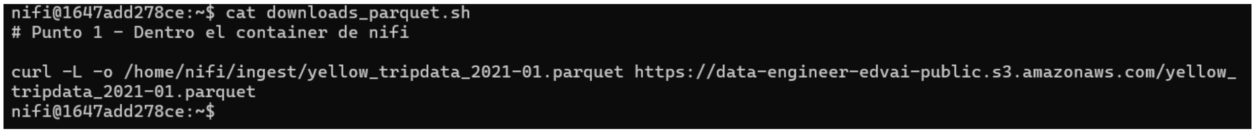
1. En el container de Nifi, crear un .sh que permita descargar el archivo yellow_tripdata_2021-01.parquet desde

   wget -O /home/fpineyro/test/yellow_tripdata_2021-01.parquet https://data-engineer-edvai-public.s3.amazonaws.com/yellow_tripdata_2021-01.parquet

   y lo guarde en /home/nifi/ingest.

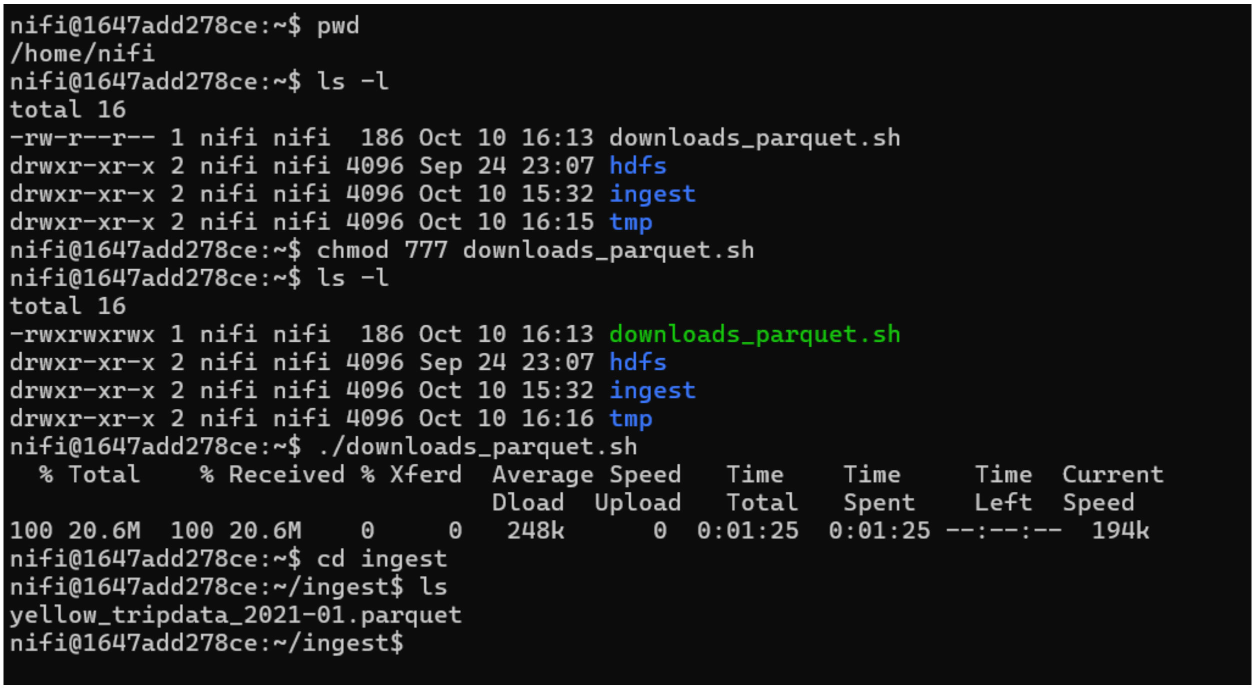
   Ejecutarlo

Uso curl porque no tiene instalado wget el contenedor



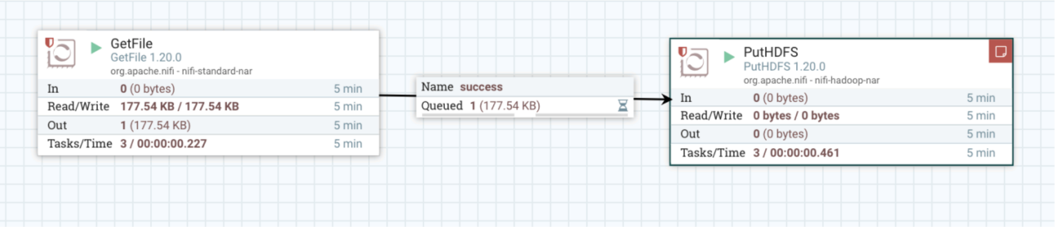
```
nifi@1647add278ce:~$ cat downloads_parquet.sh
# Punto 1 - Dentro el container de nifi

curl -L -o /home/nifi/ingest/yellow_tripdata_2021-01.parquet https://data-engineer-edvai-public.s3.amazonaws.com/yellow_
tripdata_2021-01.parquet
nifi@1647add278ce:~$
```



```
nifi@1647add278ce:~$ pwd
/home/nifi
nifi@1647add278ce:~$ ls -l
total 16
-rw-r--r-- 1 nifi nifi  186 Oct 10 16:13 downloads_parquet.sh
drwxr-xr-x 2 nifi nifi 4096 Sep 24 23:07 hdfs
drwxr-xr-x 2 nifi nifi 4096 Oct 10 15:32 ingest
drwxr-xr-x 2 nifi nifi 4096 Oct 10 16:15 tmp
nifi@1647add278ce:~$ chmod 777 downloads_parquet.sh
nifi@1647add278ce:~$ ls -l
total 16
-rwxrwxrwx 1 nifi nifi  186 Oct 10 16:13 downloads_parquet.sh
drwxr-xr-x 2 nifi nifi 4096 Sep 24 23:07 hdfs
drwxr-xr-x 2 nifi nifi 4096 Oct 10 15:32 ingest
drwxr-xr-x 2 nifi nifi 4096 Oct 10 16:16 tmp
nifi@1647add278ce:~$ ./downloads_parquet.sh
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 20.6M  100 20.6M    0     0   248k      0  0:01:25  0:01:25 --:--:--  194k
nifi@1647add278ce:~$ cd ingest
nifi@1647add278ce:~/ingest$ ls
yellow_tripdata_2021-01.parquet
nifi@1647add278ce:~/ingest$
```

2. Por medio de la interfaz gráfica de Nifi, crear un job que tenga dos procesos.
   a) GetFile para obtener el archivo del punto 1 (/home/nifi/ingest)
   b) putHDFS para ingestarlo a HDFS (directorio nifi)



GetFile
GetFile 1.20.0
org.apache.nifi - nifi-standard-nar

In	0 (0 bytes)	5 min
Read/Write	177.54 KB / 177.54 KB	5 min
Out	1 (177.54 KB)	5 min
Tasks/Time	3 / 00:00:00.227	5 min

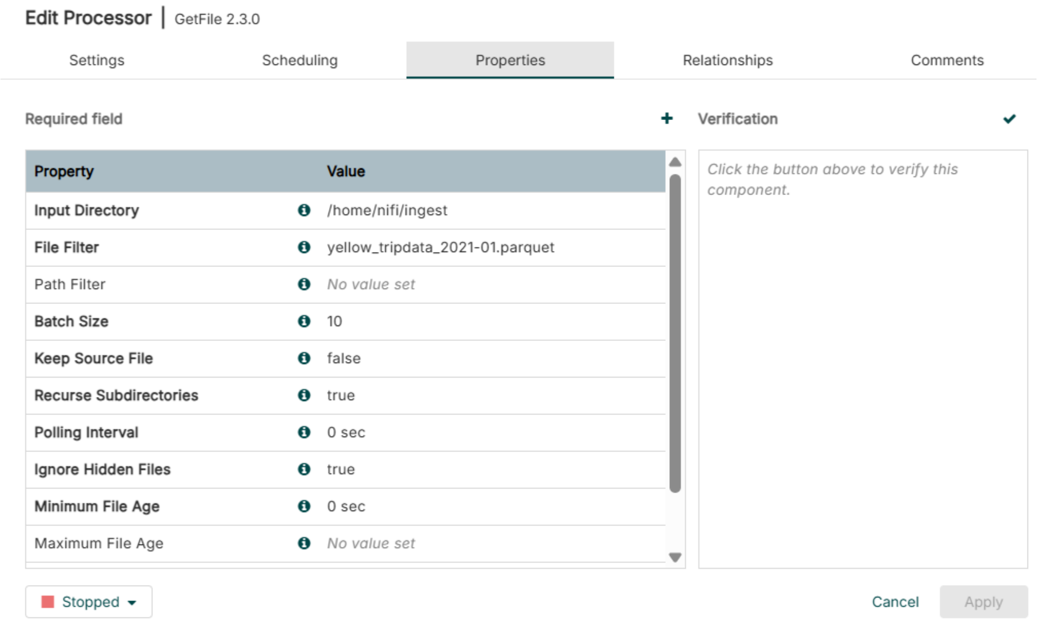
Name success
Queued 1 (177.54 KB)

PutHDFS
PutHDFS 1.20.0
org.apache.nifi - nifi-hadoop-nar

In	0 (0 bytes)	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	0 (0 bytes)	5 min
Tasks/Time	3 / 00:00:00.461	5 min

a) Creo el job de Nifi

b) Configuro GetFile y PutHDFS



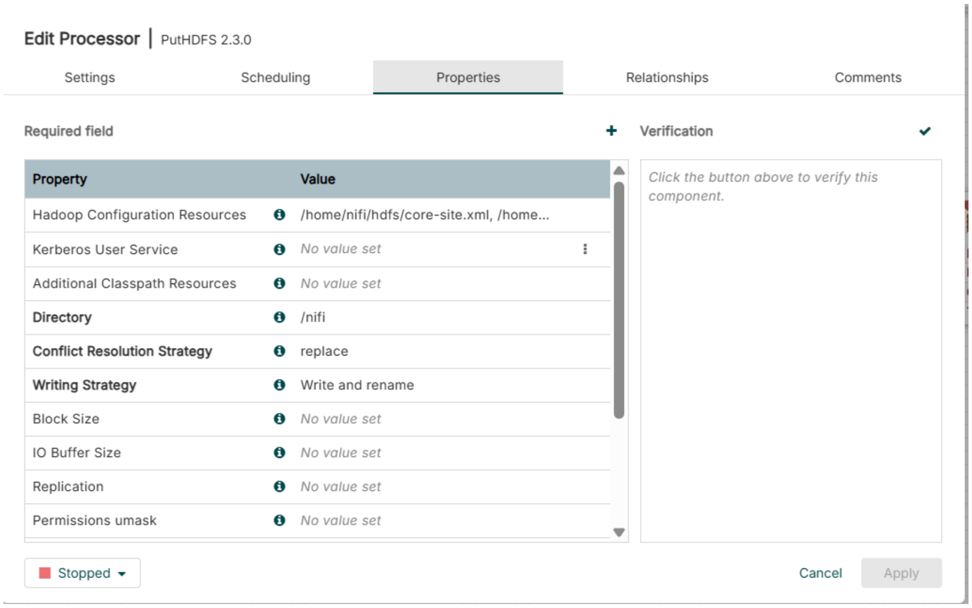
Edit Processor | GetFile 2.3.0

Required field

Verification

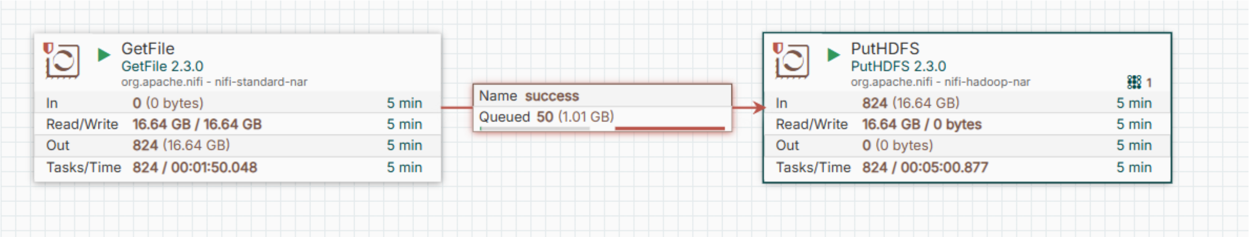
Property	Value
Input Directory	/home/nifi/ingest
File Filter	yellow_tripdata_2021-01.parquet
Path Filter	*No value set*
Batch Size	10
Keep Source File	false
Recurse Subdirectories	true
Polling Interval	0 sec
Ignore Hidden Files	true
Minimum File Age	0 sec
Maximum File Age	*No value set*

Click the button above to verify this component.



Property	Value
Hadoop Configuration Resources	/home/nifi/hdfs/core-site.xml, /home...
Kerberos User Service	No value set
Additional Classpath Resources	No value set
Directory	/nifi
Conflict Resolution Strategy	replace
Writing Strategy	Write and rename
Block Size	No value set
IO Buffer Size	No value set
Replication	No value set
Permissions umask	No value set

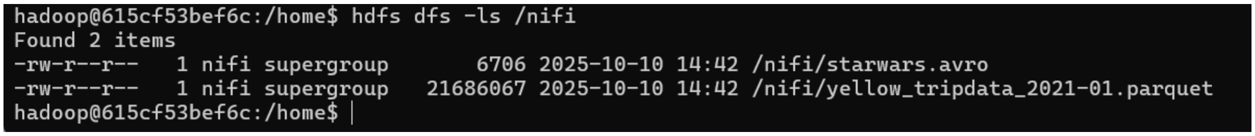
Click the button above to verify this component.



	GetFile	
In	0 (0 bytes)	5 min
Read/Write	16.64 GB / 16.64 GB	5 min
Out	824 (16.64 GB)	5 min
Tasks/Time	824 / 00:01:50.048	5 min

	PutHDFS	
In	824 (16.64 GB)	5 min
Read/Write	16.64 GB / 0 bytes	5 min
Out	0 (0 bytes)	5 min
Tasks/Time	824 / 00:05:00.877	5 min

c) Compruebo que el archivo se movió a hdfs

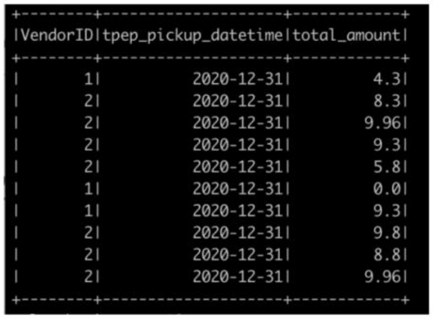


```
hadoop@615cf53bef6c:/home$ hdfs dfs -ls /nifi
Found 2 items
-rw-r--r--   1 nifi supergroup       6706 2025-10-10 14:42 /nifi/starwars.avro
-rw-r--r--   1 nifi supergroup   21686067 2025-10-10 14:42 /nifi/yellow_tripdata_2021-01.parquet
hadoop@615cf53bef6c:/home$ 
```

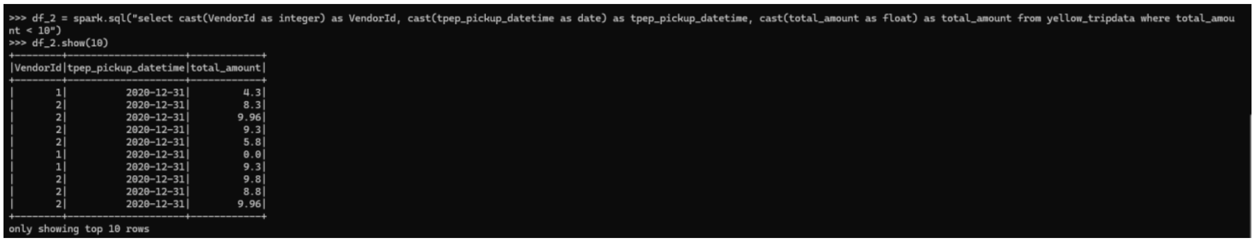
3. Con el archivo ya ingestado en HDFS/nifi, escribir las consultas y agregar captura de pantalla del resultado. Para los ejercicios puedes usar SQL mediante la creación de una vista llamada yellow_tripdata.
   También debes chequear el diccionario de datos por cualquier duda que tengas respecto a las columnas del archivo
   https://www.nyc.gov/assets/tlc/downloads/pdf/data_dictionary_trip_records_yellow.pdf

3.1) Mostrar los resultados siguientes

a. VendorId Integer
b. Tpep_pickup_datetime date
c. Total_amount double
d. Donde el total (total_amount sea menor a 10 dólares)

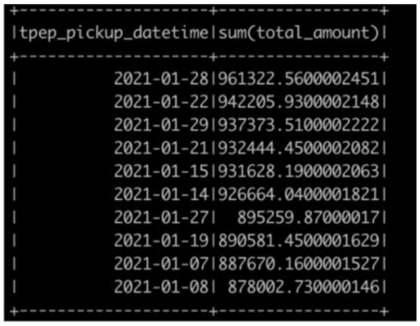


```
+--------+--------------------+------------+
|VendorID|tpep_pickup_datetime|total_amount|
+--------+--------------------+------------+
|       1|          2020-12-31|         4.3|
|       2|          2020-12-31|         8.3|
|       2|          2020-12-31|        9.96|
|       2|          2020-12-31|         9.3|
|       2|          2020-12-31|         5.8|
|       1|          2020-12-31|         0.0|
|       1|          2020-12-31|         9.3|
|       2|          2020-12-31|         9.8|
|       2|          2020-12-31|         8.8|
|       2|          2020-12-31|        9.96|
+--------+--------------------+------------+
```

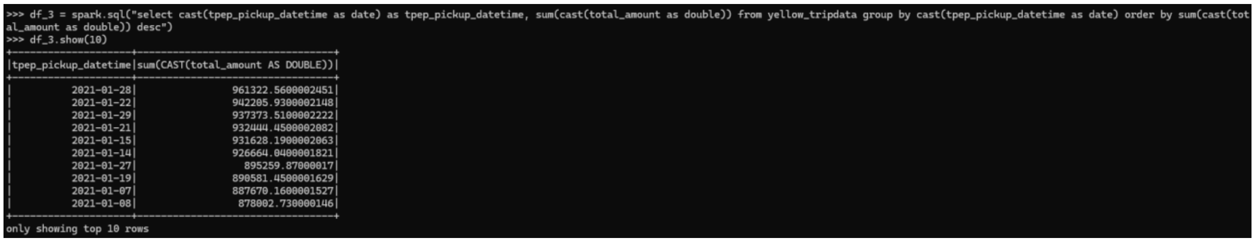


```
>>> df_2 = spark.sql("select cast(VendorId as integer) as VendorId, cast(tpep_pickup_datetime as date) as tpep_pickup_datetime, cast(total_amount as float) as total_amount from yellow_tripdata where total_amou
nt < 10")
>>> df_2.show(10)
+--------+--------------------+------------+
|VendorId|tpep_pickup_datetime|total_amount|
+--------+--------------------+------------+
|       1|          2020-12-31|         4.3|
|       2|          2020-12-31|         8.3|
|       2|          2020-12-31|        9.96|
|       2|          2020-12-31|         9.3|
|       2|          2020-12-31|         5.8|
|       1|          2020-12-31|         0.0|
|       1|          2020-12-31|         9.3|
|       2|          2020-12-31|         9.8|
|       2|          2020-12-31|         8.8|
|       2|          2020-12-31|        9.96|
+--------+--------------------+------------+
only showing top 10 rows
```

3.2) Mostrar los 10 días que más se recaudó dinero (tpep_pickup_datetime, total amount)

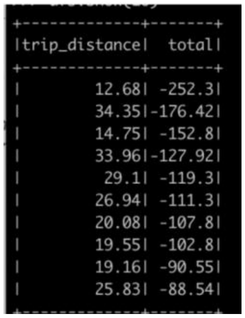


```
+--------------------+-----------------+
|tpep_pickup_datetime|sum(total_amount)|
+--------------------+-----------------+
|          2021-01-28|961322.5600002451|
|          2021-01-22|942205.9300002148|
|          2021-01-29|937373.5100002222|
|          2021-01-21|932444.4500002082|
|          2021-01-15|931628.1900002063|
|          2021-01-14|926664.0400001821|
|          2021-01-27|  895259.87000017|
|          2021-01-19|890581.4500001629|
|          2021-01-07|887670.1600001527|
|          2021-01-08| 878002.730000146|
+--------------------+-----------------+
```

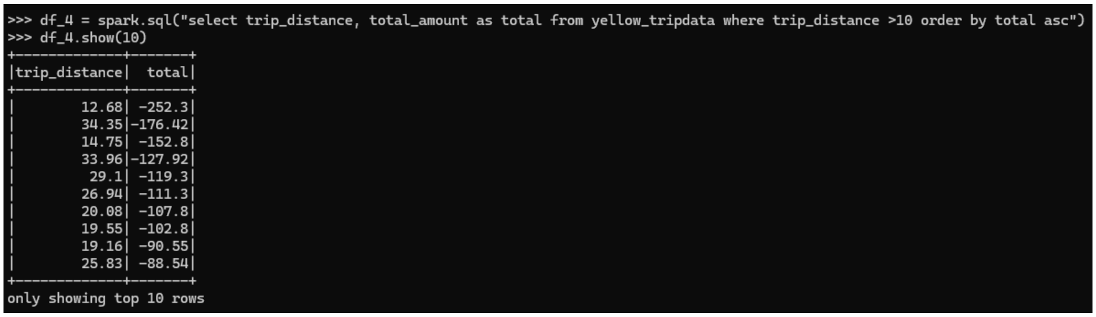


```
>>> df_3 = spark.sql("select cast(tpep_pickup_datetime as date) as tpep_pickup_datetime, sum(cast(total_amount as double)) from yellow_tripdata group by cast(tpep_pickup_datetime as date) order by sum(cast(tot
al_amount as double)) desc")
>>> df_3.show(10)
+--------------------+--------------------------------+
|tpep_pickup_datetime|sum(CAST(total_amount AS DOUBLE))|
+--------------------+--------------------------------+
|          2021-01-28|               961322.5600002451|
|          2021-01-22|               942205.9300002148|
|          2021-01-29|               937373.5100002222|
|          2021-01-21|               932444.4500002082|
|          2021-01-15|               931628.1900002063|
|          2021-01-14|               926664.0400001821|
|          2021-01-27|                 895259.87000017|
|          2021-01-19|               890581.4500001629|
|          2021-01-07|               887670.1600001527|
|          2021-01-08|                878002.730000146|
+--------------------+--------------------------------+
only showing top 10 rows
```

3.3) Mostrar los 10 viajes que menos dinero recaudó en viajes mayores a 10 millas (trip_distance, total_amount)

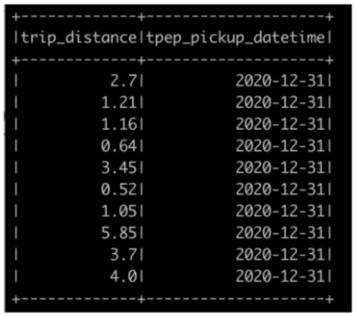


```
+-------------+-------+
|trip_distance|  total|
+-------------+-------+
|        12.68| -252.3|
|        34.35|-176.42|
|        14.75| -152.8|
|        33.96|-127.92|
|         29.1| -119.3|
|        26.94| -111.3|
|        20.08| -107.8|
|        19.55| -102.8|
|        19.16| -90.55|
|        25.83| -88.54|
+-------------+-------+
```



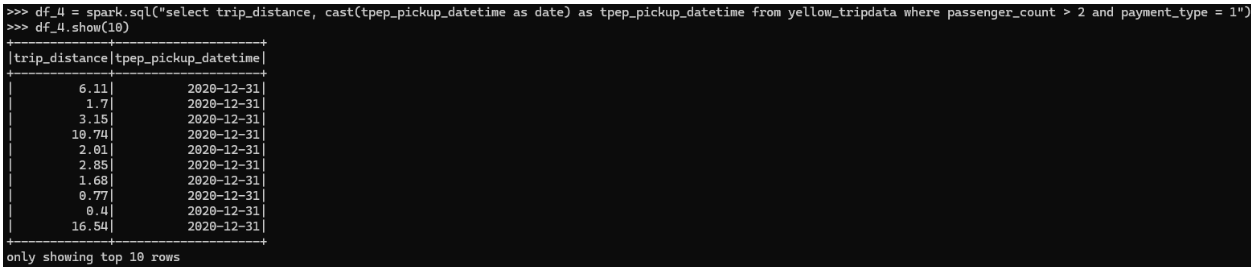
```
>>> df_4 = spark.sql("select trip_distance, total_amount as total from yellow_tripdata where trip_distance >10 order by total asc")
>>> df_4.show(10)
+-------------+-------+
|trip_distance|  total|
+-------------+-------+
|        12.68| -252.3|
|        34.35|-176.42|
|        14.75| -152.8|
|        33.96|-127.92|
|         29.1| -119.3|
|        26.94| -111.3|
|        20.08| -107.8|
|        19.55| -102.8|
|        19.16| -90.55|
|        25.83| -88.54|
+-------------+-------+
only showing top 10 rows
```

3.4) Mostrar los viajes de más de dos pasajeros que hayan pagado con tarjeta de crédito (mostrar solo las columnas trip_distance y tpep_pickup_datetime)



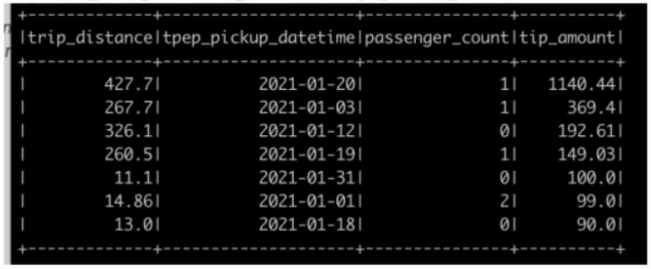
```
+-------------+--------------------+
|trip_distance|tpep_pickup_datetime|
+-------------+--------------------+
|          2.7|          2020-12-31|
|         1.21|          2020-12-31|
|         1.16|          2020-12-31|
|         0.64|          2020-12-31|
|         3.45|          2020-12-31|
|         0.52|          2020-12-31|
|         1.05|          2020-12-31|
|         5.85|          2020-12-31|
|          3.7|          2020-12-31|
|          4.0|          2020-12-31|
+-------------+--------------------+
```

NO SALE IGUAL

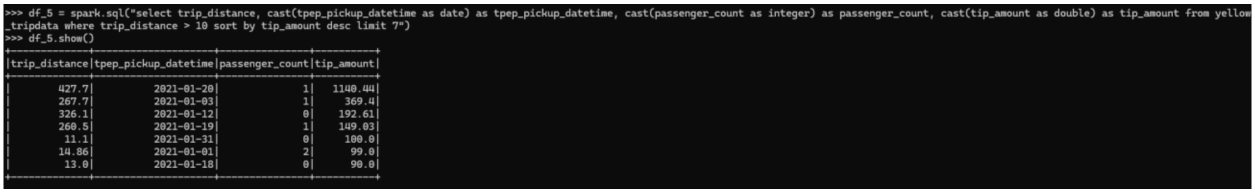


```
>>> df_4 = spark.sql("select trip_distance, cast(tpep_pickup_datetime as date) as tpep_pickup_datetime from yellow_tripdata where passenger_count > 2 and payment_type = 1")
>>> df_4.show(10)
+-------------+--------------------+
|trip_distance|tpep_pickup_datetime|
+-------------+--------------------+
|         6.11|          2020-12-31|
|          1.7|          2020-12-31|
|         3.15|          2020-12-31|
|        10.74|          2020-12-31|
|         2.01|          2020-12-31|
|         2.85|          2020-12-31|
|         1.68|          2020-12-31|
|         0.77|          2020-12-31|
|          0.4|          2020-12-31|
|        16.54|          2020-12-31|
+-------------+--------------------+
only showing top 10 rows
```

3.5) Mostrar los 7 viajes con mayor propina en distancias mayores a 10 millas (mostrar campos tpep_pickup_datetime, trip_distance, passenger_count, tip_amount)

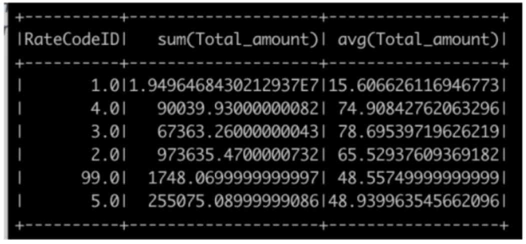


```
+-------------+--------------------+---------------+----------+
|trip_distance|tpep_pickup_datetime|passenger_count|tip_amount|
+-------------+--------------------+---------------+----------+
|        427.7|          2021-01-20|              1|   1140.44|
|        267.7|          2021-01-03|              1|     369.4|
|        326.1|          2021-01-12|              0|    192.61|
|        260.5|          2021-01-19|              1|    149.03|
|         11.1|          2021-01-31|              0|     100.0|
|        14.86|          2021-01-01|              2|      99.0|
|         13.0|          2021-01-18|              0|      90.0|
+-------------+--------------------+---------------+----------+
```

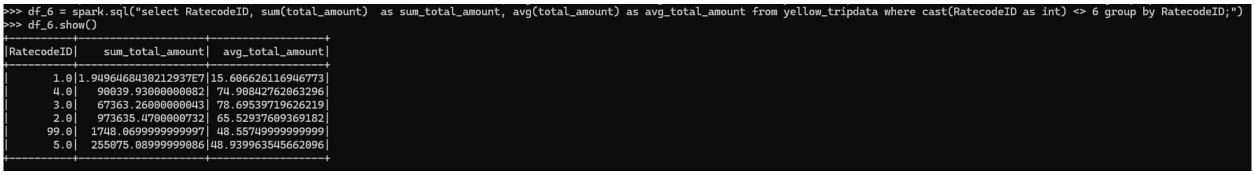


```
>>> df_5 = spark.sql("select trip_distance, cast(tpep_pickup_datetime as date) as tpep_pickup_datetime, cast(passenger_count as integer) as passenger_count, cast(tip_amount as double) as tip_amount from yellow
_tripdata where trip_distance > 10 sort by tip_amount desc limit 7")
>>> df_5.show()
+-------------+--------------------+---------------+----------+
|trip_distance|tpep_pickup_datetime|passenger_count|tip_amount|
+-------------+--------------------+---------------+----------+
|        427.7|          2021-01-20|              1|   1140.44|
|        267.7|          2021-01-03|              1|     369.4|
|        326.1|          2021-01-12|              0|    192.61|
|        260.5|          2021-01-19|              1|    149.03|
|         11.1|          2021-01-31|              0|     100.0|
|        14.86|          2021-01-01|              2|      99.0|
|         13.0|          2021-01-18|              0|      90.0|
+-------------+--------------------+---------------+----------+
```

3.6) Mostrar para cada uno de los valores de RateCodeID, el monto total y el monto promedio. Excluir los viajes en donde RateCodeID es 'Group Ride'



```
+----------+--------------------+------------------+
|RateCodeID|   sum(Total_amount)| avg(Total_amount)|
+----------+--------------------+------------------+
|       1.0|1.9496468430212937E7|15.606626116946773|
|       4.0|   90039.93000000082| 74.90842762063296|
|       3.0|   67363.26000000043| 78.69539719626219|
|       2.0|   973635.4700000732| 65.52937609369182|
|      99.0|  1748.0699999999997| 48.55749999999999|
|       5.0|  255075.08999999086|48.939963545662096|
+----------+--------------------+------------------+
```



```
>>> df_6 = spark.sql("select RatecodeID, sum(total_amount)  as sum_total_amount, avg(total_amount) as avg_total_amount from yellow_tripdata where cast(RatecodeID as int) <> 6 group by RatecodeID;")
>>> df_6.show()
+----------+--------------------+------------------+
|RatecodeID|    sum_total_amount|  avg_total_amount|
+----------+--------------------+------------------+
|       1.0|1.9496468430212937E7|15.606626116946773|
|       4.0|   90039.93000000082| 74.90842762063296|
|       3.0|   67363.26000000043| 78.69539719626219|
|       2.0|   973635.4700000732| 65.52937609369182|
|      99.0|  1748.0699999999997| 48.55749999999999|
|       5.0|  255075.08999999086|48.939963545662096|
+----------+--------------------+------------------+
```