sonar RULES

Products

- 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...

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

"FETCH ... BULK COLLECT INTO" should be used
Code Smell

Column aliases should be defined using "AS"

Loops with at most one iteration should be refactored

Analyze your code

Bug Major

A loop with at most one iteration is equivalent to the use of an `IF` statement to conditionally execute one piece of code. No developer expects to find such usage of a loop statement. If the initial intention of the author was really to conditionally execute one piece of code, an `IF` statement should be used in place.

At worst that was not the initial intention of the author and so the body of the loop should be fixed to use the nested `RETURN`, `EXIT`, `RAISE` or `GOTO` statements in a more appropriate way.

Noncompliant Code Example

```
LOOP
  counter := counter + 1;
  dbms_output.put_line(counter);
  EXIT;   -- Noncompliant
END LOOP;
```

Compliant Solution

```
LOOP
  counter := counter + 1;
  IF counter > 10 THEN
    EXIT;
  ELSE
    dbms_output.put_line(counter);
  END IF;
END LOOP;
```

Available In:
sonarlint sonarcloud sonarqube Developer Edition