Федеральное государственное бюджетное образовательное учреждение высшего образования «Российская академия народного хозяйства и государственной службы при Президенте Российской Федерации».

Кафедра «Прикладная Информатика»

ОТЧЕТ

О ПРОДЕЛАННОЙ ПРАКТИЧЕСКОЙ РАБОТЕ №15

по курсу «Базы данных»

Выполнил: студент группы

ИК-721

**(наименование группы)**

***Соколов Дмитрий Александрович***

**(Ф.И.О.)**

Нижний Новгород

**2025 г.**

Оглавление

[1 Цель выполнения работы 3](#_Toc200711963)

[2 Описание содержания выполненных работ 3](#_Toc200711964)

[3 Выводы 6](#_Toc200711966)

# 1 Цель выполнения работы

**Цель работы** — изучить основы разделения ролей. Создать пользователей для управления базой данных.

**Задание:**

Определить 2-3 должностных лица, которые смогут работать с таблицами БД. Для каждого должностного лица определить набор привилегий, которыми он может пользоваться.

# 2 Описание содержания выполненных работ

Создаём файл users.txt:

1. admin:admin:admin1
2. user:user:user1

**Алгоритм работы программы**

1. Ввод логина и пароля

При запуске программа запрашивает у пользователя логин и пароль, и сверяет их с содержимым файла.

2. Проверка роли

После успешной авторизации определяется роль пользователя:

admin — полный доступ

user — только INSERT, UPDATE

3. Ограничение доступа к операциям

Роль определяет, какие SQL-команды доступны:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| | **Роль** | **Разрешённые операции** | | --- | --- | | | **Роль** | **Разрешённые операции** | | --- | --- | |
| |  |  | | --- | --- | | admin | CREATE TABLE, DROP TABLE, INSERT, UPDATE, SELECT | | |  |  | | --- | --- | | admin | CREATE TABLE, DROP TABLE, INSERT, UPDATE, SELECT | |
| |  |  | | --- | --- | | user | INSERT, UPDATE, SELECT | | |  |  | | --- | --- | | user | INSERT, UPDATE, SELECT | |

**Код программы:**

public class AuthService {

private static Map<String, User> users = new HashMap<>();

public static void loadUsers(String filename) throws IOException {

List<String> lines = Files.readAllLines(Paths.get(filename));

for (String line : lines) {

String[] parts = line.split(":");

if (parts.length == 3) {

users.put(parts[0], new User(parts[0], parts[1], parts[2]));

}

}

}

public static User login(Scanner scanner) {

System.out.print("Login: ");

String login = scanner.nextLine();

System.out.print("Password: ");

String password = scanner.nextLine();

User user = users.get(login);

if (user != null && user.getPassword().equals(password)) {

return user;

} else {

System.out.println("Неверный логин или пароль");

return null;

}

}

}

class User {

private final String username;

private final String password;

private final String role;

public User(String username, String password, String role) {

this.username = username;

this.password = password;

this.role = role;

}

public String getPassword() { return password; }

public String getRole() { return role; }

}

public class RoleChecker {

public static boolean canCreateTable(String role) {

return "admin".equals(role);

}

public static boolean canInsert(String role) {

return "admin".equals(role) || "editor".equals(role);

}

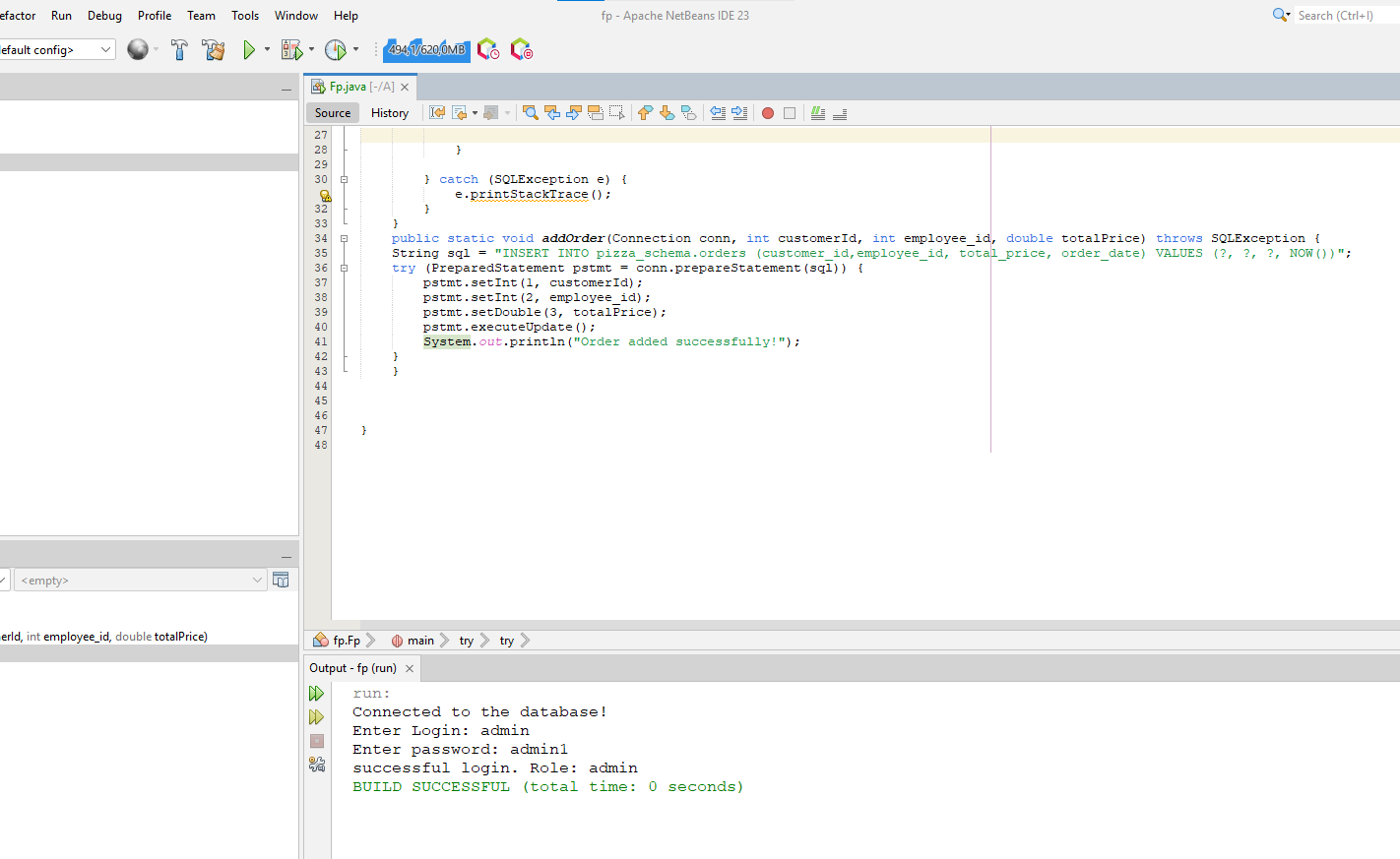
public static boolean canSelect(String role) {

return true; // Все роли могут SELECT

}

}

**Результат работы:**



# 3 Выводы

В ходе выполнения данной работы была реализована простая система авторизации и контроля доступа пользователей в Java-приложении с использованием базы данных PostgreSQL. Несмотря на то, что PostgreSQL поддерживает встроенное управление пользователями и правами, для целей гибкости и наглядности был использован внешний файл (users.txt) для хранения логинов, паролей и ролей.

Были реализованы следующие функции:

1. Загрузка учетных данных пользователей из текстового файла;
2. Авторизация пользователя при запуске программы;
3. Назначение ролей (admin, user) с различными уровнями доступа;
4. Контроль выполнения SQL-операций (например, SELECT, INSERT, UPDATE, CREATE TABLE) в зависимости от роли пользователя.

Таким образом, программа демонстрирует, как можно реализовать многопользовательский доступ и разграничение прав на уровне бизнес-логики Java-приложения без использования механизмов безопасности самой СУБД.