Project title

Soccer

Name of the project

Soccer Game Management System

Team Name

Fantastic 5

Team members

Huzefa Sadikot
Akshat Shanbhag
Rene Irias
Atreya Raorane
Raj Parekh

Project Scope and Planning

As the name suggests the main purpose of the database is to keep record of all events, games, player Profile, country leagues, private games. We are planning to make a working database which can store data related to games, and player which are related to each other. We want to make a system which can make viewing and storing all data related to this sport easy and fast. We want a system which is reliable and consistent. Each table has a unique significance to it with minimal redundant data.

The main purpose to make this project is to bring all the data from different database under one system. Currently there are different databases for player Profile, Games, Different Leagues, World cups etc. We intend to calculate different score boards, number of matches played by players, number of goals made by players depending upon this query can be fired to find a combined data and a customized view according to user's needs as calculated fields. Also, this can be customized to view the statistics for favorite players.

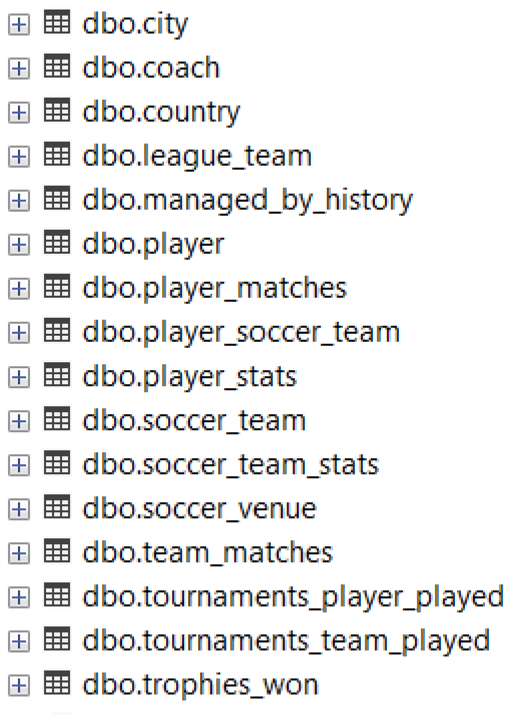
How will scouts use this database:

When scouts need young players for their team, they need player data to see many factors before picking them up for a particular role.

Keys factors that scout can use from the database: -

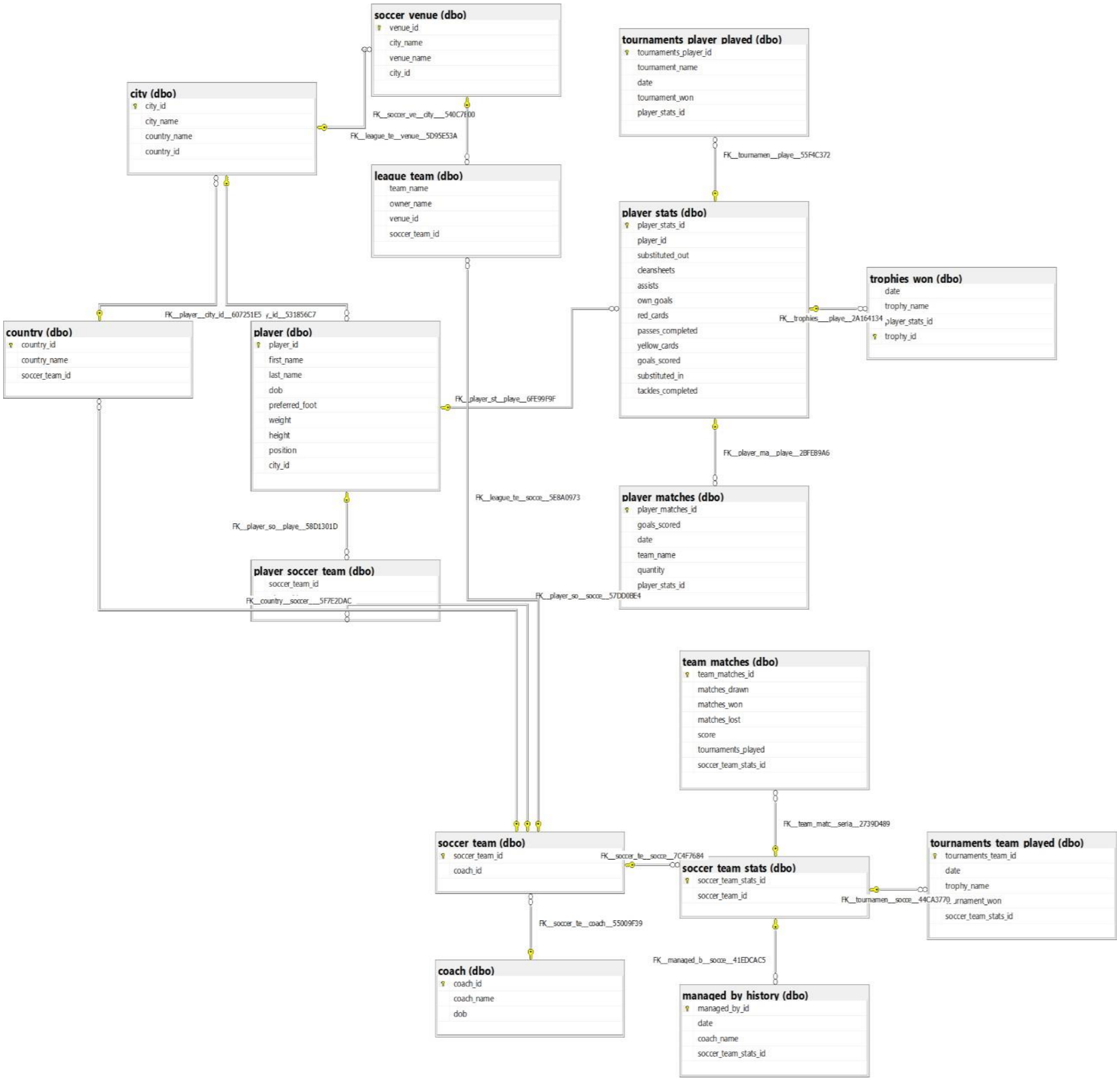
- Players Stats
- Players playing for which country
- Players team stats
- How many tournaments did the player play
- Player's availability

Here the schema is dbo and the table names used are displayed below:



dbo.city
dbo.coach
dbo.country
dbo.league_team
dbo.managed_by_history
dbo.player
dbo.player_matches
dbo.player_soccer_team
dbo.player_stats
dbo.soccer_team
dbo.soccer_team_stats
dbo.soccer_venue
dbo.team_matches
dbo.tournaments_player_played
dbo.tournaments_team_played
dbo.trophies_won

Logical Data Model



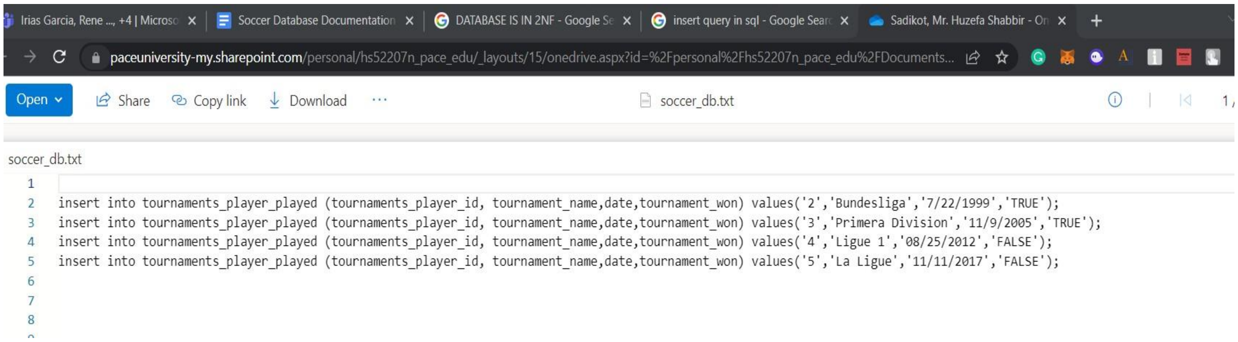
Create Database and load Data

Database we created is in 2NF

To create the database the following command is used;

CREATE DATABASE soccer;

To load data in the database insert query is used: -


Irias Garcia, Rene ..., +4 | Microso
Soccer Database Documentation
DATABASE IS IN 2NF - Google Se
insert query in sql - Google Searc
Sadikot, Mr. Huzefa Shabbir - On
paceuniversity-my.sharepoint.com/personal/hs52207n_pace_edu/_layouts/15/onedrive.aspx?id=%2Fpersonal%2Fhs52207n_pace_edu%2FDocuments...
Open
Share
Copy link
Download
soccer_db.txt
soccer_db.txt

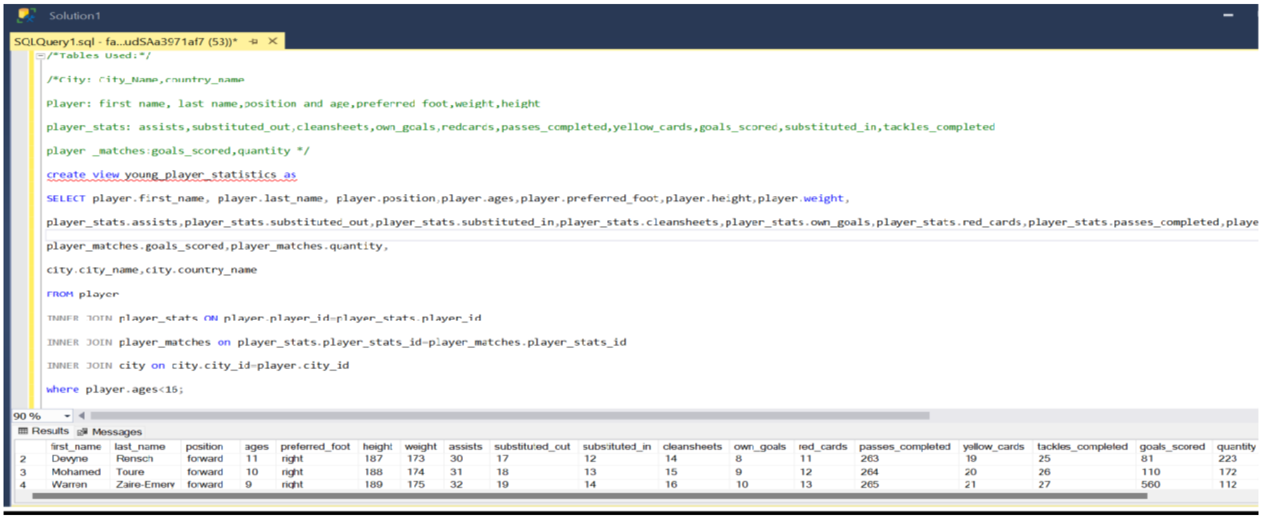
```
insert into tournaments_player_played (tournaments_player_id, tournament_name,date,tournament_won) values('2','Bundesliga','7/22/1999','TRUE');
insert into tournaments_player_played (tournaments_player_id, tournament_name,date,tournament_won) values('3','Primera Division','11/9/2005','TRUE');
insert into tournaments_player_played (tournaments_player_id, tournament_name,date,tournament_won) values('4','Ligue 1','08/25/2012','FALSE');
insert into tournaments_player_played (tournaments_player_id, tournament_name,date,tournament_won) values('5','La Ligue','11/11/2017','FALSE');
```

A view can be created for the user to see precise data referring to a specific topic

We have created 3 views

1. Top_Player_Statistics
2. Player_unavailibility_view
3. Young_Player_Statistics

Let us see how to create one view



```
/*Tables Used:*/

/*City: City_Name,country_name

Player: first name, last name,position and age,preferred foot,weight,height

player_stats: assists,substituted_out,cleansheets,own_goals,redcards,passes_completed,yellow_cards,goals_scored,substituted_in,tackles_completed

player _matches:goals_scored,quantity */

create view young_player_statistics as

SELECT player.first_name, player.last_name, player.position,player.ages,player.preferred_foot,player.height,player.weight,

player_stats.assists,player_stats.substituted_out,player_stats.substituted_in,player_stats.cleansheets,player_stats.own_goals,player_stats.red_cards,player_stats.passes_completed,playe

player_matches.goals_scored,player_matches.quantity,

city.city_name,city.country_name

FROM player

INNER JOIN player_stats ON player.player_id=player_stats.player_id

INNER JOIN player_matches on player_stats.player_stats_id=player_matches.player_stats_id

INNER JOIN city on city.city_id=player.city_id

where player.ages<16;
```

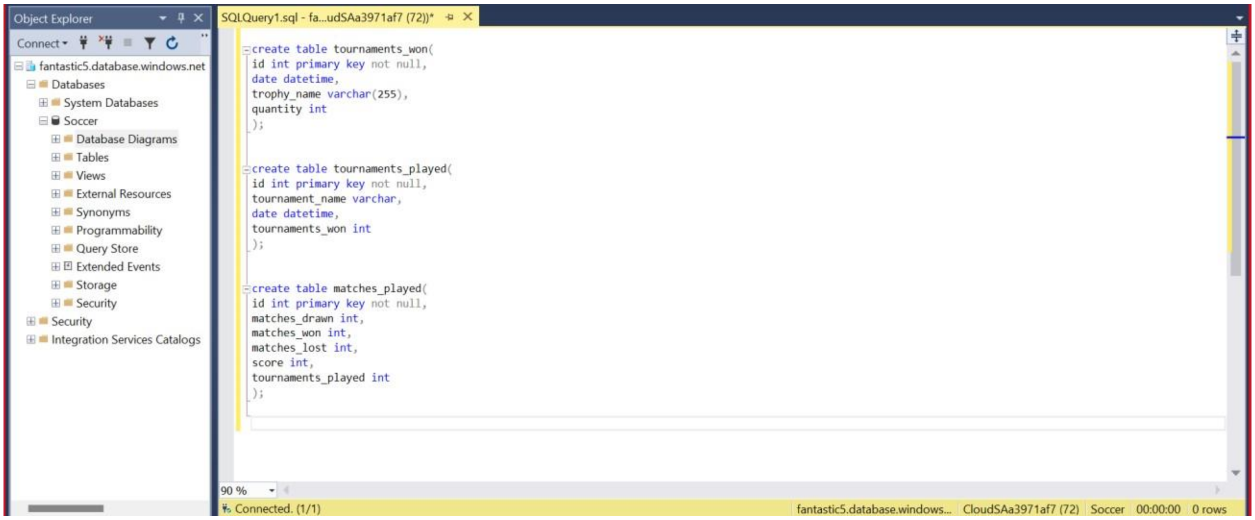
	first_name	last_name	position	ages	preferred_foot	height	weight	assists	substituted_out	substituted_in	cleansheets	own_goals	red_cards	passes_completed	yellow_cards	tackles_completed	goals_scored	quantity
2	Devyne	Rensch	forward	11	right	187	173	30	17	12	14	8	11	263	19	25	81	223
3	Mohamed	Toure	forward	10	right	188	174	31	18	13	15	9	12	264	20	26	110	172
4	Warren	Zaire-Emery	forward	9	right	189	175	32	19	14	16	10	13	265	21	27	560	112

Report and Visualization

Create, Insert, and view are the query that we have used for the soccer database

Create query

While creating the database for soccer we mainly focused on the primary key, the foreign key of the table created, and the datatypes of the column.



```sql
create table tournaments_won(
id int primary key not null,
date datetime,
trophy_name varchar(255),
quantity int
);


create table tournaments_played(
id int primary key not null,
tournament_name varchar,
date datetime,
tournaments_won int
);


create table matches_played(
id int primary key not null,
matches_drawn int,
matches_won int,
matches_lost int,
score int,
tournaments_played int
);
```

Also, we have used the following queries:

1. **Update Query:** UPDATE table_name SET column1 = value1, column2 = value2, WHERE condition;
2. **Alter Query**: ALTER TABLE table_name ADD column_name datatype;

Conclusion & Lessons learned

Around 5 data sets are added to each table and more data can be added to the database as per the user's requirements. This database can be implemented in docker, but it requires WSL (Windows Subsystem for Linux) and docker desktop to run, which makes its usage heavy.

AWS can be used instead but it can be expensive if there's a budget limit for the user. Hence, we used azure as it comes with some credits which are free for college students, also it works well with the Microsoft environment security is good and setup steps are easy as most of the parts are pre-installed or easily available. We have designed the database in which the tables are in second normal form(2NF).

Further scope

We can increase normalization further (as we have used 2NF). Apart from this, we can use indexing to increase query performance. Apart from this Data Defragmentation can be used to improve database performance.

Also, CPU speeds and memory can be increased or decreased, depending on the customization made on the virtual machine depending upon the clients/user's budget.

References

- Fundamentals of Database Systems, 7/e Elmasri, Navathe, 2017, Pearson, ISBN-10:1292097612 and ISBN-13: 9781292097619.
- https://www.w3schools.com/sql/