

DATABASE ORGANIZATION – CS425

Project Phase - 3

Group members: (Group 12)

- Udhay Chander Bharatha – A20518701
- Vaishnavi Prasanna Shetty – A20519894
- Gautham Kumar Ramappa Shanmugam – A20512380
- Arjun Mohan Kumar - A20512469

Test a variety of SQL queries of your choice on your database. Additional points will be awarded for queries derived from interesting use cases. Ensure that you test at least 15 different (including advanced window features, and OLAP) queries and provide:

- 1) the query explanation/description,
- 2) the SQL statements/commands, and
- 3) the output/result for each query.

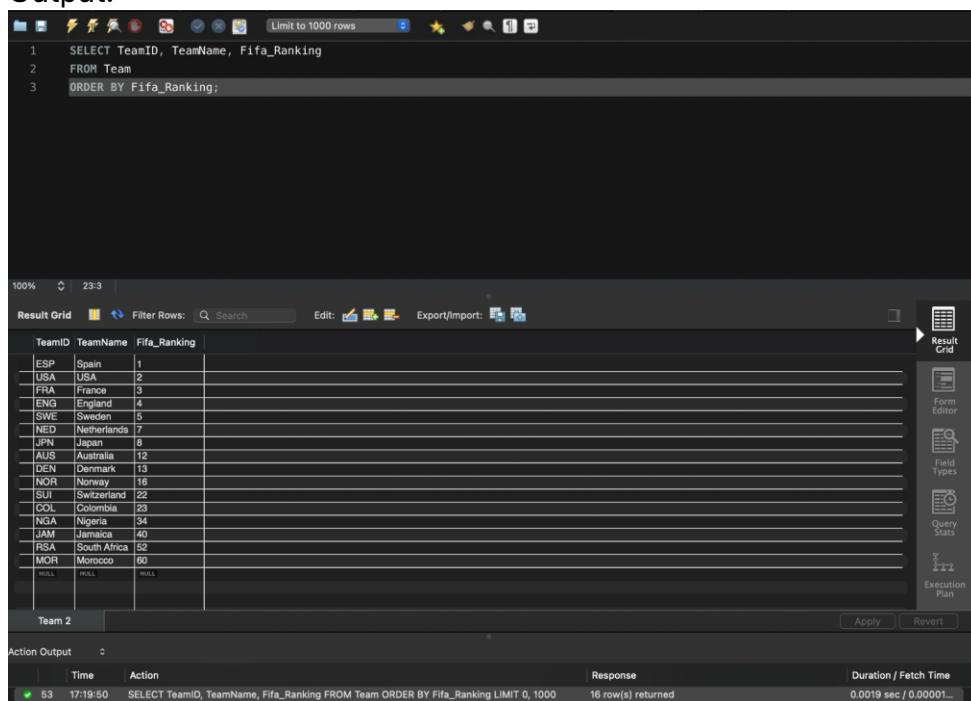
1. Query description: To retrieve team details such as TeamID and TeamName in the order of their FIFA ranking.

SQL statement:



```
Query 1
60
61   SELECT TeamID, TeamName, Fifa_Ranking
62   FROM Team
63   ORDER BY Fifa_Ranking;
64
```

Output:



| TeamID | TeamName | Fifa_Ranking |
|--------|--------------|--------------|
| ESP | Spain | 1 |
| USA | USA | 2 |
| FRA | France | 3 |
| ENG | England | 4 |
| SWE | Sweden | 5 |
| NED | Netherlands | 7 |
| JPN | Japan | 8 |
| AUS | Australia | 12 |
| DEN | Denmark | 13 |
| NOR | Norway | 16 |
| SUI | Switzerland | 22 |
| COL | Colombia | 23 |
| NGA | Nigeria | 34 |
| JAM | Jamaica | 40 |
| RSA | South Africa | 52 |
| MOR | Morocco | 60 |
| | | |
| | | |
| | | |

Action Output: 53 17:19:50 SELECT TeamID, TeamName, Fifa_Ranking FROM Team ORDER BY Fifa_Ranking LIMIT 0, 1000 16 row(s) returned Duration / Fetch Time: 0.0019 sec / 0.00001...

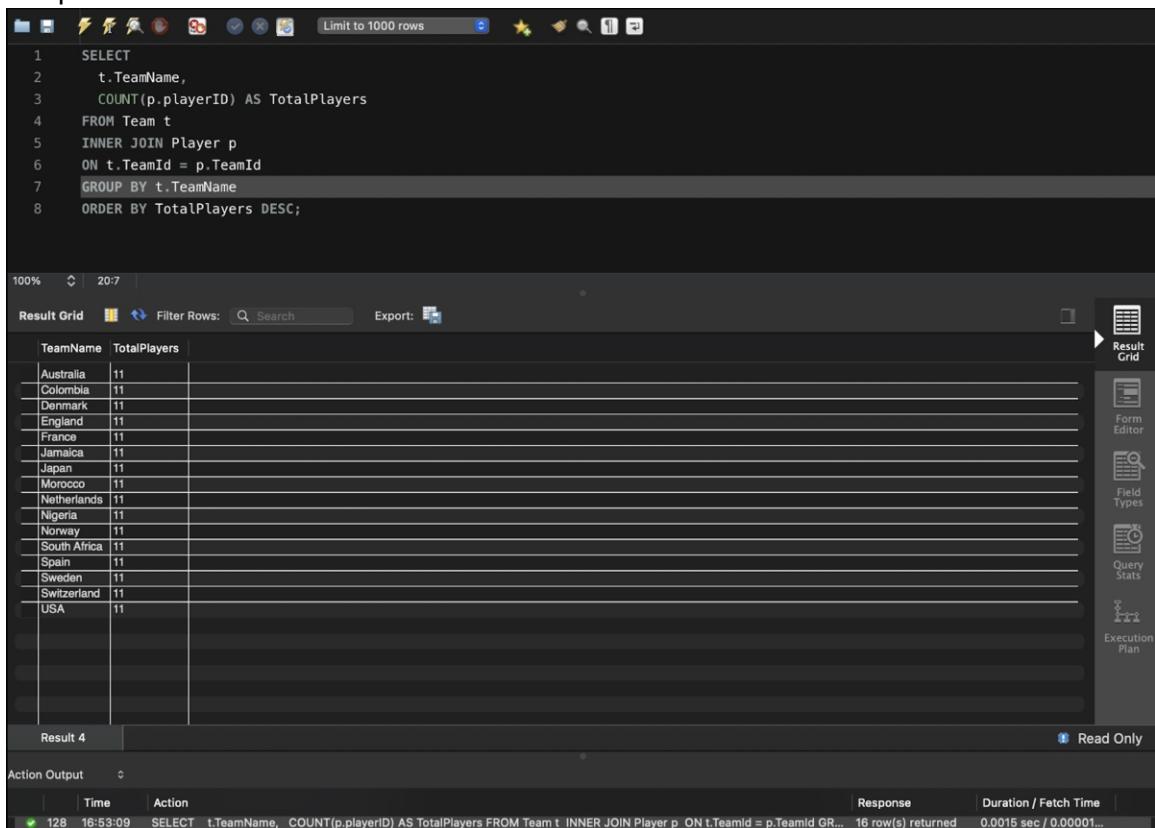
2. Query description: Retrieving the total number of players in each team.

SQL Statement:



```
58
59 *  SELECT
60      t.TeamName,
61      COUNT(p.playerID) AS TotalPlayers
62  FROM Team t
63  INNER JOIN Player p
64  ON t.TeamId = p.TeamId
65  GROUP BY t.TeamName
66  ORDER BY TotalPlayers DESC;
```

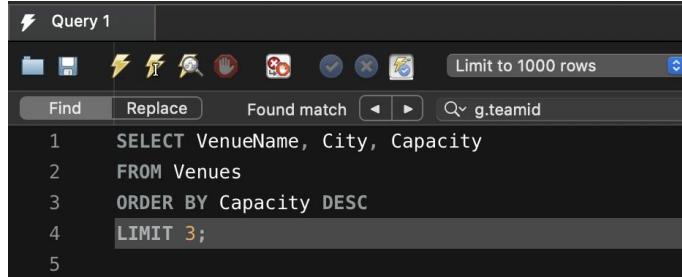
Output:



| TeamName | TotalPlayers |
|--------------|--------------|
| Australia | 11 |
| Colombia | 11 |
| Denmark | 11 |
| England | 11 |
| France | 11 |
| Jamaica | 11 |
| Japan | 11 |
| Morocco | 11 |
| Netherlands | 11 |
| Nigeria | 11 |
| Norway | 11 |
| South Africa | 11 |
| Spain | 11 |
| Sweden | 11 |
| Switzerland | 11 |
| USA | 11 |

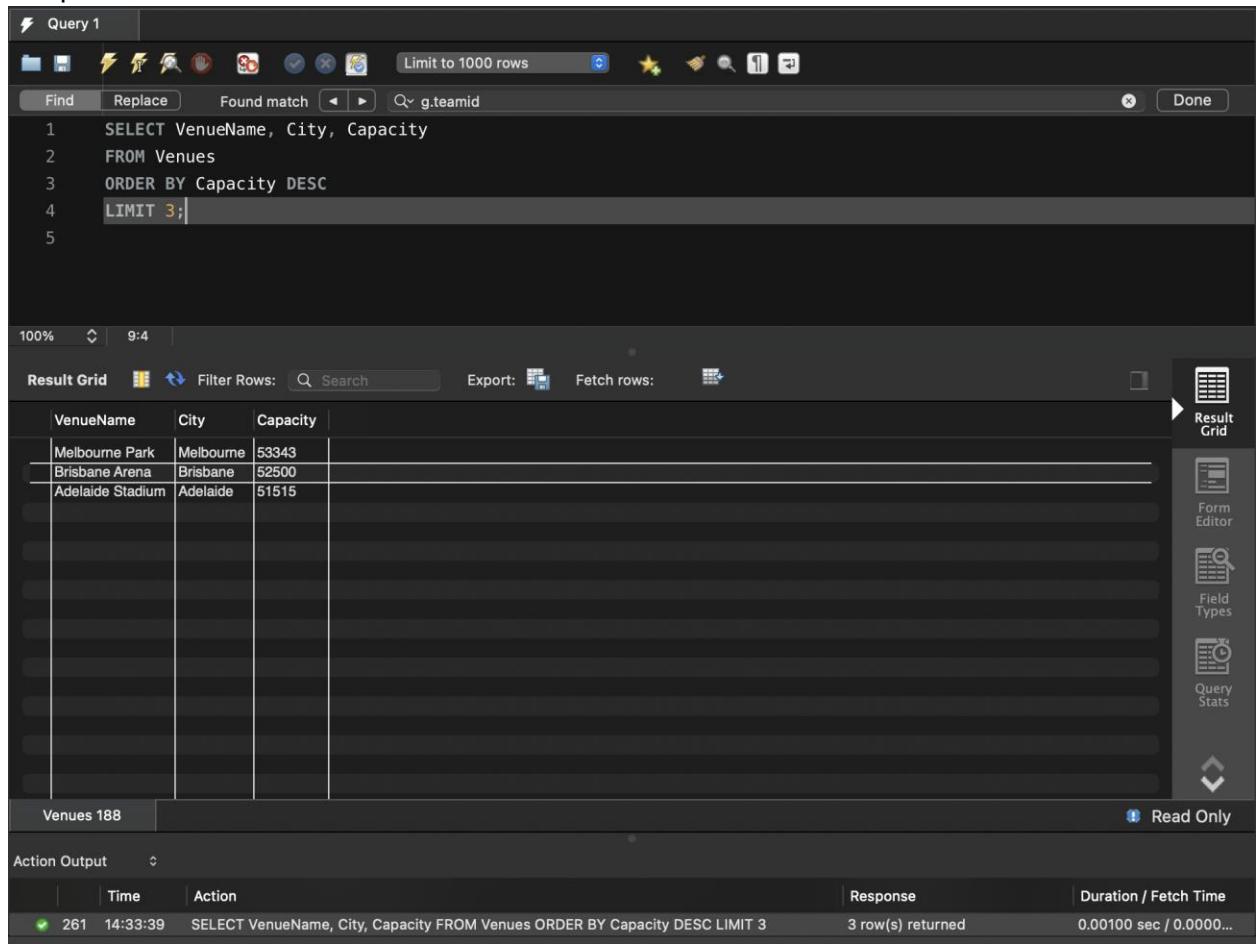
3. Query Description: Retrieving the top 3 venues with the maximum capacity.

SQL statement:



```
Query 1
Limit to 1000 rows
Find Replace Found match < > Q g.teamid
1 SELECT VenueName, City, Capacity
2 FROM Venues
3 ORDER BY Capacity DESC
4 LIMIT 3;
5
```

Output:



Query 1
Limit to 1000 rows
Find Replace Found match < > Q g.teamid
1 SELECT VenueName, City, Capacity
2 FROM Venues
3 ORDER BY Capacity DESC
4 LIMIT 3;
5

100% | 9:4 |

Result Grid Filter Rows: Search Export: Fetch rows: □

| VenueName | City | Capacity |
|------------------|-----------|----------|
| Melbourne Park | Melbourne | 53343 |
| Brisbane Arena | Brisbane | 52500 |
| Adelaide Stadium | Adelaide | 51515 |

Venues 188 Read Only

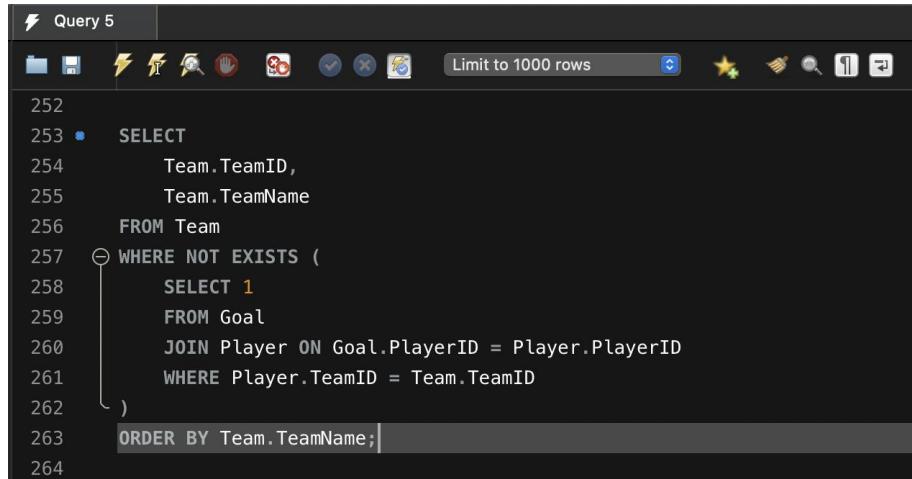
Action Output

| Time | Action | Response | Duration / Fetch Time |
|--------------|---|-------------------|-------------------------|
| 261 14:33:39 | SELECT VenueName, City, Capacity FROM Venues ORDER BY Capacity DESC LIMIT 3 | 3 row(s) returned | 0.00100 sec / 0.0000... |

Result Grid
Form Editor
Field Types
Query Stats

4. Query Description: List of the teams which have not scored any goals in the tournament.

SQL Statement:



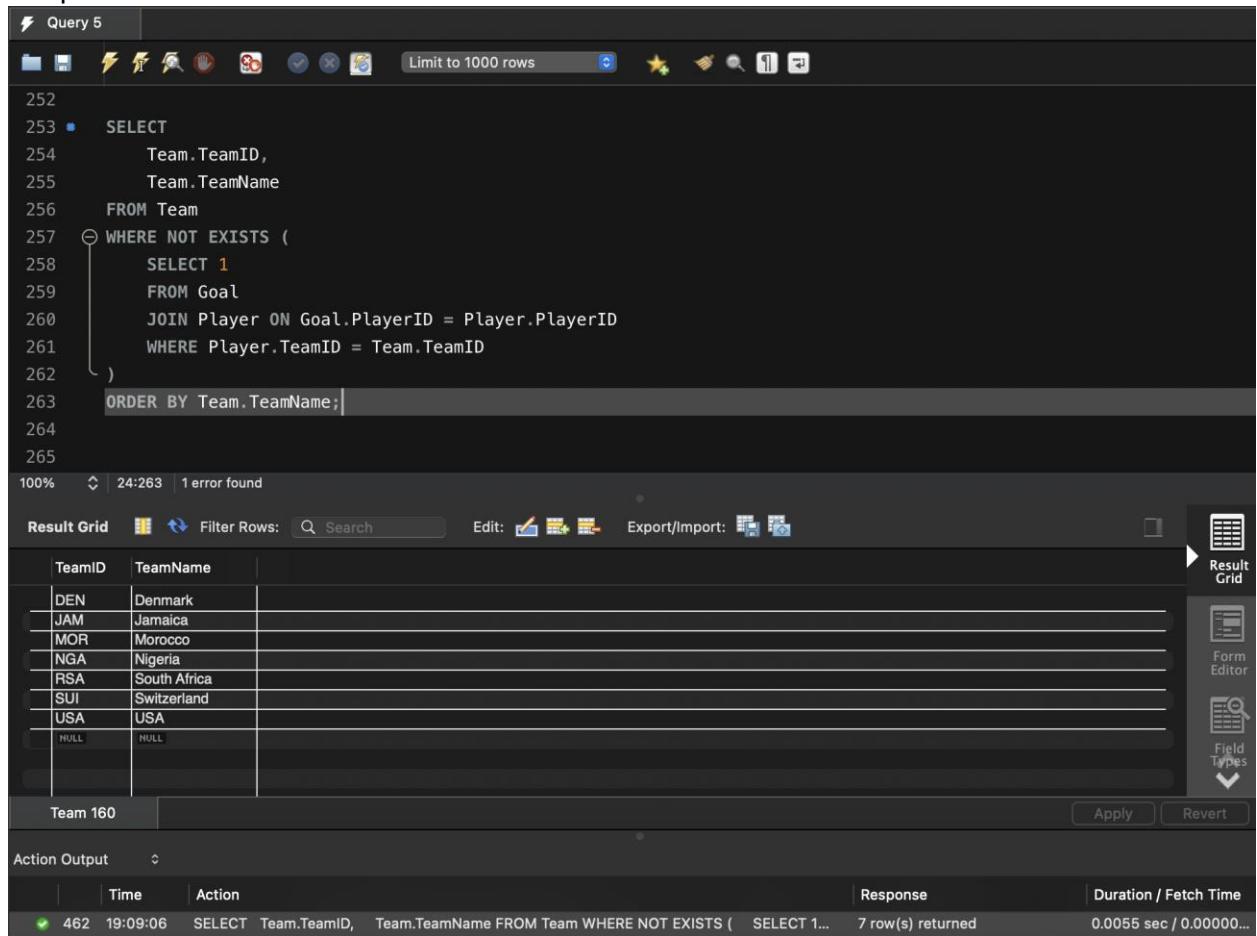
```

Query 5

252
253 •   SELECT
254     Team.TeamID,
255     Team.TeamName
256   FROM Team
257   ⊖ WHERE NOT EXISTS (
258       SELECT 1
259       FROM Goal
260       JOIN Player ON Goal.PlayerID = Player.PlayerID
261       WHERE Player.TeamID = Team.TeamID
262   )
263   ORDER BY Team.TeamName;
264

```

Output:



```

Query 5

252
253 •   SELECT
254     Team.TeamID,
255     Team.TeamName
256   FROM Team
257   ⊖ WHERE NOT EXISTS (
258       SELECT 1
259       FROM Goal
260       JOIN Player ON Goal.PlayerID = Player.PlayerID
261       WHERE Player.TeamID = Team.TeamID
262   )
263   ORDER BY Team.TeamName;
264
265
100% | 24:263 | 1 error found

Result Grid | Filter Rows: | Search | Edit: | Export/Import: | Result Grid | Form Editor | Field Types |
TeamID | TeamName |
DEN | Denmark |
JAM | Jamaica |
MOR | Morocco |
NGA | Nigeria |
RSA | South Africa |
SUI | Switzerland |
USA | USA |
NULL | NULL |

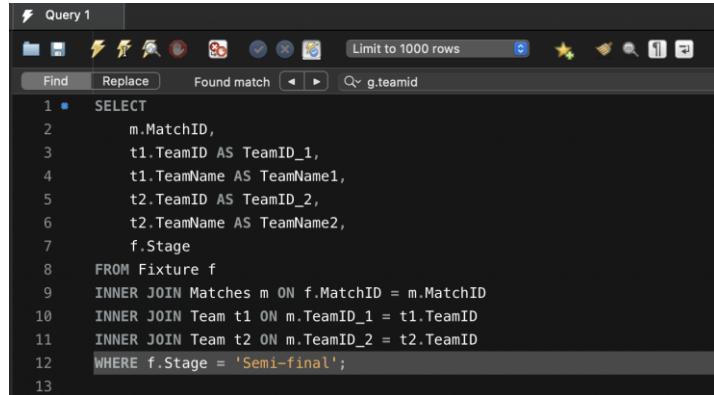
Team 160 | Apply | Revert |

Action Output | Time | Action | Response | Duration / Fetch Time
462 19:09:06 | SELECT Team.TeamID, Team.TeamName FROM Team WHERE NOT EXISTS ( | SELECT 1... | 7 row(s) returned | 0.0055 sec / 0.00000...

```

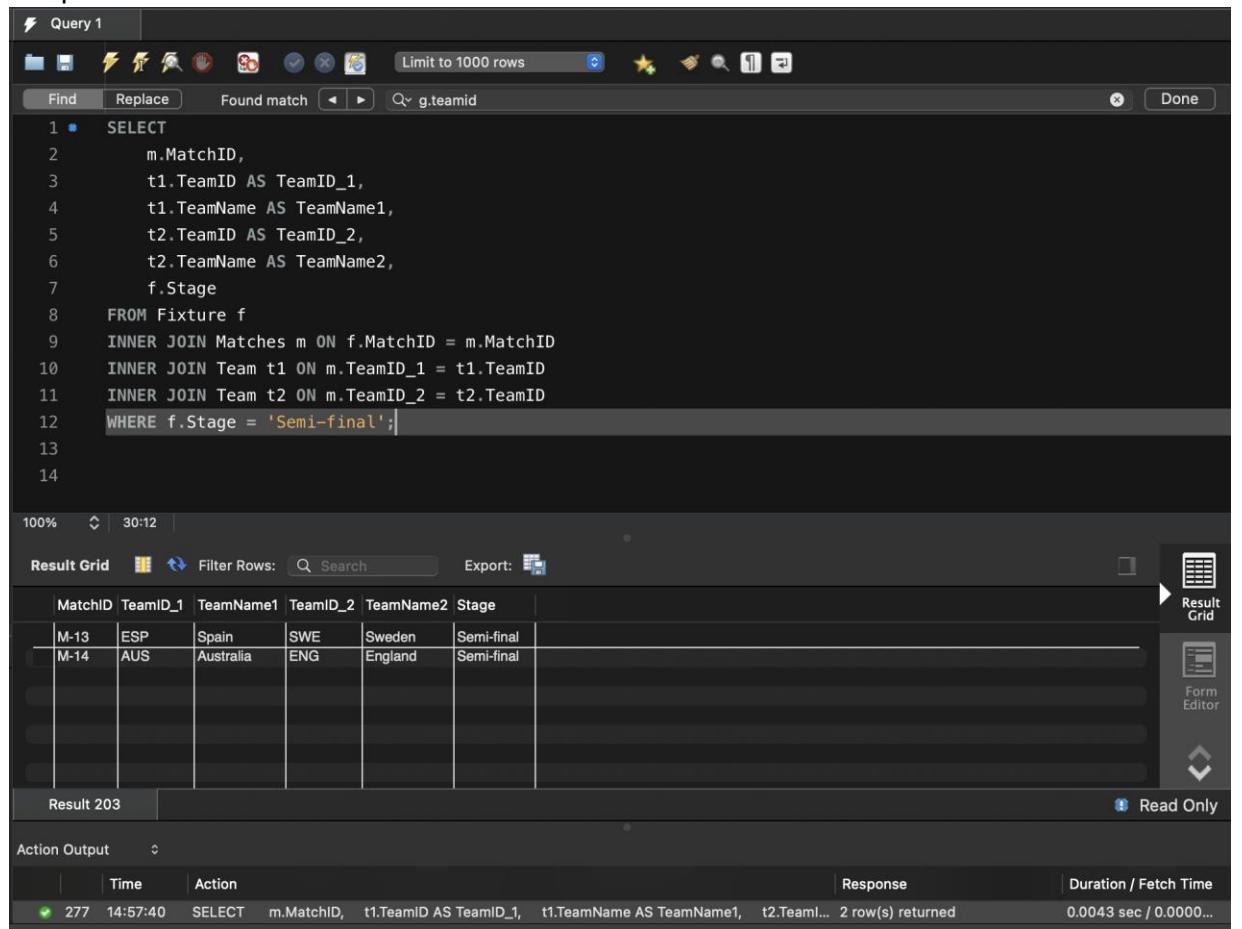
5. Query Description: To Retrieve the Teams which have been qualified for semifinals.

SQL Statement:



```
Query 1
SELECT
    m.MatchID,
    t1.TeamID AS TeamID_1,
    t1.TeamName AS TeamName1,
    t2.TeamID AS TeamID_2,
    t2.TeamName AS TeamName2,
    f.Stage
FROM Fixture f
INNER JOIN Matches m ON f.MatchID = m.MatchID
INNER JOIN Team t1 ON m.TeamID_1 = t1.TeamID
INNER JOIN Team t2 ON m.TeamID_2 = t2.TeamID
WHERE f.Stage = 'Semi-final';
```

Output:



```
Query 1
SELECT
    m.MatchID,
    t1.TeamID AS TeamID_1,
    t1.TeamName AS TeamName1,
    t2.TeamID AS TeamID_2,
    t2.TeamName AS TeamName2,
    f.Stage
FROM Fixture f
INNER JOIN Matches m ON f.MatchID = m.MatchID
INNER JOIN Team t1 ON m.TeamID_1 = t1.TeamID
INNER JOIN Team t2 ON m.TeamID_2 = t2.TeamID
WHERE f.Stage = 'Semi-final';
```

Result Grid

| MatchID | TeamID_1 | TeamName1 | TeamID_2 | TeamName2 | Stage |
|---------|----------|-----------|----------|-----------|------------|
| M-13 | ESP | Spain | SWE | Sweden | Semi-final |
| M-14 | AUS | Australia | ENG | England | Semi-final |

Result 203

Action Output

| Time | Action | Response | Duration / Fetch Time |
|--------------|--|-------------------|------------------------|
| 277 14:57:40 | SELECT m.MatchID, t1.TeamID AS TeamID_1, t1.TeamName AS TeamName1, t2.TeamID AS TeamID_2, t2.TeamName AS TeamName2, f.Stage FROM Fixture f INNER JOIN Matches m ON f.MatchID = m.MatchID INNER JOIN Team t1 ON m.TeamID_1 = t1.TeamID INNER JOIN Team t2 ON m.TeamID_2 = t2.TeamID WHERE f.Stage = 'Semi-final'; | 2 row(s) returned | 0.0043 sec / 0.0000... |

6. Query Description: Listing the total numbers of goals scored in each match from highest to lowest.

SQL statement:

```

Query 1
58
59 •   SELECT
60     distinct (m.MatchId),
61     m.TeamID_1,
62     m.TeamID_2,
63     COUNT(g.goalID) OVER(PARTITION BY g.MatchId) AS TotalGoals
64   FROM Matches m
65   INNER JOIN Goal g ON m.MatchId = g.matchid
66   order by TotalGoals desc;
67

```

Output:

```

Query 5
1 •   SELECT
2     distinct (m.MatchId),
3     m.TeamID_1,
4     m.TeamID_2,
5     COUNT(g.goalID) OVER(PARTITION BY g.MatchId) AS TotalGoals
6   FROM Matches m
7   INNER JOIN Goal g ON m.MatchId = g.matchid
8   ORDER BY TotalGoals DESC;

100% 26:8
Result Grid Filter Rows: Search Export: 
Result Grid
Form Editor
Field Types
Query Stats
Read Only

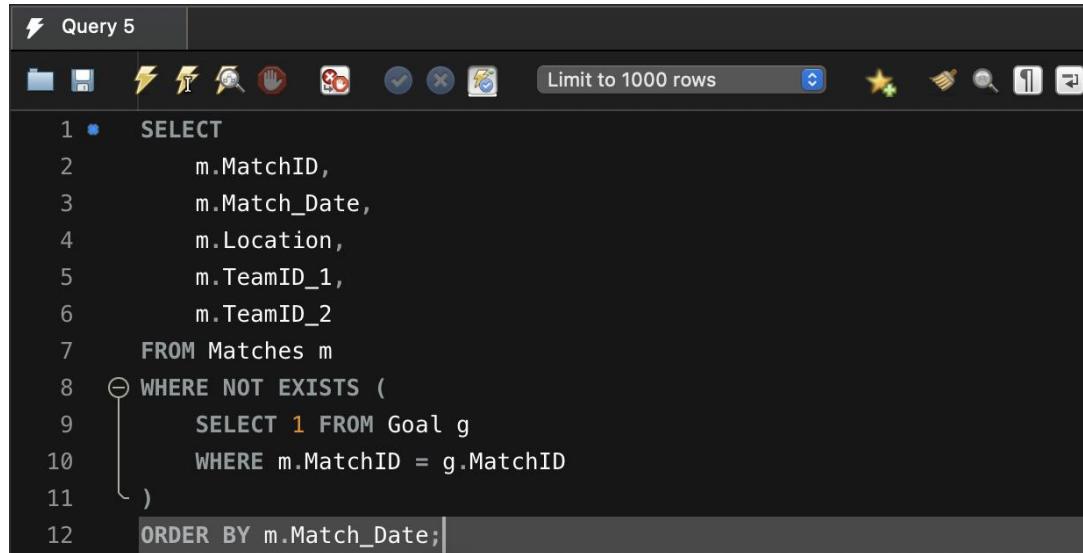
Action Output
Time Action Response Duration / Fetch Time
305 15:53:28 SELECT distinct (m.MatchId), m.TeamID_1, m.TeamID_2, COUNT(g.goalID) OVER(PARTITION... 13 row(s) returned 0.0047 sec / 0.00000...

```

| MatchId | TeamID_1 | TeamID_2 | TotalGoals |
|---------|----------|----------|------------|
| M-1 | SUI | ESP | 5 |
| M-14 | AUS | ENG | 4 |
| M-2 | JPN | NOR | 4 |
| M-7 | FRA | MOR | 4 |
| M-10 | JPN | SWE | 3 |
| M-12 | ENG | COL | 3 |
| M-13 | ESP | SWE | 3 |
| M-9 | ESP | NED | 3 |
| M-15 | SWE | AUS | 2 |
| M-3 | NED | RSA | 2 |
| M-5 | AUS | DEN | 2 |
| M-16 | ESP | ENG | 1 |
| M-8 | COL | JAM | 1 |

7. Query Description: To Retrieve the Match details where teams have not scored any goals.

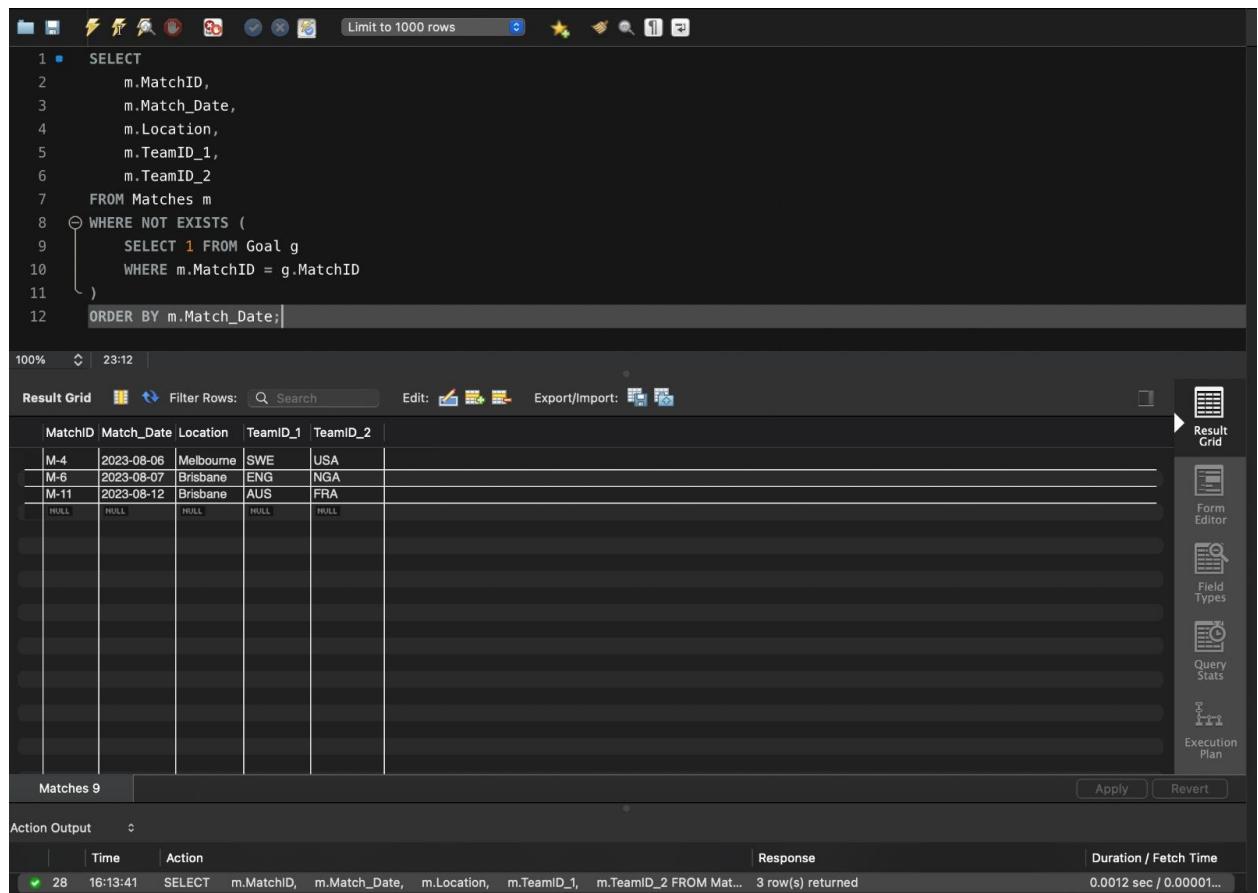
SQL Statement:



```

    ⚡ Query 5
    ⌂ SELECT
        2     m.MatchID,
        3     m.Match_Date,
        4     m.Location,
        5     m.TeamID_1,
        6     m.TeamID_2
        FROM Matches m
        ⌂ WHERE NOT EXISTS (
        9         SELECT 1 FROM Goal g
        10        WHERE m.MatchID = g.MatchID
        11    )
        ORDER BY m.Match_Date;
    
```

Output:



The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Displays the same SQL query as above.
- Result Grid:** Shows the output of the query, listing three matches:

| MatchID | Match_Date | Location | TeamID_1 | TeamID_2 |
|---------|------------|-----------|----------|----------|
| M-4 | 2023-08-06 | Melbourne | SWE | USA |
| M-6 | 2023-08-07 | Brisbane | ENG | NGA |
| M-11 | 2023-08-12 | Brisbane | AUS | FRA |
- Action Output:** Shows the executed query and its duration:

| Action | Time | Response | Duration / Fetch Time |
|-------------|--|-------------------|-------------------------|
| 28 16:13:41 | SELECT m.MatchID, m.Match_Date, m.Location, m.TeamID_1, m.TeamID_2 FROM Mat... | 3 row(s) returned | 0.0012 sec / 0.00001... |

8. Query Description: Getting the Player names who scored the goal and are positioned as Defenders.

SQL Statement:

```
⚡ Query 5
193
194 •  SELECT
195     Distinct(Player.PlayerID),
196     Player.FirstName,
197     Player.LastName,
198     Player.Position
199   FROM Player
200   JOIN Goal ON Player.PlayerID = Goal.PlayerID
201 WHERE Player.Position = 'Defender';
202
```

Output:

```
⚡ Query 5
193
194 •  SELECT
195     Distinct(Player.PlayerID),
196     Player.FirstName,
197     Player.LastName,
198     Player.Position
199   FROM Player
200   JOIN Goal ON Player.PlayerID = Goal.PlayerID
201 WHERE Player.Position = 'Defender';
202
```

100% 36:201 Refresh data re-executing the original query

Result Grid | Filter Rows: | Search | Export: |

| PlayerID | FirstName | LastName | Position |
|----------|-----------|---------------|----------|
| ESP3 | Laia | Codina | Defender |
| ESP4 | Olga | Carmona | Defender |
| JPN10 | Saki | KUMAGAI | Defender |
| NED11 | Stefanie | van der Graaf | Defender |
| SWE3 | Arnanda | Ilestedt | Defender |

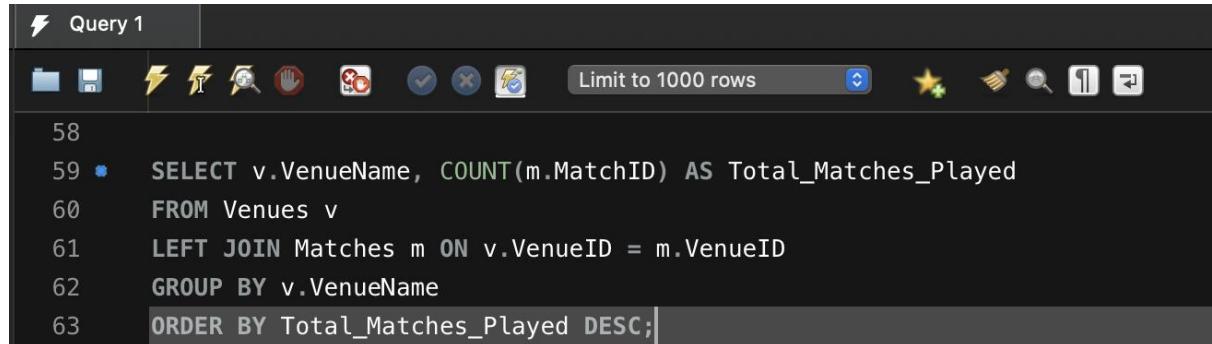
Result 153 | Read Only

Action Output | Time | Action | Response | Duration / Fetch Time

| | | | | |
|-----|----------|---|-------------------|-------------------------|
| 455 | 18:44:18 | SELECT Distinct(Player.PlayerID), Player.FirstName, Player.LastName, Player.Position... | 5 row(s) returned | 0.0029 sec / 0.00000... |
|-----|----------|---|-------------------|-------------------------|

9. Query Description: To get the total number of matches played in each Venues in the order of most played to least played.

SQL Statement:

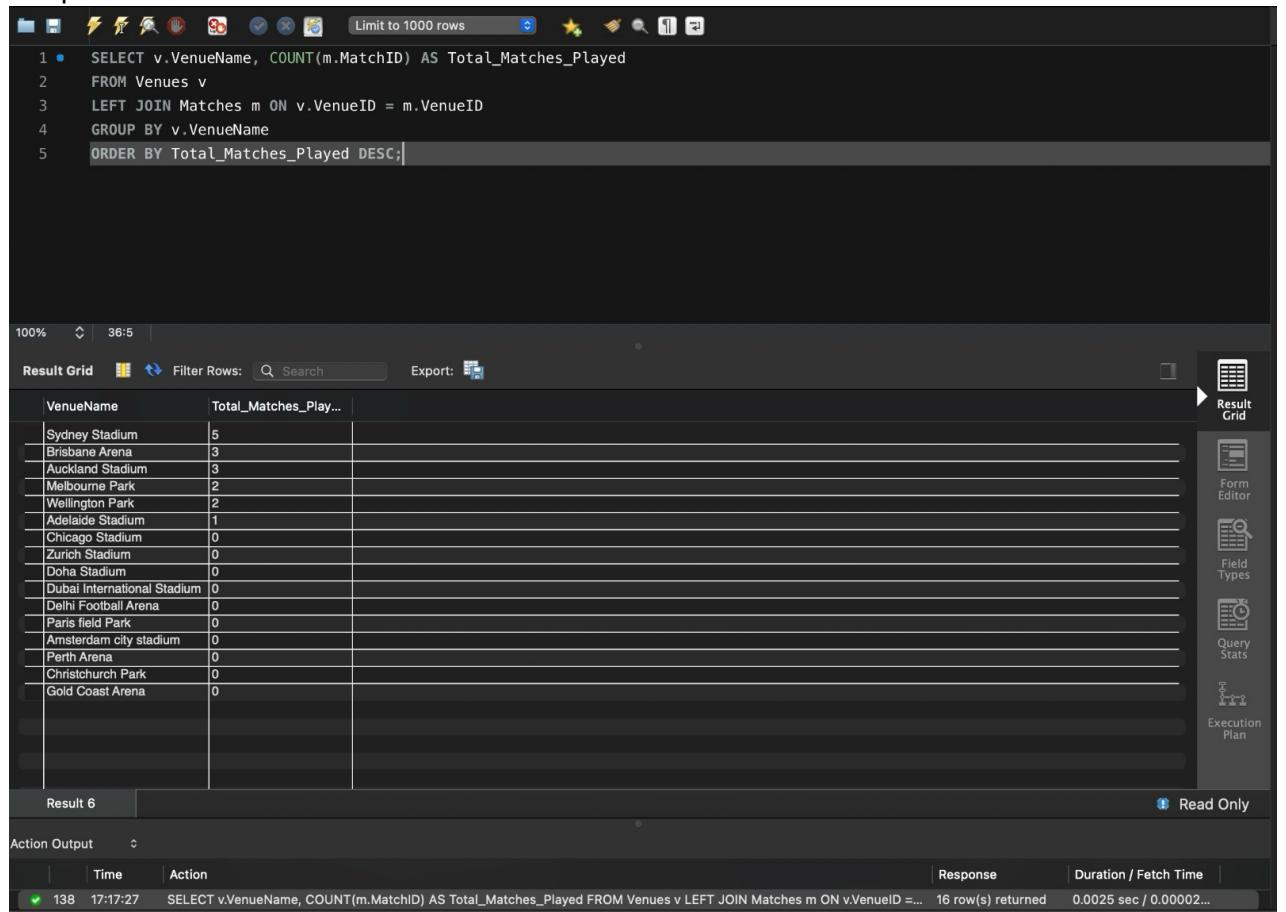


```

58
59 •  SELECT v.VenueName, COUNT(m.MatchID) AS Total_Matches_Played
60   FROM Venues v
61   LEFT JOIN Matches m ON v.VenueID = m.VenueID
62   GROUP BY v.VenueName
63   ORDER BY Total_Matches_Played DESC;

```

Output:



| VenueName | Total_Matches_Played |
|-----------------------------|----------------------|
| Sydney Stadium | 5 |
| Brisbane Arena | 3 |
| Auckland Stadium | 3 |
| Melbourne Park | 2 |
| Wellington Park | 2 |
| Adelaide Stadium | 1 |
| Chicago Stadium | 0 |
| Zurich Stadium | 0 |
| Doha Stadium | 0 |
| Dubai International Stadium | 0 |
| Delhi Football Arena | 0 |
| Paris field Park | 0 |
| Amsterdam city stadium | 0 |
| Perth Arena | 0 |
| Christchurch Park | 0 |
| Gold Coast Arena | 0 |

Result 6 | Read Only

Action Output | Time | Action | Response | Duration / Fetch Time

138 17:17:27 SELECT v.VenueName, COUNT(m.MatchID) AS Total_Matches_Played FROM Venues v LEFT JOIN Matches m ON v.VenueID =... 16 row(s) returned 0.0025 sec / 0.00002...

10. Query description: Query to list players by their first and last names, count how many goals each has scored, and rank them based on the number of goals scored.
SQL statement:

```
⚡ Query 1
SELECT
    p.FirstName,
    p.LastName,
    COUNT(g.GoalID) AS TotalGoals,
    RANK() OVER (ORDER BY COUNT(g.GoalID) DESC) AS Player_Rank_By_Goals
FROM Player p
LEFT JOIN Goal g ON p.playerId = g.playerId
GROUP BY p.FirstName, p.LastName
ORDER BY TotalGoals DESC;
```

Output:

```
⚡ Query 5
SELECT
    p.FirstName,
    p.LastName,
    COUNT(g.GoalID) AS TotalGoals,
    RANK() OVER (ORDER BY COUNT(g.GoalID) DESC) AS Player_Rank_By_Goals
FROM Player p
LEFT JOIN Goal g ON p.playerId = g.playerId
GROUP BY p.FirstName, p.LastName
ORDER BY TotalGoals DESC;
```

Result Grid

| FirstName | LastName | TotalGoals | Player_Rank_By_Goals |
|-----------|---------------|------------|----------------------|
| Alessia | Russo | 2 | 1 |
| Lauren | Hemp | 2 | 1 |
| Salma | Paralluelo | 2 | 1 |
| Olga | Carmona | 2 | 1 |
| Jennifer | Hermoso | 2 | 1 |
| Sam | Kerr | 1 | 6 |
| Caitlin | Foord | 1 | 6 |
| Emily | van Egmond | 1 | 6 |
| Leicy | Santos | 1 | 6 |
| Catalina | Usme | 1 | 6 |
| Ella | Toone | 1 | 6 |
| Alba | Redondo | 1 | 6 |
| Laia | Codina | 1 | 6 |
| Aitana | Bonmatí | 1 | 6 |
| Mariona | Caldentey | 1 | 6 |
| Pauline | Peyraud-Ma... | 1 | 6 |

Result 15

Action Output

| Time | Action | Response | Duration / Fetch Time |
|--------------|--|---------------------|-------------------------|
| 307 15:55:45 | SELECT p.FirstName, p.LastName, COUNT(g.GoalID) AS TotalGoals, RANK() OVER (ORD... | 177 row(s) returned | 0.0060 sec / 0.00001... |

100% | 26:9

Result Grid Filter Rows: Search Export:

| FirstName | LastName | TotalGoals | Player_Rank_By_Goals |
|------------|----------------|------------|----------------------|
| Pauline | Peyraud-Ma... | 1 | 6 |
| Kensa | Dali | 1 | 6 |
| Eugénie | Le Sommer | 1 | 6 |
| Kadidiatou | Diani | 1 | 6 |
| Saki | KUMAGAI | 1 | 6 |
| Hinata | Miyazawa | 1 | 6 |
| Honoka | Hayashi | 1 | 6 |
| Stefanie | van der Gragt | 1 | 6 |
| Jill | Roord | 1 | 6 |
| Lineth | Beerensteyn | 1 | 6 |
| Guro | Reiten | 1 | 6 |
| Ingrid | Syrstad Eng... | 1 | 6 |
| Filippa | Angeldal | 1 | 6 |
| Amanda | Ilestedt | 1 | 6 |
| Kosovare | Asllani | 1 | 6 |
| Rebecka | Blomqvist | 1 | 6 |
| Fridolina | Rolfö | 1 | 6 |
| Lydia | Williams | 0 | 33 |
| Ellie | Carpenter | 0 | 33 |
| Alanna | Kennedy | 0 | 33 |
| Steph | Catley | 0 | 33 |
| Clare | Polkinghorne | 0 | 33 |
| Tameka | Yallop | 0 | 33 |
| Katrina | Gorry | 0 | 33 |

Result 15

Result Grid Filter Rows: Search Export:

| FirstName | LastName | TotalGoals | Player_Rank_By_Goals |
|-----------|---------------|------------|----------------------|
| Katrina | Gorry | 0 | 33 |
| Alex | Chidiac | 0 | 33 |
| Catalina | Pérez | 0 | 33 |
| Maria | Camila Reyes | 0 | 33 |
| Mayra | Ramírez | 0 | 33 |
| Daniela | Arias | 0 | 33 |
| Jorelyn | Carabali | 0 | 33 |
| Manuela | Vanegas | 0 | 33 |
| Daniela | Montoya | 0 | 33 |
| Carolina | Arias | 0 | 33 |
| Linda | Caicedo | 0 | 33 |
| Katrín | Larsen | 0 | 33 |
| Karen | Holmgaard | 0 | 33 |
| Sofie | Svava | 0 | 33 |
| Rikke | Sevecke | 0 | 33 |
| Simone | Boye Søren... | 0 | 33 |
| Katrine | Veje | 0 | 33 |
| Nicoline | Sorensen | 0 | 33 |
| Emma | Snerle | 0 | 33 |
| Pernille | Harder | 0 | 33 |
| Marie | Madsen | 0 | 33 |
| Signe | Bruun | 0 | 33 |
| Mary | Earps | 0 | 33 |
| Georgia | Stanway | 0 | 33 |

Result 15

Result Grid Filter Rows: Search Export:

| FirstName | LastName | TotalGoals | Player_Rank_By_Goals |
|-----------|-----------|------------|----------------------|
| Georgia | Stanway | 0 | 33 |
| Alex | Greenwood | 0 | 33 |
| Lucy | Bronze | 0 | 33 |
| Millie | Bright | 0 | 33 |
| Niamh | Charles | 0 | 33 |
| Keira | Walsh | 0 | 33 |
| Chloe | Kelly | 0 | 33 |
| Misa | Rodriguez | 0 | 33 |
| Neil | Cook | 0 | 33 |
| Irene | Paredes | 0 | 33 |
| Maria | Perez | 0 | 33 |
| Alexia | Putellas | 0 | 33 |
| Sakina | Karchaoui | 0 | 33 |
| Wendie | Renard | 0 | 33 |
| Estelle | Cascarino | 0 | 33 |
| Sandie | Toletti | 0 | 33 |
| Amel | Majri | 0 | 33 |
| Grace | Geyoro | 0 | 33 |
| Viviane | Asseyi | 0 | 33 |
| Sydney | Schneider | 0 | 33 |
| Drew | Spenser | 0 | 33 |
| Vyan | Sampson | 0 | 33 |
| Chantelle | Swaby | 0 | 33 |
| Deneisha | Blackwood | 0 | 33 |

Result 15

Result Grid Filter Rows: Search Export:

| FirstName | LastName | TotalGoals | Player_Rank_By_Goals |
|-----------|------------|------------|----------------------|
| Deneisha | Blackwood | 0 | 33 |
| Konya | Plummer | 0 | 33 |
| Havana | Solaun | 0 | 33 |
| Solai | Washington | 0 | 33 |
| Kameron | Simmonds | 0 | 33 |
| Khadija | Shaw | 0 | 33 |
| Cheyne | Matthews | 0 | 33 |
| Ayaka | Yamashita | 0 | 33 |
| Risa | Shimizu | 0 | 33 |
| Hikaru | Naomoto | 0 | 33 |
| Hina | Sugita | 0 | 33 |
| Yui | Hasegawa | 0 | 33 |
| Maika | Hamano | 0 | 33 |
| Riko | Ueki | 0 | 33 |
| Mina | Tanaka | 0 | 33 |
| Assia | Zouhair | 0 | 33 |
| Anissa | Lahmari | 0 | 33 |
| Yasim | Mrabet | 0 | 33 |
| Zineb | Redouani | 0 | 33 |
| Nauhaila | Benzina | 0 | 33 |
| Nesryne | El Chad | 0 | 33 |
| Sarah | Kassi | 0 | 33 |
| Salma | Amani | 0 | 33 |
| Ibtissam | Iraidi | 0 | 33 |

Result 15

Result Grid Filter Rows: Search Export:

| FirstName | LastName | TotalGoals | Player_Rank_By_Goals |
|-----------|---------------|------------|----------------------|
| Ibtissam | Jraidi | 0 | 33 |
| Fatima | Gharbi | 0 | 33 |
| Fatima | Tagnaout | 0 | 33 |
| Daphne | van Domselaar | 0 | 33 |
| Daniëlle | van de Donk | 0 | 33 |
| Dominique | Janssen | 0 | 33 |
| Aniek | Nouwen | 0 | 33 |
| Merel | van Dongen | 0 | 33 |
| Sherida | Spitse | 0 | 33 |
| Lieke | Martens | 0 | 33 |
| Renate | Jansen | 0 | 33 |
| Tochukwu | Oluwahi | 0 | 33 |
| Christy | Ucheibe | 0 | 33 |
| Evelyn | Nwabuoku | 0 | 33 |
| Osinachi | Ohale | 0 | 33 |
| Michelle | Alozie | 0 | 33 |
| Onome | Ebi | 0 | 33 |
| Halimatou | Ayinde | 0 | 33 |
| Toni | Payne | 0 | 33 |
| Asisat | Oshoala | 0 | 33 |
| Desire | Oparanozie | 0 | 33 |
| Francisca | Ordega | 0 | 33 |
| Maren | Mjelde | 0 | 33 |
| Karina | Saevik | 0 | 33 |

Result 15

Result Grid Filter Rows: Search Export:

| FirstName | LastName | TotalGoals | Player_Rank_By_Goals |
|-----------|---------------|------------|----------------------|
| Karina | Saevik | 0 | 33 |
| Emilie | Haavi | 0 | 33 |
| Caroline | Graham Hansen | 0 | 33 |
| Ada | Hegerberg | 0 | 33 |
| Anja | Sonstevold | 0 | 33 |
| Cecilie | Fiskerstrand | 0 | 33 |
| Vilde | Bøe Risa | 0 | 33 |
| Sara | Horte | 0 | 33 |
| Andile | Dlamini | 0 | 33 |
| Kholosa | Biyana | 0 | 33 |
| Thembi | Kgatlana | 0 | 33 |
| Lebohang | Ramalepe | 0 | 33 |
| Noko | Matlou | 0 | 33 |
| Bambanani | Mbane | 0 | 33 |
| Fikile | Magama | 0 | 33 |
| Refiloe | Jane | 0 | 33 |
| Mapula | Kgoale | 0 | 33 |
| Linda | Mothlalo | 0 | 33 |
| Hildah | Magaia | 0 | 33 |
| Gaelle | Thalmann | 0 | 33 |
| Alisha | Lehmann | 0 | 33 |
| Fabienne | Humm | 0 | 33 |
| Noëlle | Maritz | 0 | 33 |
| Julia | Stierli | 0 | 33 |

Result 15

Result Grid Filter Rows: Search Export:

| FirstName | LastName | TotalGoals | Player_Rank_By_Goals |
|-----------|--------------|------------|----------------------|
| Luana | Bühler | 0 | 33 |
| Lia | Wälti | 0 | 33 |
| Coumba | Sow | 0 | 33 |
| Amira | Arfaoui | 0 | 33 |
| Ana-Maria | Crnogorčević | 0 | 33 |
| Ramona | Bachmann | 0 | 33 |
| Zecira | Musovic | 0 | 33 |
| Jonna | Andersson | 0 | 33 |
| Magdalena | Eriksson | 0 | 33 |
| Linda | Sembrant | 0 | 33 |
| Caroline | Seger | 0 | 33 |
| Sofia | Jakobsson | 0 | 33 |
| Alyssa | Naeher | 0 | 33 |
| Alex | Morgan | 0 | 33 |
| Megan | Rapinoe | 0 | 33 |
| Crystal | Dunn | 0 | 33 |
| Alana | Cook | 0 | 33 |
| Emily | Fox | 0 | 33 |
| Kelley | O'Hara | 0 | 33 |
| Rose | Lavelle | 0 | 33 |
| Ashley | Sanchez | 0 | 33 |
| Julie | Ertz | 0 | 33 |
| Lynn | Williams | 0 | 33 |

Result 15

11. Query description: Retrieving top 5 teams using Dense_Rank() that scored maximum goals in a single match.

SQL statement:

Query 1

```

58
59 •  SELECT
60     g.ScoredforTeamID AS TeamID,
61     m.MatchID,
62     COUNT(g.goalID) AS Total_Goals,
63     DENSE_RANK() OVER (ORDER BY COUNT(g.goalID) DESC) AS Team_rank
64     FROM Goal g
65     JOIN Matches m ON g.matchID = m.matchID
66     GROUP BY g.ScoredforTeamID, m.matchID
67     ORDER BY Total_Goals DESC
68     LIMIT 5;
69

```

Output:

The screenshot shows the MySQL Workbench interface with a query editor and a result grid.

```

1 •   SELECT
2     g.ScoredforTeamID as TeamID,
3     m.MatchID,
4     COUNT(g.goalID) AS Total_Goals,
5     DENSE_RANK() OVER (ORDER BY COUNT(g.goalID) DESC) AS Team_rank
6   FROM Goal g
7   JOIN Matches m ON g.matchID = m.matchID
8   GROUP BY g.ScoredforTeamID, m.matchID
9   ORDER BY Total_Goals DESC
10  LIMIT 5;
  
```

Result Grid

| TeamID | MatchID | Total_Goals | Team_rank |
|--------|---------|-------------|-----------|
| FRA | M-7 | 4 | 1 |
| ESP | M-1 | 4 | 1 |
| ENG | M-14 | 3 | 2 |
| JPN | M-2 | 3 | 2 |
| NED | M-3 | 2 | 3 |

Action Output

| Action | Time | Response | Duration / Fetch Time |
|--|------|----------|-------------------------|
| 314 16:04:57 SELECT g.ScoredforTeamID as TeamID, m.MatchID, COUNT(g.goalID) AS Total_Goals, DENSE... 5 row(s) returned | | | 0.0017 sec / 0.00000... |

12. Query description: Getting the List of player details who scored a Self-Goal.
(Self-goal is scoring a goal in their own team)

SQL statement:

The screenshot shows the MySQL Workbench interface with a query editor and a result grid.

```

80
81 •   SELECT
82     p.playerid,
83     p.firstname,
84     p.lastname,
85     g.goalid,
86     g.matchid,
87     'Self Goal' as goal_type
88   FROM player p
89   JOIN goal g ON p.playerid = g.playerid
90   JOIN matches m ON g.matchid = m.matchid
91   JOIN team t1 ON m.teamid_1 = t1.teamid
92   JOIN team t2 ON m.teamid_2 = t2.teamid
93   WHERE (m.teamid_1 = g.ScoredforTeamID AND p.teamid = m.teamid_2)
94     OR (m.teamid_2 = g.ScoredforTeamID AND p.teamid = m.teamid_1);
  
```

Output:

```

1 • SELECT
2     p.playerid,
3     p.firstname,
4     p.lastname,
5     g.goalid,
6     g.matchid,
7     'Self Goal' as goal_type
8 FROM player p
9 JOIN goal g ON p.playerid = g.playerid
10 JOIN matches m ON g.matchid = m.matchid
11 JOIN team t1 ON m.teamid_1 = t1.teamid
12 JOIN team t2 ON m.teamid_2 = t2.teamid
13 WHERE (m.teamid_1 = g.ScoredForTeamID AND p.teamid = m.teamid_2)
14     OR (m.teamid_2 = g.ScoredForTeamID AND p.teamid = m.teamid_1);

```

Result Grid

| playerid | firstname | lastname | goalid | matchid | goal_type |
|----------|-----------|---------------|--------|---------|-----------|
| ESP7 | Aitana | Bonmatí | G-1 | M-1 | Self Goal |
| NOR5 | Ingrid | Syrstad Engen | G-6 | M-2 | Self Goal |

Action Output

| | Time | Action | Response | Duration / Fetch Time |
|-----|----------|--|----------|------------------------|
| 315 | 16:05:56 | SELECT p.playerid, p.firstname, p.lastname, g.goalid, g.matchid, 'Self Goal' as goal_ty... 2 row(s) returned | | 0.043 sec / 0.00016... |

13. Query description: Retrieving the player details who scored against top 3 FIFA ranking teams.

SQL Statement:

```

37
38 • WITH TopTeams AS (
39     SELECT TeamID
40     FROM Team
41     ORDER BY Fifa_Ranking ASC
42     LIMIT 3)
43     SELECT
44         g.goalid,
45         g.matchid,
46         p.firstname,
47         p.lastname,
48         g.ScoredForTeamID
49     FROM Goal g
50     JOIN Matches m ON g.matchid = m.matchid
51     JOIN Player p ON g.playerid = p.playerid
52     JOIN TopTeams tt ON (m.teamid_1 = tt.teamid AND g.ScoredforTeamID = m.teamid_2)
53         OR (m.teamid_2 = tt.teamid AND g.ScoredforTeamID = m.teamid_1);

```

Output:

The screenshot shows a database query editor interface with the following details:

- Query 1:** The current query being run.
- Toolbar:** Includes icons for file operations, search, and navigation.
- Search Bar:** Shows "Found match" and a search term "g.teamid".
- Code Area:** Displays the SQL query with line numbers. The query uses common table expressions (CTEs) to find top teams and then joins them with goals, matches, and players to get the names of the scorers.
- Result Grid:** A table showing the results of the query. It has columns: goalid, matchid, firstname, lastname, and ScoredForTeamID. The data is as follows:

| goalid | matchid | firstname | lastname | ScoredForTeamID |
|--------|---------|-----------|---------------|-----------------|
| G-1 | M-1 | Aitana | Bonmatí | SUI |
| G-30 | M-13 | Rebecka | Björnqvist | SWE |
| G-21 | M-9 | Stefanie | van der Graat | NED |

- Action Output:** Shows the execution details: 258 rows, 18:48:41 timestamp, 3 rows returned, and a duration of 0.0024 sec / 0.000000.

14. Query Description: Getting the summary of goals based on location and the team for which the goals were scored.

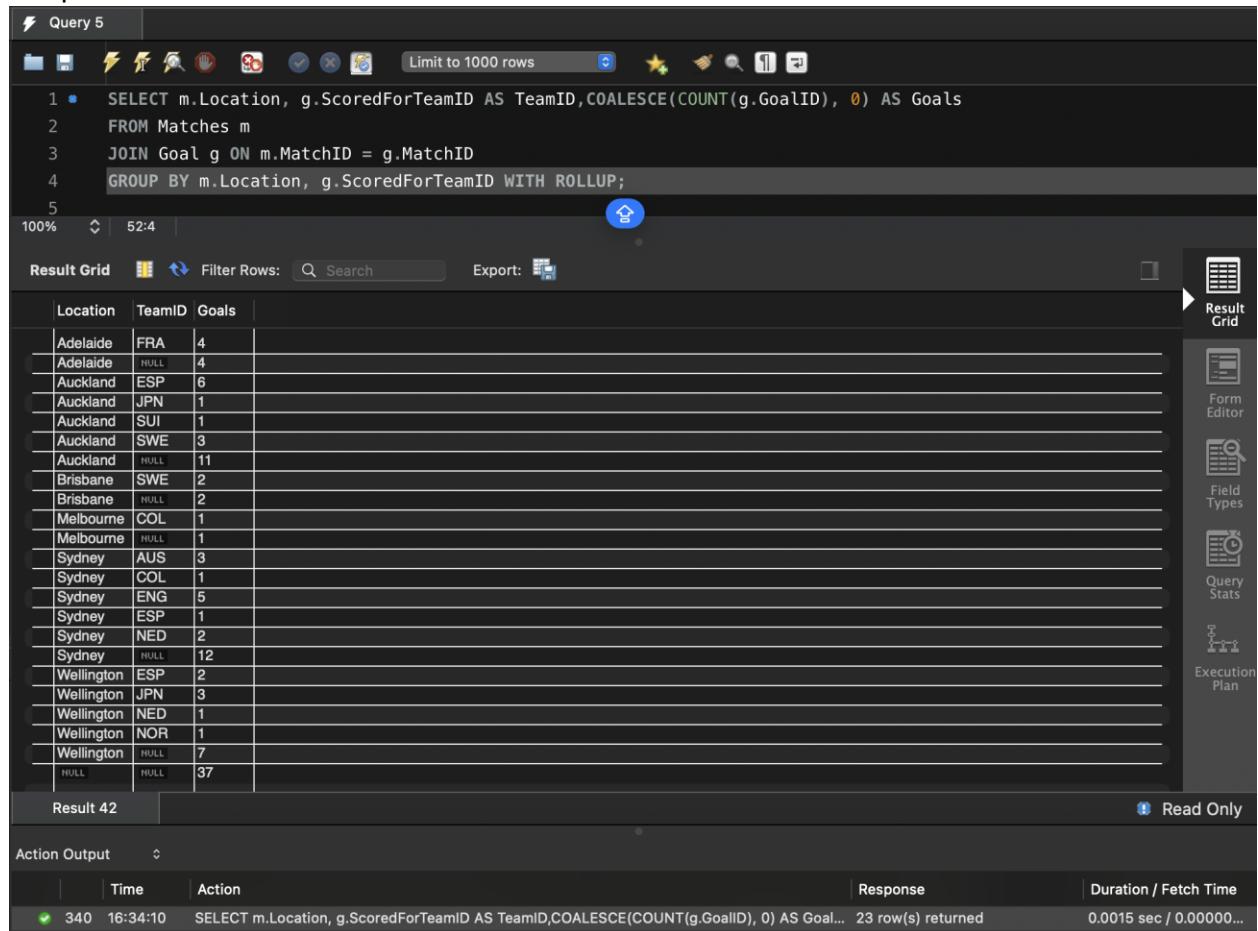
SQL Statement:

The screenshot shows a database query editor interface with the following details:

- Query 5:** The current query being run.
- Toolbar:** Includes icons for file operations, search, and navigation.
- Code Area:** Displays the SQL query to group goals by location and team, using COALESCE for NULL handling and ROLLUP for summary rows.
- Result Grid:** A table showing the results of the query. It has columns: m.Location, g.ScoredForTeamID AS TeamID, and Goals (the count of goals). The data is as follows:

| m.Location | TeamID | Goals |
|------------|--------|-------|
| M-1 | SUI | 1 |
| M-13 | SWE | 1 |
| M-9 | NED | 1 |

Output:



The screenshot shows a database query results interface. At the top, there's a toolbar with various icons for file operations, search, and export. Below the toolbar, the query text is displayed:

```

1 •  SELECT m.Location, g.ScoredForTeamID AS TeamID, COALESCE(COUNT(g.GoalID), 0) AS Goals
2   FROM Matches m
3   JOIN Goal g ON m.MatchID = g.MatchID
4   GROUP BY m.Location, g.ScoredForTeamID WITH ROLLUP;
5

```

The results grid shows the following data:

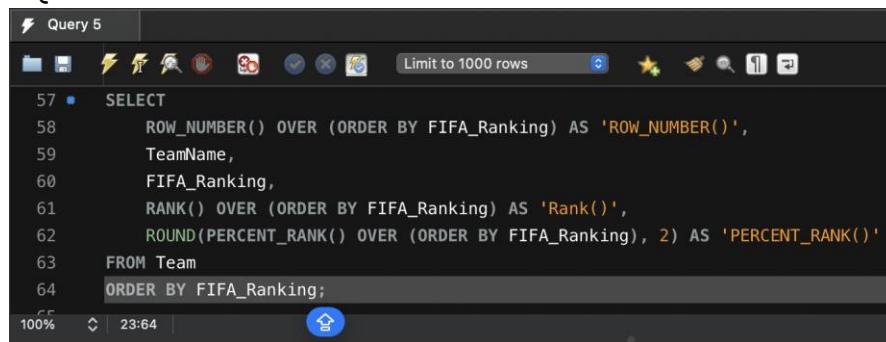
| Location | TeamID | Goals |
|------------|--------|-------|
| Adelaide | FRA | 4 |
| Adelaide | HULL | 4 |
| Auckland | ESP | 6 |
| Auckland | JPN | 1 |
| Auckland | SUI | 1 |
| Auckland | SWE | 3 |
| Auckland | HULL | 11 |
| Brisbane | SWE | 2 |
| Brisbane | HULL | 2 |
| Melbourne | COL | 1 |
| Melbourne | HULL | 1 |
| Sydney | AUS | 3 |
| Sydney | COL | 1 |
| Sydney | ENG | 5 |
| Sydney | ESP | 1 |
| Sydney | NED | 2 |
| Sydney | HULL | 12 |
| Wellington | ESP | 2 |
| Wellington | JPN | 3 |
| Wellington | NED | 1 |
| Wellington | NOR | 1 |
| Wellington | HULL | 7 |
| HULL | HULL | 37 |

Below the table, it says "Result 42" and "Read Only". At the bottom, there's an "Action Output" section with a table showing the execution details:

| Action | Time | Action | Response | Duration / Fetch Time |
|--------|----------|--|--------------------|-------------------------|
| 340 | 16:34:10 | SELECT m.Location, g.ScoredForTeamID AS TeamID, COALESCE(COUNT(g.GoalID), 0) AS Goals... | 23 row(s) returned | 0.0015 sec / 0.00000... |

15. Query Description: Using ROW_NUMBER(), RANK(), and PERCENT_RANK() functions to provide a comprehensive ranking of teams based on their FIFA rankings.

SQL Statement:



The screenshot shows a database query results interface. At the top, there's a toolbar with various icons for file operations, search, and export. Below the toolbar, the query text is displayed:

```

57 •  SELECT
58      ROW_NUMBER() OVER (ORDER BY FIFA_Ranking) AS 'ROW_NUMBER',
59      TeamName,
60      FIFA_Ranking,
61      RANK() OVER (ORDER BY FIFA_Ranking) AS 'Rank',
62      ROUND(PERCENT_RANK() OVER (ORDER BY FIFA_Ranking), 2) AS 'PERCENT_RANK'
63  FROM Team
64  ORDER BY FIFA_Ranking;

```

Output:

```

57 • SELECT
58     ROW_NUMBER() OVER (ORDER BY FIFA_Ranking) AS 'ROW_NUMBER(),
59     TeamName,
60     FIFA_Ranking,
61     RANK() OVER (ORDER BY FIFA_Ranking) AS 'Rank(),
62     ROUND(PERCENT_RANK() OVER (ORDER BY FIFA_Ranking), 2) AS 'PERCENT_RANK()
63 FROM Team
64 ORDER BY FIFA_Ranking;

```

Result Grid Filter Rows: Search Export:

| ROW_NUMBER() | TeamName | FIFA_Ranking | Rank() | PERCENT_RANK() |
|--------------|--------------|--------------|--------|----------------|
| 1 | Spain | 1 | 1 | 0 |
| 2 | USA | 2 | 2 | 0.07 |
| 3 | France | 3 | 3 | 0.13 |
| 4 | England | 4 | 4 | 0.2 |
| 5 | Sweden | 5 | 5 | 0.27 |
| 6 | Netherlands | 7 | 6 | 0.33 |
| 7 | Japan | 8 | 7 | 0.4 |
| 8 | Australia | 12 | 8 | 0.47 |
| 9 | Denmark | 13 | 9 | 0.53 |
| 10 | Norway | 16 | 10 | 0.6 |
| 11 | Switzerland | 22 | 11 | 0.67 |
| 12 | Colombia | 23 | 12 | 0.73 |
| 13 | Nigeria | 34 | 13 | 0.8 |
| 14 | Jamaica | 40 | 14 | 0.87 |
| 15 | South Africa | 52 | 15 | 0.93 |
| 16 | Morocco | 60 | 16 | 1 |

Result 75 Read Only

Action Output

| Time | Action | Response | Duration / Fetch Time |
|--------------|---|----------|-------------------------|
| 375 17:23:55 | SELECT ROW_NUMBER() OVER (ORDER BY FIFA_Ranking) AS 'ROW_NUMBER()', TeamN... 16 row(s) returned | | 0.0041 sec / 0.00000... |

16. Query Description: Calculating a running total of goals for each team, partitioned by the team, and ordered by match date.

SQL Statement:

```

1 • SELECT
2     m.Match_Date,
3     g.ScoredforTeamID AS TeamID,
4     t.TeamName,
5     SUM(1) OVER (
6         PARTITION BY g.ScoredforTeamID
7         ORDER BY m.Match_Date
8         ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW
9     ) AS RunningTotalGoals
10    FROM
11        Goal g
12    JOIN
13        Matches m ON g.MatchID = m.MatchID
14    JOIN
15        Team t ON g.ScoredforTeamID = t.TeamID;

```

Output:

SQL Query:

```

1   SELECT
2     m.Match_Date,
3     g.ScoredforTeamID AS TeamID,
4     t.TeamName,
5     SUM(1) OVER (
6       PARTITION BY g.ScoredforTeamID
7       ORDER BY m.Match_Date
8       ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW
9     ) AS RunningTotalGoals
10    FROM Goal g
11   JOIN Matches m ON g.MatchID = m.MatchID
12   JOIN Team t ON g.ScoredforTeamID = t.TeamID;

```

Action Output:

| Time | Action | Response | Duration / Fetch Time |
|-------------|--|----------|-------------------------|
| 59 17:38:16 | SELECT m.Match_Date, g.ScoredforTeamID AS TeamID, t.TeamName, SUM(1) OVER (PA... 37 row(s) returned | | 0.0011 sec / 0.00000... |

Result Grid:

| Match_Date | TeamID | TeamName | RunningTotalGoals |
|------------|--------|-----------|-------------------|
| 2023-08-07 | AUS | Australia | 1 |
| 2023-08-07 | AUS | Australia | 2 |
| 2023-08-16 | AUS | Australia | 3 |
| 2023-08-08 | COL | Colombia | 1 |
| 2023-08-12 | COL | Colombia | 2 |
| 2023-08-12 | ENG | England | 1 |
| 2023-08-12 | ENG | England | 2 |
| 2023-08-16 | ENG | England | 3 |
| 2023-08-16 | ENG | England | 4 |
| 2023-08-16 | ENG | England | 5 |
| 2023-08-05 | ESP | Spain | 1 |
| 2023-08-05 | ESP | Spain | 2 |
| 2023-08-05 | ESP | Spain | 3 |
| 2023-08-05 | ESP | Spain | 4 |
| 2023-08-11 | ESP | Spain | 5 |
| 2023-08-11 | ESP | Spain | 6 |
| 2023-08-15 | ESP | Spain | 7 |
| 2023-08-16 | ESP | Spain | 8 |
| 2023-08-20 | ESP | Spain | 9 |
| 2023-08-08 | FRA | France | 1 |

Result 14 | Read Only

Action Output:

| Time | Action | Response | Duration / Fetch Time |
|-------------|--|----------|-------------------------|
| 59 17:38:16 | SELECT m.Match_Date, g.ScoredforTeamID AS TeamID, t.TeamName, SUM(1) OVER (PA... 37 row(s) returned | | 0.0011 sec / 0.00000... |

Result Grid:

| Match_Date | TeamID | TeamName | RunningTotalGoals |
|------------|--------|-------------|-------------------|
| 2023-08-20 | ESP | Spain | 9 |
| 2023-08-20 | FRA | France | 1 |
| 2023-08-08 | FRA | France | 2 |
| 2023-08-08 | FRA | France | 3 |
| 2023-08-08 | FRA | France | 4 |
| 2023-08-05 | JPN | Japan | 1 |
| 2023-08-05 | JPN | Japan | 2 |
| 2023-08-05 | JPN | Japan | 3 |
| 2023-08-11 | JPN | Japan | 4 |
| 2023-08-06 | NED | Netherlands | 1 |
| 2023-08-06 | NED | Netherlands | 2 |
| 2023-08-11 | NED | Netherlands | 3 |
| 2023-08-08 | NOR | Norway | 1 |
| 2023-08-05 | SUI | Switzerland | 1 |
| 2023-08-11 | SWE | Sweden | 1 |
| 2023-08-11 | SWE | Sweden | 2 |
| 2023-08-15 | SWE | Sweden | 3 |
| 2023-08-19 | SWE | Sweden | 4 |
| 2023-08-19 | SWE | Sweden | 5 |

Result 14 | Read Only

Action Output:

| Time | Action | Response | Duration / Fetch Time |
|-------------|--|----------|-------------------------|
| 59 17:38:16 | SELECT m.Match_Date, g.ScoredforTeamID AS TeamID, t.TeamName, SUM(1) OVER (PA... 37 row(s) returned | | 0.0011 sec / 0.00000... |

17. Query Description: Displaying each team along with a row number, their total goals, and the cumulative distribution of these goals across teams, with the results ordered from the team with the lowest to the highest total goals.

SQL Statement:

```

Query 5
WITH TeamGoals AS (
    SELECT
        g.ScoredForTeamID AS TeamID,
        COUNT(g.GoalID) AS TotalGoals
    FROM Goal g
    GROUP BY g.ScoredForTeamID
)
SELECT
    ROW_NUMBER() OVER (ORDER BY TotalGoals) AS 'Row_Number(),
    tg.TeamID,
    t.TeamName,
    tg.TotalGoals,
    ROUND(CUME_DIST() OVER (ORDER BY tg.TotalGoals), 2) AS 'CUME_DIST()
FROM TeamGoals tg
JOIN Team t ON tg.TeamID = t.TeamID
ORDER BY tg.TotalGoals;

```

Output:

```

Query 5
WITH TeamGoals AS (
    SELECT g.ScoredForTeamID AS TeamID,COUNT(g.GoalID) AS TotalGoals
    FROM Goal g
    GROUP BY g.ScoredForTeamID
)
SELECT
    ROW_NUMBER() OVER (ORDER BY TotalGoals) AS 'Row_Number(),
    tg.TeamID,t.TeamName,tg.TotalGoals,
    ROUND(CUME_DIST() OVER (ORDER BY tg.TotalGoals), 2) AS 'CUME_DIST()
FROM TeamGoals tg
JOIN Team t ON tg.TeamID = t.TeamID
ORDER BY tg.TotalGoals;

```

| Row_Number() | TeamID | TeamName | TotalGoals | CUME_DIST() |
|--------------|--------|-------------|------------|-------------|
| 1 | NOR | Norway | 1 | 0.2 |
| 2 | SUI | Switzerland | 1 | 0.2 |
| 3 | COL | Colombia | 2 | 0.3 |
| 4 | AUS | Australia | 3 | 0.5 |
| 5 | NED | Netherlands | 3 | 0.5 |
| 6 | FRA | France | 4 | 0.7 |
| 7 | JPN | Japan | 4 | 0.7 |
| 8 | ENG | England | 5 | 0.9 |
| 9 | SWE | Sweden | 5 | 0.9 |
| 10 | ESP | Spain | 9 | 1 |

Result 125 | Read Only

Action Output | Time | Action | Response | Duration / Fetch Time

425 18:09:29 WITH TeamGoals AS (SELECT g.ScoredForTeamID AS TeamID,COUNT(g.GoalID) AS TotalGoals... 10 row(s) returned 0.0056 sec / 0.0000...

18. Query Description: Listing of teams with their total goals in ascending order, a row number and NTILE(4) based on total goals.

SQL Statement:

```
⚡ Query 5
166
167 • ⊖ WITH TeamGoals AS (
168     SELECT g.ScoredForTeamID AS TeamID,COUNT(g.GoalID) AS TotalGoals
169     FROM Goal g
170     GROUP BY g.ScoredForTeamID)
171     SELECT
172         ROW_NUMBER() OVER (ORDER BY tg.TotalGoals) AS 'Row_Number()',*
173         tg.TeamID,t.TeamName,tg.TotalGoals,
174         NTILE(4) OVER (ORDER BY tg.TotalGoals) AS 'NTILE(4)'
175     FROM TeamGoals tg
176     JOIN Team t ON tg.TeamID = t.TeamID
177     ORDER BY tg.TotalGoals;
178

100% ⚡ | 24:177 |
```

Output:

```
⚡ Query 5
166
167 • ⊖ WITH TeamGoals AS (
168     SELECT g.ScoredForTeamID AS TeamID,COUNT(g.GoalID) AS TotalGoals
169     FROM Goal g
170     GROUP BY g.ScoredForTeamID)
171     SELECT
172         ROW_NUMBER() OVER (ORDER BY tg.TotalGoals) AS 'Row_Number()',*
173         tg.TeamID,t.TeamName,tg.TotalGoals,
174         NTILE(4) OVER (ORDER BY tg.TotalGoals) AS 'NTILE(4)'
175     FROM TeamGoals tg
176     JOIN Team t ON tg.TeamID = t.TeamID
177     ORDER BY tg.TotalGoals;
178

100% ⚡ | 24:177 |
```

Result Grid Filter Rows: Search Export:

| Row_Number() | TeamID | TeamName | TotalGoals | NTILE(4) |
|--------------|--------|-------------|------------|----------|
| 1 | NOR | Norway | 1 | 1 |
| 2 | SUI | Switzerland | 1 | 1 |
| 3 | COL | Colombia | 2 | 1 |
| 4 | AUS | Australia | 3 | 2 |
| 5 | NED | Netherlands | 3 | 2 |
| 6 | FRA | France | 4 | 2 |
| 7 | JPN | Japan | 4 | 3 |
| 8 | ENG | England | 5 | 3 |
| 9 | SWE | Sweden | 5 | 4 |
| 10 | ESP | Spain | 9 | 4 |

Result 131

Action Output

| Time | Action | Response | Duration / Fetch Time |
|--------------|---------------------|---|--------------------------|
| 431 18:20:16 | WITH TeamGoals AS (| SELECT g.ScoredForTeamID AS TeamID,COUNT(g.GoalID) AS TotalG... | 10 row(s) returned |
| | | | 0.052 sec / 0.00014 s... |

19. Query Description: ROLLUP functions is used for generating aggregated data across two levels of grouping i.e. team name and player position, while ensuring that NULL values introduced by ROLLUP for subtotal and grand total rows are sorted to the bottom.

SQL Statement:

```
⚡ Query 5
239
240 •  SELECT
241      Team.TeamName,
242      Player.Position,
243      COUNT(Goal.PlayerID) AS TotalGoals
244  FROM Player
245  JOIN Goal ON Player.PlayerID = Goal.PlayerID
246  JOIN Team ON Player.TeamID = Team.TeamID
247  GROUP BY ROLLUP(Team.TeamName, Player.Position)
248  ORDER BY
249      CASE WHEN Team.TeamName IS NULL THEN 1 ELSE 0 END, Team.TeamName,
250      CASE WHEN Player.Position IS NULL THEN 1 ELSE 0 END, Player.Position;
251
```

Output:

```
⚡ Query 5
240 •  SELECT
241      Team.TeamName,
242      Player.Position,
243      COUNT(Goal.PlayerID) AS TotalGoals
244  FROM Player
245  JOIN Goal ON Player.PlayerID = Goal.PlayerID
246  JOIN Team ON Player.TeamID = Team.TeamID
247  GROUP BY ROLLUP(Team.TeamName, Player.Position)
248  ORDER BY
249      CASE WHEN Team.TeamName IS NULL THEN 1 ELSE 0 END, Team.TeamName,
250      CASE WHEN Player.Position IS NULL THEN 1 ELSE 0 END, Player.Position;
251
```

100% 74:250

Result Grid Filter Rows: Search Export:

| TeamName | Position | TotalGoals |
|-----------|----------------------|------------|
| Australia | Forward | 2 |
| Australia | Midfielder | 1 |
| Australia | NULL | 3 |
| Colombia | Forward | 1 |
| Colombia | Midfielder | 1 |
| Colombia | NULL | 2 |
| England | Forward | 4 |
| England | Midfielder | 1 |
| England | NULL | 5 |
| France | Forward | 2 |
| France | Goalkee... ...ler | 1 |
| France | Midfielder | 1 |
| France | NULL | 4 |

Result 157 Read Only

Action Output

| Time | Action | Response | Duration / Fetch Time |
|--------------|--|--------------------|-------------------------|
| 459 18:54:14 | SELECT Team.TeamName, Player.Position, COUNT(Goal.PlayerID) AS TotalGoals FRO... | 31 row(s) returned | 0.0041 sec / 0.00000... |

Query 5

```

240 •  SELECT
241      Team.TeamName,
242      Player.Position,
243      COUNT(Goal.PlayerID) AS TotalGoals|
244  FROM Player
100%   39:243

```

Result Grid Filter Rows: Search Export:

| TeamName | Position | TotalGoals |
|-------------|------------|------------|
| England | NULL | 5 |
| France | Forward | 2 |
| France | Goalkee... | 1 |
| France | Midfielder | 1 |
| France | NULL | 4 |
| Japan | Defender | 1 |
| Japan | Midfielder | 2 |
| Japan | NULL | 3 |
| Netherlands | Defender | 1 |
| Netherlands | Forward | 1 |
| Netherlands | Midfielder | 1 |
| Netherlands | NULL | 3 |
| Norway | Midfielder | 2 |
| Norway | NULL | 2 |
| Spain | Defender | 3 |
| Spain | Forward | 6 |
| Spain | Midfielder | 1 |
| Spain | NULL | 10 |
| Sweden | Defender | 1 |
| Sweden | Forward | 1 |
| Sweden | Midfielder | 3 |
| Sweden | NULL | 5 |
| NULL | NULL | 37 |

Result 157 Read Only

Action Output

| Action | Time | Action | Response | Duration / Fetch Time |
|--------|----------|--|--------------------|--------------------------|
| 459 | 18:54:14 | SELECT Team.TeamName, Player.Position, COUNT(Goal.PlayerID) AS TotalGoals FRO... | 31 row(s) returned | 0.0041 sec / 0.000000... |

20. Query Description: Calculating the moving average for each team based on total goals scored by the team.

SQL Statement:

Query 5

```

268 •  SELECT
269      g.ScoredForTeamID AS TeamID,
270      m.Match_Date,
271      COUNT(g.GoalID) AS GoalsScored,
272      Round(AVG(COUNT(g.GoalID)) OVER (PARTITION BY g.ScoredForTeamID
273      ORDER BY m.Match_Date ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW),2) AS MovingAvgGoals
274  FROM Goal g
275  JOIN Matches m ON g.MatchID = m.MatchID
276  GROUP BY g.ScoredForTeamID, m.MatchID, m.Match_Date
277  ORDER BY g.ScoredForTeamID, m.Match_Date;
278

```

Output:

Query 5

```

268 • SELECT g.ScoredForTeamID AS TeamID, m.Match_Date,
269     COUNT(g.GoalID) AS GoalsScored,
270     Round(AVG(COUNT(g.GoalID))) OVER (PARTITION BY g.ScoredForTeamID
271     ORDER BY m.Match_Date ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW),2) AS MovingAvgGoals
272     FROM Goal g JOIN Matches m ON g.MatchID = m.MatchID
273     GROUP BY g.ScoredForTeamID, m.MatchID, m.Match_Date ORDER BY g.ScoredForTeamID, m.Match_Date;

```

100% 94:273

Result Grid Filter Rows: Search Export:

| TeamID | Match_Date | GoalsScored | MovingAvgGoals |
|--------|------------|-------------|----------------|
| AUS | 2023-08-07 | 2 | 2.00 |
| AUS | 2023-08-16 | 1 | 1.50 |
| COL | 2023-08-08 | 1 | 1.00 |
| COL | 2023-08-12 | 1 | 1.00 |
| ENG | 2023-08-12 | 2 | 2.00 |
| ENG | 2023-08-16 | 3 | 2.50 |
| ESP | 2023-08-05 | 4 | 4.00 |
| ESP | 2023-08-11 | 2 | 3.00 |
| ESP | 2023-08-15 | 2 | 2.67 |
| ESP | 2023-08-20 | 1 | 2.25 |
| FRA | 2023-08-08 | 4 | 4.00 |
| JPN | 2023-08-05 | 3 | 3.00 |
| JPN | 2023-08-11 | 1 | 2.00 |
| NED | 2023-08-06 | 2 | 2.00 |
| NED | 2023-08-11 | 1 | 1.50 |
| NOR | 2023-08-05 | 1 | 1.00 |
| SUI | 2023-08-05 | 1 | 1.00 |
| SWE | 2023-08-11 | 2 | 2.00 |
| SWE | 2023-08-15 | 1 | 1.50 |
| SWE | 2023-08-19 | 2 | 1.67 |

Result 163 Read Only

Action Output

| Time | Action | Response | Duration / Fetch Time |
|----------|---|--------------------|-------------------------|
| 19:19:38 | SELECT g.ScoredForTeamID AS TeamID, m.Match_Date, COUNT(g.GoalID) AS GoalsScored, Round(AVG(COUNT(g.GoalID))) OVER (PARTITION BY g.ScoredForTeamID ORDER BY m.Match_Date ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW),2) AS MovingAvgGoals FROM Goal g JOIN Matches m ON g.MatchID = m.MatchID GROUP BY g.ScoredForTeamID, m.MatchID, m.Match_Date ORDER BY g.ScoredForTeamID, m.Match_Date; | 20 row(s) returned | 0.019 sec / 0.000007... |