

































-  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

Tags



Search by name...



"WHERE" clauses

 Bug

"FORALL" statements should use the "SAVE EXCEPTIONS" clause

 Bug

Pipelined functions should have at least one "PIPE ROW" statement and not return an expression (PLS-00633)

 Bug

GOTO should not be used to jump backwards

 Code Smell

Quoted identifiers should not be used

 Code Smell

Loop start and end labels should match

 Code Smell

Block start and end labels should match

 Code Smell

Cipher algorithms should be robust

 Vulnerability

"COMMIT" should not be used inside a loop

 Bug

Individual "WHERE" clause conditions should not be unconditionally true or false

 Bug

Nullable subqueries should not be used in "NOT IN" conditions

 Bug

"COMMIT" and "ROLLBACK" should not be called from non-autonomous transaction triggers

 Bug

Positional and named arguments should not be mixed in invocations

 Bug

Functions should end with "RETURN" statements

"DELETE" and "UPDATE" statements should contain "WHERE" clauses

Analyze your code

 Bug

 Blocker



 sql

UPDATE and DELETE statements should contain WHERE clauses to keep the modification of records under control. Otherwise unexpected data loss could result.

Noncompliant Code Example

```
DECLARE
    maxAge PLS_INTEGER := 60;
BEGIN
    UPDATE employee SET status = 'retired'; -- Noncompliant - the WHERE was forgotten
END;
/
```

Compliant Solution

```
DECLARE
    maxAge PLS_INTEGER := 60;
BEGIN
    UPDATE employee SET status = 'retired' WHERE age > maxAge;
END;
/
```

Available In:

sonarlint



sonarcloud



sonarqube



Developer  
Edition