Technical Report for Player's Performance in IPL from year 2016-2019

By

Name: Koushik Chikkegowda
Student No : 10515678

Name: Nikhil Raut
Student No: 10521287

Name: Santhosh Vinayaka
Student No: 10518683

Name: Rakesh Shah
Student No: 10520940

CONTENTS

1. INTRODUCTION

The Indian Premier League (IPL) is a professional cricket league in India contested every year by nine teams representing nine different cities in India. In this report, we are showing the player's performance based on various facts like tournament that had taken place each year, teams which are involved in each game, etc. The dataset for this project has been extracted from the data world, which is sub-categorized containing information based on each team, each player, tournaments played, etc...

1.2. REASONS FOR SELECTING THE SUBJECT AREA AND DATA

IPL being a huge franchise, we can see that there are numbers of investors who are making a lot of money from this particular game. Hence, using data analytics to analyze the data and making a report which gives information on each player and his performance helps the investors to bid for the next best player for future games which in return helps them gaining a huge sum of profit and also helps the franchise to grow bigger keeping the glory of the game forever. The link for the dataset is given below –

Link: https://data.world/cclayford/cricinfo-statsguru-data

1.3. INDIAN PREMIER LEAGUE VISION AND GOALS

1. The IPL has been designed to entice an entirely new generation of sports fans into the grounds throughout the country.
2. The dynamic Twenty20 format has been designed to attract the young fan base, which also includes women and children.

1.4. KEY STAKEHOLDERS

The League has delivered monetary for the players, franchisees, sponsors and India as a whole, enticing a strong desire among a range of stakeholders to suitably value it.

Key stakeholders for IPL:

1. The Ambanis
2. Shah Rukh Khan
3. Dia Mirza
4. Srinivasan
5. Juhi Chawla
6. Preeti Zinta
7. Raj kundra
8. Shilpa Shetty

- The most exciting feature of this premier league concept is that the state teams are owned by corporate houses and celebrities. Owners were allotted teams and their respective players through the bidding process. Now the teams were to buy cricketers within the maximum equally fixed amount in a way that formed the team with the right mix of foreign and home players.
- IPL is a very huge and successful platform to showcase the cricketing talent for the cricket player. Also, cricket players are earning a great amount of money via bidding and per match payment process.

Brand value

- American Appraisal, the valuation firm has valued IPL at $3.2 Billion. It has also predicted that the league's value will skyrocket to over $400 Billion by 2020.
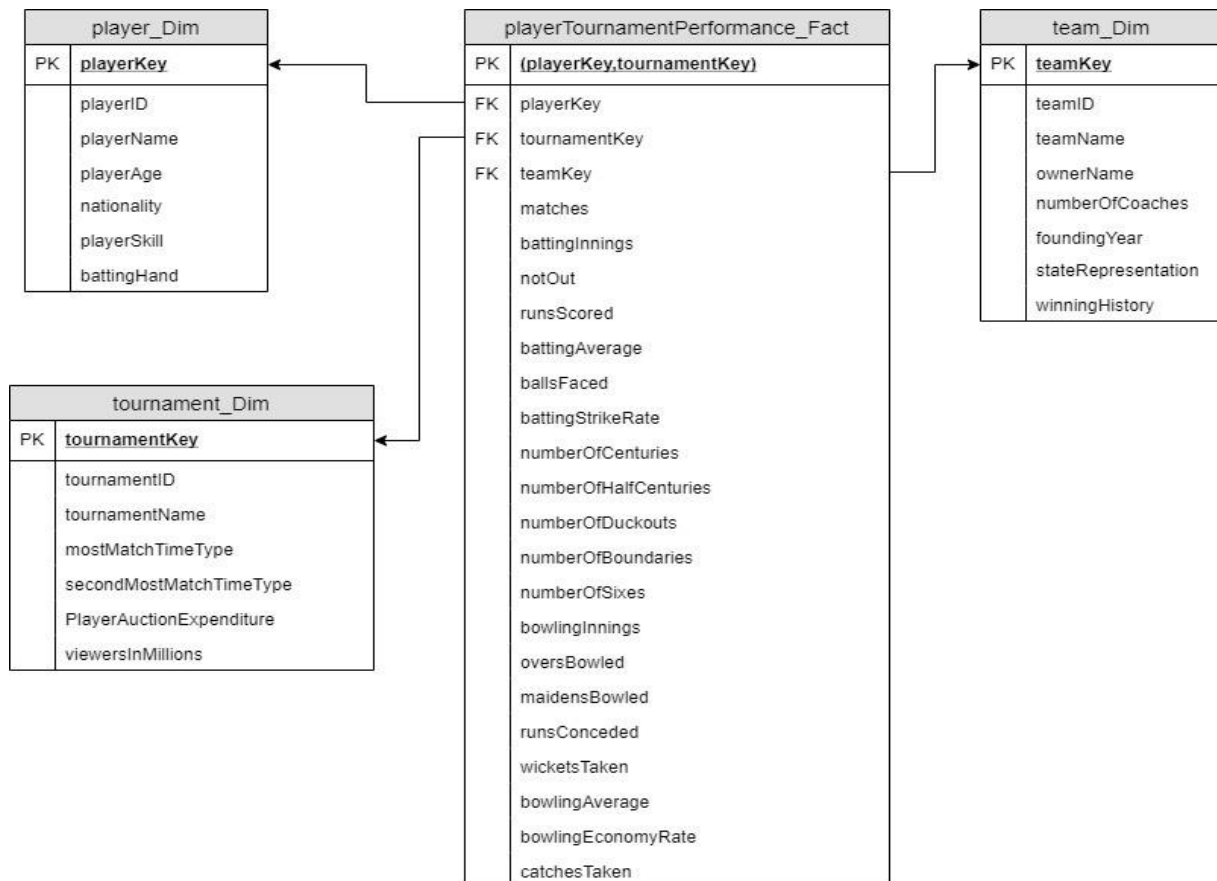
Brand values of IPL franchises are: -

- Sunrisers Hyderabad: $25 million
- Kings XI Punjab: $32 million
- Delhi Daredevils: $40 million
- Rajasthan Royals: $45 million
- Royal Challengers Bangalore: $51 million
- Kolkata Knight Riders: $69 million
- Mumbai Indians: $72 million
- Chennai Super Kings: $72 million

Project Significance

- We are analyzing players based on various graphs and reports established on their history records. Plotting player performance as the age varies or as the experience grows.
- When it comes to investing money, foreign players are the most expensive players in IPL history. We are showing a graph in the SSRS report with title "foreign players to invest money", so that investors will have an overview of the scenario.

2. SCHEMA

- In a Data Warehouse, the schema is more important for designing it. All the schematic diagrams have their advantages and disadvantages over the other. Based on the subject of analysis, the schematic representation can be decided and designed.
- Here, for this project, we have used star schema because it has a normalized data structure and the values can be extracted easily from the dimension tables.
- The star schema for this project is given below-

player_Dim

PK	playerKey
	playerID
	playerName
	playerAge
	nationality
	playerSkill
	battingHand

playerTournamentPerformance_Fact

PK	(playerKey,tournamentKey)
FK	playerKey
FK	tournamentKey
FK	teamKey
	matches
	battingInnings
	notOut
	runsScored
	battingAverage
	ballsFaced
	battingStrikeRate
	numberOfCenturies
	numberOfHalfCenturies
	numberOfDuckouts
	numberOfBoundaries
	numberOfSixes
	bowlingInnings
	oversBowled
	maidensBowled
	runsConceded
	wicketsTaken
	bowlingAverage
	bowlingEconomyRate
	catchesTaken

team_Dim

PK	teamKey
	teamID
	teamName
	ownerName
	numberOfCoaches
	foundingYear
	stateRepresentation
	winningHistory

tournament_Dim

| PK	tournamentKey
	tournamentID
	tournamentName
	mostMatchTimeType
	secondMostMatchTimeType
	PlayerAuctionExpenditure
	viewersInMillions

3. ETL

- In general, Extract, Transform and Load (ETL) is the general procedure of copying data from one or more sources into a destination system that represents the data differently from the source or in a different context than the source.
- In this project, we have implemented ETL on one fact table and three-dimension tables. The same has been explained below with screenshots.

1. <u>Player Dimension</u> – Here, all unique information about each player contested in the IPL tournament has been listed respectively. The information contains attributes such as – Nationality, Age, Player Skill, Batting Hand. Now we have assigned player ID uniquely to each player. Finally, the ETL process has been implemented as given in the below screenshot –
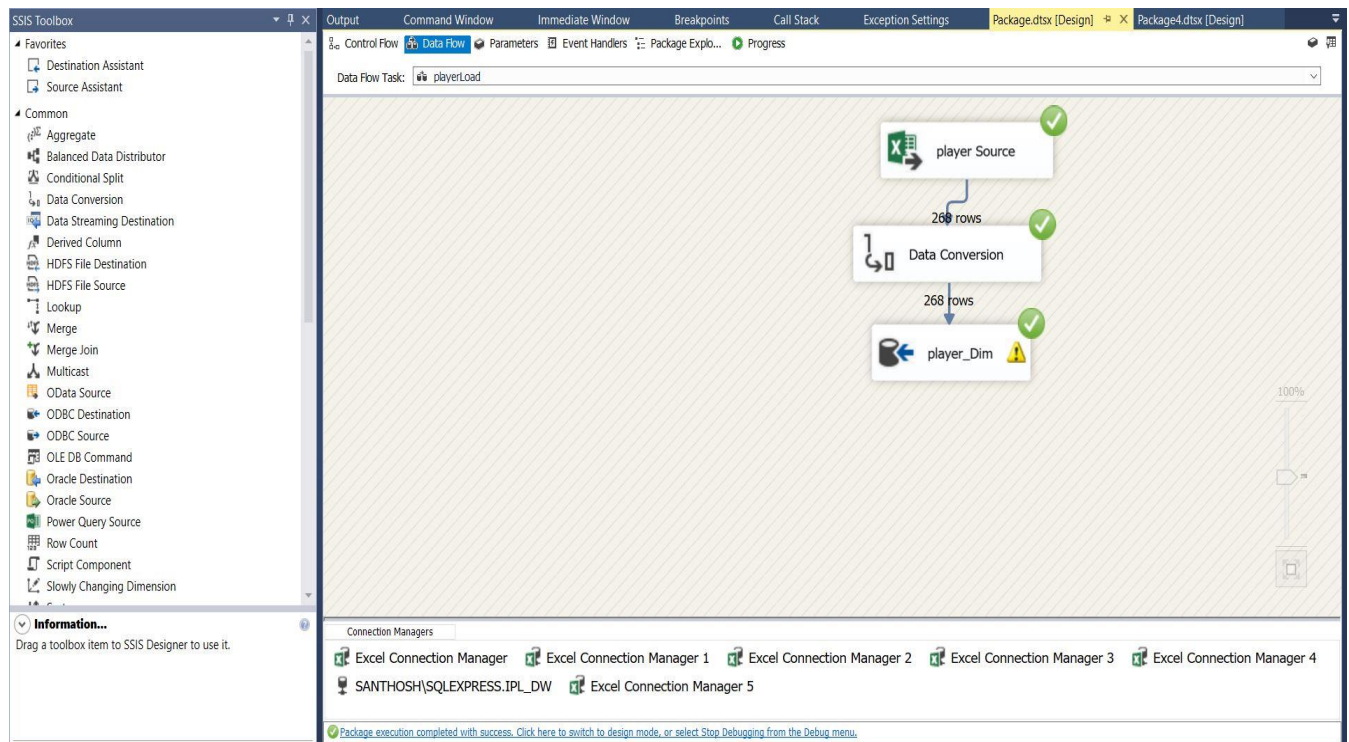
2. <u>Team Dimension</u> – All the teams competing every year are listed in the team dimension. This table includes attributes such as team name, Team owner, Average number of coaches, IPL winning history, founding year and state representation. Each team has been assigned a unique team ID. The screenshot representing this table is shown below-
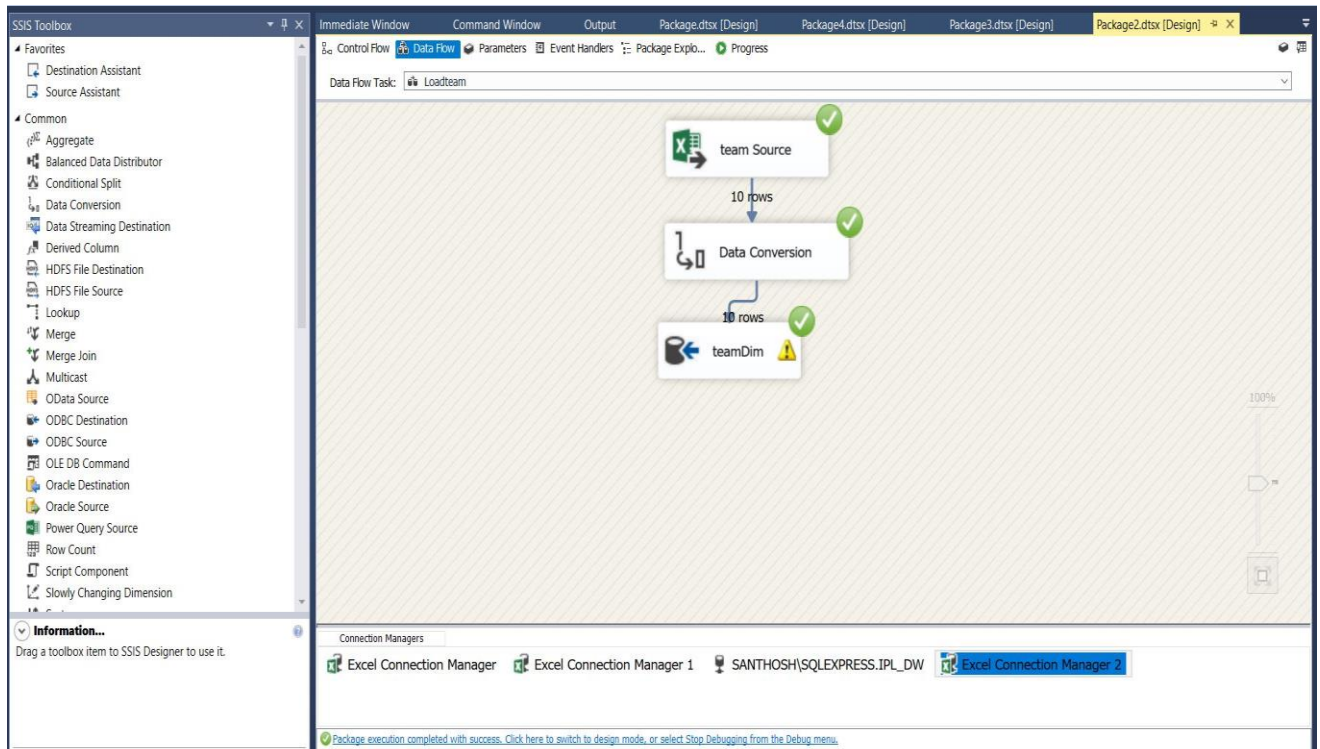
3. <u>Tournament Dimension</u> –The IPL tournament takes place once in a year. Now, we have a database for the IPL tournaments from the year 2016 to 2019. This dimension also contains information about the most matches time type, overall player auction expenditure in crores and viewers in millions.
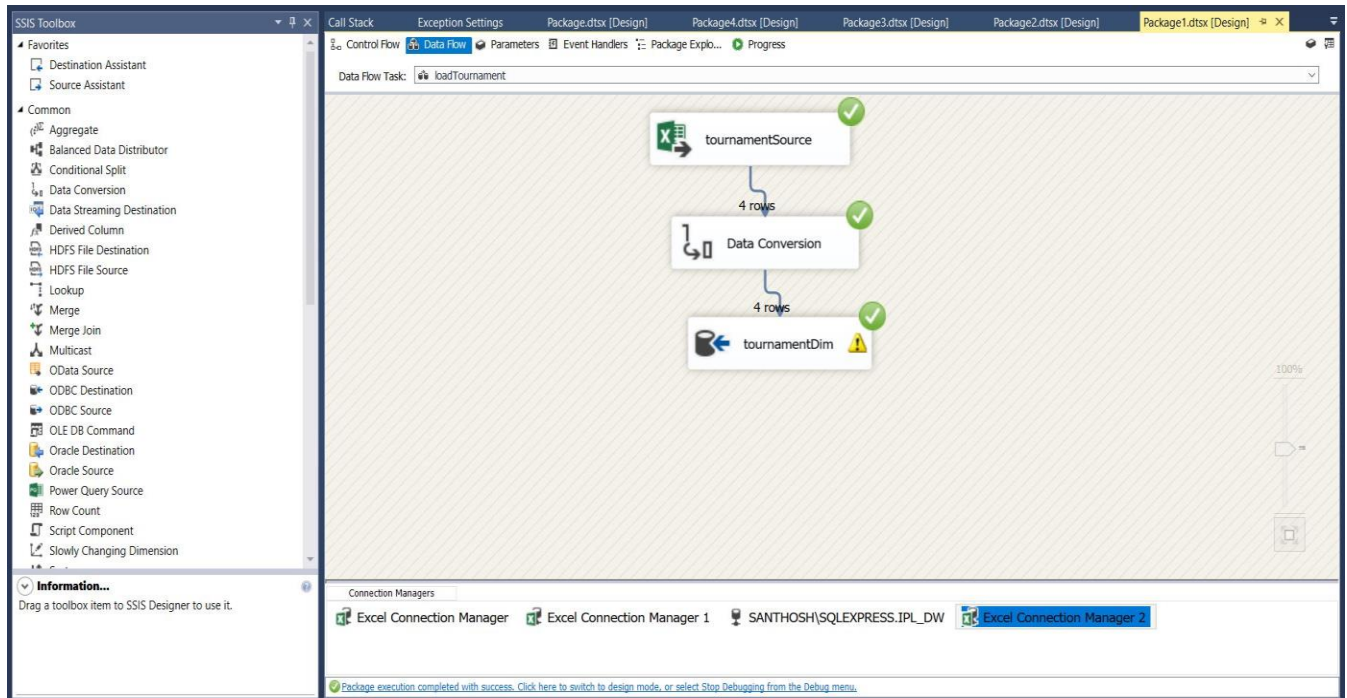
4. <u>Player Performance (Fact)</u> – The main purpose of this fact table is to showcase the player performance based on various dimensions and its attributes. This fact table includes primary keys from various dimensions as the foreign keys to implement the ETL process. The screenshot for the same is given below –
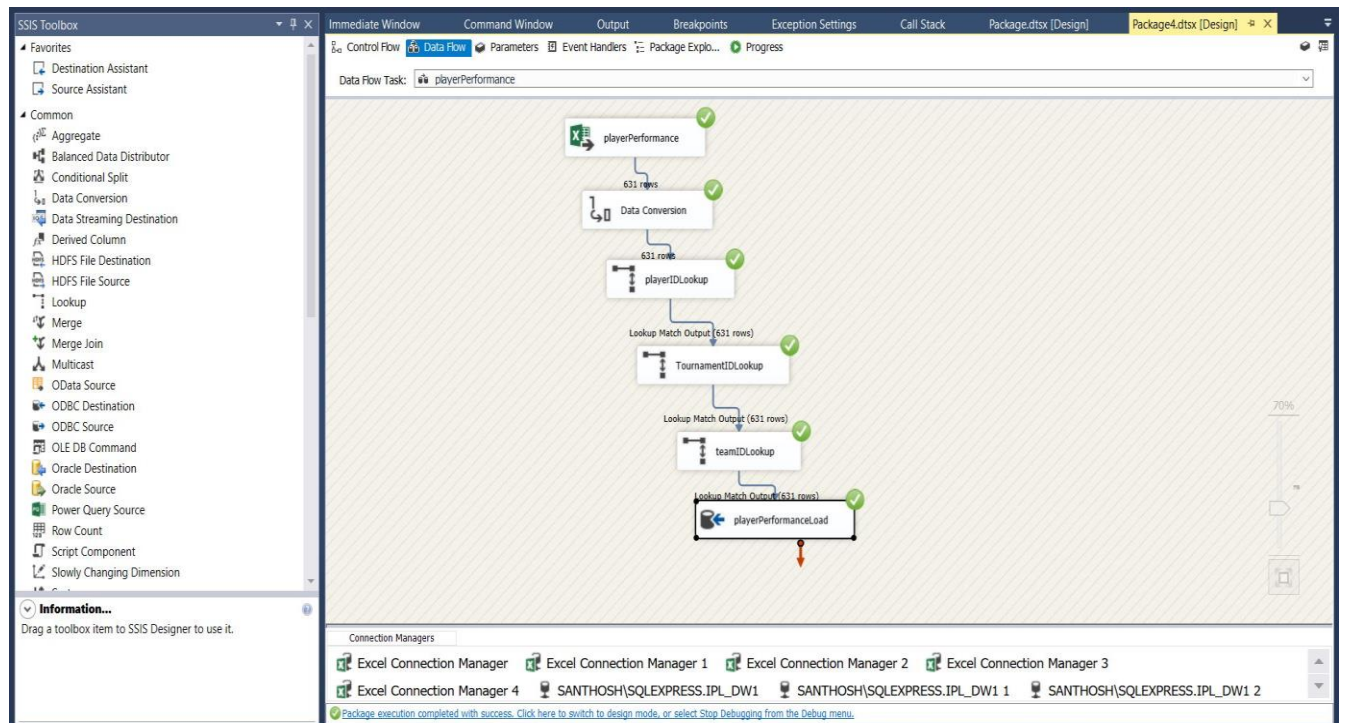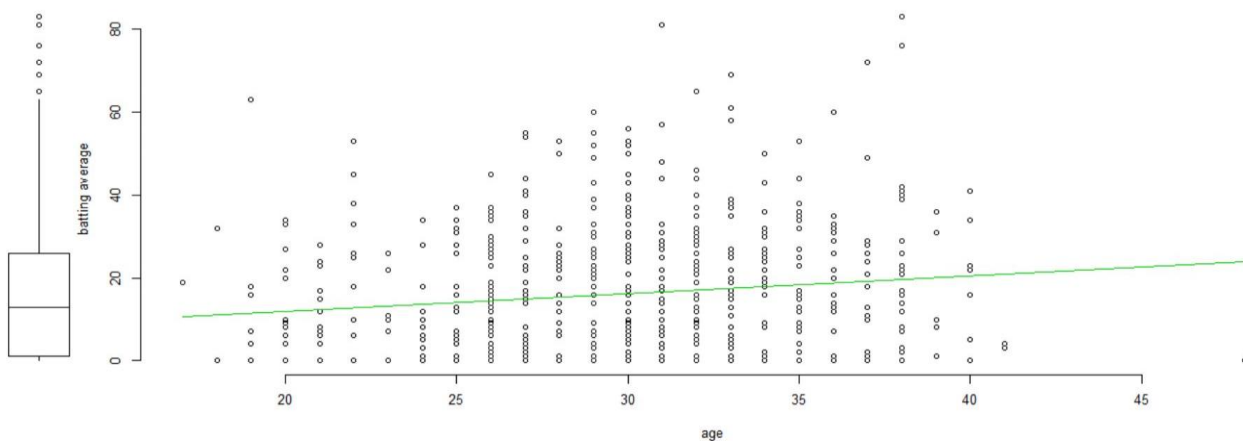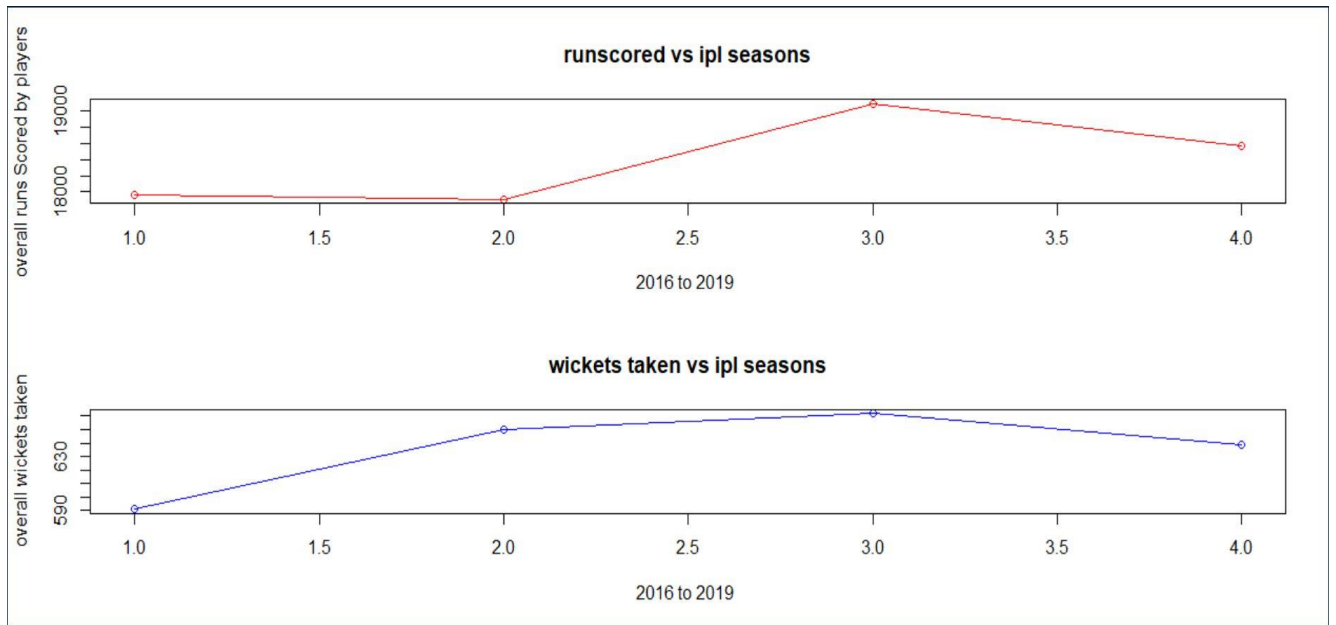
4. VISUALIZATIONS AND REPORTS

4.1. VISUALIZATIONS

- Visualizing the overall wickets taken by each team.
- The visualization is done using a bar chart which is as shown below-

overall wickets taken
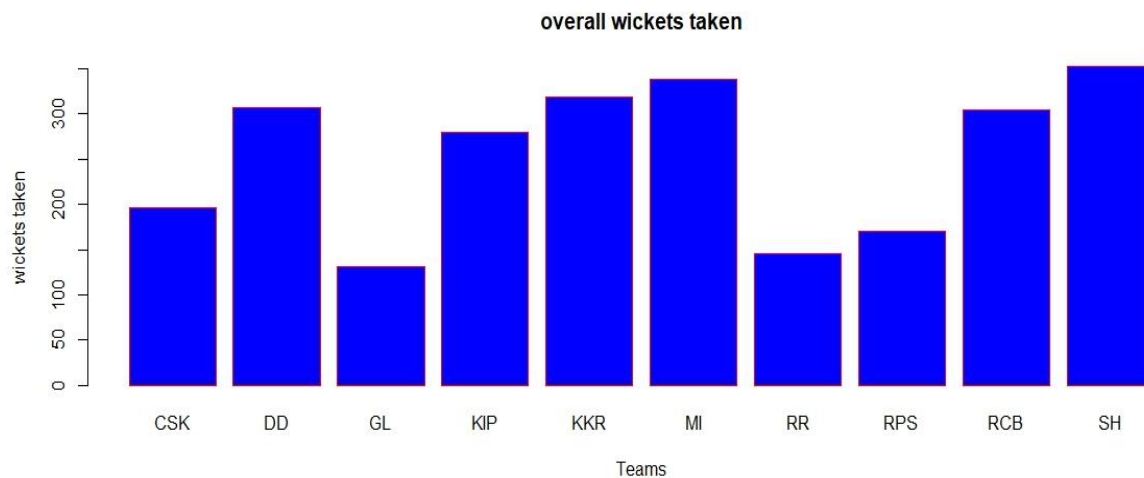
- Visualizing player's strike rate according to their respective age.
- This visualization is plotted using scatter plot which is as shown below –

- Visualizing overall runs scored for each IPL tournament conducted from 2016 to 2019.
- This visualization is done using a line graph which is as shown below –

- Visualizing the total number of the fifties scored by each team from the year 2016 to 2019.
- This visualization is plotted using a pie chart which is as given below

chart to show number of Half centuries by each team from 2016

KIP 11%
GL 7%
DD 11%
KKR 14%
CSK 7%
MI 12%
SH 14%
RR 5%
RPS 5%
RCB 13%
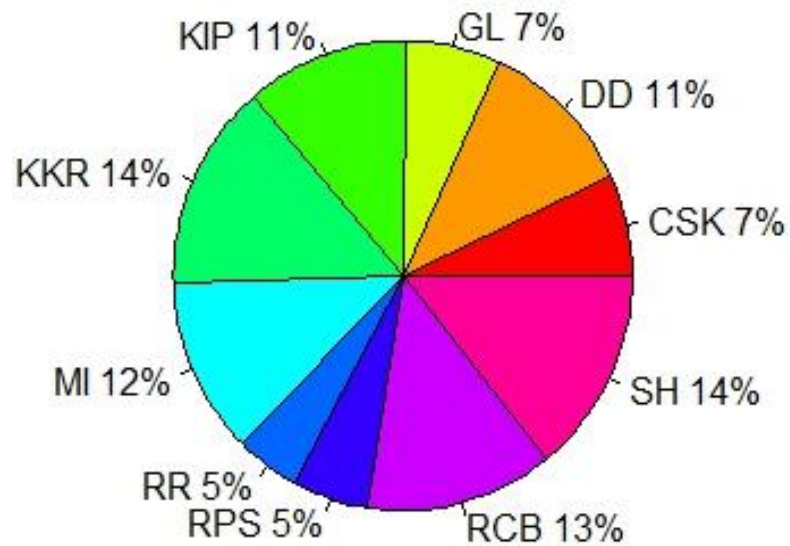
SQL QUERY AND REPORT FOR SSRS

Foreign Players to Invest Money Query (Bar Graph)

SELECT replace(player_Dim.nationality,char(160),'') as Nation,

isnull(sum (playerTournamentPerformance_Fact.runsScored),0) as [Runs Scored]

FROM   player_Dim

INNER JOIN

playerTournamentPerformance_Fact ON player_Dim.playerKey = playerTournamentPerformance_Fact.playerKey where replace(player_Dim.nationality,char(160),'') <> 'India'

group by replace(player_Dim.nationality,char(160),'')

Description – In this query, we are summing up the runs scored in all four seasons and grouping it by player nationality.

Foreign Players to Invest Money
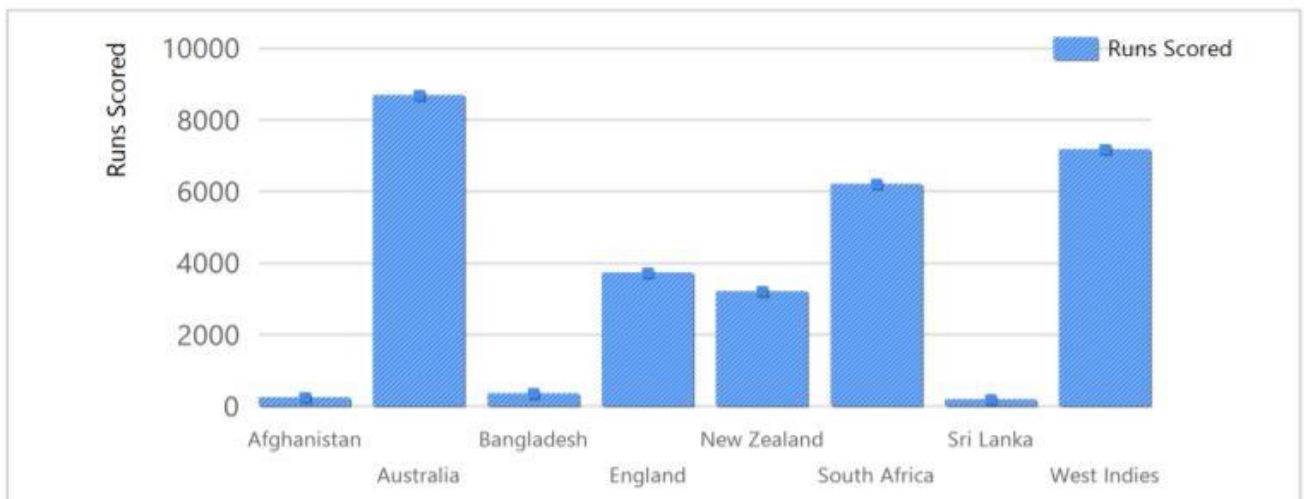
Player Batting Performance Query

SELECT player_Dim.playerName,

isnull(Sum (playerTournamentPerformance_Fact.battingInnings),0) as [Batting Innings],

isnull(Sum (playerTournamentPerformance_Fact.runsScored),0) as [Total Runes Scored],

isnull(Sum (playerTournamentPerformance_Fact.ballsFaced),0) as [Balls Faced],

isnull(Sum (playerTournamentPerformance_Fact.numberOfCenturies),0) as [Number Of Centuries],

isnull(Sum (playerTournamentPerformance_Fact.numberOfHalfCenturies),0) as [Number Of Half Centuries],

isnull(Sum (playerTournamentPerformance_Fact.numberOfSixes),0) as [Number Of Sixes],

isnull(Sum (playerTournamentPerformance_Fact.numberOfBoundaries),0) as [Number Of Boundaries]

FROM player_Dim INNER JOIN

playerTournamentPerformance_Fact ON player_Dim.playerKey = playerTournamentPerformance_Fact.playerKey

group by player_Dim.playerName,player_Dim.playerID

Description – Here we are summing up all the respective attributes of a batsman in four IPL seasons and grouping it by player name and ID.

Player Batting Performance in Overall Seasons

Player Name	Batting Innings	Total Runes Scored	Number Of Sixes	Number Of Boundaries	Number Of Centuries	Number Of Half Centuries	Balls Faced
A Ashish Reddy	3	47	4	2	0	0	29
A Choudhary		25	1	1	0	0	20
A Dananjaya	1	4	0	0	0	0	5
A Mishra	13	75	2	4	0	0	92
A Nehra	4	1	0	0	0	0	1
A Zampa	2	5	0	1	0	0	8
AB de Villiers	49	1825	109	139	1	18	1132
AB Dinda	1	7	0	1	0	0	7
Abhishek Sharma	6	72	5	4	0	0	42
AD Hales		148	6	13	0	0	118
AD Mathews	2	32	1	2	0	0	27
AD Nath	10	90	2	7	0	0	98
AD Russell	35	1014	98	58	0	5	534
AF Milne	3	7	0	0	0	0	12
AJ Finch	34	826	38	76	0	7	579
AJ Turner	4	3	0	0	0	0	10
AJ Tye	12	85	4	6	0	0	70
AM Rahane	57	1625	32	173	1	10	1300
Ankit Sharma	2	28	1	2	0	0	18

Player Bowling Performance Query

SELECT player_Dim.playerName,
sum (playerTournamentPerformance_Fact.wicketsTaken) as

Wickets_Taken,

Sum (playerTournamentPerformance_Fact.catchesTaken) as
Catches_Taken,

sum (playerTournamentPerformance_Fact.runsConceded) as
Runes_Conceded,

sum (playerTournamentPerformance_Fact.maidensBowled) as
Maidens_Bowled,

sum (playerTournamentPerformance_Fact.oversBowled) as
Overs_Bowled,

sum (playerTournamentPerformance_Fact.bowlingInnings) as
BowlingInnings

FROM   player_Dim INNER JOIN

playerTournamentPerformance_Fact ON player_Dim.playerKey =
playerTournamentPerformance_Fact.playerKey

group by player_Dim.playerName,player_Dim.playerID

Description – Here we are summing up all the respective attributes of a
bowler in four IPL seasons and grouping it by player name and ID.

Player Overall Season's Bowling Performance

Player Name	Wickets Taken	Maidens Bowled	Runes Conceded	Overs Bowled	Bowling Innings	Catches Taken
A Ashish Reddy	1	0	39	4	2	1
A Choudhary	5	2	144	16	5	0
A Dananjaya	0	0	47	4	1	0
A Mishra	46	2	1221	161	49	7
A Nehra	17	1	396	47	14	2
A Zampa	19	0	278	37	11	0
AB de Villiers						37
AB Dinda	12	2	350	40	12	0
Abhishek Sharma	1	0	21	2	2	2
AD Hales						2
AD Mathews	0	0	56	5	2	2
AD Nath						2
AD Russell	39	0	933	104	39	14
AF Milne	4		177	18	5	5
AJ Finch						12
AJ Turner						0
AJ Tye	39	0	822	99	26	7
AM Rahane						22
Ankit Sharma	1	0	111	13	5	1
Ankit Soni	2		144	18	6	1
Anureet Singh	1		111	12	6	1

Player Performance Based on Nationality Query

SELECT replace(player_Dim.nationality,char(160),'') as Nation,

isnull(sum (playerTournamentPerformance_Fact.runsScored),0) as [Runs Scored] ,

isnull (sum(playerTournamentPerformance_Fact.numberOfCenturies),0)as [Number Of Centuries],

isnull(sum (playerTournamentPerformance_Fact.numberOfHalfCenturies),0)as [Number Of HalfCenturies],

isnull(sum (playerTournamentPerformance_Fact.numberOfBoundaries),0) as [Number Of Boundaries],

isnull(sum (playerTournamentPerformance_Fact.numberOfSixes),0) as [Number Of Sixes],

isnull(sum (playerTournamentPerformance_Fact.wicketsTaken),0) as[Wickets Taken]

FROM   player_Dim INNER JOIN

playerTournamentPerformance_Fact ON player_Dim.playerKey = playerTournamentPerformance_Fact.playerKey

group by replace(player_Dim.nationality,char(160),'') order by 1 asc

Description – The overall performance of a player as a batsman and a bowler based on the nationality is shown above. We are summing up all the four season's attributes and grouping it based on nationality.

Player Performance Based on Nationality

Nation	Runs Scored	Number Of Centuries	Number Of Half Centuries	Number Of Boundaries	Number Of Sixes	Wickets Taken
Afghanistan	249	0	0	18	16	83
Australia	8691	4	53	779	377	253
Bangladesh	364	0	1	33	10	58
England	3727	2	18	344	151	119
India	43714	11	242	3980	1552	1491
New Zealand	3216	0	16	272	148	137
South Africa	6210	4	41	527	283	182
Sri Lanka	193	0	0	9	13	36
West Indies	7176	2	41	582	449	183

5. INCLUDE XML AND SCHEMA

1.a. XML document of Player Dimension.

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Player_Info xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <data>
        <ID>1001</ID>
        <Player>A Ashish Reddy</Player>
        <Nationality>India</Nationality>
        <Player_skill>Batsman</Player_skill>
        <Batting_hand>Right</Batting_hand>
        <Age>28</Age>
    </data>
    <data>
        <ID>1002</ID>
        <Player>A Choudhary</Player>
        <Nationality>India</Nationality>
        <Player_skill>Bowler</Player_skill>
        <Batting_hand>Right</Batting_hand>
        <Age>29</Age>
    </data>
    <data>
        <ID>1003</ID>
        <Player>A Dananjaya</Player>
        <Nationality> Sri Lanka</Nationality>
        <Player_skill>All Rounder</Player_skill>
        <Batting_hand>Left</Batting_hand>
        <Age>26</Age>
    </data>
    <data>
        <ID>1004</ID>
        <Player>A Mishra</Player>
        <Nationality>India</Nationality>
        <Player_skill>All Rounder</Player_skill>
        <Batting_hand>Right</Batting_hand>
        <Age>37</Age>
    </data>
    <data>
        <ID>1005</ID>
        <Player>A Nehra</Player>
```

1.b. XSD document of Player Dimension.

```xml
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<xsd:element name="Player_Info">
<xsd:complexType><!--Containing child elements-->
<xsd:sequence><!--child elements appear in a sequence-->
 <xsd:element ref="data" minOccurs="1" maxOccurs="unbounded" />
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="data">
<xsd:complexType>
<xsd:sequence>
 <xsd:element ref="ID" minOccurs="1" maxOccurs="unbounded" /><!--minOccurs and maxOccurs both are 1 by default-->
 <xsd:element ref="Player" minOccurs="1" maxOccurs="unbounded" />
 <xsd:element ref="Nationality" minOccurs="1" maxOccurs="unbounded" />
 <xsd:element ref="Player_skill" minOccurs="1" maxOccurs="unbounded" />
 <xsd:element ref="Batting_hand" minOccurs="1" maxOccurs="unbounded" />
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<!--Now writing simple type elements containing data-->
<xsd:element name="ID" type="xsd:int"/>
<xsd:element name="Player" type="xsd:string"/>
<xsd:element name="Nationality" type="xsd:string" />
<xsd:element name="Player_skill" type="xsd:string" />
<xsd:element name="Batting_hand" type="xsd:string" />
<xsd:element name="Age" type="xsd:int"/>
</xsd:schema>
```

1.c. Validate XML against XSD of Player Dimension.

XML

```
                <Player_skill>Bowler</Player_skill>
                <Batting_hand>Right</Batting_hand>
                <Age>29</Age>
        </data>
        <data>
                <ID>1267</ID>
                <Player>Yuvraj Singh</Player>
                <Nationality>India</Nationality>
                <Player_skill>Batsman</Player_skill>
                <Batting_hand>Left</Batting_hand>
                <Age>37</Age>
        </data>
        <data>
                <ID>1268</ID>
                <Player>Z Khan</Player>
                <Nationality>India</Nationality>
                <Player_skill>Bowler</Player_skill>
                <Batting_hand>Right</Batting_hand>
                <Age>41</Age>
        </data>
</Player_Info>
```

XSD Schema

```
<xsd:element name="data">
<xsd:complexType>
<xsd:sequence>
<xsd:element ref="ID" minOccurs="1" maxOccurs="unbounded" /><!--minOcc
<xsd:element ref="Player" minOccurs="1" maxOccurs="unbounded" />
<xsd:element ref="Nationality" minOccurs="1" maxOccurs="unbounded" />
<xsd:element ref="Player_skill" minOccurs="1" maxOccurs="unbounded" /;
<xsd:element ref="Batting_hand" minOccurs="1" maxOccurs="unbounded" /;
<xsd:element ref="Age" minOccurs="1" maxOccurs="unbounded" />
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<!--Now writing simple type elements containing data-->
<xsd:element name="ID" type="xsd:int"/>
<xsd:element name="Player" type="xsd:string"/>
<xsd:element name="Nationality" type="xsd:string" />
<xsd:element name="Player_skill" type="xsd:string" />
<xsd:element name="Batting_hand" type="xsd:string" />
<xsd:element name="Age" type="xsd:int"/>
</xsd:schema>
```

Check XML Well Formed   Check XSD Validity

Validate XML against XSD

Result

The XML is Well Formed and Valid.

2.a. XML document of Team Dimension.

2.b. XSD document of Team Dimension.

```xml
1  <?xml version="1.0"?>
2  <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
3  <xsd:element name="Ipl_Teams">
4  <xsd:complexType><!--Containing child elements-->
5  <xsd:sequence><!--child elements appear in a sequence-->
6  <xsd:element ref="data" minOccurs="1" maxOccurs="unbounded" />
7  </xsd:sequence>
8  </xsd:complexType>
9  </xsd:element>
10 <xsd:element name="data">
11 <xsd:complexType>
12 <xsd:sequence>
13 <xsd:element ref="Team_id" minOccurs="1" maxOccurs="unbounded" /><!--minOccurs and maxOccurs both are 1 by default-->
14 <xsd:element ref="Team" minOccurs="1" maxOccurs="unbounded" />
15 <xsd:element ref="Team_Owner" minOccurs="1" maxOccurs="unbounded" />
16 <xsd:element ref="AvgNumberOfCoaches" minOccurs="1" maxOccurs="unbounded" />
17 <xsd:element ref="IPLWinningHistory" minOccurs="1" maxOccurs="unbounded" />
18 <xsd:element ref="FoundingYear" minOccurs="1" maxOccurs="unbounded" />
19 <xsd:element ref="StateRepresentation" minOccurs="1" maxOccurs="unbounded" />
20 </xsd:sequence>
21 </xsd:complexType>
22 </xsd:element>
23 <!--Now writing simple type elements containing data-->
24 <xsd:element name="Team_id" />
25 <xsd:element name="Team"  type="xsd:string" />
26 <xsd:element name="Team_Owner" type="xsd:string" />
27 <xsd:element name="AvgNumberOfCoaches" type="xsd:int" />
28 <xsd:element name="IPLWinningHistory" type="xsd:int" />
29 <xsd:element name="FoundingYear" type="xsd:int"/>
30 <xsd:element name="StateRepresentation" type="xsd:string"/>
31 </xsd:schema>
```

2.c. Validate XML against XSD of Team Dimension.

```xml
        </data>
        <data>
                <Team_id>T009</Team_id>
                <Team>Chennai Super Kings</Team>
                <Team_Owner>India Cements</Team_Owner>
                <AvgNumberOfCoaches>1</AvgNumberOfCoaches>
                <IPLWinningHistory>2</IPLWinningHistory>
                <FoundingYear>2008</FoundingYear>
                <StateRepresentation>Tamil Nadu</StateRepresentation>
        </data>
        <data>
                <Team_id>T0010</Team_id>
                <Team>Rajasthan Royals</Team>
                <Team_Owner>Manoj Badale</Team_Owner>
                <AvgNumberOfCoaches>5</AvgNumberOfCoaches>
                <IPLWinningHistory>1</IPLWinningHistory>
                <FoundingYear>2008</FoundingYear>
                <StateRepresentation>Rajasthan</StateRepresentation>
        </data>
</Ipl_Teams>
```

```xml
<xsd:sequence>
<xsd:element ref="Team_id" minOccurs="1" maxOccurs="unbounded" /><!--minOccurs ar
<xsd:element ref="Team" minOccurs="1" maxOccurs="unbounded" />
<xsd:element ref="Team_Owner" minOccurs="1" maxOccurs="unbounded" />
<xsd:element ref="AvgNumberOfCoaches" minOccurs="1" maxOccurs="unbounded" />
<xsd:element ref="IPLWinningHistory" minOccurs="1" maxOccurs="unbounded" />
<xsd:element ref="FoundingYear" minOccurs="1" maxOccurs="unbounded" />
<xsd:element ref="StateRepresentation" minOccurs="1" maxOccurs="unbounded" />
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<!--Now writing simple type elements containing data-->
<xsd:element name="Team_id" />
<xsd:element name="Team"  type="xsd:string" />
<xsd:element name="Team_Owner"  type="xsd:string" />
<xsd:element name="AvgNumberOfCoaches" type="xsd:int" />
<xsd:element name="IPLWinningHistory" type="xsd:int" />
<xsd:element name="FoundingYear" type="xsd:int"/>
<xsd:element name="StateRepresentation" type="xsd:string"/>
</xsd:schema>
```

Check XML Well Formed   Check XSD Validity

Validate XML against XSD

Result

The XML is Well Formed and Valid.

3.a. XML documents of Tournament Dimension.

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Tournament_Info xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <tournament>

        <tournamentID>I001</tournamentID>
        <tournamentname>IPL 2016</tournamentname>
        <most_match_time_type>day night</most_match_time_type>
        <secondMost_match_time_type>day</secondMost_match_time_type>
        <overallPlayerAuctionExpenditureInCrore>329</overallPlayerAuctionExpenditureInCrore>
        <viewersInMillions>335</viewersInMillions>
    </tournament>
    <tournament>
        <tournamentID>I001</tournamentID>
        <tournamentname>IPL 2017</tournamentname>
        <most_match_time_type>night</most_match_time_type>
        <secondMost_match_time_type>day night</secondMost_match_time_type>
        <overallPlayerAuctionExpenditureInCrore>384</overallPlayerAuctionExpenditureInCrore>
        <viewersInMillions>411</viewersInMillions>
    </tournament>
    <tournament>
        <tournamentID>I003</tournamentID>
        <tournamentname>IPL 2018</tournamentname>
        <most_match_time_type>day night</most_match_time_type>
        <secondMost_match_time_type>day</secondMost_match_time_type>
        <overallPlayerAuctionExpenditureInCrore>431</overallPlayerAuctionExpenditureInCrore>
        <viewersInMillions>700</viewersInMillions>
    </tournament>
    <tournament>
        <tournamentID>I004</tournamentID>
        <tournamentname>IPL 2019</tournamentname>
        <most_match_time_type>day night</most_match_time_type>
        <secondMost_match_time_type>day</secondMost_match_time_type>
        <overallPlayerAuctionExpenditureInCrore>325</overallPlayerAuctionExpenditureInCrore>
        <viewersInMillions>462</viewersInMillions>
    </tournament>
</Tournament_Info>
```

3.b. XSD document of Tournament Dimension.

```xml
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<xsd:element name="Tournament_Info">
<xsd:complexType><!--Containing child elements-->
<xsd:sequence><!--child elements appear in a sequence-->
<xsd:element ref="tournament" minOccurs="1" maxOccurs="unbounded" />
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="tournament">
<xsd:complexType>
<xsd:sequence>
<xsd:element ref="tournamentID" minOccurs="1" maxOccurs="unbounded" /><!--minOccurs and maxOccurs both are 1 by default-->
<xsd:element ref="tournamentname" minOccurs="1" maxOccurs="unbounded" />
<xsd:element ref="most_match_time_type" minOccurs="1" maxOccurs="unbounded" />
<xsd:element ref="secondMost_match_time_type" minOccurs="1" maxOccurs="unbounded" />
<xsd:element ref="overallPlayerAuctionExpenditureInCrore" minOccurs="1" maxOccurs="unbounded" />
<xsd:element ref="viewersInMillions" minOccurs="1" maxOccurs="unbounded" />
<xsd:element ref="Tournament_Start_Date" minOccurs="1" maxOccurs="unbounded" />
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<!--Now writing simple type elements containing data-->
<xsd:element name="tournamentID" />
<xsd:element name="tournamentname" />
<xsd:element name="most_match_time_type" type="xsd:string" />
<xsd:element name="secondMost_match_time_type" type="xsd:string" />
<xsd:element name="overallPlayerAuctionExpenditureInCrore" type="xsd:int" />
<xsd:element name="viewersInMillions" type="xsd:int"/>
<xsd:element name="Tournament_Start_Date" />
</xsd:schema>
```

3.c. Validate XML against XSD of Tour Dimension.

XML

```
                <overallPlayerAuctionExpenditureInCrore>384</c
                <viewersInMillions>411</viewersInMillions>
        </tournament>
        <tournament>
                <tournamentID>I003</tournamentID>
                <tournamentname>IPL 2018</tournamentname>
                <most_match_time_type>day night</most_match_ti
                <secondMost_match_time_type>day</secondMost_ma
                <overallPlayerAuctionExpenditureInCrore>431</c
                <viewersInMillions>700</viewersInMillions>
                </tournament>
        <tournament>
                <tournamentID>I004</tournamentID>
                <tournamentname>IPL 2019</tournamentname>
                <most_match_time_type>day night</most_match_ti
                <secondMost_match_time_type>day</secondMost_ma
                <overallPlayerAuctionExpenditureInCrore>325</c
                <viewersInMillions>462</viewersInMillions>
        </tournament>
</Tournament_Info>
```

XSD Schema

```
<xsd:element name="tournament">
<xsd:complexType>
<xsd:sequence>
<xsd:element ref="tournamentID" minOccurs="1" maxOccurs="unbounded" />
<xsd:element ref="tournamentname" minOccurs="1" maxOccurs="unbounded"
<xsd:element ref="most_match_time_type" minOccurs="1" maxOccurs="unbou
<xsd:element ref="secondMost_match_time_type" minOccurs="1" maxOccurs=
<xsd:element ref="overallPlayerAuctionExpenditureInCrore" minOccurs="1
<xsd:element ref="viewersInMillions" minOccurs="1" maxOccurs="unbounde
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<!--Now writing simple type elements containing data-->
<xsd:element name="tournamentID" />
<xsd:element name="tournamentname" />
<xsd:element name="most_match_time_type" type="xsd:string" />
<xsd:element name="secondMost_match_time_type" type="xsd:string" />
<xsd:element name="overallPlayerAuctionExpenditureInCrore" type="xsd:i
<xsd:element name="viewersInMillions" type="xsd:int"/>
</xsd:schema>
```

Check XML Well Formed   Check XSD Validity

Validate XML against XSD

Result

The XML is Well Formed and Valid.

4.a. XML document of Player Performance Fact Table.

```
1   <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2   <Player_Performance xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
3       <data>
4           <Playerid>1045</Playerid>
5           <Tournament_ID>I001</Tournament_ID>
6           <Matches>12</Matches>
7           <BattingInnings>7</BattingInnings>
8           <NotOut>4</NotOut>
9           <RunsScored>195</RunsScored>
10          <BattingAverage>65</BattingAverage>
11          <BallsFaced>109</BallsFaced>
12          <BattingStrikeRate>178.89</BattingStrikeRate>
13          <Hundred>0</Hundred>
14          <fifty>1</fifty>
15          <zero>1</zero>
16          <fours>15</fours>
17          <sixes>12</sixes>
18          <BowlingInnings>12</BowlingInnings>
19          <OversBowled>44</OversBowled>
20          <MaidensBowled>0</MaidensBowled>
21          <RunsConceded>308</RunsConceded>
22          <WicketsTaken>13</WicketsTaken>
23          <BowlingAverage>23.69</BowlingAverage>
24          <BowlingEconomyRate>7</BowlingEconomyRate>
25          <CatchesTaken>8</CatchesTaken>
26          <Team_id>T001</Team_id>
27      </data>
28      <data>
29          <Playerid>1045</Playerid>
30          <Tournament_ID>I002</Tournament_ID>
31          <Matches>9</Matches>
32          <BattingInnings>9</BattingInnings>
33          <NotOut>4</NotOut>
34          <RunsScored>154</RunsScored>
35          <BattingAverage>30.8</BattingAverage>
36          <BallsFaced>94</BallsFaced>
37          <BattingStrikeRate>163.82</BattingStrikeRate>
```

4.b. XSD document of PlayerPerformance_Fact.


```xml
1   <?xml version="1.0"?>
2   <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
3   <xsd:element name="Player_Performance">
4   <xsd:complexType><!--Containing child elements-->
5   <xsd:sequence><!--child elements appear in a sequence-->
6   <xsd:element ref="data" minOccurs="1" maxOccurs="unbounded" />
7   </xsd:sequence>
8   </xsd:complexType>
9   </xsd:element>
10  <xsd:element name="data">
11  <xsd:complexType>
12  <xsd:sequence>
13  <xsd:element ref="Playerid" minOccurs="1" maxOccurs="unbounded" /><!--minOccurs and maxOccurs both are 1 by default-->
14  <xsd:element ref="Tournament_ID" minOccurs="1" maxOccurs="unbounded" />
15  <xsd:element ref="Matches" minOccurs="1" maxOccurs="unbounded" />
16  <xsd:element ref="BattingInnings" minOccurs="1" maxOccurs="unbounded" />
17  <xsd:element ref="NotOut" minOccurs="1" maxOccurs="unbounded" />
18  <xsd:element ref="RunsScored" minOccurs="1" maxOccurs="unbounded" />
19  <xsd:element ref="HighestScore" minOccurs="1" maxOccurs="unbounded" />
20  <xsd:element ref="BattingAverage" minOccurs="1" maxOccurs="unbounded" />
21  <xsd:element ref="BallsFaced" minOccurs="1" maxOccurs="unbounded" />
22  <xsd:element ref="BattingStrikeRate" minOccurs="1" maxOccurs="unbounded" />
23  <xsd:element ref="Hundred" minOccurs="1" maxOccurs="unbounded" />
24  <xsd:element ref="fifty" minOccurs="1" maxOccurs="unbounded" />
25  <xsd:element ref="zero" minOccurs="1" maxOccurs="unbounded" />
26  <xsd:element ref="fours" minOccurs="1" maxOccurs="unbounded" />
27  <xsd:element ref="sixes" minOccurs="1" maxOccurs="unbounded" />
28  <xsd:element ref="BowlingInnings" minOccurs="1" maxOccurs="unbounded" />
29  <xsd:element ref="OversBowled" minOccurs="1" maxOccurs="unbounded" />
30  <xsd:element ref="MaidensBowled" minOccurs="1" maxOccurs="unbounded" />
31  <xsd:element ref="RunsConceded" minOccurs="1" maxOccurs="unbounded" />
32  <xsd:element ref="WicketsTaken" minOccurs="1" maxOccurs="unbounded" />
33  <xsd:element ref="BowlingAverage" minOccurs="1" maxOccurs="unbounded" />
34  <xsd:element ref="BowlingEconomyRate" minOccurs="1" maxOccurs="unbounded" />
35  <xsd:element ref="CatchesTaken" minOccurs="1" maxOccurs="unbounded" />
36  <xsd:element ref="Team_id" minOccurs="1" maxOccurs="unbounded" />
```

4.c. Validate XML against XSD of PlayerPerformance_Dim.

XML

```xml
            <RunsScored>4</RunsScored>
            <BattingAverage>2</BattingAverage>
            <BallsFaced>8</BallsFaced>
            <BattingStrikeRate>50</BattingStrikeRate>
            <Hundred>0</Hundred>
            <fifty>0</fifty>
            <zero>0</zero>
            <fours>0</fours>
            <sixes>0</sixes>
            <BowlingInnings>1</BowlingInnings>
            <OversBowled>1</OversBowled>
            <MaidensBowled>0</MaidensBowled>
            <RunsConceded>9</RunsConceded>
            <WicketsTaken>0</WicketsTaken>
            <BowlingAverage>-</BowlingAverage>
            <BowlingEconomyRate>9</BowlingEconomyRate>
            <CatchesTaken>0</CatchesTaken>
            <Team_id>T001</Team_id>
        </data>
</Player_Performance>
```

XSD Schema

```xml
<xsd:element name="RunsScored" type="xsd:int"/>
<xsd:element name="HighestScore" />
<xsd:element name="BattingAverage" type="xsd:decimal" />
<xsd:element name="BallsFaced" type="xsd:int" />
<xsd:element name="BattingStrikeRate" type="xsd:decimal" />
<xsd:element name="Hundred" type="xsd:int"/>
<xsd:element name="fifty" type="xsd:int" />
<xsd:element name="zero" type="xsd:int" />
<xsd:element name="fours" type="xsd:int" />
<xsd:element name="sixes" type="xsd:int"/>
<xsd:element name="BowlingInnings" type="xsd:int" />
<xsd:element name="OversBowled" type="xsd:int" />
<xsd:element name="MaidensBowled" type="xsd:int" />
<xsd:element name="RunsConceded" type="xsd:int"/>
<xsd:element name="WicketsTaken" type="xsd:int" />
<xsd:element name="BowlingAverage" type="xsd:decimal" />
<xsd:element name="BowlingEconomyRate" type="xsd:decimal" />
<xsd:element name="CatchesTaken" type="xsd:int" />
<xsd:element name="Team_id" type="xsd:int" />
</xsd:schema>
```

Check XML Well Formed   Check XSD Validity

Validate XML against XSD

Result

The XML is Well Formed and Valid.

6. GRAPH DATABASES

PLAYER DIMENSION

```
$ Load csv with headers from "file:///PLY.csv" as row Create(c:PLY) set
  c=row{playerKey:row.playerKey,Pid:row.playerID,Name:row.playerName,Age:row.playerAge,Nati
  on:row.nationality,Skill:row.playerSkill,hand:row.battingHand}return c
```


TEAM DIMENSION

```
$ Load csv with headers from "file:///TEAM.csv" as row Create(s:TEAM) set
  s=row{teamKey:row. teamKey, teamID:row.teamID,Name:row.teamName,OWNER:row.ownerName,
  numberOfCoaches:row.numberOfCoaches,foundingYear:row.foundingYear,
  stateRepresentation:row.stateRepresentation, winningHistory:row.winningHistory}return s
```

TOURNAMENT DIMENSION

```
$ Load csv with headers from "file:///TOR.csv" as row Create(p:TORN) set
  p=row{tourKey:row.tournamentKey,TournamentID:row.tournamentID,Name:row.tournamentName,Mos
  tMatchTimeType:row.mostMatchTimeType,secondMostMatchTimeType:row.secondMostMatchTimeType,
  AuctionExpenditure:row.playerAuctionExpenditure,viewersInMillions:row.viewersInMillions}r
  eturn p
```


PLAYER PERFORMANCE FACT

```
$ Load csv with headers from "file:///Per_FACT.csv" as row Create(f:FACT) set
  f=row{playerKey:row.playerKey,
  teamKey:row.teamKey,tournamentKey:row.tournamentKey,maches:row.maches,battingInnings:row
  .battingInnings,NtOut:row.NtOut,runsScored:row.runsScored,battingAverage:row.battingAver
  age,ballsFaced:row.ballsFaced,battingStrikeRate:row.battingStrikeRate,numberOfCenturies:
  row.numberOfCenturies,numberOfHalfCenturies:row.numberOfHalfCenturies,numberOfDuckouts:r
  ow.numberOfDuckouts,numberOfBoundaries:row.numberOfBoundaries,numberOfSixes:row.numberOf
```

```
$ Load csv with headers from "file:///Per_FACT.csv" as row Create(f:FACT) set f=row{player...
```
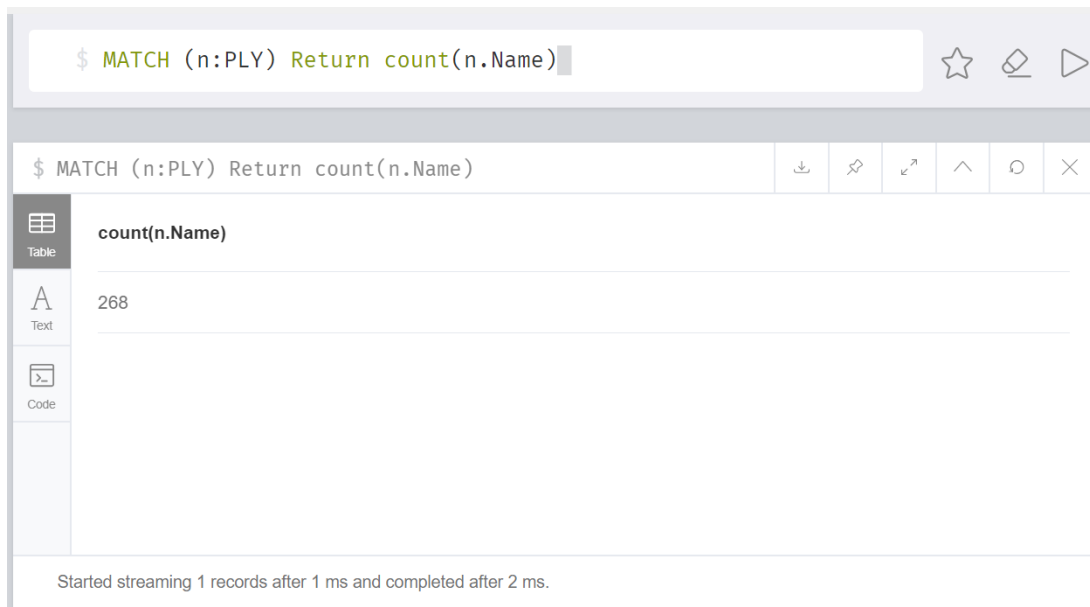
Graph to find players whose Name starts with M

Graph to count number of players

GRAPH FOR CYPHER QUERY

Graph to find player of maximum and minimum age

```
$ MATCH (n:PLY) Return max(n.Age), min(n.Age)
```

```
$ MATCH (n:PLY) Return max(n.Age), min(n.Age)
```

max(n.Age)	min(n.Age)
"48"	"17"

Started streaming 1 records after 3 ms and completed after 3 ms.

Graph to Plot relation between Player, Fact and Team
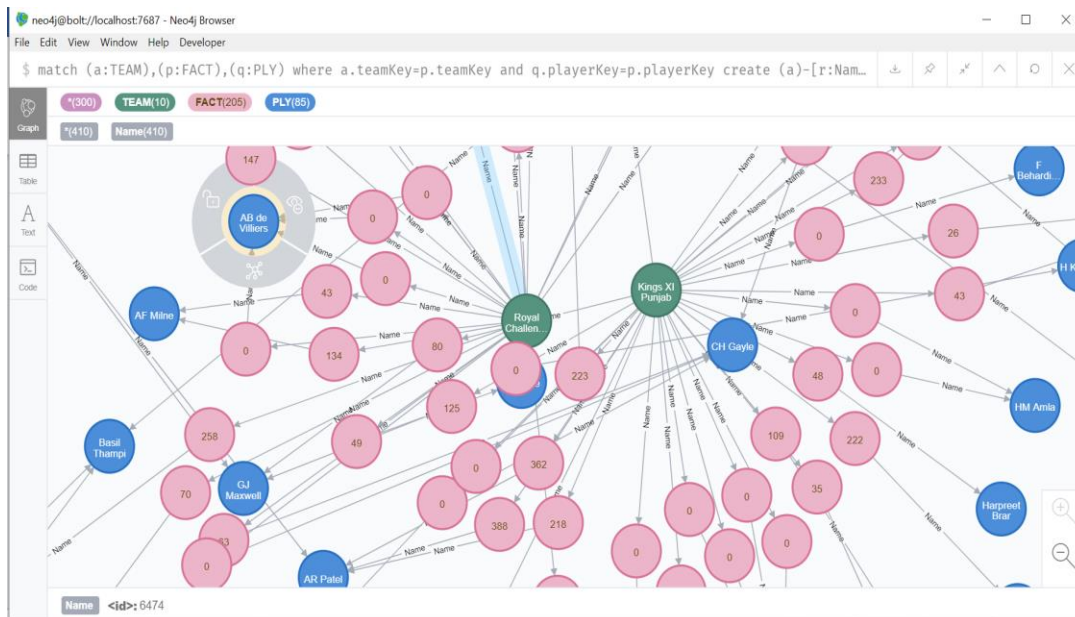
Graph to Plot number of boundaries in each season.
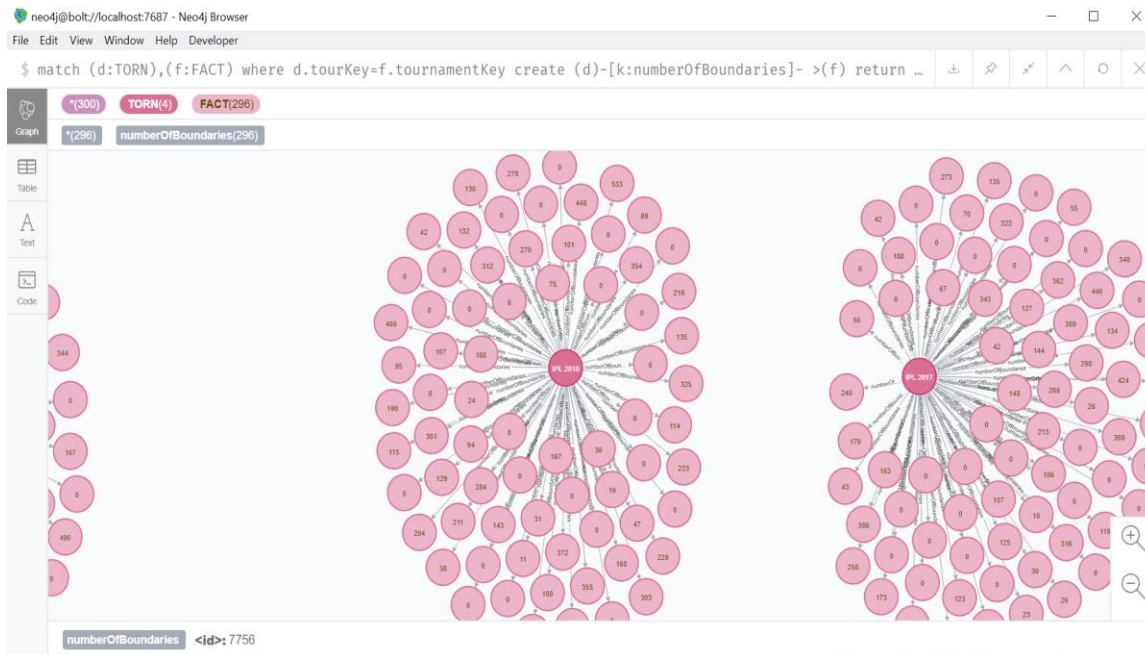
6.1. COMPARISON TO RELATIONAL DATABASES

- The major difference between the Relational database and Graph database is that the Relational database has highly structured data stored in tables and the Graph database deals with an edge, node or attribute.

- As in the Relational model, there is entity i.e. table and its attributes as column name, references to other rows and tables are indicated by referring to primary key attributes via foreign key columns. While in Neo4j, we give a primary or unique key by creating constraints.

- In the Relational database, the concept of the foreign keys is important and plays an important role in joins, as in Neo4j, relationships are used in place of foreign keys or join tables. Relationships can only be between existing nodes, so in that respect, you cannot create a dangling relationship pointing to nothing, nor can you delete a node and leave a dangling relationship.

- In Neo4j, we could retrieve data very fast by accessing the respective node on contrary in relational we retrieve data by accessing select query on the table.

- While working we realized that cypher query is easier to work once we understand the concept behind, compared to SQL. As in Neo4j, the relationships could be formed using constraints alone rather in a relational database there are various concepts of surrogate keys, foreign keys etc.

7. CONCLUSIONS

- In this project, we have made visualizations based on team, tournament, player information, so that the analyzations would be applied to the real-time sport to choose the next best player based on these factors.
- The reports show intricate details on players batting and bowling performance based on IPL season, team and player information. which plays a very important role for investors and sponsors to invest money in these players etc.
- All in all, using data warehousing, the number of visualizations can be done on IPL or any other sport which would give critical information that gives an intuitive analysis of player performance.

8. BIBLIOGRAPHY

- Kimball, R. & Ross, M. (2013). The Data Warehouse Toolkit: The Definitive Guide to dimensional modelling. 3rd Edition. John wiley & sons, Inc.
- Connolly, T, & Begg, C. (2015). Database Systems: A Practical Approach to Design, Implementation, and Management, 6th Edition.

APPENDIX A – VISUALIZATIONS CODE

LIBRARIES REQUIRED FOR VISUALIZATIONS

```
library(tidyr)

library("stringr")

library(RODBC)

library(car)

library(dplyr)

library(ggplot2)
```

CONNECTION STRING WHICH IS COMMON FOR ALL VISUALIZATIONS

```
conn <- odbcDriverConnect('driver={SQL
Server};server=SANTHOSH\\SQLEXPRESS;database=IPL_DW;trusted_
connection=true')
```

BATTING AVERAGE VS PLAYER'S AGE

```
agePlayerBattingAverage        =        sqlQuery(conn,        "SELECT
pd.playerAge,pf.playerKey,pf.tournamentKey,pf.battingAverage
        FROM player_Dim pd
        LEFT JOIN playerTournamentPerformance_Fact pf
        ON pf.playerKey=pd.playerKey
        ")
agePlayerBattingAverage<-agePlayerBattingAverage            %>%
replace_na(list(playerAge=0,battingAverage=0))
scatterplot(agePlayerBattingAverage$battingAverage                ~
agePlayerBattingAverage$playerAge,            data            =
agePlayerBattingAverage,xlab = "age",ylab = "batting average",
        smoother = FALSE, grid = FALSE, frame = FALSE)
```

BATTING AND BOWLING PERFORMANCE OF EVERY IPL TOURNAMENT

```
tournamentPerformance = sqlQuery(conn, "SELECT
td.tournamentName,pf.runsScored,pf.wicketsTaken,pf.playerKey,pf.tourn
amentKey
            FROM tournament_Dim td
            LEFT JOIN playerTournamentPerformance_Fact pf
            ON pf.tournamentKey=td.tournamentKey
            ")
tournamentrPerformance<-tournamentPerformance %>%
replace_na(list(runsScored=0,wicketsTaken=0))
only2016<-
subset(tournamentrPerformance,tournamentrPerformance$tournamentNa
me=="IPL 2016")
only2017<-
subset(tournamentrPerformance,tournamentrPerformance$tournamentNa
me=="IPL 2017")
only2018<-
subset(tournamentrPerformance,tournamentrPerformance$tournamentNa
me=="IPL 2018")
only2019<-
subset(tournamentrPerformance,tournamentrPerformance$tournamentNa
me=="IPL 2019")
runScoredValues<-
c(sum(only2016$runsScored),sum(only2017$runsScored),sum(only2018
$runsScored),sum(only2019$runsScored))
wicketsTaken<-
c(sum(only2016$wicketsTaken),sum(only2017$wicketsTaken),sum(only2
018$wicketsTaken),sum(only2019$wicketsTaken))
par(mfrow=c(2,1))
plot(runScoredValues,type="o",col="red",xlab="2016 to 2019
",ylab="overall runs Scored by players",main = "runscored vs ipl
seasons")
plot(wicketsTaken,type="o",col="blue",xlab="2016 to 2019 ",ylab="overall
wickets taken",main = "wickets taken vs ipl seasons")
```

WICKETS TAKEN BY ALL TEAMS

```
#bar graph
iplTeam = sqlQuery(conn, "SELECT
td.teamName,pf.wicketsTaken,pf.playerKey,pf.tournamentKey
            FROM team_Dim td
            LEFT JOIN playerTournamentPerformance_Fact pf
            ON pf.teamKey=td.teamKey
            ")
#replace all the NA with 0
newiplTeam<-iplTeam %>% replace_na(list(wicketsTaken=0))
#use aggregate function to calculate the total wickets based on teams
aggResult<-
aggregate(newiplTeam$wicketsTaken,by=list(Category=newiplTeam$tea
mName),FUN=sum)
#use spread to make the data horizontal
updatedIPLwickets<-spread(aggResult,"Category","x",1:2,fill = NA)
#plot bar graph
values<-c(updatedIPLwickets$`Chennai Super
Kings`,updatedIPLwickets$`Delhi Daredevils`,
      updatedIPLwickets$`Gujarat Lions`,updatedIPLwickets$`Kings XI
Punjab`,
      updatedIPLwickets$`Kolkata Knight
Riders`,updatedIPLwickets$`Mumbai Indians`,
      updatedIPLwickets$`Rajasthan Royals`,updatedIPLwickets$`Rising
Pune Supergiant`,
      updatedIPLwickets$`Royal Challengers
Bangalore`,updatedIPLwickets$`Sunrisers Hyderabad`)
names<-c("CSK","DD","GL","KIP","KKR","MI","RR","RPS","RCB","SH")
barplot(values,names.arg=names,xlab="Teams",ylab="wickets
taken",col="blue",main="overall wickets taken",border="red")
```

NUMBER OF HALF CENTURIES SCORED BY EACH TEAM

```
playerTeamHalfCenturies = sqlQuery(conn, "SELECT
td.teamName,pf.numberOfHalfCenturies,pf.playerKey,pf.tournamentKey
          FROM team_Dim td
          LEFT JOIN playerTournamentPerformance_Fact pf
          ON pf.teamKey=td.teamKey
          ")
newplayerTeamHalfCenturies<-playerTeamHalfCenturies %>%
replace_na(list(numberOfHalfCenturies=0))
teamNumberOfCenturies<-
aggregate(newplayerTeamHalfCenturies$numberOfHalfCenturies,by=list(
Category=newplayerTeamHalfCenturies$teamName),FUN=sum)
teamNumberOfHalfCenturiesHoriz<-
spread(teamNumberOfCenturies,"Category","x",1:2,fill = NA)
slices<-c(teamNumberOfHalfCenturiesHoriz$`Chennai Super
Kings`,teamNumberOfHalfCenturiesHoriz$`Delhi Daredevils`,
      teamNumberOfHalfCenturiesHoriz$`Gujarat
Lions`,teamNumberOfHalfCenturiesHoriz$`Kings XI Punjab`,
      teamNumberOfHalfCenturiesHoriz$`Kolkata Knight
Riders`,teamNumberOfHalfCenturiesHoriz$`Mumbai Indians`,
      teamNumberOfHalfCenturiesHoriz$`Rajasthan
Royals`,teamNumberOfHalfCenturiesHoriz$`Rising Pune Supergiant`,
      teamNumberOfHalfCenturiesHoriz$`Royal Challengers
Bangalore`,teamNumberOfHalfCenturiesHoriz$`Sunrisers Hyderabad`)
lbls<-c("CSK","DD","GL","KIP","KKR","MI","RR","RPS","RCB","SH")
pct<-round(slices/sum(slices)*100)
lbls<-paste(lbls,pct)
lbls<-paste(lbls,"%",sep="")
par(mfrow=c(1,1))
pie(slices,labels=lbls,col=rainbow(length(lbls)),main="pie chart to show
number of Half centuries by each team from 2016 to 2019" )
```

CQL QUERY AND GRAPH FOR NEO4J

Player Dimension

Load csv with headers from "file:///PLY.csv" as row Create(c:PLY) set c=row{playerKey:row.playerKey,Pid:row.playerID,Name:row.playerName, Age:row.playerAge,Nation:row.nationality,Skill:row.playerSkill,hand:row.battingHand}return c

Create CONSTRAINT on (c:PLY) Assert c. playerKey IS UNIQUE

CQL QUERY AND GRAPH FOR NEO4J

Team Dimension

Load csv with headers from "file:///TEAM.csv" as row Create(s:TEAM) set s=row{teamKey:row. teamKey, teamID:row.teamID,Name:row.teamName,OWNER:row.ownerName, numberOfCoaches:row.numberOfCoaches,foundingYear:row.foundingYear, stateRepresentation:row.stateRepresentation, winningHistory:row.winningHistory}return s

Create CONSTRAINT on (s:TEAM) Assert s.teamKey IS UNIQUE

CQL QUERY AND GRAPH FOR NEO4J

Tournament Dimension

Load csv with headers from "file:///TOR.csv" as row Create(p:TORN) set p=row{tourKey:row.tournamentKey,TournamentID:row.tournamentID,Name:row.tournamentName,MostMatchTimeType:row.mostMatchTimeType, secondMostMatchTimeType:row.secondMostMatchTimeType,AuctionExpenditure:row.playerAuctionExpenditure,viewersInMillions:row.viewersInMillions}return p

Create CONSTRAINT on (p:TORN) Assert p.tourKey IS UNIQUE

Fact Dimension

Load csv with headers from "file:///Per_FACT.csv" as row Create(f:FACT)
set f=row{playerKey:row.playerKey,
teamKey:row.teamKey,tournamentKey:row.tournamentKey,maches:row.
maches,battingInnings:row.battingInnings,NtOut:row.NtOut,runsScored:ro
w.runsScored,battingAverage:row.battingAverage,ballsFaced:row.ballsFa
ced,battingStrikeRate:row.battingStrikeRate,numberOfCenturies:row.num
berOfCenturies,numberOfHalfCenturies:row.numberOfHalfCenturies,num
berOfDuckouts:row.numberOfDuckouts,numberOfBoundaries:row.numbe
rOfBoundaries,numberOfSixes:row.numberOfSixes,bowlingInnings:row.b
owlingInnings,oversBowled:row.oversBowled,maidensBowled:row.maide
nsBowled,runsConceded:row.runsConceded,wicketsTaken:row.wicketsT
aken,bowlingAverage:row.bowlingAverage,bowlingEconomyRate:row.bo
wlingEconomyRate,catchesTaken:row.catchesTaken}return f

Query to find players whose Name starts with M

match (c:PLY)where c.Name starts with "M" return c

Query to count number of players

MATCH (n:PLY) Return count(n.Name)

Query to find player of maximum and minimum age

MATCH (n:PLY) Return max(n.Age), min(n.Age)

Query to Plot relation between Player, Fact and Team

match (a:TEAM),(p:FACT),(q:PLY) where a.teamKey=p.teamKey and
q.playerKey=p.playerKey create (a)-[r:Name]- >(p)- [t:Name]- >(q) return
a,p,q,r,t

Query to Plot number of boundaries in each season(Relation).

match (d:TORN),(f:FACT) where d.tourKey=f.tournamentKey create (d)-
[k:numberOfBoundaries]- >(f) return d,f,k

THANK YOU