**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.

* **first create an external csv table which points to the data**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 | **create** **external** **table** park\_vio\_ext\_csv ( SUMMONS\_NUMBER int, 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\_CODE\_1 int, STREET\_CODE\_2 int, STREET\_CODE\_3 int, VEHICLE\_EXPIRATION\_DATE int, VIOLATION\_LOCATION string, 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\_INFRONT\_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** ',' tblproperties("skip.header.line.count"="1"); |

* **Load data into park\_vio\_ext\_csv table**

|  |  |
| --- | --- |
| 1 | **load** **data** **local** inpath 'file:///home/cloudera/Desktop/csv\_files/Parking\_Violations\_Issued\_-\_Fiscal\_Year\_2017.csv' **into** **table** parking\_vio\_ext\_csv; |

* **ORC is the optimized file format used in Hive for better analysis.**
* **So, create an external orc table to load the csv data into it.**
* **And also perform partitioning and bucketing on top of it.**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 | **create** **external** **table** park\_vio\_ext\_orc ( SUMMONS\_NUMBER int, 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\_CODE\_1 int, STREET\_CODE\_2 int, STREET\_CODE\_3 int, VEHICLE\_EXPIRATION\_DATE int, VIOLATION\_LOCATION string, VIOLATION\_PRECINCT int, ISSUER\_PRECINCT int, ISSUER\_CODE int, ISSUER\_COMMAND string, ISSUER\_SQUAD string, VIOLATION\_TIME string, TIME\_FIRST\_OBSERVED string, VIOLATION\_INFRONT\_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** (violation\_country string) clustered **by** (violation\_code) sorted **by** (violation\_code) into **5** buckets stored **as** orc; |

* **set some hive properties for dynamic\_partition and bucketing**

|  |  |
| --- | --- |
| 1 2 3 | **set** hive.**exec**.**dynamic**.partition=**true**; set hive.**exec**.**dynamic**.partition.**mode**=nonstrict; set hive.enforce.bucketing=**true**; |

* **overwrite the orc table with csv table**
* **Filter the data of year 2017.**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 | **insert** **into** park\_vio\_ext\_orc 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\_INFRONT\_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\_COUNTRY from park\_vio\_ext\_csv where issue\_date **like** '%2017'; |

**Part-I: Examine the data**

**1.)**

|  |  |
| --- | --- |
| 1 2 3 | /\* total tickets \*/  select **count**(\*) **as** total\_tickets **from** park\_viol\_ext\_orc; |

**2.)**

|  |  |
| --- | --- |
| 1 2 | /\* unique registration states \*/ select **count**(**distinct**(registration\_state)) **as** number\_of\_states **from** park\_vio\_ext\_orc; |

**3.)**

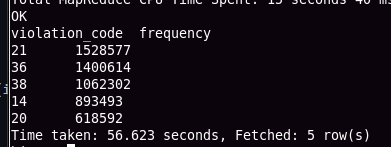
|  |  |
| --- | --- |
| 1 2 3 4 | /\* parking tickets don’t have addresses on them \*/  select **count**(\*) **as** tickets\_not\_having\_address **from** park\_viol\_ext\_orc where street\_code\_1 = **0** **or** street\_code\_2 = **0** **or** street\_code\_3 = **0**; |

**Part-II: Aggregation tasks**

**1.)**

* **frequency of violation codes - find the top 5**

|  |  |
| --- | --- |
| 1 2 3 4 5 6  7 | /\* group the data by violation code to find the frequency of each code \*/  select violation\_code , **count**(\*) **as** frequency **from** park\_vio\_ext\_orc group **by** violation\_code sort **by** frequency **desc** **limit** **5**; |

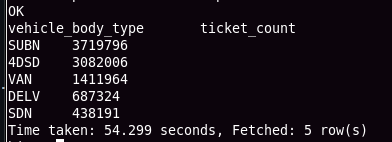


**2.)**

**a.**

* **Frequencies of vehicle body top 5**

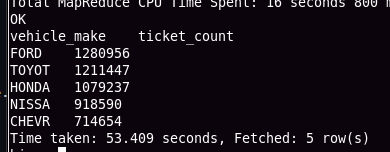
|  |  |
| --- | --- |
| 1 2 3 4 5 6 7 | /\* group the data by vehicle body to find the frequency \*/  select vehicle\_body\_type, **count**(\*) **as** ticket\_count from park\_vio\_ext\_orc group **by** vehicle\_body\_type sort **by** ticket\_count **desc** **limit** **5**; |



**b.**

* **Frequencies of vehicle make top 5**

|  |  |
| --- | --- |
| 1 2 3 4 5 6 7 | /\* group the data by vehicle body to find the frequency \*/  select vehicle\_make, **count**(\*) **as** ticket\_count from park\_vio\_ext\_orc group **by** vehicle\_make sort **by** ticket\_count **desc** **limit** **5**; |

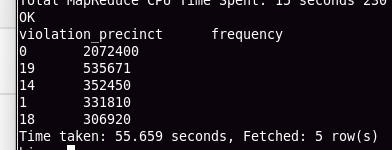


**3.)**

**a.**

* **Frequencies of Violation precincts**

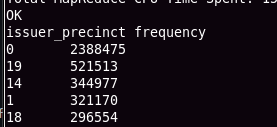
|  |  |
| --- | --- |
| 1 2 3 4 5 6 7 | /\* group the data by violation precinct and find frequencies of each \*/  select violation\_precinct, **count**(\*) **as** frequency from park\_vio\_ext\_orc group **by** violation\_precinct sort **by** frequency **desc** **limit** **5**; |



**b.**

* **Frequencies of issuer precincts**

|  |  |
| --- | --- |
| 1 2 3 4 5 6 7 | /\* group by issuer precinct and find the frequencies of each \*/  select issuer\_precinct, **count**(\*) **as** frequency from park\_vio\_ext\_orc group **by** issuer\_precinct sort **by** frequency **desc** **limit** **5**; |

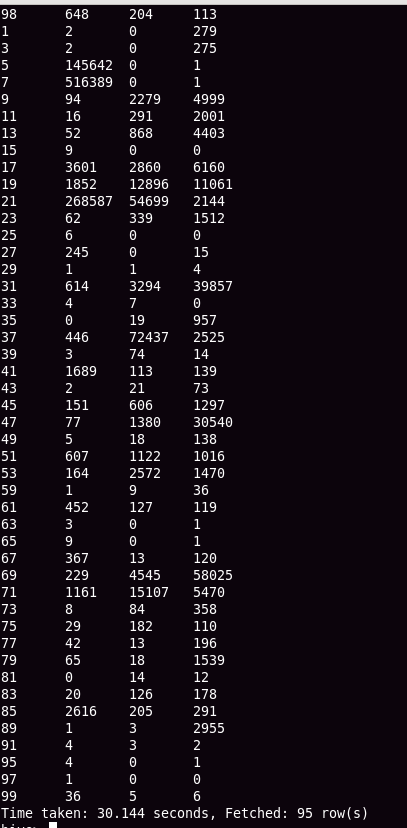
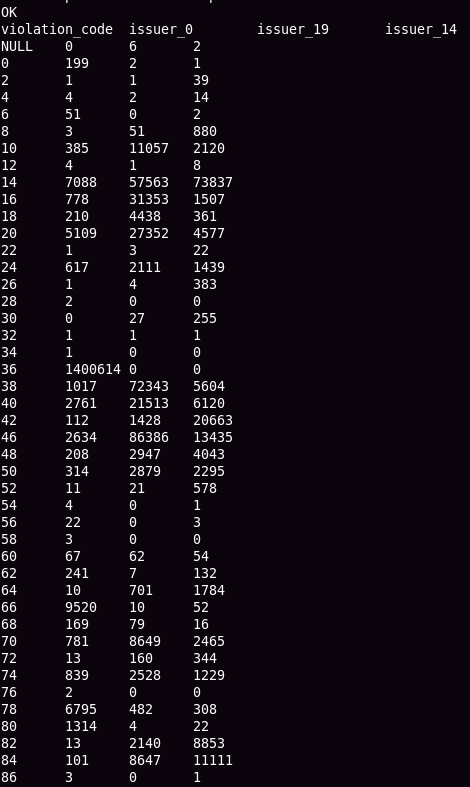


**4.)**

* **Find the violation code frequency across 3 precincts which have issued the most number of tickets**

|  |  |
| --- | --- |
| 1 2 3 4 5 6 7 | /\* find the top 3 issuer\_precinct \*/  select issuer\_precinct,**count**(\*) **as** frequency from park\_vio\_ext\_orc group **by** issuer\_precinct sort **by** frequency **desc** **limit** **3**; |

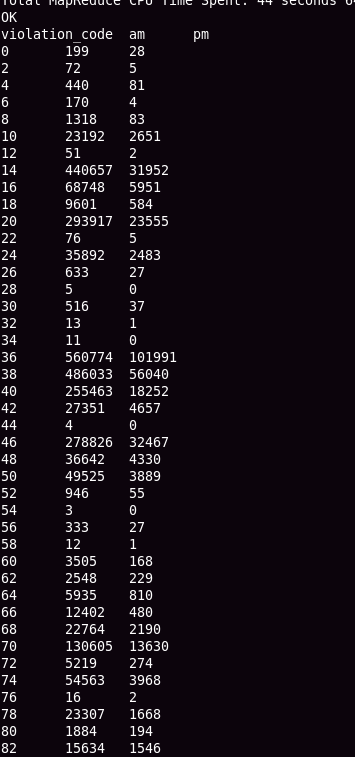
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 10 | /\* from the above queryissuer\_precincts are 0, 19,14. Based on the result perform group by violation\_code \*/  select violation\_code, sum(**case** **when** issuer\_precinct = **0** **then** **1** **else** **0** **end**) **as** issuer\_0, sum(**case** **when** issuer\_precinct = **19** **then** **1** **else** **0** **end**) **as** issuer\_19, sum(**case** **when** issuer\_precinct = **14** **then** **1** **else** **0** **end**) **as** issuer\_14 from park\_vio\_ext\_orc a where a.issuer\_precinct **in** (**0**,**19**,**14**) group **by** violation\_code; |



**5.)**

* **Divide time based on AM and PM**
* **Find the frequencies for them.**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 | /\*  1.) filter data where violation\_time has A or P at the end. 2.) And consider the data where hour in violation\_time is from 1 - 12 3.) Append the date time and convert them into unix\_timestamp. 4.) Now perform group by on violation\_code which results the count of each violation\_code in AM and PM  \*/  with data\_2017 **as** ( select registration\_state, issue\_date, violation\_code, violation\_precinct,  issuer\_precinct, violation\_time, violation\_country from park\_vio\_ext\_orc  where (violation\_time **like** '%P' **or** violation\_time **like** '%A') **and**  (**cast**(**substring**(violation\_time,**1**,**2**) **as** int) > **0** **and**  cast(**substring**(violation\_time,**1**,**2**) **as** int) <=**12**) ), change\_time **as** ( select registration\_state,issue\_date, violation\_code, violation\_precinct, issuer\_precinct,  concat(**substring**(violation\_time,**1**,**2**),':',**substring**(violation\_time,**3**,**2**),':00 ',**substring**(violation\_time,**5**,**1**),'M') **as** violation\_time, violation\_country **from** data\_2017 ) select violation\_code, sum(**case** **when** violation\_time **like** '%AM' **then** **1** **else** **0** **end**) **as** AM, sum(**case** **when** violation\_time **like** '%PM' **then** **1** **else** **0** **end**) **as** PM from change\_date group **by** violation\_code; |



**6.)**

* **create external table to store only required fileds from main data after applying all the filtering.**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 10 | **create** **external** **table** filtered\_data\_orc  ( registration\_state string, date\_time string, violation\_code int, violation\_precinct int, issuer\_precinct int, violation\_country string ) stored **as** orc; |

* **The below code is written with the help of common table expressions.**
* **1.) filter the data**
* **2.) convert the date in string format to unix\_timestamp and time to hh:mm:ss format**
* **3.) combine the time and date and convert both to unix\_time**
* **4.) insert the final modified data into filtered\_data\_orc table**

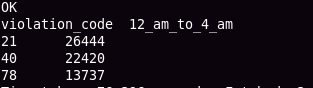
|  |
| --- |
| **with** data\_2017 **as** ( select registration\_state, issue\_date, violation\_code, violation\_precinct, issuer\_precinct, violation\_time, violation\_country from park\_vio\_ext\_orc  where (violation\_time **like** '%P' **or** violation\_time **like** '%A') **and**  (**cast**(**substring**(violation\_time,**1**,**2**) **as** int) > **0** **and** **cast**(**substring**(violation\_time,**1**,**2**) **as** int) <=**12**) ), change\_date **as** ( select registration\_state, from\_unixtime(unix\_timestamp(issue\_date,'MM/dd/yyyy'),'yyyy-MM-dd') **as** date, violation\_code, violation\_precinct, issuer\_precinct,  concat(**substring**(violation\_time,**1**,**2**),':',**substring**(violation\_time,**3**,**2**),':00 ',**substring**(violation\_time,**5**,**1**),'M') **as** violation\_time,  violation\_country  from data\_2017 ),  combine\_date\_time **as** ( select registration\_state, from\_unixtime(unix\_timestamp(concat(date,violation\_time),'yyyy-MM-ddhh:mm:ss a'),'yyyy-MM-dd hh:mm:ss a') **as** date\_time, violation\_code, violation\_precinct, i ssuer\_precinct,violation\_country  from change\_date ) from combine\_date\_time  insert overwrite **table** filtered\_data\_orc **select** \*; |

* **Now from the filtered\_data\_orc table group the hours into six groups.**
* **Find the frequency of each violation\_code in that particular interval group**

|  |
| --- |
| **with** time\_groups **as** ( select violation\_code , sum(**case** **when** (date\_time **like** '%AM' **and** hour(date\_time) **in** (**12**,**1**,**2**,**3**)) **then** **1** **else** **0** **end**) **as** **12**\_AM\_to\_4\_AM, sum(**case** **when** (date\_time **like** '%AM' **and** hour(date\_time) **in** (**4**,**5**,**6**,**7**)) **then** **1** **else** **0** **end**) **as** **4**\_AM\_to\_8\_AM, sum(**case** **when** (date\_time **like** '%AM' **and** hour(date\_time) **in** (**8**,**9**,**10**,**11**)) **then** **1** **else** **0** **end**) **as** **8**\_AM\_to\_12\_PM, sum(**case** **when** (date\_time **like** '%PM' **and** hour(date\_time) **in** (**12**,**1**,**2**,**3**)) **then** **1** **else** **0** **end**) **as** **12**\_PM\_to\_4\_PM, sum(**case** **when** (date\_time **like** '%PM' **and** hour(date\_time) **in** (**4**,**5**,**6**,**7**)) **then** **1** **else** **0** **end**) **as** **4**\_PM\_to\_8\_PM, sum(**case** **when** (date\_time **like** '%PM' **and** hour(date\_time) **in** (**8**,**9**,**10**,**11**)) **then** **1** **else** **0** **end**) **as** **8**\_PM\_to\_12\_PM from filtered\_data\_orc group **by** violation\_code ) select violation\_code, **12**\_AM\_to\_4\_AM  from time\_groups sort **by** **12**\_AM\_to\_4\_AM **desc** **limit** **3**; |

* **Similarly for all the time intervals**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 | /\* similarly for all other groups \*/  select violation\_code, **4**\_AM\_to\_8\_AM  from time\_groups sort **by** **4**\_AM\_to\_8\_AM **desc** **limit** **3**;   select violation\_code, **8**\_AM\_to\_12\_PM  from time\_groups sort **by** **8**\_AM\_to\_12\_PM **desc** **limit** **3**;   select violation\_code, **12**\_PM\_to\_4\_PM  from time\_groups sort **by** **12**\_PM\_to\_4\_PM **desc** **limit** **3**;   select violation\_code, **4**\_PM\_to\_8\_PM  from time\_groups sort **by** **4**\_PM\_to\_8\_PM **desc** **limit** **3**;   select violation\_code, **8**\_PM\_to\_12\_PM from time\_groups sort **by** **8**\_PM\_to\_12\_PM **desc** **limit** **3**; |



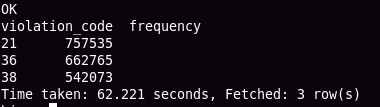
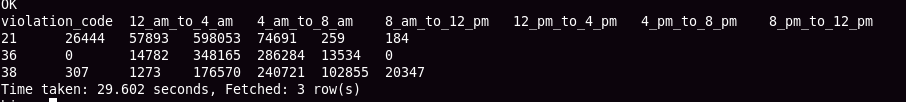
**7.)**

* **first find the 3 most commonly occurring violation codes**

|  |  |
| --- | --- |
| 1 2 3 4 5 | **select** violation\_code, **count**(\*) **as** frequency from filtered\_data\_orc group **by** violation\_code sort **by** frequency **desc** **limit** **3;** |

* **So, the top 3 violation\_codes are 21, 36, 38**
* **And the frequencies of these top 3 violation\_code in all the intervals**

|  |
| --- |
| **select** violation\_code, sum(**case** **when** (date\_time **like** '%AM' **and** hour(date\_time) **in** (**12**,**1**,**2**,**3**)) **then** **1** **else** **0** **end**) **as** **12**\_AM\_to\_4\_AM, sum(**case** **when** (date\_time **like** '%AM' **and** hour(date\_time) **in** (**4**,**5**,**6**,**7**)) **then** **1** **else** **0** **end**) **as** **4**\_AM\_to\_8\_AM, sum(**case** **when** (date\_time **like** '%AM' **and** hour(date\_time) **in** (**8**,**9**,**10**,**11**)) **then** **1** **else** **0** **end**) **as** **8**\_AM\_to\_12\_PM, sum(**case** **when** (date\_time **like** '%PM' **and** hour(date\_time) **in** (**12**,**1**,**2**,**3**)) **then** **1** **else** **0** **end**) **as** **12**\_PM\_to\_4\_PM, sum(**case** **when** (date\_time **like** '%PM' **and** hour(date\_time) **in** (**4**,**5**,**6**,**7**)) **then** **1** **else** **0** **end**) **as** **4**\_PM\_to\_8\_PM, sum(**case** **when** (date\_time **like** '%PM' **and** hour(date\_time) **in** (**8**,**9**,**10**,**11**)) **then** **1** **else** **0** **end**) **as** **8**\_PM\_to\_12\_PM from filtered\_data\_orc  where violation\_code **in** (**21**,**36**,**38**) group **by** violation\_code; |



**8)**

**a.)**

* **frequency of tickets for each of the seasons**

|  |
| --- |
| **select** **sum**(**case** **when** **month**(date\_time) **in** (**3**,**4**,**5**) **then** **1** **else** **0** **end**) **as** spring, sum(**case** **when** **month**(date\_time) **in** (**6**,**7**,**8**) **then** **1** **else** **0** **end**) **as** summer, sum(**case** **when** **month**(date\_time) **in** (**9**,**10**,**11**) **then** **1** **else** **0** **end**) **as** fall, sum(**case** **when** **month**(date\_time) **in** (**12**,**1**,**2**) **then** **1** **else** **0** **end**) **as** winter from filtered\_data\_orc; |

**b.)**

* **3 most common violations in these seasons**

|  |
| --- |
| **with** cte **as** ( select violation\_code, sum(**case** **when** **month**(date\_time) **in** (**3**,**4**,**5**) **then** **1** **else** **0** **end**) **as** spring, sum(**case** **when** **month**(date\_time) **in** (**6**,**7**,**8**) **then** **1** **else** **0** **end**) **as** summer, sum(**case** **when** **month**(date\_time) **in** (**9**,**10**,**11**) **then** **1** **else** **0** **end**) **as** fall, sum(**case** **when** **month**(date\_time) **in** (**12**,**1**,**2**) **then** **1** **else** **0** **end**) **as** winter from filtered\_data\_orc group **by** violation\_code ) /\* This selects for spring season \*/ select violation\_code,spring from cte order **by** spring **desc** **limit** **3**; |

* **Similarly for all the seasons**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 | /\* similarly for all the seasons \*/  select violation\_code,summer from cte order **by** summer **desc** **limit** **3**;  select violation\_code,fall from cte order **by** fall **desc** **limit** **3**;   select violation\_code,winter from cte order **by** winter **desc** **limit** **3**; |

