Wednesday, 10/14/2020

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.

Extract Dangerous Building Dataset

```
[10]:
#Extract Dangerous Buildings data
dangerous_file = "../Resource/Dangerous_Buildings_List.csv"
dangerous_df = pd.read_csv(dangerous_file)
dangerous_df.head(2)
```

Out[10]:

	Case Number	Address	ZIP Code	Case Opened	Kiva PIN	Status of Case	Latitude	Longitude	Location
0	1226669	1104 Ewing Ave	64126	08/20/2019	5246	Pre-Bid Process Ongoing	39.098316	-94.503382	1104 Ewing Ave\nKansas City, MO 64126\n(39.098...
1	1180446	5412 E 17th St	64127	06/21/2017	9421	Pre-Bid Process Ongoing	39.090801	-94.518413	5412 E 17th St\nKansas City, MO 64127\n(39.090...

Extract 10M Buildings Dataset

```
In [13]:
#Extract $10M Buildings
ten_mil_file = "../Resource/_10M_Demolition_List.csv"
ten_mil_df = pd.read_csv(ten_mil_file)
ten_mil_df.head(2)
```

Out[13]:

	Service Order Number	Date Opened	Structure Status	Address	City	State	Zip Code	Neighborhood	Latitude	Longitude	Kival
0	1149487	12/03/2015	Monitoring Owner Compliance	1115 N Bellefontaine Ave	KANSAS CITY	MO	NaN	NaN	NaN	NaN	385
1	1116561	08/18/2014	Downgraded/No Longer a DB	12504 E 54TH TER	KANSAS CITY	MO	64133.0	Fairway Hills	39.022486	-94.430896	630

- 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.

Transform Dangerous Building Dataset

In [11]:
```
#Transform dangerous building DataFrame
limit_dangerous_df = dangerous_df[['Case Number','Case Opened','Status of Case', 'Latitude','Lo
limit_dangerous_df = limit_dangerous_df.rename(columns={"Case Number":"case_number","Latitude"
# limit_dangerous_df.set_index("Location", inplace = True)
limit_dangerous_df.head(2)
```

Out[11]:

	case_number	opened	case_status	latitude	longitude	location
0	1226669	08/20/2019	Pre-Bid Process Ongoing	39.098316	-94.503382	1104 Ewing Ave\nKansas City, MO 64126\n(39.098...
1	1180446	06/21/2017	Pre-Bid Process Ongoing	39.090801	-94.518413	5412 E 17th St\nKansas City, MO 64127\n(39.090...

Transform 10M Buildings Dataset

In [14]:
```
#Transform $10M DataFrame
limit_ten_mil_df = ten_mil_df[['Service Order Number','Date Opened','Structure Status','Propert
limit_ten_mil_df = limit_ten_mil_df.rename(columns={"Service Order Number":"service_order_numbe
# limit_ten_mil_df.set_index('Location', inplace = True)
limit_ten_mil_df.head(2)
```

Out[14]:

	service_order_number	opened	structure_status	property_owner	structure_type	structure_rating	latitude	lon
0	1149487	12/03/2015	Monitoring Owner Compliance	Private	House	Repair/Receivership	NaN	
1	1116561	08/18/2014	Downgraded/No Longer a DB	Private	House	Regular Demolition	39.022486	-94.4

- **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
# Thats why when we confirm the tabeles second time it gives us the table names
limit_dangerous_df.to_sql(name='build', con=engine, if_exists='append', index=False)
```

In [20]:
```
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']

Querying the tables

In [22]:
```
connection = engine.connect()
```

In [23]:
```
build_query = pd.read_sql ("SELECT * FROM build ",connection )

build_query.head(2)
```

Out[23]:

	case_number	opened	case_status	location	latitude	longitude
0	1226669	2019-08-20	Pre-Bid Process Ongoing	1104 Ewing Ave\nKansas City, MO 64126\n(39.098...	39.098316	-94.503382
1	1180446	2017-06-21	Pre-Bid Process Ongoing	5412 E 17th St\nKansas City, MO 64127\n(39.090...	39.090801	-94.518413

In [24]:
```
demo_query  = pd.read_sql ("SELECT*FROM demo",connection )

demo_query.head(2)
```

Out[24]:

	service_order_number	opened	structure_status	neighborhood	property_owner	structure_type	structure_rating
0	1149487	12/03/2015	Monitoring Owner Compliance	None	Private	House	Repair/Receivership
1	1116561	08/18/2014	Downgraded/No Longer a DB	None	Private	House	Regular Demolition