1. Objective

We are going to discuss the two types of Hive Table such as Internal Table (Managed Table) and External table. At last, we will also cover the difference between Hive Internal tables vs External Tables in this Hive tutorial.

2. Apache Hive Internal and External Tables

Hive is an open source data warehouse system used for querying and analyzing large datasets. Data in Apache Hive can be categorized into Table, Partition, and Bucket. The table in Hive is logically made up of the data being stored. Hive has two types of tables which are as follows:

- Managed Table (Internal Table)
- External Table

Hive Managed Tables-

It is also know an internal table. When we create a table in Hive, it by default manages the data. This means that Hive moves the data into its warehouse directory.

Hive External Tables-

We can also create an external table. It tells Hive to refer to the data that is at an existing location outside the warehouse directory.

Let's now discuss the Hive Internal tables vs External tables comparison.

3. Featured Difference between Hive Internal Tables vs External Tables

Here we are going to cover the comparison between Hive Internal tables vs External tables on the basis of different features. Let's discuss them one by one-

3.1. LOAD and DROP Semantics

We can see the main difference between the two table type in the LOAD and DROP semantics.

- **Managed Tables –** When we load data into a Managed table, then Hive moves data into Hive warehouse directory.

For example:

```
1. CREATE TABLE managed_table (dummy STRING);
2. LOAD DATA INPATH '/user/tom/data.txt' INTO table managed_table;
```

This moves the file hdfs://user/tom/data.txt into Hive's warehouse directory for the managed_table table, which is hdfs://user/hive/warehouse/managed_table.

Further, if we drop the table using:

```
DROP TABLE managed_table
```

Then this will delete the table metadata including its data. The data no longer exists anywhere. This is what it means for HIVE to manage the data.

- **External Tables –** External table behaves differently. In this, we can control the creation and deletion of the data. The location of the external data is specified at the table creation time:

```
1. CREATE EXTERNAL TABLE external_table(dummy STRING)
2. LOCATION '/user/tom/external_table';
3. LOAD DATA INPATH '/user/tom/data.txt' INTO TABLE external_table;
```

Now, with the EXTERNAL keyword, Apache Hive knows that it is not managing the data. So it doesn't move data to its warehouse directory. It does not even check whether the external location exists at the time it is defined. This very useful feature because it means we create the data lazily after creating the table.

The important thing to notice is that when we drop an external table, Hive will leave the data untouched and only delete the metadata.

3.2. Security

- **Managed Tables –** Hive solely controls the Managed table security. Within Hive, security needs to be managed; probably at the schema level (depends on organization).
- **External Tables –** These tables' files are accessible to anyone who has access to **HDFS** file structure. So, it needs to manage security at the HDFS file/folder level.

3.3. When to use Managed and external table

Use Managed table when –

- We want Hive to completely manage the lifecycle of the data and table.
- Data is temporary

Use External table when –

- Data is used outside of Hive. For example, the data files are read and processed by an existing program that does not lock the files.
- We are not creating a table based on the existing table.
- We need data to remain in the underlying location even after a **DROP TABLE**. This may apply if we are pointing multiple schemas at a single data set.
- The hive shouldn’t own data and control settings, directories etc., we may have another program or process that will do these things.