**Hive DDL:-**

Hive Data Definition Language is similar to SQL Data Definition Language that is used for creating, altering, dropping the database, tables, views etc.,

**DATABASES:-**

* **Creating Database:-**

Database can be created by using the **following syntax.**

**SYNTAX:-**

create database database\_name;

* **Dropping Databases:-**

Database can be dropped by the following syntax.

**SYNTAX:-**

Drop database database\_name;

* **Displaying the list of Hive Databases that is available in HDFS:-**

List of available created databases can be displayed by query with the following Syntax.

**SYNTAX:-**

show database;

* **Using Database:-**

Out of available database, if you want to work on one specific database, then you can do that using the query with following Syntax.

**SYNTAX:-**

use database\_name;

* **Displaying the currently working Database name:-**

Database name that you are currently working on can be displayed with the help of following Syntax.

**SYNTAX:-**

set hive.cli.print.current.db=true;

* **Describing Database:-**

If you describe your database, you can know the “name of the Database”, “location of Database that gets stored in HDFS”, “name of the user who created the Database” etc. You can describe the database using following Syntax.

**SYNTAX:-**

describe database database\_name;

**TABLES:-**

* **Creating Internal Table:-**

Internal or Managed Table can be created with the help of following syntax.

**SYNTAX:-**

Create table table\_name

(

Col\_name1 type,

Col\_name2 type,

.

.

.

.

.

Col\_nameN type

)

row format delimited fields terminated by ‘,’;

Using the last line, we can load the file into table that is delimited(separated) by comma(‘,’).

This table is known as **Internal table** which is **managed by hive.**

* **Creating External Table:-**

External table can be created when the data file that needs to be loaded is **already available in HDFS**. It can be created using the query with the following Syntax.

**SYNTAX:-**

Create **external** table table\_name

(

Col\_name1 type,

Col\_name2 type,

.

.

.

.

.

Col\_nameN type

)

row format delimited fields terminated by ‘,’;

LOCATION ‘/../…’;

The table will pick the file from the specified location in the query.

* **Dropping Tables:**

Table can be dropped by the query with the following Syntax.

**SYNTAX:-**

drop table table\_name;

* **ALTERING TABLE NAME:-**

Table Name can be altered using followingsyntax.

**SYNTAX:-**

alter table old\_table\_name to new\_table\_name;

**VIEWS:-**

* **Creating Views:-**

Views can be created with the help of following syntax.

**SYNTAX:-**

create view view\_name as (query that you want to insert to that view)

* **Dropping View:-**

**SYNTAX:-**

drop view view\_name;

**HIVE DATA MANIPULATIONS:-**

Data Manipulation Language is used to

1. Load the data into tables
2. To extract the data from table in HDFS to directories in local machine.

**Loading Data into Managed Tables:-**

We can **load the files from our local machine into the managed or internal tables in HDFS** using the load command with the below Syntax.

**SYNTAX:-**

Load data **local** inpath ‘/…/../..’ into table table\_name;

Here the file will be taken from local machine in the specified path ‘/../../..’ and will put the data into the specified table in HDFS. Here **local** keyword denotes that **specified path of the file is in local machine.**

**Extracting Data from Tables in HDFS to directories in local machine :-**

Generally the tables will be stored under some database in HDFS. Now we can extract the stored data from the table that stored in HDFS to the directories in local machine.

**EXAMPLE QUERY:-**

Insert overwrite local directory ‘/home/acadgild/hiv’ select \* from employee;

Here the entire contents of the table employee will be extracted using the query “select \* from employee” and will be put into the directory in the local machine named ‘/home/acadgild/hiv’. After running this above query, the data will be extracted into auto created file that is named like **“000000\_0”** under local machine directory ‘/home/acadgild/hiv’.

**HiveQL Manipulations:-**

Now we are going to have a look at HiveQL.

* **Select Clause:-**

Select Clause is used to display the entire table or the specified columns of the table.

**SYNTAX TO DISPLAY THE ENTIRE TABLE CONTENTS:-**

Select \* from table\_name;

**SYNTAX TO DISPLAY THE PARTICULAR COLUMNS OF THE TABLE:-**

Select col\_name1,col\_name2**,…** from table\_name;

* **Where Clause:-**

Where clause is **used for specifying the condition.**

For example, we can display the table values for the condition.

**SYNTAX:-**

Select col\_name1,col\_name2,… from table\_name **where col\_name=value;**

* **Limit clause;**

The result of the normal query might return large number of rows. Limit clause will put upper limit on number of rows returned.

**EXAMPLE QUERY:-**

Select \* from employee limit 5;

Here this example query will display the first 5 rows of the table employee.

* **GROUP BY CLAUSE:-**

Group By Clause will group the contents of that specified column.

**SYNTAX:-**

Select column\_name from table\_name group by(col1);

* **ORDER BY:-**

Order By clause is used to perform **total order sorting**. Total order sorting means **data will be sorted across all the reducers.**

**SYNTAX:-**

Select \* from table\_name order by column\_name;

* **SORT BY:-**

Sort By wont perform total order sort. Instead **it will perform sorting only within the reducer.**

**SYNTAX:-**

Select \* from table\_name sort by column\_name;