**Hive Assignment**

**Shuvam Sanyal (PRN-19070243015)**

**Tanay Goswami (PRN-19070243020)**

**Pritha Roy(PRN-19070243030)**

**Partitioning and Bucketing**

Apache Hive organizes tables into partitions for grouping same type of data together based on a column or partition key. Each table in the hive can have one or more partition keys to identify a particular partition. Using partition, we can make it faster to do queries on slices of the data.

##### **Hive Partition**can be further subdivided into **Clusters** or **Buckets**

Hive Buckets is nothing but another technique of decomposing data or decreasing the data into more manageable parts or equal parts. we can’t create number of Hive Buckets the reason is we should declare the number of buckets for a table in the time of table creation.

**Description of Input Data Source**

We have taken the air traffic San Francisco dataset from Kaggle. The dataset has several columns in it. San Francisco International Airport Report on Monthly Passenger Traffic Statistics by Airline. Airport data is seasonal in nature; therefore, any comparative analyses should be done on a period-over-period basis (i.e. January 2010 vs. January 2009) as opposed to period-to-period (i.e. January 2010 vs. February 2010). It is also important to note that fact and attribute field relationships are not always 1-to-1. For example, Passenger Counts belonging to United Airlines will appear in multiple attribute fields and are additive, which provides flexibility for the user to derive categorical Passenger Counts as desired. They are mentioned below: -

* **Activity Period**: - The period of the airlines being active.
* **Operating Airline**: - List of names of operating airlines.
* **Operating Airline IATA Code**: - The IATA codes of operation airlines.
* **Published Airline**: -List of published airlines out of operating airlines.
* **Published Airline IATA Code**: -Published airlines IATA Code.
* **GEO Summary**: -Whether the airline is domestic or international.
* **GEO Region**: - Geographical location of various operating airlines in SF airport.
* **Activity Type Code**: - Whether the flight is deplaned, enplaned or in transit through SF.
* **Price Category Code**: - Whether the airline is low cost carrier or normal carrier.
* **Terminal**: - Which terminal of SF airport the airline is operating. (Terminal 1,2,3, International, Other).
* **Boarding Area**: - Boarding Area of the passenger in airport (A, B, C, D, E, F, G, Other).
* **Passenger Count**: - Total number of passengers on daily basis at airport during the activity periods.
* **Adjusted Activity Type Code**: - Adjusted activity type code.
* **Adjusted Passenger Count**: - Adjusted count of the passengers.
* **Year**: - The whole data indicates this year.
* **Month**: - The whole data indicates this month.

**Initial Partitioning Strategy of the Dataset**

Now among all the columns, potential partitioning columns are Operating Airlines, Geo Summary, Geo Region, Price Category Code, Terminal, Year and Month.

So according to the above columns, the initial partitioning strategy would be, first we will create year wise folders. So here total 12 folder would be created corresponding to each mentioned year. Now for each year, 12 separate partitioning folders would be created referring to each of the 12 months in a year. Then inside each month’s folder we will have two partitioning folder such as domestic and international of geo summaries. Now inside geo summaries, seven folders (Asia, Australia, US, Mexico, Canada, Central America, Europe, Mexico) will be created. Now inside this folder, we will create 44 odd folders referring to each of the mentioned operation airlines.

**Hive Partitioning Command for Initial Partitioning**

### **Step-1: Create a hive table**

**create table air traffic** (Activity Period, Operating Airline, Operating Airline IATA Code, Published Airline, Published Airline IATA Code, GEO Summary, GEO Region, Activity Type Code, Price Category Code, Terminal,Boarding Area, Passenger Count, Adjusted Activity Type Code, Adjusted Passenger Count, Year, Month)

row format delimited fields

terminated by ','

stored as textfile;

### **Step-2: Load data into the hive table**

**load data local inpath** 'C:\Users\Shuvam\Desktop\air\_traffic\_data’ into table air traffic;

### **Step-3: Create a table in hive with partition**

**create table air\_traffic\_partition\_bucket** (Activity Period, Operating Airline, Operating Airline IATA Code, Published Airline, Published Airline IATA Code, GEO Summary, GEO Region, Activity Type Code, Price Category Code, Terminal Boarding Area, Passenger Count, Adjusted Activity Type Code, Adjusted Passenger Count, Year, Month) partitioned by (Year, Month, GEO Summary, GEO Region,, Price Category Code, Operating Airline);

### **Step-4: Set the properties for partition**

SET hive. exec. dynamic. partition=true;

SET hive. exec.dynamic.partition.mode=non-strict;

SET hive.enforce.bucketing =true;

### **Step-5: Final Hive Command for Initial Partitioning**

CREATE TABLE air traffic (Activity Period, Operating Airline, Operating Airline IATA Code, Published Airline, Published Airline IATA Código Summering Region, Activity Type Code, Price Category Code, Terminal, Boarding Area Passenger Count, Adjusted Activity Type Code, Adjusted Passenger Count, Year, Month)

PARTITIONED BY (Year, Month, GEO Summary, GEO Region, Price Category Code, Operating Airline)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY '\t'

STORED AS TEXTFILE

;

**Actual Partition Strategy of the dataset**

Partitioning gives effective results when,

1.There are limited number of partitions,

2.Comparatively equal sized partitions.

But this may not possible in all scenarios, like when are partitioning our files based geographic locations like country, some bigger countries will have large partitions where as small countries data will create small partitions. So, in these cases Partitioning will not be ideal.

  To overcome the problem of over partitioning, Hive provides Bucketing concept, another technique for decomposing table data sets into more manageable parts.

So here our actual partitioning strategy would be a little different than our initial partitioning strategy keeping in mind that the number of fields should not be too long.

We will doing both partitioning and bucketing here.

First we will create year wise folders. So here total 12 folder would be created corresponding to each mentioned year. Now for each year, 12 separate partitioning folder would be created referring to each of the 12 months in a year. Then inside each month’s folder we will have two partitioning separate folders indicating low fare and other price activity folders. Inside this, we will have two more partitioning folders such as domestic and international of geo summaries.

Now to achieve optimal partitioning, instead of further partitioning we will shift to bucketing by decomposing data or decreasing the data into more manageable parts or equal parts.

So here under each of the created domestic and international of geo summaries, we will subdivide **partition into** **clusters** or **buckets. Each cluster will consist of equal size of buckets of Geo Region along with Operating Airlines, terminal, boarding area,** Activity Type Code). This can be an actual partitioning strategy compared to initial one in order to achieve the optimal portioning.

Now the new hive command with partitioning and further bucketing becomes: -

**CREATE TABLE air traffic (Activity Period, Operating Airline, Operating Airline IATA Code, Published Airline, Published Airline IATA Código Summering Region, Activity Type Code, Price Category Code, Terminal, Boarding Area Passenger Count, Adjusted Activity Type Code, Adjusted Passenger Count, Year, Month)**

**PARTITIONED BY (Year, Month, Price Category Code, Geo Summaries)**

**CLUSTERED BY (Geo Region) INTO 9 BUCKETS**

**ROW FORMAT DELIMITED**

**FIELDS TERMINATED BY '\t'**

**STORED AS TEXTFILE**

**;**

Thank You

The End