


















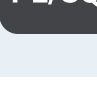














-  Secrets
-  ABAP
-  Apex
-  C
-  C++
-  CloudFormation
-  COBOL
-  C#
-  CSS
-  Flex
-  Go
-  HTML
-  Java
-  JavaScript
-  Kotlin
-  Kubernetes
-  Objective C
-  PHP
-  PL/I
-  **PL/SQL**
-  Python
-  RPG
-  Ruby
-  Scala
-  Swift
-  Terraform
-  Text
-  TypeScript
-  T-SQL
-  VB.NET
-  VB6
-  XML

# 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...

🔍

convention

 Code Smell

Output parameters should be assigned

 Bug

"ROWNUM" should not be used at the same query level as "ORDER BY"

 Bug

All branches in a conditional structure should not have exactly the same implementation

 Bug

Strings should only be moved to variables or columns which are large enough to hold them

 Bug

"WHERE" clause conditions should not be contradictory

 Bug

Unary prefix operators should not be repeated

 Bug

DML events clauses should not include multiple "OF" clauses

 Bug

"PACKAGE BODY" initialization sections should not contain "RETURN" statements

 Bug

"NULL" should not be compared directly

 Bug

"MLSLABEL" should not be used

 Bug

"VARCHAR2" and "NVARCHAR2" should be used

 Bug

Related "IF/ELSIF" statements and "WHEN" clauses in a "CASE" should not have the same condition

## Constant names should comply with a naming convention

Analyze your code

 Code Smell

 Critical



 convention

Shared coding conventions allow teams to collaborate efficiently. This rule checks that all constant names match a provided regular expression.

### Noncompliant Code Example

With the default regular expression `[ a-zA-Z ] ( [ a-zA-Z0-9_ ] * [ a-zA-Z0-9 ] ) ?`:

```
DECLARE
    constant_ CONSTANT PLS_INTEGER := 42; -- Noncompliant
BEGIN
    NULL;
END;
/
```

### Compliant Solution

```
DECLARE
    constant CONSTANT PLS_INTEGER := 42; -- Compliant
BEGIN
    NULL;
END;
/
```

Available In:

 |  | 

Developer Edition