PROGRAMMING IN JAVA

Assignment 11

TYPE OF QUESTION: MCQ

Number of questions: 10 Total marks: **10 × 1 = 10**

QUESTION 1:

How do you establish a connection to a database using JDBC?

a. **By creating an instance of the Connection interface**

b. **By using the DriverManager.getConnection() method**

c. **By implementing the Connection interface**

d. **By extending the Connection class**

Correct Answer:

b. **By using the DriverManager.getConnection() method**

Detailed Solution:

To establish a connection to a database using JDBC, you use the DriverManager.getConnection() method. This method takes a JDBC URL, username, and password as parameters and returns a Connection object, which represents a connection to the database. The JDBC URL specifies the database type, location, and other connection details.

Load the JDBC Driver:

This is usually done with Class.forName() for some drivers, but many modern drivers load automatically.
Class.forName("com.mysql.cj.jdbc.Driver");

Establish a Connection:

Use DriverManager.getConnection() to establish a connection.
Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydatabase",

QUESTION 2:

Which method executes a simple query and returns a single Result Set object?

a. **executeQuery()**

b. **executeUpdate()**

c. **execute()**

d. **run()**

Correct Answer:

a. **executeQuery()**

Detailed Solution:

The executeQuery() method is used to execute a simple SQL query that returns a single ResultSet object.

Create a Statement:

Use the Connection object to create a Statement or PreparedStatement to execute SQL queries.
Statement stmt = conn.createStatement();

Execute a Query:

Execute the query using executeQuery() for SELECT statements or executeUpdate() for INSERT, UPDATE, and DELETE statements.

ResultSet rs = stmt.executeQuery("SELECT * FROM employees");

Process the ResultSet:

Use ResultSet to iterate through the results.

```
while (rs.next()) {
    System.out.println("Employee Name: " + rs.getString("name"));
}
```

QUESTION 3:

What is the correct order to close database resources?

a. **Connection then Statement then ResultSet**

b. **ResultSet then Statement then Connection**

c. **Statement then Connection then ResultSet**

d. **Statement then ResultSet then Connection**

Correct Answer:

b. **ResultSet then Statement then Connection**

Detailed Solution:

The correct order to close database resources is to first close the ResultSet, then the Statement, and finally the Connection. This ensures that all resources are properly released in the correct sequence.

QUESTION 4:

Which of the following ensures that the correct driver is used to access each data source.

a. **java.sql.Connection**

b. **java.sql.DriverManager**

c. **java.sql.Statement**

d. **java.sql.Driver**

Correct Answer:

b. **java.sql.DriverManager**

Detailed Solution:

The DriverManager class ensures that the correct driver is used to access each data source. It manages a list of database drivers and handles establishing connections to the database.

QUESTION 5:

What is the purpose of the ResultSet interface in JDBC?

a. **To store the result of a query**

b. **To execute SQL queries**

c. **To manage database connections**

d. **To update data in the database**

Correct Answer:

a. **To store the result of a query**

Detailed Solution:

The ResultSet interface in JDBC represents the result set returned by a query. It provides methods to navigate through the result set, retrieve data, and get information about the columns in the result set. The ResultSet interface allows you to process and extract data from a database query.

QUESTION 6:

The following is a statement in Java using JDBC.

```
Connection con = DriverManager.getConnection(
    "jdbc:mysql://localhost:3306/nptel","joy","java"
    );
```

USER! PWD!

Which of the following statement is FALSE?

a. **3306 is the default MySQL port.**

b. **Database name is 'nptel'** NOTE!!!!!!!!!!

c. **The database server is hosted on IP 127.0.0.1**

d. **Password for 'java' user is 'joy'**

Correct Answer:

d. **Password for 'java' user is 'joy'**

Detailed Solution:

In the statement, joy is the username and java is the password. Therefore, statement d, which incorrectly claims that the password for the java user is joy, is false.

QUESTION 7:

Which resources have their close() method called when this code runs?

```
public static void runQuery(Connection conn) throws SQLException {
    try (Statement stmt = conn.createStatement()) {
        ResultSet rs = stmt.executeQuery("select * from clowns");
        rs.next();
    }
}
```

a. No close() methods are called

b. Only Statement

c. Only Statement and Connection

d. Only Statement and ResultSet

Correct Answer:

d. Only Statement and ResultSet

Detailed Solution:

Since this code opens Statement using a try-with-resources, Statement gets closed automatically at the end of the block. Further, closing a Statement automatically closes a ResultSet created by it, making Option D the answer. Remember that you should close any resources you open in the code you write.

QUESTION 8:

Which of the following is used to call stored procedure?

a. **Statement**

b. **PreparedStatement**

c. **CallableStatment**

d. **CalledStatement**

Correct Answer:

c. **CallableStatment**

Detailed Solution:

CallableStatement is used in JDBC to call stored procedure from Java program.

QUESTION 9:

The executeUpdate method can be used with

a. **Statements(Select and Update both)**

b. **Select statement.**

c. **Update/delete/insert operations in the database.**

d. **Only insert operation.**

Correct Answer:

c. **Update/delete/insert operations in the database.**

Detailed Solution:

The executeUpdate method is used to execute SQL statements that change the data in the database, such as UPDATE, DELETE, and INSERT operations.

QUESTION 10:

What does setAutoCommit(false) do?

a. **commits transaction after each query**

b. **explicitly commits transaction**

c. **does not commit transaction automatically after each query**

d. **never commits transaction**

Correct Answer:

c. **does not commit transaction automatically after each query**

Detailed Solution:

setAutoCommit(false) does not commit transaction automatically after each query. That saves a lot of time of the execution and hence improves performance.