U.C Berkeley | Data Analytics

**ETL Project**

Quotes and Books

Misha | Namitha | Ivan | Eli



# A person wearing a hat Description automatically generated

# Introduction

The purpose of this project was a two parter - exploration of the powers of utilizing the ETL process, while developing a tool to dive deeper into a previous project to get more information out of it. We chose to look into the book scraping activity activity we reviewed in week 12 for our project.

The ETL process is laid out as a three step process.

**Extraction:** Reading the data from multiple sources, that are oftentimes not in the same format.

**Transformation:** Clean and structure the data to the desired format that will allow ease of manipulation.

**Load:** Writing the data into a database for storage and quick retrieval.

## Our Approach

## Extraction:

The dataset of interest for the extraction process looked at two different sources. The first source is a mockup of an online bookstore and contains information on roughly 1000 books. We were primarily interested in the title, description, price, and ratings. Since we were interested in books, and how they are related to quotes, the title and description were the first pieces of information we chose to gather. We decided to gather price and ratings data just in case we observed some interesting correlations during the ETL process.

The second source to scrape is a website containing quotes from many famous speakers. This dataset had many endpoints showing the quotes in undesirable formats. We were interested in scraping the quotes, speakers and available tags. Again, similarly to the books website scraping, we were primarily interested in the quotes, and speakers - but we also grabbed the tags in case we observed any interesting correlations during the ETL process.

**Source-1:** [Books to scrape](http://books.toscrape.com/)

**Source-2:** [Quotes to scrape](http://quotes.toscrape.com/)

## Transformation:

During the transformation process step, we utilized a Jupyter notebook to read in the data and begin previewing formats with Pandas and .head() function. At first glance, we noticed the data was ordered in a decent format, but lacked any obvious columns to merge dataframes on. We suspected that if somebody liked a particular quote, they might like books about or by the quote’s speaker. So, we created a dataframe to store the unique speaker names (24 in total, out of 100 quotes), then we wrote an iterative loop to look for each name within the bookstore by scanning product descriptions. We created a dataframe to store these “matches”. Within the 1000 books, there were 19 matches.

## Loading:

After transforming the final data frames, books, matches, quotes and speakers were loaded onto SQL through python by creating an engine and using the pd.to\_sql method.

During the loading process, matches data frame with book\_id & speaker\_id were sent straight to a matches table in postgresQL database. Records were concise and clean.

These are the ETL process steps we applied for this project:

1. Choose data to extract.
2. Scrape data and place it into a dataframe.
3. Transform  Data Frames by most relevant data.
4. Create database connection
5. Load the file into a relational SQL database for future business or analytical use.
6. Confirmed data has been added by querying matches table

### ETL Schema

A screenshot of a cell phone

Description automatically generated

### Conclusion

After viewing the populated tables, we observed within the 1000 books scraped, there were 19 matched books that had the quoted speaker referenced.

We were ultimately forced to stagger our approach (ETLTL); during the loading process, we realized that our references to database foreign keys (book\_id and speaker\_id) within the matches table were incorrectly looking for *indexical* values from the dataframe. Because the former started incrementing from a value of 1, and the latter started incrementing from a value of 0, our attempt to match records fell prey to the infamous “off by one” problem. We corrected the issue by re-coding the matches dataframe so that it contained titles and speakers, loading the books and speakers databases back into Pandas, and merging them with the original matches dataframe to grab the correct database IDs for each table, before ultimately loading the matches dataframe into Postgres.

### Output

|  |  |
| --- | --- |
| A screenshot of a cell phone  Description automatically generated | A screenshot of a cell phone  Description automatically generated |