

EXPERIMENT 4

AIM: Execute HIVE commands to load, insert, retrieve, update, or delete data in the tables.

THEORY:

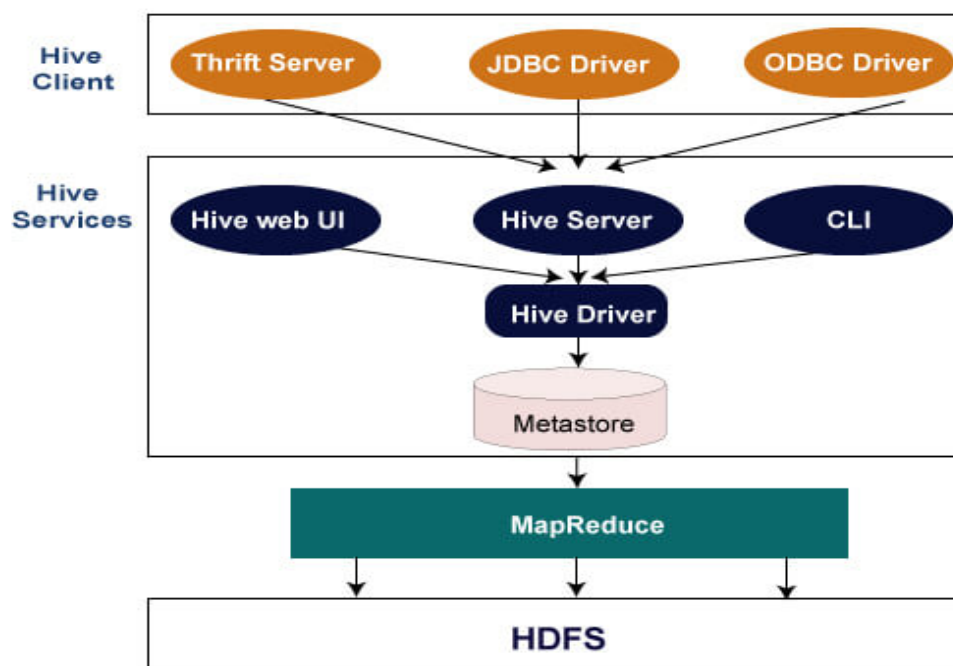
Hive is a data warehouse system which is used to analyze structured data. It is built on the top of Hadoop. Hive provides the functionality of reading, writing, and managing large datasets residing in distributed storage. It runs SQL like queries called HQL (Hive query language) which gets internally converted to MapReduce jobs.

Using Hive, we can skip the requirement of the traditional approach of writing complex MapReduce programs. Hive supports Data Definition Language (DDL), Data Manipulation Language (DML), and User Defined Functions (UDF).

Features of Hive:

- Hive is fast and scalable.
- It provides SQL-like queries (i.e., HQL) that are implicitly transformed to MapReduce or Spark jobs.
- It is capable of analyzing large datasets stored in HDFS.
- It allows different storage types such as plain text, RCFile, and HBase.
- It uses indexing to accelerate queries.
- It can operate on compressed data stored in the Hadoop ecosystem.

Architecture of Hive:



Hive Client

Hive allows writing applications in various languages, including Java, Python, and C++. It supports different types of clients such as:-

- **Thrift Server** - It is a cross-language service provider platform that serves the request from all those programming languages that supports Thrift.
- **JDBC Driver** - It is used to establish a connection between hive and Java applications.
- **ODBC Driver** - It allows the applications that support the ODBC protocol to connect to Hive.

Hive Services

The following are the services provided by Hive:-

- **Hive CLI** - The Hive CLI (Command Line Interface) is a shell where we can execute Hive queries and commands.
- **Hive Web User Interface** - The Hive Web UI is just an alternative of Hive CLI. It provides a web-based GUI for executing Hive queries and commands.
- **Hive MetaStore** - It is a central repository that stores all the structure information of various tables and partitions in the warehouse. It also includes metadata of column and its type information, the serializers and deserializers which is used to read and write data and the corresponding HDFS files where the data is stored.
- **Hive Server** - It is referred to as Apache Thrift Server. It accepts the request from different clients and provides it to Hive Driver.
- **Hive Driver** - It receives queries from different sources like web UI, CLI, Thrift, and JDBC/ODBC driver. It transfers the queries to the compiler.
- **Hive Compiler** - The purpose of the compiler is to parse the query and perform semantic analysis on the different query blocks and expressions. It converts HiveQL statements into MapReduce jobs.
- **Hive Execution Engine** - Optimizer generates the logical plan in the form of DAG of map-reduce tasks and HDFS tasks. In the end, the execution engine executes the incoming tasks in the order of their dependencies.

Working of Hive:

Step 1: executeQuery: The user interface calls the execute interface to the driver.

Step 2: getPlan: The driver accepts the query, creates a session handle for the query, and passes the query to the compiler for generating the execution plan.

Step 3: getMetaData: The compiler sends the metadata request to the metastore.

Step 4: sendMetaData: The metastore sends the metadata to the compiler.

The compiler uses this metadata for performing type-checking and semantic analysis on the expressions in the query tree. The compiler then generates the execution plan (**Directed acyclic Graph**). For Map Reduce jobs, the plan contains **map operator trees** (operator trees which are executed on mapper) and **reduce operator tree** (operator trees which are executed on reducer).

Step 5: sendPlan: The compiler then sends the generated execution plan to the driver.

Step 6: executePlan: After receiving the execution plan from compiler, driver sends the execution plan to the execution engine for executing the plan.

Step 7: submit job to MapReduce: The execution engine then sends these stages of DAG to appropriate components.

For each task, either mapper or reducer, the deserializer associated with a table or intermediate output is used in order to read the rows from HDFS files. These are then passed through the associated operator tree.

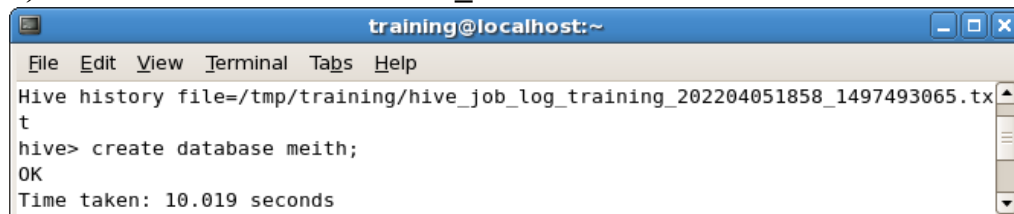
Once the output gets generated, it is then written to the HDFS temporary file through the serializer. These temporary HDFS files are then used to provide data to the subsequent map/reduce stages of the plan.

For DML operations, the final temporary file is then moved to the table's location.

Step 8, 9, 10: sendResult: Now for queries, the execution engine reads the contents of the temporary files directly from HDFS as part of a fetch call from the driver. The driver then sends results to the Hive interface.

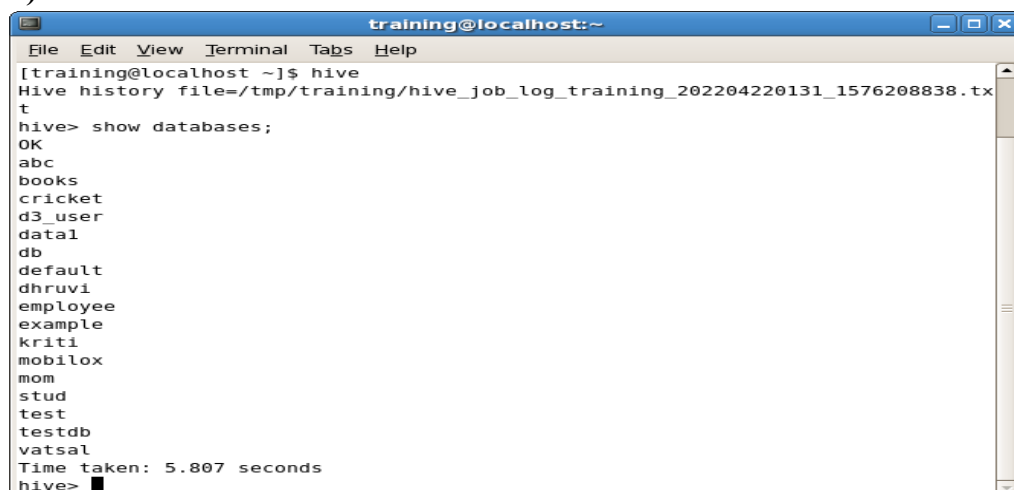
QUERIES:

1) create database <database_name>

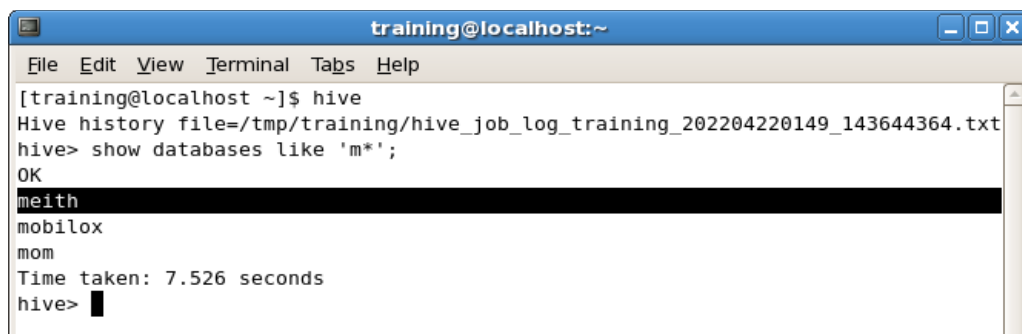


```
training@localhost:~  
File Edit View Terminal Tabs Help  
Hive history file=/tmp/training/hive_job_log_training_202204051858_1497493065.tx  
t  
hive> create database meith;  
OK  
Time taken: 10.019 seconds
```

2) show databases



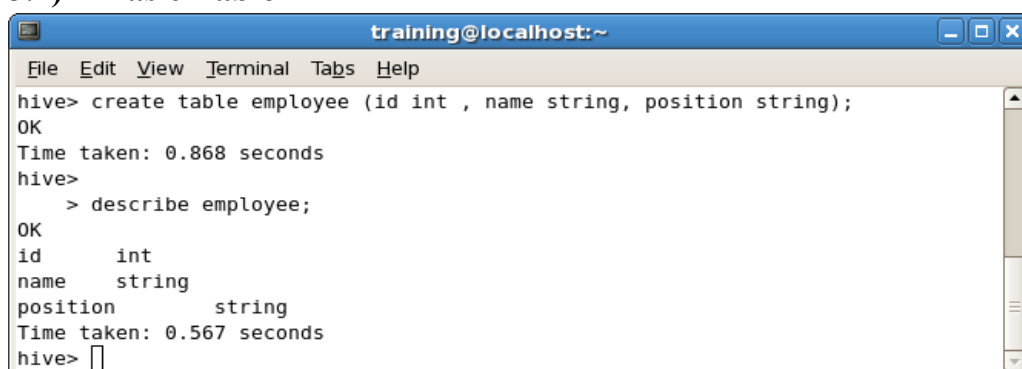
```
training@localhost:~  
File Edit View Terminal Tabs Help  
[training@localhost ~]$ hive  
Hive history file=/tmp/training/hive_job_log_training_202204220131_1576208838.tx  
t  
hive> show databases;  
OK  
abc  
books  
cricket  
d3_user  
data1  
db  
default  
dhruvi  
employee  
example  
kriti  
mobilox  
mom  
stud  
test  
testdb  
vatsal  
Time taken: 5.807 seconds  
hive>
```



```
training@localhost:~  
File Edit View Terminal Tabs Help  
[training@localhost ~]$ hive  
Hive history file=/tmp/training/hive_job_log_training_202204220149_143644364.txt  
hive> show databases like 'm*';  
OK  
meith  
mobiloX  
mom  
Time taken: 7.526 seconds  
hive>
```

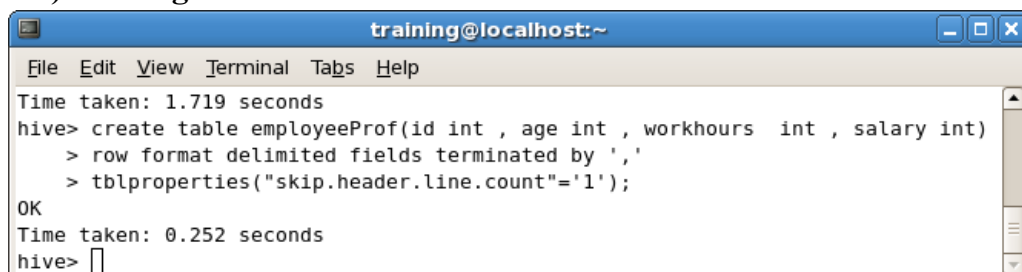
3) Create Table

3.1) Basic Table



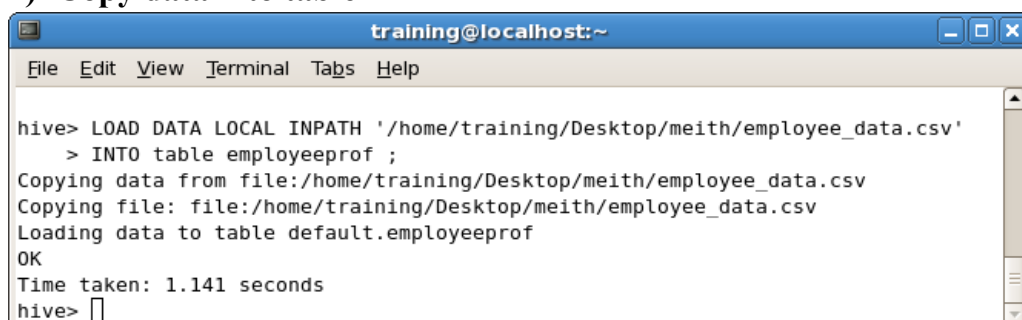
```
training@localhost:~  
File Edit View Terminal Tabs Help  
hive> create table employee (id int , name string, position string);  
OK  
Time taken: 0.868 seconds  
hive>  
    > describe employee;  
OK  
id      int  
name    string  
position string  
Time taken: 0.567 seconds  
hive>
```

3.2) Using Delimiter



```
training@localhost:~  
File Edit View Terminal Tabs Help  
Time taken: 1.719 seconds  
hive> create table employeeProf(id int , age int , workhours int , salary int)  
    > row format delimited fields terminated by ','  
    > tblproperties("skip.header.line.count"='1');  
OK  
Time taken: 0.252 seconds  
hive>
```

4) Copy data into table



```
training@localhost:~  
File Edit View Terminal Tabs Help  
hive> LOAD DATA LOCAL INPATH '/home/training/Desktop/meith/employee_data.csv'  
    > INTO table employeeProf ;  
Copying data from file:/home/training/Desktop/meith/employee_data.csv  
Copying file: file:/home/training/Desktop/meith/employee_data.csv  
Loading data to table default.employeeProf  
OK  
Time taken: 1.141 seconds  
hive>
```

```
training@localhost:~  
File Edit View Terminal Tabs Help  
Time taken: 0.677 seconds  
hive> LOAD DATA LOCAL INPATH '/home/training/Desktop/meith/names.csv' into table EmpPersonal ;  
Copying data from file:/home/training/Desktop/meith/names.csv  
Copying file: file:/home/training/Desktop/meith/names.csv  
Loading data to table default.emppersonal  
OK  
Time taken: 0.555 seconds  
hive> █
```

5) Display data of table

```
training@localhost:~  
File Edit View Terminal Tabs Help  
Time taken: 0.365 seconds  
hive> SELECT * FROM EmpProf LIMIT 15;  
OK  
ID      AGE      WORK_HOURS      SALARY  
0       0       36             5  
1       1       55             5  
2       2       61             1  
3       3       29             6  
4       4       34             2  
5       5       42             3  
6       6       53             6  
7       7       41             6  
8       8       47             6  
9       9       31             8  
10      10      47             9  
11      11      40             1  
12      12      41             2  
13      13      46             8  
Time taken: 0.309 seconds  
hive> █
```

6) Describe table

```
training@localhost:~  
File Edit View Terminal Tabs Help  
hive> describe formatted emp;  
OK  
# col_name      data_type      comment  
  
employee_id     string         None  
first_name      string         None  
last_name       string         None  
email           string         None  
phone_number    string         None  
  
# Detailed Table Information  
Database:        rahuldb  
Owner:          training  
CreateTime:      Tue Apr 05 20:14:19 PDT 2022  
LastAccessTime:  UNKNOWN  
Protect Mode:    None  
Retention:       0  
Location:        hdfs://localhost/user/hive/warehouse/  
Table Type:      MANAGED_TABLE  
Table Parameters:  
    transient_lastDdlTime 1649214859  
  
# Storage Information  
SerDe Library:   org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe  
InputFormat:     org.apache.hadoop.mapred.TextInputFormat  
OutputFormat:    org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat  
Compressed:      No  
Num Buckets:     -1  
Bucket Columns:  []  
Sort Columns:    []  
Storage Desc Params:  
    field.delim      ,  
    serialization.format ,  
Time taken: 0.439 seconds  
hive> █
```

7) Create External Table

```
training@localhost:~  
File Edit View Terminal Tabs Help  
hive> create external table EmpProfExt(D string, AGE string,WORK_HOURS string, S  
ALARY string) row format delimited fields terminated by ',' ;  
OK  
Time taken: 0.112 seconds  
hive> █
```

8) External Table in Web Browser UI

HDFS:/user/hive/warehouse/empprof - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost.localdomain:50075/browseDirectory.jsp?dir=%2Fuser%2Fhive%2Fwarehouse%2Fempprof&namenodeinfoPort=50070

HDFS:/user/hive/warehouse/em...

Contents of directory /user/hive/warehouse/empprof

Goto: /user/hive/warehouse/empp go

Go to parent directory

Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
employee_data.csv	file	18.26 KB	1	64 MB	2022-04-05 20:20	rw-r--r--	training	supergroup

Go back to DFS home

Local logs

Log directory

Cloudera's Distribution including Apache Hadoop, 2022.

9) Drop Table

```
training@localhost:~  
File Edit View Terminal Tabs Help  
hive> drop table empz;  
OK  
Time taken: 0.16 seconds  
hive> select * from empz;  
FAILED: Error in semantic analysis: Line 1:14 Table not found empz  
hive> █
```

10) Nested Query

```
training@localhost:~  
File Edit View Terminal Tabs Help  
hive> select name from (select concat_ws(" ", FIRST_NAME, LAST_NAME) as name FROM emp) emp;  
Total MapReduce jobs = 1  
Launching Job 1 out of 1  
Number of reduce tasks is set to 0 since there's no reduce operator  
Starting Job = job_202204010036_0012, Tracking URL = http://localhost:50030/jobdetails.jsp?jobid=job_202204010036_0012  
Kill Command = /usr/lib/hadoop/bin/hadoop job -Dmapred.job.tracker=localhost:8021 -kill job_202204010036_0012  
2022-04-05 23:33:58,253 Stage-1 map = 0%, reduce = 0%  
2022-04-05 23:34:00,339 Stage-1 map = 100%, reduce = 0%  
2022-04-05 23:34:02,469 Stage-1 map = 100%, reduce = 100%  
Ended Job = job_202204010036_0012  
OK  
FIRST_NAME LAST_NAME  
Donald OConnell  
Douglas Grant  
Jennifer Whalen  
Michael Hartstein  
Pat Fay  
Susan Mavris  
Hermann Baer  
Shelley Higgins  
William Gietz  
Steven King  
Neena Kochhar  
Lex De Haan  
Alexander Hunold  
Bruce Ernst  
David Austin  
Valli Pataballa  
Diana Lorentz
```

11) Static Partition

11.1) Static Partitioning

```
training@localhost:~  
File Edit View Terminal Tabs Help  
Time taken: 0.327 seconds  
hive> CREATE TABLE empPartition (EMPLOYEE_ID string,FIRST_NAME string,LAST_NAME string,EMAIL s  
tring,PHONE_NUMBER string)  
  > PARTITIONED BY (CATEGORY string)  
  > ROW FORMAT DELIMITED  
  > FIELDS TERMINATED BY ','  
  > LINES TERMINATED BY '\n';  
OK  
Time taken: 0.215 seconds  
hive>
```

11.2) Load Data into Partitions

```
training@localhost:~  
File Edit View Terminal Tabs Help  
Time taken: 0.215 seconds  
hive> LOAD DATA  
  > LOCAL INPATH '/home/training/Desktop/Meith/employees.csv'  
  > INTO TABLE empPartition  
  > PARTITION (CATEGORY = '1');  
Copying data from file:/home/training/Desktop/Meith/employees.csv  
Copying file: file:/home/training/Desktop/Meith/employees.csv  
Loading data to table default.emppartition partition (category=1)  
OK  
Time taken: 1.368 seconds  
hive> LOAD DATA  
  > LOCAL INPATH '/home/training/Desktop/Meith/employees.csv'  
  > INTO TABLE empPartition  
  > PARTITION (CATEGORY = '2');  
Copying data from file:/home/training/Desktop/Meith/employees.csv  
Copying file: file:/home/training/Desktop/Meith/employees.csv  
Loading data to table default.emppartition partition (category=2)  
OK  
Time taken: 0.949 seconds  
hive> LOAD DATA  
  > LOCAL INPATH '/home/training/Desktop/Meith/employees.csv'  
  > INTO TABLE empPartition  
  > PARTITION (CATEGORY = '3');  
Copying data from file:/home/training/Desktop/Meith/employees.csv  
Copying file: file:/home/training/Desktop/Meith/employees.csv  
Loading data to table default.emppartition partition (category=3)  
OK  
Time taken: 0.698 seconds  
hive>
```

11.3) Static Partitions on Web Browser

HDFS:/user/hive/warehouse/emppartition - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost.localdomain:50075/browseDirectory.jsp?dir=%2Fuser%2Fhive%2Fwarehouse%2Femppartition&namenodeInfoPort=50070

HDFS:/user/hive/warehouse/em...

Contents of directory /user/hive/warehouse/emppartition

Goto: /user/hive/warehouse/empp go

Go to parent directory

Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
category=1	dir				2022-04-22 02:31	rw-r-xr-x	training	supergroup
category=2	dir				2022-04-22 02:32	rw-r-xr-x	training	supergroup
category=3	dir				2022-04-22 02:32	rw-r-xr-x	training	supergroup

Go back to DFS home

Local logs

Log directory

Cloudera's Distribution including Apache Hadoop, 2022.

12)Dynamic Partition

12.1) Dynamic Partitioning

```
training@localhost:~  
File Edit View Terminal Tabs Help  
OK  
Time taken: 3.704 seconds  
hive> CREATE TABLE empDynamicPartition (EMPLOYEE_ID string,FIRST_NAME string,  
> LAST_NAME string,EMAIL string,PHONE_NUMBER string)  
> PARTITIONED BY (CATEGORY string);  
OK  
Time taken: 0.579 seconds  
hive> █
```

12.2) Load Data into Partitions

```
training@localhost:~  
File Edit View Terminal Tabs Help  
hive> INSERT OVERWRITE TABLE empDynamicPartition  
> PARTITION (CATEGORY)  
> SELECT * FROM employees;  
Total MapReduce jobs = 2  
Launching Job 1 out of 2  
Number of reduce tasks is set to 0 since there's no reduce operator  
Starting Job = job_202204220121_0001, Tracking URL = http://localhost:50030/jobdetails.jsp?jobid=job_2022  
04220121_0001  
Kill Command = /usr/lib/hadoop/bin/hadoop job -Dmapred.job.tracker=localhost:8021 -kill job_202204220121  
_0001  
2022-04-22 03:06:54,234 Stage-1 map = 0%, reduce = 0%  
2022-04-22 03:06:58,376 Stage-1 map = 100%, reduce = 0%  
2022-04-22 03:07:01,490 Stage-1 map = 100%, reduce = 100%  
Ended Job = job_202204220121_0001  
Ended Job = -173417231, job is filtered out (removed at runtime).  
Moving data to: hdfs://localhost/tmp/hive-training/hive_2022-04-22_03-06-33_295_1836412915131451468/-ext-  
10000  
Loading data to table default.empdynamicpartition partition (category=null)  
Loading partition {category=1}  
Loading partition {category=2}  
Loading partition {category=3}  
Loading partition {category=CATEGORY}  
Partition default.empdynamicpartition{category=1} stats: [num_files: 1, num_rows: 0, total_size: 701]  
Partition default.empdynamicpartition{category=2} stats: [num_files: 1, num_rows: 0, total_size: 772]  
Partition default.empdynamicpartition{category=3} stats: [num_files: 1, num_rows: 0, total_size: 448]  
Partition default.empdynamicpartition{category=CATEGORY} stats: [num_files: 1, num_rows: 0, total_size: 5  
2]  
Table default.empdynamicpartition stats: [num_partitions: 4, num_files: 4, num_rows: 0, total_size: 1973]  
51 Rows loaded to empdynamicpartition  
OK  
Time taken: 31.439 seconds  
hive> █
```

12.3) Dynamic Partitions on Web Browser

HDFS:/user/hive/warehouse/empdynamicpartition - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost.localdomain:50075/browseDirectory.jsp?dir=/user/hive/warehouse/empdynamicpartition&namenodeinfoPort=50070

HDFS:/user/hive/warehouse/em...

Contents of directory /user/hive/warehouse/empdynamicpartition

Goto: /user/hive/warehouse/empd go

Go to parent directory

Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
category=1	dir				2022-04-22 03:06	rwxr-xr-x	training	supergroup
category=2	dir				2022-04-22 03:06	rwxr-xr-x	training	supergroup
category=3	dir				2022-04-22 03:06	rwxr-xr-x	training	supergroup
category=CATEGORY	dir				2022-04-22 03:06	rwxr-xr-x	training	supergroup

Go back to DFS home

Local logs

Log directory

Cloudera's Distribution including Apache Hadoop, 2022.

13) Bucketing

13.1) Create Buckets

```
training@localhost:~  
File Edit View Terminal Tabs Help  
hive> set hive.enforce.bucketing =true;  
hive> CREATE TABLE emp_Bucket(EMPLOYEE_ID string,FIRST_NAME string, LAST_NAME string, EMAIL string, PHONE_NUMBER string, CATEGORY string)  
> CLUSTERED BY(CATEGORY) INTO 3 BUCKETS  
> ROW FORMAT DELIMITED  
> FIELDS TERMINATED BY ','  
> LINES TERMINATED BY '\n';  
OK  
Time taken: 0.392 seconds  
hive>
```

13.2) Load Data into Buckets

```
training@localhost:~  
File Edit View Terminal Tabs Help  
Time taken: 0.392 seconds  
hive> INSERT OVERWRITE TABLE emp_Bucket  
> SELECT * FROM employees;  
Total MapReduce jobs = 1  
Launching Job 1 out of 1  
Number of reduce tasks determined at compile time: 3  
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 mapred.reduce.tasks=<number>  
Starting Job = job_202204220121_0002, Tracking URL = http://localhost:50030/jobdetails.jsp?jobid=job_202204220121_0002  
Kill Command = /usr/lib/hadoop/bin/hadoop job -Dmapred.job.tracker=localhost:8021 -kill job_202204220121_0002  
2022-04-22 03:21:03,349 Stage-1 map = 0%, reduce = 0%  
2022-04-22 03:21:15,086 Stage-1 map = 100%, reduce = 0%  
2022-04-22 03:21:26,428 Stage-1 map = 100%, reduce = 33%  
2022-04-22 03:21:27,440 Stage-1 map = 100%, reduce = 67%  
2022-04-22 03:21:34,554 Stage-1 map = 100%, reduce = 100%  
Ended Job = job_202204220121_0002  
Loading data to table default.emp_bucket  
Deleted hdfs://localhost/user/hive/warehouse/emp_bucket  
Table default.emp_bucket stats: [num_partitions: 0, num_files: 3, num_rows: 0, total_size: 2082]  
51 Rows loaded to emp_bucket  
OK  
Time taken: 48.55 seconds  
hive>
```

13.3) Buckets on Web Browser

HDFS://user/hive/warehouse/emp_bucket - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost.localdomain:50075/browseDirectory.jsp?dir=%2Fuser%2Fhive%2Fwarehouse%2Femp_bucket&namenodeInfoPort=50070

Contents of directory /user/hive/warehouse/emp_bucket

Goto: /user/hive/warehouse/emp_ go

Go to parent directory

Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
000000_0	file	0.46 KB	1	64 MB	2022-04-22 03:21	rw-r--r--	training	supergroup
000001_0	file	0.78 KB	1	64 MB	2022-04-22 03:21	rw-r--r--	training	supergroup
000002_0	file	0.79 KB	1	64 MB	2022-04-22 03:21	rw-r--r--	training	supergroup

Go back to DFS home

Local logs

Log directory

Cloudera's Distribution including Apache Hadoop, 2022.

HDFS:/user/hive/warehouse/emp_bucket/000000_0 - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost.localdomain:50075/browseBlock.jsp?blockId=6952872933853196942&blockSize=472&genstamp=2852&filename=%2Fuser%2Fhive%2Fw...

HDFS:/user/hive/warehouse/em...

File: /user/hive/warehouse/emp_bucket/000000_0

Goto: /user/hive/warehouse/emp_ go

[Go back to dir listing](#)
[Advanced view/download options](#)

```
134,Michael,Rogers,MROGERS,650.127.1834,3
140,Joshua,Patel,JPATEL,650.121.1834,3
139,Jason,Mallin,JMALLIN,650.127.1934,3
132,TJ,Olson,TJOLSON,650.124.8234,3
131,James,Marlow,JAMLOW,650.124.7234,3
130,Mozhe,Atkinson,MATKINSO,650.124.6234,3
129,Laura,Bissot,LBISSOT,650.124.5234,3
139,John,Seo,JSEO,650.121.2019,3
138,Stephen,Stiles,SSTILES,650.121.2034,3
137,Renske,Ladwig,RLADWIG,650.121.1234,3
136,Hazel,Philtanker,HPHILTAN,650.127.1634,3
135,Ki,Gee,KGEE,650.127.1734,3
```

[Download this file](#)
[Tail this file](#)

Chunk size to view (in bytes, up to file's DFS block size):

14) JOINS

14.1) Inner join

training@localhost:~

File Edit View Terminal Tabs Help

```
hive> SELECT b.EMPLOYEE_ID, b.FIRST_NAME, a.SALARY
> FROM info2 b JOIN info a
> ON (b.EMPLOYEE_ID = a.ID);
Total MapReduce 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 mapred.reduce.tasks=<number>
Starting Job = job_202204290221_0001, Tracking URL = http://localhost:50030/jobdetails.jsp?jobid=job_202204290221_0001
Kill Command = /usr/lib/hadoop/bin/hadoop job -Dmapred.job.tracker=localhost:8021 -kill job_202204290221_0001
2022-04-29 02:51:20,182 Stage-1 map = 0%, reduce = 0%
2022-04-29 02:51:26,313 Stage-1 map = 100%, reduce = 0%
2022-04-29 02:51:37,550 Stage-1 map = 100%, reduce = 100%
Ended Job = job_202204290221_0001
OK
100      Steven      6
101      Neena       7
102      Lex         6
103      Alexander    8
104      Bruce       7
105      David       5
106      Valli       5
107      Diana       9
108      Nancy       5
109      Daniel      5
110      John        7
111      Ismael      5
112      Jose Manuel  8
113      Luis        3
114      Den         6
115      Alexander    7
116      Shelli      3
117      Sigal       6
118      Guy         8
119      Karen       9
120      Matthew     5
```

14.2) Left Outer Join

```
training@localhost:~  
File Edit View Terminal Tabs Help  
hive> SELECT b.EMPLOYEE_ID, b.FIRST_NAME, a.SALARY  
  > FROM info2 b  
  > LEFT OUTER JOIN info a  
  > ON (b.EMPLOYEE_ID = a.ID);  
Total MapReduce 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 mapred.reduce.tasks=<number>  
Starting Job = job_202204290221_0002, Tracking URL = http://localhost:50030/jobdetails.jsp?jobid=job_202204290221_0002  
Kill Command = /usr/lib/hadoop/bin/hadoop job -Dmapred.job.tracker=localhost:8021 -kill job_202204290221_0002  
2022-04-29 02:55:04,874 Stage-1 map = 0%, reduce = 0%  
2022-04-29 02:55:10,494 Stage-1 map = 100%, reduce = 0%  
2022-04-29 02:55:18,629 Stage-1 map = 100%, reduce = 17%  
2022-04-29 02:55:19,651 Stage-1 map = 100%, reduce = 100%  
Ended Job = job_202204290221_0002  
OK  
NULL    FIRST_NAME    NULL  
100      Steven    6  
101      Neena     7  
102      Lex       6  
103      Alexander 8  
104      Bruce     7  
105      David     5  
106      Valli     5  
107      Diana     9  
108      Nancy     5  
109      Daniel    5  
110      John      7  
111      Ismael    5  
112      Jose Manuel 8  
113      Luis      3  
114      Den       6  
115      Alexander 7  
116      Shelli    3  
117      Sigal     6
```

11.3) Right Outer Join

```
training@localhost:~  
File Edit View Terminal Tabs Help  
hive> SELECT b.EMPLOYEE_ID, b.FIRST_NAME, a.AGE  
  > FROM info a  
  > RIGHT OUTER JOIN info2 b  
  > ON (b.EMPLOYEE_ID = a.ID);  
Total MapReduce 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 mapred.reduce.tasks=<number>  
Starting Job = job_202204290221_0005, Tracking URL = http://localhost:50030/jobdetails.jsp?jobid=job_202204290221_0005  
Kill Command = /usr/lib/hadoop/bin/hadoop job -Dmapred.job.tracker=localhost:8021 -kill job_202204290221_0005  
2022-04-29 03:03:17,630 Stage-1 map = 0%, reduce = 0%  
2022-04-29 03:03:23,890 Stage-1 map = 100%, reduce = 0%  
2022-04-29 03:03:32,011 Stage-1 map = 100%, reduce = 33%  
2022-04-29 03:03:34,044 Stage-1 map = 100%, reduce = 100%  
Ended Job = job_202204290221_0005  
OK  
NULL    FIRST_NAME    NULL  
100      Steven    100  
101      Neena     101  
102      Lex       102  
103      Alexander 103  
104      Bruce     104  
105      David     105  
106      Valli     106  
107      Diana     107  
108      Nancy     108  
109      Daniel    109  
110      John      110  
111      Ismael    111  
112      Jose Manuel 112  
113      Luis      113  
114      Den       114  
115      Alexander 115  
116      Shelli    116  
117      Sigal     117
```

CONCLUSION:

In this experiment, I have implemented HIVE commands (HIVEQL) like creating database, tables, inserting data into the tables from csv files, partitioning and bucketing the tables, different joins on the tables and finally deleting the tables on Cloudera using VMware player. The queries were much like SQL queries, and it masked the mapReduce of the Hadoop framework which was evident while executing the queries in the terminal.