**ETL Project**

GROUP 7 Final Write Up: January 30, 2021

Group members: Alvin McCray, JB Kinlacheeny

**DATA SETS:**

NFL Statistics Scrape data from Kaggle: https://www.kaggle.com/kendallgillies/nflstatistics?select=Basic\_Stats.csv

And NBA Players by State data from Kaggle: https://www.kaggle.com/rishidamarla/nba-players-by-us-state

Update: 1/27: We realized we cannot merge the original datasets and only used the NBA datasets.

Update: 1/29: Narrowed down to two datasets: NBA players since 1950 and NBA 2017 twitter players. (Datasets 1 & 2 below).

New datasets:

1. NBA Players stats since 1950. <https://www.kaggle.com/drgilermo/nba-players-stats?select=Players.csv>
2. Social Power NBA. <https://www.kaggle.com/noahgift/social-power-nba?select=nba_2017_twitter_players.csv>
3. NBA 2017 Salary. <https://www.kaggle.com/noahgift/social-power-nba?select=nba_2017_salary.csv>

**EXTRACT:**

Both files will be downloaded as CSVs (Alvin: 10 min)

**CHALLENGE:** Alvin will load one of the CSVs into Postgres using PG Admin’s import tool (Alvin: 15 min)

One CSV will be imported to pandas using from\_csv, the other using from\_sql with pandas (JB: 25 min)

Update: 1/27 – We both imported all the CSVs into Pandas. We imported all 3 datasets above. We transformed the datasets by using the rename function of the Player/NAME column to “Player.” We then created Pandas DateFrames for all 3 CSVs and only used selected columns. We dropped null values using the dropna function. We tried to merge all three dfs on=”Player” in one pd.merge function and got an error. We then merged two CSVs and were able to merge on=”Player”. We then merged the last CSV to the merged df on=”Player” and was successful, however, the df only consists of 3 rows.

Update: 1/29 – Removed the 3rd data set and undid the merge in pandas to allow the loading to happen to postgres.

Graphical user interface

Description automatically generated with medium confidence

**TRANSFORM**

Once imported, we will join the tables on State/Birthplace Player\_ID, state\_id, city\_id (JB: 15 min)

Update: 1/27 – we eventually joined the tables on Players.

Rename all columns per instructions (see load) (JB:10 min) 1/27 – Done in pandas.

**OPTIONAL:** We will look to clean the joined data and select data only from DC if time permits. – the original dataset was removed.

Table

Description automatically generated

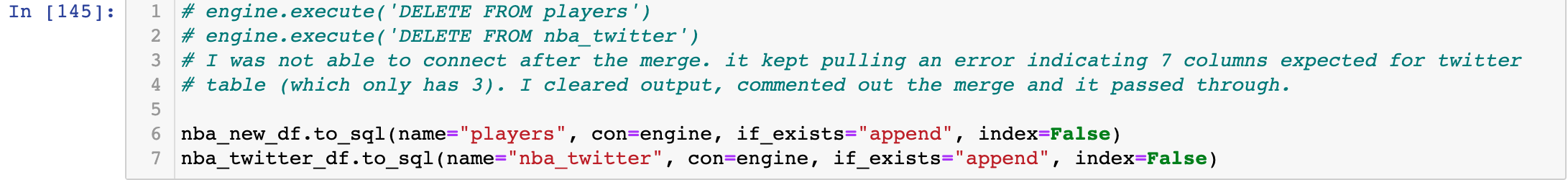
**LOAD**

While JB is extracting/loading, Alvin will read through each dataset and create a Postgres table to load into. (Alvin/JB: 1 hr)

JB will work with Alvin to rename columns appropriately (JB: 15 min)

Alvin will write the portion of the python script that loads the data into the Postgres database. (20 min)

Update: 1/29 – JB used pandas to connect to postgres and connection had failed a few times. I had to use the DROP TABLE and DELETE TABLE in pgadmin and pandas.



Update: 1/29 – JB matched the variables and created two tables in SQL called ETL-Project.sql and connected the engine. Used to\_sql function to load dataframes into SQL. I was not successful at first because I had the datasets already merged in pandas. I undid the merge and loaded the datasets successfully. After all was successful, I pushed the work into Alvin’s repo.

Text

Description automatically generated

**CONCLUSION:**

The ETL Project was a new assignment that both group members have never undertook. Finding data was not as problematic as the first group project. We were able to find data on Kaggle.com and chose to use NBA and NFL player data. Our initial intent was to generate these datasets into one dataset to identify where professional players come from (hometown city and state). After importing the datasets into pandas, we encountered issues with the data (i.e., no common values between the two). The final data used is two datasets described above.

For the transformation portion, we imported the datasets into pandas and removed columns not necessary for the final two tables. We also renamed a few column names and transformed it to 2nd normal form. After cleaning the tables, we connected the engine to the postgres database server using python. Before loading the data, we generated the SQL code and created two tables (players & nba\_twitter) in pgAdmin. After the tables were created in pgAdmin, we connected the engine and loaded the pandas dataframe into pgAdmin using the to.sql function.

Once loaded, we were able to join the tables in SQL.

Graphical user interface, text

Description automatically generated

Graphical user interface, application

Description automatically generated

Finally, once we understood the ETL process in more depth and taking into consideration key elements such as project management, deadlines, and available resources, we were able to complete the ETL in a timely manner. From the initial proposal to the final output, there were major alterations with the main being our datasets. A common source or key variable needs to be represented in both data sets. For our project, it was “players.”