# 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...

- "NULL" should not be compared directly
  Bug
- Related "IF"/"ELSE IF" statements and "WHEN" clauses in a "CASE" should not have the same condition
  Bug
- Identical expressions should not be used on both sides of a binary operator
  Bug
- All code should be reachable
  Bug
- Loops with at most one iteration should be refactored
  Bug
- Variables should not be self-assigned
  Bug
- "GOTO" statements should not be used
  Code Smell
- Deprecated system tables and views should not be used
  Code Smell
- "ANSI_NULLS", "ANSI_PADDING" and "CONCAT_NULL_YIELDS_NULL" should not be configured
  Code Smell
- "@@IDENTITY" should not be used
  Code Smell
- "COALESCE", "IIF", and "CASE" input expressions should not contain subqueries
  Code Smell

## "NULL" should not be compared directly

Analyze your code

Bug Major sql

In a Zen-like manner, "NULL" is never equal to anything, even itself. Therefore comparisons using equality operators will always return `False`, even when the value actually `IS NULL`.

For that reason, comparison operators should never be used to make comparisons with `NULL`; `IS NULL` and `IS NOT NULL` should be used instead.

### Noncompliant Code Example

```
UPDATE books
SET title = 'unknown'
WHERE title = NULL -- Noncompliant
```

### Compliant Solution

```
UPDATE books
SET title = 'unknown'
WHERE title IS NULL
```

Available In:

sonarlint sonarcloud sonarqube Developer Edition