


















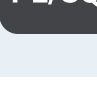














-  Secrets
-  ABAP
-  Apex
-  C
-  C++
-  CloudFormation
-  COBOL
-  C#
-  CSS
-  Flex
-  Go
-  HTML
-  Java
-  JavaScript
-  Kotlin
-  Kubernetes
-  Objective C
-  PHP
-  PL/I
-  **PL/SQL**
-  Python
-  RPG
-  Ruby
-  Scala
-  Swift
-  Terraform
-  Text
-  TypeScript
-  T-SQL
-  VB.NET
-  VB6
-  XML

# 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...



The "result\_cache" hint should be avoided

 Bug

"GOTO" statements should not be used

 Code Smell

"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

The "result\_cache" hint should be avoided

Analyze your code

 Bug

 Major



The `result_cache` Oracle hint can vastly improve performance, but it comes at the cost of extra memory consumption, so one should double-check that the gain in performance is significant, and avoid overusing this feature in general.

### Noncompliant Code Example

```
SELECT /*+ result_cache */ * FROM DUAL;  -- Noncompliant
```

Available In:

sonarlint



| sonarcloud



| sonarqube



Developer Edition