


















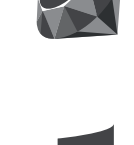




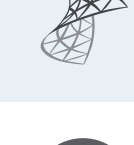









-  Secrets
-  ABAP
-  Apex
-  C
-  C++
-  CloudFormation
-  COBOL
-  C#
-  CSS
-  Flex
-  Go
-  HTML
-  Java
-  JavaScript
-  Kotlin
-  Kubernetes
-  Objective C
-  PHP
-  PL/I
-  PL/SQL
-  Python
-  RPG
-  Ruby
-  Scala
-  Swift
-  Terraform
-  Text
-  TypeScript
-  **T-SQL**
-  VB.NET
-  VB6
-  XML















# 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... 

"IF ... ELSEIF" constructs should end with "ELSE" clauses	 Code Smell
Control structures should use BEGIN...END blocks	 Code Smell
String literals should not be duplicated	 Code Smell
Expressions should not be too complex	 Code Smell
"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
Track use of SQL Server system functions	 Code Smell

## "IF ... ELSEIF" constructs should end with "ELSE" clauses

Analyze your code

 Code Smell  Critical 

This rule applies whenever an `IF` statement is followed by one or more `ELSE IF` statements; the final `ELSE IF` should be followed by an `ELSE` statement.

The requirement for a final `ELSE` statement is defensive programming.

The `ELSE` statement should either take appropriate action or contain a suitable comment as to why no action is taken. This is consistent with the requirement to have a final `ELSE` clause in a `CASE` statement.

### Noncompliant Code Example

```
IF @x = 1
    PRINT 'A'
ELSE IF @x = 2
    PRINT 'B'
ELSE IF @x = 3
    PRINT 'C'
-- Noncompliant; final ELSE is missing
```

### Compliant Solution

```
IF @x = 1
    PRINT 'A'
ELSE IF @x = 2
    PRINT 'B'
ELSE IF @x = 3
    PRINT 'C'
ELSE
    PRINT 'Z'
```

### See

- [CERT, MSC01-C](#) - Strive for logical completeness
- [CERT, MSC57-J](#) - Strive for logical completeness

Available In:

 |  |  Developer Edition