



# Визуальное создание баз данных с помощью MySQL Workbench

Author: [Pablo Pastor](#) on Apr 14th 2010

Translated by: [Jevgeni Agapitov](#) 20th March 2011

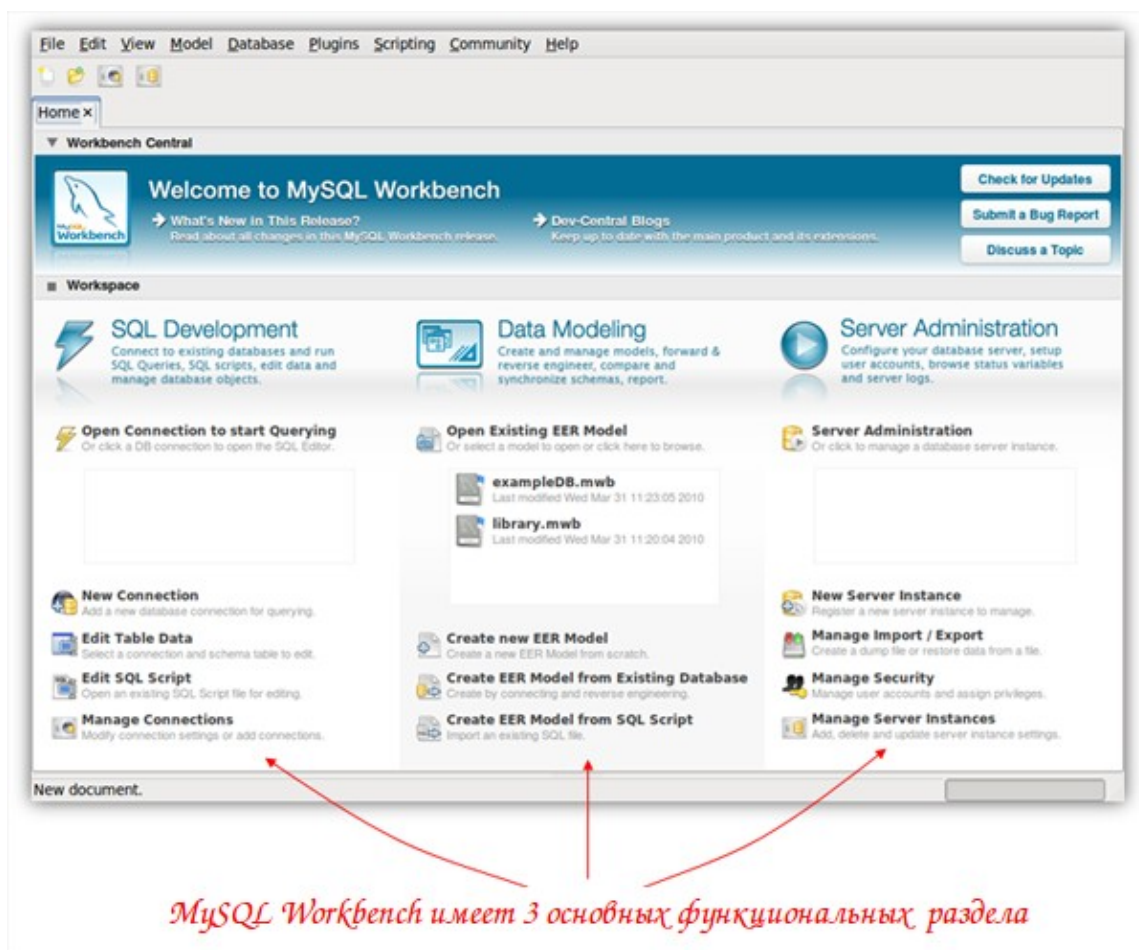
## Содержание

Визуальное создание баз данных с помощью MySQL Workbench.....	1
Что такое MySQL Workbench?.....	2
Планирование базы данных.....	3
Отправляемся в MySQL Workbench.....	4
Уточнение концепции.....	5
Создание таблиц.....	6
Переход в визуальный режим.....	7
Рисование отношений.....	8
Создание кода SQL.....	11
Подключение к серверу MySQL.....	13
Заключение.....	14

# Что такое MySQL Workbench?

MySQL Workbench является мощным инструментом, разработанным для решения трех основных задач:

- **Разработка на языке SQL:** Позволяет пользователю подключаться к существующей базе данных для редактирования и выполнения SQL запросов.
- **Моделирование данных:** полное визуальное проектирование и моделирование баз данных.
- **Администрирование баз данных:** Заменяет программу администрирования `mysql`. Имеющийся графический интерфейс служит для запуска / остановки серверов, создания учетных записей, редактирования конфигурационных файлов и т.д.

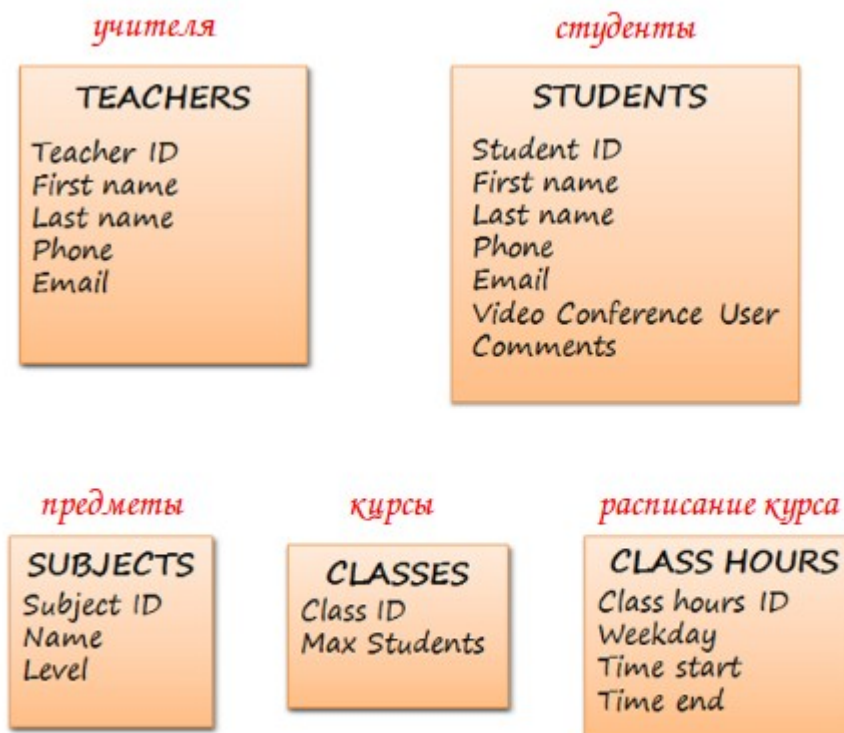


В этом учебном пособии мы сосредоточимся на аспекте *моделирования данных*, создадим базу данных с нуля, а затем только бегло познакомимся с редактором выражений SQL, чтобы выполнить наш сценарий SQL и создать базу данных на сервере MySQL.

Программа MySQL Workbench доступна для Windows, Linux и Mac OSX. Существуют два варианта программы: свободный *Community OSS Edition* и коммерческий *Standard Edition*. Community Edition, как и следовало ожидать, поставляется с открытым исходным кодом и по лицензии GPL. Это полностью функциональная программа, и именно ее мы будем использовать в этом учебном пособии. В коммерческом варианте добавлены некоторые дополнительные функции, такие как проверка схемы и модели и автоматизированное создание документации.

## Планирование базы данных

Для изучения MySQL Workbench мы будем использовать в качестве примера очень простую базу данных для дистанционного обучения. Пусть группа учителей хочет предложить дистанционное обучение на различные темы, используя Skype или любое другое программное обеспечение для видеоконференций. Будем считать, что в нашем небольшом проекте мы должны хранить следующую информацию:



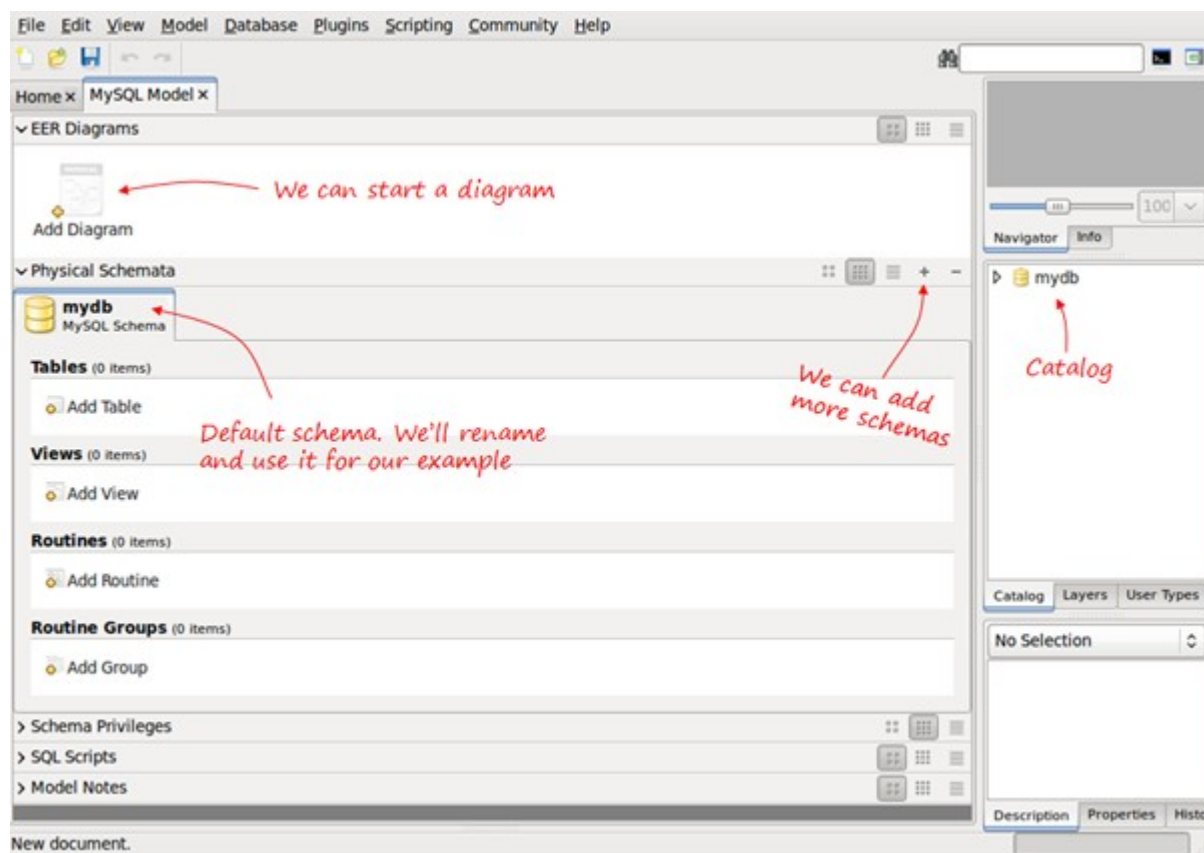
При разработке схемы мы должны знать отношения между этими группами данных, так что лучше подумать об этом сейчас!

- Один учитель может преподавать многие предметы
- Один предмет может преподаваться многими учителями
- Каждого курс преподает только один учитель
- Один учитель может преподавать много курсов
- Один студент может изучать много курсов
- Один курс изучает много студентов
- Один курс может иметь несколько часов (в неделю)
- В какой-то конкретный день и час может быть несколько курсов
- В одном курсе изучается один предмет
- Один предмет может изучаться во многих к курсах

Теперь у нас есть все данные, необходимые для решения задачи...

# Отправляемся в MySQL Workbench

Пришло время запустить MySQL Workbench. В разделе моделирования данных на главном экране нажмите на ссылке 'Create new EER Model' (Создать новую EER модель), и появится следующее окно:



Когда мы создаем новую модель базы данных, она содержит схему по умолчанию MyDB. Мы можем переименовать ее и использовать в качестве нашей схемы БД. Модель базы данных может иметь несколько различных схем.

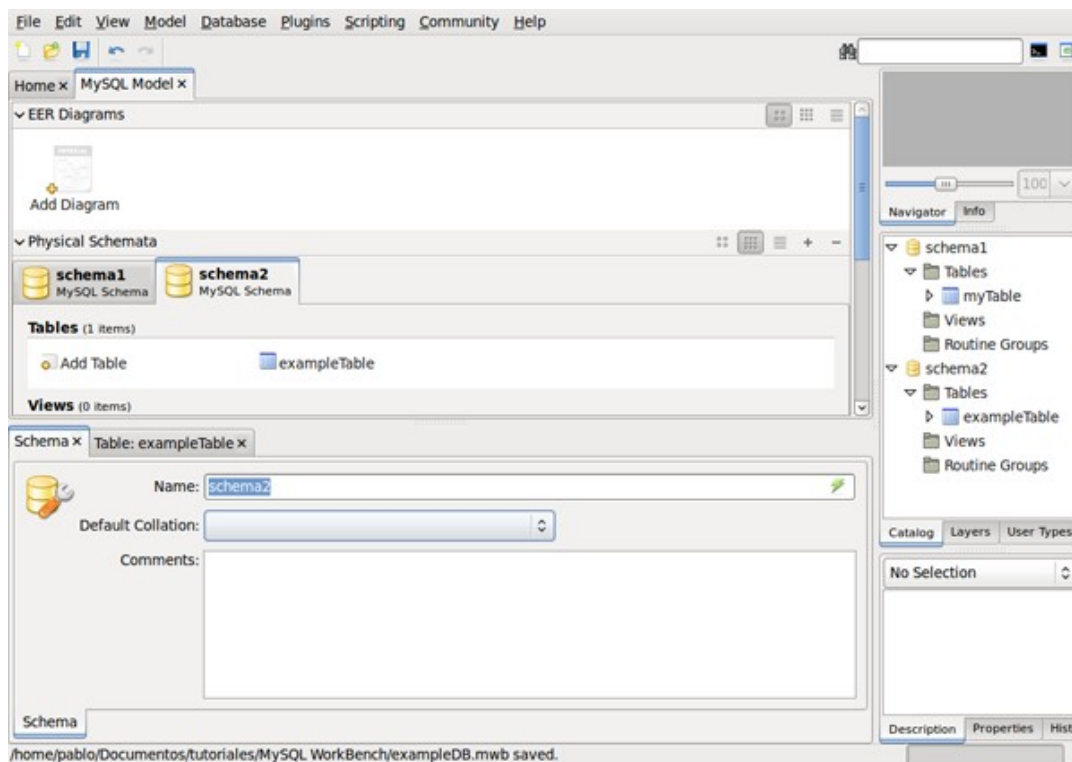
Каталог справа показывает все элементы нашей схемы и позволяет в случае необходимости перетаскивать элементы на диаграмму.

Наличие отдельных разделов для физической схемы и диаграммы EER и возможность включать несколько схем в одну модель базы данных может привести к путанице. В следующем разделе будут объяснены эти понятия и связь между ними.

## Уточнение концепции

Физическая схема содержит все необходимое для определения базы данных: таблицы, столбцы, типы, индексы, ограничения и т.д. Это то, что мы действительно определяем. Каждый объект, добавленный в графическую модель, также показывается в физической схеме. Это, по сути, визуальный способ определить наши схемы.

Мы можем иметь несколько схем для одной и той же модели базы данных, так же мы можем иметь несколько баз данных на сервере MySQL. Каждая схема будет базой данных MySQL. Например, на следующем экране у нас есть две закладки схем:



Если мы сгенерируем сценарий SQL для создания базы данных, у нас будет два отдельных выражения CREATE DATABASE - на самом деле в сценарии будут выражения CREATE SCHEMA, которые являются просто синонимами.

```
CREATE SCHEMA IF NOT EXISTS `schema1`;  
CREATE SCHEMA IF NOT EXISTS `schema2`;
```

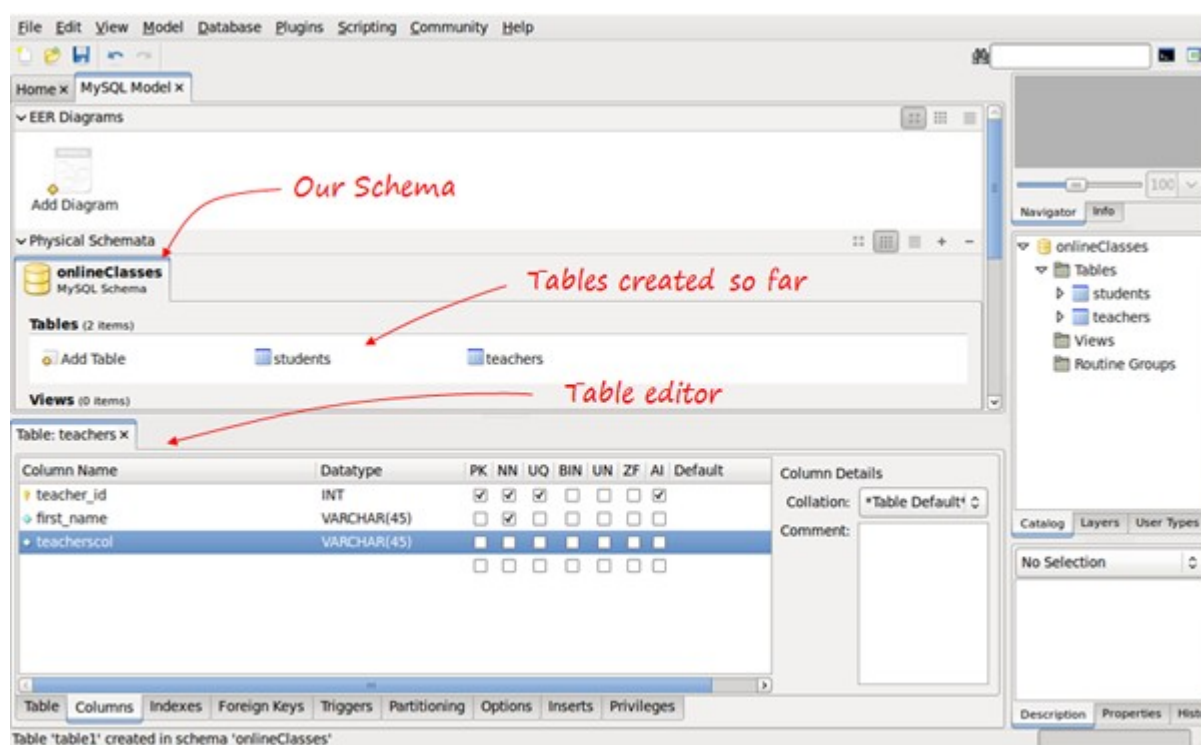
Эти две схемы будут показаны как базы данных на сервере MySQL, если использовать выражение SHOW DATABASES.

Теперь, что такое EER диаграммы? EER обозначает *Extended (or Enhanced) Entity-Relationship* (расширенная диаграмма сущность-связь). EER диаграммы просто способ моделирования данных и отношений между данными с использованием стандартных символов. EER модели могут быть достаточно сложными, но MySQL Workbench использует только подмножество всех возможных графических элементов, потому что целью такой диаграммы (в этой программе) является, чтобы каждый элемент отображался в физической схеме.

Мы можем использовать EER диаграммы для определения всей базы данных или только ее частей. Например, у нас имеется схема с пятью таблицами, затем мы можем создать новую диаграмму, чтобы определить с помощью визуального редактора еще две таблицы. Схема будет содержать только две таблицы, но эти две таблицы также будут включены в схему вместе с пятью предыдущими.

## Создание таблиц

Вернемся к нашему примеру. Мы должны переименовать схему по умолчанию, дважды щелкнув имя **mydb**. Теперь у нас есть две возможности продолжить работу: мы можем начать добавлять таблицы в физическую схему (на вкладке *Model*) с помощью значка *Add Table* или мы можем открыть вкладку *EER Diagram* и добавить все таблицы на ней. Я предпочитаю с самого начала добавить новую диаграмму и визуально создавать на ней таблицы, однако для того, чтобы показать, как сделать это обоими методами, мы создадим первые две таблицы на вкладке *Model*, а затем продолжим на вкладке *EER Diagram*. При нажатии на кнопку *Add Table* (Добавить таблицу), редактор таблиц откроется в виде вкладки как показано ниже:



Используя редактор таблиц, изменим имя таблицы и перейдем на вкладку *Columns* (на вкладках внизу редактора) для доступа к столбцам. Мы можем выбрать тип данных (есть выпадающий список всех типов данных MySQL), присвоить, если это необходимо, значение по умолчанию, и у нас есть семь флажков, чтобы отметить какое-либо из следующих свойств:

- PK – Primary key (первичный ключ)
- NN – Not null (не null, т.е. значение должно быть определено)
- UQ – Unique (уникальное значение)
- BIN – Binary (двоичное значение)
- UN – Unsigned (беззнаковое значение)
- ZF – Zero fill (заполнено нулями)
- AI – Autoincrement (автоинкремент — автоматическое увеличение на 1)

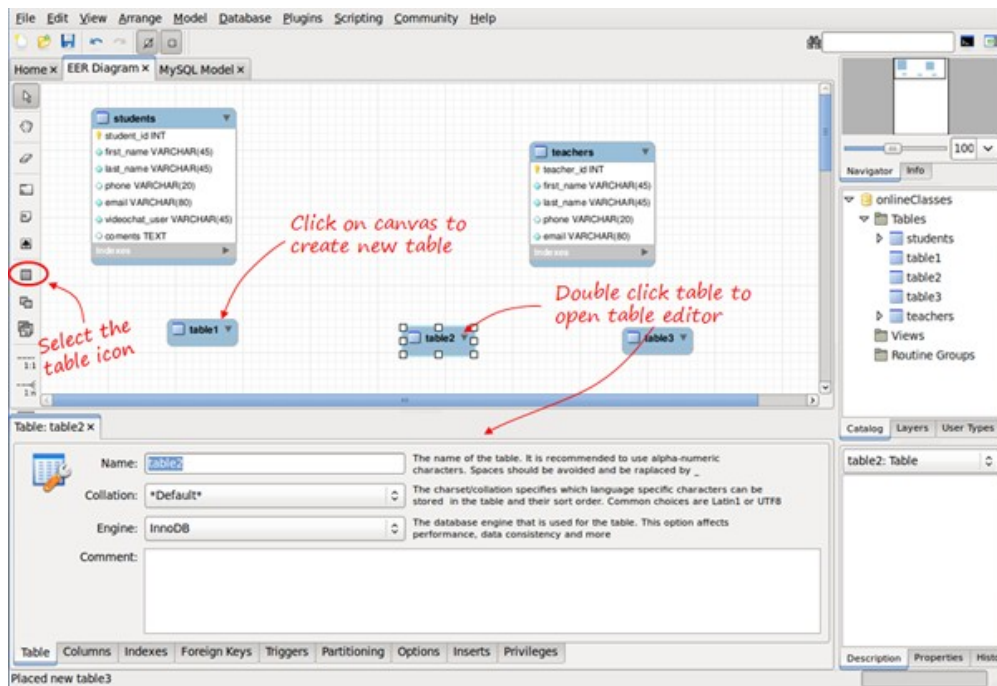


## Переход в визуальный режим

Это еще один из способов добавления таблиц, хотя мы также можем создавать их с помощью диаграмм. Если сейчас мы нажмем на значок *Add Diagram* (Добавить диаграмму), откроется новая, пустая вкладка диаграмм, и это не то, что мы хотим. Мы хотим, чтобы две таблицы, которые мы только что создали, были на диаграмме.

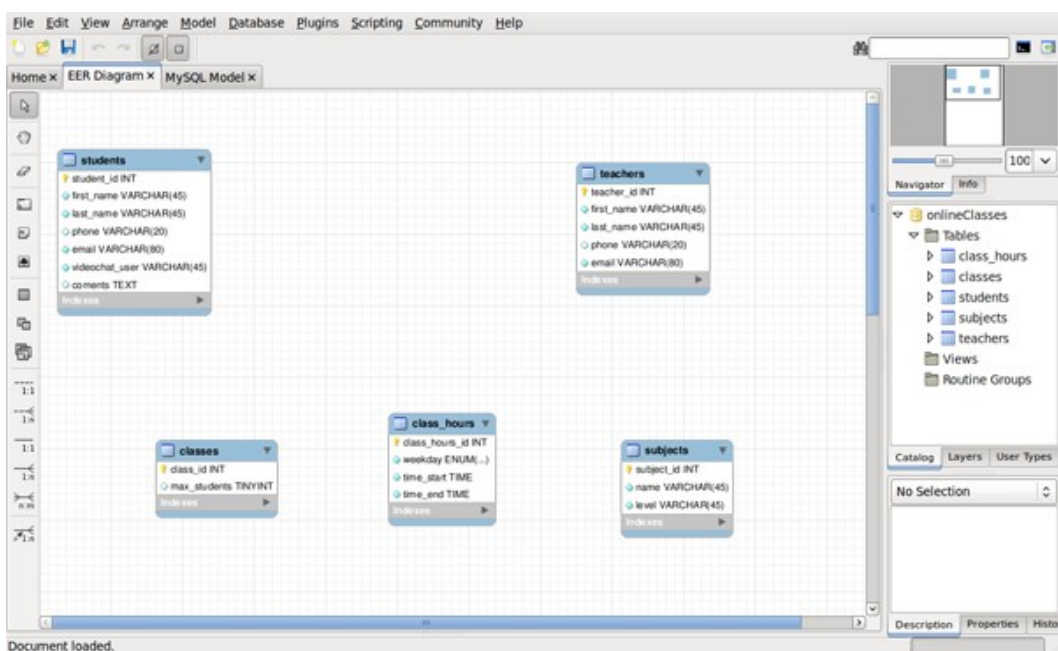
Перейдите в меню и выберите *Model/Create Diagram from Catalog Objects* (Модель/Создание диаграммы из каталога объектов), теперь у нас есть схема, и мы готовы продолжать.

Выберите слева значок таблицы; указатель примет форму руки с таблицей. Затем щелкните в любом месте холста, чтобы создать новую таблицу.



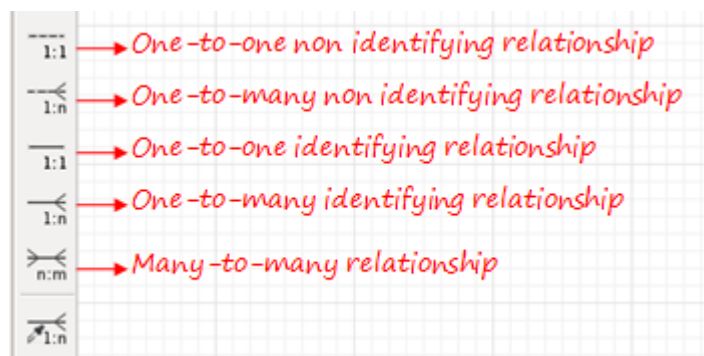
Теперь вы просто должны дважды щелкнуть таблицу, и в редакторе появится вкладка для изменения имени, столбцов, типов и т.д. - так же, как это делалось раньше.

После ввода деталей столбца для новых таблиц, мы будем готовы начать рисовать отношения.



## Рисование отношений

В вертикальной панели инструментов слева есть шесть инструментов, доступных для создания отношений.



Не беспокойтесь о последнем, мы объясним его назначение позже. Для отношений 1:1 и 1: n, мы имеем два различных типа символов: *идентифицирующие* и *неидентифицирующие*. Что это значит?

Отношение считается *идентифицирующим*, когда одна таблица полностью зависит от других, и не может существовать без них. Строка в такой таблице зависит от строки в другой таблице. Типичным примером является создание отдельной таблицы для хранения телефонов пользователей. Их необходимо хранить в другой таблице, потому что у одного пользователя может быть несколько телефонов, но каждая строка в этой таблице, полностью зависит от пользователя - она *относится* к пользователю.

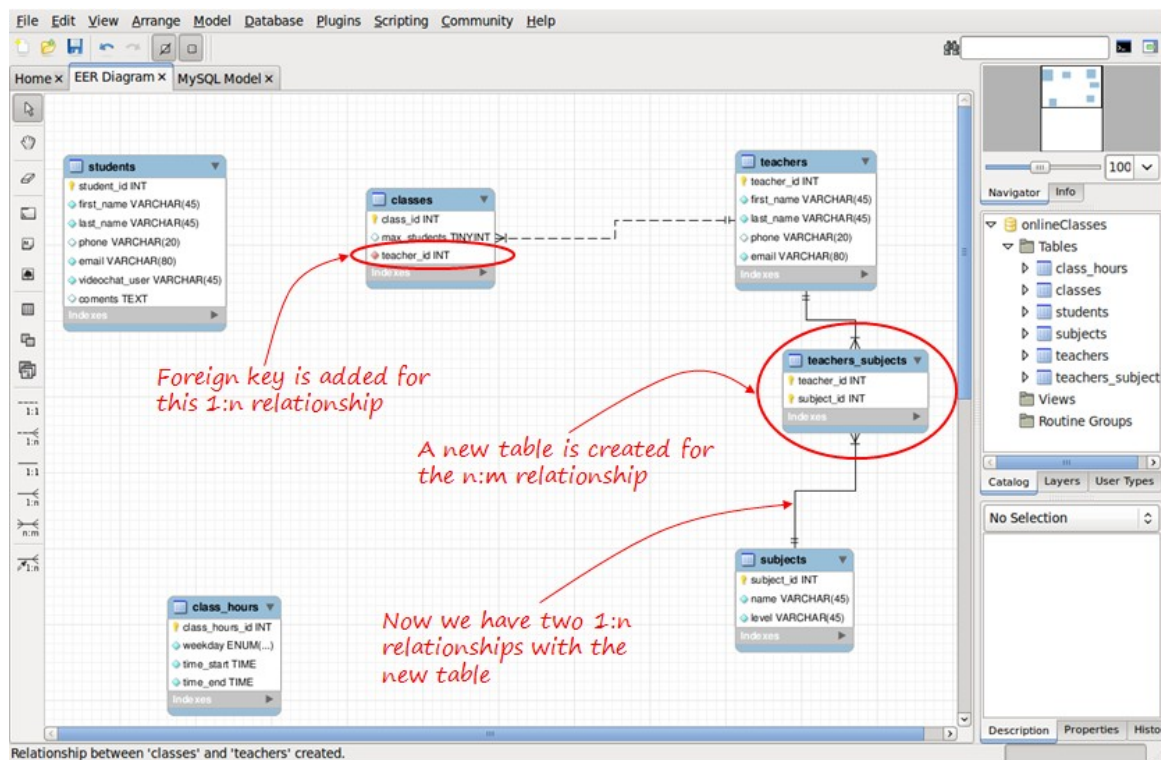
Вы должны знать, что отношения бывают определенных типов. Если мы хотим создать физические таблицы в MySQL, то отношения между таблицами должны быть каким-то образом отображены. Есть несколько правил, которые определяют отношения между таблицами:

- **отношение 1:1.** Первичный ключ для одной из таблиц включен в качестве внешнего ключа в другой таблице.
- **отношение 1:n.** Первичный ключ из таблицы **1** добавляется в качестве внешнего ключа в таблицу **n**.
- **отношение n:m.** Создается новая таблица (таблица связи). Первичный ключ состоит из первичных ключей двух оригинальных таблиц.

Выявление отношения обычно используется для создания связующих таблиц для отношения "многие-ко-многим". Эти новые таблицы полностью зависят от двух оригинальных таблиц. Кроме того, в случае идентифицирующих отношений 1:1 и 1: n введенный внешний ключ становится частью первичного ключа для этой таблицы, образуя составной первичный ключ. Хорошим является то, что MySQL Workbench знает эти правила лучше, чем большинство из нас. Мы просто рисуем линии отношений, и внешние ключи или таблицы связей будут создаваться автоматически. Как мы убедимся в ближайшее время, это также можно сделать вручную.

Чтобы нарисовать отношение, щелкните по значку, а затем по двум таблицам, которые необходимо связать. Для отношения "один-ко-многим" щелкните вначале по таблице "многие", а затем по таблице "один". Давайте посмотрим, как сделать это для отношения **n:m** учителя-предметы и для отношения **1:n** учителя-курсы.





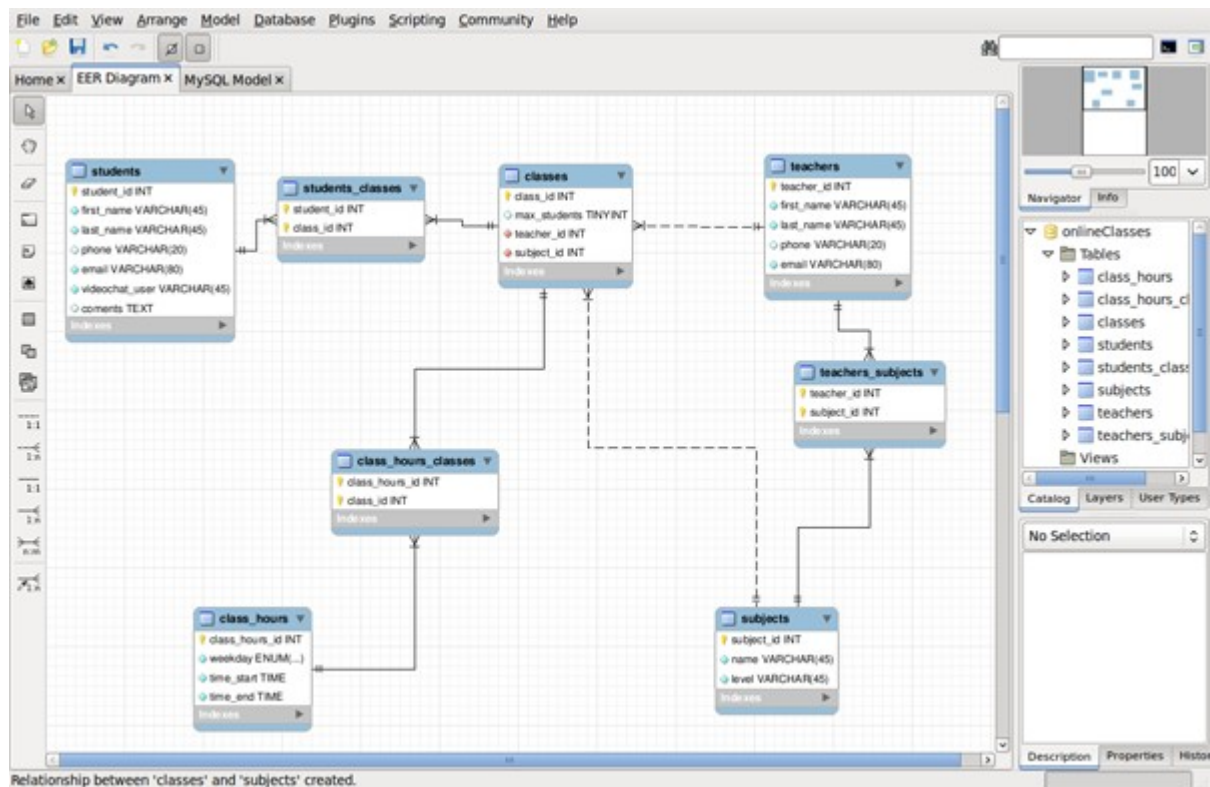
Имя по умолчанию, назначаемое для внешних ключей, а также для таблиц связи может быть изменено на глобальном уровне в *Edit/Preferences/Model Tab*, или только для данного проекта в *Model/Model Options*.

Если мы не хотим, чтобы таблицы и внешние ключи были созданы таким образом, мы можем использовать таинственный "шестой символ".

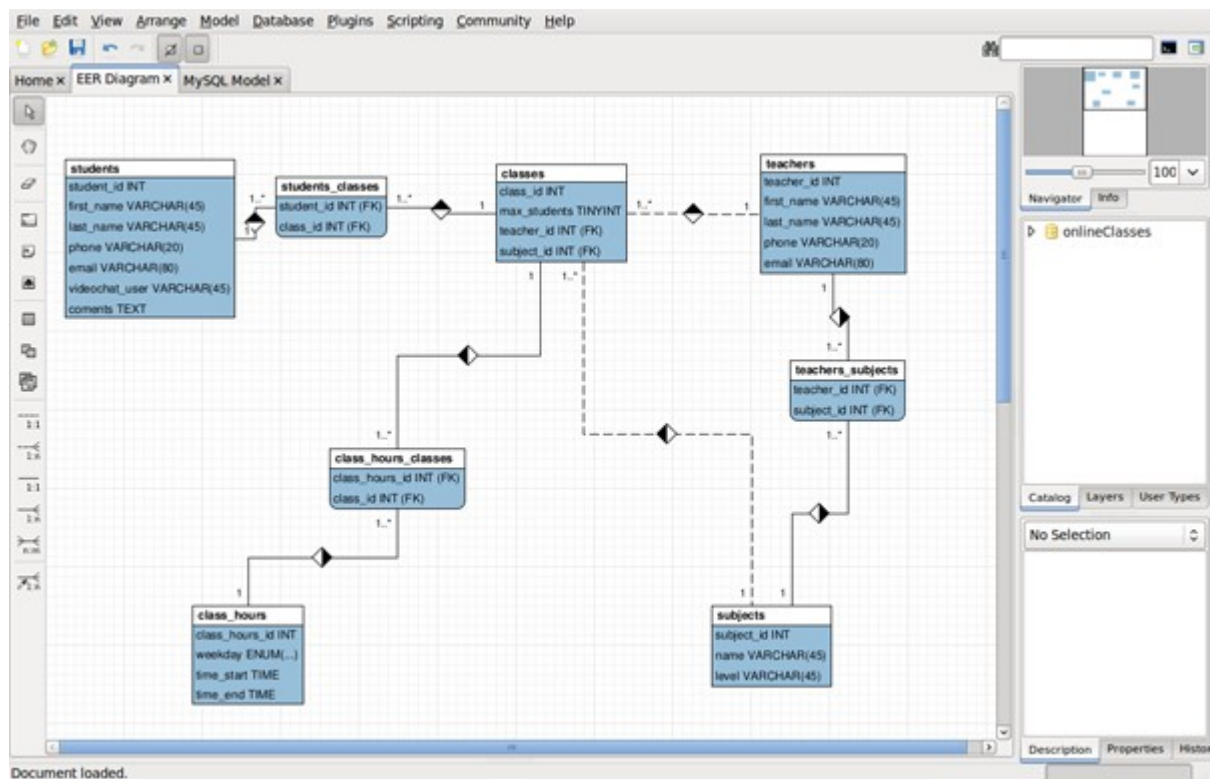


"Шестой символ" создает связь с использованием уже существующих колонок, т.е. вы уже включили необходимые внешние ключи в таблицы и создали необходимые таблицы связи (таблицы, отображающие отношение **n:m**). Поскольку таблицы связи уже созданы, нам не нужно отношение **n:m**, доступно только отношение **1:n**.

Когда все отношения определены, наша схема должна выглядеть так:



Помните, что для диаграмм используются обозначения по умолчанию MySQL Workbench, но вы можете их изменить в *Model/Object Notation* и *Model/Relationship Notation*. Это пример нашей модели с классическими обозначениями:



На данный момент наша модель готова, и мы можем сгенерировать код SQL для создания базы данных MySQL.

# Создание кода SQL

Выберите *File/Export/Forward Engineer SQL CREATE Script*. Для генерации файла с кодом SQL понадобятся только три экрана мастера!

The screenshot shows the 'SQL Export Options' dialog box. On the left, there is a sidebar with three options: 'SQL Export Options' (selected), 'Filter Objects', and 'Review SQL Script'. The main area is titled 'SQL Export Options' and contains the following elements:

- 'Output SQL Script File:' with a text field containing 'nents/tutorials/MySQL WorkBench/testSQLscript' and a 'Browse...' button.
- A note: 'Leave blank to view generated script but not save to a file.'
- A section titled 'SQL Options' containing a list of checkboxes:
  - ☐ Generate DROP Statements Before Each CREATE Statement
  - ☐ Generate DROP SCHEMA
  - ☐ Skip Creation of FOREIGN KEYS
  - ☐ Omit Schema Qualifier in Object Names
  - ☐ Generate USE statements
  - ☐ Generate Separate CREATE INDEX Statements
  - ☐ Add SHOW WARNINGS After Every DDL Statement
  - ☐ Do Not Create Users. Only Export Privileges
  - ☐ Generate INSERT Statements for Tables

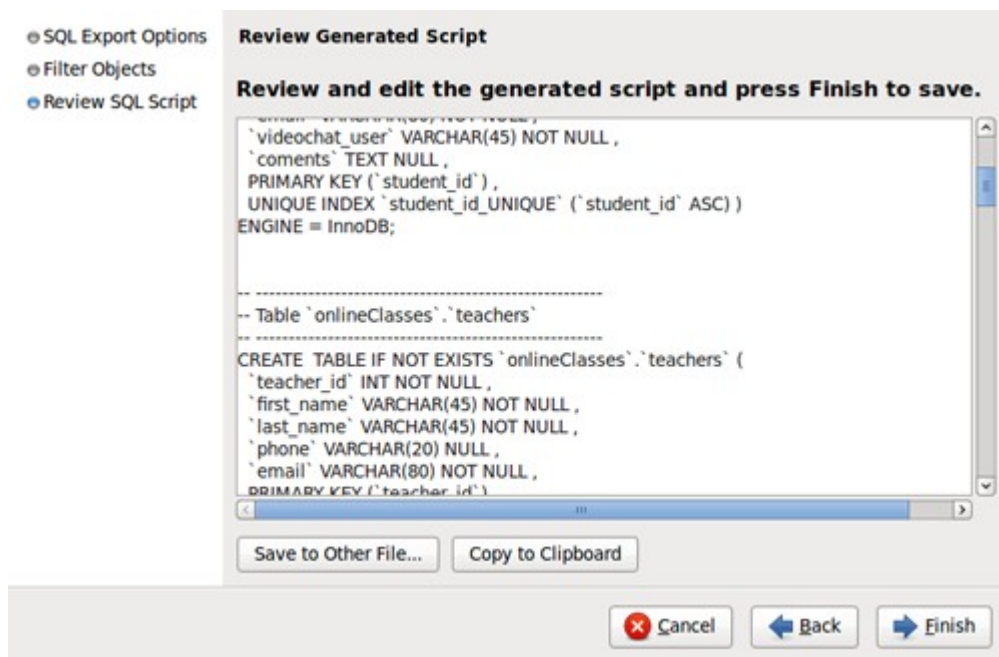
At the bottom right, there are three buttons: 'Cancel' (with a red X icon), 'Back' (with a left arrow icon), and 'Next' (with a right arrow icon).

The screenshot shows the 'SQL Object Export Filter' dialog box. On the left, there is a sidebar with three options: 'SQL Export Options', 'Filter Objects' (selected), and 'Review SQL Script'. The main area is titled 'SQL Object Export Filter' and contains the following elements:

- A text block: 'To exclude objects of a specific type from the SQL Export, disable the corresponding checkbox. Press Show Filter and add objects or patterns to the ignore list to exclude them from the export.'
- A list of object types with checkboxes and 'Show Filter' buttons:
  - ☒ Export MySQL Table Objects (8 Total Objects, 8 Selected)
  - ☐ Export MySQL View Objects (0 Total Objects, 0 Selected)
  - ☐ Export MySQL Routine Objects (0 Total Objects, 0 Selected)
  - ☐ Export MySQL Trigger Objects (0 Total Objects, 0 Selected)

At the bottom right, there are three buttons: 'Cancel' (with a red X icon), 'Back' (with a left arrow icon), and 'Next' (with a right arrow icon).

У нас даже есть возможность просматривать и редактировать сгенерированный код SQL перед его сохранением:



И это все. Щелкните по кнопке *Finish*, скрипт будет создан и сохранен. Теперь мы можем использовать его когда захотим. Например, мы можем загрузить его с помощью командной строки MySQL клиента:

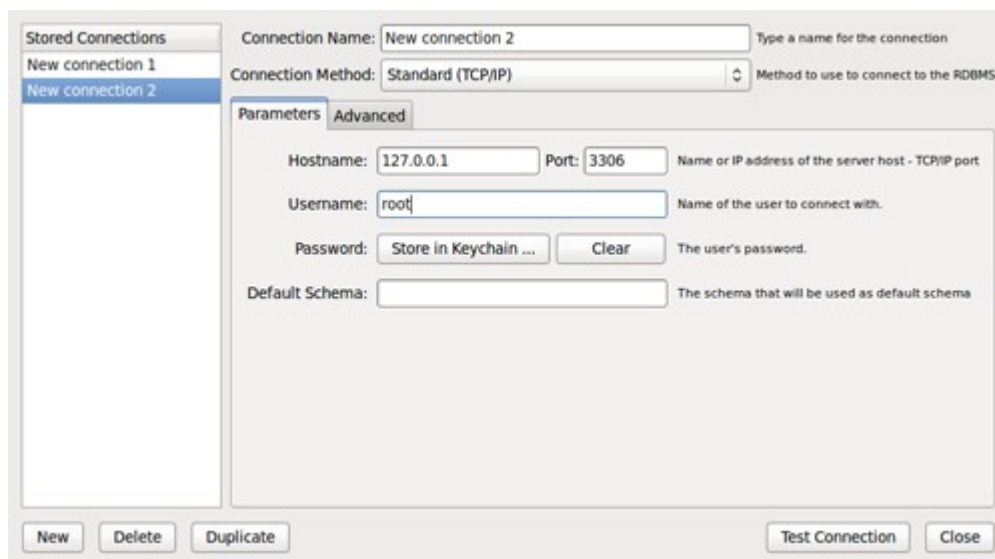
```
mysql> SOURCE scriptName.sql
```

Или, чтобы закончить работу, можно использовать MySQL Workbench, подключившись к серверу MySQL и запустив сценарий.



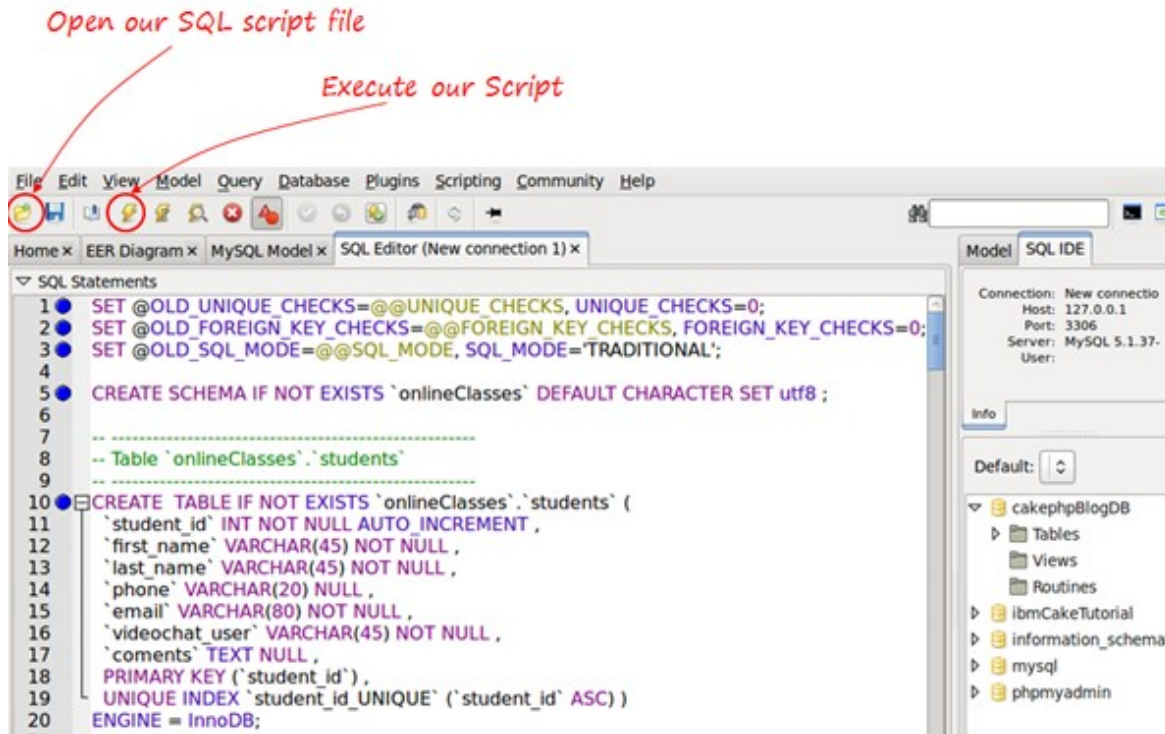
# Подключение к серверу MySQL

Выберите *Database/Manage Connections* , и щелкните по кнопке *New*.



Если вы не установите пароль здесь, то позже при необходимости вам будет предложено это сделать. Нажмите кнопку "Test Connection (Проверить подключение)", чтобы проверить, верны ли параметры, а затем нажмите кнопку "Close (Заккрыть)".

Теперь, чтобы загрузить сценарий, мы будем использовать редактор SQL. В главном меню выберите *Database/Query Database*; в появившемся окне вам будет предложено выбрать подключение, а затем на вкладке откроется редактор SQL.



Теперь, чтобы выполнить сценарий SQL, щелкните значок молнии, и ваша база данных будет создана!

Мы могли бы также сформировать базу данных MySQL непосредственно из модели, без ссылки на сам файл, используя меню *Database/Forward Engineer*, однако, я считаю полезным создать сценарий, а затем использовать его как хочется.



## Заключение

MySQL Workbench является мощной программой. Мы просмотрели только несколько основных возможностей в части моделирования данных, и заглянули в редактор SQL только во второй половине этого учебного пособия. Мы узнали, как визуально создавать базу данных и рисовать диаграммы, которые могут храниться в качестве документации. Вы можете экспортировать диаграммы как PNG, SVG, PDF или PostScript файл.

Спасибо за чтение, и дайте мне знать, что вы думаете!