Big Data – cs522

Lab 6 – Day 14 (Tuesday)

- Submit your *own work* on time. No credit will be given if the assignment is submitted after the due date.
- Note that the completed lab should be submitted in .doc, .docx, .rtf or .pdf format only.

This document is divided into two parts.

1. Hive Practice Lab

   Just try to run through all the steps and see if they work properly for you.
   No need to submit this part.

   We'll look at the following things:

   - Create Database
   - Observe Warehouse and Metastore
   - Few Commands

2. Hive Homework Lab

   You need to submit a document (not .zip) wherein I should be able to find all the instructions and commands.

   Paste screenshots wherever applicable.

Hive Practice Lab

In this practice lab, we'll do some analysis on the weather dataset.

1. Get the simplified version of NCDC sample weather dataset and put the file in the path '/home/cloudera/cs522/input/weatherHive.txt'

2. Now we'll load this data into Hive's managed storage. Here we'll have Hive use the local filesystem for storage; later we'll see how to store tables in HDFS. So follow the steps below.

   i. Just like in RDBMS, Hive organizes its data into tables. We create a table to hold the weather data using the CREATE TABLE statement:

   ```
   CREATE TABLE records (year STRING, temperature INT, quality INT)
   ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t';
   ```

   The first line declares a records table with three columns: year, temperature, and quality. The type of each column must be specified, too. Hive expects there to be three fields in each row, corresponding to the table columns, with fields separated by tabs and rows by newlines.
   After executing this create table statement, check the Metastore and the Warehouse directory and verify that "records" is created.

   ii. Next, populate Hive with the sample weather data.

   ```
   LOAD DATA LOCAL INPATH '/home/cloudera/cs522/input/weatherHive.txt'
   OVERWRITE INTO TABLE records;
   ```

   Running this command tells Hive to put the specified local file in its warehouse directory. There is no attempt, for example, to parse the file and store it in an internal database format, because Hive does not mandate any particular file format.

   Tables are stored as directories under Hive's warehouse directory, which is controlled by the *hive.metastore.warehouse.dir* property and defaults to `/user/hive/warehouse`.

   Thus, the files for the `records` table are found in the */user/hive/warehouse/records* directory on the local filesystem:
   In this case, there is only one file, `weatherHive.txt`, but in general there can be more, and Hive will read all of them when querying the table.

3. Now that the data is in Hive, we can run a query against it:

```
hive> SELECT year, MAX(temperature) FROM records WHERE temperature !=
9999 AND quality IN (0, 1, 4, 5)  GROUP BY year;
```

This SQL query is unremarkable. It is a SELECT statement with a GROUP BY clause for grouping rows into years, which uses the MAX aggregate function to find the maximum temperature for each year group.

The remarkable thing is that Hive transforms this query into a job, which it executes on our behalf, then prints the results to the console. It is the ability to execute SQL queries against our raw data that gives Hive its power.

Hive Homework Lab

In this HW lab, you need to analyze the city of Chicago employees' dataset.

- Get the data set from the following location:

  https://data.cityofchicago.org/Administration-Finance/Current-Employee-Names-Salaries-and-Position-Title/xzkq-xp2w

- Download sample data for at least 3 departments in CSV format.
- Create external (not managed) table in Hive for this sample data set and draw some interesting facts from this data like for example, what's the maximum salary of employees in each department.

Show at least 3 different analysis with "limited" rows.

What to submit :

1. All the HQL statements used to complete the above requirements.
2. Screenshots of the step by step process also showing the output.