Task 3: PreparedStatement

Modify the SELECT query program to use PreparedStatement to parameterize the query and prevent SQL injection.

```
package test;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.sql.*;

public class UserAuthenticationPreparedStatement {

  public static void main(String[] args) {
    String url = "jdbc:sqlite:users.db";
    Connection conn = null;

    try {
      conn = DriverManager.getConnection(url);
      System.out.println("Connected to the SQLite database.");

      createTable(conn);
      String userId="john_doe";
      String password="password123";
      insertUser(conn, userId, password);
      String inputUserId="john_doe";
      String inputPassword="password123";
      boolean isAuthenticated=authenticateUser(conn, inputUserId, inputPassword);
      if (isAuthenticated) {
        System.out.println("User authentication successful.");
      } else {
        System.out.println("Invalid credentials. Access denied.");
      }
```

```java
    }catch(SQLException e) {
      System.out.println("SQL Exception: " + e.getMessage());
    } catch (NoSuchAlgorithmException e) {
      System.out.println("Error creating password hash: " + e.getMessage());
    } finally {
      try {
        if (conn!=null) {
          conn.close();
        }
      } catch (SQLException ex) {
        System.out.println("Error closing connection: " + ex.getMessage());
      }
    }
  }
  private static void createTable(Connection conn) throws SQLException {
    String sql = "CREATE TABLE IF NOT EXISTS User (" +
          "UserID TEXT PRIMARY KEY," +
          "PasswordHash TEXT)";
    try (Statement stmt = conn.createStatement()) {
      stmt.execute(sql);
    }
  }
  private static void insertUser(Connection conn, String userId, String password) throws
SQLException, NoSuchAlgorithmException {
    String hashedPassword = hashPassword(password);
    String sql = "INSERT INTO User(UserID, PasswordHash) VALUES(?, ?)";
    try (PreparedStatement pstmt = conn.prepareStatement(sql)) {
      pstmt.setString(1, userId);
      pstmt.setString(2, hashedPassword);
      pstmt.executeUpdate();
```

```
        }
        System.out.println("User inserted into the table.");
    }

    private static boolean authenticateUser(Connection conn, String userId, String password) throws SQLException, NoSuchAlgorithmException {
        String hashedPassword = hashPassword(password);

        String sql = "SELECT UserID FROM User WHERE UserID = ? AND PasswordHash = ?";
        try (PreparedStatement pstmt = conn.prepareStatement(sql)) {
            pstmt.setString(1, userId);
            pstmt.setString(2, hashedPassword);
            ResultSet rs = pstmt.executeQuery();
            return rs.next();
        }
    }

    private static String hashPassword(String password) throws NoSuchAlgorithmException {
        MessageDigest digest = MessageDigest.getInstance("SHA-256");
        byte[] hash = digest.digest(password.getBytes());
        StringBuilder hexString = new StringBuilder();
        for (byte b : hash) {
            String hex = Integer.toHexString(0xff & b);
            if (hex.length() == 1) hexString.append('0');
            hexString.append(hex);
        }
        return hexString.toString();
    }
```

```
}
```