

































-  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

PL/SQL static code analysis

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

All rules

188

 Vulnerability

4

 Bug

45

 Security Hotspot

2

 Code Smell

137

Tags

▼

Search by name...

🔍

"TO_NUMBER" should be used with a format model

 Code Smell

Labels should not be reused in inner scopes

 Code Smell

"FUNCTIONS" should not have "OUT" parameters

 Code Smell

Variables should be nullable

 Code Smell

"VARCHAR2" should be used

 Code Smell

Native SQL joins should be used

 Code Smell

"FORALL" should be used

 Code Smell

"FETCH ... BULK COLLECT INTO" should be used

 Code Smell

Column aliases should be defined using "AS"

 Code Smell

Procedures and functions should be encapsulated in packages

 Code Smell

Procedures should have parameters

 Code Smell

"EXECUTE IMMEDIATE" should be used instead of DBMS_SQL procedure calls

 Code Smell

"NATURAL JOIN" queries should not be used

 Code Smell

"TO_NUMBER" should be used with a format model

Analyze your code

 Code Smell

 Major

?

The TO_NUMBER function is converting the value of BINARY_FLOAT, BINARY_DOUBLE, CHAR, VARCHAR2, NCHAR, or NVARCHAR2 datatype to a value of NUMBER datatype.

Without providing the format of the input, TO_NUMBER will consider the provided value is compliant with the default format. Relying on a default configuration is a source of error because it creates a dependency between the code and the configuration of the ORACLE server where this code is deployed.

The behaviour of the TO_NUMBER function will certainly not be the expected one if the configuration of the ORACLE server is changing.

Noncompliant Code Example

The following code with return 0 on an ORACLE server running with its default US configuration with p_string = "2,540"

```
IF ( TO_NUMBER(p_string) >= 0 and TO_NUMBER(p_string) <= TO_NUMBER('50.00') ) THEN
    RETURN 1;
ELSE
    RETURN 0;
END IF;
```

Compliant Solution

The following code with return 1 on an ORACLE server running with its default FR configuration with p_string = "2,540" because the comma will be interpreted as decimal separator instead of thousand separator.

```
IF ( TO_NUMBER(p_string, '99.99') >= 0 and TO_NUMBER(p_string, '99.99') <= TO_NUMBER('50.00','99.99') ) THEN
    RETURN 1;
ELSE
    RETURN 0;
END IF;
```

Available In:

sonarlint



|

sonarcloud



|

sonarqube



Developer Edition