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 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

"CASE" should be used rather than "DECODE"
Code Smell

"CROSS JOIN" queries should not be used
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" should be used rather than "DECODE"

Analyze your code

Code Smell | Major | obsolete

`DECODE` is an old function that has been replaced by the easier to understand and more common `CASE`. Unlike `DECODE`, `CASE` may also be used directly within PL/SQL.

Noncompliant Code Example

```
SET SERVEROUTPUT ON

DECLARE
  operand CHAR(1) := 'B';
  l_result PLS_INTEGER;
BEGIN
  -- Noncompliant
  SELECT DECODE(operand, 'A', 1
                       , 'B', 2
                       , 'C', 3
                       , 'D', 4
                       , 'E', 5
                       , 'F', 6
                       , 7)
  INTO l_result
  FROM dual;

  DBMS_OUTPUT.PUT_LINE('l_result = ' || l_result); -- 2
END;
/
```

Compliant Solution

```
SET SERVEROUTPUT ON

DECLARE
  operand CHAR(1) := 'B';
  l_result PLS_INTEGER;
BEGIN

  l_result := CASE operand
                WHEN 'A' THEN 1
                WHEN 'B' THEN 2
                WHEN 'C' THEN 3
                WHEN 'D' THEN 4
                WHEN 'E' THEN 5
                WHEN 'F' THEN 6
                ELSE 7
              END;

  DBMS_OUTPUT.PUT_LINE('l_result = ' || l_result); -- 2
END;
/
```

Exceptions

No issue is raised when `DECODE` contains only one case (i.e. only one `search` and one `result` components) because using `CASE` would make the code less readable in this scenario.

Available In:
sonarlint | sonarcloud | sonarqube Developer Edition