

- Secrets
- ABAP
- Apex
- C
- C++
- CloudFormation
- COBOL
- C#
- CSS
- Flex
- Go
- HTML
- Java
- JavaScript
- Kotlin
- Kubernetes
- Objective C
- PHP
- PL/I
- PL/SQL**
- Python
- RPG
- Ruby
- Scala
- Swift
- Terraform
- Text
- TypeScript
- T-SQL
- VB.NET
- VB6
- XML

PL/SQL static code analysis

Unique rules to find Bugs, Vulnerabilities, Security Hotspots, and Code Smells in your PL/SQL code

All rules 188

Vulnerability 4

Bug 45

Security Hotspot 2

Code Smell 137

Tags

Search by name...

Code Smell

Variables should be nullable

Code Smell

"VARCHAR2" should be used

Code Smell

Native SQL joins should be used

Code Smell

"FORALL" should be used

Code Smell

"FETCH ... BULK COLLECT INTO" should be used

Code Smell

Column aliases should be defined using "AS"

Code Smell

Procedures and functions should be encapsulated in packages

Code Smell

Procedures should have parameters

Code Smell

"EXECUTE IMMEDIATE" should be used instead of DBMS_SQL procedure calls

Code Smell

"NATURAL JOIN" queries should not be used

Code Smell

"END" statements of labeled loops should be labeled

Code Smell

In labeled loops "EXIT" should exit the label

Code Smell

"EXIT WHEN" should be used rather than "IF ... THEN

Variables should be nullable

Analyze your code

Code Smell Major performance

Declaring a variable with the NOT NULL constraint incurs a small performance cost - while this constraint may not really be required. Using such a constraint should be avoided.

Noncompliant Code Example

```
DECLARE
    counter PLS_INTEGER NOT NULL := 0; -- Noncompliant
BEGIN
    NULL;
END;
/
```

Compliant Solution

```
DECLARE
    counter PLS_INTEGER := 0; -- Compliant
BEGIN
    NULL;
END;
/
```

Available In:

sonarlint | sonarcloud | sonarqube Developer Edition