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

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 EXIT; END IF;"
Code Smell

"FOR" loop end conditions should not be hard-coded
Code Smell

Large item lists should not be used with "IN" clauses
Code Smell

"GOTO" should not be used within loops
Code Smell

"FULL OUTER JOINS" should be used with caution
Code Smell

"CASE" should be used rather than "DECODE"
Code Smell

"CROSS JOIN" queries should not be used

Procedures should have parameters

Analyze your code

Code Smell Major design

Procedures which don't accept parameters are likely to either not be reused that often or to depend on global variables instead. Refactoring those procedures to take parameters will make them both more flexible and reusable.

Noncompliant Code Example

```
SET SERVEROUTPUT ON

DECLARE
  name VARCHAR2(42) := 'John';

  PROCEDURE print_name; -- Noncompliant

  PROCEDURE print_name AS -- Noncompliant
  BEGIN
    DBMS_OUTPUT.PUT_LINE('Name: ' || name);
  END;

BEGIN
  print_name;
END;
/
```

Compliant Solution

```
SET SERVEROUTPUT ON

DECLARE
  PROCEDURE print_name(name VARCHAR2) AS -- Compliant
  BEGIN
    DBMS_OUTPUT.PUT_LINE('Name: ' || name);
  END;
BEGIN
  print_name('John');
END;
/
```

Available In:

sonarlint | sonarcloud | sonarqube Developer Edition