1. Create Required directories in HDFS using hdfs-linux commands and Load the data into HDFS.

For data file,

hadoop fs -mkdir /Project/TaxiData/

For External Hive table:

hadoop fs -mkdir /Project/hive-tables/

For Moving Batch file from Local to HDFS,

hdfs dfs -get /TaxiRides/TaxiRides/ /home/cloudera/Desktop

1. Create Staging and Actual tables in MySQL.

|  |  |
| --- | --- |
| Staging table | Actual table |
| create table if not exists TaxiRides\_S (  VendorID bigint,  tpep\_pickup\_datetime VARCHAR(30),  tpep\_dropoff\_datetime VARCHAR(30),  passenger\_count double,  trip\_distance double,  RatecodeID double,  PULocationID bigint,  DOLocationID bigint,  payment\_type bigint,  fare\_amount double,  extra double,  mta\_tax double,  tip\_amount double,  tolls\_amount double,  improvement\_surcharge double,  total\_amount double,  congestion\_surcharge double,  airport\_fee double  ); | create table if not exists TaxiRides(  VendorID bigint,  tpep\_pickup\_datetime VARCHAR(30),  tpep\_dropoff\_datetime VARCHAR(30),  passenger\_count double,  trip\_distance double,  RatecodeID double,  PULocationID bigint,  DOLocationID bigint,  payment\_type bigint,  fare\_amount double,  extra double,  mta\_tax double,  tip\_amount double,  tolls\_amount double,  improvement\_surcharge double,  total\_amount double,  congestion\_surcharge double,  airport\_fee double  ); |
| To check the data has been entered,  select sNo, VendorID, tpep\_pickup\_datetime, passenger\_count from TaxiRides\_S limit 20; | To check the data has been entered,  select sNo, VendorID, tpep\_pickup\_datetime, passenger\_count from TaxiRides\_S limit 20; |

1. Load the data into MySQL tables using Sqoop Export.

|  |  |
| --- | --- |
| Staging table | Actual table |
| sqoop export \  --connect "jdbc:mysql://quickstart.cloudera:3306/taxi\_rides" \  --username root \  --password cloudera \  --table TaxiRides\_S \  --export-dir /Project/TaxiData/ride\_data\_2022\_01\_01.csv \  --fields-terminated-by ',' | sqoop export \  --connect "jdbc:mysql://quickstart.cloudera:3306/taxi\_rides" \  --username root \  --password cloudera \  --table TaxiRides \  --export-dir /Project/TaxiData/ride\_data\_2022\_01\_01.csv \  --fields-terminated-by ',' |

1. Create Sqoop jobs to import data from MySQL to HDFS.

|  |
| --- |
| sqoop job \  --create job\_taxiDatas \  -- import \  --connect "jdbc:mysql://quickstart.cloudera:3306/taxi\_rides" \  --username root \  --P \  --table TaxiRides\_S \  --warehouse-dir /user/cloudera/Project/hive-tables/imported\_data \  --incremental append \  --check-column sNo \  --last-value 0 |

Execute Sqoop job

sqoop job –exec job\_taxiDatas

1. Create Hive External table

|  |
| --- |
| create external table if not exists rideData(  VendorID bigint,  tpep\_pickup\_datetime VARCHAR(30),  tpep\_dropoff\_datetime VARCHAR(30),  passenger\_count double,  trip\_distance double,  RatecodeID double,  store\_and\_fwd\_flag VARCHAR(15),  PULocationID bigint,  DOLocationID bigint,  payment\_type bigint,  fare\_amount double,  extra double,  mta\_tax double,  tip\_amount double,  tolls\_amount double,  improvement\_surcharge double,  total\_amount double,  congestion\_surcharge double,  airport\_fee double  )  row format delimited fields terminated by ','  location '/HiveTable/data/'; |

1. HDFS cmds for loading data into Hive tables;

|  |  |
| --- | --- |
| If data saved in **HDFS** | If data stored in **local** |
| load data inpath  "/Project/TaxiData/TaxiDataAnalysis/TaxiDataAnalysis/RideData.csv"  into table rideData; | load data local inpath  "/home/cloudera/Desktop/TaxiDataAnalysis/RideData.csv"  into table rideData; |

1. Creating a shadow table for creating a data column for Partitioning.

|  |
| --- |
| create external table if not exists rideData\_s(  VendorID bigint,  date int,  tpep\_pickup\_datetime varchar(30),  tpep\_dropoff\_datetime varchar(30),  passenger\_count double,  trip\_distance double,  PULocationID bigint,  DOLocationID bigint,  payment\_type bigint,  total\_amount double  )  row format delimited fields terminated by ','  location '/HiveTable\_s/data/'; |

1. Inserting data into the shadow table

|  |
| --- |
| insert into table  rideData\_s  select  VendorID, date\_format(tpep\_pickup\_datetime,'dd') AS date,  tpep\_pickup\_datetime, tpep\_dropoff\_datetime, passenger\_count,  trip\_distance, PULocationID, DOLocationID, payment\_type,  total\_amount  from  ridedata; |

1. Create an optimized table

|  |
| --- |
| create external table if not exists rideData\_optimized(  VendorID bigint,  tpep\_pickup\_datetime timestamp,  tpep\_dropoff\_datetime timestamp,  passenger\_count double,  trip\_distance double,  PULocationID bigint,  DOLocationID bigint,  payment\_type bigint,  total\_amount double  )  PARTITIONED BY (date int)  STORED AS ORC  LOCATION '/Hive\_optimized/RideData/' |

1. Insert data into the optimized table

|  |
| --- |
| insert into table  rideData\_optimized  partition(date)  select  VendorID, tpep\_pickup\_datetime, tpep\_dropoff\_datetime, passenger\_count, trip\_distance, PULocationID, DOLocationID, payment\_type, total\_amount, date  from  ridedata\_s; |

Hive Properties;

SET hive.exec.dynamic.partition=true;

SET hive.exec.dynamic.partition.mode=nonstrict;