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...

"SELECT" statements used as argument of "EXISTS" statements should be selective
Code Smell

"LIKE" clauses should not be used without wildcards
Code Smell

Weak "REF CURSOR" types should not be used
Code Smell

Whitespace and control characters in string literals should be explicit
Code Smell

Blocks containing "EXECUTE IMMEDIATE" should trap all exceptions
Code Smell

"EXCEPTION_INIT -20,NNN" calls should be centralized
Code Smell

"CREATE OR REPLACE" should be used instead of "CREATE"
Code Smell

Block labels should appear on the same lines as "END"
Code Smell

"LOOP ... END LOOP;" constructs should be avoided
Code Smell

"IF" statements should not be nested too deeply
Code Smell

"CASE" expressions should end with "ELSE" clauses
Code Smell

String literals should not be duplicated
Code Smell

Constant names should comply with a naming convention

"SELECT" statements used as argument of "EXISTS" statements should be selective

Analyze your code

Code Smell | Critical

An "EXISTS" statement is generally used to select/update/delete some rows of a table based on the content of columns of other tables.

If the "SELECT" statement used as argument of the "EXISTS" statement is always returning "true" for all rows of the main `SELECT` statement, the `EXISTS` statement is useless and has the same effect as if it was not there. Still, this is probably not the original intend of the developer to have an `EXISTS` statement that is always true.

As a consequence, the `SELECT` statement of an `EXISTS` statement should always contain a `WHERE` clause.

What is true for `EXISTS` is also true for `NOT EXISTS`.

Noncompliant Code Example

```
SELECT  *
FROM    sys.[databases] AS [sd]
WHERE EXISTS (SELECT  1
              FROM    [sys].[master_files] AS [mf])
```

Compliant Solution

```
SELECT  *
FROM    sys.[databases] AS [sd]
WHERE EXISTS (SELECT  1
              FROM    [sys].[master_files] AS [mf]
              WHERE [mf].[database_id] = [sd].[database_id])
```

Exceptions

This rule doesn't raise an issue when `EXISTS` is used in the context of a `WHILE` or a `IF` statement.

Available In:

sonarlint | sonarcloud | sonarqube Developer Edition