**NBA Salary with GP, FG%, and PPG: ETL Project**

**By: Shamil Tajuddin**

**Objective:**

I have always wondered about player performance in regards to their salary. Salary is a good indicator to see how much a team values a certain player, and I was curious to see how players were performing (offensively), while viewing their salary. However, its oftentimes tedious to search multiple locations for this type of player data. My objective of this project was to create a single database that contains basic offensive data (field goal percentage and points per game) alongside the number of games played and the salary of each player. My dataset includes player data for the current NBA season (2019-2020).

**Key:**

* **gp:** games played
* **fg\_percent:** field goal percentage (field goals made/field goals attempted)
* **ppg:** points per game (total points scored/games played)
* **salary:** how much money the player made in the 2019-2020 season

**Extract:**

The source of my salary data came from Basketball Reference (<https://www.basketball-reference.com/contracts/players.html>), which provided 2019-2020 salary information. The source of my player statistic data came from NBA Stuffer (<https://www.nbastuffer.com/2019-2020-nba-player-stats/>), which provided the data for gp, fg\_percent, and ppg.

**Transform:**

Initially I read the CSV files with Jupyter Notebook. I then realized I needed to clean the dataset, as there were some duplicates in the set. Additionally, I edited the salary Excel file to represent the salaries in the dataset without commas or dollar signs. I then cut unnecessary columns from the player statistic dataset and renamed the important columns for ease of use (renamed to lower case letters).

**Load:**

I then pushed the two pandas dataframes into SQL tables. I created two distinct SQL tables in postgres, and then created a database connection via pandas to the postgres SQL tables. In postgres, I selected each part of the table and joined the two datasets via the primary key (player – unique identifier in both sets). Finally, I queried each table in Jupyter Notebook to ensure that both datasets were functioning and accurate.