EXPERIMENT 3: HIVE

Aim: To understand Data Processing Tool – Hive and HQL (Hive query language)

Objectives:

1. **Create Managed and External tables in HIVE**
2. **Load data in HIVE table from Local File System**
3. **Load data in HIVE table from HDFS**
4. **Query data sets using Hive QL**
5. **Create partitions and buckets**

Key concept:

- Hive is a Data warehousing tool in Hadoop ecosystem.
- HIVE Facilitates reading, writing, and managing large datasets residing in distributed storage and queried using SQL syntax.
- It is used for analyzing structured and semi-structured data. Hive abstracts the complexity of Hadoop MapReduce. Basically, it provides a mechanism to project structure onto the data and perform queries written in HQL (Hive Query Language) that are similar to SQL statements.
- Internally, these queries or HQL gets converted to map reduce jobs by the Hive compiler. Therefore, you don’t need to worry about writing complex MapReduce programs to process your data using Hadoop.
- Tools to enable easy access to data via SQL, thus enabling data warehousing tasks such as extract/transform/load (ETL), reporting, and data analysis.
- A mechanism to impose structure on a variety of data formats.
- There is not a single "Hive format" in which data must be stored. Hive comes with built in connectors for comma and tab-separated values (CSV/TSV) text file etc.
- Hive is not designed for online transaction processing (OLTP) workloads. It is best used for traditional data warehousing tasks.

Q1: How to enter the HIVE Shell?

Go to the Terminal and type hive, you will see the hive on the prompt.

```
[cloudera@quickstart Desktop]$ hive
```

Q2: Create a database

```
create database emp_details;

use emp_details;
```

Q3: How to create Managed Table in HIVE?

```
create table emp(empno int, ename string, job string, sal int,
deptno int)
row format delimited fields terminated by ',';
```

Q4: How to load the data from LOCAL to HIVE TABLE

Suppose you created a comma separated file in local system named empdetails.txt

```
1,A,clerk,4000,10
2,A,clerk,4000,30
3,B,mgr,8000,20
4,C,peon,2000,40
5,D,clerk,4000,10
6,E,mgr,8000,50
```

```
hive> LOAD DATA LOCAL INPATH
'/home/cloudera/Desktop/empdetails.txt' OVERWRITE INTO TABLE
emp;
```

# Note: If 'LOCAL' is omitted then it looks for the file in HDFS.

The keyword 'OVERWRITE' signifies that existing data in the table is deleted. If the 'OVERWRITE' keyword is omitted, data files are appended to existing data sets.

Q5: How to check where the managed table is created in hive db

```
[cloudera@quickstart Desktop]$ hadoop fs -ls
/user/hive/warehouse/emp_details.db

Found 2 items
drwxrwxrwx   - cloudera supergroup          0 2018-07-24 02:40
/user/hive/warehouse/emp_details.db/emp
drwxrwxrwx   - cloudera supergroup          0 2018-07-24 02:28
/user/hive/warehouse/emp_details.db/emp1
```

Also check the contents inside emp:

```
[cloudera@quickstart Desktop]$ hadoop fs -ls
/user/hive/warehouse/emp_details.db/emp

Found 1 items
-rwxrwxrwx   1 cloudera supergroup        104 2018-07-24 02:40
/user/hive/warehouse/emp_details.db/emp/empdetails.txt
```

Now see the contents inside empdetails.txt

```
[cloudera@quickstart Desktop]$ hadoop fs -cat
/user/hive/warehouse/emp_details.db/emp/empdetails.txt

,A,clerk,4000,10
2,A,clerk,4000,30
3,B,mgr,8000,20
4,C,peon,2000,40
5,D,clerk,4000,10
6,E,mgr,8000,50
```

Q6:Check the schema of the created table emp?

```
describe emp;
```

For a detailed schema use:

```
describe extended emp;
```

Q7: How to see all the tables present in database

```
show tables;
```

Q8: Select all the enames from emp table

```
select ename from emp;
```

Q9: Get the records where name is 'A'

```
select * from emp where ename='A';
```

Q10: Count the total number of records in the created table

Count aggregate function is used count the total number of the records in a table.

```
select count(1) from emp;
OR
Select count(*) from emp;
```

Q11: Group the sum of salaries as per the deptno

```
select deptno, sum(sal) from emp group by  deptno;
```

Q12: Get the salary of people between 1000 and 2000

```
select * from emp  where sal between 1000 and 2000;
```

Q13: Select the name of employees where job has exactly 5 characters

```
hive> select ename from emp where job LIKE '_____';
```

Q14: List the employee names where job has l as the second character

```
hive> select ename from emp where job LIKE '_l%';
```

Q15: Retrieve the total salary for each department

```
select deptno, sum(sal) from emp group by deptno;
```

Q16: Add a column to the table

```
alter table emp add COLUMNS(lastname string);
```

Q17: How to Rename a table

```
alter table emp rename to emp1;
```

Q18: How to drop table

```
drop table emp;
```

Q19: How to create External Table:

Syntax:

```
CREATE EXTERNAL TABLE <table_name> (column1 data_type, column2 data_type)
row format delimited fields terminated by ','
LOCATION '<table_hive_location>';
```

Eg. I have created a comma separated file in local machine called extdata.txt

```
1,2,3
4,5,6
```

Then I have copied this file in HDFS

```
[cloudera@quickstart Desktop]$ hadoop fs -put extdata.txt /user/cloudera/
```

Then I copied this file in a directory named hivedata

```
[cloudera@quickstart Desktop]$ hadoop fs -cp extdata.txt hivedata
```

Then I created a table ext1 and loaded it with the data

```
hive> create external table ext1(a int, b int, c int)
    > row format delimited fields terminated by ','
    > LOCATION '/user/cloudera/hivedata'
    > ;
```

Now check if table is populated with data

```
hive> select * from ext1;
```

NOTE: You will not see this external table in the location /user/hive/warehouse/emp_details.db as you saw in case of managed table, this is because external table is created by referring the data to the location where txt file is there and not by loading it in the hive table.

Moreover if you drop the managed table all the data will be lost in location /user/hive/warehouse/emp_details.db where as in case of external data your data will still remain in hdfs.

```
[cloudera@quickstart Desktop]$ hadoop fs -ls
/user/hive/warehouse/emp_details.db
Found 2 items
drwxrwxrwx   - cloudera supergroup          0 2018-07-24 02:40
/user/hive/warehouse/emp_details.db/emp
drwxrwxrwx   - cloudera supergroup          0 2018-07-24 02:28
/user/hive/warehouse/emp_details.db/emp1
```

In above output we saw that two managed tables only being seen and not ext1 which is an external table.

If you drop the external table we will not lose the data in hdfs as shown below.

```
[cloudera@quickstart Desktop]$ hadoop fs -ls hivedata
Found 1 items
-rw-r--r--   1 cloudera cloudera         12 2018-09-25 20:51
hivedata/extdata.txt
[cloudera@quickstart Desktop]$ hadoop fs -cat
hivedata/extdata.txt
1,2,3
4,5,6
```

If you drop the managed table , you will see that you will not find your data in location: /user/hive/warehouse/emp_details.db

..................WORKING WITH MOVIES DATA SET ……………………….

Q1: Create a database called movies

```
     create database movies;
```

Q2: Work with database movies

```
     use movies;
```

Q3: create a table movies_details inside movies database

```
hive> create table movie_details(no int,
    > name string,
    > year int,
    > rating decimal,
    > views int)
    > row format delimited fields terminated by ',';
```

Q4: Load the data set of movies from local to hive table

```
hive> LOAD DATA LOCAL INPATH
'/home/cloudera/Desktop/hive_demo/movies_new' INTO table
movie_details;
```

Q5: Check the table created inside database.

```
hive> show tables;
```

Q6: Retrieve all the records in movies_details?

```
hive> select * from movie_details;
```

Q7: Print all movies between year 1920 and 1990

```
hive> select * from movie_details where year between 1920 and
1990;
```

Q8: Select all records where movie name starts from letter c or C

```
hive> select * from movie_details where name LIKE 'C%' or name
LIKE 'c%';
```

Q9: select all records where movie name starts with The

```
hive> select * from movie_details where name LIKE 'The%';
```

Q10: What is the maximum rating of the movie

```
hive> select max(rating) from movie_details;
```

Q11: count the number of records

```
hive> select count(*) from movie_details;
```

Q12: select rating of the movie School Ties

```
hive> SELECT name,rating FROM movie_details WHERE name = 'School
Ties';
```

Q13: List all the years with total number of views in each year (hint group by year), restrict the records to 5

```
hive> select year, sum(views) from movie_details group by year
LIMIT 5;
OK
```

PARTITIONING AND BUCKETING

Q1: Explain the concept of Partitioning and bucketing?

Assume that you are storing information of people in entire world spread across 196+ countries spanning around 500 crores of entries. If you want to query people from a particular country (Vatican city), in absence of partitioning, you have to scan all 500 crores of entries even to fetch thousand entries of a country. If you partition the table based on country, you can fine tune querying process by just checking the data for only one country partition. Hive partition creates a separate directory for a column(s) value.

Bucketing decomposes data into more manageable or equal parts.

With partitioning, there is a possibility that you can create multiple small partitions based on column values. If you go for bucketing, you are restricting number of buckets to store the data. This number is defined during table creation scripts

Q2: create a database shopping

```
hive>create database shopping;
use shopping;
```

Q3: create table (shopping1) inside the database shopping

```
create table shopping1(code INT, item_name STRING, category
string place string)
    > row format delimited fields terminated by ',';
```

Q4: Load the data in HIVE table from local

Suppose you have a file shop.txt (see the contents below) in your local machine, you can create a CSV file using gedit command

```
1,purse,bag,shimla
2,lipstick,cosmetic,delhi
3,bowl,utensils,jammu
4,mobile,electronic gadget,hyderabad
5,skirt,apparel,chennai
6,bed cover,furnishing,chandigarh
7,car,toys,karnal
8,hand purse,bag,solan
9,cream,cosmetic,jhodpur
10,plate,utensils,mohali
11,head phones,electronic gadget,calicut
12,top,apparel,mumbai
13,table cover,furnishing,agra
17,truck,toys,jaipur
18,wallet,bag,solan
19,foundation,cosmetic,jhodpur
20,spoon,utensils,mohali
21,speaker,electronic gadget,calicut
22,suit,apparel,mumbai
33,table sheet,furnishing,agra
24,auto,toys,jaipur
```

```
hive> LOAD DATA LOCAL INPATH '/home/cloudera/Desktop/shop.txt'
OVERWRITE INTO TABLE shopping1;
```

Q5: create a partition (shopping3) for table shopping1 and also create 3 buckets inside each partition

```
hive> create table shopping3(code INT, item_name STRING, place
string)
    > partitioned by (category string)
    > clustered by (place) into 3 buckets
    > row format delimited fields terminated by ','
    > stored as texfile;
```

Q4: Populate the partition with data

```
hive> from shopping1 txn INSERT OVERWRITE TABLE shopping3
PARTITION(category)
    select txn.code, txn.item_name, txn.place,
    txn.category DISTRIBUTE by category;
```

Q5: Check your partition

```
[cloudera@quickstart Desktop]$ hadoop fs -ls
/user/hive/warehouse/shopping.db/


[cloudera@quickstart Desktop]$ hadoop fs -ls
/user/hive/warehouse/shopping.db/shopping3

Found 8 items
drwxrwxrwx   - cloudera supergroup          0 2018-07-24 10:48
/user/hive/warehouse/shopping.db/shopping3/category=__HIVE_DEFAU
LT_PARTITION__
drwxrwxrwx   - cloudera supergroup          0 2018-07-24 10:48
/user/hive/warehouse/shopping.db/shopping3/category=apparel
drwxrwxrwx   - cloudera supergroup          0 2018-07-24 10:48
/user/hive/warehouse/shopping.db/shopping3/category=bag
drwxrwxrwx   - cloudera supergroup          0 2018-07-24 10:48
/user/hive/warehouse/shopping.db/shopping3/category=cosmetic
drwxrwxrwx   - cloudera supergroup          0 2018-07-24 10:48
/user/hive/warehouse/shopping.db/shopping3/category=electronic
gadget
drwxrwxrwx   - cloudera supergroup          0 2018-07-24 10:48
/user/hive/warehouse/shopping.db/shopping3/category=furnishing
drwxrwxrwx   - cloudera supergroup          0 2018-07-24 10:48
/user/hive/warehouse/shopping.db/shopping3/category=toys
drwxrwxrwx   - cloudera supergroup          0 2018-07-24 10:48
/user/hive/warehouse/shopping.db/shopping3/category=utensils
```

Q6: Check out the buckets for the partition “utensils”?

```
[cloudera@quickstart Desktop]$ hadoop fs -ls
/user/hive/warehouse/shopping.db/shopping3/category=utensils

Found 3 items
-rwxrwxrwx   1 cloudera supergroup          0 2018-07-24 10:48
/user/hive/warehouse/shopping.db/shopping3/category=utensils/000
000_0
-rwxrwxrwx   1 cloudera supergroup         13 2018-07-24 10:48
/user/hive/warehouse/shopping.db/shopping3/category=utensils/000
001_0
-rwxrwxrwx   1 cloudera supergroup         32 2018-07-24 10:48
/user/hive/warehouse/shopping.db/shopping3/category=utensils/000
002_0
```

Q7: Check out the contents of a particular bucket suppose 000001_0

```
[cloudera@quickstart Desktop]$ hadoop fs -cat
/user/hive/warehouse/shopping.db/shopping3/category=utensils/000
001_0
3,bowl,jammu
```