Day11\_Answers:

1.Write a Java program to connect to a MySQL database using JDBC.

package hospital;

import java.sql.Connection;

import java.sql.DriverManager;

public class DBConnection {

private static final String ***URL*** = "jdbc:mysql://localhost:3306/hospital\_db";

private static final String ***USER*** = "root";

private static final String ***PASSWORD*** = "Swapna@123";

Class.*forName*("com.mysql.cj.jdbc.Driver");

Connection con=DriverManager.getConnection(URL,USER,PASSWORD);

System.***out***.println("Connected");

}

Output:

Connected

2. Create a Java class to insert student records into a database table.

package jdbc\_con;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.SQLException;

import java.util.Scanner;

public class Prepare\_stmt {

public static void main(String[] args) throws ClassNotFoundException, SQLException {

Scanner sc=new Scanner(System.***in***);

String url="jdbc:mysql://localhost:3306/mydb";

String user="root";

String password="Swapna@123";

Class.*forName*("com.mysql.cj.jdbc.Driver");

Connection con=DriverManager.*getConnection*(url,user,password);

// System.out.println("Connection created");

String qry="insert into students(rollno,name,per) values(?,?,?)";

PreparedStatement pst=con.prepareStatement(qry);

for(int i=1;i<=2;i++) {

System.***out***.println("rollno:");

int roll=sc.nextInt();

System.***out***.println("name:");

String name=sc.next();

System.***out***.println("per:");

int per=sc.nextInt();

pst.setInt(1,roll);

pst.setString(2, name);

pst.setInt(3,per);

System.***out***.println(i+"row inserted");

pst.executeUpdate();

}

pst.close();

con.close();

}

}

Output:

rollno:

5

name:

abc

per:

90

1row inserted

3. Write a JDBC program to fetch and display all student records from the database.

package jdbc\_con;

import java.sql.\*;

public class FetchStudents {

public static void main(String[] args) {

String url = "jdbc:mysql://localhost:3306/mydb";

String user = "root";

String pass = "Swapna@123";

String sql = "SELECT \* FROM student";

try (Connection c = DriverManager.*getConnection*(url, user, pass);

Statement s = c.createStatement();

ResultSet r = s.executeQuery(sql)) {

while (r.next())

System.***out***.println(r.getInt(1) + " " + r.getString(2) );

} catch (SQLException e) {

e.printStackTrace();

}

}

}

4. Develop a program to search a student by ID using JDBC.

package jdbc\_con;

import java.sql.\*;

public class FindStudentById {

public static void main(String[] args) {

String url = "jdbc:mysql://localhost:3306/mydb";

String user = "root";

String pass = "Swapna@123";

String sql = "SELECT \* FROM students WHERE id=102";

try (Connection c = DriverManager.*getConnection*(url, user, pass);

PreparedStatement p = c.prepareStatement(sql)) {

try (ResultSet r = p.executeQuery()) {

if (r.next())

System.***out***.println(r.getInt("id") + " " + r.getString("name") );

else

System.***out***.println("No student found with ID " + args[0]);

}

} catch (Exception e) {

e.printStackTrace();

}

}

}

Output:

102 Reeva sharma

5. Implement an update operation to modify student details in the database using JDBC.

package jdbc\_con;

import java.sql.\*;

public class UpdateStudent {

public static void main(String[] args) {

String url = "jdbc:mysql://localhost:3306/mydb";

String user = "root";

String pass = "Swapna@123";

String sql = "UPDATE students SET name = ?, marks = ? WHERE id = ?";

try (Connection c = DriverManager.*getConnection*(url, user, pass);

PreparedStatement p = c.prepareStatement(sql)) {

p.setString(1, "Person1");

p.setInt(2, 98);

p.setInt(3, 104);

int rows = p.executeUpdate();

System.***out***.println(rows > 0 ? "Updated successfully" : "No matching student found");

} catch (Exception e) {

e.printStackTrace();

}

}

}

Output:

Updated successfully

6. Write a Java program to delete a student record from the database using JDBC.

package jdbc\_con;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

public class DeleteStudent {

public static void main(String[] args) {

String url = "jdbc:mysql://localhost:3306/mydb";

String user = "root";

String pass = "Swapna@123";

String sql = "delete from students where id=104";

try (Connection c = DriverManager.*getConnection*(url, user, pass);

PreparedStatement p = c.prepareStatement(sql)) {

int rows = p.executeUpdate();

System.***out***.println(rows > 0 ? "Deleted successfully" : "No matching student found");

} catch (Exception e) {

e.printStackTrace();

}

}

}

Output:

Deleted successfully

7. Design a Java application to perform all CRUD (Create, Read, Update, Delete) operations on an **Employee** table using JDBC.

package jdbc\_con;

import java.sql.\*;

public class EmployeeCRUD {

public static void main(String[] args) throws SQLException {

String url = "jdbc:mysql://localhost:3306/mydb";

String user = "root";

String pass = "Swapna@123";

Connection conn = DriverManager.*getConnection*(url, user, pass);

Statement stmt = conn.createStatement();

int rows;

rows = stmt.executeUpdate("INSERT INTO employee(id, name) VALUES(105, 'Alice')");

System.***out***.println("Rows inserted = " + rows);

rows = stmt.executeUpdate("UPDATE employee SET name = 'Prathap' WHERE id = 105");

System.***out***.println("Rows updated = " + rows);

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

while (rs.next())

System.***out***.println(rs.getInt("id") + " " + rs.getString("name") );

rows = stmt.executeUpdate("DELETE FROM employee WHERE name = 'Prathap'");

System.***out***.println("Rows deleted = " + rows);

conn.close();

}

}

Output:

Rows inserted = 1

Rows updated = 1

501 Swapna

502 Spandana

503 Jyoshna

504 Swapna

105 Prathap

Rows deleted = 1

8. Create a JDBC-based program to count the total number of rows in a table.

package jdbc\_con;

import java.sql.\*;

public class CountRows {

public static void main(String[] args) throws Exception {

Connection conn = DriverManager.*getConnection*(

"jdbc:mysql://localhost:3306/mydb", "root", "Swapna@123");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("SELECT COUNT(\*) FROM employee");

rs.next();

System.***out***.println(rs.getInt(1));

conn.close();

}

}

Output:

4

9. Develop a program to sort student data in ascending order by name using SQL in JDBC.

package jdbc\_con;

import java.sql.\*;

public class SortStudentsByName {

public static void main(String[] args) throws Exception {

Connection conn = DriverManager.*getConnection*(

"jdbc:mysql://localhost:3306/mydb", "root", "Swapna@123");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("SELECT id, name, marks FROM students ORDER BY name ASC");

while (rs.next())

System.***out***.println(rs.getInt("id") + " " + rs.getString("name") + " " + rs.getInt("marks"));

conn.close();

}

}

Output:

103 mitisha shah 90

101 Neeva sharma 87

102 Reeva sharma 88

10. Write a program to display all students whose percentage is greater than 75 using JDBC and SQL WHERE clause.

package jdbc\_con;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.Statement;

public class StudentsAbove75 {

public static void main(String[] args) throws Exception {

Connection conn = DriverManager.*getConnection*(

"jdbc:mysql://localhost:3306/mydb", "root", "Swapna@123");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("SELECT id, name, marks FROM students where marks>=90");

while (rs.next())

System.***out***.println(rs.getInt("id") + " " + rs.getString("name") + " " + rs.getInt("marks"));

conn.close();

}

}

Output:

103 mitisha shah 90

11. Use **PreparedStatement** to insert multiple student records into the database.

package jdbc\_con;

import java.sql.\*;

public class BatchInsertStudents {

public static void main(String[] args) {

String url = "jdbc:mysql://localhost:3306/mydb";

String user = "root";

String pass = "Swapna@123";

String sql = "INSERT INTO students VALUES (?, ?, ?)";

try (Connection conn = DriverManager.*getConnection*(url, user, pass);

PreparedStatement pstmt = conn.prepareStatement(sql)) {

conn.setAutoCommit(false);

pstmt.setString(2, "Alice");

pstmt.setInt(1, 20);

pstmt.setInt(3, 88);

pstmt.addBatch();

pstmt.setString(2, "Bob");

pstmt.setInt(1, 22);

pstmt.setInt(3, 75);

pstmt.addBatch();

pstmt.setString(2, "Charlie");

pstmt.setInt(1, 21);

pstmt.setInt(3, 90);

pstmt.addBatch();

int[] updateCounts = pstmt.executeBatch();

conn.commit();

System.***out***.println("Inserted " + updateCounts.length + " records.");

} catch (SQLException e) {

e.printStackTrace();

}

}

}

Output:

Inserted 3 records.

12. Implement a program using **transaction management** in JDBC (i.e., commit and rollback).

package jdbc\_con;

import java.sql.\*;

public class TransactionExample {

public static void main(String[] args) {

String url = "jdbc:mysql://localhost:3306/mydb";

String user = "root";

String pass = "Swapna@123";

String insertSQL = "INSERT INTO students VALUES (?, ?, ?)";

String updateSQL = "UPDATE students SET marks = ? WHERE name = ?";

try (Connection conn = DriverManager.*getConnection*(url, user, pass);

PreparedStatement insertStmt = conn.prepareStatement(insertSQL);

PreparedStatement updateStmt = conn.prepareStatement(updateSQL)) {

conn.setAutoCommit(false);

insertStmt.setString(2, "John");

insertStmt.setInt(1, 201);

insertStmt.setInt(3, 85);

insertStmt.executeUpdate();

updateStmt.setInt(1, 90);

updateStmt.setString(2, "Alice");

updateStmt.executeUpdate();

conn.commit();

System.***out***.println("Transaction committed successfully.");

} catch (SQLException e) {

e.printStackTrace();

try (Connection conn = DriverManager.*getConnection*(url, user, pass)) {

conn.rollback();

System.***out***.println("Transaction rolled back.");

} catch (SQLException ex) {

ex.printStackTrace();

}

}

}

}

Output:

Transaction committed successfully.

13. Write a JDBC program to handle exceptions (like invalid ID, connection errors) gracefully.

package jdbc\_con;

import java.sql.\*;

public class FetchStudentsById {

public static void main(String[] args) {

String url = "jdbc:mysql://localhost:3306/mydb";

String user = "root";

String pass = "Swapna@123";

String sql = "SELECT id, name, marks FROM students WHERE id = ?";

try (Connection conn = DriverManager.*getConnection*(url, user, pass);

PreparedStatement pstmt = conn.prepareStatement(sql)) {

int id = 101;

pstmt.setInt(1, id);

try (ResultSet rs = pstmt.executeQuery()) {

if (rs.next()) {

System.***out***.println("ID: " + rs.getInt("id")

+ ", Name: " + rs.getString("name")

+ ", Marks: " + rs.getInt("marks"));

} else {

System.***out***.println("No student found with ID " + id);

}

}

} catch (NumberFormatException e) {

System.***err***.println("Invalid ID format: must be an integer.");

} catch (SQLException e) {

System.***err***.println("Database error:");

System.***err***.println("Message: " + e.getMessage());

System.***err***.println("SQLState: " + e.getSQLState());

System.***err***.println("ErrorCode: " + e.getErrorCode());

}

}

}

Output:

ID: 101, Name: Neeva sharma, Marks: 87

14. Create a login system using JDBC where user credentials are verified from the database.

package jdbc\_con;

import java.sql.\*;

public class LoginConsole {

public static void main(String[] args) {

String url = "jdbc:mysql://localhost:3306/mydb";

String dbUser = "root";

String dbPass = "Swapna@123";

String sql = "SELECT \* FROM users WHERE name = ? AND Location = ?";

try (Connection conn = DriverManager.*getConnection*(url, dbUser, dbPass);

PreparedStatement pstmt = conn.prepareStatement(sql)) {

pstmt.setString(1, "Swapna");

pstmt.setString(2, "kmp");

try (ResultSet rs = pstmt.executeQuery()) {

if (rs.next()) {

System.***out***.println("Login successful. Welcome, " + "Swapna" + "!");

} else {

System.***out***.println("Login failed: invalid username or password.");

}

}

} catch (SQLException e) {

System.***err***.println("Database error:");

System.***err***.println("Message: " + e.getMessage());

System.***err***.println("SQLState: " + e.getSQLState());

System.***err***.println("ErrorCode: " + e.getErrorCode());

}

}

}

Output:

Login successful. Welcome, Swapna!

15. Design the schema for a **Library Management System** and write JDBC programs for:

* Adding a book
* Viewing all books
* Issuing a book to a member
* Returning a book

package jdbc\_con;

import java.sql.\*;

public class LibraryApp {

public static void main(String[] args) throws Exception {

String url = "jdbc:mysql://localhost:3306/mydb", u = "root", p = "Swapna@123";

Connection c = DriverManager.*getConnection*(url, u, p);

c.createStatement().execute("INSERT INTO books (title, author, available) VALUES ('1984','Orwell',TRUE)");

ResultSet rs = c.createStatement().executeQuery("SELECT id, title, author, available FROM books");

while (rs.next()) System.***out***.println(rs.getInt(1) + " " + rs.getString(2) + " " + rs.getString(3) + " " + rs.getBoolean(4));

c.setAutoCommit(false);

int bookId = 1, memberId = 1;

int r1 = c.prepareStatement("UPDATE books SET available = FALSE WHERE id = " + bookId).executeUpdate();

int r2 = c.prepareStatement("INSERT INTO issued\_books (book\_id, member\_id) VALUES (" + bookId + ", " + memberId + ")").executeUpdate();

if (r1 > 0 && r2 > 0) {

c.commit();

System.***out***.println("Issued");

} else {

c.rollback();

System.***out***.println("Issue failed");

}

c.setAutoCommit(false);

int r3 = c.prepareStatement("UPDATE books SET available = TRUE WHERE id = " + bookId).executeUpdate();

int r4 = c.prepareStatement("UPDATE issued\_books SET return\_date = CURRENT\_DATE WHERE book\_id = " + bookId + " AND return\_date IS NULL").executeUpdate();

if (r3 > 0 && r4 > 0) {

c.commit();

System.***out***.println("Returned");

} else {

c.rollback();

System.***out***.println("Return failed");

}

c.close();

}

}

Output:

1 1984 Orwell true

Issued

Returned