2.9.x Final Assignment - Virgillito A. L.

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 *2.9.x European Soccer Database.zip* compressed folder, containing:

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

Make a copy of this Google Doc and, for each of the tasks that you'll find in the next page:

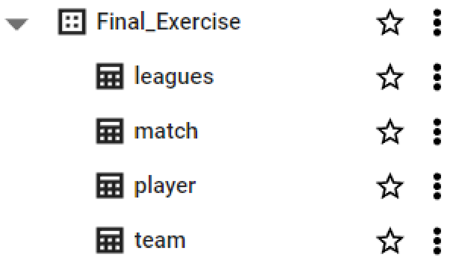
- Paste the SQL query that generates the solution right below the question;
- Write the answer to the question (when possible) in the following table.

Question #	Answer
1	Not Required
2	**Lucidchart_schema**
3	**2010** -> Look under the question
4	**England Premier League with 645 goal** ->Look under the question
5	**Belgium Jupiler League** ->Look under the question
6	**10637** ->Look under the question
7	**337** ->Look under the question
8	**FC Barcelona with 112 tot_goal** ->Look under the question
9	**Real Madrid with 121 goal** ->Look under the question
10	**Look under the question**

Data Analysis with SQL

Using the abovementioned 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.



- Final_Exercise
  - leagues
  - match
  - player
  - team

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).

3. How many days have passed from the oldest **Match** to the most recent one (dataset time interval)?

```
SELECT
    MIN (extract(year from date)) as oldest_match,
    MAX (extract(year from date)) as most_rec_match,
   round ((MAX (extract(year from date)) - MIN (extract(year from
date))/365)) as Interval_days_passed
 FROM `adept-bond-365418.Final_Exercise.match`
```

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?

```
SELECT match.season,
       leagues.name,
      min(match.home_team_goal) as min_of_goal_at_home,
      max(match.home_team_goal) as max_goal_at_home,
      sum(match.home_team_goal) as sum_of_goal_at_home,
      round(avg(match.home_team_goal),1) as avg_for_goal_at_home,
      (max(match.home_team_goal)+ min(match.home_team_goal))/2 as
mid_range_at_home,
 FROM `adept-bond-365418.Final_Exercise.match` as match
 Left join `adept-bond-365418.Final_Exercise.leagues` as leagues
 on match.league_id = leagues.id
 group by match.season, leagues.name
 order by sum_of_goal_at_home DESC
```

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?

```
SELECT distinct match.season,
       leagues.name,
       count(match.stage) as match_for_leagues
 FROM `adept-bond-365418.Final_Exercise.match` as match
 Left join `adept-bond-365418.Final_Exercise.leagues` as leagues
 on match.league_id = leagues.id
 group by match.season, leagues.name
 order by match_for_leagues
```

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?

```
CREATE TABLE Final_Exercise.PlayerBMI as
SELECT *,
FROM  (SELECT distinct player_name ,
       round (weight / 2.205, 2) as Kg_weight,
       round (height / 100 , 2)  as M_height,
       round (((weight / 2.205) / (height / 100))/ 2) as BMI
  FROM `adept-bond-365418.Final_Exercise.player`
  where  round (((weight / 2.205) / (height / 100))/ 2) >= 18.5 and  round
(((weight / 2.205) / (height / 100))/ 2) <= 24.9
  Group by player_name ,weight , height
  ORDER BY BMI Desc
  )
```

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

```
SELECT *,
FROM ( SELECT distinct player_name ,
       round (weight / 2.205, 2) as Kg_weight,
       round (height / 100 , 2)  as M_height,
      round (((weight / 2.205) / (height / 100))/ 2) as BMI
  FROM `adept-bond-365418.Final_Exercise.player`
  where  round (((weight / 2.205) / (height / 100))/ 2) < 18.5
  Group by player_name ,weight , height
  ORDER BY BMI )
```

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?

```
  with table1 as
(SELECT season, team_long_name,team_api_id,
  sum(home_team_goal) as home_goal
FROM `adept-bond-365418.Final_Exercise.match` as match
left join `adept-bond-365418.Final_Exercise.team` as team
on match.home_team_api_id = team.team_api_id
where season = '2015/2016'
group by season, team_long_name, team_api_id
)


SELECT match.season, team.team_long_name,home_goal,
  sum(away_team_goal) as away_total,
  home_goal + sum(away_team_goal) as total_goal
FROM `adept-bond-365418.Final_Exercise.match`  as match
left join `adept-bond-365418.Final_Exercise.team` as team
on match.away_team_api_id = team.team_api_id
left join table1 as table1
on match.away_team_api_id = table1.team_api_id
where match.season = '2015/2016'
group by season, team_long_name, table1.home_goal
order by (table1.home_goal + sum(away_team_goal)) desc
```

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?

```
 with tab_home as (select season, home_team_api_id ,sum(home_team_goal) as
sum_home
from `adept-bond-365418.Final_Exercise.match`
group by season, home_team_api_id),

tab_away as (select season ,away_team_api_id, sum(away_team_goal) as
sum_away
from `adept-bond-365418.Final_Exercise.match`
group by season, away_team_api_id)



select team_api_id, season, team_long_name, total_goals
from (select *, rank() over (partition by season order by total_goals desc)
as max_goal
from (select team.team_api_id, away.season, team_long_name,
home.sum_home+away.sum_away as total_goals
from tab_home as home
left join tab_away as away
  on home.home_team_api_id= away.away_team_api_id
  and home.season=away.season
join `adept-bond-365418.Final_Exercise.team` team
  on away.away_team_api_id=team.team_api_id
  group by team.team_api_id, season, team_long_name,
home.sum_home,away.sum_away
  order by season desc,total_goals desc)
  group by team_api_id, season, team_long_name, total_goals)
  where max_goal=1
  group by team_api_id, season, team_long_name, total_goals
order by total_goals DESC
```

10. From the query above 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?

```
CREATE TABLE `adept-bond-365418.Final_Exercise.TopScorer` as
with tab_home as (select season, home_team_api_id ,sum(home_team_goal) as
sum_home
from `adept-bond-365418.Final_Exercise.match`
group by season, home_team_api_id),

tab_away as (select season ,away_team_api_id, sum(away_team_goal) as
sum_away
from `adept-bond-365418.Final_Exercise.match`
group by season, away_team_api_id)


select team_api_id, season, team_long_name, total_goals
from (select *, rank() over (partition by season order by total_goals desc)
as max_goal
from (select team.team_api_id, away.season, team_long_name,
home.sum_home+away.sum_away as total_goals
from tab_home as home
left join tab_away as away
  on home.home_team_api_id= away.away_team_api_id
  and home.season=away.season
join `adept-bond-365418.Final_Exercise.team` team
  on away.away_team_api_id=team.team_api_id
  group by team.team_api_id, season, team_long_name,
home.sum_home,away.sum_away
  order by season desc,total_goals desc)
  group by team_api_id, season, team_long_name, total_goals)
  where max_goal=1
  group by team_api_id, season, team_long_name, total_goals
order by total_goals DESC
```