**Московский авиационный институт**

**(Национальный исследовательский университет)**

Факультет прикладной математики и информационных технологий Кафедра вычислительной математики и программирования

**Лабораторная работа**

По курсу «Базы данных»

Тема: Статистика турниров по киберспорту в дисциплине

Counter Strike

Студенты: Андреев А. О.,

Попов И. П.,

Семин А. В.

Группа: М8О-306Б-20

Преподаватель: Сеченых П.А.

Оценка:

Москва, 2022

Оглавление

[Постановка задачи 3](#_Toc120561766)

[База данных 4](#_Toc120561767)

[**Таблица Tournaments (турниры)** 4](#_Toc120561768)

[**Таблица Teams (команды):** 4](#_Toc120561769)

[**Таблица Tournaments\_Teams:** 4](#_Toc120561770)

[**Таблица Players (игроки):** 5](#_Toc120561771)

[Разработка модели в ERWIN 6](#_Toc120561772)

[Логическая модель: 6](#_Toc120561773)

[Физическая модель: 7](#_Toc120561774)

[Сложные запросы 8](#_Toc120561776)

# **Постановка задачи**

Разработать и реализовать базу данных по индивидуальному заданию. Разрабатываемая база данных должна содержать не менее трех таблиц, две из которых находятся в отношении «многие-ко-многим». Таблицы (хотя бы две) должны содержать не менее пяти полей.

Необходимо описать включенные в БД таблицы. При описании каждой таблицы необходимо указывать для каждого поля тип данных, а также другие характеристики (длину поля, маску ввода, значение по умолчанию и т.п.), для некоторых полей выполнить проверку правильности ввода исходных данных. Определить первичные ключи и ключи связи.

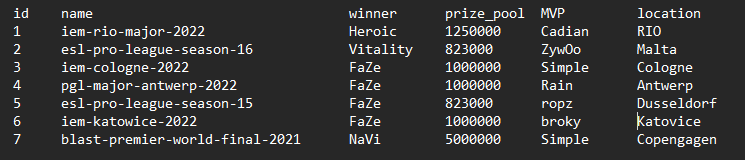
Для разработанной БД дать логическую схему, физическую схему и описание сгенерированной БД на DDL SQL для PostgreSQL (или СУБД на выбор).

Разработать запросы:

* использующие реляционные и булевы операторы в предикатах;
* с использованием операторов IN, BETWEEN, LIKE в условиях;
* с использованием групповых (агрегирующих) функций;
* на вычислимое поле с форматированием результата;
* с использованием нескольких таблиц;
* на соединение таблицы самой с собой;
* с использованием вложенных запросов;
* на связанные подзапросы;
* с использованием операторов EXIST, ANY, ALL, SOME;
* с использованием операторов UNION, INTERSECT, EXCEPT;
* с использованием оператора GROUP BY;
* с использованием оператора ORDER BY;
* с командами обновления.

# **База данных**

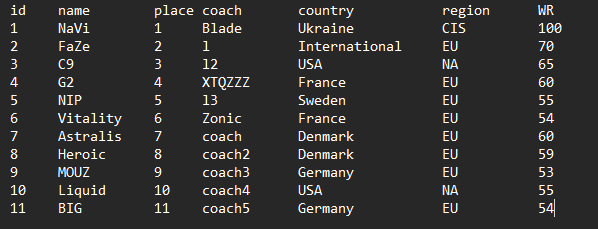
**Таблица Tournaments (турниры):**



Типы данных:

* Первичный ключ (serial)
* Название турнира (char)
* Победитель (char)
* Призовой фонд (integer)
* MVP (лучший игрок турнира по статистике) (char)
* Место проведения (char)

## **Таблица Teams (команды):**



Типы данных:

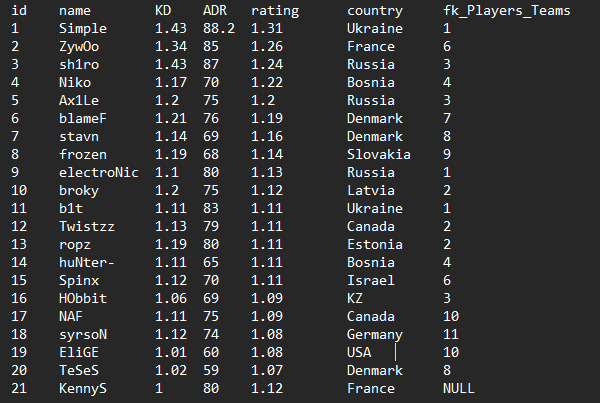
* Первичный ключ (serial)
* Название команды (char)
* Место в мировом рейтинге (integer)
* Тренер команды (char)
* Страна (char)
* Регион (char)
* Win rate (процент побед) (double)

## **Таблица Tournaments\_Teams:**

Типы данных:

* Первичный ключ
* Внешний ключ команды
* Внешний ключ турнира

## **Таблица Players (игроки):**

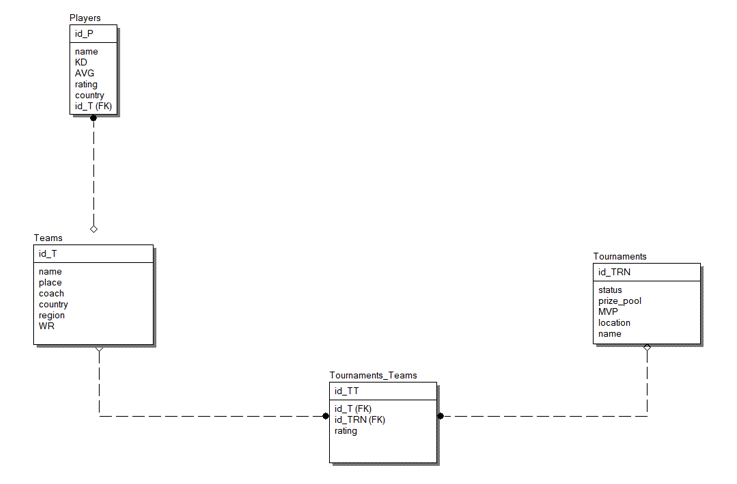
****

Типы данных:

* Первичный ключ (serial)
* Nickname игрока (char)
* KD (double)
* ADR (double)
* Рейтинг (double)
* Страна (char)
* Внешний ключ команды (integer)

# **Разработка модели в ERWIN**

## Логическая модель:



## Физическая модель:

## 

# **Сложные запросы**

**1. Все команды, участвовавшие в iem-rio-major-2022**

SELECT Teams.name AS "Название команды", Tournaments.name AS "Название турнира"

FROM Tournaments

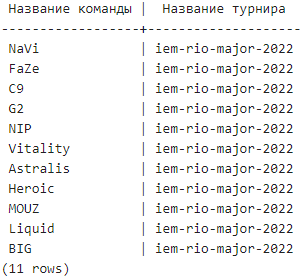
INNER JOIN Tournaments\_Teams

ON Tournaments\_Teams.tournament\_id = Tournaments.id

INNER JOIN Teams

ON Tournaments\_Teams.team\_id = Teams.id

WHERE Tournaments.name = 'iem-rio-major-2022';



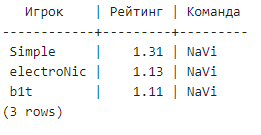
**2. Все игроки, выступающие за команду 'NaVi'**

SELECT Players.name AS "Игрок", Players.rating AS "Рейтинг", Teams.name AS "Команда"

FROM Players, Teams

WHERE Teams.id = Players.fk\_Players\_Teams AND

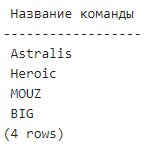
Teams.name = 'NaVi';



**3. Команды из Дании и Германии**

SELECT Teams.name AS "Название команды" FROM Teams

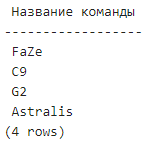
WHERE country in ('Denmark', 'Germany');



**4. Команды, имеющие процент WR от 60 до 80**

SELECT Teams.name AS "Название команды" FROM Teams

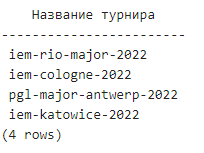
WHERE WR BETWEEN 60 AND 80;



**5. Турниры, в названии которых есть указание 2022 года** проведения

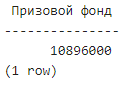
SELECT Tournaments.name AS "Название турнира" FROM Tournaments

WHERE name LIKE '%-2022';



**6. Общий призовой фонд всех указанных турниров**

SELECT SUM(prize\_pool) AS "Призовой фонд" FROM Tournaments;



**7. Сортировка списка игроков по стране, за которую он выступают**

SELECT name AS "Игрок", country AS "Страна"

FROM Players

ORDER BY country;



**8. Подсчет количества игроков из каждой страны**

SELECT country AS "Страна", COUNT(\*) AS "Количество"

FROM Players

GROUP BY country

ORDER BY COUNT(\*) DESC;



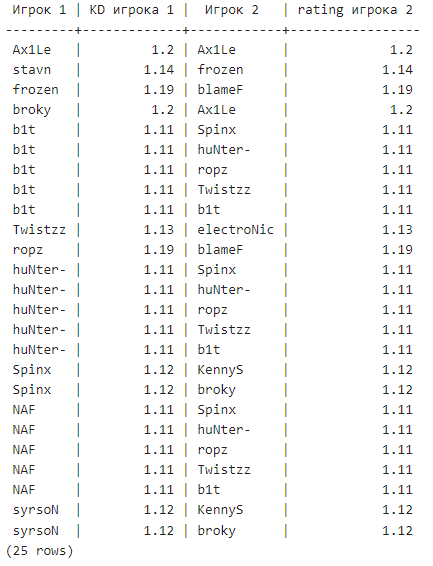
**9. Найти игроков, у которых KD совпадает с рейтингом других игроков**

SELECT a.name AS "Игрок 1", a.KD AS "KD игрока 1",

b.name AS "Игрок 2", b.rating AS "rating игрока 2"

FROM Players AS a INNER JOIN Players AS b

ON a.KD = b.rating;



**10. Все игроки, выступающие за команду 'NaVi'**

SELECT Players.name AS "Игрок", Players.rating AS "Рейтинг"

FROM Players

WHERE Players.name = ANY(

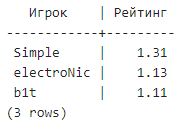
SELECT Players.name

FROM Players, Teams

WHERE Teams.id = Players.fk\_Players\_Teams AND

Teams.name = 'NaVi'

);



**11. Турниры, в которых участвует > 10 команд**

SELECT \* FROM Tournaments AS a

WHERE 10 < (

SELECT COUNT(\*)

FROM Tournaments AS b

INNER JOIN Tournaments\_Teams

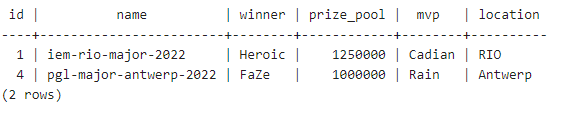
ON Tournaments\_Teams.tournament\_id = b.id

INNER JOIN Teams

ON Tournaments\_Teams.team\_id = Teams.id

WHERE b.name = a.name

);



**12. Команды – победители каждого турнира**

SELECT Tournaments.name AS "Название турнира", Tournaments.winner AS "Победитель"

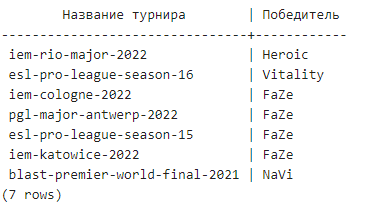
FROM Tournaments

WHERE EXISTS

(SELECT \*

FROM Teams

WHERE Tournaments.winner = Teams.name);



**13. Сортируем игроков Vitality: сначала те, у кого страна совпадает со страной команды, затем те, у кого не совпадает**

SELECT Players.name AS "Игрок", Players.country AS "Страна игрока", Teams.name AS "Команда", Teams.country AS "Страна команды"

FROM Players CROSS JOIN Teams

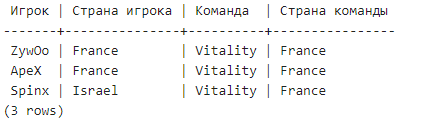
WHERE Players.fk\_Players\_Teams = Teams.id AND Teams.name = 'Vitality' AND Players.country = Teams.country

UNION ALL

SELECT Players.name AS "Игрок", Players.country AS "Страна игрока", Teams.name AS "Команда", Teams.country AS "Страна команды"

FROM Players CROSS JOIN Teams

WHERE Players.fk\_Players\_Teams = Teams.id AND Teams.name = 'Vitality' AND Players.country != Teams.country;



**14. Все турниры, где призовой фонд был 1000000 и победили FaZe**

SELECT Tournaments.name AS "Название турнира", Tournaments.winner AS "Победитель", Tournaments.prize\_pool AS "Призовой фонд"

FROM Tournaments

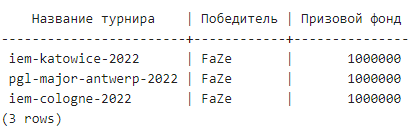
WHERE Tournaments.winner = 'FaZe'

INTERSECT

SELECT Tournaments.name AS "Название турнира", Tournaments.winner AS "Победитель", Tournaments.prize\_pool AS "Призовой фонд"

FROM Tournaments

WHERE Tournaments.prize\_pool = 1000000;



**15. Все турниры, где призовой фонд был не равен 1000000 и победили FaZe**

SELECT Tournaments.name AS "Название турнира", Tournaments.winner AS "Победитель", Tournaments.prize\_pool AS "Призовой фонд"

FROM Tournaments

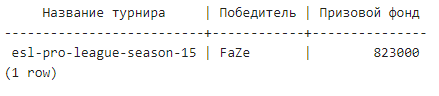
WHERE Tournaments.winner = 'FaZe'

EXCEPT

SELECT Tournaments.name AS "НАзвание турнира", Tournaments.winner AS "Победитель", Tournaments.prize\_pool AS "Призовой фонд"

FROM Tournaments

WHERE Tournaments.prize\_pool = 1000000;



**16. Все игроки, выступающие за команду 'NaVi'**

SELECT Players.name AS "Игрок", Players.rating AS "Рейтинг"

FROM Players

WHERE Players.name = SOME(

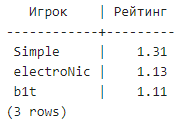
SELECT Players.name

FROM Players, Teams

WHERE Teams.id = Players.fk\_Players\_Teams AND

Teams.name = 'NaVi'

);



**17. Переводим всех игроков команды NaVi в BIG**

UPDATE Players SET

fk\_Players\_Teams = 11 WHERE fk\_Players\_Teams = 1;

SELECT Players.name AS "Игрок", Teams.name AS "Команда"

FROM Players, Teams

WHERE Teams.id = Players.fk\_Players\_Teams AND

Teams.name = 'BIG';

