SCS1309 Database Management

Lab Sheet - MySQL 04

In the last practical, you created a database to store information about Annual Inter-Faculty Sports Event of 2025. The event involves multiple teams from various departments and faculties, each participating in a series of sports competitions. You have been tasked with creating and managing a database to store information about the participants, teams, events, and the results of the sports competitions.

Before starting this practical, ensure that you have the database from the previous session.

Import the Schema:

- ➔ **Begin by importing the schema and data of the database created in the last practical into your SQL environment.**
- ➔ **Make sure all tables, data, and relationships are restored correctly before proceeding.**

1. Write a query to list all sports where at least one match has been conducted. Use a subquery to determine the relevant sports.

2. Write a query to list teams whose total score across all matches is greater than the average score of all teams. Use a nested subquery and aggregate functions.

3. Identify players who participated in more than one sport. Use a subquery to find such players based on their participation records.

4. Write a query to list match details along with the names of both participating teams and their respective scores.

5. Create a view to display the rankings of teams based on their total scores across all matches. Include columns for team name, total score, and rank.

6. Create a view that includes only non-sensitive details (e.g., name, age, and team) of players. Exclude details like nationality and other private attributes.

7. Create a view that lists only faculties that participated in at least three sports.

8. Write a query to retrieve the details of all players, including their name, age, team name, and the faculty their team represents.

9. Write a query to list the matches where both participating teams belong to the same faculty. Include match details, team names, and faculty name in the result.

10. Write a query to calculate the total score of all teams in each faculty across all matches. Include faculty name and total score in the result.

11. Write a query to identify players who have participated in matches where the opposing team belongs to their own faculty. Include player name, team name, and match details.

12. Write a query to rank faculties based on the total number of matches played by their teams. Include faculty name and the match count, sorted by match count in descending order.

13. You have the following two tables already. The Matches table has millions of records, and queries often join the **Matches table with the Teams table** to retrieve match details and team information.

    a. To optimize performance when querying match results for a specific team, create an index that will improve the performance of JOIN operations on the team1_id and team2_id columns in the Matches table.

    b. After creating the index, write a SQL query that retrieves the match_id, match_date, team1_name, team2_name, and the final score for all matches where the team with team_id = 5 was either team1 or team2. The results should be sorted by match_date in descending order.