tab of Ribbon or press **Ctrl+2**.



Name	Type	Row Format	Generate	Comment
Cars	InnoDB (Transaction-safe tables with row locking)	DEFAULT	☑	A
Cities	InnoDB (Transaction-safe tables with row locking)	DEFAULT	☑	A
Customers	InnoDB (Transaction-safe tables with row locking)	DEFAULT	☑	A
Orders	InnoDB (Transaction-safe tables with row locking)	DEFAULT	☑	A
Table_05	MYISAM (Standard)	DEFAULT	☑	a

Temporary
Pack Keys
Delay Updates
Check Sum
Description
Annotation

Modifying parameter values

The grid in **Table Manager** represents the basic parameters of all diagram tables. The rows stand for the tables, and columns for the table parameters. This parameters can be changed by using Table Editor for each table one by one, but with **Table Manager** you can do it much more quickly.

Click on the cell to modify the parameter value. Depending on the parameter type, the activated in-place editor can be a text box, a drop-down list, a check box, etc. Please refer to the Table Editor topic to find out more about all table parameters.

To save your changes click the **OK** button. If you want to store changes and continue editing, click on the **Apply** button.

Customizing the parameter visibility

By default, there are only a few table parameters accessible in the grid, but you can make other table parameters visible as well. Click on the **Customize** button ( ) in the dialog toolbar for this purpose. The **Customize** tool window will appear.

The **Customize** tool window contains the list of table parameters, which are inaccessible in the main grid. To add a new column to the grid, select the required parameter in the list and drag to the new position in the grid. Two green arrows will indicate that dropping is allowed, and after you drop it, the new column will appear. To remove a column from the grid, drag the column header to the **Customize** list in the similar way.

Thus you can change the following tables parameters in the **Table Manager**:

Name

Type

Row Format

Generate

Comment

Temporary

Pack Keys

Check Sum

Description

Annotation.

Please refer to the Table Editor topic to find out more about this parameters.

Searching table in the list

If your diagram contains large number of tables, it may be important to search for a table in a most easy way. To find a table quickly by its name, click the **Find** button () at the dialog toolbar. The **Find Name** dialog will appear. Type in the name of table to find and click **OK**. If the table is found the grid cursor positions on the appropriate row.

Running the Table Editor

If you want to change some unavailable table parameters and attributes (such as columns and indexes), select the appropriate table in the dialog and click the **Edit** button () of the dialog toolbar. The Table Editor will appear, where you can edit all the parameters for the selected table.

See also:
Diagram Objects: Table Editor

14.15. How to Execute an SQL Script

With **SQL Executor** you can send SQL queries to the connected MySQL database and display the result. The queries can contain any possible statements, e.g. UPDATE, DELETE, INSERT, SELECT statements etc. Note, that it is possible to run multiple SQL queries simultaneously.

The result of the SELECT-containing queries will be shown in the grid-based dialogues.

To execute SQL queries:

1. Connect to the database using Database Connection Manager
or
Use the already established database connection.

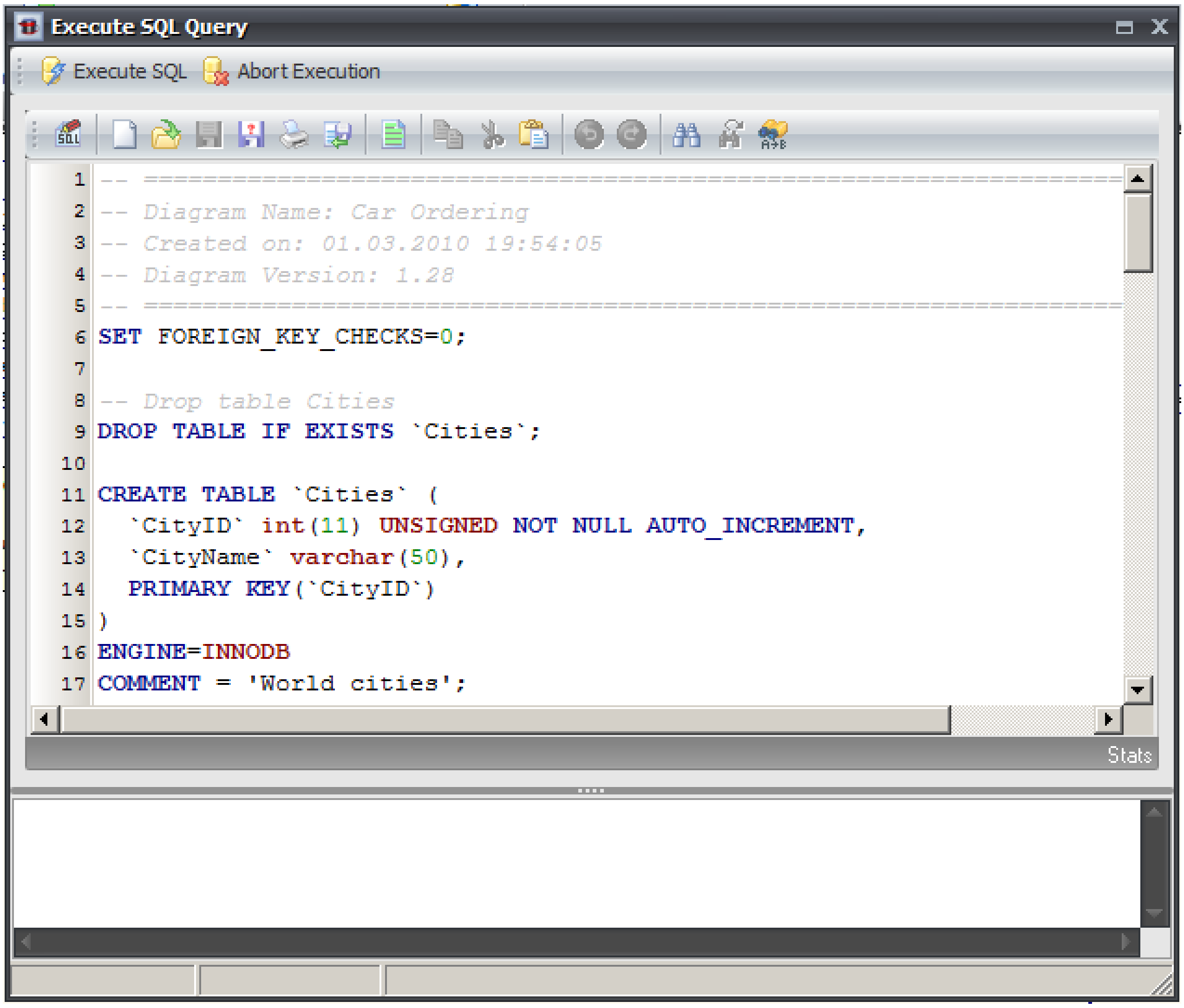
2. Call the **SQL Executor** by selecting **SQL Executor** (⚡) item on **Database tab** of the Ribbon or pressing **Ctrl+Shift+E**.

3. Enter one or more SQL queries.

4. Press **F9** or click the **Execute SQL** (⚡) button on the dialog window toolbar to execute your queries. A data grid window will be displayed for each SELECT-based query. To close the data grid window, click on the **Close** button. The execution status for each query will be displayed at the bottom of the **SQL Executor** dialog window.

5. You can save your queries into a file. Click on the **Save** (💾) button on the **SQL Executor** toolbar.

Please note, that **SQL Executor** dialogue window is used by Database Generation and Database Modification tools.



```
-- ======================================================================
-- Diagram Name: Car Ordering
-- Created on: 01.03.2010 19:54:05
-- Diagram Version: 1.28
-- ======================================================================
SET FOREIGN_KEY_CHECKS=0;

-- Drop table Cities
DROP TABLE IF EXISTS `Cities`;

CREATE TABLE `Cities` (
  `CityID` int(11) UNSIGNED NOT NULL AUTO_INCREMENT,
  `CityName` varchar(50),
  PRIMARY KEY(`CityID`)
)
ENGINE=INNODB
COMMENT = 'World cities';
```

See also:
Database Functions: Database Generation | Database Modification

14.16. How to Merge Diagrams

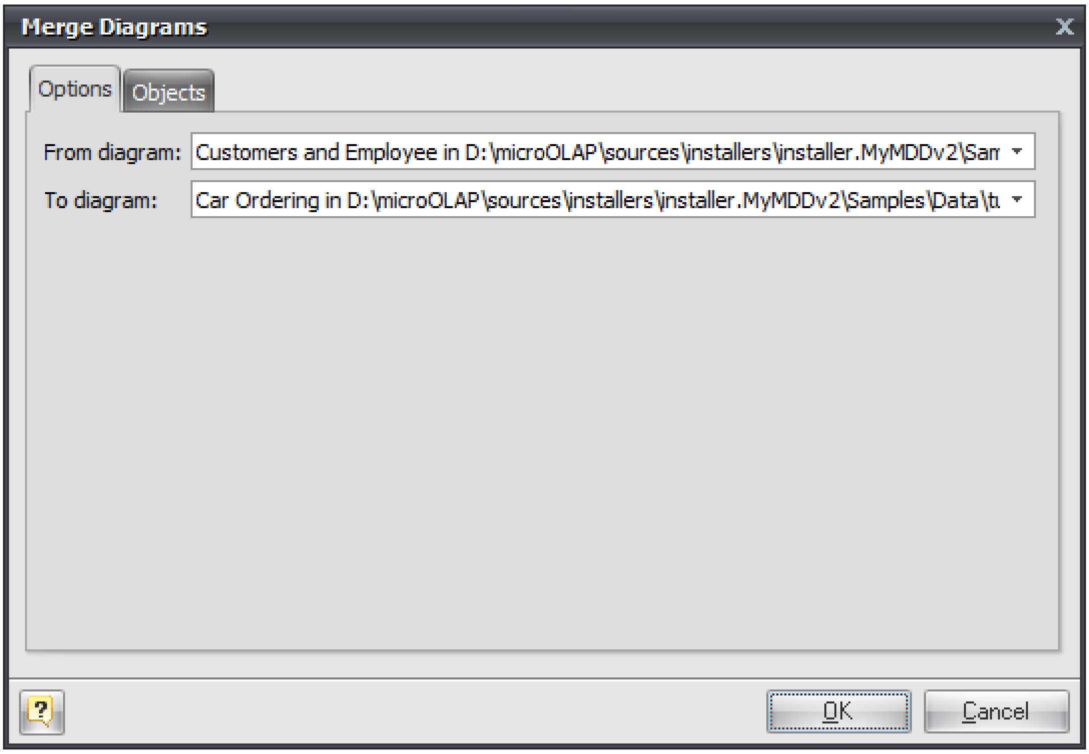
The **Merge Diagram** tool of the **Database Designer for MySQL** allows you to merge content of two diagrams. This allows you to create new cumulative diagram, which will include content of your two diagrams.

To merge two diagrams, please follow these steps:

1. Open two diagrams you want to merge. One of them will accumulate its own content and content of other diagram.

2. Start Merge Diagram tool by selecting the Merge Diagram ( ) item on Diagram tab of the Ribbon.

3. Select diagrams you want to merge.



From diagram: Customers and Employee in D:\microOLAP\sources\installers\installer.MyMDDv2\San

To diagram: Car Ordering in D:\microOLAP\sources\installers\installer.MyMDDv2\Samples\Data\t

From diagram
Choose the source diagram from drop-down menu, which contains list of opened diagrams. Objects of this diagram will be added to the destination diagram.

To diagram
Choose the destination diagram from the drop-down menu, which contains the list of the opened diagrams. This diagram will contain its own objects and objects of source diagram.

4. Select the source diagram objects to merge. Go to the Objects tab.

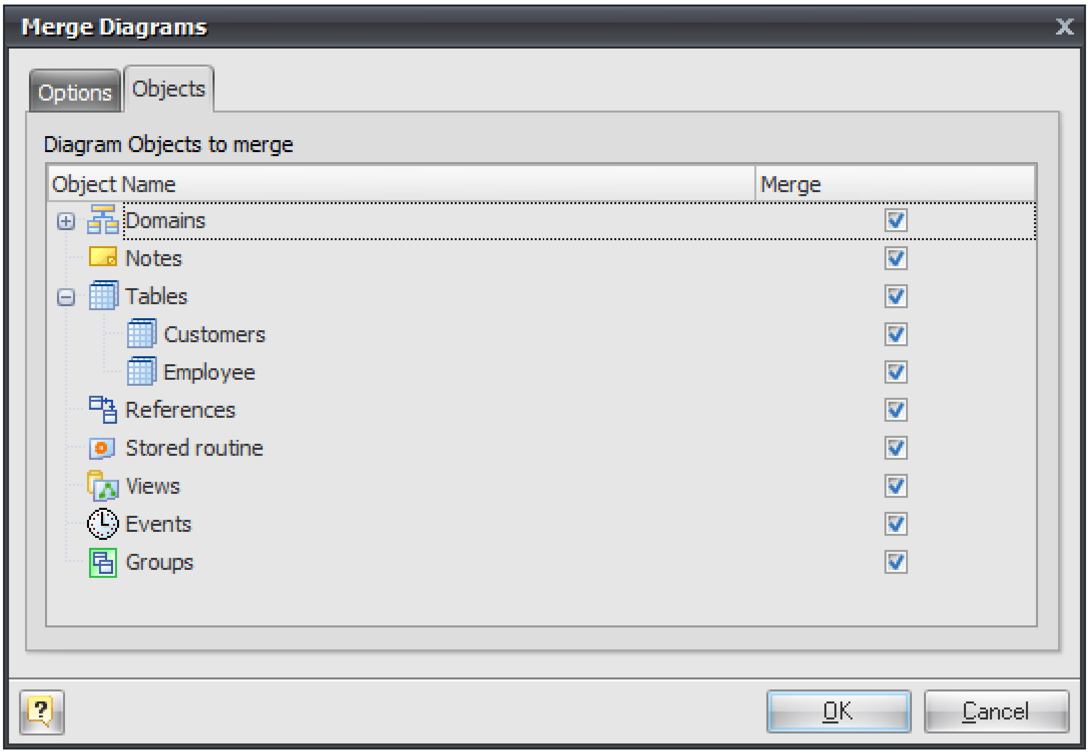


Diagram Objects to merge

Object Name	Merge
Domains	☑
Notes	☑
Tables	☑
Customers	☑
Employee	☑
References	☑
Stored routine	☑
Views	☑
Events	☑
Groups	☑

5. Click on the appropriate checkboxes in Merge column to select objects you want to add to the destination diagram. To select/deselect all objects of particular type, click on appropriate checkbox near them (by default all objects are already selected).

6. Click OK to add the selected objects of the source diagram to the destination diagram.

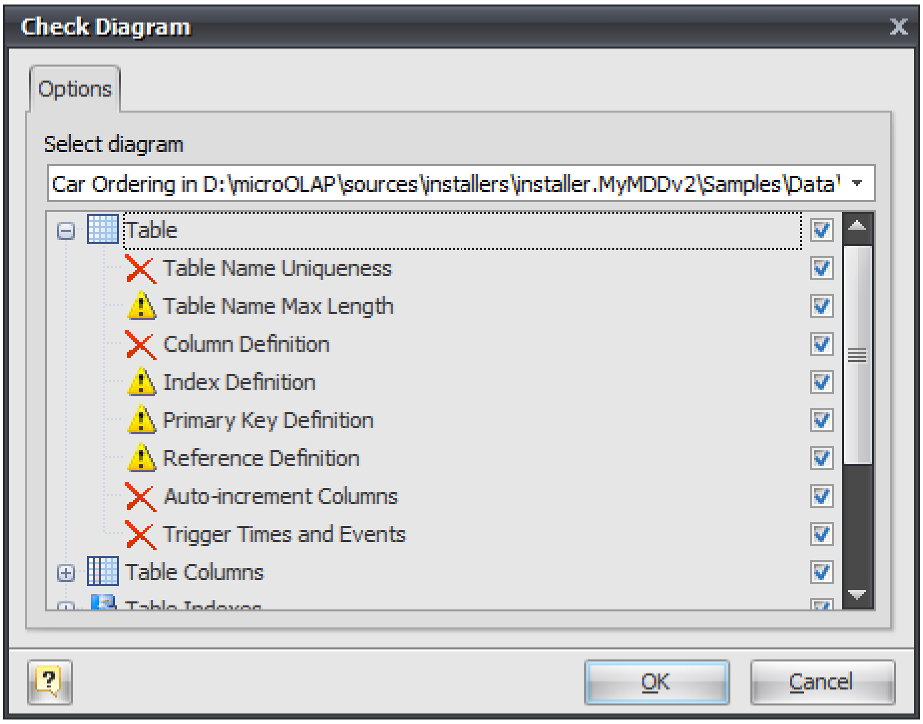
7. (optional) Save the resulted (destination) diagram to a new file.

See also:
Diagram: Open an Existing Diagram | Saving a Diagram

14.17. How to Find Errors in a Diagram

The **Check Diagram** tool of the **Database Designer for MySQL** allows you to check your database diagram for most typical errors and defects. The result of the check goes in a well structured form, using which you can easily bring your diagram to correspondence with the common standards of database modeling.

To open the **Check Diagram** dialog use the **Check Diagram** ( ) item on **Diagram** tab of the Ribbon or press **F4**.



Select diagram

Car Ordering in D:\microOLAP\sources\installers\installer.MyMDDv2\Samples\Data'

- Table ☑
  - Table Name Uniqueness ☑
  - Table Name Max Length ☑
  - Column Definition ☑
  - Index Definition ☑
  - Primary Key Definition ☑
  - Reference Definition ☑
  - Auto-increment Columns ☑
  - Trigger Times and Events ☑
- Table Columns ☑
- Table Indexes ☑

Use the **Select diagram** drop-down list to select the diagram from the list of currently opened diagrams. Current opened diagram is selected by default.

The tree list below allows you to select what warnings and errors should be taken into account during the check. All warnings and errors are divided into categories, which correspond to the diagram objects. Remove selection from the warning/error or from the whole category to exclude it from the check.

These are the descriptions for all available warnings and errors.

Table

Error "Table Name Uniqueness"
Check the diagram for the uniqueness of each table name within a diagram.

Warning "Table Name Max Length"
MySQL allows only 64 characters in table names and cuts names if they are longer than this.

Error "Column Definition"
Check if each table within the diagram owns at least one column.

Warning "Index Definition"
Check if each table within the diagram owns at least one index.

Warning "Primary Key Definition"
Check if a primary key is defined for each table within the diagram.

Warning "Reference Definition"
Check if each table within the diagram is linked with other tables.

Error "Auto-increment columns"

Check if each table within the diagram has no more than one auto-increment column, as otherwise MySQL will not allow such table to be created.

Error "Trigger Times and Events"
Check if every trigger in each table within diagram has unique Time and Event combination.

Table Columns

Error "Column Name Uniqueness"
Check diagram tables for the uniqueness of each column name within the table.

Warning "Column Name Max Length"
MySQL allows only 64 characters in column names and cuts names if they are longer than this.

Warning "Auto-increment Column Definition"
Check if each auto-increment column within a table is a part of a primary key.

Table Indexes

Error "Index Name Uniqueness"
Check diagram tables for the uniqueness of each index name within the table.

Warning "Index Name Max Length"
MySQL allows only 64 characters in index names and cuts names if they are longer than this.

Warning "Duplicate Index Column"
Check if each table column is indexed only once.

References

Warning "Reference Table Type"
Check if both linked tables use storage engine that supports references (for example InnoDB or Falcon) for each diagram reference.

Error "Reference Column Data Types"
Check whether the linked columns are of the same data type for each diagram reference.

Error "Reference Column does not exists"
Check an existence of referenced column in parent table for each diagram reference.

Error "Reference Name uniqueness"
Check diagram references for uniqueness of each reference name.

Error "Check "SET NULL" FK-action for NOT NULL column"
MySQL does not allow SET NULL action for foreign keys that reference column marked with NOT NULL attribute.

Domains

Error "Domain Name Uniqueness"
Check diagram for the uniqueness of each domain name within a diagram.

Stored Routine

Error "Stored Routine Name Uniqueness"
Check diagram stored procedures and functions for the uniqueness of name within a schema.

Warning "Stored Routine Name Max Length"
MySQL allows only 64 characters in stored routine names and cuts names if they are longer than this

Views

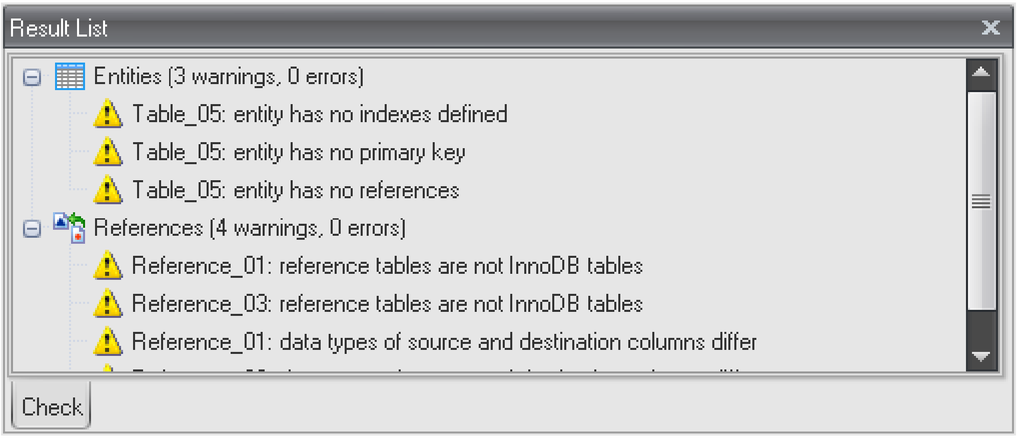
Error "View Name Uniqueness"
Check diagram stored procedures and functions for the uniqueness of name within a schema.

Warning "View Name Max Length"
MySQL allows only 63 characters in stored routine names and cuts names if they are longer than this.

Error "View was created on a not existing table"
Check existence of tables, which are used in the view.

Error "View was created on a not existing column"
check existence of table columns, which are used in the view.

After you click **OK** the check process will be displayed within the **Output** window and the result of the check will be displayed within the **Result** window in the same categorized view as described above.



Entities (3 warnings, 0 errors)
Table_05: entity has no indexes defined
Table_05: entity has no primary key
Table_05: entity has no references
References (4 warnings, 0 errors)
Reference_01: reference tables are not InnoDB tables
Reference_03: reference tables are not InnoDB tables
Reference_01: data types of source and destination columns differ

Double-click on a warning or an error in the list opens the editor window for the appropriate object (Table Editor, View Editor, Domain Manager and so on).

See also:
Interface User Guide: Docking Windows
Diagram Objects: Table Editor | Column Editor | Domain Manager | Reference Editor | Index Editor | Stored Routine Editor | Trigger Editor | View Editor

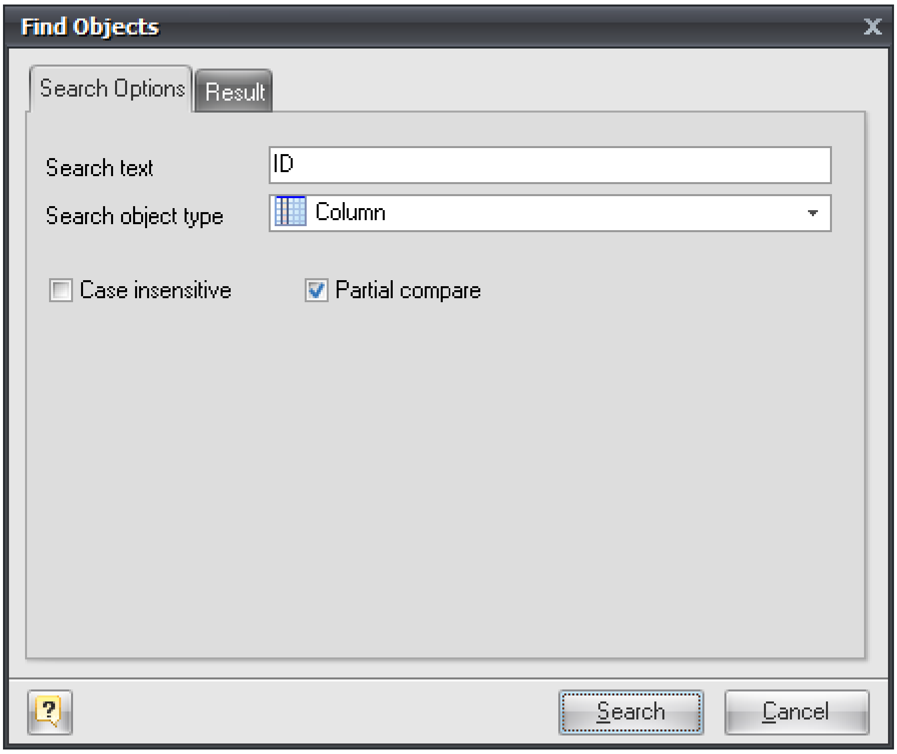
14.18. How to Find Objects

The **Find Objects** feature lets you search objects within the entire **Database Designer for MySQL** diagram so that you could locate the necessary objects in the diagram easily, find all the attributes related to a given object.

To call **Find Objects** dialog window, click the **Find Objects** (🔍) item on **Main** tab of Ribbon or press **Ctrl-F**.

Defining find parameters

You can define the search parameters in the **Find Objects** dialog window.



Search text: ID

Search object type: Column

□ Case insensitive ☑ Partial compare

Search text
This options allows you to define the text that you want to search in the objects and attributes names.

Search object type
You can define the object type that you want to search for. Also you can search for any type of objects. You can select the following object types:

- Table
- Column
- Index
- Trigger
- Reference
- View
- Stored Routine

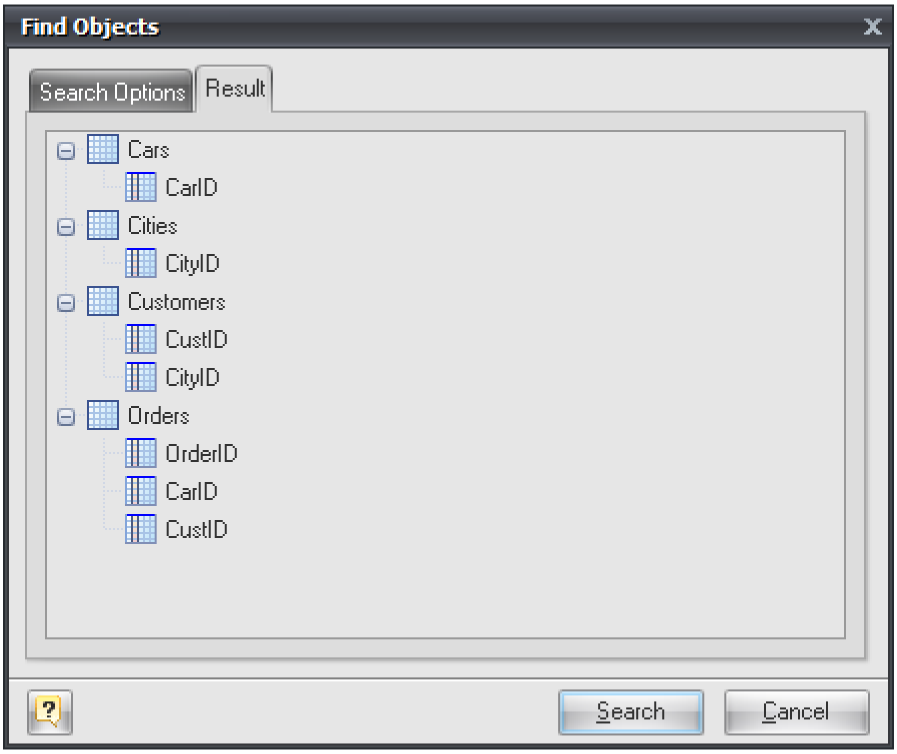
- Event.

Case insensitive
If you enable this option case matching will be off and it will not affect the results of your search.

Partial compare
If you enable this option you will find objects with partial name matching.

Using the Result List

The **Result** tab displays the result of the search in the result tree.



You can use the result tree to:

- learn to which objects the found objects (attributes) belong to;
- modify the found objects

Double click on the object in the result tree to call an appropriate editor for them.

14.19. How to Print a Diagram

Printing a database diagram gives you a picture of your database structure to refer to or distribute.

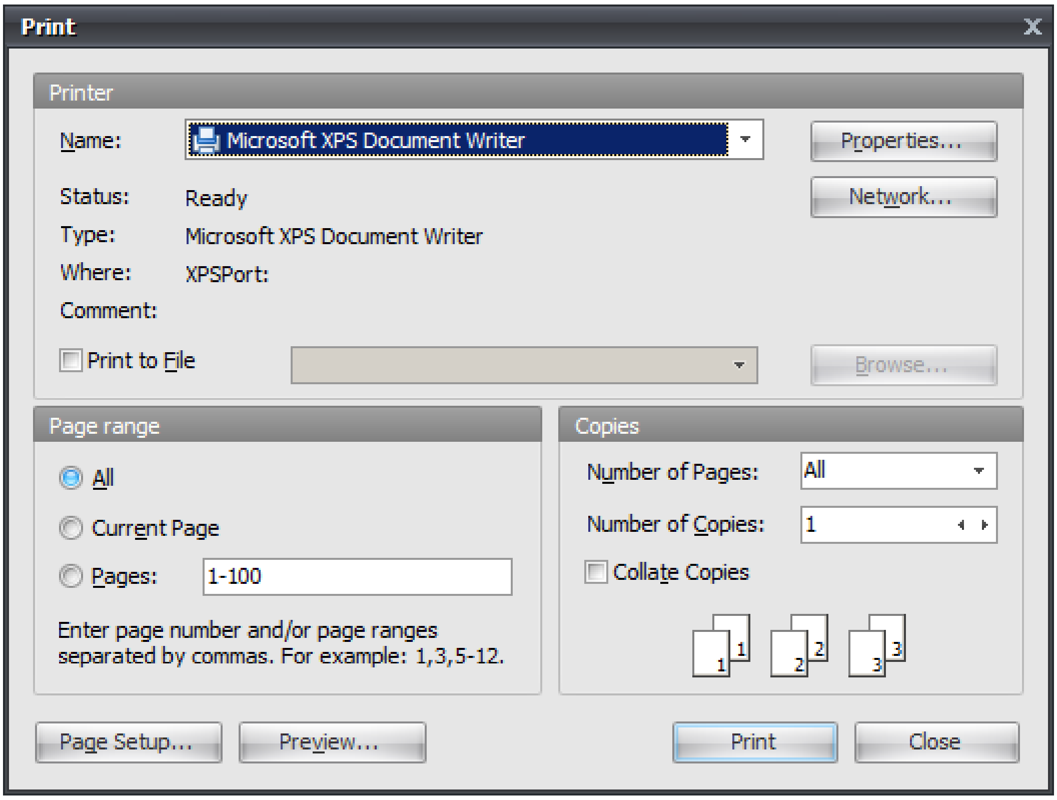
Before you start printing, arrange the objects in the database diagram to your satisfaction. You can change the shape, and position of the objects in the diagram without affecting their definitions in the database.

To change the layout of the diagram, move, size, and shape the objects. For example, you can use the mouse to move tables, or use the Auto Layout Diagram command to automatically reposition the objects.

You can change physical parameters of your paper, on which diagram will be printed out, specify margins, page headers and footers information using Page Setup dialog.

Before printing out a diagram, you can see how it will look paper, use the Print Preview tool for that.

To print out a diagram, select the **Print** (🖨) menu item from Application menu or press **Ctrl+P**.



Printer

Name: Microsoft XPS Document Writer

Status: Ready

Type: Microsoft XPS Document Writer

Where: XPSPort:

Comment:

□ Print to File

Page range

● All

○ Current Page

○ Pages: 1-100

Enter page number and/or page ranges separated by commas. For example: 1,3,5-12.

Copies

Number of Pages: All

Number of Copies: 1

□ Collate Copies

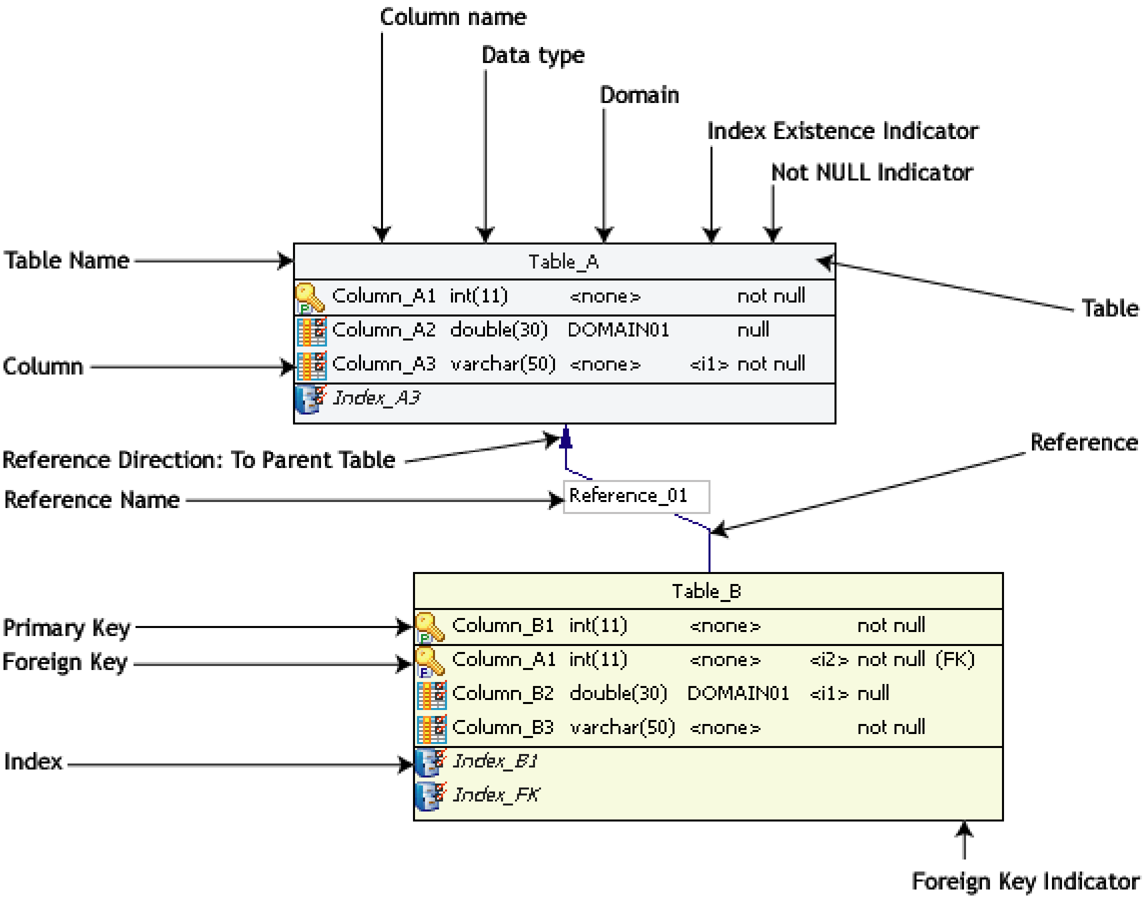
The **Print Setup** dialog is displayed. In the **Print Setup** dialog select the printer you want to print on, and then click the **OK** button to print a diagram.

See also:
Printing a Diagram: Page Setup | Print Preview

14.20. How to Read a Diagram

Diagram notation

The following picture describes the diagram objects notation:


Column name
Data type
Domain
Index Existence Indicator
Not NULL Indicator
Table Name
Table
Column
Reference Direction: To Parent Table
Reference Name
Reference
Primary Key
Foreign Key
Index
Foreign Key Indicator
Table_A
Column_A1 int(11) <none> not null
Column_A2 double(30) DOMAIN01 null
Column_A3 varchar(50) <none> <i1> not null
Index_A3
Reference_01
Table_B
Column_B1 int(11) <none> not null
Column_A1 int(11) <none> <i2> not null (FK)
Column_B2 double(30) DOMAIN01 <i1> null
Column_B3 varchar(50) <none> not null
Index_B1
Index_FK

This way of diagram objects displaying is the most informative. You can change the displaying preferences according to your likes and dislikes. Please, see the Diagram Display Preferences section for more information.

See also:
Entity Relationship Diagram: Diagram Display Preferences

14.21. How to View an SQL Table Definition

Table SQL preview

You can see the SQL representation of your table, it includes columns, indexes, foreign keys, etc. Actually this is SQL-code to be executed while generating database.

To see SQL representation of the table, double click on the table symbol and Table Editor will appear. Go to the **Preview** tab. SQL representation is shown using standard Text Editor. So you can save to file, copy to clipboard, print SQL-code and so on.

See also:

Diagram Objects: Table Editor

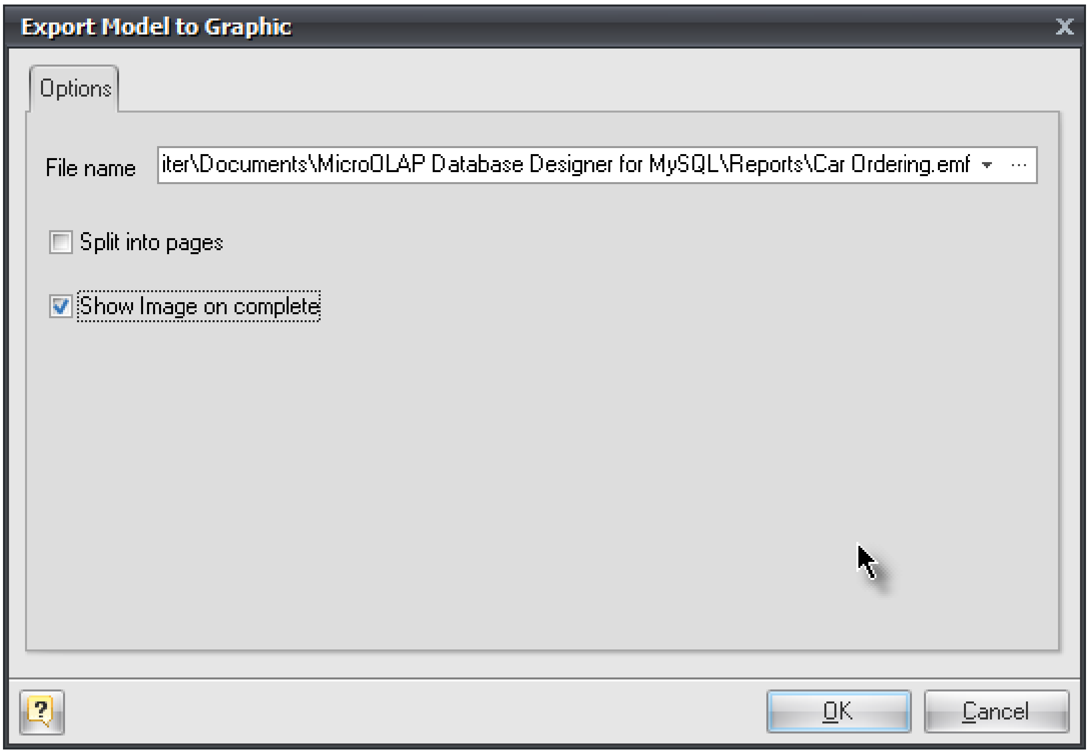
14.22. How to Export a Diagram to Graphics

You can export your diagram to an image file and then use its graphical representation in external applications. For example, insert a diagram image into your program documentation (e.g. use MS Word), create big posters with your diagram, publish them to Web.

Database Designer for MySQL supports the following image types:

- Vector graphics: Windows Enhanced Metafile (EMF).
- Bitmap images: PNG, GIF, JPEG, BMP.

Use the **Application Menu | Export** menu item. Export Model to Graphic dialog will appear.



File name: iter\Documents\MicroOLAP Database Designer for MySQL\Reports\Car Ordering.emf

□ Split into pages

☑ Show Image on complete

Select file name and type in **Select image file** dialog showed by "**...**" button in **File name** editor.

You can also check **Split into pages** and/or **Show image on complete** options.

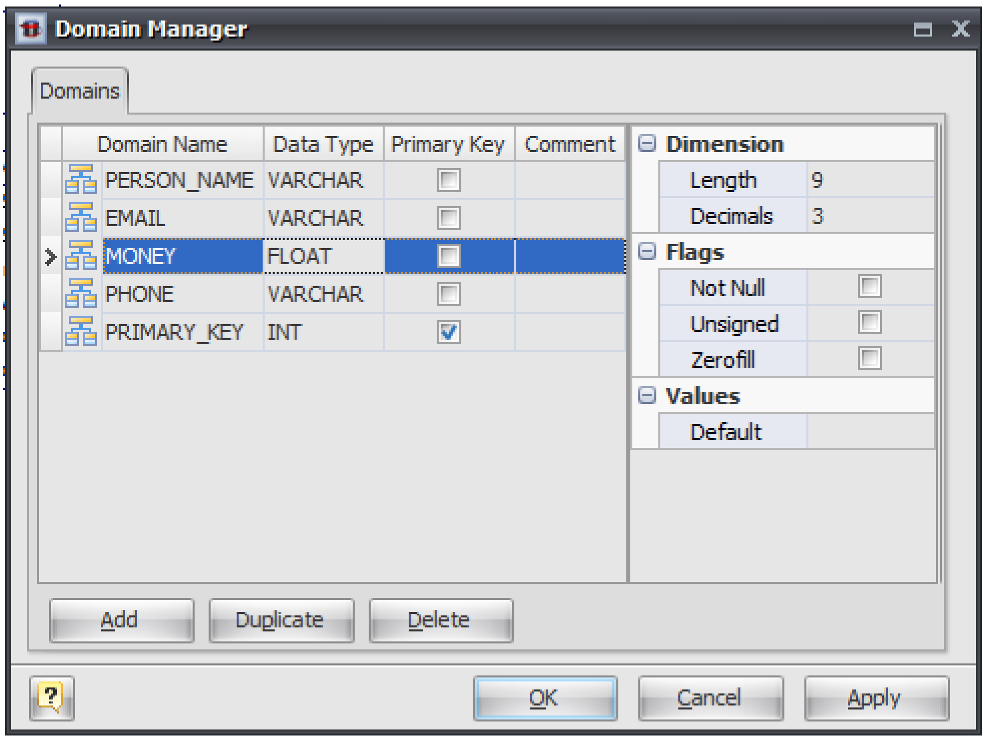
Click **OK** to save the file.

> **Please note:**
> Exporting a diagram to the vector graphics format allows you to freely scale a diagram of any size without any noticeable distortions. It enables you to create big posters with the diagram to demonstrate the data logic of your application.

14.23. How to Create a Domain

The **Domain Manager** is intended for managing diagram domains, which can be used for faster creating and modifying table columns.

To open the **Domain Manager** use the **Domain Manager** item on **Diagram** tab of Ribbon or press **Ctrl + 1**.



Domain Name	Data Type	Primary Key	Comment
PERSON_NAME	VARCHAR	□	
EMAIL	VARCHAR	□	
MONEY	FLOAT	□	
PHONE	VARCHAR	□	
PRIMARY_KEY	INT	☑	

Dimension

Length	9
Decimals	3

Flags

Not Null	□
Unsigned	□
Zerofill	□

Values

Default	

The **Domain Manager** consists of the following areas:

- Domain List
- Properties Pane
- Button Pane.

Domain List

The domain list displays all the domains in the diagram and allows you to modify the following domain properties:

Domain name
The name of the domain, which must be unique within the diagram.

Data type
The type of the domain, which specifies data to store in the domain-based column; it can be one of integer types, which differ in data size (TINYINT, SMALLINT, MEDIUMINT, INT, BIGINT), decimal type (FLOAT, DOUBLE, DECIMAL), a type for storing date and time values (DATE, DATETIME, TIMESTAMP, TIME, YEAR), string type (CHAR, VARCHAR), a large binary type (TINYBLOB, BLOB,

MEDUIMBLOB, LONGBLOB), a multi-line text (TINYTEXT, TEXT, MEDIUMTEXT, LONGTEXT), enumeration type, which can contain only values from the Set attribute (ENUM), or a set of values from the Set attribute (SET).

Primary key
Specify this option to include the domain-based columns into the table primary key.

Comment
An arbitrary description for the domain.

Properties Pane

The properties pane allows you to define the advanced properties of the domain, selected in the **Domain List**. The appearance of this pane changes according to the data type of the domain. These properties are:

Not null
This option indicates that a domain-based column value cannot be NULL;

Default
This attribute defines the default value, which the domain-column accepts if no other is specified;

Decimals
This attribute defines the number of digits, which follow the decimal point in the FLOAT, DOUBLE, and DECIMAL types.

Length
This attribute defines the maximum allowed length of the stored values; it applies to all integer, decimal, and string types.

Binary
This attribute indicates that the string value is stored in a binary form within the domain.

Unsigned
This option indicates that the stored integer or decimal value must be above zero.

Zerofill
With this attribute all unspecified digits of the domain-column value are filled as zeros according to the column size and precision; it applies to all integer and decimal types.

Unique
This option indicates that the domain-column value must be unique within the table, creating a unique index for the domain.

Autoinc
This attribute makes the domain-column value autoincrement, i.e. each new value is set automatically according to the previous value; it applies to all integer values.

Set
This attribute defines the list of permissible values for the domain of the ENUM or SET type.

Buttons Pane

The buttons under the list of domains allows you to perform the following actions:

Add
Add a new domain with the default properties to the end of the list;

Duplicate
Add a new domain with the same properties as the selected domain to the end of the list;

Delete
Remove the selected domain from the list.

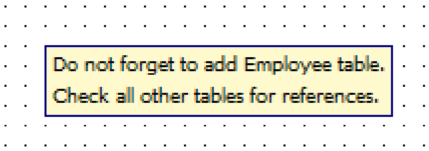
See also:
Diagram Objects: Domains | Columns
Tutorials: Using Domains

14.24. How to Insert a Comment

Notes

You can type in the **Note box** any comment concerning your diagram. **Notes** are useful for writing various comments, such as object functionality or role description, things to do, etc in your diagram.

The following picture demonstrates a sample **Note**:



Do not forget to add Employee table.
Check all other tables for references.

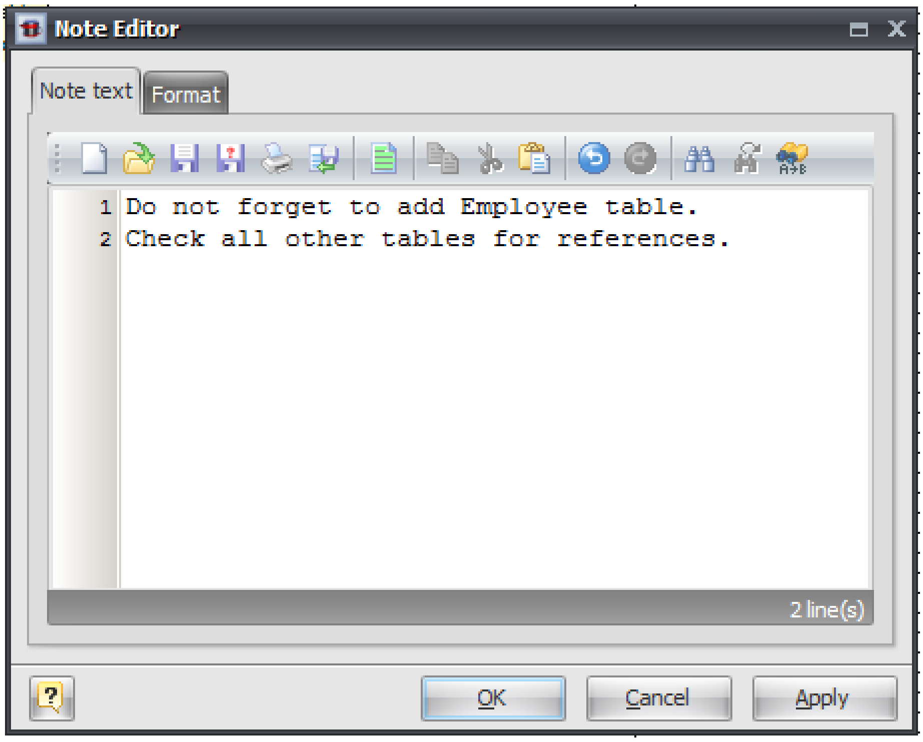
> **Please note:**
> **Note** is auxiliary object, it doesn't affect database generation.

Placing Note to the diagram

To place a note in the diagram, select the **Note** ( ) item on **Main** tab of the Ribbon. Then click anywhere in the diagram. Certainly, you can place multiple notes in one diagram.

Modify the text in Note

To modify the information, that is displayed in **Note**, double click on it. Then the **Note Editor** will appear:

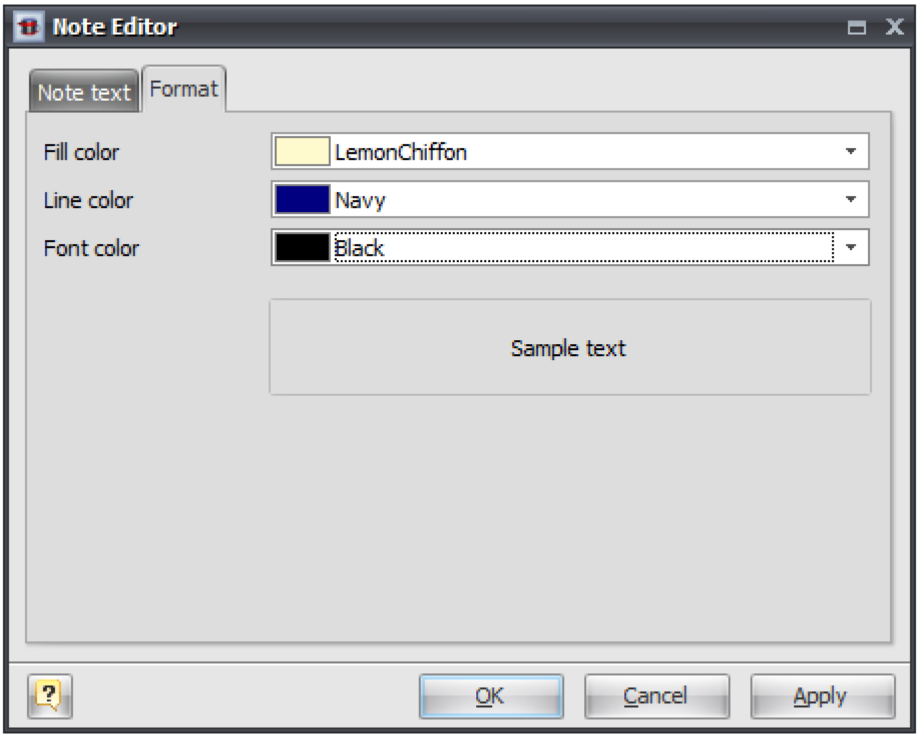


```
Do not forget to add Employee table.
Check all other tables for references.
```

2 line(s)

You can edit, save or load **Note** text. Press **OK** to apply your changes and close this dialog.

Formatting Note



Fill color: LemonChiffon
Line color: Navy
Font color: Black

Format tab of **Note Editor** allows you to adjust **Note** appearance:

Fill color;

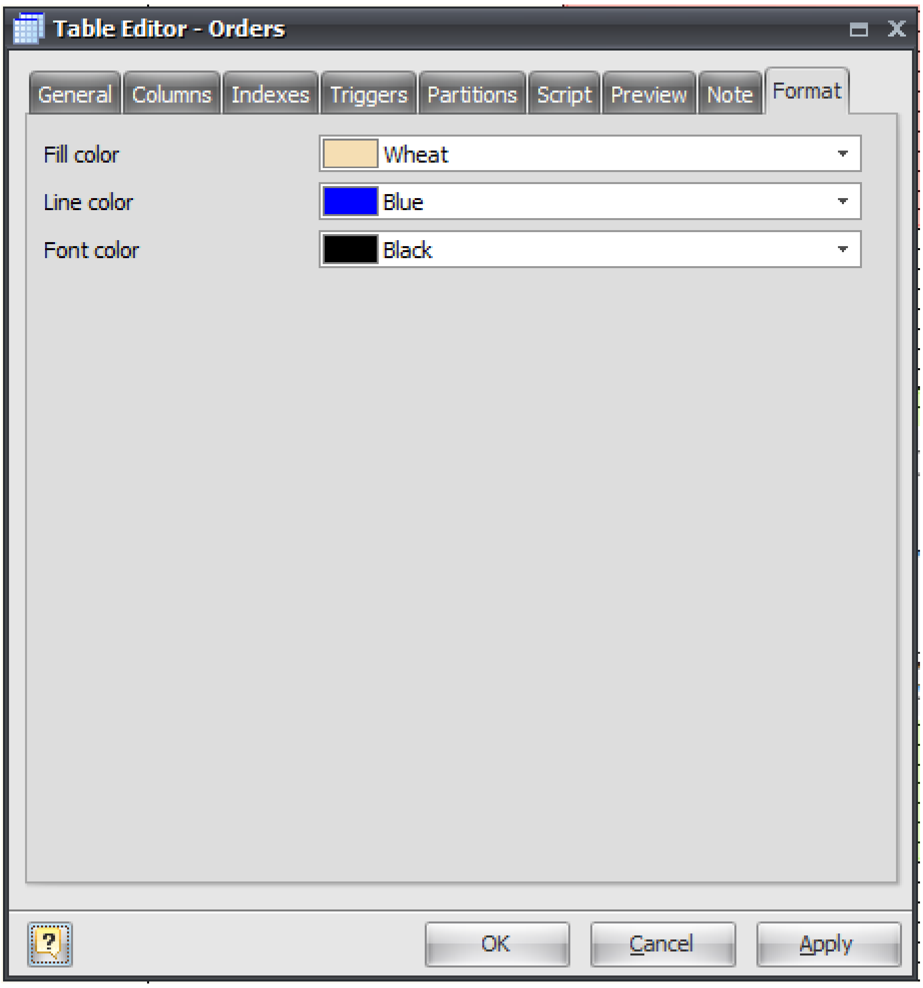
Line color;

Font color.

14.25. How to Change Table Formatting

You can set table line and fill color for displaying on the diagram, different from the default table colors, which are defined within the Diagram Display Preferences dialog.

Double click on the table symbol on diagram and Table Editor will appear. Go to the **Format** tab.



Fill color: Wheat

Line color: Blue

Font color: Black

Fill Color
Use this option to set a background color of the table. Choose the color you need from this drop-down list.

Line Color
Use this option to set a border color of the table. Choose the color you need from this drop-down list.

Font Color
Use this option to set a font color of the table. Choose the color you need from this drop-down

list.

See also:
Diagram Objects: Table Editor
Diagram: Diagram Display Preferences

15. Installation

To install **Database Designer for MySQL** on your PC:

- **Unzip** the downloaded file to any location you prefer;
- Run the **.exe** file and follow the installation application instructions.

The unregistered version of **Database Designer for MySQL** (30-days trial) has the same features as the registered one.

> **Please note:**
> Before installation of **Database Designer for MySQL**, please check for updates at MicroOLAP.com

See also:
Purchase

16. Purchase

Thank you for your interest in purchasing Database Designer for MySQL!

You can select the purchasing method (CC online, Purchase Order, Fax or wire transfer) and licensing options you prefer and also register for **Database Designer for MySQL** at http://www.microolap.com/products/database/mysql-designer/order/.

17. Support

We provide our customers with efficient and high-professional support, which is absolutely free. Whether you have any questions about **Database Designer for MySQL** or you want to report a bug, feel free to do it by using our on-line Support System.

18. License Agreement

NOTICE TO USER:

THIS IS AN AGREEMENT GOVERNING YOUR USE OF THE SOFTWARE TITLED **Microolap Database Designer for MySQL**, FURTHER DEFINED HEREIN AS "PRODUCT," AND THE LICENSOR OF THE PRODUCT IS WILLING TO PROVIDE YOU WITH ACCESS TO THE PRODUCT ONLY ON THE CONDITION THAT YOU ACCEPT ALL OF THE TERMS AND CONDITIONS CONTAINED IN THIS AGREEMENT. BELOW, YOU ARE ASKED TO ACCEPT THIS AGREEMENT AND CONTINUE TO INSTALL OR, IF YOU DO NOT WISH TO ACCEPT THIS AGREEMENT, TO DECLINE THIS AGREEMENT, IN WHICH CASE YOU WILL NOT BE ABLE TO INSTALL OR OPERATE THE PRODUCT. BY ACKNOWLEDGING YOUR CONSENT HERETO AND BY INSTALLING AND OPERATING THIS PRODUCT YOU ACCEPT ALL THE TERMS AND CONDITIONS OF THIS AGREEMENT. For purposes hereof ***"Operating"*** shall mean accessing, storing, loading, installing, Using (as defined below), and copying the Product into the memory of a Client Device, as defined below. ***"Using"*** shall mean executing and displaying the product on a Client Device or otherwise benefiting from utilizing, deploying or using the Product or its functionality.

This Electronic End User License Agreement (the ***"Agreement"***) is a legal agreement between you (either an individual or an entity), the Licensee, and MicroOLAP Technologies Ltd., (the ***"Licensor"***), regarding the Product and related support service you are about to install and Operate and/or other related services, including without limitation:

a) all of the contents of the files, including disk(s), CD-ROM(s) or other media with which this Agreement is provided and including all forms of code, such as Source Code, Object Code, dynamic or static libraries, and/or executable files as provided and in a form that is provided by Licensor to you (the ***"Software"***). For the avoidance of doubt, by way of example, but not exclusion, if a specific file is provided by Licensor in Object Code only, the Source Code for such files shall not be deemed a part of the Software provided by Licensor to you. For purposes hereof ***"Source Code"*** shall mean the human-readable form of the computer programming code and related system documentation including all comments and any procedural code such as job control language and ***"Object Code"*** shall mean computer programs assembled or compiled in magnetic or electronic binary form on software media, which are readable and usable by machines, but not generally readable by humans without reverse-assembly, reverse-compiling, or reverse-engineering.

b) all support services provided to you by Licensor in connection with the Software (the ***"Services"***);

c) and all successor upgrades, modified versions, modified modules, revisions, patches, enhancements, fixes, modifications, copies, additions or maintenance releases of the Software, if any, licensed to you by the Licensor (collectively, the ***"Updates"***), and

d) related user documentation and explanatory materials or files provided in written, "online" or electronic form (the ***"Documentation"*** and together with the Software, Samples, Updates, and Services the ***"Product"***).

For the purposes of this Agreement, ***"Licensor Site"*** shall mean the Internet website maintained by or on behalf of Licensor from which the Software is available for download pursuant to a license from Licensor. The Licensor Site is currently located at ***http://www.microolap.com***.

You are subject to the terms and conditions of this End User License Agreement whether you access or obtain the Product directly from the Licensor, or through any other source. For purposes hereof, ***"you"*** or ***"Licensee"*** means the individual person installing or using the Product on his or her own behalf; or, if the Product is being downloaded or installed on behalf of an organization, such as an employer, ***"you"*** means the organization for which the Product is downloaded or installed, then the person accepting this agreement represents hereby that such organization has authorized such person to accept this agreement on the organization's behalf. For purposes hereof the term ***"organization,"*** without limitation, includes any partnership, limited liability company, corporation, association, joint stock company, trust, joint venture, labor organization, unincorporated organization, or governmental authority.

If you do not agree to the terms and conditions of this Agreement, the Licensor is unwilling to license the Product to you. In such event, you may not Operate the Product in any way.

IF THE LICENSOR AND YOU HAVE AGREED ON AND PROPERLY EXECUTED A SEPARATE CONTEMPORANEOUS OR SUBSEQUENT TERMS OF USE OR EXHIBITS (the *"Terms of Use"*), WHICH ARE SUPPLEMENTAL, DIFFERENT OR INCONSISTENT WITH THE TERMS OF THIS AGREEMENT, SUCH TERMS OF USE SHALL CONTROL, PROVIDED THAT (i) SUCH TERMS OF USE

SPECIFICALLY ACKNOWLEDGE AND REFER TO THIS AGREEMENT, AND (ii) ALL OTHER TERMS AND CONDITIONS OF THIS AGREEMENT REMAIN IN FULL FORCE AND EFFECT.

BEFORE YOU PUT A CHECKMARK AT THE "I ACCEPT THE AGREEMENT" BUTTON AND PRESS "NEXT," PLEASE CAREFULLY READ THE TERMS AND CONDITIONS OF THIS AGREEMENT, AS SUCH ACTIONS ARE A SYMBOL OF YOUR SIGNATURE AND BY CLICKING ON THE "I ACCEPT THE AGREEMENT" AND "NEXT" BUTTONS, YOU ARE CONSENTING TO BE BOUND BY AND ARE BECOMING A PARTY TO THIS AGREEMENT AND AGREE THAT THIS AGREEMENT IS ENFORCEABLE LIKE ANY WRITTEN NEGOTIATED AGREEMENT SIGNED BY YOU. IF YOU DO NOT AGREE TO ALL OF THE TERMS OF THIS AGREEMENT, CLICK THE "CANCEL" BUTTON AND THE PRODUCT WILL NOT BE INSTALLED ON YOUR CLIENT DEVICE, AS SUCH TERM IS DEFINED BELOW.

For your reference, you may refer to the copy of this Agreement that can be found in installed files of the Software as **license.rtf**.

You may also receive an electronic copy of this Agreement by contacting Licensor at ***sales@microolap.com***.

1. Proprietary Rights and Non-Disclosure.

1.1. Ownership Rights. You agree that the Product and the authorship, systems, ideas, methods of operation, derivative Documentation and other information contained in the Product, are proprietary intellectual properties and or the valuable trade secrets of the Licensor and are protected by applicable civil and criminal law, and by the law of copyright, trade secret, trademark and patent and international treaties. You may use trademarks only insofar as to identify printed output produced by the Product in accordance with accepted trademark practice, including identification of trademark owner's name. Such use of any trademark does not give you any rights of ownership in that trademark. The Licensor and its suppliers own and retain all right, title, and interest in and to the Product, including all copyrights, patents, trade secret rights, trademarks, and other intellectual property rights therein. Your possession, installation or use of the Product does not transfer to you any title to the intellectual property in the Product, and you will not acquire any rights to the Product except as expressly set forth in this Agreement. All copies of the Product made hereunder must contain the same proprietary notices that appear on and in the Product. Except as stated herein, this Agreement does not grant you any intellectual property rights in the Product.

1.2. Source Code and Modifications. You acknowledge that the source code for the Product is proprietary to the Licensor and constitutes trade secrets of the Licensor. Except as otherwise specifically provided herein or in Terms of Use, you agree not to disassemble, decompile or "unlock", decode or otherwise reverse-translate or reverse-engineer, or attempt in any manner to reconstruct or discover any source code or underlying algorithms of the Product or any part thereof provided solely in Object Code form but you may change, add or delete any files of the licensed copy of the Products and you may adapt or modify the Source Code solely for purposes of Operating a licensed copy of the Product and as expressly permitted pursuant to the type of the License purchased hereunder *provided* that you may not, in any event, remove or alter any copyright notices or other proprietary notices on any copies of the Product, whether so modified or not, *and further provided* that any such change, addition, deletion, adaptation or modification voids any express warranty provided herein and terminates any right to support services.

1.3. Confidential Information. You agree that, unless otherwise specifically provided herein the Product, including the specific design and structure of individual programs and the Product, constitute confidential proprietary information of the Licensor or its suppliers and/or licensors. You agree not to transfer, copy, disclose, provide or otherwise make available such confidential information in any form to any third party. For purposes hereof, ***"License Key"*** or ***"Registration Key"*** shall mean a file or a unique sequence of digit and/or symbols provided to you by the Licensor confirming the purchase of the license from the Licensor, which may carry the information about the License, i.e. its type, the user name and the number of licenses purchased, and enabling the full functionality of the Product in accordance with the License granted under this Agreement. You agree to implement reasonable security measures to protect such confidential information. If you download the Software from the Internet or similar on-line source, you must include the copyright notices resident on the Software with any on-line distribution and on any media you distribute that includes the Software.

2. **Grant of License.**

2.1. License. Unless otherwise specifically indicated under a valid Terms of Use, the Licensor grants you a non-exclusive and non-transferable license without the right to sublicense (the ***"License"***) and Licensee hereby accepts such License as follows, *provided* that unless otherwise agreed by Licensor or specified by Terms of Use, each License is granted per one Licensee:

a) Trial License. If you have received, downloaded and/or installed a Trial Version of the Product (the "Trial Version") and are hereby granted a Trial License for the Software and you may Operate the Product only for evaluation and demonstration purposes and only during the single applicable evaluation period of thirty (30) days (the ***"Trial Period"***), unless otherwise indicated, from the date of the initial installation. Any use of the Product for other purposes or beyond the applicable evaluation period is strictly prohibited, *provided however* that, subject to the restrictions contained herein, you may copy and distribute the Trial Version of the Software without any modifications whatsoever, and including this Agreement, to any third party. Licensee shall have no technical support rights during the Trial Period. The Licensor shall not be required to provide any support and Updates, as stated below, for the Trial Version of the Product. During the Trial Period, the Licensor provides no warranty whatsoever and assumes and bears no liability whatsoever for the Trial Version of the Product.

b) Personal Use License. If the Product is licensed under Personal License with Software provided in Object Code only upon the terms specified in the applicable registration, Terms of Use, invoicing or packaging for the Product, you personally may use of the Product solely for Personal Use (***"Personal License"***). ***"Personal Use"*** shall mean personal non-commercial, non-business, non-government Use, and not on behalf or for the benefit of any clients and excludes any commercial purposes whatsoever, which include without limitation: advertising marketing and promotional materials/services on behalf of an actual client, employer, employee or for your own benefit, any products that are commercially distributed, whether or not for a fee, any materials or services for sale or for which fees or charges are paid or received. Upon payment for the License and registration of the Product, you are granted a non-exclusive and non-transferable personal License to (i) install one (1) copy of the Product on up to three (3) Client Devices owned by you, and (ii) subject to the payment of the applicable fees and your compliance with the terms hereof, to Use one (1) copy of the specified version of the Product during the Term of this Agreement, on one (1) of such Client Devices at any given time. Additionally, the individual licensing terms may specify other terms, conditions and restrictions of Operating the Product.

c) Site License. If the Product is licensed with site license terms specified in the applicable product invoicing or packaging for the Product, you may Operate and Use the Product on an unlimited number of Client Devices within a single building owned or leased by your company. Additionally, the individual licensing terms may specify other terms, conditions and restrictions of Using the Product (the ***"Site License"***).

d) Business License. If the Product is licensed under Business License with the Software provided in Object Code only upon the terms specified in the applicable Terms of Use, invoicing or packaging for the Product, you may Use of the Product solely for Business Use (the ***"Business License"***). For purposes hereof, ***"Business Use"*** shall mean business, commercial, government Use only for internal business purposes of such entity without the right to distribute the applications, frameworks or components developed Using the Software (the ***"Results"***) to third parties *provided that* nothing herein shall be construed as creating any obligations of the Licensor to any end user of the Results. Under the Business License, any Use or Operating of the Product or transfer of the results of Using of the Product on a computer device owned by a third party is strictly prohibited unless a separate License is purchased therefor.

e) Commercial License. If the Product is licensed under Commercial License with the Software provided in Object Code only upon the terms specified in the applicable Terms of Use, invoicing or packaging for the Product, you may Use of the Product solely for Commercial Use (the ***"Commercial License"***). For purposes hereof, ***"Commercial Use"*** shall mean business, commercial, government Use with the right to distribute the Results to third parties. Licensee may Use the Product licensed under the Commercial License on an unlimited number of web servers and domains owned, rented or leased by Licensee. Under the Commercial License, any Use or Operating of the Product or transfer of the results of Using of the Product on a computer device owned by a third party is strictly prohibited unless a separate License is purchased therefor.

f) Intranet License. If the Product is licensed with Intranet License terms specified in the applicable product invoicing or packaging for the Product, you may Operate and Use the Product on an unlimited number of Client Devices within a single local area network and/or private computer network under your control (the ***"Intranet License***"). Additionally, the individual licensing terms may specify other terms, conditions and restrictions of Using the Product.

g) Internet License. If the Product is licensed with Internet License terms specified in the applicable product invoicing or packaging for the Product, you may Operate and Use the Product on an unlimited number of Client Devices on a single server accessible via internet (the ***"Internet License"***). Additionally, the individual licensing terms may specify other terms, conditions and restrictions of Using the Product.

h) Compiled Units. If you are granted a Commercial License pursuant to Section 2.1(e) hereof, in addition to the licenses and rights granted therein, Licensor grants you a nonexclusive, deployment-free, royalty-free right to reproduce and distribute the Object Code version of those portions of the Software which are identified in the Documentation as *'compiled units'* (the ***"Compiled Units"***) *provided* that you comply with all of the following requirements:

i). you distribute the Compiled Units in Object Code form only in conjunction with and as part of your software application product which adds significant and primary functionality and when the absence of Compiled Units will make your software application inoperable;

ii). you do not use Licensor name, logo or trademarks to market your software application product; and

iii). you include a valid copyright, trademark or any other proprietary notices on your Software identifying the Licensor as the owner of the Compiled Units.

i) Source Code License. The Licensor, pursuant to Terms of Use or invoicing terms, and in conjunction with one of Licenses granted under Section 2 hereof, may grant you certain rights to Software provided in Source Code as follows (the ***"Source Code License"***):

i). For purposes hereof, "Source Code" shall mean the human-readable form of the computer programming code and related system documentation including all comments and any procedural code such as job control language. Provided you have purchased a license to a part of the Software supplied in Source Code form, you may make modifications, enhancements, derivative works and/or extensions to that licensed Source Code provided to you under the terms set forth in this Section 2.1(f).

ii). While the Licensor does not claim any ownership rights in the Results, in the event you develop any modifications, enhancements, derivative works and/or extensions to the licensed Source Code (the ***"Derivatives"***), either independently or jointly with the Licensor, such Derivatives and all rights associated therewith will be the exclusive property of the Licensor.

iii). You shall not grant, either expressly or by implication, any rights, title, interest, or licenses to any Derivatives to any third party. You will, however, be entitled to use such Derivatives under the terms set forth in this Agreement. You hereby assign all right, title and interest in and to such Derivatives to the licensed Source Code to the Licensor.

iv). You also agree to execute, acknowledge and deliver to the Licensor all documents and instruments and do all things and actions Licensor deems necessary or desirable, at no cost to you but at Licensor's expense, to enable the Licensor to obtain and secure such Derivatives anywhere in the World. You agree to secure all necessary rights and obligations from relevant employees, or third parties in order to satisfy the above obligations. You may not distribute the Licensor's Source Code, or any Derivatives, in Source Code form.

v). Under no circumstances may any portion of the Source Code be distributed, disclosed or otherwise made available to any third party without the express, prior written consent of the Licensor. Under no circumstances may the Source Code be used in whole or in part, as the basis for creating a product that provides the same, or substantially the same, functionality as any of the Licensor's product. You will not take any action, or assist or otherwise aid anyone else in taking any action that would, in any way, limit the Licensor's independent development, sale, assignment, licensing or use of its Software or any

Derivatives thereof. You will not modify or delete, in whole or part, any copyright, trade secret, proprietary, confidential or other notice thereon or therein, including a prominent notice on the Results "***Powered by Microolap Database Designer for MySQL***" without the express, prior written consent of the Licensor.

vi). YOU UNDERSTAND, ACKNOWLEDGE AND AGREE THAT SOURCE CODE IS LICENSED "AS IS," AND THAT THE LICENSOR DOES NOT PROVIDE ANY TECHNICAL SUPPORT FOR SOURCE CODE.

j) Business License with Source Code License. If the Product is licensed under Business License with Source Code License upon the terms specified in the applicable Terms of Use, invoicing or packaging for the Product, your rights to Use the Product shall be the rights granted under the Business License together with the rights granted under the Source Code License (the ***"Business License with Source Code License"***).

k) Commercial License with Source Code License. If the Product is licensed under Commercial License with Source Code License upon the terms specified in the applicable Terms of Use, invoicing or packaging for the Product, your rights to may Use of the Product shall be the rights granted under the Commercial License together with the rights granted under the Source Code License (the ***"Commercial License with Source Code License"***).

l) Customized Licenses. The Licensor may grant you a specific customized terms of License pursuant to a valid Terms of Use signed by both parties hereto in which case such the terms and conditions of the Terms of Use shall be controlling and supersede any conflicting terms of this Agreement.

m) Educational Purpose License, Educational Institution Classroom License, and Educational Institution Site License. If the Product is licensed under an Educational Purpose License upon the terms specified in the applicable Educational Purpose License invoicing or packaging for the Product, you may make use of the Product solely for Educational Purpose. ***"Educational Purpose"*** means any non-commercial study or research that is undertaken solely in furtherance of one's education, whether or not completed by a student in pursuit of an educational degree, certificate or diploma and as used by teachers or facilitates teaching of a class, and all administrative staff, faculty and employees, of any college, university, trade school or other school (***"Educational Institution"***). With the acquisition of an Educational Institution Classroom License, Licensee may install and Operate the Product by a number of Users determined by the applicable invoicing terms within one Educational Institution in one classroom. Within these limitations, you may install the Product as a "Network" Product and run the Product from any networked Client Devices on Licensee's LAN, provided that such Client Devices are located exclusively within one classroom. With the acquisition of an Educational Institution Site License, Licensee may install and Operate the Product by a number of Users determined by the applicable invoicing terms within one Educational Institution in one geographic location. Within these limitations, you may install the Product as a "Network" Product and run the Product from any networked Client Devices on Licensee's LAN, provided that such Client Devices are located exclusively within one office complex within one geographic location. Educational License may be granted exclusively at the discretion of the Licensor upon your submission of a written request discussing your and your employer/ employees activities, when applicable, and your reasons for and purposes of Operating the Product.

2.2. Multiple Environment Product; Multiple Language Product; Dual Media Product; Multiple Copies; Bundles. If you use different versions of the Product or different language editions of the Product, if you receive the Product on multiple media, if you otherwise receive multiple copies of the Product, or if you received the Product bundled with other software, the total permitted number of your Client Devices on which all versions of the Product are installed shall correspond to the number of licenses you have obtained from the Licensor provided that unless the licensing terms and the License Key provides otherwise, each purchased license entitles you to install and Use the Product on one (1) Client Device. You may not rent, lease, sublicense, lend or transfer any versions or copies of the Product regardless of whether you use the Product or not, *provided that* the terms specified in the applicable product invoicing or packaging for the Product specify otherwise.

2.3. Run-time License. If the Product is licensed under Commercial License in accordance to Sections 2.1(e) or 2.1(k), in addition to the licenses and rights granted therein, Licensor grants Licensee a right to display, install, copy and distribute a portion of Licensor Software (the ***"Run-***

time Edition") on Client Devices of its customers as part of the Licensee's bundled products (the ***"Run-time License"***) in accordance with terms and conditions specified in the License Key, and/ or invoicing terms, including the limitations relating to the number Client Devices of such customers or the number of such customers as the case may be, *provided that*, Licensee shall include a valid and prominent copyright, trademark or any other proprietary notices in its bundled products identifying the Licensor as the owner of the Run-time Edition as may be specified by Licensor from time to time.

2.4. Updates. During the Term of this Agreement, you may download Updates to the Product when and as the Licensor publishes them in its website or through other online services. Notwithstanding any provision to the contrary herein, nothing in this Agreement shall be construed as to grant you any rights or licenses with regard to the New Releases of the Product or to entitle you to any New Release. This Agreement does not obligate the Licensor to provide any Updates. Notwithstanding the foregoing, any Updates that you may receive become part of the Product and the terms of this Agreement apply to them (unless this Agreement is superceded by a further Agreement accompanying such Update or modified version of to the Product).

2.5. Material Terms and Conditions. You specifically agree that each of the terms and conditions of this Section 2 are material and that failure of you to comply with these terms and conditions shall constitute sufficient cause for Licensor to immediately terminate this Agreement and the License granted under this Agreement. The presence of this Section 2.5 shall not be relevant in determining the materiality of any other provision or breach of this Agreement by either party hereto.

3. Additional Covenants; Assignment of Intellectual Property Rights.

3.1. Additional Limitations. Notwithstanding anything to the contrary herein, you may not Operate, Use, or modify the Product in any way as to form the basis for creating a product that provides the same, or substantially the same, functionality as the Product; and in the event you develop any modifications, enhancements, derivative works and/or extensions to the Product, either independently or jointly with Licensor, such modifications, enhancements, derivative works and/or extensions and all rights associated therewith will be the exclusive property of Licensor. You will not grant, either expressly or impliedly, any rights, title, interest, or licenses to any such modifications, enhancements, derivative works and/or extensions to any third party. You will, however, be entitled to use such modifications, enhancements, derivative works and/or extensions under the terms set forth in this Agreement. You hereby assign all right, title and interest in and to such modifications, enhancements, derivative works and/or extensions to the Product to Licensor. You also agree to execute, acknowledge and deliver to Licensor all documents and do all things Licensor deems necessary or desirable, at no cost to but at Licensor expense, to enable Licensor to obtain and secure such modifications, enhancements, derivative works and/or extensions anywhere in the world. You agree to secure all necessary rights and obligations from relevant employees or third parties in order to satisfy the above obligations.

3.2. Reservations of all Rights. The Licensor reserves all rights not expressly granted herein.

3.3. Back-up Copies. You can make one (1) copy of the Product for backup and archival purposes, *provided, however*, that the original and each copy is kept in your possession or control, and that your installation, Operation and Use of the Product does not exceed that which is allowed in Section 2 hereof.

3.4. Additional Protection Measures. Solely for the purpose of preventing unlicensed and/or unauthorized Use of the Product, including without limitation Run-time Edition, the Software may collect certain non-personal information relating to the hardware of your or your customer's Client Devices and/or install on your or your customer's Client Devices certain technological measures that are designed to prevent unlicensed and/or unauthorized Use and Operation of the Product, and the Licensor may use this technology to confirm that you have a licensed copy of the Product. Such installation or collection of information or updates of these technological measures may occur through and/or during the installation or activation of the Product or Updates. The Product and/or Updates will not install or may fail to Operate if installed contrary to the rights granted under the License or if attempted to be installed or Operated on unlicensed copies of the Product. If you are not using a licensed copy of the Product, you are not allowed to install the Updates. The Licensor will not collect any personally identifiable information from your computer during this process.

4. Term and Termination.

4.1. Term. The term of this Agreement (***"Term"***) shall begin when you download, access or install the Product, whichever is earlier, and shall continue in perpetuity unless otherwise designated in the purchase order, Terms of Use, exhibit or unless otherwise terminated pursuant hereto. Without prejudice to any other rights, this Agreement will terminate automatically if you fail to comply with any of the limitations or other requirements described herein. Upon any termination or expiration of this Agreement, you must immediately cease Operating the Product and all of its components and destroy, uninstall and erase all copies of the Product and all of its components, including without limitation on all systems and all types of media and in computer memory.

4.2. No Rights upon Termination. Upon termination of this Agreement you will no longer be authorized to Operate or Use the Product in any way.

5. Support and Updates.

5.1. Terms of Support. During the Warranty Period as defined below, you are entitled to technical services and support for the Product which is provided to you by Licensor free of charge during the regular business hours (GMT+3), except for locally-observed holidays, and includes the support provided through a special technical support section of the Licensor Site (http://www.microolap.com/support/) (***"Warranty Support"***). During the Warranty Period, Warranty Support is unlimited and includes technical and support questions and patch fixes. After the expiration of the Warranty Period, you may purchase additional support services from Licensor at current rates as listed at Licensor Site (http://www.microolap.com/support/).

5.2. Updates. During the Warranty Period hereunder and, if your purchasing terms provide for a certain extended period of time during which you are entitled to Updates free of charge (i.e., one year from the time of the purchase of the License), then for such extended period of time, you may download Updates to the Product when and as the Licensor publishes them on the Licensor Site, or through other online services. If the Product is an Update to a previous version of the Product, you must possess a valid license to such previous version in order to use the Update. You may continue to use the previous version of the Product on your Client Device after you receive the Update to assist you in the transition to the Update, provided that: (i) the Update and the previous version are installed on the same computer device; (ii) the previous version or copies thereof are not transferred to another party unless all copies of the Update are also transferred to such party; and (iii) you acknowledge that any obligation the Licensor may have to support the previous version of the Product may be ended upon availability of the Update. Except for the rights to free Updates during the Warranty Period, as further defined herein, nothing in this Agreement shall be construed as to grant you any rights or licenses with regard to the new version or releases of the Product or to entitle you to any new version, upgrade or release. This Agreement does not obligate the Licensor to provide any Updates. Notwithstanding the foregoing, any Updates that you may receive become part of the Product and the terms of this Agreement apply to them (unless this Agreement is superseded by a succeeding agreement accompanying such Update or modified version of the Product). Notwithstanding anything to the contrary herein, nothing herein shall be deemed to entitle you to any Licensor's Optional Updates that are provided by Licensor for a fee, or to any new releases, versions or substitutes of the Product.

5.3. Additional Support and Updates. In addition to the free Support and free Updates provided for in Sections 5.1 and 5.2, you may purchase additional Services, additional Updates or additional Support beyond the applicable period at the ongoing rates and prices published or provided by Licensor.

6. Restrictions.

6.1. No Transfer of Rights. Except as otherwise specifically provided herein or in the applicable Terms of Use, you may not transfer any rights pursuant to this Agreement nor rent, sublicense, lease, loan or resell the Product or permanently or temporarily transfer the Product in any other manner. You may not permit third parties, including any subcontractors, to benefit from the use or functionality of the Product including, without limitations, via a timesharing, service bureau or other arrangement, except to the extent such use is specified in the application price list, purchase order or product packaging for the Product. Except as otherwise provided in Section 1.2 hereof, you may not, without the Licensor's prior written consent, reverse engineer, decompile, disassemble or otherwise reduce any part of the Product to human readable form nor permit any third party to do so, except to the extent the foregoing restriction is expressly prohibited by applicable law. Notwithstanding the foregoing sentence, decompiling the Software is permitted to the extent the laws of your jurisdiction expressly give you the non-waivable right to do so to obtain information necessary to render the Software interoperable with other software; provided, however, that you must first request such information from the Licensor and the Licensor may, in its discretion, either provide such information to you (subject to confidentiality terms) or impose reasonable conditions, including a reasonable fee, on such use of the Software to ensure that the Licensor's and its affiliates' proprietary rights in the Software are protected. Except for the modification permitted under Section 1.2, you may not modify, or create derivative works based upon the Product in whole or in part.

6.2. No Extraction for Separate Use. You shall not have the right to extract or to Use any functionality of this Software, including any DLLs or other compiled units or Object Code fragments other than as part of normal Operation of the Product described in the Documentation and as integral part of Operation and functionality of the Product as a whole.

6.3. Proprietary Notices and Copies. You may not remove any proprietary notices or labels on the Product. You may not copy the Product except as expressly permitted in Section 2 above.

6.4. Compliance with Law. You agree that in Operating the Product and in using any report or information derived as a result of Operating this Product, you will comply with all applicable international, national, state, regional and local laws and regulations, including, without limitation, privacy, trademark, patent, copyright, export control and obscenity law and you shall not use the Product for unethical or illegal business practices or in violation of any obligation to a third party in using, Operating, accessing or running any of the Product and shall not knowingly assist any other person or entity to so violate any obligation to a third party.

7. WARRANTIES AND DISCLAIMERS.

7.1. Limited Warranty. The Licensor warrants that for two (2) months (the ***"Warranty Period"***) from the date the Product has been downloaded by you or was made otherwise available to you by Licensor if the Product was supplied to you on other media, the media on which Product has been provided will be free from defects in materials and workmanship and that the Software will perform substantially in accordance with the Documentation or generally conform to the Product's specifications published by the Licensor. Non-substantial variations of performance from the Documentation do not establish a warranty right. THIS LIMITED WARRANTY DOES NOT APPLY TO UPDATES AS APPLIED TO ANY MODIFIED PRODUCT, WHETHER OR NOT SUCH MODIFICATION IS PERMISSIBLE HEREUNDER, TRIAL AND DEMO OR EVALUATION VERSIONS, UPDATES, PRE-RELEASE, TRYOUT, PRODUCT SAMPLER, OR NOT FOR RESALE (NFR) COPIES OF PRODUCT. This limited warranty is void and your support right terminates if the defect or damage has resulted from accident, abuse, or misapplication or any modification, whether or not such modification is permitted hereunder. Notwithstanding anything to the contrary herein, any and all modifications shall be deemed derivative works within the meaning of copyright law and remain subject to the terms of this Agreement, including without limitation Section 1.1 hereof, *provided however*, that any damage or claims relating to or arising out of any modifications made by Licensee, whether or not permitted hereunder, shall be the sole responsibility of Licensee. To make a warranty claim, you must return the Product to the location where you obtained it along with proof of purchase within such sixty (60) day period of the license fee you paid for the Product. THE LIMITED WARRANTY SET FORTH IN THIS SECTION GIVES YOU SPECIFIC LEGAL RIGHTS. YOU MAY HAVE ADDITIONAL RIGHTS WHICH VARY FROM JURISDICTION TO JURISDICTION.

7.2. Customer Remedies. The Licensor and its suppliers' entire liability and your exclusive remedy for any breach of the foregoing warranty shall be at the Licensor's option: (i) return of the purchase price paid for the License, if any, (ii) replacement of the defective media in which the Product is contained, or (iii) correction of the defects, "bugs" or errors within reasonable period of time. You must return the defective media to the place of purchase at your expense with a copy of your receipt. Any replacement media will be warranted for the remainder of the original Warranty Period.

7.3. NO OTHER WARRANTIES. EXCEPT FOR THE FOREGOING LIMITED WARRANTY, AND FOR ANY WARRANTY, CONDITION, REPRESENTATION OR TERM TO THE EXTENT TO WHICH THE SAME CANNOT OR MAY NOT BE EXCLUDED OR LIMITED BY LAW APPLICABLE TO YOU IN YOUR JURISDICTION, THE PRODUCT IS PROVIDED "AS-IS" WITHOUT ANY WARRANTY WHATSOEVER AND THE LICENSOR MAKES NO PROMISES, REPRESENTATIONS OR WARRANTIES, WHETHER EXPRESSED OR IMPLIED, WHETHER BY STATUTE, COMMON LAW, CUSTOM, USAGE OR OTHERWISE, REGARDING OR RELATING

TO THE PRODUCT OR CONTENT THEREIN OR TO ANY OTHER MATERIAL FURNISHED OR PROVIDED TO YOU PURSUANT TO THIS AGREEMENT OR OTHERWISE. YOU ASSUME ALL RISKS AND RESPONSIBILITIES FOR SELECTION OF THE PRODUCT TO ACHIEVE YOUR INTENDED RESULTS, AND FOR THE INSTALLATION OF, USE OF, AND RESULTS OBTAINED FROM THE PRODUCT. THE LICENSOR MAKES NO WARRANTY THAT THE PRODUCT WILL BE ERROR FREE OR FREE FROM INTERRUPTION OR FAILURE, OR THAT IT IS COMPATIBLE WITH ANY PARTICULAR HARDWARE OR SOFTWARE. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, LICENSOR DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT OF THIRD PARTY RIGHTS, INTEGRATION, SATISFACTORY QUALITY OR FITNESS FOR ANY PARTICULAR PURPOSE WITH RESPECT TO THE PRODUCT AND THE ACCOMPANYING WRITTEN MATERIALS OR THE USE THEREOF. SOME JURISDICTIONS DO NOT ALLOW LIMITATIONS ON IMPLIED WARRANTIES, SO THE ABOVE LIMITATION MAY NOT APPLY TO YOU. YOU HEREBY ACKNOWLEDGE THAT THE PRODUCT MAY NOT BE OR BECOME AVAILABLE DUE TO ANY NUMBER OF FACTORS INCLUDING WITHOUT LIMITATION PERIODIC SYSTEM MAINTENANCE, SCHEDULED OR UNSCHEDULED, ACTS OF GOD, TECHNICAL FAILURE OF THE SOFTWARE, TELECOMMUNICATIONS INFRASTRUCTURE, OR DELAY OR DISRUPTION ATTRIBUTABLE TO VIRUSES, DENIAL OF SERVICE ATTACKS, INCREASED OR FLUCTUATING DEMAND, AND ACTIONS AND OMISSIONS OF THIRD PARTIES. THEREFORE, THE LICENSOR EXPRESSLY DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY REGARDING SYSTEM AND/OR SOFTWARE AVAILABILITY, ACCESSIBILITY, OR PERFORMANCE. THE LICENSOR DISCLAIMS ANY AND ALL LIABILITY FOR THE LOSS OF DATA DURING ANY COMMUNICATIONS AND ANY LIABILITY ARISING FROM OR RELATED TO ANY FAILURE BY THE LICENSOR TO TRANSMIT ACCURATE OR COMPLETE INFORMATION TO YOU.

7.4. LIMITED LIABILITY; NO LIABILITY FOR CONSEQUENTIAL DAMAGES. YOU ASSUME THE ENTIRE COST OF ANY DAMAGE RESULTING FROM YOUR USE OF THE PRODUCT AND THE INFORMATION CONTAINED IN OR COMPILED BY THE PRODUCT, AND THE INTERACTION (OR FAILURE TO INTERACT PROPERLY) WITH ANY OTHER HARDWARE OR SOFTWARE WHETHER PROVIDED BY THE LICENSOR OR A THIRD PARTY. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT WILL THE LICENSOR OR ITS SUPPLIERS OR LICENSORS BE LIABLE FOR ANY DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL OR INDIRECT DAMAGES, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, LOSS OF DATA, LOSS OF GOODWILL, WORK STOPPAGE, HARDWARE OR SOFTWARE DISRUPTION IMPAIRMENT OR FAILURE, REPAIR COSTS, TIME VALUE OR OTHER PECUNIARY LOSS) ARISING OUT OF THE USE OR INABILITY TO USE THE PRODUCT, OR THE INCOMPATIBILITY OF THE PRODUCT WITH ANY HARDWARE SOFTWARE OR USAGE, EVEN IF SUCH PARTIES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN NO EVENT WILL LICENSOR'S TOTAL LIABILITY TO YOU FOR ALL DAMAGES IN ANY ONE OR MORE CAUSE OF ACTION, WHETHER IN CONTRACT, TORT OR OTHERWISE EXCEED THE AMOUNT PAID BY YOU FOR THE PRODUCT. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY TO THE EXTENT THAT APPLICABLE LAW PROHIBITS SUCH LIMITATION. FURTHERMORE, BECAUSE SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES, THE ABOVE LIMITATION MAY NOT APPLY TO YOU.

8. Indemnification

8.1. Indemnification for Violations. In Operating the Product, you agree to use only those materials for which you have the necessary patent, copyright and other permissions, licenses, and/or clearances. You agree to indemnify, defend and hold harmless the Licensor and its respective officers, directors, employees, agents, successors, and assigns (the ***"Licensor Indemnitees"***) from any and all losses, liabilities, damages and claims, and all related expenses including without limitation reasonable legal fees and disbursements and costs of investigation, litigation, settlement, judgment, interest and penalties and costs related to, arising from, or in connection with any third-party claim related to, arising from, or in connection with the actual or alleged: (i) infringement by you or by Compiled Units (except when such breach is exclusively attributable to Product) of any third-party intellectual property and/or proprietary right, including, but not limited to, patent, trademark, copyright, trade secret, publicity and/or privacy, (ii) personal injury (including death) or property damage due to gross negligence or intentional misconduct of the Licensee, and (iii) breach by the Licensee of any of its representations, warranties, obligations, and/or covenants set forth herein. You shall promptly notify the Licensor in writing after you become aware of any such claims, but failure to give such notice shall not relieve you of indemnity obligations hereunder. You shall have exclusive control over the settlement or defense of such claims or actions, except that Licensor may appear in the action, at its own expense, through counsel reasonably acceptable to you, only in the event it is mutually determined by the parties that an actual conflict of interest would exist by your representation of the Licensor and you in such action. Licensor shall give you, at your expense, all information and assistance reasonably requested by you to settle or defend such claims or actions. You shall be entitled to retain all monetary proceeds, attorneys' fees, costs and other rewards you receive as a result of defending or settling such claims. In the event you fail to promptly indemnify and defend such claims and/or pay Licensor's expenses, as provided above, Licensor shall have the right to defend itself, and in that case, you shall reimburse the Licensor Indemnitees for all of their attorneys' fees, costs and damages incurred in settling or defending such claims within thirty (30) days of each of Licensor's written requests. Nothing in this Section 3.2 or this Agreement shall be interpreted as to exclude any possible legal recourse against the Licensee.

9. U.S. Government-Restricted Rights.

9.1. Notice to U.S. Government End Users. The Product and accompanying Documentation are deemed to be "Commercial Items," as that term is defined at 48 C.F.R. §2.101, consisting of "Commercial Computer Software" and "Commercial Computer Software Documentation," respectively, as such terms are used in 48 C.F.R. §12.212 or 48 C.F.R. §227.7202, as applicable. Consistent with 48 C.F.R. §12.212 or 48 C.F.R. §§227.7202-1 through 227.7202-4, as applicable, the Commercial Computer Software and Commercial Computer Software Documentation are being licensed to U.S. Government end users (a) only as Commercial Items and (b) with only those rights, including any use, modification, reproduction, release, performance, display or disclosure of the Product and accompanying Documentation, as are granted to all other end users pursuant to the terms and conditions herein. Unpublished rights are reserved under the copyright laws of the United States.

9.2. Export Restrictions. You acknowledge and agree that the Product may be subject to restrictions and controls imposed by the Export Administration Act and the Export Administration Regulations of the United States (the ***"Acts"***). You agree and certify that neither the Product nor any direct product thereof is being or will be used for any purpose prohibited by the Acts. You may not Operate, download, export, or re-export the Product (a) into, or to a national or resident of, any country to which the United States has embargoed goods, or (b) to anyone on the United States Treasury Department's list of Specially Designated Nationals or the U.S. Commerce Department's Table of Deny Orders. By downloading or using the Product, you are representing and warranting that you are not located in, under the control of, or a national or resident of any such country or on any such list. You acknowledge that it is your sole responsibility to comply with any and all government export and other applicable laws and that the Licensor has no further responsibility for such after the initial license to you. You warrant and represent that neither the U.S. Commerce Department, Bureau of Export Administration nor any other U.S. federal agency has suspended, revoked or denied your export privileges. For more information on the U.S. Export Administration Regulations (EAR), 15 C.F.R. Parts 730-774, and the Bureau of Export Administration (***"BXA"***), please see the BXA homepage (http://www.bxa.doc.gov).

10. Your Information and the Licensor's Privacy Policy

10.1. Privacy Policy. You acknowledge receipt of and agree to the Licensor's privacy statement which is made available to you in connection with installation and is set forth in full at http://www.microolap/about/privacy/. You hereby expressly consent to the Licensor's processing of your personal data (which may be collected by the Licensor or its distributors) according to the Licensor's current privacy policy as of the date of the effectiveness hereof which is incorporated into this Agreement by reference. By entering into this Agreement, you agree that the Licensor may collect and retain information about you, including your name, email address and credit card information. The Licensor may employ other companies and individuals to perform certain functions on its behalf. Examples include fulfilling orders, delivering packages, sending postal mail and e-mail, removing repetitive information from customer lists, analyzing data, providing marketing assistance, processing credit card payments, implementing fraud check policies, and providing customer service. Such companies and individuals may have access to personal information needed to perform their functions, but may not use it for other purposes. The Licensor publishes a privacy policy on the Licensor Site and may amend such policy from time to time in its sole discretion. You should refer to the Licensor's privacy policy prior to agreeing to this Agreement for a more detailed explanation of how your information will be stored and used by the Licensor. If "you" are an organization, you will ensure that each member of your organization (including employees and contractors) about whom personal data may be provided to the Licensor has given his or her express consent to the Licensor's processing of such personal data. Personal data will be processed by the Licensor or its distributors in the country where it was collected. The relevant laws in such jurisdictions regarding processing of personal data may be less or more stringent than the laws in your jurisdiction.

10.2. Public Announcements. The Licensor may identify you to the public as a customer of the Licensor and describe in a customer case study the services and solutions delivered by the Licensor to you. The Licensor may also issue one or more press releases, containing an announcement of the execution and delivery of this Agreement and/or the implementation of the Product by you. Nothing contained in this Section 10.2 shall be construed as an obligation by you to disclose any of your proprietary or confidential information to any third party. In addition, you may opt-out from this Section 10.2 by writing an opt-out request to the Licensor at sales@microolap.com.

11. Miscellaneous.

11.1. Governing Law; Jurisdiction and Venue. This Agreement shall be governed by and construed and enforced in accordance with the laws of the British Virgin Islands without reference to conflicts of law rules and principles. This Agreement shall not be governed by the United Nations Convention on Contracts for the International Sale of Goods, the application of which is expressly disclaimed and excluded. The courts within the British Virgin Islands shall have exclusive jurisdiction to adjudicate any dispute arising out of this Agreement. You agree that this Agreement and any action, dispute, controversy, or claim that may be instituted based on this Agreement, or arising out of or related to this Agreement or any alleged breach thereof, shall be prosecuted exclusively in the courts of the British Virgin Islands and you, to the extent permitted by applicable law, hereby waive the right to change venue to any other state, county, district or jurisdiction; *provided, however*, that the Licensor as claimant shall be entitled to initiate proceedings in any court of competent jurisdiction.

11.2. Period for Bringing Actions. No action, regardless of form, arising out of the transactions under this Agreement, may be brought by either party hereto more than one (1) year after the cause of action has occurred, or was discovered to have occurred, except that an action for infringement of intellectual property rights may be brought within the maximum applicable statutory period.

11.3. Entire Agreement; Severability; No Waiver. This Agreement is the entire agreement between you and Licensor and supersedes any other prior agreements, proposals, communications or advertising, oral or written, with respect to the Product or to subject matter of this Agreement. You acknowledge that you have read this Agreement, understand it and agree to be bound by its terms. If any provision of this Agreement is found by a court of competent jurisdiction to be invalid, void, or unenforceable for any reason, in whole or in part, such provision will be more narrowly construed so that it becomes legal and enforceable, and the entire Agreement will not fail on account thereof and the balance of the Agreement will continue in full force and effect to the maximum extent permitted by law or equity while preserving, to the fullest extent possible, its original intent. No waiver of any provision or condition herein shall be valid unless in writing and signed by you and an authorized representative of Licensor provided that no waiver of any breach of any provisions of this Agreement will constitute a waiver of any prior, concurrent or subsequent breach. Licensor's failure to insist upon or enforce strict performance of any provision of this Agreement or any right shall not be construed as a waiver of any such provision or right.

11.4. Contact Information. Should you have any questions concerning this Agreement, or if you desire to contact the Licensor for any reason, please contact our Customer Department at http://microolap.com/support/.