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 №16

по курсу «Базы данных»

Выполнил: студент группы

ИК-721

**(наименование группы)**

***Соколов Дмитрий Александрович***

**(Ф.И.О.)**

## Нижний Новгород

**2024 г.**

Оглавление

[1 Цель выполнения работы 3](#_Toc200714634)

[2 Описание содержания выполненных работ 3](#_Toc200714636)

[3 Выводы 11](#_Toc200714637)

# 1 Цель выполнения работы

# **Цель работы** — Разработка клиентского приложения на языке Java для учета хоккейных матчей с использованием базы данных PostgreSQL. Целью является демонстрация возможностей SQL в построении клиент-серверного приложения, работающего с распределённой базой данных через JDBC.

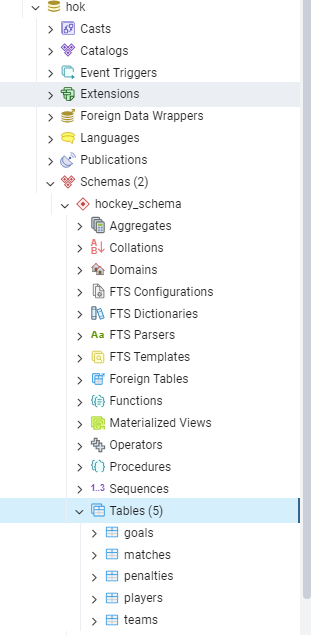
**Задание:**

Согласно своему варианту разработайте клиентское приложение, соответствующее следующим предметным областям: (Вариант 18) Клиентское приложение для учета хоккейных матчей: учет результатов игр (забитые голы, штрафы).

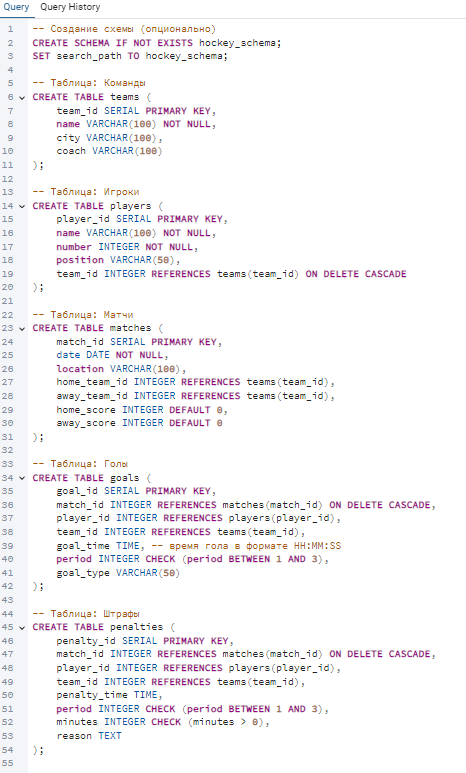
# 2 Описание содержания выполненных работ

**Создание базы данных и таблиц**

В ходе выполнения работы была создана схема со всеми необходимыми таблицами и связями.

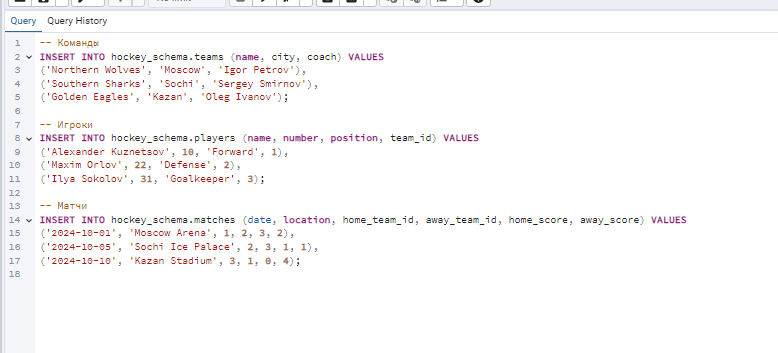


Скрипт на создание таблиц в базе:



**Создание клиентского приложения для работы с базой на Java:**

**Заполнение базы данными:**



Создание интерфейса на Swing (Основной файл):

// HockeyApp.java

import javax.swing.\*;

import java.awt.event.\*;

public class HockeyApp {

public static void main(String[] args) {

JFrame frame = new JFrame("Учет хоккейных матчей");

frame.setSize(400, 200);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setLocationRelativeTo(null);

JPanel panel = new JPanel();

JButton btnTeams = new JButton("Показать команды");

JButton btnPlayers = new JButton("Показать игроков команды");

JButton btnMatches = new JButton("Показать матчи");

panel.add(btnTeams);

panel.add(btnPlayers);

panel.add(btnMatches);

// Кнопка: Показать команды

btnTeams.addActionListener(e -> HockeyDB.getTeams());

// Кнопка: Показать игроков команды

btnPlayers.addActionListener(e -> {

String teamIdStr = JOptionPane.showInputDialog("Введите ID команды:");

if (teamIdStr != null && !teamIdStr.isEmpty()) {

int teamId = Integer.parseInt(teamIdStr);

HockeyDB.getPlayersByTeam(teamId);

}

});

// Кнопка: Показать матчи

btnMatches.addActionListener(e -> HockeyDB.getMatches());

frame.add(panel);

frame.setVisible(true);

}

}

**Класс для работы с базой:**

// HockeyDB.java

import java.sql.\*;

public class HockeyDB {

private static final String URL = "jdbc:postgresql://localhost:5432/your\_database";

private static final String USER = "your\_user";

private static final String PASSWORD = "your\_password";

private static Connection connect() throws SQLException {

return DriverManager.getConnection(URL, USER, PASSWORD);

}

public static void getTeams() {

String sql = "SELECT team\_id, name, city, coach FROM hockey\_schema.teams";

try (Connection conn = connect();

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery(sql)) {

System.out.println("Список команд:");

while (rs.next()) {

System.out.printf("ID: %d | Название: %s | Город: %s | Тренер: %s%n",

rs.getInt("team\_id"),

rs.getString("name"),

rs.getString("city"),

rs.getString("coach"));

}

} catch (SQLException e) {

e.printStackTrace();

}

}

public static void getPlayersByTeam(int teamId) {

String sql = "SELECT name, number, position FROM hockey\_schema.players WHERE team\_id = ?";

try (Connection conn = connect();

PreparedStatement pstmt = conn.prepareStatement(sql)) {

pstmt.setInt(1, teamId);

ResultSet rs = pstmt.executeQuery();

System.out.println("Игроки команды ID " + teamId + ":");

while (rs.next()) {

System.out.printf("Имя: %s | №%d | Позиция: %s%n",

rs.getString("name"),

rs.getInt("number"),

rs.getString("position"));

}

} catch (SQLException e) {

e.printStackTrace();

}

}

public static void getMatches() {

String sql = "SELECT m.match\_id, m.date, m.location, " +

"ht.name AS home\_team, at.name AS away\_team, " +

"m.home\_score, m.away\_score " +

"FROM hockey\_schema.matches m " +

"JOIN hockey\_schema.teams ht ON m.home\_team\_id = ht.team\_id " +

"JOIN hockey\_schema.teams at ON m.away\_team\_id = at.team\_id";

try (Connection conn = connect();

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery(sql)) {

System.out.println("Список матчей:");

while (rs.next()) {

System.out.printf("Матч #%d | %s: %s (%d) vs %s (%d)%n",

rs.getInt("match\_id"),

rs.getDate("date"),

rs.getString("home\_team"),

rs.getInt("home\_score"),

rs.getString("away\_team"),

rs.getInt("away\_score"));

}

} catch (SQLException e) {

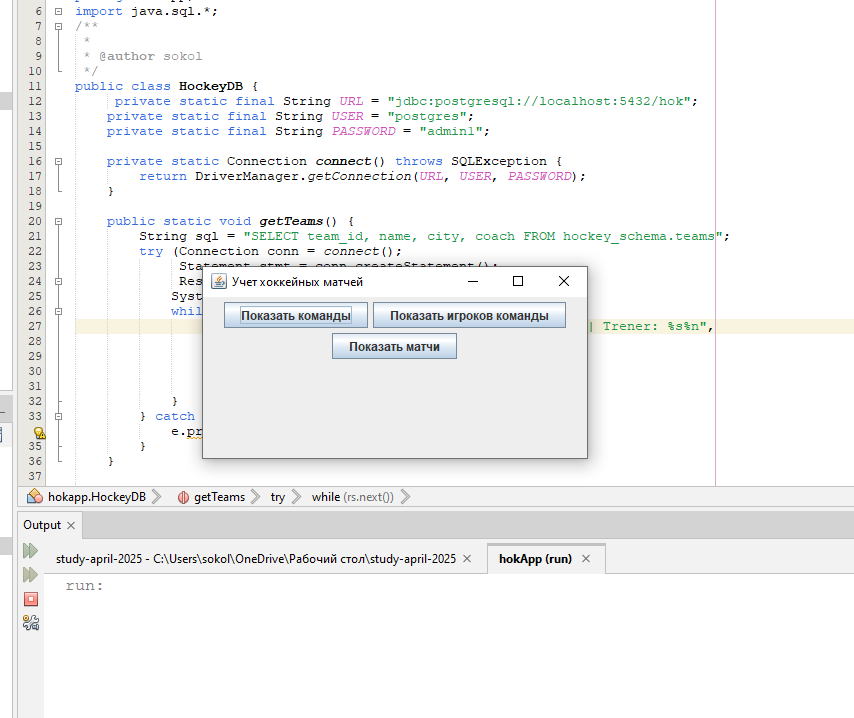
e.printStackTrace();

}

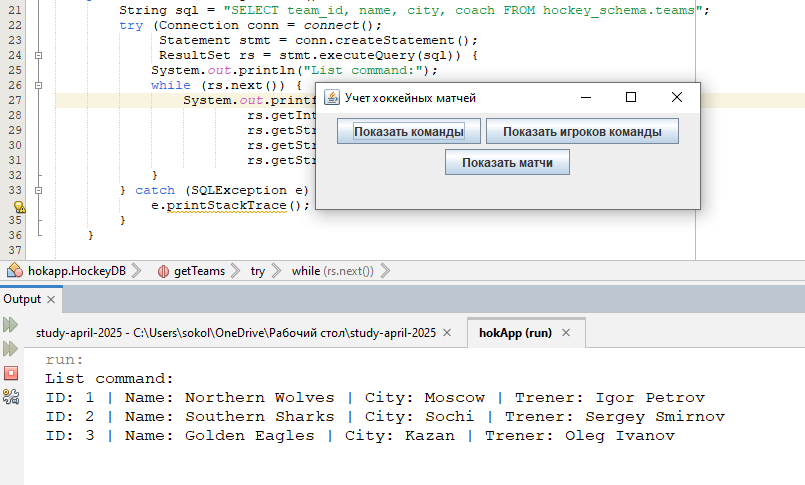
}

}

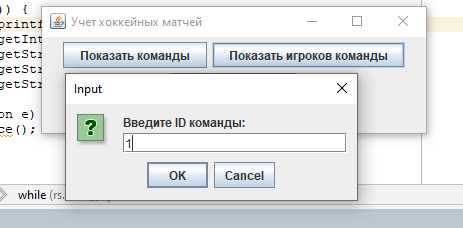
Результат (Вывод графического интерфейса):

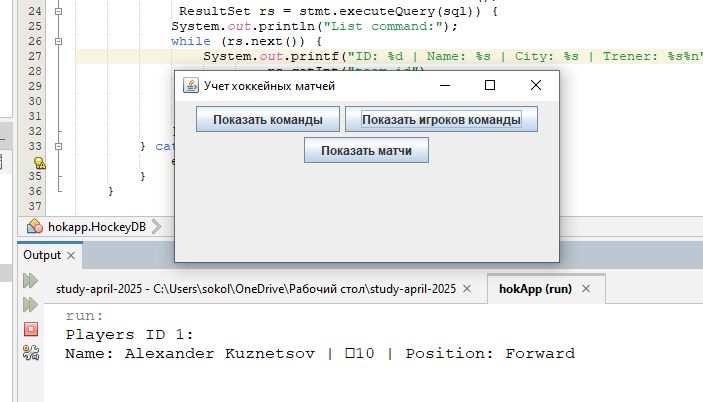


Результат (Нажатие кнопки «Показать команды»):

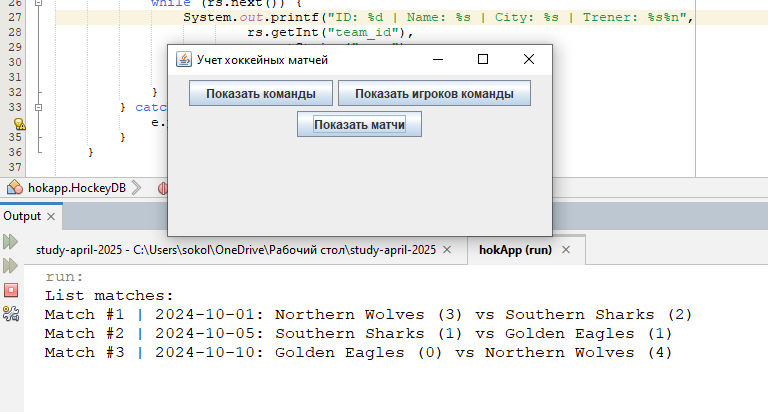


Результат (Нажатие кнопки «Показать игроков команды»):





Результат (Нажатие кнопки «Показать матчи»):



# 3 Выводы

Было создано простое клиентское приложение на Java с использованием библиотеки Swing, которое позволяет:

1. Просматривать список команд.
2. Просматривать игроков по команде.
3. Просматривать список проведённых матчей с результатами.

Пользовательский интерфейс интуитивно понятен и минималистичен (3 кнопки). Это удобно для начального уровня и быстрой навигации.