**Session 7 Assignment 3**

* **Hive Data Definitions**
* CREATE Database, Table – Used to create database and table.

**Syntax:** create database db\_name;

**Ex:** create database employee;

In the above example employee is the database name.

**Syntax:** create table table\_name;

**Ex:** create table emp (id int, name string);

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

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

* DROP Database, Table – Used to drop database and table.

**Syntax:** drop database [IF EXISTS] database\_name [RESTRICT | CASCADE];

**Ex:** drop database IF EXISTS employee;

**Syntax:** drop database [IF EXISTS] table\_name [RESTRICT|CASCADE];

**Ex:** drop database IF EXISTS emp;

The default behaviour 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.

* USE Database

USE sets the current database for all subsequent Hive QL statements. To revert to the default database, use the keyword "default" instead of a database name.

**Syntax:** USE database\_name;

**Ex:** USE emp;

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

* ALTER Table and DESCRIBE statement

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

**Syntax:** alter table old\_name RENAME TO new\_name;

**Ex:** alter table emp RENAME TO employee;

To rename a table:

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

**Syntax:** describe [FORMATTED] table;

**Ex:** describe employee;

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

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.

* LOAD

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;

**Ex:** load data LOCAL inpath '/home/acadgild/emp.txt' into table employee\_record;

Loading the data from local file system from E:/emp.txt into table employee\_record.

* INSERT

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

**Syntax:** insert into table table\_name select\_statement1 from from\_statement;

**Ex:** insert into table page\_views values ('divya', 'mail.com', 'sports.com'), ('aks', 'mail.com', null);

* Update

To update the existing table.

**Syntax:** update table\_name set column = value [, column = value ...] [where expression]

**Ex:** update employee\_record SET value = 10 WHERE key = 3;

* Delete

Deletes the data from the table.

**Syntax:** DELETE FROM table\_name [WHERE expression]

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

* **HiveQL Manipulations**
* Loading Data into Managed Tables.

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.

**Syntax:** load data local inpath

**Ex:** '${env:HOME}/ag-employees' overwrite into table employees partition (country = 'INDIA', state = 'KA');

* **Inserting Data into Tables from Queries**

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

**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.

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

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

**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.**