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

T-SQL static code analysis

Unique rules to find Bugs, Vulnerabilities, Security Hotspots, and Code Smells in your T-SQL code

All rules 80 | Vulnerability 1 | Bug 16 | Security Hotspot 4 | Code Smell 59

Tags

Search by name...

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

Size should be specified for "varchar" variables and parameters
Code Smell

Conditionally executed code should be denoted by either indentation or BEGIN...END block
Code Smell

Conditionals should start on new lines
Code Smell

"INSERT" statements should explicitly list the columns to be set
Code Smell

Output parameters should be assigned
Bug

Queries that use "TOP" should have an "ORDER BY"
Bug

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

"WHERE" clause conditions should not be contradictory
Bug

Unary prefix operators should not be repeated
Bug

"NULL" should not be compared directly
Bug

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

"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

© 2008-2022 SonarSource S.A., Switzerland. All content is copyright protected. SONAR, SONARSOURCE, SONARLINT, SONARQUBE and SONARCLOUD are trademarks of SonarSource S.A. All other trademarks and copyrights are the property of their respective owners. All rights are expressly reserved.
Privacy Policy