ETL\_Project\_Group\_2  
Solving IPL Madness

June 2021

**Team Members:** Srichakra Vegunta & Josh Cullen

## Project Background and Description

Shape, circle

Description automatically generatedTo look outside what has become the new norm in data analysis, Covid Statistics, we searched for an interesting topic through relevant platforms such as Kaggle and ABS.gov.au, landing on some data on the Indian Cricket League 😊 and who does not like, love, or know about cricket

We prepared a data set which could be used to uncover relationships between India Premier League matches, players, best batters/ bowlers & match winners during the period 2008-2016.

We had six CSV files and key information was captured in different tables. The current structure made it very time consuming to derive basic statistics about the players, their performance, matches and their relevance to the team and seasons.

## Extract

Data was loaded into a Jupyter Notebook for analysis and transformation.

**Data Source:** Player.csv

(<https://www.kaggle.com/harsha547/indian-premier-league-csv-dataset> )

**Format:** CSV

**Data Source:** Match.csv

(<https://www.kaggle.com/datasnaek/youtube-new>)

**Format:** CSV

**Data Source:** Season.csv

(<https://www.kaggle.com/harsha547/indian-premier-league-csv-dataset> )

**Format:** CSV

**Data Source:** Team.csv

(<https://www.kaggle.com/harsha547/indian-premier-league-csv-dataset> )

**Format:** CSV

**Data Source:** Indian Premier League.csv

(<https://www.kaggle.com/harsha547/indian-premier-league-csv-dataset> )

**Format:** CSV

## Transform

Pandas was used to clean, filter, join and aggregate our data into the three clean data tables ins SQL.

Final Tables

* Players
* Matches
* Player\_Match

The first port of call was to Create Aggregate Tables from the Season Data to identify Best Players in the Season.

We created summary data frames to group the Best Batters, Best Bowlers and Man of the Series.

Table

Description automatically generated with low confidence Table

Description automatically generated with low confidence Table

Description automatically generated with low confidence

This was swiftly followed by a tranformation leg spin attack to Clean, Create new variables and Merging of the best player aggregate Data to the Player Table to make a new Player Table.

A screenshot of a computer

Description automatically generated with medium confidence

With a left knee drop and a right to left hook swing of the bat, the leg spin attack was set to the boundary to tackle the Final transformation requirements of Merging Team and Season Data to Match to have all information in Match Data table. We also eliminated the need for Team and Season Data whilst retaining unique match ID's.

Graphical user interface, application

Description automatically generated

## Load

After we pulled in the CSV files and loaded them into the data frames, we did an initial connection to the Postgres database using PG admin to store our new data frames in SQL tables. We used the quick database website to create the initial table schema that got loaded into the Postgres database that generated the first set of tables.

Graphical user interface, text

Description automatically generated

## Limitations:

The data period is set between 2008-2016.

## Conclusion

We created three data tables from the original five CSV files to enhance the performance of data analysis and provide key insights about players and matches.

We thought about the user in mind and created new variables that may be helpful (and fun) in analysing player performance by age.