Practica Hive

Consigna: Por cada ejercicio, escribir el código y agregar una captura de pantalla del resultado obtenido.

Diccionario de datos:
https://www.nyc.gov/assets/tlc/downloads/pdf/data_dictionary_trip_records_yellow.pdf

1. En Hive, crear las siguientes tablas (internas) en la base de datos tripdata en hive:
    a. payments(VendorID, tpep_pickup_datetetime, payment_type, total_amount)
    b. passengers(tpep_pickup_datetetime, passenger_count, total_amount)
    c. tolls (tpep_pickup_datetetime, passenger_count, tolls_amount, total_amount)
    d. congestion (tpep_pickup_datetetime, passenger_count, congestion_surcharge, total_amount)
    e. distance (tpep_pickup_datetetime, passenger_count, trip_distance, total_amount)

2. En Hive, hacer un 'describe' de las tablas passengers y distance.

3. Hacer ingest del file: Yellow_tripodata_2021-01.csv
https://dataengineerpublic.blob.core.windows.net/data-engineer/yellow_tripdata_2021-01.csv

Para los siguientes ejercicios, debes usar PySpark (obligatorio). Si deseas practicar más, también puedes repetir los mismos en SQL (opcional)

4. (Opcional SQL) Generar una vista

5. Insertar en la tabla payments (VendorID, tpep_pickup_datetetime, payment_type, total_amount) Solamente los pagos con tarjeta de crédito

Dataframe

```
root
 |-- VendorID: integer (nullable = true)
 |-- tpep_pickup_datetime: date (nullable = true)
 |-- payment_type: integer (nullable = true)
 |-- total_amount: double (nullable = true)
```

```
+--------+-------------------+------------+------------+
|VendorID|tpep_pickup_datetime|payment_type|total_amount|
+--------+-------------------+------------+------------+
|       1|         2021-01-01|           4|        28.8|
|       1|         2021-01-01|           4|        37.8|
|       1|         2021-01-01|           4|        16.3|
|       1|         2021-01-01|           4|         8.3|
|       1|         2021-01-01|           4|        10.8|
|       2|         2021-01-01|           4|       -45.3|
|       2|         2021-01-01|           4|        -8.8|
|       2|         2021-01-01|           4|       -15.8|
|       2|         2021-01-01|           4|       -25.8|
|       2|         2021-01-01|           4|       -15.8|
+--------+-------------------+------------+------------+
```

Table

```
hive> describe payments;
OK
vendorid                int
tpep_pickup_datetime    date
payment_type            int
total_amount            double
```

```
hive> select * from payments limit 10;
OK
1       2021-01-01      4       28.8
1       2021-01-01      4       37.8
1       2021-01-01      4       16.3
1       2021-01-01      4       8.3
1       2021-01-01      4       10.8
2       2021-01-01      4       -45.3
2       2021-01-01      4       -8.8
2       2021-01-01      4       -15.8
2       2021-01-01      4       -25.8
2       2021-01-01      4       -15.8
```

6. Insertar en la tabla passengers (tpep_pickup_datetetime, passenger_count, total_amount) los registros cuya cantidad de pasajeros sea mayor a 2 y el total del viaje cueste más de 8 dólares.

Dataframe

```
root
 |-- tpep_pickup_datetime: date (nullable = true)
 |-- passenger_count: integer (nullable = true)
 |-- total_amount: double (nullable = true)
```

```
+--------------------+---------------+------------+
|tpep_pickup_datetime|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|
+--------------------+---------------+------------+
```

Table

```
hive> describe passengers;
OK
tpep_pickup_datetime    date
passenger_count         int
total_amount            double
```

```
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
```

7. Insertar en la tabla tolls (tpep_pickup_datetetime, passenger_count, tolls_amount, total_amount) los registros que tengan pago de peajes mayores a 0.1 y cantidad de pasajeros mayores a 1.

Dataframe

```
root
 |-- tpep_pickup_datetime: date (nullable = true)
 |-- passenger_count: integer (nullable = true)
 |-- tolls_amount: double (nullable = true)
 |-- total_amount: double (nullable = true)
```

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
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

Table

```
hive> describe tolls;
OK
tpep_pickup_datetime    date
passenger_count         int
tolls_amount            double
total_amount            double
```

```
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
```

8. Insertar en la tabla congestion (tpep_pickup_datetetime, passenger_count, congestion_surcharge, total_amount) los registros que hayan tenido congestión en los viajes en la fecha 2021-01-18

Dataframe

```
root
 |-- tpep_pickup_datetime: date (nullable = true)
 |-- passenger_count: integer (nullable = true)
 |-- congestion_surcharge: double (nullable = true)
 |-- total_amount: double (nullable = true)
```

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	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
2021-01-18	1	2.5	20.8

Table

```
hive> describe congestion;
OK
tpep_pickup_datetime    date
passenger_count         int
congestion_surcharge    double
total_amount            double
```

```
hive> select * from congestion limit 10;
OK
2021-01-18      1       2.5     10.8
2021-01-18      1       2.5     16.56
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
2021-01-18      1       2.5     20.8
```

9. Insertar en la tabla distance (tpep_pickup_datetetime, passenger_count, trip_distance, total_amount) los registros de la fecha 2020-12-31 que hayan tenido solamente un pasajero (passenger_count = 1) y hayan recorrido más de 15 millas (trip_distance).

Dataframe

```
root
 |-- tpep_pickup_datetime: date (nullable = true)
 |-- passenger_count: integer (nullable = true)
 |-- trip_distance: double (nullable = true)
 |-- total_amount: double (nullable = true)
```

```
+-------------------+---------------+-------------+------------+
|tpep_pickup_datetime|passenger_count|trip_distance|total_amount|
+-------------------+---------------+-------------+------------+
|         2020-12-31|              1|        17.96|        53.3|
+-------------------+---------------+-------------+------------+
```

Table

```
hive> describe distance ;
OK
tpep_pickup_datetime    date
passenger_count         int
trip_distance           double
total_amount            double
```

```
hive> select * from distance limit 10 ;
OK
2020-12-31      1       17.96   53.3
```