

































-  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...

🔍

All code should be reachable

 Bug

Loops with at most one iteration should be refactored

 Bug

Variables and columns should not be self-assigned

 Bug

"IF" statement conditions should not evaluate unconditionally to "TRUE" or to "FALSE"

 Bug

The "result_cache" hint should be avoided

 Bug

"GOTO" statements should not be used

 Code Smell

"TO_NUMBER" should be used with a format model

 Code Smell

Labels should not be reused in inner scopes

 Code Smell

"FUNCTIONS" should not have "OUT" parameters

 Code Smell

Variables should be nullable

 Code Smell

"VARCHAR2" should be used

 Code Smell

Native SQL joins should be used

 Code Smell

"FORALL" should be used

 Code Smell

All code should be reachable

Analyze your code

 Bug

 Major

?



cwe unused

Jump statements (EXIT, CONTINUE, RETURN, RAISE, and RAISE_APPLICATION_ERROR), move control flow out of the current code block. So any statements that come after a jump are dead code.

This rule detects for statements that follow:

- EXIT without a WHEN
- CONTINUE without a WHEN
- RETURN
- RAISE
- RAISE_APPLICATION_ERROR

Noncompliant Code Example

```
SET SERVEROUTPUT ON

BEGIN
  LOOP
    DBMS_OUTPUT.PUT_LINE('This will be printed out');
    EXIT;

    DBMS_OUTPUT.PUT_LINE('This will NEVER be printed out'); -- Non-Compliant
  END LOOP;
END;
/
```

Compliant Solution

```
SET SERVEROUTPUT ON

BEGIN
  LOOP
    DBMS_OUTPUT.PUT_LINE('This will be printed out');
    EXIT;
  END LOOP;
END;
/
```

See

- [MITRE, CWE-561](#) - Dead Code
- [CERT, MSC56-J](#). - Detect and remove superfluous code and values
- [CERT, MSC12-C](#). - Detect and remove code that has no effect or is never executed

Available In:

sonarlint

|

sonarcloud

|

sonarqube



Developer
Edition