
















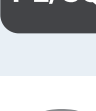
















-  Secrets
-  ABAP
-  Apex
-  C
-  C++
-  CloudFormation
-  COBOL
-  C#
-  CSS
-  Flex
-  Go
-  HTML
-  Java
-  JavaScript
-  Kotlin
-  Kubernetes
-  Objective C
-  PHP
-  PL/I
-  **PL/SQL**
-  Python
-  RPG
-  Ruby
-  Scala
-  Swift
-  Terraform
-  Text
-  TypeScript
-  T-SQL
-  VB.NET
-  VB6
-  XML

PL/SQL

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

 Bug
Global public variables should not be defined
 Code Smell
A primary key should be specified during table creation
 Code Smell
Track lack of copyright and license headers
 Code Smell
SHA-1 and Message-Digest hash algorithms should not be used in secure contexts
 Vulnerability
Sensitive "SYS" owned functions should not be used
 Vulnerability
"FORMS_DDL('COMMIT')" and "FORMS_DDL('ROLLBACK')" should not be used
 Bug
"DBMS_OUTPUT.PUT_LINE" should not be used
 Code Smell
"WHEN OTHERS" should not be the only exception handler
 Code Smell
"INSERT" statements should explicitly list the columns to be set
 Code Smell
%TYPE" and %ROWTYPE" should not be used in package specification
 Code Smell
Functions and procedures should not be too complex
 Code Smell

Tags ▾

Search by name... 🔍

Global public variables should not be defined

Analyze your code

 Code Smell

 Blocker



 design

When data structures (scalar variables, collections, cursors) are declared in the package specification (not within any specific program), they can be referenced directly by any program running in a session with `EXECUTE` rights to the package.

Instead, declare all package-level data in the package body and provide getter and setter functions in the package specification. Developers can then access the data using these methods and will automatically follow all rules you set upon data modification.

By doing so you can guarantee data integrity, change your data structure implementation, and also track access to those data structures.

Noncompliant Code Example

```
-- Package specification
CREATE PACKAGE employee AS
    name VARCHAR2(42); -- Non-Compliant
END employee;
/

DROP PACKAGE employee;
```

Compliant Solution

```
-- Package specification
CREATE PACKAGE employee AS
    PROCEDURE setName (newName VARCHAR2);
    FUNCTION getName RETURN VARCHAR2;
END employee;
/

-- Package body
CREATE PACKAGE BODY employee AS
    name VARCHAR2(42);

    PROCEDURE setName (newName VARCHAR2) IS
    BEGIN
        name := newName;
    END;

    FUNCTION getName RETURN VARCHAR2 IS
    BEGIN
        RETURN name;
    END;
END employee;
/

DROP PACKAGE BODY employee;

DROP PACKAGE employee;
```

Available In:

sonarlint 

sonarcloud 

sonarqube  Developer Edition