



















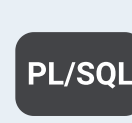













-  Secrets
-  ABAP
-  Apex
-  C
-  C++
-  CloudFormation
-  COBOL
-  C#
-  CSS
-  Flex
-  Go
-  HTML
-  Java
-  JavaScript
-  Kotlin
-  Kubernetes
-  Objective C
-  PHP
-  PL/I
-  **PL/SQL**
-  Python
-  RPG
-  Ruby
-  Scala
-  Swift
-  Terraform
-  Text
-  TypeScript
-  T-SQL
-  VB.NET
-  VB6
-  XML

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... 

| |
|--|
|  Code Smell |
| "WHEN OTHERS" clauses should be used for exception handling  Code Smell |
| "CREATE_TIMER" should not be used  Code Smell |
| "TO_DATE" and "TO_TIMESTAMP" should be used with a datetime format model  Code Smell |
| Features deprecated in Oracle 12 should not be used  Code Smell |
| Tables should be aliased  Code Smell |
| "SIMPLE_INTEGER" should be used instead of "PLS_INTEGER"  Code Smell |
| Queries should not "SELECT" too many columns  Code Smell |
| "ROWID" and "UROWID" data types should not be used  Code Smell |
| "RETURN" should not be used from within a loop  Code Smell |
| "NUMBER" variables should be declared with precision  Code Smell |
| Nested subqueries should be avoided  Code Smell |

"WHEN OTHERS" clauses should be used for exception handling

Analyze your code

 Code Smell

 Critical 

 cwe error-handling

Ensure that every possible exception is caught by using a `WHEN OTHERS` clause.

Noncompliant Code Example

```
SET SERVEROUTPUT ON

DECLARE
    result PLS_INTEGER;
    custom_exception EXCEPTION;
BEGIN
    result := 42 / 0;                                -- "Unexpected" division by 0

    RAISE custom_exception;
EXCEPTION
    WHEN custom_exception THEN                        -- Non-Compliant
        DBMS_OUTPUT.PUT_LINE ('custom_exception: ' || DBMS_UTILITY.FORMAT_ERROR_STACK);
END;
/
```

Compliant Solution

```
SET SERVEROUTPUT ON

DECLARE
    result PLS_INTEGER;
    custom_exception EXCEPTION;
BEGIN
    result := 42 / 0;                                -- "Unexpected" division by 0

    RAISE custom_exception;
EXCEPTION
    WHEN custom_exception THEN                        -- Compliant
        DBMS_OUTPUT.PUT_LINE ('custom_exception: ' || DBMS_UTILITY.FORMAT_ERROR_STACK);
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE ('other: ' || DBMS_UTILITY.FORMAT_ERROR_STACK);
END;
/
```

See

- [MITRE, CWE-391](#) - Unchecked Error Condition

Available In:

sonarlint 

sonarcloud 

sonarqube  Developer Edition