Data Management Group Project

By: Jelmer Spreen and Geetanshu Tehlan

Submitted to: Prof. Yi-Ting Chen

1A: Find the average defending rating of players in each club and display the result based on the average defending rating, ranked from highest to lowest.

The first query displays all the football players who **have** a defending rating, in this query SQL **does not** count goalkeepers. Therefore the average is different.

Query:

```
1 select club_name, avg(player.defending) as average_defending
2 FROM Player
3 join Club on player.club_team_id = Club.cid
4 group by club_name
5 order by average_defending DESC
```

Results:

club_name	average_defending
Paris Saint-Germain	66.0689655172414
Juventus	65.8695652173913
Inter	65.8333333333333
Atalanta	64.7727272727273
Liverpool	63.9642857142857
Sporting CP	63.3333333333333
Roma	63.2758620689655
Clube Atlético Mineiro	62.6111111111111
Chelsea	62.2068965517241
AS Monaco	61.875
SL Benfica	61.8461538461539
Athletic Club de Bilbao	61.7777777777778
Leicester City	61.6206896551724
Atlético de Madrid	61.5357142857143
Borussia Dortmund	61.4642857142857
Granada CF	61.448275862069
Real Valladolid CF	61.3703703703704
Villarreal CF	61.3666666666667
Lazio	61.0
TSG Hoffenheim	60.7241379310345
Napoli	60.5833333333333
Fiorentina	60.5833333333333
Genoa	60.5714285714286
AC Milan	60.56
West Ham United	60.1724137931034
Fortaleza	60.0555555555556
Real Sociedad	59.7826086956522
Sevilla FC	59.551724137931
Manchester United	59.3448275862069
Cagliari	59.2692307692308
RB Leipzig	59.2413793103448
Olympique Lyonnais	59.2222222222222
AEK Athens	59.1923076923077
Tottenham Hotspur	59.1290322580645
U.S. Sassuolo Calcio	59.125
İstanbul Başakşehir FK	59.0833333333333
Bayer 04 Leverkusen	59.0416666666667
Levante Unión Deportiva	59.0333333333333
Sport Club Corinthians Paulista	59.0
PFC CSKA Moscow	59.0
FC Bayern München	59.0
Club América	58.9565217391304
CA Osasuna	58.9333333333333
CF Monterrey	58.8333333333333
Arsenal	58.8
Manchester City	58.7586206896552
U.C. Sampdoria	58.7272727272727
Burnley	58.7
River Plate	58.68
FC Porto	58.6521739130435
FC Lorient	58.5454545454545
Newcastle United	58.4642857142857
Club Brugge KV	58.2916666666667
Cádiz CF	58.1379310344828
Everton	58.1333333333333
Club Atlético Independiente	58.0833333333333
FC Red Bull Salzburg	57.84
Hellas Verona	57.7916666666667
LOSC Lille	57.7692307692308
Spartak Moskva	57.7391304347826
SK Slavia Praha	57.72
Olympiacos CFP	57.72
Shandong Taishan	57.6
Club Athletico Paranaense	57.5555555555556
Getafe CF	57.5185185185185
VfL Wolfsburg	57.4642857142857
FC Barcelona	57.448275862069
Crystal Palace	57.3448275862069
AC Monza	57.32
Rangers FC	57.28
RCD Espanyol de Barcelona	57.2666666666667
Real Zaragoza	57.2592592592593
Torino F.C.	57.2068965517241
Borussia Mönchengladbach	57.1724137931034
OGC Nice	57.1666666666667
Fulham	57.1111111111111
Brentford	56.9677419354839
Eintracht Frankfurt	56.9285714285714
US Salernitana 1919	56.9166666666667
RC Strasbourg Alsace	56.9090909090909
Ajax	56.7916666666667
Club Atlético Colón	56.72
RB Bragantino	56.6111111111111
SC Braga	56.5925925925926
Hertha BSC	56.5925925925926
Cruz Azul	56.5416666666667
Flamengo	56.4444444444444
Valencia CF	56.4137931034483
Sheffield United	56.32
Real Madrid CF	56.2333333333333
Udinese Calcio	56.2083333333333
Olympique de Marseille	56.1724137931035

In this query, SQL counts all the players including the goalkeeper by assign them a value zero.

Query:

```
1 SELECT club_name, AVG(COALESCE(player.defending, 0)) as average_defending
2 FROM Player
3 JOIN Club ON player.club_team_id = Club.cid
4 GROUP BY club_name
5 ORDER BY average_defending DESC;
```

Now the results will be different as now the Goalkeepers will be counted and their value will be counted as well. Therefore the average will be much different if a team has more goalkeepers.

Results:

club_name	average_defending
Juventus	58.2692307692308
Paris Saint-Germain	58.0606060606061
Roma	57.34375
Atalanta	57.0
Inter	56.4285714285714
Clube Atlético Mineiro	56.35
Napoli	55.9230769230769
Villarreal CF	55.7878787878788
Athletic Club de Bilbao	55.6
Tottenham Hotspur	55.5454545454546
Borussia Dortmund	55.5161290322581
SL Benfica	55.448275862069
Lazio	55.28125
Real Valladolid CF	55.2333333333333
Real Sociedad	55.0
AS Monaco	55.0
AEK Athens	54.9642857142857
Chelsea	54.6666666666667
Liverpool	54.2727272727273
Leicester City	54.1515151515152
AC Milan	54.0714285714286
Fortaleza	54.05
Granada CF	54.0
Fiorentina	53.8518518518519
Atlético de Madrid	53.84375
RB Leipzig	53.6875
Levante Unión Deportiva	53.6666666666667
CA Osasuna	53.5757575757576
Torino F.C.	53.5161290322581
Brentford	53.5151515151515
Arsenal	53.4545454545455
TSG Hoffenheim	53.3636363636364
Burnley	53.3636363636364
Olympique Lyonnais	53.3
Manchester City	53.25
Sporting CP	53.2
Cagliari	53.1379310344828
Sport Club Corinthians Paulista	53.1
West Ham United	52.8787878787879
Everton	52.8484848484849
Cádiz CF	52.6875
FC Bayern München	52.6785714285714
U.S. Sassuolo Calcio	52.5555555555556
İstanbul Başakşehir FK	52.5185185185185
Bayer 04 Leverkusen	52.4814814814815
River Plate	52.3928571428572
Sevilla FC	52.3333333333333
CF Monterrey	52.2962962962963
PFC CSKA Moscow	52.1923076923077
Club América	52.1538461538462
Manchester United	52.1515151515152
Empoli	52.1071428571429
RCD Espanyol de Barcelona	52.0606060606061
Udinese Calcio	51.8846153846154
FC Porto	51.8846153846154
Club Athletico Paranaense	51.8
Getafe CF	51.7666666666667
PAOK	51.75
U.C. Sampdoria	51.68
FC Red Bull Salzburg	51.6428571428571
Wolverhampton Wanderers	51.6363636363636
Club Atlético Independiente	51.6296296296296
SK Slavia Praha	51.5357142857143
Olympiacos CFP	51.5357142857143
Real Zaragoza	51.5333333333333
Shandong Taishan	51.4285714285714
Eintracht Frankfurt	51.4193548387097
Fulham	51.4
Genoa	51.3939393939394
Hellas Verona	51.3703703703704
FC Arouca	51.2857142857143
AC Monza	51.1785714285714
Rangers FC	51.1428571428572
Real Madrid CF	51.1212121212121
Boca Juniors	51.0714285714286
Brighton & Hove Albion	51.0606060606061
RB Bragantino	50.95
SC Braga	50.9333333333333
RC Celta de Vigo	50.8787878787879
Real Betis Balompié	50.8181818181818
OGC Nice	50.8148148148148
Flamengo	50.8
Club Atlético Colón	50.6428571428572
US Salernitana 1919	50.5925925925926
Nottingham Forest	50.5
FC Barcelona	50.4848484848485
Tigres U.A.N.L.	50.48
Crystal Palace	50.3939393939394
Demir Grup Sivasspor	50.3333333333333
Palmeiras	50.3
Ceará Sporting Club	50.3
VfL Wolfsburg	50.28125

1B: Find the names of players, their clubs, and their attacking ratings for players who have an attacking rating greater than 80.

Query:

```
1 select Club.club_name, Player.long_name as Player_Name,
2 sum(Attacking.crossing + Attacking.finishing + Attacking.heading_accuracy +
  Attacking.short_passing + Attacking.volleys)/5 as Attack_Total_score
3 from player
4 join Club on Player.club_team_id = Club.cid
5 join Attacking on Player.pid = Attacking.aid
6 group by Player.pid
7 having Attack_Total_score >= 80
8 order by Attack_Total_score DESC
```

Results:

club_name	Player_Name	Attack_Total_score
Manchester United	Cristiano Ronaldo dos Santos Aveiro	87
Atlético de Madrid	Luis Alberto Suárez Díaz	86
FC Bayern München	Robert Lewandowski	86
Tottenham Hotspur	Harry Kane	86
Paris Saint-Germain	Lionel Andrés Messi Cuccittini	85
Real Madrid CF	Karim Benzema	85
FC Bayern München	Thomas Müller	84
Atlético de Madrid	Antoine Griezmann	84
Real Madrid CF	Gareth Frank Bale	83
Villarreal CF	Gerard Moreno Balagueró	82
Manchester United	Bruno Miguel Borges Fernandes	82
Paris Saint-Germain	Kylian Mbappé Lottin	82
AC Milan	Zlatan Ibrahimović	81
FC Barcelona	Sergio Leonel Agüero del Castillo	81
Athletic Club de Bilbao	Raúl García Escudero	81
Manchester United	Edinson Roberto Cavani Gómez	81
Chelsea	Romelu Lukaku Menama	81
RC Celta de Vigo	Iago Aspas Juncal	81
Manchester City	Kevin De Bruyne	81
Everton	James David Rodríguez Rubio	81
AS Monaco	Kevin Volland	81
Wolverhampton Wanderers	Raúl Alonso Jiménez Rodríguez	81
Liverpool	Sadio Mané	81
FC Barcelona	Luuk de Jong	80
Paris Saint-Germain	Neymar da Silva Santos Júnior	80
Ajax	Dušan Tadić	80
AS Monaco	Wissam Ben Yedder	80
Tottenham Hotspur	손흥민 孙兴慜	80
Juventus	Álvaro Borja Morata Martín	80
Sevilla FC	Lucas Ariel Ocampos	80
Stade Rennais FC	Gaëtan Laborde	80
Juventus	Paulo Bruno Exequiel Dybala	80
FC Bayern München	Leroy Aziz Sané	80

1C: Find clubs (i.e., the id and the name of the club) where the average shooting rating is higher than or equal to the average shooting rating of 'AC Milan'.

Query:

```
SELECT Club.cid as Club_ID, Club.club_name, AVG(player.shooting) as Average_Shooting
FROM Club
JOIN player ON Club.cid = player.club_team_id
GROUP BY Club.club_name
HAVING AVG(player.shooting) >= (
    SELECT AVG(player.shooting)
    FROM club
    JOIN player ON Club.cid = Player.club_team_id
    WHERE Club.club_name = 'AC Milan'
)
```

Results:

Club_ID	club_name	Average_Shooting
47	AC Milan	61.88
245	Ajax	64.4583333333333
39	Atalanta	63.0454545454546
240	Atlético de Madrid	63.25
21	FC Bayern München	65.28
236	FC Porto	62.9130434782609
44	Inter	64.9166666666667
45	Juventus	66.1739130434783
46	Lazio	61.9310344827586
10	Manchester City	62.9655172413793
11	Manchester United	66.0
48	Napoli	62.0
247	PSV	63.8
73	Paris Saint-Germain	63.9310344827586
449	Real Betis Balompié	63.0666666666667
243	Real Madrid CF	66.5
234	SL Benfica	65.2692307692308
481	Sevilla FC	63.3103448275862

2A: . Grapes are the raw material for making red wine. Which country has the most varieties of grapes, please report the name of the country and the number of types of grapes

Query:

```
SELECT Winery.country as Country_Name, COUNT(DISTINCT Wine.gid) as N_Type_Grapes
FROM Wine
JOIN Grape ON Wine.gid = Grape.id
JOIN Winery ON Wine.wid = Winery.id
GROUP BY Winery.country
ORDER BY N_Type_Grapes DESC
LIMIT 1
```

Results:

	Country_Name	N_Type_Grapes
1	US	294

2B: According to the guide published by wine enthusiasts, red wine can be divided into four classes based on its price: everyday wine, mid-range wine, fine wine, and super fine wine. If the price of red wine is higher than or equal to $340, it will be classified as super fine wine. Please report the average price and points of the super fine wines

Query:

```
SELECT AVG(Wine.Price) as Average_Price, AVG(Wine.points) as Average_Points
FROM Wine
WHERE Wine.price >= 340
```

Results:

Average_Price	Average_Points
629.189602446483	95.3975535168196

2C: A rater evaluates the score and price of the wine fairly and unbiasedly based on the quality of the wine. Please write SQL code and python code to analyse the data statistics of points and prices of wine evaluated by each taster and use the format rater_name.xlsx to export the data statistics

First we wrote a SQL code in DBbrowser to create a table that contains raters with all their points and prices.

Query:

```sql
SELECT Rater.name, wine.points, wine.price
FROM Rater
JOIN Wine on Rater.id = Wine.rid
```

Then we created a python code which executes the query and then transforms the table, using the pandas groupby function combined with the describe function. Now the table will consists of descriptive statists per individual rater Then we created a for-loop, which fetches all the raters name from the transformed table and returns an excel file per rater. These excel files can be found back in the Report Zip file under the map name 'Excelfiles Rater Names'.

Python code:

```python
### Importing the file
import pandas as pd
import sqlite3
import os

root = os.getcwd()
dbname = os.path.join(root,'Wine.db')

### Creating a SQL Query for a new dataset that only contains names, scores and price
query_sql = 'SELECT Rater.name, wine.points, wine.price FROM Rater JOIN Wine on Rater.id = Wine.rid'
try:
    conn = sqlite3.connect(dbname)
    df = pd.read_sql_query(query_sql, conn)

except sqlite3.Error as error:
    print('The error message:', error)
finally:
    if conn:
        conn.close()
        print('The connection has been closed')
### Grouping the data per name and analyse per person
grouped_data = df.groupby('name')[['price', 'points']].describe()

### Converting every person analysis to an excel file
for name, group_data in grouped_data.iterrows():
    filename = f"{name}.xlsx"
    group_data.to_excel(filename)
    print(f"Descriptive stats for {name} saved to {filename}")
```

This Python code can be found back in the Report Zip file under the name: 'SQL Project Jelmer & Geetanshu.ipynb'