European Soccer Database

Covered topics: Databases & SQL

Assignment Instructions

You will be working with the European Soccer Database, a collection of four individual CSV files that you will find in the *European Soccer Database.zip* compressed folder, containing:

- leagues.csv
- match.csv
- player.csv
- match.csv

Make a copy of this Google Doc and:

- Paste the SQL query that generates the solution right below the question;
- Write the answer to the question (when possible) in the following table.

Question #	Answer
1	Not Required
2	Link to lucidchart: link
3	2,868
4	2009/2010 - England Premier League
5	There are 8 unique seasons. "2013/2014 Belgium Jupiler League only has 12 home goals in 2013/2014
6	10,197
7	863
8	FC Barcelona, 112
9	Real Madrid CF (4 times)
10	45

Data Analysis with SQL

Using this database, complete the following tasks:

1. Create a new data set called "Final_Exercise" in Google BigQuery and load each csv file as a separate table.
2. Using https://lucid.app/, create a schema that represents the relationship between all the tables:
    a. For each table, write to the left of the variable's name if it is a primary key (PK), a foreign key (FK) or just a simple variable (leave blank).
    b. For each table, write its shape (write the number of rows and columns near the table name).
    c. With a line, link the tables to each other through their keys (when possible).

    Answer: lucidchart ER Diagram link
3. How many days have passed from the oldest **Match** to the most recent one (dataset time interval)?

    Answer: 2868

```
SELECT DATE_DIFF(max(date), min(date), day) AS DateDiff
FROM `project-prova-384317.Final_Exercise.match`;
```

Row	DateDiff
1	2868

4. Produce a table which, for each Season and **League** Name, shows the following statistics about the home goals scored:
    a. min
    b. average
    c. mid-range
    d. max
    e. sum

    Hint: there is no function for the mid-range, research it and calculate it.

    Which combination of Season-League has the highest number of goals?

    Answer: 2009/2010 - England Premier League

```
SELECT a.season, b.name as league_name,
 min(a.home_team_goal) as min_home_team_goal,
 avg(a.home_team_goal) as avg_home_team_goal,
 (max(a.home_team_goal) + min(a.home_team_goal))/2 as midrange_home_team_goal,
 max(a.home_team_goal) as max_home_team_goal,
 sum(a.home_team_goal) as sum_home_team_goal
FROM `project-prova-384317.Final_Exercise.match` a
LEFT JOIN `project-prova-384317.Final_Exercise.leagues` b
```

```
on a.league_id = b.id
GROUP BY a.season, b.name
ORDER BY sum_home_team_goal desc
```

Row	season	league_name	min_home_team_goal	avg_home_team_goal	midrange_home_team_goal	max_home_team_goal	sum_home_team_goal
1	2009/2010	England Premier League	0	1.6973684210526321	4.5	9	645
2	2012/2013	Spain LIGA BBVA	0	1.6868421052631584	3.0	6	641
3	2011/2012	Spain LIGA BBVA	0	1.6789473684210523	4.0	8	638
4	2008/2009	Spain LIGA BBVA	0	1.660526315789473	3.5	7	631
5	2010/2011	Spain LIGA BBVA	0	1.6368421052631583	4.0	8	622
6	2013/2014	Spain LIGA BBVA	0	1.6315789473684206	3.5	7	620
7	2010/2011	England Premier League	0	1.6236842105263158	3.5	7	617
8	2015/2016	Spain LIGA BBVA	0	1.6184210526315788	5.0	10	615
9	2009/2010	Spain LIGA BBVA	0	1.5999999999999994	3.0	6	608
10	2011/2012	England Premier League	0	1.5894736842105257	4.0	8	604

5. Find out how many unique seasons there are in the **Match** table.
   Then write a query that shows, for each Season, the number of matches played by each League. Do you notice anything out of the ordinary?
   Answer: Belgium Jupiler League only has 12 home goals in 2013/2014

```
SELECT distinct season
FROM `project-prova-384317.Final_Exercise.match`
```

Row	season
1	2008/2009
2	2009/2010
3	2010/2011
4	2011/2012
5	2012/2013
6	2013/2014
7	2014/2015
8	2015/2016

```
SELECT a.season, b.name as league_name,
count(a.id) as Matches_played
FROM `project-prova-384317.Final_Exercise.match` a
LEFT JOIN `project-prova-384317.Final_Exercise.leagues` b
on a.league_id = b.id
GROUP BY a.season, b.name
```

Row	season	league_name	min_home_team_goal
1	2008/2009	Belgium Jupiler League	306
2	2009/2010	Belgium Jupiler League	210
3	2010/2011	Belgium Jupiler League	240
4	2011/2012	Belgium Jupiler League	240
5	2012/2013	Belgium Jupiler League	240
6	2013/2014	Belgium Jupiler League	12
7	2014/2015	Belgium Jupiler League	240
8	2015/2016	Belgium Jupiler League	240
9	2008/2009	England Premier League	380
10	2009/2010	England Premier League	380
11	2010/2011	England Premier League	380

6. Using Players as the starting point, create a new table (PlayerBMI) and add:
    a. a new variable that represents the players' weight in kg (divide the mass value by 2.205) and call it kg_weight;
    b. a variable that represents the height in metres (divide the cm value by 100) and call it m_height;
    c. a variable that shows the body mass index (BMI) of the player;
    Hint: research how to calculate the formula of the BMI
    d. Filter the table to show only the players with an optimal BMI (from 18.5 to 24.9).

    How many rows does this table have?

    Answer: 10,197

```
CREATE TABLE Final_Exercise.PlayerBMI AS
SELECT *,
 weight/2.205 as kg_weight,
 height/100 as m_height,
 (weight/2.205)/power(height/100, 2) as BMI
FROM `project-prova-384317.Final_Exercise.player`
WHERE (weight/2.205)/power(height/100, 2) between 18.5 and 24.9
```

Row	id	player_api_id	player_name	birthday	height	weight	kg_weight	m_height	BMI
1	2868	451335	Edmilson Fernandes	1996-04-15 00:00:00 UTC	190.5	154	69.841269841269835	1.905	19.245188402193381
2	2901	144989	Efe Ambrose	1988-10-18 00:00:00 UTC	190.5	154	69.841269841269835	1.905	19.245188402193381
3	8840	575789	Reece Oxford	1998-12-16 00:00:00 UTC	190.5	157	71.201814058956913	1.905	19.620094669768577
4	18	23499	Aaron Wilbraham	1979-10-21 00:00:00 UTC	190.5	159	72.10884353741497	1.905	19.870032181485378
5	4928	523414	Jhonder Cadiz	1995-07-29 00:00:00 UTC	190.5	161	73.015873015873012	1.905	20.119969693202172
6	7535	213167	Mike Te Wierik	1992-06-08 00:00:00 UTC	190.5	161	73.015873015873012	1.905	20.119969693202172
7	761	24336	Andrew O'Brien	1979-06-29 00:00:00 UTC	190.5	163	73.922902494331069	1.905	20.36990720491897
8	1489	201733	Callum Ball	1992-10-08 00:00:00 UTC	190.5	163	73.922902494331069	1.905	20.36990720491897
9	1716	34421	Chris Kirkland	1981-05-02 00:00:00 UTC	190.5	163	73.922902494331069	1.905	20.36990720491897

7. How many players do not have an optimal BMI?

    Answer: 863

```
SELECT
 (SELECT count(id)
 FROM `project-prova-384317.Final_Exercise.player`) -
 (SELECT count(id)
 FROM `project-prova-384317.Final_Exercise.PlayerBMI') as PlayerNoBMI
```

    Or, alternatively:

```
SELECT count(*) as PlayerNoBMI
FROM `project-prova-384317.Final_Exercise.player` a
LEFT JOIN `project-prova-384317.Final_Exercise.PlayerBMI` b
 on a.id = b.id
WHERE b.id is null
```

Row	PlayerNoBMI
1	863

8. Which **Team** has scored the highest total number of goals (home + away) during the most recent available season? How many goals has it scored?

   Answer: FC Barcelona, 112

```
select h.team_long_name, h.SumOfGoalHome, a.SumOfGoalAway, h.SumOfGoalHome +
a.SumOfGoalAway as TotalGoal
from
 (select t.team_long_name, sum(m.home_team_goal) as SumOfGoalHome
 from `project-prova-384317.Final_Exercise.match` m
 inner join project-prova-384317.Final_Exercise.team` t
   on m.home_team_api_id = t.team_api_id
 where m.season = (select max(season) from
`project-prova-384317.Final_Exercise.match`)
 group by t.team_long_name order by SumOfGoalHome) h
inner join
 (select t.team_long_name, sum(m.away_team_goal) as SumOfGoalAway
 from `project-prova-384317.Final_Exercise.match` m
 inner join `project-prova-384317.Final_Exercise.team` t
   on m.away_team_api_id = t.team_api_id
 where m.season = (select max(season) from
`project-prova-384317.Final_Exercise.match`)
 group by t.team_long_name order by SumOfGoalAway) a
   on h.team_long_name = a.team_long_name
order by TotalGoal desc
limit 1
```

Or, alternatively:

```
select b.team_long_name, sum(a.goals) as goals
FROM
 (SELECT 'home' as game_loc, home_team_api_id as team_api_id, sum(home_team_goal)
as goals
 FROM `project-prova-384317.Final_Exercise.match`
 WHERE season = '2015/2016'
 GROUP BY home_team_api_id
 UNION ALL
 SELECT 'away' as game_loc, away_team_api_id as team_api_id, sum(away_team_goal)
as goals
 FROM `project-prova-384317.Final_Exercise.match`
 WHERE season = '2015/2016'
 GROUP BY away_team_api_id
```

```
 ORDER BY goals desc) a
LEFT JOIN `project-prova-384317.Final_Exercise.team` b
  on a.team_api_id = b.team_api_id
GROUP BY b.team_long_name
ORDER BY goals desc
LIMIT 1
```

Row	team_long_name	SumOfGoalHome	SumOfGoalAway	TotalGoal
1	FC Barcelona	67	45	112

9. Create a query that, for each season, shows the name of the team that ranks first in terms of total goals scored (the output table should have as many rows as the number of seasons).
   Which team was the one that ranked first in most of the seasons?

   Answer: FC Real_madrid

```
select *
from
(select h.season, h.team_long_name, h.SumOfGoalHome, a.SumOfGoalAway,
h.SumOfGoalHome + a.SumOfGoalAway as TotalGoal,
rank() over (partition by a.season order by h.SumOfGoalHome +
a.SumOfGoalAway desc) as rank_season
from
(select m.season, t.team_long_name, sum(m.home_team_goal) as SumOfGoalHome
from `project-prova-384317.Final_Exercise.match` m inner join
`project-prova-384317.Final_Exercise.team` t on m.home_team_api_id =
t.team_api_id
group by m.season, t.team_long_name order by SumOfGoalHome) h
inner join
(select m.season,t.team_long_name, sum(m.away_team_goal) as SumOfGoalAway
from `project-prova-384317.Final_Exercise.match` m inner join
`project-prova-384317.Final_Exercise.team` t on m.away_team_api_id =
t.team_api_id
group by m.season, t.team_long_name order by SumOfGoalAway) a
on h.team_long_name = a.team_long_name and h.season=a.season)
where rank_season = 1
order by season desc
```

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	season	team_long_name	SumOfGoalHome	SumOfGoalAway	TotalGoal	rank_season
1	2015/2016	FC Barcelona	67	45	112	1
2	2014/2015	Real Madrid CF	65	53	118	1
3	2013/2014	Real Madrid CF	63	41	104	1
4	2012/2013	FC Barcelona	63	52	115	1
5	2011/2012	Real Madrid CF	70	51	121	1
6	2010/2011	Real Madrid CF	61	41	102	1
7	2009/2010	Ajax	64	42	106	1
8	2008/2009	FC Barcelona	61	44	105	1

10. From the query above (question 8) create a new table (TopScorer) containing the top 10 teams in terms of total goals scored (*hint: add the team id as well*).
    Then write a query that shows all the possible "pair combinations" between those 10 teams. How many "pair combinations" did it generate?

    Answer: 45

```
create table `project-prova-384317.Final_Exercise.TopScorer` as
(select h.team_api_id ,h.team_long_name, h.SumOfGoalHome, a.SumOfGoalAway,
h.SumOfGoalHome + a.SumOfGoalAway as TotalGoal
from
(select t.team_api_id ,t.team_long_name, sum(m.home_team_goal) as SumOfGoalHome
from `project-prova-384317.Final_Exercise.match` m inner join
`project-prova-384317.Final_Exercise.team` t on m.home_team_api_id =
t.team_api_id
where m.season = (select max(season) from
`project-prova-384317.Final_Exercise.match`)
group by t.team_api_id, t.team_long_name order by SumOfGoalHome) h inner join
(select t.team_long_name, sum(m.away_team_goal) as SumOfGoalAway
from `project-prova-384317.Final_Exercise.match` m inner join
`project-prova-384317.Final_Exercise.team` t on m.away_team_api_id =
t.team_api_id
where m.season = "2015/2016"
group by t.team_long_name order by SumOfGoalAway) a on h.team_long_name =
a.team_long_name
order by TotalGoal desc
limit 10)
SELECT a.team_long_name, b.team_long_name
FROM `project-prova-384317.Final_Exercise.TopScorer` a
LEFT JOIN `project-prova-384317.Final_Exercise.TopScorer' b
on a.team_api_id > b.team_api_id
where a.team_long_name is not null and  b.team_long_name is not null
```

Or, alternatively:

```
CREATE TABLE Final_Exercise.TopScorer AS
select b.team_api_id, b.team_long_name, sum(a.goals) as goals
FROM
 (SELECT 'home' as game_loc, home_team_api_id as team_api_id, sum(home_team_goal)
as goals
 FROM `project-prova-384317.Final_Exercise.match'
 WHERE season = '2015/2016'
 GROUP BY home_team_api_id
 UNION ALL
 SELECT 'away' as game_loc, away_team_api_id as team_api_id, sum(away_team_goal)
as goals
 FROM `project-prova-384317.Final_Exercise.team`
 WHERE season = '2015/2016'
 GROUP BY away_team_api_id
 ORDER BY goals desc) a
LEFT JOIN `project-prova-384317.Final_Exercise.team` b
  on a.team_api_id = b.team_api_id
GROUP BY b.team_api_id, b.team_long_name
ORDER BY goals desc
LIMIT 10
```

```
SELECT a.team_long_name, b.team_long_name
FROM `project-prova-384317.Final_Exercise.TopScorer` a
INNER JOIN `project-prova-384317.Final_Exercise.TopScorer` b
 on a.team_api_id > b.team_api_id
```

Row	team_long_name	team_long_name_1
1	Real Madrid CF	Ajax
2	PSV	Ajax
3	PSV	Real Madrid CF
4	PSV	FC Barcelona
5	Celtic	Ajax
6	Celtic	Paris Saint-Germain
7	Celtic	Real Madrid CF
8	Celtic	Borussia Dortmund