**Hadoop\_Assignment 7.3**

**Explain the below concepts with an example in brief**

* **Hive Data Definitions**

The Driver for Apache Hive supports a broad set of DDL, including (but not limited to) the following:

* CREATE Database and DROP Database
* CREATE Table and DROP Table
* ALTER Database and USE Database
* ALTER Table and DESCRIBE statement

**Create Database**:

A database in Hive is a name space or a collection of tables.

Syntax: Create Database db\_name

Ex : hive**>** CREATE DATABASE emp;

In the above example emp is the name of the database.

hive**>** SHOW DATABASES;

Above command displays all the names of a database including the default one.

**Drop Database:**

Drop command is used to drop(data and structure) the database and checks whether the database exists or not

Syntax :DROP (DATABASE|SCHEMA) [IF EXISTS] database\_name [RESTRICT|CASCADE];

The default behavior is RESTRICT, where DROP DATABASE will fail if the database is not empty. To drop the tables in the database as well, use DROP DATABASE ... CASCADE.

Ex: hive> DROP DATABASE IF EXISTS emp;

**Create table**:

A database in Hive is a name space or a collection of tables.

Syntax: Create table table\_name

Ex : hive**>** CREATE TABLE emprec(foo INT, bar STRING);

In the above example emprec is the name of the table.Create a table called

emprec with two columns, the first being an integer and the other a string.

hive**>** SHOW tables;

Above command displays all the names of a table in the database.

**Drop table:**

Drop command is used to drop(data and structure) the database and checks whether the database exists or not

Syntax :DROP (DATABASE|SCHEMA) [IF EXISTS] table\_name [RESTRICT|CASCADE];

The default behavior is RESTRICT, where DROP DATABASE will fail if the database is not empty. To drop the tables in the database as well, use DROP DATABASE ... CASCADE.

Ex: hive> DROP DATABASE IF EXISTS emprec;

### **Use Database**

USE sets the current database for all subsequent HiveQL statements. To revert to the default database, use the keyword "default" instead of a database name. To check which database is currently being used:

Syntax : USE database\_name;

Ex: hive > USE emp;

Tables created after the above command will get stored in emp;

**ALTER TABLE Statement**

The ALTER TABLE statement changes the structure or properties of an existing table.

Syntax : ALTER TABLE old\_name RENAME TO new\_name;

To rename a table :

Ex: hive>ALTER TABLE emprec RENAME TO employee\_record;

**Describe statement**

The DESCRIBE statement displays metadata about a table, such as the column names and their data types.

Syntax: DESCRIBE [FORMATTED] table;

The DESCRIBE FORMATTED variation displays additional information, in a format familiar to users of Apache Hive.

Ex :DESCRIBE employee\_record;

It displays information such as whether the table is internal or external, when it was created, the file format, the location of the data in HDFS, whether the object is a table or a view, and (for views) the text of the query from the view definition.

* **Hive Data Manipulations**

DML refers to "Data Manipulation Language", a subset of SQL statements that modify the data stored in tables.Supports :

* LOAD
* INSERT
* Update
* Delete

**Loading files in to table**

Loading data into a Hive table is one of the variants of inserting data into a Hive table. In this method, the entire file is copied/moved to a directory that corresponds to Hive tables. If the table is partitioned, then data is loaded into partitions one at a time.

Syntax : LOAD DATA [LOCAL] INPATH 'filepath' [OVERWRITE] INTO TABLE table name [PARTITION (partcol1=val1, partcol2=val2 ...)]

Loading the data from local file system

Ex :LOAD DATA LOCAL INPATH 'E:/emp.txt' INTO TABLE employee\_record ;

Loads the data from the local system I.e from E:/emp.txt into table employee\_record.

**Insert Statement**

**Inserting data into Hive Tables from queries**

Query Results can be inserted into tables by using the insert clause.

Syntax: INSERT INTO TABLE tablename1 [PARTITION (partcol1=val1, partcol2=val2 ...)] select\_statement1 FROM from\_statement;

INSERT INTO will append to the table or partition, keeping the existing data intact

Ex: INSERT INTO TABLE pageviews PARTITION (datestamp = '2014-09-23')

   VALUES ('divya', 'mail.com', 'sports.com'), ('aks', 'mail.com', null);

**Update**

To update the existing table

Syntax: UPDATE tablename SET column = value [, column = value ...] [WHERE expression]

Ex:hive > UPDATE employee\_record SET value = 10 WHERE key = 3;

**Delete**

Deletes the data from the table.

Syntax: DELETE FROM tablename [WHERE expression]

Ex:DELETE FROM employee\_record WHERE key = 2;

* **HiveQL Manipulations**
* **Loading Data into Managed Tables.**
* **Inserting Data into Tables from Queries**
* **Creating Tables and Loading them in query**

**1.Loading Data into Managed Tables.**

Syntax Ex: LOAD DATA LOCAL INPATH

'${env:HOME}/ag-employees'

OVERWRITE INTO TABLE employees

PARTITION (country = '

INDIA', state = 'KA');

This command will first create the directory for the partition, if it doesn’t already exist,then copy the data to it.  
If the target table is not partitioned, you omit the PARTITION clause.

1. **Inserting Data into Tables from Queries**

The INSERT statement lets you load data into a table from a query.

Syntax Ex: INSERT OVERWRITE TABLE employees PARTITION (country = 'INDIA',state= 'KARNATAKA') SELECT \* FROM staged\_employees se WHERE se.cnty = 'US' AND se.st = 'OR';

this is an example for the state of KARNATAKA,

where we presume the data is already in another table called staged\_employees.

With OVERWRITE, any previous contents of the partition (or whole table if not partitioned) are replaced.

If you drop the keyword OVERWRITE or replace it with INTO, Hive appends the data rather than replaces it.

**3.Creating Tables and Loading them in query**

Creating Tables and Loading them in One Query

You can also create a table and insert query results into it in one statement:

Syntax Ex: CREATE TABLE ca\_employees AS SELECT name, salary, address

FROM employees WHERE se.state = 'KA';

This table contains just the name, salary, and address columns from the employee table records for employees in KA. The schema for the new table is taken from the

SELECT clause.

A common use for this feature is to extract a convenient subset of data from a larger,

**more unwieldy table.**