


















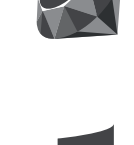




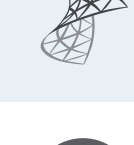









-  Secrets
-  ABAP
-  Apex
-  C
-  C++
-  CloudFormation
-  COBOL
-  C#
-  CSS
-  Flex
-  Go
-  HTML
-  Java
-  JavaScript
-  Kotlin
-  Kubernetes
-  Objective C
-  PHP
-  PL/I
-  PL/SQL
-  Python
-  RPG
-  Ruby
-  Scala
-  Swift
-  Terraform
-  Text
-  TypeScript
-  **T-SQL**
-  VB.NET
-  VB6
-  XML














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... 🔍


Expressions should not be too complex		Code Smell
"WHERE" clauses should not contain redundant conditions		Bug
Track lack of SQL Server session configuration		Code Smell
Duplicate values should not be passed as arguments		Code Smell
Track parsing failures		Code Smell
Functions and stored procedure should not have too many lines of code		Code Smell
Track uses of "NOSONAR" comments		Code Smell
Statements should be on separate lines		Code Smell
"WHEN" clauses should not have too many lines of code		Code Smell
Files should not have too many lines of code		Code Smell
Lines should not be too long		Code Smell
Reserved keywords should not be used as identifiers or object names		

Expressions should not be too complex

Analyze your code

 Code Smell

 Critical ?

 brain-overload

The complexity of an expression is defined by the number of AND and OR operators it contains.

A single expression's complexity should not become too high to keep the code readable.

Noncompliant Code Example

With the default threshold value of 3

```
IF ((@a = 1 AND @b > 2) OR (@c <> 3 AND @d <= 4)) AND @e IS NULL
...

```

Available In:

sonarlint

sonarcloud

sonarqube

Developer Edition