

































-  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...



"END LOOP" should be followed by a semicolon

 Bug

Constraints should not be applied to types that cannot be constrained

 Bug

Improper constraint forms should not be used

 Bug

Scale should not be specified for float types

 Bug

Constant declarations should contain initialization assignments

 Bug

"RAISE_APPLICATION_ERROR" should only be used with error codes from -20,000 to -20,999

 Bug

"FETCH ... BULK COLLECT INTO" should not be used without a "LIMIT" clause

 Bug

Anchored types should not be constrained

 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

Constraints should not be applied to types that cannot be constrained

Analyze your code

 Bug

 Blocker



Some types cannot be constrained, and attempting to do so results in the exception `PLS-00566: type name "..." cannot be constrained` being raised.

Noncompliant Code Example

```
DECLARE
    foo BLOB(42); -- Noncompliant - raises PLS-00566: type name "BLOB" cannot be constrained
BEGIN
    NULL;
END;
/
```

Compliant Solution

```
DECLARE
    foo BLOB;
BEGIN
    NULL;
END;
/
```

Available In:

sonarlint



sonarcloud



sonarqube



Developer
Edition