Contents

Introduction

The Indian Premier League (IPL) has emerged as a global phenomenon, captivating audiences with its electrifying blend of athletic excellence, strategic acumen, and vibrant entertainment. Since its inception in 2008, the IPL has transcended the boundaries of sport, fostering a dynamic ecosystem encompassing diverse stakeholders and generating vast data.

Witnessing its exponential growth over the past decade, we are intrigued by the vast amount of data generated by the league and its potential to unlock more profound insights into its various facets. This project delves into this exciting domain by developing a comprehensive Database Management System (DBMS) dedicated to the IPL.

Our project aims to curate a robust DBMS meticulously. This system will act as a central repository for a diverse range of data points, encompassing player statistics, intricate match details, and team performance metrics; this project extends beyond mere data collection, aiming to empower a diverse group of stakeholders within the IPL ecosystem. Team management stands to gain significant value from the project. They can leverage the system to make informed decisions regarding player selection, meticulously analyze opponent strategies, and optimize team compositions based on comprehensive statistical insights from past seasons and performances.

Figure I.1: C. Gayle hits 175 in 66 Balls RCB *vs* PW 2013.**Ref:** Bleacher Report

The IPL DBMS serves as more than just a data repository for players. It transforms into a valuable tool for self-assessment and career advancement. The system efficiently stores and tracks their performance statistics enabling players to analyze their strengths and weaknesses, track their progress over time, and negotiate better deals with franchises based on their data-driven value.

Administrators and broadcasters benefit immensely from the real-time capabilities of the IPL DBMS. The system empowers them to generate instant reports, identify future trends, and make informed decisions regarding scheduling, marketing strategies, and revenue generation opportunities. By gleaning insights from the database, they can craft a more engaging viewing experience for fans and maximize the league's commercial potential. Finally, the IPL DBMS empowers the lifeblood of the league – its passionate fans. The system fosters a personalized experience and interactive features, allowing them to stay deeply connected to the action.

By constructing this system, we aim to contribute towards a more comprehensive understanding of the league's structure, performance dynamics, and economic impact. This project is a testament to our dedication to leveraging the power of data management to unlock hidden insights within the IPL ecosystem, ultimately contributing to this remarkable sporting phenomenon's continued success and evolution.

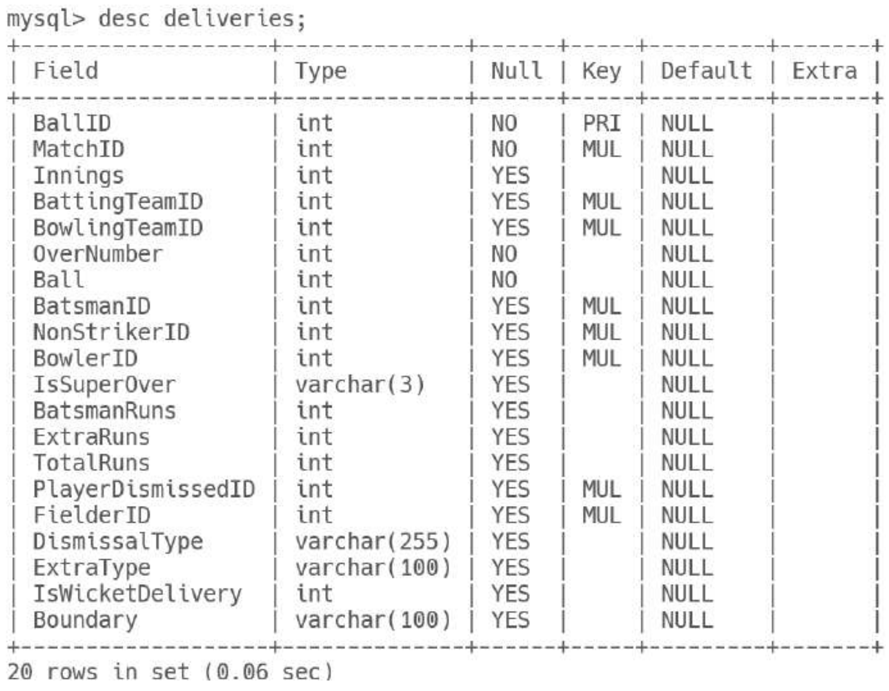
Master & Transaction Tables

The database contains four master tables. All tables are created to cater to the three transaction tables of the database. The comperehensive list of tables is as follows:

deliveries This table stores detailed information about each delivery in the matches, including ball ID, match ID, innings, batting and bowling team IDs, batsman ID, bowler ID, runs scored (including extras), dismissal details, etc.

It has foreign key constraints to link other tables such as matches, teamsmaster, and playermaster to maintain referential integrity. The DDL command mentioned below was used to create the table in the database.

```sql
-- ipldatabase.deliveries definition
CREATE TABLE 'deliveries' (
 'BallID' int NOT NULL,
 'MatchID' int NOT NULL,
 'Innings' int DEFAULT NULL,
 'BattingTeamID' int DEFAULT NULL,
 'BowlingTeamID' int DEFAULT NULL,
 'OverNumber' int NOT NULL,
 'Ball' int NOT NULL,
 'BatsmanID' int DEFAULT NULL,
 'NonStrikerID' int DEFAULT NULL,
 'BowlerID' int DEFAULT NULL,
 'IsSuperOver' varchar(3) DEFAULT NULL,
 'BatsmanRuns' int DEFAULT NULL,
 'ExtraRuns' int DEFAULT NULL,
 'TotalRuns' int DEFAULT NULL,
 'PlayerDismissedID' int DEFAULT NULL,
 'FielderID' int DEFAULT NULL,
 'DismissalType' varchar(255) DEFAULT NULL,
 'ExtraType' varchar(100) DEFAULT NULL,
 'IsWicketDelivery' int DEFAULT NULL,
 'Boundary' varchar(100) DEFAULT NULL,
 PRIMARY KEY ('BallID'),
 KEY 'deliveries_matches_fk' ('MatchID'),
 KEY 'deliveries_playermaster_fk' ('BatsmanID'),
 KEY 'deliveries_playermaster_fk_1' ('NonStrikerID'),
 KEY 'deliveries_playermaster_fk_2' ('BowlerID'),
 KEY 'deliveries_playermaster_fk_3' ('PlayerDismissedID'),
 KEY 'deliveries_teamsmaster_FK' ('BattingTeamID'),
 KEY 'deliveries_teamsmaster_FK_1' ('BowlingTeamID'),
 KEY 'deliveries_playermaster_fk_4' ('FielderID'),
 CONSTRAINT 'deliveries_matches_fk' FOREIGN KEY ('MatchID') REFERENCES 'matches'
    ('MatchID'),
 CONSTRAINT 'deliveries_playermaster_FK' FOREIGN KEY ('BatsmanID') REFERENCES
    'playermaster' ('PlayerID'),
 CONSTRAINT 'deliveries_playermaster_FK_1' FOREIGN KEY ('BowlerID') REFERENCES
    'playermaster' ('PlayerID'),
 CONSTRAINT 'deliveries_playermaster_FK_2' FOREIGN KEY ('NonStrikerID') REFERENCES
    'playermaster' ('PlayerID'),
 CONSTRAINT 'deliveries_playermaster_FK_3' FOREIGN KEY ('PlayerDismissedID')
    REFERENCES 'playermaster' ('PlayerID'),
 CONSTRAINT 'deliveries_playermaster_FK_4' FOREIGN KEY ('FielderID') REFERENCES
    'playermaster' ('PlayerID'),
 CONSTRAINT 'deliveries_teamsmaster_FK' FOREIGN KEY ('BattingTeamID') REFERENCES
    'teamsmaster' ('TeamID'),
 CONSTRAINT 'deliveries_teamsmaster_FK_1' FOREIGN KEY ('BowlingTeamID') REFERENCES
    'teamsmaster' ('TeamID')
) ENGINE = InnoDB DEFAULT CHARSET = latin1;
```

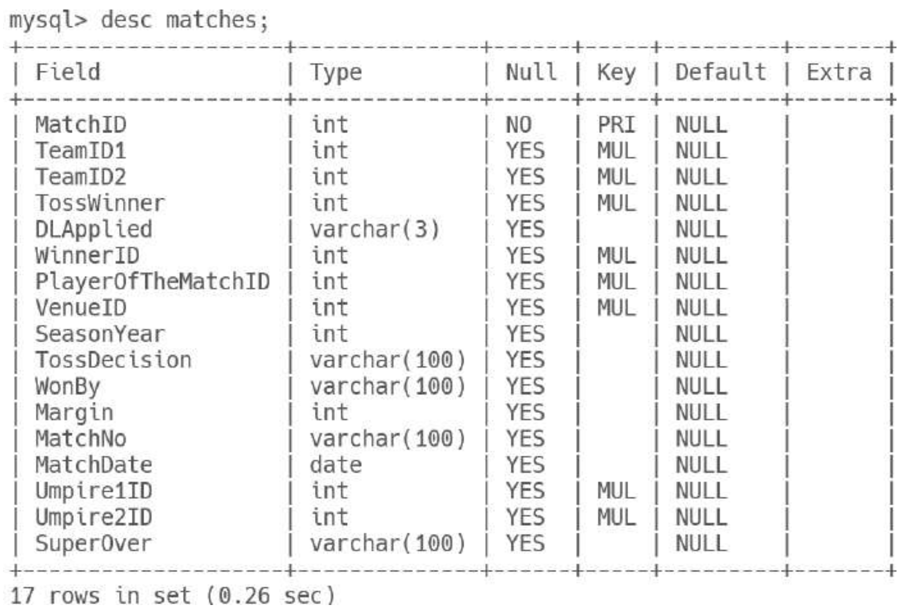


```
mysql> desc deliveries;
+-------------------+--------------+------+-----+---------+-------+
| Field             | Type         | Null | Key | Default | Extra |
+-------------------+--------------+------+-----+---------+-------+
| BallID            | int          | NO   | PRI | NULL    |       |
| MatchID           | int          | NO   | MUL | NULL    |       |
| Innings           | int          | YES  |     | NULL    |       |
| BattingTeamID     | int          | YES  | MUL | NULL    |       |
| BowlingTeamID     | int          | YES  | MUL | NULL    |       |
| OverNumber        | int          | NO   |     | NULL    |       |
| Ball              | int          | NO   |     | NULL    |       |
| BatsmanID         | int          | YES  | MUL | NULL    |       |
| NonStrikerID      | int          | YES  | MUL | NULL    |       |
| BowlerID          | int          | YES  | MUL | NULL    |       |
| IsSuperOver       | varchar(3)   | YES  |     | NULL    |       |
| BatsmanRuns       | int          | YES  |     | NULL    |       |
| ExtraRuns         | int          | YES  |     | NULL    |       |
| TotalRuns         | int          | YES  |     | NULL    |       |
| PlayerDismissedID | int          | YES  | MUL | NULL    |       |
| FielderID         | int          | YES  | MUL | NULL    |       |
| DismissalType     | varchar(255) | YES  |     | NULL    |       |
| ExtraType         | varchar(100) | YES  |     | NULL    |       |
| IsWicketDelivery  | int          | YES  |     | NULL    |       |
| Boundary          | varchar(100) | YES  |     | NULL    |       |
+-------------------+--------------+------+-----+---------+-------+
20 rows in set (0.06 sec)
```

Matches This table contains information about each match, including match ID, participating teams, toss winner, DL (Duckworth-Lewis) application, winner, win by runs/wickets, player of the match, venue ID, season year, toss decision, result type, etc.

It has foreign key constraints to link to other tables such as teamsmaster, playermaster, and venuesmaster. The DDL command mentioned below was used to create the table in the database.

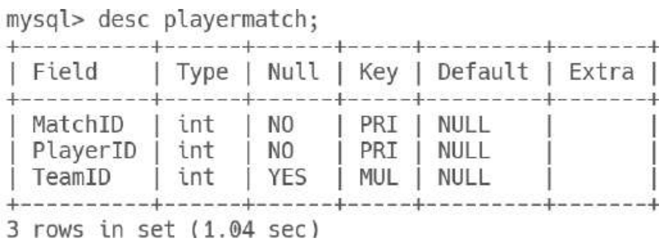
```
-- ipldatabase.matches definition
CREATE TABLE 'matches' (
 'MatchID' int NOT NULL,
 'TeamID1' int DEFAULT NULL,
 'TeamID2' int DEFAULT NULL,
 'TossWinner' int DEFAULT NULL,
 'DLApplied' varchar(3) DEFAULT NULL,
 'WinnerID' int DEFAULT NULL,
 'PlayerOfTheMatchID' int DEFAULT NULL,
 'VenueID' int DEFAULT NULL,
 'SeasonYear' int DEFAULT NULL,
 'TossDecision' varchar(100) DEFAULT NULL,
 'WonBy' varchar(100) DEFAULT NULL,
 'Margin' int DEFAULT NULL,
 'MatchNo' varchar(100) DEFAULT NULL,
 'MatchDate' date DEFAULT NULL,
 'Umpire1ID' int DEFAULT NULL,
 'Umpire2ID' int DEFAULT NULL,
 'SuperOver' varchar(100) DEFAULT NULL,
 PRIMARY KEY ('MatchID'),
 KEY 'matches_teamsmaster_FK' ('TeamID1'),
 KEY 'matches_teamsmaster_FK_1' ('TeamID2'),
 KEY 'matches_teamsmaster_FK_2' ('TossWinner'),
 KEY 'matches_teamsmaster_FK_3' ('WinnerID'),
 KEY 'matches_playermaster_FK' ('PlayerOfTheMatchID'),
 KEY 'matches_umpiresmaster_FK' ('Umpire1ID'),
 KEY 'matches_umpiresmaster_FK_1' ('Umpire2ID'),
 KEY 'matches_venuesmaster_FK' ('VenueID'),
 CONSTRAINT 'matches_playermaster_FK' FOREIGN KEY ('PlayerOfTheMatchID') REFERENCES
    'playermaster' ('PlayerID'),
 CONSTRAINT 'matches_teamsmaster_FK' FOREIGN KEY ('TeamID1') REFERENCES
    'teamsmaster' ('TeamID'),
 CONSTRAINT 'matches_teamsmaster_FK_1' FOREIGN KEY ('TeamID2') REFERENCES
    'teamsmaster' ('TeamID'),
 CONSTRAINT 'matches_teamsmaster_FK_2' FOREIGN KEY ('TossWinner') REFERENCES
    'teamsmaster' ('TeamID'),
 CONSTRAINT 'matches_teamsmaster_FK_3' FOREIGN KEY ('WinnerID') REFERENCES
    'teamsmaster' ('TeamID'),
 CONSTRAINT 'matches_umpiresmaster_FK' FOREIGN KEY ('Umpire1ID') REFERENCES
    'umpiresmaster' ('UmpireID'),
 CONSTRAINT 'matches_umpiresmaster_FK_1' FOREIGN KEY ('Umpire2ID') REFERENCES
    'umpiresmaster' ('UmpireID'),
 CONSTRAINT 'matches_venuesmaster_FK' FOREIGN KEY ('VenueID') REFERENCES
    'venuesmaster' ('VenueID')
) ENGINE = InnoDB DEFAULT CHARSET = latin1;
```



```
mysql> desc matches;
+--------------------+--------------+------+-----+---------+-------+
| Field              | Type         | Null | Key | Default | Extra |
+--------------------+--------------+------+-----+---------+-------+
| MatchID            | int          | NO   | PRI | NULL    |       |
| TeamID1            | int          | YES  | MUL | NULL    |       |
| TeamID2            | int          | YES  | MUL | NULL    |       |
| TossWinner         | int          | YES  | MUL | NULL    |       |
| DLApplied          | varchar(3)   | YES  |     | NULL    |       |
| WinnerID           | int          | YES  | MUL | NULL    |       |
| PlayerOfTheMatchID | int          | YES  | MUL | NULL    |       |
| VenueID            | int          | YES  | MUL | NULL    |       |
| SeasonYear         | int          | YES  |     | NULL    |       |
| TossDecision       | varchar(100) | YES  |     | NULL    |       |
| WonBy              | varchar(100) | YES  |     | NULL    |       |
| Margin             | int          | YES  |     | NULL    |       |
| MatchNo            | varchar(100) | YES  |     | NULL    |       |
| MatchDate          | date         | YES  |     | NULL    |       |
| Umpire1ID          | int          | YES  | MUL | NULL    |       |
| Umpire2ID          | int          | YES  | MUL | NULL    |       |
| SuperOver          | varchar(100) | YES  |     | NULL    |       |
+--------------------+--------------+------+-----+---------+-------+
17 rows in set (0.26 sec)
```

playermatch The playermatch table is an intersection of the playermaster and matches table. Its sole purpose is to records the teams of each player playing in a match. The fields MatchID and PlayerID are a composite key to prevent duplication of records for each match.

```
-- ipldatabase.playermatch definition
CREATE TABLE 'playermatch' (
 'MatchID' int NOT NULL,
 'PlayerID' int NOT NULL,
 'TeamID' int DEFAULT NULL,
 PRIMARY KEY ('MatchID', 'PlayerID'),
 KEY 'playermatch_playermaster_FK' ('PlayerID'),
 KEY 'playermatch_teamsmaster_FK' ('TeamID'),
 CONSTRAINT 'playermatch_matches_FK' FOREIGN KEY ('MatchID') REFERENCES 'matches'
    ('MatchID'),
 CONSTRAINT 'playermatch_playermaster_FK' FOREIGN KEY ('PlayerID') REFERENCES
    'playermaster' ('PlayerID'),
 CONSTRAINT 'playermatch_teamsmaster_FK' FOREIGN KEY ('TeamID') REFERENCES
    'teamsmaster' ('TeamID')
) ENGINE = InnoDB DEFAULT CHARSET = latin1;
```

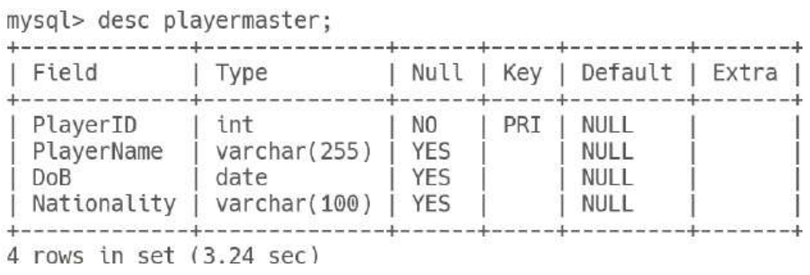


```
mysql> desc playermatch;
+----------+------+------+-----+---------+-------+
| Field    | Type | Null | Key | Default | Extra |
+----------+------+------+-----+---------+-------+
| MatchID  | int  | NO   | PRI | NULL    |       |
| PlayerID | int  | NO   | PRI | NULL    |       |
| TeamID   | int  | YES  | MUL | NULL    |       |
+----------+------+------+-----+---------+-------+
3 rows in set (1.04 sec)
```

playermaster This table stores details about cricket players, including player ID, player name, date of birth, and nationality.

It is referenced by the deliveries table to identify batsmen, bowlers, dismissed players, and fielders. The DDL command mentioned below was used to create the table in the database.

```
-- ipldatabase.playermaster definition
CREATE TABLE 'playermaster' (
 'PlayerID' int NOT NULL,
 'PlayerName' varchar(255) DEFAULT NULL,
 'DoB' date DEFAULT NULL,
 'Nationality' varchar(100) DEFAULT NULL,
 PRIMARY KEY ('PlayerID')
) ENGINE = InnoDB DEFAULT CHARSET = latin1;
```

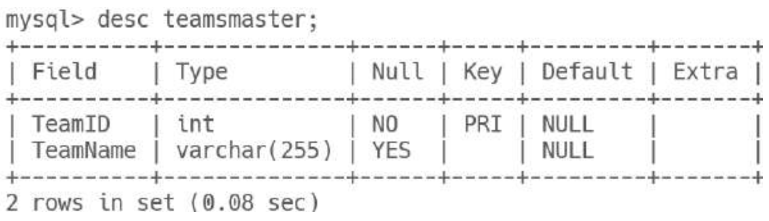


```
mysql> desc playermaster;
+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| PlayerID    | int          | NO   | PRI | NULL    |       |
| PlayerName  | varchar(255) | YES  |     | NULL    |       |
| DoB         | date         | YES  |     | NULL    |       |
| Nationality | varchar(100) | YES  |     | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
4 rows in set (3.24 sec)
```

teamsmaster This table contains information about cricket teams participating in the league, including team ID and team name.

It is referenced by the matches table to identify participating teams, winners and tosswinner. It is also used by the deliveries table to identify batting and bowling teams. The DDL command mentioned below was used to create the table in the database.

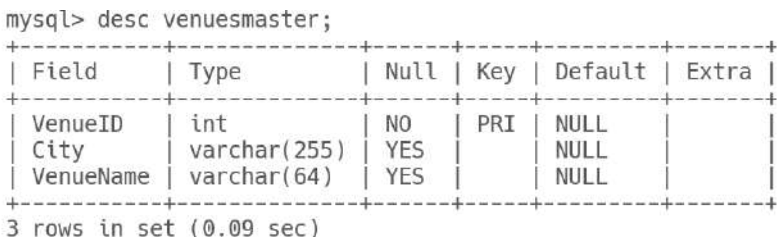
```
-- ipldatabase.teamsmaster definition
CREATE TABLE 'teamsmaster' (
 'TeamID' int NOT NULL,
 'TeamName' varchar(255) DEFAULT NULL,
 PRIMARY KEY ('TeamID')
) ENGINE = InnoDB DEFAULT CHARSET = latin1;
```



```
mysql> desc teamsmaster;
+----------+--------------+------+-----+---------+-------+
| Field    | Type         | Null | Key | Default | Extra |
+----------+--------------+------+-----+---------+-------+
| TeamID   | int          | NO   | PRI | NULL    |       |
| TeamName | varchar(255) | YES  |     | NULL    |       |
+----------+--------------+------+-----+---------+-------+
2 rows in set (0.08 sec)
```

venuesmaster This table stores details about cricket venues, including venue ID, venue name, and city. It is referenced by the matches table to identify the venue for each match. The DDL command mentioned below was used to create the table in the database.

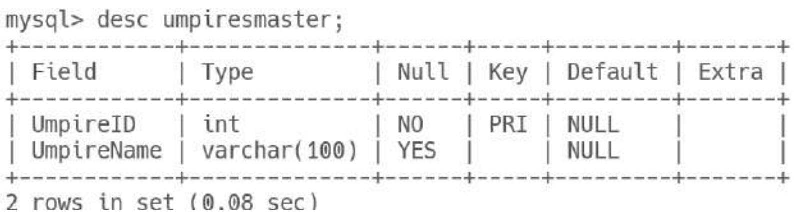
```
-- ipldatabase.venuesmaster definition
CREATE TABLE 'venuesmaster' (
 'VenueID' int NOT NULL,
 'City' varchar(255) DEFAULT NULL,
 'VenueName' varchar(64) DEFAULT NULL,
 PRIMARY KEY ('VenueID')
) ENGINE = InnoDB DEFAULT CHARSET = latin1;
```



```
mysql> desc venuesmaster;
+-----------+--------------+------+-----+---------+-------+
| Field     | Type         | Null | Key | Default | Extra |
+-----------+--------------+------+-----+---------+-------+
| VenueID   | int          | NO   | PRI | NULL    |       |
| City      | varchar(255) | YES  |     | NULL    |       |
| VenueName | varchar(64)  | YES  |     | NULL    |       |
+-----------+--------------+------+-----+---------+-------+
3 rows in set (0.09 sec)
```

umpiresmaster The umpiresmaster table has a comprehensive record of all the umpires who have officiated any IPL match from 2008 to 2022.

```
-- ipldatabase.umpiresmaster definition
CREATE TABLE 'umpiresmaster' (
 'UmpireID' int NOT NULL,
 'UmpireName' varchar(100) DEFAULT NULL,
 PRIMARY KEY ('UmpireID')
) ENGINE = InnoDB DEFAULT CHARSET = latin1;
```

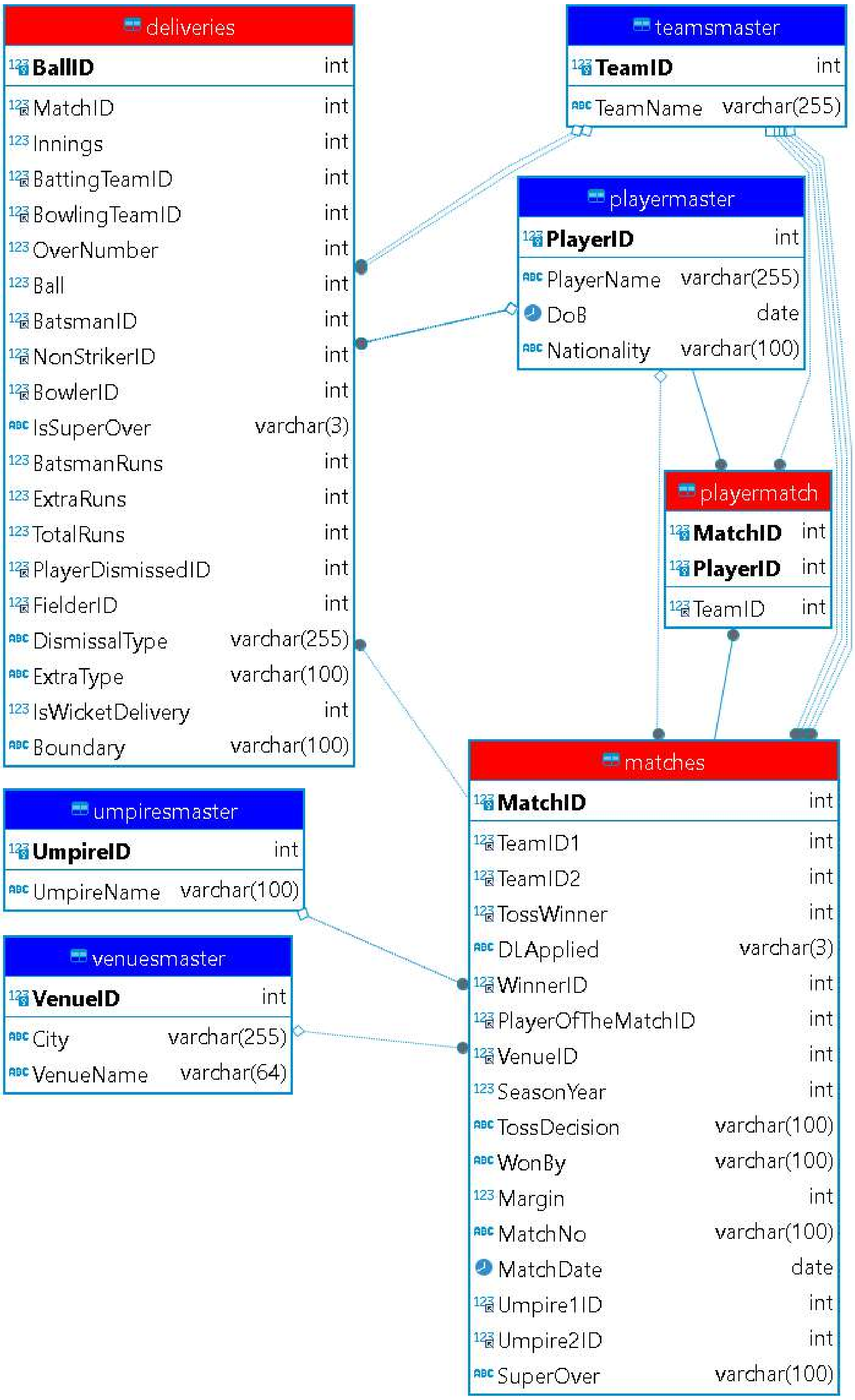


```
mysql> desc umpiresmaster;
+------------+--------------+------+-----+---------+-------+
| Field      | Type         | Null | Key | Default | Extra |
+------------+--------------+------+-----+---------+-------+
| UmpireID   | int          | NO   | PRI | NULL    |       |
| UmpireName | varchar(100) | YES  |     | NULL    |       |
+------------+--------------+------+-----+---------+-------+
2 rows in set (0.08 sec)
```

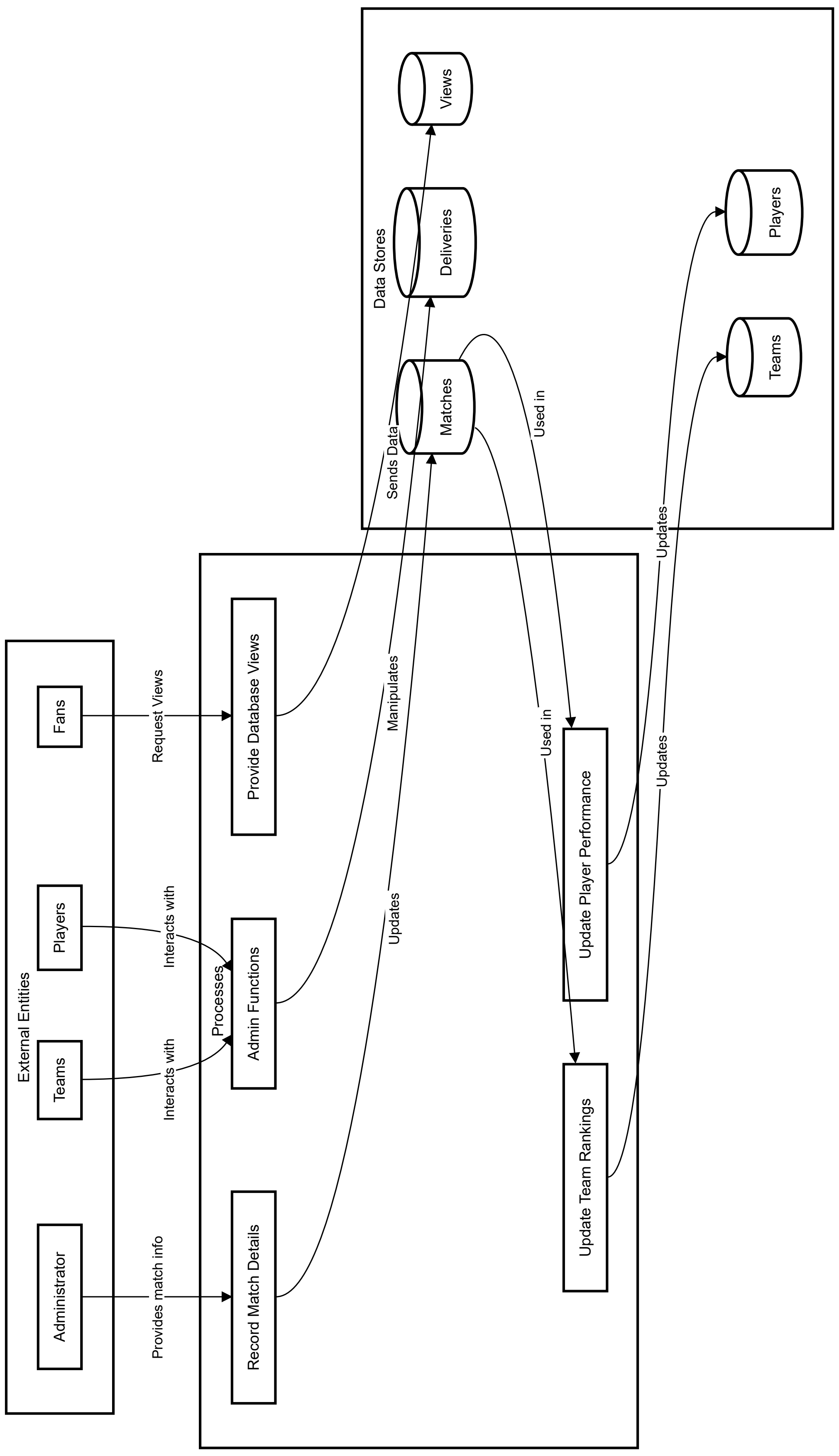
Record Details

Table Name	**Type**	**No. of Records**
matches	transaction	950
deliveries	transaction	2,25,954
playermatch	transaction	20,900
teamsmaster	master	19
playermaster	master	1,344
venuesmaster	master	52
umpiresmaster	master	66

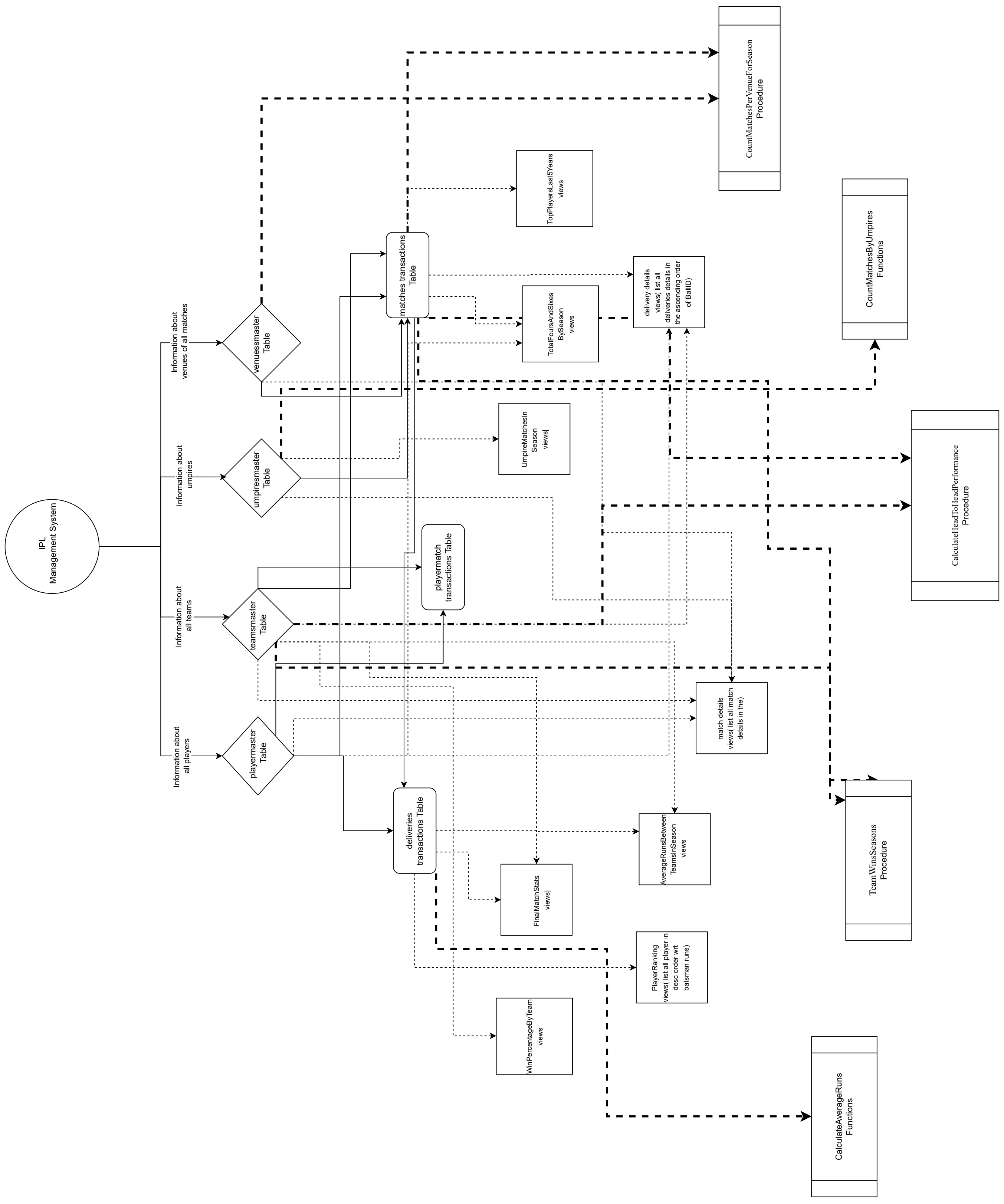
Entity Relationship Diagram

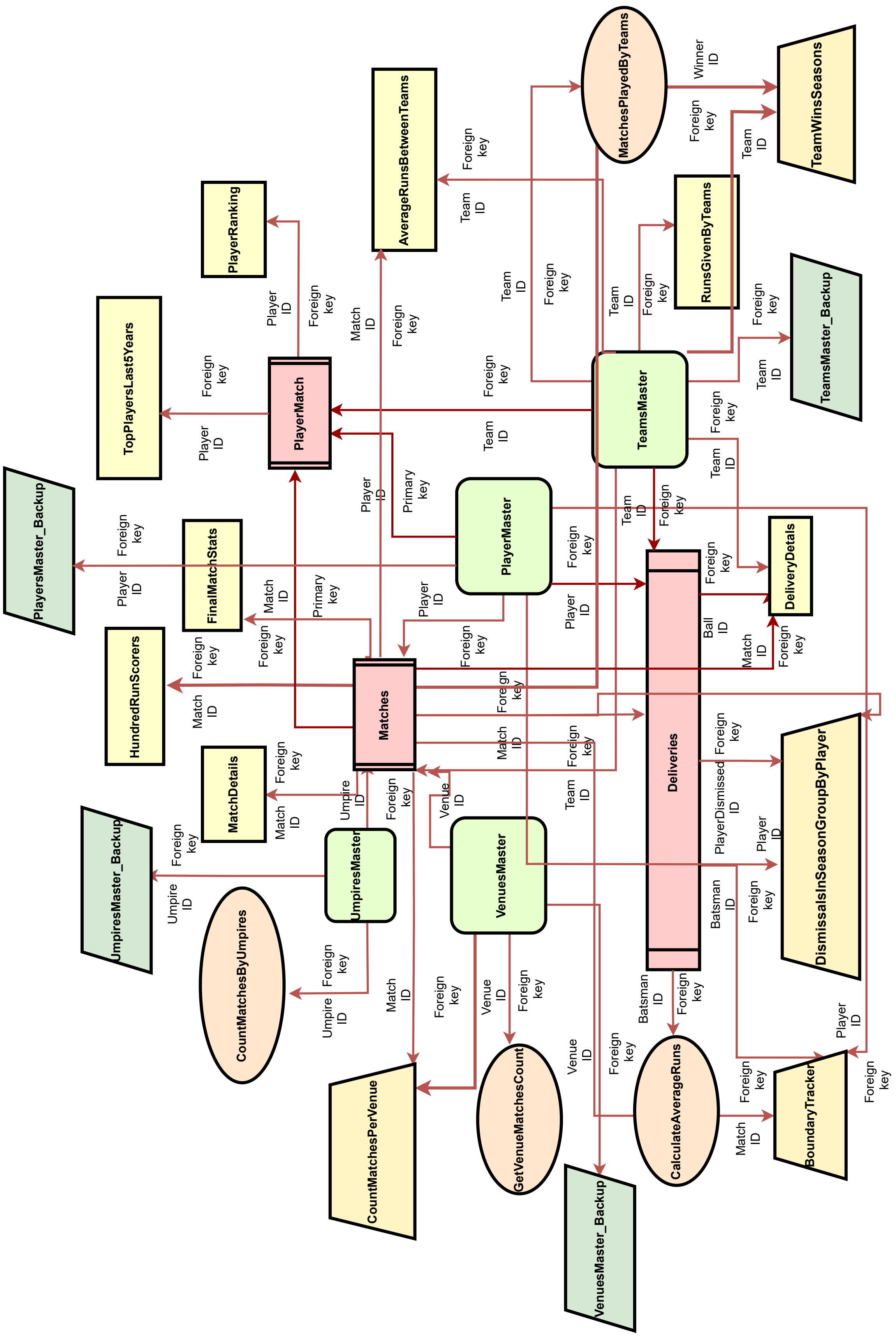


Flowchart


Data Stores
Views
Deliveries
Matches
Players
Teams
External Entities
Fans
Players
Teams
Administrator
Processes
Provide Database Views
Admin Functions
Record Match Details
Update Player Performance
Update Team Rankings
Request Views
Interacts with
Interacts with
Provides match info
Sends Data
Manipulates
Updates
Used in
Used in
Updates
Updates

Data Flow Diagram


IPL Management System
Information about venues of all matches
Information about umpires
Information about all teams
Information about all players
venuessmaster Table
umpiresmaster Table
teamsmaster Table
playermaster Table
matches transactions Table
playermatch transactions Table
deliveries transactions Table
TopPlayersLast5Years views
delivery details views(list all deliveries details in the ascending order of BallID)
TotalFoursAndSixes BySeason views
UmpireMatchesIn Season views|
match details views(list all match details in the)
AverageRunsBetween TeamsInSeason views
FinalMatchStats views|
PlayerRanking views(list all player in desc order wrt batsman runs)
WinPercentageByTeam views
CountMatchesPerVenueForSeason Procedure
CountMatchesByUmpires Functions
CalculateHeadToHeadPerformance Procedure
TeamWinsSeasons Procedure
CalculateAverageRuns Functions

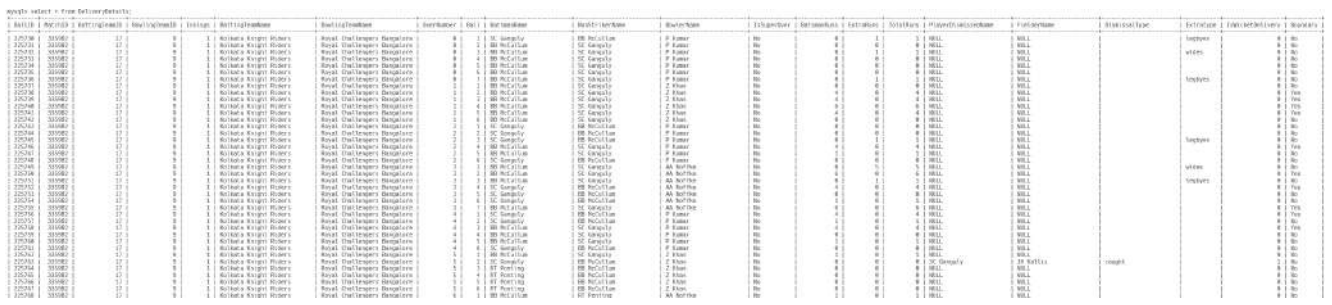

MatchesPlayedByTeams
TeamWinsSeasons
AverageRunsBetweenTeams
RunsGivenByTeams
PlayerRanking
TopPlayersLast5Years
PlayerMatch
TeamsMaster
TeamsMaster_Backup
PlayersMaster_Backup
FinalMatchStats
PlayerMaster
DeliveryDetails
HundredRunScorers
Matches
Deliveries
DismissalsInSeasonGroupByPlayer
MatchDetails
UmpiresMaster_Backup
UmpiresMaster
VenuesMaster
CountMatchesByUmpires
CountMatchesPerVenue
GetVenueMatchesCount
CalculateAverageRuns
BoundaryTracker
VenuesMaster_Backup
Winner ID
Foreign key
Team ID
Player ID
Match ID
Primary key
Ball ID
Umpire ID
Venue ID
PlayerDismissed ID
Batsman ID

Database Views

Q: What is the list of all deliveries in all seasons with all necessary details?

A: A view was created to list all the deliveries of all the details in the ascending order of BallID. Since the deliveries table is comprised of several foreign keys, making sense out of the raw table can be challenging for a user. Hence the relevant fields were inner joined with the master table to retrieve their respective records. The following code was used to create the view:

```sql
CREATE VIEW DeliveryDetails AS
SELECT d.BallID,
 m.MatchID,
 d.Innings,
 bt.TeamName AS BattingTeam,
 bb.TeamName AS BowlingTeam,
 d.Over,
 d.Ball,
 pb.PlayerName AS Batsman,
 ns.PlayerName AS NonStriker,
 bw.PlayerName AS Bowler,
 d.IsSuperOver,
 d.BatsmanRuns,
 d.ExtraRuns,
 d.TotalRuns,
 pd.PlayerName AS PlayerDismissed,
 fd.PlayerName AS Fielder,
 d.DismissalType,
 d.ExtraType,
 d.IsWicketDelivery,
 d.NonBoundary
FROM deliveries d
 INNER JOIN matches m ON d.MatchID = m.MatchID
 INNER JOIN teamsmaster bt ON d.BattingTeamID = bt.TeamID
 INNER JOIN teamsmaster bb ON d.BowlingTeamID = bb.TeamID
 LEFT JOIN playermaster pb ON d.BatsmanID = pb.PlayerID
 LEFT JOIN playermaster ns ON d.NonStrikerID = ns.PlayerID
 LEFT JOIN playermaster bw ON d.BowlerID = bw.PlayerID
 LEFT JOIN playermaster pd ON d.PlayerDismissedID = pd.PlayerID
 LEFT JOIN playermaster fd ON d.FielderID = fd.PlayerID;
```

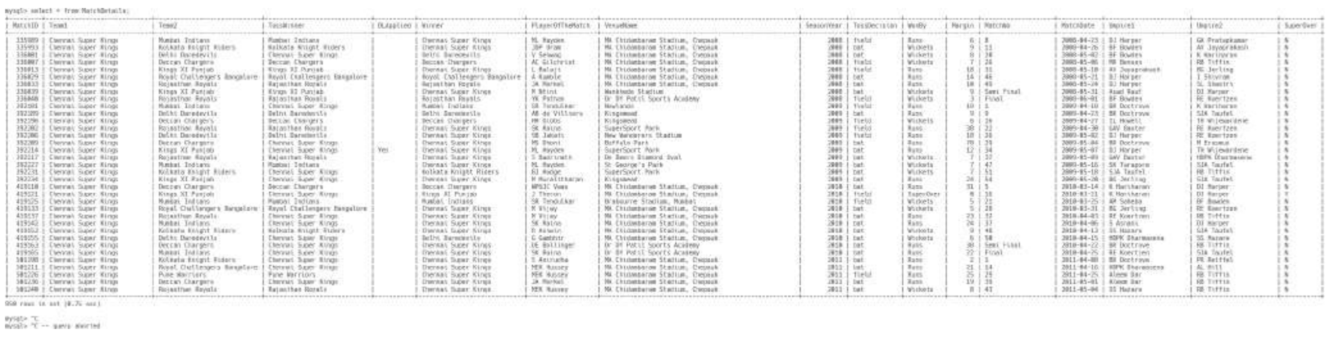


```
mysql> select * from DeliveryDetails;
[illegible]
```

Q: What is the list of all matches in all seasons inclusive of all the necessary details?

A: Similar to the view DeliveryDetails, we created the view MatchDetails. This view was created to ensure that the users can easily view information all information regarding all the matches from 2008 to 2017. Since the raw data of the Matches table contains several foreign keys it can be difficult for a user to make sense out of it.

```sql
CREATE VIEW MatchDetails AS
SELECT m.MatchID,
 t1.TeamName AS Team1,
 t2.TeamName AS Team2,
 tm.TeamName AS TossWinner,
 m.DLApplied,
 w.TeamName AS Winner,
 pm.PlayerName AS PlayerOfTheMatch,
 v.VenueName,
 m.SeasonYear,
 m.TossDecision,
 m.WonBy,
 m.Margin,
 m.MatchNo,
 m.MatchDate,
 u1.UmpireName AS Umpire1,
 u2.UmpireName AS Umpire2,
 m.SuperOver
FROM matches m
 INNER JOIN teamsmaster t1 ON m.TeamID1 = t1.TeamID
 INNER JOIN teamsmaster t2 ON m.TeamID2 = t2.TeamID
 LEFT JOIN teamsmaster tm ON m.TossWinner = tm.TeamID
 LEFT JOIN teamsmaster w ON m.WinnerID = w.TeamID
 LEFT JOIN playermaster pm ON m.PlayerOfTheMatchID = pm.PlayerID
 LEFT JOIN venuesmaster v ON m.VenueID = v.VenueID
 LEFT JOIN umpiresmaster u1 ON m.Umpire1ID = u1.UmpireID
 LEFT JOIN umpiresmaster u2 ON m.Umpire2ID = u2.UmpireID;
```



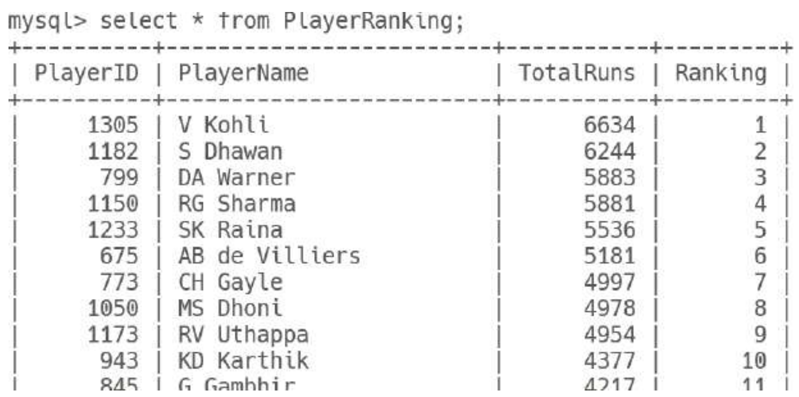
```
mysql> select * from MatchDetails;
+---------+---------------------+-----------------------------+-----------------------------+-----------+-----------------------------+----------------+---------------------------------+------------+--------------+-----------+--------+-------------+------------+-----------------+-----------------+-----------+
| MatchID | Team1               | Team2                       | TossWinner                  | DLApplied | Winner                      | PlayerOfTheMatch | VenueName                     | SeasonYear | TossDecision | WonBy     | Margin | MatchNo     | MatchDate  | Umpire1         | Umpire2         | SuperOver |
+---------+---------------------+-----------------------------+-----------------------------+-----------+-----------------------------+----------------+---------------------------------+------------+--------------+-----------+--------+-------------+------------+-----------------+-----------------+-----------+
|  335989 | Chennai Super Kings | Mumbai Indians              | Mumbai Indians              |           | Chennai Super Kings         | ML Hayden      | MA Chidambaram Stadium, Chepauk |       2008 | field        | Runs      |      6 | 8           | 2008-04-23 | DJ Harper       | GA Pratapkumar  | N         |
|  335993 | Chennai Super Kings | Kolkata Knight Riders       | Kolkata Knight Riders       |           | Chennai Super Kings         | JDP Oram       | MA Chidambaram Stadium, Chepauk |       2008 | bat          | Wickets   |      9 | 11          | 2008-04-26 | BF Bowden       | AV Jayaprakash  | N         |
|  336001 | Chennai Super Kings | Delhi Daredevils            | Chennai Super Kings         |           | Delhi Daredevils            | V Sehwag       | MA Chidambaram Stadium, Chepauk |       2008 | bat          | Wickets   |      8 | 20          | 2008-05-02 | BF Bowden       | K Hariharan     | N         |
|  336007 | Chennai Super Kings | Deccan Chargers             | Deccan Chargers             |           | Deccan Chargers             | AC Gilchrist   | MA Chidambaram Stadium, Chepauk |       2008 | field        | Wickets   |      7 | 26          | 2008-05-06 | MR Benson       | RB Tiffin       | N         |
|  336013 | Chennai Super Kings | Kings XI Punjab             | Kings XI Punjab             |           | Chennai Super Kings         | L Balaji       | MA Chidambaram Stadium, Chepauk |       2008 | field        | Runs      |     18 | 31          | 2008-05-10 | AV Jayaprakash  | BG Jerling      | N         |
|  336029 | Chennai Super Kings | Royal Challengers Bangalore | Royal Challengers Bangalore |           | Royal Challengers Bangalore | A Kumble       | MA Chidambaram Stadium, Chepauk |       2008 | bat          | Runs      |     14 | 46          | 2008-05-21 | DJ Harper       | I Shivram       | N         |
|  336033 | Chennai Super Kings | Rajasthan Royals            | Rajasthan Royals            |           | Rajasthan Royals            | JA Morkel      | MA Chidambaram Stadium, Chepauk |       2008 | bat          | Runs      |     10 | 49          | 2008-05-24 | DJ Harper       | SL Shastri      | N         |
|  336039 | Chennai Super Kings | Kings XI Punjab             | Kings XI Punjab             |           | Chennai Super Kings         | M Ntini        | Wankhede Stadium                |       2008 | bat          | Wickets   |      9 | Semi Final  | 2008-05-31 | Asad Rauf       | DJ Harper       | N         |
|  336040 | Chennai Super Kings | Rajasthan Royals            | Rajasthan Royals            |           | Rajasthan Royals            | YK Pathan      | Dr DY Patil Sports Academy      |       2008 | field        | Wickets   |      3 | Final       | 2008-06-01 | BF Bowden       | RE Koertzen     | N         |
|  392181 | Chennai Super Kings | Mumbai Indians              | Chennai Super Kings         |           | Mumbai Indians              | SR Tendulkar   | Newlands                        |       2009 | field        | Runs      |     19 | 1           | 2009-04-18 | BR Doctrove     | K Hariharan     | N         |
|  392189 | Chennai Super Kings | Delhi Daredevils            | Delhi Daredevils            |           | Delhi Daredevils            | AB de Villiers | Kingsmead                       |       2009 | bat          | Runs      |      9 | 9           | 2009-04-23 | BR Doctrove     | SJA Taufel      | N         |
|  392196 | Chennai Super Kings | Deccan Chargers             | Deccan Chargers             |           | Deccan Chargers             | HH Gibbs       | Kingsmead                       |       2009 | field        | Wickets   |      6 | 16          | 2009-04-27 | IL Howell       | TH Wijewardene  | N         |
|  392202 | Chennai Super Kings | Rajasthan Royals            | Rajasthan Royals            |           | Chennai Super Kings         | SK Raina       | SuperSport Park                 |       2009 | field        | Runs      |     38 | 22          | 2009-04-30 | GAV Baxter      | RE Koertzen     | N         |
|  392206 | Chennai Super Kings | Delhi Daredevils            | Delhi Daredevils            |           | Chennai Super Kings         | SB Jakati      | New Wanderers Stadium           |       2009 | field        | Runs      |     18 | 26          | 2009-05-02 | DJ Harper       | RE Koertzen     | N         |
|  392209 | Chennai Super Kings | Deccan Chargers             | Chennai Super Kings         |           | Chennai Super Kings         | MS Dhoni       | Buffalo Park                    |       2009 | bat          | Runs      |     78 | 29          | 2009-05-04 | BR Doctrove     | M Erasmus       | N         |
|  392214 | Chennai Super Kings | Kings XI Punjab             | Chennai Super Kings         | Yes       | Chennai Super Kings         | ML Hayden      | SuperSport Park                 |       2009 | bat          | Runs      |     12 | 34          | 2009-05-07 | DJ Harper       | TH Wijewardene  | N         |
|  392217 | Chennai Super Kings | Rajasthan Royals            | Rajasthan Royals            |           | Chennai Super Kings         | S Badrinath    | De Beers Diamond Oval           |       2009 | bat          | Wickets   |      7 | 37          | 2009-05-09 | GAV Baxter      | HDPK Dharmasena | N         |
|  392227 | Chennai Super Kings | Mumbai Indians              | Mumbai Indians              |           | Chennai Super Kings         | ML Hayden      | St George's Park                |       2009 | bat          | Wickets   |      7 | 47          | 2009-05-16 | SK Tarapore     | SJA Taufel      | N         |
|  392231 | Chennai Super Kings | Kolkata Knight Riders       | Chennai Super Kings         |           | Kolkata Knight Riders       | BJ Hodge       | SuperSport Park                 |       2009 | bat          | Wickets   |      7 | 51          | 2009-05-18 | SJA Taufel      | RB Tiffin       | N         |
|  392234 | Chennai Super Kings | Kings XI Punjab             | Chennai Super Kings         |           | Chennai Super Kings         | M Muralitharan | Kingsmead                       |       2009 | bat          | Runs      |     24 | 54          | 2009-05-20 | BG Jerling      | SJA Taufel      | N         |
|  419110 | Chennai Super Kings | Deccan Chargers             | Deccan Chargers             |           | Deccan Chargers             | WPUJC Vaas     | MA Chidambaram Stadium, Chepauk |       2010 | bat          | Runs      |     31 | 5           | 2010-03-14 | K Hariharan     | DJ Harper       | N         |
|  419121 | Chennai Super Kings | Kings XI Punjab             | Chennai Super Kings         |           | Kings XI Punjab             | J Theron       | MA Chidambaram Stadium, Chepauk |       2010 | field        | SuperOver |      0 | 16          | 2010-03-21 | K Hariharan     | DJ Harper       | Y         |
|  419125 | Chennai Super Kings | Mumbai Indians              | Mumbai Indians              |           | Mumbai Indians              | SR Tendulkar   | Brabourne Stadium, Mumbai       |       2010 | field        | Wickets   |      5 | 21          | 2010-03-25 | AM Saheba       | BF Bowden       | N         |
|  419133 | Chennai Super Kings | Royal Challengers Bangalore | Royal Challengers Bangalore |           | Chennai Super Kings         | M Vijay        | MA Chidambaram Stadium, Chepauk |       2010 | bat          | Wickets   |      5 | 28          | 2010-03-31 | BG Jerling      | RE Koertzen     | N         |
|  419137 | Chennai Super Kings | Rajasthan Royals            | Chennai Super Kings         |           | Chennai Super Kings         | M Vijay        | MA Chidambaram Stadium, Chepauk |       2010 | bat          | Runs      |     23 | 32          | 2010-04-03 | RE Koertzen     | RB Tiffin       | N         |
|  419142 | Chennai Super Kings | Mumbai Indians              | Chennai Super Kings         |           | Chennai Super Kings         | SK Raina       | MA Chidambaram Stadium, Chepauk |       2010 | bat          | Runs      |     24 | 37          | 2010-04-06 | S Asnani        | DJ Harper       | N         |
|  419152 | Chennai Super Kings | Kolkata Knight Riders       | Kolkata Knight Riders       |           | Chennai Super Kings         | R Ashwin       | MA Chidambaram Stadium, Chepauk |       2010 | bat          | Wickets   |      9 | 46          | 2010-04-13 | SS Hazare       | SJA Taufel      | N         |
|  419155 | Chennai Super Kings | Delhi Daredevils            | Chennai Super Kings         |           | Delhi Daredevils            | G Gambhir      | MA Chidambaram Stadium, Chepauk |       2010 | bat          | Wickets   |      6 | 50          | 2010-04-15 | HDPK Dharmasena | SS Hazare       | N         |
|  419163 | Chennai Super Kings | Deccan Chargers             | Chennai Super Kings         |           | Chennai Super Kings         | DE Bollinger   | Dr DY Patil Sports Academy      |       2010 | bat          | Runs      |     38 | Semi Final  | 2010-04-22 | BR Doctrove     | RB Tiffin       | N         |
|  419165 | Chennai Super Kings | Mumbai Indians              | Chennai Super Kings         |           | Chennai Super Kings         | SK Raina       | Dr DY Patil Sports Academy      |       2010 | bat          | Runs      |     22 | Final       | 2010-04-25 | RE Koertzen     | SJA Taufel      | N         |
|  501198 | Chennai Super Kings | Kolkata Knight Riders       | Chennai Super Kings         |           | Chennai Super Kings         | S Anirudha     | MA Chidambaram Stadium, Chepauk |       2011 | bat          | Runs      |      2 | 1           | 2011-04-08 | BR Doctrove     | PR Reiffel      | N         |
|  501211 | Chennai Super Kings | Royal Challengers Bangalore | Chennai Super Kings         |           | Chennai Super Kings         | MEK Hussey     | MA Chidambaram Stadium, Chepauk |       2011 | bat          | Runs      |     21 | 14          | 2011-04-16 | HDPK Dharmasena | AL Hill         | N         |
|  501226 | Chennai Super Kings | Pune Warriors               | Pune Warriors               |           | Chennai Super Kings         | MEK Hussey     | MA Chidambaram Stadium, Chepauk |       2011 | field        | Runs      |     25 | 29          | 2011-04-25 | Aleem Dar       | RB Tiffin       | N         |
|  501230 | Chennai Super Kings | Deccan Chargers             | Chennai Super Kings         |           | Chennai Super Kings         | JA Morkel      | MA Chidambaram Stadium, Chepauk |       2011 | bat          | Runs      |     19 | 33          | 2011-05-01 | Aleem Dar       | RB Tiffin       | N         |
|  501240 | Chennai Super Kings | Rajasthan Royals            | Rajasthan Royals            |           | Chennai Super Kings         | MEK Hussey     | MA Chidambaram Stadium, Chepauk |       2011 | bat          | Wickets   |      8 | 41          | 2011-05-04 | SS Hazare       | RB Tiffin       | N         |
+---------+---------------------+-----------------------------+-----------------------------+-----------+-----------------------------+----------------+---------------------------------+------------+--------------+-----------+--------+-------------+------------+-----------------+-----------------+-----------+
950 rows in set (0.76 sec)

mysql> ^C
mysql> ^C -- query aborted
```

Q: What is the ranking of all the players batsmen in IPL?

A: To rank all the players in descending order, Batsman Runs was used as a metreic. The sum of their Batsman Runs was done and rank number was given to them using the `Row_Number()` function.

```sql
CREATE VIEW PlayerRanking AS
SELECT p.PlayerID,
 p.PlayerName,
 SUM(d.BatsmanRuns) AS TotalRuns,
 ROW_NUMBER() OVER (
  ORDER BY SUM(d.BatsmanRuns) DESC
 ) AS Ranking
FROM playermaster p
 INNER JOIN deliveries d ON p.PlayerID = d.BatsmanID
GROUP BY p.PlayerID,
 p.PlayerName;
```

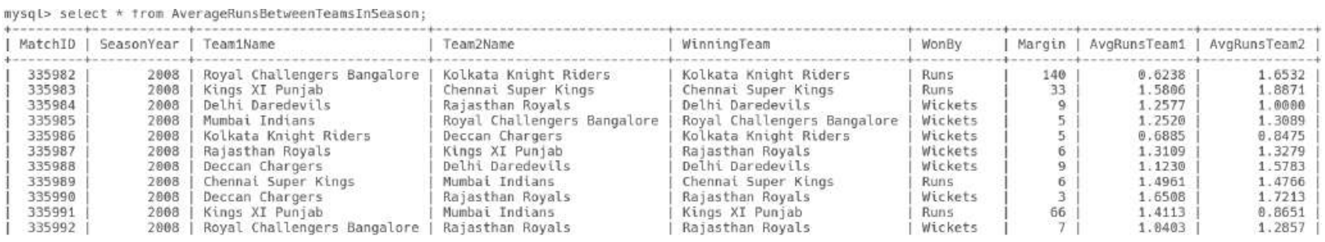


```
mysql> select * from PlayerRanking;
+----------+------------------------+-----------+---------+
| PlayerID | PlayerName             | TotalRuns | Ranking |
+----------+------------------------+-----------+---------+
|     1305 | V Kohli                |      6634 |       1 |
|     1182 | S Dhawan               |      6244 |       2 |
|      799 | DA Warner              |      5883 |       3 |
|     1150 | RG Sharma              |      5881 |       4 |
|     1233 | SK Raina               |      5536 |       5 |
|      675 | AB de Villiers         |      5181 |       6 |
|      773 | CH Gayle               |      4997 |       7 |
|     1050 | MS Dhoni               |      4978 |       8 |
|     1173 | RV Uthappa             |      4954 |       9 |
|      943 | KD Karthik             |      4377 |      10 |
|      845 | G Gambhir              |      4217 |      11 |
```

Q: What was the summary of all the matches for all the seasons?

A: The fields TeamID1, TeamID2, WiinerID were used to call TeamNames. Whileas the WonBy field and the Margin field was called from the matches table directly. Average runs for both the teams was calculated to have some reasonable assesment of each match.

```sql
CREATE VIEW AverageRunsBetweenTeamsInSeason AS
SELECT m.MatchID,
 m.SeasonYear,
 t1.TeamName AS Team1Name,
 t2.TeamName AS Team2Name,
 CASE
  WHEN m.WinnerID = m.TeamID1 THEN t1.TeamName
  WHEN m.WinnerID = m.TeamID2 THEN t2.TeamName
  ELSE 'Draw'
 END AS WinningTeam,
 CASE
  WHEN m.WinnerID = m.TeamID1 THEN m.Margin
  WHEN m.WinnerID = m.TeamID2 THEN m.Margin
  ELSE 0
 END AS MarginOfVictory,
 AVG(
  CASE
   WHEN d.BattingTeamID = m.TeamID1 THEN d.TotalRuns
   ELSE 0
  END
 ) AS AvgRunsTeam1,
 AVG(
  CASE
   WHEN d.BattingTeamID = m.TeamID2 THEN d.TotalRuns
   ELSE 0
  END
 ) AS AvgRunsTeam2
FROM matches m
 JOIN deliveries d ON m.MatchID = d.MatchID
 JOIN teamsmaster t1 ON m.TeamID1 = t1.TeamID
 JOIN teamsmaster t2 ON m.TeamID2 = t2.TeamID
GROUP BY m.MatchID,
 m.SeasonYear,
 t1.TeamName,
 t2.TeamName,
 m.WinnerID,
 m.Margin;
```

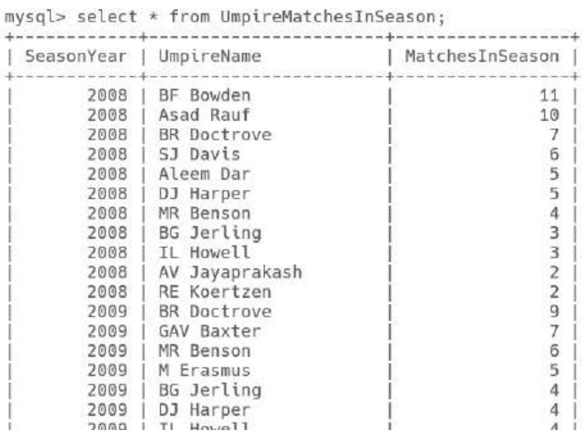


```
mysql> select * from AverageRunsBetweenTeamsInSeason;
+---------+------------+-----------------------------+-----------------------------+-----------------------------+---------+--------+--------------+--------------+
| MatchID | SeasonYear | Team1Name                   | Team2Name                   | WinningTeam                 | WonBy   | Margin | AvgRunsTeam1 | AvgRunsTeam2 |
+---------+------------+-----------------------------+-----------------------------+-----------------------------+---------+--------+--------------+--------------+
|  335982 |       2008 | Royal Challengers Bangalore | Kolkata Knight Riders       | Kolkata Knight Riders       | Runs    |    140 |       0.6238 |       1.6532 |
|  335983 |       2008 | Kings XI Punjab             | Chennai Super Kings         | Chennai Super Kings         | Runs    |     33 |       1.5806 |       1.8871 |
|  335984 |       2008 | Delhi Daredevils            | Rajasthan Royals            | Delhi Daredevils            | Wickets |      9 |       1.2577 |       1.0000 |
|  335985 |       2008 | Mumbai Indians              | Royal Challengers Bangalore | Royal Challengers Bangalore | Wickets |      5 |       1.2520 |       1.3089 |
|  335986 |       2008 | Kolkata Knight Riders       | Deccan Chargers             | Kolkata Knight Riders       | Wickets |      5 |       0.6885 |       0.8475 |
|  335987 |       2008 | Rajasthan Royals            | Kings XI Punjab             | Rajasthan Royals            | Wickets |      6 |       1.3109 |       1.3279 |
|  335988 |       2008 | Deccan Chargers             | Delhi Daredevils            | Delhi Daredevils            | Wickets |      9 |       1.1230 |       1.5783 |
|  335989 |       2008 | Chennai Super Kings         | Mumbai Indians              | Chennai Super Kings         | Runs    |      6 |       1.4961 |       1.4766 |
|  335990 |       2008 | Deccan Chargers             | Rajasthan Royals            | Rajasthan Royals            | Wickets |      3 |       1.6508 |       1.7213 |
|  335991 |       2008 | Kings XI Punjab             | Mumbai Indians              | Kings XI Punjab             | Runs    |     66 |       1.4113 |       0.8651 |
|  335992 |       2008 | Royal Challengers Bangalore | Rajasthan Royals            | Rajasthan Royals            | Wickets |      7 |       1.0403 |       1.2857 |
```

Q: How many matches were officiated by all the umpires in all the seasons?

A: To find out the count of matches for each umpire, a simple `Count()` funciton was used. The grouping was done by UmpireName and SeasonYear to retrieve the records for each umpire in each season.

```sql
CREATE VIEW UmpireMatchesInSeason AS
SELECT SeasonYear,
 UmpireName,
 COUNT(*) AS MatchesInSeason
FROM matches
 INNER JOIN umpiresmaster ON matches.Umpire1ID = umpiresmaster.UmpireID
GROUP BY SeasonYear,
 UmpireName
ORDER BY SeasonYear,
 MatchesInSeason DESC;
```

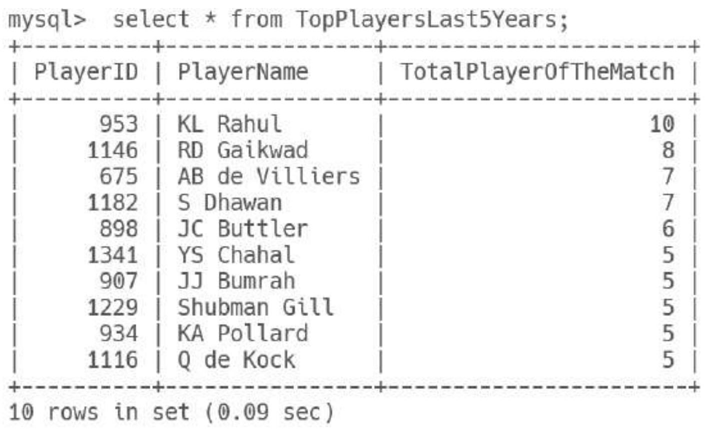


```
mysql> select * from UmpireMatchesInSeason;
+------------+-------------------------+-----------------+
| SeasonYear | UmpireName              | MatchesInSeason |
+------------+-------------------------+-----------------+
|       2008 | BF Bowden               |              11 |
|       2008 | Asad Rauf               |              10 |
|       2008 | BR Doctrove             |               7 |
|       2008 | SJ Davis                |               6 |
|       2008 | Aleem Dar               |               5 |
|       2008 | DJ Harper               |               5 |
|       2008 | MR Benson               |               4 |
|       2008 | BG Jerling              |               3 |
|       2008 | IL Howell               |               3 |
|       2008 | AV Jayaprakash          |               2 |
|       2008 | RE Koertzen             |               2 |
|       2009 | BR Doctrove             |               9 |
|       2009 | GAV Baxter              |               7 |
|       2009 | MR Benson               |               6 |
|       2009 | M Erasmus               |               5 |
|       2009 | BG Jerling              |               4 |
|       2009 | DJ Harper               |               4 |
|       2009 | IL Howell               |               4 |
```

Q: Who were the top players in the past 5 years of IPL

A: To retrieve the top players in IPL, the PlayerOfTheMatch field from the matches table was used. This is a good metric to get the best players of the tournmanet. We ranked the players in descending order of the number of PlayerOfTheMatches awarded to them.

```sql
CREATE VIEW TopPlayersLast5Years AS
SELECT PlayerID,
 PlayerName,
 COUNT(*) AS TotalPlayerOfTheMatch
FROM playermaster
 INNER JOIN matches ON playermaster.PlayerID = matches.PlayerOfTheMatchID
WHERE matches.MatchDate >= DATE_SUB(CURRENT_DATE(), INTERVAL 5 YEAR)
GROUP BY PlayerID,
 PlayerName
ORDER BY TotalPlayerOfTheMatch DESC
LIMIT 10;
```



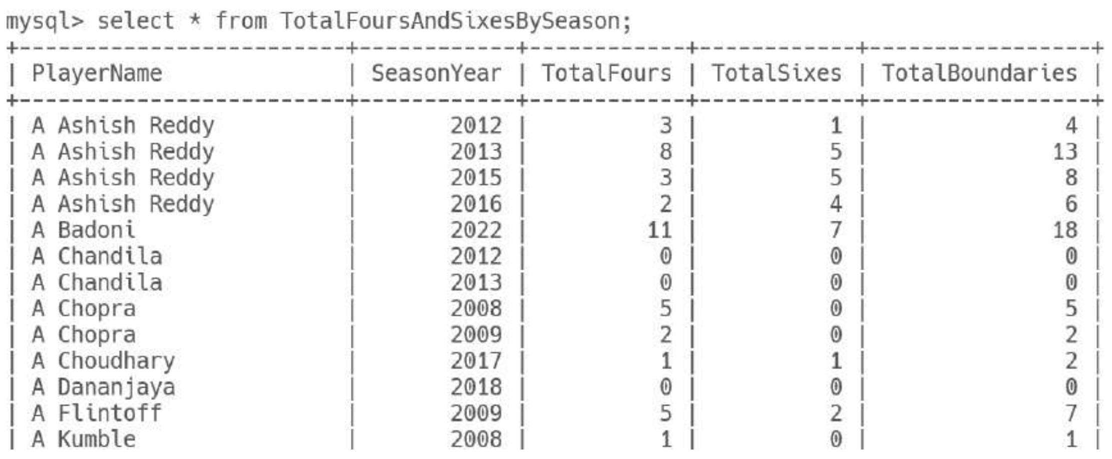
```
mysql>  select * from TopPlayersLast5Years;
+----------+----------------+----------------------+
| PlayerID | PlayerName     | TotalPlayerOfTheMatch |
+----------+----------------+----------------------+
|      953 | KL Rahul       |                   10 |
|     1146 | RD Gaikwad     |                    8 |
|      675 | AB de Villiers |                    7 |
|     1182 | S Dhawan       |                    7 |
|      898 | JC Buttler     |                    6 |
|     1341 | YS Chahal      |                    5 |
|      907 | JJ Bumrah      |                    5 |
|     1229 | Shubman Gill   |                    5 |
|      934 | KA Pollard     |                    5 |
|     1116 | Q de Kock      |                    5 |
+----------+----------------+----------------------+
10 rows in set (0.09 sec)
```

Q: What were the total number of 4's & 6's by all the players in all the seasons?

A: We utilized the sum-case method to count the instance wherin the batsman runs were either 4 or 6. Grouping was done by PlayerName and SeasonYear to given 4's and 6's for each year for each player. This view helps stakeholders understand which players hit the most boundaries.

```sql
CREATE VIEW TotalFoursAndSixesBySeason AS
SELECT p.PlayerName,
 m.SeasonYear,
 COUNT(
  CASE
   WHEN d.BatsmanRuns = 4 THEN 1
  END
 ) AS TotalFours,
 COUNT(
  CASE
   WHEN d.BatsmanRuns = 6 THEN 1
  END
 ) AS TotalSixes,
 COUNT(
  CASE
   WHEN d.BatsmanRuns = 4
   OR d.BatsmanRuns = 6 THEN 1
  END
 ) AS TotalBoundaries
```

```sql
FROM deliveries d
 JOIN playermaster p ON d.BatsmanID = p.PlayerID
 JOIN matches m ON d.MatchID = m.MatchID
GROUP BY p.PlayerID,
 p.PlayerName,
 m.SeasonYear;
```

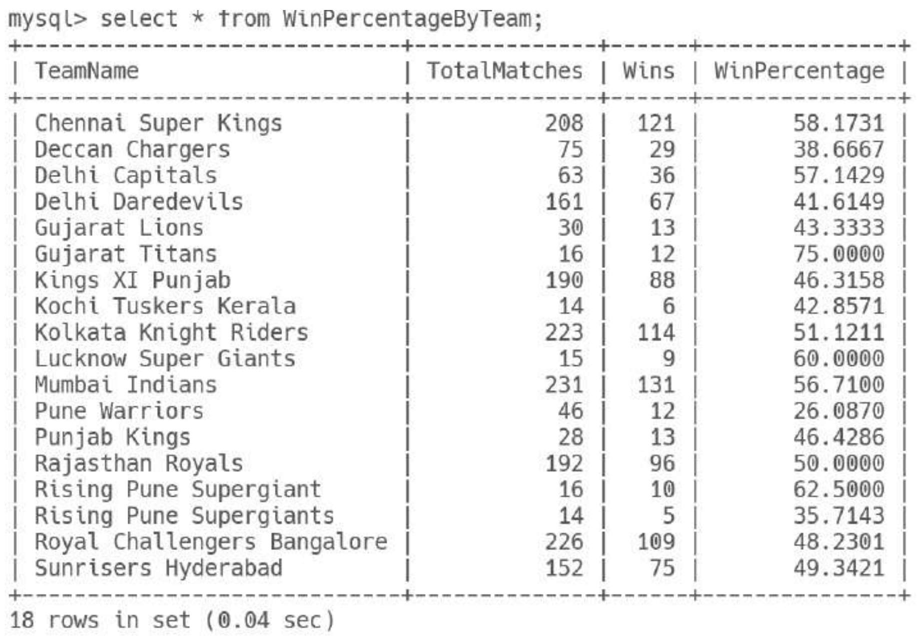


```
mysql> select * from TotalFoursAndSixesBySeason;
+-------------------------+------------+------------+------------+-----------------+
| PlayerName              | SeasonYear | TotalFours | TotalSixes | TotalBoundaries |
+-------------------------+------------+------------+------------+-----------------+
| A Ashish Reddy          |       2012 |          3 |          1 |               4 |
| A Ashish Reddy          |       2013 |          8 |          5 |              13 |
| A Ashish Reddy          |       2015 |          3 |          5 |               8 |
| A Ashish Reddy          |       2016 |          2 |          4 |               6 |
| A Badoni                |       2022 |         11 |          7 |              18 |
| A Chandila              |       2012 |          0 |          0 |               0 |
| A Chandila              |       2013 |          0 |          0 |               0 |
| A Chopra                |       2008 |          5 |          0 |               5 |
| A Chopra                |       2009 |          2 |          0 |               2 |
| A Choudhary             |       2017 |          1 |          1 |               2 |
| A Dananjaya             |       2018 |          0 |          0 |               0 |
| A Flintoff              |       2009 |          5 |          2 |               7 |
| A Kumble                |       2008 |          1 |          0 |               1 |
```

Q: What was the win percentage of all teams consolidating all seasons?

A: Wins percentage is a good metric to understand each teams performance. We also include the TotalMatches column in order to make a viable the assessment of the teams who may be new to the tournament. Teams like *Gujrat Titans* have the best wins percentage, however they have only played 16 matches compared to older teams. The sum case method was once again used to count the wins for teams and the output was grouped TeamID & TeamName.

```sql
CREATE VIEW WinPercentageByTeam AS
SELECT t.TeamName,
 COUNT(*) AS TotalMatches,
 SUM(
  CASE
   WHEN m.WinnerID = t.TeamID THEN 1
   ELSE 0
  END
 ) AS Wins,
 (
  SUM(
   CASE
    WHEN m.WinnerID = t.TeamID THEN 1
    ELSE 0
   END
  ) / COUNT(*)
 ) * 100 AS WinPercentage
FROM matches m
 JOIN teamsmaster t ON m.TeamID1 = t.TeamID
 OR m.TeamID2 = t.TeamID
GROUP BY t.TeamID,
 t.TeamName;
```



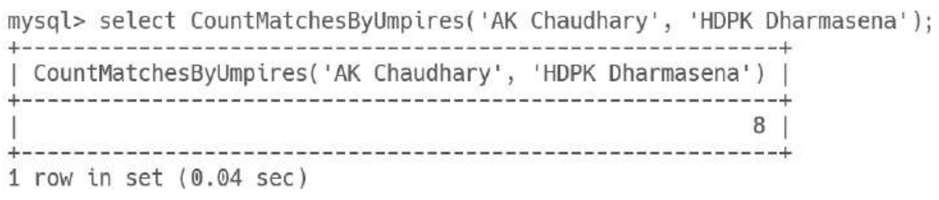
```
mysql> select * from WinPercentageByTeam;
+-----------------------------+--------------+------+---------------+
| TeamName                    | TotalMatches | Wins | WinPercentage |
+-----------------------------+--------------+------+---------------+
| Chennai Super Kings         |          208 |  121 |       58.1731 |
| Deccan Chargers             |           75 |   29 |       38.6667 |
| Delhi Capitals              |           63 |   36 |       57.1429 |
| Delhi Daredevils            |          161 |   67 |       41.6149 |
| Gujarat Lions               |           30 |   13 |       43.3333 |
| Gujarat Titans              |           16 |   12 |       75.0000 |
| Kings XI Punjab             |          190 |   88 |       46.3158 |
| Kochi Tuskers Kerala        |           14 |    6 |       42.8571 |
| Kolkata Knight Riders       |          223 |  114 |       51.1211 |
| Lucknow Super Giants        |           15 |    9 |       60.0000 |
| Mumbai Indians              |          231 |  131 |       56.7100 |
| Pune Warriors               |           46 |   12 |       26.0870 |
| Punjab Kings                |           28 |   13 |       46.4286 |
| Rajasthan Royals            |          192 |   96 |       50.0000 |
| Rising Pune Supergiant      |           16 |   10 |       62.5000 |
| Rising Pune Supergiants     |           14 |    5 |       35.7143 |
| Royal Challengers Bangalore |          226 |  109 |       48.2301 |
| Sunrisers Hyderabad         |          152 |   75 |       49.3421 |
+-----------------------------+--------------+------+---------------+
18 rows in set (0.04 sec)
```

Database Functions

Q: What is the count of matches officiated by two given umpires

A: A function was created to count the matches officiated by two given umpires together. The function first declares two temporary variables Umpire1_ ID & Umpire2_ ID. Values to these variables are assigned using the select into statement with a where clause to match the provided names in the umpiresmaster table. These variables and than used in a second select statement with a where clause to identify if they officiated any matches together.

```sql
DELIMITER //
CREATE FUNCTION CountMatchesByUmpires(
 umpire1_name VARCHAR(100),
 umpire2_name VARCHAR(100)
) RETURNS INT
BEGIN
DECLARE umpire1_id,
 umpire2_id INT;
DECLARE total_matches INT;
SELECT UmpireID INTO umpire1_id
FROM umpiresmaster
WHERE UmpireName = umpire1_name;
SELECT UmpireID INTO umpire2_id
FROM umpiresmaster
WHERE UmpireName = umpire2_name;
SELECT COUNT(*) INTO total_matches
FROM matches
WHERE Umpire1ID = umpire1_id
 and Umpire2ID = umpire2_id
 OR (
  Umpire1ID = umpire2_id
  and Umpire2ID = umpire1_id
 );
RETURN total_matches;
END //
DELIMITER;
```



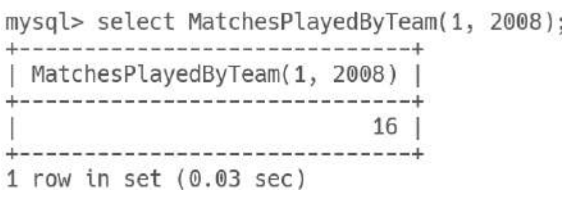
```
mysql> select CountMatchesByUmpires('AK Chaudhary', 'HDPK Dharmasena');
+--------------------------------------------------------+
| CountMatchesByUmpires('AK Chaudhary', 'HDPK Dharmasena') |
+--------------------------------------------------------+
|                                                      8 |
+--------------------------------------------------------+
1 row in set (0.04 sec)
```

Q: What were the total matches played by a team in a given season

A: The function uses the `count()` function to count the matches where it matches the users input of TeamID & SeasonYear.

```sql
DELIMITER //
CREATE FUNCTION MatchesPlayedByTeam(team_id INT, season_year INT) RETURNS INT
BEGIN
DECLARE total_matches INT;
SELECT COUNT(*) INTO total_matches
```

```
FROM matches
WHERE (
  TeamID1 = team_id
  OR TeamID2 = team_id
 )
 AND SeasonYear = season_year;
RETURN total_matches;
END //
DELIMITER;
```

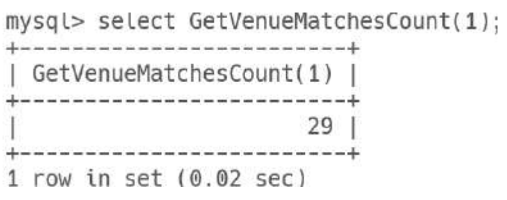


```
mysql> select MatchesPlayedByTeam(1, 2008);
+------------------------------+
| MatchesPlayedByTeam(1, 2008) |
+------------------------------+
|                           16 |
+------------------------------+
1 row in set (0.03 sec)
```

Q: What is the total number of matches played at a given venue?

A: The function use the `count()` function to count the matches. A where clause filter the matches wherin the VenueID matches the given VenueID.

```
DELIMITER //
CREATE FUNCTION GetVenueMatchesCount(venue_id INT) RETURNS INT
BEGIN
DECLARE matches_count INT;
SELECT COUNT(*) INTO matches_count
FROM matches
WHERE VenueID = venue_id;
RETURN matches_count;
END //
DELIMITER;
```



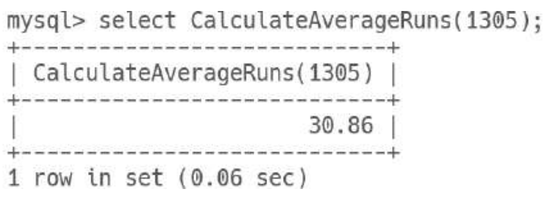
```
mysql> select GetVenueMatchesCount(1);
+-------------------------+
| GetVenueMatchesCount(1) |
+-------------------------+
|                      29 |
+-------------------------+
1 row in set (0.02 sec)
```

Q: What is the average runs scored per match by a given player?

A: To calculate the average runs per match, we declared three variable, namely, total_ runs, total_ matches & avg_ runs. The sum of batsman runs was stored in total_ runs. The sum of matches was stored in total_ matches. The two were divided to retrieve the average runs.

```
DELIMITER //
CREATE FUNCTION CalculateAverageRuns(batsman_id INT) RETURNS DECIMAL(10, 2)
BEGIN
DECLARE total_runs DECIMAL(10, 2);
DECLARE total_matches INT;
DECLARE avg_runs DECIMAL(10, 2);
SELECT SUM(BatsmanRuns) INTO total_runs
FROM deliveries
WHERE BatsmanID = batsman_id;
```

```
SELECT COUNT(DISTINCT MatchID) INTO total_matches
FROM deliveries
WHERE BatsmanID = batsman_id;
IF total_matches > 0 THEN
SET avg_runs = total_runs / total_matches;
ELSE
SET avg_runs = 0;
END IF;
RETURN avg_runs;
END //
DELIMITER;
```



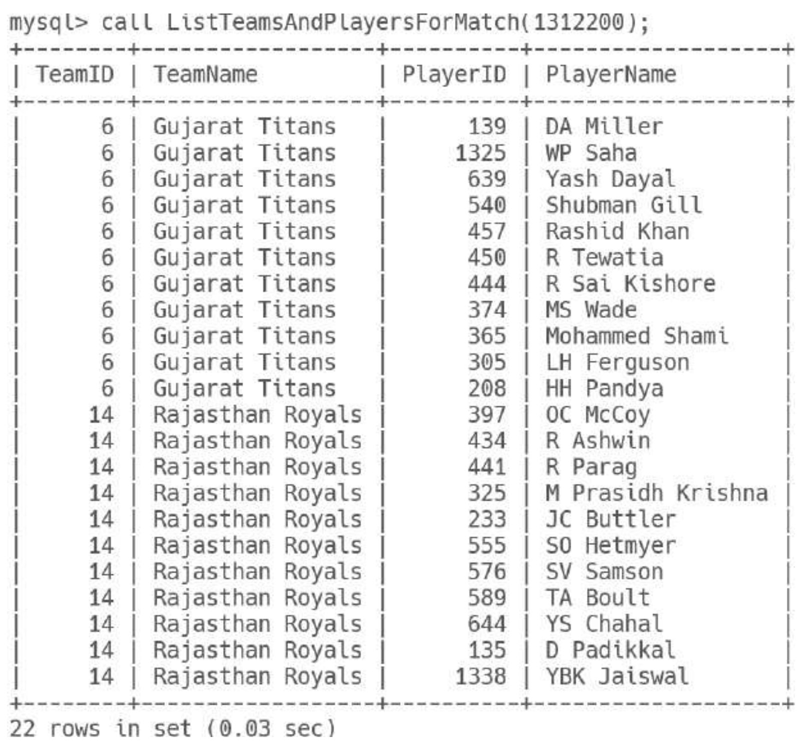
```
mysql> select CalculateAverageRuns(1305);
+----------------------------+
| CalculateAverageRuns(1305) |
+----------------------------+
|                      30.86 |
+----------------------------+
1 row in set (0.06 sec)
```

Database Procedures

Q: What was the team of all the players playing in a given match?

A: Since players can be traded mid-season, deriving the player line up for each match is essential. The table transaction table playermatch records the team of each player for each match. Thus, a procedure was created to retrieve the list of players and their teams for a given MatchID.

```sql
DELIMITER //
CREATE PROCEDURE ListTeamsAndPlayersForMatch(IN p_MatchID INT)
BEGIN
SELECT DISTINCT tm.TeamID,
 tm.TeamName,
 pm.PlayerID,
 pm.PlayerName
FROM teamsmaster tm
 INNER JOIN playermatch pmatch ON tm.TeamID = pmatch.TeamID
 INNER JOIN playermaster pm ON pmatch.PlayerID = pm.PlayerID
WHERE pmatch.MatchID = p_MatchID
ORDER BY tm.TeamName;
END //
DELIMITER;
```

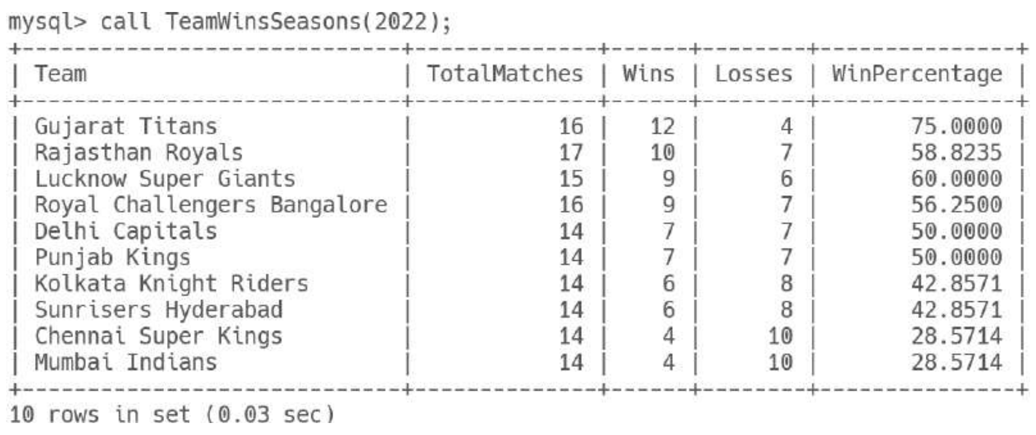


```
mysql> call ListTeamsAndPlayersForMatch(1312200);
+--------+-------------------+----------+-------------------+
| TeamID | TeamName          | PlayerID | PlayerName        |
+--------+-------------------+----------+-------------------+
|      6 | Gujarat Titans    |      139 | DA Miller         |
|      6 | Gujarat Titans    |     1325 | WP Saha           |
|      6 | Gujarat Titans    |      639 | Yash Dayal        |
|      6 | Gujarat Titans    |      540 | Shubman Gill      |
|      6 | Gujarat Titans    |      457 | Rashid Khan       |
|      6 | Gujarat Titans    |      450 | R Tewatia         |
|      6 | Gujarat Titans    |      444 | R Sai Kishore     |
|      6 | Gujarat Titans    |      374 | MS Wade           |
|      6 | Gujarat Titans    |      365 | Mohammed Shami    |
|      6 | Gujarat Titans    |      305 | LH Ferguson       |
|      6 | Gujarat Titans    |      208 | HH Pandya         |
|     14 | Rajasthan Royals  |      397 | OC McCoy          |
|     14 | Rajasthan Royals  |      434 | R Ashwin          |
|     14 | Rajasthan Royals  |      441 | R Parag           |
|     14 | Rajasthan Royals  |      325 | M Prasidh Krishna |
|     14 | Rajasthan Royals  |      233 | JC Buttler        |
|     14 | Rajasthan Royals  |      555 | SO Hetmyer        |
|     14 | Rajasthan Royals  |      576 | SV Samson         |
|     14 | Rajasthan Royals  |      589 | TA Boult          |
|     14 | Rajasthan Royals  |      644 | YS Chahal         |
|     14 | Rajasthan Royals  |      135 | D Padikkal        |
|     14 | Rajasthan Royals  |     1338 | YBK Jaiswal       |
+--------+-------------------+----------+-------------------+
22 rows in set (0.03 sec)
```

Q: What was the win percentage of all teams for a given season?

A: To calculate the wins percentage for all the teams in a given season, we created another procedure. This procedure uses the WinnerID field to identify which team won a given match. The procedure uses the sum case method to identify the wins as well as the losses. It then groups them by the TeamID feild in the teamsmaster table. The 'Having MatchesPlayed > 0' at line 55 is used to elimiate all those teams which did not play in the given season.

```sql
Delimiter //
Create Procedure TeamWinsSeasons(IN Season_Year INT)
BEGIN
Select t.TeamName AS Team,
 Count(*) AS TotalMatches,
 Sum(
  CASE
   WHEN m.WinnerID = t.TeamID THEN 1
   ELSE 0
  END
 ) AS Wins,
 Count(*) - Sum(
  Case
   When m.WinnerID = t.TeamID THEN 1
   ELSE 0
  END
 ) AS Losses,
 (
  Sum(
   CASE
    WHEN m.WinnerID = t.TeamID THEN 1
    ELSE 0
   END
  ) / Count(*)
 ) * 100 AS WinPercentage
From matches m
 INNER JOIN teamsmaster t on m.TeamID1 = t.TeamID
 OR m.TeamID2 = t.TeamID
Where m.SeasonYear = Season_Year
GROUP BY t.TeamName
ORDER BY Wins desc;
End //
Delimiter;
```

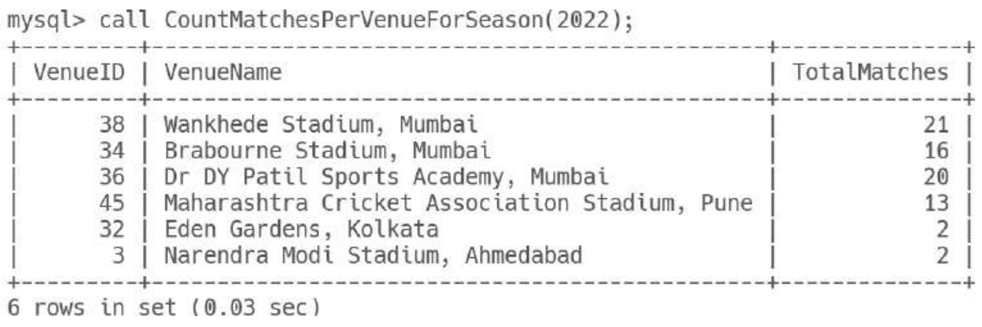


```
mysql> call TeamWinsSeasons(2022);
+-----------------------------+--------------+------+--------+---------------+
| Team                        | TotalMatches | Wins | Losses | WinPercentage |
+-----------------------------+--------------+------+--------+---------------+
| Gujarat Titans              |           16 |   12 |      4 |       75.0000 |
| Rajasthan Royals            |           17 |   10 |      7 |       58.8235 |
| Lucknow Super Giants        |           15 |    9 |      6 |       60.0000 |
| Royal Challengers Bangalore |           16 |    9 |      7 |       56.2500 |
| Delhi Capitals              |           14 |    7 |      7 |       50.0000 |
| Punjab Kings                |           14 |    7 |      7 |       50.0000 |
| Kolkata Knight Riders       |           14 |    6 |      8 |       42.8571 |
| Sunrisers Hyderabad         |           14 |    6 |      8 |       42.8571 |
| Chennai Super Kings         |           14 |    4 |     10 |       28.5714 |
| Mumbai Indians              |           14 |    4 |     10 |       28.5714 |
+-----------------------------+--------------+------+--------+---------------+
10 rows in set (0.03 sec)
```

Q: How many matches were held at each venue for a given season?

A: Each year matches are held at various venues either across the country or sometimes even abroad. The procedure simply counts the number of match using the `Count()` function and groups them according to the VenueName.

```
DELIMITER //
CREATE PROCEDURE CountMatchesPerVenueForSeason (IN season_year INT)
BEGIN
SELECT v.VenueID,
 v.VenueName,
 COUNT(m.MatchID) AS TotalMatches
FROM venuesmaster v
 LEFT JOIN matches m ON v.VenueID = m.VenueID
WHERE m.SeasonYear = season_year
GROUP BY v.VenueID,
 v.VenueName;
END //
DELIMITER;
```

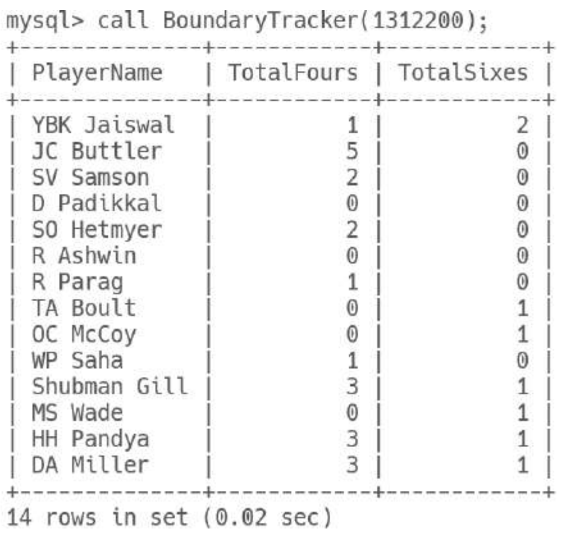


```
mysql> call CountMatchesPerVenueForSeason(2022);
+---------+-----------------------------------------------+--------------+
| VenueID | VenueName                                     | TotalMatches |
+---------+-----------------------------------------------+--------------+
|      38 | Wankhede Stadium, Mumbai                      |           21 |
|      34 | Brabourne Stadium, Mumbai                     |           16 |
|      36 | Dr DY Patil Sports Academy, Mumbai            |           20 |
|      45 | Maharashtra Cricket Association Stadium, Pune |           13 |
|      32 | Eden Gardens, Kolkata                         |            2 |
|       3 | Narendra Modi Stadium, Ahmedabad              |            2 |
+---------+-----------------------------------------------+--------------+
6 rows in set (0.03 sec)
```

Q: What were the total number of 4's & 6's by all the players in a given match?

A: This procedure also utilizes the sum case method, to sum the number of fours and sixes by all the players in a given season.

```
DELIMITER //
CREATE PROCEDURE BoundaryTracker(IN match_id INT)
BEGIN
SELECT pm.PlayerName,
 COUNT(
  CASE
   WHEN d.BatsmanRuns = 4 THEN 1
  END
 ) AS TotalFours,
 COUNT(
  CASE
   WHEN d.BatsmanRuns = 6 THEN 1
  END
 ) AS TotalSixes
FROM deliveries d
 JOIN playermaster pm ON d.BatsmanID = pm.PlayerID
WHERE d.MatchID = match_id
GROUP BY pm.PlayerName;
END //
DELIMITER;
```

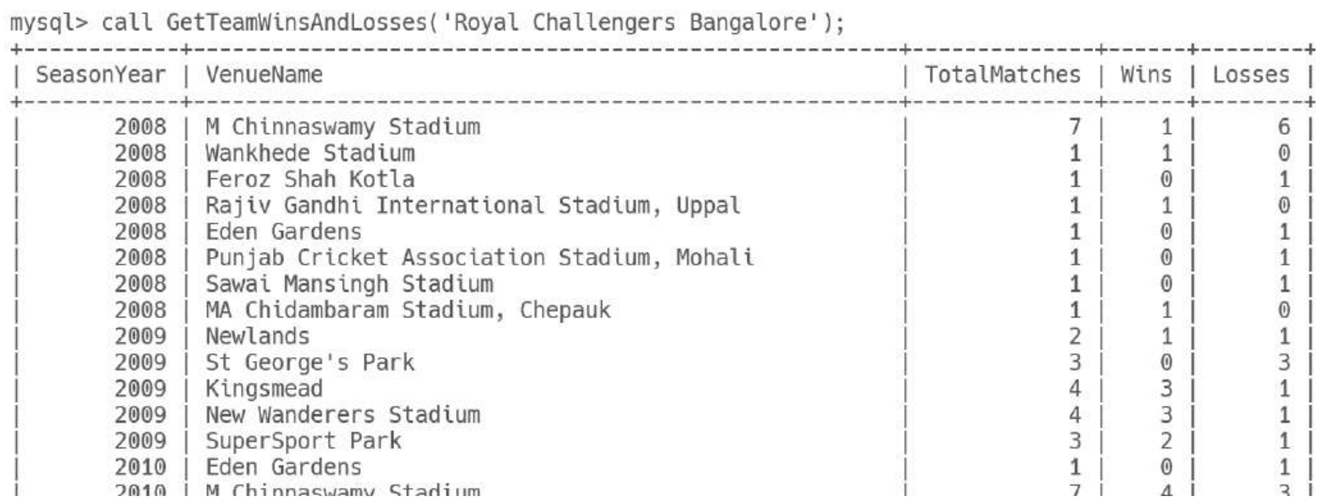


```
mysql> call BoundaryTracker(1312200);
+--------------+------------+------------+
| PlayerName   | TotalFours | TotalSixes |
+--------------+------------+------------+
| YBK Jaiswal  |          1 |          2 |
| JC Buttler   |          5 |          0 |
| SV Samson    |          2 |          0 |
| D Padikkal   |          0 |          0 |
| SO Hetmyer   |          2 |          0 |
| R Ashwin     |          0 |          0 |
| R Parag      |          1 |          0 |
| TA Boult     |          0 |          1 |
| OC McCoy     |          0 |          1 |
| WP Saha      |          1 |          0 |
| Shubman Gill |          3 |          1 |
| MS Wade      |          0 |          1 |
| HH Pandya    |          3 |          1 |
| DA Miller    |          3 |          1 |
+--------------+------------+------------+
14 rows in set (0.02 sec)
```

Q: What were the total wins and losses by a given team for all the seasons at all the venues?

A: To retrieve the records of total matches, wins & losses for a given team, a procedure was created. The objective of this procedure was to help the stakeholders understand which venue is the most scoring for a given team. The procedure uses the sum case method to count the number of wins and losses.

```
DELIMITER //
CREATE PROCEDURE GetTeamWinsAndLosses(IN team_name VARCHAR(255))
BEGIN
SELECT m.SeasonYear,
 v.VenueName,
 COUNT(*) AS TotalMatches,
 SUM(
  CASE
   WHEN m.WinnerID = t.TeamID THEN 1
   ELSE 0
  END
 ) AS Wins,
 COUNT(*) - SUM(
  CASE
   WHEN m.WinnerID = t.TeamID THEN 1
   ELSE 0
  END
 ) AS Losses
FROM matches m
 JOIN teamsmaster t ON m.TeamID1 = t.TeamID
 OR m.TeamID2 = t.TeamID
 JOIN venuesmaster v ON m.VenueID = v.VenueID
WHERE t.TeamName = team_name
GROUP BY m.SeasonYear,
 v.VenueName;
END //
DELIMITER;
```

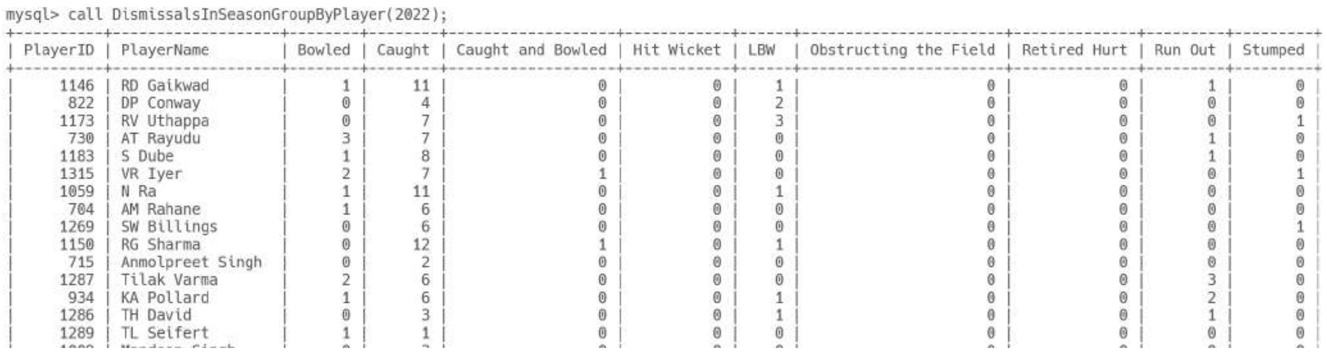


```
mysql> call GetTeamWinsAndLosses('Royal Challengers Bangalore');
+------------+------------------------------------------------------+--------------+------+--------+
| SeasonYear | VenueName                                            | TotalMatches | Wins | Losses |
+------------+------------------------------------------------------+--------------+------+--------+
|       2008 | M Chinnaswamy Stadium                                |            7 |    1 |      6 |
|       2008 | Wankhede Stadium                                     |            1 |    1 |      0 |
|       2008 | Feroz Shah Kotla                                     |            1 |    0 |      1 |
|       2008 | Rajiv Gandhi International Stadium, Uppal            |            1 |    1 |      0 |
|       2008 | Eden Gardens                                         |            1 |    0 |      1 |
|       2008 | Punjab Cricket Association Stadium, Mohali           |            1 |    0 |      1 |
|       2008 | Sawai Mansingh Stadium                               |            1 |    0 |      1 |
|       2008 | MA Chidambaram Stadium, Chepauk                      |            1 |    1 |      0 |
|       2009 | Newlands                                             |            2 |    1 |      1 |
|       2009 | St George's Park                                     |            3 |    0 |      3 |
|       2009 | Kingsmead                                            |            4 |    3 |      1 |
|       2009 | New Wanderers Stadium                                |            4 |    3 |      1 |
|       2009 | SuperSport Park                                      |            3 |    2 |      1 |
|       2010 | Eden Gardens                                         |            1 |    0 |      1 |
|       2010 | M Chinnaswamy Stadium                                |            7 |    4 |      3 |
```

Q: What was the count of all types of dismissals by every bowler in a given season?

A: To assess the player performance by each kind of dismissal they had, a procedure was created. The procedure simpy counts all the instance where a particular type of dismissal occurred from the DismissalType field. The output was grouped by PlayerName.

```sql
DELIMITER //
CREATE PROCEDURE DismissalsInSeasonGroupByPlayer (IN p_SeasonYear YEAR)
BEGIN
SELECT pm.PlayerID,
 pm.PlayerName,
 SUM(
  CASE
   WHEN d.DismissalType = 'bowled' THEN 1
   ELSE 0
  END
 ) AS Bowled,
 SUM(
  CASE
   WHEN d.DismissalType = 'caught' THEN 1
   ELSE 0
  END
 ) AS Caught,
 SUM(
  CASE
   WHEN d.DismissalType = 'caught and bowled' THEN 1
   ELSE 0
  END
 ) AS `Caught and Bowled`,
 SUM(
  CASE
   WHEN d.DismissalType = 'hit wicket' THEN 1
   ELSE 0
  END
 ) AS `Hit Wicket`,
 SUM(
  CASE
   WHEN d.DismissalType = 'lbw' THEN 1
   ELSE 0
  END
 ) AS LBW,
 SUM(
  CASE
   WHEN d.DismissalType = 'obstructing the field' THEN 1
   ELSE 0
  END
 ) AS `Obstructing the Field`,
 SUM(
  CASE
   WHEN d.DismissalType = 'retired hurt' THEN 1
   ELSE 0
  END
 ) AS `Retired Hurt`,
 SUM(
  CASE
   WHEN d.DismissalType = 'run out' THEN 1
   ELSE 0
  END
 ) AS `Run Out`,
 SUM(
  CASE
   WHEN d.DismissalType = 'stumped' THEN 1
   ELSE 0
  END
 ) AS Stumped
FROM deliveries d
 INNER JOIN matches m ON d.MatchID = m.MatchID
 INNER JOIN playermaster pm ON d.PlayerDismissedID = pm.PlayerID
WHERE m.SeasonYear = p_SeasonYear
GROUP BY pm.PlayerID,
 pm.PlayerName;
END //
DELIMITER;
```

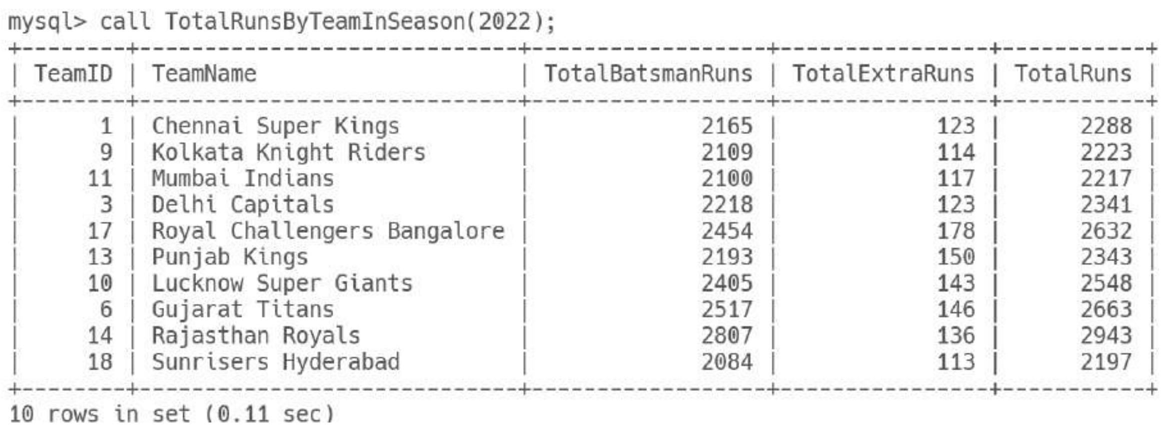


```
mysql> call DismissalsInSeasonGroupByPlayer(2022);
+----------+------------------+--------+--------+--------------------+------------+-----+-----------------------+--------------+---------+---------+
| PlayerID | PlayerName       | Bowled | Caught | Caught and Bowled  | Hit Wicket | LBW | Obstructing the Field | Retired Hurt | Run Out | Stumped |
+----------+------------------+--------+--------+--------------------+------------+-----+-----------------------+--------------+---------+---------+
|     1146 | RD Gaikwad       |      1 |     11 |                  0 |          0 |   1 |                     0 |            0 |       1 |       0 |
|      822 | DP Conway        |      0 |      4 |                  0 |          0 |   2 |                     0 |            0 |       0 |       0 |
|     1173 | RV Uthappa       |      0 |      7 |                  0 |          0 |   3 |                     0 |            0 |       0 |       1 |
|      730 | AT Rayudu        |      3 |      7 |                  0 |          0 |   0 |                     0 |            0 |       1 |       0 |
|     1183 | S Dube           |      1 |      8 |                  0 |          0 |   0 |                     0 |            0 |       1 |       0 |
|     1315 | VR Iyer          |      2 |      7 |                  1 |          0 |   0 |                     0 |            0 |       0 |       1 |
|     1059 | N Ra             |      1 |     11 |                  0 |          0 |   1 |                     0 |            0 |       0 |       0 |
|      704 | AM Rahane        |      1 |      6 |                  0 |          0 |   0 |                     0 |            0 |       0 |       0 |
|     1269 | SW Billings      |      0 |      6 |                  0 |          0 |   0 |                     0 |            0 |       0 |       1 |
|     1150 | RG Sharma        |      0 |     12 |                  1 |          0 |   1 |                     0 |            0 |       0 |       0 |
|      715 | Anmolpreet Singh |      0 |      2 |                  0 |          0 |   0 |                     0 |            0 |       0 |       0 |
|     1287 | Tilak Varma      |      2 |      6 |                  0 |          0 |   0 |                     0 |            0 |       3 |       0 |
|      934 | KA Pollard       |      1 |      6 |                  0 |          0 |   1 |                     0 |            0 |       2 |       0 |
|     1286 | TH David         |      0 |      3 |                  0 |          0 |   1 |                     0 |            0 |       1 |       0 |
|     1289 | TL Seifert       |      1 |      1 |                  0 |          0 |   0 |                     0 |            0 |       0 |       0 |
```

Q: What were the total runs scored by all the teams in a given season?

A: Another metric to assess team performance is the runs scored by them. A procedure was created to sum these runs by the teams and retrieve the relevant records. Breakup of total runs was provided as Batsman runs and Extra Runs.

```sql
DELIMITER //
CREATE PROCEDURE TotalRunsByTeamInSeason (IN p_SeasonYear YEAR)
BEGIN
SELECT tm.TeamID,
 tm.TeamName,
 SUM(d.BatsmanRuns) AS TotalBatsmanRuns,
 SUM(d.ExtraRuns) AS TotalExtraRuns,
 SUM(d.TotalRuns) AS TotalRuns
FROM deliveries d
 INNER JOIN matches m ON d.MatchID = m.MatchID
 INNER JOIN teamsmaster tm ON d.BattingTeamID = tm.TeamID
WHERE m.SeasonYear = p_SeasonYear
GROUP BY tm.TeamID,
 tm.TeamName;
END //
DELIMITER;
```



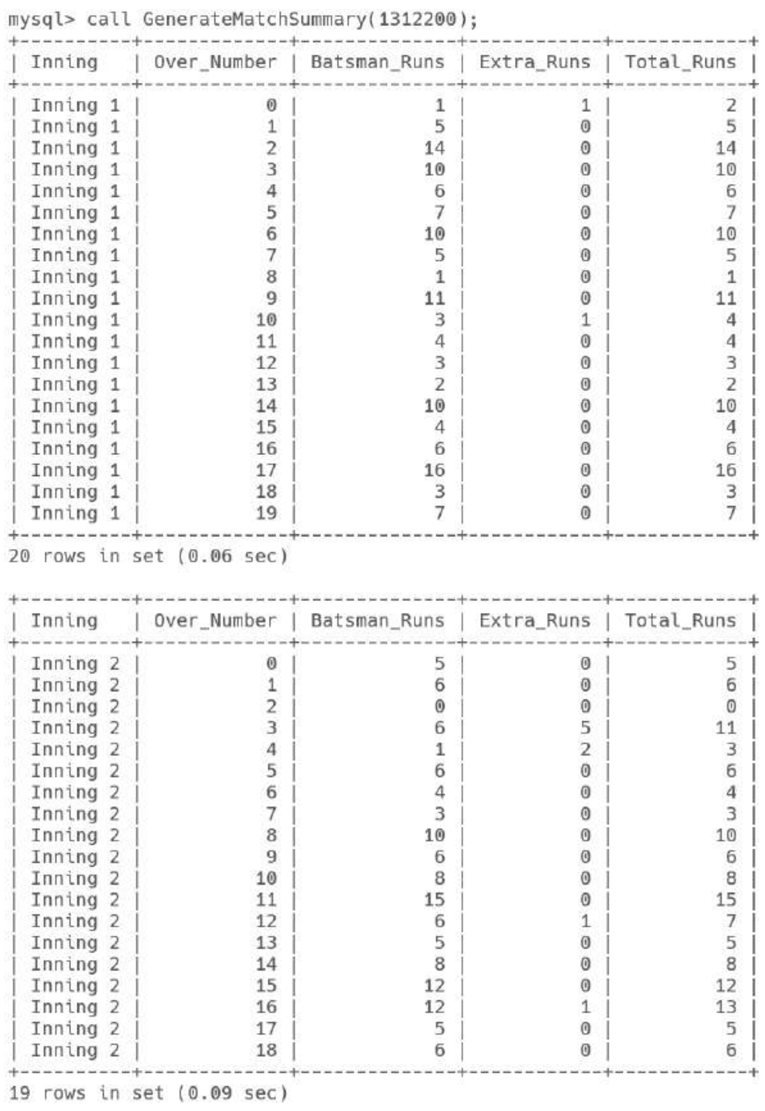
```
mysql> call TotalRunsByTeamInSeason(2022);
+--------+-----------------------------+------------------+----------------+-----------+
| TeamID | TeamName                    | TotalBatsmanRuns | TotalExtraRuns | TotalRuns |
+--------+-----------------------------+------------------+----------------+-----------+
|      1 | Chennai Super Kings         |             2165 |            123 |      2288 |
|      9 | Kolkata Knight Riders       |             2109 |            114 |      2223 |
|     11 | Mumbai Indians              |             2100 |            117 |      2217 |
|      3 | Delhi Capitals              |             2218 |            123 |      2341 |
|     17 | Royal Challengers Bangalore |             2454 |            178 |      2632 |
|     13 | Punjab Kings                |             2193 |            150 |      2343 |
|     10 | Lucknow Super Giants        |             2405 |            143 |      2548 |
|      6 | Gujarat Titans              |             2517 |            146 |      2663 |
|     14 | Rajasthan Royals            |             2807 |            136 |      2943 |
|     18 | Sunrisers Hyderabad         |             2084 |            113 |      2197 |
+--------+-----------------------------+------------------+----------------+-----------+
10 rows in set (0.11 sec)
```

Q: What is the over wise summary for a given match?

A: The over wise summary for a given match is important information for a given match. The procedure has two select statements to generate to separate tables for both the innings. The query outputs the over wise sum of BatsmanRuns, ExtraRuns & TotalRuns.

```
DELIMITER //
CREATE PROCEDURE GenerateMatchSummary (IN match_id INT)
BEGIN
DECLARE inning1_id,
 inning2_id INT;
DECLARE max_overs INT;
-- Get the inning IDs for the match
SELECT TeamID1,
 TeamID2 INTO inning1_id,
 inning2_id
FROM matches
WHERE MatchID = match_id;
-- Get the maximum number of overs for the match
SELECT MAX(OverNumber) INTO max_overs
FROM deliveries
WHERE MatchID = match_id;
-- Summary for Inning 1
SELECT 'Inning 1' AS Inning,
 OverNumber AS Over_Number,
 SUM(BatsmanRuns) AS Batsman_Runs,
 SUM(ExtraRuns) AS Extra_Runs,
 SUM(TotalRuns) AS Total_Runs
FROM deliveries
WHERE MatchID = match_id
```

```sql
 AND Innings = 1
GROUP BY OverNumber
HAVING OverNumber <= max_overs;
-- Summary for Inning 2
SELECT 'Inning 2' AS Inning,
 OverNumber AS Over_Number,
 SUM(BatsmanRuns) AS Batsman_Runs,
 SUM(ExtraRuns) AS Extra_Runs,
 SUM(TotalRuns) AS Total_Runs
FROM deliveries
WHERE MatchID = match_id
 AND Innings = 2
GROUP BY OverNumber
HAVING OverNumber <= max_overs;
END //
DELIMITER;
```



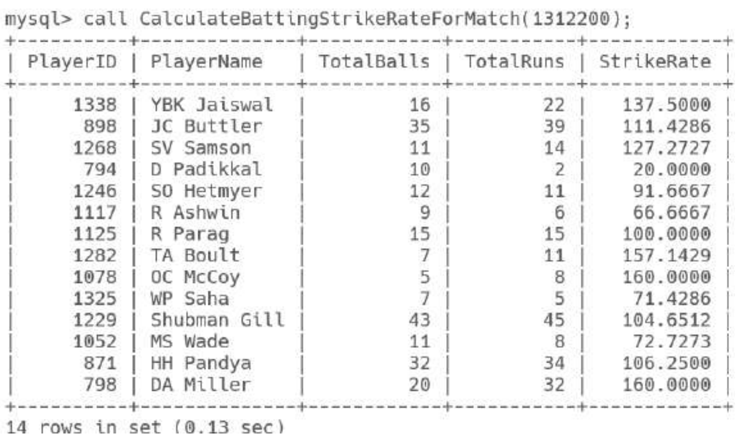
```
mysql> call GenerateMatchSummary(1312200);
+----------+-------------+--------------+------------+------------+
| Inning   | Over_Number | Batsman_Runs | Extra_Runs | Total_Runs |
+----------+-------------+--------------+------------+------------+
| Inning 1 |           0 |            1 |          1 |          2 |
| Inning 1 |           1 |            5 |          0 |          5 |
| Inning 1 |           2 |           14 |          0 |         14 |
| Inning 1 |           3 |           10 |          0 |         10 |
| Inning 1 |           4 |            6 |          0 |          6 |
| Inning 1 |           5 |            7 |          0 |          7 |
| Inning 1 |           6 |           10 |          0 |         10 |
| Inning 1 |           7 |            5 |          0 |          5 |
| Inning 1 |           8 |            1 |          0 |          1 |
| Inning 1 |           9 |           11 |          0 |         11 |
| Inning 1 |          10 |            3 |          1 |          4 |
| Inning 1 |          11 |            4 |          0 |          4 |
| Inning 1 |          12 |            3 |          0 |          3 |
| Inning 1 |          13 |            2 |          0 |          2 |
| Inning 1 |          14 |           10 |          0 |         10 |
| Inning 1 |          15 |            4 |          0 |          4 |
| Inning 1 |          16 |            6 |          0 |          6 |
| Inning 1 |          17 |           16 |          0 |         16 |
| Inning 1 |          18 |            3 |          0 |          3 |
| Inning 1 |          19 |            7 |          0 |          7 |
+----------+-------------+--------------+------------+------------+
20 rows in set (0.06 sec)

+----------+-------------+--------------+------------+------------+
| Inning   | Over_Number | Batsman_Runs | Extra_Runs | Total_Runs |
+----------+-------------+--------------+------------+------------+
| Inning 2 |           0 |            5 |          0 |          5 |
| Inning 2 |           1 |            6 |          0 |          6 |
| Inning 2 |           2 |            0 |          0 |          0 |
| Inning 2 |           3 |            6 |          5 |         11 |
| Inning 2 |           4 |            1 |          2 |          3 |
| Inning 2 |           5 |            6 |          0 |          6 |
| Inning 2 |           6 |            4 |          0 |          4 |
| Inning 2 |           7 |            3 |          0 |          3 |
| Inning 2 |           8 |           10 |          0 |         10 |
| Inning 2 |           9 |            6 |          0 |          6 |
| Inning 2 |          10 |            8 |          0 |          8 |
| Inning 2 |          11 |           15 |          0 |         15 |
| Inning 2 |          12 |            6 |          1 |          7 |
| Inning 2 |          13 |            5 |          0 |          5 |
| Inning 2 |          14 |            8 |          0 |          8 |
| Inning 2 |          15 |           12 |          0 |         12 |
| Inning 2 |          16 |           12 |          1 |         13 |
| Inning 2 |          17 |            5 |          0 |          5 |
| Inning 2 |          18 |            6 |          0 |          6 |
+----------+-------------+--------------+------------+------------+
19 rows in set (0.09 sec)
```

Q: What was the strike rate for all the players playing in given match?

A: The batting strike rate is an essential metric to assessing player performance. A procedure was created to assess the player strike rates. The procedure counts the total balls faced using the `Count()` function. The Sum of Batsman runs is done, and later divided by the TotalBalls faced to calculate the strike rate. The procedure calculates this for both the innings and groups them by PlayerID & PlayerName.

```sql
DELIMITER //
CREATE PROCEDURE CalculateBattingStrikeRateForMatch(IN match_id INT)
BEGIN
SELECT d.BatsmanID AS PlayerID,
 pm.PlayerName,
 COUNT(*) AS TotalBalls,
 SUM(d.BatsmanRuns) AS TotalRuns,
 (SUM(d.BatsmanRuns) / COUNT(*)) * 100 AS StrikeRate
FROM deliveries d
 INNER JOIN playermaster pm ON d.BatsmanID = pm.PlayerID
WHERE d.MatchID = match_id
 AND d.Innings IN (1, 2)
GROUP BY
 PlayerID,
 pm.PlayerName;
END //
DELIMITER;
```



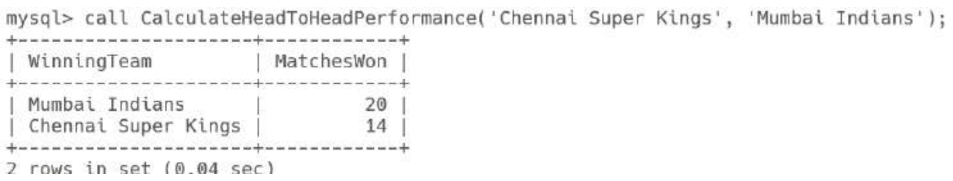
```
mysql> call CalculateBattingStrikeRateForMatch(1312200);
+----------+--------------+------------+-----------+------------+
| PlayerID | PlayerName   | TotalBalls | TotalRuns | StrikeRate |
+----------+--------------+------------+-----------+------------+
|     1338 | YBK Jaiswal  |         16 |        22 |   137.5000 |
|      898 | JC Buttler   |         35 |        39 |   111.4286 |
|     1268 | SV Samson    |         11 |        14 |   127.2727 |
|      794 | D Padikkal   |         10 |         2 |    20.0000 |
|     1246 | SO Hetmyer   |         12 |        11 |    91.6667 |
|     1117 | R Ashwin     |          9 |         6 |    66.6667 |
|     1125 | R Parag      |         15 |        15 |   100.0000 |
|     1282 | TA Boult     |          7 |        11 |   157.1429 |
|     1078 | OC McCoy     |          5 |         8 |   160.0000 |
|     1325 | WP Saha      |          7 |         5 |    71.4286 |
|     1229 | Shubman Gill |         43 |        45 |   104.6512 |
|     1052 | MS Wade      |         11 |         8 |    72.7273 |
|      871 | HH Pandya    |         32 |        34 |   106.2500 |
|      798 | DA Miller    |         20 |        32 |   160.0000 |
+----------+--------------+------------+-----------+------------+
14 rows in set (0.13 sec)
```

Q: What were the head to head wins against two given teams for all seasons?

A: To calculate the head to head wins, first two variables team1_id and team2_id were declared. These will be important as they are used to reference the teams at the later stage. The declared variables are given values using the `select into` command, using a where clause to identify the TeamName of both the teams. The `select case` command is used to select instances wherein the declared TeamID's match the WinnerID of all the matches. The where clause in this select statment ensures that only those instances are selected whrein both teams indeed played.

```sql
DELIMITER //
CREATE PROCEDURE CalculateHeadToHeadPerformance(
 IN team1_name VARCHAR(255),
 IN team2_name VARCHAR(255)
)
BEGIN
DECLARE team1_id INT;
DECLARE team2_id INT;
SELECT TeamID INTO team1_id
FROM teamsmaster
WHERE TeamName = team1_name;
SELECT TeamID INTO team2_id
```

```
FROM teamsmaster
WHERE TeamName = team2_name;
SELECT CASE
  WHEN TeamID1 = team1_id
  AND WinnerID = team1_id THEN team1_name
  WHEN TeamID2 = team1_id
  AND WinnerID = team1_id THEN team1_name
  ELSE team2_name
 END AS WinningTeam,
 COUNT(*) AS MatchesWon
FROM matches
WHERE (
  TeamID1 = team1_id
  AND TeamID2 = team2_id
 )
 OR (
  TeamID1 = team2_id
  AND TeamID2 = team1_id
 )
 AND WinnerID IS NOT NULL
GROUP BY WinningTeam;
END //
DELIMITER;
```



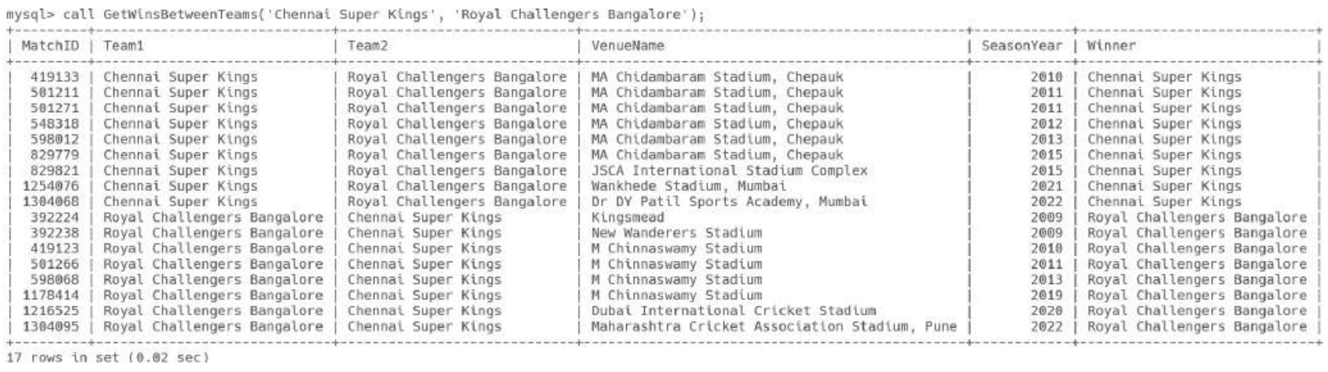
```
mysql> call CalculateHeadToHeadPerformance('Chennai Super Kings', 'Mumbai Indians');
+---------------------+------------+
| WinningTeam         | MatchesWon |
+---------------------+------------+
| Mumbai Indians      |         20 |
| Chennai Super Kings |         14 |
+---------------------+------------+
2 rows in set (0.04 sec)
```

Q: What were the detials of the head to head to wins between two wins?

A: To draw an effective comparision betweem teams, head to head wins were counted using a procedure. We went further to get the details of those matches for their head to head wins and losses. It using a similar logic to the previous procedure, however it also selects the VenueName and SeasonYear to see which venues may be favourable to either team.

```
DELIMITER //
CREATE PROCEDURE GetWinsBetweenTeams(
 IN team1_name VARCHAR(255),
 IN team2_name VARCHAR(255)
)
BEGIN
DECLARE team1_id,
 team2_id INT;
SELECT TeamID INTO team1_id
FROM teamsmaster
WHERE TeamName = team1_name;
SELECT TeamID INTO team2_id
FROM teamsmaster
WHERE TeamName = team2_name;
SELECT m.MatchID,
 t1.TeamName AS Team1,
 t2.TeamName AS Team2,
 v.VenueName,
```

```sql
 m.SeasonYear,
 CASE
  WHEN m.WinnerID = team1_id THEN team1_name
  WHEN m.WinnerID = team2_id THEN team2_name
  ELSE 'Draw' -- You may handle draws accordingly
 END AS Winner
FROM matches m
 INNER JOIN teamsmaster t1 ON m.TeamID1 = t1.TeamID
 INNER JOIN teamsmaster t2 ON m.TeamID2 = t2.TeamID
 INNER JOIN venuesmaster v ON m.VenueID = v.VenueID
WHERE (
  m.TeamID1 = team1_id
  AND m.TeamID2 = team2_id
  AND m.WinnerID = team1_id
 )
 OR (
  m.TeamID1 = team2_id
  AND m.TeamID2 = team1_id
  AND m.WinnerID = team2_id
 );
END //
DELIMITER;
```

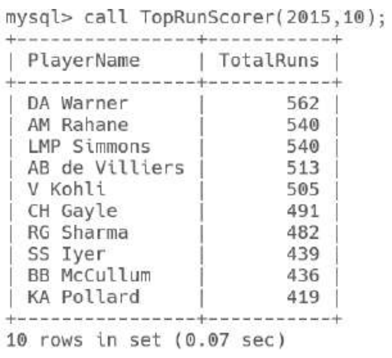


```
mysql> call GetWinsBetweenTeams('Chennai Super Kings', 'Royal Challengers Bangalore');
+---------+-----------------------------+-----------------------------+-----------------------------------------------+------------+-----------------------------+
| MatchID | Team1                       | Team2                       | VenueName                                     | SeasonYear | Winner                      |
+---------+-----------------------------+-----------------------------+-----------------------------------------------+------------+-----------------------------+
|  419133 | Chennai Super Kings         | Royal Challengers Bangalore | MA Chidambaram Stadium, Chepauk               |       2010 | Chennai Super Kings         |
|  501211 | Chennai Super Kings         | Royal Challengers Bangalore | MA Chidambaram Stadium, Chepauk               |       2011 | Chennai Super Kings         |
|  501271 | Chennai Super Kings         | Royal Challengers Bangalore | MA Chidambaram Stadium, Chepauk               |       2011 | Chennai Super Kings         |
|  548318 | Chennai Super Kings         | Royal Challengers Bangalore | MA Chidambaram Stadium, Chepauk               |       2012 | Chennai Super Kings         |
|  598012 | Chennai Super Kings         | Royal Challengers Bangalore | MA Chidambaram Stadium, Chepauk               |       2013 | Chennai Super Kings         |
|  829779 | Chennai Super Kings         | Royal Challengers Bangalore | MA Chidambaram Stadium, Chepauk               |       2015 | Chennai Super Kings         |
|  829821 | Chennai Super Kings         | Royal Challengers Bangalore | JSCA International Stadium Complex            |       2015 | Chennai Super Kings         |
| 1254076 | Chennai Super Kings         | Royal Challengers Bangalore | Wankhede Stadium, Mumbai                      |       2021 | Chennai Super Kings         |
| 1304068 | Chennai Super Kings         | Royal Challengers Bangalore | Dr DY Patil Sports Academy, Mumbai            |       2022 | Chennai Super Kings         |
|  392224 | Royal Challengers Bangalore | Chennai Super Kings         | Kingsmead                                     |       2009 | Royal Challengers Bangalore |
|  392238 | Royal Challengers Bangalore | Chennai Super Kings         | New Wanderers Stadium                         |       2009 | Royal Challengers Bangalore |
|  419123 | Royal Challengers Bangalore | Chennai Super Kings         | M Chinnaswamy Stadium                         |       2010 | Royal Challengers Bangalore |
|  501266 | Royal Challengers Bangalore | Chennai Super Kings         | M Chinnaswamy Stadium                         |       2011 | Royal Challengers Bangalore |
|  598068 | Royal Challengers Bangalore | Chennai Super Kings         | M Chinnaswamy Stadium                         |       2013 | Royal Challengers Bangalore |
| 1178414 | Royal Challengers Bangalore | Chennai Super Kings         | M Chinnaswamy Stadium                         |       2019 | Royal Challengers Bangalore |
| 1216525 | Royal Challengers Bangalore | Chennai Super Kings         | Dubai International Cricket Stadium           |       2020 | Royal Challengers Bangalore |
| 1304095 | Royal Challengers Bangalore | Chennai Super Kings         | Maharashtra Cricket Association Stadium, Pune |       2022 | Royal Challengers Bangalore |
+---------+-----------------------------+-----------------------------+-----------------------------------------------+------------+-----------------------------+
17 rows in set (0.02 sec)
```

Q: Who were the top runs scorers for a given season?

A: Top run scorer is an important metric for assessing player performance. The sum function simply sums the batsman runs scored by players. The output is grouped by PlayerName. The output is ordered by decending order of sum of BatsmanRuns. In order to retrieve only select records, the limit function is used.

```sql
DELIMITER //
CREATE PROCEDURE TopRunScorer (IN season_year INT, IN limit_count INT) BEGIN
SELECT pm.PlayerName,
 SUM(d.BatsmanRuns) AS TotalRuns
FROM deliveries d
 JOIN matches m ON d.MatchID = m.MatchID
 JOIN playermaster pm ON d.BatsmanID = pm.PlayerID
WHERE m.SeasonYear = season_year
GROUP BY d.BatsmanID
ORDER BY TotalRuns DESC
LIMIT limit_count;
END //
DELIMITER;
```

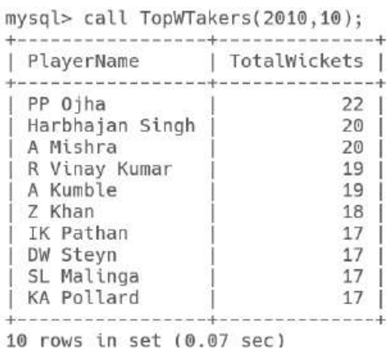


```
mysql> call TopRunScorer(2015,10);
+-----------------+-----------+
| PlayerName      | TotalRuns |
+-----------------+-----------+
| DA Warner       |       562 |
| AM Rahane       |       540 |
| LMP Simmons     |       540 |
| AB de Villiers  |       513 |
| V Kohli         |       505 |
| CH Gayle        |       491 |
| RG Sharma       |       482 |
| SS Iyer         |       439 |
| BB McCullum     |       436 |
| KA Pollard      |       419 |
+-----------------+-----------+
10 rows in set (0.07 sec)
```

Q: Who were the top wicket takers for a given season?

A: Top wicket is another important metric for assessing players. The procedure, uses a count function to count the number of dismissals that took place in a year. The output is grouped by players and ordered in decending order of wickets taken. The limit function is used to limit the records that are to be selected.

```sql
DELIMITER //
CREATE PROCEDURE TopWTakers (IN season_year INT, IN limit_count INT)
BEGIN
SELECT pm.PlayerName,
 COUNT(d.PlayerDismissedID) AS TotalWickets
FROM deliveries d
 JOIN matches m ON d.MatchID = m.MatchID
 JOIN playermaster pm ON d.BowlerID = pm.PlayerID
WHERE m.SeasonYear = season_year
 AND d.PlayerDismissedID IS NOT NULL
GROUP BY d.BowlerID
ORDER BY TotalWickets DESC
LIMIT limit_count;
END //
DELIMITER;
```



```
mysql> call TopWTakers(2010,10);
+-----------------+--------------+
| PlayerName      | TotalWickets |
+-----------------+--------------+
| PP Ojha         |           22 |
| Harbhajan Singh |           20 |
| A Mishra        |           20 |
| R Vinay Kumar   |           19 |
| A Kumble        |           19 |
| Z Khan          |           18 |
| IK Pathan       |           17 |
| DW Steyn        |           17 |
| SL Malinga      |           17 |
| KA Pollard      |           17 |
+-----------------+--------------+
10 rows in set (0.07 sec)
```

Database Triggers

Several triggers were created that would backup master data into a separate table when deleted. We created these triggers as master data should be kept intact and deletion of it shoulf be avoided at all cost to ensure referetial integrity in future.

playermaster_backup This trigger would backup playermaster data if it were ever deleted.

```
DELIMITER //
CREATE TRIGGER BackupPlayerDataBeforeDelete BEFORE DELETE ON playermaster
FOR EACH ROW
BEGIN
INSERT INTO playermaster_backup (PlayerID, PlayerName, DoB, Nationality)
VALUES (
  OLD.PlayerID,
  OLD.PlayerName,
  OLD.DoB,
  OLD.Nationality
 );
END //
DELIMITER;
```

BackupTeamDataBeforeDelete This trigger would backup team data before deletion.

```
DELIMITER //
CREATE TRIGGER BackupTeamDataBeforeDelete BEFORE DELETE ON teamsmaster
FOR EACH ROW
BEGIN
INSERT INTO teamsmaster_backup (TeamID, TeamName)
VALUES (OLD.TeamID, OLD.TeamName);
END //
DELIMITER;
```

BackupUmpireDataBeforeDelete This trigger would backup umpire data before deletion.

```
DELIMITER //
CREATE TRIGGER BackupUmpireDataBeforeDelete BEFORE DELETE ON umpiresmaster
FOR EACH ROW
BEGIN
INSERT INTO umpiresmaster_backup (UmpireID, UmpireName)
VALUES (OLD.UmpireID, OLD.UmpireName);
END //
DELIMITER;
```

BackupVenueDataBeforeDelete This trigger would backup venue data before deletion

```
DELIMITER //
CREATE TRIGGER BackupVenueDataBeforeDelete BEFORE DELETE ON venuesmaster
FOR EACH ROW
BEGIN
INSERT INTO venuesmaster_backup (VenueID, City, VenueName)
VALUES (OLD.VenueID, OLD.City, OLD.VenueName);
END //
DELIMITER;
```

Database Normalization

Most of the tables are in the third-standard form

Transaction Tables

deliveries

1. **First Normal Form (1NF):** In the Deliveries table follows 1NF as it possesses a primary key, "Ball ID"," and all columns contain atomic values. They cannot be broken further. The information conveyed does not depend on the row order, as it would have violated the first normal form.
2. **Second Normal Form(2NF):** The table follows the criteria of 2NF since there are no partial dependencies. All non-key columns are entirely reliant on the primary key.
3. **Third Normal Form (3NF):** The table does follow the criteria of 3NF since it lacks transitive dependencies. All non-key columns depend on the primary key, not other non-key ones.

matches

1. **First Normal Form (1NF):** The matches table follows 1NF as it possesses a primary key, "Match ID", and all columns contain atomic values. They cannot be broken further. The information conveyed does not depend on the row order, as it would have violated the first normal form.
2. **Second Normal Form(2NF):** The table follows the criteria of 2NF since there are no partial dependencies. All non-key columns are entirely reliant on the primary key.
3. **Third Normal Form (3NF):** The table does follow the criteria of 3NF since it lacks transitive dependencies. All non-key columns depend on the primary key, not other non-key ones.

playermatch

1. **First Normal Form (1NF):** The player match table follows 1NF as it possesses a composite primary key of "Match ID" and "Player ID" and all columns contain atomic values. They cannot be broken further. The information conveyed does not depend on the row order, as it would have violated the first normal form.
2. **Second Normal Form(2NF):** The table follows the criteria of 2NF since there are no partial dependencies. All non-key columns are entirely reliant on the composite key.
3. **Third Normal Form (3NF):** The table does follow the criteria of 3NF since it lacks transitive dependencies. All non-key columns depend on the primary key, not other non-key ones.

Master Tables

playermaster

1. **First Normal Form (1NF):** The players master table follows 1NF as it possesses a primary key of "PlayerID" and all columns contain atomic values. They cannot be broken further. The information conveyed does not depend on the row order, as it would have violated the first normal form.
2. **Second Normal Form(2NF):** The table follows the criteria of 2NF since there are no partial dependencies. All non-key columns are entirely reliant on the primary key.

3. **Third Normal Form (3NF):** The table does follow the criteria of 3NF since it lacks transitive dependencies. All non-key columns depend on the primary key, not other non-key ones.

teamsmaster

1. **First Normal Form (1NF):** The teams master table follows 1NF as it possesses a primary key of "TeamID" and all columns contain atomic values. They cannot be broken further. The information conveyed does not depend on the row order, as it would have violated the first normal form.
2. **Second Normal Form(2NF):** The table follows the criteria of 2NF since there are no partial dependencies. All non-key columns are entirely reliant on the primary key.
3. **Third Normal Form (3NF):** The table does follow the criteria of 3NF since it lacks transitive dependencies. All non-key columns depend on the primary key, not other non-key ones.

venuesmaster

1. **First Normal Form (1NF):** The Venues master table follows 1NF as it possesses a primary key of "Venue ID," and all columns contain atomic values. They cannot be broken further. The information conveyed does not depend on the row order, as it would have violated the first normal form.
2. **Second Normal Form(2NF):** The table follows the criteria of 2NF since there are no partial dependencies. All non-key columns are entirely reliant on the primary key.
3. **Third Normal Form (3NF):** The table does follow the criteria of 3NF since it lacks transitive dependencies. All non-key columns depend on the primary key, not other non-key ones.

umpiresmaster

1. **First Normal Form (1NF):** The Umpires master table follows 1NF as it possesses a primary key of "Umpire ID," and all columns contain atomic values. They cannot be broken further. The information conveyed does not depend on the row order, as it would have violated the first normal form.
2. **Second Normal Form(2NF):** The table follows the criteria of 2NF since there are no partial dependencies. All non-key columns are entirely reliant on the primary key.
3. **Third Normal Form (3NF):** The table does follow the criteria of 3NF since it lacks transitive dependencies. All non-key columns depend on the primary key, not other non-key ones.