Data manipulations basically refer to the operation or processing that is done on data. Hive doesn’t have row-level insert, update or delete operations, the only way to put data into a table is to use one of the bulk load operation. Since it is very similar to MySQL so we have almost all DML facilities are available with HIVE.

Below are some of the data manipulation operations that HIVE provides-

1. **LOAD**
2. **INSERT**
3. **UPDATE**
4. **DELETE**
5. **MERGE**
6. **EXPORT**
7. **IMPORT**
8. **LOAD-** Hive does not do any transformation while loading data into tables. Load operations are currently pure copy/move operations that move datafiles into locations corresponding to Hive tables.

Below is the syntax-

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

This command will first create the directory for the partition, if it doesn’t already exist, then copy the data to it. Here the target can be a table or partition. If the OVERWRITE keyword is used then the contents of the target table (or partition) will be deleted and replaced by the files referred to by filepath; otherwise the files referred by filepath will be added to the table.

1. **INSERT-** Using INSERT we can insert data into HIVE tables from queries, into directories from queries or we can insert into HIVE tables from SQL.

Below is the syntax-

INSERT OVERWRITE TABLE tablename1 [PARTITION (partcol1=val1, partcol2=val2 ...) [IF NOT EXISTS]] select\_statement1 FROM from\_statement;

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

If we use INSERT INTO instead of INSERT OVERWRITE, it will append to the table or partition, keeping the existing data intact. We can INSERT to a table or partition as well.

We can also insert data to multiple tables as well at a time by giving multiple FROM “from statement”.

1. **UPDATE-** We can do an UPDATE on HIVE tables provided they are ACID compliant i.e. they must follow below rules-

Atomicity (an operation either succeeds completely or fails; it does not leave partial data).

Consistency (once an application performs an operation the results of that operation are visible to it in every subsequent operation).

[Isolation](https://en.wikipedia.org/wiki/Isolation_(database_systems)) (an incomplete operation by one user does not cause unexpected side effects for other users).

Durability (once an operation is complete it will be preserved even in the face of machine or system failure).

Below is the syntax-

UPDATE tablename SET column = value [, column = value ...] [WHERE expression]

Here only those rows will be updated which match the WHERE expression.

1. **DELETE-** Like UPDATE, DELETE is also supported but the table must be ACID compliant.

Below is the syntax-

DELETE FROM tablename [WHERE expression]

Here in above query only rows which match the WHERE clause will be deleted.

1. **MERGE-** This one also can be performed only on those tables that are ACID compliant. MERGE can be performed on target table based on results of a JOIN on source table.

Below is the SYNTAX-

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

ON <boolean expression1>

WHEN MATCHED [AND <boolean expression2>] THEN UPDATE SET <set clause list>

WHEN MATCHED [AND <boolean expression3>] THEN DELETE

WHEN NOT MATCHED [AND <boolean expression4>] THEN INSERT VALUES<value list>

1. **EXPORT-** The EXPORT command exports the data of a table or partition, along with the metadata, into a specified output location. This output location can then be moved over to a different Hadoop or Hive instance.

Below is the syntax-

**EXPORT TABLE *tablename* TO '*targeted export location*';**

Under the directory identified as the targeted export location, the export command will create a \_metadata file containing table definition information. If the table is un-partitioned, a subdirectory named data will be created under the target export location and within it will be copied the data for the table. If the table is partitioned, then a subdirectory for each partition is created under the targeted export location to hold that partition’s data.

1. **IMPORT-** The basic form of the Hive IMPORT command identifies the target table and the directory containing the \_metadata file and data subdirectories that will be used with the operation. This is the form of the command used when importing a table that has been exported in its entirety, i.e. not one partition at a time. The optional EXTERNAL clause indicates the newly created table should be created as an external table as opposed to a managed table:

Below is the syntax-

IMPORT TABLE *tablename* FROM '*targeted import location*';  
IMPORT EXTERNAL TABLE *tablename* FROM '*targeted import location*';

Here in above syntax the table being created should not already exist as this command will attempt to create a new table using information in the \_metadata file.