EXPERIMENT 02

CLASS: BE CMPN A ROLL NO. : 19

NAME: REBECCA DIAS PID: 182027

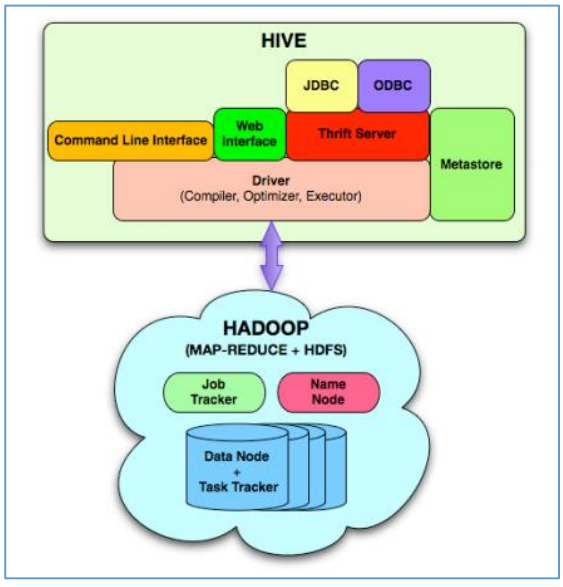
Aim:- To implement Hive commands in hadoop.

Theory :- Write about HIVE

Hive is a data warehouse infrastructure tool to process structured data in Hadoop. It resides on top of Hadoop to summarize Big Data, and makes querying and analyzing easy.

Apache Hive is an open source data warehouse software for reading, writing and managing large data set files that are stored directly in either the Apache Hadoop Distributed File System (HDFS) or other data storage systems such as Apache HBase. Hive enables SQL developers to write Hive Query Language (HQL) statements that are similar to standard SQL statements for data query and analysis. It is designed to make MapReduce programming easier because you don't have to know and write lengthy Java code. Instead, you can write queries more simply in HQL, and Hive can then create the map and reduce the functions.

Included with the installation of Hive is the Hive metastore, which enables you to apply a table structure onto large amounts of unstructured data. Once you create a Hive table, defining the columns, rows, data types, etc., all of this information is stored in the metastore and becomes part of the Hive architecture. Other tools such as Apache Spark and Apache Pig can then access the data in the metastore.



Implementation:- Implement the following commands in HIVE

1. Creating a database in HIVE
   `Create database rebecca;`

```
hive> create database rebecca
    > ;
OK
Time taken: 3.971 seconds
hive> create database temp
    > ;
OK
Time taken: 0.191 seconds
```

2. To display the existing databases
   `show databases;`

```
hive> show databases
    > ;
OK
default
rebecca
temp
Time taken: 0.619 seconds, Fetched: 3 row(s)
```

3. To drop a database use
   `drop database temp;`

```
hive> drop database temp
    > ;
OK
Time taken: 0.423 seconds
hive>
```

```
hive> show databases;
OK
default
rebecca
Time taken: 0.024 seconds, Fetched: 2 row(s)
hive>
```

4. To create tables inside a database. First use that particular database and then use the same create table syntax for creating tables inside the database

```
use rebecca;
create table emp(empid INT, ename STRING);
create table emp(empid INT, ename STRING) row format delimited
fields terminated by ',';
```

```
hive> use rebecca
    > ;
OK
Time taken: 0.085 seconds
```

```
hive> create table emp(empid INT , ename STRING);
OK
Time taken: 0.556 seconds
hive> create table dept(deptid INT , deptname STRING);
OK
Time taken: 0.201 seconds
```

5. To display the tables in a particular database use the show tables syntax
   `show tables;`

```
hive> show tables;
OK
dept
emp
Time taken: 0.085 seconds, Fetched: 2 row(s)
```

6. In hive the table names can be altered. So the following command is used for altering the table name
   `alter table dept rename to department;`

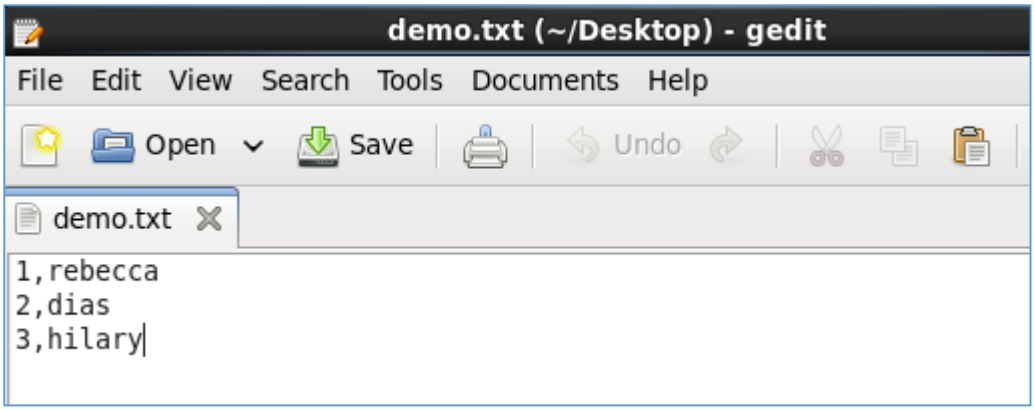
```
hive> alter table dept rename to department
    > ;
OK
Time taken: 0.419 seconds
```

```
hive> show tables
    > ;
OK
department
emp
Time taken: 0.037 seconds, Fetched: 2 row(s)
```

7. To drop tables use the following command. Drop table tablename
   `drop table department;`

```
hive> drop table department;
OK
Time taken: 1.062 seconds
hive> show tables;
OK
emp
Time taken: 0.042 seconds, Fetched: 1 row(s)
```

8. To create file in a local host and save it as demo.txt



9. To load data from local path use the following command;

```
load data local inpath '/home/cloudera/Desktop/demo.txt' overwrite
into table emp;
```

```
hive> create table emp(empid INT,ename STRING)row format delimited fields terminated by ',';
OK
Time taken: 0.252 seconds
```

```
hive> load data local inpath '/home/cloudera/Desktop/demo.txt' overwrite into table emp;
Loading data to table rebecca.emp
Table rebecca.emp stats: [numFiles=1, numRows=0, totalSize=26, rawDataSize=0]
OK
Time taken: 1.795 seconds
```

10. To view the contents of the table

```
Select * from emp;
```

```
hive> select * from emp;
OK
1       rebecca
2       dias
3       hilary
Time taken: 0.898 seconds, Fetched: 3 row(s)
hive>
```

11. To give filter conditions

```
Select * from emp where id=1;
```

```
hive> select * from emp where empid=1;
OK
1       rebecca
Time taken: 0.56 seconds, Fetched: 1 row(s)
hive>
```

12. To calculate the number of rows in a table

```
select count(1) from emp;
```

```
hive> select count(1) from emp;
Query ID = cloudera_20210811030404_e1f198d2-e6a2-4e08-9e22-2adec03285ea
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1628071667662_0002, Tracking URL = http://quickstart.cloudera
:8088/proxy/application_1628071667662_0002/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1628071667662_0002
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2021-08-11 03:04:31,456 Stage-1 map = 0%,  reduce = 0%
2021-08-11 03:04:44,522 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.44 sec
2021-08-11 03:04:57,187 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 3.46 sec
MapReduce Total cumulative CPU time: 3 seconds 460 msec
Ended Job = job_1628071667662_0002
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 3.46 sec   HDFS Read: 7405 HDFS Write: 2 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 460 msec
OK
3
Time taken: 43.882 seconds, Fetched: 1 row(s)
hive>
```

13. Insert into the tables

```
Insert into table_name values (‘values’,1);
```

```
hive> insert into table emp values (1,'reb');
Query ID = cloudera_20210811030909_a6b3d60b-3273-44b6-aba2-7da4a744363b
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1628071667662_0003, Tracking URL = http://quickstart.cloudera:8088/proxy/appli
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1628071667662_0003
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2021-08-11 03:09:16,640 Stage-1 map = 0%,  reduce = 0%
2021-08-11 03:09:27,880 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.17 sec
MapReduce Total cumulative CPU time: 2 seconds 170 msec
Ended Job = job_1628071667662_0003
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to: hdfs://quickstart.cloudera:8020/user/hive/warehouse/rebecca.db/emp/.hive-staging_
Loading data to table rebecca.emp
Table rebecca.emp stats: [numFiles=2, numRows=1, totalSize=32, rawDataSize=5]
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 2.17 sec   HDFS Read: 3965 HDFS Write: 73 SUCCESS
Total MapReduce CPU Time Spent: 2 seconds 170 msec
OK
Time taken: 25.118 seconds
hive> select * from emp;
OK
1       reb
1       rebecca
2       dias
3       hilary
Time taken: 0.077 seconds, Fetched: 4 row(s)
hive>
```

14. Sort by clause

```
Select * from emp SORT BY empid DESC;
```

```
hive> select * from emp SORT BY empid DESC;
Query ID = cloudera_20210811031515_9d47b6e3-950e-41b2-b877-b61bd1d8d5c7
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1628071667662_0004, Tracking URL = http://quickstart.cloudera:8088/proxy/applicatio
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1628071667662_0004
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2021-08-11 03:15:38,508 Stage-1 map = 0%,  reduce = 0%
2021-08-11 03:15:48,430 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.41 sec
2021-08-11 03:16:00,551 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 3.37 sec
MapReduce Total cumulative CPU time: 3 seconds 370 msec
Ended Job = job_1628071667662_0004
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 3.37 sec   HDFS Read: 6745 HDFS Write: 32 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 370 msec
OK
3       hilary
2       dias
1       rebecca
1       reb
Time taken: 36.259 seconds, Fetched: 4 row(s)
hive>
```

15. Group by Clause

```
Select empid , count(1) from emp GROUP BY empid;
```

```
hive> select empid , count(1) from emp GROUP BY empid;
Query ID = cloudera_20210811031818_bb69036c-e4a0-4905-bfe9-cdee89e2dd6c
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1628071667662_0005, Tracking URL = http://quickstart.cloudera:8088/proxy/applicati
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1628071667662_0005
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2021-08-11 03:18:50,823 Stage-1 map = 0%,  reduce = 0%
2021-08-11 03:19:01,892 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.52 sec
2021-08-11 03:19:14,070 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 3.89 sec
MapReduce Total cumulative CPU time: 3 seconds 890 msec
Ended Job = job_1628071667662_0005
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 3.89 sec   HDFS Read: 7912 HDFS Write: 12 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 890 msec
OK
1       2
2       1
3       1
Time taken: 36.358 seconds, Fetched: 3 row(s)
hive>
```

Conclusion:-

In this experiment we learned the usage of Hive. We learnt the use of different commands for making databases, commands for reading and writing files, commands for making a directory and copying contents inside the created database.The usage of the databases used in hive was understood by us.