



















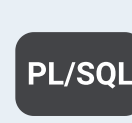













-  Secrets
-  ABAP
-  Apex
-  C
-  C++
-  CloudFormation
-  COBOL
-  C#
-  CSS
-  Flex
-  Go
-  HTML
-  Java
-  JavaScript
-  Kotlin
-  Kubernetes
-  Objective C
-  PHP
-  PL/I
-  **PL/SQL**
-  Python
-  RPG
-  Ruby
-  Scala
-  Swift
-  Terraform
-  Text
-  TypeScript
-  T-SQL
-  VB.NET
-  VB6
-  XML

PL/SQL

PL/SQL static code analysis

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




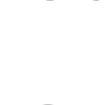







All rules 188


 Vulnerability 4


 Bug 45

 Security Hotspot 2

 Code Smell 137

"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	 Code Smell
"END" statements of labeled blocks should be labeled	 Code Smell
Two branches in a conditional structure should not have exactly the same implementation	 Code Smell
Unused assignments should be removed	 Code Smell
"LIKE" clauses should not start with wildcard characters	 Code Smell
Column names should be used in a SQL "ORDER BY"	

Tags 

Search by name... 

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

Analyze your code

 Code Smell

 Major 

 clumsy

The `EXIT WHEN` syntax can exit a loop depending on a condition. It should be preferred to the more verbose and error-prone `IF ... THEN EXIT; END IF;` syntax.

Noncompliant Code Example

```
SET SERVEROUTPUT ON

DECLARE
    i PLS_INTEGER;
BEGIN

    i := 0;
    LOOP
        IF i > 10 THEN -- Noncompliant
            EXIT;
        END IF;

        DBMS_OUTPUT.PUT_LINE('i = ' || i);
        i := i + 1;
    END LOOP;

END;
/
```

Compliant Solution

```
SET SERVEROUTPUT ON

DECLARE
    i PLS_INTEGER;
BEGIN

    i := 0;
    LOOP
        EXIT WHEN i > 10;

        DBMS_OUTPUT.PUT_LINE('i = ' || i);
        i := i + 1;
    END LOOP;

END;
/
```

Available In:

sonarlint 

sonarcloud 

sonarqube  Developer Edition