

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ АЭРОКОСМИЧЕСКОГО
ПРИБОРОСТРОЕНИЯ»

КАФЕДРА №33

Отчет защищен с оценкой _____

Преподаватель

Проф., д.т.н.

Н.Н. Мошак

должность, уч. степень,
звание

подпись, дата

инициалы,
фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №3
«АДМИНИСТРИРОВАНИЕ И НАСТРОЙКА ПОЛИТИКИ БЕЗОПАСНОСТИ СЕРВЕРА
РЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ»

по курсу: Безопасность информационных систем

Студент гр. №

3932

И.К. Лобач

номер
группы

подпись,
дата

инициалы,
фамилия

Санкт-Петербург 2022

Цель работы: изучить команды MySQL и систему привилегий (privilege system). Научиться устанавливать и администрировать SQL-сервер на примере сервера MySQL, а также настраивать его параметры безопасности. Используемое программное обеспечение: ОС версии не ниже Windows 7, ПО сервера MySQL.

Основные сведения

1.1 Реляционные базы данных

В реляционных базах данных данные собраны в таблицы, которые в свою очередь состоят из столбцов и строк, на пересечении которых расположены ячейки. Запросы к таким базам данных возвращает таблицу, которая повторно может участвовать в следующем запросе.

Модель реляционной базы данных представляет данные в виде таблиц, разбитых на строки и столбцы, на пересечении которых находятся данные. Кратко особенности реляционной базы данных можно описать следующим образом:

- Данные хранятся в таблицах, состоящих из столбцов и строк;
- На пересечении каждого столбца и строчки стоит в точности одно значение;
- У каждого столбца есть своё имя, которое служит его названием, и все значения в одном столбце имеют один тип.
- Столбцы располагаются в определённом порядке, который определяется при создании таблицы, в отличие от строк, которые располагаются в произвольном порядке. В таблице может не быть не одной строчки, но обязательно должен быть хотя бы один столбец;
- Запросы к базе данных возвращают результат в виде таблиц, которые тоже могут выступать как объект запросов.

Для работы с базами данных используется язык SQL. MySQL — это система управления реляционными базами данных

1.2 Некоторые команды MySQL

1.2.1 Создание базы данных

CREATE DATABASE database_name

database_name - имя, которое будет присвоено создаваемой базе данных.

1.2.2 Удаление базы данных

DROP DATABASE database_name

database_name - задает имя базы данных, которую необходимо удалить.

1.2.3 Создание таблицы

CREATE TABLE table_name(column_name1 type, column_name2 type,...)

table_name - имя новой таблицы, column_name - имена колонок (полей), которые будут присутствовать в создаваемой таблице, type - определяет тип данных создаваемой колонки.

1.2.4 Удаление таблицы

DROP TABLE table_name

table_name - имя удаляемой таблицы.

1.2.5 Вставка записи

INSERT INTO table_name(field_name1, field_name2,...) values('content1', 'content2',...)

Данная команда добавляет в таблицу table_name запись, у которой поля, обозначенные как field_nameN, установлены в значение contentN.

1.2.6 Поиск записей

*SELECT * FROM table_name WHERE (выражение) [order by field_name [desc][asc]]*

Эта команда ищет все записи в таблице table_name, которые удовлетворяют выражению выражение. Если записей несколько, то при указанном предложении order by они будут отсортированы по тому полю, имя которого записывается правее этого ключевого слова (если задано слово desc, то упорядочивание происходит в обратном порядке). В предложении order by могут также задаваться несколько полей. Особое значение имеет символ *. Он предписывает, что из отобранных записей следует извлечь все поля, когда будет выполнена команда получения выборки. С другой стороны, вместо звездочки можно через запятую непосредственно перечислить имена полей, которые требуют извлечения.

1.3 Настройка параметров безопасности

Основой системы безопасности является система привилегий (privilege system), позволяющая очень гибко управлять правами доступа как к управлению сервером, так и к отдельным базам, таблицам и полям таблиц, а также встроенным функциям и хранимым процедурам.

GRANT [тип прав] ON [имя базы данных].[имя таблицы] TO 'имя пользователя'@'тип доступа на сервер';

Нужно заменить значение «тип прав» на тот вид прав доступа, который необходимо предоставить новому пользователю.

В MySQL есть несколько типов прав доступа, некоторые из них описаны ниже:

- CREATE – позволяет пользователям создавать базы данных/таблицы
- SELECT – позволяет пользователям делать выборку данных
- INSERT – позволяет пользователям добавлять новые записи в таблицы
- UPDATE – позволяет пользователям изменять существующие записи в таблицах
- DELETE – позволяет пользователям удалять записи из таблиц
- DROP – позволяет пользователям удалять записи в базе данных/таблицах

Установка и администрирование

2.1 Установка MySQL

Дистрибутив можно скачать с www.mysql.com. Документация (на английском языке) расположена по адресу <http://dev.mysql.com/doc/>. Для установки необходимо иметь административные права.

Перед установкой необходимо установить Microsoft Net Framework 4.0, использующийся в программах администрирования в среде Microsoft Windows.

При выборе обязательно должны быть установлены:

- В группе MySQL Server (MySQL Server, Client Programs, Server Data Files)
- В группе Applications – MySQL Notifier (утилита для управления сервером) и MySQL Workbench ()
- Группу Documentation – рекомендуется установить полностью (документация на английском языке для работы с сервером MySQL Server 8.0.18 - x64)

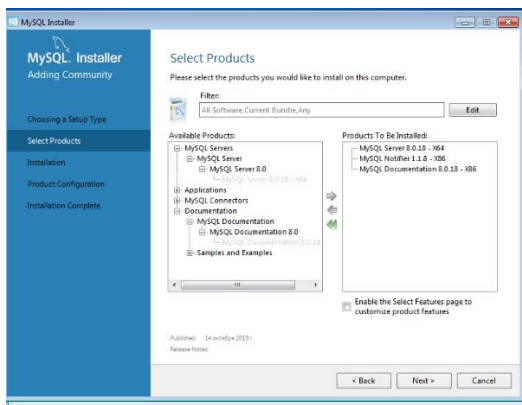


Рисунок 1 - Выбор для установки

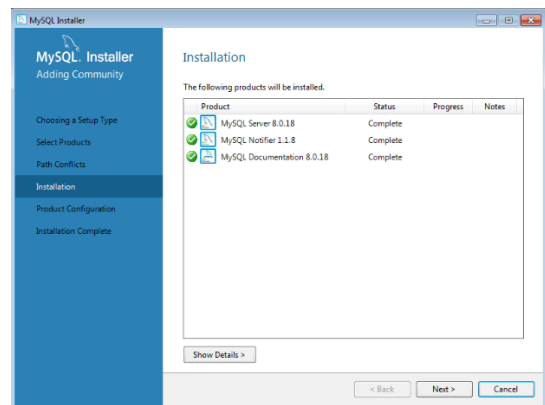


Рисунок 2 - Выбор для установки

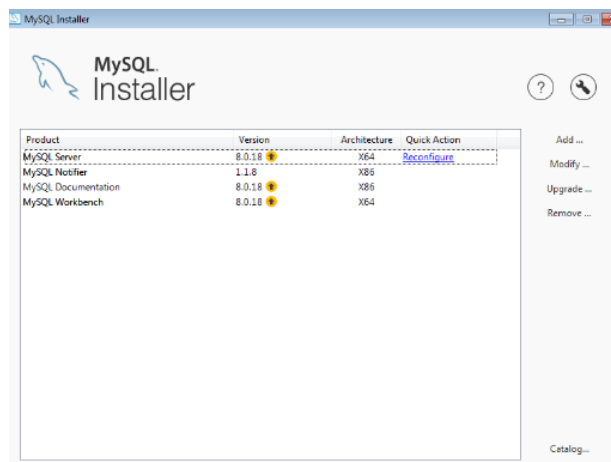


Рисунок 3 - Выбор для установки

Далее окно настройки серверной части. Рекомендуется выбрать конфигурацию *Development Computer* – этот тип установки предназначен для разработки и тестирования сайтов, в этом случае ресурсы компьютера будут подвергаться минимальной нагрузке.:

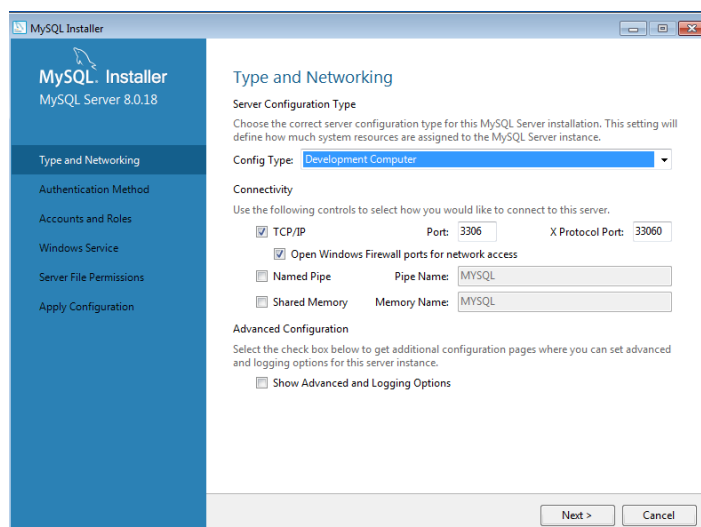


Рисунок 4 - Настройка серверной части

Далее создается главный администратор сервера с именем *root*. Для него необходимо придумать пароль, который должен соответствовать требованиям к классу безопасности 1Д, т.е. не менее 6 буквенно-цифровых символов. Пароль 8w%SiUW6 удовлетворяет данным условиям.

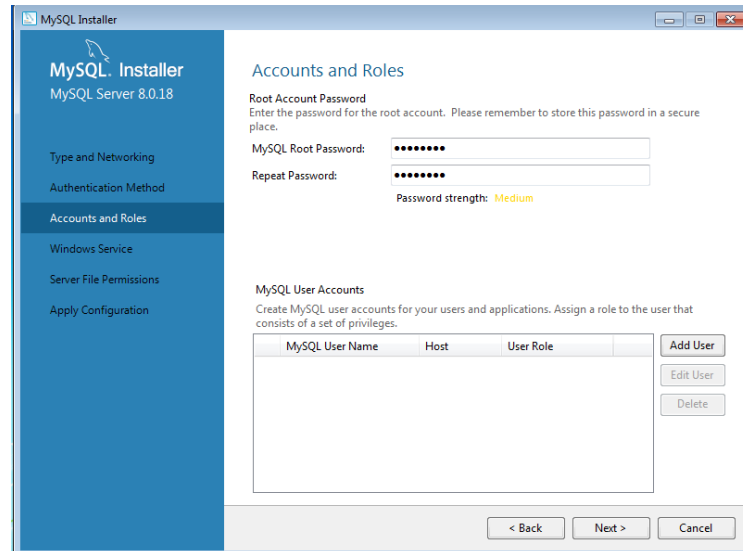


Рисунок 5 - Создание пользователя *root*

2.2 Создание базы данных MySQL

На данный момент имеется настроенный сервер MySQL Server и утилита для взаимодействия с ним MySQL Notifier.

Рассмотрим имеющиеся по умолчанию базы данных. Для этого введем команду *show databases* и увидим:

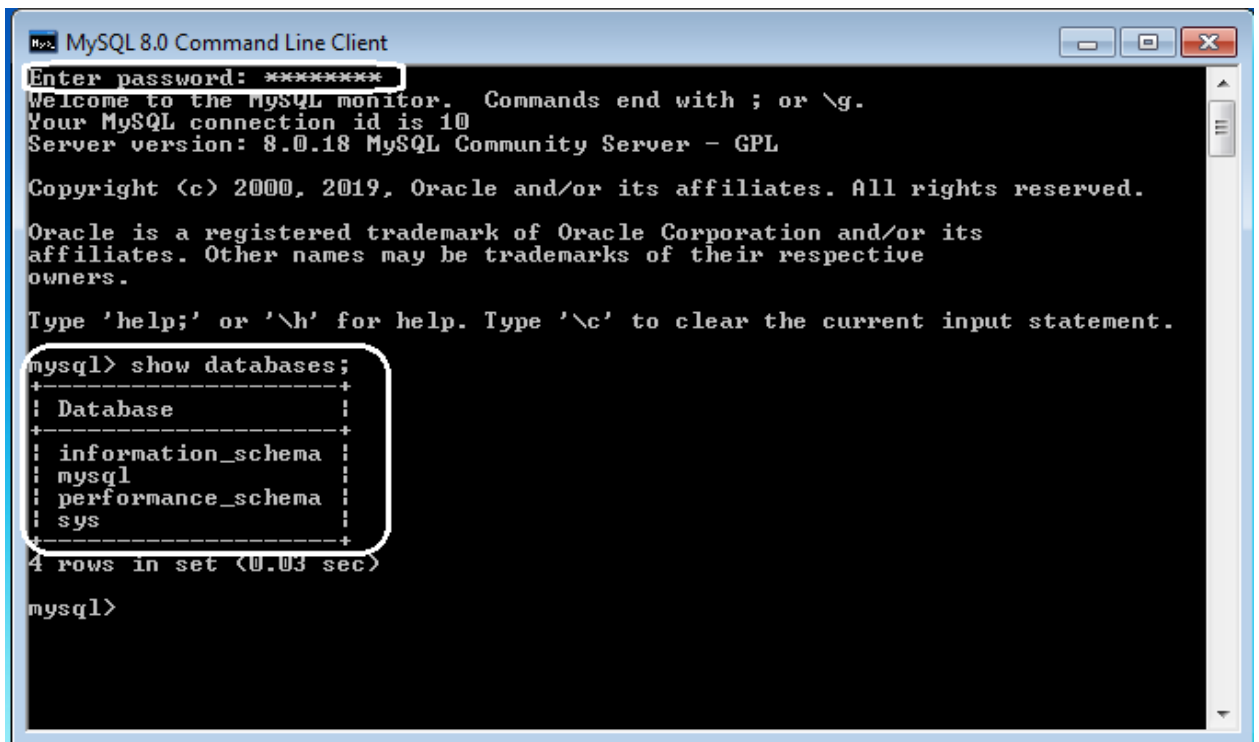


Рисунок 6 - Просмотр существующих баз данных

Теперь создадим новую базу данных с названием *laboratoryWork3* для этого используем команду ***create database laboratoryWork3*** и посмотрим обновленный список баз данных с помощью команды ***show databases***:

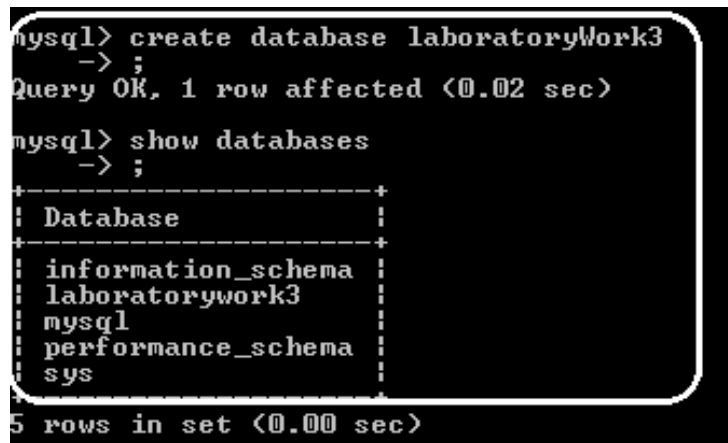


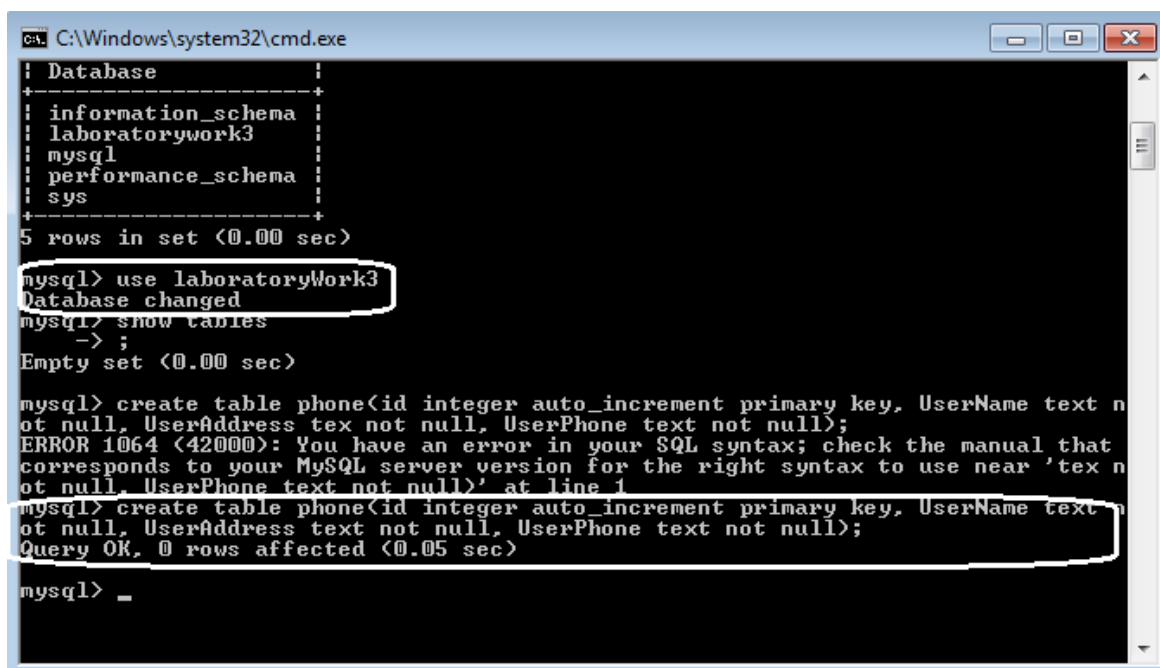
Рисунок 7 - Создание новой базы данных

Как видно, новая база данных была создана успешно.

2.3 Создание новой таблицы и добавление записей

Создадим новую таблицу *phone* в базе данных *laboratoryWork3*, которая на данный момент не содержит таблиц. Таблица будет содержать следующие поля: *UserName* – тип данных Text, *UserAddress* – тип данных Text, *UserPhone* – тип данных Text. Также добавим дополнительную колонку, в которой пропишем автоматический счетчик записей

auto_increment primary key типа Integer – это значение будет увеличиваться с каждой новой записью и позволит более гибко оперировать содержимым таблицы (например, для связывания между собой нескольких таблиц). Для этого используем команду ***create table phone (id integer auto_increment primary key, UserName text not null, UserAddress text not null, UserPhone text not null)***:



```
C:\Windows\system32\cmd.exe
+-----+
| Database |
+-----+
| information_schema |
| laboratorywork3    |
| mysql              |
| performance_schema |
| sys                 |
+-----+
5 rows in set (0.00 sec)

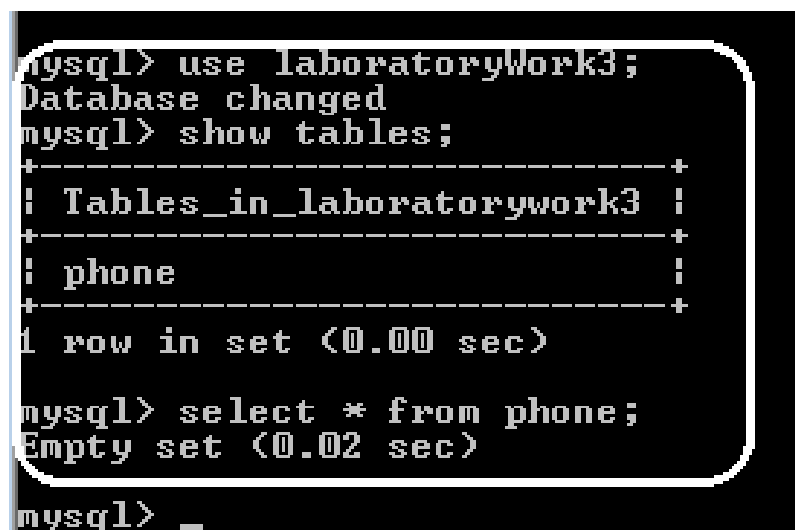
mysql> use laboratoryWork3
Database changed
mysql> show tables
+-----+
| Tables_in_laboratorywork3 |
+-----+
|                             |
+-----+
Empty set (0.00 sec)

mysql> create table phone(id integer auto_increment primary key, UserName text n
ot null, UserAddress tex not null, UserPhone text not null);
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MySQL server version for the right syntax to use near 'tex n
ot null, UserPhone text not null)' at line 1
mysql> create table phone(id integer auto_increment primary key, UserName text n
ot null, UserAddress text not null, UserPhone text not null);
Query OK, 0 rows affected (0.05 sec)

mysql> _
```

Рисунок 8 - Создание новой таблицы

Проверим, что таблица была успешно создана. Для этого используем команду ***show tables***:



```
mysql> use laboratoryWork3;
Database changed
mysql> show tables;
+-----+
| Tables_in_laboratorywork3 |
+-----+
| phone                      |
+-----+
1 row in set (0.00 sec)

mysql> select * from phone;
Empty set (0.02 sec)

mysql> _
```

Рисунок 9 - Проверка создания новой таблицы

Далее необходимо добавить несколько записей. Для этого используется команда ***insert into phone (UserName, UserAddress, UserPhone) values ('Iya, 'Moscow', '11111')***:


```
mysql> insert into phone(UserName, UserAddress, UserPhone) values ('Vika','Moscow','22222');
Query OK, 1 row affected (0.01 sec)

mysql> insert into phone(UserName, UserAddress, UserPhone) values ('Sveta','Moscow','33333');
Query OK, 1 row affected (0.01 sec)

mysql> insert into phone(UserName, UserAddress, UserPhone) values ('Ivan','Moscow','44444');
Query OK, 1 row affected (0.01 sec)

mysql> insert into phone(UserName, UserAddress, UserPhone) values ('Vasya','Moscow','55555');
Query OK, 1 row affected (0.01 sec)
```

Рисунок 10 - Добавление новых записей

Проверим, что записи добавлены. Для этого используем команду *select * from phone*:

```
mysql> select * from phone;
+----+-----+-----+-----+
| id | UserName | UserAddress | UserPhone |
+----+-----+-----+-----+
| 1  | Iya      | Moscow     | 11111     |
| 2  | Vika     | Moscow     | 22222     |
| 3  | Sveta    | Moscow     | 33333     |
| 4  | Ivan     | Moscow     | 44444     |
| 5  | Vasya    | Moscow     | 55555     |
+----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql>
```

Рисунок 11 - Просмотр всех записей

2.4 Создание нового пользователя и настройка его прав доступа

После первоначальной настройки сервера MySQL, будет предоставлено имя пользователя и пароль, причем эти начальные учётные данные дают привилегии «root-доступа», то есть привилегированного пользователя.

Пользователь с правами доступа root имеют полный доступ ко всем базам данных и таблицам внутри этих баз – но на предприятиях обычно требуется предоставить доступ к базе данных для работников, которым не требуется и не рекомендовано полное управление.

Создадим нового пользователя базы данных *laboratoryWork3*. Необходимо создание нового пользователя и предоставление ему неограниченных прав доступа, создание нового соединения и назначение специальных прав доступа для данного пользователя. Рассмотрим каждый из этих шагов.

2.4.1 Создание нового пользователя с неограниченными правами доступа

Создадим нового пользователя с именем **LocalUser** и паролем **password** базы данных **laboratoryWork3**. Данный пользователь будет локальным, так как нет необходимости предоставлять пользователю возможность войти в систему из любого АРМ. Для этого введем команду **create user 'LocalUser'@'localhost' identified by 'password'**:

```
mysql> create user 'LocalUser'@'localhost' identified by 'password'  
-> ;  
Query OK, 0 rows affected (0.02 sec)  
  
mysql>
```

Рисунок 12 - Создание нового пользователя

Назначим созданному пользователю неограниченные права. Для этого введем команду **grant all privileges on laboratoryWork3.* to 'LocalUser'@'localhost'**. Проверим назначены ли права с помощью команды **show grants for 'LocalUser'@'localhost'**:

```
mysql> grant all privileges on laboratoryWork3.* to 'LocalUser'@'localhost';  
Query OK, 0 rows affected (0.01 sec)  
  
mysql> show grants for 'LocalUser'@'localhost';  
ERROR 1141 (42000): There is no such grant defined for user 'LocalUser' on host 'localhost'  
  
mysql> show grants for 'LocalUser'@'localhost';  
+-----+  
| Grants for LocalUser@localhost |  
+-----+  
| GRANT USAGE ON *.* TO 'LocalUser'@'localhost' |  
| GRANT ALL PRIVILEGES ON `laboratorywork3`.* TO 'LocalUser'@'localhost' |  
+-----+  
2 rows in set (0.00 sec)  
  
mysql>
```

Рисунок 13 - Назначение прав

2.4.2 Создание нового соединения

Чтобы запустить командную строку в качестве другого пользователя воспользуемся ранее установленной утилитой MySQL Workbench и создадим новое подключение. Для этого перейдем в MySQL Workbench и создадим новое подключение для пользователя **LocalUser**:

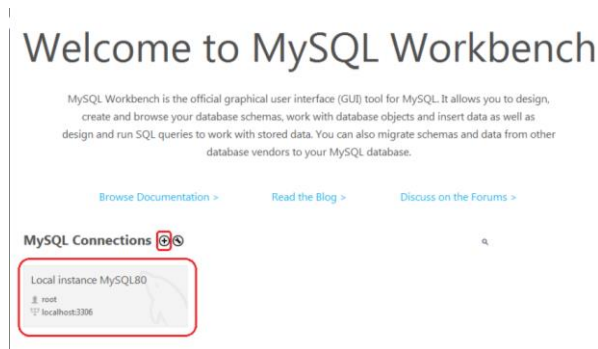


Рисунок 14 - Добавление соединения

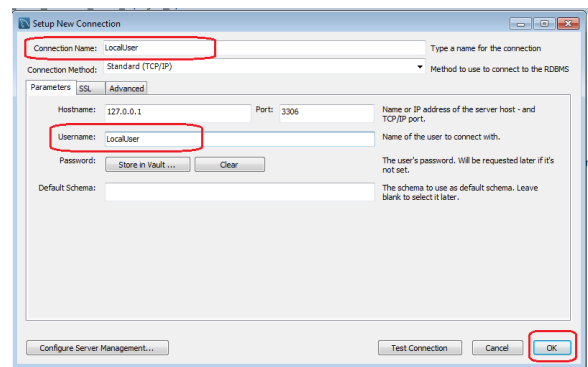


Рисунок 15 - Новое соединение

Welcome to MySQL Workbench

MySQL Workbench is the official graphical user interface (GUI) tool for MySQL. It allows you to design, create and browse your database schemas, work with database objects and insert data as well as design and run SQL queries to work with stored data. You can also migrate schemas and data from other database vendors to your MySQL database.

[Browse Documentation >](#) [Read the Blog >](#) [Discuss on the Forums >](#)

MySQL Connections

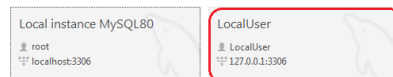


Рисунок 16 - Созданное соединение

Как видно подключение было установлено. Теперь можно запустить командную строку в качестве не привилегированного пользователя *LocalUser*. Для этого кликнем на подключение правой кнопкой мыши и перейдем в **Start Command Line Client**:



Рисунок 17 - Новое подключение

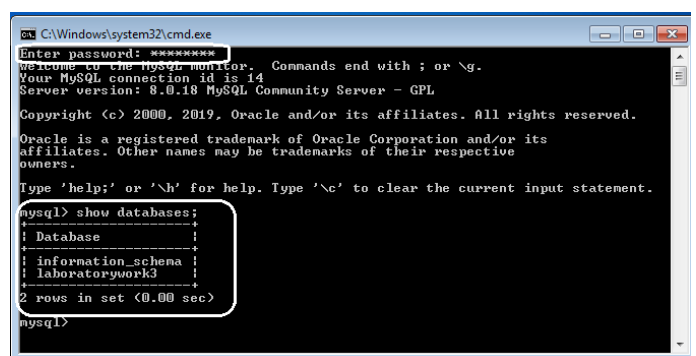


Рисунок 18 - Подключение пользователя LocalUser

2.4.3 Назначение специальных прав доступа

Разрешим пользователю только просмотр записей *остальных баз данных*, имеющихся в файловой системе SQL: их изменение недопустимо для обычного непривилегированного пользователя, потому что может привести к искажению и даже потере важных данных. Для этого используем команду ***grant SELECT on *.* to 'LocalUser'@'localhost'***:

```
mysql> grant select on *.* to 'LocalUser'@'localhost';  
Query OK, 0 rows affected (0.01 sec)  
  
mysql> _
```

Рисунок 19 – Разрешение просмотра баз данных

Установим права SELECT и INSERT на базу данных *laboratoryWork3* для таблицы *phone*. Для начала необходимо удалить ранее установленные привилегии. Для этого используем команду ***revoke all privileges on *.* from 'LocalUser'@'localhost'***. Затем назначим новые. Для этого используем команды ***grant SELECT, INSERT on laboratoryWork3.phone to 'LocalUser'@'localhost'***:

```
mysql> create user 'LocalUser'@'localhost' identified by 'password';  
Query OK, 0 rows affected (0.02 sec)  
  
mysql> grant select on *.* to 'LocalUser'@'localhost';  
Query OK, 0 rows affected (0.01 sec)  
  
mysql> grant select,insert on laboratoryWork3.phone to 'LocalUser'@'localhost';  
Query OK, 0 rows affected (0.02 sec)  
  
mysql> _
```

Рисунок 20 - Установка прав для пользователя

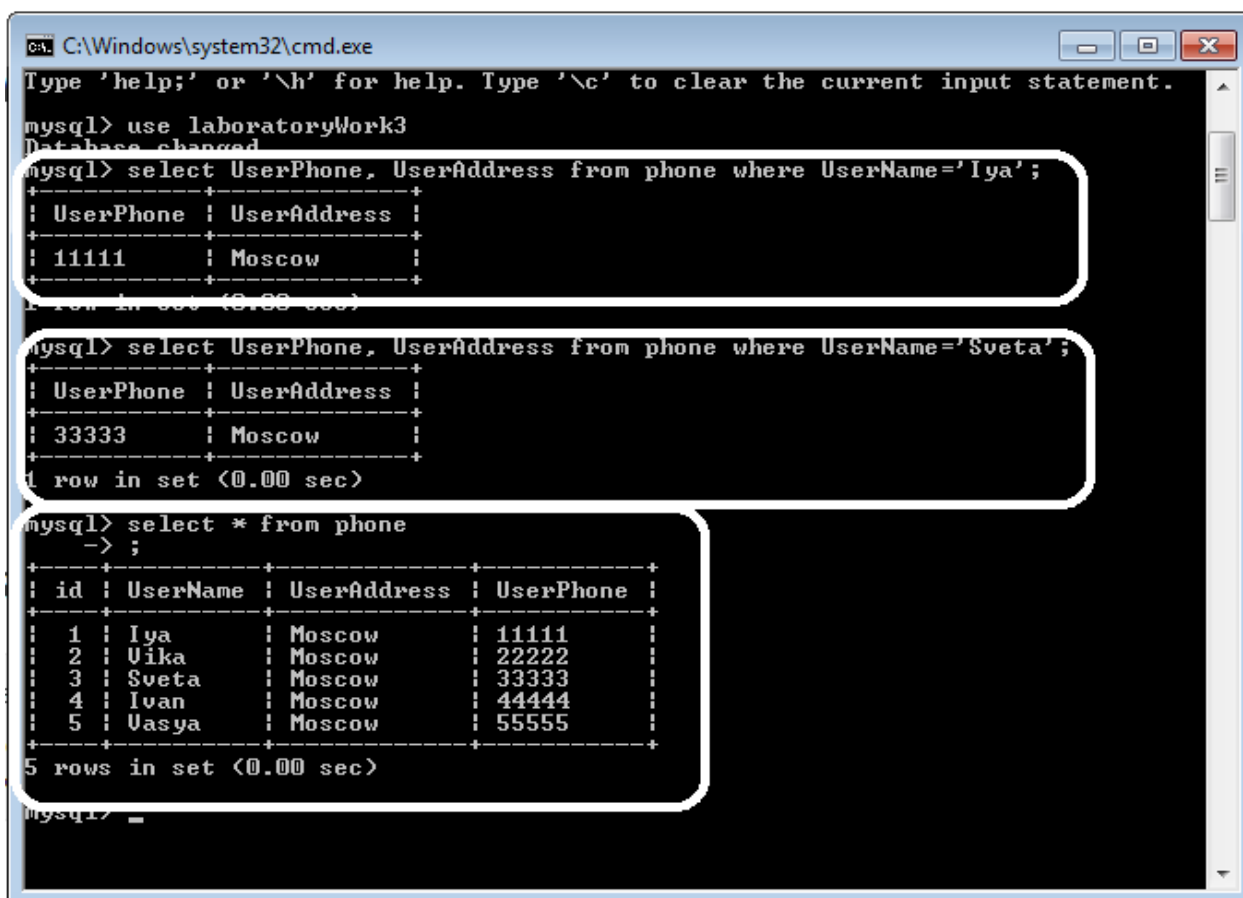
Проверим привилегии для данного пользователя. Для этого используем команду ***show grants for 'LocalUser'@'localhost'***:

```
mysql> show grants for 'LocalUser'@'localhost';  
+-----+  
| Grants for LocalUser@localhost |  
+-----+  
| GRANT SELECT ON *.* TO 'LocalUser'@'localhost' |  
| GRANT SELECT, INSERT ON 'laboratorywork3`.`phone` TO 'LocalUser'@'localhost' |  
+-----+
```

Рисунок 21 - Проверка установленных прав

2.5 Выборка адреса и телефона для указанного UserName

Необходимо сделать выборку адреса и телефона для указанного имени в базе данных *laboratoryWork3* в таблице *phone*. Такую выборку, согласно установленным правам, может выполнить и созданный ранее пользователь *LocalUser*. Откроем командную строку в качестве этого пользователя через MySQL Workbench. Затем перейдем в нужную базу данных и создадим требуемую выборку с помощью команды *select UserAddress, UserPhone from phone where UserName='Iya'*. Также введем команду *select * from phone*, чтобы убедиться в том, что выборка верная:



```
C:\Windows\system32\cmd.exe
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use laboratoryWork3
Database changed
mysql> select UserPhone, UserAddress from phone where UserName='Iya';
+-----+-----+
| UserPhone | UserAddress |
+-----+-----+
| 11111     | Moscow     |
+-----+-----+
1 row in set (0.00 sec)

mysql> select UserPhone, UserAddress from phone where UserName='Sveta';
+-----+-----+
| UserPhone | UserAddress |
+-----+-----+
| 33333     | Moscow     |
+-----+-----+
1 row in set (0.00 sec)

mysql> select * from phone
-> ;
+----+-----+-----+-----+
| id | UserName | UserAddress | UserPhone |
+----+-----+-----+-----+
| 1  | Iya      | Moscow     | 11111     |
| 2  | Oika     | Moscow     | 22222     |
| 3  | Sveta    | Moscow     | 33333     |
| 4  | Ivan     | Moscow     | 44444     |
| 5  | Vasya    | Moscow     | 55555     |
+----+-----+-----+-----+
5 rows in set (0.00 sec)
```

Рисунок 22 - Выборка по имени

2.6 Выборка с сортировкой по полю UserName в алфавитном порядке

Необходимо сделать выборку с сортировкой по атрибуту *UserName* в базе данных *laboratoryWork3* в таблице *phone*. Такую выборку, согласно установленным правам, может выполнить созданный ранее пользователь *LocalUser*. Откроем командную строку в качестве этого пользователя через MySQL Workbench. Затем перейдем в нужную базу данных и создадим требуемую выборку с сортировкой по возрастанию с помощью *select * from phone order by UserName asc* и выборку с сортировкой по убыванию с помощью *select * from phone order by UserName desc*:

```
C:\Windows\system32\cmd.exe

+----+-----+-----+-----+
| 1 | Iya   | Moscow | 11111 |
| 2 | Vika  | Moscow | 22222 |
| 3 | Sveta | Moscow | 33333 |
| 4 | Ivan  | Moscow | 44444 |
| 5 | Uasya | Moscow | 55555 |
+----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> select * from phone order by UserName asc;
+----+-----+-----+-----+
| id | UserName | UserAddress | UserPhone |
+----+-----+-----+-----+
| 4 | Ivan    | Moscow    | 44444    |
| 1 | Iya     | Moscow    | 11111    |
| 3 | Sveta   | Moscow    | 33333    |
| 5 | Uasya   | Moscow    | 55555    |
| 2 | Vika    | Moscow    | 22222    |
+----+-----+-----+-----+
5 rows in set (0.01 sec)

mysql> select * from phone order by UserName desc;
+----+-----+-----+-----+
| id | UserName | UserAddress | UserPhone |
+----+-----+-----+-----+
| 2 | Vika    | Moscow    | 22222    |
| 5 | Uasya   | Moscow    | 55555    |
| 3 | Sveta   | Moscow    | 33333    |
| 1 | Iya     | Moscow    | 11111    |
| 4 | Ivan    | Moscow    | 44444    |
+----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> _
```

Рисунок 23 - Результаты выборки

2.7 Удаление созданного пользователя, таблицы и базы данных

Согласно настройке прав, пользователь LocalUser не может удалять ни таблицы, ни базы данных, ни других пользователей. Убедимся в этом, используя команду *drop database laboratoryWork3* и *drop table phone*:

```
mysql> drop database laboratoryWork3;
ERROR 1044 (42000): Access denied for user 'LocalUser'@'localhost' to database 'laboratorywork3'
mysql> use laboratoryWork3;
Database changed
mysql> show tables;
+-----+
| Tables_in_laboratorywork3 |
+-----+
| phone                     |
+-----+
1 row in set (0.00 sec)

mysql> drop table phone;
ERROR 1142 (42000): DROP command denied to user 'LocalUser'@'localhost' for table 'phone'
mysql> _
```

Рисунок 24 - Попытка удаления

Как видно, настроенные права не позволяют пользователю удалять таблицы, базы данных и пользователей. Поэтому далее все команды будут осуществляться привилегированным пользователем *root*.

2.7.1 Удаление таблицы

Необходимо удалить таблицу *phone* базы данных *laboratoryWork3*. Для этого используем команду ***drop table phone***, предварительно перейдя в нужную базу данных командой ***use laboratoryWork3***. Для проверки корректности выполнения команды посмотрим таблицы в базе данных с помощью команды ***show tables***:

```
mysql> use laboratoryWork3
Database changed
mysql> show tables;
+-----+
| Tables_in_laboratorywork3 |
+-----+
| phone                      |
+-----+
1 row in set (0.00 sec)

mysql> drop table phone;
Query OK, 0 rows affected (0.04 sec)

mysql> show tables;
Empty set (0.00 sec)

mysql>
```

Рисунок 25 - Удаление таблицы

Как видно, таблица была успешно удалена.

2.7.2 Удаление базы данных

Необходимо удалить базу данных *laboratoryWork3*. Для этого используем команду ***drop database laboratoryWork3***. Для проверки корректности выполнения команды посмотрим таблицы в базе данных с помощью команды ***show databases***:

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| laboratorywork3    |
| mysql             |
| performance_schema |
| sys               |
+-----+
5 rows in set (0.00 sec)

mysql> drop database laboratoryWork3;
Query OK, 0 rows affected (0.04 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql         |
| performance_schema |
| sys          |
+-----+
4 rows in set (0.00 sec)

mysql>
```

Рисунок 26 - Удаление базы данных

Как видно, база данных была успешно удалена.

2.7.3 Удаление пользователя

Необходимо удалить пользователя *LocalUser*. Для этого предварительно необходимо удалить все привилегии с помощью команды *revoke all privileges on *.* from 'LocalUser'@'localhost'*, затем удалим самого пользователя, используя команду *drop user 'LocalUser'@'localhost'*. Чтобы проверить корректность выполнения команды, необходимо посмотреть весь список пользователей с помощью команды *select User, Host from mysql.user*:

```
mysql> select User,Host from mysql.user;
+-----+-----+
| User          | Host      |
+-----+-----+
| LocalUser     | localhost |
| mysql.infoschema | localhost |
| mysql.session | localhost |
| mysql.sys     | localhost |
| root          | localhost |
+-----+-----+
5 rows in set (0.01 sec)

mysql> revoke all privileges on *.* from 'LocalUser'@'localhost';
Query OK, 0 rows affected (0.03 sec)

mysql> show grants for 'LocalUser'@'localhost';
+-----+-----+
| Grants for LocalUser@localhost |
+-----+-----+
| GRANT USAGE ON *.* TO 'LocalUser'@'localhost' |
+-----+-----+
1 row in set (0.00 sec)

mysql> drop user 'LocalUser'@'localhost';
Query OK, 0 rows affected (0.02 sec)

mysql> select User,Host from mysql.user;
+-----+-----+
| User          | Host      |
+-----+-----+
| mysql.infoschema | localhost |
| mysql.session   | localhost |
| mysql.sys       | localhost |
| root           | localhost |
+-----+-----+
4 rows in set (0.00 sec)

mysql>
```

Рисунок 27 -Удаление пользователя

Как видно, пользователь успешно удален.

Выводы: в ходе выполнения лабораторной работы были изучены команды MySQL и система привилегий (privilege system). А также осуществлена установка и администрирование SQL-сервера на примере сервера MySQL, и настройка его параметров безопасности.

Была создана база данных *laboratoryWork3*. В данной базе данных была создана таблица *phone*, содержащая четыре поля: *UserName*, *UserAddress*, *UserPhone* и автоинкрементируемое поле *id*, которое позволит более гибко оперировать содержимым

таблицы (например, связывать между собой несколько таблиц). В данную таблицу были добавлены записи.

Был создан новый непривилегированный пользователь *LocalUser*. В целях обеспечения безопасности прав доступа к ресурсам сервера базы данных его привилегии были ограничены следующим образом: пользователю разрешен только просмотр записей остальных баз данных, имеющихся в файловой системе SQL и разрешено добавление и просмотр таблицы *phone*.

От лица пользователя *LocalUser* в таблице *phone* были осуществлены следующие выборки: выборка по полю *UserName* и выборка с сортировкой. Была произведена попытка удаления записей и таблицы, что позволило убедиться в том, что настроенные права действительно ограничивают действия пользователя.

От лица привилегированного пользователя была удалена таблица *phone*, база данных *laboratoryWork3* и пользователь *LocalUser*.