

Ejercicio 3.1 Mostrar los resultados siguientes

- VendorId Integer
- Tpep_pickup_datetime date
- Total_amount double
- Donde el total (total_amount sea menor a 10 dólares)

Resultado:

```
>>> ejercicio_3_1=spark.sql("Select VendorId, cast(Tpep_pickup_datetime as date), Total_amount from yellow_tripdata where Total_amount<10")
>>> ejercicio_3_1.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)

```
>>> ejercicio_3_2=spark.sql("Select cast(Tpep_pickup_datetime as date) as Fecha, SUM(Total_amount) as Total_amount FROM yellow_tripdata Group by Fecha order by Total_amount desc LIMIT 10")
>>> ejercicio_3_2.show()
```

Fecha	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

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

```
>>> spark.sql("Select trip_distance, total_amount FROM yellow_tripdata where trip_distance>10 order by total_amount asc LIMIT 10").show()
```

trip_distance	total_amount
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

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)

```
>>> spark.sql("Select trip_distance, Tpep_pickup_datetime FROM yellow_tripdata where payment_type=1 and passenger_count>2").show(10)
```

trip_distance	Tpep_pickup_datetime
6.11	2020-12-31 21:15:52
1.7	2020-12-31 21:31:06
3.15	2020-12-31 21:34:37
10.74	2020-12-31 21:19:57
2.01	2020-12-31 21:28:07
2.85	2020-12-31 21:08:04
1.68	2020-12-31 21:22:02
0.77	2020-12-31 21:33:33
0.4	2020-12-31 21:45:29
16.54	2020-12-31 21:36:53

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)

```
>>> spark.sql("Select trip_distance, cast(Tpep_pickup_datetime as date), cast(passenger_count as int),tip_amount FROM yellow_tripdata where trip_distance>10 order by tip_amount desc LIMIT 7").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'

```
>>> spark.sql("Select RateCodeID, SUM(total_amount), AVG(total_amount) FROM yellow_tripdata where RateCodeID<>6 group by RateCodeID").show()
```

RateCodeID	sum(total_amount)	avg(total_amount)
1.0	1.9496468430212937E7	15.606626116946773
4.0	90039.930000000082	74.90842762063296
3.0	67363.260000000043	78.69539719626219
2.0	973635.47000000732	65.52937609369182
99.0	1748.0699999999997	48.55749999999999
5.0	255075.089999999086	48.939963545662096