

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

"cursor%NOTFOUND" should be used instead of "NOT cursor%FOUND"

 Code Smell

Object attributes should comply with a naming convention

 Code Smell

Record fields should comply with a naming convention

 Code Smell

Cursors should follow a naming convention

 Code Smell

Types should follow a naming convention

 Code Smell

Cursor parameters should follow a naming convention

 Code Smell

Exceptions should follow a naming convention

 Code Smell

Exceptions should not be ignored

 Code Smell

Variables should not be initialized with "NULL"

 Code Smell

Boolean checks should not be inverted

 Code Smell

Unused local variables should be removed

 Code Smell

Package names should comply with a naming

"cursor%NOTFOUND" should be used instead of "NOT cursor%FOUND"

Analyze your code

 Code Smell
  Minor
  clumsy

`cursor%NOTFOUND` is clearer and more readable than `NOT cursor%FOUND`, and is preferred.

Noncompliant Code Example

```

SET SERVEROUTPUT ON

DECLARE

    CURSOR c IS SELECT DUMMY FROM DUAL;
    x VARCHAR2(1);

BEGIN

    OPEN c;

    FETCH c INTO x;

    IF NOT c%FOUND THEN    -- Noncompliant
        DBMS_OUTPUT.PUT_LINE('uh?');
    ELSE
        DBMS_OUTPUT.PUT_LINE('all good: ' || x);
    END IF;

    CLOSE c;

END;

/

```

Compliant Solution

```
SET SERVEROUTPUT ON

DECLARE

  CURSOR c IS SELECT DUMMY FROM DUAL;
  x VARCHAR2(1);

BEGIN

  OPEN c;

  FETCH c INTO x;

  IF c%NOTFOUND THEN

    DBMS_OUTPUT.PUT_LINE('uh?');

  ELSE

    DBMS_OUTPUT.PUT_LINE('all good: ' || x);

  END IF;

  CLOSE c;

END;
```

Available In:

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