Extraction, Transformation, and Load Technical Report

Headlines and Stock Price Trends

**TABLE OF CONTENTS**

1. [Introduction 3](#_heading=h.2xcytpi)
   1. [Summary 3](#_heading=h.30j0zll)
   2. [Scope 3](#_heading=h.3znysh7)
   3. [Technologies and resource contributions 3](#_heading=h.tyjcwt)
   4. [Definitions, Acronyms and Abbreviations](#_heading=h.1t3h5sf) 4
2. [ETL Details 4](#_heading=h.1ci93xb)
   1. [Data Import/Extract Sources and Method 4](#_heading=h.2s8eyo1)
   2. [Data Acquisition 4](#_heading=h.3rdcrjn)
   3. [Data Transform 4](#_heading=h.lnxbz9)
   4. [Data Integrity 4](#_heading=h.1ksv4uv)
   5. [Data Refresh Frequency 4](#_heading=h.2jxsxqh)
   6. [Data Security 4](#_heading=h.3j2qqm3)
   7. [Data Loading and Availability 5](#_heading=h.4i7ojhp)
3. [Data Quality 6](#_heading=h.3whwml4)



*The purpose of the Extraction, Transformation, and Load (ETL) Technical Report is to capture details that pertain specifically to ETL portion of the data pipeline that is to be used in a data science project. This however does keep in mind the final target objective while performing the ETL.*

# Summary

The objective of this project is to track the relationship between news headlines and fluctuations in stock prices. Looking at these two data sets allows us to further analyze the correlation between the success/decline of stock and how the company is reflected in the media.

# Scope

The integrated data sources are comprised of the Alpha Advantage API and the News API. Alpha Advantage provides real-time and historical stock data. News API provides a means to search worldwide news articles and headlines from all over the web in real-time. Determining positive or negative words that influence stock price is outside the scope of this project.

# Technologies and resource contributions

**Janita Brock** - Assisted in API interface for obtaining data, cleaning data, and providing information for the technical report.

**Ola Browne** - Created visualizations for report.

**Parul Garg** - Created Postgres server and exposed client connection for team to access common DB server. Successfully established a connection off campus from home network where client and server connection worked. But could not troubleshoot beyond GT network due to security issues. DB setup and obtained view queries for various analysis.

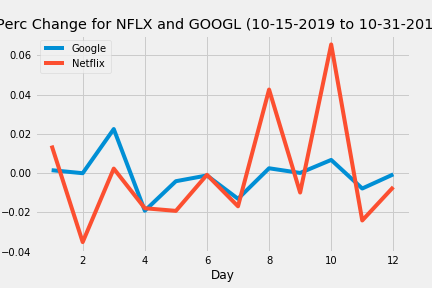
**Reza Gharahgozly** - Created an ER diagram for SQL databases and used Postgres SQL to make tables in the database.

**Sandy Lake** - Worked with Robert on obtaining headlines from News API and converted headlines to CSV

**Robert Patterson** - Worked with Sandy on obtaining headlines from News API. Munged data in pandas to be consistent with how it would be loaded into the Postgresql database. Helped to construct Second Normal Form data structure with Parul and Reza.

**Selvi Ramalingam** - Retrieved stock API for 5 companies of choice, and set a template for further retrieval if there is more interest.

We did a “Percentage change Analysis” for 2 companies over 2weeks of time. Here, we used Python to connect to Posgres server, obtained data in dataframe and used plot library.



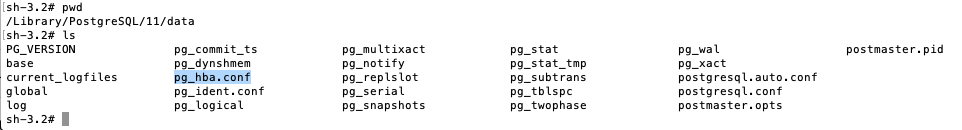
**Challenges encountered:**

While setting up the Postgres Server, we tried to set up one machine as Postgres Server which can be accessed by other systems in the same network. Considering security settings of the GTVisitor network - this kind of setup was taking more effort and time than the time allotted.

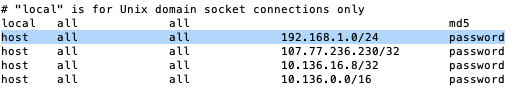
With the similar setting change, this kind of setup was achieved in a home intranet setup.

Setting change:

> For the system setup as a server, login to root as ‘su’ and make changes to the ‘pg\_hba.conf’ file



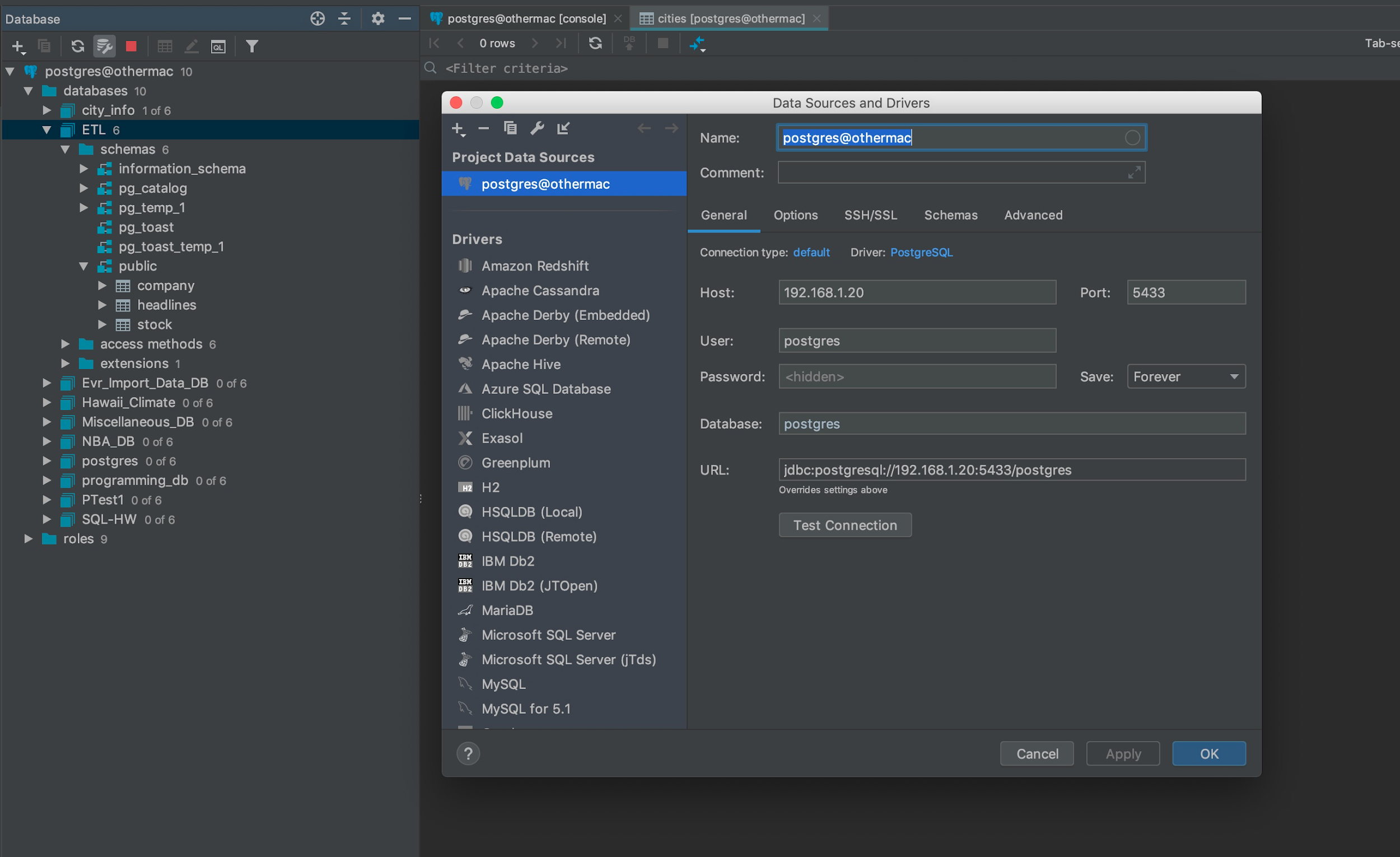
> Add a new line for the IP setting for the network:



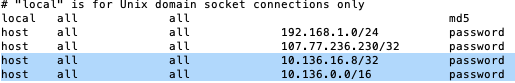
The highlighted setting above is for the home network where ‘client’ DB was able to connect to Server system. (Screenshot of client shown below)

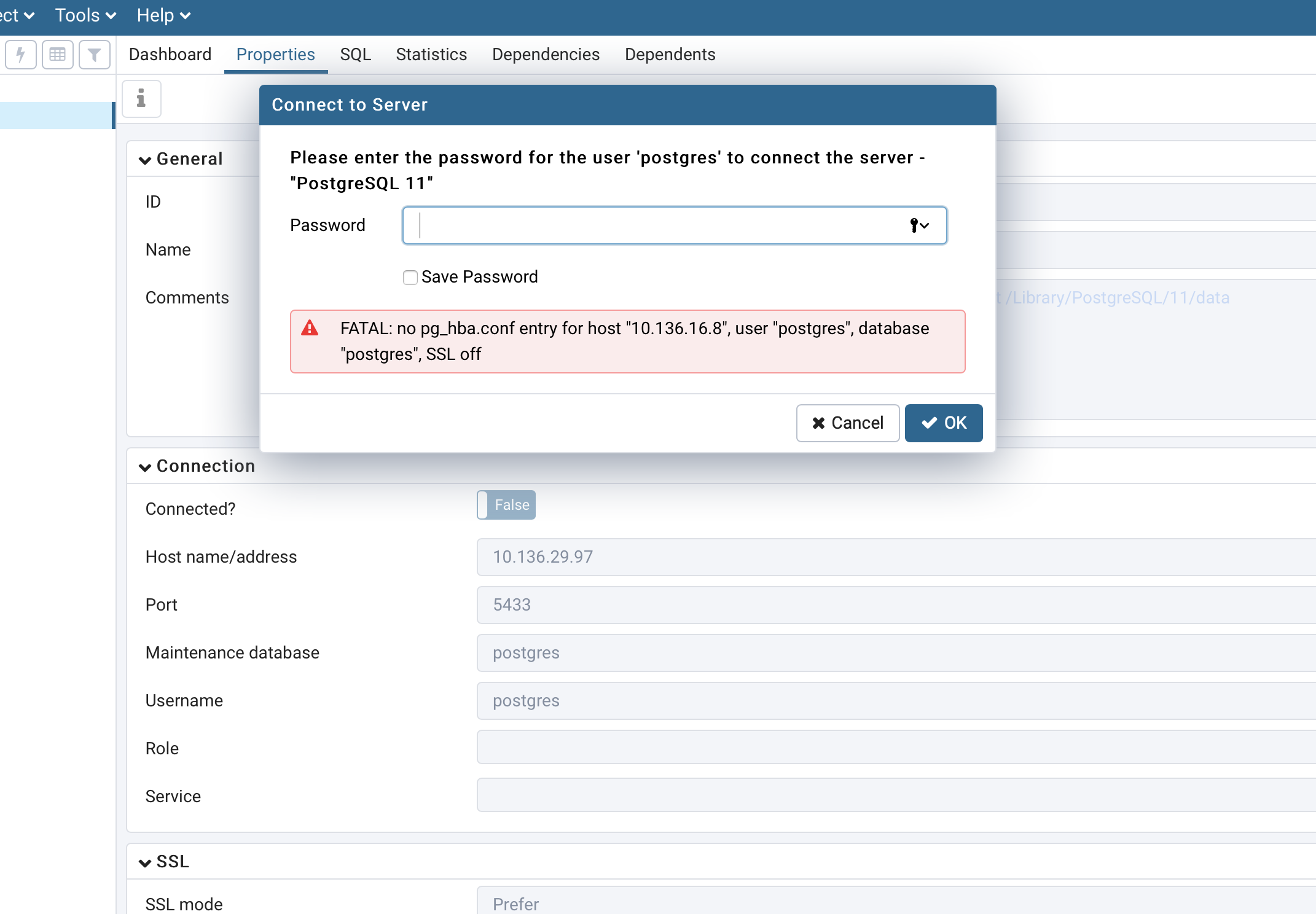
# Definitions, Acronyms, and Abbreviations

API - Application Program Interface



> Similar settings on GT network did not allow the client to access server DB. (SSL error)





This was a great learning and challenging experience.



# Data Import/Extract Sources and Method

|  |  |  |
| --- | --- | --- |
|  | Source - APIs | Urls |
| 1 | Alpha Vantage | [https://www.alphavantage.co/documentation](https://www.alphavantage.co/documentation/) |
| 2 | News API | <https://newsapi.org/> |

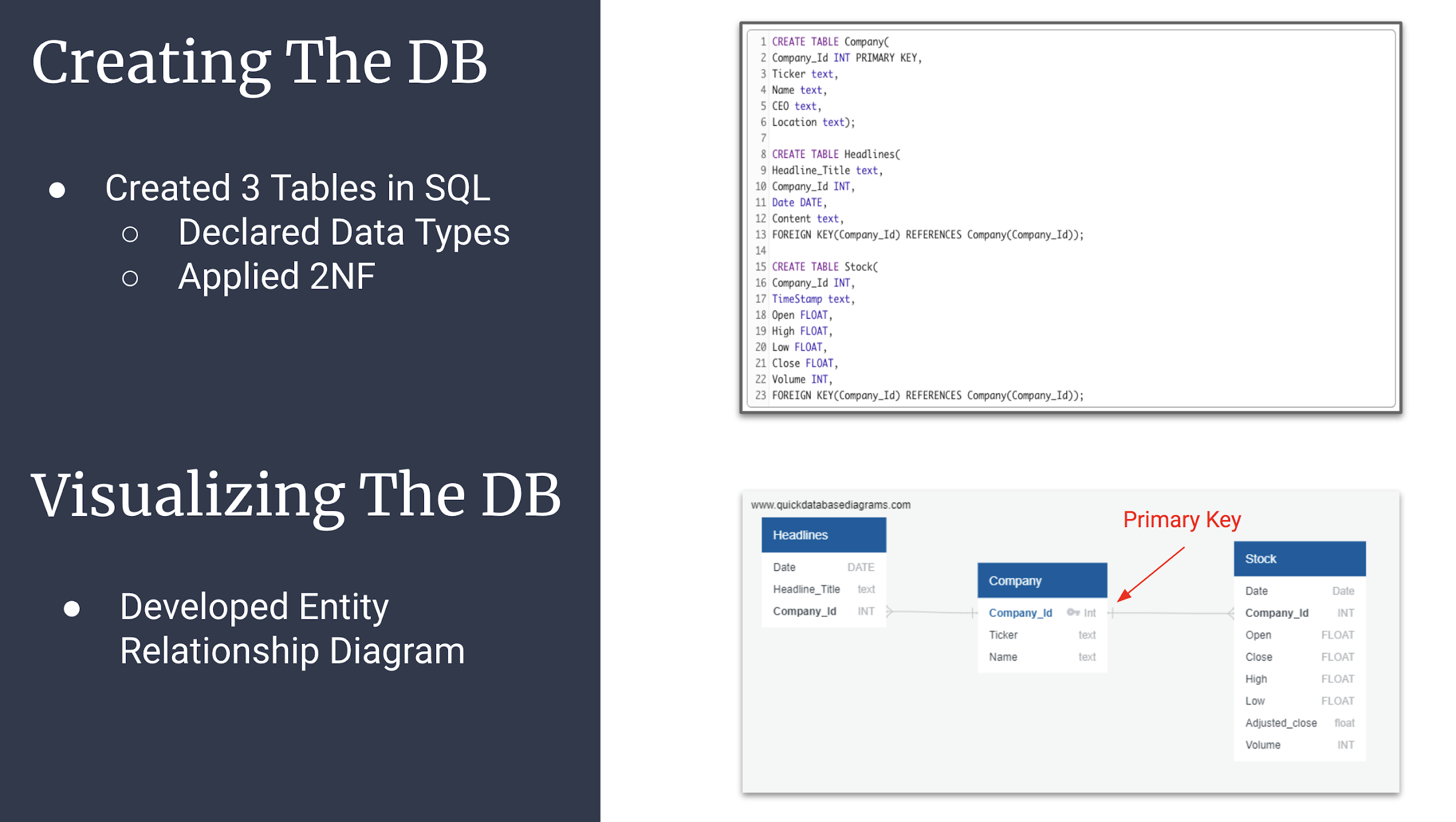
Method included in 2.2 Data Acquisition

# Data Acquisition

We used two APIs to help gather data for this project.

The first used was [Alpha Vantage (https://www.alphavantage.co/documentation](https://www.alphavantage.co/documentation/)) which is a free API that provides realtime and historical data on stocks. We are able to choose the parameters of the data as described in their documentation which led us to use the ‘Time Series Daily Adjusted’ API so we were able to receive daily information on the opening and closing prices, the low and high for the day, along with volume. This allowed us to truly send the trends of the stocks during the date range we selected. There were additional options to see the API data in a full or compact output size. The full data would return all current data along with 20+ years of historical data, but the compact would provide the last 100 data points on all stocks collected. As we knew we would be working with two API datasets we chose to utilize the compact output.

Additionally, newsapi.org was used to collect headlines related to the companies we previously gathered stock data for. This news API is also free and the API key allows for 60 requests per minute. Creating the request required a call to newsapi.org with appended search parameters (i.e. to\_date, from\_date, keyword, etc.) A difficulty we encountered was that the API provided a max return of 20 hits/page, and in order to obtain all hits, we created a for loop which iterates through all of the return pages. This data was munged in a pandas data frame to have clean and concise columns. This data frame was exported straight to a CSV file to be uploaded into the SQL database.



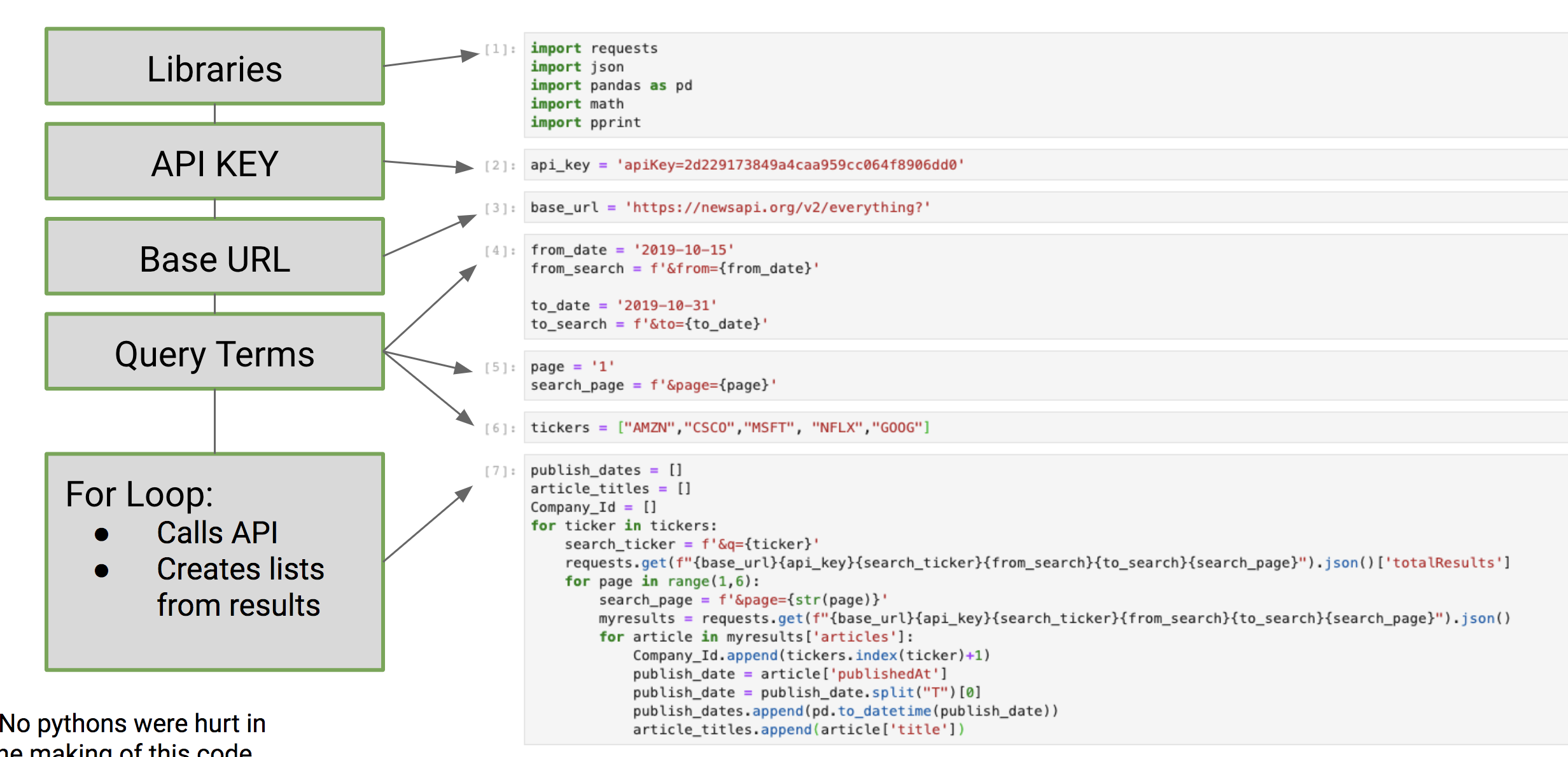
# Data Transform

The data was transformed to allow for Second Normal Form. The primary key company\_id was used to normalize data across three different tables. The first table contained information about various companies and is where the primary key is assigned. Each company has a unique company\_id. Stock and headline data was transformed to only present information for a two-week period. This was done primarily to reduce gaps in information and to quickly confirm gaps and discrepancies between data tables. In the stock data table, price information and the corresponding date is itemized across rows for each day in the two week period. In headline data table, headlines and dates are itemized across rows for each day in the two week period.

# Data Integrity

The data from both resources had high data integrity as they both provided full data sets on a daily basis. The one gap we did have between APIs was the news API provided information on each day of the week (Sunday - Saturday), while the Alpha Vantage API provided information during the week (Monday - Friday) when the stock exchange is open.

Both APIs are updated daily and the code is written to allow the user to place in desired data points, i.e. reference dates and companies.



# Data Refresh Frequency

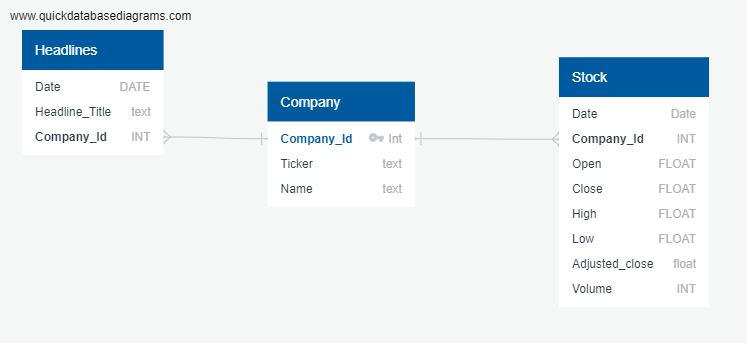
Due to the data automatically refreshing on a daily basis, we have built our code to allow us to input the desired dates and companies.

# Data Security

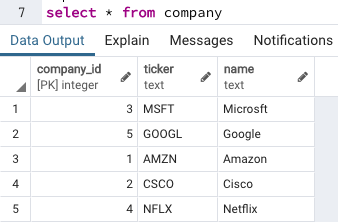
No data anonymity and security requirements needed to be satisfied in this ETL project.

# 2.6 Data Loading and Availability

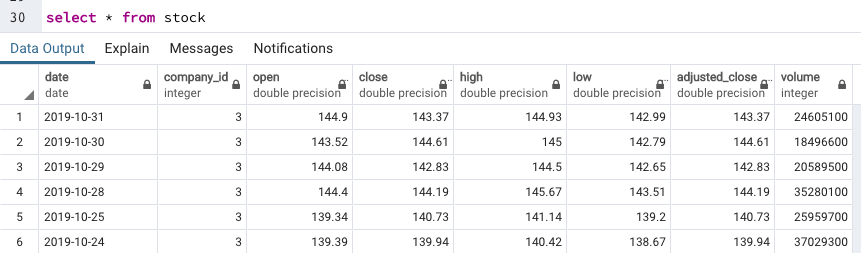
ERD for this project includes three tables - “company” (master table), “headlines” and “stock”. The database for this ETL project has been created in Postgres server.



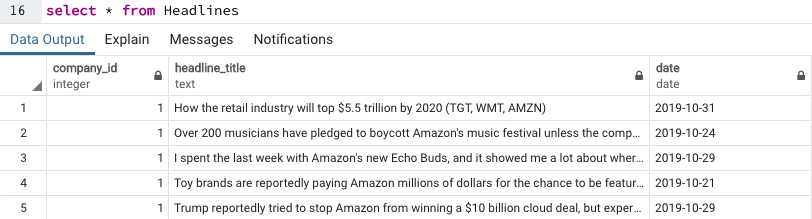
Company table targets to store the companies which are being considered to be queried and compared during the data search. Currently this database tries to capture stocks and headline data for Amazon, Cisco, Microsoft, netflix and Google.



“Stock” data is a set of open, close, high low values of each company for a particular date range. This is retrieved using respective exposed API in the form of CSV files which were later imported in the database.



“Headlines” data is data set retrieved using news API for respective dates and requested companies. This is also saved as CSV files and imported into DB





Address in this section success criteria for this project. Summarize the parameter KPIs such as Totals and expected counts. What user acceptance testing was performed and what were the outcomes. What is the recommended site acceptance testing that your client can perform to ensure the expected outcomes meets their expectations?

If time permitted, this System would expose an API which will give user the power to make a search for a company (from Company table) stocks data and news headlines regarding the same company in a given date range. The expected result would look like this but response structure would be JSON:

