**Practical 4**

**Title:**Write an application using Hbase and HiveQl for Flight information system which will include

1.Creating,Dropping and altering database tables

2.Creating an external Hive table to connect to the Hbase for customer information table

3.Load table with data,insert new values and field in the table,join tables with hive

4.Create index on flight information table

5.Find the average deparature delay per day in 2008.

**Objective**:

To study how to create internal and external table in Hive.

**Theory:**

**Types of Table**

Managed table and External table in Hive

There are two types of tables in Hive ,one is Managed table and second is external table.  
the difference is , when you drop a table, if it is managed table hive deletes both data and meta data,if it is external table Hive only deletes metadata.

by default It is Managed table .  
If you want to create a external table ,you will use external keyword

As mentioned above, Hive has two types of tables:

* Managed table
* External table

## Managed table

Managed table is also called as Internal table. This is the default table in Hive. When we create a table in Hive without specifying it as external, by default we will get a Managed table.

If we create a table as a managed table, the table will be created in a specific location in HDFS.

By default, the table data will be created in **/usr/hive/warehouse**directory of HDFS

If we delete a Managed table, both the table data and meta data for that table will be deleted from the HDFS.

## External Table

External table is created for external use as when the data is used outside Hive. Whenever we want to delete the table’s meta data and we want to keep the table’s data as it is, we use External table. External table only deletes the schema of the table.

Create table:

> Create table cust\_info(cust\_id int,name string,deptno int,address string,salary int)

> row format delimited

> fields terminated by ‘ ;’

> desc cust\_info

Select the table

>select \*from cust\_info;

For loading data

>load data local inpath ‘/home/mcoerc/Desktop/cust\_info’ into table cust\_info;

For altering data

>alter table cust\_info change cust\_id id string;

Load second table

>load data local inpath ‘/home/mcoerc/Desktop/cust\_info1’ into table cust\_info;

To create database:

>create database custdb

>use default

>show database //to display all databases

On second terminal

>su –hdpusr1

>password:

>hdfs dfs-mkdir/mcoerchivddata

>hdfs dfs –put/home/mcoerc/desktop/cust\_info\*/mcoerchivedata

On main terminal

Create external table:

>Create external table cust\_info2(custid string,name string,deptno int,dob string,salary int,post string)

>row format delimited

>fields terminated by ‘;’

>location ‘mcoerchiveddata’

>desc cust\_info2

>Select \*from cust\_info2

Back to the second terminal

>Hdfs dfs/home/mcoerc/Desktop/cudt\_info/mcoerchiveddata

Join the table:

>select \*from cust\_info; //internal table

>select \*from cust\_info2; //External table

>select c.name,c.id from cust\_info join cust\_info2 c2 on (c.deptno=c2.deptno)

To calculate average:

>Select avg(salary) from cust\_info;

For creating index:

>Create index id1 on table cust\_info(cust\_id,name) as ‘org.apache.hadoop.hive.ql.index.compact.CompactIndexHandler’ with DEFERRED REBUILD;

>show formatted index on cust\_info; //display index

**Hbase :**

**There can be two cases while creating a Hive table on top of HBase:**

1. We do not know the column names or need all the columns for which we could explode all the data into a map as key value pairs.
2. We need only specific columns in which case we need to specify the mappings for every column.

Let’s look at an example:

The first step is to create a sample HBase table.

create ‘cust\_info’, {NAME=>’cust\_name’, VERSIONS => ’3′, COMPRESSION => ‘NONE’, TTL => ’2147483647′, BLOCKSIZE => ’65536′, IN\_MEMORY => ‘false’, BLOCKCACHE => ‘false’}  
describe ‘cust\_info’  
enable ‘cust\_info’

The next step is to insert some sample data into cust\_info

put ‘cust\_info’ , ’1′ ,’cust\_name:bhimrao′ ,’001′  
put ‘cust\_info’ , ’2′ ,’cust\_name:aruna′ ,’002′  
put ‘cust\_info’ , ’3′ ,’cust\_name:abdul′ ,’003′  
put ‘cust\_info’ , ’4′ ,’cust\_name:vinoba′ ,’004′  
put ‘cust\_info’ , ’5′ ,’cust\_name:gopinath′ ,’005′  
put ‘cust\_info’ , ’6′ ,’cust\_name:mother teresa′ ,’006′

As mentioned above we can create a Hive external table in two ways:

Use all columns:

CREATE EXTERNAL TABLE cust\_name\_all (id string,colname map<string,string>)  
STORED BY ‘org.apache.hadoop.hive.hbase.HBaseStorageHandler’  
WITH SERDEPROPERTIES (“hbase.columns.mapping” = “:key,cust\_name:”)  
TBLPROPERTIES(“hbase.table.name” = “cust\_info”);

hive> SELCT \* FROM cust\_name\_all;  
1 {“bhimroa″:”001″}  
2 {“aruna″:”002″}  
3 {“abdul″:”003″}  
4 {“vinoba″:”004″}  
5 {“gopinath″:”005″,”mother teresa″:”006″}

or map every column by name  
CREATE EXTERNAL TABLE cust\_name\_map(id string,colname1 string,colname2 string)  
STORED BY ‘org.apache.hadoop.hive.hbase.HBaseStorageHandler’  
WITH SERDEPROPERTIES (“hbase.columns.mapping” = “:key,cust\_name:mydata1,cust\_name:mydata5″)  
TBLPROPERTIES(“hbase.table.name” = “cust\_info”);

hive> SELECT \* FROM cust\_name\_map;  
1 001 NULL  
5 001 005

**Conclusion**: Hence we studied how to create internal and external tables and perform operations on table.