**Introduction**

Apache Hive is a data warehousing tool used to perform queries and analyze structured data in Apache Hadoop. It uses a SQL-like language called HiveQL.

In this tutorial, you will learn how to:

* Create Hive Managed Table
* Load Data into Hive Managed Table

To perform the HiveQL code covered in this tutorial you can create a Zeppelin note and use **hive** interpreter (**%hive**).

%hive

show databases;

**Hive Tables Types**

There are two types of tables in Hive:

* 1. The **Hive** Managed tables
  2. The **User** Managed tables.

Each table has as metadata and data. The difference is, who will manage and what. Is it Hive who will manage both or only one the metadata?

* **Managed or Internal Tables**

When you create a table in Apache hive, by default it is treated as **Managed** or **Internal** table. If you want to create an external table, you will have to use the  “**External**” keyword explicitly.

By default, Hive stores the managed tables in the warehouse folder under hive. The metadata is stored in the Hive metastore while the data will be located in a folder named after the table within the Hive data warehouse, which is essentially just a file location in HDFS.

The data warehouse location is user-configurable when Hive is installed (default location on HDFS  is **/user/hive/warehouse**). By managed or controlled we mean that Hive is taking care about the metadata and the schema. If you drop (delete) a managed table, then Hive will delete **BOTH** the Schema (the description of the table) and the data files associated with the table.

Managed tables follow ACID properties whereas you cannot perform update and delete operations on External tables (User Managed Tables). Insert operation is slow in Managed operation whereas it is fast in External tables.

You can perform any HiveQL operation on managed tables such as insert, update, and delete*. But keep in mind, when you drop a managed table,***the data along with schema will also get deleted**.

**Creating Hive Managed Tables**

Follow the steps below to create the Hive managed table and load the data into it.

The first thing you would like to do is Create a Database and Create few tables into it. Before we start with the HiveQL commands, it is good to know how HIVE stores the data. Since in HDFS everything is FILE based so HIVE stores all the information in FILEs only. By default,  in Hadoop, Hive uses **/user/hive/warehouse** as the location to store all the files/data.

A table in Hive is a set of data that uses a schema to sort the data by given identifiers.

The general syntax for creating a table in Hive is:

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

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

[COMMENT 'table\_comment']

[ROW FORMAT row\_format]

[FIELDS TERMINATED BY char]

[STORED AS file\_format];

Follow the steps below to create a table in Hive.

**Step 1: Create a Database**

1. Create a database named “tutorials” by running the **create** command:

%hive

-- Create a database

create database if not exists tutorials;

Zeppelin prints a confirmation message and the time needed to perform the action.

2. Next, verify the database is created by running the **show** command:

%hive

-- verify the database is created

show databases;

**Step 2: Create The Managed Table**

The “tutorials” database will be created if it doesn’t exists. The database will not contain any tables after initial creation. Before creating the Hive table, let’s explore the customers.csv file you will load into the Hive table.

1. The “customers.csv“ file in located on the sandbox local file system in the **/home/training/Data/northwind** directory. The file contains data about customers.

Columns description:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Column 1 | Column 2 | Column 3 | Column 4 | Column 5 | Column 6 | Column 7 | Column 8 | Column 9 | Column 10 | Column 11 |
| CustomerID | CompanyName | ContactName | ContactTitle | Address | City | Region | PostalCode | Country | Phone | Fax |

Data sample from the file:

[](http://localhost/wp-content/uploads/2023/07/customers-csv.png)

2. Arrange the data from the “customers.csv” file in columns. The column names in our example are:

* CustomerID
* CompanyName
* ContactName
* ContactTitle
* Address
* City
* Region
* PostalCode
* Country
* Phone
* Fax

3. Use column names when creating a table.

4. Create a logical schema that arranges data from the .csv file to the corresponding columns. In the “customers.csv” file, data is separated by a ','. To create a logical schema type, add the Row statement to the create table command.

Create the table by running the following command:

%hive

-- Create the customers table

create table tutorials.customers (

CustomerID string,

CompanyName string,

ContactName string,

ContactTitle string,

Address string,

city string,

Region string,

PostalCode string,

Country string,

Phone string,

Fax string)

Row format

delimited fields terminated by ',';

Zeppelin prints out a confirmation message.

%hive

-- Show table columns

desc tutorials.customers;

5. Verify if the table is created by running the **show** command:

%hive

-- verify the table is created

show tables in tutorials;

**Load Data Into Managed Tables**

When you create a Managed table you need to load its data explicitly by using the Load command. In this section we will explain how to do it.

**Step 3: Load Data From a File**

You have created a table, but it is empty because data is not loaded from the “customers.csv” file.  You need to upload the file to HDFS first. Then you can load the data into the Hive table.

1. Load the “customers.csv” file to HDFS by running the **put** command:*(This is a Linux Shell command, you need to use the %sh interpreter in Zeppelin to run this command)*

%sh

# Create a new directory on HDFS

hdfs dfs -mkdir -p /tutorials/hive/managed

# Upload the customers.csv file to HDFS

hdfs dfs -put -f /home/training/Data/northwind/customers.csv /tutorials/hive/managed

2. Check the data loaded on HDFS and show the first 10 lines using the **head** command:

%sh

#show the first ten lines from the input file

hdfs dfs -head /tutorials/hive/managed/customers.csv

3. Load data into the Hive Managed table by running the **load** command:

%hive

-- Load data into hive table

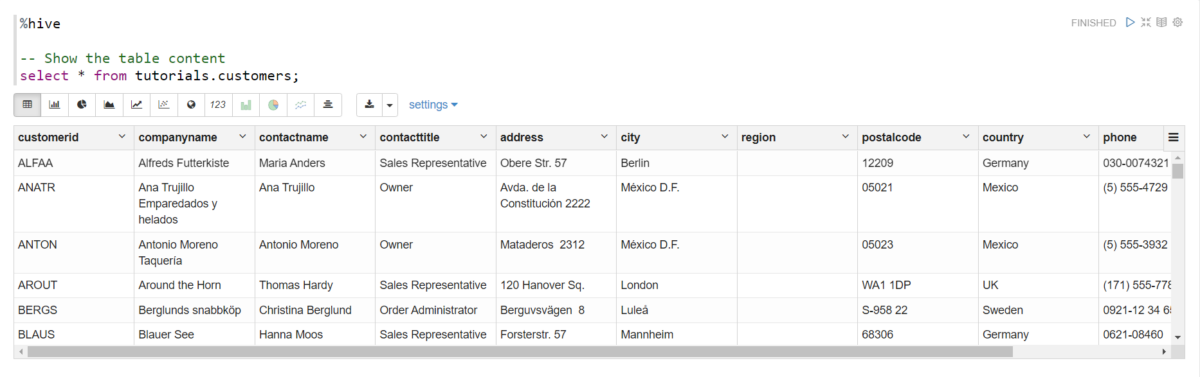
load data inpath '/tutorials/hive/managed/customers.csv' overwrite into table tutorials.customers;

4. Verify if the data is loaded by running the **select**command:

%hive

-- Show the table content

select \* from tutorials.customers;

[](http://localhost/wp-content/uploads/2023/07/select-curstomers.png)

**Show Managed Tables MetaData**

You have several options for displaying data from the table. By using the following options, you can manipulate large amounts of data more efficiently.

* **Display Columns**

You can display columns of a table by running the **desc/describe**command.

The **Describe has following variation. Followings can be given with or without giving database information.**

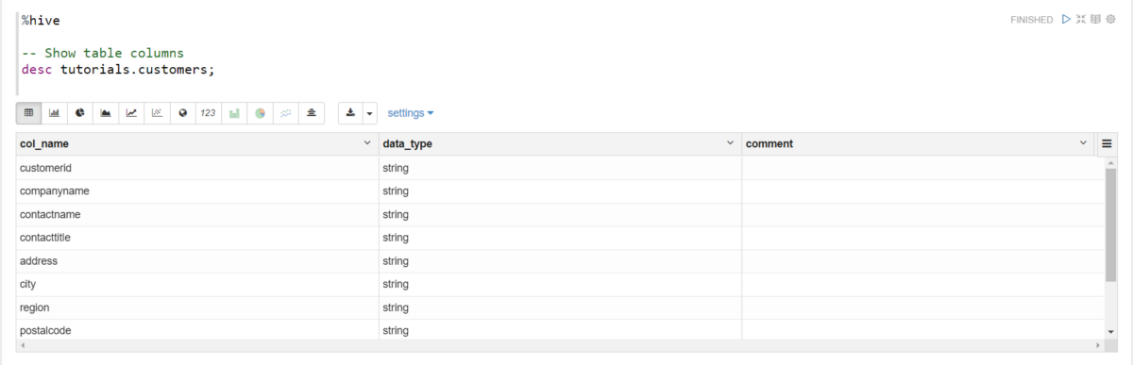
* **describe tablename:-** provides column\_name , column\_type and comment information of the table.
* **describe  formatted** **tablename** provides column\_name, column\_type and commment . It also provides  owner, location(dfs://hostname/user/hive/warehouse/database/tablename),  table\_type (EXTERNAL OR INTERNAL) , if it is Hive on HBase then it will provide information such as  hbase.tablename, serde library (org.apache.hadoop.hive.hbase.HBaseSerDe) , Hbase column mapping (rowkey, ColumnFamily:column\_name, etc….)
* **describe extended tablename**provides same information as**describe formatted table name**but it is not pretty format like the later one.
* **Show Create Table / View**

The  SHOW CREATE TABLE command shows the CREATE TABLE statement that creates a given table, or the CREATE VIEW statement that creates a given view

*Syntax :*

*show create table “your tablename”*

So the ***describe***command will give the detail information about table such as its location, number of columns and their data type etc. where as ***show create table*** provides the actual command syntax that used to create the table.

[](http://localhost/wp-content/uploads/2023/07/describe-customers.png)

**Display Selected Data**

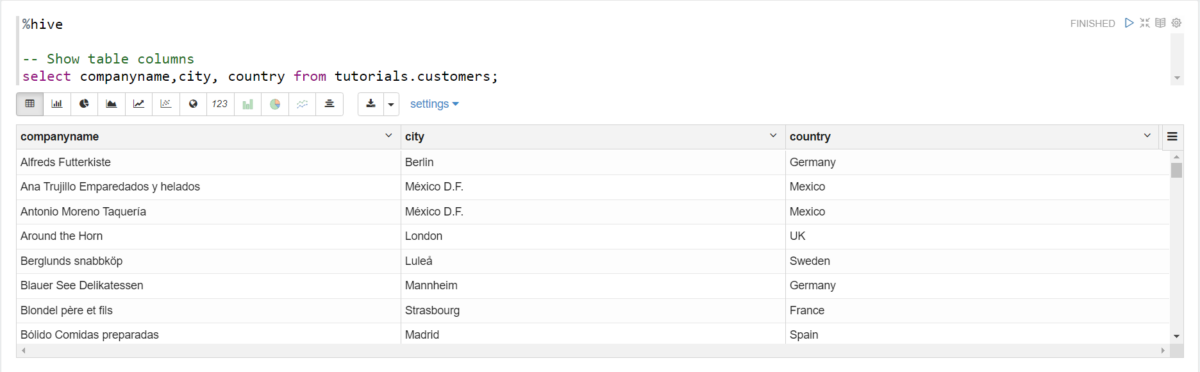
Let’s assume that you want to display company names, their city and their country of origin. Select and display data by running the **select**command:

%hive

-- Show table columns

select name,city from tutorials.customers;

The output contains the list of customers company names, city  and their country:

[](http://localhost/wp-content/uploads/2023/07/companynames-city-country.png)

**Summary**

In this tutorial we covered Hive Manage Tables basic concepts. You should have learned how to create a managed table in Hive and how to load data into it. It is important to understand how managed tables works. Keep in mind, in managed table, schema and data are managed by Hive. Managed tables follow ACID properties whereas you cannot perform update and delete operations on external tables. Insert operation is slow in managed operation whereas it is fast in external tables.

Open the Zeppelin note

[Hive Managed Tables Basics](http://localhost:19995/#/notebook/2J8B6EN2G)