ASSIGNMENT 2

CSA0593

-192311419

DEVA PRIYA S

QUESTION:

Design a database to manage teams, players, matches, and scores for a sports tournament.

- Model tables for teams, players, matches, and scores.

- Write stored procedures for scheduling and updating match results.

- Implement triggers to update team rankings and player statistics.

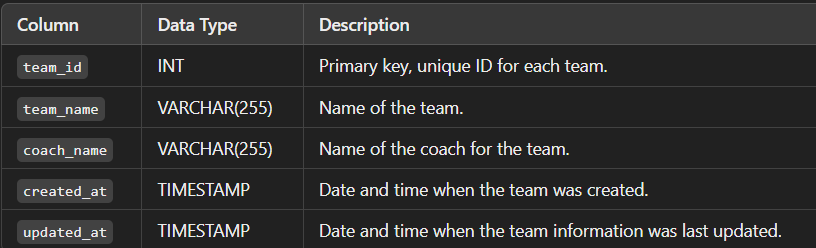
- Write SQL queries to analyze team performance and match statistics.

ANSWER:

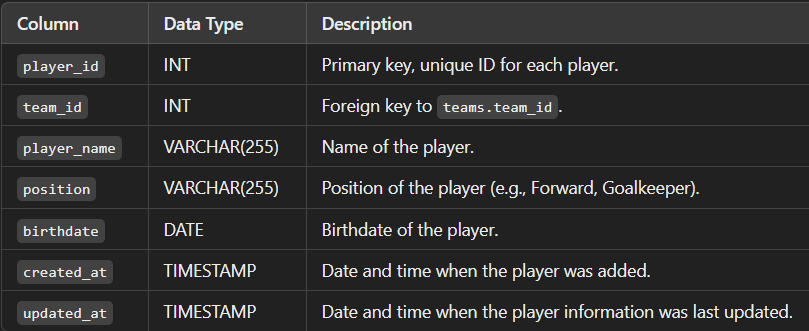
The database design for managing teams, players, matches, and scores for a sports tournament consists of several tables that store relevant information. The "Teams" table stores team details such as team ID, name, and description. The "Players" table stores player information including player ID, name, date of birth, and team ID (foreign key referencing the Teams table). The "Matches" table stores match details such as match ID, date, time, venue, and team IDs for both teams (foreign keys referencing the Teams table). The "Scores" table stores score information including score ID, match ID (foreign key referencing the Matches table), team ID (foreign key referencing the Teams table), and score. Additionally, there can be a "Tournaments" table to store tournament details such as tournament ID, name, and description, and a "Tournament\_Matches" table to store the matches for each tournament. This design captures the key information for managing teams, players, matches, and scores for a sports tournament.



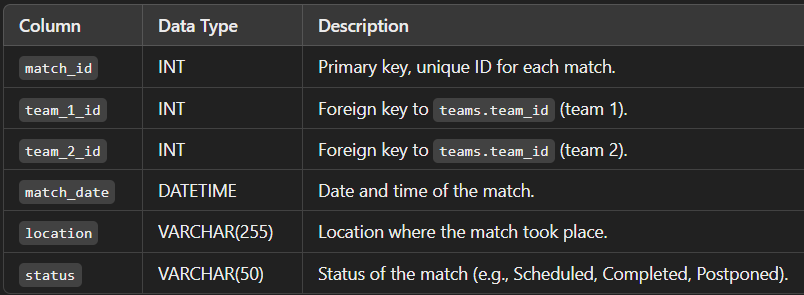
1.Teams Table



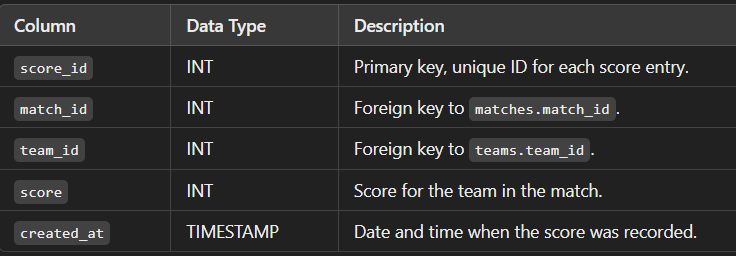
2. **Players Table**



3. **Matches Table**

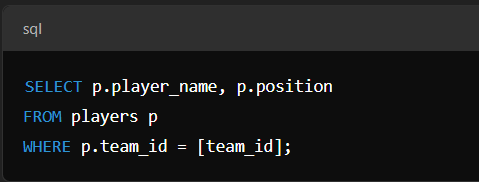


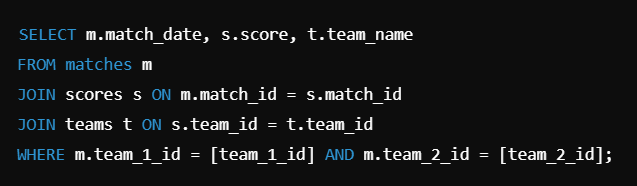
4. **Scores Table**



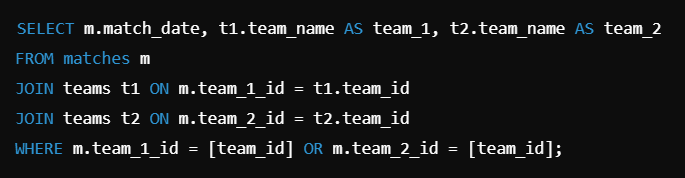
Query Scenarios:

**1.Get all players for a particular team**:

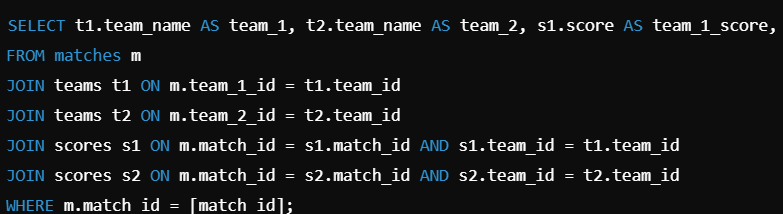
2. **Get the match scores between two teams**:



**2.Get all matches played by a specific team**:



3. **Get the final score of a specific match**:



**Conclusion**

The proposed database design efficiently models the key entities involved in managing a sports tournament: **teams**, **players**, **matches**, and **scores**. By structuring the data into distinct tables with clear relationships, we ensure the system is scalable, maintainable, and flexible to accommodate a variety of tournament formats and requirements.

**Key Features of the Design:**

1. **Normalization**: The schema is normalized to avoid redundancy, ensuring that each piece of information is stored only once (e.g., player and team details are kept in separate tables).
2. **Flexibility**: The design supports a variety of sports, allowing for the management of teams, players, and match results in a simple yet robust way. You can easily add new data points, such as player statistics or tournament stages, as needed.
3. **Referential Integrity**: Foreign key relationships ensure that data across tables remains consistent (e.g., a match must involve valid teams, and scores must reference valid matches).
4. **Scalability**: The structure can handle a large number of teams, players, matches, and scores, making it suitable for both small local tournaments and large-scale events.