Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет» Кафедра ИИТ

Лабораторная работа №9

Выполнил:

Студент группы ПО-5

Пищик А. В.

Проверил:

Крощенко А. А.

Брест 2022

**Вариант 11**

**Цель**: приобрести практические навыки разработки баз данных и начальной интеграции БД с кодом Java с помощью JDBC.

**Задание**

Реализовать базу данных из не менее 5 таблиц на заданную тематику. При реализации продумать типизацию полей и внешние ключи в таблицах. Визуализировать разработанную БД с помощью схемы, на которой отображены все таблицы и связи между ними (пример, схема на рис. 1). На языке Java с использованием JDBC реализовать подключение к БД и выполнить основные типы запросов, продемонстрировать результаты преподавателю и включить тексты составленных запросов в отчет. Основные типы запросов – 1. На выборку/на выборку с упорядочиванием (SELECT); 2. На добавление (INSERT INTO); 3. На удаление (DELETE FROM); 4. На модификацию (UPDATE).

11) База данных «Системный администратор»

**Текст программы:**

**Main.java**

import java.sql.\*;  
  
public class Main {  
  
 public static void main(String[] args) throws SQLException, ClassNotFoundException {  
 String url = "jdbc:mysql://localhost:3306/java", user = "root", passwd = "root";  
 Class.*forName*("com.mysql.cj.jdbc.Driver");  
 try(Connection connection = DriverManager.*getConnection*(url, user, passwd)) {  
 System.*out*.println("Completed!");  
 Statement statement = connection.createStatement();  
 try {  
 ResultSet result;  
 result = statement.executeQuery("SELECT \* FROM `Company` WHERE 1");  
  
 while (result.next()) {  
 String name = result.getString("Company.name");  
 System.*out*.println(  
 result.getString("Company.id") +  
 " " + result.getString("Company.name")  
 + " " + name);  
 }  
 }  
 catch (Exception exception) {  
 System.*out*.println(exception);  
 }  
 }  
 }  
  
 public void createSourceDB() { }  
}

**Schema.sql**

CREATE TABLE `java`.`Company` ( `id` INT NOT NULL AUTO\_INCREMENT , `name` VARCHAR(255) NOT NULL , PRIMARY KEY (`id`)) ENGINE = InnoDB;

CREATE TABLE `java`.`Partments` ( `id` INT NOT NULL AUTO\_INCREMENT , `partment` INT NOT NULL , `name` VARCHAR(255) NOT NULL , PRIMARY KEY (`id`)) ENGINE = InnoDB;

ALTER TABLE `Partments` ADD FULLTEXT(`name`);

ALTER TABLE `Partments` ADD FOREIGN KEY (`partment`) REFERENCES `Company`(`id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

ALTER TABLE `Partments` ADD INDEX(`name`);

CREATE TABLE `java`.`Adress` ( `id` INT NOT NULL AUTO\_INCREMENT , `name` VARCHAR(255) NOT NULL , `adress` VARCHAR(255) NOT NULL , PRIMARY KEY (`id`)) ENGINE = InnoDB;

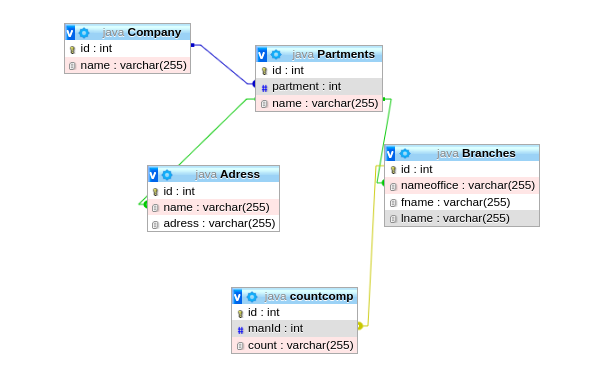
ALTER TABLE `Adress` ADD FOREIGN KEY (`name`) REFERENCES `Partments`(`name`) ON DELETE RESTRICT ON UPDATE RESTRICT;

CREATE TABLE `java`.`Branches` ( `id` INT NOT NULL AUTO\_INCREMENT , `nameoffice` VARCHAR(255) NOT NULL , `fname` VARCHAR(255) NOT NULL , `lname` VARCHAR(255) NOT NULL , PRIMARY KEY (`id`)) ENGINE = InnoDB;

ALTER TABLE `Branches` ADD FOREIGN KEY (`nameoffice`) REFERENCES `Partments`(`name`) ON DELETE RESTRICT ON UPDATE RESTRICT;

CREATE TABLE `java`.`countcomp` ( `id` INT NOT NULL AUTO\_INCREMENT , `manId` INT NOT NULL , `count` VARCHAR(255) NOT NULL , PRIMARY KEY (`id`)) ENGINE = InnoDB;

ALTER TABLE `countcomp` ADD FOREIGN KEY (`manId`) REFERENCES `Branches`(`id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

****

**Вывод:** приобрел практические навыки разработки баз данных и начальной интеграции БД с кодом Java с помощью JDBC.