In this assignment, you will practice writing MapReduce codes in a 4.5GB dataset. This session contains the following segments-

* Dataset Description
* Problem Statement & Sample Dataset
* Steps for Downloading the Data on CDH
* Sample Output
* Clean-Up after the Assignment

# **Dataset Description**

The dataset which will be used for this assignment is the TLC trip record data for Yellow taxis. The fields present in the data and their meaning is given below for your reference:

**Please note that the input data is for the months July 2017 to Dec 2017 and total size is ~4.5 GB.**

| **Field Name** | |  | **Description** | | --- | --- | |
| --- | --- | --- | --- |
| VendorID | A code indicating the TPEP provider that provided the record. 1= Creative Mobile Technologies, LLC; 2= VeriFone Inc. |
| tpep\_pickup\_datetime | The date and time when the meter was engaged |
| tpep\_dropoff\_datetime | The date and time when the meter was disengaged |
| passenger\_count | The number of passengers in the vehicle. This is a driver-entered value |
| trip\_distance | The elapsed trip distance in miles reported by the taximeter |
| RatecodeID | |  | The final rate code in effect at the end of the trip.  1= Standard rate  2=JFK  3=Newark  4=Nassau or Westchester  5=Negotiated fare 6=Group ride | | --- | --- | |
| store\_and\_fwd\_flag | This flag indicates whether the trip record was held in vehicle memory before sending to the vendor, aka “store and forward,” because the vehicle did not have a connection to the server.  Y= store and forward trip  N= not a store and forward trip |
| PULocationID | The ID of the location from where the passenger was picked. |
| DOLocationID | |  | The ID of the location from where the passenger was dropped. | | --- | --- | |
| payment\_type | A numeric code signifying how the passenger paid for the trip.  1= Credit card  2= Cash  3= No charge  4= Dispute  5= Unknown  6= Voided trip |
| fare\_amount | The time-and-distance fare calculated by the meter. |
| extra | |  | Miscellaneous extras and surcharges. Currently, this only includes the $0.50 and $1 rush hour and overnight charges | | --- | --- | |
| mta\_tax | |  | $0.50 MTA tax that is automatically triggered based on the metered rate in use. | | --- | --- | |
| tip\_amount | This field is automatically populated for credit card tips. Cash tips are not included. |
| tolls\_amount | Total amount of all tolls paid in trip . |
| | improvement\_surcharge |  | | --- | --- | | $0.30 improvement surcharge assessed trips at the flag drop. The improvement surcharge began being levied in 2015. |
| total\_amount | The total amount charged to passengers. Does not include cash tips. |

# **Problem Statement & Sample Dataset**

The assignment contains three use cases i.e. single record lookup, filter and GroupBy accompanied by count. The respective problem statements are mentioned below.

1. Fetch the record having VendorID as '2' AND tpep\_pickup\_datetime as '2017-10-01 00:15:30' AND tpep\_dropoff\_datetime as '2017-10-01 00:25:11' AND passenger\_count as '1' AND trip\_distance as '2.17'
2. Filter all the records having RatecodeID as 4.
3. Group By all the records based on payment type and find the count for each group.

**Hint**:

* For the Problem-01 & Problem-02, you are not required to write the reducer program.

You can use the following sample dataset for your code debugging in the local environment.

You can use the following sample dataset for your code debugging in the local environment.

<https://drive.google.com/file/d/1uizSyw7FfXKVvNLWCinxsz_2nLjZdEqY/view?usp=share_link>

Once you are done with the above 3 problems, please feel free to explore the dataset and practice MapReduce programming on your own. :)

# **Sample Output**

## **\*\*\* Sample Output \*\*\***

**Note**: You may have a different format for the below output.

## **Problem-01: Single Row Lookup.**

**Reference Command**-

hadoop fs -cat /user/hadoop/mapr\_assignment/output/q1/part\*

**Output**-

2,2017-10-01 00:15:30,2017-10-01 00:25:11,1,2.17,1,N,141,142,1,9,0.5,0.5,2.06,0,0.3,12.36

## **Problem-02: Filter Specific Rows.**

In this problem, you can only verify the number of rows in the final output.

**Reference Command-**

hadoop fs -cat /user/hadoop/mapr\_assignment/output/q2/part\* | wc -l

**Output**-

31029

## **Problem-03: Group by and Count**

**Reference Command-**

hadoop fs -cat /user/hadoop/mapr\_assignment/output/q3/part\*

**Output**-

4,85287

3,292090

2,17648209

1,36492406