

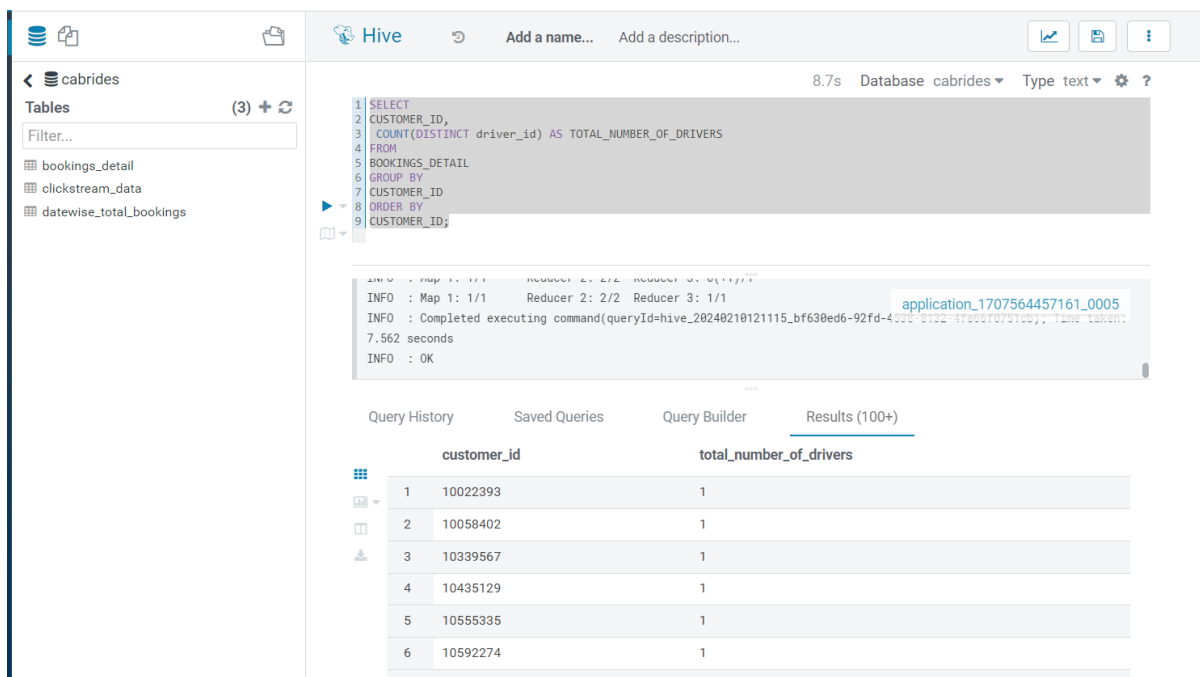
Queries

Task 5: Calculate the total number of different drivers for each customer

QUERY:

```
SELECT CUSTOMER_ID, COUNT(DISTINCT driver_id) AS TOTAL_NUMBER_OF_DRIVERS
FROM BOOKINGS_DETAIL
GROUP BY CUSTOMER_ID
ORDER BY CUSTOMER_ID;
```

OUTPUT:



The screenshot shows the Hive query interface. On the left, there is a sidebar with a table list under the 'cabrides' database, including 'bookings_detail', 'clickstream_data', and 'datewise_total_bookings'. The main area displays the SQL query: `SELECT CUSTOMER_ID, COUNT(DISTINCT driver_id) AS TOTAL_NUMBER_OF_DRIVERS FROM BOOKINGS_DETAIL GROUP BY CUSTOMER_ID ORDER BY CUSTOMER_ID;`. Below the query, the execution status is shown as 'INFO : Map 1: 1/1 Reducer 2: 2/2 Reducer 3: 1/1' and 'INFO : Completed executing command(queryId=hive_20240210121115_bf630ed6-92fd-4330-9122-9f000107510b); Time taken: 7.562 seconds'. The results are displayed in a table with columns 'customer_id' and 'total_number_of_drivers'.

customer_id	total_number_of_drivers
10022393	1
10058402	1
10339567	1
10435129	1
10555335	1
10592274	1

VALIDATION: Exact Match

1. When you run the query to calculate the total number of different drivers for each customer, you would get an output as shown below:

```
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2020-11-17 12:23:06,034 Stage-1 map = 0%, reduce = 0%
2020-11-17 12:23:12,394 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 3.27 sec
2020-11-17 12:23:20,727 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 7.69 sec
MapReduce Total cumulative CPU time: 7 seconds 690 msec
Ended Job = job_1605615116654_0005
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 7.69 sec HDFS Read: 43007 HDFS Write: 11000 SUCCESS
Total MapReduce CPU Time Spent: 7 seconds 690 msec
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
11600791      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
13015449      1
13229062      1
```

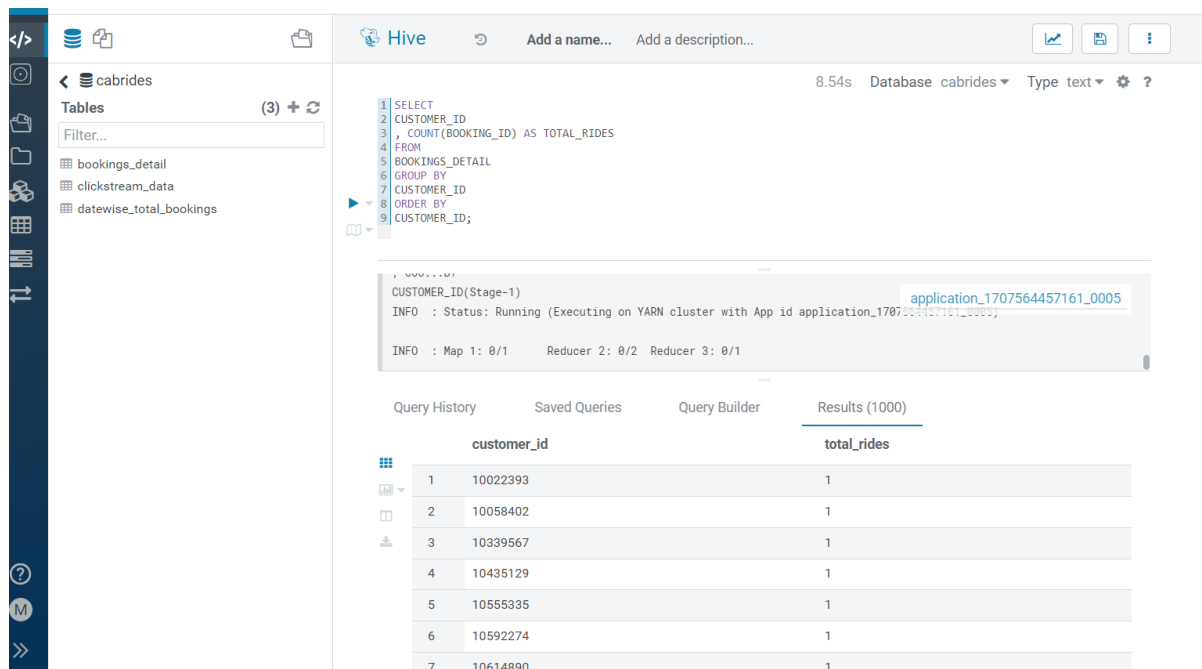
• Task 6:

Calculate the total rides taken by each customer.

QUERY:

```
SELECT CUSTOMER_ID, COUNT(BOOKING_ID) AS TOTAL_RIDES
FROM BOOKINGS_DETAIL
GROUP BY CUSTOMER_ID
ORDER BY CUSTOMER_ID;
```

OUTPUT:



The screenshot shows the Hive query interface. On the left, there's a sidebar with a file explorer and a table list for the 'cabrides' database, including 'bookings_detail', 'clickstream_data', and 'datewise_total_bookings'. The main area displays a SQL query:

```
1 SELECT
2 CUSTOMER_ID
3 , COUNT(BOOKING_ID) AS TOTAL_RIDES
4 FROM
5 BOOKINGS_DETAIL
6 GROUP BY
7 CUSTOMER_ID
8 ORDER BY
9 CUSTOMER_ID;
```

Below the query, the execution status is shown: 'INFO : Status: Running (Executing on YARN cluster with App id application_1707564457161_0005)'. The results are displayed in a table with 1000 rows (shown as 7 in the screenshot):

customer_id	total_rides
10022393	1
10058402	1
10339567	1
10435129	1
10555335	1
10592274	1
10614890	1

VALIDATION: Exact Match

```
Ended Job = job_1605615116654_0008
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 6.65 sec HDFS Read: 38721 HDFS Write: 11000 SUCCESS
Total MapReduce CPU Time Spent: 6 seconds 650 msec
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
12334666      1
```

• Task 7:

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 'fcb68aa-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 PAGE_ID = 'e7bc5fb2-1231-11eb-adc1-0242ac120002' THEN 1 ELSE 0 END) AS

TOTAL_PAGE_VISITS,

SUM(CASE WHEN BUTTON_ID = 'fcb68aa-1231-11eb-adc1-0242ac120002' THEN 1 ELSE 0 END) AS

TOTAL_BUTTON_PRESSED,

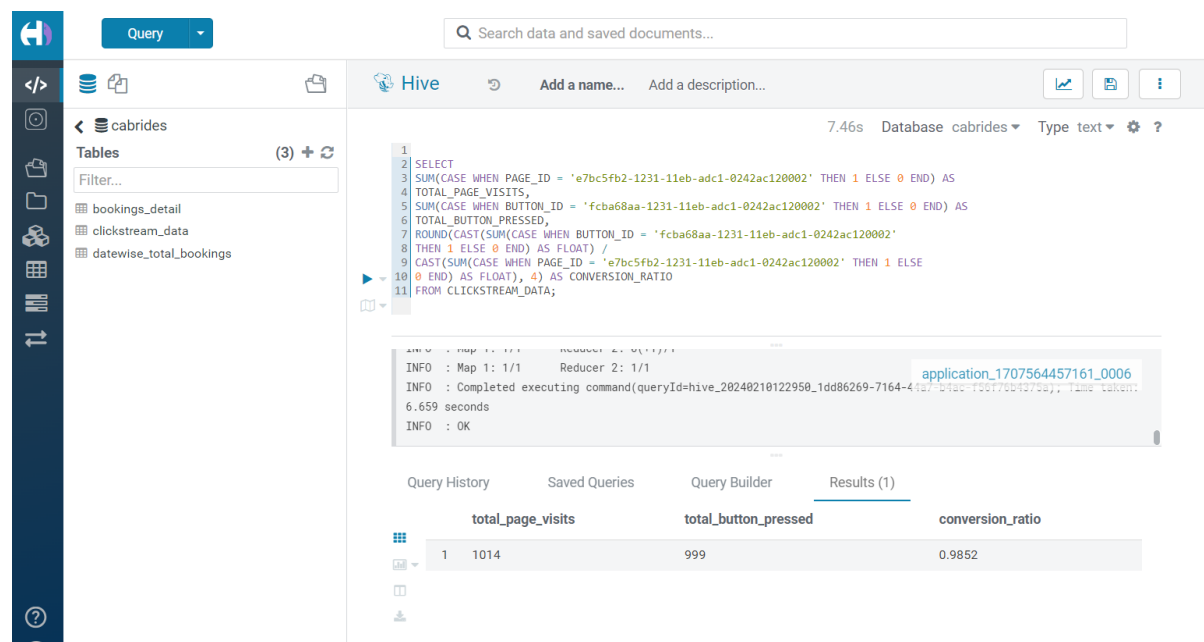
ROUND(CAST(SUM(CASE WHEN BUTTON_ID = 'fcb68aa-1231-11eb-adc1-0242ac120002' THEN 1 ELSE 0 END) AS FLOAT) /

CAST(SUM(CASE WHEN PAGE_ID = 'e7bc5fb2-1231-11eb-adc1-0242ac120002' THEN 1 ELSE

0 END) AS FLOAT), 4) AS CONVERSION_RATIO

FROM CLICKSTREAM_DATA;

OUTPUT:



The screenshot shows the Hive query execution interface. The query is displayed in the top panel, and the results are shown in the bottom panel. The results table has three columns: total_page_visits, total_button_pressed, and conversion_ratio. The values are 1014, 999, and 0.9852 respectively.

	total_page_visits	total_button_pressed	conversion_ratio
1	1014	999	0.9852

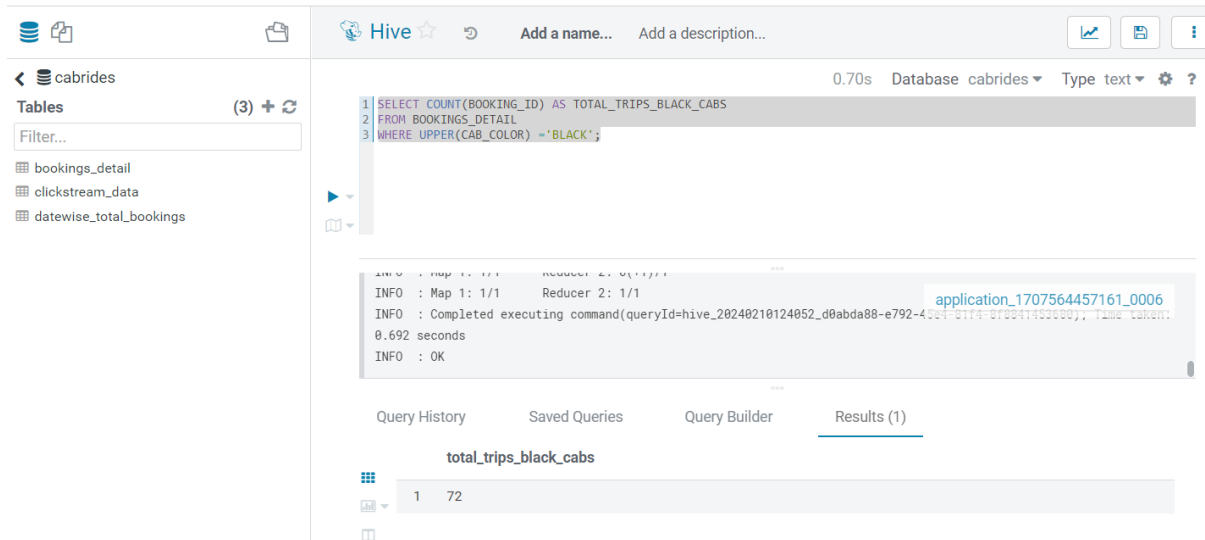
VALIDATION: close to Match (0.9852), since Kafka had extra 16 records compare to validation it should get the conversion ratio as 0.9688.

- **Task 8:** Calculate the count of all trips done on black cabs.

QUERY:

```
SELECT COUNT(BOOKING_ID) AS TOTAL_TRIPS_BLACK_CABS
FROM BOOKINGS_DETAIL
WHERE UPPER(CAB_COLOR) = 'BLACK';
```

OUTPUT:



The screenshot shows the Hive query execution interface. The query is: `SELECT COUNT(BOOKING_ID) AS TOTAL_TRIPS_BLACK_CABS FROM BOOKINGS_DETAIL WHERE UPPER(CAB_COLOR) = 'BLACK';`. The execution time is 0.70s. The results are displayed in a table with one row: `total_trips_black_cabs` with a value of 72.

VALIDATION: Exact Match

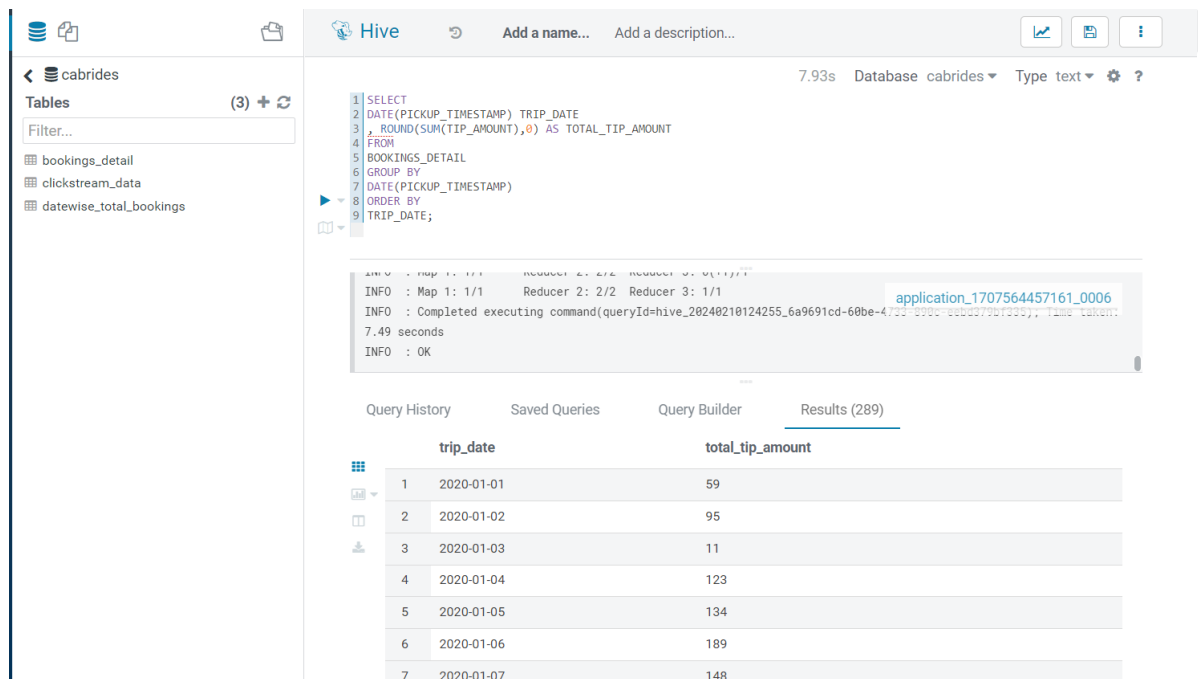
Count of all trips done on black cabs -72.

- **Task 9:** Calculate the total amount of tips given date wise to all drivers by customers.

QUERY:

```
SELECT DATE(PICKUP_TIMESTAMP) TRIP_DATE
, ROUND(SUM(TIP_AMOUNT),0) AS TOTAL_TIP_AMOUNT
FROM BOOKINGS_DETAIL
GROUP BY DATE(PICKUP_TIMESTAMP)
ORDER BY TRIP_DATE;
```

OUTPUT:



The screenshot shows the Hive query interface. On the left, a sidebar lists tables: bookings_detail, clickstream_data, and datewise_total_bookings. The main area displays a SQL query:

```
1 SELECT
2 DATE(PICKUP_TIMESTAMP) TRIP_DATE
3 , ROUND(SUM(TIP_AMOUNT),0) AS TOTAL_TIP_AMOUNT
4 FROM
5 BOOKINGS_DETAIL
6 GROUP BY
7 DATE(PICKUP_TIMESTAMP)
8 ORDER BY
9 TRIP_DATE;
```

Below the query, the execution progress is shown: Map 1: 1/1, Reducer 2: 2/2, Reducer 3: 1/1. The status is "Completed executing command(queryId=hive_20240210124255_6a9691cd-60be-4793-b00c-b00079079393), Time taken: 7.49 seconds". The application ID is application_1707564457161_0006.

At the bottom, the "Results (289)" tab is active, showing a table with two columns: trip_date and total_tip_amount. The results are as follows:

trip_date	total_tip_amount
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

VALIDATION: Exact Match

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

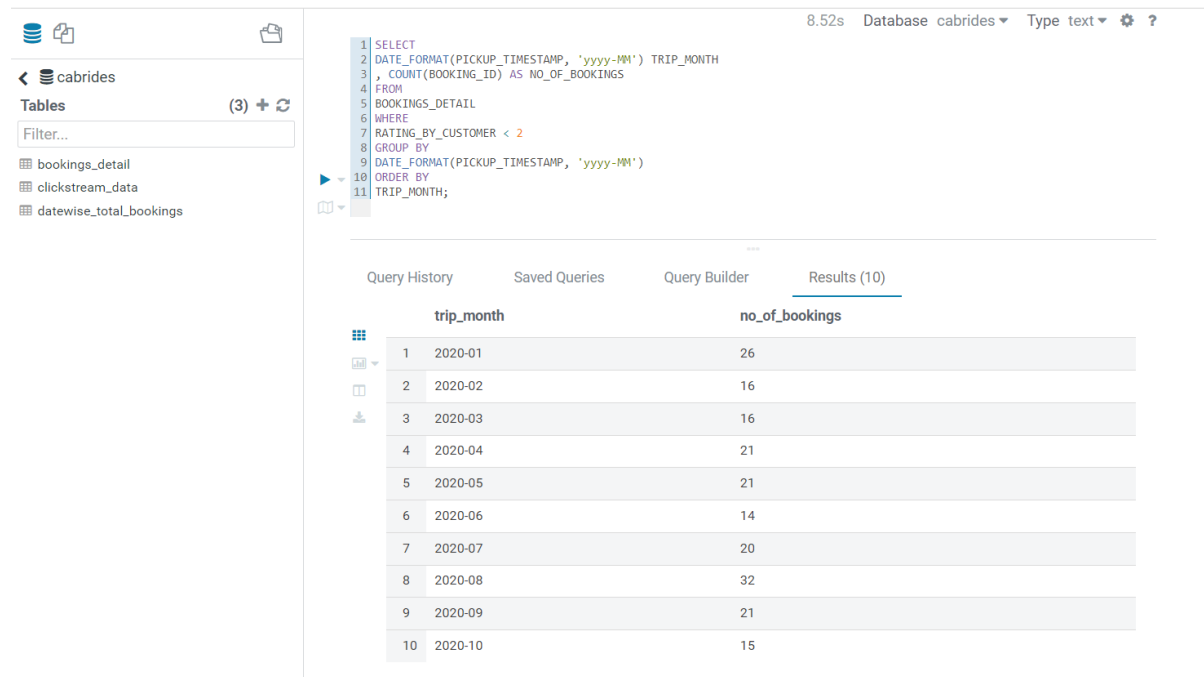
• Task 10:

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') TRIP_MONTH
, COUNT(BOOKING_ID) AS NO_OF_BOOKINGS
FROM BOOKINGS_DETAIL
WHERE RATING_BY_CUSTOMER < 2
GROUP BY DATE_FORMAT(PICKUP_TIMESTAMP, 'yyyy-MM')
ORDER BY TRIP_MONTH;
```

OUTPUT:



The screenshot shows a SQL query editor interface. On the left, there's a sidebar with a database icon and a list of tables: bookings_detail, clickstream_data, and datewise_total_bookings. The main area displays the SQL query:

```
1 SELECT
2 DATE_FORMAT(PICKUP_TIMESTAMP, 'yyyy-MM') TRIP_MONTH
3 , COUNT(BOOKING_ID) AS NO_OF_BOOKINGS
4 FROM
5 BOOKINGS_DETAIL
6 WHERE
7 RATING_BY_CUSTOMER < 2
8 GROUP BY
9 DATE_FORMAT(PICKUP_TIMESTAMP, 'yyyy-MM')
10 ORDER BY
11 TRIP_MONTH;
```

 Below the query, there's a tab labeled 'Results (10)' which shows a table with two columns: 'trip_month' and 'no_of_bookings'. The table contains 10 rows of data for the months from 2020-01 to 2020-10.

trip_month	no_of_bookings
1 2020-01	26
2 2020-02	16
3 2020-03	16
4 2020-04	21
5 2020-05	21
6 2020-06	14
7 2020-07	20
8 2020-08	32
9 2020-09	21
10 2020-10	15

VALIDATION: Exact Match

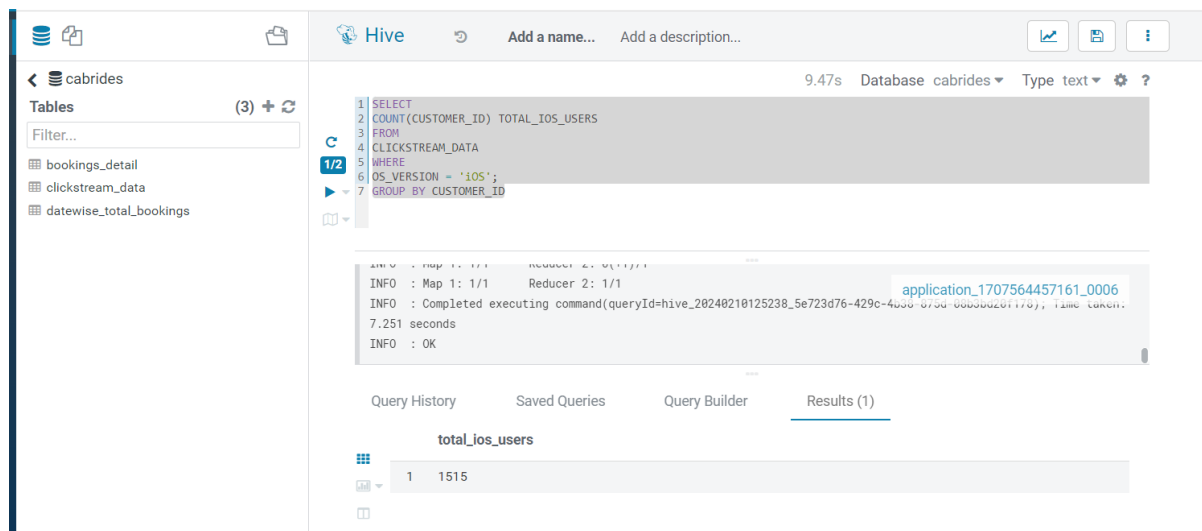
```
Total MapReduce CPU Time Spent: 7 seconds 970 msec
OK
2020-01 26
2020-02 16
2020-03 16
2020-04 21
2020-05 21
2020-06 14
2020-07 20
2020-08 32
2020-09 21
2020-10 15
```

- **Task 11:** Calculate the count of total iOS users.

QUERY:

```
SELECT COUNT(CUSTOMER_ID) TOTAL_IOS_USERS
FROM CLICKSTREAM_DATA
WHERE OS_VERSION = 'iOS';
GROUP BY CUSTOMER_ID;
```

OUTPUT:



The screenshot shows the Hive query interface. On the left, there is a sidebar with a list of tables: bookings_detail, clickstream_data, and datewise_total_bookings. The main area displays the query: `SELECT COUNT(CUSTOMER_ID) TOTAL_IOS_USERS FROM CLICKSTREAM_DATA WHERE OS_VERSION = 'iOS'; GROUP BY CUSTOMER_ID;`. Below the query, the execution progress is shown as 1/2. The results section shows a single row with the value 1515 for the column total_ios_users. The status bar at the bottom indicates that the query was completed successfully.

VALIDATION: close to Match(1515) , since Kafka had extra 16 records compare to validation
You should get the count of all iOS users as 1503.