

DATABASE PROJECT

Indian Premier League data project



ASSIGNMENT 5: SELECT QUERIES AND FINAL REPORT

Team The Index Warriors

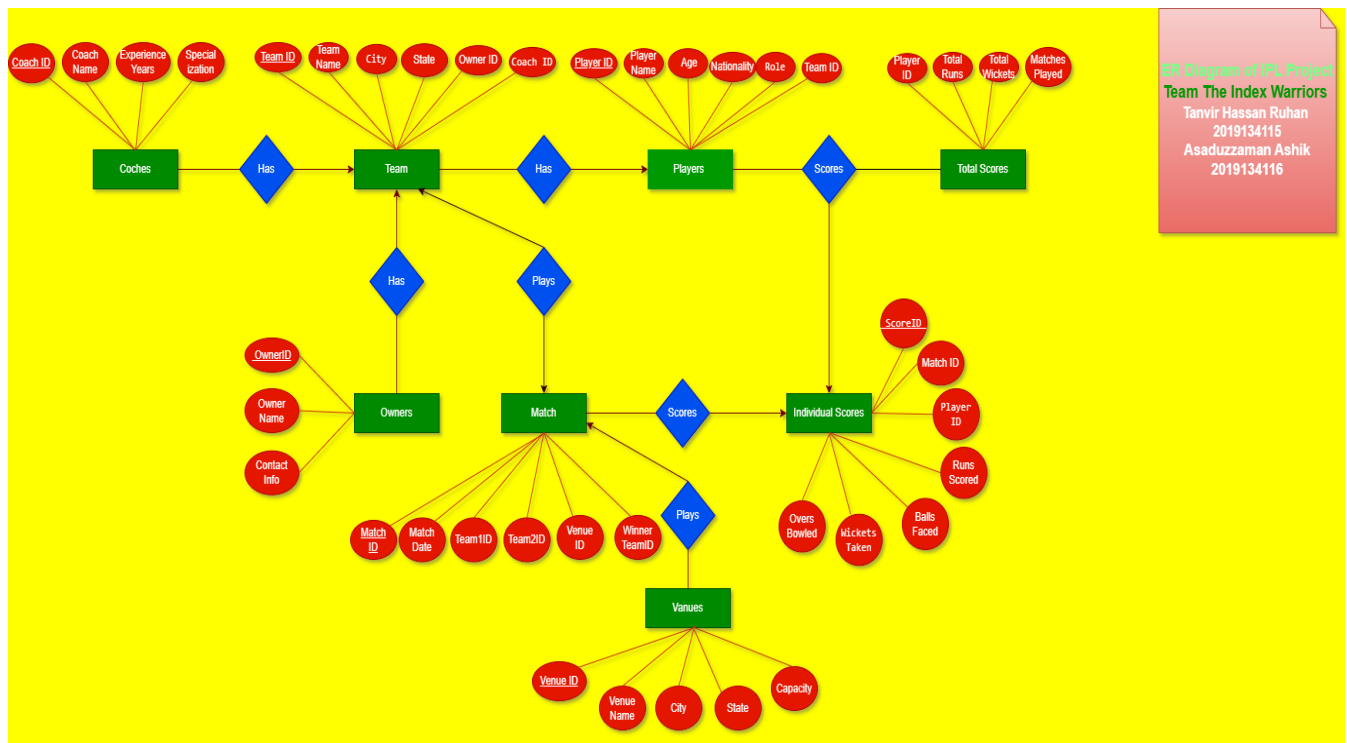
◆ Tanvir Hassan Ruhan (Reg.No.2019134115)

◆ Asaduzzaman Ashik (Reg.No:2019134116)

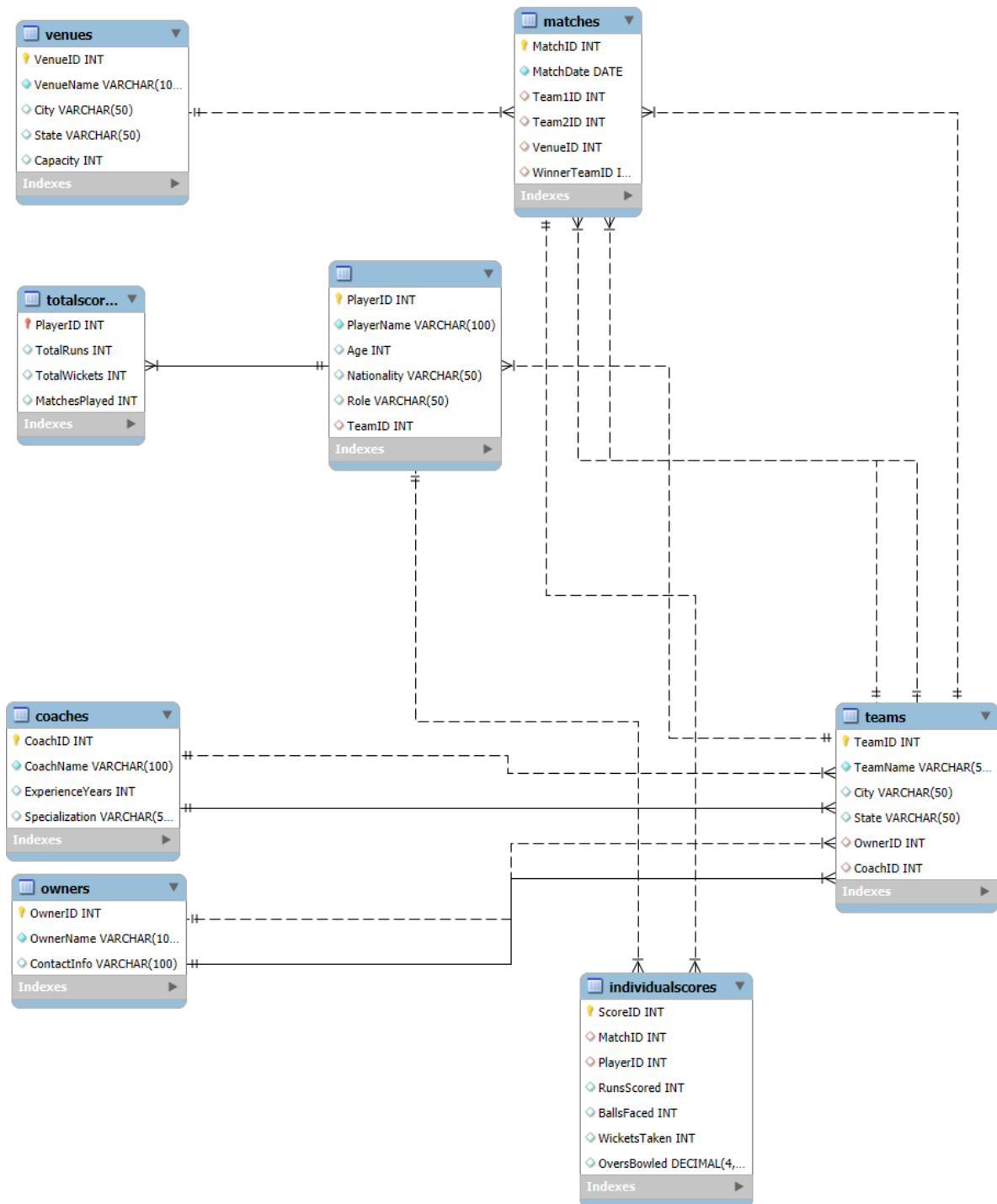
Introduction

The IPL Project on MySQL is designed to efficiently manage and analyze data related to the Indian Premier League (IPL), one of the most popular T20 cricket leagues in the world. This project focuses on creating a well-structured database to store crucial information, including match details, team performances, player statistics, venues, and points tables. By utilizing MySQL, users can perform various operations such as retrieving match outcomes, analysing player performances, and generating insights using SQL queries. The project involves creating multiple relational tables and implementing key SQL concepts like joins, aggregate functions, subqueries, and indexing to optimize data handling. Through this project, users can extract valuable trends, such as identifying the highest run-scorers, best bowlers, and team performances over multiple seasons. Additionally, it serves as a great learning experience in database management, data analysis, and sports analytics, providing a hands-on approach to working with large datasets efficiently.

E-R Diagram (After Modifications and Normalization)



Schema Diagram (After Modifications and Normalization)



Queries

1. Retrieve all players who are batsmen from the Players table.

```
SELECT PlayerName, Age, Nationality
FROM Players
WHERE Role = 'Batsman';
```

2. Find the total number of runs scored by each team.

```
SELECT T.TeamName, SUM(TS.TotalRuns) AS TotalRunsScored
FROM TotalScores TS
JOIN Players P ON TS.PlayerID = P.PlayerID
JOIN Teams T ON P.TeamID = T.TeamID
GROUP BY T.TeamName
ORDER BY TotalRunsScored DESC;
```

3. Find players who have scored more runs than the average total runs of all players.

```
SELECT PlayerName, TotalRuns
FROM TotalScores TS
JOIN Players P ON TS.PlayerID = P.PlayerID
WHERE TS.TotalRuns > (SELECT AVG(TotalRuns) FROM TotalScores);
```

4. Retrieve match details along with the venue where each match was played.

```
SELECT M.MatchID, M.MatchDate, T1.TeamName AS Team1, T2.TeamName AS Team2,
V.VenueName, V.City
FROM Matches M
INNER JOIN Teams T1 ON M.Team1ID = T1.TeamID
INNER JOIN Teams T2 ON M.Team2ID = T2.TeamID
INNER JOIN Venues V ON M.VenueID = V.VenueID;
```

5. Retrieve all players along with their respective teams, even if they do not belong to any team.

```
SELECT P.PlayerName, P.Age, P.Role, T.TeamName  
FROM Players P  
LEFT JOIN Teams T ON P.TeamID = T.TeamID;
```

Challenges that we faced while implementing the Project

- 1.While creating the ER diagram, W couldn't make it accurately. I was making mistakes in identifying the relationships.
- 2.At first, we couldn't assign the foreign keys in the schema diagram, but later we were able to do it.

Conclusion

This project demonstrated how MySQL can be used to efficiently manage and analyze IPL data. We designed a structured database and used SQL queries to extract insights on players, teams, and matches. Despite initial challenges with ER diagrams and foreign keys, we successfully built a functional and insightful system. Overall, this project enhanced our skills in database design, SQL, and sports data analysis.