

T-SQL static code analysis: Nullable subqueries should not be used in "NOT IN" conditions

1-2 minutes

A WHERE clause condition that uses NOT IN with a subquery will have unexpected results if that subquery returns NULL. On the other hand, NOT EXISTS subqueries work reliably under the same conditions.

This rule raises an issue when NOT IN is used with a subquery. This rule doesn't check if the selected column is a nullable column because the rules engine has no information about the table definition. It's up to the developer to review manually if the column is nullable.

Noncompliant Code Example

```
SELECT *  
FROM my_table  
WHERE my_column NOT IN (SELECT nullable_column FROM  
another_table) -- Noncompliant; "nullable_column" may contain  
'NULL' value and the whole SELECT query will return nothing
```

Compliant Solution

```
SELECT *  
FROM my_table
```

```
WHERE NOT EXISTS (SELECT 1 FROM another_table  
WHERE nullable_column = my_table.my_column)
```

```
SELECT *
```

```
FROM my_table
```

```
WHERE my_column NOT IN (SELECT nullable_column FROM  
another_table WHERE nullable_column IS NOT NULL)
```