Problem 1841: League Statistics

Problem Information

Difficulty: Medium
Acceptance Rate: 53.51%
Paid Only: Yes
Tags: Database

Problem Description

Table: `Teams`

+----------------+---------+ | Column Name | Type | +----------------+---------+ | team_id | int | | team_name | varchar | +----------------+---------+ team_id is the column with unique values for this table. Each row contains information about one team in the league.

Table: `Matches`

+-----------------+---------+ | Column Name | Type | +-----------------+---------+ | home_team_id | int | | away_team_id | int | | home_team_goals | int | | away_team_goals | int | +-----------------+---------+ (home_team_id, away_team_id) is the primary key (combination of columns with unique values) for this table. Each row contains information about one match. home_team_goals is the number of goals scored by the home team. away_team_goals is the number of goals scored by the away team. The winner of the match is the team with the higher number of goals.

Write a solution to report the statistics of the league. The statistics should be built using the played matches where the **winning** team gets **three points** and the **losing** team gets **no points**. If a match ends with a **draw** , both teams get **one point**.

Each row of the result table should contain:

* `team_name` \- The name of the team in the `Teams` table. * `matches_played` \- The number of matches played as either a home or away team. * `points` \- The total points the team has so far. * `goal_for` \- The total number of goals scored by the team across all matches. * `goal_against` \- The total number of goals scored by opponent teams against this

team across all matches. * `goal_diff` \- The result of `goal_for - goal_against`.

Return the result table ordered by `points` **in descending order**. If two or more teams have the same `points`, order them by `goal_diff` **in descending order**. If there is still a tie, order them by `team_name` in **lexicographical order**.

The result format is in the following example.

Example 1:

Input: Teams table: +---------+-----------+ | team_id | team_name | +---------+-----------+ | 1 | Ajax | | 4 | Dortmund | | 6 | Arsenal | +---------+-----------+ Matches table: +--------------+--------------+----------------+----------------+ | 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 | +--------------+--------------+----------------+----------------+ **Output:** +-----------+---------------+--------+----------+--------------+-----------+ | 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 | +-----------+---------------+--------+----------+--------------+-----------+ **Explanation:** Ajax (team_id=1) played 4 matches: 2 losses and 2 draws. Total points = 0 + 0 + 1 + 1 = 2. Dortmund (team_id=4) played 2 matches: 2 wins. Total points = 3 + 3 = 6. Arsenal (team_id=6) played 2 matches: 2 draws. Total points = 1 + 1 = 2. Dortmund is the first team in the table. Ajax and Arsenal have the same points, but since Arsenal has a higher goal_diff than Ajax, Arsenal comes before Ajax in the table.

Code Snippets

MySQL:

```
# Write your MySQL query statement below
```

MS SQL Server:

```
/* Write your T-SQL query statement below */
```

PostgreSQL:

```
-- Write your PostgreSQL query statement below
```