

ГУАП

КАФЕДРА № 41

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

Старший преподаватель

должность, уч. степень, звание

подпись, дата

Б.К. Акопян

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №8

ОСНОВЫ АДМИНИСТРИРОВАНИЯ СУБД MYSQL

по курсу: БАЗЫ ДАННЫХ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. № 4217

подпись, дата

Д.М. Никитин

инициалы, фамилия

Санкт-Петербург 2024

1. **Цель работы:** получение практических навыков по администрированию СУБД MySQL.

2. **Номер варианта и описание предметной области в текстовом варианте:**

24 – база данных железнодорожной станции («Новосибирск-главный»).

1. Внимательно изучите материалы раздела.

2. Выполните упражнения по администрированию СУБД с помощью консольного клиента и программными средствами приложения MySQL Workbench.

3. Выявите на основе анализа предметной области выбранного варианта не менее трех пользователей с разными привилегиями, в частности, администратор БД, аналитик, бухгалтер, рядовой пользователь. Опишите в текстовом или табличном формате права каждого по доступу к БД, в частности, полный спектр прав, на доступ к конкретной таблице, на модификацию объектов БД и т.д.

Пример описания пользователей. Магазин состоит из трех отделов: склад, торговый зал, офис. На складе работает кладовщик, в торговом зале – продавец, в офисе – экономист. Все они работают в одной информационной системе. Также есть администратор, который занимается поддержкой данной ИС. Экономист ведет учет сотрудников, составляет рабочий график, утверждает заявки на приобретение товара и получает счета. Кладовщик вносит и изымает данные о товарах, присутствующих/отсутствующих на складе. Продавец должен обладать информацией о наличии товара на складе и его количестве, также продавец составляет товарный чек.

4. Назначьте права доступа каждому пользователю. Среду выполнения администрирования выберите самостоятельно. Команды в отчете должны быть представлены в текстовом формате. Результаты их выполнения в виде скриншотов с поясняющими подписями. Не

забудьте вручную обновлять настройки сервера, чтобы продемонстрировать изменения.

В обязательном порядке должны присутствовать следующие команды.

4.1. Создание пользователя `CREATE USER 'user2' IDENTIFIED BY '123456';`

Присвоения всех привилегий во всех БД и таблицах

4.2. Создание пользователя без привилегий `CREATE USER 'user3' IDENTIFIED BY '123456'; GRANT USAGE ON *.* TO user3 WITH GRANT OPTION;`

4.3. Просмотр прав пользователей `USE mysql; SELECT * FROM user;`

4.4. Передача выборочных прав пользователю `GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, ALTER, DROP ON *.* TO user2 WITH GRANT OPTION;`

4.5. Удаление привилегий для одного пользователя. `REVOKE ALL PRIVILEGES ON *.* FROM 'user2';`

5. Создайте 2 роли и назначьте на них пользователей: один пользователь на одну роль и несколько - на другую.

6. Покажите список ролей на доступ к БД.

7. Добавьте еще две команды самостоятельно.

3. Схема данных из MySQL Workbench:

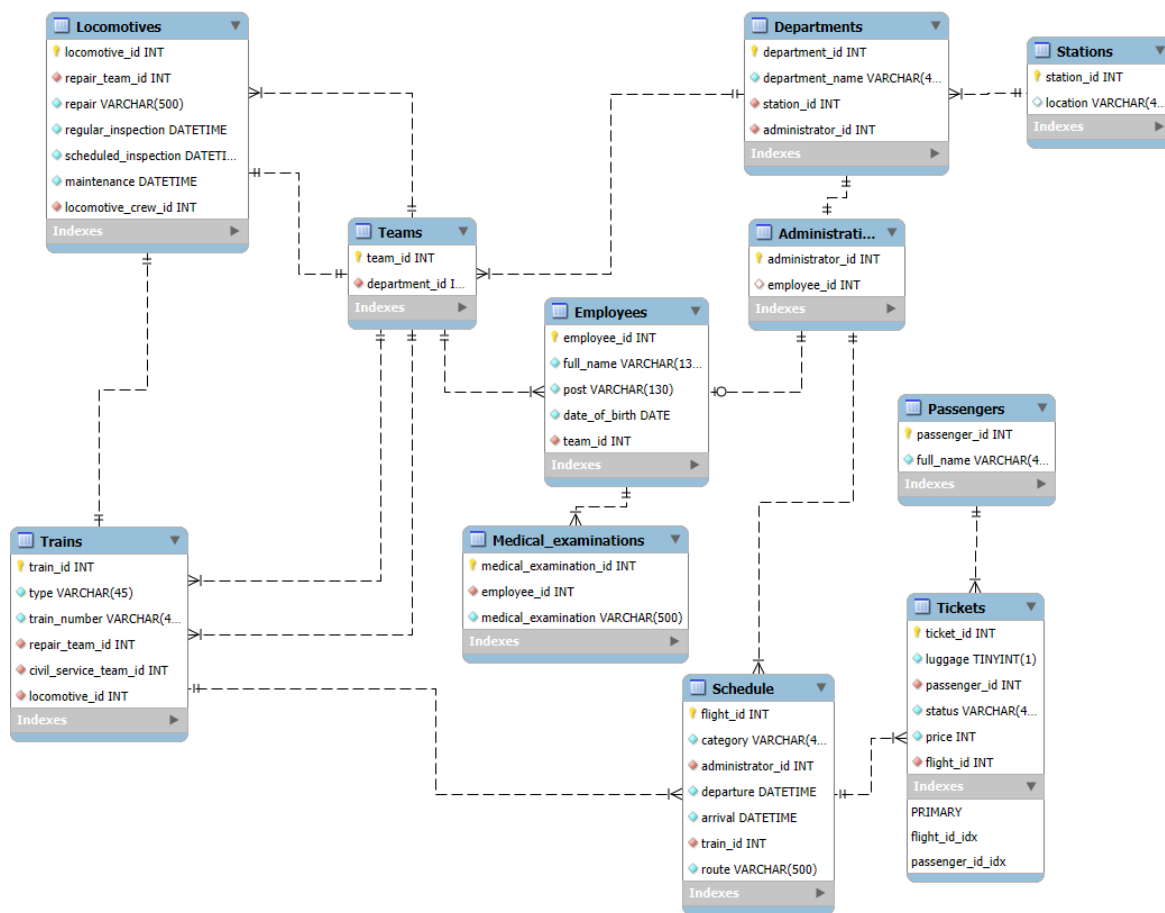


Рисунок 1 – Схема данных

4. Список пользователей с привилегиями на сервере с кодом запроса и результатом выполнения:

```

mysql> SELECT user, host FROM mysql.user;
+-----+-----+
| user          | host          |
+-----+-----+
| admin_role    | %             |
| dispatcher    | %             |
| mysql         | %             |
| administrator | localhost     |
| chief_dispatcher | localhost    |
| dispatcher1    | localhost     |
| dispatcher2    | localhost     |
| medical_service | localhost     |
| mysql.infoschema | localhost    |
| mysql.session | localhost     |
| mysql.sys     | localhost     |
| passenger     | localhost     |
| repair_master  | localhost     |
| root          | localhost     |
| spectator     | localhost     |
+-----+-----+
15 rows in set (0.00 sec)
    
```

Рисунок 2 – Список пользователей

Администратор был получен с помощью постоянной роли admin_role.

```
mysql> CREATE ROLE 'admin_role';
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT ALL PRIVILEGES ON *.* TO 'admin_role' WITH GRANT OPTION;
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT 'admin_role' TO 'administrator'@'localhost';
Query OK, 0 rows affected (0.00 sec)

mysql> SHOW GRANTS FOR 'administrator'@'localhost';
+-----+
| Grants for administrator@localhost |
+-----+
| GRANT USAGE ON *.* TO `administrator`@`localhost` |
| GRANT `admin_role`@`%` TO `administrator`@`localhost` |
+-----+
2 rows in set (0.00 sec)

mysql> SET DEFAULT ROLE 'admin_role' TO 'administrator'@'localhost';
Query OK, 0 rows affected (0.00 sec)

mysql> SHOW GRANTS FOR 'administrator'@'localhost';
+-----+
| Grants for administrator@localhost |
+-----+
| GRANT USAGE ON *.* TO `administrator`@`localhost` |
| GRANT `admin_role`@`%` TO `administrator`@`localhost` |
+-----+
2 rows in set (0.00 sec)
```

Рисунок 3 – Привилегии и создание администратора

```
mysql> SHOW GRANTS FOR 'chief_dispatcher'@'localhost';
+-----+
| Grants for chief_dispatcher@localhost |
+-----+
| GRANT USAGE ON *.* TO `chief_dispatcher`@`localhost` |
| GRANT SELECT, INSERT, UPDATE, DROP ON `train_station`.`schedule` TO `chief_dispatcher`@`localhost` |
+-----+
2 rows in set (0.00 sec)
```

Рисунок 4 – Привилегии главного диспетчера

```
mysql> CREATE ROLE 'dispatcher';
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT SELECT, INSERT, UPDATE ON `train_station`.`schedule` TO 'dispatcher';
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT 'dispatcher' TO 'dispatcher1'@'localhost', 'dispatcher2'@'localhost';
Query OK, 0 rows affected (0.00 sec)
```

Рисунок 5 – Создание роли и её установка для диспетчеров (часть 1)

```
mysql> SET DEFAULT ROLE 'dispatcher' TO 'dispatcher1'@'localhost';
Query OK, 0 rows affected (0.00 sec)

mysql> SET DEFAULT ROLE 'dispatcher' TO 'dispatcher2'@'localhost';
Query OK, 0 rows affected (0.00 sec)

mysql> SHOW GRANTS FOR 'dispatcher1'@'localhost';
+-----+
| Grants for dispatcher1@localhost |
+-----+
| GRANT USAGE ON *.* TO `dispatcher1`@`localhost` |
| GRANT `dispatcher`@`%` TO `dispatcher1`@`localhost` |
+-----+
2 rows in set (0.00 sec)

mysql> SHOW GRANTS FOR 'dispatcher2'@'localhost';
+-----+
| Grants for dispatcher2@localhost |
+-----+
| GRANT USAGE ON *.* TO `dispatcher2`@`localhost` |
| GRANT `dispatcher`@`%` TO `dispatcher2`@`localhost` |
+-----+
2 rows in set (0.00 sec)
```

Рисунок 6 – Создание роли и её установка для диспетчеров (часть 2)

```
mysql> SHOW GRANTS FOR 'medical_service'@'localhost';
+-----+
| Grants for medical_service@localhost |
+-----+
| GRANT USAGE ON *.* TO `medical_service`@`localhost` |
| GRANT SELECT, SELECT ('employee_id', 'medical_examination_id'), UPDATE ('medical_examination') ON `train_station`.`medical_examinations` TO `medical_service`@`localhost` |
+-----+
2 rows in set (0.00 sec)
```

Рисунок 7 – Привилегии медицинской службы

```
mysql> SHOW GRANTS FOR 'passenger'@'localhost';
+-----+
| Grants for passenger@localhost |
+-----+
| GRANT USAGE ON *.* TO `passenger`@`localhost` |
| GRANT SELECT ('arrival', 'category', 'departure', 'route', 'train_id') ON `train_station`.`schedule` TO `passenger`@`localhost` |
| GRANT SELECT ('flight_id', 'luggage', 'price', 'status') ON `train_station`.`tickets` TO `passenger`@`localhost` |
| GRANT SELECT ('train_number', 'type') ON `train_station`.`trains` TO `passenger`@`localhost` |
+-----+
4 rows in set (0.00 sec)
```

Рисунок 8 – Привилегии пассажира

```
mysql> SHOW GRANTS FOR 'repair_master'@'localhost';
+-----+
| Grants for repair_master@localhost |
+-----+
| GRANT USAGE ON *.* TO `repair_master`@`localhost` |
| GRANT SELECT, SELECT ('locomotive_crew_id', 'locomotive_id'), INSERT, UPDATE ('maintenance', 'regular_inspection', 'repair', 'repair_team_id', 'scheduled_inspection') ON `train_station`.`locomotives` TO `repair_master`@`localhost` |
+-----+
2 rows in set (0.00 sec)
```

Рисунок 9 – Привилегии мастера ремонтной бригады

```
mysql> SHOW GRANTS FOR 'spectator'@'localhost';
+-----+
| Grants for spectator@localhost |
+-----+
| GRANT USAGE ON *.* TO `spectator`@`localhost` WITH GRANT OPTION |
+-----+
1 row in set (0.00 sec)
```

Рисунок 10 – Привилегии наблюдателя

```
mysql> SHOW GRANTS FOR 'dispatcher';
+-----+
| Grants for dispatcher@% |
+-----+
| GRANT USAGE ON *.* TO `dispatcher`@`%` |
| GRANT SELECT, INSERT, UPDATE ON `train_station`.`schedule` TO `dispatcher`@`%` |
+-----+
2 rows in set (0.00 sec)
```

Рисунок 11 – Привилегии роли диспетчера

5. Описание выявленных пользователей предметной области:

Администратор (administrator) – имеет все привилегии, администрирует базу данных.

Главный диспетчер (chief_dispatcher) – Может производить любые действия с таблицей schedule, как и обычные диспетчеры.

Медицинская служба (medical_service) – может выбрать id карточки медслужбы (medical_examination_id) и id работника (employee_id), изменять и выбирать состояние медосмотра (medical_examination) из таблицы medical_examinations.

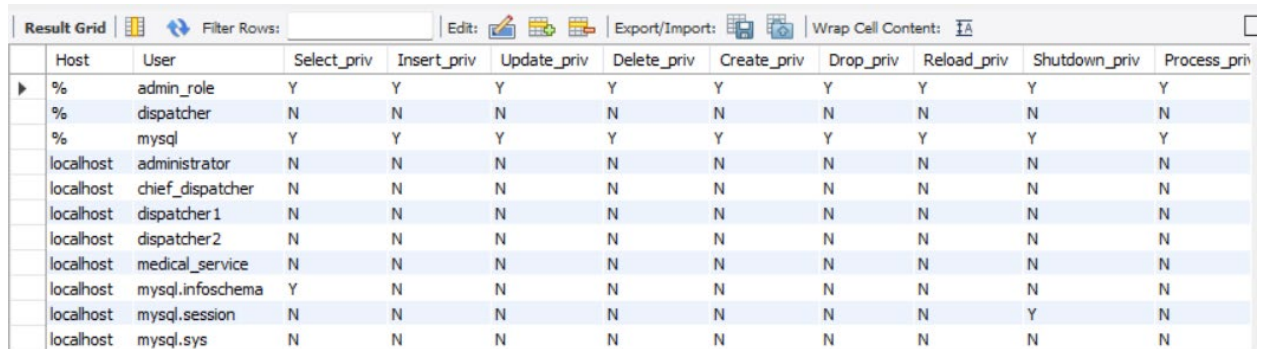
Пассажир (passenger) – может выбрать время отправления (departure), прибытия (arrival), категорию (category), маршрут поезда (route) и его id (train_id) из таблицы schedule, по id из таблицы trains он может получить номер поезда (train_number) и его тип (type). Также пользователь может получить доступ к своим старым заказам, а конкретно, к полям багаж (luggage), цена (price), статус (status) и id отправления (flight_id) в таблице tickets.

Мастер ремонтной бригады (repair_master) – может просматривать id локомотивной бригады (locomotive_crew_id) и id локомотива (locomotive_id), а также вставлять и изменять данные в таблице locomotives для полей техническое обслуживание (maintenance), регулярный осмотр, (regular_inspection), ремонт (repair), id ремонтной бригады (repair_team_id),

плановый осмотр (scheduled_inspection).

Наблюдатель (spectator) – пользователь без прав.

Далее был проведён просмотр прав пользователей с помощью `USE mysql; SELECT * FROM user;.`



	Host	User	Select_priv	Insert_priv	Update_priv	Delete_priv	Create_priv	Drop_priv	Reload_priv	Shutdown_priv	Process_priv
▶	%	admin_role	Y	Y	Y	Y	Y	Y	Y	Y	Y
	%	dispatcher	N	N	N	N	N	N	N	N	N
	%	mysql	Y	Y	Y	Y	Y	Y	Y	Y	Y
	localhost	administrator	N	N	N	N	N	N	N	N	N
	localhost	chief_dispatcher	N	N	N	N	N	N	N	N	N
	localhost	dispatcher1	N	N	N	N	N	N	N	N	N
	localhost	dispatcher2	N	N	N	N	N	N	N	N	N
	localhost	medical_service	N	N	N	N	N	N	N	N	N
	localhost	mysql.infoschema	Y	N	N	N	N	N	N	N	N
	localhost	mysql.session	N	N	N	N	N	N	N	Y	N
	localhost	mysql.sys	N	N	N	N	N	N	N	N	N

Рисунок 12 – Просмотр прав пользователей

Далее была использована команда `REVOKE ALL PRIVILEGES ON *.* FROM 'testrm'@'localhost';` для удаления всех привилегий для пользователя. После пользователь был удалён.

```
mysql> CREATE USER 'testrm'@'localhost';
Query OK, 0 rows affected (0.01 sec)

mysql> GRANT ALL PRIVILEGES ON *.* TO 'testrm'@'localhost' WITH GRANT OPTION;
Query OK, 0 rows affected (0.01 sec)

mysql> REVOKE ALL PRIVILEGES ON *.* FROM 'testrm'@'localhost';
Query OK, 0 rows affected (0.01 sec)

mysql> DROP USER 'testrm'@'localhost';
Query OK, 0 rows affected (0.01 sec)
```

Рисунок 13 – Процесс применения вышеописанной команды

6. **Список пользователей, подключенных только к БД, разработанной по варианту с кодом запроса и результатом выполнения в виде скриншота:**

administrator	localhost
chief_dispatcher	localhost
dispatcher1	localhost
dispatcher2	localhost
medical_service	localhost
mysql.infoschema	localhost
mysql.session	localhost
mysql.sys	localhost
passenger	localhost
repair_master	localhost
root	localhost
spectator	localhost

Рисунок 14 – Список пользователей, подключённых только к данной базе данных

7. Запросы на назначение прав каждому пользователю:

Установка прав для администратора:

```
CREATE ROLE 'admin_role';
GRANT ALL PRIVILEGES ON *.* TO 'admin_role' WITH GRANT OPTION;
GRANT 'admin_role' TO 'administrator'@'localhost';
SET DEFAULT ROLE 'admin_role' TO 'administrator'@'localhost';
```

Установка прав для мастера ремонтной бригады:

```
GRANT SELECT, INSERT, UPDATE (repair_team_id, repair,
regular_inspection, scheduled_inspection, maintenance) ON
train_station.locomotives TO repair_master@localhost;
GRANT SELECT (locomotive_id, locomotive_crew_id) ON
train_station.locomotives TO repair_master@localhost;
```

Установка прав для медсервиса:

```
GRANT SELECT, UPDATE (medical_examination) ON
`train_station`.`medical_examinations` TO
medical_service@localhost;
GRANT SELECT (employee_id, medical_examination_id) ON
`train_station`.`medical_examinations` TO
medical_service@localhost;
```

Установка прав для главного диспетчера:

```
GRANT SELECT, INSERT, UPDATE ON `train_station`.`schedule` TO
```

```
chief_dispatcher@localhost;
```

Установка прав для пассажира:

```
GRANT USAGE ON *.* TO 'passenger'@'localhost';
GRANT SELECT (`arrival`, `category`, `departure`, `route`,
`train_id`) ON `train_station`.`schedule` TO
'passenger'@'localhost';
GRANT SELECT (`flight_id`, `luggage`, `price`, `status`) ON
`train_station`.`tickets` TO 'passenger'@'localhost';
GRANT SELECT (`train_number`, `type`) ON
`train_station`.`trains` TO 'passenger'@'localhost';
```

Установка прав для диспетчера 1 и 2:

```
CREATE ROLE 'dispatcher';
GRANT SELECT, INSERT, UPDATE ON `train_station`.`schedule` TO
'dispatcher'@'%';
GRANT 'dispatcher' TO 'dispatcher1'@'localhost',
'dispatcher2'@'localhost';
SET DEFAULT ROLE 'dispatcher' TO 'dispatcher1'@'localhost';
SET DEFAULT ROLE 'dispatcher' TO 'dispatcher2'@'localhost';
```

8. Вывод в формате эссе:

В ходе выполнения лабораторной работы по администрированию СУБД MySQL я приобрел полезные навыки, которые значительно расширили мое понимание работы с базами данных. В частности, я изучил процессы создания и управления пользователями, назначение привилегий, а также использование ролей для упрощения распределения прав доступа. Основное внимание было уделено настройке прав для разных категорий пользователей, таких как администраторы, диспетчеры, медицинский персонал и пассажиры, что позволило четко разграничить доступ к различным данным и функциям базы данных. При этом я столкнулся с рядом технических проблем, таких как ошибки при назначении ролей с помощью команды SET DEFAULT ROLE и проблемы с корректным указанием привилегий для столбцов таблиц. Решение этих проблем потребовало дополнительных исследований и корректировки синтаксиса команд, что укрепило мои навыки работы с MySQL и повысило

уверенность в администрировании баз данных.