Lab # : 2; Lab Name : An Introduction to Hive ; Subject Name : Information Storage and Retrieval; Week #: 3; Lab Duration : 20 to 30 mins

Intro

In this lab, we will go over few features in Hive, which are useful for managing data in Hive. We will go over the following features:

- Load a data file into a Hive table
- Create a table using RC File Format
- Query a table using HQL
- Managed tables vs external tables
- Create a table using ORC File Format
- Create a table using Parquet File Format

Here are a few points to get to know Hive :

- Hive data warehouse software facilitates loading, querying and transforming data on top of mainly Hadoop.
- Hive is used for ETL processing.
- Hive is very much SQL like.
- Hive query executes MapReduce processes on a hadoop cluster.
- Hive store it's meta information in it's metastore.
- Hive enables ad-hoc querying
- Hive works great for batch processing
- Latency for Hive queries is usually high

Let's go!

Step-1. Load a data file into a Hive table

As hive stores data in HDFS, loading data into a Hive table primarily means loading data files into HDFS and mapping the Hive table definition to the content of the file/s.

On your server, you can type "hive" and you will go to hive prompt as hive is already available on your AMI/instance.

Now, let's create a table as follows:

```
CREATE TABLE Web_Session_Log
(DATETIME varchar(500),
USERID varchar(500),
SESSIONID varchar(500),
PRODUCTID varchar(500),
REFERERURL varchar(500))
row format delimited
fields terminated by '\t'
stored as textfile;
```

Now let's load some data into the table.

From the Amazon S3, pull the data file : https://s3.amazonaws.com/ucbdatasciencew205/labs/weblog_lab.csv into one of your local folders. After that run the following command to load data into the above Hive table.

```
LOAD  DATA  LOCAL INPATH  '/mnt/weblog_lab.csv'
OVERWRITE INTO TABLE Web_Session_Log;
```

The above command loaded the data file which is stored in your local unix directory into the hive table.

Step-2. Create a table using RC(Record Columnar) file format

As Hive reads datafiles stored in hdfs, to optimize the storage of data format, there are various ways to do so. RC File format is one format which has been very much used in Hive warehouses. RC File format is a structure which is a combination of data storage format, data compression approach, and optimization techniques for data reading. RC File format has the advantage of row oriented structure and column oriented structure both. Basically, you can slice and dice a table horizontally and vertically for storing. This gives the maximum flexibility to design your table based on access patterns.

Now, let's create a table with RC File Format:

```
CREATE TABLE Web_Session_Log_RC
(DATETIME varchar(500),
USERID varchar(500),
SESSIONID varchar(500),
PRODUCTID varchar(500),
REFERERURL varchar(500))
row format delimited
fields terminated by '\t'
STORED AS RCFILE;
```

Let's load data from the previously built Hive Table which was in Text File format.

```
INSERT OVERWRITE TABLE Web_Session_Log_RC

select * from Web_Session_Log;
```

Step-3. Query a tables

Let's query the same table. Here is an example query. You can create your own query and run as well.

```
SELECT SESSIONID,count(*) as count from Web_Session_Log_RC GROUP BY
SESSIONID ORDER BY count;
```

Step-4. Managed tables vs External tables

In Hive, there are two kinds of tables, Managed and External. Let's create these tables.

Managed:

```
CREATE TABLE Web_Session_Log_Managed
(DATETIME varchar(500),
USERID varchar(500),
SESSIONID varchar(500),
PRODUCTID varchar(500),
REFERERURL varchar(500))
row format delimited
fields terminated by '\t'
stored as textfile;
```

External:

```
CREATE EXTERNAL TABLE IF NOT EXIST Web_Session_Log_External
(DATETIME varchar(500),
USERID varchar(500),
SESSIONID varchar(500),
PRODUCTID varchar(500),
REFERERURL varchar(500))
row format delimited
fields terminated by '\t'
stored as textfile
LOCATION '<hdfs_location>';
```

You must add the datafile into hdfs first and point the external table to that location.

Note:
When you drop a Managed table, it deletes the data as well along with metadata.
When you drop an External table, it only deletes the metadata.

Step-5. ORC format

Now let's create a table with ORC format. ORC format is Optimized Row Columnar file format. It provides efficient way to store Hive data by doing better compression. This improves reading, writing and processing data in Hive.

```
CREATE TABLE ORCFileFormatExample(

DATETIME varchar(500),

USERID varchar(500),

SESSIONID varchar(500),

PRODUCTID varchar(500),

REFERERURL varchar(500))

COMMENT 'This is the Web Session Log data'

ROW FORMAT DELIMITED

FIELDS TERMINATED BY '\t'
```

STORED AS ORC tblproperties ("orc.compress"="GLIB");

Step-5. Parquet format

Here we will create a table with Parquet file format. This will give you another option to store your data in another efficient format. However, you can compare the above file formats by reading further.

```
CREATE TABLE ParqFileFormatExample(
DATETIME varchar(500),
USERID varchar(500),
SESSIONID varchar(500),
PRODUCTID varchar(500),
REFERERURL varchar(500))
COMMENT 'This is the Web Session Log data'
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\t'
STORED AS parquet;
```

Questions:

Q1 : Where data for a Managed table is stored?
Q2 : Will the data for an external table drop if the table is dropped?
Q3 : What is the difference between a RC and an ORC file format?
Q4 : In which case, you would use Parquet file?
Q5 : What are the compression types allowed while using Parquet file?