DBMS Mini Project

CRICKET TOURNAMENT DATABASE SYSTEM

Submitted by:
Anuraag Mallinamadugu | PES1UG20CS067 | V Sem Section B

About the project

The IPL is among the richest tournaments in the world. The planning and scheduling of the various events in the 2-month period in which this tournament is held is a challenge. However, it is facilitated by sophisticated databases that store all the required information about every aspect of the tournament including player stats, team stats, umpire information, stadium information, staff information etc.

In this project, I am attempting to replicate the process of storing and modifying information about every aspect of a cricket tournament by creating a defined schema, populating the schema with values and applying various operations to the data. Finally, I also created an interactive UI using Streamlit to automatically make changes to the database in the backend when the user interacts with the front end.

The objective of this project is to create a database management system for IPL which can add and manipulate data about the tournament in a reliable fashion. The system should be able to compute and address various queries given by the user, some of which include join, aggregate, set queries, etc. To make the process of building the project easier, functions and stored procedures have been used to store commonly used pieces of code. SQL triggers have also been implemented to modify the behaviour of the system while executing certain operations.

Apart from all these, an interactive UI has also been developed using Streamlit which is capable of CRUD (Create, Read Update, Delete) operations.

Most local tournaments have smaller budgets and cannot afford sophisticated computer-based management systems. The scope of this project is to help in easing the process of scheduling and management in smaller tournaments by creating a low-cost database system.

ER Diagram

The database contains the following entities:

- Team
  The key attribute for this entity is the team's name which is of type varchar. It also has attributes such as city, number of losses, number of draws, number of wins, and rank. This entity also has a **multivalued attribute** called colours which represents the various colours on the team's jersey. I have assumed that a team may have multiple colours on the jersey.

- Player
  The key attribute for the player entity is the jersey number, which I have assumed is going to be unique for each player in a particular season. This value also cannot be NULL. Name is another attribute of type varchar. Two of the attributes, namely, DOB and debut are of date type. The attribute 'matches' is a **complex attribute** and consists of Test, ODI and T20 representing the number of matches played in the respective formats. There also exists a derived attribute 'age' which was derived from the DOB.

- Batsman
  This is a **weak entity type** which is connected via an **identifying relationship** from the player entity. This table basically holds all the batting stats of a player if he is a batsman, including runs, batting average, sixes, fifties, fours and hundreds. If there is a foreign key in this table pointing to an entity in the player table, it means that the corresponding player is a batsman.
- Bowler
  This is a **weak entity type** which is connected via an **identifying relationship** from the player entity. This table basically holds all the bowling stats of a player if he is a bowler, including balls bowled, bowling average, maidens, wickets and economy. If there is a foreign key in this table pointing to an entity in the player table, it means that the corresponding player is a bowler.

- Umpire
  This table holds information about all the umpires present in the tournament. The main attributes include 'name' of type varchar, 'Id' which is the primary key of type int, 'Number of matches' of type int, DOB of type date and 'Country of origin' of type varchar. There also exists a derived attribute 'age' which was derived from the DOB. The relationship connecting this entity to the rest of the entities also has an attribute called 'on/off field' telling if the umpire is an on-field umpire or not.

- Coach
  This table holds information about all the coaches present in each team in the tournament. The main attributes include ‘name’ of type varchar, ‘Id’ which is the primary key of type int, ‘Teams coached’ of type varchar, DOB of type date and ‘Country of origin’ of type varchar. There also exists a derived attribute ‘age’ which was derived from the DOB.

- Match
  This entity represents a match held between two teams and stores information about the same. It is identifiable by the match number which is its primary key. Other attributes include result, toss and man of the match of type varchar, and date of type ‘date’. The result attribute stores the name of the team which won the match. There is a relationship between this entity and the team entity to store the values of teams playing in this match.

- Stadium
  This entity stores information about all the stadiums in which the matches are played. Its attributes include name, address, city of type varchar, Id of type int, which is also the primary key, and capacity which is also of type int.

The database has the following relations:

- Bowls (1-1)
  Relationship between player and bowler. In this case, bowler is a weak entity and player is not. Hence, this relationship is an identifying relationship. If a player is indeed a bowler or an all-rounder, the details of his bowling stats will be stored in the bowler table. It is compulsory for the bowler to be a player but not vice versa. Hence, there is total participation on the side of the bowler entity and partial participation on the other side.

- Bats (1-1)
  Relationship between player and batsman. In this case, batsman is a weak entity and player is not. Hence, this relationship is an identifying relationship. If a player is indeed a batsman or an all-rounder, the details of his batting stats will be stored in the batsman table. It is compulsory for the batsman to be a player but not vice versa. Hence, there is total participation on the side of the batsman entity and partial participation on the other side.

- Plays for (N-1)
  This relationship exists between the entities: player and team. Multiple players can exist in a team but each player is allowed to exist in only one team. Hence, this is an N-1 relationship. Each team must have at least one player and each player must be in a team. Hence there is total participation on both sides.

- Captain (1-1)
  This relationship exists between the entities: player and team. A team has only one captain and only one of the players in the team can be a captain. Hence, this is a 1-1 relationship. Each team must have a captain but a player need not be a captain. Hence there is total participation from team and partial participation from player.

- Coached by (N-1)
  This relationship exists between the entities: coach and team. Multiple coaches can exist in a team but each coach is allowed to exist in only one team. Hence, this is an N-1 relationship. Every team must have at least one coach and each coach stored must be associated with a team. Hence, there is total participation from both entities.

- Rival (1-1)
  This is a recursive relationship and hence, it exists between two instances of the same entity type. It is used to relate two teams which are considered to be rivals in terms of competition. Some teams might not have rivals. Hence there is only partial participation on both sides. Since only each team has only one rival, this relationship has a cardinality ratio of 1:1.

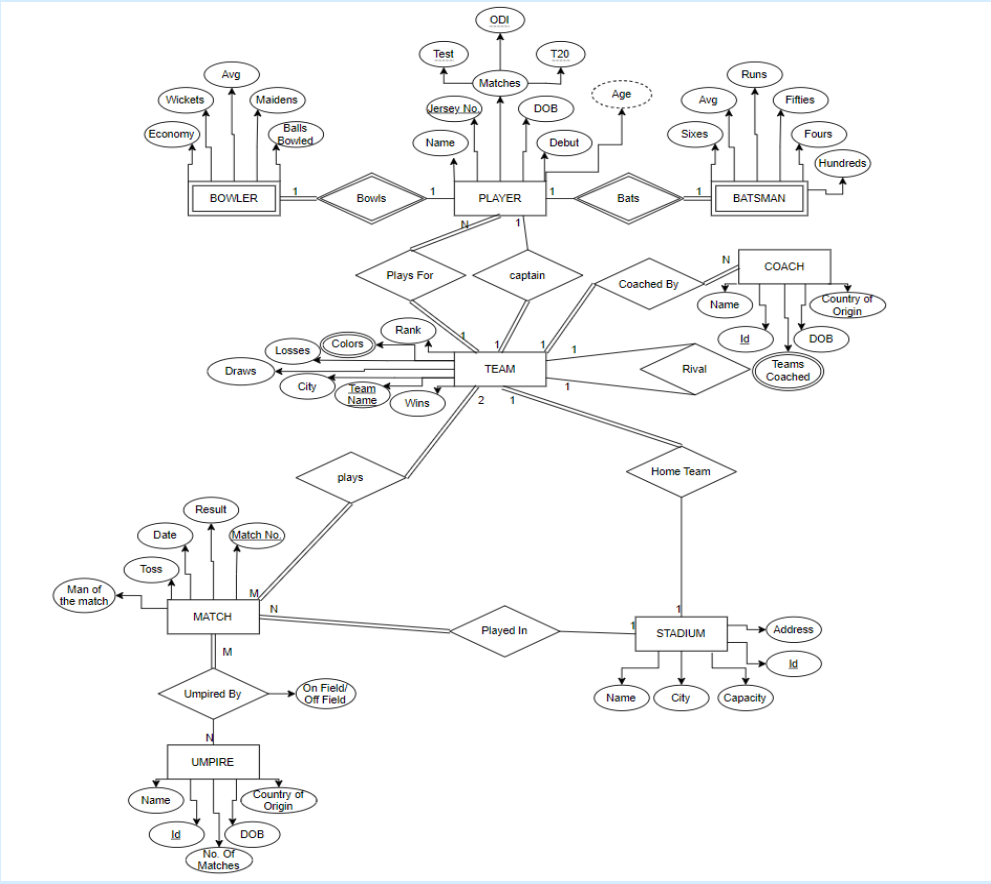
- plays (N-M)
  This relationship exists between the entities: match and team. A team plays multiple matches through the course of the tournament, but only two teams play in a match. Hence, this is an N-M relationship, specifically an N-2 relationship. Every team must play at least one match and each match needs two teams. Hence, there is total participation from both entities.

- Home team (1-1)
  This relationship exists between the entities: stadium and team. A team must have exactly one home stadium and each stadium may be the home stadium of at most 1 team. Hence, this is a 1-1 relationship with total participation on the team entity and partial participation on the stadium entity.

- Played in (N-1)
  This relationship exists between the entities: match and stadium. Multiple matches may be played in a stadium but only one stadium can be used for a match. Hence, this is an N-1 relationship. Each match requires at least one stadium but some of the stadiums may not be used for holding any of the matches. Hence, there is total participation on the match entity and partial participation on the stadium entity.

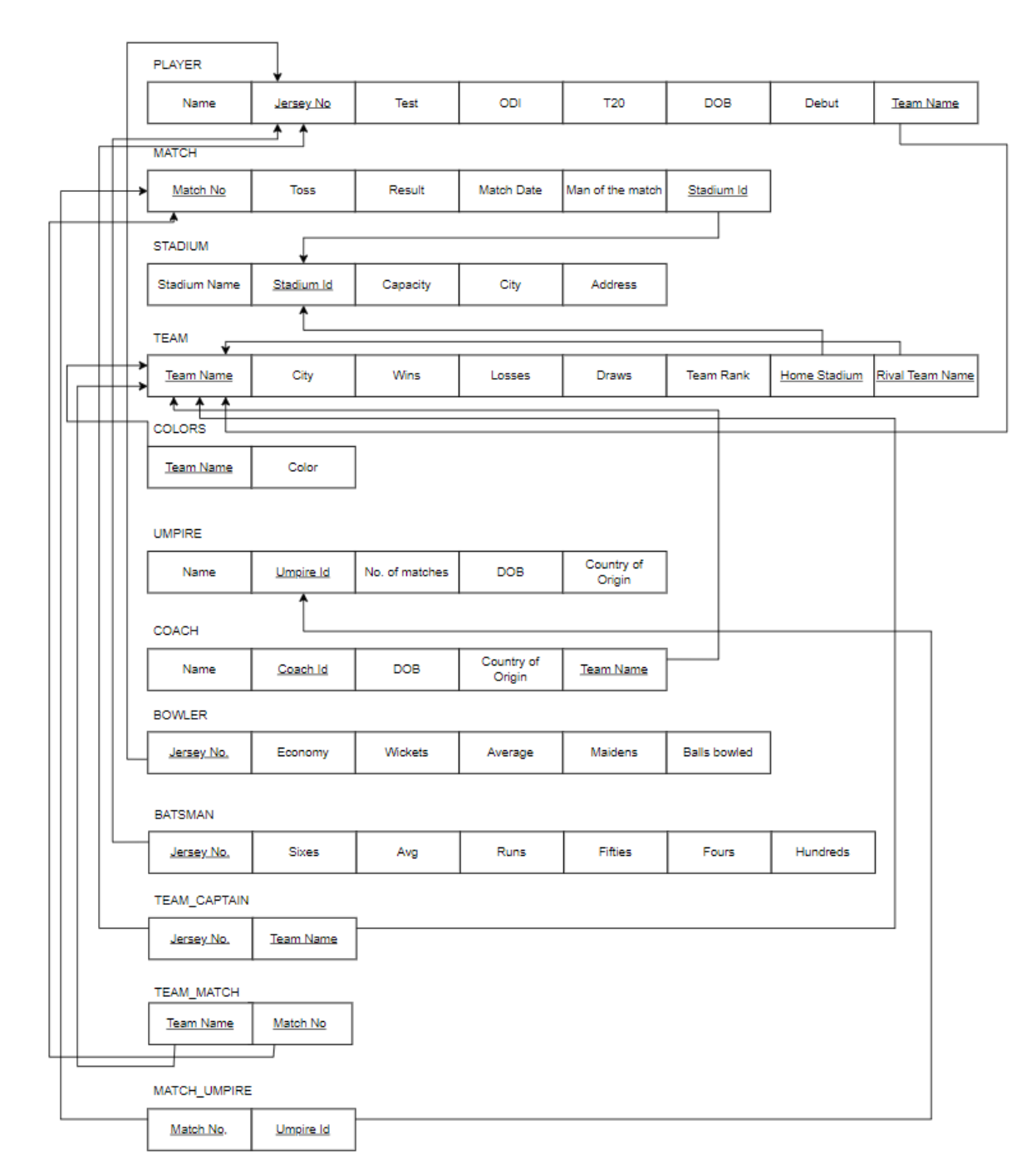
- Umpired by (N-1)
  This relationship exists between the entities: match and umpire. An umpire may be present in multiple matches and a match has multiple umpires, in general, 3. Hence, this is an N-M relationship. Each match requires at least one umpire but some of the umpires may not be present in any of the matches. Hence, there is total participation on the match entity and partial participation on the umpire entity. This relationship also contains an attribute on-field/off-field which indicates if the specific umpire in the match is an on-field umpire or an off-field umpire.

  The final ER Diagram is as follows:



Relational Schema

The following diagram represents the relational schema for the ER diagram show above:



DDL Statements – building the database

These are the statements used to create all the tables in the database:

```
CREATE TABLE STADIUM(stadium_name VARCHAR(50), stadium_id INT PRIMARY KEY,
capacity INT, city VARCHAR(20), address VARCHAR(50));

CREATE TABLE TEAM(team_name VARCHAR(50) PRIMARY KEY, city VARCHAR(20), wins INT,
losses INT, draws INT, team_rank INT, home_stadium_id INT, rival_team_name
VARCHAR(50), FOREIGN KEY(home_stadium_id) REFERENCES STADIUM(stadium_id) ON
DELETE SET NULL ON UPDATE CASCADE, FOREIGN KEY(rival_team_name) REFERENCES
TEAM(team_name) ON UPDATE CASCADE ON DELETE SET NULL);

CREATE TABLE PLAYER(player_name VARCHAR(50), jersey_no INT PRIMARY KEY, test INT,
odi INT, t20 INT, dob DATE, debut DATE, keeper VARCHAR(10), team_name
VARCHAR(50), FOREIGN KEY(team_name) REFERENCES TEAM(team_name) ON DELETE SET NULL
ON UPDATE CASCADE);

CREATE TABLE MATCHES(match_no INT PRIMARY KEY, toss VARCHAR(10), result
VARCHAR(50), match_date DATE, man_of_match VARCHAR(50), stadium_id INT, FOREIGN
KEY(stadium_id) REFERENCES STADIUM(stadium_id) ON DELETE CASCADE ON UPDATE
CASCADE);

CREATE TABLE COLORS(team_name VARCHAR(50) NOT NULL, color VARCHAR(20), FOREIGN
KEY(team_name) REFERENCES TEAM(team_name) ON DELETE CASCADE ON UPDATE CASCADE);

CREATE TABLE UMPIRE(umpire_name VARCHAR(50), umpire_id INT PRIMARY KEY,
no_of_matches INT, dob DATE, country_origin VARCHAR(20));

CREATE TABLE COACH(coach_name VARCHAR(50), coach_id INT PRIMARY KEY, dob DATE,
country_origin VARCHAR(20), team_name VARCHAR(50), FOREIGN KEY(team_name)
REFERENCES TEAM(team_name) ON DELETE SET NULL ON UPDATE CASCADE);

CREATE TABLE BOWLER(jersey_no INT PRIMARY KEY, economy FLOAT, wickets INT,
average FLOAT, runs INT, balls_bowled INT, FOREIGN KEY(jersey_no) REFERENCES
PLAYER(jersey_no) ON DELETE CASCADE ON UPDATE CASCADE);

CREATE TABLE BATSMAN(jersey_no INT PRIMARY KEY, sixes INT, runs INT, average
FLOAT, fifties INT, fours INT, hundreds INT, FOREIGN KEY(jersey_no) REFERENCES
PLAYER(jersey_no) ON DELETE CASCADE ON UPDATE CASCADE);

CREATE TABLE TEAM_CAPTAIN(jersey_no INT PRIMARY KEY, team_name VARCHAR(50)
UNIQUE, FOREIGN KEY(jersey_no) REFERENCES PLAYER(jersey_no) ON UPDATE CASCADE,
```

```
FOREIGN KEY(team_name) REFERENCES TEAM(team_name) ON DELETE CASCADE ON UPDATE
CASCADE);

CREATE TABLE TEAM_MATCH(team_name VARCHAR(50), match_no INT NOT NULL, FOREIGN
KEY(team_name) REFERENCES TEAM(team_name) ON UPDATE CASCADE, FOREIGN
KEY(match_no) REFERENCES MATCHES(match_no) ON DELETE CASCADE ON UPDATE CASCADE);

CREATE TABLE MATCH_UMPIRE(match_no INT NOT NULL, umpire_id INT NOT NULL, FOREIGN
KEY(match_no) REFERENCES MATCHES(match_no) ON DELETE CASCADE ON UPDATE CASCADE,
FOREIGN key(umpire_id) REFERENCES UMPIRE(umpire_id) ON DELETE CASCADE ON UPDATE
CASCADE);
```

On delete and on update cascade/set nulls have been taken care of.

Populating the database

To populate the database, I mostly used the insert into statements to add initial values. After developing the front end, it was easier to add it through the interface. The following shows all the commands used to insert initial data into the database.

```
INSERT INTO STADIUM VALUES
("M. Chinnaswamy Stadium",1,40000,"Bengaluru","Mahatma Gandhi Rd, near Cubbon
Road"),
("Arun Jaitley Ground",2,41842,"Delhi","Jawaharlal Nehru Marg, Feroze Shah
Kotla"),
("Eden Gardens",3,66000,"Kolkata","Maidan, B.B.D. Bagh"),
("Wankhede Stadium",4,33500,"Mumbai","Vinoo Mankad Rd, Churchgate"),
("Rajiv Gandhi International Cricket Stadium",5,55000,"Hyderabad","RGI Stadium
Rd, Uppal"),
("M. A. Chidambaram Stadium",6,50000,"Chennai","Wallajah Road, Chepauk"),
("Sawai Mansingh Stadium",7,30000,"Jaipur","Janpath, Jaipur Nagar Nigam"),
("PCA Stadium",8,27000,"Mohali"," I.S. Bindra Stadium, Sukhna Path");


INSERT INTO TEAM(team_name,city,wins,losses,draws,team_rank,home_stadium_id)
VALUES
("Royal Challengers Bangalore","Bengaluru",7,7,0,4,1),
("Sunrisers Hyderabad","Hyderabad",7,7,0,3,5),
("Mumbai Indians","Mumbai",9,5,0,1,4),
("Chennai Super Kings","Chennai",6,8,0,7,6),
("Kolkata Knight Riders","Kolkata",7,7,0,5,3),
("Rajasthan Royals","Jaipur",6,8,0,8,7),
("Punjab Kings","Mohali",6,8,0,6,8),
("Delhi Capitals","Delhi",8,6,0,2,2);
```

```
UPDATE TEAM SET rival_team_name="Sunrisers Hyderabad" WHERE team_name="Royal
Challengers Bangalore";
UPDATE TEAM SET rival_team_name="Royal Challengers Bangalore" WHERE
team_name="Sunrisers Hyderabad";
UPDATE TEAM SET rival_team_name="Mumbai Indians" WHERE team_name="Chennai Super
Kings";
UPDATE TEAM SET rival_team_name="Chennai Super Kings" WHERE team_name="Mumbai
Indians";


INSERT INTO COLORS VALUES
("Royal Challengers Bangalore","red"),
("Royal Challengers Bangalore","black"),
("Sunrisers Hyderabad","orange"),
("Sunrisers Hyderabad","black"),
("Mumbai Indians","light Blue"),
("Chennai Super Kings","yellow"),
("Kolkata Knight Riders","purple"),
("Kolkata Knight Riders","gold"),
("Rajasthan Royals","pink"),
("Rajasthan Royals","dark blue"),
("Punjab Kings","red"),
("Punjab Kings","gold"),
("Delhi Capitals","red"),
("Delhi Capitals","blue");


INSERT INTO UMPIRE VALUES
("Sundaram Ravi",1,131,"1966-04-22","India"),
("Anil Chaudhary",2,109,"1965-03-12","India"),
("Kumar Dharmasena",3,94,"1971-04-24","Sri Lanka"),
("Chettithody Shamshuddin",4,89,"1970-03-22","India"),
("Nitin Menon",5,78,"1983-11-02","India"),
("Marais Erasmus",6,69,"1964-02-27","South Africa"),
("Chris Gaffaney",7,64,"1975-11-30","New Zealand"),
("C. K. Nandan",8,59,"1963-10-14","India"),
("Bruce Oxenford",9,55,"1960-03-05","Australia"),
("Asad Rauf",10,51,"1956-05-12","Pakistan");


--LOAD DATA INFILE "C:\Users\anura\Desktop\PESU SEM 5\DBMS\Project\players.csv"
INTO TABLE PLAYER COLUMNS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"' ESCAPED BY
'"' LINES TERMINATED BY '\n' IGNORE 1 LINES;
```

```
INSERT INTO PLAYER VALUES
("Virat Kohli",18,102,262,115,"1988-11-05","2008-12-18",0,"Royal Challengers
Bangalore"),
("AB De Villiers",17,114,228,78,"1984-02-17","2005-02-02",1,"Royal Challengers
Bangalore"),
("Mohammed Siraj",73,13,13,6,"1994-03-13","2019-01-15",0,"Royal Challengers
Bangalore"),
("Glenn Maxwell",34,7,127,98,"1988-10-14","2008-12-18",0,"Royal Challengers
Bangalore"),
("Wanindu Hasaranga",41,4,31,32,"1997-06-29","2019-09-01",0,"Royal Challengers
Bangalore"),
("MS Dhoni",7,90,350,98,"1981-07-07","2005-12-02",1,"Chennai Super Kings"),
("Robin Uthappa",8,60,171,64,"1985-11-11","2006-04-09",1,"Chennai Super Kings"),
("Ravindra Jadeja",88,102,262,115,"1988-12-06","2008-04-19",0,"Chennai Super
Kings"),
("Deepak Chahar",90,9,24,63,"1992-08-07","2018-07-08",0,"Chennai Super Kings"),
("Ambati rayudu",5,0,55,6,"1985-09-23","2013-07-24",0,"Chennai Super Kings"),
("Rohit Sharma",45,45,223,128,"1987-04-30","2013-11-06",0,"Mumbai Indians"),
("Suryakumar Yadav",63,13,40,123,"1990-09-14","2021-07-18",0,"Mumbai Indians"),
("Ishan Kishan",32,0,9,19,"1998-07-18","2021-07-18",1,"Mumbai Indians"),
("Jasprit Bumrah",93,30,72,60,"1993-12-06","2018-01-05",0,"Mumbai Indians"),
("Kieron Pollard",55,0,123,101,"1987-05-12","2007-04-10",0,"Mumbai Indians"),
("Kane Williamson",78,102,262,115,"1988-11-05","2010-11-04",0,"Sunrisers
Hyderabad"),
("Bhuvneshwar Kumar",15,21,121,85,"1990-02-05","2013-02-22",0,"Sunrisers
Hyderabad"),
("Glenn Phillips",23,1,6,54,"1996-12-06","2020-01-02",1,"Sunrisers Hyderabad"),
("Abdul Samad",28,0,0,0,"2001-10-28","2020-12-29",0,"Sunrisers Hyderabad"),
("Umran Malik",24,0,0,3,"1999-11-22","2022-07-26",0,"Sunrisers Hyderabad"),
("Shreyas Iyer",43,5,33,47,"1994-12-06","2017-11-01",0,"Kolkata Knight Riders"),
("Sam Billings",97,3,25,37,"1991-07-15","2015-07-09",1,"Kolkata Knight Riders"),
("Alex Hales",2,11,70,75,"1989-01-03","2015-12-26",0,"Kolkata Knight Riders"),
("Andre Russell",12,1,56,67,"1988-04-29","2010-11-15",0,"Kolkata Knight Riders"),
("Pat Cummins",30,43,73,50,"1993-05-08","2011-11-17",0,"Kolkata Knight Riders"),
("Sanju Samson",9,0,10,16,"1994-11-11","2015-07-19",0,"Rajasthan Royals"),
("Jos Buttler",64,57,157,103,"1990-07-08","2014-07-27",1,"Rajasthan Royals"),
("Ravichandran Ashwin",99,96,103,65,"1986-07-17","2011-11-06",0,"Rajasthan
Royals"),
("Riyan Parag",56,0,0,0,"2001-11-10","2019-04-11",0,"Rajasthan Royals"),
("Trent Boult",58,78,99,55,"1989-07-22","2011-12-06",0,"Rajasthan Royals"),
("KL Rahul",1,43,45,72,"1992-04-18","2014-12-26",0,"Punjab Kings"),
("Mayank Agarwal",48,102,262,115,"1991-02-16","2018-12-26",1,"Punjab Kings"),
("Arshdeep Singh",26,0,0,19,"1999-02-05","2022-07-07",0,"Punjab Kings"),
("Shikhar Dhawan",42,34,161,68,"1985-12-05","2013-03-14",0,"Punjab Kings"),
("Jonny Bairstow",51,89,95,66,"1989-11-26","2011-11-16",0,"Punjab Kings"),
```

```
("Rishabh Pant",87,31,27,64,"1997-10-04","2018-08-08",1,"Delhi Capitals"),
("David Warner",31,96,138,99,"1986-10-27","2011-12-01",0,"Delhi Capitals"),
("Prithvi Shaw",100,5,6,1,"1999-11-09","2020-12-17",0,"Delhi Capitals"),
("Axar Patel",20,6,44,37,"1994-01-20","2014-07-15",0,"Delhi Capitals"),
("Shardul Thakur",54,8,27,25,"1991-10-16","2017-08-31",0,"Delhi Capitals");


INSERT INTO BATSMAN VALUES
(18,218,6624,36.2,44,578,5),
(7,229,4978,39.2,24,346,0),
(45,240,5879,30.3,40,419,1),
(63,84,2644,30.05,16,284,0),
(42,136,6243,34.88,47,701,2),
(87,129,2838,34.61,15,260,1),
(88,90,2502,26.6,2,182,0);
--jersey_no,sixes,runs,avg, fifties,fours,hundreds


INSERT INTO BOWLER VALUES
(73,8.78,59,33.07,1951,1334),
(88,7.61,132,30.79,4064,3205),
(90,7.8,59,29.19,1772,1324),
(93,7.4,145,23.31,3380,2742),
(30,8.54,45,30.16,1357,953),
(24,8.83,24,22.5,540,367);
--jersey_no,economy,wickets,avg,runs,balls_bowled


INSERT INTO MATCHES VALUES
(1,"head","Mumbai Indians","2020-10-01","Kieron Pollard",4),
(2,"tails","Royal Challengers Bangalore","2020-10-02","Virat Kohli",7),
(3,"tails","Delhi Capitals","2020-10-03","Shreyas Iyer",2),
(4,"head","Chennai Super Kings","2020-10-04","Shane Watson",6),
(5,"head","Delhi Capitals","2020-10-05","Axar Patel",1),
(6,"tails","Mumbai Indians","2020-10-06","Suryakumar Yadav",8),
(7,"tails","Kolkata Knight Riders","2020-10-07","Rahul Tripathi",3),
(8,"head","Rajasthan Royals","2020-10-08","Jos Buttler",5);




INSERT INTO TEAM_MATCH VALUES
("Mumbai Indians",1),
("Punjab Kings",1),
("Royal Challengers Bangalore",2),
("Rajasthan Royals",2),
```

```
("Delhi Capitals",3),
("Kolkata Knight Riders",3),
("Chennai Super Kings",4),
("Sunrisers Hyderabad",4),
("Royal Challengers Bangalore",5),
("Delhi Capitals",5),
("Mumbai Indians",6),
("Punjab Kings",6),
("Kolkata Knight Riders",7),
("Chennai Super Kings",7),
("Rajasthan Royals",8),
("Sunrisers Hyderabad",8);


INSERT INTO COACH VALUES
("Stephen Fleming",1,"1973-04-01","Australia","Chennai Super Kings"),
("Mahela Jayawardene",2,"1977-03-27","Sri Lanka","Mumbai Indians"),
("Sanjay Bangar",3,"1972-10-11","India","Royal Challengers Bangalore"),
("Ricky Ponting",4,"1974-12-19","Australia","Delhi Capitals"),
("Brian Lara",5,"1969-03-02","West Indies","Sunrisers Hyderabad"),
("Anil Kumble",6,"1970-10-17","India","Punjab Kings"),
("Brendon McCullum",7,"1971-10-21","New Zealand","Kolkata Knight Riders"),
("Kumar Sangakkara",8,"1977-10-27","Sri Lanka","Rajasthan Royals");


INSERT INTO TEAM_CAPTAIN VALUES
(18,"Royal Challengers Bangalore"),
(7,"Chennai Super Kings"),
(45,"Mumbai Indians"),
(9,"Rajasthan Royals"),
(87,"Delhi Capitals"),
(1,"Punjab Kings"),
(43,"Kolkata Knight Riders"),
(78,"Sunrisers Hyderabad");


INSERT INTO MATCH_UMPIRE VALUES
(1,1),
(1,2),
(1,3),
(2,4),
(2,5),
(2,6),
(3,7),
(3,8),
```

```
(3,9),
(4,10),
(4,1),
(4,2),
(5,3),
(5,4),
(5,5),
(6,6),
(6,7),
(6,8),
(7,9),
(7,10),
(7,1),
(8,2),
(8,3),
(8,4);
```

Join queries

For all the queries two outputs are shown as follows: one from the command prompt and one from the front end which has a custom query feature which shows the output in a tabular form.

1. Use left join to get the stats of each batsman in the team

```
select * from
BATSMAN LEFT JOIN PLAYER
ON BATSMAN.jersey_no=PLAYER.jersey_no;
```

Front-end output:

	jersey no	sixes	runs	avg	fifties	fours	hundreds	player name	jersey_no	test	odi	t20	dob	debut	keeper	team name	age
0	7	235	4978	39.2000	24	346	0	MS Dhoni	7	90	350	98	1981-07-07	2005-12-02	1	Chennai Super Kings	41
1	18	218	6624	36.2000	44	578	5	Virat Kohli	18	102	262	115	1988-11-05	2008-12-18	0	Royal Challengers Bangalore	34
2	42	136	6243	34.8800	47	701	2	Shikhar Dhawan	42	34	161	68	1985-12-05	2013-03-14	0	Punjab Kings	36
3	45	240	5879	30.3000	40	419	1	Rohit Sharma	45	45	223	128	1987-04-30	2013-11-06	0	Mumbai Indians	35
4	63	84	2644	30.0500	16	284	0	Suryakumar Yadav	63	13	40	123	1990-09-14	2021-07-18	0	Mumbai Indians	32
5	87	129	2838	34.6100	15	260	1	Rishabh Pant	87	31	27	64	1997-10-04	2018-08-08	1	Delhi Capitals	25
6	88	90	2502	26.6000	2	182	0	Ravindra Jadeja	88	102	262	115	1988-12-06	2008-04-19	0	Chennai Super Kings	33
7	999	343	345	32.2000	345	543	12	test	999	54	67	10	2002-09-09	2003-09-09	1	Mumbai Indians	20

Command prompt output:

```
mysql> select * from
    -> BATSMAN LEFT JOIN PLAYER
    -> ON BATSMAN.jersey_no=PLAYER.jersey_no;
+----------+-------+------+---------+--------+-------+----------+-------------+----------+------+------+------+------------+------------+--------+--------------------------+------+
| jersey_no | sixes | runs | average | fifties | fours | hundreds | player_name | jersey_no | test | odi | t20 | dob | debut | keeper | team_name | age |
+----------+-------+------+---------+--------+-------+----------+-------------+----------+------+------+------+------------+------------+--------+--------------------------+------+
|         7 |   235 | 4978 |    39.2 |     24 |   346 |        0 | MS Dhoni    |         7 |   90 |  350 |   98 | 1981-07-07 | 2005-12-02 | 1      | Chennai Super Kings      |   41 |
|        18 |   218 | 6624 |    36.2 |     44 |   578 |        5 | Virat Kohli |        18 |  102 |  262 |  115 | 1988-11-05 | 2008-12-18 | 0      | Royal Challengers Bangalore |   34 |
|        42 |   136 | 6243 |   34.88 |     47 |   701 |        2 | Shikhar Dhawan |        42 |   34 |  161 |   68 | 1985-12-05 | 2013-03-14 | 0      | Punjab Kings             |   36 |
|        45 |   240 | 5879 |    30.3 |     40 |   419 |        1 | Rohit Sharma |        45 |   45 |  223 |  128 | 1987-04-30 | 2013-11-06 | 0      | Mumbai Indians           |   35 |
|        63 |    84 | 2644 |   30.05 |     16 |   284 |        0 | Suryakumar Yadav |        63 |   13 |   40 |  123 | 1990-09-14 | 2021-07-18 | 0      | Mumbai Indians           |   32 |
|        87 |   129 | 2838 |   34.61 |     15 |   260 |        1 | Rishabh Pant |        87 |   31 |   27 |   64 | 1997-10-04 | 2018-08-08 | 1      | Delhi Capitals           |   25 |
|        88 |    90 | 2502 |    26.6 |      2 |   182 |        0 | Ravindra Jadeja |        88 |  102 |  262 |  115 | 1988-12-06 | 2008-04-19 | 0      | Chennai Super Kings      |   33 |
|       999 |   343 |  345 |    32.2 |    345 |   543 |       12 | test        |       999 |   54 |   67 |   10 | 2002-09-09 | 2003-09-09 | 1      | Mumbai Indians           |   20 |
+----------+-------+------+---------+--------+-------+----------+-------------+----------+------+------+------+------------+------------+--------+--------------------------+------+
8 rows in set (0.00 sec)
```

2. Using left join to get the stats of each bowler in the team:

```
select * from
BOWLER LEFT JOIN PLAYER
ON BOWLER.jersey_no=PLAYER.jersey_no;
```

Front-end output:

	jersey no	economy	wickets	avg	runs	balls bowled	player name	jersey_no	test	odi	t20	dob	debut	keeper	team name	age
0	24	8.8300	24	22.5000	540	367	Umran Malik	24	0	0	3	1999-11-22	2022-07-26	0	Sunrisers Hyderabad	22
1	30	8.5400	45	30.1600	1357	953	Pat Cummins	30	43	73	50	1993-05-08	2011-11-17	0	Kolkata Knight Riders	29
2	73	8.7800	59	33.0700	1951	1334	Mohammed Siraj	73	13	13	6	1994-03-13	2019-01-15	0	Royal Challengers Bangalore	28
3	88	7.6100	132	30.7900	4064	3205	Ravindra Jadeja	88	102	262	115	1988-12-06	2008-04-19	0	Chennai Super Kings	33
4	90	7.8000	59	29.1900	1772	1324	Deepak Chahar	90	9	24	63	1992-08-07	2018-07-08	0	Chennai Super Kings	30
5	93	7.4000	145	23.3100	3380	2742	Jasprit Bumrah	93	30	72	60	1993-12-06	2018-01-05	0	Mumbai Indians	28
6	323232	90.0900	87	87.9898	98978	8989	test2	323232	8787	8776	323	2009-09-09	2009-09-09	1	Mumbai Indians	13

Command prompt output:

```
mysql> select * from
    -> BOWLER LEFT JOIN PLAYER
    -> ON BOWLER.jersey_no=PLAYER.jersey_no;
+----------+---------+---------+---------+------+-------------+-------------+----------+------+------+------+------------+------------+--------+--------------------------+------+
| jersey_no | economy | wickets | average | runs | balls_bowled | player_name | jersey_no | test | odi | t20 | dob | debut | keeper | team_name | age |
+----------+---------+---------+---------+------+-------------+-------------+----------+------+------+------+------------+------------+--------+--------------------------+------+
|        24 |    8.83 |      24 |    22.5 |  540 |         367 | Umran Malik |        24 |    0 |    0 |    3 | 1999-11-22 | 2022-07-26 | 0      | Sunrisers Hyderabad      |   22 |
|        30 |    8.54 |      45 |   30.16 | 1357 |         953 | Pat Cummins |        30 |   43 |   73 |   50 | 1993-05-08 | 2011-11-17 | 0      | Kolkata Knight Riders    |   29 |
|        73 |    8.78 |      59 |   33.07 | 1951 |        1334 | Mohammed Siraj |        73 |   13 |   13 |    6 | 1994-03-13 | 2019-01-15 | 0      | Royal Challengers Bangalore |   28 |
|        88 |    7.61 |     132 |   30.79 | 4064 |        3205 | Ravindra Jadeja |        88 |  102 |  262 |  115 | 1988-12-06 | 2008-04-19 | 0      | Chennai Super Kings      |   33 |
|        90 |     7.8 |      59 |   29.19 | 1772 |        1324 | Deepak Chahar |        90 |    9 |   24 |   63 | 1992-08-07 | 2018-07-08 | 0      | Chennai Super Kings      |   30 |
|        93 |     7.4 |     145 |   23.31 | 3380 |        2742 | Jasprit Bumrah |        93 |   30 |   72 |   60 | 1993-12-06 | 2018-01-05 | 0      | Mumbai Indians           |   28 |
|    323232 |   90.09 |      87 | 87.9898 | 98978 |        8989 | test2       |    323232 | 8787 | 8776 |  323 | 2009-09-09 | 2009-09-09 | 1      | Mumbai Indians           |   13 |
+----------+---------+---------+---------+------+-------------+-------------+----------+------+------+------+------------+------------+--------+--------------------------+------+
7 rows in set (0.00 sec)
```

3. Using right join to get the player details of the captain of each team

```
select * from
PLAYER RIGHT JOIN TEAM_CAPTAIN
ON TEAM_CAPTAIN.jersey_no=PLAYER.jersey_no;
```

Front-end output:

	player name	jersey no	test	odi	t20	dob	debut	keeper	team name	age	jersey_no	team_name
0	MS Dhoni	7	90	350	98	1981-07-07	2005-12-02	1	Chennai Super Kings	41	7	Chennai Super Kings
1	Rishabh Pant	87	31	27	64	1997-10-04	2018-08-08	1	Delhi Capitals	25	87	Delhi Capitals
2	Shreyas Iyer	43	5	33	47	1994-12-06	2017-11-01	0	Kolkata Knight Riders	27	43	Kolkata Knight Riders
3	Rohit Sharma	45	45	223	128	1987-04-30	2013-11-06	0	Mumbai Indians	35	45	Mumbai Indians
4	KL Rahul	1	43	45	72	1992-04-18	2014-12-26	0	Punjab Kings	30	1	Punjab Kings
5	Sanju Samson	9	0	10	16	1994-11-11	2015-07-19	0	Rajasthan Royals	28	9	Rajasthan Royals
6	Virat Kohli	18	102	262	115	1988-11-05	2008-12-18	0	Royal Challengers Bangalore	34	18	Royal Challengers Bangalore
7	Kane Williamson	78	102	262	115	1988-11-05	2010-11-04	0	Sunrisers Hyderabad	34	78	Sunrisers Hyderabad

Command prompt output:

```
mysql> select * from
    -> PLAYER RIGHT JOIN TEAM_CAPTAIN
    -> ON TEAM_CAPTAIN.jersey_no=PLAYER.jersey_no;
+---------------+-----------+------+------+------+------------+------------+--------+---------------------------+------+-----------+---------------------------+
| player_name   | jersey_no | test | odi  | t20  | dob        | debut      | keeper | team_name                 | age  | jersey_no | team_name                 |
+---------------+-----------+------+------+------+------------+------------+--------+---------------------------+------+-----------+---------------------------+
| MS Dhoni      |         7 |   90 |  350 |   98 | 1981-07-07 | 2005-12-02 | 1      | Chennai Super Kings       |   41 |         7 | Chennai Super Kings       |
| Rishabh Pant  |        87 |   31 |   27 |   64 | 1997-10-04 | 2018-08-08 | 1      | Delhi Capitals            |   25 |        87 | Delhi Capitals            |
| Shreyas Iyer  |        43 |    5 |   33 |   47 | 1994-12-06 | 2017-11-01 | 0      | Kolkata Knight Riders     |   27 |        43 | Kolkata Knight Riders     |
| Rohit Sharma  |        45 |   45 |  223 |  128 | 1987-04-30 | 2013-11-06 | 0      | Mumbai Indians            |   35 |        45 | Mumbai Indians            |
| KL Rahul      |         1 |   43 |   45 |   72 | 1992-04-18 | 2014-12-26 | 0      | Punjab Kings              |   30 |         1 | Punjab Kings              |
| Sanju Samson  |         9 |    0 |   10 |   16 | 1994-11-11 | 2015-07-19 | 0      | Rajasthan Royals          |   28 |         9 | Rajasthan Royals          |
| Virat Kohli   |        18 |  102 |  262 |  115 | 1988-11-05 | 2008-12-18 | 0      | Royal Challengers Bangalore |   34 |        18 | Royal Challengers Bangalore |
| Kane Williamson |      78 |  102 |  262 |  115 | 1988-11-05 | 2010-11-04 | 0      | Sunrisers Hyderabad       |   34 |        78 | Sunrisers Hyderabad       |
+---------------+-----------+------+------+------+------------+------------+--------+---------------------------+------+-----------+---------------------------+
8 rows in set (0.00 sec)
```

4. Natural join to fetch the details of the all-rounders:

```
select*
from PLAYER as p NATURAL JOIN
(SELECT(bat.jersey_no)
FROM BATSMAN as bat JOIN
BOWLER as bol
ON bat.jersey_no=bol.jersey_no) as allr;
```

Front-end output:

	jersey no	player name	test	odi	t20	dob	debut	keeper	team name	age
0	88	Ravindra Jadeja	102	262	115	1988-12-06	2008-04-19	0	Chennai Super Kings	33

Command prompt output:

```
+-----------+-----------------+------+------+------+------------+------------+--------+---------------------+------+
| jersey_no | player_name     | test | odi  | t20  | dob        | debut      | keeper | team_name           | age  |
+-----------+-----------------+------+------+------+------------+------------+--------+---------------------+------+
|        88 | Ravindra Jadeja |  102 |  262 |  115 | 1988-12-06 | 2008-04-19 | 0      | Chennai Super Kings |   33 |
+-----------+-----------------+------+------+------+------------+------------+--------+---------------------+------+
```

Aggregate results

1. Find the player with the most test games in his career

```
SELECT * FROM PLAYER
WHERE test=(SELECT MAX(test)
FROM PLAYER);
```

Front-end output:

	player name	jersey no	test	odi	t20	dob	debut	keeper	team name	age
0	AB De Villiers	17	114	228	78	1984-02-17	2005-02-02	1	Royal Challengers Bangalore	38

Command prompt output:

```
mysql> SELECT * FROM PLAYER
    -> WHERE test=(SELECT MAX(test)
    -> FROM PLAYER);
+---------------+-----------+------+------+------+------------+------------+--------+-----------------------------+------+
| player_name   | jersey_no | test | odi  | t20  | dob        | debut      | keeper | team_name                   | age  |
+---------------+-----------+------+------+------+------------+------------+--------+-----------------------------+------+
| AB De Villiers |        17 |  114 |  228 |   78 | 1984-02-17 | 2005-02-02 | 1      | Royal Challengers Bangalore |   38 |
+---------------+-----------+------+------+------+------------+------------+--------+-----------------------------+------+
1 row in set (0.00 sec)
```

2. Find the stadium with the lowest capacity:

```
SELECT * FROM STADIUM
WHERE capacity=(SELECT MIN(capacity)
FROM STADIUM);
```

Front-end output:

	stadium name	stadium ID	capacity	city	address
0	PCA Stadium	8	27000	Mohali	I.S. Bindra Stadium, Sukhna Path

Command prompt output:

```
mysql> SELECT * FROM STADIUM
    -> WHERE capacity=(SELECT MIN(capacity)
    -> FROM STADIUM);
+--------------+------------+----------+--------+-----------------------------------+
| stadium_name | stadium_id | capacity | city   | address                           |
+--------------+------------+----------+--------+-----------------------------------+
| PCA Stadium  |          8 |    27000 | Mohali |  I.S. Bindra Stadium, Sukhna Path |
+--------------+------------+----------+--------+-----------------------------------+
1 row in set (0.02 sec)
```

3. Finding the average economy of all the bowlers in the tournament

```
SELECT AVG(economy)
FROM BOWLER;
```

Front-end output:

	average
0	8.1600

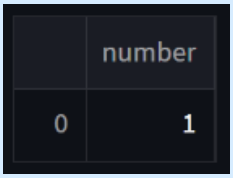
Command prompt output:

```
mysql> SELECT AVG(economy)
    -> FROM BOWLER;
+-------------------+
| AVG(economy)      |
+-------------------+
| 8.160000006357828 |
+-------------------+
1 row in set (0.00 sec)
```

4. Finding the number of umpires in Australia

```
SELECT COUNT(umpire_id)
FROM UMPIRE
WHERE country_origin="Australia";
```

Front-end output:



	number
0	1

Command prompt output:

```
mysql> SELECT COUNT(umpire_id)
    -> FROM UMPIRE
    -> WHERE country_origin="Australia";
+------------------+
| COUNT(umpire_id) |
+------------------+
|                1 |
+------------------+
1 row in set (0.00 sec)
```

Set operations

1. Find the jersey numbers of all the all-rounders

```
SELECT jersey_no FROM BATSMAN
INTERSECT
SELECT jersey_no FROM BOWLER;
```

Front-end output:

	jersey number
0	88

Command prompt output:

```
mysql> SELECT jersey_no FROM BATSMAN INTERSECT SELECT jersey_no FROM BOWLER;
+-----------+
| jersey_no |
+-----------+
|        88 |
+-----------+
1 row in set (0.00 sec)
```

2. Find the players who were born in either 1988 or 1990

```
select * from player where dob like "1988-%"
UNION
select * from player where dob like "1990-%";
```

Front end output

	player name	jersey no	test	odi	t20	dob	debut	keeper	team name	age
0	Andre Russell	12	1	56	67	1988-04-29	2010-11-15	0	Kolkata Knight Riders	34
1	Virat Kohli	18	102	262	115	1988-11-05	2008-12-18	0	Royal Challengers Bangalore	34
2	Glenn Maxwell	34	7	127	98	1988-10-14	2008-12-18	0	Royal Challengers Bangalore	34
3	Kane Williamson	78	102	262	115	1988-11-05	2010-11-04	0	Sunrisers Hyderabad	34
4	Ravindra Jadeja	88	102	262	115	1988-12-06	2008-04-19	0	Chennai Super Kings	33
5	Bhuvneshwar Kumar	15	21	121	85	1990-02-05	2013-02-22	0	Sunrisers Hyderabad	32
6	Suryakumar Yadav	63	13	40	123	1990-09-14	2021-07-18	0	Mumbai Indians	32
7	Jos Buttler	64	57	157	103	1990-07-08	2014-07-27	1	Rajasthan Royals	32

Command prompt output

```
mysql> select * from player where dob like "1988-%"
    -> UNION
    -> select * from player where dob like "1990-%";
+------------------+-----------+------+------+------+------------+------------+--------+-----------------------------+------+
| player_name      | jersey_no | test | odi  | t20  | dob        | debut      | keeper | team_name                   | age  |
+------------------+-----------+------+------+------+------------+------------+--------+-----------------------------+------+
| Andre Russell    |        12 |    1 |   56 |   67 | 1988-04-29 | 2010-11-15 | 0      | Kolkata Knight Riders       |   34 |
| Virat Kohli      |        18 |  102 |  262 |  115 | 1988-11-05 | 2008-12-18 | 0      | Royal Challengers Bangalore |   34 |
| Glenn Maxwell    |        34 |    7 |  127 |   98 | 1988-10-14 | 2008-12-18 | 0      | Royal Challengers Bangalore |   34 |
| Kane Williamson  |        78 |  102 |  262 |  115 | 1988-11-05 | 2010-11-04 | 0      | Sunrisers Hyderabad         |   34 |
| Ravindra Jadeja  |        88 |  102 |  262 |  115 | 1988-12-06 | 2008-04-19 | 0      | Chennai Super Kings         |   33 |
| Bhuvneshwar Kumar |       15 |   21 |  121 |   85 | 1990-02-05 | 2013-02-22 | 0      | Sunrisers Hyderabad         |   32 |
| Suryakumar Yadav |        63 |   13 |   40 |  123 | 1990-09-14 | 2021-07-18 | 0      | Mumbai Indians              |   32 |
| Jos Buttler      |        64 |   57 |  157 |  103 | 1990-07-08 | 2014-07-27 | 1      | Rajasthan Royals            |   32 |
+------------------+-----------+------+------+------+------------+------------+--------+-----------------------------+------+
8 rows in set (0.00 sec)
```

3. Find the players who were born in 1988 and debuted in 2008

```
select * from player where dob like "1988-%"
INTERSECT
select * from player where debut like "2008-%";
```

Front-end output:

	player name	jersey no	test	odi	t20	dob	debut	keeper	team name	age
0	Virat Kohli	18	102	262	115	1988-11-05	2008-12-18	0	Royal Challengers Bangalore	34
1	Glenn Maxwell	34	7	127	98	1988-10-14	2008-12-18	0	Royal Challengers Bangalore	34
2	Ravindra Jadeja	88	102	262	115	1988-12-06	2008-04-19	0	Chennai Super Kings	33

Command prompt output:

```
mysql> select * from player where dob like "1988-%"
    -> INTERSECT
    -> select * from player where debut like "2008-%";
+----------------+-----------+------+------+------+------------+------------+--------+-----------------------------+------+
| player_name    | jersey_no | test | odi  | t20  | dob        | debut      | keeper | team_name                   | age  |
+----------------+-----------+------+------+------+------------+------------+--------+-----------------------------+------+
| Virat Kohli    |        18 |  102 |  262 |  115 | 1988-11-05 | 2008-12-18 | 0      | Royal Challengers Bangalore |   34 |
| Glenn Maxwell  |        34 |    7 |  127 |   98 | 1988-10-14 | 2008-12-18 | 0      | Royal Challengers Bangalore |   34 |
| Ravindra Jadeja |       88 |  102 |  262 |  115 | 1988-12-06 | 2008-04-19 | 0      | Chennai Super Kings         |   33 |
+----------------+-----------+------+------+------+------------+------------+--------+-----------------------------+------+
3 rows in set (0.00 sec)

mysql>
```

4. Find the players who played greater than 50 t20 games but not greater than 50 test games

```
select * from player where t20>50
EXCEPT
select * from player where test>50;
```

Front-end output:

	player name	jersey no	test	odi	t20	dob	debut	keeper	team name	age
0	KL Rahul	1	43	45	72	1992-04-18	2014-12-26	0	Punjab Kings	30
1	Alex Hales	2	11	70	75	1989-01-03	2015-12-26	0	Kolkata Knight Riders	33
2	Andre Russell	12	1	56	67	1988-04-29	2010-11-15	0	Kolkata Knight Riders	34
3	Bhuvneshwar Kur	15	21	121	85	1990-02-05	2013-02-22	0	Sunrisers Hyderabad	32
4	Glenn Phillips	23	1	6	54	1996-12-06	2020-01-02	1	Sunrisers Hyderabad	25
5	Glenn Maxwell	34	7	127	98	1988-10-14	2008-12-18	0	Royal Challengers Bangalore	34
6	Shikhar Dhawan	42	34	161	68	1985-12-05	2013-03-14	0	Punjab Kings	36
7	Rohit Sharma	45	45	223	128	1987-04-30	2013-11-06	0	Mumbai Indians	35
8	Kieron Pollard	55	0	123	101	1987-05-12	2007-04-10	0	Mumbai Indians	35
9	Suryakumar Yada	63	13	40	123	1990-09-14	2021-07-18	0	Mumbai Indians	32
10	Rishabh Pant	87	31	27	64	1997-10-04	2018-08-08	1	Delhi Capitals	25
11	Deepak Chahar	90	9	24	63	1992-08-07	2018-07-08	0	Chennai Super Kings	30
12	Jasprit Bumrah	93	30	72	60	1993-12-06	2018-01-05	0	Mumbai Indians	28

Command prompt output:

```
mysql> select * from player where t20>50
    -> EXCEPT
    -> select * from player where test>50;
+-------------------+-----------+------+------+------+------------+------------+--------+-----------------------------+------+
| player_name       | jersey_no | test | odi  | t20  | dob        | debut      | keeper | team_name                   | age  |
+-------------------+-----------+------+------+------+------------+------------+--------+-----------------------------+------+
| KL Rahul          |         1 |   43 |   45 |   72 | 1992-04-18 | 2014-12-26 | 0      | Punjab Kings                |   30 |
| Alex Hales        |         2 |   11 |   70 |   75 | 1989-01-03 | 2015-12-26 | 0      | Kolkata Knight Riders       |   33 |
| Andre Russell     |        12 |    1 |   56 |   67 | 1988-04-29 | 2010-11-15 | 0      | Kolkata Knight Riders       |   34 |
| Bhuvneshwar Kumar |        15 |   21 |  121 |   85 | 1990-02-05 | 2013-02-22 | 0      | Sunrisers Hyderabad         |   32 |
| Glenn Phillips    |        23 |    1 |    6 |   54 | 1996-12-06 | 2020-01-02 | 1      | Sunrisers Hyderabad         |   25 |
| Glenn Maxwell     |        34 |    7 |  127 |   98 | 1988-10-14 | 2008-12-18 | 0      | Royal Challengers Bangalore |   34 |
| Shikhar Dhawan    |        42 |   34 |  161 |   68 | 1985-12-05 | 2013-03-14 | 0      | Punjab Kings                |   36 |
| Rohit Sharma      |        45 |   45 |  223 |  128 | 1987-04-30 | 2013-11-06 | 0      | Mumbai Indians              |   35 |
| Kieron Pollard    |        55 |    0 |  123 |  101 | 1987-05-12 | 2007-04-10 | 0      | Mumbai Indians              |   35 |
| Suryakumar Yadav  |        63 |   13 |   40 |  123 | 1990-09-14 | 2021-07-18 | 0      | Mumbai Indians              |   32 |
| Rishabh Pant      |        87 |   31 |   27 |   64 | 1997-10-04 | 2018-08-08 | 1      | Delhi Capitals              |   25 |
| Deepak Chahar     |        90 |    9 |   24 |   63 | 1992-08-07 | 2018-07-08 | 0      | Chennai Super Kings         |   30 |
| Jasprit Bumrah    |        93 |   30 |   72 |   60 | 1993-12-06 | 2018-01-05 | 0      | Mumbai Indians              |   28 |
+-------------------+-----------+------+------+------+------------+------------+--------+-----------------------------+------+
13 rows in set (0.00 sec)
```

Functions and procedures

I have used various functions and procedures to connect and summarize the results of various tables in the database. Some of them are shown as follows:

- Stored Procedure to get the age of any table if the date of birth is mentioned in the schema

```
DELIMITER $$
CREATE procedure dob_to_age(IN date_of_birth DATE, OUT age int)
BEGIN
SELECT TIMESTAMPDIFF(YEAR, date_of_birth, CURDATE()) into age;
END $$
DELIMITER ;

DELIMITER $$
CREATE procedure update_age_player()
BEGIN
ALTER TABLE PLAYER
ADD age INT;
UPDATE PLAYER
SET age=TIMESTAMPDIFF(YEAR, dob, CURDATE());
END $$
DELIMITER ;

CALL update_age_player()


DELIMITER $$
CREATE procedure update_age_umpire()
BEGIN
ALTER TABLE UMPIRE
ADD age INT;
UPDATE UMPIRE
SET age=TIMESTAMPDIFF(YEAR, dob, CURDATE());
END $$
DELIMITER ;

CALL update_age_umpire()


DELIMITER $$
CREATE procedure update_age_coach()
BEGIN
ALTER TABLE COACH
```

```
ADD age INT;
UPDATE COACH
SET age=TIMESTAMPDIFF(YEAR, dob, CURDATE());
END $$
DELIMITER ;

CALL update_age_coach();
```

This function basically adds another column named 'age' to three tables: player, umpire and coach. The following for example, shows the age column in the player table

	player name	jersey no	test	odi	t20	dob	debut	keeper	team name	age
0	KL Rahul	1	43	45	72	1992-04-18	2014-12-26	0	Punjab Kings	30
1	Alex Hales	2	11	70	75	1989-01-03	2015-12-26	0	Kolkata Knight Riders	33
2	Ambati rayudu	5	0	55	6	1985-09-23	2013-07-24	0	Chennai Super Kings	37
3	MS Dhoni	7	90	350	98	1981-07-07	2005-12-02	1	Chennai Super Kings	41
4	Robin Uthappa	8	60	171	64	1985-11-11	2006-04-09	1	Chennai Super Kings	37

- Stored Procedure to delete batsman based on the player's name

```
DELIMITER $$
CREATE procedure delete_batsman(IN playerName varchar(50))
BEGIN
DECLARE jn int;
set jn= (SELECT jersey_no from PLAYER where player_name=playerName);
DELETE FROM batsman WHERE jersey_no=jn;
END $$
DELIMITER ;
```

Deleting the batsman Virat Kohli:

CALL delete_batsman("Virat Kohli");

Now, Virat's name does not appear in the batsman list:

	jersey_no	player_name	sixes	runs	average	fifties	fours	hundreds
0	7	MS Dhoni	235	4978	39.2000	24	346	0
1	42	Shikhar Dhawan	136	6243	34.8800	47	701	2
2	45	Rohit Sharma	240	5879	30.3000	40	419	1
3	63	Suryakumar Yadav	84	2644	30.0500	16	284	0
4	87	Rishabh Pant	129	2838	34.6100	15	260	1
5	88	Ravindra Jadeja	90	2502	26.6000	2	182	0

- Stored Procedure to delete bowler based on the player name

```
DELIMITER $$
CREATE procedure delete_bowler(IN playerName varchar(50))
BEGIN
DECLARE jn int;
set jn= (SELECT jersey_no from PLAYER where player_name=playerName);
DELETE FROM bowler WHERE jersey_no=jn;
END $$
DELIMITER ;
```

Now, we try to delete the bowler Jasprit Bumrah:

```
CALL delete_bowler("Jasprit Bumrah")
```

Now, the bowler's name is not present in the list:

	jersey_no	player_name	economy	wickets	average	runs	balls_bowled
0	24	Umran Malik	8.8300	24	22.5000	540	367
1	30	Pat Cummins	8.5400	45	30.1600	1357	953
2	73	Mohammed Siraj	8.7800	59	33.0700	1951	1334
3	88	Ravindra Jadeja	7.6100	132	30.7900	4064	3205
4	90	Deepak Chahar	7.8000	59	29.1900	1772	1324

The above two functions are used in the delete functionality of the front-end.

- Function to return "Money well spent" if the home team wins

```
DELIMITER $$
CREATE FUNCTION home_team_wins(result VARCHAR(50), this_stadium INT)
RETURNS varchar(50)
DETERMINISTIC
BEGIN
DECLARE usr_msg VARCHAR(50);
IF result=(select team_name from TEAM WHERE home_stadium_id=this_stadium) THEN
SET usr_msg="Money well spent";
ELSE SET usr_msg="Money wasted";
END IF;
RETURN usr_msg;
END $$
DELIMITER ;
```

The result is shown as called as follows:

```
SELECT result,home_team_wins(result,stadium_id)
FROM MATCHES;
```

Output:

	result	money_well_spent
0	Mumbai Indians	Money well spent
1	Royal Challengers Bangalore	Money wasted
2	Delhi Capitals	Money well spent
3	Chennai Super Kings	Money well spent
4	Delhi Capitals	Money wasted
5	Mumbai Indians	Money wasted
6	Kolkata Knight Riders	Money well spent
7	Rajasthan Royals	Money wasted

- Function to get the colors of the team altogether within the team column

In our database, the “colors“ table represents a multivalued attribute of the team which represents multiple colours which are part of the team’s jersey. For example, Mumbai Indians have only one colour blue, while RCB has black and red colours in their jersey.

This function fetches all the colours for each team and represents them in comma separated values:

```
DELIMITER $$
CREATE FUNCTION team_colors(teamName VARCHAR(50))
RETURNS varchar(50)
DETERMINISTIC
BEGIN
DECLARE result VARCHAR(50);
SET result=(SELECT t.colors FROM
(SELECT team_name,GROUP_CONCAT(color) as "colors"
FROM colors
```

```
GROUP BY team_name) as t
WHERE team_name=teamName);
RETURN result;
END $$
DELIMITER ;
```

Result:

After calling the function as follows:

```
SELECT team_name,team_colors(team_name)
FROM TEAM;
```

The result is:

	team	colours
0	Royal Challengers Bangalore	red,black,pink
1	Delhi Capitals	red,blue,purple
2	Kolkata Knight Riders	purple,gold
3	Mumbai Indians	light Blue
4	Sunrisers Hyderabad	orange,black
5	Chennai Super Kings	yellow
6	Rajasthan Royals	pink,dark blue
7	Punjab Kings	red,gold

- Function to get the captain of a team

```
DELIMITER $$
CREATE FUNCTION get_captain(teamName VARCHAR(50))
RETURNS VARCHAR(50)
DETERMINISTIC
BEGIN
DECLARE result VARCHAR(50);
DECLARE jn INT;
SET jn=(SELECT jersey_no FROM TEAM_CAPTAIN WHERE team_name=teamName);
SET result=(SELECT player_name FROM PLAYER WHERE jersey_no=jn);
RETURN result;
END $$
DELIMITER ;
```

The function is called as follows:

```
SELECT *,get_captain(team_name)
FROM TEAM;
```

Output:

```
mysql> SELECT *,get_captain(team_name)
    -> FROM TEAM;
+--------------------------+-----------+------+--------+-------+-----------+----------------+--------------------------+----------------------+
| team_name                | city      | wins | losses | draws | team_rank | home_stadium_id | rival_team_name          | get_captain(team_name) |
+--------------------------+-----------+------+--------+-------+-----------+----------------+--------------------------+----------------------+
| Chennai Super Kings      | Chennai   |    6 |      8 |     0 |         7 |              6 | Mumbai Indians           | MS Dhoni             |
| Delhi Capitals           | Delhi     |    8 |      6 |     0 |         2 |              2 | NULL                     | Rishabh Pant         |
| Kolkata Knight Riders    | Kolkata   |    7 |      7 |     0 |         5 |              3 | NULL                     | Shreyas Iyer         |
| Mumbai Indians           | Mumbai    |    9 |      5 |     0 |         1 |              4 | Chennai Super Kings      | Rohit Sharma         |
| Punjab Kings             | Mohali    |    6 |      8 |     0 |         6 |              8 | NULL                     | KL Rahul             |
| Rajasthan Royals         | Jaipur    |    6 |      8 |     0 |         8 |              7 | NULL                     | Sanju Samson         |
| Royal Challengers Bangalore | Bengaluru |    7 |      7 |     0 |         4 |              1 | Sunrisers Hyderabad      | Virat Kohli          |
| Sunrisers Hyderabad      | Hyderabad |    7 |      7 |     0 |         3 |              5 | Royal Challengers Bangalore | Kane Williamson      |
+--------------------------+-----------+------+--------+-------+-----------+----------------+--------------------------+----------------------+
8 rows in set (0.00 sec)
```

The rightmost column shows the captain's name for each team.

- Function to get the teams playing in a certain match

```
DELIMITER $$
CREATE FUNCTION get_first_team_for_match(matchNo INT)
RETURNS VARCHAR(50)
DETERMINISTIC
BEGIN
DECLARE result VARCHAR(50);
SET result=(SELECT team_name FROM
(SELECT* FROM team_match
GROUP BY match_no) as m
WHERE match_no=matchNo);
RETURN result;
END $$
DELIMITER ;

DELIMITER $$
CREATE FUNCTION get_second_team_for_match(matchNo INT)
RETURNS VARCHAR(50)
DETERMINISTIC
BEGIN
DECLARE result VARCHAR(50);
SET result=(SELECT team_name FROM team_match
WHERE match_no=matchNo
AND team_name<>get_first_team_for_match(matchNo));
RETURN result;
END $$
```

```
DELIMITER ;
```

We call it as follows:

```
SELECT *,get_first_team_for_match(match_no),get_second_team_for_match(match_no)
FROM matches;
```

Output:

```
mysql>
mysql> SELECT *,get_first_team_for_match(match_no),get_second_team_for_match(match_no)
    -> FROM matches;
+----------+-------+---------------------------+------------+-----------------+------------+-----------------------------------+------------------------------------+
| match_no | toss  | result                    | match_date | man_of_match    | stadium_id | get_first_team_for_match(match_no) | get_second_team_for_match(match_no) |
+----------+-------+---------------------------+------------+-----------------+------------+-----------------------------------+------------------------------------+
|        1 | head  | Mumbai Indians            | 2020-10-01 | Kieron Pollard  |          4 | Mumbai Indians                    | Punjab Kings                       |
|        2 | tails | Royal Challengers Bangalore | 2020-10-02 | Virat Kohli     |          7 | Royal Challengers Bangalore       | Rajasthan Royals                   |
|        3 | tails | Delhi Capitals            | 2020-10-03 | Shreyas Iyer    |          2 | Delhi Capitals                    | Kolkata Knight Riders              |
|        4 | head  | Chennai Super Kings       | 2020-10-04 | Shane Watson    |          6 | Chennai Super Kings               | Sunrisers Hyderabad                |
|        5 | head  | Delhi Capitals            | 2020-10-05 | Axar Patel      |          1 | Royal Challengers Bangalore       | Delhi Capitals                     |
|        6 | tails | Mumbai Indians            | 2020-10-06 | Suryakumar Yadav |          8 | Mumbai Indians                    | Punjab Kings                       |
|        7 | tails | Kolkata Knight Riders     | 2020-10-07 | Rahul Tripathi  |          3 | Kolkata Knight Riders             | Chennai Super Kings                |
|        8 | head  | Rajasthan Royals          | 2020-10-08 | Jos Buttler     |          5 | Rajasthan Royals                  | Sunrisers Hyderabad                |
+----------+-------+---------------------------+------------+-----------------+------------+-----------------------------------+------------------------------------+
8 rows in set (0.01 sec)
```

This function is used in the view matches feature of the front-end.

- Function to get the umpires in certain match

```
DELIMITER $$
CREATE FUNCTION get_first_umpire(matchNo INT)
RETURNS VARCHAR(50)
DETERMINISTIC
BEGIN
DECLARE result VARCHAR(50);
DECLARE umpid INT;
SET umpid=(SELECT umpire_id FROM
(SELECT* FROM MATCH_UMPIRE
GROUP BY match_no) as m
WHERE match_no=matchNo);
SET result=(SELECT umpire_name
FROM UMPIRE
WHERE umpire_id=umpid);
RETURN result;
END $$
DELIMITER ;
```

We call it as follows:

```
SELECT *,get_first_umpire(match_no)
FROM matches;
```

Output:

```
mysql> SELECT *,get_first_umpire(match_no)
    -> FROM matches;
+----------+-------+----------------------------+------------+------------------+------------+----------------------------+
| match_no | toss  | result                     | match_date | man_of_match     | stadium_id | get_first_umpire(match_no) |
+----------+-------+----------------------------+------------+------------------+------------+----------------------------+
|        1 | head  | Mumbai Indians             | 2020-10-01 | Kieron Pollard   |          4 | Sundaram Ravi              |
|        2 | tails | Royal Challengers Bangalore | 2020-10-02 | Virat Kohli      |          7 | Chettithody Shamshuddin    |
|        3 | tails | Delhi Capitals             | 2020-10-03 | Shreyas Iyer     |          2 | Chris Gaffaney             |
|        4 | head  | Chennai Super Kings        | 2020-10-04 | Shane Watson     |          6 | Asad Rauf                  |
|        5 | head  | Delhi Capitals             | 2020-10-05 | Axar Patel       |          1 | Kumar Dharmasena           |
|        6 | tails | Mumbai Indians             | 2020-10-06 | Suryakumar Yadav |          8 | Marais Erasmus             |
|        7 | tails | Kolkata Knight Riders      | 2020-10-07 | Rahul Tripathi   |          3 | Bruce Oxenford             |
|        8 | head  | Rajasthan Royals           | 2020-10-08 | Jos Buttler      |          5 | Anil Chaudhary             |
+----------+-------+----------------------------+------------+------------------+------------+----------------------------+
8 rows in set (0.00 sec)
```

This function was also used in the view umpires feature of the front-end

Triggers and cursors

- Trigger to limit the number of colors in a team to less than or equal to 3.

```
DELIMITER $$
CREATE TRIGGER color_limit
AFTER INSERT
ON COLORS FOR EACH ROW
BEGIN
DECLARE error_msg VARCHAR(255);
SET error_msg = ('Total number of colors should not exceed 3');
IF (select COUNT(color)
FROM colors
where team_name=new.team_name) >3 THEN
SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT = error_msg;
END IF;
END $$
DELIMITER ;
```

From the existing table, Delhi Capitals has exactly 3colourss: red, blue and purple

team_colors	team_captain	points
red,blue,purple	Rishabh Pant	4

So now if we add another colour, it should throw an error since the trigger exists.

```
mysql> insert into colors values
    -> ("Delhi Capitals","pink");
ERROR 1644 (45000): Total number of colors should not exceed 3
mysql>
```

Hence the trigger was successful.

- Cursor:

The cursor is a Temporary Memory or Temporary Work Station. It is allocated by the Database Server at the Time of Performing DML (Data Manipulation Language) operations on the table by the user.

Cursors have been used extensively in this project to define the backend for the streamlit app. The following code shows how the cursor was defined:

```
mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    database="trial1",
    password="manofsteel"
)
c = mydb.cursor()
```

Now, the cursor 'c' has been used everywhere whenever some backend database functionality needs to be called. For example:
To define the custom query functionality in the backend:

```
def custom_query(query):
    c.execute(query)
    data = c.fetchall()
    return data
```

Methods like c.execute and c.fetchall have been used. Another example:

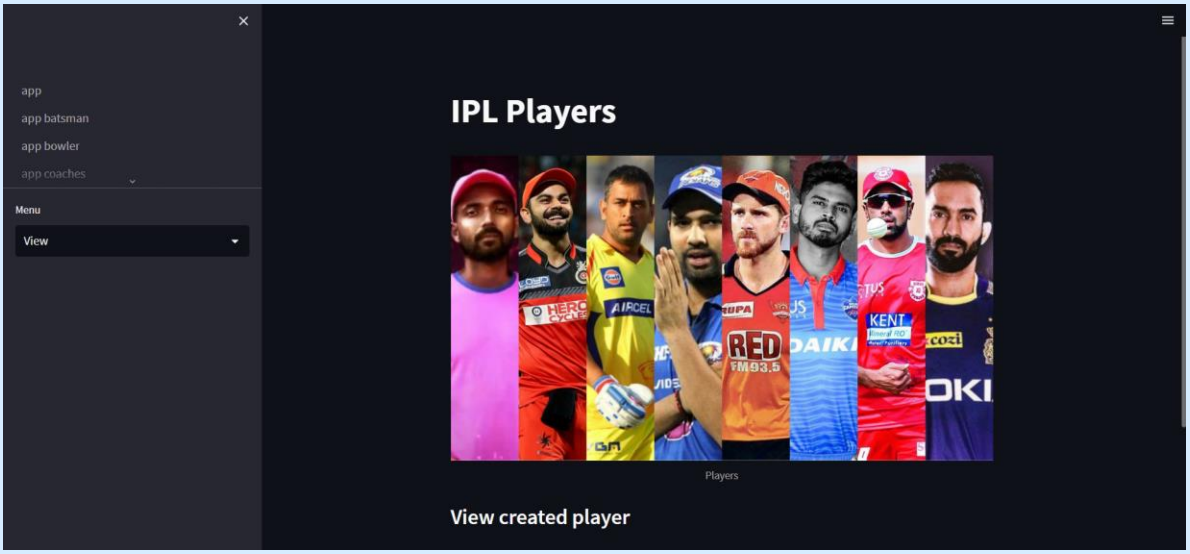
```
def add_player(player_name,jersey_no,test,odi,t20,dob,debut,keeper,team_name):
    c.execute('INSERT INTO
PLAYER(player_name,jersey_no,test,odi,t20,dob,debut,keeper,team_name) VALUES
(%s,%s,%s,%s,%s,%s,%s,%s,%s)',
              (player_name,jersey_no,test,odi,t20,dob,debut,keeper,team_name))
    c.execute('UPDATE PLAYER SET age=TIMESTAMPDIFF(YEAR, dob, CURDATE()) WHERE
player_name="{}"'.format(player_name))
    mydb.commit()
```

Front-end Development

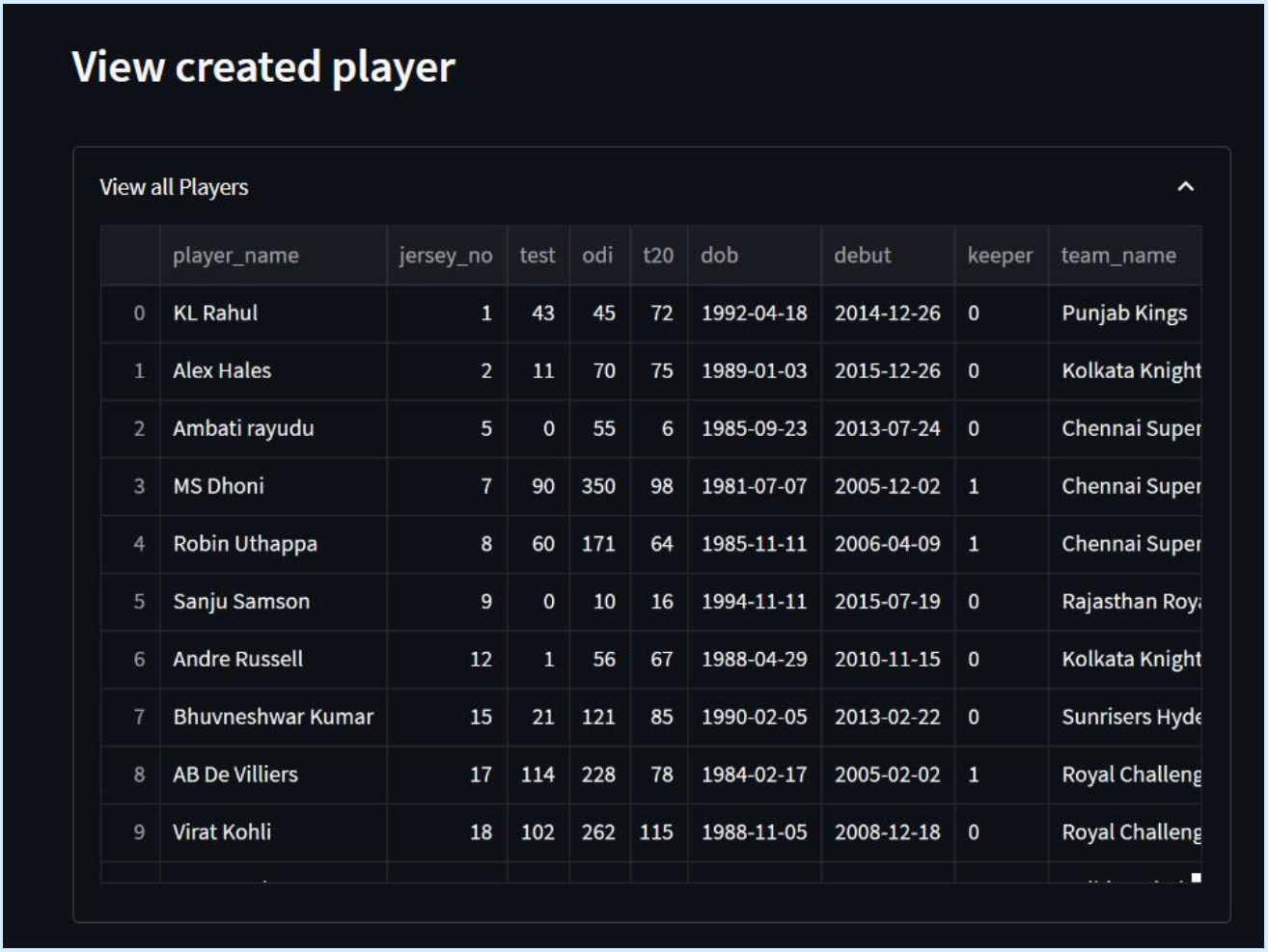
For this application, I have used a python library called Streamlit. With streamlit, it is easy and fast to build attractive websites without any front-end coding experience. The main file app.py contains links to various web pages in the application. There is a file called database.py which essentially holds all the SQL operations that can be used on any table in the database. There are directories for each table in the database which clearly define the front end and its connections to the backend database operations. Here, I will demonstrate the operations on one of the tables – player.

Homepage:

Main page in players.py:

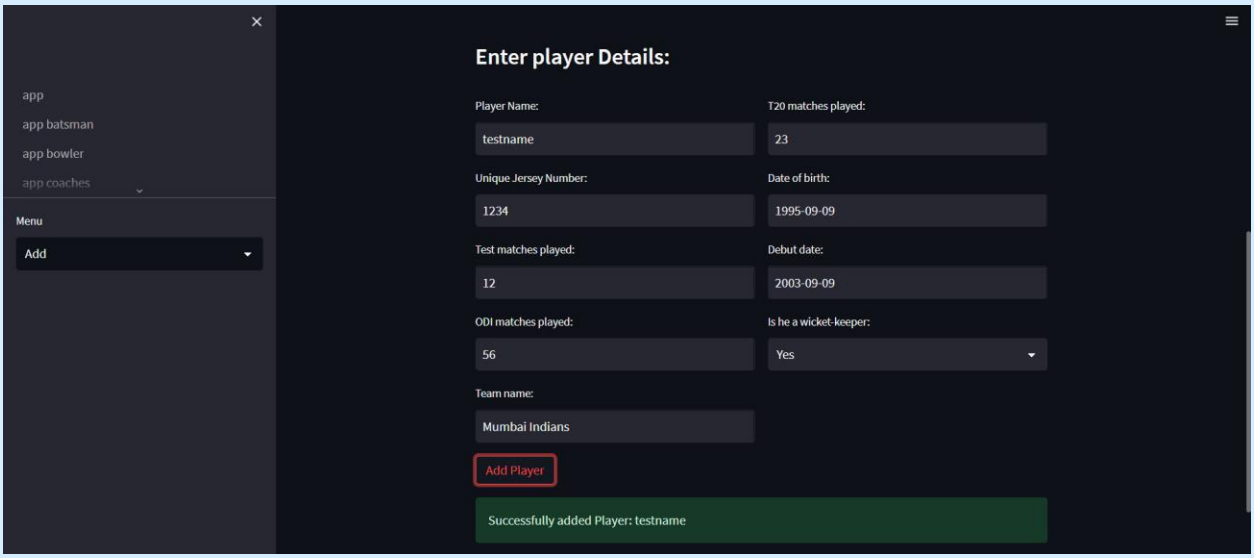


View all players:

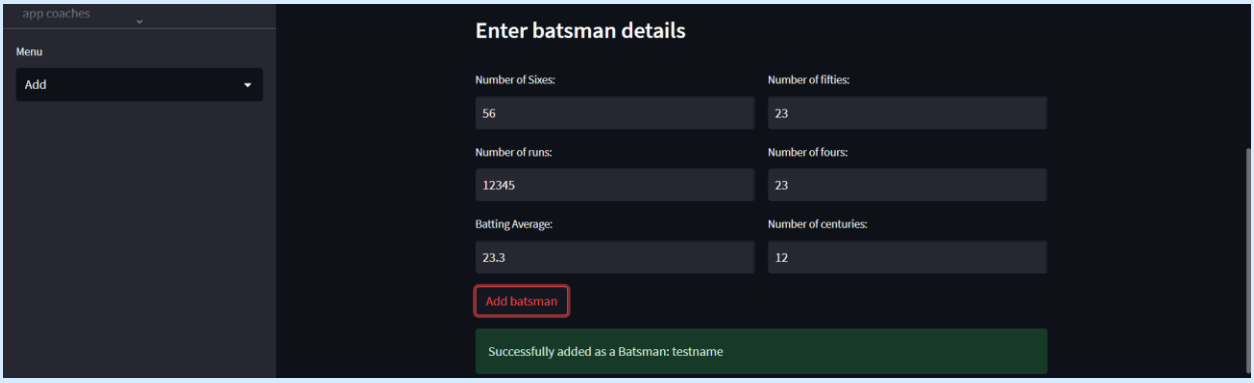


	player_name	jersey_no	test	odi	t20	dob	debut	keeper	team_name
0	KL Rahul	1	43	45	72	1992-04-18	2014-12-26	0	Punjab Kings
1	Alex Hales	2	11	70	75	1989-01-03	2015-12-26	0	Kolkata Knight
2	Ambati rayudu	5	0	55	6	1985-09-23	2013-07-24	0	Chennai Super
3	MS Dhoni	7	90	350	98	1981-07-07	2005-12-02	1	Chennai Super
4	Robin Uthappa	8	60	171	64	1985-11-11	2006-04-09	1	Chennai Super
5	Sanju Samson	9	0	10	16	1994-11-11	2015-07-19	0	Rajasthan Roy
6	Andre Russell	12	1	56	67	1988-04-29	2010-11-15	0	Kolkata Knight
7	Bhuvneshwar Kumar	15	21	121	85	1990-02-05	2013-02-22	0	Sunrisers Hyde
8	AB De Villiers	17	114	228	78	1984-02-17	2005-02-02	1	Royal Challeng
9	Virat Kohli	18	102	262	115	1988-11-05	2008-12-18	0	Royal Challeng

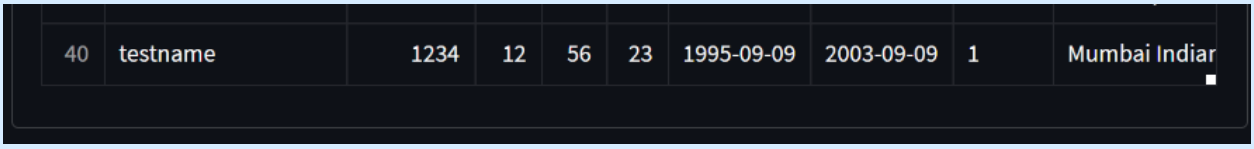
Adding a player named testname



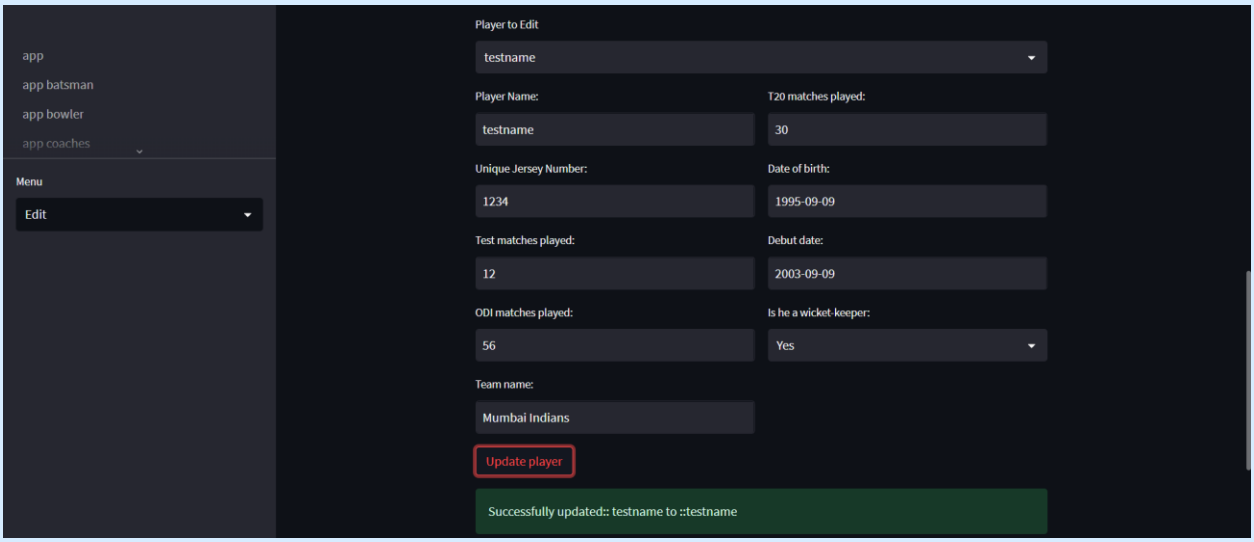
We can simultaneously add batsman or bowler details for the same player:



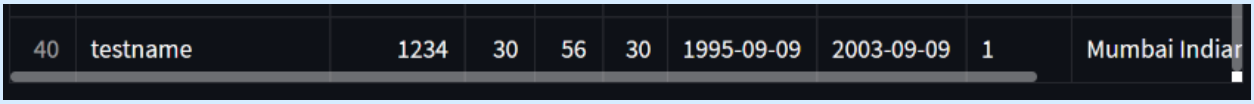
Now, testname also appears in the view players section:



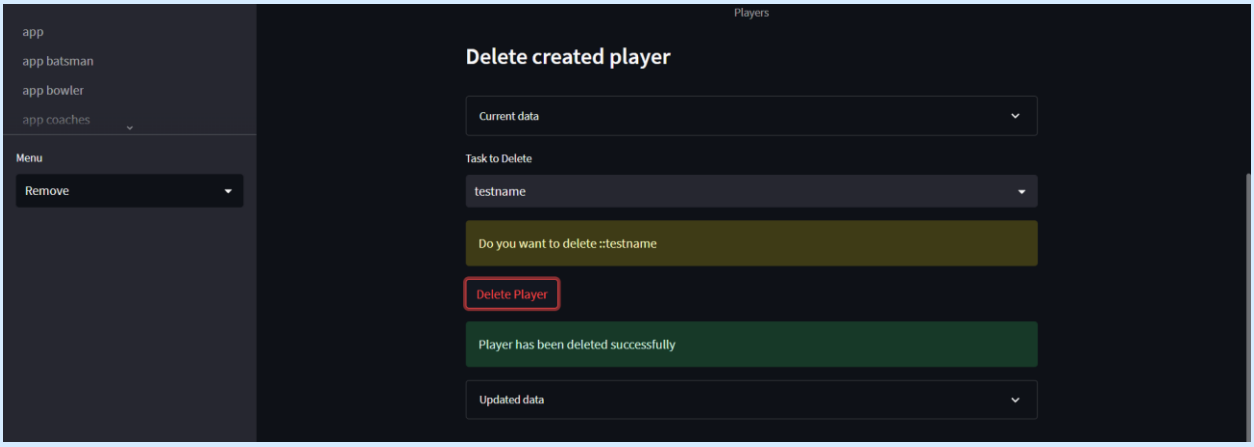
Now let us edit testname details by changing the number of tests to 30:



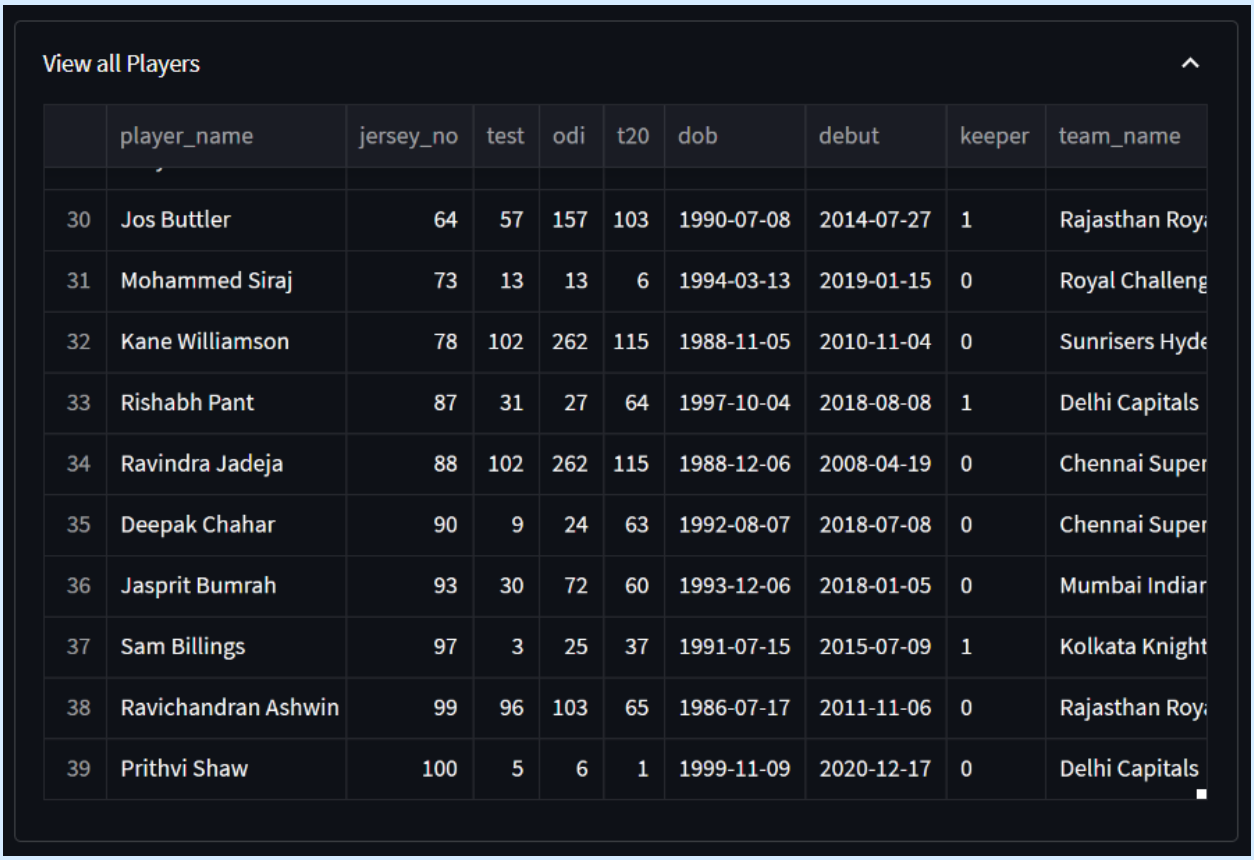
Now testname details have changed:



Now, let us try to delete the testname element in the table by using the delete option in the player menu:



Now, when we view all players, we can see the testname does not exist.



View all Players

	player_name	jersey_no	test	odi	t20	dob	debut	keeper	team_name
30	Jos Buttler	64	57	157	103	1990-07-08	2014-07-27	1	Rajasthan Roy
31	Mohammed Siraj	73	13	13	6	1994-03-13	2019-01-15	0	Royal Challeng
32	Kane Williamson	78	102	262	115	1988-11-05	2010-11-04	0	Sunrisers Hyde
33	Rishabh Pant	87	31	27	64	1997-10-04	2018-08-08	1	Delhi Capitals
34	Ravindra Jadeja	88	102	262	115	1988-12-06	2008-04-19	0	Chennai Super
35	Deepak Chahar	90	9	24	63	1992-08-07	2018-07-08	0	Chennai Super
36	Jasprit Bumrah	93	30	72	60	1993-12-06	2018-01-05	0	Mumbai Indian
37	Sam Billings	97	3	25	37	1991-07-15	2015-07-09	1	Kolkata Knight
38	Ravichandran Ashwin	99	96	103	65	1986-07-17	2011-11-06	0	Rajasthan Roy
39	Prithvi Shaw	100	5	6	1	1999-11-09	2020-12-17	0	Delhi Capitals

Hence, all the CRUD operations have been demonstrated in one of the tables in the database.