CASSANDRA - DROP TABLE

http://www.tutorialspoint.com/cassandra/cassandra_drop_table.htm

Dropping a Table using Cqlsh

You can drop a table using the command **Drop Table**. Its syntax is as follows:

Syntax

```
DROP TABLE <tablename>
```

Example

The following code drops an existing table from a KeySpace.

```
cqlsh:tutorialspoint> DROP TABLE emp;
```

Verification

Use the Describe command to verify whether the table is deleted or not. Since the emp table has been deleted, you will not find it in the column families list.

```
cqlsh:tutorialspoint> DESCRIBE COLUMNFAMILIES;

employee
```

Deleting a Table using Java API

You can delete a table using the execute method of Session class. Follow the steps given below to delete a table using Java API.

Step1: Create a Cluster Object

First of all, create an instance of **Cluster.builder** class of **com.datastax.driver.core** package as shown below:

```
//Creating Cluster.Builder object
Cluster.Builder builder1 = Cluster.builder();
```

Add a contact point *IPaddressofthenode* using **addContactPoint** method of **Cluster.Builder** object. This method returns **Cluster.Builder**.

```
//Adding contact point to the Cluster.Builder object
Cluster.Builder builder2 = build.addContactPoint( "127.0.0.1" );
```

Using the new builder object, create a cluster object. To do so, you have a method called **build** in the **Cluster.Builder** class. The following code shows how to create a cluster object.

```
//Building a cluster
Cluster cluster = builder.build();
```

You can build a cluster object using a single line of code as shown below.

```
Cluster cluster = Cluster.builder().addContactPoint("127.0.0.1").build();
```

Step 2: Create a Session Object

Create an instance of Session object using the connect method of Cluster class as shown below.

```
Session session = cluster.connect( );
```

This method creates a new session and initializes it. If you already have a keyspace, you can set it to the existing one by passing the KeySpace name in string format to this method as shown below.

```
Session session = cluster.connect(“Your keyspace name”);
```

Here we are using the keyspace named **tp**. Therefore, create the session object as shown below.

```
Session session = cluster.connect(“tp”);
```

Step 3: Execute Query

You can execute CQL queries using execute method of Session class. Pass the query either in string format or as a Statement class object to the execute method. Whatever you pass to this method in string format will be executed on the **cqlsh**.

In the following example, we are deleting a table named **emp**. You have to store the query in a string variable and pass it to the execute method as shown below.

```
// Query

String query = "DROP TABLE emp1;”;
session.execute(query);
```

Given below is the complete program to drop a table in Cassandra using Java API.

```
import com.datastax.driver.core.Cluster;
import com.datastax.driver.core.Session;

public class Drop_Table {

   public static void main(String args[]){

      //Query
      String query = "DROP TABLE emp1;";
      Cluster cluster = Cluster.builder().addContactPoint("127.0.0.1").build();

      //Creating Session object
      Session session = cluster.connect("tp");

      //Executing the query
      session.execute(query);

      System.out.println("Table dropped");
   }
}
```

Save the above program with the class name followed by .java, browse to the location where it is saved. Compile and execute the program as shown below.

```
$javac Drop_Table.java
$java Drop_Table
```

Under normal conditions, it should produce the following output:

```
Table dropped
```