

**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

**"END" statements of labeled blocks should be labeled**

 Code Smell

## Two branches in a conditional structure should not have exactly the same implementation

 Code Smell

## Unused assignments should be removed

 Code Smell

## "LIKE" clauses should not start with wildcard characters

 Code Smell

Column names should be used in a SQL "ORDER BY" clause

 Code Smell

## Procedures and functions should be documented

## Code Smell

## SQL statements should not join too many tables

 Code Smell

Function and procedure names should comply with a naming convention

 Code Smell

Columns to be read with a "SELECT" statement should be clearly defined

 Code Smell

"CASE" structures should not have too many "WHEN" clauses

 Code Smell

## Deprecated LONG and LONG RAW datatypes should no longer be used

## "CROSS JOIN" queries should not be used

## Analyze your code

 Code Smell  Major   suspicious

A **CROSS JOIN** query will return all records where each row from the first table is combined with each row from the second table. This means that such a query returns the Cartesian product of the sets of rows from the joined tables, which is why it is also known as "Cartesian product query".

Such a query can return a huge amount of data, and therefore should be used only with great caution and only when really needed.

## Noncompliant Code Example

```
BEGIN
    -- Standard ANSI syntax
    SELECT *
        INTO employeeArray
        FROM employee CROSS JOIN department; -- Noncompliant; explicit cross join
END;
/

BEGIN
    -- Old syntax
    SELECT *
        INTO employeeArray
        FROM employee, department; -- Noncompliant; also a cross join
END;
/
```

Available In:

sonarlint  | sonarcloud  | sonarqube  Developer Edition