TABLES IN HIVE

Hive Internal Tables (or) managed Tables:

- The internal tables are also called managed tables as the lifecycle of their data is controlled by the Hive.
- By default, these tables are stored in a subdirectory under the directory defined by hive.metastore.warehouse.dir (i.e. /user/hive/warehouse).
- If we try to drop the internal table, Hive deletes both table schema and data.
  - Let's create an internal table by using the following command:-

```
CREATE TABLE test(
id INT,
Name STRING)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ',';
```

```
hive> CREATE TABLE test(
    > id INT,
    > Name STRING)
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY ',';
OK
Time taken: 0.992 seconds
hive>
```

Step 2: load data to this table **test** with the below command.

```
LOAD DATA LOCAL INPATH '/home/dikshant/Desktop/data.csv' INTO TABLE test;
```

```
hive> LOAD DATA LOCAL INPATH '/home/dikshant/Desktop/data.csv' INTO TABLE test;
Loading data to table default.test
OK
Time taken: 1.041 seconds
hive>
```

Step 3: Let's see whether the data is loaded into the table or not

```
select * from test;
```

```
hive> select * from test;
OK
1       dikshant
2       rajesh
Time taken: 0.094 seconds, Fetched: 2 row(s)
hive>
```

Step 4: We can describe the table to see it is Internal or External

```
describe extended test;
```

```
hive> describe extended test;
OK
id                      int
name                    string

Detailed Table Information      Table(tableName:test, dbName:default, owner:diks
hant, createTime:1608806135, lastAccessTime:0, retention:0, sd:StorageDescriptor
(cols:[FieldSchema(name:id, type:int, comment:null), FieldSchema(name:name, type
:string, comment:null)], location:hdfs://localhost:9000/user/hive/warehouse/test
, inputFormat:org.apache.hadoop.mapred.TextInputFormat, outputFormat:org.apache.
hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat, compressed:false, numBuckets:-1
, serdeInfo:SerDeInfo(name:null, serializationLib:org.apache.hadoop.hive.serde2.
lazy.LazySimpleSerDe, parameters:{serialization.format=,, field.delim=,}), bucke
tCols:[], sortCols:[], parameters:{}, skewedInfo:SkewedInfo(skewedColNames:[], s
kewedColValues:[], skewedColValueLocationMaps:{}), storedAsSubDirectories:false)
, partitionKeys:[], parameters:{totalSize=20, numRows=0, rawDataSize=0, numFiles
=1, transient_lastDdlTime=1608806284, bucketing_version=2}, viewOriginalText:nul
l, viewExpandedText:null, tableType:MANAGED_TABLE, rewriteEnabled:false, catName
:hive, ownerType:USER)
Time taken: 0.229 seconds, Fetched: 4 row(s)
hive>
```

Hive External Tables:

The external table allows us to create and access a table and a data externally. The **external** keyword is used to specify the external table, whereas the **location** keyword is used to determine the location of loaded data.

As the table is external, the data is not present in the Hive directory. Therefore, if we try to drop the table, the metadata of the table will be deleted, but the data still exists.

1. Create an External Table then insert data directly.
2. Create an External Table and load the data from Local file System.
3. Create an External Table and load the data from HDFS.

Method 1:

hive> create EXTERNAL table employee_external(eid int,name string, salary string, designation string) ROW FORMAT DELIMITED FIELDS TERMINATED BY ‘,’ ;

Step2: Insert the data directly

Hive> insert into employee_external values(111,’xyz’,’50000’,’manager’);

EXAMPLE:

```
create EXTERNAL table employe_1(eid int, name String, salary
String, designation String)
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY ','
    > ;
```

```
insert into employe_1 values(111, 'xyz', '5000', 'Manager');
```

To display the output, use the command: select * from employe_1;

Method 2:

Hive> create EXTERNAL table employee_external(eid int,name string, salary string, designation string) ROW FORMAT DELIMITED FIELDS TERMINATED BY ‘,’ ;

Hive> LOAD DATA LOCAL INPATH ‘/home/cloudera/training/hivedata/emp.txt’ into TABLE employee_external;

** select count(*) from employee_external; o/p: 10

Hive> LOAD DATA LOCAL INPATH ‘/home/cloudera/training/hivedata/emp.txt’ into TABLE employee_external;

** select count (*) from employee_external; o/p:20 (data will be doubled)

**** if we load the data again and again into the same table,then the data will be increased and also duplicate values also increased.so at that time we have to use OVERWRITE command.

Hive> LOAD DATA LOCAL INPATH ‘/home/cloudera/training/hivedata/emp.txt’ OVERWRITE INTO TABLE employee_external;

EXAMPLE:

Hive>
```
CREATE  TABLE EMPLOYE (EMPLOYEE_ID Int,
FIRST_NAME String,
LAST_NAME String,
EMAIL String,
PHONE_NUMBER String,
HIRE_DATE String,
JOB_ID String,
SALARY Int,
COMMISSION_PCT String,
MANAGER_ID Int,
DEPARTMENT_ID Int)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
TBLPROPERTIES("skip.header.line.count"="1");
```

Now the EMPLOYE table has been created

```
load data local inpath
'/home/nishanthnishu24025349/bigdata/employees.csv' into table
employe;
```

/home/nishanthnishu24025349/bigdata/employees.csv (i.e. This is local file system i.e.. Unix files)
This file should be placed with single quotes.

Method 3:

Hive> create EXTERNAL table employee_external(eid int,name string, salary string, designation string) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' ;

Hive> LOAD DATA INPATH '/mydata/emp.txt' into table employee_external;

EXAMPLE:

```
create EXTERNAL table employe_1(eid int, name String, salary
String, designation String)
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY ','
    > ;
```

Hive> `load data  inpath '/home/nishanthnishu24025349/bigdata/employees.csv' into table employee_external;`

"Row format delimited" is used to store the data in the hdfs row per line.

"fields terminated by" is used to store the data in the hdfs where the columns are separated by the specified character. Ex: '\t' (tab space), ' , '(comma)

We created table mentioning property *"skip.header.line.count"="1"*, since we want to skip the top 1 lines from the file and *"skip.footer.line.count"="1"*, since we want to skip bottom 1 lines from the file.

We created table mentioning property *"skip.header.line.count"="2"*, since we want to skip the top 2 lines from the file and *"skip.footer.line.count"="2"*, since we want to skip bottom 2 lines from the file.

**Data will be available at either HDFS or LOCAL Fule System but on top of data,we can create external tables.

**External tables will be stored on HDFS.
Internal(or) Managed tables will be stored Local File system.

**when we are creating the table while mentioning the location. I.e. telling that where to store that table in which path.(what location we are mentioning in that location only the data will be stored.)

** To show the data along with header line, use this command.
set.hive.cli.print.header=true;

**in external tables, when we delete the table only the table will be deleted but not the data. The data will remain the same.