# **Лабораторная работа №3**

**Разработка клиент-серверного консольного приложения автоматического создания базы данных на языке java**

Гусев В.Е.

ИС-З-21

**Код программы:**

import java.sql.\*;

import java.util.Scanner;

public class lab3 {

    private static final String URL = "jdbc:mariadb://localhost:3306/";

    private static final String USER = "admin";

    private static Connection connection;

    public static void main(String[] args) {

        try {

            Class.forName("org.mariadb.jdbc.Driver");

            Scanner scanner = new Scanner(System.in);

            System.out.print("Введите пароль пользователя admin: ");

            String password = scanner.nextLine();

            connection = DriverManager.getConnection(URL, USER, password);

            System.out.println("\n✅ Успешное подключение к серверу MariaDB/MySQL!");

            try (Statement stmt = connection.createStatement();

                 ResultSet rs = stmt.executeQuery("SELECT VERSION()")) {

                if (rs.next()) {

                    System.out.println("Версия сервера: " + rs.getString(1));

                }

            }

            showMenu();

        } catch (ClassNotFoundException e) {

            System.err.println("\n❌ Ошибка: Драйвер MariaDB не найден!");

        } catch (SQLException e) {

            System.err.println("\n❌ Ошибка подключения:");

            System.err.println(e.getMessage());

        } finally {

            if (connection != null) {

                try {

                    connection.close();

                } catch (SQLException e) {

                    System.err.println("Ошибка при закрытии соединения: " + e.getMessage());

                }

            }

        }

    }

    private static void showMenu() throws SQLException {

        Scanner scanner = new Scanner(System.in);

        int choice;

        do {

            System.out.println("\n=== МЕНЮ ===");

            System.out.println("1. Вывести список баз данных");

            System.out.println("2. Создать базу данных и таблицы");

            System.out.println("3. Просмотреть структуру базы данных");

            System.out.println("4. Удалить базу данных");

            System.out.println("5. Выйти из программы");

            System.out.print("Выберите пункт: ");

            try {

                choice = Integer.parseInt(scanner.nextLine());

            } catch (NumberFormatException e) {

                System.out.println("Ошибка: введите число от 1 до 5");

                choice = 0;

                continue;

            }

            switch (choice) {

                case 1:

                    showDatabases();

                    break;

                case 2:

                    createDatabaseAndTables();

                    break;

                case 3:

                    viewDatabaseStructure();

                    break;

                case 4:

                    deleteDatabase();

                    break;

                case 5:

                    System.out.println("Выход из программы...");

                    break;

                default:

                    System.out.println("Неверный выбор. Попробуйте снова.");

            }

        } while (choice != 5);

    }

    private static void showDatabases() throws SQLException {

        System.out.println("\nСписок баз данных:");

        try (Statement stmt = connection.createStatement();

             ResultSet rs = stmt.executeQuery("SHOW DATABASES")) {

            while (rs.next()) {

                System.out.println("- " + rs.getString(1));

            }

        }

    }

    private static void createDatabaseAndTables() throws SQLException {

        try (Statement stmt = connection.createStatement()) {

            stmt.execute("CREATE DATABASE IF NOT EXISTS Gusev");

            System.out.println("База данных Gusev создана");

            stmt.execute("USE Gusev");

            stmt.execute("CREATE TABLE IF NOT EXISTS readers (" +

                "readers\_id BIGINT AUTO\_INCREMENT, " +

                "fam VARCHAR(50) NOT NULL, " +

                "name VARCHAR(50) NOT NULL, " +

                "otch VARCHAR(50) NOT NULL, " +

                "address VARCHAR(255) NOT NULL, " +

                "phone VARCHAR(11) NOT NULL, " +

                "card\_num VARCHAR(10) NOT NULL, " +

                "PRIMARY KEY (readers\_id))");

            System.out.println("Таблица readers создана");

            stmt.execute("CREATE TABLE IF NOT EXISTS books (" +

                "books\_id BIGINT AUTO\_INCREMENT, " +

                "name VARCHAR(50) NOT NULL, " +

                "genre VARCHAR(50) NOT NULL, " +

                "publishing VARCHAR(50) NOT NULL, " +

                "rel\_year YEAR NOT NULL, " +

                "isbn VARCHAR(13) NOT NULL, " +

                "inv\_num VARCHAR(10) NOT NULL, " +

                "state VARCHAR(20) NOT NULL, " +

                "binding VARCHAR(20) NOT NULL, " +

                "authors\_id BIGINT NOT NULL, " +

                "PRIMARY KEY (books\_id))");

            System.out.println("Таблица books создана");

            stmt.execute("CREATE TABLE IF NOT EXISTS book\_list (" +

                "list\_id BIGINT AUTO\_INCREMENT, " +

                "readers\_id BIGINT NOT NULL, " +

                "datcis DATE NOT NULL, " +

                "dateret DATE NOT NULL, " +

                "books\_id BIGINT NOT NULL, " +

                "PRIMARY KEY (list\_id))");

            System.out.println("Таблица book\_list создана");

            stmt.execute("CREATE TABLE IF NOT EXISTS authors (" +

                "authors\_id BIGINT AUTO\_INCREMENT, " +

                "fam VARCHAR(50) NOT NULL, " +

                "name VARCHAR(50) NOT NULL, " +

                "otch VARCHAR(50) NOT NULL, " +

                "PRIMARY KEY (authors\_id))");

            System.out.println("Таблица authors создана");

        }

    }

    private static void viewDatabaseStructure() throws SQLException {

        try (Statement stmt = connection.createStatement()) {

            stmt.execute("USE Gusev");

            System.out.println("\nТаблицы в базе данных Gusev:");

            try (ResultSet rs = stmt.executeQuery("SHOW TABLES")) {

                while (rs.next()) {

                    System.out.println("- " + rs.getString(1));

                }

            }

            describeTable(stmt, "books");

            describeTable(stmt, "readers");

            describeTable(stmt, "book\_list");

            describeTable(stmt, "authors");

        } catch (SQLException e) {

            System.out.println("Ошибка: база данных Gusev не существует или недоступна");

        }

    }

    private static void describeTable(Statement stmt, String tableName) throws SQLException {

        System.out.println("\nСтруктура таблицы " + tableName + ":");

        try (ResultSet rs = stmt.executeQuery("DESCRIBE " + tableName)) {

            System.out.println("+------------+--------------+------+-----+---------+----------------+");

            System.out.println("| Field      | Type         | Null | Key | Default | Extra          |");

            System.out.println("+------------+--------------+------+-----+---------+----------------+");

            while (rs.next()) {

                System.out.printf("| %-10s | %-12s | %-4s | %-3s | %-7s | %-14s |\n",

                    rs.getString("Field"),

                    rs.getString("Type"),

                    rs.getString("Null"),

                    rs.getString("Key"),

                    rs.getString("Default") != null ? rs.getString("Default") : "NULL",

                    rs.getString("Extra"));

                System.out.println("+------------+--------------+------+-----+---------+----------------+");

            }

        }

    }

    private static void deleteDatabase() throws SQLException {

        try (Statement stmt = connection.createStatement()) {

            stmt.execute("DROP DATABASE IF EXISTS Gusev");

            System.out.println("База данных Gusev удалена");

        } catch (SQLException e) {

            System.out.println("Ошибка при удалении базы данных: " + e.getMessage());

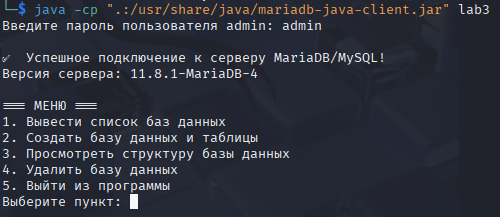
        }

    }

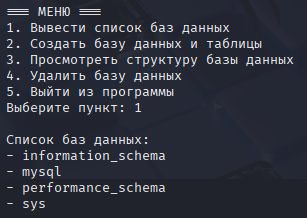
}

**Скриншоты работы программы:**

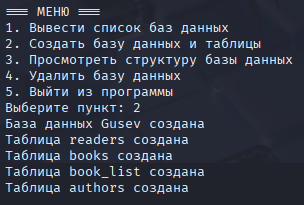
Запуск программы и ввод пароля от пользователя



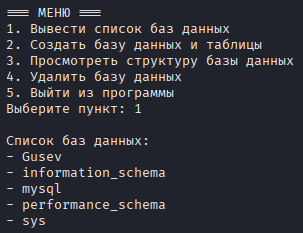
Вводим цифру 1 для вывода списка баз данных



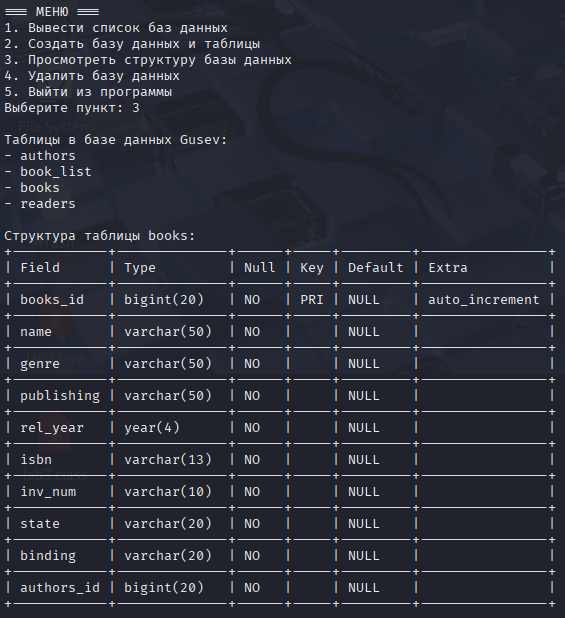
Вводим цифру 2 для создания нашей базы данных «Gusev»

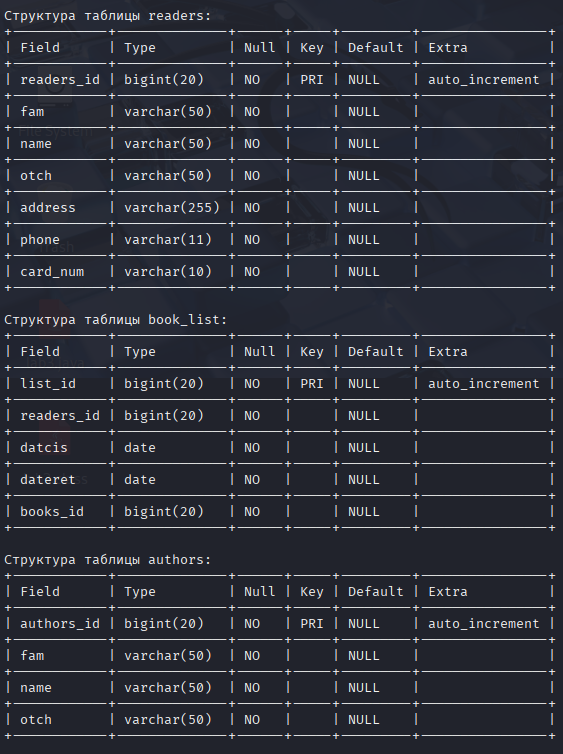


Вводим цифру 1 для вывода списка баз данных. Как видим, наша БД создалась.

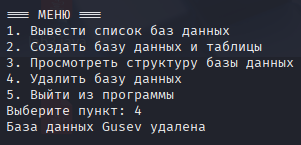


Вводим цифру 3 для вывода структуры нашей базы данных «Gusev»:

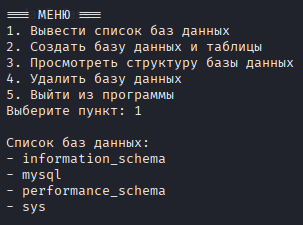




Вводим цифру 4 для удаления нашей базы данных «Gusev»:



Вводим цифру 1 для вывода списка баз данных. Как видим, наша БД «Gusev» удалена и ее нет в списке:



Вводим цифру 5 для выхода из программы:

