


















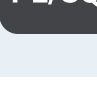














-  Secrets
-  ABAP
-  Apex
-  C
-  C++
-  CloudFormation
-  COBOL
-  C#
-  CSS
-  Flex
-  Go
-  HTML
-  Java
-  JavaScript
-  Kotlin
-  Kubernetes
-  Objective C
-  PHP
-  PL/I
-  **PL/SQL**
-  Python
-  RPG
-  Ruby
-  Scala
-  Swift
-  Terraform
-  Text
-  TypeScript
-  T-SQL
-  VB.NET
-  VB6
-  XML

PL/SQL 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...

🔍

Whitespace and control characters in string literals should be explicit

 Code Smell

Blocks containing "EXECUTE IMMEDIATE" should trap all exceptions

 Code Smell

"EXCEPTION_INIT -20,NNN" calls should be centralized

 Code Smell

"CREATE OR REPLACE" should be used instead of "CREATE"

 Code Smell

Block labels should appear on the same lines as "END"

 Code Smell

"LOOP ... END LOOP;" constructs should be avoided

 Code Smell

"IF" statements should not be nested too deeply

 Code Smell

"CASE" expressions should end with "ELSE" clauses

 Code Smell

String literals should not be duplicated

 Code Smell

Constant names should comply with a naming convention

 Code Smell

Output parameters should be assigned

 Bug

"ROWNUM" should not be used at the same query level as "ORDER BY"

 Bug

All branches in a conditional structure should not have exactly the same implementation

Whitespace and control characters in string literals should be explicit

Analyze your code

 Code Smell  Critical   pitfall

New lines and control characters can be injected in the source code by bad manipulations. Control characters aren't visible to maintainers, so whether or not they are actually wanted should be double-checked. Note that this rule can optionally also report violations on literals containing the tabulation character.

Noncompliant Code Example

```
SET SERVEROUTPUT ON

BEGIN
  /* Non-Compliant */ DBMS_OUTPUT.PUT_LINE('Hello
world!');

  DBMS_OUTPUT.PUT_LINE('Hello'); -- Compliant, this is preferred
  DBMS_OUTPUT.PUT_LINE('world!');
END;
/
```

Exceptions

By default no issue will be raised on tabulation characters. This exception can be disabled.

Available In:

 |  |  Developer Edition