**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

отчет

**по лабораторной работе №2**

**по дисциплине «Базы данных»**

Тема: Реализация базы данных в СУБД PostgreSQL

|  |  |  |
| --- | --- | --- |
| Студентка гр. 1303 |  | Королева П.А. |
| Преподаватель |  | Заславский М.М. |

Санкт-Петербург

2023

**Цель работы.**

Развернуть локально PostgreSQL, написать запросы для создания и заполнения таблиц, написать запросы к БД, отвечающие на вопросы в задания.

**Задание.**

Вариант 12

Пусть требуется создать программную систему, предназначенную для организаторов соревнований по футболу в рамках первенства страны. Такая система должна обеспечивать хранение сведений о командах, участвующих в первенстве, об игроках команд, о расписании встреч и их результатах, о цене билетов на игры. Сведения о команде — название команды, город, где она базируется, имя главного тренера, место в таблице прошлого сезона, расписание встреч. В один день команда может участвовать только в одной встрече. Сведения об игроке включают в себя фамилию и имя игрока, его возраст, номер и амплуа в команде. Сведения о стадионе, на котором происходит встреча содержат город, в котором он находится, название стадиона, и его вместимость. Цена билета на матч зависит от вместимости стадиона и положения встречающихся команд в турнирной таблице прошлого сезона (наибольшая - при игре тройки призеров, наименьшая — при игре тройки аутсайдеров). Организаторы соревнований должны иметь возможность внести изменения в данные о составе команд, перенести встречу. Им могут потребоваться следующие сведения:

* Даты встреч указанной команды, ее противники и счет?
* Номера и фамилии игроков команд, участвовавших во встрече, которая проходила в указанный день в указанном городе?
* Цена, билета на матч между указанными командами?
* Игрок, забивший в турнире наибольшее количество мячей?
* Команды, имеющие наилучшую и наихудшую разницу забитых и пропущенных мячей?
* Самый молодой участник турнира?
* Команды, занявшие призовые места?

**Выполнение работы.**

Создана база данных «football\_manager», ее наполнение представлено на рисунке 1. Попутно созданы 6 схем «Игрок», «Команда», «Команды\_в\_игре», «Организатор», «Расписание\_игр», «Стадион».

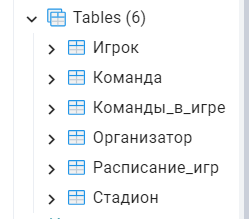
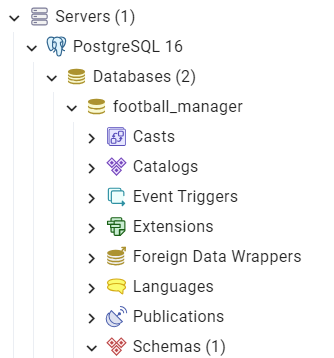


Рисунок 1. БД «football\_manager»

Схемы заполнены значениями, в каждой 5-10 кортежей. Результат представлен на рисунках 2-7.



Рисунок 2 – Наполнение схемы «Игрок»

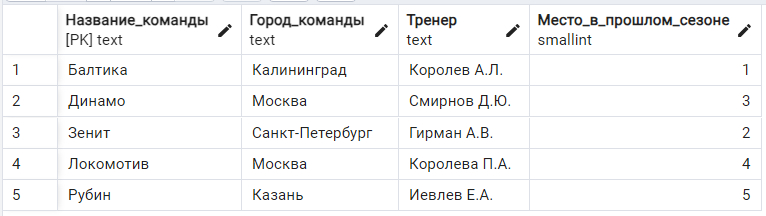


Рисунок 3 – Наполнение схемы «Команда»

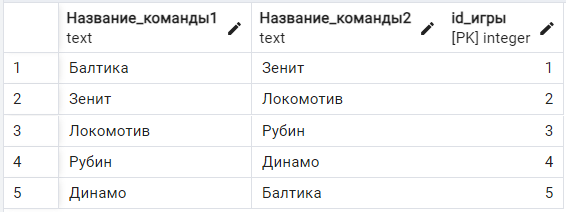


Рисунок 4 – Наполнение схемы «Команды\_в\_игре»

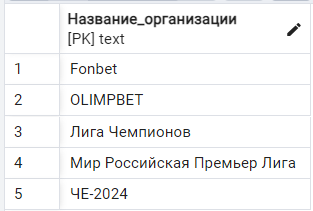


Рисунок 5 – Наполнение схемы «Организатор»



Рисунок 6 – Наполнение схемы «Расписание\_игр»

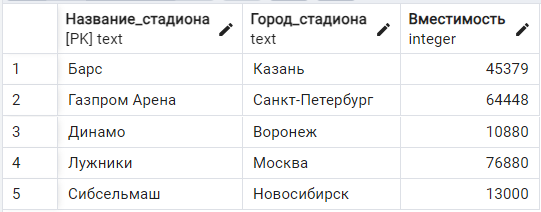


Рисунок 7 – Наполнение схемы «Стадион»

Написаны запросы для ответа на вопросы в задании. Результаты приведены на рисунках 8-14.

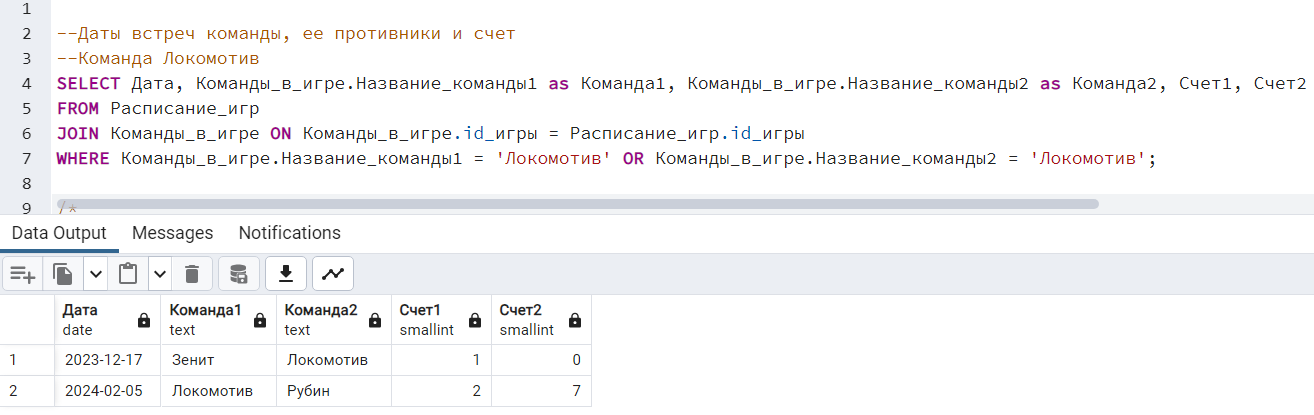


Рисунок 8. Даты встреч команды, ее противники и счет.

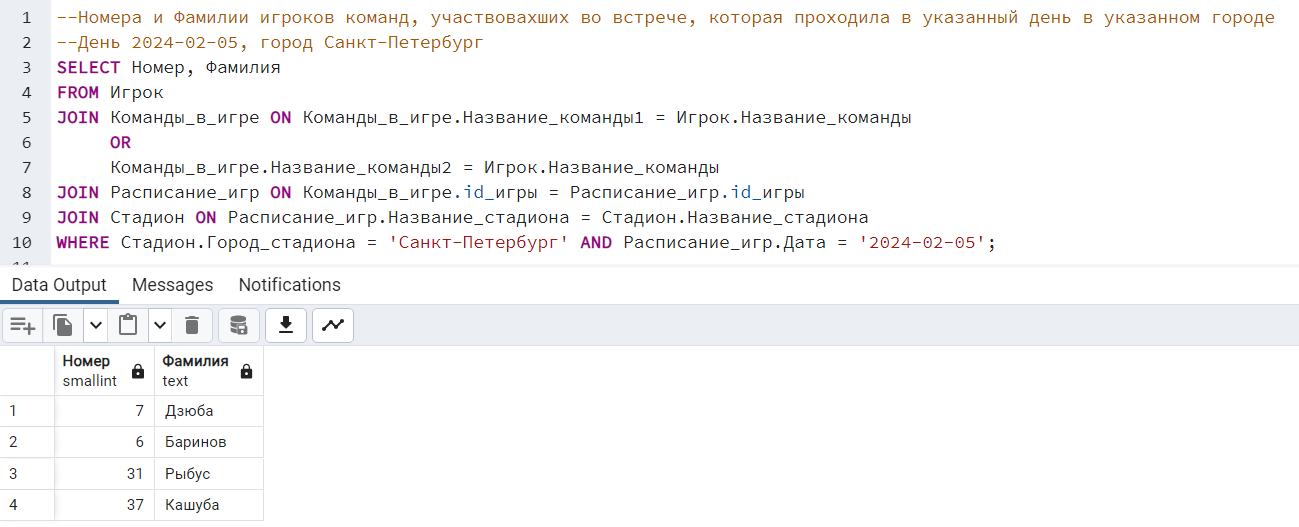


Рисунок 9. Номера и фамилии игроков команд, участвовавших во встрече, которая проходила в указанный день в указанном городе.

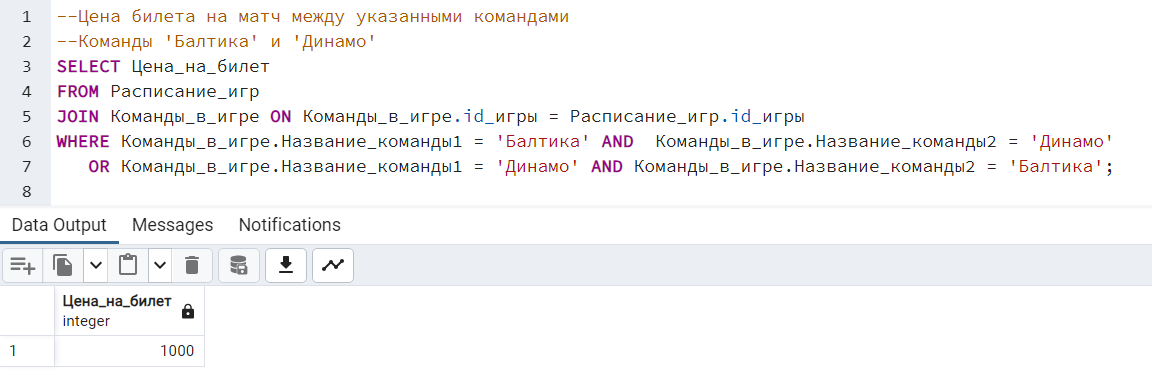


Рисунок 10. Цена, билета на матч между указанными командами.

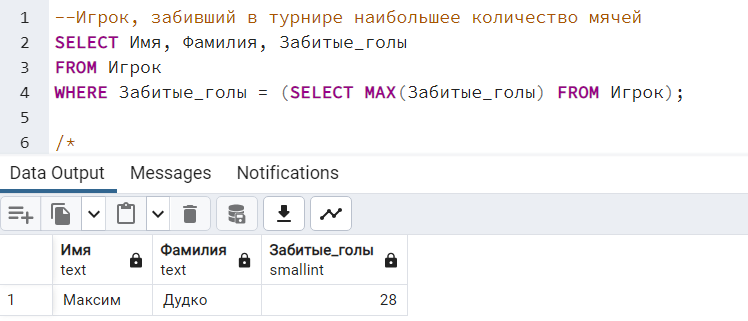


Рисунок 11. Игрок, забивший в турнире наибольшее количество мячей.

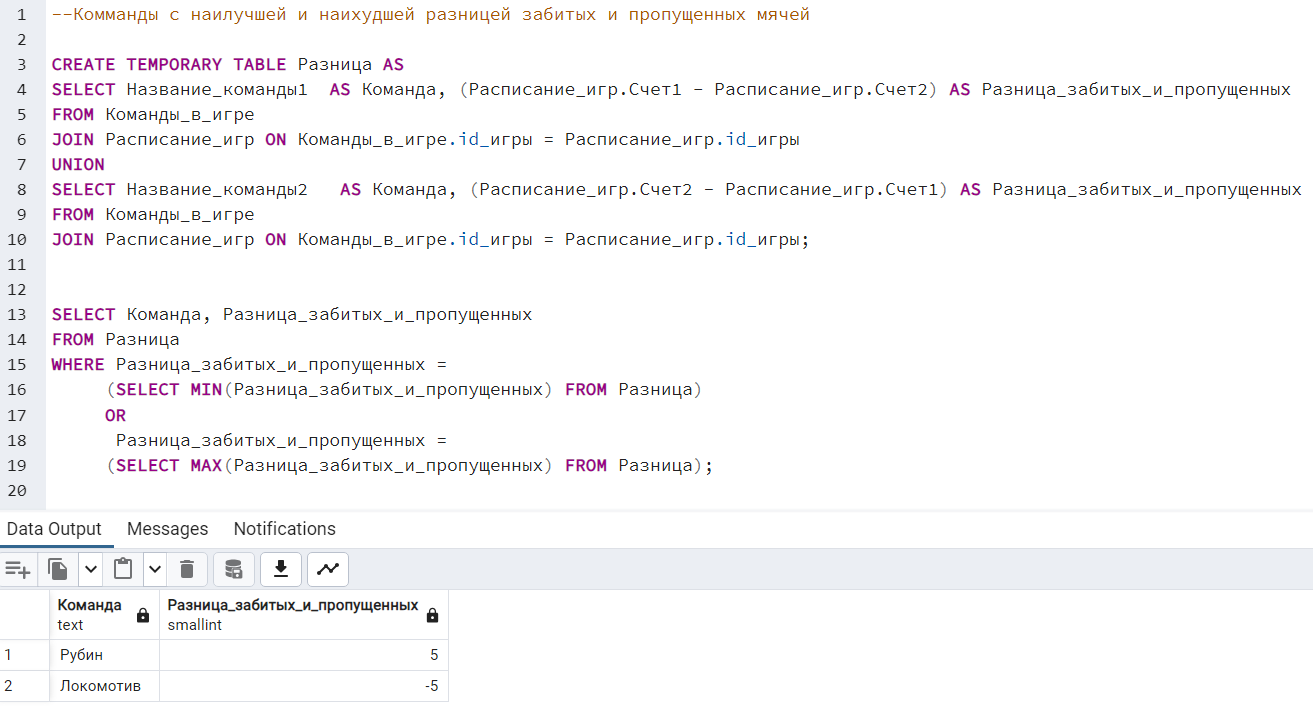


Рисунок 12. Команды, имеющие наилучшую и наихудшую разницу забитых и пропущенных мячей.

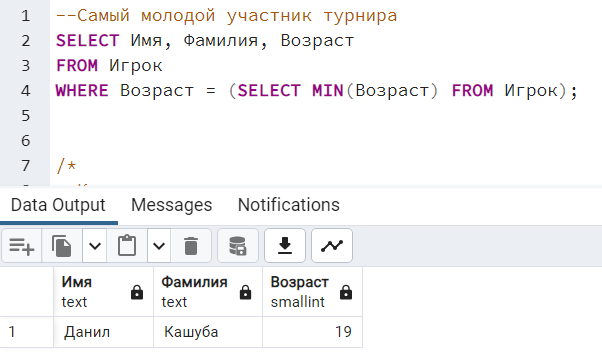


Рисунок 13. Самый молодой участник турнира.

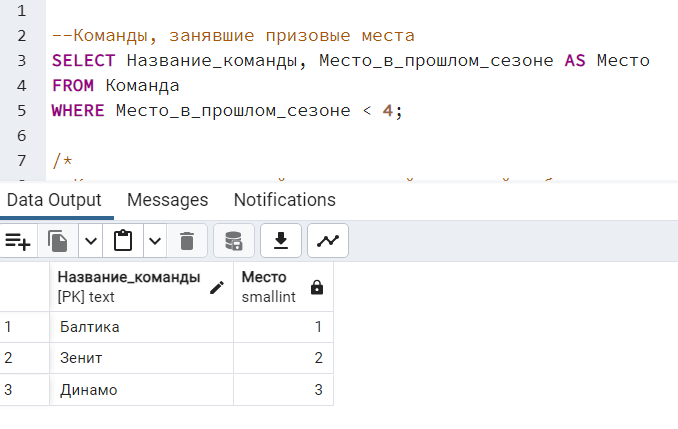


Рисунок 14. Команды, занявшие призовые места.

**Вывод.**

Был развернут локально PostgreSQL, написаны запросы для создания и заполнения таблиц в соответствии со структурой БД, написаны запросы к БД, отвечающие на вопросы в задания.

**Приложение А**

**Ссылки**

DB Fiddle: <https://www.db-fiddle.com/f/8P9RYRw1Db3HcXQywc6TsU/0>

**Приложение Б**

**Исходный код**

Файл lab2.sql:

CREATE TABLE Команда(

Название\_команды TEXT PRIMARY KEY,

Город\_команды TEXT,

Тренер TEXT,

Место\_в\_прошлом\_сезоне SMALLSERIAL

);

CREATE TABLE Игрок(

id\_игрока SERIAL PRIMARY KEY,

Название\_команды TEXT,

Фамилия TEXT,

Имя TEXT,

Возраст SMALLSERIAL,

Амплуа TEXT,

Номер SMALLSERIAL,

Забитые\_голы SMALLSERIAL,

FOREIGN KEY (Название\_команды) REFERENCES Команда(Название\_команды)

);

CREATE TABLE Стадион(

Название\_стадиона TEXT PRIMARY KEY,

Город\_стадиона TEXT,

Вместимость SERIAL

);

CREATE TABLE Организатор(

Название\_организации TEXT PRIMARY KEY

);

CREATE TABLE Расписание\_игр(

id\_игры SERIAL PRIMARY KEY,

Дата date,

Счет1 SMALLSERIAL,

Счет2 SMALLSERIAL,

Цена\_на\_билет SERIAL,

Название\_стадиона TEXT,

Название\_организации TEXT,

FOREIGN KEY (Название\_стадиона) REFERENCES Стадион(Название\_стадиона),

FOREIGN KEY (Название\_организации) REFERENCES Организатор(Название\_организации)

);

CREATE TABLE Команды\_в\_игре(

Название\_команды1 TEXT,

Название\_команды2 TEXT,

id\_игры SERIAL PRIMARY KEY,

FOREIGN KEY (Название\_команды1) REFERENCES Команда(Название\_команды),

FOREIGN KEY (Название\_команды2) REFERENCES Команда(Название\_команды),

FOREIGN KEY (id\_игры) REFERENCES Расписание\_игр(id\_игры)

);

INSERT INTO Команда(Название\_команды, Город\_команды, Тренер, Место\_в\_прошлом\_сезоне ) VALUES

('Балтика', 'Калининград', 'Королев А.Л.', 1),

('Зенит', 'Санкт-Петербург', 'Гирман А.В.', 2),

('Динамо', 'Москва', 'Смирнов Д.Ю.', 3),

('Локомотив', 'Москва', 'Королева П.А.', 4),

('Рубин', 'Казань', 'Иевлев Е.А.', 5);

INSERT INTO Игрок(Название\_команды, Фамилия, Имя, Возраст, Амплуа, Номер, Забитые\_голы) VALUES

('Балтика', 'Лазарев', 'Владислав', 21, 'Полузащитник', 89, 1),

('Балтика', 'Шишкин', 'Федор', 20, 'Нападающий', 52, 7),

('Зенит', 'Мостовой', 'Андрей', 25, 'Полузащитник', 17, 0),

('Зенит', 'Самохин', 'Кирилл', 22, 'Нападающий', 8, 12),

('Динамо', 'Смолов', 'Фёдор', 33, 'Нападающий', 10, 5),

('Динамо', 'Дудко', 'Максим', 33, 'Вратарь', 2, 28),

('Локомотив', 'Дзюба', 'Артём', 35, 'Нападающий', 7, 3),

('Локомотив', 'Баринов', 'Дмитрий', 27, 'Полузащитник', 6, 0),

('Рубин', 'Рыбус', 'Мацей', 30, 'Защитник', 31, 0),

('Рубин', 'Кашуба', 'Данил', 19, 'Нападающий', 37, 5);

INSERT INTO Организатор(Название\_организации) VALUES

('Fonbet'),

('Мир Российская Премьер Лига'),

('ЧЕ-2024'),

('Лига Чемпионов'),

('OLIMPBET');

INSERT INTO Стадион(Название\_стадиона, Город\_стадиона, Вместимость) VALUES

('Лужники', 'Москва', 76880),

('Газпром Арена', 'Санкт-Петербург', 64448),

('Динамо', 'Воронеж', 10880),

('Сибсельмаш', 'Новосибирск', 13000),

('Барс', 'Казань', 45379);

INSERT INTO Расписание\_игр(id\_игры, Дата, Счет1, Счет2, Цена\_на\_билет, Название\_стадиона, Название\_организации) VALUES

(1, '2023-12-13', 3, 1, 7000, 'Лужники', 'Fonbet'),

(2, '2023-12-17', 1, 0, 7000, 'Лужники', 'Fonbet'),

(3, '2024-02-05', 2, 7, 3000, 'Газпром Арена', 'ЧЕ-2024'),

(4, '2024-03-01', 0, 1, 3000, 'Газпром Арена', 'ЧЕ-2024'),

(5, '2024-04-10', 6, 9, 1000, 'Динамо', 'OLIMPBET');

INSERT INTO Команды\_в\_игре(id\_игры, Название\_команды1, Название\_команды2) VALUES

(1, 'Балтика', 'Зенит'),

(2, 'Зенит', 'Локомотив'),

(3, 'Локомотив', 'Рубин'),

(4, 'Рубин', 'Динамо'),

(5, 'Динамо', 'Балтика');

\*/

--Даты встреч команды, ее противники и счет

--Команда Локомотив

SELECT Дата, Команды\_в\_игре.Название\_команды1 as Команда1, Команды\_в\_игре.Название\_команды2 as Команда2, Счет1, Счет2

FROM Расписание\_игр

JOIN Команды\_в\_игре ON Команды\_в\_игре.id\_игры = Расписание\_игр.id\_игры

WHERE Команды\_в\_игре.Название\_команды1 = 'Локомотив' OR Команды\_в\_игре.Название\_команды2 = 'Локомотив';

--Номера и Фамилии игроков команд, участвовахших во встрече, которая проходила в указанный день в указанном городе

--День 2024-02-05, город Санкт-Петербург

SELECT Номер, Фамилия

FROM Игрок

JOIN Команды\_в\_игре ON Команды\_в\_игре.Название\_команды1 = Игрок.Название\_команды OR Команды\_в\_игре.Название\_команды2 = Игрок.Название\_команды

JOIN Расписание\_игр ON Команды\_в\_игре.id\_игры = Расписание\_игр.id\_игры

JOIN Стадион ON Расписание\_игр.Название\_стадиона = Стадион.Название\_стадиона

WHERE Стадион.Город\_стадиона = 'Санкт-Петербург' AND Расписание\_игр.Дата = '2024-02-05';

--Цена билета на матч между указанными командами

--Команды 'Балтика' и 'Динамо'

SELECT Цена\_на\_билет

FROM Расписание\_игр

JOIN Команды\_в\_игре ON Команды\_в\_игре.id\_игры = Расписание\_игр.id\_игры

WHERE Команды\_в\_игре.Название\_команды1 = 'Балтика' AND Команды\_в\_игре.Название\_команды2 = 'Динамо'

OR Команды\_в\_игре.Название\_команды1 = 'Динамо' AND Команды\_в\_игре.Название\_команды2 = 'Балтика';

--Игрок, забивший в турнире наибольшее количество мячей

SELECT Имя, Фамилия, Забитые\_голы

FROM Игрок

WHERE Забитые\_голы = (SELECT MAX(Забитые\_голы) FROM Игрок);

--Самый молодой участник турнира

SELECT Имя, Фамилия, Возраст

FROM Игрок

WHERE Возраст = (SELECT MIN(Возраст) FROM Игрок);

--Команды, занявшие призовые места

SELECT Название\_команды, Место\_в\_прошлом\_сезоне AS Место

FROM Команда

WHERE Место\_в\_прошлом\_сезоне < 4;

--Комманды с наилучшей и наихудшей разницей забитых и пропущенных мячей

CREATE TEMPORARY TABLE Разница AS

SELECT Название\_команды1 AS Команда, (Расписание\_игр.Счет1 - Расписание\_игр.Счет2) AS Разница\_забитых\_и\_пропущенных

FROM Команды\_в\_игре

JOIN Расписание\_игр ON Команды\_в\_игре.id\_игры = Расписание\_игр.id\_игры

UNION

SELECT Название\_команды2 AS Команда, (Расписание\_игр.Счет2 - Расписание\_игр.Счет1) AS Разница\_забитых\_и\_пропущенных

FROM Команды\_в\_игре

JOIN Расписание\_игр ON Команды\_в\_игре.id\_игры = Расписание\_игр.id\_игры;

SELECT Команда, Разница\_забитых\_и\_пропущенных

FROM Разница

WHERE Разница\_забитых\_и\_пропущенных =

(SELECT MIN(Разница\_забитых\_и\_пропущенных) FROM Разница)

OR

Разница\_забитых\_и\_пропущенных =

(SELECT MAX(Разница\_забитых\_и\_пропущенных) FROM Разница);