THE UNIVERSITY OF ARIZONA

UpGrad

Hive Assignment

Submission by:

Merrymel George (merrymelgeorge@gmail.com)

THE UNIVERSITY OF ARIZONA

UpGrad

Steps for setting up

Step 1 Downloading the data from the public S3 bucket to EMR Cluster

Code

```
wget https://hiveassignmentdatabde.s3.amazonaws.com/Parking_Violations_Issued_-_Fiscal_Year_2017.csv;
```

```
[hadoop@ip-172-31-27-89 ~]$ wget https://hiveassignmentdatabde.s3.amazonaws.com/Parking_Violations_Issued_-_Fiscal_Year_2017.csv;
--2022-10-25 03:09:11--  https://hiveassignmentdatabde.s3.amazonaws.com/Parking_Violations_Issued_-_Fiscal_Year_2017.csv
Resolving hiveassignmentdatabde.s3.amazonaws.com (hiveassignmentdatabde.s3.amazonaws.com)... 3.5.3.112
Connecting to hiveassignmentdatabde.s3.amazonaws.com (hiveassignmentdatabde.s3.amazonaws.com)|3.5.3.112|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2086913576 (1.9G) [text/csv]
Saving to: 'Parking_Violations_Issued_-_Fiscal_Year_2017.csv'

100%[=======================================================================================================================================================>] 2,086,913,576 39.9MB/s   in 55s

2022-10-25 03:10:07 (35.9 MB/s) - 'Parking_Violations_Issued_-_Fiscal_Year_2017.csv' saved [2086913576/2086913576]

[hadoop@ip-172-31-27-89 ~]$
```

Step 2 Launch Hive

Code

```
hive
```

```
[hadoop@ip-172-31-27-89 ~]$ hive

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.properties Async: false
hive>
```

THE UNIVERSITY OF ARIZONA

UpGrad

Steps for setting up

Step 3

Creating the database, using it, creating an external table and loading data into the table

Code

```
create database if not exists assignment;

use assignment;

create external table if not exists parking_violations (summons_number bigint, plate_id varchar(255),
registration_state varchar(255), plate_type varchar(255), issue_date varchar(255), violation_code
bigint, vehicle_body_type varchar(255), vehicle_make varchar(255), issuing_agency varchar(255),
street_code1 bigint, street_code2 bigint, street_code3 bigint, vehicle_expiration_date bigint,
violation_location varchar(255), violation_precinct bigint, issuer_precinct bigint, issuer_code
bigint, issuer_command varchar(255), issuer_squad varchar(255), violation_time varchar(255),
time_first_observed varchar(255), violation_county varchar(255), violation_in_front_of_or_opposite
varchar(255), house_number varchar(255), street_name varchar(255), intersecting_street varchar(255),
date_first_observed bigint, law_section bigint, sub_division varchar(255), violation_legal_code
varchar(255), days_parking_in_effect varchar(255), from_hours_in_effect varchar(255),
to_hours_in_effect varchar(255), vehicle_color varchar(255), unregistered_vehicle varchar(255),
vehicle_year bigint, meter_number varchar(255), feet_from_curb bigint, violation_post_code
varchar(255), violation_description varchar(255), no_standing_or_stopping_violation varchar(255),
hydrant_violation varchar(255), double_parking_violation varchar(255)) row format delimited fields
terminated by ',' lines terminated by '\n' stored as textfile
tblproperties("skip.header.line.count"="1");

load data local inpath '/home/hadoop/Parking_Violations_Issued_-_Fiscal_Year_2017.csv' into table
parking_violations;
```

THE UNIVERSITY OF ARIZONA

UpGrad

Steps for setting up

```
hive> create database if not exists assignment;
OK
Time taken: 0.698 seconds
hive> use assignment;
OK
Time taken: 0.029 seconds
hive> create external table if not exists parking_violations (summons_number bigint, plate_id varchar(255), registration_state varchar(255), plate_type varchar(255), issue_date varchar(255)
, violation_code bigint, vehicle_body_type varchar(255), vehicle_make varchar(255), issuing_agency varchar(255), street_code1 bigint, street_code2 bigint, street_code3 bigint, vehicle_expir
ation_date bigint, violation_location varchar(255), violation_precinct bigint, issuer_precinct bigint, issuer_code bigint, issuer_command varchar(255), issuer_squad varchar(255), violation_
time varchar(255), time_first_observed varchar(255), violation_county varchar(255), violation_in_front_of_or_opposite varchar(255), house_number varchar(255), street_name varchar(255), inte
rsecting_street varchar(255), date_first_observed bigint, law_section bigint, sub_division varchar(255), violation_legal_code varchar(255), days_parking_in_effect varchar(255), from_hours_i
n_effect varchar(255), to_hours_in_effect varchar(255), vehicle_color varchar(255), unregistered_vehicle varchar(255), vehicle_year bigint, meter_number varchar(255), feet_from_curb bigint,
 violation_post_code varchar(255), violation_description varchar(255), no_standing_or_stopping_violation varchar(255), hydrant_violation varchar(255), double_parking_violation varchar(255))
 row format delimited fields terminated by ',' lines terminated by '\n' stored as textfile tblproperties("skip.header.line.count"="1");
OK
Time taken: 0.521 seconds
hive> load data local inpath '/home/hadoop/Parking_Violations_Issued_-_Fiscal_Year_2017.csv' into table parking_violations;
Loading data to table assignment.parking_violations
OK
Time taken: 19.352 seconds
```

Step 4

Creating separate table only for 2017 data

Code

```
create table if not exists parking_violations_2017 as select * from parking_violations where
substr(issue_date,7,4) = '2017';
```

```
hive> create table if not exists parking_violations_2017 as select * from parking_violations where substr(issue_date,7,4) = '2017';
Query ID = hadoop_20221025035800_cc58d7c4-f1a6-40f8-8e10-c3e1d0872cff
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1666667007666_0006)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 01/01  [==========================>>] 100%  ELAPSED TIME: 92.43 s
--------------------------------------------------------------------------------
Moving data to directory hdfs://ip-172-31-27-89.ec2.internal:8020/user/hive/warehouse/assignment.db/parking_violations_2017
OK
Time taken: 93.506 seconds
```

THE UNIVERSITY OF ARIZONA

UpGrad

Question No: 1.1

Find the total number of tickets for the year.

Answer

5431903 tickets in 2017

Code

```
SELECT COUNT(summons_number) FROM parking_violations_2017;
SELECT COUNT(DISTINCT summons_number) FROM parking_violations_2017;
```

```
hive> SELECT COUNT(summons_number) FROM parking_violations_2017;
Query ID = hadoop_20221025040059_41b5b9f0-eba6-43c5-8710-ebc8fd4cce7a
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1666667007666_0006)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED     10         10        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 26.69 s
--------------------------------------------------------------------------------
OK
5431903
Time taken: 27.183 seconds, Fetched: 1 row(s)
hive> SELECT COUNT(distinct summons_number) FROM parking_violations_2017;
Query ID = hadoop_20221025040146_dc8198ad-2f85-456e-a53c-e9e393c22618
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1666667007666_0006)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED     10         10        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 40.01 s
--------------------------------------------------------------------------------
OK
5431903
Time taken: 40.543 seconds, Fetched: 1 row(s)
```

THE UNIVERSITY OF ARIZONA

UpGrad

Question No: 1.2

Find out the total number of states to which the cars with tickets belong.

Answer

65 distinct registration states

Code

```
SELECT COUNT(DISTINCT registration_state) FROM parking_violations_2017;
SELECT COUNT(DISTINCT registration_state) FROM parking_violations_2017 WHERE
registration_state IS NOT NULL;
```

```
hive> SELECT COUNT(DISTINCT registration_state) FROM parking_violations_2017;
Query ID = hadoop_20221025040319_d665a6e1-a3c2-4053-8e7e-af5298235cce
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1666667007666_0006)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED     10         10        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 30.68 s
--------------------------------------------------------------------------------
OK
65
Time taken: 31.202 seconds, Fetched: 1 row(s)
hive> SELECT COUNT(DISTINCT registration_state) FROM parking_violations_2017 where registration_state IS NOT NULL;
Query ID = hadoop_20221025040428_771ced69-439d-44a5-b2d9-0754e5b7e9d7
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1666667007666_0006)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED     10         10        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 29.39 s
--------------------------------------------------------------------------------
OK
65
Time taken: 29.91 seconds, Fetched: 1 row(s)
```

THE UNIVERSITY OF ARIZONA

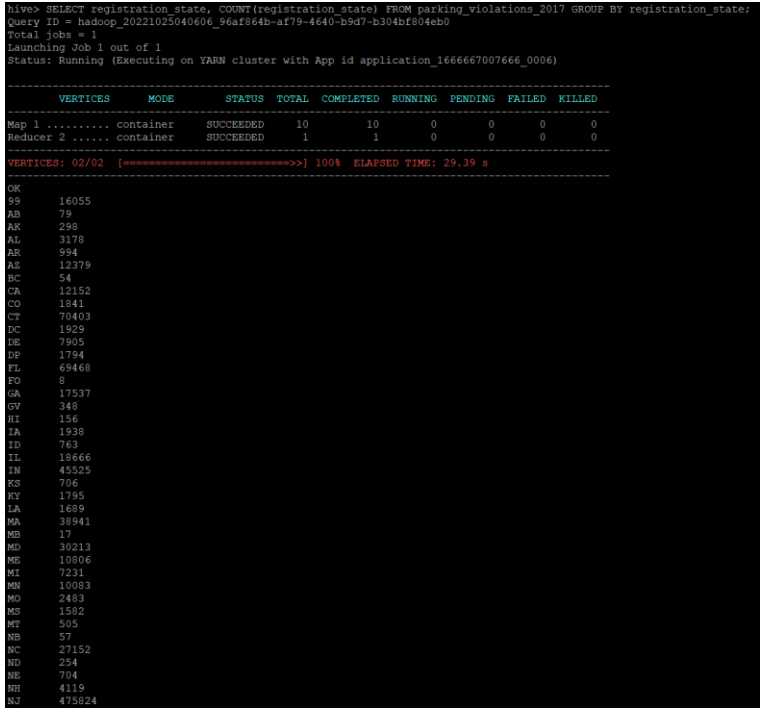
UpGrad

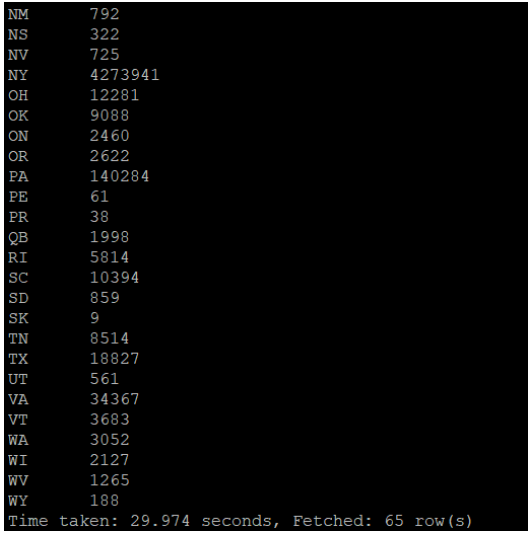
Question No: 1.2 - Optional

Exact list of states with counts

Code

```
SELECT registration_state, COUNT(registration_state) FROM parking_violations_2017 GROUP BY registration_state;
```





THE UNIVERSITY OF ARIZONA

UpGrad

Question No: 1.3

Find out the number of such tickets which have no addresses.

Answer

1816816 tickets in the year 2017 doesn't have a full valid address

Code

```
SELECT COUNT(summons_number) FROM parking_violations_2017 WHERE street_code1 IS NULL OR
street_code2 IS NULL OR street_code3 IS NULL OR street_code1=0 OR street_code2=0 OR
street_code3=0;
```

```
hive> SELECT COUNT(summons_number) FROM parking_violations_2017 WHERE street_code1 IS NULL OR street_code2 IS NULL OR street_code3 IS NULL OR street_code1=0 OR street_code2=0 OR street_code
3=0;
Query ID = hadoop_20221025041050_d493d6b2-7d0d-4a39-853b-5c204235a19b
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1666667007666_0006)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED     10         10        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 28.42 s
--------------------------------------------------------------------------------
OK
1816816
Time taken: 28.902 seconds, Fetched: 1 row(s)
```

THE UNIVERSITY OF ARIZONA

UpGrad

Steps before Part II – Aggregation tasks

Step 1

Creating a new table named parking_violations_2017_part2 which has cleansed, properly date formatted, required and feature-engineered attributes to solve the Part II questions

Logic

1. Selection of relevant attributes like summons_number, violation_code, violation_description, violation_time, issue_date
2. Tagging of violation_time values which don't have 'A' or 'P', which have '+' or '.', which is blank or NULL as 'INVALID
3. Formating the violation_time values into 24-hour format based on various conditional checks
4. Converting issue_date and formatted violation time into timestamp datatype
5. Feature engineering violation_time_bucket by creating 6 buckets of time slots:
   a. Morning Slot – 06:00 AM to 10:00 AM
   b. Mid Day Slot – 10:00 AM to 02:00 PM
   c. Afternoon Slot – 02:00 PM to 06:00 PM
   d. Evening Slot – 06:00 PM to 10:00 PM
   e. Night Slot – 10:00 PM to 02:00 AM
   f. Early Morning Slot – 02:00 AM to 06:00 AM
6. Feature engineering issue_date_season extracting month from the formatted violation timestamp and using conditional checks to tag to the corresponding season:
   a. Months of December (12), January (1) and February (2) as Winter
   b. Months of March (3), April (4) and May (5) as Spring
   c. Months of June (6), July (7) and August (8) as Summer
   d. Months of September (9), October (10) and November (11) as Fall

THE UNIVERSITY OF ARIZONA

UpGrad

Steps before Part II – Aggregation tasks

Code

```
CREATE TABLE IF NOT EXISTS parking_violations_2017_part2 AS SELECT A.summons_number, A.violation_code, A.violation_description,
A.violation_time, A.violation_time_formatted, from_unixtime(unix_timestamp(concat(A.issue_date,' ',A.violation_time_formatted),
'MM/dd/yyyy hh:mm')) as violation_timestamp_formatted, CASE WHEN HOUR(from_unixtime(unix_timestamp(concat(A.issue_date,'
',A.violation_time_formatted), 'MM/dd/yyyy hh:mm'))) BETWEEN 6 AND 09 THEN 'Morning Slot - 06:00AM to 10:00AM' WHEN
HOUR(from_unixtime(unix_timestamp(concat(A.issue_date,' ',A.violation_time_formatted), 'MM/dd/yyyy hh:mm'))) BETWEEN 10 AND 13
THEN 'Mid Day Slot - 10:00AM to 02:00PM' WHEN HOUR(from_unixtime(unix_timestamp(concat(A.issue_date,'
',A.violation_time_formatted), 'MM/dd/yyyy hh:mm'))) BETWEEN 14 AND 17 THEN 'Afternoon Slot - 02:00PM to 06:00PM' WHEN
HOUR(from_unixtime(unix_timestamp(concat(A.issue_date,' ',A.violation_time_formatted), 'MM/dd/yyyy hh:mm'))) BETWEEN 18 AND 21
THEN 'Evening Slot - 06:00PM to 10:00PM' WHEN HOUR(from_unixtime(unix_timestamp(concat(A.issue_date,'
',A.violation_time_formatted), 'MM/dd/yyyy hh:mm'))) BETWEEN 22 AND 23 THEN 'Night Slot - 10:00PM to 02:00AM' WHEN
HOUR(from_unixtime(unix_timestamp(concat(A.issue_date,' ',A.violation_time_formatted), 'MM/dd/yyyy hh:mm'))) BETWEEN 0 AND 1
THEN 'Night Slot - 10:00PM to 02:00AM' WHEN HOUR(from_unixtime(unix_timestamp(concat(A.issue_date,'
',A.violation_time_formatted), 'MM/dd/yyyy hh:mm'))) BETWEEN 2 AND 5 THEN 'Early Morning Slot - 02:00AM to 06:00AM' END as
violation_time_bucket, A.issue_date, CASE WHEN MONTH(from_unixtime(unix_timestamp(concat(A.issue_date,'
',A.violation_time_formatted), 'MM/dd/yyyy hh:mm'))) IN (1,2,12) THEN 'Winter' WHEN
MONTH(from_unixtime(unix_timestamp(concat(A.issue_date,' ',A.violation_time_formatted), 'MM/dd/yyyy hh:mm'))) IN (3,4,5) THEN
'Spring' WHEN MONTH(from_unixtime(unix_timestamp(concat(A.issue_date,' ',A.violation_time_formatted), 'MM/dd/yyyy hh:mm'))) IN
(6,7,8) THEN 'Summer' WHEN MONTH(from_unixtime(unix_timestamp(concat(A.issue_date,' ',A.violation_time_formatted), 'MM/dd/yyyy
hh:mm'))) IN (9,10,11) THEN 'Fall' ELSE 'INVALID' END AS issue_date_season FROM (SELECT summons_number, violation_code,
violation_description, violation_time, CASE WHEN LENGTH(violation_time)=5 AND SUBSTR(violation_time,5,1) NOT IN ('A', 'P') THEN
'INVALID' WHEN violation_time LIKE '%+%' THEN 'INVALID' WHEN violation_time LIKE '%.%' THEN 'INVALID' WHEN
CAST(SUBSTRING(violation_time, 0, 2) AS INT)=0 THEN SUBSTRING(violation_time, 0, 2)|| ":" || SUBSTRING(violation_time, 3, 2)
WHEN LENGTH(violation_time)=4 AND CAST(SUBSTRING(violation_time, 0, 2) AS INT)<13 THEN SUBSTRING(violation_time, 0, 2)|| ":" ||
SUBSTRING(violation_time, 3, 2) WHEN LENGTH(violation_time)=4 AND CAST(SUBSTRING(violation_time, 0, 2) AS INT)>12 THEN
CAST((CAST(SUBSTRING(violation_time, 0, 2) AS INT)+ 12) AS STRING)|| ":" || SUBSTRING(violation_time, 3, 2) WHEN
CAST(SUBSTRING(violation_time, 0, 2) AS INT)>12 AND CAST(SUBSTRING(violation_time, 0, 2) AS INT)<24 THEN
SUBSTRING(violation_time, 0, 2)|| ":" || SUBSTRING(violation_time, 3, 2) WHEN CAST(SUBSTRING(violation_time, 0, 2) AS INT)>23
THEN 'INVALID' WHEN violation_time IS NULL THEN 'INVALID' WHEN violation_time = '' THEN 'INVALID' WHEN violation_time LIKE '%A'
THEN SUBSTRING(violation_time, 0, 2)|| ":" || SUBSTRING(violation_time, 3, 2) WHEN violation_time LIKE '12%P' THEN
SUBSTRING(violation_time, 0, 2)|| ":" || SUBSTRING(violation_time, 3, 2) ELSE CAST((CAST(SUBSTRING(violation_time, 0, 2) AS
INT)+ 12) AS STRING)|| ":" || SUBSTRING(violation_time, 3, 2) END AS violation_time_formatted, issue_date FROM
parking_violations_2017)A;
```

THE UNIVERSITY OF ARIZONA

UpGrad

Steps before Part II – Aggregation tasks

```
hive> CREATE TABLE IF NOT EXISTS parking_violations_2017_part2 AS SELECT A.summons_number, A.violation_code, A.violation_description, A.violation_time, A.violation_time_formatted, from_unix
time(unix_timestamp(concat(A.issue_date,' ',A.violation_time_formatted), 'MM/dd/yyyy hh:mm')) as violation_timestamp_formatted, CASE WHEN HOUR(from_unixtime(unix_timestamp(concat(A.issue_da
te,' ',A.violation_time_formatted), 'MM/dd/yyyy hh:mm'))) BETWEEN 6 AND 09 THEN 'Morning Slot - 06:00AM to 10:00AM' WHEN HOUR(from_unixtime(unix_timestamp(concat(A.issue_date,' ',A.violatio
n_time_formatted), 'MM/dd/yyyy hh:mm'))) BETWEEN 10 AND 13 THEN 'Mid Day Slot - 10:00AM to 02:00PM' WHEN HOUR(from_unixtime(unix_timestamp(concat(A.issue_date,' ',A.violation_time_formatted
), 'MM/dd/yyyy hh:mm'))) BETWEEN 14 AND 17 THEN 'Afternoon Slot - 02:00PM to 06:00PM' WHEN HOUR(from_unixtime(unix_timestamp(concat(A.issue_date,' ',A.violation_time_formatted), 'MM/dd/yyyy
 hh:mm'))) BETWEEN 18 AND 21 THEN 'Evening Slot - 06:00PM to 10:00PM' WHEN HOUR(from_unixtime(unix_timestamp(concat(A.issue_date,' ',A.violation_time_formatted), 'MM/dd/yyyy hh:mm'))) BETWE
EN 22 AND 23 THEN 'Night Slot - 10:00PM to 02:00AM' WHEN HOUR(from_unixtime(unix_timestamp(concat(A.issue_date,' ',A.violation_time_formatted), 'MM/dd/yyyy hh:mm'))) BETWEEN 0 AND 1 THEN 'N
ight Slot - 10:00PM to 02:00AM' WHEN HOUR(from_unixtime(unix_timestamp(concat(A.issue_date,' ',A.violation_time_formatted), 'MM/dd/yyyy hh:mm'))) BETWEEN 2 AND 5 THEN 'Early Morning Slot -
02:00AM to 06:00AM' END as violation_time_bucket, A.issue_date, CASE WHEN MONTH(from_unixtime(unix_timestamp(concat(A.issue_date,' ',A.violation_time_formatted), 'MM/dd/yyyy hh:mm'))) IN (1
,2,12) THEN 'Winter' WHEN MONTH(from_unixtime(unix_timestamp(concat(A.issue_date,' ',A.violation_time_formatted), 'MM/dd/yyyy hh:mm'))) IN (3,4,5) THEN 'Spring' WHEN MONTH(from_unixtime(uni
x_timestamp(concat(A.issue_date,' ',A.violation_time_formatted), 'MM/dd/yyyy hh:mm'))) IN (6,7,8) THEN 'Summer' WHEN MONTH(from_unixtime(unix_timestamp(concat(A.issue_date,' ',A.violation_t
ime_formatted), 'MM/dd/yyyy hh:mm'))) IN (9,10,11) THEN 'Fall' ELSE 'INVALID' END AS issue_date_season FROM (SELECT summons_number, violation_code, violation_description, violation_time, CA
SE WHEN LENGTH(violation_time)=5 AND SUBSTR(violation_time,5,1) NOT IN ('A', 'P') THEN 'INVALID' WHEN violation_time LIKE '%+%' THEN 'INVALID' WHEN violation_time LIKE '%.%' THEN 'INVALID'
WHEN CAST(SUBSTRING(violation_time, 0, 2) AS INT)=0 THEN SUBSTRING(violation_time, 0, 2)|| ":" || SUBSTRING(violation_time, 3, 2) WHEN LENGTH(violation_time)=4 AND CAST(SUBSTRING(violation_
time, 0, 2) AS INT)<13 THEN SUBSTRING(violation_time, 0, 2)|| ":" || SUBSTRING(violation_time, 3, 2) WHEN LENGTH(violation_time)=4 AND CAST(SUBSTRING(violation_time, 0, 2) AS INT)>12 THEN C
AST((CAST(SUBSTRING(violation_time, 0, 2) AS INT)+ 12) AS STRING)|| ":" || SUBSTRING(violation_time, 3, 2) WHEN CAST(SUBSTRING(violation_time, 0, 2) AS INT)>12 AND CAST(SUBSTRING(violation_
time, 0, 2) AS INT)<24 THEN SUBSTRING(violation_time, 0, 2)|| ":" || SUBSTRING(violation_time, 3, 2) WHEN CAST(SUBSTRING(violation_time, 0, 2) AS INT)>23 THEN 'INVALID' WHEN violation_time
IS NULL THEN 'INVALID' WHEN violation_time = '' THEN 'INVALID' WHEN violation_time LIKE '%A' THEN SUBSTRING(violation_time, 0, 2)|| ":" || SUBSTRING(violation_time, 3, 2) WHEN violation_tim
e LIKE '12%P' THEN SUBSTRING(violation_time, 0, 2)|| ":" || SUBSTRING(violation_time, 3, 2) ELSE CAST((CAST(SUBSTRING(violation_time, 0, 2) AS INT)+ 12) AS STRING)|| ":" || SUBSTRING(violat
ion_time, 3, 2) END AS violation_time_formatted, issue_date FROM parking_violations_2017)A;
Query ID = hadoop_20221027073953_8ed44327-f428-4bdc-94c4-58c9f1e422df
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1666845057966_0008)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      8          8        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 01/01  [==========================>>] 100%  ELAPSED TIME: 133.98 s
----------------------------------------------------------------------------------------------
Moving data to directory hdfs://ip-172-31-15-32.ec2.internal:8020/user/hive/warehouse/assignment.db/parking_violations_2017_part2
OK
Time taken: 144.488 seconds
```

THE UNIVERSITY OF ARIZONA

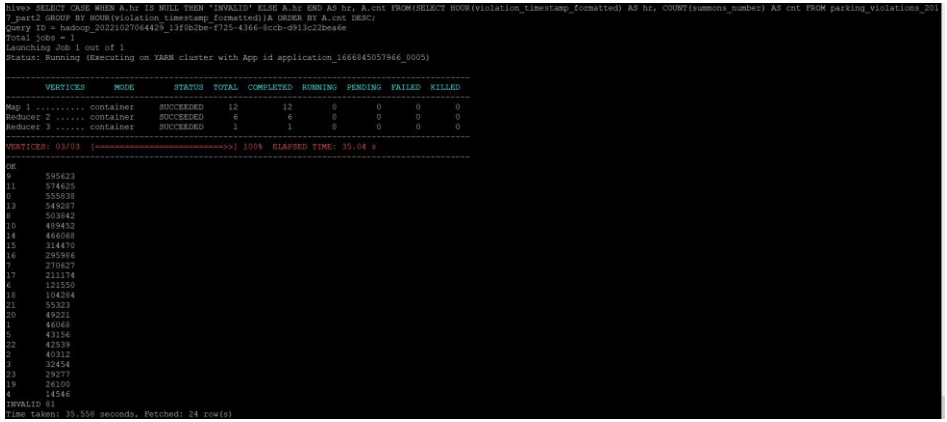
UpGrad

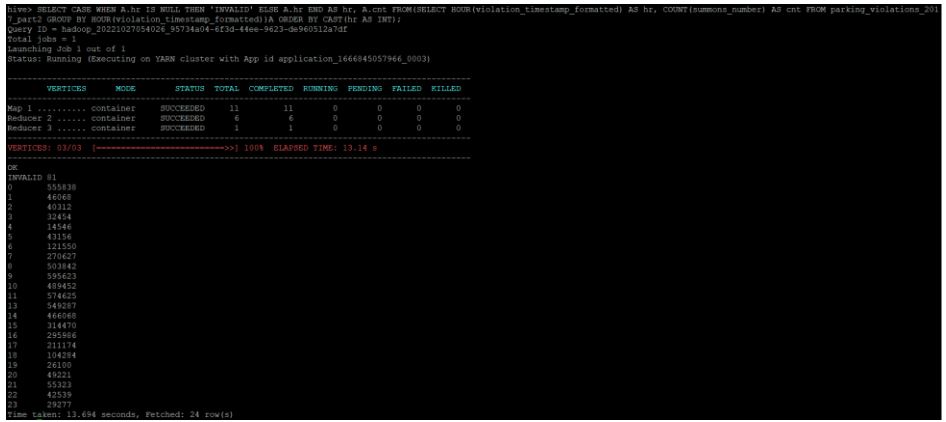
Question No: 2.1

Find out the frequency of parking violations across different times of the day.

Code

```
--ordered by frequency
SELECT CASE WHEN A.hr IS NULL THEN 'INVALID' ELSE A.hr END AS hr, A.cnt FROM(SELECT
HOUR(violation_timestamp_formatted) AS hr, COUNT(summons_number) AS cnt FROM
parking_violations_2017_part2 GROUP BY HOUR(violation_timestamp_formatted))A ORDER BY A.cnt
DESC;
--ordered by hour
SELECT CASE WHEN A.hr IS NULL THEN 'INVALID' ELSE A.hr END AS hr, A.cnt FROM(SELECT
HOUR(violation_timestamp_formatted) AS hr, COUNT(summons_number) AS cnt FROM
parking_violations_2017_part2 GROUP BY HOUR(violation_timestamp_formatted))A ORDER BY CAST(hr
AS INT);
```





THE UNIVERSITY OF ARIZONA

UpGrad

Question No: 2.2

Divide 24 hours into six 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.

Code

```
SELECT A.violation_time_bucket, A.violation_code, A.cnt FROM (SELECT violation_time_bucket, violation_code, COUNT(summons_number) as cnt, RANK() OVER(PARTITION BY violation_time_bucket ORDER BY COUNT(summons_number) DESC) AS rnk FROM parking_violations_2017_part2 GROUP BY violation_time_bucket, violation_code)A WHERE A.RNK<4 ORDER BY A.violation_time_bucket, A.cnt DESC;
```

```
hive> SELECT A.violation_time_bucket, A.violation_code, A.cnt FROM (SELECT violation_time_bucket, violation_code, COUNT(summons_number) as cnt, RANK() OVER(PARTITION BY violation_time_bucke
t ORDER BY COUNT(summons_number) DESC) AS rnk FROM parking_violations_2017_part2 GROUP BY violation_time_bucket, violation_code)A WHERE A.RNK<4 ORDER BY A.violation_time_bucket, A.RNK;
FAILED: SemanticException [Error 10002]: Line 1:363 Invalid column reference 'RNK'
hive> SELECT A.violation_time_bucket, A.violation_code, A.cnt FROM (SELECT violation_time_bucket, violation_code, COUNT(summons_number) as cnt, RANK() OVER(PARTITION BY violation_time_bucke
t ORDER BY COUNT(summons_number) DESC) AS rnk FROM parking_violations_2017_part2 GROUP BY violation_time_bucket, violation_code)A WHERE A.RNK<4 ORDER BY A.violation_time_bucket, A.cnt DESC;

Query ID = hadoop_20221027065613_fd7b00c4-2bfb-4dcd-8d19-4f27531c759d
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1666845057966_0005)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED     12         12        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      6          6        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      4          4        0        0       0       0
Reducer 4 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 04/04  [==========================>>] 100%  ELAPSED TIME: 30.70 s
----------------------------------------------------------------------------------------------
OK
NULL    21      30
NULL    94      15
NULL    46      9
Afternoon Slot - 02:00PM to 06:00PM     38      198713
Afternoon Slot - 02:00PM to 06:00PM     37      141575
Afternoon Slot - 02:00PM to 06:00PM     14      129037
Early Morning Slot - 02:00AM to 06:00AM 40      27365
Early Morning Slot - 02:00AM to 06:00AM 21      16920
Early Morning Slot - 02:00AM to 06:00AM 14      13495
Evening Slot - 06:00PM to 10:00PM       7       37587
Evening Slot - 06:00PM to 10:00PM       38      33958
Evening Slot - 06:00PM to 10:00PM       14      24559
Mid Day Slot - 10:00AM to 02:00PM       36      293985
Mid Day Slot - 10:00AM to 02:00PM       21      222467
Mid Day Slot - 10:00AM to 02:00PM       38      170274
Morning Slot - 06:00AM to 10:00AM       21      432725
Morning Slot - 06:00AM to 10:00AM       36      167209
Morning Slot - 06:00AM to 10:00AM       14      147892
Night Slot - 10:00PM to 02:00AM 36      101991
Night Slot - 10:00PM to 02:00AM 21      95517
Night Slot - 10:00PM to 02:00AM 38      58137
Time taken: 31.468 seconds, Fetched: 21 row(s)
```

THE UNIVERSITY OF ARIZONA

UpGrad

Question No: 2.3

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

Code

```
SELECT B.violation_code, B.violation_time_bucket, B.scnt FROM(SELECT violation_time_bucket,
violation_code, COUNT(summons_number) as scnt, RANK() OVER(PARTITION BY violation_code ORDER BY
COUNT(summons_number) DESC) AS crnk FROM parking_violations_2017_part2 WHERE violation_code IN
(SELECT A.violation_code FROM(SELECT violation_code, RANK() OVER(ORDER BY COUNT(summons_number)
DESC) as rnk FROM parking_violations_2017_part2 GROUP BY violation_code)A WHERE A.rnk<4) GROUP BY
violation_time_bucket, violation_code)B WHERE B.crnk = 1 ORDER BY violation_code;
```

```
hive> SELECT B.violation_code, B.violation_time_bucket, B.scnt FROM(SELECT violation_time_bucket, violation_code, COUNT(summons_number) as scnt, RANK() OVER(PARTITION BY violation_code ORDE
R BY COUNT(summons_number) DESC) AS crnk FROM parking_violations_2017_part2 WHERE violation_code IN (SELECT A.violation_code FROM(SELECT violation_code, RANK() OVER(ORDER BY COUNT(summons_n
umber) DESC) as rnk FROM parking_violations_2017_part2 GROUP BY violation_code)A WHERE A.rnk<4) GROUP BY violation_time_bucket, violation_code)B WHERE B.crnk = 1 ORDER BY violation_code;
Query ID = hadoop_20221027063227_05fda92f-c9e8-459b-a924-7aa44829eabd
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1666845057966_0005)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED     12         12        0        0       0       0
Map 5 .......... container     SUCCEEDED     12         12        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      6          6        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      4          4        0        0       0       0
Reducer 4 ...... container     SUCCEEDED      1          1        0        0       0       0
Reducer 6 ...... container     SUCCEEDED      6          6        0        0       0       0
Reducer 7 ...... container     SUCCEEDED      4          4        0        0       0       0
Reducer 8 ...... container     SUCCEEDED      2          2        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 08/08  [==========================>>] 100%  ELAPSED TIME: 44.40 s
--------------------------------------------------------------------------------
OK
21      Morning Slot - 06:00AM to 10:00AM       432725
36      Mid Day Slot - 10:00AM to 02:00PM       293985
38      Afternoon Slot - 02:00PM to 06:00PM     198713
Time taken: 45.303 seconds, Fetched: 3 row(s)
```

THE UNIVERSITY OF ARIZONA

UpGrad

Question No: 2.4.1

First, divide the year into seasons, and find the frequencies of tickets for each season.

Code

```
SELECT issue_date_season, COUNT(summons_number) as cnt from parking_violations_2017_part2
WHERE issue_date_season<> 'INVALID' GROUP BY issue_date_season;
```

```
hive> SELECT issue_date_season, COUNT(summons_number) as cnt from parking_violations_2017_part2 WHERE issue_date_season<> 'INVALID' GROUP BY issue_date_season;
Query ID = hadoop_20221027161434_799aa62c-df0e-444a-b450-882fc304570b
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1666886443989_0001)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      8          8        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      6          6        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 30.45 s
--------------------------------------------------------------------------------
OK
Summer  852849
Fall    979
Spring  2873330
Winter  1704664
Time taken: 31.198 seconds, Fetched: 4 row(s)
```

THE UNIVERSITY OF ARIZONA

UpGrad

Question No: 2.4.2

Find the 3 most common violations for each of these seasons

Code

```
SELECT A.issue_date_season, A.violation_code, A.cnt FROM(SELECT issue_date_season,
violation_code, COUNT(summons_number) as cnt, RANK() OVER(PARTITION BY issue_date_season ORDER
BY COUNT(summons_number) DESC) as rnk FROM parking_violations_2017_part2 WHERE
issue_date_season <> 'INVALID' GROUP BY issue_date_season, violation_code)A WHERE A.rnk<4;
```

```
hive> SELECT A.issue_date_season, A.violation_code, A.cnt FROM(SELECT issue_date_season, violation_code, COUNT(summons_number) as cnt, RANK() OVER(PARTITION BY issue_date_season ORDER BY CO
UNT(summons_number) DESC) as rnk FROM parking_violations_2017_part2 WHERE issue_date_season <> 'INVALID' GROUP BY issue_date_season, violation_code)A WHERE A.rnk<4;
Query ID = hadoop_20221027082825_e4a4d9c8-d367-4cb9-b3b4-71cddb40fccd
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1666858262517_0001)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      8          8        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      6          6        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      4          4        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 31.33 s
--------------------------------------------------------------------------------
OK
Spring  21      402405
Spring  36      344834
Spring  38      271167
Winter  21      238177
Winter  36      221268
Winter  38      187385
Fall    46      231
Fall    21      128
Fall    40      116
Summer  21      127342
Summer  36      96663
Summer  38      83518
Time taken: 32.145 seconds, Fetched: 12 row(s)
```