















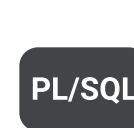


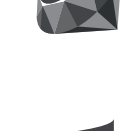




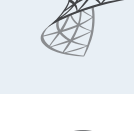









-  Secrets
-  ABAP
-  Apex
-  C
-  C++
-  CloudFormation
-  COBOL
-  C#
-  CSS
-  Flex
-  Go
-  HTML
-  Java
-  JavaScript
-  Kotlin
-  Kubernetes
-  Objective C
-  PHP
-  PL/I
-  PL/SQL
-  Python
-  RPG
-  Ruby
-  Scala
-  Swift
-  Terraform
-  Text
-  TypeScript
-  **T-SQL**
-  VB.NET
-  VB6
-  XML















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... 🔍

 Code Smell
Duplicate values should not be passed as arguments
 Code Smell
Track parsing failures
 Code Smell
Functions and stored procedure should not have too many lines of code
 Code Smell
Track uses of "NOSONAR" comments
 Code Smell
Statements should be on separate lines
 Code Smell
"WHEN" clauses should not have too many lines of code
 Code Smell
Files should not have too many lines of code
 Code Smell
Lines should not be too long
 Code Smell
Reserved keywords should not be used as identifiers or object names
 Code Smell
Non-standard comparison operators should not be used
 Code Smell
Tabulation characters should not be used
 Code Smell

Files should not have too many lines of code

[Analyze your code](#)

 Code Smell  Major   brain-overload

A source file that grows too much tends to aggregate too many responsibilities and inevitably becomes harder to understand and therefore to maintain. Above a specific threshold, it is strongly advised to refactor it into smaller pieces of code which focus on well defined tasks. Those smaller files will not only be easier to understand but also probably easier to test.

Available In:

sonarlint  | **sonarcloud**  | **sonarqube**  Developer Edition

© 2008-2022 SonarSource S.A., Switzerland. All content is copyright protected. SONAR, SONARSOURCE, SONARLINT, SONARQUBE and SONARCLOUD are trademarks of SonarSource S.A. All other trademarks and copyrights are the property of their respective owners. All rights are expressly reserved.
[Privacy Policy](#)