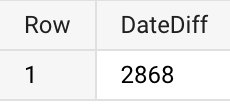
1. How many days have passed from the oldest **Match** to the most recent one (dataset time interval)?  
   **Answer: 2868**   
   SELECT DATE\_DIFF(max(date), min(date), day) AS DateDiff

FROM `sql-sandbox-347110.Final\_Exercise.match`;  


1. Produce a table which, for each Season and **League** Name, shows the following statistics about the home goals scored:
   1. min
   2. average
   3. mid-range
   4. max
   5. 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 `sql-sandbox-347110.Final\_Exercise.match` a

LEFT JOIN `sql-sandbox-347110.Final\_Exercise.leagues` b

on a.league\_id = b.id

GROUP BY a.season, b.name

ORDER BY sum\_home\_team\_goal desc  


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?  
   **Answer: Belgium Jupiler League only has 12 home goals in 2013/2014**  
   SELECT distinct season

FROM `sql-sandbox-347110.Final\_Exercise.match`

  
SELECT a.season, b.name as league\_name,

count(a.id) as Matches\_played

FROM `sql-sandbox-347110.Final\_Exercise.match` a

LEFT JOIN `sql-sandbox-347110.Final\_Exercise.leagues` b

on a.league\_id = b.id

GROUP BY a.season, b.name  


1. 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?

**Answer: 10,197**   
CREATE TABLE Final\_Exercise.PlayerBMI AS

SELECT \*,

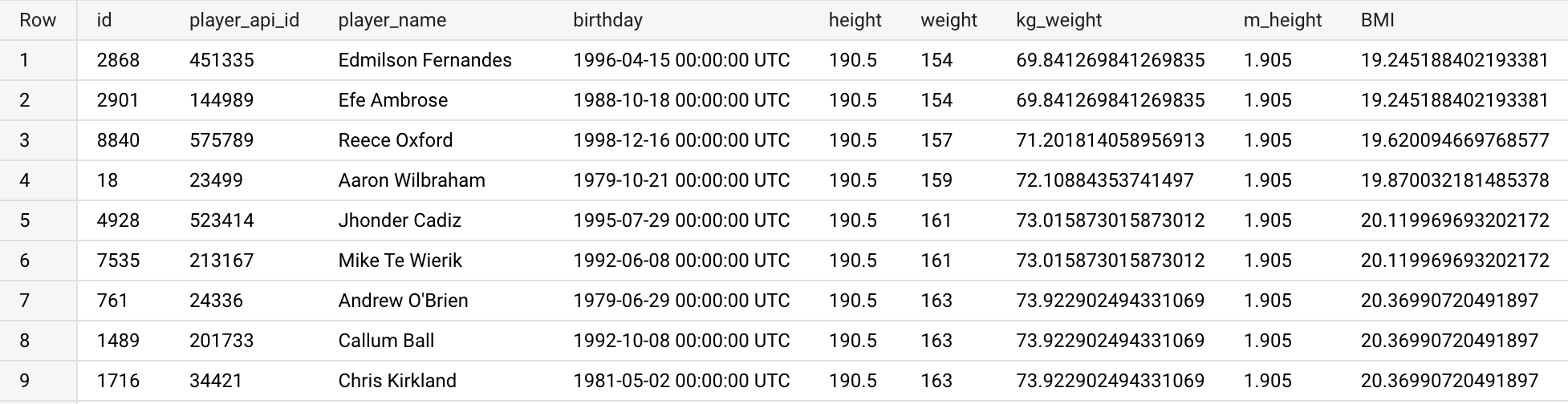
weight/2.205 as kg\_weight,

height/100 as m\_height,

(weight/2.205)/power(height/100, 2) as BMI

FROM `sql-sandbox-347110.Final\_Exercise.player`

WHERE (weight/2.205)/power(height/100, 2) between 18.5 and 24.9



1. How many players do not have an optimal BMI?   
   **Answer: 863**  
   SELECT

(SELECT count(id)

FROM `sql-sandbox-347110.Final\_Exercise.player`) -

(SELECT count(id)

FROM `sql-sandbox-347110.Final\_Exercise.PlayerBMI`) as PlayerNoBMI

Or, alternatively:

SELECT count(\*) as PlayerNoBMI

FROM `sql-sandbox-347110.Final\_Exercise.player` a

LEFT JOIN `sql-sandbox-347110.Final\_Exercise.PlayerBMI` b

on a.id = b.id

WHERE b.id is null



1. 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 `sql-sandbox-347110.Final\_Exercise.match` m

inner join `sql-sandbox-347110.Final\_Exercise.team` t

on m.home\_team\_api\_id = t.team\_api\_id

where m.season = (select max(season) from `sql-sandbox-347110.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 `sql-sandbox-347110.Final\_Exercise.match` m

inner join `sql-sandbox-347110.Final\_Exercise.team` t

on m.away\_team\_api\_id = t.team\_api\_id

where m.season = (select max(season) from `sql-sandbox-347110.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

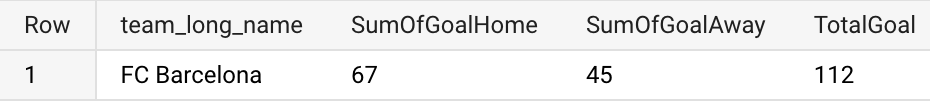
limit 1

on a.team\_api\_id = b.team\_api\_id

GROUP BY b.team\_long\_name

ORDER BY goals desc

LIMIT 1



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?   
   **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 `sql-sandbox-351014.Final\_Exercise.Match` m inner join

`sql-sandbox-351014.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 `sql-sandbox-351014.Final\_Exercise.Match` m inner join

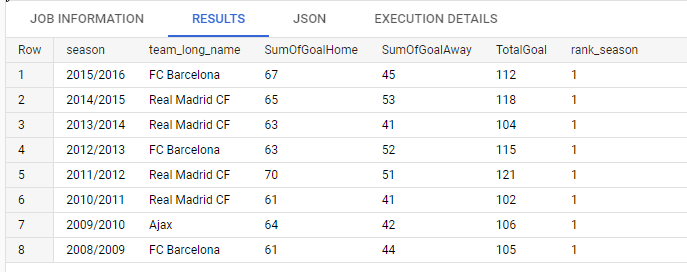
`sql-sandbox-351014.Final\_Exercise.Team` t on m.away\_team\_api\_id = t.team\_api\_id

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



1. 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 `sql-sandbox-347110.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 `sql-sandbox-347110.Final\_Exercise.match` m inner join

`sql-sandbox-347110.Final\_Exercise.team` t on m.home\_team\_api\_id = t.team\_api\_id

where m.season = (select max(season) from `sql-sandbox-347110.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 `sql-sandbox-347110.Final\_Exercise.match` m inner join

`sql-sandbox-347110.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 `sql-sandbox-347110.Final\_Exercise.TopScorer` a

LEFT JOIN `sql-sandbox-347110.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

SELECT a.team\_long\_name, b.team\_long\_name

FROM `sql-sandbox-347110.Final\_Exercise.TopScorer` a

INNER JOIN `sql-sandbox-347110.Final\_Exercise.TopScorer` b

on a.team\_api\_id > b.team\_api\_id

