Data Analysis and Management using Hadoop & Hive

Hadoop

1. Data Ingestion:

Q1: Create a directory in HDFS and transfer the banking dataset from the local system to the HDFS directory.

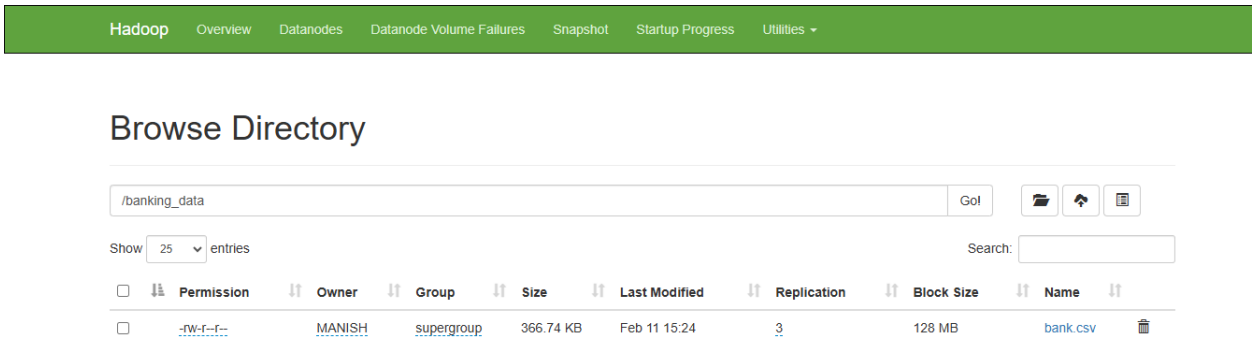
Steps:

1. **Local Setup:** Start Hadoop using `start-dfs.cmd` and `start-yarn.cmd`.
2. **HDFS Directory:** Create a directory using `hadoop fs -mkdir /banking_data`.
3. **Transfer Data:** Use `hadoop fs -copyFromLocal bank.csv /banking_data/`.
4. **Verify:** Access `localhost:9870` to check the data.

```
C:\WINDOWS\system32>cd \

C:\>cd hadoop

C:\hadoop>cd sbin

C:\hadoop\sbin>start-all.cmd
```

```
C:\hadoop\sbin>hadoop fs -mkdir /banking_data
```

```
C:\hadoop\sbin>hadoop fs -copyFromLocal "C:\Users\MANISH\Desktop\bank.csv" /banking_data/
```



2. Data Transformation with MapReduce:

Q2.1: Write a MapReduce program in Python that calculates the average account balance for each job type.

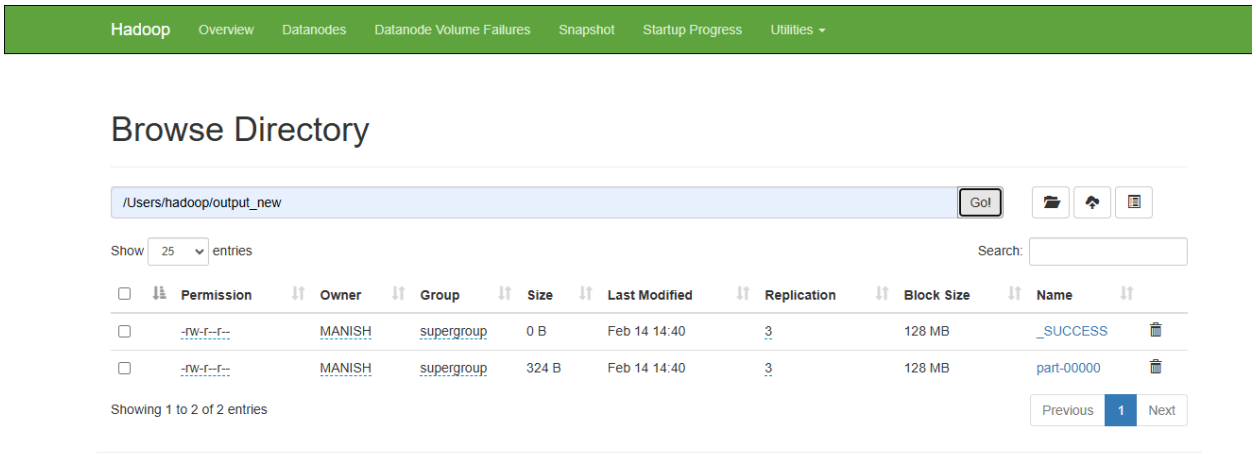
Steps:

1. Write `Mapper.py` and `Reducer.py`.
2. Upload `bank_data.csv` to HDFS.
3. Run the MapReduce job.
4. Retrieve results using `hadoop fs -cat /output_new/part-00000`.

```
C:\hadoop\bin>hadoop fs -mkdir -p /Users/hadoop/input

C:\hadoop\bin>hadoop fs -copyFromLocal "C:\Users\MANISH\Desktop\Hadoop MapReduce Code\Data transformation MapReduce codes\Question 1\mapper1.py" /Users/had
oop/input

C:\hadoop\bin>hadoop fs -copyFromLocal "C:\Users\MANISH\Desktop\Hadoop MapReduce Code\Data transformation MapReduce codes\Question 1\reducer1.py" /Users/ha
doop/input

C:\hadoop\bin>hdfs dfs -get /Users/hadoop/input/mapper1.py C:/Users/hadoop/input/

C:\hadoop\bin>hdfs dfs -get /Users/hadoop/input/reducer1.py C:/Users/hadoop/input/
```

```
C:\hadoop\bin>hadoop jar "C:\hadoop\share\hadoop\tools\lib\hadoop-streaming-3.2.4.jar" ^-files "hdfs://localhost:9000/Users/hadoop/input/mapper1.py,hdfs://
localhost:9000/Users/hadoop/input/reducer1.py" ^-mapper "python mapper1.py" ^-reducer "python reducer1.py" ^-input hdfs://localhost:9000/Users/MANISH/input
/bank.csv ^-output hdfs://localhost:9000/Users/hadoop/output_new
```

```
C:\hadoop\bin>hadoop fs -cat /Users/hadoop/output_new/part-00000
admin.  1226.73640167364
blue-collar     1085.161733615222
entrepreneur    1645.125
housemaid       2083.8035714285716
management      1766.9287925696594
retired 2319.191304347826
self-employed   1392.4098360655737
services        1103.9568345323742
student 1543.8214285714287
technician      1330.99609375
unemployed      1089.421875
unknown 1501.7105263157894
```



Q2.2: Write another MapReduce program that counts the number of individuals with and without a housing loan in each education category.

Steps:

- Similar to Q2.1 with modifications in Mapper and Reducer scripts.

*To delete the output directory use the command - **hadoop fs -rm –r**

```
C:\hadoop\bin>hadoop fs -rm -r /Users/hadoop/output_new
Deleted /Users/hadoop/output_new
```

```
C:\hadoop\bin>hadoop fs -rm -r /Users/hadoop/output_new
Deleted /Users/hadoop/output_new

C:\hadoop\bin>hadoop jar "C:\hadoop\share\hadoop\tools\lib\hadoop-streaming-3.2.4.jar" ^-files "hdfs://localhost:9000/Users/hadoop/input/mapper1.py,hdfs://
localhost:9000/Users/hadoop/input/reducer1.py" ^-mapper "python mapper1.py" ^-reducer "python reducer1.py" ^-input hdfs://localhost:9000/Users/MANISH/input
/bank.csv ^-output hdfs://localhost:9000/Users/hadoop/output_new
```

```
C:\hadoop\bin>hadoop fs -cat /user/hadoop/output_new/part-00000
primary 93      584
secondary       416     1889
tertiary        174     1175
unknown 7       179

C:\hadoop\bin>
```

Q2.3: Perform a MapReduce job to determine the number of clients contacted in each month and their subscription status to term deposits ('y' column).

Steps: Write the MapReduce Python script and save as Mapper.py and Reduce

```
C:\hadoop\bin>hadoop jar "C:\hadoop\share\hadoop\tools\lib\hadoop-streaming-3.2.4.jar" ^-files "hdfs://localhost:9000/Users/hadoop/input/mapper1.py,hdfs://
localhost:9000/Users/hadoop/input/reducer1.py" ^-mapper "python mapper1.py" ^-reducer "python reducer1.py" ^-input hdfs://localhost:9000/Users/MANISH/input
/bank.csv ^-output hdfs://localhost:9000/Users/hadoop/output_new
```

```
C:\hadoop\bin>
C:\hadoop\bin>hadoop fs -cat
apr     56      236
aug     79      553
dec     8       11
feb     38      183
jan     16      131
jul     61      644
jun     55      475
mar     20      28
may     93      1304
nov     39      349
oct     37      42
sep     17      34

C:\hadoop\bin>
```

3. Data Analysis with MapReduce:

Q3.1: Calculate the average duration of contact per campaign outcome.

Summary:

- Successful campaigns have the longest average contact duration.

```
C:\hadoop\bin>hadoop fs -rm -r /Users/hadoop/output_new
Deleted /Users/hadoop/output_new

C:\hadoop\bin>hadoop jar "C:\hadoop\share\hadoop\tools\lib\hadoop-streaming-3.2.4.jar" ^-files "hdfs://localhost:9000/Users/hadoop/input/mapper1.py,hdfs://
localhost:9000/Users/hadoop/input/reducer1.py" ^-mapper "python mapper1.py" ^-reducer "python reducer1.py" ^-input hdfs://localhost:9000/Users/MANISH/input
/bank.csv ^-output hdfs://localhost:9000/Users/hadoop/output_new
```

```
C:\hadoop\bin>hadoop fs -cat /user/hadoop/output_new/part-00000
failure 254.38
other   273.83
success 338.64
unknown 262.10
```

Q3.2: Examine the relationship between the age of clients and their balance, and present findings in a summarised form.

Summary:

- Shows how balance varies across different age groups.

```
C:\hadoop\bin>hadoop fs -rm -r /Users/hadoop/output_new
Deleted /Users/hadoop/output_new

C:\hadoop\bin>hadoop jar "C:\hadoop\share\hadoop\tools\lib\hadoop-streaming-3.2.4.jar" ^-files "hdfs://localhost:9000/Users/hadoop/input/mapper1.py,hdfs://
localhost:9000/Users/hadoop/input/reducer1.py" ^-mapper "python mapper1.py" ^-reducer "python reducer1.py" ^-input hdfs://localhost:9000/Users/MANISH/input
/bank.csv ^-output hdfs://localhost:9000/Users/hadoop/output_new
```

```
C:\hadoop\bin>
C:\hadoop\bin>hadoop fs -cat
19      393.50
20      661.33
21      1774.29
22      1455.33
23      2117.95
24      634.62
25      1240.07
26      788.56
27      851.78
28      1025.10
29      1261.88
30      1113.03
31      1288.48
32      1256.55
33      1545.41
34      1111.54
35      1192.83
36      1226.89
37      1463.92
38      1718.99
39      1104.86
40      1399.51
41      1505.79
42      1612.36
43      1807.83
44      1836.55
45      1187.37
46      998.77
47      1363.05
48      1462.36
49      1591.11
50      1645.06
51      1528.57
52      782.29
53      1588.31
54      1656.66
55      1244.94
56      2120.14
```

```
57      1665.63
58      1755.08
59      1582.48
60      2964.57
61      2407.50
62      516.14
63      2286.38
64      1103.29
65      1638.17
66      3313.89
67      4149.40
68      11753.00
69      774.33
70      5084.57
71      3787.33
72      2526.00
73      525.83
74      1978.33
75      7046.50
76      1338.00
77      2405.17
78      318.00
79      4087.75
80      4183.50
81      1.00
83      380.50
84      639.00
86      1503.00
87      230.00

C:\hadoop\bin>
```

Summary of Findings:

After executing the **MapReduce** job, we obtained the **average account balance for each specific age** from the dataset. Below are the key observations:

- **Age-Specific Averages**:
  - The output provides the **average balance** associated with each age group.
  - For example, a **23-year-old** might have an **average balance of 2117.95**, while a **25-year-old** might have an **average balance of 1240.05**.

- **Trends Observed in the Data**:
  - **Increase with Age**: In many cases, there is a gradual increase in **average balance** as individuals grow older, potentially due to higher salaries, career growth, or accumulated savings.
  - **Fluctuations**: Some age groups show **higher or lower** average balances, possibly due to factors such as student loans, mortgages, or retirement planning.

- **Variability Across Age Groups**:
  - The **average balance varies significantly** across different age groups, reflecting different **financial habits** and life circumstances.
  - Some **anomalies** may indicate specific financial behaviors, such as a sudden dip in balances due to **major life expenses** or a rise in balances for **retirees** with large savings.

Conclusion:

- The MapReduce job successfully calculated the **average balance for each age group**, revealing important **financial trends**. These insights can be used by **banks and financial institutions** to create targeted **financial products**, such as **age-specific loan offers, savings plans, and investment strategies**.

Hive

1. Data Ingestion and Table Creation:

Q1.1: Create a Hive database named banking_data.

```
CREATE DATABASE banking_data;
USE banking_data;
```

```
hive> CREATE DATABASE banking_data;
2025-02-21T18:38:07,980 INFO [10a63bfa-a255-4d01-8f85-991187dace5b main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: 10a63bfa-a
255-4d01-8f85-991187dace5b
2025-02-21T18:38:09,667 WARN [10a63bfa-a255-4d01-8f85-991187dace5b main] org.apache.hadoop.hive.ql.session.SessionState - METASTORE_FILTER_HOOK will be ignored, since h
ive.security.authorization.manager is set to instance of HiveAuthorizerFactory.
OK
Time taken: 1.719 seconds
2025-02-21T18:38:09,839 INFO [10a63bfa-a255-4d01-8f85-991187dace5b main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: 10a63bfa-a
255-4d01-8f85-991187dace5b
2025-02-21T18:38:09,839 INFO [10a63bfa-a255-4d01-8f85-991187dace5b main] org.apache.hadoop.hive.ql.session.SessionState - Resetting thread name to  main
```

```
hive> USE banking_data;
2025-02-21T18:39:00,796 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: 10a63bfa-a255-4d01-8f85-991187dace5b
2025-02-21T18:39:00,796 INFO [main] org.apache.hadoop.hive.ql.session.SessionState - Updating thread name to 10a63bfa-a255-4d01-8f85-991187dace5b main
OK
Time taken: 0.094 seconds
2025-02-21T18:39:00,890 INFO [10a63bfa-a255-4d01-8f85-991187dace5b main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: 10a63bfa-a
255-4d01-8f85-991187dace5b
2025-02-21T18:39:00,890 INFO [10a63bfa-a255-4d01-8f85-991187dace5b main] org.apache.hadoop.hive.ql.session.SessionState - Resetting thread name to  main
hive>
```

Q1.2: Define and create a Hive table client_info with appropriate data types for the bank.csv dataset.

```
CREATE TABLE client_info (
    age INT, job STRING, balance FLOAT, ...
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE;
```

```
hive> CREATE TABLE client_info(
    > age INT,
    > job STRING,
    > marital STRING,
    > education STRING,
    > default STRING,
    > balance INT,
    > housing STRING,
    > loan STRING,
    > contact STRING,
    > day INT,
    > month STRING,
    > duration INT,
    > campaign INT,
    > pdays INT,
    > previous INT,
    > poutcome STRING,
    > y STRING
    > )
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY ','
    > STORED AS TEXTFILE;
2025-02-21T18:47:27,940 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: 10a63bfa-a255-4d01-8f85-991187dace5b
2025-02-21T18:47:27,940 INFO [main] org.apache.hadoop.hive.ql.session.SessionState - Updating thread name to 10a63bfa-a255-4d01-8f85-991187dace5b main
OK
Time taken: 2.437 seconds
2025-02-21T18:47:30,393 INFO [10a63bfa-a255-4d01-8f85-991187dace5b main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: 10a63bfa-a
255-4d01-8f85-991187dace5b
2025-02-21T18:47:30,424 INFO [10a63bfa-a255-4d01-8f85-991187dace5b main] org.apache.hadoop.hive.ql.session.SessionState - Resetting thread name to  main
hive>
```

Q1.3: Load the data from the bank.csv file into the client_info table.

```
LOAD DATA INPATH '/banking_data/bank.csv' INTO TABLE client_info;
```

```
hive> LOAD DATA LOCAL INPATH 'C:/Users/MANISH/Desktop/bank.csv' INTO TABLE client_info;
2025-02-22T11:08:16,105 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: fa9f5dee-fdd8-44d7-bc75-b4465eb86d
2025-02-22T11:08:16,105 INFO [main] org.apache.hadoop.hive.ql.session.SessionState - Updating thread name to fa9f5dee-fdd8-44d7-bc75-b4465eb86d38 main
Loading data to table banking_data.client_info
OK
Time taken: 3.469 seconds
2025-02-22T11:08:19,580 INFO [fa9f5dee-fdd8-44d7-bc75-b4465eb86d38 main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log i
9f5dee-fdd8-44d7-bc75-b4465eb86d38
2025-02-22T11:08:19,580 INFO [fa9f5dee-fdd8-44d7-bc75-b4465eb86d38 main] org.apache.hadoop.hive.ql.session.SessionState - Resetting thread name to  main
```

```
hive> SELECT * FROM client_info LIMIT 10;
2025-02-22T11:08:57,899 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: fa9f5dee-fdd8-44d7-bc75-b4465eb86d38
2025-02-22T11:08:57,900 INFO [main] org.apache.hadoop.hive.ql.session.SessionState - Updating thread name to fa9f5dee-fdd8-44d7-bc75-b4465eb86d38 main
2025-02-22T11:09:01,313 INFO [fa9f5dee-fdd8-44d7-bc75-b4465eb86d38 main] org.apache.hadoop.hive.common.FileUtils - Creating directory if it doesn't exist: hdfs:
//localhost:9000/tmp/hive/MANISH/fa9f5dee-fdd8-44d7-bc75-b4465eb86d38/hive_2025-02-22_11-08-57_925_790476432229534492 9-1/-mr-10001/.hive-staging_hive_2025-02-22
_11-08-57_925_7904764322295344929-1
2025-02-22T11:09:01,635 INFO [fa9f5dee-fdd8-44d7-bc75-b4465eb86d38 main] org.apache.hadoop.conf.Configuration.deprecation - mapred.task.is.map is deprecated. In
stead, use mapreduce.task.ismap
OK
NULL    job     marital education       default NULL    housing loan    contact NULL    month   NULL    NULL    NULL    NULL    poutcome        y
30      unemployed      married primary no      1787    no      no      cellular        19      oct     79      1       -1      0       unknown no
33      services        married secondary       no      4789    yes     yes     cellular        11      may     220     1       339     4       failure no
35      management      single  tertiary        no      1350    yes     no      cellular        16      apr     185     1       330     1       failure no
30      management      married tertiary        no      1476    yes     yes     unknown 3       jun     199     4       -1      0       unknown no
59      blue-collar     married secondary       no      0       yes     no      unknown 5       may     226     1       -1      0       unknown no
35      management      single  tertiary        no      747     no      no      cellular        23      feb     141     2       176     3       failure no
36      self-employed   married tertiary        no      307     yes     no      cellular        14      may     341     1       330     2       other   no
39      technician      married secondary       no      147     yes     no      cellular        6       may     151     2       -1      0       unknown no
41      entrepreneur    married tertiary        no      221     yes     no      unknown 14      may     57      2       -1      0       unknown no
Time taken: 3.735 seconds, Fetched: 10 row(s)
2025-02-22T11:09:02,072 INFO [fa9f5dee-fdd8-44d7-bc75-b4465eb86d38 main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: fa
9f5dee-fdd8-44d7-bc75-b4465eb86d38
2025-02-22T11:09:02,072 INFO [fa9f5dee-fdd8-44d7-bc75-b4465eb86d38 main] org.apache.hadoop.hive.ql.session.SessionState - Resetting thread name to  main
```


---

2. Basic Data Exploration:

Q2.1: Write a HiveQL query to count the total number of clients in the dataset.

```
SELECT COUNT(*) AS total_clients FROM client_info;
```

```
hive> select count(*) AS total_clients FROM client_info;
2025-02-22T11:13:11,970 INFO [main] org.apache.hadoop.hive.conf.HiveConf -
2025-02-22T11:13:11,971 INFO [main] org.apache.hadoop.hive.ql.session.Sess
2025-02-22T11:13:12,905 INFO [fa9f5dee-fdd8-44d7-bc75-b4465eb86d38 main] o
ost:9000/tmp/hive/MANISH/fa9f5dee-fdd8-44d7-bc75-b4465eb86d38/hive_2025-02-
762052515892251 47-1
Query ID = MANISH_20250222111311_06e0cea0-5ae0-4deb-b6ab-be8152e4b487
Total jobs = 1
Launching Job 1 out of 1
```

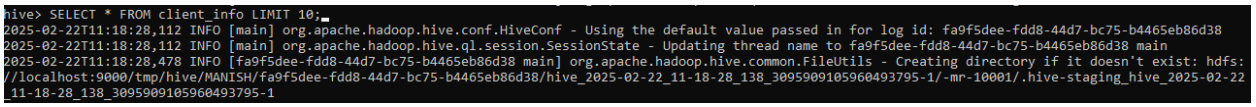
```
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 9.371 sec   HDFS Read: 390574 HDFS Write: 104 SUCCESS
Total MapReduce CPU Time Spent: 9 seconds 371 msec
OK
4522
Time taken: 60.881 seconds, Fetched: 1 row(s)
2025-02-22T11:14:13,019 INFO [fa9f5dee-fdd8-44d7-bc75-b4465eb86d38 main] org.apache.hadoop.hive.conf.HiveCo
dd8-44d7-bc75-b4465eb86d38
2025-02-22T11:14:13,020 INFO [fa9f5dee-fdd8-44d7-bc75-b4465eb86d38 main] org.apache.hadoop.hive.ql.session.
hive>
```

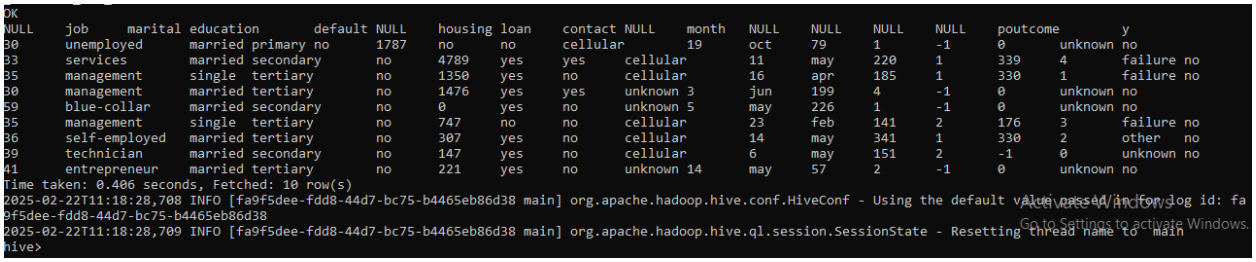
Summary of the Results:

Total Number of Clients: The query returns a single number representing the total number of clients in the dataset. This number gives you a quick overview of the dataset size, indicating how many client records are available for analysis. So, here we can see that the total number of clients is 4522.

Q2.2: Display first 10 rows.

```
SELECT * FROM client_info LIMIT 10;
```





Summary of the Results:

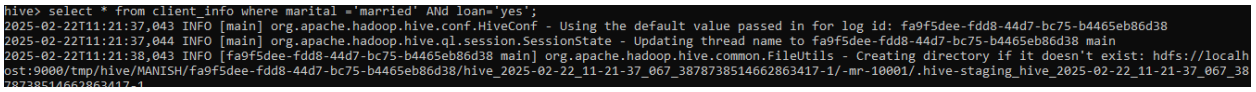
● **First 10 Rows of the Dataset:** Here, we can see that the output displays all columns and their values for the first 10 clients in the client_info table. These rows represent a small sample of the overall dataset, providing a snapshot of the data structure and contents.

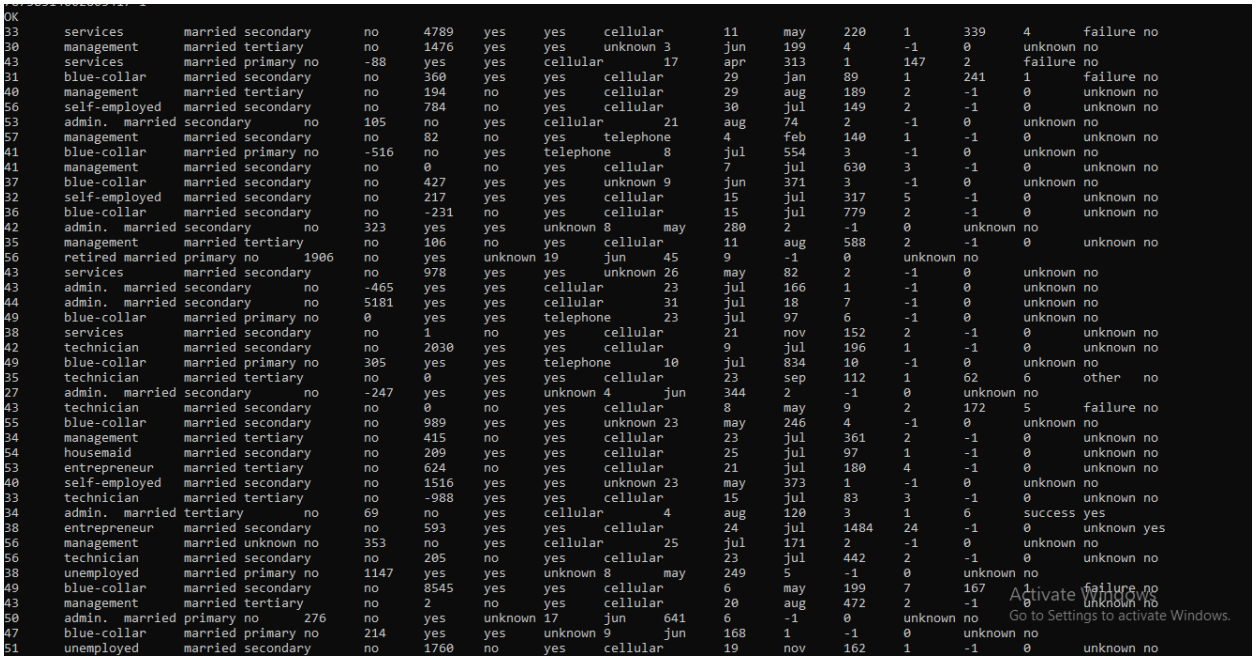
---

3. Data Filtering and Sorting

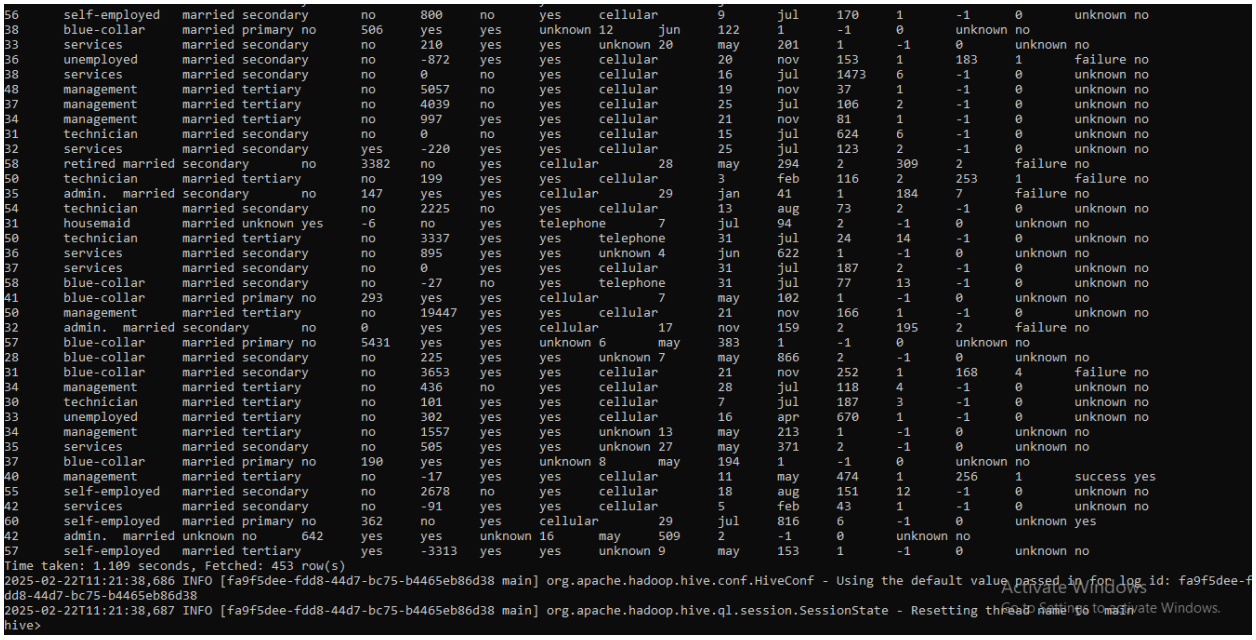
Q3.1: Retrieve all records of clients who are married and have a personal loan.

```
SELECT * FROM client_info WHERE marital = 'married' AND loan = 'yes';
```





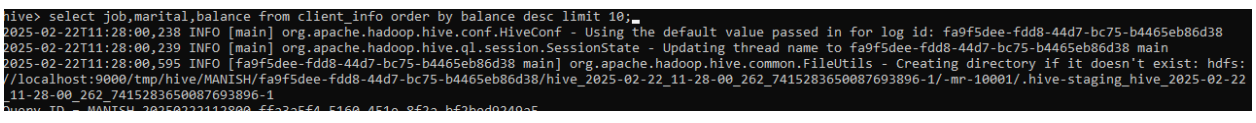
```
OK
33      services        married secondary       no      4789    yes     yes     cellular        11      may     220     1       339     4       failure no
30      management      married tertiary        no      1476    yes     yes     unknown 3       jun     199     4       -1      0       unknown no
43      services        married primary no      -88     yes     yes     cellular        17      apr     313     1       147     2       failure no
31      blue-collar     married secondary       no      360     yes     yes     cellular        29      jan     89      1       241     1       failure no
40      management      married tertiary        no      194     no      yes     cellular        29      aug     189     2       -1      0       unknown no
56      self-employed   married secondary       no      784     no      yes     cellular        30      jul     149     2       -1      0       unknown no
53      admin.  married secondary       no      105     no      yes     cellular        21      aug     74      2       -1      0       unknown no
57      management      married secondary       no      82      no      yes     telephone       4       feb     140     1       -1      0       unknown no
41      blue-collar     married primary no      -516    no      yes     telephone       8       jul     554     3       -1      0       unknown no
41      management      married secondary       no      0       no      yes     cellular        7       jul     630     3       -1      0       unknown no
37      blue-collar     married secondary       no      427     yes     yes     unknown 9       jun     371     3       -1      0       unknown no
32      self-employed   married secondary       no      217     yes     yes     cellular        15      jul     317     5       -1      0       unknown no
36      blue-collar     married secondary       no      -231    no      yes     cellular        15      jul     779     2       -1      0       unknown no
42      admin.  married secondary       no      323     yes     yes     unknown 8       may     280     2       -1      0       unknown no
35      management      married tertiary        no      106     no      yes     cellular        11      aug     588     2       -1      0       unknown no
56      retired married primary no      1906    no      yes     unknown 19      jun     45      9       -1      0       unknown no
43      services        married secondary       no      978     yes     yes     unknown 26      may     82      2       -1      0       unknown no
43      admin.  married secondary       no      -465    yes     yes     cellular        23      jul     166     1       -1      0       unknown no
44      admin.  married secondary       no      5181    yes     yes     cellular        31      jul     18      7       -1      0       unknown no
49      blue-collar     married primary no      0       yes     yes     telephone       23      jul     97      6       -1      0       unknown no
38      services        married secondary       no      1       no      yes     cellular        21      nov     152     2       -1      0       unknown no
42      technician      married secondary       no      2030    yes     yes     cellular        9       jul     196     1       -1      0       unknown no
49      blue-collar     married primary no      305     yes     yes     telephone       10      jul     834     10      -1      0       unknown no
35      technician      married tertiary        no      0       yes     yes     cellular        23      sep     112     1       62      6       other   no
27      admin.  married secondary       no      -247    yes     yes     unknown 4       jun     344     2       -1      0       unknown no
43      technician      married secondary       no      0       no      yes     cellular        8       may     9       2       172     5       failure no
55      blue-collar     married secondary       no      989     yes     yes     unknown 23      may     246     4       -1      0       unknown no
34      management      married tertiary        no      415     no      yes     cellular        23      jul     361     2       -1      0       unknown no
54      housemaid       married secondary       no      209     yes     yes     cellular        25      jul     97      1       -1      0       unknown no
53      entrepreneur    married tertiary        no      624     no      yes     cellular        21      jul     180     4       -1      0       unknown no
40      self-employed   married secondary       no      1516    yes     yes     unknown 23      may     373     1       -1      0       unknown no
33      technician      married tertiary        no      -988    yes     yes     cellular        15      jul     83      3       -1      0       unknown no
34      admin.  married tertiary        no      69      no      yes     cellular        4       aug     120     3       1       6       success yes
38      entrepreneur    married secondary       no      593     yes     yes     cellular        24      jul     1484    24      -1      0       unknown yes
56      management      married unknown no      353     no      yes     cellular        25      jul     171     2       -1      0       unknown no
56      technician      married secondary       no      205     no      yes     cellular        23      jul     442     2       -1      0       unknown no
38      unemployed      married primary no      1147    yes     yes     unknown 8       may     249     5       -1      0       unknown no
49      blue-collar     married secondary       no      8545    yes     yes     cellular        6       may     199     7       167     1       failure no
43      management      married tertiary        no      2       no      yes     cellular        20      aug     472     2       -1      0       unknown no
50      admin.  married primary no      276     no      yes     unknown 17      jun     641     6       -1      0       unknown no
47      blue-collar     married primary no      214     yes     yes     unknown 9       jun     168     1       -1      0       unknown no
51      unemployed      married secondary       no      1760    no      yes     cellular        19      nov     162     1       -1      0       unknown no
```

```
56      self-employed   married secondary       no      800     no      yes     cellular        9       jul     170     1       -1      0       unknown no
38      blue-collar     married primary no      506     yes     yes     unknown 12      jun     122     1       -1      0       unknown no
33      services        married secondary       no      210     yes     yes     unknown 20      may     201     1       -1      0       unknown no
36      unemployed      married secondary       no      -872    yes     yes     cellular        20      nov     153     1       183     1       failure no
38      services        married secondary       no      0       no      yes     cellular        16      jul     1473    6       -1      0       unknown no
48      management      married tertiary        no      5057    no      yes     cellular        19      nov     37      1       -1      0       unknown no
37      management      married tertiary        no      4039    no      yes     cellular        25      jul     106     2       -1      0       unknown no
34      management      married tertiary        no      997     yes     yes     cellular        21      nov     81      1       -1      0       unknown no
31      technician      married secondary       no      0       no      yes     cellular        15      jul     624     6       -1      0       unknown no
32      services        married secondary       yes     -220    yes     yes     cellular        25      jul     123     2       -1      0       unknown no
58      retired married secondary       no      3382    no      yes     cellular        28      may     294     2       309     2       failure no
50      technician      married tertiary        no      199     yes     yes     cellular        3       feb     116     2       253     1       failure no
35      admin.  married secondary       no      147     yes     yes     cellular        29      jan     41      1       184     7       failure no
54      technician      married secondary       no      2225    no      yes     cellular        13      aug     73      2       -1      0       unknown no
31      housemaid       married unknown yes     -6      no      yes     telephone       7       jul     94      2       -1      0       unknown no
50      technician      married tertiary        no      3337    yes     yes     telephone       31      jul     24      14      -1      0       unknown no
36      services        married secondary       no      895     yes     yes     unknown 4       jun     622     1       -1      0       unknown no
37      services        married secondary       no      0       yes     yes     cellular        31      jul     187     2       -1      0       unknown no
58      blue-collar     married secondary       no      -27     no      yes     telephone       31      jul     77      13      -1      0       unknown no
41      blue-collar     married primary no      293     yes     yes     cellular        7       may     102     1       -1      0       unknown no
50      management      married tertiary        no      19447   yes     yes     cellular        21      nov     166     1       -1      0       unknown no
32      admin.  married secondary       no      0       yes     yes     cellular        17      nov     159     2       195     2       failure no
57      blue-collar     married primary no      5431    yes     yes     unknown 6       may     383     1       -1      0       unknown no
28      blue-collar     married secondary       no      225     yes     yes     unknown 7       may     866     2       -1      0       unknown no
31      blue-collar     married secondary       no      3653    yes     yes     cellular        21      nov     252     1       168     4       failure no
34      management      married tertiary        no      436     no      yes     cellular        28      jul     118     4       -1      0       unknown no
30      technician      married tertiary        no      101     yes     yes     cellular        7       jul     187     3       -1      0       unknown no
33      unemployed      married tertiary        no      302     yes     yes     cellular        16      apr     670     1       -1      0       unknown no
34      management      married tertiary        no      1557    yes     yes     unknown 13      may     213     1       -1      0       unknown no
35      services        married secondary       no      505     yes     yes     unknown 27      may     371     2       -1      0       unknown no
37      blue-collar     married primary no      190     yes     yes     unknown 8       may     194     1       -1      0       unknown no
40      management      married tertiary        no      -17     yes     yes     cellular        11      may     474     1       256     1       success yes
55      self-employed   married secondary       no      2678    no      yes     cellular        18      aug     151     12      -1      0       unknown no
42      services        married secondary       no      -91     yes     yes     cellular        5       feb     43      1       -1      0       unknown no
60      self-employed   married primary no      362     no      yes     cellular        29      jul     816     6       -1      0       unknown yes
42      admin.  married unknown no      642     yes     yes     unknown 16      may     509     2       -1      0       unknown no
57      self-employed   married tertiary        yes     -3313   yes     yes     unknown 9       may     153     1       -1      0       unknown no
Time taken: 1.109 seconds, Fetched: 453 row(s)
2025-02-22T11:21:38,686 INFO [fa9f5dee-fdd8-44d7-bc75-b4465eb86d38 main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: fa9f5dee-f
dd8-44d7-bc75-b4465eb86d38
2025-02-22T11:21:38,687 INFO [fa9f5dee-fdd8-44d7-bc75-b4465eb86d38 main] org.apache.hadoop.hive.ql.session.SessionState - Resetting thread name to main
hive>
```


Q3.2: List the top 10 clients with the highest balance, displaying their job, marital status, and balance.

```
SELECT job, marital, balance FROM client_info ORDER BY balance DESC LIMIT 10;
```



```
hive> select job,marital,balance from client_info order by balance desc limit 10;
2025-02-22T11:28:00,238 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: fa9f5dee-fdd8-44d7-bc75-b4465eb86d38
2025-02-22T11:28:00,239 INFO [main] org.apache.hadoop.hive.ql.session.SessionState - Updating thread name to fa9f5dee-fdd8-44d7-bc75-b4465eb86d38 main
2025-02-22T11:28:00,595 INFO [fa9f5dee-fdd8-44d7-bc75-b4465eb86d38 main] org.apache.hadoop.hive.common.FileUtils - Creating directory if it doesn't exist: hdfs:
//localhost:9000/tmp/hive/MANISH/fa9f5dee-fdd8-44d7-bc75-b4465eb86d38/hive_2025-02-22_11-28-00_262_7415283650087693896-1/-mr-10001/.hive-staging_hive_2025-02-22
_11-28-00_262_7415283650087693896-1
Query ID = MANISH_20250222112800_ffa3a5f4-5160-451e-8f3a-bf3bad9349a5
```

```
OK
retired married 71188
entrepreneur    married 42045
technician      single  27733
management      married 27359
technician      married 27069
housemaid       single  26965
retired married 26452
services        married 26394
management      divorced        26306
retired single  25824
Time taken: 47.462 seconds, Fetched: 10 row(s)
2025-02-22T11:28:47,829 INFO [fa9f5dee-fdd8-44d7-bc75-b4465eb86d38
9f5dee-fdd8-44d7-bc75-b4465eb86d38
2025-02-22T11:28:47,830 INFO [fa9f5dee-fdd8-44d7-bc75-b4465eb86d38
hive>
```

---

4. Data Aggregation and Grouping

Q4.1: Average age per job category.

```
SELECT job, AVG(age) AS average_age FROM client_info GROUP BY job;
```

```
hive> select job,AVG(age) AS average_age from client_info group by job;
2025-02-22T11:30:32,696 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: fa9f5dee-fdd8-44d7-bc75-b4465eb86d38
2025-02-22T11:30:32,697 INFO [main] org.apache.hadoop.hive.ql.session.SessionState - Updating thread name to fa9f5dee-fdd8-44d7-bc75-b4465eb86d38 main
2025-02-22T11:30:33,040 INFO [fa9f5dee-fdd8-44d7-bc75-b4465eb86d38 main] org.apache.hadoop.hive.common.FileUtils - Creating directory if it doesn't exist: hdfs:
//localhost:9000/tmp/hive/MANISH/fa9f5dee-fdd8-44d7-bc75-b4465eb86d38/hive_2025-02-22_11-30-32_722_5607926747041170607-1/-mr-10001/.hive-staging_hive_2025-02-22
_11-30-32_722_5607926747041170607-1
```

```
OK
admin.  39.68200836820084
blue-collar     40.15644820295983
entrepreneur    42.01190476190476
housemaid       47.339285714285715
job     NULL
management      40.54076367389061
retired 61.869565217391305
self-employed   41.45355191256831
services        38.57074340527578
student 26.821428571428573
technician      39.470052083333336
unemployed      40.90625
unknown 48.10526315789474
Time taken: 46.533 seconds, Fetched: 13 row(s)
2025-02-22T11:31:19,353 INFO [fa9f5dee-fdd8-44d7-bc75-b4465eb86d38 main] org.
9f5dee-fdd8-44d7-bc75-b4465eb86d38
2025-02-22T11:31:19,354 INFO [fa9f5dee-fdd8-44d7-bc75-b4465eb86d38 main] org.
hive>
```

Q4.2: Find the total number of clients for each education level who have defaulted on credit.

```
SELECT education, COUNT(*) AS total_defaulted_clients FROM client_info WHERE
default = 'yes' GROUP BY education,default;
```

```
hive> select education,default,count(*) AS total_defaulted_client from client_info where default ='yes' group by education,default;
2025-02-22T11:33:59,681 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: fa9f5dee-fdd8-44d7-bc75-b4465eb86d38
2025-02-22T11:33:59,681 INFO [main] org.apache.hadoop.hive.ql.session.SessionState - Updating thread name to fa9f5dee-fdd8-44d7-bc75-b4465eb86d38 main
2025-02-22T11:34:00,010 INFO [fa9f5dee-fdd8-44d7-bc75-b4465eb86d38 main] org.apache.hadoop.hive.common.FileUtils - Creating directory if it doesn't exist: hdfs:
//localhost:9000/tmp/hive/MANISH/fa9f5dee-fdd8-44d7-bc75-b4465eb86d38/hive_2025-02-22_11-33-59_703_582299728363026005O-1/-mr-10001/.hive-staging_hive_2025-02-22
_11-33-59_703_582299728363026005O-1
```

```
OK
primary yes     10
secondary       yes     46
tertiary        yes     17
unknown yes     3
Time taken: 49.307 seconds, Fetched: 4 row(s)
2025-02-22T11:34:49,107 INFO [fa9f5dee-fdd8-44d7-bc75-b4465eb86d38 ma
9f5dee-fdd8-44d7-bc75-b4465eb86d38
2025-02-22T11:34:49,107 INFO [fa9f5dee-fdd8-44d7-bc75-b4465eb86d38 ma
hive>
```

---

5. Complex Queries for Insights

Q5.1: Identify the top 5 job categories with the highest average balance and the percentage of clients in each of these job categories who have subscribed to a term deposit.

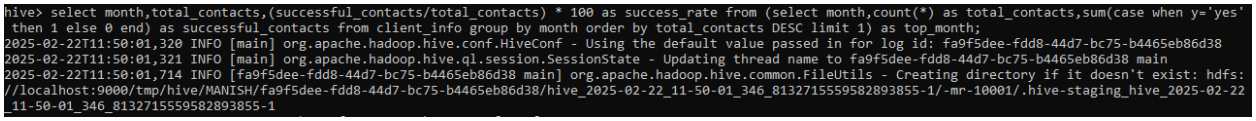
```
Select sc.job,sc.avg_balance,(sc.subscribed_clients / sc.total_clients) * 100
as subscription_percentage from (select job,AVG(balance) as
avg_balance,count(*) as total_clients,sum(case when y='yes' then 1 else 0
end) as subscribed_clients from client_info group by job order by avg_balance
desc limit 5)sc;
```

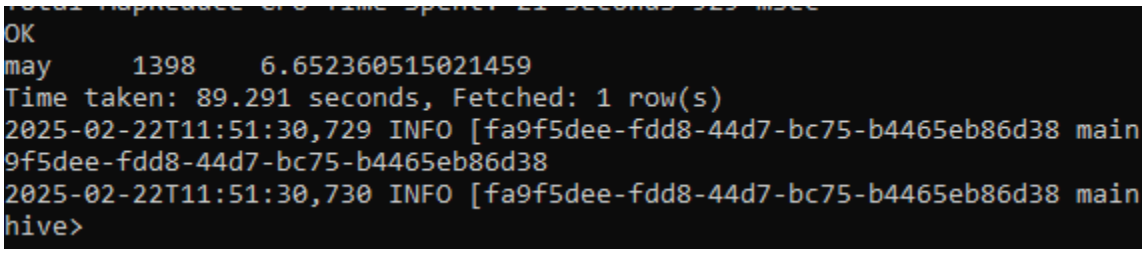
```
hive> select sc.job,sc.avg_balance,(sc.subscribed_clients / sc.total_clients) * 100 as subscription_percentage from (select job,AVG(balance) as avg_balance,coun
t(*) as total_clients,sum(case when y='yes' then 1 else 0 end) as subscribed_clients from client_info group by job order by avg_balance desc limit 5)sc;
2025-02-22T11:41:03,384 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: fa9f5dee-fdd8-44d7-bc75-b4465eb86d38
2025-02-22T11:41:03,384 INFO [main] org.apache.hadoop.hive.ql.session.SessionState - Updating thread name to fa9f5dee-fdd8-44d7-bc75-b4465eb86d38 main
2025-02-22T11:41:03,873 INFO [fa9f5dee-fdd8-44d7-bc75-b4465eb86d38 main] org.apache.hadoop.hive.common.FileUtils - Creating directory if it doesn't exist: hdfs:
//localhost:9000/tmp/hive/MANISH/fa9f5dee-fdd8-44d7-bc75-b4465eb86d38/hive_2025-02-22_11-41-03_415_1402845492661504678-1/-mr-10001/.hive-staging_hive_2025-02-22
_11-41-03_415_1402845492661504678-1
Query ID = MANISH_20250222114103_2c1652cd-a34d-41d8-9d48-c5f509aaf881
```

```
OK
retired 2319.191304347826       23.47826086956522
housemaid       2083.8035714285716      12.5
management      1766.9287925696594      13.519091847265221
entrepreneur    1645.125        8.928571428571429
student 1543.8214285714287      22.61904761904762
Time taken: 90.571 seconds, Fetched: 5 row(s)
2025-02-22T11:42:34,072 INFO [fa9f5dee-fdd8-44d7-bc75-b4465eb86d38 main] org.apac
9f5dee-fdd8-44d7-bc75-b4465eb86d38
2025-02-22T11:42:34,073 INFO [fa9f5dee-fdd8-44d7-bc75-b4465eb86d38 main] org.apac
hive>
```

Q5.2: Determine the month with the highest number of contacts and the success rate of the campaign in that month (percentage of clients who subscribed to a term deposit).

```
Select month,total_contacts,(successful_contacts/total_contacts)*100 as
success_rate from (select month,count(*) as total_contacts,sum(case when
y='yes' then 1 else 0 end)as successful_contacts from client_info group by
month order by total_contacts DESC limit 1)as top_month;
```



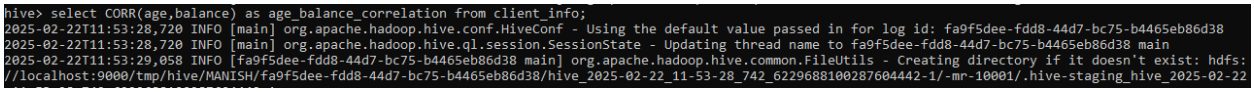


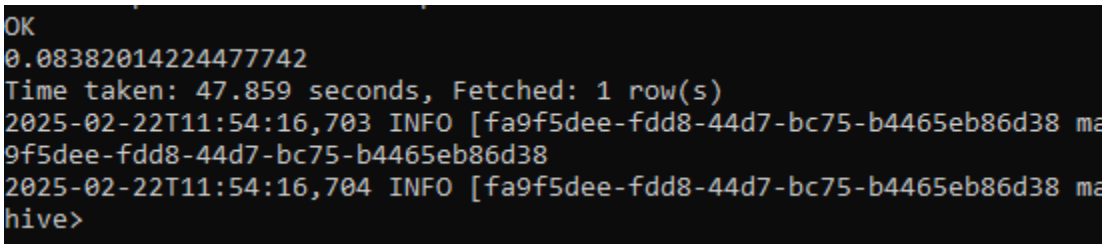
---

6. Correlation Analysis

Q6: Calculate the correlation between age and balance for the clients.

```
SELECT CORR(age, balance) as age_balance_correlation from client_info;
```





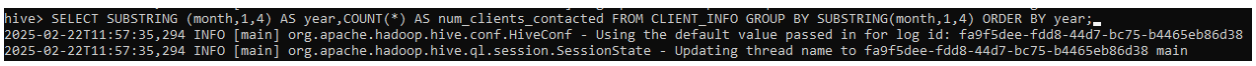
---

7. Trend Analysis

Q7: Analyse the year-over-year trend in the number of clients contacted:

*There is no data in the bank_data.csv dataset which represents the year. But let's say the first four characters from the month column represent the year (e.g., 2023 from 2023-Jan).

```
SELECT SUBSTRING(month, 1, 4) AS year, COUNT(*) AS num_clients_contacted from
client_info group by substring(month,1,4)order by year;
```



```
OK
apr     293
aug     633
dec     20
feb     222
jan     148
jul     706
jun     531
mar     49
may     1398
mont    1
nov     389
oct     80
sep     52
Time taken: 87.188 seconds, Fetched: 13 row(s)
2025-02-22T11:59:02,604 INFO [fa9f5dee-fdd8-44d7-bc
9f5dee-fdd8-44d7-bc75-b4465eb86d38
2025-02-22T11:59:02,604 INFO [fa9f5dee-fdd8-44d7-bc
hive>
```

---

8. Anomaly Detection

Q8: Detect unusual patterns in average yearly balance across education levels.

Identify any unusual patterns in the average yearly balance across different education levels.

```
SELECT year, education, (avg_yearly_balance - overall_avg_balance) /
stddev_balance AS z_score FROM (
    SELECT SUBSTRING(month, 1, 4) AS year, education, AVG(balance) AS
avg_yearly_balance,
           AVG(AVG(balance)) OVER (PARTITION BY year) AS overall_avg_balance,
           STDDEV(AVG(balance)) OVER (PARTITION BY year) AS stddev_balance
    FROM client_info GROUP BY year, education
) AS subquery;
```

```
hive> select year,education,(avg_yearly_balance - overall_avg_balance) / stddev_balance AS z_score
    > FROM (
    > SELECT
    > SUBSTRING(month,1,4) AS year,
    > education,
    > AVG(balance) AS avg_yearly_balance,
    > AVG(AVG(balance)) OVER (PARTITION BY SUBSTRING(month,1,4)) AS overall_avg_balance,
    > STDDEV(AVG(balance)) OVER (PARTITION BY SUBSTRING(month,1,4)) AS stddev_balance
    > FROM client_info
    > GROUP BY SUBSTRING (month,1,4),education
    > ) AS subquery;
2025-02-22T12:15:04,927 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: fa9f5dee-fdd8-44d7-bc75-b4465eb86d3
2025-02-22T12:15:04,927 INFO [main] org.apache.hadoop.hive.ql.session.SessionState - Updating thread name to fa9f5dee-fdd8-44d7-bc75-b4465eb86d38 main
2025-02-22T12:15:05,304 INFO [fa9f5dee-fdd8-44d7-bc75-b4465eb86d38 main] org.apache.hadoop.hive.common.FileUtils - Creating directory if it doesn't exist: h
//localhost:9000/tmp/hive/MANISH/fa9f5dee-fdd8-44d7-bc75-b4465eb86d38/hive_2025-02-22_12-15-04_950_6068760278293976634-1/-mr-10001/.hive-staging_hive_2025-0
```

```
OK
apr     primary 1.267794378462834
apr     secondary       -0.7235770026264813
apr     tertiary        0.6555354947097891
apr     unknown -1.1997528705461413
aug     primary -0.8113812295586758
aug     secondary       -0.5775732890112949
aug     tertiary        -0.31633238372712924
aug     unknown 1.7052869022970998
dec     primary -0.8088635843923103
dec     secondary       -0.397568378390461
dec     tertiary        -0.5058241544774784
dec     unknown 1.7122561172602497
feb     primary -1.0701923004119014
feb     secondary       -0.9216091477916096
feb     tertiary        0.8917649610011248
feb     unknown 1.1000364872023862
jan     primary 1.1404156384061253
jan     secondary       -0.4369029752604172
jan     tertiary        0.7115176049750513
jan     unknown -1.41503026812076
jul     primary 0.9649291030842655
jul     secondary       -0.4258456259680814
jul     tertiary        0.9014112560610348
jul     unknown -1.4404947331772213
jun     primary -0.15682779176634196
jun     secondary       -1.0365919570822748
jun     tertiary        1.6428352994501707
jun     unknown -0.44941555060155386
mar     primary -1.0722876073529977
mar     secondary       -0.2462824810074777
mar     tertiary        1.6391398006892093
mar     unknown -0.3205697123287344
may     primary -1.0048248682462826
may     secondary       -0.9457054758622359
may     tertiary        0.6640708393343474
may     unknown 1.2864595047741691
mont    education       NULL
nov     primary -0.8137802862217836
nov     secondary       0.2770762379753206
nov     tertiary        1.5167439470393023
```

```
nov     tertiary        1.5167439470393023
nov     unknown -0.9800398987928378
oct     primary 1.7097889755337383
oct     secondary       -0.37052787477325194
oct     tertiary        -0.5238214197432853
oct     unknown -0.8154396810172013
sep     primary -1.3086820436917062
sep     secondary       -0.4707154583719365
sep     tertiary        0.3984500713422833
sep     unknown 1.38094743072136
Time taken: 45.104 seconds, Fetched: 49 row(s)
2025-02-22T12:15:50,234 INFO [fa9f5dee-fdd8-44d7-bc75-b4465eb86d38 main] org.
dd8-44d7-bc75-b4465eb86d38
2025-02-22T12:15:50,235 INFO [fa9f5dee-fdd8-44d7-bc75-b4465eb86d38 main] org.
hive>
```

---

9. Advanced Analysis

Q9.1: Analyze the impact of previous campaign outcomes (poutcome) on the current campaign's success. Calculate the subscription rate (to term deposits) for each poutcome category.

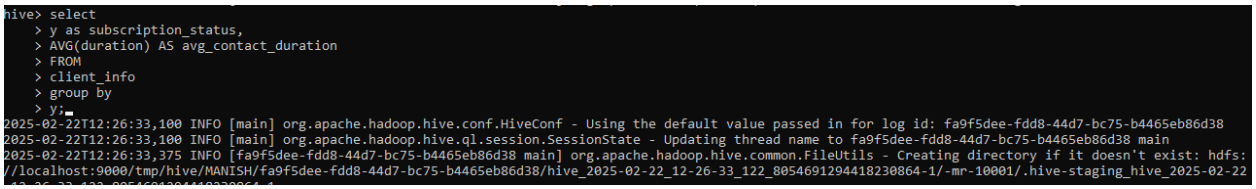
```
SELECT poutcome, COUNT(*) AS total_clients, SUM(CASE WHEN y = 'yes' THEN 1
ELSE 0 END) AS subscribed_clients,
       (SUM(CASE WHEN y = 'yes' THEN 1 ELSE 0 END) / COUNT(*)) * 100 AS
subscription_rate
FROM client_info GROUP BY poutcome ORDER BY subscription_rate DESC;
```

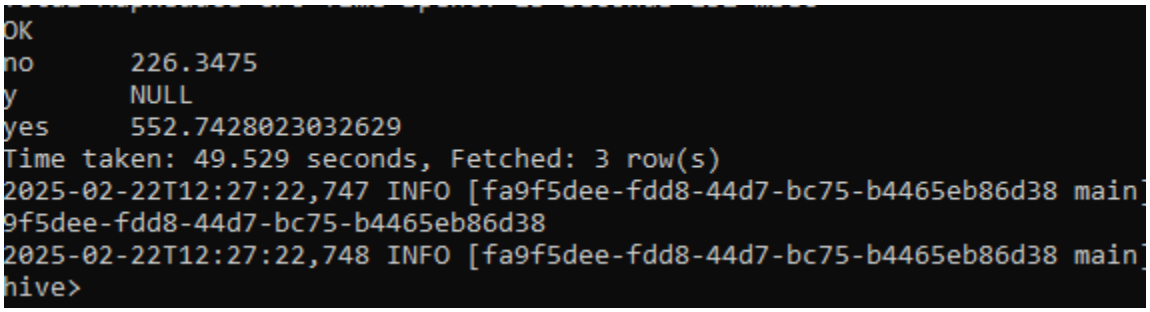
```
hive> select
    > poutcome,
    > count(*) AS total_clients,
    > sum(case when y='yes' then 1 else 0 end)as subscribed_clients,
    > ROUND(SUM(case when y='yes' then 1 else 0 end) * 100.0 / COUNT(*),2) AS subscription_rate
    > FROM
    > client_info
    > GROUP BY
    > poutcome
    > ORDER BY
    > subscription_rate DESC;
2025-02-22T12:22:51,730 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: fa9f5dee-fdd8-44d7-bc75-b4465eb86d38
2025-02-22T12:22:51,730 INFO [main] org.apache.hadoop.hive.ql.session.SessionState - Updating thread name to fa9f5dee-fdd8-44d7-bc75-b4465eb86d38 main
2025-02-22T12:22:52,056 INFO [fa9f5dee-fdd8-44d7-bc75-b4465eb86d38 main] org.apache.hadoop.hive.common.FileUtils - Creating directory if it doesn't exist: hdfs:
//localhost:9000/tmp/hive/MANISH/fa9f5dee-fdd8-44d7-bc75-b4465eb86d38/hive_2025-02-22_12-22-51_752_1777490856644151194-1/-mr-10001/.hive-staging_hive_2025-02-22
_12-22-51_752_1777490856644151194-1
```

```
OK
success 129     83      64.34
other   197     38      19.29
failure 490     63      12.86
unknown 3705    337     9.10
poutcome        1       0       0.00
Time taken: 91.906 seconds, Fetched: 5 row(s)
2025-02-22T12:24:23,731 INFO [fa9f5dee-fdd8-44d7-bc75-b4465eb86
9f5dee-fdd8-44d7-bc75-b4465eb86d38
2025-02-22T12:24:23,732 INFO [fa9f5dee-fdd8-44d7-bc75-b4465eb86
hive>
```

Q9.2: Compare the average contact duration for clients who subscribed and who did not subscribe to a term deposit.

```
SELECT y AS subscription_status, AVG(duration) AS avg_contact_duration FROM
client_info GROUP BY y;
```





The result provides insight into whether there is a difference in the average contact duration between clients who subscribed to the term deposit and those who did not. For instance, here higher average contact duration for the yes group suggests that longer interactions are more effective in convincing clients to subscribe