

**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>"EXCEPTION WHEN ... THEN" clauses should do more than "RAISE"</b>	
 Code Smell	
<b>Single line comments should start with "--"</b>	
 Code Smell	
<b>An "ORDER BY" direction should be specified explicitly</b>	
 Code Smell	
<b>Oracle's join operator (+) should not be used</b>	
 Code Smell	
<b>"cursor%NOTFOUND" should be used instead of "NOT cursor%FOUND"</b>	
 Code Smell	
<b>Object attributes should comply with a naming convention</b>	
 Code Smell	
<b>Record fields should comply with a naming convention</b>	
 Code Smell	
<b>Cursors should follow a naming convention</b>	
 Code Smell	
<b>Types should follow a naming convention</b>	
 Code Smell	
<b>Cursor parameters should follow a naming convention</b>	
 Code Smell	
<b>Exceptions should follow a naming convention</b>	
 Code Smell	

## "EXCEPTION WHEN ... THEN" clauses should do more than "RAISE"

## Analyze your code

Code Smell Minor ? error-handling unused finding clumsy

An `EXCEPTION WHEN ... THEN` clause that only rethrows the caught exception has the same effect as omitting the `EXCEPTION` clause altogether and letting it bubble up automatically, but with more code and the additional detriment of leaving maintainers scratching their heads.

Such clauses should either be eliminated or populated with the appropriate logic.

## Noncompliant Code Example

```
BEGIN
    SELECT 1/0;
EXCEPTION
    WHEN ZERO_DIVIDE THEN
        RAISE; -- Noncompliant
    WHEN OTHERS THEN
        RAISE; -- Noncompliant
END;
```

## Compliant Solution

```
BEGIN
    SELECT 1/0;
EXCEPTION
    WHEN ZERO_DIVIDE THEN -- Compliant: handles 'division by zero' error
        -- do something to manage the division by zero
    COMMIT;

    WHEN OTHERS THEN
        ROLLBACK;
END;
```

Available In:

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