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...

Vulnerability

"FORMS_DDL('COMMIT')" and "FORMS_DDL('ROLLBACK')" should not be used
Bug

"DBMS_OUTPUT.PUT_LINE" should not be used
Code Smell

"WHEN OTHERS" should not be the only exception handler
Code Smell

"INSERT" statements should explicitly list the columns to be set
Code Smell

"%TYPE" and "%ROWTYPE" should not be used in package specification
Code Smell

Functions and procedures should not be too complex
Code Smell

"IF ... ELSEIF" constructs should end with "ELSE" clauses
Code Smell

"WHEN OTHERS" clauses should be used for exception handling
Code Smell

"CREATE_TIMER" should not be used
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

"FORMS_DDL('COMMIT')" and "FORMS_DDL('ROLLBACK')" should not be used

Analyze your code

Bug Critical

`FORMS_DDL` command, like every DDL statements, is performing an implicit COMMIT. It should be used only if there is no pending transaction otherwise this transaction is automatically committed without updating the Form statuses. Also, the potentially acquired locks are lost in case of this implicit COMMIT.

"FORMS_DDL('COMMIT')" and "FORMS_DDL('ROLLBACK')" should be used with care and most of the time, "COMMIT_FORM" or "ROLLBACK_FORM" should be preferred.

Check the Oracle Forms documentation for more details.

Noncompliant Code Example

```
FORMS_DDL('COMMIT');
```

Compliant Solution

```
COMMIT_FORM;
```

Available In:

sonarlint sonarcloud sonarqube Developer Edition