**ASSIGNMENT – 24.1**

**Problem Statement:**

Explain with an example in brief.

* Hive Data Definitions
* Hive Data Manipulations
* HiveQL Manipulations

**Solution:**

**1. Hive Data Definitions:**

* It includes commands that define the different structures in a database.
* HDL is a subset of Hive SQL statements that describe the data structure in Hive by creating, deleting, or altering schema objects such as databases, tables, views.
* **The commands are:**

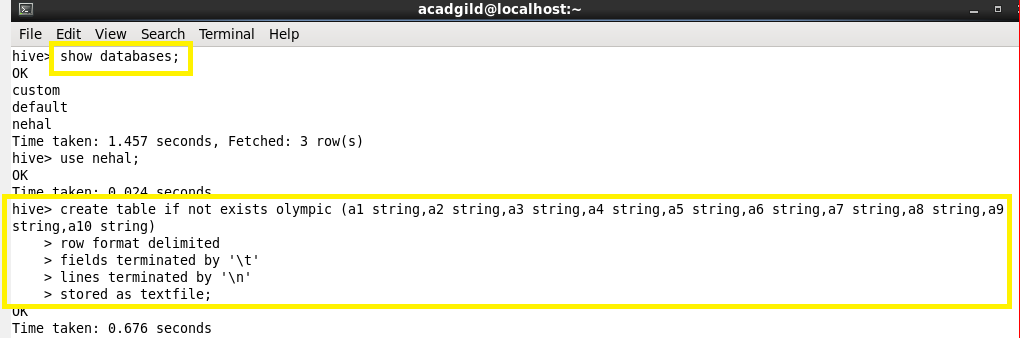
CREATE, DROP, TRUNCATE, DESCRIBE, AND ALTER.

HiveQL DDL statements are documented here, including:

* CREATE DATABASE/SCHEMA, TABLE, VIEW, FUNCTION, INDEX
* DROP DATABASE/SCHEMA, TABLE, VIEW, INDEX
* TRUNCATE TABLE
* ALTER DATABASE/SCHEMA, TABLE, VIEW
* MSCK REPAIR TABLE (or ALTER TABLE RECOVER PARTITIONS)
* SHOW DATABASES/SCHEMAS, TABLES, TBLPROPERTIES, VIEWS, PARTITIONS, FUNCTIONS, INDEX[ES], COLUMNS, CREATE TABLE
* DESCRIBE DATABASE/SCHEMA, table\_name, view\_name

**Example:**

**Create table:**



**2. Hive Data Manipulations:**

**Manipulating data is the process of exchanging, moving, sorting, and transforming the data. This technique is used in many situations, such as cleaning data, searching patterns, creating trends, and so on. Hive offers various query statements, keywords, operators, and functions to carry out data manipulation.**

* It includes commands used to modify the values in the table or to extract the data from the table.
* The commands are:
* Loading files into tables
* Inserting data into Hive tables from queries
* Inserting data into dynamic partitions
* Writing data into files from queries
* Enabling transactions in Hive
* Inserting values into tables from SQL
* Updating data
* Deleting data

**Example:**

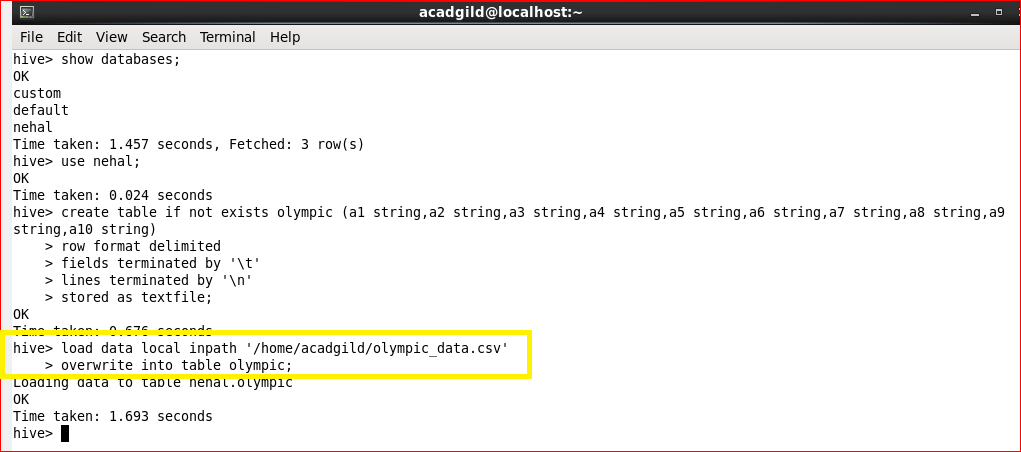
Loading files into tables

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. The general syntax of loading the data into a table is as follows:

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

Where,

* [LOCAL]: This is an optional clause. If this clause is specified, the preceding command will look for the file in the local filesystem. The command will follow the file path in the local filesystem.
* FILEPATH: This is the path where files reside either in the local filesystem or HDFS.
* [OVERWRITE]: Is an optional clause. If this clause is specified, the data in the table or partition is deleted and new data is loaded based on the file path in the statement.
* tablename: This is the name of the table.
* [PARTITION (partcol1=val1, partcol2=val2 ...)]: This is an optional clause for partitioned tables.



Inserting data into Hive tables from queries

In this recipe, you will learn how to insert data through queries into a table in Hive.

This is another variant of inserting data into a Hive table. Data can be appended into a Hive table that already contains data. Data can also be overwritten in the Hive table. Data can also be inserted into multiple tables through a single statement only. The general format of inserting data into a table from queries is as follows:

INSERT OVERWRITE TABLE tablename [PARTITION (partcol1=val1, partcol2=val2 ...) [IF NOT EXISTS]] select select\_statement FROM from\_statement;

Where:

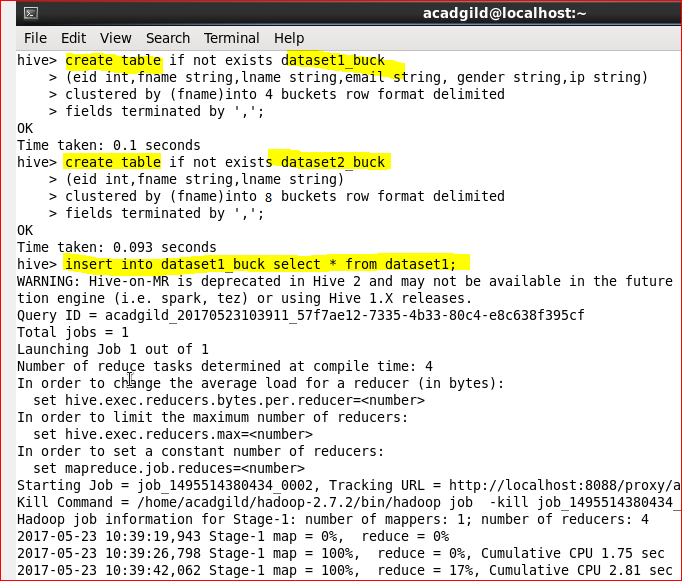
tablename: This is the name of the table

OVERWRITE: This is used to overwrite existing data in the table

[PARTITION (partcol1=val1]: This option is used when data needs to be inserted into a partitioned table

[IF NOT EXISTS]: This is an optional clause

The second syntax of inserting the data into a Hive table is as follows:



**3. HiveQL Manipulations:**

HiveQL Manipulations are of three types:

a) Select where

b) Select order by /sort by

c) Select group by

**Select where**: - used to filter the dataset with the where clause and apply the conditions.

**Select order by clause**: - used to arrange the dataset in the ascending or descending order by one field or multiple fields.

**Select group by**: - used to form subsets of the database using different fields of the data.

