1841. League Statistics

Problem Explanation

In this problem, we are given information about football teams and their performance in various matches. We need to generate a statistics table containing information about the teams' performance, such as the number of matches played, points earned, goals scored, goals conceded, and goal difference.

We have two tables:

1. `Teams`: This table contains the team_id and the respective team_name.
2. `Matches`: This table contains information about the matches, such as the home_team_id, away_team_id, home_team_goals, and away_team_goals.

Our goal is to write an SQL query to generate a statistics table with team_name, matches_played, points, goal_for, goal_against, and goal_diff columns.

Let's walk through an example:

Example

Teams:

team_id	team_name
1	Ajax
4	Dortmund
6	Arsenal

Matches:

home_team_id	away_team_id	home_team_goals	away_team_goals
1	4	0	1
1	6	3	3
4	1	5	2
6	1	0	0

Result:

team_name	matches_played	points	goal_for	goal_against	goal_diff
Dortmund	2	6	6	2	4
Arsenal	2	2	3	3	0
Ajax	4	2	5	9	-4

Approach

To generate the required statistics table, we can use the following approach:

1. Calculate matches played, goals scored, and goals conceded for each team in the home and away matches.
2. Combine the home and away statistics for each team.
3. Calculate total points and goal difference for each team.
4. Order the teams by total points, goal difference, and team_name.

Solution

SQL

```sql
WITH home_stats AS (
  SELECT
    home_team_id AS team_id,
    COUNT(*) AS matches_played,
    SUM(home_team_goals) AS goal_for,
    SUM(away_team_goals) AS goal_against,
    SUM(CASE
      WHEN home_team_goals > away_team_goals THEN 3
      WHEN home_team_goals = away_team_goals THEN 1
      ELSE 0
    END) AS points
  FROM
    Matches
  GROUP BY
    home_team_id
),
away_stats AS (
  SELECT
    away_team_id AS team_id,
    COUNT(*) AS matches_played,
    SUM(away_team_goals) AS goal_for,
    SUM(home_team_goals) AS goal_against,
    SUM(CASE
      WHEN away_team_goals > home_team_goals THEN 3
      WHEN away_team_goals = home_team_goals THEN 1
      ELSE 0
    END) AS points
  FROM
    Matches
  GROUP BY
    away_team_id
),
combined_stats AS (
  SELECT
    T.team_id,
    T.team_name,
    COALESCE(H.matches_played, 0) + COALESCE(A.matches_played, 0) AS matches_played,
    COALESCE(H.goal_for, 0) + COALESCE(A.goal_for, 0) AS goal_for,
    COALESCE(H.goal_against, 0) + COALESCE(A.goal_against, 0) AS goal_against,
    COALESCE(H.points, 0) + COALESCE(A.points, 0) AS points
  FROM
    Teams T
    LEFT JOIN home_stats H ON T.team_id = H.team_id
    LEFT JOIN away_stats A ON T.team_id = A.team_id
)
SELECT
  team_name,
  matches_played,
  points,
  goal_for,
  goal_against,
  goal_for - goal_against AS goal_diff
FROM
  combined_stats
ORDER BY
  points DESC,
  goal_diff DESC,
  team_name ASC;
```

Solution Implementation

Python

Java

C++

TypeScript

Recommended Readings