CHRISTOS PARAPANISIOS

PROJECT – I YELLOW TAXI TRIP ANALYSIS USING HIVE

Note: For the following Hive analysis, I used the CloudxLab platform's web console terminal. The data was downloaded in CSV format from the Kaggle website, titled '2018_Yellow_Taxi_Trip_Data'. The following PDF represents extracts from the Hive environment, along with the commands executed, including my observations and insights.

Step 1: Create a Database named 'Yellowtaxi_2018' then inside create a Table named 'yellow_taxi-trip_data'

```
hive> use Yellowtaxi_2018;

OK

Time taken: 0.278 seconds

hive> show tables;

OK

yellow_taxi_trip_data

Time taken: 0.227 seconds, Fetched: 1 row(s)

hive>
```

Step 2: Display all the column names of the created table

```
hive> DESCRIBE yellow_taxi_trip_data;
OK
vendorid            int
tpep_pickup_datetime    string
tpep_dropoff_datetime   string
passenger_count       int
trip_distance         double
ratecodeid            int
store_and_fwd_flag     string
pulocationid          int
dolocationid          int
payment_type           int
fare_amount           double
extra               double
mta_tax             double
tip_amount            double
tolls_amount          double
improvement_surcharge   double
total_amount          double
Time taken: 0.463 seconds, Fetched: 17 row(s)
hive>
```

Step 3: Display the first 20 rows of the table

```
hive> SELECT * FROM yellow_taxi_trip_data LIMIT 20;
OK
```

2	7/17/2018 10:14	7/17/2018 10:22	2	1	1	N	186	234	1	7	0	0.5	2.34	0	0.3	10.14
2	7/17/2018 10:23	7/17/2018 10:37	2	1.24	1	N	234	186	1	9.5	0	0.5	1.03	0	0.3	11.33
2	7/17/2018 10:39	7/17/2018 11:07	2	5.12	1	N	186	88	1	22	0	0.5	2	0	0.3	24.8
2	7/17/2018 10:12	7/17/2018 10:22	1	0.88	1	N	48	186	2	7.5	0	0.5	0	0	0.3	8.3
2	7/17/2018 10:24	7/17/2018 10:48	1	2.07	1	N	186	163	1	15	0	0.5	0.79	0	0.3	16.59
1	7/17/2018 10:29	7/17/2018 10:41	1	1.2	1	N	50	100	1	8.5	0	0.5	1.85	0	0.3	11.15
1	7/17/2018 10:44	7/17/2018 10:59	1	1.3	1	N	48	186	2	10	0	0.5	0	0	0.3	10.8
2	7/17/2018 10:02	7/17/2018 10:31	1	4.03	1	N	246	209	1	20.5	0	0.5	4.26	0	0.3	25.56
2	7/17/2018 10:23	7/17/2018 10:50	5	2.31	1	N	237	233	1	16.5	0	0.5	2	0	0.3	19.3
2	7/17/2018 10:01	7/17/2018 10:06	1	0.63	1	N	230	163	1	5	0	0.5	1.16	0	0.3	6.96
2	7/17/2018 10:12	7/17/2018 10:24	1	1.16	1	N	161	163	1	9	0	0.5	1.96	0	0.3	11.76
2	7/17/2018 10:28	7/17/2018 10:40	1	1.02	1	N	163	162	1	8	0	0.5	1	0	0.3	9.8
2	7/17/2018 10:55	7/17/2018 11:10	1	4.57	1	N	140	79	1	15.5	0	0.5	1	0	0.3	17.3
1	7/17/2018 10:26	7/17/2018 11:24	1	10.5	1	N	138	163	1	42.5	0	0.5	9.8	5.76	0.3	58.86
1	7/17/2018 10:22	7/17/2018 10:35	1	1.1	1	N	186	234	1	9	0	0.5	2	0	0.3	11.8
1	7/17/2018 10:38	7/17/2018 11:00	2	2.2	1	N	234	161	2	15	0	0.5	0	0	0.3	15.8
2	7/17/2018 10:17	7/17/2018 10:46	1	2.22	1	N	162	140	1	17.5	0	0.5	3.66	0	0.3	21.96
2	7/17/2018 10:55	7/17/2018 11:04	1	2.23	1	N	262	233	1	9.5	0	0.5	2.06	0	0.3	12.36
1	7/17/2018 10:25	7/17/2018 11:02	1	4.7	1	N	186	88	1	25	0	0.5	7.7	0	0.3	33.5
2	7/17/2018 9:58	7/17/2018 10:13	1	3.33	1	N	151	50	1	14.5	0	0.5	0	0	0.3	15.3

```
Time taken: 0.501 seconds, Fetched: 20 row(s)
hive>
```

Observations: Now that the data is ready and well set inside Hive let's start by answering the following questions.

Question 1: What is the total Number of trips (equal to the number of rows)?

```
hive> SELECT COUNT(*) AS total_trips FROM yellow_taxi_trip_data;
Query ID = christosparapanisios6631_20240925061323_745edcea-6c9c-451c-a18c-42de77f541de
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_1712746078021_1746, Tracking URL = http://cxln2.c.thelab-240901.internal:8088/proxy/application_1712746078021_1746/
Kill Command = /usr/hdp/2.6.2.0-205/hadoop/bin/hadoop job  -kill job_1712746078021_1746
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2024-09-25 06:13:33,883 Stage-1 map = 0%,  reduce = 0%
2024-09-25 06:13:44,292 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 5.08 sec
2024-09-25 06:13:50,545 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 7.97 sec
MapReduce Total cumulative CPU time: 7 seconds 970 msec
Ended Job = job_1712746078021_1746
```

MapReduce Jobs Launched:

Stage-Stage-1: Map: 1 Reduce: 1   Cumulative CPU: 7.97 sec   HDFS Read: 16262264 HDFS Write: 7 SUCCESS

Total MapReduce CPU Time Spent: 7 seconds 970 msec

OK

total_trips

203111

Time taken: 27.77 seconds, Fetched: 1 row(s)

hive>

Responses and Observations:

The query executed successfully and returned the total number of trips (rows) in the yellow_taxi_trip_data table,

which is 203,111. The query took 27.77 seconds to complete using Hadoop's MapReduce framework.

Question 2: What is the total revenue generated by all the trips? The fare is stored in the column total_amount.

```
hive> SELECT SUM(total_amount) AS total_revenue FROM yellow_taxi_trip_data;
Query ID = christosparapanisios6631_20240925062549_69f4caec-1409-4695-b975-6c2722ed87d5
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_1712746078021_1747, Tracking URL = http://cxln2.c.thelab-240901.internal:8088/proxy/application_1712746078021_1747/
Kill Command = /usr/hdp/2.6.2.0-205/hadoop/bin/hadoop job  -kill job_1712746078021_1747
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2024-09-25 06:25:56,538 Stage-1 map = 0%,  reduce = 0%
2024-09-25 06:26:03,892 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 5.4 sec
2024-09-25 06:26:11,171 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 8.41 sec
MapReduce Total cumulative CPU time: 8 seconds 410 msec
```

Ended Job = job_1712746078021_1747

MapReduce Jobs Launched:

Stage-Stage-1: Map: 1 Reduce: 1   Cumulative CPU: 8.41 sec   HDFS Read: 16262449 HDFS Write: 18 SUCCESS

Total MapReduce CPU Time Spent: 8 seconds 410 msec

OK

total_revenue

3403081.249995992

Time taken: 22.917 seconds, Fetched: 1 row(s)

hive>

Responses and Observations:

The output process involved Hive launching a MapReduce job to calculate the total revenue by summing the values in the total_amount column.

Map Phase: Read from HDFS (around 16 MB) of data.

Reduce Phase: The reducer collects all the total_amount values from the mappers and adds them together to calculate the total revenue,

but the final result is a single sum of all the taxi trips revenue.

Result: The total revenue was 3 403 081,25$ dollars.

Question 3: What fraction of the total is paid for tolls? The toll is stored in tolls_amount.

```
hive> SELECT SUM(tolls_amount) / SUM(total_amount) AS tolls_fraction FROM yellow_taxi_trip_data;
Query ID = christosparapanisios6631_20240925064632_19cfb57b-c5fa-4ba7-8203-bb9f41e034e9
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_1712746078021_1748, Tracking URL = http://cxln2.c.thelab-240901.internal:8088/proxy/application_1712746078021_1748/
Kill Command = /usr/hdp/2.6.2.0-205/hadoop/bin/hadoop job  -kill job_1712746078021_1748
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2024-09-25 06:46:39,969 Stage-1 map = 0%,  reduce = 0%
2024-09-25 06:46:47,294 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 4.69 sec
2024-09-25 06:46:53,555 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 9.08 sec
MapReduce Total cumulative CPU time: 9 seconds 80 msec
```

Ended Job = job_1712746078021_1748

MapReduce Jobs Launched:

Stage-Stage-1: Map: 1 Reduce: 1   Cumulative CPU: 9.08 sec   HDFS Read: 16263738 HDFS Write: 21 SUCCESS

Total MapReduce CPU Time Spent: 9 seconds 80 msec

OK

tolls_fraction

0.018015247211649047

Time taken: 22.719 seconds, Fetched: 1 row(s)

hive>

Responses and Observations:

The query executed successfully and returned the fraction of the total revenue paid for tolls as 0.018 or 1.8% of the total revenue.

The process involved Hive running a MapReduce job to calculate the sums of tolls_amount and total_amount and then computing the fraction.

Question 4: What fraction of it is driver tips? The tip is stored in tip_amount.

```
hive> SELECT SUM(tip_amount) / SUM(total_amount) AS tips_fraction FROM yellow_taxi_trip_data;
Query ID = christosparapanisios6631_20240925065304_a2e9fee7-d199-4cfe-a7df-2a86dad36bea
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_1712746078021_1749, Tracking URL = http://cxln2.c.thelab-240901.internal:8088/proxy/application_1712746078021_1749/
Kill Command = /usr/hdp/2.6.2.0-205/hadoop/bin/hadoop job  -kill job_1712746078021_1749
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2024-09-25 06:53:11,969 Stage-1 map = 0%,  reduce = 0%
2024-09-25 06:53:19,240 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 5.19 sec
2024-09-25 06:53:25,478 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 8.96 sec
MapReduce Total cumulative CPU time: 8 seconds 960 msec
```

Ended Job = job_1712746078021_1749

MapReduce Jobs Launched:

Stage-Stage-1: Map: 1 Reduce: 1   Cumulative CPU: 8.96 sec   HDFS Read: 16263727 HDFS Write: 20 SUCCESS

Total MapReduce CPU Time Spent: 8 seconds 960 msec

OK

tips_fraction

0.11492920423233527

Time taken: 21.636 seconds, Fetched: 1 row(s)

hive>

Responses and Observations:

The query executed successfully and returned the fraction of the total revenue coming from driver tips as 0.115 or 11.5% of the total revenue.

The process involved Hive running a MapReduce job to calculate the sums of tip_amount and total_amount, and then computing the fraction.

Question 5: What is the average trip amount?

```
hive> SELECT AVG(total_amount) AS average_trip_amount FROM yellow_taxi_trip_data;
Query ID = christosparapanisios6631_20240925065631_e90ed945-a1a0-4f53-b5bb-49309c945579
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_1712746078021_1750, Tracking URL = http://cxln2.c.thelab-240901.internal:8088/proxy/application_1712746078021_1750/
Kill Command = /usr/hdp/2.6.2.0-205/hadoop/bin/hadoop job  -kill job_1712746078021_1750
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2024-09-25 06:56:38,249 Stage-1 map = 0%,  reduce = 0%
2024-09-25 06:56:45,534 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 4.03 sec
2024-09-25 06:56:51,868 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 6.89 sec
```

MapReduce Total cumulative CPU time: 6 seconds 890 msec

Ended Job = job_1712746078021_1750

MapReduce Jobs Launched:

Stage-Stage-1: Map: 1 Reduce: 1   Cumulative CPU: 6.89 sec   HDFS Read: 16262883 HDFS Write: 18 SUCCESS

Total MapReduce CPU Time Spent: 6 seconds 890 msec

OK

average_trip_amount

16.75478556058506

Time taken: 21.48 seconds, Fetched: 1 row(s)

hive>

Responses and Observations:

The query executed successfully and returned the average trip amount as $16.75.

Using the command AVG involved Hive running a MapReduce job to calculate the average of the total_amount column.

Question 6: What is the average distance of the trips? Distance is stored in the column trip_distance.

```
hive> SELECT AVG(trip_distance) AS average_trip_distance FROM yellow_taxi_trip_data;
Query ID = christosparapanisios6631_20240925070041_5335f419-e12e-4278-ad27-3456388287de
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_1712746078021_1751, Tracking URL = http://cxln2.c.thelab-240901.internal:8088/proxy/application_1712746078021_1751/
Kill Command = /usr/hdp/2.6.2.0-205/hadoop/bin/hadoop job  -kill job_1712746078021_1751
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2024-09-25 07:00:48,947 Stage-1 map = 0%,  reduce = 0%
2024-09-25 07:00:56,302 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 4.64 sec
2024-09-25 07:01:02,601 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 7.97 sec
MapReduce Total cumulative CPU time: 7 seconds 970 msec
```

Ended Job = job_1712746078021_1751

MapReduce Jobs Launched:

Stage-Stage-1: Map: 1 Reduce: 1   Cumulative CPU: 7.97 sec   HDFS Read: 16262888 HDFS Write: 19 SUCCESS

Total MapReduce CPU Time Spent: 7 seconds 970 msec

OK

average_trip_distance

2.7259977844517747

Time taken: 22.819 seconds, Fetched: 1 row(s)

hive>

Responses and Observations:

The folowing command calculated the average of all the values in the trip_distance column

(it summed up all the trip distances from the table and then divides the total by the number of trips)

giving us the average distance travelled per trip and returned the average trip distance as 2.73 miles.

Question 7: How many different payment types are used?

```
hive> SELECT COUNT(DISTINCT payment_type) AS different_payment_types FROM yellow_taxi_trip_data;
Query ID = christosparapanisios6631_20240925071820_2197c13e-26bc-404c-ad70-9cec496ebc18
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_1712746078021_1752, Tracking URL = http://cxln2.c.thelab-240901.internal:8088/proxy/application_1712746078021_1752/
Kill Command = /usr/hdp/2.6.2.0-205/hadoop/bin/hadoop job  -kill job_1712746078021_1752
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2024-09-25 07:18:27,602 Stage-1 map = 0%,  reduce = 0%
2024-09-25 07:18:34,842 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 4.17 sec
2024-09-25 07:18:42,076 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 7.13 sec
MapReduce Total cumulative CPU time: 7 seconds 130 msec
```

Ended Job = job_1712746078021_1752

MapReduce Jobs Launched:

Stage-Stage-1: Map: 1 Reduce: 1   Cumulative CPU: 7.13 sec   HDFS Read: 16262170 HDFS Write: 2 SUCCESS

Total MapReduce CPU Time Spent: 7 seconds 130 msec

OK

different_payment_types

5

Time taken: 23.123 seconds, Fetched: 1 row(s)

hive>

Responses and Observations:

The query counts the number of unique payment types from the payment_type column by scanning all the payment types in the table and identifies how many distinct methods of payment are recorded and finally returned the number of different payment types as 5.

Question 8: For each payment type, display the following details:

- Average fare generated
- Average tip
- Average tax – tax is stored in column mta_tax

```
hive> SELECT
    >     payment_type,
    >     AVG(fare_amount) AS average_fare,
    >     AVG(tip_amount) AS average_tip,
    >     AVG(mta_tax) AS average_tax
    > FROM yellow_taxi_trip_data
    > WHERE tip_amount >= 0
    > GROUP BY payment_type;
Query ID = christosparapanisios6631_20240925084629_ba8368f9-8d74-4418-bae2-73f03e3dca52
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_1712746078021_1767, Tracking URL = http://cxln2.c.thelab-240901.internal:8088/proxy/application_1712746078021_1767/
Kill Command = /usr/hdp/2.6.2.0-205/hadoop/bin/hadoop job  -kill job_1712746078021_1767
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2024-09-25 08:46:37,077 Stage-1 map = 0%,  reduce = 0%
2024-09-25 08:46:44,428 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 6.28 sec
2024-09-25 08:46:50,720 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 10.06 sec
MapReduce Total cumulative CPU time: 10 seconds 60 msec
Ended Job = job_1712746078021_1767
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 10.06 sec   HDFS Read: 16265850 HDFS Write: 242 SUCCESS
Total MapReduce CPU Time Spent: 10 seconds 60 msec
OK
1      13.630492777235489      2.7466350414687124      0.49800206465023844
2      12.483717187683926      0.0     0.49844451544554125
3      10.480680203045688      0.0017730496453900709   0.5117527862208714
4      10.852217898832684      0.006614785992217899    0.311284046692607
264    2.0     0.0     NULL
Time taken: 22.895 seconds, Fetched: 5 row(s)
hive>
```

Responses and Observations:

Below the calculated average fare tip | tax for each payment type:

- Payment Type 1: Average fare is $13.63 tip is $2.75 | tax is $0.50
- Payment Type 2: Average fare is $12.48 no tip | tax is $0.50.
- Payment Type 3: Average fare is $10.48 a very small tip 0.0018 | tax is $0.51
- Payment Type 4: Average fare is $10.85 a very small tip | tax is $0.31
- Payment Type 5 (Shown as Type 264): Average fare is $2.00 with no tip | no tax. (an anomaly is detected as payment_type 264 is not a standard value)

To investigate further we can check how many records fall under Payment Type 264 (Type 5) and inspect their details:

```
hive> SELECT * FROM yellow_taxi_trip_data WHERE payment_type = 264;
OK
2	7/17/2018 15:19	7/18/2018 13:41	1	NULL	NULL	1	NULL	264	264	2.0	NULL	NULL	0.0	0.5	0.0	0.0
Time taken: 0.173 seconds, Fetched: 1 row(s)
hive>
```

Observation:
From the result of the query, it appears that there is only one row with payment_type = 264 it seems to be a data anomaly or an incomplete entry. The value 264 for payment_type does not correspond to a standard payment method and the NULL values indicate that the data might not be valid or useful. Since it doesn't represent a valid or recognized payment type and contains many NULL values, it is not useful so we can disregard it.

Also, from the results of question 7 and now 8 we can conclude that payment Type 5 exists as a possible category in the dataset but has no actual records associated with it.

Question 9: On an average which hour of the day generates the highest revenue?

---

hive> SELECT tpep_pickup_datetime FROM yellow_taxi_trip_data LIMIT 10;

OK

tpep_pickup_datetime

7/17/2018 10:14

7/17/2018 10:23

7/17/2018 10:39

7/17/2018 10:12

7/17/2018 10:24

7/17/2018 10:29

7/17/2018 10:44

7/17/2018 10:02

7/17/2018 10:23

7/17/2018 10:01

Time taken: 0.044 seconds, Fetched: 10 row(s)

Observations:

First we checked for 10 rows the tpep_pickup_datetime column with (SELECT tpep_pickup_datetime FROM yellow_taxi_trip_data LIMIT 10;)

to see how the dates were formatted.

We found that the format was MM/DD/YYYY HH:mm (for example, 7/17/2018 10:14).

Since this format is not what Hive expects

we will use Hive functions to convert the datetime values into a format that could be processed.

```
hive> SELECT
    >     HOUR(FROM_UNIXTIME(UNIX_TIMESTAMP(tpep_pickup_datetime, 'MM/dd/yyyy HH:mm'))) AS hour_of_day,
    >     AVG(total_amount) AS average_revenue
    > FROM yellow_taxi_trip_data
    > WHERE tpep_pickup_datetime IS NOT NULL
    >   AND tpep_pickup_datetime != ''
    > GROUP BY HOUR(FROM_UNIXTIME(UNIX_TIMESTAMP(tpep_pickup_datetime, 'MM/dd/yyyy HH:mm')))
    > ORDER BY average_revenue DESC
    > LIMIT 1;
Query ID = christosparapanisios6631_20240925075017_1c30e2e1-77e9-43d1-9fd2-3b6d384c631c
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_1712746078021_1762, Tracking URL = http://cxln2.c.thelab-240901.internal:8088/proxy/application_1712746078021_1762/

Kill Command = /usr/hdp/2.6.2.0-205/hadoop/bin/hadoop job -kill job_1712746078021_1762

Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1

2024-09-25 07:50:24,465 Stage-1 map = 0%, reduce = 0%

2024-09-25 07:50:33,751 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 7.57 sec

2024-09-25 07:50:41,181 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 11.84 sec

MapReduce Total cumulative CPU time: 11 seconds 840 msec

Ended Job = job_1712746078021_1762

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_1712746078021_1763, Tracking URL = http://cxln2.c.thelab-240901.internal:8088/proxy/application_1712746078021_1763/

Kill Command = /usr/hdp/2.6.2.0-205/hadoop/bin/hadoop job -kill job_1712746078021_1763

Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1

2024-09-25 07:50:49,129 Stage-2 map = 0%, reduce = 0%

2024-09-25 07:50:55,376 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 2.55 sec

2024-09-25 07:51:01,571 Stage-2 map = 100%, reduce = 100%, Cumulative CPU 5.4 sec

MapReduce Total cumulative CPU time: 5 seconds 400 msec

Ended Job = job_1712746078021_1763

MapReduce Jobs Launched:

Stage-Stage-1: Map: 1 Reduce: 1   Cumulative CPU: 11.84 sec   HDFS Read: 16263346 HDFS Write: 694 SUCCESS

Stage-Stage-2: Map: 1 Reduce: 1   Cumulative CPU: 5.4 sec   HDFS Read: 6233 HDFS Write: 8 SUCCESS

Total MapReduce CPU Time Spent: 17 seconds 240 msec

OK

hour_of_day   average_revenue

1   38.58

Time taken: 45.112 seconds, Fetched: 1 row(s)

hive>

Observations:

HOUR(FROM_UNIXTIME(UNIX_TIMESTAMP(tpep_pickup_datetime, 'MM/dd/yyyy HH:mm'))) AS hour_of_day

The above comand Hive reads the tpep_pickup_datetime column, which contains dates like 7/17/2018 10:14.

It uses UNIX_TIMESTAMP() to convert the date to a standard format and then extracts the hour using HOUR().

For example, from 7/17/2018 10:14, Hive extracts the hour 10. It does this for all rows,

grouping the data by hour to prepare for calculating the average revenue in the next step.

```
 AVG(total_amount) AS average_revenue
FROM yellow_taxi_trip_data
WHERE tpep_pickup_datetime IS NOT NULL
 AND tpep_pickup_datetime != ''
GROUP BY HOUR(FROM_UNIXTIME(UNIX_TIMESTAMP(tpep_pickup_datetime, 'MM/dd/yyyy HH:mm')))
ORDER BY average_revenue DESC
LIMIT 1;
```

After the data grouped and prepared we use the above commands for Hive to take those data's grouped by hour calculating the average revenue for each hour using AVG(total_amount).Going through each hour (like 10 AM, 11 AM, etc.) and computes the average of all trip revenues during that hour. Once those averages done Hive sorts the results by the highest average revenue through the command 'ORDER BY' average_revenue 'DESC' and finally the command 'LIMIT 1' is applied to return just the hour with the highest average revenue.

Result:

From the output we can see that 1 AM is the hour that generates the highest average revenue with an average of 38.58$ dollars.
