**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 for year 2017 only/

create table parking\_violation\_temp

(

summons\_number bigint,

plate\_id string,

registration\_state string,

plate\_type string,

issue\_date date,

violation\_code int,

vehicle\_body\_type string,

vehicle\_make string,

issuing\_agency string,

street\_code\_1 int,

street\_code\_2 int,

street\_code\_3 int,

vehicle\_expiration\_date date,

violation\_location int,

violation\_precinct int,

issuer\_precinct int,

issuer\_code int,

issuer\_command string,

issuer\_squad string,

violation\_time string,

time\_first\_observed string,

violation\_country string,

violation\_in\_front\_of\_or\_opposite string,

house\_number string,

street\_name string,

intersecting\_street string,

date\_first\_observed date,

law\_section int,

sub\_division string,

violation\_legal\_code string,

days\_parking\_in\_effect string,

from\_hours\_in\_effect string,

to\_hours\_in\_effect string,

vehicle\_color string,

unregistered\_vehicle int,

vehicle\_year int,

meter\_number string,

feet\_from\_curb int,

violation\_post\_code string,

violation\_description string,

no\_standing\_stopping\_violation string,

hydrant\_violation string,

double\_parking\_violation string

)

row format serde 'org.apache.hadoop.hive.serde2.OpenCSVSerde'

with serdeproperties(

"separatorChar" = ",",

"quoteChar" = "\"",

"escapeChar" = "\\"

)

stored as textfile

tblproperties("skip.header.line.count" = "1");

hive> load data local inpath '/home/cloudera/data/Parking\_Violations\_Issued\_-\_Fiscal\_Year\_2017.csv' into table parking\_violation\_temp;

create table parking\_violation

(

summons\_number bigint,

plate\_id string,

registration\_state string,

plate\_type string,

issue\_date date,

violation\_code int,

vehicle\_body\_type string,

vehicle\_make string,

issuing\_agency string,

street\_code\_1 int,

street\_code\_2 int,

street\_code\_3 int,

vehicle\_expiration\_date date,

violation\_location int,

violation\_precinct int,

issuer\_precinct int,

issuer\_code int,

issuer\_command string,

issuer\_squad string,

violation\_time string,

time\_first\_observed string,

violation\_in\_front\_of\_or\_opposite string,

house\_number string,

street\_name string,

intersecting\_street string,

date\_first\_observed date,

law\_section int,

sub\_division string,

violation\_legal\_code string,

days\_parking\_in\_effect string,

from\_hours\_in\_effect string,

to\_hours\_in\_effect string,

vehicle\_color string,

unregistered\_vehicle int,

vehicle\_year int,

meter\_number string,

feet\_from\_curb int,

violation\_post\_code string,

violation\_description string,

no\_standing\_stopping\_violation string,

hydrant\_violation string,

double\_parking\_violation string

)

partitioned by (violation\_country string)

clustered by (violation\_code)

sorted by (violation\_code)

into 4 buckets

tblproperties("skip.header.line.count" = "1");

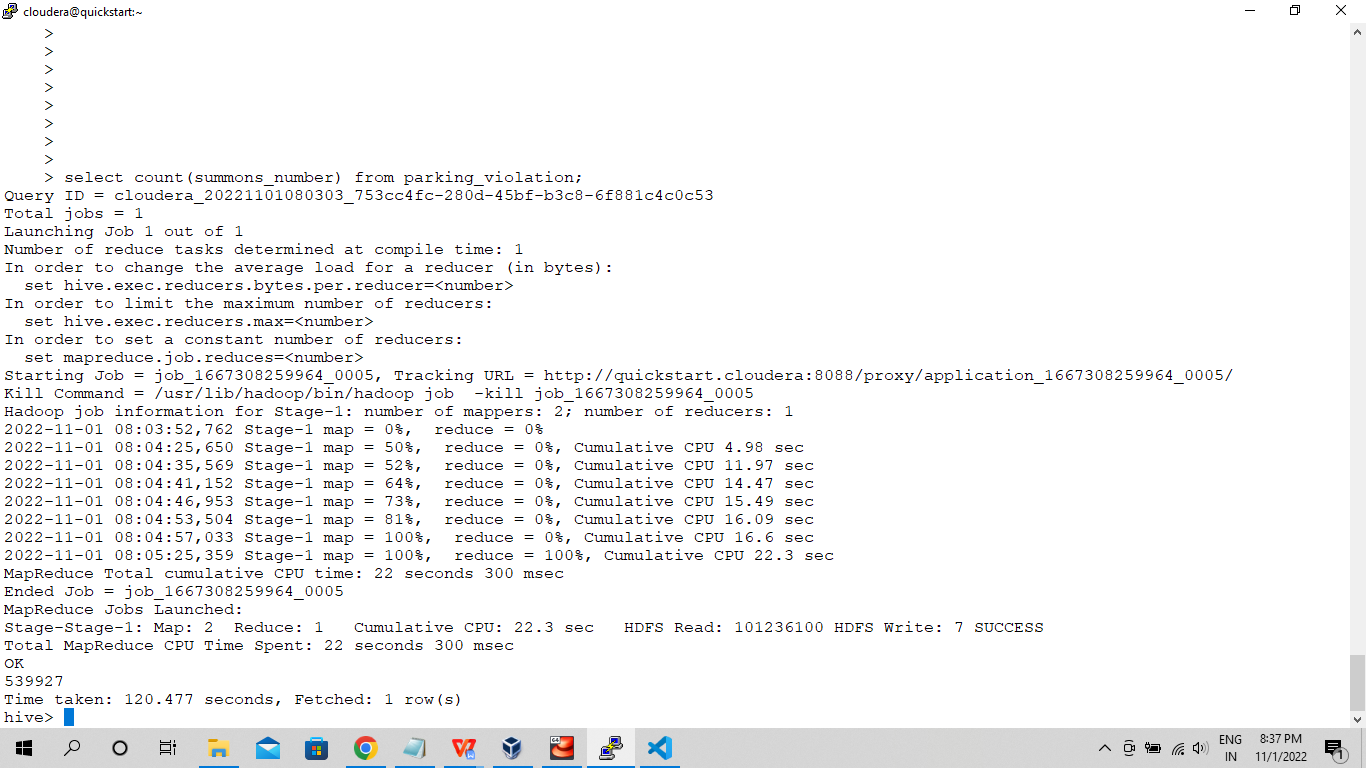
insert overwrite table parking\_violation partition(violation\_country) select summons\_number, plate\_id, registration\_state, plate\_type, issue\_date, violation\_code, vehicle\_body\_type, vehicle\_make, issuing\_agency, street\_code\_1, street\_code\_2, street\_code\_3, vehicle\_expiration\_date, violation\_location, violation\_precinct, issuer\_precinct, issuer\_code, issuer\_command, issuer\_squad, violation\_time, time\_first\_observed , violation\_in\_front\_of\_or\_opposite, house\_number, street\_name, intersecting\_street, date\_first\_observed, law\_section, sub\_division, violation\_legal\_code, days\_parking\_in\_effect, from\_hours\_in\_effect, to\_hours\_in\_effect, vehicle\_color, unregistered\_vehicle, vehicle\_year, meter\_number, feet\_from\_curb, violation\_post\_code, violation\_description, no\_standing\_stopping\_violation, hydrant\_violation, double\_parking\_violation, violation\_country from parking\_violation\_temp where year(issue\_date) = '2017';

The analysis can be divided into two parts:

**Part-I: Examine the data**

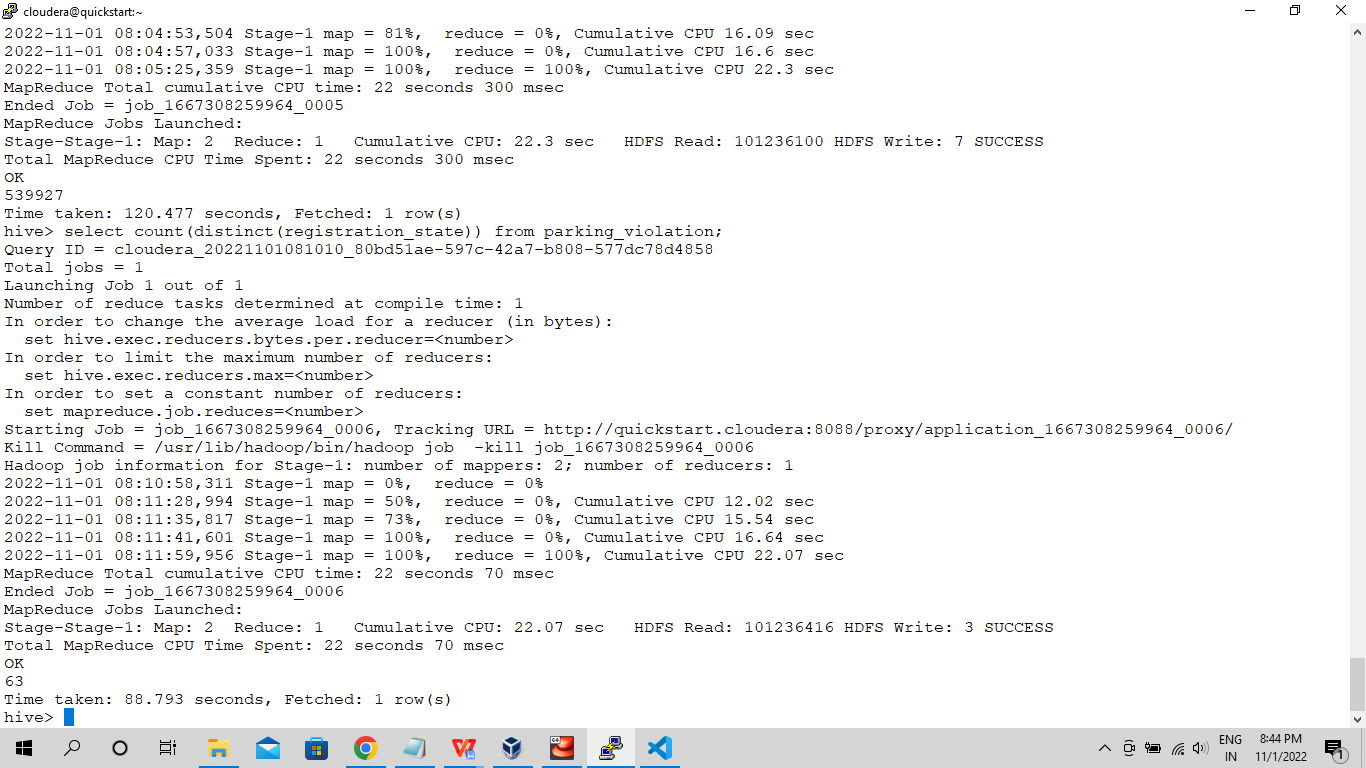
1. **Find the total number of tickets for the year.**

select count(summons\_number) from parking\_violation;



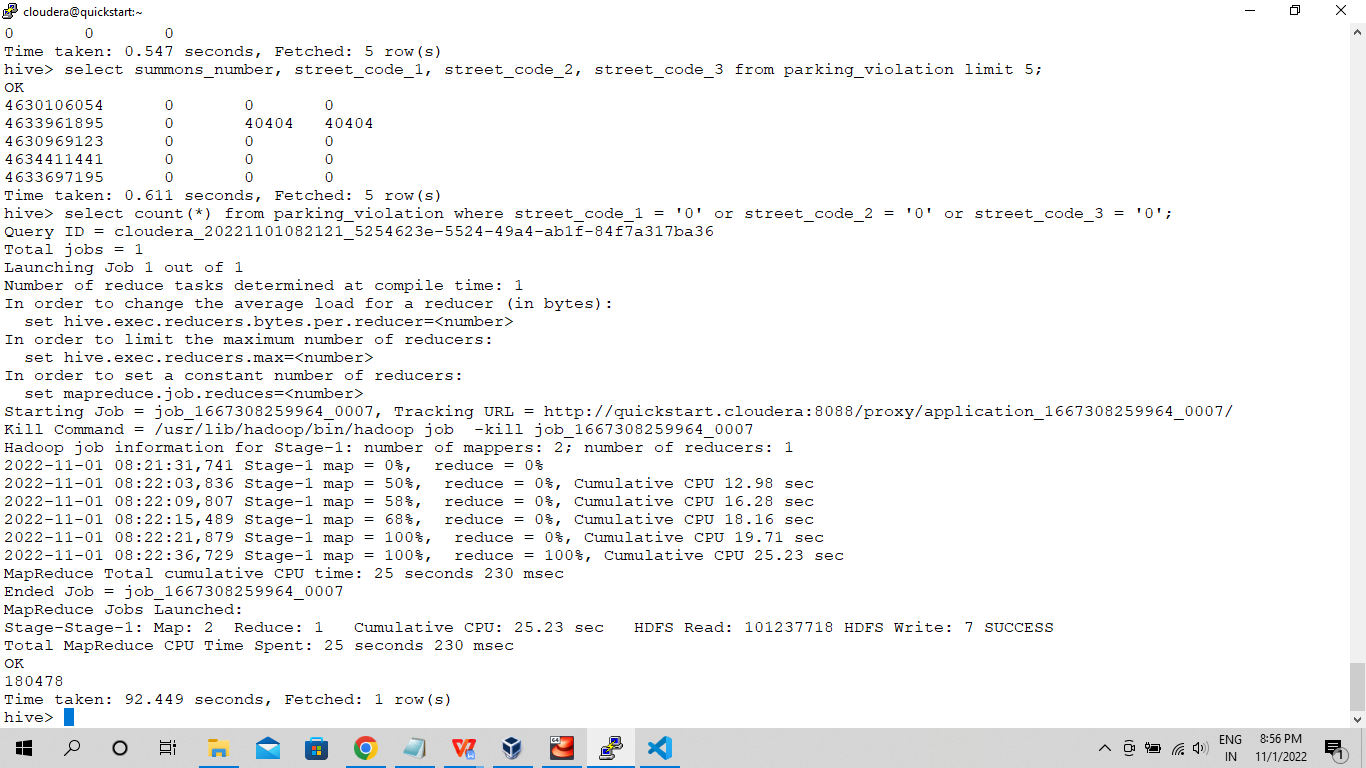
1. **Find out how many unique states the cars which got parking tickets came from.**

select count(distinct(registration\_state)) from parking\_violation;



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

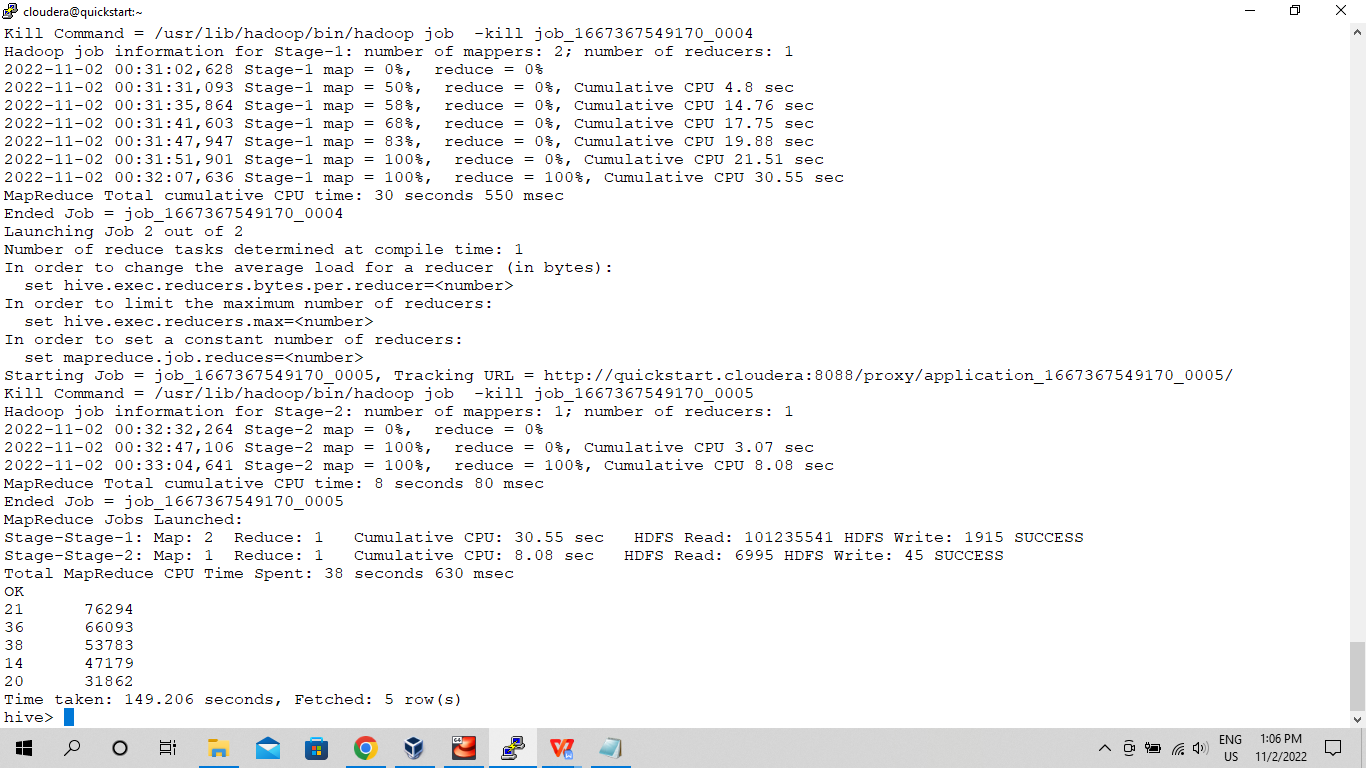
**select count(\*) from parking\_violation where street\_code\_1 = '0' or street\_code\_2 = '0' or street\_code\_3 = '0';**



**Part-II: Aggregation tasks**

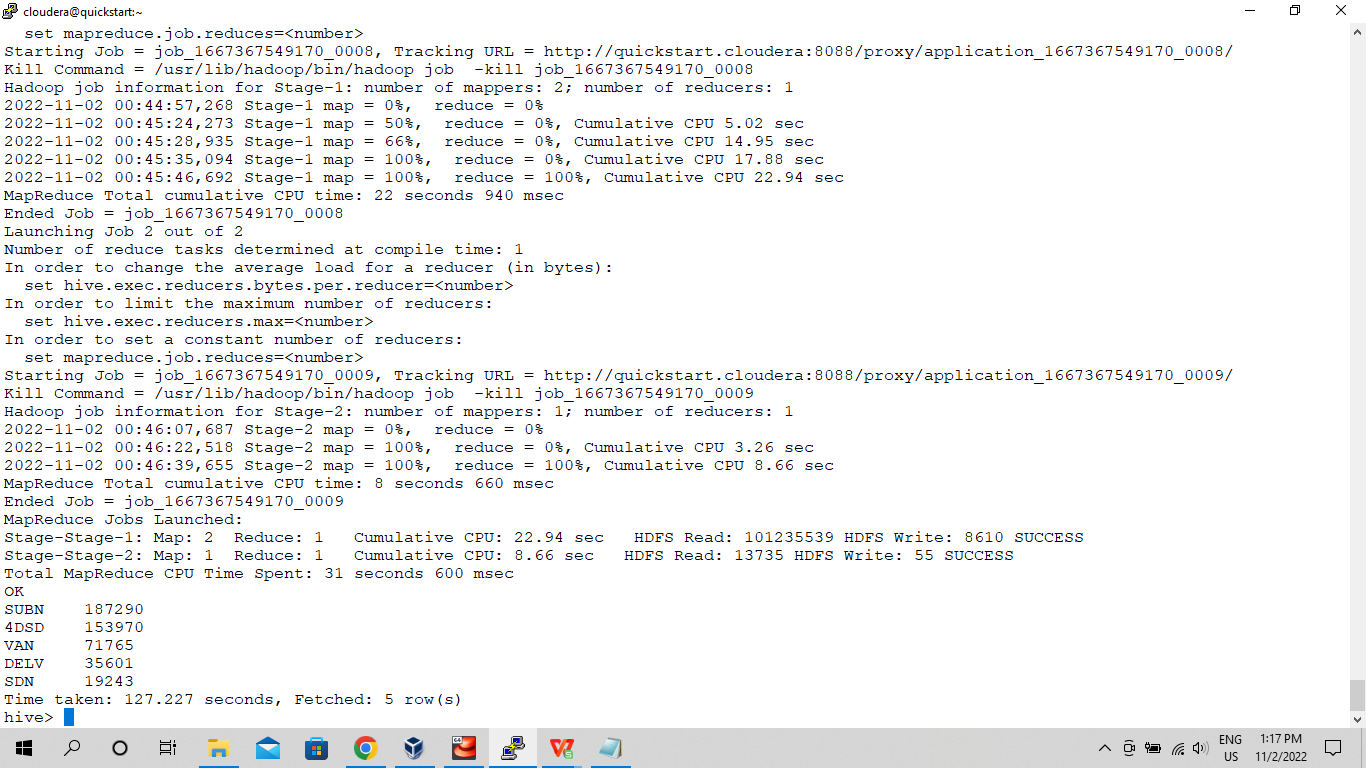
1. **How often does each violation code occur? (frequency of violation codes - find the top 5)**

select violation\_code, count(violation\_code) as frequency from parking\_violation group by violation\_code order by frequency desc limit 5;

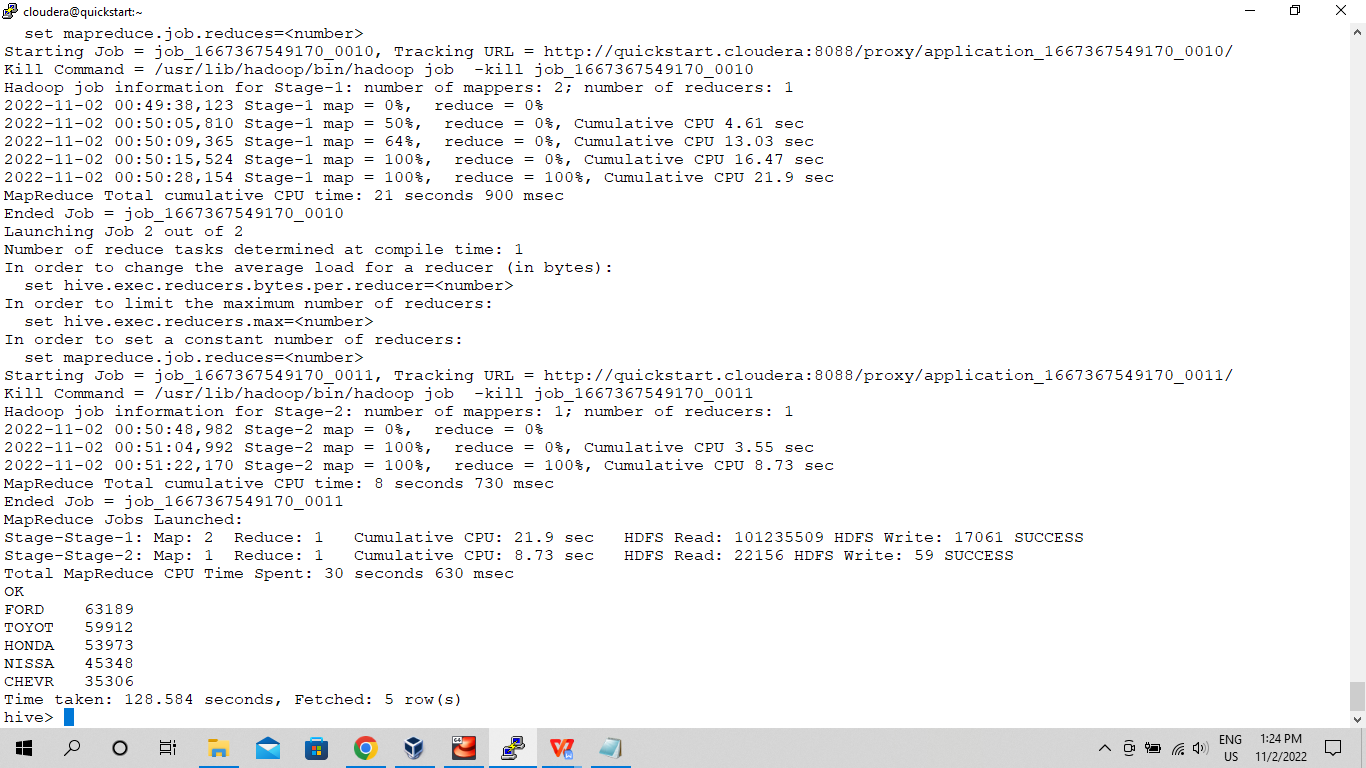


1. **How often does each vehicle body type get a parking ticket? How about the vehicle make? (find the top 5 for both)**

select vehicle\_body\_type, count(vehicle\_body\_type) as frequency from parking\_violation group by vehicle\_body\_type order by frequency desc limit 5;

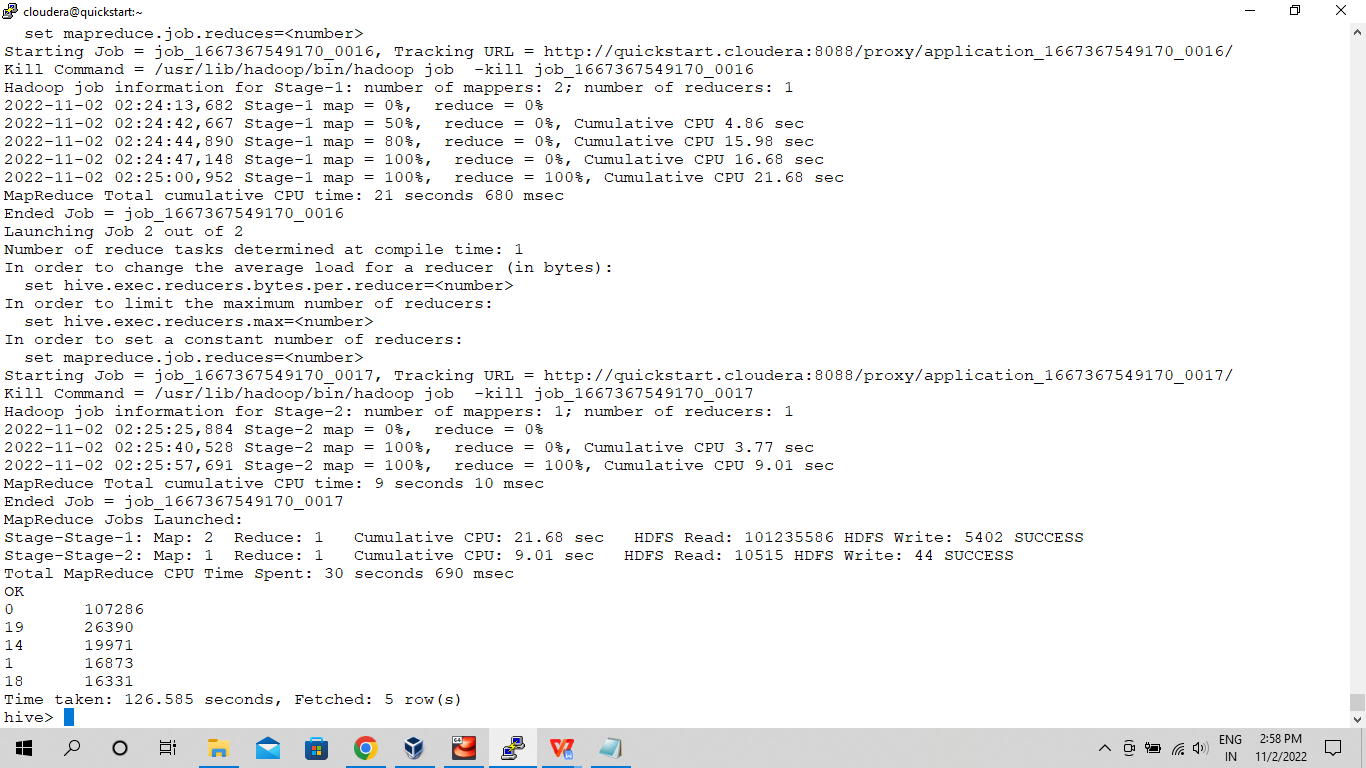


select vehicle\_make, count(vehicle\_make) as frequency from parking\_violation group by vehicle\_make order by frequency desc limit 5;



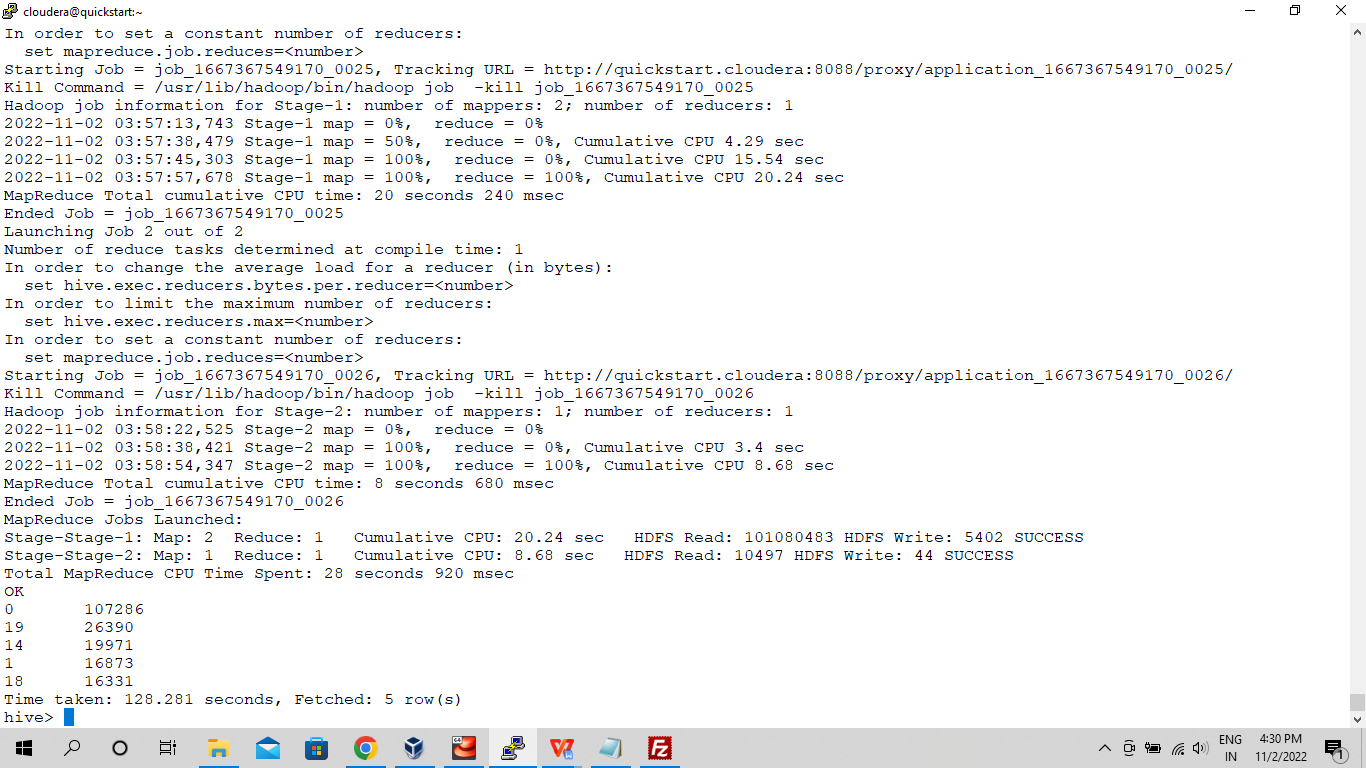
1. **A precinct is a police station that has a certain zone of the city under its command. Find the (5 highest) frequencies of:**
2. **Violating Precincts (this is the precinct of the zone where the violation occurred)**

select violation\_precinct, count(violation\_precinct) as frequency from parking\_violation group by violation\_precinct order by frequency desc limit 5;



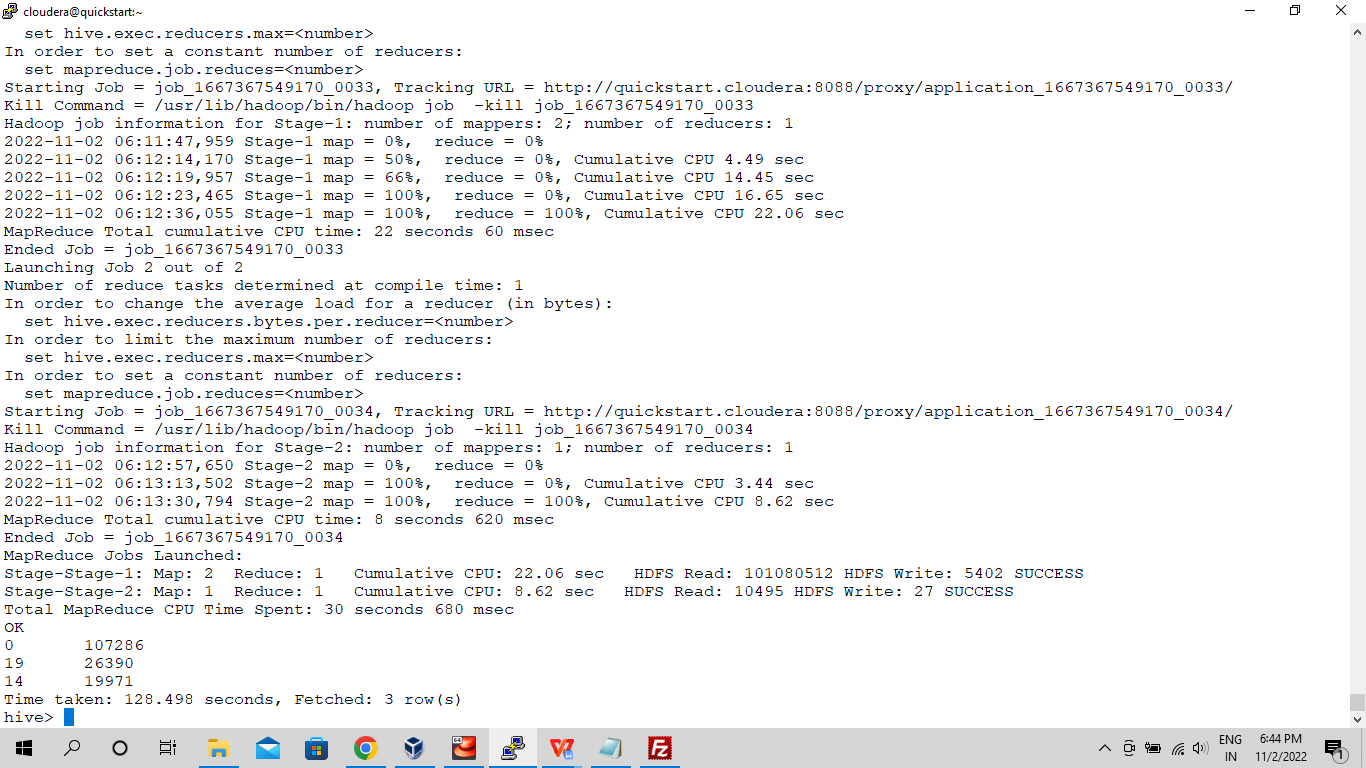
1. **Issuer Precincts (this is the precinct that issued the ticket)**

select issuer\_precinct, count(issuer\_precinct) as frequency from parking\_violation group by issuer\_precinct order by frequency desc limit 5;



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

select issuer\_precinct, count(violation\_code) as frequency from parking\_violation group by issuer\_precinct order by frequency desc limit 3;



1. **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.**
2. **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**

create view parking\_violations\_bins

partitioned on (Violation\_Code)

as

SELECT

summons\_number,

violation\_time,

case

when substring(Violation\_Time,1,2) in ('00','01','02','03') and substring(violation\_time,-1)='A' then 1

when substring(Violation\_Time,1,2) in ('04','05','06','07') and substring(violation\_time,-1)='A' then 2

when substring(Violation\_Time,1,2) in ('08','09','10','11','12') and substring(violation\_time,-1)='A' then 3

when substring(Violation\_Time,1,2) in ('00','01','02','03') and substring(violation\_time,-1)='P' then 4

when substring(Violation\_Time,1,2) in ('04','05','06','07') and substring(violation\_time,-1)='P' then 5

when substring(Violation\_Time,1,2) in ('08','09','10','11','12') and substring(violation\_time,-1)='P'then 6

else null

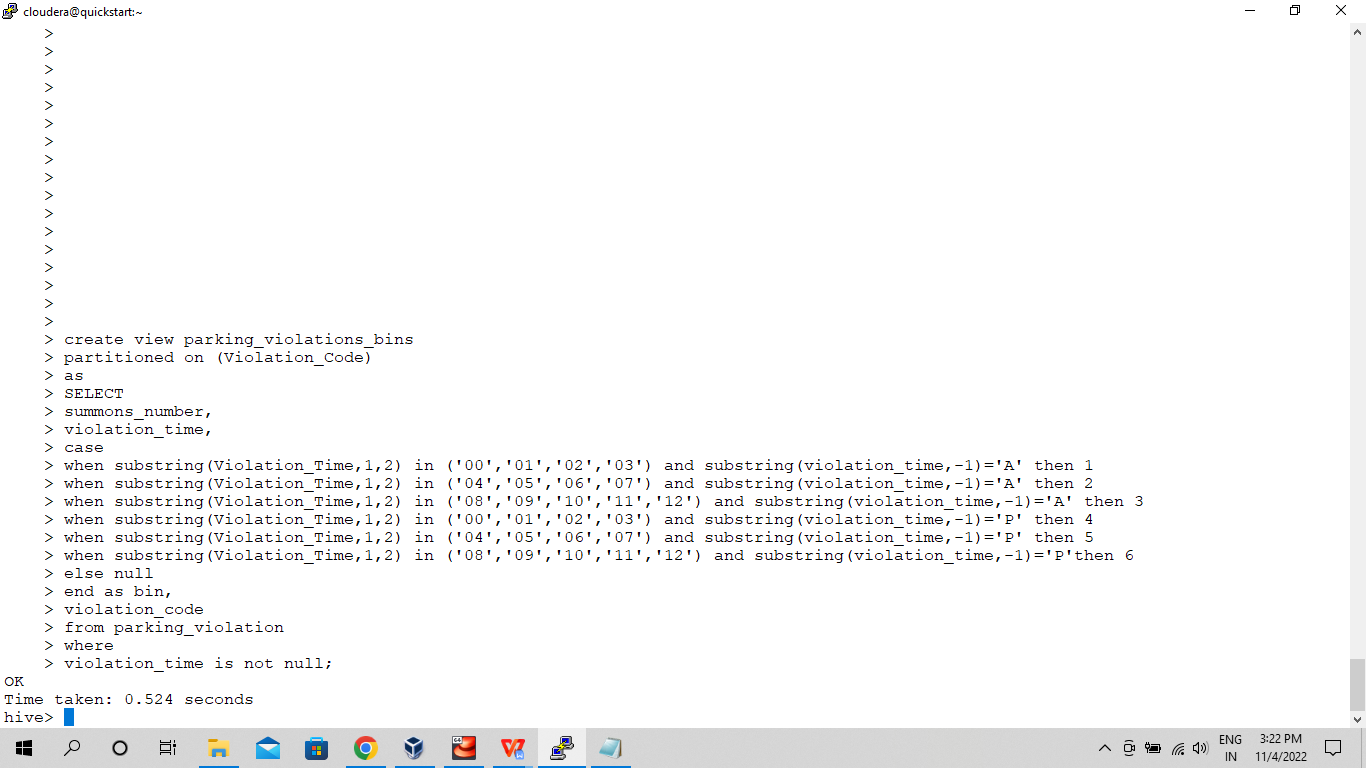
end as bin,

violation\_code

from parking\_violation

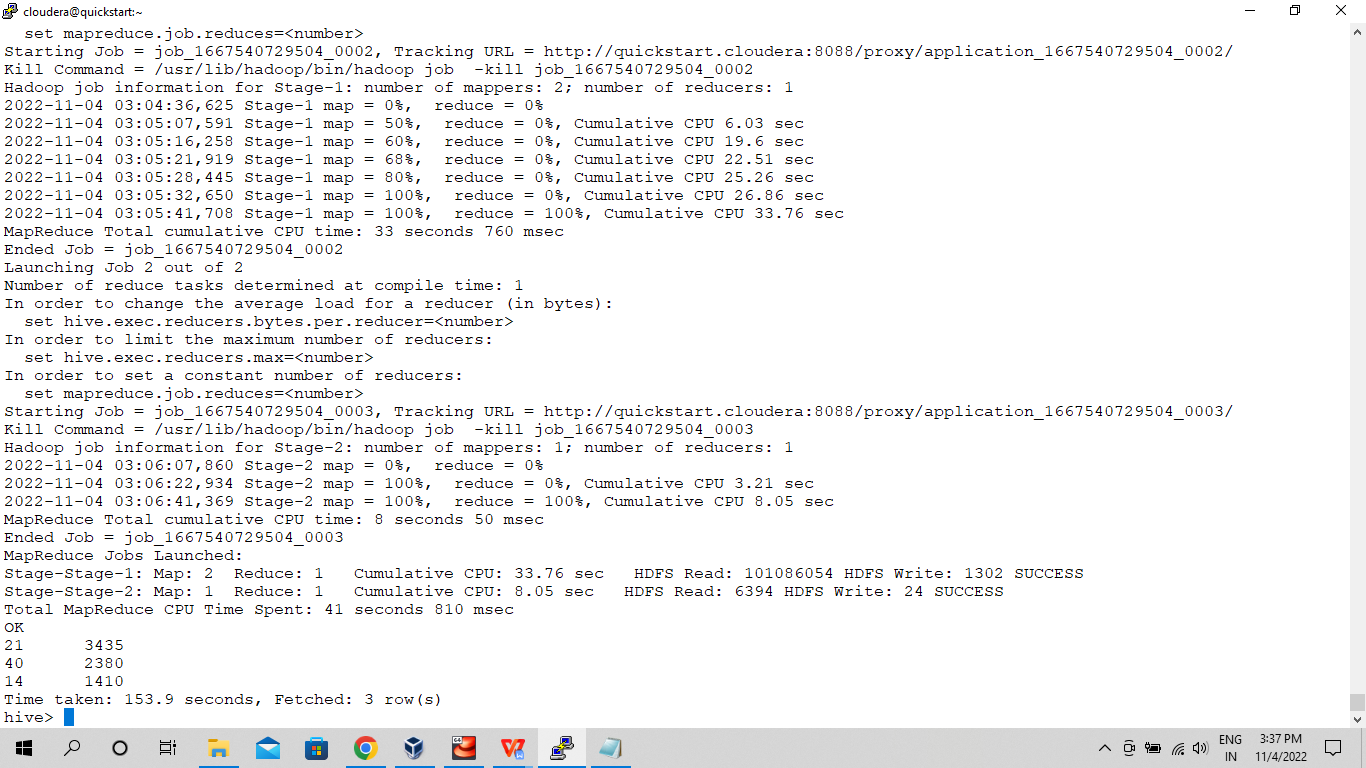
where

violation\_time is not null;



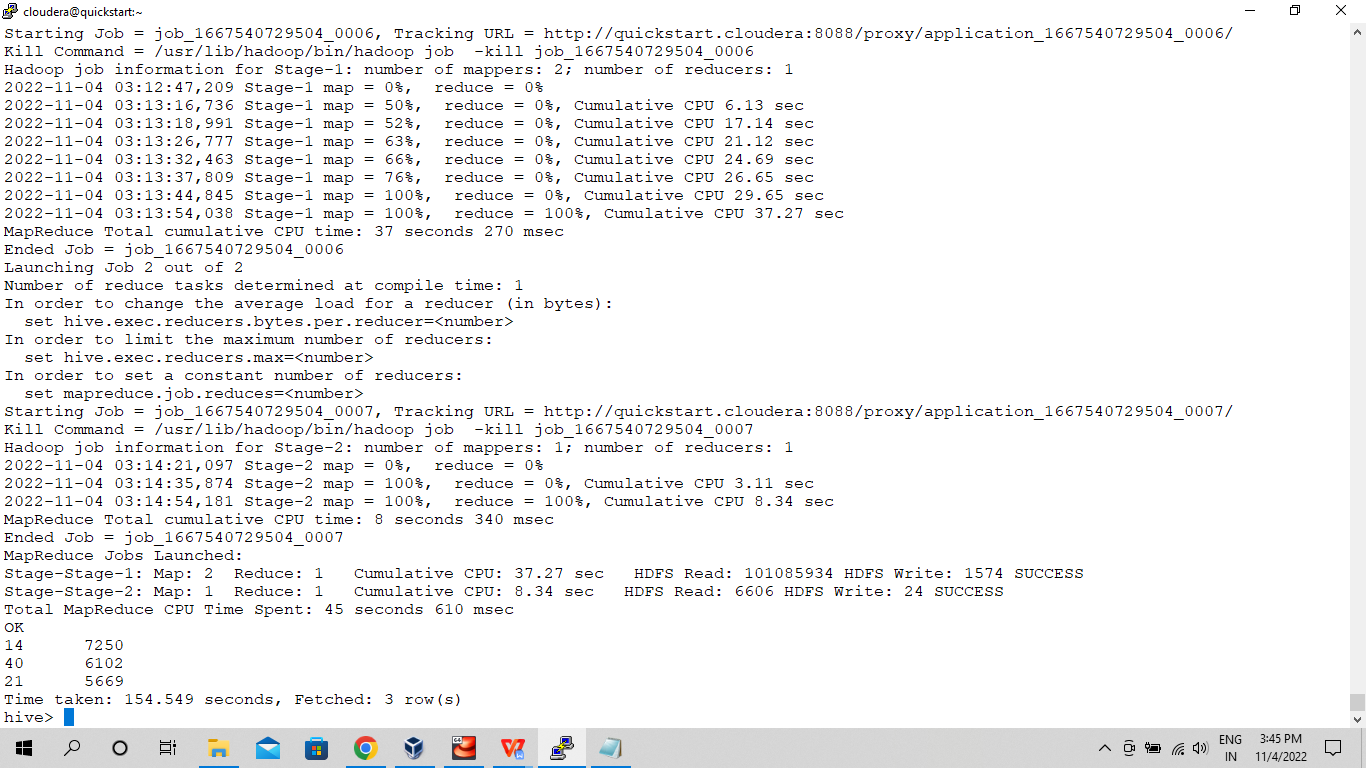
**Bin 1 -**

select violation\_code, count(\*) as occurence from parking\_violations\_bins where bin = '1' group by violation\_code order by occurence desc limit 3;



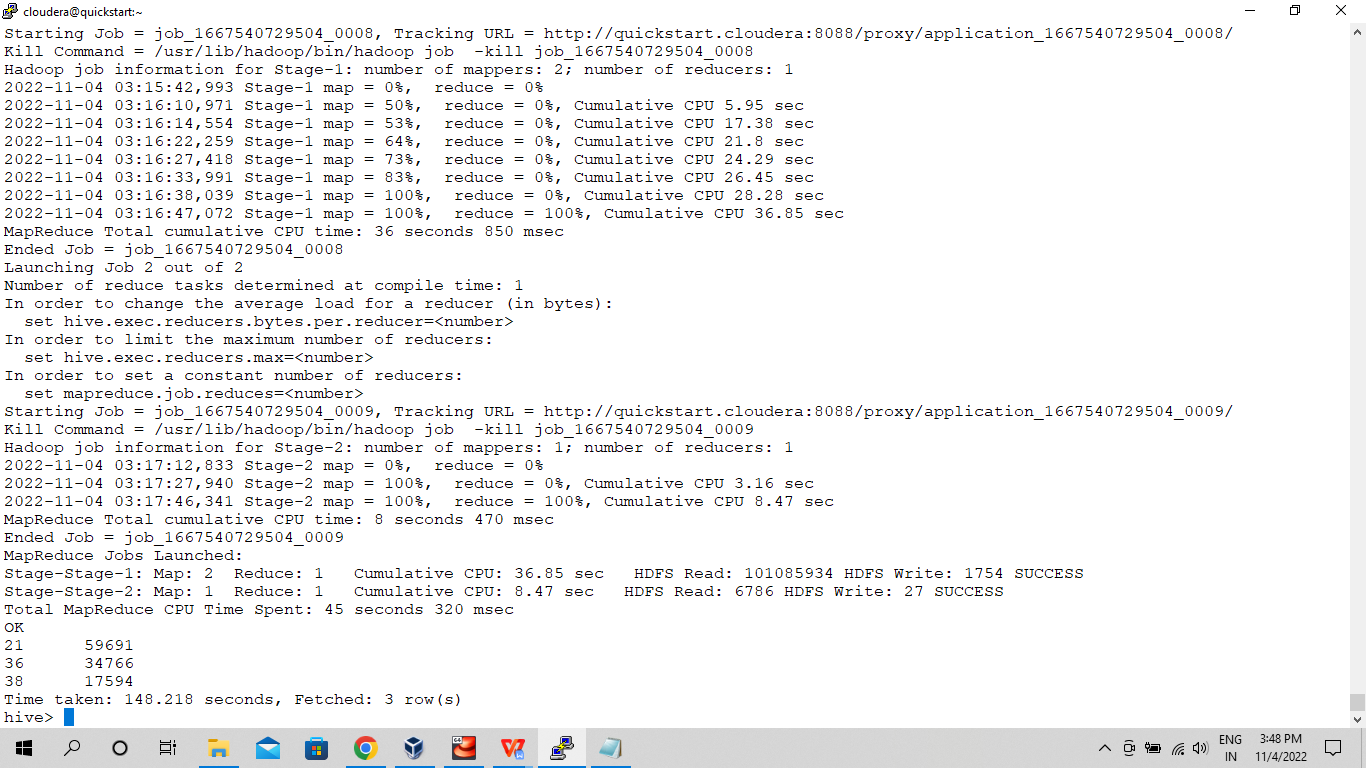
**Bin 2 -**

select violation\_code, count(\*) as occurence from parking\_violations\_bins where bin = '2' group by violation\_code order by occurence desc limit 3;



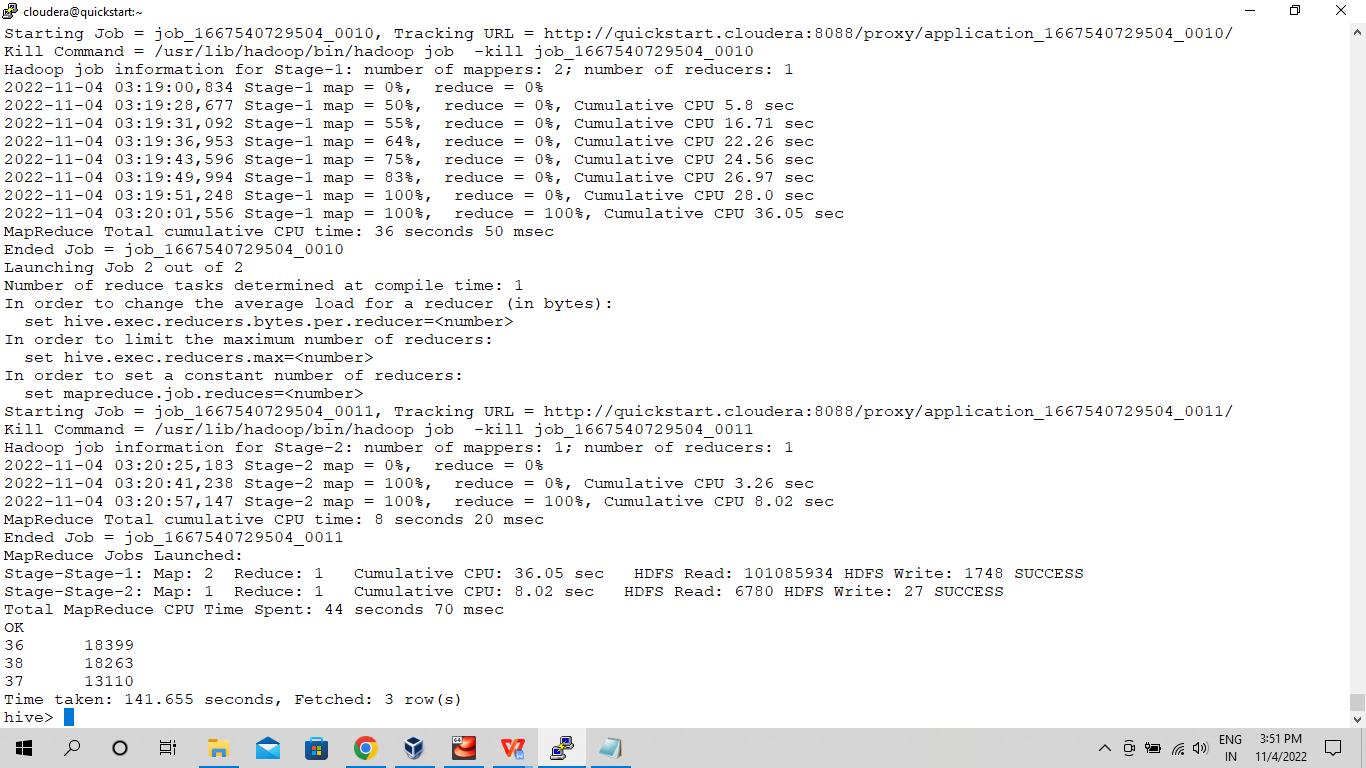
**Bin 3 -**

select violation\_code, count(\*) as occurence from parking\_violations\_bins where bin = '3' group by violation\_code order by occurence desc limit 3;



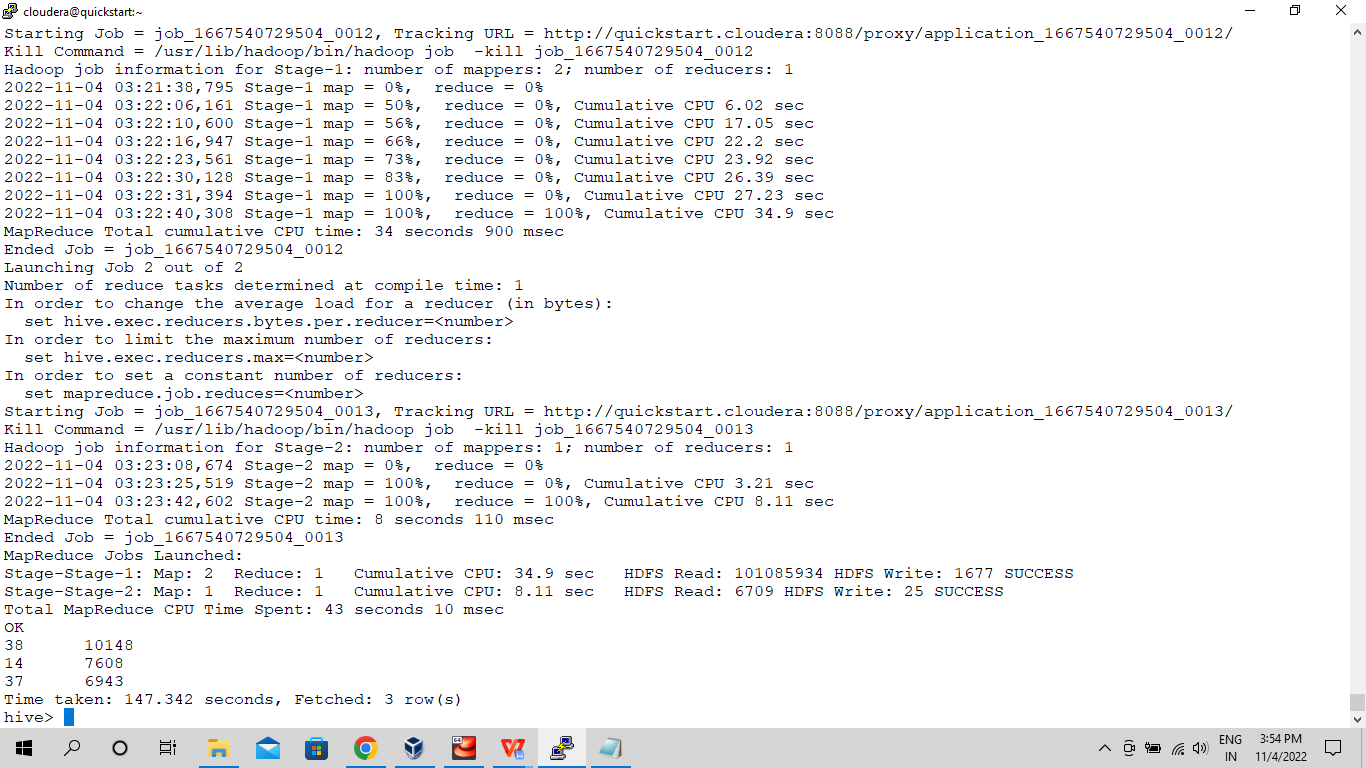
**Bin 4 -**

select violation\_code, count(\*) as occurence from parking\_violations\_bins where bin = '4' group by violation\_code order by occurence desc limit 3;



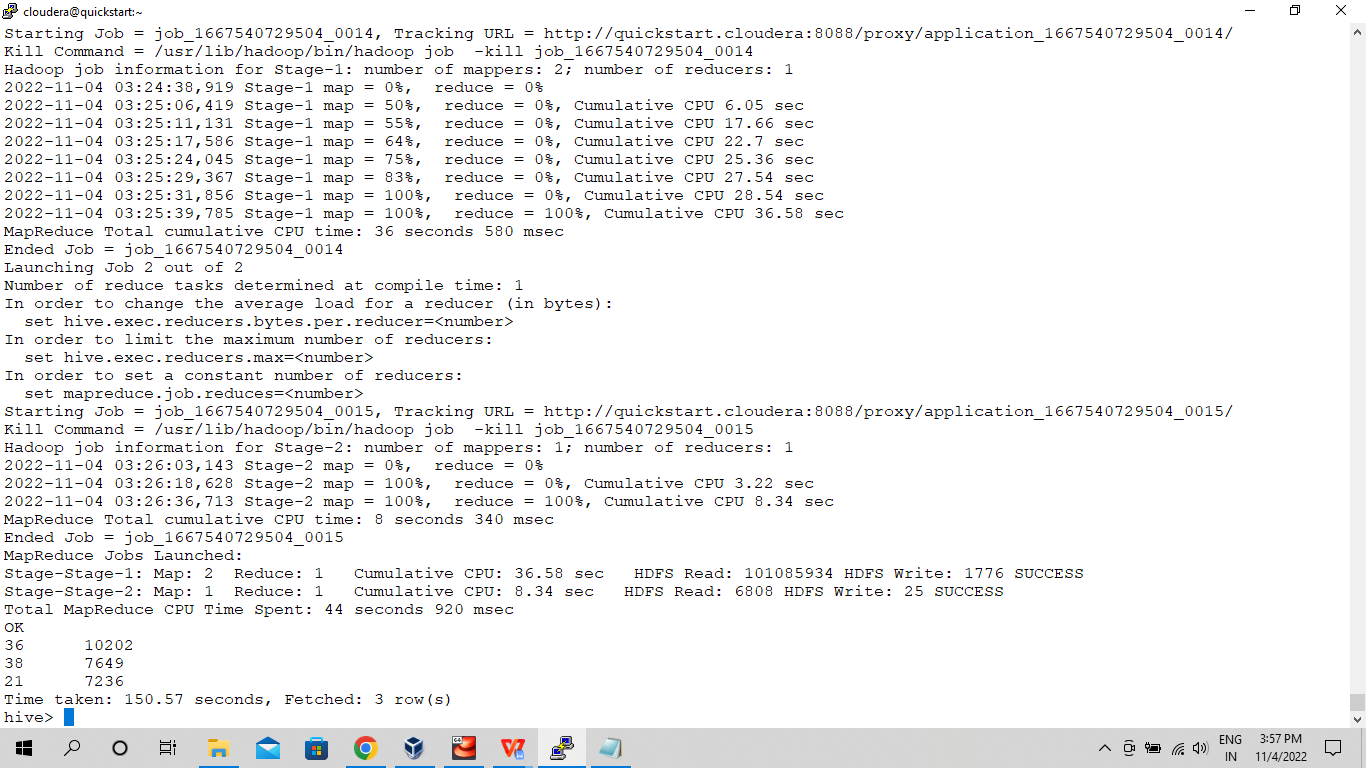
**Bin 5 -**

select violation\_code, count(\*) as occurence from parking\_violations\_bins where bin = '5' group by violation\_code order by occurence desc limit 3;



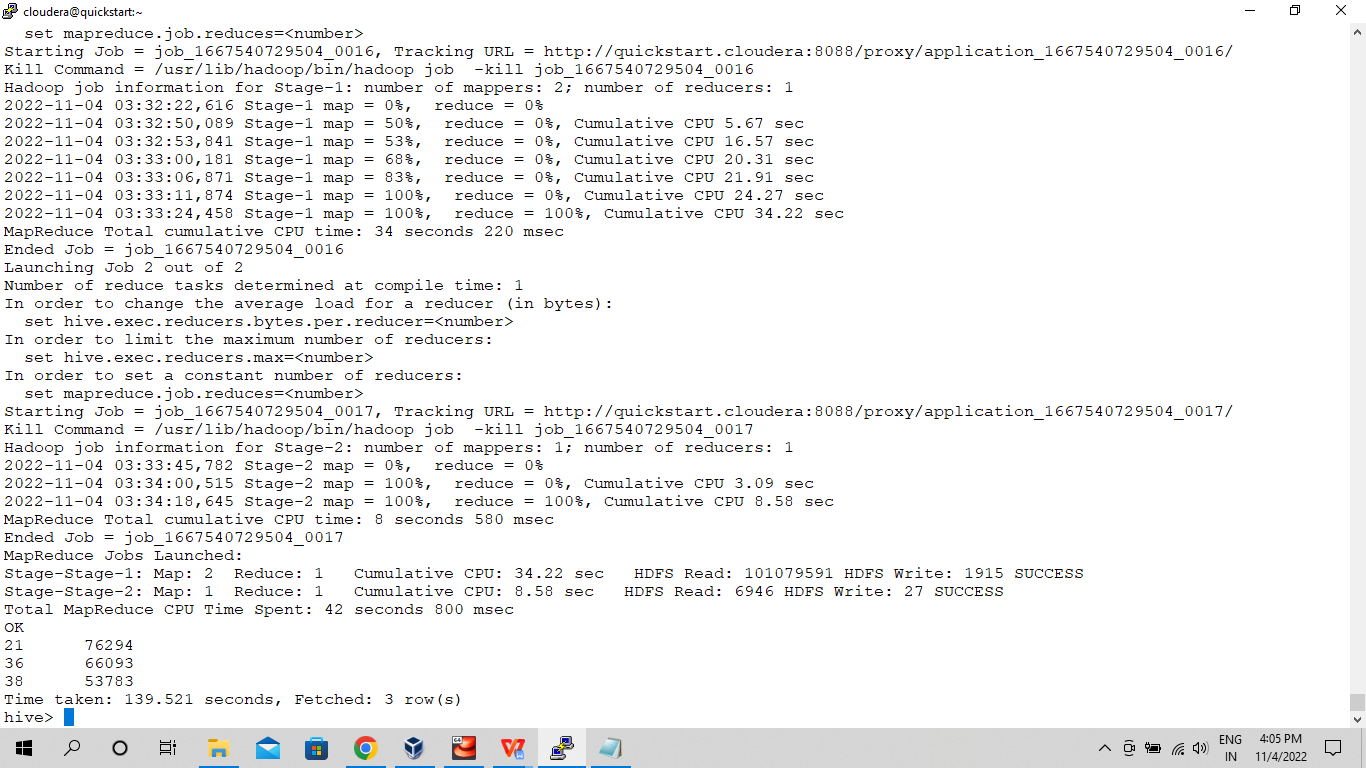
**Bin 6 -**

select violation\_code, count(\*) as occurence from parking\_violations\_bins where bin = '6' group by violation\_code order by occurence desc limit 3;

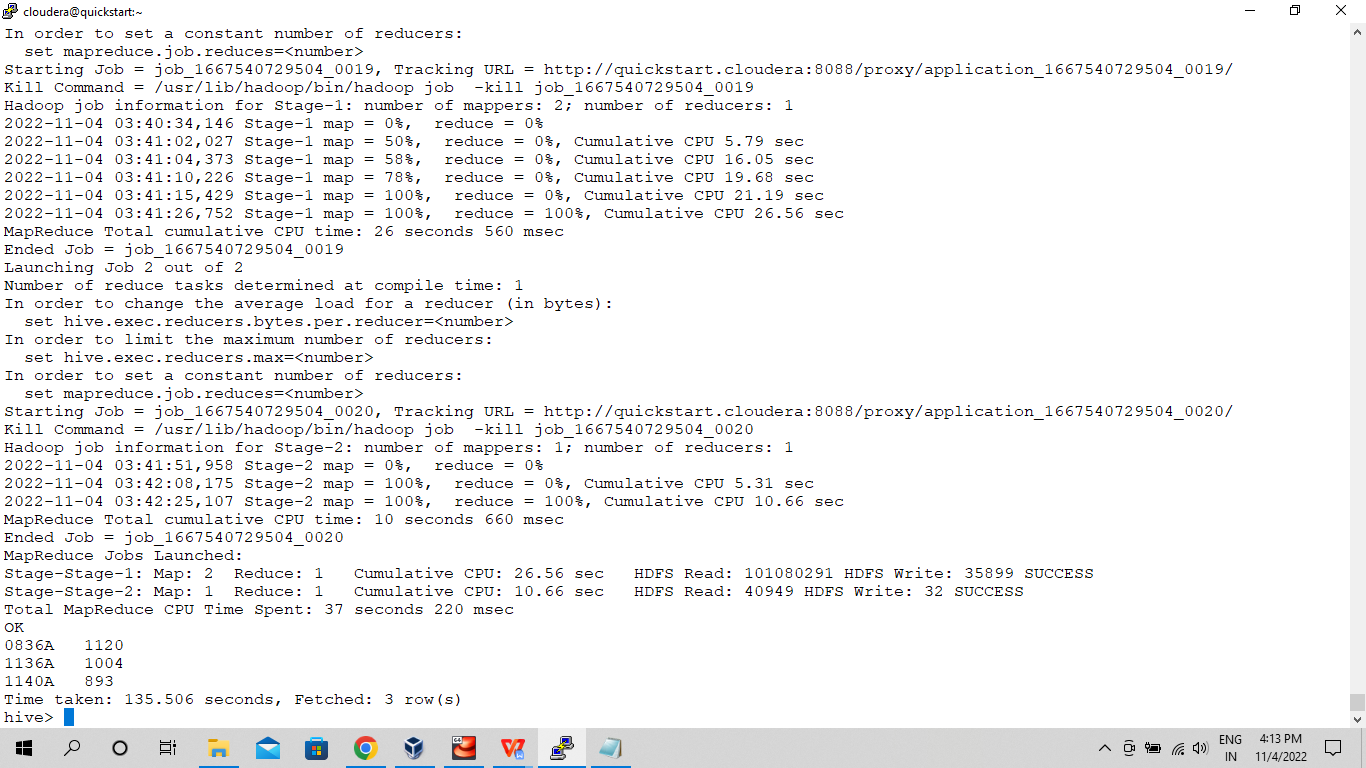


1. **Now, try another direction. For the 3 most commonly occurring violation codes, find the most common times of day (in terms of the bins from the previous part)**

select violation\_code, count(\*) as occurence from parking\_violations\_bins group by violation\_code order by occurence desc limit 3;



select violation\_time, count(\*) as frequency from parking\_violations\_bins where violation\_code in ('21','36','38') group by violation\_time order by frequency desc limit 3;



1. **Let’s try and find some seasonality in this data**
2. **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))**

create view parking\_violations\_seasons

partitioned on (Violation\_Code)

as

SELECT

summons\_number,

case

when month(issue\_date) between '03' and '05' then 'spring'

when month(issue\_date) between '06' and '08' then 'summer'

when month(issue\_date) between '09' and '11' then 'fall'

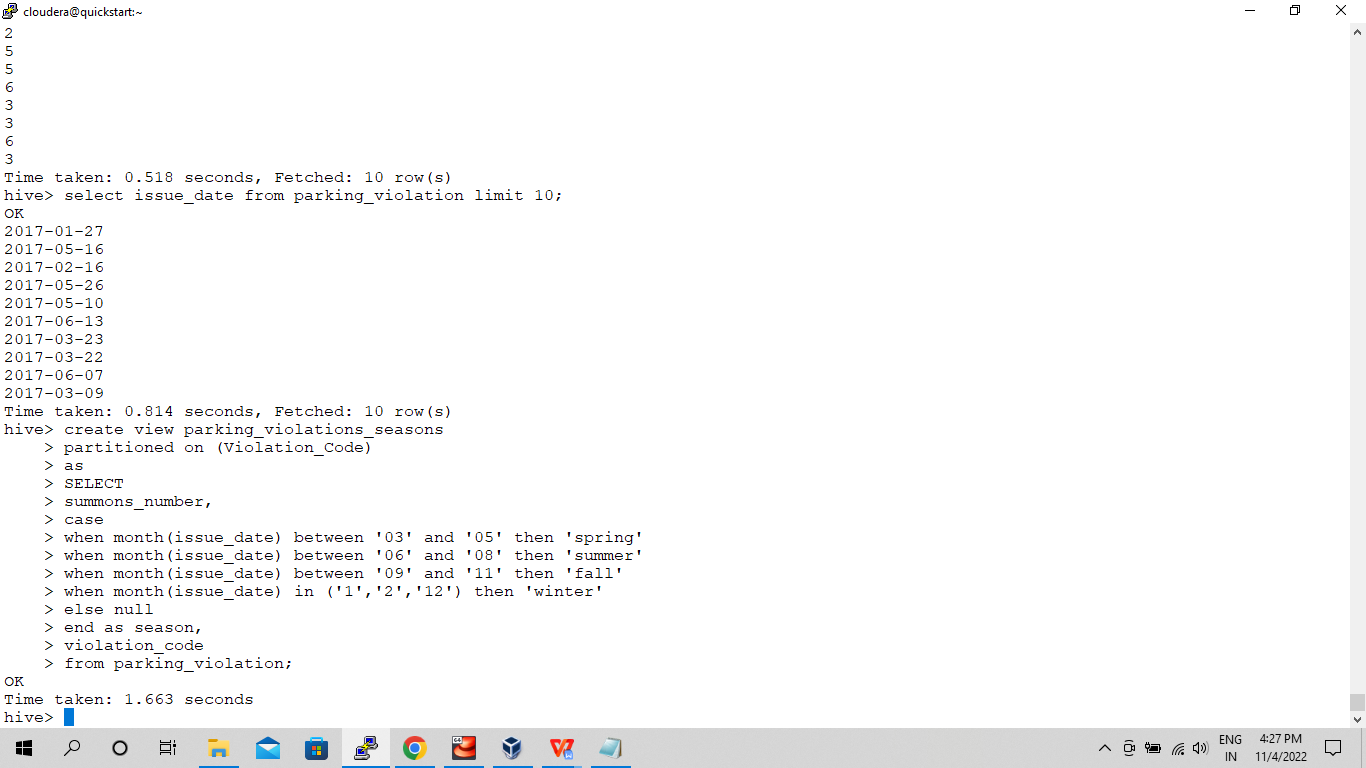
when month(issue\_date) in ('1','2','12') then 'winter'

else null

end as season,

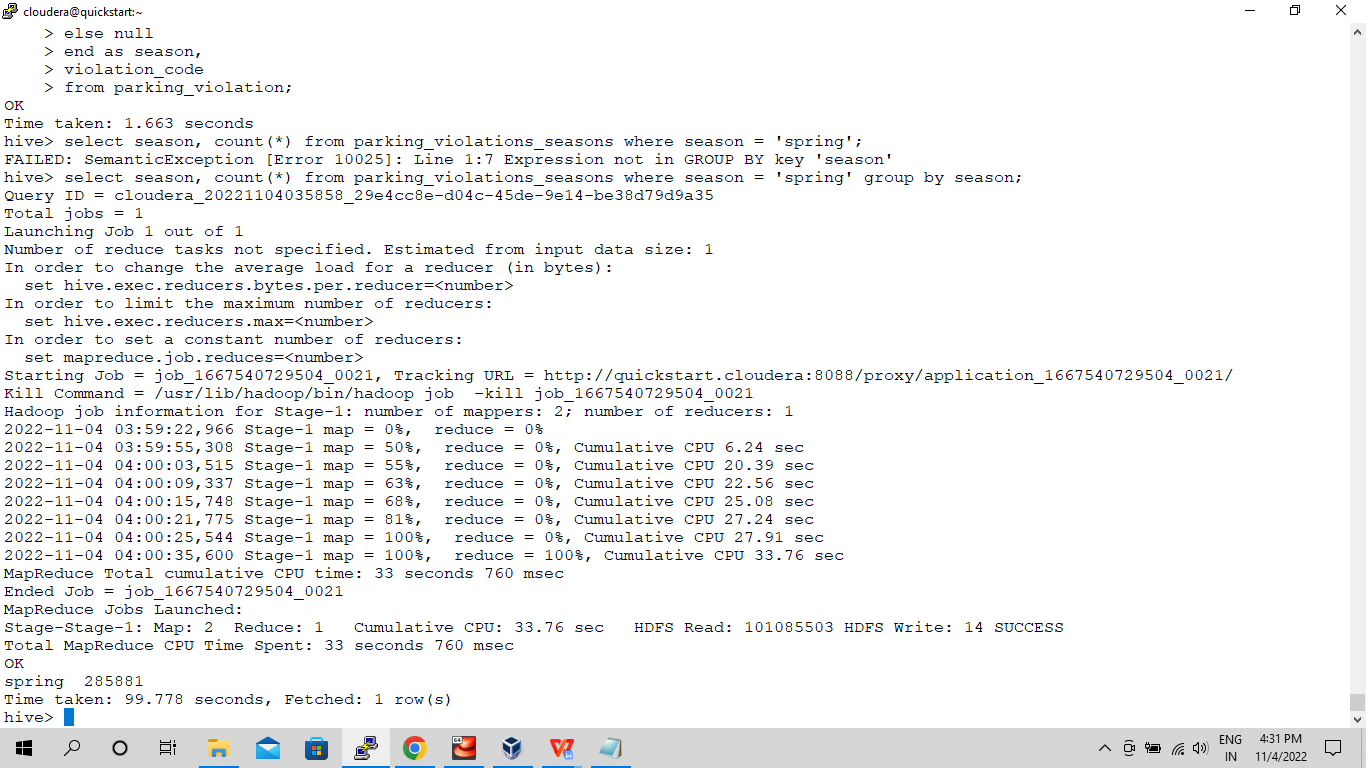
violation\_code

from parking\_violation;

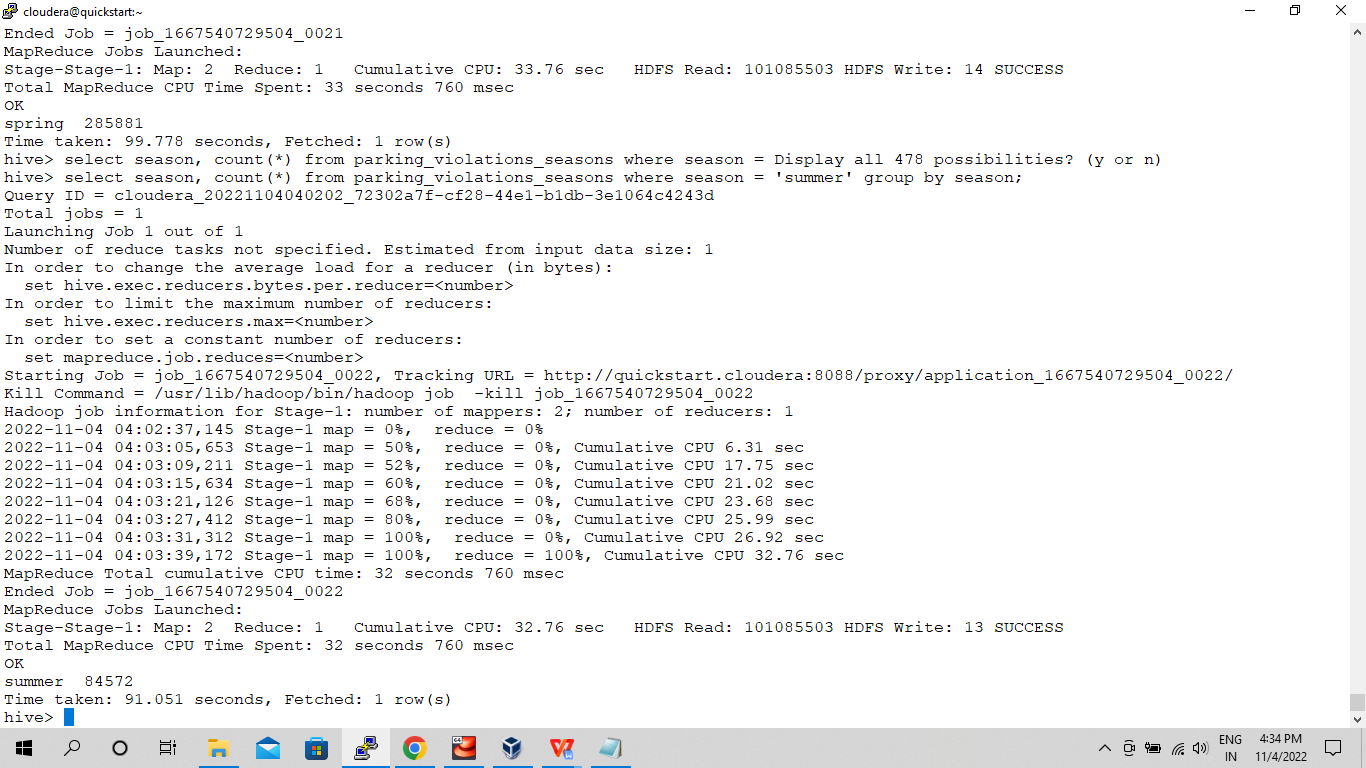


**Spring -**

select season, count(\*) from parking\_violations\_seasons where season = 'spring' group by season;

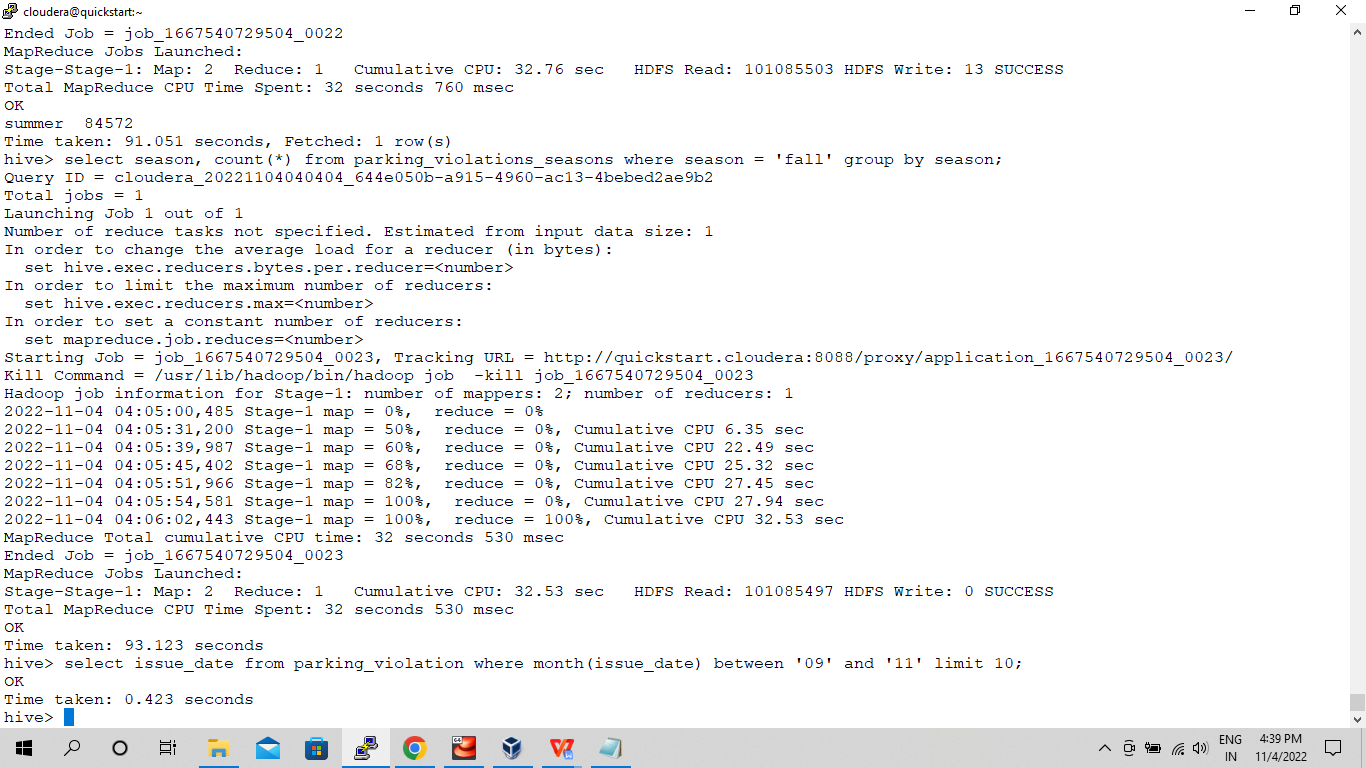


**Summer -**

select season, count(\*) from parking\_violations\_seasons where season = 'summer' group by season; 

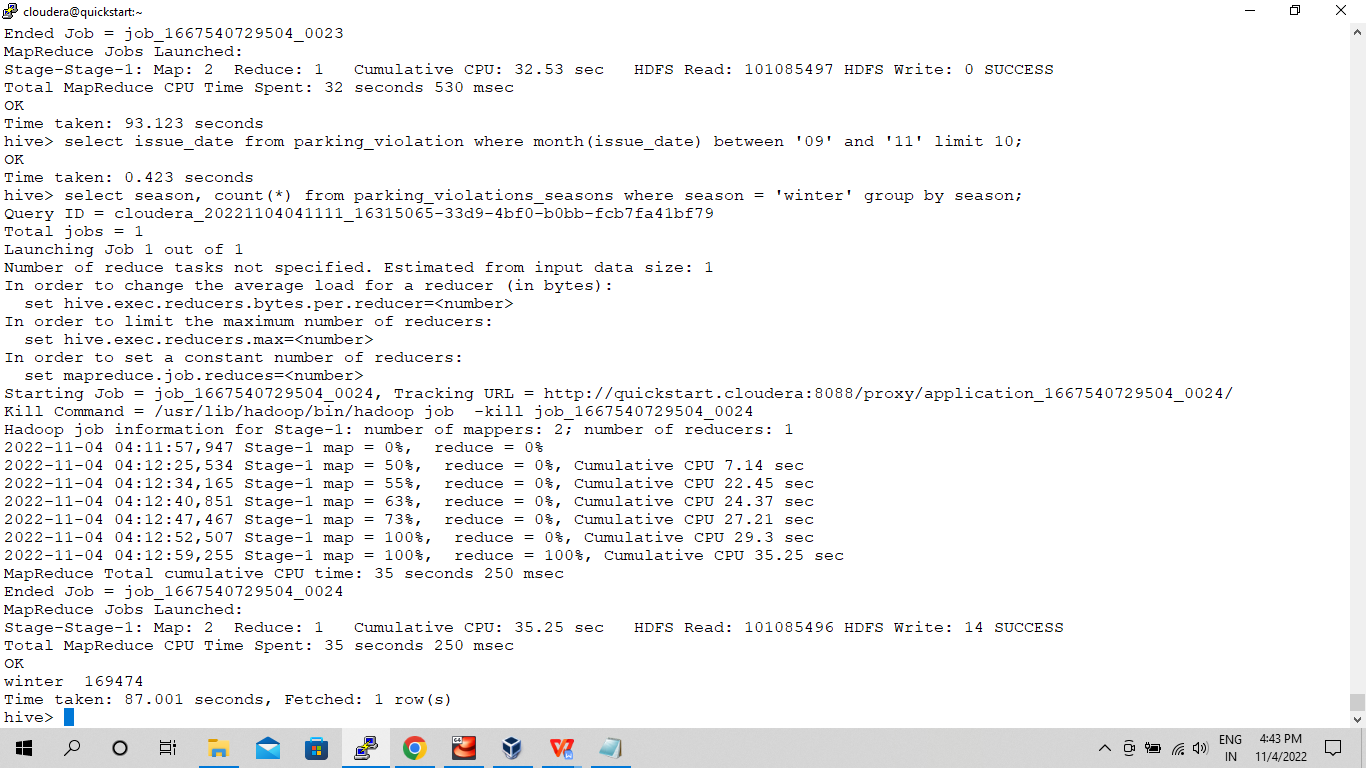
**Fall -**

select season, count(\*) from parking\_violations\_seasons where season = 'fall' group by season;



**Winter** -

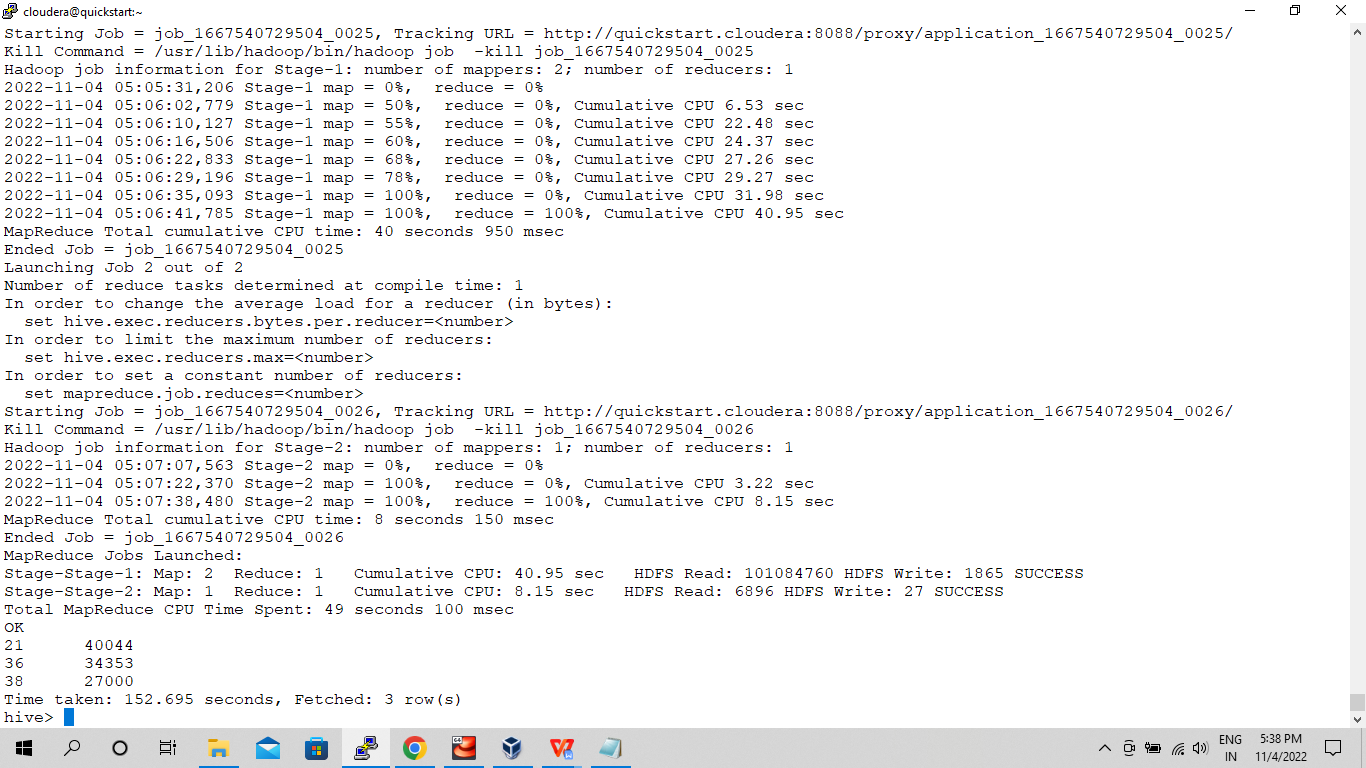
select season, count(\*) from parking\_violations\_seasons where season = 'winter' group by season;



1. **Then, find the 3 most common violations for each of these seasons.**

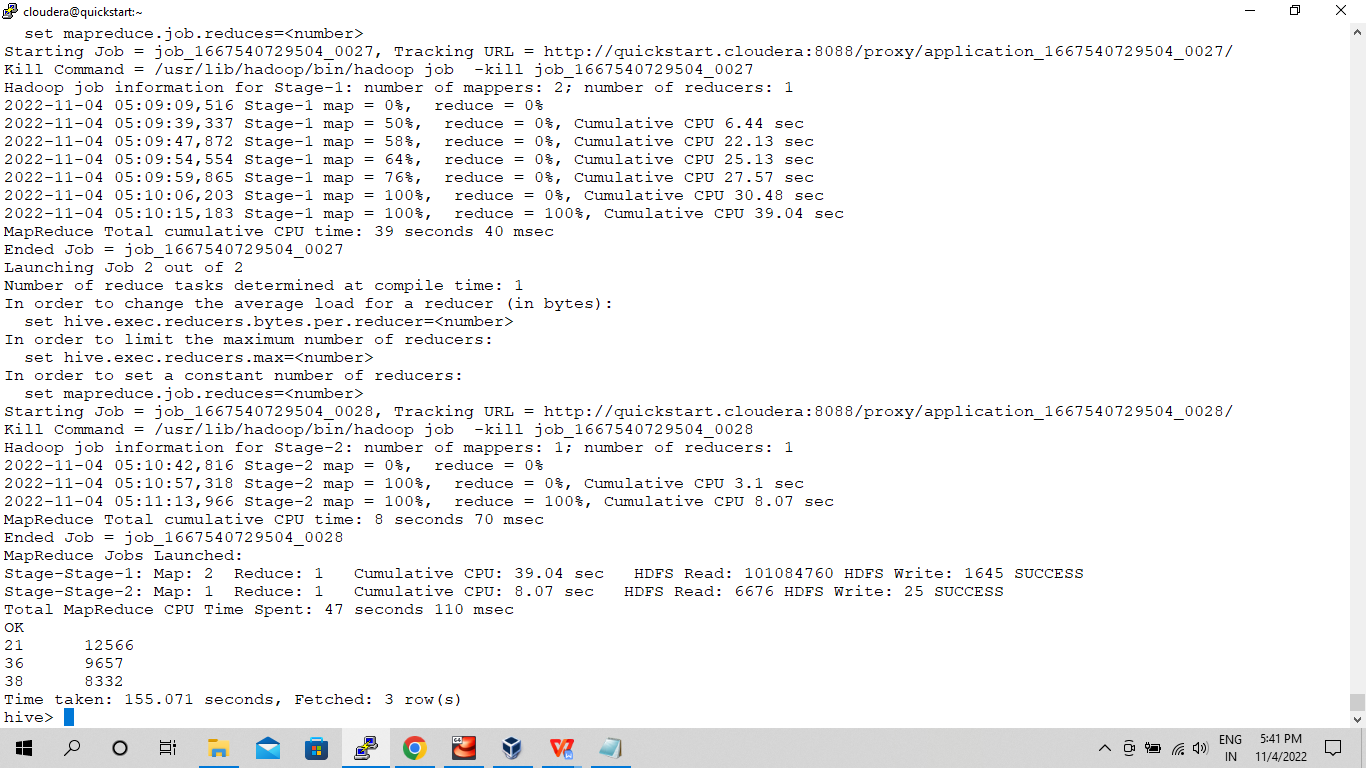
**Spring -**

select violation\_code, count(\*) as frequency from parking\_violations\_seasons where season = 'spring' group by violation\_code order by frequency desc limit 3;



**Summer -**

select violation\_code, count(\*) as frequency from parking\_violations\_seasons where season = 'summer' group by violation\_code order by frequency desc limit 3;



**Fall -**

select violation\_code, count(\*) as frequency from parking\_violations\_seasons where season = 'fall' group by violation\_code order by frequency desc limit 3;



**Winter -**

select violation\_code, count(\*) as frequency from parking\_violations\_seasons where season = 'winter' group by violation\_code order by frequency desc limit 3;

