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

```
Hive> create table parking_violations_issued
(
Summons_Number int,
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,
```

Violation_Location int,

Violation_Precinct int,

Issuer_Precinct int,

Issuer_Code int,

Issuer_Command string,

Issuer_Squad string,

Violation_Time int,

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");



alter table  parking_violations_issued change Summons_Number Summons_Number bigint;

alter table  parking_violations_issued change Violation_Time Violation_Time string;


load data local inpath'/home/cloudera/sidd/Challenge/mini_project_2/Parking_Violations_Issued_-
_Fiscal_Year_2017.csv' into table parking_violations_issued;


select * from parking_violations_issued limit 20;
```

Note: Consider only the year 2017 for analysis and not the Fiscal year.

for year 2017 only

```
Hive>  create table parking_violations_issued_2017
(
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,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)
COMMENT 'A bucketed sorted parking_violations_issued_2017'
partitioned by (Violation_County string)
CLUSTERED BY (Violation_Code) sorted by (Violation_Code) INTO 8 BUCKETS
row format delimited
fields terminated by ','
tblproperties ("skip.header.line.count" = "1");
```

To load data into partition and bucket table we need to 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;
```

Hive> insert into parking_violations_issued_2017 partition(Violation_County) select

Summons_Number,Plate_ID,Registration_State,Plate_Type,Issue_Date,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

year(Issue_Date) = '2017';

```
Last login: Mon Sep 19 09:33:12 2022 from 192.168.56.1
[cloudera@quickstart ~]$ hadoop fs -ls /user/hive/warehouse/challenge.db/
Found 8 items
drwxrwxrwx   - cloudera supergroup          0 2022-09-17 05:56 /user/hive/wareho
use/challenge.db/agentlogingreport
drwxrwxrwx   - cloudera supergroup          0 2022-09-19 02:06 /user/hive/wareho
use/challenge.db/agentlogingreport_partitioned
drwxrwxrwx   - cloudera supergroup          0 2022-09-17 06:17 /user/hive/wareho
use/challenge.db/agentperformance
drwxrwxrwx   - cloudera supergroup          0 2022-09-19 02:11 /user/hive/wareho
use/challenge.db/agentperformance_partitioned
drwxrwxrwx   - cloudera supergroup          0 2022-09-14 10:34 /user/hive/wareho
use/challenge.db/air_quality
drwxrwxrwx   - cloudera supergroup          0 2022-09-14 10:59 /user/hive/wareho
use/challenge.db/air_quality2
drwxrwxrwx   - cloudera supergroup          0 2022-09-19 09:31 /user/hive/wareho
use/challenge.db/parking_violations_issued
drwxrwxrwx   - cloudera supergroup          0 2022-09-19 21:09 /user/hive/wareho
use/challenge.db/parking_violations_issued_2017
[cloudera@quickstart ~]$ hadoop fs -ls /user/hive/warehouse/challenge.db/parking_violations_issued_2017/
Found 11 items
drwxrwxrwx   - cloudera supergroup          0 2022-09-19 21:09 /user/hive/warehouse/challenge.db/parking_violations_issued_2017/violation_county=BK
drwxrwxrwx   - cloudera supergroup          0 2022-09-19 21:08 /user/hive/warehouse/challenge.db/parking_violations_issued_2017/violation_county=BX
drwxrwxrwx   - cloudera supergroup          0 2022-09-19 21:08 /user/hive/warehouse/challenge.db/parking_violations_issued_2017/violation_county=K
drwxrwxrwx   - cloudera supergroup          0 2022-09-19 21:08 /user/hive/warehouse/challenge.db/parking_violations_issued_2017/violation_county=MN
drwxrwxrwx   - cloudera supergroup          0 2022-09-19 21:08 /user/hive/warehouse/challenge.db/parking_violations_issued_2017/violation_county=NY
drwxrwxrwx   - cloudera supergroup          0 2022-09-19 21:08 /user/hive/warehouse/challenge.db/parking_violations_issued_2017/violation_county=Q
drwxrwxrwx   - cloudera supergroup          0 2022-09-19 21:08 /user/hive/warehouse/challenge.db/parking_violations_issued_2017/violation_county=QN
drwxrwxrwx   - cloudera supergroup          0 2022-09-19 21:08 /user/hive/warehouse/challenge.db/parking_violations_issued_2017/violation_county=QNS
drwxrwxrwx   - cloudera supergroup          0 2022-09-19 21:08 /user/hive/warehouse/challenge.db/parking_violations_issued_2017/violation_county=R
drwxrwxrwx   - cloudera supergroup          0 2022-09-19 21:08 /user/hive/warehouse/challenge.db/parking_violations_issued_2017/violation_county=ST
drwxrwxrwx   - cloudera supergroup          0 2022-09-19 21:08 /user/hive/warehouse/challenge.db/parking_violations_issued_2017/violation_county=__HIVE_DEFAULT_PARTITION__
[cloudera@quickstart ~]$ hadoop fs -ls /user/hive/warehouse/challenge.db/parking_violations_issued_2017/violation_county=BK/
Found 8 items
-rwxrwxrwx   1 cloudera supergroup          0 2022-09-19 21:08 /user/hive/warehouse/challenge.db/parking_violations_issued_2017/violation_county=BK/000000_0
-rwxrwxrwx   1 cloudera supergroup          0 2022-09-19 21:08 /user/hive/warehouse/challenge.db/parking_violations_issued_2017/violation_county=BK/000001_0
-rwxrwxrwx   1 cloudera supergroup          0 2022-09-19 21:08 /user/hive/warehouse/challenge.db/parking_violations_issued_2017/violation_county=BK/000002_0
-rwxrwxrwx   1 cloudera supergroup          0 2022-09-19 21:08 /user/hive/warehouse/challenge.db/parking_violations_issued_2017/violation_county=BK/000003_0
-rwxrwxrwx   1 cloudera supergroup    5127904 2022-09-19 21:08 /user/hive/warehouse/challenge.db/parking_violations_issued_2017/violation_county=BK/000004_0
-rwxrwxrwx   1 cloudera supergroup     327904 2022-09-19 21:08 /user/hive/warehouse/challenge.db/parking_violations_issued_2017/violation_county=BK/000005_0
-rwxrwxrwx   1 cloudera supergroup          0 2022-09-19 21:08 /user/hive/warehouse/challenge.db/parking_violations_issued_2017/violation_county=BK/000006_0
-rwxrwxrwx   1 cloudera supergroup    1188550 2022-09-19 21:08 /user/hive/warehouse/challenge.db/parking_violations_issued_2017/violation_county=BK/000007_0
[cloudera@quickstart ~]$
```

The analysis can be divided into two parts:

Part-I: Examine the data

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

Hive> select count(distinct summons_number) No_Tickets ,year(issue_date) as year from parking_violations_issued_2017 group by year(issue_date);

```
hive> set hive.cli.print.header = true;
hive> select count(distinct summons_number) No_Tickets  ,year(issue_date) as year from parking_violations_issued_2017 group by year(issue_date);
Query ID = cloudera_20220919211313_46b22b8a-0259-4a14-8691-9bd18ff14f8b
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1663642807450_0006, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1663642807450_0006/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1663642807450_0006
Hadoop job information for Stage-1: number of mappers: 2; number of reducers: 1
2022-09-19 21:13:10,904 Stage-1 map = 0%,  reduce = 0%
2022-09-19 21:13:20,400 Stage-1 map = 50%,  reduce = 0%, Cumulative CPU 2.62 sec
2022-09-19 21:13:26,720 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 13.69 sec
2022-09-19 21:13:33,036 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 18.86 sec
MapReduce Total cumulative CPU time: 18 seconds 860 msec
Ended Job = job_1663642807450_0006
MapReduce Jobs Launched:
Stage-Stage-1: Map: 2  Reduce: 1   Cumulative CPU: 18.86 sec   HDFS Read: 100613653 HDFS Write: 12 SUCCESS
Total MapReduce CPU Time Spent: 18 seconds 860 msec
OK
no_tickets      year
539901  2017
Time taken: 31.025 seconds, Fetched: 1 row(s)
hive>
```

Total number of tickets for the year 2017 are 539901

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

Hive> select count(distinct Registration_State) as Reg_state_count from parking_violations_issued_2017;

```
hive> select count(distinct Registration_State)as Reg_state_count from parking_violations_issued_2017;
Query ID = cloudera_20220919211818_dd322532-6a96-46c1-9ece-f8879650ef89
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1663642807450_0008, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1663642807450_0008/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1663642807450_0008
Hadoop job information for Stage-1: number of mappers: 2; number of reducers: 1
2022-09-19 21:19:03,221 Stage-1 map = 0%,  reduce = 0%
2022-09-19 21:19:13,857 Stage-1 map = 50%,  reduce = 0%, Cumulative CPU 2.16 sec
2022-09-19 21:19:14,944 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 6.95 sec
2022-09-19 21:19:29,180 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 12.79 sec
MapReduce Total cumulative CPU time: 12 seconds 790 msec
Ended Job = job_1663642807450_0008
MapReduce Jobs Launched:
Stage-Stage-1: Map: 2  Reduce: 1   Cumulative CPU: 12.79 sec   HDFS Read: 100612455 HDFS Write: 3 SUCCESS
Total MapReduce CPU Time Spent: 12 seconds 790 msec
OK
reg_state_count
63
Time taken: 36.847 seconds, Fetched: 1 row(s)
hive>
```

Hive> select distinct(Registration_State) as Reg_state from parking_violations_issued_2017;

```
hive> select distinct(Registration_State) as Reg_state from parking_violations_issued_2017;
Query ID = cloudera_20220919212323_35b19739-81df-46c2-aef8-56fedaadee95
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1663642807450_0009, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1663642807450_0009/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1663642807450_0009
Hadoop job information for Stage-1: number of mappers: 2; number of reducers: 1
2022-09-19 21:23:34,038 Stage-1 map = 0%,  reduce = 0%
2022-09-19 21:23:45,278 Stage-1 map = 50%,  reduce = 0%, Cumulative CPU 2.64 sec
2022-09-19 21:23:48,620 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 8.99 sec
2022-09-19 21:24:00,505 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 12.67 sec
MapReduce Total cumulative CPU time: 12 seconds 670 msec
Ended Job = job_1663642807450_0009
MapReduce Jobs Launched:
Stage-Stage-1: Map: 2  Reduce: 1   Cumulative CPU: 12.67 sec   HDFS Read: 100611540 HDFS Write: 189 SUCCESS
Total MapReduce CPU Time Spent: 12 seconds 670 msec
OK
reg_state
99
AB
AK
AL
AR
AZ
BC
CA
CO
CT
DC
DE
DP
FL
GA
```

Hive> SELECT Registration_State,Count(1) as Number_of_Records from parking_violations_issued_2017 group by Registration_State order by Number_of_Records;

```
hive> SELECT Registration_State,Count(1) as Number_of_Records from parking_violations_issued_2017  group by Registration_State order by Number_of_Records desc limit 7;
Query ID = cloudera_20220921045959_7fcc3690-3b48-4dc7-b044-c415074fc6b2
Total jobs = 2
Launching Job 1 out of 2
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1663734229721_0031, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1663734229721_0031/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1663734229721_0031
Hadoop job information for Stage-1: number of mappers: 2; number of reducers: 1
2022-09-21 05:00:05,143 Stage-1 map = 0%,  reduce = 0%
2022-09-21 05:00:15,856 Stage-1 map = 50%,  reduce = 0%, Cumulative CPU 2.28 sec
2022-09-21 05:00:17,999 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 7.01 sec
2022-09-21 05:00:26,428 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 8.98 sec
MapReduce Total cumulative CPU time: 8 seconds 980 msec
Ended Job = job_1663734229721_0031
Launching Job 2 out of 2
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1663734229721_0032, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1663734229721_0032/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1663734229721_0032
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2022-09-21 05:00:36,633 Stage-2 map = 0%,  reduce = 0%
2022-09-21 05:00:42,922 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 1.27 sec
2022-09-21 05:00:52,581 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 3.48 sec
MapReduce Total cumulative CPU time: 3 seconds 480 msec
Ended Job = job_1663734229721_0032
MapReduce Jobs Launched:
Stage-Stage-1: Map: 2  Reduce: 1   Cumulative CPU: 8.98 sec   HDFS Read: 100611624 HDFS Write: 1490 SUCCESS
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 3.48 sec   HDFS Read: 6565 HDFS Write: 60 SUCCESS
Total MapReduce CPU Time Spent: 12 seconds 460 msec
OK
NY      424504
NJ      47479
PA      14044
CT      7034
FL      6957
IN      4507
MA      3849
Time taken: 60.387 seconds, Fetched: 7 row(s)
hive>
```

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)

Hive> select count(distinct summons_number) as No_Tickets_without_address from parking_violations_issued where Street_code1 = 0 or Street_code2 = 0 or Street_code3 = 0;

```
hive> SELECT Registration_State,Count(1) as Number_of_Records from parking_violations_issued_2017  group by Registration_State order by Number_of_Records desc limit 5;
Query ID = cloudera_20220919213838_331cc847-a3d9-41da-b012-c12ed78459a8
Total jobs = 2
Launching Job 1 out of 2
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1663642807450_0012, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1663642807450_0012/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1663642807450_0012
Hadoop job information for Stage-1: number of mappers: 2; number of reducers: 1
2022-09-19 21:38:30,241 Stage-1 map = 0%,  reduce = 0%
2022-09-19 21:38:40,340 Stage-1 map = 50%,  reduce = 0%, Cumulative CPU 2.53 sec
2022-09-19 21:38:48,098 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 9.98 sec
2022-09-19 21:38:58,979 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 12.77 sec
MapReduce Total cumulative CPU time: 12 seconds 770 msec
Ended Job = job_1663642807450_0012
Launching Job 2 out of 2
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1663642807450_0013, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1663642807450_0013/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1663642807450_0013
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2022-09-19 21:39:10,612 Stage-2 map = 0%,  reduce = 0%
2022-09-19 21:39:17,036 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 1.35 sec
2022-09-19 21:39:25,649 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 3.8 sec
MapReduce Total cumulative CPU time: 3 seconds 800 msec
Ended Job = job_1663642807450_0013
MapReduce Jobs Launched:
Stage-Stage-1: Map: 2  Reduce: 1   Cumulative CPU: 12.77 sec   HDFS Read: 100611624 HDFS Write: 1490 SUCCESS
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 3.8 sec   HDFS Read: 6565 HDFS Write: 44 SUCCESS
Total MapReduce CPU Time Spent: 16 seconds 570 msec
OK
registration_state      number_of_records
NY      424504
NJ      47479
PA      14044
CT      7034
FL      6957
Time taken: 66.549 seconds, Fetched: 5 row(s)
hive>
```

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 frequency_of_violation,Violation_Code from parking_violations_issued_2017 group by Violation_Code order by frequency_of_violation desc limit 5;

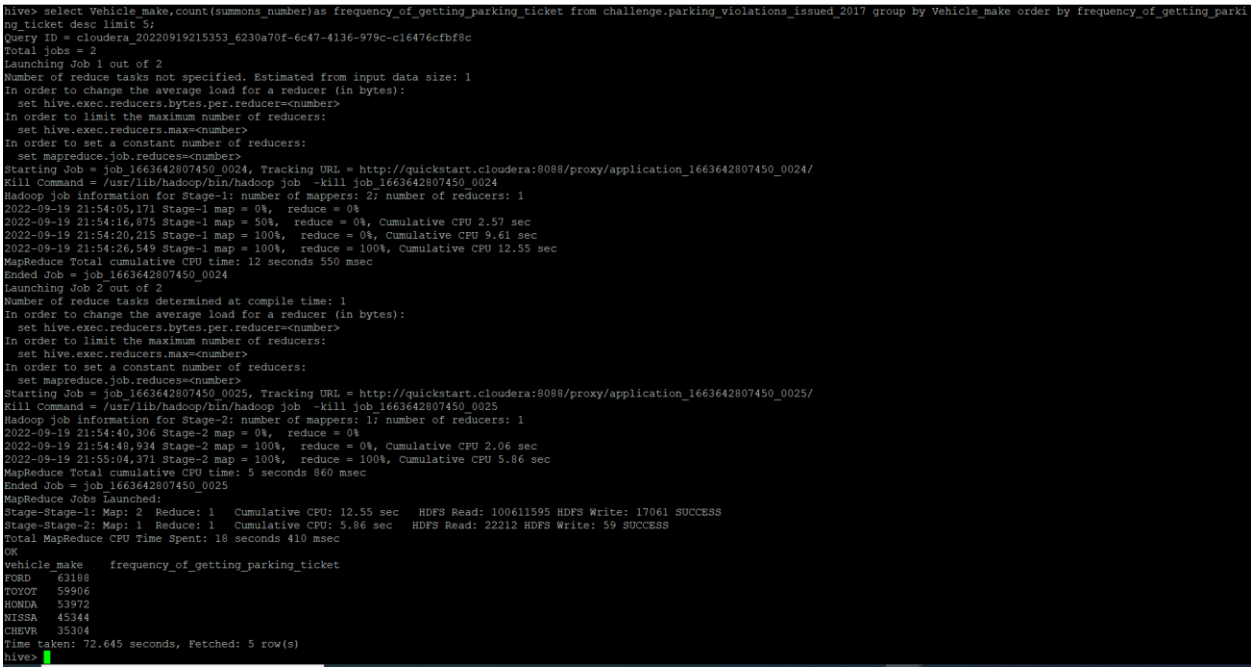
```
hive> select count(Violation_Code) as frequency_of_violation,Violation_Code from parking_violations_issued_2017 group by Violation_Code order by frequency_of_violation desc limit 5;
Query ID = cloudera_20220919214343_e295b181-a6a1-4ed4-a499-620c335a9d31
Total jobs = 2
Launching Job 1 out of 2
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1663642807450_0016, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1663642807450_0016/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1663642807450_0016
Hadoop job information for Stage-1: number of mappers: 2; number of reducers: 1
2022-09-19 21:43:54,561 Stage-1 map = 0%,  reduce = 0%
2022-09-19 21:44:04,195 Stage-1 map = 50%,  reduce = 0%, Cumulative CPU 2.88 sec
2022-09-19 21:44:11,824 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 12.85 sec
2022-09-19 21:44:20,416 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 17.51 sec
MapReduce Total cumulative CPU time: 17 seconds 510 msec
Ended Job = job_1663642807450_0016
Launching Job 2 out of 2
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1663642807450_0017, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1663642807450_0017/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1663642807450_0017
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2022-09-19 21:44:33,931 Stage-2 map = 0%,  reduce = 0%
2022-09-19 21:44:41,650 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 1.47 sec
2022-09-19 21:44:49,044 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 3.61 sec
MapReduce Total cumulative CPU time: 3 seconds 610 msec
Ended Job = job_1663642807450_0017
MapReduce Jobs Launched:
Stage-Stage-1: Map: 2  Reduce: 1   Cumulative CPU: 17.51 sec   HDFS Read: 100611778 HDFS Write: 1915 SUCCESS
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 3.61 sec   HDFS Read: 6835 HDFS Write: 45 SUCCESS
Total MapReduce CPU Time Spent: 21 seconds 120 msec
OK
frequency_of_violation  violation_code
76294   21
66093   36
53782   38
47181   14
31858   20
Time taken: 68.22 seconds, Fetched: 5 row(s)
hive>
```

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 frequency_of_getting_parking_ticket from challenge.parking_violations_issued_2017 group by Vehicle_Body_Type order by frequency_of_getting_parking_ticket desc limit 5; --done

```
hive> select Vehicle_Body_Type,count(summons_number)as frequency_of_getting_parking_ticket  from parking_violations_issued_2017 group by Vehicle_Body_Type order by frequency_of_getting_park
ing_ticket desc limit 5;
Query ID = cloudera_20220919215050_28c9b0ba-9cb1-4134-97f1-14a1ddd72743
Total jobs = 2
Launching Job 1 out of 2
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1663642807450_0020, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1663642807450_0020/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1663642807450_0020
Hadoop job information for Stage-1: number of mappers: 2; number of reducers: 1
2022-09-19 21:50:12,059 Stage-1 map = 0%,  reduce = 0%
2022-09-19 21:50:21,740 Stage-1 map = 50%,  reduce = 0%, Cumulative CPU 2.12 sec
2022-09-19 21:50:24,912 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 8.63 sec
2022-09-19 21:50:34,570 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 11.19 sec
MapReduce Total cumulative CPU time: 11 seconds 190 msec
Ended Job = job_1663642807450_0020
Launching Job 2 out of 2
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1663642807450_0021, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1663642807450_0021/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1663642807450_0021
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2022-09-19 21:50:47,752 Stage-2 map = 0%,  reduce = 0%
2022-09-19 21:50:55,237 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 1.45 sec
2022-09-19 21:51:05,045 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 4.21 sec
MapReduce Total cumulative CPU time: 4 seconds 210 msec
Ended Job = job_1663642807450_0021
MapReduce Jobs Launched:
Stage-Stage-1: Map: 2  Reduce: 1   Cumulative CPU: 11.19 sec   HDFS Read: 100611610 HDFS Write: 8610 SUCCESS
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 4.21 sec   HDFS Read: 13781 HDFS Write: 55 SUCCESS
Total MapReduce CPU Time Spent: 15 seconds 400 msec
OK
vehicle_body_type       frequency_of_getting_parking_ticket
SUBN    187281
4DSD    153964
VAN     71764
DELV    35600
SDN     19241
Time taken: 65.654 seconds, Fetched: 5 row(s)
hive>
```

Hive> select Vehicle_make,count(summons_number)as frequency_of_getting_parking_ticket from challenge.parking_violations_issued_2017 group by Vehicle_make order by frequency_of_getting_parking_ticket desc limit 5; --done



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(*) as IssuedTicket from challenge.parking_violations_issued group by Violation_Precinct order by IssuedTicket desc limit 6;--correct

```
hive> select Violation_Precinct,count(*) as IssuedTicket from challenge.parking_violations_issued group by  Violation_Precinct order by IssuedTicket desc limit 6;
Query ID = cloudera_20220919215959_ab7962d2-f879-484c-9e06-6d9b44081f73
Total jobs = 2
Launching Job 1 out of 2
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1663642807450_0032, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1663642807450_0032/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1663642807450_0032
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2022-09-19 22:00:13,142 Stage-1 map = 0%,  reduce = 0%
2022-09-19 22:00:38,735 Stage-1 map = 44%,  reduce = 0%, Cumulative CPU 7.17 sec
2022-09-19 22:00:40,901 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 7.99 sec
2022-09-19 22:00:52,688 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 10.82 sec
MapReduce Total cumulative CPU time: 10 seconds 820 msec
Ended Job = job_1663642807450_0032
Launching Job 2 out of 2
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1663642807450_0033, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1663642807450_0033/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1663642807450_0033
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2022-09-19 22:01:07,610 Stage-2 map = 0%,  reduce = 0%
2022-09-19 22:01:17,322 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 2.18 sec
2022-09-19 22:01:32,689 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 4.72 sec
MapReduce Total cumulative CPU time: 4 seconds 720 msec
Ended Job = job_1663642807450_0033
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 10.82 sec   HDFS Read: 201950849 HDFS Write: 2815 SUCCESS
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 4.72 sec   HDFS Read: 7866 HDFS Write: 54 SUCCESS
Total MapReduce CPU Time Spent: 15 seconds 540 msec
OK
violation_precinct      issuedticket
0       193893
19      52521
14      34506
1       32797
18      30183
114     29134
Time taken: 95.341 seconds, Fetched: 6 row(s)
hive>
```

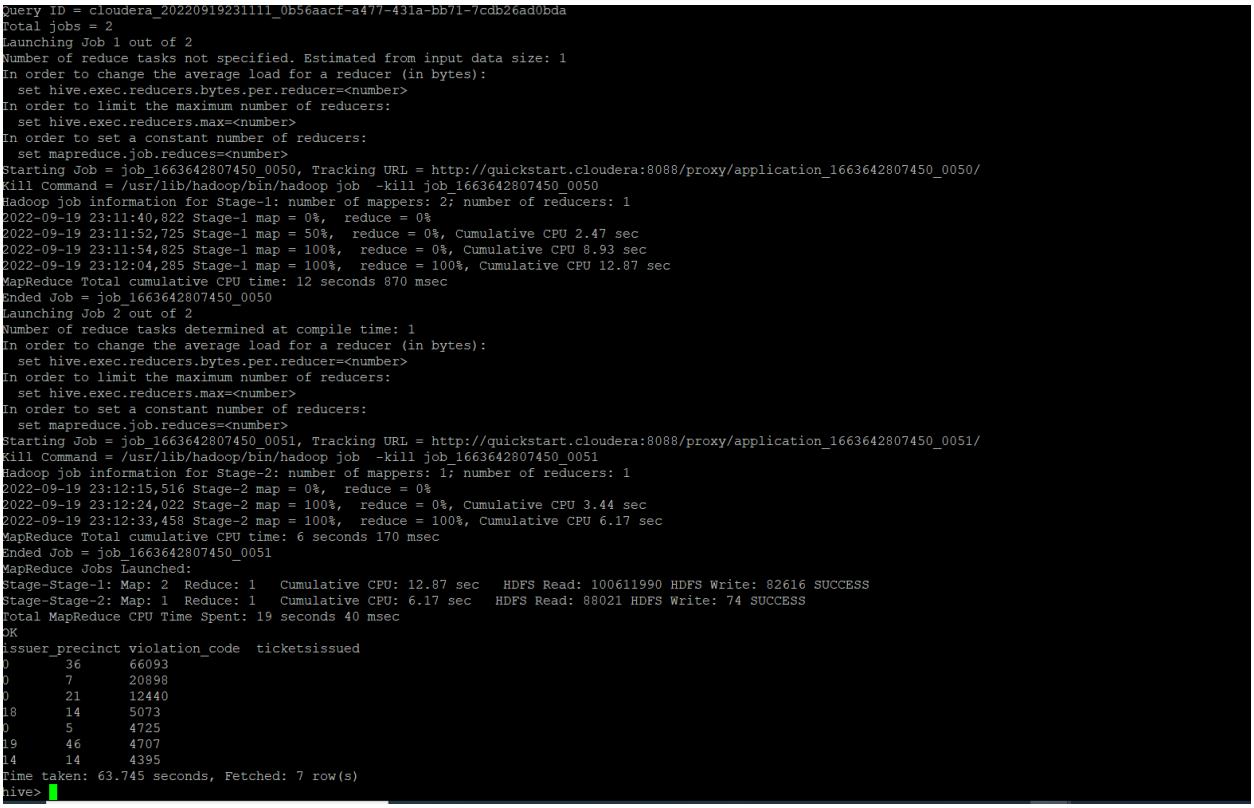
b.) Issuer Precincts (this is the precinct that issued the ticket)

Hive> select Issuer_Precinct,count(*) as IssuedTicket from challenge.parking_violations_issued group by Issuer_Precinct order by IssuedTicket desc limit 6;--correct

```
hive> select Issuer_Precinct,count(*) as IssuedTicket from challenge.parking_violations_issued group by Issuer_Precinct order by IssuedTicket desc limit 6;
Query ID = cloudera_20220919220404_9b5b6a86-0a3e-4065-a99d-bf5bff3ab36d
Total jobs = 2
Launching Job 1 out of 2
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1663642807450_0036, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1663642807450_0036/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1663642807450_0036
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2022-09-19 22:05:02,280 Stage-1 map = 0%,  reduce = 0%
2022-09-19 22:05:14,341 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 7.14 sec
2022-09-19 22:05:22,850 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 9.41 sec
MapReduce Total cumulative CPU time: 9 seconds 410 msec
Ended Job = job_1663642807450_0036
Launching Job 2 out of 2
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1663642807450_0037, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1663642807450_0037/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1663642807450_0037
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2022-09-19 22:05:37,434 Stage-2 map = 0%,  reduce = 0%
2022-09-19 22:05:48,220 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 2.94 sec
2022-09-19 22:05:58,977 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 6.28 sec
MapReduce Total cumulative CPU time: 6 seconds 280 msec
Ended Job = job_1663642807450_0037
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 9.41 sec   HDFS Read: 201950843 HDFS Write: 6889 SUCCESS
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 6.28 sec   HDFS Read: 11928 HDFS Write: 54 SUCCESS
Total MapReduce CPU Time Spent: 15 seconds 690 msec
OK
issuer_precinct issuedticket
0       224057
19      51157
14      33792
1       31784
18      29222
114     28511
Time taken: 71.394 seconds, Fetched: 6 row(s)
hive>
```

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 Issuer_Precinct,Violation_Code, count(*) as TicketsIssued from challenge.parking_violations_issued_2017 group by Issuer_Precinct, Violation_Code order by TicketsIssued desc limit 7;



We will not be considering 0. Therefore 18,19,14 are the three issuer precincts which have the maximum number of violations. Lets analyze the Issuer Precincts one by one.

--Issuer Precinct 18

select Violation_Code, count(*) as TicketsIssued from challenge.parking_violations_issued_2017 where Issuer_Precinct=18 group by Violation_Code order by TicketsIssued desc limit 7;

```
hive> select Violation_Code, count(*) as TicketsIssued from challenge.parking_violations_issued_2017 where Issuer_Precinct=18 group by Violation_Code order by TicketsIssued desc limit 7;
Query ID = cloudera_20220921051212_fc693675-1031-4c7c-af92-61fc7cafc878
Total jobs = 2
Launching Job 1 out of 2
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1663734229721_0033, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1663734229721_0033/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1663734229721_0033
Hadoop job information for Stage-1: number of mappers: 2; number of reducers: 1
2022-09-21 05:12:50,172 Stage-1 map = 0%,  reduce = 0%
2022-09-21 05:13:01,988 Stage-1 map = 50%,  reduce = 0%, Cumulative CPU 2.77 sec
2022-09-21 05:13:04,084 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 9.37 sec
2022-09-21 05:13:14,626 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 12.98 sec
MapReduce Total cumulative CPU time: 12 seconds 980 msec
Ended Job = job_1663734229721_0033
Launching Job 2 out of 2
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1663734229721_0034, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1663734229721_0034/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1663734229721_0034
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2022-09-21 05:13:27,655 Stage-2 map = 0%,  reduce = 0%
2022-09-21 05:13:34,176 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 1.41 sec
2022-09-21 05:13:42,643 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 4.32 sec
MapReduce Total cumulative CPU time: 4 seconds 320 msec
Ended Job = job_1663734229721_0034
MapReduce Jobs Launched:
Stage-Stage-1: Map: 2  Reduce: 1   Cumulative CPU: 12.98 sec   HDFS Read: 100613044 HDFS Write: 1268 SUCCESS
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 4.32 sec   HDFS Read: 6306 HDFS Write: 53 SUCCESS
Total MapReduce CPU Time Spent: 17 seconds 300 msec
OK
14      5073
69      2027
47      1367
31      1224
46      810
42      619
38      612
Time taken: 63.134 seconds, Fetched: 7 row(s)
hive>
```

--Issuer Precinct 19

select Violation_Code, count(*) as TicketsIssued from challenge.parking_violations_issued_2017 where Issuer_Precinct=19 group by Violation_Code order by TicketsIssued desc limit 7;

```
hive> select Violation_Code, count(*) as TicketsIssued from challenge.parking_violations_issued_2017 where Issuer_Precinct=19 group by Violation_Code order by TicketsIssued desc limit 7;
Query ID = cloudera_20220921051717_a4e1d48e-c083-4161-abfb-db78a370274e
Total jobs = 2
Launching Job 1 out of 2
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1663734229721_0035, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1663734229721_0035/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1663734229721_0035
Hadoop job information for Stage-1: number of mappers: 2; number of reducers: 1
2022-09-21 05:17:19,695 Stage-1 map = 0%,  reduce = 0%
2022-09-21 05:17:31,381 Stage-1 map = 50%,  reduce = 0%, Cumulative CPU 2.87 sec
2022-09-21 05:17:33,494 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 9.58 sec
2022-09-21 05:17:42,961 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 13.01 sec
MapReduce Total cumulative CPU time: 13 seconds 10 msec
Ended Job = job_1663734229721_0035
Launching Job 2 out of 2
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1663734229721_0036, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1663734229721_0036/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1663734229721_0036
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2022-09-21 05:17:53,568 Stage-2 map = 0%,  reduce = 0%
2022-09-21 05:17:59,928 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 1.31 sec
2022-09-21 05:18:08,296 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 3.22 sec
MapReduce Total cumulative CPU time: 3 seconds 220 msec
Ended Job = job_1663734229721_0036
MapReduce Jobs Launched:
Stage-Stage-1: Map: 2  Reduce: 1   Cumulative CPU: 13.01 sec   HDFS Read: 100613043 HDFS Write: 1176 SUCCESS
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 3.22 sec   HDFS Read: 6205 HDFS Write: 56 SUCCESS
Total MapReduce CPU Time Spent: 16 seconds 230 msec
OK
46      4707
37      3648
38      3573
14      2888
21      2842
20      1479
40      1070
Time taken: 57.808 seconds, Fetched: 7 row(s)
hive>
```

--Issuer Precinct 14

Hive> select Violation_Code, count(*) as TicketsIssued from challenge.parking_violations_issued_2017 where Issuer_Precinct=14 group by Violation_Code order by TicketsIssued desc limit 7;

--Common codes accross precincts

Hive> select Issuer_Precinct,Violation_Code, count(*) as TicketsIssued from challenge.parking_violations_issued_2017 where Issuer_Precinct in (18,19,14)

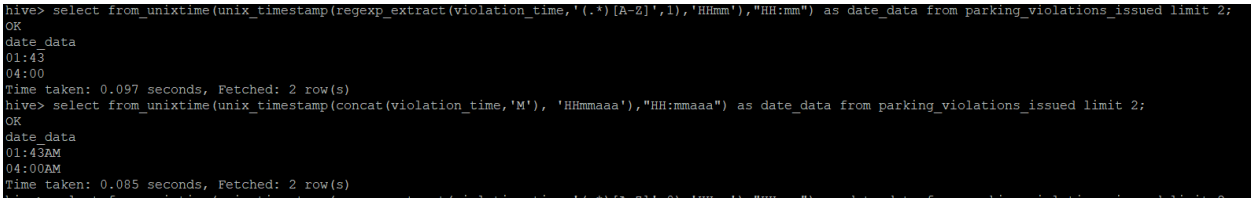
group by Issuer_Precinct,Violation_Code order by TicketsIssued desc limit 10;

```
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1663642807450_0054, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1663642807450_0054/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1663642807450_0054
Hadoop job information for Stage-1: number of mappers: 2; number of reducers: 1
2022-09-19 23:39:04,177 Stage-1 map = 0%,  reduce = 0%
2022-09-19 23:39:14,793 Stage-1 map = 50%,  reduce = 0%, Cumulative CPU 2.5 sec
2022-09-19 23:39:16,937 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 7.96 sec
2022-09-19 23:39:24,333 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 10.28 sec
MapReduce Total cumulative CPU time: 10 seconds 280 msec
Ended Job = job_1663642807450_0054
Launching Job 2 out of 2
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1663642807450_0055, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1663642807450_0055/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1663642807450_0055
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2022-09-19 23:39:34,476 Stage-2 map = 0%,  reduce = 0%
2022-09-19 23:39:41,826 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 1.46 sec
2022-09-19 23:39:50,349 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 3.78 sec
MapReduce Total cumulative CPU time: 3 seconds 780 msec
Ended Job = job_1663642807450_0055
MapReduce Jobs Launched:
Stage-Stage-1: Map: 2  Reduce: 1   Cumulative CPU: 10.28 sec   HDFS Read: 100610641 HDFS Write: 3782 SUCCESS
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 3.78 sec   HDFS Read: 9188 HDFS Write: 110 SUCCESS
Total MapReduce CPU Time Spent: 14 seconds 60 msec
OK
issuer_precinct violation_code  ticketsissued
18      14      5073
19      46      4707
14      14      4395
19      37      3648
19      38      3573
14      69      2997
19      14      2888
19      21      2842
14      31      2279
18      69      2027
Time taken: 57.783 seconds, Fetched: 10 row(s)
hive>
```

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(regexp_extract(violation_time,'(.*)[A-Z]',1),'HHmm'),"HH:mm") as date_data from parking_violations_issued limit 2;--> converted to time format 01:43

Hive> select from_unixtime(unix_timestamp(concat(violation_time,'M'), 'HHmmaaa'),"HH:mmaaa") as date_data from parking_violations_issued limit 2;--> working 01:43AM



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

partitoned view :

hive> create view vw_parking_violations_2017_partitioned_bins 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_issued_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'));

---

bin1

select Violation_Code,count(*) TicketsIssued from vw_parking_violations_2017_partitioned_bins where Violation_Time_bin == 1 group by Violation_Code order by TicketsIssued desc limit 3;

Violation_code	TicktesIssued
21	3660
40	2584
14	1574

bin2

select Violation_Code,count(*) TicketsIssued from vw_parking_violations_2017_partitioned_bins where Violation_Time_bin == 2 group by Violation_Code order by TicketsIssued desc limit 3;

Violation_code	TicktesIssued
14	7250
40	6403
21	5669

bin3

select Violation_Code,count(*) TicketsIssued from vw_parking_violations_2017_partitioned_bins where Violation_Time_bin == 3 group by Violation_Code order by TicketsIssued desc limit 3;

Violation_code	TicktesIssued
21	59465
36	37767
38	17587

bin4

select Violation_Code,count(*) TicketsIssued from vw_parking_violations_2017_partitioned_bins where Violation_Time_bin == 4 group by Violation_Code order by TicketsIssued desc limit 3;

Violation_code	TicktesIssued
36	28600
38	23877
37	16777

bin5

select Violation_Code,count(*) TicketsIssued from vw_parking_violations_2017_partitioned_bins where Violation_Time_bin == 5 group by Violation_Code order by TicketsIssued desc limit 3;

Violation_code	TicktesIssued
38	10148
14	7609
37	6944

bin6

select Violation_Code,count(*) TicketsIssued from vw_parking_violations_2017_partitioned_bins where Violation_Time_bin == 6 group by Violation_Code order by TicketsIssued desc limit 3;

Violation_code	TicktesIssued
7	2602
40	2159
14	2091

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(*) TicketsIssued from vw_parking_violations_2017_partitioned_bins where Violation_Code in (21, 37, 38,36)

group by Violation_Time_bin order by TicketsIssued desc limit 3;

Violation_Time_bin	TicketsIssued
3	116785
4	76701
5	18437

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))

Season Month interval

spring March, April, May

summer June, July, August

autumn September, October, November

winter December, January, February

normal view

```
Hive>  create view vw_tickets_issued_2017_bins as
select Violation_Code , Issuer_Precinct,
case
when MONTH(Issue_Date) between 03 and 05 then 'spring'
when MONTH(Issue_Date) between 06 and 08 then 'summer'
when MONTH(Issue_Date) between 09 and 11 then 'autumn'
when MONTH(Issue_Date) in (1,2,12) then 'winter'
else 'unknown' end  as season from parking_violations_issued_2017;
```

----

partioned view :

```
Hive>  create view vw_tickets_issued_2017_partitioned_bins partitioned on (Violation_Code) as

select Issuer_Precinct,

case

when MONTH(Issue_Date) between 03 and 05 then 'spring'

when MONTH(Issue_Date) between 06 and 08 then 'summer'

when MONTH(Issue_Date) between 09 and 11 then 'autumn' select

when MONTH(Issue_Date) in (1,2,12) then 'winter'

else 'unknown' end  as season,Violation_Code from parking_violations_issued_2017;
```

```
Hive>  select season, count(*) as TicketsIssued from vw_tickets_issued_2017_partitioned_bins group
by season order by TicketsIssued desc;
```

Season	TicktesIssued
Spring	285875
Winter	169466
Summer	84560
autumn	0

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

 # spring season

```
select Violation_Code, count(*) as TicketsIssued from vw_tickets_issued_2017_partitioned_bins
where

season = 'spring' group by Violation_Code order by TicketsIssued desc limit 3;
```

Violation_Code	TicketsIssued

21	40045
36	34354
38	27001

 # winter season

```
select Violation_Code, count(*) as TicketsIssued from vw_tickets_issued_2017_partitioned_bins
where

season = 'winter' group by Violation_Code order by TicketsIssued desc limit 3;
```

Violation_Code	TicketsIssued
21	23684
36	22084
38	18450

 # summer season

```
select Violation_Code, count(*) as TicketsIssued from vw_tickets_issued_2017_partitioned_bins
where

season = 'summer' group by Violation_Code order by TicketsIssued desc limit 3;
```

Violation_Code	TicketsIssued
21	12565
36	9655
38	8331

 # autumn season

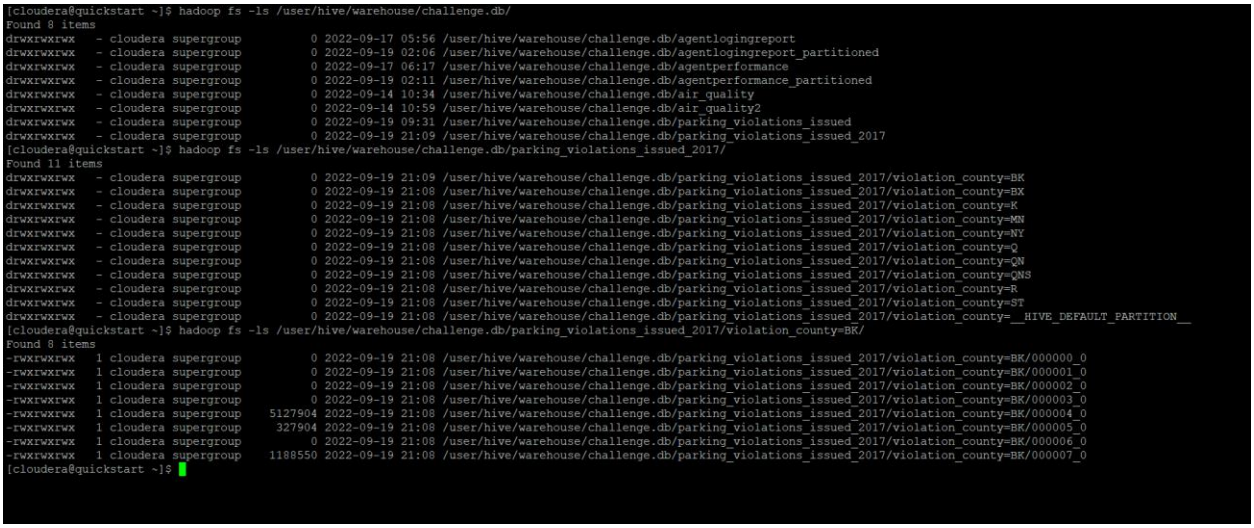
```
select Violation_Code, count(*) as TicketsIssued from vw_tickets_issued_2017_partitioned_bins
where

season = 'autumn' group by Violation_Code order by TicketsIssued desc limit 3;
```

Violation_Code	TicketsIssued

I have used partionsed and Bucketing on Table - **parking_violations_issued_2017** , and partions on views - **vw_parking_violations_2017_partitioned_bins partitioned, vw_tickets_issued_2017_bins**

That improves query performance.



Note: Please ensure you make necessary optimizations to your queries like selecting the appropriate table format, using partitioned/bucketed tables. Marks will be awarded for keeping the performance also in mind.