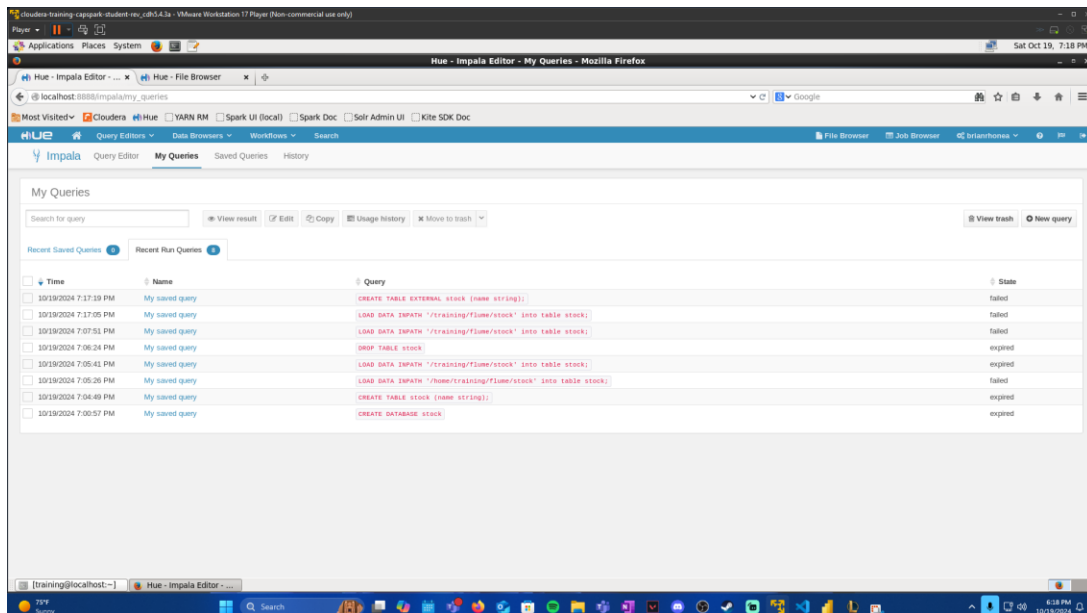


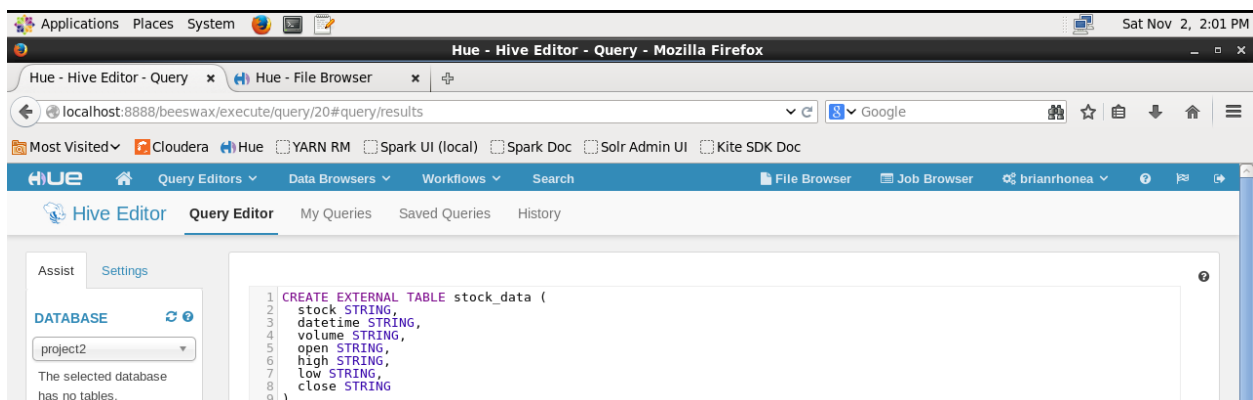
Project 2 Part B:



This was on October 19th. I created a table and imported the data; however, it did not import how I wanted it to. Without thinking about the consequences, I dropped the table and lost all data. I tried to find it in HDFS trash to restore but could not. So, for this reason, I will be gathering and using data from October 20th to October 30th for this project. The dates will not be matching from Project 1. The dates will not be 5 consecutive days because there were days, I was unable to run my script.

Project 2: Part B

Create table in Hive to import data from file. With sample data shown as well.



Metastore Manager

Databases > project2 > stock_data

Columns | Sample | Properties

| | stock | datetime | volume | open | high | low | close |
|----|-------|---------------------|--------|-----------|-----------|-----------|-----------|
| 0 | stock | datetime | volume | open | high | low | close |
| 1 | AAPL | 2024-10-21 12:28:00 | 484 | 236.27000 | 236.27000 | 236.26000 | 236.26000 |
| 2 | IBM | 2024-10-21 12:25:00 | 120 | 231.53000 | 231.53000 | 231.53000 | 231.53000 |
| 3 | GOOGL | 2024-10-21 12:28:00 | 346 | 163.53500 | 163.53500 | 163.50000 | 163.50000 |
| 4 | MSFT | 2024-10-21 12:27:00 | 456 | 415.16000 | 415.16000 | 415.06000 | 415.06000 |
| 5 | TSLA | 2024-10-21 12:28:00 | 732 | 218.56000 | 218.58000 | 218.52000 | 218.58000 |
| 6 | AAPL | 2024-10-22 14:54:00 | 1097 | 235.32500 | 235.35000 | 235.32000 | 235.32500 |
| 7 | IBM | 2024-10-22 14:53:00 | 333 | 231.78000 | 231.78000 | 231.77000 | 231.77000 |
| 8 | GOOGL | 2024-10-22 14:54:00 | 102 | 164.69000 | 164.69000 | 164.69000 | 164.69000 |
| 9 | MSFT | 2024-10-22 14:54:00 | 590 | 429.27000 | 429.27000 | 429.19000 | 429.19000 |
| 10 | TSLA | 2024-10-22 14:54:00 | 441 | 217.26000 | 217.33500 | 217.26000 | 217.32500 |

a. How many records are there in the table?

SELECT COUNT(*) from stock_data; There is 3111 records in the table.

Hive Editor - Query

1 SELECT COUNT(*) from stock_data;

Execute Save as... Explain or create a New query

Recent queries Query Log Columns Results Chart

| | _c0 |
|---|------|
| 0 | 3111 |

- b. How many different days are there in the table?

```
SELECT COUNT(DISTINCT from_unixtime(unix_timestamp(datetime, 'yyyy-MM-dd HH:mm:ss'), 'yyyy-MM-dd')) AS unique_days  
FROM stock_data; There are 5 unique days.
```

The screenshot shows a SQL query editor with the following query:

```
1 SELECT COUNT(DISTINCT from_unixtime(unix_timestamp(datetime, 'yyyy-MM-dd HH:mm:ss')  
2 FROM stock_data;
```

Below the query editor are buttons: **Execute**, **Save as...**, **Explain**, **or create a**, and **New query**.

The results tab is active, showing a table with one column: **unique_days**. The table has one row with the value **5**.

| unique_days |
|-------------|
| 5 |

- c. How many records per each day are there in the table?

```
SELECT from_unixtime(unix_timestamp(datetime, 'yyyy-MM-dd HH:mm:ss'), 'yyyy-MM-dd') AS day, COUNT(*) AS records_per_day FROM stock_data GROUP BY  
from_unixtime(unix_timestamp(datetime, 'yyyy-MM-dd HH:mm:ss'), 'yyyy-MM-dd')  
ORDER BY day; There is some discrepancy in the days because I had forgotten to run  
the script some days.
```

The screenshot shows a SQL query editor with the following query:

```
1 SELECT from_unixtime(unix_timestamp(datetime, 'yyyy-MM-dd HH:mm:ss'), 'yyyy-MM-dd')  
2 FROM stock_data  
3 GROUP BY from_unixtime(unix_timestamp(datetime, 'yyyy-MM-dd HH:mm:ss'), 'yyyy-MM-dd')  
4 ORDER BY day;
```

Below the query editor are buttons: **Execute**, **Save as...**, **Explain**, **or create a**, and **New query**.

The results tab is active, showing a table with two columns: **day** and **records_per_day**. The table has six rows.

| | day | records_per_day |
|---|------------|-----------------|
| 0 | NULL | 1 |
| 1 | 2024-10-21 | 800 |
| 2 | 2024-10-22 | 800 |
| 3 | 2024-10-23 | 635 |
| 4 | 2024-10-24 | 800 |
| 5 | 2024-10-30 | 75 |

- d. What are the symbols in the table?

```
SELECT DISTINCT stock AS unique_symbols  
FROM stock_data;
```

1 **SELECT DISTINCT** stock **AS** unique_symbols
2 **FROM** stock_data;

Execute

Save as...

Explain

or create a

New query

Recent queries

Query

Log

Columns

Results

Chart

| | unique_symbols |
|---|----------------|
| 0 | AAPL |
| 1 | GOOGL |
| 2 | IBM |
| 3 | MSFT |
| 4 | TSLA |
| 5 | stock |

- e. What is the highest price for each symbol?

```
SELECT stock, MAX(CAST(high AS DOUBLE)) AS highest_price  
FROM stock_data GROUP BY stock;
```

1 **SELECT** stock, **MAX**(**CAST**(high **AS** DOUBLE)) **AS** highest_price
2 **FROM** stock_data
3 **GROUP BY** stock;

Execute

Save as...

Explain

or create a

New query

Recent queries

Query

Log

Columns

Results

Chart

| | stock | highest_price |
|---|-------|--------------------|
| 0 | AAPL | 236.80000000000001 |
| 1 | GOOGL | 175.66 |
| 2 | IBM | 233.32499999999999 |
| 3 | MSFT | 434.91000000000003 |
| 4 | TSLA | 259.43000000000001 |
| 5 | stock | NULL |

- f. What is the lowest price for each symbol?

```
SELECT stock, MIN(CAST(low AS DOUBLE)) AS lowest_price  
FROM stock_data
```

GROUP BY stock;

```
1 SELECT stock, MIN(CAST(low AS DOUBLE)) AS lowest_price
2 FROM stock_data
3 GROUP BY stock;
```

Execute Save as... Explain or create a New query

Recent queries Query Log Columns Results Chart

| | stock | lowest_price |
|---|-------|--------------------|
| 0 | AAPL | 228.06 |
| 1 | GOOGL | 161.92500000000001 |
| 2 | IBM | 204.50999999999999 |
| 3 | MSFT | 414.92000000000002 |
| 4 | TSLA | 212.40000000000001 |
| 5 | stock | NULL |

- g. What is the average price for each symbol?

```
SELECT stock, AVG(CAST(close AS DOUBLE)) AS average_price
FROM stock_data
GROUP BY stock;
```

```
1 SELECT stock, AVG(CAST(close AS DOUBLE)) AS average_price
2 FROM stock_data
3 GROUP BY stock;
4
```

Execute Save as... Explain or create a New query

Recent queries Query Log Columns Results Chart

| | stock | average_price |
|---|-------|--------------------|
| 0 | AAPL | 233.31817524115741 |
| 1 | GOOGL | 163.89336816720268 |
| 2 | IBM | 227.56491157556266 |
| 3 | MSFT | 424.2107717041801 |
| 4 | TSLA | 227.69815916398719 |
| 5 | stock | NULL |

- h. What is the range of price for each symbol?

```
SELECT stock, (MAX(CAST(high AS DOUBLE)) - MIN(CAST(low AS DOUBLE))) AS
price_range
FROM stock_data
GROUP BY stock;
```

```

1 SELECT stock, (MAX(CAST(high AS DOUBLE)) - MIN(CAST(low AS DOUBLE))) AS price_rang
2 FROM stock_data
3 GROUP BY stock;
4

```

Execute

Save as...

Explain

or create a

New query

Recent queries

Query

Log

Columns

Results

Chart

| | stock | price_range |
|---|-------|--------------------|
| 0 | AAPL | 8.7400000000000091 |
| 1 | GOOGL | 13.734999999999985 |
| 2 | IBM | 28.814999999999998 |
| 3 | MSFT | 19.990000000000009 |
| 4 | TSLA | 47.030000000000001 |
| 5 | stock | NULL |

i. What is the date on which each symbol experienced the highest price?

SELECT sd.stock,

sd.datetime,

CAST(sd.high AS DOUBLE) AS high_price

FROM stock_data sd

JOIN (

SELECT stock, MAX(CAST(high AS DOUBLE)) AS max_high

FROM stock_data

GROUP BY stock

) max_price ON sd.stock = max_price.stock AND CAST(sd.high AS DOUBLE) = max_price.max_high;

```

1 SELECT sd.stock,
2       sd.datetime,
3       CAST(sd.high AS DOUBLE) AS high_price
4 FROM stock_data sd
5 JOIN (
6       SELECT stock, MAX(CAST(high AS DOUBLE)) AS max_high
7       FROM stock_data
8       GROUP BY stock
9 ) max_price ON sd.stock = max_price.stock AND CAST(sd.high AS DOUBLE) = max_price.max_high;
10

```

[Execute](#)
[Save as...](#)
[Explain](#)
[or create a](#)
[New query](#)

[Recent queries](#)
[Query](#)
[Log](#)
[Columns](#)
[Results](#)
[Chart](#)

| | sd.stock | sd.datetime | high_price |
|---|----------|---------------------|--------------------|
| 0 | AAPL | 2024-10-21 14:04:00 | 236.80000000000001 |
| 1 | GOOGL | 2024-10-30 15:44:00 | 175.66 |
| 2 | GOOGL | 2024-10-30 15:43:00 | 175.66 |
| 3 | IBM | 2024-10-23 15:56:00 | 233.32499999999999 |
| 4 | IBM | 2024-10-23 15:56:00 | 233.32499999999999 |
| 5 | MSFT | 2024-10-30 15:47:00 | 434.91000000000003 |
| 6 | MSFT | 2024-10-30 15:46:00 | 434.91000000000003 |
| 7 | TSLA | 2024-10-30 15:49:00 | 259.43000000000001 |
| 8 | TSLA | 2024-10-30 15:45:00 | 259.43000000000001 |