HIVE-MINI-PROJECT-2

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

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

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

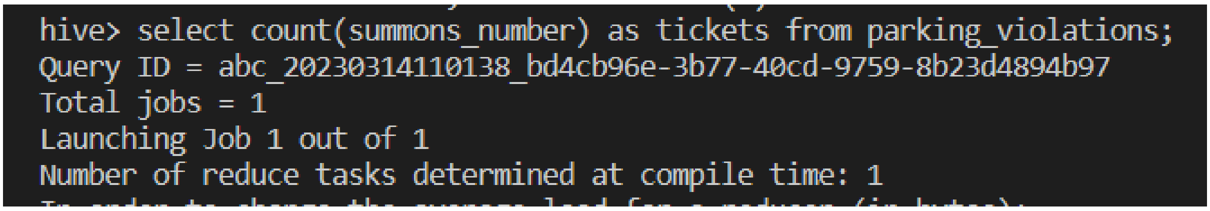
Download Dataset - https://data.cityofnewyork.us/browse?q=parking+tickets

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

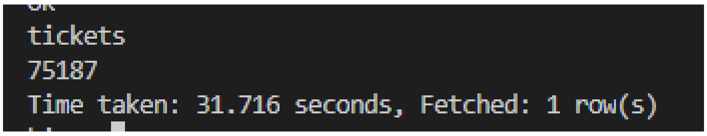
The analysis can be divided into two parts:

Part-I: Examine the data

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

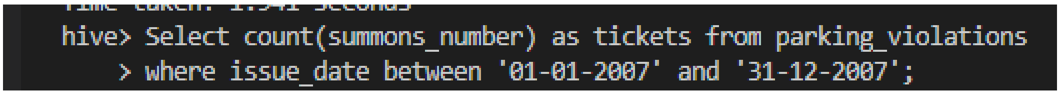


```
hive> select count(summons_number) as tickets from parking_violations;
Query ID = abc_20230314110138_bd4cb96e-3b77-40cd-9759-8b23d4894b97
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
```

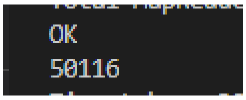


```
tickets
75187
Time taken: 31.716 seconds, Fetched: 1 row(s)
```

For a particular year excluding 2006

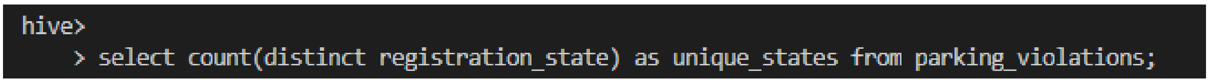


```
hive> Select count(summons_number) as tickets from parking_violations
    > where issue_date between '01-01-2007' and '31-12-2007';
```

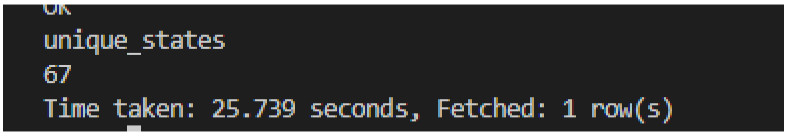


```
OK
50116
```

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

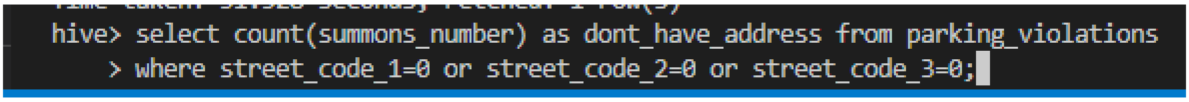


```
hive>
    > select count(distinct registration_state) as unique_states from parking_violations;
```

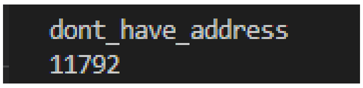


```
unique_states
67
Time taken: 25.739 seconds, Fetched: 1 row(s)
```

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



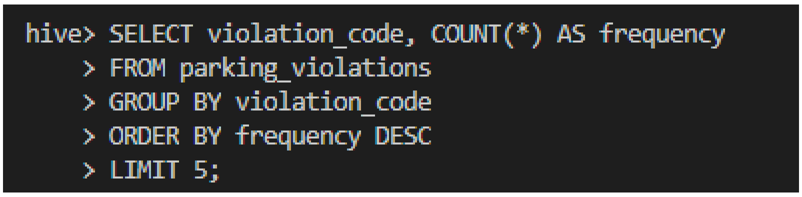
```
hive> select count(summons_number) as dont_have_address from parking_violations
    > where street_code_1=0 or street_code_2=0 or street_code_3=0;
```



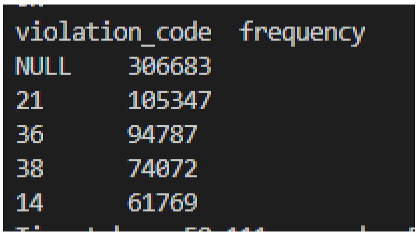
```
dont_have_address
11792
```

Part-II: Aggregation tasks

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

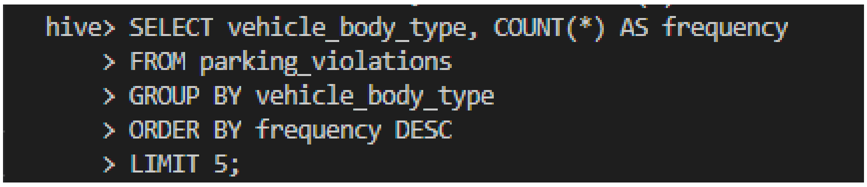


```
hive> SELECT violation_code, COUNT(*) AS frequency
    > FROM parking_violations
    > GROUP BY violation_code
    > ORDER BY frequency DESC
    > LIMIT 5;
```

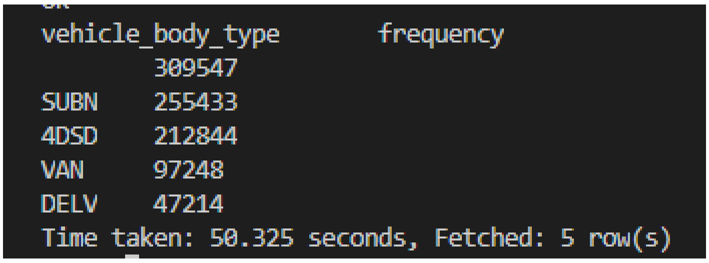


```
violation_code  frequency
NULL    306683
21      105347
36      94787
38      74072
14      61769
```

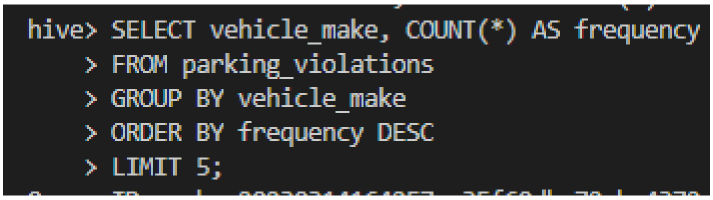
2.) How often does each vehicle body type get a parking ticket? How about the vehicle make? (find the top 5 for both)



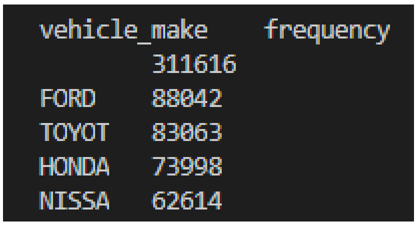
```
hive> SELECT vehicle_body_type, COUNT(*) AS frequency
    > FROM parking_violations
    > GROUP BY vehicle_body_type
    > ORDER BY frequency DESC
    > LIMIT 5;
```



```
vehicle_body_type       frequency
        309547
SUBN    255433
4DSD    212844
VAN     97248
DELV    47214
Time taken: 50.325 seconds, Fetched: 5 row(s)
```



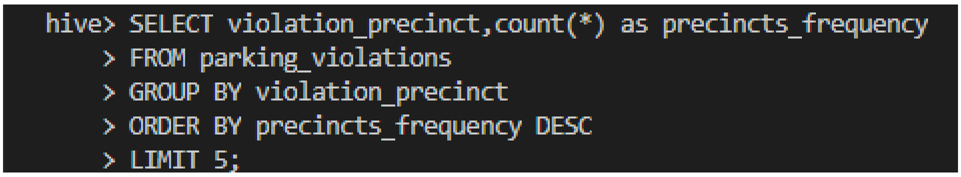
```
hive> SELECT vehicle_make, COUNT(*) AS frequency
    > FROM parking_violations
    > GROUP BY vehicle_make
    > ORDER BY frequency DESC
    > LIMIT 5;
```



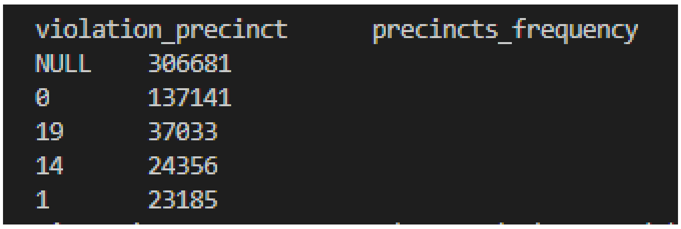
```
vehicle_make    frequency
        311616
FORD    88042
TOYOT   83063
HONDA   73998
NISSA   62614
```

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

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

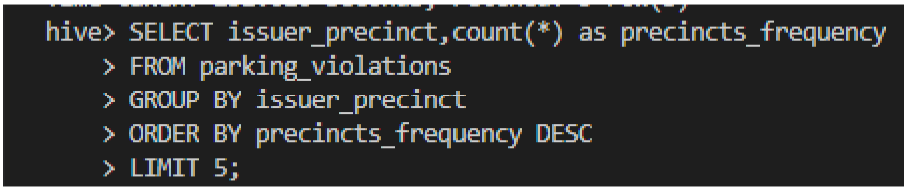


```
hive> SELECT violation_precinct,count(*) as precincts_frequency
    > FROM parking_violations
    > GROUP BY violation_precinct
    > ORDER BY precincts_frequency DESC
    > LIMIT 5;
```

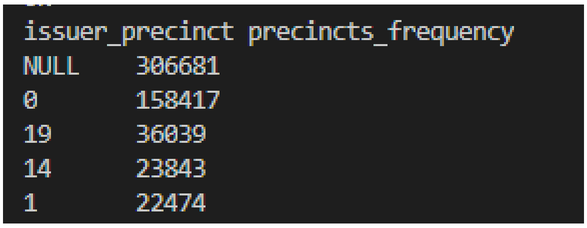


```
violation_precinct      precincts_frequency
NULL    306681
0       137141
19      37033
14      24356
1       23185
```

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

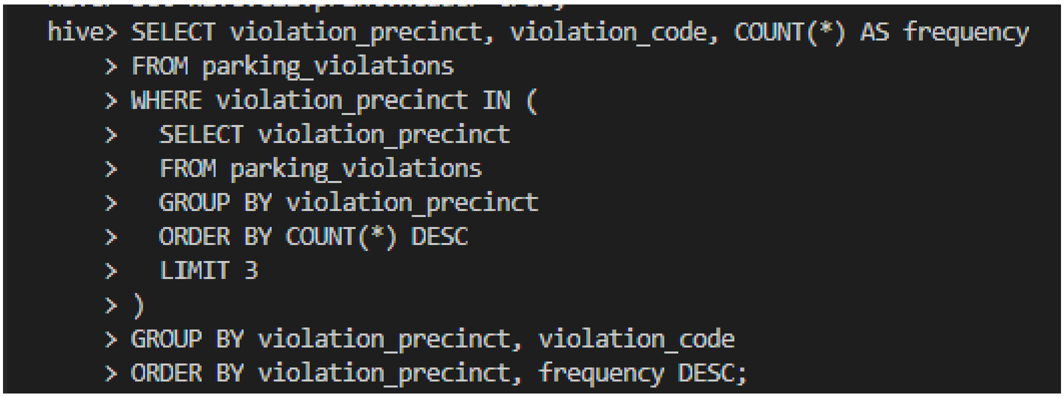


```
hive> SELECT issuer_precinct,count(*) as precincts_frequency
    > FROM parking_violations
    > GROUP BY issuer_precinct
    > ORDER BY precincts_frequency DESC
    > LIMIT 5;
```

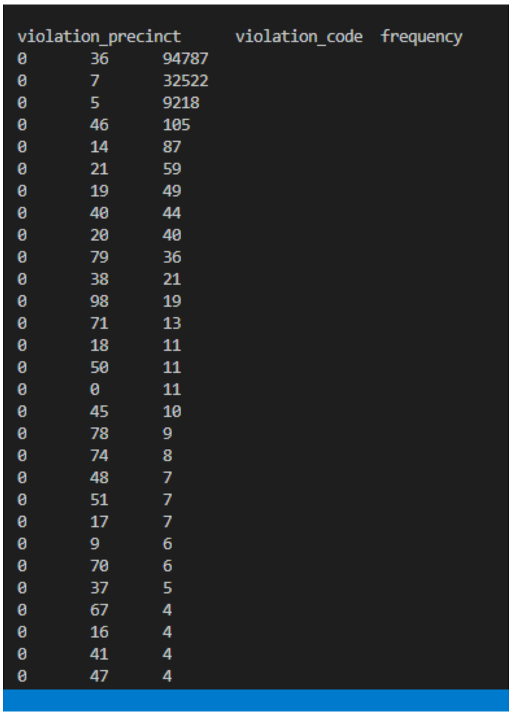


```
issuer_precinct precincts_frequency
NULL    306681
0       158417
19      36039
14      23843
1       22474
```

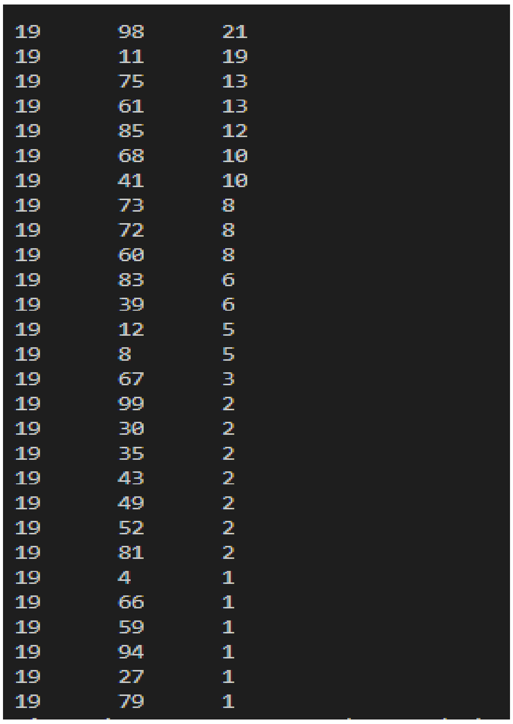
4.) Find the violation code frequency across 3 precincts which have issued the most number of tickets - do these precinct zones have an exceptionally high frequency of certain violation codes?



```
hive> SELECT violation_precinct, violation_code, COUNT(*) AS frequency
    > FROM parking_violations
    > WHERE violation_precinct IN (
    >   SELECT violation_precinct
    >   FROM parking_violations
    >   GROUP BY violation_precinct
    >   ORDER BY COUNT(*) DESC
    >   LIMIT 3
    > )
    > GROUP BY violation_precinct, violation_code
    > ORDER BY violation_precinct, frequency DESC;
```

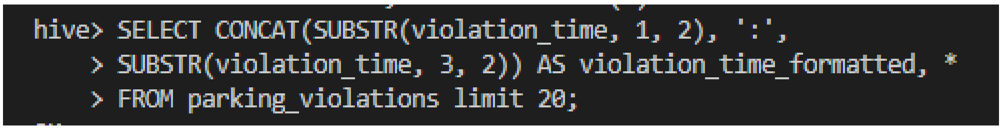


```
violation_precinct      violation_code  frequency
0       36      94787
0       7       32522
0       5       9218
0       46      105
0       14      87
0       21      59
0       19      49
0       40      44
0       20      40
0       79      36
0       38      21
0       98      19
0       71      13
0       18      11
0       50      11
0       0       11
0       45      10
0       78      9
0       74      8
0       48      7
0       51      7
0       17      7
0       9       6
0       70      6
0       37      5
0       67      4
0       16      4
0       41      4
0       47      4
```

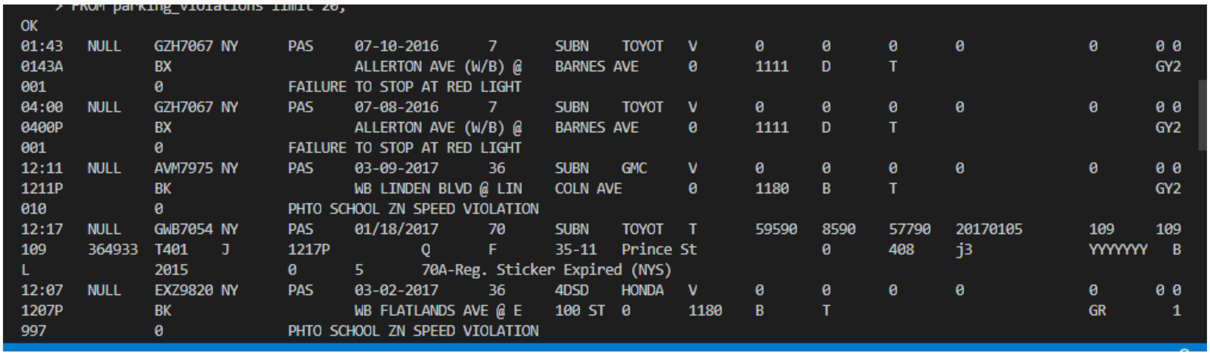


```
19      98      21
19      11      19
19      75      13
19      61      13
19      85      12
19      68      10
19      41      10
19      73      8
19      72      8
19      60      8
19      83      6
19      39      6
19      12      5
19      8       5
19      67      3
19      99      2
19      30      2
19      35      2
19      43      2
19      49      2
19      52      2
19      81      2
19      4       1
19      66      1
19      59      1
19      94      1
19      27      1
19      79      1
```

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

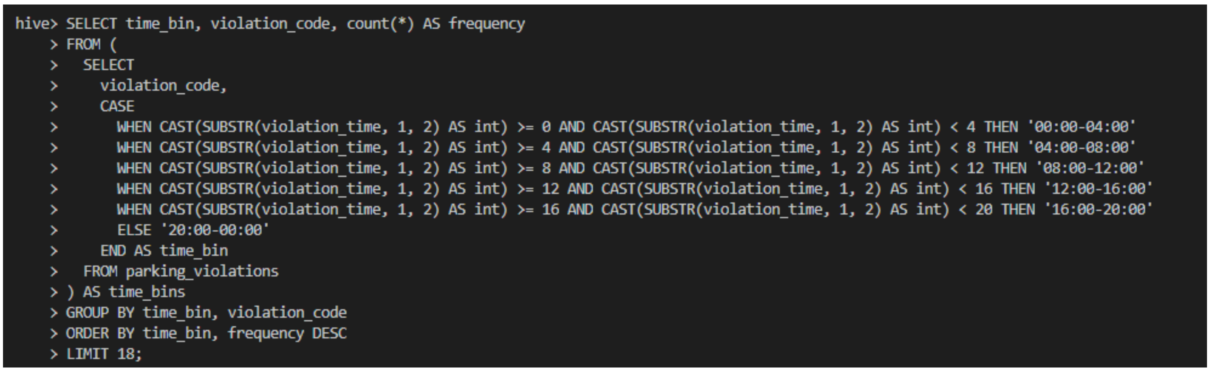


```
hive> SELECT CONCAT(SUBSTR(violation_time, 1, 2), ':',
    > SUBSTR(violation_time, 3, 2)) AS violation_time_formatted, *
    > FROM parking_violations limit 20;
```

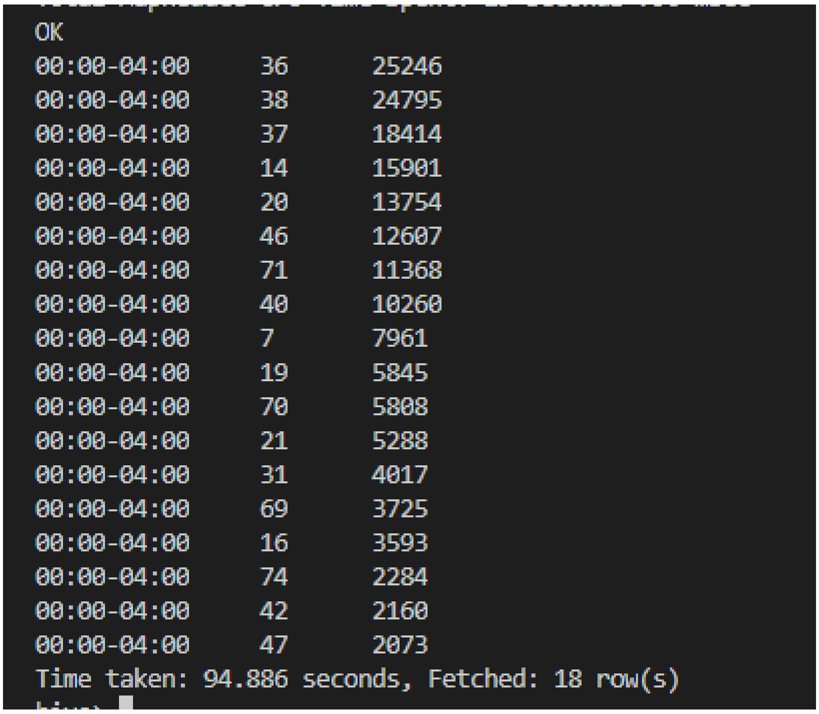


```
    > FROM parking_violations limit 20;
OK
01:43   NULL    GZH7067 NY      PAS     07-10-2016      7       SUBN    TOYOT   V       0       0       0       0               0       0 0
0143A           BX                      ALLERTON AVE (W/B) @    BARNES AVE      0       1111    D       T                               GY2
001             0               FAILURE TO STOP AT RED LIGHT
04:00   NULL    GZH7067 NY      PAS     07-08-2016      7       SUBN    TOYOT   V       0       0       0       0               0       0 0
0400P           BX                      ALLERTON AVE (W/B) @    BARNES AVE      0       1111    D       T                               GY2
001             0               FAILURE TO STOP AT RED LIGHT
12:11   NULL    AVM7975 NY      PAS     03-09-2017      36      SUBN    GMC     V       0       0       0       0               0       0 0
1211P           BK                      WB LINDEN BLVD @ LIN    COLN AVE        0       1180    B       T                               GY2
010             0               PHTO SCHOOL ZN SPEED VIOLATION
12:17   NULL    GWB7054 NY      PAS     01/18/2017      70      SUBN    TOYOT   T       59590   8590    57790   20170105        109     109
109     364933  T401    J       1217P           Q       F       35-11   Prince St               0       408     j3              YYYYYYY   B
L               2015            0       5       70A-Reg. Sticker Expired (NYS)
12:07   NULL    EXZ9820 NY      PAS     03-02-2017      36      4DSD    HONDA   V       0       0       0       0               0       0 0
1207P           BK                      WB FLATLANDS AVE @ E    100 ST  0       1180    B       T                       GR              1
997             0               PHTO SCHOOL ZN SPEED VIOLATION
```

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

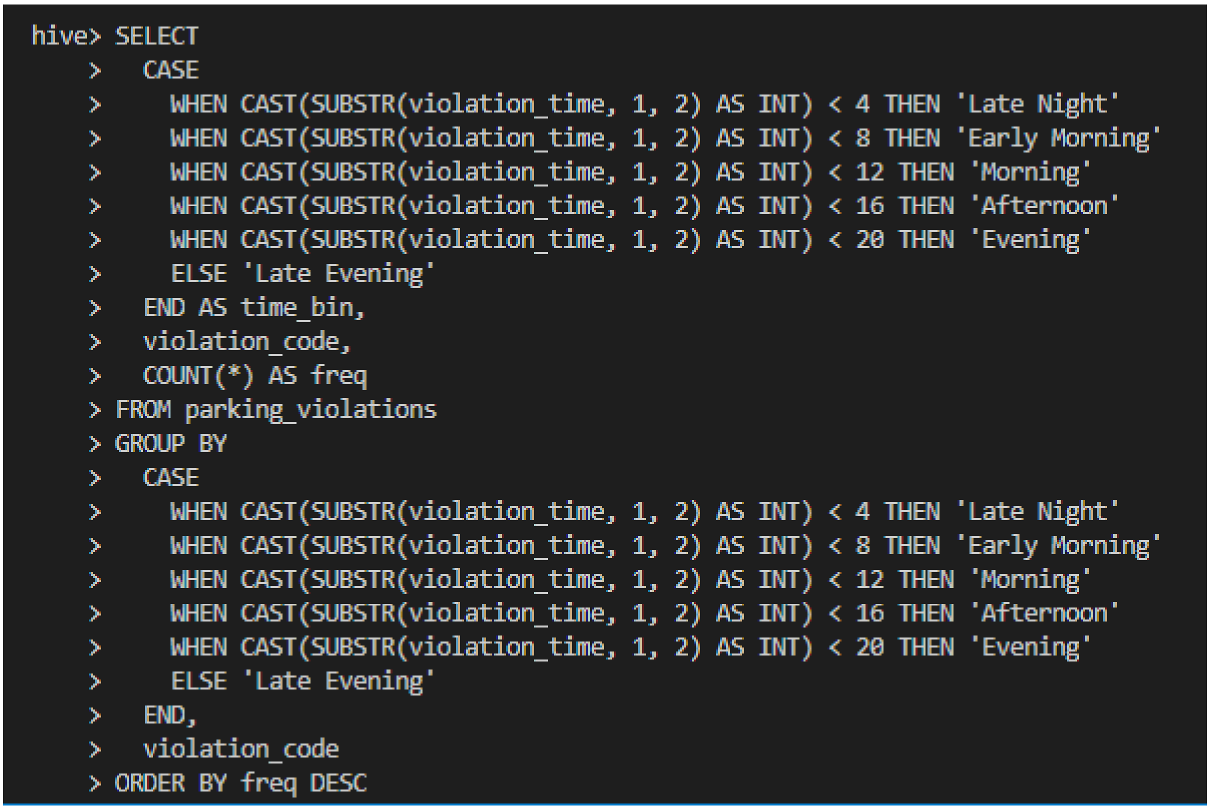


```
hive> SELECT time_bin, violation_code, count(*) AS frequency
    > FROM (
    >   SELECT
    >     violation_code,
    >     CASE
    >       WHEN CAST(SUBSTR(violation_time, 1, 2) AS int) >= 0 AND CAST(SUBSTR(violation_time, 1, 2) AS int) < 4 THEN '00:00-04:00'
    >       WHEN CAST(SUBSTR(violation_time, 1, 2) AS int) >= 4 AND CAST(SUBSTR(violation_time, 1, 2) AS int) < 8 THEN '04:00-08:00'
    >       WHEN CAST(SUBSTR(violation_time, 1, 2) AS int) >= 8 AND CAST(SUBSTR(violation_time, 1, 2) AS int) < 12 THEN '08:00-12:00'
    >       WHEN CAST(SUBSTR(violation_time, 1, 2) AS int) >= 12 AND CAST(SUBSTR(violation_time, 1, 2) AS int) < 16 THEN '12:00-16:00'
    >       WHEN CAST(SUBSTR(violation_time, 1, 2) AS int) >= 16 AND CAST(SUBSTR(violation_time, 1, 2) AS int) < 20 THEN '16:00-20:00'
    >       ELSE '20:00-00:00'
    >     END AS time_bin
    >   FROM parking_violations
    > ) AS time_bins
    > GROUP BY time_bin, violation_code
    > ORDER BY time_bin, frequency DESC
    > LIMIT 18;
```

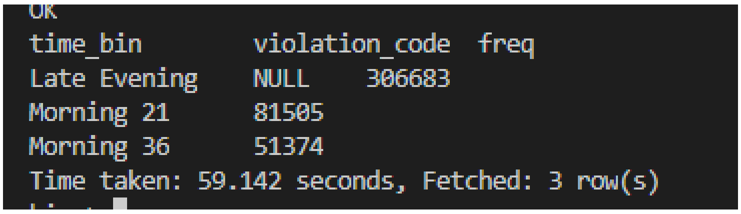


```
OK
00:00-04:00     36      25246
00:00-04:00     38      24795
00:00-04:00     37      18414
00:00-04:00     14      15901
00:00-04:00     20      13754
00:00-04:00     46      12607
00:00-04:00     71      11368
00:00-04:00     40      10260
00:00-04:00     7       7961
00:00-04:00     19      5845
00:00-04:00     70      5808
00:00-04:00     21      5288
00:00-04:00     31      4017
00:00-04:00     69      3725
00:00-04:00     16      3593
00:00-04:00     74      2284
00:00-04:00     42      2160
00:00-04:00     47      2073
Time taken: 94.886 seconds, Fetched: 18 row(s)
hive>
```

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



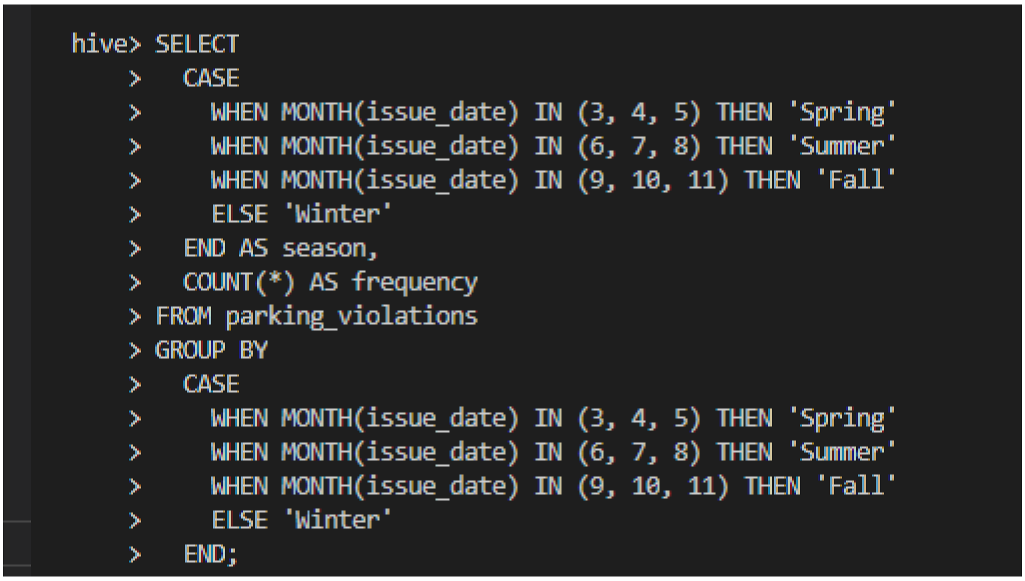
```
hive> SELECT
    >   CASE
    >     WHEN CAST(SUBSTR(violation_time, 1, 2) AS INT) < 4 THEN 'Late Night'
    >     WHEN CAST(SUBSTR(violation_time, 1, 2) AS INT) < 8 THEN 'Early Morning'
    >     WHEN CAST(SUBSTR(violation_time, 1, 2) AS INT) < 12 THEN 'Morning'
    >     WHEN CAST(SUBSTR(violation_time, 1, 2) AS INT) < 16 THEN 'Afternoon'
    >     WHEN CAST(SUBSTR(violation_time, 1, 2) AS INT) < 20 THEN 'Evening'
    >     ELSE 'Late Evening'
    >   END AS time_bin,
    >   violation_code,
    >   COUNT(*) AS freq
    > FROM parking_violations
    > GROUP BY
    >   CASE
    >     WHEN CAST(SUBSTR(violation_time, 1, 2) AS INT) < 4 THEN 'Late Night'
    >     WHEN CAST(SUBSTR(violation_time, 1, 2) AS INT) < 8 THEN 'Early Morning'
    >     WHEN CAST(SUBSTR(violation_time, 1, 2) AS INT) < 12 THEN 'Morning'
    >     WHEN CAST(SUBSTR(violation_time, 1, 2) AS INT) < 16 THEN 'Afternoon'
    >     WHEN CAST(SUBSTR(violation_time, 1, 2) AS INT) < 20 THEN 'Evening'
    >     ELSE 'Late Evening'
    >   END,
    >   violation_code
    > ORDER BY freq DESC
```



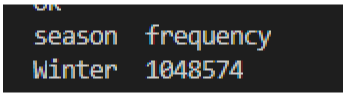
```
OK
time_bin        violation_code  freq
Late Evening    NULL    306683
Morning 21      81505
Morning 36      51374
Time taken: 59.142 seconds, Fetched: 3 row(s)
```

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

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

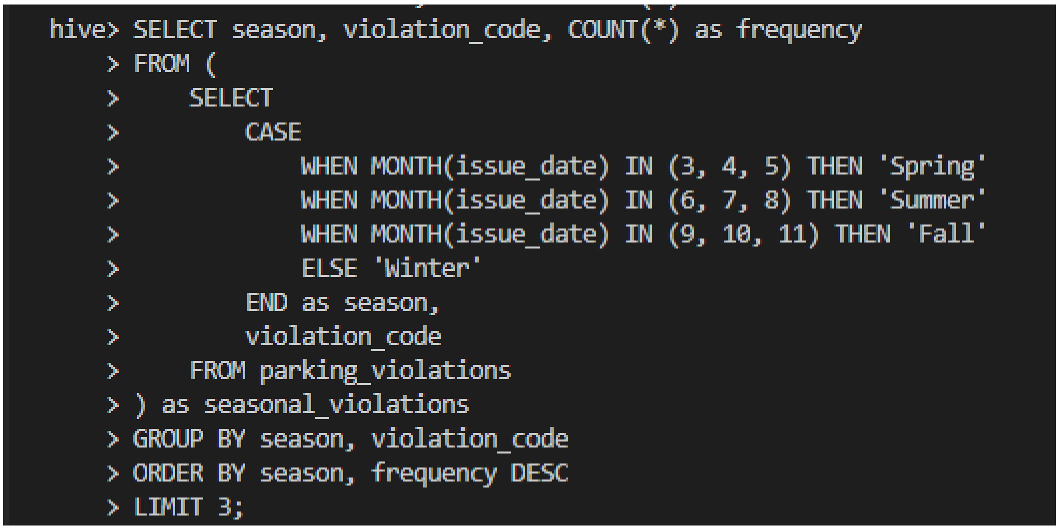


```
hive> SELECT
    >   CASE
    >     WHEN MONTH(issue_date) IN (3, 4, 5) THEN 'Spring'
    >     WHEN MONTH(issue_date) IN (6, 7, 8) THEN 'Summer'
    >     WHEN MONTH(issue_date) IN (9, 10, 11) THEN 'Fall'
    >     ELSE 'Winter'
    >   END AS season,
    >   COUNT(*) AS frequency
    > FROM parking_violations
    > GROUP BY
    >   CASE
    >     WHEN MONTH(issue_date) IN (3, 4, 5) THEN 'Spring'
    >     WHEN MONTH(issue_date) IN (6, 7, 8) THEN 'Summer'
    >     WHEN MONTH(issue_date) IN (9, 10, 11) THEN 'Fall'
    >     ELSE 'Winter'
    >   END;
```

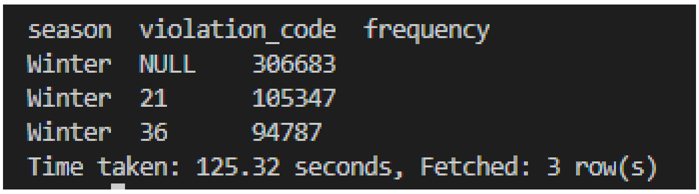


```
season  frequency
Winter  1048574
```

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



```
hive> SELECT season, violation_code, COUNT(*) as frequency
    > FROM (
    >     SELECT
    >         CASE
    >             WHEN MONTH(issue_date) IN (3, 4, 5) THEN 'Spring'
    >             WHEN MONTH(issue_date) IN (6, 7, 8) THEN 'Summer'
    >             WHEN MONTH(issue_date) IN (9, 10, 11) THEN 'Fall'
    >             ELSE 'Winter'
    >         END as season,
    >         violation_code
    >     FROM parking_violations
    > ) as seasonal_violations
    > GROUP BY season, violation_code
    > ORDER BY season, frequency DESC
    > LIMIT 3;
```



```
season  violation_code  frequency
Winter  NULL    306683
Winter  21      105347
Winter  36      94787
Time taken: 125.32 seconds, Fetched: 3 row(s)
```