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...

"CASE" input expressions should be invariant
Bug

Nullable subqueries should not be used in "NOT IN" conditions
Bug

Using weak hashing algorithms is security-sensitive
Security Hotspot

Dynamically executing code is security-sensitive
Security Hotspot

"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

"CASE" input expressions should be invariant

Analyze your code

Bug | Critical | unpredictable

Under the covers, Simple `CASE` expressions are evaluated as searched `CASE` expressions. That is,

```
CASE @foo
WHEN 1 THEN 'a'
WHEN 2 THEN 'b'
```

is actually evaluated as

```
CASE
WHEN @foo = 1 THEN 'a'
WHEN @foo = 2 THEN 'b'
```

In most situations the difference is inconsequential, but when the input expression isn't fixed, for instance if `RAND()` is involved, it is likely to yield unexpected results. For that reason, it is better to evaluate the input expression once, assign it to a variable, and use the variable as the `CASE`'s input expression.

This rule raises an issue when any of the following is used in a `CASE` input expression: `RAND`, `NEWID`, `CRYPT_GEN_RANDOM`.

Noncompliant Code Example

```
CASE CONVERT(SMALLINT, RAND()*@foo)  -- Noncompliant
WHEN 1 THEN 'a'
WHEN 2 THEN 'b'
```

Compliant Solution

```
DECLARE @bar SMALLINT = CONVERT(SMALLINT, RAND()*@foo)
CASE @bar
WHEN 1 THEN 'a'
WHEN 2 THEN 'b'
```

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