# **Лабораторная работа №4**

**Разработка клиент-серверного консольного приложения управления учетными записями на языке java**

Гусев В.Е.

ИС-З-21

**Код программы:**

import java.sql.\*;

import java.util.Scanner;

public class lab4 {

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

    private static final String ADMIN\_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\_USER + ": ");

            String password = scanner.nextLine();

            connection = DriverManager.getConnection(URL, ADMIN\_USER, password);

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

            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=== УПРАВЛЕНИЕ УЧЕТНЫМИ ЗАПИСЯМИ ===");

            System.out.println("1. Показать список пользователей");

            System.out.println("2. Добавить пользователя");

            System.out.println("3. Показать права пользователя");

            System.out.println("4. Удалить пользователя");

            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:

                    showUsers();

                    break;

                case 2:

                    addUser(scanner);

                    break;

                case 3:

                    showUserPrivileges(scanner);

                    break;

                case 4:

                    deleteUser(scanner);

                    break;

                case 5:

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

                    break;

                default:

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

            }

        } while (choice != 5);

    }

    private static void showUsers() throws SQLException {

        System.out.println("\nСписок пользователей:");

        try (Statement stmt = connection.createStatement();

             ResultSet rs = stmt.executeQuery("SELECT User FROM mysql.user WHERE User NOT LIKE 'mysql.%'")) {

            int count = 1;

            while (rs.next()) {

                System.out.println(count++ + ". " + rs.getString("User"));

            }

        }

    }

    private static void addUser(Scanner scanner) throws SQLException {

        System.out.print("\nВведите имя нового пользователя: ");

        String username = scanner.nextLine();

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

        String password = scanner.nextLine();

        System.out.println("Выберите права (можно комбинировать цифры, например 234):");

        System.out.println("1 - ALL PRIVILEGES");

        System.out.println("2 - SELECT");

        System.out.println("3 - INSERT");

        System.out.println("4 - UPDATE");

        System.out.println("5 - DELETE");

        System.out.print("Введите права: ");

        String privileges = scanner.nextLine();

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

            stmt.executeUpdate("CREATE USER '" + username + "'@'localhost' IDENTIFIED BY '" + password + "'");

            if (privileges.contains("1")) {

                stmt.executeUpdate("GRANT ALL PRIVILEGES ON \*.\* TO '" + username + "'@'localhost'");

            } else {

                StringBuilder grantQuery = new StringBuilder("GRANT ");

                if (privileges.contains("2")) grantQuery.append("SELECT, ");

                if (privileges.contains("3")) grantQuery.append("INSERT, ");

                if (privileges.contains("4")) grantQuery.append("UPDATE, ");

                if (privileges.contains("5")) grantQuery.append("DELETE, ");

                grantQuery.setLength(grantQuery.length() - 2);

                grantQuery.append(" ON \*.\* TO '").append(username).append("'@'localhost'");

                stmt.executeUpdate(grantQuery.toString());

            }

            stmt.executeUpdate("FLUSH PRIVILEGES");

            System.out.println("✅ Пользователь '" + username + "' успешно создан с указанными правами");

        } catch (SQLException e) {

            System.out.println("❌ Ошибка при создании пользователя: " + e.getMessage());

        }

    }

    private static void showUserPrivileges(Scanner scanner) throws SQLException {

        showUsers();

        System.out.print("\nВведите имя пользователя для просмотра прав: ");

        String username = scanner.nextLine();

        try (Statement stmt = connection.createStatement();

             ResultSet rs = stmt.executeQuery("SHOW GRANTS FOR '" + username + "'@'localhost'")) {

            System.out.println("\nПрава пользователя '" + username + "':");

            while (rs.next()) {

                System.out.println(rs.getString(1));

            }

        } catch (SQLException e) {

            System.out.println("❌ Ошибка: " + e.getMessage());

        }

    }

    private static void deleteUser(Scanner scanner) throws SQLException {

        showUsers();

        System.out.print("\nВведите имя пользователя для удаления: ");

        String username = scanner.nextLine();

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

            stmt.executeUpdate("DROP USER IF EXISTS '" + username + "'@'localhost'");

            stmt.executeUpdate("FLUSH PRIVILEGES");

            System.out.println("✅ Пользователь '" + username + "' успешно удален");

        } catch (SQLException e) {

            System.out.println("❌ Ошибка при удалении пользователя: " + e.getMessage());

        }

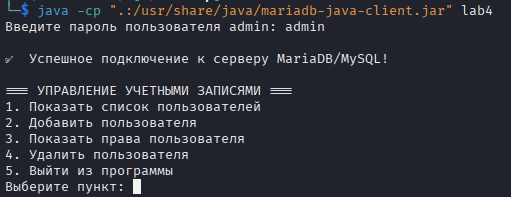
    }

}

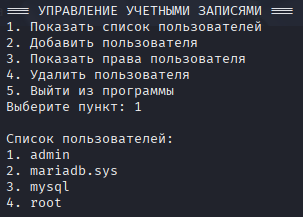
**Скриншоты работы программы:**

Компиляция и запуск программы и ввод пароля от пользователя

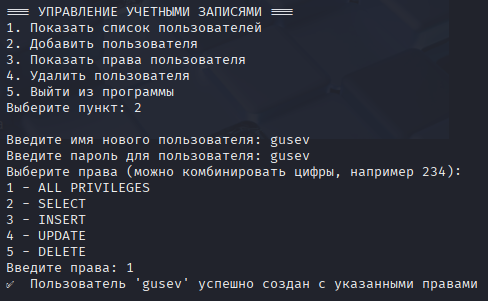




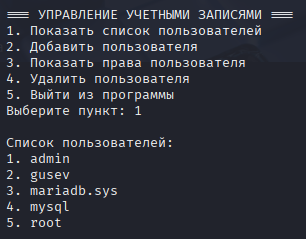
Вводим цифру 1 для вывода списка пользователей



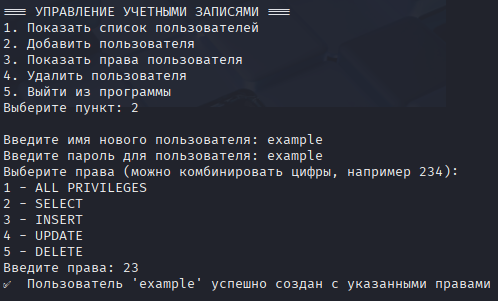
Вводим цифру 2 для добавления пользователя. Вводим имя пользователя, пароль и привилегии (например пользователь gusev с паролем gusev и со всеми привилегиями):



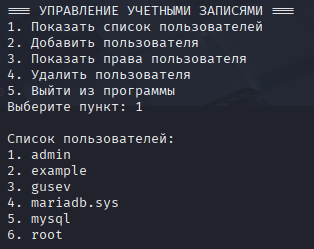
Вводим цифру 1 для вывода списка пользователей. Как видим, пользователь gusev создался.



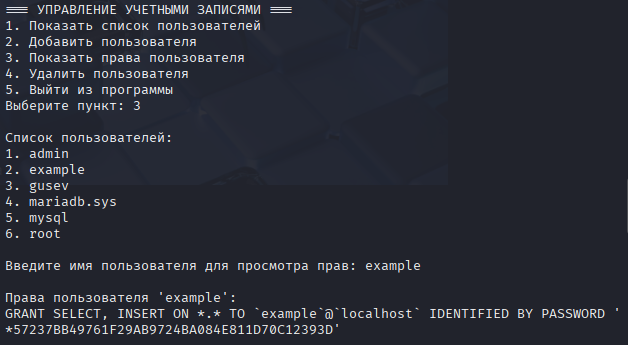
Вводим цифру 2 для добавления пользователя. Вводим имя пользователя, пароль и привилегии (В этот раз создадим пользователя example с паролем example и с привилегиями SELECT и INSERT – 23):



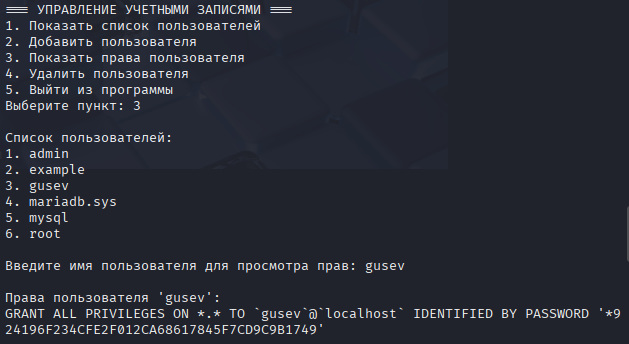
Вводим цифру 1 для вывода списка пользователей. Как видим, пользователь example создался.



Вводим цифру 3 для просмотра привилегий пользователя. Нам отображается список пользователей, нам нужно ввести наименование пользователя, у которого мы хотим просмотреть права:

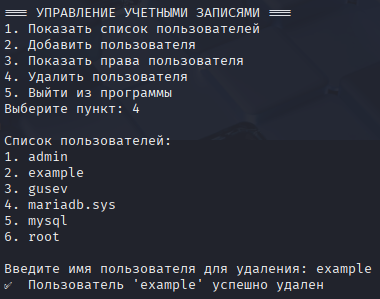


Слова «SELECT, INSERT» в строке нам дают понять, что пользователь «example» может только запрашивать и вносить данные.

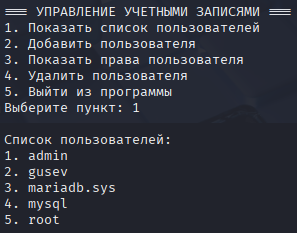


Слова «ALL PRIVILEGES» в строке нам дают понять, что пользователь «gusev» может абсолютно всё.

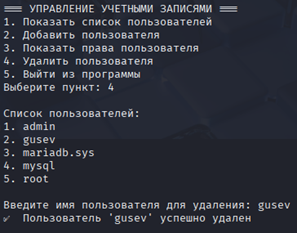
Вводим цифру 4 для удаления пользователя. Нам отображается список пользователей, нам нужно ввести наименование пользователя, которого мы хотим удалить (Удаляем пользователя example).



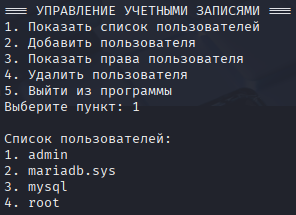
Вводим цифру 1 для просмотра списка пользователей. Как видим, пользователь example удалился.



Вводим цифру 4 для удаления пользователя. Нам отображается список пользователей, нам нужно ввести наименование пользователя, которого мы хотим удалить (Удаляем пользователя gusev).



Вводим цифру 1 для просмотра списка пользователей. Как видим, пользователь gusev удалился.



Вводим цифру 5 для выхода из программы

