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 branches in a conditional structure should not have exactly the same implementation
Bug

Strings should only be moved to variables or columns which are large enough to hold them
Bug

"WHERE" clause conditions should not be contradictory
Bug

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

All branches in a conditional structure should not have exactly the same implementation

Analyze your code

Bug | Major

Having all branches in a `CASE` or `IF/ELSIF` chain with the same implementation is an error. Either a copy-paste error was made and something different should be executed, or there shouldn't be a `CASE/IF/ELSIF` chain at all.

Noncompliant Code Example

```
IF param = 1 THEN
  result := 'A';
ELSIF param = 2 THEN
  result := 'A';
ELSE
  result := 'A';
END IF;

result := CASE param
   WHEN 1 THEN 'A'
   WHEN 2 THEN 'A'
   ELSE 'A'
END;
```

Compliant Solution

```
IF param = 1 THEN
  result := 'A';
ELSIF param = 2 THEN
  result := 'B';
ELSE
  result := 'C';
END IF;

result := CASE param
   WHEN 1 THEN 'A'
   WHEN 2 THEN 'B'
   ELSE 'C'
END;
```

Exceptions

This rule does not apply to `IF/CASE` chains without `ELSE` clauses.

```
IF param = 1 THEN   -- no issue, this could have been done on purpose to make the code more readabl
  result := 'A';
ELSIF param = 2 THEN
  result := 'A';
END IF;
```

Available In:

sonarlint | sonarcloud | sonarqube Developer Edition