NOTES:

1. I have done all this in Cloudera VM which has just 8 GB RAM and 4 CPU cores. I am using HDFS (not s3). I have just tried to save cost associated with use of EC2 instance on AWS.
2. I have used hive CLI and not hue.
3. Section I contains hive queries, explanations, comments, logs, output etc.
4. Section II contains just queries for analysis I and II and output.
5. Section III contains just queries in an order as per assignment which can be executed as is.
6. All hive queries are formatted as such but when copied in word document, formatting got disturbed at places but if same query is copied from this document and pasted in hive CLI, code will still be seen as formatted.

SECTION I

1. Copy file from s3 location as provided in the assignment and copy it to HDFS.

[cloudera@quickstart ~]$ **aws s3 cp s3://hiveassignmentdatabde/Parking_Violations_Issued_-_Fiscal_Year_2017.csv nyc_data_2017.csv**
download: s3://hiveassignmentdatabde/Parking_Violations_Issued_-_Fiscal_Year_2017.csv to ./nyc_data_2017.csv

[cloudera@quickstart ~]$ ls -lrt nyc_data_2017.csv
-rw-rw-r-- 1 cloudera cloudera 2086913576 Jun 12 21:54 nyc_data_2017.csv
[cloudera@quickstart ~]$ **hadoop fs -put nyc_data_2017.csv /user/cloudera/hiveassignment/**
[cloudera@quickstart ~]$ hadoop fs -ls /user/cloudera/hiveassignment
Found 1 items
-rw-r--r--   1 cloudera cloudera 2086913576 2018-11-09 10:02 /user/cloudera/hiveassignment/nyc_data_2017.csv
[cloudera@quickstart ~]$

2. Login to hive CLI and create a separate database namely “hive_assignment” and create tables in this database.

create database hive_assignment;

hive> create database hive_assignment;
OK
Time taken: 0.34 seconds
hive>

use hive_assignment;

hive> use hive_assignment;
OK
Time taken: 0.038 seconds
hive>

3. Set few parameters required for partitioning, bucketing, orc file format and compression. I was facing lot of issues while converting into orc file format along with compression. In order to resolve issues with conversion to orc file format along with compression, I am using last few statements below which I found on internet while troubleshooting and it was very helpful. I have highlighted below.

```
set hive.exec.dynamic.partition= true ;
set hive.exec.dynamic.partition.mode=nonstrict;
set hive.exec.max.dynamic.partitions= 1000 ;
set hive.exec.max.dynamic.partitions.pernode= 1000 ;
set hive.enforce.bucketing= true ;
set hive.stats.autogather=true;
SET hive.optimize.sort.dynamic.partition=true;
SET orc.compress=SNAPPY;
SET hive.exec.compress.output=true;
SET mapred.output.compression.codec=org.apache.hadoop.io.compress.SnappyCodec;
SET mapred.output.compression.type=BLOCK;
set mapreduce.map.memory.mb=5120;
set mapreduce.reduce.memory.mb=5120;
set mapreduce.map.java.opts=-Xmx5G;
set mapreduce.reduce.java.opts=-Xmx5G;
SET mapred.child.java.opts=-Xmx5G -XX:+UseConcMarkSweepGC  -XX:-UseGCOverheadLimit;
```

```
hive> set hive.exec.dynamic.partition= true ;
hive> set hive.exec.dynamic.partition.mode=nonstrict;
hive> set hive.exec.max.dynamic.partitions= 1000 ;
hive> set hive.exec.max.dynamic.partitions.pernode= 1000 ;
hive> set hive.enforce.bucketing= true ;
hive> set hive.stats.autogather=true;
hive> SET hive.optimize.sort.dynamic.partition=true;
hive> SET orc.compress=SNAPPY;
hive> SET hive.exec.compress.output=true;
hive> SET mapred.output.compression.codec=org.apache.hadoop.io.compress.SnappyCodec;
hive> SET mapred.output.compression.type=BLOCK;
hive> set mapreduce.map.memory.mb=5120;
hive> set mapreduce.reduce.memory.mb=5120;
hive> set mapreduce.map.java.opts=-Xmx5G;
hive> set mapreduce.reduce.java.opts=-Xmx5G;
hive> SET mapred.child.java.opts=-Xmx5G -XX:+UseConcMarkSweepGC  -XX:-UseGCOverheadLimit;
hive>
```

4. First create an external table namely “NYC_DATA_EXT” so data can be accessed from file present in HDFS.

--- Create external table NYC_DATA_EXT

```
CREATE EXTERNAL TABLE IF NOT EXISTS NYC_DATA_EXT(
`SUMMONS_NUMBER` INT,
`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_DATE` INT,
`VIOLATION_LOCATION` STRING,
```

```
`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` STRING,
`VEHICLE_YEAR` INT,
`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 ','
LOCATION '/user/cloudera/hiveassignment/'
TBLPROPERTIES("skip.header.line.count"="1");

hive> CREATE EXTERNAL TABLE IF NOT EXISTS NYC_DATA_EXT(
    > `SUMMONS_NUMBER` INT,
    > `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_DATE` INT,
    > `VIOLATION_LOCATION` STRING,
    > `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` STRING,
  > `VEHICLE_YEAR` INT,
  > `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 ','
  > LOCATION '/user/cloudera/hiveassignment/'
  > TBLPROPERTIES("skip.header.line.count"="1");
OK
Time taken: 0.115 seconds
```

5. Creating an external table which is partitioned and bucketed. **Partitioning key chosen is "VIOLATION_CODE" and bucketing is done on "SUMMONS_NUMBER".**
   (a) There are quite a few queries on violation codes and violation code 0 is not a valid violation code. I am supposed to ignore violation code 0 so I decided to partition on violation_code and partition violation_code=0 will be ignored and full table scan will be avoided.
   (b) There are quite a few queries on summons_number and it contains NULL (empty) for many records which is invalid. I am supposed to ignore NULL for summons_number so I decided to create buckets using summons_number. Same values go in same bucket so NULL values go in same bucket and will help in running queries faster which are using summons_number and can ignore that bucket (or most of entries in that bucket).
   (c) I am creating **11 buckets**. I have chosen a prime number as it will be used in MOD operation of hash function internally and it helps in order to get better distribution. Creating lesser buckets and more than 11, were not helping in query performance. I found 11 number as optimal number of buckets.
   (d) I am using **ORC** file format for storage and using **SNAPPY** compression. This will help in reducing size of data and will result in faster I/O which eventually helps in running query faster.
   (e) I am creating a table namely "**NYC_DATA_PARTITIONED_BUCKETED_ORC**" in order to get partitions and buckets created, converting to ORC format and compressing using SNAPPY compression.
   (f) I am using external tables this time as well just in case if I need to see how much size has been reduced after converting into ORC file format using SNAPPY compression.
   (g) Field **issue_date** is not in appropriate format. Converting it into hive date format and populating data only for year **2017** as per mentioned in the assignment to use data of only 2017 for analysis.
   (h) Just in case needed, run the statistics for the table by running this command after table creation: **analyze table nyc_data_partitioned_bucketed_orc partition(violation_code) compute statistics;**

--- Create external table NYC_DATA_PARTITIONED_BUCKETED_ORC

```
CREATE EXTERNAL TABLE IF NOT EXISTS NYC_DATA_PARTITIONED_BUCKETED_ORC(
`SUMMONS_NUMBER` INT,
`PLATE_ID` STRING,
`REGISTRATION_STATE` STRING,
`PLATE_TYPE` STRING,
`ISSUE_DATE` DATE,
`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` STRING,
`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` STRING,
`VEHICLE_YEAR` INT,
`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_CODE INT)
CLUSTERED BY (SUMMONS_NUMBER) INTO 11 BUCKETS
STORED AS ORC
LOCATION '/user/cloudera/hiveassignment_orc/'
TBLPROPERTIES("orc.compress"="SNAPPY");
```

```
hive> CREATE EXTERNAL TABLE IF NOT EXISTS NYC_DATA_PARTITIONED_BUCKETED_ORC(
    > `SUMMONS_NUMBER` INT,
    > `PLATE_ID` STRING,
    > `REGISTRATION_STATE` STRING,
    > `PLATE_TYPE` STRING,
    > `ISSUE_DATE` DATE,
    > `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` STRING,
    > `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` STRING,
    > `VEHICLE_YEAR` INT,
    > `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_CODE INT)
    > CLUSTERED BY (SUMMONS_NUMBER) INTO 11 BUCKETS
    > STORED AS ORC
    > LOCATION '/user/cloudera/hiveassignment_orc/'
    > TBLPROPERTIES("orc.compress"="SNAPPY");
OK
Time taken: 0.19 seconds
hive>
```

--- Populate table NYC_DATA_PARTITIONED_BUCKETED_ORC from NYC_DATA_EXT

```
INSERT OVERWRITE TABLE NYC_DATA_PARTITIONED_BUCKETED_ORC PARTITION(VIOLATION_CODE)
SELECT SUMMONS_NUMBER,
    PLATE_ID,
    REGISTRATION_STATE,
    PLATE_TYPE,
    to_date(from_unixtime(unix_timestamp(issue_date, 'MM/dd/yyyy'), 'yyyy-MM-dd')),
    VEHICLE_BODY_TYPE,
    VEHICLE_MAKE,
    ISSUING_AGENCY,
    STREET_CODE1,
    STREET_CODE2,
    STREET_CODE3,
    VEHICLE_EXPIRATION_DATE,
    VIOLATION_LOCATION,
    VIOLATION_PRECINCT,
    ISSUER_PRECINCT,
    ISSUER_CODE,
    ISSUER_COMMAND,
    ISSUER_SQUAD,
    VIOLATION_TIME,
    TIME_FIRST_OBSERVED,
    VIOLATION_COUNTY,
    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_CODE
FROM NYC_DATA_EXT
WHERE year(to_date(from_unixtime(unix_timestamp(issue_date, 'MM/dd/yyyy'), 'yyyy-MM-dd')))='2017';

hive> INSERT OVERWRITE TABLE NYC_DATA_PARTITIONED_BUCKETED_ORC PARTITION(VIOLATION_CODE)
   > SELECT SUMMONS_NUMBER,
   >       PLATE_ID,
   >       REGISTRATION_STATE,
```

```
  >     PLATE_TYPE,
  >     to_date(from_unixtime(unix_timestamp(issue_date, 'MM/dd/yyyy'), 'yyyy-MM-dd')),
  >     VEHICLE_BODY_TYPE,
  >     VEHICLE_MAKE,
  >     ISSUING_AGENCY,
  >     STREET_CODE1,
  >     STREET_CODE2,
  >     STREET_CODE3,
  >     VEHICLE_EXPIRATION_DATE,
  >     VIOLATION_LOCATION,
  >     VIOLATION_PRECINCT,
  >     ISSUER_PRECINCT,
  >     ISSUER_CODE,
  >     ISSUER_COMMAND,
  >     ISSUER_SQUAD,
  >     VIOLATION_TIME,
  >     TIME_FIRST_OBSERVED,
  >     VIOLATION_COUNTY,
  >     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_CODE
  > FROM NYC_DATA_EXT
  > WHERE year(to_date(from_unixtime(unix_timestamp(issue_date, 'MM/dd/yyyy'), 'yyyy-MM-dd')))='2017';
Query ID = cloudera_20181118052222_c72f9cb8-301b-4e84-822f-0fbf423b4b4e
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 9
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_1542539368466_0049, Tracking URL =
http://quickstart.cloudera:8088/proxy/application_1542539368466_0049/
```

```
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1542539368466_0049
Hadoop job information for Stage-1: number of mappers: 8; number of reducers: 9
2018-11-18 05:22:35,163 Stage-1 map = 0%,  reduce = 0%
2018-11-18 05:23:12,222 Stage-1 map = 4%,  reduce = 0%, Cumulative CPU 30.27 sec
2018-11-18 05:23:30,291 Stage-1 map = 13%,  reduce = 0%, Cumulative CPU 49.34 sec
2018-11-18 05:24:00,996 Stage-1 map = 17%,  reduce = 0%, Cumulative CPU 76.87 sec
2018-11-18 05:24:24,415 Stage-1 map = 25%,  reduce = 0%, Cumulative CPU 100.58 sec
2018-11-18 05:25:01,006 Stage-1 map = 29%,  reduce = 0%, Cumulative CPU 134.91 sec
2018-11-18 05:25:25,419 Stage-1 map = 36%,  reduce = 0%, Cumulative CPU 160.23 sec
2018-11-18 05:25:26,504 Stage-1 map = 38%,  reduce = 0%, Cumulative CPU 160.98 sec
2018-11-18 05:25:56,108 Stage-1 map = 42%,  reduce = 0%, Cumulative CPU 187.43 sec
2018-11-18 05:26:13,975 Stage-1 map = 46%,  reduce = 0%, Cumulative CPU 207.06 sec
2018-11-18 05:26:16,041 Stage-1 map = 50%,  reduce = 0%, Cumulative CPU 208.71 sec
2018-11-18 05:26:45,437 Stage-1 map = 54%,  reduce = 0%, Cumulative CPU 236.07 sec
2018-11-18 05:27:03,267 Stage-1 map = 60%,  reduce = 0%, Cumulative CPU 254.53 sec
2018-11-18 05:27:05,360 Stage-1 map = 63%,  reduce = 0%, Cumulative CPU 256.42 sec
2018-11-18 05:27:40,120 Stage-1 map = 67%,  reduce = 0%, Cumulative CPU 288.63 sec
2018-11-18 05:27:58,043 Stage-1 map = 75%,  reduce = 0%, Cumulative CPU 307.58 sec
2018-11-18 05:28:28,622 Stage-1 map = 79%,  reduce = 0%, Cumulative CPU 334.23 sec
2018-11-18 05:28:51,788 Stage-1 map = 88%,  reduce = 0%, Cumulative CPU 358.34 sec
2018-11-18 05:29:27,995 Stage-1 map = 93%,  reduce = 0%, Cumulative CPU 392.38 sec
2018-11-18 05:29:33,196 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 395.99 sec
2018-11-18 05:29:52,547 Stage-1 map = 100%,  reduce = 8%, Cumulative CPU 412.21 sec
2018-11-18 05:30:28,252 Stage-1 map = 100%,  reduce = 11%, Cumulative CPU 449.27 sec
2018-11-18 05:30:50,403 Stage-1 map = 100%,  reduce = 19%, Cumulative CPU 468.14 sec
2018-11-18 05:31:02,931 Stage-1 map = 100%,  reduce = 21%, Cumulative CPU 480.62 sec
2018-11-18 05:31:14,468 Stage-1 map = 100%,  reduce = 22%, Cumulative CPU 493.14 sec
2018-11-18 05:31:38,566 Stage-1 map = 100%,  reduce = 30%, Cumulative CPU 512.15 sec
2018-11-18 05:32:02,622 Stage-1 map = 100%,  reduce = 33%, Cumulative CPU 536.1 sec
2018-11-18 05:32:29,944 Stage-1 map = 100%,  reduce = 41%, Cumulative CPU 559.85 sec
2018-11-18 05:33:06,495 Stage-1 map = 100%,  reduce = 44%, Cumulative CPU 596.04 sec
2018-11-18 05:33:28,510 Stage-1 map = 100%,  reduce = 52%, Cumulative CPU 615.91 sec
2018-11-18 05:33:34,829 Stage-1 map = 100%,  reduce = 53%, Cumulative CPU 621.92 sec
2018-11-18 05:33:59,132 Stage-1 map = 100%,  reduce = 54%, Cumulative CPU 647.1 sec
2018-11-18 05:34:11,720 Stage-1 map = 100%,  reduce = 55%, Cumulative CPU 658.58 sec
2018-11-18 05:34:15,881 Stage-1 map = 100%,  reduce = 56%, Cumulative CPU 661.81 sec
2018-11-18 05:34:35,203 Stage-1 map = 100%,  reduce = 63%, Cumulative CPU 678.79 sec
2018-11-18 05:34:40,577 Stage-1 map = 100%,  reduce = 64%, Cumulative CPU 684.8 sec
2018-11-18 05:34:58,631 Stage-1 map = 100%,  reduce = 67%, Cumulative CPU 702.02 sec
2018-11-18 05:35:20,948 Stage-1 map = 100%,  reduce = 74%, Cumulative CPU 720.31 sec
2018-11-18 05:35:27,262 Stage-1 map = 100%,  reduce = 76%, Cumulative CPU 726.43 sec
2018-11-18 05:35:39,791 Stage-1 map = 100%,  reduce = 78%, Cumulative CPU 736.88 sec
2018-11-18 05:35:57,638 Stage-1 map = 100%,  reduce = 86%, Cumulative CPU 751.76 sec
2018-11-18 05:36:15,471 Stage-1 map = 100%,  reduce = 89%, Cumulative CPU 770.7 sec
2018-11-18 05:36:38,490 Stage-1 map = 100%,  reduce = 96%, Cumulative CPU 786.96 sec
2018-11-18 05:36:56,347 Stage-1 map = 100%,  reduce = 97%, Cumulative CPU 806.36 sec
2018-11-18 05:37:26,697 Stage-1 map = 100%,  reduce = 99%, Cumulative CPU 836.1 sec
2018-11-18 05:37:38,322 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 845.68 sec
MapReduce Total cumulative CPU time: 14 minutes 5 seconds 680 msec
Ended Job = job_1542539368466_0049
Loading data to table hive_assignment.nyc_data_partitioned_bucketed_orc partition (violation_code=null)
        Time taken for load dynamic partitions : 15081
        Loading partition {violation_code=10}
        Loading partition {violation_code=92}
```

Loading partition {violation_code=41}
Loading partition {violation_code=2}
Loading partition {violation_code=65}
Loading partition {violation_code=60}
Loading partition {violation_code=6}
Loading partition {violation_code=66}
Loading partition {violation_code=23}
Loading partition {violation_code=88}
Loading partition {violation_code=48}
Loading partition {violation_code=47}
Loading partition {violation_code=75}
Loading partition {violation_code=53}
Loading partition {violation_code=71}
Loading partition {violation_code=24}
Loading partition {violation_code=4}
Loading partition {violation_code=95}
Loading partition {violation_code=98}
Loading partition {violation_code=39}
Loading partition {violation_code=94}
Loading partition {violation_code=31}
Loading partition {violation_code=9}
Loading partition {violation_code=97}
Loading partition {violation_code=21}
Loading partition {violation_code=64}
Loading partition {violation_code=19}
Loading partition {violation_code=59}
Loading partition {violation_code=25}
Loading partition {violation_code=1}
Loading partition {violation_code=56}
Loading partition {violation_code=69}
Loading partition {violation_code=54}
Loading partition {violation_code=49}
Loading partition {violation_code=70}
Loading partition {violation_code=77}
Loading partition {violation_code=8}
Loading partition {violation_code=30}
Loading partition {violation_code=45}
Loading partition {violation_code=40}
Loading partition {violation_code=57}
Loading partition {violation_code=58}
Loading partition {violation_code=20}
Loading partition {violation_code=11}
Loading partition {violation_code=84}
Loading partition {violation_code=32}
Loading partition {violation_code=46}
Loading partition {violation_code=76}
Loading partition {violation_code=37}
Loading partition {violation_code=42}
Loading partition {violation_code=35}
Loading partition {violation_code=29}
Loading partition {violation_code=90}
Loading partition {violation_code=13}
Loading partition {violation_code=62}
Loading partition {violation_code=52}

Loading partition {violation_code=82}
Loading partition {violation_code=28}
Loading partition {violation_code=33}
Loading partition {violation_code=26}
Loading partition {violation_code=55}
Loading partition {violation_code=93}
Loading partition {violation_code=63}
Loading partition {violation_code=72}
Loading partition {violation_code=85}
Loading partition {violation_code=16}
Loading partition {violation_code=81}
Loading partition {violation_code=67}
Loading partition {violation_code=0}
Loading partition {violation_code=5}
Loading partition {violation_code=3}
Loading partition {violation_code=38}
Loading partition {violation_code=18}
Loading partition {violation_code=87}
Loading partition {violation_code=51}
Loading partition {violation_code=80}
Loading partition {violation_code=89}
Loading partition {violation_code=79}
Loading partition {violation_code=96}
Loading partition {violation_code=34}
Loading partition {violation_code=14}
Loading partition {violation_code=12}
Loading partition {violation_code=78}
Loading partition {violation_code=73}
Loading partition {violation_code=68}
Loading partition {violation_code=17}
Loading partition {violation_code=15}
Loading partition {violation_code=43}
Loading partition {violation_code=22}
Loading partition {violation_code=83}
Loading partition {violation_code=7}
Loading partition {violation_code=86}
Loading partition {violation_code=61}
Loading partition {violation_code=91}
Loading partition {violation_code=36}
Loading partition {violation_code=50}
Loading partition {violation_code=44}
Loading partition {violation_code=27}
Loading partition {violation_code=99}
Loading partition {violation_code=74}
Time taken for adding to write entity : 39
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=0} stats: [numFiles=11, numRows=227, totalSize=19570, rawDataSize=583471]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=1} stats: [numFiles=11, numRows=674, totalSize=72776, rawDataSize=1763035]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=10} stats: [numFiles=11, numRows=25923, totalSize=1022257, rawDataSize=67559154]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=11} stats: [numFiles=11, numRows=5592, totalSize=259153, rawDataSize=14664624]

Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=12} stats: [numFiles=11, numRows=53, totalSize=47892, rawDataSize=138604]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=13} stats: [numFiles=11, numRows=11673, totalSize=497292, rawDataSize=30558134]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=14} stats: [numFiles=11, numRows=476660, totalSize=21191770, rawDataSize=1243988156]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=15} stats: [numFiles=11, numRows=7, totalSize=18430, rawDataSize=18257]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=16} stats: [numFiles=11, numRows=74790, totalSize=3403618, rawDataSize=196432899]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=17} stats: [numFiles=11, numRows=38449, totalSize=1796879, rawDataSize=100880532]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=18} stats: [numFiles=11, numRows=10188, totalSize=458344, rawDataSize=26651844]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=19} stats: [numFiles=11, numRows=149061, totalSize=6634001, rawDataSize=389756193]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=2} stats: [numFiles=11, numRows=77, totalSize=41985, rawDataSize=201312]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=20} stats: [numFiles=11, numRows=319646, totalSize=14719326, rawDataSize=837168499]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=21} stats: [numFiles=11, numRows=768082, totalSize=35256139, rawDataSize=2013656818]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=22} stats: [numFiles=11, numRows=81, totalSize=26816, rawDataSize=212559]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=23} stats: [numFiles=11, numRows=9697, totalSize=426469, rawDataSize=25424170]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=24} stats: [numFiles=11, numRows=38460, totalSize=1790747, rawDataSize=101039732]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=25} stats: [numFiles=11, numRows=115, totalSize=35161, rawDataSize=302592]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=26} stats: [numFiles=11, numRows=660, totalSize=39640, rawDataSize=1737720]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=27} stats: [numFiles=11, numRows=3039, totalSize=223592, rawDataSize=7940051]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=28} stats: [numFiles=11, numRows=5, totalSize=15294, rawDataSize=13123]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=29} stats: [numFiles=11, numRows=35, totalSize=17426, rawDataSize=91757]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=3} stats: [numFiles=11, numRows=407, totalSize=61591, rawDataSize=1064625]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=30} stats: [numFiles=11, numRows=553, totalSize=71818, rawDataSize=1445492]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=31} stats: [numFiles=11, numRows=80593, totalSize=3234309, rawDataSize=211465245]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=32} stats: [numFiles=11, numRows=14, totalSize=5946, rawDataSize=36866]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=33} stats: [numFiles=11, numRows=28, totalSize=32360, rawDataSize=73396]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=34} stats: [numFiles=11, numRows=11, totalSize=20610, rawDataSize=28804]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=35} stats: [numFiles=11, numRows=2034, totalSize=102195, rawDataSize=5355080]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=36} stats: [numFiles=11, numRows=662765, totalSize=13667929, rawDataSize=1737107057]

Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=37} stats: [numFiles=11, numRows=293147, totalSize=13981393, rawDataSize=770080563]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=38} stats: [numFiles=11, numRows=542079, totalSize=23320478, rawDataSize=1423461862]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=39} stats: [numFiles=11, numRows=1177, totalSize=106611, rawDataSize=3096767]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=4} stats: [numFiles=11, numRows=521, totalSize=62484, rawDataSize=1370453]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=40} stats: [numFiles=11, numRows=277184, totalSize=12670414, rawDataSize=723353903]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=41} stats: [numFiles=11, numRows=2621, totalSize=225013, rawDataSize=6835542]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=42} stats: [numFiles=11, numRows=32008, totalSize=1213864, rawDataSize=84372908]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=43} stats: [numFiles=11, numRows=174, totalSize=18185, rawDataSize=458810]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=44} stats: [numFiles=11, numRows=4, totalSize=11639, rawDataSize=10534]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=45} stats: [numFiles=11, numRows=6107, totalSize=377797, rawDataSize=15912553]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=46} stats: [numFiles=11, numRows=312327, totalSize=13605171, rawDataSize=816983670]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=47} stats: [numFiles=11, numRows=65440, totalSize=2091649, rawDataSize=171043973]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=48} stats: [numFiles=11, numRows=40987, totalSize=1709885, rawDataSize=106737398]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=49} stats: [numFiles=11, numRows=477, totalSize=69957, rawDataSize=1249536]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=5} stats: [numFiles=11, numRows=48081, totalSize=924205, rawDataSize=125443336]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=50} stats: [numFiles=11, numRows=53749, totalSize=2359855, rawDataSize=140473514]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=51} stats: [numFiles=11, numRows=32764, totalSize=1596057, rawDataSize=85309941]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=52} stats: [numFiles=11, numRows=1001, totalSize=88899, rawDataSize=2615593]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=53} stats: [numFiles=11, numRows=19488, totalSize=902126, rawDataSize=50807738]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=54} stats: [numFiles=11, numRows=3, totalSize=10922, rawDataSize=7856]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=55} stats: [numFiles=11, numRows=89, totalSize=49344, rawDataSize=233077]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=56} stats: [numFiles=11, numRows=367, totalSize=65716, rawDataSize=957407]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=57} stats: [numFiles=11, numRows=3, totalSize=7964, rawDataSize=7849]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=58} stats: [numFiles=11, numRows=13, totalSize=26887, rawDataSize=34080]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=59} stats: [numFiles=11, numRows=132, totalSize=46883, rawDataSize=345703]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=6} stats: [numFiles=11, numRows=192, totalSize=59933, rawDataSize=502364]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=60} stats: [numFiles=11, numRows=3691, totalSize=237723, rawDataSize=9626933]

Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=61} stats: [numFiles=11, numRows=5524, totalSize=333186, rawDataSize=14387080]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=62} stats: [numFiles=11, numRows=2810, totalSize=203023, rawDataSize=7347671]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=63} stats: [numFiles=11, numRows=250, totalSize=64939, rawDataSize=654616]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=64} stats: [numFiles=11, numRows=6764, totalSize=291788, rawDataSize=17730522]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=65} stats: [numFiles=11, numRows=26, totalSize=40409, rawDataSize=67926]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=66} stats: [numFiles=11, numRows=13142, totalSize=653788, rawDataSize=34327096]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=67} stats: [numFiles=11, numRows=7381, totalSize=501656, rawDataSize=19264055]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=68} stats: [numFiles=11, numRows=25036, totalSize=1065649, rawDataSize=65587094]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=69} stats: [numFiles=11, numRows=96914, totalSize=3377872, rawDataSize=254689369]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=7} stats: [numFiles=11, numRows=210176, totalSize=4049687, rawDataSize=551081440]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=70} stats: [numFiles=11, numRows=144646, totalSize=6346616, rawDataSize=379353999]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=71} stats: [numFiles=11, numRows=263392, totalSize=11282972, rawDataSize=691028847]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=72} stats: [numFiles=11, numRows=5519, totalSize=306512, rawDataSize=14426479]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=73} stats: [numFiles=11, numRows=2081, totalSize=146019, rawDataSize=5445783]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=74} stats: [numFiles=11, numRows=58939, totalSize=2815275, rawDataSize=154137828]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=75} stats: [numFiles=11, numRows=4345, totalSize=271763, rawDataSize=11385712]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=76} stats: [numFiles=11, numRows=18, totalSize=35326, rawDataSize=47086]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=77} stats: [numFiles=11, numRows=6081, totalSize=315896, rawDataSize=15934426]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=78} stats: [numFiles=11, numRows=26776, totalSize=1368674, rawDataSize=70161396]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=79} stats: [numFiles=11, numRows=5208, totalSize=239694, rawDataSize=13622026]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=8} stats: [numFiles=11, numRows=1405, totalSize=96721, rawDataSize=3671235]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=80} stats: [numFiles=11, numRows=2084, totalSize=170634, rawDataSize=5451902]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=81} stats: [numFiles=11, numRows=14, totalSize=12455, rawDataSize=36759]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=82} stats: [numFiles=11, numRows=17289, totalSize=739055, rawDataSize=45279310]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=83} stats: [numFiles=11, numRows=5111, totalSize=296395, rawDataSize=13371192]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=84} stats: [numFiles=11, numRows=40943, totalSize=1477771, rawDataSize=107346244]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=85} stats: [numFiles=11, numRows=9316, totalSize=487690, rawDataSize=24419734]

Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=86} stats: [numFiles=11, numRows=6, totalSize=18512, rawDataSize=15733]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=87} stats: [numFiles=11, numRows=1, totalSize=3975, rawDataSize=2614]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=88} stats: [numFiles=11, numRows=10, totalSize=29561, rawDataSize=26165]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=89} stats: [numFiles=11, numRows=2325, totalSize=90949, rawDataSize=6107724]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=9} stats: [numFiles=11, numRows=28685, totalSize=1045781, rawDataSize=75001208]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=90} stats: [numFiles=11, numRows=27, totalSize=43394, rawDataSize=70481]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=91} stats: [numFiles=11, numRows=433, totalSize=75645, rawDataSize=1131842]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=92} stats: [numFiles=11, numRows=20, totalSize=28374, rawDataSize=52318]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=93} stats: [numFiles=11, numRows=8, totalSize=16032, rawDataSize=20936]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=94} stats: [numFiles=11, numRows=199, totalSize=43584, rawDataSize=520909]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=95} stats: [numFiles=11, numRows=87, totalSize=46123, rawDataSize=226920]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=96} stats: [numFiles=11, numRows=41, totalSize=38193, rawDataSize=107198]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=97} stats: [numFiles=11, numRows=55, totalSize=48321, rawDataSize=143942]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=98} stats: [numFiles=11, numRows=23952, totalSize=1377234, rawDataSize=62438050]
Partition hive_assignment.nyc_data_partitioned_bucketed_orc{violation_code=99} stats: [numFiles=11, numRows=1439, totalSize=145750, rawDataSize=3754212]
MapReduce Jobs Launched:
Stage-Stage-1: Map: 8 Reduce: 9   Cumulative CPU: 845.68 sec   HDFS Read: 2087212846 HDFS Write: 221156249 **SUCCESS**
Total MapReduce CPU Time Spent: 14 minutes 5 seconds 680 msec
OK
Time taken: 964.082 seconds
hive>

Analysis I

1. Field summons_number is considered as ticket number. Based on study I found that this field is either is NULL (empty) or has 10 digits. I am taking count of summons_number where length of summons_number is 10. If I use just count function without any where clause I still get the same output but it takes more time as count function will still go through all records so filtering out invalid records using where clause first and it takes lesser time.

   Output is the total number of tickets for year 2017.

--- Count number of valid tickets

```
SELECT count(summons_number)
FROM nyc_data_partitioned_bucketed_orc
WHERE length(summons_number) = 10;
```

```
hive> SELECT count(summons_number)
    > FROM nyc_data_partitioned_bucketed_orc
    > WHERE length(summons_number) = 10;
Query ID = cloudera_20181118044545_71e2a5fd-9d6e-46e1-b4cf-271e4c7088c1
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_1542539368466_0021, Tracking URL =
http://quickstart.cloudera:8088/proxy/application_1542539368466_0021/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1542539368466_0021
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-11-18 04:45:16,246 Stage-1 map = 0%,  reduce = 0%
2018-11-18 04:45:31,861 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 13.01 sec
2018-11-18 04:45:39,162 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 14.97 sec
MapReduce Total cumulative CPU time: 14 seconds 970 msec
Ended Job = job_1542539368466_0021
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 14.97 sec   HDFS Read: 11784780 HDFS Write: 17 SUCCESS
Total MapReduce CPU Time Spent: 14 seconds 970 msec
OK
501916
Time taken: 32.134 seconds, Fetched: 1 row(s)
hive>
```

2. Based on study, considering if plate_id field contains any special character then should be ignored. Also, considering that valid field registration_state should not contain any digit. I am taking count of distinct registration_state field where length of summons_number is 10, plate_id not containing any special character and registration_state does not contain any digit. I am also showing distinct registration_state in the next statement based on same criteria as it is bit more elaborative.

   First query shows count of unique registration states of the cars which got parking tickets. And next query shows those unique registration states.

--- Count distinct states of cars which got parking tickets

```
SELECT count(DISTINCT registration_state)
FROM nyc_data_partitioned_bucketed_orc
WHERE length(summons_number) = 10
  AND registration_state rlike '[^0-9]'
  AND plate_id not rlike '[^a-zA-Z0-9]';
```

```
hive> SELECT count(DISTINCT registration_state)
    > FROM nyc_data_partitioned_bucketed_orc
    > WHERE length(summons_number) = 10
    >   AND registration_state rlike '[^0-9]'
    >   AND plate_id not rlike '[^a-zA-Z0-9]';
Query ID = cloudera_20181118045252_7a638caa-6e20-4baf-aee7-07cd63a09040
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_1542539368466_0022, Tracking URL =
http://quickstart.cloudera:8088/proxy/application_1542539368466_0022/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1542539368466_0022
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-11-18 04:52:08,038 Stage-1 map = 0%,  reduce = 0%
2018-11-18 04:52:24,698 Stage-1 map = 36%,  reduce = 0%, Cumulative CPU 14.64 sec
2018-11-18 04:52:31,013 Stage-1 map = 58%,  reduce = 0%, Cumulative CPU 20.88 sec
2018-11-18 04:52:34,137 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 23.93 sec
2018-11-18 04:52:41,596 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 27.06 sec
MapReduce Total cumulative CPU time: 27 seconds 60 msec
Ended Job = job_1542539368466_0022
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 27.06 sec   HDFS Read: 51239947 HDFS Write: 13 SUCCESS
Total MapReduce CPU Time Spent: 27 seconds 60 msec
OK
63
Time taken: 42.401 seconds, Fetched: 1 row(s)
hive>
```

--- Show distinct states of cars which got parking tickets

```
SELECT DISTINCT registration_state
FROM nyc_data_partitioned_bucketed_orc
WHERE length(summons_number) = 10
  AND registration_state rlike '[^0-9]'
  AND plate_id not rlike '[^a-zA-Z0-9]';
```

```
hive> SELECT DISTINCT registration_state
    > FROM nyc_data_partitioned_bucketed_orc
    > WHERE length(summons_number) = 10
    >   AND registration_state rlike '[^0-9]'
    >   AND plate_id not rlike '[^a-zA-Z0-9]';
Query ID = cloudera_20181118045353_acba5579-3c32-4036-aa65-a6c4c89e0a9a
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_1542539368466_0023, Tracking URL =
http://quickstart.cloudera:8088/proxy/application_1542539368466_0023/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1542539368466_0023
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-11-18 04:53:30,288 Stage-1 map = 0%,  reduce = 0%
2018-11-18 04:53:48,086 Stage-1 map = 32%,  reduce = 0%, Cumulative CPU 15.34 sec
2018-11-18 04:53:54,351 Stage-1 map = 57%,  reduce = 0%, Cumulative CPU 21.81 sec
2018-11-18 04:53:57,487 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 25.17 sec
2018-11-18 04:54:05,941 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 28.73 sec
MapReduce Total cumulative CPU time: 28 seconds 730 msec
Ended Job = job_1542539368466_0023
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 28.73 sec   HDFS Read: 51239313 HDFS Write: 201 SUCCESS
Total MapReduce CPU Time Spent: 28 seconds 730 msec
OK
AB
AK
AL
AR
AZ
BC
CA
CO
CT
DC
DE
DP
FL
GA
GV
HI
```

IA
ID
IL
IN
KS
KY
LA
MA
MB
MD
ME
MI
MN
MO
MS
MT
NB
NC
ND
NE
NH
NJ
NM
NS
NV
NY
OH
OK
ON
OR
PA
PE
PR
QB
RI
SC
SD
SK
TN
TX
UT
VA
VT
WA
WI
WV
WY

Time taken: 43.885 seconds, Fetched: 63 row(s)
hive>

3. Fields street_code1, street_code2 and street_code3 are being checked if NULL or 0, in order to check for absence of address. I am taking count of summons_number where length of summons_number is 10 and street_code1 or street_code2 or street_code3 IS NULL (empty) or 0.

   Output is the total number of valid ticket numbers where address is empty.

--- Count tickets where address is empty

```
SELECT count(summons_number)
FROM nyc_data_partitioned_bucketed_orc
WHERE length(summons_number) = 10
  AND (street_code1 IS NULL
      OR street_code2 IS NULL
      OR street_code3 IS NULL
      OR street_code1 == 0
      OR street_code2 == 0
      OR street_code3 == 0);
```

```
hive> SELECT count(summons_number)
    > FROM nyc_data_partitioned_bucketed_orc
    > WHERE length(summons_number) = 10
    >   AND (street_code1 IS NULL
    >       OR street_code2 IS NULL
    >       OR street_code3 IS NULL
    >       OR street_code1 == 0
    >       OR street_code2 == 0
    >       OR street_code3 == 0);
Query ID = cloudera_20181118045555_bed05c3f-b722-4317-be91-9f85c71d99c1
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_1542539368466_0024, Tracking URL =
http://quickstart.cloudera:8088/proxy/application_1542539368466_0024/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1542539368466_0024
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-11-18 04:55:10,800 Stage-1 map = 0%,  reduce = 0%
2018-11-18 04:55:28,398 Stage-1 map = 58%,  reduce = 0%, Cumulative CPU 14.09 sec
2018-11-18 04:55:30,490 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 17.24 sec
2018-11-18 04:55:38,901 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 20.7 sec
MapReduce Total cumulative CPU time: 20 seconds 700 msec
Ended Job = job_1542539368466_0024
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 20.7 sec   HDFS Read: 45294706 HDFS Write: 16 SUCCESS
Total MapReduce CPU Time Spent: 20 seconds 700 msec
OK
77367
Time taken: 35.553 seconds, Fetched: 1 row(s)
```

Analysis II

1. Based on study, considering violation_code with value 0 as invalid and so ignoring it. In inner query, I am selecting violation_code and count of same where violation_code != 0 and grouping by violation_code. This intermediate result set then used in outer query to order by on count of violation_code in descending order and then limit the output to top 5.

   Output result set contains top 5 violation_code along with frequency of occurrence in descending order.

--- Top 5 violation codes and frequency of occurrence

```
SELECT cnt.violation_code,
     cnt.cnt_violation_code
FROM
  (SELECT violation_code,
       count(violation_code) AS cnt_violation_code
   FROM nyc_data_partitioned_bucketed_orc
   WHERE violation_code != 0
   GROUP BY violation_code) cnt
ORDER BY cnt.cnt_violation_code DESC
LIMIT 5;
```

```
hive> SELECT cnt.violation_code,
    >       cnt.cnt_violation_code
    > FROM
    >   (SELECT violation_code,
    >         count(violation_code) AS cnt_violation_code
    >    FROM nyc_data_partitioned_bucketed_orc
    >    WHERE violation_code != 0
    >    GROUP BY violation_code) cnt
    > ORDER BY cnt.cnt_violation_code DESC
    > LIMIT 5;
Query ID = cloudera_20181118045656_d34ed11e-a2ad-4873-be09-bbe55af09acf
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_1542539368466_0025, Tracking URL =
http://quickstart.cloudera:8088/proxy/application_1542539368466_0025/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1542539368466_0025
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-11-18 04:56:27,754 Stage-1 map = 0%,  reduce = 0%
2018-11-18 04:56:44,437 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 14.64 sec
2018-11-18 04:56:52,787 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 16.42 sec
```

```
MapReduce Total cumulative CPU time: 16 seconds 420 msec
Ended Job = job_1542539368466_0025
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_1542539368466_0026, Tracking URL =
http://quickstart.cloudera:8088/proxy/application_1542539368466_0026/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1542539368466_0026
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2018-11-18 04:57:02,086 Stage-2 map = 0%,  reduce = 0%
2018-11-18 04:57:08,522 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 2.68 sec
2018-11-18 04:57:15,832 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 4.6 sec
MapReduce Total cumulative CPU time: 4 seconds 600 msec
Ended Job = job_1542539368466_0026
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 16.42 sec   HDFS Read: 9650695 HDFS Write: 2151 SUCCESS
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 4.6 sec   HDFS Read: 7272 HDFS Write: 60 SUCCESS
Total MapReduce CPU Time Spent: 21 seconds 20 msec
OK
21      768082
36      662765
38      542079
14      476660
20      319646
Time taken: 56.635 seconds, Fetched: 5 row(s)
hive>
```

2. **(a)** Based on study, considering vehicle_body_type with value '00' as invalid and so ignoring it. Also considering only those records where vehicle_body_type does not start with a special character and does not end with a special character. In inner query, I am selecting vehicle_body_type and count of summons_number where length of summons_number is 10 and vehicle_body_type !='00' and vehicle_body_type should not start with a special character and should not end with a special character and group by vehicle_body_type. This intermediate result set is then used in outer query to order by on count of summons_number in descending order and then limit the output to top 5.

   Output result set contains top 5 vehicle_body_type along with count of summons_number in descending order.

--- Top 5 vehicle body type and frequency of tickets

```
SELECT cnt.vehicle_body_type,
     cnt.cnt_summons_number
FROM
  (SELECT vehicle_body_type,
        count(summons_number) AS cnt_summons_number
   FROM nyc_data_partitioned_bucketed_orc
   WHERE length(summons_number) = 10
    AND vehicle_body_type != '00'
```

```
    AND vehicle_body_type rlike '^[a-zA-Z0-9]'
    AND vehicle_body_type rlike '[a-zA-Z0-9]$'
  GROUP BY vehicle_body_type) cnt
ORDER BY cnt.cnt_summons_number DESC
LIMIT 5;

hive> SELECT cnt.vehicle_body_type,
    >       cnt.cnt_summons_number
    > FROM
    >   (SELECT vehicle_body_type,
    >           count(summons_number) AS cnt_summons_number
    >    FROM nyc_data_partitioned_bucketed_orc
    >    WHERE length(summons_number) = 10
    >      AND vehicle_body_type != '00'
    >      AND vehicle_body_type rlike '^[a-zA-Z0-9]'
    >      AND vehicle_body_type rlike '[a-zA-Z0-9]$'
    >    GROUP BY vehicle_body_type) cnt
    > ORDER BY cnt.cnt_summons_number DESC
    > LIMIT 5;
Query ID = cloudera_20181118045858_8b449cd7-70e6-44a9-aa1b-a1c5d8702a90
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_1542539368466_0027, Tracking URL =
http://quickstart.cloudera:8088/proxy/application_1542539368466_0027/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1542539368466_0027
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-11-18 04:58:21,092 Stage-1 map = 0%,  reduce = 0%
2018-11-18 04:58:38,846 Stage-1 map = 36%,  reduce = 0%, Cumulative CPU 15.38 sec
2018-11-18 04:58:45,136 Stage-1 map = 60%,  reduce = 0%, Cumulative CPU 21.99 sec
2018-11-18 04:58:48,308 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 25.94 sec
2018-11-18 04:58:56,745 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 28.83 sec
MapReduce Total cumulative CPU time: 28 seconds 830 msec
Ended Job = job_1542539368466_0027
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_1542539368466_0028, Tracking URL =
http://quickstart.cloudera:8088/proxy/application_1542539368466_0028/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1542539368466_0028
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2018-11-18 04:59:07,744 Stage-2 map = 0%,  reduce = 0%
2018-11-18 04:59:15,182 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 2.4 sec
```

2018-11-18 04:59:22,562 Stage-2 map = 100%, reduce = 100%, Cumulative CPU 5.43 sec
MapReduce Total cumulative CPU time: 5 seconds 430 msec
Ended Job = job_1542539368466_0028
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1   Cumulative CPU: 28.83 sec   HDFS Read: 15553592 HDFS Write: 21962 SUCCESS
Stage-Stage-2: Map: 1 Reduce: 1   Cumulative CPU: 5.43 sec   HDFS Read: 27113 HDFS Write: 63 SUCCESS
Total MapReduce CPU Time Spent: 34 seconds 260 msec
OK
SDN 182773
SUBN 148334
VAN 60943
DELV 47816
P-U 9072
Time taken: 70.619 seconds, Fetched: 5 row(s)
hive>

2. **(b)** Based on study, considering only those records where vehicle_make does not start with a special character and does not end with a special character. In inner query, I am selecting vehicle_make and count of summons_number where length of summons_number is 10 and vehicle_make should not start with a special character and should not end with a special character and group by vehicle_make. This intermediate result set is then used in outer query to order by on count of summons_number in descending order and then limit the output to top 5.

   Output result set contains top 5 vehicle_make along with count of summons_number in descending order.

--- Top 5 vehicle make and frequency of tickets

```
SELECT cnt.vehicle_make,
     cnt.cnt_summons_number
FROM
  (SELECT vehicle_make,
        count(summons_number) AS cnt_summons_number
   FROM nyc_data_partitioned_bucketed_orc
   WHERE length(summons_number) = 10
    AND vehicle_make rlike '^[a-zA-Z0-9]'
    AND vehicle_make rlike '[a-zA-Z0-9]$'
   GROUP BY vehicle_make) cnt
ORDER BY cnt.cnt_summons_number DESC
LIMIT 5;
```

```
hive> SELECT cnt.vehicle_make,
    >       cnt.cnt_summons_number
    > FROM
    >   (SELECT vehicle_make,
    >         count(summons_number) AS cnt_summons_number
    >    FROM nyc_data_partitioned_bucketed_orc
    >    WHERE length(summons_number) = 10
    >      AND vehicle_make rlike '^[a-zA-Z0-9]'
    >      AND vehicle_make rlike '[a-zA-Z0-9]$'
    >    GROUP BY vehicle_make) cnt
    > ORDER BY cnt.cnt_summons_number DESC
    > LIMIT 5;
```

```
Query ID = cloudera_20181118050000_4240839b-7bc0-499d-95f1-1b6e8bd4e17b
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_1542539368466_0029, Tracking URL =
http://quickstart.cloudera:8088/proxy/application_1542539368466_0029/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1542539368466_0029
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-11-18 05:00:18,321 Stage-1 map = 0%,  reduce = 0%
2018-11-18 05:00:36,036 Stage-1 map = 36%,  reduce = 0%, Cumulative CPU 15.59 sec
2018-11-18 05:00:42,352 Stage-1 map = 60%,  reduce = 0%, Cumulative CPU 22.0 sec
2018-11-18 05:00:45,522 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 26.32 sec
2018-11-18 05:00:54,997 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 30.21 sec
MapReduce Total cumulative CPU time: 30 seconds 210 msec
Ended Job = job_1542539368466_0029
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_1542539368466_0030, Tracking URL =
http://quickstart.cloudera:8088/proxy/application_1542539368466_0030/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1542539368466_0030
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2018-11-18 05:01:03,596 Stage-2 map = 0%,  reduce = 0%
2018-11-18 05:01:12,088 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 3.23 sec
2018-11-18 05:01:19,506 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 6.54 sec
MapReduce Total cumulative CPU time: 6 seconds 540 msec
Ended Job = job_1542539368466_0030
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 30.21 sec   HDFS Read: 17460488 HDFS Write: 64230 SUCCESS
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 6.54 sec   HDFS Read: 69361 HDFS Write: 69 SUCCESS
Total MapReduce CPU Time Spent: 36 seconds 750 msec
OK
FORD    53667
TOYOT   53549
HONDA   48689
NISSA   40888
CHEVR   27128
Time taken: 70.751 seconds, Fetched: 5 row(s)
hive>
```

3. **(a)** Based on study, considering violation_precinct with value 0 as invalid and so ignoring it. In inner query, I am selecting violation_precinct and count of summons_number where length of summons_number is 10 and violation_precinct !=0 and group by violation_precinct. This intermediate result set is then used in outer query to order by on count of summons_number in descending order and then limit the output to top 5.

   Output result set contains top 5 violation_precinct along with count of summons_number in descending order.

--- Top 5 violation precinct and frequency of tickets

```
SELECT cnt.violation_precinct,
     cnt.cnt_summons_number
FROM
  (SELECT violation_precinct,
       count(summons_number) AS cnt_summons_number
   FROM nyc_data_partitioned_bucketed_orc
   WHERE length(summons_number) = 10
    AND violation_precinct != 0
   GROUP BY violation_precinct) cnt
ORDER BY cnt.cnt_summons_number DESC
LIMIT 5;
```

```
hive> SELECT cnt.violation_precinct,
    >       cnt.cnt_summons_number
    > FROM
    >   (SELECT violation_precinct,
    >         count(summons_number) AS cnt_summons_number
    >    FROM nyc_data_partitioned_bucketed_orc
    >    WHERE length(summons_number) = 10
    >     AND violation_precinct != 0
    >    GROUP BY violation_precinct) cnt
    > ORDER BY cnt.cnt_summons_number DESC
    > LIMIT 5;
Query ID = cloudera_20181118050202_720c1c6a-9d22-4a6c-9447-9144bf1d1ff8
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_1542539368466_0031, Tracking URL =
http://quickstart.cloudera:8088/proxy/application_1542539368466_0031/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1542539368466_0031
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-11-18 05:02:51,882 Stage-1 map = 0%,  reduce = 0%
2018-11-18 05:03:08,628 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 13.83 sec
2018-11-18 05:03:17,059 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 16.92 sec
MapReduce Total cumulative CPU time: 16 seconds 920 msec
Ended Job = job_1542539368466_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_1542539368466_0032, Tracking URL =
http://quickstart.cloudera:8088/proxy/application_1542539368466_0032/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1542539368466_0032
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2018-11-18 05:03:26,531 Stage-2 map = 0%,  reduce = 0%
2018-11-18 05:03:33,899 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 2.42 sec
2018-11-18 05:03:42,358 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 5.77 sec
MapReduce Total cumulative CPU time: 5 seconds 770 msec
Ended Job = job_1542539368466_0032
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 16.92 sec   HDFS Read: 16244233 HDFS Write: 3568 SUCCESS
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 5.77 sec   HDFS Read: 8705 HDFS Write: 54 SUCCESS
Total MapReduce CPU Time Spent: 22 seconds 690 msec
OK
19      15865
18      14720
70      14402
72      13880
1       13778
Time taken: 58.583 seconds, Fetched: 5 row(s)
hive>
```

3. **(b)** Based on study, considering issuer_precinct with value 0 as invalid and so ignoring it. In inner query, I am selecting issuer_precinct and count of summons_number where length of summons_number is 10 and issuer_precinct !=0 and group by issuer_precinct. This intermediate result set is then used in outer query to order by on count of summons_number in descending order and then limit the output to top 5.

   Output result set contains top 5 issuer_precinct along with count of summons_number in descending order.

--- Top 5 issuer precinct and frequency of tickets

```
SELECT cnt.issuer_precinct,
     cnt.cnt_summons_number
FROM
  (SELECT issuer_precinct,
        count(summons_number) AS cnt_summons_number
   FROM nyc_data_partitioned_bucketed_orc
   WHERE length(summons_number) = 10
     AND issuer_precinct != 0
   GROUP BY issuer_precinct) cnt
ORDER BY cnt.cnt_summons_number DESC
LIMIT 5;
```

```
hive> SELECT cnt.issuer_precinct,
```

```
    >       cnt.cnt_summons_number
    > FROM
    >   (SELECT issuer_precinct,
    >           count(summons_number) AS cnt_summons_number
    >    FROM nyc_data_partitioned_bucketed_orc
    >    WHERE length(summons_number) = 10
    >      AND issuer_precinct != 0
    >    GROUP BY issuer_precinct) cnt
    > ORDER BY cnt.cnt_summons_number DESC
    > LIMIT 5;
Query ID = cloudera_20181118050404_c9c496ac-5028-40e5-baba-2152c81b1803
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_1542539368466_0033, Tracking URL =
http://quickstart.cloudera:8088/proxy/application_1542539368466_0033/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1542539368466_0033
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-11-18 05:04:24,427 Stage-1 map = 0%,  reduce = 0%
2018-11-18 05:04:41,246 Stage-1 map = 66%,  reduce = 0%, Cumulative CPU 15.82 sec
2018-11-18 05:04:42,339 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 17.07 sec
2018-11-18 05:04:50,746 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 19.82 sec
MapReduce Total cumulative CPU time: 19 seconds 820 msec
Ended Job = job_1542539368466_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_1542539368466_0034, Tracking URL =
http://quickstart.cloudera:8088/proxy/application_1542539368466_0034/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1542539368466_0034
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2018-11-18 05:05:01,833 Stage-2 map = 0%,  reduce = 0%
2018-11-18 05:05:09,207 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 2.9 sec
2018-11-18 05:05:18,683 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 6.43 sec
MapReduce Total cumulative CPU time: 6 seconds 430 msec
Ended Job = job_1542539368466_0034
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 19.82 sec   HDFS Read: 16213000 HDFS Write: 10764 SUCCESS
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 6.43 sec   HDFS Read: 15889 HDFS Write: 56 SUCCESS
Total MapReduce CPU Time Spent: 26 seconds 250 msec
OK
110     12216
109     10268
```

70 10148
401 9907
34 9446
Time taken: 62.761 seconds, Fetched: 5 row(s)
hive>

4. **I am showing 2 queries. 1st to show top most occurring violation_code in top 3 issuer_precinct with most number of tickets and 2nd to show top 5 most occurring violation_code within top 3 issuer_precicnt with most number of tickets.**

4. **(a)** Using With clause, I am first preparing an intermediate result set which contains top 3 issuer_precinct in terms of highest number of tickets issued. In inner query of With clause, I am selecting issuer_precinct and count of summons_number where length of summons_number is 10 and issuer_precinct != 0 and group by issuer_precinct. This intermediate result set is used in outer query of With clause to order by on count of summons_number in descending order and limit output to top 3. This result set contains top 3 issuer_precinct along with count of summons_number. This subquery output is then joined with main query where I am selecting issuer_precinct, violation_code and taking count of violation_code where issuer_precinct matches top 3 issuer_precinct and violation_code != 0 and group by issuer_precinct and violation_code. This output is then considered as subquery for outer query where I am selecting issuer_precinct, violation_code, count of violation_code and highest count of violation of code among issuer_precinct by using partition by clause where I am partitioning by issuer_precinct and using max function on count of violation_code, getting out highest count. This output is then used subquery for final outer query where I am selecting issuer_precinct and violation_code where count of violation_code is equal to highest(maximum) count of violation_code.

   Output contains top 3 issuer_precinct which have issued most number of tickets, most occurring violation_code in same issuer_precinct and frequency of most occurring violation_code.

--- Top 3 issuer precinct with most number of tickets and top most occurring violation code

```
WITH precinct AS
  (SELECT cnt.issuer_precinct,
          cnt.cnt_summons_number
   FROM
     (SELECT issuer_precinct,
             count(summons_number) AS cnt_summons_number
      FROM nyc_data_partitioned_bucketed_orc
      WHERE length(summons_number) = 10
        AND issuer_precinct != 0
      GROUP BY issuer_precinct) cnt
   ORDER BY cnt.cnt_summons_number DESC
   LIMIT 3)
SELECT cnt_max_code.issuer_precinct,
       cnt_max_code.violation_code,
       cnt_max_code.max_cnt_violation_code
FROM
  (SELECT cnt_code.issuer_precinct,
          cnt_code.violation_code,
          cnt_code.cnt_violation_code,
          max(cnt_code.cnt_violation_code) OVER (PARTITION BY cnt_code.issuer_precinct) AS max_cnt_violation_code
   FROM
     (SELECT p.issuer_precinct,
```

```
        ndp.violation_code,
        count(ndp.violation_code) AS cnt_violation_code
    FROM nyc_data_partitioned_bucketed_orc ndp
    JOIN precinct p ON p.issuer_precinct = ndp.issuer_precinct
    WHERE ndp.violation_code != 0
    GROUP BY p.issuer_precinct,
          ndp.violation_code) cnt_code)cnt_max_code
WHERE cnt_max_code.max_cnt_violation_code = cnt_max_code.cnt_violation_code;

hive> WITH precinct AS
   >  (SELECT cnt.issuer_precinct,
   >         cnt.cnt_summons_number
   >   FROM
   >    (SELECT issuer_precinct,
   >            count(summons_number) AS cnt_summons_number
   >     FROM nyc_data_partitioned_bucketed_orc
   >     WHERE length(summons_number) = 10
   >       AND issuer_precinct != 0
   >     GROUP BY issuer_precinct) cnt
   >   ORDER BY cnt.cnt_summons_number DESC
   >   LIMIT 3)
   > SELECT cnt_max_code.issuer_precinct,
   >      cnt_max_code.violation_code,
   >      cnt_max_code.max_cnt_violation_code
   > FROM
   >  (SELECT cnt_code.issuer_precinct,
   >         cnt_code.violation_code,
   >         cnt_code.cnt_violation_code,
   >         max(cnt_code.cnt_violation_code) OVER (PARTITION BY cnt_code.issuer_precinct) AS max_cnt_violation_code
   >   FROM
   >    (SELECT p.issuer_precinct,
   >            ndp.violation_code,
   >            count(ndp.violation_code) AS cnt_violation_code
   >     FROM nyc_data_partitioned_bucketed_orc ndp
   >     JOIN precinct p ON p.issuer_precinct = ndp.issuer_precinct
   >     WHERE ndp.violation_code != 0
   >     GROUP BY p.issuer_precinct,
   >              ndp.violation_code) cnt_code)cnt_max_code
   > WHERE cnt_max_code.max_cnt_violation_code = cnt_max_code.cnt_violation_code;
Query ID = cloudera_20181118050606_3d84526f-6c7a-4f44-804d-7b26f3de0073
Total jobs = 7
Launching Job 1 out of 7
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_1542539368466_0035, Tracking URL =
http://quickstart.cloudera:8088/proxy/application_1542539368466_0035/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1542539368466_0035
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-11-18 05:06:12,149 Stage-1 map = 0%,  reduce = 0%
```

```
2018-11-18 05:06:28,954 Stage-1 map = 65%,  reduce = 0%, Cumulative CPU 16.07 sec
2018-11-18 05:06:31,035 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 17.87 sec
2018-11-18 05:06:40,503 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 20.89 sec
MapReduce Total cumulative CPU time: 20 seconds 890 msec
Ended Job = job_1542539368466_0035
Launching Job 2 out of 7
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_1542539368466_0036, Tracking URL =
http://quickstart.cloudera:8088/proxy/application_1542539368466_0036/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1542539368466_0036
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2018-11-18 05:06:50,287 Stage-2 map = 0%,  reduce = 0%
2018-11-18 05:06:57,658 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 2.56 sec
2018-11-18 05:07:08,291 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 6.69 sec
MapReduce Total cumulative CPU time: 6 seconds 690 msec
Ended Job = job_1542539368466_0036
Stage-10 is selected by condition resolver.
Stage-11 is filtered out by condition resolver.
Stage-3 is filtered out by condition resolver.
Execution log at: /tmp/cloudera/cloudera_20181118050606_3d84526f-6c7a-4f44-804d-7b26f3de0073.log
2018-11-18 05:07:15     Starting to launch local task to process map join;          maximum memory = 932184064
2018-11-18 05:07:16     Dump the side-table for tag: 1 with group count: 3 into file: file:/tmp/cloudera/cf173f5b-937c-4f46-
8126-45ec508c5e36/hive_2018-11-18_05-06-02_867_3326036539492447971-1/-local-10007/HashTable-Stage-7/MapJoin-
mapfile21--.hashtable
2018-11-18 05:07:17     Uploaded 1 File to: file:/tmp/cloudera/cf173f5b-937c-4f46-8126-45ec508c5e36/hive_2018-11-
18_05-06-02_867_3326036539492447971-1/-local-10007/HashTable-Stage-7/MapJoin-mapfile21--.hashtable (314 bytes)
2018-11-18 05:07:17     End of local task; Time Taken: 1.195 sec.
Execution completed successfully
MapredLocal task succeeded
Launching Job 4 out of 7
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1542539368466_0037, Tracking URL =
http://quickstart.cloudera:8088/proxy/application_1542539368466_0037/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1542539368466_0037
Hadoop job information for Stage-7: number of mappers: 1; number of reducers: 0
2018-11-18 05:07:28,330 Stage-7 map = 0%,  reduce = 0%
2018-11-18 05:07:47,338 Stage-7 map = 98%,  reduce = 0%, Cumulative CPU 16.76 sec
2018-11-18 05:07:48,380 Stage-7 map = 100%,  reduce = 0%, Cumulative CPU 19.24 sec
MapReduce Total cumulative CPU time: 19 seconds 240 msec
Ended Job = job_1542539368466_0037
Launching Job 5 out of 7
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_1542539368466_0038, Tracking URL =
http://quickstart.cloudera:8088/proxy/application_1542539368466_0038/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1542539368466_0038
Hadoop job information for Stage-4: number of mappers: 1; number of reducers: 1
2018-11-18 05:07:58,486 Stage-4 map = 0%,  reduce = 0%
2018-11-18 05:08:06,933 Stage-4 map = 100%,  reduce = 0%, Cumulative CPU 1.84 sec
2018-11-18 05:08:15,365 Stage-4 map = 100%,  reduce = 100%, Cumulative CPU 4.71 sec
MapReduce Total cumulative CPU time: 4 seconds 710 msec
Ended Job = job_1542539368466_0038
Launching Job 6 out of 7
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_1542539368466_0039, Tracking URL =
http://quickstart.cloudera:8088/proxy/application_1542539368466_0039/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1542539368466_0039
Hadoop job information for Stage-5: number of mappers: 1; number of reducers: 1
2018-11-18 05:08:26,278 Stage-5 map = 0%,  reduce = 0%
2018-11-18 05:08:33,692 Stage-5 map = 100%,  reduce = 0%, Cumulative CPU 2.17 sec
2018-11-18 05:08:42,080 Stage-5 map = 100%,  reduce = 100%, Cumulative CPU 5.89 sec
MapReduce Total cumulative CPU time: 5 seconds 890 msec
Ended Job = job_1542539368466_0039
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 20.89 sec   HDFS Read: 16213024 HDFS Write: 10764 SUCCESS
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 6.69 sec   HDFS Read: 15271 HDFS Write: 150 SUCCESS
Stage-Stage-7: Map: 1   Cumulative CPU: 19.24 sec   HDFS Read: 14082909 HDFS Write: 4197 SUCCESS
Stage-Stage-4: Map: 1  Reduce: 1   Cumulative CPU: 4.71 sec   HDFS Read: 8622 HDFS Write: 4197 SUCCESS
Stage-Stage-5: Map: 1  Reduce: 1   Cumulative CPU: 5.89 sec   HDFS Read: 12240 HDFS Write: 48 SUCCESS
Total MapReduce CPU Time Spent: 57 seconds 420 msec
OK
70      21      21935
109     38      16425
110     21      13840
Time taken: 160.306 seconds, Fetched: 3 row(s)
hive>
```

4. **(b)** Using With clause, I am first preparing an intermediate result set which contains top 3 issuer_precinct in terms of highest number of tickets issued. In inner query of With clause, I am selecting issuer_precinct and count of summons_number where length of summons_number is 10 and issuer_precinct != 0 and group by issuer_precinct. This intermediate result set is used in outer query of With clause to order by on count of summons_number in descending order and limit output to top 3. This result set contains top 3 issuer_precinct along with count of summons_number. This subquery output is then joined with main query where I am selecting issuer_precinct, violation_code and taking count of violation_code where issuer_precinct matches top 3 issuer_precinct and violation_code != 0 and group by issuer_precinct and violation_code. This output is then considered as subquery for outer query where I am selecting issuer_precinct, violation_code, count of violation_code and rank of count of violation of code among issuer_precinct by using partition by clause where I am partitioning by issuer_precinct, ordering by count of violation_code in descending order and using rank function on count of violation_code, getting out

rank. This output is then used subquery for final outer query where I am selecting issuer_precinct and violation_code and count of violation_code where rank is between 1 and 5.

Output contains top 3 issuer_precinct which have issued most number of tickets, top 5 most occurring violation_code in same issuer_precinct along with frequency.

--- Top 3 issuer precinct with most number of tickets and top 5 most occuring violation codes

```
WITH precinct AS
 (SELECT cnt.issuer_precinct,
       cnt.cnt_summons_number
  FROM
   (SELECT issuer_precinct,
         count(summons_number) AS cnt_summons_number
    FROM nyc_data_partitioned_bucketed_orc
    WHERE length(summons_number) = 10
     AND issuer_precinct != 0
    GROUP BY issuer_precinct) cnt
  ORDER BY cnt.cnt_summons_number DESC
  LIMIT 3)
SELECT cnt_max_code.issuer_precinct,
     cnt_max_code.violation_code,
     cnt_max_code.cnt_violation_code
FROM
 (SELECT cnt_code.issuer_precinct,
       cnt_code.violation_code,
       cnt_code.cnt_violation_code,
       rank() OVER (PARTITION BY cnt_code.issuer_precinct
                ORDER BY cnt_code.cnt_violation_code DESC) AS rank_cnt_violation_code
  FROM
   (SELECT p.issuer_precinct,
         ndp.violation_code,
         count(ndp.violation_code) AS cnt_violation_code
    FROM nyc_data_partitioned_bucketed_orc ndp
    JOIN precinct p ON p.issuer_precinct = ndp.issuer_precinct
    WHERE ndp.violation_code != 0
    GROUP BY p.issuer_precinct,
          ndp.violation_code) cnt_code)cnt_max_code
WHERE cnt_max_code.rank_cnt_violation_code BETWEEN 1 AND 5;
```

```
hive> WITH precinct AS
   >  (SELECT cnt.issuer_precinct,
   >        cnt.cnt_summons_number
   >   FROM
   >    (SELECT issuer_precinct,
   >          count(summons_number) AS cnt_summons_number
   >     FROM nyc_data_partitioned_bucketed_orc
   >     WHERE length(summons_number) = 10
   >      AND issuer_precinct != 0
   >     GROUP BY issuer_precinct) cnt
   >   ORDER BY cnt.cnt_summons_number DESC
   >   LIMIT 3)
   > SELECT cnt_max_code.issuer_precinct,
```

```
   >       cnt_max_code.violation_code,
   >       cnt_max_code.cnt_violation_code
   > FROM
   >   (SELECT cnt_code.issuer_precinct,
   >         cnt_code.violation_code,
   >         cnt_code.cnt_violation_code,
   >         rank() OVER (PARTITION BY cnt_code.issuer_precinct
   >                   ORDER BY cnt_code.cnt_violation_code DESC) AS rank_cnt_violation_code
   >    FROM
   >     (SELECT p.issuer_precinct,
   >           ndp.violation_code,
   >           count(ndp.violation_code) AS cnt_violation_code
   >      FROM nyc_data_partitioned_bucketed_orc ndp
   >      JOIN precinct p ON p.issuer_precinct = ndp.issuer_precinct
   >      WHERE ndp.violation_code != 0
   >      GROUP BY p.issuer_precinct,
   >            ndp.violation_code) cnt_code)cnt_max_code
   > WHERE cnt_max_code.rank_cnt_violation_code BETWEEN 1 AND 5;
Query ID = cloudera_20181118050909_e7ab61e0-4b9e-49d3-b95f-10e241d7339a
Total jobs = 7
Launching Job 1 out of 7
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_1542539368466_0040, Tracking URL =
http://quickstart.cloudera:8088/proxy/application_1542539368466_0040/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1542539368466_0040
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-11-18 05:09:58,762 Stage-1 map = 0%,  reduce = 0%
2018-11-18 05:10:16,490 Stage-1 map = 66%,  reduce = 0%, Cumulative CPU 15.65 sec
2018-11-18 05:10:17,539 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 16.06 sec
2018-11-18 05:10:25,988 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 19.82 sec
MapReduce Total cumulative CPU time: 19 seconds 820 msec
Ended Job = job_1542539368466_0040
Launching Job 2 out of 7
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_1542539368466_0041, Tracking URL =
http://quickstart.cloudera:8088/proxy/application_1542539368466_0041/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1542539368466_0041
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2018-11-18 05:10:36,519 Stage-2 map = 0%,  reduce = 0%
2018-11-18 05:10:42,828 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 2.5 sec
2018-11-18 05:10:52,290 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 6.17 sec
MapReduce Total cumulative CPU time: 6 seconds 170 msec
```

```
Ended Job = job_1542539368466_0041
Stage-10 is selected by condition resolver.
Stage-11 is filtered out by condition resolver.
Stage-3 is filtered out by condition resolver.
Execution log at: /tmp/cloudera/cloudera_20181118050909_e7ab61e0-4b9e-49d3-b95f-10e241d7339a.log
2018-11-18 05:10:57     Starting to launch local task to process map join;          maximum memory = 932184064
2018-11-18 05:10:59     Dump the side-table for tag: 1 with group count: 3 into file: file:/tmp/cloudera/cf173f5b-937c-4f46-8126-45ec508c5e36/hive_2018-11-18_05-09-50_041_8158809044064905987-1/-local-10007/HashTable-Stage-7/MapJoin-mapfile41--.hashtable
2018-11-18 05:10:59     Uploaded 1 File to: file:/tmp/cloudera/cf173f5b-937c-4f46-8126-45ec508c5e36/hive_2018-11-18_05-09-50_041_8158809044064905987-1/-local-10007/HashTable-Stage-7/MapJoin-mapfile41--.hashtable (314 bytes)
2018-11-18 05:10:59     End of local task; Time Taken: 1.71 sec.
Execution completed successfully
MapredLocal task succeeded
Launching Job 4 out of 7
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1542539368466_0042, Tracking URL =
http://quickstart.cloudera:8088/proxy/application_1542539368466_0042/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1542539368466_0042
Hadoop job information for Stage-7: number of mappers: 1; number of reducers: 0
2018-11-18 05:11:09,481 Stage-7 map = 0%,  reduce = 0%
2018-11-18 05:11:26,265 Stage-7 map = 99%,  reduce = 0%, Cumulative CPU 16.4 sec
2018-11-18 05:11:27,348 Stage-7 map = 100%,  reduce = 0%, Cumulative CPU 18.47 sec
MapReduce Total cumulative CPU time: 18 seconds 470 msec
Ended Job = job_1542539368466_0042
Launching Job 5 out of 7
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_1542539368466_0043, Tracking URL =
http://quickstart.cloudera:8088/proxy/application_1542539368466_0043/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1542539368466_0043
Hadoop job information for Stage-4: number of mappers: 1; number of reducers: 1
2018-11-18 05:11:37,503 Stage-4 map = 0%,  reduce = 0%
2018-11-18 05:11:44,857 Stage-4 map = 100%,  reduce = 0%, Cumulative CPU 1.72 sec
2018-11-18 05:11:53,279 Stage-4 map = 100%,  reduce = 100%, Cumulative CPU 4.76 sec
MapReduce Total cumulative CPU time: 4 seconds 760 msec
Ended Job = job_1542539368466_0043
Launching Job 6 out of 7
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_1542539368466_0044, Tracking URL =
http://quickstart.cloudera:8088/proxy/application_1542539368466_0044/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1542539368466_0044
Hadoop job information for Stage-5: number of mappers: 1; number of reducers: 1
```

2018-11-18 05:12:02,829 Stage-5 map = 0%, reduce = 0%
2018-11-18 05:12:10,188 Stage-5 map = 100%, reduce = 0%, Cumulative CPU 2.23 sec
2018-11-18 05:12:19,699 Stage-5 map = 100%, reduce = 100%, Cumulative CPU 6.17 sec
MapReduce Total cumulative CPU time: 6 seconds 170 msec
Ended Job = job_1542539368466_0044
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1   Cumulative CPU: 19.82 sec   HDFS Read: 16213024 HDFS Write: 10764 SUCCESS
Stage-Stage-2: Map: 1 Reduce: 1   Cumulative CPU: 6.17 sec   HDFS Read: 15271 HDFS Write: 150 SUCCESS
Stage-Stage-7: Map: 1   Cumulative CPU: 18.47 sec   HDFS Read: 14082909 HDFS Write: 4197 SUCCESS
Stage-Stage-4: Map: 1 Reduce: 1   Cumulative CPU: 4.76 sec   HDFS Read: 8622 HDFS Write: 4197 SUCCESS
Stage-Stage-5: Map: 1 Reduce: 1   Cumulative CPU: 6.17 sec   HDFS Read: 12472 HDFS Write: 180 SUCCESS
Total MapReduce CPU Time Spent: 55 seconds 390 msec
OK

70	**21**	**21935**
70	**38**	**20133**
70	**20**	**7791**
70	**37**	**6547**
70	**71**	**6460**
109	**38**	**16425**
109	**21**	**14696**
109	**14**	**12939**
109	**20**	**11056**
109	**37**	**10051**
110	**21**	**13840**
110	**38**	**9718**
110	**20**	**7141**
110	**37**	**5936**
110	**40**	**5245**
110	**14**	**5245**

Time taken: 151.78 seconds, Fetched: 16 row(s)
hive>

5. **I am showing two queries to show different properties. 1st to show count of violation_code and 2nd to show count of tickets.**

5. **(a)** Based on study, considering that violation_time should not be empty and should contain 4 numbers and should end with either ‘A’ or ‘P’, considering violation_time format as HHMM (A/P) format. In inner query, I am selecting violation_time and converting violation_time using a CASE construct (when violation_time ends with ‘P’, casting first 2 characters as int and checking if equal to 12, if so then considering it as casting first 2 characters as int and next, when violation_time ends with ‘P’, casting first 2 characters as int and adding 12 in it and checking if between 13 and 23, if so then considering it as casting first 2 characters as int and adding 12 in it and next, when violation_time ends with ‘A’, casting first 2 characters as int and checking if between 0 and 11, if so then considering it as casting first 2 characters as int then in else part, considering everything else as 24) and naming this field as v_time and selecting violation_code where violation_code != 0 and violation_time != ‘’ and violation_time ends with ‘A’ or ‘P’ and violation_time should not have any character other than numbers or ‘A’ or ‘P’. In outer query, I am filtering and using only those records where v_time is between 0 and 23 (leaving out 24). I am selecting v_time and count of violation_code and group by on v_time and order by v_time so get to see ordered output.

As per assignment, I have divided a day into 24 hours where 0 represents data between 00:00 to 00:59, 1 represents data between 01:00 – 01:59 and so on. Output contains hourly time in numbered format from 0-23 and count of violation_code in every hour.

--- Count of violation code in each hour of the day

```
SELECT cnt.v_time,
     count(cnt.violation_code)
FROM
  (SELECT CASE
          WHEN violation_time LIKE '%P'
             AND cast(substr(violation_time, 1, 2) AS int) == 12 THEN cast(substr(violation_time, 1, 2) AS int)
          WHEN violation_time LIKE '%P'
             AND cast(substr(violation_time, 1, 2) AS int)+12 BETWEEN 13 AND 23 THEN cast(substr(violation_time, 1, 2)
AS int)+12
          WHEN violation_time LIKE '%A'
             AND cast(substr(violation_time, 1, 2) AS int) BETWEEN 0 AND 11 THEN cast(substr(violation_time, 1, 2) AS int)
          ELSE 24
        END AS v_time,
        violation_code
   FROM nyc_data_partitioned_bucketed_orc
   WHERE violation_time != ''
    AND violation_time rlike '^[0-9]{4}[A|P]$'
    AND violation_code !=0) AS cnt
WHERE cnt.v_time BETWEEN 0 AND 23
GROUP BY cnt.v_time
ORDER BY cnt.v_time;
```

```
hive>
    > SELECT cnt.v_time,
    >       count(cnt.violation_code)
    > FROM
    >   (SELECT CASE
    >           WHEN violation_time LIKE '%P'
    >              AND cast(substr(violation_time, 1, 2) AS int) == 12 THEN cast(substr(violation_time, 1, 2) AS int)
    >           WHEN violation_time LIKE '%P'
    >              AND cast(substr(violation_time, 1, 2) AS int)+12 BETWEEN 13 AND 23 THEN cast(substr(violation_time, 1, 2)
AS int)+12
    >           WHEN violation_time LIKE '%A'
    >              AND cast(substr(violation_time, 1, 2) AS int) BETWEEN 0 AND 11 THEN cast(substr(violation_time, 1, 2) AS
int)
    >           ELSE 24
    >         END AS v_time,
    >         violation_code
    >    FROM nyc_data_partitioned_bucketed_orc
    >    WHERE violation_time != ''
    >     AND violation_time rlike '^[0-9]{4}[A|P]$'
    >     AND violation_code !=0) AS cnt
    > WHERE cnt.v_time BETWEEN 0 AND 23
    > GROUP BY cnt.v_time
    > ORDER BY cnt.v_time;
Query ID = cloudera_20181118061919_d40a2a1d-b2eb-4dc7-b6ff-b7b97dac202b
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_1542549294357_0017, Tracking URL =
http://quickstart.cloudera:8088/proxy/application_1542549294357_0017/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1542549294357_0017
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-11-18 06:19:36,009 Stage-1 map = 0%,  reduce = 0%
2018-11-18 06:19:53,934 Stage-1 map = 15%,  reduce = 0%, Cumulative CPU 14.64 sec
2018-11-18 06:20:00,263 Stage-1 map = 28%,  reduce = 0%, Cumulative CPU 21.14 sec
2018-11-18 06:20:06,573 Stage-1 map = 47%,  reduce = 0%, Cumulative CPU 27.15 sec
2018-11-18 06:20:11,886 Stage-1 map = 60%,  reduce = 0%, Cumulative CPU 33.45 sec
2018-11-18 06:20:16,123 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 38.02 sec
2018-11-18 06:20:24,555 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 40.55 sec
MapReduce Total cumulative CPU time: 40 seconds 550 msec
Ended Job = job_1542549294357_0017
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_1542549294357_0018, Tracking URL =
http://quickstart.cloudera:8088/proxy/application_1542549294357_0018/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1542549294357_0018
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2018-11-18 06:20:36,154 Stage-2 map = 0%,  reduce = 0%
2018-11-18 06:20:43,529 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 2.64 sec
2018-11-18 06:20:51,976 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 2.64 sec
MapReduce Total cumulative CPU time: 4 seconds 630 msec
Ended Job = job_1542549294357_0018
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 40.55 sec   HDFS Read: 21334578 HDFS Write: 613 SUCCESS
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 4.63 sec   HDFS Read: 5577 HDFS Write: 231 SUCCESS
Total MapReduce CPU Time Spent: 45 seconds 180 msec
OK
0       28463
1       46068
2       40312
3       32453
4       14545
5       43151
6       121545
7       270618
8       503817
9       595606
10      489423
```

```
11	574594
12	509983
13	549260
14	466046
15	314455
16	295974
17	211161
18	104279
19	26099
20	49220
21	55320
22	42536
23	29277
Time taken: 88.202 seconds, Fetched: 24 row(s)
hive>
```

5. **(b)** Based on study, considering that violation_time should not be empty and should contain 4 numbers and should end with either 'A' or 'P', considering violation_time format as HHMM (A/P) format. In inner query, I am selecting violation_time and converting violation_time using a CASE construct (when violation_time ends with 'P', casting first 2 characters as int and checking if equal to 12, if so then considering it as casting first 2 characters as int and next, when violation_time ends with 'P', casting first 2 characters as int and adding 12 in it and checking if between 13 and 23, if so then considering it as casting first 2 characters as int and adding 12 in it and next, when violation_time ends with 'A', casting first 2 characters as int and checking if between 0 and 11, if so then considering it as casting first 2 characters as int then in else part, considering everything else as 24) and naming this field as v_time and selecting summons_number where length of summons_number is 10 and violation_time != '' and violation_time ends with 'A' or 'P' and violation_time should not have any character other than numbers or 'A' or 'P'. In outer query, I am filtering and using only those records where v_time is between 0 and 23 (leaving out 24). I am selecting v_time and count of summons_number and group by on v_time and order by v_time so get to see ordered output.

   As per assignment, I have divided a day into 24 hours where 0 represents data between 00:00 to 00:59, 1 represents data between 01:00 – 01:59 and so on. Output contains hourly time in numbered format from 0-23 and count of tickets in every hour.

--- Count of tickets in each hour of the day

```
SELECT cnt.v_time,
    count(cnt.summons_number)
FROM
 (SELECT CASE
         WHEN violation_time LIKE '%P'
            AND cast(substr(violation_time, 1, 2) AS int) == 12 THEN cast(substr(violation_time, 1, 2) AS int)
         WHEN violation_time LIKE '%P'
            AND cast(substr(violation_time, 1, 2) AS int)+12 BETWEEN 13 AND 23 THEN cast(substr(violation_time, 1, 2)
AS int)+12
         WHEN violation_time LIKE '%A'
            AND cast(substr(violation_time, 1, 2) AS int) BETWEEN 0 AND 11 THEN cast(substr(violation_time, 1, 2) AS int)
         ELSE 24
       END AS v_time,
       summons_number
  FROM nyc_data_partitioned_bucketed_orc
```

```
  WHERE violation_time != ''
    AND violation_time rlike '^[0-9]{4}[A|P]$'
    AND length(summons_number) = 10) AS cnt
WHERE cnt.v_time BETWEEN 0 AND 23
GROUP BY cnt.v_time
ORDER BY cnt.v_time;

hive> SELECT cnt.v_time,
    >       count(cnt.summons_number)
    > FROM
    >   (SELECT CASE
    >             WHEN violation_time LIKE '%P'
    >                 AND cast(substr(violation_time, 1, 2) AS int) == 12 THEN cast(substr(violation_time, 1, 2) AS int)
    >             WHEN violation_time LIKE '%P'
    >                 AND cast(substr(violation_time, 1, 2) AS int)+12 BETWEEN 13 AND 23 THEN cast(substr(violation_time, 1, 2)
AS int)+12
    >             WHEN violation_time LIKE '%A'
    >                 AND cast(substr(violation_time, 1, 2) AS int) BETWEEN 0 AND 11 THEN cast(substr(violation_time, 1, 2) AS
int)
    >             ELSE 24
    >          END AS v_time,
    >          summons_number
    >    FROM nyc_data_partitioned_bucketed_orc
    >    WHERE violation_time != ''
    >      AND violation_time rlike '^[0-9]{4}[A|P]$'
    >      AND length(summons_number) = 10) AS cnt
    > WHERE cnt.v_time BETWEEN 0 AND 23
    > GROUP BY cnt.v_time
    > ORDER BY cnt.v_time;
Query ID = cloudera_20181118062121_bbd2686c-8374-46d4-9e7c-fc3b0c407c92
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_1542549294357_0019, Tracking URL =
http://quickstart.cloudera:8088/proxy/application_1542549294357_0019/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1542549294357_0019
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-11-18 06:22:04,852 Stage-1 map = 0%,  reduce = 0%
2018-11-18 06:22:25,479 Stage-1 map = 28%,  reduce = 0%, Cumulative CPU 16.03 sec
2018-11-18 06:22:30,983 Stage-1 map = 52%,  reduce = 0%, Cumulative CPU 22.22 sec
2018-11-18 06:22:37,497 Stage-1 map = 64%,  reduce = 0%, Cumulative CPU 28.91 sec
2018-11-18 06:22:40,786 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 32.7 sec
2018-11-18 06:22:49,240 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 36.36 sec
MapReduce Total cumulative CPU time: 36 seconds 360 msec
Ended Job = job_1542549294357_0019
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_1542549294357_0020, Tracking URL =
http://quickstart.cloudera:8088/proxy/application_1542549294357_0020/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1542549294357_0020
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2018-11-18 06:22:59,168 Stage-2 map = 0%,  reduce = 0%
2018-11-18 06:23:07,728 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 1.32 sec
2018-11-18 06:23:16,158 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 4.3 sec
MapReduce Total cumulative CPU time: 4 seconds 300 msec
Ended Job = job_1542549294357_0020
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 36.36 sec   HDFS Read: 23469996 HDFS Write: 600 SUCCESS
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 4.3 sec   HDFS Read: 5564 HDFS Write: 210 SUCCESS
Total MapReduce CPU Time Spent: 40 seconds 660 msec
OK
0       2729
1       22253
2       17447
3       13800
4       8009
5       6513
6       7623
7       31001
8       55528
9       56780
10      42704
11      54140
12      31883
13      15506
14      14660
15      12530
16      19170
17      17792
18      12843
19      9733
20      8751
21      9916
22      11022
23      6080
Time taken: 83.855 seconds, Fetched: 24 row(s)
hive>
```

6. In innermost subquery, I am selecting violation_time and converting violation_time using a CASE construct (when violation_time ends with 'P', casting first 2 characters as int and checking if equal to 12, if so then considering it as casting first 2 characters as int and next, when violation_time ends with 'P', casting first 2 characters as int and adding 12 in it and checking if between 13 and 23, if so then considering it as casting first 2 characters as int and adding 12 in it and next, when violation_time ends with 'A', casting first 2 characters as int and checking if between 0 and 11, if so then considering it as casting first 2 characters as int then in else part, considering everything else as 24) and naming this field

as v_time and selecting violation_code where violation_code != 0 and violation_time != '' and violation_time ends with 'A' or 'P' and violation_time should not have any character other than numbers or 'A' or 'P'. In immediate outer query, I am filtering and using only those records where v_time is between 0 and 23 (leaving out 24). I am selecting v_time and creating 6 bins/buckets using CASE construct where v_time with value 0,1,2,3 are numbered as 1, with value 4,5,6,7 are numbered as 2, with value 8,9,10,11 are numbered as 3, with value 12,13,14,15 are numbered as 4, with value 16,17,18,19 are numbered as 5, with value 20,21,22,23 are numbered as 6, naming this field as time_bin and selecting violation_code. In immediate outer query, I am selecting time_bin, violation_code and count of violation_code and group by on time_bin and violation_code. In immediate outer query, I am selecting time_bin, violation_code, count of violation_code and creating a rank of records by using partition by clause on time_bin and ordering on count of violation_code in descending order and naming this field as rk. In final outer query, I am selecting time_bin, violation_code where rk is between 1 and 3.

Output contains 6 bins/buckets of time and 3 most commonly occurring violation_code in each of these.

--- Top 3 most occurring violation codes in 6 bins/buckets created out of 24 hours of a day

```
SELECT ranked_bin.time_bin,
    ranked_bin.violation_code
FROM
 (SELECT grouped_bin.time_bin,
      grouped_bin.violation_code,
      grouped_bin.cnt_violation_code,
      rank() OVER (PARTITION BY grouped_bin.time_bin
            ORDER BY cnt_violation_code DESC) AS rk
  FROM
   (SELECT cnt_bin.time_bin,
       cnt_bin.violation_code,
       count(cnt_bin.violation_code) AS cnt_violation_code
    FROM
     (SELECT CASE
           WHEN cnt.v_time IN (0,
                     1,
                     2,
                     3) THEN 1
           WHEN cnt.v_time IN (4,
                     5,
                     6,
                     7) THEN 2
           WHEN cnt.v_time IN (8,
                     9,
                     10,
                     11) THEN 3
           WHEN cnt.v_time IN (12,
                     13,
                     14,
                     15) THEN 4
           WHEN cnt.v_time IN (16,
                     17,
                     18,
                     19) THEN 5
           WHEN cnt.v_time IN (20,
```

```
                        21,
                        22,
                        23) THEN 6
          END AS time_bin,
          cnt.violation_code
      FROM
       (SELECT CASE
                WHEN violation_time LIKE '%P'
                   AND cast(substr(violation_time, 1, 2) AS int) == 12 THEN cast(substr(violation_time, 1, 2) AS int)
                WHEN violation_time LIKE '%P'
                   AND cast(substr(violation_time, 1, 2) AS int)+12 BETWEEN 13 AND 23 THEN cast(substr(violation_time, 1,
2) AS int)+12
                WHEN violation_time LIKE '%A'
                   AND cast(substr(violation_time, 1, 2) AS int) BETWEEN 0 AND 11 THEN cast(substr(violation_time, 1, 2)
AS int)
                ELSE 24
             END AS v_time,
             violation_code
        FROM nyc_data_partitioned_bucketed_orc
        WHERE violation_time != ''
         AND violation_time rlike '^[0-9]{4}[A|P]$'
         AND violation_code !=0) AS cnt
      WHERE cnt.v_time BETWEEN 0 AND 23 ) cnt_bin
    GROUP BY cnt_bin.time_bin,
          cnt_bin.violation_code) grouped_bin) ranked_bin
WHERE ranked_bin.rk BETWEEN 1 AND 3;

hive> SELECT ranked_bin.time_bin,
   >       ranked_bin.violation_code
   > FROM
   >   (SELECT grouped_bin.time_bin,
   >         grouped_bin.violation_code,
   >         grouped_bin.cnt_violation_code,
   >         rank() OVER (PARTITION BY grouped_bin.time_bin
   >                   ORDER BY cnt_violation_code DESC) AS rk
   >    FROM
   >     (SELECT cnt_bin.time_bin,
   >           cnt_bin.violation_code,
   >           count(cnt_bin.violation_code) AS cnt_violation_code
   >      FROM
   >       (SELECT CASE
   >                 WHEN cnt.v_time IN (0,
   >                                     1,
   >                                     2,
   >                                     3) THEN 1
   >                 WHEN cnt.v_time IN (4,
   >                                     5,
   >                                     6,
   >                                     7) THEN 2
   >                 WHEN cnt.v_time IN (8,
   >                                     9,
   >                                     10,
   >                                     11) THEN 3
   >                 WHEN cnt.v_time IN (12,
```

```
   >                         13,
   >                         14,
   >                         15) THEN 4
   >                WHEN cnt.v_time IN (16,
   >                         17,
   >                         18,
   >                         19) THEN 5
   >                WHEN cnt.v_time IN (20,
   >                         21,
   >                         22,
   >                         23) THEN 6
   >              END AS time_bin,
   >              cnt.violation_code
   >         FROM
   >           (SELECT CASE
   >                     WHEN violation_time LIKE '%P'
   >                       AND cast(substr(violation_time, 1, 2) AS int) == 12 THEN cast(substr(violation_time, 1, 2) AS int)
   >                     WHEN violation_time LIKE '%P'
   >                       AND cast(substr(violation_time, 1, 2) AS int)+12 BETWEEN 13 AND 23 THEN cast(substr(violation_time,
1, 2) AS int)+12
   >                     WHEN violation_time LIKE '%A'
   >                       AND cast(substr(violation_time, 1, 2) AS int) BETWEEN 0 AND 11 THEN cast(substr(violation_time, 1, 2)
AS int)
   >                     ELSE 24
   >                   END AS v_time,
   >                   violation_code
   >            FROM nyc_data_partitioned_bucketed_orc
   >            WHERE violation_time != ''
   >              AND violation_time rlike '^[0-9]{4}[A|P]$'
   >              AND violation_code !=0) AS cnt
   >         WHERE cnt.v_time BETWEEN 0 AND 23 ) cnt_bin
   >       GROUP BY cnt_bin.time_bin,
   >                cnt_bin.violation_code) grouped_bin) ranked_bin
   > WHERE ranked_bin.rk BETWEEN 1 AND 3;
Query ID = cloudera_20181118060404_a3e44a2a-0afe-4fa7-83cf-2197e848db89
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_1542549294357_0005, Tracking URL =
http://quickstart.cloudera:8088/proxy/application_1542549294357_0005/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1542549294357_0005
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-11-18 06:04:47,336 Stage-1 map = 0%,  reduce = 0%
2018-11-18 06:05:05,661 Stage-1 map = 9%,  reduce = 0%, Cumulative CPU 16.86 sec
2018-11-18 06:05:12,012 Stage-1 map = 25%,  reduce = 0%, Cumulative CPU 23.11 sec
2018-11-18 06:05:18,342 Stage-1 map = 28%,  reduce = 0%, Cumulative CPU 29.37 sec
2018-11-18 06:05:23,675 Stage-1 map = 43%,  reduce = 0%, Cumulative CPU 35.33 sec
2018-11-18 06:05:30,024 Stage-1 map = 55%,  reduce = 0%, Cumulative CPU 41.19 sec
```

2018-11-18 06:05:36,499 Stage-1 map = 65%, reduce = 0%, Cumulative CPU 47.42 sec
2018-11-18 06:05:39,668 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 50.8 sec
2018-11-18 06:05:48,192 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 53.07 sec
MapReduce Total cumulative CPU time: 53 seconds 70 msec
Ended Job = job_1542549294357_0005
Launching Job 2 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_1542549294357_0006, Tracking URL =
http://quickstart.cloudera:8088/proxy/application_1542549294357_0006/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1542549294357_0006
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2018-11-18 06:05:59,605 Stage-2 map = 0%, reduce = 0%
2018-11-18 06:06:07,011 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 1.46 sec
2018-11-18 06:06:15,454 Stage-2 map = 100%, reduce = 100%, Cumulative CPU 5.86 sec
MapReduce Total cumulative CPU time: 5 seconds 860 msec
Ended Job = job_1542549294357_0006
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1   Cumulative CPU: 53.07 sec   HDFS Read: 21337089 HDFS Write: 11668 SUCCESS
Stage-Stage-2: Map: 1 Reduce: 1   Cumulative CPU: 5.86 sec   HDFS Read: 19823 HDFS Write: 85 SUCCESS
Total MapReduce CPU Time Spent: 58 seconds 930 msec
OK
1 21
1 40
1 14
2 14
2 40
2 21
3 21
3 36
3 38
4 36
4 38
4 37
5 38
5 14
5 37
6 7
6 40
6 14
Time taken: 100.139 seconds, Fetched: 18 row(s)
hive>

7. Using With clause, I am first preparing an intermediate result set which contains top 3 violation_code in terms of highest number of occurrences. In inner query of With clause, I am selecting violation_code and count of same where violation_code != 0 and group by violation_code. This intermediate result set is used in outer query of With clause to order by on count of violation_code in descending order and limit

output to top 3. This result set contains top 3 violation_code along with count of occurrences. This subquery output is then joined with main query.

Using With clause, I am preparing another intermediate result set which contains 6-time bins, violation_code and count of violation_code. In innermost subquery, I am selecting violation_time and converting violation_time using a CASE construct (when violation_time ends with 'P', casting first 2 characters as int and checking if equal to 12, if so then considering it as casting first 2 characters as int and next, when violation_time ends with 'P', casting first 2 characters as int and adding 12 in it and checking if between 13 and 23, if so then considering it as casting first 2 characters as int and adding 12 in it and next, when violation_time ends with 'A', casting first 2 characters as int and checking if between 0 and 11, if so then considering it as casting first 2 characters as int then in else part, considering everything else as 24) and naming this field as v_time and selecting violation_code where violation_code != 0 and violation_time != '' and violation_time ends with 'A' or 'P' and violation_time should not have any character other than numbers or 'A' or 'P'. In immediate outer query, I am filtering and using only those records where v_time is between 0 and 23 (leaving out 24). I am selecting v_time and creating 6 bins/buckets using CASE construct where v_time with value 0,1,2,3 are numbered as 1, with value 4,5,6,7 are numbered as 2, with value 8,9,10,11 are numbered as 3, with value 12,13,14,15 are numbered as 4, with value 16,17,18,19 are numbered as 5, with value 20,21,22,23 are numbered as 6, naming this field as time_bin and selecting violation_code. In immediate outer query, I am selecting time_bin, violation_code and count of violation_code and group by on time_bin and violation_code.

In main query, in inner query, I am joining above 2 outputs of subqueries, selecting time_bin, violation_code, count of violation_code and creating a new field by using partition by clause on violation_code and using max function on count of violation_code to get the highest count of violation_code where violation_code matches in both result set. And then in outer query, I am filtering and selecting time_bin and violation_code for those records where count of violation_code matches with maximum count of violation_code.

Output contains 3 most commonly occurring violation_code and bin/bucket in which they have occurred most of the time.

--- Most common bin/bucket for top 3 most occurring violation code

```
WITH c_violation_code AS
 (SELECT cnt.violation_code,
       cnt.cnt_violationcode
  FROM
   (SELECT violation_code,
         count(violation_code) AS cnt_violationcode
    FROM nyc_data_partitioned_bucketed_orc
    WHERE violation_code != 0
    GROUP BY violation_code) cnt
  ORDER BY cnt.cnt_violationcode DESC
  LIMIT 3),
   grouped_bin AS
 (SELECT cnt_bin.time_bin,
       cnt_bin.violation_code,
       count(cnt_bin.violation_code) AS cnt_violation_code
  FROM
   (SELECT CASE
           WHEN cnt.v_time IN (0,
```

```
                    1,
                    2,
                    3) THEN 1
            WHEN cnt.v_time IN (4,
                    5,
                    6,
                    7) THEN 2
            WHEN cnt.v_time IN (8,
                    9,
                    10,
                    11) THEN 3
            WHEN cnt.v_time IN (12,
                    13,
                    14,
                    15) THEN 4
            WHEN cnt.v_time IN (16,
                    17,
                    18,
                    19) THEN 5
            WHEN cnt.v_time IN (20,
                    21,
                    22,
                    23) THEN 6
          END AS time_bin,
          cnt.violation_code
     FROM
       (SELECT CASE
              WHEN violation_time LIKE '%P'
                 AND cast(substr(violation_time, 1, 2) AS int) == 12 THEN cast(substr(violation_time, 1, 2) AS int)
              WHEN violation_time LIKE '%P'
                 AND cast(substr(violation_time, 1, 2) AS int)+12 BETWEEN 13 AND 23 THEN cast(substr(violation_time, 1,
2) AS int)+12
              WHEN violation_time LIKE '%A'
                 AND cast(substr(violation_time, 1, 2) AS int) BETWEEN 0 AND 11 THEN cast(substr(violation_time, 1, 2) AS
int)
              ELSE 24
            END AS v_time,
            violation_code
        FROM nyc_data_partitioned_bucketed_orc
        WHERE violation_time != ''
         AND violation_time rlike '^[0-9]{4}[A|P]$'
         AND violation_code != 0) AS cnt
     WHERE cnt.v_time BETWEEN 0 AND 23 ) cnt_bin
   GROUP BY cnt_bin.time_bin,
          cnt_bin.violation_code)
SELECT final.violation_code,
      final.time_bin
FROM
  (SELECT gb.time_bin,
         gb.violation_code,
         gb.cnt_violation_code,
         max(gb.cnt_violation_code) OVER (PARTITION BY gb.violation_code) AS max_violation_code
   FROM grouped_bin gb
   JOIN c_violation_code cvc ON cvc.violation_code = gb.violation_code) FINAL
```

WHERE final.cnt_violation_code = final.max_violation_code;

```
hive> WITH c_violation_code AS
   >  (SELECT cnt.violation_code,
   >          cnt.cnt_violationcode
   >   FROM
   >     (SELECT violation_code,
   >             count(violation_code) AS cnt_violationcode
   >      FROM nyc_data_partitioned_bucketed_orc
   >      WHERE violation_code != 0
   >      GROUP BY violation_code) cnt
   >   ORDER BY cnt.cnt_violationcode DESC
   >   LIMIT 3),
   >     grouped_bin AS
   >  (SELECT cnt_bin.time_bin,
   >          cnt_bin.violation_code,
   >          count(cnt_bin.violation_code) AS cnt_violation_code
   >   FROM
   >     (SELECT CASE
   >               WHEN cnt.v_time IN (0,
   >                                   1,
   >                                   2,
   >                                   3) THEN 1
   >               WHEN cnt.v_time IN (4,
   >                                   5,
   >                                   6,
   >                                   7) THEN 2
   >               WHEN cnt.v_time IN (8,
   >                                   9,
   >                                   10,
   >                                   11) THEN 3
   >               WHEN cnt.v_time IN (12,
   >                                   13,
   >                                   14,
   >                                   15) THEN 4
   >               WHEN cnt.v_time IN (16,
   >                                   17,
   >                                   18,
   >                                   19) THEN 5
   >               WHEN cnt.v_time IN (20,
   >                                   21,
   >                                   22,
   >                                   23) THEN 6
   >             END AS time_bin,
   >             cnt.violation_code
   >      FROM
   >        (SELECT CASE
   >                  WHEN violation_time LIKE '%P'
   >                     AND cast(substr(violation_time, 1, 2) AS int) == 12 THEN cast(substr(violation_time, 1, 2) AS int)
   >                  WHEN violation_time LIKE '%P'
   >                     AND cast(substr(violation_time, 1, 2) AS int)+12 BETWEEN 13 AND 23 THEN cast(substr(violation_time, 1,
2) AS int)+12
   >                  WHEN violation_time LIKE '%A'
```

```
   >                     AND cast(substr(violation_time, 1, 2) AS int) BETWEEN 0 AND 11 THEN cast(substr(violation_time, 1, 2)
AS int)
   >                 ELSE 24
   >               END AS v_time,
   >               violation_code
   >         FROM nyc_data_partitioned_bucketed_orc
   >         WHERE violation_time != ''
   >           AND violation_time rlike '^[0-9]{4}[A|P]$'
   >           AND violation_code != 0) AS cnt
   >       WHERE cnt.v_time BETWEEN 0 AND 23 ) cnt_bin
   >    GROUP BY cnt_bin.time_bin,
   >             cnt_bin.violation_code)
   > SELECT final.violation_code,
   >        final.time_bin
   > FROM
   >    (SELECT gb.time_bin,
   >            gb.violation_code,
   >            gb.cnt_violation_code,
   >            max(gb.cnt_violation_code) OVER (PARTITION BY gb.violation_code) AS max_violation_code
   >     FROM grouped_bin gb
   >     JOIN c_violation_code cvc ON cvc.violation_code = gb.violation_code) FINAL
   > WHERE final.cnt_violation_code = final.max_violation_code;
Query ID = cloudera_20181118060707_2cd712fe-d39b-45e4-a5ef-81471e632cdb
Total jobs = 7
Launching Job 1 out of 7
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_1542549294357_0007, Tracking URL =
http://quickstart.cloudera:8088/proxy/application_1542549294357_0007/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1542549294357_0007
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-11-18 06:07:22,763 Stage-1 map = 0%,  reduce = 0%
2018-11-18 06:07:42,164 Stage-1 map = 11%,  reduce = 0%, Cumulative CPU 14.98 sec
2018-11-18 06:07:48,582 Stage-1 map = 26%,  reduce = 0%, Cumulative CPU 20.97 sec
2018-11-18 06:07:53,897 Stage-1 map = 32%,  reduce = 0%, Cumulative CPU 27.39 sec
2018-11-18 06:08:00,251 Stage-1 map = 47%,  reduce = 0%, Cumulative CPU 33.33 sec
2018-11-18 06:08:06,679 Stage-1 map = 55%,  reduce = 0%, Cumulative CPU 39.38 sec
2018-11-18 06:08:12,120 Stage-1 map = 60%,  reduce = 0%, Cumulative CPU 45.36 sec
2018-11-18 06:08:16,386 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 50.42 sec
2018-11-18 06:08:27,305 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 53.38 sec
MapReduce Total cumulative CPU time: 53 seconds 380 msec
Ended Job = job_1542549294357_0007
Launching Job 2 out of 7
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_1542549294357_0008, Tracking URL =
http://quickstart.cloudera:8088/proxy/application_1542549294357_0008/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1542549294357_0008
Hadoop job information for Stage-4: number of mappers: 1; number of reducers: 1
2018-11-18 06:08:40,069 Stage-4 map = 0%,  reduce = 0%
2018-11-18 06:08:59,562 Stage-4 map = 54%,  reduce = 0%, Cumulative CPU 16.05 sec
2018-11-18 06:09:05,042 Stage-4 map = 100%,  reduce = 0%, Cumulative CPU 22.72 sec
2018-11-18 06:09:14,730 Stage-4 map = 100%,  reduce = 100%, Cumulative CPU 25.17 sec
MapReduce Total cumulative CPU time: 25 seconds 170 msec
Ended Job = job_1542549294357_0008
Launching Job 3 out of 7
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_1542549294357_0009, Tracking URL =
http://quickstart.cloudera:8088/proxy/application_1542549294357_0009/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1542549294357_0009
Hadoop job information for Stage-5: number of mappers: 1; number of reducers: 1
2018-11-18 06:09:27,018 Stage-5 map = 0%,  reduce = 0%
2018-11-18 06:09:35,479 Stage-5 map = 100%,  reduce = 0%, Cumulative CPU 2.37 sec
2018-11-18 06:09:44,888 Stage-5 map = 100%,  reduce = 100%, Cumulative CPU 5.82 sec
MapReduce Total cumulative CPU time: 5 seconds 820 msec
Ended Job = job_1542549294357_0009
Stage-9 is selected by condition resolver.
Stage-10 is filtered out by condition resolver.
Stage-2 is filtered out by condition resolver.
Execution log at: /tmp/cloudera/cloudera_20181118060707_2cd712fe-d39b-45e4-a5ef-81471e632cdb.log
2018-11-18 06:09:51     Starting to launch local task to process map join;          maximum memory = 932184064
2018-11-18 06:09:53     Dump the side-table for tag: 1 with group count: 3 into file: file:/tmp/cloudera/bf5de87a-ee56-
4970-9987-14413bf12131/hive_2018-11-18_06-07-10_851_6346573440742795070-1/-local-10007/HashTable-Stage-
6/MapJoin-mapfile01--.hashtable
2018-11-18 06:09:53     Uploaded 1 File to: file:/tmp/cloudera/bf5de87a-ee56-4970-9987-14413bf12131/hive_2018-11-
18_06-07-10_851_6346573440742795070-1/-local-10007/HashTable-Stage-6/MapJoin-mapfile01--.hashtable (314 bytes)
2018-11-18 06:09:53     End of local task; Time Taken: 1.721 sec.
Execution completed successfully
MapredLocal task succeeded
Launching Job 5 out of 7
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1542549294357_0010, Tracking URL =
http://quickstart.cloudera:8088/proxy/application_1542549294357_0010/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1542549294357_0010
Hadoop job information for Stage-6: number of mappers: 1; number of reducers: 0
2018-11-18 06:10:04,673 Stage-6 map = 0%,  reduce = 0%
2018-11-18 06:10:13,158 Stage-6 map = 100%,  reduce = 0%, Cumulative CPU 1.75 sec
MapReduce Total cumulative CPU time: 1 seconds 750 msec
Ended Job = job_1542549294357_0010
Launching Job 6 out of 7
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_1542549294357_0011, Tracking URL =
http://quickstart.cloudera:8088/proxy/application_1542549294357_0011/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1542549294357_0011
Hadoop job information for Stage-3: number of mappers: 1; number of reducers: 1
2018-11-18 06:10:23,742 Stage-3 map = 0%,  reduce = 0%
2018-11-18 06:10:31,155 Stage-3 map = 100%,  reduce = 0%, Cumulative CPU 2.19 sec
2018-11-18 06:10:40,680 Stage-3 map = 100%,  reduce = 100%, Cumulative CPU 7.1 sec
MapReduce Total cumulative CPU time: 7 seconds 100 msec
Ended Job = job_1542549294357_0011
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 53.38 sec   HDFS Read: 21337073 HDFS Write: 11668 SUCCESS
Stage-Stage-4: Map: 1  Reduce: 1   Cumulative CPU: 25.17 sec   HDFS Read: 9650765 HDFS Write: 2151 SUCCESS
Stage-Stage-5: Map: 1  Reduce: 1   Cumulative CPU: 5.82 sec   HDFS Read: 6656 HDFS Write: 150 SUCCESS
Stage-Stage-6: Map: 1   Cumulative CPU: 1.75 sec   HDFS Read: 15614 HDFS Write: 453 SUCCESS
Stage-Stage-3: Map: 1  Reduce: 1   Cumulative CPU: 7.1 sec   HDFS Read: 8220 HDFS Write: 25 SUCCESS
Total MapReduce CPU Time Spent: 1 minutes 33 seconds 220 msec
OK
21      3
36      3
38      4
Time taken: 210.904 seconds, Fetched: 3 row(s)
hive>
```

8. **(a)** In innermost query, I am selecting month from issue_date and summons_number where length of summons_number is 10. In immediate outer query, I am selecting month converting it into 4 seasons using CASE construct where month with value 3,4,5 are named as 'Spring', month with value 6,7,8 are named as 'Summer', month with value 9,10,11 are named as 'Fall' and month with value 12,1,2 are named as 'Winter' and selecting summons_number. In immediate outer query, I am selecting season and count of summons_number group by season. In final outer query, I am selecting season and count of summons_number order by season and count of summons_number so get to see orderd output.

   Output contains 4 seasons and number of tickets in the seasons.

```
SELECT grouped_season.season,
    grouped_season.cnt_tickets
FROM
 (SELECT seasoned.season,
      count(seasoned.summons_number) AS cnt_tickets
  FROM
   (SELECT CASE
          WHEN month_vc.month_issue_date IN (3,
                                             4,
                                             5) THEN 'Spring'
          WHEN month_vc.month_issue_date IN (6,
                                             7,
                                             8) THEN 'Summer'
          WHEN month_vc.month_issue_date IN (9,
                                             10,
```

```
                                11) THEN 'Fall'
           WHEN month_vc.month_issue_date IN (12,
                                1,
                                2) THEN 'Winter'
       END AS season,
       month_vc.summons_number
   FROM
     (SELECT month(issue_date) month_issue_date,
          summons_number
      FROM nyc_data_partitioned_bucketed_orc
      WHERE length(summons_number) = 10) AS month_vc) AS seasoned
  GROUP BY seasoned.season) AS grouped_season
ORDER BY grouped_season.season,
      grouped_season.cnt_tickets;

hive> SELECT grouped_season.season,
    >       grouped_season.cnt_tickets
    > FROM
    >   (SELECT seasoned.season,
    >         count(seasoned.summons_number) AS cnt_tickets
    >    FROM
    >     (SELECT CASE
    >               WHEN month_vc.month_issue_date IN (3,
    >                                   4,
    >                                   5) THEN 'Spring'
    >               WHEN month_vc.month_issue_date IN (6,
    >                                   7,
    >                                   8) THEN 'Summer'
    >               WHEN month_vc.month_issue_date IN (9,
    >                                   10,
    >                                   11) THEN 'Fall'
    >               WHEN month_vc.month_issue_date IN (12,
    >                                   1,
    >                                   2) THEN 'Winter'
    >           END AS season,
    >           month_vc.summons_number
    >      FROM
    >       (SELECT month(issue_date) month_issue_date,
    >             summons_number
    >        FROM nyc_data_partitioned_bucketed_orc
    >        WHERE length(summons_number) = 10) AS month_vc) AS seasoned
    >    GROUP BY seasoned.season) AS grouped_season
    > ORDER BY grouped_season.season,
    >         grouped_season.cnt_tickets;
Query ID = cloudera_20181118042222_6d8c2949-78b5-4b3e-a30e-40a173d06ce1
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_1542539368466_0016, Tracking URL =
http://quickstart.cloudera:8088/proxy/application_1542539368466_0016/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1542539368466_0016
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-11-18 04:22:29,056 Stage-1 map = 0%,  reduce = 0%
2018-11-18 04:22:45,676 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 14.36 sec
2018-11-18 04:22:54,025 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 16.15 sec
MapReduce Total cumulative CPU time: 16 seconds 150 msec
Ended Job = job_1542539368466_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_1542539368466_0017, Tracking URL =
http://quickstart.cloudera:8088/proxy/application_1542539368466_0017/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1542539368466_0017
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2018-11-18 04:23:03,268 Stage-2 map = 0%,  reduce = 0%
2018-11-18 04:23:09,610 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 2.44 sec
2018-11-18 04:23:18,041 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 5.28 sec
MapReduce Total cumulative CPU time: 5 seconds 280 msec
Ended Job = job_1542539368466_0017
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 16.15 sec   HDFS Read: 22080253 HDFS Write: 204 SUCCESS
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 5.28 sec   HDFS Read: 5364 HDFS Write: 60 SUCCESS
Total MapReduce CPU Time Spent: 21 seconds 430 msec
OK
```

Fall	**979**
Spring	**277294**
Summer	**41735**
Winter	**181908**

```
Time taken: 59.31 seconds, Fetched: 4 row(s)
hive>
```

8. **(b)** In innermost query, I am selecting month from issue_date and violation_code where violation_code != 0. In immediate outer query, I am selecting month converting it into 4 seasons using CASE construct where month with value 3,4,5 are named as 'Spring', month with value 6,7,8 are named as 'Summer', month with value 9,10,11 are named as 'Fall' and month with value 12,1,2 are named as 'Winter' and selecting violation_code. In immediate outer query, I am selecting season, violation_code and count of violation_code. In immediate outer query, I am selecting season, violation_code, count of violation_code and creating a new filed by using partition by clause on season order by count of violation_code in descending order and using rank function and naming this field as rk. In final outer query, I am selecting season and violation_code where rk is between 1 and 3.

   Output contains top 3 most occurring violation_code for each season.

--- 3 most occurring violation codes for 4 seasons of the year

```
SELECT ranked.season,
    ranked.violation_code
FROM
 (SELECT cnt.season,
       cnt.violation_code,
       cnt.cnt_violation_code,
       rank() OVER (PARTITION BY cnt.season
              ORDER BY cnt.cnt_violation_code DESC) AS rk
  FROM
   (SELECT seasoned.season,
         seasoned.violation_code,
         count(seasoned.violation_code) AS cnt_violation_code
    FROM
     (SELECT CASE
            WHEN month_vc.month_issue_date IN (3,
                                               4,
                                               5) THEN 'Spring'
            WHEN month_vc.month_issue_date IN (6,
                                               7,
                                               8) THEN 'Summer'
            WHEN month_vc.month_issue_date IN (9,
                                               10,
                                               11) THEN 'Fall'
            WHEN month_vc.month_issue_date IN (12,
                                               1,
                                               2) THEN 'Winter'
          END AS season,
          month_vc.violation_code
      FROM
       (SELECT month(issue_date) month_issue_date,
            violation_code
        FROM nyc_data_partitioned_bucketed_orc
        WHERE violation_code !=0) AS month_vc) AS seasoned
    GROUP BY seasoned.season,
          seasoned.violation_code) AS cnt) AS ranked
WHERE ranked.rk BETWEEN 1 AND 3;

hive> SELECT ranked.season,
    >       ranked.violation_code
    > FROM
    >   (SELECT cnt.season,
    >         cnt.violation_code,
    >         cnt.cnt_violation_code,
    >         rank() OVER (PARTITION BY cnt.season
    >                 ORDER BY cnt.cnt_violation_code DESC) AS rk
    >    FROM
    >     (SELECT seasoned.season,
    >           seasoned.violation_code,
    >           count(seasoned.violation_code) AS cnt_violation_code
    >      FROM
    >       (SELECT CASE
    >               WHEN month_vc.month_issue_date IN (3,
    >                                                  4,
    >                                                  5) THEN 'Spring'
```

```
    >                 WHEN month_vc.month_issue_date IN (6,
    >                                                    7,
    >                                                    8) THEN 'Summer'
    >                 WHEN month_vc.month_issue_date IN (9,
    >                                                    10,
    >                                                    11) THEN 'Fall'
    >                 WHEN month_vc.month_issue_date IN (12,
    >                                                    1,
    >                                                    2) THEN 'Winter'
    >               END AS season,
    >               month_vc.violation_code
    >          FROM
    >            (SELECT month(issue_date) month_issue_date,
    >                    violation_code
    >             FROM nyc_data_partitioned_bucketed_orc
    >             WHERE violation_code !=0) AS month_vc) AS seasoned
    >        GROUP BY seasoned.season,
    >                 seasoned.violation_code) AS cnt) AS ranked
    > WHERE ranked.rk BETWEEN 1 AND 3;
Query ID = cloudera_20181118043131_4d0476b1-579c-4da0-82b7-2a518d7b2819
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_1542539368466_0018, Tracking URL =
http://quickstart.cloudera:8088/proxy/application_1542539368466_0018/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1542539368466_0018
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-11-18 04:31:56,658 Stage-1 map = 0%,  reduce = 0%
2018-11-18 04:32:14,719 Stage-1 map = 43%,  reduce = 0%, Cumulative CPU 15.08 sec
2018-11-18 04:32:20,990 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 22.39 sec
2018-11-18 04:32:31,652 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 24.24 sec
MapReduce Total cumulative CPU time: 24 seconds 240 msec
Ended Job = job_1542539368466_0018
Launching Job 2 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_1542539368466_0019, Tracking URL =
http://quickstart.cloudera:8088/proxy/application_1542539368466_0019/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1542539368466_0019
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2018-11-18 04:32:43,413 Stage-2 map = 0%,  reduce = 0%
2018-11-18 04:32:49,748 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 2.47 sec
2018-11-18 04:32:59,254 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 6.94 sec
```

MapReduce Total cumulative CPU time: 6 seconds 940 msec
Ended Job = job_1542539368466_0019
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1   Cumulative CPU: 24.24 sec   HDFS Read: 19945342 HDFS Write: 9111 SUCCESS
Stage-Stage-2: Map: 1 Reduce: 1   Cumulative CPU: 6.94 sec   HDFS Read: 17450 HDFS Write: 86 SUCCESS
Total MapReduce CPU Time Spent: 31 seconds 180 msec
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: 72.025 seconds, Fetched: 12 row(s)
hive>

SECTION II

Analysis I

1.

```
SELECT count(summons_number)
FROM nyc_data_partitioned_bucketed_orc
WHERE length(summons_number) = 10;
```

501916

2.

```
SELECT count(DISTINCT registration_state)
FROM nyc_data_partitioned_bucketed_orc
WHERE length(summons_number) = 10
  AND registration_state rlike '[^0-9]'
  AND plate_id not rlike '[^a-zA-Z0-9]';
```

63

```
SELECT DISTINCT registration_state
FROM nyc_data_partitioned_bucketed_orc
WHERE length(summons_number) = 10
  AND registration_state rlike '[^0-9]'
  AND plate_id not rlike '[^a-zA-Z0-9]';
```

AB
AK
AL
AR
AZ
BC
CA
CO
CT
DC
DE
DP
FL
GA
GV
HI
IA
ID
IL
IN
KS
KY
LA
MA
MB
MD

ME
MI
MN
MO
MS
MT
NB
NC
ND
NE
NH
NJ
NM
NS
NV
NY
OH
OK
ON
OR
PA
PE
PR
QB
RI
SC
SD
SK
TN
TX
UT
VA
VT
WA
WI
WV
WY

3.

```
SELECT count(summons_number)
FROM nyc_data_partitioned_bucketed_orc
WHERE length(summons_number) = 10
  AND (street_code1 IS NULL
      OR street_code2 IS NULL
      OR street_code3 IS NULL
      OR street_code1 == 0
      OR street_code2 == 0
      OR street_code3 == 0);
```

77367

Analysis II

1.

```
SELECT cnt.violation_code,
     cnt.cnt_violation_code
FROM
 (SELECT violation_code,
       count(violation_code) AS cnt_violation_code
  FROM nyc_data_partitioned_bucketed_orc
  WHERE violation_code != 0
  GROUP BY violation_code) cnt
ORDER BY cnt.cnt_violation_code DESC
LIMIT 5;
```

21	**768082**
36	**662765**
38	**542079**
14	**476660**
20	**319646**

2(a).

```
SELECT cnt.vehicle_body_type,
     cnt.cnt_summons_number
FROM
 (SELECT vehicle_body_type,
       count(summons_number) AS cnt_summons_number
  FROM nyc_data_partitioned_bucketed_orc
  WHERE length(summons_number) = 10
    AND vehicle_body_type != '00'
    AND vehicle_body_type rlike '^[a-zA-Z0-9]'
    AND vehicle_body_type rlike '[a-zA-Z0-9]$'
  GROUP BY vehicle_body_type) cnt
ORDER BY cnt.cnt_summons_number DESC
LIMIT 5;
```

SDN	**182773**
SUBN	**148334**
VAN	**60943**
DELV	**47816**
P-U	**9072**

2(b).

```
SELECT cnt.vehicle_make,
     cnt.cnt_summons_number
FROM
 (SELECT vehicle_make,
       count(summons_number) AS cnt_summons_number
  FROM nyc_data_partitioned_bucketed_orc
  WHERE length(summons_number) = 10
    AND vehicle_make rlike '^[a-zA-Z0-9]'
    AND vehicle_make rlike '[a-zA-Z0-9]$'
  GROUP BY vehicle_make) cnt
ORDER BY cnt.cnt_summons_number DESC
```

```
LIMIT 5;
```

FORD	**53667**
TOYOT	**53549**
HONDA	**48689**
NISSA	**40888**
CHEVR	**27128**

3(a).

```
SELECT cnt.violation_precinct,
      cnt.cnt_summons_number
FROM
  (SELECT violation_precinct,
         count(summons_number) AS cnt_summons_number
   FROM nyc_data_partitioned_bucketed_orc
   WHERE length(summons_number) = 10
     AND violation_precinct != 0
   GROUP BY violation_precinct) cnt
ORDER BY cnt.cnt_summons_number DESC
LIMIT 5;
```

19	**15865**
18	**14720**
70	**14402**
72	**13880**
1	**13778**

3(b).

```
SELECT cnt.issuer_precinct,
      cnt.cnt_summons_number
FROM
  (SELECT issuer_precinct,
         count(summons_number) AS cnt_summons_number
   FROM nyc_data_partitioned_bucketed_orc
   WHERE length(summons_number) = 10
     AND issuer_precinct != 0
   GROUP BY issuer_precinct) cnt
ORDER BY cnt.cnt_summons_number DESC
LIMIT 5;
```

110	**12216**
109	**10268**
70	**10148**
401	**9907**
34	**9446**

4(a).

```
WITH precinct AS
  (SELECT cnt.issuer_precinct,
         cnt.cnt_summons_number
   FROM
     (SELECT issuer_precinct,
           count(summons_number) AS cnt_summons_number
      FROM nyc_data_partitioned_bucketed_orc
```

```
    WHERE length(summons_number) = 10
      AND issuer_precinct != 0
    GROUP BY issuer_precinct) cnt
  ORDER BY cnt.cnt_summons_number DESC
  LIMIT 3)
SELECT cnt_max_code.issuer_precinct,
     cnt_max_code.violation_code,
     cnt_max_code.max_cnt_violation_code
FROM
 (SELECT cnt_code.issuer_precinct,
       cnt_code.violation_code,
       cnt_code.cnt_violation_code,
       max(cnt_code.cnt_violation_code) OVER (PARTITION BY cnt_code.issuer_precinct) AS max_cnt_violation_code
  FROM
   (SELECT p.issuer_precinct,
         ndp.violation_code,
         count(ndp.violation_code) AS cnt_violation_code
    FROM nyc_data_partitioned_bucketed_orc ndp
    JOIN precinct p ON p.issuer_precinct = ndp.issuer_precinct
    WHERE ndp.violation_code != 0
    GROUP BY p.issuer_precinct,
          ndp.violation_code) cnt_code)cnt_max_code
WHERE cnt_max_code.max_cnt_violation_code = cnt_max_code.cnt_violation_code;
```

70	**21**	**21935**
109	**38**	**16425**
110	**21**	**13840**

4(b).

```
WITH precinct AS
 (SELECT cnt.issuer_precinct,
       cnt.cnt_summons_number
  FROM
   (SELECT issuer_precinct,
         count(summons_number) AS cnt_summons_number
    FROM nyc_data_partitioned_bucketed_orc
    WHERE length(summons_number) = 10
      AND issuer_precinct != 0
    GROUP BY issuer_precinct) cnt
  ORDER BY cnt.cnt_summons_number DESC
  LIMIT 3)
SELECT cnt_max_code.issuer_precinct,
     cnt_max_code.violation_code,
     cnt_max_code.cnt_violation_code
FROM
 (SELECT cnt_code.issuer_precinct,
       cnt_code.violation_code,
       cnt_code.cnt_violation_code,
       rank() OVER (PARTITION BY cnt_code.issuer_precinct
                    ORDER BY cnt_code.cnt_violation_code DESC) AS rank_cnt_violation_code
  FROM
   (SELECT p.issuer_precinct,
         ndp.violation_code,
         count(ndp.violation_code) AS cnt_violation_code
```

```
    FROM nyc_data_partitioned_bucketed_orc ndp
    JOIN precinct p ON p.issuer_precinct = ndp.issuer_precinct
    WHERE ndp.violation_code != 0
    GROUP BY p.issuer_precinct,
          ndp.violation_code) cnt_code)cnt_max_code
WHERE cnt_max_code.rank_cnt_violation_code BETWEEN 1 AND 5;
```

70	**21**	**21935**
70	**38**	**20133**
70	**20**	**7791**
70	**37**	**6547**
70	**71**	**6460**
109	**38**	**16425**
109	**21**	**14696**
109	**14**	**12939**
109	**20**	**11056**
109	**37**	**10051**
110	**21**	**13840**
110	**38**	**9718**
110	**20**	**7141**
110	**37**	**5936**
110	**40**	**5245**
110	**14**	**5245**

5(a).

```
SELECT cnt.v_time,
     count(cnt.violation_code)
FROM
  (SELECT CASE
          WHEN violation_time LIKE '%P'
             AND cast(substr(violation_time, 1, 2) AS int) == 12 THEN cast(substr(violation_time, 1, 2) AS int)
          WHEN violation_time LIKE '%P'
             AND cast(substr(violation_time, 1, 2) AS int)+12 BETWEEN 13 AND 23 THEN cast(substr(violation_time, 1, 2) AS
int)+12
          WHEN violation_time LIKE '%A'
             AND cast(substr(violation_time, 1, 2) AS int) BETWEEN 0 AND 11 THEN cast(substr(violation_time, 1, 2) AS int)
          ELSE 24
        END AS v_time,
        violation_code
   FROM nyc_data_partitioned_bucketed_orc
   WHERE violation_time != ''
    AND violation_time rlike '^[0-9]{4}[A|P]$'
    AND violation_code !=0) AS cnt
WHERE cnt.v_time BETWEEN 0 AND 23
GROUP BY cnt.v_time
ORDER BY cnt.v_time;
```

0	**28463**
1	**46068**
2	**40312**
3	**32453**
4	**14545**
5	**43151**
6	**121545**

7	270618
8	503817
9	595606
10	489423
11	574594
12	509983
13	549260
14	466046
15	314455
16	295974
17	211161
18	104279
19	26099
20	49220
21	55320
22	42536
23	29277

5(b).

```
SELECT cnt.v_time,
     count(cnt.summons_number)
FROM
 (SELECT CASE
          WHEN violation_time LIKE '%P'
             AND cast(substr(violation_time, 1, 2) AS int) == 12 THEN cast(substr(violation_time, 1, 2) AS int)
          WHEN violation_time LIKE '%P'
             AND cast(substr(violation_time, 1, 2) AS int)+12 BETWEEN 13 AND 23 THEN cast(substr(violation_time, 1, 2) AS
int)+12
          WHEN violation_time LIKE '%A'
             AND cast(substr(violation_time, 1, 2) AS int) BETWEEN 0 AND 11 THEN cast(substr(violation_time, 1, 2) AS int)
          ELSE 24
        END AS v_time,
        summons_number
  FROM nyc_data_partitioned_bucketed_orc
  WHERE violation_time != ''
    AND violation_time rlike '^[0-9]{4}[A|P]$'
    AND length(summons_number) = 10) AS cnt
WHERE cnt.v_time BETWEEN 0 AND 23
GROUP BY cnt.v_time
ORDER BY cnt.v_time;
```

0	2729
1	22253
2	17447
3	13800
4	8009
5	6513
6	7623
7	31001
8	55528
9	56780
10	42704
11	54140
12	31883

13	**15506**
14	**14660**
15	**12530**
16	**19170**
17	**17792**
18	**12843**
19	**9733**
20	**8751**
21	**9916**
22	**11022**
23	**6080**

6.

```
SELECT ranked_bin.time_bin,
     ranked_bin.violation_code
FROM
  (SELECT grouped_bin.time_bin,
        grouped_bin.violation_code,
        grouped_bin.cnt_violation_code,
        rank() OVER (PARTITION BY grouped_bin.time_bin
                ORDER BY cnt_violation_code DESC) AS rk
   FROM
    (SELECT cnt_bin.time_bin,
          cnt_bin.violation_code,
          count(cnt_bin.violation_code) AS cnt_violation_code
     FROM
      (SELECT CASE
              WHEN cnt.v_time IN (0,
                                  1,
                                  2,
                                  3) THEN 1
              WHEN cnt.v_time IN (4,
                                  5,
                                  6,
                                  7) THEN 2
              WHEN cnt.v_time IN (8,
                                  9,
                                  10,
                                  11) THEN 3
              WHEN cnt.v_time IN (12,
                                  13,
                                  14,
                                  15) THEN 4
              WHEN cnt.v_time IN (16,
                                  17,
                                  18,
                                  19) THEN 5
              WHEN cnt.v_time IN (20,
                                  21,
                                  22,
                                  23) THEN 6
            END AS time_bin,
            cnt.violation_code
       FROM
```

```
        (SELECT CASE
                    WHEN violation_time LIKE '%P'
                       AND cast(substr(violation_time, 1, 2) AS int) == 12 THEN cast(substr(violation_time, 1, 2) AS int)
                    WHEN violation_time LIKE '%P'
                       AND cast(substr(violation_time, 1, 2) AS int)+12 BETWEEN 13 AND 23 THEN cast(substr(violation_time, 1, 2)
AS int)+12
                    WHEN violation_time LIKE '%A'
                       AND cast(substr(violation_time, 1, 2) AS int) BETWEEN 0 AND 11 THEN cast(substr(violation_time, 1, 2) AS
int)
                    ELSE 24
                END AS v_time,
                violation_code
         FROM nyc_data_partitioned_bucketed_orc
         WHERE violation_time != ''
          AND violation_time rlike '^[0-9]{4}[A|P]$'
          AND violation_code !=0) AS cnt
       WHERE cnt.v_time BETWEEN 0 AND 23 ) cnt_bin
    GROUP BY cnt_bin.time_bin,
           cnt_bin.violation_code) grouped_bin) ranked_bin
WHERE ranked_bin.rk BETWEEN 1 AND 3;
```

1	**21**
1	**40**
1	**14**
2	**14**
2	**40**
2	**21**
3	**21**
3	**36**
3	**38**
4	**36**
4	**38**
4	**37**
5	**38**
5	**14**
5	**37**
6	**7**
6	**40**
6	**14**

7.

```
WITH c_violation_code AS
 (SELECT cnt.violation_code,
       cnt.cnt_violationcode
  FROM
   (SELECT violation_code,
          count(violation_code) AS cnt_violationcode
    FROM nyc_data_partitioned_bucketed_orc
    WHERE violation_code != 0
    GROUP BY violation_code) cnt
  ORDER BY cnt.cnt_violationcode DESC
  LIMIT 3),
   grouped_bin AS
 (SELECT cnt_bin.time_bin,
```

```
       cnt_bin.violation_code,
       count(cnt_bin.violation_code) AS cnt_violation_code
  FROM
    (SELECT CASE
              WHEN cnt.v_time IN (0,
                                  1,
                                  2,
                                  3) THEN 1
              WHEN cnt.v_time IN (4,
                                  5,
                                  6,
                                  7) THEN 2
              WHEN cnt.v_time IN (8,
                                  9,
                                  10,
                                  11) THEN 3
              WHEN cnt.v_time IN (12,
                                  13,
                                  14,
                                  15) THEN 4
              WHEN cnt.v_time IN (16,
                                  17,
                                  18,
                                  19) THEN 5
              WHEN cnt.v_time IN (20,
                                  21,
                                  22,
                                  23) THEN 6
            END AS time_bin,
            cnt.violation_code
     FROM
       (SELECT CASE
                 WHEN violation_time LIKE '%P'
                    AND cast(substr(violation_time, 1, 2) AS int) == 12 THEN cast(substr(violation_time, 1, 2) AS int)
                 WHEN violation_time LIKE '%P'
                    AND cast(substr(violation_time, 1, 2) AS int)+12 BETWEEN 13 AND 23 THEN cast(substr(violation_time, 1, 2)
AS int)+12
                 WHEN violation_time LIKE '%A'
                    AND cast(substr(violation_time, 1, 2) AS int) BETWEEN 0 AND 11 THEN cast(substr(violation_time, 1, 2) AS
int)
                 ELSE 24
               END AS v_time,
               violation_code
        FROM nyc_data_partitioned_bucketed_orc
        WHERE violation_time != ''
          AND violation_time rlike '^[0-9]{4}[A|P]$'
          AND violation_code != 0) AS cnt
     WHERE cnt.v_time BETWEEN 0 AND 23 ) cnt_bin
  GROUP BY cnt_bin.time_bin,
           cnt_bin.violation_code)
SELECT final.violation_code,
       final.time_bin
FROM
  (SELECT gb.time_bin,
```

```
        gb.violation_code,
        gb.cnt_violation_code,
        max(gb.cnt_violation_code) OVER (PARTITION BY gb.violation_code) AS max_violation_code
   FROM grouped_bin gb
   JOIN c_violation_code cvc ON cvc.violation_code = gb.violation_code) FINAL
WHERE final.cnt_violation_code = final.max_violation_code;
```

21	**3**
36	**3**
38	**4**

8(a).

```
SELECT grouped_season.season,
      grouped_season.cnt_tickets
FROM
  (SELECT seasoned.season,
        count(seasoned.summons_number) AS cnt_tickets
   FROM
     (SELECT CASE
             WHEN month_vc.month_issue_date IN (3,
                                                4,
                                                5) THEN 'Spring'
             WHEN month_vc.month_issue_date IN (6,
                                                7,
                                                8) THEN 'Summer'
             WHEN month_vc.month_issue_date IN (9,
                                                10,
                                                11) THEN 'Fall'
             WHEN month_vc.month_issue_date IN (12,
                                                1,
                                                2) THEN 'Winter'
          END AS season,
          month_vc.summons_number
      FROM
       (SELECT month(issue_date) month_issue_date,
             summons_number
        FROM nyc_data_partitioned_bucketed_orc
        WHERE length(summons_number) = 10) AS month_vc) AS seasoned
   GROUP BY seasoned.season) AS grouped_season
ORDER BY grouped_season.season,
        grouped_season.cnt_tickets;
```

Fall	**979**
Spring	**277294**
Summer	**41735**
Winter	**181908**

8(b).

```
SELECT ranked.season,
      ranked.violation_code
FROM
  (SELECT cnt.season,
        cnt.violation_code,
        cnt.cnt_violation_code,
```

```
       rank() OVER (PARTITION BY cnt.season
                    ORDER BY cnt.cnt_violation_code DESC) AS rk
  FROM
    (SELECT seasoned.season,
            seasoned.violation_code,
            count(seasoned.violation_code) AS cnt_violation_code
     FROM
       (SELECT CASE
                   WHEN month_vc.month_issue_date IN (3,
                                                      4,
                                                      5) THEN 'Spring'
                   WHEN month_vc.month_issue_date IN (6,
                                                      7,
                                                      8) THEN 'Summer'
                   WHEN month_vc.month_issue_date IN (9,
                                                      10,
                                                      11) THEN 'Fall'
                   WHEN month_vc.month_issue_date IN (12,
                                                      1,
                                                      2) THEN 'Winter'
               END AS season,
               month_vc.violation_code
        FROM
          (SELECT month(issue_date) month_issue_date,
                  violation_code
           FROM nyc_data_partitioned_bucketed_orc
           WHERE violation_code !=0) AS month_vc) AS seasoned
     GROUP BY seasoned.season,
              seasoned.violation_code) AS cnt) AS ranked
WHERE ranked.rk BETWEEN 1 AND 3;
```

Fall	**46**
Fall	**21**
Fall	**40**
Spring	**21**
Spring	**36**
Spring	**38**
Summer	**21**
Summer	**36**
Summer	**38**
Winter	**21**
Winter	**36**
Winter	**38**

SECTION III

Start hive CLI and run all these in the same order as shown below:

```
create database hive_assignment;
use hive_assignment;

set hive.exec.dynamic.partition= true ;
set hive.exec.dynamic.partition.mode=nonstrict;
set hive.exec.max.dynamic.partitions= 1000 ;
set hive.exec.max.dynamic.partitions.pernode= 1000 ;
set hive.enforce.bucketing= true ;
set hive.stats.autogather=true;
SET hive.optimize.sort.dynamic.partition=true;
SET orc.compress=SNAPPY;
SET hive.exec.compress.output=true;
SET mapred.output.compression.codec=org.apache.hadoop.io.compress.SnappyCodec;
SET mapred.output.compression.type=BLOCK;
set mapreduce.map.memory.mb=5120;
set mapreduce.reduce.memory.mb=5120;
set mapreduce.map.java.opts=-Xmx5G;
set mapreduce.reduce.java.opts=-Xmx5G;
SET mapred.child.java.opts=-Xmx5G -XX:+UseConcMarkSweepGC  -XX:-UseGCOverheadLimit;

CREATE EXTERNAL TABLE IF NOT EXISTS NYC_DATA_EXT(
`SUMMONS_NUMBER` INT,
`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_DATE` INT,
`VIOLATION_LOCATION` STRING,
`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` STRING,
`VEHICLE_YEAR` INT,
`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 ','
LOCATION '/user/cloudera/hiveassignment/'
TBLPROPERTIES("skip.header.line.count"="1");



CREATE EXTERNAL TABLE IF NOT EXISTS NYC_DATA_PARTITIONED_BUCKETED_ORC(
`SUMMONS_NUMBER` INT,
`PLATE_ID` STRING,
`REGISTRATION_STATE` STRING,
`PLATE_TYPE` STRING,
`ISSUE_DATE` DATE,
`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` STRING,
`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` STRING,
`VEHICLE_YEAR` INT,
`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_CODE INT)
CLUSTERED BY (SUMMONS_NUMBER) INTO 11 BUCKETS
STORED AS ORC
LOCATION '/user/cloudera/hiveassignment_orc/'
TBLPROPERTIES("orc.compress"="SNAPPY");


INSERT OVERWRITE TABLE NYC_DATA_PARTITIONED_BUCKETED_ORC PARTITION(VIOLATION_CODE)
SELECT SUMMONS_NUMBER,
    PLATE_ID,
    REGISTRATION_STATE,
    PLATE_TYPE,
    to_date(from_unixtime(unix_timestamp(issue_date, 'MM/dd/yyyy'), 'yyyy-MM-dd')),
    VEHICLE_BODY_TYPE,
    VEHICLE_MAKE,
    ISSUING_AGENCY,
    STREET_CODE1,
    STREET_CODE2,
    STREET_CODE3,
    VEHICLE_EXPIRATION_DATE,
    VIOLATION_LOCATION,
    VIOLATION_PRECINCT,
    ISSUER_PRECINCT,
    ISSUER_CODE,
    ISSUER_COMMAND,
    ISSUER_SQUAD,
    VIOLATION_TIME,
    TIME_FIRST_OBSERVED,
    VIOLATION_COUNTY,
    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_CODE
FROM NYC_DATA_EXT
WHERE year(to_date(from_unixtime(unix_timestamp(issue_date, 'MM/dd/yyyy'), 'yyyy-MM-dd')))='2017';
```

--- Analysis I

1.

```
SELECT count(summons_number)
FROM nyc_data_partitioned_bucketed_orc
WHERE length(summons_number) = 10;
```

2.

```
SELECT count(DISTINCT registration_state)
FROM nyc_data_partitioned_bucketed_orc
WHERE length(summons_number) = 10
  AND registration_state rlike '[^0-9]'
  AND plate_id not rlike '[^a-zA-Z0-9]';

SELECT DISTINCT registration_state
FROM nyc_data_partitioned_bucketed_orc
WHERE length(summons_number) = 10
  AND registration_state rlike '[^0-9]'
  AND plate_id not rlike '[^a-zA-Z0-9]';
```

3.

```
SELECT count(summons_number)
FROM nyc_data_partitioned_bucketed_orc
WHERE length(summons_number) = 10
  AND (street_code1 IS NULL
    OR street_code2 IS NULL
    OR street_code3 IS NULL
    OR street_code1 == 0
    OR street_code2 == 0
    OR street_code3 == 0);
```

--- Analysis II

1.

```
SELECT cnt.violation_code,
    cnt.cnt_violation_code
FROM
  (SELECT violation_code,
       count(violation_code) AS cnt_violation_code
   FROM nyc_data_partitioned_bucketed_orc
```

```
  WHERE violation_code != 0
  GROUP BY violation_code) cnt
ORDER BY cnt.cnt_violation_code DESC
LIMIT 5;
```

2(a).

```
SELECT cnt.vehicle_body_type,
      cnt.cnt_summons_number
FROM
 (SELECT vehicle_body_type,
         count(summons_number) AS cnt_summons_number
  FROM nyc_data_partitioned_bucketed_orc
  WHERE length(summons_number) = 10
    AND vehicle_body_type != '00'
    AND vehicle_body_type rlike '^[a-zA-Z0-9]'
    AND vehicle_body_type rlike '[a-zA-Z0-9]$'
  GROUP BY vehicle_body_type) cnt
ORDER BY cnt.cnt_summons_number DESC
LIMIT 5;
```

2(b).

```
SELECT cnt.vehicle_make,
      cnt.cnt_summons_number
FROM
 (SELECT vehicle_make,
         count(summons_number) AS cnt_summons_number
  FROM nyc_data_partitioned_bucketed_orc
  WHERE length(summons_number) = 10
    AND vehicle_make rlike '^[a-zA-Z0-9]'
    AND vehicle_make rlike '[a-zA-Z0-9]$'
  GROUP BY vehicle_make) cnt
ORDER BY cnt.cnt_summons_number DESC
LIMIT 5;
```

3(a).

```
SELECT cnt.violation_precinct,
      cnt.cnt_summons_number
FROM
 (SELECT violation_precinct,
         count(summons_number) AS cnt_summons_number
  FROM nyc_data_partitioned_bucketed_orc
  WHERE length(summons_number) = 10
    AND violation_precinct != 0
  GROUP BY violation_precinct) cnt
ORDER BY cnt.cnt_summons_number DESC
LIMIT 5;
```

3(b).

```
SELECT cnt.issuer_precinct,
      cnt.cnt_summons_number
FROM
 (SELECT issuer_precinct,
         count(summons_number) AS cnt_summons_number
  FROM nyc_data_partitioned_bucketed_orc
```

```
  WHERE length(summons_number) = 10
    AND issuer_precinct != 0
  GROUP BY issuer_precinct) cnt
ORDER BY cnt.cnt_summons_number DESC
LIMIT 5;
```

4(a).

```
WITH precinct AS
 (SELECT cnt.issuer_precinct,
        cnt.cnt_summons_number
  FROM
    (SELECT issuer_precinct,
          count(summons_number) AS cnt_summons_number
     FROM nyc_data_partitioned_bucketed_orc
     WHERE length(summons_number) = 10
       AND issuer_precinct != 0
     GROUP BY issuer_precinct) cnt
  ORDER BY cnt.cnt_summons_number DESC
  LIMIT 3)
SELECT cnt_max_code.issuer_precinct,
      cnt_max_code.violation_code,
      cnt_max_code.max_cnt_violation_code
FROM
 (SELECT cnt_code.issuer_precinct,
        cnt_code.violation_code,
        cnt_code.cnt_violation_code,
        max(cnt_code.cnt_violation_code) OVER (PARTITION BY cnt_code.issuer_precinct) AS max_cnt_violation_code
  FROM
    (SELECT p.issuer_precinct,
          ndp.violation_code,
          count(ndp.violation_code) AS cnt_violation_code
     FROM nyc_data_partitioned_bucketed_orc ndp
     JOIN precinct p ON p.issuer_precinct = ndp.issuer_precinct
     WHERE ndp.violation_code != 0
     GROUP BY p.issuer_precinct,
           ndp.violation_code) cnt_code)cnt_max_code
WHERE cnt_max_code.max_cnt_violation_code = cnt_max_code.cnt_violation_code;
```

4(b).

```
WITH precinct AS
 (SELECT cnt.issuer_precinct,
        cnt.cnt_summons_number
  FROM
    (SELECT issuer_precinct,
          count(summons_number) AS cnt_summons_number
     FROM nyc_data_partitioned_bucketed_orc
     WHERE length(summons_number) = 10
       AND issuer_precinct != 0
     GROUP BY issuer_precinct) cnt
  ORDER BY cnt.cnt_summons_number DESC
  LIMIT 3)
SELECT cnt_max_code.issuer_precinct,
      cnt_max_code.violation_code,
      cnt_max_code.cnt_violation_code
```

```
FROM
 (SELECT cnt_code.issuer_precinct,
       cnt_code.violation_code,
       cnt_code.cnt_violation_code,
       rank() OVER (PARTITION BY cnt_code.issuer_precinct
                ORDER BY cnt_code.cnt_violation_code DESC) AS rank_cnt_violation_code
  FROM
   (SELECT p.issuer_precinct,
         ndp.violation_code,
         count(ndp.violation_code) AS cnt_violation_code
    FROM nyc_data_partitioned_bucketed_orc ndp
    JOIN precinct p ON p.issuer_precinct = ndp.issuer_precinct
    WHERE ndp.violation_code != 0
    GROUP BY p.issuer_precinct,
          ndp.violation_code) cnt_code)cnt_max_code
WHERE cnt_max_code.rank_cnt_violation_code BETWEEN 1 AND 5;
```

5(a).

```
SELECT cnt.v_time,
     count(cnt.violation_code)
FROM
 (SELECT CASE
          WHEN violation_time LIKE '%P'
             AND cast(substr(violation_time, 1, 2) AS int) == 12 THEN cast(substr(violation_time, 1, 2) AS int)
          WHEN violation_time LIKE '%P'
             AND cast(substr(violation_time, 1, 2) AS int)+12 BETWEEN 13 AND 23 THEN cast(substr(violation_time, 1, 2) AS
int)+12
          WHEN violation_time LIKE '%A'
             AND cast(substr(violation_time, 1, 2) AS int) BETWEEN 0 AND 11 THEN cast(substr(violation_time, 1, 2) AS int)
          ELSE 24
       END AS v_time,
       violation_code
  FROM nyc_data_partitioned_bucketed_orc
  WHERE violation_time != ''
    AND violation_time rlike '^[0-9]{4}[A|P]$'
    AND violation_code !=0) AS cnt
WHERE cnt.v_time BETWEEN 0 AND 23
GROUP BY cnt.v_time
ORDER BY cnt.v_time;
```

5(b).

```
SELECT cnt.v_time,
     count(cnt.summons_number)
FROM
 (SELECT CASE
          WHEN violation_time LIKE '%P'
             AND cast(substr(violation_time, 1, 2) AS int) == 12 THEN cast(substr(violation_time, 1, 2) AS int)
          WHEN violation_time LIKE '%P'
             AND cast(substr(violation_time, 1, 2) AS int)+12 BETWEEN 13 AND 23 THEN cast(substr(violation_time, 1, 2) AS
int)+12
          WHEN violation_time LIKE '%A'
             AND cast(substr(violation_time, 1, 2) AS int) BETWEEN 0 AND 11 THEN cast(substr(violation_time, 1, 2) AS int)
          ELSE 24
       END AS v_time,
```

```
        summons_number
   FROM nyc_data_partitioned_bucketed_orc
   WHERE violation_time != ''
    AND violation_time rlike '^[0-9]{4}[A|P]$'
    AND length(summons_number) = 10) AS cnt
WHERE cnt.v_time BETWEEN 0 AND 23
GROUP BY cnt.v_time
ORDER BY cnt.v_time;
```

6.

```
SELECT ranked_bin.time_bin,
      ranked_bin.violation_code
FROM
  (SELECT grouped_bin.time_bin,
         grouped_bin.violation_code,
         grouped_bin.cnt_violation_code,
         rank() OVER (PARTITION BY grouped_bin.time_bin
                  ORDER BY cnt_violation_code DESC) AS rk
   FROM
    (SELECT cnt_bin.time_bin,
          cnt_bin.violation_code,
          count(cnt_bin.violation_code) AS cnt_violation_code
     FROM
      (SELECT CASE
                WHEN cnt.v_time IN (0,
                                    1,
                                    2,
                                    3) THEN 1
                WHEN cnt.v_time IN (4,
                                    5,
                                    6,
                                    7) THEN 2
                WHEN cnt.v_time IN (8,
                                    9,
                                    10,
                                    11) THEN 3
                WHEN cnt.v_time IN (12,
                                    13,
                                    14,
                                    15) THEN 4
                WHEN cnt.v_time IN (16,
                                    17,
                                    18,
                                    19) THEN 5
                WHEN cnt.v_time IN (20,
                                    21,
                                    22,
                                    23) THEN 6
            END AS time_bin,
            cnt.violation_code
       FROM
        (SELECT CASE
                  WHEN violation_time LIKE '%P'
                     AND cast(substr(violation_time, 1, 2) AS int) == 12 THEN cast(substr(violation_time, 1, 2) AS int)
```

```
            WHEN violation_time LIKE '%P'
                AND cast(substr(violation_time, 1, 2) AS int)+12 BETWEEN 13 AND 23 THEN cast(substr(violation_time, 1, 2)
AS int)+12
            WHEN violation_time LIKE '%A'
                AND cast(substr(violation_time, 1, 2) AS int) BETWEEN 0 AND 11 THEN cast(substr(violation_time, 1, 2) AS
int)
            ELSE 24
          END AS v_time,
          violation_code
      FROM nyc_data_partitioned_bucketed_orc
      WHERE violation_time != ''
       AND violation_time rlike '^[0-9]{4}[A|P]$'
       AND violation_code !=0) AS cnt
    WHERE cnt.v_time BETWEEN 0 AND 23 ) cnt_bin
  GROUP BY cnt_bin.time_bin,
        cnt_bin.violation_code) grouped_bin) ranked_bin
WHERE ranked_bin.rk BETWEEN 1 AND 3;
```

7.

```
WITH c_violation_code AS
 (SELECT cnt.violation_code,
      cnt.cnt_violationcode
  FROM
   (SELECT violation_code,
       count(violation_code) AS cnt_violationcode
    FROM nyc_data_partitioned_bucketed_orc
    WHERE violation_code != 0
    GROUP BY violation_code) cnt
  ORDER BY cnt.cnt_violationcode DESC
  LIMIT 3),
  grouped_bin AS
 (SELECT cnt_bin.time_bin,
      cnt_bin.violation_code,
      count(cnt_bin.violation_code) AS cnt_violation_code
  FROM
   (SELECT CASE
           WHEN cnt.v_time IN (0,
                               1,
                               2,
                               3) THEN 1
           WHEN cnt.v_time IN (4,
                               5,
                               6,
                               7) THEN 2
           WHEN cnt.v_time IN (8,
                               9,
                               10,
                               11) THEN 3
           WHEN cnt.v_time IN (12,
                               13,
                               14,
                               15) THEN 4
           WHEN cnt.v_time IN (16,
                               17,
```

```
                    18,
                    19) THEN 5
            WHEN cnt.v_time IN (20,
                    21,
                    22,
                    23) THEN 6
        END AS time_bin,
        cnt.violation_code
    FROM
     (SELECT CASE
            WHEN violation_time LIKE '%P'
               AND cast(substr(violation_time, 1, 2) AS int) == 12 THEN cast(substr(violation_time, 1, 2) AS int)
            WHEN violation_time LIKE '%P'
               AND cast(substr(violation_time, 1, 2) AS int)+12 BETWEEN 13 AND 23 THEN cast(substr(violation_time, 1, 2)
AS int)+12
            WHEN violation_time LIKE '%A'
               AND cast(substr(violation_time, 1, 2) AS int) BETWEEN 0 AND 11 THEN cast(substr(violation_time, 1, 2) AS
int)
            ELSE 24
          END AS v_time,
          violation_code
      FROM nyc_data_partitioned_bucketed_orc
      WHERE violation_time != ''
       AND violation_time rlike '^[0-9]{4}[A|P]$'
       AND violation_code != 0) AS cnt
    WHERE cnt.v_time BETWEEN 0 AND 23 ) cnt_bin
  GROUP BY cnt_bin.time_bin,
        cnt_bin.violation_code)
SELECT final.violation_code,
     final.time_bin
FROM
 (SELECT gb.time_bin,
       gb.violation_code,
       gb.cnt_violation_code,
       max(gb.cnt_violation_code) OVER (PARTITION BY gb.violation_code) AS max_violation_code
  FROM grouped_bin gb
  JOIN c_violation_code cvc ON cvc.violation_code = gb.violation_code) FINAL
WHERE final.cnt_violation_code = final.max_violation_code;
```

8(a).

```
SELECT grouped_season.season,
     grouped_season.cnt_tickets
FROM
 (SELECT seasoned.season,
       count(seasoned.summons_number) AS cnt_tickets
  FROM
   (SELECT CASE
            WHEN month_vc.month_issue_date IN (3,
                                               4,
                                               5) THEN 'Spring'
            WHEN month_vc.month_issue_date IN (6,
                                               7,
                                               8) THEN 'Summer'
            WHEN month_vc.month_issue_date IN (9,
```

```
                                10,
                                11) THEN 'Fall'
            WHEN month_vc.month_issue_date IN (12,
                                1,
                                2) THEN 'Winter'
          END AS season,
          month_vc.summons_number
    FROM
      (SELECT month(issue_date) month_issue_date,
            summons_number
       FROM nyc_data_partitioned_bucketed_orc
       WHERE length(summons_number) = 10) AS month_vc) AS seasoned
  GROUP BY seasoned.season) AS grouped_season
ORDER BY grouped_season.season,
       grouped_season.cnt_tickets;
```

8(b).

```
SELECT ranked.season,
     ranked.violation_code
FROM
 (SELECT cnt.season,
        cnt.violation_code,
        cnt.cnt_violation_code,
        rank() OVER (PARTITION BY cnt.season
                ORDER BY cnt.cnt_violation_code DESC) AS rk
  FROM
    (SELECT seasoned.season,
          seasoned.violation_code,
          count(seasoned.violation_code) AS cnt_violation_code
     FROM
      (SELECT CASE
              WHEN month_vc.month_issue_date IN (3,
                                4,
                                5) THEN 'Spring'
              WHEN month_vc.month_issue_date IN (6,
                                7,
                                8) THEN 'Summer'
              WHEN month_vc.month_issue_date IN (9,
                                10,
                                11) THEN 'Fall'
              WHEN month_vc.month_issue_date IN (12,
                                1,
                                2) THEN 'Winter'
            END AS season,
            month_vc.violation_code
       FROM
        (SELECT month(issue_date) month_issue_date,
              violation_code
         FROM nyc_data_partitioned_bucketed_orc
         WHERE violation_code !=0) AS month_vc) AS seasoned
     GROUP BY seasoned.season,
            seasoned.violation_code) AS cnt) AS ranked
WHERE ranked.rk BETWEEN 1 AND 3;
```