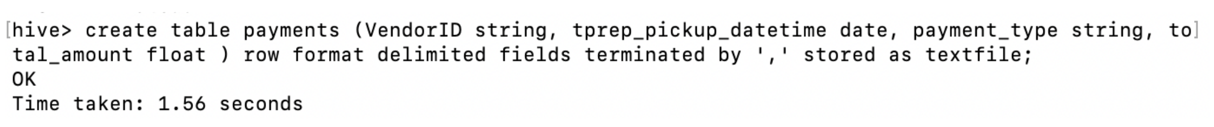
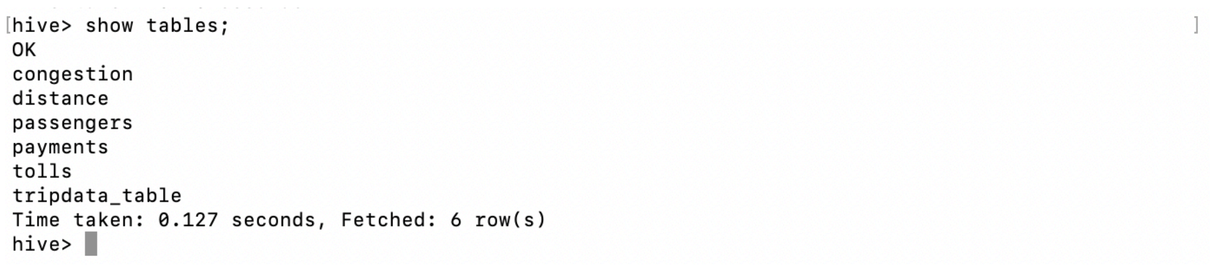
1.

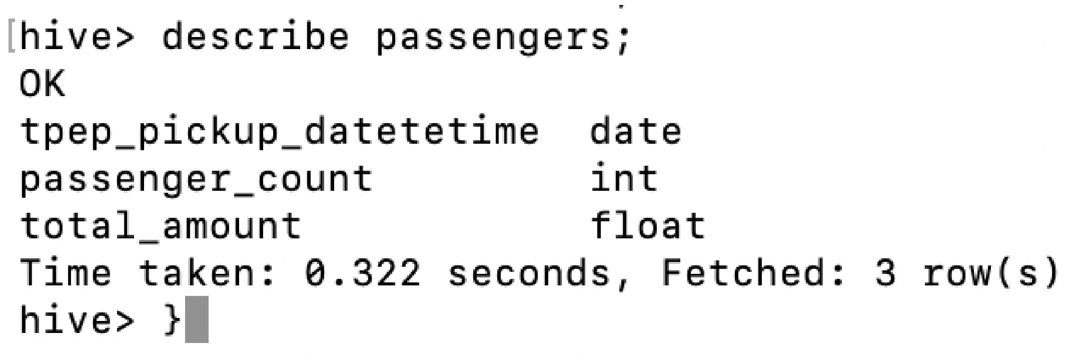


```
hive> create table payments (VendorID string, tprep_pickup_datetime date, payment_type string, to
tal_amount float ) row format delimited fields terminated by ',' stored as textfile;
OK
Time taken: 1.56 seconds
```

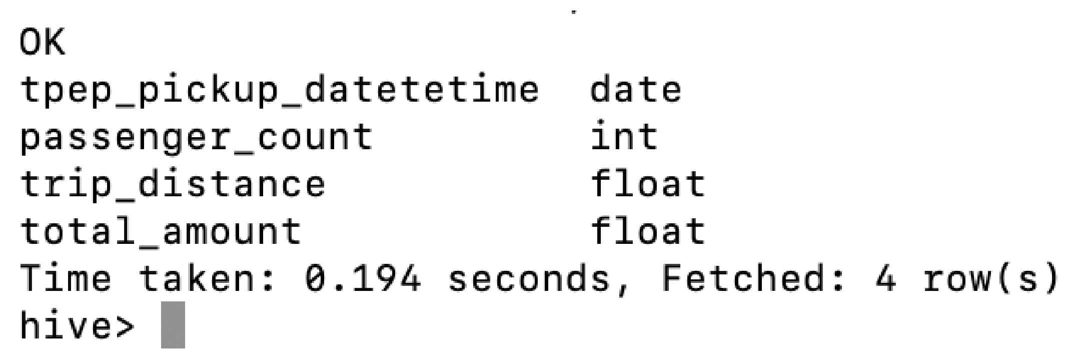


```
hive> show tables;
OK
congestion
distance
passengers
payments
tolls
tripdata_table
Time taken: 0.127 seconds, Fetched: 6 row(s)
hive>
```

2.

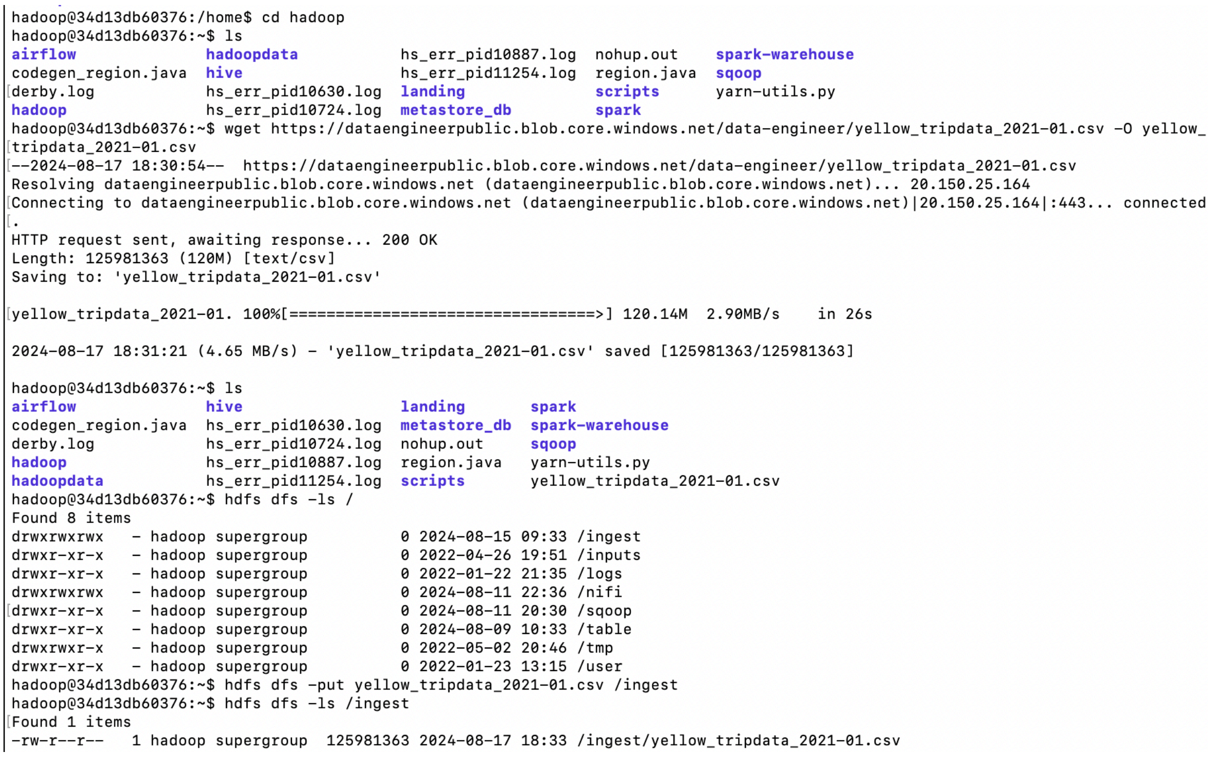


```
hive> describe passengers;
OK
tpep_pickup_datetetime  date
passenger_count         int
total_amount            float
Time taken: 0.322 seconds, Fetched: 3 row(s)
hive> }
```

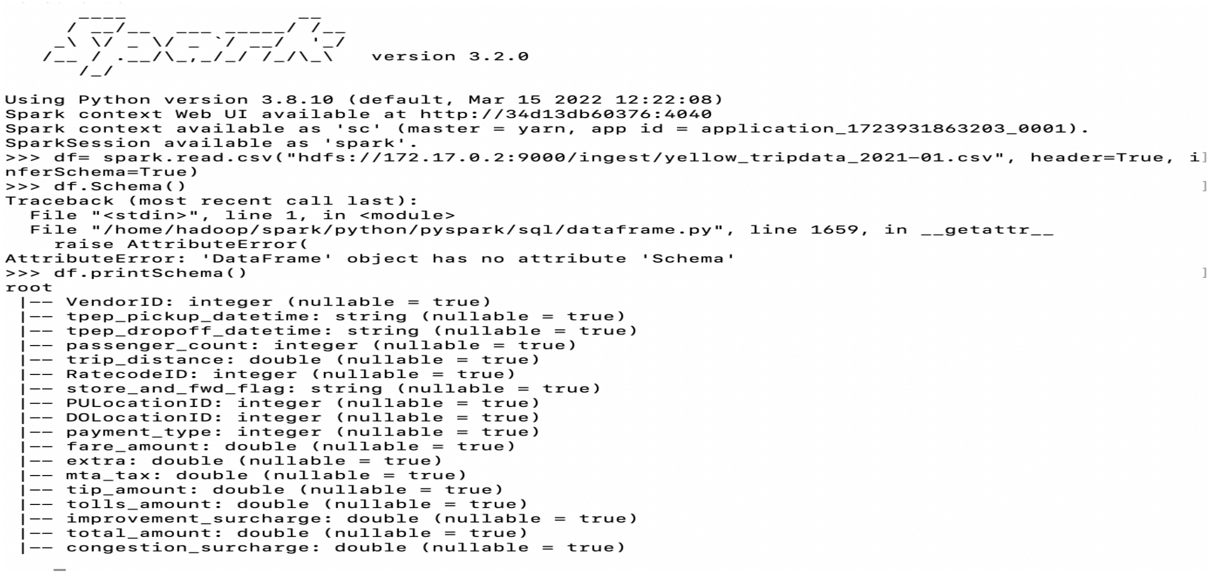


```
OK
tpep_pickup_datetetime  date
passenger_count         int
trip_distance           float
total_amount            float
Time taken: 0.194 seconds, Fetched: 4 row(s)
hive>
```

3.

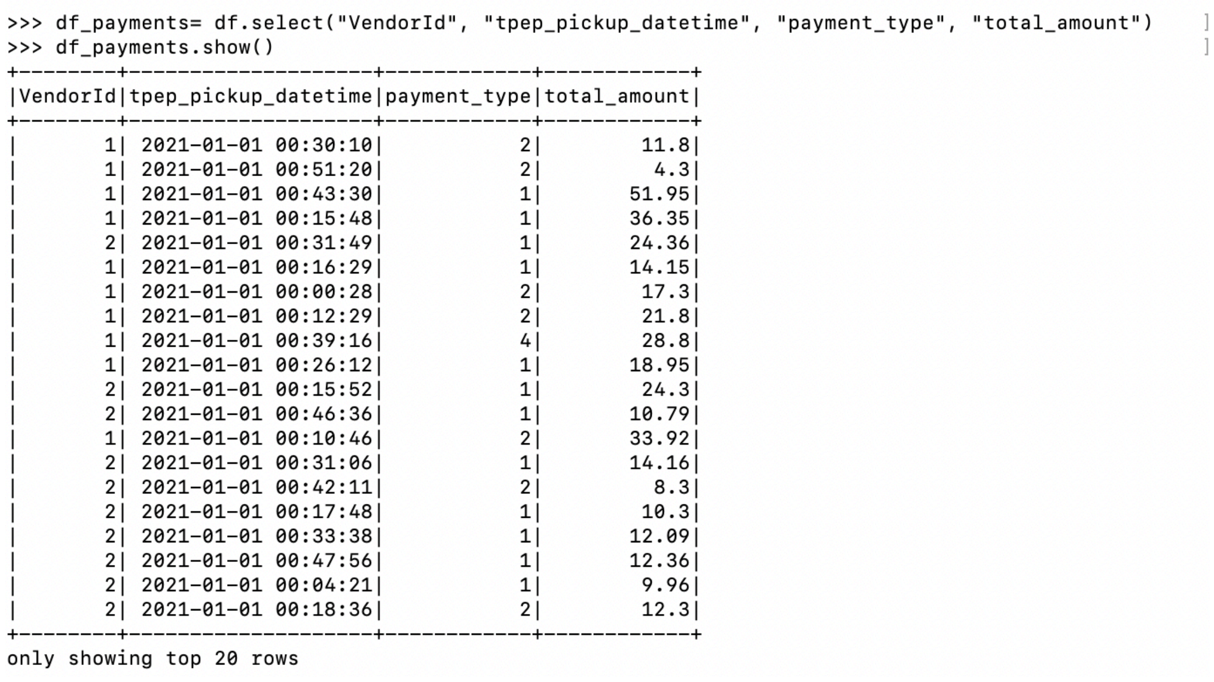


```
hadoop@34d13db60376:/home$ cd hadoop
hadoop@34d13db60376:~$ ls
airflow             hadoopdata           hs_err_pid10887.log  nohup.out    spark-warehouse
codegen_region.java hive                 hs_err_pid11254.log  region.java  sqoop
derby.log           hs_err_pid10630.log  landing              scripts      yarn-utils.py
hadoop              hs_err_pid10724.log  metastore_db         spark
hadoop@34d13db60376:~$ wget https://dataengineerpublic.blob.core.windows.net/data-engineer/yellow_tripdata_2021-01.csv -O yellow_
tripdata_2021-01.csv
--2024-08-17 18:30:54--  https://dataengineerpublic.blob.core.windows.net/data-engineer/yellow_tripdata_2021-01.csv
Resolving dataengineerpublic.blob.core.windows.net (dataengineerpublic.blob.core.windows.net)... 20.150.25.164
Connecting to dataengineerpublic.blob.core.windows.net (dataengineerpublic.blob.core.windows.net)|20.150.25.164|:443... connected
.
HTTP request sent, awaiting response... 200 OK
Length: 125981363 (120M) [text/csv]
Saving to: 'yellow_tripdata_2021-01.csv'

yellow_tripdata_2021-01. 100%[================================>] 120.14M  2.90MB/s    in 26s

2024-08-17 18:31:21 (4.65 MB/s) - 'yellow_tripdata_2021-01.csv' saved [125981363/125981363]

hadoop@34d13db60376:~$ ls
airflow             hive                 landing       spark
codegen_region.java hs_err_pid10630.log  metastore_db  spark-warehouse
derby.log           hs_err_pid10724.log  nohup.out     sqoop
hadoop              hs_err_pid10887.log  region.java   yarn-utils.py
hadoopdata          hs_err_pid11254.log  scripts       yellow_tripdata_2021-01.csv
hadoop@34d13db60376:~$ hdfs dfs -ls /
Found 8 items
drwxrwxrwx   - hadoop supergroup          0 2024-08-15 09:33 /ingest
drwxr-xr-x   - hadoop supergroup          0 2022-04-26 19:51 /inputs
drwxr-xr-x   - hadoop supergroup          0 2022-01-22 21:35 /logs
drwxrwxrwx   - hadoop supergroup          0 2024-08-11 22:36 /nifi
drwxr-xr-x   - hadoop supergroup          0 2024-08-11 20:30 /sqoop
drwxr-xr-x   - hadoop supergroup          0 2024-08-09 10:33 /table
drwxrwxr-x   - hadoop supergroup          0 2022-05-02 20:46 /tmp
drwxr-xr-x   - hadoop supergroup          0 2022-01-23 13:15 /user
hadoop@34d13db60376:~$ hdfs dfs -put yellow_tripdata_2021-01.csv /ingest
hadoop@34d13db60376:~$ hdfs dfs -ls /ingest
Found 1 items
-rw-r--r--   1 hadoop supergroup  125981363 2024-08-17 18:33 /ingest/yellow_tripdata_2021-01.csv
```

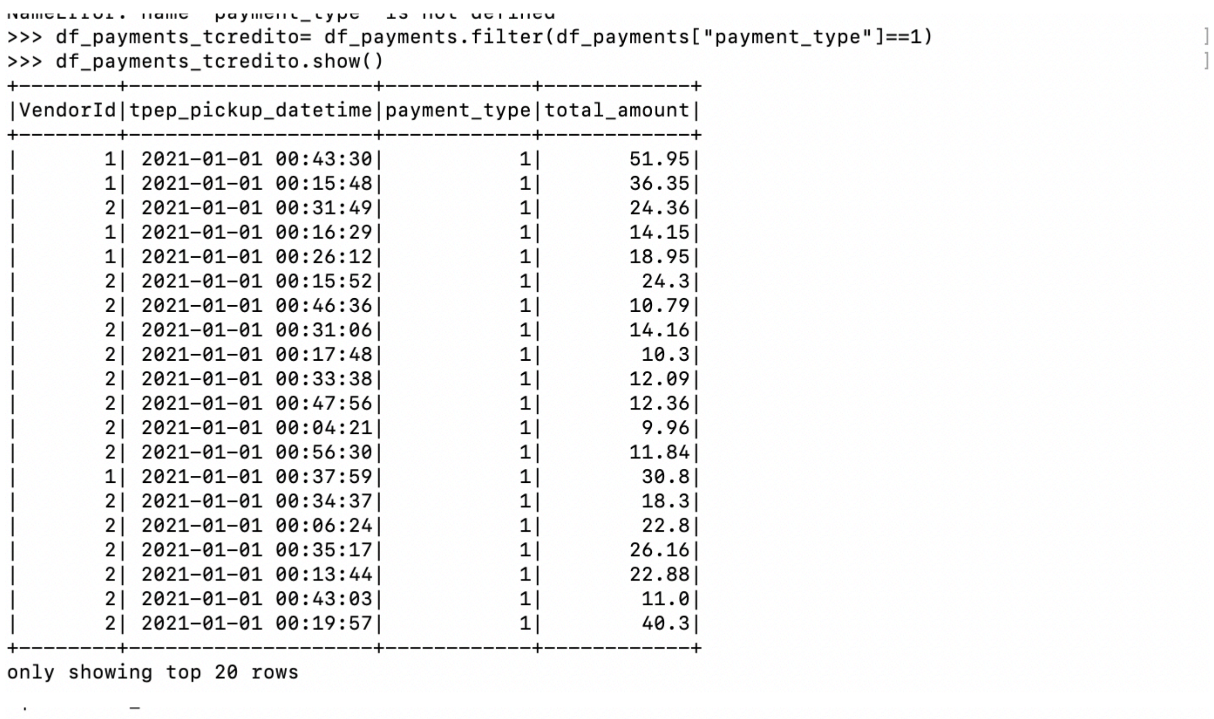


```
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.2.0
      /_/

Using Python version 3.8.10 (default, Mar 15 2022 12:22:08)
Spark context Web UI available at http://34d13db60376:4040
Spark context available as 'sc' (master = yarn, app id = application_1723931863203_0001).
SparkSession available as 'spark'.
>>> df= spark.read.csv("hdfs://172.17.0.2:9000/ingest/yellow_tripdata_2021-01.csv", header=True, i
nferSchema=True)
>>> df.Schema()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "/home/hadoop/spark/python/pyspark/sql/dataframe.py", line 1659, in __getattr__
    raise AttributeError(
AttributeError: 'DataFrame' object has no attribute 'Schema'
>>> df.printSchema()
root
 |-- VendorID: integer (nullable = true)
 |-- tpep_pickup_datetime: string (nullable = true)
 |-- tpep_dropoff_datetime: string (nullable = true)
 |-- passenger_count: integer (nullable = true)
 |-- trip_distance: double (nullable = true)
 |-- RatecodeID: integer (nullable = true)
 |-- store_and_fwd_flag: string (nullable = true)
 |-- PULocationID: integer (nullable = true)
 |-- DOLocationID: integer (nullable = true)
 |-- payment_type: integer (nullable = true)
 |-- fare_amount: double (nullable = true)
 |-- extra: double (nullable = true)
 |-- mta_tax: double (nullable = true)
 |-- tip_amount: double (nullable = true)
 |-- tolls_amount: double (nullable = true)
 |-- improvement_surcharge: double (nullable = true)
 |-- total_amount: double (nullable = true)
 |-- congestion_surcharge: double (nullable = true)
```

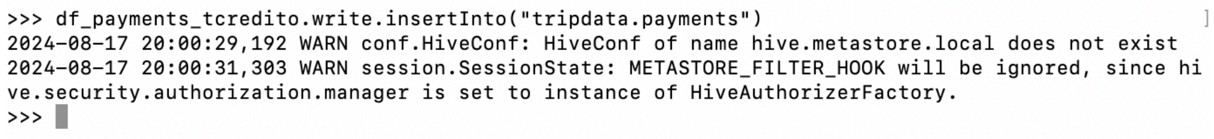
5.



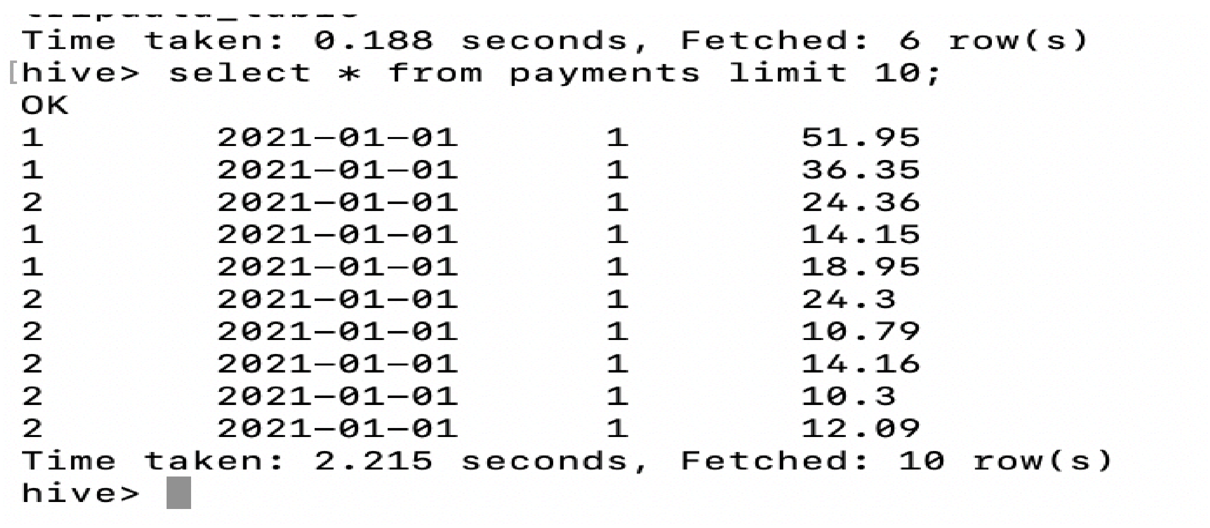
```
>>> df_payments= df.select("VendorId", "tpep_pickup_datetime", "payment_type", "total_amount")
>>> df_payments.show()
+--------+--------------------+------------+------------+
|VendorId|tpep_pickup_datetime|payment_type|total_amount|
+--------+--------------------+------------+------------+
|       1| 2021-01-01 00:30:10|           2|        11.8|
|       1| 2021-01-01 00:51:20|           2|         4.3|
|       1| 2021-01-01 00:43:30|           1|       51.95|
|       1| 2021-01-01 00:15:48|           1|       36.35|
|       2| 2021-01-01 00:31:49|           1|       24.36|
|       1| 2021-01-01 00:16:29|           1|       14.15|
|       1| 2021-01-01 00:00:28|           2|        17.3|
|       1| 2021-01-01 00:12:29|           2|        21.8|
|       1| 2021-01-01 00:39:16|           4|        28.8|
|       1| 2021-01-01 00:26:12|           1|       18.95|
|       2| 2021-01-01 00:15:52|           1|        24.3|
|       2| 2021-01-01 00:46:36|           1|       10.79|
|       1| 2021-01-01 00:10:46|           2|       33.92|
|       2| 2021-01-01 00:31:06|           1|       14.16|
|       2| 2021-01-01 00:42:11|           2|         8.3|
|       2| 2021-01-01 00:17:48|           1|        10.3|
|       2| 2021-01-01 00:33:38|           1|       12.09|
|       2| 2021-01-01 00:47:56|           1|       12.36|
|       2| 2021-01-01 00:04:21|           1|        9.96|
|       2| 2021-01-01 00:18:36|           2|        12.3|
+--------+--------------------+------------+------------+
only showing top 20 rows
```



```
NameError: name 'payment_type' is not defined
>>> df_payments_tcredito= df_payments.filter(df_payments["payment_type"]==1)
>>> df_payments_tcredito.show()
+--------+--------------------+------------+------------+
|VendorId|tpep_pickup_datetime|payment_type|total_amount|
+--------+--------------------+------------+------------+
|       1| 2021-01-01 00:43:30|           1|       51.95|
|       1| 2021-01-01 00:15:48|           1|       36.35|
|       2| 2021-01-01 00:31:49|           1|       24.36|
|       1| 2021-01-01 00:16:29|           1|       14.15|
|       1| 2021-01-01 00:26:12|           1|       18.95|
|       2| 2021-01-01 00:15:52|           1|        24.3|
|       2| 2021-01-01 00:46:36|           1|       10.79|
|       2| 2021-01-01 00:31:06|           1|       14.16|
|       2| 2021-01-01 00:17:48|           1|        10.3|
|       2| 2021-01-01 00:33:38|           1|       12.09|
|       2| 2021-01-01 00:47:56|           1|       12.36|
|       2| 2021-01-01 00:04:21|           1|        9.96|
|       2| 2021-01-01 00:56:30|           1|       11.84|
|       1| 2021-01-01 00:37:59|           1|        30.8|
|       2| 2021-01-01 00:34:37|           1|        18.3|
|       2| 2021-01-01 00:06:24|           1|        22.8|
|       2| 2021-01-01 00:35:17|           1|       26.16|
|       2| 2021-01-01 00:13:44|           1|       22.88|
|       2| 2021-01-01 00:43:03|           1|        11.0|
|       2| 2021-01-01 00:19:57|           1|        40.3|
+--------+--------------------+------------+------------+
only showing top 20 rows
```

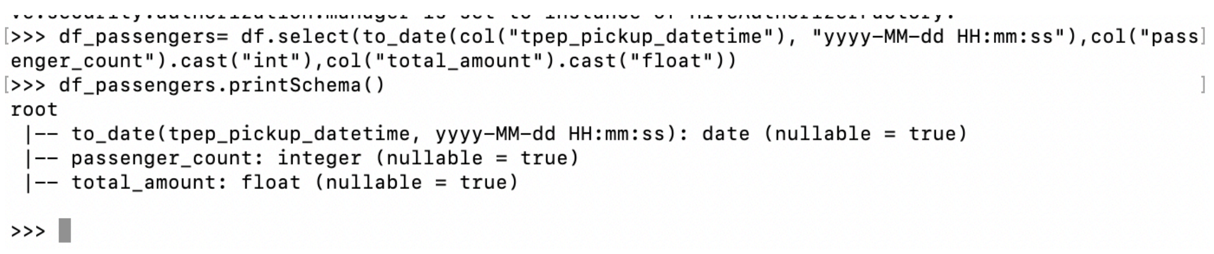


```
>>> df_payments_tcredito.write.insertInto("tripdata.payments")
2024-08-17 20:00:29,192 WARN conf.HiveConf: HiveConf of name hive.metastore.local does not exist
2024-08-17 20:00:31,303 WARN session.SessionState: METASTORE_FILTER_HOOK will be ignored, since hi
ve.security.authorization.manager is set to instance of HiveAuthorizerFactory.
>>> 
```

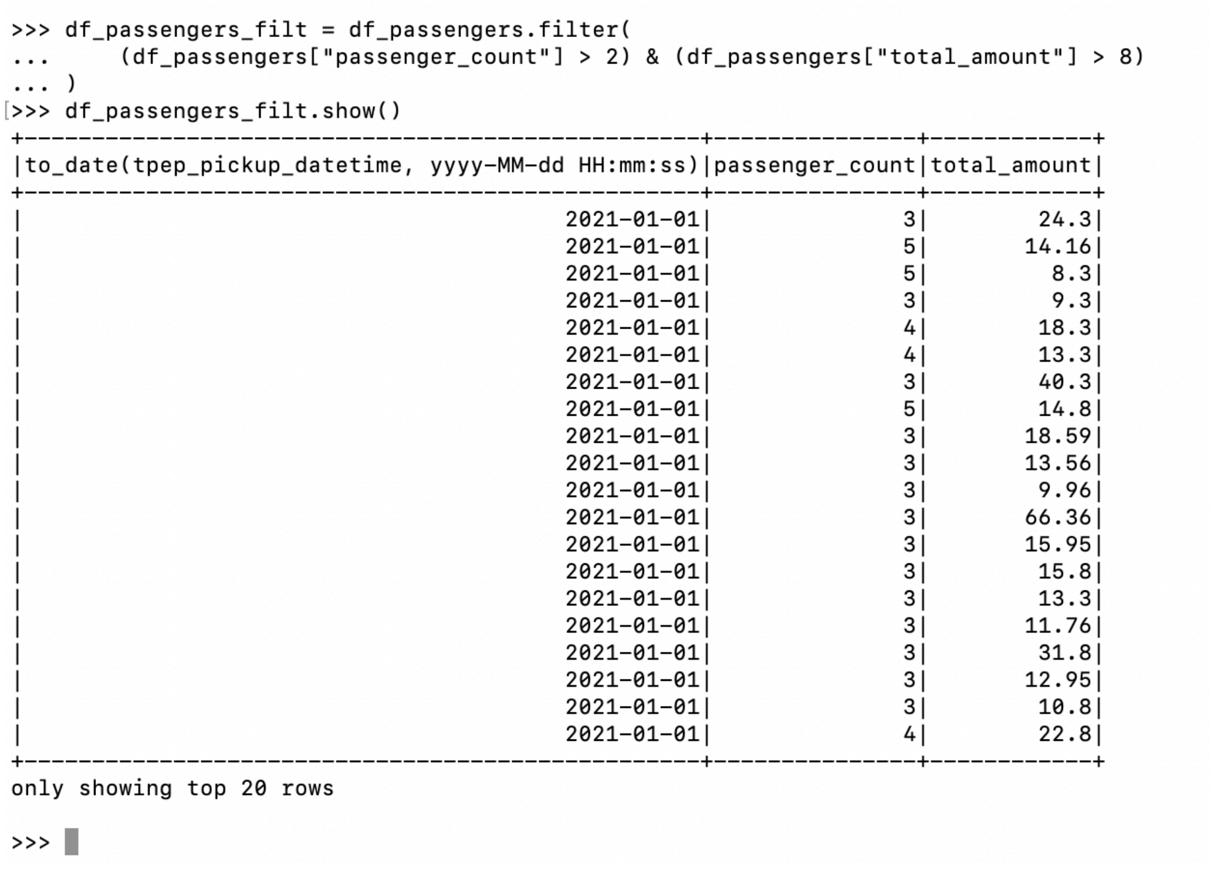


```
Time taken: 0.188 seconds, Fetched: 6 row(s)
[hive> select * from payments limit 10;
OK
1       2021-01-01      1       51.95
1       2021-01-01      1       36.35
2       2021-01-01      1       24.36
1       2021-01-01      1       14.15
1       2021-01-01      1       18.95
2       2021-01-01      1       24.3
2       2021-01-01      1       10.79
2       2021-01-01      1       14.16
2       2021-01-01      1       10.3
2       2021-01-01      1       12.09
Time taken: 2.215 seconds, Fetched: 10 row(s)
hive> 
```

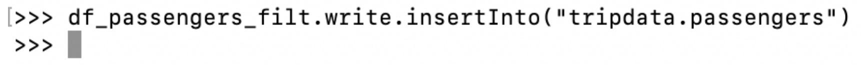
6.



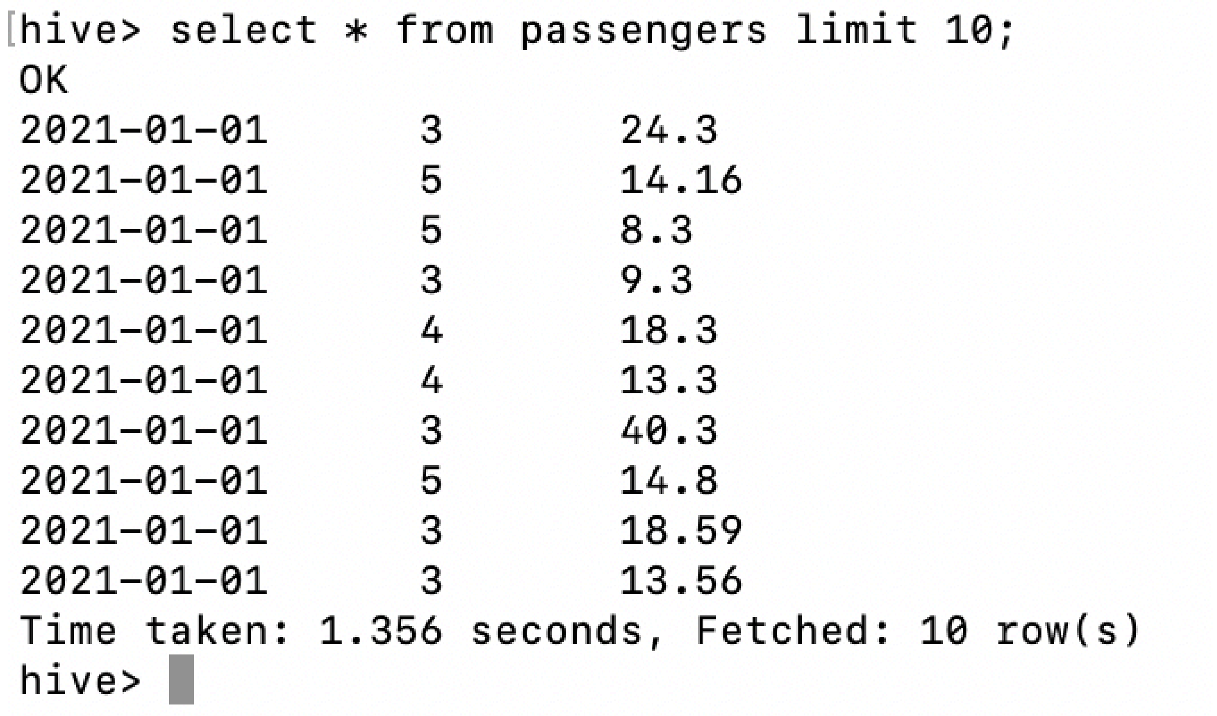
```
[>>> df_passengers= df.select(to_date(col("tpep_pickup_datetime"), "yyyy-MM-dd HH:mm:ss"),col("pass]
enger_count").cast("int"),col("total_amount").cast("float"))
[>>> df_passengers.printSchema()                                                                   ]
root
 |-- to_date(tpep_pickup_datetime, yyyy-MM-dd HH:mm:ss): date (nullable = true)
 |-- passenger_count: integer (nullable = true)
 |-- total_amount: float (nullable = true)

>>> 
```



```
>>> df_passengers_filt = df_passengers.filter(
...     (df_passengers["passenger_count"] > 2) & (df_passengers["total_amount"] > 8)
... )
[>>> df_passengers_filt.show()
+---------------------------------------------------+---------------+------------+
|to_date(tpep_pickup_datetime, yyyy-MM-dd HH:mm:ss)|passenger_count|total_amount|
+---------------------------------------------------+---------------+------------+
|                                         2021-01-01|              3|        24.3|
|                                         2021-01-01|              5|       14.16|
|                                         2021-01-01|              5|         8.3|
|                                         2021-01-01|              3|         9.3|
|                                         2021-01-01|              4|        18.3|
|                                         2021-01-01|              4|        13.3|
|                                         2021-01-01|              3|        40.3|
|                                         2021-01-01|              5|        14.8|
|                                         2021-01-01|              3|       18.59|
|                                         2021-01-01|              3|       13.56|
|                                         2021-01-01|              3|        9.96|
|                                         2021-01-01|              3|       66.36|
|                                         2021-01-01|              3|       15.95|
|                                         2021-01-01|              3|        15.8|
|                                         2021-01-01|              3|        13.3|
|                                         2021-01-01|              3|       11.76|
|                                         2021-01-01|              3|        31.8|
|                                         2021-01-01|              3|       12.95|
|                                         2021-01-01|              3|        10.8|
|                                         2021-01-01|              4|        22.8|
+---------------------------------------------------+---------------+------------+
only showing top 20 rows

>>> 
```

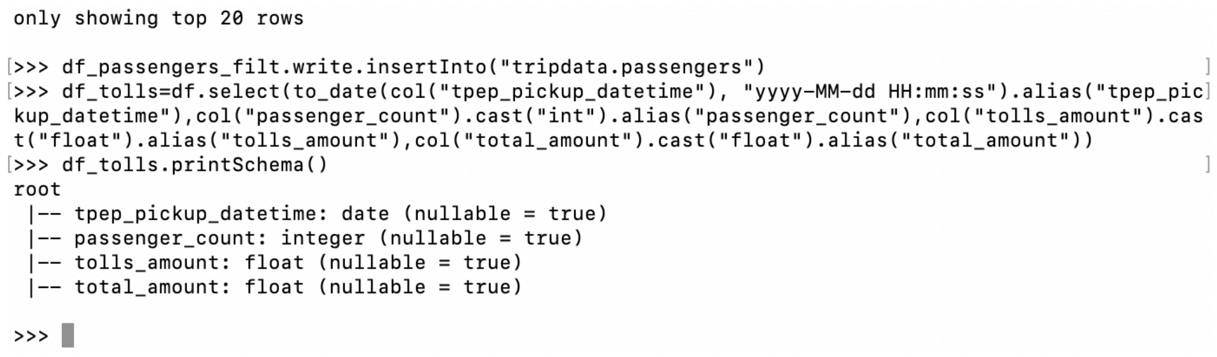


```
[>>> df_passengers_filt.write.insertInto("tripdata.passengers")
 >>> 
```

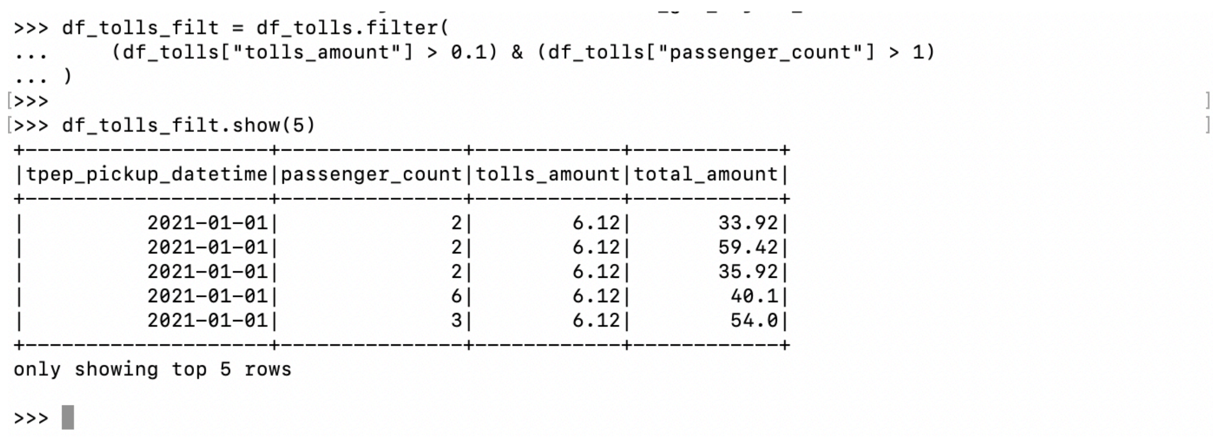


```
[hive> select * from passengers limit 10;
 OK
 2021-01-01      3       24.3
 2021-01-01      5       14.16
 2021-01-01      5       8.3
 2021-01-01      3       9.3
 2021-01-01      4       18.3
 2021-01-01      4       13.3
 2021-01-01      3       40.3
 2021-01-01      5       14.8
 2021-01-01      3       18.59
 2021-01-01      3       13.56
 Time taken: 1.356 seconds, Fetched: 10 row(s)
 hive> 
```

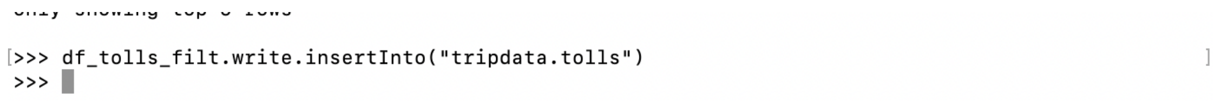
7.



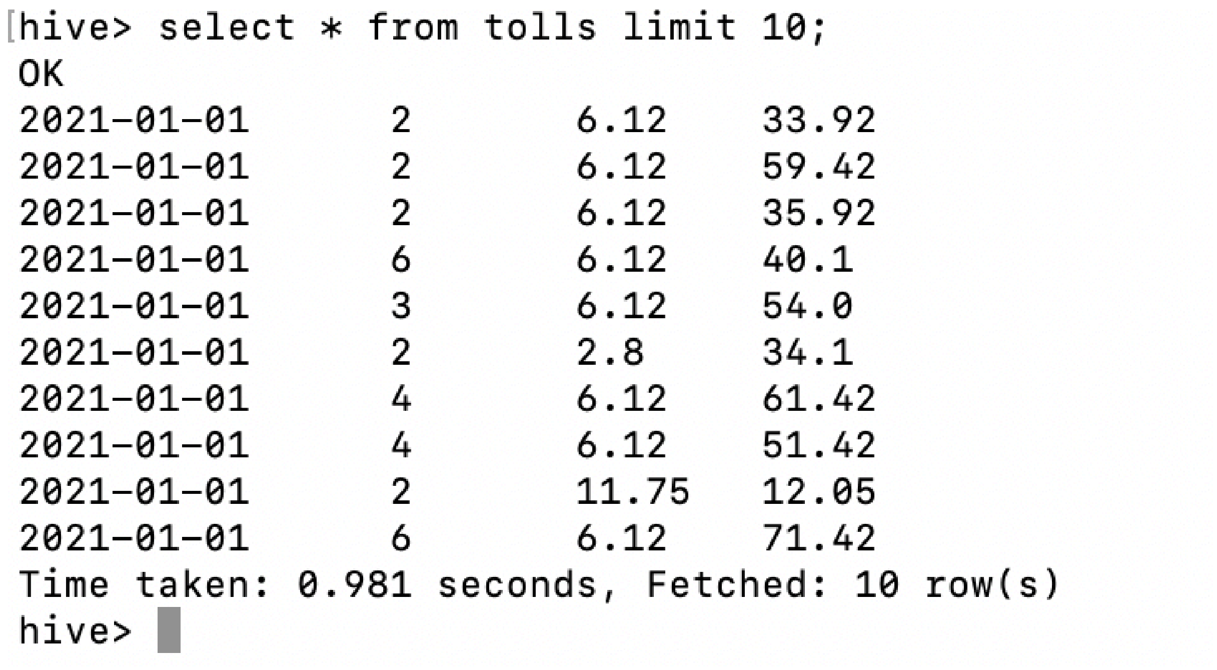
```
 only showing top 20 rows

[>>> df_passengers_filt.write.insertInto("tripdata.passengers")                                    ]
[>>> df_tolls=df.select(to_date(col("tpep_pickup_datetime"), "yyyy-MM-dd HH:mm:ss").alias("tpep_pic]
 kup_datetime"),col("passenger_count").cast("int").alias("passenger_count"),col("tolls_amount").cas
 t("float").alias("tolls_amount"),col("total_amount").cast("float").alias("total_amount"))
[>>> df_tolls.printSchema()                                                                         ]
 root
  |-- tpep_pickup_datetime: date (nullable = true)
  |-- passenger_count: integer (nullable = true)
  |-- tolls_amount: float (nullable = true)
  |-- total_amount: float (nullable = true)

 >>> 
```



```
 >>> df_tolls_filt = df_tolls.filter(
 ...     (df_tolls["tolls_amount"] > 0.1) & (df_tolls["passenger_count"] > 1)
 ... )
[>>>                                                                                                ]
[>>> df_tolls_filt.show(5)                                                                          ]
 +--------------------+---------------+------------+------------+
 |tpep_pickup_datetime|passenger_count|tolls_amount|total_amount|
 +--------------------+---------------+------------+------------+
 |          2021-01-01|              2|        6.12|       33.92|
 |          2021-01-01|              2|        6.12|       59.42|
 |          2021-01-01|              2|        6.12|       35.92|
 |          2021-01-01|              6|        6.12|        40.1|
 |          2021-01-01|              3|        6.12|        54.0|
 +--------------------+---------------+------------+------------+
 only showing top 5 rows

 >>> 
```

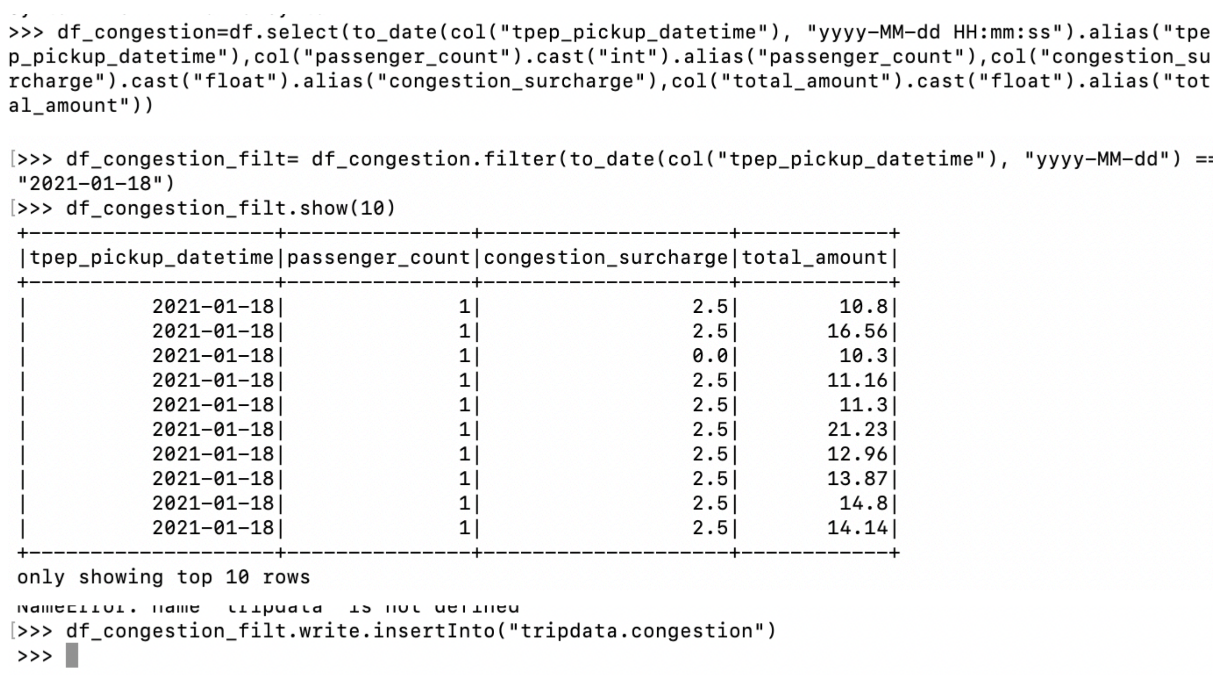


```
[>>> df_tolls_filt.write.insertInto("tripdata.tolls")                                              ]
 >>> 
```

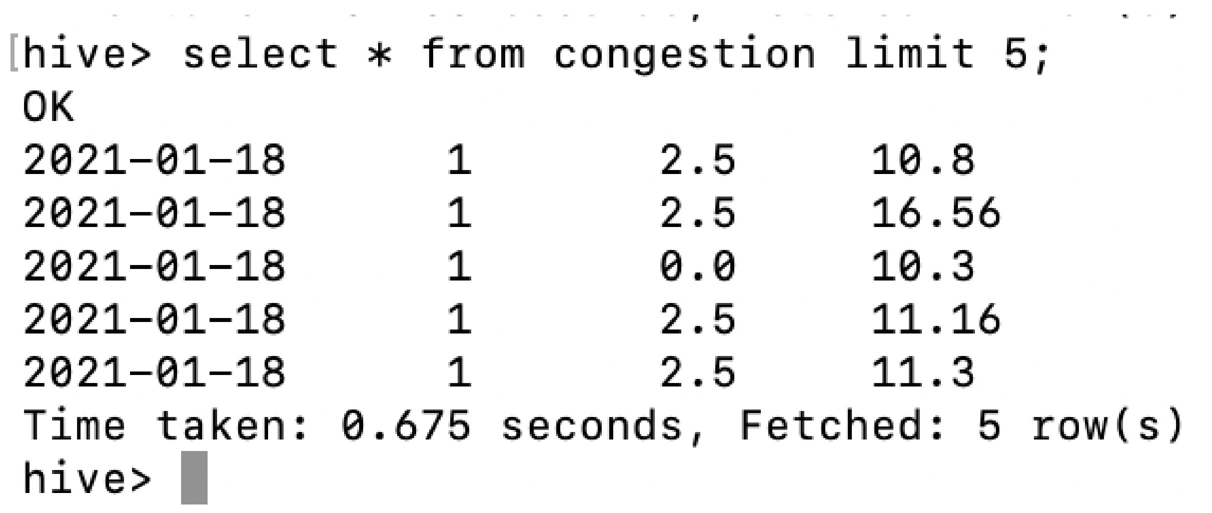


```
[hive> select * from tolls limit 10;
OK
2021-01-01      2       6.12    33.92
2021-01-01      2       6.12    59.42
2021-01-01      2       6.12    35.92
2021-01-01      6       6.12    40.1
2021-01-01      3       6.12    54.0
2021-01-01      2       2.8     34.1
2021-01-01      4       6.12    61.42
2021-01-01      4       6.12    51.42
2021-01-01      2       11.75   12.05
2021-01-01      6       6.12    71.42
Time taken: 0.981 seconds, Fetched: 10 row(s)
hive>
```

8.

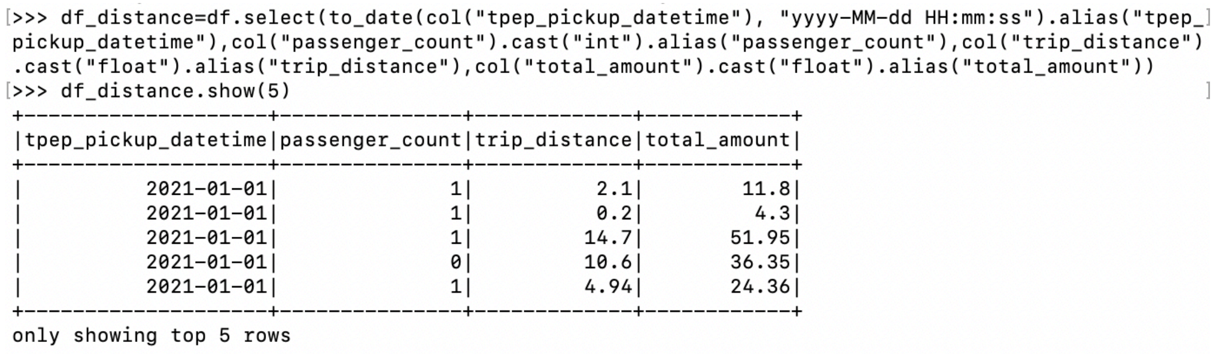


```
>>> df_congestion=df.select(to_date(col("tpep_pickup_datetime"), "yyyy-MM-dd HH:mm:ss").alias("tpe
p_pickup_datetime"),col("passenger_count").cast("int").alias("passenger_count"),col("congestion_su
rcharge").cast("float").alias("congestion_surcharge"),col("total_amount").cast("float").alias("tot
al_amount"))

[>>> df_congestion_filt= df_congestion.filter(to_date(col("tpep_pickup_datetime"), "yyyy-MM-dd") ==
 "2021-01-18")
[>>> df_congestion_filt.show(10)
+--------------------+---------------+--------------------+------------+
|tpep_pickup_datetime|passenger_count|congestion_surcharge|total_amount|
+--------------------+---------------+--------------------+------------+
|          2021-01-18|              1|                 2.5|        10.8|
|          2021-01-18|              1|                 2.5|       16.56|
|          2021-01-18|              1|                 0.0|        10.3|
|          2021-01-18|              1|                 2.5|       11.16|
|          2021-01-18|              1|                 2.5|        11.3|
|          2021-01-18|              1|                 2.5|       21.23|
|          2021-01-18|              1|                 2.5|       12.96|
|          2021-01-18|              1|                 2.5|       13.87|
|          2021-01-18|              1|                 2.5|        14.8|
|          2021-01-18|              1|                 2.5|       14.14|
+--------------------+---------------+--------------------+------------+
only showing top 10 rows

NameError: name 'tripdata' is not defined
[>>> df_congestion_filt.write.insertInto("tripdata.congestion")
>>>
```

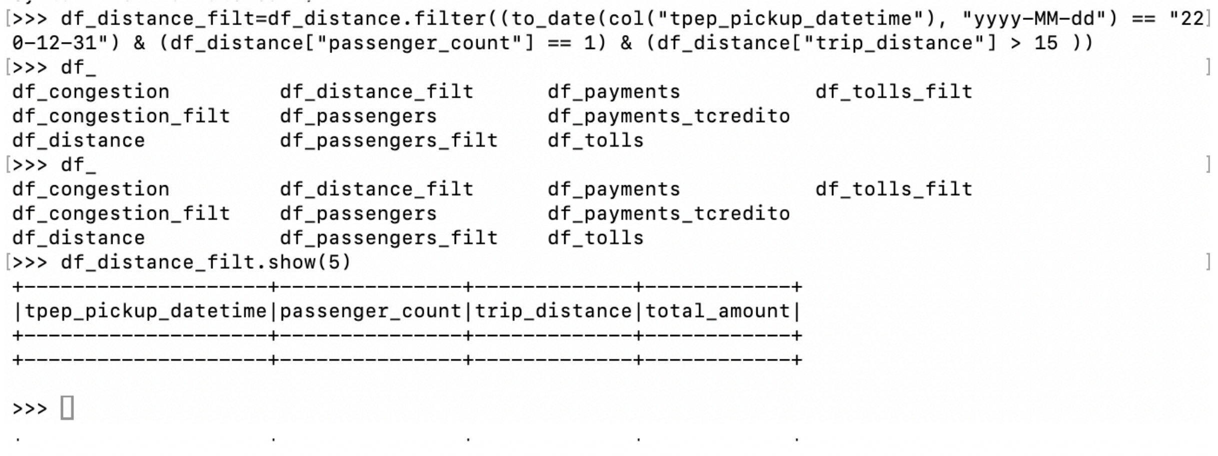


```
hive> select * from congestion limit 5;
OK
2021-01-18      1       2.5     10.8
2021-01-18      1       2.5     16.56
2021-01-18      1       0.0     10.3
2021-01-18      1       2.5     11.16
2021-01-18      1       2.5     11.3
Time taken: 0.675 seconds, Fetched: 5 row(s)
hive> 
```

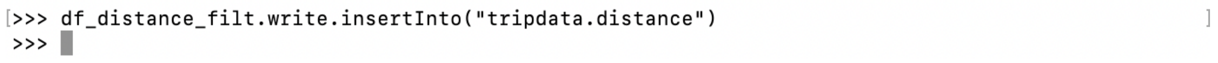
9.



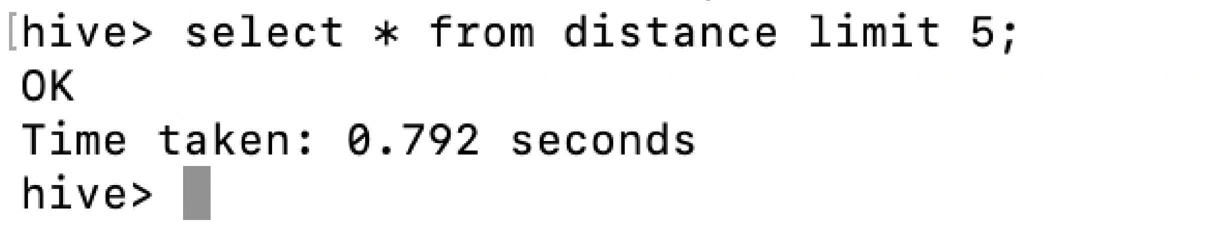
```
>>> df_distance=df.select(to_date(col("tpep_pickup_datetime"), "yyyy-MM-dd HH:mm:ss").alias("tpep_
pickup_datetime"),col("passenger_count").cast("int").alias("passenger_count"),col("trip_distance")
.cast("float").alias("trip_distance"),col("total_amount").cast("float").alias("total_amount"))
>>> df_distance.show(5)
+--------------------+---------------+-------------+------------+
|tpep_pickup_datetime|passenger_count|trip_distance|total_amount|
+--------------------+---------------+-------------+------------+
|          2021-01-01|              1|          2.1|        11.8|
|          2021-01-01|              1|          0.2|         4.3|
|          2021-01-01|              1|         14.7|       51.95|
|          2021-01-01|              0|         10.6|       36.35|
|          2021-01-01|              1|         4.94|       24.36|
+--------------------+---------------+-------------+------------+
only showing top 5 rows
```



```
>>> df_distance_filt=df_distance.filter((to_date(col("tpep_pickup_datetime"), "yyyy-MM-dd") == "22
0-12-31") & (df_distance["passenger_count"] == 1) & (df_distance["trip_distance"] > 15 ))
>>> df_
df_congestion           df_distance_filt        df_payments             df_tolls_filt
df_congestion_filt      df_passengers           df_payments_tcredito
df_distance             df_passengers_filt      df_tolls
>>> df_
df_congestion           df_distance_filt        df_payments             df_tolls_filt
df_congestion_filt      df_passengers           df_payments_tcredito
df_distance             df_passengers_filt      df_tolls
>>> df_distance_filt.show(5)
+--------------------+---------------+-------------+------------+
|tpep_pickup_datetime|passenger_count|trip_distance|total_amount|
+--------------------+---------------+-------------+------------+
+--------------------+---------------+-------------+------------+

>>> 
```



```
>>> df_distance_filt.write.insertInto("tripdata.distance")
>>> 
```



```
hive> select * from distance limit 5;
OK
Time taken: 0.792 seconds
hive> 
```