

































-  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... 

"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
 Vulnerability
"NOCOUNT" should be activated on "PROCEDURE" and "TRIGGER" definitions
 Code Smell
Control flow statements "IF" "WHILE" and

"CATCH" clauses should do more than rethrow

[Analyze your code](#)

 Code Smell

 Minor 

 error-handling unused finding clumsy

A `CATCH` clause that only rethrows the caught exception has the same effect as omitting the `CATCH` altogether and letting it bubble up automatically, but with more code and the additional detriment of leaving maintainers scratching their heads.

Such clauses should either be eliminated or populated with the appropriate logic.

Noncompliant Code Example

```
BEGIN TRY
    SELECT 1/0;
END TRY
BEGIN CATCH -- Noncompliant
    THROW;
END CATCH;
```

Compliant Solution

```
BEGIN TRY
    SELECT 1/0;
END TRY
BEGIN CATCH
    EXECUTE usp_GetErrorInfo;
    THROW;
END CATCH;
```

or

```
SELECT 1/0;
```

Available In:

sonarlint 

sonarcloud 

sonarqube  Developer Edition