

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...

Statements should be on separate lines

 Code Smell

The "RELIES_ON" clause should not be used


[Analyze your code](#)

<p>"WHEN" clauses should not have too many lines</p> <p>Code Smell</p>	<p>Code Smell</p> <p>Minor</p> <p>obsolete</p>	
--	--	--


<p>Magic numbers should not be used</p>	<p>Since Oracle 11.2, RELIES_ON has been deprecated because the dependencies of result cache-enabled functions are automatically computed.</p>
--	--

 Code Smell

Files should not have too many lines of code	<div data-bbox="1118 1049 1549 1051">Noncompliant Code Example</div> <div data-bbox="1118 1051 2454 1054"></div>
--	--

 Code Smell	<pre>CREATE OR REPLACE FUNCTION foo RETURN PLS_INTEGER RESULT_CACHE RELIES_ON(DUAL) AS -- Noncompliant BEGIN</pre>
--	--

<p>Lines should not be too long</p>	<pre>RETURN 0; END; /</pre>
-------------------------------------	-------------------------------

 Code Smell

DROP FUNCTION foo:

Explicitly opened cursors should be closed	
--	--

Bug	Compliant Solution
<p>1. Unnecessary Complexity: The code uses a <code>HashMap</code> to store the results of the recursive calls, which is not necessary for this problem. A simple array or list can be used to store the results.</p> <p>2. Redundant Recursion: The code uses a recursive function to calculate the number of ways to reach the target sum, which is inefficient. A dynamic programming approach can be used to calculate the number of ways to reach the target sum.</p> <p>3. Unnecessary Recursion: The code uses a recursive function to calculate the number of ways to reach the target sum, which is inefficient. A dynamic programming approach can be used to calculate the number of ways to reach the target sum.</p>	<p>1. Dynamic Programming: The code uses a dynamic programming approach to calculate the number of ways to reach the target sum. It uses a <code>dp</code> array to store the results of the recursive calls, which is more efficient than using a <code>HashMap</code>.</p> <p>2. Iterative Solution: The code uses an iterative solution to calculate the number of ways to reach the target sum. It uses a <code>dp</code> array to store the results of the recursive calls, which is more efficient than using a <code>HashMap</code>.</p> <p>3. Compliant Solution: The code uses a compliant solution to calculate the number of ways to reach the target sum. It uses a <code>dp</code> array to store the results of the recursive calls, which is more efficient than using a <code>HashMap</code>.</p>

Identifiers should be written in lower case	<pre>CREATE OR REPLACE FUNCTION foo RETURN PLS_INTEGER RESULT_CACHE AS</pre>
---	--

Code Smell	<pre>BEGIN RETURN 0;</pre>
--	--------------------------------

<p>"PLS_INTEGER" types should be used</p>	<pre>END; /</pre>
---	-----------------------


Code Smell	DROP FUNCTION foo;
------------	--------------------

Reserved words should be written in upper case	Available In:
--	---------------

 Code Smell

 |
 |
 Developer Edition

Parameter "IN" mode should be specified explicitly	
--	--

 Code Smell

© 2008-2022 SonarSource S.A., Switzerland. All content is copyright protected. SONAR, SONARSOURCE, SONARLINT, SONARQUBE and SONARCLOUD are trademarks of SonarSource S.A. All other trademarks and copyrights are the property of their respective owners. All rights are reserved.

<p>Lines in a multiline comment should start with "*" </p>	<p> All trademarks and copyrights are the property of their respective owners. All rights are expressly reserved. Privacy Policy </p>
--	--


CASE should be used for sequences of simple tests	
---	--


<div>  Code Smell </div>	
---	--


SQL tables should be joined with the "JOIN" keyword


The "RELIES_ON" clause should not be used

Analyze your code

 Code Smell

 Minor



 obsolete

Since Oracle 11.2, RELIES_ON has been deprecated because the dependencies of result cache-enabled functions are automatically computed.

Noncompliant Code Example

```
CREATE OR REPLACE FUNCTION foo RETURN PLS_INTEGER RESULT_CACHE RELIES_ON(DUAL) AS -- Noncompliant
BEGIN
    RETURN 0;
END;
/


DROP FUNCTION foo;
```


Compliant Solution


```
CREATE OR REPLACE FUNCTION foo RETURN PLS_INTEGER RESULT_CACHE AS
BEGIN
    RETURN 0;
END;
/

DROP FUNCTION foo;
```

Available In:

 sonarlint

 sonarcloud

 sonarqube

Developer Edition