













**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
<b>Tables should be aliased</b>  Code Smell
<b>"SIMPLE_INTEGER" should be used instead of "PLS_INTEGER"</b>  Code Smell
<b>Queries should not "SELECT" too many columns</b>  Code Smell
<b>"ROWID" and "UROWID" data types should not be used</b>  Code Smell
<b>"RETURN" should not be used from within a loop</b>  Code Smell
<b>"NUMBER" variables should be declared with precision</b>  Code Smell
<b>Nested subqueries should be avoided</b>  Code Smell
<b>Nested loops should be labeled</b>  Code Smell
<b>Nested blocks should be labeled</b>  Code Smell
<b>"RESULT_CACHE" should not be used</b>  Code Smell
<b>Columns should be aliased</b>  Code Smell
<b>Track parsing failures</b>

## Tables should be aliased

## Analyze your code

 Code Smell
  Major
 
 brain-overload sql

When multiple tables are involved in a query, using table aliases helps to make it more understandable and keeps it short.

## Noncompliant Code Example

```
BEGIN
  SELECT
    name,
    firstname,
    location
  INTO employeesArray
  FROM employee -- Noncompliant - should be aliased
  INNER JOIN department -- Noncompliant - should be aliased
  ON employee.DepartmentID = department.ID;
END;
/
```

## Compliant Solution

```
BEGIN
  SELECT
    empl.name,
    empl.firstname,
    dpt.location
  INTO employeesArray
  FROM employee empl
  INNER JOIN department dpt
  ON empl.DepartmentID = dpt.ID;
END;
```

Available In:

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