

CS 350 – Database Systems Term Project

Part 4: Implementation

Muhammed Fatih Öztel
218262744

Part 1: Motivation and Requirements

Motivation

In a sense, Scouting at the right time and right place costs a lot of money for the clubs. In this project, our aim is to put football fans, players and clubs together on one platform. Users will analyze the players from 1-100 in different skills like technical and physical. This platform serves football clubs to find and explore new young players for the positions that their teams need. So clubs can explore new young talents and add them to their academies. Also this platform provides users to be a professional scout in big clubs and make money from scout-missions. This will keep users motivated to share and analyze the players.

Requirements

- The tool will allow users to analyze and share their comments about football players for general and specific matches.
- Users will be able to create an account free and see all information and analysis about the players.
- Clubs will be able to list the trend and talented players and buy the video based and well detailed analysis by the scouts.
- Players will be able to see which skills they need to improve and plan their career with the tool.
- Active users will be able to get scout-missions which they scout the specific players and record the players match.

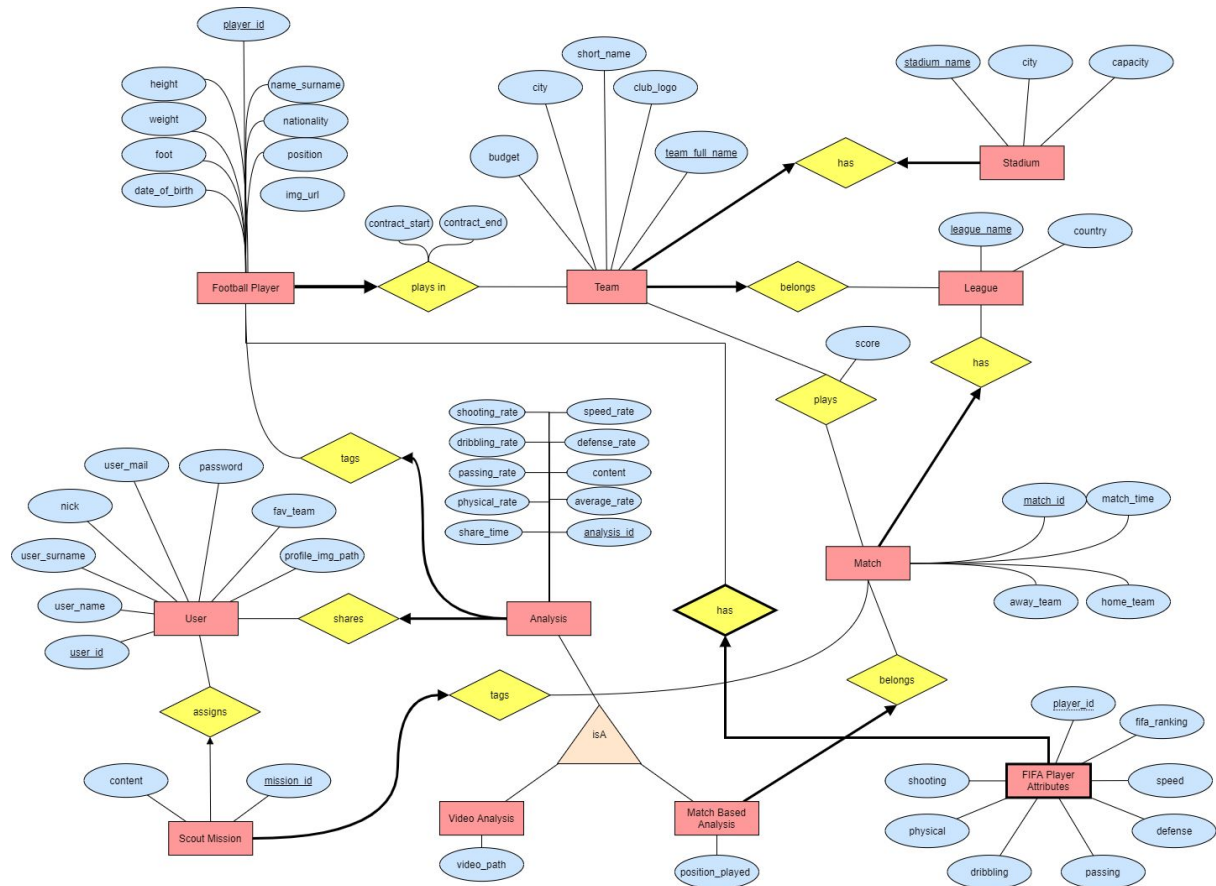
Part 2: Logical Design

Final Data Requirements

- Each football player has a player_id, name_surname, nationality, position, height, weight, foot and date_of_birth, img_url.
- Each team has an team_fullname, short_name, city and budget, club_logo.
- Each stadium has a stadium_name, city and capacity.
- Each league has an league_name and country.
- Each matches has a match_id, match_time, home_team, away_team.
- Each FIFA player attributes has a player_id, fifa_ranking, dribbling, shooting, physical, speed, defense and passing.

- Each user has a user_id, user_name, user_surname, nick, user_mail, fav_team, password and profile_img_path.
- Each scout mission has a mission_id and content.
- Each analysis has a analysis_id, share_time, content, shooting_rate, dribbling_rate, passing_rate, physical_rate, speed_rate, defense_rate and average_rate.
- Each video analysis **is** A analysis and has a video attribute.
- Each match based analysis **is** A analysis and has position_played.
- Each match based analysis belongs to **exactly one match**.
- A match may have **0 or more match based analysis**.
- Each team has **exactly one** stadium.
- Each stadium has **exactly one** team.
- Each Football Player plays in **exactly one** team during the contract_start and contract_end time.
- A Football Player may have **0 or more FIFA Attributes**.
- FIFA Player Attributes entity is weak, it exists **if and only if there exists a Football player**.
- A team may have **0 or more football players**.
- Each team belongs to **exactly one** league.
- A league may have **0 or more teams**.
- A team may play **0 or more** matches where this relation has score attribute.
- A match should be played by **exactly 2 team**.(*Since there is no representation of this relationship, it is shown as many to many relationship in ER diagram)
- Each match has **exactly one** league to be played.
- A league may have **0 or more matches**.
- A user may share **0 or more analysis**.
- Each analysis has **exactly one User** (owner).
- Each analysis tags **exactly one player**.
- A football player may have **0 or more analysis**.
- A user may assign to **0 or more Scout missions**.
- A scout mission should have **either 0 or 1 user**.
- Each scout mission has **exactly one** match.

ER Design



**As I also mentioned above Team and Match has many to many relationship because of the case.

Logic Design

DDL statements of Tables

```

create table if not exists Player(
    player_id int auto_increment,
    name_surname varchar(75) not null,
    nationality varchar(20) not null,
    position varchar(3) not null,
    date_of_birth date not null,
    height float not null,
    weight float not null,
    foot varchar(5) check (foot in ('Right','Left','Both')) not null,
    img_url varchar(255) not null,
    primary key (player_id)
);
    
```

```
create table if not exists League(  
    league_name varchar(50) not null,  
    country varchar(25) not null,  
    primary key (league_name)  
);
```

```
create table if not exists Team(  
    team_fullname varchar(50),  
    short_name char(3) unique,  
    league_name varchar(50),  
    city varchar(15),  
    budget float,  
    primary key (team_fullname),  
    foreign key (league_name) references League(league_name)  
);
```

```
create table if not exists Stadium(  
    stadium_name varchar(50) not null,  
    city varchar(25) not null,  
    capacity int,  
    primary key (stadium_name)  
);
```

```
create table if not exists Matches(  
    match_id int auto_increment,  
    home_team varchar(50),  
    away_team varchar(50),  
    match_time timestamp,  
    league_name varchar(30),  
    primary key (match_id),  
    foreign key (home_team) references Team(team_fullname) on delete cascade,  
    foreign key (away_team) references Team(team_fullname) on delete cascade,  
    foreign key (league_name) references League(league_name) on delete cascade  
);
```

```
create table if not exists Score(  
    match_id int primary key references Matches(match_id) on delete cascade,  
    score varchar(5) not null  
);
```

```
create table if not exists fifaAttributes(  
    player_id int primary key references Player(player_id) on delete cascade,
```

```
fifa_ranking float not null,  
dribbling float not null,  
shooting float not null,  
physical float not null,  
speed float not null,  
defense float not null,  
passing float not null  
);
```

```
create table if not exists Users(  
    user_id int auto_increment,  
    user_name varchar(50) not null,  
    user_surname varchar(50) not null,  
    user_mail varchar(255) not null unique,  
    nick varchar(50) not null unique,  
    fav_team varchar(50),  
    password varchar(255),  
    profile_img_path varchar(256) default null,  
    primary key (user_id),  
    foreign key (fav_team) references Team(team_fullname)  
);
```

```
create table if not exists ScoutMission(  
    mission_id int auto_increment,  
    match_id int not null,  
    content varchar(256),  
    primary key(mission_id),  
    foreign key (match_id) references Matches(match_id) on delete cascade  
);
```

```
create table if not exists Analysis(  
    analysis_id int not null auto_increment,  
    share_time timestamp,  
    content varchar(140),  
    owner_id int not null ,  
    player_id int not null ,  
    shooting_rate float not null,  
    dribbling_rate float not null,  
    passing_rate float not null,  
    physical_rate float not null,  
    speed_rate float not null,  
    defense_rate float not null,  
    average_rate float not null,  
    primary key (analysis_id),  
    foreign key (owner_id) references Users(user_id) on delete cascade,  
    foreign key (player_id) references Player(player_id) on delete cascade
```

);

```
create table if not exists VideoAnalysis(  
    analysis_id int primary key references Analysis(analysis_id),  
    video_path varchar(256)  
);
```

```
create table if not exists MatchBasedAnalysis(  
    matchanalysis_id int primary key references Analysis(analysis_id),  
    position_played varchar(3),  
    match_id int not null,  
    foreign key (match_id) references Matches(match_id)  
);
```

```
create table if not exists StadiumOwner(  
    team_name varchar(50) unique references Team(team_fullname) on delete cascade,  
    stadium varchar(50) unique references Stadium(stadium_name) on delete cascade,  
    primary key (team_name)  
);
```

```
create table if not exists plays_in(  
    player_id int not null,  
    team varchar(50) not null,  
    contract_start date not null,  
    contract_end date not null,  
    primary key(player_id, contract_start,contract_end),  
    foreign key (player_id) references Player(player_id) on delete cascade,  
    foreign key (team) references Team(team_fullname) on delete cascade  
);
```

```
create table if not exists MissionAssigning(  
    user_id int references Users(user_id) on delete cascade,  
    mission_id int references ScoutMission(mission_id) on delete cascade,  
    primary key (mission_id)  
);
```

Part 3: Normalization

Functional Dependencies

Player Table FDs:

$\{\underline{\text{player_id}}\} \rightarrow \{\underline{\text{player_id}}, \text{name_surname}, \text{nationality}, \text{position}, \text{date_of_birth}, \text{height}, \text{weight}, \text{foot}, \text{img_url}\}$

League Table FDs:

$\{\underline{\text{league_name}}\} \rightarrow \{\underline{\text{league_name}}, \text{country}\}$

Team Table FDs:

$\{\text{team_fullname}\} \rightarrow \{\text{team_fullname}, \text{short_name}, \text{league_name}, \text{city}, \text{budget}\}$
 $\{\text{short_name}\} \rightarrow \{\text{team_fullname}, \text{short_name}, \text{league_name}, \text{city}, \text{budget}\}$
 $\{\text{team_fullname}, \text{short_name}\} \rightarrow \{\text{team_fullname}, \text{short_name}, \text{league_name}, \text{city}, \text{budget}\}$

Stadium Table FDs:

$\{\text{stadium_name}\} \rightarrow \{\text{stadium_name}, \text{city}, \text{capacity}\}$

Matches Table FDs:

$\{\text{match_id}\} \rightarrow \{\text{match_id}, \text{home_team}, \text{away_team}, \text{match_time}, \text{league_name}\}$
 $\{\text{home_team}, \text{away_team}, \text{match_time}\} \rightarrow \{\text{match_id}, \text{home_team}, \text{away_team}, \text{match_time}, \text{league_name}\}$
 $\{\text{match_id}, \text{home_team}, \text{away_team}, \text{match_time}\} \rightarrow \{\text{match_id}, \text{home_team}, \text{away_team}, \text{match_time}, \text{league_name}\}$

Score Table FDs:

$\{\text{match_id}\} \rightarrow \{\text{match_id}, \text{score}\}$

fifaAttributes Table FDs:

$\{\text{player_id}\} \rightarrow \{\text{player_id}, \text{fifa_ranking}, \text{dribbling}, \text{shooting}, \text{physical}, \text{speed}, \text{defense}, \text{passing}\}$

Users Table FDs:

$\{\text{user_id}\} \rightarrow \{\text{user_id}, \text{user_name}, \text{user_surname}, \text{user_mail}, \text{nick}, \text{password}, \text{fav_team}, \text{password}, \text{profile_img_path}\}$
 $\{\text{nick}\} \rightarrow \{\text{user_id}, \text{user_name}, \text{user_surname}, \text{user_mail}, \text{nick}, \text{password}, \text{fav_team}, \text{password}, \text{profile_img_path}\}$
 $\{\text{user_mail}\} \rightarrow \{\text{user_id}, \text{user_name}, \text{user_surname}, \text{user_mail}, \text{nick}, \text{password}, \text{fav_team}, \text{password}, \text{profile_img_path}\}$
 $\{\text{nick}, \text{user_mail}\} \rightarrow \{\text{user_id}, \text{user_name}, \text{user_surname}, \text{user_mail}, \text{nick}, \text{password}, \text{fav_team}, \text{password}, \text{profile_img_path}\}$
 $\{\text{user_id}, \text{nick}\} \rightarrow \{\text{user_id}, \text{user_name}, \text{user_surname}, \text{user_mail}, \text{nick}, \text{password}, \text{fav_team}, \text{password}, \text{profile_img_path}\}$
 $\{\text{user_id}, \text{mail}\} \rightarrow \{\text{user_id}, \text{user_name}, \text{user_surname}, \text{user_mail}, \text{nick}, \text{password}, \text{fav_team}, \text{password}, \text{profile_img_path}\}$
 $\{\text{user_id}, \text{nick}, \text{mail}\} \rightarrow \{\text{user_id}, \text{user_name}, \text{user_surname}, \text{user_mail}, \text{nick}, \text{password}, \text{fav_team}, \text{password}, \text{profile_img_path}\}$

ScoutMission Table FDs:

$\{\text{mission_id}\} \rightarrow \{\text{mission_id}, \text{match_id}, \text{content}\}$

Analysis Table FDs:

$\{\text{analysis_id}\} \rightarrow \{\text{analysis_id}, \text{share_time}, \text{content}, \text{owner_id}, \text{player_id}, \text{shooting_rate}, \text{dribbling_rate}, \text{passing_rate}, \text{physical_rate}, \text{speed_rate}, \text{defense_rate}, \text{average_rate}\}$

VideoAnalysis Table FDs:

$\{\text{analysis_id}\} \rightarrow \{\text{analysis_id}, \text{video_path}\}$

MatchBasedAnalysis Table FDs:

$\{\text{analysis_id}\} \rightarrow \{\text{analysis_id}, \text{position_played}, \text{match_id}\}$

StadiumOwner Table FDs:

$\{\text{team_name}\} \rightarrow \{\text{team_name}, \text{stadium_name}\}$

plays_in Table FDs:

$\{\text{player_id}, \text{contract_start}\} \rightarrow \{\text{player_id}, \text{contract_start}, \text{team}, \text{contract_end}\}$

MissionAssigning Table FDs:

$\{\text{user_id}, \text{mission_id}\} \rightarrow \{\text{user_id}, \text{mission_id}\}$

Normalization

In this project, there was no bad FD. I mentioned the steps of normalization for each table to be in BCNF.

Player Table:

There exists only one functional dependency which is already determined by a super/candidate key (this key is primary also) in this relation. Since this dependency does not violates the BCNF condition, the Player table is in BCNF.

League Table:

There exists only one functional dependency which is already determined by a super/candidate key (this key is primary also) in this relation. So there is no bad FD, the League table is in BCNF.

Team Table:

There are few functional dependencies in this table. First one is already primary key that uniquely identifies each row in table. In Second dependency team short name is unique

for each team and can identifies each row of the table, so it's also a super/candidate key and determines all columns. Third dependency is combination of these 2 candidate keys. So there is no bad FD, this table is in BCNF.

Stadium Table:

Stadium table has just one functional dependency and this dependency is determined by a super/candidate key. So there is no bad FD, Stadium table is in BCNF

Matches Table:

First dependency is already primary key that uniquely identifies each row in table. In Second dependency home and away team and time is super/candidate key. Third dependency is combination of these 2 super key. So it's also a super/candidate key and determines all columns. Third dependency is combination of these 2 candidate keys. So there is no bad FD, this table is in BCNF.

Score Table:

Score table has just one functional dependency and this dependency has a candidate key as a determinant. So there is no bad FD, Score table is in BCNF.

fifaAttributes Table:

fifaAttributes table has just one functional dependency and this dependency has a candidate key as a determinant. So there is no bad FD, fifaAttributes table is in BCNF.

Users Table:

All functional dependencies are good functional dependencies where the dependencies determined by super/candidate keys. So, Users table is in BCNF.

ScoutMission Table:

Scout Mission table has just one functional dependency and this dependency is determined by a super/candidate key. So there is no bad FD, Scout Mission table is in BCNF.

Analysis Table:

Analysis table has just one functional dependency and this dependency is determined by a super/candidate key. So there is no bad FD, Analysis table is in BCNF.

VideoAnalysis Table:

Video Analysis table has just one functional dependency and this dependency is determined by a super/candidate key. So there is no bad FD, Video Analysis table is in BCNF.

MatchBasedAnalysis Table:

Match Based Analysis table has just one functional dependency and this dependency is determined by a super/candidate key. So there is no bad FD, Match Based Analysis table is in BCNF.

StadiumOwner Table:

Stadium Owner table has just one functional dependency and this dependency is determined by a super/candidate key. So there is no bad FD, Stadium Owner table is in BCNF.

plays_in Table:

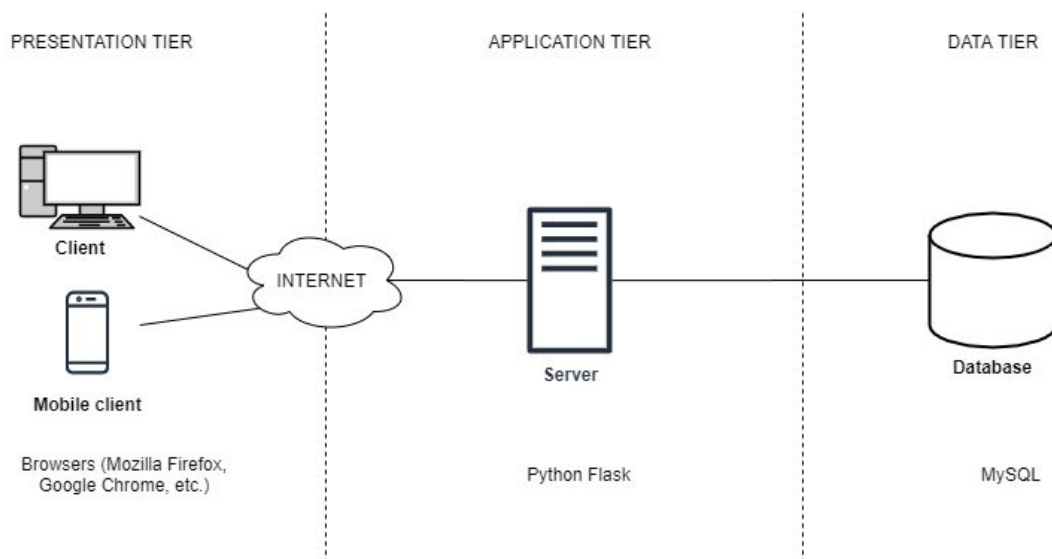
Plays in table has just one functional dependency and this dependency is determined by a super/candidate key. So there is no bad FD, Plays in table is in BCNF.

MissionAssigning Table:

Mission Assigning table has just one functional dependency and this dependency is determined by a super/candidate key. So there is no bad FD, Mission Assigning table is in BCNF.

Part 4: Implementation

Architecture of the application at high level:



Our application is web based application and it's in three tier architecture so it has presentation, application and data tiers. In the figure above, it shows the related tier and their connections.

Presentation Tier: This tier is built with HTML, CSS and JavaScript. It is represented as a displaying and showing through a web browser for clients. It communicates with other

tiers through the interface of application. It takes related computations from application server. Our web design is responsible for this tier in our project

Application Tier: The application tier is written in Python Flask. In our application Flask provides us to compute related datas and compile those informations for users. This tier takes data from database and represent it in related places after calculations. It manages communication between an the client interface and the database.

Data Tier: This tier consists of a database and read, write access to a database. In our example datas are hosted on local device. For database managing, writing and reading database, we used MySQL in this project.

List of SQL Queries:

I listed all queries here. I removed format operators and put related examples for queries and basically explained with semantics.

Query: SELECT * FROM ANALYSIS a

LEFT JOIN USERS u ON a.owner_id = u.user_id

LEFT JOIN PLAYS_IN p ON a.player_id = p.player_id and

p.contract_start<a.share_time<p.contract_end

LEFT JOIN PLAYER pl ON pl.player_id = a.player_id order by share_time desc

*Semantics:*Join analysis and users based on user_id, and join analysis and player based on player_id and contract validity and player with plays_in based on player_id and select all by descending order.

Query: select * from Users

Semantics: select all user information from users table.

Query: insert into

Users(user_name,user_surname,user_mail,nick,fav_team,password,profile_img_path)

values('Muhammed','Oztel','fatihozte@hotmail.com','ofli','Şehir

Club','1234','','../static/img/avatar-auto/default_scout.png')

Semantics: Insert into Users a new record with corresponding values.

Query: select * from Analysis where owner_id=(select user_id from Users where nick = 'ofli')

Semantics: select all analysis info where analysis owner id equals with the user nick's id

Query: select * from users where nick = 'ofli'

Semantics: select data of user where users' nick equal to ofli

Query: select player_id,name_surname, position,img_url from player where player_id = 1

Semantics: select player_id, name_surname,position and img_url that player id equals 1

Query: select team from plays_in where player_id=1

Semantics: select team that player_id is 1 from plays_in

Query: select * from MatchBasedAnalysis where matchanalysis_id = 1

Semantics: select matchbasedanalysis's id equals to 1 records

Query: select * from VideoAnalysis where analysis_id = 1

Semantics: select videoanalysis record with analysis_id equals 1

Query: select * from Player where player_id = 5

Semantics: select all columns of player with player_id equals to 5

Query: select * from (select p.player_id,p.team,p.contract_start,p.contract_end,
t.club_logo from plays_in p
JOIN team t on p.team = t.team_fullname) as drv
where player_id = 20

Semantics: Join plays_in and team table based on their team names and select player_id,
team contract_start, contract_end and club logo where player_id is 20

Query: select * from fifaattributes where player_id = 20"

Semantics: select all attributes from fifaattributes which has player_id as 20

Query: select * from analysis where player_id = 20

Semantics: select all columns of row which player_id is 20

Query: select * from users where user_id = 5

Semantics: select user row where user_id is 5

Query: select player_id,name_surname, position,img_url from player where player_id = 5

Semantics: select player_id, name surname, position and img_url of player where
player_id = 5

Query: select team from plays_in where player_id=5 and contract_start< now() <
contract_end

Semantics: select team of player with id 5 and existing contract term

Query: select * from MatchBasedAnalysis where matchanalysis_id = 5

Semantics: select all comuns of match based analysis with id 5

Query: select * from (select aa.player_id,aa.owner_id,a.position_played,
a.matchanalysis_id,a.match_id,
m.home_team, m.away_team, s.score from matchbasedanalysis a
left join matches m on a.match_id = m.match_id
left join score s on m.match_id = s.match_id
left join analysis aa on a.matchanalysis_id = aa.analysis_id) as drv
where drv.player_id = 5 and drv.match_id = 10

Semantics: First join matchbasedanalysis with matches based on match_ids, then join with score based on match id, then join wit analysis on analysis id, second select player id, owner id, position_played, analysis id, match id, home team, away team and score where player id is 5 and match id is 10

Query: select * from VideoAnalysis where analysis_id = 41

Semantics: select all columns from videoanalysis where analysis id is 5

Query: insert into analysis(share_time,content,owner_id,player_id,shooting_rate,
dribbling_rate,passing_rate,physical_rate,
speed_rate,defense_rate, average_rate)
values (NOW(), 'Muhammed is very talented
player', 5,6 ,80,73,66,74,90,80,83)

Semantics: insert a new analysis with corresponding values

Query: SELECT * from Team WHERE team_fullname LIKE 'Şehir Club'

Semantics: select all columns from team name is similar to Şehir Club

Query: SELECT * from Team

Semantics: select all item from team table

Query: select * from Stadium where stadium_name = (select stadium from StadiumOwner
where team_name = 'Şehir Club')

Semantics: select all stadium records where the team name is Şehir Club and stadium owners stadium is equal team stadium

Query: select * from (SELECT p.player_id,p.name_surname,
p.date_of_birth,p.nationality,pi.team,p.position,
fa.fifa_ranking,p.img_url, contract_start,pi.contract_end
FROM player p JOIN plays_in pi ON p.player_id=pi.player_id
JOIN fifaattributes fa ON fa.player_id=pi.player_id) as drv
where drv.team = 'Şehir Club' and drv.contract_start< now() < contract_end

Semantics: Join player with plays_in and fifaattributes based on player_id and select related columns where team is Şehir Club and contract is valid.

Query: select * from (select m.match_id, m.home_team, m.away_team,
m.match_time, t1.league_name,
t1.team_fullname as home_name, t1.short_name as home_code,
t1.club_logo as t_1, t2.team_fullname as away_name, t2.short_name as
away_code, t2.club_logo as t_2 from matches m
LEFT JOIN team t1 on t1.team_fullname = m.home_team
LEFT JOIN team t2 on t2.team_fullname = m.away_team) as drv
where (drv.home_team='Şehir Club' or drv.away_team='Şehir Club') and
drv.match_time>now()

Semantics: Join matches based on team names with home_team and away_team columns, then select Şehir Clubs' next matches and other match informations

Query: select * from (select m.match_id, m.home_team,
m.away_team, m.match_time, t1.league_name,
t1.team_fullname as home_name, t1.short_name as home_code,
t1.club_logo as t_1, t2.team_fullname as away_name, t2.short_name as
away_code, t2.club_logo as t_2, s.score as sc from matches m
LEFT JOIN team t1 on t1.team_fullname = m.home_team
LEFT JOIN team t2 on t2.team_fullname = m.away_team
LEFT JOIN score s on s.match_id = m.match_id) as drv
where (drv.home_team='Şehir Club' or drv.away_team='Şehir Club') and
drv.match_time<now()

Semantics: Join matches based on team names with home_team and away_team columns, then select Şehir Clubs' old matches and other match informations

Query: select name_surname from player where player_id=16

Semantics: select name and surname of player with id 16

Query: select * from (SELECT p.player_id, p.name_surname, p.nationality, pi.team,
p.position, fa.fifa_ranking, p.img_url"
FROM player p JOIN plays_in pi ON p.player_id=pi.player_id
JOIN fifaattributes fa ON fa.player_id=pi.player_id order by p.player_id) as drv
where drv.name_surname like 'Muhammed Oztel'

Semantics: Join player with plays_in and fifaattributes and select related id, name, nationality, position, fifa ranking and img where name like 'Muhammed Oztel'

Query: select * from (SELECT p.player_id, p.name_surname, p.nationality, pi.team,
p.position, fa.fifa_ranking, p.img_url

```

FROM player p
JOIN plays_in pi ON p.player_id=pi.player_id
JOIN fifaattributes fa ON fa.player_id=pi.player_id order by p.player_id) as drv
where drv.team like 'Şehir Club'

```

Semantics: Join player with plays_in and fifaattributes and select related id, name, nationality, position, fifa ranking and img where team 'Şehir Club'

Query: select *from (SELECT p.player_id,p.name_surname,p.nationality,pi.team,
p.position,fa.fifa_ranking,p.img_url
FROM player p
JOIN plays_in pi ON p.player_id=pi.player_id
JOIN fifaattributes fa ON fa.player_id=pi.player_id order by p.player_id) as drv
where drv.nationality like 'Turkey'

Semantics: Join player with plays_in and fifaattributes and select related id, name, nationality, position, fifa ranking and img where nationality is similar to 'Turkey'

Query: select *from (SELECT p.player_id,p.name_surname,p.nationality,pi.team,
p.position,fa.fifa_ranking,p.img_url
FROM player p
JOIN plays_in pi ON p.player_id=pi.player_id
JOIN fifaattributes fa ON fa.player_id=pi.player_id order by p.player_id) as drv
where drv.position like 'CB'

Semantics: Join player with plays_in and fifaattributes and select related id, name, nationality, position, fifa ranking and img where position is similar to CB

Query: select *from (SELECT p.player_id,p.name_surname,p.nationality,pi.team,
p.position,fa.fifa_ranking,p.img_url
FROM player p
JOIN plays_in pi ON p.player_id=pi.player_id
JOIN fifaattributes fa ON fa.player_id=pi.player_id order by p.player_id) as drv
where drv.fifa_ranking >= 80

Semantics: Join player with plays_in and fifaattributes and select id, name, nationality, position, fifa ranking and img where fifa ranking is over 80

Programming Languages and DBMS:

This project coded in Python, in Flask. Also for dynamic web JavaScript is used. For database, MySQL used to store datas and reading and writing operations.

Data Size and Source:

There is the data size (number of rows) for each table:A

Player:492 Plays_in:492 FifaAttributes:492 Users:8 Analysis:18

Matchbasedanalysis:1 League:2 Team:19 Stadium:17 StadiumOwner:19
Matches:306 ScoutMission:3 MissionAssigning:1 VideoAnalysis:1 Score:234

i. I fetched data from internet websites with scraping tools and my own code and also I used ready dataset for players fifa rankings. I fetched and scraped data from transfermarkt for teams, from stadiumdb stadiums, from TFF official website matches. For players I used fifa dataset from kaggle.

ii. Yes, I entered manually some test data while connecting database with the application. But then i removed garbage datas, it's now functionally works

List of Working Features with Corresponding Screenshots:

- Registering

The screenshot displays the Scout application interface. At the top, there is a navigation bar with the 'Scout' logo on the left and links for 'Home', 'Log In', and 'Register' on the right. The 'Register' link is highlighted. Below the navigation bar, a dark teal modal box titled 'Register Form' is centered. This form contains five input fields: a username field with 'muhammedDB', a first name field with 'Muhammed Fatih', a last name field with 'Öztel', an email field with 'ffatihoztel@gmail.com', and a password field represented by seven dots. Below these fields is a white 'REGISTER' button. At the bottom of the modal, there is a link that says 'Do yo have an Account? [Login Now!](#)'.

- Login

Scout

Home Log In Register

Log In

muhammedDB

.....

LOGIN

- Searching for team with their names or all for all team, results displayed with table and it can be sorted with the related attribute

Scout

Stream Teams Players Log Out

Library

al

Result Table

Team Name	Shortcut	League	City	Market Value
Alanyaspor	ANY	Super Lig	Antalya	22.14
Antalyaspor	ANT	Super Lig	Antalya	18.43
Besiktas JK	BES	Super Lig	Istanbul	60.09
Caykur Rizespor	CAY	Super Lig	Rize	20.82
Denizlispor	DEN	Super Lig	Denizli	15.9
Fenerbahce SK	FEN	Super Lig	Istanbul	72.52
Galatasaray SK	GAL	Super Lig	Istanbul	91.74
Gaziantep FK	GAG	Super Lig	Gaziantep	21.12
Genclerbirligi Ankara	GSK	Super Lig	Ankara	14.71
Göztepe	GOZ	Super Lig	Izmir	16.01
Istanbul Basaksehir FK	KIBA	Super Lig	Istanbul	53.02
Kasimpasa	KSK	Super Lig	Istanbul	27.53
Kayserispor	KAY	Super Lig	Kayseri	17.05
Konyaspor	KON	Super Lig	Konya	18.84
MKE Ankaragücü	ANK	Super Lig	Ankara	11.91
Sehir Club	SCT	No League	Istanbul	1000.0
Sivasspor	SIV	Super Lig	Sivas	29.04
Trabzonspor	TRA	Super Lig	Trabzon	80.88
Yeni Malatyaspor	YMA	Super Lig	Malatya	18.81

Quick Links

- For each team there is a page can be seen related team information, old and next matches and all players plays in the team

The screenshot displays a web application interface for a football team, Besiktas JK. The browser address bar shows the URL: 127.0.0.1:5000/teams/team_code=BES.

Team Information:

- Team:** Besiktas JK
- Stadium:** Vodafone Park
- League:** Super Lig
- City:** Istanbul
- Market Value:** \$60.09M
- Capacity:** 41903

Team Fixture:

Opponent	Score	Date	Time
ANY	1 - 2	2020-02-28	20:00:00
BES	2 - 1	2020-03-06	20:00:00
GAL	0 - 0	2020-03-15	19:00:00

Football Players:

Player	Position	Nationality
Caner Erkin	LB	Turkey
Jeremain Lens	RM	Netherlands
Adem Ujajić	CAM	Serbia
Douglas Pereira dos Santos	RB	Brazil
Víctor Ruiz Torre	CB	Spain
Gary Alexis Medel Soto	CDM	Chile
Necip Uysal	CDM	Turkey

- In player pages we can search players with their names, positions, nationalities, overall ranking and teams.

0.0.1:5000/players

...

Scout

StreamTeamsPlayersLog Out











Library

Choose a Filter: Position

RM


Q

Result Table

Player Photo	Player Name	Nationality	Current Team	Position	Overall
	Ricardo Andrade Quaresma Bernardo	Portugal	Besiktas JK	RM	81.0
	Issiar Dia	Senegal	Yeni Malatyaspor	RM	68.0
	Victor Moses	Nigeria	Fenerbahce SK	RM	78.0
	Nabil Dirar	Morocco	Fenerbahce SK	RM	75.0
	Kenan Özer	Turkey	Gaziantep FK	RM	66.0
	Jeremain Lens	Netherlands	Besiktas JK	RM	77.0
	Niill De Pauw	Belgium	Caykur Rizespor	RM	71.0
	Sofiane Feghouli	Algeria	Galatasaray SK	RM	80.0
	Tunay Torun	Turkey	Istanbul Basaksehir FK	RM	71.0
	Gökhan Töre	Turkey	Yeni Malatyaspor	RM	74.0

- In player profile pages we can see related player informations and analysis about him.

Player Data



Player Name
 Ricardo Andrade Quaresma
 Bernardo

Nationality
 Portugal

Birthday
 1983-09-26

Height
 175.0
 cm

Weight
 67.0 kg


Preferred Foot
 Right

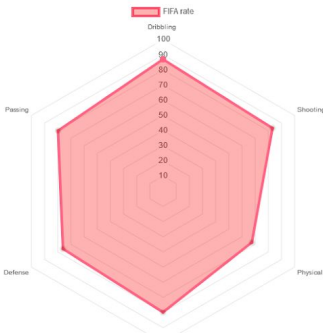
Team
 Besiktas JK

Position
 RM

Joined
 2015-07-22


Contract ends
 2020-12-12






+ ADD ANALYSIS

Analysis



Muhammed Oztel

off




Ricardo Andrade Quaresma Bernardo

Besiktas JK - RM

85.0


SHOOTING: 85.0 DRIBBLING: 85.0 PASSING: 85.0
 PHYSICAL: 85.0 SPEED: 85.0 DEFENSE: 85.0

very talented player



Muhammed Oztel

off



Ricardo Andrade Quaresma Bernardo

Besiktas JK - RM

75.0

SHOOTING: 60.0 DRIBBLING: 50.0 PASSING: 100.0
 PHYSICAL: 70.0 SPEED: 80.0 DEFENSE: 90.0

- We can analyze football player from his profile page with add analysis button.

Form

Dribbling:

65

Shooting:

75

Passing:

85

Speed:

85

Defense:

68

Physical:


75

Your Comment:


This is for db project manuel

SUBMIT

- In stream we can see latest analysis from all users.



Muhammed fatih Oztel
muhammedDB
2020-05-31 21:02:23




Selçuk İnan
Galatasaray SK - CM


77.8333

SHOOTING: 84.0 DRIBBLING: 75.0 PASSING: 85.0
PHYSICAL: 80.0 SPEED: 73.0 DEFENSE: 70.0

Very nice talented player



Muhammed fatih Oztel
muhammedDB
2020-05-31 21:00:27



Ricardo Andrade Quaresma Bernardo
Besiktas JK - RM

75.5

SHOOTING: 75.0 DRIBBLING: 65.0 PASSING: 85.0
PHYSICAL: 75.0 SPEED: 85.0 DEFENSE: 68.0

This is for db project manuel

- In users profile page we can see users analysis.

The screenshot shows the Scout app interface. At the top, there's a navigation bar with the 'Scout' logo and buttons for 'Stream', 'Teams', 'Players', and 'Log Out'. The main content area displays the profile of 'Muhammed fatih Oztel' (username: muhammedDB). Below the profile header, there's a section for 'Ricardo Andrade Quaresma Bernardo' (Besiktas JK - RM) with an overall rating of 75.5. This section includes a grid of performance metrics: SHOOTING: 75.0, DRIBBLING: 65.0, PASSING: 85.0, PHYSICAL: 75.0, SPEED: 85.0, and DEFENSE: 68.0. A note below the metrics reads 'This is for db project manuel'. Below this, another section for 'Muhammed fatih Oztel' (muhammedDB) displays the profile of 'Selçuk İnan' (Galatasaray SK - CM) with an overall rating of 77.8333. This section also includes a grid of performance metrics: SHOOTING: 84.0, DRIBBLING: 75.0, PASSING: 85.0, PHYSICAL: 80.0, SPEED: 73.0, and DEFENSE: 70.0. A note below the metrics reads 'Very nice talented player'.

- we can easily logout with logout button and redirect to the home page.

The screenshot shows the Scout app home page. At the top, there's a navigation bar with the 'Scout' logo and buttons for 'Home', 'Log In', and 'Register'. The main content area features a large, dark image of a football stadium at night. Overlaid on this image is the text 'Welcome to world best scouting platform' and 'Do you really think you are good at football?'. At the bottom center, there's a green button labeled 'Join Us!'.

List of non-working Features:

- Users can not assign to Scout Missions.
- Users can not add match based analysis and video analysis.
- Users can not see trend players.
- User can not edit their informations and set their favourite teams and profile pages.
- Overall rating calculation for user analysis is not based on player position. It's just average of all skills..