|  |  |
| --- | --- |
| Jeffrey Mychalchyk | Mike Toriello |
| Paul Aggarwal | Dean Daley |

**Purpose**

The purpose of this database is to provide all relevant information regarding fortune 500 companies. This ranges from fortune 500 ranking, to why they are ranked, to employee reviews, to stock prices, and much more. An end user for this database should be able to compile some fascinating correlations between the many different aspects of a fortune 500 company.

The ETL process was used to create a database in MySQL containing 6 datasets. Each dataset has a unique identifier, and a variable to relate to at least one other table in the database via joins or sub-querys. Below is a closer look at the implementation of the ETL process in more detail.

**Extract**

* Download Fortune 500 csv via Databahn
* API to pull stock quotes from www.alphavantage.co
* Download countries.csv from wiki
* API to pull exchange rates from www.alphavantage.co
* Web Scraping jobs posted by Fortune 500 companies from Indeed.com using ‘python, sql, etl’ and ‘company’ as filter inside the url query. Used combination of Beautiful Soup and simple Python text matching methods to target key data points from Indeed web page. Extracted from one to ten jobs for each company on the Fortune 500 list of year 2018.
* Downloaded population csv from http://worldpopulationreview.com/us-cities
* Indeed.com company reviews, web scrape using Beautiful Soup. The retrieval URL is https://indeed.com/[company-name]/reviews. Where ‘[company-name]’ is changed for each retrieval.

To Reproduce the extract - The code simply needs the Indeed.com company name, which can be extracted by retrieving a list of names from the ‘indeed’ table in the database. The company name is stored in the ‘company\_name’ variable which is used to compose the URL. The output file will be directed to the sub-folder ‘./data’ with the name of the company appended to ‘\_reviews.csv’ as the filename. If a company name has spaces, the spaces receive hyphens in the URL for manual entries. Examples of company names: Apple, Verizon, Bank-of-America, Walmart. Company names.

**Transform**

* Loop through and pull data per date and use try/except to bypass missing codes.
* Create Output.csv
* Loop through and pull data and use try/except to bypass missing currency types.
* Create Exchange.csv
* Read Fortune 500 list csv and loop thru each company name and check/replace special characters (ampersands, spaces, dashes, commas etc.) in the company name using urllib.quote feature and apply to url query. Example: if ‘AT&T’ then co\_quote= ‘AT%26T’ which will be applied to the first url query "https://www.indeed.com/jobs?q="+ co\_quote +"&sort=date"
* Indeed jobs page was web scraped and loaded into a list of dictionaries and converted to dataframe then exported to CSV file ‘indeed.dat’ so it can be finally imported into SQL database tables.
* The company name from the job posting is needed to form the URL to retrieve the company review web page. The interface is a multi-page result set. There is a rudimentary retrieval of the number of pages, then proceeding through each page retrieving the stripped data elements. It is stored in a DataFrame then exported to a CSV bearing the company name in the ‘./data’ folder. The company name is stored with reviews dataset to act as a link back to the company table.

**Load**

The procedure for loading all the datasets into MySQL using Python is as follows:

* Create a connection in python file to MySQL using SQLAlchemy
* Read the dataset into a Pandas DataFrame
* Clean the dataset (Adjust column headers to make SQL friendly names. Drop unnecessary columns)
* Commit table as a new table in the MySQL database.

The final product is a database with 6 unique tables. Each table has a unique identifier, and all tables can be referenced in another table by a common identifier. Below are examples of how this database can be used:

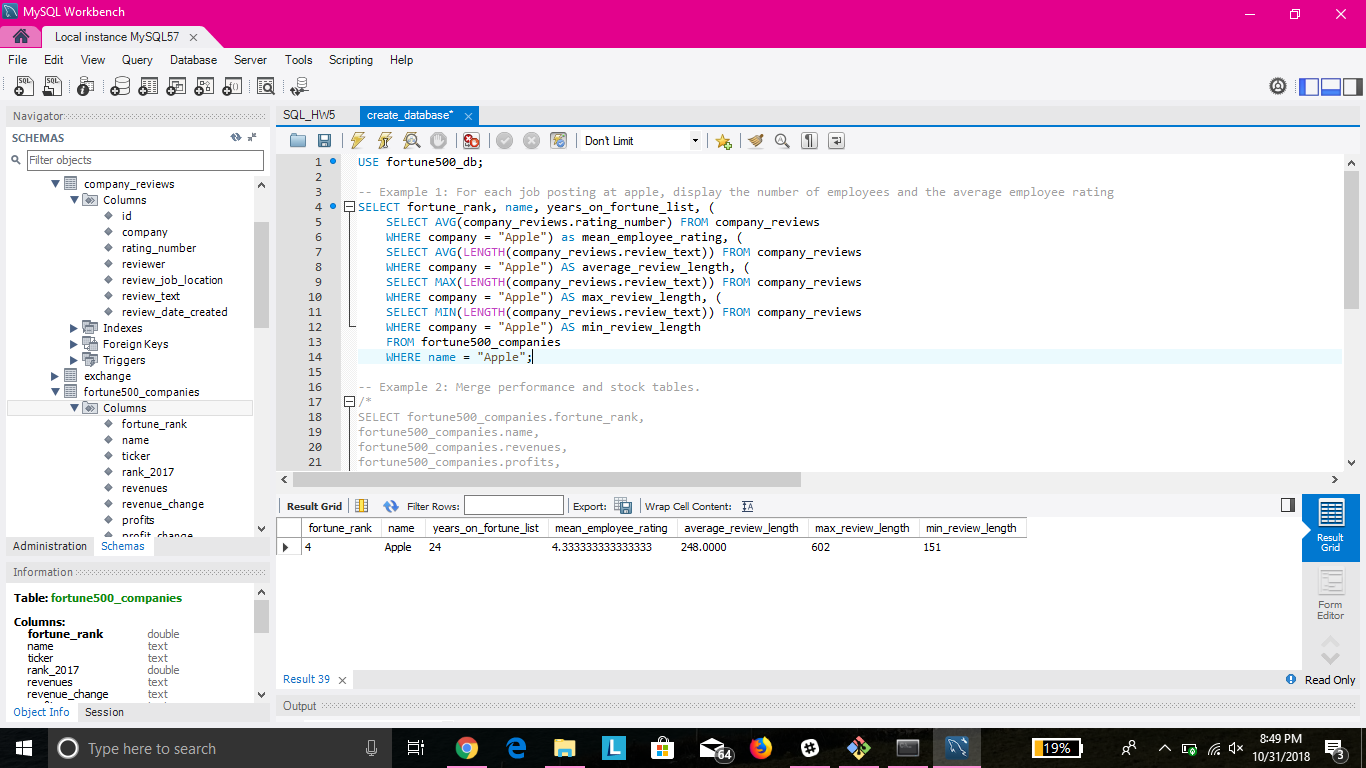


Figure 1: Job Reviews Per Company

* Figure 1 is an example of using this database to extract performances of a company and compare them to employee reviews and review status. On a larger scale, this could be done for all fortune 500 companies.

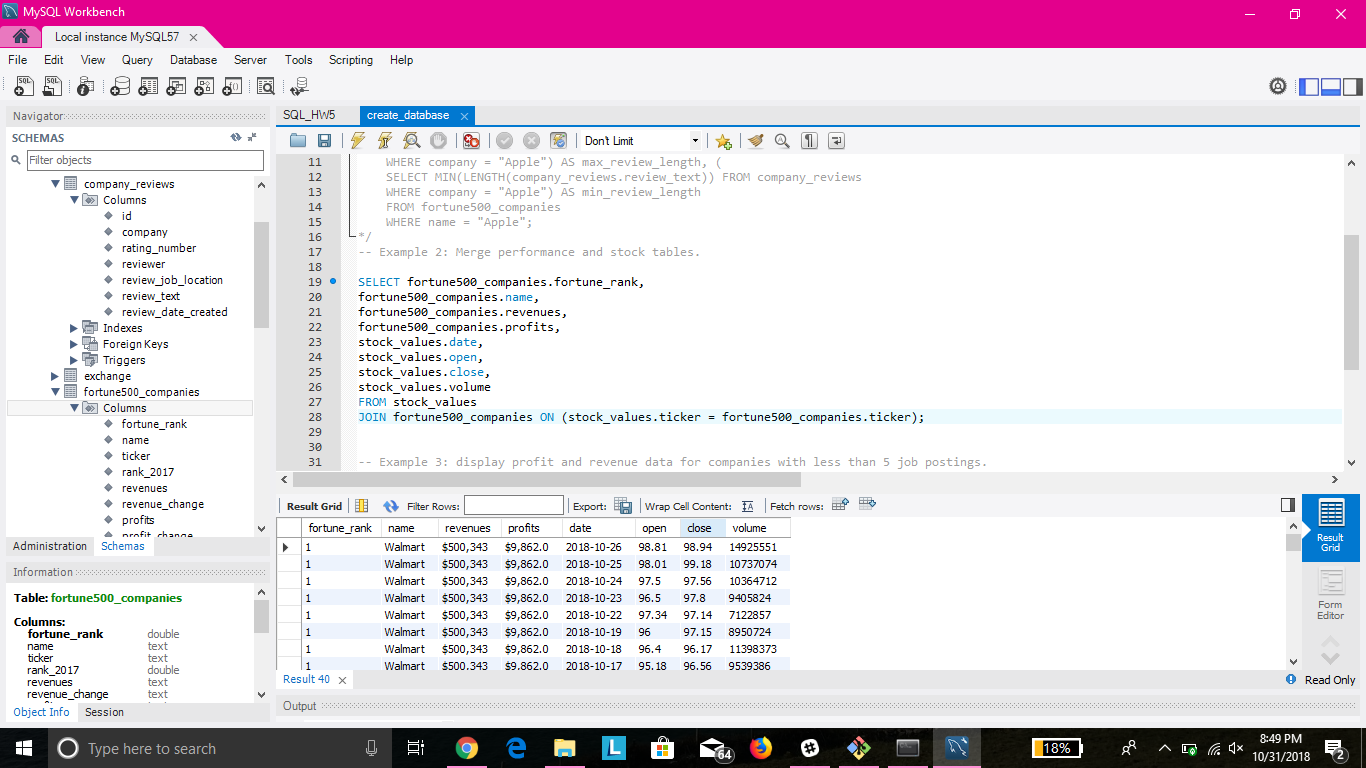


Figure 2: Revenues vs. stocks

* Figure 2 uses the database to extract stock performances of a company and compare them to revenues and profits.

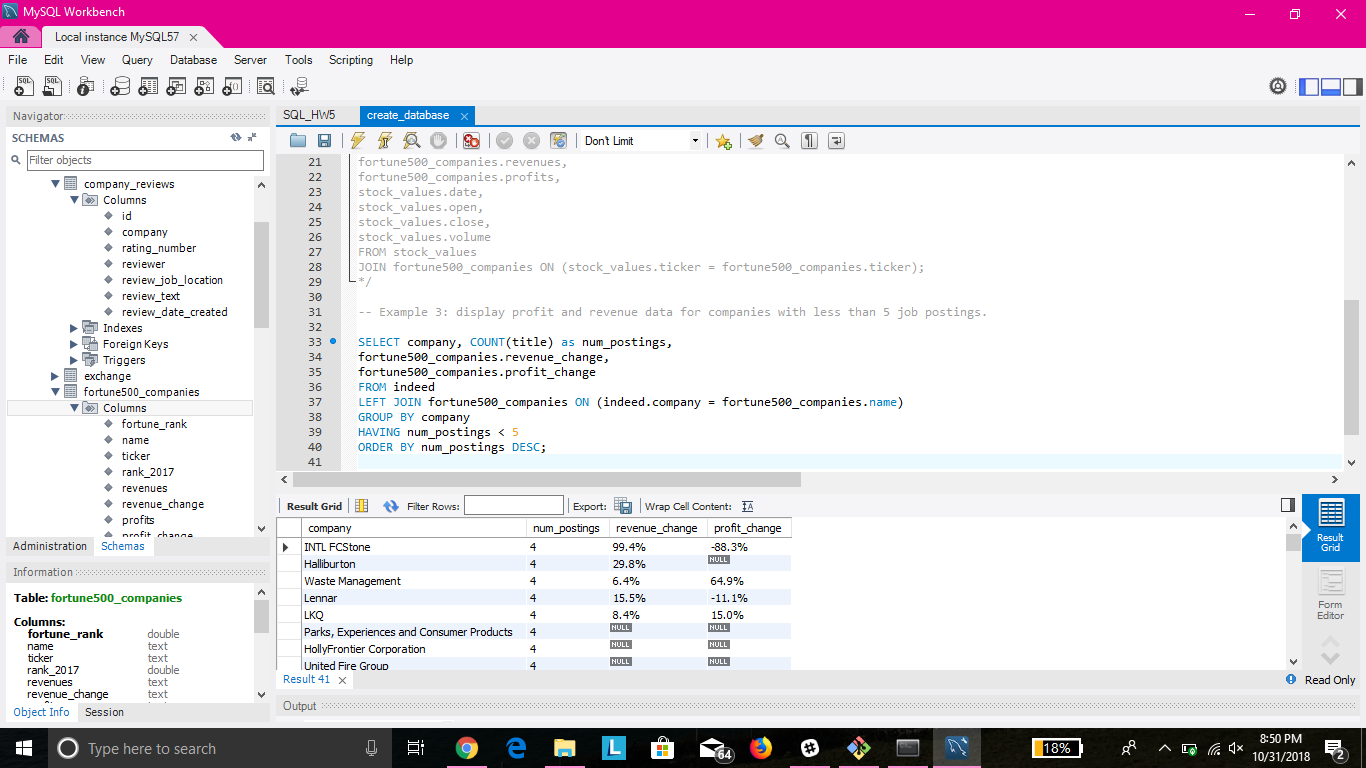


Figure 3: Job Postings vs. Revenue change

* Figure 3 uses the database to find if there is any correlation between the number of job postings, and the revenue/profit change at a fortune 500 company.