1. **Hive Data Definition**:

Hive Data Definition (DDL) commands are the statements that are responsible for defining and changing the structure of a database or table in Hive.

Some of DDL commands are as below:

1. **Create Database** is a statement used to create a database in Hive. A database in Hive is a **namespace** or a collection of tables. The **syntax** for this statement is as follows:

CREATE DATABASE|SCHEMA [IF NOT EXISTS] <database name>

Example:

CREATE DATABASE IF NOT EXISTS employeeDB

ii. Show databases is used to show all the databases

Example:

SHOW DATABASES

1. **Drop Database**: It is a statement that drops all the tables and deletes the database

**Example:**

DROP DATABASE IF EXISTS EMPLOYEEDB

1. **CREATE TABLE:** It is statement used to create table in Hive. Its syntax is as below:

CREATE [TEMPORARY] [EXTERNAL] TABLE [IF NOT EXISTS] [db\_name.] table\_name

[(col\_name data\_type [COMMENT col\_comment], ...)]

[COMMENT table\_comment]

[ROW FORMAT row\_format]

[STORED AS file\_format]

Example is as below:

CREATE TABLE IF NOT EXISTS employee ( eid int, name String,

salary String, destination String)

COMMENT ‘Employee details’

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ‘\t’

LINES TERMINATED BY ‘\n’

STORED AS TEXTFILE;

1. **Alter Table**: This statement is used to alter table in Hive.

Syntax is as below:

ALTER TABLE name RENAME TO new\_name

ALTER TABLE name ADD COLUMNS (col\_spec[, col\_spec ...])

ALTER TABLE name DROP [COLUMN] column\_name

ALTER TABLE name CHANGE column\_name new\_name new\_type

ALTER TABLE name REPLACE COLUMNS (col\_spec[, col\_spec ...])

Example:

ALTER TABLE employee RENAME TO emp;

1. **DROP TABLE**: It is used to drop a table

Syntax is as below:

DROP TABLE [IF EXISTS] table\_name

Exampe:

DROP TABLE IF EXISTS employee

1. **CREATE VIEW**

create view statement is used to create a The syntax is as follows:

CREATE VIEW [IF NOT EXISTS] view\_name [(column\_name [COMMENT column\_comment], ...) ]

[COMMENT table\_comment]

AS SELECT ...

Example:

CREATE VIEW emp\_30000 AS

SELECT \* FROM employee

WHERE salary>30000;

1. **Drop View**

This statement is used to drop a view. Syntax is as below:

DROP VIEW emp\_30000;

Example:

DROP VIEW emp\_30000;

1. **Create Index**

An Index is nothing but a pointer on a particular column of a table. Creating an index means creating a pointer on a particular column of a table.

Syntax is as below:

CREATE INDEX index\_name

ON TABLE base\_table\_name (col\_name, ...)

AS 'index.handler.class.name'

[WITH DEFERRED REBUILD]

[IDXPROPERTIES (property\_name=property\_value, ...)]

[IN TABLE index\_table\_name]

[PARTITIONED BY (col\_name, ...)]

[

[ ROW FORMAT ...] STORED AS ...

| STORED BY ...

]

[LOCATION hdfs\_path]

[TBLPROPERTIES (...)]

Example:

CREATE INDEX index\_salary ON TABLE employee(salary)

1. **Drop Index**

The following syntax is used to drop an index:

DROP INDEX <index\_name> ON <table\_name>

DROP INDEX index\_salary ON employee;

2. **Hive Data Manipulations**

Hive Data Manipulation ( DML) commands in Hive are used for inserting and querying the data from hive tables once the structure and architecture of the database has been defined using the DDL commands.

Data can be loaded into Hive tables using –

1. LOAD command:

LOAD Command is used for Inserting Data Into Hive Tables.

Syntax is as below:

LOAD DATA [LOCAL] INPATH ‘hdfs filepath/local filepath’ [OVERWRITE] INTO TABLE table\_name

Example:

LOAD DATA LOCAL INPATH ‘employee.txt’ OVERWRITE INTO TABLE employee

1. **SELECT**

SELECT statement is used to retrieve the data from a table. WHERE clause works similar to a condition. It filters the data using the condition and gives you a finite result. The built-in operators and functions generate an expression, which fulfils the condition.

Syntax is as below:

SELECT [ALL | DISTINCT] select\_expr, select\_expr, ...

FROM table\_reference

[WHERE where\_condition]

[GROUP BY col\_list]

[HAVING having\_condition]

[CLUSTER BY col\_list | [DISTRIBUTE BY col\_list] [SORT BY col\_list]]

[LIMIT number];

Example is as below:

SELECT \* FROM employee WHERE salary>10000;

1. JOIN

JOIN is a clause that is used for combining specific fields from two tables by using values common to each one. It is used to combine records from two or more tables in the database.

Example is as below:

SELECT c.ID, c.NAME, c.AGE, o.AMOUNT

FROM CUSTOMERS c JOIN ORDERS o

ON (c.ID = o.CUSTOMER\_ID);

1. LEFT OUTER JOIN

The HiveQL LEFT OUTER JOIN returns all the rows from the left table, even if there are no matches in the right table. This means, if the ON clause matches 0 (zero) records in the right table, the JOIN still returns a row in the result, but with NULL in each column from the right table.

Example is as below:

SELECT c.ID, c.NAME, o.AMOUNT, o.DATE

FROM CUSTOMERS c

LEFT OUTER JOIN ORDERS o

ON (c.ID = o.CUSTOMER\_ID);

1. LEFT OUTER JOIN

The HiveQL RIGHT OUTER JOIN returns all the rows from the right table, even if there are no matches in the left table. If the ON clause matches 0 (zero) records in the left table, the JOIN still returns a row in the result, but with NULL in each column from the left table.

Example is as below:

SELECT c.ID, c.NAME, o.AMOUNT, o.DATE FROM CUSTOMERS c RIGHT OUTER JOIN ORDERS o ON (c.ID = o.CUSTOMER\_ID);

1. **HIVEQL Manipulations**

Hive provides a CLI to write Hive queries using Hive Query Language (HiveQL). Generally HQL syntax is similar to the SQL syntax that most data analysts are familiar with.

Hive's SQL-inspired language separates the user from the complexity of Map Reduce programming. It reuses familiar concepts from the relational database world, such as tables, rows, columns and schema, to ease learning.

Hive provides Built-in operators for Data operations to be implemented on the tables present inside Hive warehouse.

These operators are used for mathematical operations on operands, and it will return specific value as per the logic applied.

Types of Built-in Operators in HIVE are:

Relational Operators

Arithmetic Operators

Logical Operators

Operators on Complex types

Complex type Constructors

Example:

SELECT \* FROM employee