Assignment 7.3

Problem Statement :-

Explain the below concepts with an example in brief.

- Hive Data Definitions
- Hive Data Manipulations
- HiveQL Manipulations

Solution:-

1. HIVE data Definition:

HIVE DATA DEFINITION LANGUAGE (DDL) 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, partitions, and buckets. Most HIVE DDL statements start with the keywords **CREATE DROP** or ALTER. The syntax of HIVE DDL is very similar to that of DDL is SQL.

Following are few operations

a. To **CREATE** table

```
CREATE TABLE Batch (

   ID INT,

   Name STRING,

   Score INT,

   Skill STRING

)
```

b. To **ALTER** table

ALTER TABLE ID RENAME TO Sno;

c. To DROP Table:

DROP TABLE IF EXISTS Batch;

2. HIVE Data Manipulation:

In Data Manipulation, we can perform following functions:

a. Loading files into Table: HIVE does not do any transformation while loading data into tables. Load operations are currently pure copy/move operations that move data files into locations corresponding to HIVE tables.

LOAD DATA INPATH 'file path' OVERWRITE INTO TABLE <table name> [PARTITION (partcol1-val1, partcol2-val2,…)]

b. Inserting data into HIVE table: Query Results can be inserted into tables by using the insert clause.

INSERT OVERWRITE TABLE <tablename1> [PARTITION (partcol1-val1, partcol2-val2,…) [IF NOT EXISTS]] select_statement1 FROM <from_statement>;

INSERT INTO TABLE <tablename1> [PARTITION (partcol1-val1, partcol2-val2,…)] <select_statement1> FROM <select_statement>;

INSERT OVERWRITE will overwrite any existing data in the table or partition.

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

c. **Writing data into the file system from queries:** Queries results can be inserted into file system directories.

INSERT OVERWRITE <LOCAL> DIRECTORY directory1 <select_statement1

d. **Inserting values into tables from SQL:**

INSERT INTO TABLE <table name> [PARTITION (partcol1 [=val1], partcol2 [=val2]…)] VALUES values_row [values_row..]

e. **Update**

UPDATE <tablename> SET column = value [, column=value…] <WHERE clause>

f. **Delete**

DELETE FROM <table name> <WHERE clause>

g. **Merge**

MERGE INTO <target table> AS T USING <source expression/table> AS S ON <Boolean expression1>

h. **Modify** : There are multiple ways to modify data in HIVE:

- ♦ LOAD
- ♦ INSERT
  - ➢ Into HIVE tables
  - ➢ Into directories
- ♦ UPDATE
- ♦ DELETE

- ♦ MERGE

3. **HQL Data Manipulation:**
   - **Loading data into managed tables**: Since HIVE has no row-level insert, update and delete operations, the only way to put data into a table is to use one of the “bulk” load operations. Or you can just write files in the correct directories by other means.

   LOAD DATA LOCAL INPATH <local file system path> OVERWIRTE INTO TABLE <table name>;

   This command will first create the directory for the partition, if it doesn’t already exist, then cope the data to it.

   - **Insert Data into tables from Queries (Dynamic partition inserts):** The INSERT statement lets you load data into a table from a query.

   INSERT OVERWRITE TABLE <table name> PARTITION (partcol1 (=val1),…) <select statement>;

   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 replacing it.

   HIVE also supports a **DYNAMIC PARTITION** feature, where it can infer the partitions to create based on query parameters.

   - **Creating tables and Loading them in One query:** You can also create a table and insert query results into it in one statement

CREATE TABLE <table name1> AS SELECT <col1, col2,…> FROM <table name> (WHERE clause)

   - **Exporting Data:** How do we get data out of tables?/ If the data files are already formatted the way you want, then it’s simple enough to copy the directories or files:

Hadoop fs –copy <source path> <target path>

Or, we can use **INSERT… DIRECTORY…,** example:

INSERT OVERWRITE LOCAL DIRECTORY <local file path> SELECT <col1,col2..> FROM <table name> (WHERE clause);

Submitted By:-

Hardik Kaushik