**Hive - Task 3**

**Hive:**

Hive gives a SQL-like interface to manage data. It is used in Amazon H3 file system. It also support HiveQL. Hive implicitly codeword hiveQL statement into Adirected acyclic graph of mapreduce those the spark jobs.

It support Advance features such as indices, partition, buckets, asummetric transactions, custom user defined functions, joins, sampling & many others. Lots of them take a considerable amount of time if implemented manually.

**Create directory in Hadoop:**

hdfs dfs –mkdir /expedia

The above query will create directory with name expedia in hdfss.

**Copy file to Hadoop File System:**

hdfs dfs –put train.csv /expedia/

This query will print train.csv in expedia directory in hdfs.

**Create Database in hive:**

CREATE DATABASE IF NOT EXISTS expedia;

This query will create database with expedia. If database is already exists than this query will not do any change.

**Create table in hive:**

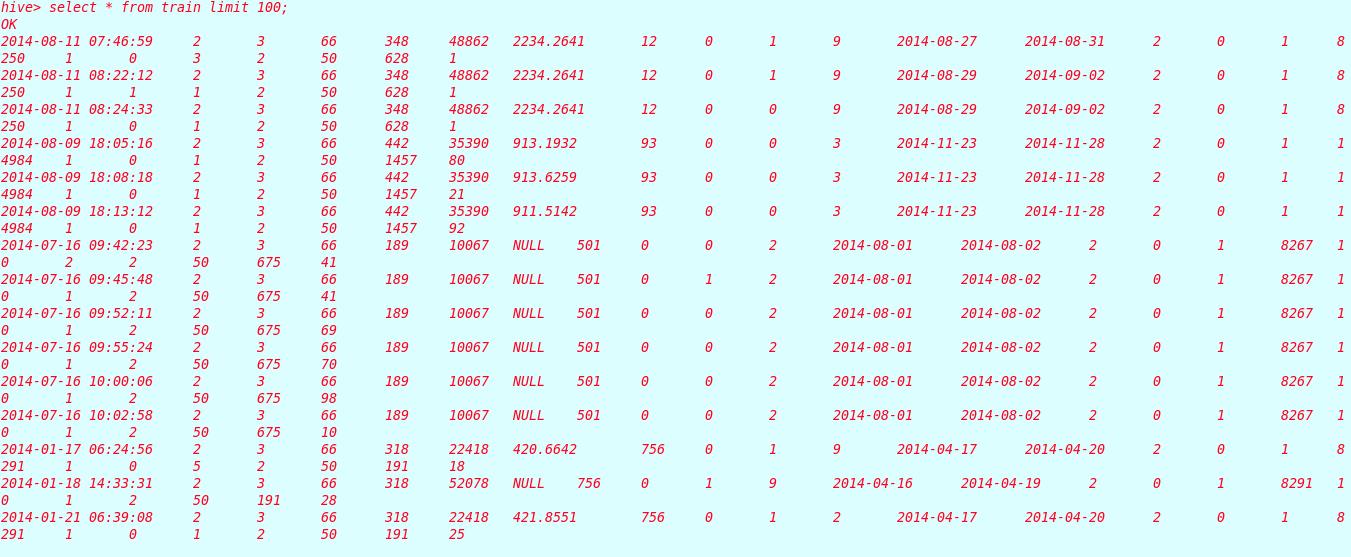
CREATE TABLE IF NOT EXIST train (date\_time string, site\_name int, posa\_continent int, user\_location\_country int, user\_location\_region int, user\_location\_city int, orig\_destination\_distance double, user\_id int, is\_mobile tinyint, is\_package int, channel int, srch\_ci string, srch\_co string, srch\_adults\_cnt int, srch\_children\_cnt int, srch\_rm\_cnt int, srch\_destination\_id int, srch\_destination\_type\_id int, is\_booking tinyint, cnt bigint, hotel\_continent int, hotel\_country int, hotel\_market int, hotel\_cluster int) ROW FORMATE DELIMITED FIELDS TERMINATED BY "," STORED AS TEXTFILE LOCATION "/expedia/"

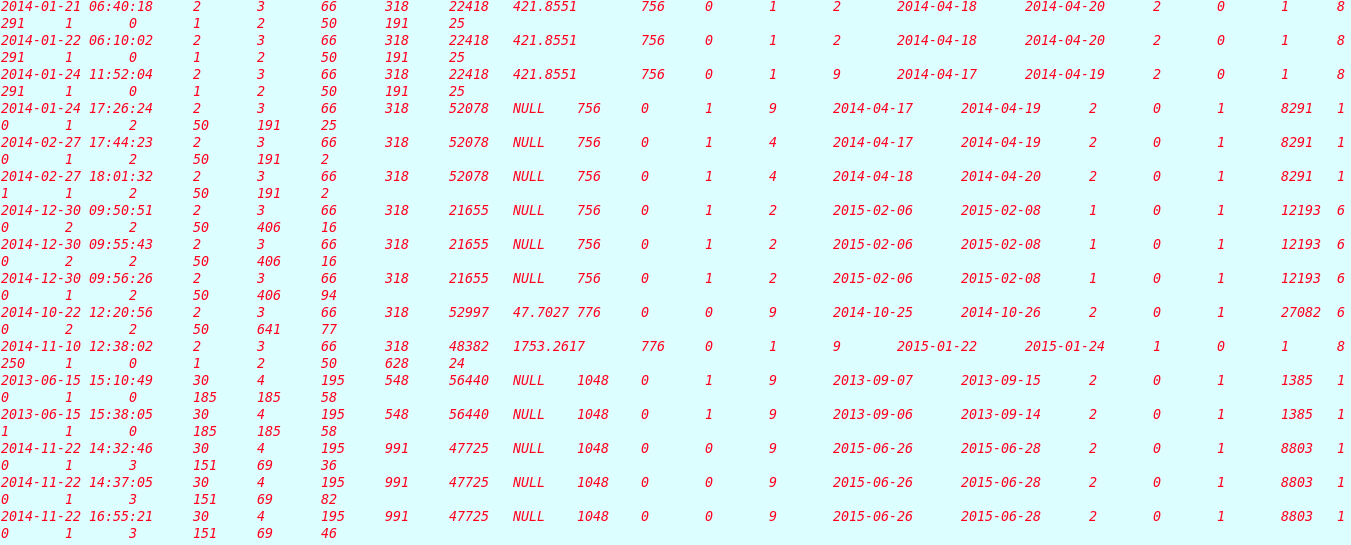
In above query, I have given the hdfs location of train.csv which is “/expedia/”. So, this table is form over it.

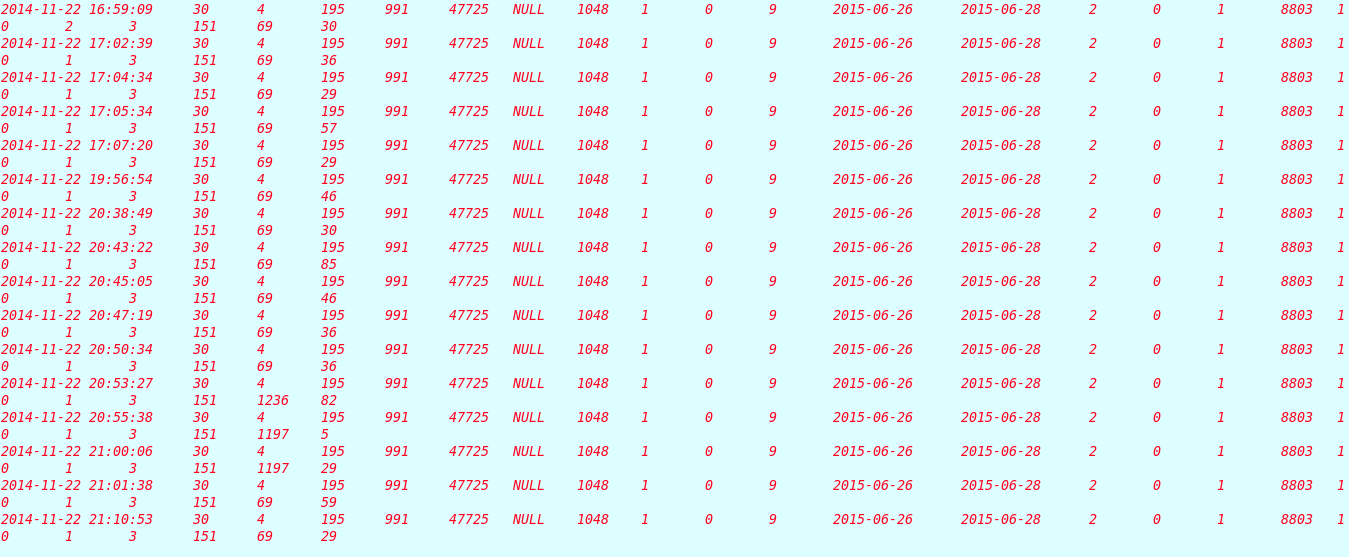
**Displaying first 100 rows**

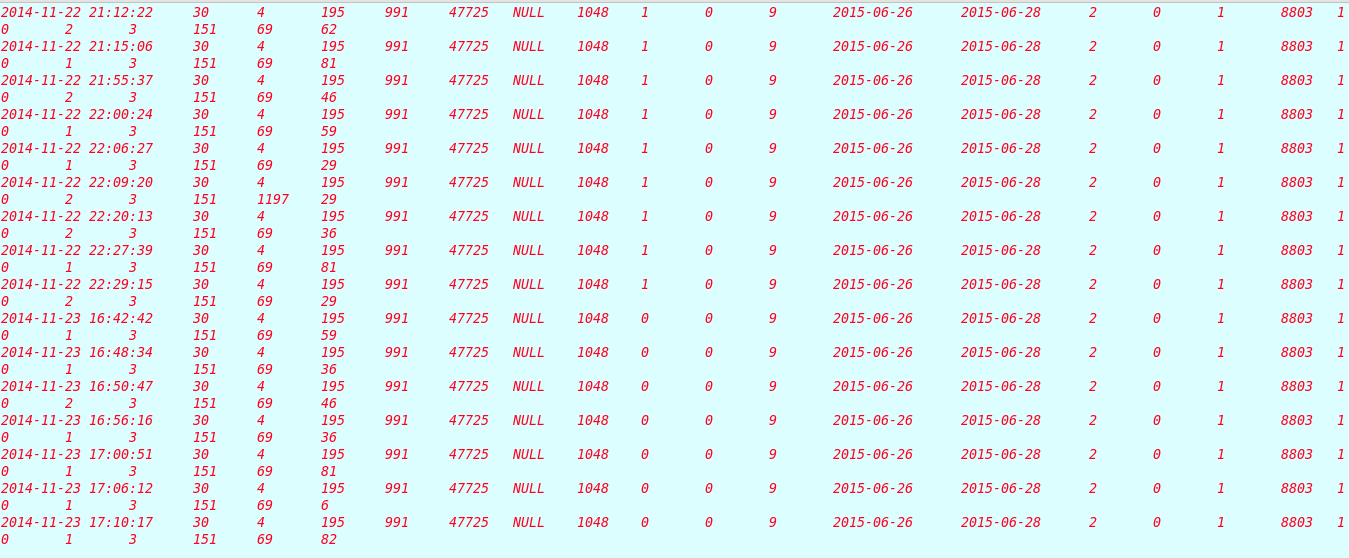
You can see here first 100 rows which are same as actual train.csv file.

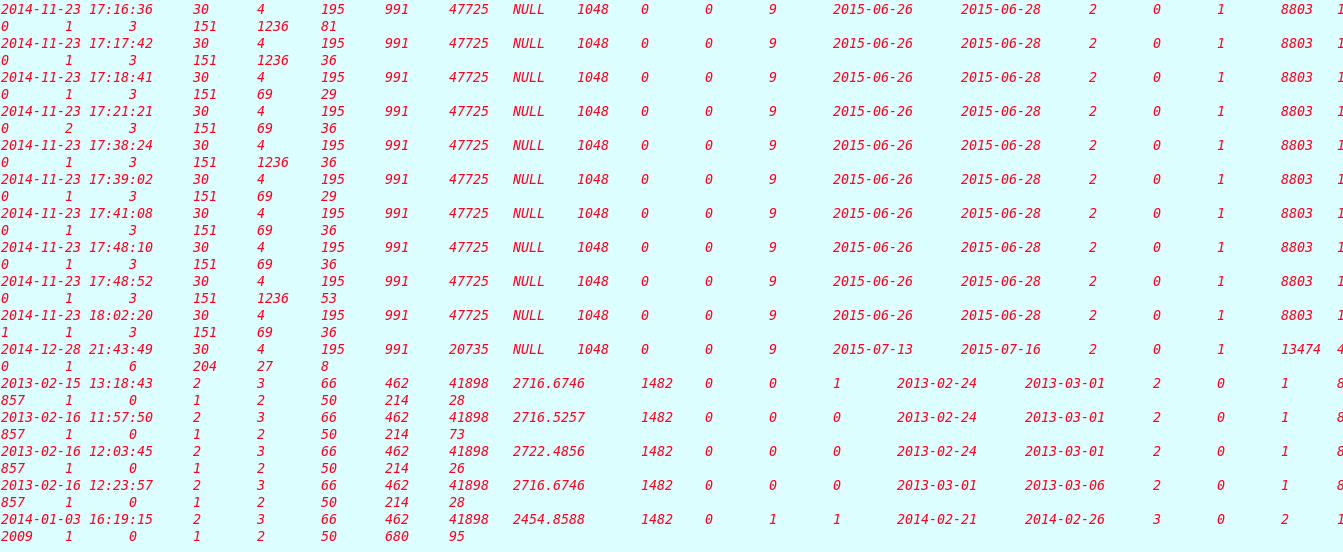
SELECT \* from train LIMIT 100;

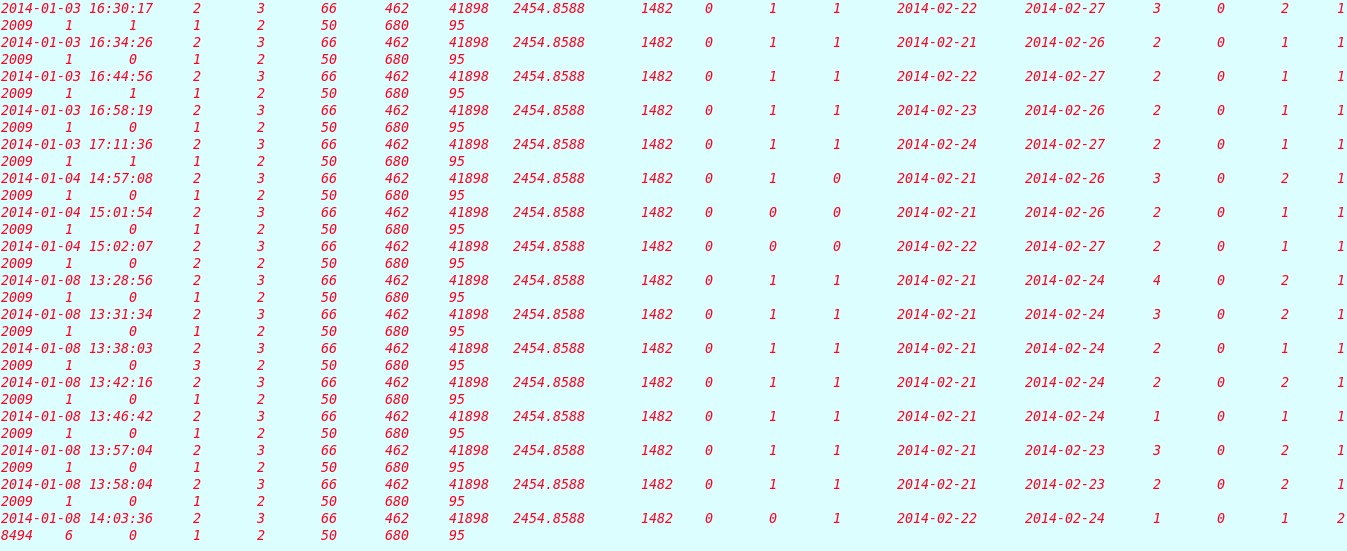


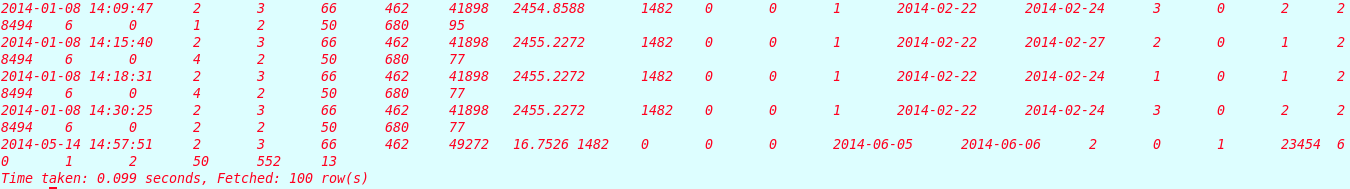










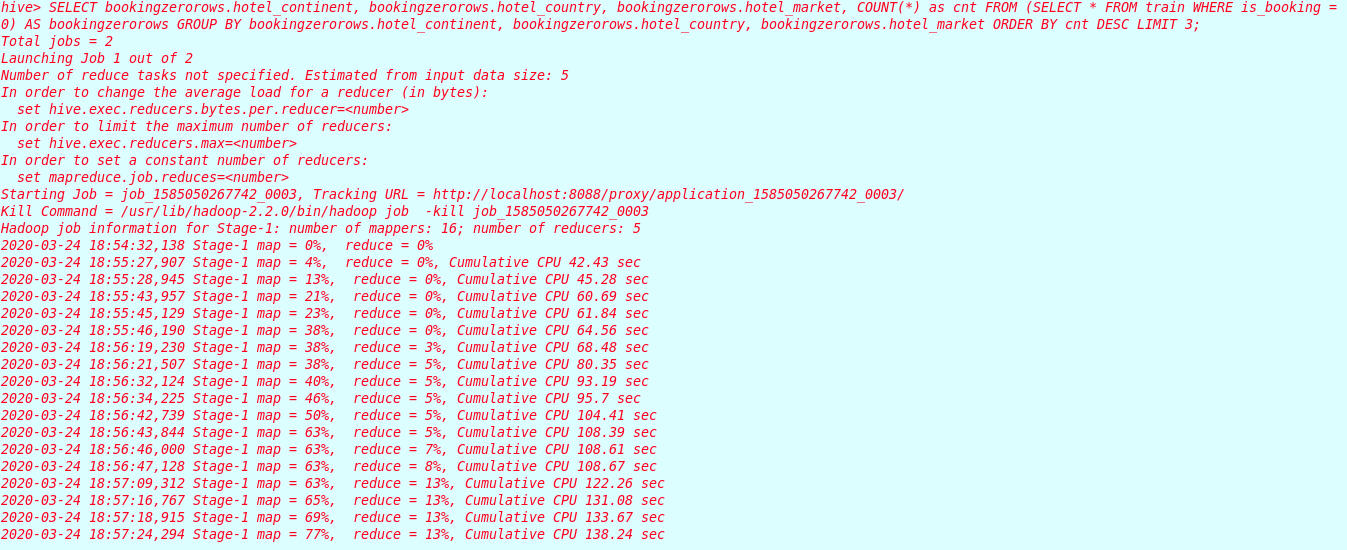


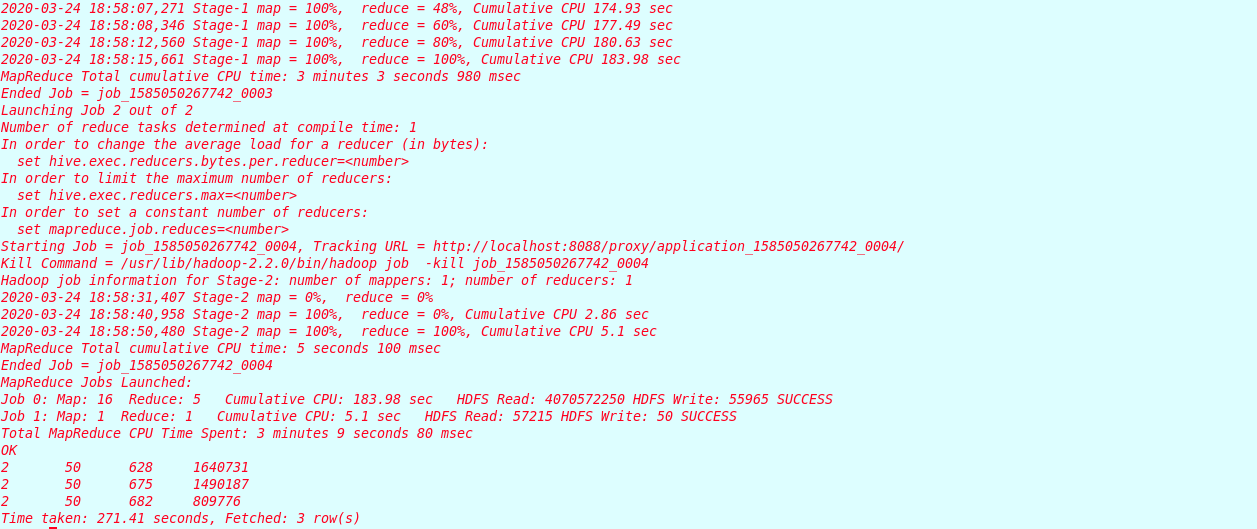
**Task 3**

**Write hive script to calculate top 3 most popular hotels (treat hotel as composite key of continent, country and market) which were not booked. Most popular means most searches in dataset. Not booked means column booked = 0, make screenshots before and after script execution, copy hive script to archive.**

SELECT bookingzerorows.hotel\_continent, bookingzerorows.hotel\_country, bookingzerorows.hotel\_market, COUNT(\*) AS cnt FROM (SELECT \* FROM train WHERE is\_booking = 0) AS bookingzerorows GROUP BY bookingzerorows.hotel\_continent, bookingzerorows.hotel\_country, bookingzerorows.hotel\_market ORDER BY cnt DESC LIMIT 3;

In the above query, first I have filtered out all those records whose booking status is 0. Then group by all those records on the basis of continent, country & market.





**Query Result:**

|  |  |  |  |
| --- | --- | --- | --- |
| Continent | Country | Market | Count |
| 2 | 50 | 628 | 1640731 |
| 2 | 50 | 675 | 1490187 |
| 2 | 50 | 682 | 809776 |