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

Track parsing failures
Code Smell

Files should not be too complex
Code Smell

Function and procedure parameters should comply with a naming convention
Code Smell

Magic literals should not be used
Code Smell

"UNION" should be used with caution
Code Smell

"GROUP BY" should not be used in SQL "SELECT" statements
Code Smell

Track breaches of an XPath rule
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

Track parsing failures

Analyze your code

Code Smell | Major | suspicious

When the PL/SQL parser fails, it is possible to record the failure as an issue on the file. This way, not only it is possible to track the number of files that do not parse but also to easily find out why they do not parse.

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