BDP - HomeWork 3 - Code and Output - Shivani Bhoite

In [4]:
```
import sqlite3
import pandas as pd
```

In [5]:
```
cnx = sqlite3.connect(r'/Users/shivanibhoite/Desktop/database.sqlite')

country = pd.read_sql_query("SELECT * FROM Country", cnx)
league = pd.read_sql_query("SELECT * FROM League", cnx)
match = pd.read_sql_query("SELECT * FROM Match", cnx)
player = pd.read_sql_query("SELECT * FROM Player", cnx)
player_attributes = pd.read_sql_query("SELECT * FROM Player_attributes",
team = pd.read_sql_query("SELECT * FROM Team", cnx)
team_attributes = pd.read_sql_query("SELECT * FROM Team_attributes", cnx
```

Question 1 (20 points): Write a SQL query that lists all the players born between 1987 and 1990 inclusive, sort them from the oldest to the youngest. The output of this query should be of the form:

Player Name | Birthday

In [ ]:
```
q1 = pd.read_sql_query("""

SELECT player_name as "Player Name", birthday as Birthday
FROM player WHERE birthday between '1987' and '1991'
ORDER BY birthday;"""

                       , cnx)

q1
```

Question 2 (20 points): Write a SQL query that ranks all countries and leagues based on the total amount of total goals scored per game in the whole dataset. Sort them by the largest to the smallest amount of goals. Note: Read this carefully. The output of this query should be of the form:

Country | League Name |Total Goals Scored

In [ ]:
```
q2 = pd.read_sql_query("""

SELECT country.name as Country, league.name as 'League Name',
sum(match.home_team_goal+match.away_team_goal) as 'Total Goals Scored'
from country inner join league on country.id = league.country_id
join match on league.id = match.league_id group by league.name
order by sum(match.home_team_goal+match.away_team_goal) desc ;""",cnx)

q2
```

Question 3 (20 points): Write a SQL query that ranks all teams by the average of all their attributes (not the players' attributes), sort them from best to worst. The output of this query should be of the form:

Team Long Name | Average of Attributes

In [ ]:
```
q3 = pd.read_sql_query("""
SELECT Team.team_long_name as "Team Long Name",
((AVG(Team_Attributes.buildUpPlaySpeed)+ AVG(Team_Attributes.buildUpPlay
AVG(Team_Attributes.buildUpPlayPassing) + AVG(Team_Attributes.chanceCrea
AVG(Team_Attributes.chanceCreationCrossing)+AVG(Team_Attributes.chanceCr
AVG(Team_Attributes.defencePressure)+AVG(Team_Attributes.defenceAggressi
AVG(Team_Attributes.defenceTeamWidth))/9)
as Average_of_Attributes
FROM Team join Team_Attributes on Team.team_api_id=Team_Attributes.team_
group by team.team_api_id order by
Average_of_Attributes desc""",cnx)

q3
```

Question 4 (20 points): Write a SQL query that ranks all teams by the average of their players' attributes, sort them by descending order displaying only the top 5. The output of this query should be of the form:

Team Name| Number of Players | Player Attribute Average

Cannot find the number of players from this database as there is no primary key and foreign key available between the team table and player table to do so

In [ ]:
```
q4 = pd.read_sql_query("""
    SELECT Team.team_long_name as 'Team Name',
     'Number of Player',
((AVG(Team_Attributes.buildUpPlaySpeed)+ AVG(Team_Attributes.buildUpPlay
AVG(Team_Attributes.buildUpPlayPassing) + AVG(Team_Attributes.chanceCrea
AVG(Team_Attributes.chanceCreationCrossing)+AVG(Team_Attributes.chanceCr
AVG(Team_Attributes.defencePressure)+AVG(Team_Attributes.defenceAggressi
AVG(Team_Attributes.defenceTeamWidth))/9)
as 'Player Attribute Average'
FROM Team join Team_Attributes on Team.team_api_id=Team_Attributes.team_
group by team.team_api_id order by
'Player Attribute Average' desc limit 5
;""",cnx)

q4.head()
```

Question 5 (40 points): Write a SINGLE SQL query that finds the date that had the most goals scored on, per each different season and league. The output of this query should be of the form:

Date (dd/mm/yy) | Season | League Name | Goals scored

In [ ]:
```
q5 = pd.read_sql_query("""

Select Date, Season, League_Name, max(goal) as Goals_Scored from
(
SELECT match.date as 'Date', match.season as 'Season',

    league.name as League_Name,   sum(match.home_team_goal+match.away_te

from league join match on league.id = match.league_id

group by match.date,league.name,match.season

 order by goal desc)

 group by League_Name, Season order by Goals_Scored desc ;

 """,cnx)

q5
```

Graduate Student Task (40 points): Write a SINGLE SQL query that finds the top 5 teams in terms of goals scored PER league for the 2008/2009 season. The output of this query should be of the form:

Season | League | Rank | Team Name | Goals Scored

In [ ]:
```
q6 = pd.read_sql_query("""
Select Season, League,
        Rank() over (order by League) as Rank, TeamName , Sum(Goals) as
                (SELECT match.season as Season,league.name as League,
                team.team_long_name as TeamName,
                sum(match.home_team_goal) as Goals
                from country join league on country.id = league.country_
                join match on league.id = match.league_id
                join team on team.team_api_id = match.home_team_api_id
                where season ='2008/2009'
                group by league.id,season
                        union all
               SELECT match.season as Season,league.name as League,
                team.team_long_name as TeamName,
                sum(match.away_team_goal) as Goals
               from country join league on country.id = league.country_id
               join match on league.id = match.league_id
               join team on team.team_api_id = match.away_team_api_id
               where season ='2008/2009'
              group by league.id,Season)

    group by league, TeamName
    order by league , GoalsScored desc  ;""",cnx)
q6
```

In [ ]:
```

```