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

Text

Description automatically generated

1. Go to the path where files are present on the local computer

Text

Description automatically generated

1. Write the following SCP command: -

scp \* [parth2@131.252.208.103:/u/parth2/Desktop//db\_project/files](mailto:parth2@131.252.208.103:/u/parth2/Desktop//db_project/files)

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

Text

Description automatically generated

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: -

Timeline

Description automatically generated

1. 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: -

Text

Description automatically generated

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;**

Text

Description automatically generated with medium confidence

1. 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: -

Text

Description automatically generated

1. 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: -

Graphical user interface

Description automatically generated with low confidence

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;**

A picture containing timeline

Description automatically generated

1. 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: -

A picture containing timeline

Description automatically generated

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: -

Timeline

Description automatically generated with medium confidence

1. 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: -

A picture containing text

Description automatically generated

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

***PRIMARY KEYS: -***

1. teams: -

Text

Description automatically generated with low confidence

1. players\_ipl: -

Text, timeline

Description automatically generated

1. most\_runs\_average\_strikerate: -

Timeline

Description automatically generated

1. matches: -

A picture containing treemap chart

Description automatically generated

1. 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.

A picture containing chart

Description automatically generated

1. teamwise\_home\_and\_away: -

Treemap chart

Description automatically generated with low confidence

***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);

1. 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);

1. 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);

1. 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: -

Text

Description automatically generated

A picture containing timeline

Description automatically generated

Timeline

Description automatically generated

Text

Description automatically generated with medium confidence

**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: -**

Diagram

Description automatically generated

**NEW: -**

Diagram

Description automatically generated

***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;

Text

Description automatically generated

1. players\_ipl

The command used is: -

\copy players\_ipl from Players.csv with csv header;

Text

Description automatically generated

1. most\_runs\_average\_strikerate

The command used for copying is: -

\copy most\_runs\_average\_strikerate from most\_runs\_average\_strikerate.csv with csv header;

Text

Description automatically generated

1. matches

The command used to copy is given below: -

\copy matches from matches.csv with csv header;

Text

Description automatically generated

1. deliveries

The command used to copy is given below: -

\copy deliveries from deliveries.csv with csv header;

Text

Description automatically generated

1. 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;

Text

Description automatically generated

***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.

Text

Description automatically generated with medium confidence

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: -

A picture containing text

Description automatically generated

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.

Text

Description automatically generated

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

Text

Description automatically generated with medium confidence

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.

A picture containing diagram

Description automatically generated

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: -

A computer screen capture

Description automatically generated with medium confidence

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: -

Text

Description automatically generated

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: -

A picture containing chart

Description automatically generated

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: -

Text

Description automatically generated

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: -

A picture containing timeline

Description automatically generated

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: -

Text, timeline

Description automatically generated

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: -

Graphical user interface, text

Description automatically generated with medium confidence

A picture containing text

Description automatically generated

Now after getting this view, we will query this view to get the required result.

select \* from question12 where player\_of\_match\_awards > 3;

Text

Description automatically generated

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: -

Text

Description automatically generated

A picture containing timeline

Description automatically generated

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: -

Text

Description automatically generated

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: -

Text

Description automatically generated

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.

Text

Description automatically generated with medium confidence

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: -

Text

Description automatically generated

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: -

Text

Description automatically generated

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: -

Text

Description automatically generated

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
3. 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.

Timeline

Description automatically generated

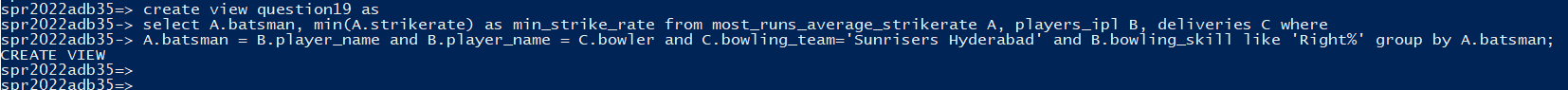
1. By using a separate view

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;



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%');

Text

Description automatically generated

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: -

Text

Description automatically generated