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...

Code Smell

"IF ... ELSEIF" constructs should end with "ELSE" clauses
Code Smell

"WHEN OTHERS" clauses should be used for exception handling
Code Smell

"CREATE_TIMER" should not be used
Code Smell

"TO_DATE" and "TO_TIMESTAMP" should be used with a datetime format model
Code Smell

Features deprecated in Oracle 12 should not be used
Code Smell

Tables should be aliased
Code Smell

"SIMPLE_INTEGER" should be used instead of "PLS_INTEGER"
Code Smell

Queries should not "SELECT" too many columns
Code Smell

"ROWID" and "UROWID" data types should not be used
Code Smell

"RETURN" should not be used from within a loop
Code Smell

"NUMBER" variables should be declared with precision
Code Smell

"IF ... ELSEIF" constructs should end with "ELSE" clauses

Analyze your code

Code Smell Critical

This rule applies whenever an `IF` statement is followed by one or more `ELSEIF` statements; the final `ELSEIF` should be followed by an `ELSE` statement. The requirement for a final `ELSE` statement is defensive programming. The `ELSE` statement should either take appropriate action or contain a suitable comment as to why no action is taken. This is consistent with the requirement to have a final `ELSE` clause in a `CASE` statement.

Noncompliant Code Example

```
IF my_variable = 0 THEN
  do_something;
ELSIF my_variable = 1 THEN
  do_something_else;
END IF;
```

Compliant Solution

```
IF my_variable = 0 THEN
  do_something;
ELSIF my_variable = 1 THEN
  do_something_else;
ELSE
  -- Nothing has to be done.
  NULL;
END IF;
```

See

- CERT, MSC01-C. - Strive for logical completeness
- CERT, MSC57-J. - Strive for logical completeness

Available In:

sonarlint | sonarcloud | sonarqube Developer Edition

© 2008-2022 SonarSource S.A., Switzerland. All content is copyright protected. SONAR, SONARSOURCE, SONARLINT, SONARQUBE and SONARCLOUD are trademarks of SonarSource S.A. All other trademarks and copyrights are the property of their respective owners. All rights are expressly reserved.
Privacy Policy