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 3ИП-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.**
2. 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**

**);**

1. Код написания таблицы «Emloyee».

**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**

**);**

1. 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**

**);**

1. 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**

**);**

1. 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)**

**);**

1. **Написать SQL-код создания представлений.**
2. 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`.`expirience` AS `expirience`,**

**`auditor`.`id\_scanner` AS `id\_scanner`**

**FROM**

**`auditor`**

1. 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`.`expirience` AS `expirience`,**

**`s`.`id\_section` AS `id\_section`**

**FROM**

**(`employee` `e`**

**JOIN `section` `s` ON ((`e`.`id\_employee` = `s`.`id\_employee`)))**

**ORDER BY `e`.`first\_name`**

1. **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), expirience 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 |
| 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 | Изменяет фамилию ревизора |
| CREATE DEFINER=`Ревизор`@`%` PROCEDURE `auditor\_middle\_name`()  BEGIN  update auditor set middle\_name='Овалов'  where id\_auditor=2;  END | Изменяет отчество ревизора |
| CREATE DEFINER=`Ревизор`@`%` PROCEDURE `expirience`()  BEGIN  UPDATE auditor set expirience=12  WHERE id\_auditor=4;  END | Изменяет стаж ревизора |
| CREATE DEFINER=`Ревизор`@`%` PROCEDURE `scanner`()  BEGIN  UPDATE auditor set id\_scanner=4  WHERE id\_auditor=2;  END | Изменяет сканнер ревизора |
| CREATE DEFINER=`Ревизор`@`%` PROCEDURE `delete\_auditor`()  BEGIN  delete from auditor  where id\_auditor=2;  END | Удаляет ревизора по идентификатору |
| add\_employee(first\_name varchar(30), last\_name varchar(30), middle\_name varchar(30), expirience float(11)) | Добавляет сотрудника в базу данных |
| CREATE DEFINER=`Ревизор`@`%` PROCEDURE `employee\_first\_name`()  BEGIN  update employee set first\_name='Роман'  where id\_employee=2;  END | Изменяет имя сотрудника |
| CREATE DEFINER=`Ревизор`@`%` PROCEDURE `employee\_last\_name`()  BEGIN  update employee set last\_name='Попов'  where id\_employee=2;  END | Изменяет фамилию сотрудника |
| CREATE DEFINER=`Ревизор`@`%` PROCEDURE `employee\_middle\_name`()  BEGIN  update employee set middle\_name='Александрович'  where id\_employee=2;  END | Изменяет Отчество сотрудника |
| CREATE DEFINER=`Ревизор`@`%` PROCEDURE `employee\_expirience`()  BEGIN  update employee set expirience='5'  where id\_employee=1;  END | Изменяет стаж сотрудника |
| CREATE DEFINER=`Ревизор`@`%` PROCEDURE `delete\_employee`()  BEGIN  delete from employee  where id\_employee=4;  END | Удаляет сотрудника по идентификатору |
| add\_scanner(name varchar(30), operation\_system(30), producer(30)) | Добавляет сканнер в базу данных |
| CREATE DEFINER=`Ревизор`@`%` PROCEDURE `scanner\_producer`()  BEGIN  update scanner set producer='Maxwell'  where id\_scanner=4;  END | Изменяет производителя сканнера |
| CREATE DEFINER=`Ревизор`@`%` PROCEDURE `scanner\_name`()  BEGIN  update scanner set name='BPM'  where id\_scanner=5;  END | Изменяет название сканнера |
| CREATE DEFINER=`Ревизор`@`%` PROCEDURE `scanner\_operation\_system`()  BEGIN  update scanner set operation\_system='WAT'  where id\_scanner=1;  END | Изменяет операционную систему сканнера |
| CREATE DEFINER=`Ревизор`@`%` PROCEDURE `delete\_scanner`()  BEGIN  delete from scanner  where id\_scanner=2;  END | Удалить сканнер по идентификатору |
| 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 | Добавляет товар в базу данных |
| CREATE DEFINER=`Ревизор`@`%` PROCEDURE `delete\_item`()  BEGIN  delete from items  where id\_item=6;  END | Удаляет товар по идентификатору |
| 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 |

1. **Написать SQL-код создания пользователей.**
2. Создание пользователя «Ревизор»

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

Присваиваем ему все права

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

1. 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;

