ETL Project - Technical Report - Group # 1

Group Members:

Parijat Bhardwaj Hali Bielser Kristen Harnack Brittany Ouellette Cheyenne Parrott

Steps to Reproduce

- 1. Clone the Git Repository located at: https://github.com/hbielser/DataMiniProject 10.20
- 2. Using pgAdmin 4, create a database called project_test2
- 3. In pgAdmin 4, open the file called create_tables.sql using the query tool and run it to create the tables
 - a. Close the file to avoid re-creating tables
- 4. Open the file called DataMiniProject1_CleanData_Final.ipynb using Jupyter Notebook and run all cells
- 5. Navigate to pgAdmin 4 and open a new query tool
- 6. Open the file called query_join.sql and run
- 7. The results will show which dangerous buildings are on the list to be demolished
- 8. Once finished reviewing the data in pgAdmin 4, be sure to close the connection

Note: Perform steps 2 and 3 only upon initial database load.

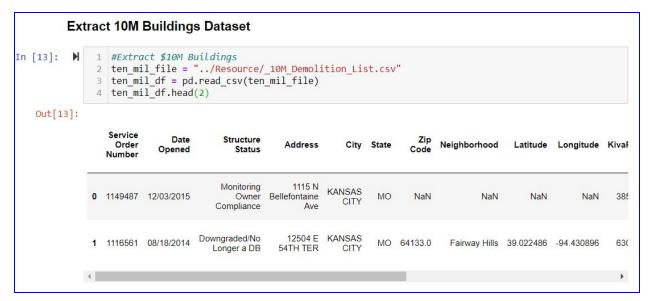
Thoughts for future: Ensure that existing records are not re-inserted into the database each time the Python file runs.

Project Report

• Extract:

- We chose two datasets from Open Data KC in .csv format. The data is also available via an API, and due to the frequent updates, it would be a good idea to utilize the API if we continue to work with this information for future projects.
- We compared structures listed on the Dangerous Buildings List to those scheduled to be demolished with funds from the \$10 million demolition program. We chose to compare this information because the datasets shared latitude and longitude information that we could use to join them.
 Many of the other datasets available from Open Data KC were subsets of one another.

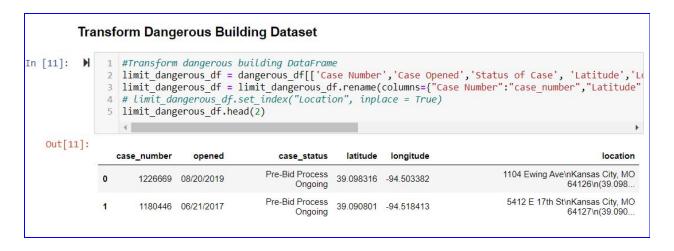




Additional information on the \$10 million demolition list program:
 https://www.kshb.com/news/local-news/kansas-city-surpasses-goal-in-2-year-dangerous-buildings-initiative.

• Transform:

To clean the data, we started with our first dataset- Dangerous Buildings
List - and we had to narrow down and rename the columns that we were
needing the information from. Then, we cleaned the second dataset \$10M Demolition List - to pull the pertinent columns needed which will be
used in the join later on.





Load:

• We loaded the final data frames into the SQL database where we created the tables used, with the 2 tables that we created - build and demo - we joined the tables on the columns "latitude" and "longitude". We chose to utilize an SQL database because we knew the format of the data, and it will remain in this format as new records are added. We chose to join the tables on latitude and longitude because the data points were unique by location and available in each data set.

```
Load Dataframes into Database

In [19]:  

# used replace instead of append. With replace we are creating the tables with given table name 2 # Thats why when we confirm the tabeles second time it gives us the table names 3 limit_dangerous_df.to_sql(name='build', con=engine, if_exists='append', index=False)

In [20]:  

I limit_ten_mil_df.to_sql(name='demo', con=engine, if_exists='append', index=False)

In [21]:  

# Confirm tables engine.table_names()

Out[21]: ['build', 'demo']
```

