



PL/SQL


# PL/SQL static code analysis


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

All rules188

 Vulnerability4

 Bug45


 Security Hotspot2

 Code Smell137


Tags

Search by name...


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


Lines in a multiline comment should start with "⌘"

 Code Smell


CASE should be used for sequences of simple tests

 Code Smell


SQL tables should be joined with the "JOIN" keyword

 Code Smell


Constraint names should comply with a naming convention

 Code Smell


Reserved words should be written in lower case

 Code Smell


Comments should not be located at the end of lines of code

 Code Smell

Lines should not end with trailing whitespaces


 Code Smell


The "RELIES\_ON" clause should not be used


 Code Smell

Comments should not be located at the end of lines of code

Analyze your code

 Code Smell

 Minor

 convention

This rule verifies that single-line comments are not located at the ends of lines of code. The main idea behind this rule is that in order to be really readable, trailing comments would have to be properly written and formatted (correct alignment, no interference with the visual structure of the code, not too long to be visible) but most often, automatic code formatters would not handle this correctly: the code would end up less readable. Comments are far better placed on the previous empty line of code, where they will always be visible and properly formatted.

Noncompliant Code Example

```
a := b + c; -- This is a trailing comment that can be very very long
```

Compliant Solution

```
-- This very long comment is better placed before the line of code
a := b + c;
```

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](#)