By using Data Manipulation Language we can-

* Insert new information in database.
* Retrieve information from database.

Hive provides a CLI to write Hive queries using Hive Query Language (HiveQL). 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.

Hive Query Language, focuses on DML parts that are used to put data into tables and to extract data from tables to the console

Generally, HiveQL syntax is similar to the[SQL](https://www.guru99.com/sql.html)syntax.  Hive supports four file formats those are TEXTFILE, SEQUENCEFILE, ORC and RCFILE (Record Columnar File).

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:

1. **Relational Operators-** We use Relational operators for relationship comparisons between two operands.
2. **Arithmetic Operators-** We use Arithmetic operators for performing arithmetic operations on operands.
3. **Logical Operators-** We use Logical operators for performing Logical operations on operands.
4. **Operators on Complex types-** These are operators which will provide a different mechanism to access elements in complex types.
5. **Complex type Constructors-** It will construct instances on complex data types. These are of complex data types such as Array, Map and Struct types in Hive.

Apart from above mentioned operators we have some built-in functions also available in HIVE which can be used for manipulation purpose. Functions are built for a specific purpose to perform operations like Mathematical, arithmetic, logical and relational on the operands of table column names.

There are some built-in functions available in HIVE. Some of them are-

1. **Collection Functions-** These functions are used for collections. Collections mean the grouping of elements and returning single or array of elements depends on return type mentioned in function name.

Ex- size(Map<K.V>)🡪It returns value of INT type. It will fetch and give the components number in the map type.

1. **Date Functions-** These are used to perform Date Manipulations and Conversion of Date types from one type to another type. There are many DATE functions available.

Ex- To\_date(string timestamp)

1. **Mathematical Functions-** These functions are used for Mathematical Operations.

For Ex- ceil(DOUBLE a)

1. **Conditional Functions-** These functions used for conditional values checks.

For Ex- if(Boolean testCondition, T valueTrue, T valueFalseOrNull)🡪 It will fetch and gives value True when Test Condition is of true, gives value False Or Null otherwise.

1. **String Functions-** We can perform operations on STRING also using some STRING Functions.

For Ex- reverse(string X)🡪It return value of type STRING. It will give the reversed string of X.

Apart from above we can also define our own functions also which is called UDF (User Defined Function)-

We can specifically define UDFs in JAVA for the reusability of codes in application frameworks.

Depending on the use cases the UDFs can be written, it will accept and produce different numbers of input and output values. The general type of UDF will accept single input value and produce a single output value. If the UDF used in the query, then UDF will be called once for each row in the result data set. In the other way, it can accept a group of values as input and return single output value as well.

User Defined Functions (UDFs) in hive are used to plug in our own logic in terms of code into hive when we are not able to get the desired result from hive's built in functions.

Now apart from above mentioned features we can have a different type of Clauses associated with Hive to perform different type data manipulations and querying. For better connectivity with different nodes outside the environment, HIVE provide JDBC connectivity as well.

Hive queries provides the following features:

* Data modeling such as Creation of databases, tables, etc.
* ETL functionalities such as Extraction, Transformation, and Loading data into tables
* Joins to merge different data tables
* User specific custom scripts for ease of code
* Faster querying tool on top of Hadoop

Some of the manipulations for data extraction are-

1. **Order By Query-** Order by clause use columns on Hive tables for grouping particular column values mentioned with Order by. For whatever the column name we are defining the order by clause the query will selects and display results by ascending or descending order the particular column values.

**Ex-** SELECT \* FROM employee ORDER BY Department.

It is the query that performing on the "employee" table with the ORDER BY clause with Department as defined ORDER BY column name.

1. **Group By Query-** Group by clause use columns on Hive tables for grouping particular column values mentioned with the group by. For whatever the column name we are defining a "groupby" clause the query will select and display results by grouping the particular column values.

**Ex-** SELECT Department, count(\*) FROM employee GROUP BY Department;

 in the above example it's going to display the total count of employees present in each department. Here we have "Department" as Group by value.

1. **SORT BY:** Sort by clause performs on column names of Hive tables to sort the output. We can mention DESC for sorting the order in descending order and mention ASC for Ascending order of the sort.

In this sort by it will sort the rows before feeding to the reducer. Always sort by depends on column types.

For instance, if column types are numeric it will sort in numeric order if the columns types are string it will sort in lexicographical order.

Ex- Select \* FROM employee SORT BY id DESC;

In above example output displayed will be in descending order of "id".

1. **Cluster By-** Cluster By used as an alternative for both Distribute BY and Sort BY clauses in Hive-QL. Cluster BY clause used on tables present in Hive. Hive uses the columns in Cluster by to distribute the rows among reducers. Cluster BY columns will go to the multiple reducers. It ensures sorting orders of values present in multiple reducers.

Ex- SELECT Id, Name from employee CLUSTER BY Id;

Cluster By clause mentioned on the Id column name of the table employee table. The output when executing this query will give results to multiple reducers at the back end. But as front end it is an alternative clause for both Sort By and Distribute By. This is actually back end process when we perform a query with sort by, group by, and cluster by in terms of Map reduce framework. So if we want to store results into multiple reducers, we go with Cluster By.

1. **Distribute By-** Distribute BY clause used on tables present in Hive. Hive uses the columns in Distribute by to distribute the rows among reducers. All Distribute BY columns will go to the same reducer. It ensures each of N reducers gets non-overlapping ranges of column. It doesn't sort the output of each reducer

**Ex-** SELECT Id, Name from employees\_guru DISTRIBUTE BY Id;

Here in above query DISTRIBUTE BY Clause performs on Id of "empoloyee" table Output showing Id, Name. At back end, it will go to the same reducer