

1. Write a JDBC program to update table records using SQL update query

```
import java.sql.*;

public class UpdateTableExample {
    static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
    static final String DB_URL = "jdbc:mysql://localhost/mydatabase";

    static final String USER = "root";
    static final String PASS = "password";

    public static void main(String[] args) {
        Connection conn = null;
        Statement stmt = null;
        try {
            // Register JDBC driver
            Class.forName(JDBC_DRIVER);

            // Open a connection
            conn = DriverManager.getConnection(DB_URL, USER, PASS);

            // Execute a query to update table records
            stmt = conn.createStatement();
            String sql = "UPDATE Employees " +
                "SET age = 30 WHERE id in (100, 101)";
            int rowsUpdated = stmt.executeUpdate(sql);
            System.out.println("Rows updated: " + rowsUpdated);

            // Clean-up environment
            stmt.close();
            conn.close();
        } catch (SQLException se) {
            // Handle errors for JDBC
            se.printStackTrace();
        } catch (Exception e) {
            // Handle errors for Class.forName
            e.printStackTrace();
        } finally {
            // finally block used to close resources
            try {
                if(stmt!=null)
                    stmt.close();
            } catch (SQLException se2) {
```

```

    } // nothing we can do
    try {
        if(conn!=null)
            conn.close();
    } catch(SQLException se) {
        se.printStackTrace();
    } // end finally try
} // end try
}
}

```

2. Write a JDBC program to validate user credentials with the database.

```

import java.sql.*;

public class ValidateUserCredentials {
    static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
    static final String DB_URL = "jdbc:mysql://localhost/mydatabase";

    static final String USER = "root";
    static final String PASS = "password";

    public static void main(String[] args) {
        Connection conn = null;
        PreparedStatement stmt = null;
        try {
            // Register JDBC driver
            Class.forName(JDBC_DRIVER);

            // Open a connection
            conn = DriverManager.getConnection(DB_URL, USER, PASS);

            // Prepare a query to retrieve user credentials
            String sql = "SELECT username FROM Users " +
                "WHERE username=? AND password=?";
            stmt = conn.prepareStatement(sql);

            // Set query parameters
            String username = "john_doe";
            String password = "secret";
            stmt.setString(1, username);
            stmt.setString(2, password);

            // Execute query

```

```

        ResultSet rs = stmt.executeQuery();

        // Check if user credentials are valid
        if (rs.next()) {
            System.out.println("User credentials are valid.");
        } else {
            System.out.println("User credentials are invalid.");
        }

        // Clean-up environment
        rs.close();
        stmt.close();
        conn.close();
    } catch(SQLException se) {
        // Handle errors for JDBC
        se.printStackTrace();
    } catch(Exception e) {
        // Handle errors for Class.forName
        e.printStackTrace();
    } finally {
        // finally block used to close resources
        try {
            if(stmt!=null)
                stmt.close();
        } catch(SQLException se2) {
        } // nothing we can do
        try {
            if(conn!=null)
                conn.close();
        } catch(SQLException se) {
            se.printStackTrace();
        } // end finally try
    } // end try
}
}

```

3. Write a JDBC program to insert a row using Prepared Statement.

```
import java.sql.*;
```

```

public class InsertRowWithPreparedStatement {
    static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
    static final String DB_URL = "jdbc:mysql://localhost/mydatabase";

    static final String USER = "root";

```

```

static final String PASS = "password";

public static void main(String[] args) {
    Connection conn = null;
    PreparedStatement stmt = null;
    try {
        // Register JDBC driver
        Class.forName(JDBC_DRIVER);

        // Open a connection
        conn = DriverManager.getConnection(DB_URL, USER, PASS);

        // Prepare a statement to insert a row into the table
        String sql = "INSERT INTO Employees " +
            "(id, first_name, last_name, age) " +
            "VALUES (?, ?, ?, ?)";
        stmt = conn.prepareStatement(sql);

        // Set the parameter values for the statement
        stmt.setInt(1, 101);
        stmt.setString(2, "John");
        stmt.setString(3, "Doe");
        stmt.setInt(4, 30);

        // Execute the statement
        int rowsInserted = stmt.executeUpdate();
        System.out.println("Rows inserted: " + rowsInserted);

        // Clean-up environment
        stmt.close();
        conn.close();
    } catch (SQLException se) {
        // Handle errors for JDBC
        se.printStackTrace();
    } catch (Exception e) {
        // Handle errors for Class.forName
        e.printStackTrace();
    } finally {
        // finally block used to close resources
        try {
            if(stmt!=null)
                stmt.close();
        } catch (SQLException se2) {
        } // nothing we can do
    }
}

```

```

        try {
            if(conn!=null)
                conn.close();
        } catch(SQLException se) {
            se.printStackTrace();
        } // end finally try
    } // end try
}
}

```

4. Write a JDBC program to retrieve the records from the student table and display records whose marks are greater than 75

```
import java.sql.*;
```

```

public class RetrieveRecords {
    static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
    static final String DB_URL = "jdbc:mysql://localhost/mydatabase";

    static final String USER = "root";
    static final String PASS = "password";

    public static void main(String[] args) {
        Connection conn = null;
        Statement stmt = null;
        try {
            // Register JDBC driver
            Class.forName(JDBC_DRIVER);

            // Open a connection
            conn = DriverManager.getConnection(DB_URL, USER, PASS);

            // Execute a query to retrieve records from the table where marks are greater than 75
            String sql = "SELECT id, name, marks FROM Student WHERE marks > 75";
            stmt = conn.createStatement();
            ResultSet rs = stmt.executeQuery(sql);

            // Iterate through the result set and display the records
            while (rs.next()) {
                // Retrieve by column name
                int id = rs.getInt("id");
                String name = rs.getString("name");
                int marks = rs.getInt("marks");

                // Display the record
            }
        }
    }
}

```

```

        System.out.println("ID: " + id);
        System.out.println("Name: " + name);
        System.out.println("Marks: " + marks);
        System.out.println();
    }

    // Clean-up environment
    rs.close();
    stmt.close();
    conn.close();
} catch(SQLException se) {
    // Handle errors for JDBC
    se.printStackTrace();
} catch(Exception e) {
    // Handle errors for Class.forName
    e.printStackTrace();
} finally {
    // finally block used to close resources
    try {
        if(stmt!=null)
            stmt.close();
    } catch(SQLException se2) {
    } // nothing we can do
    try {
        if(conn!=null)
            conn.close();
    } catch(SQLException se) {
        se.printStackTrace();
    } // end finally try
} // end try
}
}

```

5. Write a JDBC program to retrieve the records from the student table and display records : Use create and select statements.

```

import java.sql.*;

public class RetrieveRecordsWithCreate {
    static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
    static final String DB_URL = "jdbc:mysql://localhost/mydatabase";

    static final String USER = "root";
    static final String PASS = "password";

```

```

public static void main(String[] args) {
    Connection conn = null;
    Statement stmt = null;
    try {
        // Register JDBC driver
        Class.forName(JDBC_DRIVER);

        // Open a connection
        conn = DriverManager.getConnection(DB_URL, USER, PASS);

        // Create a table called "Student" if it doesn't already exist
        stmt = conn.createStatement();
        String createSql = "CREATE TABLE IF NOT EXISTS Student " +
            "(id INT NOT NULL, " +
            " name VARCHAR(255), " +
            " marks INT, " +
            " PRIMARY KEY ( id ))";
        stmt.executeUpdate(createSql);
        System.out.println("Table created successfully");

        // Insert records into the table
        String insertSql = "INSERT INTO Student " +
            "(id, name, marks) " +
            "VALUES (101, 'John Doe', 80), " +
            "(102, 'Jane Smith', 90), " +
            "(103, 'Bob Johnson', 70), " +
            "(104, 'Alice Lee', 85)";
        int rowsInserted = stmt.executeUpdate(insertSql);
        System.out.println(rowsInserted + " rows inserted successfully");

        // Execute a query to retrieve all records from the table
        String selectSql = "SELECT id, name, marks FROM Student";
        ResultSet rs = stmt.executeQuery(selectSql);

        // Iterate through the result set and display the records
        while (rs.next()) {
            // Retrieve by column name
            int id = rs.getInt("id");
            String name = rs.getString("name");
            int marks = rs.getInt("marks");

            // Display the record
            System.out.println("ID: " + id);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

```

        System.out.println("Name: " + name);
        System.out.println("Marks: " + marks);
        System.out.println();
    }

    // Clean-up environment
    rs.close();
    stmt.close();
    conn.close();
} catch(SQLException se) {
    // Handle errors for JDBC
    se.printStackTrace();
} catch(Exception e) {
    // Handle errors for Class.forName
    e.printStackTrace();
} finally {
    // finally block used to close resources
    try {
        if(stmt!=null)
            stmt.close();
    } catch(SQLException se2) {
    } // nothing we can do
    try {
        if(conn!=null)
            conn.close();
    } catch(SQLException se) {
        se.printStackTrace();
    } // end finally try
} // end try
}
}

```

6. Write a JDBC program to retrieve the records from the student table and display records : Use insert, update and delete queries

```

import java.sql.*;

public class CRUDOperations {
    static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
    static final String DB_URL = "jdbc:mysql://localhost/mydatabase";

    static final String USER = "root";
    static final String PASS = "password";

```



```

public static void main(String[] args) {
    Connection conn = null;
    Statement stmt = null;
    try {
        // Register JDBC driver
        Class.forName(JDBC_DRIVER);

        // Open a connection
        conn = DriverManager.getConnection(DB_URL, USER, PASS);

        // Insert a record into the table
        stmt = conn.createStatement();
        String insertSql = "INSERT INTO Student " +
            "(id, name, marks) " +
            "VALUES (105, 'Tom Smith', 78)";
        int rowsInserted = stmt.executeUpdate(insertSql);
        System.out.println(rowsInserted + " rows inserted successfully");

        // Update a record in the table
        String updateSql = "UPDATE Student " +
            "SET marks = 85 " +
            "WHERE id = 102";
        int rowsUpdated = stmt.executeUpdate(updateSql);
        System.out.println(rowsUpdated + " rows updated successfully");

        // Delete a record from the table
        String deleteSql = "DELETE FROM Student " +
            "WHERE id = 103";
        int rowsDeleted = stmt.executeUpdate(deleteSql);
        System.out.println(rowsDeleted + " rows deleted successfully");

        // Execute a query to retrieve all records from the table
        String selectSql = "SELECT id, name, marks FROM Student";
        ResultSet rs = stmt.executeQuery(selectSql);

        // Iterate through the result set and display the records
        while (rs.next()) {
            // Retrieve by column name
            int id = rs.getInt("id");
            String name = rs.getString("name");
            int marks = rs.getInt("marks");

            // Display the record
            System.out.println("ID: " + id);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

```

        System.out.println("Name: " + name);
        System.out.println("Marks: " + marks);
        System.out.println();
    }

    // Clean-up environment
    rs.close();
    stmt.close();
    conn.close();
} catch(SQLException se) {
    // Handle errors for JDBC
    se.printStackTrace();
} catch(Exception e) {
    // Handle errors for Class.forName
    e.printStackTrace();
} finally {
    // finally block used to close resources
    try {
        if(stmt!=null)
            stmt.close();
    } catch(SQLException se2) {
    } // nothing we can do
    try {
        if(conn!=null)
            conn.close();
    } catch(SQLException se) {
        se.printStackTrace();
    } // end finally try
} // end try
}
}

```

7. Write a JDBC program to retrieve the duplicate records from the database.

```

import java.sql.*;

public class DuplicateRecords {
    static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
    static final String DB_URL = "jdbc:mysql://localhost/mydatabase";

    static final String USER = "root";
    static final String PASS = "password";

    public static void main(String[] args) {
        Connection conn = null;
    }
}

```

```

Statement stmt = null;
try {
    // Register JDBC driver
    Class.forName(JDBC_DRIVER);

    // Open a connection
    conn = DriverManager.getConnection(DB_URL, USER, PASS);

    // Execute a query to retrieve duplicate records
    String selectSql = "SELECT name, COUNT(*) FROM Student GROUP BY name HAVING
COUNT(*) > 1";
    ResultSet rs = stmt.executeQuery(selectSql);

    // Iterate through the result set and display the duplicate records
    while (rs.next()) {
        // Retrieve by column name
        String name = rs.getString("name");
        int count = rs.getInt(2);

        // Display the record
        System.out.println("Name: " + name);
        System.out.println("Count: " + count);
        System.out.println();
    }

    // Clean-up environment
    rs.close();
    stmt.close();
    conn.close();
} catch(SQLException se) {
    // Handle errors for JDBC
    se.printStackTrace();
} catch(Exception e) {
    // Handle errors for Class.forName
    e.printStackTrace();
} finally {
    // finally block used to close resources
    try {
        if(stmt!=null)
            stmt.close();
    } catch(SQLException se2) {
    } // nothing we can do
    try {
        if(conn!=null)

```

```

        conn.close();
    } catch(SQLException se) {
        se.printStackTrace();
    } // end finally try
} // end try
}
}

```

8.Create the necessary servlets for the application chosen.Check the authenticity of the login details with the information available in database. If he is a valid user it must redirect to site resources otherwise it should stay in same page with invalid username/password message.

```

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

```

```

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

```

```

public class LoginServlet extends HttpServlet {

```

```

    private static final long serialVersionUID = 1L;

```

```

    public LoginServlet() {
        super();
    }

```

```

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

```

```

        String username = request.getParameter("username");
        String password = request.getParameter("password");

```

```

        Connection conn = null;
        PreparedStatement stmt = null;

```

```

ResultSet rs = null;

try {
    // Register JDBC driver
    Class.forName("com.mysql.jdbc.Driver");

    // Open a connection
    conn = DriverManager.getConnection("jdbc:mysql://localhost/mydatabase", "root",
"password");

    // Check if user exists in the database
    stmt = conn.prepareStatement("SELECT * FROM users WHERE username=? AND
password=?");
    stmt.setString(1, username);
    stmt.setString(2, password);
    rs = stmt.executeQuery();

    if (rs.next()) {
        // If user exists, create a session and redirect to the resources page
        HttpSession session = request.getSession();
        session.setAttribute("username", username);
        response.sendRedirect("resources.jsp");
    } else {
        // If user doesn't exist, display an error message
        PrintWriter out = response.getWriter();
        out.println("<html><body><h2>Invalid username or password</h2></body></html>");
    }
} catch (SQLException se) {
    // Handle errors for JDBC
    se.printStackTrace();
} catch (Exception e) {
    // Handle errors for Class.forName
    e.printStackTrace();
} finally {
    // finally block used to close resources
    try {
        if (rs != null)
            rs.close();
    } catch (SQLException se) {
    } // nothing we can do
    try {
        if (stmt != null)
            stmt.close();
    } catch (SQLException se) {

```

```

        } // nothing we can do
    try {
        if (conn != null)
            conn.close();
    } catch (SQLException se) {
        se.printStackTrace();
    } // end finally try
    } // end try
}
}

```

9. Create the necessary servlets for the application chosen. Check the authenticity of the login details with the information available in database. Insert the details of the registration page into the database. If registration is successful it must display “Registration is successful”.

```

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

public class LoginServlet extends HttpServlet {

    private static final long serialVersionUID = 1L;

    public LoginServlet() {
        super();
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        String action = request.getParameter("action");

        if (action.equals("login")) {

```

```

// Handle login request
String username = request.getParameter("username");
String password = request.getParameter("password");

Connection conn = null;
PreparedStatement stmt = null;
ResultSet rs = null;

try {
    // Register JDBC driver
    Class.forName("com.mysql.jdbc.Driver");

    // Open a connection
    conn = DriverManager.getConnection("jdbc:mysql://localhost/mydatabase", "root",
"password");

    // Check if user exists in the database
    stmt = conn.prepareStatement("SELECT * FROM users WHERE username=? AND
password=?");
    stmt.setString(1, username);
    stmt.setString(2, password);
    rs = stmt.executeQuery();

    if (rs.next()) {
        // If user exists, create a session and redirect to the resources page
        HttpSession session = request.getSession();
        session.setAttribute("username", username);
        response.sendRedirect("resources.jsp");
    } else {
        // If user doesn't exist, display an error message
        PrintWriter out = response.getWriter();
        out.println("<html><body><h2>Invalid username or
password</h2></body></html>");
    }
} catch (SQLException se) {
    // Handle errors for JDBC
    se.printStackTrace();
} catch (Exception e) {
    // Handle errors for Class.forName
    e.printStackTrace();
} finally {
    // finally block used to close resources
    try {
        if (rs != null)

```

```

        rs.close();
    } catch (SQLException se) {
    } // nothing we can do
    try {
        if (stmt != null)
            stmt.close();
    } catch (SQLException se) {
    } // nothing we can do
    try {
        if (conn != null)
            conn.close();
    } catch (SQLException se) {
        se.printStackTrace();
    } // end finally try
    } // end try
} else if (action.equals("register")) {
    // Handle registration request
    String username = request.getParameter("username");
    String password = request.getParameter("password");
    String email = request.getParameter("email");

    Connection conn = null;
    PreparedStatement stmt = null;

    try {
        // Register JDBC driver
        Class.forName("com.mysql.jdbc.Driver");

        // Open a connection
        conn = DriverManager.getConnection("jdbc:mysql://localhost/mydatabase", "root",
"password");

        // Insert user into the database
        stmt = conn.prepareStatement("INSERT INTO users (username, password, email)
VALUES (?, ?, ?)");
        stmt.setString(1, username);
        stmt.setString(2, password);
        stmt.setString(3, email);
        int rows = stmt.executeUpdate();

        if (rows > 0) {
            // If registration successful, display a success message
            PrintWriter out = response.getWriter();
            out.println("<html><body><h2>Registration is successful</h2></body></html>");

```



```
    }  
    } catch (SQLException se
```

10.Create the necessary servlets for the application chosen.Check the authenticity of the login details with the information available in database. Update the password field in the database.

```
import java.io.IOException;  
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.PreparedStatement;  
import java.sql.SQLException;  
  
import javax.servlet.ServletException;  
import javax.servlet.annotation.WebServlet;  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;
```

```
@WebServlet("/updatePassword")  
public class UpdatePasswordServlet extends HttpServlet {  
    private static final long serialVersionUID = 1L;  
  
    // Database credentials  
    static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";  
    static final String DB_URL = "jdbc:mysql://localhost:3306/mydatabase";  
    static final String USER = "root";  
    static final String PASS = "password";  
  
    protected void doPost(HttpServletRequest request, HttpServletResponse response)  
        throws ServletException, IOException {  
        String email = request.getParameter("email");  
        String password = request.getParameter("password");  
  
        // Update password in the database  
        try {  
            Class.forName(JDBC_DRIVER);  
            Connection conn = DriverManager.getConnection(DB_URL, USER, PASS);  
            String sql = "UPDATE users SET password=? WHERE email=?";  
            PreparedStatement pstmt = conn.prepareStatement(sql);  
            pstmt.setString(1, password);
```

```

        pstmt.setString(2, email);
        int rowsAffected = pstmt.executeUpdate();

        if (rowsAffected > 0) {
            response.sendRedirect("updateSuccess.jsp"); // Redirect to success page
        } else {
            response.sendRedirect("updateFailure.jsp"); // Redirect to failure page
        }

        pstmt.close();
        conn.close();
    } catch (ClassNotFoundException | SQLException e) {
        e.printStackTrace();
    }
}
}

```

11.Design the Employee table using database.Write a JDBC program to retrieve the records from the table Use insert, update and delete queries

```

CREATE TABLE Employee (
    id INT PRIMARY KEY,
    name VARCHAR(50) NOT NULL,
    age INT,
    gender VARCHAR(10),
    address VARCHAR(100),
    salary DOUBLE
);

```

```

import java.sql.*;

```

```

public class EmployeeDatabase {
    static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
    static final String DB_URL = "jdbc:mysql://localhost/mydatabase";

    static final String USER = "root";
    static final String PASS = "password";

    public static void main(String[] args) {
        Connection conn = null;
        Statement stmt = null;
        try {
            // Register JDBC driver

```

```

Class.forName(JDBC_DRIVER);

// Open a connection
conn = DriverManager.getConnection(DB_URL, USER, PASS);

// Insert a record
String insertSql = "INSERT INTO Employee (id, name, age, gender, address, salary)
VALUES (1, 'John Doe', 30, 'Male', '123 Main St, Anytown USA', 50000)";
stmt.executeUpdate(insertSql);
System.out.println("Record inserted successfully");

// Update a record
String updateSql = "UPDATE Employee SET salary = 55000 WHERE id = 1";
stmt.executeUpdate(updateSql);
System.out.println("Record updated successfully");

// Delete a record
String deleteSql = "DELETE FROM Employee WHERE id = 1";
stmt.executeUpdate(deleteSql);
System.out.println("Record deleted successfully");

// Clean-up environment
stmt.close();
conn.close();
} catch(SQLException se) {
    // Handle errors for JDBC
    se.printStackTrace();
} catch(Exception e) {
    // Handle errors for Class.forName
    e.printStackTrace();
} finally {
    // finally block used to close resources
    try {
        if(stmt!=null)
            stmt.close();
    } catch(SQLException se2) {
    } // nothing we can do
    try {
        if(conn!=null)
            conn.close();
    } catch(SQLException se) {
        se.printStackTrace();
    } // end finally try
} // end try

```

```
}  
}
```

12. Design the Student table using Database. Write a JDBC program to retrieve the records from the student table and display records : Use insert, update and delete queries

```
CREATE TABLE Employee (  
  id INT PRIMARY KEY,  
  name VARCHAR(50) NOT NULL,  
  age INT,  
  gender VARCHAR(10),  
  address VARCHAR(100),  
  salary DOUBLE  
);
```

```
import java.sql.*;
```

```
public class StudentDatabase {  
  static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";  
  static final String DB_URL = "jdbc:mysql://localhost/mydatabase";  
  
  static final String USER = "root";  
  static final String PASS = "password";  
  
  public static void main(String[] args) {  
    Connection conn = null;  
    Statement stmt = null;  
    try {  
      // Register JDBC driver  
      Class.forName(JDBC_DRIVER);  
  
      // Open a connection  
      conn = DriverManager.getConnection(DB_URL, USER, PASS);  
  
      // Insert a record  
      String insertSql = "INSERT INTO Student (id, name, age, gender, email, phone) VALUES  
(1, 'John Doe', 18, 'Male', 'johndoe@example.com', '555-1234')";  
      stmt.executeUpdate(insertSql);  
  
      // Update a record  
      String updateSql = "UPDATE Student SET age = 19 WHERE id = 1";  
      stmt.executeUpdate(updateSql);  
    }  
  }  
}
```

```

// Delete a record
String deleteSql = "DELETE FROM Student WHERE id = 1";
stmt.executeUpdate(deleteSql);

// Select records
String selectSql = "SELECT * FROM Student";
ResultSet rs = stmt.executeQuery(selectSql);
while (rs.next()) {
    int id = rs.getInt("id");
    String name = rs.getString("name");
    int age = rs.getInt("age");
    String gender = rs.getString("gender");
    String email = rs.getString("email");
    String phone = rs.getString("phone");
    System.out.println("ID: " + id + ", Name: " + name + ", Age: " + age + ", Gender: " +
gender + ", Email: " + email + ", Phone: " + phone);
}

// Clean-up environment
rs.close();
stmt.close();
conn.close();
} catch(SQLException se) {
    // Handle errors for JDBC
    se.printStackTrace();
} catch(Exception e) {
    // Handle errors for Class.forName
    e.printStackTrace();
} finally {
    // finally block used to close resources
    try {
        if(stmt!=null)
            stmt.close();
    } catch(SQLException se2) {
    } // nothing we can do
    try {
        if(conn!=null)
            conn.close();
    } catch(SQLException se) {
        se.printStackTrace();
    } // end finally try
} // end try
}
}

```

