Yellow taxi trip records analysis by Sean Hensel-Coe

Create a new database and two new table called locations and *trip_data* to insert the dataset using CREATE TABLE and relative parameters. Then, load the data into each table using LOAD DATA LOCAL INFILE and INTO TABLE.

```
CREATE DATABASE yellow_taxi;

CREATE TABLE locations(LocationID INT, Borough VARCHAR(100), Zone VARCHAR(100), service_zone VARCHAR(100));

LOAD DATA LOCAL INFILE 'taxi_zone_lookup.csv'
INTO TABLE locations
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;
```

```
CREATE TABLE trip_data(VendorID INT, tpep_pickup_datetime DATETIME, tpep_dropoff_datetime DATETIME, passenger_count INT,
trip_distance FLOAT, RatecodeID INT, store_and_fwd_flag VARCHAR(2), PULocationID INT, DOLocationID INT, payment_type INT, fare_amount
FLOAT, extra FLOAT, mta_tax FLOAT, tip_amount FLOAT, tolls_amount FLOAT, improvement_surcharge FLOAT, total_amount FLOAT,
congestion_surcharge FLOAT
);

LOAD DATA LOCAL INFILE 'yellow_tripdata_2020-04.csv'
INTO TABLE trip_data
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;
```

Selecting the *trip_data* table

```
mysql> USE yellow_taxi;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----------------------+
| Tables_in_yellow_taxi |
+-----------------------+
| locations             |
| trip_data             |
+-----------------------+
2 rows in set (0.00 sec)
```

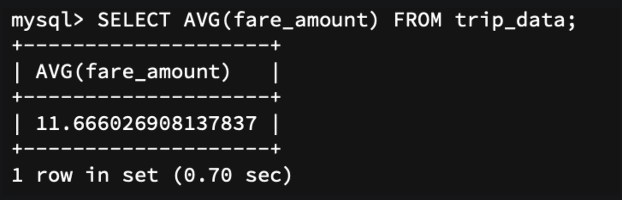
Describing the *trip_data* table

```
mysql> DESC trip_data;
+-----------------------+------------+------+-----+---------+-------+
| Field                 | Type       | Null | Key | Default | Extra |
+-----------------------+------------+------+-----+---------+-------+
| VendorID              | int(11)    | YES  |     | NULL    |       |
| tpep_pickup_datetime  | datetime   | YES  |     | NULL    |       |
| tpep_dropoff_datetime | datetime   | YES  |     | NULL    |       |
| passenger_count       | int(11)    | YES  |     | NULL    |       |
| trip_distance         | float      | YES  |     | NULL    |       |
| RatecodeID            | int(11)    | YES  |     | NULL    |       |
| store_and_fwd_flag    | varchar(2) | YES  |     | NULL    |       |
| PULocationID          | int(11)    | YES  |     | NULL    |       |
| DOLocationID          | int(11)    | YES  |     | NULL    |       |
| payment_type          | int(11)    | YES  |     | NULL    |       |
| fare_amount           | float      | YES  |     | NULL    |       |
| extra                 | float      | YES  |     | NULL    |       |
| mta_tax               | float      | YES  |     | NULL    |       |
| tip_amount            | float      | YES  |     | NULL    |       |
| tolls_amount          | float      | YES  |     | NULL    |       |
| improvement_surcharge | float      | YES  |     | NULL    |       |
| total_amount          | float      | YES  |     | NULL    |       |
| congestion_surcharge  | float      | YES  |     | NULL    |       |
+-----------------------+------------+------+-----+---------+-------+
```

Selecting all rows and columns of *trip_data* after import

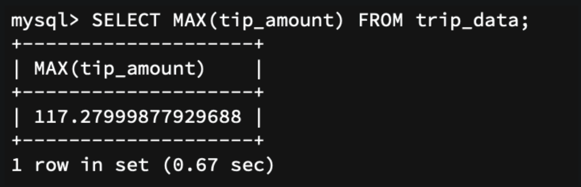
```
mysql> SELECT * FROM trip_data LIMIT 5;
+----------+----------------------+-----------------------+-----------------+---------------+------------+--------------------+--------------+------
--------+--------------+-------------+-------+---------+------------+--------------+-----------------------+--------------+----------------------+
| VendorID | tpep_pickup_datetime | tpep_dropoff_datetime | passenger_count | trip_distance | RatecodeID | store_and_fwd_flag | PULocationID | DOLoc
ationID | payment_type | fare_amount | extra | mta_tax | tip_amount | tolls_amount | improvement_surcharge | total_amount | congestion_surcharge |
+----------+----------------------+-----------------------+-----------------+---------------+------------+--------------------+--------------+------
--------+--------------+-------------+-------+---------+------------+--------------+-----------------------+--------------+----------------------+
|        1 | 2020-04-01 00:41:22  | 2020-04-01 01:01:53   |               1 |           1.2 |          1 | N                  |           41 |
      24 |            2 |         5.5 |   0.5 |     0.5 |          0 |            0 |                   0.3 |          6.8 |                    0 |
|        1 | 2020-04-01 00:56:00  | 2020-04-01 01:09:25   |               1 |           3.4 |          1 | N                  |           95 |
     197 |            1 |        12.5 |   0.5 |     0.5 |       2.75 |            0 |                   0.3 |        16.55 |                    0 |
|        1 | 2020-04-01 00:00:26  | 2020-04-01 00:09:25   |               1 |           2.8 |          1 | N                  |          237 |
     137 |            1 |          10 |     3 |     0.5 |          1 |            0 |                   0.3 |         14.8 |                  2.5 |
|        1 | 2020-04-01 00:24:38  | 2020-04-01 00:34:38   |               0 |           2.6 |          1 | N                  |           68 |
     142 |            1 |          10 |     3 |     0.5 |          1 |            0 |                   0.3 |         14.8 |                  2.5 |
|        2 | 2020-04-01 00:13:24  | 2020-04-01 00:18:26   |               1 |          1.44 |          1 | Y                  |          263 |
      74 |            1 |         6.5 |   0.5 |     0.5 |          3 |            0 |                   0.3 |         13.3 |                  2.5 |
+----------+----------------------+-----------------------+-----------------+---------------+------------+--------------------+--------------+------
--------+--------------+-------------+-------+---------+------------+--------------+-----------------------+--------------+----------------------+
5 rows in set (0.00 sec)
```

Average fare amount



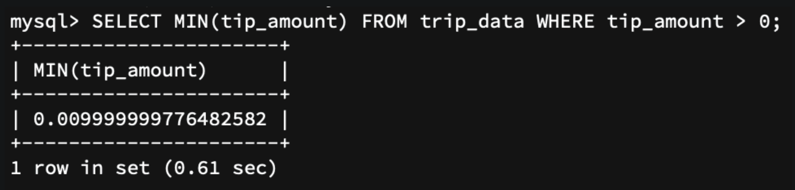
```
mysql> SELECT AVG(fare_amount) FROM trip_data;
+--------------------+
| AVG(fare_amount)   |
+--------------------+
| 11.666026908137837 |
+--------------------+
1 row in set (0.70 sec)
```

Largest tip



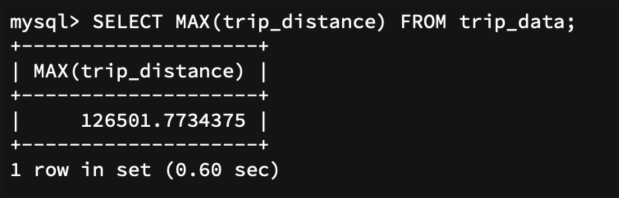
```
mysql> SELECT MAX(tip_amount) FROM trip_data;
+--------------------+
| MAX(tip_amount)    |
+--------------------+
| 117.27999877929688 |
+--------------------+
1 row in set (0.67 sec)
```

Lowest tip that does not equal zero



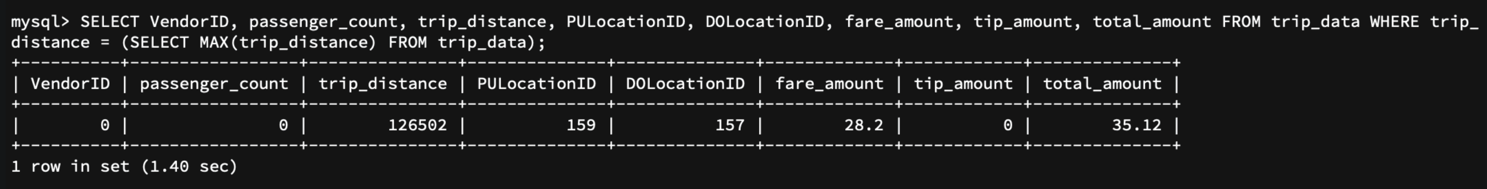
```
mysql> SELECT MIN(tip_amount) FROM trip_data WHERE tip_amount > 0;
+----------------------+
| MIN(tip_amount)      |
+----------------------+
| 0.009999999776482582 |
+----------------------+
1 row in set (0.61 sec)
```

Longest distance travelled



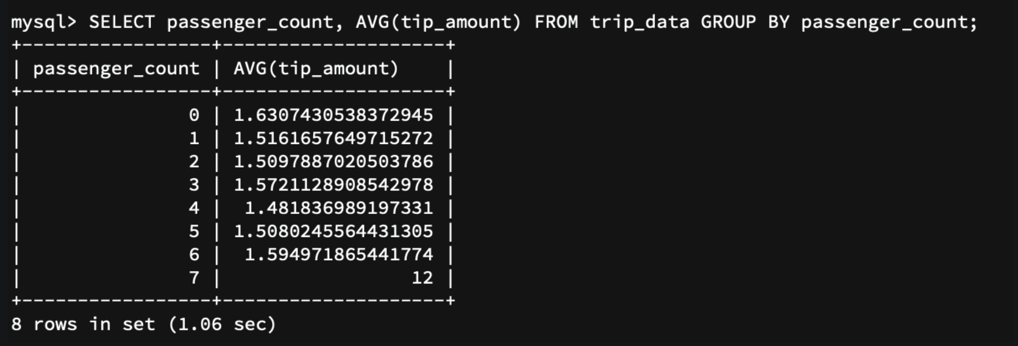
```
mysql> SELECT MAX(trip_distance) FROM trip_data;
+-------------------+
| MAX(trip_distance) |
+-------------------+
|     126501.7734375 |
+-------------------+
1 row in set (0.60 sec)
```

This seems like an extremely long trip. Further investigations illustrates a lot of missing information.



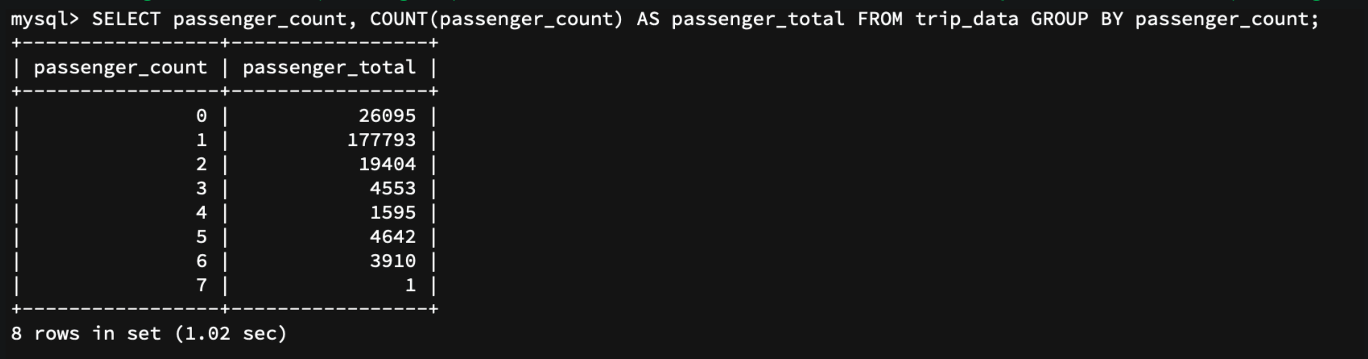
```
mysql> SELECT VendorID, passenger_count, trip_distance, PULocationID, DOLocationID, fare_amount, tip_amount, total_amount FROM trip_data WHERE trip_
distance = (SELECT MAX(trip_distance) FROM trip_data);
+----------+-----------------+---------------+--------------+--------------+-------------+------------+--------------+
| VendorID | passenger_count | trip_distance | PULocationID | DOLocationID | fare_amount | tip_amount | total_amount |
+----------+-----------------+---------------+--------------+--------------+-------------+------------+--------------+
|        0 |               0 |        126502 |          159 |          157 |        28.2 |          0 |        35.12 |
+----------+-----------------+---------------+--------------+--------------+-------------+------------+--------------+
1 row in set (1.40 sec)
```

The average tips stay roughly equal, except for 7 passengers which has a very high mean



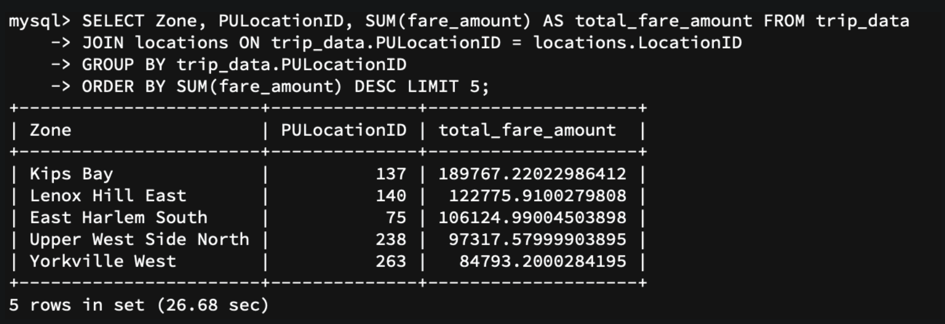
```
mysql> SELECT passenger_count, AVG(tip_amount) FROM trip_data GROUP BY passenger_count;
+-----------------+--------------------+
| passenger_count | AVG(tip_amount)    |
+-----------------+--------------------+
|               0 | 1.6307430538372945 |
|               1 | 1.5161657649715272 |
|               2 | 1.5097887020503786 |
|               3 | 1.5721128908542978 |
|               4 |  1.481836989197331 |
|               5 | 1.5080245564431305 |
|               6 |  1.594971865441774 |
|               7 |                 12 |
+-----------------+--------------------+
8 rows in set (1.06 sec)
```

After counting the number of passengers per ride, we can see that this is an outlier as we only had one ride with 7 passengers



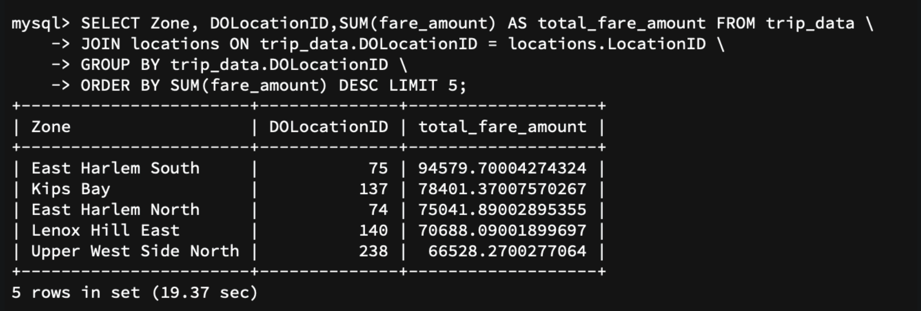
```
mysql> SELECT passenger_count, COUNT(passenger_count) AS passenger_total FROM trip_data GROUP BY passenger_count;
+-----------------+-----------------+
| passenger_count | passenger_total |
+-----------------+-----------------+
|               0 |           26095 |
|               1 |          177793 |
|               2 |           19404 |
|               3 |            4553 |
|               4 |            1595 |
|               5 |            4642 |
|               6 |            3910 |
|               7 |               1 |
+-----------------+-----------------+
8 rows in set (1.02 sec)
```

By joining the two tables, we can see below the most lucrative pickup and dropoff locations.
Pickup



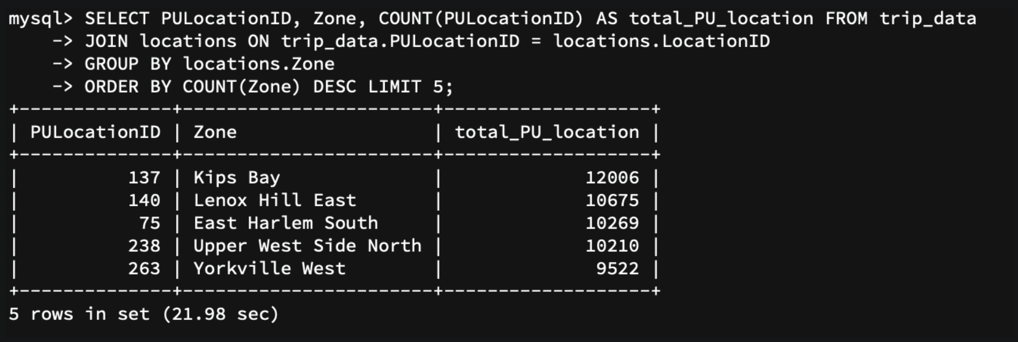
```
mysql> SELECT Zone, PULocationID, SUM(fare_amount) AS total_fare_amount FROM trip_data
    -> JOIN locations ON trip_data.PULocationID = locations.LocationID
    -> GROUP BY trip_data.PULocationID
    -> ORDER BY SUM(fare_amount) DESC LIMIT 5;
+-----------------------+--------------+--------------------+
| Zone                  | PULocationID | total_fare_amount  |
+-----------------------+--------------+--------------------+
| Kips Bay              |          137 | 189767.22022986412 |
| Lenox Hill East       |          140 |  122775.9100279808 |
| East Harlem South     |           75 | 106124.99004503898 |
| Upper West Side North |          238 |  97317.57999903895 |
| Yorkville West        |          263 |   84793.2000284195 |
+-----------------------+--------------+--------------------+
5 rows in set (26.68 sec)
```

Dropoff



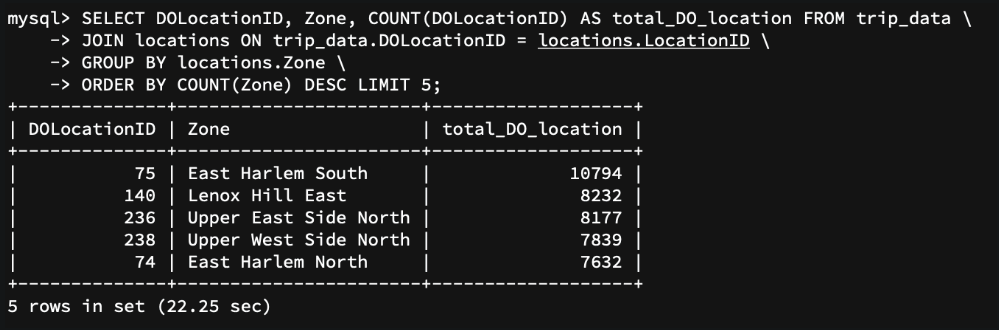
```
mysql> SELECT Zone, DOLocationID,SUM(fare_amount) AS total_fare_amount FROM trip_data \
    -> JOIN locations ON trip_data.DOLocationID = locations.LocationID \
    -> GROUP BY trip_data.DOLocationID \
    -> ORDER BY SUM(fare_amount) DESC LIMIT 5;
+-----------------------+--------------+-------------------+
| Zone                  | DOLocationID | total_fare_amount |
+-----------------------+--------------+-------------------+
| East Harlem South     |           75 | 94579.70004274324 |
| Kips Bay              |          137 | 78401.37007570267 |
| East Harlem North     |           74 | 75041.89002895355 |
| Lenox Hill East       |          140 | 70688.09001899697 |
| Upper West Side North |          238 |  66528.2700277064 |
+-----------------------+--------------+-------------------+
5 rows in set (19.37 sec)
```

By joining the two tables, we can see that the most lucrative pickup and dropoff spots are also the most frequented.
Pickup



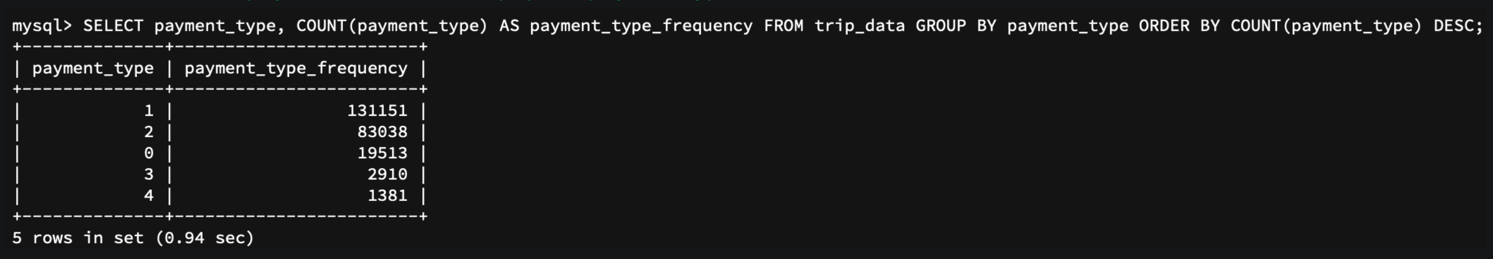
```
mysql> SELECT PULocationID, Zone, COUNT(PULocationID) AS total_PU_location FROM trip_data
    -> JOIN locations ON trip_data.PULocationID = locations.LocationID
    -> GROUP BY locations.Zone
    -> ORDER BY COUNT(Zone) DESC LIMIT 5;
+--------------+-----------------------+-------------------+
| PULocationID | Zone                  | total_PU_location |
+--------------+-----------------------+-------------------+
|          137 | Kips Bay              |             12006 |
|          140 | Lenox Hill East       |             10675 |
|           75 | East Harlem South     |             10269 |
|          238 | Upper West Side North |             10210 |
|          263 | Yorkville West        |              9522 |
+--------------+-----------------------+-------------------+
5 rows in set (21.98 sec)
```

Dropoff



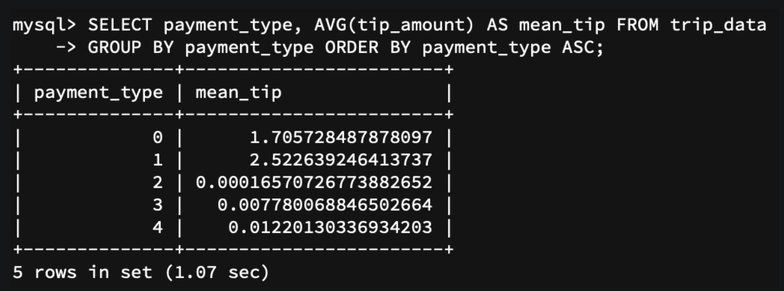
```
mysql> SELECT DOLocationID, Zone, COUNT(DOLocationID) AS total_DO_location FROM trip_data \
    -> JOIN locations ON trip_data.DOLocationID = locations.LocationID \
    -> GROUP BY locations.Zone \
    -> ORDER BY COUNT(Zone) DESC LIMIT 5;
+--------------+-----------------------+-------------------+
| DOLocationID | Zone                  | total_DO_location |
+--------------+-----------------------+-------------------+
|           75 | East Harlem South     |             10794 |
|          140 | Lenox Hill East       |              8232 |
|          236 | Upper East Side North |              8177 |
|          238 | Upper West Side North |              7839 |
|           74 | East Harlem North     |              7632 |
+--------------+-----------------------+-------------------+
5 rows in set (22.25 sec)
```

Here we see that card payments are the most popular payment type



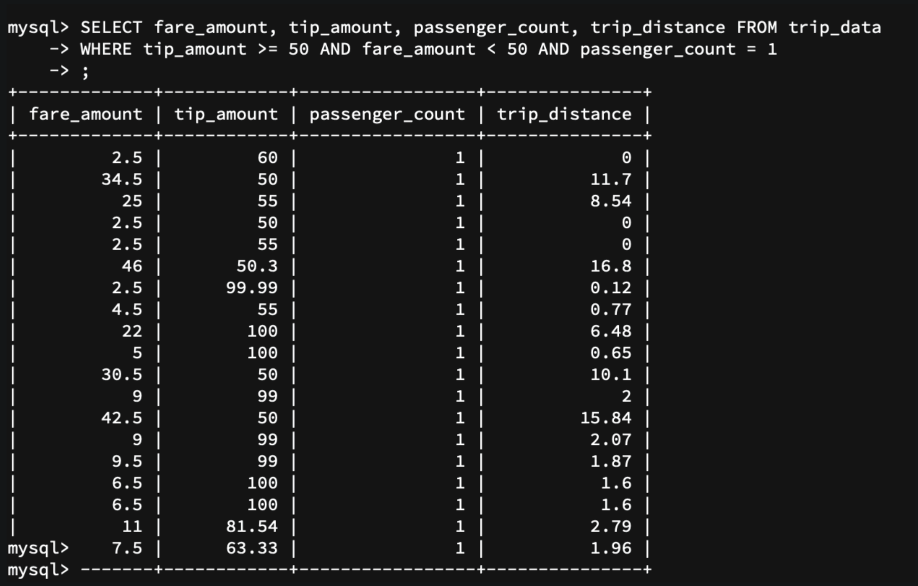
```
mysql> SELECT payment_type, COUNT(payment_type) AS payment_type_frequency FROM trip_data GROUP BY payment_type ORDER BY COUNT(payment_type) DESC;
+--------------+------------------------+
| payment_type | payment_type_frequency |
+--------------+------------------------+
|            1 |                 131151 |
|            2 |                  83038 |
|            0 |                  19513 |
|            3 |                   2910 |
|            4 |                   1381 |
+--------------+------------------------+
5 rows in set (0.94 sec)
```

The average tip amount is much higher when passengers pay by card



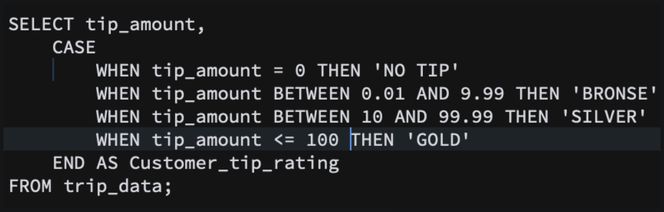
```
mysql> SELECT payment_type, AVG(tip_amount) AS mean_tip FROM trip_data
    -> GROUP BY payment_type ORDER BY payment_type ASC;
+--------------+------------------------+
| payment_type | mean_tip               |
+--------------+------------------------+
|            0 |      1.705728487878097 |
|            1 |      2.522639246413737 |
|            2 | 0.00016570726773882652 |
|            3 |   0.007780068846502664 |
|            4 |    0.01220130336934203 |
+--------------+------------------------+
5 rows in set (1.07 sec)
```

Tip data tip amount is equal to or greater than $50 and the fare amount is less than $50

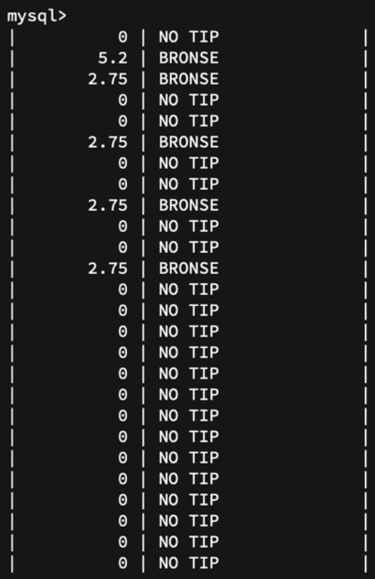


```
mysql> SELECT fare_amount, tip_amount, passenger_count, trip_distance FROM trip_data
    -> WHERE tip_amount >= 50 AND fare_amount < 50 AND passenger_count = 1
    -> ;
+-------------+------------+-----------------+---------------+
| fare_amount | tip_amount | passenger_count | trip_distance |
+-------------+------------+-----------------+---------------+
|         2.5 |         60 |               1 |             0 |
|        34.5 |         50 |               1 |          11.7 |
|          25 |         55 |               1 |          8.54 |
|         2.5 |         50 |               1 |             0 |
|         2.5 |         55 |               1 |             0 |
|          46 |       50.3 |               1 |          16.8 |
|         2.5 |      99.99 |               1 |          0.12 |
|         4.5 |         55 |               1 |          0.77 |
|          22 |        100 |               1 |          6.48 |
|           5 |        100 |               1 |          0.65 |
|        30.5 |         50 |               1 |          10.1 |
|           9 |         99 |               1 |             2 |
|        42.5 |         50 |               1 |         15.84 |
|           9 |         99 |               1 |          2.07 |
|         9.5 |         99 |               1 |          1.87 |
|         6.5 |        100 |               1 |           1.6 |
|         6.5 |        100 |               1 |           1.6 |
|          11 |      81.54 |               1 |          2.79 |
mysql>    7.5 |      63.33 |               1 |          1.96 |
mysql> -------+------------+-----------------+---------------+
```

Case statement that adds a column that contains ride rating based off tip amount. This rating is either NO TIP, BRONSE, SILVER or GOLD depending on how high the tip is per ride.



```
SELECT tip_amount,
    CASE
        WHEN tip_amount = 0 THEN 'NO TIP'
        WHEN tip_amount BETWEEN 0.01 AND 9.99 THEN 'BRONSE'
        WHEN tip_amount BETWEEN 10 AND 99.99 THEN 'SILVER'
        WHEN tip_amount <= 100 THEN 'GOLD'
    END AS Customer_tip_rating
FROM trip_data;
```



```
mysql>
|          0 | NO TIP              |
|        5.2 | BRONSE              |
|       2.75 | BRONSE              |
|          0 | NO TIP              |
|          0 | NO TIP              |
|       2.75 | BRONSE              |
|          0 | NO TIP              |
|          0 | NO TIP              |
|       2.75 | BRONSE              |
|          0 | NO TIP              |
|          0 | NO TIP              |
|       2.75 | BRONSE              |
|          0 | NO TIP              |
|          0 | NO TIP              |
|          0 | NO TIP              |
|          0 | NO TIP              |
|          0 | NO TIP              |
|          0 | NO TIP              |
|          0 | NO TIP              |
|          0 | NO TIP              |
|          0 | NO TIP              |
|          0 | NO TIP              |
|          0 | NO TIP              |
|          0 | NO TIP              |
|          0 | NO TIP              |
|          0 | NO TIP              |
```

Analysis

My first observation when eyeballing the data is the congestion surcharge doesn't affect the total amount. It seems like something that would be deducted from the total amount or the taxi company should be charging their customers.

The average ride cost a customer is $11.66. The largest tip was $117.28, it could be worth speaking to this driver to see how they attain such a large tip. If it was down to great customer service, we could look at understanding the steps this driver took to make improvements company-wide.

The longest trip of 126,501.77 miles is extremely long. To put this into perspective, a google search on average car mileage reveals that the average yearly mileage is 7,134 miles. This discrepancy must be a mistake and is worth investigating.

After calculating the mean of the average tips per passenger, we see this is roughly equal. Going forward, it would be worth understanding when multi-passenger rides take place. My hypothesis is there are more multi-passenger rides in the evening and close to weekends.

We can see that single passenger rides heavily out ways multi-passenger rides. This could be a marketing opportunity to encourage passengers to carpool more frequently making for a greener service. However, we also see consistent tipping patterns when passengers ride alone.

Kips Bay and East Harlem South Bay are the most lucrative pickup points and dropoff points. These neighbourhoods are very close to each other. This could be a market research opportunity.

Kips Bay and East Harlem South are also the most popular pickup and drop off points. This is useful for scheduling drivers to make sure there is enough coverage in these locations.

The majority of payments are paid by card; therefore, it could be worth investing in IT infostructure such as contactless payment methods and other card payment options in order to ensure customers can pay quickly and efficiently. As the average tip amount is much higher when clients pay by card, I would advise incentives to ensure customers are more likely to use card payments over cash.