

Тема 12. Введение в базы данных, типы баз данных.

Работа с MySQL, создание БД.

Цель занятия:

**Ознакомиться с основными
понятиями и видами баз данных.**

Учебные вопросы:

**1. Основные
понятия**

2. Типы СУБД

3. Типы БД

4. Реляционные БД

**5. Structured Query
Language (SQL)**

**6. Установка
MySQL.**

**7. DDL запросы.
Создание ролей и
БД через
консоль.**

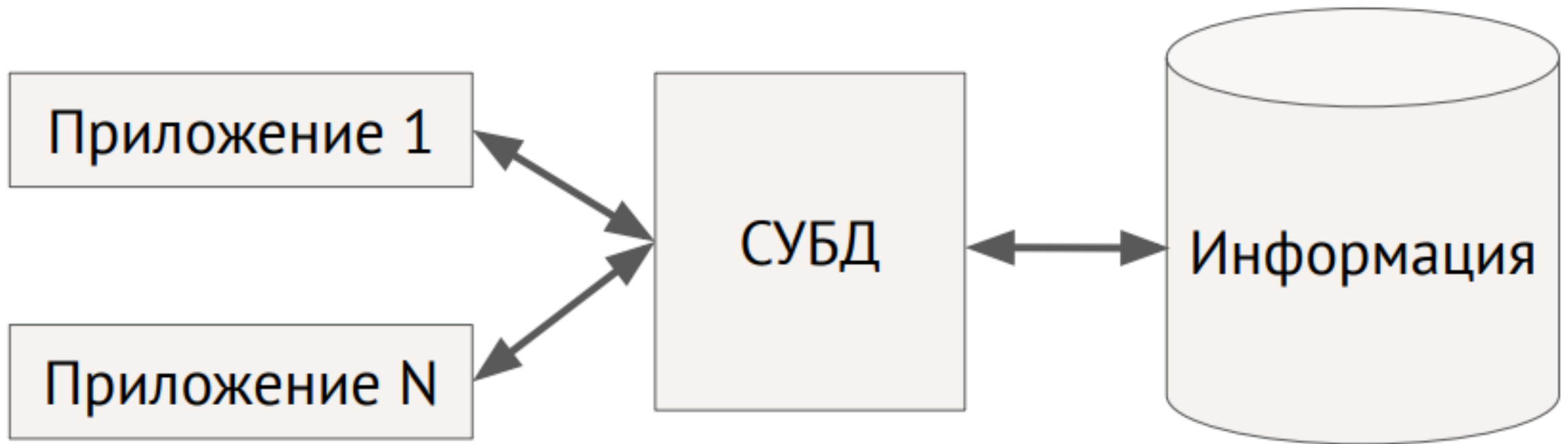
1. Основные понятия.

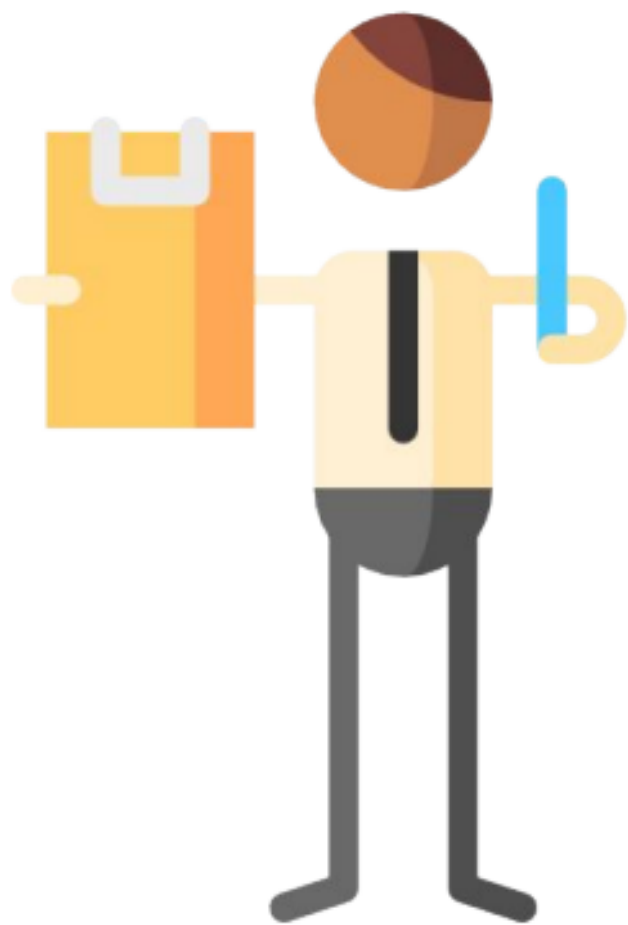
База данных – это организованная совокупность данных, хранимая в соответствии со схемой данных и предназначенная для эффективного поиска, обновления, управления и использования этих данных.

База данных – это набор взаимосвязанных данных и правила хранения этих данных.

Система Управления Базами Данных (СУБД)

Система управления базами данных (СУБД) - это программное обеспечение, которое используется для управления базами данных. Она предоставляет средства для создания, организации, хранения, обновления и извлечения данных из базы данных.





Архив с данными – это информация, которую хотим хранить.

Архивариус – СУБД, через него идут все манипуляции с данными, будь то удаление, добавление или получение.

Посетитель – программа, которой необходимо производить операции с данными.

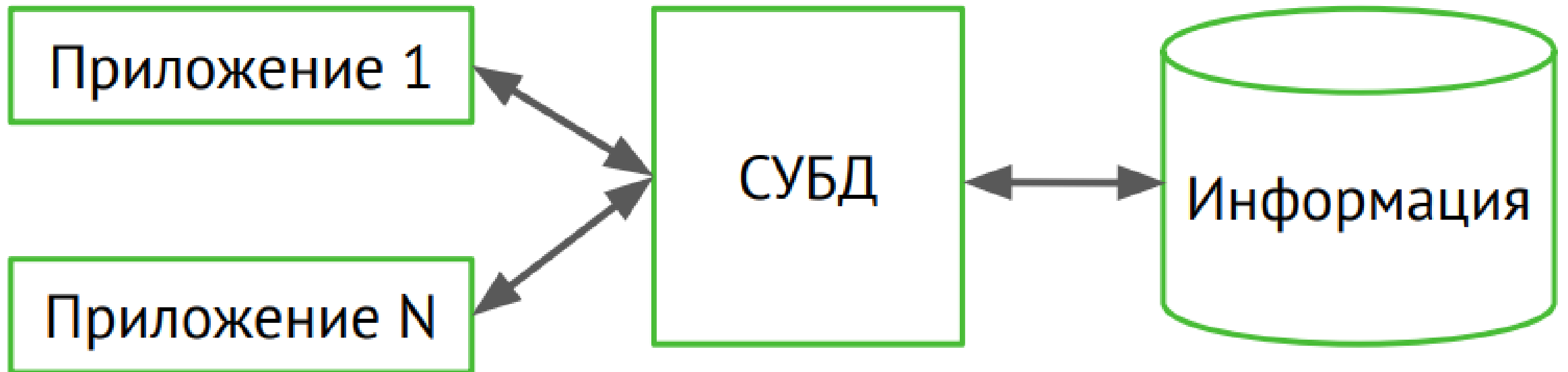
Зачем нужны базы данных?

- **Хранение информации:** Базы данных позволяют хранить огромные объемы информации в структурированном виде, что облегчает ее поиск и управление.
- **Обеспечение доступа:** Множество пользователей могут одновременно получать доступ к данным и вносить изменения.
- **Управление данными:** Базы данных предоставляют инструменты для добавления, удаления, обновления и поиска данных.
- **Сохранение целостности:** Благодаря механизмам контроля целостности, данные в базе данных остаются согласованными и точными.

2. Типы СУБД.

Типы СУБД

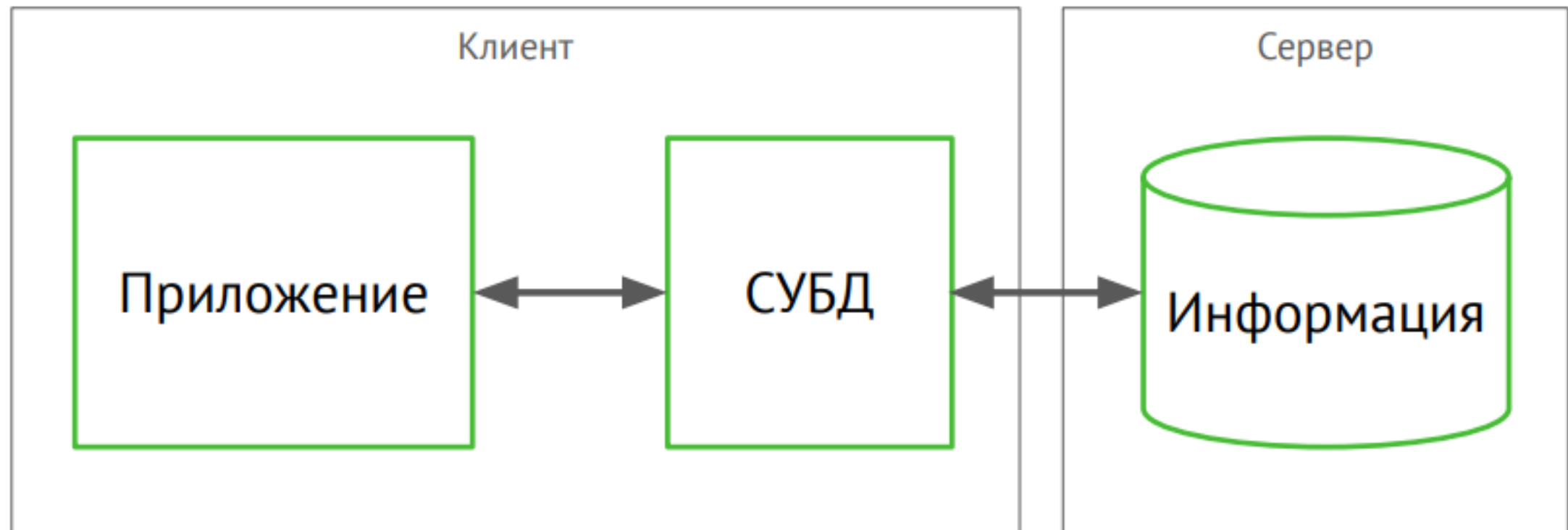
- файл-серверные
- клиент-серверные
- встраиваемые



Файл-серверные СУБД

Файлы с информацией хранятся на сервере, а СУБД на клиенте.

Программа: Microsoft Access.



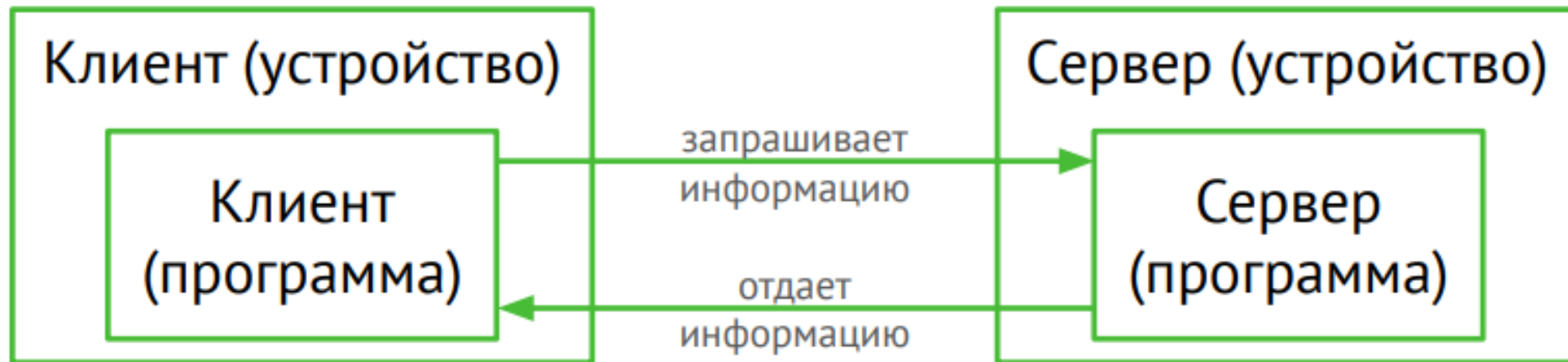
Клиент и сервер

Клиент:

1. программа, которая хочет получить информацию;
2. физическое устройство, на котором работает программа-клиент.

Сервер:

1. специальная программа, которая дает информацию;
2. физическое устройство, на котором запущена программа-сервер.

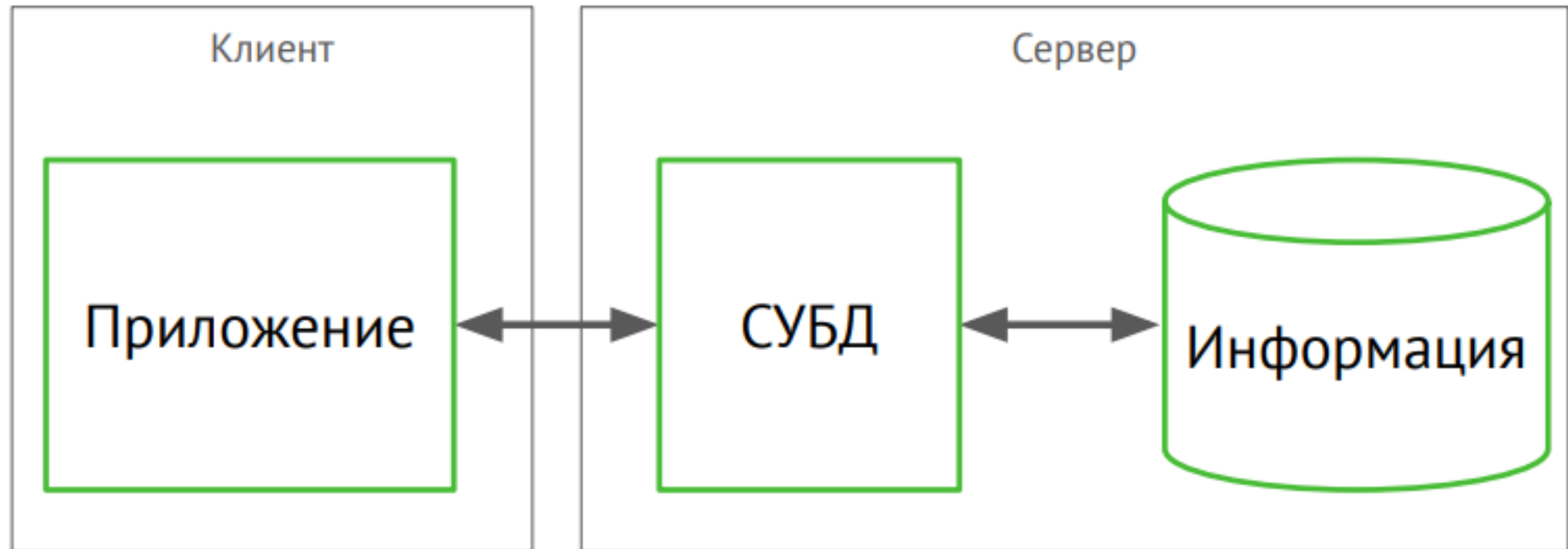


Обычно эти программы расположены на разных вычислительных машинах и взаимодействуют между собой по различным протоколам, но они могут быть расположены и на одной машине.

Клиент-серверные СУБД

И файлы с информацией и СУБД находятся на сервере, а клиент обращается за информацией через легковесную вспомогательную программу.

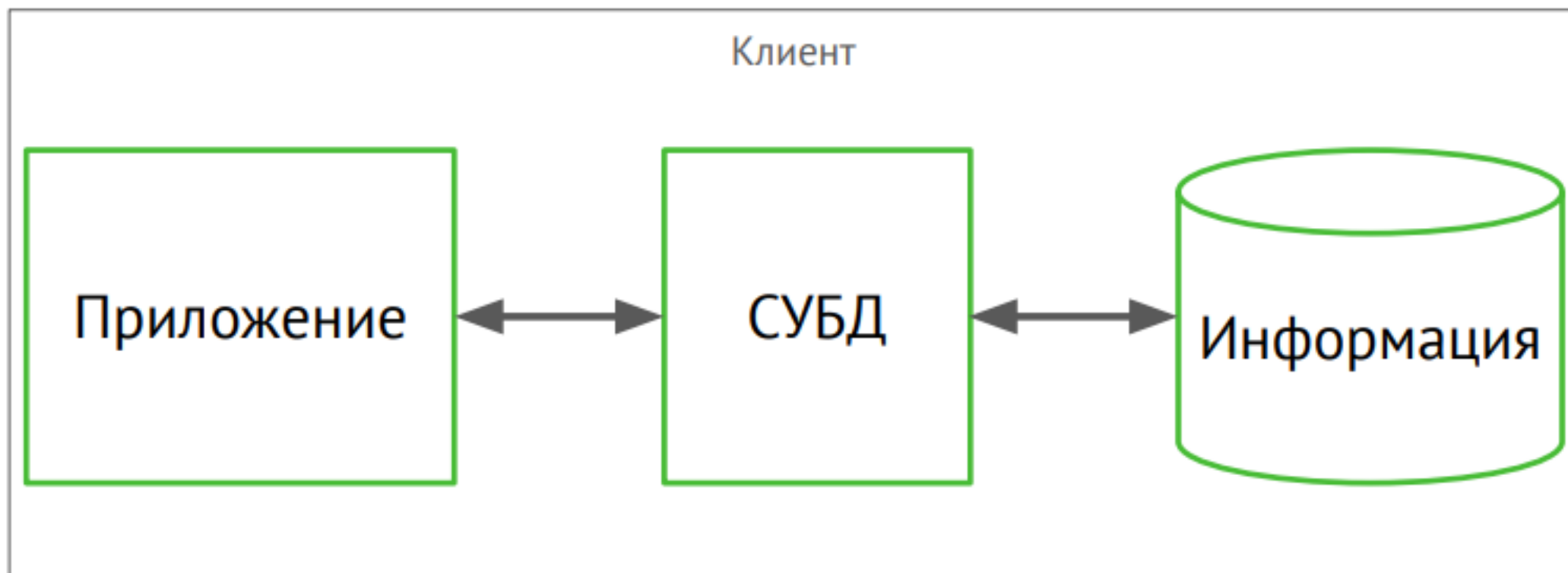
Программы: MySQL, PostgreSQL, Microsoft SQL, Oracle, MongoDB, Cassandra.



Встраиваемые СУБД

Файлы и СУБД хранятся на клиенте.

Программа: SQLite.



Какими плюсами и минусами обладает каждый тип СУБД?

Тип СУБД	Плюсы	Минусы
Файл-серверные	<ul style="list-style-type: none">Сервер может быть обычным файловым хранилищемЛегко переносить базу	<ul style="list-style-type: none">Плохо параллелятся действия от разных клиентовТребуется установка СУБД на каждом клиенте
Клиент-серверные	<ul style="list-style-type: none">На клиенте не надо устанавливать СУБДХорошо параллелятся действия от разных клиентов	<ul style="list-style-type: none">Сервер должен быть достаточно производительным => дорого
Встраиваемые	<ul style="list-style-type: none">Не надо ничего устанавливать	<ul style="list-style-type: none">Подходит только для локального хранения

3. Типы баз данных.

- **Реляционные базы данных:**

Самый распространенный тип.

Данные организованы в таблицы, состоящие из строк и столбцов.

Используют язык SQL для выполнения запросов.

Примеры СУБД: MySQL, PostgreSQL, Oracle.

- **NoSQL базы данных:**

Не имеют жесткой структуры.

Подходят для хранения больших объемов неструктурированных данных.

Типы:

- Ключ-значение (Redis)
- Документоориентированные (MongoDB)
- Колоночные (Cassandra)
- Графовые (Neo4j)

- **Другие типы:**

Объектно-реляционные базы данных

Иерархические базы данных

Сетевые базы данных

Типы баз данных

Реляционные – это БД, в которых информация строго структурирована и связана с другой информацией жёсткими правилами.

Пример:

- Microsoft Access
- SQLite
- PostgreSQL
- MySQL
- Microsoft SQL

Нереляционные (NoSQL) – это БД, в которых нет жёстких ограничений ни по структуре, ни по связи между информацией.

Пример:

- Redis
- MongoDB
- Cassandra

4. Реляционные БД.

Реляционная база данных — это тип базы данных, которая хранит и организует данные в виде таблиц (отношений), состоящих из строк и столбцов.

Каждая строка представляет собой запись (например, информацию о конкретном объекте), а каждый столбец — атрибут (характеристику этого объекта).

Таблицы могут быть связаны между собой с помощью ключей, что позволяет эффективно управлять и анализировать данные.

Проще говоря:

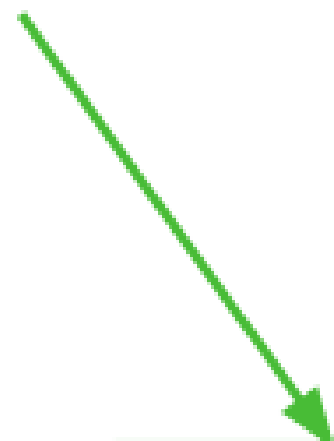
Реляционная база данных - совокупность взаимосвязанных таблиц.

Основные понятия реляционных баз данных:

Таблица: Основной элемент реляционной базы данных, состоящая из строк и столбцов.

- **Запись:** Отдельная **строка** в таблице, представляющая один экземпляр данных.
- **Поле:** Отдельный **столбец** в таблице, содержащий определенный тип данных.
- **Ключ:** Уникальный идентификатор записи.

Отношение – таблица.



id	name	gpa
1	Егор	4.82
2	Егор	4.11
3	Егор	3.88

Пример отношения «Успеваемость студентов»

Атрибут (или поле) – столбец.

Запись (или кортеж) – строка.

Атрибут



id	name	gpa
1	Егор	4.82
2	Егор	4.11
3	Егор	3.88

Кортеж



Пример отношения «Успеваемость студентов»

Преимущества реляционных баз данных

- **Структурированность:** Данные хранятся в четко определенной структуре, что облегчает их поиск и управление.
- **Целостность:** Механизмы обеспечения целостности данных гарантируют, что информация остается точной и согласованной.
- **Гибкость:** Реляционные базы данных легко адаптируются к изменениям в структуре данных.
- **Мощность выразительных средств:** Язык SQL позволяет выполнять сложные запросы к данным.
- **Широкая поддержка:** Существует множество СУБД, поддерживающих реляционную модель данных.

Особенности реляционных баз данных:

Таблицы: Данные в реляционных базах данных хранятся в виде таблиц, которые состоят из строк и столбцов. Каждая таблица представляет собой отдельную сущность, а каждая строка в таблице представляет отдельную запись.

Схема данных: РБД используют предварительно определенную схему данных, которая определяет структуру таблиц и связи между ними. Схема данных определяет типы данных, ограничения целостности, связи и другие атрибуты таблиц.

Отношения: Реляционные базы данных поддерживают связи между таблицами с помощью ключей. Связи могут быть один-к-одному, один-ко-многим или многие-ко-многим. Это позволяет эффективно организовывать данные и выполнять операции объединения, фильтрации и связи данных.

SQL: Реляционные базы данных используют язык структурированных запросов SQL (**Structured Query Language**) для выполнения операций доступа к данным, таких как выборка, вставка, обновление и удаление данных. SQL предоставляет мощные возможности для манипуляции данными и выполнения сложных запросов.

ACID-свойства: РБД обеспечивают ACID-свойства, которые гарантируют надежность и целостность данных. ACID означает атомарность (atomicity), согласованность (consistency), изолированность (isolation) и долговечность (durability) операций в базе данных.

Пример базы данных

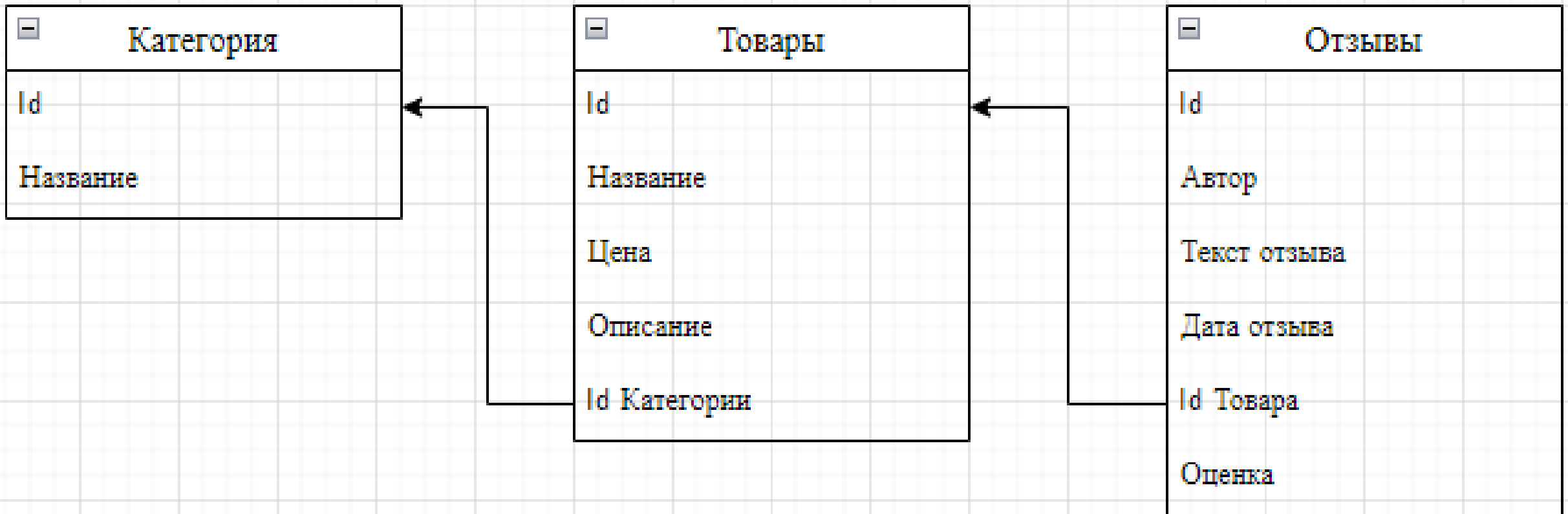


Схема БД. Таблицы и данные

Таблица 1

Атрибут 1	Атрибут 2	Атрибут 3
1	Егор	4.25
2	Дима	3.82
3	Миша	4.15

Таким способом описываются конкретные данные в таблице.

Таблица 1
Атрибут 1
Атрибут 2
Атрибут 3

Таким способом описываются таблицы и их атрибуты: информацию какого вида таблица содержит.

Писатель		
Id	Name	Birthday
1	Лев Толстой	1828
2	Александр Солженицын	1918
3	Иван Тургенев	1818
4	Антон Чехов	1860
5	Иван Бунин	1870
6	Михаил Булгаков	1891
7	Николай Гоголь	1809
8	Александр Пушкин	1799
9	Федор Достоевский	1821
10	Михаил Лермонтов	1814

Ссылка			
Id	AuthorId	Start	End
1	9	1850	1854
2	2	1945	1953
3	8	1824	1826
4	10	1837	1837
5	10	1840	1841

Произведение				
Id	AuthorId	Name	Start	End
1	10	Герой нашего времени	1838	1840
2	1	Война и мир	1863	1873
3	8	Капитанская дочка	1836	1836
4	10	Смерть поэта	1837	1837
5	2	Архипелаг ГУЛАГ	1958	1968
6	10	Бородино	1837	1837
7	3	Отцы и дети	1860	1861
8	4	Три сестры	1900	1901
9	5	Косцы	1921	1921
10	6	Белая гвардия	1922	1924
11	7	Мертвые души	1835	1835
12	9	Идиот	1867	1869
13	8	Моцарт и Сальери	1830	1830

Сколько произведений написал Лермонтов?

Сколько авторов писали свои произведения с 1830 по 1840 годы?

Пример базы данных интернет-магазина

Интернет-Магазин.

- Есть товары.
- Есть категории товаров.
- Каждый товар принадлежит строго одной категории.
- К товарам могут написать отзывы (к одному товару можно написать множество отзывов).
- Необходимо хранить информацию о категориях, товарах и отзывах.

Пример решения:



Категории	
id	Название
1	Футболки
2	Шорты
3	Обувь
4	Спортивные костюмы

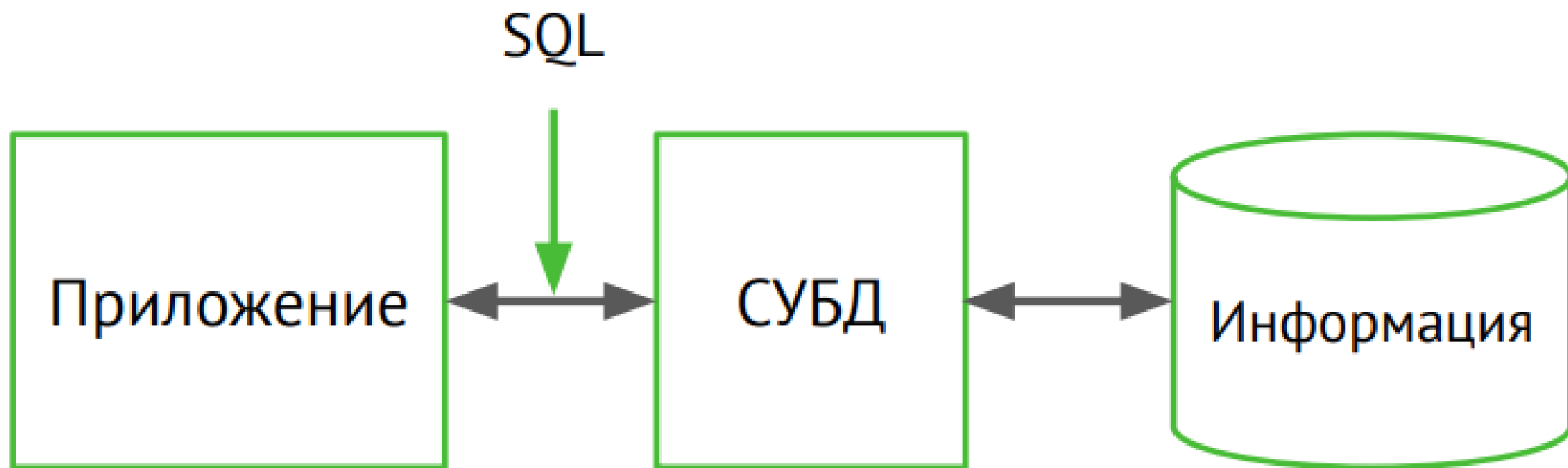
Товары			
id	Название	Цена	id категории
1	Костюм спор. Puma	10000	4
2	Кроссовки Adidas	20000	3
3	Футболка Reebok	5000	1
4	Футболка Noname	3000	1

Отзывы				
id	Оценка	Текст	Автор	id Товара
1	5	"Супер"	Егор	1
2	4	"Норм"	Степан	3
3	3	"Так себе"	Алина	4
4	5	"Отлично"	Никтиа	1

5. Structured Query Language (SQL)

Structured Query Language (SQL)

– язык для извлечения/изменения/удаления/добавления данных. Данный язык понимает СУБД, которая и производит соответствующие операции с данными.



Structured Query Language (SQL)

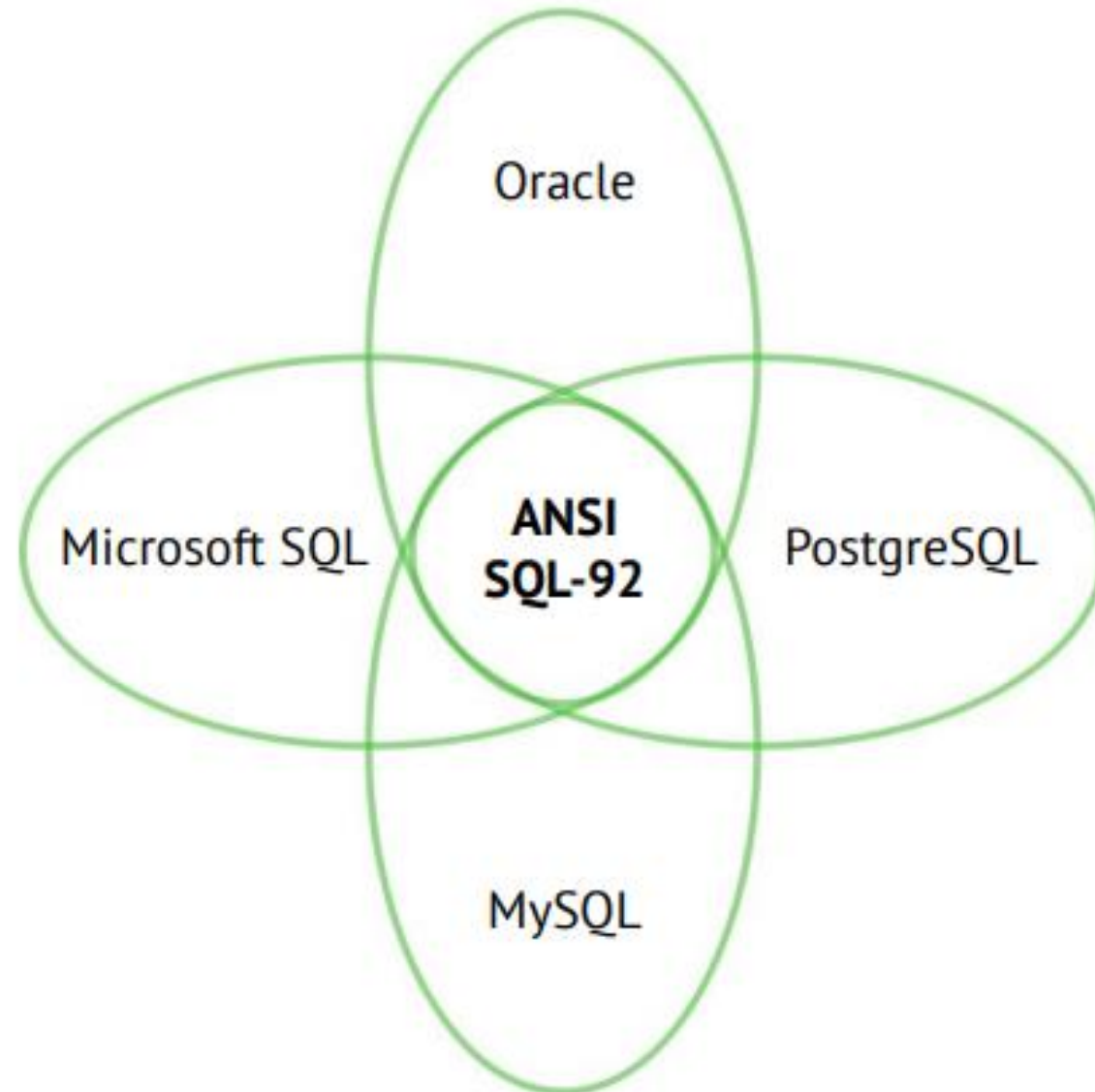
Пример запроса

```
SELECT * FROM student;
```

Результат выполнения

	123 id ↑↓	abc name ↑↓	123 gpa ↑↓	🕒 birth ↑↓
1	12	Карина	4.7	2000-09-12 00:00:00
2	13	Игорь	3.8	2000-01-26 00:00:00
3	15	Илья	4.2	1999-05-08 00:00:00
4	17	Вова	[NULL]	1999-04-14 00:00:00

Стандартизация SQL. ANSI SQL-92



Типы запросов в SQL

DDL (Data Definition Language) в SQL представляет набор команд, используемых для **определения и изменения структуры** базы данных. Они позволяют создавать, изменять и удалять таблицы, индексы, представления и другие объекты базы данных.

DML (Data Manipulation Language) в SQL - это набор команд, используемых для **манипулирования данными** в базе данных. Они позволяют вставлять, обновлять, удалять и извлекать данные из таблиц.

TCL (Transaction Control Language) в SQL - это набор команд, используемых для **управления транзакциями** в базе данных. Транзакция представляет собой логическую операцию или набор операций, которые должны быть выполнены как единое целое, либо все операции должны быть отменены.

DCL (Data Control Language) в SQL - это набор команд, используемых для **управления правами доступа** и безопасностью в базе данных. Они позволяют управлять разрешениями пользователей на выполнение определенных операций с данными.

Типы запросов в SQL

- **DDL** (Data Definition Language) – CREATE, ALTER, DROP
- **DML** (Data Manipulation Language) – SELECT, INSERT, UPDATE, DELETE
- **TCL** (Transaction Control Language) – COMMIT, ROLLBACK, SAVEPOINT
- **DCL** (Data Control Language) – GRANT, REVOKE, DENY

6. Установка MySQL.

MySQL - это одна из самых популярных в мире **свободных реляционных** систем управления базами данных (**СУБД**).

Она широко используется для создания и управления веб-приложений, а также в других областях, где требуется надежное хранение и обработка структурированных данных.

Порядок установки MySQL:

Шаг 1: Скачивание MySQL

- 1.Перейдите на официальный сайт MySQL: [MySQL Downloads](#).
- 2.В разделе **"Select Operating System"** выберите **"Microsoft Windows"**.
- 3.Выберите **MySQL Installer for Windows** (32-bit and 64-bit) и нажмите **Download**.
- 4.Выберите версию MySQL и скачайте инсталляционный файл.

MySQL Community Downloads

MySQL Community Server

General Availability (GA) Releases

Archives



MySQL Community Server 8.4.2 LTS

Select Version:

8.4.2 LTS

Select Operating System:

Microsoft Windows

Windows (x86, 64-bit), MSI Installer

8.4.2

128.5M

Download

(mysql-8.4.2-winx64.msi)

MD5: 888dc0f177ce11ed461294ff797824c7 | Signature

⬇ MySQL Community Downloads

Login Now or Sign Up for a free account.

An Oracle Web Account provides you with the following advantages:

- Fast access to MySQL software downloads
- Download technical White Papers and Presentations
- Post messages in the MySQL Discussion Forums
- Report and track bugs in the MySQL bug system

Login »

using my Oracle Web account

Sign Up »

for an Oracle Web account

MySQL.com is using Oracle SSO for authentication. If you already have an Oracle Web account, click the Login link. Otherwise, you can signup for a free account by clicking the Sign Up link and following the instructions.

No thanks, just start my download.

Шаг 2: Установка MySQL

1. Запустите скачанный установочный файл.
2. В установщике выберите тип установки.
Рекомендуется выбрать "**Developer Default**", чтобы установить все необходимые компоненты для разработки.
3. Следуйте инструкциям мастера установки, принимая лицензионное соглашение и выбирая каталог установки.

На этапе настройки MySQL, вам нужно будет:

- Выбрать тип конфигурации сервера (по умолчанию оставьте "Standalone MySQL Server").
- Настроить порт сервера (обычно 3306).
- **Создать root-пользователя и задать пароль.**
- Выбрать метод аутентификации (рекомендуется оставить Use Strong Password Encryption).

Проверьте корректность установки:

```
C:\Users\user>mysql -u root -p
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.4.2 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SELECT VERSION();
+-----+
| VERSION() |
+-----+
| 8.4.2      |
+-----+
1 row in set (0.00 sec)

mysql>
```

Если получите ошибку: **"mysql" не является внутренней или внешней командой...**, то нужно настроить переменные окружения.

- Откройте Панель управления.
- Перейдите в раздел Система и безопасность > Система.
- В левой колонке выберите Дополнительные параметры системы.
- В окне "Свойства системы" на вкладке Дополнительно нажмите кнопку Переменные среды.
- В разделе Системные переменные найдите переменную Path и выберите ее.
- Нажмите Изменить.
- В появившемся окне нажмите Создать и добавьте путь к папке bin MySQL. Например: C:\Program Files\MySQL\MySQL Server 8.4\bin
- Нажмите **ОК** во всех окнах, чтобы сохранить изменения.

Найти параметр

Система

Дисплей

Звук

Уведомления и д

Фокусировка вни

Питание и спящи

Батарея

Память

Планшет

Свойства системы

Имя компьютера

Оборудование

Дополнительно

Защита системы

Удаленный доступ

Необходимо иметь права администратора для изменения большинства перечисленных параметров.

Быстродействие

Визуальные эффекты, использование процессора, оперативной и виртуальной памяти

Параметры...

Профили пользователей

Параметры рабочего стола, относящиеся ко входу в систему

Параметры...

Загрузка и восстановление

Загрузка и восстановление системы, отладочная информация

Параметры...

Переменные среды...

OK

Отмена

Применить

Переименовать этот ПК

Н.

Безопасность

тва

JR16D02E19

Core(TM) i7-3612QM CPU @

z 2.10 GHz

(доступно: 15,9 ГБ)

9B-6124-436F-BDF4-7512D3B2FF56

0000-00001-AA268

рядная операционная система,

ssor x64

ого монитора недоступен ввод с

ью пера и сенсорный ввод

Переменные среды

Переменные среды пользователя для user

Переменная	Значение
Path	C:\Users\user\AppData\Local\Programs\Python\Python312\Scripts...
PyCharm Community Edition	C:\Program Files\JetBrains\PyCharm Community Edition 2024.1.4\b...
TEMP	C:\Users\user\AppData\Local\Temp
TMP	C:\Users\user\AppData\Local\Temp

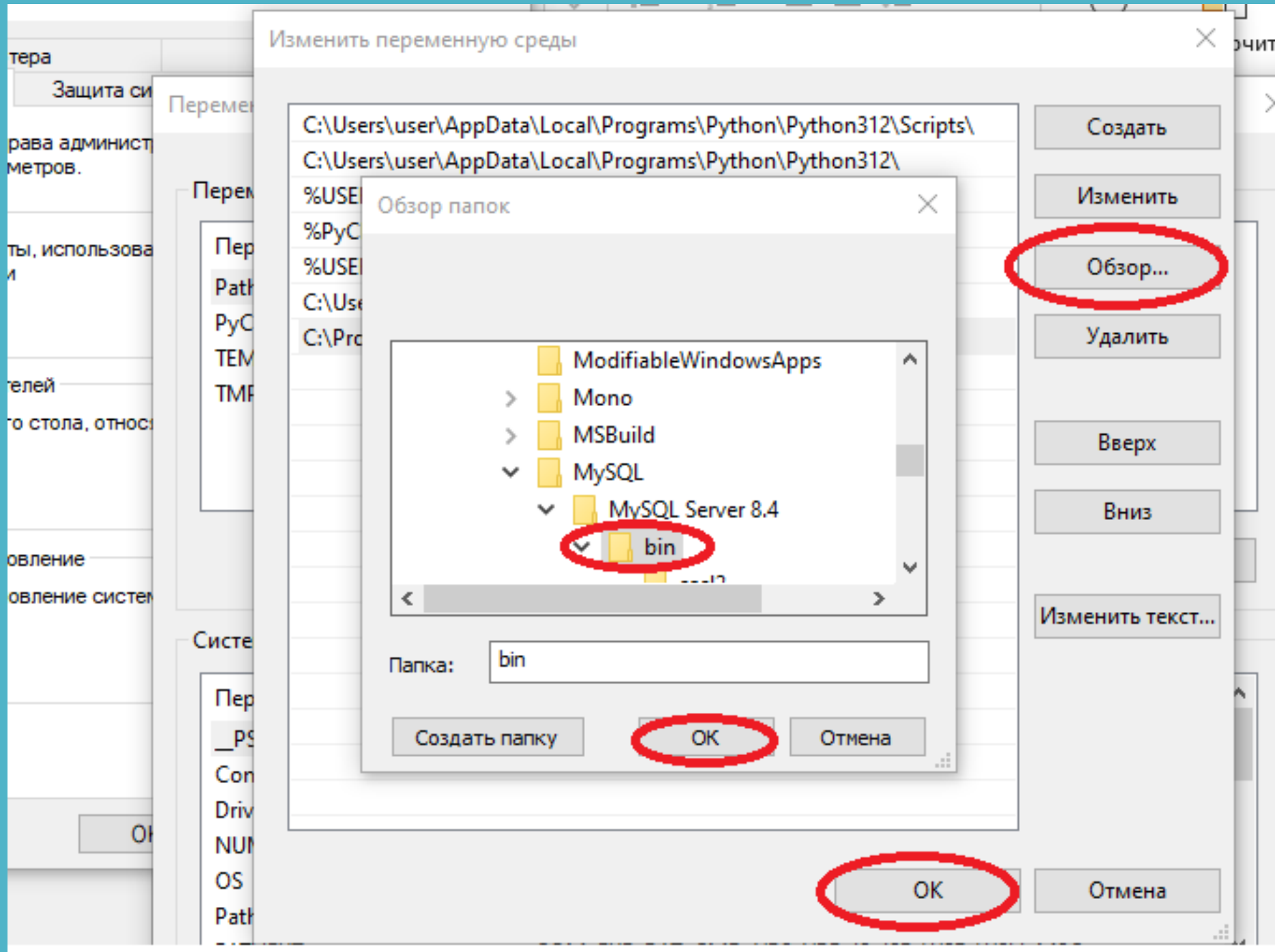
Создать..

Изменить...

Удалить

Системные переменные

Переменная	Значение
_PSLockDownPolicy	0
ComSpec	C:\Windows\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PROCESSORS	8
OS	Windows NT



Шаг 3: Установка MySQL Workbench

Скачайте установщик:

<https://dev.mysql.com/downloads/workbench/>

и установите аналогичным образом.

MySQL Workbench 8.0.38

Select Operating System:

Microsoft Windows ▼

Recommended Download:

MySQL Installer for Windows

**All MySQL Products. For All Windows Platforms.
In One Package.**

Starting with MySQL 5.6 the MySQL Installer package replaces the standalone MSI packages.



Windows (x86, 32 & 64-bit), MySQL Installer MSI

[Go to Download Page >](#)

Other Downloads:

Windows (x86, 64-bit), MSI Installer

8.0.38

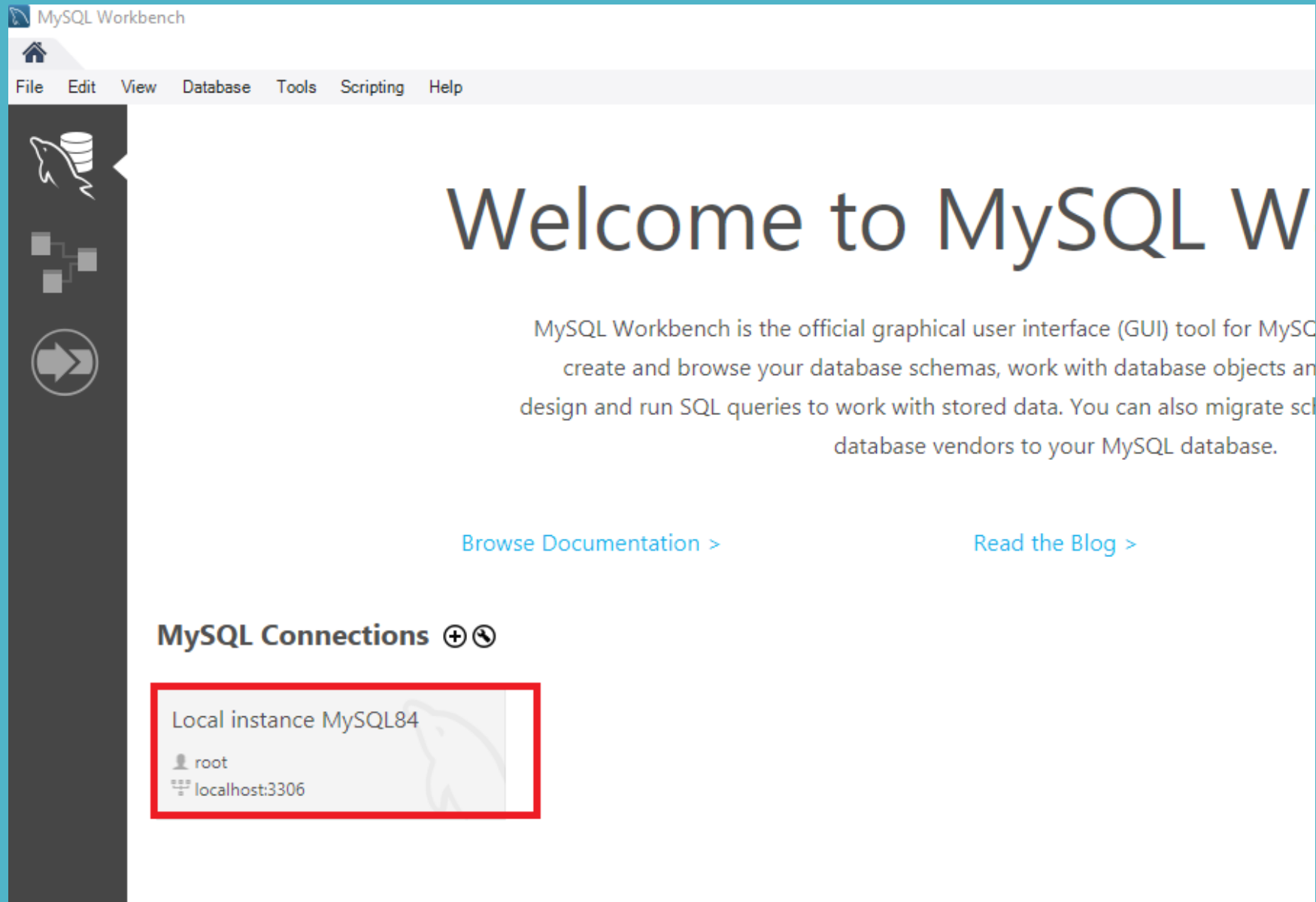
41.7M

[Download](#)

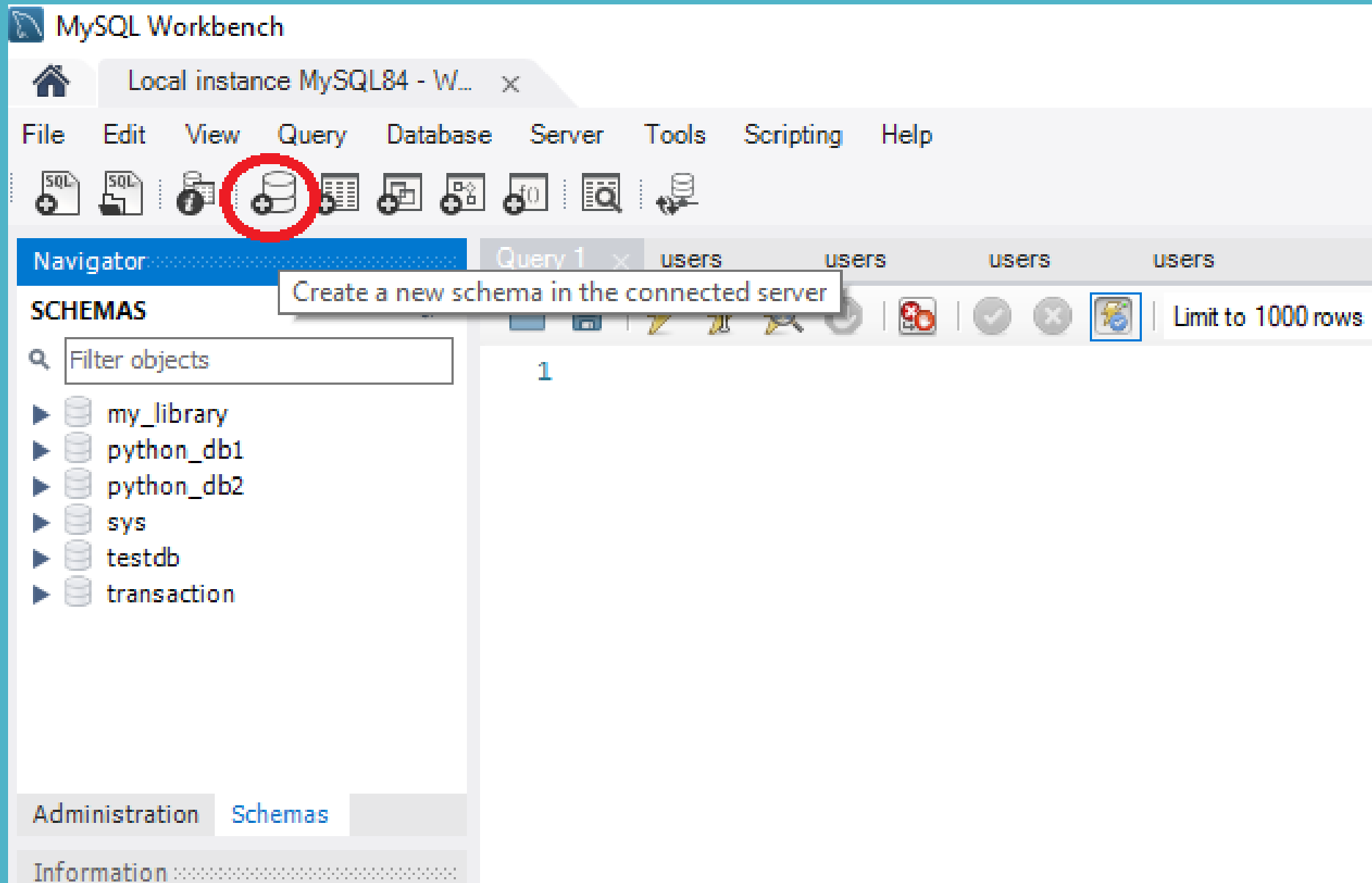
(mysql-workbench-community-8.0.38-winx64.msi)

MD5: 30ea58c9f40816566ac4ccd2f136f1e2 | [Signature](#)

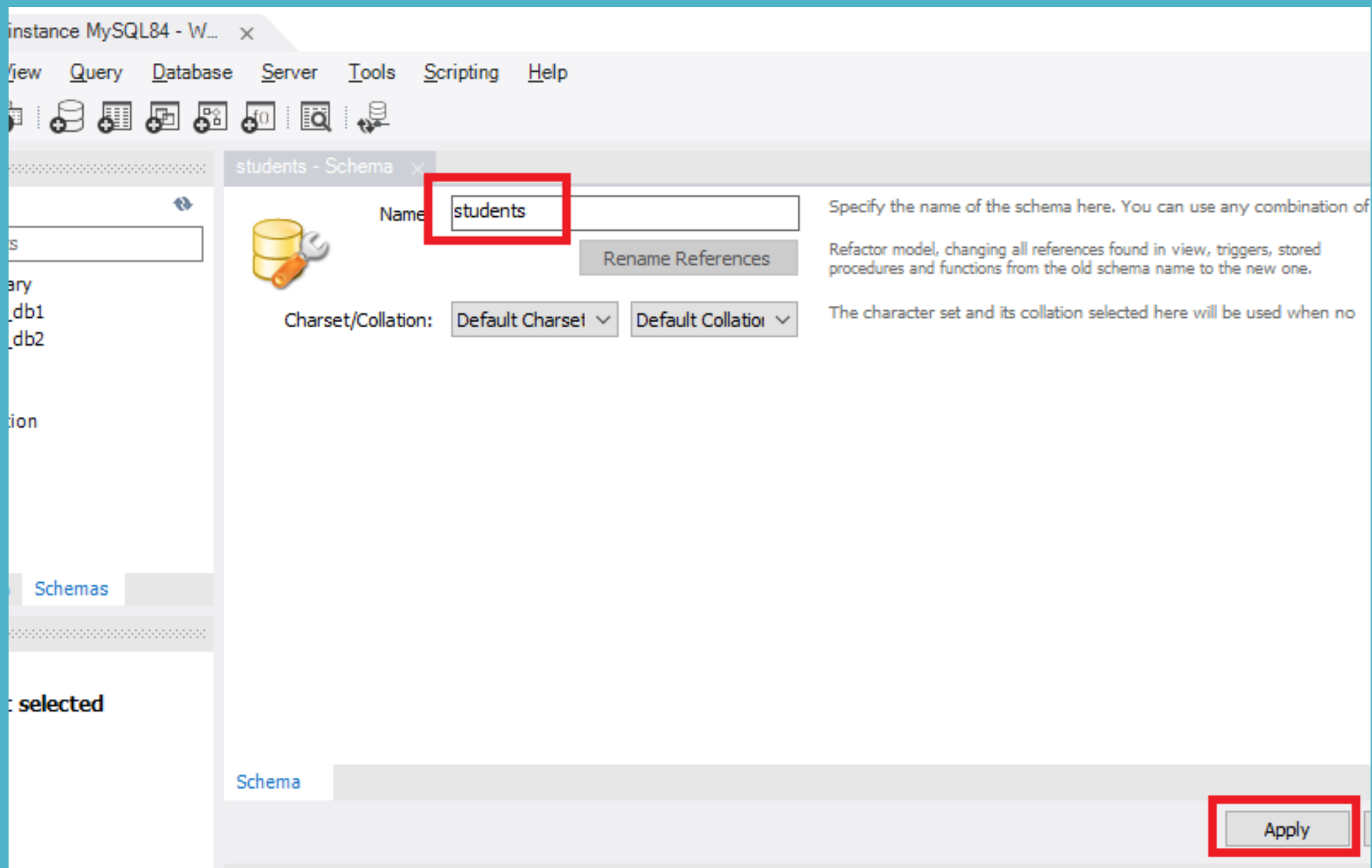
Подключение к базе данных:



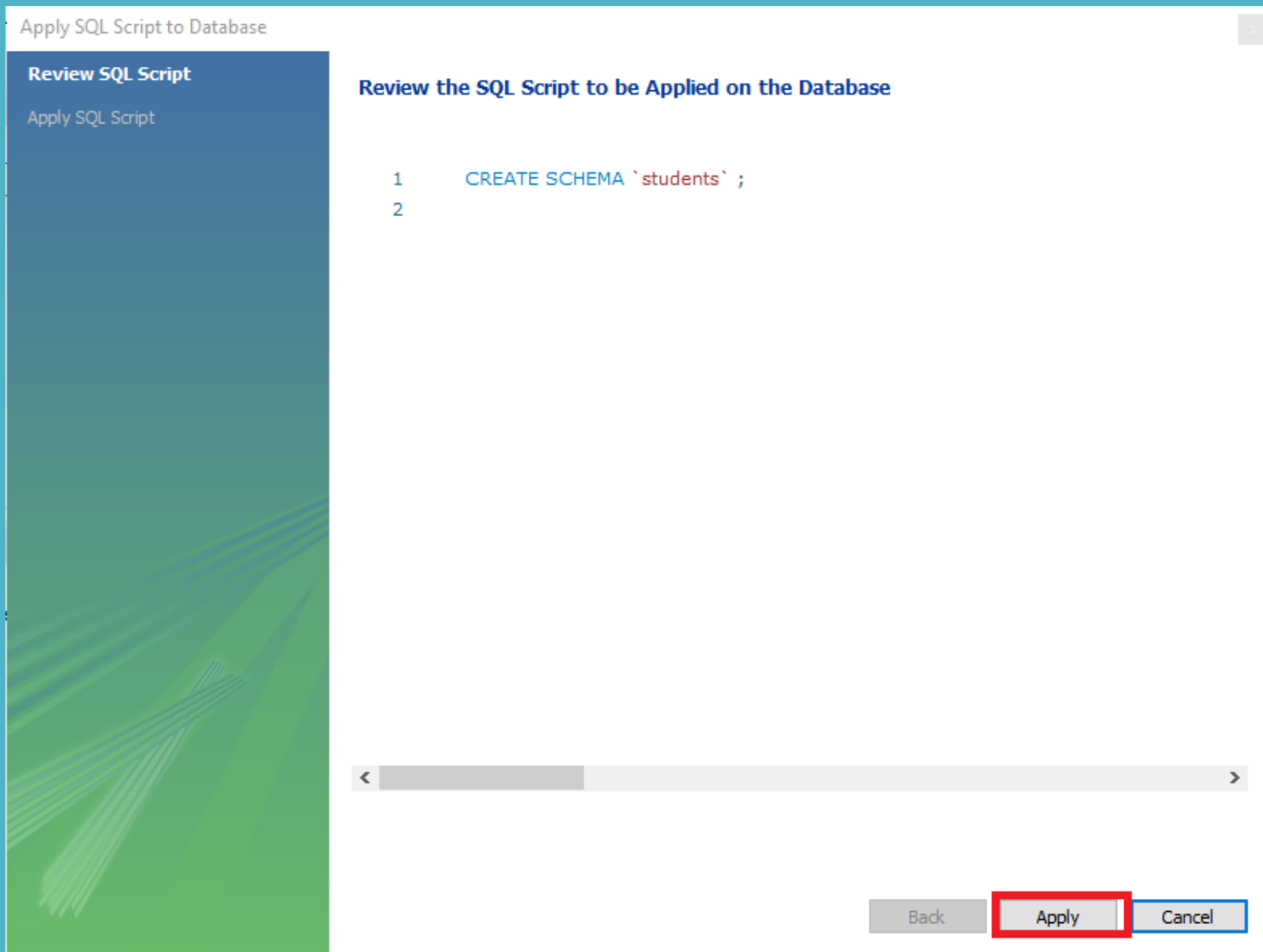
Создание новой БД:



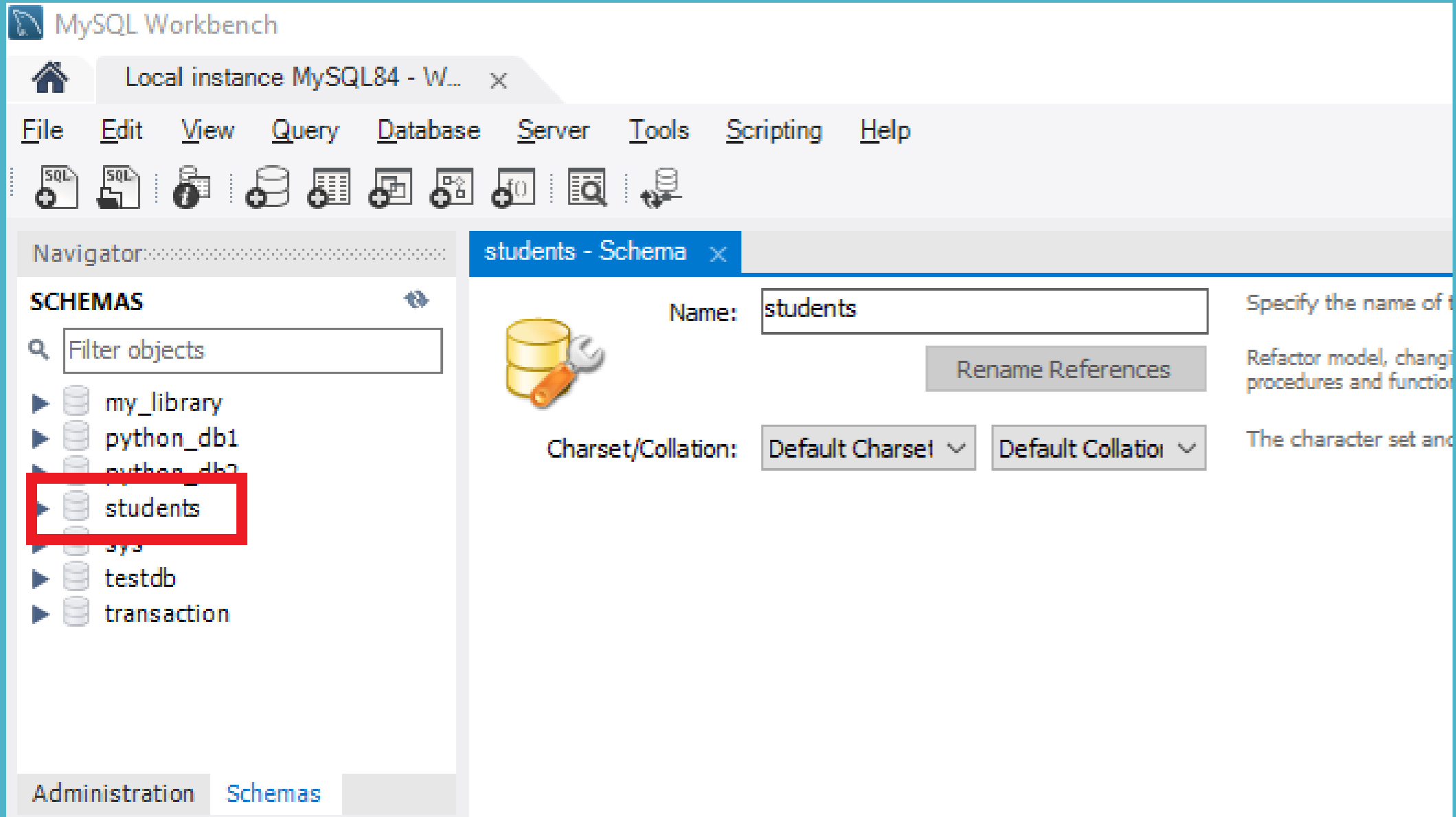
Далее



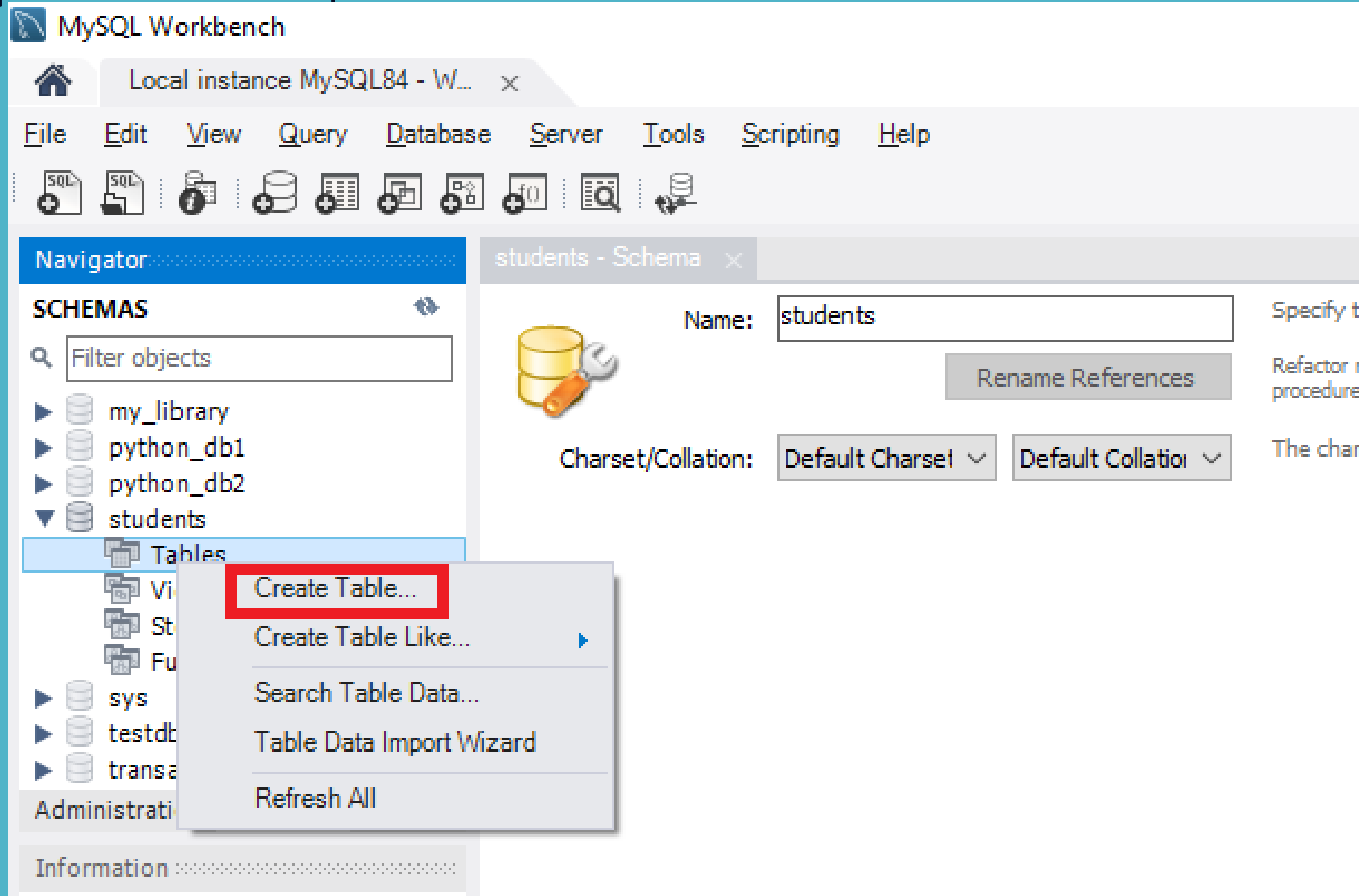
Далее



Далее



Создание таблицы



Далее

students - Schema users - Table

Table Name: **users** Schema: **students**

Charset/Collation: Default Charset Default Collation Engine: InnoDB

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
name	VARCHAR(45)	<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"/>	
rating	DECIMAL(4,2)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Column Name: rating Data Type: VARCHAR(45)

Charset/Collation: Default Charset Default Collation Default:

Comments:

Storage: ☐ Virtual ☐ Stored

☐ Primary Key ☐ Not Null ☐ Unique

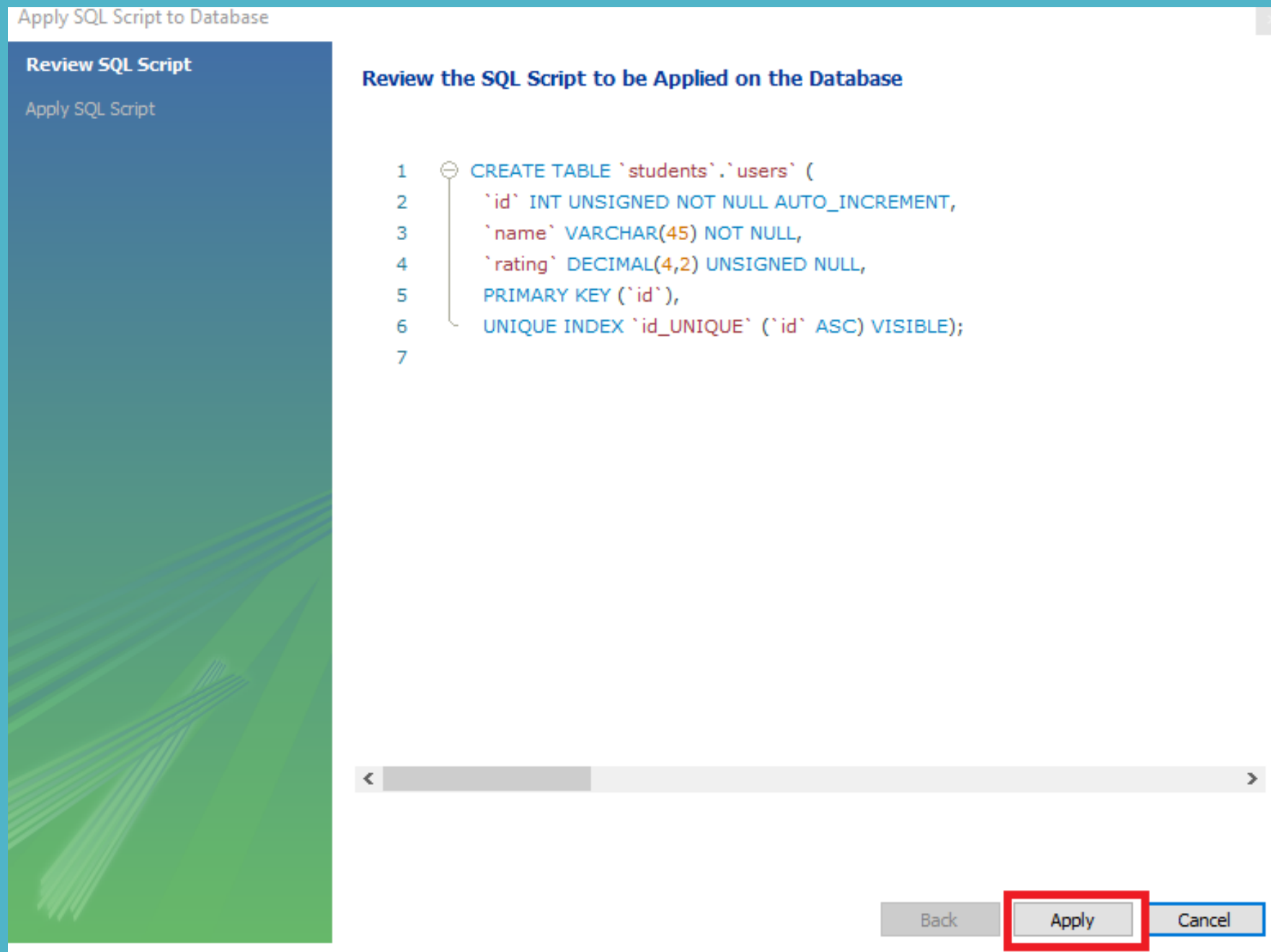
☐ Binary ☒ Unsigned ☐ Zero Fill

☐ Auto Increment ☐ Generated

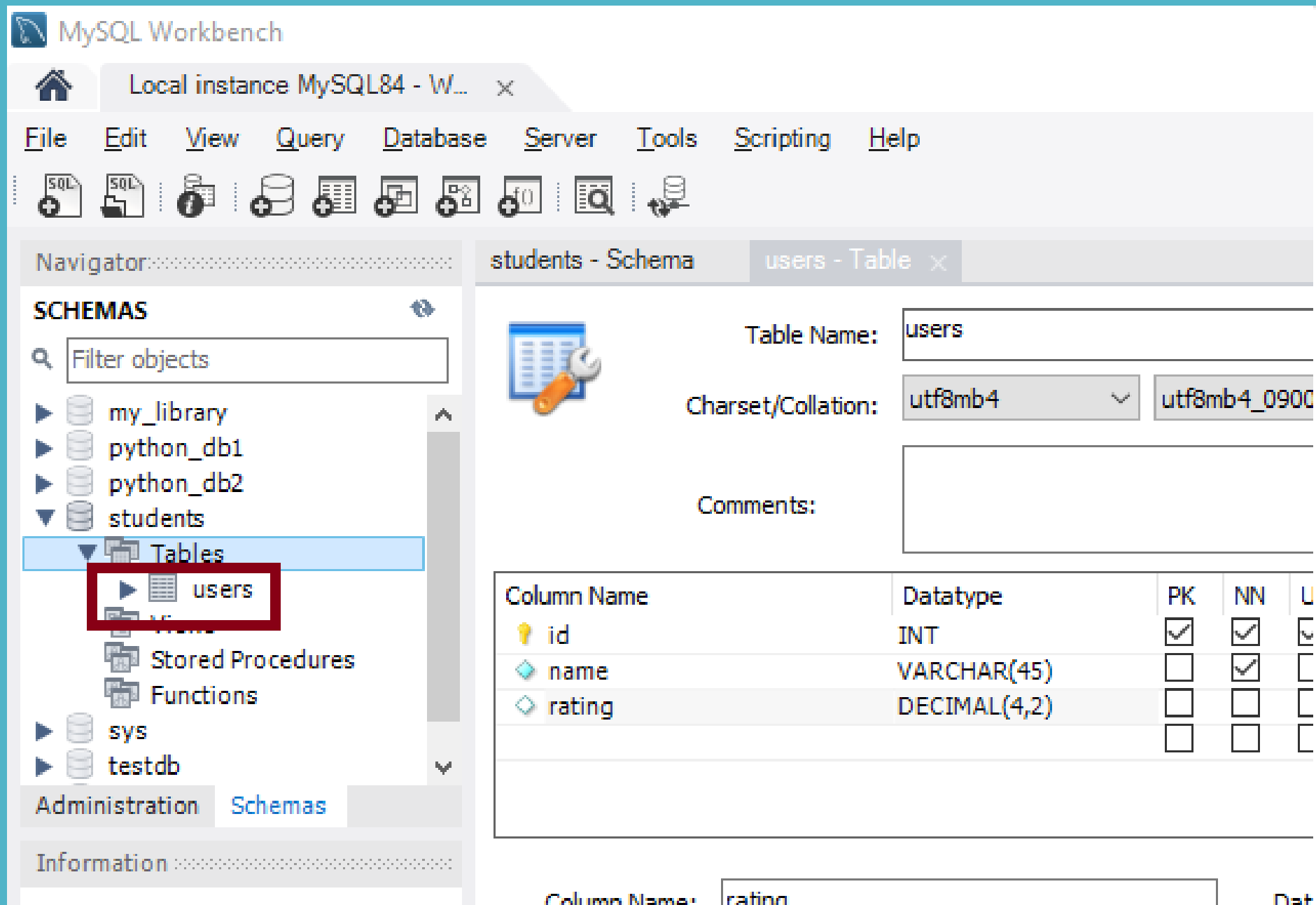
Columns Indexes Foreign Keys Triggers Partitioning Options

Apply Revert

Далее

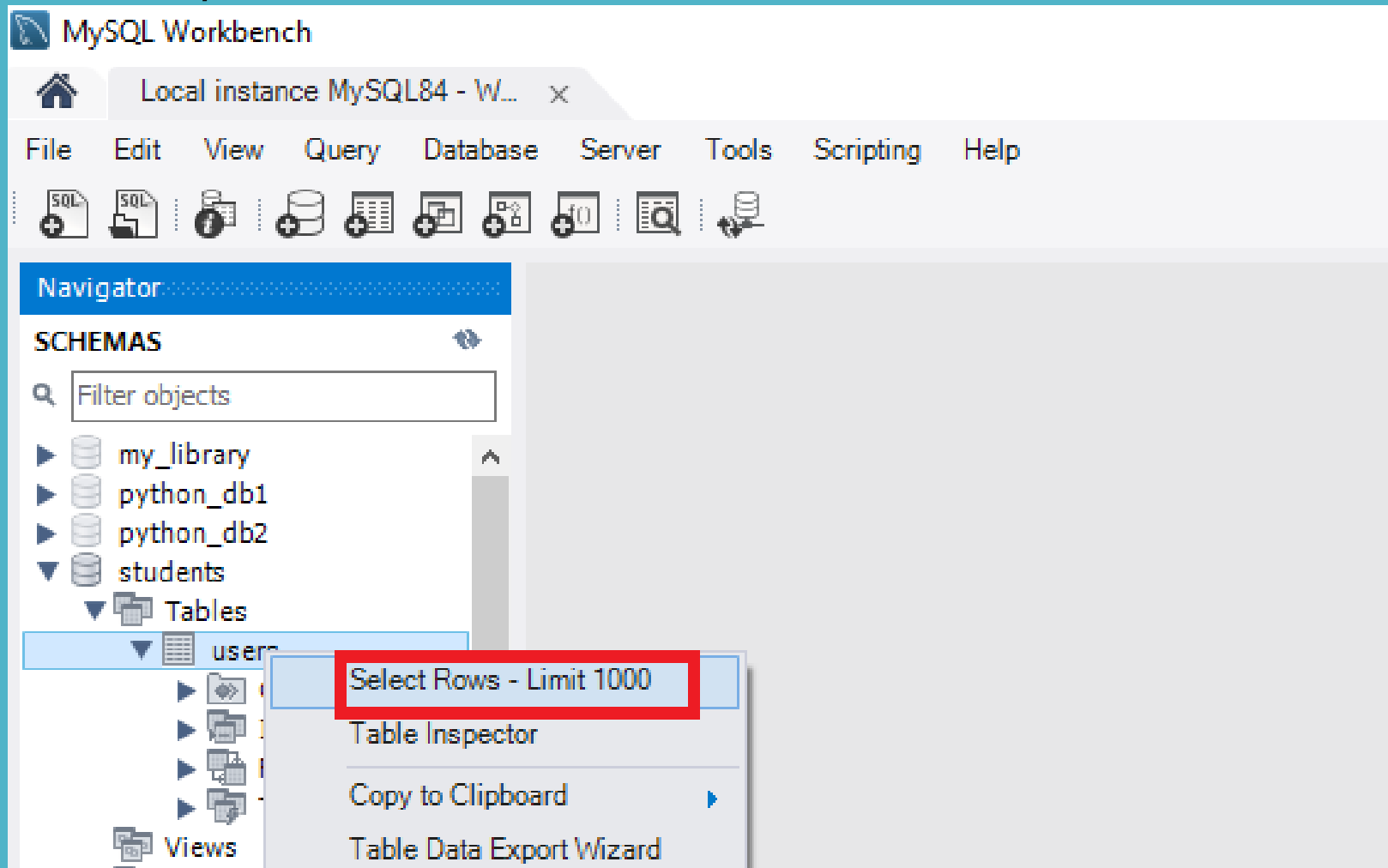


Далее



Вставка данных в таблицу:

Правой кнопкой мыши кликаем на таблицу, выбираем «Select Rows», затем «Insert new Row»:



Заполняем строки и нажимаем «Apply»:

The screenshot shows a web-based data management interface. At the top, there's a toolbar with various icons for editing and exporting data. Below the toolbar is a table with three columns: 'id', 'name', and 'rating'. The table contains two rows of data: one with 'Petrov' and a rating of 60, and another with 'Ivanov' and a rating of 80. The 'id' column for both rows shows 'NULL'. A red rectangle highlights the two data rows. In the bottom right corner, there are two buttons: 'Apply' and 'Revert'. The 'Apply' button is highlighted with a red rectangle. The bottom left corner shows a tab labeled 'users 1'.

id	name	rating
NULL	Petrov	60
NULL	Ivanov	80

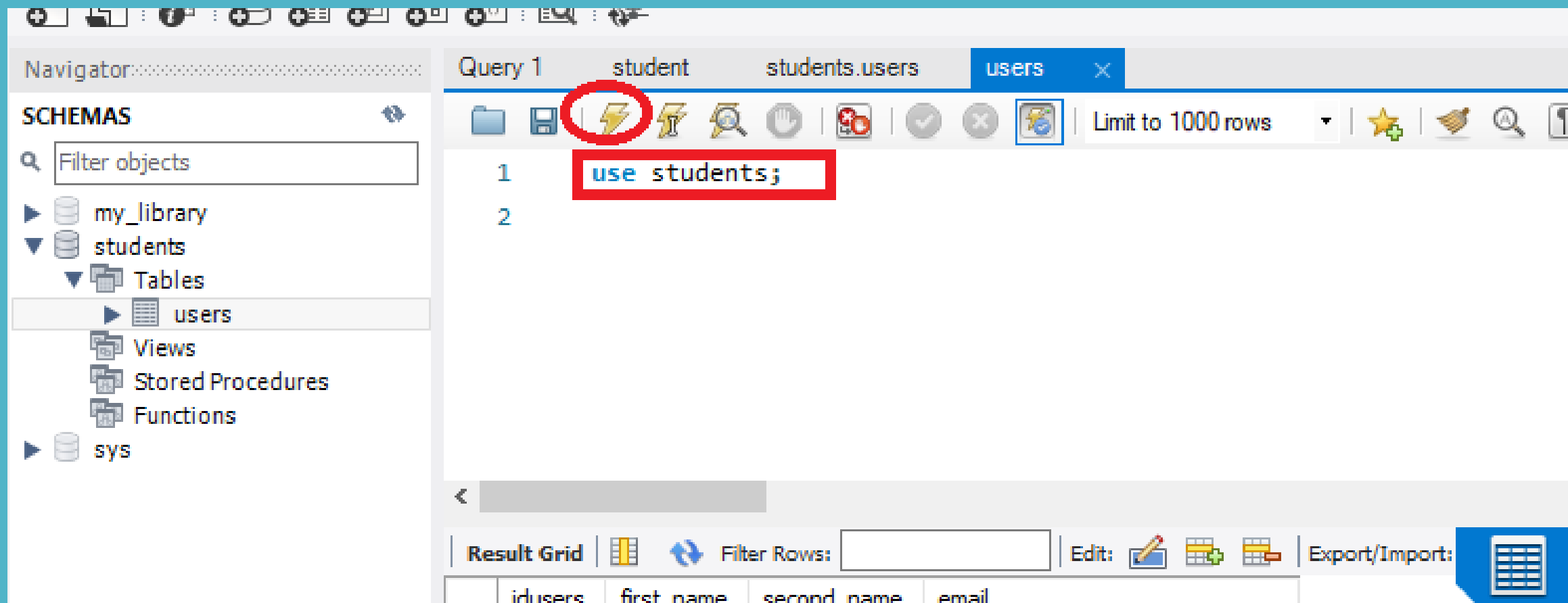
users 1 x

Apply Revert

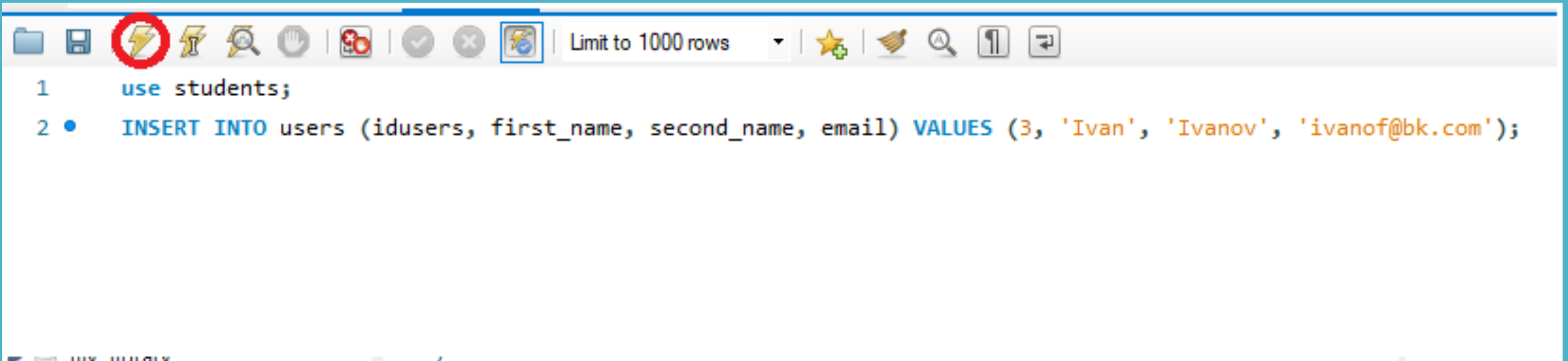
Вставка данных в таблицу с помощью SQL-запросов. В поле SQL-запроса пишем:

`use <имя_БД>`

И нажимаем «Execute»:



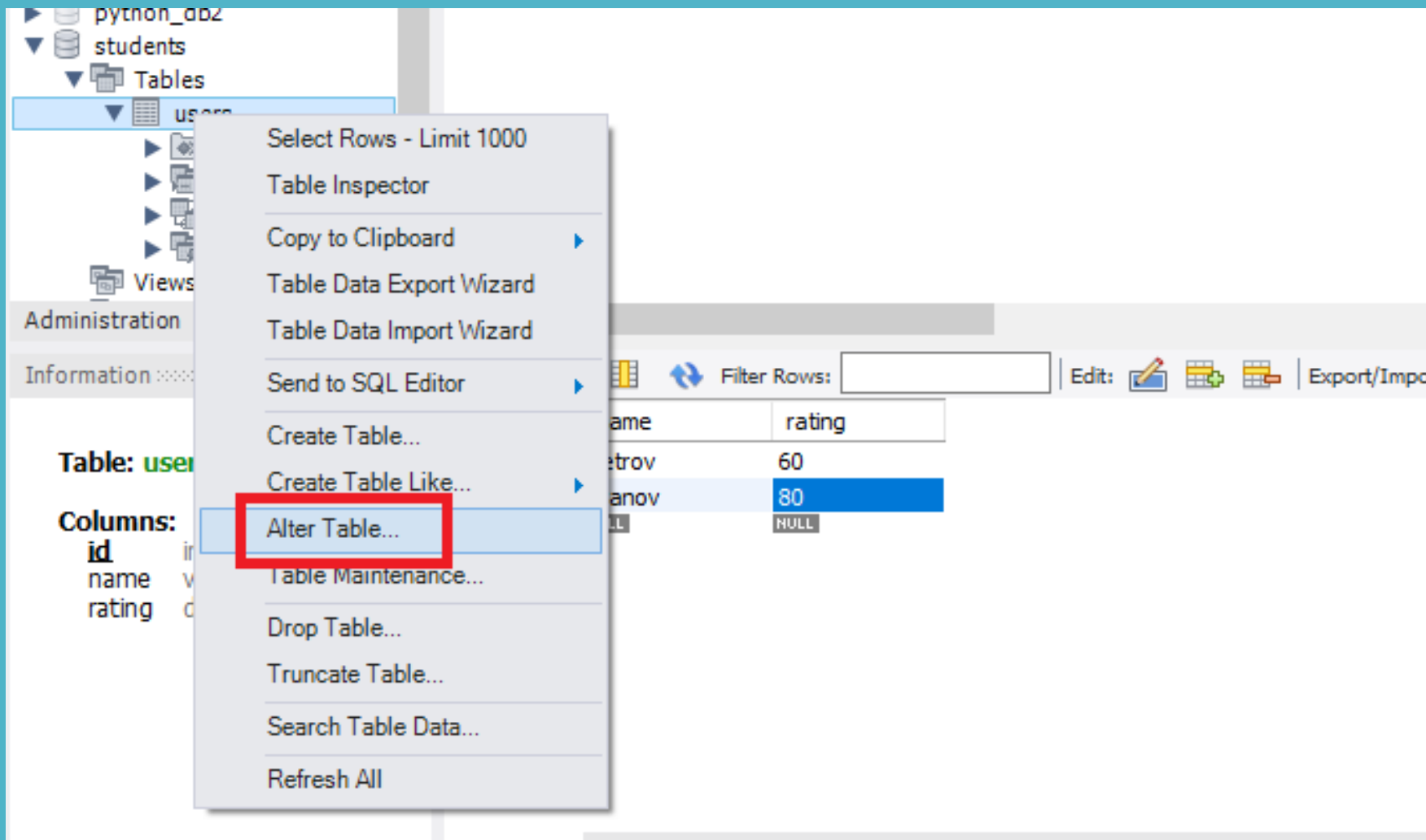
Пишем SQL-запрос и выполняем:



```
1 use students;
2 • INSERT INTO users (idusers, first_name, second_name, email) VALUES (3, 'Ivan', 'Ivanov', 'ivanof@bk.com');
```

✓	4	14:28:16	use students	0 row(s) affected
✓	5	14:28:16	INSERT INTO users (idusers, first_name, second_name, email) VALUES (3, 'Ivan', 'Ivanov', 'iv...	1 row(s) affected

Редактирование таблицы. Кликаем ПКМ на нужной таблице, выбираем «Alter Table», затем меняем нужное свойство:



Можно добавить еще один столбец, email:

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
💡 id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
💠 name	VARCHAR(45)	<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"/>	
💠 rating	DECIMAL(4,2)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
💠 email	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Column Name:

Charset/Collation:

Default Charset

Default Collation

Comments:

Data Type:

Default:

Storage:

☐ Virtual

☐ Stored

☐ Primary Key

☐ Not Null

☐ Unique

☐ Binary

☐ Unsigned

☐ Zero Fill

☐ Auto Increment

☐ Generated

Columns

Indexes

Foreign Keys

Triggers

Partitioning

Options

Apply

Revert

Что посмотреть содержимое страницы, кликаем на ней ПКМ и выбираем «Select Rows»:

The screenshot shows a database management tool interface. At the top, there are tabs labeled 'users', 'users - Table', and 'users'. Below the tabs is a toolbar with various icons. A red box highlights a lightning bolt icon in the toolbar. Below the toolbar, a SQL query is entered: `1 • SELECT * FROM students.users;`. At the bottom, there is a 'Result Grid' section. A red box highlights the first four rows of the grid, which are: a header row with columns 'id', 'name', and 'rating'; and three data rows with values (1, Ivanov, 60.00), (2, Petrov, 80.00), and (3, Stepanov, 95.00). The fourth row is a row with NULL values. The interface also includes a 'Filter Rows' field and various action buttons like 'Edit', 'Export/Import', and 'Wrap Cell Content'.

	id	name	rating
▶	1	Ivanov	60.00
	2	Petrov	80.00
	3	Stepanov	95.00
*	NULL	NULL	NULL

Экспорт данных в csv-файл. Нажмите иконку экспорта, выберите название файла и сохраните.

The screenshot shows a database management interface. At the top, there are tabs for different tables: 'student', 'users', 'users', 'users - Table', 'students.users', 'users', and 'users'. Below the tabs is a toolbar with various icons. The main area displays a SQL query: `1 • SELECT * FROM students.users;`. Below the query, there is a 'Result Grid' section. The 'Result Grid' has a toolbar with icons for 'Filter Rows', 'Edit', and 'Export/Import'. The 'Export/Import' icon is highlighted with a red box. Below the toolbar, a table displays the results of the query. The table has five columns: 'idusers', 'first_name', 'second_name', 'email', and 'average_rating'. The data rows are as follows:

	idusers	first_name	second_name	email	average_rating
▶	1	Nikita	Stepanov	stepa@gmail.com	NULL
	2	Alina	Kabirova	alina@ya.com	NULL
	3	Ivan	Ivanov	ivanof@bk.com	NULL
*	NULL	NULL	NULL	NULL	NULL



Export Resultset



<< Users > user > Документы



Поиск в: Документы



Упорядочить ▾

Новая папка



LW



Этот компьютер



Видео



Документы



Загрузки



Изображения



Музыка



Рабочий стол



Windows 10 (C:)



DiscD (D:)



Имя



Дата изменения

Тип



1234

03.11.2024 17:42

Папка с фай



Atmel Studio

02.07.2024 13:19

Папка с фай



csharp_projects

29.08.2024 20:54

Папка с фай



dumps

14.09.2024 21:18

Папка с фай



IISExpress

31.07.2024 10:13

Папка с фай



Lightshot

23.02.2025 12:48

Папка с фай



My Web Sites

31.07.2024 10:13

Папка с фай



SharpDevelop Projects

28.08.2024 15:01

Папка с фай



Visual Studio 2010

02.07.2024 13:17

Папка с фай



Visual Studio 2022

01.08.2024 14:20

Папка с фай



Имя файла:

students_users



Тип файла:

CSV (; separated)



^ Скрыть папки

Сохранить

Отмена

[illegible]

7. DDL запросы. Создание ролей и БД через консоль.

DDL-запросы (Data Definition Language) в MySQL – это язык определения данных, используемый для создания и управления структурами базы данных, такими как базы данных, таблицы и индексы.

Основные команды DDL включают создание, изменение и удаление объектов базы данных.

Далее рассмотрим основные команды DDL-запросов в MySQL.

1. CREATE: создание объектов базы данных

- Создание базы данных:

```
CREATE DATABASE db_name;
```

Пример. Создание БД «SchoolDB»:

```
CREATE DATABASE SchoolDB;
```

- **Создание пользователя**

```
CREATE USER 'username'@'hostname' IDENTIFIED BY 'password';
```

Пример:

```
CREATE USER 'student'@'localhost' IDENTIFIED BY 'password';
```

'username' — имя пользователя.

'hostname' — хост, с которого пользователь может подключаться (например, 'localhost').

'password' — пароль для этого пользователя.

- **Создание таблицы:**

```
CREATE TABLE table_name (  
    column1 datatype [constraints],  
    column2 datatype [constraints],  
    ...  
);
```

Пример:

```
CREATE TABLE Students (  
    StudentID INT PRIMARY KEY,  
    FirstName VARCHAR(50),  
    LastName VARCHAR(50),  
    BirthDate DATE  
);
```

- **Создание индекса*:**

```
CREATE INDEX index_name ON table_name (column_name);
```

Пример:

```
CREATE INDEX idx_lastname ON Students (LastName);
```

2. ALTER: изменение структуры существующих объектов

- Добавление нового столбца в таблицу:

```
ALTER TABLE table_name ADD column_name datatype;
```

Пример:

```
ALTER TABLE Students ADD Email VARCHAR(100);
```

- **Изменение типа данных столбца:**

```
ALTER TABLE table_name MODIFY column_name new_datatype;
```

Пример:

```
ALTER TABLE Students MODIFY Email TEXT;
```

- **Переименование таблицы:**

```
ALTER TABLE old_table_name RENAME TO new_table_name;
```

Пример:

```
ALTER TABLE Students RENAME TO Pupils;
```

- Удаление столбца из таблицы:

```
ALTER TABLE table_name DROP COLUMN column_name;
```

Пример:

```
ALTER TABLE Students DROP COLUMN Email;
```

3. **DROP**: удаление объектов базы данных

- Удаление базы данных:

```
DROP DATABASE db_name;
```

Пример:

```
DROP DATABASE SchoolDB;
```

- Удаление таблицы:

```
DROP TABLE table_name;
```

Пример:

```
DROP TABLE Students;
```


- Удаление индекса*:

```
DROP INDEX index_name ON table_name;
```

Пример:

```
DROP INDEX idx_lastname ON Students;
```

4. TRUNCATE: очистка всех данных из таблицы без удаления её структуры

- Очистка таблицы:

```
TRUNCATE TABLE table_name;
```

Пример:

```
TRUNCATE TABLE Students;
```

Основные принципы работы с DDL-запросами:

- Команды DDL обычно выполняются сразу и автоматически сохраняют изменения в базе данных (в отличие от DML-запросов, которые можно отменить с помощью команды ROLLBACK).
- После выполнения DDL-команды изменения **нельзя отменить** с помощью команды ROLLBACK.

Как установить владельца базы данных?

В MySQL нет прямой команды для назначения владельца базы данных, как, например, в PostgreSQL. Обычно права на базу данных контролируются через привилегии пользователей.

То есть, для пользователя можно установить полный доступ к базе данных, но назначить его "владельцем" в явном виде невозможно.

Однако, можно обеспечить полные привилегии пользователю на конкретную базу данных, что по сути эквивалентно понятию "владельца".

- Установка привилегий:

```
GRANT ALL PRIVILEGES ON database_name.* TO 'username'@'hostname';
```

Пример:

```
GRANT ALL PRIVILEGES ON SchoolDB.* TO 'student'@'localhost';
```

GRANT — ключевое слово, начинающее команду предоставления прав пользователю.

ALL PRIVILEGES — указывает, что пользователю будут предоставлены все возможные привилегии (права) на выполнение операций в указанной базе данных.

SchoolDB — имя базы данных, на которую предоставляются права.

***** — означает, что права распространяются на все таблицы и объекты в указанной базе данных.

TO 'student'@'localhost' — указывает, какому пользователю и откуда предоставляются права:

'student' — имя пользователя MySQL, которому предоставляются права.

'localhost' — хост, с которого этот пользователь может подключаться к базе данных. В данном случае 'localhost' указывает, что пользователь может подключаться только с локального компьютера, на котором запущен MySQL.

Как подключиться к созданной базе данных?

- Через командную строку:

```
mysql -u username -p -h hostname database_name
```

-u username — имя пользователя.

-p — запрос пароля.

-h hostname — адрес сервера (можно пропустить для подключения к localhost).

database_name — имя базы данных, к которой нужно подключиться.

Пример:

```
mysql -u student -p -h localhost SchoolDB
```

- **Через MySQL Workbench**
 - Откройте MySQL Workbench.
 - Создайте новое подключение:
 - В поле "Connection Name" укажите название подключения.
 - В поле "Hostname" укажите сервер (например, localhost).
 - В поле "Username" укажите имя пользователя (например, student).
 - Нажмите "Test Connection" и введите пароль для пользователя.
 - После успешного теста подключения нажмите "ОК".
 - Двойным кликом по созданному подключению вы подключитесь к базе данных.
 - Теперь можно выбирать базу данных и работать с ней через MySQL Workbench.

Домашнее задание:

Задание 1. Работа с MySQL через консоль

Создайте пользователя с именем test_user и паролем password123.

Создайте базу данных с именем test_db.

Проверьте, что пользователь и база данных созданы:

Дайте пользователю test_user все права на базу данных test_db.

Подключитесь к базе данных test_db.

Создайте таблицу users с полями:

- id (INT, первичный ключ, автоинкремент)
- username (VARCHAR(50), не может быть NULL)
- email (VARCHAR(100), не может быть NULL)

Добавьте несколько записей в таблицу users.

Выберите все записи из таблицы users, чтобы убедиться, что данные были добавлены.

Задание 2. Базовые операции в MySQL Workbench.

1. Создание пользователя:

- Откройте вкладку "Administration" в MySQL Workbench.
- Перейдите в раздел "Users and Privileges".
- Создайте нового пользователя с именем `workbench_user` и паролем `workbench123`.
- Убедитесь, что пользователь создан, проверив его в списке пользователей.

2. Создание базы данных:

- В главном окне MySQL Workbench выберите вкладку "Schemas".
- Создайте новую базу данных с именем `workbench_db`.
- Убедитесь, что база данных появилась в списке схем.
- Предоставление прав пользователю на базу данных:
- Вернитесь в раздел "Users and Privileges".
- Найдите пользователя `workbench_user` и предоставьте ему все права на базу данных `workbench_db`.
- Сохраните изменения.

Задание 3. Создание таблицы в базе данных:

- Перейдите к базе данных workbench_db в разделе "Schemas".
- Создайте новую таблицу с именем products с полями:
- id (INT, первичный ключ, автоинкремент)
- product_name (VARCHAR(100), не может быть NULL)
- price (DECIMAL(10, 2), не может быть NULL)

Используйте графический интерфейс для добавления столбцов и настройки их свойств.

Задание 4. Вставка записей в таблицу:

- Откройте таблицу products и перейдите на вкладку "Table Data".
- Добавьте несколько записей вручную через интерфейс, например:
- product_name: "Laptop", price: 1200.00
- product_name: "Smartphone", price: 800.00
- product_name: "Tablet", price: 500.00

Задание 5. Просмотр таблицы:

- Убедитесь, что данные были добавлены, просмотрев таблицу products через вкладку "Table Data".

Выполнение SQL-запросов:

Откройте вкладку "Query" и выполните SQL-запрос для выборки всех данных из таблицы products:

```
SELECT * FROM workbench_db.products;
```

Убедитесь, что результат отображается корректно.

Список литературы:

1. [Руководство по MySQL.](#)
2. Видеокурс.

Материалы лекций:

<https://github.com/ShViktor72/Education2025>