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

КАФЕДРА №51

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

доктор технич. наук, доцент		Н.Н. Мошак
Должность, уч. степень, звание	Подпись, дата	Инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 3

АДМИНИСТРИРОВАНИЕ И НАСТРОЙКА ПОЛИТИК БЕЗОПАСНОСТИ СЕРВЕРА РЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ

по курсу: БЕЗОПАСНОСТЬ ИНФОРМАЦИОННЫХ СИСТЕМ

Работу выполнил

студент группы	5712		В. Г. Корж
		Подпись, дата	Инициалы, фамилия

Санкт-Петербург
2020

1 Цель

Изучить команды MySQL и систему привилегий (privilege system). Научиться устанавливать и администрировать SQL-сервер на примере сервера MySQL, а также настраивать его параметры безопасности.

2 Теоретические сведения

2.1 Реляционные базы данных. Общие сведения

Для решения задачи длительного хранения и обработки информации в конце 60-х годов были разработаны специализированные программы, получившие название систем управления базами данных (СУБД). Чтобы унифицировать работу с СУБД, был разработан структурированный язык запросов (SQL), который представляет собой язык управления именно реляционными базами данных.

Существуют следующие разновидности баз данных:

1. Иерархические — базы данных, основанные на древовидной структуре хранения информации. В этом смысле иерархические базы данных очень напоминают файловую систему компьютера

2. Реляционные — базы данных, данные в которых собраны в таблицы, которые в свою очередь состоят из столбцов и строк, на пересечении которых расположены ячейки. Запросы к таким базам данных возвращают таблицу, которая повторно может участвовать в следующем запросе. Данные в одних таблицах, как правило, связаны с данными других таблиц, откуда и произошло название "реляционные"

3. Объектно-ориентированные — базы данных, в которых данные хранятся в виде объектов. С объектно-ориентированными базами данных удобно работать, применяя объектно ориентированное программирование. Однако, на сегодняшний день такие базы данных еще не достигли популярности реляционных, поскольку пока значительно уступают им в производительности

4. Гибридные — это СУБД, которые совмещают в себе возможности реляционных и объектно-ориентированных баз данных. Эти модели характеризуются простой структурой данных, удобным для пользователя табличными представлением и возможностью использования

формального аппарата алгебры отношений и реляционного исчисления для обработки данных

Модель реляционной базы данных представляет данные в виде таблиц, разбитые 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 свое имя, которое служит его названием, и все значения в одном столбце имеют один тип
- Столбы располагаются в определённом порядке, который определяется при создании таблицы, в отличие от строк, которые располагаются в произвольном порядке. В таблице может не быть ни 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. Он предназначен для манипуляции данными, которые хранятся в Системах управления реляционными базами данных. SQL является общим языком запросов для нескольких баз данных различных типов.

2.2 Базы данных MySQL. Общие сведения

MySQL, которая является СУРБД с открытым исходным кодом, доступна для загрузки на сайте MySQL.com. Характеристики программного обеспечения MySQL:

1. Написан на C и C++. Протестирован на множестве различных компиляторов.
2. Работает на различных аппаратных платформах и разных операционных системах.
3. Высокая производительность за счёт максимально оптимизированного кода, эффективной системы распределения памяти и продуманной системы дисковых таблиц.

4. Способность работать с очень большими базами данных (десятки и сотни миллионов записей).

5. Возможность кластеризации серверов и распределения обработки информации между серверами.

6. Является системой клиент-сервер, которая содержит многопоточный SQL сервер, обеспечивающий поддержку различных вычислительных машин баз данных, а также несколько различных клиентских программ и библиотек, средства администрирования и широкий спектр программных интерфейсов.

7. Система безопасности основана на привилегиях и паролях с возможностью верификации с удалённого компьютера, за счет чего обеспечивается гибкость и безопасность. Пароли при передачи по сети при соединении с сервером шифруются. Клиенты могут соединяться с MySQL, используя сокеты TCP/IP, сокеты UNIX или именованные каналы.

2.3 Краткий обзор команд MySQL

1. Создание бд выполняется с помощью команды CREATE DATABASE. CREATE DATABASE db_name

2. Для удаления базы данных используется команда DROP DATABASE. DROP DATABASE db_name

3. Создание таблицы производится командой CREATE TABLE.
   CREATE TABLE table_name (column_name1 type, column_name2 tupe, ...)

4. Удаление таблицы производится командой DROP TABLE. DROP TABLE table_name

5. Вставка записи осуществляется командой INSERT INTO.
   INSERT INTO table_name(field_name1, field_name2, ...) values(’content1’, ’content2’, ...)

6. Поиск записей осуществляется командой SELECT.
   SELECT * FROM table_name WHERE (выражение) [order by filed_name [desc][asc]]

3 Ход работы

3.1 Установка MySQL

Дистрибутив был скачан с сайта *mysql.com*. 1. Окна настройки серверной части. Необходимо выбрать конфигурацию ’Development Machine’ — этот тип установки предназначен для разработки и тестирования сайтов, в этом случае ресурсы компьютера будут подвергаться минимальной нагрузке.

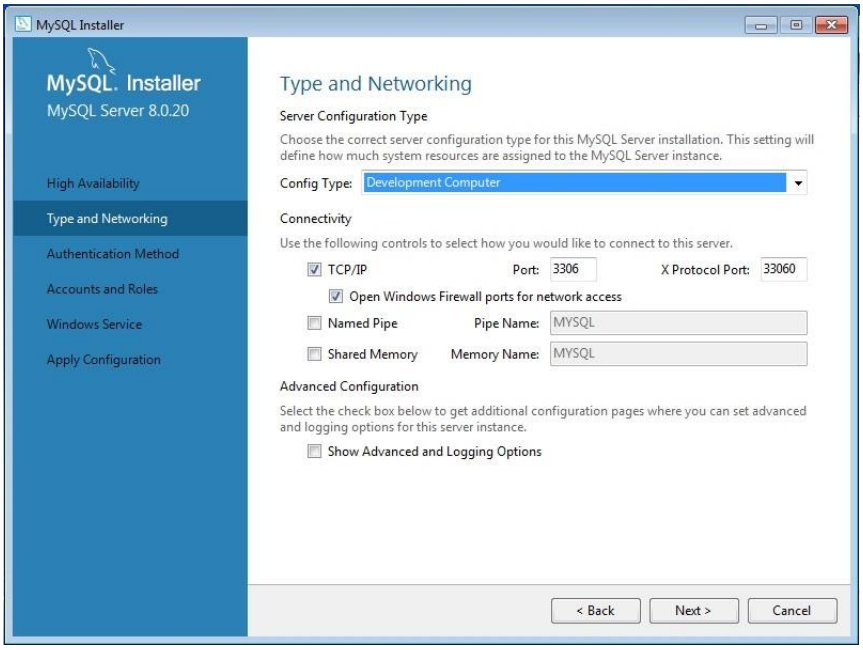

Рис. 1: Окно настройки серверной части

В следующем окне необходимо настроить пароль (учётная запись — root) главного администратора сервера. Оставлять это поле пустым не рекомендуется, поскольку в таком случае любой может получить доступ к администрированию ваших баз данных, что негативно скажется на безопасности.

Так как требования к типу ИС 1В указано, что ’должны осуществляться идентификация и проверка подлинности субъектов доступа при входе в систему по идентификатору (коду) и паролю условно-постоянного действия длиной не менее шести буквенно-цифровых символов’, то необходимо установить пароль согласно этому требованию.

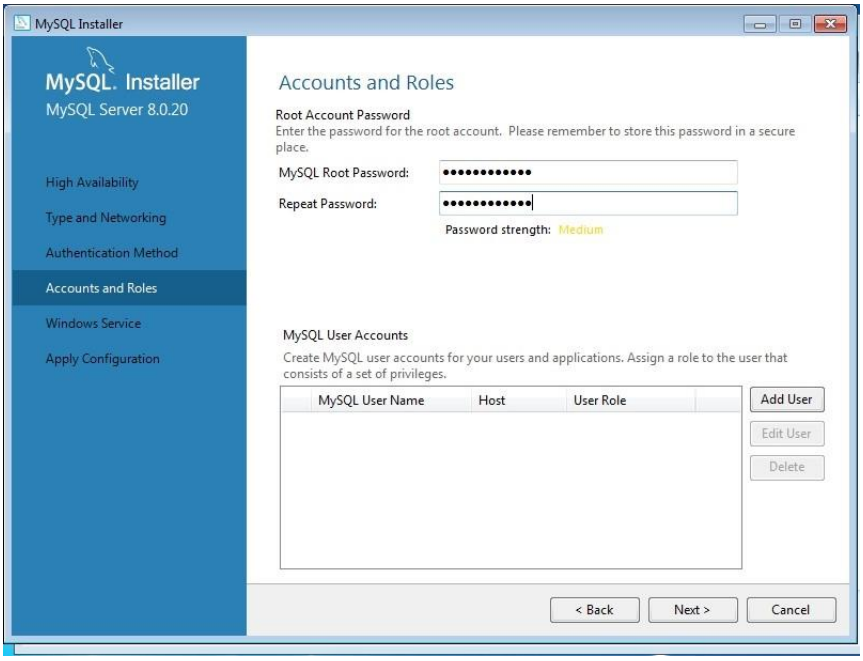

Рис. 2: Указание пароля

Окно конфигурации сервера можно оставить без изменения.

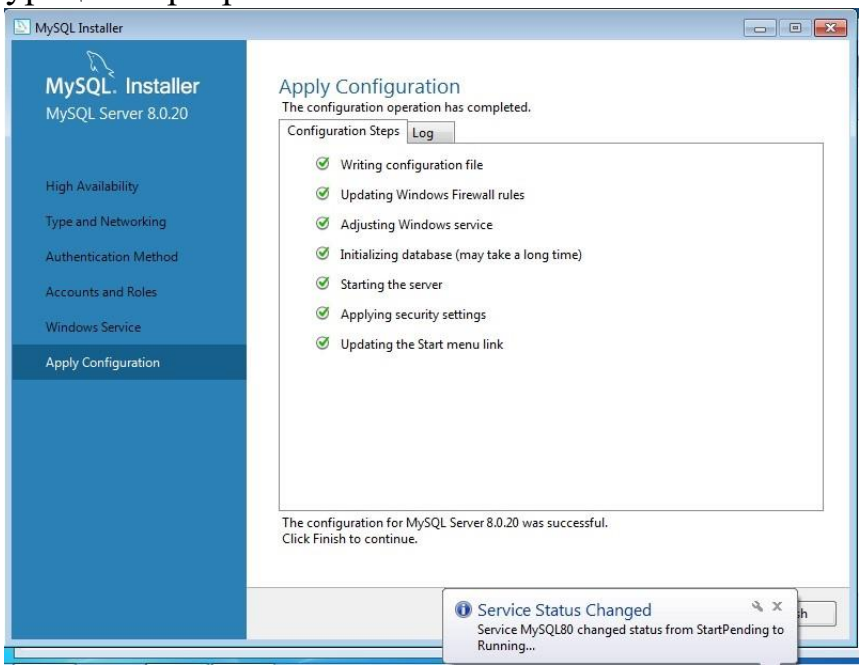

Рис. 3: Окно настройки конфигурации

В окне выбора дополнительных возможностей можно указать, чтобы установить помимо основного сервера, ещё и файлы документации и различные расширения.

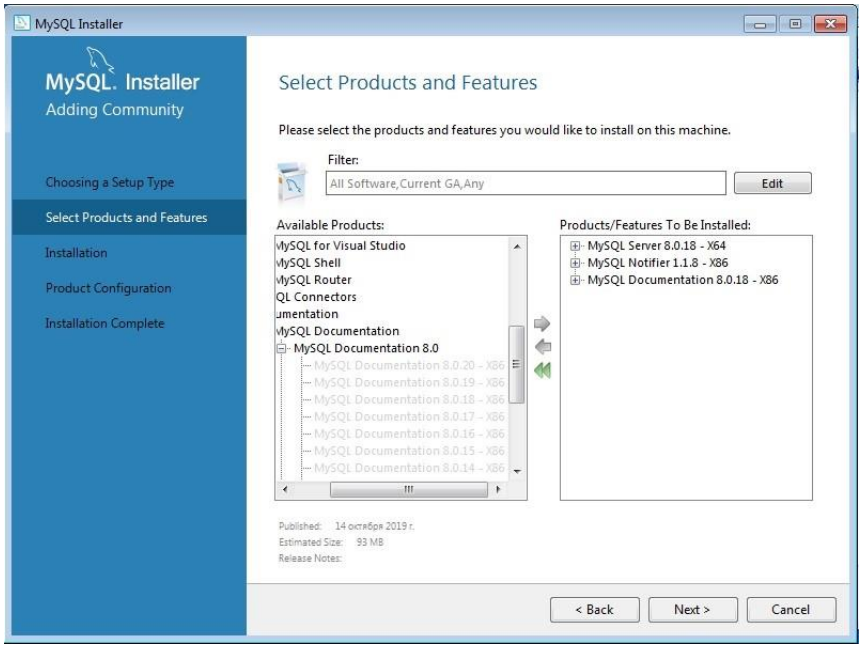

Рис. 4: Дополнительные инструменты и возможности

3.2 Создание базы данных MySQL

На данный момент имеется настроенный сервер ’MySQL Server’ и утилита для взаимодействия с ним ’MySQL Notifier’. Для того, чтобы можно было использовать возможности сервера, такие как создание и назначение прав пользователям на таблицы и базы данных, требуется сначала создать пользовательскую базу данных и таблицу в ней. Имеющиеся по умолчанию базы данных можно посмотреть с использованием команды show database.

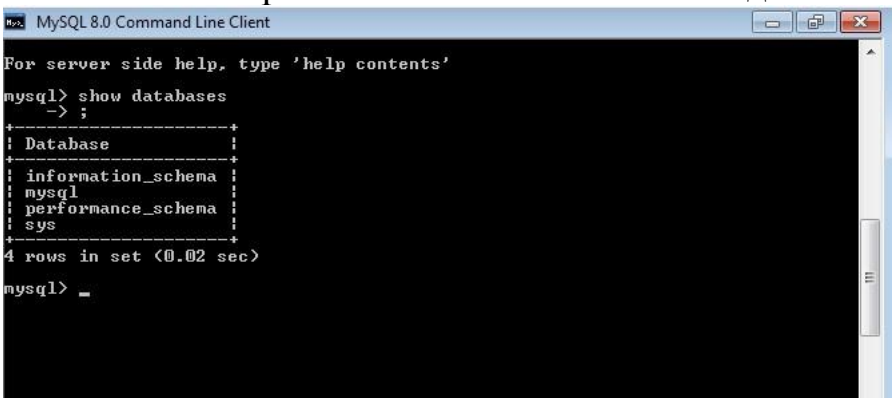

Рис. 5: Базы данных по умолчанию

Поскольку среди перечисленных баз данных имеются те, изменения в которых нежелательны (в частности, в базе данных mysql находятся таблица user с перечнем всех пользователей, их паролей и способов подключения к базам данных), необходимо создать собственную базу данных с помощью

команды create database laboratoryWork3. После этого необходимо удостовериться в корректности создания.

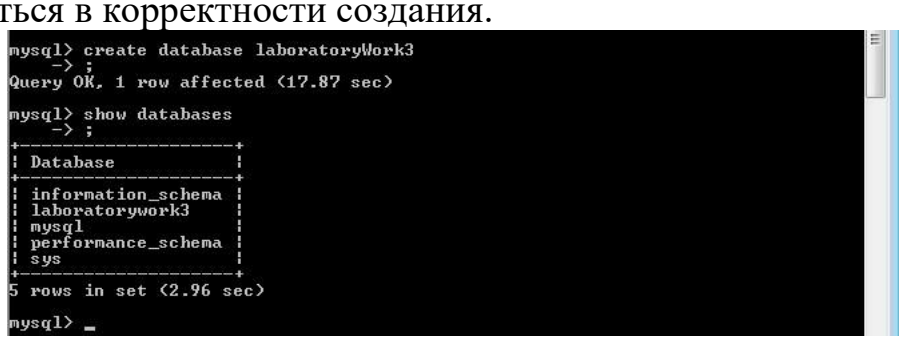

Рис. 6: Создание собственной базы данных

Необходимо указать серверу, что далее работа будет происходить именно с ней. И создадим нового пользователя с именем ’LocalUser’.

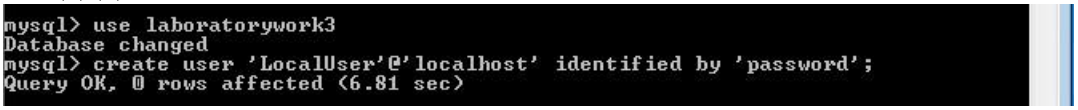

Рис. 7: Задание используемой базы данных и нового пользователя

Предоставим пользователю полные права на все таблицы, созданной ранее базы данных laboratoryWork3

Рис. 8: Предоставление прав пользователя бд

Проверим права доступа, воспользовавшись командой show grants for ’LocalUser’’localhost’

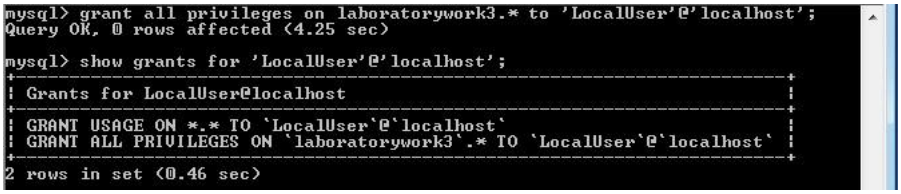

Рис. 9: Проверка прав доступа

Для того, чтобы организовать доступ к серверу не привилегированного пользователя установим утилиту ’MySQL Workbench’:

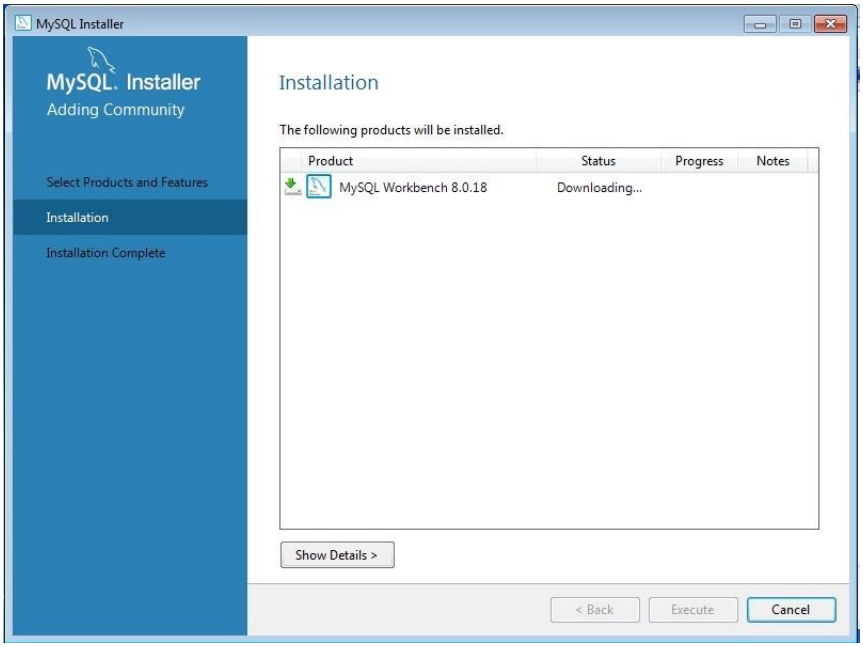

Рис. 10: Установка ’MySQL Workbench’

Чтобы запустить командную строку под новым пользователем необходимо зайти в интерфейс программы "MySQL Workbench"и создать новое подключение, вызвав Manage Server Connection, для которого настроить следующие параметры, а остальные оставим по умолчанию:

- Connection Name: LocalUser
- Password: password

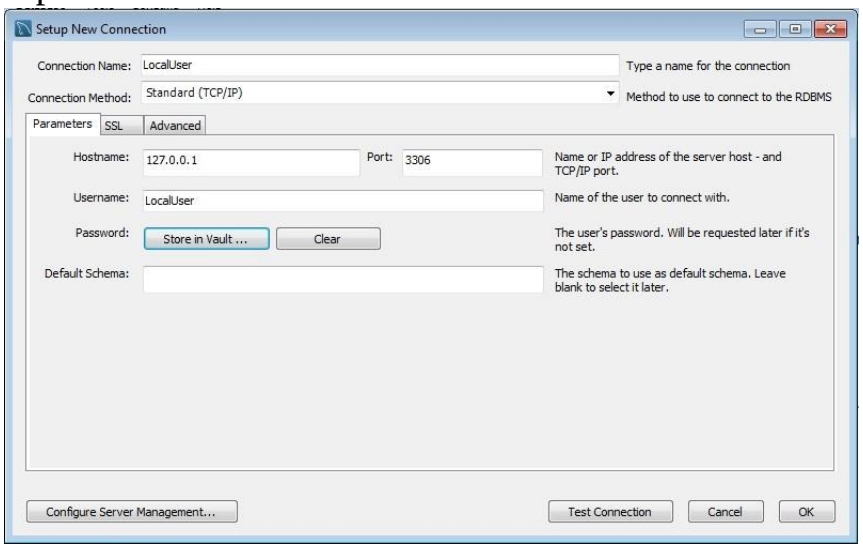

Рис. 11: Настройка Manage Server Connection

Можно попробовать подключиться к серверу с использованием этого нового соединения, выполнив следующие действия: кликнуть на подключение

LocalUser правой кнопкой мыши и затем Start Command Line Client. Посмотрев доступные базы данных можно заметить, что некоторые базы данных пропали, так как у данного пользователя нет доступа к ним.

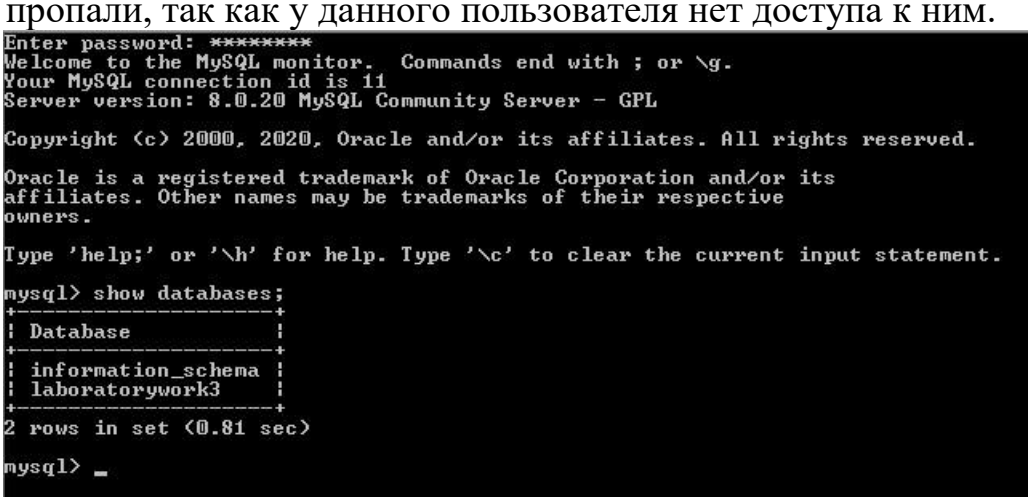

Рис. 12: Успешное подключение пользователя LocalUser к серверу

Теперь назначим пользователю специальные права доступа.

В процессе настроек сервера 'MySQL Server' была пропущена возможность создания нового пользователя с заданными правами. Используем ее при выполнении данного этапа задания.

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, как уже упоминалось выше. Сделаем это следующей командой:

Рис. 13: Назначение прав пользователю LocalUser

Согласно заданию создадим таблицу PhoneNumber с полями UserName, UserAddres UserPhone с помощью команды create table

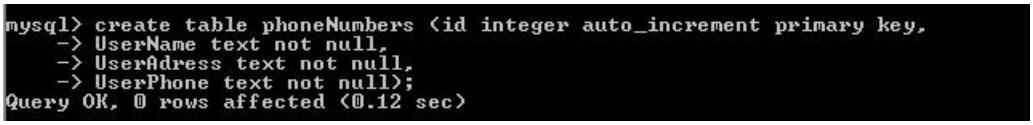

Рис. 14: Создание таблицы

Заполним созданную таблицу произвольными данными.

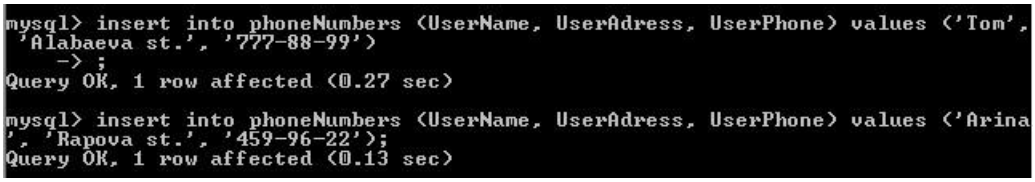

Рис. 15: Добавление записей в таблицу

Просмотрим содержимое таблицы. Для этого необходимо извлечь все столбцы и строки с помощью команды select.

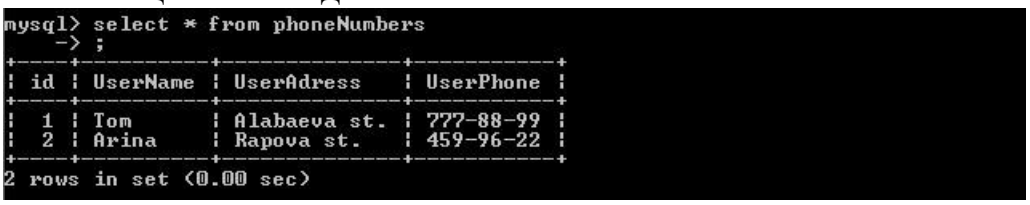

Рис. 16: Настройка Manage Server Connection

Предположим, что АРМ, за которой работает пользователь, расположено в 'закрытом' контуре и на нём обрабатываются важные данные. Поскольку в предыдущем задании была создана таблица-телефонный справочник, логично предположить, что пользователь АРМ оперирует данными, предположим, клиентов предприятия. Следовательно, ему можно разрешить права SELECT, делать выборку из всех записей таблицы, и INSERT, право на добавление новых записей. Однако пользователю нельзя разрешать создавать новые таблицы, модифицировать уже имеющиеся записи и тем более удалять записи и таблицы из базы данных.

Установим права SELECT и INSERT на базу данных laboratoryWork3 и непосредственно для таблицы phoneNumber для пользователя LocalUser:

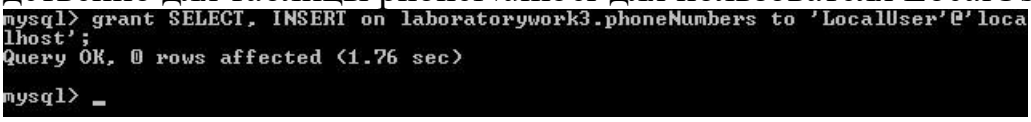

Рис. 17: Назначение прав пользователя LocalUser

В таблице phoneNumber пользователь UserLocal имеет право просматривать любые поля и записи, поскольку в соответствии с его ролью на предприятии он оперирует этими данными, поэтому ограничивать на доступ к отдельным полям таблицы устанавливать не нужно. Теперь проверим, применялись ли права, просмотрев их для пользователя:

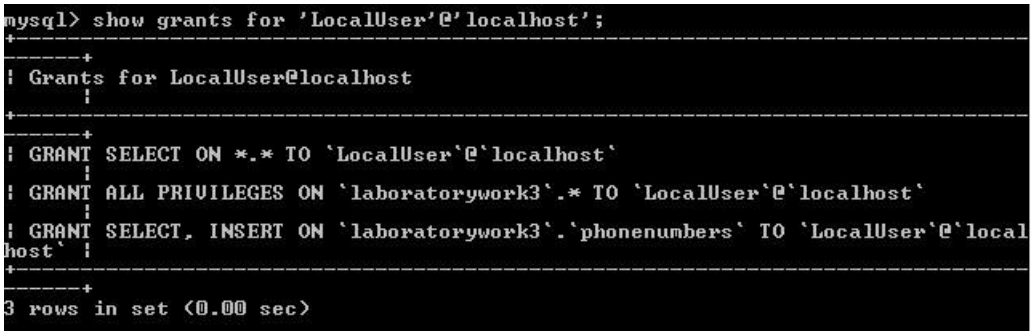

Рис. 18: Просмотр прав пользователя LocalUser

Очевидно, что изменение прав доступа к базе данных и содержащимся в них таблицам для пользователя LocalUser успешно осуществлено в соответствии с ролью пользователя на предприятии.

Теперь проверим корректность настроенных прав для пользователя LocalUser, а также ознакомимся с механизмом запроса записей из таблицы в SQL. Сделаем несколько выборок из таблицы phoneNumber:

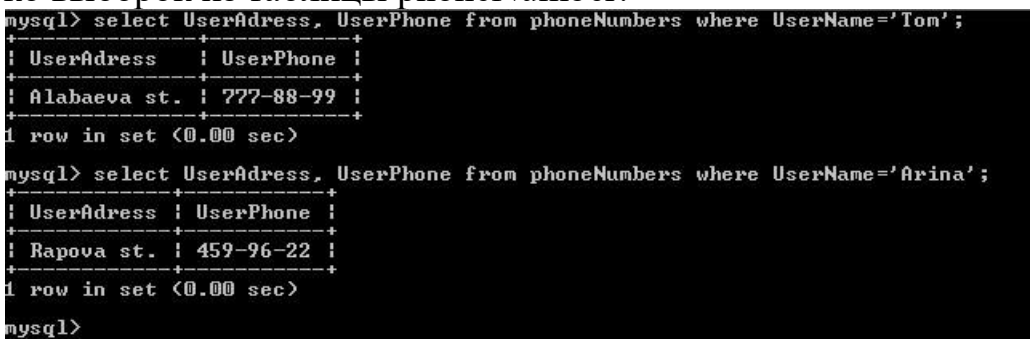

Рис. 19: Результат выборки по значениям адреса и телефона для двух произвольных пользователей

Выборка всех записей из таблица phoneNumber с сортировкой по полю UserName в алфавитном порядке:

Рис. 20: Результат выборки по всем записям в алфавитном порядке Проверим корректно ли работает механизм задания прав пользователя, попробовав удалить таблицу phoneNumber:

Рис. 21: Удаление таблицы

```
mysql> show tables;
Empty set (0.00 sec)

mysql>
```

Рис. 22: Удаление прошло успешно

Теперь удалим базу данных laboratoryWork3 и просмотрим список таблиц снова:

```
mysql> drop database laboratorywork3;
Query OK, 0 rows affected (0.13 sec)

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

Рис. 23: Удаление бд

Теперь удалим все права доступа пользователя LocalUser:

```
mysql> revoke all privileges on *.* from 'LocalUser'@'localhost';
Query OK, 0 rows affected (0.06 sec)

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

Рис. 24

Просмотрим список всех пользователей до удаления LocalUser, обратившись к таблице mysql.user и запросив из неё всех пользователей и их способ подключения к серверу:

```
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 (0.00 sec)
```

Рис. 25

После чего удалим пользователя LocalUser:

```
mysql> drop user 'LocalUser'@'localhost';
Query OK, 0 rows affected (0.10 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)
```

Рис. 26

4 Выводы

В работе проведена установка MySQL, создана база данных. Назначены и проверены права пользователя по доступу к ресурсам сервера базы данных. Настройка параметров безопасности с учетом заданного типа безопасности ИС — прав доступа к ресурсам сервера базы данных, а именно, SELECT, права на выборку из всех записей таблицы и INSERT позволяет повысить уровень защиты базы данных от НСД, запрет на добавление новых таблиц, пользователей и т.д., так как это может повлиять на сохранность данных.