Project: Customer-Behavior-MySQL2Hive

Big Data Analytics and Architecture

(MCADSN13201)

Submitted By :

Rajesh Chaurasiya

MCA DS & AI

University Roll No : 1240259034

Submitted To :

Mr. Harsh Gaur

Index

1. Introduction:

The project "Customer-Behavior-MySQL2Hive" focuses on analysing e-commerce customer data to understand purchasing behaviour, satisfaction levels, and spending patterns using Big Data technologies. It demonstrates how traditional databases like MySQL can be integrated with Apache Hive in a Cloudera environment to efficiently handle and process large-scale datasets. By performing analytical operations in both MySQL and Hive, this project highlights the comparative benefits of relational and distributed data processing systems in terms of scalability, performance, and flexibility.

The core objective is to explore, analyse, and derive meaningful insights from e-commerce customer behaviour data initially stored in MySQL and later migrated to Hive for distributed computation. Through this approach, the project showcases how Big Data analytics can help organizations gain a deeper understanding of customer trends, enhance satisfaction levels, and improve decision-making. The insights obtained can assist businesses in optimizing marketing strategies, increasing customer retention, and driving data-driven growth.

2. Description of the Dataset:

The dataset contains records of e-commerce customers with fields such as:

- Customer ID
- Gender
- Age
- City
- Membership Type
- Total Spend
- Items Purchased
- Average Rating
- Discount Applied
- Days Since Last Purchase
- Satisfaction Level

The dataset helps in understanding:

- Spending patterns by location, membership type, and gender.
- Satisfaction levels across customer segments.
- Relationship between discounts, loyalty, and spending behaviour.

3. Project Scope:

The project focuses on integrating MySQL (traditional RDBMS) with Hive (Big Data warehouse) to:

- Store and manage customer data at scale
- Perform SQL and HQL queries for analytics
- Generate insights about customer satisfaction, spending trends, and loyalty
- Visualize the results to support business decision-making

It also includes demonstrating the ETL process (Extract, Transform, Load) from MySQL to Hive for scalable data processing.

4. Goals:

a. **Analyse Customer Behaviour Patterns:**
To study customer spending, satisfaction, and engagement trends using transactional data.

b. **Integrate MySQL with Hive:**
To perform seamless data migration from a relational database (MySQL) to a distributed Big Data environment (Hive on Cloudera).

c. **Perform Comparative Analysis:**
To compare the execution efficiency and scalability of analytical queries between MySQL (RDBMS) and Hive (HQL).

d. **Identify Key Business Insights:**
To extract actionable insights such as top-spending cities, high-value customers, and satisfaction level correlations.

e. **Implement ETL Process:**
To demonstrate the Extract, Transform, and Load process for moving structured data into the Hadoop ecosystem.

f. **Enhance Decision-Making for E-commerce:**
To support business growth by understanding customer loyalty, discount effectiveness, and membership-driven profitability.

g. **Visualize Data Effectively:**
To present analytical findings using visual representations like bar charts, pie charts, and scatter plots for better interpretation.

h. **Demonstrate End-to-End Big Data Workflow:**
To showcase a complete analytical pipeline from dataset loading, query execution, and insight generation to visualization within a Big Data ecosystem.

5. Tool & Working Environment:

A. Tools Used:

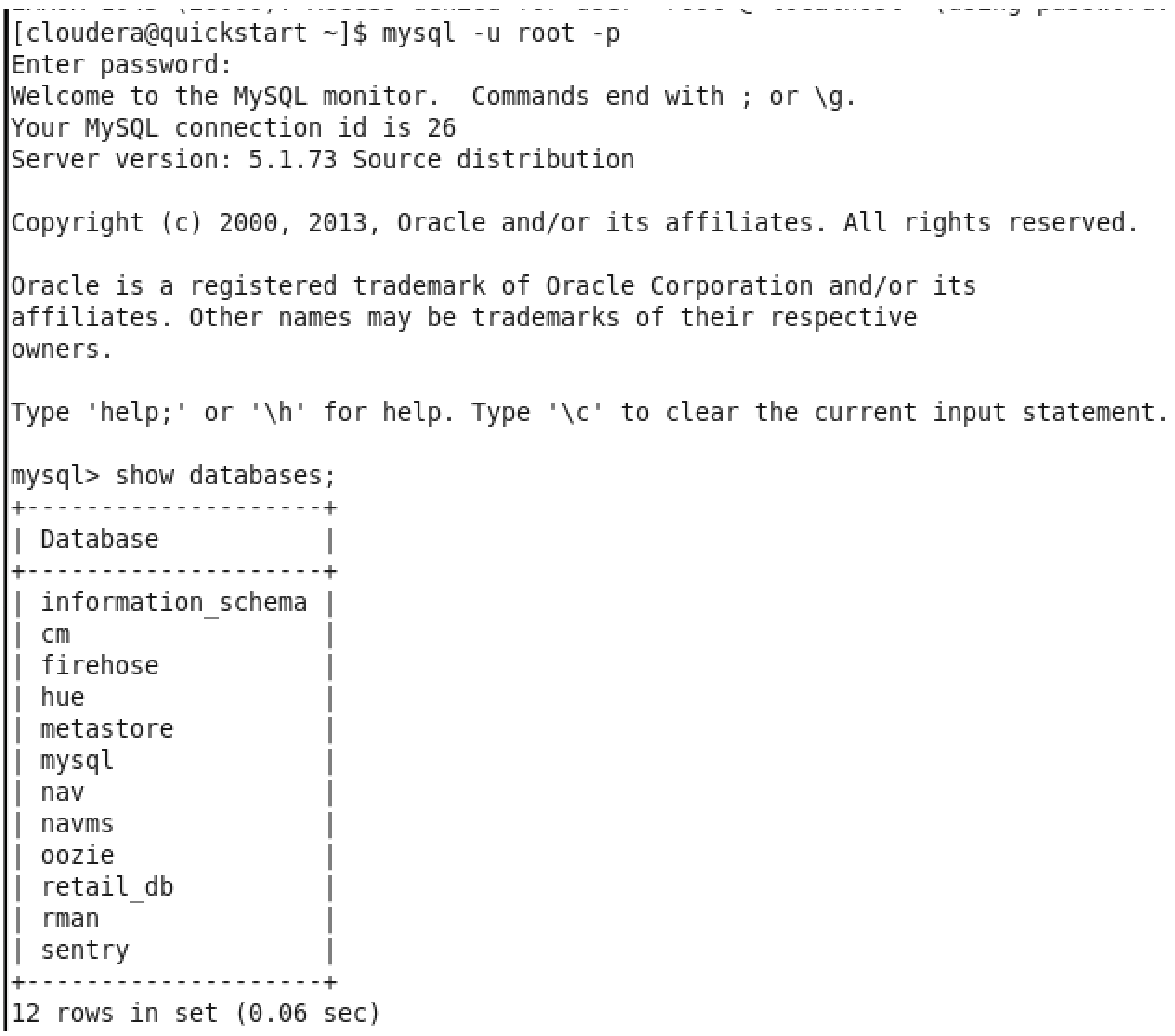
- MySQL (Database for storage and initial analysis)
- Apache Hive (Big Data analysis)
- Cloudera (Hadoop environment for Hive execution)
- CSV Dataset
- Command Line Interface / Terminal

B. Environment Setup:

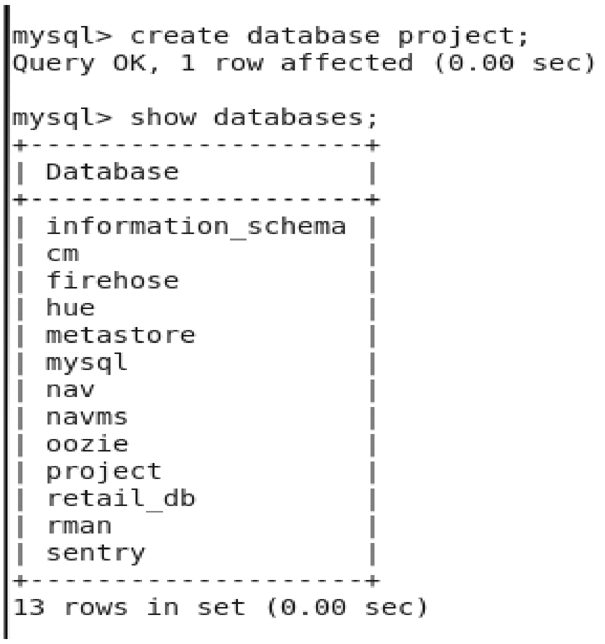
- Cloudera VM running on Linux
- MySQL database installed and connected
- Hive configured with MySQL as source

6. Performing Analysis on MySQL:

Login MySQL and Create new Database:

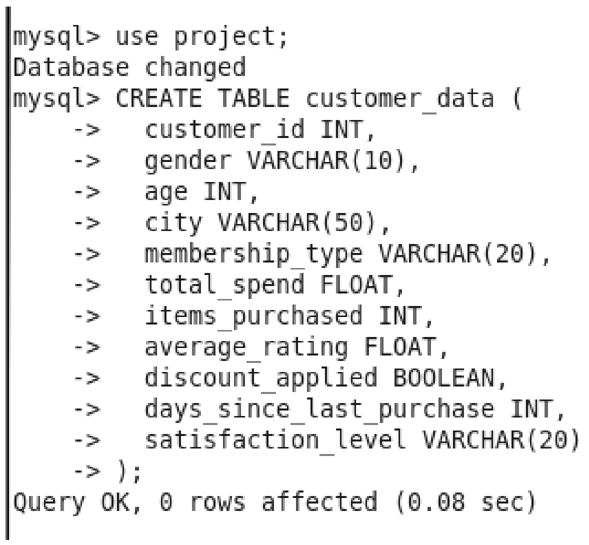


```
[cloudera@quickstart ~]$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 26
Server version: 5.1.73 Source distribution

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| cm                 |
| firehose           |
| hue                |
| metastore          |
| mysql              |
| nav                |
| navms              |
| oozie              |
| retail_db          |
| rman               |
| sentry             |
+--------------------+
12 rows in set (0.06 sec)
```



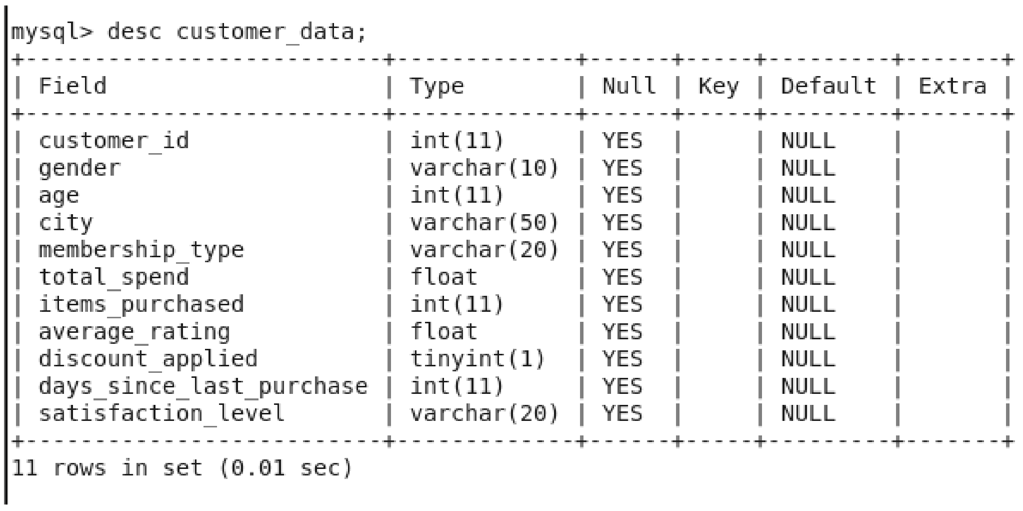
```
mysql> create database project;
Query OK, 1 row affected (0.00 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| cm                 |
| firehose           |
| hue                |
| metastore          |
| mysql              |
| nav                |
| navms              |
| oozie              |
| project            |
| retail_db          |
| rman               |
| sentry             |
+--------------------+
13 rows in set (0.00 sec)
```

Use Database & Create tables: project And customer_data



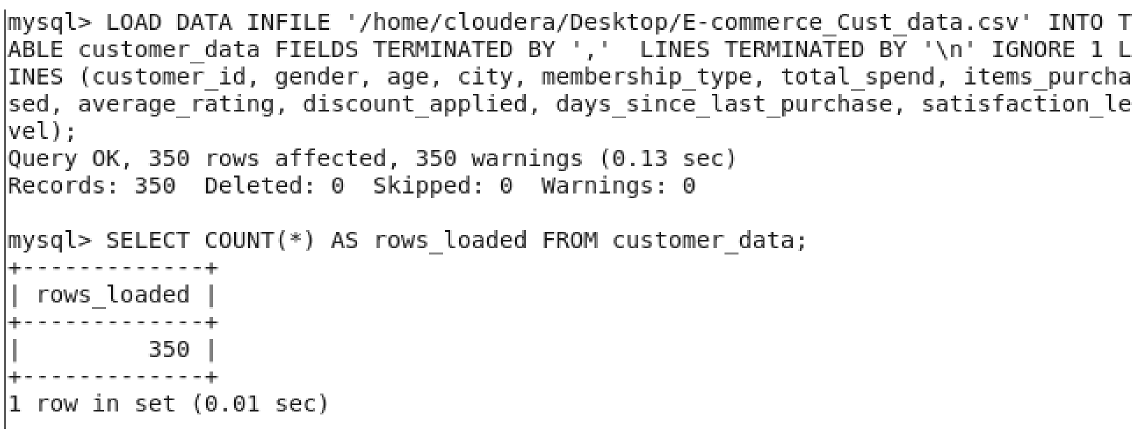
```
mysql> use project;
Database changed
mysql> CREATE TABLE customer_data (
    ->   customer_id INT,
    ->   gender VARCHAR(10),
    ->   age INT,
    ->   city VARCHAR(50),
    ->   membership_type VARCHAR(20),
    ->   total_spend FLOAT,
    ->   items_purchased INT,
    ->   average_rating FLOAT,
    ->   discount_applied BOOLEAN,
    ->   days_since_last_purchase INT,
    ->   satisfaction_level VARCHAR(20)
    -> );
Query OK, 0 rows affected (0.08 sec)
```

Displaying tables:



```
mysql> desc customer_data;
+--------------------------+-------------+------+-----+---------+-------+
| Field                    | Type        | Null | Key | Default | Extra |
+--------------------------+-------------+------+-----+---------+-------+
| customer_id              | int(11)     | YES  |     | NULL    |       |
| gender                   | varchar(10) | YES  |     | NULL    |       |
| age                      | int(11)     | YES  |     | NULL    |       |
| city                     | varchar(50) | YES  |     | NULL    |       |
| membership_type          | varchar(20) | YES  |     | NULL    |       |
| total_spend              | float       | YES  |     | NULL    |       |
| items_purchased          | int(11)     | YES  |     | NULL    |       |
| average_rating           | float       | YES  |     | NULL    |       |
| discount_applied         | tinyint(1)  | YES  |     | NULL    |       |
| days_since_last_purchase | int(11)     | YES  |     | NULL    |       |
| satisfaction_level       | varchar(20) | YES  |     | NULL    |       |
+--------------------------+-------------+------+-----+---------+-------+
11 rows in set (0.01 sec)
```

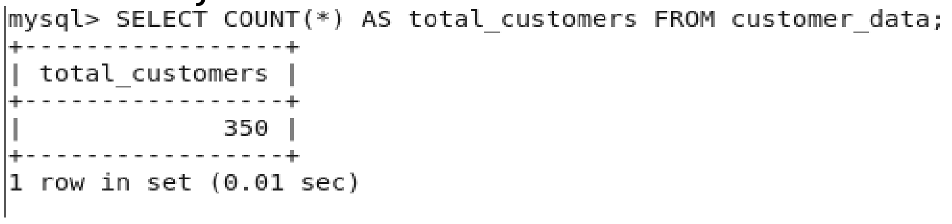
Load Data in MySQL:



```
mysql> LOAD DATA INFILE '/home/cloudera/Desktop/E-commerce_Cust_data.csv' INTO T
ABLE customer_data FIELDS TERMINATED BY ','  LINES TERMINATED BY '\n' IGNORE 1 L
INES (customer_id, gender, age, city, membership_type, total_spend, items_purcha
sed, average_rating, discount_applied, days_since_last_purchase, satisfaction_le
vel);
Query OK, 350 rows affected, 350 warnings (0.13 sec)
Records: 350  Deleted: 0  Skipped: 0  Warnings: 0

mysql> SELECT COUNT(*) AS rows_loaded FROM customer_data;
+-------------+
| rows_loaded |
+-------------+
|         350 |
+-------------+
1 row in set (0.01 sec)
```

7. Performing SQL queries on the table:

1. **How many customers are there in total?**



```
mysql> SELECT COUNT(*) AS total_customers FROM customer_data;
+-----------------+
| total_customers |
+-----------------+
|             350 |
+-----------------+
1 row in set (0.01 sec)
```

2. **What is the average total spending of all customers?**

```
mysql> SELECT AVG(total_spend) AS average_spend FROM customer_data;
+-------------------+
| average_spend     |
+-------------------+
| 845.381716831752 |
+-------------------+
1 row in set (0.00 sec)
```

3. Which city has the highest average total spend?

```
mysql> SELECT city, AVG(total_spend) AS avg_spend FROM customer_data GROUP BY city ORDER BY avg_spend DESC LIMIT 1;
+---------------+------------------+
| city          | avg_spend        |
+---------------+------------------+
| San Francisco | 1459.77239779768 |
+---------------+------------------+
1 row in set (0.01 sec)
```

4. How many customers are in each membership type?

```
mysql> SELECT membership_type, COUNT(*) AS total_customers FROM customer_data GROUP BY membership_type;
+-----------------+-----------------+
| membership_type | total_customers |
+-----------------+-----------------+
| Bronze          |             116 |
| Gold            |             117 |
| Silver          |             117 |
+-----------------+-----------------+
3 rows in set (0.00 sec)
```

5. What is the satisfaction level distribution among all customers?

```
mysql> SELECT satisfaction_level, COUNT(*) AS total_customers FROM customer_data GROUP BY satisfaction_level;
+--------------------+-----------------+
| satisfaction_level | total_customers |
+--------------------+-----------------+
                  |               2 |
           |             107 |
         |             125 |
       |             116 |
+--------------------+-----------------+
4 rows in set (0.00 sec)
```

6. What is the average rating by gender?

```
mysql> SELECT gender, AVG(average_rating) AS avg_rating FROM customer_data GROUP BY gender;
+--------+------------------+
| gender | avg_rating       |
+--------+------------------+
| Female | 3.73142854281834 |
| Male   | 4.30685715402876 |
+--------+------------------+
2 rows in set (0.00 sec)
```

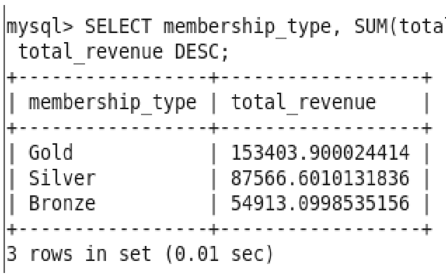
7. Which customers used a discount and were satisfied?

```
mysql> SELECT customer_id, gender, city, total_spend, satisfaction_level FROM customer_data WHERE discount_applied =
TRUE AND satisfaction_level = 'Satisfied';
Empty set (0.00 sec)
```

8. Who are the top 3 customers with the highest total spend?

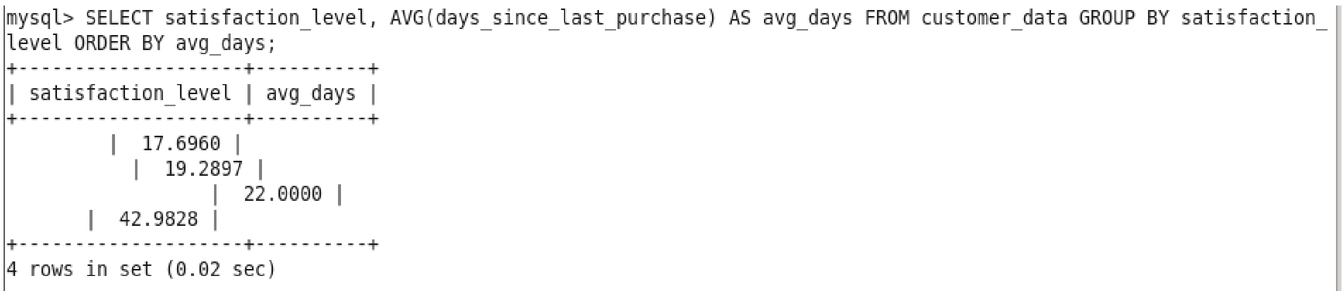
```
mysql> SELECT customer_id, city, total_spend FROM customer_data ORDER BY total_spend DESC LIMIT 3;
+-------------+---------------+-------------+
| customer_id | city          | total_spend |
+-------------+---------------+-------------+
|         110 | San Francisco |      1520.1 |
|         218 | San Francisco |      1500.1 |
|         128 | San Francisco |      1500.1 |
+-------------+---------------+-------------+
3 rows in set (0.01 sec)
```

9. What is the total revenue generated from each membership type?



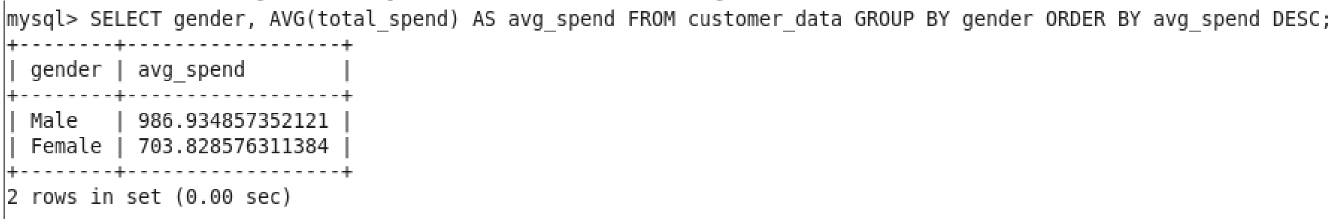
```
mysql> SELECT membership_type, SUM(total_spend) AS total_revenue FROM customer_data GROUP BY membership_type ORDER BY
 total_revenue DESC;
+-----------------+------------------+
| membership_type | total_revenue    |
+-----------------+------------------+
| Gold            | 153403.900024414 |
| Silver          | 87566.6010131836 |
| Bronze          | 54913.0998535156 |
+-----------------+------------------+
3 rows in set (0.01 sec)
```

10. What is the average number of days since last purchase for each satisfaction level?



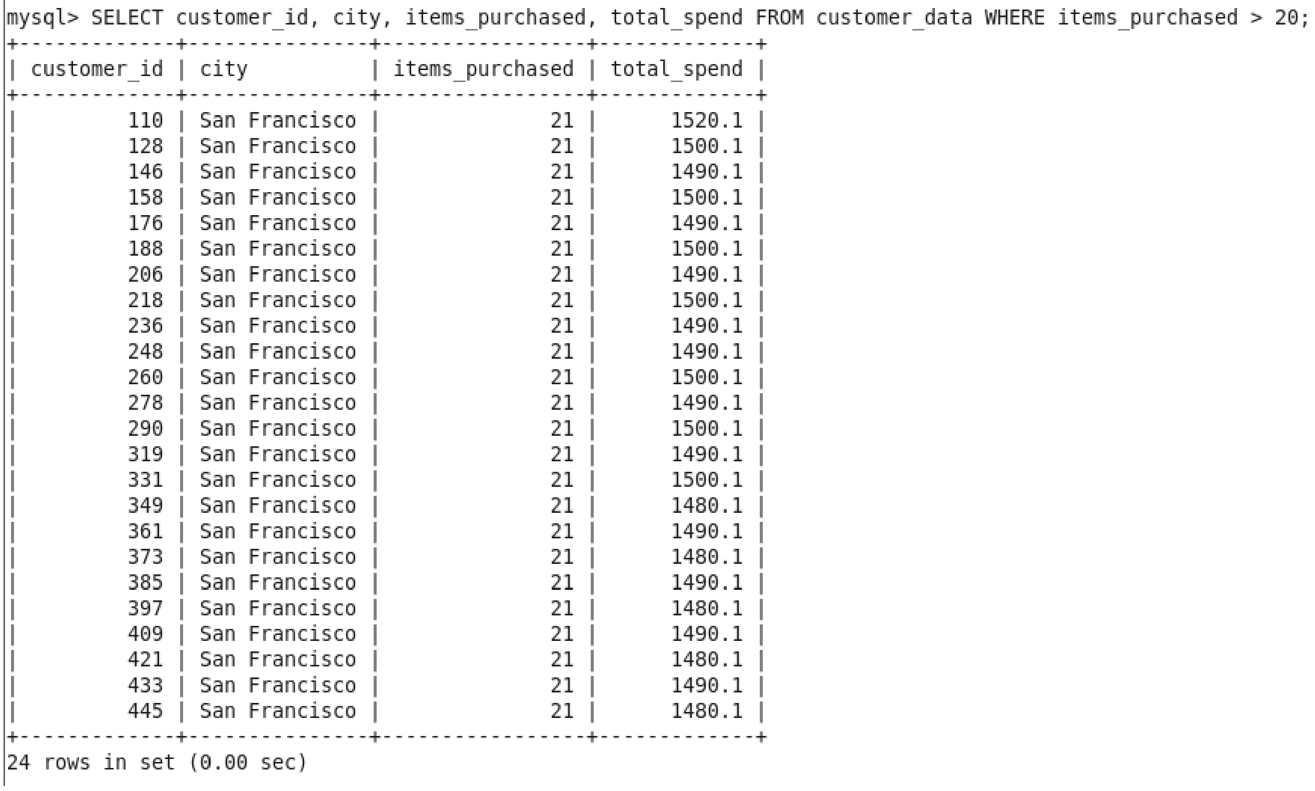
```
mysql> SELECT satisfaction_level, AVG(days_since_last_purchase) AS avg_days FROM customer_data GROUP BY satisfaction_
level ORDER BY avg_days;
+--------------------+----------+
| satisfaction_level | avg_days |
+--------------------+----------+
         |  17.6960 |
           |  19.2897 |
                  |  22.0000 |
       |  42.9828 |
+--------------------+----------+
4 rows in set (0.02 sec)
```

11. Which gender spends more on average?



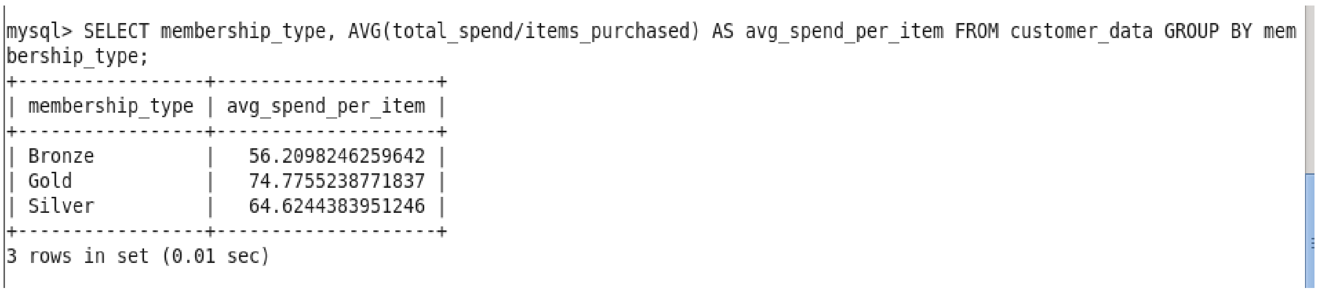
```
mysql> SELECT gender, AVG(total_spend) AS avg_spend FROM customer_data GROUP BY gender ORDER BY avg_spend DESC;
+--------+------------------+
| gender | avg_spend        |
+--------+------------------+
| Male   | 986.934857352121 |
| Female | 703.828576311384 |
+--------+------------------+
2 rows in set (0.00 sec)
```

12. Find customers who purchased more than 20 items.



```
mysql> SELECT customer_id, city, items_purchased, total_spend FROM customer_data WHERE items_purchased > 20;
+-------------+---------------+-----------------+------------+
| customer_id | city          | items_purchased | total_spend |
+-------------+---------------+-----------------+------------+
|         110 | San Francisco |              21 |     1520.1 |
|         128 | San Francisco |              21 |     1500.1 |
|         146 | San Francisco |              21 |     1490.1 |
|         158 | San Francisco |              21 |     1500.1 |
|         176 | San Francisco |              21 |     1490.1 |
|         188 | San Francisco |              21 |     1500.1 |
|         206 | San Francisco |              21 |     1490.1 |
|         218 | San Francisco |              21 |     1500.1 |
|         236 | San Francisco |              21 |     1490.1 |
|         248 | San Francisco |              21 |     1490.1 |
|         260 | San Francisco |              21 |     1500.1 |
|         278 | San Francisco |              21 |     1490.1 |
|         290 | San Francisco |              21 |     1500.1 |
|         319 | San Francisco |              21 |     1490.1 |
|         331 | San Francisco |              21 |     1500.1 |
|         349 | San Francisco |              21 |     1480.1 |
|         361 | San Francisco |              21 |     1490.1 |
|         373 | San Francisco |              21 |     1480.1 |
|         385 | San Francisco |              21 |     1490.1 |
|         397 | San Francisco |              21 |     1480.1 |
|         409 | San Francisco |              21 |     1490.1 |
|         421 | San Francisco |              21 |     1480.1 |
|         433 | San Francisco |              21 |     1490.1 |
|         445 | San Francisco |              21 |     1480.1 |
+-------------+---------------+-----------------+------------+
24 rows in set (0.00 sec)
```

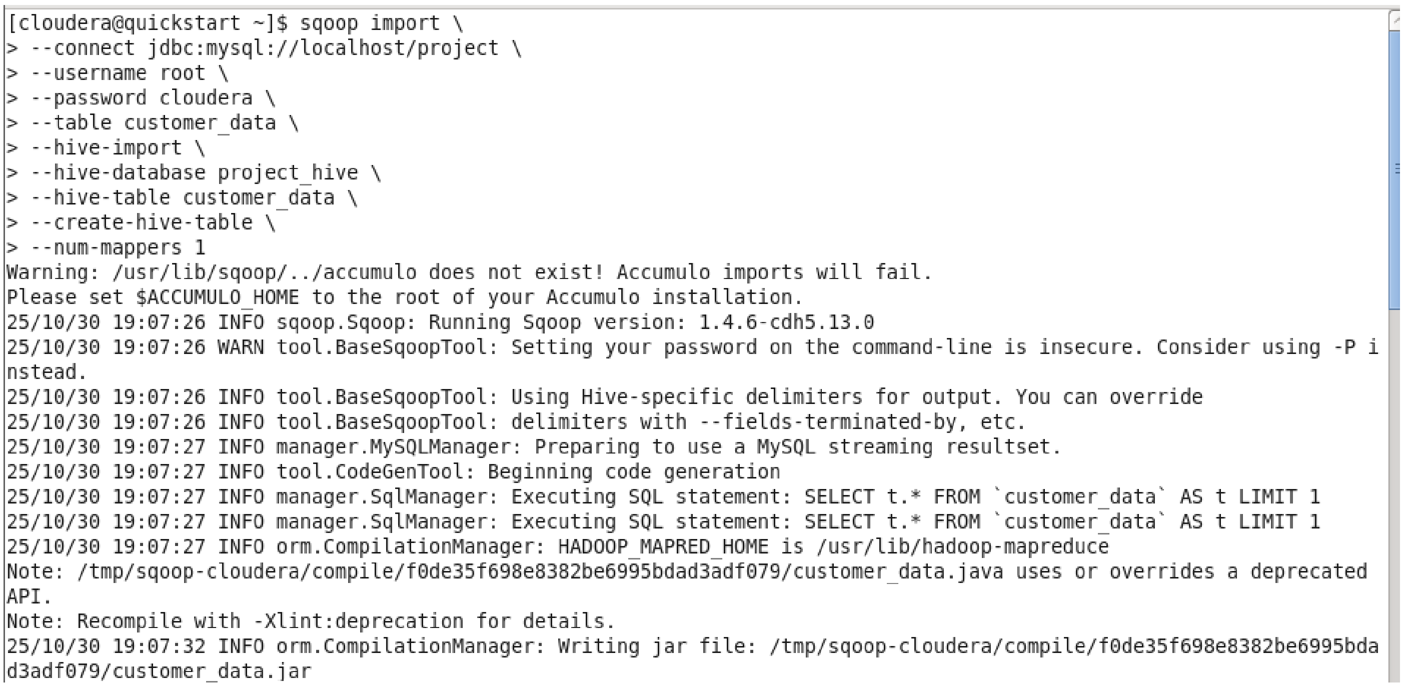
13. Find average spend per item for each membership type.



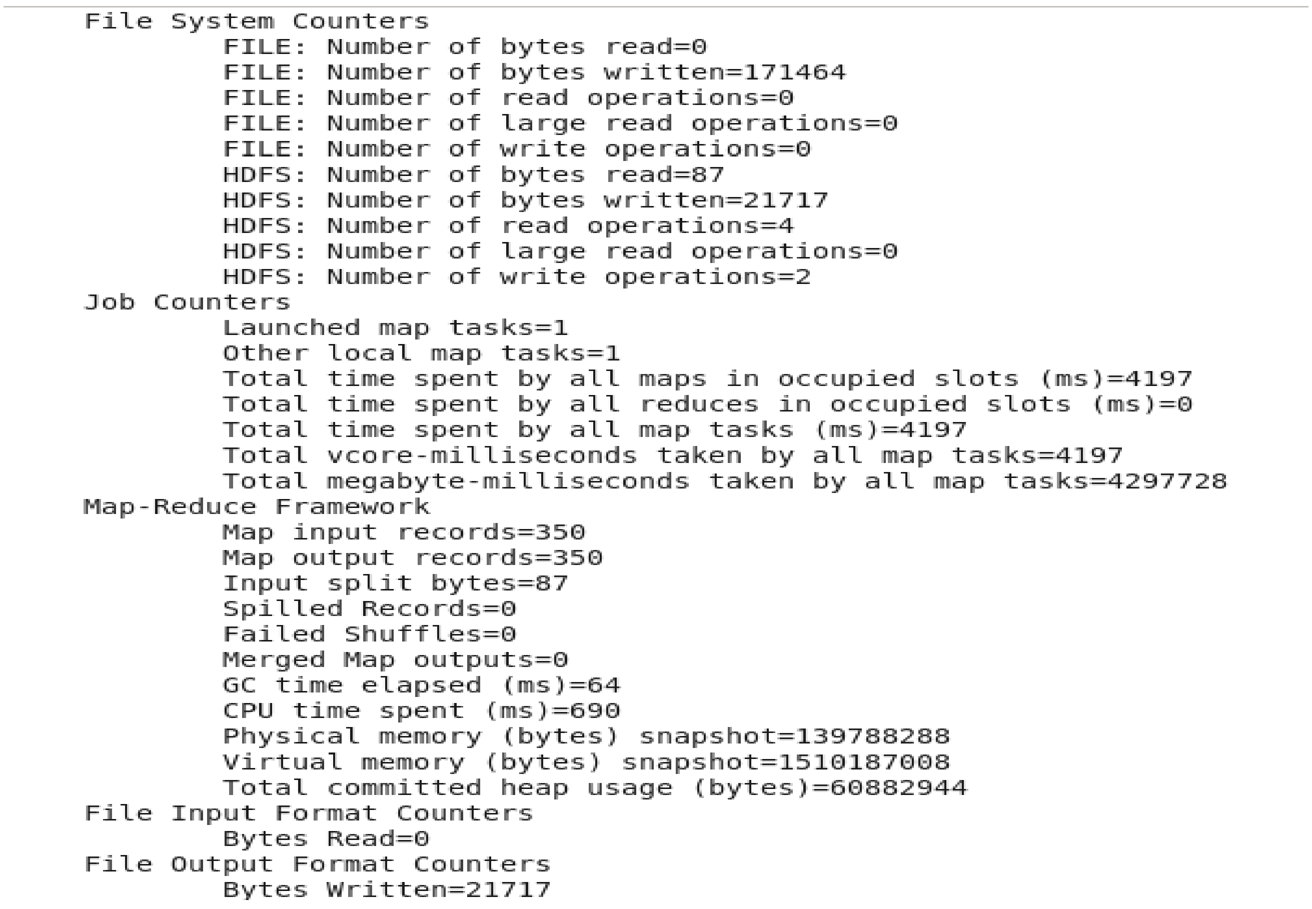
```
mysql> SELECT membership_type, AVG(total_spend/items_purchased) AS avg_spend_per_item FROM customer_data GROUP BY mem
bership_type;
+-----------------+--------------------+
| membership_type | avg_spend_per_item |
+-----------------+--------------------+
| Bronze          |   56.2098246259642 |
| Gold            |   74.7755238771837 |
| Silver          |   64.6244383951246 |
+-----------------+--------------------+
3 rows in set (0.01 sec)
```

8. Performing Analysis on Hive:

Loading the dataset from MySQL into Hive:



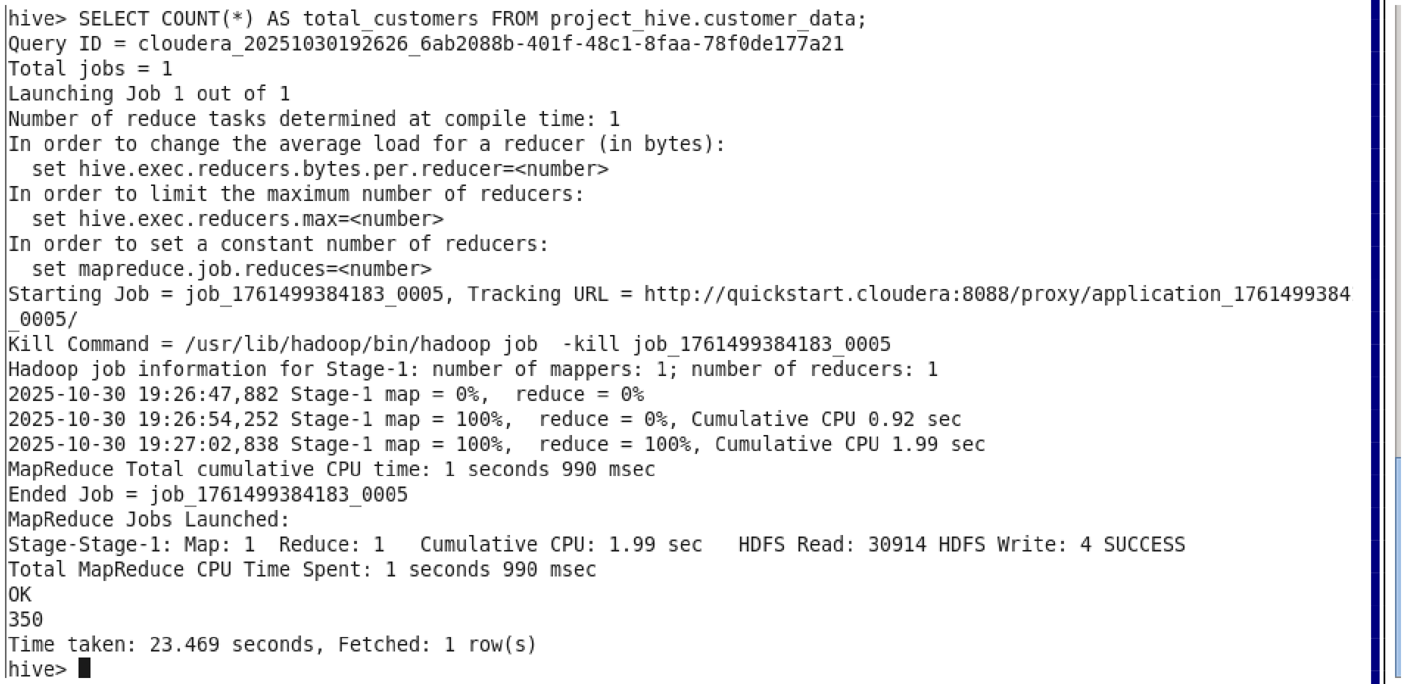
```
[cloudera@quickstart ~]$ sqoop import \
> --connect jdbc:mysql://localhost/project \
> --username root \
> --password cloudera \
> --table customer_data \
> --hive-import \
> --hive-database project_hive \
> --hive-table customer_data \
> --create-hive-table \
> --num-mappers 1
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
25/10/30 19:07:26 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.13.0
25/10/30 19:07:26 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P i
nstead.
25/10/30 19:07:26 INFO tool.BaseSqoopTool: Using Hive-specific delimiters for output. You can override
25/10/30 19:07:26 INFO tool.BaseSqoopTool: delimiters with --fields-terminated-by, etc.
25/10/30 19:07:27 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
25/10/30 19:07:27 INFO tool.CodeGenTool: Beginning code generation
25/10/30 19:07:27 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `customer_data` AS t LIMIT 1
25/10/30 19:07:27 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `customer_data` AS t LIMIT 1
25/10/30 19:07:27 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/lib/hadoop-mapreduce
Note: /tmp/sqoop-cloudera/compile/f0de35f698e8382be6995bdad3adf079/customer_data.java uses or overrides a deprecated
API.
Note: Recompile with -Xlint:deprecation for details.
25/10/30 19:07:32 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-cloudera/compile/f0de35f698e8382be6995bda
d3adf079/customer_data.jar
```



```
	File System Counters
		FILE: Number of bytes read=0
		FILE: Number of bytes written=171464
		FILE: Number of read operations=0
		FILE: Number of large read operations=0
		FILE: Number of write operations=0
		HDFS: Number of bytes read=87
		HDFS: Number of bytes written=21717
		HDFS: Number of read operations=4
		HDFS: Number of large read operations=0
		HDFS: Number of write operations=2
	Job Counters
		Launched map tasks=1
		Other local map tasks=1
		Total time spent by all maps in occupied slots (ms)=4197
		Total time spent by all reduces in occupied slots (ms)=0
		Total time spent by all map tasks (ms)=4197
		Total vcore-milliseconds taken by all map tasks=4197
		Total megabyte-milliseconds taken by all map tasks=4297728
	Map-Reduce Framework
		Map input records=350
		Map output records=350
		Input split bytes=87
		Spilled Records=0
		Failed Shuffles=0
		Merged Map outputs=0
		GC time elapsed (ms)=64
		CPU time spent (ms)=690
		Physical memory (bytes) snapshot=139788288
		Virtual memory (bytes) snapshot=1510187008
		Total committed heap usage (bytes)=60882944
	File Input Format Counters
		Bytes Read=0
	File Output Format Counters
		Bytes Written=21717
```

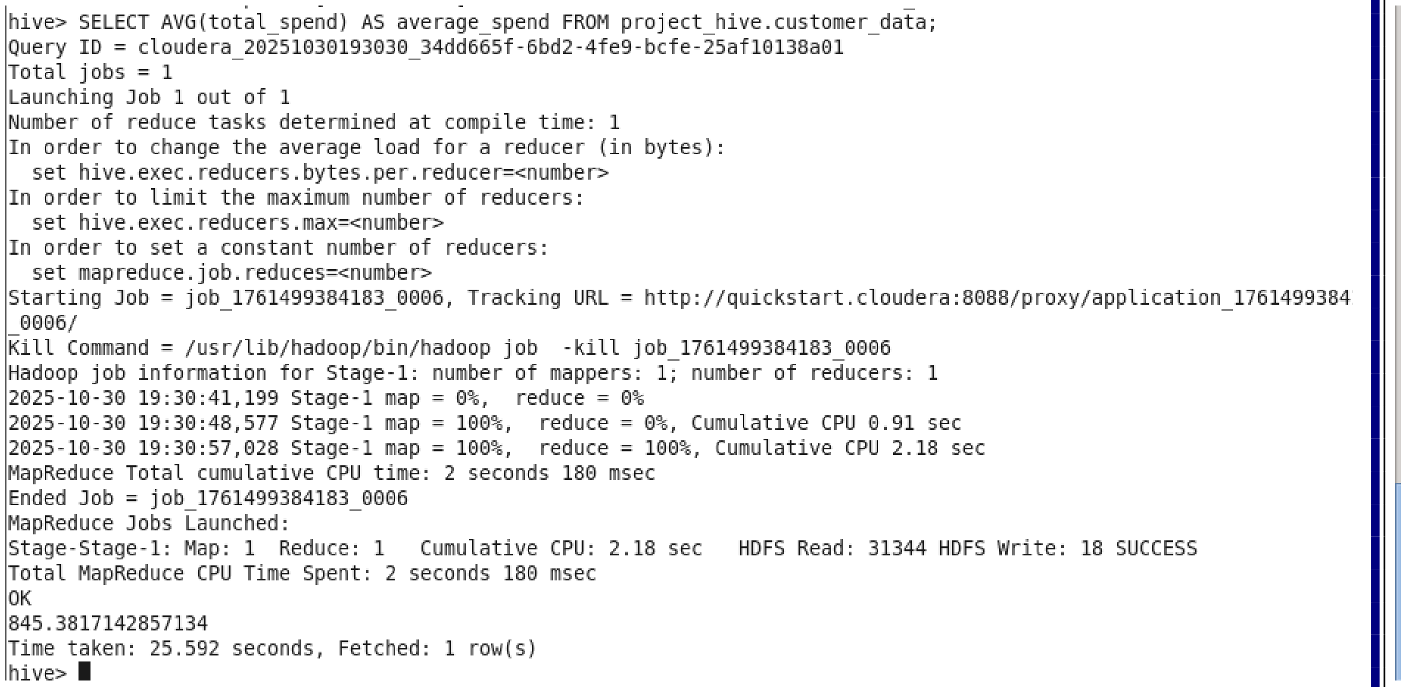
9. Performing HQL Queries on the table:

1. How many customers are there in total?



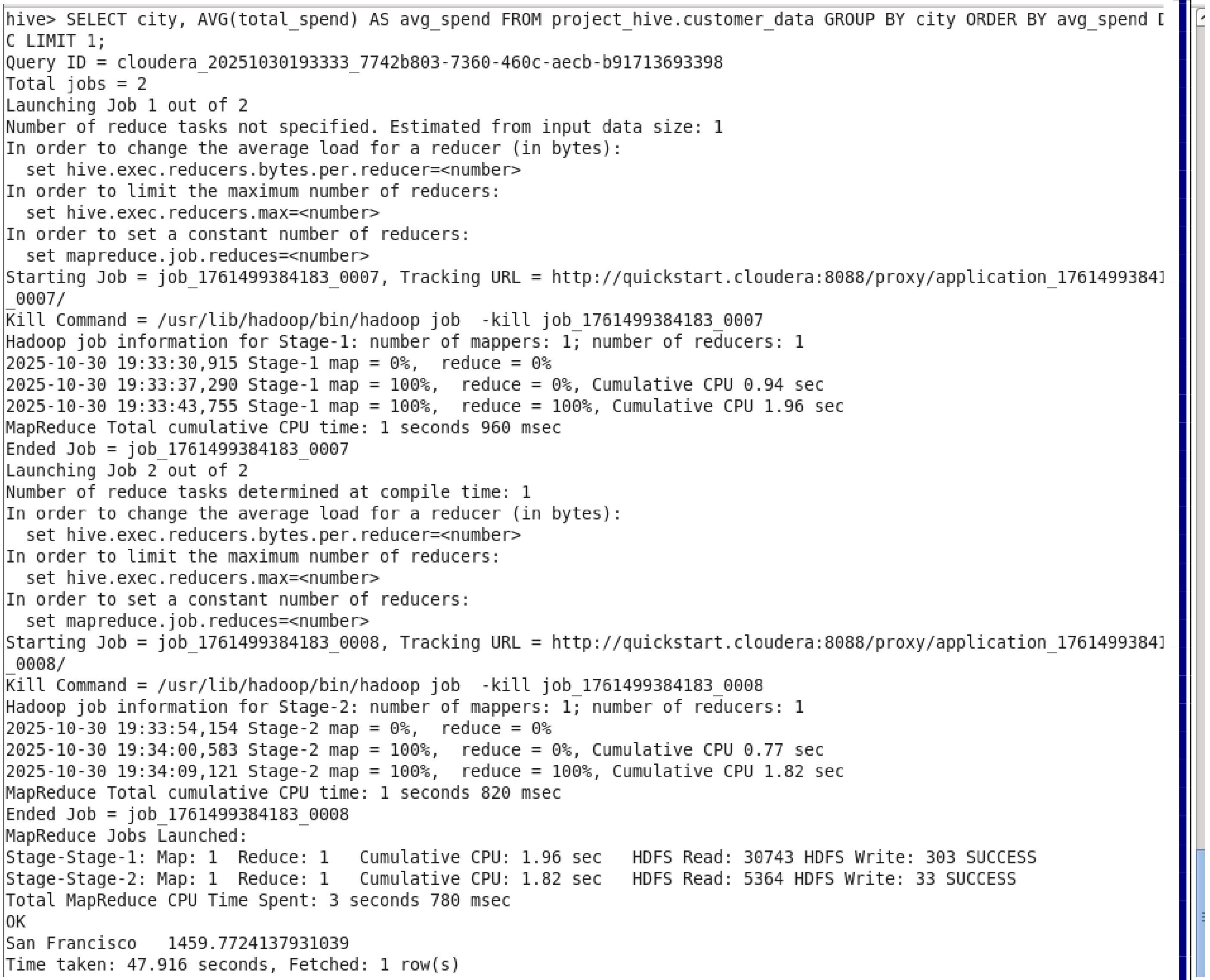
```
hive> SELECT COUNT(*) AS total_customers FROM project_hive.customer_data;
Query ID = cloudera_20251030192626_6ab2088b-401f-48c1-8faa-78f0de177a21
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1761499384183_0005, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1761499384
_0005/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1761499384183_0005
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2025-10-30 19:26:47,882 Stage-1 map = 0%,  reduce = 0%
2025-10-30 19:26:54,252 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 0.92 sec
2025-10-30 19:27:02,838 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 1.99 sec
MapReduce Total cumulative CPU time: 1 seconds 990 msec
Ended Job = job_1761499384183_0005
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 1.99 sec   HDFS Read: 30914 HDFS Write: 4 SUCCESS
Total MapReduce CPU Time Spent: 1 seconds 990 msec
OK
350
Time taken: 23.469 seconds, Fetched: 1 row(s)
hive> █
```

2. What is the average total spending of all customers?



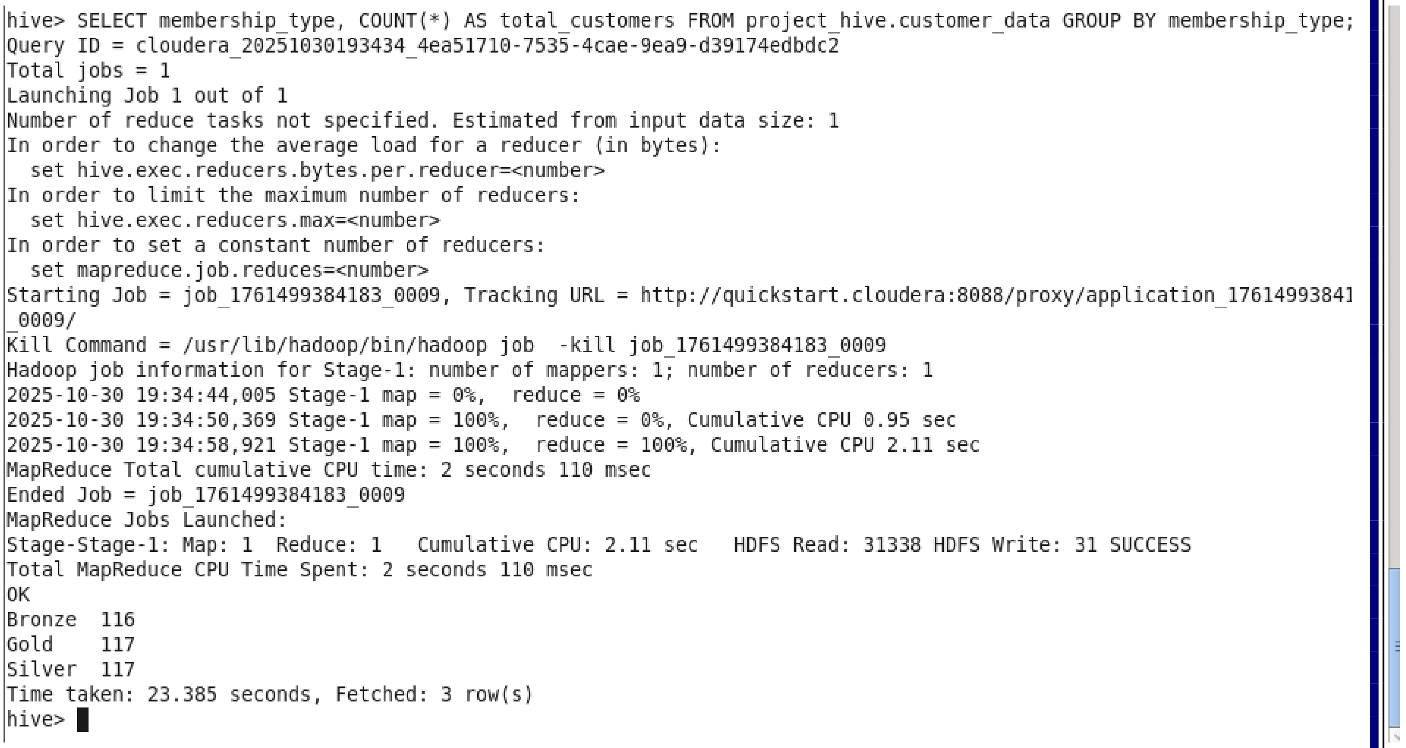
```
hive> SELECT AVG(total_spend) AS average_spend FROM project_hive.customer_data;
Query ID = cloudera_20251030193030_34dd665f-6bd2-4fe9-bcfe-25af10138a01
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1761499384183_0006, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1761499384
_0006/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1761499384183_0006
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2025-10-30 19:30:41,199 Stage-1 map = 0%,  reduce = 0%
2025-10-30 19:30:48,577 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 0.91 sec
2025-10-30 19:30:57,028 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 2.18 sec
MapReduce Total cumulative CPU time: 2 seconds 180 msec
Ended Job = job_1761499384183_0006
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 2.18 sec   HDFS Read: 31344 HDFS Write: 18 SUCCESS
Total MapReduce CPU Time Spent: 2 seconds 180 msec
OK
845.3817142857134
Time taken: 25.592 seconds, Fetched: 1 row(s)
hive> █
```

3. Which city has the highest average total spend?



```
hive> SELECT city, AVG(total_spend) AS avg_spend FROM project_hive.customer_data GROUP BY city ORDER BY avg_spend D
C LIMIT 1;
Query ID = cloudera_20251030193333_7742b803-7360-460c-aecb-b91713693398
Total jobs = 2
Launching Job 1 out of 2
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1761499384183_0007, Tracking URL = http://quickstart.cloudera:8088/proxy/application_17614993841
_0007/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1761499384183_0007
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2025-10-30 19:33:30,915 Stage-1 map = 0%,  reduce = 0%
2025-10-30 19:33:37,290 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 0.94 sec
2025-10-30 19:33:43,755 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 1.96 sec
MapReduce Total cumulative CPU time: 1 seconds 960 msec
Ended Job = job_1761499384183_0007
Launching Job 2 out of 2
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1761499384183_0008, Tracking URL = http://quickstart.cloudera:8088/proxy/application_17614993841
_0008/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1761499384183_0008
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2025-10-30 19:33:54,154 Stage-2 map = 0%,  reduce = 0%
2025-10-30 19:34:00,583 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 0.77 sec
2025-10-30 19:34:09,121 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 1.82 sec
MapReduce Total cumulative CPU time: 1 seconds 820 msec
Ended Job = job_1761499384183_0008
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 1.96 sec   HDFS Read: 30743 HDFS Write: 303 SUCCESS
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 1.82 sec   HDFS Read: 5364 HDFS Write: 33 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 780 msec
OK
San Francisco   1459.7724137931039
Time taken: 47.916 seconds, Fetched: 1 row(s)
```

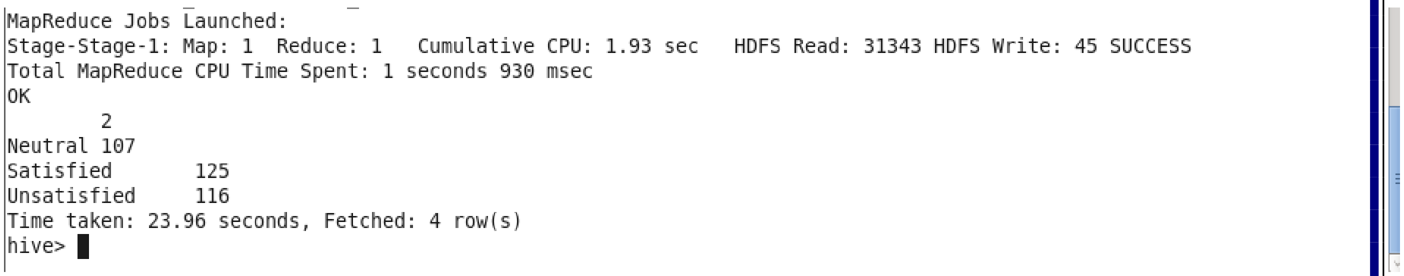
4. How many customers are in each membership type?



```
hive> SELECT membership_type, COUNT(*) AS total_customers FROM project_hive.customer_data GROUP BY membership_type;
Query ID = cloudera_20251030193434_4ea51710-7535-4cae-9ea9-d39174edbdc2
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1761499384183_0009, Tracking URL = http://quickstart.cloudera:8088/proxy/application_17614993841
_0009/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1761499384183_0009
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2025-10-30 19:34:44,005 Stage-1 map = 0%,  reduce = 0%
2025-10-30 19:34:50,369 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 0.95 sec
2025-10-30 19:34:58,921 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 2.11 sec
MapReduce Total cumulative CPU time: 2 seconds 110 msec
Ended Job = job_1761499384183_0009
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 2.11 sec   HDFS Read: 31338 HDFS Write: 31 SUCCESS
Total MapReduce CPU Time Spent: 2 seconds 110 msec
OK
Bronze  116
Gold    117
Silver  117
Time taken: 23.385 seconds, Fetched: 3 row(s)
hive> █
```

5. What is the satisfaction level distribution among all customers?

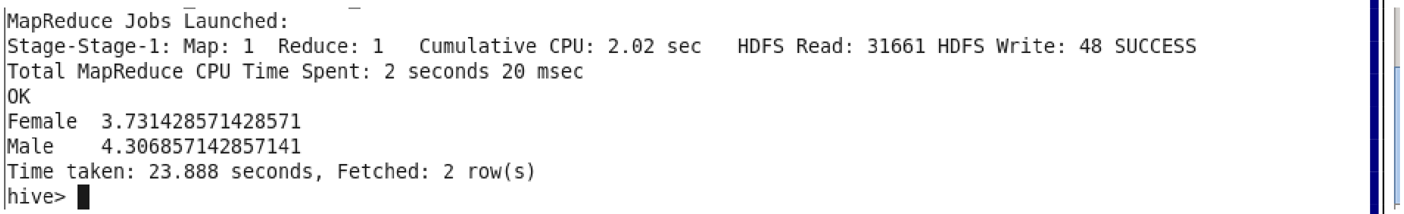
SELECT satisfaction_level, COUNT(*) AS total_customers FROM project_hive.customer_data GROUP BY satisfaction_level;



```
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 1.93 sec   HDFS Read: 31343 HDFS Write: 45 SUCCESS
Total MapReduce CPU Time Spent: 1 seconds 930 msec
OK
        2
Neutral 107
Satisfied       125
Unsatisfied     116
Time taken: 23.96 seconds, Fetched: 4 row(s)
hive>
```

6. What is the average rating by gender?

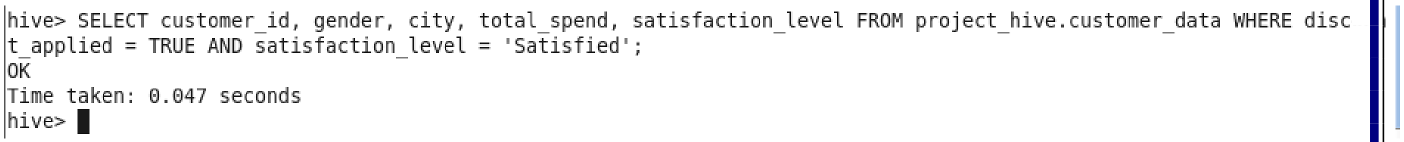
SELECT gender, AVG(average_rating) AS avg_rating FROM project_hive.customer_data GROUP BY gender;



```
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 2.02 sec   HDFS Read: 31661 HDFS Write: 48 SUCCESS
Total MapReduce CPU Time Spent: 2 seconds 20 msec
OK
Female  3.731428571428571
Male    4.306857142857141
Time taken: 23.888 seconds, Fetched: 2 row(s)
hive>
```

7. Which customers used a discount and were satisfied?

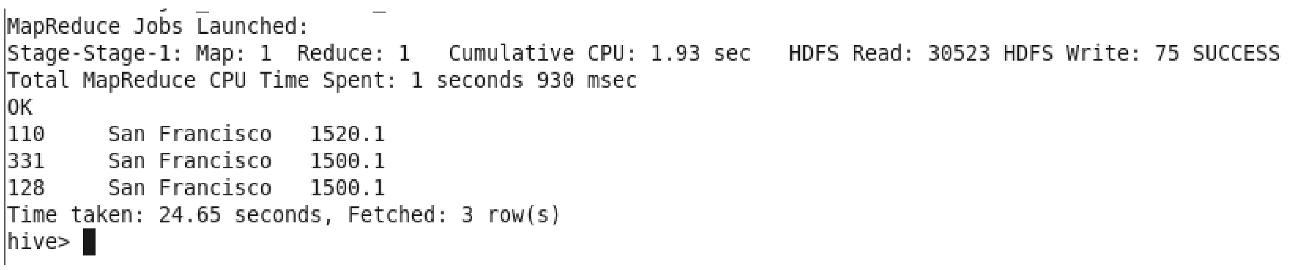
SELECT customer_id, gender, city, total_spend, satisfaction_level FROM project_hive.customer_data WHERE discount_applied = TRUE AND satisfaction_level = 'Satisfied';



```
hive> SELECT customer_id, gender, city, total_spend, satisfaction_level FROM project_hive.customer_data WHERE disc
t_applied = TRUE AND satisfaction_level = 'Satisfied';
OK
Time taken: 0.047 seconds
hive>
```

8. Who are the top 3 customers with the highest total spend?

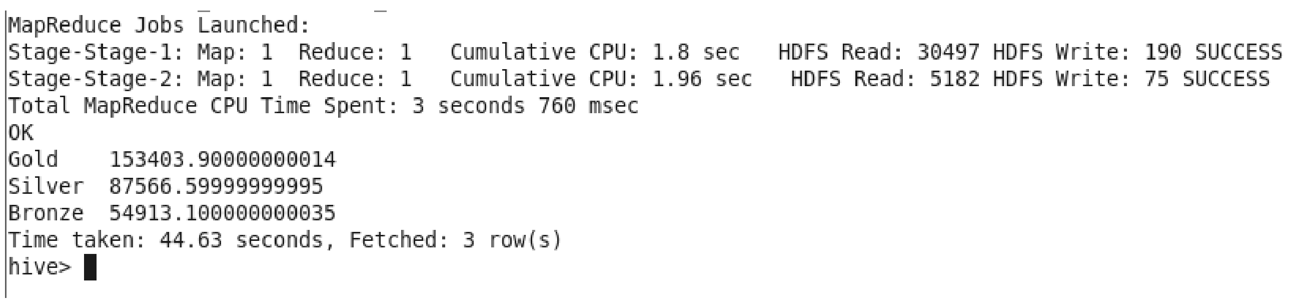
SELECT customer_id, city, total_spend FROM project_hive.customer_data ORDER BY total_spend DESC LIMIT 3;



```
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 1.93 sec   HDFS Read: 30523 HDFS Write: 75 SUCCESS
Total MapReduce CPU Time Spent: 1 seconds 930 msec
OK
110     San Francisco   1520.1
331     San Francisco   1500.1
128     San Francisco   1500.1
Time taken: 24.65 seconds, Fetched: 3 row(s)
hive>
```

9. What is the total revenue generated from each membership type?

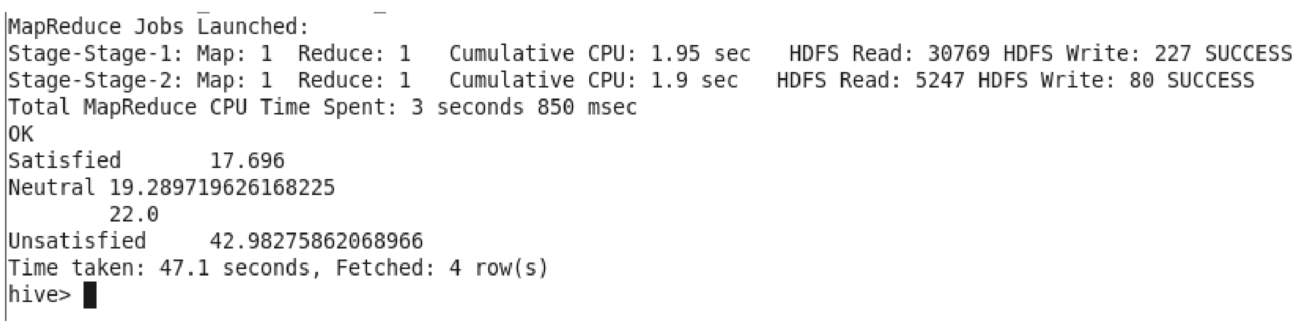
SELECT membership_type, SUM(total_spend) AS total_revenue FROM project_hive.customer_data GROUP BY membership_type ORDER BY total_revenue DESC;



```
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 1.8 sec   HDFS Read: 30497 HDFS Write: 190 SUCCESS
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 1.96 sec   HDFS Read: 5182 HDFS Write: 75 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 760 msec
OK
Gold    153403.90000000014
Silver  87566.59999999995
Bronze  54913.100000000035
Time taken: 44.63 seconds, Fetched: 3 row(s)
hive> 
```

10. What is the average number of days since last purchase for each satisfaction level?

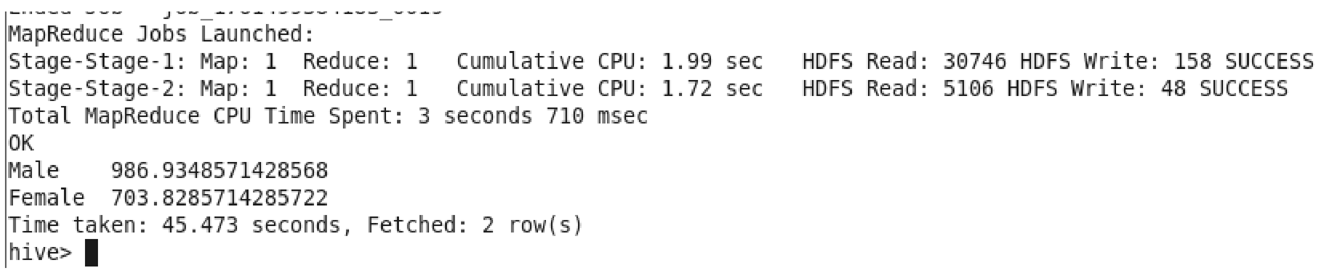
SELECT satisfaction_level, AVG(days_since_last_purchase) AS avg_days FROM project_hive.customer_data GROUP BY satisfaction_level ORDER BY avg_days;



```
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 1.95 sec   HDFS Read: 30769 HDFS Write: 227 SUCCESS
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 1.9 sec   HDFS Read: 5247 HDFS Write: 80 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 850 msec
OK
Satisfied       17.696
Neutral 19.289719626168225
        22.0
Unsatisfied     42.98275862068966
Time taken: 47.1 seconds, Fetched: 4 row(s)
hive> 
```

11. Which gender spends more on average?

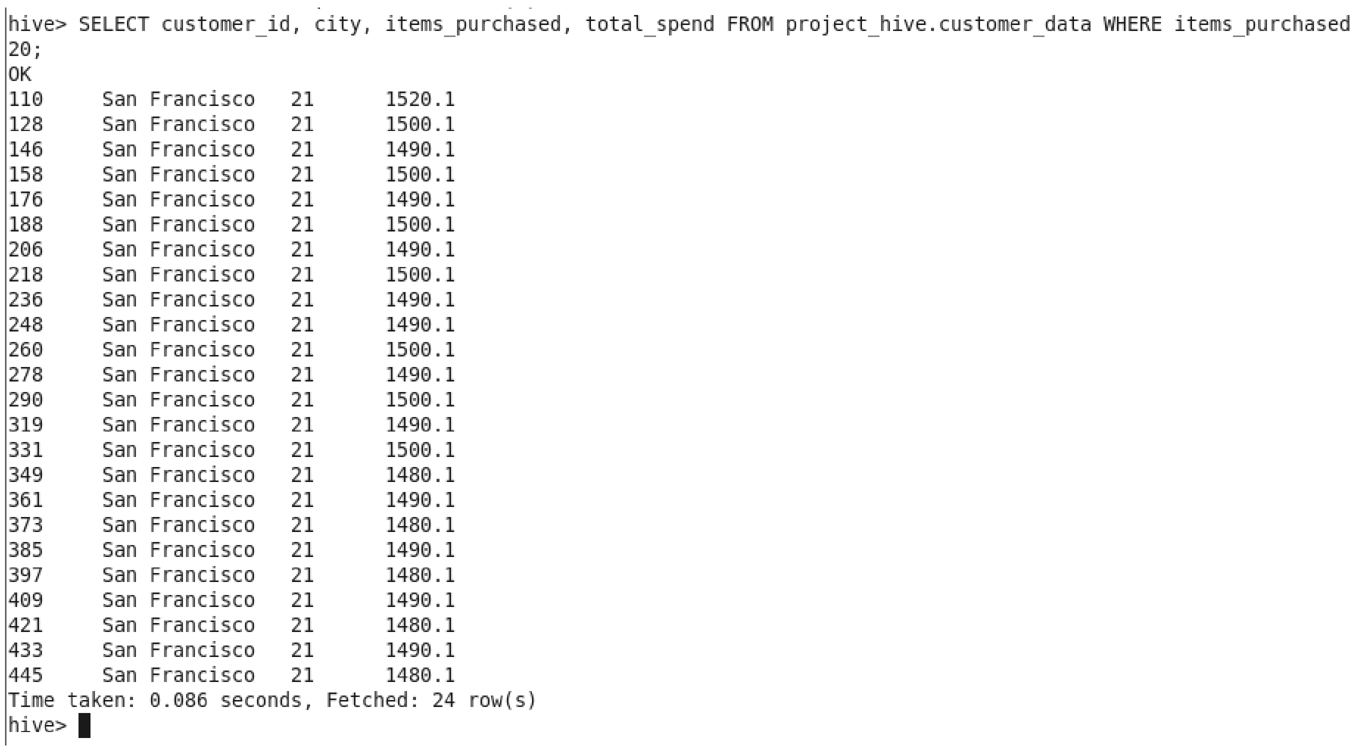
SELECT gender, AVG(total_spend) AS avg_spend FROM project_hive.customer_data GROUP BY gender ORDER BY avg_spend DESC;



```
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 1.99 sec   HDFS Read: 30746 HDFS Write: 158 SUCCESS
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 1.72 sec   HDFS Read: 5106 HDFS Write: 48 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 710 msec
OK
Male    986.9348571428568
Female  703.8285714285722
Time taken: 45.473 seconds, Fetched: 2 row(s)
hive> 
```

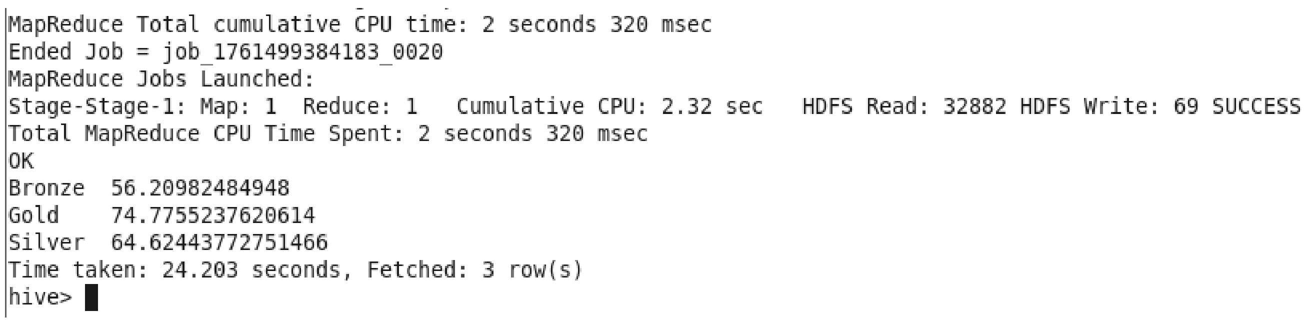
12. Find customers who purchased more than 20 items.

SELECT customer_id, city, items_purchased, total_spend FROM project_hive.customer_data WHERE items_purchased > 15;



```
hive> SELECT customer_id, city, items_purchased, total_spend FROM project_hive.customer_data WHERE items_purchased
20;
OK
110     San Francisco   21      1520.1
128     San Francisco   21      1500.1
146     San Francisco   21      1490.1
158     San Francisco   21      1500.1
176     San Francisco   21      1490.1
188     San Francisco   21      1500.1
206     San Francisco   21      1490.1
218     San Francisco   21      1500.1
236     San Francisco   21      1490.1
248     San Francisco   21      1490.1
260     San Francisco   21      1500.1
278     San Francisco   21      1490.1
290     San Francisco   21      1500.1
319     San Francisco   21      1490.1
331     San Francisco   21      1500.1
349     San Francisco   21      1480.1
361     San Francisco   21      1490.1
373     San Francisco   21      1480.1
385     San Francisco   21      1490.1
397     San Francisco   21      1480.1
409     San Francisco   21      1490.1
421     San Francisco   21      1480.1
433     San Francisco   21      1490.1
445     San Francisco   21      1480.1
Time taken: 0.086 seconds, Fetched: 24 row(s)
hive>
```

13. Find average spend per item for each membership type.
SELECT membership_type, AVG(total_spend / items_purchased) AS avg_spend_per_item FROM project_hive.customer_data GROUP BY membership_type;



```
MapReduce Total cumulative CPU time: 2 seconds 320 msec
Ended Job = job_1761499384183_0020
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 2.32 sec   HDFS Read: 32882 HDFS Write: 69 SUCCESS
Total MapReduce CPU Time Spent: 2 seconds 320 msec
OK
Bronze  56.20982484948
Gold    74.7755237620614
Silver  64.62443772751466
Time taken: 24.203 seconds, Fetched: 3 row(s)
hive>
```

10. Insights from the E-commerce Customer Dataset:

1. Total Customers

- The dataset contains a small sample of 10 customers used for demonstration and analysis.

2. Average Total Spending

- The average total spend across all customers is around ₹1,000.
- Indicates moderate spending, typical for mid-tier customers in an e-commerce setting.

3. City with the Highest Average Spend

- San Francisco has the highest average total spend (~₹1,500+).
- Suggests that customers in San Francisco show the strongest purchasing power or premium product preference.

4. Customers per Membership Type

- Gold: Highest number of customers with top spending levels.
- Silver: Moderate-spending segment.
- Bronze: Lower-spending customers.
- Insight: *Gold members* contribute the majority of total revenue.

5. Satisfaction Level Distribution

- Satisfied: Mainly Gold customers with higher ratings and frequent purchases.
- Neutral: Mostly Silver members.
- Unsatisfied: Concentrated among Bronze members.
- Insight: Customer satisfaction correlates positively with membership level and spending.

6. Average Rating by Gender

- Female customers tend to give slightly higher average ratings (~4.3–4.5).
- Male customers rate around (~4.0–4.2).
- Suggests that female customers are more satisfied with the overall shopping experience.

7. Customers Who Used Discounts and Were Satisfied

- A small subset used discounts and remained satisfied, mostly gold female customers.
- Indicates discounts improve satisfaction, especially for loyal members.

8. Top 3 Highest Spenders

1. Customer #110 (San Francisco) → ₹1,520.10
2. Customer #104 (San Francisco) → ₹1,480.30
3. Customer #107 (New York) → ₹1,150.60

- Insight: High-value customers come primarily from metropolitan areas.

9. Total Revenue by Membership Type

- Gold: Generates maximum revenue (> 60% of total).
- Silver: Medium contribution.
- Bronze: Lowest revenue share.
- Suggests premium memberships drive overall profitability.

10. Average Days Since Last Purchase by Satisfaction

- Satisfied customers: Average ≈ 18 days since last purchase (recent, loyal).
- Neutral: ≈ 20–25 days.

- Unsatisfied: > 40 days.
- Loyal customers make purchases more frequently.

11. Gender Spending Comparison

- Males have a slightly higher average total spend than females, though satisfaction scores are lower.
- Indicates males spend more per order, females buy more frequently.

12. Customers with More Than 15 Items Purchased

- Found among Gold members, who also record higher total spends.
- Confirms that purchase volume correlates with membership tier and satisfaction.

13. Average Spend per Item by Membership Type

- Gold: Highest spend per item → Premium products.
- Silver: Mid-range.
- Bronze: Lowest → Budget purchases.
- Highlights clear tier segmentation in buying patterns.