



# 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 

 Code Smell 

## Tags

Search by name...

**"@@IDENTITY" should not be used**

 Code Smell

**"COALESCE", "IIF", and "CASE" input expressions should not contain subqueries**

 Code Smell

**"CHECK" or "NOCHECK" should be specified explicitly when constraints are activated**

 Code Smell

## Deprecated features should not be used

 Code Smell

Multiline blocks should be enclosed in BEGIN...END blocks

 Code Smell

## Two branches in a conditional structure should not have exactly the same implementation

 Code Smell

## "LIKE" clauses should not start with wildcard characters

 Code Smell

Column names should be used in an "ORDER BY" clause

 Code Smell

## Queries should not join too many tables

 Code Smell

Function and procedure names should comply with a naming convention

 Code Smell

Columns to be read with a "SELECT" statement should be clearly defined

 Code Smell

"CASE" expressions should not have too many "WHEN" clauses

**"@@IDENTITY" should not be used**

## Analyze your code

Code Smell   Major   unpredictable   suspicious

@@IDENTITY returns the last identity column value created on a connection, regardless of the scope. That means it could return the last identity value you produced, or it could return a value generated by a user defined function or trigger, possibly one fired because of your insert. In order to access the last identity value created in your scope, use SCOPE\_IDENTITY() instead.

## Noncompliant Code Example

```
INSERT ...

SET @id = @@IDENTITY -- Noncompliant
```

## Compliant Solution

```
INSERT ...

SET @id = SCOPE_IDENTITY()
```

Available In:

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