

Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение  
высшего образования

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Факультет программной инженерии и компьютерной техники

**Дисциплина:** Базы данных

Отчёт по лабораторной работе № 4

**Выполнил:** До Зыонг Мань

Бу Минь Хиеу

**Студент группы:** Р33201

**Преподаватель:** Машина Екатерина Алексеевна

Санкт-Петербург

2022 г.

## I. Текст задания

Для выполнения лабораторной работы №4 необходимо:

- Реализовать разработанную в рамках лабораторной работы №3 даталогическую модель в реляционной СУБД PostgreSQL.
- Заполнить созданные таблицы данными.
- Обеспечить целостность данных при помощи средств языка DDL.
- В рамках лабораторной работы должны быть разработаны скрипты для создания/удаления требуемых объектов базы данных, заполнения/удаления содержимого созданных таблиц.

Отчёт по лабораторной работе должен содержать:

- титульный лист;
- текст задания;
- описание предметной области;
- DDL-скрипты, часть DML-скриптов;
- выводы по работе.

## II. Реализация даталогической модели на SQL

Типы перечислений

### 1. Position

```
CREATE TYPE POSITION AS ENUM ('GK', 'CB', 'LB', 'RB',  
                              'CDM', 'CAM', 'CM', 'CF', 'RF', 'LF', 'LM', 'RM');
```

### 2. Level\_league

```
CREATE TYPE LEVEL AS ENUM ('CLUB', 'INTERNATIONAL');
```

Стержневые сущности

### 1. Attribute

```
CREATE TABLE IF NOT EXISTS attribute (  
    id SERIAL PRIMARY KEY,
```

```
attack INTEGER DEFAULT 50,  
defense INTEGER DEFAULT 50,  
dribble INTEGER DEFAULT 50,  
pass INTEGER DEFAULT 50,  
speed INTEGER DEFAULT 50,  
power INTEGER DEFAULT 50,  
stamina INTEGER DEFAULT 50,  
);
```

## 2. Player

```
CREATE TABLE IF NOT EXISTS player (  
    id SERIAL PRIMARY KEY,  
    full_name VARCHAR(50) NOT NULL,  
    height INTEGER NOT NULL,  
    weight DOUBLE NOT NULL,  
    position POSITION NOT NULL,  
    nationality VARCHAR(50) NOT NULL,  
    shirt_number INTEGER NOT NULL,  
    birthday DATE NOT NULL,  
    image VARCHAR(100),  
    id_attribute INTEGER REFERENCES attribute NOT  
NULL  
);
```

## 3. Sponsor

```
CREATE TABLE IF NOT EXISTS sponsor (  
    id SERIAL PRIMARY KEY,  
    name VARCHAR(50) NOT NULL,  
    headquarter VARCHAR(50) NOT NULL,  
    management VARCHAR(50) NOT NULL,  
    assets BIGINT NOT NULL
```

```
|| );
```

#### 4. Goal

```
|| CREATE TABLE IF NOT EXISTS goal (  
||     id SERIAL PRIMARY KEY,  
||     time TIMESTAMP NOT NULL,  
||     video VARCHAR(100) NOT NULL,  
||     rating INTEGER  
|| );
```

#### 5. League

```
|| CREATE TABLE IF NOT EXISTS league (  
||     id SERIAL PRIMARY KEY,  
||     name VARCHAR(50) NOT NULL,  
||     region VARCHAR(50) NOT NULL,  
||     level LEVEL NOT NULL  
|| );
```

#### 6. Stadium

```
|| CREATE TABLE IF NOT EXISTS stadium (  
||     id SERIAL PRIMARY KEY,  
||     name VARCHAR(50) NOT NULL,  
||     coordinate VARCHAR(50) NOT NULL,  
||     capacity BIGINT NOT NULL,  
||     owner VARCHAR(50) NOT NULL  
|| );
```

#### 7. Coach

```
|| CREATE TABLE IF NOT EXISTS coach (  
||     id SERIAL PRIMARY KEY,  
||     full_name VARCHAR(50) NOT NULL,  
||     birthday DATE NOT NULL,  
||     nationality VARCHAR(50) NOT NULL
```

```
|| );
```

## 8. Referee

```
|| CREATE TABLE IF NOT EXISTS referee (  
||     id SERIAL PRIMARY KEY,  
||     full_name VARCHAR(50) NOT NULL,  
||     birthday DATE NOT NULL,  
||     nationality VARCHAR(50) NOT NULL,  
||     rating FLOAT4 NOT NULL  
|| );
```

## 9. Club

```
|| CREATE TABLE IF NOT EXISTS club (  
||     id SERIAL PRIMARY KEY,  
||     full_name VARCHAR(50) NOT NULL,  
||     nationality VARCHAR(50) NOT NULL,  
||     owner VARCHAR(50) NOT NULL,  
||     logo VARCHAR(100) NOT NULL,  
||     id_coach INTEGER REFERENCES coach NOT NULL,  
||     id_stadium INTEGER REFERENCES stadium NOT NULL  
|| );
```

## 10. Match

```
|| CREATE TABLE IF NOT EXISTS match (  
||     id SERIAL PRIMARY KEY,  
||     special_name VARCHAR(50) NOT NULL UNIQUE,  
||     time TIMESTAMP NOT NULL,  
||     weather VARCHAR(50) NOT NULL,  
||     audience INTEGER NOT NULL  
|| );
```

## Ассоциативные сущности

### 1. Player\_Club

```
CREATE TABLE IF NOT EXISTS player_club (  
    id_player INTEGER REFERENCES player NOT NULL,  
    id_team INTEGER REFERENCES club NOT NULL  
);
```

### 2. Club\_League

```
CREATE TABLE IF NOT EXISTS club_league (  
    club_id INTEGER REFERENCES club NOT NULL,  
    league_id INTEGER REFERENCES league NOT NULL  
);
```

### 3. Club\_Match

```
CREATE TABLE IF NOT EXISTS club_match (  
    club_id INTEGER REFERENCES club NOT NULL,  
    match_id INTEGER REFERENCES match NOT NULL  
);
```

### 4. Match\_League

```
CREATE TABLE IF NOT EXISTS match_league(  
    match_id INTEGER REFERENCES match NOT NULL,  
    league_id INTEGER REFERENCES league NOT NULL  
);
```

### 5. Match\_Goal

```
CREATE TABLE IF NOT EXISTS match_goal(  
    match_id INTEGER REFERENCES match NOT NULL,  
    goal_id INTEGER REFERENCES goal NOT NULL,  
);
```

### 6. Match\_Referee

```
CREATE TABLE IF NOT EXISTS match_referee (
```

```
match_id INTEGER REFERENCES match NOT NULL,  
referee_id INTEGER REFERENCES referee NOT NULL  
);
```

#### 7. Match\_Stadium

```
CREATE TABLE IF NOT EXISTS match_stadium (  
    match_id INTEGER REFERENCES match NOT NULL,  
    stadium_id INTEGER REFERENCES stadium NOT NULL  
);
```

#### 8. Player\_Goal

```
CREATE TABLE IF NOT EXISTS player_goal (  
    player_id INTEGER REFERENCES player NOT NULL,  
    goal_id INTEGER REFERENCES goal NOT NULL  
);
```

#### 9. Player\_MOTM

```
CREATE TABLE IF NOT EXISTS play_MOTM (  
    player_id INTEGER REFERENCES player NOT NULL,  
    match_id INTEGER REFERENCES match NOT NULL  
);
```

#### 10.Sponsor\_Club

```
CREATE TABLE IF NOT EXISTS sponsor_club (  
    sponsor_id INTEGER REFERENCES sponsor NOT NULL,  
    club_id INTEGER REFERENCES club NOT NULL  
);
```

### **III. Заполнение данными**

Я использовал генератор случайных запросов по следующей ссылке

### **IV. Вывод**

Мы научились вычленять данные из моего предметной области, составлять инфологическую и даталогическую таблицы.