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...

- 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"
  Code Smell
- Procedures and functions should be encapsulated in

Variables and columns should not be self-assigned

Analyze your code

Bug Major

There is no reason to re-assign a variable to itself. Either this statement is redundant and should be removed, or the re-assignment is a mistake and some other value or variable was intended for the assignment instead.

Noncompliant Code Example

```
UPDATE person
SET name = name;
```

Compliant Solution

```
UPDATE person
SET name = UPPER(name);
```

See

- CERT, MSC12-C. - Detect and remove code that has no effect or is never executed

Available In:
sonarlint sonarcloud sonarqube Developer Edition