There are 2 types of tables in Hive, Internal and External.

Creating Internal Table

Internal table are like normal database table where data can be stored and queried on. On dropping these tables the data stored in them also gets deleted and data is lost forever. So one should be careful while using internal tables as one drop command can destroy the whole data. Open new terminal and fire up hive by just typing hive. Create table on weather data.

CREATE TABLE weather (wban INT, date STRING, precip INT)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ‘,’

LOCATION ‘ /hive/data/weather’;

ROW FORMAT should have delimiters used to terminate the fields and lines like in the above example the fields are terminated with comma (“,”). The default location of Hive table is overwritten by using LOCATION. So the data now is stored in data/weather folder inside hive.

Load the Data in Table

Data can be loaded in 2 ways in Hive either from local file or from HDFS to Hive. To load the data from local to Hive use the following command in NEW terminal:

hadoop fs –copyFromLocal /home/user/data/weather/2012.txt hdfs://hname:10001/hive/data/weather

Here the hdfs path was initially made in the create statement using LOCATION ‘ /hive/data/weather’.

Another way to load data is to load it from HDFS to hive using the following command:

LOAD DATA INPATH ‘hdfs:/data/2012.txt’ INTO TABLE weather;

Views

Views are used for creating virtual tables. They are faster than creating actual tables and they can work as table while using them in any other query. For example precipitation view can be made as follows and can be used in other query just like a table is used.

CREATE VIEW AS precipitation\_data SELECT \* FROM weatherext WHERE precip IS NOT NULL AND precip>0;

Following query can be used to retrieve data from precipitation\_data

SELECT \* FROM precipitation\_data;

Indexing

Indexes are made on top of tables so that they speed up queries. Most popular column that are used very often in WHERE clause should be indexed to make the query run faster. Partition can be built on weather table’s date column in following way:

CREATE INDEX date\_index ON TABLE weather (date) AS ‘COMPACT’ WITH REBUILD;

After making this index any query that uses date column of weather table will be faster than running it before creating index.

Learn Hadoop by working on interesting Big Data and Hadoop Projects for just $9

Drop table

On dropping the table loaded by second method that is from HDFS to Hive, the data gets deleted and there is no copy of data on HDFS. This means that on creating internal table the data gets moved from HDFS to Hive. Table can be dropped using:

DROP TABLE weather;

Hive: External Tables

Creating external table

Open new terminal and fire up hive by just typing hive. Create table on weather data.

CREATE EXTERNAL TABLE weatherext ( wban INT, date STRING)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ‘,’

LOCATION ‘ /hive/data/weatherext’;

ROW FORMAT should have delimiters used to terminate the fields and lines like in the above example the fields are terminated with comma (“,”). The default location of Hive table is overwritten by using LOCATION. So the data now is stored in data/weatherext folder inside hive.

Load the data in table

Load the data from HDFS to Hive using the following command:

LOAD DATA INPATH ‘hdfs:/data/2012.txt’ INTO TABLE weatherext;

Partitioning of table

Hive stores tables in partitions. Partitions are used to divide the table into related parts. Partitions make data querying more efficient. For example in the above weather table the data can be partitioned on the basis of year and month and when query is fired on weather table this partition can be used as one of the column.

CREATE EXTERNAL TABLE IF NOT EXSISTS weatherext ( wban INT, date STRING)

PARTITIONED BY (year INT, month STRING)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ‘,’

LOCATION ‘ /hive/data/weatherext’;

Loading data in partitioned tables is different than non-partitioned one. There is little manual work of mentioning the partition data. Data can be loaded in partition, year 2012 and month 01 and 02 as follows:

LOAD DATA INPATH ‘hdfs:/data/2012.txt’ INTO TABLE weatherext PARTITION (year=2012, month=’01’);

LOAD DATA INPATH ‘hdfs:/data/2012.txt’ INTO TABLE weatherext PARTITION (year=2012, month=’02’);

This creates the partitioned table and makes different folder for each partition which helps in querying data.

Querying of partitioned table

Partitioned tables can use partition parameters as one of the column for querying. To retrieve all the data for month of ‘02’ following query can be used on weather table.

SELECT \* FROM weatherext WHERE month = ‘02’;

Drop table

On dropping the external table, the data does not get deleted from HDFS. Thus it is evident that the external table are just pointers on HDFS data. Table can be dropped using:

DROp TABLE weather;