**Types of table:-**

In Hive, You can create 2 kinds of table.

1. Managed table or Internal Table
2. External Table

**Managed/Internal Table**

The table which maintains the data with themselves are called Managed or Internal table. The data will be managed by hive, and if you delete the table you will lose the data.

The table which we created till now, comes under internal table. Lets create a database in Hive & work with Internal table first

hive>create database financials;

hive>set hive.cli.print.current.db=true;

hive (default)>use financials;

hive (financials)>

**Show the current database name in hive terminal**

hive> set hive.cli.print.current.db=true;

**Create an internal table in hive**

hive>create table NYSE\_DAILY\_PRICES\_HIVE\_INT

(

 stock\_exchange string,

 stock\_symbol string,

 tdate timestamp,

 stock\_price\_open DOUBLE,

 stock\_price\_high DOUBLE,

 stock\_price\_low DOUBLE,

 stock\_price\_close DOUBLE,

 stock\_volume bigint,

 stock\_price\_adj\_close DOUBLE

) row format delimited fields terminated by ',';

**Make sure you extract nyse.7z file(present along with this document) to /home/hduser**

**Create a directory in hdfs to copy nyse data into it**

**$**  hadoop fs -mkdir /mystockdata1

**copy all the relevant data file from /home/hduser/nyse directory to /mystockdata1 created in above step.**

**$** hadoop fs -put nyse/NASDAQ\_daily\_prices\_\*.csv /mystockdata1

Delete the NASDAQ\_daily\_prices\_0.csv file from HDFS in case it has got copied. This file just contains metadata ie (column information) & it has nothing to do with data.

**$** hadoop fs -rm /mystockdata1/NASDAQ\_daily\_prices\_0.csv

**Load the data from hdfs to hive table**

**hive (financials)>**load data inpath '/mystockdata1/NASDAQ\_daily\_prices\_\*.csv' overwrite into table NYSE\_DAILY\_PRICES\_HIVE\_int;

**Note:**

1. The above command will move all the datafiles from /mystockdata1 to /user/hive/warehouse/financials.db/nyse\_daily\_prices\_hive\_int directory. Later when you drop the table, you will lose the data. This is called internal table or Managed table.
2. Hive by default uses derby database to maintain the table schema. This derby database is created in the local present working directory from where you launched hive terminal. The table related metadata is not stored in HDFS.

**Select total records present in NYSE\_DAILY\_PRICES\_HIVE table**

hive(financials)>select count(\*) from **NYSE\_DAILY\_PRICES\_HIVE\_int;**

**Find out the first 5 symbol with its volume**

**hive (financials)>**select stock\_symbol,max(stock\_volume) from nyse\_daily\_prices\_hive\_int group by **stock\_symbol** limit 5;

AACC 2154200

AAME 581500

AAPL 265069000

AAUK 1822100

AAWW 2972200

**External Table**

These are the tables where data is not maintained by hive. The data will not be present in /user/hduser/warehouse/table directory. The data will be maintained somewhere else in HDFS and will just be referred by Hive. When you drop the table, the data will not be deleted from its location. Its just the metadata that gets deleted from metastore folder.

**How To Create an External Hive table**

1. You need to use External keyword to create an External Hive table.
2. While creating external table you can specify the location while defining the table itself, or you may create the external table without specifying the location now, and later you can load the data into that table.

**CREATING AN EXTERNAL TABLE WITH LOCATION**

hive (financials) >

create  external table NYSE\_DAILY\_PRICES\_HIVE\_EXT1

(

stock\_exchange string,

stock\_symbol string,

tdate timestamp,

stock\_price\_open DOUBLE,

stock\_price\_high DOUBLE,

stock\_price\_low DOUBLE,

stock\_price\_close DOUBLE,

stock\_volume bigint,

stock\_price\_adj\_close DOUBLE

)row format delimited fields terminated by ','

location '/mystockdata1';

**Note:**

When You create an external table by specifying the location, hive will not create the table related directory inside your warehouse. This means You will not see /user/hive/warehouse/financials.db/NYSE\_DAILY\_PRICES\_HIVE\_EXT1 directory because hive is not going to manage any data inside it, so there is no point of creating a directory and wasting the memory.

When you drop the table, the data will not be deleted, It's just the metadata that gets deleted from metastore.

**CREATING AN EXTERNAL TABLE WITHOUT LOCATION**

hive (financials) >

create  external table NYSE\_DAILY\_PRICES\_HIVE\_EXT2

(

stock\_exchange string,

stock\_symbol string,

tdate timestamp,

stock\_price\_open DOUBLE,

stock\_price\_high DOUBLE,

stock\_price\_low DOUBLE,

stock\_price\_close DOUBLE,

stock\_volume bigint,

stock\_price\_adj\_close DOUBLE

)

row format delimited fields terminated by ',';

When you create a table without location ,then you need to load the data into it. Perform below operation to do that.

**hive ( financials ) >**load data inpath '/mystockdata1/NASDAQ\_daily\_prices\_\*.csv' overwrite into table NYSE\_DAILY\_PRICES\_HIVE\_EXT2;

External Table without location == internal table(managed table)

**Note:**

In this case, we have created an External table but without providing any location. Since the location was not specified while defining the table, hive will create a table related folder inside warehouse directory.

This means /user/hive/warehouse/financials.db/NYSE\_DAILY\_PRICES\_HIVE\_EXT2 folder will be created.

Later when you load the data into this External table(without location),the data files will be moved from hdfs to warehouse/financials.db/NYSE\_DAILY\_PRICES\_HIVE\_EXT2 directory. When you delete this table or directory, you will lose the data.

**Important Points**

If we have data in HDFS and if you create internal table then data will be moved and if you drop table, data will be deleted.

If we have data in HDFS and if you create external table without location, then data will be moved into warehouse folder, and data will be deleted when you drop a table.

If we have data in HDFS and if you create external table with location, then data will not be moved into warehouse folder, and data will not be deleted when you drop a table.