```
import com.ephesoft.dcma.script.IJDomScript;
import com.ephesoft.dcma.util.logger.EphesoftLogger;
import com.ephesoft.dcma.util.logger.ScriptLoggerFactory;
import org.jdom.Document;
import org.jdom.Element;
import org.jdom.input.SAXBuilder;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;

/**
 * @author Ephesoft
 * @version 1.0
 */

public class ScriptDocumentAssembler implements IJDomScript {

    private static final EphesoftLogger LOGGER = 
ScriptLoggerFactory.getLogger(ScriptDocumentAssembler.class);

    public Object execute(Document document, String methodName, String 
docIdentifier) {

        Exception exception = null;
        try {

            LOGGER.info("------> Start execution of the 
ScriptDocumentAssembler.java <------");

            if (null == document) {
                LOGGER.error("###### Batch Instance XML does not exist ######");
            } else {
                  Element root = document.getRootElement();
                  String batchInstance = 
root.getChildText("BatchInstanceIdentifier");

                        fillCustomColumns(batchInstance);
                LOGGER.info("------> End execution of the 
ScriptDocumentAssembler.java <------");
            }
        } catch (Exception e) {
            LOGGER.error("######## Error in the execution of 
ScriptDocumentAssembler.java >>>> ", e.getMessage());
            e.printStackTrace();
            exception = e;
        }
        return exception;
    }

      private void fillCustomColumns(String batchId) {

            String SQL_USER = "ephesoft";
            String SQL_PASSWORD = "P@ssw0rd";
            String SQL_DRIVER = "net.sourceforge.jtds.jdbc.Driver";
            String DATABASE_NAME = "Ephesoft";
            String SQL_SERVER = "localhost";
            String PORT = "1433";
            String SQL_CONNECTION_STRING = "jdbc:jtds:sqlserver://" + SQL_SERVER +
":" + PORT + "/" + DATABASE_NAME;

            PreparedStatement updateStatement = null;
            Connection connection = null;

            String SQLUpdate = "UPDATE batch_instance SET custom_column1=?,
custom_column2=?, custom_column3=?, custom_column4=? WHERE identifier=?";

            try {
                  Class.forName(SQL_DRIVER);

                  connection = DriverManager.getConnection(SQL_CONNECTION_STRING,
SQL_USER, SQL_PASSWORD);
                  updateStatement = connection.prepareStatement(SQLUpdate);
                  updateStatement.setString(1, "This");
                  updateStatement.setString(2, "Is");
                  updateStatement.setString(3, "A");
                  updateStatement.setString(4, "Test");
                  updateStatement.setString(5, batchId);
                  updateStatement.executeUpdate();

            } catch (SQLException sql_e) {
                  LOGGER.error("Unable to insert batch fields into DB - sql
exception >>>> " + new Object(){}.getClass().getEnclosingMethod().getName() + "
>>>> " + sql_e.getMessage());
                  sql_e.printStackTrace();
            } catch (Exception e) {
                  LOGGER.error("Unable to insert batch fields into DB - general
exception >>>> " + new Object(){}.getClass().getEnclosingMethod().getName() + "
>>>> " + e.getMessage());
                  e.printStackTrace();
            } finally {
                  if(updateStatement != null){
                        try {
                              updateStatement.close();
                        } catch (SQLException e) {
                              LOGGER.error("Exception in " + new Object()
{}.getClass().getEnclosingMethod().getName() + " >>>> " + e.getMessage());
                              e.printStackTrace();
                        }
                  }
                  if(connection != null){
                        try {
                              connection.close();
                        } catch (SQLException e) {
                              LOGGER.error("Exception in " + new Object()
{}.getClass().getEnclosingMethod().getName() + " >>>> " + e.getMessage());
                              e.printStackTrace();
                        }
                  }
            }
      }

    public static void main(String[] args) {

        String filePath = "";
        try {
            SAXBuilder sb = new SAXBuilder();
            Document doc = sb.build(filePath);
            new ScriptDocumentAssembler().execute(doc, null, null);
        } catch (Exception ignored) {
        }
    }
}
```