Lab 8: Hive Bucketing

The usual preambles from the previous labs about having your repo up to date apply. The lab also uses the '/user/mara_dev/data' directory as a data staging area as in previous labs. The have commands are listed in text format in the commands.txt file in this lab directorty.

In the previous lab, you partitioned a table based on a specific column. However, if the distribution of values in that column is highly skewed, performance improvements with partitioning may be minimal. We can distribute the data into buckets which will provide a more even distribution than a partitioning might provide.

Data Setup

1. This lab will use the the same data as in previous labs
2. Upload the CSV file sampledatafoodsales.csv into the folder which he have used before: /user/maria_dev/data
3. The schema for the data that will be loaded into the table is:

```
orderdate   string;
region      string;
city        string;
category    string;
product     string;
quantity    int;
unitprice   double;
total       double;
```

Creating the Base Table

1. The base table is the data without bucketing.

```
create table foodbase(
      orderdate   string,
      region      string,
      city        string,
      category    string,
      product     string,
      quantity    int,
      unitprice   double,
      total       double)
row format delimited fields terminated by ','
lines terminated by '\n' stored as textfile;
```

2. Modify the table to skip the first header row when reading.

```
alter table foodbase set tblproperties("skip.header.line.count"="1");
```

3. Load the data

```
load data inpath'/user/maria_dev/data/sampledatafoodsales.csv'
    into table foodbase;
```

4. Check to see that the data has been loaded correctly.

Create a Bucket Table

1. Create a second table that tells Hive to bucket on the city column using three buckets

```
create table food2 (
      orderdate   string,
      region      string,
      city        string,
      category    string,
      product     string,
      quantity    int,
      unitprice   double,
      total       double)
clustered by (city) into 3 buckets
row format delimited fields terminated by ','
lines terminated by '\n' stored as textfile;
```

2. Insert the base table into the bucketed table.

```
insert insert into table food2 select * from foodbase;
```

3. Looking at the resulting directory, you can see three bucket directories based on cities. You can also see the buckets described in the output of

```
describe formatted food2;
```

More Bucket Tables

1. Now bucket the original data into two buckets based on product category..
2. Create the bucket table and insert the data

```
create table food3 (
      orderdate   string,
      region      string,
      city        string,
      category    string,
      product     string,
      quantity    int,
      unitprice   double,
      total       double)
clustered by (category) into 2 buckets
row format delimited fields terminated by ','
lines terminated by '\n' stored as textfile;

insert into food3 select * from foodbase;
```

3. Inspect this table to ensure the bucketing worked the same way you did the previous table.
4. Now create create a new bucketed table food 4 with three buckets based on quantity. The code for this is in the commands.txt file. Inspect the results.

Bucketing a Partitioned Table

Now partition by region and bucket by quantity. Inspect the result - it is easy to see that there are four buckets in each partition.

```
create table food5 (
      orderdate string,
      city string,
      category string,
      product string,
      quantity int,
      unitprice double,
      total double) partitioned by (region string)
      clustered by (quantity) into 4 buckets
      row format delimited fields terminated by ','
      lines terminated by '\n' stored as textfile;

insert into table food5 partition(region) select orderdate, city,
      category, product, quantity, unitprice, total, region
      from foodbase;
```