```
CREATE DATABASE
hive>  CREATE DATABASE IF NOT EXISTS userdb;
OK
Time taken: 0.145 seconds
hive>

SHOW DATABASES
hive> show databases;
OK
default
test_demo
userdb
Time taken: 0.328 seconds, Fetched: 3 row(s)

DROP DATABSE
hive> DROP DATABASE IF EXISTS userdb;
OK
Time taken: 0.366 seconds
hive> SHOW DATABASES;
OK
default
test_demo
Time taken: 0.179 seconds, Fetched: 2 row(s)

USE DATABASE
hive> use test_demo;
OK
Time taken: 0.081 seconds
```

```
CREATEV TABLE IN HIVE
hive> CREATE TABLE IF NOT EXISTS employee2(eid int,name String,salary
String,designation String)
    ROW FORMAT DELIMITED
    FIELDS TERMINATED BY ','
    LINES TERMINATED BY '\n'
    STORED AS TEXTFILE;
OK
Time taken: 0.058 seconds


LOAD DATA FROM FILE INTO TABLE
hive> load data local inpath '/home/cloudera/Documents/sample_emp.txt' overwrite
into table employee2;


Loading data to table default.employee2
Table default.employee2 stats: [numFiles=1, numRows=0, totalSize=162,
rawDataSize=0]
OK
Time taken: 0.255 seconds

DISPLAY DATA FROM TABLE
hive> select * from employee2;
OK
1201 Gopal    45000    Technical manager
1202 Manisha  45000    Proof reader
```

```
1203 Masthanvali  40000    Technical writer
1204 Kiran    40000    Hr Admin
1205 Kranthi  30000    Op Admin
NULL NULL NULL NULL
Time taken: 0.049 seconds, Fetched: 6 row(s)

ALTER TABLE IN HIVE
hive> ALTER TABLE employee2 CHANGE name empname String;
OK
Time taken: 0.149 seconds

hive> ALTER TABLE employee2 CHANGE salary salary Double;
OK
Time taken: 0.11 seconds
hive> describe employee2;
OK
eid                     int
empname                 string
salary                  double
designation             string
Time taken: 0.105 seconds, Fetched: 4 row(s)

hive> alter table employee2 add columns ( dept string comment'Department Name');
OK
Time taken: 0.083 seconds

DESCRIBE TABLE IN HIVE
hive> describe employee2;
OK
```

```
eid                     int
empname                 string
salary                  double
designation             string
dept                    string                  Department Name
Time taken: 0.056 seconds, Fetched: 5 row(s)
```

DROP TABLE IN HIVE

```
hive> drop table employee;
OK
Time taken: 0.464 seconds

hive> drop table employee1;
OK
Time taken: 0.129 seconds
```

SHOW TABLE IN HIVE

```
hive> show tables;
OK
airports
employee2
flight_data
Time taken: 0.022 seconds, Fetched: 3 row(s)
```

CREATE EXTERNAL TABLE IN HIVE

External Table

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.

To create an external table, follow the below steps: -
Let's create a directory on HDFS by using the following command: -

```
>hdfs dfs -mkdir /HiveDirectory
```

Now, store the file on the created directory.

```
>hdfs dfs -put hive/emp_details /HiveDirectory
```

Let's create an external table using the following command: -

```
hive> create external table emplist (Id int, Name string , Salary float)
row format delimited
 fields terminated by ','
location '/HiveDirectory';
```

Hive Create Table

Now, we can use the following command to retrieve the data: -

```
select * from emplist;
```

FLIGHT INFORMATION SYSTEM ANALYSIS USING HIVE

Datasets

There are 2 datasets in the repo.

a) The first dataset contains on-time flight performance data from 2008, originally released by Research and Innovative Technology Administration (RITA). The source of this dataset is http://stat-computing.org/dataexpo/2009/the-data.html.
Link for 2008.csv dataset:
https://github.com/markgrover/cloudcon-hive/blob/master/2008.tar.gz?raw=true

b) The second dataset contains listing of various airport codes in continental US, Puerto Rico and US Virgin Islands. The source of this dataset is http://www.world-airport-codes.com/ The data was scraped from this website and then cleansed to be in its present CSV form.
link for airports.csv:
https://github.com/markgrover/hive-impala-bdtc/blob/master/airports.csv

1. On hive shell: Create hive table, flight_data:

```
CREATE TABLE flight_data(
   year INT,
   month INT,
   day INT,
   day_of_week INT,
   dep_time INT,
   crs_dep_time INT,
   arr_time INT,
   crs_arr_time INT,
   unique_carrier STRING,
   flight_num INT,
   tail_num STRING,
   actual_elapsed_time INT,
   crs_elapsed_time INT,
   air_time INT,
   arr_delay INT,
   dep_delay INT,
   origin STRING,
   dest STRING,
   distance INT,
   taxi_in INT,
   taxi_out INT,
   cancelled INT,
   cancellation_code STRING,
   diverted INT,
   carrier_delay STRING,
   weather_delay STRING,
   nas_delay STRING,
   security_delay STRING,
```

```
   late_aircraft_delay STRING
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ',';
```

2.Load the data into the table:

```
LOAD DATA LOCAL INPATH '/home/cloudera/2008.csv' OVERWRITE INTO TABLE flight_data;
```

3.Ensure the table got created and loaded fine:

```
SHOW TABLES;
SELECT
   *
FROM
   flight_data
LIMIT 10;
```

4.Query the table. Find average arrival delay for all flights departing SFO in January:

```
SELECT
   avg(arr_delay)
FROM
   flight_data
WHERE
```

```
   month=1
   AND origin='SFO';
```

5. On hive shell: create the airports table

```
CREATE TABLE airports(
   name STRING,
   country STRING,
   area_code INT,
   code STRING)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ',';
```

6.Load data into airports table:

```
LOAD DATA LOCAL INPATH 'hive-impala-bdtc/airports.csv' OVERWRITE INTO TABLE
airports;
```

7.On hive shell, list some rows from the airports table:

```
SELECT
   *
FROM
   airports
LIMIT 10
```

8.On hive shell: run a join query to find the average delay in January 2008 for each airport and to print out the airport's name:

```
SELECT
   name,
   AVG(arr_delay)
FROM
   flight_data f
   INNER JOIN airports a
   ON (f.origin=a.code)
WHERE
   month=1
GROUP BY
   name;
```

9.Find average departure delay per day in 2008

```
hive>select day,avg(dep_delay) from flight_data group by day;
```

10.Create Index on Flight Information System Table

```
hive> CREATE INDEX origin_index ON TABLE flight_data (origin) AS
    > 'COMPACT' WITH DEFERRED REBUILD;
OK
Time taken: 0.388 seconds
SHOW CREATED INDEX

hive> CREATE INDEX origin_index ON TABLE flight_data (origin) AS
```

```
    > 'COMPACT' WITH DEFERRED REBUILD;
OK
Time taken: 0.388 seconds
```