Comprehensive Cricket Analysis by devising a Relational DBMS

Navanshu Khare (khare050@umn.edu)
Shashank Magdi (magdi007@umn.edu)
Seetharam Reddy Ramireddy (ramir570@umn.edu)

I. Introduction

1.1 Project background : Why IPL Cricket and more importantly what is Cricket?

Cricket is played between two teams of 11 players on each team. The matches are officiated by two on field and two off field umpires. Like the heavily popular sport of Baseball, one team will bat while the other team takes the field. A bowler is the one who bowls the ball to the batsmen while the batting side starts the match by sending out two players to bat and score runs.

Of course there are other aspects to the sport like Run : which is the most simple unit of scoring in cricket. A run can be 1,2,3,4,5,6 depending on the outcome after the batsmen makes a shot. Dismissal : This term marks the end of respective batters batting period (can be compared to the 'out' call in Baseball). The goal for the team fielding is to dismiss 10 players of the opposition's 11.

Important to note in cricket, an over consists of six consecutive legal deliveries bowled by the bowler to the player batting. The different formats of cricket often vary depending on the number of overs present in the match. There can be many formats of cricket like the longest format which can go on until five days(Test Cricket), a significantly shorter format(One-day match) and the shortest format(Twenty20 cricket) which comprises of 20 overs.

The sole aim is to score more runs than the opposition and you win! Yes, it's that simple! Now that it is briefly outlined how cricket is played, we have chosen to carry out our analysis on IPL cricket.

A 20 over game in the sport of cricket is such a volatile format where numbers and large volumes of data(mostly usually numbers) are often used to make key strategic decisions right from acquiring players from an auction (at either an inflated price or an inexpensive steal) to acquiring local players to perform exceedingly well for their respective franchises and thus nurture local talent and also selecting young prodigies to represent their respective countries.

1.2 The need:

Key attributes like strike rate(how quickly one can score), player averages, win percentages as a captain / coach, most number of runs scored consistently, Man of the Match awards. can be used and represented initially in an ERD and then used to build a relational Database system to answer curious questions which the end users certainly seem to have. Many not so keen on eye statistics can be retrieved and talked about offering a different perspective on the player's performance / importance to the respective teams.

1.2.1 Target Audience:

1) Ardent and inquisitive Cricket fans and fantasy team owners : No one wants to miss being in the thick of action, particularly fantasy owners who would want to arm themselves with such data to make changes prior to everyone and get a head start!
2) Sports Analysts : To analyse and get key insights and offer invaluable inputs to team management to make miniscule changes which might work wonders in the long-run.
3) Selection Panels and Franchise scouts : They would like to understand peculiar trends and try to unearth prodigal talents, therefore attracting more talented players to pursue cricket as a profession.
4) Podcasts and Sports shows : Offer a visual treat to the audience by unravelling the large messy data.
And of course the players themselves!

1.3 The goal(s) of the project:

a. Understand the complex Kaggle Indian Premier Dataset precisely and create a relational DBMS using the concepts we learn from the class like constructing an ERD, also mapping it to the tables among other steps.

b. Pre-process the chosen data and implement some insightful Exploratory Data Analysis (EDA) like understanding the distribution particularly from a cricket point of view as in how many left hand batters exist, how many right hand batters exist, also how many matches are being played at different venues among others.

c. To ultimately formulate desired user-oriented SQL queries that can effectively use the massive amounts of available data of Indian Premier League in the world to answer some inquisitive questions the end user might have.

d. Also understand how much time each query is taking to execute, and discuss various query optimization techniques to improve this run time and finally plot the results.

1.4 The Scope:

The scope of the project as a use case to answer inquisitive questions with the help of queries is undeniable. The ease of obtaining results for the questions of end users can be made an even better experience if query optimization is properly implemented to offer a seamless experience. This field of sports analytics is still expanding and can be something very magnanimous if the right data cleaning and parameter picking and optimization is performed. This is the reason we had done some EDA to understand what is the data we are dealing with.

II. Dataset Description

The Indian Premier League CSV Dataset from Kaggle has been used here, with extensive information consisting of player's match's metadata, most importantly ball by ball data of about 577 IPL matches played between the years 2008 - 2016 to give us enough data to work on. We are having a total of 21 excel sheets that maintain all the information about all the seasons of IPL. Some of them are Ball_by_Ball.csv and which contains ball by ball details of all the 577 matches, Match.csv contains Match metadata (like match date, teams involved in the match, venue of the match, who won the toss, and also winner information), Player.csv has player metadata (like player date of birth, batting hand, bowling skill), Player_Match.csv contains information about players involved in a particular match and the team with which they were involved, Season.csv has information about the Purple cap, Orange cap winners and also the man of the series of each IPL season, Team.csv has information about all franchises that are involved in IPL along with their unique acronyms.

III. Method

3.1 Relationship Diagram:

The relationship diagram is one of the key aspects of the project as it gives us the roadmap on how to decide which attributes to use and eventually map this onto the tables that we are supposed to create. The dashed line means that foreign key column is not a primary key in a referencing table and the solid line means that the foreign key column is also a primary key in referencing table. It is as follows:

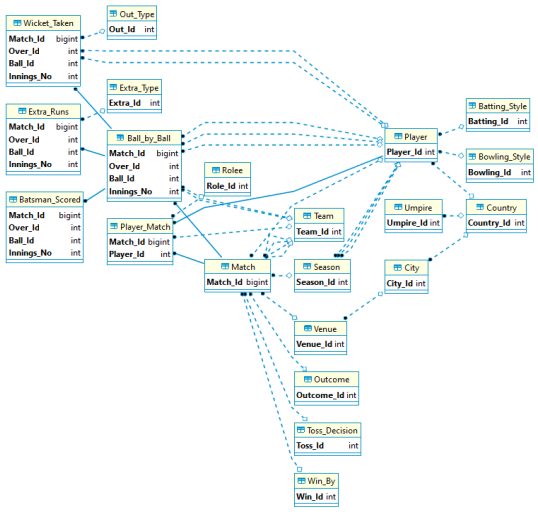


Fig 1. Relationship Diagram

3.2 Defining the Tables:

We have created various tables, mentioning the required constraints like NOT NULL constraints forcing the column not to accept null values. An example of our Ball_by_ball table has been attached for reference:

```sql
CREATE TABLE IPL.dbo.Ball_by_Ball (
    Match_Id bigint NOT NULL,
    Over_Id int NOT NULL,
    Ball_Id int NOT NULL,
    Innings_No int NOT NULL,
    Team_Batting int NOT NULL,
    Team_Bowling int NOT NULL,
    Striker_Batting_Position int NOT NULL,
    Striker int NOT NULL,
    Non_Striker int NOT NULL,
    Bowler int NOT NULL,
    CONSTRAINT BallByBall_pk PRIMARY KEY (Match_Id,Over_Id,Ball_Id,Innings_No),
    CONSTRAINT fk_bowler FOREIGN KEY (Bowler) REFERENCES IPL.dbo.Player(Player_Id),
    CONSTRAINT fk_match1 FOREIGN KEY (Match_Id) REFERENCES IPL.dbo.[Match](Match_Id),
    CONSTRAINT fk_nonstriker FOREIGN KEY (Non_Striker) REFERENCES IPL.dbo.Player(Player_Id),
    CONSTRAINT fk_striker FOREIGN KEY (Striker) REFERENCES IPL.dbo.Player(Player_Id),
    CONSTRAINT fk_team3 FOREIGN KEY (Team_Batting) REFERENCES IPL.dbo.Team(Team_Id),
    CONSTRAINT fk_team4 FOREIGN KEY (Team_Bowling) REFERENCES IPL.dbo.Team(Team_Id)
);
```

Fig 2. Creation Ball_by_ball relation

In a similar way all the other required tables have been created too :

```sql
CREATE TABLE IPL.dbo.Batsman_Scored (
    Match_Id bigint NOT NULL,
    Over_Id int NOT NULL,
    Ball_Id int NOT NULL,
    Runs_Scored int NOT NULL,
    Innings_No int NOT NULL,
    CONSTRAINT Batsmanscco_pk PRIMARY KEY (Match_Id,Over_Id,Ball_Id,Innings_No),
    CONSTRAINT FK_Matchh_Ba FOREIGN KEY (Match_Id,Over_Id,Ball_Id,Innings_No) REFERENCES IPL.dbo.Ball_by_Ball(Match_Id,Over_Id,Ball_Id,Innings_No)
);
```

Fig 3. Creation of Batsman_scored relation

3.3 Outlook of the tables:

Table Name	Total bytes	Type	Used bytes
Ball_by_Ball	8,134,656	U	8,101,888
Batsman_Scored	4,530,176	U	4,505,600
Batting_Style	16,384	U	16,384
Bowling_Style	16,384	U	16,384
City	16,384	U	16,384
Country	16,384	U	16,384
Extra_Runs	401,408	U	303,104
Extra_Type	16,384	U	16,384
Match	57,344	U	57,344
Out_Type	16,384	U	16,384
Outcome	16,384	U	16,384
Player	40,960	U	40,960
Player_Match	729,088	U	712,704
Rolee	16,384	U	16,384
Season	16,384	U	16,384
sysdiagrams	106,496	U	106,496
Team	16,384	U	16,384
Toss_Decision	16,384	U	16,384
Umpire	16,384	U	16,384
Venue	16,384	U	16,384
Wicket_Taken	344,064	U	303,104
Win_By	16,384	U	16,384

Fig4. List of Relations

Column Name	#	Type	Length	Scale	Precision	Not Null	Identity	Default	Collation
Match_Id	1	bigint	8		19	[v]	[ ]		
Over_Id	2	int	4		10	[v]	[ ]		
Ball_Id	3	int	4		10	[v]	[ ]		
Player_Out	4	int	4		10	[v]	[ ]		
Kind_Out	5	int	4		10	[v]	[ ]		
Fielders	6	int	4		10	[ ]	[ ]		
Innings_No	7	int	4		10	[v]	[ ]		

Fig5. Structure of Ball_by_Ball relation

	Match_Id	Over_Id	Ball_Id	Player_Out	Kind_Out	Fielders	Innings_No
1	335,987	2	1	6	2	[NULL]	2
2	335,987	3	2	8	2	[NULL]	2
3	335,987	5	5	9	1	83	2
4	335,987	6	2	1	1	9	1
5	335,987	6	2	7	1	3	2
6	335,987	8	5	11	1	83	2
7	335,987	9	2	12	1	3	2
8	335,987	9	8	10	1	62	2
9	335,987	12	1	13	3	82	2

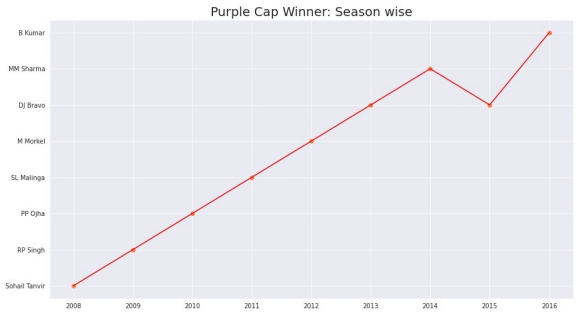
Fig6. Outlook of a few rows of Ball_by_Ball

Now the method to set up the DB has been mentioned below:

```
In [4]: # Connect to Database

        server = 'AUSTRALIA-PC\SQLEXPRESS'
        database = 'IPL'
        username = 'sa'
        password = 'admin'
        conn = pyodbc.connect('DRIVER={ODBC Driver 17 for SQL Server};SERVER='+server+';DATABASE='+database+';UID='+username+';PWD='+

        print("Database Connected")

        Database Connected
```

Fig7. Creating and seeding the DB

4.2 Exploratory Data Analysis:



=

This graph displays the Purple Cap (given to player who picks most number of wickets in a season) winner of the IPL each year since its inauguration and you can observe that only DJ Bravo won it twice in 2013 and 2015.

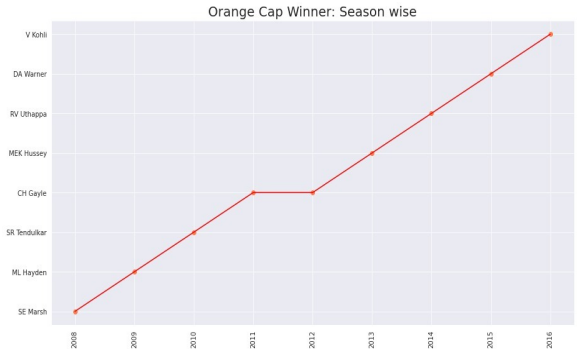


Fig9. Orange Cap Winner: season wise

This graph shows the Orange Cap (given to player who scores most number of runs in a season) winner of the IPL each year and only CH Gayle was able to win it twice and made it possible in consecutive years.

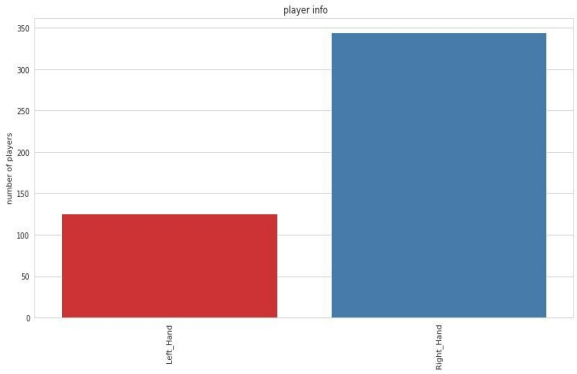


Fig10. Left-handed and Right-handed comparison

This graph shows an intriguing comparison between the number of left handed and right handed batsman who participated in IPL and it is evident that right handed batsman are larger compared to left handed batsman.

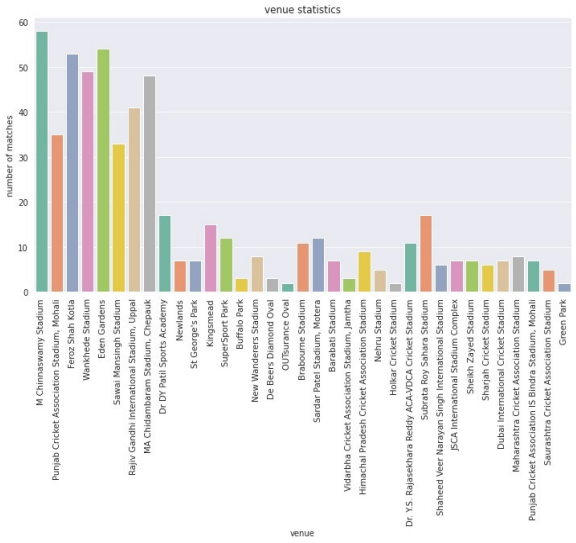


Fig 11. Match venue depiction

This graph depicts the number of matches played at each venue during the stint of IPL.

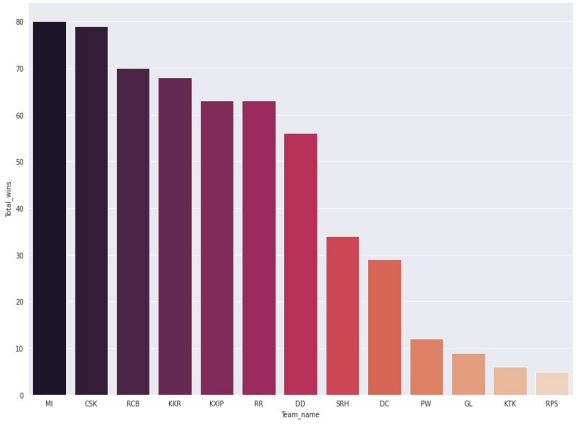


Fig 12. No. of matches

This graph displays the number of matches won by each franchise during the stint of IPL and don't get deluded by this massive variation in number of matches won by franchises as some franchises participated only in one to two seasons.

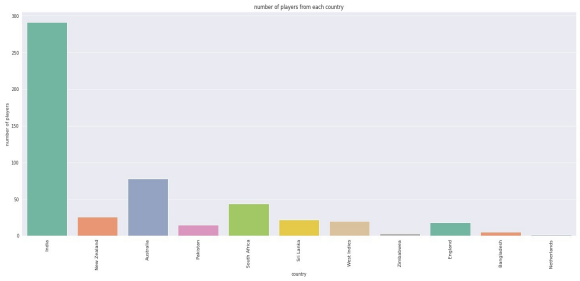


Fig13. Country wise players

This graph portrays the total number of players from different countries to ever take part in IPL and the fact that a greater number of players are from India is undeniable because IPL is a tournament held to encourage young Indian players to perform at world stage.

4.2.1 After EDA:

We downloaded the dataset from source and then Deeply analyzed and established an understanding of different tables and their columns along with their connection with different tables. This helped us to decide on the attributes that we need for our project. Our next plan of action was to design and create a Database. We started by making an Entity Relationship diagram with 21 tables and set a primary key and established a connection between them with foreign Key. Once we knew what our DB should be designed like. We finalized to move ahead with Microsoft SQL server after giving exploring Oracle and Postgres and MYSQL servers. We wrote create Queries for the tables and started designing DB in a server where we faced quite challenges technical as well theoretical due to which we modified our DB and well as queries multiple times. Our database was ready with all 21 tables with primary and foreign keys. Next, we pondered which language to choose implementing for SQL queries. Java or Python. We preferred Python over Java due to ease of Visualization in Python. We established a connection between db and python. We switched various configurations in Server side like enabling TCP-IP, or Disabling SSH and many more. We started in Python by executing simple queries like fetching the data from Table or Getting the count.

4.3 SQL Queries

4.3.1 Players who were able to captain most number of matches in IPL:

```
## Players who were able to captain most number of matches
start = time.time()
cursor = conn.cursor()
query = """ SELECT C.Player_Name , COUNT(*) As 'Matches_captained'
    FROM Player_Match A
        INNER JOIN Rolee B
            ON A.Role_Id = B.Role_Id
        INNER JOIN Player C
            ON A.Player_Id = C.Player_Id
    WHERE A.Role_Id = 1 OR A.Role_Id = 4
    GROUP BY C.Player_Name
    ORDER BY Matches_captained DESC;"""
sql = pd.read_sql(query, conn)
end = time.time()
total_times.append(end - start)
print(sql.head())
print("Total time taken ::", end - start)
plot_query_results(cursor,query,50,'green')
cursor.close()
```

Fig14. Captain query

The result for the query is formulated below with MS Dhoni being the player with the most number of matches captained.

```
    Player_Name  Matches_captained
0      MS Dhoni                142
1     G Gambhir                107
2  AC Gilchrist                 74
3       V Kohli                 71
4     RG Sharma                 58
Total time taken :: 0.009979963302612305
0.006063756942749024
```

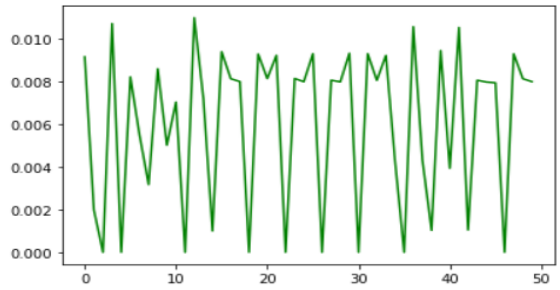


Fig 15. Captain query result

4.3.2 Players who took highest number of catches in IPL:

```
## Players who took highest number of catches in IPL
start = time.time()
cursor = conn.cursor()
query = """ SELECT TOP 10 Player.Player_Name ,
        COUNT(*) AS 'Catches'
    FROM Wicket_Taken
    INNER JOIN Out_Type
        ON Wicket_Taken.Kind_Out = Out_Type.Out_Id
    INNER JOIN Player
        ON Player.Player_Id = Wicket_Taken.Fielders
    WHERE Out_Type.Out_Name = 'caught'
    GROUP BY Player.Player_Name
    ORDER BY Catches DESC"""
sql = pd.read_sql(query , conn)
end = time.time()
total_times.append(end - start)
print("Total time taken ::", end - start)
print(sql.head())
plot_query_results(cursor,query,50,'green')
cursor.close()
```

Fig16. Catches Query

Clearly, KD Karthik took the highest number of catches which is factually correct according to the records till 2016.

```
Total time taken :: 0.03955411911010742
       Player_Name  Catches
0       KD Karthik        80
1          SK Raina        79
2  AB de Villiers        77
3          MS Dhoni        66
4       RV Uthappa        66
0.016487321853637694
```

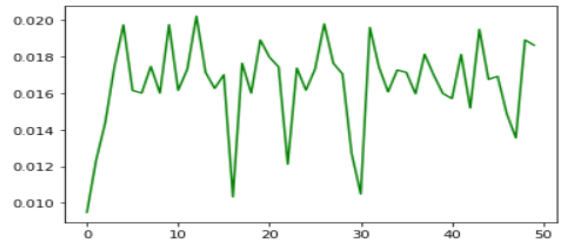


Fig 17. Catches Query result

4.3.3 Player of the match awards won by Kohli

```
## Player of the match awards won by Kohli
start = time.time()
cursor = conn.cursor()
query = """SELECT ( SELECT COUNT(*)
    FROM Player_Match A
    WHERE A.Player_Id = 8 ) As 'Matches_Played' ,
    ( SELECT COUNT(*)
        FROM Match B
    WHERE B.Man_of_the_Match = 8 ) As 'Man_Of_The_Match'  ; """
sql = pd.read_sql(query, conn)
end = time.time()
total_times.append(end - start)
print("Total time taken ::", end - start)
print(sql.head())
plot_query_results(cursor,query,50,'green')
cursor.close()
```

Fig 18. Kohli Man of match award

Virat Kohli being a hugely popular figure in world cricket, some fans might be inclined to know some inquisitive facts about him.

```
Total time taken :: 0.011814355850219727
   Matches_Played  Man_Of_The_Match
0             138                11
0.000858306884765625
```

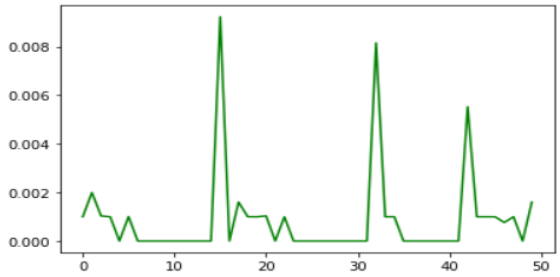


Fig19. Kohli Man of the Match awards

4.3.4 Top scorers in an innings in IPL history.

```
   Match_Id      Player_Name  Runs
0    598032        CH Gayle   175
1    335987      BB McCullum   158
2    829800  AB de Villiers   133
3    980992  AB de Villiers   129
4    548377        CH Gayle   128
1.3306163787841796
```

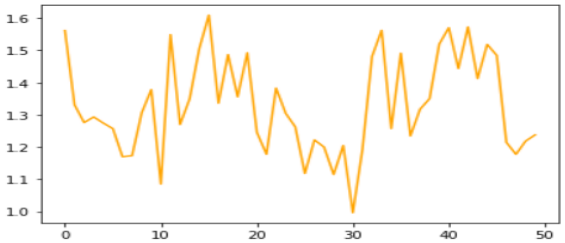


Fig 20. Top scorers result

Clearly the top scorer of all time in a single IPL match is CH Gayle with a monstrous 175. Also important to note, the next best score of 158 was posted by BB McCullum in the very first match of the inaugural 2008 edition. Just a fan trivia!

4.3.5 Players who hit highest number of sixes in IPL

```
Total time taken :: 0.3030693531036377
  Player_Name  Sixes
0    CH Gayle    252
1   RG Sharma    164
2    SK Raina    161
3     V Kohli    148
4   YK Pathan    143
0.22306978225708007
```

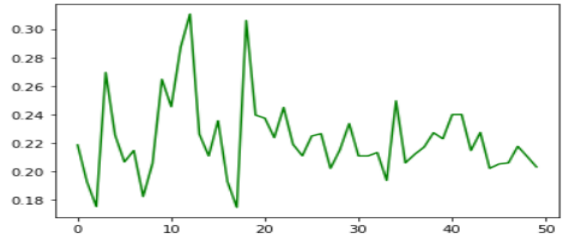


Fig 21. Highest number of sixes.

4.3.6 Most number of player of the match awards won by players

```
Total time taken :: 0.02301478385925293
      Player_Name  Man_Of_The_Matches
0        CH Gayle                 17
1       YK Pathan                 16
2  AB de Villiers                 15
3       DA Warner                 14
4       RG Sharma                 13
0.002364687919616699
```

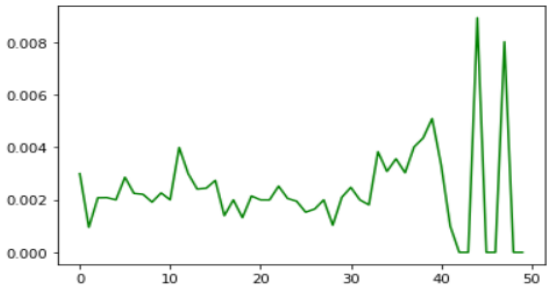


Fig 22. Highest no. of Man of the Matches

Man of the Match awards are extremely prestigious in cricket, these are analogous to MVP awards and emphasize the importance of the player(s) to the team.

4.3.7 Virat Kohli Captaincy stats comparison

```
Total time taken :: 0.03376483917236328
```

	Player_Name	Matches_Captained	Won_Matches
0	V Kohli	71	37

Fig 23. Kohli as a captain stats

This could turn out to be a heavily demanded one particularly when you can compare a captain's stats against their individual ones.

4.3.8 Successful teams' comparison:

While we analyzed individual contributions, it was important to understand how each team fared and the success it achieved.

	Team_Name	Total_Matches	Won_Matches	Winning_Percentage
0	Chennai Super Kings	131	79	60
1	Mumbai Indians	140	80	57
2	Gujarat Lions	16	9	56
3	Sunrisers Hyderabad	62	34	54
4	Rajasthan Royals	118	63	53
5	Kolkata Knight Riders	132	68	51
6	Royal Challengers Bangalore	139	70	50
7	Kings XI Punjab	134	63	47
8	Delhi Daredevils	133	56	42
9	Kochi Tuskers Kerala	14	6	42

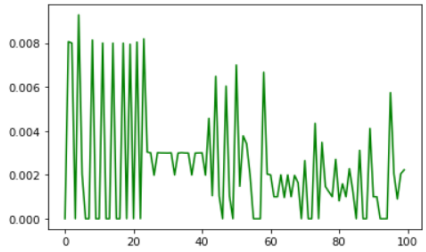
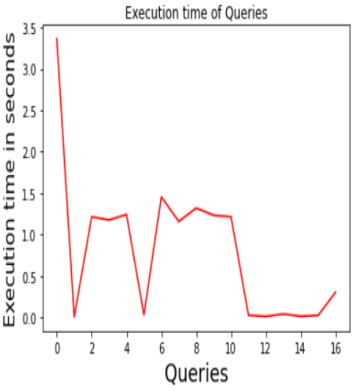


Fig 24. Matches won, win % comparison

4.3.8 Time execution comparison:

The time taken to execute each of our query has been plotted against the number of the queries themselves and it can be explained that first one has a high run time because that is when we had loaded our data in.

```
[3.362942695617676,
 0.0029954910278320312,
 1.2137389183044434,
 1.175901174545288,
 1.2411046028137207,
 0.02770709991455078,
 1.4545128345489502,
 1.1569514274597168,
 1.3192124366760254,
 1.2323222160339355,
 1.2136082649230957,
 0.02431654930114746,
 0.009979963302612305,
 0.03955411911010742,
 0.011814355850219727,
 0.02301478385925293,
 0.3030693531036377]
```



V. Conclusion

1) Objectives achieved:

We were able to implement working relational database and perform some insightful exploratory data analysis. Also understood how to implement a working relational DB and the various steps involved right from creating a useful Entity relationship Diagram. Then the process of mapping ERD to tables and setting up the DB was successfully achieved. Understanding how to formulate queries to achieve the user-oriented goals and trying to understand methods to optimize these queries such as Top n Queries was delightful to learn.

2) Challenges faced and lessons learnt

a) Finalizing a Server after trying and failing for Postgres, Oracle and many more (Many other factors due to which some did not work and ultimately went for MS SQL).
b) While setting up connection with Python, issues like not able to identify the login and DB not found problems were solved.
c) Designing the queries, journey of getting errors and wrong results to Getting the desired Output.

VI. Future Work

The future work of the project can be creating something that is much more scalable, something like creating a R-shiny dashboard visualizing and showcasing work in an extensive format, while also looking into the prediction analysis part of the aspect as the professor mentioned in the presentation so as to try predicting future outcome of a possible auction and what price would a player would be sold at after a successful / unsuccessful season and comparing the results. As talked about, this field of sports analytics is still budding and can be something very magnanimous if the right data cleaning and parameter picking and optimization is performed.

VII. Acknowledgement

We chose Python as the programming language for this project since it includes modules that make working with databases easier. Data cleansing was aided with pandas and numpy, for example. Meanwhile, Dbeaver assisted us in converting our data from CSV files to tables. We chose MS SQL because our team had prior experience with it.

In terms of task division, we all discussed the concept and the direction we intended to take this project. Shashank had put all his efforts to deeply understand the complex IPL dataset and create relationship diagram and also set up the MS SQL server and further went on to create tables and mapped ERD to tables. Seetharam performed insightful exploratory data analysis and Navanshu worked on writing inquisitive queries to answer questions of some ardent cricket fans and also calculated execution time of queries.

VIII. References

1) Indian Premier League Dataset Analytics using Hadoop-Hive by Sapna S., Sandhya S. - International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-9 Issue-2, December, 2019.

2) https://cricsheet.org/downloads/ : This website has an enormous number of resources of freely available structured data including ball by ball data for all the major 20 over tournaments.

3) https://knoema.com/insights/cricket : This website has several examples where people have modelled, visualised the data particularly with projects in the sports domain. Very inspiring.

4) https://cricsheet.org/format/#csv This webpage has the data available to download in JSON, YAML, XML and 2 versions in CSV file format (their current YAML format currently provides the data in both the CSV, XML formats). The CSV format which is usually is easier to work with and the same

data opens up directly in Excel has a couple of formats: ‘Ashwin’ and ‘Original’

5) https://pandimi.wordpress.com/tag/ipl/ : This fantastic webpage houses so many examples of how data in sports can be visualised and depicted in a way that ardent fans can engage further to gain deep insights.

6)https://www.kaggle.com/harsha547/indian-premier-league-csv-dataset : This dataset has CSV files with the given metadata of all the 577 IPL matches.

7) Top n Queries : https://use-the-index-luke.com/sql/partial-results/top-n-queries

8) Comprehensive Data Analysis and Prediction on IPL using Machine Learning Algorithms Amala Kaviya V.S.1 , Amol Suraj Mishra2 and Valarmathi B.3 : International Journal on Emerging Technologies 11(3): 218-228(2020)

9) Previous year’s Project presentations as a reference to formulate the workflow of the project.