■ Sports Tournament Tracker - MySQL Report

1. Schema Design

```
-- Teams
CREATE TABLE Teams (
    team_id INT AUTO_INCREMENT PRIMARY KEY,
    team_name VARCHAR(100) NOT NULL
);

-- Players
CREATE TABLE Players (
    player_id INT AUTO_INCREMENT PRIMARY KEY,
    player_name VARCHAR(100) NOT NULL,
    team_id INT,
    FOREIGN KEY (team_id) REFERENCES Teams(team_id)
);

-- Matches
CREATE TABLE Matches (
    match_id INT AUTO_INCREMENT PRIMARY KEY,
    team1_id INT,
    team2_id INT,
    match_date DATE,
    winner_team_id INT,
    FOREIGN KEY (team1_id) REFERENCES Teams(team_id),
    FOREIGN KEY (team2_id) REFERENCES Teams(team_id),
    FOREIGN KEY (winner_team_id) REFERENCES Teams(team_id)
);

-- Stats
CREATE TABLE Stats (
    stat_id INT AUTO_INCREMENT PRIMARY KEY,
    match_id INT,
    player_id INT,
    runs INT DEFAULT 0,
    wickets INT DEFAULT 0,
    points INT GENERATED ALWAYS AS (runs + (wickets * 10)) STORED,
    FOREIGN KEY (match_id) REFERENCES Matches(match_id),
    FOREIGN KEY (player_id) REFERENCES Players(player_id)
);
```

2. Sample Data Inserts

```
INSERT INTO Teams (team_name) VALUES ('Tigers'), ('Lions'), ('Eagles');

INSERT INTO Players (player_name, team_id) VALUES
('Alice', 1), ('Bob', 1),
('Charlie', 2), ('David', 2),
('Eve', 3), ('Frank', 3);

INSERT INTO Matches (team1_id, team2_id, match_date, winner_team_id) VALUES
(1, 2, '2025-08-20', 1),
(2, 3, '2025-08-22', 3),
(1, 3, '2025-08-25', 1);

INSERT INTO Stats (match_id, player_id, runs, wickets) VALUES
(1, 1, 40, 1), (1, 2, 20, 0), (1, 3, 15, 2), (1, 4, 10, 0),
(2, 3, 30, 1), (2, 4, 25, 0), (2, 5, 50, 2), (2, 6, 10, 1),
(3, 1, 60, 2), (3, 2, 15, 0), (3, 5, 20, 0), (3, 6, 5, 1);
```

3. Queries

```
-- Match Results
SELECT m.match_id, t1.team_name AS Team1, t2.team_name AS Team2,
       w.team_name AS Winner, m.match_date
FROM Matches m
JOIN Teams t1 ON m.team1_id = t1.team_id
JOIN Teams t2 ON m.team2_id = t2.team_id
JOIN Teams w ON m.winner_team_id = w.team_id;

-- Player Scores
SELECT p.player_name, t.team_name, s.match_id, s.runs, s.wickets, s.points
FROM Stats s
JOIN Players p ON s.player_id = p.player_id
JOIN Teams t ON p.team_id = t.team_id
ORDER BY s.match_id, p.player_name;
```

4. Views

```
-- Player Leaderboard
CREATE VIEW PlayerLeaderboard AS
SELECT p.player_name, t.team_name,
       SUM(s.runs) AS total_runs,
       SUM(s.wickets) AS total_wickets,
       SUM(s.points) AS total_points
FROM Players p
JOIN Stats s ON p.player_id = s.player_id
JOIN Teams t ON p.team_id = t.team_id
GROUP BY p.player_id
ORDER BY total_points DESC;

-- Team Points Table
CREATE VIEW TeamPoints AS
SELECT t.team_name,
       COUNT(CASE WHEN m.winner_team_id = t.team_id THEN 1 END) AS Wins,
       COUNT(CASE WHEN m.winner_team_id IS NOT NULL
                  AND m.winner_team_id != t.team_id
                  AND (m.team1_id = t.team_id OR m.team2_id = t.team_id)
             THEN 1 END) AS Losses
FROM Teams t
LEFT JOIN Matches m ON (t.team_id = m.team1_id OR t.team_id = m.team2_id)
GROUP BY t.team_id;
```

5. CTE for Average Player Performance

```
WITH AvgPerformance AS (
    SELECT p.player_name, AVG(s.runs) AS avg_runs, AVG(s.wickets) AS avg_wickets, AVG(s.points) AS av
    FROM Players p
    JOIN Stats s ON p.player_id = s.player_id
    GROUP BY p.player_id
)
SELECT * FROM AvgPerformance ORDER BY avg_points DESC;
```

6. Export Reports

```
SELECT * FROM PlayerLeaderboard
INTO OUTFILE '/tmp/player_leaderboard.csv'
FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n';
```