

**Санкт-Петербургский государственный
университет**

Факультет прикладной математики - процессов управления

Пелешок Иван Александрович

Курсовая работа

**Профессиональная лига
(Professional League)**

Направление 010302

Прикладная математика и информатика

Преподаватель

Филиппов Р.О

Санкт-Петербург

2017

Содержание

Глава 1. Схема и описание базы данных.....	3
Глава 2. Лёгкие запросы.....	7
Глава 3. Средние запросы.....	9
Глава 4. Сложные запросы.....	12

Все файлы располагаются на гитхабе по адресу:

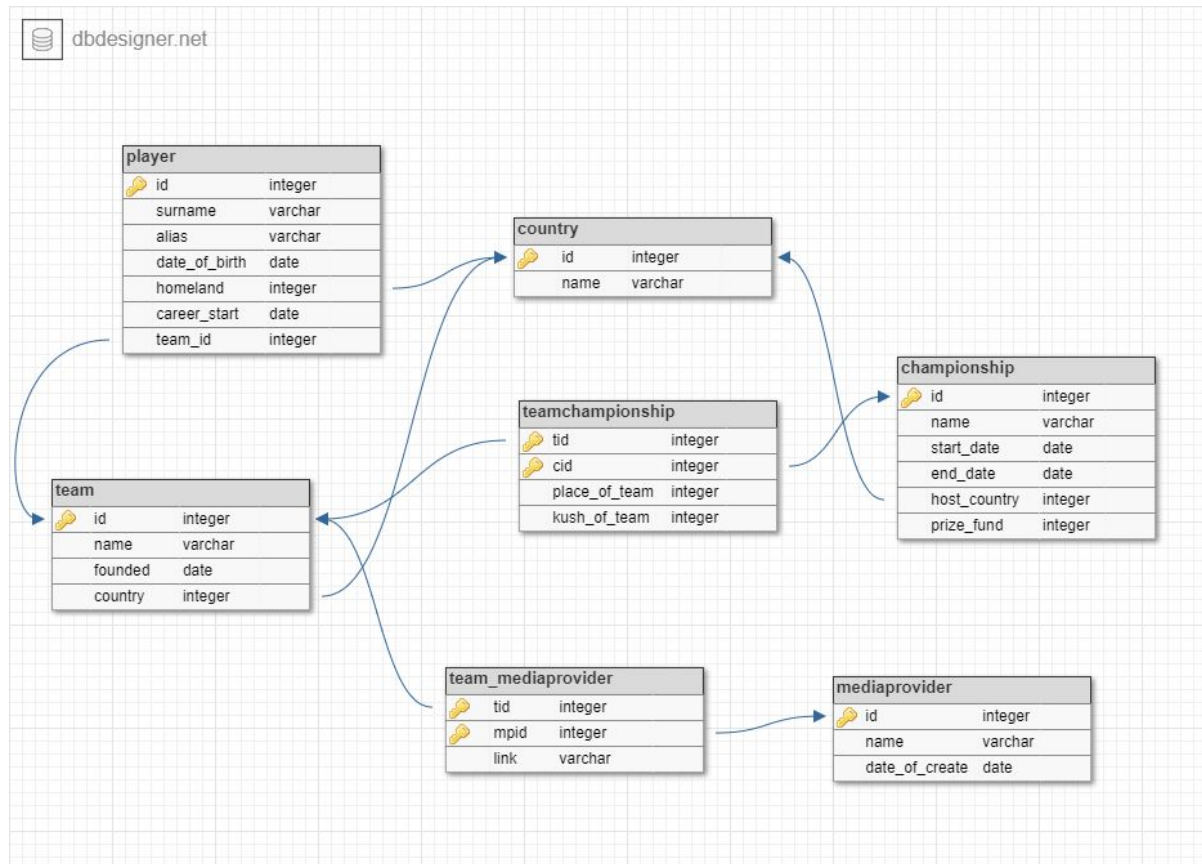
github.com/hEAh4rd/Database

Глава 1. Схема и описание базы данных.

База данных основана на профессиональной лиге по игре Counter-Strike: Global Offensive.

В данной базе данных будут рассмотрены наиболее важные, на мой взгляд, вещи затрагивающие профессиональную лигу по игре Counter-Strike: Global Offensive.

Ниже приведена схема, отражающая таблицы, поля, а также вся имеющиеся между ними связи.



Теперь коротко о каждой из таблиц базы данных:

1. player: Содержит всю информацию о игроке.

Поля:

id: Порядковый номер игрока, никак не определяет его место и приоритетность.

1. **PRIMARY KEY**
2. type **serial**
3. modifiers **NOT NULL**

surname: Фамилия игрока в реальной жизни.

1. type **varchar(50)**.
2. modifiers **NOT NULL**

alias: Игровое имя игрока.

1. type **varchar(30)**
2. modifiers **NOT NULL**

date_of_birth: Дата рождения игрока.

1. type **date**
2. modifiers **NOT NULL**

homeland (foreign key): Ссылка на страну, где он родился, в таблицу country.

1. type **integer**
2. modifiers **NOT NULL**

career_start: Дата начала карьеры.

1. type **date**
2. modifiers **NOT NULL**

team_id (foreign key): Ссылка на команду, за которую он играет, в таблицу team.

1. type **integer**
2. modifiers **NOT NULL**

2. country: Содержит информацию о стране.

Поля:

id: Порядковый номер страны, никак не определяет его место и приоритетность.

1. type **serial**
2. modifiers **NOT NULL**
3. **PRIMARY KEY**

name: Название страны

1. type: **varchar(30)**
2. modifiers **NOT NULL**

3. team: Содержит информацию о команде. (Таблица player к team имеет отношение **1:m**, т.к игрок может состоять только в одной команде, в команде должно быть 5 игроков).

Поля:

id: Порядковый номер команды, никак не определяет его место и приоритетность.

1. type **serial**
2. modifiers **NOT NULL**
3. **PRIMARY KEY**

name: Название команды.

1. type **varchar(30)**
2. modifiers **NOT NULL**

founded: Дата создания данной команды.

1. type **date**
2. modifiers **NOT NULL**

country (foreign key): Ссылка на страну, в которой была создана команды, в таблицу country.

1. type **integer**
2. modifiers **NOT NULL**

4. mediaprovider: Содержит информацию о СМИ.

Поля:

id: Порядковый номер СМИ, никак не определяет его место и приоритетность.

1. **PRIMARY KEY**
2. type **serial**
3. modifiers **NOT NULL**

name: Название данного СМИ.

1. type **varchar(30)**
2. modifiers **NOT NULL**

date_of_create: Дата создания данного СМИ.

1. type **date**
2. modifiers **NOT NULL**

5. team_mediaprovider: Содержит промежуточную информацию о команде и сми, тем самым придавая таблицам team и mediaprovider отношение **m:m**, т.к команда может пользоваться разными СМИ, а в свою очередь, одним СМИ может пользоваться несколько команд.

Поля:

tid (foreign key): Ссылка на команду, которая использует данный линк для оповещения своих фанатов, в таблицу team;

1. type **integer**
2. modifiers **NOT NULL**

cid (foreign key): Ссылка на СМИ, линк которой является провайдером, в таблицу mediaprovider;

1. type **integer**
2. modifiers **NOT NULL**

PRIMARY KEY(tid,cid)

link: Ссылка команды на определённый хостинг, которую могут просматривать фанаты данной команды.

1. type **varchar(70)**
2. modifiers **NOT NULL**

6. championship:

id: Порядковый номер чемпионата, никак не определяет его место и приоритетность.

name: Название чемпионата.

1. **PRIMARY KEY**
2. type **serial**
3. modifiers **NOT NULL**

start_date: Дата начала чемпионата.

1. type **date**
2. modifiers **NOT NULL**

end_date: Дата конца чемпионата.

1. type **date**
2. modifiers **NOT NULL**

host_country (foreign key): Ссылка на страну, где проводится чемпионат, в таблицу country.

1. type **integer**
2. modifiers **NOT NULL**

prize_found: Призовой фонд чемпионата.

1. type **integer**
2. modifiers **NOT NULL**

7. teamchampionship: содержит промежуточную информацию о команде и чемпионате, тем самым придавая таблицам отношение team и championship отношение **m:m**, т.к команда может принимать участие в разных чемпионатах, а в свою очередь несколько команд может принимать участие в одном чемпионате.

tid (foreign key): Ссылка на команду, которая принимала участие на чемпионата.

1. type **integer**
2. modifiers **NOT NULL**

cid (foreign key): Ссылка на чемпионат, на котором команды принимали участие.

1. type **integer**
2. modifiers **NOT NULL**

PRIMARY KEY(tid,cid)

place_of_team: Место команды на данном чемпионате.

1. type **integer**
2. modifiers **NOT NULL**

kush_of_team: Выигрыш команды.

1. type **integer**
2. modifiers **NOT NULL**

Глава 2. Лёгкие запросы.

1. Данная выборка демонстрирует команды, которые были созданы до 2010-01-01 в порядке возрастания года создания.

```
SELECT *FROM team
WHERE founded < '2010-01-01'
order by founded ASC;
```

Оптимизация:

```
CREATE INDEX team_founded_index ON team USING HASH (founded);
```

Добавлен индекс team_founded_index для фильтрации и сортировки по дате основания команды.

2. Данная выборка выводит количество игроков для каждой команды, которые родились до 1995-1-1, сортировка по id в порядке возрастания.

```
SELECT team_id, count(id) AS Number_of_players_from_the_team FROM player WHERE
date_of_birth<'1995-1-1' GROUP BY team_id ORDER BY team_id ASC;
```

Оптимизация:

```
CREATE INDEX ON player (date_of_birth);
```

Добавлен индекс player_date_of_birth_idx для фильтрации и сортировки по дате окончания чемпионата.

3. Данная выборка демонстрирует количество чемпионатов, где призовой фонд был более 100000.

```
SELECT COUNT(id) AS Number_of_championships FROM championship WHERE
prize_fund > 100000;
```

Оптимизация:

```
CREATE INDEX championship_prize_fund_index ON championship (prize_fund);
```

Добавлен индекс championship_prize_fund_index для фильтрации и сортировки по дате рождения игрока.

4. Данная выборка демонстрирует чемпионаты, призовые фонды не повторяются, а сортируются относительно даты, если есть два похожих призовых фонда по количеству денег, то они сравниваются: выбирается тот, который закончился позже, призовой фонд не может быть меньше 100000.

```
SELECT DISTINCT ON (prize_fund) name, end_date FROM championship  
where prize_fund > 100000  
ORDER BY prize_fund, end_date ASC;
```

Оптимизация:

```
CREATE INDEX Championship_prize_fund_index ON championship (prize_fund);
```

Добавлен индекс Championship_prize_fund_index для фильтрации и сортировки призового фонда чемпионата.

Глава 3. Средние запросы.

1. Данная выборка демонстрирует процесс вывода тех команд, которые играли в родных странах на различных чемпионатах, которые закончились позже 2016-01-01, сортируется по занятому месту на чемпионате в порядке возрастания (начиная с первого).

```
SELECT t.name, t.country, tc.place_of_team, tc.kush_of_team, c.name, c.end_date,  
c.host_country, c.prize_fund FROM team t  
JOIN teamchampionship  
tc ON t.id = tc.tid  
JOIN championship c  
ON t.country = c.host_country  
AND tc.cid = c.id  
WHERE c.end_date > '2016-01-01'  
ORDER BY tc.place_of_team ASC;
```

Оптимизация:

```
CREATE INDEX team_country_index ON team (country);  
CREATE INDEX championship_host_country_index ON championship  
(host_country)  
CREATE INDEX championship_end_date_index ON championship USING HASH  
(end_date);
```

Добавлены индексы team_country_index, championship_host_country_index,
championship_end_date_index

Используется team_pkey, teamchampionship_pkey, championship_pkey.

2. Данная выборка подсчитывает количество СМИ, которые есть у команды (т.е для каждой команды своё количество СМИ), причём СМИ должны быть созданы позднее 2000-1-1.

```
SELECT t.name, count(tm.tid) AS Number_of_media  
FROM mediaprovider m  
JOIN team_mediaprovider tm  
ON m.id=tm.mpid  
JOIN team t  
ON tm.tid = t.id  
WHERE m.date_of_create>'2000-1-1'  
GROUP BY t.id  
ORDER BY count(tm.tid) DESC;
```

Оптимизация:

```
CREATE INDEX ON mediaprovider (date_of_create);
```

Добавлены индексы mediaprovider_date_of_create_idx

Используются mediaprovider_pkey, team_mediaprovider_pkey, team_pkey.

3. Данная выборка подсчитывает сколько заработала каждая команда, заняв при этом от первого до третьего места.

```
SELECT t.name, SUM(tc.kush_of_team)
FROM team t
JOIN teamchampionship tc
ON t.id=tc.tid
WHERE tc.place_of_team<4
GROUP BY t.id
ORDER BY sum(tc.kush_of_team) ASC;
```

Оптимизация:

```
CREATE INDEX ON teamchampionship (place_of_team)
```

Добавлены индексы teamchampionship_place_of_team_idx.

Используются индексы team_pkey, team_pkey, teamchampionship_pkey.

Глава 4. Сложные запросы.

1. Выводит название команды и количество чемпионатов, на которых эта команда принимала участие.

```
SELECT DISTINCT(t.name) AS name_of_team,  
count_of_championships  
FROM  
(SELECT t.id, t.name, COUNT(t.id) AS count_of_championships FROM team t  
JOIN teamchampionship tc  
ON t.id = tc.tid  
JOIN championship c  
ON tc.cid = c.id  
GROUP BY t.id)  
AS t  
JOIN teamchampionship tc  
ON t.id=tc.tid  
JOIN championship c  
ON c.id=tc.cid  
ORDER BY count_of_championships DESC;
```

2. Данная выборка демонстрирует команды, которые за один чемпионат выиграли больше денег, чем минимальный призовой фонд среди всех чемпионатов.

```
SELECT DISTINCT ON (t.name) t.name, tc.kush_of_team from team t  
JOIN teamchampionship tc  
ON t.id = tc.tid  
JOIN championship c  
ON tc.cid = c.id  
WHERE tc.kush_of_team  
>  
(SELECT MIN(prize_fund) FROM championship  
ORDER BY t.name, tc.kush_of_team DESC);
```

3. Найти максимальное и минимальное количество, которое денег, которые заработали команды, а также их разницу между максимальным суммарным количеством денег, которая команда "А" выиграла, и минимальным суммарным количеством денег, которая выиграла команда "Б", где А и Б две разных команды.

```
SELECT
(SELECT max(sum)
FROM
(SELECT t.name, sum(tc.kush_of_team)
FROM team t JOIN teamchampionship tc ON t.id = tc.tid
GROUP BY t.id) AS MAX),
(SELECT min(sum)
FROM
(SELECT t.name, sum(tc.kush_of_team)
FROM team t JOIN teamchampionship tc ON t.id = tc.tid
GROUP BY t.id) AS MIN),
(SELECT max(sum) FROM
(SELECT t.name, sum(tc.kush_of_team)
FROM team t JOIN teamchampionship tc ON t.id = tc.tid
GROUP BY t.id) AS MAX)
-
(SELECT min(sum)
FROM (select t.name, sum(tc.kush_of_team)
FROM team t JOIN teamchampionship tc ON t.id = tc.tid
GROUP BY t.id) AS MIN)
AS dif_between_max_kush_of_team_and_min_kush_of_team;
```