

JAVA EXAMPLES - EXECUTE MULTIPLE SQL STATEMENTS

http://www.tutorialspoint.com/javaexamples/jdbc_executebatch.htm

Copyright © tutorialspoint.com

Problem Description:

How to execute multiple SQL commands on a database simultaneously?

Solution:

Following example uses addBatch & executeBatch commands to execute multiple SQL commands simultaneously.

```
import java.sql.*;

public class jdbcConn {
    public static void main(String[] args) throws Exception{
        Class.forName("org.apache.derby.jdbc.ClientDriver");
        Connection con = DriverManager.getConnection
            ("jdbc:derby://localhost:1527/testDb", "name", "pass");
        Statement stmt = con.createStatement
            (ResultSet.TYPE_SCROLL_SENSITIVE,
            ResultSet.CONCUR_UPDATABLE);
        String insertEmp1 = "insert into emp values
            (10, 'jay', 'trainee')";
        String insertEmp2 = "insert into emp values
            (11, 'jayes', 'trainee')";
        String insertEmp3 = "insert into emp values
            (12, 'shail', 'trainee')";
        con.setAutoCommit(false);
        stmt.addBatch(insertEmp1);
        stmt.addBatch(insertEmp2);
        stmt.addBatch(insertEmp3);
        ResultSet rs = stmt.executeQuery("select * from emp");
        rs.last();
        System.out.println("rows before batch execution= "
            + rs.getRow());
        stmt.executeBatch();
        con.commit();
        System.out.println("Batch executed");
        rs = stmt.executeQuery("select * from emp");
        rs.last();
        System.out.println("rows after batch execution= "
            + rs.getRow());
    }
}
```

Result:

The above code sample will produce the following result. The result may vary.

```
rows before batch execution= 6
Batch executed
rows after batch execution= 9
```