

- Secrets
- SAP ABAP
- Apex Apex
- C
- C++
- CloudFormation
- COBOL COBOL
- C# C#
- CSS
- X Flex
- GO GO
- Java

HTML

- Js JavaScript
- Kubernetes

Kotlin

- **Ó** Objective C
- PL/I
- PL/SQL
- Python
- RPG RPG
- Ruby
- Scala
- Swift
- Terraform
- **T**ext
- Ts TypeScript
- 1) | |



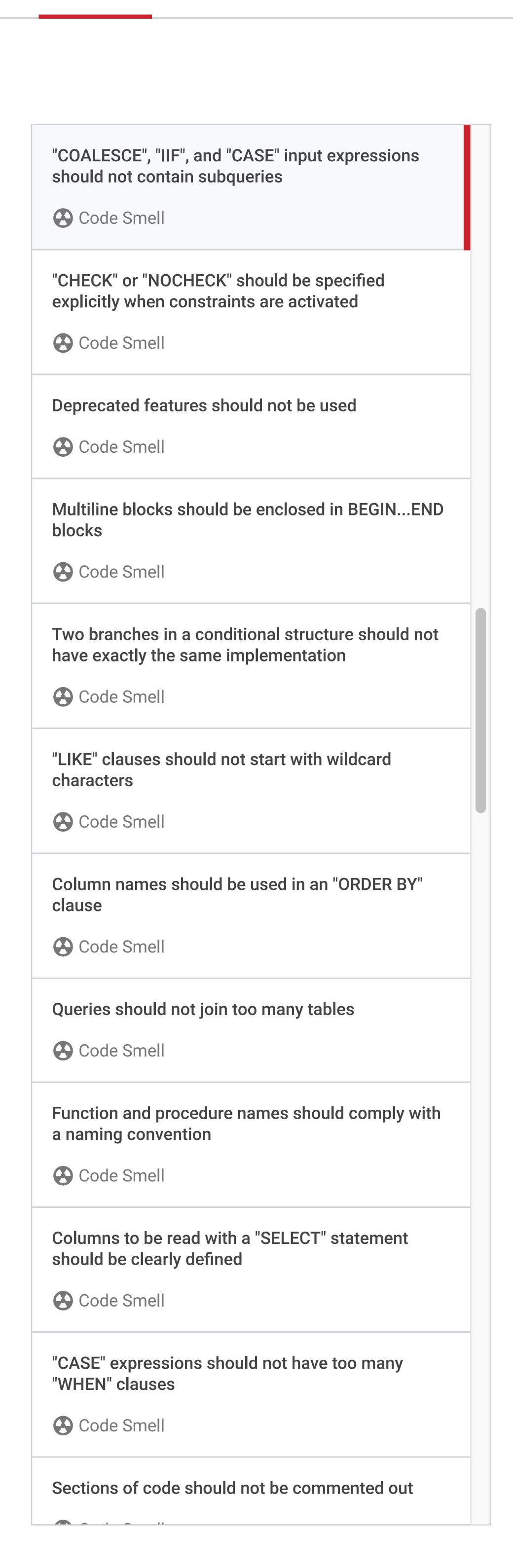
VB.NET

- VB6 VB6
- XML XML



T-SQL static code analysis

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



```
Tags
                                                                                             Search by name...
"COALESCE", "IIF", and "CASE" input expressions should not contain
                                                                                        Analyze your code
subqueries
COALESCE and IIF (which evaluate to CASE expressions under the covers), as well as CASE input expressions should not be used with
subqueries because the subquery will be evaluated once for each option in the expression, and each evaluation could return different
results depending on the isolation level. To ensure consistent results, use the SNAPSHOT ISOLATION isolation level. To ensure
consistent results and better performance, move the subquery out of the expression.
Note it is also an option to replace COALESCE with ISNULL.
Noncompliant Code Example
 COALESCE((SELECT a FROM b WHERE c) , 1) -- Noncompliant
  • • •
  • • •
 CASE
  WHEN (SELECT COUNT(*) FROM A) > 0 THEN (SELECT COUNT(*) FROM A) + 42
  • • •
  ELSE otherExpression
  END
  • • •
Compliant Solution
  SET @a = SELECT a FROM b WHERE c
  • • •
 COALESCE(@a, 1)
  • • •
or
  SET TRANSACTION ISOLATION LEVEL SNAPSHOT
  BEGIN TRANSACTION
  COALESCE ((SELECT a FROM b WHERE c) , 1)
  • • •
  • • •
  SET @a = SELECT COUNT(*) FROM A
  CASE
  WHEN @a > 0 THEN @a + 42
```

ELSE otherExpression

sonarlint sonarcloud sonarqube Developer Edition

END

• • •

Available In: