**ETL PROJECT**

**RESTAURANT INSPECTIONS vs. RAT SIGHTINGS**

**Team**: Alla Zbirun, Hernan Hernandez, Ryan Michel

**EXECUTIVE SUMMARY**:

Rats in NYC persist to be an issue. As residents of NY, we volunteered to investigate this issue and determine if there is a correlation between restaurant inspections by NY Health department and rat sightings in the area. This document summarizes step by set process the team performed from extraction of the data, through data clean up and normalization.

**EXTRACTION**:

For this project we extracted data from three different sources -

1. NYC Open Data (<https://data.cityofnewyork.us/Health/DOHMH-New-York-City-Restaurant-Inspection-Results/43nn-pn8j/data>) – we were able to download csv files of data containing information on violations determined under inspection conducted during preceding three years for restaurants and college cafeterias
2. Kaggle.com ( <https://www.kaggle.com/new-york-city/nyc-rat-sightings>) – we were able to download csv files on rats sightings in the NYC areas.
3. Yelp.com ( <https://www.yelp.com/dataset>) – we were able to download a JASON file which contained data of restaurant credentials, review ratings and much more info on a global scale.

**TRANSFORMATION**:

The data we received was more than we needed and held number of inconsistencies or missing information. In order to normalize the data we performed following steps in Jupyter notebook to extract only sought-after elements and wrote it into a databases –

1. NYC Open Data - The restaurant dataset consisted of 394,890 rows with 26 columns. In viewing the dataset we noted obvious columns that we removed as the data had nothing to do with the focus of our project. In total we removed 8 columns. We next looked for the sum of nulls in each column. Out of the 18 columns remaining, only 5 didn't have nulls. We decided to drop the rows that had null values. Lastly, the dataset contained 7 rows with duplications and we removed all of them. After the cleaning, the final dataset contains 193,397 rows of the restaurant inspection data.
2. Kaggle data- There were a total of 52 columns found in the original dataset. Since the goal was to see if there was a relationship with health grade ratings and number of rat findings, we only preserved a zip-code column. Further, we used a groupby function to determine number of rat findings by zip-code; which we then saved the total number of observations as a column in the new dataframe. We removed all null values and one entry that listed incomplete zip-code. The final dataset contains 186 rows of rat sightings by zip-code.
3. Yelp data – Dataset file was downloaded in TAR archive format. In order to extract data files, we downloaded/utilized 7-Zip, which decompressed the file and extracted contents of the archive into a folder. Out of the data available (see<https://www.yelp.com/dataset/documentation/main> for details) we chose business.json file as it contained most relevant information (i.e. - address, restaurant category, ratings, lat & long, and review counts). After detailed review of the file contents we dropped number of insignificant columns (attributes, hours, is\_open). We re-ordered columns for more convenient read of the data by sending column headers to a list; then changing the order of the appearance and reflecting it in the dataframe. Further, we groupped data by states and noted that the dataset is not exclusive to USA and contains only 22 entries for NYS. We extracted NYS data and were able to determine that this set contained only one entry for NYC which was a beauty salon. At this point we decided to drop this dataset from the analysis.

Please see Data Dictionary table at the bottom or the report for detailed listing and description of the data elements retained for queering.

**LOAD:**

Based on the retained data we generated following dataframes and uploaded into pgAdmin for convenience of segregation, maintenance and queering process (see ERD chart and table creation script in Exhibit B):

* Based on NYC OpenData set –
  + Violations – holds a list of violations and their descriptions
  + Restaurant – holds a list of restaurants in the database including name of the business, borough, zip-code, type of cuisine and grade.
  + Inspections – holds a list of inspection claim ids, violation does, zip-codes and inspection date
  + Geo-Location – holds a list of inspection claim ids with relevant latitude and longitude of the restaurant.
* Rats – based on the rat sighting analysis which shows zip-code and associated rat sightings.

**QUERIES:**

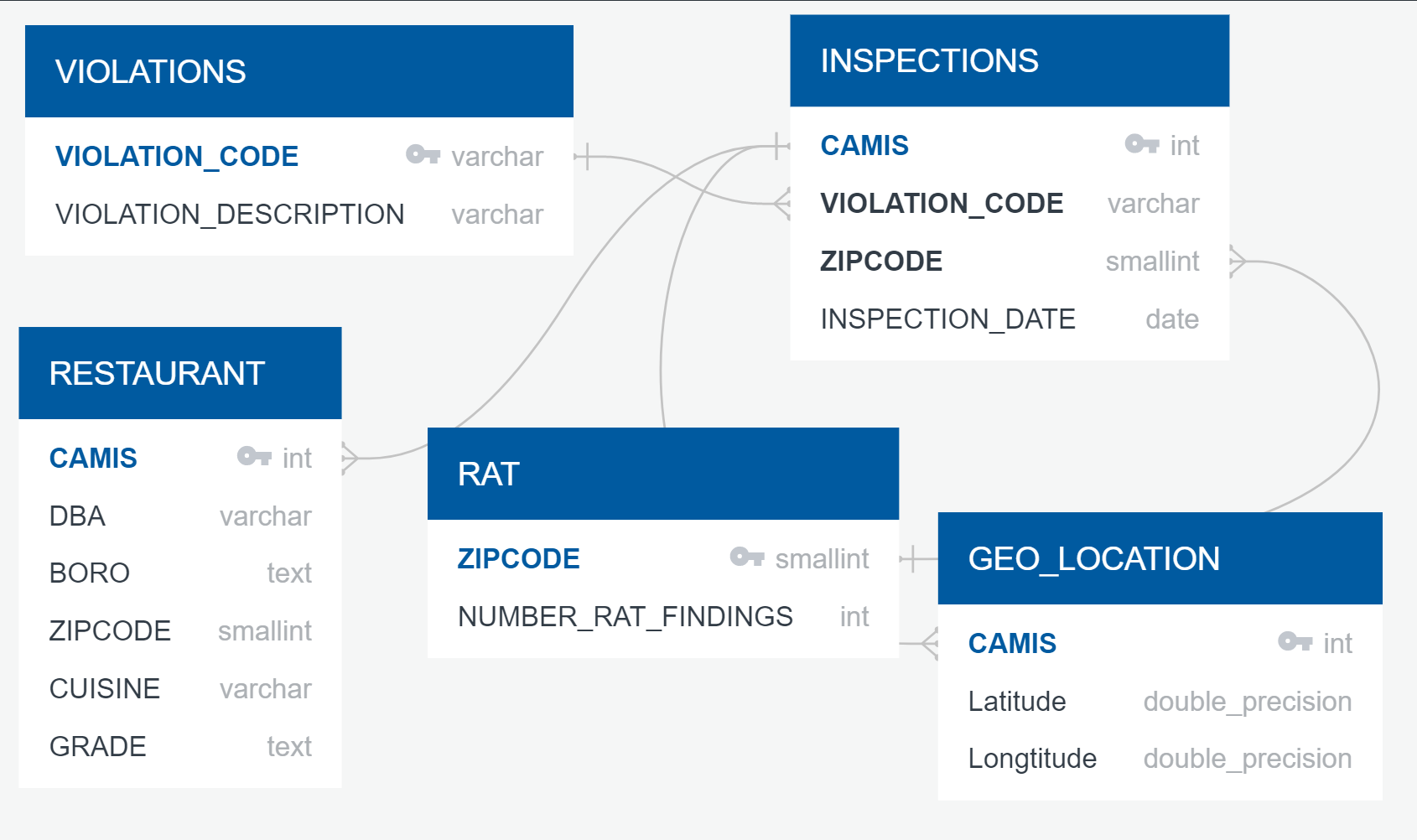
We ran following queries based on the developed dataset –

1. Zip-codes with highest rat sightings
2. Restaurants in the Columbia University zip code
3. Restaurants with P-grade (P= Grade Pending issued on re-opening following an initial inspection that resulted in a closure)
4. All restaurants with A Grade
5. Italian restaurants

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **DATA DICTIONARY -** | |  |  |  |
|  |  |  |  |  |
| **FIELD** | **TYPE** | **FIELD SIZE / FORMAT** | **DESCRIPTION** | **EXAMPLE** |
| CAMIS | integer |  | This is an unique identifier for the entity (restaurant); 10-digit integer, static per restaurant permit | 50090702 |
| DBA | varchar | 255 | This field represents the name (doing business as) of the entity (restaurant); Public business name, may change at discretion of restaurant owner | KIND OF SOUL |
| BORO | text |  | Borough in which the entity (restaurant) is located. NOTE: There may be discrepancies between zip code and listed boro due to differences in an establishment's mailing address and physical location | • 1 = MANHATTAN • 2 = BRONX • 3 = BROOKLYN • 4 = QUEENS • 5 = STATEN ISLAND • Missing |
| ZIPCODE | smallint |  | Zip code of establishment/location | 11217 |
| CUISINE DESCRIPTION | varchar | 255 | This field describes the entity (restaurant) cuisine. ; Optional field provided by provided by restaurant owner/manager | American |
| INSPECTION DATE | text | DD/MM/YYYY | This field represents the date of inspection; NOTE: Inspection dates of 1/1/1900 mean an establishment has not yet had an inspection | 05/01/2019 |
| VIOLATION CODE | varchar | 5 | Violation code associated with an establishment (restaurant) inspection | 10F |
| VIOLATION DECRIPTION | varchar |  | Violation description associated with an establishment (restaurant) inspection | Food Protection Certificate not held by supervisor of food operations. |
| GRADE | text |  | Grade associated with the inspection | • N = Not Yet Graded• A = Grade A• B = Grade B• C = Grade C• Z = Grade Pending• P= Grade Pending issued on re-opening following an initial inspection that resulted in a closure |
| LATITUDE | double\_precision |  | geo coordinate | 40.68558614 |
| LONGITUDE | double\_precision |  | geo coordinate | -74.00501092 |
| NUMBER OF FINDINGS | integer |  | This field represents number of rat sightings per zip-code calculated based on the information pulled from kaggle.com | 1112 |

**Exhibit B**

**ERD Chart**



**Table Creation Script**

**Violations Table:**

-- Table: public.violations

-- DROP TABLE public.violations;

CREATE TABLE public.violations

(

violation\_code character varying(5) COLLATE pg\_catalog."default" NOT NULL,

violation\_description character varying COLLATE pg\_catalog."default",

CONSTRAINT violations\_pkey PRIMARY KEY (violation\_code)

)

TABLESPACE pg\_default;

ALTER TABLE public.violations

OWNER to admin;

**Restaurants table:**

-- Table: public.restaurants

-- DROP TABLE public.restaurants;

CREATE TABLE public.restaurants

(

camis integer NOT NULL,

dba character varying(255) COLLATE pg\_catalog."default",

boro text COLLATE pg\_catalog."default",

zipcode smallint,

cuisine character varying(255) COLLATE pg\_catalog."default",

grade text COLLATE pg\_catalog."default",

CONSTRAINT restaurants\_pkey PRIMARY KEY (camis)

)

TABLESPACE pg\_default;

ALTER TABLE public.restaurants

OWNER to admin;

**Rats table:**

-- Table: public.rats

-- DROP TABLE public.rats;

CREATE TABLE public.rats

(

zipcode smallint NOT NULL,

rat\_findings integer,

CONSTRAINT rats\_pkey PRIMARY KEY (zipcode)

)

TABLESPACE pg\_default;

ALTER TABLE public.rats

OWNER to admin;

**Inspections table:**

-- Table: public.inspections

-- DROP TABLE public.inspections;

CREATE TABLE public.inspections

(

camis integer NOT NULL,

violation\_code character varying(5) COLLATE pg\_catalog."default",

zipcode smallint,

inspection\_date text COLLATE pg\_catalog."default",

CONSTRAINT inspections\_pkey PRIMARY KEY (camis)

)

TABLESPACE pg\_default;

ALTER TABLE public.inspections

OWNER to admin;

**Geo\_locations table:**

-- Table: public.geo\_locations

-- DROP TABLE public.geo\_locations;

CREATE TABLE public.geo\_locations

(

camis integer NOT NULL,

"Latitude" double precision NOT NULL,

"Longitude" double precision NOT NULL,

CONSTRAINT geo\_locations\_pkey PRIMARY KEY (camis)

)

TABLESPACE pg\_default;

ALTER TABLE public.geo\_locations

OWNER to admin;