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...

Code Smell

Record fields should comply with a naming convention
Code Smell

Cursors should follow a naming convention
Code Smell

Types should follow a naming convention
Code Smell

Cursor parameters should follow a naming convention
Code Smell

Exceptions should follow a naming convention
Code Smell

Exceptions should not be ignored
Code Smell

Variables should not be initialized with "NULL"
Code Smell

Boolean checks should not be inverted
Code Smell

Unused local variables should be removed
Code Smell

Package names should comply with a naming convention
Code Smell

Variables should comply with a naming convention
Code Smell

Return of boolean expressions should not be

Record fields should comply with a naming convention

Analyze your code

Code Smell | Minor | convention

Shared coding conventions allow teams to collaborate efficiently. This rule checks that all record field names match the provided regular rexpression.

Noncompliant Code Example

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

```
DECLARE
  TYPE my_type IS RECORD(
    foo__bar PLS_INTEGER   -- Non-Compliant
  );
BEGIN
  NULL;
END;
/
```

Compliant Solution

```
DECLARE
  TYPE my_type IS RECORD(
    foo_bar PLS_INTEGER    -- Compliant
  );
BEGIN
  NULL;
END;
/
```

Available In:

sonarlint | sonarcloud | sonarqube Developer Edition