

PL/SQL static code analysis

Unique rules to find Bugs, Vulnerabilities, Security Hotspots, and Code Smells in your PL/SQL code

Vulnerability

 Bug (45)

 Security Hotspot 2

Code Smell
137

Tags

Search by name...

 Code Smell

Track uses of "NOSONAR" comments

 Code Smell

Track comments matching a regular expression

 Code Smell

Statements should be on separate lines

 Code Smell

"WHEN" clauses should not have too many lines

 Code Smell

Magic numbers should not be used

Code Smell

Files should not have too many lines of code

 Code Smell

Lines should not be too long

 Code Smell

Explicitly opened cursors should be closed

 Bug

Identifiers should be written in lower case

 Code Smell

"PLS_INTEGER" types should be used

Code Smell

Reserved words should be written in upper case

Code Smell

Parameter "IN" mode should be specified explicitly

 Code Smell

Track uses of "NOSONAR" comments

Analyze your code

 Code Smell  Major   bad-practice

Any issue to quality rule can be deactivated with the `NOSONAR` marker. This marker is pretty useful to exclude false-positive results but it can also be used abusively to hide real quality flaws.

This rule raises an issue when NOSONAR is used.

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](#)