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 a large number of parking tickets.

In an attempt to scientifically analyse this phenomenon, the NYC Police Department regularly collects data related to parking tickets. This data is made available by NYC Open Data portal. We will try and perform some analysis on this data.

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

```
create table parking_violations_v2(
Summons_Number bigint,
Plate_ID string,
Registration_State string,
Plate_Type string,
Issue_Date date,
Violation_Code int,
Vehicle_Body_Type string,
Vehicle_Make string,
Issuing_Agency string,
Street_Code1 int,
Street_Code2 int,
Street_Code3 int,
Vehicle_Expiration_Date int,
Violation_Location float,
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 float,
Vehicle_Year int,
Meter_Number string,
Feet_From_Curb int,
Violation_Post_Code string,
Violation_Description string,
No_Standing_or_Stopping_Violation float,
Hydrant_Violation float,
```

```
Double_Parking_Violation float
)

ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
WITH SERDEPROPERTIES (
   "escapeChar"    = "\\",
   "separatorChar" = ",",
   "quoteChar" = "\""
)
stored as textfile
TBLPROPERTIES("skip.header.line.count" = "1");

LOAD DATA INPATH '/tmp/hiveminiproject2/Parking_Violations_Issued_-
_Fiscal_Year_2017.csv' INTO TABLE  Parking_Violations;


create table Parking_Violations_Partitioned_Bucketed(
Summons_Number bigint,
Plate_ID string,
Plate_Type string,
Issue_Date date,
Violation_Code int,
Vehicle_Body_Type string,
Vehicle_Make string,
Issuing_Agency string,
Street_Code1 int,
Street_Code2 int,
Street_Code3 int,
Vehicle_Expiration_Date int,
Violation_Location float,
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 float,
Vehicle_Year int,
Meter_Number string,
Feet_From_Curb int,
Violation_Post_Code string,
Violation_Description string,
No_Standing_or_Stopping_Violation float,
Hydrant_Violation float,
Double_Parking_Violation float
```

```
)
Partitioned By (Registration_State string)
Clustered By (Summons_Number)
Into 8 Buckets;

set hive.exec.dynamic.partition = true;
set hive.exec.dynamic.partition.mode = nonstrict;
set hive.enforce.bucketing = true;

INSERT OVERWRITE TABLE Parking_Violations_Partitioned_Bucketed
PARTITION(registration_state) SELECT
summons_number,plate_id,plate_type,issue_date,violation_code,vehicle_body_type,vehi
cle_make,issuing_agency,street_code1,street_code2,street_code3,vehicle_expiration_d
ate,violation_location,violation_precinct,issuer_precinct,issuer_code,issuer_comman
d,issuer_squad,violation_time,time_first_observed,violation_county,violation_in_fro
nt_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_num
ber,feet_from_curb,violation_post_code,violation_description,no_standing_or_stoppin
g_violation,hydrant_violation,double_parking_violation,registration_state FROM
Parking_Violations_v2
where year(cast(to_date(from_unixtime(unix_timestamp(issue_date,'MM/dd/yyyy'))) as
date)) = '2017' LIMIT 10803028;



Part-I: Examine the data

1. Find the total number of tickets for the year.
SELECT COUNT(DISTINCT summons_number) FROM Parking_Violations_Partitioned_Bucketed;
Ans: 501920

2. Find out how many unique states the cars which got parking tickets came from.
SELECT COUNT(DISTINCT registration_state) FROM
Parking_Violations_Partitioned_Bucketed;
Ans: 65

3. 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) FROM Parking_Violations_Partitioned_Bucketed
WHERE Street_Code1 = 0 OR Street_Code2 = 0 OR Street_Code3 = 0;
Ans:
OK
77365
Time taken: 42.442 seconds, Fetched: 1 row(s)

Part-II: Aggregation tasks

1. How often does each violation code occur? (frequency of violation codes - find
the top 5)
SELECT Violation_Code, COUNT(*) as Violation_Code_Freq FROM
Parking_Violations_Partitioned_Bucketed
GROUP BY Violation_Code ORDER BY Violation_Code_Freq DESC LIMIT 5;
Ans:
Violation_Code, Violation_Code_Freq
21      768087
36      662765
38      542079
```

```
14      476664
20      319646
Time taken: 57.838 seconds, Fetched: 5 row(s)
```

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

Ans:

```
SELECT COUNT(DISTINCT summons_number) ticket_count_per_bodytype, Vehicle_Body_Type
FROM Parking_Violations_Partitioned_Bucketed GROUP BY Vehicle_Body_Type ORDER BY
ticket_count_per_bodytype DESC LIMIT 6
ticket_count_per_bodytype, vehicle_body_type
182773  SDN
148335  SUBN
60945   VAN
47817   DELV
9072    P-U
Time taken: 64.984 seconds, Fetched: 6 row(s)
```

```
SELECT COUNT(DISTINCT summons_number) ticket_count_per_vehiclemake, Vehicle_Make
FROM Parking_Violations_Partitioned_Bucketed GROUP BY Vehicle_Make ORDER BY
ticket_count_per_vehiclemake DESC LIMIT 6
ticket_count_per_vehiclemake, Vehicle_Make
53669   FORD
53550   TOYOT
48689   HONDA
40888   NISSA
27128   CHEVR
26437   FRUEH
Time taken: 65.394 seconds, Fetched: 6 row(s)
```

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)

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

Ans:

a)

```
SELECT COUNT(DISTINCT summons_number) as freq_of_violation_precincts,
Violation_Precinct FROM Parking_Violations_Partitioned_Bucketed GROUP BY
Violation_Precinct ORDER BY freq_of_violation_precincts DESC LIMIT 5
OK
15865   19
14720   18
14402   70
13880   72
13778   1
Time taken: 62.667 seconds, Fetched: 5 row(s)
```

b)

```
SELECT COUNT(DISTINCT summons_number) as freq_of_issuer_precinct,
Issuer_Precinct FROM Parking_Violations_Partitioned_Bucketed GROUP BY
Issuer_Precinct ORDER BY freq_of_issuer_precinct DESC LIMIT 5

freq_of_issuer_precinct, issuer_precinct
156560  0
12216   110
10268   109
10148   70
9907    401
Time taken: 62.329 seconds, Fetched: 5 row(s)
```

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?

```
SELECT COUNT(DISTINCT summons_number) tickes_issued, count(violation_code) as
freq_of_violation_code, Violation_Precinct FROM
Parking_Violations_Partitioned_Bucketed GROUP BY Violation_Precinct ORDER BY
tickes_issued DESC LIMIT 3;
```

```
tickes_issued, freq_of_violation_code, violation_precinct
15865   274445  19
14720   169131  18
14402   96552   70
Time taken: 65.938 seconds, Fetched: 3 row(s)
```

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.

Ans:

```
SELECT CONCAT_WS('', from_unixtime(unix_timestamp(REGEXP_EXTRACT(violation_time,
'^.{4}', 0),'HHmm'), 'HH:mm'), ' ',REGEXP_EXTRACT(violation_time, '.{1}$'  , 0),
'M') from parking_violations_partitioned_bucketed limit 10;
```

```
OK
05:12 PM
11:45 AM
02:25 AM
03:30 PM
02:31 AM
09:02 AM
10:41 AM
02:18 AM
08:18 AM
10:58 PM
Time taken: 0.198 seconds, Fetched: 10 row(s)
```

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

Ans:

```
WITH violation_group_cte AS(
SELECT violation_time_group, Violation_Code, COUNT(Violation_Code) AS
freq_of_violation_code FROM(SELECT Violation_Code,
CASE
WHEN SUBSTR(violation_time, 1, 2) IN ('00', '01', '02', '03') AND
SUBSTR(violation_time, -1) = 'A' THEN 'G1'
WHEN SUBSTR(violation_time, 1, 2) IN ('04', '05', '06', '07') AND
SUBSTR(violation_time, -1) = 'A' THEN 'G2'
WHEN SUBSTR(violation_time, 1, 2) IN ('08', '09', '10', '11') AND
SUBSTR(violation_time, -1) = 'A' THEN 'G3'
WHEN SUBSTR(violation_time, 1, 2) IN ('12', '01', '02', '03') AND
SUBSTR(violation_time, -1) = 'P' THEN 'G4'
WHEN SUBSTR(violation_time, 1, 2) IN ('04', '05', '06', '07') AND
SUBSTR(violation_time, -1) = 'P' THEN 'G5'
WHEN SUBSTR(violation_time, 1, 2) IN ('08', '09', '10', '11') AND
```

```
SUBSTR(violation_time, -1) = 'P' THEN 'G6'
ELSE 'G7' END AS violation_time_group
from parking_violations_partitioned_bucketed) AS T GROUP BY violation_time_group,
Violation_Code)

SELECT violation_time_group, Violation_Code FROM(
SELECT *, ROW_NUMBER() OVER(PARTITION BY violation_time_group ORDER BY
freq_of_violation_code DESC) AS RNK FROM violation_group_cte) AS T WHERE RNK<=3;

Total MapReduce CPU Time Spent: 52 seconds 660 msec
OK
G1      21
G1      40
G1      14
G2      14
G2      40
G2      21
G3      21
G3      36
G3      38
G4      36
G4      38
G4      37
G5      38
G5      14
G5      37
G6      7
G6      40
G6      14
G7      7
G7      21
G7      40
Time taken: 66.815 seconds, Fetched: 21 row(s)
```

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)

Ans:

```
WITH violation_group_cte AS(
SELECT violation_time_group, Violation_Code, COUNT(Violation_Code) AS
freq_of_violation_code FROM(SELECT Violation_Code,
CASE
WHEN SUBSTR(violation_time, 1, 2) IN ('00', '01', '02', '03') AND
SUBSTR(violation_time, -1) = 'A' THEN 'G1'
WHEN SUBSTR(violation_time, 1, 2) IN ('04', '05', '06', '07') AND
SUBSTR(violation_time, -1) = 'A' THEN 'G2'
WHEN SUBSTR(violation_time, 1, 2) IN ('08', '09', '10', '11') AND
SUBSTR(violation_time, -1) = 'A' THEN 'G3'
WHEN SUBSTR(violation_time, 1, 2) IN ('12', '01', '02', '03') AND
SUBSTR(violation_time, -1) = 'P' THEN 'G4'
WHEN SUBSTR(violation_time, 1, 2) IN ('04', '05', '06', '07') AND
SUBSTR(violation_time, -1) = 'P' THEN 'G5'
WHEN SUBSTR(violation_time, 1, 2) IN ('08', '09', '10', '11') AND
SUBSTR(violation_time, -1) = 'P' THEN 'G6'
ELSE 'G7' END AS violation_time_group
from parking_violations_partitioned_bucketed) AS T GROUP BY violation_time_group,
Violation_Code)

SELECT violation_time_group, Violation_Code FROM(
```

```
SELECT *, ROW_NUMBER() OVER(PARTITION BY violation_time_group ORDER BY
freq_of_violation_code DESC) AS RNK FROM violation_group_cte) AS T WHERE RNK<=3;

OK
G3      21      598069  1
G3      36      348165  2
G4      36      286284  3
```

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))
   b.)Then, find the 3 most common violations for each of these seasons.

Ans:

```
WITH season_cte AS(
SELECT
CASE
WHEN violation_month IN ('12', '1', '2') THEN 'Winter'
WHEN violation_month IN ('3', '4', '5') THEN 'Spring'
WHEN violation_month IN ('6', '7', '8') THEN 'Summer'
WHEN violation_month IN ('9', '10', '11') THEN 'Fall' END AS season, violation_code
FROM(
SELECT
MONTH(cast(to_date(from_unixtime(unix_timestamp(issue_date,'MM/dd/yyyy'))) as
date)) AS violation_month, Violation_Code from
parking_violations_partitioned_bucketed) AS T)

SELECT season, violation_code FROM(
SELECT *, ROW_NUMBER() OVER(PARTITION BY season ORDER BY freq_of_violation_code
DESC) AS RNK FROM(
SELECT season, violation_code, count(*) as freq_of_violation_code FROM season_cte
GROUP BY season, violation_code) AS T1) AS T2 WHERE RNK <=3;

OK
Fall    46
Fall    21
Fall    40
Spring  21
Spring  36
Spring  38
Summer  21
Summer  36
Summer  38
Winter  21
Winter  36
Winter  38
Time taken: 70.458 seconds, Fetched: 12 row(s)
```