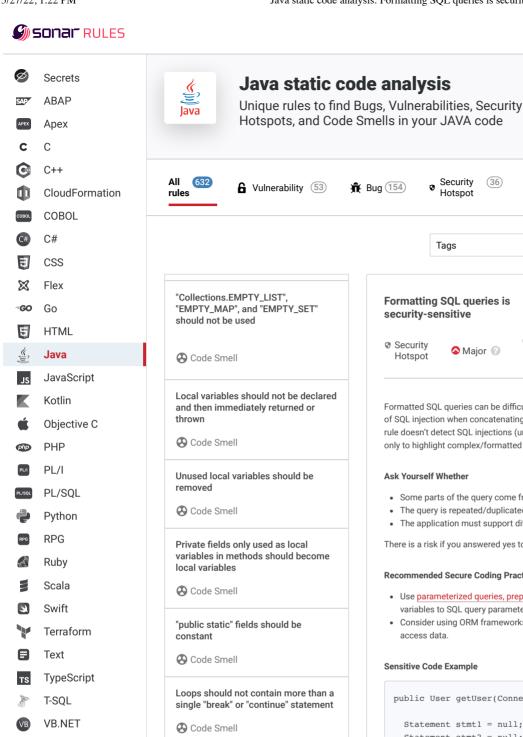
Products ✓

O Quick 42 Fix

Analyze your code



Declarations should use Java

classes such as "LinkedList"

Code Smell

Code Smell

of a "for" loop Code Smell

not be used

Code Smell

least 3 "case" clauses

collection interfaces such as "List"

rather than specific implementation

"switch" statements should have at

A "while" loop should be used instead

The default unnamed package should

"aguala(Ohioot ahi)" ahauld ha

Formatting SQL queries is • cwe spring owasp sans-top25 Major 🕝 bad-practice cert hibernate sql Formatted SQL queries can be difficult to maintain, debug and can increase the risk of SQL injection when concatenating untrusted values into the query. However, this rule doesn't detect SQL injections (unlike rule {rule:javasecurity:S3649}), the goal is only to highlight complex/formatted queries. Some parts of the query come from untrusted values (like user inputs). The guery is repeated/duplicated in other parts of the code. The application must support different types of relational databases. There is a risk if you answered yes to any of those questions. **Recommended Secure Coding Practices** • Use parameterized queries, prepared statements, or stored procedures and bind variables to SQL query parameters. Consider using ORM frameworks if there is a need to have an abstract layer to public User getUser(Connection con, String user) throws SQLE Statement stmt1 = null; Statement stmt2 = null: PreparedStatement pstmt; try { stmt1 = con.createStatement(); ResultSet rs1 = stmt1.executeQuery("GETDATE()"); // No i stmt2 = con.createStatement(); ResultSet rs2 = stmt2.executeQuery("select FNAME, LNAME,

⊗ Code

Smell

(389)

Search by name.

"from USERS where UNAME=" + user); // Sens

"from USERS where UNAME=" + user); // Sens

pstmt = con.prepareStatement("select FNAME, LNAME, SSN "

public User getUserHibernate(org.hibernate.Session session,

"FROM students where fname = " + data);

org.hibernate.Query query = session.createQuery(

ResultSet rs3 = pstmt.executeQuery();

//...

}

}

VB<sub>6</sub>

**XML** 

// Sens

equals(Object Obj) should be overridden along with the "compareTo(T obj)" method

Code Smell

Package names should comply with a naming convention

Code Smell

Nested code blocks should not be used

Code Smell

Array designators "[]" should be on the type, not the variable

Code Smell

## Compliant Solution

```
public User getUser(Connection con, String user) throws SQLE
  Statement stmt1 = null;
  PreparedStatement pstmt = null;
  String query = "select FNAME, LNAME, SSN " +
                 "from USERS where UNAME=?"
  try {
    stmt1 = con.createStatement();
    ResultSet rs1 = stmt1.executeQuery("GETDATE()");
   pstmt = con.prepareStatement(query);
    pstmt.setString(1, user); // Good; PreparedStatements e
    ResultSet rs2 = pstmt.executeQuery();
    //...
 }
public User getUserHibernate(org.hibernate.Session session,
  org.hibernate.Query query = session.createQuery("FROM stu
  query = query.setParameter(0,data); // Good; Parameter bi
  org.hibernate.Query query2 = session.createQuery("FROM st
```

## See

- OWASP Top 10 2021 Category A3 Injection
- OWASP Top 10 2017 Category A1 Injection
- MITRE, CWE-89 Improper Neutralization of Special Elements used in an SQL Command
- MITRE, CWE-564 SQL Injection: Hibernate
- MITRE, CWE-20 Improper Input Validation
- <u>MITRE, CWE-943</u> Improper Neutralization of Special Elements in Data Query Logic
- CERT, IDS00-J. Prevent SQL injection
- SANS Top 25 Insecure Interaction Between Components
- Derived from FindSecBugs rules Potential SQL/JPQL Injection (JPA), Potential SQL/JDOQL Injection (JDO), Potential SQL/HQL Injection (Hibernate)

Available In:

sonarcloud 🙆 | sonarqube

© 2008-2022 SonarSource S.A., Switzerland. All content is copyright protected. SONAR, SONARSOURCE, SONARLINT, SONARQUBE and SONARCLOUD are trademarks of SonarSource S.A. All other trademarks and copyrights are the property of their respective owners. All rights are expressly reserved.

Privacy Policy