

# TLC Trips Record

## Practice Case 6

Nurjanah

In [1]: `import pyspark`

In [2]: `from pyspark.sql import SparkSession  
from pyspark.sql.functions import to_timestamp`

In [3]: `!wget https://d37ci6vzurychx.cloudfront.net/trip-data/yellow_tripdata_2021-02.parquet  
!wget https://d37ci6vzurychx.cloudfront.net/trip-data/fhv_tripdata_2021-02.parquet  
!wget https://d37ci6vzurychx.cloudfront.net/trip-data/fhvhv_tripdata_2021-02.parquet`

```
--2022-12-11 17:20:34-- https://d37ci6vzurychx.cloudfront.net/trip-data/yellow_tripdata_2021-02.parquet
Resolving d37ci6vzurychx.cloudfront.net (d37ci6vzurychx.cloudfront.net)... 18.160.201.50, 18.160.201.5, 18.160.201.126, ...
Connecting to d37ci6vzurychx.cloudfront.net (d37ci6vzurychx.cloudfront.net)|18.160.201.50|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 21777258 (21M) [application/x-www-form-urlencoded]
Saving to: 'yellow_tripdata_2021-02.parquet.2'
```

```
yellow_tripdata_202 100%[=====>] 20.77M 41.2MB/s in 0.5s
```

```
2022-12-11 17:20:34 (41.2 MB/s) - 'yellow_tripdata_2021-02.parquet.2' saved [21777258/21777258]
```

```
--2022-12-11 17:20:35-- https://d37ci6vzurychx.cloudfront.net/trip-data/fhv_tripdata_2021-02.parquet
Resolving d37ci6vzurychx.cloudfront.net (d37ci6vzurychx.cloudfront.net)... 18.160.201.131, 18.160.201.126, 18.160.201.5, ...
Connecting to d37ci6vzurychx.cloudfront.net (d37ci6vzurychx.cloudfront.net)|18.160.201.131|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 10645466 (10M) [binary/octet-stream]
Saving to: 'fhv_tripdata_2021-02.parquet.3'
```

```
fhv_tripdata_2021-0 100%[=====>] 10.15M 51.6MB/s in 0.2s
```

```
2022-12-11 17:20:35 (51.6 MB/s) - 'fhv_tripdata_2021-02.parquet.3' saved [10645466/10645466]
```

```
--2022-12-11 17:20:35-- https://d37ci6vzurychx.cloudfront.net/trip-data/fhvhv_tripdata_2021-02.parquet
Resolving d37ci6vzurychx.cloudfront.net (d37ci6vzurychx.cloudfront.net)... 18.160.201.131, 18.160.201.126, 18.160.201.5, ...
Connecting to d37ci6vzurychx.cloudfront.net (d37ci6vzurychx.cloudfront.net)|18.160.201.131|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 302633211 (289M) [application/x-www-form-urlencoded]
Saving to: 'fhvhv_tripdata_2021-02.parquet.1'
```

```
fhhvh_tripdata_2021 100%[=====>] 288.61M 147MB/s in 2.0s
```

```
2022-12-11 17:20:37 (147 MB/s) - 'fhhvh_tripdata_2021-02.parquet.1' saved [30263321
1/302633211]
```

In [4]:

```
spark = SparkSession.builder \
    .master("local[*]") \
    .appName('test') \
    .getOrCreate()
```

```
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.spark.unsafe.Platform (file:/usr/local/spark-3.1.2-bin-hadoop3.2/jars/spark-unsafe_2.12-3.1.2.jar) to constructor java.nio.DirectByteBuffer(long,int)
WARNING: Please consider reporting this to the maintainers of org.apache.spark.unsafe.Platform
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
22/12/11 17:20:40 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
22/12/11 17:20:41 WARN Utils: Service 'SparkUI' could not bind on port 4040. Attempting port 4041.
22/12/11 17:20:41 WARN Utils: Service 'SparkUI' could not bind on port 4041. Attempting port 4042.
22/12/11 17:20:41 WARN Utils: Service 'SparkUI' could not bind on port 4042. Attempting port 4043.
```

In [5]:

```
df_yellow = spark.read.parquet("yellow_tripdata_2021-02.parquet")
df_fhv = spark.read.parquet("fhv_tripdata_2021-02.parquet")
df_fhhvh = spark.read.parquet("fhhvh_tripdata_2021-02.parquet")
```

In [6]:

```
df_yellow.show(5)
```

```
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+
|VendorID|tpep_pickup_datetime|tpep_dropoff_datetime|passenger_count|trip_distance|RatecodeID|store_and_fwd_flag|PULocationID|DOLocationID|payment_type|fare_amount|extra|mta_tax|tip_amount|tolls_amount|improvement_surcharge|total_amount|congestion_surcharge|airport_fee|
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+
|1|2021-02-01 00:40:47|2021-02-01 00:48:28|1.0|2.3|1.0|N|141|226|2|8.5|3.0|0.5|0.0|0.0|0.3|12.3|2.5|null|
|1|2021-02-01 00:07:44|2021-02-01 00:20:31|1.0|1.6|1.0|N|43|263|2|9.5|3.0|0.5|0.0|0.0|0.3|13.3|0.0|null|
```

1.0	1	2021-02-01 00:59:36	2021-02-01 01:24:13	1.0	5.3
0.5	N	114	263	2	19.0  3.0
0.0	0.0	0.0	0.3	22.8	2.5
1.0	2	2021-02-01 00:03:26	2021-02-01 00:16:32	1.0	2.79
0.5	N	236	229	1	11.0  0.5
0.0	2.96	0.0	0.3	17.76	2.5
1.0	2	2021-02-01 00:20:20	2021-02-01 00:24:03	2.0	0.64
0.5	N	229	140	1	4.5  0.5
0.0	1.66	0.0	0.3	9.96	2.5

only showing top 5 rows

In [7]:

df\_fhv.show(5)

dispatching_base_num	pickup_datetime	dropOff_datetime	PULocationID	DOLocationID
B00013	2021-02-01 00:01:00	2021-02-01 01:33:00	null	n
B00014				
B00021	2021-02-01 00:55:40	2021-02-01 01:06:20	173.0	8
B00021	2021-02-01 00:14:03	2021-02-01 00:28:37	173.0	5
B00021	2021-02-01 00:27:48	2021-02-01 00:35:45	82.0	12
B00037	2021-02-01 00:12:50	2021-02-01 00:26:38	null	22
B00037				

only showing top 5 rows

In [8]:

df\_fhvhw.show(5)

[Stage 8:=====> (2 + 1) / 3]

hvfhs_license_num	dispatching_base_num	originating_base_num	request_datetime	on_scene_datetime	pickup_datetime	dropoff_datetime	PULocationID	DOLocationID	trip_miles	trip_time	base_passenger_fare	tolls	bcf	sales_tax	congestion_surcharge	airport_fee	tips	driver_pay	shared_request_flag	shared_match_flag	access_a_ride_flag	wav_request_flag	wav_match_flag
HV0003	B02764	B02764	2021-01-31 23:59:00	2021-01-31 23:59:00	2021-01-31 23:59:00	2021-01-31 23:59:00	2021-01-31 23:59:00	2021-01-31 23:59:00	2021-01-31 23:59:00	2021-01-31 23:59:00	2021-01-31 23:59:00	2021-01-31 23:59:00	2021-01-31 23:59:00	2021-01-31 23:59:00	2021-01-31 23:59:00	2021-01-31 23:59:00	2021-01-31 23:59:00	2021-01-31 23:59:00	2021-01-31 23:59:00	2021-01-31 23:59:00	2021-01-31 23:59:00	2021-01-31 23:59:00	

```

1-02-01 00:10:19|2021-02-01 00:10:40|2021-02-01 00:21:09|          35|          39|
2.06|          629|          17.14|  0.0|0.51|          1.52|          0.0|
null| 0.0|          9.79|          N|          N|          |
N|          N|
|          HV0003|          B02764|          B02764|2021-02-01 00:13:35|202
1-02-01 00:25:23|2021-02-01 00:27:23|2021-02-01 00:44:01|          39|          35|
3.15|          998|          32.11|  0.0|0.96|          2.85|          0.0|
null| 0.0|          24.01|          N|          N|          |
N|          N|
|          HV0005|          B02510|          null|2021-02-01 00:12:55|
null|2021-02-01 00:28:38|2021-02-01 00:38:27|          39|          91|          1.776|
589|          12.67|  0.0|0.38|          1.12|          0.0|          null| 0.0|
6.91|          N|          N|          N|          N|
N|
|          HV0005|          B02510|          null|2021-02-01 00:36:01|
null|2021-02-01 00:43:37|2021-02-01 01:23:20|          91|          228|          13.599|
2383|          37.82|  0.0|0.98|          2.91|          0.0|          null| 7.0|
35.05|          N|          N|          N|          N|
N|
|          HV0003|          B02872|          B02872|2021-01-31 23:57:50|202
1-02-01 00:08:25|2021-02-01 00:08:42|2021-02-01 00:17:57|          126|          250|
2.62|          555|          15.56|  0.0|0.47|          1.38|          0.0|
null| 0.0|          8.53|          N|          N|          |
N|          N|
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+
only showing top 5 rows

```

In [9]: `df_yellow.columns`

Out[9]:

```

['VendorID',
 'tpep_pickup_datetime',
 'tpep_dropoff_datetime',
 'passenger_count',
 'trip_distance',
 'RatecodeID',
 'store_and_fwd_flag',
 'PUlocationID',
 'DOlocationID',
 'payment_type',
 'fare_amount',
 'extra',
 'mta_tax',
 'tip_amount',
 'tolls_amount',
 'improvement_surcharge',
 'total_amount',
 'congestion_surcharge',
 'airport_fee']

```

In [10]: `df_fhv.columns`

Out[10]:

```

['dispatching_base_num',
 'pickup_datetime',
 'dropOff_datetime',
 'PUlocationID',
 'DOlocationID',

```

```
'SR_Flag',
'Affiliated_base_number']
```

```
In [11]: df_fhvvhv.columns
```

```
Out[11]: ['hvfhs_license_num',
'dispatching_base_num',
'originating_base_num',
'request_datetime',
'on_scene_datetime',
'pickup_datetime',
'dropoff_datetime',
'PULocationID',
'DOLocationID',
'trip_miles',
'trip_time',
'base_passenger_fare',
'tolls',
'bcf',
'sales_tax',
'congestion_surcharge',
'airport_fee',
'tips',
'driver_pay',
'shared_request_flag',
'shared_match_flag',
'access_a_ride_flag',
'wav_request_flag',
'wav_match_flag']
```

## How many taxi trips were there on February 15?

```
In [12]: df_yellow.registerTempTable('df_yellow_table')
total_trip_2021_02 = spark.sql("""
                                SELECT COUNT(tpep_pickup_datetime) AS total_trip_202
                                FROM df_yellow_table
                                WHERE tpep_pickup_datetime >= '2021-02-15 00:00:00'

                                """)
total_trip_2021_02.show()
```

```
+-----+
|total_trip_2021_02|
+-----+
|              40322|
+-----+
```

```
In [14]: df_fhvvhv.registerTempTable('df_fhvvhv_table')
total_trip_2021_02 = spark.sql("""
                                SELECT COUNT(pickup_datetime) AS total_trip_2021_02
                                FROM df_fhvvhv_table
                                WHERE pickup_datetime >= '2021-02-15 00:00:00' AND p

                                """)
total_trip_2021_02.show()
```

```
[Stage 11:=====> (3 + 1) / 4]
+-----+
|total_trip_2021_02|
+-----+
```

```
|          367170|
+-----+
```

## Find the longest trip for each day ?

```
In [16]: df_yellow = df_yellow.withColumn("pickup_date", df_yellow["tpep_pickup_datetime"].cast('date'))
df_yellow.registerTempTable('df_table_yellow')

longest_trip_byday_yellow = spark.sql("""
    SELECT pickup_date, MAX(trip_distance) as longest_distance_trip
    FROM df_table_yellow
    WHERE pickup_date >= "2021-02-01" AND pickup_date < "2021-03-01"
    GROUP BY 1
    ORDER BY longest_distance_trip_by_day DESC
""")
longest_trip_byday_yellow.show(10)
```

[Stage 14:=====> (133 + 4) / 200]

```
+-----+
|pickup_date|longest_distance_trip_by_day|
+-----+
| 2021-02-16|                221188.25|
| 2021-02-20|                188054.03|
| 2021-02-08|                186617.92|
| 2021-02-07|                186510.67|
| 2021-02-03|                186079.73|
| 2021-02-17|                140145.44|
| 2021-02-13|                115928.92|
| 2021-02-05|                 91134.16|
| 2021-02-26|                 90796.21|
| 2021-02-24|                 90073.44|
+-----+
```

only showing top 10 rows

```
In [18]: df_fhvvhv = df_fhvvhv.withColumn("pickup_date", df_fhvvhv["pickup_datetime"].cast('date'))
df_fhvvhv.registerTempTable('df_fhvvhv_table')

longest_trip_byday_fhvvhv = spark.sql("""
    SELECT pickup_date, MAX(trip_miles) as longest_distance_trip_by
    FROM df_fhvvhv_table
    WHERE pickup_date >= "2021-02-01" AND pickup_date < "2021-03-01"
    GROUP BY 1
    ORDER BY longest_distance_trip_by_day DESC
""")
longest_trip_byday_fhvvhv.show(10)
```

[Stage 16:=====> (160 + 4) / 200]

```
+-----+
|pickup_date|longest_distance_trip_by_day|
+-----+
| 2021-02-18|                527.11|
| 2021-02-10|                512.5|
| 2021-02-09|                480.73|
| 2021-02-27|                454.49|
| 2021-02-22|                347.41|
+-----+
```

2021-02-20	340.64
2021-02-19	329.16
2021-02-17	324.19
2021-02-16	307.661
2021-02-24	301.73

+-----+

only showing top 10 rows

## Find Top 5 Most frequent dispatching\_base\_num ?

In [19]:

```
df_fhv.registerTempTable('df_fhv_table')

top5_most_dbm_yellow = spark.sql("""
    SELECT dispatching_base_num, count(dispatching_base_num) as count
    FROM df_fhv_table
    GROUP BY 1
    ORDER BY 2 DESC
    """)
top5_most_dbm_yellow.show(5)
```

[Stage 18:===== > (188 + 4) / 200]

dispatching_base_num	count
B00856	35077
B01312	33089
B01145	31114
B02794	30397
B03016	29794

+-----+

only showing top 5 rows

In [20]:

```
top5_most_dbm_fhvfv = spark.sql("""
    SELECT dispatching_base_num, count(dispatching_base_num) as count
    FROM df_fhvfv_table
    GROUP BY 1
    ORDER BY 2 DESC
    """)
top5_most_dbm_fhvfv.show(5)
```

[Stage 19:===== > (3 + 1) / 4]

dispatching_base_num	count
B02510	3233664
B02764	965568
B02872	882689
B02875	685390
B02765	559768

+-----+

only showing top 5 rows

## Find Top 5 Most common location pairs (PUlocationID and DOlocationID)

```
In [21]: top5_most_PUlocationID_yellow = spark.sql("""
        SELECT PUlocationID, COUNT(PUlocationID) as Count_DOlocationID
        FROM df_yellow_table
        GROUP BY 1
        ORDER BY 2 DESC
        """)

top5_most_PUlocationID_yellow.show(5)
```

```
+-----+-----+
|PUlocationID|Count_DOlocationID|
+-----+-----+
|          236|          74898|
|          237|          72887|
|          141|          46222|
|          239|          45036|
|          186|          44295|
+-----+-----+
only showing top 5 rows
```

```
In [22]: top5_most_PUlocationID_fhvfv = spark.sql("""
        SELECT PUlocationID, COUNT(PUlocationID) as Count_DOlocationID
        FROM df_fhvfv_table
        GROUP BY 1
        ORDER BY 2 DESC
        """)

top5_most_PUlocationID_fhvfv.show(5)
```

```
[Stage 23:=====> (3 + 1) / 4]
+-----+-----+
|PUlocationID|Count_DOlocationID|
+-----+-----+
|          61|          203777|
|          76|          166959|
|          37|          140636|
|          79|          137901|
|          42|          137246|
+-----+-----+
only showing top 5 rows
```

```
In [92]: top5_most_DOlocationID_yellow = spark.sql("""
        SELECT DOlocationID, COUNT(DOlocationID) as Count_DOlocationID
        FROM df_yellow_table
        GROUP BY 1
        ORDER BY 2 DESC
        """)

top5_most_DOlocationID_yellow.show(5)
```

```
+-----+-----+
|DOlocationID|Count_DOlocationID|
+-----+-----+
|          236|          73310|
|          237|          61979|
|          141|          44436|
+-----+-----+
```



	239	42309
	238	41055
+-----+		

In [23]:

```
top5_most_D0locationID_fhv hv = spark.sql("""
    SELECT D0locationID, COUNT(D0locationID) as Count_D0locatio
    FROM df_fhv hv_table
    GROUP BY 1
    ORDER BY 2 DESC
    """)

top5_most_D0locationID_fhv hv.show(5)
```

[Stage 25:=====&gt;

(3 + 1) / 4]

+-----+		
D0locationID	Count_D0locationID	
+-----+		
	265	357369
	61	202250
	76	168094
	37	140826
	42	127432
+-----+		

only showing top 5 rows

In [ ]: