**ASSIGNMENT 5**

**Topics: HIVE + SQOOP + MYSQL**

**Dataset:** Flights\_Delay.csv

Dataset Description:

ID: Rows ID

YEAR: 2015

MONTH: 1-12

DAY: 1-31

DAY\_OF\_WEEK: 1 (Monday) - 7 (Sunday)

AIRLINE: Airline CODE

FLIGHT\_NUMBER: Flight Number

TAIL\_NUMBER: Flight’s tail number

ORIGIN\_AIRPORT: Origin IATA airport code

DESTINATION\_AIRPORT: Destination IATA airport code

SCHEDULED\_DEPARTURE: Actual departure time (local, hhmm)

DEPARTURE\_TIME: Scheduled departure time (local, hhmm)

DEPARTURE\_DELAY: Departure delay, in minutes

TAXI\_OUT: Taxi out time in minutes

WHEELS\_OFF:

SCHEDULED\_TIME: Scheduled arrival time (local, hhmm)

ELAPSED\_TIME: in Minutes

AIR\_TIME: in Minutes

DISTANCE: in Miles

WHEELS\_ON:

TAXI\_IN: Taxi in time, in minutes

SCHEDULED\_ARRIVAL: Scheduled arrival time (local, hhmm)

ARRIVAL\_TIME: Actual arrival time (local, hhmm)

ARRIVAL\_DELAY: Arrival delay, in minutes

DIVERTED: 1 = yes, 0 = no

CANCELLED: 1 = yes, 0 = no

1. **Create mysql table named flights\_table in database flightsDB**

create database if not exsts flightsDB;

use flightsDB;

create table if not exists flights\_table(

ID INT, FLY\_YEAR INT, FLY\_MONTH INT, FLY\_DAY INT, DAY\_OF\_WEEK INT, AIRLINE TEXT, FLIGHT\_NUMBER INT, TAIL\_NUMBER TEXT, ORIGIN\_AIRPORT TEXT, DESTINATION\_AIRPORT TEXT, SCHEDULED\_DEPARTURE TEXT, DEPARTURE\_TIME INT, DEPARTURE\_DELAY INT, TAXI\_OUT INT, WHEELS\_OFF INT, SCHEDULED\_TIME INT, ELAPSED\_TIME INT, AIR\_TIME INT, DISTANCE INT, WHEELS\_ON INT ,TAXI\_IN INT, SCHEDULED\_ARRIVAL INT, ARRIVAL\_TIME INT, ARRIVAL\_DELAY INT, DIVERTED INT, CANCELLED INT

);

1. **Load the data into flights\_table using files.**

load data local infile ‘/home/hadoop/flights\_delay.csv’

into table flights\_table

fields terminated by ‘,’

lines terminated by ‘\n’

ignore 1 lines;

1. **Move table data from mysql table to location /airlines using sqoop command.**

sqoop import –connect jdbc:mysql://localhost/flightsDB --username root --password hadoop123 --table flights\_table -m 1 --target-dir ‘/airlines/’

1. **Create a Hive external table “flights” in Database “airline\_delayDB”**

create database if not exists airline\_delayDB;

use airline\_delayDB;

create external table if not exists flights(

ID INT, FLY\_YEAR INT, FLY\_MONTH INT, FLY\_DAY INT, DAY\_OF\_WEEK INT, AIRLINE TEXT, FLIGHT\_NUMBER INT, TAIL\_NUMBER TEXT, ORIGIN\_AIRPORT TEXT, DESTINATION\_AIRPORT TEXT, SCHEDULED\_DEPARTURE TEXT, DEPARTURE\_TIME INT, DEPARTURE\_DELAY INT, TAXI\_OUT INT, WHEELS\_OFF INT, SCHEDULED\_TIME INT, ELAPSED\_TIME INT, AIR\_TIME INT, DISTANCE INT, WHEELS\_ON INT ,TAXI\_IN INT, SCHEDULED\_ARRIVAL INT, ARRIVAL\_TIME INT, ARRIVAL\_DELAY INT, DIVERTED INT, CANCELLED INT

)

row format delimited

fields terminated by ‘,’

lines terminated by ‘\n’

location ‘/airlines/’;

1. **Create another hive table as parquet table “parquet\_flights” & insert the data into this using “flights” external table**

create external table if not exists parquet\_flights(

ID INT, FLY\_YEAR INT, FLY\_MONTH INT, FLY\_DAY INT, DAY\_OF\_WEEK INT, AIRLINE TEXT, FLIGHT\_NUMBER INT, TAIL\_NUMBER TEXT, ORIGIN\_AIRPORT TEXT, DESTINATION\_AIRPORT TEXT, SCHEDULED\_DEPARTURE TEXT, DEPARTURE\_TIME INT, DEPARTURE\_DELAY INT, TAXI\_OUT INT, WHEELS\_OFF INT, SCHEDULED\_TIME INT, ELAPSED\_TIME INT, AIR\_TIME INT, DISTANCE INT, WHEELS\_ON INT ,TAXI\_IN INT, SCHEDULED\_ARRIVAL INT, ARRIVAL\_TIME INT, ARRIVAL\_DELAY INT, DIVERTED INT, CANCELLED INT

)

row format delimited

fields terminated by ‘,’

lines terminated by ‘\n’

stored as parquetfile;

insert into parquet\_flights

select \* from flights;

1. **Describe the table schema & show top 10 rows of Dataset**

describe parquet\_flights;

select \* from parquet\_flights limit 10;

Write Hive queries (HQL) to show following analysis

1. **Average arrival delay caused by airlines**

select avg(arrival\_delay) as Average\_Arrival\_Delay

from parquet\_flights;

1. **Days of months with respected to average of arrival delays**

select fly\_month, fly\_day, avg(arrival\_delay) as Average\_Arrival\_Delay

from parquet\_flights

group by fly\_month, fly\_day

order by fly\_month, fly\_day;

1. **Arrange weekdays with respect to the average arrival delays caused**

select day\_of\_week, avg(arrival\_delay) as Average\_Arrival\_Delay

from parquet\_flights

group by day\_of\_week

order by day\_of\_week;

1. **Arrange Days of month as per cancellations done in Descending**

select fly\_month, fly\_day, sum(cancelled) as Cancellations\_done

from parquet\_flights

group by fly\_month, fly\_day

order by Cancellations\_done desc;

1. **Finding busiest airports with respect to day of week**

select day\_of\_week, airport, total\_flights

from

(select \*, dense\_rank() over(partition by day\_of\_week order by total\_flights desc) as rank\_

from

(select t11.day\_of\_week, t11.origin\_airport as airport, total\_flights1+total\_flights2 as total\_flights from

(SELECT day\_of\_week, origin\_airport, COUNT(\*) AS total\_flights2

FROM parquet\_flights

GROUP BY day\_of\_week, origin\_airport) t11

join

(SELECT day\_of\_week, destination\_airport, COUNT(\*) AS total\_flights1

FROM parquet\_flights

GROUP BY day\_of\_week, destination\_airport) t12

on t11.origin\_airport = t12.destination\_airport and t11.day\_of\_week = t12.day\_of\_week) as t1) as t

where rank\_=1;

1. **Finding airlines that make the maximum number of cancellations**

select airline, sum(cancelled) as Cancellations\_done

from parquet\_flights

group by airline

order by Cancellations\_done desc;

1. **Find and order airlines in descending that make the most number of diversions**

select airline, sum(diverted) as Divertions\_done

from parquet\_flights

group by airline

order by Divertions\_done desc;

1. **Finding days of month that see the most number of diversion**

select fly\_month, fly\_day, sum(diverted) as Diversions

from parquet\_flights

group by fly\_month, fly\_day

order by diversions desc;

1. **Calculating mean and standard deviation of departure delay for all flights in minutes**

select avg(departure\_delay) avg\_delay, std(departure\_delay) as standard\_deviation

from parquet\_flights

where departure\_delay>0;

1. **Calculating mean and standard deviation of arrival delay for all flights in minutes**

select avg(arrival\_delay) as avg\_delay, std(arrival\_delay) as standard\_deviation

from parquet\_flights

where arrival\_delay>0;

1. **Create a partitioning table “flights\_partition” using partitioned by schema “CANCELLED”**

set hive.exec.dynamic.partition=true;

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

create external table if not exists flights\_partition(

ID INT, FLY\_YEAR INT, FLY\_MONTH INT, FLY\_DAY INT, DAY\_OF\_WEEK INT, AIRLINE TEXT, FLIGHT\_NUMBER INT, TAIL\_NUMBER TEXT, ORIGIN\_AIRPORT TEXT, DESTINATION\_AIRPORT TEXT, SCHEDULED\_DEPARTURE TEXT, DEPARTURE\_TIME INT, DEPARTURE\_DELAY INT, TAXI\_OUT INT, WHEELS\_OFF INT, SCHEDULED\_TIME INT, ELAPSED\_TIME INT, AIR\_TIME INT, DISTANCE INT, WHEELS\_ON INT ,TAXI\_IN INT, SCHEDULED\_ARRIVAL INT, ARRIVAL\_TIME INT, ARRIVAL\_DELAY INT, DIVERTED INT)

partitioned by (CANCELLED INT)

row format delimited

fields terminated by ‘,’

lines terminated by ’\n’;

insert overwrite table flights\_partition

partition(CANCELLED)

select \* from parquet\_flights;

1. **Create Bucketing table “Flights\_Bucket” using clustered by MONTH into 3 Buckets**

set hive.enforce.bucketing=true;

create external table if not exists flights\_bucket(

ID INT, FLY\_YEAR INT, FLY\_MONTH INT, FLY\_DAY INT, DAY\_OF\_WEEK INT, AIRLINE TEXT, FLIGHT\_NUMBER INT, TAIL\_NUMBER TEXT, ORIGIN\_AIRPORT TEXT, DESTINATION\_AIRPORT TEXT, SCHEDULED\_DEPARTURE TEXT, DEPARTURE\_TIME INT, DEPARTURE\_DELAY INT, TAXI\_OUT INT, WHEELS\_OFF INT, SCHEDULED\_TIME INT, ELAPSED\_TIME INT, AIR\_TIME INT, DISTANCE INT, WHEELS\_ON INT ,TAXI\_IN INT, SCHEDULED\_ARRIVAL INT, ARRIVAL\_TIME INT, ARRIVAL\_DELAY INT, DIVERTED INT, CANCELLED INT)

clustered by (FLY\_MONTH) into 3 buckets

row format delimited

fields terminated by ‘,’

lines terminated by ’\n’;

insert overwrite table flights\_bucket

select \* from parquet\_flights;

1. **Get count of data of each bucket.**

select fly\_month, count(\*) as count

from flights\_bucket

group by fly\_month;

1. **Finding all diverted Route from a source to destination Airport & which route is the most diverted**

select origin\_airport, destination\_airport, sum(diverted) as Divertions

from parquet\_flights

group by origin\_airport, destination\_airport

order by Divertions desc;

1. **Finding AIRLINES with its total flight count, total number of flights arrival delayed by more than 30 Minutes, % of such flights delayed by more than 30 minutes when it is not Weekends with minimum count of flights from Airlines by more than 10. Also Exclude some of Airlines 'AK', 'HI', 'PR', 'VI' and arrange output in descending order by % of such count of flights.**

select t1.airline, tot\_count, arrival\_delay\_greater\_than\_30mins, arrival\_delay\_greater\_than\_30mins\_wd,

arrival\_delay\_greater\_than\_30mins\_wd\*100/arrival\_delay\_greater\_than\_30mins as arrival\_delay\_greater\_than\_30mins\_weekdays\_perc

from (select airline, count(\*) as tot\_count from parquet\_flights

group by airline) t1

join (select airline, count(\*) as arrival\_delay\_greater\_than\_30mins from parquet\_flights where arrival\_delay>30

group by airline) t2 on t1.airline=t2.airline

join (select airline, count(\*) as arrival\_delay\_greater\_than\_30mins\_wd from parquet\_flights where arrival\_delay>30 and day\_of\_week in (1,2,3,4,5)

group by airline having arrival\_delay\_greater\_than\_30mins\_wd>10) t3 on t2.airline=t3.airline

where t1.airline not in ('AK', 'HI', 'PR', 'VI')

order by arrival\_delay\_greater\_than\_30mins\_weekdays\_perc desc;

1. **Finding AIRLINES with its total flight count with total number of flights departure delayed by less than 30 Minutes, % of such flights delayed by less than 30 minutes when it is Weekends with minimum count of flights from Airlines by more than 10. Also Exclude some of Airlines 'AK', 'HI', 'PR', 'VI' and arrange output in descending order by % of such count of flights.**

select t1.airline, tot\_count, arrival\_delay\_lesser\_than\_30mins,

arrival\_delay\_lesser\_than\_30mins\_wknd\*100/arrival\_delay\_lesser\_than\_30mins as arrival\_delay\_lesser\_than\_30mins\_wknd\_perc

from (select airline, count(\*) as tot\_count from parquet\_flights

group by airline) t1

join (select airline, count(\*) as arrival\_delay\_lesser\_than\_30mins from parquet\_flights where arrival\_delay<30

group by airline) t2 on t1.airline=t2.airline

join (select airline, count(\*) as arrival\_delay\_lesser\_than\_30mins\_wknd from parquet\_flights where arrival\_delay<30 and day\_of\_week in (6,7)

group by airline having arrival\_delay\_lesser\_than\_30mins\_wknd>10) t3 on t2.airline=t3.airline

where t1.airline not in ('AK', 'HI', 'PR', 'VI')

order by arrival\_delay\_lesser\_than\_30mins\_wknd\_perc desc;

1. **When is the best time of day/day of week/time of a year to fly with minimum delays?**

select scheduled\_departure, day\_of\_week, fly\_day, fly\_month, avg((departure\_delay+arrival\_delay)/2) as average\_journey\_delay

from parquet\_flights

where arrival\_delay>=0 and departure\_delay>=0

group by scheduled\_departure, day\_of\_week, fly\_day, fly\_month

order by average\_journey\_delay, scheduled\_departure;