Exp.No: 5

Install Hive and Create Tables in Hive and write queries to access the data in the table

AIM:

To install Hive, design and test various schema models to optimize data storage and retrieval using Hive.

PROCEDURE:

Step 1: Start Hive Open a terminal and start Hive by running:

hive

```
vboxuser@ubuntu-22:~$ hive
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Hive Session ID = a99dc067-b38b-47ec-9b4c-b51dfdb62c67

Logging initialized using configuration in jar:file:/home/vboxuser/apache-hive-3.1.3-bin/lib/hive-common-3.1.3.jar!/hive-log4j2.properties Async: true
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
Hive Session ID = 05c20837-4dd2-4127-9d26-cc13e11ecaeb
hive>
```

Step 2: Create a Database: Create a new database in Hive:

hive> CREATE DATABASE courses;

```
hive> CREATE DATABASE courses;
OK
Time taken: 6.451 seconds
hive>
```

Step 3: Use the Database: Switch to the newly created database:

hive> USE courses;

```
hive> USE courses;
OK
Time taken: 0.212 seconds
hive>
```

Step 4: Create a Table: Create a simple table in your database:

hive> CREATE TABLE course_table(course_id INT, course_name STRING);

```
hive> CREATE TABLE course_table(course_id INT, course_name STRING);
OK
Time taken: 7.942 seconds
hive>
```

Step 5: Load Sample Data: You can insert sample data into the table:

hive> INSERT INTO course_table VALUES (1, 'Cloud Computing'), (2, 'Data Analytics'), (3, 'Deep Learning Concepts');

```
vboxuser@ubuntu-22: ~
hive> INSERT INTO course_table VALUES(1, 'Cloud Computing'),(2,'Data Analytics'),(3, 'Deep Learning Concepts');
Query ID = vboxuser_20240921204406_b13c5909-e28b-40e8-9506-544601407539
Total jobs = 3
Launching Job 1 out of 3
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_1726931387005_0001, Tracking URL = http://ubuntu-22.4.myguest.virtualbox.org:8088/proxy/application_1726931
387005_0001/
Kill Command = /home/vboxuser/hadoop-3.2.3/bin/mapred job  -kill job_1726931387005_0001
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2024-09-21 20:48:10,204 Stage-1 map = 0%,  reduce = 0%
2024-09-21 20:49:10,465 Stage-1 map = 0%,  reduce = 0%
2024-09-21 20:50:10,594 Stage-1 map = 0%,  reduce = 0%
2024-09-21 20:50:23,895 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 9.77 sec
2024-09-21 20:51:24,658 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 9.77 sec
2024-09-21 20:51:59,211 Stage-1 map = 100%,  reduce = 67%, Cumulative CPU 10.81 sec
2024-09-21 20:52:08,531 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 17.78 sec
MapReduce Total cumulative CPU time: 17 seconds 780 msec
Ended Job = job_1726931387005_0001
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 directory hdfs://localhost:9000/user/hive/warehouse/courses.db/course_table/.hive-staging_hive_2024-09-21_20-44
-06_914_4442615585155451475-1/-ext-10000
Loading data to table courses.course_table
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 17.78 sec   HDFS Read: 15973 HDFS Write: 325 SUCCESS
Total MapReduce CPU Time Spent: 17 seconds 780 msec
OK
Time taken: 608.137 seconds
hive>
```

Step 6: Query Your Data: Use SQL-like queries to retrieve data from your table:
hive> CREATE VIEW mycourse AS SELECT course_id, course_name FROM course_table;

```
hive> CREATE VIEW mycourse AS SELECT course_id, course_name FROM course_table;
OK
Time taken: 6.303 seconds
hive>
```

Step 7: View the data: To see the data in the view, you would need to query the view

hive> SELECT * FROM mycourse;

```
hive> SELECT * FROM mycourse;
OK
1       Cloud Computing
2       Data Analytics
3       Deep Learning Concepts
Time taken: 2.082 seconds, Fetched: 3 row(s)
hive>
```

Step 8: Describe a Table: You can describe the structure of a table using the DESCRIBE command:

hive> DESCRIBE course_table;

```
hive> DESCRIBE course_table;
OK
course_id               int
course_name             string
Time taken: 1.145 seconds, Fetched: 2 row(s)
hive>
```

Step 9: Alter a Table: You can alter the table structure by adding a new column:
hive> ALTER TABLE course_table ADD COLUMNS (students_enrolled INT);

```
hive> ALTER TABLE course_table ADD COLUMNS(students_enrolled INT);
OK
Time taken: 2.443 seconds
hive>
```

Step 10: Quit Hive: To exit the Hive CLI, simply type:
hive> quit;

```
hive> quit;
vboxuser@ubuntu-22:~$
```

RESULT:

Thus, the usage of various commands in Hive has been successfully completed.