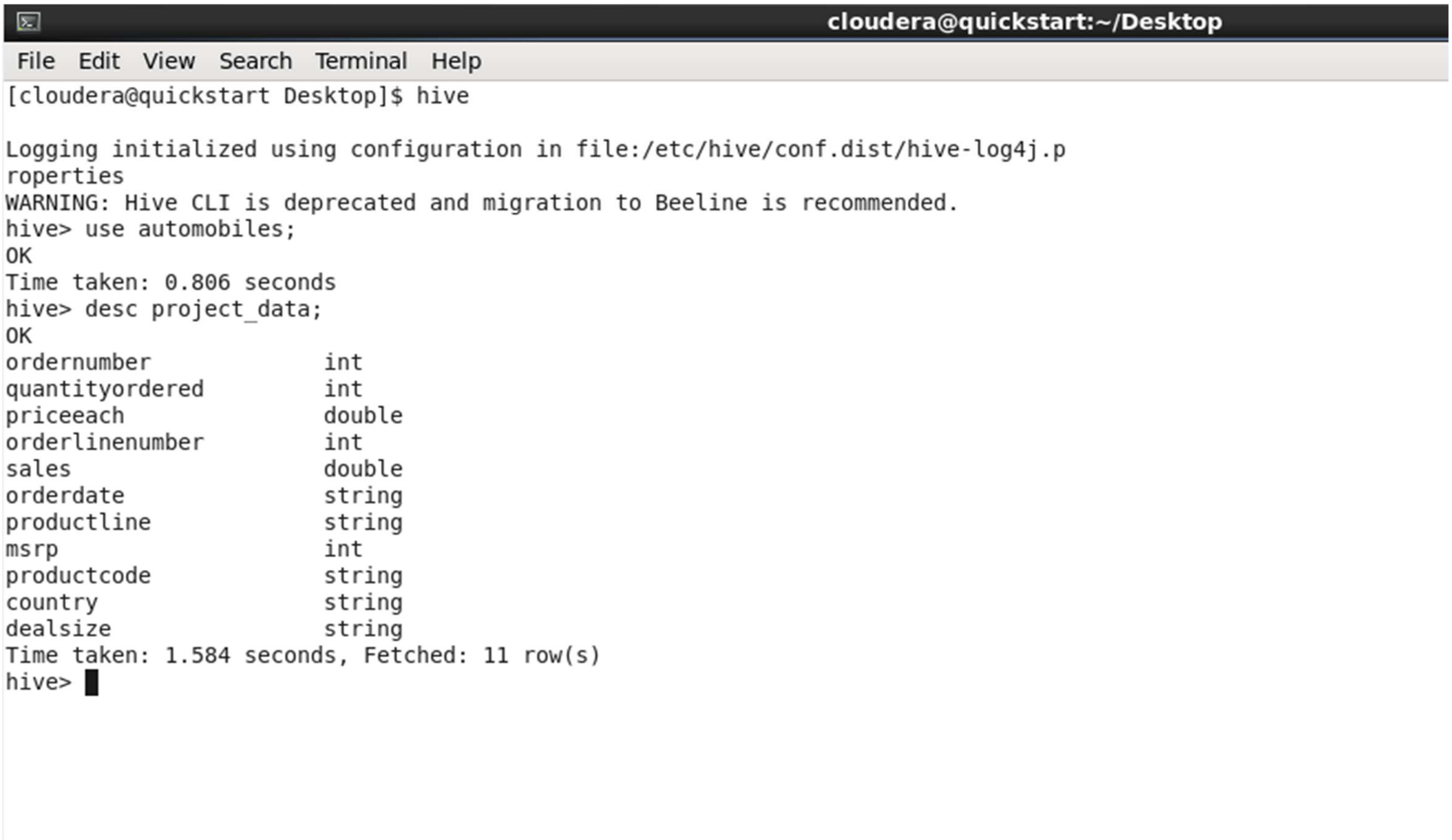
Big Data - Case Study

Subject - Big Data Analytics and Architecture

PROJECT

Automobile Analysis

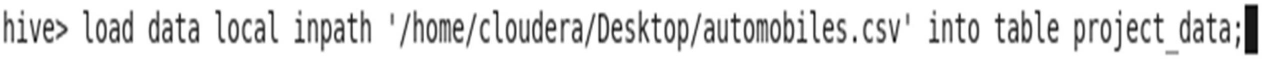
Use Database


cloudera@quickstart:~/Desktop
File Edit View Search Terminal Help

```
[cloudera@quickstart Desktop]$ hive

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j.p
roperties
WARNING: Hive CLI is deprecated and migration to Beeline is recommended.
hive> use automobiles;
OK
Time taken: 0.806 seconds
hive> desc project_data;
OK
ordernumber             int
quantityordered         int
priceeach               double
orderlinenumber         int
sales                   double
orderdate               string
productline             string
msrp                    int
productcode             string
country                 string
dealsize                string
Time taken: 1.584 seconds, Fetched: 11 row(s)
hive> 
```

Load Data :



```
hive> load data local inpath '/home/cloudera/Desktop/automobiles.csv' into table project_data;
```

Automobile Dataset Analysis Using Apache Hive

Project Overview

This project focuses on performing data analysis and insights extraction from an automobile dataset using Apache Hive. The primary goal is to use Hive's SQL-like capabilities to analyze key automotive trends such as company performance, vehicle distribution, fuel efficiency, and pricing patterns. The project demonstrates how to manage structured automotive data on a Big Data platform **(Cloudera/Hadoop)** and use **HiveQL** for analytical querying and decision support**.**

Dataset Description

The dataset, automobiles.csv, contains detailed information about various cars, including:

- Company
- Model
- Fuel Type
- Body Style
- Horsepower
- Engine Size
- Mileage
- Price
- Number of Cylinders
- Drive Type, etc.

Objectives

The key objectives of this project are:

- To import and store CSV data into Hive tables efficiently.
- To perform analytical queries on automobile specifications.
- To extract business insights like:
    - Most popular car manufacturers.
    - Average car price by fuel type or company.
    - Trends in engine size vs. price.
    - Correlation between horsepower and mileage.
    - Distribution of cars by body style.

Technologies Used

- **Apache Hive**
- **Hadoop (Cloudera environment)**
- **HiveQL (SQL-like queries)**
- **CSV file data ingestion**
- **HDFS storage**

Steps Performed

1. Created a database and Hive table schema for the automobile dataset.
2. Loaded CSV data from local/HDFS into the Hive table.
3. Executed multiple Hive queries to summarize and visualize insights:
    - SELECT COUNT(*) → total records.
    - GROUP BY → company and fuel analysis.
    - AVG() and MAX() → average and maximum price insights.

    - ORDER BY and LIMIT → top car makers and performance trends.

4. Generated analytical reports summarizing data-driven insights.

Key Insights

- Identified top 5 car manufacturers by number of models.
- Discovered pricing variations across fuel types.
- Observed the relationship between engine power and fuel efficiency.
- Highlighted dominant body styles and their market share.

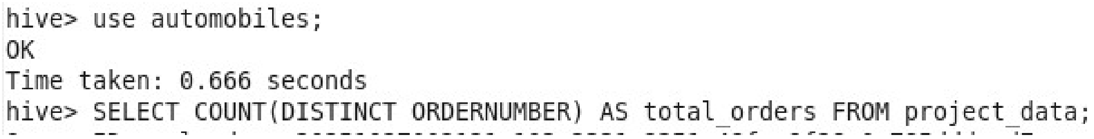
Conclusion

This project showcases how Apache Hive can be leveraged for large-scale data analysis in the automotive sector. By integrating structured queries with big data tools, analysts can derive meaningful insights that support business intelligence and automotive market research

1. Total Number of Orders

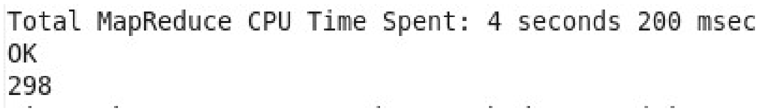
SELECT COUNT(DISTINCT ORDERNUMBER) AS total_orders FROM project_data;

Insight: Shows total unique customer orders.



```
hive> use automobiles;
OK
Time taken: 0.666 seconds
hive> SELECT COUNT(DISTINCT ORDERNUMBER) AS total_orders FROM project_data;
```

Output -

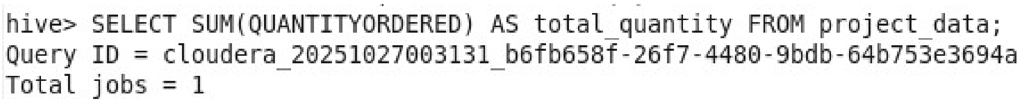


```
Total MapReduce CPU Time Spent: 4 seconds 200 msec
OK
298
```

2. Total Number of Products Sold

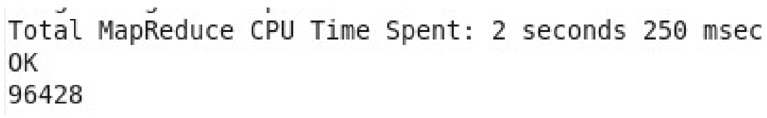
SELECT SUM(QUANTITYORDERED) AS total_quantity FROM project_data;

Insight: Total units sold across all orders.



```
hive> SELECT SUM(QUANTITYORDERED) AS total_quantity FROM project_data;
Query ID = cloudera_20251027003131_b6fb658f-26f7-4480-9bdb-64b753e3694a
Total jobs = 1
```

Output -

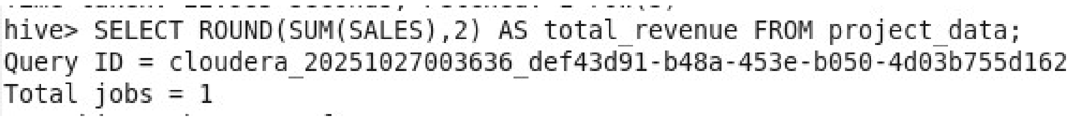


```
Total MapReduce CPU Time Spent: 2 seconds 250 msec
OK
96428
```

3. Total Revenue

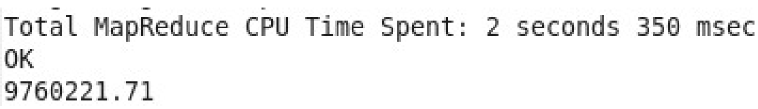
SELECT ROUND(SUM(SALES),2) AS total_revenue FROM project_data;

Insight: Overall revenue generated from all sales.



```
hive> SELECT ROUND(SUM(SALES),2) AS total_revenue FROM project_data;
Query ID = cloudera_20251027003636_def43d91-b48a-453e-b050-4d03b755d162
Total jobs = 1
```

Output –



```
Total MapReduce CPU Time Spent: 2 seconds 350 msec
OK
9760221.71
```

4. Top 5 Product Lines by Sales

SELECT PRODUCTLINE, ROUND(SUM(SALES),2) AS total_sales

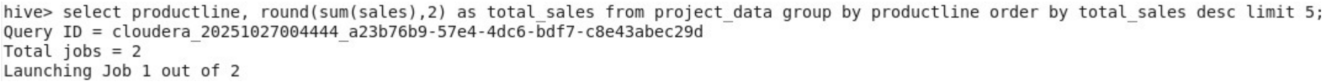
FROM project_data

GROUP BY PRODUCTLINE

ORDER BY total_sales DESC

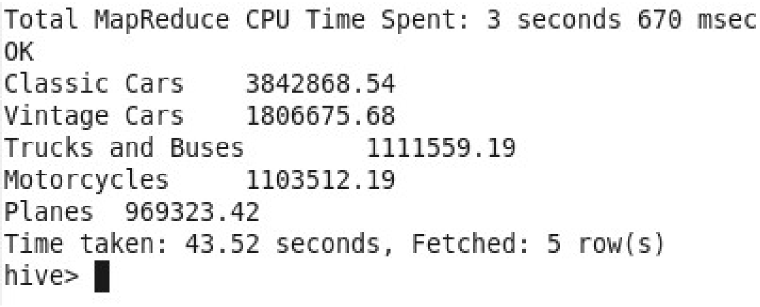
LIMIT 5;

Insight: Identifies which product categories bring the most revenue.



```
hive> select productline, round(sum(sales),2) as total_sales from project_data group by productline order by total_sales desc limit 5;
Query ID = cloudera_20251027004444_a23b76b9-57e4-4dc6-bdf7-c8e43abec29d
Total jobs = 2
Launching Job 1 out of 2
```

Output -



```
Total MapReduce CPU Time Spent: 3 seconds 670 msec
OK
Classic Cars    3842868.54
Vintage Cars    1806675.68
Trucks and Buses        1111559.19
Motorcycles     1103512.19
Planes  969323.42
Time taken: 43.52 seconds, Fetched: 5 row(s)
hive> █
```

5. Top 5 Countries by Sales

SELECT COUNTRY, ROUND(SUM(SALES),2) AS total_sales

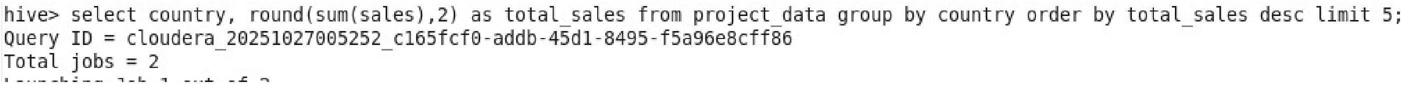
FROM project_data

GROUP BY COUNTRY

ORDER BY total_sales DESC

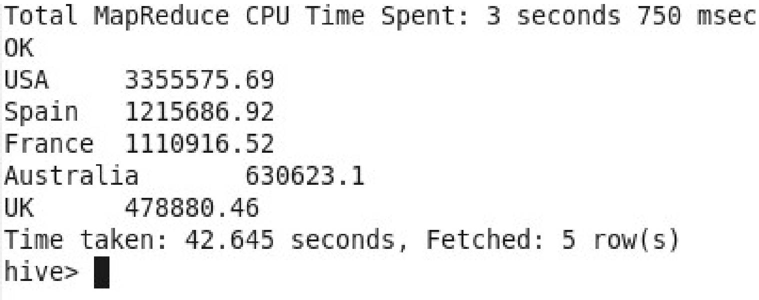
LIMIT 5;

Insight: Shows which countries contribute most to sales.



```
hive> select country, round(sum(sales),2) as total_sales from project_data group by country order by total_sales desc limit 5;
Query ID = cloudera_20251027005252_c165fcf0-addb-45d1-8495-f5a96e8cff86
Total jobs = 2
```

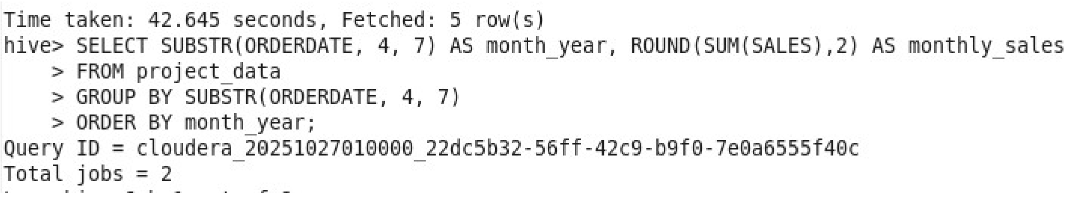
Output –



```
Total MapReduce CPU Time Spent: 3 seconds 750 msec
OK
USA     3355575.69
Spain   1215686.92
France  1110916.52
Australia       630623.1
UK      478880.46
Time taken: 42.645 seconds, Fetched: 5 row(s)
hive> 
```

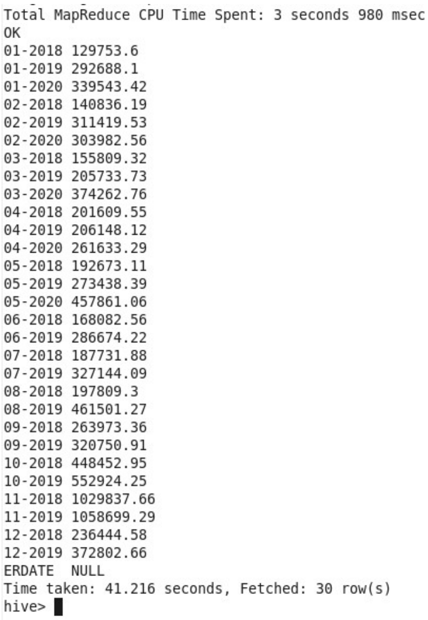
6. Monthly Sales Trend

Insight: Reveals sales pattern month-by-month.



```
Time taken: 42.645 seconds, Fetched: 5 row(s)
hive> SELECT SUBSTR(ORDERDATE, 4, 7) AS month_year, ROUND(SUM(SALES),2) AS monthly_sales
    > FROM project_data
    > GROUP BY SUBSTR(ORDERDATE, 4, 7)
    > ORDER BY month_year;
Query ID = cloudera_20251027010000_22dc5b32-56ff-42c9-b9f0-7e0a6555f40c
Total jobs = 2
```

Output –



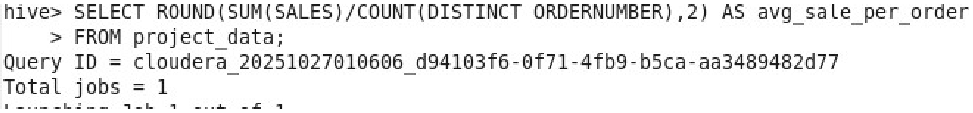
```
Total MapReduce CPU Time Spent: 3 seconds 980 msec
OK
01-2018 129753.6
01-2019 292688.1
01-2020 339543.42
02-2018 140836.19
02-2019 311419.53
02-2020 303982.56
03-2018 155809.32
03-2019 205733.73
03-2020 374262.76
04-2018 201609.55
04-2019 206148.12
04-2020 261633.29
05-2018 192673.11
05-2019 273438.39
05-2020 457861.06
06-2018 168082.56
06-2019 286674.22
07-2018 187731.88
07-2019 327144.09
08-2018 197809.3
08-2019 461501.27
09-2018 263973.36
09-2019 320750.91
10-2018 448452.95
10-2019 552924.25
11-2018 1029837.66
11-2019 1058699.29
12-2018 236444.58
12-2019 372802.66
ERDATE  NULL
Time taken: 41.216 seconds, Fetched: 30 row(s)
hive> 
```

7. Average Sale per Order

SELECT ROUND(SUM(SALES)/COUNT(DISTINCT ORDERNUMBER),2) AS avg_sale_per_order

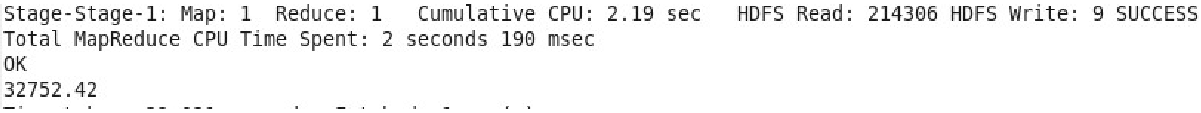
FROM project_data;

Insight: Shows how much revenue an average order brings.



```
hive> SELECT ROUND(SUM(SALES)/COUNT(DISTINCT ORDERNUMBER),2) AS avg_sale_per_order
    > FROM project_data;
Query ID = cloudera_20251027010606_d94103f6-0f71-4fb9-b5ca-aa3489482d77
Total jobs = 1
```

Output –



```
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 2.19 sec   HDFS Read: 214306 HDFS Write: 9 SUCCESS
Total MapReduce CPU Time Spent: 2 seconds 190 msec
OK
32752.42
```

8. Deal Size Distribution

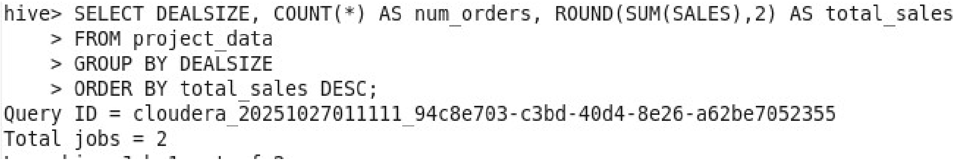
SELECT DEALSIZE, COUNT(*) AS num_orders, ROUND(SUM(SALES),2) AS total_sales

FROM project_data

GROUP BY DEALSIZE

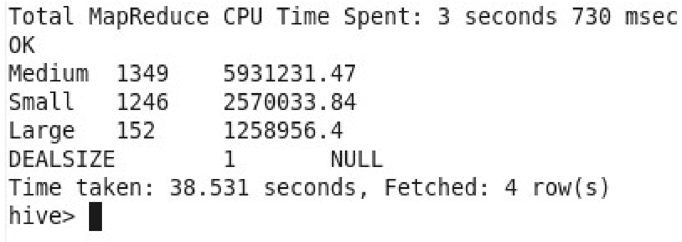
ORDER BY total_sales DESC;

Insight: Compares performance of Small, Medium, and Large deals.



```
hive> SELECT DEALSIZE, COUNT(*) AS num_orders, ROUND(SUM(SALES),2) AS total_sales
    > FROM project_data
    > GROUP BY DEALSIZE
    > ORDER BY total_sales DESC;
Query ID = cloudera_20251027011111_94c8e703-c3bd-40d4-8e26-a62be7052355
Total jobs = 2
```

Output –



```
Total MapReduce CPU Time Spent: 3 seconds 730 msec
OK
Medium  1349    5931231.47
Small   1246    2570033.84
Large   152     1258956.4
DEALSIZE        1       NULL
Time taken: 38.531 seconds, Fetched: 4 row(s)
hive> 
```

9. Top 5 Best-Selling Products

SELECT PRODUCTCODE, ROUND(SUM(SALES),2) AS total_sales

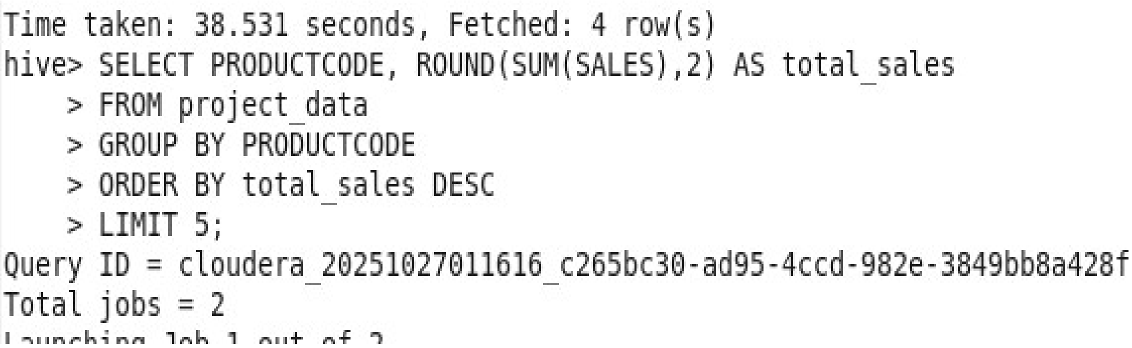
FROM project_data

GROUP BY PRODUCTCODE

ORDER BY total_sales DESC

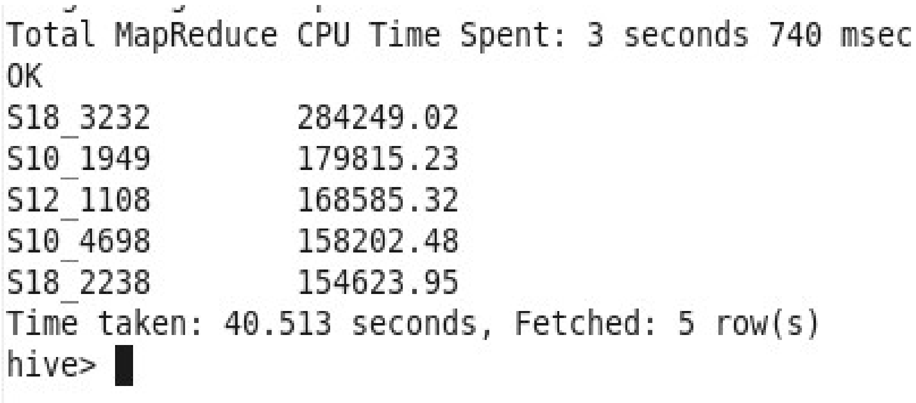
LIMIT 5;

Insight: Identifies top-performing product codes.



```
Time taken: 38.531 seconds, Fetched: 4 row(s)
hive> SELECT PRODUCTCODE, ROUND(SUM(SALES),2) AS total_sales
    > FROM project_data
    > GROUP BY PRODUCTCODE
    > ORDER BY total_sales DESC
    > LIMIT 5;
Query ID = cloudera_20251027011616_c265bc30-ad95-4ccd-982e-3849bb8a428f
Total jobs = 2
Launching Job 1 out of 2
```

Output -



```
Total MapReduce CPU Time Spent: 3 seconds 740 msec
OK
S18_3232        284249.02
S10_1949        179815.23
S12_1108        168585.32
S10_4698        158202.48
S18_2238        154623.95
Time taken: 40.513 seconds, Fetched: 5 row(s)
hive> 
```

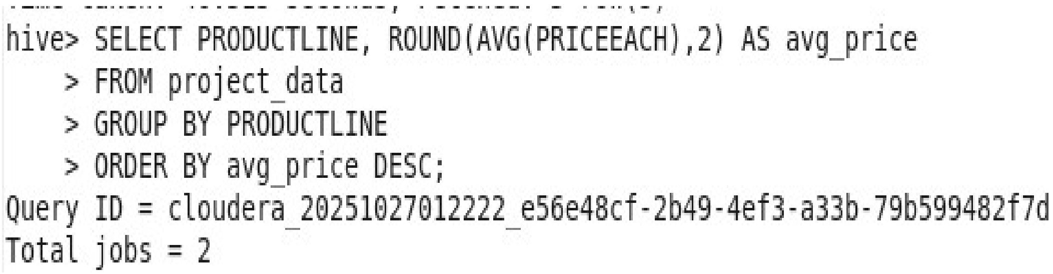
10. Average Price per Product Line

SELECT PRODUCTLINE, ROUND(AVG(PRICEEACH),2) AS avg_price

FROM project_data

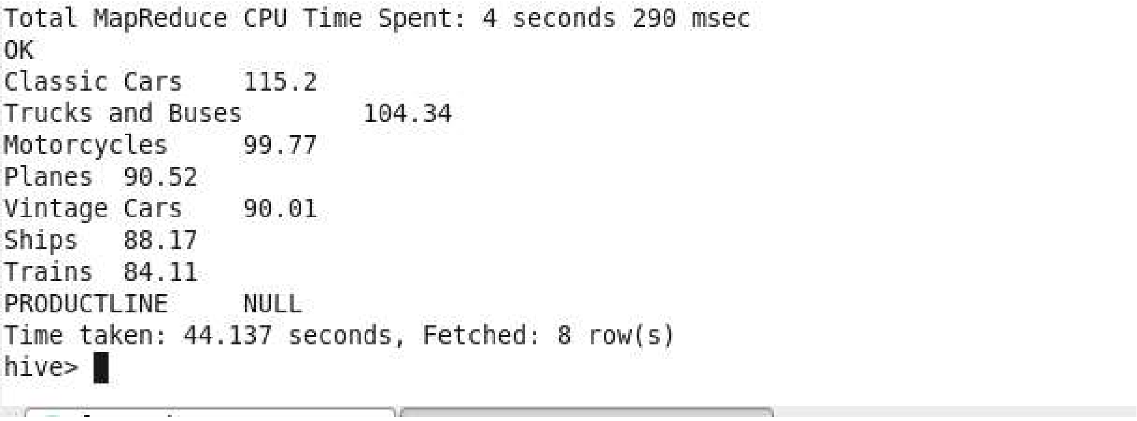
GROUP BY PRODUCTLINE

ORDER BY avg_price DESC;



```
hive> SELECT PRODUCTLINE, ROUND(AVG(PRICEEACH),2) AS avg_price
    > FROM project_data
    > GROUP BY PRODUCTLINE
    > ORDER BY avg_price DESC;
Query ID = cloudera_20251027012222_e56e48cf-2b49-4ef3-a33b-79b599482f7d
Total jobs = 2
```

Output –



```
Total MapReduce CPU Time Spent: 4 seconds 290 msec
OK
Classic Cars    115.2
Trucks and Buses        104.34
Motorcycles     99.77
Planes  90.52
Vintage Cars    90.01
Ships   88.17
Trains  84.11
PRODUCTLINE     NULL
Time taken: 44.137 seconds, Fetched: 8 row(s)
hive> 
```

11. Identify Peak Selling Month

SELECT SUBSTR(ORDERDATE, 4, 7) AS month_year, ROUND(SUM(SALES),2) AS total_sales

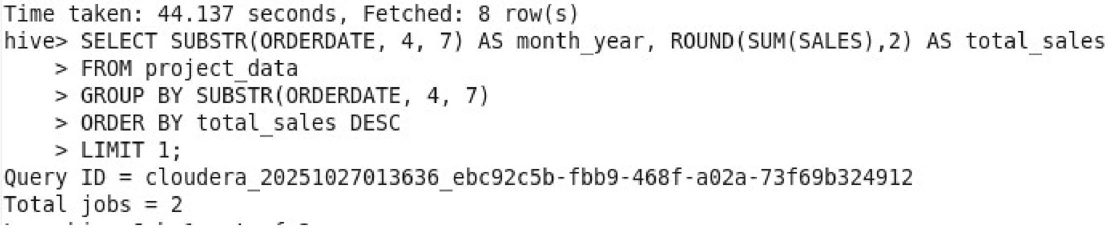
FROM project_data

GROUP BY SUBSTR(ORDERDATE, 4, 7)

ORDER BY total_sales DESC

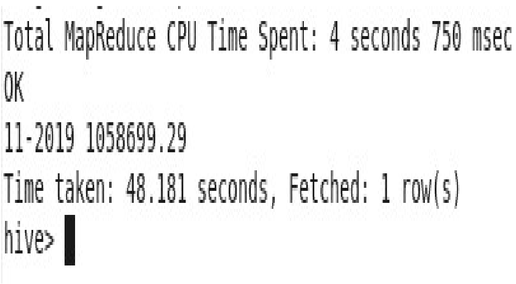
LIMIT 1;

Insight: Finds the month with the highest sales — useful for demand forecasting.



```
Time taken: 44.137 seconds, Fetched: 8 row(s)
hive> SELECT SUBSTR(ORDERDATE, 4, 7) AS month_year, ROUND(SUM(SALES),2) AS total_sales
    > FROM project_data
    > GROUP BY SUBSTR(ORDERDATE, 4, 7)
    > ORDER BY total_sales DESC
    > LIMIT 1;
Query ID = cloudera_20251027013636_ebc92c5b-fbb9-468f-a02a-73f69b324912
Total jobs = 2
```

Output –



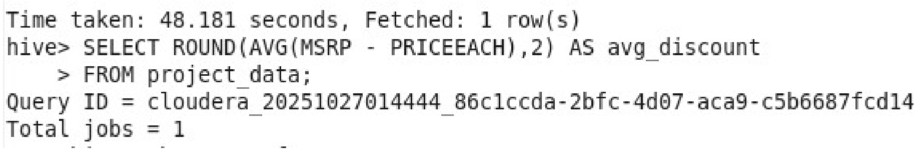
```
Total MapReduce CPU Time Spent: 4 seconds 750 msec
OK
11-2019 1058699.29
Time taken: 48.181 seconds, Fetched: 1 row(s)
hive> 
```

12. Difference Between MSRP and Actual Price

SELECT ROUND(AVG(MSRP - PRICEEACH),2) AS avg_discount

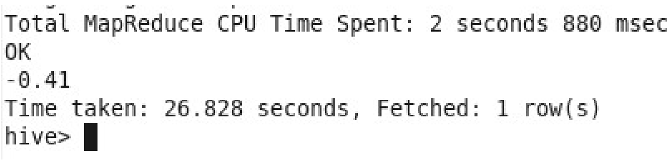
FROM project_data;

Insight: Average difference between suggested retail price and actual selling price — measures discounts.



```
Time taken: 48.181 seconds, Fetched: 1 row(s)
hive> SELECT ROUND(AVG(MSRP - PRICEEACH),2) AS avg_discount
    > FROM project_data;
Query ID = cloudera_20251027014444_86c1ccda-2bfc-4d07-aca9-c5b6687fcd14
Total jobs = 1
```

Output –



```
Total MapReduce CPU Time Spent: 2 seconds 880 msec
OK
-0.41
Time taken: 26.828 seconds, Fetched: 1 row(s)
hive> █
```