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: [European Soccer DB schema](https://lucid.app/lucidchart/28053e32-ddff-4ba0-8b84-ba3c9370d7ab/edit?viewport_loc=-52%2C12%2C1461%2C768%2CQLsZbFSKZ-id&invitationId=inv_66bdf101-cdd6-4b0f-a30e-89e4580be85a) |
| 3 | SELECT    DATETIME\_DIFF(MAX(date), MIN(date), day) AS days\_between  FROM    `homework-1-386813.Final\_Exercise.match`  2868 |
| 4 | a. MIN:  SELECT    m.season, l.name, min(m.home\_team\_goal) as min\_home\_goal  FROM    `homework-1-386813.Final\_Exercise.match` as m  JOIN    `homework-1-386813.Final\_Exercise.leagues` as l  ON    m.league\_id = l.id  GROUP BY m.season, l.name  ORDER BY m.season  b. AVERAGE  SELECT    m.season, l.name, CAST(avg(m.home\_team\_goal) AS INTEGER) as avg\_home\_goal  FROM    `homework-1-386813.Final\_Exercise.match` as m  JOIN    `homework-1-386813.Final\_Exercise.leagues` as l  ON    m.league\_id = l.id  GROUP BY m.season, l.name  ORDER BY m.season, avg\_home\_goal DESC  c. MAX  SELECT    m.season, l.name, max(m.home\_team\_goal) as max\_home\_goal  FROM    `homework-1-386813.Final\_Exercise.match` as m  JOIN    `homework-1-386813.Final\_Exercise.leagues` as l  ON    m.league\_id = l.id  GROUP BY m.season, l.name  ORDER BY m.season, max\_home\_goal DESC  d. SUM  SELECT    m.season, l.name, sum(m.home\_team\_goal) as tot\_home\_goal  FROM    `homework-1-386813.Final\_Exercise.match` as m  JOIN    `homework-1-386813.Final\_Exercise.leagues` as l  ON    m.league\_id = l.id  GROUP BY m.season, l.name  ORDER BY m.season, tot\_home\_goal DESC |
| 5 | SELECT    m.season, l.name, count(m.match\_api\_id) as tot\_match  FROM    `homework-1-386813.Final\_Exercise.match` as m  JOIN    `homework-1-386813.Final\_Exercise.leagues` as l  ON    m.league\_id = l.id  GROUP BY m.season, l.name  in the 2013/2014 season the Belgium jupiler league played only 12 matches. |
| 6 | a. b. c.  CREATE TABLE `homework-1-386813.Final\_Exercise.PlayerBMI` AS  SELECT    id, player\_api\_id, player\_name, birthday, ROUND((weight / 2.205),1) as kg\_weight, ROUND((height / 100),2) as m\_height,    ROUND((weight / 2.205)/ POW((height / 100),2),1) as BMI  FROM    `homework-1-386813.Final\_Exercise.player`  d.  SELECT \* FROM `homework-1-386813.Final\_Exercise.PlayerBMI` WHERE BMI >= 18.5 AND BMI <= 24.9  ORDER BY BMI |
| 7 | SELECT count (\*) as players\_w\_unhealth\_range  FROM  (  SELECT \* FROM `homework-1-386813.Final\_Exercise.PlayerBMI` WHERE NOT( BMI >= 18.5 AND BMI <= 24.9 )  )  815 |
| 8 | WITH    Home AS ( /\*Calculate total home\_goals for each team in the most recent season \*/    SELECT      home\_team\_api\_id AS team\_id,      SUM(home\_team\_goal) AS home\_goals    FROM      `homework-1-386813.Final\_Exercise.match`    WHERE      season = (SELECT MAX(season) FROM`homework-1-386813.Final\_Exercise.match`)    GROUP BY      home\_team\_api\_id),      Away AS ( /\*Calculate total away\_goals for each team in the most recent season\*/    SELECT      away\_team\_api\_id AS team\_id,      SUM(away\_team\_goal) AS away\_goals    FROM      `homework-1-386813.Final\_Exercise.match`    WHERE      season = (SELECT MAX(season) FROM `homework-1-386813.Final\_Exercise.match`)    GROUP BY      away\_team\_api\_id)    /\* Join the previous two subqueries on the team\_id then join it with the team table to get the team long name\*/  SELECT    Team.team\_long\_name,    Home.home\_goals AS home\_goals,    Away.away\_goals AS away\_goals,    Home.home\_goals + Away.away\_goals AS tot\_goals  FROM    Home  JOIN    Away  ON    Home.team\_id = Away.team\_id  JOIN    `homework-1-386813.Final\_Exercise.team` AS Team  ON    Home.team\_id = Team.team\_api\_id  ORDER BY    tot\_goals DESC  FC Barcelona with 112 goals |
| 9 | WITH    Home AS ( /\*Calculate total home\_goals for each team in the most recent season \*/    SELECT      season,      home\_team\_api\_id AS team\_id,      SUM(home\_team\_goal) AS home\_goals    FROM      `homework-1-386813.Final\_Exercise.match`    GROUP BY      season,      home\_team\_api\_id),      Away AS ( /\*Calculate total away\_goals for each team in the most recent season\*/    SELECT      season,      away\_team\_api\_id AS team\_id,      SUM(away\_team\_goal) AS away\_goals    FROM      `homework-1-386813.Final\_Exercise.match`    GROUP BY      season,      away\_team\_api\_id),  Tot\_goals\_each\_season AS (    /\* Show total goal scored for each season and each team ranking by season\*/  SELECT    Home.season as season,    Team.team\_long\_name as team\_name,    Home.home\_goals AS home\_goals,    Away.away\_goals AS away\_goals,    Home.home\_goals + Away.away\_goals AS tot\_goals,    RANK () OVER (PARTITION BY Home.season ORDER BY Home.home\_goals + Away.away\_goals DESC ) as rank  FROM    Home  JOIN    Away  ON    Home.team\_id = Away.team\_id AND Home.season = Away.season  JOIN    `homework-1-386813.Final\_Exercise.team` AS Team  ON    Home.team\_id = Team.team\_api\_id  ORDER BY    Home.season,    tot\_goals DESC  )  /\* Last query\*/  SELECT season, team\_name, tot\_goals  FROM Tot\_goals\_each\_season  WHERE rank = 1 |
| 10 | CREATE TABLE `homework-1-386813.Final\_Exercise.TopScorer` AS  (  WITH    Home AS ( /\*Calculate total home\_goals for each team in the most recent season \*/    SELECT      home\_team\_api\_id AS team\_id,      SUM(home\_team\_goal) AS home\_goals    FROM      `homework-1-386813.Final\_Exercise.match`    WHERE      season = (SELECT MAX(season) FROM`homework-1-386813.Final\_Exercise.match`)    GROUP BY      home\_team\_api\_id),      Away AS ( /\*Calculate total away\_goals for each team in the most recent season\*/    SELECT      away\_team\_api\_id AS team\_id,      SUM(away\_team\_goal) AS away\_goals    FROM      `homework-1-386813.Final\_Exercise.match`    WHERE      season = (SELECT MAX(season) FROM `homework-1-386813.Final\_Exercise.match`)    GROUP BY      away\_team\_api\_id)    /\* Join the previous two subqueries on the team\_id then join it with the team table to get the team long name\*/  SELECT    Team.id as team\_id,    Team.team\_long\_name as team\_name,    Home.home\_goals + Away.away\_goals AS tot\_goals  FROM    Home  JOIN    Away  ON    Home.team\_id = Away.team\_id  JOIN    `homework-1-386813.Final\_Exercise.team` AS Team  ON    Home.team\_id = Team.team\_api\_id  ORDER BY    tot\_goals DESC  LIMIT 10  )  SELECT    H.team\_name AS home\_team,    A.team\_name AS away\_team  FROM    `homework-1-386813.Final\_Exercise.TopScorer` AS H  JOIN    `homework-1-386813.Final\_Exercise.TopScorer` AS A  ON    H.team\_id < A.team\_id  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:
   1. 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).
   2. For each table, write its shape (write the number of rows and columns near the table name).
   3. 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:
   1. min
   2. average
   3. max
   4. Sum

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

1. 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?
2. Using Players as the starting point, create a new table (PlayerBMI) and add:
   1. a new variable that represents the players’ weight in kg (divide the mass value by 2.205) and call it kg\_weight;
   2. a variable that represents the height in metres (divide the cm value by 100) and call it m\_height;
   3. a variable that shows the body mass index (BMI) of the player;  
      *Hint: research how to calculate the formula of the BMI*
   4. 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?

1. How many players do not have an optimal BMI?
2. 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)

1. 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?
2. 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?