CASSANDRA - ALTER TABLE

Altering a Table using Cqlsh

You can alter a table using the command **ALTER TABLE**. Given below is the syntax for creating a table.

Syntax

```
ALTER (TABLE | COLUMNFAMILY) <tablename> <instruction>
```

Using ALTER command, you can perform the following operations:

- Add a column
- Drop a column

Adding a Column

Using ALTER command, you can add a column to a table. While adding columns, you have to take care that the column name is not conflicting with the existing column names and that the table is not defined with compact storage option. Given below is the syntax to add a column to a table.

```
ALTER TABLE table name
ADD  new column datatype;
```

Example

Given below is an example to add a column to an existing table. Here we are adding a column called **emp_email** of text datatype to the table named **emp**.

```
cqlsh:tutorialspoint> ALTER TABLE emp
   ... ADD emp_email text;
```

Verification

Use the SELECT statement to verify whether the column is added or not. Here you can observe the newly added column emp_email.

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

 emp_id | emp_city | emp_email | emp_name | emp_phone | emp_sal
--------+----------+-----------+----------+-----------+---------
```

Dropping a Column

Using ALTER command, you can delete a column from a table. Before dropping a column from a table, check that the table is not defined with compact storage option. Given below is the syntax to delete a column from a table using ALTER command.

```
ALTER table name
DROP column name;
```

Example

Given below is an example to drop a column from a table. Here we are deleting the column named **emp_email**.

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

Verification

Verify whether the column is deleted using the **select** statement, as shown below.

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

 emp_id | emp_city | emp_name | emp_phone | emp_sal
--------+----------+----------+-----------+---------
(0 rows)
```

Since **emp_email** column has been deleted, you cannot find it anymore.

Altering a Table using Java API

You can create a table using the execute method of Session class. Follow the steps given below to alter 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 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: 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 ” );
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 adding a column to a table named **emp.** To do so, you have to store the query in a string variable and pass it to the execute method as shown below.

```
//Query
String query1 = "ALTER TABLE emp ADD emp_email text";
session.execute(query);
```

Given below is the complete program to add a column to an existing table.

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

public class Add_column {

   public static void main(String args[]){

      //Query
      String query = "ALTER TABLE emp ADD emp_email text";

      //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("Column added");
   }
}
```

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 Add_Column.java
$java Add_Column
```

Under normal conditions, it should produce the following output:

```
Column added
```

Deleting a Column

Given below is the complete program to delete a column from an existing table.

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

public class Delete_Column {

   public static void main(String args[]){

      //Query
      String query = "ALTER TABLE emp DROP emp_email;";

      //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("Column 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_Column.java
$java Delete_Column
```

Under normal conditions, it should produce the following output:

```
Column deleted
```