**Topic: Health Care Recommendation System.**

**Introduction to Health Care Recommendation System (HCRS):**

In traditional approach, one must document all the minute details and need to depend on person to person feedback for health-related queries. There will be no assurance on the information provided. Also, the information provided would be limited and it may not be reliable. Big Data provides the flexibility to store a large dataset without any constraints to the type of the data. Our proposed Health Care Recommendation System (HCRS) is being implemented using Big Data features.

Yelp dataset is selected for the implementation of this recommendation system. Yelp dataset contains all the information pertaining to hospitals, doctors, ratings and feedback of the patients (reviews) based on geographical location. The main purpose is practical implementation of Big Data concepts that are being taught in the class.

**Research Aspect Analyzed:**

In the Health Care Recommendation System (HCRS), the analyses of the Yelp Dataset revealed that the required information for building HCRS was within business.json and review.json files. The careful analysis of each field was carried out to implement HRS. This system will recommend the best available health center based on city/state driven by specialization, positive and negative reviews.

**Dataset used:**

Dataset source: Yelp Academic Dataset

Data Subset used: (1) business.json (109MB) (2) review.json (3.22GB)

**Format of data subsets:**

**Business.json**

{

'type': 'business',

'business\_id': (encrypted business id),

'name': (business name),

'neighborhoods': [(hood names)],

'full\_address': (localized address),

'city': (city),

'state': (state),

'latitude': latitude,

'longitude': longitude,

'stars': (star rating, rounded to half-stars),

'review\_count': review count,

'categories': [(localized category names)]

'open': True / False (corresponds to closed, not business hours),

'hours': {

(day\_of\_week): {

'open': (HH:MM),

'close': (HH:MM)

},

...

},

'attributes': {

(attribute\_name): (attribute\_value),

...

},

}

**Review.json**

{

'type': 'review',

'business\_id': (encrypted business id),

'user\_id': (encrypted user id),

'stars': (star rating, rounded to half-stars),

'text': (review text),

'date': (date, formatted like '2012-03-14'),

'votes': {(vote type): (count)},

}

**Methodology Followed:**

**Data Cleaning and Analysis:**

* Initially downloaded the yelp dataset.
* Extracted downloaded dataset to obtain JSON files.
* Converted JSON files into more readable format like csv/tsv using python script.
* According to the problem statement, confined our analysis to review and business datasets.
* Using Python script, cleaned and analyzed the datasets.

**Data Processing:**

* Created hive tables with the relevant fields for each dataset (business and review) with respect to the problem statement.
* Performed NLTK processing on review text field and added new column to review.csv file to distinguish between positive, negative and neutral reviews.
* Creating new tables by integrating required fields from review and business tables using hive query.
* Filtered the Health Care Centers with positive and negative review separately.
* New tables where created according to positive reviews with maximum review count specialization of doctors based on city/state.
* Above created tables were converted into csv format and was visualized in tableau.

**Process Flow:**

**Load raw data into HDFS**

**Converting JSON data into TSV/CSV using python script**

**Extracting Yelp Raw data in JSON format**

**Performed lateral view on categories and filtered Medical data**

**Analyzing the required fields from the dataset**

**Creating table and loading data for each dataset using hive**

**Selected distinct business based on city/state**

**Categorized positive, negative, and neutral review using NLTK**

**Merged both tables using hive**

**Saving result set in CSV format and loading it to tableau for GUI representation**

**Extracted hospitals with max reviews based on city**

**Implementation of HRS with Failures and Success:**

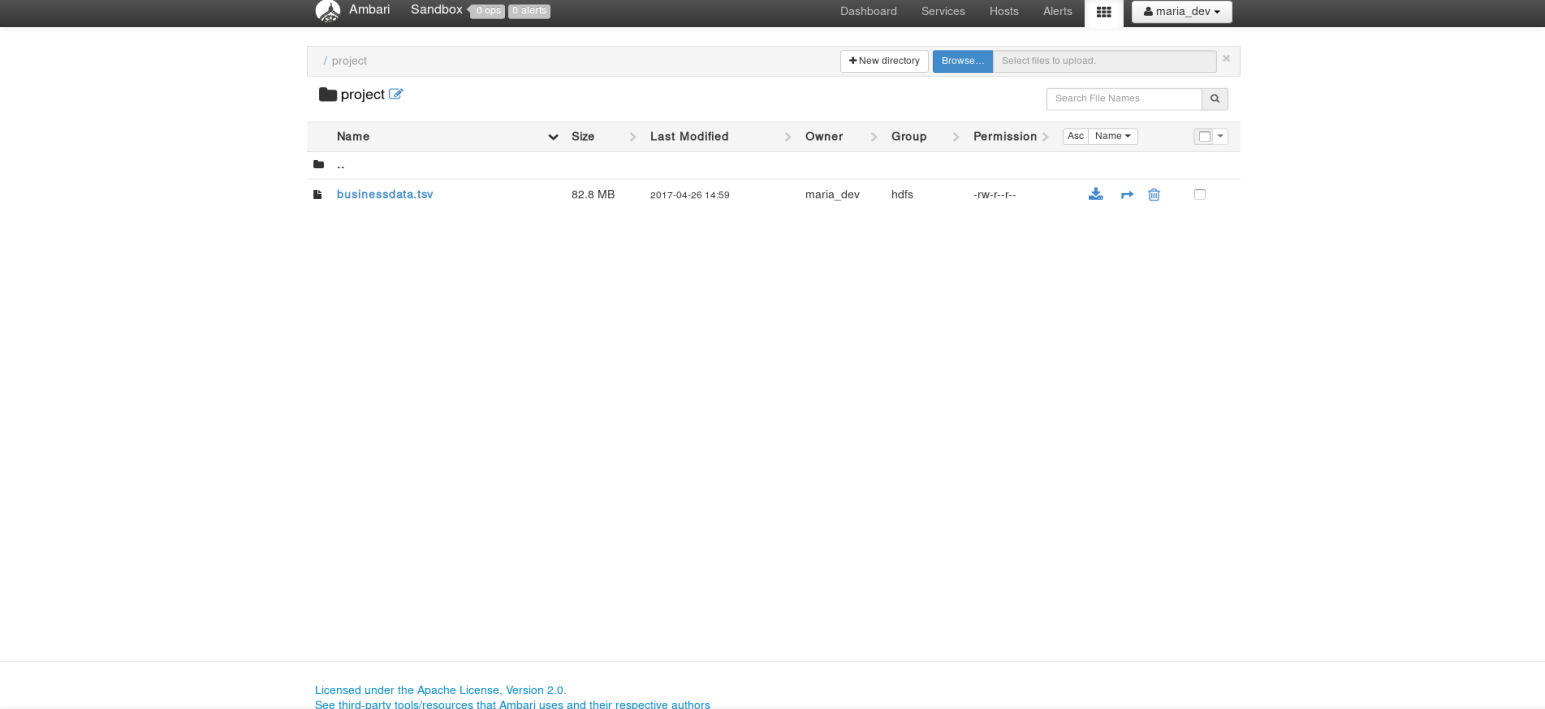
**Technology Used:** Horton sandbox ecosystem – (Hive, Pig and Ambari) and Tableau.

**1. Success in Converting dataset to readable formats using python and pig script.**

* Generated business.tsv and it was appropriate for creating hive tables.

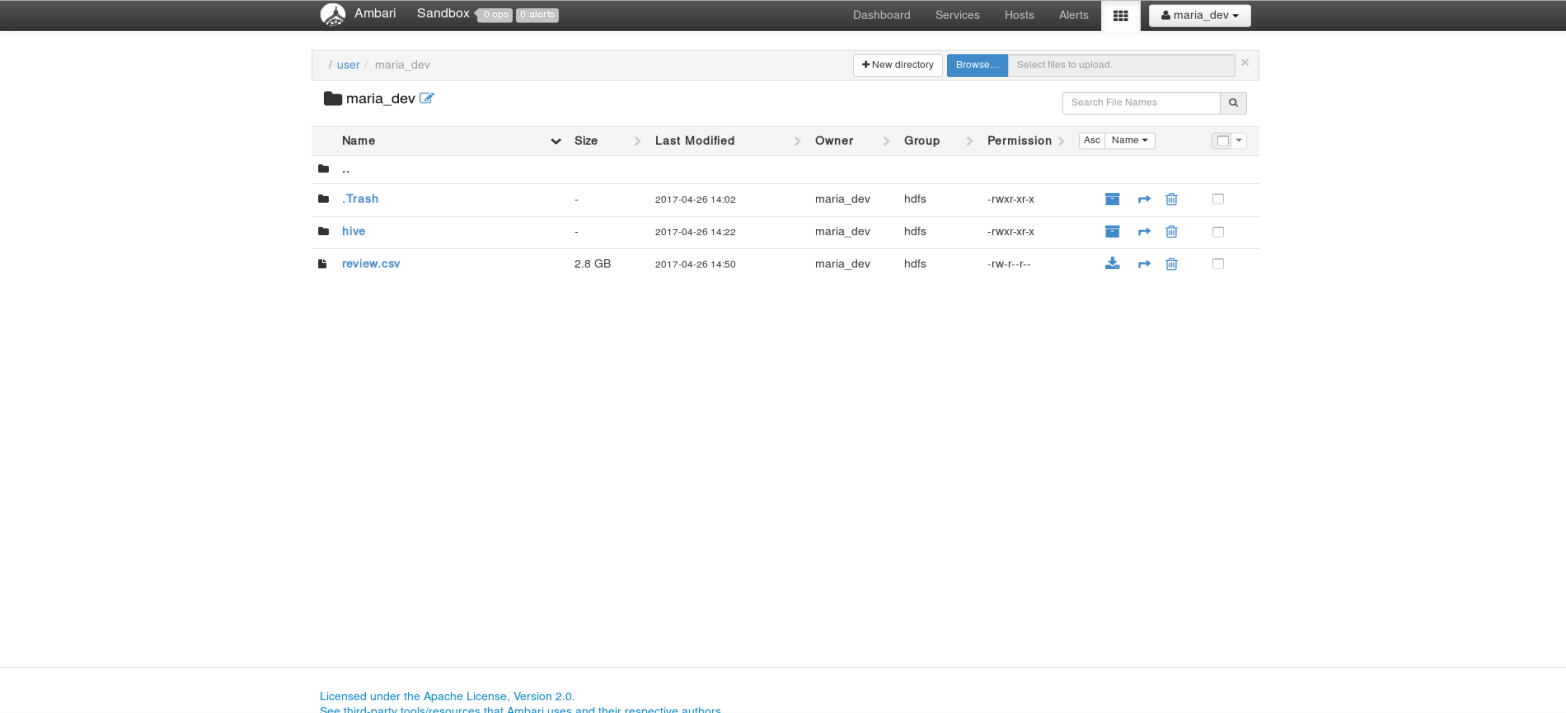
Python script for created business.tsv:

Uploaded business.tsv file on ambari:



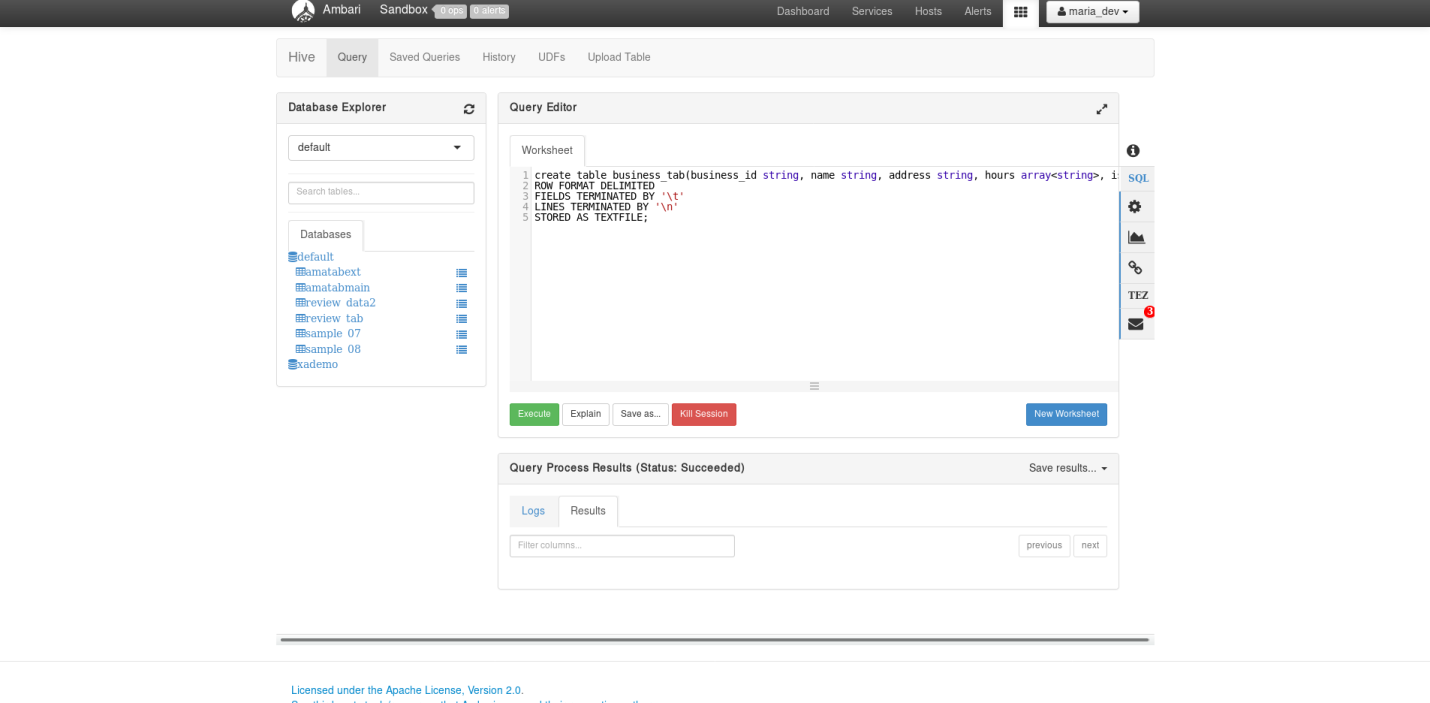
* Generated review.csv from the subset of original review.json.

Uploaded review.csv to ambari:

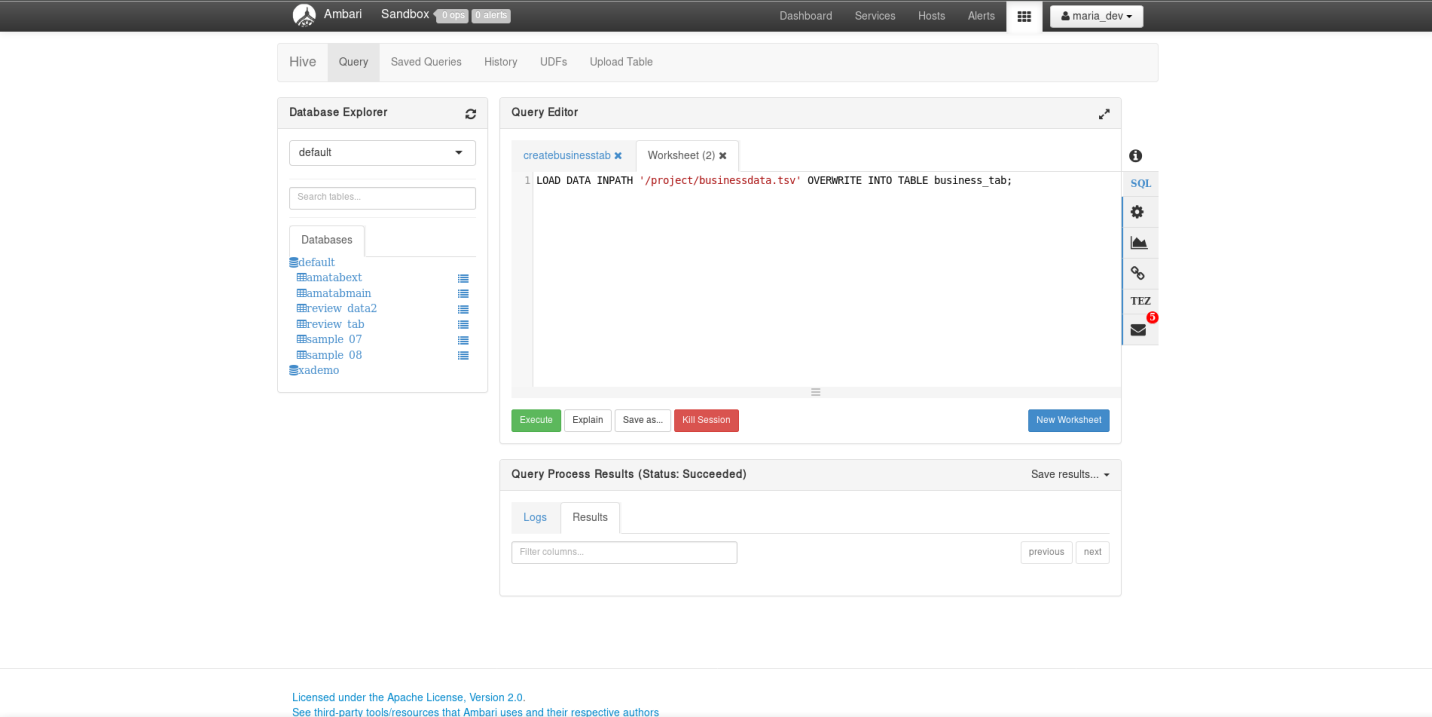


**2 Creation of table for both business.tsv and review.csv**

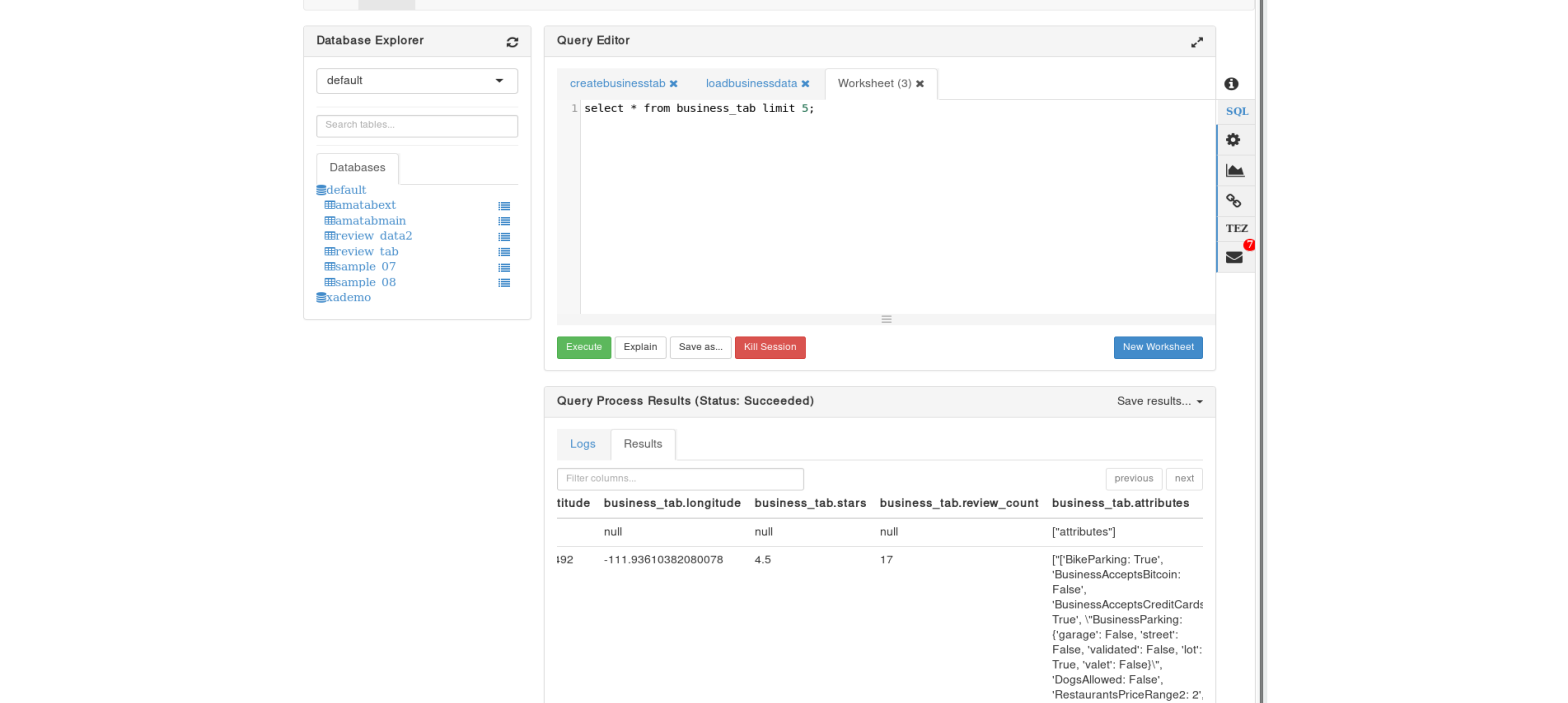
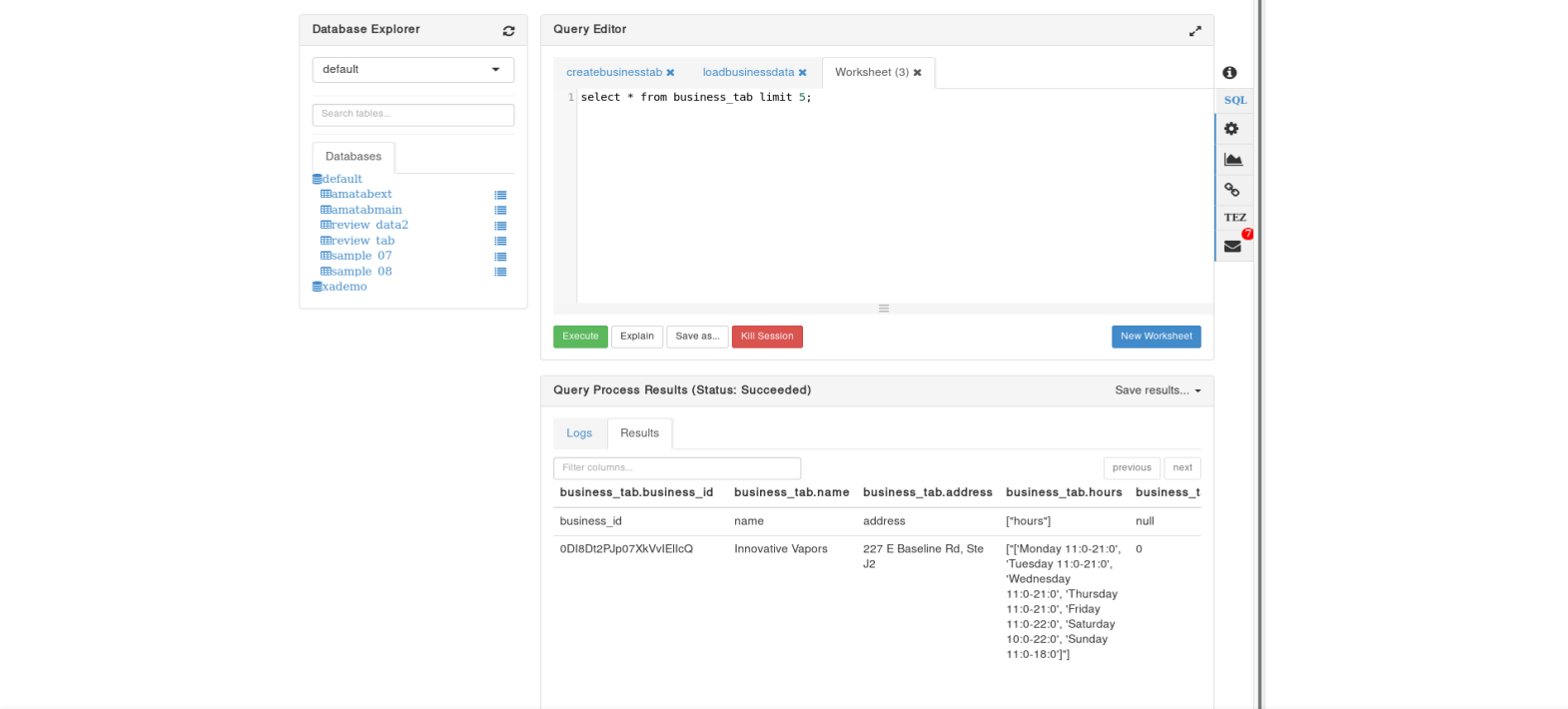
* Created hive table for business.tsv:



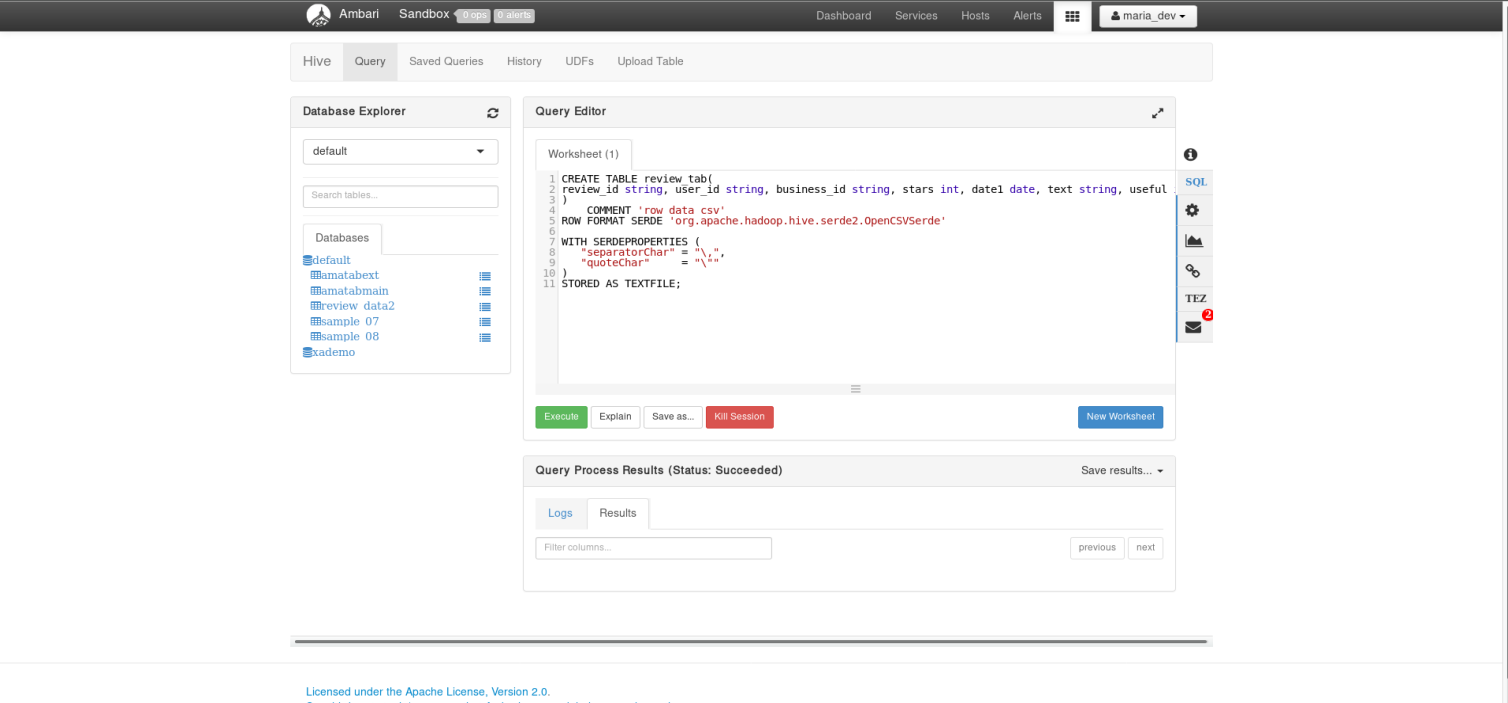
* Loading business table:



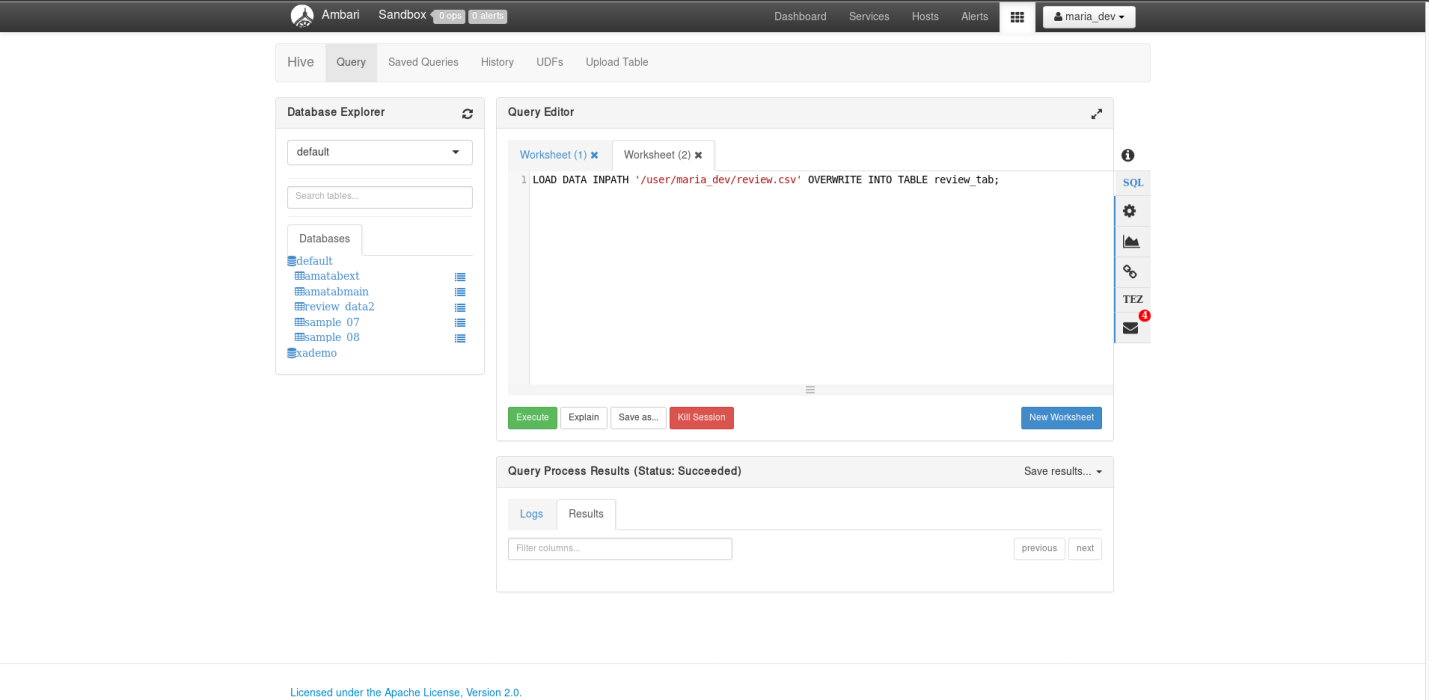
* Verifying the data from business table:



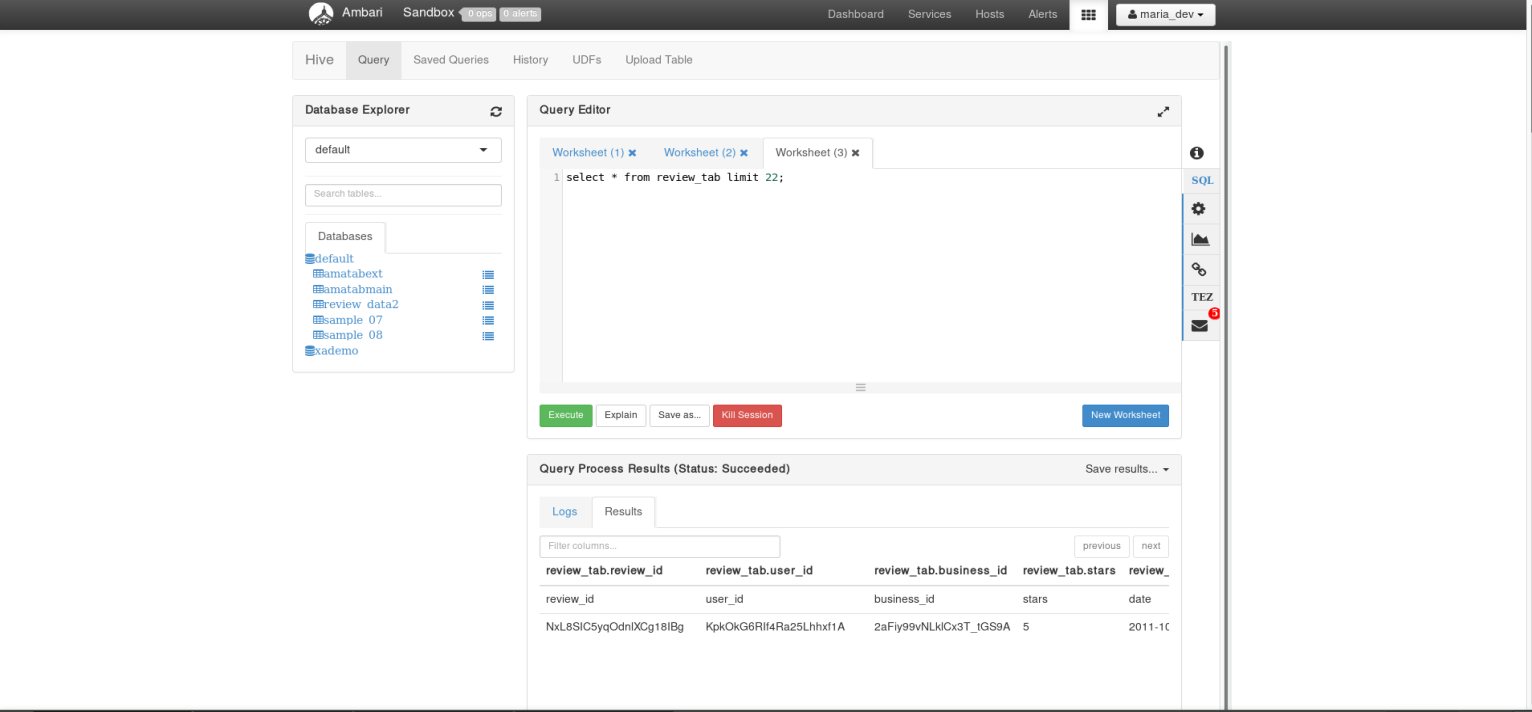
* Created hive table for review.csv:

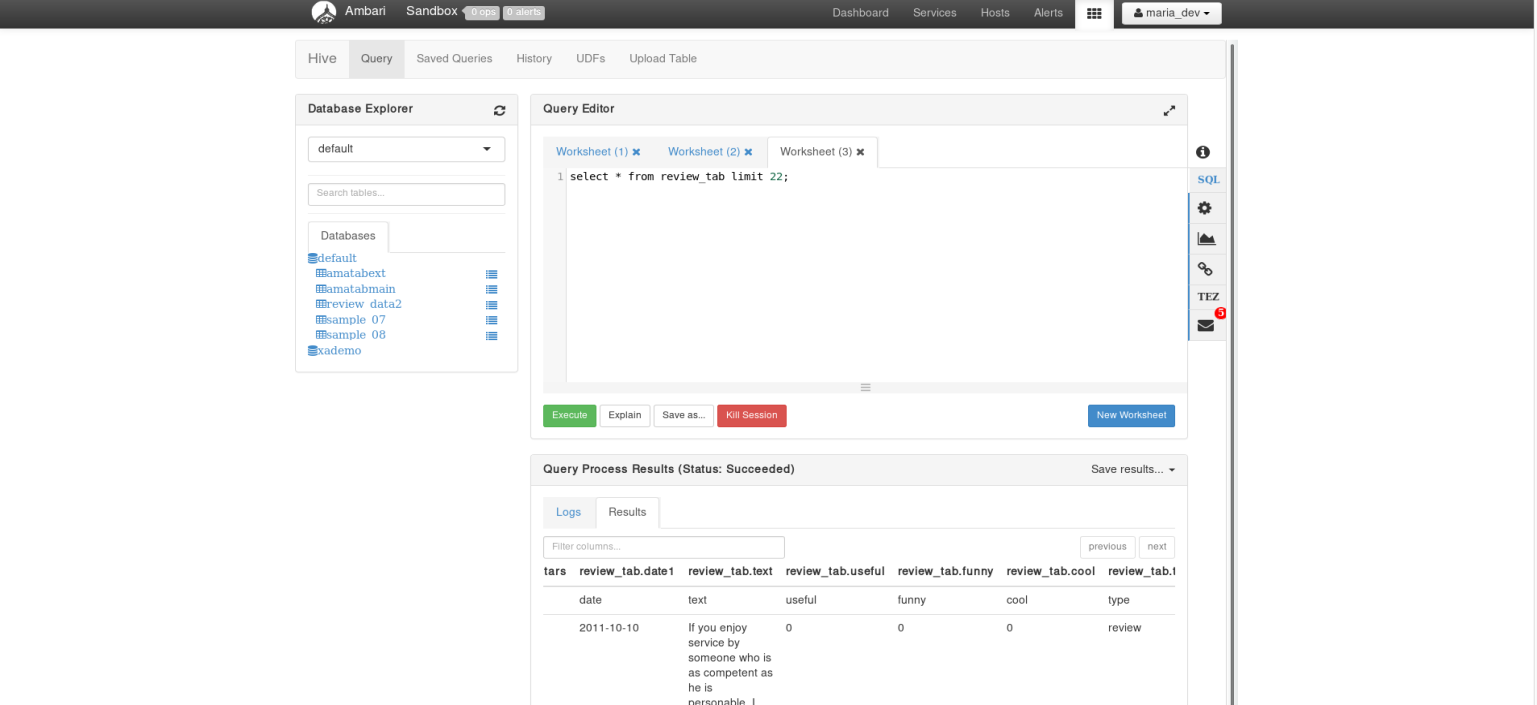


* Loading review table:

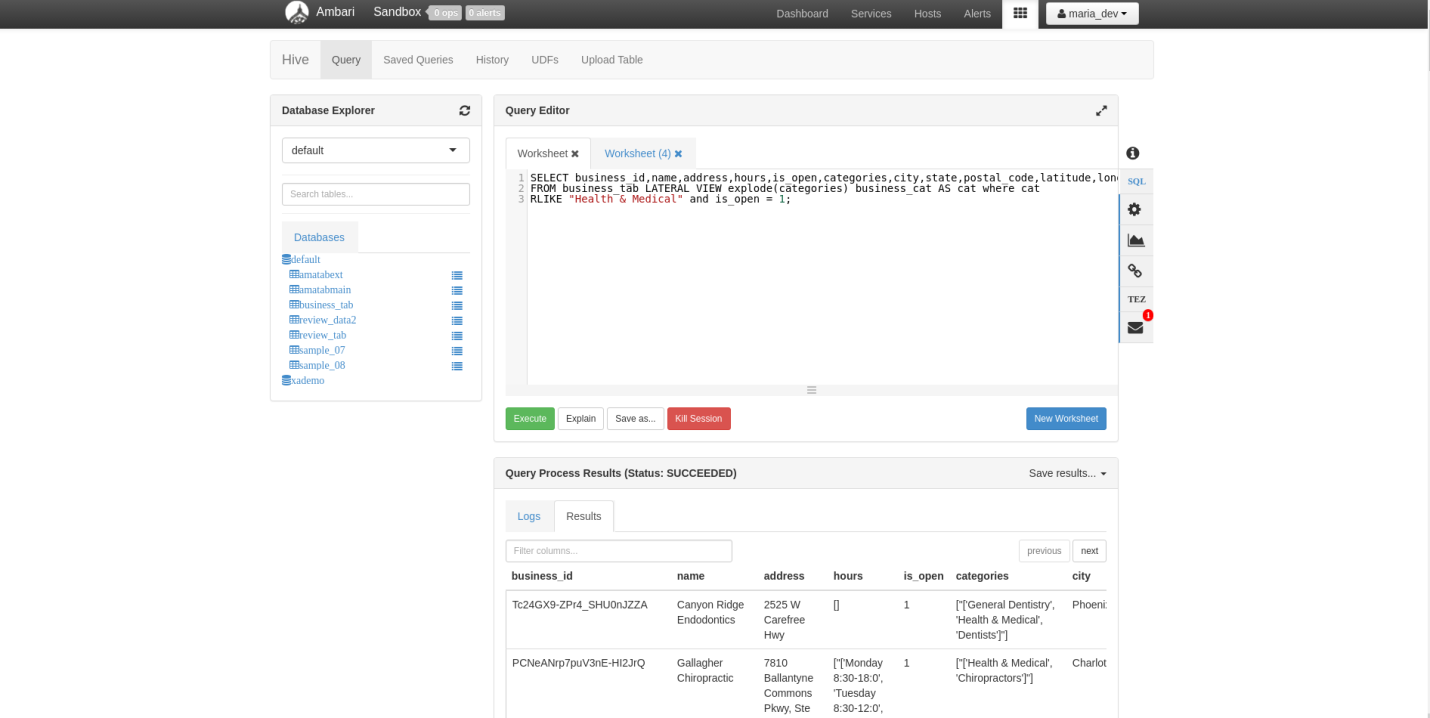


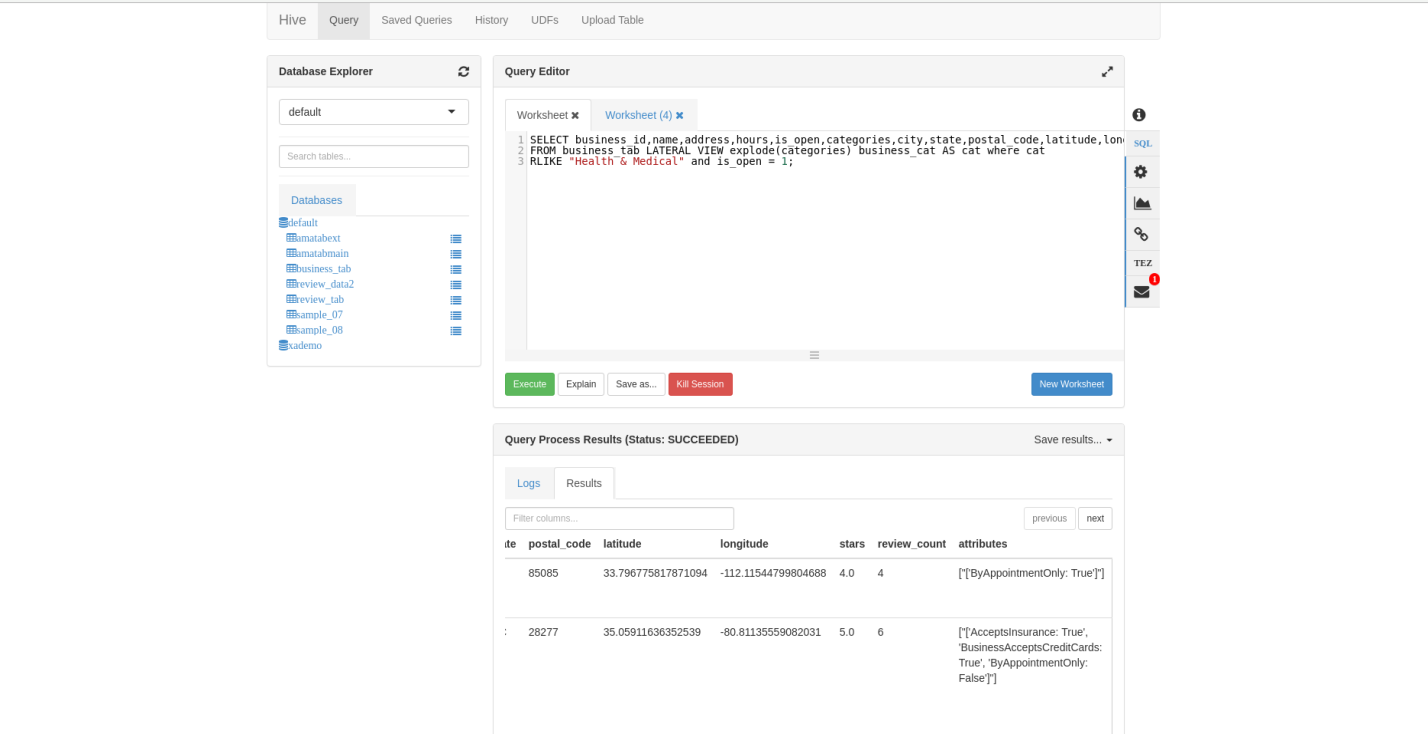
* Verifying the data from review table:

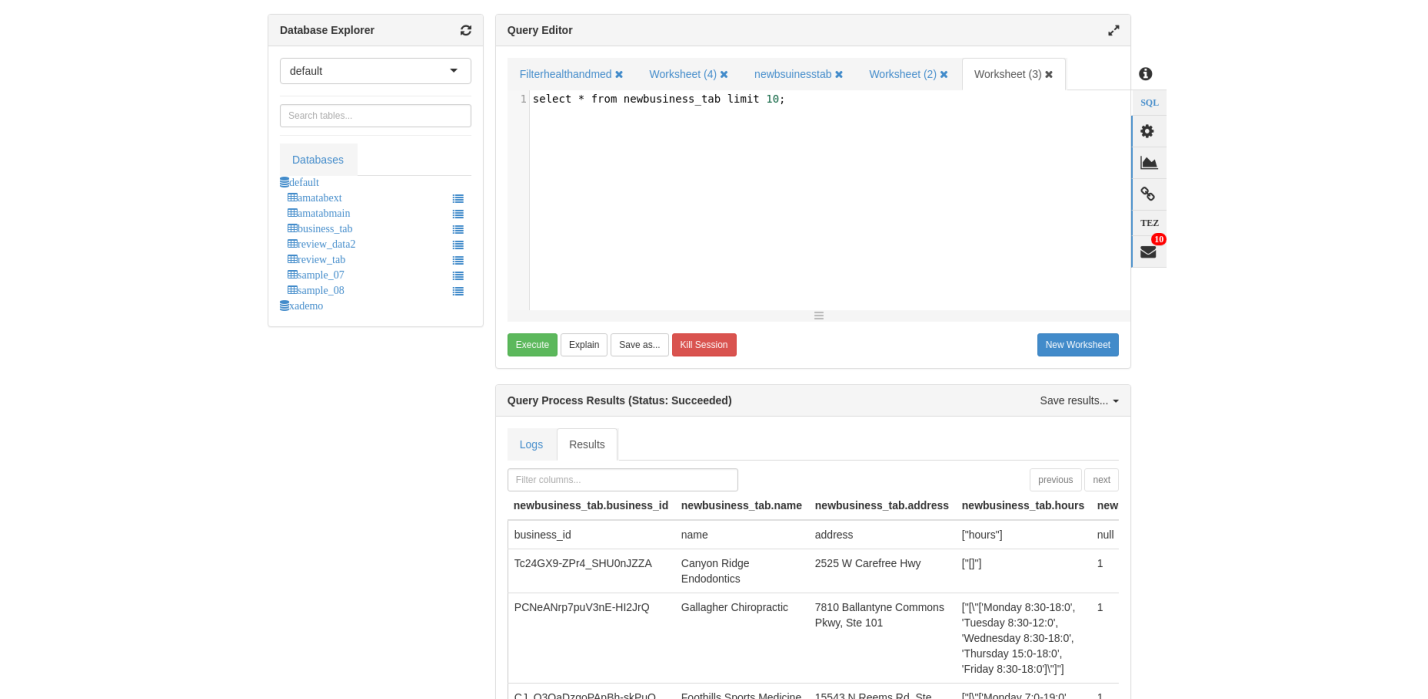




**3 Above table is saved in csv format and Lateral view of table with single categories are derived.**

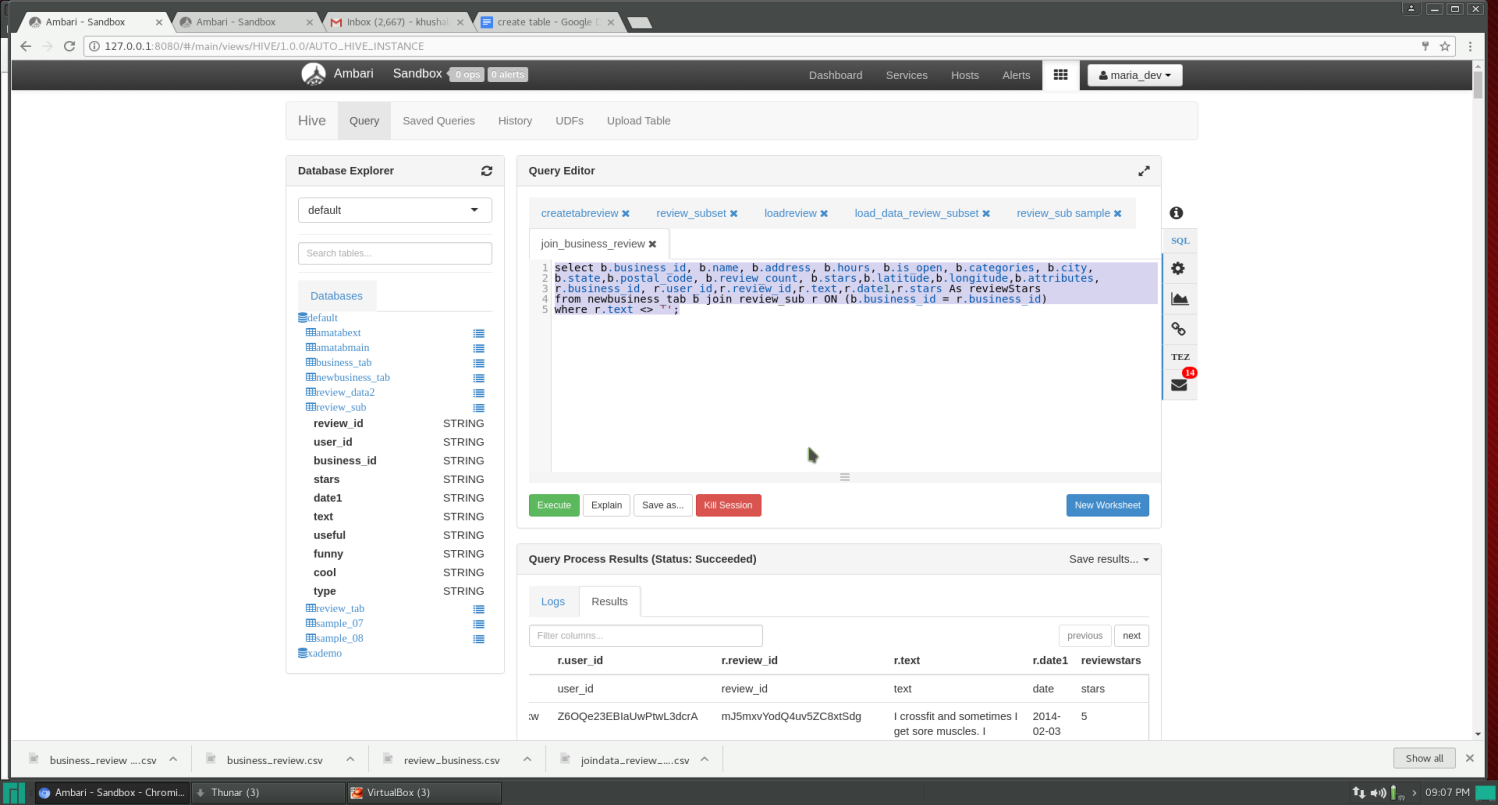




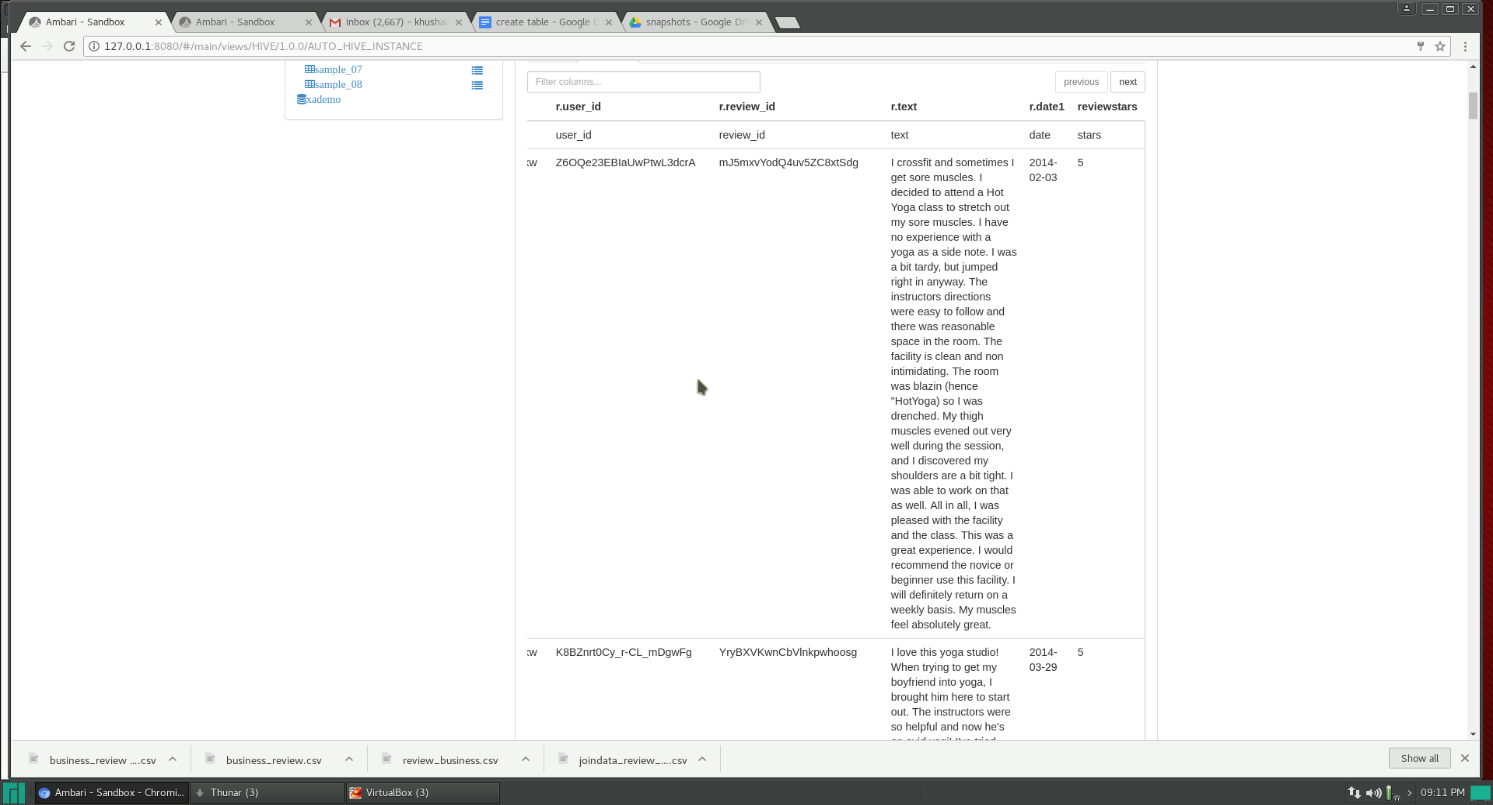


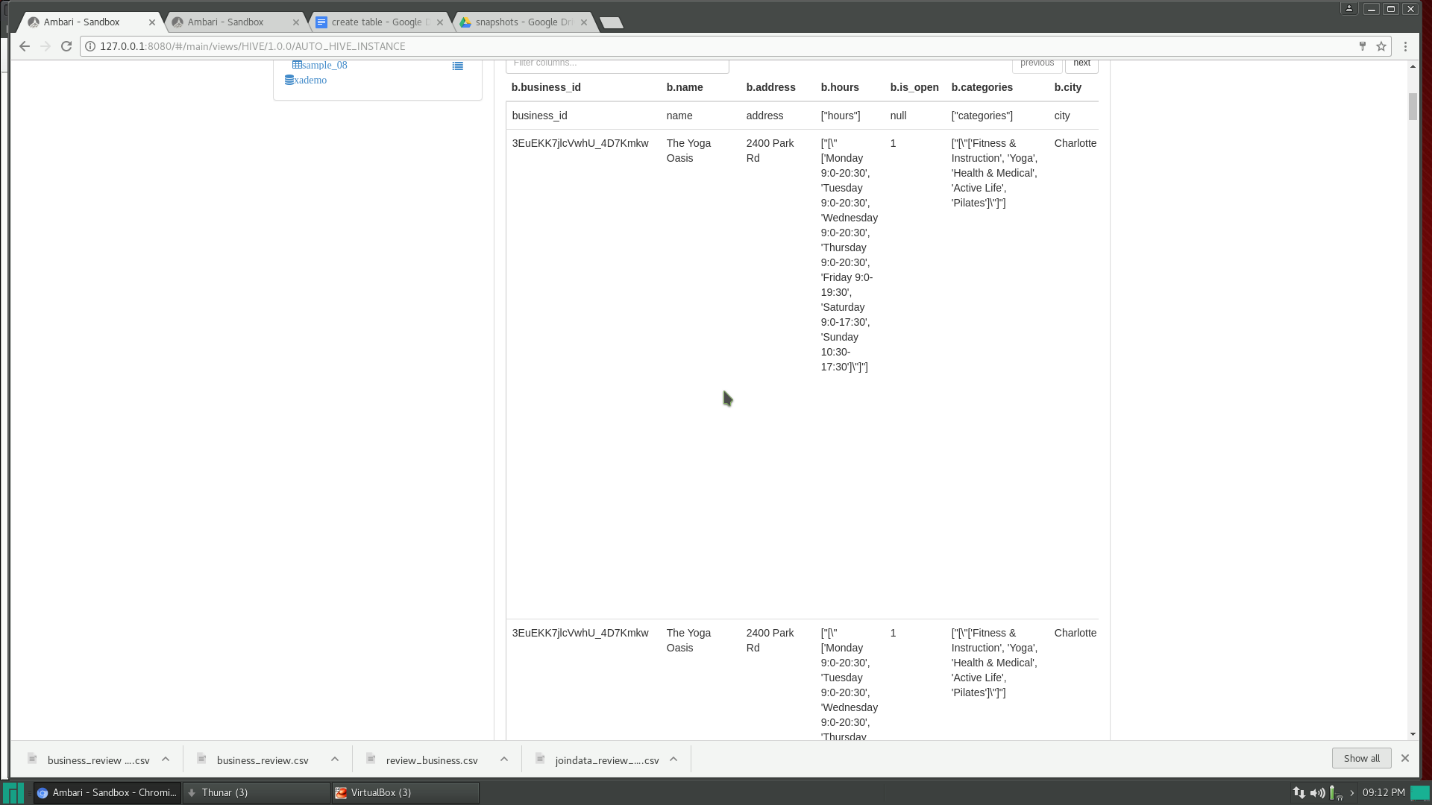
**4. Integrated business and review tables and storing the results in CSV file format.**

* Integration of business and review tables.

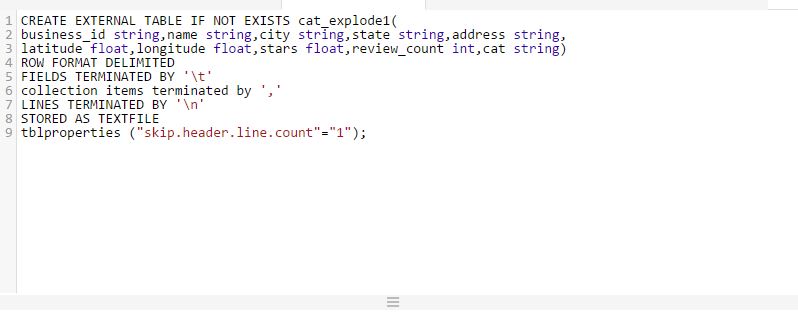


* Verifying the results of business

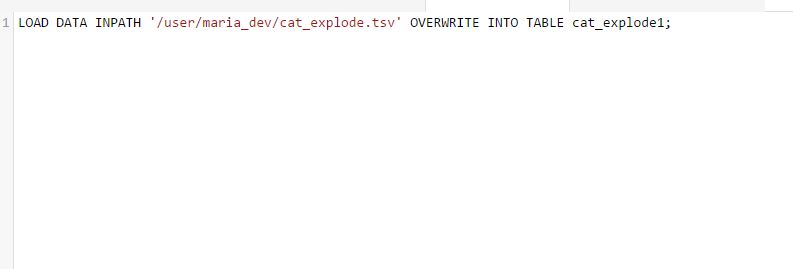




* Performed Text mining on the merged file. For text mining a file with positive and negative words is used. The python script takes single record from the merged file and then calculates number of positive and negative word on a single text review. Based on the results, if the numbers of positive words are greater than negative words then the review text is considered as positive otherwise negative. If both results are equal in number ‘NA’ is considered. Later the results are appended with the merged results of review and business.
* The resulted tsv is again uploaded on hive.
* Creating table cat\_explode1(category explode)

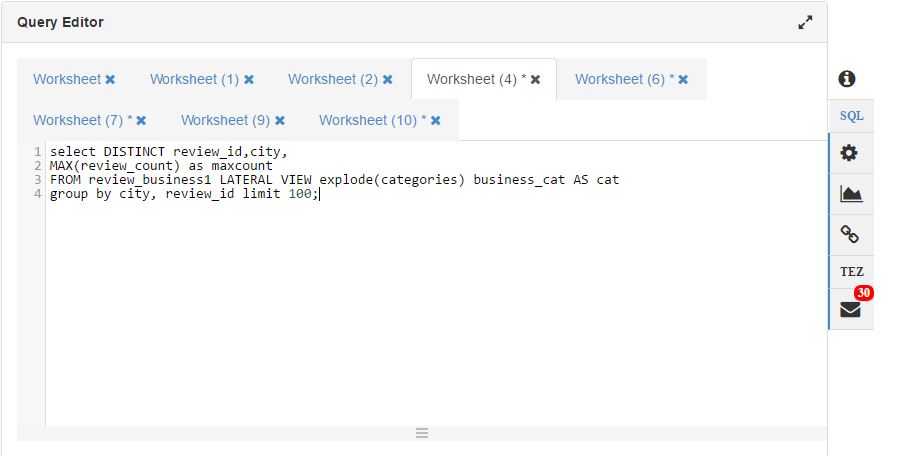


* Loading data to cat\_explode1



* Maximum review\_count of health center is derived based on specific cities. For this we need to use group by city. But there exists several reviews for a single health center. For separating this we need to use distinct review\_id.

Failure is that in hive we cannot use DISTINCT with GROUP BY without using an aggregate function with DISTINCT.

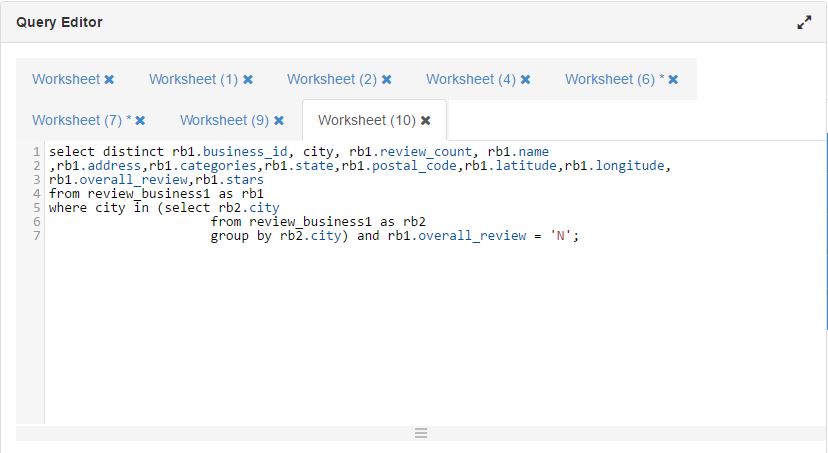


* Success- For overcoming this, a sub query was created that selects distinct business\_id and the other necessary fields. Then within this query another sub query for grouping it according to a specific city was formed. Since the main idea is coming up with the best doctor in a specific city, only the positive reviews was considered.



* For carrying out sentimental analysis, one more table for negative\_reviews was created

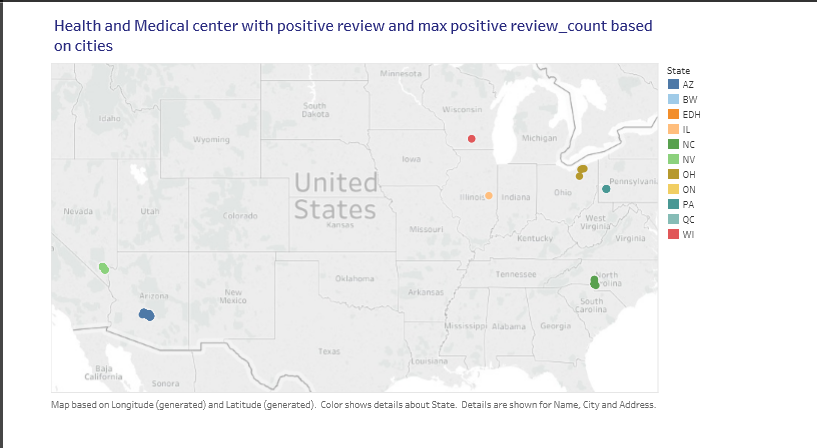
Creating separate table for negative\_review:



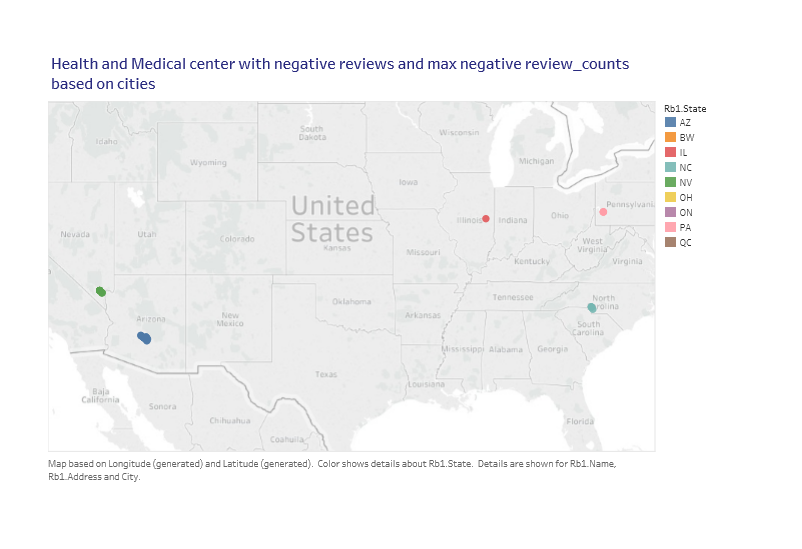
* Final files along with the max number of review\_counts for specific categories of Health and Medical like Skin Care, Dentist, Chiropractors, Yoga, Obstetricians & Gynecologist and Family Practice will be shown using tableau.

**5. Visualization Using Tableau**

* Best Health Center based on Positive Reviews for a specific city.

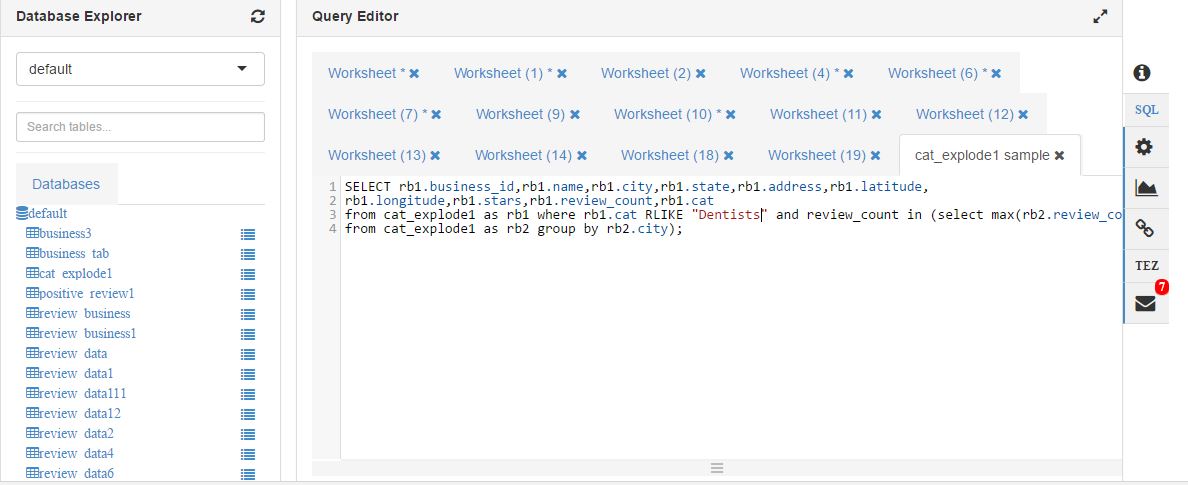


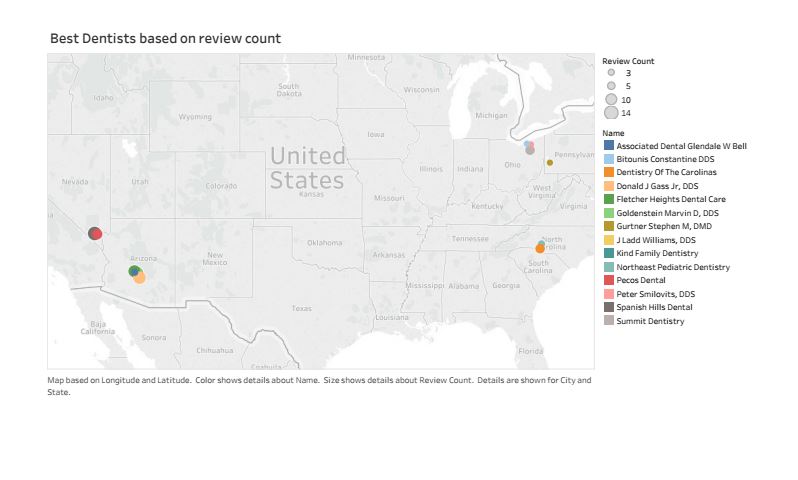
7.2. Poor Health Center based on Negative Reviews for a specific city.



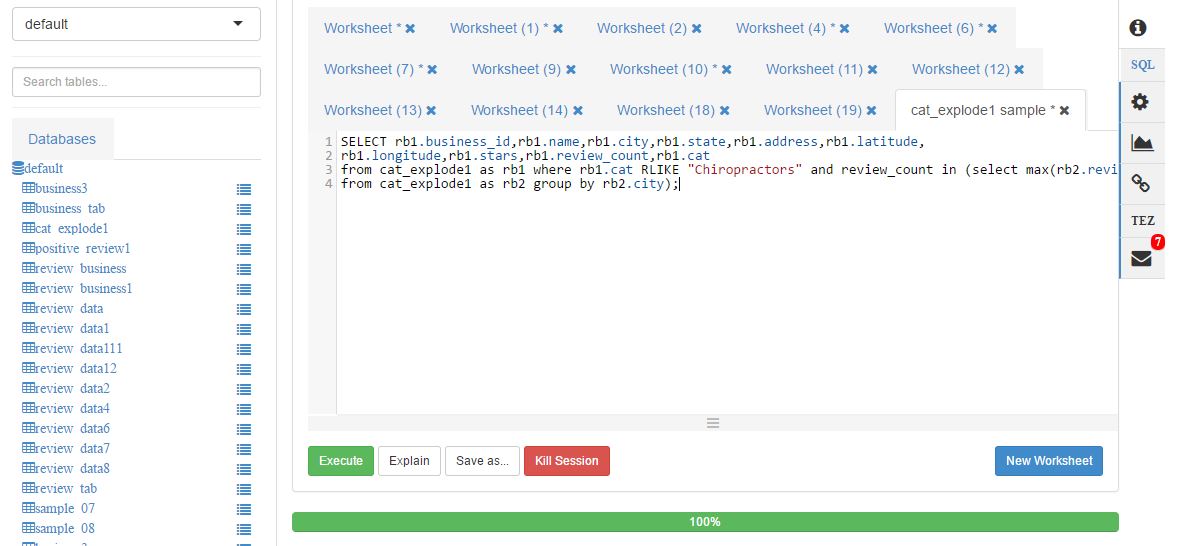
7.3. Best Health Center on basis of review\_count pertaining to Specializations for cities are plotted

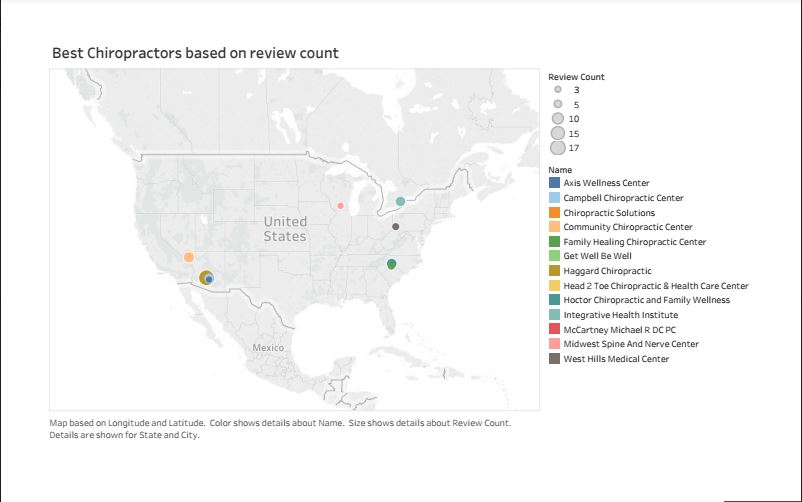
**Specialization – Dentists**



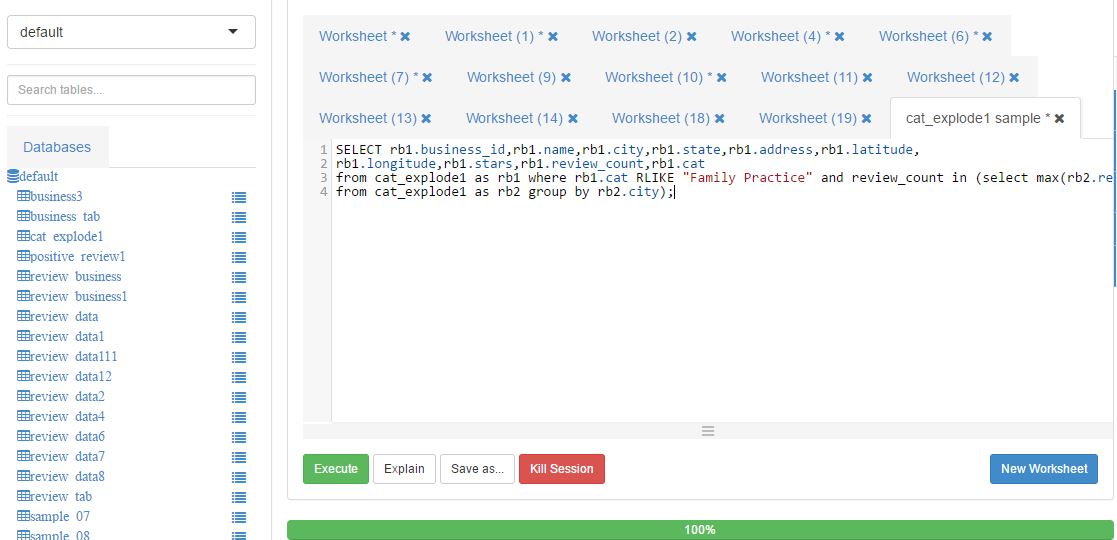


**Specialization – Chiropractors**



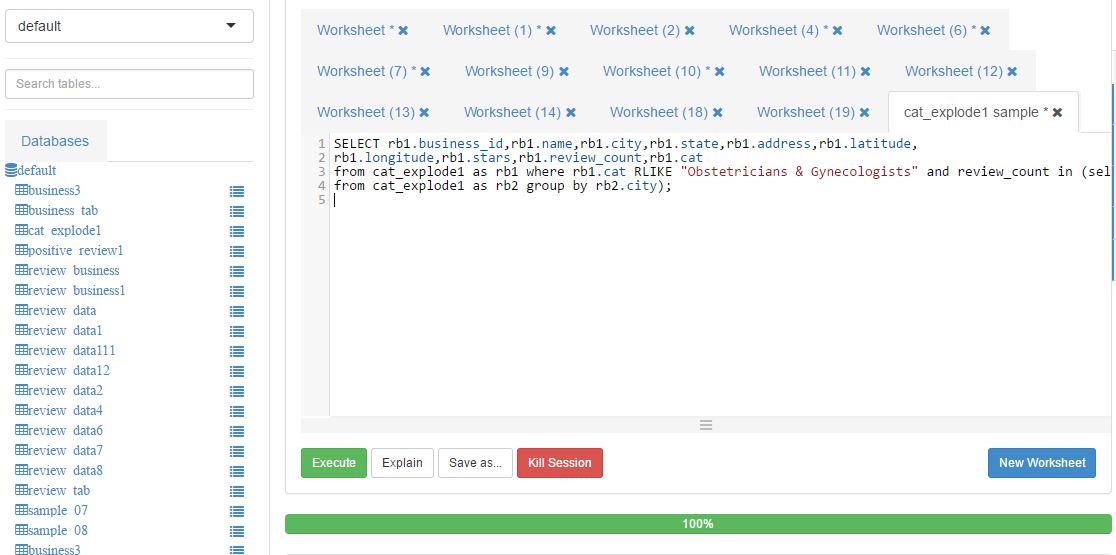


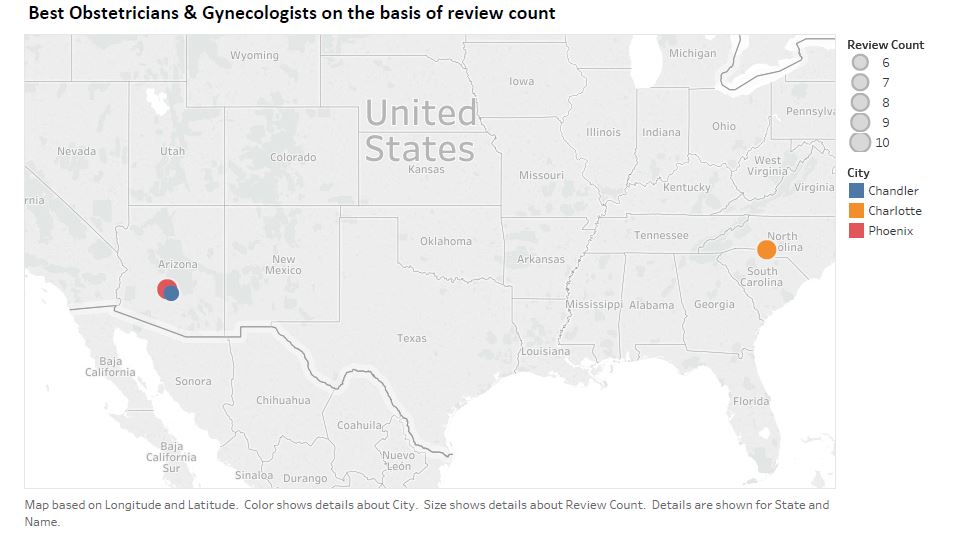
**Specialization– Family Practice**

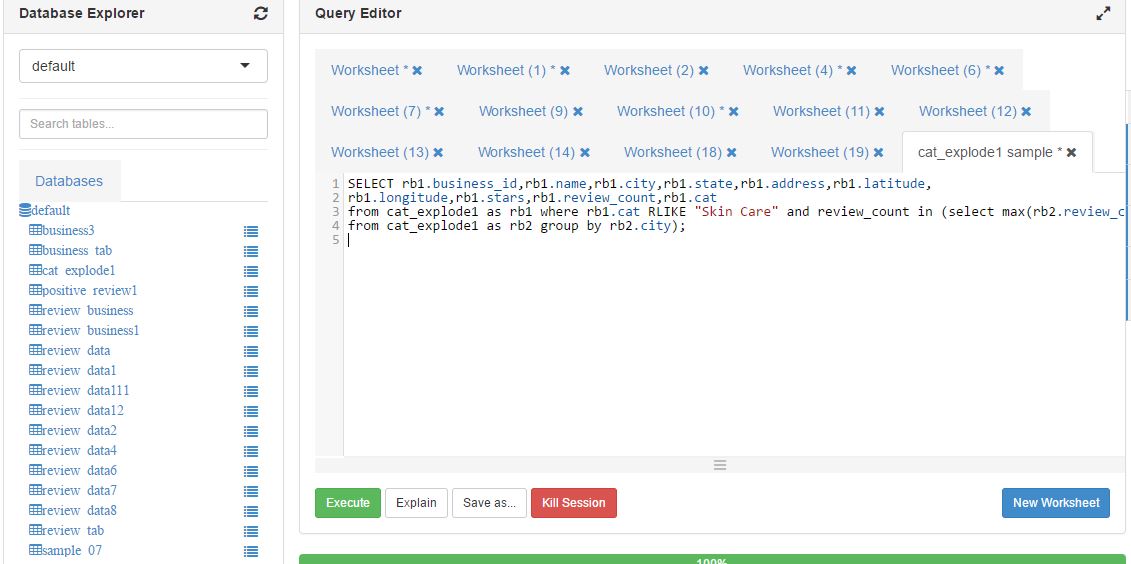


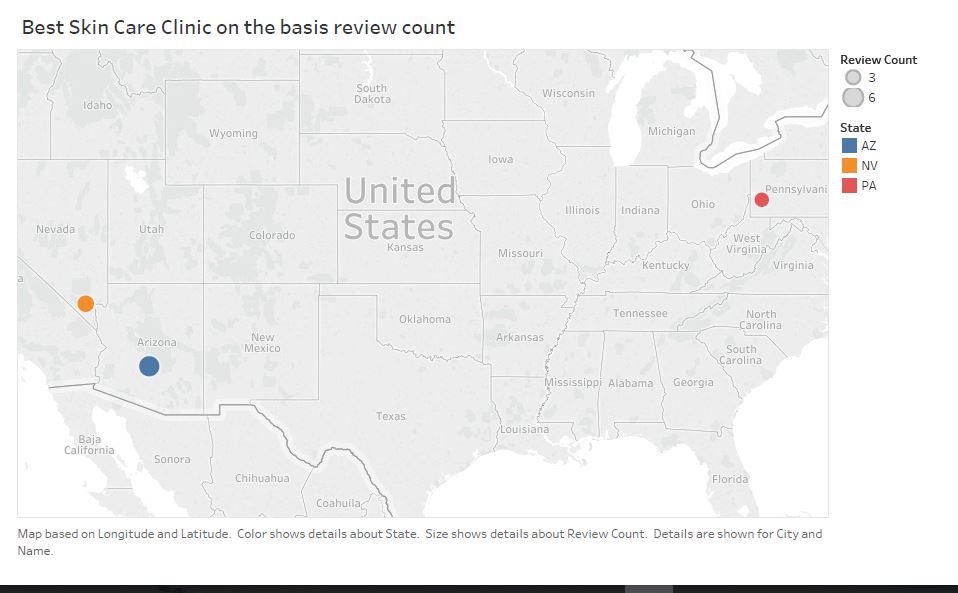


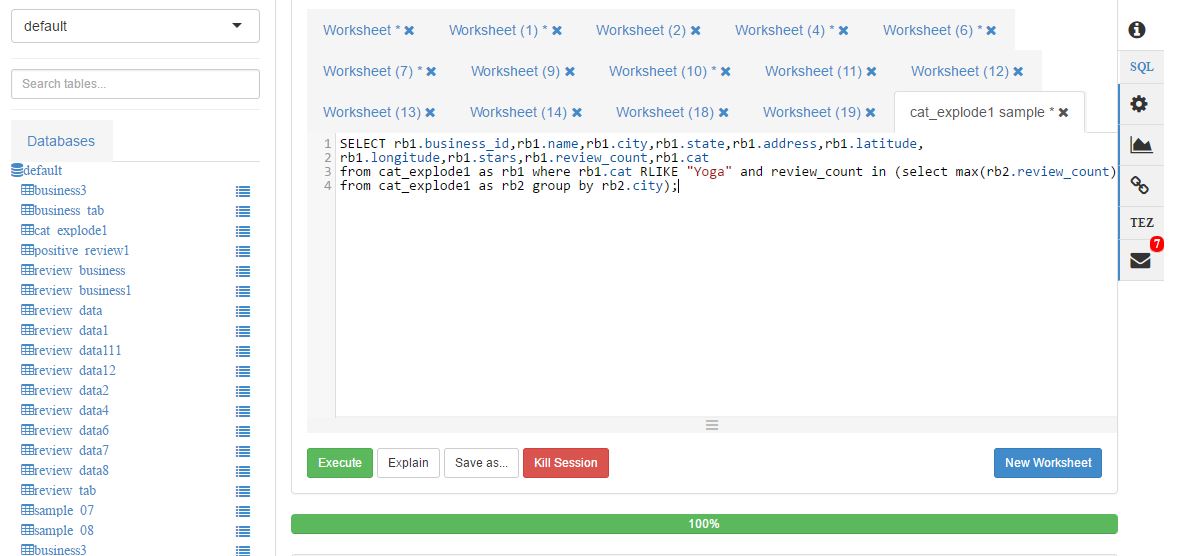
**Specialization – Obstetricians & Gynecologists**





**Specialization – Skin Care**



**Specialization - Yoga**



**Failures**

1. Extracted downloaded dataset to obtain JSON files and uploaded it to ambari.



Pic-review.json

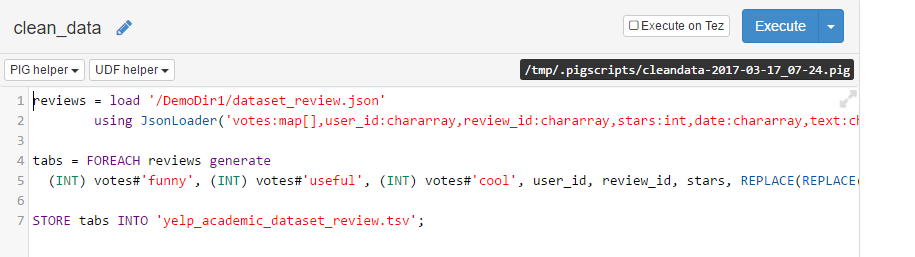
**2. Failures in Converting dataset to readable formats using python and pig script.**

* 1. Initially business.csv was successfully created using python Script. Since business.csv file was having complex data fields it was a failure to create hive tables from generated business.csv file.

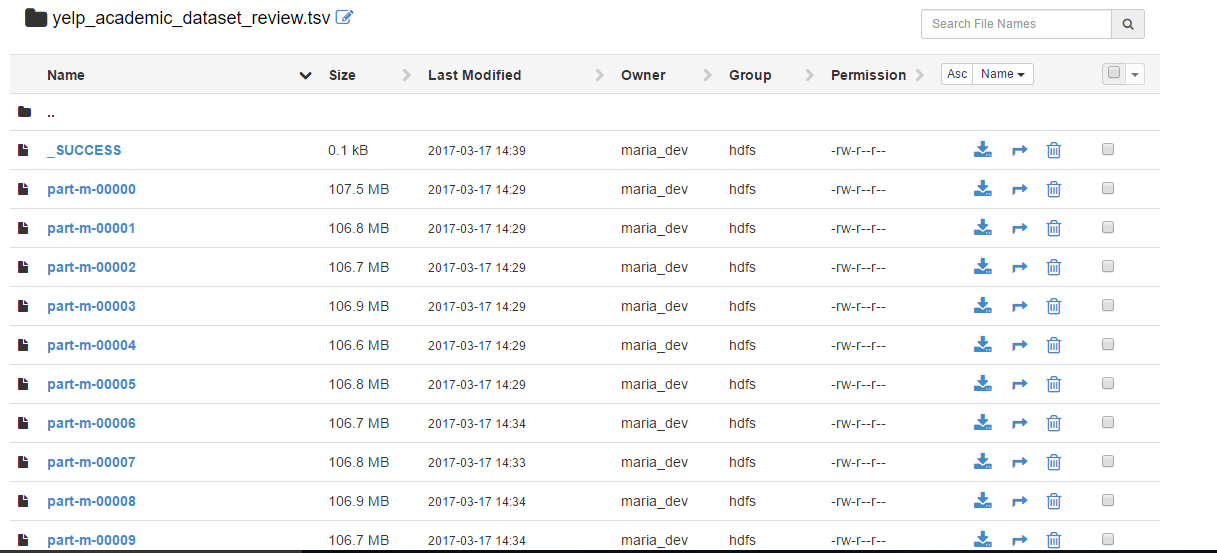
Pic-python script and failue with table creation

* 1. Initially faced problem in converting review.json to review.csv. Later followed the approach of using pig script to convert review.json to review.tsv.

Pig Script Used:



Output of Pig Script:



* 1. After careful analysis of review.tsv it was figured out that business\_id field was missing. This business\_id field was required to join review and business tables.

