















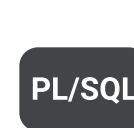


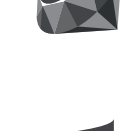




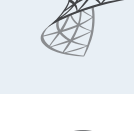









-  Secrets
-  ABAP
-  Apex
-  C
-  C++
-  CloudFormation
-  COBOL
-  C#
-  CSS
-  Flex
-  Go
-  HTML
-  Java
-  JavaScript
-  Kotlin
-  Kubernetes
-  Objective C
-  PHP
-  PL/I
-  PL/SQL
-  Python
-  RPG
-  Ruby
-  Scala
-  Swift
-  Terraform
-  Text
-  TypeScript
-  **T-SQL**
-  VB.NET
-  VB6
-  XML














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... 

"WHERE" clauses should not contain redundant conditions
 Bug
Track lack of SQL Server session configuration
 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

"WHERE" clauses should not contain redundant conditions

[Analyze your code](#)

 Bug  Major   sql

If a `WHERE` clause contains a condition which is redundant in the context of the the rest of the clause, then it can be removed. If it is redundant because it does not match the programmer's intent, then it's a bug and the sub-condition should be fixed.

Noncompliant Code Example

```
SELECT name, price
FROM product
WHERE price > 15  -- Noncompliant
      AND price < 100  -- Noncompliant
      AND price = 50
```

Compliant Solution

```
SELECT name, price
FROM product
WHERE price = 50
```

Available In:

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