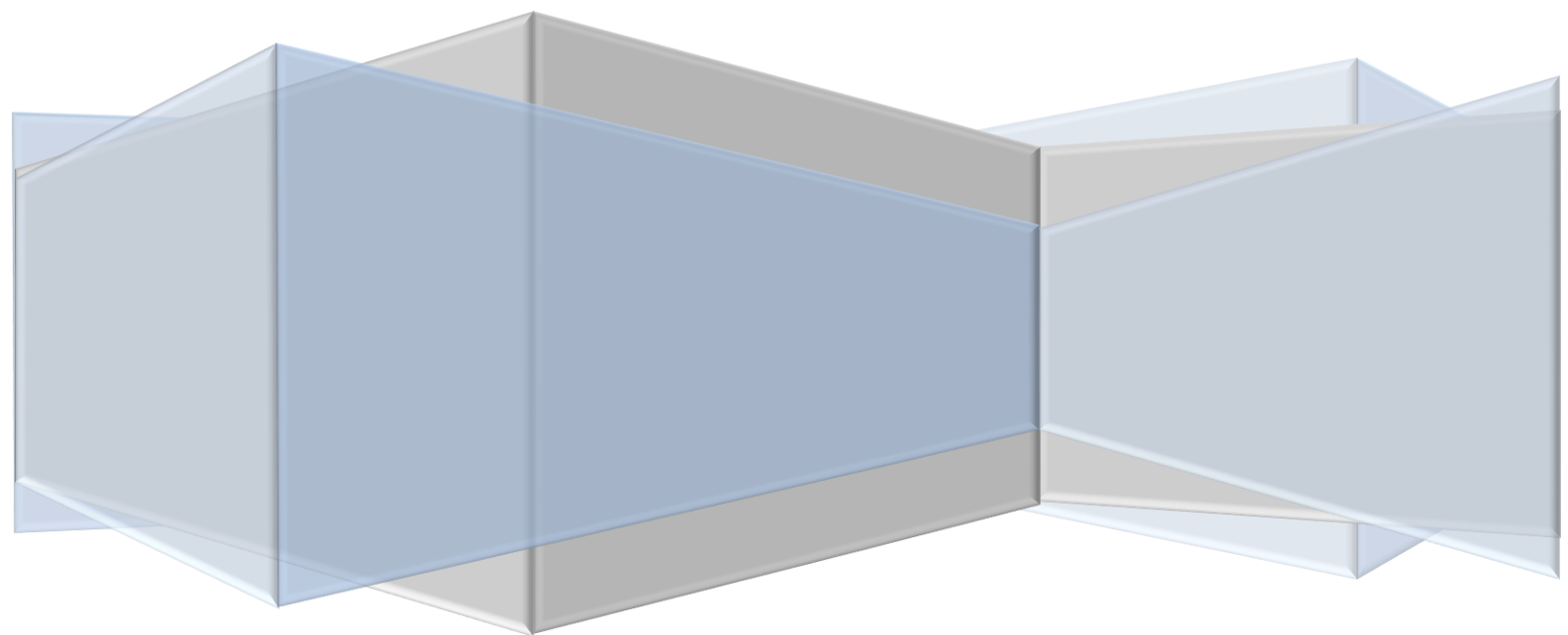


# База от данни за футболни отбори

Добрин Цветков, 81265



## СЪДЪРЖАНИЕ

1.	Описание на предметната област и предназначение на базата данни .....	2
2.	Е/R диаграма.....	3
3.	UML диаграма.....	4
4.	Релационен модел .....	4
5.	Нормализация .....	5
6.	Програмен код за създаване на таблици и добавяне на данни .....	6
7.	Примерни заявки – Условие, Код и резултат .....	10
8.	Предложение за потребителски интерфейс.....	13

## 1. ОПИСАНИЕ НА ПРЕДМЕТНАТА ОБЛАСТ И ПРЕДНАЗНАЧЕНИЕ НА БАЗАТА ДАННИ

Този документ проследява задачата за разработване на база от данни за футболни отбори и предложение за потребителски интерфейс. Тази база данни е предназначена за футболни агенции. Те имат нужда да поддържат актуална информация за футболистите и мениджърите, които представляват – лична информация за тези лица, информация в кой отбор играе всеки играч, начело на кой отбор е всеки мениджър, какви мачове са изиграли отборите (добрите представяния биха повлияли на цената на играчите и мениджърите). Базата данни няма за цел да постави цена на самите играчи/мениджъри! Тя само поддържа информация, на базата на която би могла да се направи оценка за цената.

Базата данни трябва да включва информация за отборите, играчите в тях, мениджърите на отборите и изиграните мачове между отборите, като:

- Всеки играч задължително участва в точно един отбор.
- Всеки отбор се състои от 0 или повече играчи
- Всеки отбор задължително има точно един мениджър (допустимо е при напускане на мениджърът временно като мениджър да бъде отбелязан специален запис 'Vacated')
- Всеки мениджър е начело на максимум един отбор
- Всеки отбор може да играе срещу всеки друг максимум 2 пъти – веднъж единият отбор е домакин, а следващият път същият отбор е гост

Информацията, която пазим за играчите е:

- Име
- Фамилия
- ЕГН (или Social Security Number)
- Номер на екип
- Отбор
- Възраст
- Заплата
- Адрес

Информацията, която пазим за отборите е:

- Име
- Брой трофеи
- Терени (във футбола отборите играят мачовете си на един терен, но понякога тренират на друг)
- ЕГН на мениджър

Информацията, която пазим за мениджърите е:

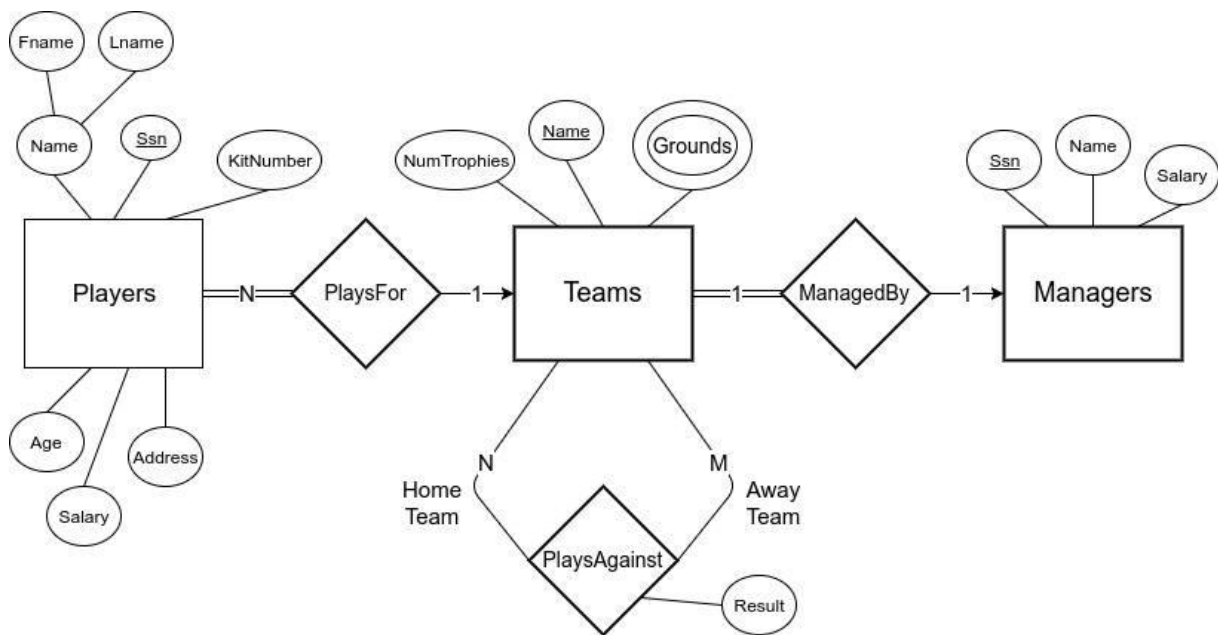
- ЕГН (или Social Security Number)

- Име
- Заплата

Информацията, която пазим за мачовете е:

- Име на домакин
- Име на гост
- Резултат

## 2. E/R ДИАГРАМА



Фигура 1. E/R диаграма на предметната област

Най-важното множество същности в нашата предметна област е Teams. То има атрибути Name – име на отбора, NumTrophies – брой трофеи (броим като трофей само титлата от първенството), и Grounds – както вече казахме е възможно един отбор да има няколко терена и затова Grounds е Multi-valued атрибут.

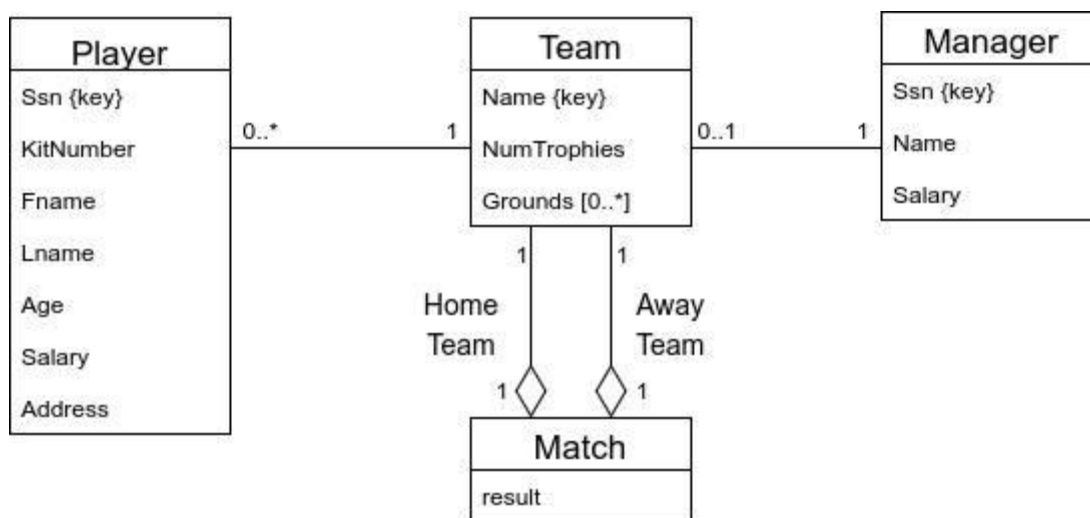
След това имаме множество от същности Players. Ключът на това множество е атрибутът SSN – ЕГН или Social Security Number на играча. Останалите атрибути са: съставен атрибут име – собствено и фамилия, номер на екип, възраст, заплата и адрес. Освен това имаме връзка PlaysFor, която е с множественост N:1, между Players и Teams, изразяваща участието на играч в отбор. Тъй като всеки играч задължително участва в точно един отбор, то при тази връзка имаме задължително участие от страна на Players.

От другата страна на Teams имаме множество същности Managers с информация за ЕГН (или social security number), име и заплата. Teams и Managers са свързани чрез връзката ManagedBy, която е 1:1 със задължително участие от страна на Teams – всеки отбор

задължително има точно един мениджър (евентуално специалният запис ,Vacated'), а всеки мениджър ръководи максимум един отбор.

Освен това имаме и връзка Matches. Тя е от Teams към Teams и е с множественост N:M – много отбори могат да играят с много други отбори, като освен това връзката има и атрибут result – резултатът след изиграване на мачът.

### 3. UML ДИАГРАМА



Фигура 2. UML диаграма на предметната област

Основната логика на UML диаграмата е същата като при E/R диаграмата. Съществената разлика е, че тук мачовете са представени като агрегиращ клас с двойна агрегираща връзка към класът Team – едната за отборът-домакин, а другата за отборът-гост, като освен това класът Match има и атрибут за резултат.

### 4. РЕЛАЦИОНЕН МОДЕЛ

Имайки готова E/R диаграма, лесно можем да преобразуваме моделът към релационен. Получените релации са:

#### 1) **Teams** (Name, NumTrophies, ManagerSsn)

Множеството същности Teams образува съответната релация Teams, като атрибутът Grounds не е от нея – тъй като той е Multi-valued, за него ще има отделна релация.

Понеже имаме връзка тип 1:1 между Teams и Managers със задължително участие от страна на Teams, вместо да правим нова релация за нея, можем да добавим атрибут ManagerSsn в релацията Teams – външен ключ,

референция към атрибутът Ssn на релацията Managers. По този начин ще избегнем аномалии в базата данни.

Ключ в релация Teams е атрибутът Name, както беше и в E/R моделът.

2) **TeamGrounds** (TeamName, GroundName)

Това е релацията за изразяване на Multi-valued атрибута Grounds на множеството същности Teams. Ключът е множеството от двата атрибута на релацията – TeamName и GroundName. TeamName е външен ключ, референция към атрибутът Name на релацията Teams.

3) **Players** (Ssn, KitNumber, TeamName, Fname, Lname, Age, Salary, Address)

Множеството същности Players образува релация Players, като ключ в тази релация е ключът от E/R диаграмата – атрибутът Ssn. Всички останали атрибути от E/R диаграмата също се пренасят и в релационната схема, като освен това имаме и допълнителен атрибут TeamName – външен ключ, референция към атрибутът Name на релацията Teams. Чрез него представяме връзката PlaysFor, която е с множественост N:1.

4) **Managers** (Ssn, Name, Salary)

Тук няма никакви уловки. Множеството Managers образува релация Managers, като атрибутите са същите и ключът е същият – атрибутът Ssn.

5) **Matches** (HomeTeamName, AwayTeamName, Result)

Тъй като връзката Matches е от тип много към много, за нея трябва да направим отделна релация. Ключът на такава релация е множеството от ключовите атрибути на множества същности, които тя свързва. В нашият случай това е два пъти множеството същности Teams – веднъж като отбор-домакин (HomeTeamName), и втори път като отбор гост (AwayTeamName). Освен това имаме и атрибут на връзката Result, който директно се пренася в релацията Matches. И HomeTeamName, и AwayTeamName са външни ключове – референции към атрибутът Name на релацията Teams.

## 5. НОРМАЛИЗАЦИЯ

a. **Teams** (Name, NumTrophies, ManagerSsn)

Ключът е {Name}. Нетривиалните функционални зависимости са:

Name -> NumTrophies, ManagerSsn

и произлизащите от нея чрез декомпозиция. Лявата страна изпълнява условието за нормална форма на Бойс-Код, тъй като множеството {Name} е самият ключ.

Заклучваме, че релацията Teams е в нормална форма на Бойс-Код.

Понеже всяка функционална зависимост е и многозначна зависимост, то имаме многозначни зависимости Name ->-> NumTrophies, ManagerSsn (но тя е тривиална, тъй като {Name, NumTrophies, ManagerSsn} е множеството от всички

атрибути), Name->NumTrophies и Name->ManagerSsn. Нямаме нетривиални многозначни зависимости, за които лявата част да не е суперключ. Следователно релацията е и в четвърта нормална форма.

b. TeamGrounds (TeamName, GroundName)

Тъй като всяка релация с два атрибута е в нормална форма на Бойс-Код, то и за TeamGrounds това е изпълнено. Нямаме нетривиални многозначни зависимости. Следователно релацията е и в четвърта нормална форма.

c. Players (Ssn, KitNumber, TeamName, Fname, Lname, Age, Salary, Address)

Ключът е атрибутът Ssn. Имаме нетривиална функционална зависимост:  
Ssn -> KitNumber, TeamName, Fname, Lname, Age, Salary, Address  
и всички, които се получават от нея чрез декомпозиция. Аналогично на точка а., заключаваме, че релацията Players също е в нормална форма на Бойс-Код.  
Аналогично на разсъжденията при Teams, релацията е и в четвърта нормална форма.

d. Managers (Ssn, Name, Salary)

Ключът е {Ssn}. Имаме нетривиална функционална зависимост:  
Ssn -> Name, Salary  
и всички, които се получават от нея чрез декомпозиция. Аналогично на точка а., заключаваме, че релацията Managers също е в нормална форма на Бойс-Код.  
Аналогично на разсъжденията при Teams, релацията е и в четвърта нормална форма.

e. Matches (HomeTeamName, AwayTeamName, Result)

Ключът е {HomeTeamName, AwayTeamName}. Имаме нетривиална функционална зависимост:  
HomeTeamName, AwayTeamName -> Result  
и всички, които се получават от нея чрез декомпозиция. Аналогично на точка а., заключаваме, че релацията Matches също е в нормална форма на Бойс-Код.

## 6. ПРОГРАМЕН КОД ЗА СЪЗДАВАНЕ НА ТАБЛИЦИ И ДОБАВЯНЕ НА ДАННИ

Първо трябва да си създадем база данни. За целта трябва да изпълним следният код:

```
create database football
go
use football
go
```

След това започваме с добавянето на отделните релации:

a. Managers (Ssn, Name, Salary)

Кодът за съставянето на тази релация е:

```
create table Managers(
    Ssn char(10),
    Name varchar(25) NOT NULL,
    Salary tinyint, --in millions
    PRIMARY KEY (Ssn)
);
```

Имаме първичен ключ Ssn и ограничение името да не е NULL.

Примерни записи в таблицата добавяме с кодът:

```
insert into Managers values('999999999', 'Vacated', 0)
insert into Managers values('1234567890', 'Alex Ferguson', 5)
insert into Managers values('1234567891', 'Jose Mourinho', 24)
insert into Managers values('1234567892', 'Mauricio Pochettino', 17)
insert into Managers values('1234567893', 'Antonio Conte', 19)
```

В резултат получаваме следната таблица:

Managers	Ssn	Name	Salary
	1234567890	Alex Ferguson	5
	1234567891	Jose Mourinho	24
	1234567892	Mauricio Pochettino	17
	1234567893	Antonio Conte	19
	999999999	Vacated	0

Фигура 3. Таблица Managers

b. Teams (Name, NumTrophies, ManagerSsn)

Кодът, за съставянето на тази релация е:

```
create table Teams(
    Name varchar(20) NOT NULL,
    NumTrophies tinyint,
    ManagerSsn char(10) NOT NULL DEFAULT '999999999',
    PRIMARY KEY (Name),
    FOREIGN KEY (ManagerSsn) REFERENCES Managers(Ssn) ON DELETE SET DEFAULT
    ON UPDATE CASCADE
);
```

Имаме първичен ключ – име на отборът, атрибути брой трофеи, Ssn на мениджърът – foreign key към атрибутът Ssn на релацията Managers, като по подразбиране се слага ,999999999’ – Ssn на специалният запис ‘Vacated’. При изтриване на информацията за мениджър, ManagerSsn става стойността по подразбиране, а при обновяване на информацията за мениджър в таблицата Managers, промените се правят и в таблицата Teams.

Примерни записи в таблицата добавяме с кодът:

```
insert into Teams values('Manchester United', 20, '1234567891')
insert into Teams values('Tottenham', 2, '1234567892')
insert into Teams values('Chelsea', 6, '1234567893')
insert into Teams(Name, NumTrophies) values('Manchester City', 4)
```

В резултат получаваме следната таблица:



Teams	Name	NumTrophies	ManagerSsn
	Chelsea	6	1234567893
	Manchester City	4	9999999999
	Manchester United	20	1234567891
	Tottenham	2	1234567892

Фигура 4. Таблица Teams

c. Matches (HomeTeamName, AwayTeamName, Result)

Кодът, за съставянето на тази релация е:

```
create table Matches(
  HomeTeamName varchar(20) NOT NULL,
  AwayTeamName varchar(20) NOT NULL,
  result char(3) CHECK(result like '_:_' ),
  PRIMARY KEY(HomeTeamName, AwayTeamName),
  FOREIGN KEY (HomeTeamName) REFERENCES Teams(Name),
  FOREIGN KEY (AwayTeamName) REFERENCES Teams(Name),
  CHECK( HomeTeamName != AwayTeamName )
);
```

Имаме ограничение първичен ключ да е множеството от HomeTeamName и AwayTeamName, като за тях имаме още ограничение тип CHECK – двете имена да са различни (няма логика отбор да играе срещу себе си), и две отделни ограничения от тип Foreign key към атрибутът Name на релацията Teams. Освен това имаме и атрибут result с CHECK ограничение – проверка дали резултатът е низ от вида ,\_:\_' (приемаме, че никой отбор няма да вкара двуцифрен брой голове в един мач).

Примерни записи в таблицата добавяме с кодът:

```
insert into Matches values('Manchester United', 'Tottenham', '2:1')
insert into Matches values('Tottenham', 'Manchester United', '0:2')
insert into Matches values('Manchester United', 'Chelsea', '3:0')
insert into Matches values('Manchester city', 'Tottenham', '1:2')
insert into Matches values('Tottenham', 'Chelsea', '2:1')
```

В резултат получаваме следната таблица:

Matches	HomeTeamName	AwayTeamName	result
	Manchester city	Tottenham	01:02
	Manchester United	Chelsea	03:00
	Manchester United	Tottenham	02:01
	Tottenham	Chelsea	02:01
	Tottenham	Manchester United	00:02

Фигура 5. Таблица Matches

d. TeamGrounds (TeamName, GroundName)

Кодът, за съставянето на тази релация е:

```
create table TeamGrounds(  
    TeamName varchar(20),  
    GroundName varchar(20),  
    PRIMARY KEY(TeamName, GroundName),  
    FOREIGN KEY (TeamName) REFERENCES Teams(Name) ON DELETE CASCADE ON UPDATE  
CASCADE,  
);
```

Двата атрибута образуват ключът на тази релация. Освен това имаме ограничение за външен ключ към атрибутът Name на релацията Teams и инструкции за каскадни операции при изтриване и обновяване на запис от релацията Teams.

Примерни записи в таблицата добавяме с кодът:

```
insert into TeamGrounds values('Manchester United', 'Old Trafford')  
insert into TeamGrounds values('Manchester City', 'Etihad')  
insert into TeamGrounds values('Tottenham', 'Wembley')  
insert into TeamGrounds values('Chelsea', 'Stamford Bridge')
```

В резултат получаваме следната таблица:

TeamGrounds	TeamName	GroundName
	Chelsea	Stamford Bridge
	Manchester City	Etihad
	Manchester United	Old Trafford
	Tottenham	Wembley

Фигура 6. Таблица TeamGrounds

- е. Players (Ssn, KitNumber, TeamName, Fname, Lname, Age, Salary, Address)

Кодът, за съставянето на тази релация е:

```
create table Players(  
    Ssn char(10),  
    KitNumber tinyint,  
    TeamName varchar(20) NOT NULL,  
    Fname varchar(10),  
    LName varchar(10),  
    Age tinyint,  
    Salary tinyint, --in millions  
    Address varchar(30),  
    PRIMARY KEY(Ssn),  
    FOREIGN KEY (TeamName) REFERENCES Teams(Name) ON DELETE CASCADE ON UPDATE  
CASCADE  
);
```

Първичен ключ е атрибутът Ssn, а освен това отново имаме ограничение за външен ключ към атрибутът Name на релацията Teams и инструкции за каскадни операции при изтриване и обновяване на запис от релацията Teams, както и ограничение TeamName да не е NULL – все пак всеки играч задължително участва в отбор.

Примерни записи в таблицата добавяме с кодът:

```
insert into Players values('0000000001', '25', 'Manchester United', 'Antonio',  
'Valencia', 32, 16, 'Downtown Abbey, Manchester')  
insert into Players values('0000000002', '7', 'Manchester United', 'Alexis',  
'Sanchez', 29, 45, 'Stretford, Manchester')  
insert into Players values('0000000003', '9', 'Manchester United', 'Romelu',  
'Lukaku', 25, 30, 'Downtown Abbey, Manchester')  
insert into Players values('0000000004', '6', 'Manchester United', 'Paul',  
'Pogba', 25, 50, 'Downtown Abbey, Manchester')  
  
insert into Players values('0000000005', '7', 'Manchester City', 'Rahim',  
'Sterling', 23, 32, 'Downtown Abbey, Manchester')  
insert into Players values('0000000006', '10', 'Manchester City', 'Sergio',  
'Aguero', 29, 27, 'Stretford, Manchester')  
  
insert into Players values('0000000007', '7', 'Tottenham', 'Heung-min', 'Son',  
25, 20, 'Trafalgar Square, London')  
insert into Players values('0000000008', '9', 'Tottenham', 'Harry', 'Kane', 25,  
22, 'Upper side, London')  
  
insert into Players values('0000000009', '7', 'Chelsea', 'Eden', 'Hazard', 25,  
36, 'Trafalgar Square, London')
```

В резултат получаваме следната таблица:

Players	Ssn	KitNumber	TeamName	Fname	LName	Age	Salary	Address
	0000000001	25	Manchester United	Antonio	Valencia	32	16	Downtown Abbey, Manchester
	0000000002	7	Manchester United	Alexis	Sanchez	29	45	Stretford, Manchester
	0000000003	9	Manchester United	Romelu	Lukaku	25	30	Downtown Abbey, Manchester
	0000000004	6	Manchester United	Paul	Pogba	25	50	Downtown Abbey, Manchester
	0000000005	7	Manchester City	Rahim	Sterling	23	32	Downtown Abbey, Manchester
	0000000006	10	Manchester City	Sergio	Aguero	29	27	Stretford, Manchester
	0000000007	7	Tottenham	Heung-min	Son	25	20	Trafalgar Square, London
	0000000008	9	Tottenham	Harry	Kane	25	22	Upper side, London
	0000000009	7	Chelsea	Eden	Hazard	25	36	Trafalgar Square, London

Фигура 7. Таблица Players

## 7. ПРИМЕРНИ ЗАЯВКИ – УСЛОВИЕ, КОД И РЕЗУЛТАТ

- 1) Условие: Изведете отбора, имената и възрастта на всички футболисти, които живеят на 'Downtown Abbey, Manchester' и тренират на Old Trafford

Код:

```
select TeamName, Fname, LName, Age  
from Players  
where Address = 'Downtown Abbey, Manchester'  
and TeamName in (select TeamName from TeamGrounds where GroundName = 'Old  
Trafford')
```

Резултат:

TeamName	Fname	LName	Age
Manchester United	Paul	Pogba	25
Manchester United	Romelu	Lukaku	25
Manchester United	Antonio	Valencia	32

Фигура 8. Резултат от първата примерна заявка

- 2) Условие: Изведете името на мениджърът начело на отбора, в който играе Harry Kane  
Код:

```
select m.Name
from Players p join Teams t on p.TeamName = t.Name and p.Fname='Harry' and
p.LName='Kane'
join Managers m on t.ManagerSsn = m.Ssn
```

Резултат:

Name
Mauricio Pochettino

Фигура 9. Резултат от втората примерна заявка

- 3) Условие: За всеки отбор, в който има поне един играч, изведете името на отбора, общата заплата, която плаща на играчи, и заплата, която плаща на мениджър  
Код:

```
select TeamName, sum(p.salary) as PlayerSalaries, max(m.salary) as ManagerSalary
from Players p join Teams t on p.TeamName = t.Name join Managers m on t.ManagerSsn =
m.Ssn
group by TeamName
```

Резултат:

TeamName	PlayerSalaries	ManagerSalary
Chelsea	36	19
Manchester City	59	0
Manchester United	141	24
Tottenham	42	17

Фигура 10. Резултат от третата примерна заявка

- 4) Условие: За всеки изигран мач, в който има двама футболисти с един и същ номер, изкарайте имената на двата отбора, номера и имената на тези футболисти

Код:

```
select m.HomeTeamName, m.AwayTeamName, pht.KitNumber, pht.Fname, pht.LName, pat.Fname,
pat.LName
from Matches m join Players pht on pht.TeamName = m.HomeTeamName join Players pat on
pat.TeamName = m.AwayTeamName
where pht.KitNumber = pat.KitNumber
```

Резултат:

HomeTeamName	AwayTeamName	KitNumber	Fname	LName	Fname	LName
Manchester United	Chelsea	7	Alexis	Sanchez	Eden	Hazard
Tottenham	Chelsea	7	Heung-min	Son	Eden	Hazard
Tottenham	Manchester United	7	Heung-min	Son	Alexis	Sanchez
Manchester city	Tottenham	7	Rahim	Sterling	Heung-min	Son
Manchester United	Tottenham	7	Alexis	Sanchez	Heung-min	Son
Tottenham	Manchester United	9	Harry	Kane	Romelu	Lukaku
Manchester United	Tottenham	9	Romelu	Lukaku	Harry	Kane

Фигура 11. Резултат от четвъртата примерна заявка

- 5) Условие: За всеки мениджър, различен от специалния запис 'Vacated', изведете всички играчи, които той ръководи. Ако за някой мениджър няма нито един играч да се изписва NULL

Код:

```
select m.Name, p.Fname, p.LName
from Managers m left join Teams t on m.Ssn = t.ManagerSsn left join Players p on
t.Name = p.TeamName
where m.Name != 'Vacated'
```

Резултат:

Name	Fname	LName
Alex Ferguson	NULL	NULL
Jose Mourinho	Paul	Pogba
Jose Mourinho	Alexis	Sanchez
Jose Mourinho	Romelu	Lukaku
Jose Mourinho	Antonio	Valencia
Mauricio Pochettino	Heung-min	Son
Mauricio Pochettino	Harry	Kane
Antonio Conte	Eden	Hazard

Фигура 12. Резултат от петата примерна заявка

- 6) Условие: За всеки играч изкарайте всички резултати от мачовете на неговия отбор.

Код:

```
select p.Fname, p.LName, t.Name as PlaysIn, m.*
from Players p join Teams t on p.TeamName = t.Name join Matches m on (t.Name =
m.HomeTeamName OR t.Name = m.AwayTeamName)
order by p.Fname, p.LName
```



Резултат:

Fname	LName	PlaysIn	HomeTeamName	AwayTeamName	result
Alexis	Sanchez	Manchester United	Manchester United	Chelsea	03:00
Alexis	Sanchez	Manchester United	Manchester United	Tottenham	02:01
Alexis	Sanchez	Manchester United	Tottenham	Manchester United	00:02
Antonio	Valencia	Manchester United	Manchester United	Chelsea	03:00
Antonio	Valencia	Manchester United	Manchester United	Tottenham	02:01
Antonio	Valencia	Manchester United	Tottenham	Manchester United	00:02
Eden	Hazard	Chelsea	Manchester United	Chelsea	03:00
Eden	Hazard	Chelsea	Tottenham	Chelsea	02:01
Harry	Kane	Tottenham	Manchester city	Tottenham	01:02
Harry	Kane	Tottenham	Manchester United	Tottenham	02:01
Harry	Kane	Tottenham	Tottenham	Chelsea	02:01
Harry	Kane	Tottenham	Tottenham	Manchester United	00:02
Heung-min	Son	Tottenham	Manchester city	Tottenham	01:02
Heung-min	Son	Tottenham	Manchester United	Tottenham	02:01
Heung-min	Son	Tottenham	Tottenham	Chelsea	02:01
Heung-min	Son	Tottenham	Tottenham	Manchester United	00:02
Paul	Pogba	Manchester United	Manchester United	Chelsea	03:00
Paul	Pogba	Manchester United	Manchester United	Tottenham	02:01
Paul	Pogba	Manchester United	Tottenham	Manchester United	00:02
Rahim	Sterling	Manchester City	Manchester city	Tottenham	01:02
Romelu	Lukaku	Manchester United	Manchester United	Chelsea	03:00
Romelu	Lukaku	Manchester United	Manchester United	Tottenham	02:01
Romelu	Lukaku	Manchester United	Tottenham	Manchester United	00:02
Sergio	Aguero	Manchester City	Manchester city	Tottenham	01:02

Фигура 13. Резултат от шестата примерна заявка

## 8. ПРЕДЛОЖЕНИЕ ЗА ПОТРЕБИТЕЛСКИ ИНТЕРФЕЙС

В този параграф ще демонстрираме идеи за потребителски интерфейс за няколко от заявките от точка 7.

Примерна заявка 3:

Футболната агенция иска да види информация за всеки отбор - какви заплати общо дава в момента на футболисти и на мениджърът. Все пак в съвременния футбол парите са една от основните мотивации на играчите и на база на информацията от тази заявка агенцията може да прецени към кой отбор да насочи свой клиент.

## Преглед на финансовото състояние на отборите:

Име на отбор	Обща заплата на играчи	Заплата на мениджър
Chelsea	36 милиона	19 милиона
Manchester City	59 милиона	В момента позицията е празна
Manchester United	141 милиона	24 милиона
Tottenham	42 милиона	17 милиона

Фигура 14. Примерен интерфейс за финансовото състояние на отборите

Примерна заявка 5:

Често срещано в днешно време е футболисти да предпочетат отбор, в който познават някой играч – било то от националния отбор или от друго място. Нека имаме следната ситуация: Маурисио Почетино иска да привлече Рахим Стърлинг – футболист от Манчестър Сити и английският национален отбор. Футболната агенция разполага със списък на текущия състав на английският национален отбор. Трябва да проверим кои играчи ръководи в момента Маурисио Почетино и дали някой от тях играе в националния отбор на Англия, за да го изпратим да вербува Стърлинг. За целта можем да използваме заявка 5, като трябва допълнително да филтрираме информацията да бъде само за определен мениджър. Това може да стане например по следният начин:

```
select task5.Name, task5.LastName
from (<кодът от заявка 5 като подзаявка>) task5
where managerName = 'Mauricio Pochettino'
```

## Справка кои играчи ръководи даден мениджър

Изберете мениджър:

Mauricio Pochettino



Играчите, които Mauricio Pochettino ръководи, са:

Име	Фамилия
Heung-min	Son
Harry	Kane

Фигура 15. Примерен интерфейс за информация кои играчи ръководи даден мениджър

Примерна заявка 6:

Цената на един играч пряко зависи от резултатите на неговият отбор. Ако например играч играе в Манчестър Юнайтед и те са в серия от 9 поредни победи, то това би покачило значително цената на всички играчи на този отбор. Футболната агенция иска за всеки играч да разполага с информация как се представя отборът му в момента. Трябва само да променим заявка 6 да показва информацията само за определен футболист и може да визуализираме резултата по следния начин:

Информация за мачовете на отбор, в който играе определен играч

Изберете играч:

Alexis Sanchez

✓

Избраният играч играе в:

Manchester United

Резултати на този отбор:

Домакин	Гост	Резултат
Manchester United	Chelsea	3:0
Manchester United	Tottenham	2:1
Tottenham	Manchester United	0:2

Фигура 16. Примерен интерфейс за информация за мачовете на отбор, в който играе определен играч

Кодът за тази заявка изглежда по следния начин:

```
select HomeTeamName, AwayTeamName, result
from (<кодът от заяка 6 като подзаявка, но без клаузата order by>) as task6
where firstName = 'Alexis' and lastName = 'Sanchez'
```