

# Day 13 Assignment

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

```
import java.sql.*;

import java.sql.DriverManager;
import java.sql.SQLException;

/*
 * Write a Java program that connects to a SQLite database and
 * prints out the connection object to confirm successful connection.
 */
public class Assignment_1 {

    private static final String url = "jdbc:mysql://localhost:3306/wiproddb";
    private static final String user = "root";
    private static final String password = "Sayan$1999";

    private static Connection con;

    public static Connection createConnection() {

        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            con = DriverManager.getConnection(url, user, password);
        } catch (SQLException e) {
            throw new RuntimeException(e);
        } catch (ClassNotFoundException e) {
            throw new RuntimeException(e);
        }
        return con;
    }

    public static void main(String[] args) {
        System.out.println(createConnection());
    }
}
```

**Output**

```
C:\Users\coolr\.jdk\openjdk-22.0.1\bin\java.exe "-javaagent:C:\Pr
com.mysql.cj.jdbc.ConnectionImpl@24aed80c
```

```
Process finished with exit code 0
```

- 
2. 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 m5_core_java_programming.day_13;

/*
    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.
*/

import java.sql.*;
import java.util.Scanner;

public class Assignment_2 {

    public static String createHash(String password) {
        return Integer.toString(password.hashCode());
    }

    public static boolean checkValidation(int userid, String password,
    Connection con) {
        password = createHash(password);
        try {
            String sqlStat = "SELECT * FROM User WHERE UserID = ? AND Password =
?";

            PreparedStatement preparedStatement = con.prepareStatement(sqlStat);
```

```

        preparedStatement.setInt(1, userid);
        preparedStatement.setString(2, password);
        ResultSet resultSet = preparedStatement.executeQuery();
        return resultSet.next();
    } catch (SQLException e) {
        throw new RuntimeException(e);
    }
}

public static void main(String[] args) {
    Scanner scan = new Scanner(System.in);
    try {
        Connection con = Assignment_1.createConnection();
        String sqlStat = "CREATE TABLE User (" +
            "    UserID INT PRIMARY KEY," +
            "    Password VARCHAR(50)" +
            ");";

        Statement statement = con.createStatement();
        statement.executeUpdate(sqlStat);
        System.out.println("User Table Successfully created");
        sqlStat = "INSERT INTO USER (UserID, Password) VALUES (?,?)";
        System.out.println("Enter User ID : ");
        int userid = scan.nextInt();
        System.out.println("Enter User Password :");
        String password = scan.next();
        password = createHash(password);
        PreparedStatement preparedStatement = con.prepareStatement(sqlStat);
        preparedStatement.setInt(1, userid);
        preparedStatement.setString(2, password);
        preparedStatement.executeUpdate();
        System.out.println("User " + userid + " is successfully inserted");
        System.out.println("For Validation ");
        System.out.println("Enter user id : ");
        userid = scan.nextInt();
        System.out.println("Enter password : ");
        password = scan.next();

        if (checkValidation(userid, password, con)) {
            System.out.println("User Allowed");
        } else {
            System.out.println("User Not Allowed");
        }
    }
}

```

```

        con.close();
    } catch (Exception e) {
        System.out.println(e);
    }
}
}

```

## Output

```

C:\Users\c00tr\Idea\openjdk-22.0.1\bin\java.exe -javaagent:C:\Program Files\JetBrains\
User Table Successfully created
Enter User ID :
127
Enter User Password :
Sayan$1999
User 127 is successfully inserted
For Validation
Enter user id :
127
Enter password :
Sayan$1999
User Allowed

Process finished with exit code 0

```

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

```

package m5_core_java_programming.day_13;

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

import java.sql.Connection;
import
java.sql.PreparedStatement;
import java.sql.SQLException;

```

```
import java.util.Scanner;

public class Assignment_3 {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        String sqlStat = "INSERT INTO USER (UserID, Password) VALUES(?,?)";
        System.out.println("Enter User ID : ");
        int userid = scan.nextInt();
        System.out.println("Enter User Password :");
        String password = scan.next();
        password = Assignment_2.createHash(password);
        try {
            Connection con = Assignment_1.createConnection();
            PreparedStatement preparedStatement = con.prepareStatement(sqlStat);
            preparedStatement.setInt(1, userid);
            preparedStatement.setString(2, password);
            preparedStatement.executeUpdate();
            System.out.println("User " + userid + " is successfully inserted");
        } catch (SQLException e) {
            throw new RuntimeException(e);
        }
    }
}
```

**Output**

```
C:\Users\coolr\.jdk\openjdk-22.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBr
Enter User ID :
238
Enter User Password :
password194
User 238 is successfully inserted

Process finished with exit code 0
```

	UserID	Password
▶	127	1143991092
	238	1403730577
▲	NULL	NULL

---

#### Tools Used :

IntelliJ IDE java version

"1.8.0\_411"

Java(TM) SE Runtime Environment (build 1.8.0\_411-b09)

Java HotSpot(TM) Client VM (build 25.411-b09, mixed mode, sharing)

MySQL Workbench 8.0 CE