JDBC API

ID | 4-36-55-56 |

Names :

محمد مجدى محمد عبدالغنى
محمد صلاح عبدالرازق عبدالمقصود
عبدالرحمن ابراهيم على ابراهيم الهنداوى
احمد حمدى ابراهيم رضوان

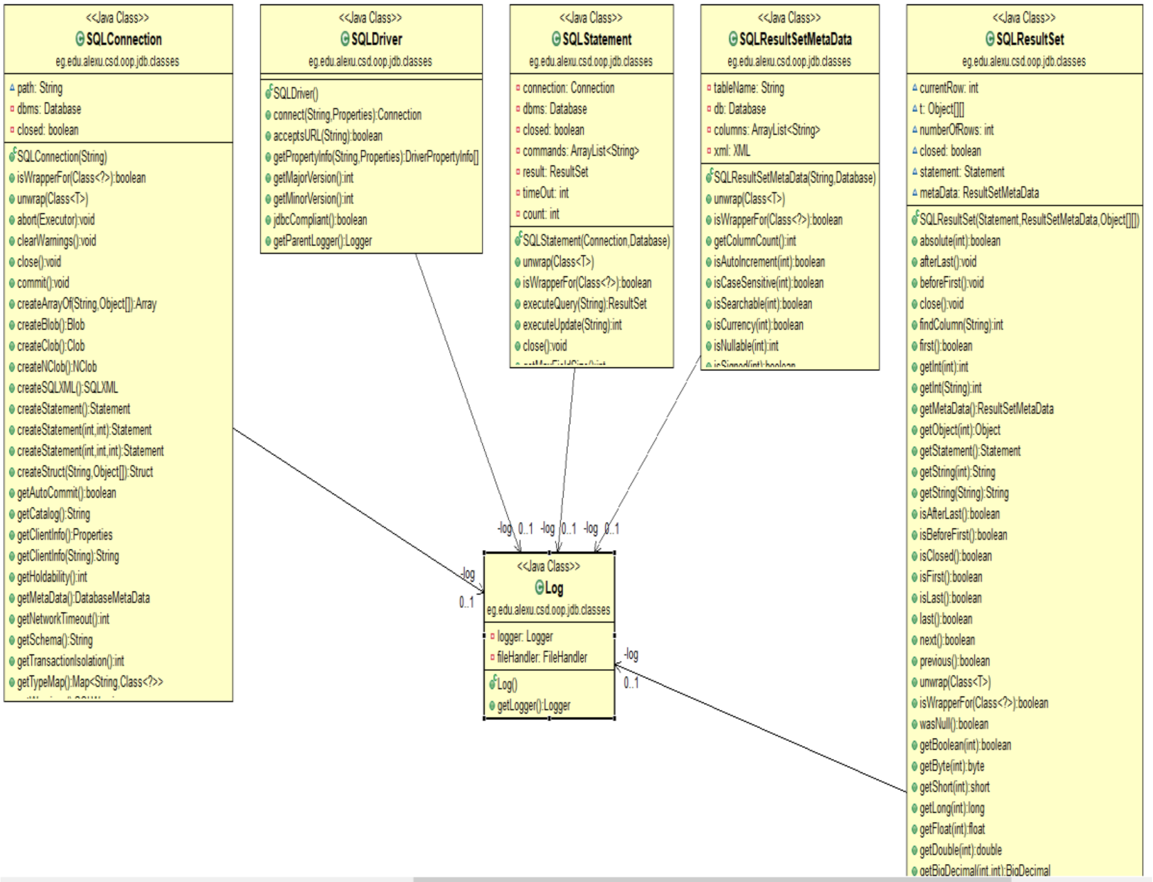
Description

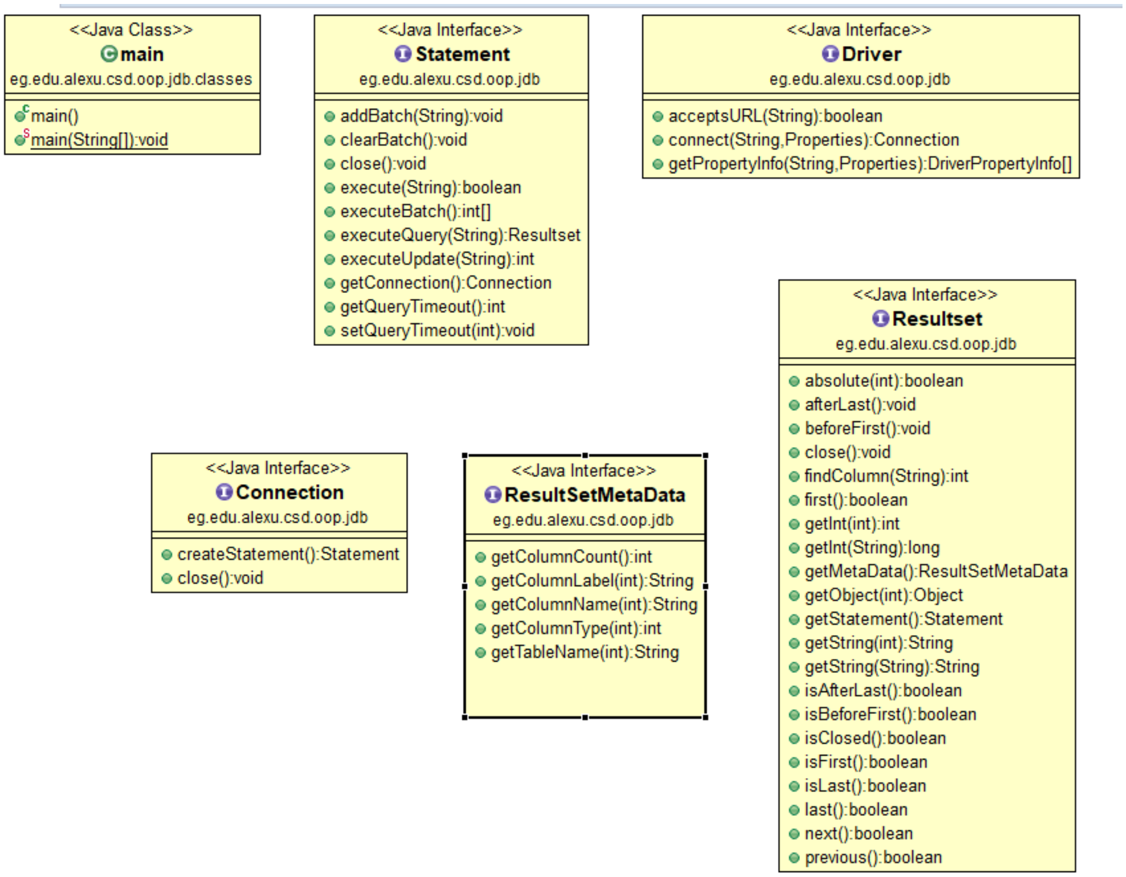
Java Database Connectivity (JDBC) provides Java developers with a standard API that is used to access

databases, regardless of the driver and database product. JDBC presents a uniform interface to databases

- change vendors and your applications only need to change their driver.

UML





USER GUIDE :

User must choose from UI interface one of four choices :

1 – if user choose ‘1’ he will allowed to add queries to the batch then he can enter the SQL commands .

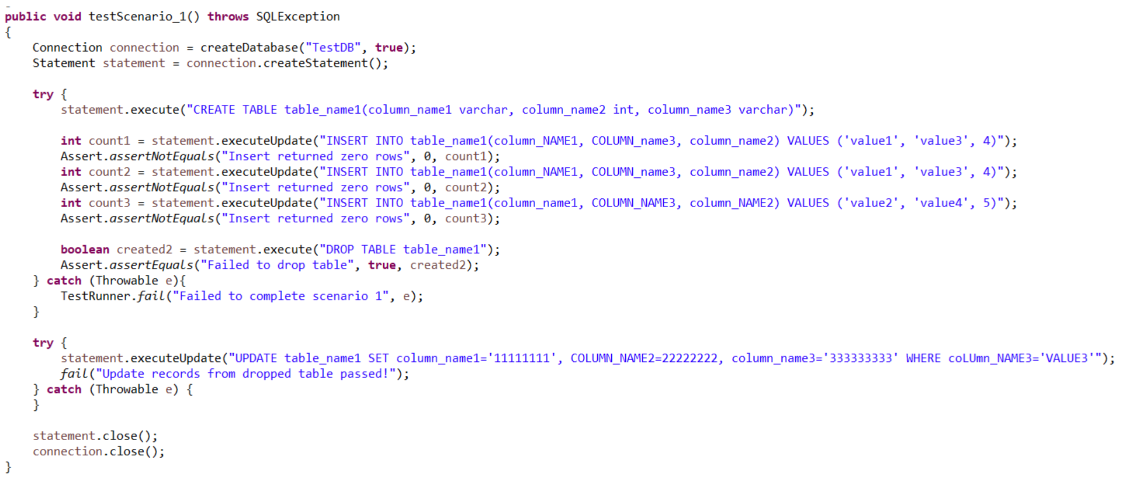
2 – if he wants to terminate his choice and make his commands to be executed he must enter ‘2’ then ‘3’ to make all the queries execute.

3 – if he wants to exit enter ‘-1’.

User can use any SQL commands create, delete, drop, update, insert and select.

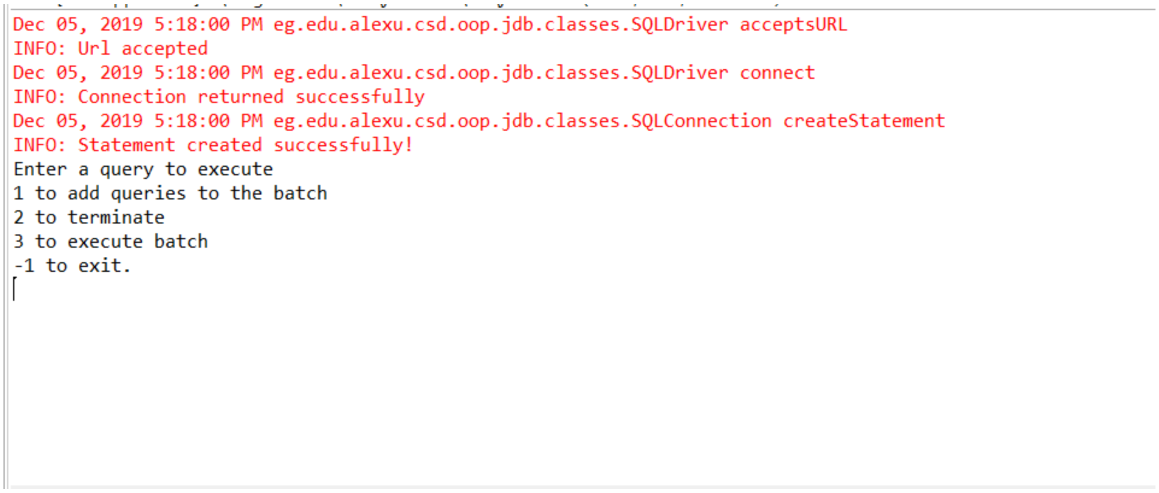
SNAPSHOTS OF UI :

HERE TEST THIS SCENARIO IN TH UI INTERFACE



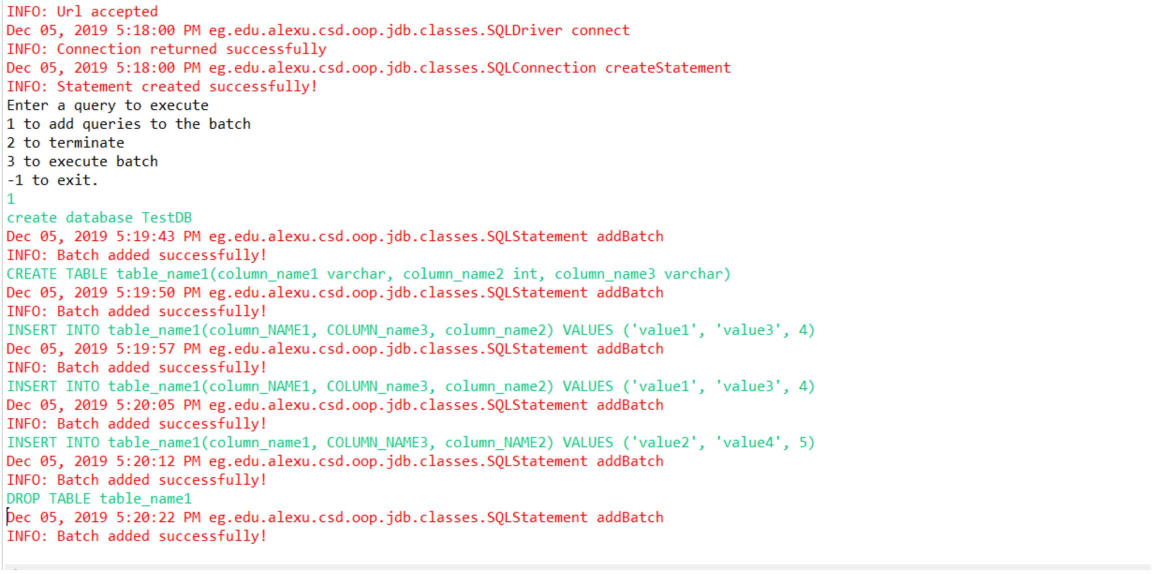
```java
public void testScenario_1() throws SQLException
{
    Connection connection = createDatabase("TestDB", true);
    Statement statement = connection.createStatement();

    try {
        statement.execute("CREATE TABLE table_name1(column_name1 varchar, column_name2 int, column_name3 varchar)");

        int count1 = statement.executeUpdate("INSERT INTO table_name1(column_NAME1, COLUMN_name3, column_name2) VALUES ('value1', 'value3', 4)");
        Assert.assertNotEquals("Insert returned zero rows", 0, count1);
        int count2 = statement.executeUpdate("INSERT INTO table_name1(column_NAME1, COLUMN_name3, column_name2) VALUES ('value1', 'value3', 4)");
        Assert.assertNotEquals("Insert returned zero rows", 0, count2);
        int count3 = statement.executeUpdate("INSERT INTO table_name1(column_name1, COLUMN_NAME3, column_NAME2) VALUES ('value2', 'value4', 5)");
        Assert.assertNotEquals("Insert returned zero rows", 0, count3);

        boolean created2 = statement.execute("DROP TABLE table_name1");
        Assert.assertEquals("Failed to drop table", true, created2);
    } catch (Throwable e){
        TestRunner.fail("Failed to complete scenario 1", e);
    }

    try {
        statement.executeUpdate("UPDATE table_name1 SET column_name1='11111111', COLUMN_NAME2=22222222, column_name3='333333333' WHERE coLUmn_NAME3='VALUE3'");
        fail("Update records from dropped table passed!");
    } catch (Throwable e) {
    }

    statement.close();
    connection.close();
}
```

First the connection will be shown to the user with data and time



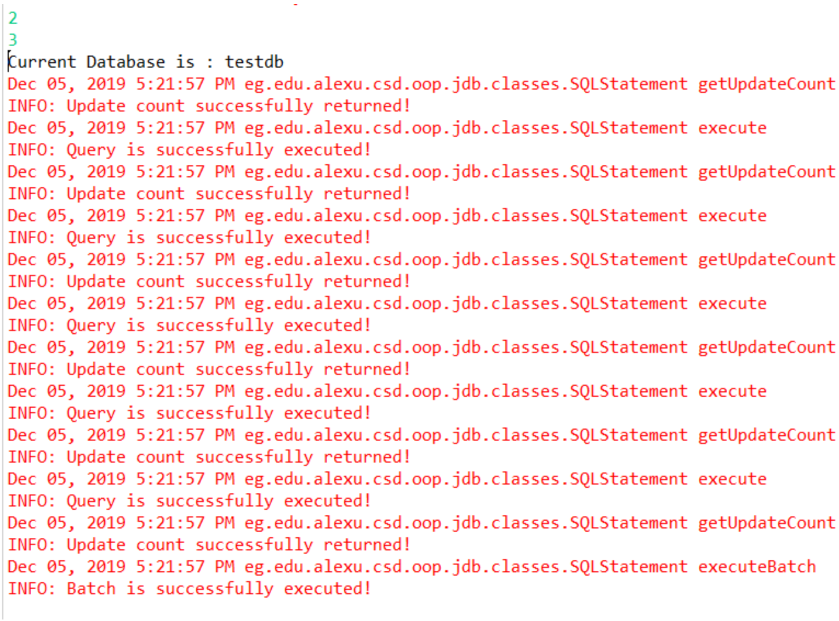
```
Dec 05, 2019 5:18:00 PM eg.edu.alexu.csd.oop.jdb.classes.SQLDriver acceptsURL
INFO: Url accepted
Dec 05, 2019 5:18:00 PM eg.edu.alexu.csd.oop.jdb.classes.SQLDriver connect
INFO: Connection returned successfully
Dec 05, 2019 5:18:00 PM eg.edu.alexu.csd.oop.jdb.classes.SQLConnection createStatement
INFO: Statement created successfully!
Enter a query to execute
1 to add queries to the batch
2 to terminate
3 to execute batch
-1 to exit.
```

Then choose '1' to enter the commands



```
INFO: Url accepted
Dec 05, 2019 5:18:00 PM eg.edu.alexu.csd.oop.jdb.classes.SQLDriver connect
INFO: Connection returned successfully
Dec 05, 2019 5:18:00 PM eg.edu.alexu.csd.oop.jdb.classes.SQLConnection createStatement
INFO: Statement created successfully!
Enter a query to execute
1 to add queries to the batch
2 to terminate
3 to execute batch
-1 to exit.
1
create database TestDB
Dec 05, 2019 5:19:43 PM eg.edu.alexu.csd.oop.jdb.classes.SQLStatement addBatch
INFO: Batch added successfully!
CREATE TABLE table_name1(column_name1 varchar, column_name2 int, column_name3 varchar)
Dec 05, 2019 5:19:50 PM eg.edu.alexu.csd.oop.jdb.classes.SQLStatement addBatch
INFO: Batch added successfully!
INSERT INTO table_name1(column_NAME1, COLUMN_name3, column_name2) VALUES ('value1', 'value3', 4)
Dec 05, 2019 5:19:57 PM eg.edu.alexu.csd.oop.jdb.classes.SQLStatement addBatch
INFO: Batch added successfully!
INSERT INTO table_name1(column_NAME1, COLUMN_name3, column_name2) VALUES ('value1', 'value3', 4)
Dec 05, 2019 5:20:05 PM eg.edu.alexu.csd.oop.jdb.classes.SQLStatement addBatch
INFO: Batch added successfully!
INSERT INTO table_name1(column_name1, COLUMN_NAME3, column_NAME2) VALUES ('value2', 'value4', 5)
Dec 05, 2019 5:20:12 PM eg.edu.alexu.csd.oop.jdb.classes.SQLStatement addBatch
INFO: Batch added successfully!
DROP TABLE table_name1
Dec 05, 2019 5:20:22 PM eg.edu.alexu.csd.oop.jdb.classes.SQLStatement addBatch
INFO: Batch added successfully!
```

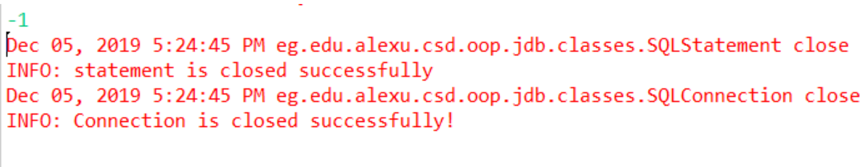
Then choose '2' to terminate and '3' to execute the commands



```
2
3
Current Database is : testdb
Dec 05, 2019 5:21:57 PM eg.edu.alexu.csd.oop.jdb.classes.SQLStatement getUpdateCount
INFO: Update count successfully returned!
Dec 05, 2019 5:21:57 PM eg.edu.alexu.csd.oop.jdb.classes.SQLStatement execute
INFO: Query is successfully executed!
Dec 05, 2019 5:21:57 PM eg.edu.alexu.csd.oop.jdb.classes.SQLStatement getUpdateCount
INFO: Update count successfully returned!
Dec 05, 2019 5:21:57 PM eg.edu.alexu.csd.oop.jdb.classes.SQLStatement execute
INFO: Query is successfully executed!
Dec 05, 2019 5:21:57 PM eg.edu.alexu.csd.oop.jdb.classes.SQLStatement getUpdateCount
INFO: Update count successfully returned!
Dec 05, 2019 5:21:57 PM eg.edu.alexu.csd.oop.jdb.classes.SQLStatement execute
INFO: Query is successfully executed!
Dec 05, 2019 5:21:57 PM eg.edu.alexu.csd.oop.jdb.classes.SQLStatement getUpdateCount
INFO: Update count successfully returned!
Dec 05, 2019 5:21:57 PM eg.edu.alexu.csd.oop.jdb.classes.SQLStatement execute
INFO: Query is successfully executed!
Dec 05, 2019 5:21:57 PM eg.edu.alexu.csd.oop.jdb.classes.SQLStatement getUpdateCount
INFO: Update count successfully returned!
Dec 05, 2019 5:21:57 PM eg.edu.alexu.csd.oop.jdb.classes.SQLStatement execute
INFO: Query is successfully executed!
Dec 05, 2019 5:21:57 PM eg.edu.alexu.csd.oop.jdb.classes.SQLStatement getUpdateCount
INFO: Update count successfully returned!
Dec 05, 2019 5:21:57 PM eg.edu.alexu.csd.oop.jdb.classes.SQLStatement executeBatch
INFO: Batch is successfully executed!
```

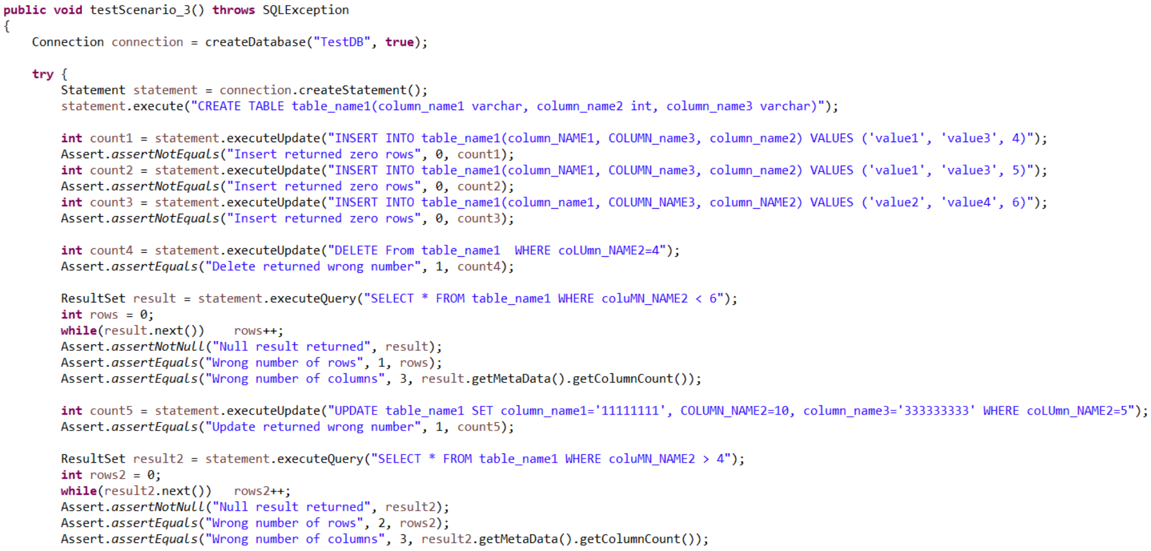
All the quieries information will be displayed as log info or warning and if there is warning will be displayed with SQLException .

And then if user want to close the connection enter '-1'

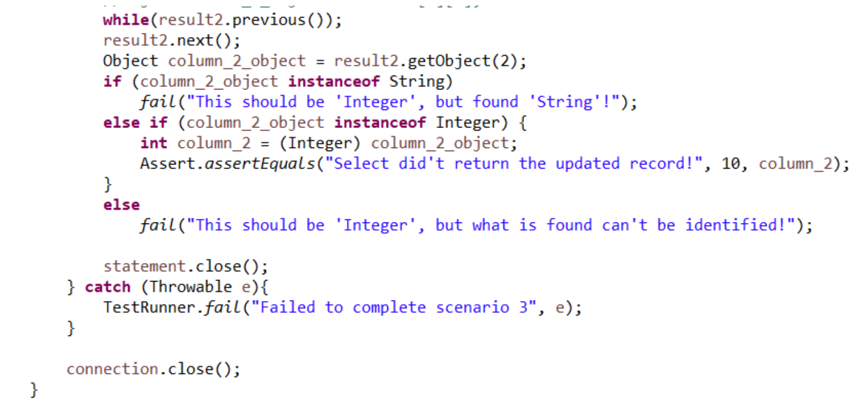


```
-1
Dec 05, 2019 5:24:45 PM eg.edu.alexu.csd.oop.jdb.classes.SQLStatement close
INFO: statement is closed successfully
Dec 05, 2019 5:24:45 PM eg.edu.alexu.csd.oop.jdb.classes.SQLConnection close
INFO: Connection is closed successfully!
```

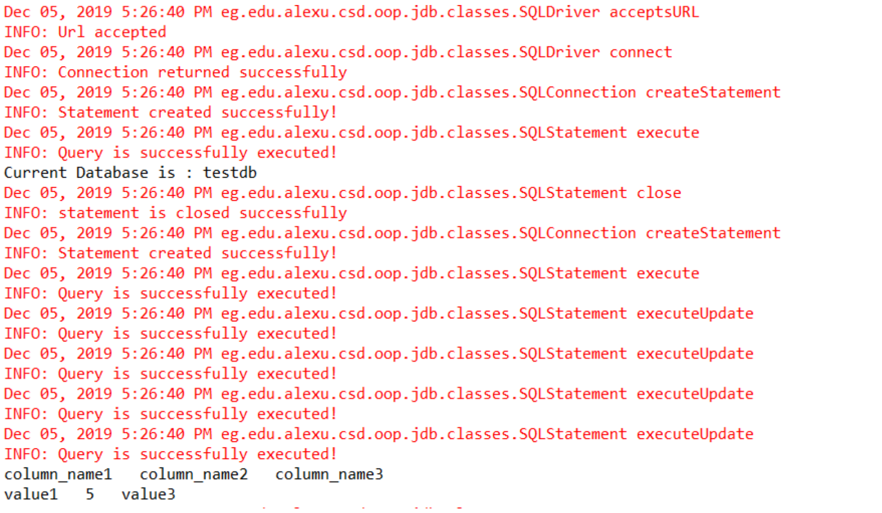
Test scenario 3



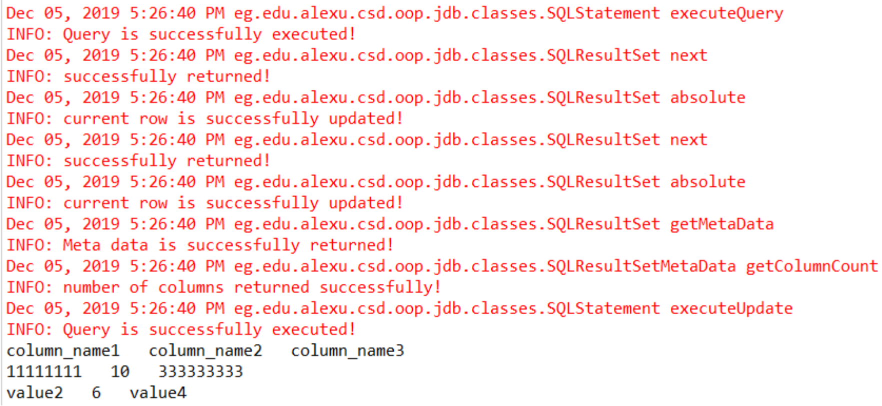
```java
public void testScenario_3() throws SQLException
{
    Connection connection = createDatabase("TestDB", true);

    try {
        Statement statement = connection.createStatement();
        statement.execute("CREATE TABLE table_name1(column_name1 varchar, column_name2 int, column_name3 varchar)");

        int count1 = statement.executeUpdate("INSERT INTO table_name1(column_NAME1, COLUMN_name3, column_name2) VALUES ('value1', 'value3', 4)");
        Assert.assertNotEquals("Insert returned zero rows", 0, count1);
        int count2 = statement.executeUpdate("INSERT INTO table_name1(column_NAME1, COLUMN_name3, column_name2) VALUES ('value1', 'value3', 5)");
        Assert.assertNotEquals("Insert returned zero rows", 0, count2);
        int count3 = statement.executeUpdate("INSERT INTO table_name1(column_name1, COLUMN_NAME3, column_NAME2) VALUES ('value2', 'value4', 6)");
        Assert.assertNotEquals("Insert returned zero rows", 0, count3);

        int count4 = statement.executeUpdate("DELETE From table_name1  WHERE coLUmn_NAME2=4");
        Assert.assertEquals("Delete returned wrong number", 1, count4);

        ResultSet result = statement.executeQuery("SELECT * FROM table_name1 WHERE coluMN_NAME2 < 6");
        int rows = 0;
        while(result.next())    rows++;
        Assert.assertNotNull("Null result returned", result);
        Assert.assertEquals("Wrong number of rows", 1, rows);
        Assert.assertEquals("Wrong number of columns", 3, result.getMetaData().getColumnCount());

        int count5 = statement.executeUpdate("UPDATE table_name1 SET column_name1='11111111', COLUMN_NAME2=10, column_name3='333333333' WHERE coLUmn_NAME2=5");
        Assert.assertEquals("Update returned wrong number", 1, count5);

        ResultSet result2 = statement.executeQuery("SELECT * FROM table_name1 WHERE coluMN_NAME2 > 4");
        int rows2 = 0;
        while(result2.next())   rows2++;
        Assert.assertNotNull("Null result returned", result2);
        Assert.assertEquals("Wrong number of rows", 2, rows2);
        Assert.assertEquals("Wrong number of columns", 3, result2.getMetaData().getColumnCount());
```



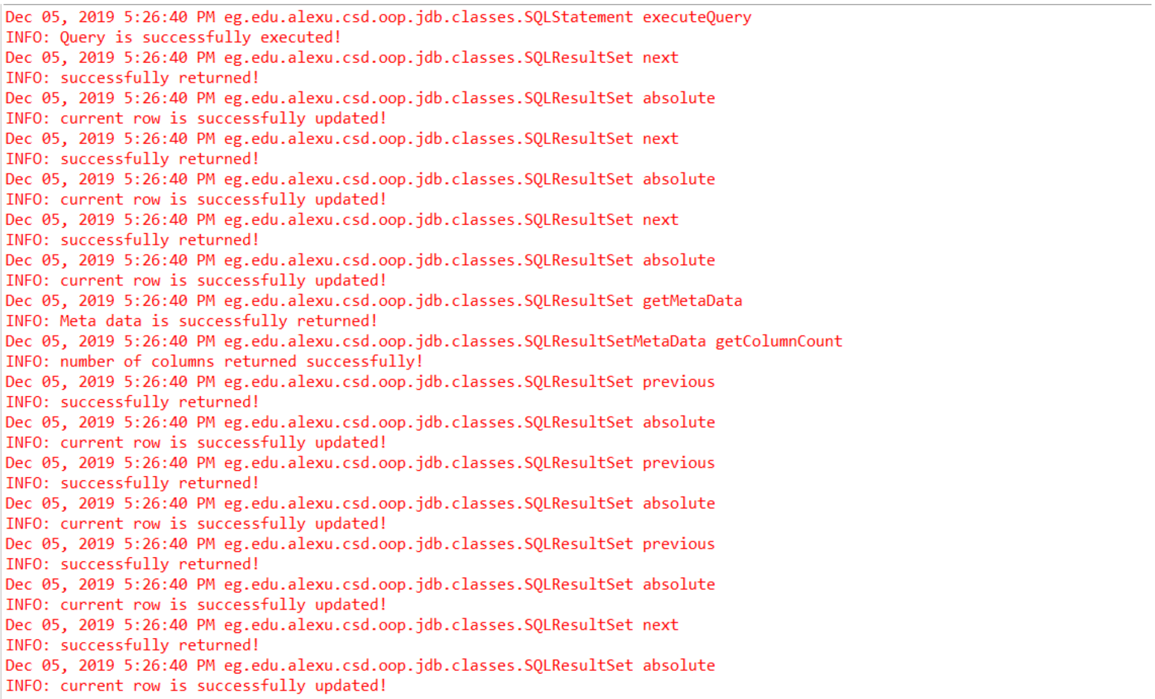
```java
        while(result2.previous());
        result2.next();
        Object column_2_object = result2.getObject(2);
        if (column_2_object instanceof String)
            fail("This should be 'Integer', but found 'String'!");
        else if (column_2_object instanceof Integer) {
            int column_2 = (Integer) column_2_object;
            Assert.assertEquals("Select did't return the updated record!", 10, column_2);
        }
        else
            fail("This should be 'Integer', but what is found can't be identified!");

        statement.close();
    } catch (Throwable e){
        TestRunner.fail("Failed to complete scenario 3", e);
    }

    connection.close();
}
```



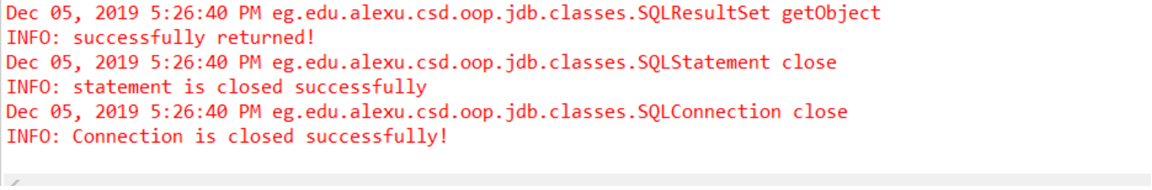
```
Dec 05, 2019 5:26:40 PM eg.edu.alexu.csd.oop.jdb.classes.SQLDriver acceptsURL
INFO: Url accepted
Dec 05, 2019 5:26:40 PM eg.edu.alexu.csd.oop.jdb.classes.SQLDriver connect
INFO: Connection returned successfully
Dec 05, 2019 5:26:40 PM eg.edu.alexu.csd.oop.jdb.classes.SQLConnection createStatement
INFO: Statement created successfully!
Dec 05, 2019 5:26:40 PM eg.edu.alexu.csd.oop.jdb.classes.SQLStatement execute
INFO: Query is successfully executed!
Current Database is : testdb
Dec 05, 2019 5:26:40 PM eg.edu.alexu.csd.oop.jdb.classes.SQLStatement close
INFO: statement is closed successfully
Dec 05, 2019 5:26:40 PM eg.edu.alexu.csd.oop.jdb.classes.SQLConnection createStatement
INFO: Statement created successfully!
Dec 05, 2019 5:26:40 PM eg.edu.alexu.csd.oop.jdb.classes.SQLStatement execute
INFO: Query is successfully executed!
Dec 05, 2019 5:26:40 PM eg.edu.alexu.csd.oop.jdb.classes.SQLStatement executeUpdate
INFO: Query is successfully executed!
Dec 05, 2019 5:26:40 PM eg.edu.alexu.csd.oop.jdb.classes.SQLStatement executeUpdate
INFO: Query is successfully executed!
Dec 05, 2019 5:26:40 PM eg.edu.alexu.csd.oop.jdb.classes.SQLStatement executeUpdate
INFO: Query is successfully executed!
Dec 05, 2019 5:26:40 PM eg.edu.alexu.csd.oop.jdb.classes.SQLStatement executeUpdate
INFO: Query is successfully executed!
column_name1   column_name2   column_name3
value1   5   value3
```



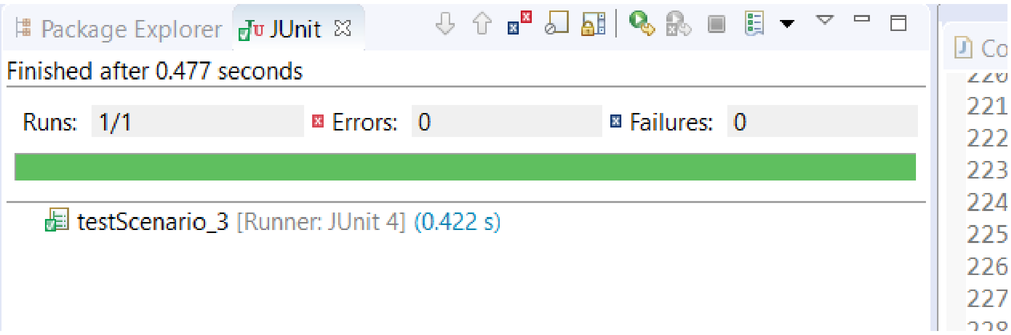
```
Dec 05, 2019 5:26:40 PM eg.edu.alexu.csd.oop.jdb.classes.SQLStatement executeQuery
INFO: Query is successfully executed!
Dec 05, 2019 5:26:40 PM eg.edu.alexu.csd.oop.jdb.classes.SQLResultSet next
INFO: successfully returned!
Dec 05, 2019 5:26:40 PM eg.edu.alexu.csd.oop.jdb.classes.SQLResultSet absolute
INFO: current row is successfully updated!
Dec 05, 2019 5:26:40 PM eg.edu.alexu.csd.oop.jdb.classes.SQLResultSet next
INFO: successfully returned!
Dec 05, 2019 5:26:40 PM eg.edu.alexu.csd.oop.jdb.classes.SQLResultSet absolute
INFO: current row is successfully updated!
Dec 05, 2019 5:26:40 PM eg.edu.alexu.csd.oop.jdb.classes.SQLResultSet getMetaData
INFO: Meta data is successfully returned!
Dec 05, 2019 5:26:40 PM eg.edu.alexu.csd.oop.jdb.classes.SQLResultSetMetaData getColumnCount
INFO: number of columns returned successfully!
Dec 05, 2019 5:26:40 PM eg.edu.alexu.csd.oop.jdb.classes.SQLStatement executeUpdate
INFO: Query is successfully executed!
column_name1   column_name2   column_name3
11111111   10   333333333
value2   6   value4
```



```
Dec 05, 2019 5:26:40 PM eg.edu.alexu.csd.oop.jdb.classes.SQLStatement executeQuery
INFO: Query is successfully executed!
Dec 05, 2019 5:26:40 PM eg.edu.alexu.csd.oop.jdb.classes.SQLResultSet next
INFO: successfully returned!
Dec 05, 2019 5:26:40 PM eg.edu.alexu.csd.oop.jdb.classes.SQLResultSet absolute
INFO: current row is successfully updated!
Dec 05, 2019 5:26:40 PM eg.edu.alexu.csd.oop.jdb.classes.SQLResultSet next
INFO: successfully returned!
Dec 05, 2019 5:26:40 PM eg.edu.alexu.csd.oop.jdb.classes.SQLResultSet absolute
INFO: current row is successfully updated!
Dec 05, 2019 5:26:40 PM eg.edu.alexu.csd.oop.jdb.classes.SQLResultSet next
INFO: successfully returned!
Dec 05, 2019 5:26:40 PM eg.edu.alexu.csd.oop.jdb.classes.SQLResultSet absolute
INFO: current row is successfully updated!
Dec 05, 2019 5:26:40 PM eg.edu.alexu.csd.oop.jdb.classes.SQLResultSet getMetaData
INFO: Meta data is successfully returned!
Dec 05, 2019 5:26:40 PM eg.edu.alexu.csd.oop.jdb.classes.SQLResultSetMetaData getColumnCount
INFO: number of columns returned successfully!
Dec 05, 2019 5:26:40 PM eg.edu.alexu.csd.oop.jdb.classes.SQLResultSet previous
INFO: successfully returned!
Dec 05, 2019 5:26:40 PM eg.edu.alexu.csd.oop.jdb.classes.SQLResultSet absolute
INFO: current row is successfully updated!
Dec 05, 2019 5:26:40 PM eg.edu.alexu.csd.oop.jdb.classes.SQLResultSet previous
INFO: successfully returned!
Dec 05, 2019 5:26:40 PM eg.edu.alexu.csd.oop.jdb.classes.SQLResultSet absolute
INFO: current row is successfully updated!
Dec 05, 2019 5:26:40 PM eg.edu.alexu.csd.oop.jdb.classes.SQLResultSet previous
INFO: successfully returned!
Dec 05, 2019 5:26:40 PM eg.edu.alexu.csd.oop.jdb.classes.SQLResultSet absolute
INFO: current row is successfully updated!
Dec 05, 2019 5:26:40 PM eg.edu.alexu.csd.oop.jdb.classes.SQLResultSet next
INFO: successfully returned!
Dec 05, 2019 5:26:40 PM eg.edu.alexu.csd.oop.jdb.classes.SQLResultSet absolute
INFO: current row is successfully updated!
```



```
Dec 05, 2019 5:26:40 PM eg.edu.alexu.csd.oop.jdb.classes.SQLResultSet getObject
INFO: successfully returned!
Dec 05, 2019 5:26:40 PM eg.edu.alexu.csd.oop.jdb.classes.SQLStatement close
INFO: statement is closed successfully
Dec 05, 2019 5:26:40 PM eg.edu.alexu.csd.oop.jdb.classes.SQLConnection close
INFO: Connection is closed successfully!
```

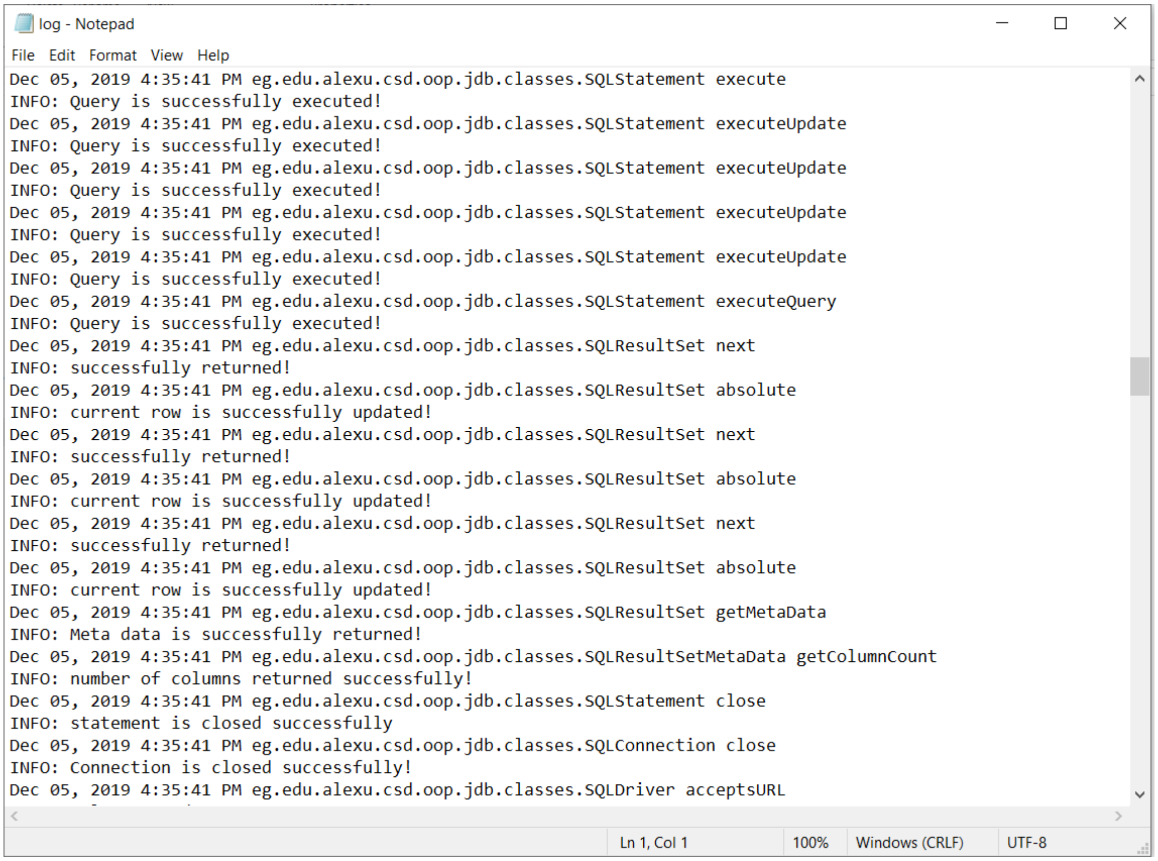


Finished after 0.477 seconds

Runs: 1/1 ☒ Errors: 0 ☒ Failures: 0

testScenario_3 [Runner: JUnit 4] (0.422 s)

All this information also stored in log file



```
Dec 05, 2019 4:35:41 PM eg.edu.alexu.csd.oop.jdb.classes.SQLStatement execute
INFO: Query is successfully executed!
Dec 05, 2019 4:35:41 PM eg.edu.alexu.csd.oop.jdb.classes.SQLStatement executeUpdate
INFO: Query is successfully executed!
Dec 05, 2019 4:35:41 PM eg.edu.alexu.csd.oop.jdb.classes.SQLStatement executeUpdate
INFO: Query is successfully executed!
Dec 05, 2019 4:35:41 PM eg.edu.alexu.csd.oop.jdb.classes.SQLStatement executeUpdate
INFO: Query is successfully executed!
Dec 05, 2019 4:35:41 PM eg.edu.alexu.csd.oop.jdb.classes.SQLStatement executeUpdate
INFO: Query is successfully executed!
Dec 05, 2019 4:35:41 PM eg.edu.alexu.csd.oop.jdb.classes.SQLStatement executeQuery
INFO: Query is successfully executed!
Dec 05, 2019 4:35:41 PM eg.edu.alexu.csd.oop.jdb.classes.SQLResultSet next
INFO: successfully returned!
Dec 05, 2019 4:35:41 PM eg.edu.alexu.csd.oop.jdb.classes.SQLResultSet absolute
INFO: current row is successfully updated!
Dec 05, 2019 4:35:41 PM eg.edu.alexu.csd.oop.jdb.classes.SQLResultSet next
INFO: successfully returned!
Dec 05, 2019 4:35:41 PM eg.edu.alexu.csd.oop.jdb.classes.SQLResultSet absolute
INFO: current row is successfully updated!
Dec 05, 2019 4:35:41 PM eg.edu.alexu.csd.oop.jdb.classes.SQLResultSet next
INFO: successfully returned!
Dec 05, 2019 4:35:41 PM eg.edu.alexu.csd.oop.jdb.classes.SQLResultSet absolute
INFO: current row is successfully updated!
Dec 05, 2019 4:35:41 PM eg.edu.alexu.csd.oop.jdb.classes.SQLResultSet getMetaData
INFO: Meta data is successfully returned!
Dec 05, 2019 4:35:41 PM eg.edu.alexu.csd.oop.jdb.classes.SQLResultSetMetaData getColumnCount
INFO: number of columns returned successfully!
Dec 05, 2019 4:35:41 PM eg.edu.alexu.csd.oop.jdb.classes.SQLStatement close
INFO: statement is closed successfully
Dec 05, 2019 4:35:41 PM eg.edu.alexu.csd.oop.jdb.classes.SQLConnection close
INFO: Connection is closed successfully!
Dec 05, 2019 4:35:41 PM eg.edu.alexu.csd.oop.jdb.classes.SQLDriver acceptsURL
```