

































-  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

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

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

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

Analyze your code

 Bug

 Blocker



 error-handling

RAISE_APPLICATION_ERROR may only be called with an error code from −20,000 to −20,999, which is the range reserved for application errors. When called with another value, Oracle raises the exception: `ORA-21000: error number argument to raise_application_error of 0 is out of range.`

Noncompliant Code Example

```
BEGIN
    RAISE_APPLICATION_ERROR(0, 'This is an application error'); -- Non-Compliant - raises ORA-21000
END;
/
```

Compliant Solution

```
BEGIN
    RAISE_APPLICATION_ERROR(-20000, 'This is an application error'); -- Compliant
END;
/
```

Available In:

sonarlint



| **sonarcloud**



| **sonarqube**



Developer Edition