IPL 2025

DATABASE

MANAGEMENT

SYSTEM

IPL 2025 DATABASE MANAGEMENT SYSTEM

Project Overview

This project is about creating a database for the **IPL 2025 season** using **MySQL**.

It stores all the important details like **teams, players, matches, umpires, and points table**, and helps in getting information quickly using SQL queries.

The main idea is to make a system that:

- Keeps match data in an organized way
- Updates the points table automatically using triggers
- Shows final standings through views
- Tracks player performances like runs and wickets

We have connected all the tables using **primary and foreign keys** so that the data remains accurate. The design is based on an **ER diagram**, which makes it easy to understand the relationships between different tables.

In short, this project is like a mini version of a real cricket tournament database, which can be used to get stats, rankings, and other match details easily.

Requirements

1. Software Requirements

- **Database**: MySQL 8.x (or compatible version)
- **Interface Tools**: MySQL Workbench or MySQL Command Line Client or Command Prompt
- **Operating System**: Windows 10/11, Linux, or macOS
- **Text Editor (optional)**: VS Code or Notepad++ (for writing SQL scripts)

2. Hardware Requirements

- **Processor**: Intel Core i3 or above
- **RAM**: Minimum 4 GB (8 GB recommended)
- **Storage**: At least 500 MB free space for database and related files
- **Display**: 1366×768 resolution or higher

3. Dataset Requirements

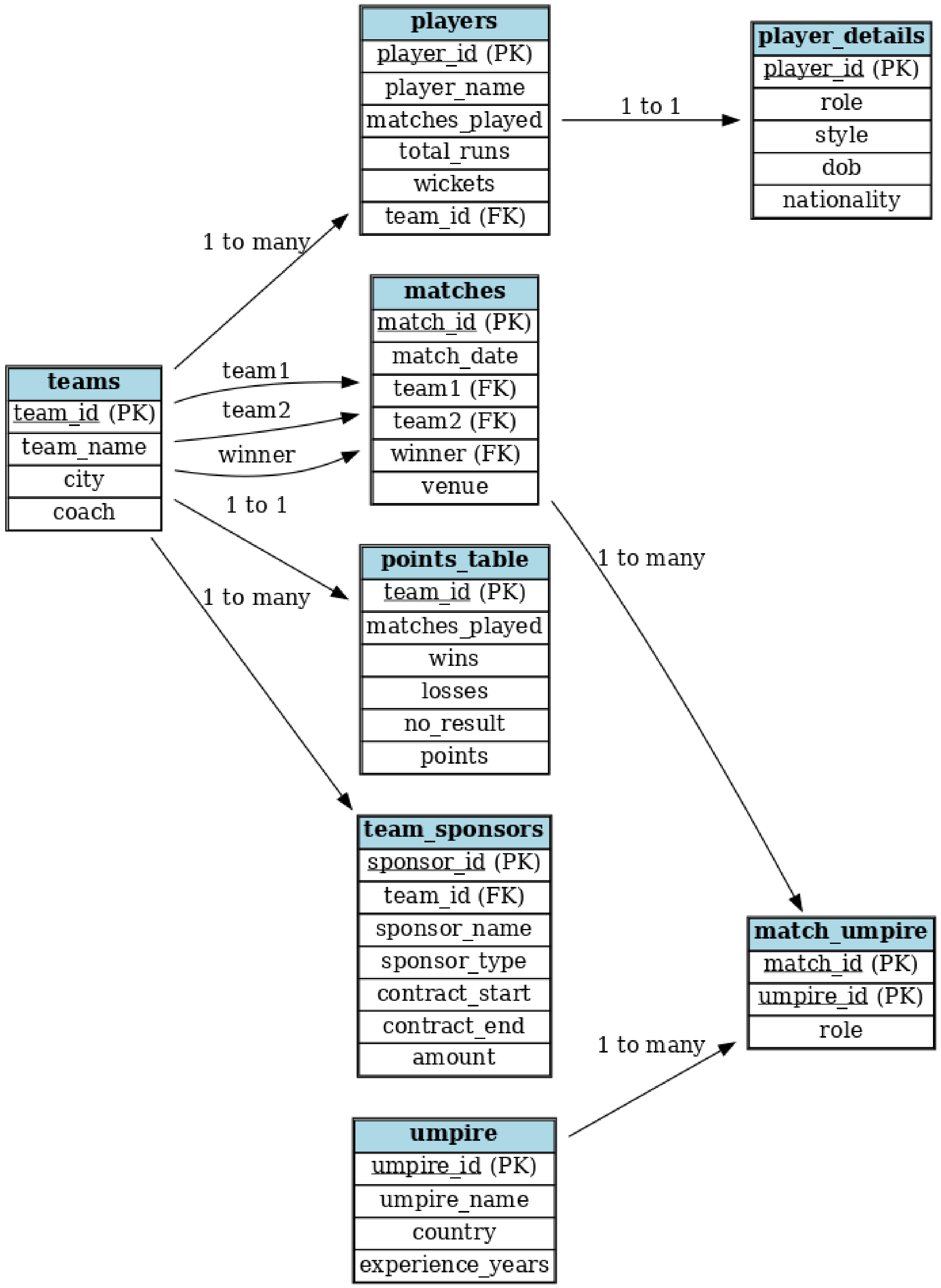
- IPL 2025 season match schedule and results
- Player details (name, team, role, stats)
- Team details (name, coach, home ground)
- Umpire and Sponsors details
- Points table rules (win = 2 points, No-Result = 1 point, loss = 0 points)

Database Design

The database for the **IPL 2025 SQL Project** is designed using the **Entity-Relationship (ER) model** to ensure the data is well-organized, normalized, and easy to maintain.
It consists of multiple interlinked tables, each responsible for storing specific information such as teams, players, matches, umpires, and points.

Entity – Relationship Diagram

IPL 2025 DATABASE MANAGEMENT SYSTEM



Schema Description

IPL 2025 DATABASE MANAGEMENT SYSTEM

Below is the list of tables with their purpose:

Table Name	Purpose
teams	Stores team details like name, coach, and home ground
Player_details	Contains player information including role, style, date-of-birth and nationality
matches	Holds match schedule, teams playing, results, and venue
match_umpire	Links matches with the umpires assigned
umpire	Contains umpire details such as name and experience and nationality
points_table	Stores wins, losses and points for each team
Players	Tracks player performance such as runs scored and wickets taken and matches played
Team_sponsors	Holds sponsor name, type, contract duration, contract value

Primary and Foreign Key Relationships

- **Primary Key (PK)**: A column (or set of columns) that uniquely identifies each record in a table.
    - Example:
        - team_id in teams table
        - match_id in matches table
        - player_id in players table
- **Foreign Key (FK)**: A column in one table that refers to the primary key in another table, creating a link between them.
    - Example:
        - team_id in players table → references teams.team_id

- match_id in match_umpire table → references matches.match_id
- umpire_id in match_umpire table → references umpire.umpire_id

Relationship Summary:

- **One-to-Many**: One team has many players (teams → players).
- **Many-to-Many**: Matches can have multiple umpires, and an umpire can officiate multiple matches (matches ↔ umpire via match_umpire).
- **One-to-One**: Each match has exactly one match referee (matches → match_referee).

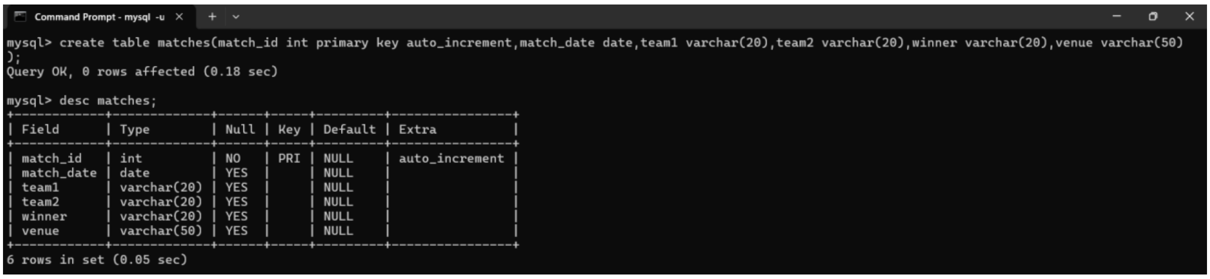
Tables Created

Below are the main tables created in the IPL 2025 database with descriptions, structures, and sample data.

Table: teams

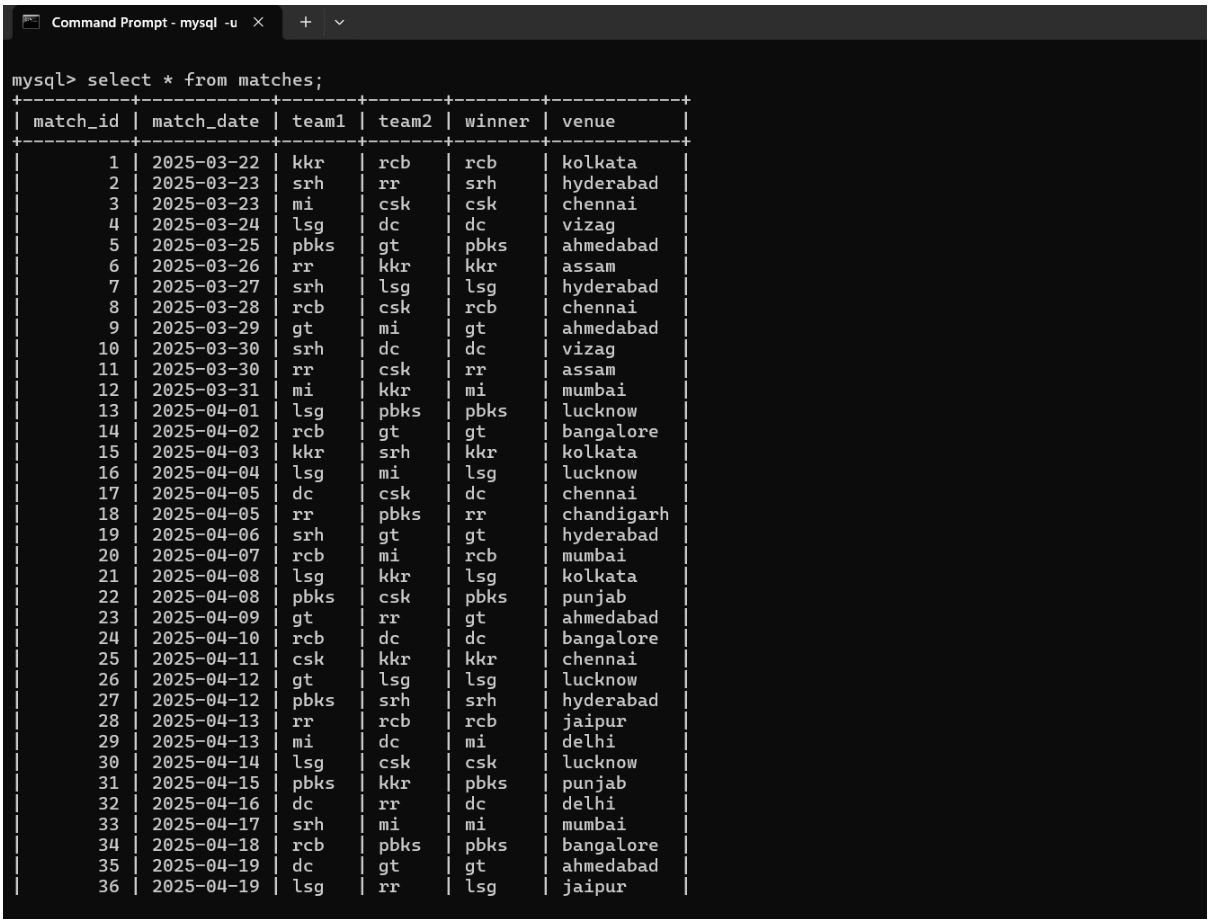
Purpose: Stores details of matches.

Structure:



```
mysql> create table matches(match_id int primary key auto_increment,match_date date,team1 varchar(20),team2 varchar(20),winner varchar(20),venue varchar(50)
);
Query OK, 0 rows affected (0.18 sec)

mysql> desc matches;
+------------+-------------+------+-----+---------+----------------+
| Field      | Type        | Null | Key | Default | Extra          |
+------------+-------------+------+-----+---------+----------------+
| match_id   | int         | NO   | PRI | NULL    | auto_increment |
| match_date | date        | YES  |     | NULL    |                |
| team1      | varchar(20) | YES  |     | NULL    |                |
| team2      | varchar(20) | YES  |     | NULL    |                |
| winner     | varchar(20) | YES  |     | NULL    |                |
| venue      | varchar(50) | YES  |     | NULL    |                |
+------------+-------------+------+-----+---------+----------------+
6 rows in set (0.05 sec)
```

Sample data:

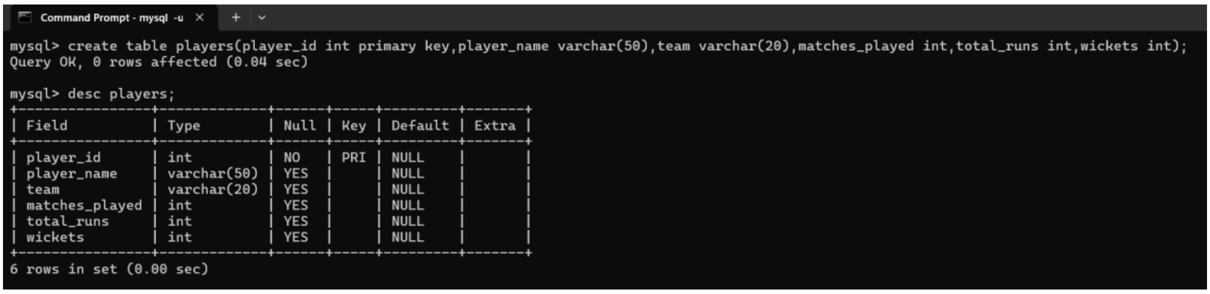


```
mysql> select * from matches;
+----------+------------+-------+-------+--------+------------+
| match_id | match_date | team1 | team2 | winner | venue      |
+----------+------------+-------+-------+--------+------------+
|        1 | 2025-03-22 | kkr   | rcb   | rcb    | kolkata    |
|        2 | 2025-03-23 | srh   | rr    | srh    | hyderabad  |
|        3 | 2025-03-23 | mi    | csk   | csk    | chennai    |
|        4 | 2025-03-24 | lsg   | dc    | dc     | vizag      |
|        5 | 2025-03-25 | pbks  | gt    | pbks   | ahmedabad  |
|        6 | 2025-03-26 | rr    | kkr   | kkr    | assam      |
|        7 | 2025-03-27 | srh   | lsg   | lsg    | hyderabad  |
|        8 | 2025-03-28 | rcb   | csk   | rcb    | chennai    |
|        9 | 2025-03-29 | gt    | mi    | gt     | ahmedabad  |
|       10 | 2025-03-30 | srh   | dc    | dc     | vizag      |
|       11 | 2025-03-30 | rr    | csk   | rr     | assam      |
|       12 | 2025-03-31 | mi    | kkr   | mi     | mumbai     |
|       13 | 2025-04-01 | lsg   | pbks  | pbks   | lucknow    |
|       14 | 2025-04-02 | rcb   | gt    | gt     | bangalore  |
|       15 | 2025-04-03 | kkr   | srh   | kkr    | kolkata    |
|       16 | 2025-04-04 | lsg   | mi    | lsg    | lucknow    |
|       17 | 2025-04-05 | dc    | csk   | dc     | chennai    |
|       18 | 2025-04-05 | rr    | pbks  | rr     | chandigarh |
|       19 | 2025-04-06 | srh   | gt    | gt     | hyderabad  |
|       20 | 2025-04-07 | rcb   | mi    | rcb    | mumbai     |
|       21 | 2025-04-08 | lsg   | kkr   | lsg    | kolkata    |
|       22 | 2025-04-08 | pbks  | csk   | pbks   | punjab     |
|       23 | 2025-04-09 | gt    | rr    | gt     | ahmedabad  |
|       24 | 2025-04-10 | rcb   | dc    | dc     | bangalore  |
|       25 | 2025-04-11 | csk   | kkr   | kkr    | chennai    |
|       26 | 2025-04-12 | gt    | lsg   | lsg    | lucknow    |
|       27 | 2025-04-12 | pbks  | srh   | srh    | hyderabad  |
|       28 | 2025-04-13 | rr    | rcb   | rcb    | jaipur     |
|       29 | 2025-04-13 | mi    | dc    | mi     | delhi      |
|       30 | 2025-04-14 | lsg   | csk   | csk    | lucknow    |
|       31 | 2025-04-15 | pbks  | kkr   | pbks   | punjab     |
|       32 | 2025-04-16 | dc    | rr    | dc     | delhi      |
|       33 | 2025-04-17 | srh   | mi    | mi     | mumbai     |
|       34 | 2025-04-18 | rcb   | pbks  | pbks   | bangalore  |
|       35 | 2025-04-19 | dc    | gt    | gt     | ahmedabad  |
|       36 | 2025-04-19 | lsg   | rr    | lsg    | jaipur     |
```

Table: Players

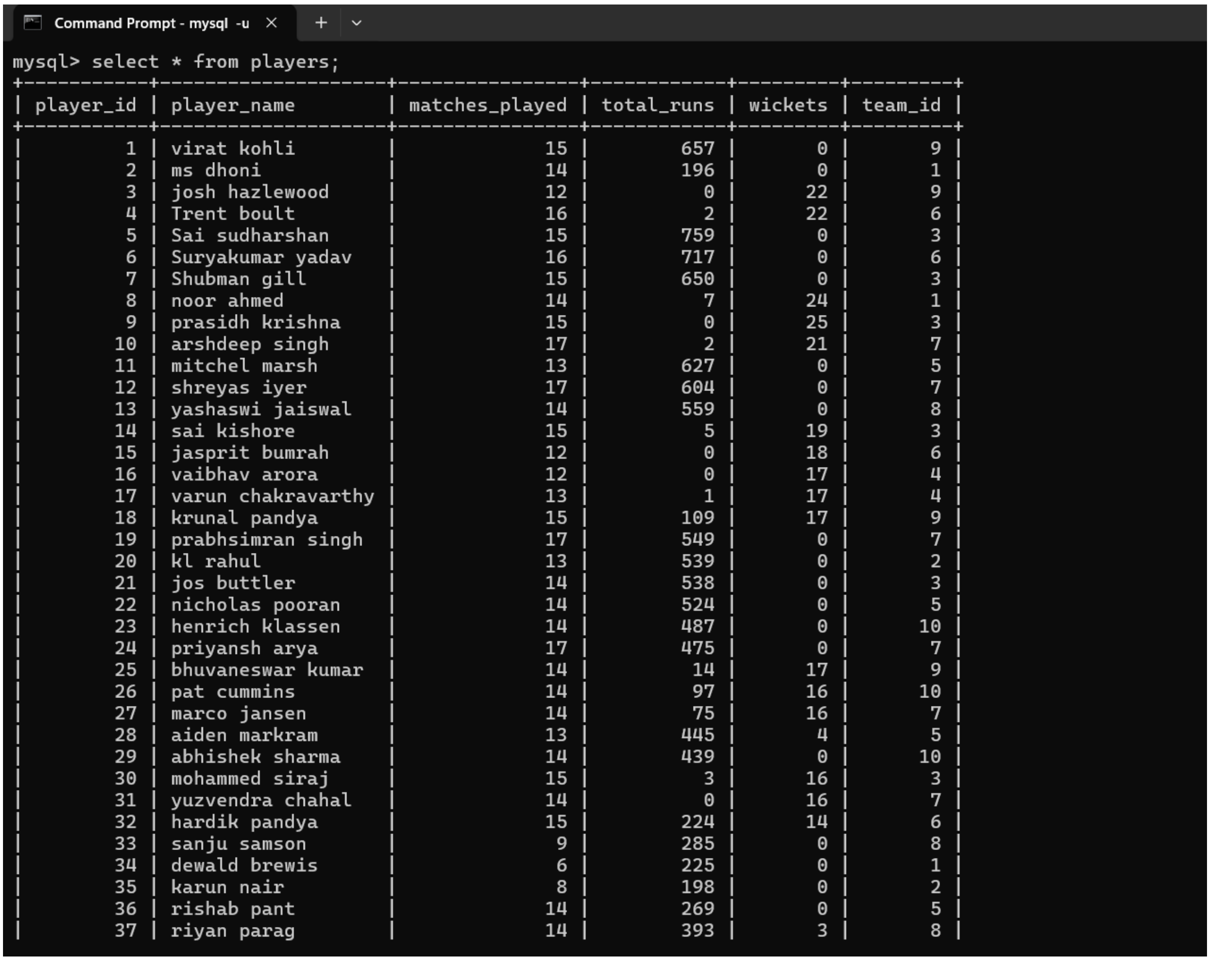
Purpose: Tracks player performance.

Structure:



```
mysql> create table players(player_id int primary key,player_name varchar(50),team varchar(20),matches_played int,total_runs int,wickets int);
Query OK, 0 rows affected (0.04 sec)

mysql> desc players;
+----------------+-------------+------+-----+---------+-------+
| Field          | Type        | Null | Key | Default | Extra |
+----------------+-------------+------+-----+---------+-------+
| player_id      | int         | NO   | PRI | NULL    |       |
| player_name    | varchar(50) | YES  |     | NULL    |       |
| team           | varchar(20) | YES  |     | NULL    |       |
| matches_played | int         | YES  |     | NULL    |       |
| total_runs     | int         | YES  |     | NULL    |       |
| wickets        | int         | YES  |     | NULL    |       |
+----------------+-------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

Sample Data:

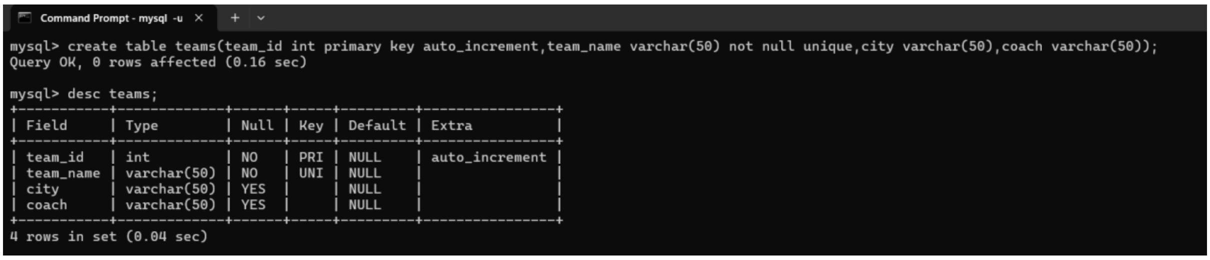


```
mysql> select * from players;
+-----------+----------------------+----------------+------------+---------+---------+
| player_id | player_name          | matches_played | total_runs | wickets | team_id |
+-----------+----------------------+----------------+------------+---------+---------+
|         1 | virat kohli          |             15 |        657 |       0 |       9 |
|         2 | ms dhoni             |             14 |        196 |       0 |       1 |
|         3 | josh hazlewood       |             12 |          0 |      22 |       9 |
|         4 | Trent boult          |             16 |          2 |      22 |       6 |
|         5 | Sai sudharshan       |             15 |        759 |       0 |       3 |
|         6 | Suryakumar yadav     |             16 |        717 |       0 |       6 |
|         7 | Shubman gill         |             15 |        650 |       0 |       3 |
|         8 | noor ahmed           |             14 |          7 |      24 |       1 |
|         9 | prasidh krishna      |             15 |          0 |      25 |       3 |
|        10 | arshdeep singh       |             17 |          2 |      21 |       7 |
|        11 | mitchel marsh        |             13 |        627 |       0 |       5 |
|        12 | shreyas iyer         |             17 |        604 |       0 |       7 |
|        13 | yashaswi jaiswal     |             14 |        559 |       0 |       8 |
|        14 | sai kishore          |             15 |          5 |      19 |       3 |
|        15 | jasprit bumrah       |             12 |          0 |      18 |       6 |
|        16 | vaibhav arora        |             12 |          0 |      17 |       4 |
|        17 | varun chakravarthy   |             13 |          1 |      17 |       4 |
|        18 | krunal pandya        |             15 |        109 |      17 |       9 |
|        19 | prabhsimran singh    |             17 |        549 |       0 |       7 |
|        20 | kl rahul             |             13 |        539 |       0 |       2 |
|        21 | jos buttler          |             14 |        538 |       0 |       3 |
|        22 | nicholas pooran      |             14 |        524 |       0 |       5 |
|        23 | henrich klassen      |             14 |        487 |       0 |      10 |
|        24 | priyansh arya        |             17 |        475 |       0 |       7 |
|        25 | bhuvaneswar kumar    |             14 |         14 |      17 |       9 |
|        26 | pat cummins          |             14 |         97 |      16 |      10 |
|        27 | marco jansen         |             14 |         75 |      16 |       7 |
|        28 | aiden markram        |             13 |        445 |       4 |       5 |
|        29 | abhishek sharma      |             14 |        439 |       0 |      10 |
|        30 | mohammed siraj       |             15 |          3 |      16 |       3 |
|        31 | yuzvendra chahal     |             14 |          0 |      16 |       7 |
|        32 | hardik pandya        |             15 |        224 |      14 |       6 |
|        33 | sanju samson         |              9 |        285 |       0 |       8 |
|        34 | dewald brewis        |              6 |        225 |       0 |       1 |
|        35 | karun nair           |              8 |        198 |       0 |       2 |
|        36 | rishab pant          |             14 |        269 |       0 |       5 |
|        37 | riyan parag          |             14 |        393 |       3 |       8 |
```

Table: teams

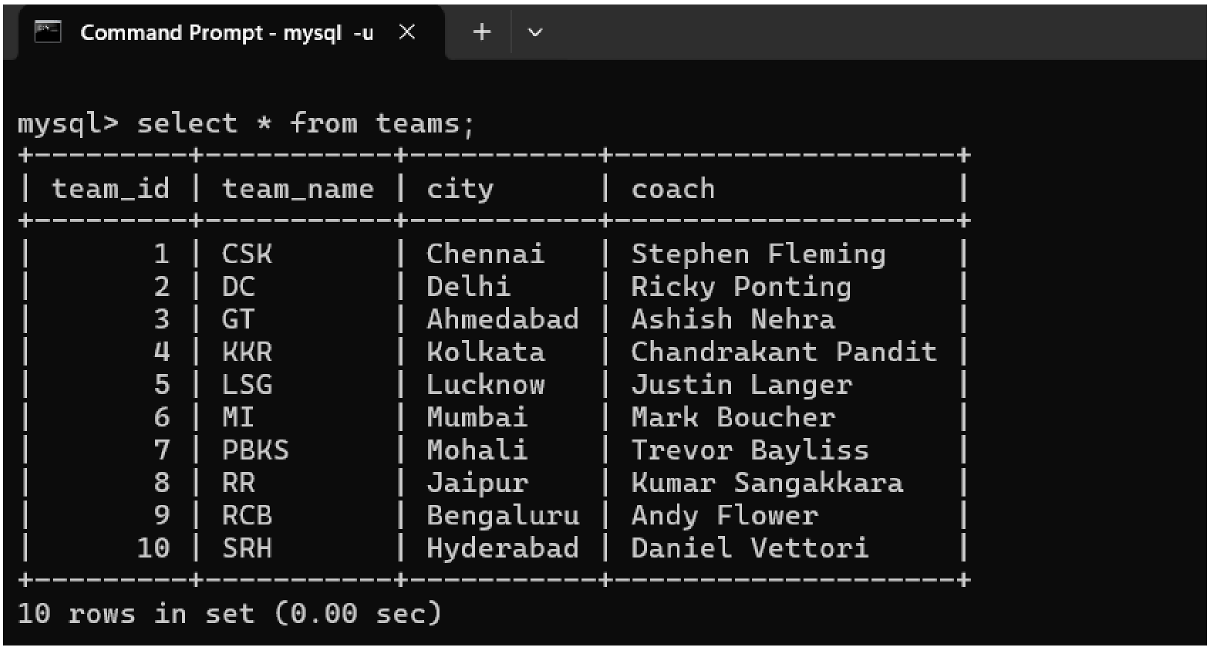
Purpose: Stores information about IPL teams.

Structure:



```
mysql> create table teams(team_id int primary key auto_increment,team_name varchar(50) not null unique,city varchar(50),coach varchar(50));
Query OK, 0 rows affected (0.16 sec)

mysql> desc teams;
+-----------+-------------+------+-----+---------+----------------+
| Field     | Type        | Null | Key | Default | Extra          |
+-----------+-------------+------+-----+---------+----------------+
| team_id   | int         | NO   | PRI | NULL    | auto_increment |
| team_name | varchar(50) | NO   | UNI | NULL    |                |
| city      | varchar(50) | YES  |     | NULL    |                |
| coach     | varchar(50) | YES  |     | NULL    |                |
+-----------+-------------+------+-----+---------+----------------+
4 rows in set (0.04 sec)
```

Sample Data:

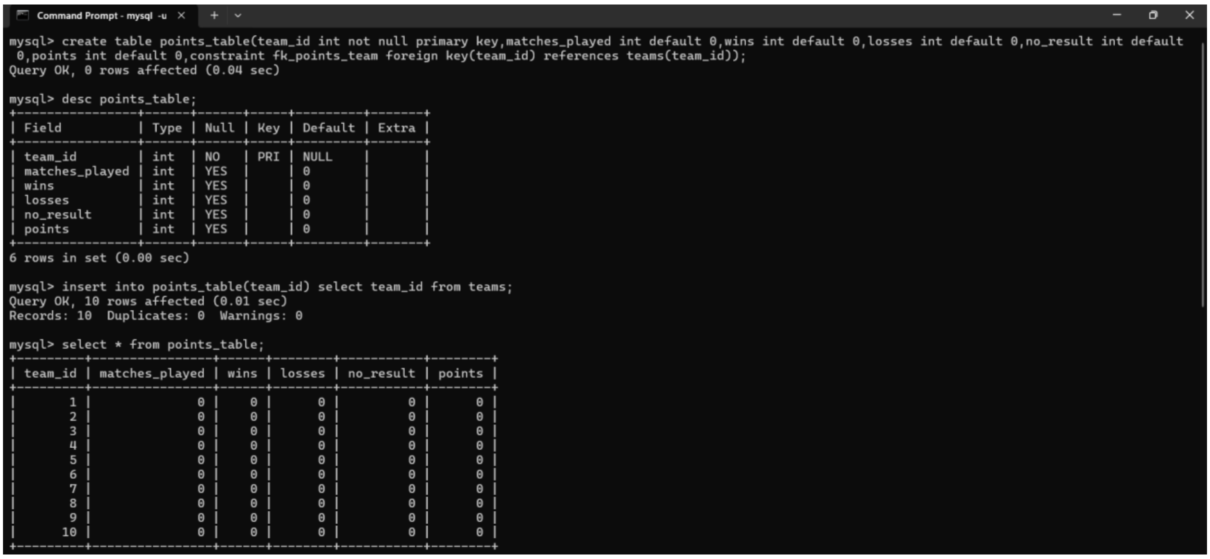


```
mysql> select * from teams;
+---------+-----------+-----------+--------------------+
| team_id | team_name | city      | coach              |
+---------+-----------+-----------+--------------------+
|       1 | CSK       | Chennai   | Stephen Fleming    |
|       2 | DC        | Delhi     | Ricky Ponting      |
|       3 | GT        | Ahmedabad | Ashish Nehra       |
|       4 | KKR       | Kolkata   | Chandrakant Pandit |
|       5 | LSG       | Lucknow   | Justin Langer      |
|       6 | MI        | Mumbai    | Mark Boucher       |
|       7 | PBKS      | Mohali    | Trevor Bayliss     |
|       8 | RR        | Jaipur    | Kumar Sangakkara   |
|       9 | RCB       | Bengaluru | Andy Flower        |
|      10 | SRH       | Hyderabad | Daniel Vettori     |
+---------+-----------+-----------+--------------------+
10 rows in set (0.00 sec)
```

Table: Points_table

Purpose: Tracks team standings in the tournament.

Structure:



```
mysql> create table points_table(team_id int not null primary key,matches_played int default 0,wins int default 0,losses int default 0,no_result int default
 0,points int default 0,constraint fk_points_team foreign key(team_id) references teams(team_id));
Query OK, 0 rows affected (0.04 sec)

mysql> desc points_table;
+----------------+------+------+-----+---------+-------+
| Field          | Type | Null | Key | Default | Extra |
+----------------+------+------+-----+---------+-------+
| team_id        | int  | NO   | PRI | NULL    |       |
| matches_played | int  | YES  |     | 0       |       |
| wins           | int  | YES  |     | 0       |       |
| losses         | int  | YES  |     | 0       |       |
| no_result      | int  | YES  |     | 0       |       |
| points         | int  | YES  |     | 0       |       |
+----------------+------+------+-----+---------+-------+
6 rows in set (0.00 sec)

mysql> insert into points_table(team_id) select team_id from teams;
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> select * from points_table;
+---------+----------------+------+--------+-----------+--------+
| team_id | matches_played | wins | losses | no_result | points |
+---------+----------------+------+--------+-----------+--------+
|       1 |              0 |    0 |      0 |         0 |      0 |
|       2 |              0 |    0 |      0 |         0 |      0 |
|       3 |              0 |    0 |      0 |         0 |      0 |
|       4 |              0 |    0 |      0 |         0 |      0 |
|       5 |              0 |    0 |      0 |         0 |      0 |
|       6 |              0 |    0 |      0 |         0 |      0 |
|       7 |              0 |    0 |      0 |         0 |      0 |
|       8 |              0 |    0 |      0 |         0 |      0 |
|       9 |              0 |    0 |      0 |         0 |      0 |
|      10 |              0 |    0 |      0 |         0 |      0 |
+---------+----------------+------+--------+-----------+--------+
```

PROCEDURE FOR UPDATING POINTS TABLE BASED ON MATCHES TABLE WITHOUT INSERTING DATA MANUALLY

1. Initialize points_table rows for all teams (if empty)

INSERT IGNORE INTO points_table (team_id)

```
SELECT team_id FROM teams;
```

2. Update matches_played for each team_id

```
UPDATE points_table p
JOIN teams t ON p.team_id = t.team_id
SET p.matches_played = (
   SELECT COUNT(*) FROM (
      SELECT team1 FROM matches WHERE team1 = t.team_name
      UNION ALL
      SELECT team2 FROM matches WHERE team2 = t.team_name
   ) AS matches_played_sub
);
```

3. Update wins for each team_id

```
UPDATE points_table p
JOIN teams t ON p.team_id = t.team_id
SET p.wins = (
   SELECT COUNT(*) FROM matches
   WHERE winner = t.team_name
);
```

4. Update no_result (matches with NULL winner) for each team_id

```
UPDATE points_table p
```

```
JOIN teams t ON p.team_id = t.team_id

SET p.no_result = (

    SELECT COUNT(*) FROM matches

    WHERE (team1 = t.team_name OR team2 = t.team_name)

      AND winner IS NULL

);
```

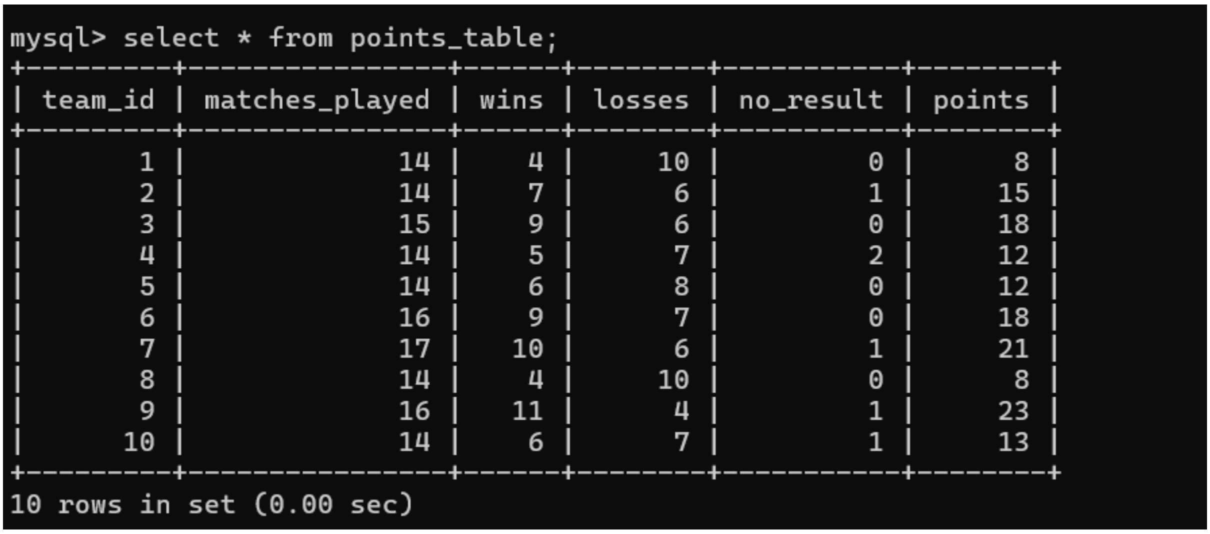
5. Update losses = matches_played - wins - no_result

```
UPDATE points_table

SET losses = matches_played - wins - no_result;
```

6. Update points = wins * 2 + no_result (assuming 2 points for win, 1 for no result)

```
UPDATE points_table

SET points = (wins * 2) + no_result;
```

Sample Data:

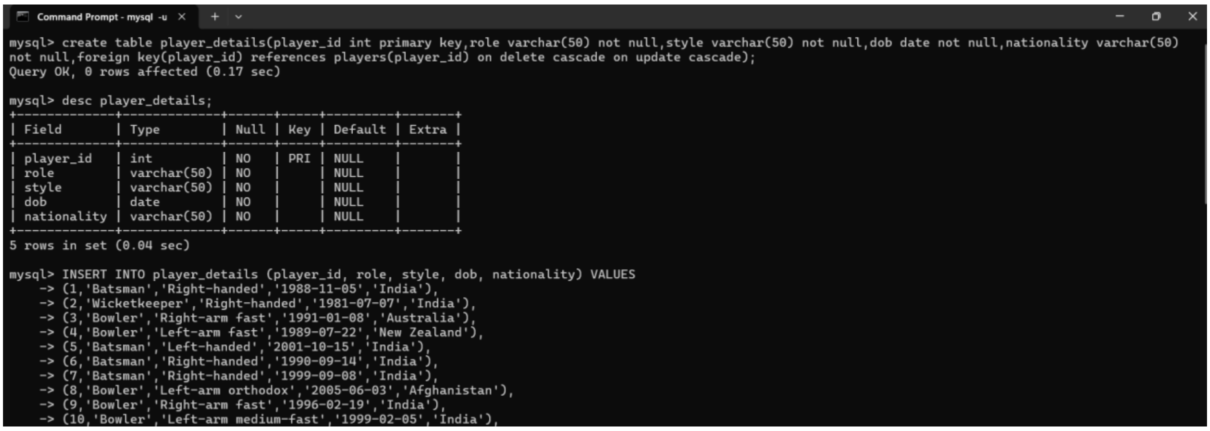


```
mysql> select * from points_table;
+---------+----------------+------+--------+-----------+--------+
| team_id | matches_played | wins | losses | no_result | points |
+---------+----------------+------+--------+-----------+--------+
|       1 |             14 |    4 |     10 |         0 |      8 |
|       2 |             14 |    7 |      6 |         1 |     15 |
|       3 |             15 |    9 |      6 |         0 |     18 |
|       4 |             14 |    5 |      7 |         2 |     12 |
|       5 |             14 |    6 |      8 |         0 |     12 |
|       6 |             16 |    9 |      7 |         0 |     18 |
|       7 |             17 |   10 |      6 |         1 |     21 |
|       8 |             14 |    4 |     10 |         0 |      8 |
|       9 |             16 |   11 |      4 |         1 |     23 |
|      10 |             14 |    6 |      7 |         1 |     13 |
+---------+----------------+------+--------+-----------+--------+
10 rows in set (0.00 sec)
```

Table: player_details

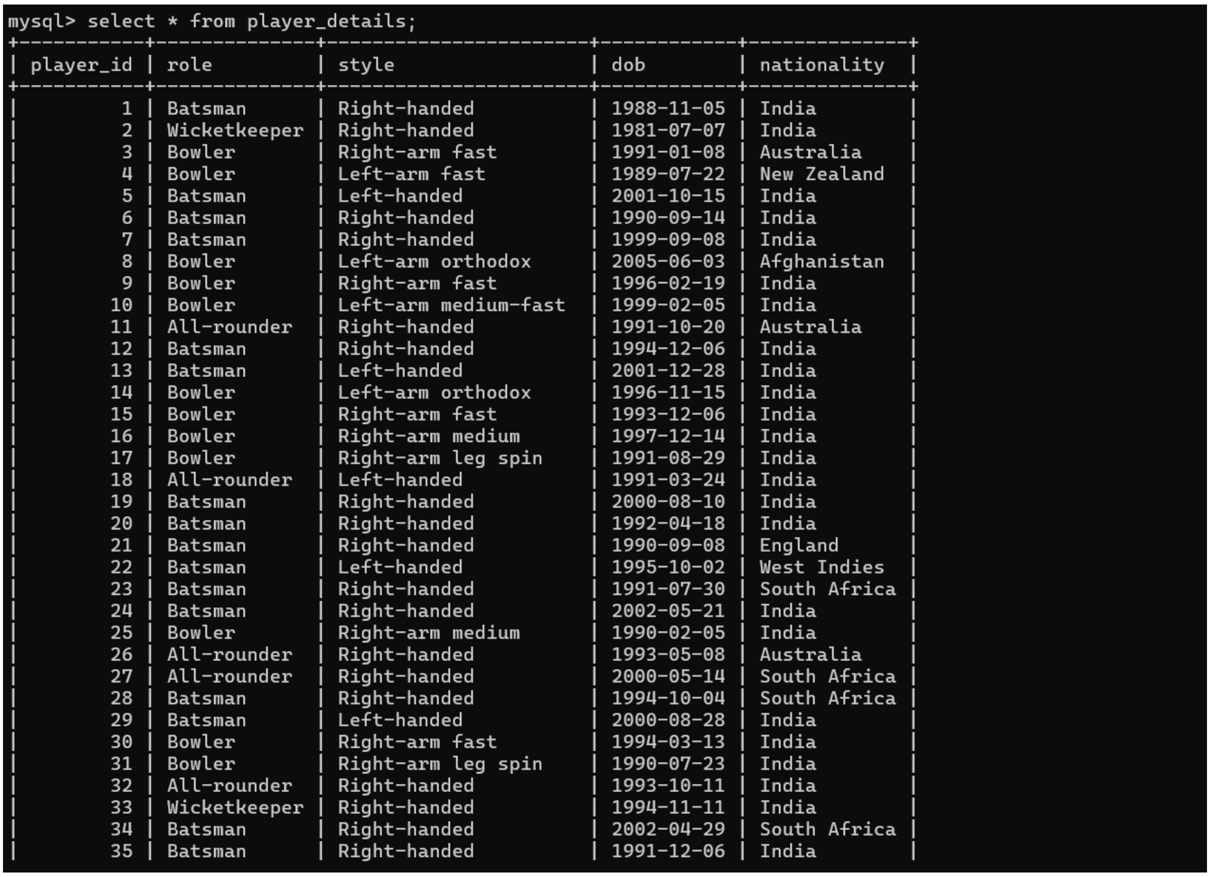
Purpose: Contains player information

Structure:



```
mysql> create table player_details(player_id int primary key,role varchar(50) not null,style varchar(50) not null,dob date not null,nationality varchar(50)
not null,foreign key(player_id) references players(player_id) on delete cascade on update cascade);
Query OK, 0 rows affected (0.17 sec)

mysql> desc player_details;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| player_id   | int         | NO   | PRI | NULL    |       |
| role        | varchar(50) | NO   |     | NULL    |       |
| style       | varchar(50) | NO   |     | NULL    |       |
| dob         | date        | NO   |     | NULL    |       |
| nationality | varchar(50) | NO   |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
5 rows in set (0.04 sec)

mysql> INSERT INTO player_details (player_id, role, style, dob, nationality) VALUES
    -> (1,'Batsman','Right-handed','1988-11-05','India'),
    -> (2,'Wicketkeeper','Right-handed','1981-07-07','India'),
    -> (3,'Bowler','Right-arm fast','1991-01-08','Australia'),
    -> (4,'Bowler','Left-arm fast','1989-07-22','New Zealand'),
    -> (5,'Batsman','Left-handed','2001-10-15','India'),
    -> (6,'Batsman','Right-handed','1990-09-14','India'),
    -> (7,'Batsman','Right-handed','1999-09-08','India'),
    -> (8,'Bowler','Left-arm orthodox','2005-06-03','Afghanistan'),
    -> (9,'Bowler','Right-arm fast','1996-02-19','India'),
    -> (10,'Bowler','Left-arm medium-fast','1999-02-05','India'),
```

Sample Data:

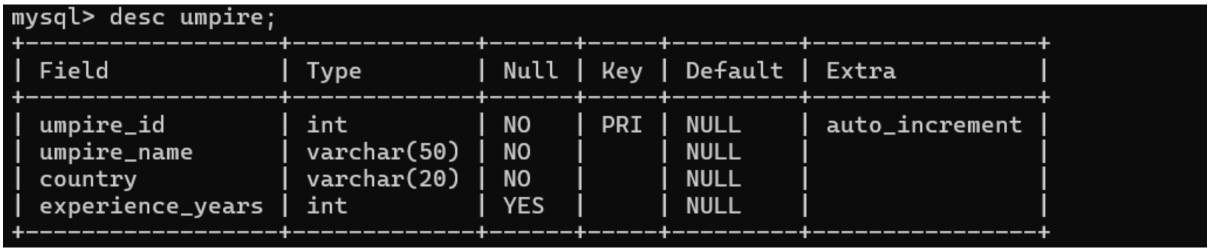


```
mysql> select * from player_details;
+-----------+--------------+-----------------------+------------+--------------+
| player_id | role         | style                 | dob        | nationality  |
+-----------+--------------+-----------------------+------------+--------------+
|         1 | Batsman      | Right-handed          | 1988-11-05 | India        |
|         2 | Wicketkeeper | Right-handed          | 1981-07-07 | India        |
|         3 | Bowler       | Right-arm fast        | 1991-01-08 | Australia    |
|         4 | Bowler       | Left-arm fast         | 1989-07-22 | New Zealand  |
|         5 | Batsman      | Left-handed           | 2001-10-15 | India        |
|         6 | Batsman      | Right-handed          | 1990-09-14 | India        |
|         7 | Batsman      | Right-handed          | 1999-09-08 | India        |
|         8 | Bowler       | Left-arm orthodox     | 2005-06-03 | Afghanistan  |
|         9 | Bowler       | Right-arm fast        | 1996-02-19 | India        |
|        10 | Bowler       | Left-arm medium-fast  | 1999-02-05 | India        |
|        11 | All-rounder  | Right-handed          | 1991-10-20 | Australia    |
|        12 | Batsman      | Right-handed          | 1994-12-06 | India        |
|        13 | Batsman      | Left-handed           | 2001-12-28 | India        |
|        14 | Bowler       | Left-arm orthodox     | 1996-11-15 | India        |
|        15 | Bowler       | Right-arm fast        | 1993-12-06 | India        |
|        16 | Bowler       | Right-arm medium      | 1997-12-14 | India        |
|        17 | Bowler       | Right-arm leg spin    | 1991-08-29 | India        |
|        18 | All-rounder  | Left-handed           | 1991-03-24 | India        |
|        19 | Batsman      | Right-handed          | 2000-08-10 | India        |
|        20 | Batsman      | Right-handed          | 1992-04-18 | India        |
|        21 | Batsman      | Right-handed          | 1990-09-08 | England      |
|        22 | Batsman      | Left-handed           | 1995-10-02 | West Indies  |
|        23 | Batsman      | Right-handed          | 1991-07-30 | South Africa |
|        24 | Batsman      | Right-handed          | 2002-05-21 | India        |
|        25 | Bowler       | Right-arm medium      | 1990-02-05 | India        |
|        26 | All-rounder  | Right-handed          | 1993-05-08 | Australia    |
|        27 | All-rounder  | Right-handed          | 2000-05-14 | South Africa |
|        28 | Batsman      | Right-handed          | 1994-10-04 | South Africa |
|        29 | Batsman      | Left-handed           | 2000-08-28 | India        |
|        30 | Bowler       | Right-arm fast        | 1994-03-13 | India        |
|        31 | Bowler       | Right-arm leg spin    | 1990-07-23 | India        |
|        32 | All-rounder  | Right-handed          | 1993-10-11 | India        |
|        33 | Wicketkeeper | Right-handed          | 1994-11-11 | India        |
|        34 | Batsman      | Right-handed          | 2002-04-29 | South Africa |
|        35 | Batsman      | Right-handed          | 1991-12-06 | India        |
```

Table: Umpire

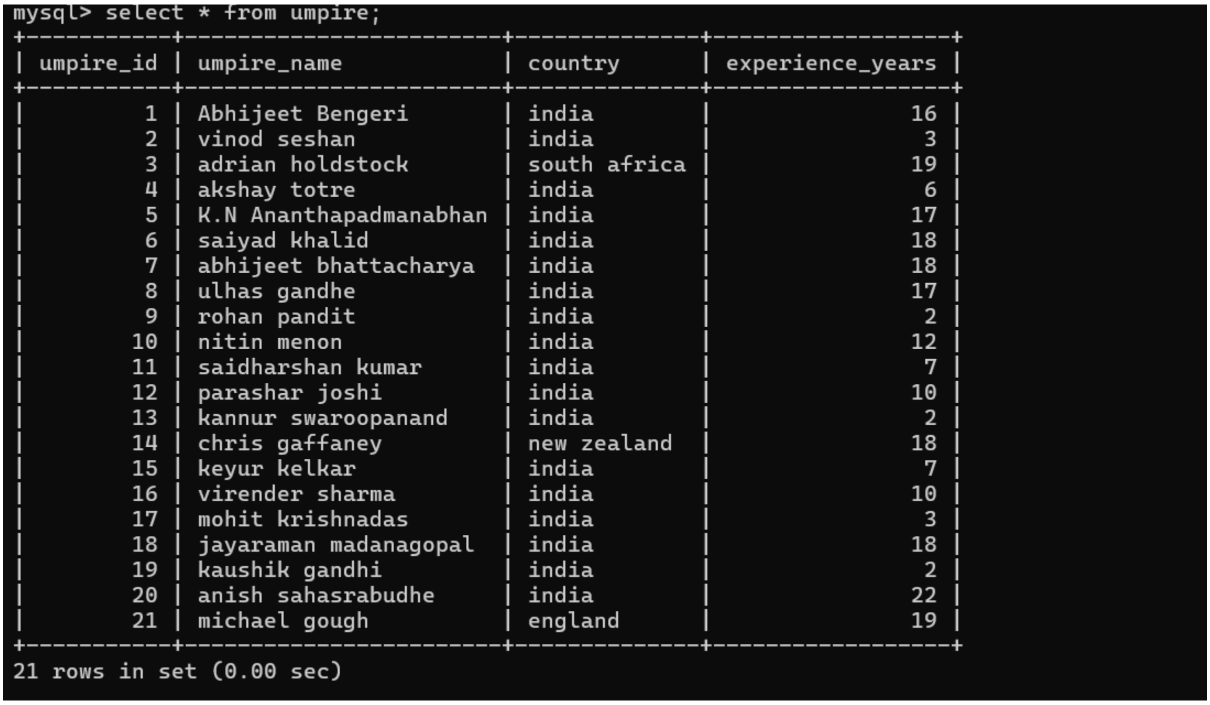
Purpose: Stores umpire details.

Structure:



```
mysql> desc umpire;
+------------------+-------------+------+-----+---------+----------------+
| Field            | Type        | Null | Key | Default | Extra          |
+------------------+-------------+------+-----+---------+----------------+
| umpire_id        | int         | NO   | PRI | NULL    | auto_increment |
| umpire_name      | varchar(50) | NO   |     | NULL    |                |
| country          | varchar(20) | NO   |     | NULL    |                |
| experience_years | int         | YES  |     | NULL    |                |
+------------------+-------------+------+-----+---------+----------------+
```

Sample Data:

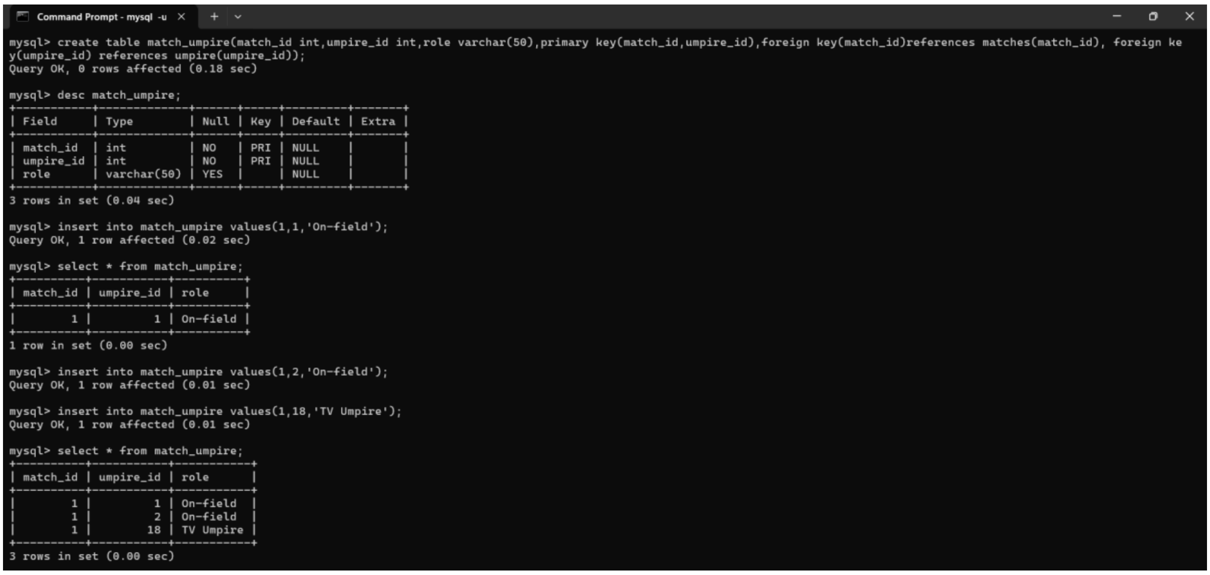


```
mysql> select * from umpire;
+-----------+-------------------------+--------------+------------------+
| umpire_id | umpire_name             | country      | experience_years |
+-----------+-------------------------+--------------+------------------+
|         1 | Abhijeet Bengeri        | india        |               16 |
|         2 | vinod seshan            | india        |                3 |
|         3 | adrian holdstock        | south africa |               19 |
|         4 | akshay totre            | india        |                6 |
|         5 | K.N Ananthapadmanabhan  | india        |               17 |
|         6 | saiyad khalid           | india        |               18 |
|         7 | abhijeet bhattacharya   | india        |               18 |
|         8 | ulhas gandhe            | india        |               17 |
|         9 | rohan pandit            | india        |                2 |
|        10 | nitin menon             | india        |               12 |
|        11 | saidharshan kumar       | india        |                7 |
|        12 | parashar joshi          | india        |               10 |
|        13 | kannur swaroopanand     | india        |                2 |
|        14 | chris gaffaney          | new zealand  |               18 |
|        15 | keyur kelkar            | india        |                7 |
|        16 | virender sharma         | india        |               10 |
|        17 | mohit krishnadas        | india        |                3 |
|        18 | jayaraman madanagopal   | india        |               18 |
|        19 | kaushik gandhi          | india        |                2 |
|        20 | anish sahasrabudhe      | india        |               22 |
|        21 | michael gough           | england      |               19 |
+-----------+-------------------------+--------------+------------------+
21 rows in set (0.00 sec)
```

Table: match_umpire

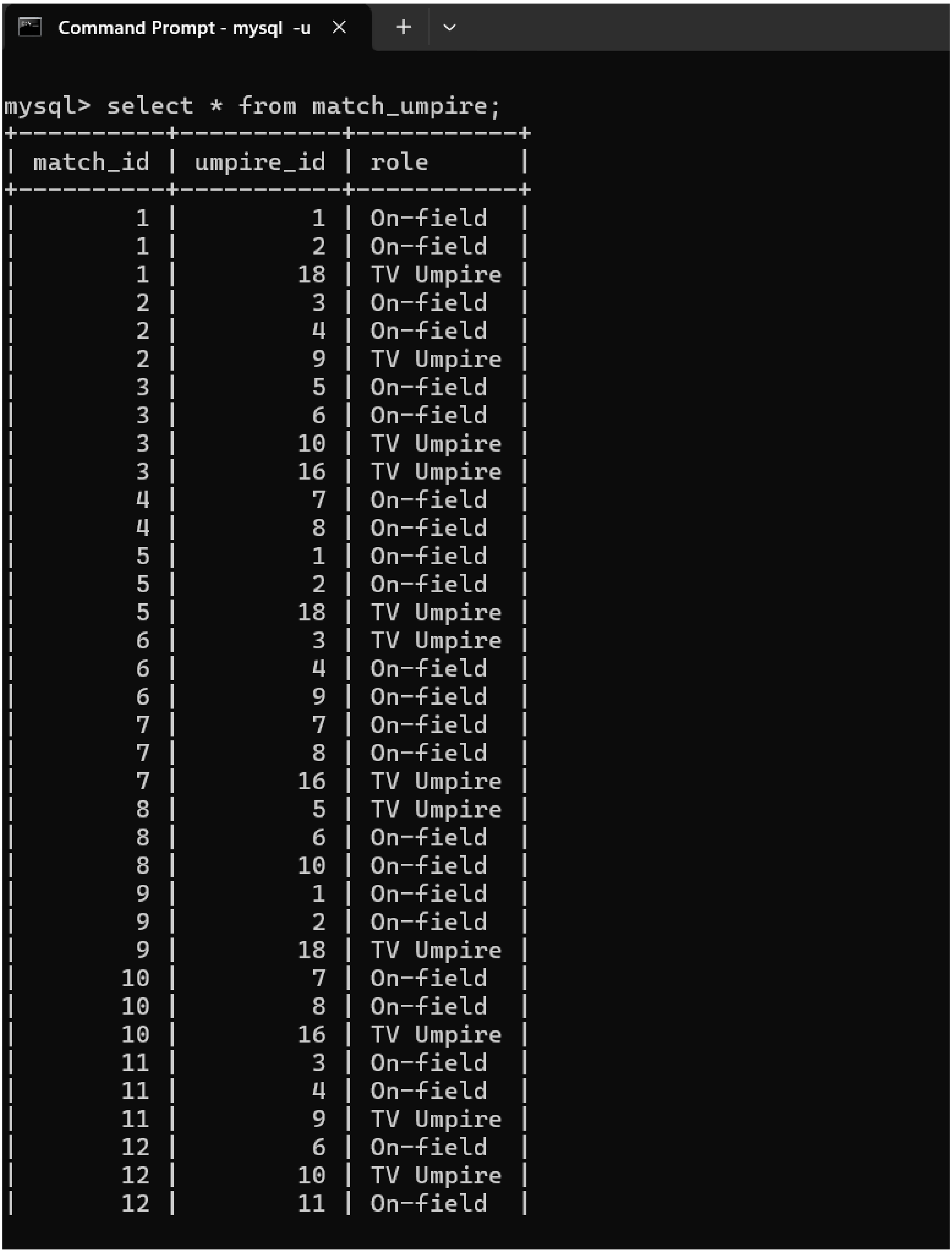
Purpose: Links matches with the umpires assigned.

Structure:



```
mysql> create table match_umpire(match_id int,umpire_id int,role varchar(50),primary key(match_id,umpire_id),foreign key(match_id)references matches(match_id), foreign ke
y(umpire_id) references umpire(umpire_id));
Query OK, 0 rows affected (0.18 sec)

mysql> desc match_umpire;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| match_id  | int         | NO   | PRI | NULL    |       |
| umpire_id | int         | NO   | PRI | NULL    |       |
| role      | varchar(50) | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
3 rows in set (0.04 sec)

mysql> insert into match_umpire values(1,1,'On-field');
Query OK, 1 row affected (0.02 sec)

mysql> select * from match_umpire;
+----------+-----------+----------+
| match_id | umpire_id | role     |
+----------+-----------+----------+
|        1 |         1 | On-field |
+----------+-----------+----------+
1 row in set (0.00 sec)

mysql> insert into match_umpire values(1,2,'On-field');
Query OK, 1 row affected (0.01 sec)

mysql> insert into match_umpire values(1,18,'TV Umpire');
Query OK, 1 row affected (0.01 sec)

mysql> select * from match_umpire;
+----------+-----------+-----------+
| match_id | umpire_id | role      |
+----------+-----------+-----------+
|        1 |         1 | On-field  |
|        1 |         2 | On-field  |
|        1 |        18 | TV Umpire |
+----------+-----------+-----------+
3 rows in set (0.00 sec)
```

Sample Data:

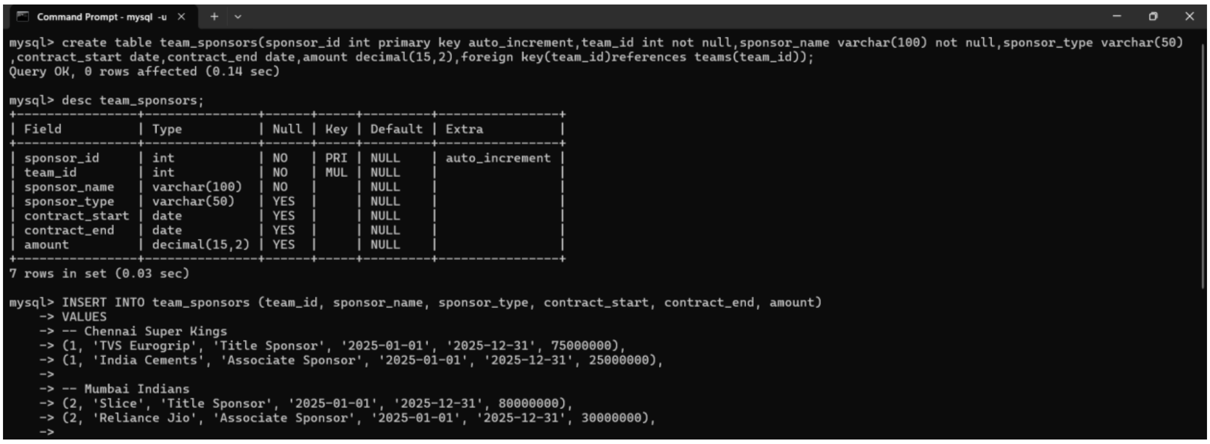


```
mysql> select * from match_umpire;
+----------+-----------+-----------+
| match_id | umpire_id | role      |
+----------+-----------+-----------+
|        1 |         1 | On-field  |
|        1 |         2 | On-field  |
|        1 |        18 | TV Umpire |
|        2 |         3 | On-field  |
|        2 |         4 | On-field  |
|        2 |         9 | TV Umpire |
|        3 |         5 | On-field  |
|        3 |         6 | On-field  |
|        3 |        10 | TV Umpire |
|        3 |        16 | TV Umpire |
|        4 |         7 | On-field  |
|        4 |         8 | On-field  |
|        5 |         1 | On-field  |
|        5 |         2 | On-field  |
|        5 |        18 | TV Umpire |
|        6 |         3 | TV Umpire |
|        6 |         4 | On-field  |
|        6 |         9 | On-field  |
|        7 |         7 | On-field  |
|        7 |         8 | On-field  |
|        7 |        16 | TV Umpire |
|        8 |         5 | TV Umpire |
|        8 |         6 | On-field  |
|        8 |        10 | On-field  |
|        9 |         1 | On-field  |
|        9 |         2 | On-field  |
|        9 |        18 | TV Umpire |
|       10 |         7 | On-field  |
|       10 |         8 | On-field  |
|       10 |        16 | TV Umpire |
|       11 |         3 | On-field  |
|       11 |         4 | On-field  |
|       11 |         9 | TV Umpire |
|       12 |         6 | On-field  |
|       12 |        10 | TV Umpire |
|       12 |        11 | On-field  |
```

Table: team_sponsors

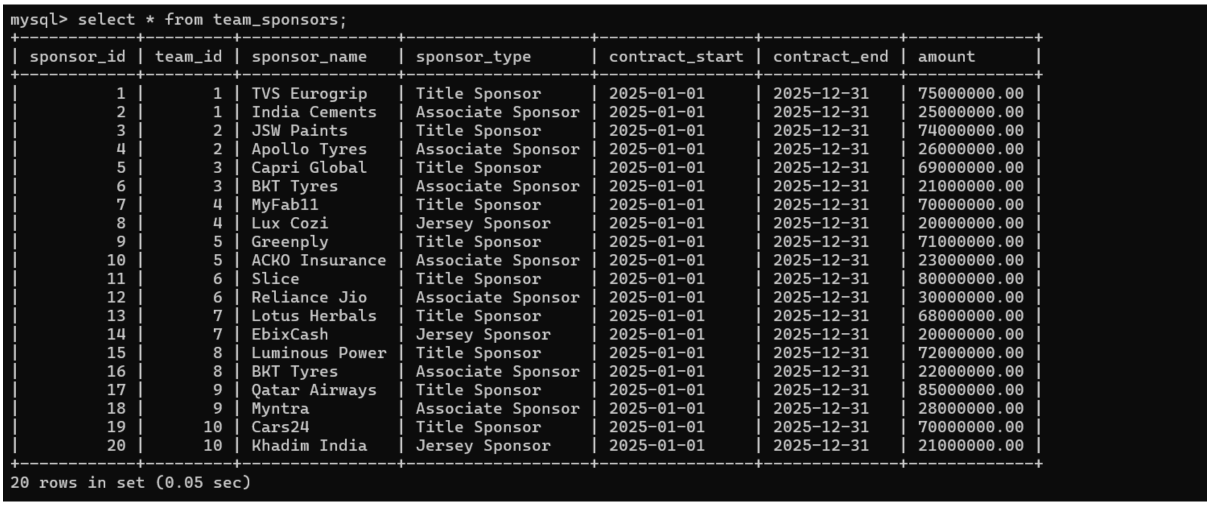
Purpose: Holds sponsor name, type, contract duration, contract value

Structure:



```
mysql> create table team_sponsors(sponsor_id int primary key auto_increment,team_id int not null,sponsor_name varchar(100) not null,sponsor_type varchar(50)
,contract_start date,contract_end date,amount decimal(15,2),foreign key(team_id)references teams(team_id));
Query OK, 0 rows affected (0.14 sec)

mysql> desc team_sponsors;
+----------------+---------------+------+-----+---------+----------------+
| Field          | Type          | Null | Key | Default | Extra          |
+----------------+---------------+------+-----+---------+----------------+
| sponsor_id     | int           | NO   | PRI | NULL    | auto_increment |
| team_id        | int           | NO   | MUL | NULL    |                |
| sponsor_name   | varchar(100)  | NO   |     | NULL    |                |
| sponsor_type   | varchar(50)   | YES  |     | NULL    |                |
| contract_start | date          | YES  |     | NULL    |                |
| contract_end   | date          | YES  |     | NULL    |                |
| amount         | decimal(15,2) | YES  |     | NULL    |                |
+----------------+---------------+------+-----+---------+----------------+
7 rows in set (0.03 sec)

mysql> INSERT INTO team_sponsors (team_id, sponsor_name, sponsor_type, contract_start, contract_end, amount)
    -> VALUES
    -> -- Chennai Super Kings
    -> (1, 'TVS Eurogrip', 'Title Sponsor', '2025-01-01', '2025-12-31', 75000000),
    -> (1, 'India Cements', 'Associate Sponsor', '2025-01-01', '2025-12-31', 25000000),
    ->
    -> -- Mumbai Indians
    -> (2, 'Slice', 'Title Sponsor', '2025-01-01', '2025-12-31', 80000000),
    -> (2, 'Reliance Jio', 'Associate Sponsor', '2025-01-01', '2025-12-31', 30000000),
    ->
```

Sample Data:



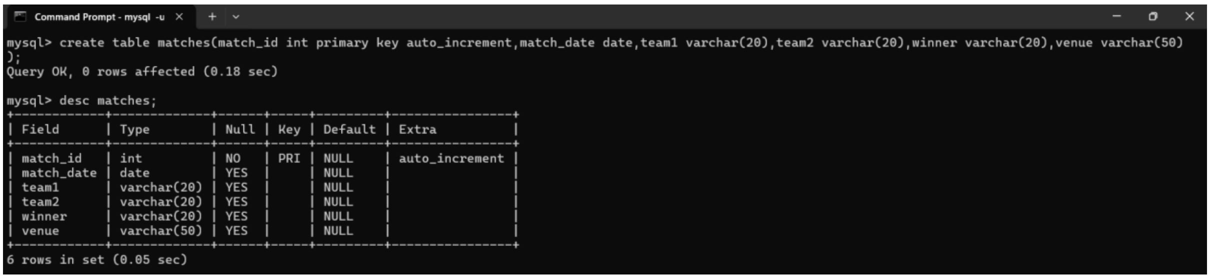
```
mysql> select * from team_sponsors;
+------------+---------+----------------+-------------------+----------------+--------------+-------------+
| sponsor_id | team_id | sponsor_name   | sponsor_type      | contract_start | contract_end | amount      |
+------------+---------+----------------+-------------------+----------------+--------------+-------------+
|          1 |       1 | TVS Eurogrip   | Title Sponsor     | 2025-01-01     | 2025-12-31   | 75000000.00 |
|          2 |       1 | India Cements  | Associate Sponsor | 2025-01-01     | 2025-12-31   | 25000000.00 |
|          3 |       2 | JSW Paints     | Title Sponsor     | 2025-01-01     | 2025-12-31   | 74000000.00 |
|          4 |       2 | Apollo Tyres   | Associate Sponsor | 2025-01-01     | 2025-12-31   | 26000000.00 |
|          5 |       3 | Capri Global   | Title Sponsor     | 2025-01-01     | 2025-12-31   | 69000000.00 |
|          6 |       3 | BKT Tyres      | Associate Sponsor | 2025-01-01     | 2025-12-31   | 21000000.00 |
|          7 |       4 | MyFab11        | Title Sponsor     | 2025-01-01     | 2025-12-31   | 70000000.00 |
|          8 |       4 | Lux Cozi       | Jersey Sponsor    | 2025-01-01     | 2025-12-31   | 20000000.00 |
|          9 |       5 | Greenply       | Title Sponsor     | 2025-01-01     | 2025-12-31   | 71000000.00 |
|         10 |       5 | ACKO Insurance | Associate Sponsor | 2025-01-01     | 2025-12-31   | 23000000.00 |
|         11 |       6 | Slice          | Title Sponsor     | 2025-01-01     | 2025-12-31   | 80000000.00 |
|         12 |       6 | Reliance Jio   | Associate Sponsor | 2025-01-01     | 2025-12-31   | 30000000.00 |
|         13 |       7 | Lotus Herbals  | Title Sponsor     | 2025-01-01     | 2025-12-31   | 68000000.00 |
|         14 |       7 | EbixCash       | Jersey Sponsor    | 2025-01-01     | 2025-12-31   | 20000000.00 |
|         15 |       8 | Luminous Power | Title Sponsor     | 2025-01-01     | 2025-12-31   | 72000000.00 |
|         16 |       8 | BKT Tyres      | Associate Sponsor | 2025-01-01     | 2025-12-31   | 22000000.00 |
|         17 |       9 | Qatar Airways  | Title Sponsor     | 2025-01-01     | 2025-12-31   | 85000000.00 |
|         18 |       9 | Myntra         | Associate Sponsor | 2025-01-01     | 2025-12-31   | 28000000.00 |
|         19 |      10 | Cars24         | Title Sponsor     | 2025-01-01     | 2025-12-31   | 70000000.00 |
|         20 |      10 | Khadim India   | Jersey Sponsor    | 2025-01-01     | 2025-12-31   | 21000000.00 |
+------------+---------+----------------+-------------------+----------------+--------------+-------------+
20 rows in set (0.05 sec)
```

SQL Operations

Data Definition Language: Data Definition Language (DDL) commands are used to define and manage the structure of the database objects such as tables, indexes, and constraints.

CREATE TABLE

Creates a new table with specified columns, data types, and constraints.

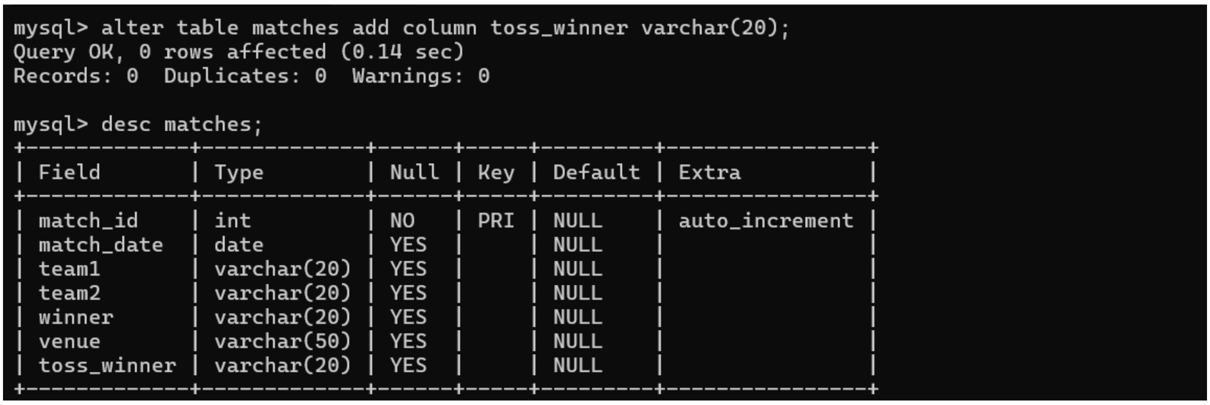


```
mysql> create table matches(match_id int primary key auto_increment,match_date date,team1 varchar(20),team2 varchar(20),winner varchar(20),venue varchar(50)
);
Query OK, 0 rows affected (0.18 sec)

mysql> desc matches;
+------------+-------------+------+-----+---------+----------------+
| Field      | Type        | Null | Key | Default | Extra          |
+------------+-------------+------+-----+---------+----------------+
| match_id   | int         | NO   | PRI | NULL    | auto_increment |
| match_date | date        | YES  |     | NULL    |                |
| team1      | varchar(20) | YES  |     | NULL    |                |
| team2      | varchar(20) | YES  |     | NULL    |                |
| winner     | varchar(20) | YES  |     | NULL    |                |
| venue      | varchar(50) | YES  |     | NULL    |                |
+------------+-------------+------+-----+---------+----------------+
6 rows in set (0.05 sec)
```

ALTER TABLE

Modifies an existing table structure, such as adding or dropping columns or constraints.

Example: Adding a new column to the table by using alter command.

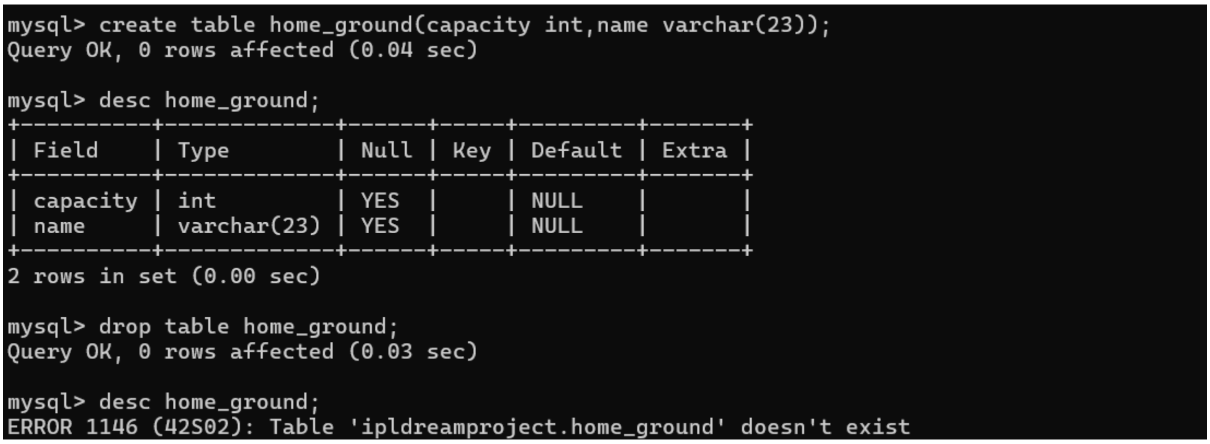


```
mysql> alter table matches add column toss_winner varchar(20);
Query OK, 0 rows affected (0.14 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc matches;
+-------------+-------------+------+-----+---------+----------------+
| Field       | Type        | Null | Key | Default | Extra          |
+-------------+-------------+------+-----+---------+----------------+
| match_id    | int         | NO   | PRI | NULL    | auto_increment |
| match_date  | date        | YES  |     | NULL    |                |
| team1       | varchar(20) | YES  |     | NULL    |                |
| team2       | varchar(20) | YES  |     | NULL    |                |
| winner      | varchar(20) | YES  |     | NULL    |                |
| venue       | varchar(50) | YES  |     | NULL    |                |
| toss_winner | varchar(20) | YES  |     | NULL    |                |
+-------------+-------------+------+-----+---------+----------------+
```

DROP TABLE

Deletes an entire table and its data permanently.

Example: Dropping the home_ground table.

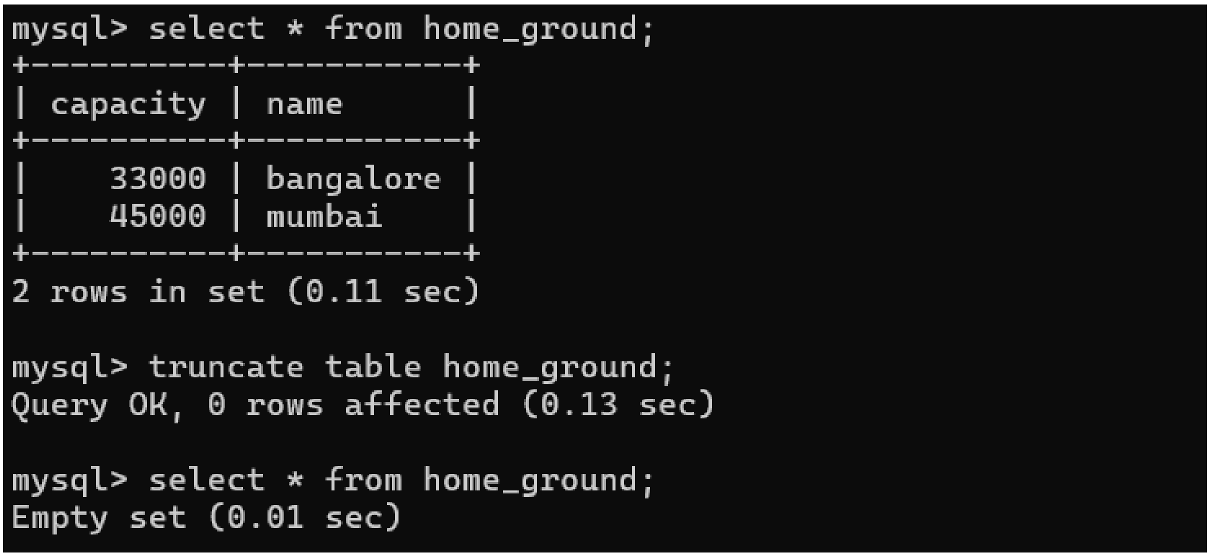


```
mysql> create table home_ground(capacity int,name varchar(23));
Query OK, 0 rows affected (0.04 sec)

mysql> desc home_ground;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| capacity | int         | YES  |     | NULL    |       |
| name     | varchar(23) | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)

mysql> drop table home_ground;
Query OK, 0 rows affected (0.03 sec)

mysql> desc home_ground;
ERROR 1146 (42S02): Table 'ipldreamproject.home_ground' doesn't exist
```

TRUNCATE TABLE

The TRUNCATE TABLE command in SQL is a Data Definition Language (DDL) statement used to quickly remove all rows from a table.

Example: Here when I execute truncate command all records in the table are deleted at a time.



```
mysql> select * from home_ground;
+----------+-----------+
| capacity | name      |
+----------+-----------+
|    33000 | bangalore |
|    45000 | mumbai    |
+----------+-----------+
2 rows in set (0.11 sec)

mysql> truncate table home_ground;
Query OK, 0 rows affected (0.13 sec)

mysql> select * from home_ground;
Empty set (0.01 sec)
```

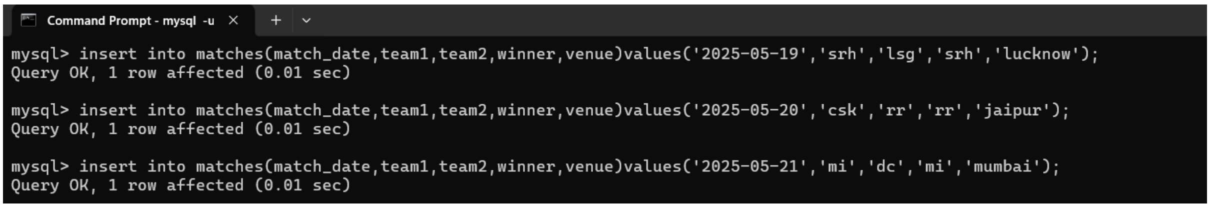
Data Manipulation Language (DML)

Data Manipulation Language (DML) commands are used to manage the data within database tables. These commands allow you to insert, update, delete, and retrieve data, enabling dynamic interaction with the database contents.

INSERT INTO

Adds new rows (records) to a table.

Example: Inserting records into the matches table

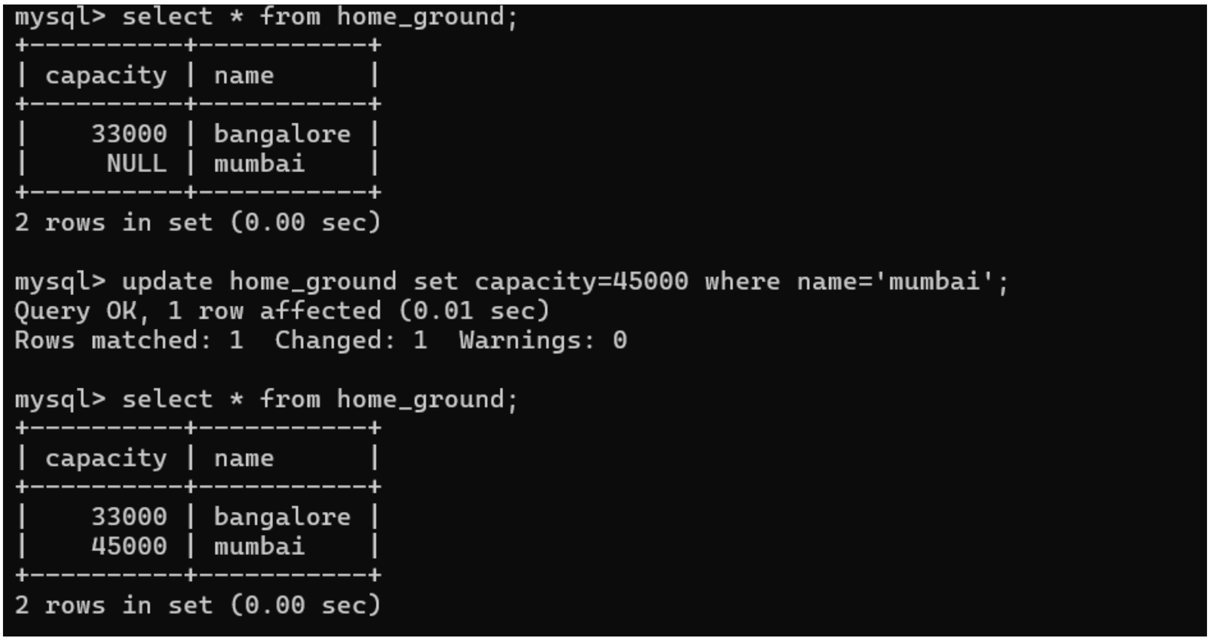


```
mysql> insert into matches(match_date,team1,team2,winner,venue)values('2025-05-19','srh','lsg','srh','lucknow');
Query OK, 1 row affected (0.01 sec)

mysql> insert into matches(match_date,team1,team2,winner,venue)values('2025-05-20','csk','rr','rr','jaipur');
Query OK, 1 row affected (0.01 sec)

mysql> insert into matches(match_date,team1,team2,winner,venue)values('2025-05-21','mi','dc','mi','mumbai');
Query OK, 1 row affected (0.01 sec)
```

UPDATE

Modifies existing records in a table.

Example: Updating the capacity of venue Mumbai.

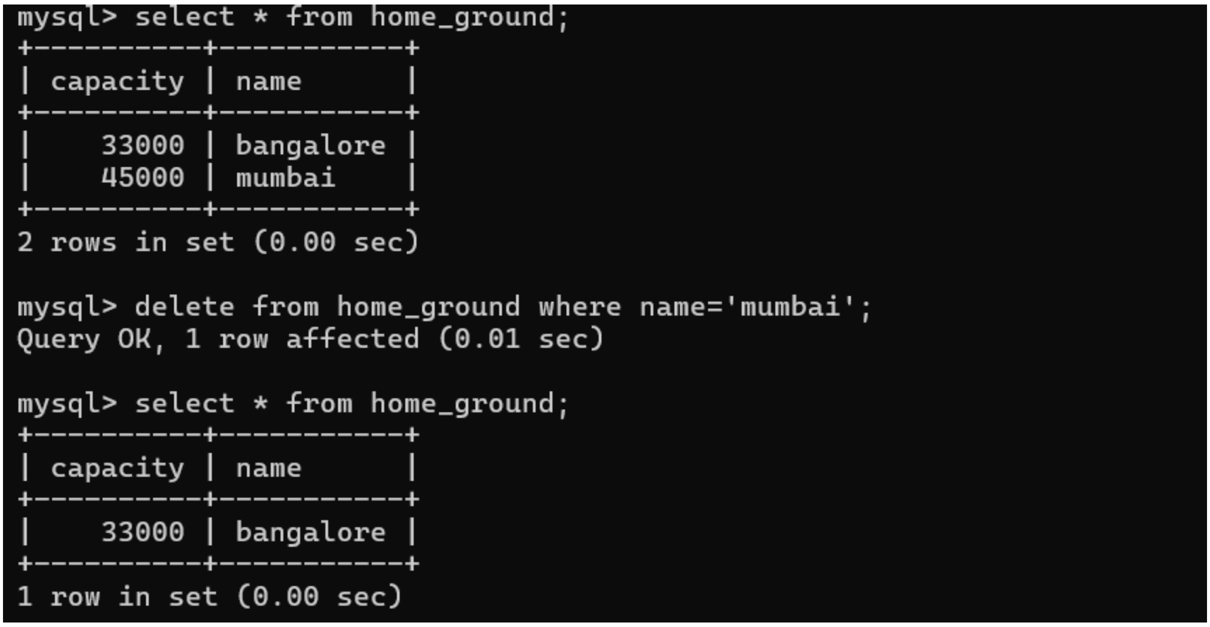


```
mysql> select * from home_ground;
+----------+-----------+
| capacity | name      |
+----------+-----------+
|    33000 | bangalore |
|     NULL | mumbai    |
+----------+-----------+
2 rows in set (0.00 sec)

mysql> update home_ground set capacity=45000 where name='mumbai';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from home_ground;
+----------+-----------+
| capacity | name      |
+----------+-----------+
|    33000 | bangalore |
|    45000 | mumbai    |
+----------+-----------+
2 rows in set (0.00 sec)
```

DELETE

Removes records from a table.

Example: Deleting a venue record.

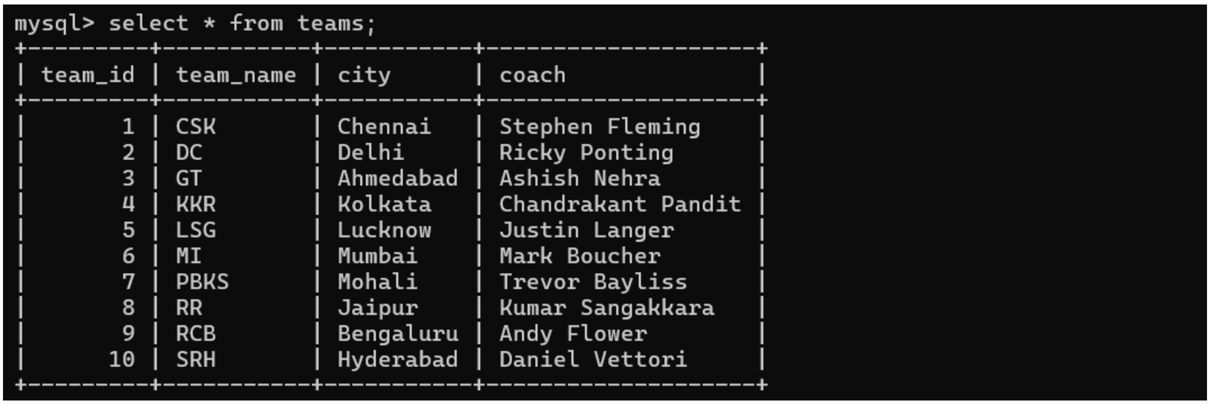


```
mysql> select * from home_ground;
+----------+-----------+
| capacity | name      |
+----------+-----------+
|    33000 | bangalore |
|    45000 | mumbai    |
+----------+-----------+
2 rows in set (0.00 sec)

mysql> delete from home_ground where name='mumbai';
Query OK, 1 row affected (0.01 sec)

mysql> select * from home_ground;
+----------+-----------+
| capacity | name      |
+----------+-----------+
|    33000 | bangalore |
+----------+-----------+
1 row in set (0.00 sec)
```

SELECT

Retrieves data from one or more tables.

Example: Selecting all records from teams table.



```
mysql> select * from teams;
+---------+-----------+-----------+--------------------+
| team_id | team_name | city      | coach              |
+---------+-----------+-----------+--------------------+
|       1 | CSK       | Chennai   | Stephen Fleming    |
|       2 | DC        | Delhi     | Ricky Ponting      |
|       3 | GT        | Ahmedabad | Ashish Nehra       |
|       4 | KKR       | Kolkata   | Chandrakant Pandit |
|       5 | LSG       | Lucknow   | Justin Langer      |
|       6 | MI        | Mumbai    | Mark Boucher       |
|       7 | PBKS      | Mohali    | Trevor Bayliss     |
|       8 | RR        | Jaipur    | Kumar Sangakkara   |
|       9 | RCB       | Bengaluru | Andy Flower        |
|      10 | SRH       | Hyderabad | Daniel Vettori     |
+---------+-----------+-----------+--------------------+
```

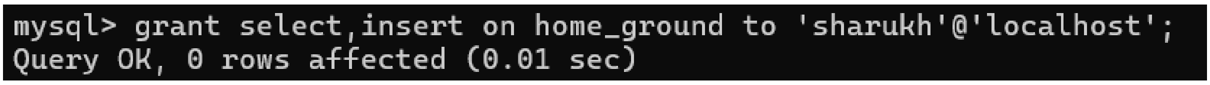
Data Control Language (DCL)

Data Control Language (DCL) commands are used to control access and permissions on the database objects. They help manage security by granting or revoking privileges to users, ensuring that only authorized personnel can perform certain operations.

GRANT

Gives specific privileges to users or roles on database objects like tables or the entire database.

Example: Grant SELECT and INSERT privileges on the home_ground table to a user named Sharukh.

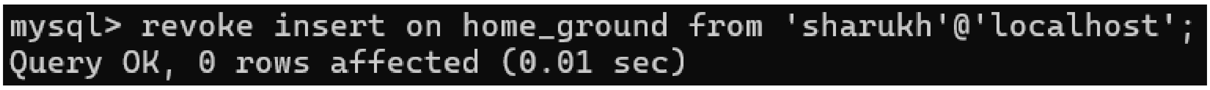


```
mysql> grant select,insert on home_ground to 'sharukh'@'localhost';
Query OK, 0 rows affected (0.01 sec)
```

REVOKE

Removes specific privileges from users or roles.

Example: Revoke INSERT privilege on the home_ground table from the user Sharukh.



```
mysql> revoke insert on home_ground from 'sharukh'@'localhost';
Query OK, 0 rows affected (0.01 sec)
```

Transaction Control Language (TCL)

Transaction Control Language (TCL) commands manage transactions within the database. A **transaction** is a sequence of one or more SQL operations executed as a single unit of work. TCL commands ensure data integrity by allowing you to **commit** or **rollback** changes based on success or failure of operations.

COMMIT

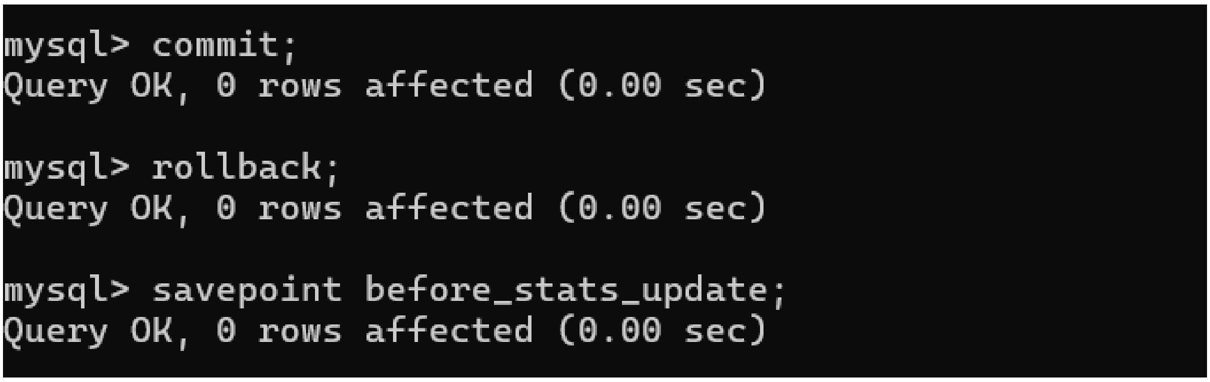
Saves all changes made during the current transaction permanently to the database.

ROLLBACK

Undoes all changes made during the current transaction, reverting the database to its previous state.

SAVEPOINT

Sets a savepoint within a transaction to which you can later rollback without undoing the entire transaction.



```
mysql> commit;
Query OK, 0 rows affected (0.00 sec)

mysql> rollback;
Query OK, 0 rows affected (0.00 sec)

mysql> savepoint before_stats_update;
Query OK, 0 rows affected (0.00 sec)
```

Joins

A **JOIN** in SQL is used to combine rows from two or more tables based on a related column between them.

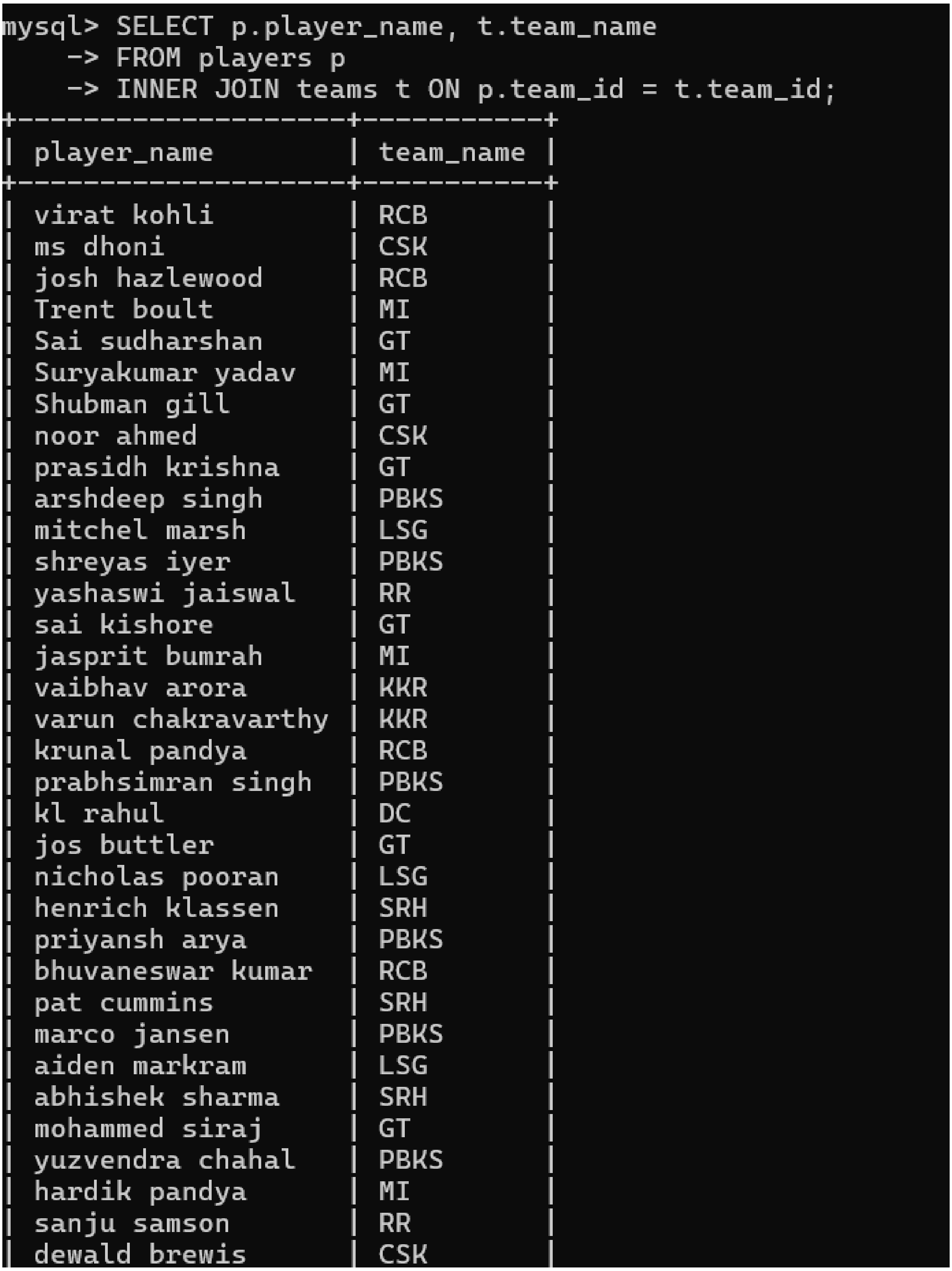
Joins help in retrieving meaningful information that is spread across multiple tables.

Types of Joins

INNER JOIN

Returns records that have matching values in both tables.

Example: List players with their team names

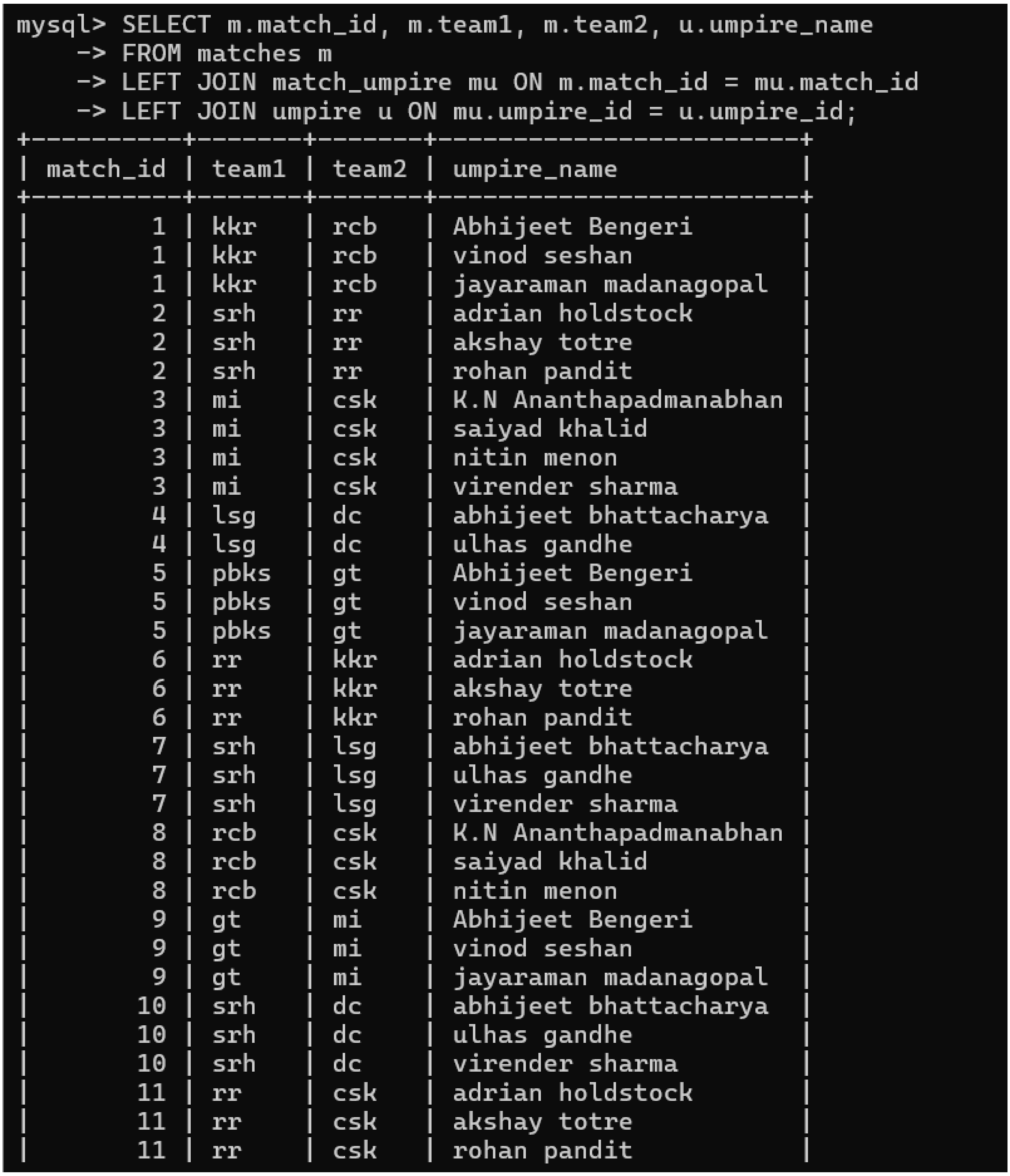


```
mysql> SELECT p.player_name, t.team_name
    -> FROM players p
    -> INNER JOIN teams t ON p.team_id = t.team_id;
+--------------------+-----------+
| player_name        | team_name |
+--------------------+-----------+
| virat kohli        | RCB       |
| ms dhoni           | CSK       |
| josh hazlewood     | RCB       |
| Trent boult        | MI        |
| Sai sudharshan     | GT        |
| Suryakumar yadav   | MI        |
| Shubman gill       | GT        |
| noor ahmed         | CSK       |
| prasidh krishna    | GT        |
| arshdeep singh     | PBKS      |
| mitchel marsh      | LSG       |
| shreyas iyer       | PBKS      |
| yashaswi jaiswal   | RR        |
| sai kishore        | GT        |
| jasprit bumrah     | MI        |
| vaibhav arora      | KKR       |
| varun chakravarthy | KKR       |
| krunal pandya      | RCB       |
| prabhsimran singh  | PBKS      |
| kl rahul           | DC        |
| jos buttler        | GT        |
| nicholas pooran    | LSG       |
| henrich klassen    | SRH       |
| priyansh arya      | PBKS      |
| bhuvaneswar kumar  | RCB       |
| pat cummins        | SRH       |
| marco jansen       | PBKS      |
| aiden markram      | LSG       |
| abhishek sharma    | SRH       |
| mohammed siraj     | GT        |
| yuzvendra chahal   | PBKS      |
| hardik pandya      | MI        |
| sanju samson       | RR        |
| dewald brewis      | CSK       |
```

LEFT JOIN

Returns all records from the left table, and the matched records from the right table. If no match, right table columns return NULL.

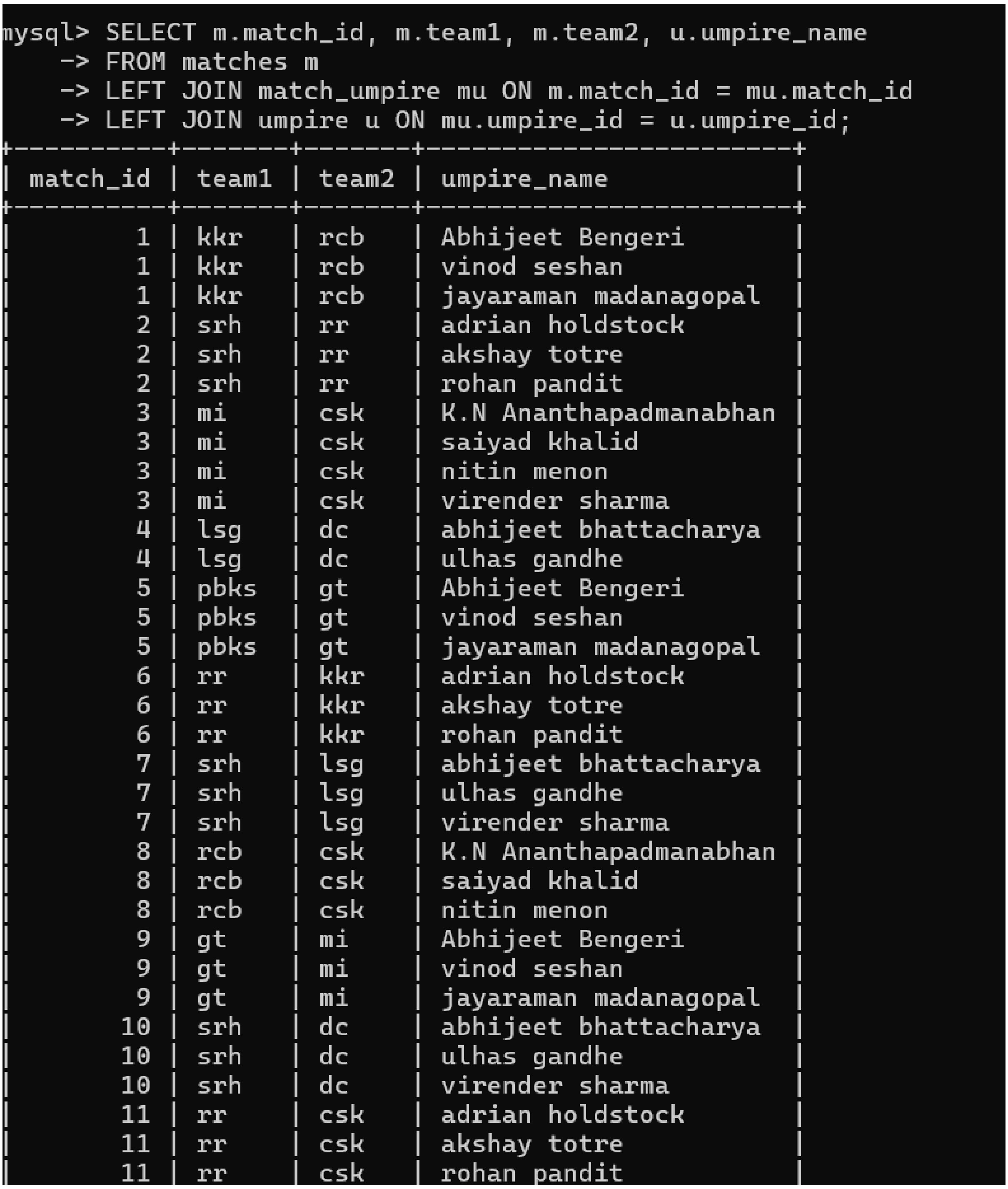
Example: List all matches and their umpires.



```
mysql> SELECT m.match_id, m.team1, m.team2, u.umpire_name
    -> FROM matches m
    -> LEFT JOIN match_umpire mu ON m.match_id = mu.match_id
    -> LEFT JOIN umpire u ON mu.umpire_id = u.umpire_id;
+----------+-------+-------+------------------------+
| match_id | team1 | team2 | umpire_name            |
+----------+-------+-------+------------------------+
|        1 | kkr   | rcb   | Abhijeet Bengeri       |
|        1 | kkr   | rcb   | vinod seshan           |
|        1 | kkr   | rcb   | jayaraman madanagopal  |
|        2 | srh   | rr    | adrian holdstock       |
|        2 | srh   | rr    | akshay totre           |
|        2 | srh   | rr    | rohan pandit           |
|        3 | mi    | csk   | K.N Ananthapadmanabhan |
|        3 | mi    | csk   | saiyad khalid          |
|        3 | mi    | csk   | nitin menon            |
|        3 | mi    | csk   | virender sharma        |
|        4 | lsg   | dc    | abhijeet bhattacharya  |
|        4 | lsg   | dc    | ulhas gandhe           |
|        5 | pbks  | gt    | Abhijeet Bengeri       |
|        5 | pbks  | gt    | vinod seshan           |
|        5 | pbks  | gt    | jayaraman madanagopal  |
|        6 | rr    | kkr   | adrian holdstock       |
|        6 | rr    | kkr   | akshay totre           |
|        6 | rr    | kkr   | rohan pandit           |
|        7 | srh   | lsg   | abhijeet bhattacharya  |
|        7 | srh   | lsg   | ulhas gandhe           |
|        7 | srh   | lsg   | virender sharma        |
|        8 | rcb   | csk   | K.N Ananthapadmanabhan |
|        8 | rcb   | csk   | saiyad khalid          |
|        8 | rcb   | csk   | nitin menon            |
|        9 | gt    | mi    | Abhijeet Bengeri       |
|        9 | gt    | mi    | vinod seshan           |
|        9 | gt    | mi    | jayaraman madanagopal  |
|       10 | srh   | dc    | abhijeet bhattacharya  |
|       10 | srh   | dc    | ulhas gandhe           |
|       10 | srh   | dc    | virender sharma        |
|       11 | rr    | csk   | adrian holdstock       |
|       11 | rr    | csk   | akshay totre           |
|       11 | rr    | csk   | rohan pandit           |
```

RIGHT JOIN

Returns all records from the right table, and matched records from the left. If no match, left table columns return NULL.

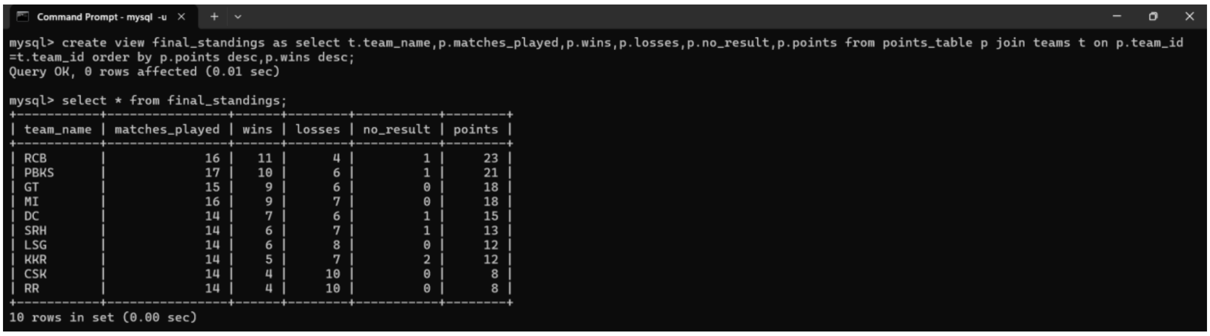


```
mysql> SELECT m.match_id, m.team1, m.team2, u.umpire_name
    -> FROM matches m
    -> LEFT JOIN match_umpire mu ON m.match_id = mu.match_id
    -> LEFT JOIN umpire u ON mu.umpire_id = u.umpire_id;
+----------+-------+-------+------------------------+
| match_id | team1 | team2 | umpire_name            |
+----------+-------+-------+------------------------+
|        1 | kkr   | rcb   | Abhijeet Bengeri       |
|        1 | kkr   | rcb   | vinod seshan           |
|        1 | kkr   | rcb   | jayaraman madanagopal  |
|        2 | srh   | rr    | adrian holdstock       |
|        2 | srh   | rr    | akshay totre           |
|        2 | srh   | rr    | rohan pandit           |
|        3 | mi    | csk   | K.N Ananthapadmanabhan |
|        3 | mi    | csk   | saiyad khalid          |
|        3 | mi    | csk   | nitin menon            |
|        3 | mi    | csk   | virender sharma        |
|        4 | lsg   | dc    | abhijeet bhattacharya  |
|        4 | lsg   | dc    | ulhas gandhe           |
|        5 | pbks  | gt    | Abhijeet Bengeri       |
|        5 | pbks  | gt    | vinod seshan           |
|        5 | pbks  | gt    | jayaraman madanagopal  |
|        6 | rr    | kkr   | adrian holdstock       |
|        6 | rr    | kkr   | akshay totre           |
|        6 | rr    | kkr   | rohan pandit           |
|        7 | srh   | lsg   | abhijeet bhattacharya  |
|        7 | srh   | lsg   | ulhas gandhe           |
|        7 | srh   | lsg   | virender sharma        |
|        8 | rcb   | csk   | K.N Ananthapadmanabhan |
|        8 | rcb   | csk   | saiyad khalid          |
|        8 | rcb   | csk   | nitin menon            |
|        9 | gt    | mi    | Abhijeet Bengeri       |
|        9 | gt    | mi    | vinod seshan           |
|        9 | gt    | mi    | jayaraman madanagopal  |
|       10 | srh   | dc    | abhijeet bhattacharya  |
|       10 | srh   | dc    | ulhas gandhe           |
|       10 | srh   | dc    | virender sharma        |
|       11 | rr    | csk   | adrian holdstock       |
|       11 | rr    | csk   | akshay totre           |
|       11 | rr    | csk   | rohan pandit           |
```

Views

A **View** is a virtual table in SQL based on the result-set of a stored query. Views simplify complex queries by encapsulating them and provide an easy way to access frequently used data without rewriting the query.

Example: View for final points standings

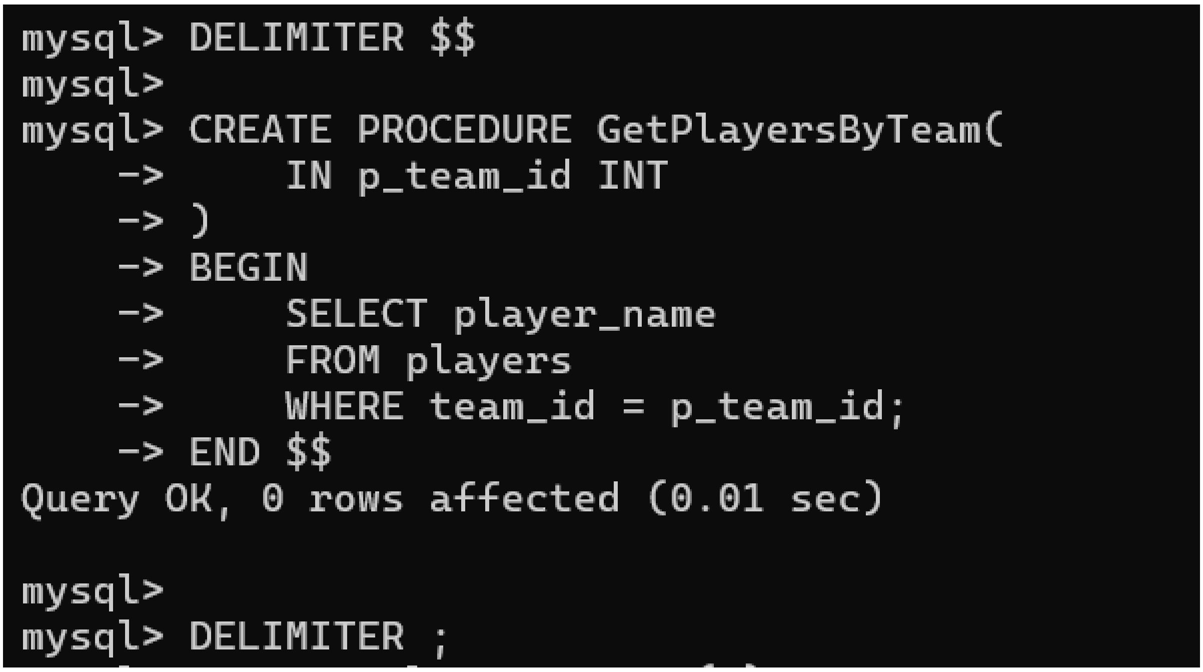


```
mysql> create view final_standings as select t.team_name,p.matches_played,p.wins,p.losses,p.no_result,p.points from points_table p join teams t on p.team_id
=t.team_id order by p.points desc,p.wins desc;
Query OK, 0 rows affected (0.01 sec)

mysql> select * from final_standings;
+-----------+----------------+------+--------+-----------+--------+
| team_name | matches_played | wins | losses | no_result | points |
+-----------+----------------+------+--------+-----------+--------+
| RCB       |             16 |   11 |      4 |         1 |     23 |
| PBKS      |             17 |   10 |      6 |         1 |     21 |
| GT        |             15 |    9 |      6 |         0 |     18 |
| MI        |             16 |    9 |      7 |         0 |     18 |
| DC        |             14 |    7 |      6 |         1 |     15 |
| SRH       |             14 |    6 |      7 |         1 |     13 |
| LSG       |             14 |    6 |      8 |         0 |     12 |
| KKR       |             14 |    5 |      7 |         2 |     12 |
| CSK       |             14 |    4 |     10 |         0 |      8 |
| RR        |             14 |    4 |     10 |         0 |      8 |
+-----------+----------------+------+--------+-----------+--------+
10 rows in set (0.00 sec)
```

Stored Procedures

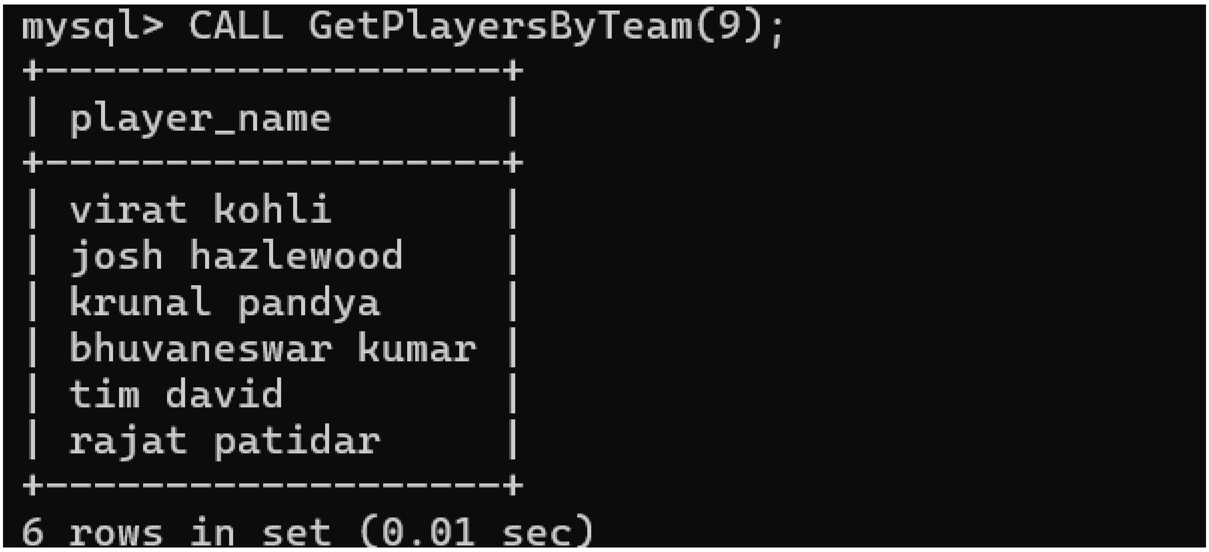
A **Stored Procedure** is a set of SQL statements that are saved and executed as a single unit on the database server. Stored procedures help to encapsulate logic, improve performance, and allow code reuse.

Example: creating a stored procedure that takes a team_id as input and returns the list of player names in that team.



```
mysql> DELIMITER $$
mysql>
mysql> CREATE PROCEDURE GetPlayersByTeam(
    ->     IN p_team_id INT
    -> )
    -> BEGIN
    ->     SELECT player_name
    ->     FROM players
    ->     WHERE team_id = p_team_id;
    -> END $$
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> DELIMITER ;
```

When we call this procedure it will return all player names who belong to the team with team_id = 9.



```
mysql> CALL GetPlayersByTeam(9);
+--------------------+
| player_name        |
+--------------------+
| virat kohli        |
| josh hazlewood     |
| krunal pandya      |
| bhuvaneswar kumar  |
| tim david          |
| rajat patidar      |
+--------------------+
6 rows in set (0.01 sec)
```

Triggers

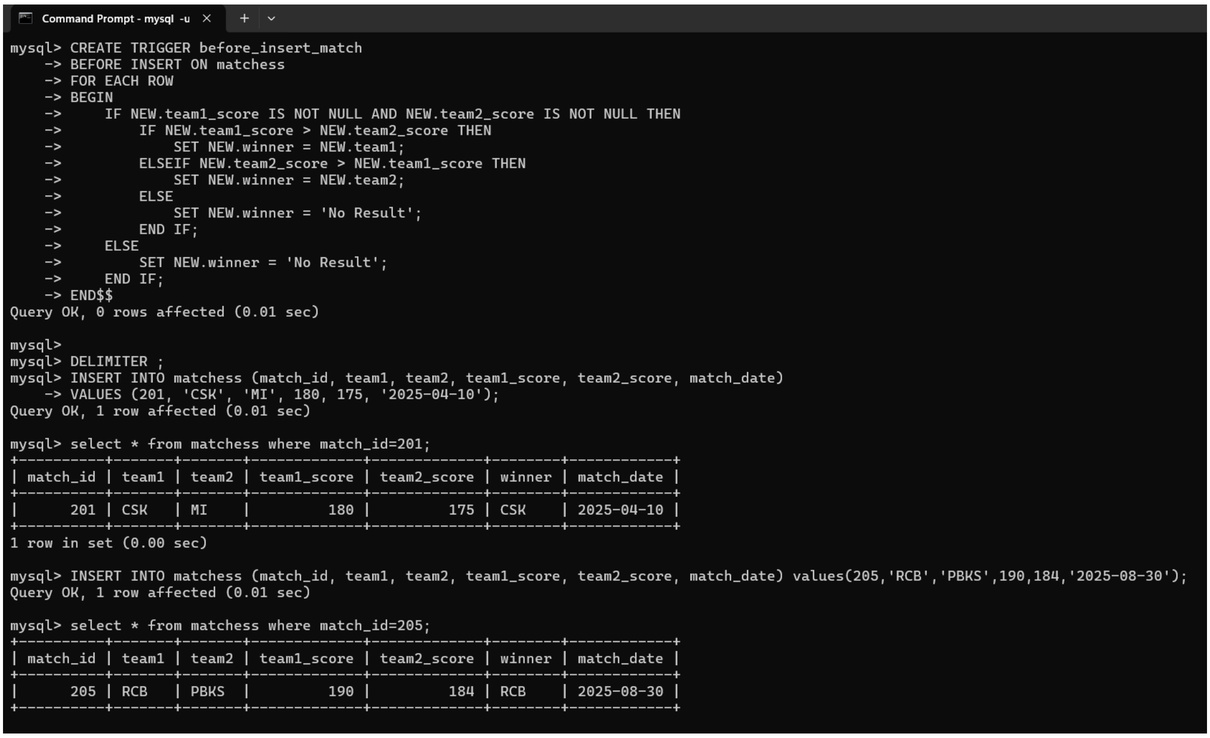
A **Trigger** is a special type of stored procedure that automatically executes or "fires" when a specific database event occurs, such as inserting, updating, or deleting data in a table. Triggers help enforce business rules, maintain data integrity, and automate system tasks.

Types of Triggers

- **BEFORE INSERT:** Executes before a new record is inserted.
- **AFTER INSERT:** Executes after a new record is inserted.
- **BEFORE UPDATE:** Executes before a record is updated.
- **AFTER UPDATE:** Executes after a record is updated.
- **BEFORE DELETE:** Executes before a record is deleted.
- **AFTER DELETE:** Executes after a record is deleted.

BEFORE INSERT Trigger Example

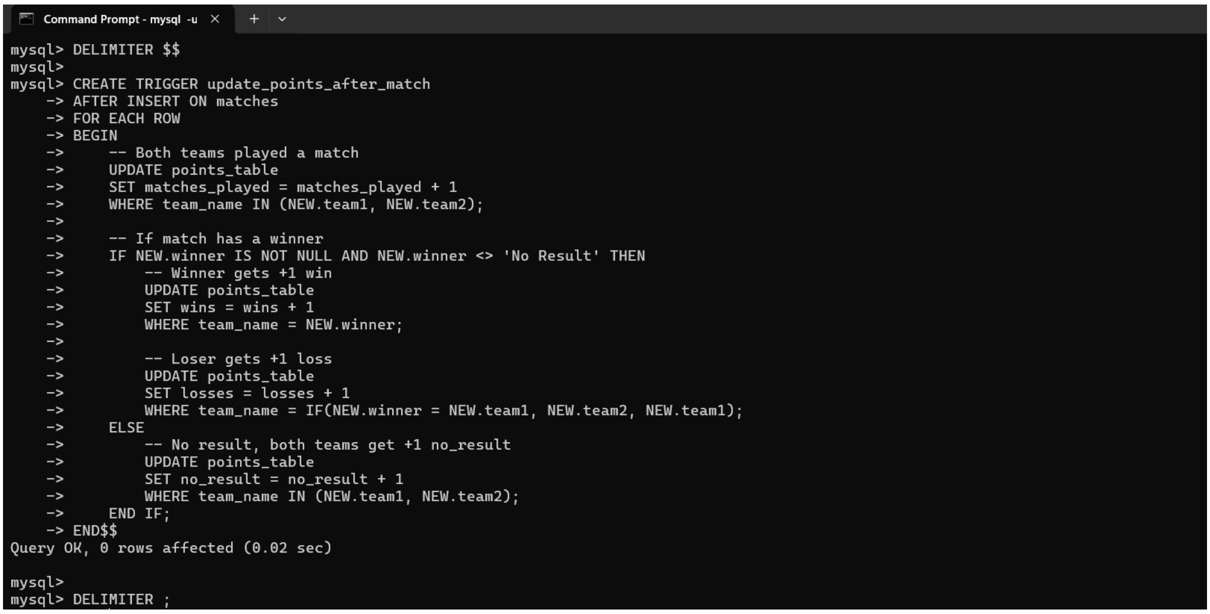
a BEFORE INSERT trigger can be used to automatically determine the winner of a match based on scores before the record is saved in the matches table.



```
mysql> CREATE TRIGGER before_insert_match
    -> BEFORE INSERT ON matchess
    -> FOR EACH ROW
    -> BEGIN
    ->     IF NEW.team1_score IS NOT NULL AND NEW.team2_score IS NOT NULL THEN
    ->         IF NEW.team1_score > NEW.team2_score THEN
    ->             SET NEW.winner = NEW.team1;
    ->         ELSEIF NEW.team2_score > NEW.team1_score THEN
    ->             SET NEW.winner = NEW.team2;
    ->         ELSE
    ->             SET NEW.winner = 'No Result';
    ->         END IF;
    ->     ELSE
    ->         SET NEW.winner = 'No Result';
    ->     END IF;
    -> END$$
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> DELIMITER ;
mysql> INSERT INTO matchess (match_id, team1, team2, team1_score, team2_score, match_date)
    -> VALUES (201, 'CSK', 'MI', 180, 175, '2025-04-10');
Query OK, 1 row affected (0.01 sec)

mysql> select * from matchess where match_id=201;
+----------+-------+-------+-------------+-------------+--------+------------+
| match_id | team1 | team2 | team1_score | team2_score | winner | match_date |
+----------+-------+-------+-------------+-------------+--------+------------+
|      201 | CSK   | MI    |         180 |         175 | CSK    | 2025-04-10 |
+----------+-------+-------+-------------+-------------+--------+------------+
1 row in set (0.00 sec)

mysql> INSERT INTO matchess (match_id, team1, team2, team1_score, team2_score, match_date) values(205,'RCB','PBKS',190,184,'2025-08-30');
Query OK, 1 row affected (0.01 sec)

mysql> select * from matchess where match_id=205;
+----------+-------+-------+-------------+-------------+--------+------------+
| match_id | team1 | team2 | team1_score | team2_score | winner | match_date |
+----------+-------+-------+-------------+-------------+--------+------------+
|      205 | RCB   | PBKS  |         190 |         184 | RCB    | 2025-08-30 |
+----------+-------+-------+-------------+-------------+--------+------------+
```

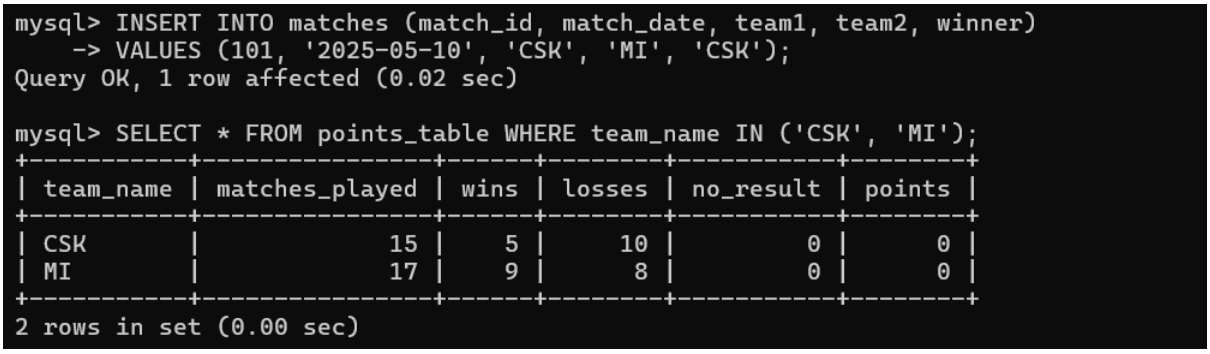
AFTER INSERT Trigger Example

Suppose you want to automatically update the total number of matches played by a team whenever a new match record is inserted involving that team.



```
mysql> DELIMITER $$
mysql>
mysql> CREATE TRIGGER update_points_after_match
    -> AFTER INSERT ON matches
    -> FOR EACH ROW
    -> BEGIN
    ->     -- Both teams played a match
    ->     UPDATE points_table
    ->     SET matches_played = matches_played + 1
    ->     WHERE team_name IN (NEW.team1, NEW.team2);
    ->
    ->     -- If match has a winner
    ->     IF NEW.winner IS NOT NULL AND NEW.winner <> 'No Result' THEN
    ->         -- Winner gets +1 win
    ->         UPDATE points_table
    ->         SET wins = wins + 1
    ->         WHERE team_name = NEW.winner;
    ->
    ->         -- Loser gets +1 loss
    ->         UPDATE points_table
    ->         SET losses = losses + 1
    ->         WHERE team_name = IF(NEW.winner = NEW.team1, NEW.team2, NEW.team1);
    ->     ELSE
    ->         -- No result, both teams get +1 no_result
    ->         UPDATE points_table
    ->         SET no_result = no_result + 1
    ->         WHERE team_name IN (NEW.team1, NEW.team2);
    ->     END IF;
    -> END$$
Query OK, 0 rows affected (0.02 sec)

mysql>
mysql> DELIMITER ;
```

After executing trigger we see that both teams number of matches played automatically increased.



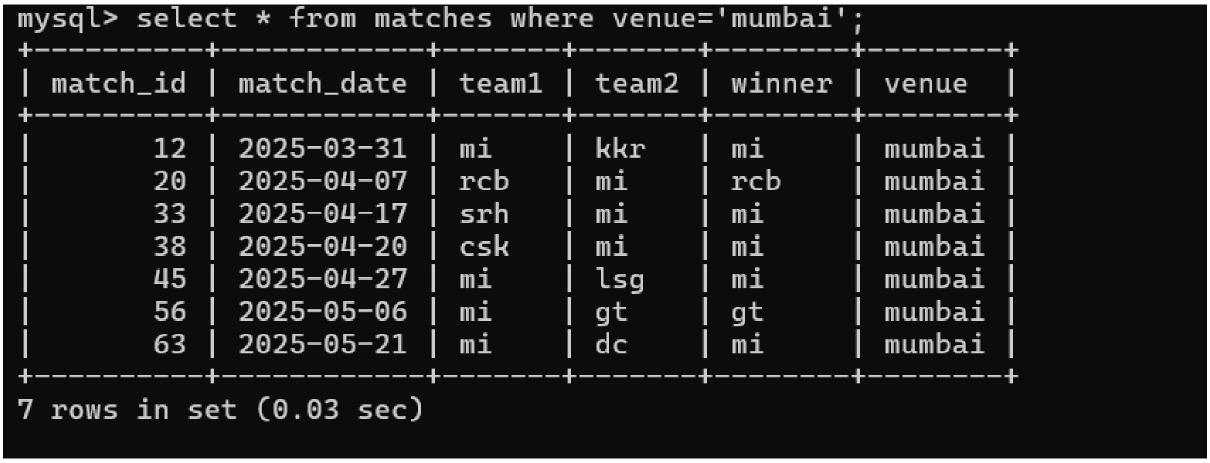
```
mysql> INSERT INTO matches (match_id, match_date, team1, team2, winner)
    -> VALUES (101, '2025-05-10', 'CSK', 'MI', 'CSK');
Query OK, 1 row affected (0.02 sec)

mysql> SELECT * FROM points_table WHERE team_name IN ('CSK', 'MI');
+-----------+----------------+------+--------+-----------+--------+
| team_name | matches_played | wins | losses | no_result | points |
+-----------+----------------+------+--------+-----------+--------+
| CSK       |             15 |    5 |     10 |         0 |      0 |
| MI        |             17 |    9 |      8 |         0 |      0 |
+-----------+----------------+------+--------+-----------+--------+
2 rows in set (0.00 sec)
```

SQL Queries

In this project, SQL queries are categorized into basic, intermediate, and advanced levels to demonstrate the progression of complexity and the depth of database operations performed.

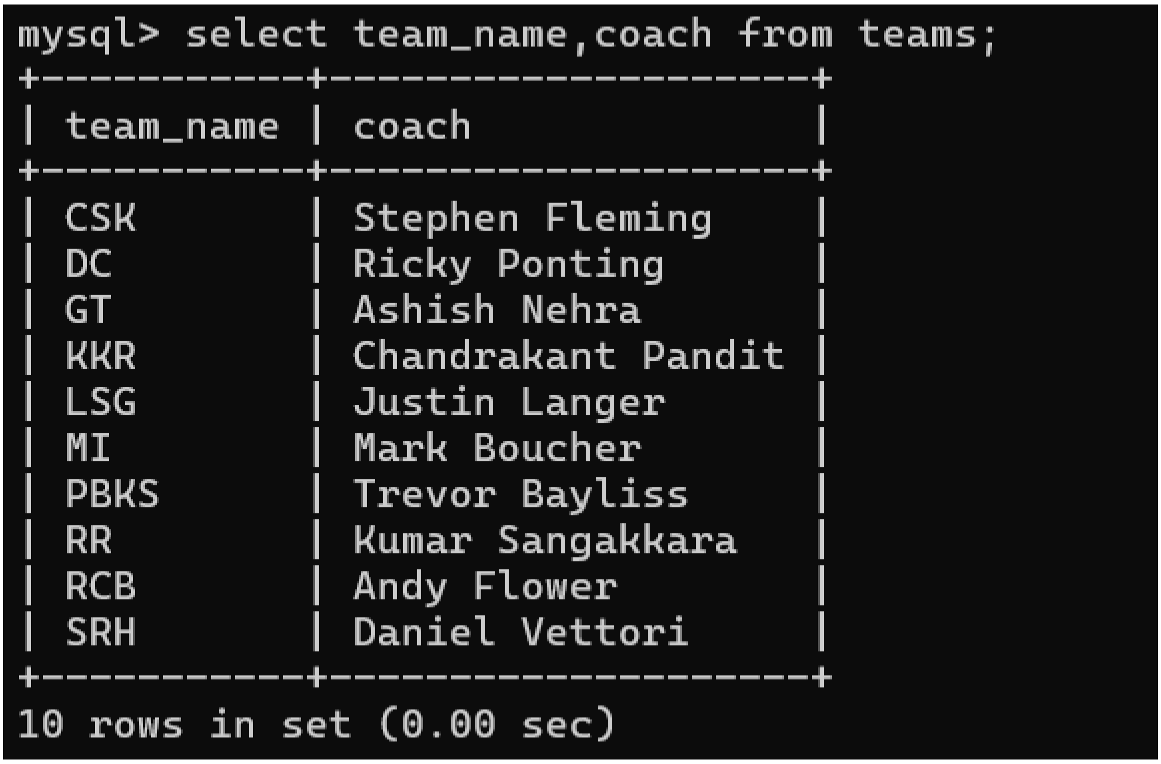
Basic Queries

Show all matches played at Mumbai.



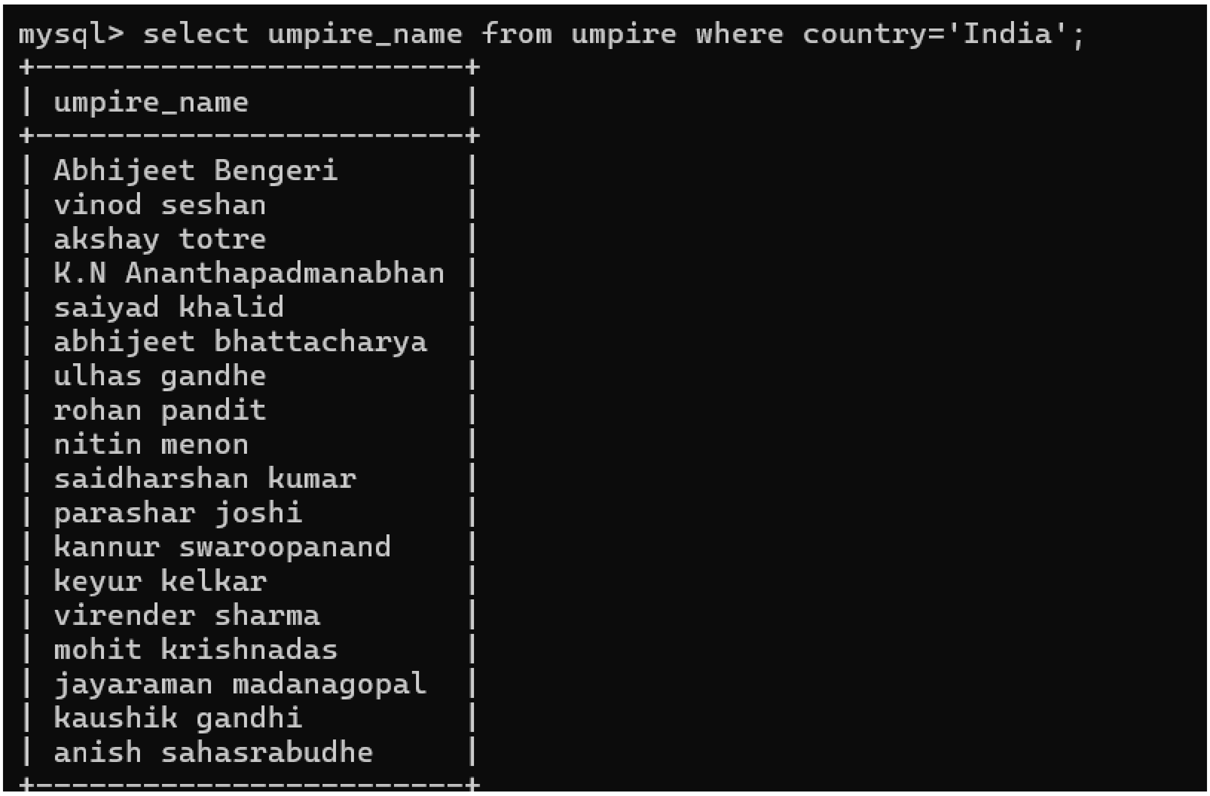
```
mysql> select * from matches where venue='mumbai';
+----------+------------+-------+-------+--------+--------+
| match_id | match_date | team1 | team2 | winner | venue  |
+----------+------------+-------+-------+--------+--------+
|       12 | 2025-03-31 | mi    | kkr   | mi     | mumbai |
|       20 | 2025-04-07 | rcb   | mi    | rcb    | mumbai |
|       33 | 2025-04-17 | srh   | mi    | mi     | mumbai |
|       38 | 2025-04-20 | csk   | mi    | mi     | mumbai |
|       45 | 2025-04-27 | mi    | lsg   | mi     | mumbai |
|       56 | 2025-05-06 | mi    | gt    | gt     | mumbai |
|       63 | 2025-05-21 | mi    | dc    | mi     | mumbai |
+----------+------------+-------+-------+--------+--------+
7 rows in set (0.03 sec)
```

Show team names and their coaches.



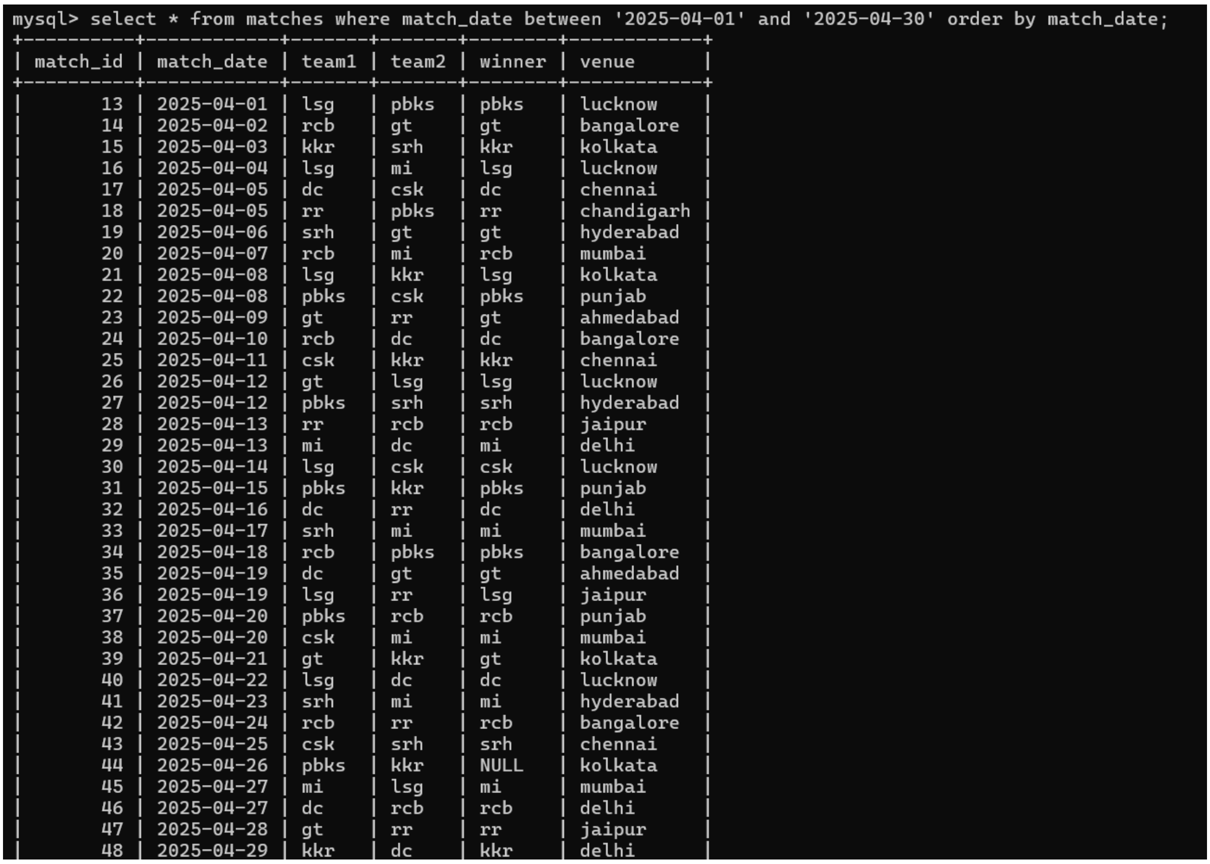
```
mysql> select team_name,coach from teams;
+-----------+---------------------+
| team_name | coach               |
+-----------+---------------------+
| CSK       | Stephen Fleming     |
| DC        | Ricky Ponting       |
| GT        | Ashish Nehra        |
| KKR       | Chandrakant Pandit  |
| LSG       | Justin Langer       |
| MI        | Mark Boucher        |
| PBKS      | Trevor Bayliss      |
| RR        | Kumar Sangakkara    |
| RCB       | Andy Flower         |
| SRH       | Daniel Vettori      |
+-----------+---------------------+
10 rows in set (0.00 sec)
```

Find all umpires from india.



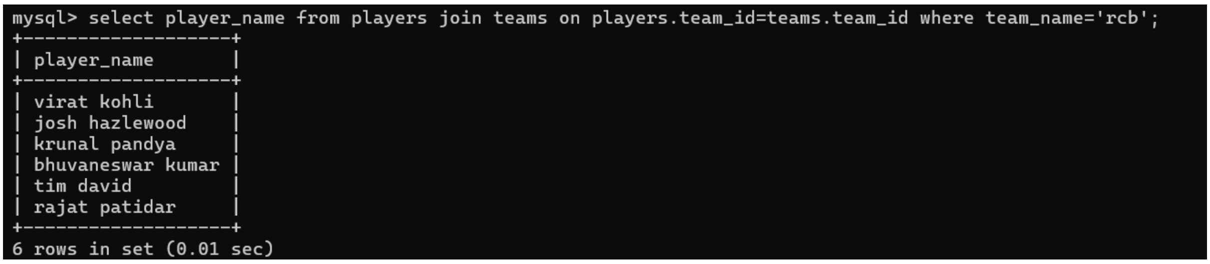
```
mysql> select umpire_name from umpire where country='India';
+-------------------------+
| umpire_name             |
+-------------------------+
| Abhijeet Bengeri        |
| vinod seshan            |
| akshay totre            |
| K.N Ananthapadmanabhan  |
| saiyad khalid           |
| abhijeet bhattacharya   |
| ulhas gandhe            |
| rohan pandit            |
| nitin menon             |
| saidharshan kumar       |
| parashar joshi          |
| kannur swaroopanand     |
| keyur kelkar            |
| virender sharma         |
| mohit krishnadas        |
| jayaraman madanagopal   |
| kaushik gandhi          |
| anish sahasrabudhe      |
+-------------------------+
```

Display match details in April 2025 sorted by date.



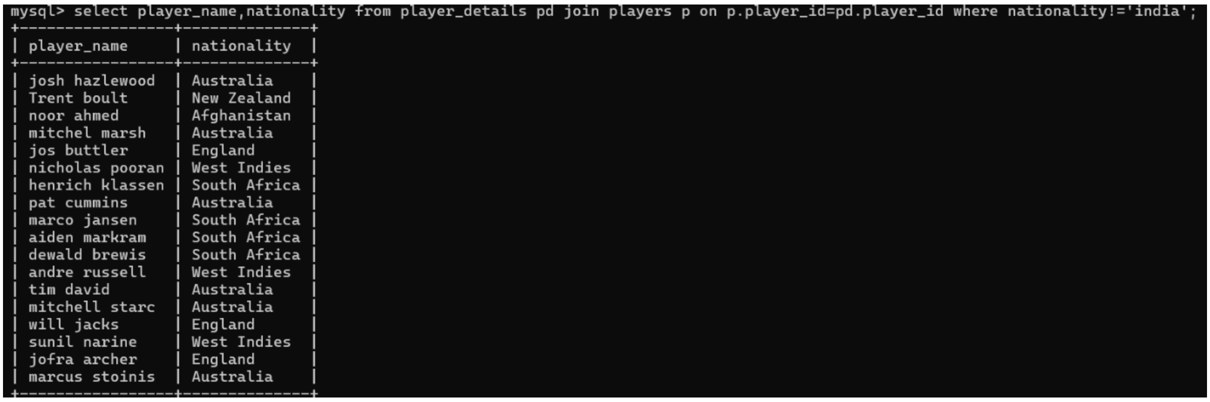
```
mysql> select * from matches where match_date between '2025-04-01' and '2025-04-30' order by match_date;
+----------+------------+-------+-------+--------+------------+
| match_id | match_date | team1 | team2 | winner | venue      |
+----------+------------+-------+-------+--------+------------+
|       13 | 2025-04-01 | lsg   | pbks  | pbks   | lucknow    |
|       14 | 2025-04-02 | rcb   | gt    | gt     | bangalore  |
|       15 | 2025-04-03 | kkr   | srh   | kkr    | kolkata    |
|       16 | 2025-04-04 | lsg   | mi    | lsg    | lucknow    |
|       17 | 2025-04-05 | dc    | csk   | dc     | chennai    |
|       18 | 2025-04-05 | rr    | pbks  | rr     | chandigarh |
|       19 | 2025-04-06 | srh   | gt    | gt     | hyderabad  |
|       20 | 2025-04-07 | rcb   | mi    | rcb    | mumbai     |
|       21 | 2025-04-08 | lsg   | kkr   | lsg    | kolkata    |
|       22 | 2025-04-08 | pbks  | csk   | pbks   | punjab     |
|       23 | 2025-04-09 | gt    | rr    | gt     | ahmedabad  |
|       24 | 2025-04-10 | rcb   | dc    | dc     | bangalore  |
|       25 | 2025-04-11 | csk   | kkr   | kkr    | chennai    |
|       26 | 2025-04-12 | gt    | lsg   | lsg    | lucknow    |
|       27 | 2025-04-12 | pbks  | srh   | srh    | hyderabad  |
|       28 | 2025-04-13 | rr    | rcb   | rcb    | jaipur     |
|       29 | 2025-04-13 | mi    | dc    | mi     | delhi      |
|       30 | 2025-04-14 | lsg   | csk   | csk    | lucknow    |
|       31 | 2025-04-15 | pbks  | kkr   | pbks   | punjab     |
|       32 | 2025-04-16 | dc    | rr    | dc     | delhi      |
|       33 | 2025-04-17 | srh   | mi    | mi     | mumbai     |
|       34 | 2025-04-18 | rcb   | pbks  | pbks   | bangalore  |
|       35 | 2025-04-19 | dc    | gt    | gt     | ahmedabad  |
|       36 | 2025-04-19 | lsg   | rr    | lsg    | jaipur     |
|       37 | 2025-04-20 | pbks  | rcb   | rcb    | punjab     |
|       38 | 2025-04-20 | csk   | mi    | mi     | mumbai     |
|       39 | 2025-04-21 | gt    | kkr   | gt     | kolkata    |
|       40 | 2025-04-22 | lsg   | dc    | dc     | lucknow    |
|       41 | 2025-04-23 | srh   | mi    | mi     | hyderabad  |
|       42 | 2025-04-24 | rcb   | rr    | rcb    | bangalore  |
|       43 | 2025-04-25 | csk   | srh   | srh    | chennai    |
|       44 | 2025-04-26 | pbks  | kkr   | NULL   | kolkata    |
|       45 | 2025-04-27 | mi    | lsg   | mi     | mumbai     |
|       46 | 2025-04-27 | dc    | rcb   | rcb    | delhi      |
|       47 | 2025-04-28 | gt    | rr    | rr     | jaipur     |
|       48 | 2025-04-29 | kkr   | dc    | kkr    | delhi      |
```

List all players of team “rcb”.



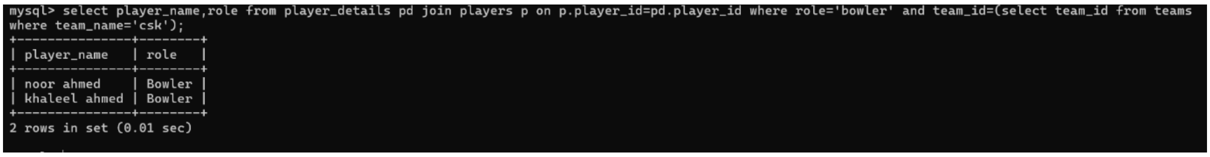
```
mysql> select player_name from players join teams on players.team_id=teams.team_id where team_name='rcb';
+--------------------+
| player_name        |
+--------------------+
| virat kohli        |
| josh hazlewood     |
| krunal pandya      |
| bhuvaneswar kumar  |
| tim david          |
| rajat patidar      |
+--------------------+
6 rows in set (0.01 sec)
```

List all foreign players.



```
mysql> select player_name,nationality from player_details pd join players p on p.player_id=pd.player_id where nationality!='india';
+-----------------+--------------+
| player_name     | nationality  |
+-----------------+--------------+
| josh hazlewood  | Australia    |
| Trent boult     | New Zealand  |
| noor ahmed      | Afghanistan  |
| mitchel marsh   | Australia    |
| jos buttler     | England      |
| nicholas pooran | West Indies  |
| henrich klassen | South Africa |
| pat cummins     | Australia    |
| marco jansen    | South Africa |
| aiden markram   | South Africa |
| dewald brewis   | South Africa |
| andre russell   | West Indies  |
| tim david       | Australia    |
| mitchell starc  | Australia    |
| will jacks      | England      |
| sunil narine    | West Indies  |
| jofra archer    | England      |
| marcus stoinis  | Australia    |
+-----------------+--------------+
```

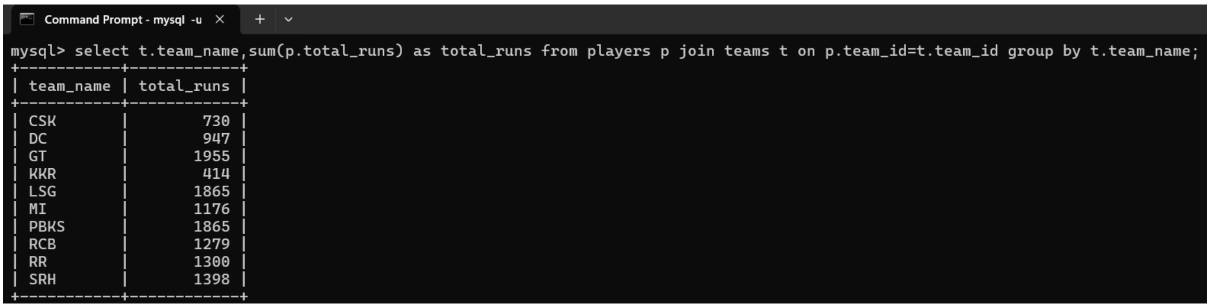
Get all bowlers from 'csk'.



```
mysql> select player_name,role from player_details pd join players p on p.player_id=pd.player_id where role='bowler' and team_id=(select team_id from teams
where team_name='csk');
+---------------+--------+
| player_name   | role   |
+---------------+--------+
| noor ahmed    | Bowler |
| khaleel ahmed | Bowler |
+---------------+--------+
2 rows in set (0.01 sec)
```

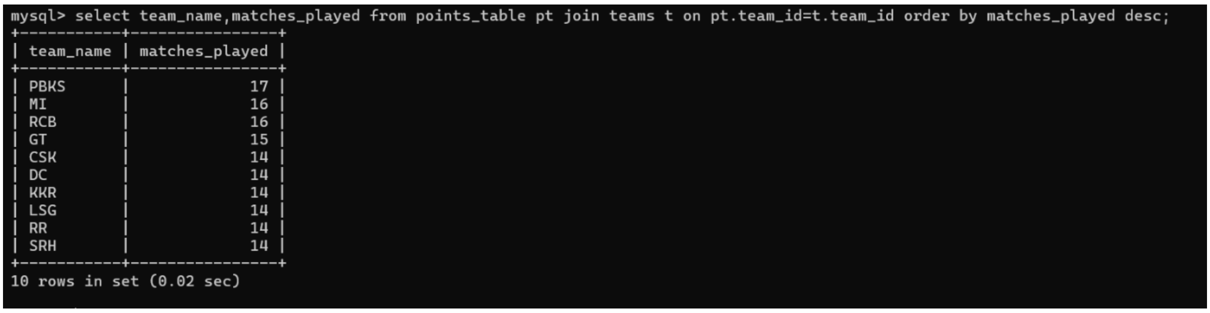
Intermediate Queries

Find the total runs scored by each team's players.



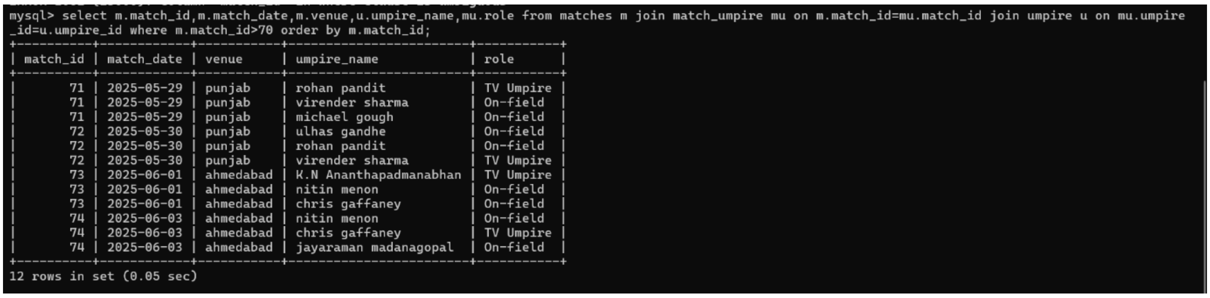
```
mysql> select t.team_name,sum(p.total_runs) as total_runs from players p join teams t on p.team_id=t.team_id group by t.team_name;
+-----------+------------+
| team_name | total_runs |
+-----------+------------+
| CSK       |        730 |
| DC        |        947 |
| GT        |       1955 |
| KKR       |        414 |
| LSG       |       1865 |
| MI        |       1176 |
| PBKS      |       1865 |
| RCB       |       1279 |
| RR        |       1300 |
| SRH       |       1398 |
+-----------+------------+
```

Get the number of matches each team has played.



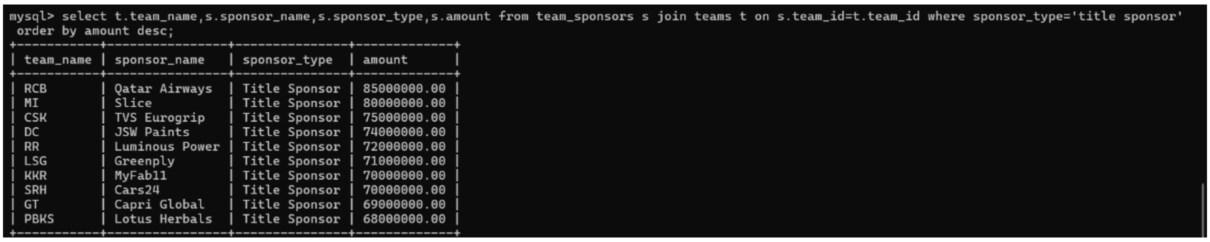
```
mysql> select team_name,matches_played from points_table pt join teams t on pt.team_id=t.team_id order by matches_played desc;
+-----------+----------------+
| team_name | matches_played |
+-----------+----------------+
| PBKS      |             17 |
| MI        |             16 |
| RCB       |             16 |
| GT        |             15 |
| CSK       |             14 |
| DC        |             14 |
| KKR       |             14 |
| LSG       |             14 |
| RR        |             14 |
| SRH       |             14 |
+-----------+----------------+
10 rows in set (0.02 sec)
```

List all matches along with the names of umpires who officiated for playoffs.



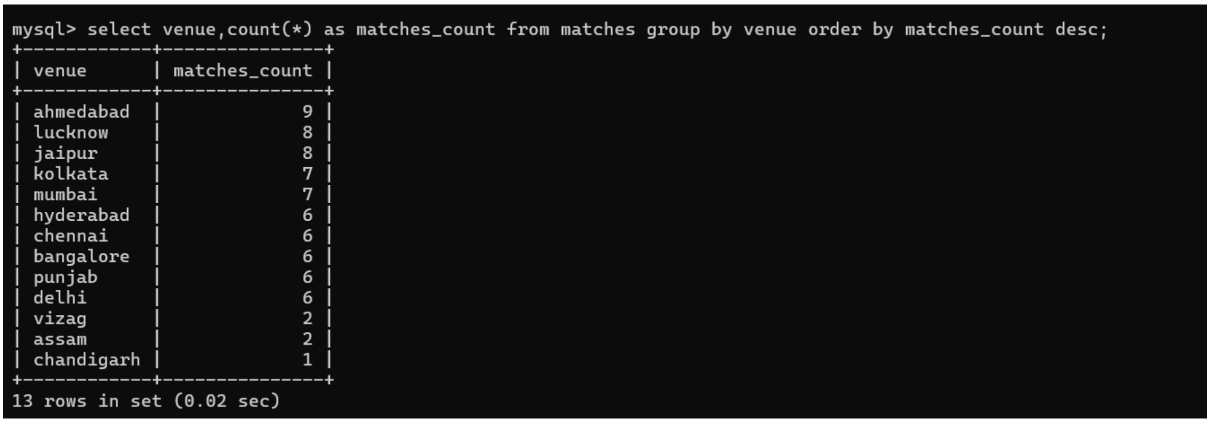
```
mysql> select m.match_id,m.match_date,m.venue,u.umpire_name,mu.role from matches m join match_umpire mu on m.match_id=mu.match_id join umpire u on mu.umpire
_id=u.umpire_id where m.match_id>70 order by m.match_id;
+----------+------------+-----------+-----------------------+------------+
| match_id | match_date | venue     | umpire_name           | role       |
+----------+------------+-----------+-----------------------+------------+
|       71 | 2025-05-29 | punjab    | rohan pandit          | TV Umpire  |
|       71 | 2025-05-29 | punjab    | virender sharma       | On-field   |
|       71 | 2025-05-29 | punjab    | michael gough         | On-field   |
|       72 | 2025-05-30 | punjab    | ulhas gandhe          | On-field   |
|       72 | 2025-05-30 | punjab    | rohan pandit          | On-field   |
|       72 | 2025-05-30 | punjab    | virender sharma       | TV Umpire  |
|       73 | 2025-06-01 | ahmedabad | K.N Ananthapadmanabhan | TV Umpire |
|       73 | 2025-06-01 | ahmedabad | nitin menon           | On-field   |
|       73 | 2025-06-01 | ahmedabad | chris gaffaney        | On-field   |
|       74 | 2025-06-03 | ahmedabad | nitin menon           | On-field   |
|       74 | 2025-06-03 | ahmedabad | chris gaffaney        | TV Umpire  |
|       74 | 2025-06-03 | ahmedabad | jayaraman madanagopal | On-field   |
+----------+------------+-----------+-----------------------+------------+
12 rows in set (0.05 sec)
```

Find the title sponsor details for each team.



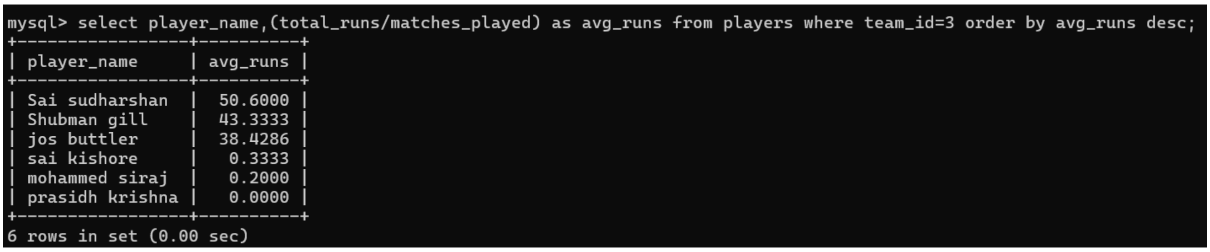
```
mysql> select t.team_name,s.sponsor_name,s.sponsor_type,s.amount from team_sponsors s join teams t on s.team_id=t.team_id where sponsor_type='title sponsor'
 order by amount desc;
+-----------+----------------+---------------+-------------+
| team_name | sponsor_name   | sponsor_type  | amount      |
+-----------+----------------+---------------+-------------+
| RCB       | Qatar Airways  | Title Sponsor | 85000000.00 |
| MI        | Slice          | Title Sponsor | 80000000.00 |
| CSK       | TVS Eurogrip   | Title Sponsor | 75000000.00 |
| DC        | JSW Paints     | Title Sponsor | 74000000.00 |
| RR        | Luminous Power | Title Sponsor | 72000000.00 |
| LSG       | Greenply       | Title Sponsor | 71000000.00 |
| KKR       | MyFab11        | Title Sponsor | 70000000.00 |
| SRH       | Cars24         | Title Sponsor | 70000000.00 |
| GT        | Capri Global   | Title Sponsor | 69000000.00 |
| PBKS      | Lotus Herbals  | Title Sponsor | 68000000.00 |
+-----------+----------------+---------------+-------------+
```

Count matches at each venue.



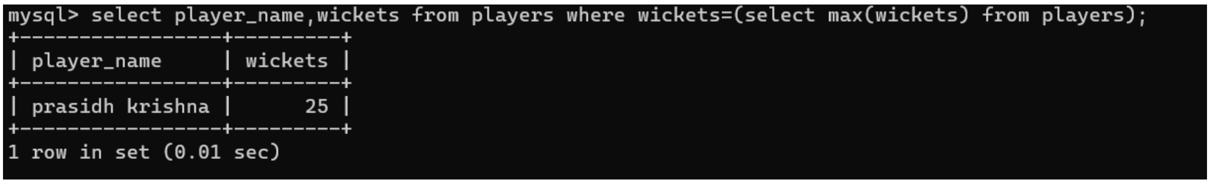
```
mysql> select venue,count(*) as matches_count from matches group by venue order by matches_count desc;
+------------+---------------+
| venue      | matches_count |
+------------+---------------+
| ahmedabad  |             9 |
| lucknow    |             8 |
| jaipur     |             8 |
| kolkata    |             7 |
| mumbai     |             7 |
| hyderabad  |             6 |
| chennai    |             6 |
| bangalore  |             6 |
| punjab     |             6 |
| delhi      |             6 |
| vizag      |             2 |
| assam      |             2 |
| chandigarh |             1 |
+------------+---------------+
13 rows in set (0.02 sec)
```

Show average runs scored by players of ‘gt’.



```
mysql> select player_name,(total_runs/matches_played) as avg_runs from players where team_id=3 order by avg_runs desc;
+-----------------+----------+
| player_name     | avg_runs |
+-----------------+----------+
| Sai sudharshan  |  50.6000 |
| Shubman gill    |  43.3333 |
| jos buttler     |  38.4286 |
| sai kishore     |   0.3333 |
| mohammed siraj  |   0.2000 |
| prasidh krishna |   0.0000 |
+-----------------+----------+
6 rows in set (0.00 sec)
```

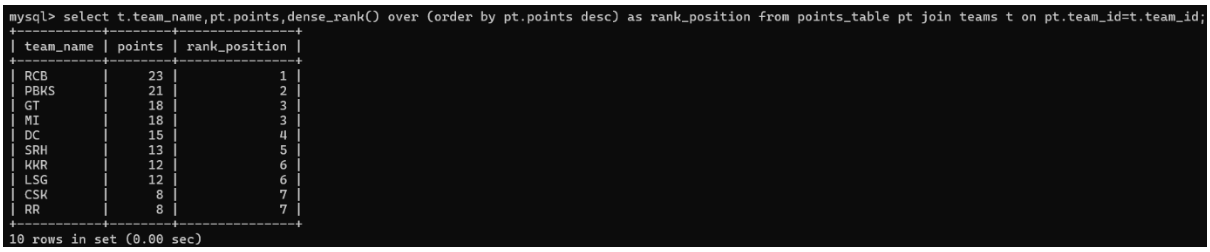
Highest Wicket-Taker(s) in ipl 2025.



```
mysql> select player_name,wickets from players where wickets=(select max(wickets) from players);
+-----------------+---------+
| player_name     | wickets |
+-----------------+---------+
| prasidh krishna |      25 |
+-----------------+---------+
1 row in set (0.01 sec)
```

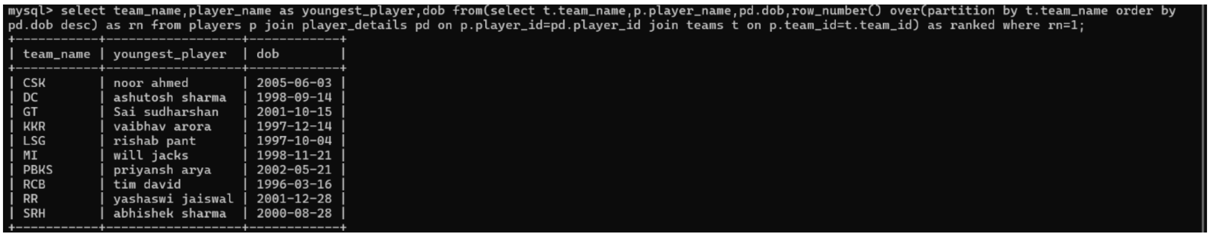
Advanced Queries

Rank teams by total points (ties get the same rank).



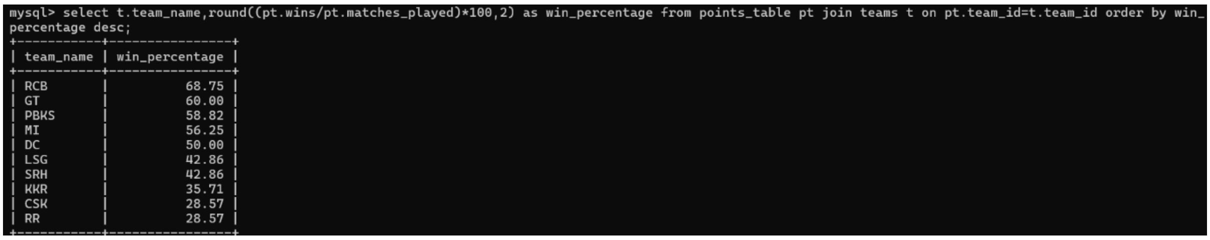
```
mysql> select t.team_name,pt.points,dense_rank() over (order by pt.points desc) as rank_position from points_table pt join teams t on pt.team_id=t.team_id;
+-----------+--------+---------------+
| team_name | points | rank_position |
+-----------+--------+---------------+
| RCB       |     23 |             1 |
| PBKS      |     21 |             2 |
| GT        |     18 |             3 |
| MI        |     18 |             3 |
| DC        |     15 |             4 |
| SRH       |     13 |             5 |
| KKR       |     12 |             6 |
| LSG       |     12 |             6 |
| CSK       |      8 |             7 |
| RR        |      8 |             7 |
+-----------+--------+---------------+
10 rows in set (0.00 sec)
```

Find the youngest player in each team.



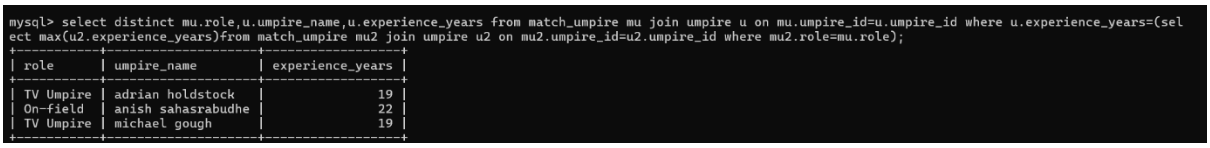
```
mysql> select team_name,player_name as youngest_player,dob from(select t.team_name,p.player_name,pd.dob,row_number() over(partition by t.team_name order by
pd.dob desc) as rn from players p join player_details pd on p.player_id=pd.player_id join teams t on p.team_id=t.team_id) as ranked where rn=1;
+-----------+------------------+------------+
| team_name | youngest_player  | dob        |
+-----------+------------------+------------+
| CSK       | noor ahmed       | 2005-06-03 |
| DC        | ashutosh sharma  | 1998-09-14 |
| GT        | Sai sudharshan   | 2001-10-15 |
| KKR       | vaibhav arora    | 1997-12-14 |
| LSG       | rishab pant      | 1997-10-04 |
| MI        | will jacks       | 1998-11-21 |
| PBKS      | priyansh arya    | 2002-05-21 |
| RCB       | tim david        | 1996-03-16 |
| RR        | yashaswi jaiswal | 2001-12-28 |
| SRH       | abhishek sharma  | 2000-08-28 |
+-----------+------------------+------------+
```

Calculate the win percentage for each team.



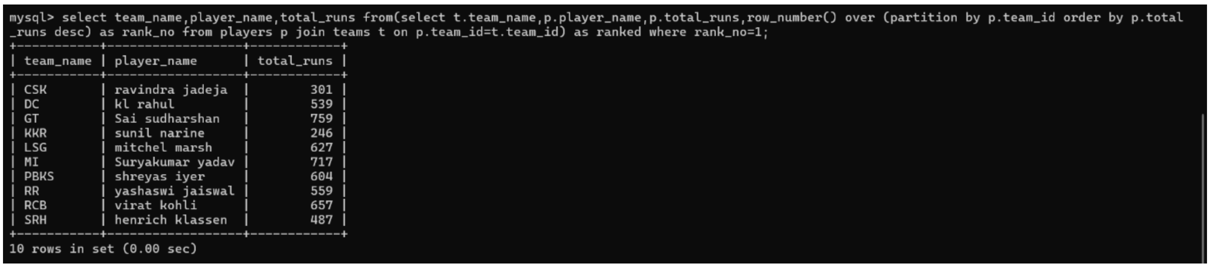
```
mysql> select t.team_name,round((pt.wins/pt.matches_played)*100,2) as win_percentage from points_table pt join teams t on pt.team_id=t.team_id order by win_
percentage desc;
+-----------+----------------+
| team_name | win_percentage |
+-----------+----------------+
| RCB       |          68.75 |
| GT        |          60.00 |
| PBKS      |          58.82 |
| MI        |          56.25 |
| DC        |          50.00 |
| LSG       |          42.86 |
| SRH       |          42.86 |
| KKR       |          35.71 |
| CSK       |          28.57 |
| RR        |          28.57 |
+-----------+----------------+
```

Most Experienced Umpire in Each Role.



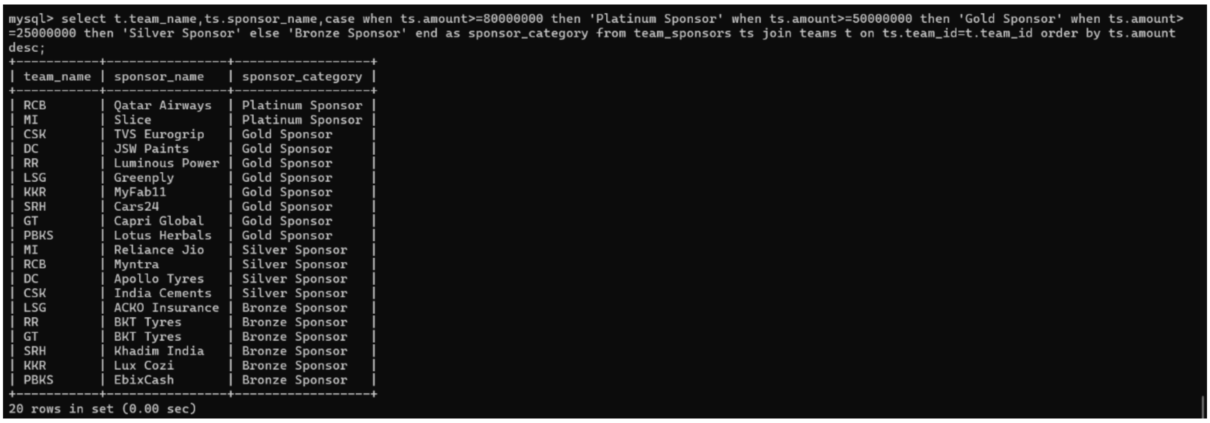
```
mysql> select distinct mu.role,u.umpire_name,u.experience_years from match_umpire mu join umpire u on mu.umpire_id=u.umpire_id where u.experience_years=(sel
ect max(u2.experience_years)from match_umpire mu2 join umpire u2 on mu2.umpire_id=u2.umpire_id where mu2.role=mu.role);
+-----------+--------------------+------------------+
| role      | umpire_name        | experience_years |
+-----------+--------------------+------------------+
| TV Umpire | adrian holdstock   |               19 |
| On-field  | anish sahasrabudhe |               22 |
| TV Umpire | michael gough      |               19 |
+-----------+--------------------+------------------+
```

Get the top scorer from each team.



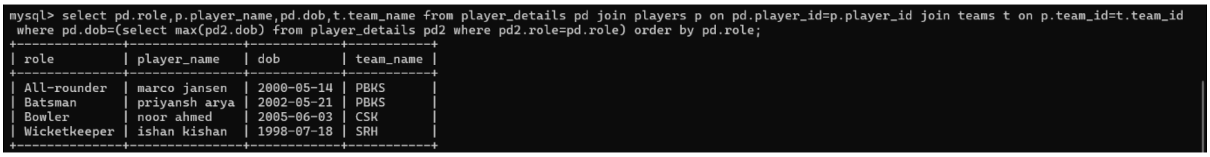
```
mysql> select team_name,player_name,total_runs from(select t.team_name,p.player_name,p.total_runs,row_number() over (partition by p.team_id order by p.total
_runs desc) as rank_no from players p join teams t on p.team_id=t.team_id) as ranked where rank_no=1;
+-----------+------------------+------------+
| team_name | player_name      | total_runs |
+-----------+------------------+------------+
| CSK       | ravindra jadeja  |        301 |
| DC        | kl rahul         |        539 |
| GT        | Sai sudharshan   |        759 |
| KKR       | sunil narine     |        246 |
| LSG       | mitchel marsh    |        627 |
| MI        | Suryakumar yadav |        717 |
| PBKS      | shreyas iyer     |        604 |
| RR        | yashaswi jaiswal |        559 |
| RCB       | virat kohli      |        657 |
| SRH       | henrich klassen  |        487 |
+-----------+------------------+------------+
10 rows in set (0.00 sec)
```

Find the sponsor category by amount.



```
mysql> select t.team_name,ts.sponsor_name,case when ts.amount>=80000000 then 'Platinum Sponsor' when ts.amount>=50000000 then 'Gold Sponsor' when ts.amount>
=25000000 then 'Silver Sponsor' else 'Bronze Sponsor' end as sponsor_category from team_sponsors ts join teams t on ts.team_id=t.team_id order by ts.amount
desc;
+-----------+-----------------+------------------+
| team_name | sponsor_name    | sponsor_category |
+-----------+-----------------+------------------+
| RCB       | Qatar Airways   | Platinum Sponsor |
| MI        | Slice           | Platinum Sponsor |
| CSK       | TVS Eurogrip    | Gold Sponsor     |
| DC        | JSW Paints      | Gold Sponsor     |
| RR        | Luminous Power  | Gold Sponsor     |
| LSG       | Greenply        | Gold Sponsor     |
| KKR       | MyFab11         | Gold Sponsor     |
| SRH       | Cars24          | Gold Sponsor     |
| GT        | Capri Global    | Gold Sponsor     |
| PBKS      | Lotus Herbals   | Gold Sponsor     |
| MI        | Reliance Jio    | Silver Sponsor   |
| RCB       | Myntra          | Silver Sponsor   |
| DC        | Apollo Tyres    | Silver Sponsor   |
| CSK       | India Cements   | Silver Sponsor   |
| LSG       | ACKO Insurance  | Bronze Sponsor   |
| RR        | BKT Tyres       | Bronze Sponsor   |
| GT        | BKT Tyres       | Bronze Sponsor   |
| SRH       | Khadim India    | Bronze Sponsor   |
| KKR       | Lux Cozi        | Bronze Sponsor   |
| PBKS      | EbixCash        | Bronze Sponsor   |
+-----------+-----------------+------------------+
20 rows in set (0.00 sec)
```

Find the youngest player in each role.



```
mysql> select pd.role,p.player_name,pd.dob,t.team_name from player_details pd join players p on pd.player_id=p.player_id join teams t on p.team_id=t.team_id
 where pd.dob=(select max(pd2.dob) from player_details pd2 where pd2.role=pd.role) order by pd.role;
+--------------+----------------+------------+-----------+
| role         | player_name    | dob        | team_name |
+--------------+----------------+------------+-----------+
| All-rounder  | marco jansen   | 2000-05-14 | PBKS      |
| Batsman      | priyansh arya  | 2002-05-21 | PBKS      |
| Bowler       | noor ahmed     | 2005-06-03 | CSK       |
| Wicketkeeper | ishan kishan   | 1998-07-18 | SRH       |
+--------------+----------------+------------+-----------+
```

Challenges and Solutions

1. Handling Complex Joins

Challenge:

Combining data from multiple tables like players, teams, matches, and stats required writing complex JOIN queries, which were initially difficult to structure correctly.

Solution:

Carefully planned table relationships and used INNER JOIN, LEFT JOIN, and simulated FULL OUTER JOINs. Tested each join step-by-step to ensure accurate data retrieval.

2. Maintaining Data Integrity with Triggers

Challenge:

Automatically updating the points table when new match results were inserted needed precise triggers without causing errors or infinite loops.

Solution:

Implemented AFTER INSERT triggers with clear logic and tested extensively with sample data. Used variables to handle team IDs and carefully managed NULL cases (e.g., tied matches).

3. User Permissions and Security

Challenge:

Ensuring that only authorized users could modify sensitive data like match results and player stats.

Solution:

Used DCL commands to grant specific privileges to different users. Created roles with limited permissions for read-only access or data entry tasks.

4. Simulating FULL OUTER JOIN in MySQL

Challenge:

MySQL does not support FULL OUTER JOIN natively, which was needed for some reports.

Solution:

Simulated FULL OUTER JOIN using a UNION of LEFT JOIN and RIGHT JOIN queries to get comprehensive data sets.

5. Managing Transaction Consistency

Challenge:

Ensuring that related updates (like inserting matches and updating points) were consistent and rollbacked properly on errors.

Solution:

Used TCL commands such as START TRANSACTION, COMMIT, and ROLLBACK to manage transactions, ensuring data integrity in multi-step operations.

Conclusion

Working on the IPL 2025 SQL project has been a valuable learning experience. It allowed me to apply database design principles by creating efficient table structures and establishing clear relationships between entities like teams, players, matches, and points.

Through writing complex SQL queries involving multiple joins, views, and stored procedures, I gained practical skills in retrieving and managing large sets of interconnected data. Implementing triggers to automate updates deepened my understanding of maintaining data integrity and workflow automation in databases.

Additionally, addressing challenges such as simulating full outer joins and managing user permissions helped me develop problem-solving abilities crucial for real-world database management.

This project has not only enhanced my technical expertise in SQL but also demonstrated the importance of careful planning, testing, and security considerations in building robust database applications. I am confident these skills will support my growth in database development and data analysis in future projects.