

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
"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
Nested loops should be labeled  Code Smell
Nested blocks should be labeled  Code Smell
"RESULT_CACHE" should not be used  Code Smell
Columns should be aliased  Code Smell
Track parsing failures Code Smell
Files should not be too complex

"SIMPLE_INTEGER" should be used instead of "PLS_INTEGER"

Analyze your code

 Code Smell Major error-handling performance

ORACLE 11g introduced the `SIMPLE_INTEGER` data type, which is a sub-type of `PLS_INTEGER`, and covers the same range. There are three main differences between the two types:

- `SIMPLE_INTEGER` is always `NOT NULL`. So when the value of the declared variable is never going to be null, you can declare it as `SIMPLE_INTEGER`.
- You will never face a numeric overflow using `SIMPLE_INTEGER` because this data type wraps around without giving any error.
- The `SIMPLE_INTEGER` data type gives a major performance boost over `PLS_INTEGER` when the code is compiled in "NATIVE" mode, because arithmetic operations on `SIMPLE_INTEGER` type are performed directly at the hardware level.

Noncompliant Code Example

```
DECLARE
    v1 PLS_INTEGER; -- Noncompliant
    v2 VARCHAR2(10);

BEGIN
    NULL;

END;

/
```

Compliant Solution

```
DECLARE
    v1 SIMPLE_INTEGER := 42;
    v2 VARCHAR2(10);
BEGIN
    NULL;
END;
/
```

Available In:

sonarlint  | **sonarcloud**  | **sonarqube**  Developer Edition