

































-  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
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
 Code Smell
Files should not have too many lines of code
 Code Smell
Lines should not be too long
 Code Smell

Function and procedure parameters should comply with a naming convention

Analyze your code

 Code Smell

 Major 

 convention

Each function and procedure parameter name must match a given regular expression.

Noncompliant Code Example

```
DECLARE
  FUNCTION myfunction2(parameter_ PLS_INTEGER) RETURN PLS_INTEGER; -- Noncompliant

  PROCEDURE myprocedure2(parameter_ PLS_INTEGER); -- Noncompliant

  FUNCTION myfunction2(parameter_ PLS_INTEGER) RETURN PLS_INTEGER AS -- Noncompliant
  BEGIN
    RETURN 42;
  END;

  PROCEDURE myprocedure2(parameter_ PLS_INTEGER) AS -- Noncompliant
  BEGIN
    NULL;
  END;
BEGIN
  NULL;
END;
/
```

Compliant Solution

```
DECLARE
  FUNCTION myfunction1(parameter PLS_INTEGER) RETURN PLS_INTEGER; -- Compliant

  PROCEDURE myprocedure1(parameter PLS_INTEGER); -- Compliant

  FUNCTION myfunction1(parameter PLS_INTEGER) RETURN PLS_INTEGER AS -- Compliant
  BEGIN
    RETURN 42;
  END;

  PROCEDURE myprocedure1(parameter PLS_INTEGER) AS -- Compliant
  BEGIN
    NULL;
  END;

BEGIN
  NULL;
END;
/
```

Available In:

sonarlint 

sonarcloud 

sonarqube  Developer Edition