2.9.x Final Assignment

Nome: ***Davide D'Angelo***

Email: ***dangelodavide.work@gmail.com***

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
- match.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	Link to lucidchart: https://lucid.app/lucidchart/c811d4d4-53f4-4b2b-9a37-e957eaac93bd/edit?viewport_loc=-2003%2C-608%2C3450%2C2840%2C0_0&invitationId=inv_8ae58b65-0203-4ccf-959e-becca4e70e89

3	`SELECT DATE_DIFF(MAX(date), MIN(date), DAY) AS number_of_days`
	`FROM ` `` `helical-loop-386715.Final_Exercise.match` ``
	`LIMIT 1000`
	Risposta: *2868 giorni*

4	`CREATE TABLE ` `` `helical-loop-386715.Final_Exercise.PlayerBMI` `` ` AS`
	`SELECT`
	`m.season,`

```
 l.name AS league_name,
 MIN(m.home_team_goal) AS min_goals,
 AVG(m.home_team_goal) AS average_goals,
 (MIN(m.home_team_goal) + MAX(m.home_team_goal)) / 2 AS mid_range_goals,
 MAX(m.home_team_goal) AS max_goals,
 SUM(m.home_team_goal) AS total_goals
FROM
 `helical-loop-386715.Final_Exercise.match` AS m
JOIN
`helical-loop-386715.Final_Exercise.leagues` AS l
ON
 m.league_id = l.id
GROUP BY
 m.season,
 L.name
ORDER BY total_goals desc
```

Risposta: *2009/2010 - England Premier League*

5

```
SELECT count(distinct(season))
FROM  `helical-loop-386715.Final_Exercise.match`
```

Risposta: *8 Stagioni Uniche*

```
SELECT m.season, l.name, league_id,count(m.league_id)
FROM `helical-loop-386715.Final_Exercise.match` AS m
JOIN `helical-loop-386715.Final_Exercise.leagues` AS l ON m.league_id = l.id
GROUP BY m.season, m.league_id, l.name
```

Risposta: *Sono state giocate più partite dal 2008 al 2016*

6

```
SELECT
player_name,
weight/2.205 as kg_weight,
height/100 AS m_height,
(weight/2.205)/((height/100)*(height/100)) AS BMI,
FROM `helical-loop-386715.Final_Exercise.player` AS player
WHERE (weight/2.205)/((height/100)*(height/100)) > 18.5 AND
(weight/2.205)/((height/100)*(height/100)) < 24.9
```

Risposta: *10197 righe*

7

```
SELECT COUNT(*) FROM(
SELECT player_name, weight/2.205 as kg_weight, height/100 AS m_height,
```

	(weight/2.205)/((height/100)*(height/100)) AS BMI,
	FROM `helical-loop-386715.Final_Exercise.player` AS player
	WHERE (weight/2.205)/((height/100)*(height/100)) <= 18.5 OR
	(weight/2.205)/((height/100)*(height/100)) >= 24.9)
	Risposta: *863*
8	SELECT season, team_long_name, SUM(home_team_goal + away_team_goal) AS
	total_goals
	FROM `helical-loop-386715.Final_Exercise.match` AS match
	JOIN `helical-loop-386715.Final_Exercise.team` AS team ON
	match.home_team_api_id = team.team_api_id
	GROUP BY season, team_long_name
	Risposta: *Real Madrid CF - 86 Goal*
9	SELECT season, team_long_name, SUM(home_team_goal + away_team_goal) AS
	total_goals
	FROM `helical-loop-386715.Final_Exercise.match` AS match
	JOIN `helical-loop-386715.Final_Exercise.team` AS team ON
	match.home_team_api_id = team.team_api_id
	GROUP BY season, team_long_name
	HAVING total_goals = (
	SELECT MAX(total_goals)
	FROM (
	SELECT season, SUM(home_team_goal + away_team_goal) AS total_goals
	FROM `helical-loop-386715.Final_Exercise.match`
	GROUP BY season, home_team_api_id
	) AS season_totals
	WHERE season_totals.season = match.season
	)
	ORDER BY season ASC;
	Risposta: *Real Madrid CF*
10	CREATE TABLE helical-loop-386715.Final_Exercise.Topscorer AS
	SELECT team_long_name, tea.id, SUM (home_team_goal + away_team_goal) AS
	total_goals
	FROM `helical-loop-386715.Final_Exercise.match`AS mat
	JOIN `helical-loop-386715.Final_Exercise.team`AS tea ON mat.home_team_api_id
	= tea.team_api_id
	GROUP BY team_long_name,tea.id

```sql
order by total_goals DESC
LIMIT 10

SELECT COUNT(*) AS pair_combinations_count
FROM (
SELECT t1.team_long_name AS team1, t2.team_long_name AS team2
FROM `helical-loop-386715.Final_Exercise.Topscorer` AS t1
JOIN `helical-loop-386715.Final_Exercise.Topscorer` AS t2
ON t1.team_long_name < t2.team_long_name
) AS combinations
```

Risposta: *45 Pair Combinations*

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. 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?

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?
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?