Deleting Data using Cqlsh

You can delete data from a table using the command **DELETE**. Its syntax is as follows:

```
DELETE FROM <identifier> WHERE <condition>;
```

**Example**

Let us assume there is a table in Cassandra called **emp** having the following data:

<table>
<thead>
<tr>
<th>emp_id</th>
<th>emp_name</th>
<th>emp_city</th>
<th>emp_phone</th>
<th>emp_sal</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ram</td>
<td>Hyderabad</td>
<td>9848022338</td>
<td>50000</td>
</tr>
<tr>
<td>2</td>
<td>robin</td>
<td>Hyderabad</td>
<td>9848022339</td>
<td>40000</td>
</tr>
<tr>
<td>3</td>
<td>rahman</td>
<td>Chennai</td>
<td>9848022330</td>
<td>45000</td>
</tr>
</tbody>
</table>

The following statement deletes the emp_sal column of last row:

```
cqlsh:tutorialspoint> DELETE emp_sal FROM emp WHERE emp_id=3;
```

**Verification**

Use SELECT statement to verify whether the data has been deleted or not. If you verify the emp table using SELECT, it will produce the following output:

```
cqlsh:tutorialspoint> select * from emp;

<table>
<thead>
<tr>
<th>emp_id</th>
<th>emp_city</th>
<th>emp_name</th>
<th>emp_phone</th>
<th>emp_sal</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Hyderabad</td>
<td>ram</td>
<td>9848022338</td>
<td>50000</td>
</tr>
<tr>
<td>2</td>
<td>Delhi</td>
<td>robin</td>
<td>9848022339</td>
<td>50000</td>
</tr>
<tr>
<td>3</td>
<td>Chennai</td>
<td>rahman</td>
<td>9848022330</td>
<td>null</td>
</tr>
</tbody>
</table>

(3 rows)
```

Since we have deleted the salary of Rahman, you will observe a null value in place of salary.

**Deleting an Entire Row**

The following command deletes an entire row from a table.

```
cqlsh:tutorialspoint> DELETE FROM emp WHERE emp_id=3;
```

**Verification**

Use SELECT statement to verify whether the data has been deleted or not. If you verify the emp table using SELECT, it will produce the following output:

```
cqlsh:tutorialspoint> select * from emp;

<table>
<thead>
<tr>
<th>emp_id</th>
<th>emp_city</th>
<th>emp_name</th>
<th>emp_phone</th>
<th>emp_sal</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Hyderabad</td>
<td>ram</td>
<td>9848022338</td>
<td>50000</td>
</tr>
<tr>
<td>2</td>
<td>Delhi</td>
<td>robin</td>
<td>9848022339</td>
<td>50000</td>
</tr>
</tbody>
</table>
```

Since we have deleted the last row, there are only two rows left in the table.

**Deleting Data using Java API**

You can delete data in a table using the execute method of Session class. Follow the steps given below to delete data from a table using java API.

**Step1: Create a Cluster Object**

Create an instance of `Cluster.builder` class of `com.datastax.driver.core` package as shown below.

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

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

```java
//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. Use the following code to create a cluster object.

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

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

```java
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.

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

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

```java
Session session = cluster.connect(" Your keyspace name ");
```

Here we are using the KeySpace called `tp`. Therefore, create the session object as shown below.

```java
Session session = cluster.connect("tp");
```

**Step 3: Execute Query**

You can execute CQL queries using the `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 data from a table named `emp`. You have to store the query in a string variable and pass it to the `execute` method as shown below.

```java
String query1 = "DELETE FROM emp WHERE emp_id=3; ";
session.execute(query1);
```
import com.datastax.driver.core.Cluster;
import com.datastax.driver.core.Session;

public class Delete_Data {
    public static void main(String args[]){
        //query
        String query = "DELETE FROM emp WHERE emp_id=3;";

        //Creating Cluster object
        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("Data deleted");
    }
}

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 Delete_Data.java
java Delete_Data

Under normal conditions, it should produce the following output:

Data deleted