

Task 1: Establishing Database Connections

Write a Java program that connects to a SQLite database and prints out the connection object to confirm successful connection.

Ans:

```
package Practice2;
```

```
import java.sql.Connection;
```

```
import java.sql.DriverManager;
```

```
import java.sql.SQLException;
```

```
public class MySQLConnection {
```

```
    public static void main(String[] args) {
```

```
        String url = "jdbc:mysql://localhost:3306/wipdb";
```

```
        String user = "root";
```

```
        String password = "root";
```

```
        try (Connection conn = DriverManager.getConnection(url, user, password)) {
```

```
            if (conn != null) {
```

```
                System.out.println("Connection to MySQL has been established.");
```

```
                System.out.println("Connection Object: " + conn);
```

```
            } else {
```

```
                System.out.println("Failed to make connection!");
```

```
            }
```

```
        } catch (SQLException e) {
```

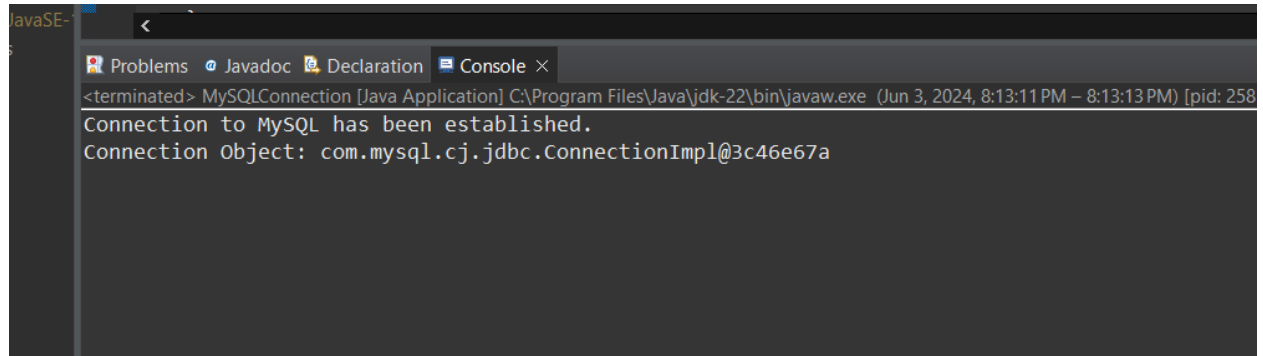
```
            System.out.println(e.getMessage());
```

```
        }
```

```
    }
```

```
}
```

OutPut:

A screenshot of an IDE's console window. The window has a dark background and a light-colored title bar. The title bar contains the text "JavaSE-" followed by a back arrow icon. Below the title bar, there are four tabs: "Problems", "Javadoc", "Declaration", and "Console". The "Console" tab is selected and shows the following output: "<terminated> MySQLConnection [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (Jun 3, 2024, 8:13:11 PM – 8:13:13 PM) [pid: 258] Connection to MySQL has been established. Connection Object: com.mysql.cj.jdbc.ConnectionImpl@3c46e67a".

```
<terminated> MySQLConnection [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (Jun 3, 2024, 8:13:11 PM – 8:13:13 PM) [pid: 258]
Connection to MySQL has been established.
Connection Object: com.mysql.cj.jdbc.ConnectionImpl@3c46e67a
```

Task 2: SQL Queries using JDBC

Create a table 'User' with a following schema 'User ID' and 'Password' stored as hash format (note you have research on how to generate hash from a string), accept "User ID" and "Password" as input and check in the table if they match to confirm whether user access is allowed or not.

```
package Practice2;
```

```
import java.security.MessageDigest;
```

```
import java.security.NoSuchAlgorithmException;
```

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

```
import java.util.Scanner;
```

```
public class UserAuthentication {
```

```
    private static final String DB_URL = "jdbc:mysql://localhost:3306/new";
```

```

private static final String DB_USER = "root";

private static final String DB_PASSWORD = "root";


public static void main(String[] args) {
    try (Connection connection = DriverManager.getConnection(DB_URL, DB_USER,
DB_PASSWORD)) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter 1 to register, 2 to login:");

        int choice = scanner.nextInt();

        scanner.nextLine();

        if (choice == 1) {
            registerUser(connection, scanner);
        } else if (choice == 2) {
            loginUser(connection, scanner);
        } else {
            System.out.println("Invalid choice.");
        }

    } catch (SQLException e) {
        e.printStackTrace();
    }
}


private static void registerUser(Connection connection, Scanner scanner) throws
SQLException {
    System.out.print("Enter User ID: ");

```

```

String userId = scanner.nextLine();

System.out.print("Enter Password: ");

String password = scanner.nextLine();


String hashedPassword = hashPassword(password);


String sql = "INSERT INTO User (UserID, PasswordHash) VALUES (?, ?)";
try (PreparedStatement statement = connection.prepareStatement(sql)) {
    statement.setString(1, userId);
    statement.setString(2, hashedPassword);
    statement.executeUpdate();
    System.out.println("User registered successfully.");
}
}

private static void loginUser(Connection connection, Scanner scanner) throws SQLException {
    System.out.print("Enter User ID: ");
    String userId = scanner.nextLine();
    System.out.print("Enter Password: ");
    String password = scanner.nextLine();

    String hashedPassword = hashPassword(password);

    String sql = "SELECT * FROM User WHERE UserID = ? AND PasswordHash = ?";
    try (PreparedStatement statement = connection.prepareStatement(sql)) {
        statement.setString(1, userId);
        statement.setString(2, hashedPassword);
    }
}

```

```

ResultSet resultSet = statement.executeQuery();

if (resultSet.next()) {
    System.out.println("Login successful.");
} else {
    System.out.println("Invalid User ID or Password.");
}
}
}

private static String hashPassword(String password) {
    try {
        MessageDigest md = MessageDigest.getInstance("SHA-256");
        byte[] hashBytes = md.digest(password.getBytes());
        StringBuilder sb = new StringBuilder();
        for (byte b : hashBytes) {
            sb.append(String.format("%02x", b));
        }
        return sb.toString();
    } catch (NoSuchAlgorithmException e) {
        throw new RuntimeException("SHA-256 algorithm not found.", e);
    }
}
}

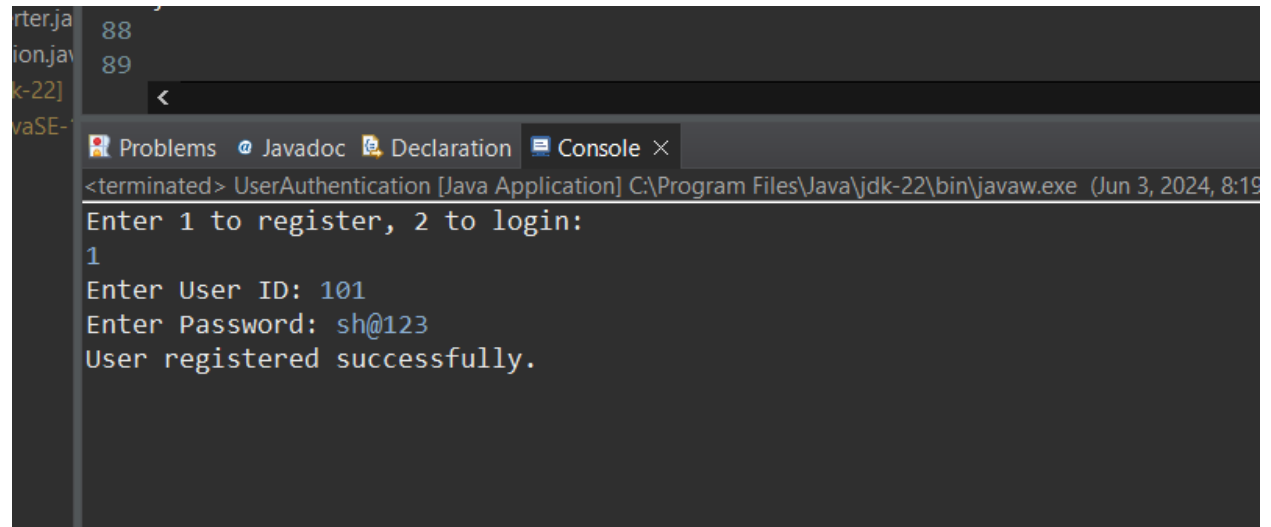
```

Sql-query:

CREATE TABLE User (

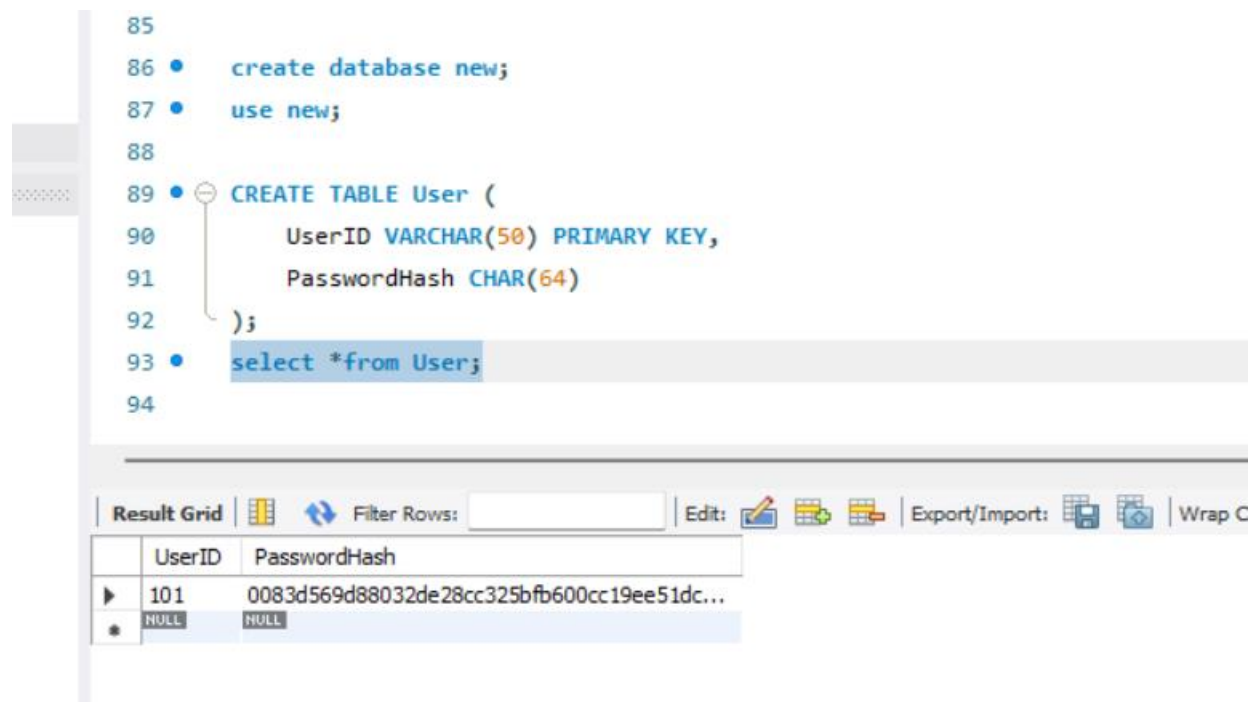
```
UserID VARCHAR(50) PRIMARY KEY,  
PasswordHash CHAR(64)  
);
```

Output:



The screenshot shows a Java IDE with a console window. The console output is as follows:

```
<terminated> UserAuthentication [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (Jun 3, 2024, 8:19  
Enter 1 to register, 2 to login:  
1  
Enter User ID: 101  
Enter Password: sh@123  
User registered successfully.
```



The screenshot shows a SQL IDE with a code editor and a result grid. The code in the editor is:

```
85  
86 • create database new;  
87 • use new;  
88  
89 • CREATE TABLE User (  
90     UserID VARCHAR(50) PRIMARY KEY,  
91     PasswordHash CHAR(64)  
92 );  
93 • select *from User;  
94
```

The result grid below the code shows the following data:

	UserID	PasswordHash
▶	101	0083d569d88032de28cc325bfb600cc19ee51dc...
*	NULL	NULL

Task 3: PreparedStatement

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

Ans:

```
package Practice2;
```

```
import java.security.MessageDigest;
```

```
import java.security.NoSuchAlgorithmException;
```

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

```
import java.util.Scanner;
```

```
public class z {
```

```
    private static final String DB_URL = "jdbc:mysql://localhost:3306/new";
```

```
    private static final String DB_USER = "root";
```

```
    private static final String DB_PASSWORD = "root";
```

```
    public static void main(String[] args) {
```

```
        try (Connection connection = DriverManager.getConnection(DB_URL,  
DB_USER, DB_PASSWORD)) {
```

```
            Scanner scanner = new Scanner(System.in);
```

```
            System.out.println("Enter 1 to register, 2 to login:");
```

```

int choice = scanner.nextInt();
scanner.nextLine();

if (choice == 1) {
    registerUser(connection, scanner);
} else if (choice == 2) {
    loginUser(connection, scanner);
} else {
    System.out.println("Invalid choice.");
}

} catch (SQLException e) {
    e.printStackTrace();
}
}

private static void registerUser(Connection connection, Scanner scanner)
throws SQLException {
    System.out.print("Enter User ID: ");
    String userId = scanner.nextLine();
    System.out.print("Enter Password: ");
    String password = scanner.nextLine();

    String hashedPassword = hashPassword(password);

```



```

String sql = "INSERT INTO User (UserID, PasswordHash) VALUES (?, ?)";
try (PreparedStatement statement = connection.prepareStatement(sql)) {
    statement.setString(1, userId);
    statement.setString(2, hashedPassword);
    statement.executeUpdate();
    System.out.println("User registered successfully.");
}
}

```

```

private static void loginUser(Connection connection, Scanner scanner) throws
SQLException {

```

```

    System.out.print("Enter User ID: ");
    String userId = scanner.nextLine();
    System.out.print("Enter Password: ");
    String password = scanner.nextLine();

```

```

    String hashedPassword = hashPassword(password);

```

```

String sql = "SELECT * FROM User WHERE UserID = ? AND PasswordHash = ?";
try (PreparedStatement statement = connection.prepareStatement(sql)) {
    statement.setString(1, userId);
    statement.setString(2, hashedPassword);
    ResultSet resultSet = statement.executeQuery();

    if (resultSet.next()) {

```

```
        System.out.println("Login successful.");
    } else {
        System.out.println("Invalid User ID or Password.");
    }
}
}
```

```
private static String hashPassword(String password) {
    try {
        MessageDigest md = MessageDigest.getInstance("SHA-256");
        byte[] hashBytes = md.digest(password.getBytes());
        StringBuilder sb = new StringBuilder();
        for (byte b : hashBytes) {
            sb.append(String.format("%02x", b));
        }
        return sb.toString();
    } catch (NoSuchAlgorithmException e) {
        throw new RuntimeException("SHA-256 algorithm not found.", e);
    }
}
}
```

```
81         }
82         return sb.toString();
83     } catch (NoSuchAlgorithmException e) {
        <
    }
}

-22]
aSE-
Problems Javadoc Declaration Console ×
<terminated> z [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (Jun 3, 2024, 8:35:33 PM – 8:
Enter 1 to register, 2 to login:
2
Enter User ID: 101
Enter Password: sh@123
Login successful.
```