Correction Building Database Applications with JDBC - 0

A - false

execute(String) method of Statement can accept all types of queries. It returns true if the first result is a ResultSet object; false if it is an update count or there are no results.

DELETE sql query returns the no. of deleted records, which is not a ResultSet hence record is deleted from the database and false is returned.

Correction Building Database Applications with JDBC - 1

A -

```
Happy New Year!
Happy New Year!
```

In the given query, both the column aliases have same name ‘msg’. ‘rs.getString(“msg”)’ always returns the first matching column.

In case of matching column names, column indexes can be used.

rs.getString(1) would return ‘Happy New Year!’ and rs.getString(2) would return ‘Happy Holidays!’.

Correction Building Database Applications with JDBC - 2

D - 101

Given query returns below records:

```
101 John Smith 12000
102 Sean Smith 15000
103 Regina Williams 15500
104 Natasha George 14600
```

Initially cursor is just before the 1st record.

‘rs.relative(-3);’ doesn’t throw any exception but keeps the cursor just before the 1st record. According to javadoc of relative method, “Attempting to move beyond the first/last row in the result set positions the cursor before/after the first/last row”. Same is true for absolute method as well.

‘rs.relative(1);’ is identical to ‘rs.next()’ so it moves the cursor to the 1st record.

‘rs.getInt(1)’ returns 101.

Correction Building Database Applications with JDBC - 3

C - Program executes successfully but no record is updated in the database

Given sql statement returns below records:

```
102 Sean Smith 15000
103 Regina Williams 15500
```

‘rs.absolute(2);’ moves the cursor pointer to 2nd record.

‘rs.updateDouble(“SALARY”, 20000);’ updates the salary of 2nd record to 20000 but to update the records in the database, ‘rs.updateRow();’ statement must be invoked.

As ‘rs.updateRow()’ statement is missing hence no record is updated in the database.

Please note: there is no need to invoke con.commit(); method as by default Connection object is in auto-commit mode.

Correction Building Database Applications with JDBC - 4

A - An exception is thrown at runtime

Connection object is created inside try-with-resources statement. Both ‘connection’ and ‘con’ refer to the same Connection object.

con.close() method is invoked implicitly just before the closing bracket of try-with-resources statement.

connection.createStatement(); throws exception at runtime as no operations are allowed on closed Connection.

Correction Building Database Applications with JDBC - 5

D - Code executes fine and doesn’t print anything on to the console

Even if there are no records in EMPLOYEE table, ResultSet object returned by ‘stmt.executeQuery(query);’ will never be null.

rs.next() returns false and control doesn’t enter while loop. Code executes fine and doesn’t print anything on to the console.

Correction Building Database Applications with JDBC - 6

B - Records for IDs 1001 and 1002 will be successfully inserted in the database table

According to javadoc of java.sql.Connection, “By default a Connection object is in auto-commit mode, which means that it automatically commits changes after executing each statement. If auto-commit mode has been disabled, the method commit must be called explicitly in order to commit changes; otherwise, database changes will not be saved”.

First 2 executeUpdate(…) method invocation inserts the records in the database table. So records for IDs 1001 and 1002 are available in the database table.

After that ‘con.setAutoCommit(false);’ is invoked which disables the auto-commit mode. The next stmt.executeUpdate(…) statement doesn’t commit the row in the database unless ‘con.commit();’ OR ‘con.setAutoCommit(true);’ is invoked.

Record for ID 1003 is not committed to the database table.

Correction Building Database Applications with JDBC - 7

A - java.sql.Statement

C - java.sql.Connection

D - java.sql.Driver

E - java.sql.ResultSet

DriverManager and Date are classes and hence not correct options.

A JDBC drive must provide implementation for Driver, Statement, PreparedStatement, CallableStatement, ResultSet and Connection interfaces.

Correction Building Database Applications with JDBC - 8

A - ResultSet object

D - Statement object

Statement object is created inside try-with-resources statement. So, close() method is invoked on Statement object implicitly.

According to the javadoc of close() method, “When a Statement object is closed, its current ResultSet object, if one exists, is also closed”. Hence, ResultSet object is also closed.

Connection object is created outside of try-with-resources statement, hence close() method of Connection object is not invoked implicitly.

Correction Building Database Applications with JDBC - 9

C - An exception is thrown at runtime

Method updateRow() must be called after every updated ResultSet row. In this case updateRow() method was called after updating salary in all 4 rows and hence it throws an exception at runtime.

NOTE: To update the salary of all the records successfully, move rs.updateRow() statement inside while loop after rs.updateDouble(…) method call.

Correction Building Database Applications with JDBC - 10

D - An exception is thrown at runtime.

As credentials are passed as java.util.Properties so user name should be passed as “user” property and password should be passed as “password” property.

In the given code, ‘prop.put(“username”, “root”);’ name of property is ‘username’ and not ‘user’ and that is why SQLException is thrown at runtime.

Correction Building Database Applications with JDBC - 11

B - con.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet .CONCUR_READ_ONLY);

D - con.createStatement();

Method createStatement is overloaded: createStatement(), createStatement(int, int) and createStatement(int, int, int).

Correction Building Database Applications with JDBC - 12

D - Regina Gales

Given query returns below records:

```
101 John    Smith   12000
102 Sean    Smith   15000
103 Regina  Williams    15500
104 Natasha George  14600
```

‘rs.absolute(3);’ moves the cursor to the 3rd row.

‘rs.updateString(3, “Gales”);’ updates LASTNAME field of 3rd row from ‘Williams’ to ‘Gales’.

‘rs.updateRow();’ commits the changes to the database.

‘rs.refreshRow();’ It refreshes the current row (which is 3rd now) with its most recent value in the database. There is no change to the cursor position.

System.out.println(rs.getString(2) + ” ” + rs.getString(3)); prints the 2nd and 3rd column values of 3rd row, which are ‘Regina’ and ‘Gales’.

Correction Building Database Applications with JDBC - 13

C - An exception is thrown at runtime

stmt.executeQurey(String) method can accept any String and it returns ResultSet instance, hence no compilation error.

But stmt.executeQuery(String) method cannot issue INSERT, UPDATE and DELETE statements. Hence, ‘stmt.executeQuery(query)’ throws SQLException at runtime.

To issue INSERT, UPDATE or DELETE statements either use stmt.execute(String) method OR stmt.executeUpdate(String) method.

```
try (Connection con = DriverManager.getConnection(url, user,
        password);
     Statement stmt = con.createStatement();)
{
    boolean res = stmt.execute(query);
    System.out.println(stmt.getUpdateCount());
}
```

OR

```
try (Connection con = DriverManager.getConnection(url, user,
        password);
     Statement stmt = con.createStatement();)
{
    System.out.println(stmt.executeUpdate(query));
}
```

Correction Building Database Applications with JDBC - 14

A - Code executes successfully and prints 4 on to the console

Even if there are no records in EMPLOYEE table, ResultSet object returned by ‘stmt.executeQuery(query);’ will never be null.

rs.getMetaData().getColumnCount() returns number of columns of EMPLOYEE table, hence 4 is printed on to the console.

Correction Building Database Applications with JDBC - 15

D - John

As credentials are passed as java.util.Properties so user name should be passed as “user” property and password should be passed as “password” property.

In the given code, correct property names ‘user’ and ‘password’ are used. As URL and DB credentials are correct, hence no issues in connecting the database.

Given sql query returns below records:

```
101 John Smith 12000
102 Sean Smith 15000
103 Regina Williams 15500
104 Natasha George 14600
```

Initially the cursor stays just before the first record.

‘rs.relative(1);’ moves the cursor to the first record.

Column index starts with 1 and not 0, so ‘rs.getString(2)’ returns ‘John’.

Correction Building Database Applications with JDBC - 16

C -

```
13000.0
16000.0
16500.0
15600.0
```

Given query returns below records:

```
101 John Smith 12000
102 Sean Smith 15000
103 Regina Williams 15500
104 Natasha George 14600
```

‘rs.afterLast();’ moves the cursor just after the last record.

‘rs.previous()’ inside while loop moves the cursor from last to first and the codes inside while loop increment the salary of each record by 1000.

‘rs.updateRow();’ makes sure that salary is updated in the database.

Next while loop simply prints the updated salaries on to the console.

Correction Building Database Applications with JDBC - 17

A - 103

As credentials are passed as java.util.Properties so user name should be passed as “user” property and password should be passed as “password” property.

In the given code, correct property names ‘user’ and ‘password’ are used. As URL and DB credentials are correct, hence no issues in connecting the database.

Given query returns below records:

```
101 John Smith 12000
102 Sean Smith 15000
103 Regina Williams 15500
104 Natasha George 14600
```

‘rs.afterLast();’ moves the cursor after 4th record.

‘rs.relative(-1);’ moves the cursor to the 4th record.

‘rs.previous()’ moves the cursor to the 3rd record.

‘rs.getInt(1)’ returns 103.

Correction Building Database Applications with JDBC - 18

D - An exception is thrown at runtime

executeQuery method closes the previously opened ResultSet object on the same Statement object.

ResultSet object referred by res1 gets closed when 2nd executeQuery method is executed. So ‘res1.next()’ throws SQLException at runtime.

Correction Building Database Applications with JDBC - 19

C - jdbc:

database url has the form: protocol:subprotocol:subname.

protocol is always ‘jdbc’.

subprotocol is database specific, for MySQL it is ‘mysql’.

subname contains database details, such as ‘//localhost:3306/ocp’.

Complete database url form MySQL db is: ‘jdbc:mysql://localhost:3306/ocp’. So it always starts with ‘jdbc:’.

Correction Building Database Applications with JDBC - 20

C - TYPE_FORWARD_ONLY

There are 3 ResultSet types: TYPE_FORWARD_ONLY, TYPE_SCROLL_INSENSITIVE and TYPE_SCROLL_SENSITIVE.

Correction Building Database Applications with JDBC - 21

D - An exception is raised by Line n1

Given query returns below records:

```
101 John    Smith   12000
102 Sean    Smith   15000
103 Regina  Williams    15500
104 Natasha George  14600
```

‘rs.moveToInsertRow();’ It moves the cursor to the insert row.

Please note, If the cursor is at insert row and refreshRow() or updateRow() or deleteRow() method is called, then SQLException is thrown.

Hence, in this case an exception is raised by Line n1.

Correction Building Database Applications with JDBC - 22

E - ‘An Error Occurred!’ is printed on to the console

As SELECT statement returns one record, code inside while loop is executed.

‘rs.getInt(“IDD”)’ throws SQLException as column name ‘IDD’ will not be found at runtime. Exception handler for SQLException is available, which prints ‘An Error Occurred!’ on to the console.

Correction Building Database Applications with JDBC - 23

D - 102

Given sql query returns below records:

```
101 John    Smith   12000
102 Sean    Smith   15000
103 Regina  Williams    15500
104 Natasha George  14600
```

‘resultSetType’ can accept 3 constants: TYPE_FORWARD_ONLY, TYPE_SCROLL_INSENSITIVE & TYPE_SCROLL_SENSITIVE.

‘resultSetConcurrency’ can accept 2 constants: CONCUR_READ_ONLY & CONCUR_UPDATABLE.

‘rs.absolute(-2);’ moves the cursor to 3rd record (2nd from last).

‘rs.relative(-1);’ moves the cursor to 2nd record (1 up from the current cursor position).

‘rs.getInt(1)’ returns 102.

NOTE: Column index starts with 1 and not 0.

Correction Building Database Applications with JDBC - 24

D - An exception is thrown at runtime

rs.getMetaData().getColumnCount(); definitely returns 2 as there are 2 columns in the table.

But ResultSet cursor is initially before the first record, hence ‘rs.getString(i)’ throws SQLException at runtime.

To print both the column values correctly, either use rs.absolute(1) OR rs.relative(1) OR rs.next() just before the for loop.

Correction Building Database Applications with JDBC - 25

A - An exception is thrown at runtime

By default ResultSet is not updatable.

‘rs.deleteRow();’ throws an exception at runtime.

To update the ResultSet in any manner (insert, update or delete), the ResultSet must come from a Statement that was created with a ResultSet type of ResultSet.CONCUR_UPDATABLE.

NOTE: If you want to successfully delete the row, then replace ‘con.createStatement();’ with

‘con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_UPDATABLE);’

OR

‘con.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_UPDATABLE);’

Correction Building Database Applications with JDBC - 26

A - Code executes fine and doesn’t print anything on to the console

Even if there are no records in EMPLOYEE table, ResultSet object returned by ‘stmt.executeQuery(query);’ will never be null.

rs.next() returns false and control doesn’t enter while loop.

‘rs.getInt(“IDD”)’ statement has wrong column name but as this statement is not executed, hence SQLException is not thrown at runtime.

Code executes fine and doesn’t print anything on to the console.

Correction Building Database Applications with JDBC - 27

D - CONCUR_READ_ONLY

There are 2 ResultSet concur types: CONCUR_READ_ONLY and CONCUR_UPDATABLE.

Correction Building Database Applications with JDBC - 28

C - An exception is raised by rs.deleteRow();

Given query returns below records:

```
101 John    Smith   12000
102 Sean    Smith   15000
103 Regina  Williams    15500
104 Natasha George  14600
```

‘rs.absolute(1);’ It moves the cursor to 1st record

‘rs.moveToInsertRow();’ It moves the cursor to the insert row.

Please note, If the cursor is at insert row and refreshRow() or updateRow() or deleteRow() method is called, then SQLException is thrown.

Hence, in this case an exception is raised by rs.deleteRow();

Correction Building Database Applications with JDBC - 29

D - An exception is thrown at runtime

Initially cursor is just before the first record. ‘rs.absolute(0);’ also moves the cursor to just before the first record.

As ResultSet cursor is initially before the first record, hence ‘rs.getInt(1)’ throws SQLException at runtime.

Correction Building Database Applications with JDBC - 30

C - An exception is thrown at runtime

As credentials are passed as java.util.Properties so user name should be passed as “user” property and password should be passed as “password” property.

In the given code, correct property names ‘user’ and ‘password’ are used. As URL and DB credentials are correct, hence no issues in connecting the database.

Given query returns just one column containing no. of records, 0 in this case.

But ResultSet cursor is initially before the first record, hence ‘rs.getInt(1)’ throws SQLException at runtime.

Correction Building Database Applications with JDBC - 31

C - An exception is thrown at runtime

It is assumed that JDBC 4.2 driver is configured in the classpath, hence Class.forName(String) is not required. But no harm in using Class.forName(String).

Class.forName(String) expects fully qualified name of the class but in this case url refers to database url and not fully qualified name of the class, hence ClassNotFoundException is thrown at runtime.

Correction Building Database Applications with JDBC - 32

B - Program executes successfully and salary of Regina Williams is updated to 20000

Given sql statement returns below records:

```
102 Sean    Smith   15000
103 Regina  Williams    15500
```

‘rs.absolute(-1);’ moves the cursor pointer to 2nd record (1st record from the bottom).

‘rs.updateDouble(“SALARY”, 20000);’ updates the salary of 2nd record to 20000.

‘rs.updateRow();’ statement updates the record in the database.

Hence, Program executes successfully and salary of Regina Williams is updated to 20000.

Please note: there is no need to invoke con.commit(); method as by default Connection object is in auto-commit mode.

Correction Building Database Applications with JDBC - 33

D - JDBC 4 and later

Starting with JDBC 4, there is no need to manually load the driver class.

For JDBC 3 drivers, java.lang.Class.forName method is used to load the driver class.

Correction Building Database Applications with JDBC - 34

A - Compilation error

Connection is an interface, hence ‘new Connection(url, user, password);’ causes compilation error.

Correct way to create Connection object in this case will be: ‘DriverManager.getConnection(url, user, password);’

Correction Building Database Applications with JDBC - 35

C - An exception is thrown at runtime

By default ResultSet is not updatable.

‘rs.moveToInsertRow();’ throws an exception at runtime.

To update the ResultSet in any manner (insert, update or delete), the ResultSet must come from a Statement that was created with a ResultSet type of ResultSet.CONCUR_UPDATABLE.

NOTE: If you want to successfully insert a new record, then replace ‘con.createStatement();’ with

‘con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_UPDATABLE);’

OR

‘con.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_UPDATABLE);’.

Correction Building Database Applications with JDBC - 36

B - true

According to javadoc of java.sql.Connection, “By default a Connection object is in auto-commit mode, which means that it automatically commits changes after executing each statement. If auto-commit mode has been disabled, the method commit must be called explicitly in order to commit changes; otherwise, database changes will not be saved”.