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...

"VARCHAR2" and "NVARCHAR2" should be used
Bug

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

Identical expressions should not be used on both sides of a binary operator
Bug

All code should be reachable
Bug

Loops with at most one iteration should be refactored
Bug

Variables and columns should not be self-assigned
Bug

"IF" statement conditions should not evaluate unconditionally to "TRUE" or to "FALSE"
Bug

The "result_cache" hint should be avoided
Bug

"GOTO" statements should not be used
Code Smell

"TO_NUMBER" should be used with a format model
Code Smell

Labels should not be reused in inner scopes
Code Smell

"FUNCTIONS" should not have "OUT" parameters
Code Smell

Variables should be nullable
Code Smell

"VARCHAR2" and "NVARCHAR2" should be used

Analyze your code

Bug Major

For fixed-length values, a `CHAR` field occupies the same amount of disk space as a `VARCHAR2` field, but for variable-length values `CHAR` fields use more storage space and make searching more difficult by right-padding values with whitespaces. Therefore `VARCHAR2` fields are preferred. Similarly, `NCHAR` should be replaced by `NVARCHAR2`.

Note that for 1-character fields, `CHAR` is naturally equivalent to `VARCHAR2`, but the latter is still preferred for consistency.

Noncompliant Code Example

```
DECLARE
  var1 CHAR; -- Noncompliant

  var2 CHAR(42); -- Noncompliant

  var3 NCHAR; -- Noncompliant

  var4 NCHAR(42); -- Noncompliant
BEGIN
  NULL;
END;
/
```

Compliant Solution

```
DECLARE
  var1 VARCHAR2(42);

  var2 VARCHAR2(42);

  var3 NVARCHAR2(42);

  var4 NVARCHAR2(42);
BEGIN
  NULL;
END;
/
```

Available In:

sonarlint | sonarcloud | sonarqube Developer Edition