**Objective** :-

The assignment is meant for you to apply learnings of the module on Hive on a real-life dataset. One of the major objectives of this assignment is gaining familiarity with how an analysis works in Hive and how you can gain insights from large datasets.

**Problem Statement:-**

New York City is a thriving metropolis and just like most other cities of similar size, one of the biggest problems its residents face is parking. The classic combination of a huge number of cars and a cramped geography is the exact recipe that leads to many parking tickets.

**Download Dataset** - <https://data.cityofnewyork.us/browse?q=parking+tickets>

create table parking\_violations\_issued

(

Summons\_Number bigint,

Plate\_ID string,

Registration\_State string,

Plate\_Type string,

Issue\_Date string,

Violation\_Code int,

Vehicle\_Body\_Type string,

Vehicle\_Make string,

Issuing\_Agency string,

Street\_Code1 int,

Street\_Code2 int,

Street\_Code3 int,

Vehicle\_Expiration string,

Violation\_Location int,

Violation\_Precinct int,

Issuer\_Precinct int,

Issuer\_Code int,

Issuer\_Command string,

Issuer\_Squad string,

Violation\_Time string,

Time\_First\_Observed string,

Violation\_County string,

Violation\_In\_Front\_Of\_Or\_Opposite string,

House\_Number string,

Street\_Name string,

Intersecting\_Street string,

Date\_First\_Observed int,

Law\_Section int,

Sub\_Division string,

Violation\_Legal\_Code string,

Days\_Parking\_In\_Effect string,

From\_Hours\_In\_Effect string,

To\_Hours\_In\_Effect string,

Vehicle\_Color string,

Unregistered\_Vehicle int,

Vehicle\_Year string,

Meter\_Number string,

Feet\_From\_Curb int,

Violation\_Post\_Code string,

Violation\_Description string,

No\_Standing\_or\_Stopping\_Violation string,

Hydrant\_Violation string,

Double\_Parking\_Violation string)

row format delimited

fields terminated by ','

tblproperties ("skip.header.line.count" = "1");

hive>load data local inpath'file///home/cloudera/Parking\_Violations\_Issued\_-\_Fiscal\_Year\_2017.csv' into table parking\_violations\_issued;

* **Creating Partitions on Violation Country and Buckets on Summons\_Number:**

create table parking\_violations\_2017

(

Summons\_Number bigint,

Plate\_ID string,

Registration\_State string,

Plate\_Type string,

Issue\_Date int,

Violation\_Code int,

Vehicle\_Body\_Type string,

Vehicle\_Make string,

Issuing\_Agency string,

Street\_Code1 int,

Street\_Code2 int,

Street\_Code3 int,

Vehicle\_Expiration string,

Violation\_Location int,

Violation\_Precinct int,

Issuer\_Precinct int,

Issuer\_Code int,

Issuer\_Command string,

Issuer\_Squad string,

Violation\_Time string,

Time\_First\_Observed string,

Violation\_In\_Front\_of\_or\_Opposite string,

House\_Number string,

Street\_Name string,

Intersecting\_Street string,

Date\_First\_Observed int,

Law\_Section int,

Sub\_Division string,

Violation\_Legal\_Code string,

Days\_Parking\_In\_Effect string,

From\_Hours\_In\_Effect string,

To\_Hours\_In\_Effect string,

Vehicle\_Color string,

Unregistered\_Vehicle int,

Vehicle\_Year string,

Meter\_Number string,

Feet\_From\_Curb int,

Violation\_Post\_Code string,

Violation\_Description string,

No\_Standing\_or\_Stopping\_Violation string,

Hydrant\_Violation string,

Double\_Parking\_Violation string)

partitioned by (Violation\_County string)

CLUSTERED BY (Summons\_Number) sorted by (Summons\_Number) INTO 8 BUCKETS

row format delimited

fields terminated by ','

tblproperties ("skip.header.line.count" = "1");

* **Set few properties to enable bucketing and Dynamic partition**

hive>set hive.exec.dynamic.partition=true;

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

hive>set hive.enforce.bucketing = true;

* **loding data into parking\_violations\_2017**

insert overwrite table parking\_violations\_2017

partition(Violation\_County) Select

Summons\_Number,

Plate\_ID,

Registration\_State,

Plate\_Type,

unix\_timestamp(Issue\_Date,'MM/DD/YYYY'),

Violation\_Code,

Vehicle\_Body\_Type,

Vehicle\_Make,

Issuing\_Agency,

Street\_Code1,

Street\_Code2,

Street\_Code3,

Vehicle\_Expiration,

Violation\_Location,

Violation\_Precinct,

Issuer\_Precinct,

Issuer\_Code,

Issuer\_Command,

Issuer\_Squad,

Violation\_Time,

Time\_First\_Observed,

Violation\_In\_Front\_Of\_Or\_Opposite,

House\_Number,

Street\_Name,

Intersecting\_Street,

Date\_First\_Observed,

Law\_Section,

Sub\_Division,

Violation\_Legal\_Code,

Days\_Parking\_In\_Effect,

From\_Hours\_In\_Effect,

To\_Hours\_In\_Effect,

Vehicle\_Color,

Unregistered\_Vehicle,

Vehicle\_Year,

Meter\_Number,

Feet\_From\_Curb,

Violation\_Post\_Code,

Violation\_Description,

No\_Standing\_or\_Stopping\_Violation,

Hydrant\_Violation,

Double\_Parking\_Violation,

Violation\_County

from parking\_violations\_issued

where unix\_timestamp(Issue\_Date,'MM/DD/YYYY') between unix\_timestamp('01/01/2017','MM/DD/YYYY') and unix\_timestamp('12/31/2017','MM/DD/YYYY');

A picture containing text

Description automatically generated

**Part-I: Examine the data**

1. **Find the total number of tickets for the year**.

select count(distinct summons\_number) as No\_of\_Tickets from parking\_violations\_2017

asn: 204038

Text

Description automatically generated

* **Find out how many unique states the cars which got parking tickets came from.**

Hive> select count(distinct Registration\_State) as Registration\_states from parking\_violations\_2017;

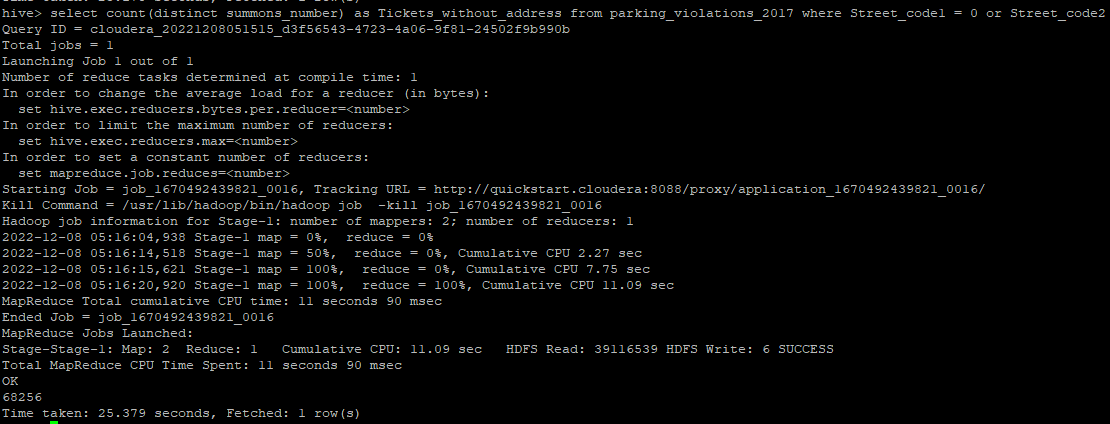
Total Number of Unique States are : 63

Text

Description automatically generated

* **Some parking tickets don’t have addresses on them, which is cause for concern. Find out how many such tickets there are(i.e. tickets where either "Street Code 1" or "Street Code 2" or "Street Code 3" is empty )**

select count(distinct summons\_number) as Tickets\_without\_address from parking\_violations\_2017 where Street\_code1 = 0 or Street\_code2 = 0 or Street\_code3 = 0;

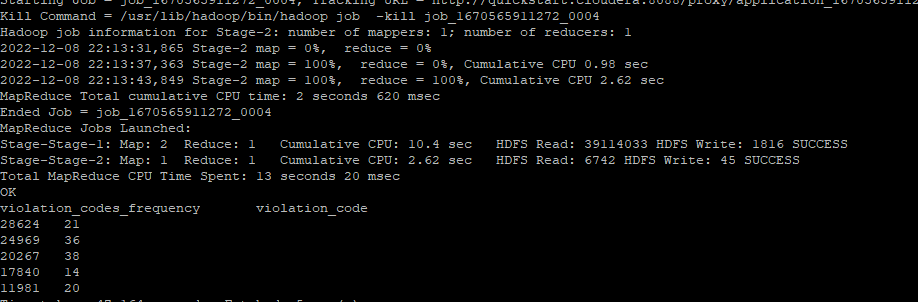


Ans : 68256

Part-II: Aggregation tasks

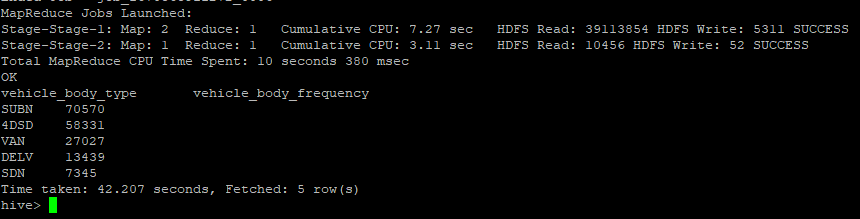
1. **How often does each violation code occur? (frequency of violation codes - find the top 5)**

Hive> select count(Violation\_Code) as violation\_codes\_frequency,Violation\_Code from parking\_violations\_2017 group by Violation\_Code order by violation\_codes\_frequency desc limit 5;

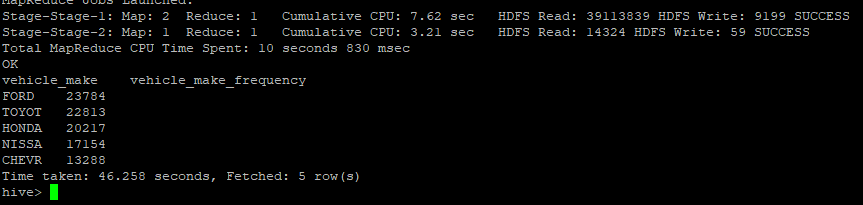


**2.) How often does each vehicle body type get a parking ticket? How about the vehicle make? (find the top 5 for both)**

Hive> select Vehicle\_Body\_Type,count(summons\_number)as Vehicle\_body\_frequency from parking\_violations\_2017 group by Vehicle\_Body\_Type order by Vehicle\_body\_frequency desc limit 5;



Hive> select Vehicle\_make,count(summons\_number)as vehicle\_make\_frequency from parking\_violations\_2017 group by Vehicle\_make order by vehicle\_make\_frequency desc limit 5;



**3.) A precinct is a police station that has a certain zone of the city under its command. Find the (5 highest) frequencies of:**

**a.) Violating Precincts (this is the precinct of the zone where the violation occurred)**

hive> select Violation\_Precinct ,count(Violation\_Precinct) as violation\_precinct\_frequency from parking\_violations\_2017 group by Violation\_Precinct order by violation\_precinct\_frequency desc limit 5; Text

Description automatically generated

**b.) Issuer Precincts (this is the precinct that issued the ticket)**

hive> select issuer\_Precinct ,count(issuer\_Precinct) as issuer\_precinct\_frequency from parking\_violations\_2017 group by issuer\_Precinct order by issuer\_precinct\_frequency desc limit 5;

A screenshot of a computer

Description automatically generated with medium confidence

**4.) Find the violation code frequency across 3 precincts which have issued the most number of tickets - do these precinct zones have an exceptionally high frequency of certain violation codes?**

Hive> select Issuer\_Precinct,Violation\_Code, count(\*) as Violation\_code\_frequency from parking\_violations\_2017 group by Issuer\_Precinct, Violation\_Code order by Violation\_code\_frequency desc limit 10;

Text

Description automatically generated

**18,19,14 are the three issuer precincts which have the maximum number of violations. Lets dive into the Issuer Precincts one by one.**

* **Issuer Precinct 18**

Hive> select Issuer\_Precinct,Violation\_Code, count(\*) as Violation\_code\_frequency from parking\_violations\_2017 where issuer\_precinct = 18 group by Issuer\_Precinct, Violation\_Code order by Violation\_code\_frequency desc limit 10;

**Text

Description automatically generated**

* **Issuer Precinct 19**

Hive> select Issuer\_Precinct,Violation\_Code, count(\*) as Violation\_code\_frequency from parking\_violations\_2017 where issuer\_precinct = 19 group by Issuer\_Precinct, Violation\_Code order by Violation\_code\_frequency desc limit 10;

Text

Description automatically generated

* **Issuer Precinct 14**

Hive> select Issuer\_Precinct,Violation\_Code, count(\*) as Violation\_code\_frequency from parking\_violations\_2017 where issuer\_precinct = 14 group by Issuer\_Precinct, Violation\_Code order by Violation\_code\_frequency desc limit 10;

**Text

Description automatically generated**

**5.) Find out the properties of parking violations across different times of the day: The Violation Time field is specified in a strange format. Find a way to make this into a time attribute that you can use to divide into groups.**

Hive> select from\_unixtime(unix\_timestamp(concat(violation\_time,'M'), 'HHmmaaa'),"HH:mmaaa") as date\_data from parking\_violations\_2017 limit 10;

Text

Description automatically generated

**6.) Divide 24 hours into 6 equal discrete bins of time. The intervals you choose are at your discretion. For each of these groups, find the 3 most commonly occurring violations**

hive> create view vw\_parking\_violations\_partitoned\_2017

partitioned on (Violation\_Code)

as

SELECT

Summons\_Number,

Violation\_Time,

Issuer\_Precinct,

case

when substring(Violation\_Time,1,2) in ('00','01','02','03','12') and upper(substring(Violation\_Time,-1))='A' then 1

when substring(Violation\_Time,1,2) in ('04','05','06','07') and upper(substring(Violation\_Time,-1))='A' then 2

when substring(Violation\_Time,1,2) in ('08','09','10','11') and upper(substring(Violation\_Time,-1))='A' then 3

when substring(Violation\_Time,1,2) in ('12','00','01','02','03') and upper(substring(Violation\_Time,-1))='P' then 4

when substring(Violation\_Time,1,2) in ('04','05','06','07') and upper(substring(Violation\_Time,-1))='P' then 5

when substring(Violation\_Time,1,2) in ('08','09','10','11') and upper(substring(Violation\_Time,-1))='P'then 6

else null

end as Violation\_Time\_bin,

Violation\_Code

from parking\_violations\_2017

where

Violation\_Time is not null

or (length(Violation\_Time)=5 and upper(substring(Violation\_Time,-1))in ('A','P')

and substring(Violation\_Time,1,2) in ('00','01','02','03','04','05','06','07', '08','09','10','11','12'));

Text

Description automatically generated

select Violation\_Code,Violation\_Time\_bin, count(\*) frequency\_violation from vw\_parking\_violations\_partitoned\_2017 group by Violation\_Code,Violation\_Time\_bin limit 5;

Text

Description automatically generated

* **bin1**

select Violation\_Code,count(\*) frequency\_violation from vw\_parking\_violations\_partitoned\_2017 where Violation\_Time\_bin =1 group by Violation\_Code order by frequency\_violation desc limit 3;

Text

Description automatically generated

* **bin2**

select Violation\_Code,count(\*) frequency\_violation from vw\_parking\_violations\_partitoned\_2017 where Violation\_Time\_bin =2 group by Violation\_Code order by frequency\_violation desc limit 3;

**Text

Description automatically generated**

* **bin3**

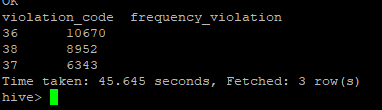
select Violation\_Code,count(\*) frequency\_violation from vw\_parking\_violations\_partitoned\_2017 where Violation\_Time\_bin =3 group by Violation\_Code order by frequency\_violation desc limit 3;

Text

Description automatically generated

* **bin4**

select Violation\_Code,count(\*) frequency\_violation from vw\_parking\_violations\_partitoned\_2017 where Violation\_Time\_bin =4 group by Violation\_Code order by frequency\_violation desc limit 3;



* **bin5**

select Violation\_Code,count(\*) frequency\_violation from vw\_parking\_violations\_partitoned\_2017 where Violation\_Time\_bin =5 group by Violation\_Code order by frequency\_violation desc limit 3;

Chart, text

Description automatically generated

* **bin6**

select Violation\_Code,count(\*) frequency\_violation from vw\_parking\_violations\_partitoned\_2017 where Violation\_Time\_bin =6 group by Violation\_Code order by frequency\_violation desc limit 3;

Text

Description automatically generated

**7.) Now, try another direction. For the 3 most commonly occurring violation codes, find the most common times of day (in terms of the bins from the previous part)**

Hive> select Violation\_Time\_bin, count(\*) frequency\_times from vw\_parking\_violations\_partitoned\_2017 where Violation\_Code in (21, 38,36)

group by Violation\_Time\_bin order by frequency\_times desc limit 3;

Text

Description automatically generated

**8.) Let’s try and find some seasonality in this data**

**a.) First, divide the year into some number of seasons, and find frequencies of tickets for each season. (Hint: A quick Google search reveals the following seasons in NYC: Spring(March, April, March); Summer(June, July, August); Fall(September, October, November); Winter(December, January, February))**

**Created View to divide the year into Seasons:**

Hive> create view vw\_tickets\_issued\_2017

partitioned on (Violation\_Code)

as

select Issuer\_Precinct,

case

when MONTH(from\_unixtime(Issue\_Date)) between 03 and 05 then 'spring'

when MONTH(from\_unixtime(Issue\_Date)) between 06 and 08 then 'summer'

when MONTH(from\_unixtime(Issue\_Date)) between 09 and 11 then 'autumn'

when MONTH(from\_unixtime(Issue\_Date)) in (1,2,12) then 'winter'

else 'unknown'

end as season,

Violation\_Code

from parking\_violations\_2017;

Hive> select season, count(\*) as frequency from vw\_tickets\_issued\_2017 group by season order by frequency desc;