**E**xtract **T**ransform **L**oad

**Technical Final Report**

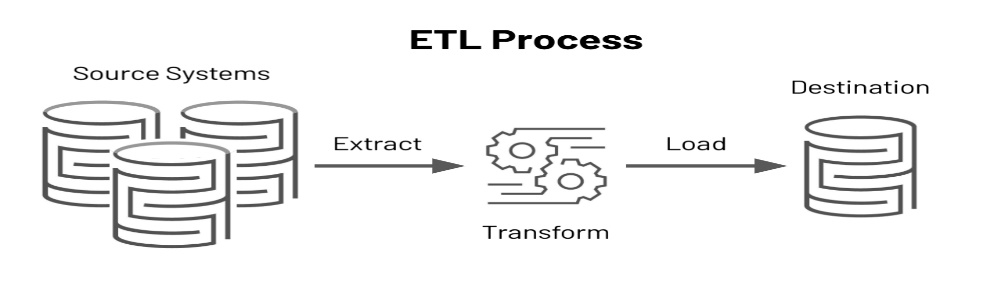
*Team Members:*

*Rachel Newberry, Madhusudan Tamrakar, and Donia Tarjam.*

​**O**verview – This ETL technical report takes the user through the stages of extraction, transformation, and loading the data. The project incorporates Olympic statistics from two different data sources for data clean up and analysis. The report will supply and include steps taken to complete the ETL process.

​

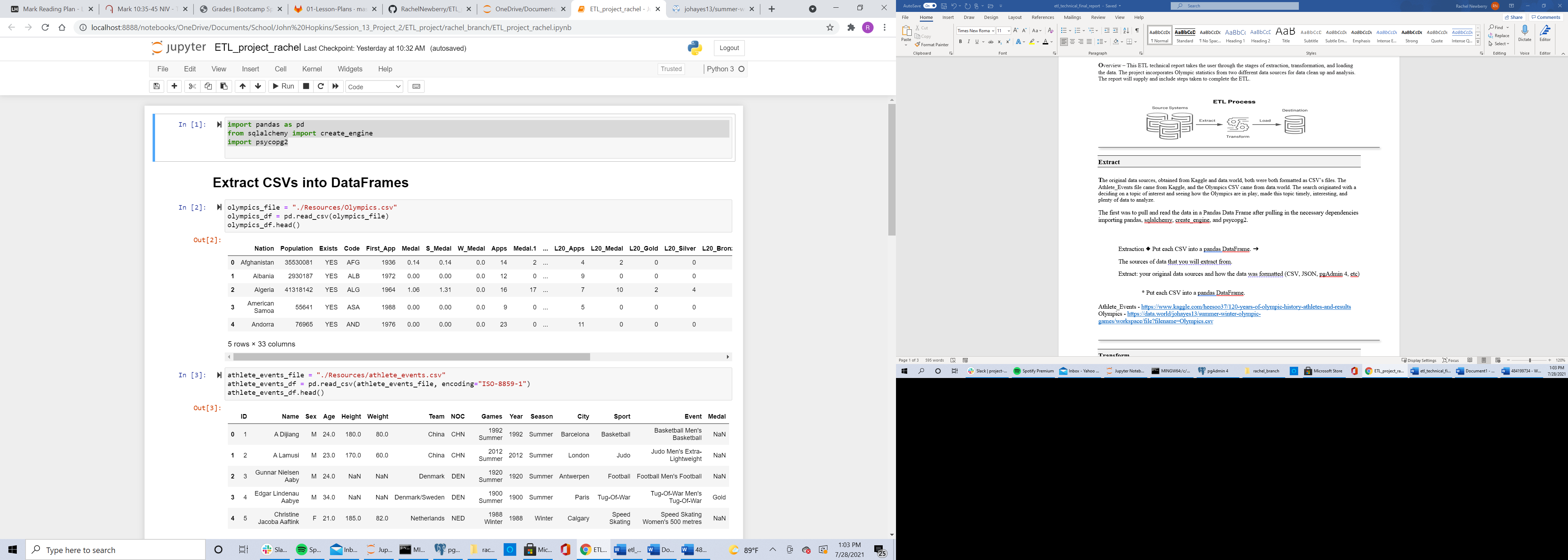
​



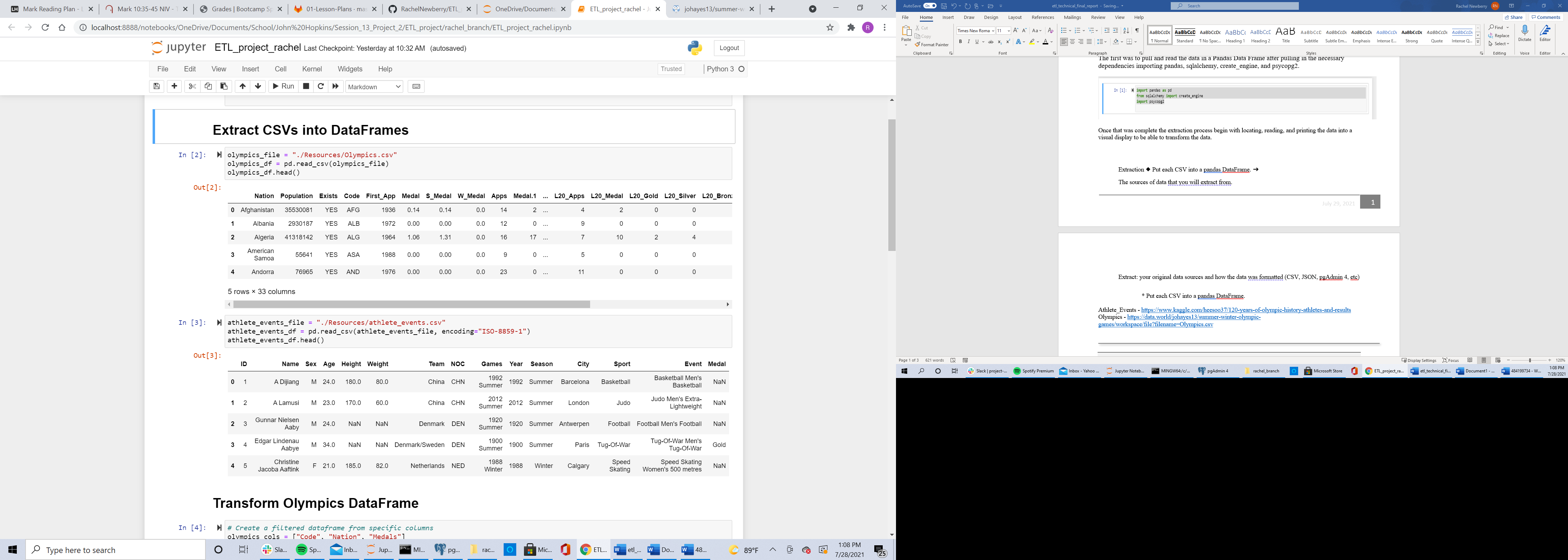
Extract

**T**he original data sources, obtained from Kaggle and data.world, were both originally formatted as CSV’s files. The Athlete\_Events file came from Kaggle, and the Olympics CSV file came from data.world. The search originated with deciding on a topic of interest and seeing how the Olympics are in play, made this topic timely, interesting, and plenty of data to analyze.

The first step was to pull and read the data in a Pandas Data Frame after pulling in the necessary dependencies importing pandas, sqlalchemy, create\_engine, and psycopg2.



Once that was complete the extraction process could begin with locating, reading, and printing the data into a visual display to be able to transform the data for each data set.



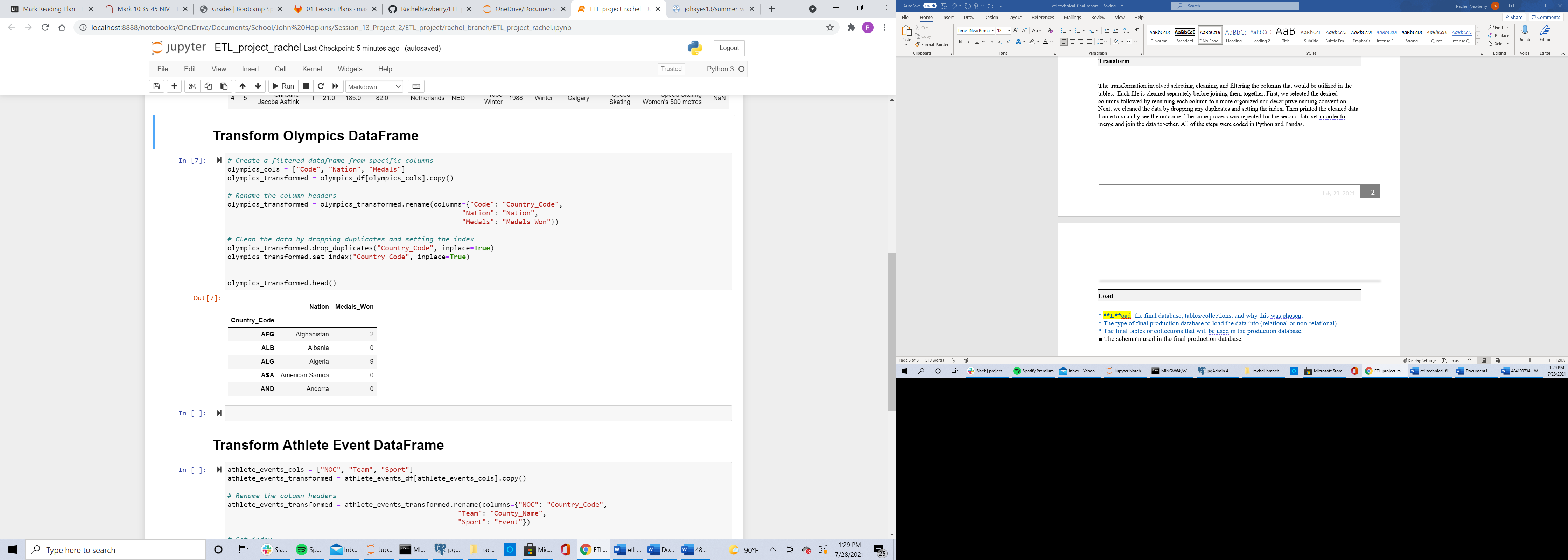
Sources of CSV Data Sets

***Olympics*** *-* [*https://data.world/johayes13/summer-winter-olympic-games/workspace/file?filename=Olympics.csv*](https://data.world/johayes13/summer-winter-olympic-games/workspace/file?filename=Olympics.csv)

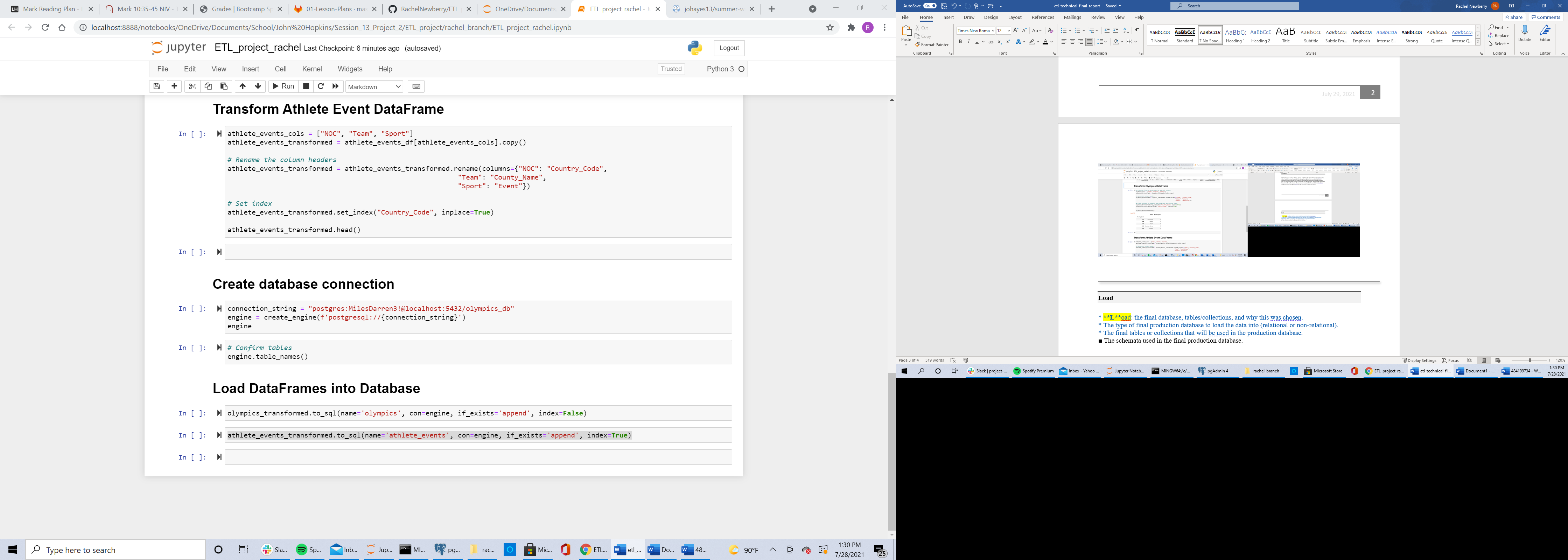
***Athlete\_Events*** *-* [*https://www.kaggle.com/heesoo37/120-years-of-olympic-history-athletes-and-results*](https://www.kaggle.com/heesoo37/120-years-of-olympic-history-athletes-and-results)

Transform

**T**he transformation portion involved selecting, cleaning, and filtering the columns that would be utilized in the tables. Each file is cleaned separately before joining them together. First, we selected the desired columns followed by renaming each column to a more organized and descriptive naming convention. Next, we cleaned the data by dropping any duplicates and setting the index. Then printed the cleaned data frame to visually see the outcome.

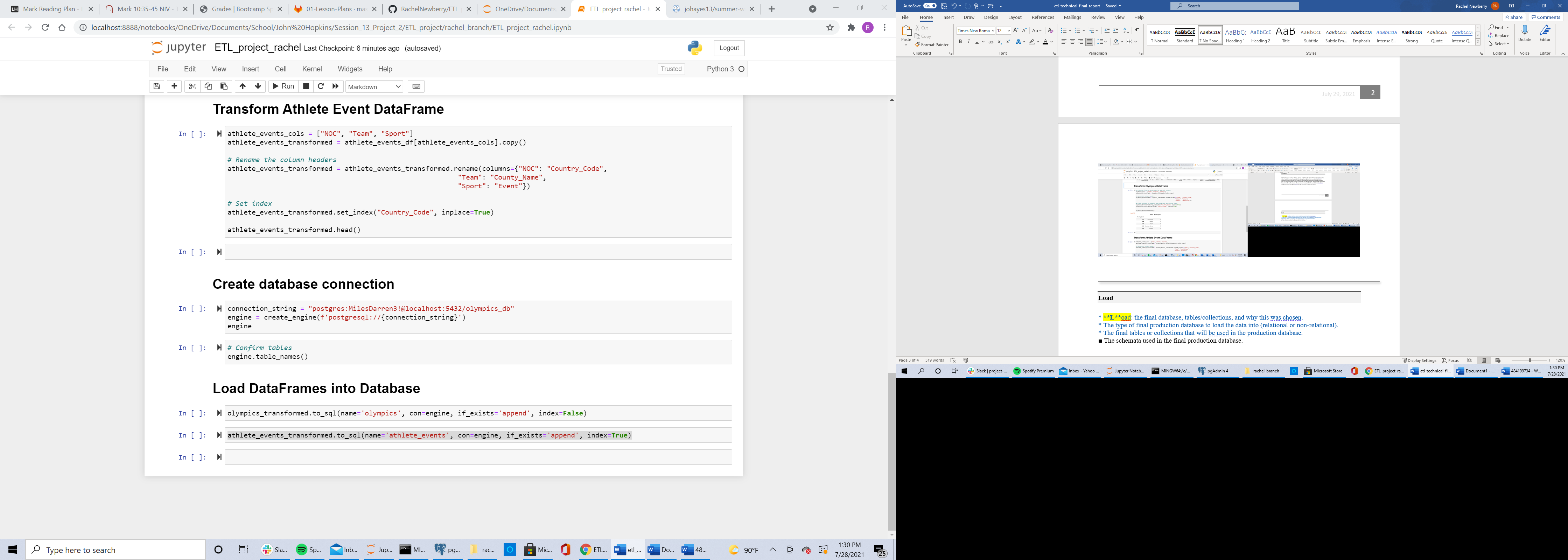


The same process was repeated for the second data set in order to merge and join the data together. All of the steps were coded in Python and Pandas…



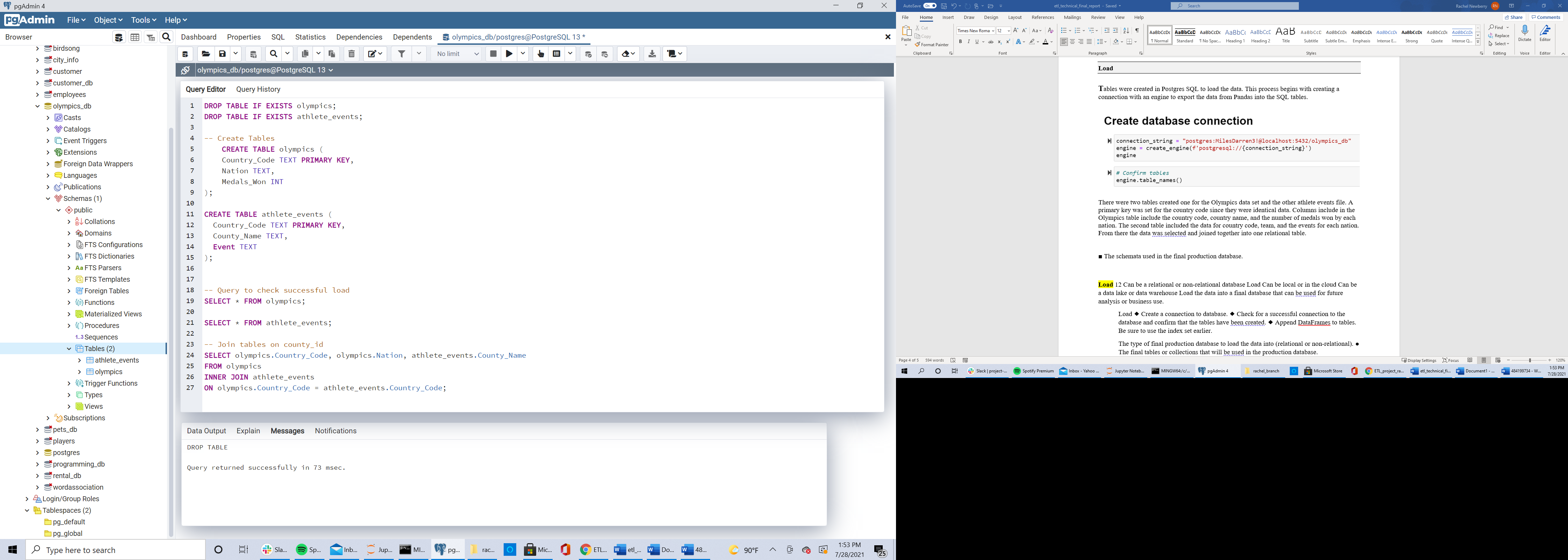
Load

**T**ables were created in Postgres SQL to load the data. This process begins with creating a connection with an engine to export the data from Pandas into the SQL tables.

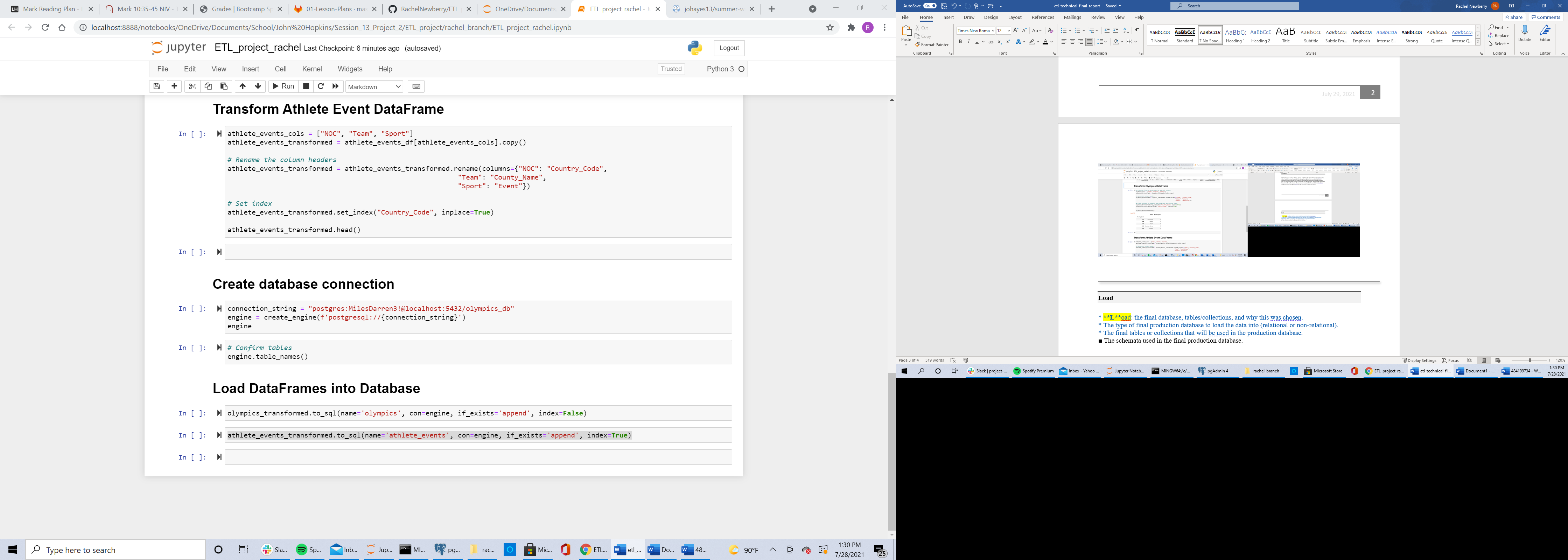


There were two tables created one for the Olympics data set and the other athlete events file. A primary key was set for the country code since they were identical data. Columns include in the Olympics table include the country code, country name, and the number of medals won by each nation. The second table included the data for country code, team, and the events for each nation. From there the data was selected and joined together into one relational table.

The schemata file used in the final production database.



The data was finally loaded and appended through the following code.



Confirm data has been added by querying the Olympics and athlete\_events table and joining the tables on country\_id column through the following code:

-- Query to check successful load

SELECT \* FROM olympics;

SELECT \* FROM athlete\_events;

-- Join tables on country\_id

SELECT olympics.Country\_Code, olympics.Nation, athlete\_events.County\_Name

FROM olympics

INNER JOIN athlete\_events

ON olympics.Country\_Code = athlete\_events.Country\_Code;

**Graphical user interface, table

Description automatically generated**