















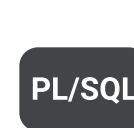


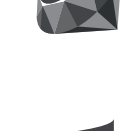




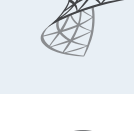









-  Secrets
-  ABAP
-  Apex
-  C
-  C++
-  CloudFormation
-  COBOL
-  C#
-  CSS
-  Flex
-  Go
-  HTML
-  Java
-  JavaScript
-  Kotlin
-  Kubernetes
-  Objective C
-  PHP
-  PL/I
-  PL/SQL
-  Python
-  RPG
-  Ruby
-  Scala
-  Swift
-  Terraform
-  Text
-  TypeScript
-  **T-SQL**
-  VB.NET
-  VB6
-  XML














# 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... 

"LIKE" clauses should not be used without wildcards
 Code Smell
Jump statements should not be redundant
 Code Smell
"CATCH" clauses should do more than rethrow
 Code Smell
Boolean checks should not be inverted
 Code Smell
Multiple variables should not be declared on the same line
 Code Smell
Unused local variables should be removed
 Code Smell
Local variable and parameter names should comply with a naming convention
 Code Smell
Empty statements should be removed
 Code Smell
Track uses of "TODO" tags
 Code Smell
A primary key should be specified during table creation
 Code Smell
Track lack of copyright and license headers
 Code Smell
SHA-1 and Message-Digest hash algorithms should not be used in secure contexts

## "LIKE" clauses should not be used without wildcards

[Analyze your code](#)

 Code Smell

 Minor 

 sql

The use of `LIKE` in a SQL query without one or more wildcards in the sought value is suspicious. A maintainer can suppose that either `=` was meant instead, or that the wildcard was unintentionally omitted.

Note that in some cases using `LIKE` without a wildcard may return different results than the use of `=`. Thus, the use of `LIKE` without a wildcard may be intentional. However, it is highly likely to confuse maintainers who either are unaware of this fact, or don't understand that such circumstances apply to the query in question.

### Noncompliant Code Example

```
SELECT name
FROM product
WHERE name LIKE 'choc'
```

### Compliant Solution

```
SELECT name
FROM product
WHERE name LIKE 'choc%'
```

or

```
SELECT name
FROM product
WHERE name = 'choc'
```

Available In:

sonarlint 

sonarcloud 

sonarqube  Developer Edition