Introduction to Database Management Systems

Graduate Project Final-Write-Up

Topic: - Indian Premier League (IPL) Dataset

Submitted by: - Parth Parashar

Source of Data: -
I will be using the below mentioned dataset from Kaggle.com
(***https://www.kaggle.com/datasets/ramjidoolla/ipl-data-set***)

This dataset contains CSV files which contains a lot of data in the form of
1) List of players who played in the IPL from 2008-2019
2) List of teams who play IPL
3) List of all the deliveries ever bowled in the IPL (contains ~1,80,000) entries
4) List of runs, average and strike rates of players
5) List of victories for all the teams along with bifurcation in the form of
Home and away wins
6) List of all the matches ever played in the IPL

Final Write-up: -
The first step towards the completion of this project is copying the required files from my personal desktop to the linux machine.

This is achieved by the scp command which generally is given as: -

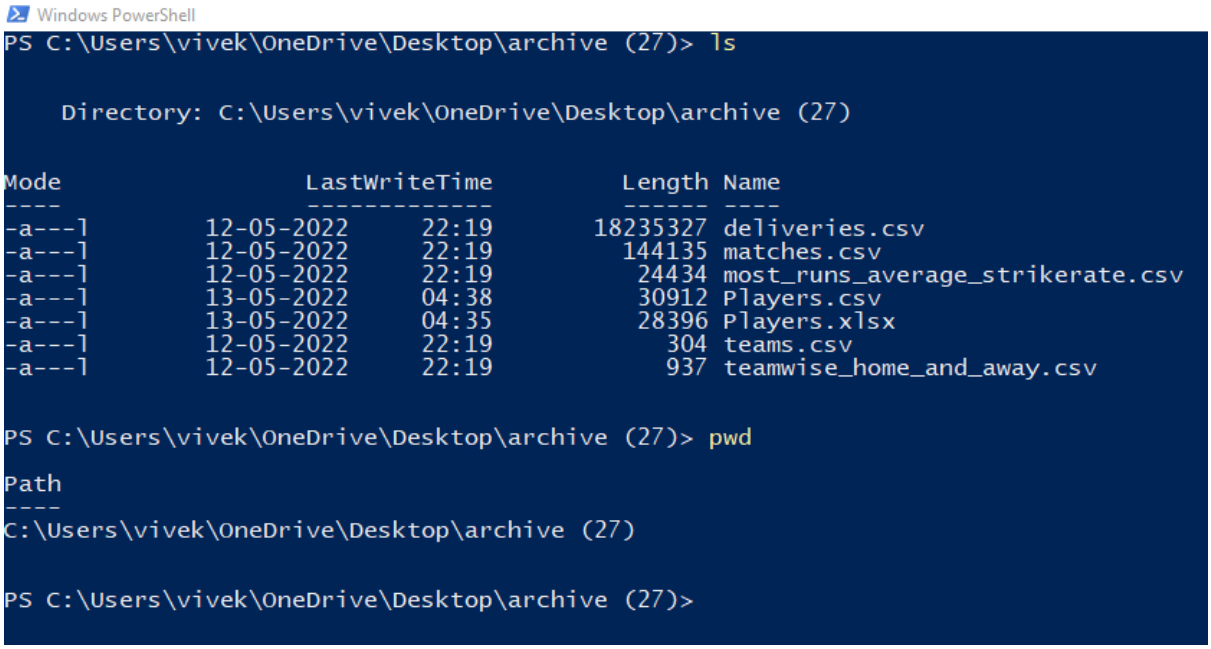
scp <filename> hostname@ipaddress:<path>

In my case, I had to transfer the csv files from my personal laptop to the linux machine which I achieved using the scp command as mentioned below: -

1) Find out the hostname and IP of the target machine

```
parth2@ada:~/Desktop/db_project/files$
parth2@ada:~/Desktop/db_project/files$
parth2@ada:~/Desktop/db_project/files$ hostname -i
131.252.208.103
parth2@ada:~/Desktop/db_project/files$ pwd
/u/parth2/Desktop/db_project/files
parth2@ada:~/Desktop/db_project/files$ hostname
ada.cs.pdx.edu
parth2@ada:~/Desktop/db_project/files$
```

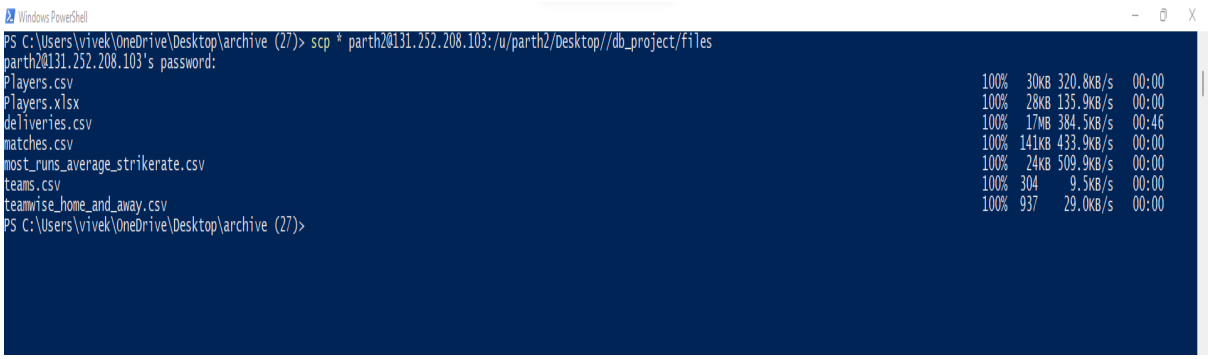
2) Go to the path where files are present on the local computer



3) Write the following SCP command: -

scp * parth2@131.252.208.103:/u/parth2/Desktop//db_project/files

4) After the command in step-3 is run, the files are transferred to the remote linux computer.



After these files have been transferred to the linux computer, we can then begin the process of uploading the data to the database.

For that, we need to create the tables first.

The various data types used are: -

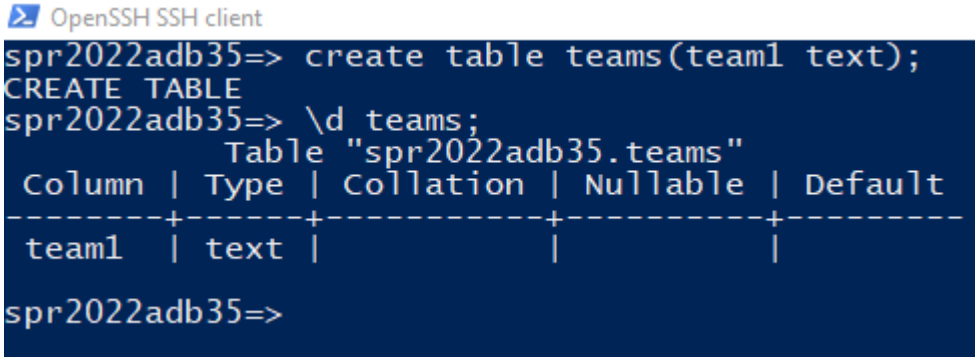
Data Type	Used by
Text	All tables
Date	Players_ipl
Integer	All tables
Double precision	Most_runs_average_strike_rate, matches, deliveries, teamwise_home_and_away

I wanted to use other data types as well but given the type of data, It would not have made sense to add other data types as well.

CREATE TABLE COMMANDS

This can be done using the create commands as mentioned below: -

1) Create command for teams table
   create table teams(team1 text);
   The screenshot for the table creation is given below: -



2) Create command for the table players_ipl;
create table players_ipl(Player_Name text, DOB text, Batting_Hand text, Bowling_Skill text, Country text);

The screenshot for the same is given below: -

```
spr2022adb35=> create table players_ipl(Player_Name text, DOB text, Batting_Hand text, Bowling_Skill text, Country text);
CREATE TABLE
spr2022adb35=> \d players_ipl;
          Table "spr2022adb35.players_ipl"
    Column     | Type | Collation | Nullable | Default
---------------+------+-----------+----------+---------
 player_name   | text |           |          |
 dob           | text |           |          |
 batting_hand  | text |           |          |
 bowling_skill | text |           |          |
 country       | text |           |          |

spr2022adb35=>
```

Now, Since I need dob as a date, I will alter the table with the alter command as given below: -
alter table players_ipl alter column dob type DATE using dob::DATE;

OpenSSH SSH client

```
spr2022adb35=> \d players_ipl;
          Table "spr2022adb35.players_ipl"
    Column     | Type | Collation | Nullable | Default
---------------+------+-----------+----------+---------
 player_name   | text |           | not null |
 dob           | date |           |          |
 batting_hand  | text |           |          |
 bowling_skill | text |           |          |
 country       | text |           |          |
```

3) Create command for the table most_runs_average_strikerate

create table most_runs_average_strikerate(batsman text, total_runs int, out int, numberofballs int , average double precision, strikerate double precision);

The screenshot for the same is given below: -

OpenSSH SSH client

```
spr2022adb35=> create table most_runs_average_strikerate(batsman text, total_runs int, out int, numberofballs int , average double precision, strikerate double precision);
CREATE TABLE
spr2022adb35=> \d most_runs_average_strikerate;
        Table "spr2022adb35.most_runs_average_strikerate"
    Column     |       Type       | Collation | Nullable | Default
---------------+------------------+-----------+----------+---------
 batsman       | text             |           |          |
 total_runs    | integer          |           |          |
 out           | integer          |           |          |
 numberofballs | integer          |           |          |
 average       | double precision |           |          |
 strikerate    | double precision |           |          |

spr2022adb35=>
```

4) Create command for the table matches
create table matches(id int, season text, city text, date text, team1 text, team2 text, toss_winner text, toss_decision text, result text, dl_applied int, winner text, win_by_runs int, win_by_wickets int, player_of_match text, venue text, umpire1 text, umpire2 text, umpire3 text);

The screenshot for the same is given below: -

OpenSSH SSH client

```
spr2022adb35=> create table matches(id int, season text, city text, date text, team1 text,
spr2022adb35(>  team2 text, toss_winner text, toss_decision text, result text,
spr2022adb35(> dl_applied int, winner text, win_by_runs int, win_by_wickets int,
spr2022adb35(> player_of_match text, venue text, umpire1 text, umpire2 text, umpire3 text);
CREATE TABLE
spr2022adb35=> \d matches;
                Table "spr2022adb35.matches"
     Column      |  Type   | Collation | Nullable | Default
-----------------+---------+-----------+----------+---------
 id              | integer |           |          |
 season          | text    |           |          |
 city            | text    |           |          |
 date            | text    |           |          |
 team1           | text    |           |          |
 team2           | text    |           |          |
 toss_winner     | text    |           |          |
 toss_decision   | text    |           |          |
 result          | text    |           |          |
 dl_applied      | integer |           |          |
 winner          | text    |           |          |
 win_by_runs     | integer |           |          |
 win_by_wickets  | integer |           |          |
 player_of_match | text    |           |          |
 venue           | text    |           |          |
 umpire1         | text    |           |          |
 umpire2         | text    |           |          |
 umpire3         | text    |           |          |

spr2022adb35=>
```

Now here, I have used date as a text(string) because the format of the date is not uniform in the table which prevents me from providing a universal date/timestamp format to it.

Also, dl_applied needs to be float which is later changed using alter table command.

alter table matches alter column dl_applied type float using dl_applied::float;

OpenSSH SSH client

```
spr2022adb35=> alter table matches alter column dl_applied type float using dl_applied::float;
ALTER TABLE
spr2022adb35=> \d matches;
                   Table "spr2022adb35.matches"
     Column      |       Type       | Collation | Nullable | Default
-----------------+------------------+-----------+----------+---------
 id              | integer          |           | not null |
 season          | text             |           |          |
 city            | text             |           |          |
 date            | text             |           |          |
 team1           | text             |           |          |
 team2           | text             |           |          |
 toss_winner     | text             |           |          |
 toss_decision   | text             |           |          |
 result          | text             |           |          |
 dl_applied      | double precision |           |          |
 winner          | text             |           |          |
 win_by_runs     | integer          |           |          |
 win_by_wickets  | integer          |           |          |
 player_of_match | text             |           |          |
 venue           | text             |           |          |
 umpire1         | text             |           |          |
 umpire2         | text             |           |          |
 umpire3         | text             |           |          |
Indexes:
    "pkey_matches" PRIMARY KEY, btree (id)

spr2022adb35=>
```

5) Create command for the deliveries table: -

create table deliveries(match_id int, inning int, batting_team text, bowling_team text, over int, ball int, batsman text, non_stricker text, bowler text, is_super_over int, wide_runs int, bye_runs int, legbye_runs int, noball_runs int, penalty_runs int, batsman_runs int, extra_runs int, total_runs int, player_dismissed text, dismissal_kind text, fielder text);

The screenshot for the same is given below: -

OpenSSH SSH client

```
spr2022adb35=> create table deliveries(match_id int, inning int,
spr2022adb35(> batting_team text, bowling_team text, over int, ball int,
spr2022adb35(> batsman text, non_stricker text, bowler text, is_super_over int,
spr2022adb35(> wide_runs int, bye_runs int, legbye_runs int, noball_runs int,
spr2022adb35(> penalty_runs int, batsman_runs int, extra_runs int, total_runs int,
spr2022adb35(> player_dismissed text, dismissal_kind text, fielder text);
CREATE TABLE
spr2022adb35=> \d deliveries;
               Table "spr2022adb35.deliveries"
      Column       |  Type   | Collation | Nullable | Default
-------------------+---------+-----------+----------+---------
 match_id          | integer |           |          |
 inning            | integer |           |          |
 batting_team      | text    |           |          |
 bowling_team      | text    |           |          |
 over              | integer |           |          |
 ball              | integer |           |          |
 batsman           | text    |           |          |
 non_stricker      | text    |           |          |
 bowler            | text    |           |          |
 is_super_over     | integer |           |          |
 wide_runs         | integer |           |          |
 bye_runs          | integer |           |          |
 legbye_runs       | integer |           |          |
 noball_runs       | integer |           |          |
 penalty_runs      | integer |           |          |
 batsman_runs      | integer |           |          |
 extra_runs        | integer |           |          |
 total_runs        | integer |           |          |
 player_dismissed  | text    |           |          |
 dismissal_kind    | text    |           |          |
 fielder           | text    |           |          |

spr2022adb35=>
```

Here also, I need to change the data type of legbye_runs, bye_runs, extra_runs to float (double precision).

This is done by the **alter table** command as given below: -

alter table deliveries alter column legbye_runs type float using legbye_runs::float;
alter table deliveries alter column bye_runs type float using bye_runs::float;
alter table deliveries alter column extra_runs type float using extra_runs::float;

The screenshot for the table is given below: -

```
spr2022adb35=>
spr2022adb35=> alter table deliveries alter column legbye_runs type float using legbye_runs::float;
ALTER TABLE
spr2022adb35=> alter table deliveries alter column bye_runs type float using bye_runs::float;
ALTER TABLE
spr2022adb35=> alter table deliveries alter column extra_runs type float using extra_runs::float;
ALTER TABLE
spr2022adb35=> \d deliveries;
                Table "spr2022adb35.deliveries"
      Column      |       Type       | Collation | Nullable | Default
------------------+------------------+-----------+----------+---------
 match_id         | integer          |           |          |
 inning           | integer          |           |          |
 batting_team     | text             |           |          |
 bowling_team     | text             |           |          |
 over             | integer          |           |          |
 ball             | integer          |           |          |
 batsman          | text             |           |          |
 non_stricker     | text             |           |          |
 bowler           | text             |           |          |
 is_super_over    | integer          |           |          |
 wide_runs        | integer          |           |          |
 bye_runs         | double precision |           |          |
 legbye_runs      | double precision |           |          |
 noball_runs      | integer          |           |          |
 penalty_runs     | integer          |           |          |
 batsman_runs     | integer          |           |          |
 extra_runs       | double precision |           |          |
 total_runs       | integer          |           |          |
 player_dismissed | text             |           |          |
 dismissal_kind   | text             |           |          |
 fielder          | text             |           |          |
```

6) Create command for the table teamwise_home_and_away
create table teamwise_home_and_away(team text,home_wins int, away_wins int, home_matches int, away_matches int, home_win_percentage double precision, away_win_matches double precision);

The screenshot for the same is given below: -

OpenSSH SSH client

```
spr2022adb35=> create table teamwise_home_and_away
spr2022adb35-> (team text,home_wins int, away_wins int,
spr2022adb35(> home_matches int, away_matches int,
spr2022adb35(> home_win_percentage double precision,
spr2022adb35(> away_win_matches double precision);
CREATE TABLE
spr2022adb35=> \d teamwise_home_and_away;
               Table "spr2022adb35.teamwise_home_and_away"
       Column        |       Type       | Collation | Nullable | Default
---------------------+------------------+-----------+----------+---------
 team                | text             |           |          |
 home_wins           | integer          |           |          |
 away_wins           | integer          |           |          |
 home_matches        | integer          |           |          |
 away_matches        | integer          |           |          |
 home_win_percentage | double precision |           |          |
 away_win_matches    | double precision |           |          |

spr2022adb35=>
```

The next step is to add the constraints to the tables created

PRIMARY KEYS: -

1) teams: -

OpenSSH SSH client

```
spr2022adb35=> alter table teams add constraint pkey_team primary key(team1);
ALTER TABLE
spr2022adb35=> \d teams;
           Table "spr2022adb35.teams"
 Column | Type | Collation | Nullable | Default
--------+------+-----------+----------+---------
 team1  | text |           | not null |
Indexes:
    "pkey_team" PRIMARY KEY, btree (team1)

spr2022adb35=>
```

2) players_ipl: -

OpenSSH SSH client

```
spr2022adb35=> \d players_ipl;
            Table "spr2022adb35.players_ipl"
    Column     | Type | Collation | Nullable | Default
---------------+------+-----------+----------+---------
 player_name   | text |           | not null |
 dob           | date |           |          |
 batting_hand  | text |           |          |
 bowling_skill | text |           |          |
 country       | text |           |          |
Indexes:
    "pkey_players_ipl" PRIMARY KEY, btree (player_name)
```

3) most_runs_average_strikerate: -

OpenSSH SSH client

```
spr2022adb35=> \d most_runs_average_strikerate;
         Table "spr2022adb35.most_runs_average_strikerate"
    Column     |       Type       | Collation | Nullable | Default
---------------+------------------+-----------+----------+---------
 batsman       | text             |           | not null |
 total_runs    | integer          |           |          |
 out           | integer          |           |          |
 numberofballs | integer          |           |          |
 average       | double precision |           |          |
 strikerate    | double precision |           |          |
Indexes:
    "pkey_most_runs_average_strikerate" PRIMARY KEY, btree (batsman)
```

4) matches: -

OpenSSH SSH client

```
spr2022adb35=> \d matches;
                  Table "spr2022adb35.matches"
     Column      |       Type        | Collation | Nullable | Default
-----------------+-------------------+-----------+----------+---------
 id              | integer           |           | not null |
 season          | text              |           |          |
 city            | text              |           |          |
 date            | text              |           |          |
 team1           | text              |           |          |
 team2           | text              |           |          |
 toss_winner     | text              |           |          |
 toss_decision   | text              |           |          |
 result          | text              |           |          |
 dl_applied      | double precision  |           |          |
 winner          | text              |           |          |
 win_by_runs     | integer           |           |          |
 win_by_wickets  | integer           |           |          |
 player_of_match | text              |           |          |
 venue           | text              |           |          |
 umpire1         | text              |           |          |
 umpire2         | text              |           |          |
 umpire3         | text              |           |          |
Indexes:
    "pkey_matches" PRIMARY KEY, btree (id)
```

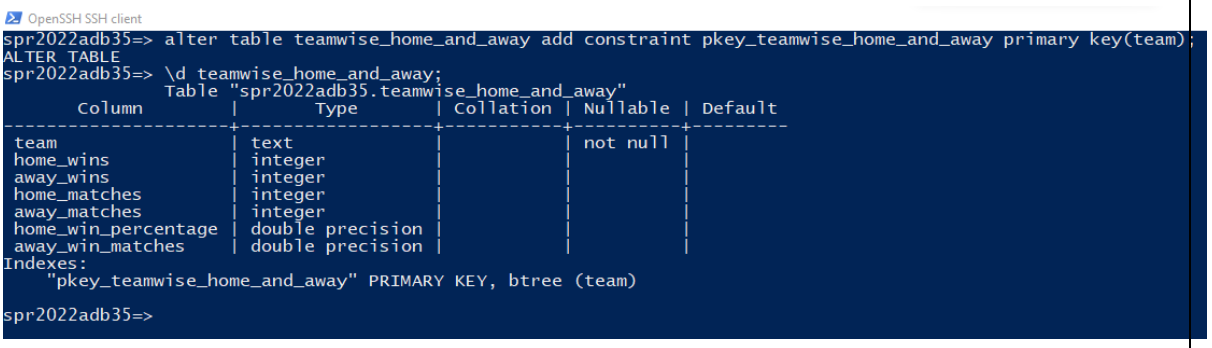
5) deliveries: -

Match_ID can be primary key but there are different columns which can have same matchids. So there is not a definite primary key or a group of keys which can be identified as primary key.

OpenSSH SSH client

```
spr2022adb35=> \d deliveries;
                 Table "spr2022adb35.deliveries"
     Column      |       Type        | Collation | Nullable | Default
-----------------+-------------------+-----------+----------+---------
 match_id        | integer           |           |          |
 inning          | integer           |           |          |
 batting_team    | text              |           |          |
 bowling_team    | text              |           |          |
 over            | integer           |           |          |
 ball            | integer           |           |          |
 batsman         | text              |           |          |
 non_stricker    | text              |           |          |
 bowler          | text              |           |          |
 is_super_over   | integer           |           |          |
 wide_runs       | integer           |           |          |
 bye_runs        | double precision  |           |          |
 legbye_runs     | double precision  |           |          |
 noball_runs     | integer           |           |          |
 penalty_runs    | integer           |           |          |
 batsman_runs    | integer           |           |          |
 extra_runs      | double precision  |           |          |
 total_runs      | integer           |           |          |
 player_dismissed | text             |           |          |
 dismissal_kind  | text              |           |          |
 fielder         | text              |           |          |
```

6) teamwise_home_and_away: -



FOREIGN KEYS: -

The foreign key constraints are updated and given below: -

1) teams and deliveries table (connected by the team_name)

The commands for the same are given below: -

alter table deliveries add constraint fk_team foreign key(batting_team) references teams(team1);

alter table deliveries add constraint fk_bowling_team foreign key (bowling_team) references teams(team1);

2) deliveries and players_ipl (connected by the player name)

The commands for the same are given below: -

alter table deliveries add constraint fk_player_name foreign key(batsman) references players_ipl(player_name);

3) players_ipl and most_runs_avg_strike_rate (connected by player_name)

The command for the same is given below: -

alter table most_runs_average_strikerate add constraint fk_batsman_player_name foreign key(player_name) references players_ipl(player_name);

4) team and team_wise_home_and_away_wins (connected by team_name)

The command for the same is given below: -

alter table teamwise_home_and_away add constraint fk_team_name foreign key(batsman) references teams(team1);

The table structures after the addition of the primary and foreign keys is given below: -

OpenSSH SSH client

```
spr2022adb35=> \d teams;
          Table "spr2022adb35.teams"
 Column | Type | Collation | Nullable | Default
--------+------+-----------+----------+---------
 team1  | text |           | not null |
Indexes:
    "pkey_team" PRIMARY KEY, btree (team1)
Referenced by:
    TABLE "deliveries" CONSTRAINT "fk_bowling_team" FOREIGN KEY (bowling_team) REFERENCES teams(team1)
    TABLE "deliveries" CONSTRAINT "fk_team" FOREIGN KEY (batting_team) REFERENCES teams(team1)
    TABLE "teamwise_home_and_away" CONSTRAINT "fk_team_name" FOREIGN KEY (team) REFERENCES teams(team1)

spr2022adb35=> \d players_ipl;
          Table "spr2022adb35.players_ipl"
    Column     | Type | Collation | Nullable | Default
---------------+------+-----------+----------+---------
 player_name   | text |           | not null |
 dob           | date |           |          |
 batting_hand  | text |           |          |
 bowling_skill | text |           |          |
 country       | text |           |          |
Indexes:
    "pkey_players_ipl" PRIMARY KEY, btree (player_name)
Referenced by:
    TABLE "most_runs_average_strikerate" CONSTRAINT "fk_batsman_player_name" FOREIGN KEY (batsman) REFERENCES players_ipl(player_name)
    TABLE "deliveries" CONSTRAINT "fk_player_name" FOREIGN KEY (batsman) REFERENCES players_ipl(player_name)

spr2022adb35=>
```

OpenSSH SSH client

```
spr2022adb35=> \d deliveries;
                  Table "spr2022adb35.deliveries"
      Column      |       Type        | Collation | Nullable | Default
------------------+-------------------+-----------+----------+---------
 match_id         | integer           |           |          |
 inning           | integer           |           |          |
 batting_team     | text              |           |          |
 bowling_team     | text              |           |          |
 over             | integer           |           |          |
 ball             | integer           |           |          |
 batsman          | text              |           |          |
 non_stricker     | text              |           |          |
 bowler           | text              |           |          |
 is_super_over    | integer           |           |          |
 wide_runs        | integer           |           |          |
 bye_runs         | double precision  |           |          |
 legbye_runs      | double precision  |           |          |
 noball_runs      | integer           |           |          |
 penalty_runs     | integer           |           |          |
 batsman_runs     | integer           |           |          |
 extra_runs       | double precision  |           |          |
 total_runs       | integer           |           |          |
 player_dismissed | text              |           |          |
 dismissal_kind   | text              |           |          |
 fielder          | text              |           |          |
Foreign-key constraints:
    "fk_bowling_team" FOREIGN KEY (bowling_team) REFERENCES teams(team1)
    "fk_player_name" FOREIGN KEY (batsman) REFERENCES players_ipl(player_name)
    "fk_team" FOREIGN KEY (batting_team) REFERENCES teams(team1)

spr2022adb35=>
```

OpenSSH SSH client

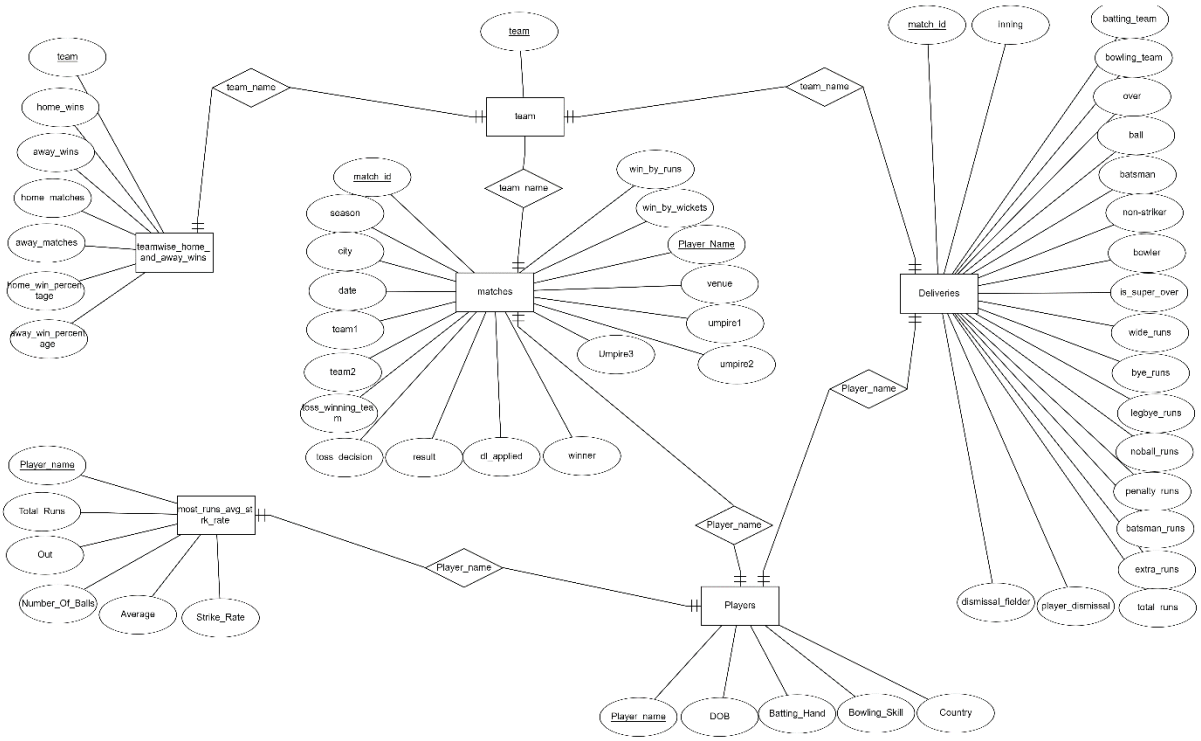
```
spr2022adb35=> \d matches;
                    Table "spr2022adb35.matches"
     Column      |       Type       | Collation | Nullable | Default
-----------------+------------------+-----------+----------+---------
 id              | integer          |           | not null |
 season          | text             |           |          |
 city            | text             |           |          |
 date            | text             |           |          |
 team1           | text             |           |          |
 team2           | text             |           |          |
 toss_winner     | text             |           |          |
 toss_decision   | text             |           |          |
 result          | text             |           |          |
 dl_applied      | double precision |           |          |
 winner          | text             |           |          |
 win_by_runs     | integer          |           |          |
 win_by_wickets  | integer          |           |          |
 player_of_match | text             |           |          |
 venue           | text             |           |          |
 umpire1         | text             |           |          |
 umpire2         | text             |           |          |
 umpire3         | text             |           |          |
Indexes:
    "pkey_matches" PRIMARY KEY, btree (id)

spr2022adb35=> \d most_runs_average_strikerate;
         Table "spr2022adb35.most_runs_average_strikerate"
    Column     |       Type       | Collation | Nullable | Default
---------------+------------------+-----------+----------+---------
 batsman       | text             |           | not null |
 total_runs    | integer          |           |          |
 out           | integer          |           |          |
 numberofballs | integer          |           |          |
 average       | double precision |           |          |
 strikerate    | double precision |           |          |
Indexes:
    "pkey_most_runs_average_strikerate" PRIMARY KEY, btree (batsman)
Foreign-key constraints:
    "fk_batsman_player_name" FOREIGN KEY (batsman) REFERENCES players_ipl(player_name)

spr2022adb35=>
```

```
spr2022adb35=>
spr2022adb35=> \d teamwise_home_and_away;
                Table "spr2022adb35.teamwise_home_and_away"
       Column         |       Type       | Collation | Nullable | Default
----------------------+------------------+-----------+----------+---------
 team                 | text             |           | not null |
 home_wins            | integer          |           |          |
 away_wins            | integer          |           |          |
 home_matches         | integer          |           |          |
 away_matches         | integer          |           |          |
 home_win_percentage  | double precision |           |          |
 away_win_matches     | double precision |           |          |
Indexes:
    "pkey_teamwise_home_and_away" PRIMARY KEY, btree (team)
Foreign-key constraints:
    "fk_team_name" FOREIGN KEY (team) REFERENCES teams(team1)

spr2022adb35=>
```

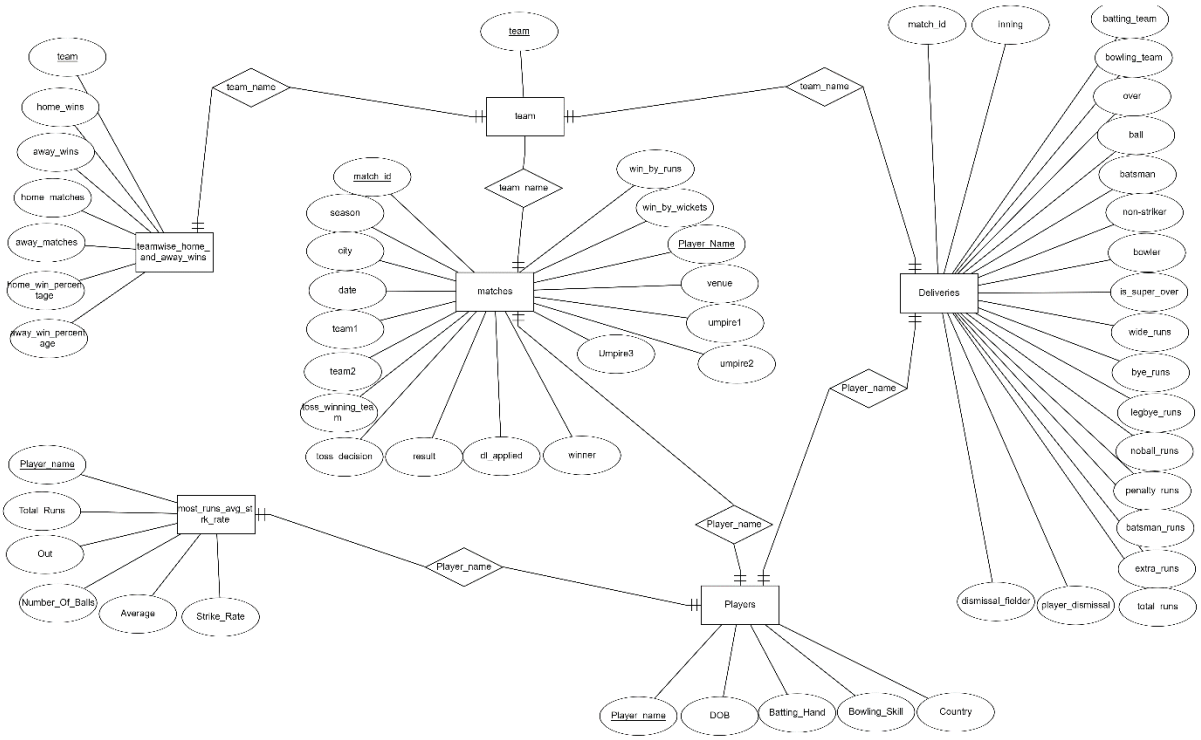
E-R DIAGRAM: -

Since the table deliveries can have multiple rows having same match_id, therefore it cannot be primary key. This is because a single match can have many overs and each over can have multiple deliveries. Therefore, each match_id will be present in multiple rows. I even tried to define a composite key but it does not make any sense to do it with so many columns combining to define the composite key. The ER diagram is changed to: -

OLD: -



NEW: -



DATA ENTRY

After the primary key and foreign key constraints have been added, the next step is to copy the data from the csv files which we had copied earlier to our linux system to the database.

This can be done by a series of copy commands whose outputs have also been displayed in the form of screenshots below. **By using the copy command, I will be avoiding the manual data entries.**

1) teams

The command used is: -

\copy teams from teams.csv with csv header;

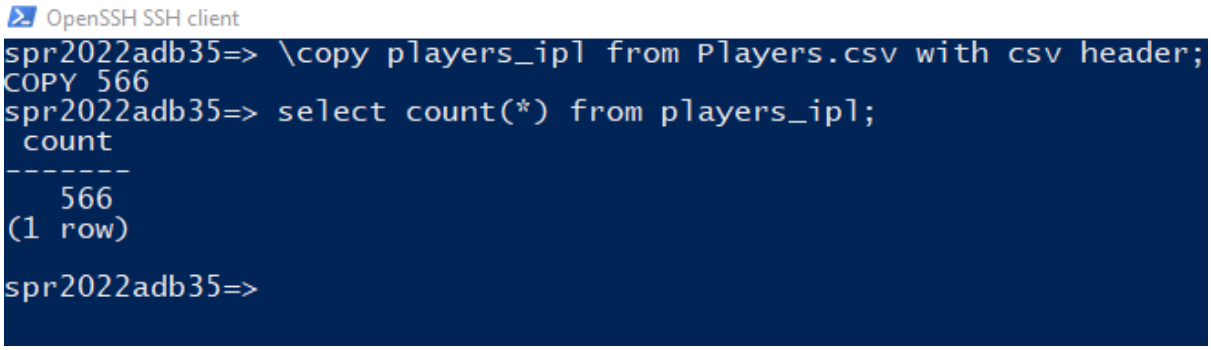
Select OpenSSH SSH client

```
spr2022adb35=> \copy teams from teams.csv with csv header;
COPY 15
spr2022adb35=> select count(*) from teams;
 count
-------
    15
(1 row)

spr2022adb35=>
```

2) players_ipl

The command used is: -

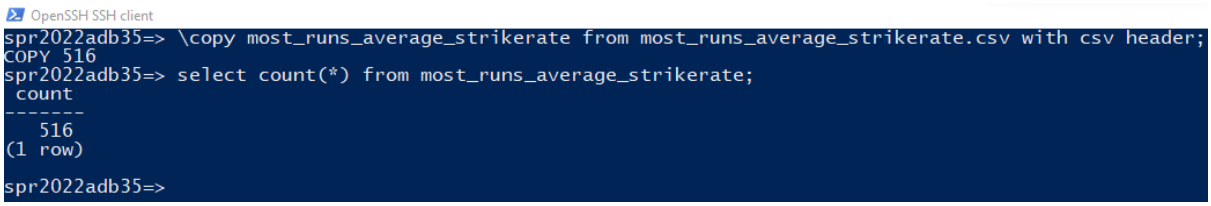
\copy players_ipl from Players.csv with csv header;



3) most_runs_average_strikerate

The command used for copying is: -

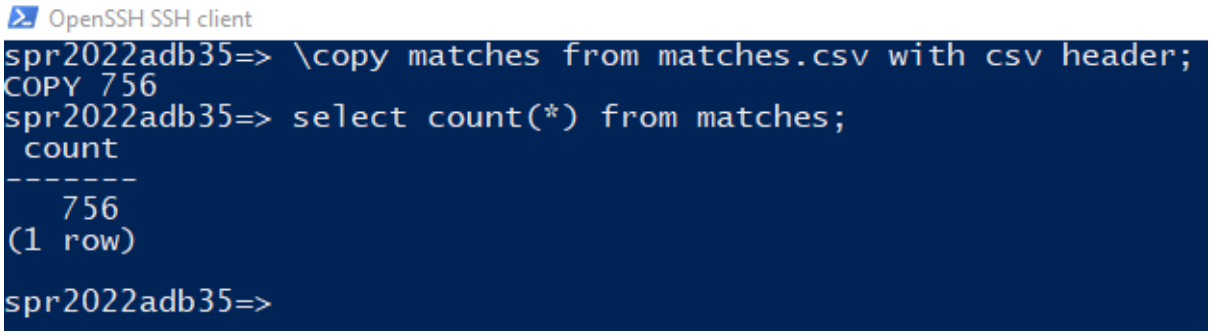
\copy most_runs_average_strikerate from most_runs_average_strikerate.csv with csv header;



4) matches

The command used to copy is given below: -

\copy matches from matches.csv with csv header;



5) deliveries

The command used to copy is given below: -

\copy deliveries from deliveries.csv with csv header;

```
OpenSSH SSH client
spr2022adb35=> \copy deliveries from deliveries.csv with csv header;
COPY 179078
spr2022adb35=> select count(*) from deliveries;
 count
--------
 179078
(1 row)

spr2022adb35=>
```

6) teamwise_home_and_away

The command used to copy is given below: -

\copy teamwise_home_and_away from teamwise_home_and_away.csv with csv header;

```
Select OpenSSH SSH client
spr2022adb35=> \copy teamwise_home_and_away from teamwise_home_and_away.csv with csv header;
COPY 14
spr2022adb35=> select count(*) from teamwise_home_and_away;
 count
-------
    14
(1 row)

spr2022adb35=>
```

QUESTIONS

Q-1) Finding out the batsman with the highest average among all the teams.

Answer: - The table which needs to be used for finding the answer is most_runs_average_strikerate.

The query which yields the result is given below: -
select batsman from most_runs_average_strikerate where average = (select max(average) from most_runs_average_strikerate);

The result is given below in the screenshot.

```
spr2022adb35=> select batsman from most_runs_average_strikerate
where
average = (select max(average) from most_runs_average_strikerate);
    batsman
---------------
 Iqbal Abdulla
(1 row)

spr2022adb35=>
```

Q-2) Finding out the batsman who were bowled on the first delivery of the match.

Answer: - The table which we will use is the deliveries table.

The query which yields the result is given below: -

select player_dismissed from deliveries where over=1 and ball=1 and dismissal_kind='bowled';

The result for this query is given below: -

```
OpenSSH SSH client
spr2022adb35=> select player_dismissed from deliveries where over=1 and ball=1 and dismissal_kind='bowled';
 player_dismissed
------------------
 MA Agarwal
 JD Ryder
 AM Rahane
 JH Kallis
 PC Valthaty
 UBT Chand
 RE Levi
 PA Patel
 AJ Finch
 AS Yadav
 DR Smith
 SR Watson
 S Dhawan
 J Denly
(14 rows)

spr2022adb35=>
```

Q-3) Finding out the bowler, his team name, his dominant bowling hand and strike rate along with the IPL season where he has bowled no more than 100 deliveries and has picked up more than 5 wickets in that season.

Answer: - The tables which we will be using are: - players_ipl and most_runs_average_strikerate.

The query which yields the results is given below: -

select A.player_name, A.bowling_skill,A.batting_hand,B.strikerate from players_ipl A, most_runs_average_strikerate B where A.player_name=B.batsman and B.numberofballs>100 and out>5 limit 10;

Here I have used limit as the result had more than 100 rows which was not possible to show.

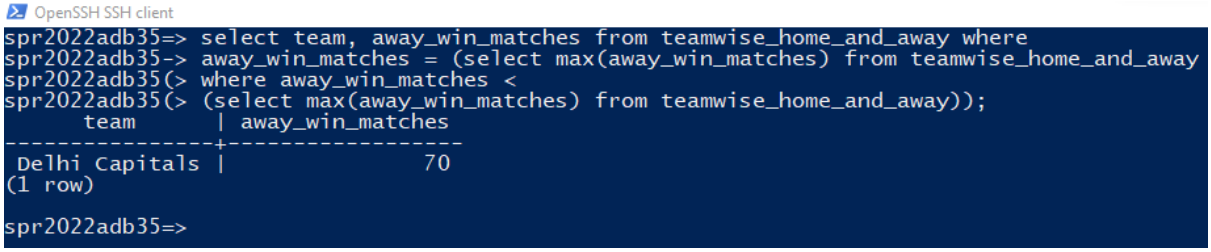
```
spr2022adb35=> select A.player_name, A.bowling_skill,A.batting_hand,B.strikerate from
players_ipl A, most_runs_average_strikerate B
where A.player_name=B.batsman
and B.numberofballs>100
and out>5 limit 10;
  player_name   |   bowling_skill    | batting_hand |     strikerate
----------------+--------------------+--------------+--------------------
 V Kohli        | Right-arm medium   | Right_Hand   | 131.98735100948676
 SK Raina       | Right-arm offbreak | Left_Hand    | 137.53830439223697
 RG Sharma      | Right-arm offbreak | Right_Hand   | 130.99946552645645
 DA Warner      | Legbreak           | Left_Hand    | 143.28675577156744
 S Dhawan       | Right-arm offbreak | Left_Hand    | 125.53888130968622
 CH Gayle       | Right-arm offbreak | Left_Hand    |  152.2543741588156
 MS Dhoni       | Right-arm medium   | Right_Hand   | 138.80224578914536
 RV Uthappa     | Right-arm medium   | Right_Hand   | 130.73055309080155
 AB de Villiers | Right-arm medium   | Right_Hand   | 152.10199862164023
 G Gambhir      | Legbreak           | Left_Hand    | 124.08823529411765
(10 rows)

spr2022adb35=>
```

Q-4) Finding out the team with second highest away win percentage among all the teams to participate.

Answer: - The table that we are going to use is the teamwise_home_and_away.
The query for finding the requisite result is: -

select team, away_win_matches from teamwise_home_and_away where
away_win_matches = (select max(away_win_matches) from teamwise_home_and_away
where away_win_matches <
(select max(away_win_matches) from teamwise_home_and_away));

The screenshot below displays the result of the query



Q-5) Finding out the all the players who are “Right-handed batsman and Right-handed bowlers” and who do not belong to “India”

Answer: - Here we will have to combine the players_ipl and most_runs_average_strike_rate table and find the required result.

The query which provides the required results is: -
select P.player_name, P.batting_hand, P.bowling_skill,P.country, B.total_runs from players_ipl P inner join most_runs_average_strikerate B on P.player_name = B.batsman where P
.batting_hand = 'Right_Hand' and P.bowling_skill='Right-arm medium' and P.country='India';

The screenshot below provides the result of the given query.

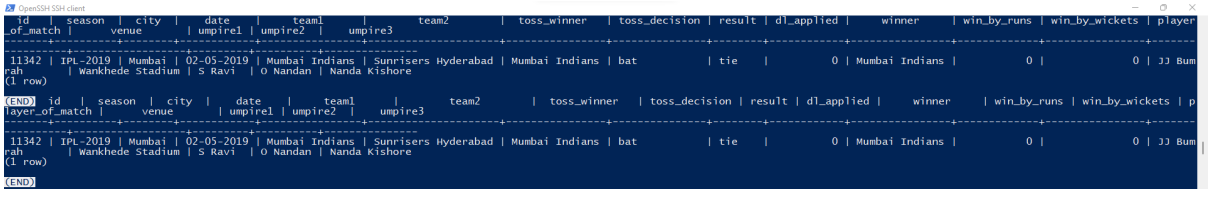
OpenSSH SSH client

```
spr2022adb35=> select P.player_name, P.batting_hand, P.bowling_skill,P.country, B.total_runs
spr2022adb35-> from players_ipl P inner join most_runs_average_strikerate B
spr2022adb35-> on P.player_name = B.batsman where
spr2022adb35-> P.batting_hand = 'Right_Hand'
spr2022adb35-> and P.bowling_skill='Right-arm medium'
spr2022adb35-> and P.country='India';
   player_name     | batting_hand |   bowling_skill    | country | total_runs
-------------------+--------------+--------------------+---------+------------
 V Kohli           | Right_Hand   | Right-arm medium   | India   |       5426
 MS Dhoni          | Right_Hand   | Right-arm medium   | India   |       4450
 RV Uthappa        | Right_Hand   | Right-arm medium   | India   |       4420
 AM Rahane         | Right_Hand   | Right-arm medium   | India   |       3834
 MK Pandey         | Right_Hand   | Right-arm medium   | India   |       2855
 Mandeep Singh     | Right_Hand   | Right-arm medium   | India   |       1535
 M Vohra           | Right_Hand   | Right-arm medium   | India   |       1013
 STR Binny         | Right_Hand   | Right-arm medium   | India   |        880
 V Shankar         | Right_Hand   | Right-arm medium   | India   |        563
 PC Valthaty       | Right_Hand   | Right-arm medium   | India   |        505
 LR Shukla         | Right_Hand   | Right-arm medium   | India   |        405
 P Kumar           | Right_Hand   | Right-arm medium   | India   |        340
 R Vinay Kumar     | Right_Hand   | Right-arm medium   | India   |        310
 A Ashish Reddy    | Right_Hand   | Right-arm medium   | India   |        280
 B Chipli          | Right_Hand   | Right-arm medium   | India   |        280
 R Sathish         | Right_Hand   | Right-arm medium   | India   |        270
 B Kumar           | Right_Hand   | Right-arm medium   | India   |        184
 MM Sharma         | Right_Hand   | Right-arm medium   | India   |        123
 HV Patel          | Right_Hand   | Right-arm medium   | India   |        107
 DS Kulkarni       | Right_Hand   | Right-arm medium   | India   |        101
 MS Gony           | Right_Hand   | Right-arm medium   | India   |         99
 DL Chahar         | Right_Hand   | Right-arm medium   | India   |         71
 AA Bilakhia       | Right_Hand   | Right-arm medium   | India   |         69
 WA Mota           | Right_Hand   | Right-arm medium   | India   |         56
 RV Gomez          | Right_Hand   | Right-arm medium   | India   |         50
 Yashpal Singh     | Right_Hand   | Right-arm medium   | India   |         47
 SK Trivedi        | Right_Hand   | Right-arm medium   | India   |         42
 AUK Pathan        | Right_Hand   | Right-arm medium   | India   |         39
 Anureet Singh     | Right_Hand   | Right-arm medium   | India   |         37
 JJ Bumrah         | Right_Hand   | Right-arm medium   | India   |         37
 SN Thakur         | Right_Hand   | Right-arm medium   | India   |         36
 A Mithun          | Right_Hand   | Right-arm medium   | India   |         34
 Sandeep Sharma    | Right_Hand   | Right-arm medium   | India   |         26
 MA Khote          | Right_Hand   | Right-arm medium   | India   |         24
 R Shukla          | Right_Hand   | Right-arm medium   | India   |         19
 Y Gnaneswara Rao  | Right_Hand   | Right-arm medium   | India   |         19
 AS Rajpoot        | Right_Hand   | Right-arm medium   | India   |         17
 K Upadhyay        | Right_Hand   | Right-arm medium   | India   |         12
 S Kaul            | Right_Hand   | Right-arm medium   | India   |         12
 T Kohli           | Right_Hand   | Right-arm medium   | India   |         11
 Jaskaran Singh    | Right_Hand   | Right-arm medium   | India   |          8
 UA Birla          | Right_Hand   | Right-arm medium   | India   |          7
 VS Malik          | Right_Hand   | Right-arm medium   | India   |          7
 AN Ghosh          | Right_Hand   | Right-arm medium   | India   |          7
 SS Shaikh         | Right_Hand   | Right-arm medium   | India   |          6
 P Awana           | Right_Hand   | Right-arm medium   | India   |          5
 RS Sodhi          | Right_Hand   | Right-arm medium   | India   |          4
 S Tyagi           | Right_Hand   | Right-arm medium   | India   |          3
 RG More           | Right_Hand   | Right-arm medium   | India   |          2
 PM Sarvesh Kumar  | Right_Hand   | Right-arm medium   | India   |          1
 DP Vijaykumar     | Right_Hand   | Right-arm medium   | India   |          1
 DJ Muthuswami     | Right_Hand   | Right-arm medium   | India   |          1
 RR Bhatkal        | Right_Hand   | Right-arm medium   | India   |          0
 IC Pandey         | Right_Hand   | Right-arm medium   | India   |          0
 V Pratap Singh    | Right_Hand   | Right-arm medium   | India   |          0
```

The total rows retrieved are: - 55 rows

Q-6) Finding out all the details of the matches which ended in a tie played at the “Wankhede stadium”.

Answer: - select * from matches where venue = 'Wankhede Stadium' and result = 'tie';
The result of the query is given in the screenshot below: -



Q-7) Finding out the DOB, batting average and strike-rate of “V Kohli”?

Answer: - The query used to find out the required result is: -
select A.player_name, A.dob, B.average, B.strikerate from players_ipl A inner join most_runs_average_strikerate B on A.player_name = B.batsman where A.player_name = 'V Kohli';

The screenshot for the same is given below: -

```
spr2022adb35=> select A.player_name, A.dob, B.average, B.strikerate from
spr2022adb35-> players_ipl A inner join most_runs_average_strikerate B
spr2022adb35-> on A.player_name = B.batsman
spr2022adb35-> where
spr2022adb35-> A.player_name = 'V Kohli';
 player_name |    dob     |      average      |     strikerate
-------------+------------+-------------------+--------------------
 V Kohli     | 1988-11-05 | 35.69736842105263 | 131.98735100948676
(1 row)

spr2022adb35=>
```

Q-8) Finding out the details of all the matches where man-of-match was “R G Sharma” when batting second and his team won by a margin of at least 1 wicket.

Answer: - The query used to find the required result is: -
(We need to use both matches and deliveries table to find out the required results)
select distinct B.match_id,B.inning, A.player_of_match,A.win_by_wickets from matches A inner join deliveries B on A.id = B.match_id and A.player_of_match = 'RG Sharma' and A.win_by_wickets>1 and B.inning=2;
The result of this query is given below: -

OpenSSH SSH client

```
spr2022adb35=> select distinct B.match_id,B.inning, A.player_of_match,A.win_by_wickets from
matches A inner join deliveries B
on A.id = B.match_id and
A.player_of_match = 'RG Sharma' and
A.win_by_wickets>1 and B.inning=2;
 match_id | inning | player_of_match | win_by_wickets
----------+--------+-----------------+----------------
       37 |      2 | RG Sharma       |              5
      163 |      2 | RG Sharma       |              6
      225 |      2 | RG Sharma       |              5
      316 |      2 | RG Sharma       |              5
      581 |      2 | RG Sharma       |              6
      590 |      2 | RG Sharma       |              6
      600 |      2 | RG Sharma       |              6
      605 |      2 | RG Sharma       |              8
     7920 |      2 | RG Sharma       |              8
(9 rows)
```

Q-9) Finding out all the matches where player “S Dhawan” played the first ball in first inning and team won the match by a margin of more than 1 run.

Answer: - Here we need to combine both the deliveries and matches tables
The query used to find out the required result is given below: -
select B.match_id, A.winner,B.batsman from matches A inner join deliveries B on A.id = B.match_id where B.batsman = 'S Dhawan' and B.inning=1 and B.over=1 and B.ball=1 and A.win_by_runs>1;

The result is shown in the screenshot below: -

```
spr2022adb35=> select B.match_id, A.winner,B.batsman from
matches A inner join deliveries B
on A.id = B.match_id
where B.batsman = 'S Dhawan' and B.inning=1 and
B.over=1 and B.ball=1 and A.win_by_runs>1;
 match_id |       winner         | batsman
----------+----------------------+----------
      199 | Mumbai Indians       | S Dhawan
      487 | Sunrisers Hyderabad  | S Dhawan
      376 | Deccan Chargers      | S Dhawan
      207 | Mumbai Indians       | S Dhawan
      231 | Mumbai Indians       | S Dhawan
      300 | Deccan Chargers      | S Dhawan
      469 | Sunrisers Hyderabad  | S Dhawan
      537 | Sunrisers Hyderabad  | S Dhawan
      558 | Sunrisers Hyderabad  | S Dhawan
      561 | Sunrisers Hyderabad  | S Dhawan
     7916 | Sunrisers Hyderabad  | S Dhawan
     7918 | Sunrisers Hyderabad  | S Dhawan
(12 rows)
```

Q-10) Finding the player(s) and their number of “players of match” awards who belong to “India” and play for “Mumbai Indians” in the IPL.

Answer: - The query used to find out the result is: -
select A.player_of_match, count(A.player_of_match) from matches A inner join players_ipl B on A.player_of_match = B.player_name where B.country = 'India'and A.winner='Mumbai Indians' group by A.player_of_match;

The result is given in the screenshot provided below: -

OpenSSH SSH client

```
spr2022adb35=> select A.player_of_match, count(A.player_of_match) from
spr2022adb35-> matches A inner join players_ipl B
spr2022adb35-> on A.player_of_match = B.player_name
spr2022adb35-> where B.country = 'India'and A.winner='Mumbai Indians'
spr2022adb35-> group by A.player_of_match;
 player_of_match | count
-----------------+-------
 AT Rayudu       |     7
 PA Patel        |     1
 AP Tare         |     1
 RG Sharma       |    14
 JJ Bumrah       |     5
 HH Pandya       |     6
 Ishan Kishan    |     1
 R Sharma        |     1
 SR Tendulkar    |     8
 KH Pandya       |     4
 MM Patel        |     2
 Harbhajan Singh |     6
 N Rana          |     2
 KD Karthik      |     1
 KV Sharma       |     1
 A Nehra         |     1
 YK Pathan       |     1
(17 rows)

spr2022adb35=>
```

Q-11) Finding out the list of team(s) who have away-win-percentage of more than 60% and have won at least one match at the “M Chinnaswamy Stadium”

Answer: -The query used for finding out the required result is given below: -
select A.team, A.away_win_matches from teamwise_home_and_away A join matches B on A.team = B.winner and A.away_win_matches>60 and B.venue='M. Chinnaswamy Stadium';

The screenshot providing the result is given below: -

OpenSSH SSH client

```
spr2022adb35=> select A.team, A.away_win_matches
from
teamwise_home_and_away A join matches B
on
A.team = B.winner and A.away_win_matches>60
and B.venue='M. Chinnaswamy Stadium';
          team           | away_win_matches
-------------------------+------------------
 Kolkata Knight Riders | 61.05263157894737
 Delhi Capitals        |                70
(2 rows)
```

Q-12) Finding right-handed batsmen with average of more than 30 and has won 3 or more man-of-match awards.

Answer: - For getting the result to this query, we will need to create a view and then write a second query to get the required results.

The query used to create the view is given below: -
create view question12 as select B.batsman, B.average, count(A.player_of_match) as player_of_match_awards
from matches A join most_runs_average_strikerate B on A.player_of_match = B.batsman where B.average > 30 group by A.player_of_match, B.batsman;

The result of this query is given below: -

OpenSSH SSH client

```
spr2022adb35=> create view question12 as
spr2022adb35-> select B.batsman, B.average, count(A.player_of_match) as player_of_match_awards
spr2022adb35-> from
spr2022adb35-> matches A join most_runs_average_strikerate B
spr2022adb35-> on A.player_of_match = B.batsman
spr2022adb35-> where B.average > 30
spr2022adb35-> group by A.player_of_match, B.batsman;
CREATE VIEW
spr2022adb35=>
```

```
spr2022adb35=> select * from question12;
    batsman      |       average        | player_of_match_awards
-----------------+----------------------+------------------------
 V Kohli         |  35.69736842105263   |                     12
 Q de Kock       | 32.130434782608695   |                      1
 RR Pant         |  36.06122448979592   |                      6
 MEK Hussey      |  38.01923076923077   |                     12
 ML Hayden       |                   41 |                      4
 JC Buttler      | 37.729729729729726   |                      5
 LMP Simmons     |  39.96296296296296   |                      3
 G Gambhir       |  31.48507462686567   |                     13
 S Badrinath     | 30.020833333333332   |                      1
 DA Warner       |  41.37719298245614   |                     17
 SR Watson       | 31.217391304347824   |                     15
 MP Stoinis      | 31.733333333333334   |                      2
 SR Tendulkar    |  32.87323943661972   |                      8
 CA Lynn         | 34.810810810810814   |                      4
 AD Russell      |  34.51219512195122   |                     11
 CH Gayle        |  41.13636363636363   |                     21
 KP Pietersen    |                35.75 |                      3
 JP Duminy       |  41.44897959183673   |                      4
 DA Miller       | 34.407407407407405   |                      3
 SPD Smith       |  34.54237288135593   |                      5
 SK Raina        |              33.6625 |                     14
 Iqbal Abdulla   |                   88 |                      2
 J Bairstow      |              57.375  |                      2
 AB de Villiers  |  42.44230769230769   |                     20
 S Gill          |  36.42857142857143   |                      1
 PD Collingwood  |              50.75   |                      1
 HM Amla         |  44.38461538461539   |                      2
 AM Rahane       |  32.76923076923077   |                     12
 KS Williamson   |                38.5  |                      3
 BJ Hodge        | 30.434782608695652   |                      6
 JJ Roy          |                36.6  |                      1
 KL Rahul        |  40.57142857142857   |                      4
 AC Voges        | 30.166666666666668   |                      1
 MK Pandey       |  30.05263157894737   |                      4
 SE Marsh        | 38.292307692307695   |                      9
 S Dhawan        | 33.583941605839414   |                      6
 RG Sharma       |   30.4472049689441   |                     17
 MS Dhoni        |  37.71186440677966   |                     17
 A Symonds       |  37.46153846153846   |                      3
 F du Plessis    | 33.232142857142854   |                      4
(40 rows)

spr2022adb35=>
```

Now after getting this view, we will query this view to get the required result.
select * from question12 where player_of_match_awards > 3;

```
spr2022adb35=> select * from question12 where player_of_match_awards > 3;
    batsman     |      average       | player_of_match_awards
----------------+--------------------+------------------------
 V Kohli        |  35.69736842105263 |                     12
 RR Pant        |  36.06122448979592 |                      6
 MEK Hussey     |  38.01923076923077 |                     12
 ML Hayden      |                 41 |                      4
 JC Buttler     | 37.729729729729726 |                      5
 G Gambhir      |  31.48507462686567 |                     13
 DA Warner      |  41.37719298245614 |                     17
 SR Watson      | 31.217391304347824 |                     15
 SR Tendulkar   |  32.87323943661972 |                      8
 CA Lynn        | 34.810810810810814 |                      4
 AD Russell     |  34.51219512195122 |                     11
 CH Gayle       |  41.13636363636363 |                     21
 JP Duminy      |  41.44897959183673 |                      4
 SPD Smith      |  34.54237288135593 |                      5
 SK Raina       |             33.6625 |                     14
 AB de Villiers |  42.44230769230769 |                     20
 AM Rahane      |  32.76923076923077 |                     12
 BJ Hodge       | 30.434782608695652 |                      6
 KL Rahul       |  40.57142857142857 |                      4
 MK Pandey      |  30.05263157894737 |                      4
 SE Marsh       | 38.292307692307695 |                      9
 S Dhawan       | 33.583941605839414 |                      6
 RG Sharma      |   30.4472049689441 |                     17
 MS Dhoni       |  37.71186440677966 |                     17
 F du Plessis   | 33.232142857142854 |                      4
(25 rows)
```

Q-13) Finding the bowler(s), his average, his strike rate and his team's name who has bowled the most and least number of deliveries to "V Kohli"?

Answer: -Here also we will require to create a view where we will store the number of balls bowled by each bowler to V Kohli.
Then using this we will find out the max and min number of bowls to Virat Kohli.

The query to create the view is given below: -
create view question13 as
select bowling_team, bowler, count(ball) as balls_bowled from deliveries where batsman='V Kohli' group by bowling_team,bowler;

The screenshot of the view created is given below: -

```
spr2022adb35=> create view question13 as
spr2022adb35-> select bowling_team, bowler, count(ball) as balls_bowled
spr2022adb35-> from deliveries
spr2022adb35-> where batsman='V kohli'
spr2022adb35-> group by bowling_team,bowler;
CREATE VIEW
```

```
spr2022adb35=> select count(*) from question13;
 count
-------
   316
(1 row)

spr2022adb35=> select * from question13 limit 10;
    bowling_team     |      bowler      | balls_bowled
---------------------+------------------+--------------
 Chennai Super Kings | A Flintoff       |            4
 Chennai Super Kings | A Nehra          |           28
 Chennai Super Kings | C Ganapathy      |            2
 Chennai Super Kings | CH Morris        |           16
 Chennai Super Kings | DE Bollinger     |           24
 Chennai Super Kings | DJ Bravo         |           54
 Chennai Super Kings | DL Chahar        |           20
 Chennai Super Kings | DP Nannes        |            2
 Chennai Super Kings | D Willey         |            4
 Chennai Super Kings | Harbhajan Singh  |            6
(10 rows)

spr2022adb35=>
```

The next step is to combine the question13 view with the most_runs_average_strike_rate table and find out the required result.

The query which gives the final result is: -

select A.bowler, B.average,B.strikerate,A.bowling_team,A.balls_bowled from question13 A join most_runs_average_strikerate B on A.bowler = B.batsman where A.balls_bowled=(select min(balls_bowled) from question13) or A.balls_bowled = (select max(balls_bowled) from question13) order by A.balls_bowled asc;

The screenshot for the result is given below: -

```
OpenSSH SSH client
spr2022adb35=> select A.bowler, B.average,B.strikerate,A.bowling_team,A.balls_bowled
spr2022adb35-> from
spr2022adb35-> question13 A join most_runs_average_strikerate B
spr2022adb35-> on
spr2022adb35-> A.bowler = B.batsman
spr2022adb35-> where A.balls_bowled=(select min(balls_bowled) from question13)
spr2022adb35-> or
spr2022adb35-> A.balls_bowled = (select max(balls_bowled) from question13)
spr2022adb35-> order by A.balls_bowled asc;
     bowler      |      average       |     strikerate     |     bowling_team      | balls_bowled
-----------------+--------------------+--------------------+-----------------------+--------------
 BAW Mendis      |                1.5 |                 50 | Pune Warriors         |            1
 GJ Maxwell      | 22.596774193548388 | 162.15277777777777 | Delhi Daredevils      |            1
 Y Venugopal Rao | 23.452380952380953 | 118.10551558752998 | Sunrisers Hyderabad   |            1
 V Sehwag        | 27.836734693877553 | 156.24284077892327 | Delhi Daredevils      |            1
 TL Suman        | 22.533333333333335 | 117.97556719022688 | Pune Warriors         |            1
 A Mishra        | 12.620689655172415 |  95.06493506493506 | Delhi Capitals        |            1
 LA Carseldine   |                 27 | 119.11764705882352 | Rajasthan Royals      |            1
 CR Woakes       |               12.6 |  95.45454545454545 | Kolkata Knight Riders |            1
 NM Coulter-Nile |  7.428571428571429 |                104 | Kolkata Knight Riders |            1
 JO Holder       |                7.6 | 122.58064516129032 | Chennai Super Kings   |            1
 AN Ahmed        |                 36 | 138.46153846153845 | Mumbai Indians        |            1
 DAJ Bracewell   |                    | 133.33333333333331 | Delhi Daredevils      |            1
 AG Murtaza      |                 10 |  71.42857142857143 | Pune Warriors         |            1
 SP Narine       |             19.625 | 171.39737991266375 | Kolkata Knight Riders |           94
(14 rows)

spr2022adb35=>
spr2022adb35=>
```

Q-14) Finding team(s) who won on “Feroz Shah Kotla” stadium while they decided to field first.

Answer: - The query which would yield the required results is given below: -
select distinct winner from matches where venue = 'Feroz Shah Kotla Ground'and win_by_runs is not null;

The screenshot of the result is given below: -

```
spr2022adb35=> select distinct winner from matches
spr2022adb35-> where
spr2022adb35-> venue = 'Feroz Shah Kotla Ground'
spr2022adb35-> and win_by_runs is not null;
       winner
--------------------
 Chennai Super Kings
 Delhi Capitals
 Mumbai Indians
 Sunrisers Hyderabad
(4 rows)

spr2022adb35=>
```

Q-15) Which batsman has bowled 1st ball of the 1st inning and has taken wickets in the 2nd over as well.

Answer: - Now, for a batsman bowling, the strike rate is more than 100.
So, for this, we will need to use two tables namely most_runs_average_strike_rate and deliveries.

The query which will yield the required result is given below: -
select distinct B.bowler from most_runs_average_strikerate A join deliveries B on A.batsman = B.bowler and A.strikerate>100 and over=2 and ball=1 and dismissal_kind is not null;

The screenshot below provides the result of the query.

OpenSSH SSH client

```
spr2022adb35=> select distinct B.bowler from most_runs_average_strikerate A
join
deliveries B on A.batsman = B.bowler
and A.strikerate>100 and over=2 and ball=1 and dismissal_kind is not null;
     bowler
----------------
 CH Morris
 DL Chahar
 IK Pathan
 Iqbal Abdulla
 JA Morkel
 JP Duminy
 JP Faulkner
 MC Henriques
 MJ McClenaghan
 P Kumar
 S Aravind
 WPUJC Vaas
 YK Pathan
(13 rows)

spr2022adb35=>
```

Q-16) Finding out all the matches where “Royal challengers Bangalore” lost in the second innings and the losing margin was more than 10 runs.

Answer: - The table which will be used to yield the required result is matches.
The query for the same is given below: -
select id as matchid, season from matches where winner!='Royal Challengers Bangalore' and win_by_runs>10;
This query yields 230 rows.
The screenshot depicting the result of the query with a limit of 10 on the rows is given: -

OpenSSH SSH client

```
spr2022adb35=> select id as matchid, season from matches where winner!='Royal Challengers Bangalore' and win_by_runs>10 limit 10;
 matchid |  season
---------+---------
       1 | IPL-2017
       9 | IPL-2017
      14 | IPL-2017
      15 | IPL-2017
      17 | IPL-2017
      21 | IPL-2017
      24 | IPL-2017
      26 | IPL-2017
      27 | IPL-2017
      32 | IPL-2017
(10 rows)
```

Q-17) Finding out the team and its players of winner of IPL- season 2017.

Answer: - The winner of a season is the team which wins the most number of matches. Also, player of the match is always chosen from the winning team.
So we have to combine 2 tables to find out the required result. The tables being combined are players_ipl and matches

The query yielding the required result is given below: -
select A.winner, A.season ,B.player_name
from
matches A join players_ipl B
on
A.player_of_match = B.player_name
where A.winner =

(select Y.winner from (select winner, count(winner) from matches where season='IPL-2017' group by winner)Y where count=(select max(X.count) from (select winner,count(winner) from matches where season='IPL-2017'group by winner)X))
and A.season='IPL-2017';

The results of this query are given below: -

```
OpenSSH SSH client
spr2022adb35=> select A.winner, A.season ,B.player_name
spr2022adb35-> from
spr2022adb35-> matches A join players_ipl B
spr2022adb35-> on
spr2022adb35-> A.player_of_match = B.player_name
spr2022adb35-> where A.winner =
spr2022adb35-> (select Y.winner from (select winner, count(winner) from matches where season='IPL-2017' group by winner)Y
spr2022adb35(> where count=(select max(X.count) from (select winner,count(winner) from matches where season='IPL-2017'group by winner)X))
spr2022adb35-> and A.season='IPL-2017';
     winner      |  season  |  player_name
-----------------+----------+---------------
 Mumbai Indians | IPL-2017 | AT Rayudu
 Mumbai Indians | IPL-2017 | JC Buttler
 Mumbai Indians | IPL-2017 | JJ Bumrah
 Mumbai Indians | IPL-2017 | KA Pollard
 Mumbai Indians | IPL-2017 | KH Pandya
 Mumbai Indians | IPL-2017 | KH Pandya
 Mumbai Indians | IPL-2017 | KV Sharma
 Mumbai Indians | IPL-2017 | LMP Simmons
 Mumbai Indians | IPL-2017 | MJ McClenaghan
 Mumbai Indians | IPL-2017 | N Rana
 Mumbai Indians | IPL-2017 | N Rana
 Mumbai Indians | IPL-2017 | RG Sharma
(12 rows)

spr2022adb35=>
```

Q-18) Finding the team(s) which has won the highest and second highest number of matches from IPL 2008-2012 while fielding first.

Answer: - The query for finding out the desired result is given below: -

Here, we will make use of the **union operator** to find the results and then display them The trick being removing the season from which the highest number of matches has been won by the resultant team.

select Y.winner from (select winner, count(winner) from matches where season in ('IPL-2008','IPL-2009','IPL-2010','IPL-2011','IPL-2012') group by winner)Y
where count in
(select max(X.count) from (select winner,count(winner)
from matches where season in ('IPL-2008','IPL-2009','IPL-2010','IPL-2011','IPL-2012')
group by winner)X)
union
select Y.winner from (select winner, count(winner) from matches where season in ('IPL-2008','IPL-2009','IPL-2010','IPL-2011','IPL-2012') group by winner)Y
where count in (select max(X.count) from (select winner,count(winner)
from matches where season in ('IPL-2009','IPL-2010','IPL-2011','IPL-2012')
group by winner)X);

The result of this query is shown below: -

```
OpenSSH SSH client
spr2022adb35=> select Y.winner from
spr2022adb35-> (select winner, count(winner) from matches
spr2022adb35(> where season in ('IPL-2008','IPL-2009','IPL-2010','IPL-2011','IPL-2012')
spr2022adb35(> group by winner)Y where count in
spr2022adb35-> (select max(X.count) from (select winner,count(winner)
spr2022adb35(> from matches where season in ('IPL-2008','IPL-2009','IPL-2010','IPL-2011','IPL-2012') group by winner)X)
spr2022adb35->  union
spr2022adb35-> select Y.winner from (select winner, count(winner) from matches where season in
spr2022adb35(> ('IPL-2008','IPL-2009','IPL-2010','IPL-2011','IPL-2012')
spr2022adb35(> group by winner)Y
spr2022adb35-> where count in
spr2022adb35-> (select max(X.count) from
spr2022adb35(> (select winner,count(winner) from matches where season in ('IPL-2009','IPL-2010','IPL-2011','IPL-2012')
spr2022adb35(> group by winner)X);
       winner
--------------------
 Chennai Super Kings
 Rajasthan Royals
(2 rows)
```

Q-19) Finding out the player(s) who played for “Sunrisers Hyderabad” and has the lowest strike rate while bowling “Right handed”.

Answer: - for finding out the result, we will need three tables namely players_ipl, most_runs_average_strike_rate and deliveries.
This can be done in two ways

1) By using subquery
2) By creating a separate view

1) By using a subquery

This is a single step query which can be written as follows: -

```
select X.batsman, X.min_strike_rate from
(select A.batsman, min(A.strikerate) as
min_strike_rate from most_runs_average_strikerate A,
players_ipl B, deliveries C where
A.batsman = B.player_name
and B.player_name = C.bowler and
C.bowling_team='Sunrisers Hyderabad' and
B.bowling_skill like 'Right%' group by A.batsman
)X
where X.min_strike_rate =
(select min(A.strikerate) as min_strike_rate from most_runs_average_strikerate A,
players_ipl B, deliveries C where A.batsman = B.player_name and B.player_name = C.bowler
and C.bowling_team='Sunrisers Hyderabad' and B.bowling_skill like 'Right%');
```

This looks very untidy and tedious to read which is why using a view is much cleaner.

```
spr2022adb35=> select X.batsman, X.min_strike_rate from
spr2022adb35-> (select A.batsman, min(A.strikerate) as
spr2022adb35(> min_strike_rate from most_runs_average_strikerate A,
spr2022adb35(> players_ipl B, deliveries C where
spr2022adb35(> A.batsman = B.player_name
spr2022adb35(> and B.player_name = C.bowler and
spr2022adb35(> C.bowling_team='Sunrisers Hyderabad' and
spr2022adb35(> B.bowling_skill like 'Right%' group by A.batsman
spr2022adb35(> )X
spr2022adb35-> where X.min_strike_rate =
spr2022adb35-> (select min(A.strikerate) as min_strike_rate from most_runs_average_strikerate A,
spr2022adb35(> players_ipl B, deliveries C where A.batsman = B.player_name and B.player_name = C.bowler
spr2022adb35(> and C.bowling_team='Sunrisers Hyderabad' and B.bowling_skill like 'Right%');
 batsman |  min_strike_rate
---------+-------------------
 S Kaul  | 42.857142857142854
(1 row)

spr2022adb35=>
```

2) By using a separate view

Create a new view which contains list of right handed bowlers with minimal strike rates and played for sunrisers Hyderabad.

create view question19 as
select A.batsman, min(A.strikerate) as min_strike_rate from
most_runs_average_strikerate A, players_ipl B, deliveries C where
A.batsman = B.player_name and B.player_name = C.bowler and
C.bowling_team='Sunrisers Hyderabad' and B.bowling_skill like 'Right%' group by
A.batsman;

```
spr2022adb35=> create view question19 as
spr2022adb35-> select A.batsman, min(A.strikerate) as min_strike_rate from most_runs_average_strikerate A, players_ipl B, deliveries C where
spr2022adb35-> A.batsman = B.player_name and B.player_name = C.bowler and C.bowling_team='Sunrisers Hyderabad' and B.bowling_skill like 'Right%' group by A.batsman;
CREATE VIEW
spr2022adb35=>
spr2022adb35=>
```

The second step is to query this view to find the lowest average of the right handed bowlers.

select batsman, min_strike_rate from question19 where min_strike_rate = (select min(A.strikerate) as min_strike_rate from most_runs_average_strikerate A, players_ipl B, deliveries C where A.batsman = B.player_name and B.player_name = C.bowler and C.bowling_team='Sunrisers Hyderabad' and B.bowling_skill like 'Right%');

OpenSSH SSH client

```
spr2022adb35=> select batsman, min_strike_rate from question19
spr2022adb35-> where min_strike_rate = (select min(A.strikerate)
spr2022adb35(> as min_strike_rate from most_runs_average_strikerate A, players_ipl B, deliveries C
spr2022adb35(> where A.batsman = B.player_name and B.player_name = C.bowler
spr2022adb35(> and C.bowling_team='Sunrisers Hyderabad' and B.bowling_skill like 'Right%');
 batsman |  min_strike_rate
---------+-------------------
 S Kaul  | 42.857142857142854
(1 row)
```

Q-20) Finding out the list of player(s), their batting average and dominant batting hand who were bowled by “TS Mills” from IPL season 2014-2019.

Answer: - For the result of this query, we need four tables namely :- players_ipl, deliveries, most_runs_average_strikerate and matches.

The query which will produce the required result is: -
select A.player_name, B.average, A.batting_hand from
players_ipl A, most_runs_average_strikerate B,
deliveries C, matches D where
C.match_id = D.id and A.player_name = B.batsman
and C.batsman = B.batsman and C.bowler='TS Mills'
and C.dismissal_kind='bowled' and
D.season in ('IPL-2014','IPL-2015','IPL-2016','IPL-2017','IPL-2018','IPL-2019');

The screenshot for the result is provided below: -

```
spr2022adb35=> select A.player_name, B.average, A.batting_hand from
spr2022adb35-> players_ipl A, most_runs_average_strikerate B,
spr2022adb35-> deliveries C, matches D where
spr2022adb35-> C.match_id = D.id and A.player_name = B.batsman
spr2022adb35-> and C.batsman = B.batsman and C.bowler='TS Mills'
spr2022adb35-> and C.dismissal_kind='bowled' and
spr2022adb35-> D.season in ('IPL-2014','IPL-2015','IPL-2016','IPL-2017','IPL-2018','IPL-2019');
 player_name  |      average       | batting_hand
--------------+--------------------+--------------
 AP Tare      |             14.125 | Right_Hand
 Yuvraj Singh | 25.275229357798164 | Left_Hand
(2 rows)
```