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

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 comply with a naming convention
Code Smell

Return of boolean expressions should not be wrapped into an "if-then-else" statement
Code Smell

Boolean literals should not be redundant
Code Smell

Comments should not be nested
Code Smell

"DBMS_UTILITY.FORMAT_ERROR_STACK" and "FORMAT_ERROR_BACKTRACE" should be used together
Code Smell

Procedures should not contain "RETURN" statements
Code Smell

Track uses of "TODO" tags
Code Smell

Neither DES (Data Encryption Standard) nor DESede (3DES) should be used
Vulnerability

"SYNCHRONIZE" should not be used
Bug

Global public variables should not be defined
Code Smell

A primary key should be specified during table creation
Code Smell

Variables should comply with a naming convention

Analyze your code

Code Smell | Minor | convention

Shared coding conventions allow teams to collaborate efficiently. This rule checks that all variable names match the provided regular expression.

Noncompliant Code Example

With the default regular expression `[a-zA-Z]([a-zA-Z0-9_]*[a-zA-Z0-9])?`:

```
DECLARE
  goodVariable PLS_INTEGER; -- Compliant
  badVariable_ PLS_INTEGER; -- Non-Compliant
BEGIN
  NULL;
END;
/
```

Available In:

sonarlint | sonarcloud | sonarqube Developer Edition