~~~~~~~~~~~~~~~ Vulnerability 1 ~~~~~~~~~~~~~~~
ID - CWE-500
Module - sql.java (Line 8)
Name - Public Static Field Not Marked Final
Description - An object contains a public static field that is not marked final, which might allow it to be modified in unexpected ways.
Severity - Low
Resources - https://cwe.mitre.org/data/definitions/500.html
Remediation Advice - Clearly identify the scope for all critical data elements, including whether they should be regarded as static. Make any static fields private and constant.
Days To Remediate - 120

~~~~~~~~~~~~~~~ Vulnerability 2 ~~~~~~~~~~~~~~~
ID - CWE-798
Module - sql.java (Line 11)
Name - Use of Hard-coded Credentials
Description - The product contains hard-coded credentials, such as a password or cryptographic key, which it uses for its own inbound authentication, outbound communication to external components, or encryption of internal data.
Severity - Low
Resources - https://cwe.mitre.org/data/definitions/798.html
Remediation Advice - Credentials should be hashed and stored safely in a password-protected external file
Days To Remediate - 120

~~~~~~~~~~~~~~~ Vulnerability 3 ~~~~~~~~~~~~~~~
ID - CWE-259
Module - sql.java (Line 15)
Name - Use of Hard-coded Password
Description - The product contains a hard-coded password, which it uses for its own inbound authentication or for outbound communication to external components.
Severity - Low
Resources - https://cwe.mitre.org/data/definitions/259.html
Remediation Advice - Passwords should be hashed and stored safely in a password-protected external file
Days To Remediate - 120

~~~~~~~~~~~~~~~ Vulnerability 4 ~~~~~~~~~~~~~~~
ID - CWE-766
Module - sql.java (Line 15)
Name - Critical Data Element Declared Public
Description - The product declares a critical variable, field, or member to be public when intended security policy requires it to be private.
Severity - Low
Resources - https://cwe.mitre.org/data/definitions/766.html
Remediation Advice - Data should be private, static, and final whenever possible. This will assure that your code is protected by instantiating early, preventing access, and preventing tampering.
Days To Remediate - 120

~~~~~~~~~~~~~~~ Vulnerability 5 ~~~~~~~~~~~~~~~

ID - CWE-89

Module - sql.java (Line 32)

Name - Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')

Description - The product constructs all or part of an SQL command using externally-influenced input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could modify the intended SQL command when it is sent to a downstream component

Severity - Critical

Resources - https://cwe.mitre.org/data/definitions/89.html

Remediation Advice - Prepared statements, client and server side input validation, safe stored procedures, or escaping user input can be used to mitigate against SQL injection attacks.

Days To Remediate - 15

~~~~~~~~~~~~~~~ Vulnerability 6 ~~~~~~~~~~~~~~~

ID - CWE-209

Module - sql.java (Line 42)

Name - Generation of Error Message Containing Sensitive Information

Description - The product generates an error message that includes sensitive information about its environment, users, or associated data.

Severity - Low

Resources - https://cwe.mitre.org/data/definitions/209.html

Remediation Advice - When an exception is caught, only print insensitive and desired data to a user.

Days To Remediate - 120