**NBA Team Data Analysis – ETL Documentation**

**ETL Objectives:**

* Create new PostgreSQL database in PGAdmin called SportsData, with Schema and ER diagram for 4 tables developed in Quick Database Diagrams.
* Extract all 4 CSV files to a different Pandas Dataframe in Jupyter Notebooks.
* Transform each Dataframe to match to Schema for PostgreSQL SportsData database.
* Load transformed Dataframes to PostgreSQL SportsData, via Pandas function and SQLAlchemy in Jupyter Notebook

**Extract:**

The data used to create the database was found online. nba\_team\_win which holds yearly regular season win/loss record was found in CSV format on Kaggle.com. Players and season stats data was found on data.world in CSV format. The extraction of the data can be seen in this link below.

<https://github.com/jacob-servidio/CWRU-2020-ETL-Project-2/blob/main/NBA.ipynb>

**Transform:**

In order to transform the public data and use it in our study we performed the following:

* Used Pandas functions in Jupyter Notebook to load all three CSV files.
* Reviewed the files and transformed into data frames
* Renamed teams to match across the data
* Rewrote broken record entries that held Months instead of number of wins
* Split record column into wins column and losses column

Once loaded into pgadmin 4 additional transformations to the data was performed.

* Entries for years outside of our consideration (1979-2018) were removed from Players\_stats
* A view was then created to see the players that remained in our database.

Transformation of code can be seen in this link below:

<https://github.com/jacob-servidio/CWRU-2020-ETL-Project-2/blob/main/NBA.ipynb>

**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 original clean data sets. 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. After we queried some views to better see the database in its entirety. For example,

seeing the highest scoring players that remained in the database. Loading of data can be seen in this code below.

<https://github.com/jacob-servidio/CWRU-2020-ETL-Project-2/blob/main/NBA.ipynb>

And the link to the ER diagram can be seen in the link below .

<https://github.com/jacob-servidio/CWRU-2020-ETL-Project-2/blob/main/images/QuickDBD-NBA%20Season%20Record_Stats%20.png>

**Summary:**

When considering how to best use this database it should be noted that seasonstats does not include the name of the player. The name of the player can be found in a relational link from seasonstats to the Players table using player id.

One of the limitations for the database was that we were unable to find yearly postseason data such as playoff record and player stats. Our database is limited to regular season record and statistics. The data base does not hold information on postseason performance or NBA title holders. The data base was created to allow for a larger sample set for statistics (82 games) and to better analyze the way player statistics impact regular season win/loss record.

Another limitation was trying to link the team column in nba\_team\_win and seasonstats. At first when creating the ERD, we tried to create a many to many relationship between the tables but ran into difficulties when importing our datasets to sql. For this reason, we took away the relationship. Once the data was loaded into the sql tables, a team\_id table was create that link the two tables to better fix this issue.

Our dataset can be used to answer questions about NBA regular season record and player statistics from the 1979 to 2018 season. Once the hard work of designing and loading the database is complete, sql queries can be used to probe the database for helpful insight. For example, determine each team’s most, least efficient opponent; determine the number of games a lower ranking team won and which categories did the lower ranking team outperform a higher-ranking team; who was the conference leading three-point shooter, shot blocker, and free throw shooter. The possibilities are endless. Sample queries were provided to the client in the ‘sql\_queries’ file as a good starting point.