

- 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...

Unary prefix operators should not be repeated

Bug

DML events clauses should not include multiple "OF" clauses

Bug

"PACKAGE BODY" initialization sections should not contain "RETURN" statements

Bug

"NULL" should not be compared directly

Bug

"MLSLABEL" should not be used

Bug

"VARCHAR2" and "NVARCHAR2" should be used

Bug

Related "IF/ELSIF" statements and "WHEN" clauses in a "CASE" should not have the same condition

Bug

Identical expressions should not be used on both sides of a binary operator

Bug

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

Unary prefix operators should not be repeated

Analyze your code

Bug Major ?

Calling the +, – or NOT prefix operator twice does nothing: the second invocation undoes the first. Such mistakes are typically caused by accidentally double-tapping the key in question without noticing.

Either this is a bug, if the operator was actually meant to be called once, or misleading if done on purpose.

Noncompliant Code Example

```
IF NOT ( NOT foo = 5 ) THEN  -- Noncompliant
    value := ++1;           -- Noncompliant
END IF;
```

Compliant Solution

```
IF foo = 5 THEN  -- Compliant
    value := +1;    -- Compliant
END IF;
```

Available In:

sonarlint | sonarcloud | sonarqube Developer Edition