

## Лабораторная работа №4. Создание базы данных средствами MySQL Workbench

С ростом проектов увеличивается сложность программной части, неизбежно возрастает количество обрабатываемых ею данных, а так же **сложность схемы данных**.

**MySQL Workbench - бесплатная программа для работы с MySQL.**

MySQL Workbench - инструмент для визуального проектирования баз данных, интегрирующий проектирование, моделирование, создание и эксплуатацию БД в единое окружение для системы баз данных MySQL.

Программа позволяет быстро делать схемы данных проекта, проектировать сущности и связи между ними, внедрять изменения в схему и так же быстро синхронизировать её с удалённым сервером.

А графический редактор **EER-диаграмм** позволяет увидеть общую картину модели данных.

### 1. Скачать MySQL Workbench

Переходим по ссылке <https://dev.mysql.com/downloads/workbench/>

Перед скачиванием требуется выбрать одну из следующих платформ:

Microsoft Windows (*доступен MSI Installer 37.2Mb*)

После выбора платформы вам предлагают зарегистрироваться или авторизоваться в Oracle.

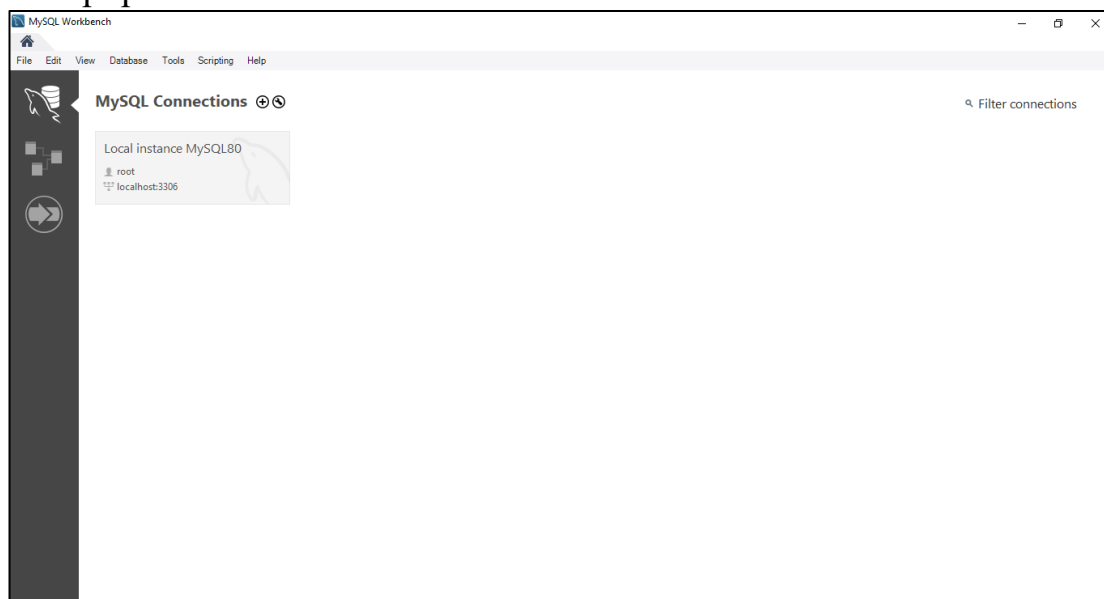
Если не хотите, внизу есть ссылка "**No thanks, just start my download**" - жмите на неё ;)

General Availability (GA) Releases			
<b>MySQL Workbench 8.0.18</b>			
Select Operating System:		<a href="#">Looking for previous GA versions?</a>	
Microsoft Windows			
<b>Recommended Download:</b>			
<div><b>MySQL Installer for Windows</b> All MySQL Products. For All Windows Platforms. In One Package. <small>Starting with MySQL 5.6 the MySQL Installer package replaces the standalone MSI packages.</small></div> <div>Windows (x86, 32 &amp; 64-bit), MySQL Installer MSI</div> <div><a href="#">Go to Download Page &gt;</a></div>			
<b>Other Downloads:</b>			
Windows (x86, 64-bit), MSI Installer (mysql-workbench-community-8.0.18-winx64.msi)	8.0.18	37.2M	<a href="#">Download</a>
MD5: fbba5574b7d5f1587b14683fe706dc88   <a href="#">Signature</a>			

Во время установки нажимать далее.

## 2. Начало работы

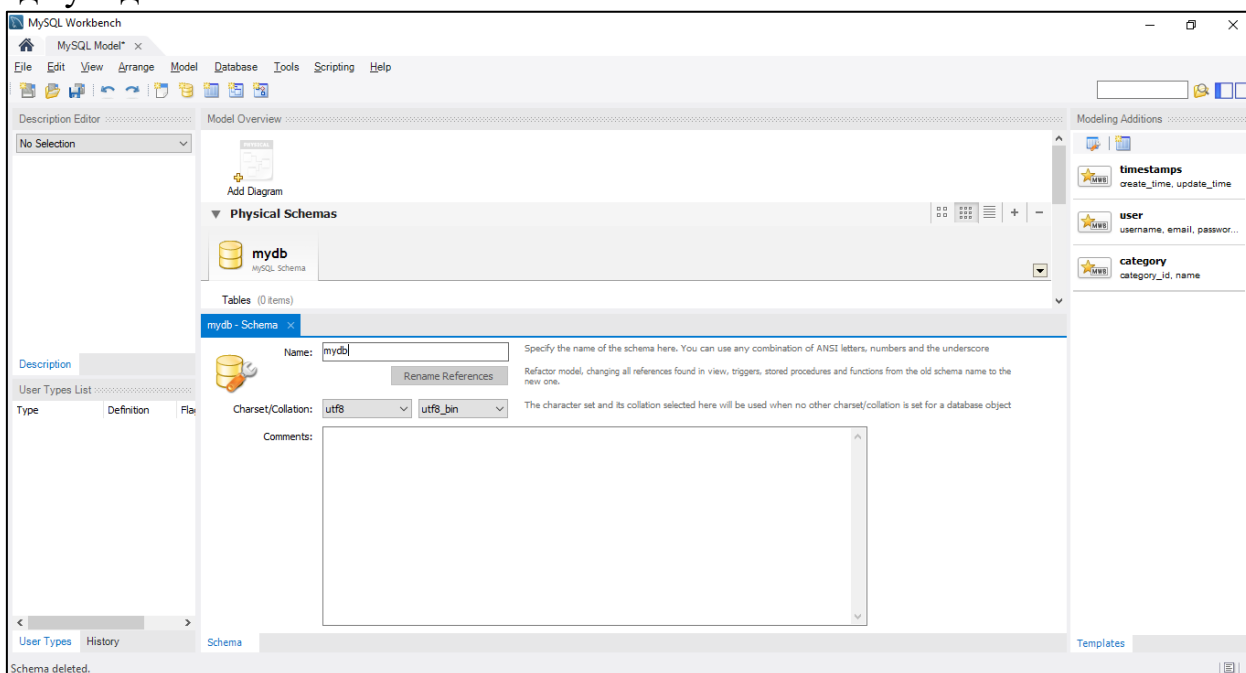
Стартовый экран программы отражает основные направления её функциональности - проектирование моделей баз данных и их администрирование:



В верхней части экрана находится список подключений к MySQL серверам ваших проектов, а список последних открытых моделей данных - в нижней части экрана. Работа обычно начинается с **создания схемы данных** или **загрузки существующей структуры в MySQL Workbench**.

## 3. Создание и редактирование модели данных

Для добавления модели нажимаем плюсик рядом с заголовком "Models" или выбираем "*File → New Model*" (*Ctrl + N*): После дважды кликаем на mydb, где увидим окно:



На этом экране вводим имя базы данных, выбираем кодировку по умолчанию и, если нужно, заполняем поле комментария. Можно приступить к созданию таблиц.

#### 4. Добавление и редактирование таблицы

Список баз данных проекта и список таблиц в пределах базы данных будет располагаться во вкладке "*Physical Schemas*".

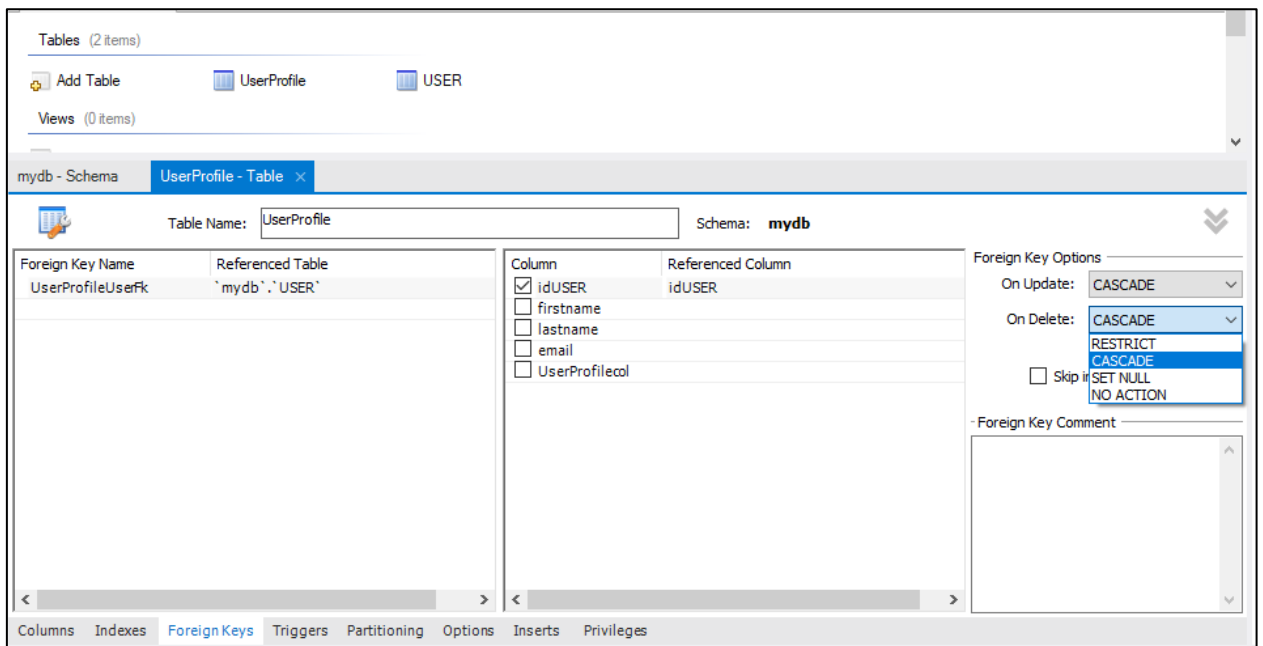
Чтобы создать таблицу, дважды кликаем на "+Add Table":

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
idUSER	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
username	VARCHAR(128)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
password	VARCHAR(32)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Откроется удобный интерфейс для редактирования списка полей и их свойств. Здесь можно задать название поля, тип данных, а так же установить для полей различные атрибуты: назначить поле *первичным ключом (PK)*, пометить его *Not Null (NN)*, *бинарным (BIN)*, *уникальным (UQ)* и другие, установить для поля *авто-инкрементирование (AI)* и *значение по умолчанию (Default)*. (Колонки username и password созданы вручную.)

#### 5. Связи между таблицами

Установка внешних ключей и связывание таблиц возможно только для таблиц *InnoDB* (эта система хранения данных выбирается по умолчанию). Для управления связями в каждой таблице находится вкладка "*Foreign Keys*":



Для добавления связи открываем вкладку *"Foreign Keys"* дочерней таблицы, вводим имя внешнего ключа и выбираем **таблицу-родителя**. Далее в средней части вкладки в графе *Column* выбираем поле-ключ из дочерней таблицы, а в графе *Referenced Column* - соответствующее поле из родительской таблицы (тип полей должен совпадать). При создании внешних ключей **в дочерней таблице автоматически создаются соответствующие индексы**.

В разделе *"Foreign Key Options"* настраиваем поведение внешнего ключа при изменении соответствующего поля (*ON UPDATE*) и удалении (*ON DELETE*) родительской записи:

- *RESTRICT* - выдавать ошибку при изменении / удалении родительской записи

- *CASCADE* - обновлять внешний ключ при изменении родительской записи, удалять дочернюю запись при удалении родителя

- *SET NULL* - устанавливать значение внешнего ключа *NULL* при изменении / удалении родителя (**неприемлемо для полей, у которых установлен флаг *NOT NULL*!**)

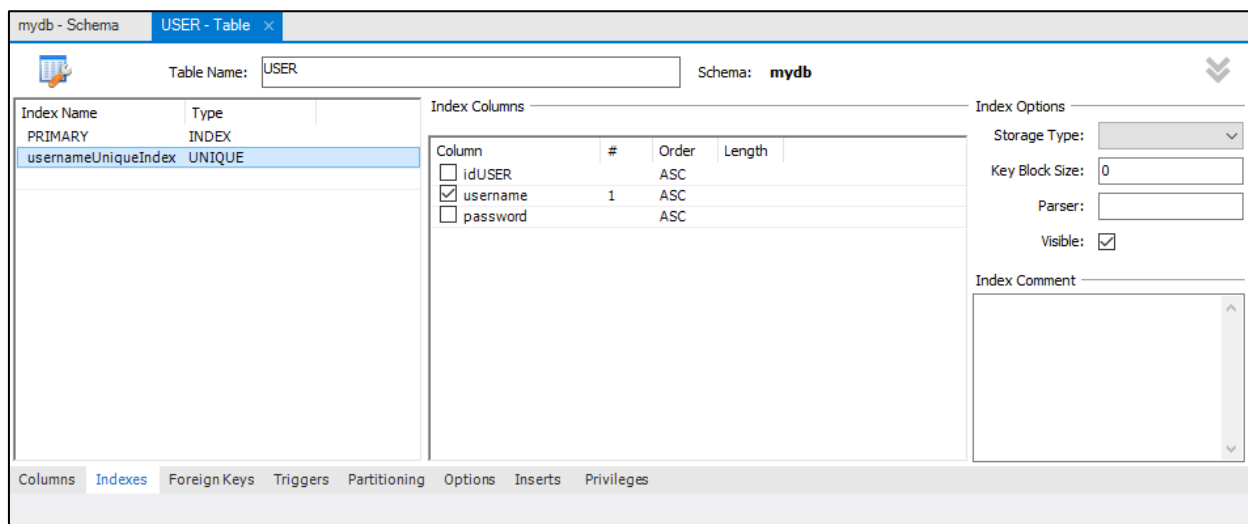
- *NO ACTION* - не делать ничего, однако по факту эффект аналогичен *RESTRICT*

В приведённом примере к дочерней таблице *UserProfile* добавлен внешний ключ для связи с родительской таблицей *User*.

При редактировании поля *userId* и удалении позиций из таблицы *User* аналогичные изменения будут **автоматически** происходить и со связанными записями из таблицы *UserProfile*.

## 6. Управление индексами

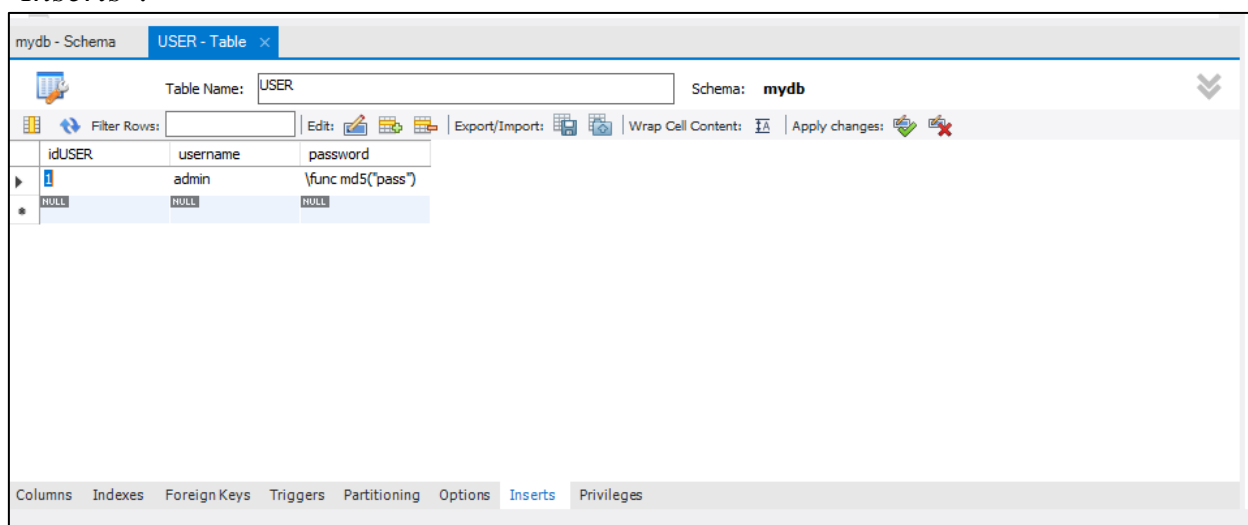
Добавлять, удалять и редактировать индексы таблиц можно во вкладке *"Indexes"* интерфейса управления таблицей:



Вводим название индекса, выбираем его тип, затем галочками помечаем в нужном порядке список полей, участвующих в данном индексе. Порядок полей будет соответствовать порядку, в котором были проставлены галочки. В данном примере я добавил уникальный индекс к полю *username*.

### 7. Заполнение таблицы базовыми данными

При создании проекта в базу данных часто нужно добавлять стартовые данные. Это могут быть корневые категории, пользователи-администраторы и т.д. В управлении таблицами MySQL Workbench для этого существует вкладка "Inserts":



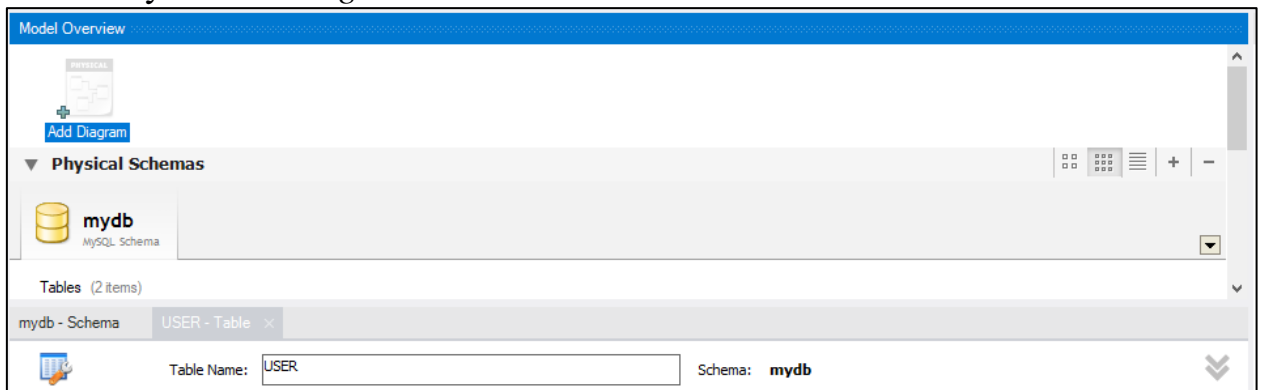
Как видно из примера, в случае, если перед записью в базу данных к данным нужно применить какую-то функцию MySQL, это делается с помощью синтаксиса `\func functionName('data')`, например, `\func md5('password')`.

После ввода данных необходимо сохранить их в локальную базу данных нажатием на кнопку "Apply Changes".

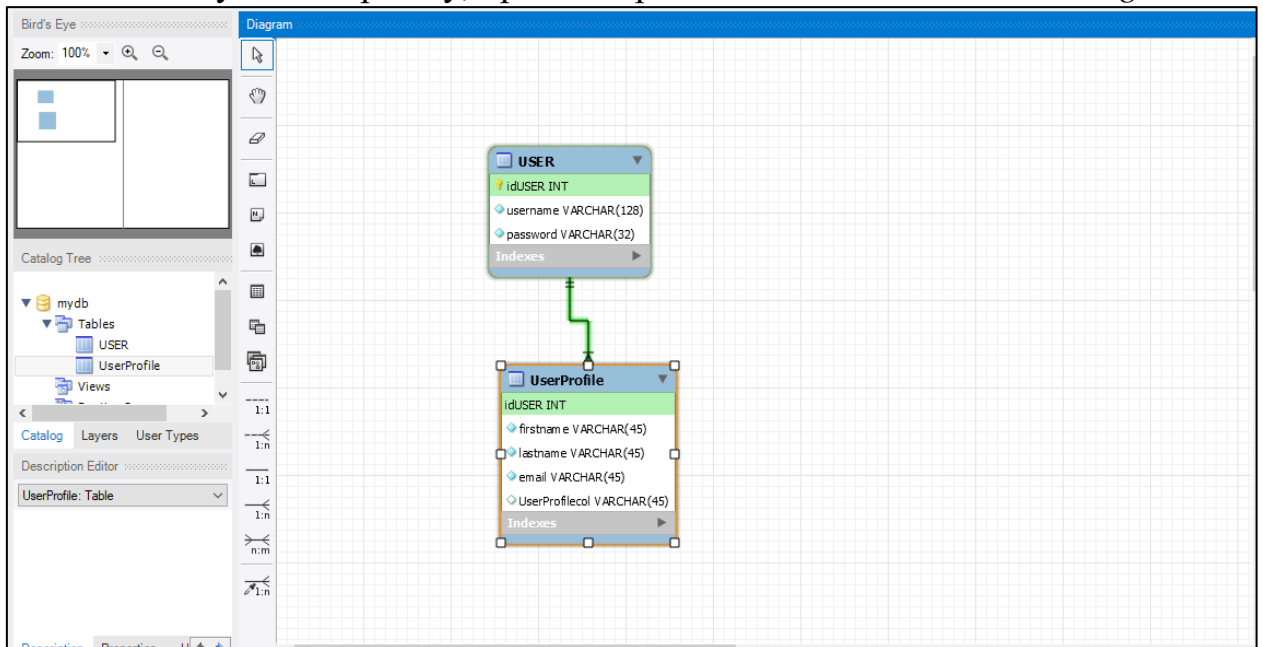
### 8. Создание EER диаграммы (диаграммы "сущность-связь")

Для представления схемы данных, сущностей и их связей в графическом виде в MySQL Workbench существует редактор EER-диаграмм. Для создания

диаграммы в верхней части экрана управления базой данных дважды кликаем на иконку "+Add Diagram":



В его интерфейсе можно создавать и редактировать таблицы, добавлять между ними связи различных типов. Чтобы добавить уже существующую в схеме таблицу на диаграмму, просто перетащите её из панели "Catalog Tree".

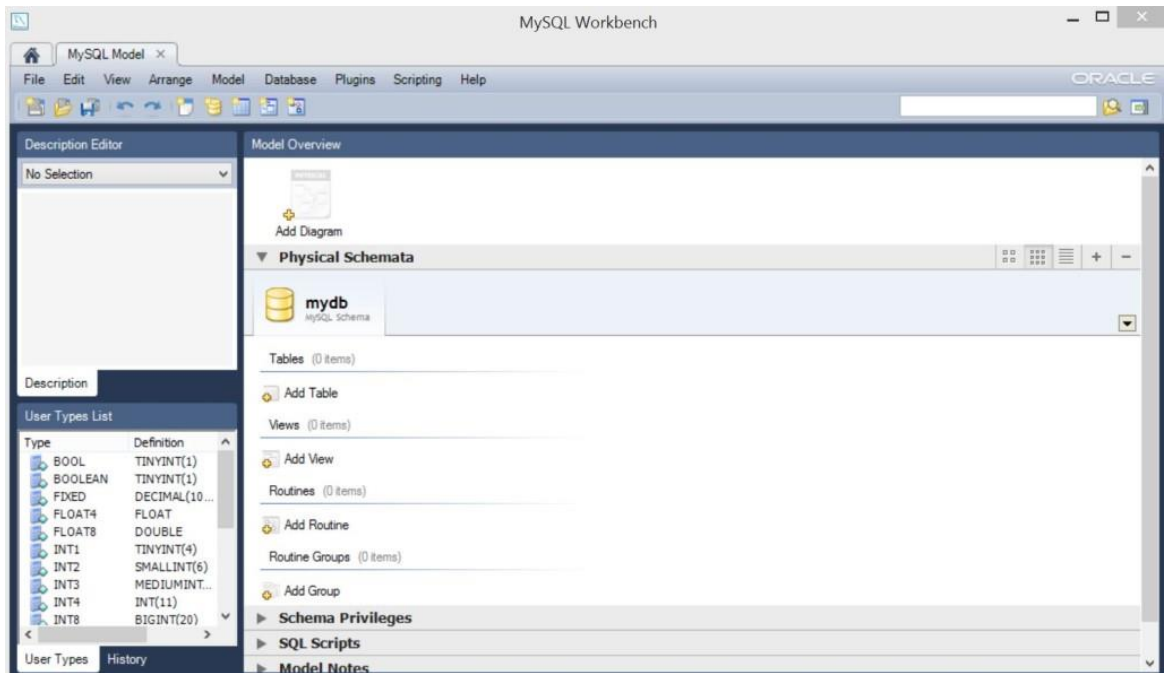


## ЗАДАНИЕ:

1. Создать модель БД.
2. Из модели через выполнение скрипта создать реальную базу с таблицами.
3. Создать диаграмму ER.

### Выполнение задания:

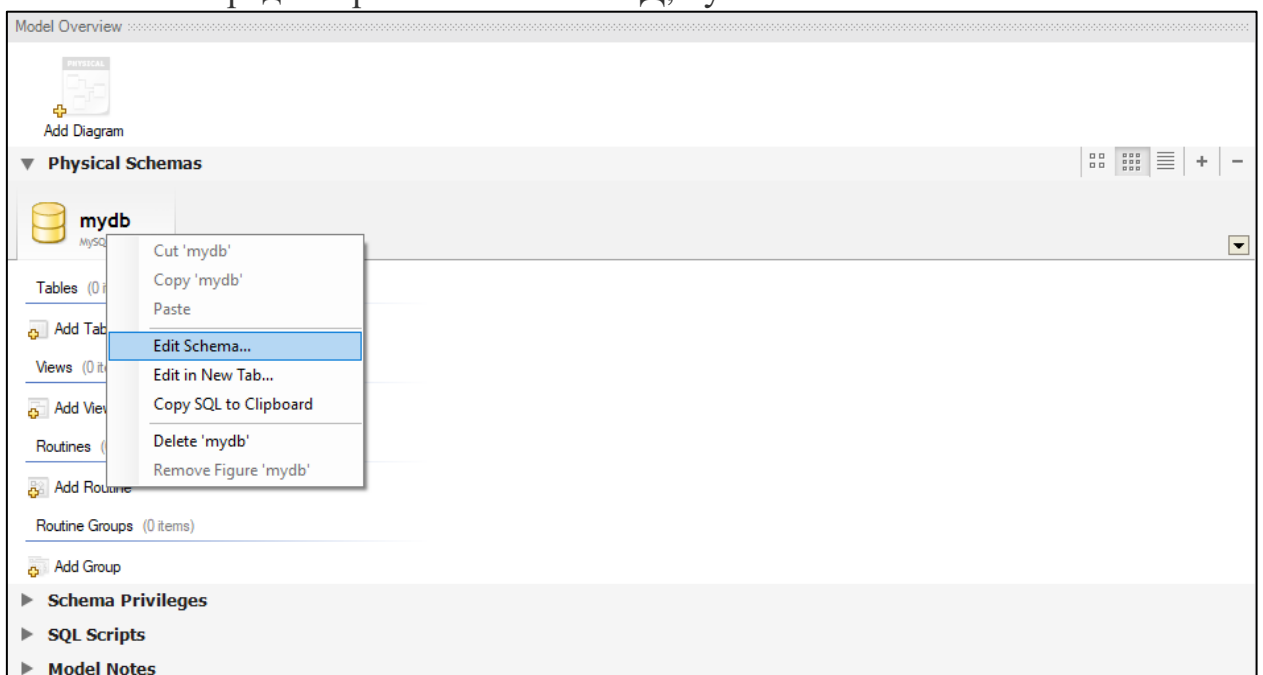
В программе MySQL WorkBench выбрать File New Model (Ctrl + N), открывается панель:



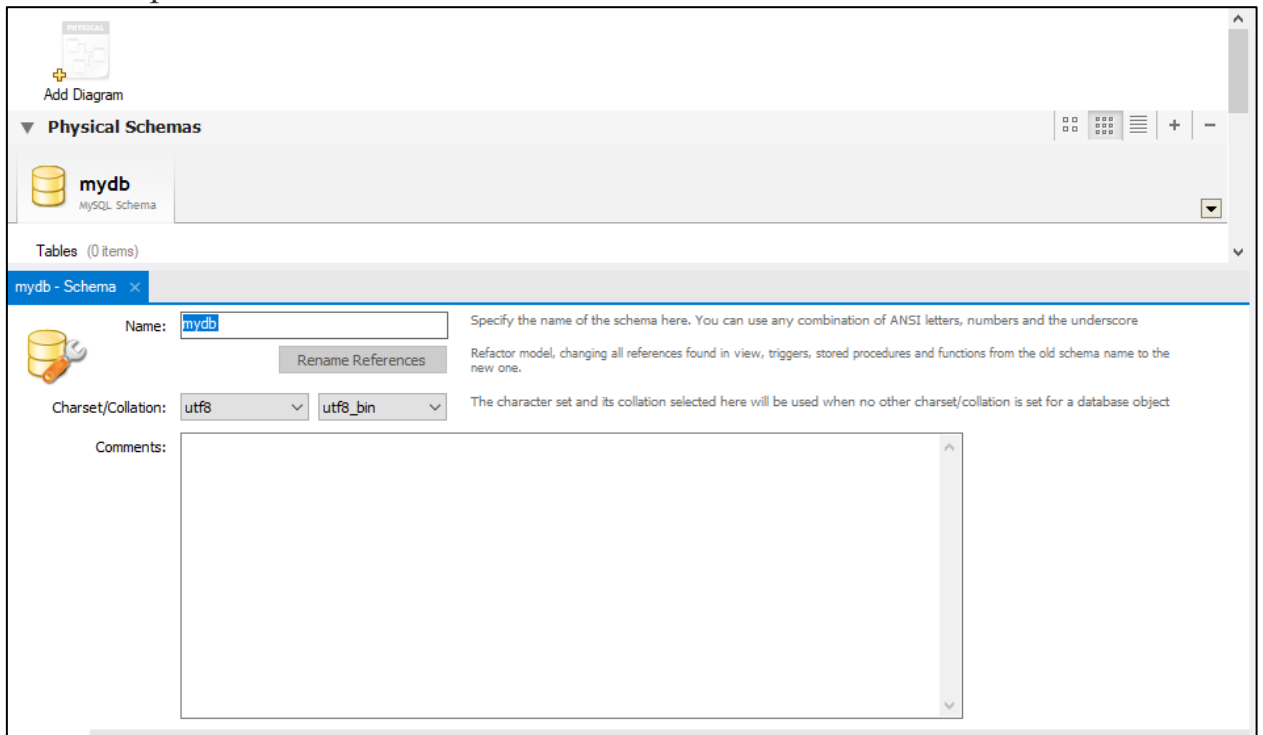
### 1. Создание базы данных (схемы) в модели

Выполните редакцию имени базы данных.

Чтобы отредактировать название БД, нужно выполнить:



Откроется окно...



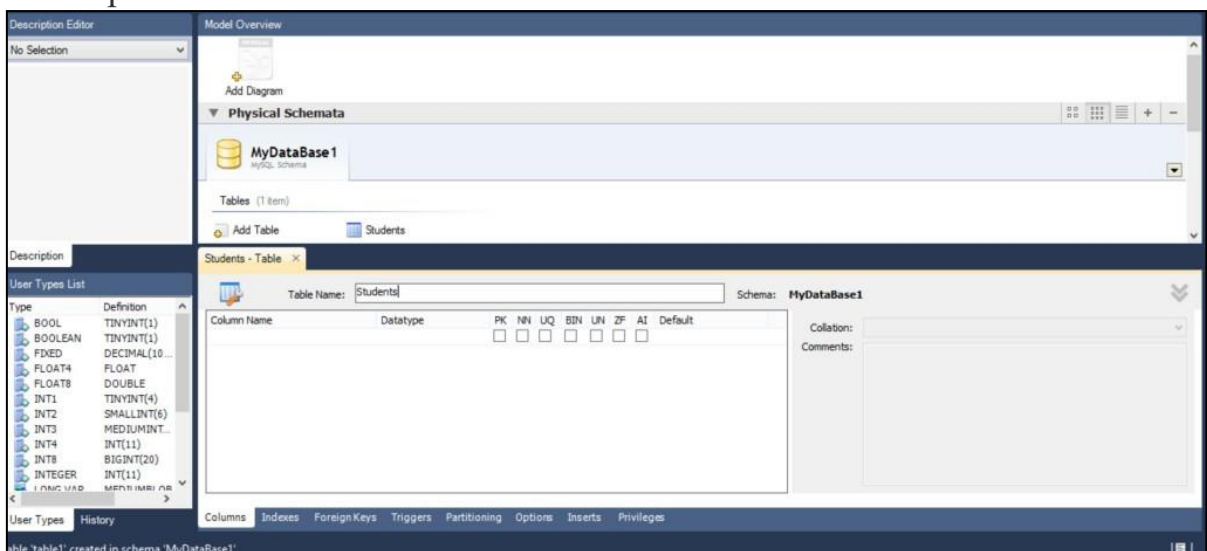
Введите название БД – **MyDataBase1**, кодировку не менять, в комментарии – “Первая база данных, учебная”;

После этого – нажать на крестик, на вкладке возле названия “MyDataBase1” и возвращаемся к предыдущему окну.

## 2 Создание таблиц в базе данных (схеме)

Создать таблицу “Students” и таблицу “Departments”

2 раза нажать на AddTable. Заполнить поле Table Name.



Заполняем имя таблицы. Обратите внимание на вкладки внизу, сейчас мы находимся на “Coloumns”.

Для таблицы студентов **Students** разработаем несколько полей.

-Primary\_key (уникальный ключ записей данной таблицы)

-Name (Имя)



-Family (Фамилия)

-Age (Возраст)

-Sex (Пол)

-Department\_id (Внешний ключ, каждый студент учится на каком-то факультете, соответственно будем отмечать это в данном поле);

И сразу же введём их, на вкладке Students -Table – Coloumns, результат получился такой:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
Primary_key	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Name	VARCHAR(128)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Family	VARCHAR(128)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Age	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

PK – поставить только у главного ключа, это поле является частью формирования уникального ключа таблицы.

NN – Not Nullled – отсутствие нулевых полей, так как у всех студентов есть Имя, Фамилия, Возраст, Пол.

AI – автоинкрементное поле – с добавлением новой записи значение в этом поле будет увеличиваться как минимум на единицу.

Аналогично создадим и настроим таблицу **Departments**.

В ней будет 2 поля Primary\_key и Department\_name.

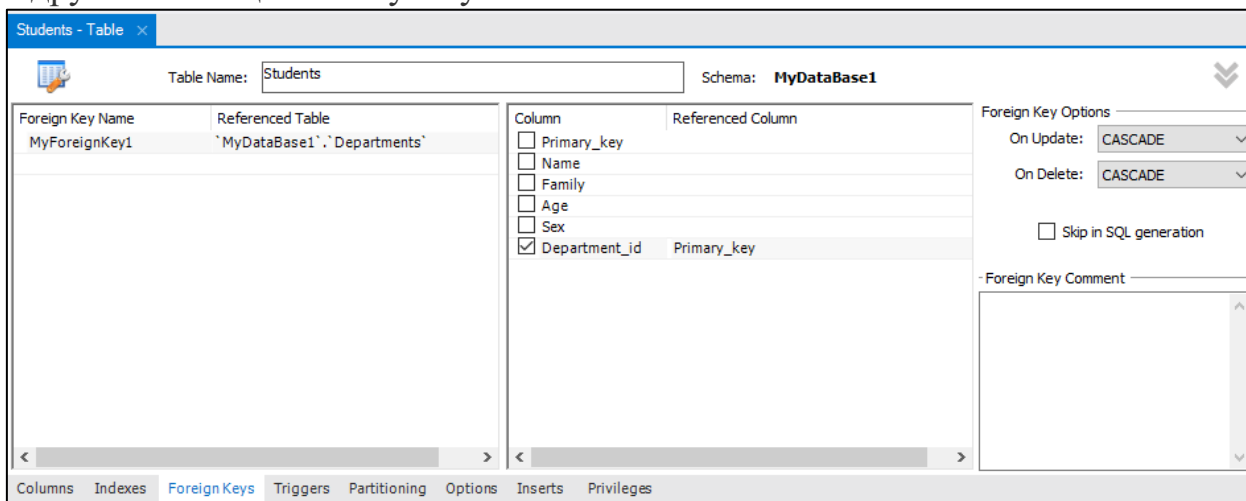
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
Primary_key	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Department_name	VARCHAR(128)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Department\_id в таблице **Students** и Primary\_key, в таблице **Departments** это практически одно и тоже, разница лишь в том, что значения внешнего

ключа Department\_id разбросаны по таблице, а Primary\_key автоинкрементен в своей таблице.

### 3. Создание связи между таблицами

На одном факультете может учиться несколько студентов, значит связь – один ко многим. Открыть вкладку Foreign Keys, дописать имя ключа (например, “MyForeignKey1”), в Referenced table выбрать Departments, а в правой части таблицы – выбрать в колонке Column Department\_id, поскольку это было имя внешнего ключа для таблицы **Students** и соответствующее поле в другой таблице Primary Key.

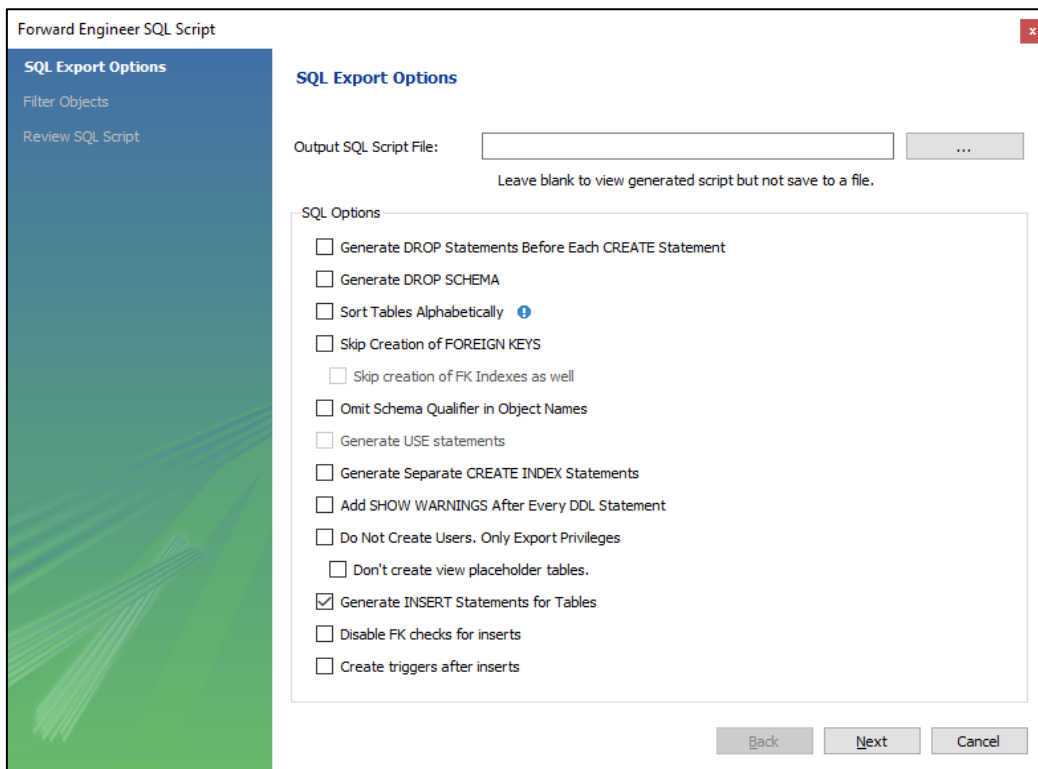


В разделе “*Foreign Key Options*” настраиваем поведение внешнего ключа при изменении соответствующего поля (*ON UPDATE*) и удалении (*ON DELETE*) родительской записи:

- *RESTRICT* – выдавать ошибку при изменении / удалении родительской записи
- *CASCADE* – обновлять внешний ключ при изменении родительской записи, удалять дочернюю запись при удалении родителя
- *SET NULL* – устанавливать значение внешнего ключа *NULL* при изменении / удалении родителя (**неприемлемо для полей, у которых установлен флаг *NOT NULL*!**)
- *NO ACTION* – не делать ничего, однако по факту эффект аналогичен *RESTRICT*

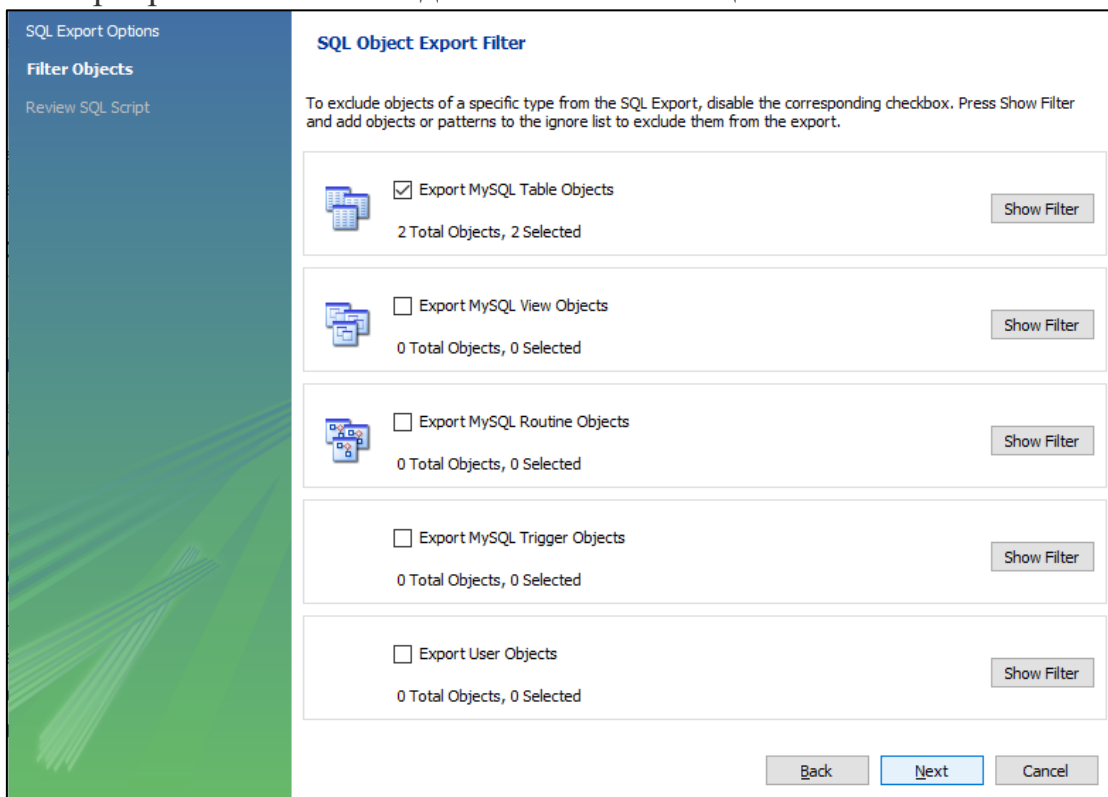
### 4. Сохранение из модели в реальную / физическую базу данных

“File → Export → Forward Engineer MySQL Create Script...”

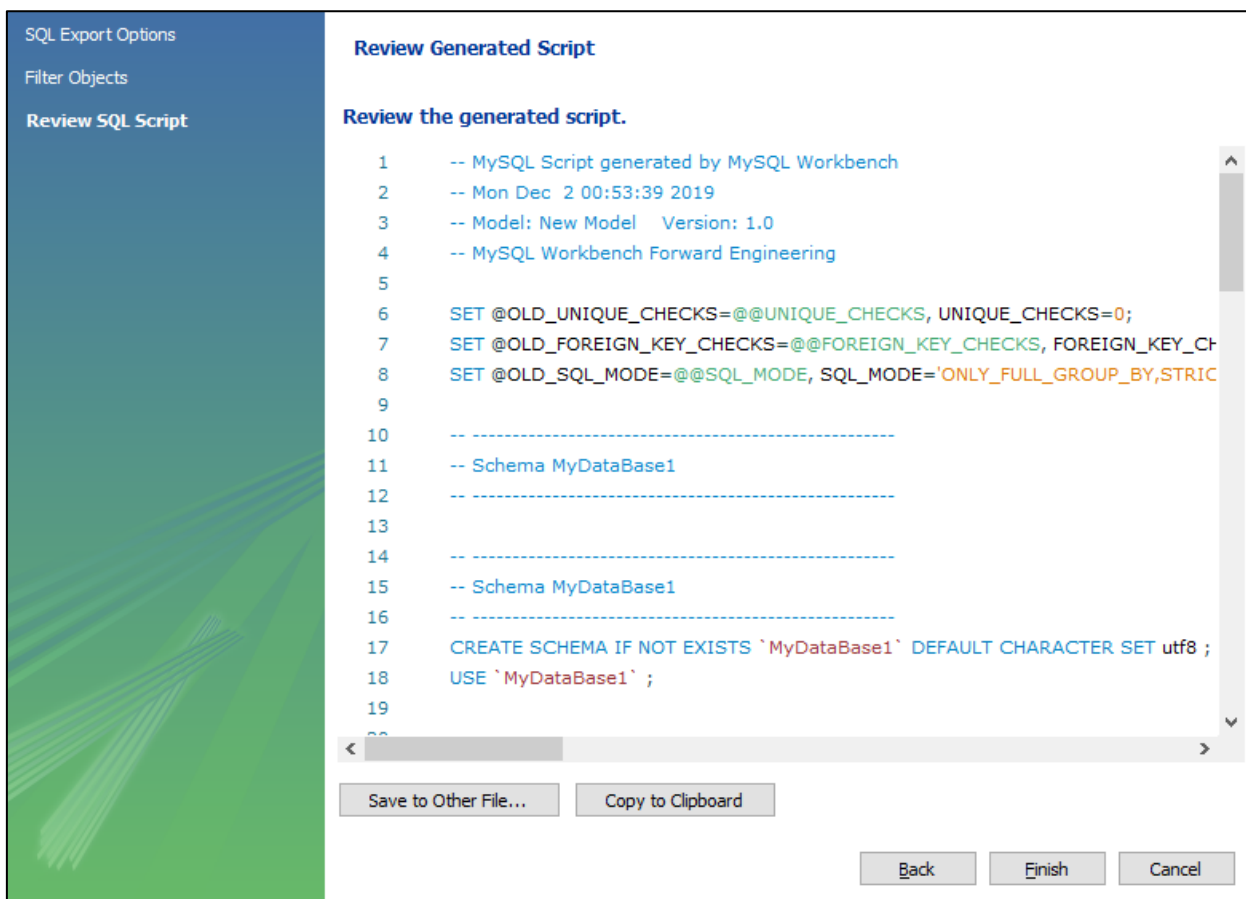


Отмечаем необходимые галочки (только одна Generate INSERT Statements for Tables).

В следующем окне можно настроить – какие объекты мы будем экспортировать. У нас создано всего 2 таблицы.



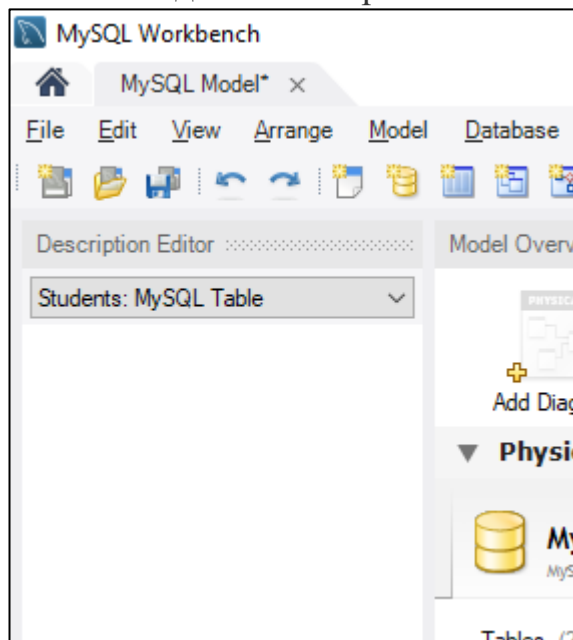
Жмем далее... и получаем скрипт



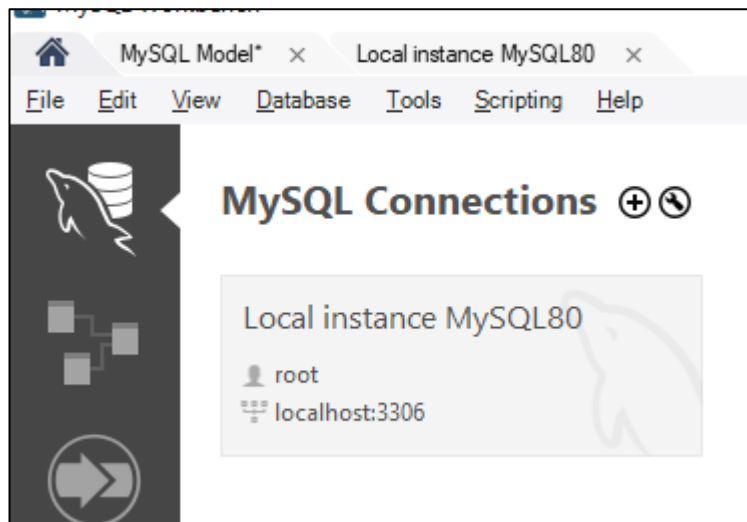
Копируем в буфер, но нужно этот скрипт где-то выполнить.

### 5. Выполнение скрипта – создания базы данных и таблиц

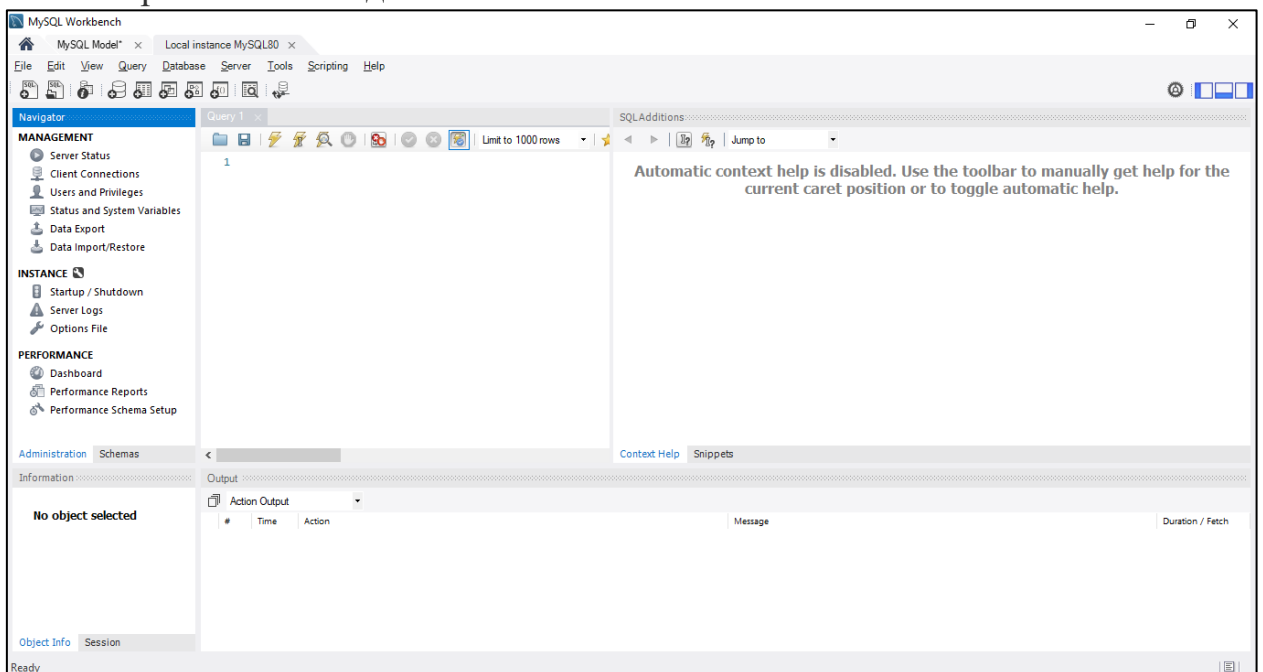
Жмем на “домик” в верхнем левом углу программы.



Потом 2 раза нажать на Local instance MySQL80. После вводим пароль пользователя root от mysql.



Открывается вкладка.

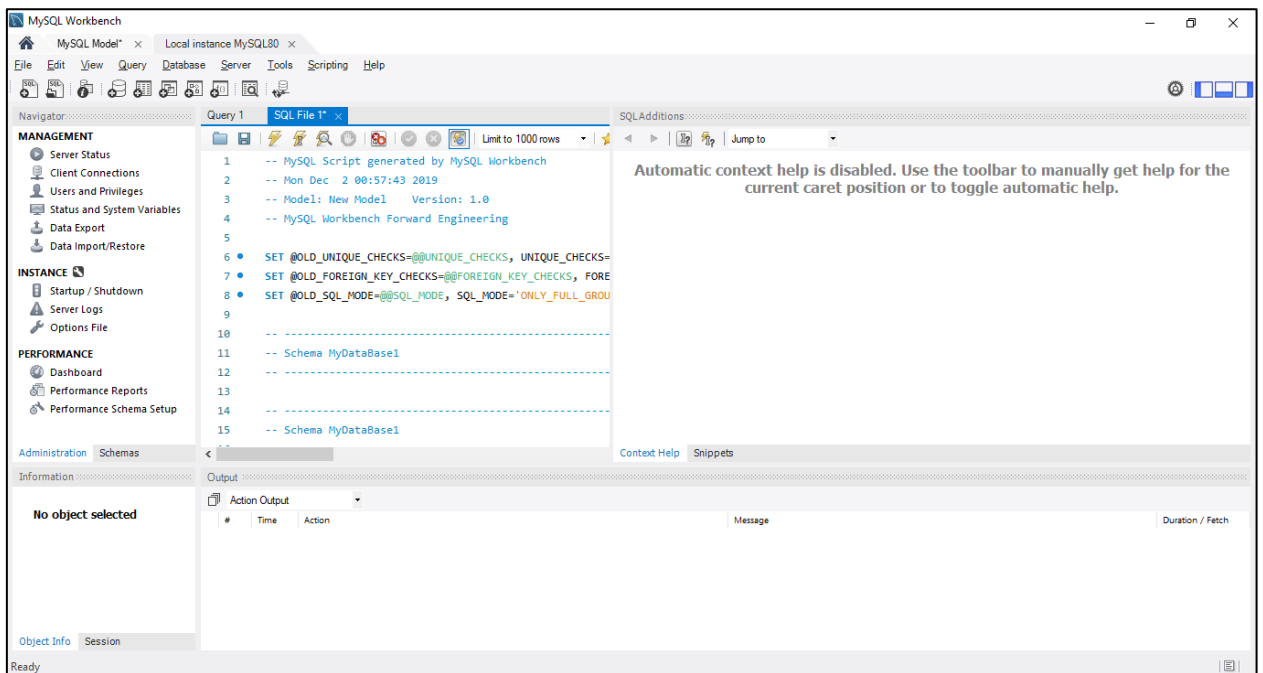


Это соединение с сервером, здесь будем выполнять скрипт.

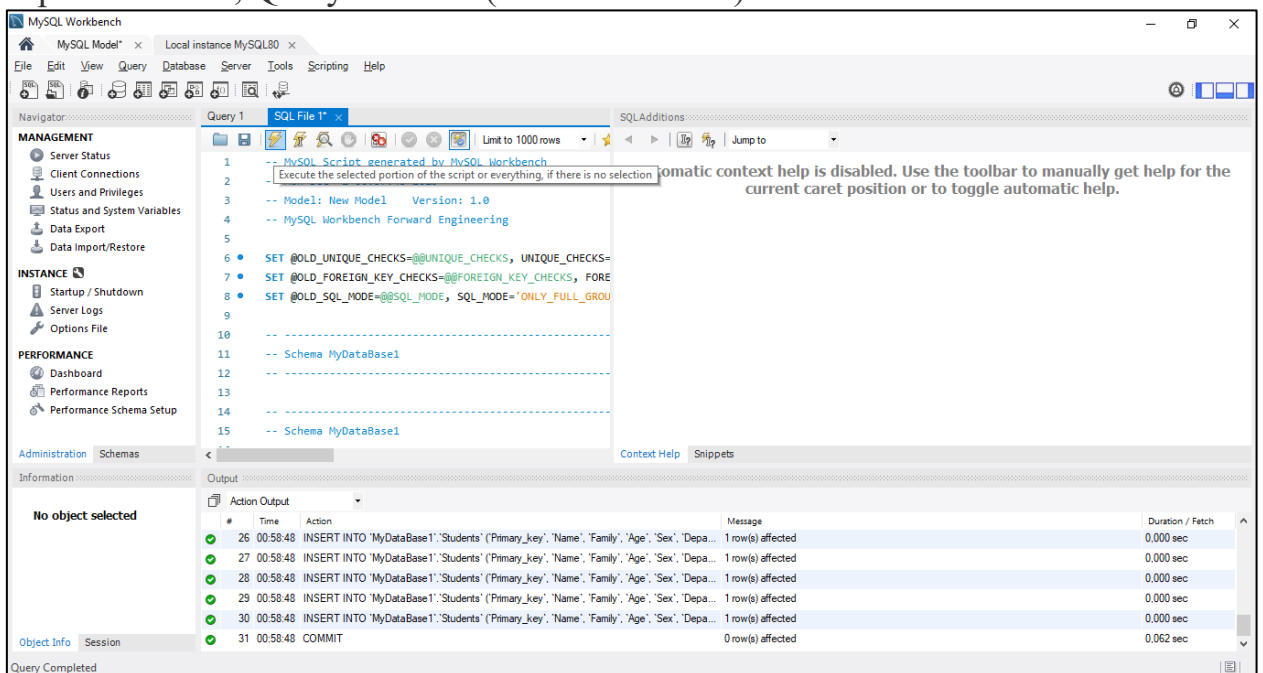
Обратите внимание, слева отображаются базы данных, которые были созданы в программе WorkBench.

Далее, File New Query Tab.

Вставляем скрипт в полученный Tab.



Теперь, нужно дать команду этот скрипт исполнить, для этого нажать в верхнем меню, Query Execute (All or Selection)

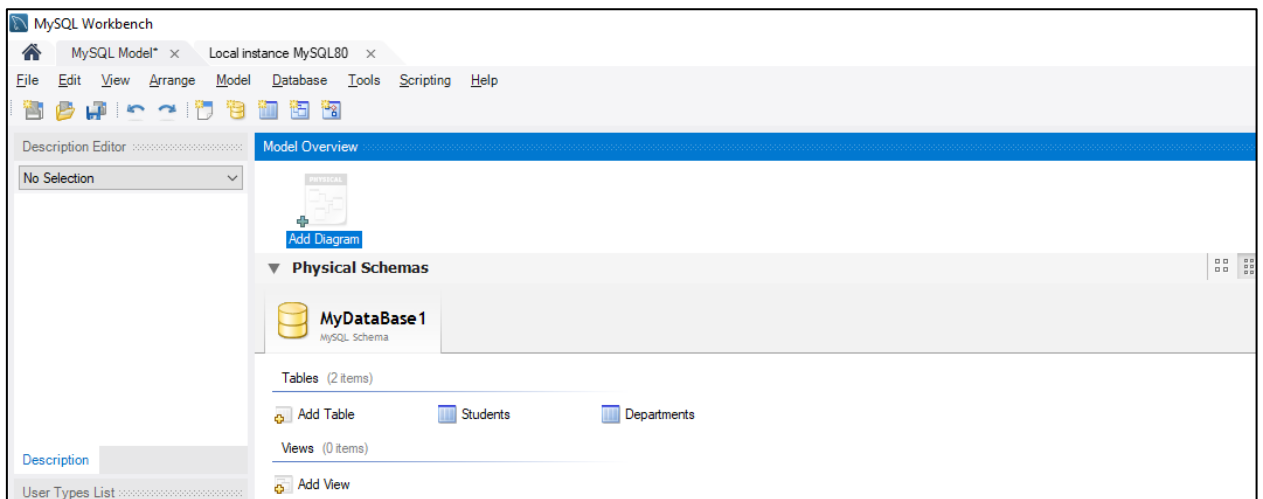


В нижнем окне output должны отобразиться все “зеленые галочки”.

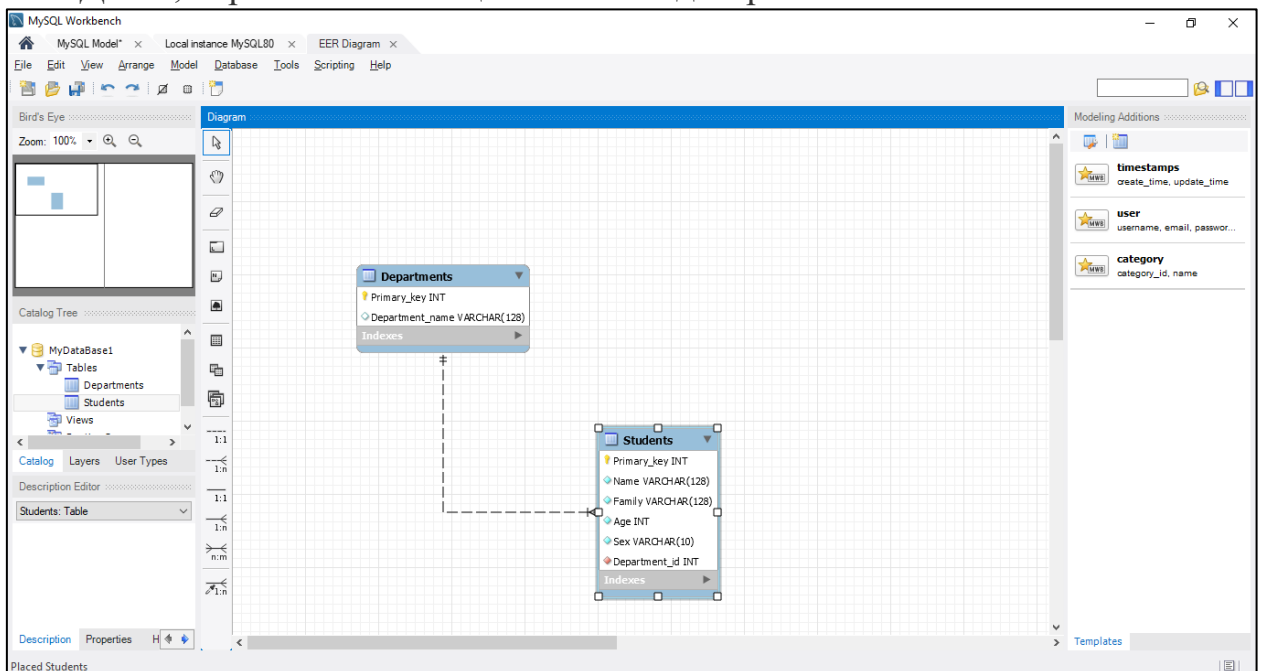
Если нажмете Refresh в контекстном меню в списке баз данных, то увидите созданную базу **MyDataBase1**.

## 6. Построим ER диаграмму.

Возвращаемся на вкладку модели и жмем на Add Diagramm



Далее, переносим таблицы в область диаграммы.



Мы создали связь один ко многим. На одном факультете могут учиться несколько студентов. Обратите внимание, связь возле таблицы Students расщепляется – это означает “ко многим”.

### 7. Заполнение созданных таблиц

Для внесения записей на каждой из таблиц перейти во вкладку **Inserts**.

Departments - Table

Table Name: Departments Schema: MyDataBase1

Filter Rows: Edit: Export/Import: Wrap Cell Content: Apply changes:

Primary_key	Department_name
1	Physics
2	Mathematics
3	History
4	Philosophy
5	Art
NULL	NULL

Columns Indexes Foreign Keys Triggers Partitioning Options Inserts Privileges

Для таблицы **Departments** делаем 5 факультетов – Physics, Mathematics, History, Philosophy, Art.

Students - Table

Table Name: Students Schema: MyDataBase1

Filter Rows: Edit: Export/Import: Wrap Cell Content: Apply changes:

Primary_key	Name	Family	Age	Sex	Department_id
1	Stanislav	Panteleev	32	Male	1
2	Alexandr	Yakovlev	43	Male	2
3	Natalia	Manina	23	Female	5
4	Oleg	Petrov	29	Male	4
5	Anna	Kononova	23	Female	3
6	Anton	Shuplin	45	Male	2
7	Andrei	Kirovtes	39	Male	1
8	Yaroslav	Mishin	28	Male	2
9	John	Travolta	23	Male	5
10	Vera	Brezneva	12	Female	4
NULL	NULL	NULL	NULL	NULL	NULL

Columns Indexes Foreign Keys Triggers Partitioning Options Inserts Privileges

Для таблицы **Students** 10 произвольных записей с соблюдением типа данных. После ввода записей, нажать “Apply Updates”

**Итак, мы создали модель, из неё через выполнение скрипта – реальную базу с таблицами. А также создали диаграмму ER.**

### Самостоятельная работа:

При выполнении каждого задания сохранять команды MySQL в отдельные файлы.

Уметь объяснить SQL код, полученный при выполнении работы.

При выполнении работы использовать только латинские символы. При работе с таблицами учесть кодировку.

1) Создать новую базу данных **Education**

2) Добавить в базу данных таблицы:

1. Таблица Students (Студенты) содержит столбцы:

StudentId – числовой код, идентифицирующий студента.

Surname – фамилия студента.



Name – имя студента.

Stipend – стипендия которую получает студент.

Kurs – курс, на котором учится студент.

City – город, в котором живет студент.

Birthday – дата рождения студента.

UnivId – идентификатор университета, в котором учится студент.

2. Таблица Lecturer (Преподаватели) содержит столбцы:

LecturerId – идентификатор преподавателя.

Surname – фамилия преподавателя.

Name – имя преподавателя.

City – город, в котором живет преподаватель.

UnivId – идентификатор университета, в котором работает преподаватель.

3. Таблица Subject (Предмет обучения) содержит столбцы:

SubjId – идентификатор предмета обучения.

SubjName – наименование предмета обучения

Hour – количество часов, отводимых на изучение предмета.

Semester – семестр, в котором изучается данный предмет.

4. Таблица University (Университет) содержит столбцы:

UnivId – идентификатор университета.

UnivName – название университета.

Rating – рейтинг университета.

City – город, в котором расположен университет.

5. Таблица ExamMarks (Экзаменационные оценки) содержит столбцы:

ExamId – идентификатор экзамена.

StudentId – идентификатор студента.

SubjId – идентификатор предмета обучения.

Mark – экзаменационная оценка.

ExamDate – дата экзамена.

6. Таблица SubjLect (Учебные дисциплины преподавателей) содержит столбцы:

LecturerId – Идентификатор преподавателя.

SubjId – идентификатор предмета обучения.

3) Создать связи между таблицами.

4) Заполнить таблицы данными, при этом каждая таблица должна содержать не менее 5–6 строк.