

# T-SQL static code analysis

Unique rules to find Bugs, Vulnerabilities, Security Hotspots, and Code Smells in your T-SQL code

1.	"DELETE" and "UPDATE" statements should contain "WHERE" clauses <a href="#">Bug</a>
2.	Hard-coded credentials are security-sensitive <a href="#">Security Hotspot</a>
3.	The number of variables in a FETCH statement should match the number of columns in the cursor <a href="#">Bug</a>
4.	"CASE" input expressions should be invariant <a href="#">Bug</a>
5.	Nullable subqueries should not be used in "NOT IN" conditions <a href="#">Bug</a>
6.	Using weak hashing algorithms is security-sensitive <a href="#">Security Hotspot</a>
7.	Dynamically executing code is security-sensitive <a href="#">Security Hotspot</a>
8.	"SELECT" statements used as argument of "EXISTS" statements should be selective <a href="#">Code Smell</a>
9.	Size should be specified for "varchar" variables and parameters <a href="#">Code Smell</a>
10.	Conditionally executed code should be denoted by either indentation or BEGIN...END block <a href="#">Code Smell</a>
11.	Conditionals should start on new lines <a href="#">Code Smell</a>
12.	"INSERT" statements should explicitly list the columns to be set <a href="#">Code Smell</a>
13.	Output parameters should be assigned <a href="#">Bug</a>
14.	Queries that use "TOP" should have an "ORDER BY" <a href="#">Bug</a>
15.	All branches in a conditional structure should not have exactly the same implementation

	<a href="#">Bug</a>
16.	"WHERE" clause conditions should not be contradictory <a href="#">Bug</a>
17.	Unary prefix operators should not be repeated <a href="#">Bug</a>
18.	"NULL" should not be compared directly <a href="#">Bug</a>
19.	Related "IF"/"ELSE IF" statements and "WHEN" clauses in a "CASE" should not have the same condition <a href="#">Bug</a>
20.	Identical expressions should not be used on both sides of a binary operator <a href="#">Bug</a>
21.	All code should be reachable <a href="#">Bug</a>
22.	Loops with at most one iteration should be refactored <a href="#">Bug</a>
23.	Variables should not be self-assigned <a href="#">Bug</a>
24.	"GOTO" statements should not be used <a href="#">Code Smell</a>
25.	Deprecated system tables and views should not be used <a href="#">Code Smell</a>
26.	"ANSI_NULLS", "ANSI_PADDING" and "CONCAT_NULL_YIELDS_NULL" should not be configured <a href="#">Code Smell</a>
27.	"@@IDENTITY" should not be used <a href="#">Code Smell</a>
28.	"COALESCE", "IIF", and "CASE" input expressions should not contain subqueries <a href="#">Code Smell</a>
29.	"CHECK" or "NOCHECK" should be specified explicitly when constraints are activated <a href="#">Code Smell</a>
30.	Deprecated features should not be used <a href="#">Code Smell</a>
31.	Multiline blocks should be enclosed in BEGIN...END blocks <a href="#">Code Smell</a>

32.	Two branches in a conditional structure should not have exactly the same implementation <a href="#">Code Smell</a>
33.	"LIKE" clauses should not start with wildcard characters <a href="#">Code Smell</a>
34.	Column names should be used in an "ORDER BY" clause <a href="#">Code Smell</a>
35.	Queries should not join too many tables <a href="#">Code Smell</a>
36.	Function and procedure names should comply with a naming convention <a href="#">Code Smell</a>
37.	Columns to be read with a "SELECT" statement should be clearly defined <a href="#">Code Smell</a>
38.	"CASE" expressions should not have too many "WHEN" clauses <a href="#">Code Smell</a>
39.	Sections of code should not be commented out <a href="#">Code Smell</a>
40.	Unused procedure and function parameters should be removed <a href="#">Code Smell</a>
41.	Track uses of "FIXME" tags <a href="#">Code Smell</a>
42.	Redundant pairs of parentheses should be removed <a href="#">Code Smell</a>
43.	Functions and procedures should not have too many parameters <a href="#">Code Smell</a>
44.	Collapsible "if" statements should be merged <a href="#">Code Smell</a>
45.	Unused labels should be removed <a href="#">Code Smell</a>
46.	Using hardcoded IP addresses is security-sensitive <a href="#">Security Hotspot</a>
47.	Column references should not have more than two-parts <a href="#">Code Smell</a>
48.	Triggers should not "PRINT", "SELECT", or "FETCH"

	<a href="#">Code Smell</a>
49.	"LIKE" clauses should not be used without wildcards <a href="#">Code Smell</a>
50.	Jump statements should not be redundant <a href="#">Code Smell</a>
51.	"CATCH" clauses should do more than rethrow <a href="#">Code Smell</a>
52.	Boolean checks should not be inverted <a href="#">Code Smell</a>
53.	Multiple variables should not be declared on the same line <a href="#">Code Smell</a>
54.	Unused local variables should be removed <a href="#">Code Smell</a>
55.	Local variable and parameter names should comply with a naming convention <a href="#">Code Smell</a>
56.	Empty statements should be removed <a href="#">Code Smell</a>
57.	Track uses of "TODO" tags <a href="#">Code Smell</a>
58.	A primary key should be specified during table creation <a href="#">Code Smell</a>
59.	Track lack of copyright and license headers <a href="#">Code Smell</a>
60.	SHA-1 and Message-Digest hash algorithms should not be used in secure contexts <a href="#">Vulnerability</a>
61.	"NOCOUNT" should be activated on "PROCEDURE" and "TRIGGER" definitions <a href="#">Code Smell</a>
62.	Control flow statements "IF", "WHILE" and "TRY" should not be nested too deeply <a href="#">Code Smell</a>
63.	"CASE" expressions should end with "ELSE" clauses <a href="#">Code Smell</a>
64.	"IF ... ELSEIF" constructs should end with "ELSE" clauses <a href="#">Code Smell</a>
65.	Control structures should use BEGIN...END blocks

	<a href="#">Code Smell</a>
66.	String literals should not be duplicated <a href="#">Code Smell</a>
67.	Expressions should not be too complex <a href="#">Code Smell</a>
68.	"WHERE" clauses should not contain redundant conditions <a href="#">Bug</a>
69.	Track lack of SQL Server session configuration <a href="#">Code Smell</a>
70.	Duplicate values should not be passed as arguments <a href="#">Code Smell</a>
71.	Track parsing failures <a href="#">Code Smell</a>
72.	Functions and stored procedure should not have too many lines of code <a href="#">Code Smell</a>
73.	Track uses of "NOSONAR" comments <a href="#">Code Smell</a>
74.	Statements should be on separate lines <a href="#">Code Smell</a>
75.	"WHEN" clauses should not have too many lines of code <a href="#">Code Smell</a>
76.	Files should not have too many lines of code <a href="#">Code Smell</a>
77.	Lines should not be too long <a href="#">Code Smell</a>
78.	Reserved keywords should not be used as identifiers or object names <a href="#">Code Smell</a>
79.	Non-standard comparison operators should not be used <a href="#">Code Smell</a>
80.	Tabulation characters should not be used <a href="#">Code Smell</a>