To compare the performance with multiple queries for the below,

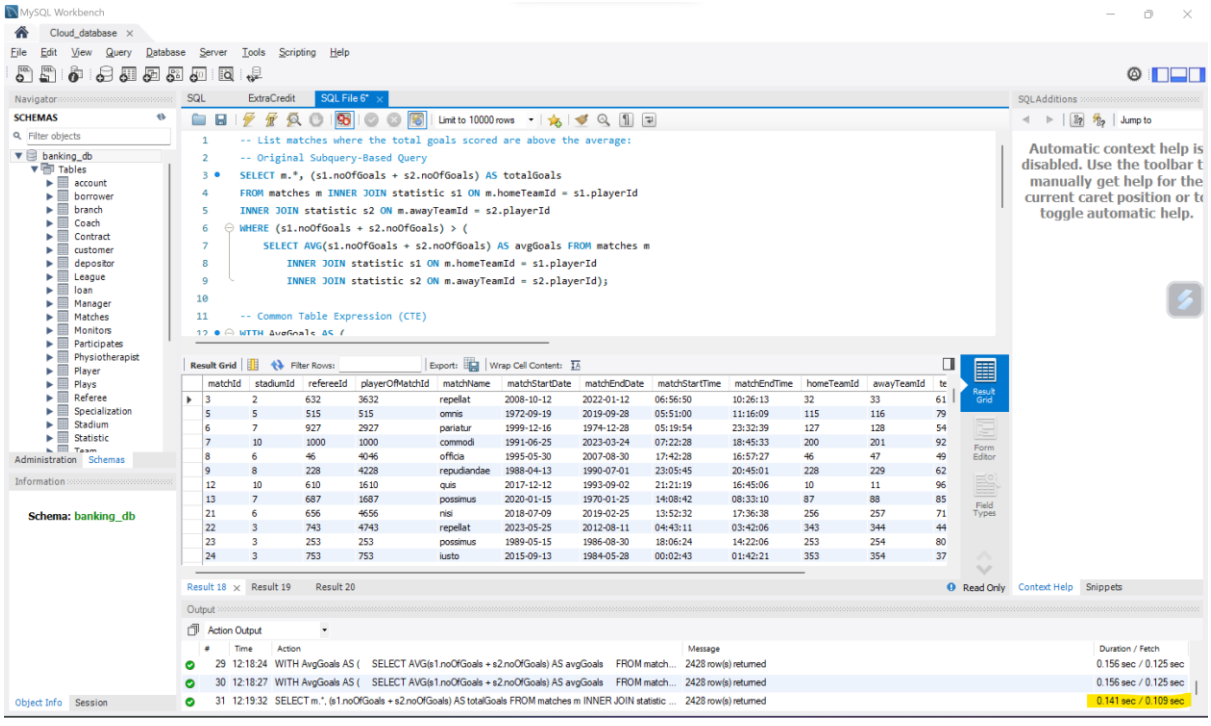
List matches where the total goals scored are above the average:

1) Original Subquery-Based Query:

```
SELECT m.*, (s1.noOfGoals + s2.noOfGoals) AS totalGoals
FROM matches m INNER JOIN statistic s1 ON m.homeTeamId = s1.playerId
INNER JOIN statistic s2 ON m.awayTeamId = s2.playerId
WHERE (s1.noOfGoals + s2.noOfGoals) > (
        SELECT AVG(s1.noOfGoals + s2.noOfGoals) AS avgGoals FROM matches m
      INNER JOIN statistic s1 ON m.homeTeamId = s1.playerId
      INNER JOIN statistic s2 ON m.awayTeamId = s2.playerId);
```

Console logs:

12:19:32 SELECT m.*, (s1.noOfGoals + s2.noOfGoals) AS totalGoals FROM matches m INNER JOIN statistic s1 ON m.homeTeamId = s1.playerId INNER JOIN statistic s2 ON m.awayTeamId = s2.playerId WHERE (s1.noOfGoals + s2.noOfGoals) > (SELECT AVG(s1.noOfGoals + s2.noOfGoals) AS avgGoals FROM matches m INNER JOIN statistic s1 ON m.homeTeamId = s1.playerId INNER JOIN statistic s2 ON m.awayTeamId = s2.playerId) LIMIT 0, 10000 **2428 row(s) returned0.141 sec / 0.109 sec**



Subquery-Based Query:

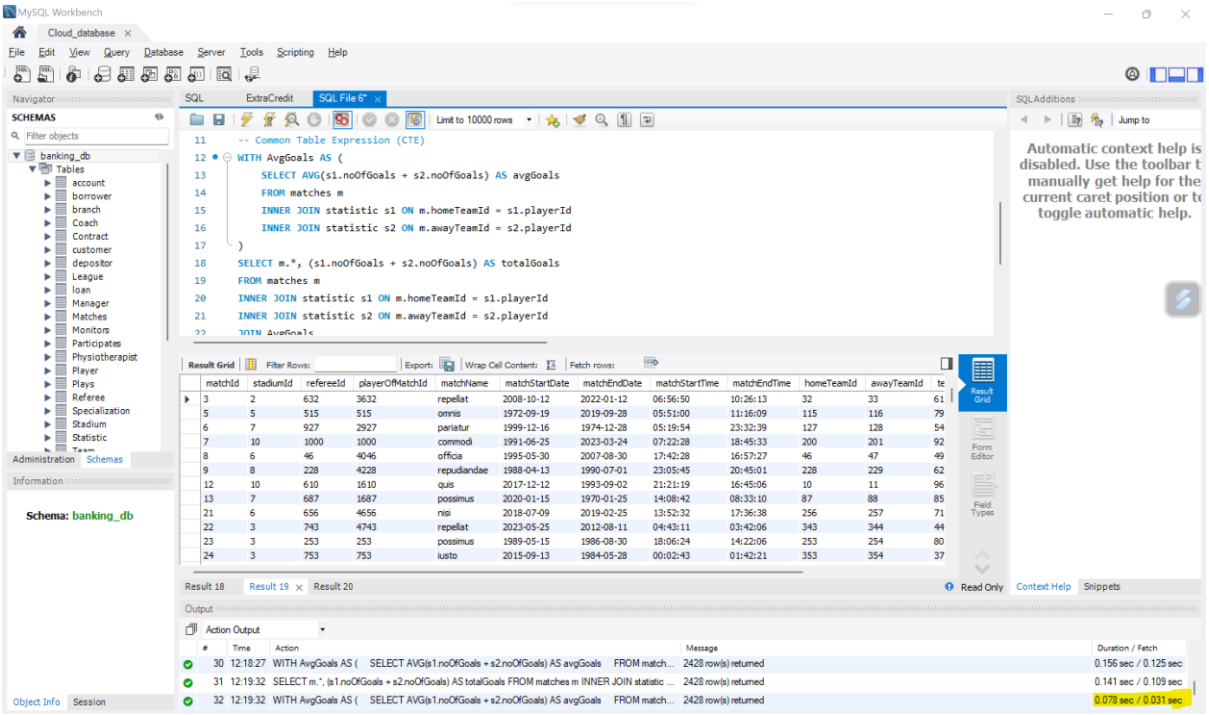
- The main query selects information about matches (m.*) and calculates the total number of goals for each match by summing the goals scored by the home and away teams (s1.noOfGoals + s2.noOfGoals).
- The WHERE clause filters the results to include only those matches where the total goals are greater than the average.
- The average is calculated using a subquery that joins the matches table with the statistic table twice to get the home and away team statistics, and then calculates the average of the total goals.
- This query achieves the desired result, its performance may be influenced by factors such as subquery overhead, repetitive joins, and indexing.

2) Common Table Expression (CTE):

```
WITH AvgGoals AS (
   SELECT AVG(s1.noOfGoals + s2.noOfGoals) AS avgGoals
   FROM matches m
   INNER JOIN statistic s1 ON m.homeTeamId = s1.playerId
   INNER JOIN statistic s2 ON m.awayTeamId = s2.playerId
)
SELECT m.*, (s1.noOfGoals + s2.noOfGoals) AS totalGoals
FROM matches m
INNER JOIN statistic s1 ON m.homeTeamId = s1.playerId
INNER JOIN statistic s2 ON m.awayTeamId = s2.playerId
JOIN AvgGoals
WHERE (s1.noOfGoals + s2.noOfGoals) > avgGoals;
```

Console logs:

12:19:32 WITH AvgGoals AS (SELECT AVG(s1.noOfGoals + s2.noOfGoals) AS avgGoals FROM matches m INNER JOIN statistic s1 ON m.homeTeamId = s1.playerId INNER JOIN statistic s2 ON m.awayTeamId = s2.playerId) SELECT m.*, (s1.noOfGoals + s2.noOfGoals) AS totalGoals FROM matches m INNER JOIN statistic s1 ON m.homeTeamId = s1.playerId INNER JOIN statistic s2 ON m.awayTeamId = s2.playerId JOIN AvgGoals WHERE (s1.noOfGoals + s2.noOfGoals) > avgGoals **2428 row(s) returned 0.078 sec / 0.031 sec**



Common Table Expression (CTE):

A temporary table (AvgGoals) is created to calculate the average number of goals scored in all matches. This CTE is computed once.

Main Query:

- The main query selects information about matches (m.*) and calculates the total number of goals for each match by summing the goals scored by the home and away teams (s1.noOfGoals + s2.noOfGoals).
- The main query then joins with the AvgGoals CTE.
- The WHERE clause filters the results to include only those matches where the total goals are greater than the average calculated in the CTE.
- This query is designed to be efficient by calculating the average only once and then using that result in the main query.

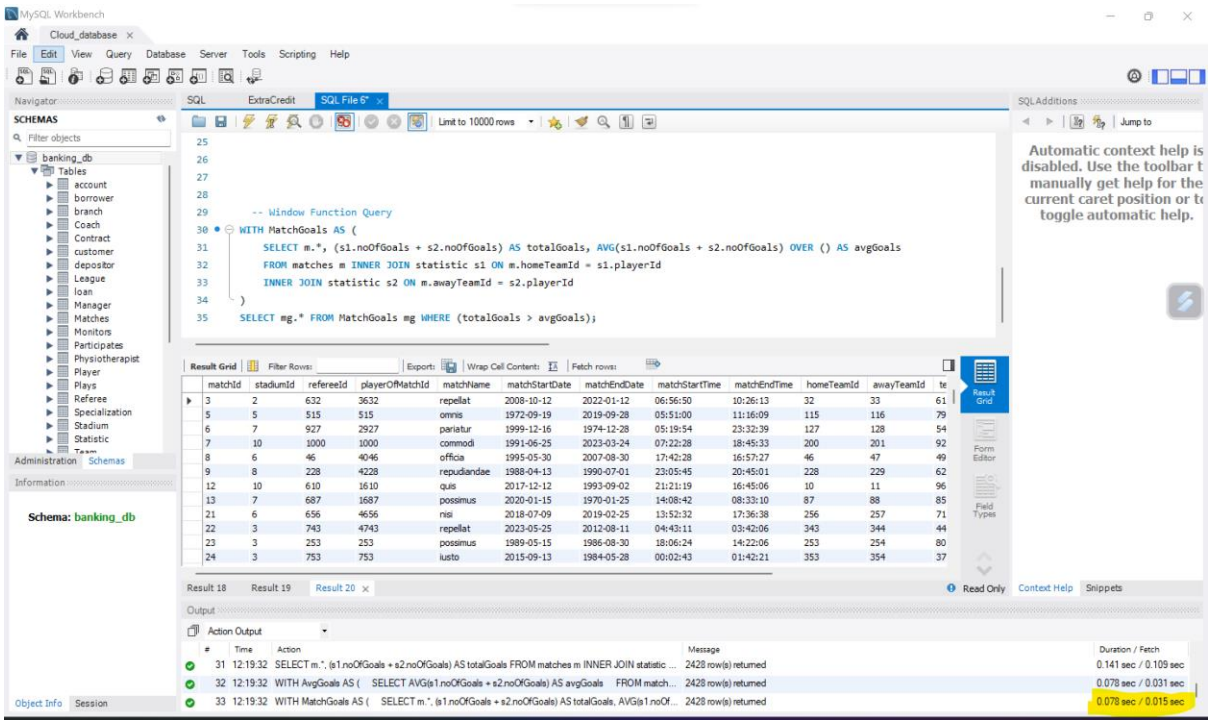
3) Window Function Query:

```
WITH MatchGoals AS (
   SELECT m.*, (s1.noOfGoals + s2.noOfGoals) AS totalGoals, AVG(s1.noOfGoals +
s2.noOfGoals) OVER () AS avgGoals
   FROM matches m INNER JOIN statistic s1 ON m.homeTeamId = s1.playerId
   INNER JOIN statistic s2 ON m.awayTeamId = s2.playerId
)
SELECT mg.* FROM MatchGoals mg WHERE (totalGoals > avgGoals);
```

Console logs:

12:19:32 WITH MatchGoals AS (SELECT m.*, (s1.noOfGoals + s2.noOfGoals) AS totalGoals, AVG(s1.noOfGoals + s2.noOfGoals) OVER () AS avgGoals FROM matches m INNER JOIN statistic s1 ON m.homeTeamId = s1.playerId INNER JOIN statistic s2 ON

m.awayTeamId = s2.playerId) SELECT mg.* FROM MatchGoals mg WHERE (totalGoals > avgGoals) **2428 row(s) returned0.078 sec / 0.015 sec**



Window Function Query:

- A Common Table Expression (CTE) named MatchGoals is created.
- Within the CTE, the query selects information about matches (m.*) and calculates the total number of goals for each match by summing the goals scored by the home and away teams (s1.noOfGoals + s2.noOfGoals).
- A window function (AVG(...) OVER ()) is applied to calculate the average of total goals for all matches.

Main Query:

- The main query selects all columns from the MatchGoals CTE.
- The WHERE clause filters the results to include only those matches where the total goals (totalGoals) are greater than the calculated average (avgGoals).
- This query efficiently calculates the average total goals using a window function and then filters matches based on this average.

Summary:

All three queries appear to achieve the same result—selecting matches where the total number of goals is greater than the average number of goals.

The differences lie in the syntax and structure of the queries, with the second and third queries utilizing CTEs for better readability and potentially improved performance.

The execution times are also provided for each query. The third query, in particular, seems to have the lowest execution time, indicating potential efficiency.