

ГУАП

КАФЕДРА № 41

ОТЧЕТ  
ЗАЩИЩЕН С ОЦЕНКОЙ  
ПРЕПОДАВАТЕЛЬ

канд. техн. наук, доцент  
\_\_\_\_\_  
должность, уч. степень, звание

\_\_\_\_\_  
подпись, дата

Е.Л. Турнецкая  
\_\_\_\_\_  
инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №1

УСТАНОВКА MYSQL И НАСТРОЙКА СЕРВЕРА БАЗ ДАННЫХ

по курсу: БАЗЫ ДАННЫХ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. № 4217

\_\_\_\_\_  
подпись, дата

Д.М. Никитин  
\_\_\_\_\_  
инициалы, фамилия

Санкт-Петербург 2024

1. **Цель работы:** установка MySQL-сервера и его компонент на персональный компьютер; изучение возможностей клиентских утилит при взаимодействии с сервером для реализации реляционной БД.

2. **Вариант:** мой номер по списку – 24 → вариант 24.

2.1. Сведения о водителе (Код водителя, Фамилия, Имя, Номер водительского удостоверения).

2.2. Автомашина (Код машины, Марка, Номер машины, Год выпуска).

2.3. Дорожные происшествия (Код события, Код машины, Код водителя, Дата происшествия, Сумма ущерба).

3. **Задание:**

3.1. Внимательно прочтите методические указания по установке сервера.

3.2. Следуя инструкциям, установите MySQL-сервер на персональный компьютер. Не делайте скриншоты этапов инсталляции сервера. Для подтверждения факта установки сервера сделайте скриншоты файлов сервера, которые будут расположены в следующих папках:

– C:\Program Files\MySQL;

– C:\Program Files (x86) \MySQL;

– C:\ProgramData\MySQL.

3.3. В списке вариантов заданий по реализации однотабличной БД найдите свой вариант, который соответствует порядковому номеру в списке учебной группы.

3.4. Запустите сервер с помощью консольного клиента MySQL 8.0 Command Line Client – Unicode.

3.4.1. Выполните упражнение по реализации однотабличной БД в среде консольного клиента из методических указаний для понимания команд по реализации БД. Сделайте скриншоты двух последних операций.

3.4.2. Реализуйте первую однотабличную БД по своему варианту с количеством записей в таблице не менее пяти. Порядок выполнения аналогичен порядку действий в упражнении. Покажите все результаты выполнения задания на скриншотах. При заполнении таблиц данными ориентируйтесь на типы данных, поддерживаемых MySQL-сервером.

3.5. Остановите сервер, выполнив команду `EXIT (QUIT)`.

3.6. Запустите графический клиент MySQL Workbench.

3.6.1. Создайте соединение с сервером БД.

3.6.2. Выполните упражнение по реализации БД средствами графического клиента из методических указаний.

3.6.3. Реализуйте вторую однотабличную БД по своему варианту с количеством записей в таблице не менее пяти. Покажите результаты выполнения задания на скриншотах. Порядок выполнения аналогичен порядку действий в упражнении. Покажите все результаты выполнения задания на скриншотах.

3.7. Выполните отчет в соответствии с требованиями ГОСТ 7.32-2017 и ГОСТ 2.105-2019: по оформлению отчетов (<https://guap.ru/standart/doc>).

#### **4. Содержание отчета**

4.1. Титульный лист.

4.2. Цель работы.

4.3. В произвольной форме напишите:

– зачем требуется устанавливать локальный сервер,

– какие объекты на нем хранят;

– что такое архитектура «клиент-сервер».

4.4. Продемонстрируйте на скриншотах факт того, что локальный сервер установлен на компьютере.

4.5. Покажите на скриншотах выполнения заданий по реализации первой однотабличной БД в среде консольного клиента,

второй БД – в среде графического клиента. Связывайте скриншоты между собой пояснительным текстом. Команды для сервера также зафиксируйте в текстовом формате: Courier New 12, межстрочный интервал 1.0.

4.6. Сделайте вывод в формате эссе о результатах выполнения заданий.

## 5. Ход работы:

В произвольной форме пишу по вопросам.

Установка локального сервера MySQL необходима для эффективной работы с базами данных в среде разработки, тестирования или локальной эксплуатации. MySQL — это реляционная система управления базами данных (СУБД), которая позволяет хранить, управлять и обрабатывать данные с помощью языка запросов SQL (Structured Query Language). Основные причины для установки локального сервера MySQL включают: разработка приложений, тестирование, обучение и эксперименты, производительность и скорость доступа.

На сервере MySQL можно хранить различные типы объектов базы данных. Основные из них включают: таблицы, численные данные, дата и время, строковые, пространственные, JSON и другие.

Архитектура «клиент-сервер» — это модель взаимодействия в компьютерных сетях, в которой один или несколько клиентов (клиентские устройства) отправляют запросы к серверу (серверному устройству), а сервер обрабатывает эти запросы и отправляет клиентам ответы.

После ответа на вопросы была произведена установка MySQL-сервера на локальный компьютер.

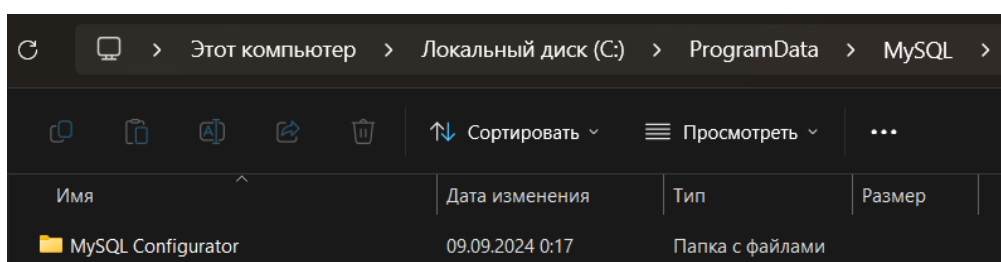


Рисунок 1 – Файлы в папке ProgramData

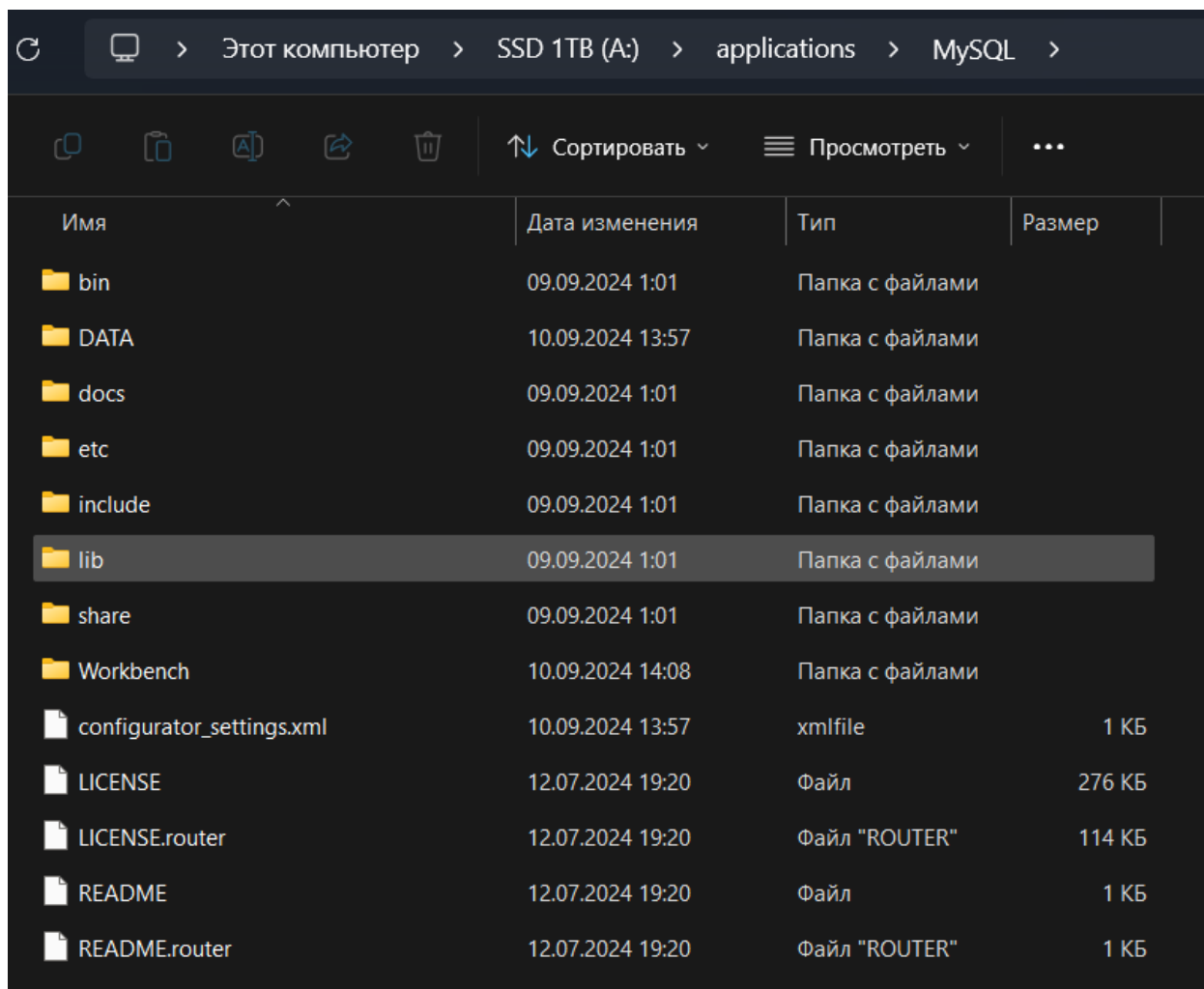


Рисунок 2 – Файлы установленного SQL

Так как MySQL был установлен в папку, выбранную вручную, то в папке Program Files (x86) файлы MySQL не появились, а были установлены вместе с файлами, которые должны были появиться в папке Program Files.

Приступим далее к созданию базы данных с помощью текстового интерфейса. Первым делом открываем MySQL 8.4 Command Line Client и вводим установленный ранее пароль.

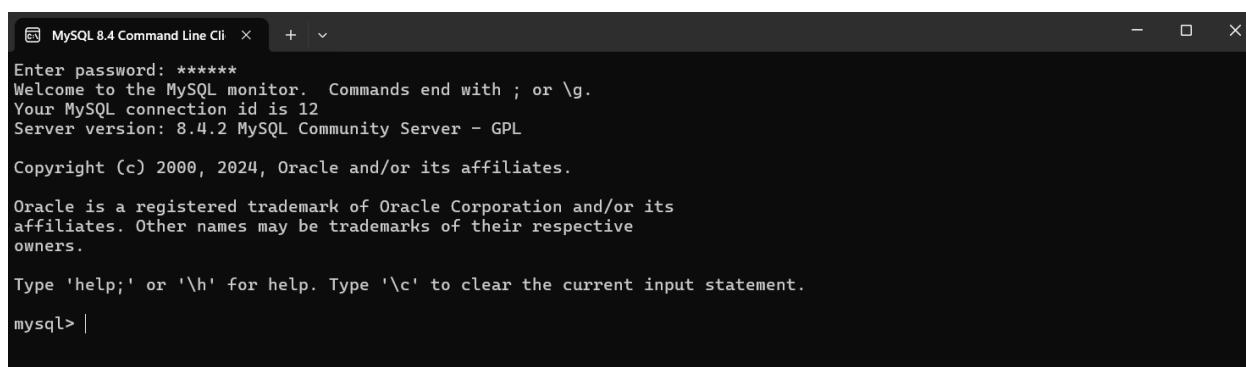


Рисунок 3 – Командная строка MySQL и введённый пароль

Сначала используем команду `SHOW DATABASES;`, чтобы посмотреть на все базы данных на сервере.

```
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql      |
| performance_schema |
| sys        |
+-----+
4 rows in set (0.01 sec)
```

Рисунок 4 – Базы данных сервера через консоль

Посмотрим состав `information_schema` с помощью команд `USE information_schema;` и `SHOW TABLES;`.

```
mysql> USE information_schema;
Database changed
mysql> SHOW TABLES;
+-----+
| Tables_in_information_schema |
+-----+
| ADMINISTRABLE_ROLE_AUTHORIZATIONS |
| APPLICABLE_ROLES |
| CHARACTER_SETS |
| CHECK_CONSTRAINTS |
| COLLATION_CHARACTER_SET_APPLICABILITY |
| COLLATIONS |
| COLUMN_PRIVILEGES |
| COLUMN_STATISTICS |
| COLUMNS |
| COLUMNS_EXTENSIONS |
| ENABLED_ROLES |
| ENGINES |
| EVENTS |
| FILES |
| INNODB_BUFFER_PAGE |
| INNODB_BUFFER_PAGE_LRU |
| INNODB_BUFFER_POOL_STATS |
| INNODB_CACHED_INDEXES |
| INNODB_CMP |
| INNODB_CMP_PER_INDEX |
| INNODB_CMP_PER_INDEX_RESET |
| INNODB_CMP_RESET |
| INNODB_CMPMEM |
| INNODB_CMPMEM_RESET |
| INNODB_COLUMNS |
| INNODB_DATAFILES |
| INNODB_FIELDS |
| INNODB_FOREIGN |
| INNODB_FOREIGN_COLS |
| INNODB_FT_BEING_DELETED |
| INNODB_FT_CONFIG |
| INNODB_FT_DEFAULT_STOPWORD |
| INNODB_FT_DELETED |
| INNODB_FT_INDEX_CACHE |
| INNODB_FT_INDEX_TABLE |
```

Рисунок 5 – Действие вышеупомянутых команд и первая часть таблицы

Для теста выполним несколько команд одновременно, например, `SELECT VERSION(), CURRENT_DATE;`.

```
mysql> SELECT VERSION(), CURRENT_DATE;
+-----+-----+
| VERSION() | CURRENT_DATE |
+-----+-----+
| 8.4.2      | 2024-09-10   |
+-----+-----+
1 row in set (0.00 sec)
```

Рисунок 6 – Выполнение двух команд

Далее создадим однотабличную базу данных с помощью текстового редактора. С помощью команды `CREATE DATABASE car_drivers;` создадим базу данных для водителей. И выберем её с помощью `USE car_drivers;`.

```
mysql> CREATE DATABASE car_drivers;
Query OK, 1 row affected (0.01 sec)

mysql> USE car_drivers;
Database changed
```

Рисунок 7 – Создание и выбор базы данных

С помощью команды `CREATE TABLE drivers`  
`-> (driver_id INTEGER AUTO_INCREMENT PRIMARY KEY,`  
`-> driver_family VARCHAR(25),`  
`-> driver_name VARCHAR(25),`  
`-> license_number INTEGER);` была создана таблица `drivers` с соответствующими полями. После этого продемонстрирована.

```
mysql> CREATE TABLE drivers
-> (driver_id INTEGER AUTO_INCREMENT PRIMARY KEY,
-> driver_family VARCHAR(25),
-> driver_name VARCHAR(25),
-> license_number INTEGER);
Query OK, 0 rows affected (0.02 sec)

mysql> SHOW TABLES;
+-----+
| Tables_in_car_drivers |
+-----+
| drivers                |
+-----+
1 row in set (0.00 sec)
```

Рисунок 8 – Таблица создана в текстовом редакторе

С помощью `DESCRIBE drivers;` получаем структуру таблицы.

```
mysql> DESCRIBE drivers;
```

Field	Type	Null	Key	Default	Extra
driver_id	int	NO	PRI	NULL	auto_increment
driver_family	varchar(25)	YES		NULL	
driver_name	varchar(25)	YES		NULL	
license_number	int	YES		NULL	

```
4 rows in set (0.00 sec)
```

Рисунок 9 – Структура таблицы

Теперь добавлю 5 записей в таблицу. С помощью команды `INSERT INTO drivers (driver_family, driver_name, license_number) VALUES ("Никитин", "Дмитрий", 20);`.

```
mysql> SELECT * FROM drivers;
```

driver_id	driver_family	driver_name	license_number
1	Никитин	Дмитрий	20
2	Медянкина	Анастасия	21
3	Афанасьев	Богдан	22
4	Никитин	Иван	23
5	Ярошенко	Михаил	24

```
5 rows in set (0.00 sec)
```

Рисунок 10 – Добавленные в таблицу пользователи

Просмотрим созданную базу данных на сервере с помощью команды `SHOW DATABASES;`.

```
mysql> SHOW DATABASES;
```

Database
car_drivers
information_schema
mysql
performance_schema
sys

```
5 rows in set (0.00 sec)
```

Рисунок 11 – База данных на сервере

Попробуем создать ещё одну базу данных с уже существующим именем.



```
mysql> CREATE DATABASE car_drivers;  
ERROR 1007 (HY000): Can't create database 'car_drivers'; database exists
```

Рисунок 12 – Попытка создать базу с существующим названием

Далее попробуем реализовать базу данных «Автомобиль» с помощью графического интерфейса.

Был запущен графический клиент для MySQL.

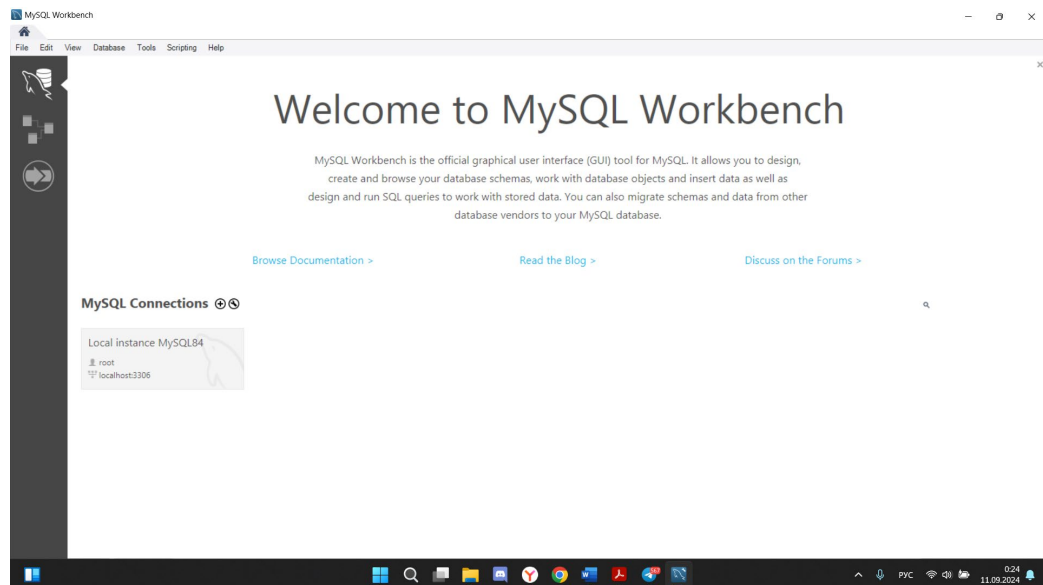


Рисунок 13 – Первый запуск MySQL Workbench

Далее было произведено подключение к локальному серверу MySQL.

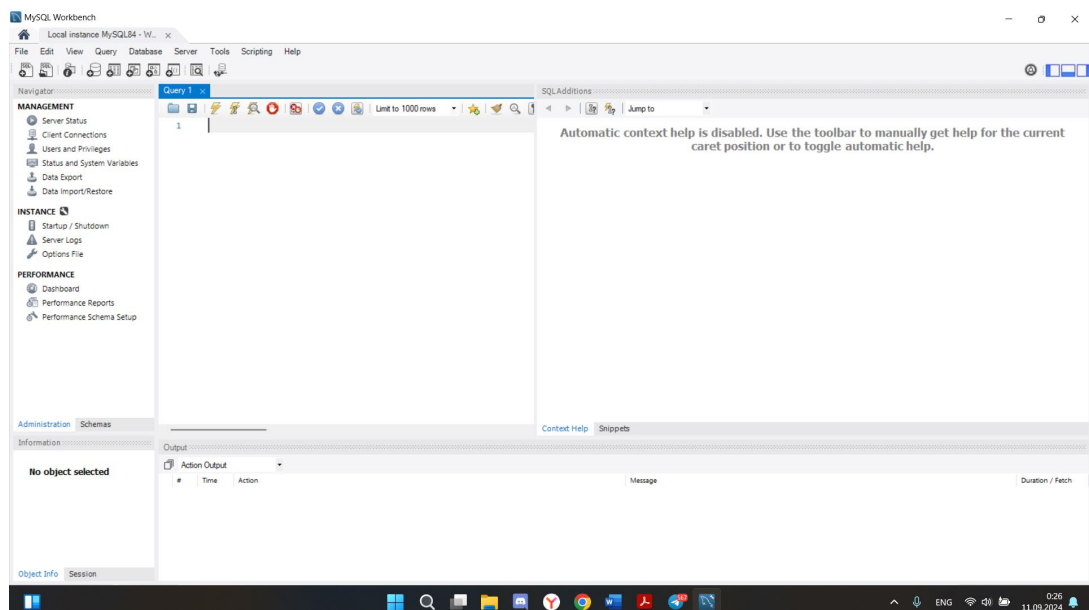


Рисунок 14 – Интерфейс взаимодействия с сервером

Был осуществлён переход к таблицам и их вывод на экран. Для вывода нажал на таблицу правой клавишей мыши, и выбрал Send to SQL Editor, далее

## Select All Statement.

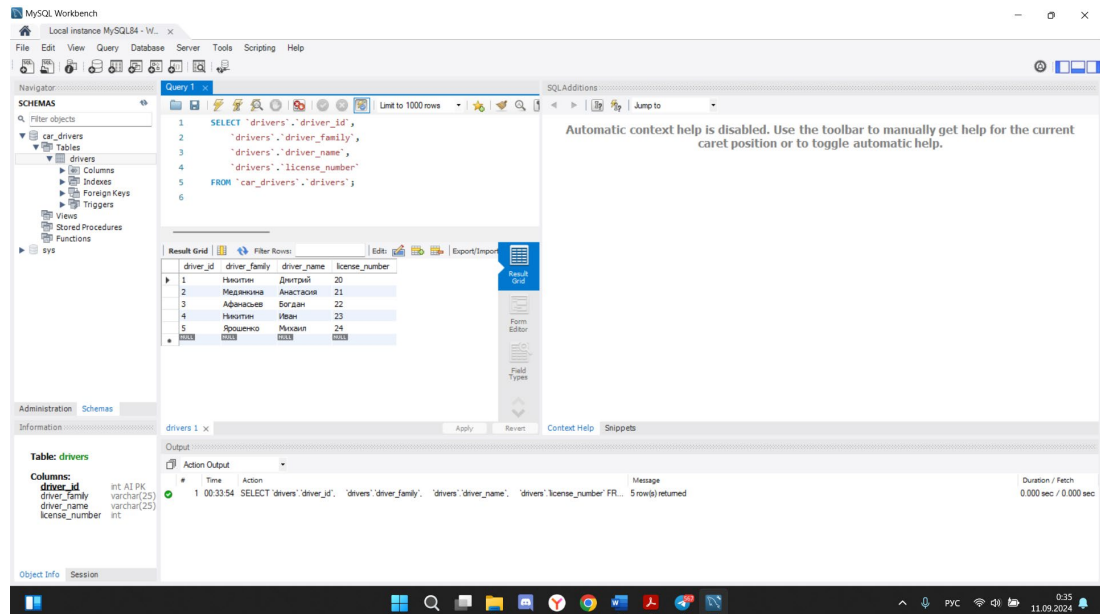


Рисунок 15 – Таблица car\_drivers

Далее я приступаю к созданию второй базы данных под названием cars с помощью команды `CREATE DATABASE IF NOT EXISTS cars;`.

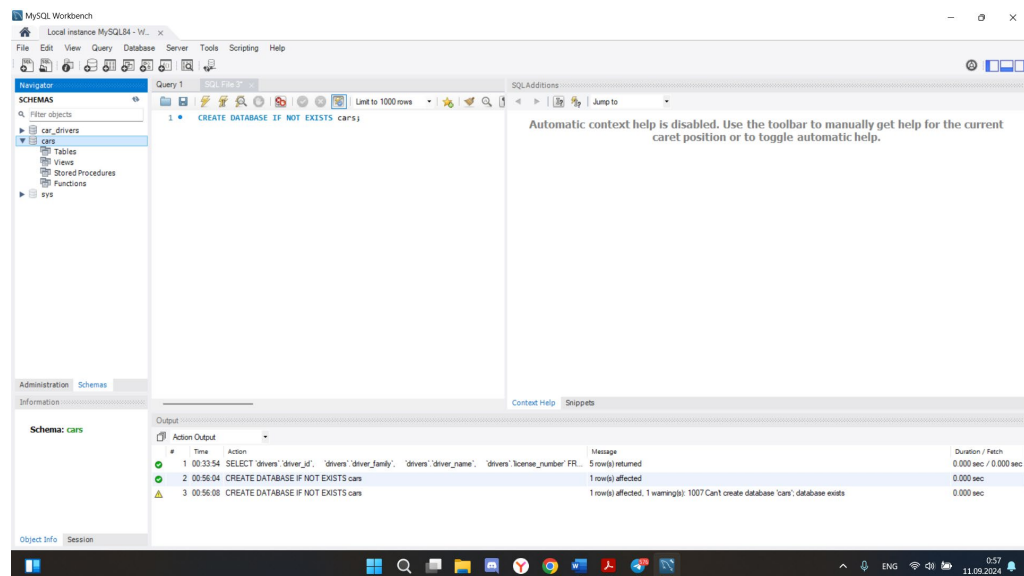


Рисунок 16 – Создание базы данных

С помощью команды `CREATE TABLE `cars`.`cars` ( `id_cars` INT NOT NULL AUTO_INCREMENT, `brand` VARCHAR(45) NULL, `number` VARCHAR(6) NULL, `year` INT NULL, PRIMARY KEY (`id_cars`));` создаём вторую таблицу.

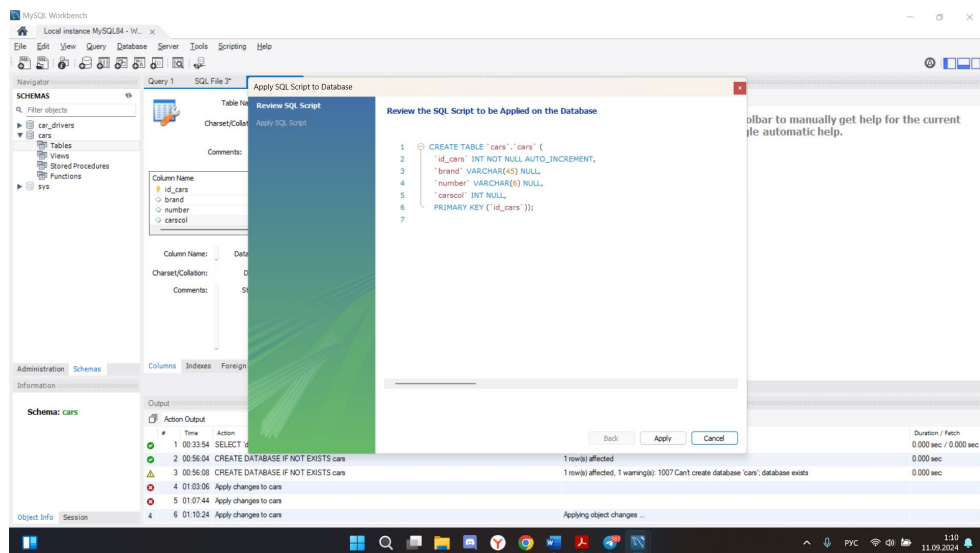


Рисунок 17 – Процесс создания второй базы данных

Разумеется, появляется интерес удалить только что созданную базу данных. Не зря же я записал весь код в файл.

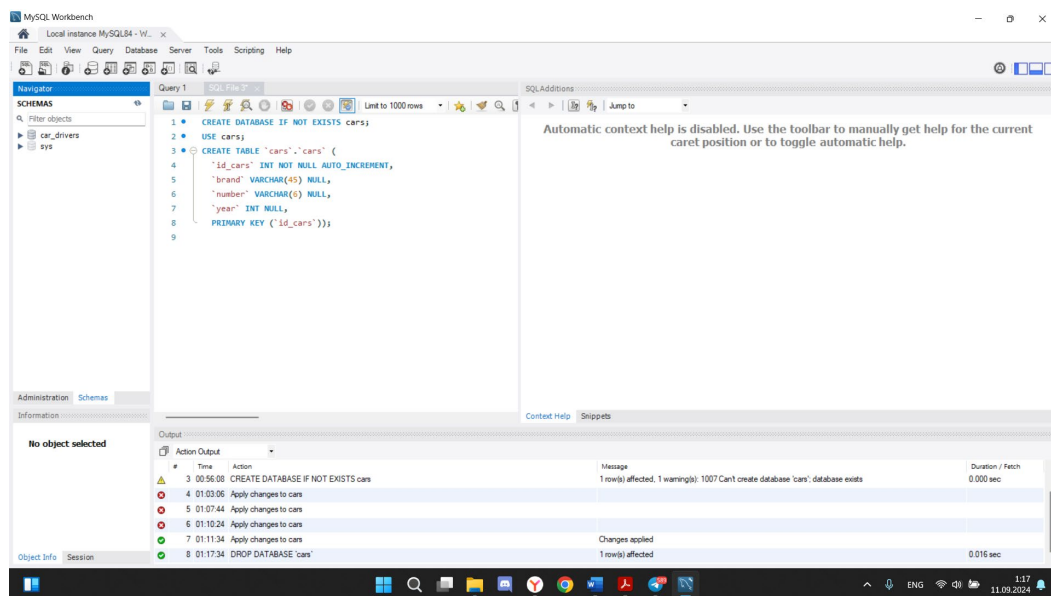


Рисунок 18 – База данных была удалена

После этого я получаю информацию о всё ещё существующей базе данных.

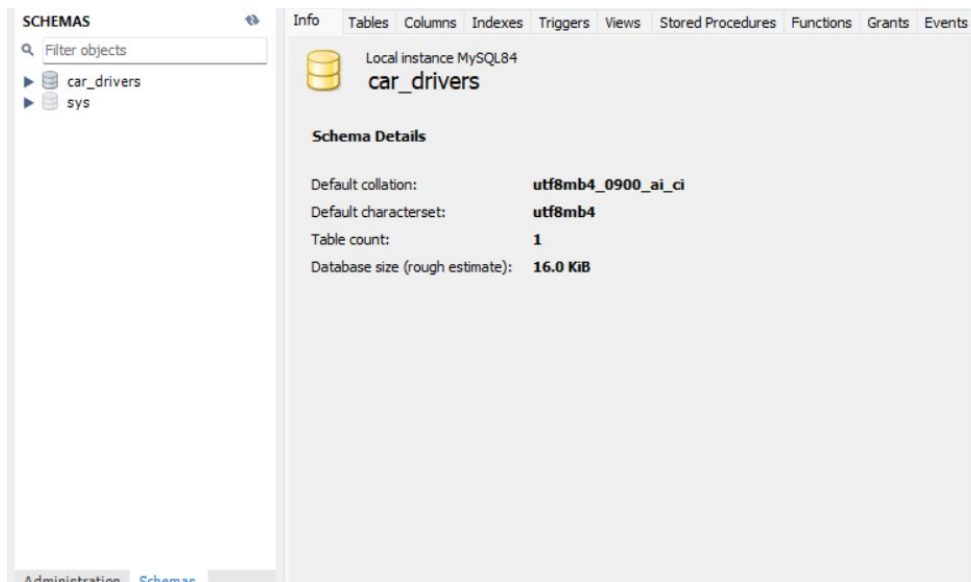


Рисунок 19 – Информация о не удалённой базе данных

После этого я восстанавливаю базу данных cars исполнением скрипта.

```
CREATE DATABASE IF NOT EXISTS cars;
USE cars;
CREATE TABLE `cars`.`cars` (
  `id_cars` INT NOT NULL AUTO_INCREMENT,
  `brand` VARCHAR(45) NULL,
  `number` VARCHAR(6) NULL,
  `year` INT NULL,
  PRIMARY KEY (`id_cars`));
```

Далее приступаю к заполнению данной базы данных.

```
С помощью команды INSERT INTO `cars`.`cars`
(`brand`,
`number`,
`year`)
VALUES
("Gazel Next",
"C121PY",
```

2020) и ей подобных было добавлено 5 строк через интерфейс графического редактора.

	id_cars	brand	number	year
▶	1	Toyota Camry	A666AA	2024
	2	Toyota Land Cruiser	X777XX	2024
	3	Lada Granta	M123OC	2022
	4	Lada Niva 4x4	B158OP	2003
	5	Gazel Next	C121PY	2020
•	NULL	NULL	NULL	NULL

Рисунок 20 – Итоговая таблица во второй базе данных

6. Вывод:

В ходе лабораторной работы была проведена установка и настройка локального сервера MySQL, а также реализованы однотабличные базы данных с использованием консольного и графического клиентов. Основные этапы включали установку сервера, создание баз данных и таблиц, а также выполнение операций с данными.

Сначала была установлена система управления базами данных MySQL на локальный компьютер. Процесс установки включал скачивание инсталлятора, настройку параметров конфигурации и создание пользователей. После успешной установки сервер был проверен на работоспособность.

Далее работа велась с консольным клиентом MySQL Command Line Client. Этот этап включал создание баз данных и таблиц, а также выполнение операций с данными: добавление, изменение и удаление записей. Консольный клиент предоставляет удобные инструменты для выполнения сложных запросов и управления данными.

Следующий этап включал работу с графическим клиентом MySQL Workbench. Через него также были созданы базы данных и таблицы, выполнены операции с данными. Графический интерфейс упрощает работу и делает её удобной для пользователей, предпочитающих визуальные инструменты.

В рамках задания были созданы однотабличные базы данных для хранения информации о водителях и машинах. В таблицы были добавлены записи, выполнены операции по извлечению данных для анализа.

Выполнение работы позволило получить навыки установки и настройки MySQL, а также работы с клиентами для управления базами данных. Эти навыки важны для разработчиков и администраторов, так как позволяют эффективно управлять данными. MySQL предоставляет удобную и безопасную среду для разработки и тестирования приложений.