















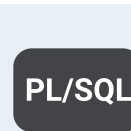


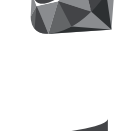





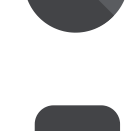









-  Secrets
-  ABAP
-  Apex
-  C
-  C++
-  CloudFormation
-  COBOL
-  C#
-  CSS
-  Flex
-  Go
-  HTML
-  Java
-  JavaScript
-  Kotlin
-  Kubernetes
-  Objective C
-  PHP
-  PL/I
-  **PL/SQL**
-  Python
-  RPG
-  Ruby
-  Scala
-  Swift
-  Terraform
-  Text
-  TypeScript
-  T-SQL
-  VB.NET
-  VB6
-  XML


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... 

Unused procedure and function parameters should be removed		Code Smell
SQL EXISTS subqueries should not be used		Code Smell
Track uses of "FIXME" tags		Code Smell
Variables should not be shadowed		Code Smell
Redundant pairs of parentheses should be removed		Code Smell
Functions and procedures should not have too many parameters		Code Smell
Collapsible "if" statements should be merged		Code Smell
Unused labels should be removed		Code Smell
Compound triggers should define at least two triggers		Code Smell
"EXIT" should not be used in loops		Code Smell
Jump statements should not be redundant		Code Smell
"EXCEPTION WHEN ... THEN" clauses should do more than "RAISE"		Code Smell

## Unused procedure and function parameters should be removed

Analyze your code

 Code Smell

 Major 

 unused

Unused parameters are misleading. Whatever the values passed to such parameters, the behavior will be the same.

### Noncompliant Code Example

```
CREATE PROCEDURE say_hello(name VARCHAR2) AS -- Noncompliant; name is not used
BEGIN
    DBMS_OUTPUT.PUT_LINE('Hello World');
END;
/
```

### Compliant Solution

```
CREATE PROCEDURE say_hello(name VARCHAR2) AS -- Compliant
BEGIN
    DBMS_OUTPUT.PUT_LINE('Hello ' || name);
END;
/
```

### See

- [CERT, MSC12-C](#). - Detect and remove code that has no effect or is never executed

Available In:

sonarlint 

sonarcloud 

sonarqube  Developer Edition