**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.**

**Step:1) Create Table:-**

hive> create table parking\_violations\_issued

> (

> 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,

> 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\_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");

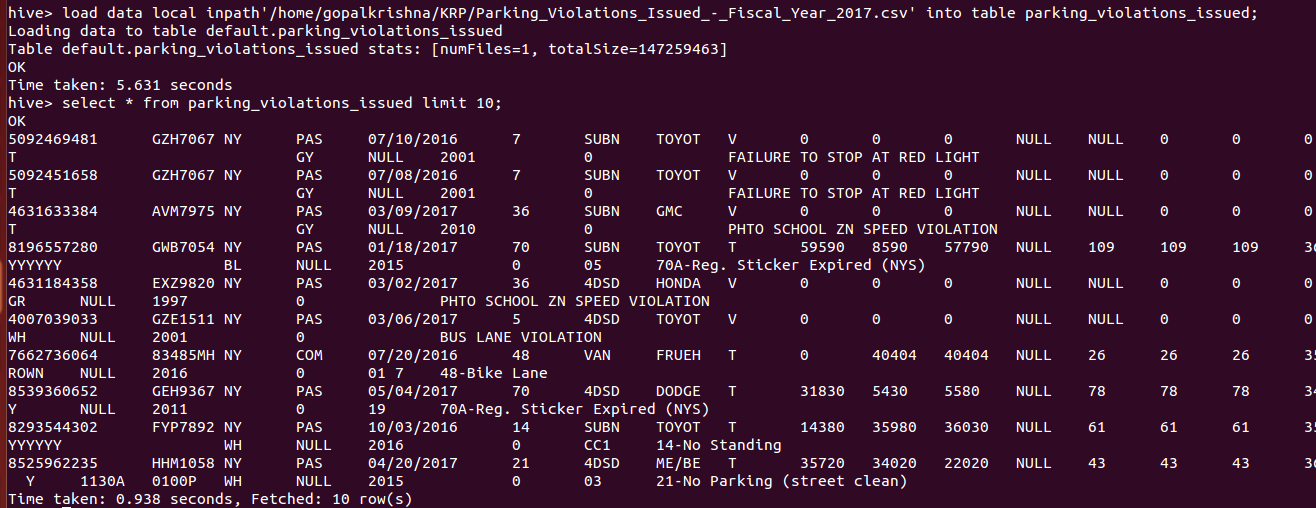
****

**Load data from LFS to the above Table:-**

hive> load data local inpath'/home/gopalkrishna/KRP/Parking\_Violations\_Issued\_-\_Fiscal\_Year\_2017.csv' into table parking\_violations\_issued;

**Check the data is loaded or not:-**

hive> select \* from parking\_violations\_issued limit 10;



-We can see below also, that the data is loaded in hive default warehouse where the table “parking\_violations\_issued” is created.



**Creating a table to store only the tickets issued in year 2017:-**

create table parking\_violations\_issued\_2017

(

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,

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'

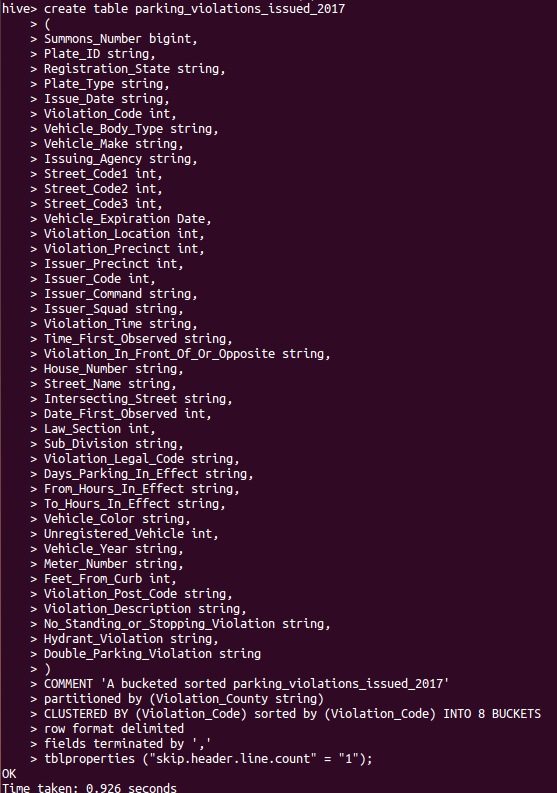
partitioned by (Violation\_County string)

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

row format delimited

fields terminated by ','

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

****

**Set below 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;

****

**Load data from parking\_violations\_issued to parking\_violations\_issued\_2017:-**

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,

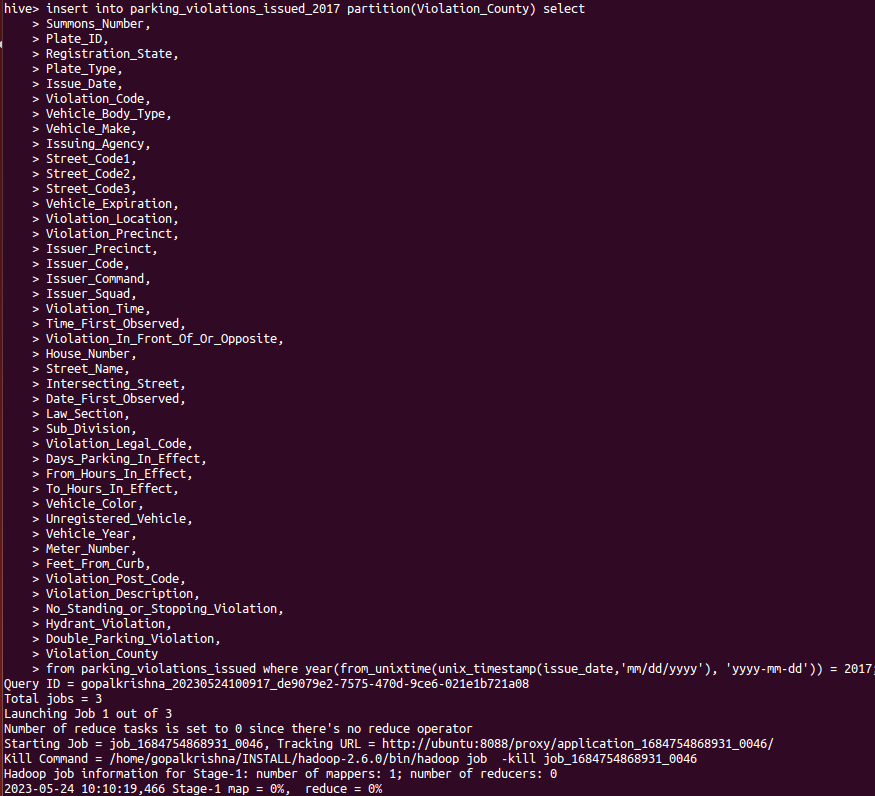
No\_Standing\_or\_Stopping\_Violation,

Hydrant\_Violation,

Double\_Parking\_Violation,

Violation\_County

from parking\_violations\_issued where year(from\_unixtime(unix\_timestamp(issue\_date,'mm/dd/yyyy'), 'yyyy-mm-dd')) = 2017;

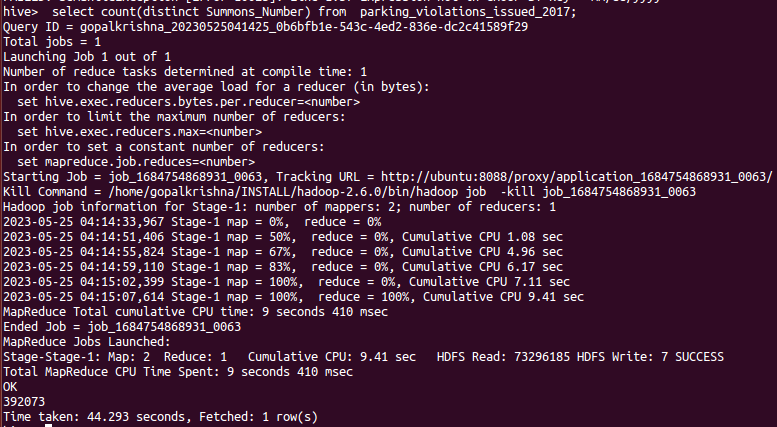
****

**The analysis can be divided into two parts:**

**Part-I: Examine the data**

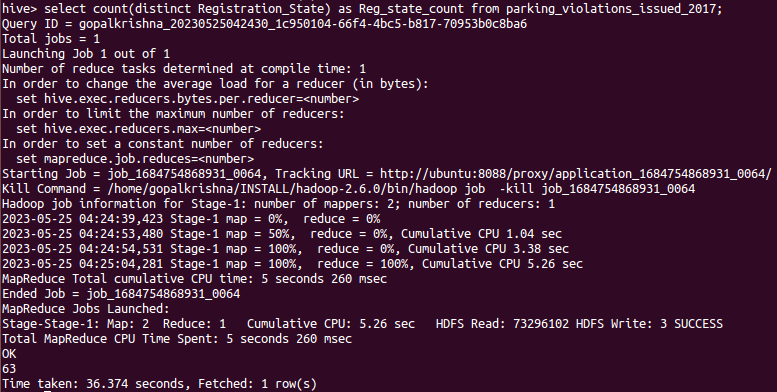
**1.) Find the total number of tickets for the year.**

hive> select count(distinct Summons\_Number) from parking\_violations\_issued\_2017;



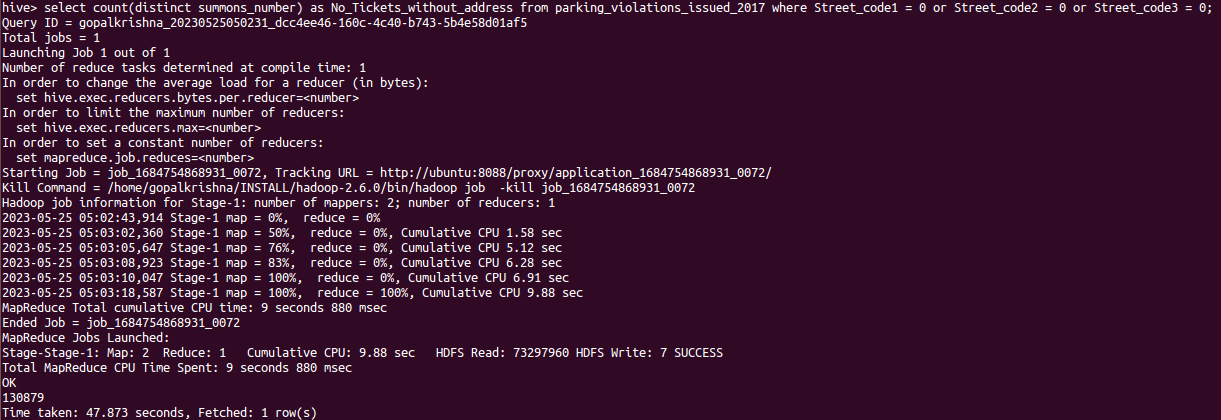
**2.) Find out how many unique states the cars which got parking tickets came from.**

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



**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 )**

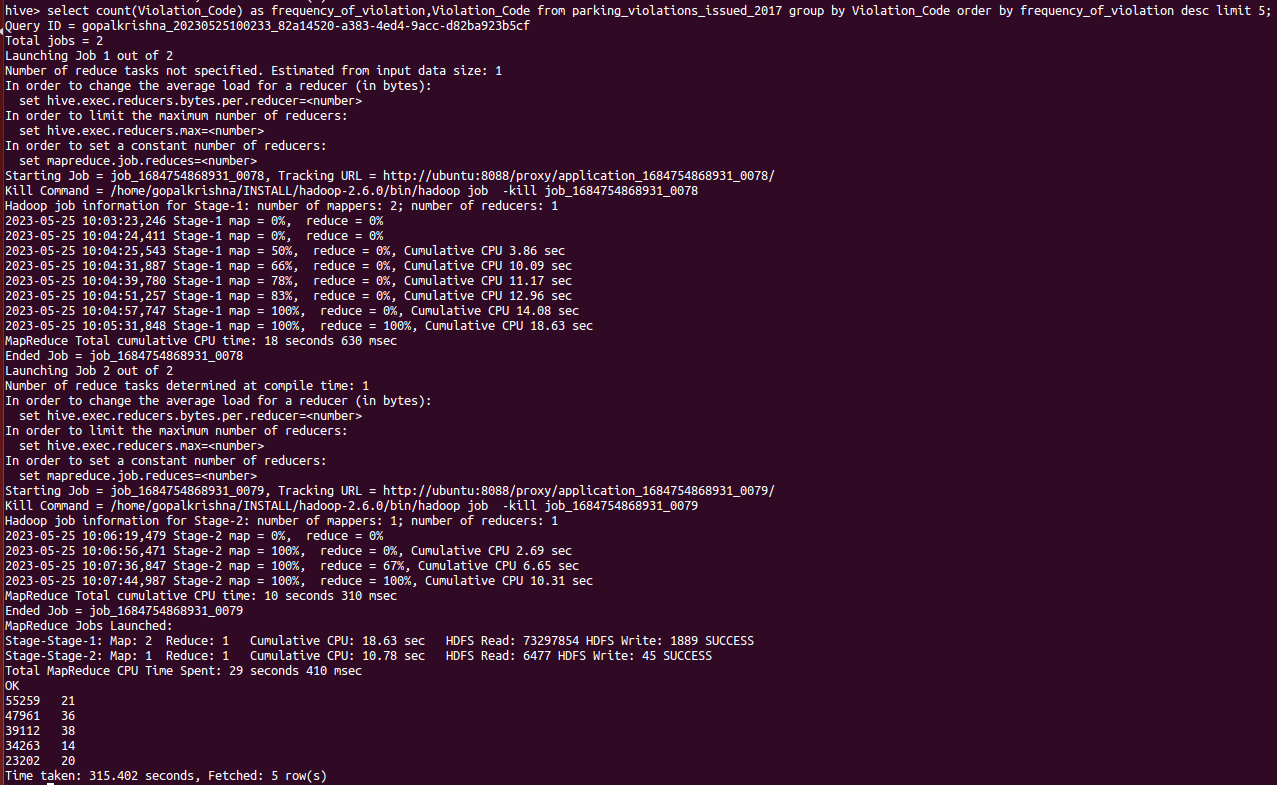
hive> select count(distinct summons\_number) as No\_Tickets\_without\_address from parking\_violations\_issued\_2017 where Street\_code1 = 0 or Street\_code2 = 0 or Street\_code3 = 0;

****

**Part-II: Aggregation tasks**

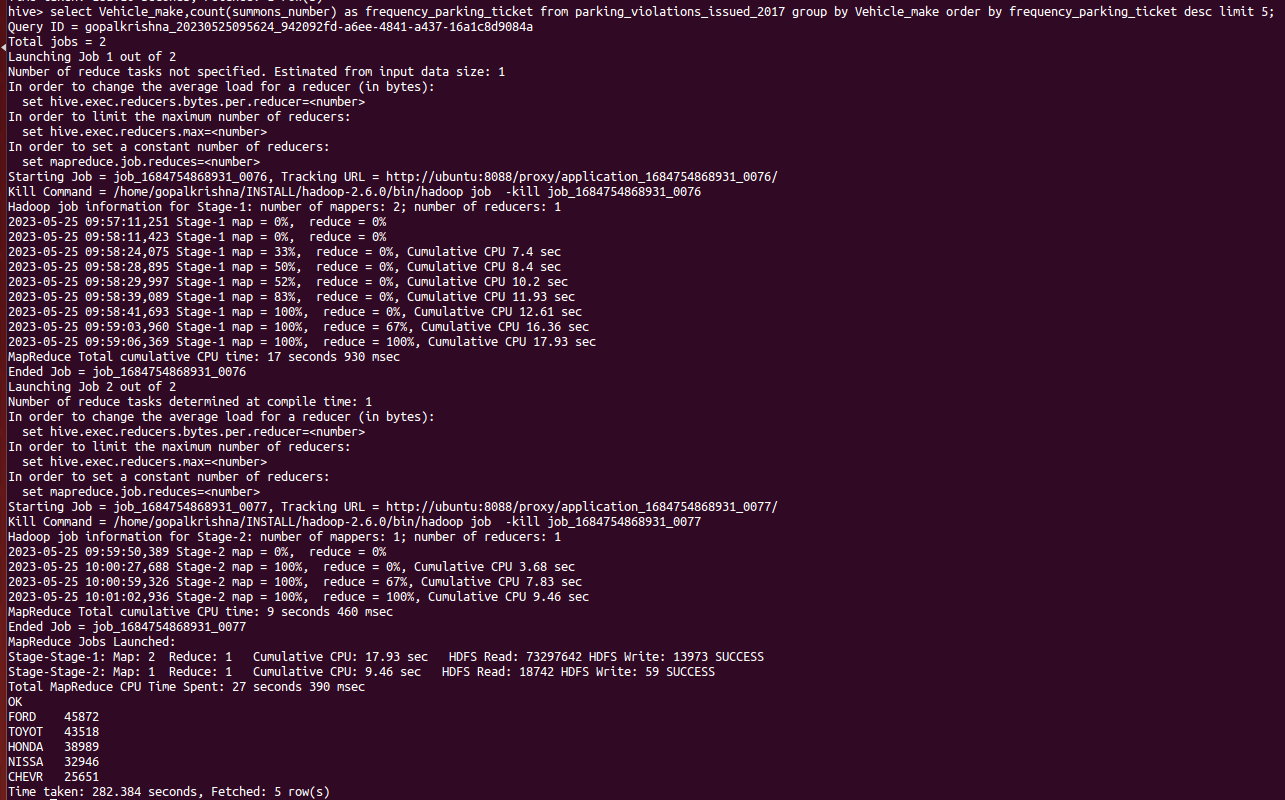
**1.) How often does each violation code occur? (frequency of violation codes - find the top 5)**

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)**

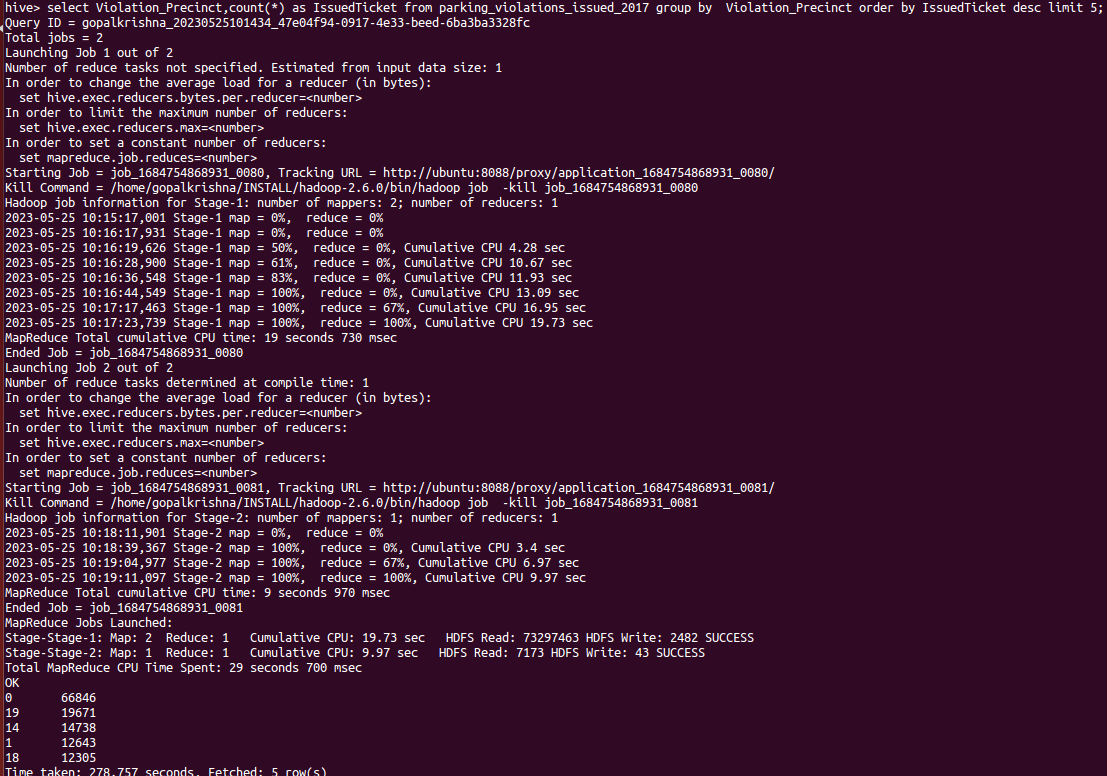
hive> select Vehicle\_make,count(summons\_number) as frequency\_parking\_ticket from parking\_violations\_issued\_2017 group by Vehicle\_make order by frequency\_parking\_ticket desc limit 5;

****

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

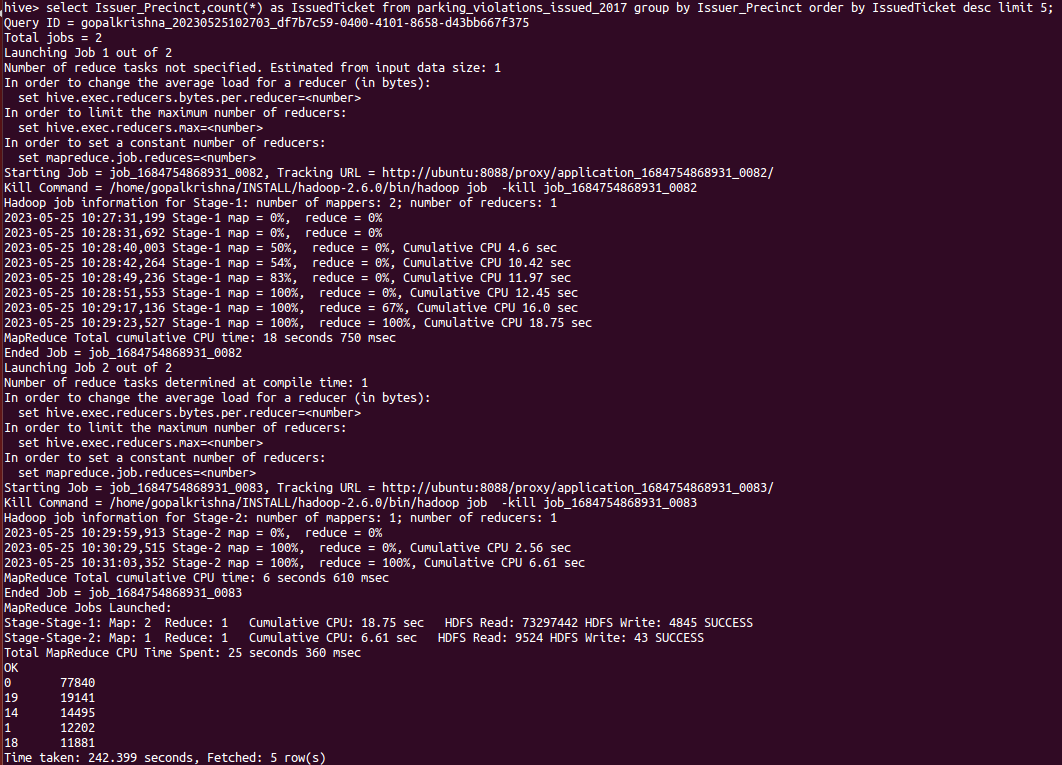
**a.) Violating Precincts (this is the precinct of the zone where the violation occurred)**

hive> select Violation\_Precinct,count(\*) as IssuedTicket from parking\_violations\_issued\_2017 group by Violation\_Precinct order by IssuedTicket desc limit 5;

****

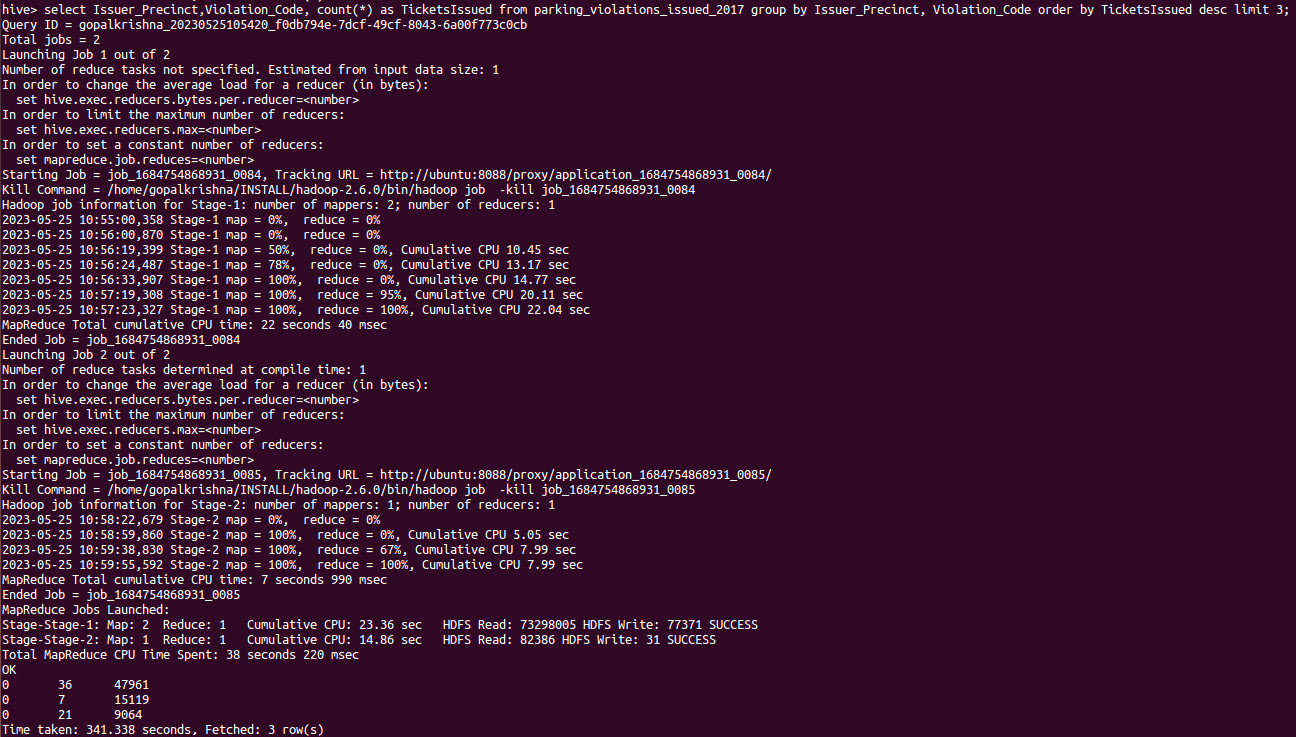
**b.) Issuer Precincts (this is the precinct that issued the ticket)**

select Issuer\_Precinct,count(\*) as IssuedTicket from parking\_violations\_issued\_2017 group by Issuer\_Precinct order by IssuedTicket desc limit 5;



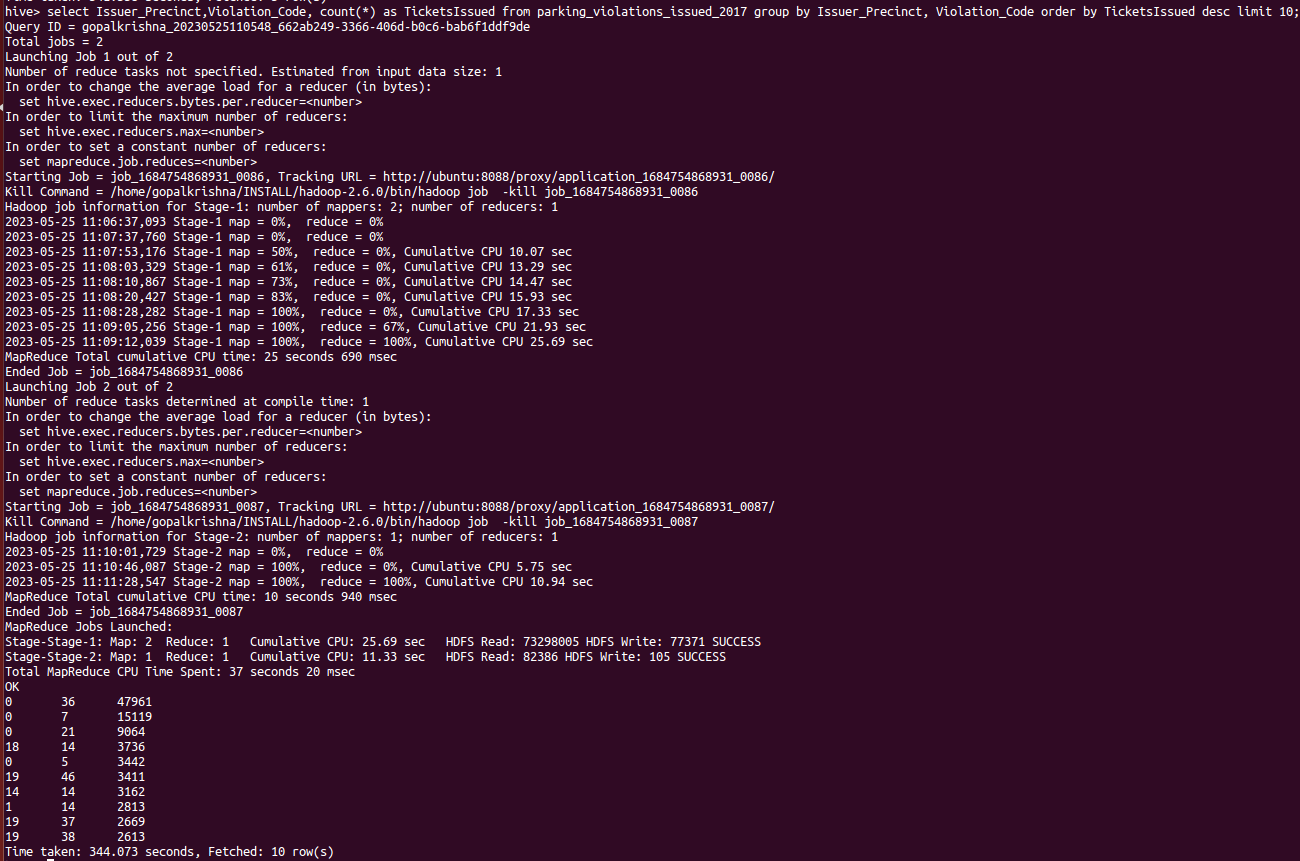
**4.) Find the violation code frequency across 3 precincts which have issued the most number of tickets - do these precinct zones have an exceptionally high frequency of certain violation codes?**

select Issuer\_Precinct,Violation\_Code, count(\*) as TicketsIssued from parking\_violations\_issued\_2017 group by Issuer\_Precinct, Violation\_Code order by TicketsIssued desc limit 3;

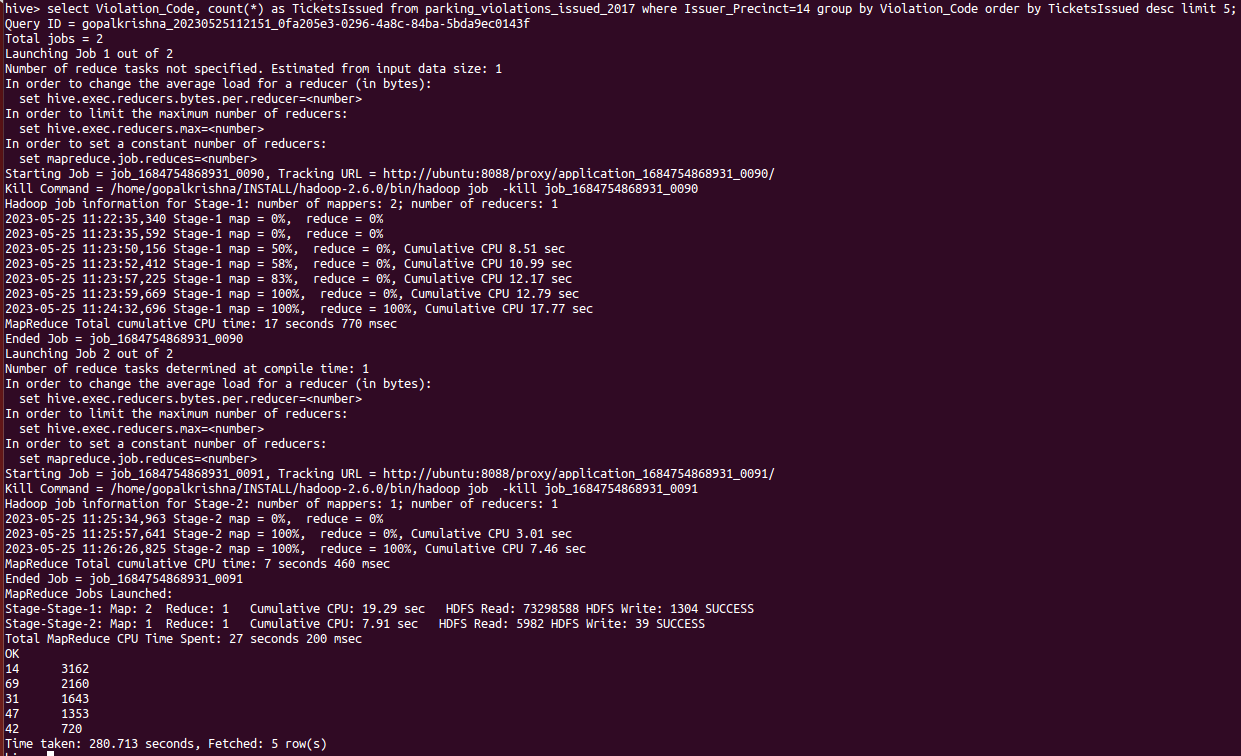


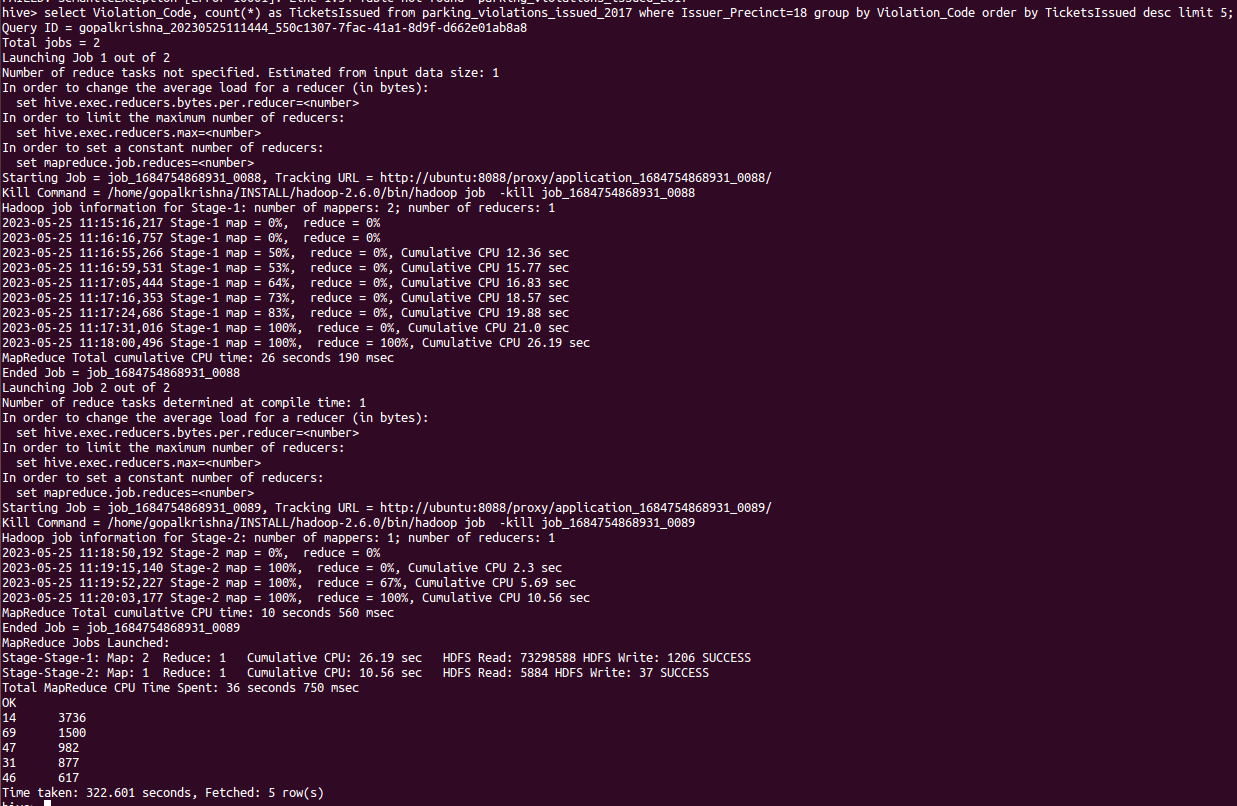
I will increase the limit from 3 to 10, since the result of Issuer\_Precinct showing 0.

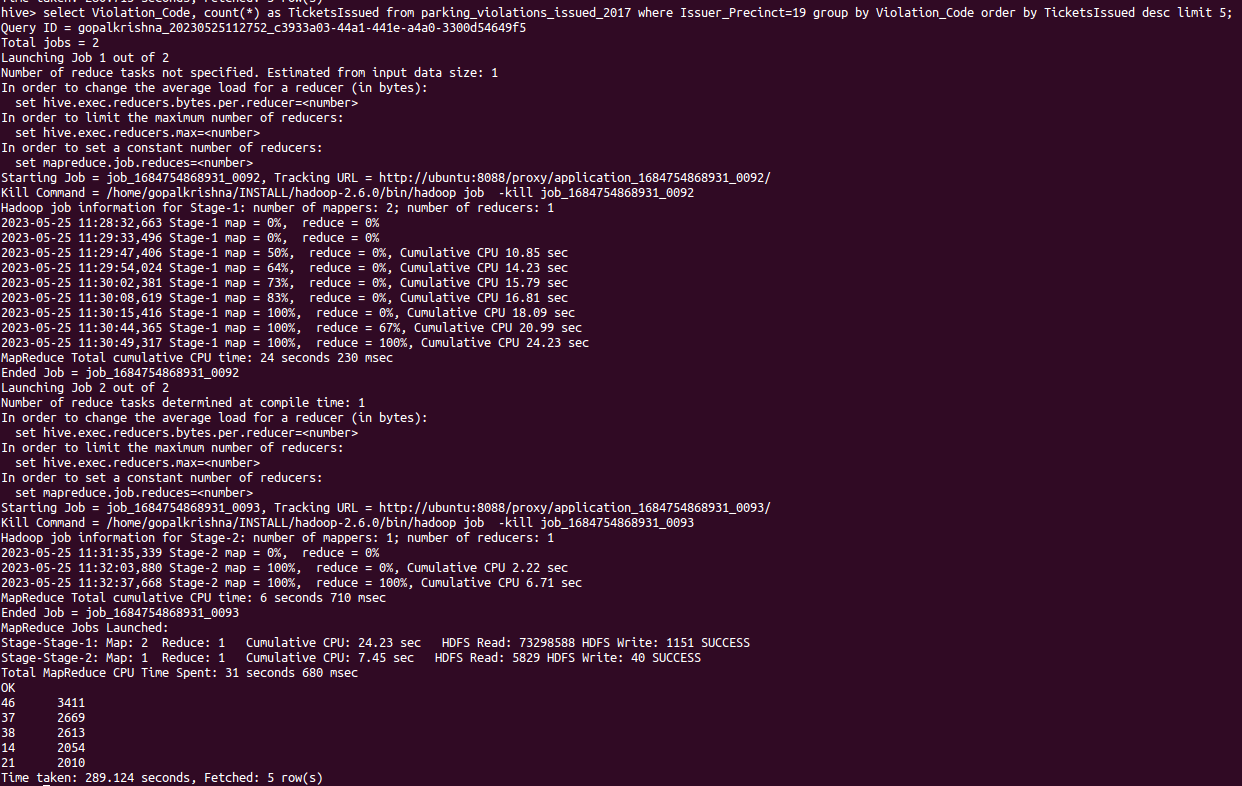
hive> select Issuer\_Precinct,Violation\_Code, count(\*) as TicketsIssued from parking\_violations\_issued\_2017 group by Issuer\_Precinct, Violation\_Code order by TicketsIssued desc limit 10;



Now from the above result, let’s see the analysis result for 14,18,19.



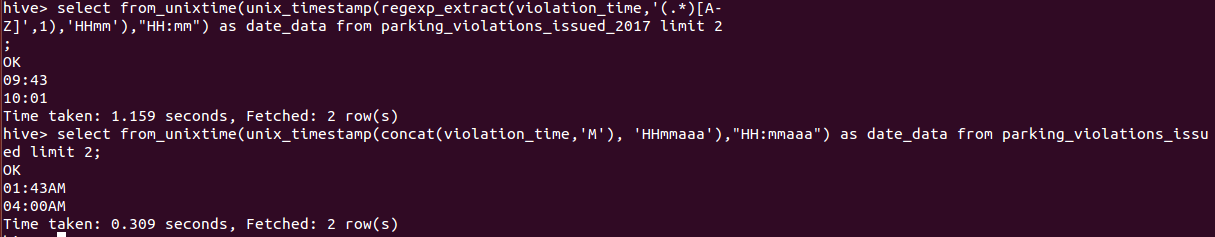




**5.) Find out the properties of parking violations across different times of the day: The Violation Time field is specified in a strange format. Find a way to make this into a time attribute that you can use to divide into groups.**

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

select from\_unixtime(unix\_timestamp(concat(violation\_time,'M'), 'HHmmaaa'),"HH:mmaaa") as date\_data from parking\_violations\_issued limit 2;

****

**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**

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

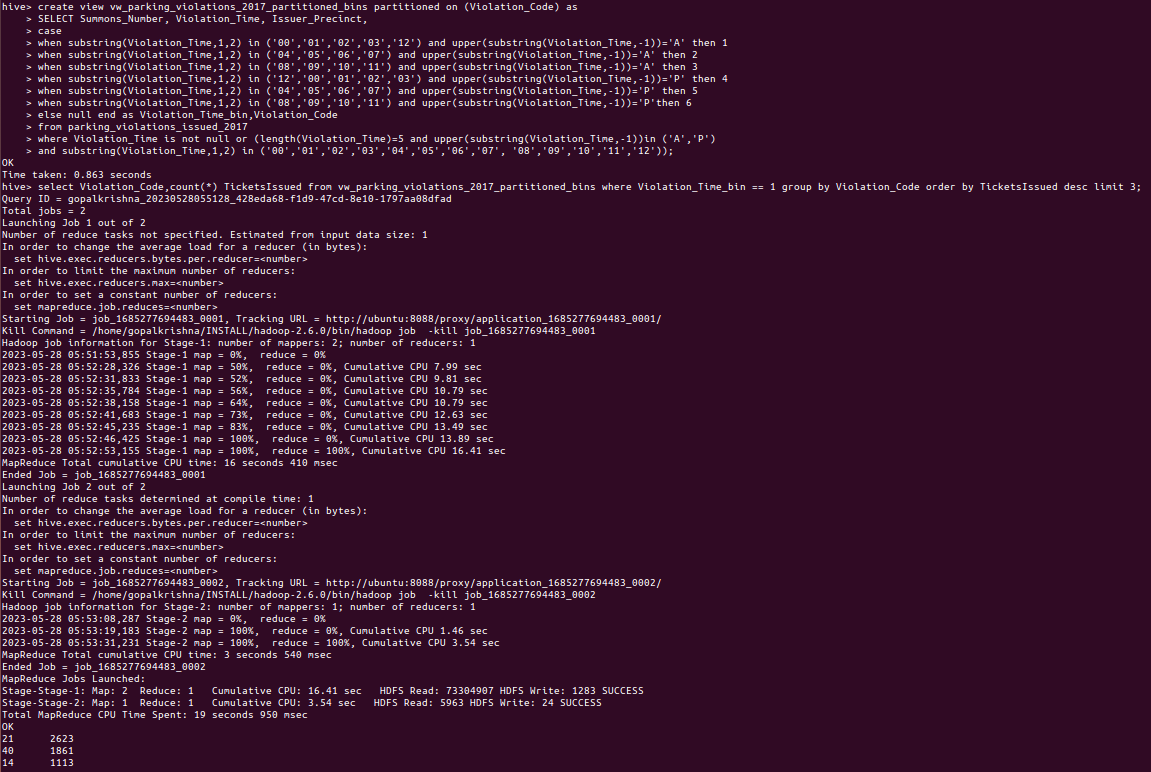
when substring(Violation\_Time,1,2) in ('08','09','10','11') and upper(substring(Violation\_Time,-1))='P'then 6

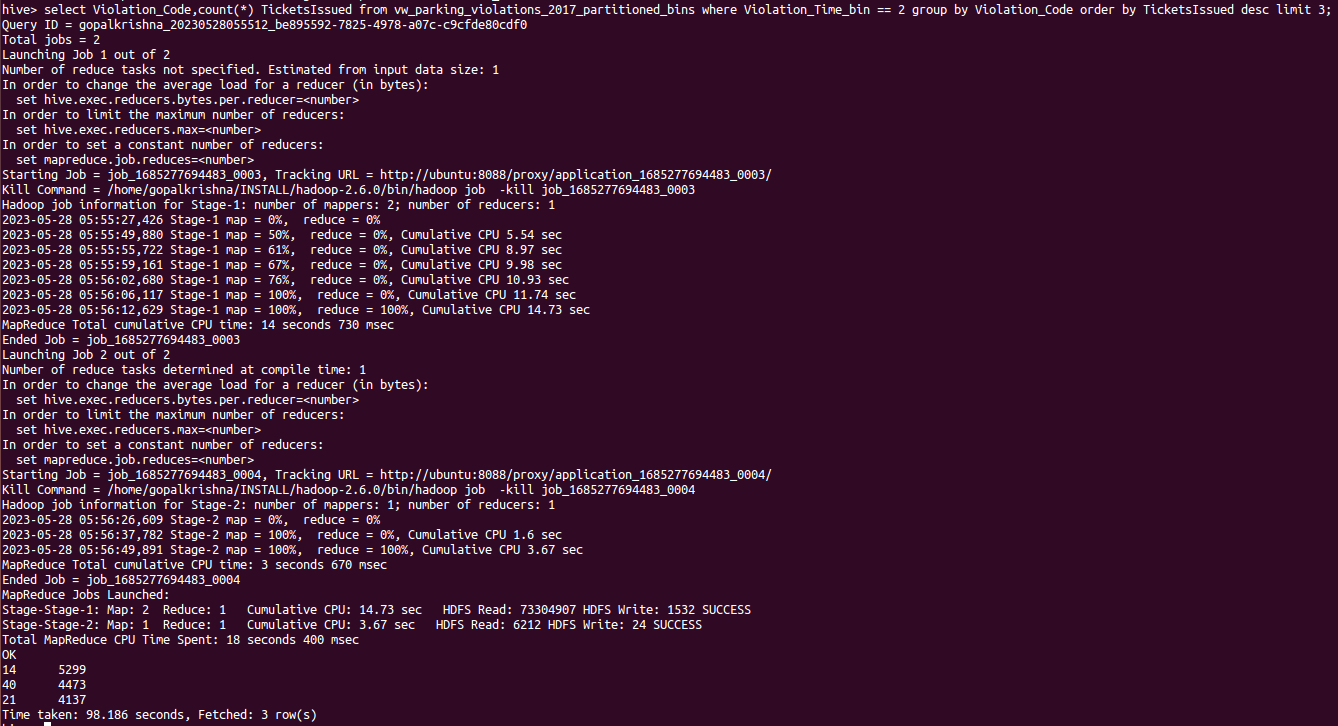
else null end as Violation\_Time\_bin,Violation\_Code

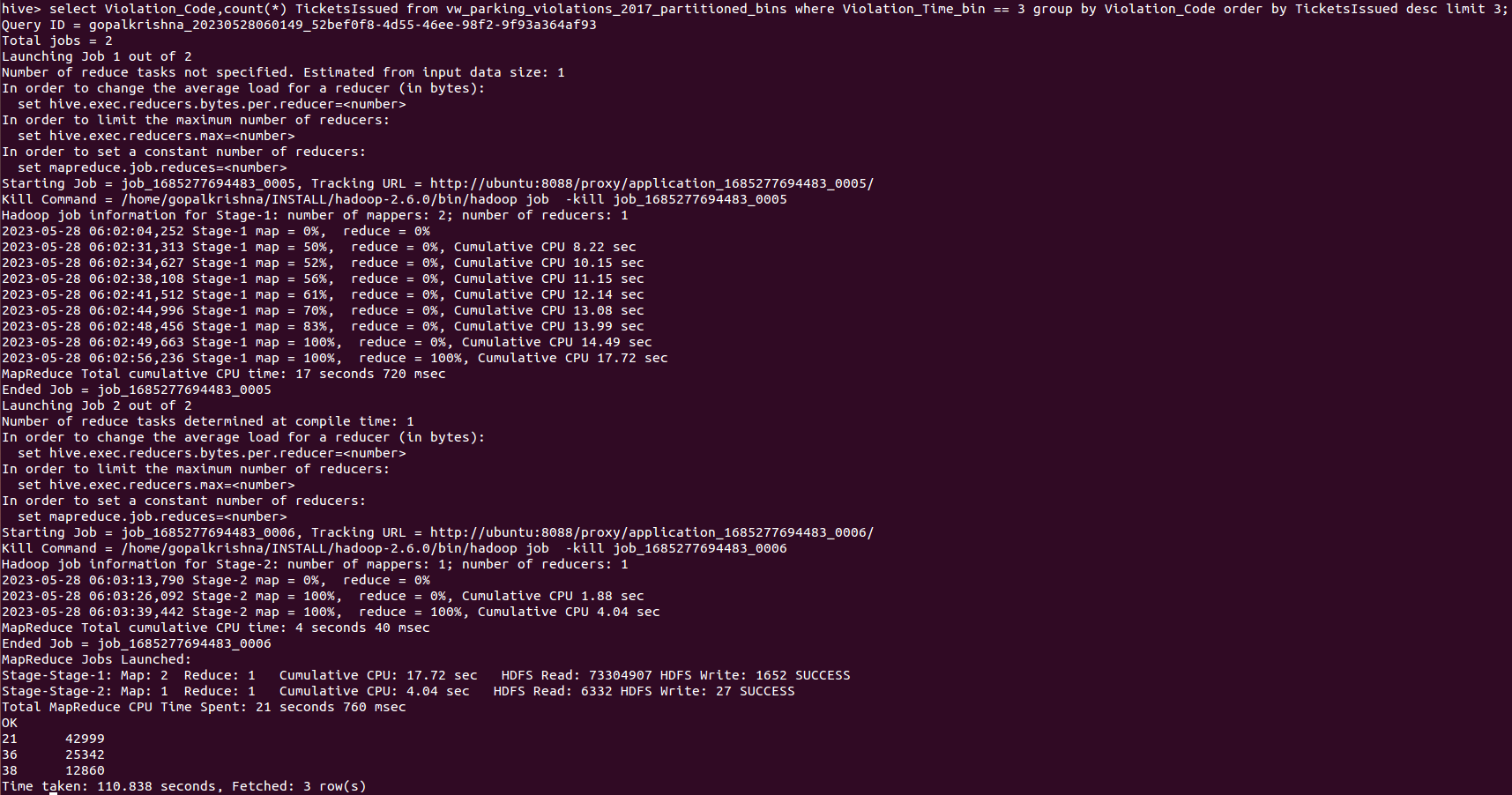
from parking\_violations\_issued\_2017

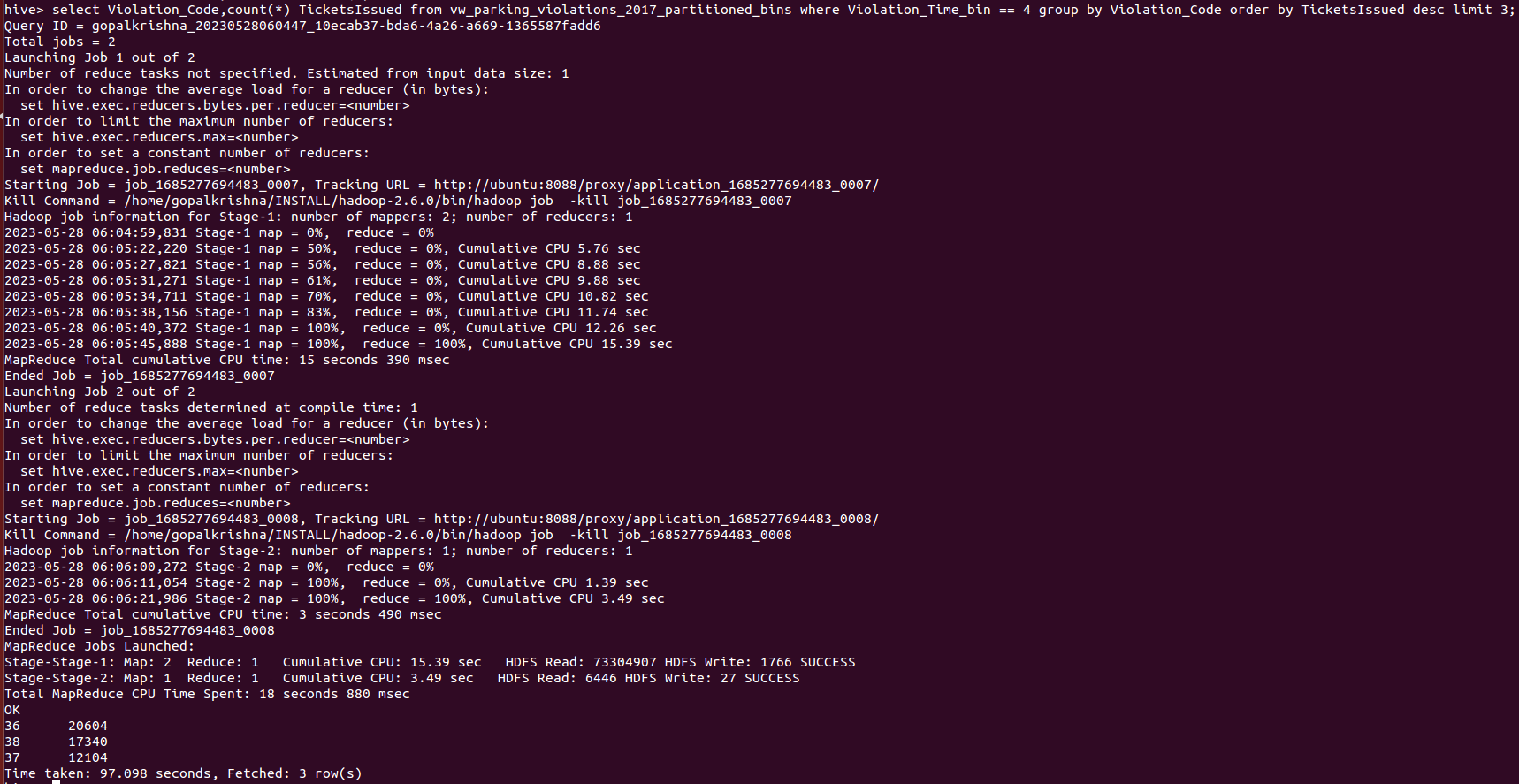
where Violation\_Time is not null or (length(Violation\_Time)=5 and upper(substring(Violation\_Time,-1))in ('A','P')

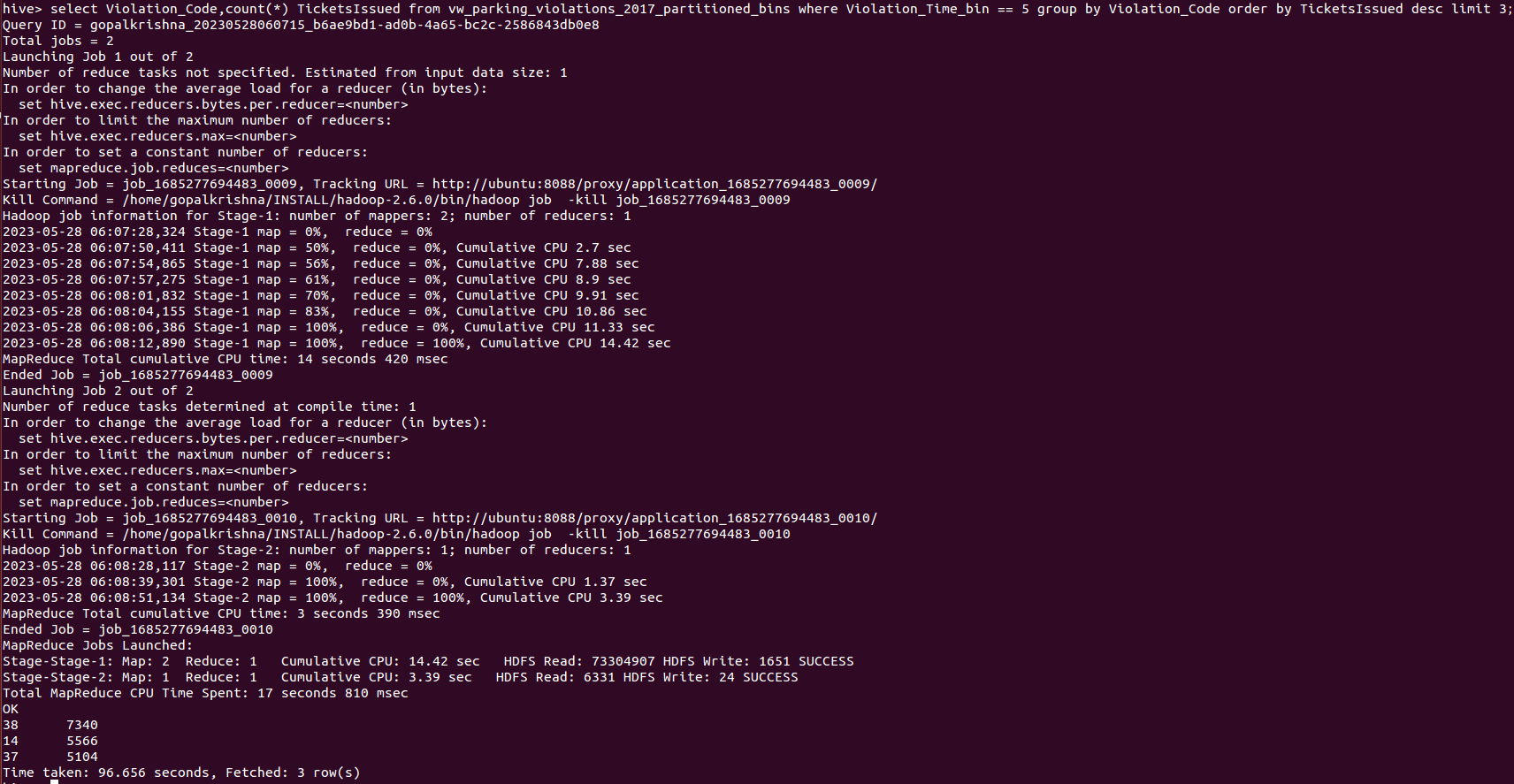
and substring(Violation\_Time,1,2) in ('00','01','02','03','04','05','06','07', '08','09','10','11','12'));

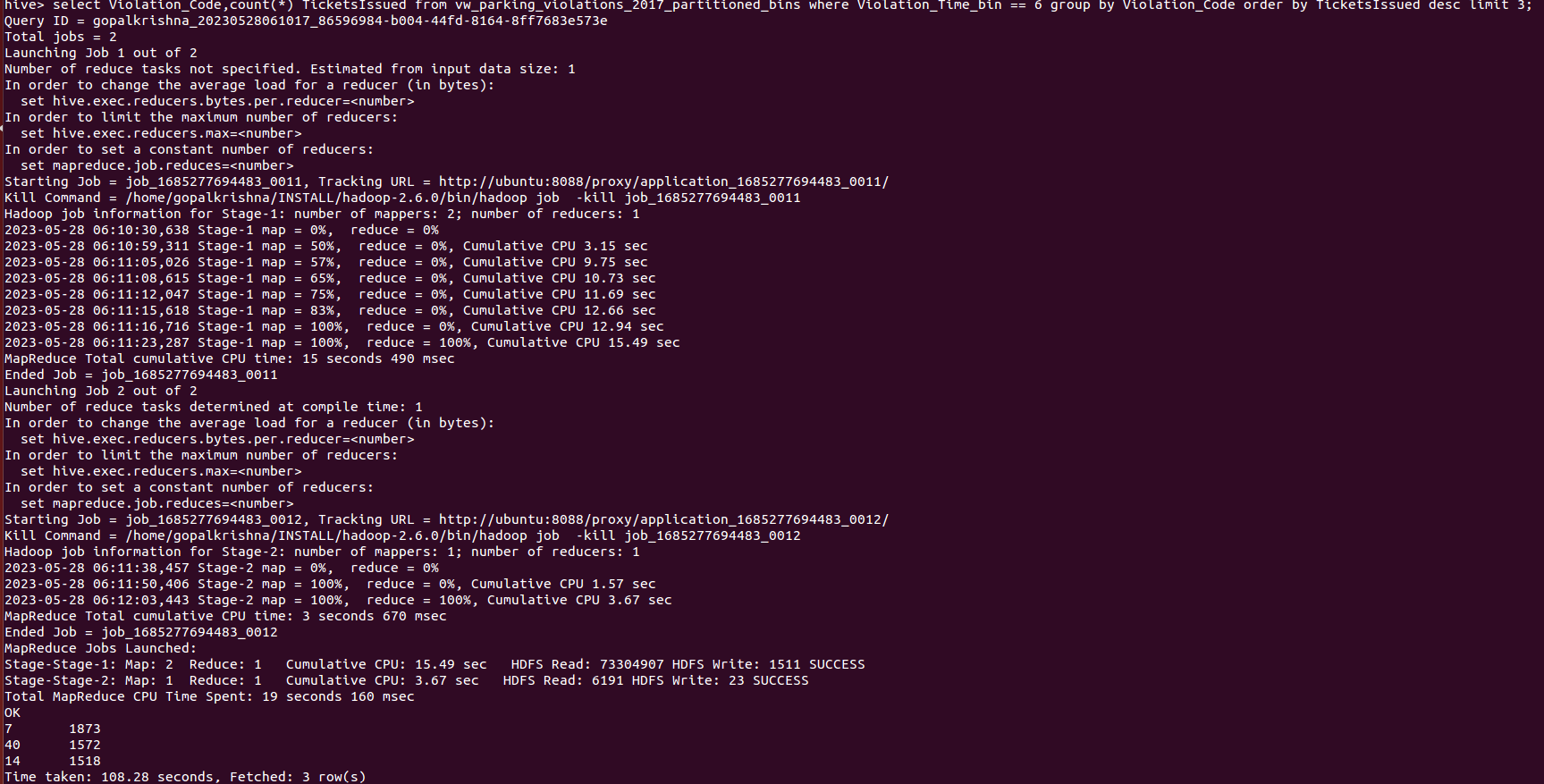
****

****

****

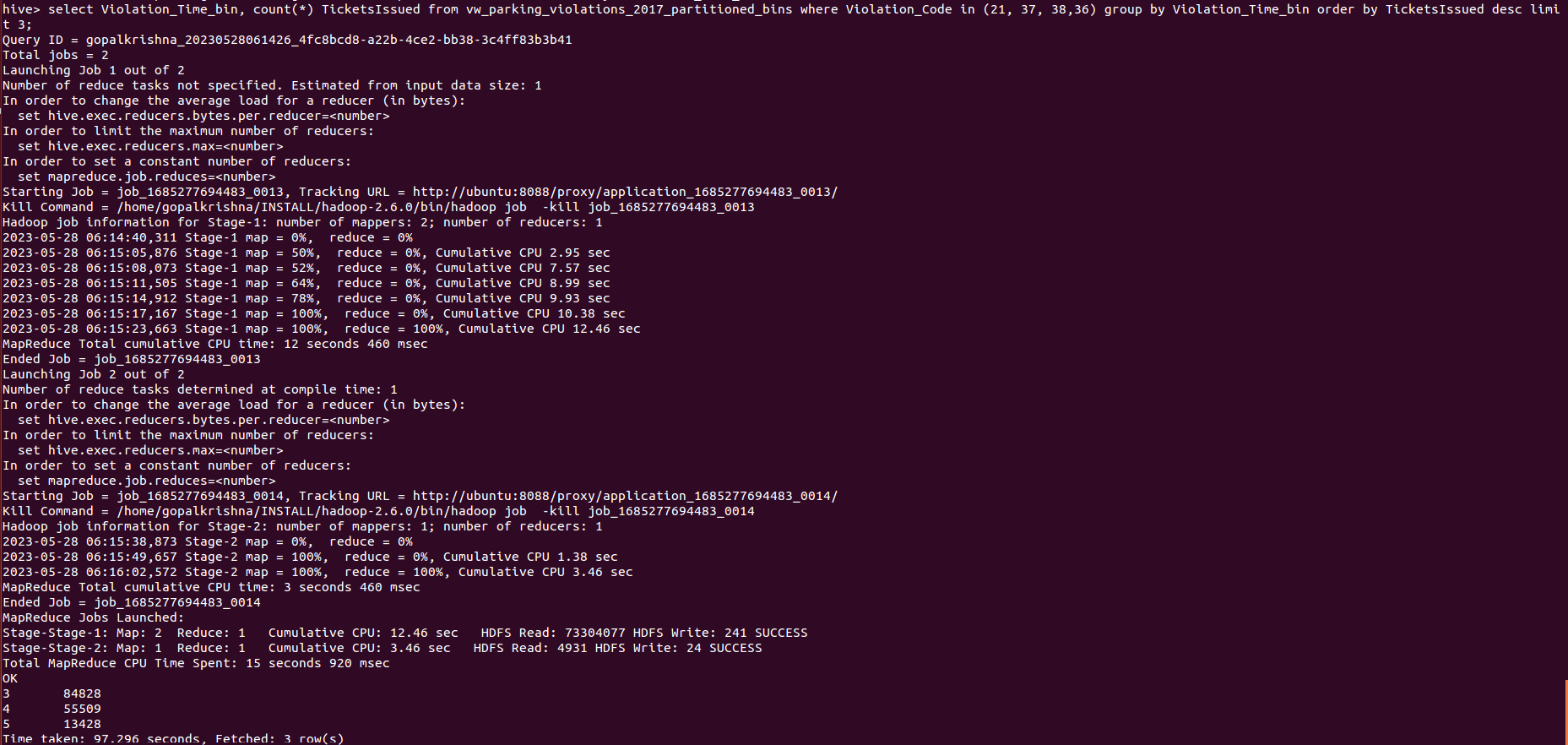
****

****

****

**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)**

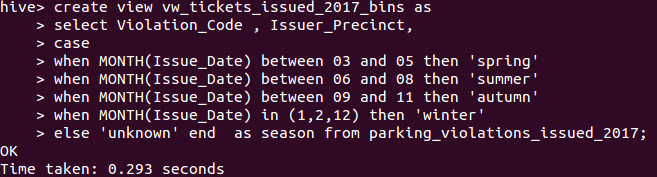
hive> select Violation\_Time\_bin, count(\*) TicketsIssued from vw\_parking\_violations\_2017\_partitioned\_bins where Violation\_Code in (21, 37, 38,36) group by Violation\_Time\_bin order by TicketsIssued desc limit 3;

****

**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))**

**Normal view:-**

****

**Partitioned View:-**

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

select Issuer\_Precinct,

case

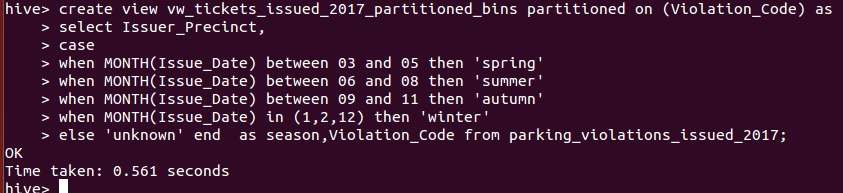
when MONTH(Issue\_Date) between 03 and 05 then 'spring'

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

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

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;

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

**=> Spring:-**

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;

**=> Winter:-**

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;

**=> Summer:-**

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;

**=> Autumn:-**

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;

**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.**