

**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

**Малинин К.А**

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:

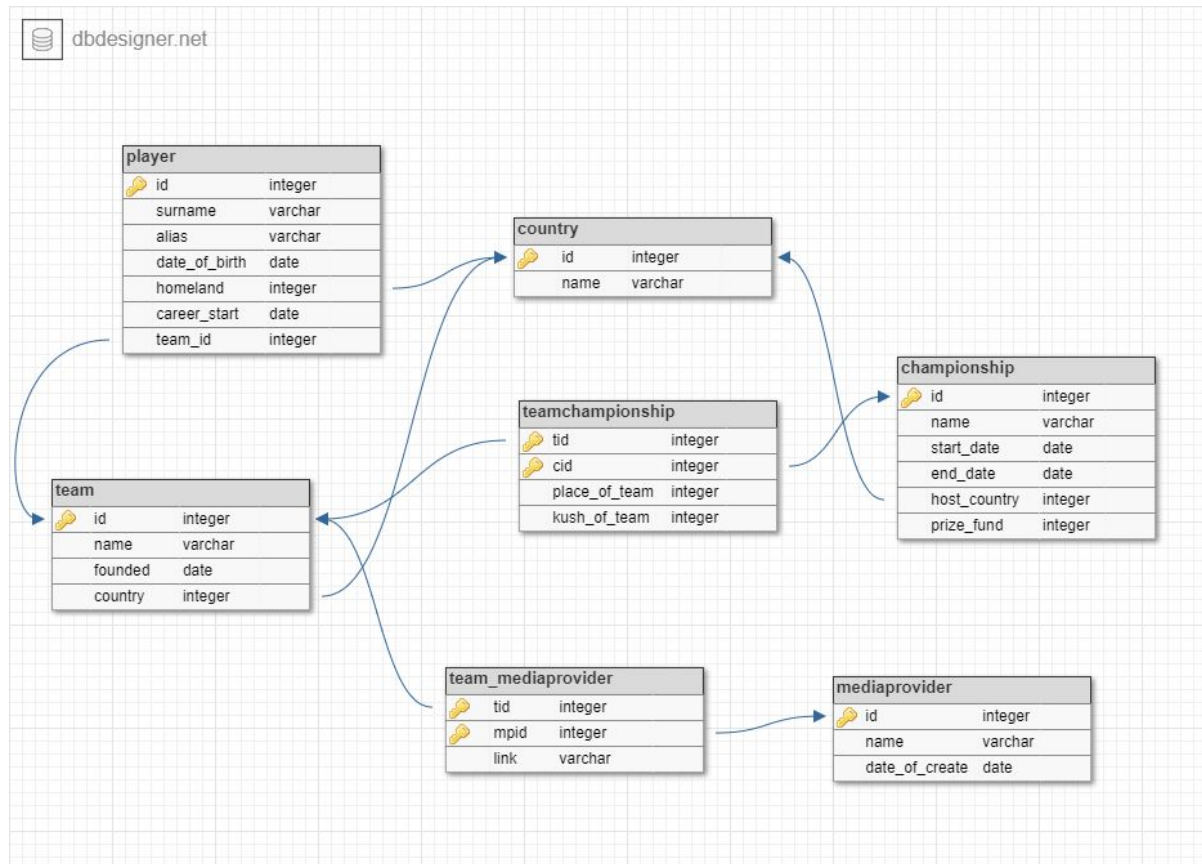
[github.com/hEAh4rd/Database](https://github.com/hEAh4rd/Database)

# 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);
```

2. Данная выборка демонстрирует чемпионаты, даты завершения чемпионата до 2016-05-14, где призовой фонд выбирается в порядке убывания.

```
SELECT id AS "Number", name, prize_fund FROM championship
where end_date < '2016-05-14'
order by prize_fund DESC;
```

**Оптимизация:**

```
CREATE INDEX championship_end_date_index ON championship USING HASH
(end_date);
```

3. Данная выборка демонстрирует игроков, дата рождения которых < 1995-01-01, сортируется в порядке убывания, выводятся первые 10 игроков.

```
SELECT *FROM player
WHERE date_of_birth < '1995-01-01'
ORDER BY career_start DESC
limit 10;
```

**Оптимизация:**

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

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);
```



## 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_sireal_id_index ON team (id);  
CREATE INDEX teamchampionship_tid_index ON teamchampionship (tid);  
CREATE INDEX team_country_index ON team (country);  
CREATE INDEX championship_host_country_index ON championship  
(host_country);  
CREATE INDEX teamchampionship_cid_index ON teamchampionship (cid);  
CREATE INDEX championship_sireal_id_index ON championship (id);  
CREATE INDEX championship_end_date_index ON championship USING HASH  
(end_date);
```

**2.** Данная выборка демонстрирует процесс вывода стран и сопоставления ей игроков из команды, которая была создана там, где родился игрок (то есть необязательно, что он состоит в этой команде: одна команда представляет одну страну), команды, которые участвуют в выборке должны быть созданы после 1990-01-01, формирование происходит в порядке возрастания id страны, выводятся первые 25 команд.

Если место создания команды не совпадает с местом рождения игрока, то поля surname и alias будут пусты, и просто выведена страна и соответствующая команда и дата её создания.

```
SELECT co.name, p.surname, p.alias, t.name, t.founded FROM country co
JOIN player p
ON co.id = p.hometown
JOIN team t
ON t.country = co.id
where t.founded > '1990-01-01'
ORDER BY t.country, t.founded ASC limit 25;
```

**Оптимизация:**

```
CREATE INDEX country_sireal_id_index country ON country (id);
CREATE INDEX player_hometown_index ON player (hometown);
CREATE INDEX team_country_index ON team (country);
CREATE INDEX team_founded_index ON team USING HASH (founded);
```

**3.** Данная выборка демонстрирует процесс вывода на экран СМИ и соответствующие ссылки от данной команды, а также видно откуда данная команда выкладывает свои новости: ролики, фотографии, презентации и прочее. Данное СМИ должно было создано до 2005-05-05, сортировка идёт по имени СМИ (m.name), выводятся первые 15 строк.

```
SELECT m.name, m.date_of_create, tm.link, t.name, t.founded, co.name FROM  
mediaprovider m  
JOIN team_mediaprovider tm  
ON m.id = tm.mpid  
JOIN team t  
ON tm.tid=t.id  
JOIN country co  
ON t.country = co.id  
WHERE m.date_of_create > '2005-05-05'  
ORDER BY t.name ASC  
limit 15;
```

**Оптимизация:**

```
CREATE INDEX mediaprovider_serial_id_index ON mediaprovider (id);  
CREATE INDEX team_mediaprovider_mpid_index ON team_mediaprovider (mpid);  
CREATE INDEX team_mediaprovider_tmidx_index ON team_mediaprovider (tmidx);  
CREATE INDEX team_mediaprovider_tid_index ON team_mediaprovider (tid);  
CREATE INDEX team_serial_id_index ON team (id);  
CREATE INDEX team_country_index ON team (country);  
CREATE INDEX media_provider_date_of_create_index ON mediaprovider USING  
HASH (date_of_create);
```

## 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 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)
AS dif_between_max_kush_of_team_and_min_kush_of_team;
```