

ALL the Subqueries EVERYWHERE (Corrected)

Learn / Courses / Data Manipulation In SQL

← Course Outline →

Daily XP 1717

Exercise

ALL the subqueries EVERYWHERE

In soccer leagues, games are played at different *stages*. Winning teams progress from one stage to the next, until they reach the final stage. In each stage, the stakes become higher than the previous one. The `match` table includes data about the different stages that each match took place in.

In this lesson, you will build a final query across 3 exercises that will contain three subqueries -- one in the `SELECT` clause, one in the `FROM` clause, and one in the `WHERE` clause. In the final exercise, your query will extract data examining the average goals scored in each stage of a match. Does the average number of goals scored change as the stakes get higher from one stage to the next?

Instructions

100 XP

- Extract the average number of home and away team goals in two `SELECT` subqueries.
- Calculate the average home and away goals for the specific stage in the main query.
- Filter both subqueries and the main query so that only data from the 2012/2013 season is included.
- Group the query by the `m.stage` column.

Take Hint (-30 XP)

query.sql

Light Mode

```
1 SELECT
2     -- Select the stage and average goals for each stage
3     m.stage,
4     ROUND(____ + m.away_goal),2) AS avg_goals,
5     -- Select the average overall goals for the 2012/2013
6     season
7     ROUND((SELECT ____ + away_goal
8             FROM match
9             WHERE season = ____),2) AS overall
10 FROM match AS m
11 -- Filter for the 2012/2013 season
12 WHERE ____ = ____
13 -- Group by stage
14 GROUP BY ____;
```

↶ Run Code Submit Answer

query result

match

No query executed yet...

Showing 0 out of 0 rows

Question

1. Extract the average number of home and away team goals in two `SELECT` subqueries.
2. Calculate the average home and away goals for the specific stage in the main query.
3. Filter both subqueries and the main query so that only data from the 2012/2013 season is included.

4. Group the query by the m.stage column.

Corrected Solution

```
SELECT
    m.stage,
    ROUND(AVG(m.home_goal + m.away_goal), 2) AS avg_goals,
    ROUND(
        (SELECT AVG(home_goal + away_goal)
         FROM match
         WHERE season = '2012/2013'),
        2) AS overall
FROM match AS m
WHERE season = '2012/2013'
GROUP BY m.stage;
```

Explanation

This query calculates the average goals scored per stage in the 2012/2013 season and compares it with the overall average goals scored for the entire season. The subquery calculates the overall average without using the alias 'm', which avoids the error caused by referencing an undefined alias in the subquery. The main query groups the results by stage for clarity.