Step 1 : External table created to load the data

Query used :

```
create external table parking_violation_data(
`Summons_Number` bigint,
`Plate_Id` string,
`Reg_State` string,
`Plate_Type` string,
`Issue_Month_Day_Year` string,
`Violation_Code` int,
`VBody_Type` string,
`VMake` string,
`Issue_Agency` string,
`Street_Code1` int,
`Street_Code2` int,
`Street_Code3` int,
`VExpiry_Date` bigint,
`Violation_Location` string,
`Violation_Precinct` int,
`Issuer_Precinct` int,
`Issuer_Code` bigint,
`Issuer_Command` string,
`Issuer_Squad` string,
`Violation_Time` string,
`Time_First_Observed` string,
`Violation_County` string,
`Violation_InFront_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_InEffect` string,

`From_Hours_InEffect` string,

`To_Hours_InEffect` string,

`VColor` string,

`Is_VUnregistered` string,

`VYear` int,

`Meter_Number` string,

`Feet_From_Curb` int,

`Violation_Post_Code` string,

`Violation_Description` string,

`NoStanding_Or_Stopping_Violation` string,

`Hydrant_Violation` string,

`Double_Parking_Violation` string)

ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'

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

Step 2 : Data is loaded into external table

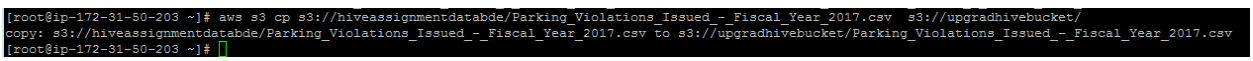
Command used : load data inpath 's3://upgradhivebucket/' into table parking_violation_data;

Here upgradhivebucket is the S3 bucket in my account

Command used to copy data into my bucket

aws s3 cp s3://hiveassignmentdatabde/Parking_Violations_Issued_-_Fiscal_Year_2017.csv s3://upgradhivebucket/

screenshot below.



Step 3 : Below commands are executed

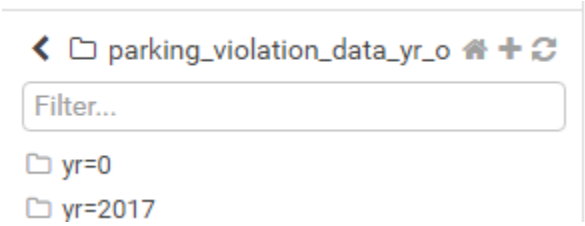
```
SET hive.exec.dynamic.partition=true;

SET hive.exec.dynamic.partition.mode=nonstrict;
```

Step 4 : ORC table created with partition based on year. And bucketing is done based on Summons_number. SNAPPY compression used.

```
create external table parking_violation_data_yr_orc(

`Summons_Number` bigint,

`Plate_Id` string,

`Reg_State` string,

`Plate_Type` string,

`Issue_Month_Day_Year` date,

`Violation_Code` int,

`VBody_Type` string,

`VMake` string,

`Issue_Agency` string,

`Street_Code1` int,

`Street_Code2` int,

`Street_Code3` int,

`VExpiry_Date` bigint,

`Violation_Location` string,

`Violation_Precinct` int,

`Issuer_Precinct` int,

`Issuer_Code` bigint,

`Issuer_Command` string,

`Issuer_Squad` string,

`Violation_Time` string,

`Time_First_Observed` string,

`Violation_County` string,
```

```
`Violation_InFront_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_InEffect` string,
`From_Hours_InEffect` string,
`To_Hours_InEffect` string,
`VColor` string,
`Is_VUnregistered` string,
`VYear` int,
`Meter_Number` string,
`Feet_From_Curb` int,
`Violation_Post_Code` string,
`Violation_Description` string,
`NoStanding_Or_Stopping_Violation` string,
`Hydrant_Violation` string,
`Double_Parking_Violation` string)
partitioned by (yr int)
clustered by (Summons_Number) into  10 buckets
stored as ORC
tblproperties("orc.compress"="SNAPPY");
```

Step 5 : Data inserted into ORC table. Here all data for year 2017 go into one partition and all other data go into another partition. Screenshot below



Insert into table parking_violation_data_yr_orc partition(yr)

Select Summons_Number,

Plate_Id,

Reg_State,

Plate_Type,

cast(to_date(from_unixtime(unix_timestamp(Issue_Month_Day_Year, 'MM/dd/yyyy'))) as date),

Violation_Code,

VBody_Type,

VMake,

Issue_Agency,

Street_Code1,

Street_Code2,

Street_Code3,

VExpiry_Date,

Violation_Location,

Violation_Precinct,

Issuer_Precinct,

Issuer_Code,

Issuer_Command,

Issuer_Squad,

Violation_Time,

Time_First_Observed,

Violation_County,

Violation_InFront_Opposite,

House_Number,

Street_Name,

Intersecting_Street,

Date_First_Observed,

Law_Section,

Sub_Division,

Violation_Legal_Code,

Days_Parking_InEffect,

From_Hours_InEffect,

To_Hours_InEffect,

VColor,

Is_VUnregistered,

VYear,

Meter_Number,

Feet_From_Curb,

Violation_Post_Code,

Violation_Description,

NoStanding_Or_Stopping_Violation,

Hydrant_Violation,

Double_Parking_Violation,

(case substring(Issue_Month_Day_Year, length(Issue_Month_Day_Year)-3, 4) when 2017 then 2017 else 0 end) as yr from parking_violation_data;

Step 6 : Custom UDFs created. One for finding Season and another for finding Bin as per requirements.

Code below. I created Java Maven project. 6 different bins are evaluated based on hour. 4 different seasons are evaluated based on months as per requirement.

```
package com.assignment.hiveudf;

import org.apache.hadoop.hive.ql.exec.UDF;

public class Season extends UDF {

	public String evaluate(String input) {

		String season = "";

		if (input == null || input.trim().equals("")) {
			return season;
		}

		try {

			String[] fields = input.split("-");
			int month = Integer.parseInt(fields[1]);

			if (month >= 3 && month <= 5) {
				season = "Spring";
			} else if (month >= 6 && month <= 8) {
				season = "Summer";
			} else if (month >= 9 && month <= 11) {
				season = "Fall";
			} else if ((month >= 1 && month <= 2) || month == 12)
{
				season = "Winter";
			}

		} catch (Exception ex) {
			ex.printStackTrace();

		}

		return season;
	}


}
```

```java
package com.assignment.hiveudf;

import org.apache.hadoop.hive.ql.exec.UDF;

public class Bin extends UDF {

	public String evaluate(String input) {

		String bin = "";

		if (input == null || input.trim().equals("") ||
input.trim().length() != 5) {
			return bin;
		}

		try {
			int hour = Integer.parseInt(input.substring(0, 2));

			String AM_PM = input.substring(input.length() - 1,
input.length());

			if ("P".equalsIgnoreCase(AM_PM)) {
				hour = hour + 12;
			}

			if (hour >= 1 && hour <= 4) {
				bin = "AfterMidnight";
			} else if (hour >= 5 && hour <= 8) {
				bin = "Morning";
			} else if (hour >= 9 && hour <= 12) {
				bin = "BeforNoon";
			} else if (hour >= 13 && hour <= 16) {
				bin = "Afternoon";
			} else if (hour >= 17 && hour <= 20) {
				bin = "Evening";
			} else if (hour >= 21 && hour <= 24) {
				bin = "Night";
			}
			return bin;
		} catch (Exception ex) {
			ex.printStackTrace();
			return bin;
		}
	}
```

```
}
```

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.assignment</groupId>
  <artifactId>hiveudf</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <properties>
          <project.build.sourceEncoding>UTF-
8</project.build.sourceEncoding>
      </properties>

      <dependencies>
          <dependency>
              <groupId>junit</groupId>
              <artifactId>junit</artifactId>
              <version>3.8.1</version>
              <scope>test</scope>
          </dependency>

          <dependency>
              <groupId>org.apache.hive</groupId>
              <artifactId>hive-exec</artifactId>
              <version>1.2.2</version>
          </dependency>

      </dependencies>

</project>
```

Step 7 : The UDF jar is uploaded into EC2 instance using FileZilla and the jar is added in hive. Temporary functions are created.

add jar /home/ec2-user/hiveudf-0.0.1-SNAPSHOT.jar

create temporary function findBin as 'com.assignment.hiveudf.Bin';

create temporary function findSeason as 'com.assignment.hiveudf.Season';

Step 8 : ORC table for bin is created partitioned by bin

```
create external table parking_violation_data_bin_orc(
`Summons_Number` bigint,
`Plate_Id` string,
`Reg_State` string,
`Plate_Type` string,
`Issue_Month_Day_Year` date,
`Violation_Code` int,
`VBody_Type` string,
`VMake` string,
`Issue_Agency` string,
`Street_Code1` int,
`Street_Code2` int,
`Street_Code3` int,
`VExpiry_Date` bigint,
`Violation_Location` string,
`Violation_Precinct` int,
`Issuer_Precinct` int,
`Issuer_Code` bigint,
`Issuer_Command` string,
`Issuer_Squad` string,
`Violation_Time` string,
`Time_First_Observed` string,
`Violation_County` string,
`Violation_InFront_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_InEffect` string,

`From_Hours_InEffect` string,

`To_Hours_InEffect` string,

`VColor` string,

`Is_VUnregistered` string,

`VYear` int,

`Meter_Number` string,

`Feet_From_Curb` int,

`Violation_Post_Code` string,

`Violation_Description` string,

`NoStanding_Or_Stopping_Violation` string,

`Hydrant_Violation` string,

`Double_Parking_Violation` string)

partitioned by (bin string)

clustered by (Summons_Number) into 10 buckets

stored as ORC

tblproperties("orc.compress"="SNAPPY");

Step 9 : ORC table for season is created partitioned by season

create external table parking_violation_data_season_orc(

`Summons_Number` bigint,

`Plate_Id` string,

`Reg_State` string,

`Plate_Type` string,

`Issue_Month_Day_Year` date,

`Violation_Code` int,

`VBody_Type` string,

`VMake` string,

`Issue_Agency` string,

`Street_Code1` int,

`Street_Code2` int,

`Street_Code3` int,

`VExpiry_Date` bigint,

`Violation_Location` string,

`Violation_Precinct` int,

`Issuer_Precinct` int,

`Issuer_Code` bigint,

`Issuer_Command` string,

`Issuer_Squad` string,

`Violation_Time` string,

`Time_First_Observed` string,

`Violation_County` string,

`Violation_InFront_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_InEffect` string,

`From_Hours_InEffect` string,

`To_Hours_InEffect` string,

`VColor` string,

`Is_VUnregistered` string,

`VYear` int,

`Meter_Number` string,

`Feet_From_Curb` int,

`Violation_Post_Code` string,

`Violation_Description` string,

`NoStanding_Or_Stopping_Violation` string,

`Hydrant_Violation` string,

`Double_Parking_Violation` string)

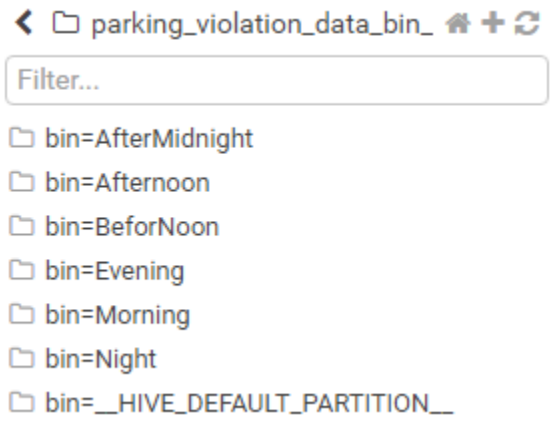
partitioned by (season string)

clustered by (Summons_Number) into 10 buckets

stored as ORC

tblproperties("orc.compress"="SNAPPY");

Step 10 : ORC table for bin is loaded with only 2017 data from ORC table partitioned by yr. While inserting data findBin function applied to load data into different partition as per following bins. Screenshot below



Insert into table parking_violation_data_bin_orc partition(bin)

Select Summons_Number,

Plate_Id,

Reg_State,

Plate_Type,

Issue_Month_Day_Year,

Violation_Code,

VBody_Type,

VMake,

Issue_Agency,

Street_Code1,

Street_Code2,

Street_Code3,

VExpiry_Date,

Violation_Location,

Violation_Precinct,

Issuer_Precinct,

Issuer_Code,

Issuer_Command,

Issuer_Squad,

Violation_Time,

Time_First_Observed,

Violation_County,

Violation_InFront_Opposite,

House_Number,

Street_Name,

Intersecting_Street,

Date_First_Observed,

Law_Section,

Sub_Division,

Violation_Legal_Code,

Days_Parking_InEffect,

From_Hours_InEffect,

To_Hours_InEffect,

VColor,

Is_VUnregistered,

VYear,

Meter_Number,

Feet_From_Curb,

Violation_Post_Code,

Violation_Description,

NoStanding_Or_Stopping_Violation,

Hydrant_Violation,

Double_Parking_Violation ,

findBin(Violation_Time) as bin from parking_violation_data_yr_orc where yr = 2017;

Step 11 : ORC table for season is loaded with only 2017 data from ORC table partitioned by yr. While inserting data findSeason function applied to load data into different partition as per following seasons. Screenshot below

Insert into table parking_violation_data_season_orc partition(season)

Select Summons_Number,

Plate_Id,

Reg_State,

Plate_Type,

Issue_Month_Day_Year,

Violation_Code,

VBody_Type,

VMake,

Issue_Agency,

Street_Code1,

Street_Code2,

Street_Code3,

VExpiry_Date,

Violation_Location,

Violation_Precinct,

Issuer_Precinct,

Issuer_Code,

Issuer_Command,

Issuer_Squad,

Violation_Time,

Time_First_Observed,

Violation_County,

Violation_InFront_Opposite,

House_Number,

Street_Name,

Intersecting_Street,

Date_First_Observed,

Law_Section,

Sub_Division,

Violation_Legal_Code,

Days_Parking_InEffect,

From_Hours_InEffect,

To_Hours_InEffect,

VColor,

Is_VUnregistered,

VYear,

Meter_Number,

Feet_From_Curb,

Violation_Post_Code,

Violation_Description,

NoStanding_Or_Stopping_Violation,

Hydrant_Violation,

Double_Parking_Violation,

findSeason(cast(Issue_Month_Day_Year as string)) as season from parking_violation_data_yr_orc where yr = 2017 ;

Part-I: Examine the data

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

Query : select count(1) from parking_violation_data_yr_orc where yr = 2017;

Answer : 5431918

Logs :

hive> select count(1) from parking_violation_data_yr_orc where yr = 2017;

Query ID = root_20190208234242_5e7033b1-06d6-4fc0-9068-e6986a4c3106

Total jobs = 1

Launching Job 1 out of 1

Number of reduce tasks determined at compile time: 1

In order to change the average load for a reducer (in bytes):

set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

set mapreduce.job.reduces=<number>

Starting Job = job_1549635067952_0167, Tracking URL = http://ip-172-31-50-203.ec2.internal:8088/proxy/application_1549635067952_0167/

Kill Command = /opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/lib/hadoop/bin/hadoop job -kill job_1549635067952_0167

Hadoop job information for Stage-1: number of mappers: 2; number of reducers: 1

2019-02-08 23:43:06,154 Stage-1 map = 0%, reduce = 0%

2019-02-08 23:43:11,536 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 5.33 sec

2019-02-08 23:43:18,765 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 8.17 sec

MapReduce Total cumulative CPU time: 8 seconds 170 msec

Ended Job = job_1549635067952_0167

MapReduce Jobs Launched:

Stage-Stage-1: Map: 2 Reduce: 1 Cumulative CPU: 8.17 sec HDFS Read: 196881 HDFS Write: 8 SUCCESS

Total MapReduce CPU Time Spent: 8 seconds 170 msec

OK

5431918

Time taken: 25.086 seconds, Fetched: 1 row(s)

2. Find out the total number of states to which the cars with tickets belong. The count of states is mandatory here, providing the exact list of states is optional.

Query : select count(distinct(Reg_State)) from parking_violation_data_yr_orc where yr = 2017 and upper(Reg_State) rlike '[A-Z]';

Here rlike has been used to filter out any non-alphabetic as state codes are always alphabets. This filters out junk data of state 99

Answer : 64

Logs :

hive> select count(distinct(Reg_State)) from parking_violation_data_yr_orc where yr = 2017 and upper(Reg_State) rlike '[A-Z]';

Query ID = root_20190208234545_3500ee5a-37f8-44fa-ae67-39fb54062dff

Total jobs = 1

Launching Job 1 out of 1

Number of reduce tasks determined at compile time: 1

In order to change the average load for a reducer (in bytes):

  set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

  set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

  set mapreduce.job.reduces=<number>

Starting Job = job_1549635067952_0168, Tracking URL = http://ip-172-31-50-203.ec2.internal:8088/proxy/application_1549635067952_0168/

Kill Command = /opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/lib/hadoop/bin/hadoop job -kill job_1549635067952_0168

Hadoop job information for Stage-1: number of mappers: 2; number of reducers: 1

2019-02-08 23:45:40,459 Stage-1 map = 0%, reduce = 0%

2019-02-08 23:45:46,816 Stage-1 map = 50%, reduce = 0%, Cumulative CPU 3.67 sec

2019-02-08 23:45:48,881 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 9.54 sec

2019-02-08 23:45:55,061 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 11.88 sec

MapReduce Total cumulative CPU time: 11 seconds 880 msec

Ended Job = job_1549635067952_0168

MapReduce Jobs Launched:

Stage-Stage-1: Map: 2 Reduce: 1   Cumulative CPU: 11.88 sec   HDFS Read: 2889771 HDFS Write: 3 SUCCESS

Total MapReduce CPU Time Spent: 11 seconds 880 msec

OK

64

Time taken: 28.191 seconds, Fetched: 1 row(s)

3. Some parking tickets don't have addresses on them, which is a cause for concern. Find out the number of such tickets, which have no addresses. (i.e. tickets where one of the Street Codes, i.e. "Street Code 1" or "Street Code 2" or "Street Code 3" is empty)

Query : select count(1) from parking_violation_data_yr_orc where (Street_Code1 = 0 or Street_Code2 = 0 or Street_Code3 = 0) and yr=2017;

Answer : 1816814

Logs :

hive> select count(1) from parking_violation_data_yr_orc where (Street_Code1 = 0 or Street_Code2 = 0 or Street_Code3 = 0) and yr=2017;

Query ID = root_20190208234848_75c62ab2-92b2-4004-8609-06e262946b1c

Total jobs = 1

Launching Job 1 out of 1

Number of reduce tasks determined at compile time: 1

In order to change the average load for a reducer (in bytes):

set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

set mapreduce.job.reduces=<number>

Starting Job = job_1549635067952_0169, Tracking URL = http://ip-172-31-50-203.ec2.internal:8088/proxy/application_1549635067952_0169/

Kill Command = /opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/lib/hadoop/bin/hadoop job -kill job_1549635067952_0169

Hadoop job information for Stage-1: number of mappers: 2; number of reducers: 1

2019-02-08 23:48:43,422 Stage-1 map = 0%, reduce = 0%

2019-02-08 23:48:49,804 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 7.25 sec

2019-02-08 23:48:54,950 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 10.24 sec

MapReduce Total cumulative CPU time: 10 seconds 240 msec

Ended Job = job_1549635067952_0169

MapReduce Jobs Launched:

Stage-Stage-1: Map: 2 Reduce: 1 Cumulative CPU: 10.24 sec HDFS Read: 38246623 HDFS Write: 8 SUCCESS

Total MapReduce CPU Time Spent: 10 seconds 240 msec

OK

1816814

Time taken: 21.939 seconds, Fetched: 1 row(s)

Part-II: Aggregation tasks

1. What are the top 5 most frequently occurring violation codes? (Note that frequency means the number of occurrences over a time period. The list should be in descending order)

Query : select Violation_Code, count(1) as count from parking_violation_data_yr_orc where yr = 2017 group by Violation_Code order by count desc limit 5;

Answer :

21 768087

36 662765

38 542079

14 476664

20 319646

Logs :

hive> select Violation_Code, count(1) as count from parking_violation_data_yr_orc where yr = 2017

> group by Violation_Code order by count desc limit 5;

Query ID = root_20190208235151_70cf1dd5-43eb-4be6-ad49-6dd509259e1f

Total jobs = 2

Launching Job 1 out of 2

Number of reduce tasks not specified. Estimated from input data size: 5

In order to change the average load for a reducer (in bytes):

set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

set mapreduce.job.reduces=<number>

Starting Job = job_1549635067952_0170, Tracking URL = http://ip-172-31-50-203.ec2.internal:8088/proxy/application_1549635067952_0170/

Kill Command = /opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/lib/hadoop/bin/hadoop job -kill job_1549635067952_0170

Hadoop job information for Stage-1: number of mappers: 2; number of reducers: 5

2019-02-08 23:51:38,887 Stage-1 map = 0%, reduce = 0%

2019-02-08 23:51:44,418 Stage-1 map = 50%, reduce = 0%, Cumulative CPU 2.69 sec

2019-02-08 23:51:45,448 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 6.36 sec

2019-02-08 23:51:52,808 Stage-1 map = 100%, reduce = 40%, Cumulative CPU 11.18 sec

2019-02-08 23:51:53,836 Stage-1 map = 100%, reduce = 60%, Cumulative CPU 14.12 sec

2019-02-08 23:51:56,989 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 18.36 sec

MapReduce Total cumulative CPU time: 18 seconds 360 msec

Ended Job = job_1549635067952_0170

Launching Job 2 out of 2

Number of reduce tasks determined at compile time: 1

In order to change the average load for a reducer (in bytes):

set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

set mapreduce.job.reduces=<number>

Starting Job = job_1549635067952_0171, Tracking URL = http://ip-172-31-50-203.ec2.internal:8088/proxy/application_1549635067952_0171/

Kill Command = /opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/lib/hadoop/bin/hadoop job -kill job_1549635067952_0171

Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1

2019-02-08 23:52:05,164 Stage-2 map = 0%, reduce = 0%

2019-02-08 23:52:10,342 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 1.59 sec

2019-02-08 23:52:15,540 Stage-2 map = 100%, reduce = 100%, Cumulative CPU 4.34 sec

MapReduce Total cumulative CPU time: 4 seconds 340 msec

Ended Job = job_1549635067952_0171

MapReduce Jobs Launched:

Stage-Stage-1: Map: 2 Reduce: 5   Cumulative CPU: 18.36 sec   HDFS Read: 5543905 HDFS Write: 2535 SUCCESS

Stage-Stage-2: Map: 1 Reduce: 1   Cumulative CPU: 4.34 sec   HDFS Read: 8695 HDFS Write: 50 SUCCESS

Total MapReduce CPU Time Spent: 22 seconds 700 msec

OK

21 768087

36 662765

38 542079

14 476664

20 319646

Time taken: 48.168 seconds, Fetched: 5 row(s)

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

Query : select VBody_Type, count(1) as count from parking_violation_data_yr_orc where yr = 2017 group by VBody_Type order by count desc limit 5;

Answer :

SUBN 1883954

4DSD 1547312

VAN 724029

DELV 358984

SDN 194197

Query : select VMake, count(1) as count from parking_violation_data_yr_orc where yr = 2017 group by VMake order by count desc limit 5;

Answer :

FORD 636844

TOYOT 605291

HONDA 538884

NISSA 462017

CHEVR 356032

Logs :

```
hive> select VBody_Type, count(1) as count from parking_violation_data_yr_orc where yr = 2017
    > group by VBody_Type order by count desc limit 5;
Query ID = root_20190208235353_b677c6c5-b273-4746-adc5-f0f12f450bc6
Total jobs = 2
Launching Job 1 out of 2
Number of reduce tasks not specified. Estimated from input data size: 5
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1549635067952_0172, Tracking URL = http://ip-172-31-50-203.ec2.internal:8088/proxy/application_1549635067952_0172/
Kill Command = /opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/lib/hadoop/bin/hadoop job  -kill job_1549635067952_0172
Hadoop job information for Stage-1: number of mappers: 2; number of reducers: 5
2019-02-08 23:53:56,112 Stage-1 map = 0%,  reduce = 0%
2019-02-08 23:54:02,518 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 7.52 sec
```

2019-02-08 23:54:09,989 Stage-1 map = 100%, reduce = 20%, Cumulative CPU 10.01 sec

2019-02-08 23:54:11,017 Stage-1 map = 100%, reduce = 60%, Cumulative CPU 14.67 sec

2019-02-08 23:54:15,248 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 18.73 sec

MapReduce Total cumulative CPU time: 18 seconds 730 msec

Ended Job = job_1549635067952_0172

Launching Job 2 out of 2

Number of reduce tasks determined at compile time: 1

In order to change the average load for a reducer (in bytes):

  set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

  set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

  set mapreduce.job.reduces=<number>

Starting Job = job_1549635067952_0173, Tracking URL = http://ip-172-31-50-203.ec2.internal:8088/proxy/application_1549635067952_0173/

Kill Command = /opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/lib/hadoop/bin/hadoop job -kill job_1549635067952_0173

Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1

2019-02-08 23:54:22,691 Stage-2 map = 0%, reduce = 0%

2019-02-08 23:54:27,889 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 1.8 sec

2019-02-08 23:54:34,059 Stage-2 map = 100%, reduce = 100%, Cumulative CPU 3.86 sec

MapReduce Total cumulative CPU time: 3 seconds 860 msec

Ended Job = job_1549635067952_0173

MapReduce Jobs Launched:

Stage-Stage-1: Map: 2 Reduce: 5   Cumulative CPU: 18.73 sec   HDFS Read: 5078266 HDFS Write: 26940 SUCCESS

Stage-Stage-2: Map: 1 Reduce: 1   Cumulative CPU: 3.86 sec   HDFS Read: 33102 HDFS Write: 60 SUCCESS

Total MapReduce CPU Time Spent: 22 seconds 590 msec

OK

SUBN 1883954

4DSD 1547312

VAN 724029

DELV 358984

SDN 194197

Time taken: 46.734 seconds, Fetched: 5 row(s)

hive> select VMake, count(1) as count from parking_violation_data_yr_orc where yr = 2017

> group by VMake order by count desc limit 5;

Query ID = root_20190208235656_7b918ca9-833a-4940-b9f0-6d912649c5ce

Total jobs = 2

Launching Job 1 out of 2

Number of reduce tasks not specified. Estimated from input data size: 5

In order to change the average load for a reducer (in bytes):

set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

set mapreduce.job.reduces=<number>

Starting Job = job_1549635067952_0174, Tracking URL = http://ip-172-31-50-203.ec2.internal:8088/proxy/application_1549635067952_0174/

Kill Command = /opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/lib/hadoop/bin/hadoop job -kill job_1549635067952_0174

Hadoop job information for Stage-1: number of mappers: 2; number of reducers: 5

2019-02-08 23:57:05,722 Stage-1 map = 0%, reduce = 0%

2019-02-08 23:57:12,021 Stage-1 map = 50%, reduce = 0%, Cumulative CPU 3.17 sec

2019-02-08 23:57:13,047 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 7.7 sec

2019-02-08 23:57:20,383 Stage-1 map = 100%, reduce = 40%, Cumulative CPU 12.96 sec

2019-02-08 23:57:21,420 Stage-1 map = 100%, reduce = 60%, Cumulative CPU 15.89 sec

2019-02-08 23:57:25,597 Stage-1 map = 100%, reduce = 80%, Cumulative CPU 18.52 sec

2019-02-08 23:57:26,619 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 20.95 sec

MapReduce Total cumulative CPU time: 20 seconds 950 msec

Ended Job = job_1549635067952_0174

Launching Job 2 out of 2

Number of reduce tasks determined at compile time: 1

In order to change the average load for a reducer (in bytes):

  set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

  set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

  set mapreduce.job.reduces=<number>

Starting Job = job_1549635067952_0175, Tracking URL = http://ip-172-31-50-203.ec2.internal:8088/proxy/application_1549635067952_0175/

Kill Command = /opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/lib/hadoop/bin/hadoop job -kill job_1549635067952_0175

Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1

2019-02-08 23:57:35,574 Stage-2 map = 0%, reduce = 0%

2019-02-08 23:57:40,775 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 1.94 sec

2019-02-08 23:57:46,943 Stage-2 map = 100%, reduce = 100%, Cumulative CPU 4.0 sec

MapReduce Total cumulative CPU time: 4 seconds 0 msec

Ended Job = job_1549635067952_0175

MapReduce Jobs Launched:

Stage-Stage-1: Map: 2 Reduce: 5   Cumulative CPU: 20.95 sec   HDFS Read: 7299454 HDFS Write: 75434 SUCCESS

Stage-Stage-2: Map: 1 Reduce: 1   Cumulative CPU: 4.0 sec   HDFS Read: 81576 HDFS Write: 64 SUCCESS

Total MapReduce CPU Time Spent: 24 seconds 950 msec

OK

FORD   636844

TOYOT   605291

HONDA   538884

NISSA 462017

CHEVR 356032

Time taken: 49.736 seconds, Fetched: 5 row(s)

3. A precinct is a police station that has a certain zone of the city under its command. You will find two further classifications of precincts:

   1. Violating Precincts - These are precincts where the violations have occurred.
   2. Issuer Precincts - These are precincts that issued the tickets.

   Find the top 5 Violating Precincts and Issuer Precincts by frequency.

Query : select Violation_Precinct, count(1) as count from parking_violation_data_yr_orc where yr = 2017 group by Violation_Precinct order by count desc limit 5;

Answer :

0 925596

19 274445

14 203553

1 174702

18 169131

Query : select Issuer_Precinct, count(1) as count from parking_violation_data_yr_orc where yr = 2017 group by Issuer_Precinct order by count desc limit 5;

Answer :

0 1078406

19 266961

14 200495

1 168740

18 162994

Logs :

```
hive> select Violation_Precinct, count(1) as count from parking_violation_data_yr_orc where yr = 2017
    > group by Violation_Precinct  order by count desc limit 5;
Query ID = root_20190208235959_93daf2f8-c7a0-4ab8-b132-33e90a99e3c3
Total jobs = 2
Launching Job 1 out of 2
Number of reduce tasks not specified. Estimated from input data size: 5
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1549635067952_0176, Tracking URL = http://ip-172-31-50-
203.ec2.internal:8088/proxy/application_1549635067952_0176/
Kill Command = /opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/lib/hadoop/bin/hadoop job  -kill
job_1549635067952_0176
Hadoop job information for Stage-1: number of mappers: 2; number of reducers: 5
2019-02-08 23:59:32,397 Stage-1 map = 0%,  reduce = 0%
2019-02-08 23:59:38,767 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 6.58 sec
2019-02-08 23:59:47,117 Stage-1 map = 100%,  reduce = 40%, Cumulative CPU 11.96 sec
2019-02-08 23:59:48,182 Stage-1 map = 100%,  reduce = 60%, Cumulative CPU 14.19 sec
2019-02-08 23:59:51,349 Stage-1 map = 100%,  reduce = 80%, Cumulative CPU 16.19 sec
2019-02-08 23:59:52,372 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 18.47 sec
MapReduce Total cumulative CPU time: 18 seconds 470 msec
Ended Job = job_1549635067952_0176
Launching Job 2 out of 2
Number of reduce tasks determined at compile time: 1
```

In order to change the average load for a reducer (in bytes):

set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

set mapreduce.job.reduces=<number>

Starting Job = job_1549635067952_0177, Tracking URL = http://ip-172-31-50-203.ec2.internal:8088/proxy/application_1549635067952_0177/

Kill Command = /opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/lib/hadoop/bin/hadoop job -kill job_1549635067952_0177

Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1

2019-02-09 00:00:00,542 Stage-2 map = 0%, reduce = 0%

2019-02-09 00:00:04,662 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 1.58 sec

2019-02-09 00:00:10,838 Stage-2 map = 100%, reduce = 100%, Cumulative CPU 4.25 sec

MapReduce Total cumulative CPU time: 4 seconds 250 msec

Ended Job = job_1549635067952_0177

MapReduce Jobs Launched:

Stage-Stage-1: Map: 2 Reduce: 5 Cumulative CPU: 18.47 sec HDFS Read: 5683674 HDFS Write: 3981 SUCCESS

Stage-Stage-2: Map: 1 Reduce: 1 Cumulative CPU: 4.25 sec HDFS Read: 10157 HDFS Write: 48 SUCCESS

Total MapReduce CPU Time Spent: 22 seconds 720 msec

OK

0 925596

19 274445

14 203553

1 174702

18 169131

Time taken: 48.155 seconds, Fetched: 5 row(s)

hive> select Issuer_Precinct, count(1) as count from parking_violation_data_yr_orc where yr = 2017

```
    > group by Issuer_Precinct order by count desc limit 5;
Query ID = root_20190209000000_7a283cbd-74b8-4871-89aa-cfd0056a271c
Total jobs = 2
Launching Job 1 out of 2
Number of reduce tasks not specified. Estimated from input data size: 5
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1549635067952_0178, Tracking URL = http://ip-172-31-50-
203.ec2.internal:8088/proxy/application_1549635067952_0178/
Kill Command = /opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/lib/hadoop/bin/hadoop job  -kill
job_1549635067952_0178
Hadoop job information for Stage-1: number of mappers: 2; number of reducers: 5
2019-02-09 00:00:49,097 Stage-1 map = 0%,  reduce = 0%
2019-02-09 00:00:54,364 Stage-1 map = 50%,  reduce = 0%, Cumulative CPU 2.49 sec
2019-02-09 00:00:55,387 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 6.45 sec
2019-02-09 00:01:02,672 Stage-1 map = 100%,  reduce = 40%, Cumulative CPU 11.06 sec
2019-02-09 00:01:03,706 Stage-1 map = 100%,  reduce = 60%, Cumulative CPU 12.91 sec
2019-02-09 00:01:06,911 Stage-1 map = 100%,  reduce = 80%, Cumulative CPU 14.85 sec
2019-02-09 00:01:07,936 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 17.35 sec
MapReduce Total cumulative CPU time: 17 seconds 350 msec
Ended Job = job_1549635067952_0178
Launching Job 2 out of 2
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
```

In order to limit the maximum number of reducers:

set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

set mapreduce.job.reduces=<number>

Starting Job = job_1549635067952_0179, Tracking URL = http://ip-172-31-50-203.ec2.internal:8088/proxy/application_1549635067952_0179/

Kill Command = /opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/lib/hadoop/bin/hadoop job -kill job_1549635067952_0179

Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1

2019-02-09 00:01:17,519 Stage-2 map = 0%, reduce = 0%

2019-02-09 00:01:21,707 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 1.73 sec

2019-02-09 00:01:26,854 Stage-2 map = 100%, reduce = 100%, Cumulative CPU 4.45 sec

MapReduce Total cumulative CPU time: 4 seconds 450 msec

Ended Job = job_1549635067952_0179

MapReduce Jobs Launched:

Stage-Stage-1: Map: 2 Reduce: 5 Cumulative CPU: 17.35 sec HDFS Read: 5423447 HDFS Write: 11197 SUCCESS

Stage-Stage-2: Map: 1 Reduce: 1 Cumulative CPU: 4.45 sec HDFS Read: 17361 HDFS Write: 49 SUCCESS

Total MapReduce CPU Time Spent: 21 seconds 800 msec

OK

0 1078406

19 266961

14 200495

1 168740

18 162994

Time taken: 47.992 seconds, Fetched: 5 row(s)

4. Find the violation code frequency across the top 3 precincts which have issued the highest number of tickets. Do these precinct zones have an exceptionally high frequency of certain violation codes? If yes, list them.

Query : SELECT * FROM

```
(
  SELECT  * FROM
  (
    SELECT Violation_Precinct  , violation_code, count, Rank() OVER ( partition BY Violation_Precinct ORDER BY count DESC ) AS rank FROM
    (
      SELECT Violation_Precinct  , violation_code, Count(1) AS count FROM
      (
        SELECT Violation_Precinct  , violation_code  FROM  parking_violation_data_yr_orc WHERE yr = 2017
      ) tbl1 GROUP BY Violation_Precinct  , violation_code
    ) tbl2
  ) tblrank where rank < 4 ORDER BY Violation_Precinct, rank
) tbltop3 WHERE tbltop3.Violation_Precinct IN
  (
    select Violation_Precinct from
    (
      select Violation_Precinct,count(1) as count from parking_violation_data_yr_orc WHERE yr = 2017 group by Violation_Precinct order by count desc limit 3
    ) tbl4
  );
```

Answer :

```
0     36     662765  1
0     7      210174  2
```

0 5 48076 3

14 14 45885 1

14 69 30465 2

14 31 22649 3

19 46 50785 1

19 38 37483 2

19 37 36468 3

In this query I am fetching the top 3 violation_precinct and the corresponding top 3 violation_code in these precinct. As the requirement does not mention which precinct to use I have used violating_precinct. Similar query can be written using issuer_precinct. I am also showing the rank at the last column

By observing the data I can conclude that violation-code of 36 and 7 have an exceptionally high frequency

Logs :

```
hive> SELECT * FROM
   > (
   >    SELECT  * FROM
   >    (
   >       SELECT Violation_Precinct  , violation_code, count, Rank() OVER ( partition BY Violation_Precinct ORDER BY count DESC ) AS rank FROM
   >       (
   >          SELECT Violation_Precinct  , violation_code, Count(1) AS count FROM
   >          (
   >             SELECT Violation_Precinct  , violation_code  FROM  parking_violation_data_yr_orc WHERE yr = 2017
   >          ) tbl1 GROUP BY Violation_Precinct  , violation_code
   >       ) tbl2
   >    ) tblrank where rank < 4 ORDER BY Violation_Precinct, rank
   > ) tbltop3 WHERE tbltop3.Violation_Precinct IN
```

```
   >    (
   >        select Violation_Precinct from
   >        (
   >            select Violation_Precinct,count(1) as count from parking_violation_data_yr_orc WHERE yr =
2017 group by Violation_Precinct order by count desc limit 3
   >        ) tbl4
   >    );
Query ID = root_20190209000303_8237819f-6b4f-472c-9a21-d8d9e7949655
Total jobs = 7
Launching Job 1 out of 7
Number of reduce tasks not specified. Estimated from input data size: 5
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1549635067952_0180, Tracking URL = http://ip-172-31-50-
203.ec2.internal:8088/proxy/application_1549635067952_0180/
Kill Command = /opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/lib/hadoop/bin/hadoop job  -kill
job_1549635067952_0180
Hadoop job information for Stage-1: number of mappers: 2; number of reducers: 5
2019-02-09 00:03:27,006 Stage-1 map = 0%,  reduce = 0%
2019-02-09 00:03:34,278 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 7.67 sec
2019-02-09 00:03:42,703 Stage-1 map = 100%,  reduce = 60%, Cumulative CPU 13.09 sec
2019-02-09 00:03:46,844 Stage-1 map = 100%,  reduce = 80%, Cumulative CPU 18.57 sec
2019-02-09 00:03:47,867 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 21.04 sec
MapReduce Total cumulative CPU time: 21 seconds 40 msec
Ended Job = job_1549635067952_0180
Launching Job 2 out of 7
```

Number of reduce tasks not specified. Estimated from input data size: 5

In order to change the average load for a reducer (in bytes):

set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

set mapreduce.job.reduces=<number>

Starting Job = job_1549635067952_0181, Tracking URL = http://ip-172-31-50-203.ec2.internal:8088/proxy/application_1549635067952_0181/

Kill Command = /opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/lib/hadoop/bin/hadoop job -kill job_1549635067952_0181

Hadoop job information for Stage-5: number of mappers: 2; number of reducers: 5

2019-02-09 00:03:56,228 Stage-5 map = 0%, reduce = 0%

2019-02-09 00:04:03,477 Stage-5 map = 100%, reduce = 0%, Cumulative CPU 6.25 sec

2019-02-09 00:04:10,742 Stage-5 map = 100%, reduce = 20%, Cumulative CPU 8.11 sec

2019-02-09 00:04:11,767 Stage-5 map = 100%, reduce = 40%, Cumulative CPU 11.05 sec

2019-02-09 00:04:12,843 Stage-5 map = 100%, reduce = 60%, Cumulative CPU 13.98 sec

2019-02-09 00:04:15,948 Stage-5 map = 100%, reduce = 80%, Cumulative CPU 16.51 sec

2019-02-09 00:04:16,970 Stage-5 map = 100%, reduce = 100%, Cumulative CPU 19.44 sec

MapReduce Total cumulative CPU time: 19 seconds 440 msec

Ended Job = job_1549635067952_0181

Launching Job 3 out of 7

Number of reduce tasks not specified. Estimated from input data size: 1

In order to change the average load for a reducer (in bytes):

set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

set mapreduce.job.reduces=<number>

Starting Job = job_1549635067952_0182, Tracking URL = http://ip-172-31-50-203.ec2.internal:8088/proxy/application_1549635067952_0182/

Kill Command = /opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/lib/hadoop/bin/hadoop job -kill job_1549635067952_0182

Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1

2019-02-09 00:04:23,932 Stage-2 map = 0%, reduce = 0%

2019-02-09 00:04:29,096 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 2.07 sec

2019-02-09 00:04:35,242 Stage-2 map = 100%, reduce = 100%, Cumulative CPU 5.45 sec

MapReduce Total cumulative CPU time: 5 seconds 450 msec

Ended Job = job_1549635067952_0182

Launching Job 4 out of 7

Number of reduce tasks determined at compile time: 1

In order to change the average load for a reducer (in bytes):

set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

set mapreduce.job.reduces=<number>

Starting Job = job_1549635067952_0183, Tracking URL = http://ip-172-31-50-203.ec2.internal:8088/proxy/application_1549635067952_0183/

Kill Command = /opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/lib/hadoop/bin/hadoop job -kill job_1549635067952_0183

Hadoop job information for Stage-6: number of mappers: 1; number of reducers: 1

2019-02-09 00:04:42,294 Stage-6 map = 0%, reduce = 0%

2019-02-09 00:04:47,419 Stage-6 map = 100%, reduce = 0%, Cumulative CPU 1.53 sec

2019-02-09 00:04:52,580 Stage-6 map = 100%, reduce = 100%, Cumulative CPU 4.96 sec

MapReduce Total cumulative CPU time: 4 seconds 960 msec

Ended Job = job_1549635067952_0183

Launching Job 5 out of 7

Number of reduce tasks determined at compile time: 1

In order to change the average load for a reducer (in bytes):

set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

set mapreduce.job.reduces=<number>

Starting Job = job_1549635067952_0184, Tracking URL = http://ip-172-31-50-203.ec2.internal:8088/proxy/application_1549635067952_0184/

Kill Command = /opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/lib/hadoop/bin/hadoop job -kill job_1549635067952_0184

Hadoop job information for Stage-3: number of mappers: 1; number of reducers: 1

2019-02-09 00:05:02,453 Stage-3 map = 0%, reduce = 0%

2019-02-09 00:05:07,595 Stage-3 map = 100%, reduce = 0%, Cumulative CPU 1.61 sec

2019-02-09 00:05:12,767 Stage-3 map = 100%, reduce = 100%, Cumulative CPU 3.62 sec

MapReduce Total cumulative CPU time: 3 seconds 620 msec

Ended Job = job_1549635067952_0184

Stage-9 is selected by condition resolver.

Stage-4 is filtered out by condition resolver.

Execution log at: /tmp/root/root_20190209000303_8237819f-6b4f-472c-9a21-d8d9e7949655.log

2019-02-09 12:05:17 Starting to launch local task to process map join; maximum memory = 1908932608

2019-02-09 12:05:18 Dump the side-table for tag: 1 with group count: 3 into file: file:/tmp/root/7f77e440-7ff6-4d94-bea4-4008899b58c6/hive_2019-02-09_00-03-22_174_6315304476652744239-1/-local-10008/HashTable-Stage-7/MapJoin-mapfile01--.hashtable

2019-02-09 12:05:18 Uploaded 1 File to: file:/tmp/root/7f77e440-7ff6-4d94-bea4-4008899b58c6/hive_2019-02-09_00-03-22_174_6315304476652744239-1/-local-10008/HashTable-Stage-7/MapJoin-mapfile01--.hashtable (314 bytes)

2019-02-09 12:05:18 End of local task; Time Taken: 0.741 sec.

Execution completed successfully

MapredLocal task succeeded

Launching Job 7 out of 7

Number of reduce tasks is set to 0 since there's no reduce operator

Starting Job = job_1549635067952_0185, Tracking URL = http://ip-172-31-50-203.ec2.internal:8088/proxy/application_1549635067952_0185/

Kill Command = /opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/lib/hadoop/bin/hadoop job -kill job_1549635067952_0185

Hadoop job information for Stage-7: number of mappers: 1; number of reducers: 0

2019-02-09 00:05:25,230 Stage-7 map = 0%, reduce = 0%

2019-02-09 00:05:30,380 Stage-7 map = 100%, reduce = 0%, Cumulative CPU 2.59 sec

MapReduce Total cumulative CPU time: 2 seconds 590 msec

Ended Job = job_1549635067952_0185

MapReduce Jobs Launched:

Stage-Stage-1: Map: 2 Reduce: 5 Cumulative CPU: 21.04 sec HDFS Read: 11009838 HDFS Write: 102889 SUCCESS

Stage-Stage-5: Map: 2 Reduce: 5 Cumulative CPU: 19.44 sec HDFS Read: 5683875 HDFS Write: 3981 SUCCESS

Stage-Stage-2: Map: 1 Reduce: 1 Cumulative CPU: 5.45 sec HDFS Read: 111087 HDFS Write: 10482 SUCCESS

Stage-Stage-6: Map: 1 Reduce: 1 Cumulative CPU: 4.96 sec HDFS Read: 9825 HDFS Write: 150 SUCCESS

Stage-Stage-3: Map: 1 Reduce: 1 Cumulative CPU: 3.62 sec HDFS Read: 14999 HDFS Write: 10482 SUCCESS

Stage-Stage-7: Map: 1 Cumulative CPU: 2.59 sec HDFS Read: 15389 HDFS Write: 123 SUCCESS

Total MapReduce CPU Time Spent: 57 seconds 100 msec

OK

0 36 662765 1

0 7 210174 2

0 5 48076 3

14 14 45885 1

14 69 30465 2

14 31 22649 3

19 46 50785 1

19 38 37483 2

19 37 36468 3

Time taken: 130.99 seconds, Fetched: 9 row(s)

5. Find out the frequency 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.

Query : select bin , count(1) as count from parking_violation_data_bin_orc WHERE bin IN ('AfterMidnight','Morning','BeforNoon','Afternoon','Evening','Night') group by bin;

Here IN clause specifying the different bins is used so that junk data in the default partition are not shown

Answer :

Evening 390778

Morning 939181

Night 637155

Afternoon 1625806

AfterMidnight 133380

BeforNoon 1677075

Logs :

hive> select bin , count(1) as count from parking_violation_data_bin_orc WHERE bin IN ('AfterMidnight','Morning','BeforNoon','Afternoon','Evening','Night')

> group by bin;

Query ID = root_20190209002828_da3cccf2-5d39-4b5d-9e7b-44dc13f30cde

Total jobs = 1

Launching Job 1 out of 1

Number of reduce tasks not specified. Estimated from input data size: 5

In order to change the average load for a reducer (in bytes):

set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

set mapreduce.job.reduces=<number>

Starting Job = job_1549635067952_0194, Tracking URL = http://ip-172-31-50-203.ec2.internal:8088/proxy/application_1549635067952_0194/

Kill Command = /opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/lib/hadoop/bin/hadoop job -kill job_1549635067952_0194

Hadoop job information for Stage-1: number of mappers: 2; number of reducers: 5

2019-02-09 00:28:38,016 Stage-1 map = 0%, reduce = 0%

2019-02-09 00:28:45,270 Stage-1 map = 50%, reduce = 0%, Cumulative CPU 2.49 sec

2019-02-09 00:28:46,292 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 6.12 sec

2019-02-09 00:28:52,551 Stage-1 map = 100%, reduce = 20%, Cumulative CPU 8.65 sec

2019-02-09 00:28:53,575 Stage-1 map = 100%, reduce = 60%, Cumulative CPU 12.62 sec

2019-02-09 00:28:57,728 Stage-1 map = 100%, reduce = 80%, Cumulative CPU 14.79 sec

2019-02-09 00:28:58,753 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 17.13 sec

MapReduce Total cumulative CPU time: 17 seconds 130 msec

Ended Job = job_1549635067952_0194

MapReduce Jobs Launched:

Stage-Stage-1: Map: 2 Reduce: 5 Cumulative CPU: 17.13 sec HDFS Read: 272385 HDFS Write: 100 SUCCESS

Total MapReduce CPU Time Spent: 17 seconds 130 msec

OK

Evening 390778

Morning 939181

Night 637155

Afternoon 1625806

AfterMidnight 133380

BeforNoon 1677075

Time taken: 27.987 seconds, Fetched: 6 row(s)

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.

Query :

```
SELECT * FROM
(
  SELECT bin, violation_code, count, Rank() OVER ( partition BY bin ORDER BY count DESC ) AS rank FROM
  (
    SELECT bin, violation_code, Count(1) AS count FROM
    (
      SELECT violation_code, bin FROM  parking_violation_data_bin_orc WHERE bin IN ('AfterMidnight','Morning','BeforNoon','Afternoon','Evening','Night')
    ) tbl1 GROUP BY bin, violation_code
  ) tbl2
) tblrank WHERE tblrank.rank < 4 ORDER BY bin, tblrank.rank;
```

Here IN clause used in query to query fata loaded in different bins. Top 3 vioaltion_code is found in each of the bins (different times of the day). There are 6 different bins each spanning for 4 hours

Answer :

AfterMidnight 21 25479 1

AfterMidnight 40 23216 2

AfterMidnight 78 13321 3

Afternoon 38 234449 1

Afternoon 36 193003 2

Afternoon 37 166793 3

BeforNoon 21 409505 1

BeforNoon 36 286999 2

BeforNoon 38 150877 3

Evening 38 61211 1

Evening 7 45290 2

Evening 14 40214 3

Morning 21 247428 1

Morning 14 109763 2

Morning 36 75948 3

Night 36 101991 1

Night 21 72510 2

Night 38 68263 3

Logs :

```
hive> SELECT * FROM
    > (
    >     SELECT bin, violation_code, count, Rank() OVER ( partition BY bin ORDER BY count DESC ) AS rank
FROM
    >     (
    >         SELECT bin, violation_code, Count(1) AS count FROM
    >         (
```

```
    >           SELECT violation_code, bin FROM  parking_violation_data_bin_orc WHERE bin IN
('AfterMidnight','Morning','BeforNoon','Afternoon','Evening','Night')
    >       ) tbl1 GROUP BY bin, violation_code
    >    ) tbl2
    > ) tblrank WHERE tblrank.rank < 4 ORDER BY bin, tblrank.rank;
Query ID = root_20190209001414_00623165-a741-4684-a12c-39b2295f79f9
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks not specified. Estimated from input data size: 5
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1549635067952_0186, Tracking URL = http://ip-172-31-50-
203.ec2.internal:8088/proxy/application_1549635067952_0186/
Kill Command = /opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/lib/hadoop/bin/hadoop job  -kill
job_1549635067952_0186
Hadoop job information for Stage-1: number of mappers: 2; number of reducers: 5
2019-02-09 00:14:58,967 Stage-1 map = 0%,  reduce = 0%
2019-02-09 00:15:05,237 Stage-1 map = 50%,  reduce = 0%, Cumulative CPU 2.76 sec
2019-02-09 00:15:07,282 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 6.84 sec
2019-02-09 00:15:14,574 Stage-1 map = 100%,  reduce = 20%, Cumulative CPU 9.42 sec
2019-02-09 00:15:15,614 Stage-1 map = 100%,  reduce = 60%, Cumulative CPU 15.13 sec
2019-02-09 00:15:19,749 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 19.83 sec
MapReduce Total cumulative CPU time: 19 seconds 830 msec
Ended Job = job_1549635067952_0186
Launching Job 2 out of 3
Number of reduce tasks not specified. Estimated from input data size: 1
```

In order to change the average load for a reducer (in bytes):

set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

set mapreduce.job.reduces=<number>

Starting Job = job_1549635067952_0187, Tracking URL = http://ip-172-31-50-203.ec2.internal:8088/proxy/application_1549635067952_0187/

Kill Command = /opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/lib/hadoop/bin/hadoop job -kill job_1549635067952_0187

Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1

2019-02-09 00:15:28,274 Stage-2 map = 0%, reduce = 0%

2019-02-09 00:15:32,386 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 1.64 sec

2019-02-09 00:15:38,581 Stage-2 map = 100%, reduce = 100%, Cumulative CPU 4.7 sec

MapReduce Total cumulative CPU time: 4 seconds 700 msec

Ended Job = job_1549635067952_0187

Launching Job 3 out of 3

Number of reduce tasks determined at compile time: 1

In order to change the average load for a reducer (in bytes):

set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

set mapreduce.job.reduces=<number>

Starting Job = job_1549635067952_0188, Tracking URL = http://ip-172-31-50-203.ec2.internal:8088/proxy/application_1549635067952_0188/

Kill Command = /opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/lib/hadoop/bin/hadoop job -kill job_1549635067952_0188

Hadoop job information for Stage-3: number of mappers: 1; number of reducers: 1

2019-02-09 00:15:46,355 Stage-3 map = 0%, reduce = 0%

2019-02-09 00:15:51,508 Stage-3 map = 100%, reduce = 0%, Cumulative CPU 1.4 sec

2019-02-09 00:15:56,694 Stage-3 map = 100%, reduce = 100%, Cumulative CPU 3.43 sec

MapReduce Total cumulative CPU time: 3 seconds 430 msec

Ended Job = job_1549635067952_0188

MapReduce Jobs Launched:

Stage-Stage-1: Map: 2 Reduce: 5   Cumulative CPU: 19.83 sec   HDFS Read: 5442810 HDFS Write: 16924 SUCCESS

Stage-Stage-2: Map: 1 Reduce: 1   Cumulative CPU: 4.7 sec   HDFS Read: 25085 HDFS Write: 672 SUCCESS

Stage-Stage-3: Map: 1 Reduce: 1   Cumulative CPU: 3.43 sec   HDFS Read: 6314 HDFS Write: 374 SUCCESS

Total MapReduce CPU Time Spent: 27 seconds 960 msec

OK

AfterMidnight   21   25479   1

AfterMidnight   40   23216   2

AfterMidnight   78   13321   3

Afternoon   38   234449 1

Afternoon   36   193003 2

Afternoon   37   166793 3

BeforNoon   21   409505 1

BeforNoon   36   286999 2

BeforNoon   38   150877 3

Evening 38   61211   1

Evening 7   45290   2

Evening 14   40214   3

Morning 21   247428 1

Morning 14   109763 2

Morning 36   75948   3

Night   36   101991 1

Night   21   72510   2

Night 38 68263 3

Time taken: 64.768 seconds, Fetched: 18 row(s)

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).

Query :

```
SELECT violation_code, bin, count FROM
(
  SELECT  violation_code, bin, count, rank FROM
  (
    SELECT violation_code, bin, count, Rank() OVER ( partition BY violation_code ORDER BY count DESC ) AS rank FROM
    (
      SELECT violation_code, bin, Count(1) AS count FROM
      (
        SELECT violation_code, bin FROM  parking_violation_data_bin_orc WHERE bin IN ('AfterMidnight','Morning','BeforNoon','Afternoon','Evening','Night')
      ) tbl1 GROUP BY violation_code , bin
    ) tbl2
  ) tblrank WHERE rank < 2 ORDER BY violation_code, rank
) tblrank2 order by count desc limit 3;
```

Answer :

```
21    BeforNoon    409505
36    BeforNoon    286999
38    Afternoon    234449
```

Logs :

```
hive> SELECT violation_code, bin, count FROM
   > (
   >    SELECT  violation_code, bin, count, rank FROM
   >    (
   >       SELECT violation_code, bin, count, Rank() OVER ( partition BY violation_code ORDER BY count DESC ) AS rank FROM
   >       (
   >          SELECT violation_code, bin, Count(1) AS count FROM
   >          (
   >             SELECT violation_code, bin FROM  parking_violation_data_bin_orc WHERE bin IN ('AfterMidnight','Morning','BeforNoon','Afternoon','Evening','Night')
   >          ) tbl1 GROUP BY violation_code , bin
   >       ) tbl2
   >    ) tblrank WHERE rank < 2 ORDER BY violation_code, rank
   > ) tblrank2 order by count desc limit 3;
Query ID = root_20190209003131_f7f684c0-0404-4008-aad5-9d4bf24eb1f4
Total jobs = 4
Launching Job 1 out of 4
Number of reduce tasks not specified. Estimated from input data size: 5
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1549635067952_0195, Tracking URL = http://ip-172-31-50-203.ec2.internal:8088/proxy/application_1549635067952_0195/
Kill Command = /opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/lib/hadoop/bin/hadoop job  -kill job_1549635067952_0195
```

Hadoop job information for Stage-1: number of mappers: 2; number of reducers: 5

2019-02-09 00:31:26,758 Stage-1 map = 0%, reduce = 0%

2019-02-09 00:31:33,034 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 6.6 sec

2019-02-09 00:31:41,416 Stage-1 map = 100%, reduce = 40%, Cumulative CPU 11.09 sec

2019-02-09 00:31:42,445 Stage-1 map = 100%, reduce = 60%, Cumulative CPU 13.4 sec

2019-02-09 00:31:46,602 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 18.25 sec

MapReduce Total cumulative CPU time: 18 seconds 250 msec

Ended Job = job_1549635067952_0195

Launching Job 2 out of 4

Number of reduce tasks not specified. Estimated from input data size: 1

In order to change the average load for a reducer (in bytes):

  set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

  set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

  set mapreduce.job.reduces=<number>

Starting Job = job_1549635067952_0196, Tracking URL = http://ip-172-31-50-203.ec2.internal:8088/proxy/application_1549635067952_0196/

Kill Command = /opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/lib/hadoop/bin/hadoop job -kill job_1549635067952_0196

Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1

2019-02-09 00:31:55,988 Stage-2 map = 0%, reduce = 0%

2019-02-09 00:32:01,132 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 1.73 sec

2019-02-09 00:32:07,312 Stage-2 map = 100%, reduce = 100%, Cumulative CPU 5.07 sec

MapReduce Total cumulative CPU time: 5 seconds 70 msec

Ended Job = job_1549635067952_0196

Launching Job 3 out of 4

Number of reduce tasks determined at compile time: 1

In order to change the average load for a reducer (in bytes):

set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

set mapreduce.job.reduces=<number>

Starting Job = job_1549635067952_0197, Tracking URL = http://ip-172-31-50-203.ec2.internal:8088/proxy/application_1549635067952_0197/

Kill Command = /opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/lib/hadoop/bin/hadoop job -kill job_1549635067952_0197

Hadoop job information for Stage-3: number of mappers: 1; number of reducers: 1

2019-02-09 00:32:15,878 Stage-3 map = 0%, reduce = 0%

2019-02-09 00:32:21,019 Stage-3 map = 100%, reduce = 0%, Cumulative CPU 1.48 sec

2019-02-09 00:32:25,137 Stage-3 map = 100%, reduce = 100%, Cumulative CPU 3.33 sec

MapReduce Total cumulative CPU time: 3 seconds 330 msec

Ended Job = job_1549635067952_0197

Launching Job 4 out of 4

Number of reduce tasks determined at compile time: 1

In order to change the average load for a reducer (in bytes):

set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

set mapreduce.job.reduces=<number>

Starting Job = job_1549635067952_0198, Tracking URL = http://ip-172-31-50-203.ec2.internal:8088/proxy/application_1549635067952_0198/

Kill Command = /opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/lib/hadoop/bin/hadoop job -kill job_1549635067952_0198

Hadoop job information for Stage-4: number of mappers: 1; number of reducers: 1

2019-02-09 00:32:33,892 Stage-4 map = 0%, reduce = 0%

2019-02-09 00:32:39,054 Stage-4 map = 100%, reduce = 0%, Cumulative CPU 1.5 sec

2019-02-09 00:32:45,276 Stage-4 map = 100%, reduce = 100%, Cumulative CPU 4.48 sec

MapReduce Total cumulative CPU time: 4 seconds 480 msec

Ended Job = job_1549635067952_0198

MapReduce Jobs Launched:

Stage-Stage-1: Map: 2 Reduce: 5   Cumulative CPU: 18.25 sec   HDFS Read: 5442828 HDFS Write: 16924 SUCCESS

Stage-Stage-2: Map: 1 Reduce: 1   Cumulative CPU: 5.07 sec   HDFS Read: 25105 HDFS Write: 3516 SUCCESS

Stage-Stage-3: Map: 1 Reduce: 1   Cumulative CPU: 3.33 sec   HDFS Read: 8142 HDFS Write: 3407 SUCCESS

Stage-Stage-4: Map: 1 Reduce: 1   Cumulative CPU: 4.48 sec   HDFS Read: 8808 HDFS Write: 60 SUCCESS

Total MapReduce CPU Time Spent: 31 seconds 130 msec

OK

21	BeforNoon	409505

36	BeforNoon	286999

38	Afternoon	234449

Time taken: 88.662 seconds, Fetched: 3 row(s)

8. Let's try and find some seasonality in this data:

    1. First, divide the year into seasons, and find frequencies of tickets for each season. (Hint: A quick Google search reveals the following seasons in NYC: Spring(March, April, May); Summer(June, July, August); Fall(September, October, November); Winter(December, January, February))
    2. Then, find the 3 most common violations for each of these seasons.

Query 1 :

```
select season , count(1) as count from parking_violation_data_season_orc where season IN ('Summer','Winter','Fall','Spring') group by season;
```

Here IN clause is used specifying the seasons so that junk data in Default partition is not fetched.

Answer 1 :

Summer 852866

Winter 1704690

Fall 979

Spring 2873383

Query 2 :

```
SELECT * FROM
(
  SELECT season, violation_code, count, Rank() OVER ( partition BY season ORDER BY count DESC ) AS rank FROM
  (
    SELECT season, violation_code, Count(1) AS count FROM
    (
      SELECT violation_code, season FROM  parking_violation_data_season_orc WHERE violation_time rlike '((1[0-2]|0[1-9])([0-5][0-9])([AaPp]))'
    ) tbl1 GROUP BY season, violation_code
```

) tbl2

) tblrank WHERE tblrank.rank < 4 ORDER BY season, tblrank.rank;

Answer 2 :

Fall 46 231 1

Fall 21 128 2

Fall 40 113 3

Spring 21 396800 1

Spring 36 344834 2

Spring 38 271164 3

Summer 21 125480 1

Summer 36 96663 2

Summer 38 83518 3

Winter 21 235127 1

Winter 36 221268 2

Winter 38 187383 3

Logs :

hive> select season , count(1) as count from parking_violation_data_season_orc where season IN ('Summer','Winter','Fall','Spring')

> group by season;

Query ID = root_20190209003636_a3ec945c-af7d-4aac-b26e-58a7fe72dee8

Total jobs = 1

Launching Job 1 out of 1

Number of reduce tasks not specified. Estimated from input data size: 5

In order to change the average load for a reducer (in bytes):

set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

set mapreduce.job.reduces=<number>

Starting Job = job_1549635067952_0202, Tracking URL = http://ip-172-31-50-203.ec2.internal:8088/proxy/application_1549635067952_0202/

Kill Command = /opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/lib/hadoop/bin/hadoop job -kill job_1549635067952_0202

Hadoop job information for Stage-1: number of mappers: 2; number of reducers: 5

2019-02-09 00:36:45,839 Stage-1 map = 0%, reduce = 0%

2019-02-09 00:36:52,082 Stage-1 map = 50%, reduce = 0%, Cumulative CPU 2.62 sec

2019-02-09 00:36:53,105 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 6.18 sec

2019-02-09 00:37:01,350 Stage-1 map = 100%, reduce = 40%, Cumulative CPU 10.5 sec

2019-02-09 00:37:02,402 Stage-1 map = 100%, reduce = 60%, Cumulative CPU 12.35 sec

2019-02-09 00:37:06,522 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 17.93 sec

MapReduce Total cumulative CPU time: 17 seconds 930 msec

Ended Job = job_1549635067952_0202

MapReduce Jobs Launched:

Stage-Stage-1: Map: 2 Reduce: 5 Cumulative CPU: 17.93 sec HDFS Read: 210808 HDFS Write: 53 SUCCESS

Total MapReduce CPU Time Spent: 17 seconds 930 msec

OK

Summer 852866

Winter 1704690

Fall 979

Spring 2873383

Time taken: 27.994 seconds, Fetched: 4 row(s)

hive>

> SELECT * FROM

> (

```
    >    SELECT season, violation_code, count, Rank() OVER ( partition BY season ORDER BY count DESC ) AS rank FROM
    >    (
    >        SELECT season, violation_code, Count(1) AS count FROM
    >        (
    >            SELECT violation_code, season FROM  parking_violation_data_season_orc WHERE violation_time rlike '((1[0-2]|0[1-9])([0-5][0-9])([AaPp]))'
    >        ) tbl1 GROUP BY season, violation_code
    >    ) tbl2
    > ) tblrank WHERE tblrank.rank < 4 ORDER BY season, tblrank.rank;
Query ID = root_20190209003434_6e2db1dc-6912-4d9d-9a29-65ff93819682
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks not specified. Estimated from input data size: 5
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1549635067952_0199, Tracking URL = http://ip-172-31-50-203.ec2.internal:8088/proxy/application_1549635067952_0199/
Kill Command = /opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/lib/hadoop/bin/hadoop job  -kill job_1549635067952_0199
Hadoop job information for Stage-1: number of mappers: 2; number of reducers: 5
2019-02-09 00:34:40,934 Stage-1 map = 0%,  reduce = 0%
2019-02-09 00:34:47,180 Stage-1 map = 50%,  reduce = 0%, Cumulative CPU 3.13 sec
2019-02-09 00:34:50,245 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 7.99 sec
2019-02-09 00:34:57,552 Stage-1 map = 100%,  reduce = 20%, Cumulative CPU 10.29 sec
2019-02-09 00:34:58,573 Stage-1 map = 100%,  reduce = 40%, Cumulative CPU 12.95 sec
```

2019-02-09 00:34:59,629 Stage-1 map = 100%, reduce = 60%, Cumulative CPU 15.69 sec

2019-02-09 00:35:02,756 Stage-1 map = 100%, reduce = 80%, Cumulative CPU 18.49 sec

2019-02-09 00:35:03,783 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 20.72 sec

MapReduce Total cumulative CPU time: 20 seconds 720 msec

Ended Job = job_1549635067952_0199

Launching Job 2 out of 3

Number of reduce tasks not specified. Estimated from input data size: 1

In order to change the average load for a reducer (in bytes):

  set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

  set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

  set mapreduce.job.reduces=<number>

Starting Job = job_1549635067952_0200, Tracking URL = http://ip-172-31-50-203.ec2.internal:8088/proxy/application_1549635067952_0200/

Kill Command = /opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/lib/hadoop/bin/hadoop job -kill job_1549635067952_0200

Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1

2019-02-09 00:35:12,816 Stage-2 map = 0%, reduce = 0%

2019-02-09 00:35:17,966 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 1.57 sec

2019-02-09 00:35:23,099 Stage-2 map = 100%, reduce = 100%, Cumulative CPU 4.78 sec

MapReduce Total cumulative CPU time: 4 seconds 780 msec

Ended Job = job_1549635067952_0200

Launching Job 3 out of 3

Number of reduce tasks determined at compile time: 1

In order to change the average load for a reducer (in bytes):

  set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

  set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

set mapreduce.job.reduces=<number>

Starting Job = job_1549635067952_0201, Tracking URL = http://ip-172-31-50-203.ec2.internal:8088/proxy/application_1549635067952_0201/

Kill Command = /opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/lib/hadoop/bin/hadoop job -kill job_1549635067952_0201

Hadoop job information for Stage-3: number of mappers: 1; number of reducers: 1

2019-02-09 00:35:30,448 Stage-3 map = 0%, reduce = 0%

2019-02-09 00:35:34,652 Stage-3 map = 100%, reduce = 0%, Cumulative CPU 1.42 sec

2019-02-09 00:35:39,801 Stage-3 map = 100%, reduce = 100%, Cumulative CPU 3.48 sec

MapReduce Total cumulative CPU time: 3 seconds 480 msec

Ended Job = job_1549635067952_0201

MapReduce Jobs Launched:

Stage-Stage-1: Map: 2 Reduce: 5 Cumulative CPU: 20.72 sec HDFS Read: 16561940 HDFS Write: 9493 SUCCESS

Stage-Stage-2: Map: 1 Reduce: 1 Cumulative CPU: 4.78 sec HDFS Read: 17675 HDFS Write: 443 SUCCESS

Stage-Stage-3: Map: 1 Reduce: 1 Cumulative CPU: 3.48 sec HDFS Read: 6091 HDFS Write: 211 SUCCESS

Total MapReduce CPU Time Spent: 28 seconds 980 msec

OK

Fall 46 231 1

Fall 21 128 2

Fall 40 113 3

Spring 21 396800 1

Spring 36 344834 2

Spring 38 271164 3

Summer 21 125480 1

Summer 36 96663 2

Summer 38 83518 3

Winter 21 235127 1

Winter 36 221268 2

Winter 38 187383 3

Time taken: 69.107 seconds, Fetched: 12 row(s)