CREATE TABLE ON CSV DATA

```
CREATE EXTERNAL TABLE IF NOT EXISTS nyc_parking_violations(
`Summons Number`  bigint,
`Plate ID`  string,
`Registration State`  string,
`Plate Type`  string,
`Issue Date`  string,
`Violation Code`  int,
`Vehicle Body Type`  string,
`Vehicle Make`  string,
`Issuing Agency`  string,
`Street Code1`  int  ,
`Street Code2`  int,
`Street Code3`  int,
`Vehicle Expiration Date`  int,
`Violation Location`  string,
`Violation Precinct`  int,
`Issuer Precinct`  int,
`Issuer Code`  bigint,
`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 ',' LINES TERMINATED BY '\n' STORED
AS TEXTFILE
LOCATION 's3://yourbucket/nyc/nyc_parking_violations/'
TBLPROPERTIES ( "skip.header.line.count"="1");
```

Create a paritioned ORC table for efficiency

```
drop table nyc_parking_violations_orc;
CREATE EXTERNAL TABLE IF NOT EXISTS nyc_parking_violations_orc( `Summons
Number`  bigint, `Plate ID`  string, `Registration State`  string, `Plate
Type`  string, `Issue Date`  string, `Violation Code`  int, `Vehicle Body
Type`  string, `Vehicle Make`  string, `Issuing Agency`  string, `Street
Code1`  int  , `Street Code2`  int, `Street Code3`  int, `Vehicle Expiration
Date`  int, `Violation Location`  string, `Violation Precinct`  int, `Issuer
Precinct`  int, `Issuer Code`  bigint, `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 (month string)
STORED AS ORC
LOCATION 's3://yourbucket/nyc/nyc_parking_violations_orc/' TBLPROPERTIES (
'orc.compress'='SNAPPY');

set hive.exec.dynamic.partition.mode=nonstrict;
INSERT OVERWRITE TABLE nyc_parking_violations_orc PARTITION(month)
SELECT *, CONCAT(SUBSTR(`ISSUE DATE`,7,4), SUBSTR(`ISSUE DATE`,1,2)) AS MONTH
FROM nyc_parking_violations
where (`ISSUE DATE` like '%2017')
;


Solutions to questions
Part-I:

1.Find total number of tickets for each year.

select count(*) AS No_Of_Tickets from parkingViolationData2017;

2.Find out how many unique states the cars which got parking tickets came
from.
select count(distinct `Registration State`)
from nyc_parking_violations_orc
;

3Some parking tickets donâ€™t have addresses on them, which is cause for
concern. Find out how many such tickets there are.

select count(*)
from nyc_parking_violations_orc
where `Street Code1` is null or `Street Code2` is null `Street Code3` is
null;

Part-II

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

```
select "Violation Code", count(*)
from nyc_parking_violations_orc
group by "Violation Code" order by count(*) desc
limit 5
;
```

2.How often does each vehicle body type get a parking ticket? How about the vehicle make?

```
select "Vehicle Body Type", count(*)
from nyc_parking_violations_orc
group by "Vehicle Body Type" order by count(*) desc
limit 5
;
select "Vehicle Make", count(*)
from nyc_parking_violations_orc
group by "Vehicle Make" order by count(*) desc
limit 5
;
```

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

```
-- Violating Precincts (this is the precinct of the zone where the violation occurred)
-- Issuing Precincts (this is the precinct that issued the ticket)
select "Violation Precinct", count(*)
from nyc_parking_violations_orc
group by "Violation Precinct" order by count(*) desc
limit 5
;

select "Issuer Precinct", count(*)
from nyc_parking_violations_orc
group by "Issuer Precinct" order by count(*) desc
limit 5
;
```

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

```
select * from
(
  select "Issuer Precinct", rank() over (partition by null order by count(*) desc) as vprnk
  from nyc_parking_violations_orc
  group by "Issuer Precinct"
)a
left outer join
(
  select "Issuer Precinct", "Violation Code", rank() over (partition by "Issuer Precinct" order by count(*) desc) as vp_vc_rnk
  from nyc_parking_violations_orc
  group by "Issuer Precinct", "Violation Code"
)b
on(a."Issuer Precinct" = b."Issuer Precinct")
where vprnk <=6 and vp_vc_rnk <=5
;
```

5.Find out the properties of parking violations across different times of the day: The Violation Time field is specified in a strange format. Find a way to make this into a time attribute that you can use to divide into groups.
--Violation Time seems to be following this format: hhmm followed by a or p depending on AM/PM. Following function converts it into hive datetime object. We can then derive hour from it.

```
select hour(from_unixtime(unix_timestamp(concat(`Issue Date`,' ',`Violation Time`,'M'), 'MM/dd/yyyy hhmma'))) as time_of_day, count(*)
from nyc_parking_violations_orc
group by hour(from_unixtime(unix_timestamp(concat(`Issue Date`,' ',`Violation Time`,'M'), 'MM/dd/yyyy hhmma')))
order by time_of_day
;
```

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

```
select *
from
(
select time_period, vc, rank() over (partition by time_period order by count(*) desc) as cv_rnk, count(*) as n_violations
from
(
  select round(hour(from_unixtime(unix_timestamp(concat(`Issue Date`,' ',`Violation Time`,'M'), 'MM/dd/yyyy hhmma')))/4) as time_period,  `Violation Code` as vc
  from nyc_parking_violations_orc
  where month = '201710'
)a
group by time_period, vc
)b
where cv_rnk < 4
order by time_period asc,  cv_rnk asc
;
```

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

```
select *
from
(
    select a.vc as vc, time_period, rank() over (partition by a.vc order by sum(time_period) desc) as vc_time_period_rnk
    from
    (
      select `Violation Code` as vc, rank() over (partition by null order by count(*) desc) as vc_rnk
```

```
      from nyc_parking_violations_orc
      where month  = '201710'
      group by `Violation Code`
    )a
    left outer join
    (
      select round(hour(from_unixtime(unix_timestamp(concat(`Issue Date`,'
',`Violation Time`,'M'), 'MM/dd/yyyy hhmma')))/4) as time_period,  `Violation
Code` as vc, count(*) as vc_time_period_count
      from nyc_parking_violations_orc
      where month = '201710'
      group by round(hour(from_unixtime(unix_timestamp(concat(`Issue Date`,'
',`Violation Time`,'M'), 'MM/dd/yyyy hhmma')))/4) ,  `Violation Code`
    )b
    on(a.vc = b.vc)
    where vc_rnk < 4
    group by a.vc, time_period
)c
where vc_time_period_rnk < 4


8.Letâ€™s try and find some seasonality in this data
--First, divide the year into some number of seasons, and find frequencies of
tickets for each season.

--A quick google search reveals following season calendar for New York
--Spring - March, April, May
--Summer - June, July, August
--Fall - September, October, November
-- Winter - December, January, February
select case when substr(month,5,2) in ('03','04','05') then 'spring'
            when substr(month,5,2) in ('06','07','08') then 'summer'
            when substr(month,5,2) in ('09','10','11') then 'fall'
            when substr(month,5,2) in ('12','01','02') then 'winter'
            end as season,
    count(*)
from nyc_parking_violations_orc
group by case when substr(month,5,2) in ('03','04','05') then 'spring'
            when substr(month,5,2) in ('06','07','08') then 'summer'
            when substr(month,5,2) in ('09','10','11') then 'fall'
            when substr(month,5,2) in ('12','01','02') then 'winter'
            end
;

--Then, find the 3 most common violations for each of these season

select *
from
(
    select *, rank() over (partition by season order by vc_count desc) as
vc_rnk
    from
    (
        select case when substr(month,5,2) in ('03','04','05') then 'spring'
                    when substr(month,5,2) in ('06','07','08') then 'summer'
                    when substr(month,5,2) in ('09','10','11') then 'fall'
                    when substr(month,5,2) in ('12','01','02') then 'winter'
```

```
                    end as season,
            `Violation Code`, count(*) as vc_count
        from nyc_parking_violations_orc
        group by case when substr(month,5,2) in ('03','04','05') then
'spring'
                    when substr(month,5,2) in ('06','07','08') then 'summer'
                    when substr(month,5,2) in ('09','10','11') then 'fall'
                    when substr(month,5,2) in ('12','01','02') then 'winter'
                    end,
            `Violation Code`
    )a
)b
where vc_rnk < 4
;
```