Winter Semester 2023

CSE250 Database Management System

Project title: Cricket Management System (IPL)

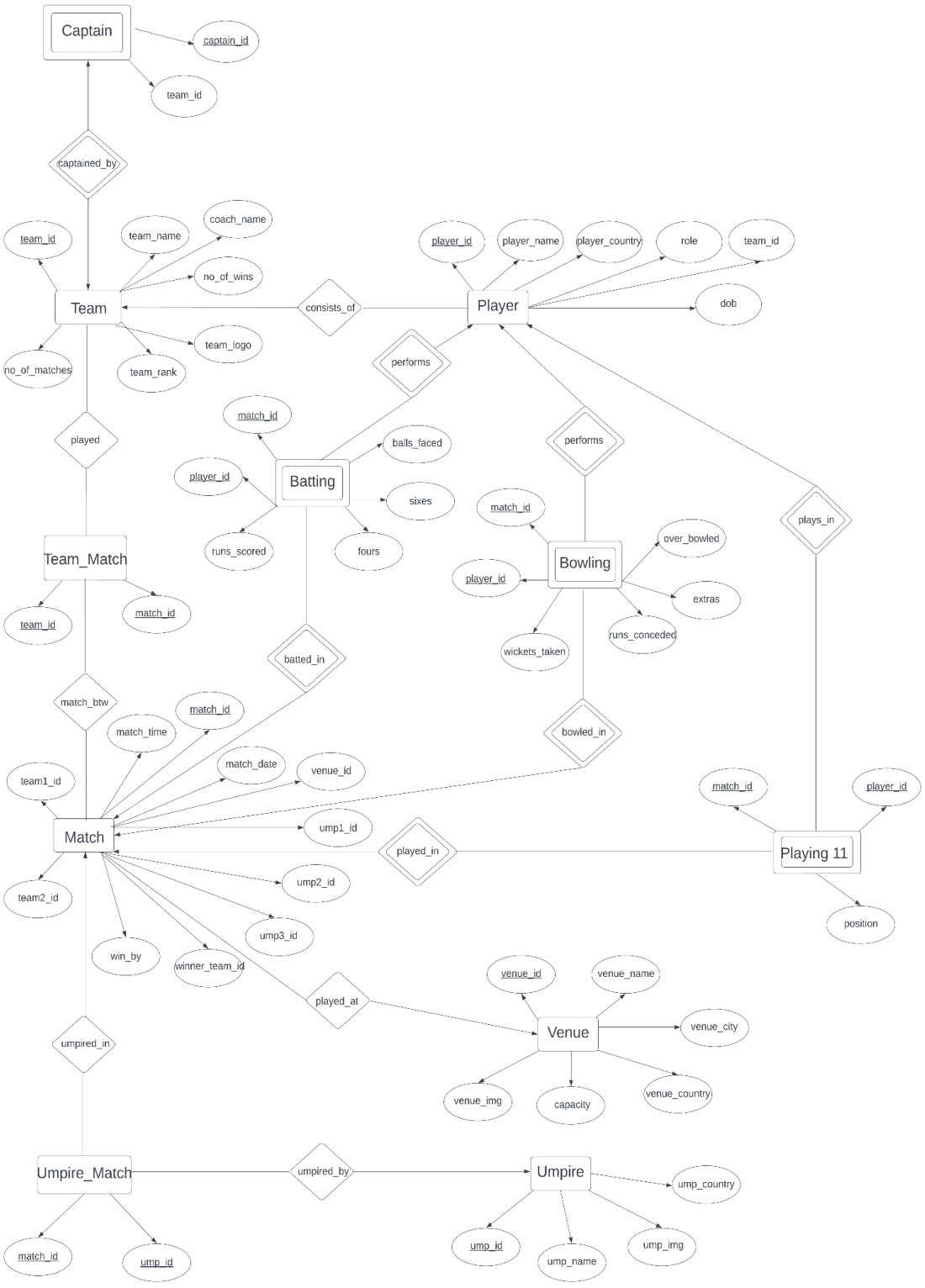
a. Description of the Project:

This is a cricket tournament system that includes IPL as a tournament and stores all the information related to the player, a match played between teams, team details, playing 11 in each match, venue information, and umpire information. The database consists of multiple tables, with each table representing a different entity and its associated attributes. For example, the team's table can have columns such as team name, owner, captain, coach, and home venue, while the player's table can have columns such as player name, team name, role, batting average, bowling average, and so on

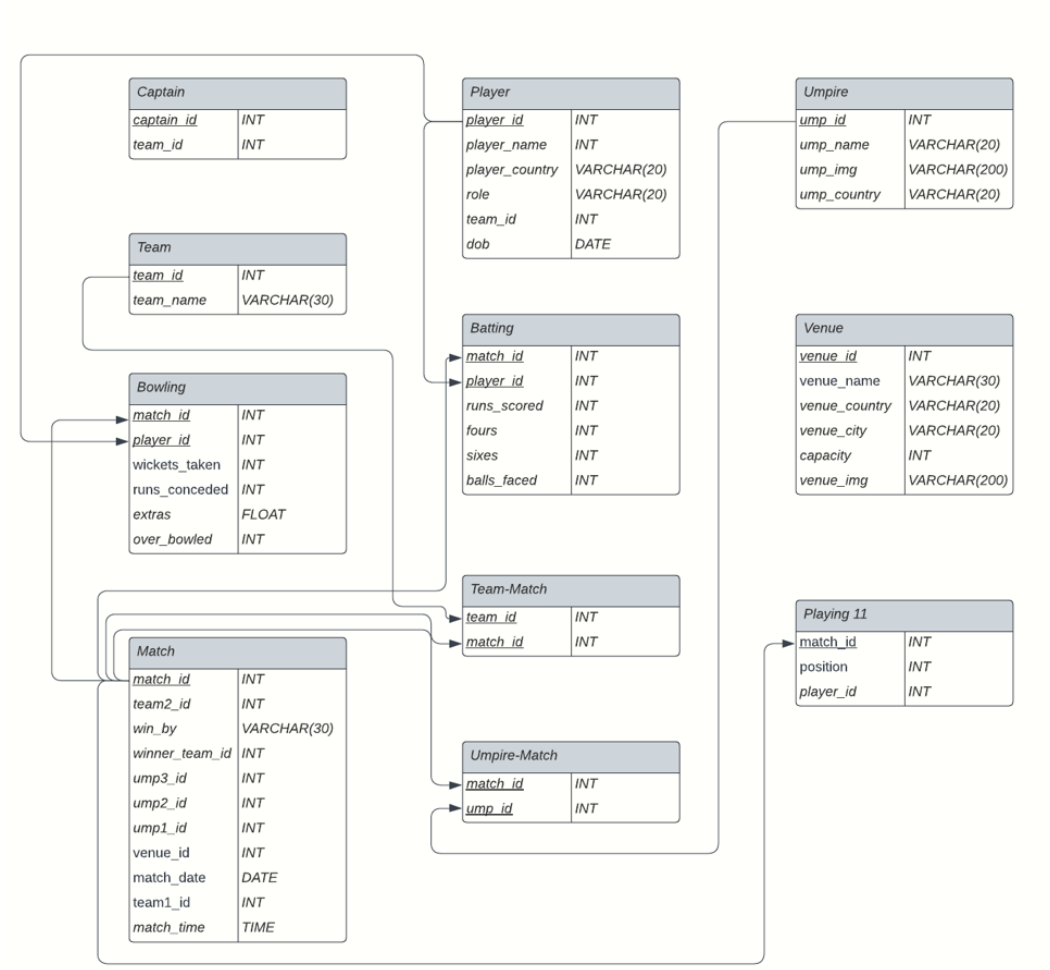
Software Specifications:

→ Xampp for storing the database
→ RDBMS: MySql
→ Backend Language: PHP

b. Entity-Relationship Diagram:



c. Table Design:



d. Stored Procedures, Function, and Triggers:

➔ Stored Procedures:

1) Procedure to add new umpire to umpire table:

```
DELIMITER //

CREATE PROCEDURE add_umpire(
   IN ump_name VARCHAR(30),
   IN ump_country VARCHAR(20),
   IN ump_img VARCHAR(200)
)
BEGIN
   INSERT INTO umpire (ump_name, ump_country, ump_img)
   VALUES (ump_name, ump_country, ump_img);
END //

DELIMITER ;
```

2) Procedure to display match schedule:

```
DELIMITER //

CREATE PROCEDURE display_match_schedule()
BEGIN
   SELECT
      m.match_id,
      m.match_date,
      m.match_time,
      t1.team_name AS team1_name,
      t2.team_name AS team2_name,
      v.venue_name,
      v.venue_city,
      v.venue_country,
      CONCAT(u1.ump_name, ', ', u2.ump_name, ', ', u3.ump_name) AS umpires,
      CASE
         WHEN EXISTS (SELECT 1 FROM batting WHERE match_id = m.match_id)
THEN 1
         ELSE 0
```

```
        END AS has_batting_details
    FROM
        `match` AS m
        INNER JOIN team AS t1 ON m.team1_id = t1.team_id
        INNER JOIN team AS t2 ON m.team2_id = t2.team_id
        INNER JOIN venue AS v ON m.venue_id = v.venue_id
        INNER JOIN umpire AS u1 ON m.ump1_id = u1.ump_id
        INNER JOIN umpire AS u2 ON m.ump2_id = u2.ump_id
        INNER JOIN umpire AS u3 ON m.ump3_id = u3.ump_id
    ORDER BY m.match_date;
END//

DELIMITER ;
```

3) Procedure to display points table:

```
DELIMITER //
CREATE PROCEDURE disp_ranking()
BEGIN
    SELECT team_rank, team_name, no_of_wins, no_of_matches, no_of_wins*2 AS
points
    FROM team
    ORDER BY team_rank ASC;
END//
DELIMITER ;
```

4) Procedure to display player details:

```
DELIMITER //
CREATE PROCEDURE display_player_details(p_player_id INT)
BEGIN
    SELECT player_name, player_country, role, team_name, dob, player_img
    FROM player
    INNER JOIN team ON player.team_id = team.team_id
    WHERE player_id = p_player_id;
END//
DELIMITER ;
```

5) Procedure to add venue:

```
DELIMITER //
CREATE PROCEDURE add_venue (
   IN venue_name VARCHAR(30),
   IN venue_city VARCHAR(20),
   IN venue_country VARCHAR(20),
   IN capacity INT,
   IN venue_img VARCHAR(200)
)
BEGIN
   INSERT INTO venue (venue_name, venue_city, venue_country, capacity,
venue_img)
   VALUES (venue_name, venue_city, venue_country, capacity, venue_img);
END //
DELIMITER ;
```

➔ **Functions:**

1) Function to calculate the batting strike rate of a player in a given match:

```
DELIMITER //

CREATE FUNCTION calculate_player_strike_rate(
    p_match_id INT,
    p_player_id INT
)
RETURNS FLOAT
READS SQL DATA
BEGIN
    DECLARE runs INT;
    DECLARE balls INT;
    DECLARE strike_rate FLOAT;

    SELECT SUM(runs_scored), SUM(balls_faced) INTO runs, balls
    FROM batting
    WHERE match_id = p_match_id AND player_id = p_player_id;

    IF balls > 0 THEN
        SET strike_rate = (runs / balls) * 100;
    ELSE
        SET strike_rate = 0;
    END IF;

    RETURN strike_rate;
END//

DELIMITER ;
```

2) Function to calculate the bowling economy of a player in a match:

```
DELIMITER //

CREATE FUNCTION calculate_bowling_economy(
    p_match_id INT,
    p_player_id INT
)
RETURNS DECIMAL(4,2)
READS SQL DATA
BEGIN
    DECLARE runs INT;
    DECLARE overs DECIMAL(3,1);
    DECLARE balls INT;
    SELECT SUM(runs_conceded), SUM(over_bowled)
    INTO runs, overs
    FROM bowling
    WHERE match_id = p_match_id AND player_id = p_player_id;
    SET balls = FLOOR(overs)*6 + ROUND(MOD(overs,1)*10);
    IF overs = 0.0 THEN
        RETURN NULL;
    ELSE
        RETURN (runs / balls) * 6;
    END IF;
END//

DELIMITER ;
```

3) Function to calculate total runs scored by a player in IPL:

```
DELIMITER //

CREATE FUNCTION get_total_runs_scored(p_player_id INT) RETURNS INT
BEGIN
    DECLARE total_runs INT;
    IF p_player_id IS NULL THEN
        SET total_runs = NULL;
    ELSE
        SELECT SUM(runs_scored) INTO total_runs FROM batting WHERE player_id =
p_player_id;
```

```
    END IF;
    RETURN total_runs;
END//

DELIMITER ;
```

4) Function to calculate total wickets taken by a player in IPL:

```
DELIMITER //

CREATE FUNCTION get_total_wickets_taken(p_player_id INT) RETURNS INT
BEGIN
    DECLARE total_wickets INT;
    IF p_player_id IS NULL THEN
        SET total_wickets = NULL;
    ELSE
        SELECT SUM(wickets_taken) INTO total_wickets FROM bowling WHERE
player_id = p_player_id;
    END IF;
    RETURN total_wickets;
END//

DELIMITER ;
```

5) Function to calculate total matches played by player in IPL:

```
DELIMITER //

CREATE FUNCTION get_total_matches_played(p_player_id INT) RETURNS INT
BEGIN
    DECLARE total_matches INT;
    IF p_player_id IS NULL THEN
        SET total_matches = NULL;
    ELSE
        SELECT COUNT(DISTINCT match_id) INTO total_matches FROM playing11
WHERE player_id = p_player_id;
    END IF;
    RETURN total_matches;
END//
```

```
DELIMITER ;
```

➔ **Triggers:**

1) **Update No of wins and no of matches in the team table if a new match is inserted in the match table:**

```
DELIMITER //

CREATE TRIGGER update_match_counts
AFTER INSERT ON `match`
FOR EACH ROW
BEGIN
   IF NEW.winner_team_id IS NOT NULL THEN
      UPDATE team SET no_of_matches = no_of_matches + 1 WHERE team_id =
NEW.team1_id;
      UPDATE team SET no_of_matches = no_of_matches + 1 WHERE team_id =
NEW.team2_id;
      UPDATE team SET no_of_wins = no_of_wins + 1 WHERE team_id =
NEW.winner_team_id;
   END IF;
END//

DELIMITER ;
```

2) **Trigger to update team rank after every match:**

```
CREATE TRIGGER update_team_rank
AFTER INSERT ON `match`
FOR EACH ROW
BEGIN
   UPDATE team t
   SET t.no_of_wins = t.no_of_wins + IF(NEW.winner_team_id = t.team_id, 1, 0),
      t.no_of_matches = t.no_of_matches + 1,
      t.team_rank = (SELECT COUNT(*)+1 FROM team WHERE no_of_wins >
t.no_of_wins OR (no_of_wins = t.no_of_wins AND no_of_matches < t.no_of_matches))
   WHERE t.team_id = NEW.team1_id OR t.team_id = NEW.team2_id;
END;
```

3) Trigger to update captain_id in the team table when the new captain is assigned:

```
CREATE TRIGGER update_team_captain
AFTER INSERT ON captain
FOR EACH ROW
BEGIN
   UPDATE team t
   SET t.captain_id = NEW.player_id
   WHERE t.team_id = NEW.team_id;
END;
```

4) Trigger to prevent insertion of players from different teams in playing 11 table:

```
CREATE TRIGGER check_player_team
BEFORE INSERT ON playing11
FOR EACH ROW
BEGIN
   IF (SELECT team_id FROM player WHERE player_id = NEW.player_id) !=
NEW.match_id THEN
      SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Player does not belong to
the team in the match';
   END IF;
END;
```

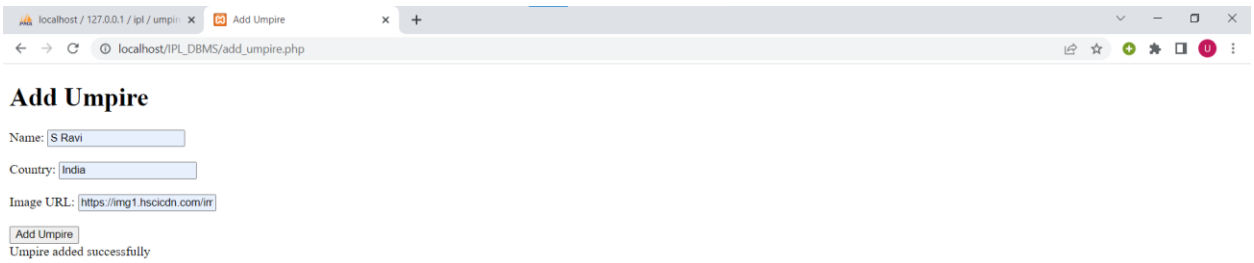
5) Trigger to print message when new player is inserted:

```
CREATE TRIGGER player_insert_trigger
AFTER INSERT ON player
FOR EACH ROW
BEGIN
   SELECT 'A new player has been added to the database.' AS message;
END;
```

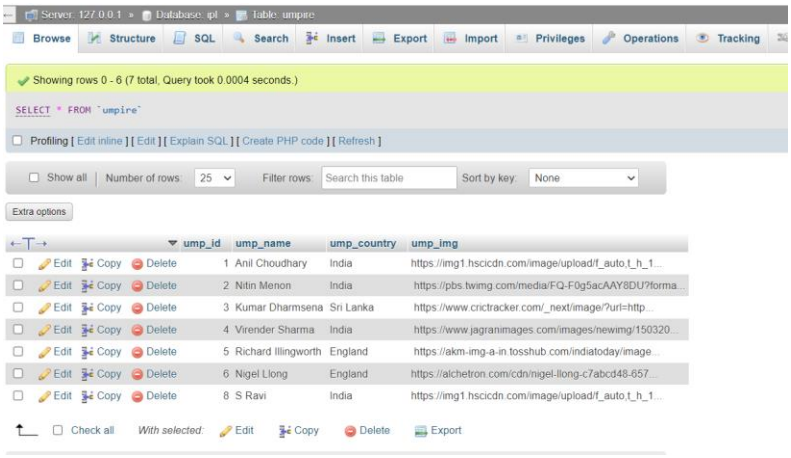
e. Results Screenshot(after procedure and function call on front-end):

➔ Procedures:

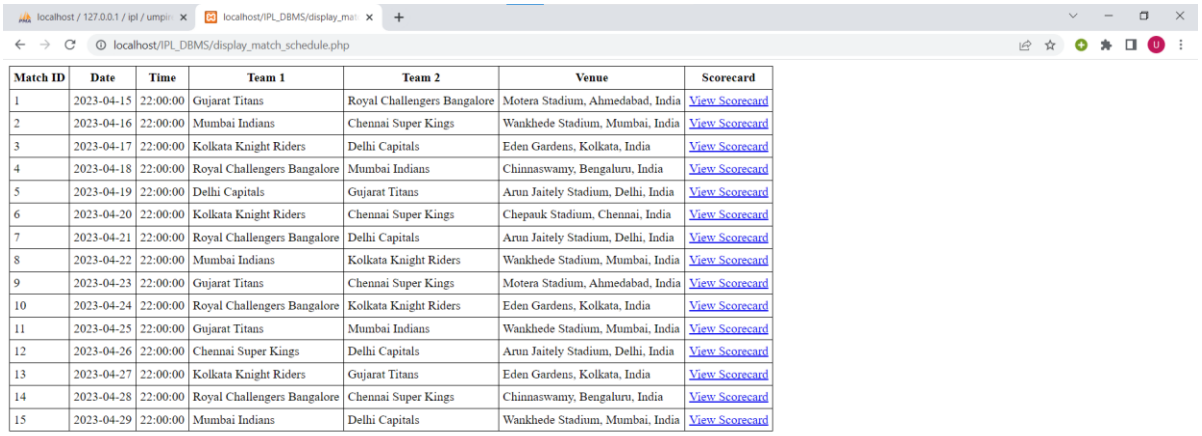
1) Add Umpire



Database updated:



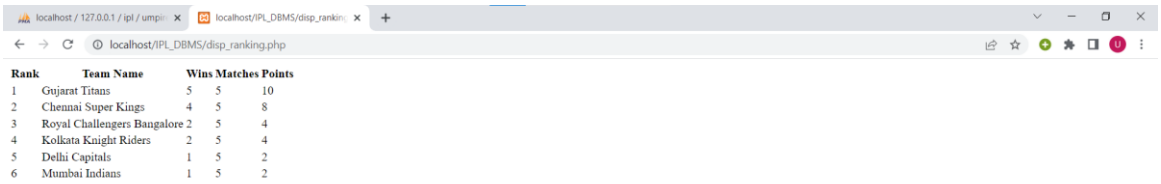
2) Display Match Schedule:



localhost/IPL_DBMS/display_match_schedule.php

Match ID	Date	Time	Team 1	Team 2	Venue	Scorecard
1	2023-04-15	22:00:00	Gujarat Titans	Royal Challengers Bangalore	Motera Stadium, Ahmedabad, India	View Scorecard
2	2023-04-16	22:00:00	Mumbai Indians	Chennai Super Kings	Wankhede Stadium, Mumbai, India	View Scorecard
3	2023-04-17	22:00:00	Kolkata Knight Riders	Delhi Capitals	Eden Gardens, Kolkata, India	View Scorecard
4	2023-04-18	22:00:00	Royal Challengers Bangalore	Mumbai Indians	Chinnaswamy, Bengaluru, India	View Scorecard
5	2023-04-19	22:00:00	Delhi Capitals	Gujarat Titans	Arun Jaitely Stadium, Delhi, India	View Scorecard
6	2023-04-20	22:00:00	Kolkata Knight Riders	Chennai Super Kings	Chepauk Stadium, Chennai, India	View Scorecard
7	2023-04-21	22:00:00	Royal Challengers Bangalore	Delhi Capitals	Arun Jaitely Stadium, Delhi, India	View Scorecard
8	2023-04-22	22:00:00	Mumbai Indians	Kolkata Knight Riders	Wankhede Stadium, Mumbai, India	View Scorecard
9	2023-04-23	22:00:00	Gujarat Titans	Chennai Super Kings	Motera Stadium, Ahmedabad, India	View Scorecard
10	2023-04-24	22:00:00	Royal Challengers Bangalore	Kolkata Knight Riders	Eden Gardens, Kolkata, India	View Scorecard
11	2023-04-25	22:00:00	Gujarat Titans	Mumbai Indians	Wankhede Stadium, Mumbai, India	View Scorecard
12	2023-04-26	22:00:00	Chennai Super Kings	Delhi Capitals	Arun Jaitely Stadium, Delhi, India	View Scorecard
13	2023-04-27	22:00:00	Kolkata Knight Riders	Gujarat Titans	Eden Gardens, Kolkata, India	View Scorecard
14	2023-04-28	22:00:00	Royal Challengers Bangalore	Chennai Super Kings	Chinnaswamy, Bengaluru, India	View Scorecard
15	2023-04-29	22:00:00	Mumbai Indians	Delhi Capitals	Wankhede Stadium, Mumbai, India	View Scorecard

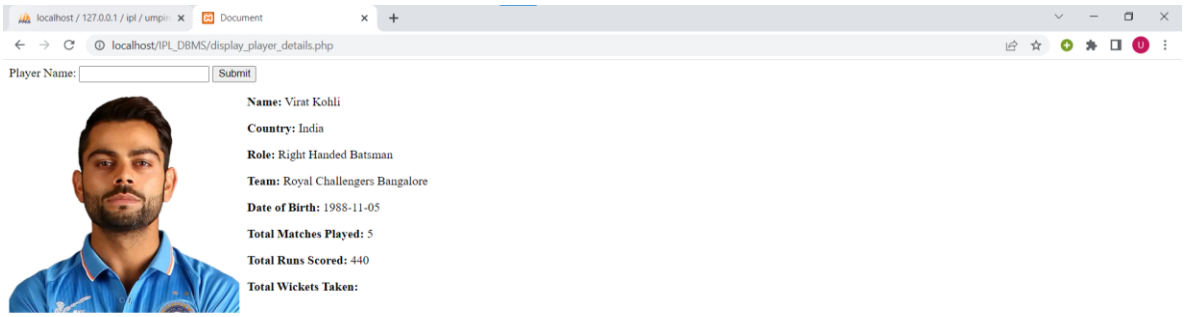
3) Display Ranking:



localhost/IPL_DBMS/disp_ranking.php

Rank	Team Name	Wins	Matches	Points
1	Gujarat Titans	5	5	10
2	Chennai Super Kings	4	5	8
3	Royal Challengers Bangalore	2	5	4
4	Kolkata Knight Riders	2	5	4
5	Delhi Capitals	1	5	2
6	Mumbai Indians	1	5	2

4) Display Player:



localhost/IPL_DBMS/display_player_details.php

Player Name: Submit

Name: Virat Kohli

Country: India

Role: Right Handed Batsman

Team: Royal Challengers Bangalore

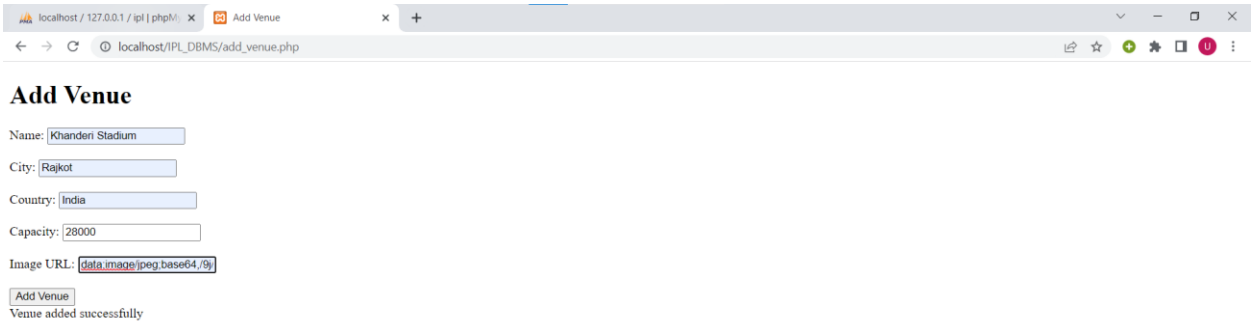
Date of Birth: 1988-11-05

Total Matches Played: 5

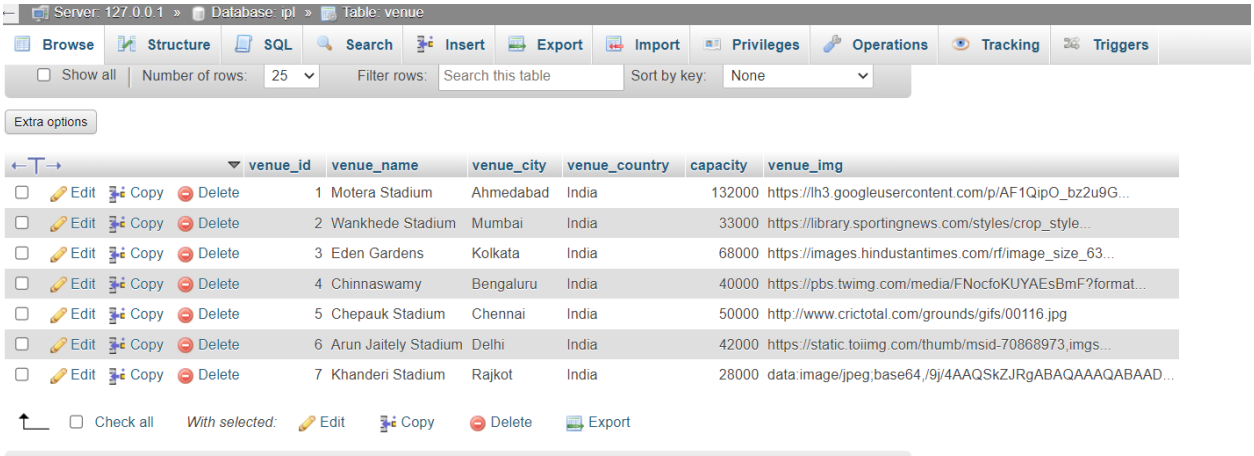
Total Runs Scored: 440

Total Wickets Taken:

5) Adding Venue to the venue table:

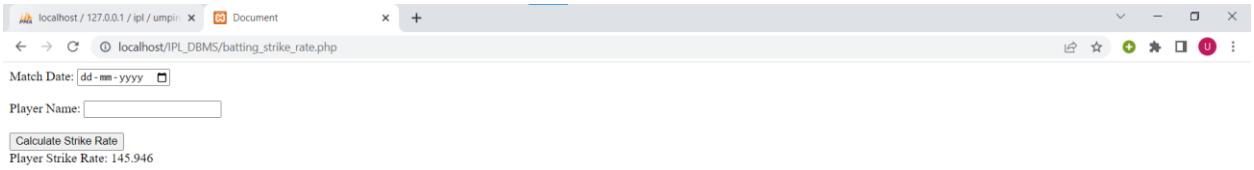


Database updated:

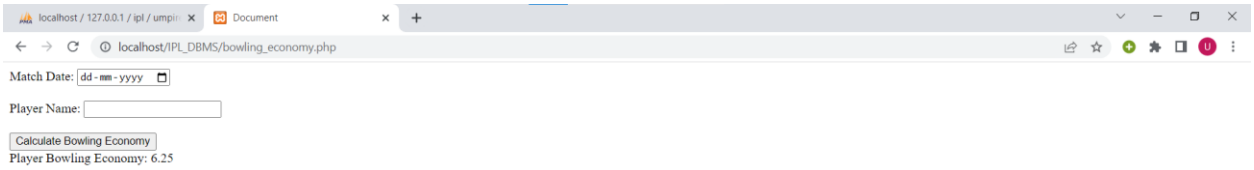


Functions:

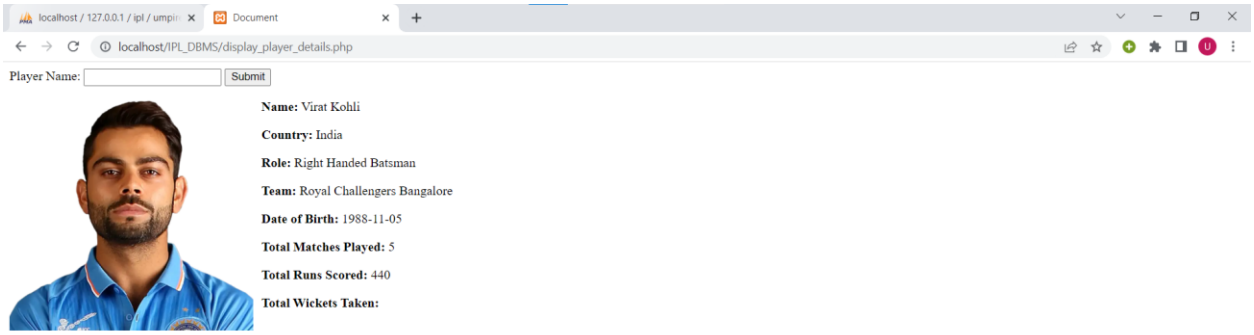
1) Calculate Batting Strike Rate:



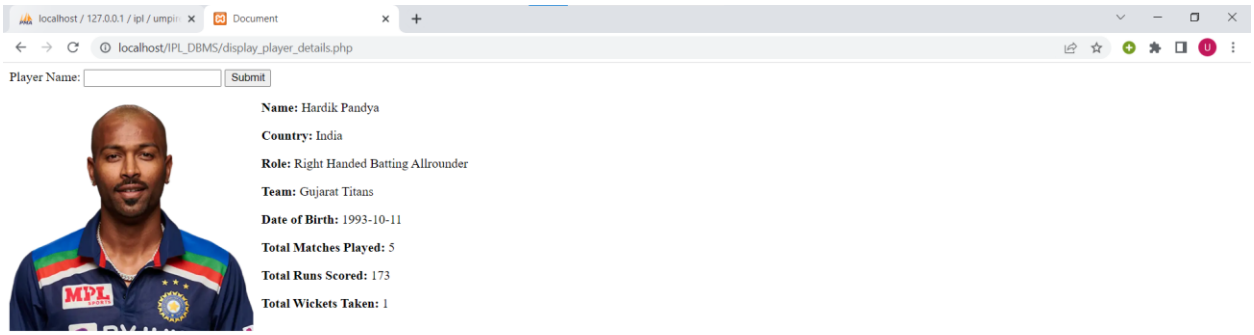
2) Calculate Bowling Economy:



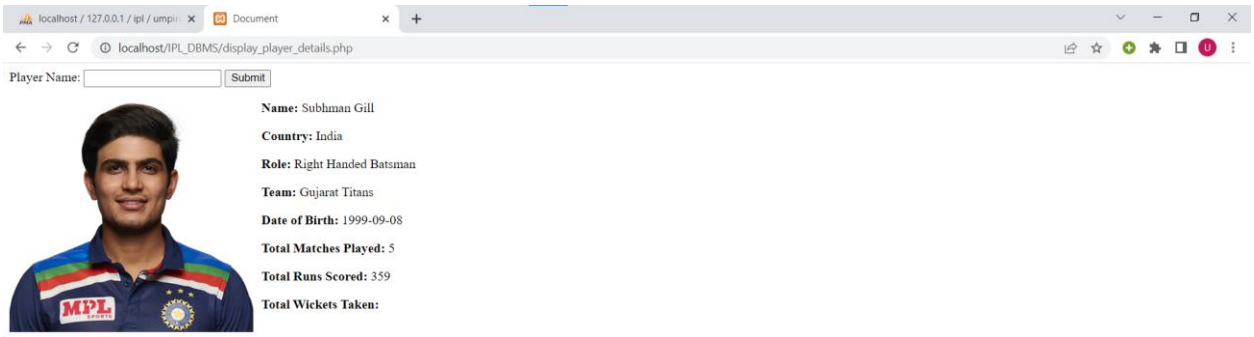
3) Get total runs scored by a player:



4) Get total wickets taken:



5) Get the total matches played by a player:

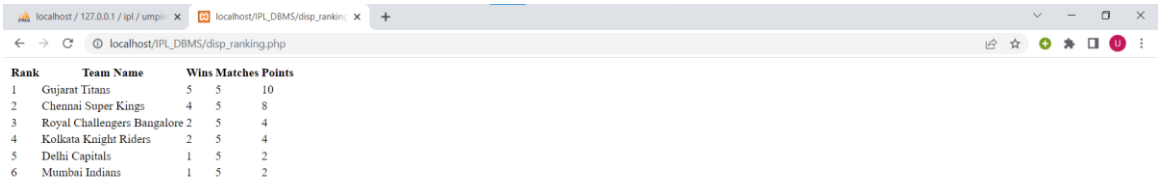


f. Errors Screenshot(after trigger fired):

1) On executing this query, the points table gets updated automatically:

INSERT INTO 'match' (match_date, match_time, team1_id, team2_id, venue_id, ump1_id, ump2_id, ump3_id, winner_team_id, win_by) VALUES ('2023-04-29', '22:00:00', 1, 2, 2, 4, 5, 1, 2, '12 runs');

Before Insertion:



After Insertion: