Best State to Open Restaurants

Identification and Preparation of Data for Analytics

By Bootcamp Grp X

**TABLE OF CONTENTS**

1. Introduction
   1. Summary
   2. Scope
   3. Technologies and resource contributions
   4. Definitions, Acronyms and Abbreviations
2. ETL Details
   1. Data Import/Extract Sources and Method
   2. Data Acquisition
   3. Data Transform
   4. Data Integrity
   5. Data Refresh Frequency
   6. Data Security
   7. Data Loading and Availability
3. Data Quality
4. Programs
5. Appendix

|  |  |
| --- | --- |
| **1.** | **INTRODUCTION** |

In addition to chosen data sources, available input data formats, and the data storage engine for the ETL, the programming environment, software packages, and developed software are described below.  
Comments on different aspects of data retrieval describe the limitation of usage as well as possible solutions for improvement of usability of retrieved data including data maintenance.

# 1.1 Business Summary

The client, a restaurant group offering multiple types of cuisine within its wholly owned holding company is looking to open ten new restaurants within a single state. The client would like to open the locations based on high consumer spending patterns, low market saturation for that type of cuisine, and high ratings for that restaurant category.

The ETL deliverable should allow the restaurant group to view the data to foster a decision in the restaurant openings analysis. Moreover, the client should be able to test at least one hypothesis.

**Hypothesis**

Based on consumer spending, market saturation, and restaurant ratings, one state for each restaurant category is better to open ten restaurants than the remaining forty-nine states.

Null Hypothesis: There is no difference between the fifty states in terms of consumer spending, market saturation, and the restaurant ratings variables.

ANOVA testing could be utilized to test the fifty-state sample for each of the three variables. If the data was cleanly available for each year, a regression analysis could be used to determine the impact of each variable on the other to further predict if a state under consideration might be better suited based on trend (or in the case if a handful of states bubble to the top of the analysis).

# 

# 1.3 Scope

The data sources within scope of this project include Yelp.com, BEA.gov, and other applicable files freely available on the internet containing state related data such as populations and abbreviation conventions.   
The years under consideration are limited to the available data. For instance, BEA has consumer spending data distributed in the restaurant category only for 2017 with the ability to calculate prior year 2016. Population data and overall consumer spending date back three full years. Conversely, the data from Yelp is current as opposed to historical.   
For our restaurant group client to decide on the state’s locations, inflation multipliers have to be utilized to transform the historical data to estimated current data as well as retrograde data to coincide with prior years. These inflation multipliers for predictive analysis are out of scope for the project deliverable. Additionally, other components that have been ruled out of scope during the project discussion include the following: (1) employment data by state, and (2) personal income by state as each were deemed loose derivatives of discretionary restaurant spending.

# 1.4 Technologies and resource contributions

The tech stack includes the following: PostgreSQL, Python, Jupyter Notebook, SQLAlchemy, API calls, JSON files, MS Excel files and various Pandas Libraries to support the development process.

Tessy Azogu (Data Sources, Report Generation)

Martin Hrbac (Database Design, ETL design and implementation, Report Generation)

Sunil Joshi (Statistics Themes, Data Sources, Data Analysis, Report Generation)

Denise McKinley (Data Sources, Report Generation)

Ninglian (Melanie) Wang (API Calls)

# 1.5 Definitions, Acronyms and Abbreviations

Market Saturation: Number of restaurant locations in a state by category

PCE: Per Capita Expenditure

FIPS: Federal Information Processing Standard (one- or two-digit code to uniquely identify to each state)

|  |  |
| --- | --- |
| **2.** | **ETL DETAILS** |

Results of basic problem analysis pointed out probable data sources. After evaluation of the proposed data resources RDBMS was identified as the most suitable storage engine, Postgres was used for ETL package development.

# 2.1 Data Import, Extract Sources and Method

This table provides information about the data.

|  |  |
| --- | --- |
| Data | Details |
| State | Url: <https://www.nrcs.usda.gov/wps/portal/nrcs/detail/?cid=nrcs143_013696> Format: MS Excel Dimensions: 2 Data Updates, Load: Static, Once Permissions: n/a |
| State Population | Url: <https://www.census.gov/newsroom/press-kits/2018/pop-estimates-national-state.html> Format: MS Excel Dimensions: 3 Data Updates, Load: Annually, Availability of New Data Permissions: n/a |
| State PCE | Url: <https://www.bea.gov/data/consumer-spending/state> Format: MS Excel Dimensions: 5 Data Updates, Load: Annually, Availability of New Data Permissions: n/a |
| Yelp | Url: <https://www.yelp.com/developers> Format: JSON Dimensions: 9 Data Updates, Load: Daily to API Restriction Maximum, Minimum For Each Analysis Permissions: see [Yelp documentation](https://www.yelp.com/developers/documentation/v3/rate_limiting) |

# 2.4 Data Acquisition

For the state, state population, state personal consumption expenditure sources, the MS Excel files are retrieved from their respective URL’s. The restaurant data from Yelp is retrieved using API calls.

Only two tabs from the state personal consumption expenditure MS Excel file were selected to generate the state PCE data – table one containing total personal consumption expenditure from 2015 to 2017 and table four containing per capita personal consumption expenditure for the “off-premises food and beverages” category for 2017. There is no immediate concern for growth of any of datasets outside the number of new restaurant openings depicted on Yelp. Each restaurant opening adds a row. Closures are handled as a simple Boolean replacement from “true” if the restaurant is open to “false” if closed so the number of records does not grow.

# 2.3 Data Transform

Of the six tables in the relation schema depicted in section 2.7 Data Loading and Availability below, the data in five are mainly imported from data sources. The remaining table restaurant\_category is a relational table between the restaurant table and the category table from Yelp. For all the tables, a new parameter named “id” was created as a surrogate key to uniquely identify each row of data. For every table excluding the state table, a new parameter named “modified\_date” was created to let the user know when that data was loaded. A “source id” column was constructed for all tables excluding the state table to inform the user of the data source – 1 for Yelp, 2 for State PCE, 3 for State Population, and 4 for State. Another interesting twist to data transformation was taking the “price\_range” in the restaurant table and converting strings ranging from “$” to “$$$$” to integers “1” to “4”.

Other notable items include keeping the FIPS number as a unique identifier in the state table and keeping the “alias” as a natural key in the category table.

# 2.4 Data Integrity

Retrieved data is stored in the normalized database. The database design enforces data integrity via references. Further, natural unique keys are used to enforce clean data import. Key values are cleaned up before data is inserted into database including removal of leading, trailing white space, usage of lowercase characters and ignoring empty values.

# See the table in 2.1 Data Import/Extract Sources and Method for details.

# 2.5 Data Refresh Frequency

# See the table in 2.1 Data Import/Extract Sources and Method for details.

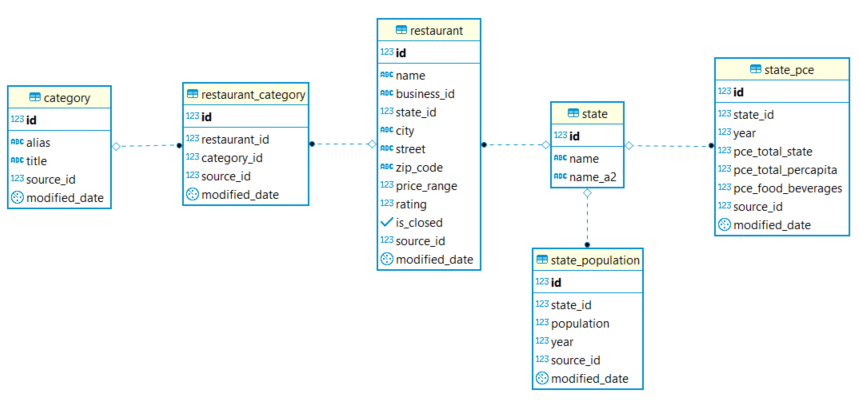
# 2.6 Data Security

There are no issues associated with data security for government and restaurant data for this application.

# 2.7 Data Loading and Availability

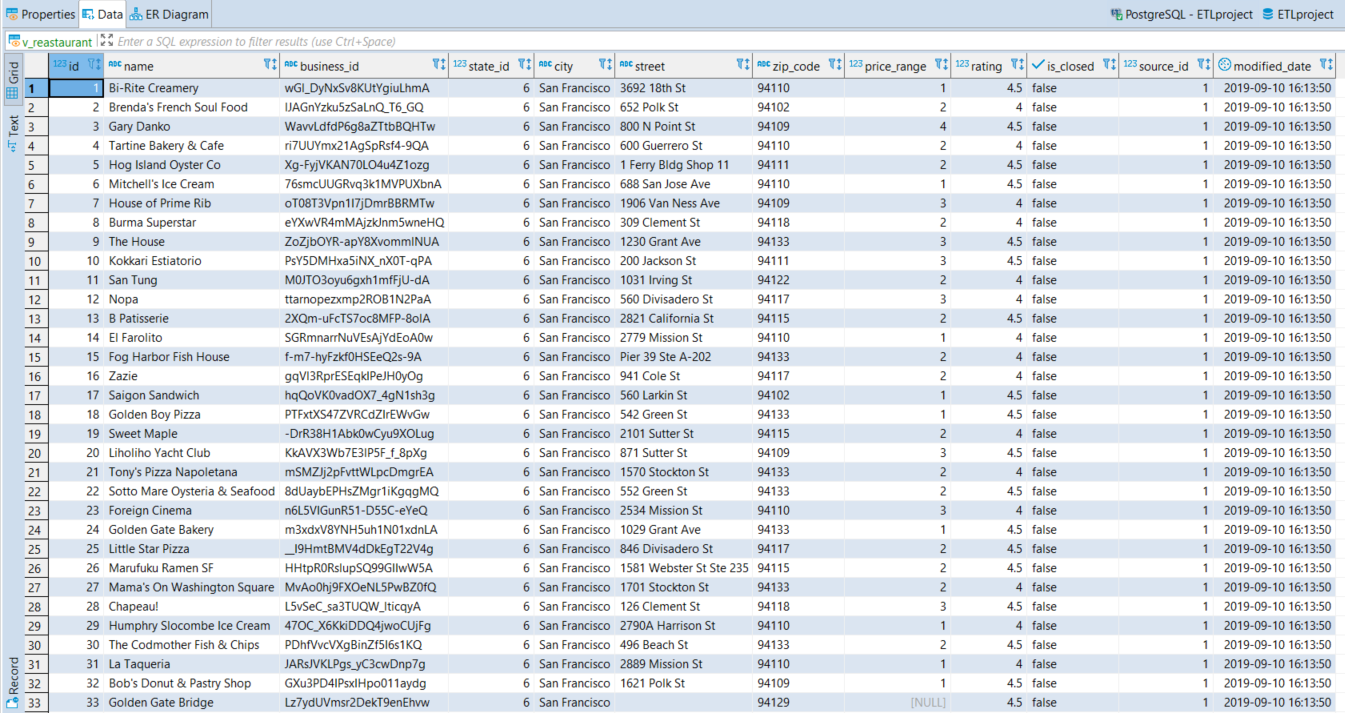
# See the table in 2.1 Data Import/Extract Sources and Method for details.

**Database Schema**

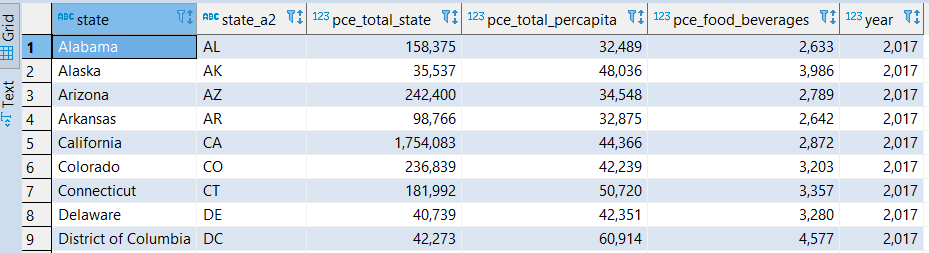


**Database views** have been created in the PostgreSQL database for the client to query data although stored procedures with a secure GUI could serve as a better alternative.

**Restaurant View**



**State PCE View**



|  |  |
| --- | --- |
| **3.** | **DATA QUALITY** |

Data quality is primarily addressed by input data verification in the loader and by the storage database design, using strict data normalization, and proper data types. The database has a built-in tracking mechanism which holds information about the data source and the date when the values were entered.

|  |  |
| --- | --- |
| **4.** | **PROGRAMS** |

For the original data load and testing 3 Jupyter notebooks were created as follows:  
•    ETL\_StatesData.ipynb – loads state data from Excel files into database,  
•    API\_yelp.ipynb – pulls data from Yelp API, stores data in local files, and  
•    ETL\_RestaurantData.ipynb – loads data from the files, loads it into the database

## 4.1 State data load

4.1.1 State Basics  
This table is comprised of state name, state id, and two characters state code. This data is considered static and is loaded once only.

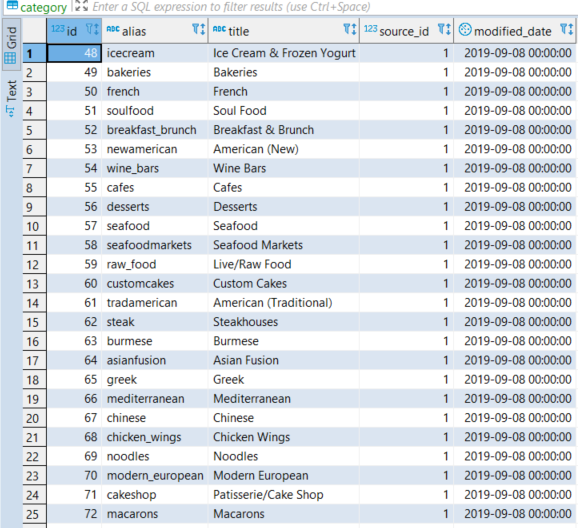
4.1.2 State PCE  
New data is released once a year. The data should be loaded after new data is released.  
Note: For the purpose of the project this data is considered static.

4.2 Restaurant data acquisition – Yelp API call  
This program utilizes and calls Yelp API and stores the raw data into local files. No data manipulation is done by this program. The data is stored in a JSON format.

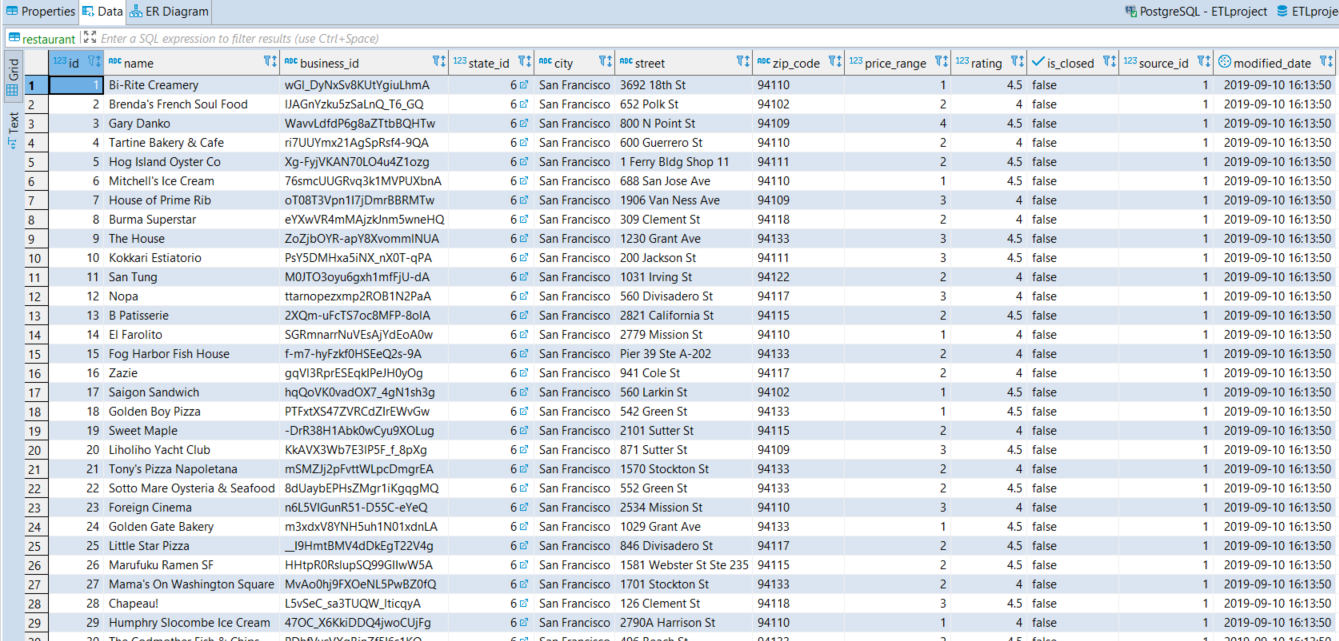
|  |  |
| --- | --- |
| **5.** | **APPENDIX** |

## 5.1 Tables Snapshots

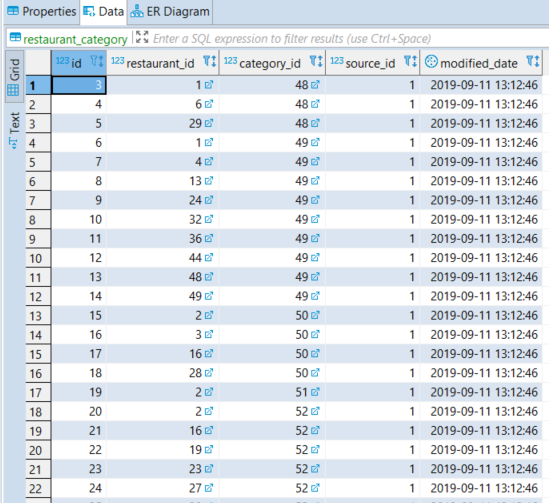
### 5.1.1 Category Table



### 5.1.2 Restaurant Table



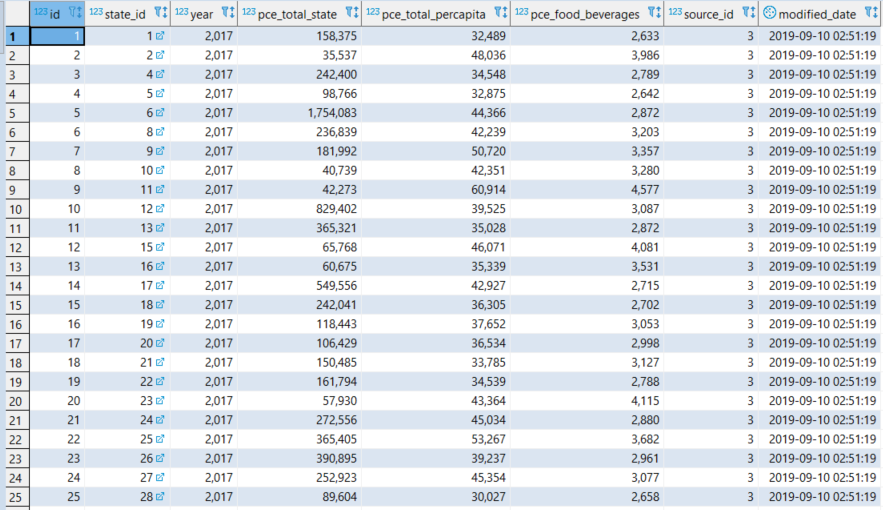
### 5.1.3 Restaurant\_Category Table



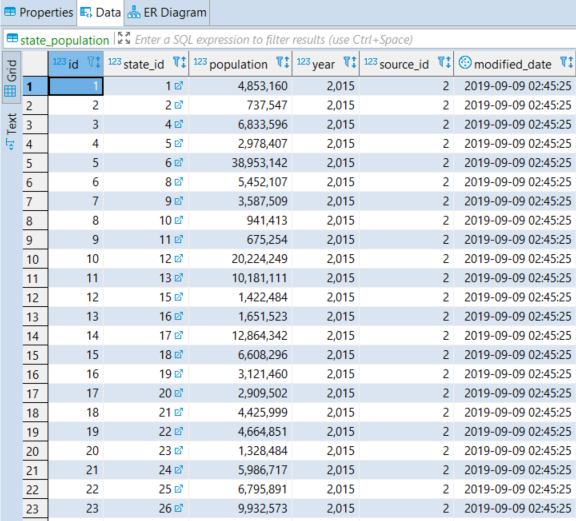
### 5.1.4 State Table



### 5.1.5 State\_PCE



### 5.1.6 State\_Population



## 5.2 Create Database

Create database SQL script

<https://github.com/marsion0245/GTATL5Project2/blob/master/Code/CreateDatabase.sql>

Create table views

<https://github.com/marsion0245/GTATL5Project2/blob/master/Code/CreateView.sql>