

































-  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...



 Bug

"DELETE" and "UPDATE" statements should contain "WHERE" clauses

 Bug

"FORALL" statements should use the "SAVE EXCEPTIONS" clause

 Bug

Pipelined functions should have at least one "PIPE ROW" statement and not return an expression (PLS-00633)

 Bug

GOTO should not be used to jump backwards

 Code Smell

Quoted identifiers should not be used

 Code Smell

Loop start and end labels should match

 Code Smell

Block start and end labels should match

 Code Smell

Cipher algorithms should be robust

 Vulnerability

"COMMIT" should not be used inside a loop

 Bug

Individual "WHERE" clause conditions should not be unconditionally true or false

 Bug

Nullable subqueries should not be used in "NOT IN" conditions

 Bug

"COMMIT" and "ROLLBACK" should not be called from non-autonomous transaction triggers

 Bug

Positional and named arguments should not be mixed in invocations

Anchored types should not be constrained

Analyze your code

 Bug

 Blocker



Anchored types, i.e. those specified using either %TYPE or %ROWTYPE, cannot be constrained. Trying to do so results in the exception PLS-00573: cannot constrain scale, precision, or range of an anchored type being raised.

Noncompliant Code Example

```
DECLARE
    foo DUAL.DUMMY%TYPE(42); -- Non-Compliant - raises PLS-00573
BEGIN
    NULL;
END;
/
```

Compliant Solution

```
DECLARE
    foo DUAL.DUMMY%TYPE; -- Compliant
BEGIN
    NULL;
END;
/
```

Available In:

 |  | 

Developer Edition