ETL Project Write-up

Victor G., Ruddy S., Patrick H.
10/13/2019

Summary:

For our ETL Project our goal was to identify salary trends across various professional sports. We decided to compare salaries within the sports themselves and against each other. The sports that we decided to go with are NBA, NFL, MLB, and MLS. We collected historical data on each sport to analyze a 10 year trend in the sports salary progression and then we used this dataset to analyze the individual sports YoY growth and how each sports compares to identify the highest paying professional sport.

Technical Questions:

1. The sources of data that you will extract from:
    a. NBA
        i. https://www.kaggle.com/hrfang1995/nba-salaries-by-players-of-season-2000-to-2019
            1. Salary information. (Does not include team).
        ii. https://query.data.world/s/3wm7o467ixh47e7db3lme63bdql7ly
            1. Team + player details
    b. MLB
        i. https://www.usatoday.com/sports/mlb/salaries/2011/player/all/
            1. Detailed salary data for each year. 10 years of data collected.
    c. NFL
        i. https://www.spotrac.com/nfl/salaries/breakdown/2011/
            1. Detailed salary data for each year. 10 years of data collected.
    d. MLS
        i. https://data.world/dataremixed/mls-player-salaries-2010-2018

2. The type of transformation needed for this data (cleaning, joining, filtering, aggregating, etc).
    a. Our raw data came from CSV's generated by the above data sources. Once we have compiled the necessary CSV's our process was as follows:
        i. Import the CSV's into jupyter notebook
        ii. Remove NA's and duplicates from the data using pandas.
        iii. Using Pandas, join player data to salary data if necessary.
        iv. Using Pandas, Extract the year, team, salary, player, and position fields from each dataset.
        v. Normalize the data to give each data source the same column names.
        vi. With the SQL Alchemy module, Import the dataset into its respective professional sport table housed in Postgres SQL.
    b. Tasks that were done prior to imported the data into Postgres SQL:
        i. Create the sports_DB database in Postgres.

        ii. Create each table in Postgres with the year, team, and position set as the primary key.
    c. This process can be duplicated as many times as needed in order to continue adding future or past years to the datasets.

3. The type of final production database to load the data into (relational or non-relational).
    a. We decided to go with a relational Postgres database to house the player salary information. This is to ensure that we have data integrity. We decided to go with a primary key of year + player + position to account for players with the same name and for players that switch positions in the same year.

4. The final tables or collections that will be used in the production database.
    a. We created a database called sports_db and created 4 main tables to use in the database:
        i. NBA_player_salary
        ii. MLB_player_salary
        iii. NFL_player_salary
        iv. MLS_player_salary
    b. After the main tables were added to the database, we created a series of views to aggregate the data:
        i. AVG_sport_salary
            1. Compares average sport salary YoY for each professional sport grouped by year and sport all housed in one view.
        ii. AVG_sport_salary_by_team
            1. Compares average sport salary YoY for each professional sport grouped by year and sport all housed in one view.
        iii. AVG_sport_salary_by_position
            1. Compares average sport salary YoY for each professional sport grouped by team, year, and sport housed in one view.
        iv. AVG_sport_salary_by_position_and_team
            1. Compares average sport salary YoY for each professional sport grouped by team, year, sport, and position all housed in one view.
    c. Having these views allows us to easily create charts using Matplotlib in order to tell a story with the data.