Cricket Match Management System

Team:

Dhruv Shrimali, 2021A7PS0008P Rudra Jewalikar, 2021A7PS0450P

ER diagram redundancies and conversion:

Redundancies

- 1) Belongs to relationship is a many to one relationship, so it is eliminated by adding Team_ID attribute to Player
- 2) Participates in relationship is a many to two relationship, so it is eliminated by adding team1 id and team2 id attributes to Matches
- 3) Achieves is a many to one relationship, it is eliminated by adding Player_ID attribute to score.
- 4) Has is a many to one relationship, it is eliminated by adding Match_ID attribute to Score

Relational Schema:

```
Tables:

1)Player
Attributes:

player_id (primary key)
player_name
player_age
player_position
team_id (foreign key referencing the Team table)
```

there are constraints to ensure that all the attributes are not null.

```
2)Team
Attributes:
team_id (primary key)
team_name
coach name
```

```
Home_ground
```

there are constraints to ensure that all the attributes are not null.

```
3)Match
Attributes:
    match_id (primary key)
    match_date
    location
    winner_team_id (foreign key referencing the Team table)
```

there are constraints to ensure that all the attributes are not null.

```
4)Score
Attributes:
    score_id (primary key)
    match_id (foreign key referencing the Match table)
    player_id (foreign key referencing the Player table)
    runs_scored
    wickets_taken
    overs bowled
```

there are constraints to ensure that all the attributes are not null.

```
Player(<u>player_id</u>, player_name, player_age, player_position, team_id)
Team(<u>team_id</u>, team_name, coach_name, Home_ground)
Match(<u>match_id</u>, match_date, location, winner_team_id)
Score(<u>score_id</u>, match_id, player_id, Runs_scored, wickets_taken, overs_bowled)
```

There are four relations: Belongs to, Participates in, Has and Achieves Player Belongs to a Team, 2 Teams Participate in a match, a Match Has Scores, a Player Achieves a Score.

Functional Dependencies:

Team:

• Team_id → coach_name, home_ground, team_name

Issues:

 It has a transitive dependency that was resolved by splitting the table to team and team name

Player:

Player_ID → player_name, player_age, player_position

Matches:

Match_id → match_date, location, team1_id, team2_id, result

Batting Score:

• Match_ID, Team_ID, Player_ID → Runs_Scored, Overs_Played

Bowling_Score:

 $\bullet \quad \mathsf{Match_ID}, \, \mathsf{Team_ID}, \, \mathsf{Player_ID} \to \mathsf{Wickets_Taken}, \, \mathsf{Overs_Bowled}$

Match Score:

Match_ID, Team_ID → Runs_Scored, Wickets_Taken, Overs_Played

The tables are already in 2NF form(since multivalued attributes are not present and partial dependencies are also not present), so we convert them to 3NF form

3NF Form:

Team:

• Team_id → coach_name, home_ground, team_name

Team Name:

 $\bullet \quad \text{Team_id} \rightarrow \text{team_name}$

<u>lssues</u>:

• It had transitive dependency earlier that was resolved by splitting the table to team and team_name

<u>Player</u>:

 $\bullet \quad \mathsf{Player_ID} \to \mathsf{player_name}, \ \mathsf{player_age}, \ \ \mathsf{player_position}$

Matches:

Match_id → match_date, location, team1_id, team2_id, result

Batting Score:

Match_ID, Team_ID, Player_ID → Runs_Scored, Overs_Played

Bowling Score:

Match_ID, Team_ID, Player_ID → Wickets_Taken, Overs_Bowled

Match Score:

Match_ID, Team_ID → Runs_Scored, Wickets_Taken, Overs_Played

Normalized Schema:

Player(<u>player_id</u>, player_name, player_age, player_position, team_id)
Team(<u>team_id</u>, team_name, coach_name, Home_ground)
Team-name(<u>team_id</u>, team_name)
Match(<u>match_id</u>, match_date, location, team1_id, team2_id, winner_team_id)
Batting_Score(<u>match_id</u>, <u>player_id</u>, runs_scored, overs_Played)
Bowling_Score(<u>match_id</u>, <u>player_id</u>, wickets_taken, overs_bowled)
Match_Score(<u>match_id</u>, <u>team_id</u>, runs_scored, wickets_taken, overs_played)

SQL Queries:

Functionalities:

Insert a new player record
 INSERT INTO Player (player_name, player_age, player_position, team_id)
 VALUES ('Virat', 33, 'Batsman', 1);



Update the team information such as coach name, home ground

UPDATE Team SET coach name = 'Ravi Shastri' WHERE team id = 1;



Insert new match records

UPDATE Matches SET result = 1 WHERE match_id = 1;

✓ 158 23:47:41 UPDATE Matches SET result = 1 WHERE match_id = 1 1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0 0.013 sec

Update previously added match records

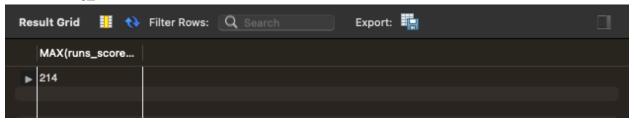
UPDATE matches SET result = 1 WHERE match id = 1;



Find the highest runs_scored by a player

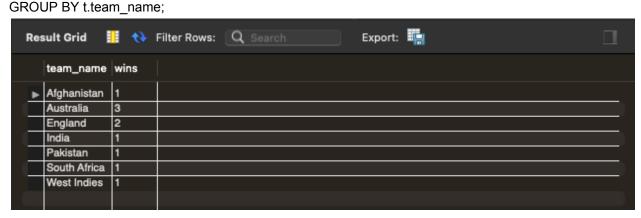
SELECT MAX(runs_scored)

FROM Batting_Score;



Find the total number of matches won by each team

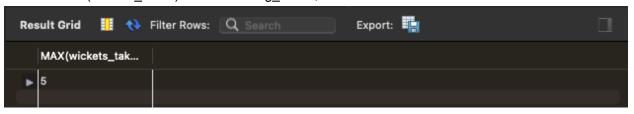
SELECT t.team_name, COUNT(*) AS wins FROM Team_Name t INNER JOIN Matches m ON t.team_id = m.result



 Find the average age of players in a team SELECT t.team_name, AVG(p.player_age) AS avg_age FROM Team_Name t
INNER JOIN Player p ON t.team_id = p.team_id
GROUP BY t.team_name;



 Find the highest wickets_taken by a player in a match SELECT MAX(wickets_taken) FROM Bowling_Score;



• Find the team with the most number of runs_scored across all matches

SELECT t.team_name, SUM(m.runs_scored) AS total_runs FROM Team_Name t, Match_Score m WHERE m.team_id = t.team_id GROUP BY t.team_name ORDER BY total_runs DESC LIMIT 1;



 Find the player_name and player_position for all players in a specific team

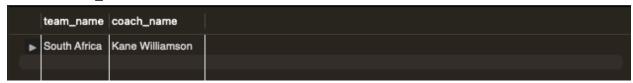
SELECT player_name, player_position FROM Player

WHERE team id = 1;



 Find the team_name and coach_name for all matches won by a specific team

SELECT n.team_name, t.coach_name FROM Team t JOIN Team_Name n ON t.team_id = n.team_id JOIN Matches m ON m.result = t.team_id WHERE t.team_id = 5;

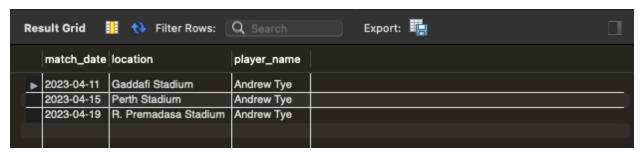


 Find the match_date, location, and player_name for all matches played by a specific player

SELECT Matches.match_date, Matches.location, Player.player_name FROM Matches
INNER JOIN Batting_Score ON Matches.match_id = Batting_Score.match_id
INNER JOIN Player ON Batting_Score.player_id = Player.player_id
WHERE Player.player_name = 'Andrew Tye'
UNION
SELECT Matches.match_date, Matches.location, Player.player_name
FROM Matches

INNER JOIN Bowling_Score ON Matches.match_id = Bowling_Score.match_id INNER JOIN Player ON Bowling_Score.player_id = Player.player_id

WHERE Player_player_name = 'Andrew Tye';



 Find the team_name, match_date, and runs_scored for all matches where a specific player scored more than 100 runs

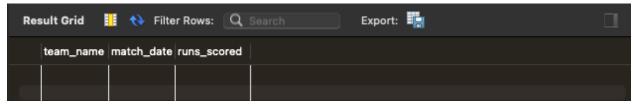
SELECT Team_Name.team_name, Matches.match_date, Batting_Score.runs_scored FROM Matches

INNER JOIN Batting Score ON Matches.match id = Batting Score.match id

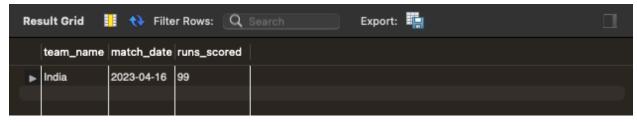
INNER JOIN Player ON Batting_Score.player_id = Player.player_id

INNER JOIN Team_Name ON Player.team_id = Team_Name.team_id

WHERE Player.player name = 'Virat Kohli' AND Batting Score.runs scored > 100;



Since the player did not score more than 100 runs in any match the table showed up empty. When the player was changed to 'Rohit Sharma" and runs to be scored were changed to 50, output is shown:



 Find the player_name, match_date, and wickets_taken for all matches where a specific player took more than 3 wickets

SELECT p.player_name, m.match_date, bs.wickets_taken FROM Bowling_Score bs INNER JOIN Player p ON bs.player_id = p.player_id INNER JOIN Matches m ON bs.match_id = m.match_id WHERE bs.wickets_taken > 3 AND p.player_name = 'Andrew Tye';



Since the player "Andrew Tye' didn't take more than 3 wickets in any match the table showed up empty.

When constraint was changed to more than 1 wickets, output was shown:



Google drive link:

https://drive.google.com/drive/folders/1C6T1Le6BZ0wDNZRSADtswC2BKUI5Ut6 | Page 1 | Page 2 | P

(Drive contains ER diagram and videos.)