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

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...

Code Smell

"TO_DATE" and "TO_TIMESTAMP" should be used with a datetime format model
Code Smell

Features deprecated in Oracle 12 should not be used
Code Smell

Tables should be aliased
Code Smell

"SIMPLE_INTEGER" should be used instead of "PLS_INTEGER"
Code Smell

Queries should not "SELECT" too many columns
Code Smell

"ROWID" and "UROWID" data types should not be used
Code Smell

"RETURN" should not be used from within a loop
Code Smell

"NUMBER" variables should be declared with precision
Code Smell

Nested subqueries should be avoided
Code Smell

Nested loops should be labeled
Code Smell

Nested blocks should be labeled
Code Smell

"TO_DATE" and "TO_TIMESTAMP" should be used with a datetime format model

Analyze your code

Code Smell | Major

The `TO_DATE` and `TO_TIMESTAMP` functions are converting char of `CHAR`, `VARCHAR2`, `NCHAR`, or `NVARCHAR2` datatype to respectively a value of `DATE` or `TIMESTAMP` datatype.

Without providing the format of the input char, `TO_DATE` will consider the provided char is compliant with the default date 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.

According to the Oracle's documentation "the default date format is determined implicitly by the NLS_TERRITORY initialization parameter or can be set explicitly by the NLS_DATE_FORMAT parameter.". It means the behaviour of the `TO_DATE` function will certainly not be the expected one if the configuration of the ORACLE server is changing.

Noncompliant Code Example

```
SELECT TO_DATE( 'January 15, 2018, 11:00 A.M.')
FROM DUAL;
```

Compliant Solution

```
SELECT TO_DATE( 'January 15, 2018, 11:00 A.M.', 'Month dd, YYYY, HH:MI A.M.')
FROM DUAL;
```

Available In:

sonarlint | sonarcloud | sonarqube Developer Edition