

# CASSANDRA - TRUNCATE TABLE

[http://www.tutorialspoint.com/cassandra/cassandra\\_truncate\\_table.htm](http://www.tutorialspoint.com/cassandra/cassandra_truncate_table.htm)

Copyright © tutorialspoint.com

## Truncating a Table using Cqlsh

You can truncate a table using the TRUNCATE command. When you truncate a table, all the rows of the table are deleted permanently. Given below is the syntax of this command.

### Syntax

```
TRUNCATE <tablename>
```

### Example

Let us assume there is a table called **student** with the following data.

s_id	s_name	s_branch	s_aggregate
1	ram	IT	70
2	rahman	EEE	75
3	robbin	Mech	72

When you execute the select statement to get the table **student**, it will give you the following output.

```
cqlsh:tp> select * from student;

 s_id | s_aggregate | s_branch | s_name
-----+-----+-----+-----
  1 |          70 |      IT | ram
  2 |          75 |      EEE | rahman
  3 |          72 |     MECH | robbin

(3 rows)
```

Now truncate the table using the TRUNCATE command.

```
cqlsh:tp> TRUNCATE student;
```

### Verification

Verify whether the table is truncated by executing the **select** statement. Given below is the output of the select statement on the student table after truncating.

```
cqlsh:tp> select * from student;

 s_id | s_aggregate | s_branch | s_name
-----+-----+-----+-----

(0 rows)
```

## Truncating a Table using Java API

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

## Step 1: 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 the **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: Creating 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, then 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 " );
Session session = cluster.connect(" tp" );
```

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

## 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 truncating 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 = "TRUNCATE emp;;";
session.execute(query);
```

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

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

public class Truncate_Table {

    public static void main(String args[]){

        //Query
```

```
String query = "Truncate student;";

//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("Table truncated");
}
}
```

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 Truncate_Table.java
$java Truncate_Table
```

Under normal conditions, it should produce the following output:

```
Table truncated
Loading [MathJax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js
```