Project 2- ETL

**Group Name: City Scavengers**

**Team Members:**

* Kamila Ibragimova
* Stacey Scott
* Ramon Martinez
* Raghu Pucha
* Aditya Bagchi

**Group Proposal:** There will be use of 2 csv files and an open weather API key used to determine and compare the index number of the cost of living, average income, and weather to give users insight into the most appropriate cities to live based on their current or future needs.

**Dataset Resources:**

* <https://data.census.gov/cedsci/table?q=average%20income%20&g=0100000US,%240400000,%243100000>
* [Kaggle: Your Home for Data Science](https://www.kaggle.com/account/login?titleType=dataset-downloads&showDatasetDownloadSkip=False&messageId=datasetsWelcome&returnUrl=%2Fdatasets%2Fdebdutta%2Fcost-of-living-index-by-country%3Fresource%3Ddownload)
* <http://api.openweathermap.org/data/2.5/weather?>

**ETL Process:**

Most of the ETL process was completed in Jupyter Notebook

*Extractions:*

* Datasets: 2 csv files, one cost of living index by cities in the world and the one on US census per income by state, API from Open Weather giving US cities by weather.
* API and postgres keys were configured and csv files imported into Jupyter Notebook along with other dependencies.

*Transform:*

Average Income File

* Average Income File was stored/extracted into database
* Creation and transformation of data frame was completed by choosing required columns of states/cities and incomes, renaming columns, checking for duplicates in cities and states, and deleting unnecessary rows with states since only cities were required.
* Clean table showing only family/non-family income, city/state was created.

Cost of Living File

* Average Cost of Living File was stored/extracted into database
* Creation and transformation of data frame was completed by choosing required columns of cities, cost of living index, grocery index, and rent index, renaming columns, and split was completed to include city, state, country.
* Filter for US cities only.
* Clean table showing city, cost of living index, grocery index, rent index, and state was created.

OpenWeather API

* Find temperature max, min and humidity for these US cities using API url.
* Build partial query URL
* Set up lists to hold response information
* Loop through the list of cities and perform a request for data on each
* Create a data frame from cities, latitude, and temperature

*Loading:*

Connection to Postgres

* A database called Cost\_living was created in postgres and then code engine = create\_engine(f'postgresql://postgres:{p\_key}@localhost:5432/Cost\_Living')

connection = engine.connect() ran

* Three tables were then created in postgres and each verified in jupyter
* Files were then joined together for comparison, using inner join for tables
* Connection was then closed

**Type of Database used:**

A structured database was used owing to the need to be specific on cities, income, and weather.

**Creation of ERD:**

* Three ERD tables were created to identify the primary and foreign keys and relationships between tables.
* ERD tables were then imported into postgres for storage for future use and further comparisons for users to have insight on cities, cost, weather based on needs

Graphical user interface, application

Description automatically generated

**Insight:**

Based on the tables created Boulder Colorado was the best place economically for cost, weather, and income.