Data Insights Using Apache Hive

STEPS (COMMANDS) TO SETUP THE ENVIRONMENT

1. Command To Create Hive Database (createDatabase.hql)

2. Command To Create Temp Table (createTempTable.hql)

   Note: A temporary table is created to load and store the CSV data as TEXTFILE from the S3 bucket location. The first line of the file is skipped, which is metadata.

3. Command To Create MainTable (createTable.hql)

   Note: The main table is created to store data in ORC format with SNAPPY compression algorithm so that it enables us to retrieve data faster than data stored in a table with text format. This table is Partitioned By Months Of The Year. The month information is extracted from the field **issue_date.**

4. Command To Load Data In Main Table (insertDataInTable.hql)

   Note: The data from the temporary table ***parking_violation_data_temp*** *is transfered to the main table* ***parking_violation_data*** *using an INSERT query. In order to work only on data pertaining to 2017, a filter condition is created to make sure only data pertaining to 2017 is transfered to the main table which will be used for all queries.*

Part I – Examine The Data

1. Find the total number of tickets for the year.

   Query (queryForAnalysis1.hql)

2. Find out how many unique states the cars which got parking tickets came from

   Query (queryForAnalysis2.hql)

3. Some parking tickets don’t have addresses on them, which is cause for concern. Find out how many such tickets there are(i.e. tickets where either "Street Code 1" or "Street Code 2" or "Street Code 3" is empty)

   Query (queryForAnalysis3.hql)

Part II – Aggregation Tasks

1. How often does each violation code occur? (frequency of violation codes - find the top 5)

   Query (queryForAnalysis1.hql)

2. How often does each vehicle body type get a parking ticket? How about the vehicle make? (find the top 5 for both)

   Query 1 For Vehicle Body Type (queryForAnalysis2_1.hql)

   Query 2 For Vehicle Make (queryForAnalysis2_2.hql)

3. A precinct is a police station that has a certain zone of the city under its command. Find the (5 highest) frequencies of:
   i. Violating Precincts (this is the precinct of the zone where the violation occurred)
   ii. Issuer Precincts (this is the precinct that issued the ticket)

   Query 1 For Violating Precincts (queryForAnalysis3_1.hql)

   Query 2 For Issuer Precincts (queryForAnalysis3_2.hql)

4. Find the violation code frequency across 3 precincts which have issued the most number of tickets - do these precinct zones have an exceptionally high frequency of certain violation codes?

   Query (queryForAnalysis4.hql)

5. Find out the properties of parking violations across different times of the day: The Violation Time field is specified in a strange format. Find a way to make this into a time attribute that you can use to divide into groups

   Query (queryForAnalysis5.hql)

6. Divide 24 hours into 6 equal discrete bins of time. The intervals you choose are at your discretion. For each of these groups, find the 3 most commonly occurring violations

   Query (queryForAnalysis6.hql)

7. Now, try another direction. For the 3 most commonly occurring violation codes, find the most common times of day (in terms of the bins from the previous part)

   Query (queryForAnalysis7.hql)

8. Let’s try and find some seasonality in this data. First, divide the year into some number of seasons, and find frequencies of tickets for each season. (Hint: A quick Google search reveals the following seasons in NYC: Spring(March, April, March); Summer(June, July, August); Fall(September, October, November); Winter(December, January, February)). Then, find the 3 most common violations for each of these seasons.

   Query (queryForAnalysis8.hql)