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

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

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

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

**Hive> create table parking\_violations\_issued**

**(**

**Summons\_Number int,**

**Plate\_ID string,**

**Registration\_State string,**

**Plate\_Type string,**

**Issue\_Date date,**

**Violation\_Code int,**

**Vehicle\_Body\_Type string,**

**Vehicle\_Make string,**

**Issuing\_Agency string,**

**Street\_Code1 int,**

**Street\_Code2 int,**

**Street\_Code3 int,**

**Vehicle\_Expiration Date,**

**Violation\_Location int,**

**Violation\_Precinct int,**

**Issuer\_Precinct int,**

**Issuer\_Code int,**

**Issuer\_Command string,**

**Issuer\_Squad string,**

**Violation\_Time int,**

**Time\_First\_Observed string,**

**Violation\_County string,**

**Violation\_In\_Front\_Of\_Or\_Opposite string,**

**House\_Number string,**

**Street\_Name string,**

**Intersecting\_Street string,**

**Date\_First\_Observed int,**

**Law\_Section int,**

**Sub\_Division string,**

**Violation\_Legal\_Code string,**

**Days\_Parking\_In\_Effect string,**

**From\_Hours\_In\_Effect string,**

**To\_Hours\_In\_Effect string,**

**Vehicle\_Color string,**

**Unregistered\_Vehicle int,**

**Vehicle\_Year string,**

**Meter\_Number string,**

**Feet\_From\_Curb int,**

**Violation\_Post\_Code string,**

**Violation\_Description string,**

**No\_Standing\_or\_Stopping\_Violation string,**

**Hydrant\_Violation string,**

**Double\_Parking\_Violation string)**

**row format delimited**

**fields terminated by ','**

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

**alter table parking\_violations\_issued change Summons\_Number Summons\_Number bigint;**

**alter table parking\_violations\_issued change Violation\_Time Violation\_Time string;**

**load data local inpath'/home/cloudera/ajay/Challenge/mini\_project\_2/Parking\_Violations\_Issued\_-\_Fiscal\_Year\_2017.csv' into table parking\_violations\_issued;**

**select \* from parking\_violations\_issued limit 20;**

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

for year 2017 only

**Hive> create table parking\_violations\_issued\_2017**

**(**

**Summons\_Number bigint,Plate\_ID string,Registration\_State string,Plate\_Type string,Issue\_Date date,**

**Violation\_Code int,Vehicle\_Body\_Type string,Vehicle\_Make string,Issuing\_Agency string,**

**Street\_Code1 int,Street\_Code2 int,Street\_Code3 int,Vehicle\_Expiration Date,Violation\_Location int,**

**Violation\_Precinct int,Issuer\_Precinct int,Issuer\_Code int,Issuer\_Command string,Issuer\_Squad string,**

**Violation\_Time string,Time\_First\_Observed string,Violation\_In\_Front\_Of\_Or\_Opposite string,**

**House\_Number string,Street\_Name string,Intersecting\_Street string,Date\_First\_Observed int,**

**Law\_Section int,Sub\_Division string,Violation\_Legal\_Code string,Days\_Parking\_In\_Effect string,**

**From\_Hours\_In\_Effect string,To\_Hours\_In\_Effect string,Vehicle\_Color string,**

**Unregistered\_Vehicle int,Vehicle\_Year string,Meter\_Number string,Feet\_From\_Curb int,**

**Violation\_Post\_Code string,Violation\_Description string,No\_Standing\_or\_Stopping\_Violation string,**

**Hydrant\_Violation string,Double\_Parking\_Violation string)**

**COMMENT 'A bucketed sorted parking\_violations\_issued\_2017'**

**partitioned by (Violation\_County string)**

**CLUSTERED BY (Violation\_Code) sorted by (Violation\_Code) INTO 8 BUCKETS**

**row format delimited**

**fields terminated by ','**

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

**To load data into partition and bucket table we need to set few properties to enable bucketing and**

**Dynamic partition**

**hive>set hive.exec.dynamic.partition=true;**

**hive>set hive.exec.dynamic.partition.mode=nonstrict;**

**hive>set hive.enforce.bucketing = true;**

**Hive> insert into parking\_violations\_issued\_2017 partition(Violation\_County) select**

**Summons\_Number,Plate\_ID,Registration\_State,Plate\_Type,Issue\_Date,Violation\_Code,Vehicle\_Body\_Type,Vehicle\_Make,**

**Issuing\_Agency,Street\_Code1,Street\_Code2,Street\_Code3,Vehicle\_Expiration,Violation\_Location,Violation\_Precinct,**

**Issuer\_Precinct,Issuer\_Code,Issuer\_Command,Issuer\_Squad,Violation\_Time,Time\_First\_Observed,**

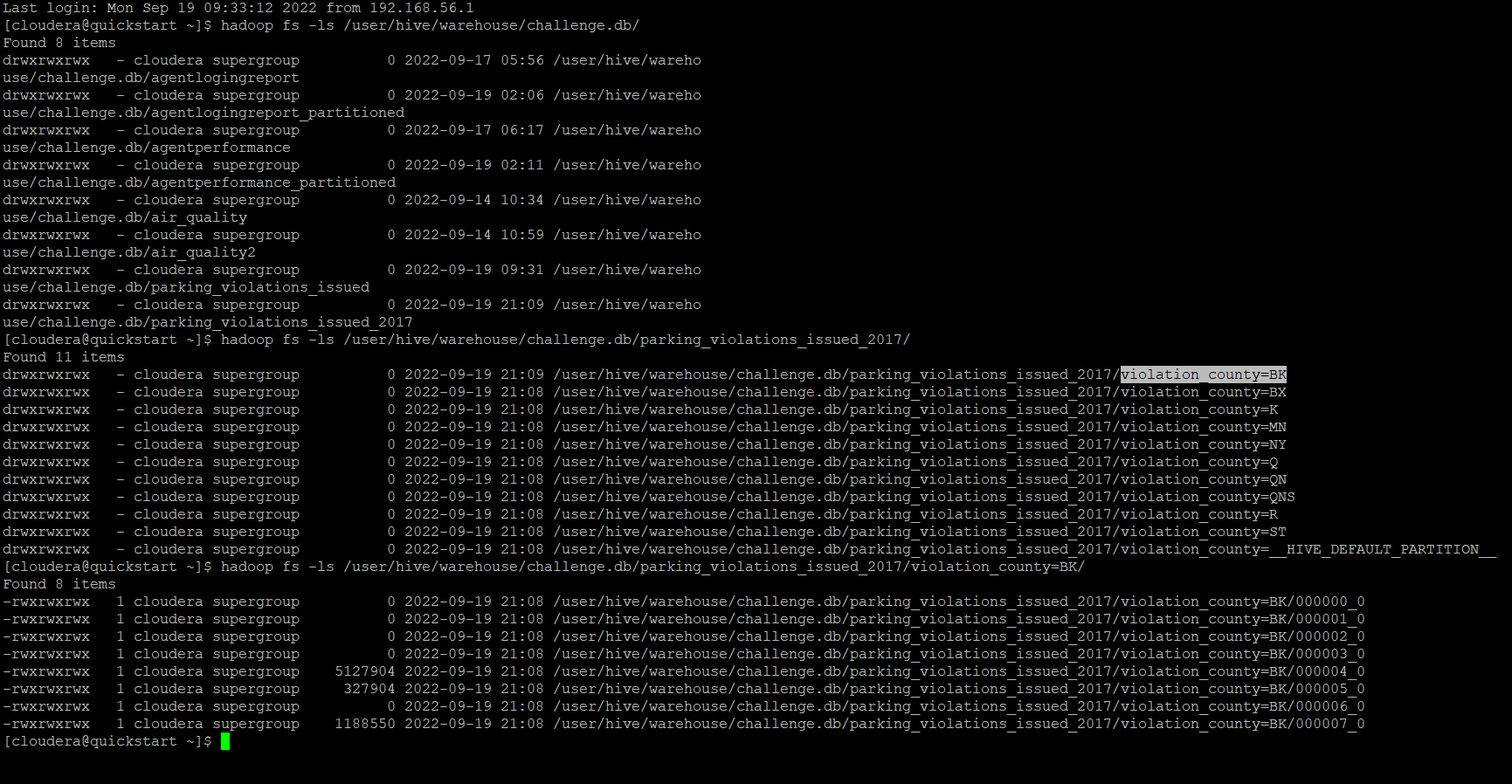
**Violation\_In\_Front\_Of\_Or\_Opposite,House\_Number,Street\_Name,Intersecting\_Street,Date\_First\_Observed,Law\_Section,**

**Sub\_Division,Violation\_Legal\_Code,Days\_Parking\_In\_Effect,From\_Hours\_In\_Effect,To\_Hours\_In\_Effect,Vehicle\_Color,**

**Unregistered\_Vehicle,Vehicle\_Year,Meter\_Number,Feet\_From\_Curb,Violation\_Post\_Code,Violation\_Description,**

**No\_Standing\_or\_Stopping\_Violation,Hydrant\_Violation,Double\_Parking\_Violation,Violation\_County from parking\_violations\_issued where**

**year(Issue\_Date) = '2017';**

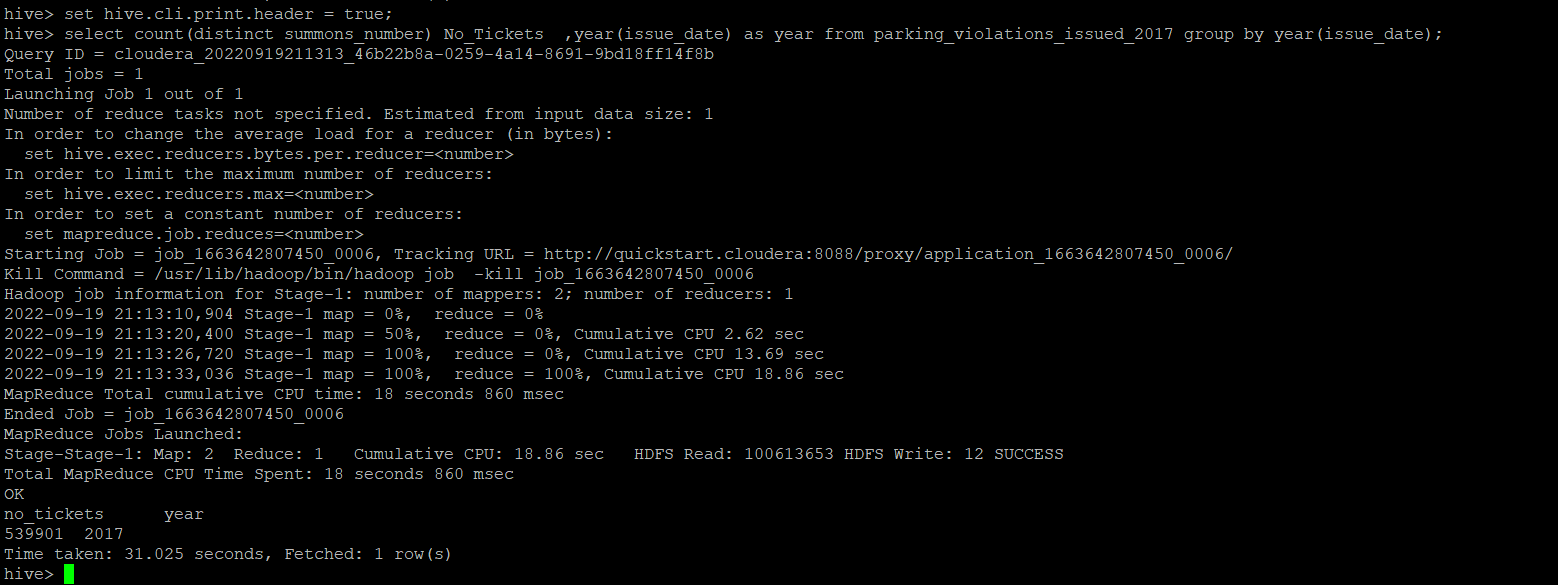


The analysis can be divided into two parts:

Part-I: Examine the data

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

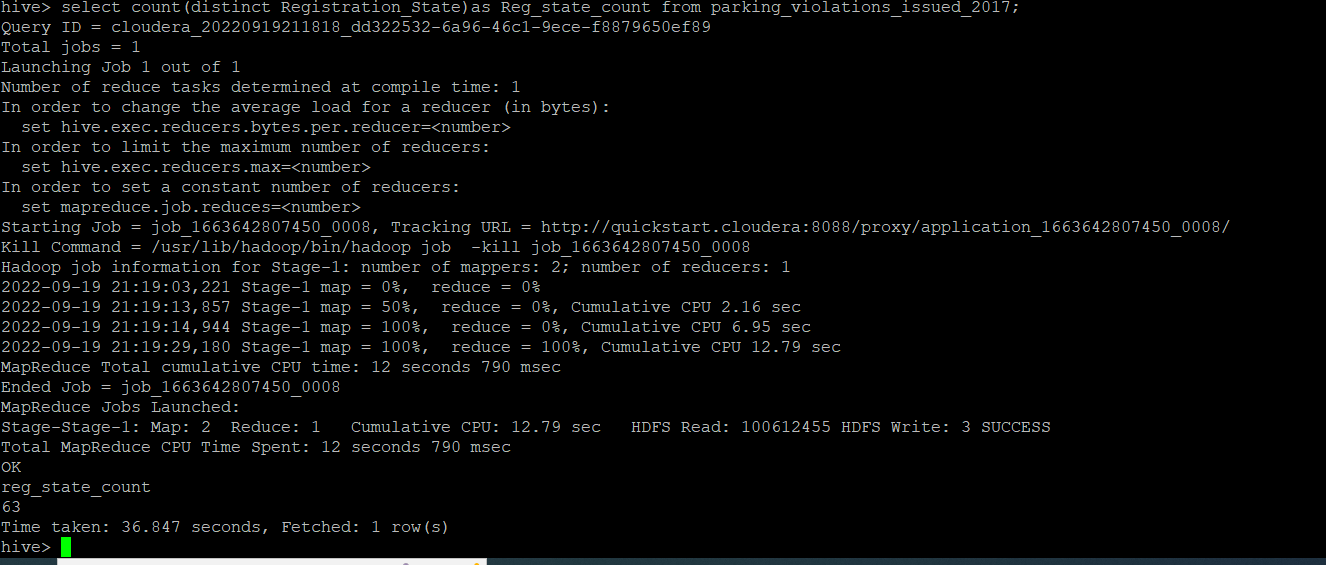
**Hive> select count(distinct summons\_number) No\_Tickets ,year(issue\_date) as year from parking\_violations\_issued\_2017 group by year(issue\_date);**



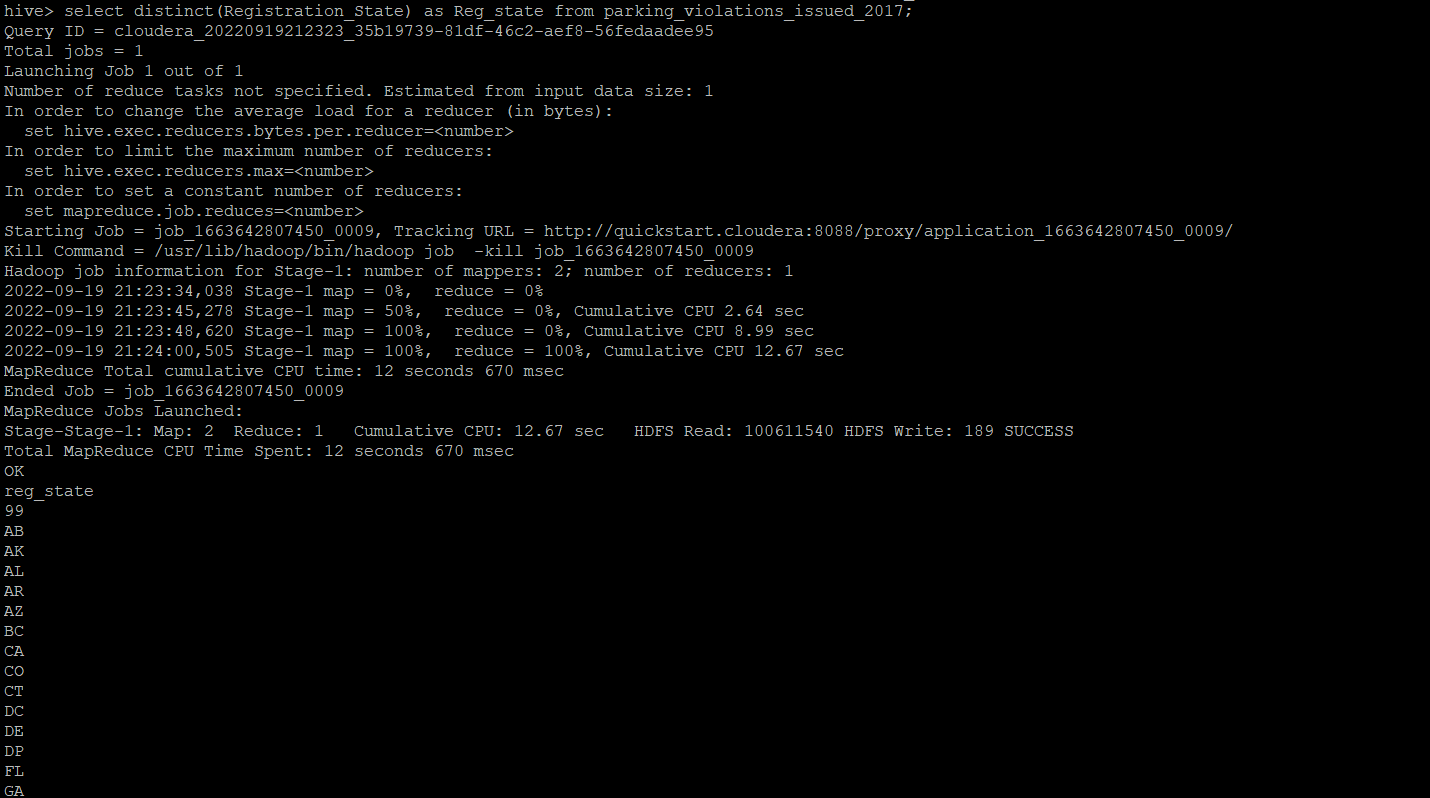
**Total number of tickets for the year 2017 are 539901**

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

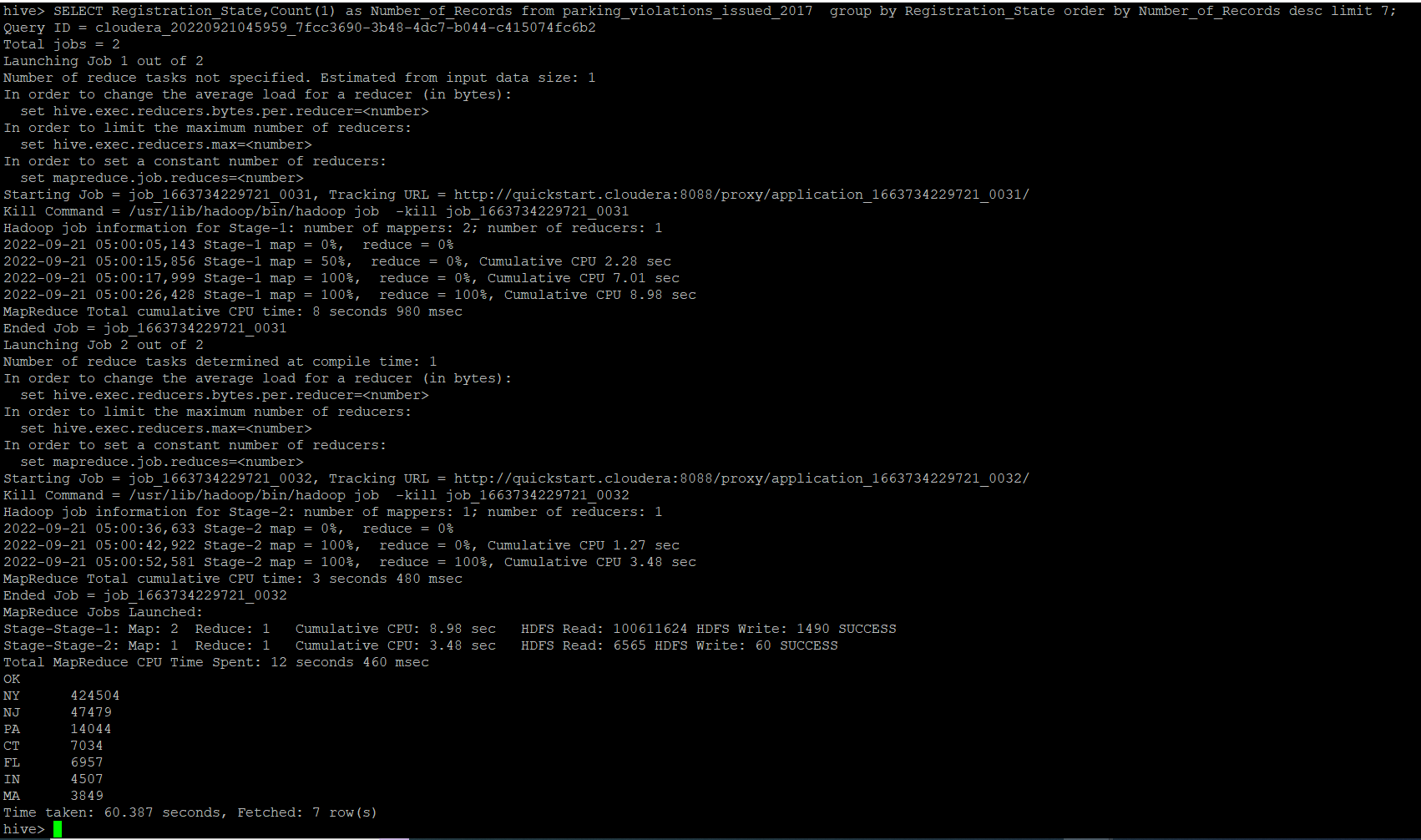
**Hive> select count(distinct Registration\_State) as Reg\_state\_count from parking\_violations\_issued\_2017;**



**Hive> select distinct(Registration\_State) as Reg\_state from parking\_violations\_issued\_2017;**

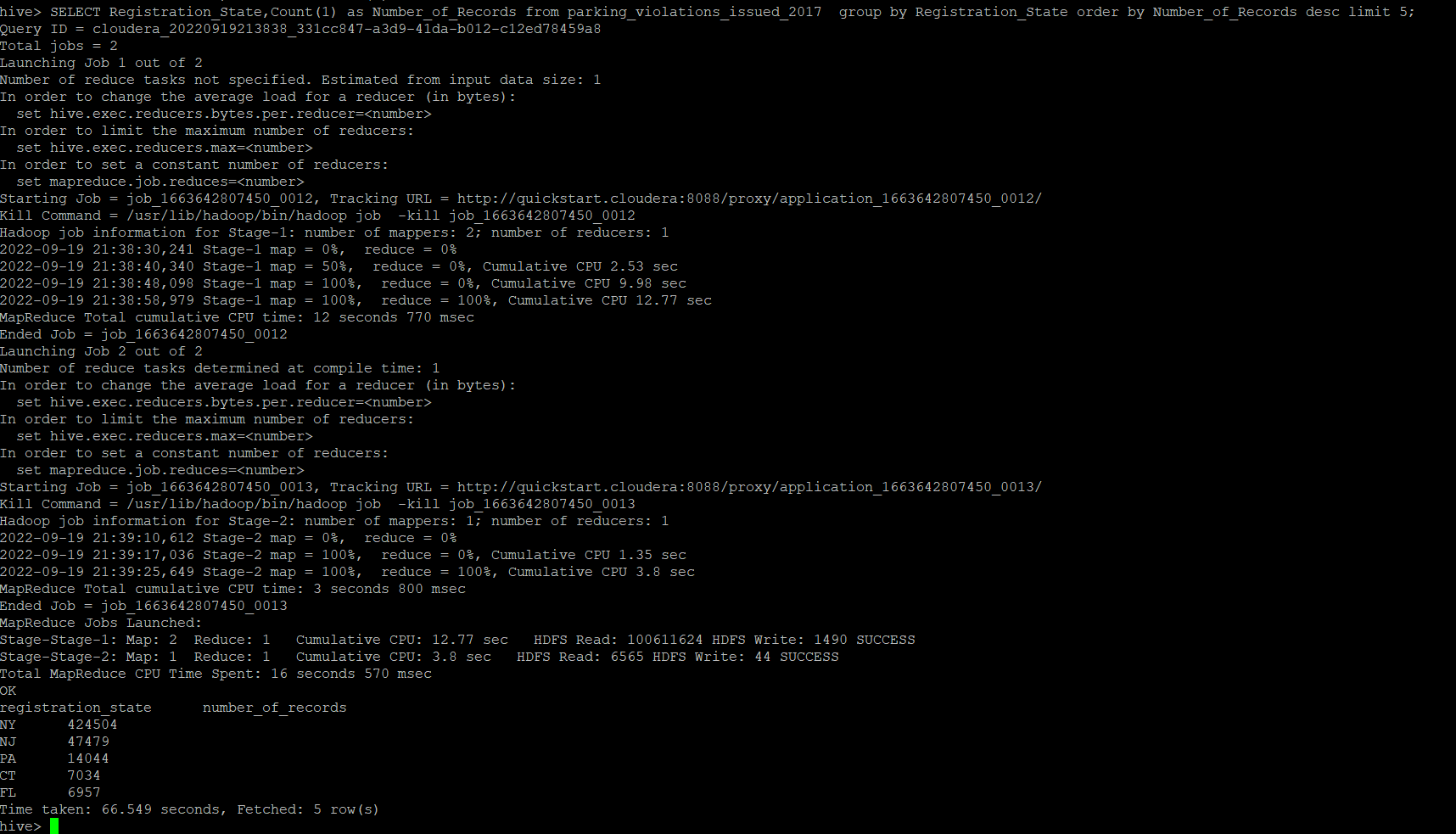


**Hive> SELECT Registration\_State,Count(1) as Number\_of\_Records from parking\_violations\_issued\_2017 group by Registration\_State order by Number\_of\_Records;**



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

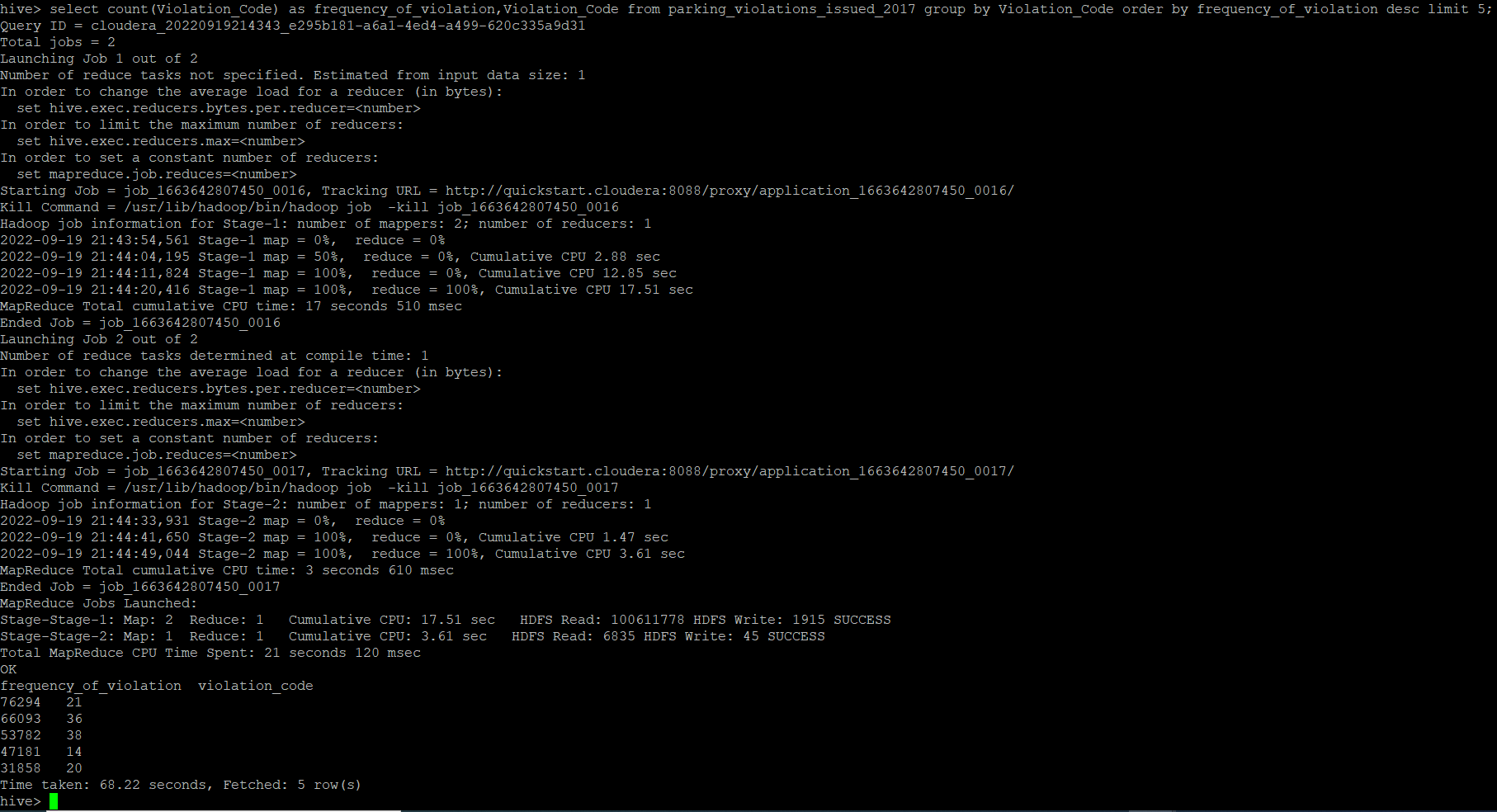
**Hive> select count(distinct summons\_number) as No\_Tickets\_without\_address from parking\_violations\_issued where Street\_code1 = 0 or Street\_code2 = 0 or Street\_code3 = 0;**



Part-II: Aggregation tasks

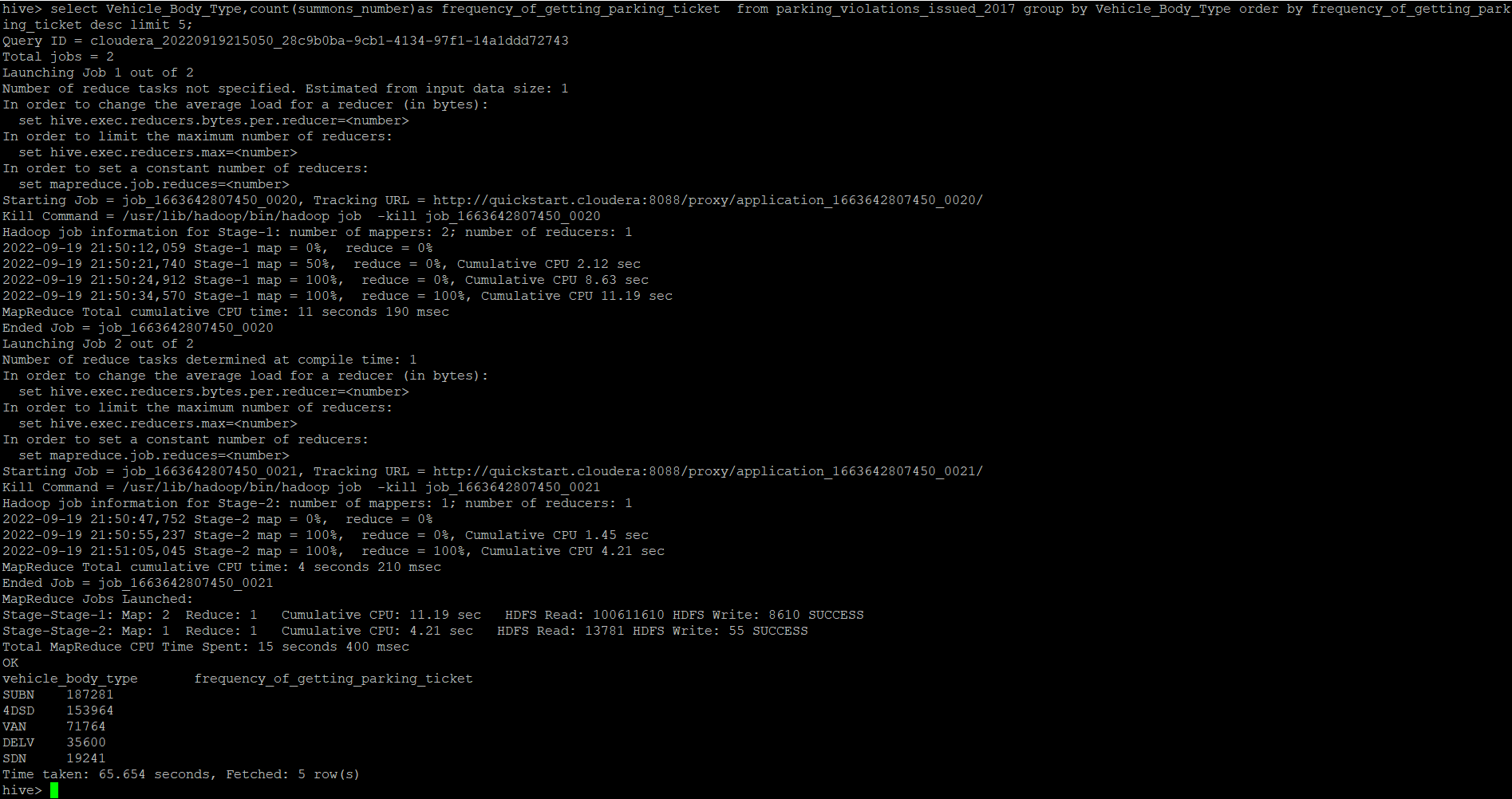
1.) How often does each violation code occur? (frequency of violation codes - find the top 5)

**Hive> select count(Violation\_Code) as frequency\_of\_violation,Violation\_Code from parking\_violations\_issued\_2017 group by Violation\_Code order by frequency\_of\_violation desc limit 5;**

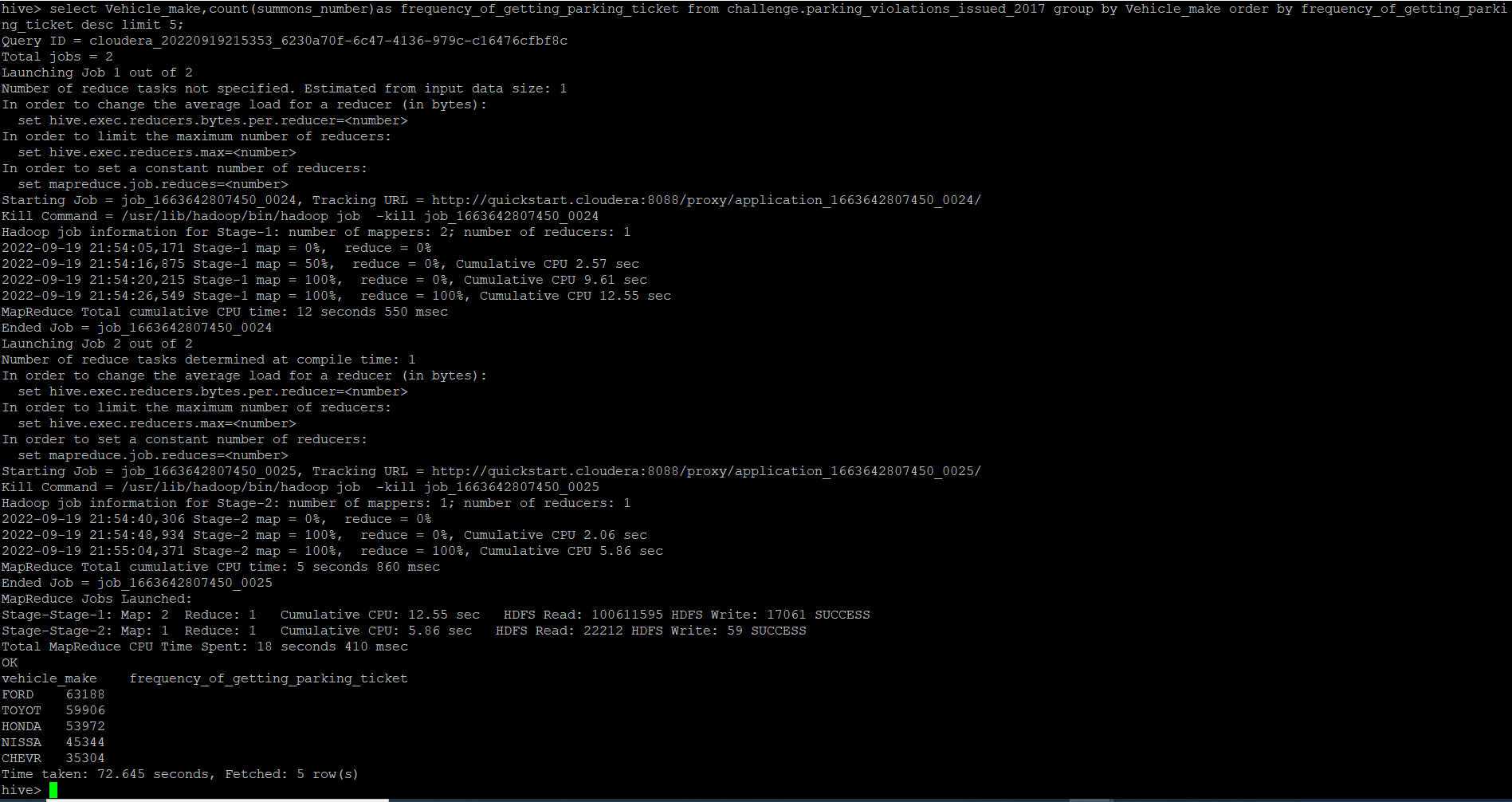


2.) How often does each vehicle body type get a parking ticket? How about the vehicle make? (find the top 5 for both)

**Hive> select Vehicle\_Body\_Type,count(summons\_number)as frequency\_of\_getting\_parking\_ticket from challenge.parking\_violations\_issued\_2017 group by Vehicle\_Body\_Type order by frequency\_of\_getting\_parking\_ticket desc limit 5; --done**



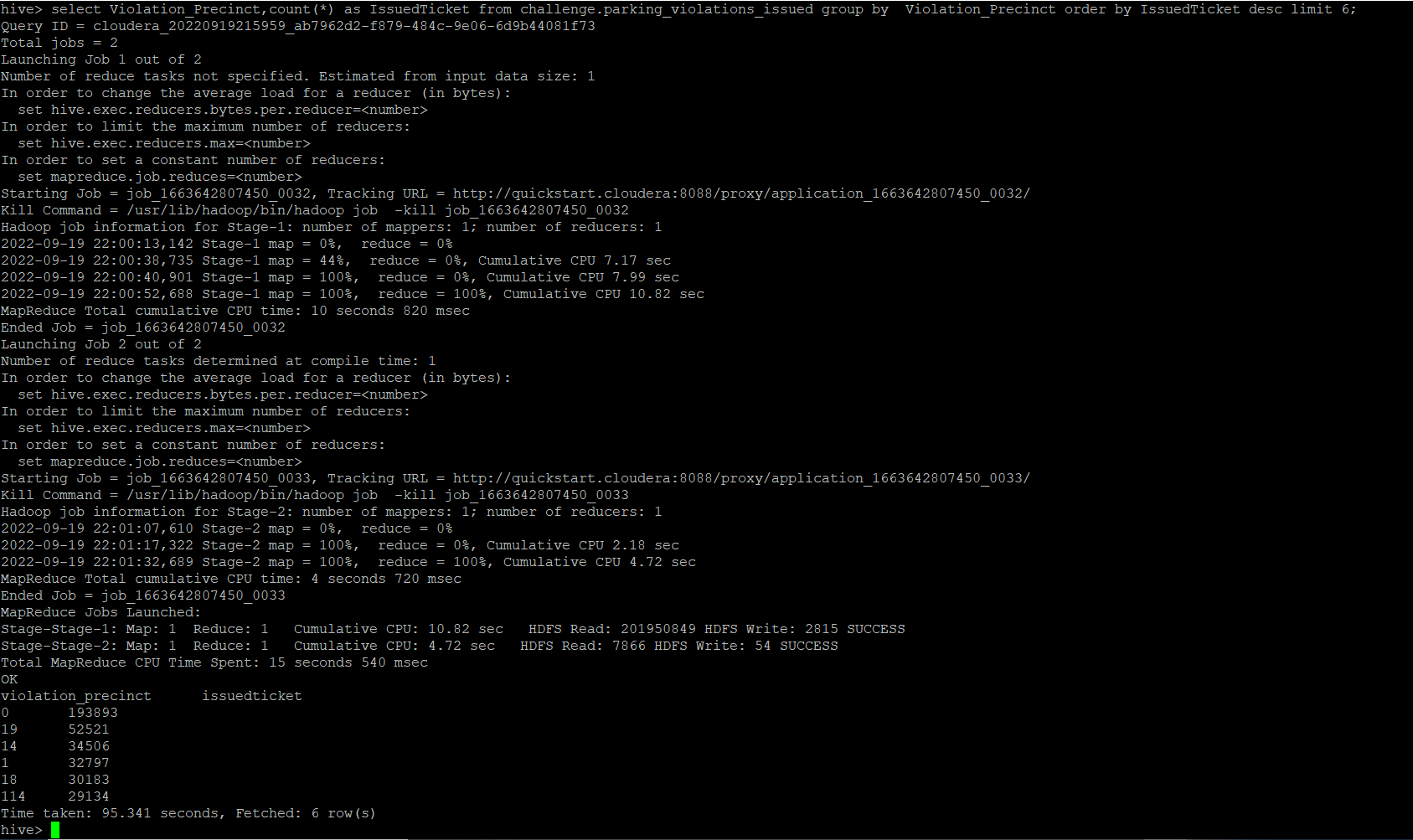
**Hive> select Vehicle\_make,count(summons\_number)as frequency\_of\_getting\_parking\_ticket from challenge.parking\_violations\_issued\_2017 group by Vehicle\_make order by frequency\_of\_getting\_parking\_ticket desc limit 5; --done**



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

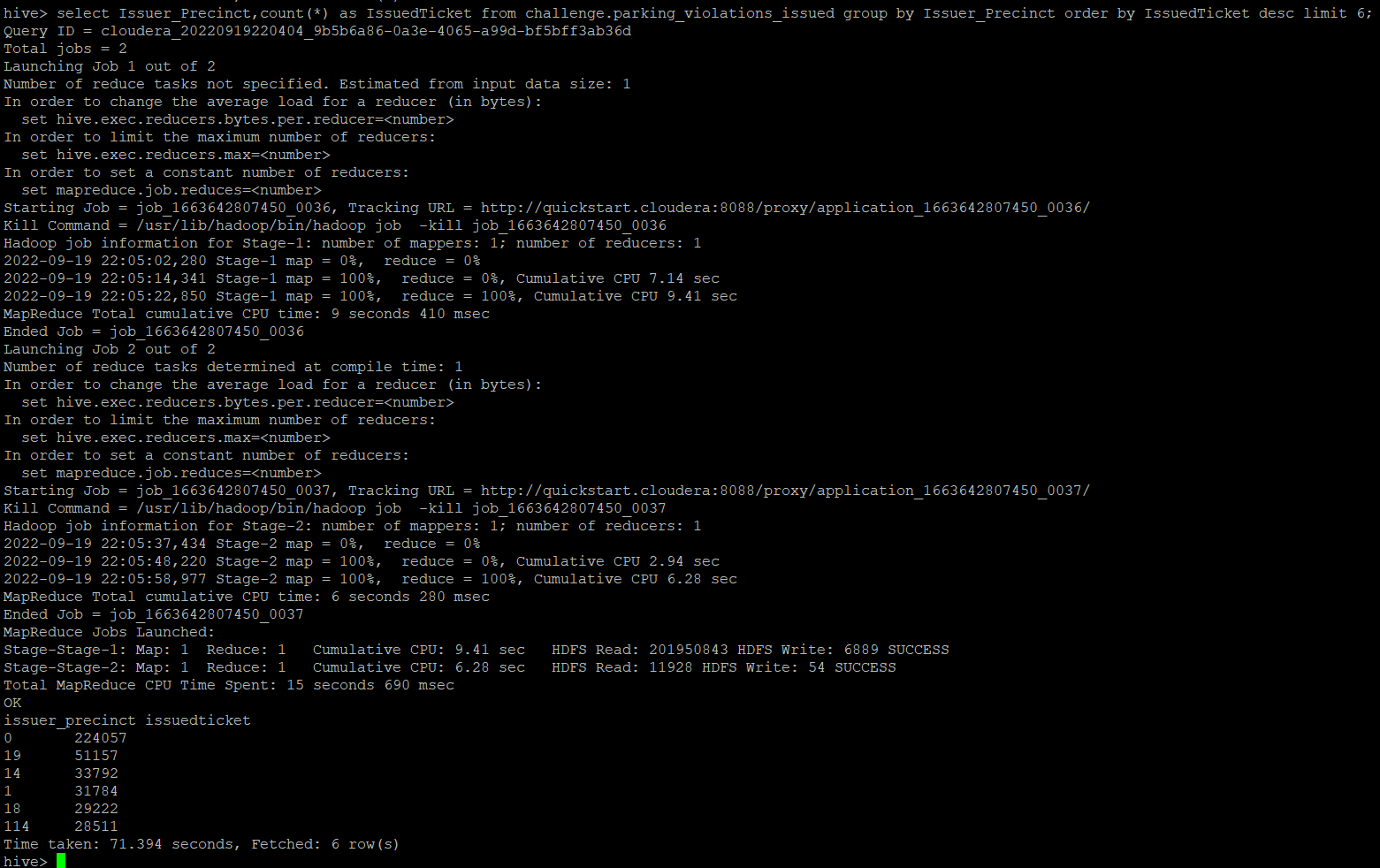
a.) Violating Precincts (this is the precinct of the zone where the violation occurred)

**hive> select Violation\_Precinct,count(\*) as IssuedTicket from challenge.parking\_violations\_issued group by Violation\_Precinct order by IssuedTicket desc limit 6;--correct**



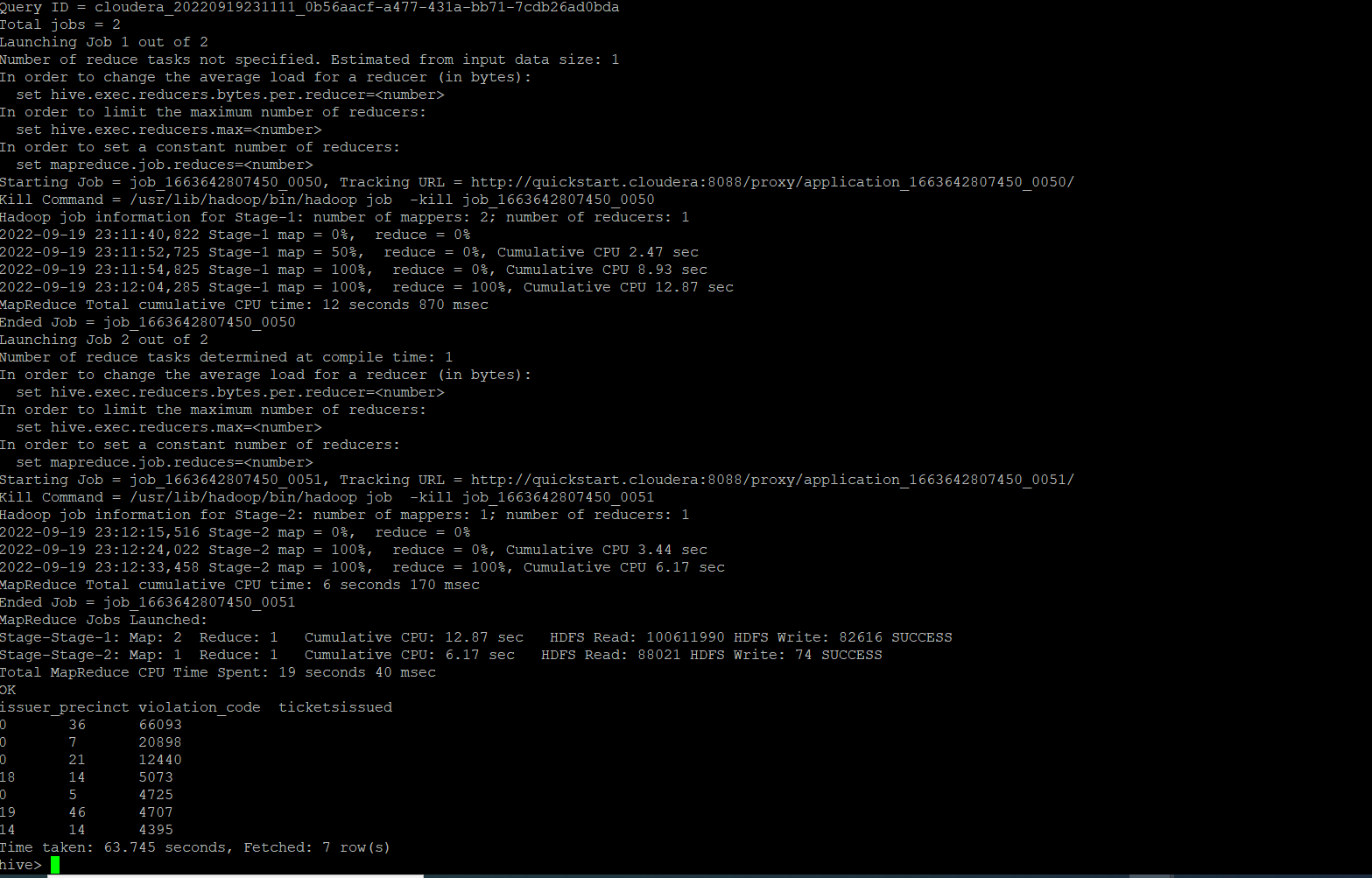
b.) Issuer Precincts (this is the precinct that issued the ticket)

**Hive> select Issuer\_Precinct,count(\*) as IssuedTicket from challenge.parking\_violations\_issued group by Issuer\_Precinct order by IssuedTicket desc limit 6;--correct**



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

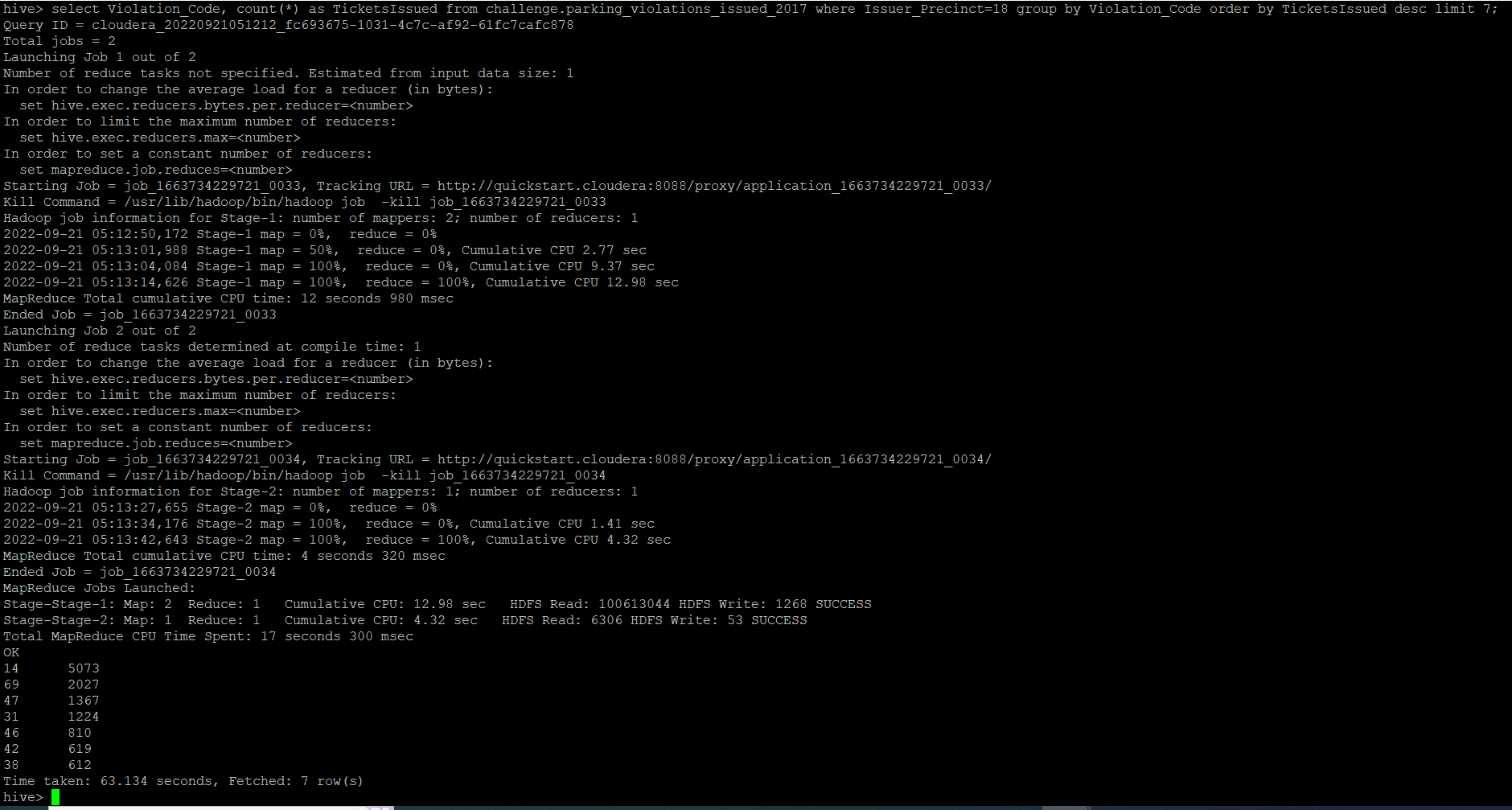
**select Issuer\_Precinct,Violation\_Code, count(\*) as TicketsIssued from challenge.parking\_violations\_issued\_2017 group by Issuer\_Precinct, Violation\_Code order by TicketsIssued desc limit 7;**



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

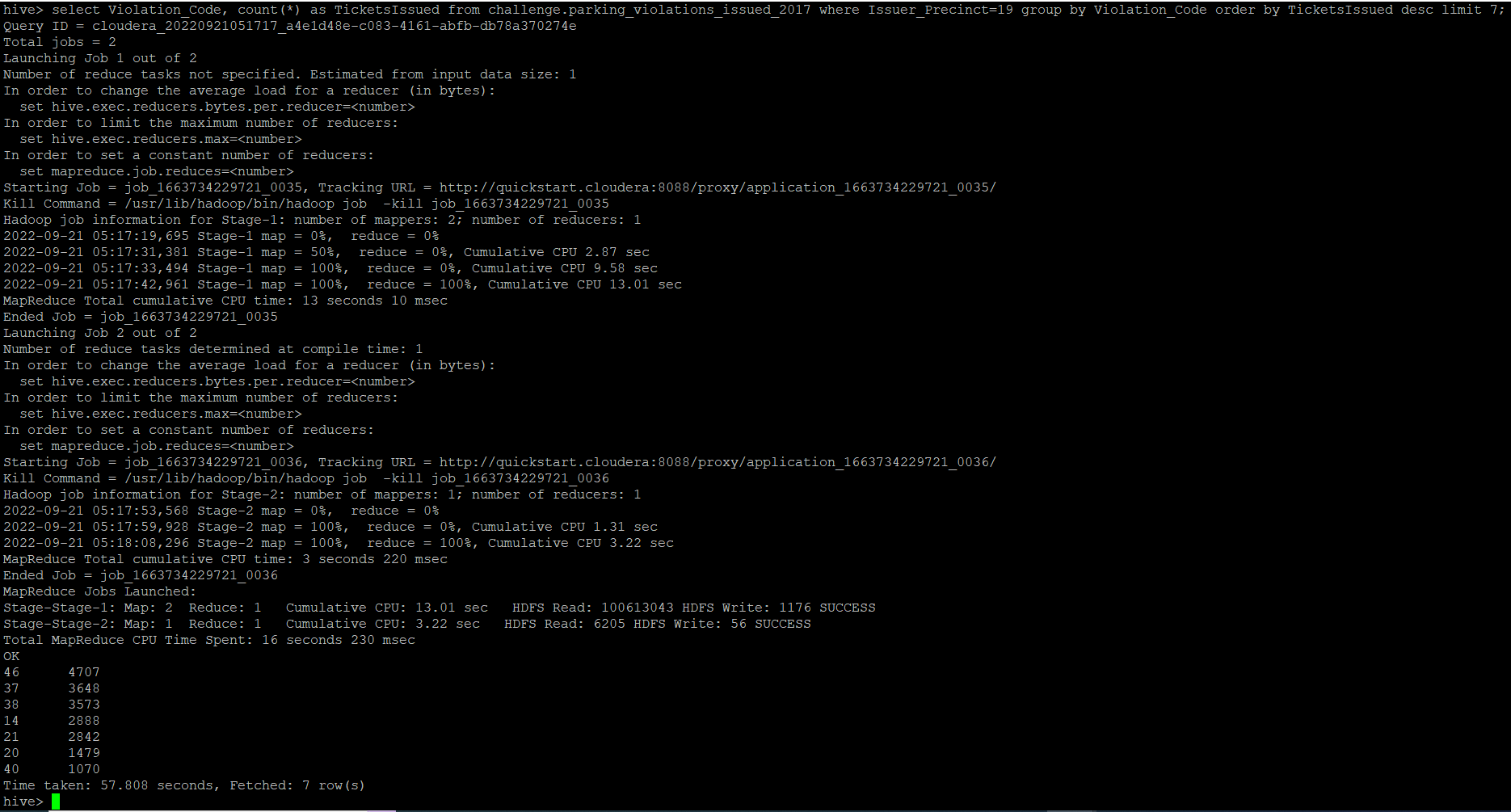
**--Issuer Precinct 18**

**select Violation\_Code, count(\*) as TicketsIssued from challenge.parking\_violations\_issued\_2017 where Issuer\_Precinct=18 group by Violation\_Code order by TicketsIssued desc limit 7;**



**--Issuer Precinct 19**

**select Violation\_Code, count(\*) as TicketsIssued from challenge.parking\_violations\_issued\_2017 where Issuer\_Precinct=19 group by Violation\_Code order by TicketsIssued desc limit 7;**



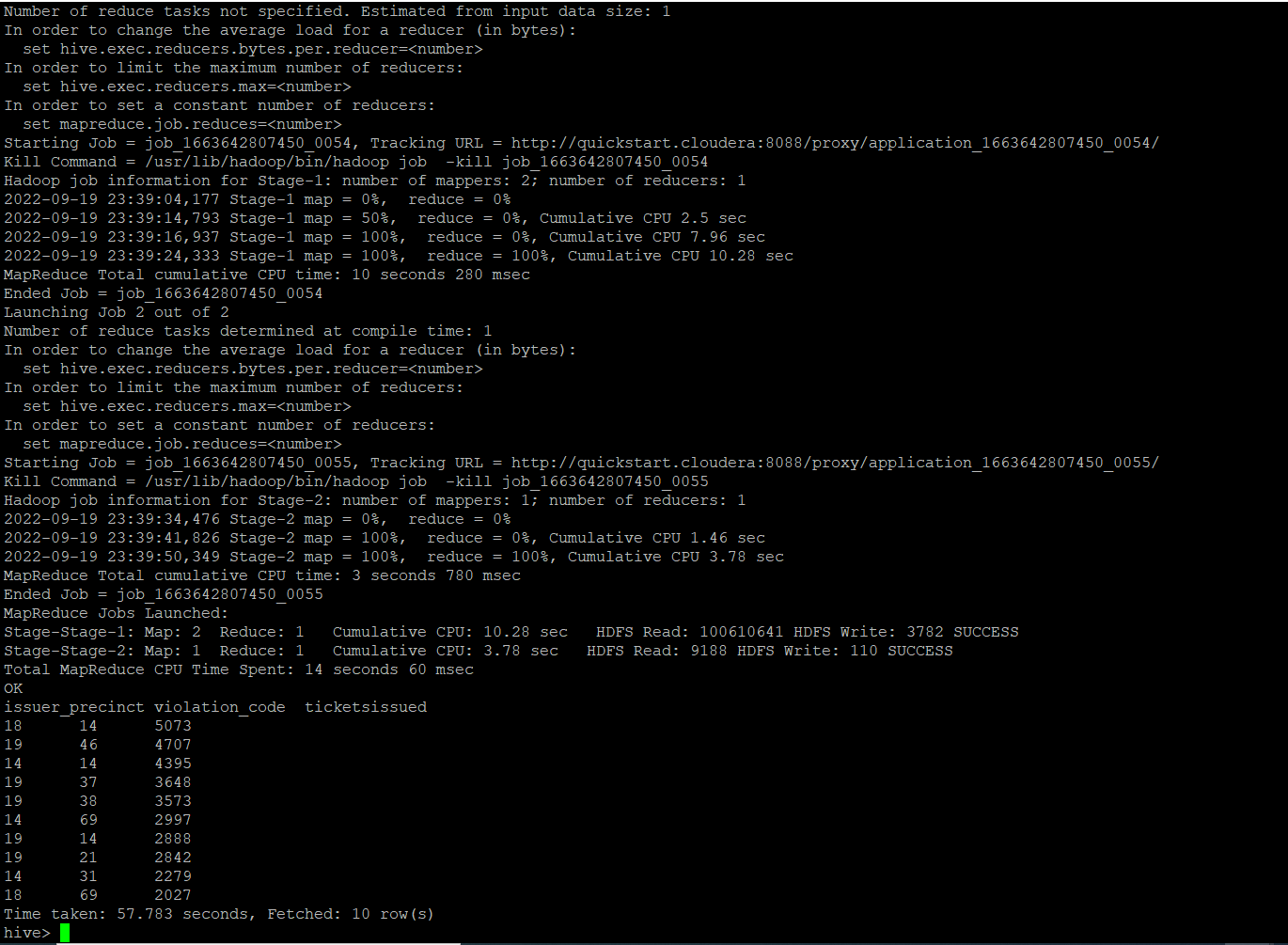
**--Issuer Precinct 14**

**Hive> select Violation\_Code, count(\*) as TicketsIssued from challenge.parking\_violations\_issued\_2017 where Issuer\_Precinct=14 group by Violation\_Code order by TicketsIssued desc limit 7;**

**--Common codes accross precincts**

**Hive> select Issuer\_Precinct,Violation\_Code, count(\*) as TicketsIssued from challenge.parking\_violations\_issued\_2017 where Issuer\_Precinct in (18,19,14)**

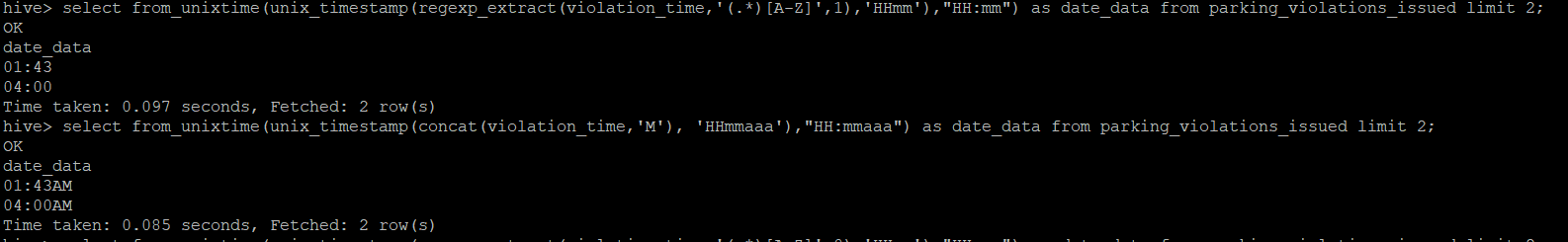
**group by Issuer\_Precinct,Violation\_Code order by TicketsIssued desc limit 10;**



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

**Hive> select from\_unixtime(unix\_timestamp(regexp\_extract(violation\_time,'(.\*)[A-Z]',1),'HHmm'),"HH:mm") as date\_data from parking\_violations\_issued limit 2;--> converted to time format 01:43**

**Hive> select from\_unixtime(unix\_timestamp(concat(violation\_time,'M'), 'HHmmaaa'),"HH:mmaaa") as date\_data from parking\_violations\_issued limit 2;--> working 01:43AM**



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

**partitoned view :**

**hive> create view vw\_parking\_violations\_2017\_partitioned\_bins partitioned on (Violation\_Code) as**

**SELECT Summons\_Number, Violation\_Time, Issuer\_Precinct,**

**case**

**when substring(Violation\_Time,1,2) in ('00','01','02','03','12') and upper(substring(Violation\_Time,-1))='A' then 1**

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

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

**when substring(Violation\_Time,1,2) in ('12','00','01','02','03') and upper(substring(Violation\_Time,-1))='P' then 4**

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

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

**else null end as Violation\_Time\_bin,Violation\_Code**

**from parking\_violations\_issued\_2017**

**where Violation\_Time is not null or (length(Violation\_Time)=5 and upper(substring(Violation\_Time,-1))in ('A','P')**

**and substring(Violation\_Time,1,2) in ('00','01','02','03','04','05','06','07', '08','09','10','11','12'));**

----------------------------------------------------------------------------------------------------------------

**bin1**

**select Violation\_Code,count(\*) TicketsIssued from vw\_parking\_violations\_2017\_partitioned\_bins where Violation\_Time\_bin == 1 group by Violation\_Code order by TicketsIssued desc limit 3;**

|  |  |
| --- | --- |
| **Violation\_code** | **TicktesIssued** |
| **21** | **3660** |
| **40** | **2584** |
| **14** | **1574** |

**bin2**

**select Violation\_Code,count(\*) TicketsIssued from vw\_parking\_violations\_2017\_partitioned\_bins where Violation\_Time\_bin == 2 group by Violation\_Code order by TicketsIssued desc limit 3;**

|  |  |
| --- | --- |
| **Violation\_code** | **TicktesIssued** |
| **14** | **7250** |
| **40** | **6403** |
| **21** | **5669** |

**bin3**

**select Violation\_Code,count(\*) TicketsIssued from vw\_parking\_violations\_2017\_partitioned\_bins where Violation\_Time\_bin == 3 group by Violation\_Code order by TicketsIssued desc limit 3;**

|  |  |
| --- | --- |
| **Violation\_code** | **TicktesIssued** |
| **21** | **59465** |
| **36** | **37767** |
| **38** | **17587** |

**bin4**

**select Violation\_Code,count(\*) TicketsIssued from vw\_parking\_violations\_2017\_partitioned\_bins where Violation\_Time\_bin == 4 group by Violation\_Code order by TicketsIssued desc limit 3;**

|  |  |
| --- | --- |
| **Violation\_code** | **TicktesIssued** |
| **36** | **28600** |
| **38** | **23877** |
| **37** | **16777** |

**bin5**

**select Violation\_Code,count(\*) TicketsIssued from vw\_parking\_violations\_2017\_partitioned\_bins where Violation\_Time\_bin == 5 group by Violation\_Code order by TicketsIssued desc limit 3;**

|  |  |
| --- | --- |
| **Violation\_code** | **TicktesIssued** |
| **38** | **10148** |
| **14** | **7609** |
| **37** | **6944** |

**bin6**

**select Violation\_Code,count(\*) TicketsIssued from vw\_parking\_violations\_2017\_partitioned\_bins where Violation\_Time\_bin == 6 group by Violation\_Code order by TicketsIssued desc limit 3;**

|  |  |
| --- | --- |
| **Violation\_code** | **TicktesIssued** |
| **7** | **2602** |
| **40** | **2159** |
| **14** | **2091** |

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

**Hive> select Violation\_Time\_bin, count(\*) TicketsIssued from vw\_parking\_violations\_2017\_partitioned\_bins where Violation\_Code in (21, 37, 38,36)**

**group by Violation\_Time\_bin order by TicketsIssued desc limit 3;**

|  |  |
| --- | --- |
| **Violation\_Time\_bin** | **TicketsIssued** |
| **3** | **116785** |
| **4** | **76701** |
| **5** | **18437** |

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

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

**Season Month interval**

**spring March, April, May**

**summer June, July, August**

**autumn September, October, November**

**winter December, January, February**

**normal view**

**----------------------**

**Hive> create view vw\_tickets\_issued\_2017\_bins as**

**select Violation\_Code , Issuer\_Precinct,**

**case**

**when MONTH(Issue\_Date) between 03 and 05 then 'spring'**

**when MONTH(Issue\_Date) between 06 and 08 then 'summer'**

**when MONTH(Issue\_Date) between 09 and 11 then 'autumn'**

**when MONTH(Issue\_Date) in (1,2,12) then 'winter'**

**else 'unknown' end as season from parking\_violations\_issued\_2017;**

**--------------------------------------------------------------------------------------------------------------**

**partioned view :**

**----------------------**

**Hive> create view vw\_tickets\_issued\_2017\_partitioned\_bins partitioned on (Violation\_Code) as**

**select Issuer\_Precinct,**

**case**

**when MONTH(Issue\_Date) between 03 and 05 then 'spring'**

**when MONTH(Issue\_Date) between 06 and 08 then 'summer'**

**when MONTH(Issue\_Date) between 09 and 11 then 'autumn' select**

**when MONTH(Issue\_Date) in (1,2,12) then 'winter'**

**else 'unknown' end as season,Violation\_Code from parking\_violations\_issued\_2017;**

**Hive> select season, count(\*) as TicketsIssued from vw\_tickets\_issued\_2017\_partitioned\_bins group by season order by TicketsIssued desc;**

|  |  |
| --- | --- |
| **Season** | **TicktesIssued** |
| **Spring** | **285875** |
| **Winter** | **169466** |
| **Summer** | **84560** |
| **autumn** | **0** |

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

**# spring season**

**select Violation\_Code, count(\*) as TicketsIssued from vw\_tickets\_issued\_2017\_partitioned\_bins where**

**season = 'spring' group by Violation\_Code order by TicketsIssued desc limit 3;**

|  |  |
| --- | --- |
| **Violation\_Code** | **TicketsIssued** |
| **21** | **40045** |
| **36** | **34354** |
| **38** | **27001** |

**# winter season**

**select Violation\_Code, count(\*) as TicketsIssued from vw\_tickets\_issued\_2017\_partitioned\_bins where**

**season = 'winter' group by Violation\_Code order by TicketsIssued desc limit 3;**

|  |  |
| --- | --- |
| **Violation\_Code** | **TicketsIssued** |
| **21** | **23684** |
| **36** | **22084** |
| **38** | **18450** |

**# summer season**

**select Violation\_Code, count(\*) as TicketsIssued from vw\_tickets\_issued\_2017\_partitioned\_bins where**

**season = 'summer' group by Violation\_Code order by TicketsIssued desc limit 3;**

|  |  |
| --- | --- |
| **Violation\_Code** | **TicketsIssued** |
| **21** | **12565** |
| **36** | **9655** |
| **38** | **8331** |

**# autumn season**

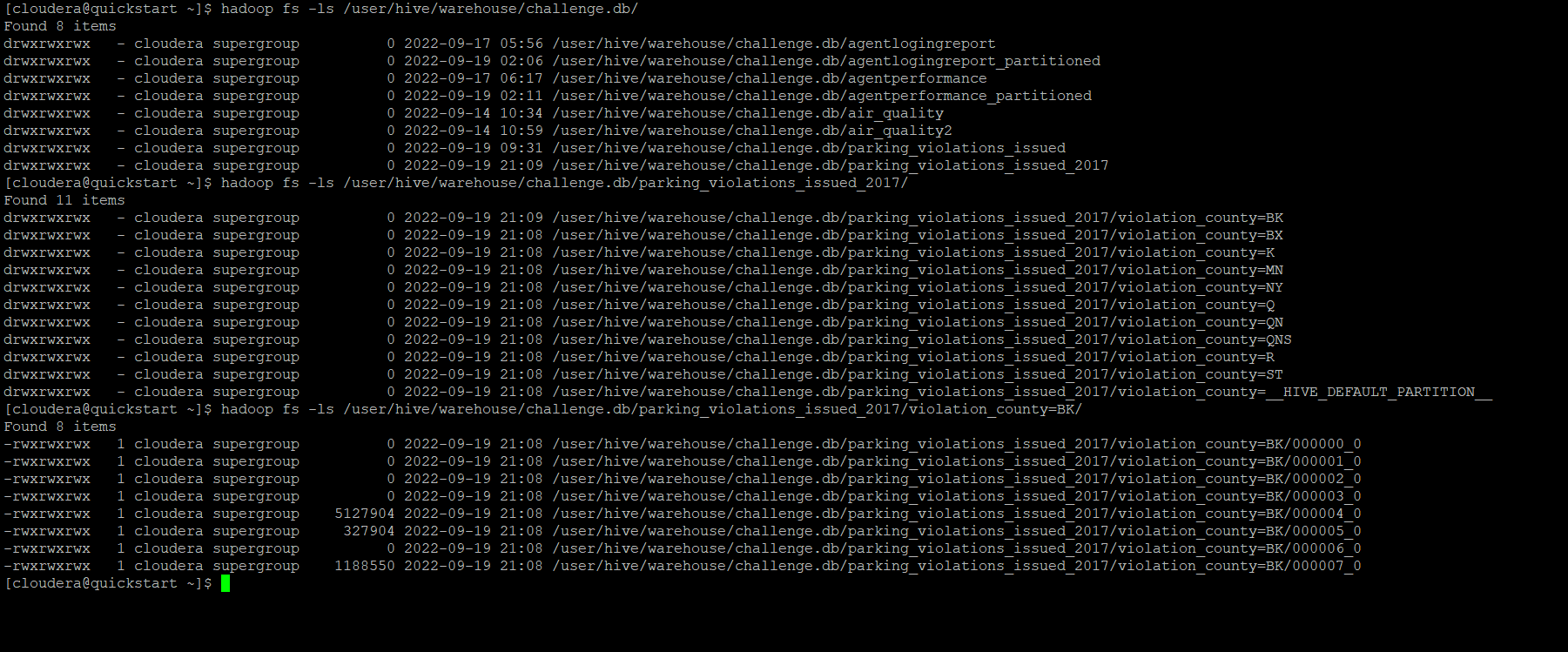
**select Violation\_Code, count(\*) as TicketsIssued from vw\_tickets\_issued\_2017\_partitioned\_bins where**

**season = 'autumn' group by Violation\_Code order by TicketsIssued desc limit 3;**

|  |  |
| --- | --- |
| **Violation\_Code** | **TicketsIssued** |
|  |  |

I have used partionsed and Bucketing on Table - **parking\_violations\_issued\_2017** , and partions on views - **vw\_parking\_violations\_2017\_partitioned\_bins partitioned, vw\_tickets\_issued\_2017\_bins**

That improves query performance.



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