ETL Project

Emmanuel Ayoo

Katie Buckner

Michelle Goodman

Polina Holland

The ETL project was intended to help us learn to extract, transform, and load our data into a usable format. As a group, we also discovered that it forced us to work more cohesively and narrow down what we wanted to do with the data, instead of simply taking a large chunk of data that we did not need.

**Extract:** To begin, we first had to decide what data we were going to be using and extract that data from their sources. As a group, we had decided to use NFL data, specifically from NFLsavant.com, several weeks ago. But when we actually dug into the data, we found that it did not really have what we needed, which forced us to branch out.

We found several sources with NFL data, and decided to use two specific ones:

* <https://www.kaggle.com/toddsteussie/nfl-play-statistics-dataset-2004-to-present> and its partner Kaggle, <https://www.kaggle.com/toddsteussie/nfl-play-statistics-secondary-datasets>
* <https://www.pro-football-reference.com/players/salary.htm>

We started out by trying to combine those CSV’s from Kaggle that we thought we might want to eventually use. The biggest problem we ran into was that there was so much data. Without knowing what we wanted to do with the data and without a final objective, we found that there was more data than our jupyter notebooks could handle. This forced us to narrow our search. This back and forth took much more time than we would have liked, and we finally had to just make a decision to go with two of the CSV’s from Kaggle for this project, and then pull in more tables for project two if we decide to use this information for that project. Polina created a Github repository for us to use and started a Jupyter Notebook in which she uploaded the csv’s we had pulled as dataframes.

We went with the draft.csv from Kaggle, the passer.csv from Kaggle, and four separate salary csv’s which Katie cleaned up and submitted.

**Transform:** After loading the csv’s the columns that we wanted to use were extracted from the original csv’s, and we were ready to transform the data to a form that we wanted.

We started by pulling out the columns that we wanted from the draft and passer csv and putting them into new\_data\_df1 and new\_data\_df2. This was followed by splitting the Player column of the four salary csv’s into two parts: a player name column and a player id column, and then dropped the columns we did not need. We opted to keep these four csv’s separate for now until we know for sure what we wanted to do with them.

This was followed by renaming some of the columns. We worked together to do this. It was determined that we liked the names for the first two dataframes and that we would leave them as they were. For the salary dataframes, we dropped the Player and Player ID columns, and renamed the other columns to match what we already had in our first two dataframes.

We were now ready to begin the process of loading the data.

**Load:** We began this by using PgAdmin to create our relational database and schema. We created six tables:

* passer
* draft
* salary\_2016
* salary\_2017
* salary\_2018
* salary\_2019

We chose to make the playerId the primary key for our passer table and the playerName the primary key for our draft table. All of the salary tables have the player\_name as their primary key. After creating all of the variables in our schema, we were ready to go back to our Jupyter Notebook to check for the databases and load the csv’s into the databases. We actually ended up deciding to use SQLite. After loading all of the dataframes into SQLite we queried each one to confirm that the data had loaded correctly.

This was definitely a learning experience for all of us. It helped us learn to work better as a team, and helped us get better at problem-solving together.