

Государственное бюджетное профессиональное образовательное учреждение
города Москвы
«Московский государственный колледж электромеханики и
информационных технологий»
(ГБПОУ МГКЭИТ)

Отчет по практической работе №4
Проектирование базы данных инвентаризация

Выполнил студент группы ЗИП-11-19

Насонов Д.С.

ВВЕДЕНИЕ

Цель – Научиться разрабатывать объекты базы данных.

Задачи:

- 1) Написать SQL-код создания таблиц;
- 2) Написать SQL-код создания представлений;
- 3) Написать SQL-код создания API (процедур и функций);
- 4) Написать SQL-код создания пользователей.

Результат: отчёт, sql-файлы с кодом реализации

Предмет исследования – магазин.

Объекты исследования – база данных.

ОСНОВНАЯ ЧАСТЬ

1. Написать SQL-код создания таблиц.

1) Код написания таблицы «auditor».

```
CREATE TABLE `инвентаризация`.`auditor` (  
  id_auditor INT AUTO_INCREMENT PRIMARY KEY,  
  name VARCHAR(30) NOT NULL,  
  last_name VARCHAR(30) NOT NULL,  
  middle_name VARCHAR(30) NOT NULL,  
  experience FLOAT NOT NULL,  
  id_scanner INT NOT NULL  
);
```

2) Код написания таблицы «Employee».

```
CREATE TABLE `инвентаризация`.`employee` (  
  id_employee INT AUTO_INCREMENT PRIMARY KEY,  
  name VARCHAR(30) NOT NULL,  
  last_name VARCHAR(30) NOT NULL,  
  middle_name VARCHAR(30) NOT NULL,  
  experience FLOAT NOT NULL  
);
```

3) Код написания таблицы «Scanner».

```
CREATE TABLE `инвентаризация`.`scanner` (  
  Id_scanner INT AUTO_INCREMENT PRIMARY KEY,  
  name VARCHAR(30) NOT NULL,  
  operation_system VARCHAR(30) NOT NULL,  
  producer VARCHAR(30) NOT NULL  
);
```

4) Код написания таблицы «Items».

```
CREATE TABLE `инвентаризация`.`items` (  
    Id_item INT AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(30) NOT NULL,  
    amount INT UNSIGNED NOT NULL,  
    trend VARCHAR(30) NOT NULL,  
    colour VARCHAR(30) NOT NULL,  
    sex VARCHAR(30) NOT NULL,  
    structure VARCHAR(30) NOT NULL  
);
```

5) Код написания таблицы «Section».

```
CREATE TABLE `инвентаризация`.`section` (  
    Id_section INT AUTO_INCREMENT PRIMARY KEY,  
    Id_auditor INT(11),  
    id_employee INT(11),  
    id_item INT(11)  
);
```

2. Написать SQL-код создания представлений.

1) Информация о ревизорах

```
CREATE  
ALGORITHM = UNDEFINED  
DEFINER = `root`@`localhost`  
SQL SECURITY DEFINER  
VIEW `informationaboutauditors` AS  
SELECT  
    `auditor`.`first_name` AS `first_name`,  
    `auditor`.`last_name` AS `last_name`,  
    `auditor`.`middle_name` AS `middle_name`,  
    `auditor`.`experience` AS `experience`,  
    `auditor`.`id_scanner` AS `id_scanner`
```

FROM

`auditor`

2) Информация о сотрудниках

CREATE

ALGORITHM = UNDEFINED

DEFINER = `root` @`localhost`

SQL SECURITY DEFINER

VIEW `informationaboutemployee` AS

SELECT

`e`.`first_name` AS `first_name`,

`e`.`last_name` AS `last_name`,

`e`.`middle_name` AS `middle_name`,

`e`.`experience` AS `experience`,

`s`.`id_section` AS `id_section`

FROM

(`employee` `e`

**JOIN `section` `s` ON ((`e`.`id_employee` =
`s`.`id_employee`)))**

ORDER BY `e`.`first_name`

3. Написать SQL-код создания API (процедур и функций).

SQL-код	Описание функции
add_auditor(first_name varchar(30), last_name varchar(30), middle_name varchar(30), experience float(11),id_scanner)	Добавляет ревизора в базу данных
CREATE DEFINER=`root`@`localhost` PROCEDURE `auditor_first_name`(id int, first_name varchar(30)) BEGIN update auditor set first_name=first_name where id_auditor=id; END	Изменяет имя ревизора

<pre>CREATE DEFINER=`root`@`localhost` PROCEDURE `auditor_last_name`(id int, last_name varchar(30)) BEGIN update auditor set last_name=last_name where id_auditor=id; END</pre>	Изменяет фамилию ревизора
<pre>CREATE DEFINER=`Ревизор`@`%` PROCEDURE `auditor_middle_name`() BEGIN update auditor set middle_name='Овалов' where id_auditor=2; END</pre>	Изменяет отчество ревизора
<pre>CREATE DEFINER=`Ревизор`@`%` PROCEDURE `experience`() BEGIN UPDATE auditor set experience=12 WHERE id_auditor=4; END</pre>	Изменяет стаж ревизора
<pre>CREATE DEFINER=`Ревизор`@`%` PROCEDURE `scanner`() BEGIN UPDATE auditor set id_scanner=4 WHERE id_auditor=2; END</pre>	Изменяет сканнер ревизора
<pre>CREATE DEFINER=`Ревизор`@`%` PROCEDURE `delete_auditor`() BEGIN delete from auditor where id_auditor=2; END</pre>	Удаляет ревизора по идентификатору
<pre>add_employee(first_name varchar(30), last_name varchar(30), middle_name varchar(30), experience float(11))</pre>	Добавляет сотрудника в базу данных

<pre>CREATE DEFINER='Ревизор'@`%` PROCEDURE `employee_first_name`() BEGIN update employee set first_name='Роман' where id_employee=2; END</pre>	Изменяет имя сотрудника
<pre>CREATE DEFINER='Ревизор'@`%` PROCEDURE `employee_last_name`() BEGIN update employee set last_name='Попов' where id_employee=2; END</pre>	Изменяет фамилию сотрудника
<pre>CREATE DEFINER='Ревизор'@`%` PROCEDURE `employee_middle_name`() BEGIN update employee set middle_name='Александрович' where id_employee=2; END</pre>	Изменяет Отчество сотрудника
<pre>CREATE DEFINER='Ревизор'@`%` PROCEDURE `employee_experience`() BEGIN update employee set experience='5' where id_employee=1; END</pre>	Изменяет стаж сотрудника
<pre>CREATE DEFINER='Ревизор'@`%` PROCEDURE `delete_employee`() BEGIN delete from employee where id_employee=4; END</pre>	Удаляет сотрудника по идентификатору
<pre>add_scanner(name varchar(30), operation_system(30), producer(30))</pre>	Добавляет сканнер в базу данных

<pre>CREATE DEFINER=`Ревизор`@`%` PROCEDURE `scanner_producer`() BEGIN update scanner set producer='Maxwell' where id_scanner=4; END</pre>	Изменяет производителя сканнера
<pre>CREATE DEFINER=`Ревизор`@`%` PROCEDURE `scanner_name`() BEGIN update scanner set name='BPM' where id_scanner=5; END</pre>	Изменяет название сканнера
<pre>CREATE DEFINER=`Ревизор`@`%` PROCEDURE `scanner_operation_system`() BEGIN update scanner set operation_system='WAT' where id_scanner=1; END</pre>	Изменяет операционную систему сканнера
<pre>CREATE DEFINER=`Ревизор`@`%` PROCEDURE `delete_scanner`() BEGIN delete from scanner where id_scanner=2; END</pre>	Удалить сканнер по идентификатору
<pre>CREATE DEFINER=`root`@`%` PROCEDURE `add_item`(name VARCHAR(30), amount int, trend VARCHAR(30), colour VARCHAR(30), sex VARCHAR(30), structure VARCHAR(30)) BEGIN insert into items values (name, amount, trend, colour, sex, structure); END</pre>	Добавляет товар в базу данных
<pre>CREATE DEFINER=`Ревизор`@`%` PROCEDURE `delete_item`()</pre>	Удаляет товар по идентификатору

BEGIN delete from items where id_item=6; END	
CREATE DEFINER='Ревизор'@`%` PROCEDURE `item_name`() BEGIN update item set name='Валенок' Where id_item=5; END	Изменяет название товара
CREATE DEFINER='Ревизор'@`%` PROCEDURE `item_trend`() BEGIN update item set trend='Спорт' Where id_item=2; END	Изменяет направление товара
CREATE DEFINER='Ревизор'@`%` PROCEDURE `item_amount`() BEGIN update item set amount='456' Where id_item=4; END	Изменяет количество товара

4. Написать SQL-код создания пользователей.

- 1) Создание пользователя «Ревизор»

```
CREATE USER 'Ревизор'@'localhost' IDENTIFIED BY '1234567';
```

Присваиваем ему все права

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

- 2) Создание пользователя «Сотрудник»

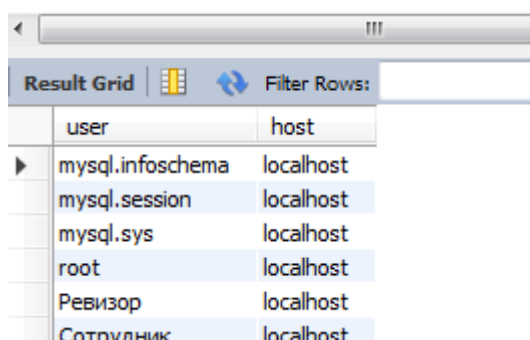
```
CREATE USER 'Сотрудник'@'localhost' IDENTIFIED BY '1234567S';
```

Присваиваем ему все права

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

Проверяем получение прав администратора с помощью команды
`select user, host from user;`

```
1 • select user, host from user;  
2
```



The screenshot shows a MySQL query result grid with two columns: 'user' and 'host'. The results are as follows:

user	host
mysql.infoschema	localhost
mysql.session	localhost
mysql.sys	localhost
root	localhost
Ревизор	localhost
Сотрудник	localhost