HIVE - ALTER TABLE

This chapter explains how to alter the attributes of a table such as changing its table name, changing column names, adding columns, and deleting or replacing columns.

Alter Table Statement

It is used to alter a table in Hive.

Syntax

The statement takes any of the following syntaxes based on what attributes we wish to modify in a table.

```
ALTER TABLE name RENAME TO new_name
ALTER TABLE name ADD COLUMNS (col_spec[, col_spec ...])
ALTER TABLE name DROP [COLUMN] column_name
ALTER TABLE name CHANGE column_name new_name new_type
ALTER TABLE name REPLACE COLUMNS (col_spec[, col_spec ...])
```

Rename To… Statement

The following query renames the table from **employee** to **emp**.

```
hive> ALTER TABLE employee RENAME TO emp;
```

JDBC Program

The JDBC program to rename a table is as follows.

```
import java.sql.SQLException;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.Statement;
import java.sql.DriverManager;

public class HiveAlterRenameTo {
   private static String driverName = "org.apache.hadoop.hive.jdbc.HiveDriver";

   public static void main(String[] args) throws SQLException {

      // Register driver and create driver instance
      Class.forName(driverName);

      // get connection
      Connection con = DriverManager.getConnection("jdbc:hive://localhost:10000/userdb",
"", "");

      // create statement
      Statement stmt = con.createStatement();

      // execute statement
      stmt.executeQuery("ALTER TABLE employee RENAME TO emp;");
      System.out.println("Table Renamed Successfully");
      con.close();
   }
}
```

Save the program in a file named HiveAlterRenameTo.java. Use the following commands to compile and execute this program.

```
$ javac HiveAlterRenameTo.java
$ java HiveAlterRenameTo
```

Output:

```
Table renamed successfully.
```

Change Statement

The following table contains the fields of **employee** table and it shows the fields to be changed *inbold*.

Field Name	Convert from Data Type	Change Field Name	Convert to Data Type
eid	int	eid	int
name	String	**ename**	String
salary	**Float**	salary	**Double**
designation	String	designation	String

The following queries rename the column name and column data type using the above data:

```
hive> ALTER TABLE employee CHANGE name ename String;
hive> ALTER TABLE employee CHANGE salary salary Double;
```

JDBC Program

Given below is the JDBC program to change a column.

```
import java.sql.SQLException;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.Statement;
import java.sql.DriverManager;

public class HiveAlterChangeColumn {
   private static String driverName = "org.apache.hadoop.hive.jdbc.HiveDriver";

   public static void main(String[] args) throws SQLException {

      // Register driver and create driver instance
      Class.forName(driverName);

      // get connection
      Connection con = DriverManager.getConnection("jdbc:hive://localhost:10000/userdb",
"", "");

      // create statement
      Statement stmt = con.createStatement();

      // execute statement
      stmt.executeQuery("ALTER TABLE employee CHANGE name ename String;");
      stmt.executeQuery("ALTER TABLE employee CHANGE salary salary Double;");

      System.out.println("Change column successful.");
      con.close();
   }
}
```

Save the program in a file named HiveAlterChangeColumn.java. Use the following commands to compile and execute this program.

```
$ javac HiveAlterChangeColumn.java
$ java HiveAlterChangeColumn
```

Output:

```
Change column successful.
```

Add Columns Statement

The following query adds a column named dept to the employee table.

```
hive> ALTER TABLE employee ADD COLUMNS (
dept STRING COMMENT 'Department name');
```

JDBC Program

The JDBC program to add a column to a table is given below.

```
import java.sql.SQLException;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.Statement;
import java.sql.DriverManager;

public class HiveAlterAddColumn {
   private static String driverName = "org.apache.hadoop.hive.jdbc.HiveDriver";

   public static void main(String[] args) throws SQLException {

      // Register driver and create driver instance
      Class.forName(driverName);

      // get connection
      Connection con = DriverManager.getConnection("jdbc:hive://localhost:10000/userdb",
"", "");

      // create statement
      Statement stmt = con.createStatement();

      // execute statement
      stmt.executeQuery("ALTER TABLE employee ADD COLUMNS " + " (dept STRING COMMENT
'Department name');");
      System.out.prinln("Add column successful.");

      con.close();
   }
}
```

Save the program in a file named HiveAlterAddColumn.java. Use the following commands to compile and execute this program.

```
$ javac HiveAlterAddColumn.java
$ java HiveAlterAddColumn
```

Output:

```
Add column successful.
```

Replace Statement

The following query deletes all the columns from the **employee** table and replaces it with **emp** and **name** columns:

```
hive> ALTER TABLE employee REPLACE COLUMNS (
eid INT empid Int,
ename STRING name String);
```

JDBC Program

Given below is the JDBC program to replace **eid** column with **empid** and **ename** column with **name**.

```
import java.sql.SQLException;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.Statement;
import java.sql.DriverManager;

public class HiveAlterReplaceColumn {

   private static String driverName = "org.apache.hadoop.hive.jdbc.HiveDriver";

   public static void main(String[] args) throws SQLException {

      // Register driver and create driver instance
      Class.forName(driverName);

      // get connection
      Connection con = DriverManager.getConnection("jdbc:hive://localhost:10000/userdb",
"", "");

      // create statement
      Statement stmt = con.createStatement();

      // execute statement
      stmt.executeQuery("ALTER TABLE employee REPLACE COLUMNS "
         +" (eid INT empid Int,"
         +" ename STRING name String);");

      System.out.println(" Replace column successful");
      con.close();
   }
}
```

Save the program in a file named HiveAlterReplaceColumn.java. Use the following commands to compile and execute this program.

```
$ javac HiveAlterReplaceColumn.java
$ java HiveAlterReplaceColumn
```

Output:

```
Replace column successful
```