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

"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

"RESULT_CACHE" should not be used
Code Smell

Columns should be aliased
Code Smell

Track parsing failures
Code Smell

Files should not be too complex
Code Smell

Function and procedure parameters should comply with a naming convention
Code Smell

Magic literals should not be used

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

Analyze your code

Code Smell Major lock-in sql

Be careful about your use of Oracle-specific data types like ROWID and UROWID. They might offer a slight improvement in performance over other means of identifying a single row (primary key or unique index value), but that is by no means guaranteed.

On the other hand, the use of ROWID or UROWID means that your SQL statement will not be portable to other SQL databases. Further, many developers are not familiar with these data types, which can make the code harder to maintain.

Noncompliant Code Example

```
SET SERVEROUTPUT ON

DECLARE
  id rowid; -- Non-Compliant
  universeId urowid; -- Non-Compliant
BEGIN
  SELECT rowid INTO id FROM DUAL;
  SELECT rowid INTO universeId FROM DUAL;

  DBMS_OUTPUT.PUT_LINE('id = ' || id);
  DBMS_OUTPUT.PUT_LINE('universe id = ' || universeId);
END;
/
```

Available In:
sonarlint sonarcloud sonarqube Developer Edition