



ABAP

Apex

C++

CloudFormation

COBOL

C#

CSS

Flex

Go

HTML

Java

JavaScript

Kotlin

Kubernetes

Objective C

PL/I

Python

PL/SQL

RPG

Ruby

Scala

Swift

Terraform

Text

TypeScript



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



6 Vulnerability 1



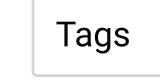








Code Smell (59)



Search by name...

Loops with at most one iteration should be refactored

Bug

Variables should not be self-assigned

Bug

"GOTO" statements should not be used

Code Smell

Deprecated system tables and views should not be used

Code Smell

"ANSI_NULLS", "ANSI_PADDING" and "CONCAT_NULL_YIELDS_NULL" should not be configured

Code Smell

"@@IDENTITY" should not be used

Code Smell

"COALESCE", "IIF", and "CASE" input expressions should not contain subqueries

Code Smell

"CHECK" or "NOCHECK" should be specified explicitly when constraints are activated

Code Smell

Deprecated features should not be used

Code Smell

Multiline blocks should be enclosed in BEGIN...END blocks

Code Smell

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

Code Smell

"LIKE" clauses should not start with wildcard characters

Loops with at most one iteration should be refactored

Analyze your code

Rug Major 🕝

A WHILE loop with at most one iteration is equivalent to the use of an IF statement to conditionally execute one piece of code. No developer expects to find such usage of a loop statement. If the initial intention of the author was really to conditionally execute one piece of code, an IF statement should be used in place.

At worst that was not the initial intention of the author and so the body of the loop should be fixed to use the nested RETURN, BREAK or THROW statements in a more appropriate way.

Noncompliant Code Example

```
WHILE @cond -- noncompliant, loop only executes once
BEGIN
 EXEC something;
  BREAK;
END;
WHILE @cond1 -- noncompliant, loop only executes once
BEGIN
 IF @cond2
  BEGIN
    EXEC something;
    BREAK;
  BEGIN
    RETURN @value;
  END;
END;
```

Compliant Solution

```
IF @cond
BEGIN
 EXEC something;
 BREAK;
END;
• • •
WHILE @cond
BEGIN
 IF @cond2
  BEGIN
    EXEC something;
  END ELSE
  BEGIN
    RETURN @value;
  END;
END;
```

Available In:

sonarlint sonarcloud sonarqube Developer Edition