## TASK 3

## Bulk import data from next two files in the dataset on your EMR cluster to your HBase Table using the relevant codes.

**Step1:**

LOAD DATA LOCAL INFILE '/home/hadoop/yellow_tripdata_2017-03.csv' INTO TABLE trip_log

FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' IGNORE 1 LINES;

**Step2:**

LOAD DATA LOCAL INFILE '/home/hadoop/yellow_tripdata_2017-04.csv' INTO TABLE trip_log

FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' IGNORE 1 LINES;

**Step3:**

sqoop import --connect jdbc:mysql://demodb.creoyuw4qfr6.us-east-1.rds.amazonaws.com/taxi_records --username admin --password admin123 --table trip_log --hbase-table trip_log_hbase --column-family cf1 --hbase-create-table --hbase-row-key tpep_pickup_datetime,tpep_dropoff_datetime --hbase-bulkload --split-by payment_type

**Step 4:**

Execute below python script in root

vi task3.py

----------Python script----------

```
import happybase

# create connection
connection = happybase.Connection('localhost', port=9090, autoconnect=False)

# open connection to perform operations
def open_connection():
    connection.open()
```

```
# close opened connection
def close_connection():
    connection.close()


# get the pointer to a table
def get_table(name):
    open_connection()
    table = connection.table(name)
    close_connection()
    return table


def batch_insert_data(filename, tablename):
    print("starting batch insert of "+filename)
    file = open(filename, 'r')
    table = get_table(tablename)
    open_connection()
    i = 0
    with table.batch(batch_size=50000) as b:
        for line in file:
            if i!=0:
                temp = line.strip().split(",")
                b.put(temp[1]+temp[2] ,{'cf1:VendorID': str(temp[0]), 'cf1:tpep_pickup_datetime':
str(temp[1]), 'cf1:tpep_dropoff_datetime': str(temp[2]), 'cf1:passenger_count': str(temp[3]),
'cf1:trip_distance': str(temp[4]), 'cf1:RatecodeID': str(temp[5]), 'cf1:store_and_fwd_flag':
str(temp[6]), 'cf1:PULocationID': str(temp[7]), 'cf1:DOLocationID': str(temp[8]),
'cf1:payment_type': str(temp[9]),'cf1:fare_amount': str(temp[10]), 'cf1:extra': str(temp[11]),
'cf1:mta_tax': str(temp[12]), 'cf1:tip_amount': str(temp[13]), 'cf1:tolls_amount':
str(temp[14]), 'cf1:improvement_surcharge': str(temp[15]), 'cf1:total_amount':
str(temp[16]), 'cf1:congestion_surcharge': str(temp[17]), 'cf1:airport_fee': str(temp[18]) })


            i+=1
```

```
    file.close()
    print("batch insert done")
    close_connection()


batch_insert_data('yellow_tripdata_2017-03.csv', 'trip_log')
batch_insert_data('yellow_tripdata_2017-04.csv', 'trip_log')
```

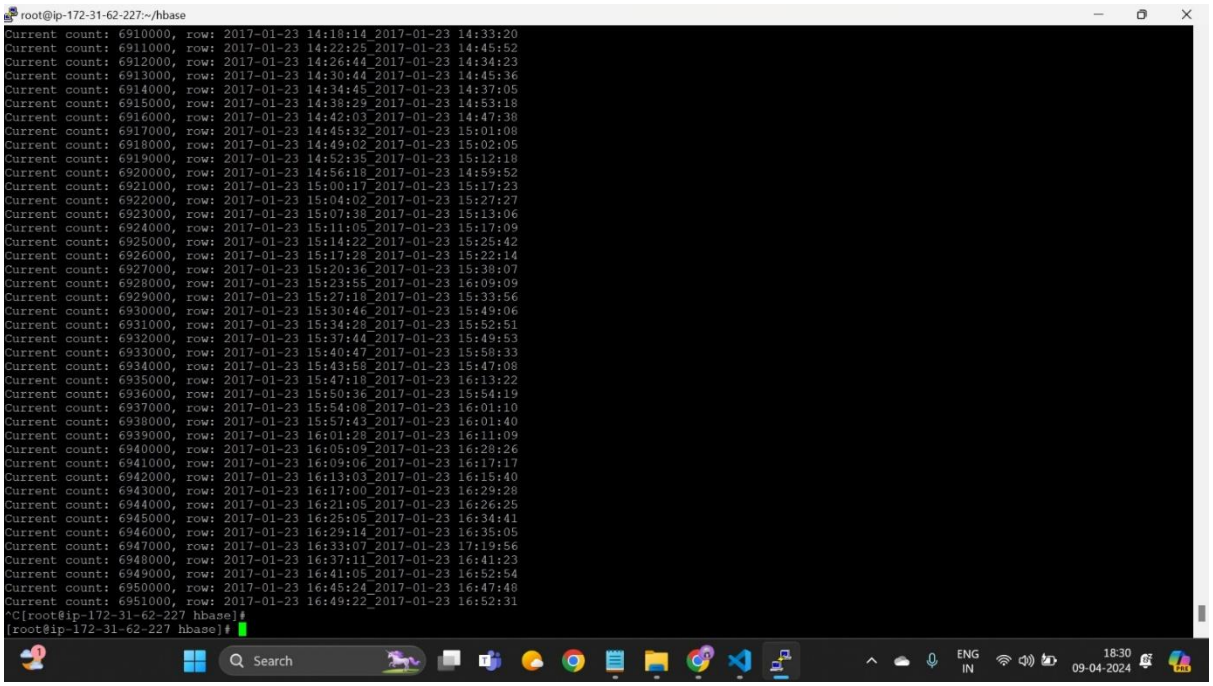
----------Python script----------

```
python task3.py
```

**Step5:**

Execute below commands

```
sudo -i
mkdir hbase
cd hbase
hbase shell
list
describe 'trip_log_hbase'
count 'trip_log_hbase'
```


root@ip-172-31-62-227:~/hbase
Current count: 6910000, row: 2017-01-23 14:18:14_2017-01-23 14:33:20
Current count: 6911000, row: 2017-01-23 14:22:25_2017-01-23 14:45:52
Current count: 6912000, row: 2017-01-23 14:26:44_2017-01-23 14:34:23
Current count: 6913000, row: 2017-01-23 14:30:44_2017-01-23 14:45:36
Current count: 6914000, row: 2017-01-23 14:34:45_2017-01-23 14:37:05
Current count: 6915000, row: 2017-01-23 14:38:29_2017-01-23 14:53:18
Current count: 6916000, row: 2017-01-23 14:42:03_2017-01-23 14:47:38
Current count: 6917000, row: 2017-01-23 14:45:32_2017-01-23 15:01:08
Current count: 6918000, row: 2017-01-23 14:49:02_2017-01-23 15:02:05
Current count: 6919000, row: 2017-01-23 14:52:35_2017-01-23 15:12:18
Current count: 6920000, row: 2017-01-23 14:56:18_2017-01-23 14:59:52
Current count: 6921000, row: 2017-01-23 15:00:17_2017-01-23 15:17:23
Current count: 6922000, row: 2017-01-23 15:04:02_2017-01-23 15:27:27
Current count: 6923000, row: 2017-01-23 15:07:38_2017-01-23 15:13:06
Current count: 6924000, row: 2017-01-23 15:11:05_2017-01-23 15:17:09
Current count: 6925000, row: 2017-01-23 15:14:22_2017-01-23 15:25:42
Current count: 6926000, row: 2017-01-23 15:17:28_2017-01-23 15:22:14
Current count: 6927000, row: 2017-01-23 15:20:36_2017-01-23 15:38:07
Current count: 6928000, row: 2017-01-23 15:23:55_2017-01-23 16:09:09
Current count: 6929000, row: 2017-01-23 15:27:18_2017-01-23 15:33:56
Current count: 6930000, row: 2017-01-23 15:30:46_2017-01-23 15:49:06
Current count: 6931000, row: 2017-01-23 15:34:28_2017-01-23 15:52:51
Current count: 6932000, row: 2017-01-23 15:37:44_2017-01-23 15:49:53
Current count: 6933000, row: 2017-01-23 15:40:47_2017-01-23 15:58:33
Current count: 6934000, row: 2017-01-23 15:43:58_2017-01-23 15:47:08
Current count: 6935000, row: 2017-01-23 15:47:18_2017-01-23 16:13:22
Current count: 6936000, row: 2017-01-23 15:50:36_2017-01-23 15:54:19
Current count: 6937000, row: 2017-01-23 15:54:08_2017-01-23 16:01:10
Current count: 6938000, row: 2017-01-23 15:57:43_2017-01-23 16:01:40
Current count: 6939000, row: 2017-01-23 16:01:28_2017-01-23 16:11:09
Current count: 6940000, row: 2017-01-23 16:05:09_2017-01-23 16:28:26
Current count: 6941000, row: 2017-01-23 16:09:06_2017-01-23 16:17:17
Current count: 6942000, row: 2017-01-23 16:13:03_2017-01-23 16:15:40
Current count: 6943000, row: 2017-01-23 16:17:00_2017-01-23 16:29:28
Current count: 6944000, row: 2017-01-23 16:21:05_2017-01-23 16:26:25
Current count: 6945000, row: 2017-01-23 16:25:05_2017-01-23 16:34:41
Current count: 6946000, row: 2017-01-23 16:29:14_2017-01-23 16:35:05
Current count: 6947000, row: 2017-01-23 16:33:07_2017-01-23 17:19:56
Current count: 6948000, row: 2017-01-23 16:37:11_2017-01-23 16:41:23
Current count: 6949000, row: 2017-01-23 16:41:05_2017-01-23 16:52:54
Current count: 6950000, row: 2017-01-23 16:45:24_2017-01-23 16:47:48
Current count: 6951000, row: 2017-01-23 16:49:22_2017-01-23 16:52:31
^C[root@ip-172-31-62-227 hbase]#
[root@ip-172-31-62-227 hbase]#
Search
ENG
IN
18:30
09-04-2024