

```
%sh
```

READY

Spark Scala HDFS Hive and Impala

```
hdfs dfs -mkdir -p /nyctaxi/data
hdfs dfs -mkdir -p /nyctaxi/lookup
```

```
hdfs dfs -put -f /home/cloudera/Case_Study/yellow_tripdata_2020-01.csv /nyctaxi/data
hdfs dfs -put -f /home/cloudera/Case_Study/yellow_tripdata_2020-02.csv /nyctaxi/data
hdfs dfs -put -f /home/cloudera/Case_Study/taxi+_zone_lookup.csv /nyctaxi/lookup
```

```
hdfs dfs -ls /nyctaxi/data
hdfs dfs -ls /nyctaxi/lookup
```

```
%spark
sc
```

READY

```
res1: org.apache.spark.SparkContext = org.apache.spark.SparkContext@49985d2
```

```
%spark
```

READY

```
val trips01_DF = sqlContext.read.format("csv")
  .option("header", "true")
  .option("inferSchema", "true")
  .load("/nyctaxi/data/yellow_tripdata 2020-01.csv")
```

```
val trips02_DF = sqlContext.read.format("csv")
  .option("header", "true")
  .option("inferSchema", "true")
  .load("/nyctaxi/data/yellow_tripdata 2020-02.csv")
```

```
trips01_DF.printSchema()
trips02_DF.printSchema()
trips01_DF.show(5)
trips02_DF.show(5)
```

```
root
|-- VendorID: integer (nullable = true)
|-- tpep_pickup_datetime: timestamp (nullable = true)
|-- tpep_dropoff_datetime: timestamp (nullable = true)
|-- passenger_count: integer (nullable = true)
|-- trip_distance: double (nullable = true)
|-- RatecodeID: integer (nullable = true)
|-- store_and_fwd_flag: string (nullable = true)
|-- PULocationID: integer (nullable = true)
|-- DOLocationID: integer (nullable = true)
|-- payment_type: integer (nullable = true)
|-- fare_amount: double (nullable = true)
|-- extra: double (nullable = true)
|-- mta_tax: double (nullable = true)
|-- tip_amount: double (nullable = true)
|-- tolls_amount: double (nullable = true)
|-- improvement_surcharge: double (nullable = true)
```

```
%spark
```

READY

```
val trips_union_DF = trips01_DF.union(trips02_DF).toDF
```

```
trips_union_DF: org.apache.spark.sql.DataFrame = [VendorID: int, tpep_pickup_datetime: timestamp ... 16 more fields]
```

```
%spark
```

READY

```
var trips_DF = trips_union_DF.withColumn("year", year(to_date($"tpep_pickup_datetime")))
                             .withColumn("month", month(to_date($"tpep_pickup_datetime")))
```

```
trips DF: org.apache.spark.sql.DataFrame = [VendorID: int, tpep pickup datetime: timestamp ... 18 more fields]
```

```
%spark
```

READY

```
trips_DF.agg(min("passenger_count"),max("passenger_count"), min("trip distance"),max("trip distance"),min("total_amount"),max("total_amount")).show()
```

min(passenger_count)	max(passenger_count)	min(trip_distance)	max(trip_distance)	min(total_amount)	max(total_amount)
0	9	-30.62	210240.07	-1242.3	6061.42

```
%spark
```

READY

```
var trips_clean_DF = trips_DF.filter(trips_DF("VendorID") === "1" || trips_DF("VendorID") === "2")
  .filter(trips_DF("year") === "2020")
  .filter(trips_DF("month") === "1" || trips_DF("month") === "2")
  .filter(trips_DF("passenger_count") >= "1")
  .filter(trips_DF("trip_distance") >= "1" && trips_DF("trip_distance") != "210240.07")
  .filter(trips_DF("total_amount") >= "0")
```

```
trips_clean_DF: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [VendorID: int, tpep_pickup_datetime: timestamp ... 18 more fields]
```

```
%spark
```

READY

```
trips_clean_DF.agg(min("passenger_count"),max("passenger_count"), min("trip_distance"),max("trip_distance"),min("total_amount"),max("total_amount")).show()
```

min(passenger_count)	max(passenger_count)	min(trip_distance)	max(trip_distance)	min(total_amount)	max(total_amount)
1	9	1.0	369.94	0.0	6061.42

```
%spark
```

READY

```
trips_clean_DF.write
    .mode("overwrite")
    .partitionBy("year","month")
    .parquet("/nycdata")
```

```
%sh
```

READY

```
hdfs dfs -ls -R -h /nycdata
```

-rw-r--r-- 1 root supergroup 0 2022-06-16 16:53 /nycdata/_SUCCESS	
drwxr-xr-x - root supergroup 0 2022-06-16 16:53 /nycdata/year=2020	
drwxr-xr-x - root supergroup 0 2022-06-16 16:53 /nycdata/year=2020/month=1	
-rw-r--r-- 1 root supergroup 20.7 M 2022-06-16 16:43 /nycdata/year=2020/month=1/part-00000-d2514daa-c589-4357-a932-224eb3716069.c000.snappy.parquet	
-rw-r--r-- 1 root supergroup 20.0 M 2022-06-16 16:46 /nycdata/year=2020/month=1/part-00001-d2514daa-c589-4357-a932-224eb3716069.c000.snappy.parquet	
-rw-r--r-- 1 root supergroup 19.9 M 2022-06-16 16:46 /nycdata/year=2020/month=1/part-00002-d2514daa-c589-4357-a932-224eb3716069.c000.snappy.parquet	
-rw-r--r-- 1 root supergroup 20.1 M 2022-06-16 16:47 /nycdata/year=2020/month=1/part-00003-d2514daa-c589-4357-a932-224eb3716069.c000.snappy.parquet	
-rw-r--r-- 1 root supergroup 7.3 M 2022-06-16 16:48 /nycdata/year=2020/month=1/part-00004-d2514daa-c589-4357-a932-224eb3716069.c000.snappy.parquet	
-rw-r--r-- 1 root supergroup 10.7 K 2022-06-16 16:49 /nycdata/year=2020/month=1/part-00005-d2514daa-c589-4357-a932-224eb3716069.c000.snappy.parquet	
drwxr-xr-x - root supergroup 0 2022-06-16 16:53 /nycdata/year=2020/month=2	
-rw-r--r-- 1 root supergroup 4.3 K 2022-06-16 16:44 /nycdata/year=2020/month=2/part-00001-d2514daa-c589-4357-a932-224eb3716069.c000.snappy.parquet	
-rw-r--r-- 1 root supergroup 4.4 K 2022-06-16 16:46 /nycdata/year=2020/month=2/part-00002-d2514daa-c589-4357-a932-224eb3716069.c000.snappy.parquet	
-rw-r--r-- 1 root supergroup 5.3 K 2022-06-16 16:48 /nycdata/year=2020/month=2/part-00004-d2514daa-c589-4357-a932-224eb3716069.c000.snappy.parquet	
-rw-r--r-- 1 root supergroup 20.1 M 2022-06-16 16:49 /nycdata/year=2020/month=2/part-00005-d2514daa-c589-4357-a932-224eb3716069.c000.snappy.parquet	
-rw-r--r-- 1 root supergroup 20.0 M 2022-06-16 16:50 /nycdata/year=2020/month=2/part-00006-d2514daa-c589-4357-a932-224eb3716069.c000.snappy.parquet	
-rw-r--r-- 1 root supergroup 20.1 M 2022-06-16 16:52 /nycdata/year=2020/month=2/part-00007-d2514daa-c589-4357-a932-224eb3716069.c000.snappy.parquet	
-rw-r--r-- 1 root supergroup 20.2 M 2022-06-16 16:53 /nycdata/year=2020/month=2/part-00008-d2514daa-c589-4357-a932-224eb3716069.c000.snappy.parquet	
-rw-r--r-- 1 root supergroup 6.4 M 2022-06-16 16:53 /nycdata/year=2020/month=2/part-00009-d2514daa-c589-4357-a932-224eb3716069.c000.snappy.parquet	

%spark trips_DF.count()	READY
res10: Long = 12704362	

%spark trips_clean_DF.count()	READY
res11: Long = 9103468	

%sh hdfs dfs -get /nycdata/year=2020/month=1/part-00000-d2514daa-c589-4357-a932-224eb3716069.c000.snappy.parquet /home/cloudera/Downloads/102.parquet	READY
--	-------

%sh parquet-tools cat --json /home/cloudera/Downloads/102.parquet	READY
{\"VendorID\":1,\"tpep_pickup_datetime\":\"AHakK7wbAADihCUA\",\"tpep_dropoff_datetime\":\"ALbHOf8bAADihCUA\",\"passenger_count\":1,\"trip_distance\":1.2,\"RatecodeID\":1,\"store_and_fwd_flag\":\"N\",\"PULocationID\":238,\"DOLocationID\":239,\"payment_type\":1,\"fare_amount\":6.0,\"extra\":3.0,\"mta_tax\":0.5,\"tip_amount\":1.47,\"tolls_amount\":0.0,\"improvement_surcharge\":0.3,\"total_amount\":11.27,\"congestion_surcharge\":2.5}	
{\"VendorID\":1,\"tpep_pickup_datetime\":\"AM4ajCMcAADihCUA\",\"tpep_dropoff_datetime\":\"APArKIscaAADihCUA\",\"passenger_count\":1,\"trip_distance\":1.2,\"RatecodeID\":1,\"store_and_fwd_flag\":\"N\",\"PULocationID\":239,\"DOLocationID\":238,\"payment_type\":1,\"fare_amount\":7.0,\"extra\":3.0,\"mta_tax\":0.5,\"tip_amount\":1.5,\"tolls_amount\":0.0,\"improvement_surcharge\":0.3,\"total_amount\":12.3,\"congestion_surcharge\":2.5}	
{\"VendorID\":1,\"tpep_pickup_datetime\":\"AJa/bDQdAADihCUA\",\"tpep_dropoff_datetime\":\"AB6mDCaEaAADihCUA\",\"passenger_count\":2,\"trip_distance\":2.4,\"RatecodeID\":1,\"store_and_fwd_flag\":\"N\",\"PULocationID\":246,\"DOLocationID\":79,\"payment_type\":1,\"fare_amount\":12.0,\"extra\":3.0,\"mta_tax\":0.5,\"tip_amount\":1.75,\"tolls_amount\":0.0,\"improvement_surcharge\":0.3,\"total_amount\":17.55,\"congestion_surcharge\":2.5}	
{\"VendorID\":1,\"tpep_pickup_datetime\":\"AI6zHkYdAADihCUA\",\"tpep_dropoff_datetime\":\"AJDyUqceAADihCUA\",\"passenger_count\":1,\"trip_distance\":3.3,\"RatecodeID\":1,\"store_and_fwd_flag\":\"N\",\"PULocationID\":161,\"DOLocationID\":144,\"payment_type\":1,\"fare_amount\":17.0,\"extra\":3.0,\"mta_tax\":0.5,\"tip_amount\":4.15,\"tolls_amount\":0.0,\"improvement_surcharge\":0.3,\"total_amount\":24.95,\"congestion_surcharge\":2.5}	
{\"VendorID\":2,\"tpep_pickup_datetime\":\"ANRFdmMbAADihCUA\",\"tpep_dropoff_datetime\":\"AL4J7bEbAADihCUA\",\"passenger_count\":1,\"trip_distance\":1.07,\"RatecodeID\":1,\"store_and_fwd_flag\":\"N\",\"PULocationID\":43,\"DOLocationID\":239,\"payment_type\":1,\"fare_amount\":6.0,\"extra\":0.5,\"mta_tax\":0.5,\"tip_amount\":1.96,\"tolls_amount\":0.0,\"improvement_surcharge\":0.3,\"total_amount\":11.76,\"congestion_surcharge\":2.5}	
{\"VendorID\":2,\"tpep_pickup_datetime\":\"ANr2m0QcAADihCUA\",\"tpep_dropoff_datetime\":\"AFPeJk4eAADihCUA\",\"passenger_count\":1,\"trip_distance\":7.76,\"RatecodeID\":1,\"store_and_fwd_flag\":\"N\",\"PULocationID\":143,\"DOLocationID\":25,\"payment_type\":1,\"fare_amount\":28.5,\"extra\":0.5,\"mta_tax\":0.5,\"tip_amount\":4.84,\"tolls_amount\":0.0,\"improvement_surcharge\":0.3,\"total_amount\":34.1,\"congestion_surcharge\":2.5}	
Paragraph received a SIGTERM ExitValue: 143	

%sh parquet-tools head -n 2 /home/cloudera/Downloads/102.parquet	READY
VendorID = 1 tpep_pickup_datetime = AHakK7wbAADihCUA tpep_dropoff_datetime = ALbHOf8bAADihCUA passenger_count = 1 trip_distance = 1.2 RatecodeID = 1 store_and_fwd_flag = N PULocationID = 238 DOLocationID = 239 payment_type = 1 fare_amount = 6.0 extra = 3.0 mta_tax = 0.5 tip_amount = 1.47 tolls_amount = 0.0 improvement_surcharge = 0.3 total_amount = 11.27 congestion_surcharge = 2.5	

%sh parquet-tools schema /home/cloudera/Downloads/102.parquet	READY
message spark_schema { optional int32 VendorID; optional int96 tpep_pickup_datetime; optional int96 tpep_dropoff_datetime; optional int32 passenger_count; optional double trip_distance; optional int32 RatecodeID; optional binary store_and_fwd_flag (UTF8); optional int32 PULocationID; optional int32 DOLocationID; optional int32 payment_type; optional double fare_amount; optional double extra; optional double mta_tax; optional double tip_amount; optional double tolls_amount; optional double improvement_surcharge; optional double total_amount;	

%spark	READY
val lookup_DF = sqlContext.read.format("csv") .option("header","true") .option("inferSchema","true") .load("/nyc taxi/lookup/taxi+_zone_lookup.csv") lookup_DF.printSchema()	

root

|-- LocationID: integer (nullable = true)

|-- Borough: string (nullable = true)

|-- Zone: string (nullable = true)

|-- service_zone: string (nullable = true)

Spark Scala HDFS Hive and Impala

LocationID	Borough	Zone	service_zone
1	EWB	Newark Airport	EWB
2	Queens	Jamaica Bay	Boro Zone
3	Bronx	Allerton/Pelham G...	Boro Zone
4	Manhattan	Alphabet City	Yellow Zone
5	Staten Island	Arden Heights	Boro Zone

only showing top 5 rows

```
spark DFs are created spark.sql DataFrames LocationID: int, Borough: string, Zone: string, service_zone: string
```

```
%spark
lookup_DF.write
  .mode("overwrite")
  .parquet("/nyclookup")
```

```
%sh
hdfs dfs -ls -R -h /nyclookup/

-rw-r--r--   1 root supergroup          0 2022-06-17 12:17 /nyclookup/_SUCCESS
-rw-r--r--   1 root supergroup    5.6 K 2022-06-17 12:17 /nyclookup/part-00000-2345d162-0606-4b62-b43a-59233588d9d6-c000.snappy.parquet
```

```
%hive
SET hive.execution.engine = spark
```

Query executed successfully. Affected rows : -1

```
%hive
CREATE DATABASE rides
```

```
%hive
USE rides
```

Query executed successfully. Affected rows : -1

```
%hive
drop table nycTaxi
```

Query executed successfully. Affected rows : -1

```
%hive
drop table taxiLookup
```

Query executed successfully. Affected rows : -1

```
%hive
create external table nycTaxi(
  VendorID int,
  tpep_pickup_datetime timestamp,
  tpep_dropoff_datetime timestamp,
  passenger_count int,
  trip_distance double,
  RatecodeID int,
  store_and_fwd_flag string,
  PULocationID int,
  DOLocationID int,
  payment_type int,
  fare_amount double,
  extra double,
  mta_tax double,
  tip_amount double,
  tolls_amount double,
  improvement_surcharge double,
  total_amount double,
  congestion_surcharge double)
partitioned by (year int, month int)
stored as parquet
location '/nycdata/'
TBLPROPERTIES ("parquet.compression"="SNAPPY")
```

Query executed successfully. Affected rows : -1

```
%hive
create external table taxiLookup(
  LocationID int,
  Borough string,
  Zone string,
  service_zone string)
stored as parquet
location '/nyclookup/'
TBLPROPERTIES ("parquet.compression"="SNAPPY")
```

Query executed successfully. Affected rows : -1

```
%hive
MSCK REPAIR TABLE nycTaxi
```

Query executed successfully. Affected rows : -1

%hive

READY

MSCK REPAIR TABLE taxiLookup

MSCK REPAIR TABLE taxiLookup

Spark Scala HDFS Hive and Impala

%hive

READY

SELECT * FROM nycTaxi LIMIT 10

settings

nyctaxi.vendorid	nyctaxi.year	nyctaxi.month	nyctaxi.tpep_pickup_datetime	nyctaxi.tpep_dropoff_datetime	nyctaxi.passenger_count
1	2020	1	2020-01-01 00:28:15.0	2020-01-01 00:33:03.0	1
1	2020	1	2020-01-01 00:35:39.0	2020-01-01 00:43:04.0	1
1	2020	1	2020-01-01 00:55:11.0	2020-01-01 01:12:03.0	2
1	2020	1	2020-01-01 00:56:27.0	2020-01-01 01:21:44.0	1
1	2020	1	2020-01-01 00:15:35.0	2020-01-01 00:27:06.0	3
1	2020	1	2020-01-01 00:56:38.0	2020-01-01 01:13:34.0	1
1	2020	1	2020-01-01 00:28:53.0	2020-01-01 00:41:59.0	1
2	2020	1	2020-01-01 00:21:54.0	2020-01-01 00:27:31.0	1

%hive

READY

SELECT * FROM taxiLookup LIMIT 10

settings

taxilookup.locationid	taxilookup.borough	taxilookup.zone	taxilookup.service_zone
1	EWB	Newark Airport	EWB
2	Queens	Jamaica Bay	Boro Zone
3	Bronx	Allerton/Pelham Gardens	Boro Zone
4	Manhattan	Alphabet City	Yellow Zone
5	Staten Island	Arden Heights	Boro Zone
6	Staten Island	Arrochar/Fort Wadsworth	Boro Zone
7	Queens	Astoria	Boro Zone
8	Queens	Astoria Park	Boro Zone

%hive

READY

SELECT count(*) FROM nycTaxi

settings

_c0
9103468

%hive

READY

SELECT count(*) FROM taxiLookup

settings

_c0
265

%hive

READY

ANALYZE TABLE nycTaxi PARTITION (year, month) COMPUTE STATISTICS

Query executed successfully. Affected rows : -1

Spark Scala HDFS Hive and Impala

READY

%hive

ANALYZE TABLE taxiLookup COMPUTE STATISTICS

Query executed successfully. Affected rows : -1

%hive

SET hive.execution.engine = mr

Query executed successfully. Affected rows : -1

%impala

invalidate metadata;

Query executed successfully. Affected rows : -1

%impala

COMPUTE STATS nycTaxi;

settings ▼

summary

Updated 2 partition(s) and 18 column(s).

%impala

COMPUTE STATS taxiLookup;

settings ▼

summary

Updated 1 partition(s) and 4 column(s).

%impala

DESCRIBE FORMATTED nycTaxi;

settings ▼

name	type	comment
# col_name	data_type	comment
	null	null
vendorid	int	null
tppep_pickup_datetime	timestamp	null
tppep_dropoff_datetime	timestamp	null
passenger_count	int	null
trip_distance	double	null
ratecodeid	int	null

%impala

DESCRIBE FORMATTED taxiLookup;

settings ▼

name	type	comment
# col_name	data_type	comment
	null	null

locationid	int	null
borough	string	null
parking_id	string	null
service_zone	string	null
	null	null

Spark Scala HDFS Hive and Impala

%hive

set hive.exec.dynamic.partition = true

Query executed successfully. Affected rows : -1

READY

%hive

set hive.exec.dynamic.partition.mode = nonstrict

Query executed successfully. Affected rows : -1

READY

%hive

set hive.enforce.bucketing = true

Query executed successfully. Affected rows : -1

READY

%hive

set hive.enforce.sorting=true

Query executed successfully. Affected rows : -1

READY

%hive

drop table nycTaxi_part_bkt

Query executed successfully. Affected rows : -1

READY

%hive

create table nycTaxi_part_bkt(
VendorID int,
tpep_pickup_datetime timestamp,
tpep_dropoff_datetime timestamp,
passenger_count int,
trip_distance double,
RatecodeID int,
store_and_fwd_flag string,
PULocationID int,
DOLocationID int,
payment_type int,
fare_amount double,
extra double,
mta_tax double,
tip_amount double,
tolls_amount double,
improvement_surcharge double,
total_amount double,
congestion_surcharge double)
partitioned by (year int, month int)
clustered by (VendorID) sorted by (passenger_count) into 2 buckets

Query executed successfully. Affected rows : -1

READY

%hive

INSERT OVERWRITE TABLE nycTaxi_part_bkt partition (year, month) SELECT
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,
year,
month
FROM nycTaxi

Query executed successfully. Affected rows : -1

READY

%hive

SELECT * FROM nycTaxi_part_bkt LIMIT 10

READY

settings

nyctaxi_part_bkt.vendorid	nyctaxi_part_bkt.tpep_pickup_datetime	nyctaxi_part_bkt.tpep_dropoff_datetime	nyctaxi_part_bkt.passenger_count
2	2020-01-15 01:44:23.0	2020-01-15 02:00:40.0	1
2	2020-01-19 10:57:06.0	2020-01-19 11:14:49.0	1
2	2020-01-19 10:41:32.0	2020-01-19 10:50:35.0	1
2	2020-01-19 10:48:34.0	2020-01-19 11:01:05.0	1
2	2020-01-19 10:33:05.0	2020-01-19 10:45:51.0	1
2	2020-01-19 10:04:28.0	2020-01-19 10:13:21.0	1

2

2020-01-19 10:58:33 0

2020-01-19 11:06:49 0

1

Spark Scala HDFS Hive and Impala

%hive

READY

SELECT count(*) FROM nycTaxi_part_bkt

settings ▼

_c0

9103468

%hive

READY

ANALYZE TABLE nycTaxi_part_bkt PARTITION (year, month) COMPUTE STATISTICS

Query executed successfully. Affected rows : -1

%hive

READY

DESCRIBE FORMATTED nycTaxi_part_bkt

settings ▼

col_name	data_type	comment
# col_name	data_type	comment
	null	null
vendorid	int	
tppe_pickup_datetime	timestamp	
tppe_dropoff_datetime	timestamp	
passenger_count	int	
trip_distance	double	
ratecodeid	int	

%sh

READY

hdfs dfs -ls -R -h /user/hive/warehouse/rides.db/

drwxrwxrwx - hive supergroup 0 2022-06-17 15:05 /user/hive/warehouse/rides.db/nyctaxi_part_bkt
drwxrwxrwx - hive supergroup 0 2022-06-17 15:05 /user/hive/warehouse/rides.db/nyctaxi_part_bkt/year=2020
drwxr-xr-x - hive supergroup 0 2022-06-17 15:05 /user/hive/warehouse/rides.db/nyctaxi_part_bkt/year=2020/month=1
-rwxr-xr-x 1 hive supergroup 291.5 M 2022-06-17 15:05 /user/hive/warehouse/rides.db/nyctaxi_part_bkt/year=2020/month=1/000000_0
-rwxr-xr-x 1 hive supergroup 133.5 M 2022-06-17 15:03 /user/hive/warehouse/rides.db/nyctaxi_part_bkt/year=2020/month=1/000001_0
drwxrwxrwx - hive supergroup 0 2022-06-17 15:05 /user/hive/warehouse/rides.db/nyctaxi_part_bkt/year=2020/month=2
-rwxrwxrwx 1 hive supergroup 287.2 M 2022-06-17 15:05 /user/hive/warehouse/rides.db/nyctaxi_part_bkt/year=2020/month=2/000000_0
-rwxrwxrwx 1 hive supergroup 130.2 M 2022-06-17 15:03 /user/hive/warehouse/rides.db/nyctaxi_part_bkt/year=2020/month=2/000001_0

%hive

READY

set hive.exec.dynamic.partition = false

Query executed successfully. Affected rows : -1

%hive

READY

set hive.exec.dynamic.partition.mode = strict

Query executed successfully. Affected rows : -1

%hive

READY

set hive.enforce.bucketing = false

Query executed successfully. Affected rows : -1

%hive

READY

set hive.enforce.sorting=false

Query executed successfully. Affected rows : -1

%impala

READY

SELECT month, avg(fare_amount) as average_fare_amount FROM nycTaxi_part_bkt
GROUP BY month
ORDER BY month ASC;

settings ▼

month	average_fare_amount
1	14.674999810592206
Spark Scala HDFS Hive and Impala	14.671802310937645

%impala

READY

SELECT passenger_count, count(*) as total_trips FROM nycTaxi_part_bkt
GROUP BY passenger_count
ORDER BY total_trips DESC;

settings

passenger_count	total_trips
1	6626157
2	1403205
3	368159
5	331514
6	194577
4	179816
7	20
8	13

%impala

READY

SELECT payment_type, count(*) as total_trips FROM nycTaxi_part_bkt
GROUP BY payment_type
ORDER BY total_trips DESC;

settings ▼

payment_type ▼	total_trips	⋮
1	6953636	
2	2110961	
3	28180	
4	10691	

%impala

READY

SELECT zone, min(trip_distance) as min_trip_distance, max(trip_distance) as max_trip_distance FROM nycTaxi_part_bkt
JOIN taxiLookup on (PULocationID = LocationID)
GROUP BY zone
ORDER BY max_trip_distance DESC
LIMIT 10;

settings ▼

zone ▼	min_trip_distance ▼	max_trip_distance	≡
Clinton East	1	369.94	
JFK Airport	1	262.88	
Lenox Hill East	1	259.22	
Midtown Center	1	211.7	
Schuylerville/Edgewater Park	1	207.11	
Flushing	1	168.4	
East Elmhurst	1	166.07	
Penn Station/Madison Sq West	1	164.2	

```
%impala
SELECT month, avg(trip_distance/(UNIX_TIMESTAMP(tpep_dropoff_datetime)-UNIX_TIMESTAMP(tpep_pickup_datetime))) as speed FROM nycTaxi_part_bkt
WHERE UNIX_TIMESTAMP(tpep_dropoff_datetime)<>UNIX_TIMESTAMP(tpep_pickup_datetime)
GROUP BY month
ORDER BY speed DESC;
```

READY

settings ▼

month ▼	speed
1	0.003939802894676546
2	0.003799794109330672

Spark Scala HDFS Hive and Impala

%impala	READY
---------	-------