**Problem Statement:**

**● Write a Java client API to perform following operations:**

**● Use HBaseAdmin class and create the table named company with one column family name details.**

**● Create the method name insert and use put class to insert records to company table**

We can create a table in HBase using the createTable( ) method of HBaseAdmin class. This class belongs to the org.apache.hadoop.hbase.client package.

Given below are the steps to create a table in HBase using java API.

**Step1:-**

**Instantiate HBaseAdmin**

This class requires the Configuration object as a parameter, therefore initially instantiate the Configuration class and pass this instance to HBaseAdmin.

Configuration conf = HBaseConfiguration.create();

HBaseAdmin admin = new HBaseAdmin(conf);

**Step2:-**

**Create TableDescriptor**

HTableDescriptor is a class that belongs to the org.apache.hadoop.hbase class. This class is like a container of table names and column families.

//creating table descriptor

HTableDescriptor table = new HTableDescriptor(toBytes("Table name"));

//creating column family descriptor

HColumnDescriptor family = new HColumnDescriptor(toBytes("column family"));

//adding coloumn family to HTable

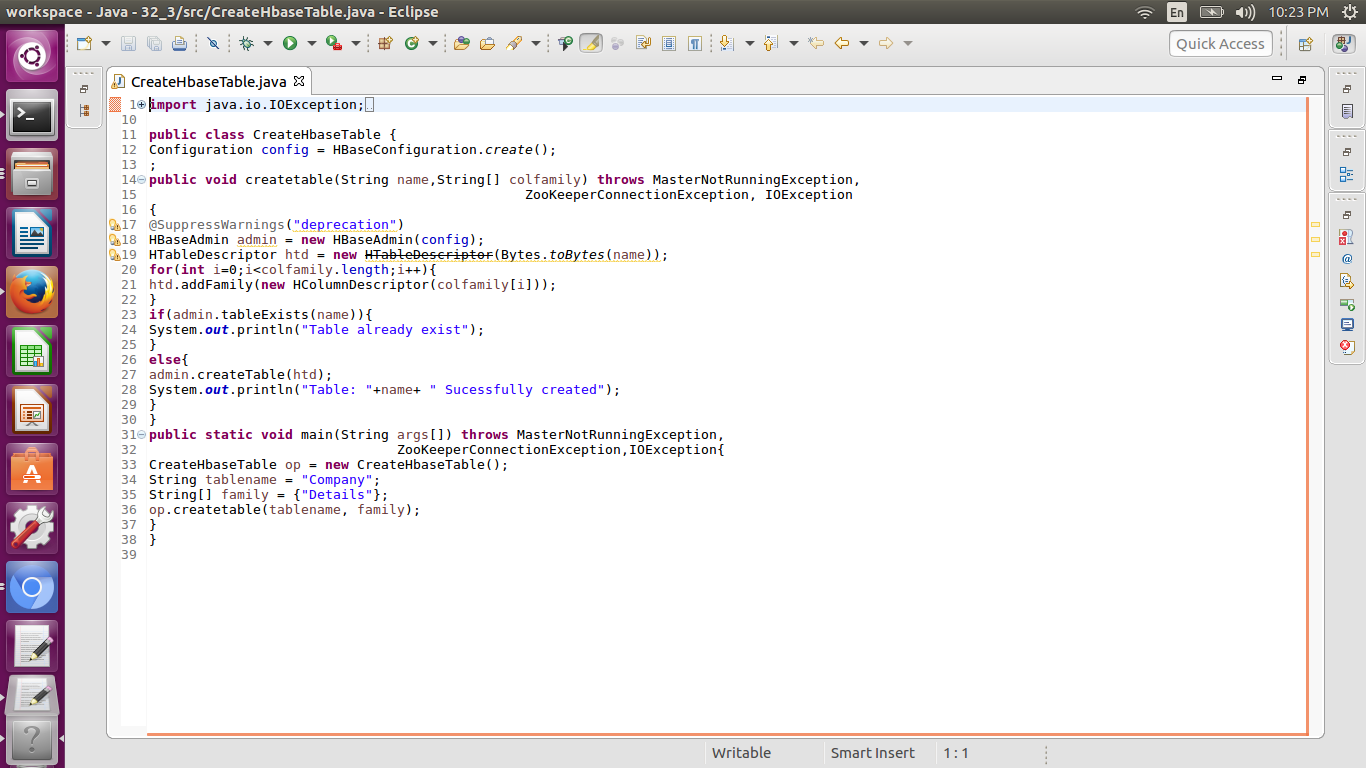
table.addFamily(family);

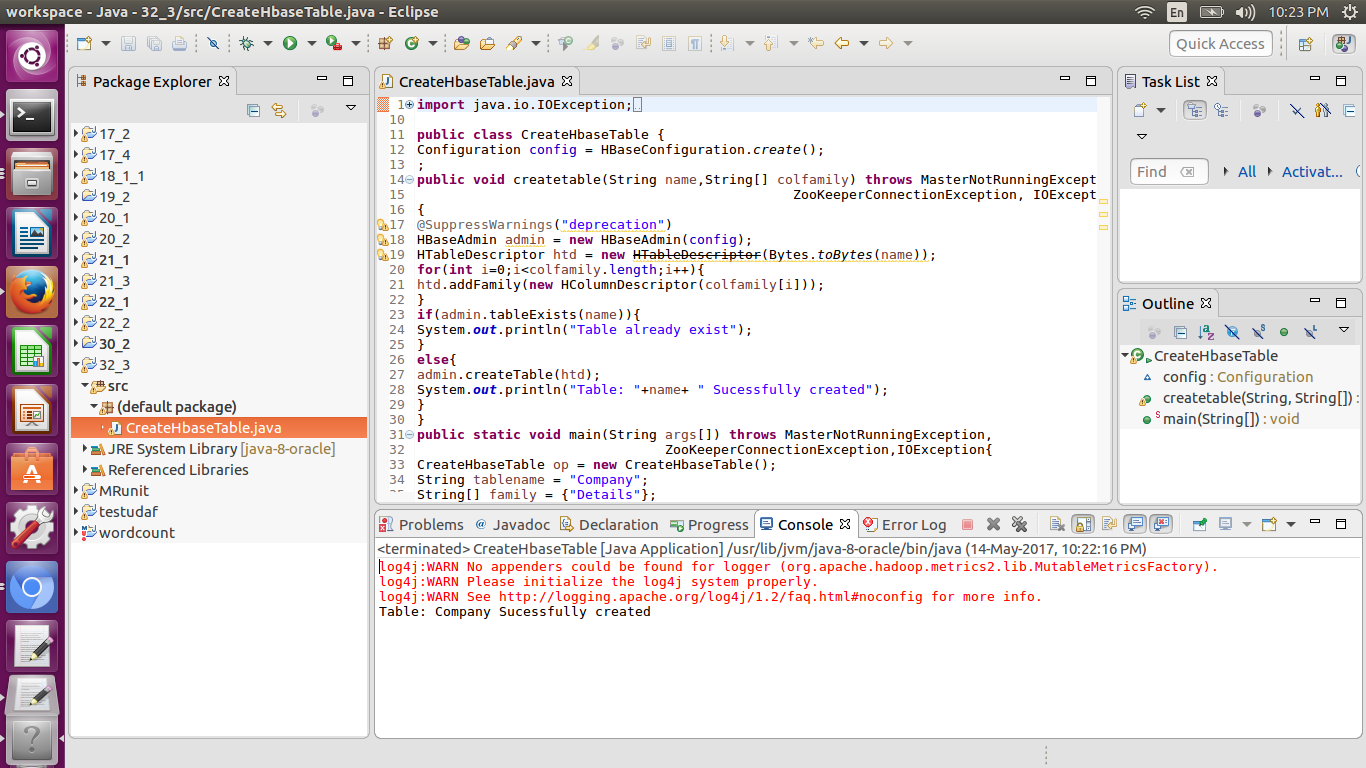
**Step 3:-**

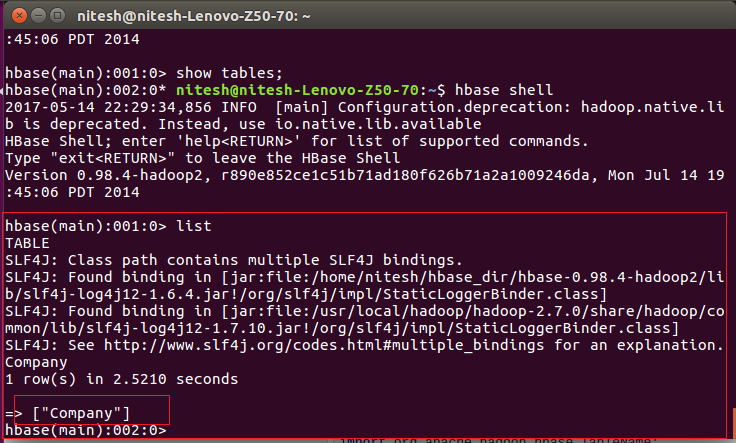
**Execute through Admin**

Using the createTable() method of HBaseAdmin class, you can execute the created table in Admin mode.

admin.createTable(table);







**● Create the method name insert and use put class to insert records to company table**

We can insert data into Hbase using the add() method of the Put class. You can save it using the put() method of the HTable class. These classes belong to the org.apache.hadoop.hbase.client package. Below given are the steps to create data in a Table of HBase.

**Step 1:-**

**Instantiate the Configuration Class**

The Configuration class adds HBase configuration files to its object. You can create a configuration object using the create() method of the HbaseConfiguration class as shown below.

Configuration conf = HbaseConfiguration.create();

**Step 2:-**

**Instantiate the HTable Class**

You have a class called HTable, an implementation of Table in HBase. This class is used to communicate with a single HBase table. While instantiating this class, it accepts configuration object and table name as parameters. You can instantiate HTable class as shown below.

HTable hTable = new HTable(conf, tableName);

**Step 3: -**

**Instantiate the PutClass**

To insert data into an HBase table, the add() method and its variants are used. This method belongs to Put, therefore instantiate the put class. This class requires the row name you want to insert the data into, in string format. You can instantiate the Put class as shown below.

Put p = new Put(Bytes.toBytes("row1"));

**Step 4:-**

**Insert Data**

The add() method of Put class is used to insert data. It requires 3 byte arrays representing column family, column qualifier (column name), and the value to be inserted, respectively. Insert data into the HBase table using the add() method as shown below.

p.add(Bytes.toBytes("coloumn family "), Bytes.toBytes("column

name"),Bytes.toBytes("value"));

**Step 5:-**

**Save the Data in Table**

After inserting the required rows, save the changes by adding the put instance to the put() method of HTable class as shown below.

hTable.put(p);

**Step 6:-**

**Close the HTable Instance**

After creating data in the HBase Table, close the HTable instance using the close() method as shown below.

hTable.close();

