#1:

Firm size: small to mid-cap size based on the amount of data collected.

Sector: Ecommerce business

Competitors: Amazon, eBay, Alibaba, JD.com, AliExpress, Wish, Shopify, and Coupang

Technology: Hadoop Ecosystem, Spark, GCP, Excel, and OpenRefine Human

Capital:

- Requires 1-2 personnel
- Several skills are required like data cleaner, data miner, and data visualizer.
- Training will take 5 weeks.

Technologies Deployed:

- Google Cloud Platform
- Hive2 and Hive Meta store
- Spark
- HQL
- HDFS system
- YARN
- OpenRefine
- Excel

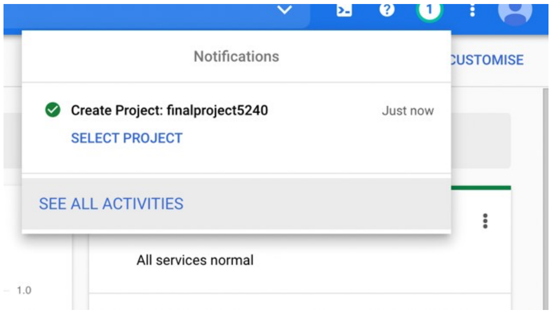
#2.1:

Action: Removed the in-between white space with excel by manually going to each column cells and deleting them.

#2.2:

Action: Removed the commas on each cell of "Product Names" by using OpenRefine's "replace" function which lets you replace a character for another, and I replaced "," with a blank argument.

#3:

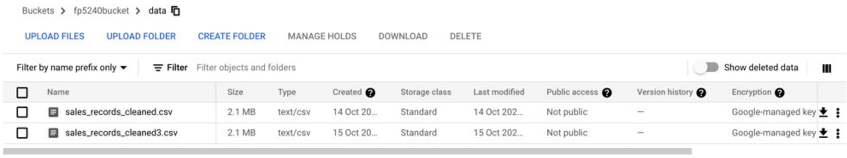


Notifications

Create Project: finalproject5240 Just now

All services normal

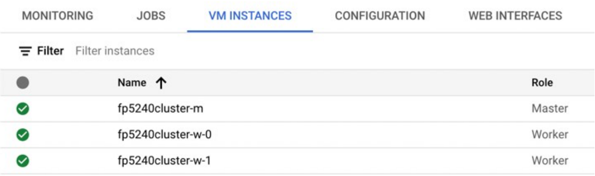
Figure 1: CREATE NEW PROJECT



Buckets > fp5240bucket > data

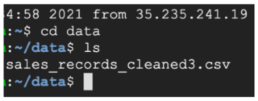
□	Name	Size	Type	Created	Storage class	Last modified	Public access	Version history	Encryption
□	sales_records_cleaned.csv	2.1 MB	text/csv	14 Oct 20...	Standard	14 Oct 202...	Not public	–	Google-managed key
□	sales_records_cleaned3.csv	2.1 MB	text/csv	15 Oct 20...	Standard	15 Oct 202...	Not public	–	Google-managed key

Figure 2: MAKE NEW BUCKET AND UPLOAD THE CLEANED DATASET



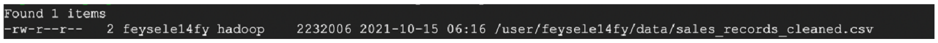
●	Name ↑	Role
✓	fp5240cluster-m	Master
✓	fp5240cluster-w-0	Worker
✓	fp5240cluster-w-1	Worker

Figure 3: CREATE THE CLUSTER WITH 1 MASTER & 2 WORKER NODES & SSH IN



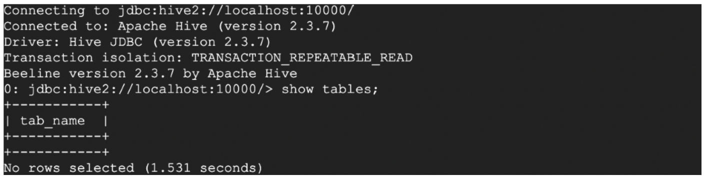
```
4:58 2021 from 35.235.241.19
:~$ cd data
:~/data$ ls
sales_records_cleaned3.csv
:~/data$ 
```

Figure 4: COPY THE DATASET USING THE "gsutil cp -n gs://fp5240bucket/data/filename" COMMAND TO THE LINUX DIRECTORY



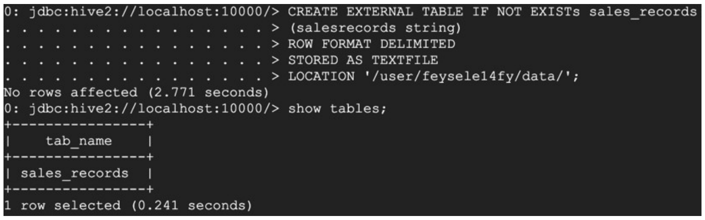
```
Found 1 items
-rw-r--r--   2 feysele14fy hadoop    2232006 2021-10-15 06:16 /user/feysele14fy/data/sales_records_cleaned.csv
```

Figure 5: PUT THE DATASET INTO THE HDFS STORAGE FROM THE LINUX DIRECTORY I CREATED



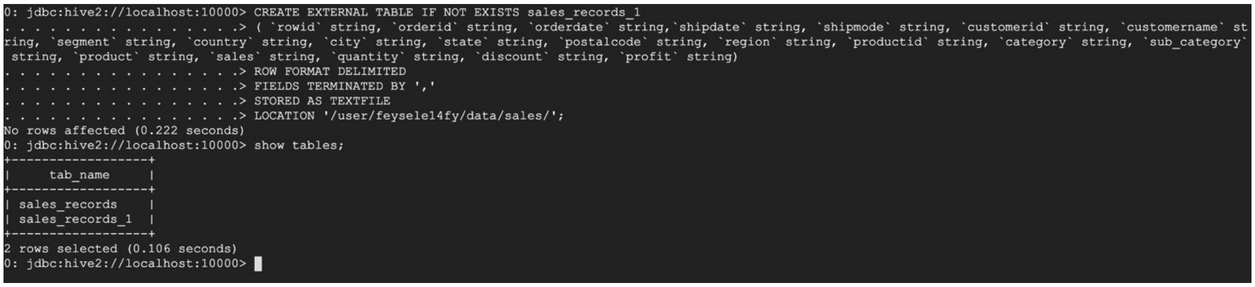
```
Connecting to jdbc:hive2://localhost:10000/
Connected to: Apache Hive (version 2.3.7)
Driver: Hive JDBC (version 2.3.7)
Transaction isolation: TRANSACTION_REPEATABLE_READ
Beeline version 2.3.7 by Apache Hive
0: jdbc:hive2://localhost:10000/> show tables;
+-----------+
| tab_name  |
+-----------+
+-----------+
No rows selected (1.531 seconds)
```

Figure 6: CONNECT TO HIVE TO START THE SCHEME CREATION



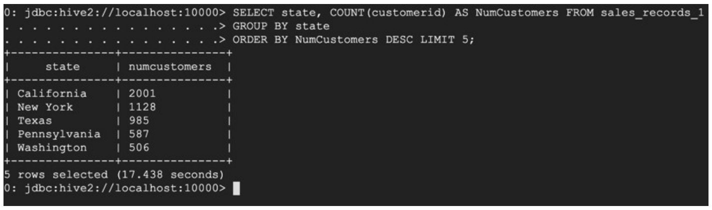
```
0: jdbc:hive2://localhost:10000/> CREATE EXTERNAL TABLE IF NOT EXISTs sales_records
. . . . . . . . . . . . . . . . > (salesrecords string)
. . . . . . . . . . . . . . . . > ROW FORMAT DELIMITED
. . . . . . . . . . . . . . . . > STORED AS TEXTFILE
. . . . . . . . . . . . . . . . > LOCATION '/user/feysele14fy/data/';
No rows affected (2.771 seconds)
0: jdbc:hive2://localhost:10000/> show tables;
+----------------+
|    tab_name    |
+----------------+
| sales_records  |
+----------------+
1 row selected (0.241 seconds)
```

Figure 7: CREATE A SIMPLE SCHEME TO SEE IF EVERYTHIN IS WORKING IN THE METASTORE



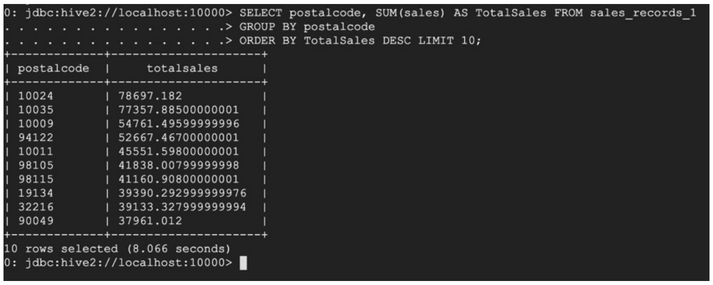
```
0: jdbc:hive2://localhost:10000> CREATE EXTERNAL TABLE IF NOT EXISTS sales_records_1
. . . . . . . . . . . . . . . . .> ( `rowid` string, `orderid` string, `orderdate` string,`shipdate` string, `shipmode` string, `customerid` string, `customername` st
ring, `segment` string, `country` string, `city` string, `state` string, `postalcode` string, `region` string, `productid` string, `category` string, `sub_category`
 string, `product` string, `sales` string, `quantity` string, `discount` string, `profit` string)
. . . . . . . . . . . . . . . . .> ROW FORMAT DELIMITED
. . . . . . . . . . . . . . . . .> FIELDS TERMINATED BY ','
. . . . . . . . . . . . . . . . .> STORED AS TEXTFILE
. . . . . . . . . . . . . . . . .> LOCATION '/user/feysele14fy/data/sales/';
No rows affected (0.222 seconds)
0: jdbc:hive2://localhost:10000> show tables;
+------------------+
|     tab_name     |
+------------------+
| sales_records    |
| sales_records_1  |
+------------------+
2 rows selected (0.106 seconds)
0: jdbc:hive2://localhost:10000>
```

Figure 8: CREATE THE COMPLEX SCHEME & TERMINATE BY ','



```
0: jdbc:hive2://localhost:10000> SELECT state, COUNT(customerid) AS NumCustomers FROM sales_records_1
. . . . . . . . . . . . . . . . .> GROUP BY state
. . . . . . . . . . . . . . . . .> ORDER BY NumCustomers DESC LIMIT 5;
+---------------+---------------+
|     state     | numcustomers  |
+---------------+---------------+
| California    | 2001          |
| New York      | 1128          |
| Texas         | 985           |
| Pennsylvania  | 587           |
| Washington    | 506           |
+---------------+---------------+
5 rows selected (17.438 seconds)
0: jdbc:hive2://localhost:10000>
```

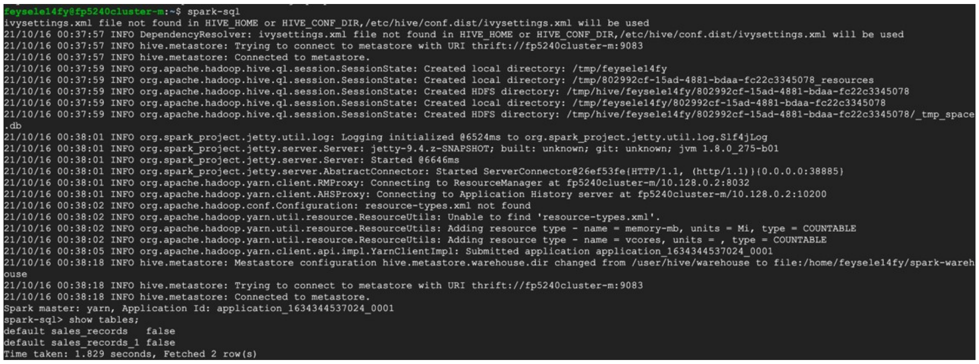
Figure 9: QUERY FOR TOP 5 STATE BASED ON # OF CUSTOMER RESIDING



```
0: jdbc:hive2://localhost:10000> SELECT postalcode, SUM(sales) AS TotalSales FROM sales_records_1
. . . . . . . . . . . . . . . . .> GROUP BY postalcode
. . . . . . . . . . . . . . . . .> ORDER BY TotalSales DESC LIMIT 10;
+-------------+---------------------+
| postalcode  |     totalsales      |
+-------------+---------------------+
| 10024       | 78697.182           |
| 10035       | 77357.88500000001   |
| 10009       | 54761.49599999996   |
| 94122       | 52667.46700000001   |
| 10011       | 45551.59800000001   |
| 98105       | 41838.00799999998   |
| 98115       | 41160.90800000001   |
| 19134       | 39390.292999999976  |
| 32216       | 39133.327999999994  |
| 90049       | 37961.012           |
+-------------+---------------------+
10 rows selected (8.066 seconds)
0: jdbc:hive2://localhost:10000>
```

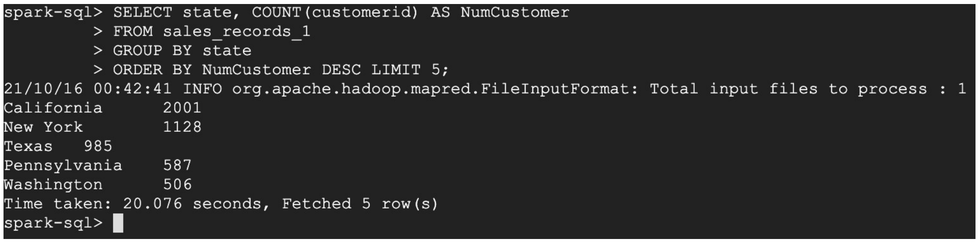
Figure 10: QUERY FOR TOP TOTAL SALES BASED ON TOP 10 ZIP CODES

#4:



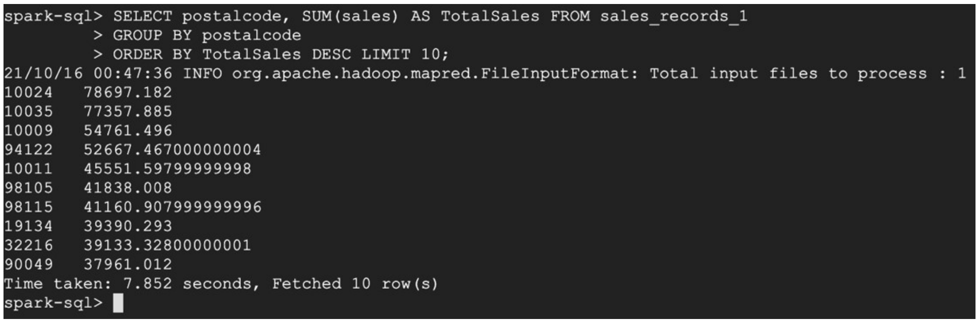
```
feysele14fy@fp5240cluster-m:~$ spark-sql
ivysettings.xml file not found in HIVE_HOME or HIVE_CONF_DIR,/etc/hive/conf.dist/ivysettings.xml will be used
21/10/16 00:37:57 INFO DependencyResolver: ivysettings.xml file not found in HIVE_HOME or HIVE_CONF_DIR,/etc/hive/conf.dist/ivysettings.xml will be used
21/10/16 00:37:57 INFO hive.metastore: Trying to connect to metastore with URI thrift://fp5240cluster-m:9083
21/10/16 00:37:57 INFO hive.metastore: Connected to metastore.
21/10/16 00:37:59 INFO org.apache.hadoop.hive.ql.session.SessionState: Created local directory: /tmp/feysele14fy
21/10/16 00:37:59 INFO org.apache.hadoop.hive.ql.session.SessionState: Created local directory: /tmp/802992cf-15ad-4881-bdaa-fc22c3345078_resources
21/10/16 00:37:59 INFO org.apache.hadoop.hive.ql.session.SessionState: Created HDFS directory: /tmp/hive/feysele14fy/802992cf-15ad-4881-bdaa-fc22c3345078
21/10/16 00:37:59 INFO org.apache.hadoop.hive.ql.session.SessionState: Created local directory: /tmp/feysele14fy/802992cf-15ad-4881-bdaa-fc22c3345078
21/10/16 00:37:59 INFO org.apache.hadoop.hive.ql.session.SessionState: Created HDFS directory: /tmp/hive/feysele14fy/802992cf-15ad-4881-bdaa-fc22c3345078/_tmp_space
.db
21/10/16 00:38:01 INFO org.spark_project.jetty.util.log: Logging initialized @6524ms to org.spark_project.jetty.util.log.Slf4jLog
21/10/16 00:38:01 INFO org.spark_project.jetty.server.Server: jetty-9.4.z-SNAPSHOT; built: unknown; git: unknown; jvm 1.8.0_275-b01
21/10/16 00:38:01 INFO org.spark_project.jetty.server.Server: Started @6646ms
21/10/16 00:38:01 INFO org.spark_project.jetty.server.AbstractConnector: Started ServerConnector@26ef53fe{HTTP/1.1, (http/1.1)}{0.0.0.0:38885}
21/10/16 00:38:01 INFO org.apache.hadoop.yarn.client.RMProxy: Connecting to ResourceManager at fp5240cluster-m/10.128.0.2:8032
21/10/16 00:38:01 INFO org.apache.hadoop.yarn.client.AHSProxy: Connecting to Application History server at fp5240cluster-m/10.128.0.2:10200
21/10/16 00:38:02 INFO org.apache.hadoop.conf.Configuration: resource-types.xml not found
21/10/16 00:38:02 INFO org.apache.hadoop.yarn.util.resource.ResourceUtils: Unable to find 'resource-types.xml'.
21/10/16 00:38:02 INFO org.apache.hadoop.yarn.util.resource.ResourceUtils: Adding resource type - name = memory-mb, units = Mi, type = COUNTABLE
21/10/16 00:38:02 INFO org.apache.hadoop.yarn.util.resource.ResourceUtils: Adding resource type - name = vcores, units = , type = COUNTABLE
21/10/16 00:38:05 INFO org.apache.hadoop.yarn.client.api.impl.YarnClientImpl: Submitted application application_1634344537024_0001
21/10/16 00:38:18 INFO hive.metastore: Mestastore configuration hive.metastore.warehouse.dir changed from /user/hive/warehouse to file:/home/feysele14fy/spark-wareh
ouse
21/10/16 00:38:18 INFO hive.metastore: Trying to connect to metastore with URI thrift://fp5240cluster-m:9083
21/10/16 00:38:18 INFO hive.metastore: Connected to metastore.
Spark master: yarn, Application Id: application_1634344537024_0001
spark-sql> show tables;
default sales_records   false
default sales_records_1 false
Time taken: 1.829 seconds, Fetched 2 row(s)
```

Figure 11: STARTING SPARK-SQL



```
spark-sql> SELECT state, COUNT(customerid) AS NumCustomer
         > FROM sales_records_1
         > GROUP BY state
         > ORDER BY NumCustomer DESC LIMIT 5;
21/10/16 00:42:41 INFO org.apache.hadoop.mapred.FileInputFormat: Total input files to process : 1
California      2001
New York        1128
Texas   985
Pennsylvania    587
Washington      506
Time taken: 20.076 seconds, Fetched 5 row(s)
spark-sql> █
```

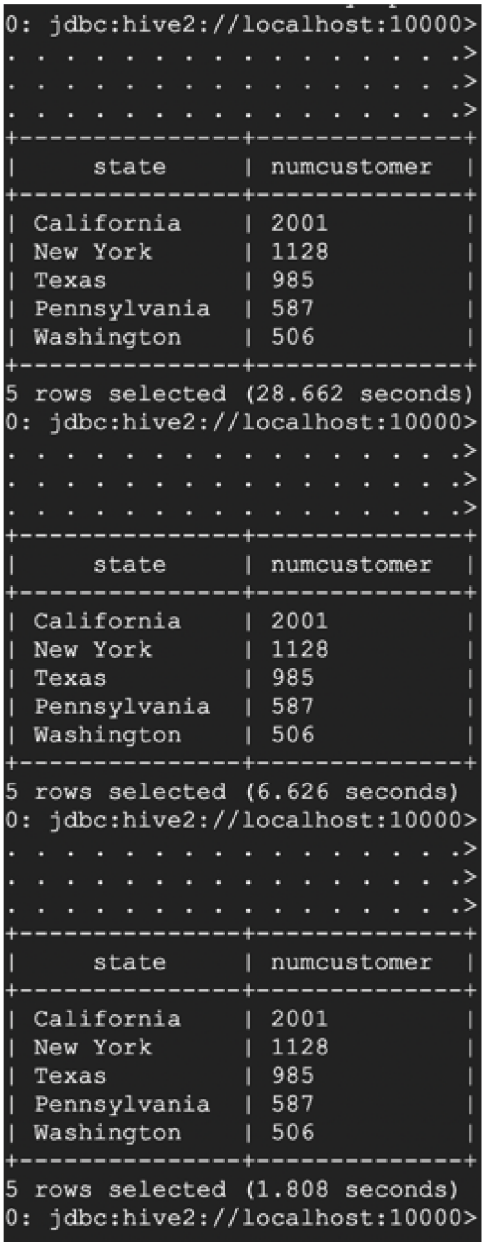
Figure 12: QUERY FOR TOP 5 STATE BASED ON # OF CUSTOMER RESIDING (SPARK)



```
spark-sql> SELECT postalcode, SUM(sales) AS TotalSales FROM sales_records_1
         > GROUP BY postalcode
         > ORDER BY TotalSales DESC LIMIT 10;
21/10/16 00:47:36 INFO org.apache.hadoop.mapred.FileInputFormat: Total input files to process : 1
10024   78697.182
10035   77357.885
10009   54761.496
94122   52667.467000000004
10011   45551.59799999998
98105   41838.008
98115   41160.907999999996
19134   39390.293
32216   39133.32800000001
90049   37961.012
Time taken: 7.852 seconds, Fetched 10 row(s)
spark-sql> █
```

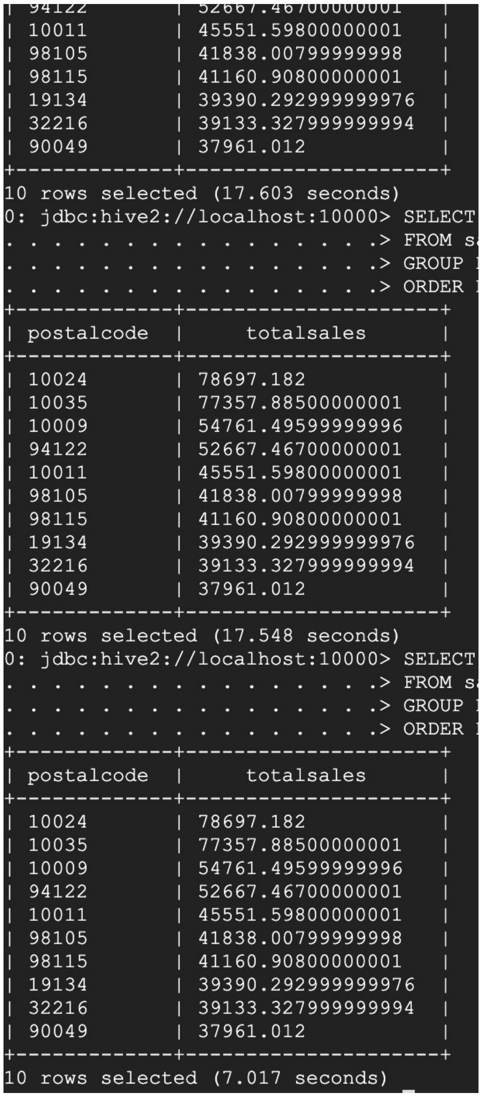
Figure 13: QUERY FOR TOTAL SALES BASED ON TOP 10 ZIP CODES (SPARK)

#5:



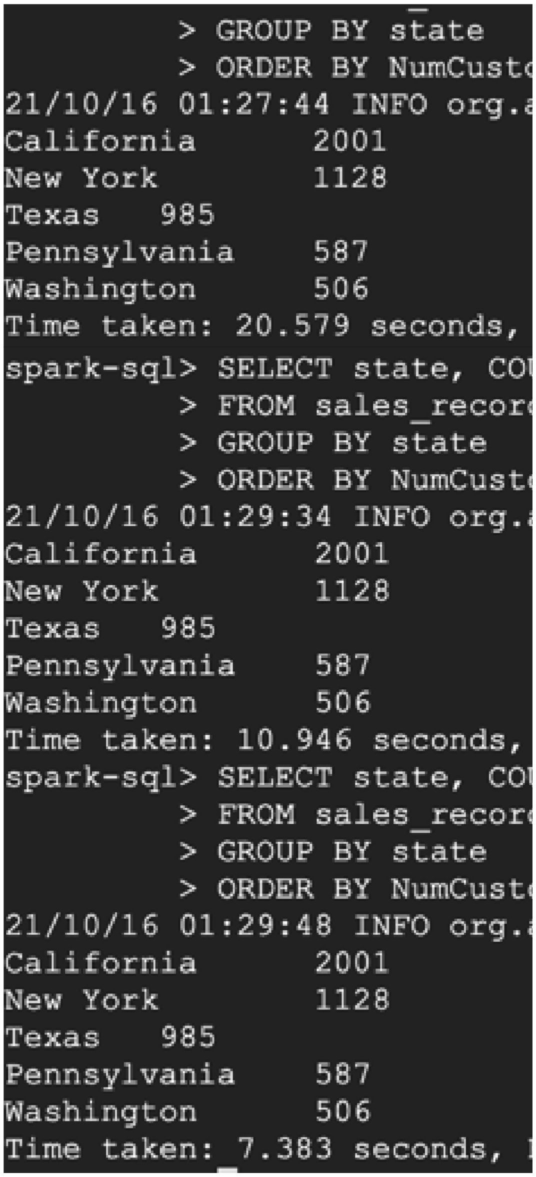
```
0: jdbc:hive2://localhost:10000>
. . . . . . . . . . . . . . . .>
. . . . . . . . . . . . . . . .>
. . . . . . . . . . . . . . . .>
+---------------+--------------+
|     state     | numcustomer  |
+---------------+--------------+
| California    | 2001         |
| New York      | 1128         |
| Texas         | 985          |
| Pennsylvania  | 587          |
| Washington    | 506          |
+---------------+--------------+
5 rows selected (28.662 seconds)
0: jdbc:hive2://localhost:10000>
. . . . . . . . . . . . . . . .>
. . . . . . . . . . . . . . . .>
. . . . . . . . . . . . . . . .>
+---------------+--------------+
|     state     | numcustomer  |
+---------------+--------------+
| California    | 2001         |
| New York      | 1128         |
| Texas         | 985          |
| Pennsylvania  | 587          |
| Washington    | 506          |
+---------------+--------------+
5 rows selected (6.626 seconds)
0: jdbc:hive2://localhost:10000>
. . . . . . . . . . . . . . . .>
. . . . . . . . . . . . . . . .>
. . . . . . . . . . . . . . . .>
+---------------+--------------+
|     state     | numcustomer  |
+---------------+--------------+
| California    | 2001         |
| New York      | 1128         |
| Texas         | 985          |
| Pennsylvania  | 587          |
| Washington    | 506          |
+---------------+--------------+
5 rows selected (1.808 seconds)
0: jdbc:hive2://localhost:10000>
```

Figure 14: QUERY SPEEDTEST ON TOP 5 STATES (HIVE)



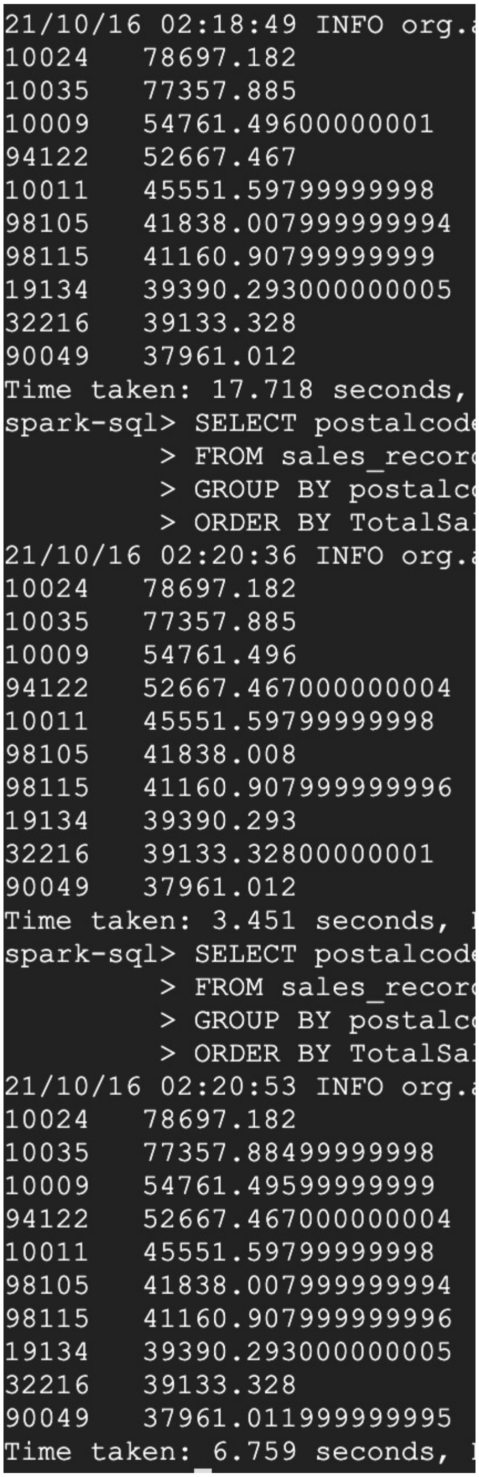
```
| 94122       | 52667.46700000001   |
| 10011       | 45551.59800000001   |
| 98105       | 41838.00799999998   |
| 98115       | 41160.90800000001   |
| 19134       | 39390.292999999976  |
| 32216       | 39133.327999999994  |
| 90049       | 37961.012           |
+-------------+---------------------+
10 rows selected (17.603 seconds)
0: jdbc:hive2://localhost:10000> SELECT
. . . . . . . . . . . . . . . .> FROM s
. . . . . . . . . . . . . . . .> GROUP
. . . . . . . . . . . . . . . .> ORDER
+-------------+---------------------+
| postalcode  |     totalsales      |
+-------------+---------------------+
| 10024       | 78697.182           |
| 10035       | 77357.88500000001   |
| 10009       | 54761.49599999996   |
| 94122       | 52667.46700000001   |
| 10011       | 45551.59800000001   |
| 98105       | 41838.00799999998   |
| 98115       | 41160.90800000001   |
| 19134       | 39390.292999999976  |
| 32216       | 39133.327999999994  |
| 90049       | 37961.012           |
+-------------+---------------------+
10 rows selected (17.548 seconds)
0: jdbc:hive2://localhost:10000> SELECT
. . . . . . . . . . . . . . . .> FROM s
. . . . . . . . . . . . . . . .> GROUP
. . . . . . . . . . . . . . . .> ORDER
+-------------+---------------------+
| postalcode  |     totalsales      |
+-------------+---------------------+
| 10024       | 78697.182           |
| 10035       | 77357.88500000001   |
| 10009       | 54761.49599999996   |
| 94122       | 52667.46700000001   |
| 10011       | 45551.59800000001   |
| 98105       | 41838.00799999998   |
| 98115       | 41160.90800000001   |
| 19134       | 39390.292999999976  |
| 32216       | 39133.327999999994  |
| 90049       | 37961.012           |
+-------------+---------------------+
10 rows selected (7.017 seconds)
```

Figure 15: QUERY SPEEDTEST ON TOP 10 ZIPCODES (HIVE)



```
         > GROUP BY state
         > ORDER BY NumCusto
21/10/16 01:27:44 INFO org.a
California      2001
New York        1128
Texas   985
Pennsylvania    587
Washington      506
Time taken: 20.579 seconds,
spark-sql> SELECT state, CO
         > FROM sales_recor
         > GROUP BY state
         > ORDER BY NumCust
21/10/16 01:29:34 INFO org.a
California      2001
New York        1128
Texas   985
Pennsylvania    587
Washington      506
Time taken: 10.946 seconds,
spark-sql> SELECT state, CO
         > FROM sales_recor
         > GROUP BY state
         > ORDER BY NumCust
21/10/16 01:29:48 INFO org.a
California      2001
New York        1128
Texas   985
Pennsylvania    587
Washington      506
Time taken: 7.383 seconds,
```

Figure 16: QUERY SPEEDTEST ON TOP 5 STATE (SPARK)



```
21/10/16 02:18:49 INFO org.a
10024   78697.182
10035   77357.885
10009   54761.49600000001
94122   52667.467
10011   45551.59799999998
98105   41838.007999999994
98115   41160.90799999999
19134   39390.293000000005
32216   39133.328
90049   37961.012
Time taken: 17.718 seconds,
spark-sql> SELECT postalcode
         > FROM sales_recor
         > GROUP BY postalc
         > ORDER BY TotalSal
21/10/16 02:20:36 INFO org.a
10024   78697.182
10035   77357.885
10009   54761.496
94122   52667.467000000004
10011   45551.59799999998
98105   41838.008
98115   41160.907999999996
19134   39390.293
32216   39133.32800000001
90049   37961.012
Time taken: 3.451 seconds,
spark-sql> SELECT postalcode
         > FROM sales_recor
         > GROUP BY postalc
         > ORDER BY TotalSal
21/10/16 02:20:53 INFO org.a
10024   78697.182
10035   77357.88499999998
10009   54761.49599999999
94122   52667.467000000004
10011   45551.59799999998
98105   41838.007999999994
98115   41160.907999999996
19134   39390.293000000005
32216   39133.328
90049   37961.011999999995
Time taken: 6.759 seconds,
```

Figure 17: QUERY SPEEDTEST ON TOP 10 ZIPCODES (SPARK) 1ST QUERY SPEED (sec):

HIVE

1. 28.7
2. 6.6
3. 1.8

SPARK

1. 20.6
2. 10.9
3. 7.4

Report:
Spark initially had a 28.2% decrease in time taken on the 1st round, but hive was able to the cut the time by 93.7% at the last round compared to SPARK's 64.1%.

2ND QUERY SPEED (sec):
HIVE

1. 17.6
2. 17.5
3. 7.0

SPARK

1. 17.7
2. 3.5
3. 6.8

Report:
There was only 0.1 second difference at the 1st round for the second query test, but at the second and last round Spark made noticeable strides in the time. However, Hive was able to close the gap to only 0.2 seconds which is not significant.

Although Spark is technically faster because it loads the dataset in ram rather than on a solid drive, there is no big gap between the two systems because the dataset is not big enough. In addition, the queries are not to intensive which results in less variances. As a result, I recommend the company uses Hive for the time being until we are presented with enormous data.