SQL Injection

SQL Injection

- A program takes user input and creates an SQL query using string concatenation.
- For example, suppose the user is prompted for user name and password.
- And then creates a query using as follows

SQL Injection

```
String queryString = "SELECT NAME, PASSWORD FROM USERS
        WHERE NAME = ' "+ name + " ' AND PASSWORD = '
                " + password + " ' "  ;
```

SQL Injection Tricks – SQL comment

- SELECT * FROM User WHERE UserName = 'john' - - ' AND Password = ' '
- -- SQL comment.
- The user can enter
  - john' - -

SQL Injection Tricks - Using OR

- WHERE Username = 'john' OR false AND password = ''
- AND has higher precedence than OR.
- Extracts all johns from the database.
- Input
  - john' OR 'a'='b
- Becomes
  - Username = 'john' OR 'a'='b' AND ....

SQL Injection Tricks - Numbers

- custId should be a number but an untyped scripting language doesn’t check it
- WHERE CustId = custId
- WHERE CustId = custId ; DELETE * FROM CUSTOMER
- The ; terminates the first statement.
- Any SQL could follow the ;

Solutions

- Never construct SQL statements using String concatenation.
- Use PreparedStatements or the equivalent.
- If not you would need to check user input values (possibly using regular expressions), and checking for all possible meta-characters.

Solution - Java

- Java – Use PreparedStatements.

```
String queryString = "SELECT NAME, PASSWORD FROM USERS
                    WHERE NAME = ? AND PASSWORD = ?";

PreparedStatement preparedStatement = con.prepareStatement(queryString);
preparedStatement.setString(1, name) ;
preparedStatement.setString(2, password) ;
```