





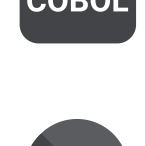



























-  Secrets
-  ABAP
-  Apex
-  C
-  C++
-  CloudFormation
-  COBOL
-  C#
-  CSS
-  Flex
-  Go
-  HTML
-  Java
-  JavaScript
-  Kotlin
-  Kubernetes
-  Objective C
-  PHP
-  PL/I
-  PL/SQL
-  Python
-  RPG
-  Ruby
-  Scala
-  Swift
-  Terraform
-  Text
-  TypeScript
-  **T-SQL**
-  VB.NET
-  VB6
-  XML





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... 

"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	 Bug
"WHERE" clause conditions should not be contradictory	 Bug
Unary prefix operators should not be repeated	 Bug
"NULL" should not be compared directly	 Bug
Related "IF"/"ELSE IF" statements and "WHEN" clauses in a "CASE" should not have the same condition	 Bug
Identical expressions should not be used on both sides of a binary operator	 Bug
All code should be reachable	 Bug
Loops with at most one iteration should be refactored	 Bug
Variables should not be self-assigned	

"INSERT" statements should explicitly list the columns to be set

[Analyze your code](#)

 Code Smell

 Critical



 sql

An `INSERT` statement that does not explicitly list the columns being inserted into, as well as the values being inserted, is dependent for correct functioning on the structure of the table not changing. Additionally, not having the explicit column list degrades the readability and understandability of the code. Therefore, `INSERT` statements should always contain an explicit column list.

Noncompliant Code Example

```
INSERT INTO PERSONS VALUES (1, 'DUPONT', 'Marcel')
```

Compliant Solution

```
INSERT INTO PERSONS (ID, LAST_NAME, FIRST_NAME)
VALUES (1, 'DUPONT', 'Marcel')
```

Available In:

sonarlint 

sonarcloud 

sonarqube  Developer Edition