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

**"CASE" structures should not have too many "WHEN" clauses**
⊙ Code Smell

Deprecated LONG and LONG RAW datatypes should no longer be used
⊙ Code Smell

Sections of code should not be commented out
⊙ Code Smell

Unused procedure and function parameters should be removed
⊙ Code Smell

SQL EXISTS subqueries should not be used
⊙ Code Smell

Track uses of "FIXME" tags
⊙ Code Smell

Variables should not be shadowed
⊙ Code Smell

Redundant pairs of parentheses should be removed
⊙ Code Smell

Functions and procedures should not have too many parameters
⊙ Code Smell

Collapsible "if" statements should be merged
⊙ Code Smell

Unused labels should be removed
⊙ Code Smell

Compound triggers should define at least two

### "CASE" structures should not have too many "WHEN" clauses

**Analyze your code**

⊙ Code Smell   🔻 Major ⑦     🏷 brain-overload

CASE structures with a large set of WHEN clauses are difficult to understand and maintain, and should be refactored to include fewer WHEN clauses.

Available In:

sonarlint  |  sonarcloud  |  sonarqube Developer Edition