SQL Final Assignment

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
- teams.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	Link to lucidchart: https://lucid.app/lucidchart/3f3a0491-a285-43a5-9bec-b99cc6fddba4/edit?page=0_0&invitationId=inv_0e5e1260-650e-4931-a868-fea205a2fd52#
3	`SELECT distinct DATE_DIFF(max(DATE) ,min(DATE), day) AS DateDiff From` `sql-sandbox-386914.Final_Exercise.Match` Risultato: 2868 giorni
4	`SELECT season, leagues.name, min(home_team_goal) as min, round(avg(home_team_goal),2) as avg, max(home_team_goal) as max, sum(home_team_goal) as sum from` `sql-sandbox-386914.Final_Exercise.Match` `as match join` `Final_Exercise.Leagues` `as leagues on match.country_id=leagues.country_id group by season, leagues.name`

	order by sum desc limit 1 Risultato: 2009/2010 Premier League 645 goal
5	SELECT distinct season FROM `sql-sandbox-386914.Final_Exercise.Match` Risultato: 8 stagioni SELECT season, league_id as League, count(match_api_id) as TotMatches FROM `sql-sandbox-386914.Final_Exercise.Match` group by league_id, season order by season desc, league desc Risultato: Noto che nella League 1 stagione 13/14, si sono giocati solamente 12 Matches
6	CREATE TABLE Final_Exercise.PlayerBMI AS SELECT * FROM (SELECT p.id, p.player_api_id, p.player_name, p.birthday, ROUND(p.height/100,2) as m_height, ROUND(p.weight/2.205,2) as kg_weight, ROUND((p.weight/2.205) / POWER(p.height/100, 2),1) as BMI FROM `Final_Exercise.Player` as p) as player --WHERE BMI BETWEEN 18.5 and 24.5 WHERE BMI >=18.5 and BMI <= 24.5; Risultato: righe 9788
7	SELECT COUNT(p.id) - COUNT(pb.id) as ExcludingPlayerBMI FROM `Final_Exercise.Player` as p LEFT JOIN `Final_Exercise.PlayerBMI` as pb ON p.id=pb.id Risultato: 1272 players
8	SELECT team.team_api_id,team.team_long_name,sum(match.home_team_goal) + awaymatch.away_team_goal as totalgoals

	from `sql-sandbox-386914.Final_Exercise.Team` as team JOIN `sql-sandbox-386914.Final_Exercise.Match`as match on team.team_api_id=match.home_team_api_id JOIN (SELECT team_api_id,team_long_name,sum(m.away_team_goal) as away_team_goal FROM `sql-sandbox-386914.Final_Exercise.Team` as t JOIN `sql-sandbox-386914.Final_Exercise.Match`as m on t.team_api_id=m.away_team_api_id WHERE m.season='2015/2016' GROUP BY team_api_id,team_long_name) as awaymatch on team.team_api_id=awaymatch.team_api_id where match.season='2015/2016' group by team_api_id,team_long_name,awaymatch.away_team_goal order by totalgoals desc LIMIT 5 Risultato: FC Barcellona
9	
10	

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.
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)?
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. max
d. Sum

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

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

(additional, but not evaluated)

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?
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?