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... 🔍

Identifiers should be written in lower case

Code Smell

"PLS_INTEGER" types should be used

Code Smell

Reserved words should be written in upper case

Code Smell

Parameter "IN" mode should be specified explicitly

Code Smell

Lines in a multiline comment should start with "*"

Code Smell

CASE should be used for sequences of simple tests

Code Smell

SQL tables should be joined with the "JOIN" keyword

Code Smell

Constraint names should comply with a naming convention

Code Smell

Reserved words should be written in lower case

Code Smell

Comments should not be located at the end of lines of code

Code Smell

Lines should not end with trailing whitespaces

Code Smell

The "RELIES_ON" clause should not be used

Code Smell

Parameter "IN" mode should be specified explicitly

Analyze your code

Code Smell | 🔽 Minor ❓ | 🏷 convention

By default, the parameter mode is `IN`. However, specifying it explicitly makes the code easier to read.

Noncompliant Code Example

```
SET SERVEROUTPUT ON

DECLARE
  PROCEDURE printName(name VARCHAR2) AS -- Noncompliant; relies on default mode
  BEGIN
    DBMS_OUTPUT.PUT_LINE('name: ' || name);
  END;

BEGIN
  printName('Foo');
END;
/
```

Compliant Solution

```
SET SERVEROUTPUT ON

DECLARE
  PROCEDURE printName(name IN VARCHAR2) AS
  BEGIN
    DBMS_OUTPUT.PUT_LINE('name: ' || name);
  END;
BEGIN
  printName('Foo');
END;
/
```

Available In:

sonarlint | sonarcloud | sonarqube Developer Edition