Search by name...



Secrets

ABAP

Apex

C++

CloudFormation

COBOL

C#

CSS

Flex

Go

HTML

JavaScript

Java

Kotlin

Kubernetes

Objective C

PL/I

PL/SQL

Python

RPG

Ruby

Scala

Terraform

TypeScript

T-SQL

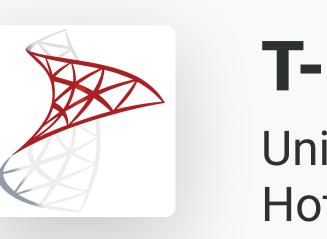
Swift

Text

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

Code Smell (59) All rules 80 6 Vulnerability 1 **R** Bug (16) Security Hotspot 4

WHEN 3

ELSE 'D'

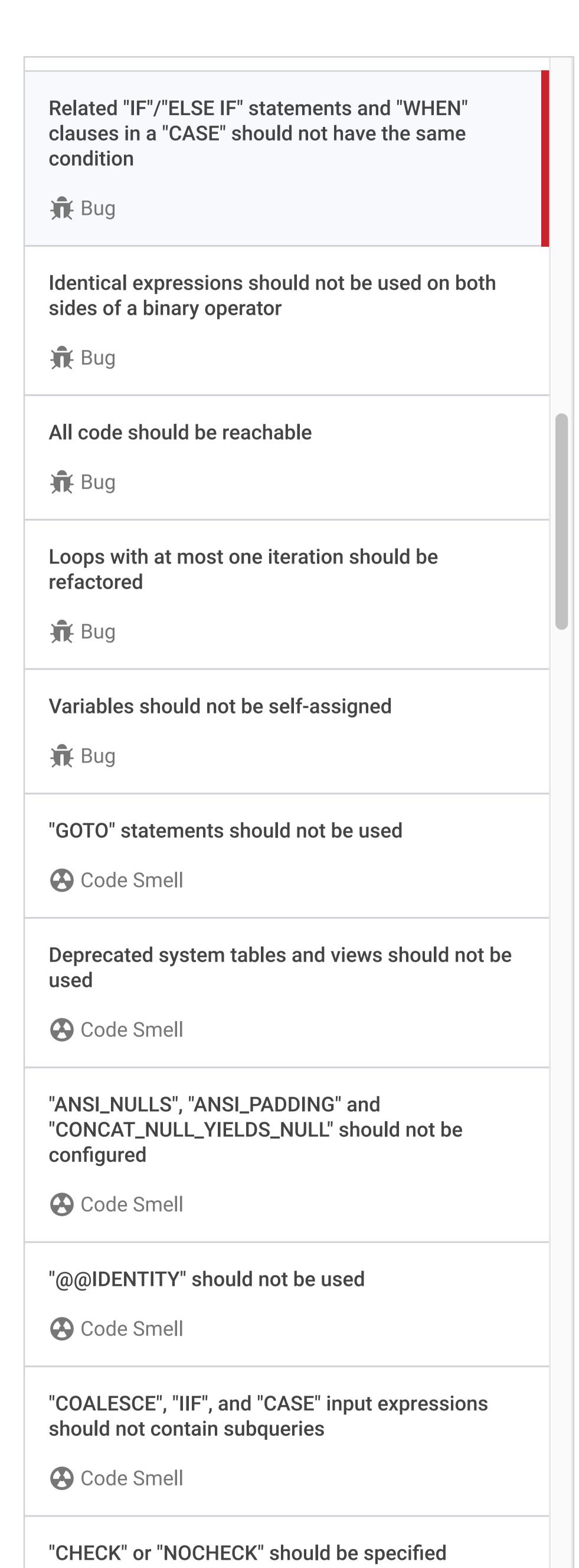
END

See

FROM table1

Available In:

THEN 'C'



explicitly when constraints are activated

Code Smell

```
Related "IF"/"ELSE IF" statements and "WHEN" clauses in a "CASE" should
                                                                                          Analyze your code
not have the same condition
Reg Major unused pitfall
A CASE and a chain of IF/ELSE IF statements is evaluated from top to bottom. At most, only one branch will be executed: the first
one with a condition that evaluates to true.
Therefore, duplicating a condition automatically leads to dead code. Usually, this is due to a copy/paste error. At best, it's simply dead
code and at worst, it's a bug that is likely to induce further bugs as the code is maintained, and obviously it could lead to unexpected
behavior.
Noncompliant Code Example
  IF @x = 1
    PRINT 'A'
  ELSE IF @x = 2
    PRINT 'B'
  ELSE IF @x = 1 -- Noncompliant
    PRINT 'C'
  SELECT
    CASE col1
       WHEN 1
         THEN 'A'
       WHEN 2
       WHEN 1 -- Noncompliant
        THEN 'C'
      ELSE 'D'
    END
  FROM table1
Compliant Solution
  IF @x = 1
    PRINT 'A'
  ELSE IF @x = 2
    PRINT 'B'
  ELSE IF @x = 3
    PRINT 'C'
  SELECT
    CASE col1
       WHEN 1
         THEN 'A'
       WHEN 2
         THEN 'B'
```

Tags

CERT, MSC12-C. - Detect and remove code that has no effect or is never executed

sonarlint sonarcloud sonarqube Developer Edition