**Лабораторная работа № 4. MySQL. Создание базы данных с помощью MySQL Workbench**

**MySQL** − это реляционная система управления базами данных, которая активно используется при обработке данных на сайтах в сети Интернет. Физическое проектирование БД заключается в создании основных объектов базы данных на языке SQL (пользователи, таблицы и др.), написании текстов создания вспомогательных объектов базы данных (представления, индексы, триггеры, роли и т. д.).

Графический клиент MySQL Workbench позволяет создавать, удалять, изменять объекты базы данных и управлять ими. Для того, чтобы начать работать в MySQL Workbench следует скачать и установить MySQL Community Server. Процесс установки описан в [Приложении 1](#Приложение1).

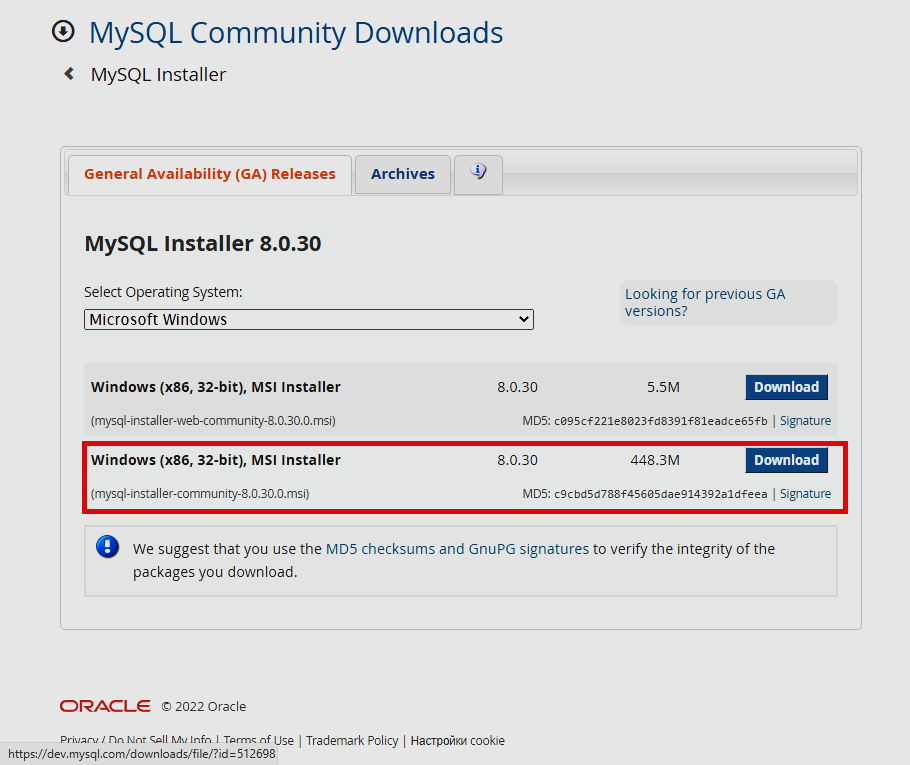
|  |  |
| --- | --- |
| **Задание** | **Краткие теоретические сведения** |
| 1. Создать с помощью языка **SQL** базу данных с именем **BaseMysql\_x**, где **x** − первые буквы своей фамилии. Таблицы взять из файла «Отчёт.docx»  Создать таблицы с использованием ограничений целостности.  Установить связи между таблицами.  Заполнить таблицы данными и вывести их. | Для ввода команд языка SQL надо в верхней левой части экрана нажать на значок **SQL** с плюсом . В центральной части программы откроется окно для ввода скрипта SQL.  Например, для создания базы данных ПРОДАЖИ следует написать следующий оператор:  **CREATE database ПРОДАЖИ;**  Чтобы скрипт выполнился на панели инструментов надо нажать на значок молнии . Пример создания и заполнения таблицы **Товары**:  **USE ПРОДАЖИ;**  **#Создание и заполнение таблицы Товары**  **CREATE TABLE Товары**  **( Наименование varchar(20) primary key,**  **Цена real,**  **Количество int**  **);**  **insert into Товары (Наименование, Цена, Количество)**  **values ('Стол', 78, 10),**  **('Диван', 400, 3),**  **('Шкаф', 450, 10),**  **('Скрепки', 5, 50),**  **('Бумага', 10, 30);**  Чтобы увидеть, как именно выглядит таблица **Товары**, надо использовать SQL-запрос, который выберет все записи из таблицы:  **select \* from Товары;**  В [приложении 1](#Приложение1) приведены скрипты для создания и заполнения базы данных ПРОДАЖИ. В сценарии создания таблицы **Заказы** определены связи между таблицами.  Если нужно наращивать значение одного из полей, то можно использовать опцию **auto\_increment**,предназначенную для автоматического увеличения поля первичного ключа таблицы на единицу.  При создании таблиц используются ограничения целостности, приведенные в таблице.     |  |  | | --- | --- | | **Ограничение целостности** | **Действие ограничения целостности** | | **data type** | Предотвращает появление значений, не соответствующих типу данных | | **not null** | Не допускает появление в столбце значений null | | **primary key** | Определяет первичный ключ | | **foreign key** | Устанавливает связь между таблицей со столбцом, имеющим свойство **foreign key** и таблицей, имеющей столбец со свойством **primary key**; | | **unique** | Предотвращает появление пустых и повторяющихся значений | | **check** | Не допускает появление значения, не удовлетворяющего логическому условию |   Для изменения строк таблицы используется оператор **UPDATE**. Например, содержимое столбца **Цена** увеличить на 1 для товара, наименование которого **Стол**, можно командой:  **UPDATE ТОВАРЫ set Цена = Цена+1 Where Наименование = 'Стол';**  Чтобы удалить одну или несколько строк используется оператор **DELETE**. В примере, приведенном ниже, удаляется строка из таблицы **Товары**, в которой в столбце **Наименование** записан текст **Стул**.  **DELETE from ТОВАРЫ Where Наименование = 'Стул';**  Для удаления таблицы служит оператор **DROP table if exists Товары;** |
| 2. Для базы данных **BaseMysql\_x** сформировать следующие запросы и проанализировать результаты: вывести информацию, содержащуюся в любых трех столбцах; вывести информацию двух столбцов и задать псевдонимы для соответствующих полей; создать запрос, используя функцию IF; придумать три запроса с использованием BETWEEN, ORDER BY, LIKE.  Сохранить запросы в sql-скрипте. | Чтобы вывести информацию о товарах, а также для каждого товара посчитать общую стоимость и вычисляемому столбцу дать имя **Общая\_стоимость**, следует написать следующий запрос:  **SELECT Наименование, Цена, Количество, Цена\*Количество**  **AS Общая\_стоимость FROM Товары;**  **Пусть требуется определить размер скидки на товар в зависимости от его общей стоимости:**  **SELECT Наименование\_товара, Цена\_продажи, Количество, Заказчик,**  **IF(Цена\_продажи\*Количество 500, 'скидка 50%',**  **IF(Цена\_продажи\*Количество 200, 'скидка 30%', 'скидка 10%'))**  **AS Скидка FROM Заказы;**  **Найти товары, наименования которых начинаются на букву «С» и цена в пределах от 30 до 100. Отсортировать результаты по цене в порядке убывания:**  **SELECT Наименование, Цена FROM Товары**  **WHERE Цена BETWEEN 30 AND 100 and Наименование LIKE 'С%' ORDER BY Цена;** |
| 3. На основе двух или трех таблиц базы данных **BaseMysql\_x** сформировать запросы с использованием *соединений* INNER JOIN и LEFT OUTER JOIN. | Оператор внутреннего соединения INNER JOIN выводит только те строки из таблиц, которые удовлетворяют записанному в запросе условию.  Левое внешнее соединение LEFT OUTER JOIN включает в результирующий набор строки, удовлетворяющие записанному в запросе условию и набор несоединенных строк таблицы, имя которой записано слева от ключевых слов LEFT OUTER JOIN.  Пример. На основании таблиц **Товары** и **Заказы** сформировать перечень товаров с ценой исходной и ценой продажи, при этом не выводить повторяющиеся строки:  **SELECT Distinct z.Наименование\_товара, t.Цена, z.Цена\_продажи**  **FROM Заказы z Inner Join Товары t**  **On z.Наименование\_товара=t.Наименование;**  В запросе использованы псевдонимы для имен таблиц. |
| 4. На основе таблиц базы данных **BaseMysql\_x** сформировать запросы с использованием *подзапросов*, *агрегатных* функций и секции GROUP BY. | Подзапрос – это SELECT-запрос, который выполняется в рамках другого запроса.  Пусть требуется определить товары и даты их поставки для фирм, которые располагаются в Минске.  **SELECT z.Наименование\_товара, z.Дата\_поставки, zk.Наименование\_фирмы**  **FROM Заказы z Inner Join Заказчики zk**  **On z.Заказчик=zk.Наименование\_фирмы**  **Where Заказчик In (Select Наименование\_фирмы**  **FROM Заказчики Where Адрес LIKE 'Минск%');**  Основное назначение группировки с помощью секции GROUP BY – разбиение множества строк на группы в соответствии со значениями в заданных столбцах, а также выполнение вычислений над группами строк с помощью наиболее часто используемых функций:  **− avg** (вычисление среднего значения);  **− count** (вычисление количества строк);  **− max** (вычисление максимального значения);  **− min** (вычисление минимального значения);  **− sum** (вычисление суммы значений).  При использовании секции **GROUP BY** в SELECT-списке допускается указывать только те столбцы, по которым осуществляется группировка.  Пример. Определить общую сумму и количество заказанных товаров для каждой фирмы.  **SELECT zk.Наименование\_фирмы,**  **sum(Цена\_продажи) Сумма,**  **count(\*) 'Количество заказанных товаров'**  **FROM Заказы z Inner Join Заказчики zk**  **On zk.Наименование\_фирмы = z.Заказчик**  **GROUP BY zk.Наименование\_фирмы;** |

**Приложение 1**

**Установка сервера MySQL**

База данных MySQL обладает кроссплатформенностью, имеются дистрибутивы под самые различные операционные системы, в том числе наиболее популярные версии Linux, Windows, MacOS.

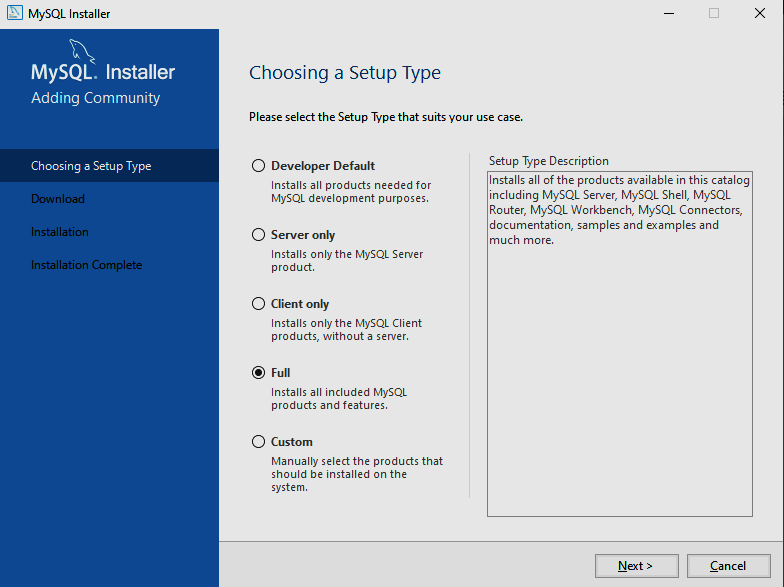
Скачать MySQL в редакции Community для Windows можно с сайта [**https://dev.mysql.com/downloads/installer**](https://dev.mysql.com/downloads/installer)**.**



1. На странице следует выбрать либо онлайн-загрузчик (если есть доступ к сети Интернет), либо полный пакет инсталлятора. Пусть выбран второй вариант.

В следующем окне следует нажать на ссылку **No thanks, just start my download**. В результате должен загрузится файл **mysql-installer-community-8.0.18.0.msi**.

2. После успешного скачивания загрузочного файла, его нужно **запустить**. Для начинающих пользователей целесообразно выбрать полную установку **Full** и нажать **Next**.



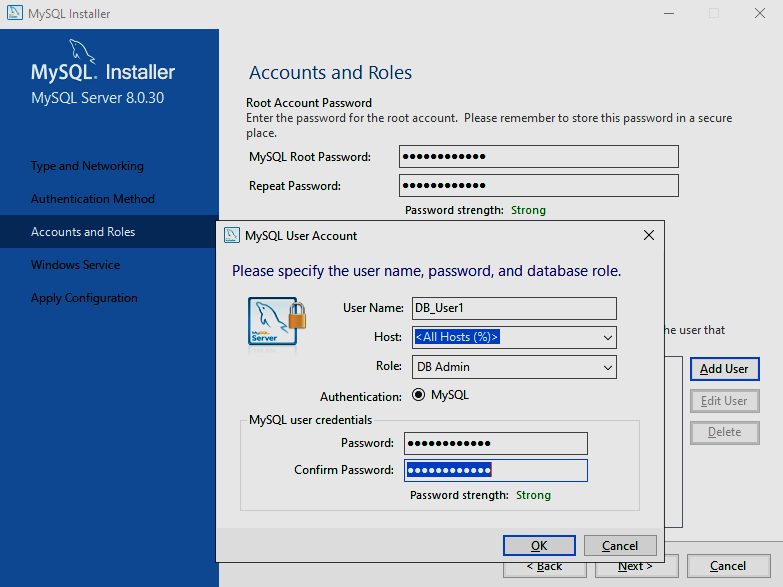
Инсталлятор предложит указать место установки компонентов, можно оставить предлагаемые по умолчанию папки, нажать **Next**, на вопрос ответить **Yes**, вновь нажать **Next**, на вопрос ответить **Yes**.

Отобразится список устанавливаемых компонентов. Для установки всех компонентов следует нажать кнопку **Execute**. После того, как компоненты будут установлены, надо нажать кнопку **Next**.

3. MySQL Server установлен, следует произвести настройку его конфигурации. При этом будет предложено установить ряд конфигурационных настроек сервера MySQL. По умолчанию для подключения будет использоваться протокол TCP/IP и порт 3306, которые следует оставить без изменения и нажать кнопку **Next**.

Если рядом с портом возникает красная (желтая) индикация (), то нужно найти и изменить номер порта с 3306 на 3307, также изменить ключ 33060 на 33070.

4. На следующем шаге будет предложено установить метод аутентификации, настройки которого также можно оставить без изменения и нажать кнопку **Next**. В окне аутентификации надо указать какой-нибудь пароль (и запомнить его).



Следующий набор конфигураций, который также надо оставить без изменений просто нажав **Next**, указывает, что сервер будет запускаться в качестве службы Windows при запуске операционной системы. И в появившемся окне необходимо применить все ранее установленные конфигурационные настройки, нажав на кнопку **Execute**.

|  |  |  |
| --- | --- | --- |
|  |  |  |

После применения конфигурационных настроек сервер MySQL будет полностью установлен и сконфигурирован. Следует нажать кнопку **Finish**.

5. Затем опять отобразится окно с перечнем продуктов, готовых к конфигурации. Надо нажать кнопку **Next**. В следующем окне все настройки нужно оставить без изменения и нажать **Finish**. Отобразится окно с перечнем продуктов, готовых к конфигурации. Следует нажать кнопку **Next**.

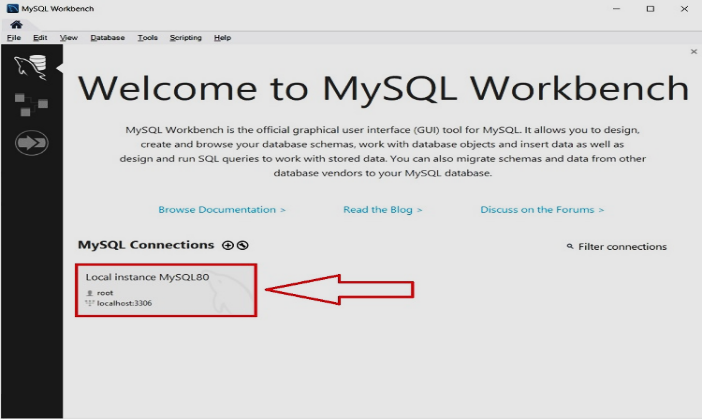
6. Далее можно установить тестовые данные на MySQL Server, например, для обучения. Чтобы это сделать, надо ввести пароль и нажать **Check** (возможно повторить нажатие этой кнопки), затем нажать **Next**.

Чтобы применить все параметры и начать извлечение тестовых данных на MySQL Server нужно на следующей странице нажать **Execute**. Процесс будет завершен, когда галочки будут проставлены и отобразится сообщение «Successful». Надо нажать **Finish**.

|  |  |  |
| --- | --- | --- |
| Image16 | Image11 | Image8 |

7. На следующей странице нужно снять отметки с полей **Start MySQL Workbench after setup** и **Start MySQL Shell after setup**. Эти поля позволяют запустить графический и консольный клиенты для управления сервером MySQL. Нажать кнопку **Finish**. MySQL полностью установлен, сконфигурирован и запущен.

## Подключение к серверу с помощью MySQL Workbench

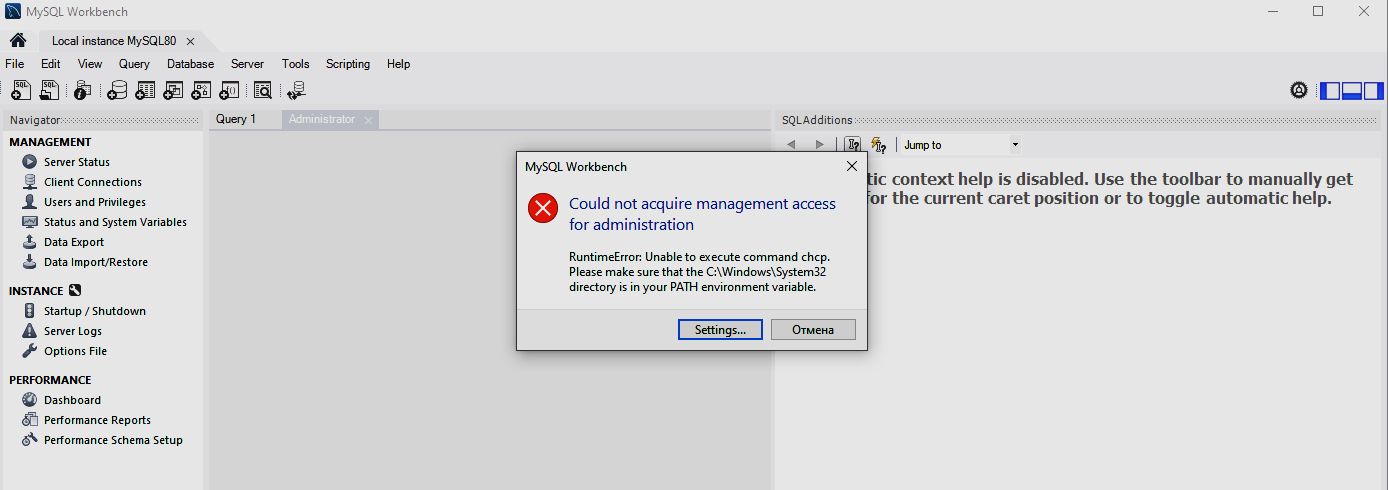


При установке сервера MySQL также устанавливается консольный клиент для работы с базами данных. В базовый комплект входит такой инструмент, как **MySQL Workbench**, с помощью которого можно создавать, удалять, изменять базы данных и управлять ими.

При запуске MySQL Workbench на стартовой странице будет отражено подключение к локальному серверу, которое будет осуществляться от имени пользователя root. Следует нажать на прямоугольник.

Затем нужно ввести пароль, чтобы сохранить его можно поставить галочку у пункта «Save password in vault» и нажать ОК. Произойдет подключение к серверу.

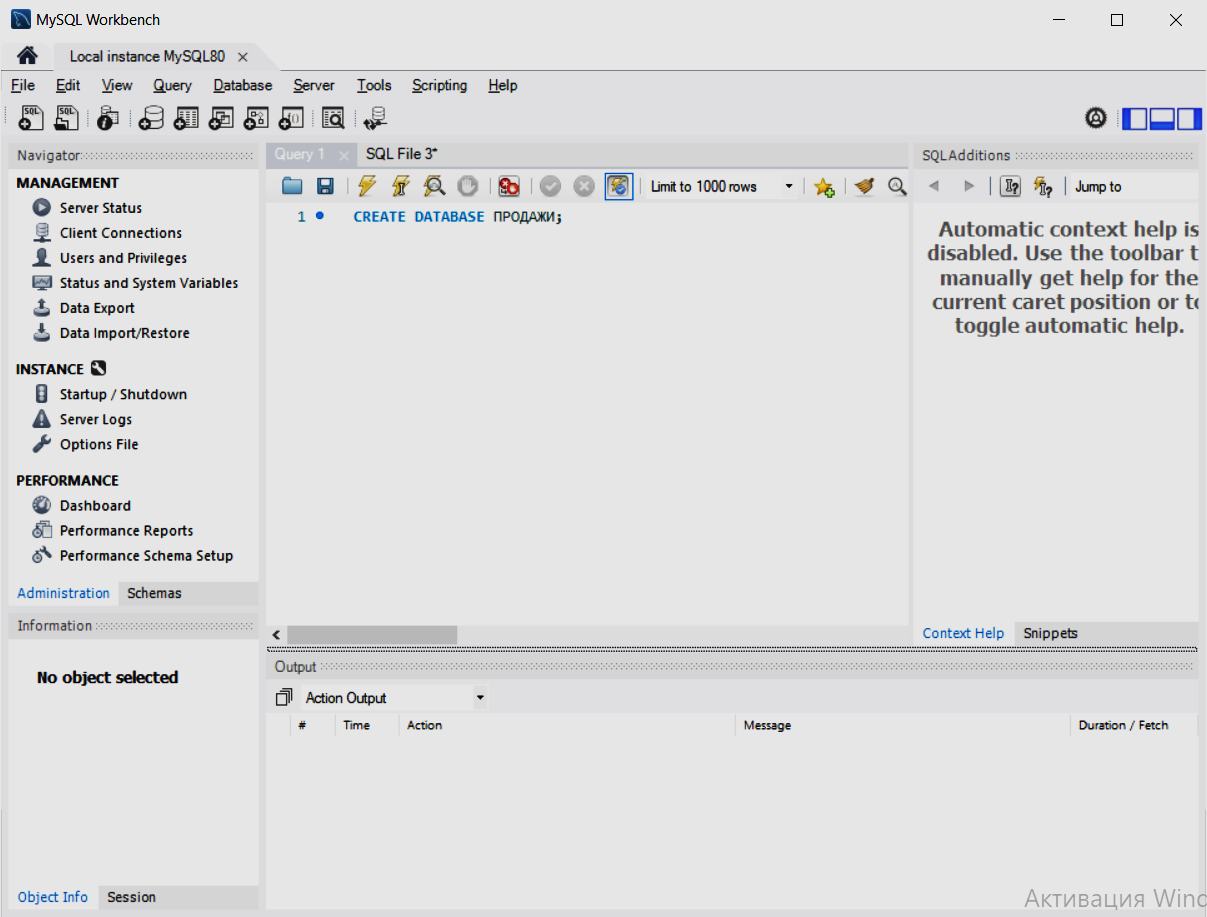
**Примечание**. При установке MySQL может не получаться проверить статус сервера и выдается ошибка, приведенная ниже:



Следует выполнить следующие шаги:

* на диске С в папке Windows скопировать файл c:\Windows\SysWOW64\chcp.com;
* поместить скопированный файл в папку с программной оболочкой сервера C:\Program Files\MySQL\MySQL Workbench 8.0;
* в папке Windows скопировать файл c:\Windows\SysWOW64\ulib.dll;
* поместить скопированный файл в папку с программной оболочкой сервера C:\Program Files\MySQL\MySQL Workbench 8.0.

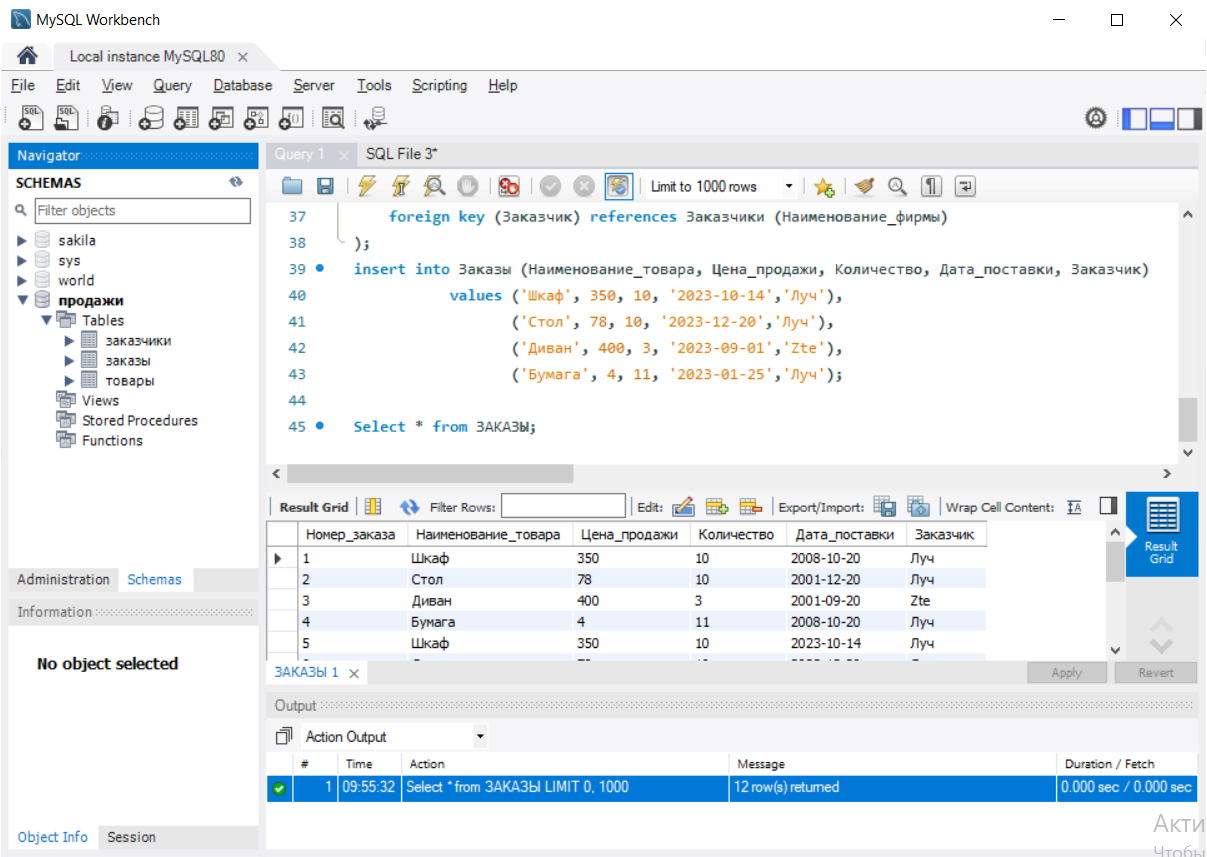
Чтобы *скрыть* ненужные окна можно использовать специальные значки в правой верхней части среды MySQL Workbench.



В окне навигатора можно перейти на вкладку **Schemas** и посмотреть на схему данных, где можно увидеть, какие объекты находятся в базе.

Для *создания базы данных* надо в верхней левой части экрана нажать на значок **SQL** с плюсом под пунктом меню **File**. В центральной части программы откроется окно для ввода скрипта SQL.

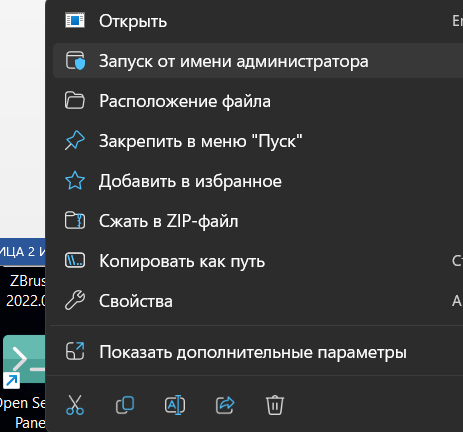
Чтобы сценарий *выполнился* следует на панели инструментов нажать на значок *молнии*, который располагается над скриптом.



В окне представлен скрипт для заполнения таблицы и выполнения SELECT-запроса на выборку данных из таблицы **Заказы**.

Если нужно внести *изменения* в структуру таблиц, то после выполнения соответствующего сценария следует в контекстном меню имени базы выполнить команду **Refresh all**.

**Освобождение портов от Open Server для MySql или наоборот**

1. В **Диспетчере задач** выбрать пункт **Процессы** или **Сведения.**

2. В открывшемся окне найти **Open Server** или **mysqld**, в контекстном меню снять задачу.

3. Перезагрузить устройство и заново запустить MySQL или Open Server от ***имени администратора*** (в зависимости от того, для чего очищался порт: если очищался от mysqld, то должен заработать Open Server, если очищался Open Server, то должен корректно заработать MySQL).

Видео с подробным описанием:

https://www.youtube.com/watch?v=8gDuqi60MLA

**Сценарии создания таблиц базы данных ПРОДАЖИ**

|  |  |
| --- | --- |
| CREATE database ПРОДАЖИ;  USE ПРОДАЖИ;  #Создание и заполнение таблицы Товары  CREATE TABLE Товары  ( Наименование varchar(20) primary key,  Цена real,  Количество int  );  insert into Товары (Наименование, Цена, Количество)  values ('Стол', 78, 10),  ('Диван', 400, 3),  ('Шкаф', 450, 10),  ('Скрепки', 5, 50),  ('Бумага', 10, 30);  #Создание и заполнение таблицы Заказчики  CREATE TABLE Заказчики  ( Наименование\_фирмы varchar(20) primary key,  Адрес varchar(50),  Расчетный\_счет varchar(20)  ); | insert into Заказчики (Наименование\_фирмы, Адрес, Расчетный\_счет)  values ('Луч', 'Минск, ул. Короля, 3', '123456'),  ('Белвест', 'Брест, ул. Герцена, 20', '342346'),  ('Радуга', 'Минск, ул. Скорины, 19', '983456'),  ('Zte', 'Смолевичи, ул. Кирова, 32', '883456');  #Создание и заполнение таблицы Заказы  CREATE TABLE Заказы  ( Номер\_заказа int primary key auto\_increment,  Наименование\_товара varchar(20),  Цена\_продажи real,  Количество int,  Дата\_поставки date,  Заказчик varchar(20),  foreign key (Наименование\_товара) references Товары (Наименование),  foreign key (Заказчик) references Заказчики (Наименование\_фирмы)  );  insert into Заказы (Наименование\_товара, Цена\_продажи, Количество, Дата\_поставки, Заказчик)  values ('Шкаф', 480, 10, '2023-10-14','Луч'),  ('Стол', 85, 10, '2023-12-20','Луч'),  ('Диван', 420, 3, '2023-09-01','Zte'),  ('Бумага', 14, 11, '2023-01-25','Луч'); |