Logic Final Submission

Explanation of the queries for the given tasks.

- **Calculate the total number of different drivers for each customer.**

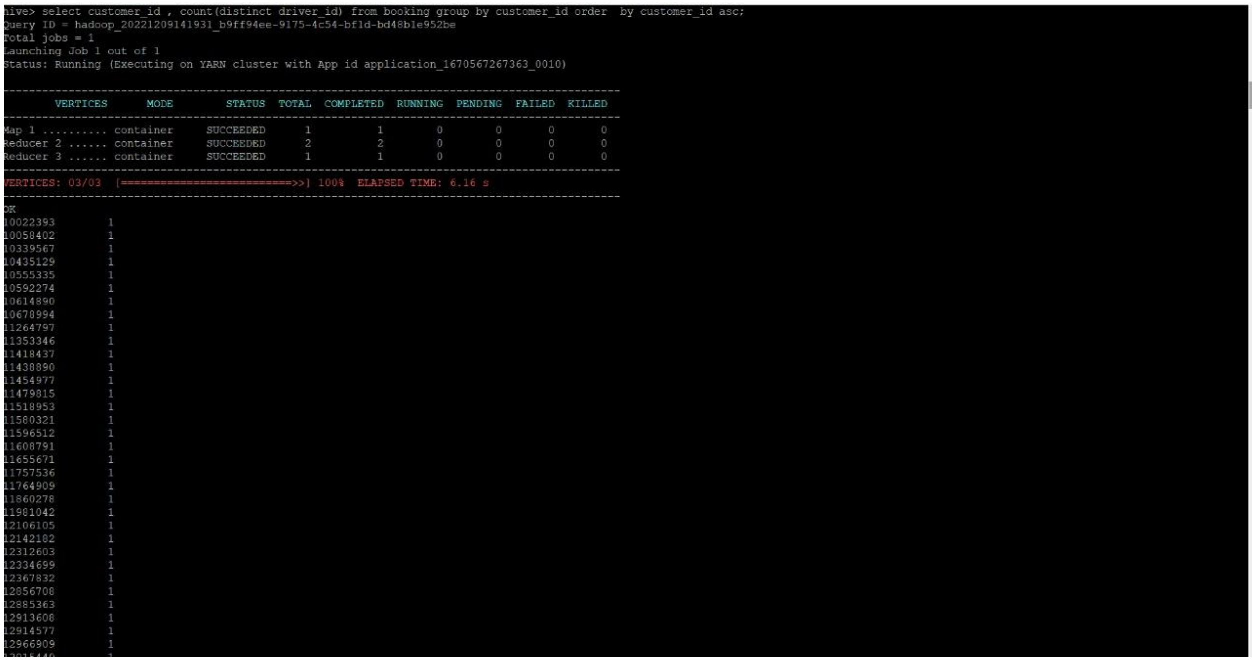
  # Query

```
SELECT customer_id,
       Count(DISTINCT driver_id)
FROM   booking
GROUP  BY customer_id
ORDER  BY customer_id ASC;
```

  # Explanation

  In this query we selected 2 columns, customer_id and driver_id from the booking table and grouped this by customer_id. We used the COUNT() function for the driver_id column to count the number of rows returned with DISTINCT to select unique rows from the table booking_data. In this way we can get the details required in Task 5.

Screenshot showing query run



```
hive> select customer_id , count(distinct driver_id) from booking group by customer_id order  by customer_id asc;
Query ID = hadoop_20221209141931_b9ff94ee-9175-4c54-bf1d-bd48b1e952be
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1670567267363_0010)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      2          2        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 6.16 s
----------------------------------------------------------------------------------------------
OK
10022393        1
10058402        1
10339567        1
10435129        1
10555335        1
10592274        1
10614890        1
10678994        1
11264797        1
11353346        1
11418437        1
11438890        1
11454977        1
11479815        1
11518953        1
11580321        1
11596512        1
11608791        1
11655671        1
11757536        1
11764909        1
11860278        1
11981042        1
12106105        1
12142182        1
12312603        1
12334699        1
12367832        1
12856708        1
12885363        1
12913608        1
12914577        1
12966909        1
```

- **Calculate the total rides taken by each customer**

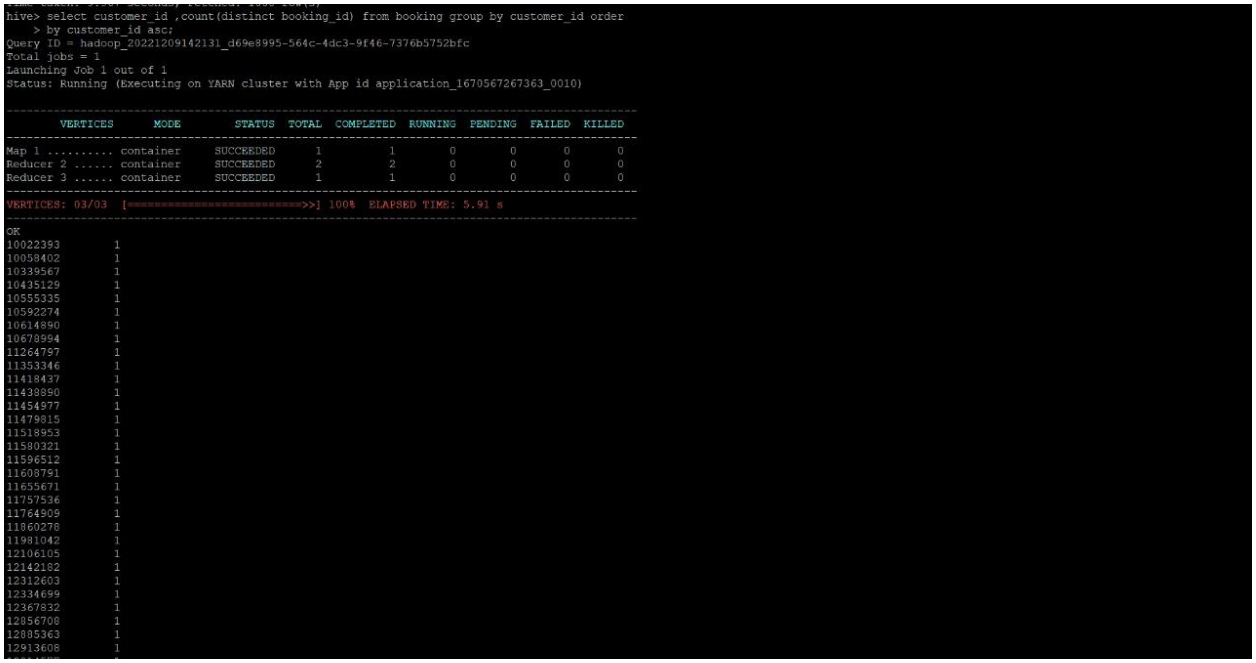
  # Query

```
SELECT customer_id,
       Count(DISTINCT booking_id)
FROM   booking
GROUP  BY customer_id
ORDER  BY customer_id ASC;
```

  # Explanation

  In this task, we selected 2 columns, customer_id and booking_id from the booking table and grouped this by customer_id and ordered by customer_id in ascending order. We used the COUNT() function for the booking_id column to count the number of rows returned with DISTINCT to select unique rows from the table booking_data.
  In this way we can get the details required in task 6

Screenshot showing query run



```
hive> select customer_id ,count(distinct booking_id) from booking group by customer_id order
    > by customer_id asc;
Query ID = hadoop_20221209142131_d69e8995-564c-4dc3-9f46-7376b5752bfc
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1670567267363_0010)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      2          2        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 5.91 s
----------------------------------------------------------------------------------------------
OK
10022393        1
10058402        1
10339567        1
10435129        1
10555335        1
10592274        1
10614890        1
10678994        1
11264797        1
11353346        1
11418437        1
11438890        1
11454977        1
11479815        1
11518953        1
11580321        1
11596512        1
11608791        1
11655671        1
11757536        1
11764909        1
11860278        1
11981042        1
12106105        1
12142182        1
12312603        1
12334699        1
12367832        1
12856708        1
12885363        1
12913608        1
```

- **Find the total visits made by each customer on the booking page and the total ‘Book Now’ button presses. This can show the conversion ratio.
  The booking page id is ‘e7bc5fb2-1231-11eb-adc1-0242ac120002’.
  The Book Now button id is ‘fcba68aa-1231-11eb-adc1-0242ac120002’. You also need to calculate the conversion ratio as part of this task. Conversion ratio can be**

calculated as Total 'Book Now' Button Press/Total Visits made by customer on the booking page.

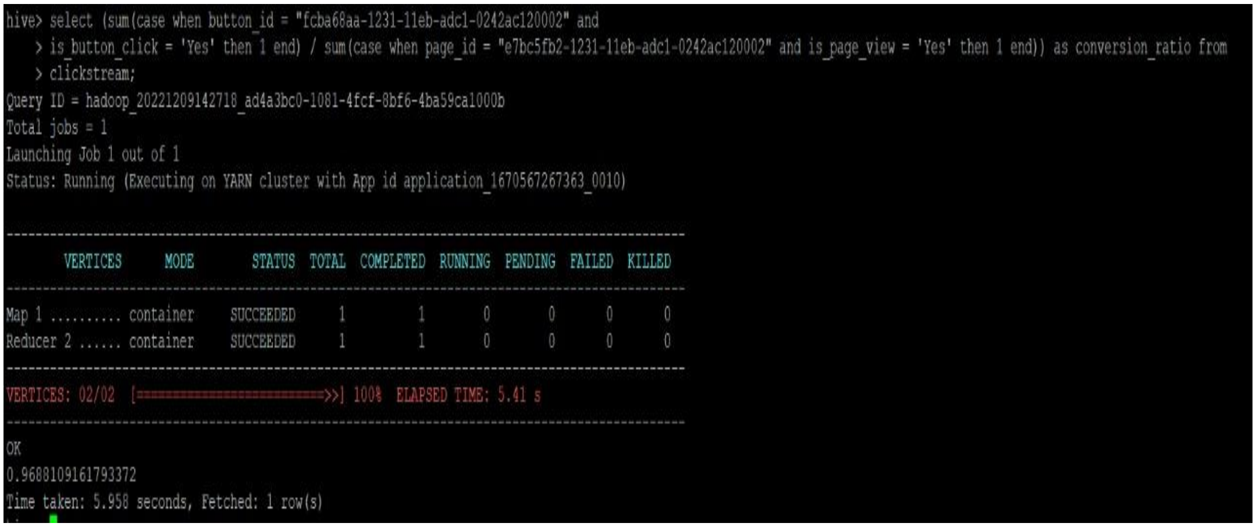
# Query

```
SELECT ( Sum(CASE
               WHEN button_id =
"fcba68aa-1231-11eb-adc1-0242ac120002"
                    AND is_button_click = 'Yes' THEN 1
             END) / Sum(CASE
                          WHEN page_id = "e7bc5fb2-1231-11eb-
adc1-0242ac120002"
                               AND is_page_view = 'Yes' THEN 1
                        END) ) AS conversion_ratio
FROM   clickstream;
```

# Explanation

In this task, first we use clickstream table, and we used SUM() function to sum up values where button_id = "fcba68aa-1231-11eb-adc1-0242ac120002" and is_button_click = 'Yes'. Once again, we used SUM() function to sum up values where page_id = "e7bc5fb2-1231-11eb- adc1-0242ac120002" and is_page_view = 'Yes' and Finally, we divide the first sum by the second sum to get the conversion ratio. In this way we can get the details required in task 7.

Screenshot showing query run



```
hive> select (sum(case when button_id = "fcba68aa-1231-11eb-adc1-0242ac120002" and
    > is_button_click = 'Yes' then 1 end) / sum(case when page_id = "e7bc5fb2-1231-11eb-adc1-0242ac120002" and is_page_view = 'Yes' then 1 end)) as conversion_ratio from
    > clickstream;
Query ID = hadoop_20221209142718_ad4a3bc0-1081-4fcf-8bf6-4ba59ca1000b
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1670567267363_0010)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 5.41 s
----------------------------------------------------------------------------------------------
OK
0.9688109161793372
Time taken: 5.958 seconds, Fetched: 1 row(s)
```

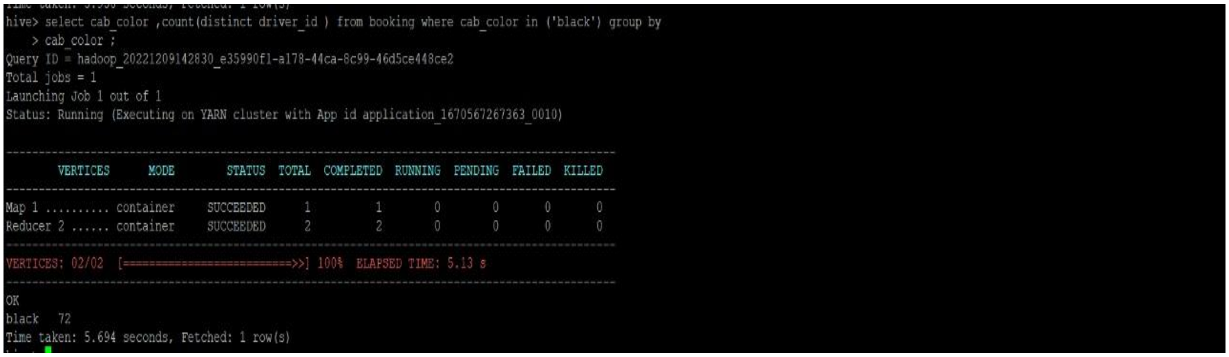
- **Calculate the count of all trips done on black cabs**

# Query

```
SELECT cab_color,
       Count(DISTINCT driver_id)
FROM   booking
WHERE  cab_color IN ( 'black' )
GROUP  BY cab_color;
```

# Explanation
In this task, we selected 2 columns, cab_color and driver_id from the booking table and we used the WHERE clause to filter the data where cab_color is ‘black’ and then we grouped this by cab_color. We used the COUNT() function for the driver_id column to count the number of rows returned with DISTINCT to select unique rows from the table booking.In this way we can get the details required in task 8.

Screenshot showing query run



```
hive> select cab_color ,count(distinct driver_id ) from booking where cab_color in ('black') group by
    > cab_color ;
Query ID = hadoop_20221209142830_e35990f1-a178-44ca-8c99-46d5ce448ce2
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1670567267363_0010)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      2          2        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 5.13 s
----------------------------------------------------------------------------------------------
OK
black   72
Time taken: 5.694 seconds, Fetched: 1 row(s)
```

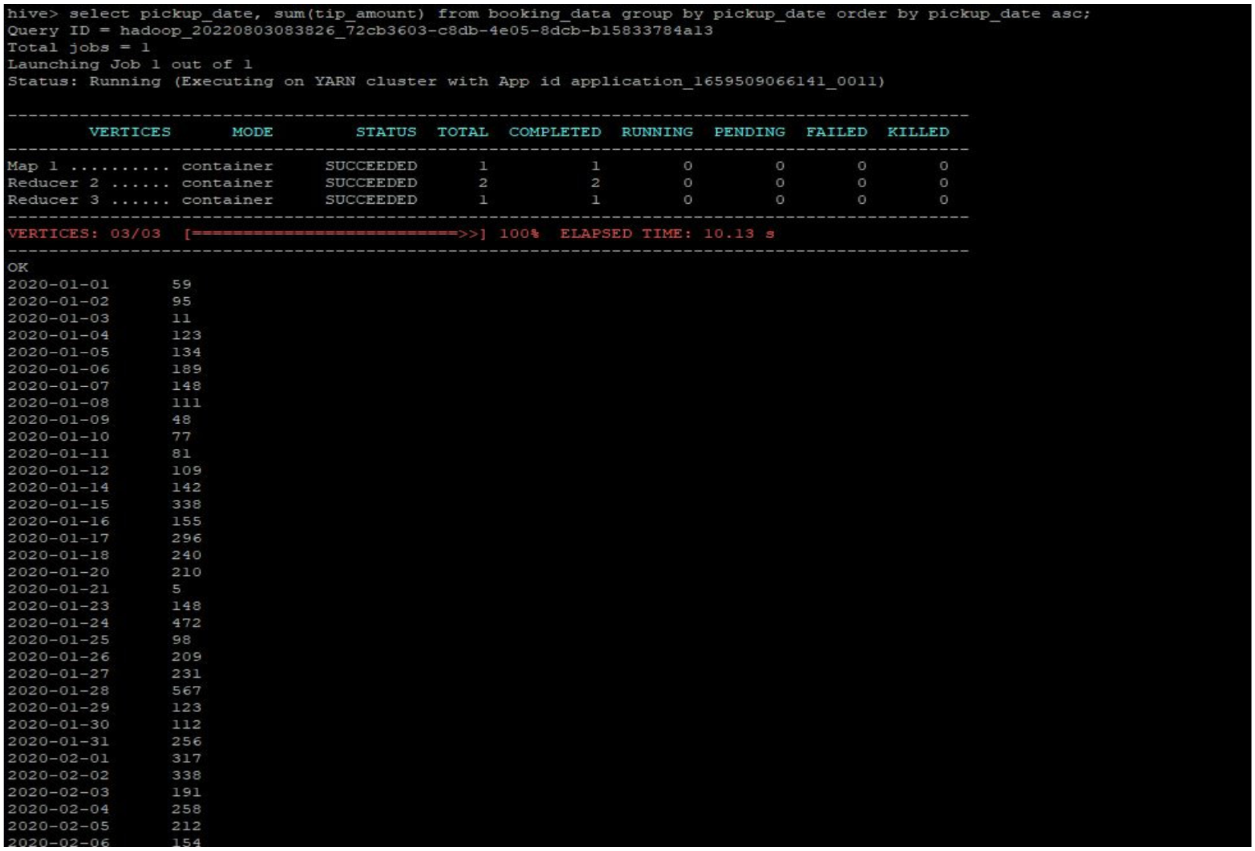
- **Calculate the total amount of tips given date wise to all drivers by customers.**

# Query

```
SELECT pickup_date,
       Sum(tip_amount)
FROM   booking
GROUP  BY pickup_date
ORDER  BY pickup_date ASC;
```

# Explanation
In this task, we selected 2 columns, pickup_date and tip_amount from the booking table and then we ordered this by pickup_date in ascending order. We used the SUM() function for the tip_amount column to return the total amount of tips given. In this way we can get the details required in task 9.

Screenshot showing query run



```
hive> select pickup_date, sum(tip_amount) from booking_data group by pickup_date order by pickup_date asc;
Query ID = hadoop_20220803083826_72cb3603-c8db-4e05-8dcb-b15833784a13
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1659509066141_0011)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      2          2        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 10.13 s
----------------------------------------------------------------------------------------------
OK
2020-01-01      59
2020-01-02      95
2020-01-03      11
2020-01-04      123
2020-01-05      134
2020-01-06      189
2020-01-07      148
2020-01-08      111
2020-01-09      48
2020-01-10      77
2020-01-11      81
2020-01-12      109
2020-01-14      142
2020-01-15      338
2020-01-16      155
2020-01-17      296
2020-01-18      240
2020-01-20      210
2020-01-21      5
2020-01-23      148
2020-01-24      472
2020-01-25      98
2020-01-26      209
2020-01-27      231
2020-01-28      567
2020-01-29      123
2020-01-30      112
2020-01-31      256
2020-02-01      317
2020-02-02      338
2020-02-03      191
2020-02-04      258
2020-02-05      212
2020-02-06      154
```

- **Calculate the total count of all the bookings with ratings lower than 2 as given by customers in a particular month**

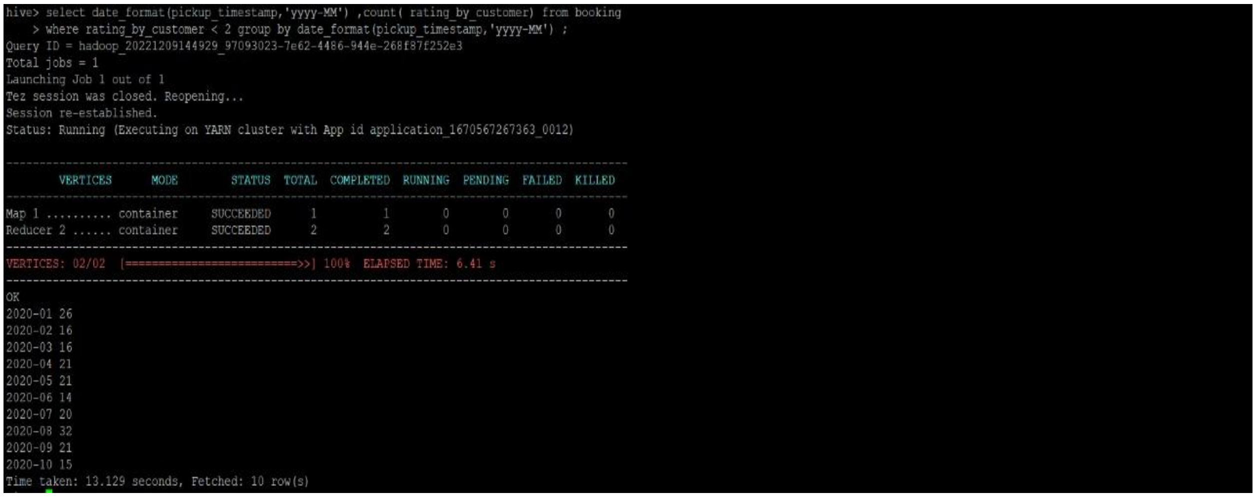
  # Query

```
SELECT Date_format(pickup_timestamp, 'yyyy-mm'),
       Count(rating_by_customer)
FROM   booking
WHERE  rating_by_customer < 2
GROUP  BY Date_format(pickup_timestamp, 'yyyy- mm');
```

# Explanation

In this task, we selected 2 columns, pickup_timestamp and rating_by_customer from booking table and we used where clause to filter the data where rating_by_customer < 2 and then we grouped this by date_format(pickup_timestamp,'yyyy-mm'). We used DATE_FORMAT() function to extract the pickup date for pickup_timestamp column in ‘yyyy-mm’ format and COUNT() function for rating_by_customer column to count the number of rows returned.In this way we can get the details required in task 10

Screenshot showing query run



```
hive> select date_format(pickup_timestamp,'yyyy-MM') ,count( rating_by_customer) from booking
    > where rating_by_customer < 2 group by date_format(pickup_timestamp,'yyyy-MM') ;
Query ID = hadoop_20221209144929_97093023-7e62-4486-944e-268f87f252e3
Total jobs = 1
Launching Job 1 out of 1
Tez session was closed. Reopening...
Session re-established.
Status: Running (Executing on YARN cluster with App id application_1670567267363_0012)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      2          2        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 6.41 s
----------------------------------------------------------------------------------------------
OK
2020-01 26
2020-02 16
2020-03 16
2020-04 21
2020-05 21
2020-06 14
2020-07 20
2020-08 32
2020-09 21
2020-10 15
Time taken: 13.129 seconds, Fetched: 10 row(s)
```

- **Calculate the count of total iOS users.**

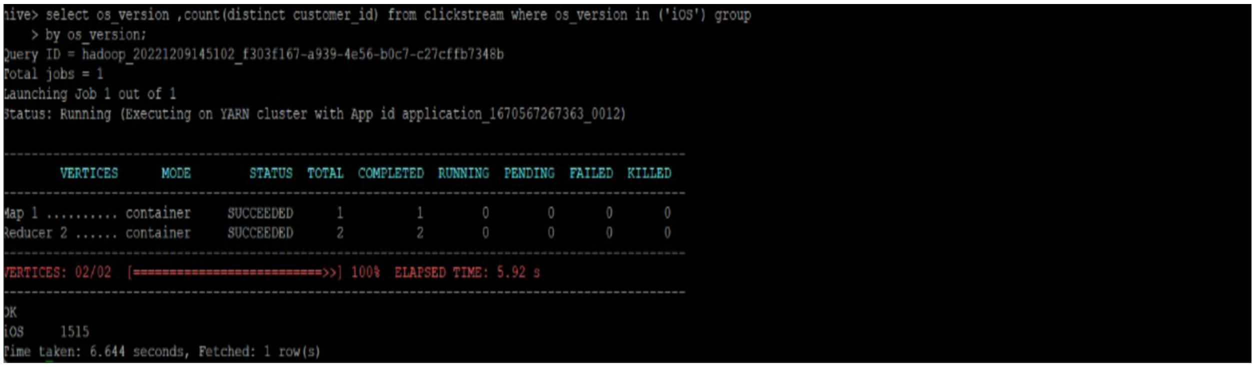
# Query

```
SELECT os_version,
       Count(DISTINCT customer_id)
FROM   clickstream
WHERE  os_version IN ( 'iOS' )
GROUP  BY os_version;
```

# Explanation

In this task, we selected 2 columns, os_version and customer_id from the clickstream_data table and we used the WHERE clause to filter the data where os_version is ‘iOS’ and then we grouped this by os_version. We used the COUNT() function for the customer_id column to count the number of rows returned with DISTINCT to select unique rows from the table clickstream. In this way we can get the details required in task 11

Screenshot showing query run



```
hive> select os_version ,count(distinct customer_id) from clickstream where os_version in ('iOS') group
    > by os_version;
Query ID = hadoop_20221209145102_f303f167-a939-4e56-b0c7-c27cffb7348b
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1670567267363_0012)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      2          2        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 5.92 s
----------------------------------------------------------------------------------------------
OK
iOS     1515
Time taken: 6.644 seconds, Fetched: 1 row(s)
```