


















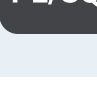














-  Secrets
-  ABAP
-  Apex
-  C
-  C++
-  CloudFormation
-  COBOL
-  C#
-  CSS
-  Flex
-  Go
-  HTML
-  Java
-  JavaScript
-  Kotlin
-  Kubernetes
-  Objective C
-  PHP
-  PL/I
-  **PL/SQL**
-  Python
-  RPG
-  Ruby
-  Scala
-  Swift
-  Terraform
-  Text
-  TypeScript
-  T-SQL
-  VB.NET
-  VB6
-  XML

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...

🔍


"MLSLABEL" should not be used

 Bug

"VARCHAR2" and "NVARCHAR2" should be used

 Bug

Related "IF/ELSIF" statements and "WHEN" clauses in a "CASE" should not have the same condition

 Bug

Identical expressions should not be used on both sides of a binary operator

 Bug

All code should be reachable

 Bug

Loops with at most one iteration should be refactored

 Bug

Variables and columns should not be self-assigned

 Bug

"IF" statement conditions should not evaluate unconditionally to "TRUE" or to "FALSE"

 Bug

The "result_cache" hint should be avoided

 Bug

"GOTO" statements should not be used

 Code Smell

"TO_NUMBER" should be used with a format model

 Code Smell

Labels should not be reused in inner scopes

 Code Smell

"FUNCTIONS" should not have "OUT" parameters

 Code Smell

"MLSLABEL" should not be used

Analyze your code

 Bug

 Major



 obsolete

The deprecated MLSLABEL datatype is still supported only for backwards compatibility with Trusted Oracle, and since Oracle8, the only valid value it can hold is NULL. Thus, the usage of this type should be progressively removed.

Noncompliant Code Example

```
DECLARE
    foo MLSLABEL; -- Noncompliant
BEGIN
    NULL;
END;
/
```

Available In:

 |  |  Developer Edition