

PL/SQL static code analysis

Unique rules to find Bugs, Vulnerabilities, Security Hotspots, and Code Smells in your PL/SQL code

1.	Inserts should include values for non-null columns Bug
2.	Variables should be declared only once in a scope Bug
3.	Size should be specified for string variables Bug
4.	Predefined exceptions should not be overridden Bug
5.	"NOT NULL" variables should be initialized Bug
6.	"NCHAR" and "NVARCHAR2" size should not be specified in bytes Bug
7.	"END LOOP" should be followed by a semicolon Bug
8.	Constraints should not be applied to types that cannot be constrained Bug
9.	Improper constraint forms should not be used Bug
10.	Scale should not be specified for float types Bug
11.	Constant declarations should contain initialization assignments Bug
12.	"RAISE_APPLICATION_ERROR" should only be used with error codes from -20,000 to -20,999 Bug
13.	"FETCH ... BULK COLLECT INTO" should not be used without a "LIMIT" clause Bug
14.	Anchored types should not be constrained Bug
15.	"DELETE" and "UPDATE" statements should contain "WHERE" clauses Bug

16.	"FORALL" statements should use the "SAVE EXCEPTIONS" clause Bug
17.	Pipelined functions should have at least one "PIPE ROW" statement and not return an expression (PLS-00633) Bug
18.	GOTO should not be used to jump backwards Code Smell
19.	Quoted identifiers should not be used Code Smell
20.	Loop start and end labels should match Code Smell
21.	Block start and end labels should match Code Smell
22.	Cipher algorithms should be robust Vulnerability
23.	"COMMIT" should not be used inside a loop Bug
24.	Individual "WHERE" clause conditions should not be unconditionally true or false Bug
25.	Nullable subqueries should not be used in "NOT IN" conditions Bug
26.	"COMMIT" and "ROLLBACK" should not be called from non-autonomous transaction triggers Bug
27.	Positional and named arguments should not be mixed in invocations Bug
28.	Functions should end with "RETURN" statements Bug
29.	Collections should not be iterated in "FOR" loops Bug
30.	Using weak hashing algorithms is security-sensitive Security Hotspot
31.	Dynamically executing code is security-sensitive Security Hotspot
32.	

	SQL "JOIN" conditions should involve all joined tables Code Smell
33.	"SELECT" statements used as argument of "EXISTS" statements should be selective Code Smell
34.	"LIKE" clauses should not be used without wildcards Code Smell
35.	Weak "REF CURSOR" types should not be used Code Smell
36.	Whitespace and control characters in string literals should be explicit Code Smell
37.	Blocks containing "EXECUTE IMMEDIATE" should trap all exceptions Code Smell
38.	"EXCEPTION_INIT -20,NNN" calls should be centralized Code Smell
39.	"CREATE OR REPLACE" should be used instead of "CREATE" Code Smell
40.	Block labels should appear on the same lines as "END" Code Smell
41.	"LOOP ... END LOOP;" constructs should be avoided Code Smell
42.	"IF" statements should not be nested too deeply Code Smell
43.	"CASE" expressions should end with "ELSE" clauses Code Smell
44.	String literals should not be duplicated Code Smell
45.	Constant names should comply with a naming convention Code Smell
46.	Output parameters should be assigned Bug
47.	"ROWNUM" should not be used at the same query level as "ORDER BY" Bug
48.	All branches in a conditional structure should not have exactly the same implementation Bug
49.	

	Strings should only be moved to variables or columns which are large enough to hold them Bug
50.	"WHERE" clause conditions should not be contradictory Bug
51.	Unary prefix operators should not be repeated Bug
52.	DML events clauses should not include multiple "OF" clauses Bug
53.	"PACKAGE BODY" initialization sections should not contain "RETURN" statements Bug
54.	"NULL" should not be compared directly Bug
55.	"MLSLABEL" should not be used Bug
56.	"VARCHAR2" and "NVARCHAR2" should be used Bug
57.	Related "IF/ELSIF" statements and "WHEN" clauses in a "CASE" should not have the same condition Bug
58.	Identical expressions should not be used on both sides of a binary operator Bug
59.	All code should be reachable Bug
60.	Loops with at most one iteration should be refactored Bug
61.	Variables and columns should not be self-assigned Bug
62.	"IF" statement conditions should not evaluate unconditionally to "TRUE" or to "FALSE" Bug
63.	The "result_cache" hint should be avoided Bug
64.	"GOTO" statements should not be used Code Smell
65.	"TO_NUMBER" should be used with a format model Code Smell

66.	Labels should not be reused in inner scopes Code Smell
67.	"FUNCTIONS" should not have "OUT" parameters Code Smell
68.	Variables should be nullable Code Smell
69.	"VARCHAR2" should be used Code Smell
70.	Native SQL joins should be used Code Smell
71.	"FORALL" should be used Code Smell
72.	"FETCH ... BULK COLLECT INTO" should be used Code Smell
73.	Column aliases should be defined using "AS" Code Smell
74.	Procedures and functions should be encapsulated in packages Code Smell
75.	Procedures should have parameters Code Smell
76.	"EXECUTE IMMEDIATE" should be used instead of DBMS_SQL procedure calls Code Smell
77.	"NATURAL JOIN" queries should not be used Code Smell
78.	"END" statements of labeled loops should be labeled Code Smell
79.	In labeled loops "EXIT" should exit the label Code Smell
80.	"EXIT WHEN" should be used rather than "IF ... THEN EXIT; END IF;" Code Smell
81.	"FOR" loop end conditions should not be hard-coded Code Smell
82.	Large item lists should not be used with "IN" clauses Code Smell

83.	"GOTO" should not be used within loops Code Smell
84.	"FULL OUTER JOINS" should be used with caution Code Smell
85.	"CASE" should be used rather than "DECODE" Code Smell
86.	"CROSS JOIN" queries should not be used Code Smell
87.	"END" statements of labeled blocks should be labeled Code Smell
88.	Two branches in a conditional structure should not have exactly the same implementation Code Smell
89.	Unused assignments should be removed Code Smell
90.	"LIKE" clauses should not start with wildcard characters Code Smell
91.	Column names should be used in a SQL "ORDER BY" clause Code Smell
92.	Procedures and functions should be documented Code Smell
93.	SQL statements should not join too many tables Code Smell
94.	Function and procedure names should comply with a naming convention Code Smell
95.	Columns to be read with a "SELECT" statement should be clearly defined Code Smell
96.	"CASE" structures should not have too many "WHEN" clauses Code Smell
97.	Deprecated LONG and LONG RAW datatypes should no longer be used Code Smell
98.	Sections of code should not be commented out Code Smell
99.	Unused procedure and function parameters should be removed

	Code Smell
100.	SQL EXISTS subqueries should not be used Code Smell
101.	Track uses of "FIXME" tags Code Smell
102.	Variables should not be shadowed Code Smell
103.	Redundant pairs of parentheses should be removed Code Smell
104.	Functions and procedures should not have too many parameters Code Smell
105.	Collapsible "if" statements should be merged Code Smell
106.	Unused labels should be removed Code Smell
107.	Compound triggers should define at least two triggers Code Smell
108.	"EXIT" should not be used in loops Code Smell
109.	Jump statements should not be redundant Code Smell
110.	"EXCEPTION WHEN ... THEN" clauses should do more than "RAISE" Code Smell
111.	Single line comments should start with "--" Code Smell
112.	An "ORDER BY" direction should be specified explicitly Code Smell
113.	Oracle's join operator (+) should not be used Code Smell
114.	"cursor%NOTFOUND" should be used instead of "NOT cursor%FOUND" Code Smell
115.	Object attributes should comply with a naming convention Code Smell
116.	Record fields should comply with a naming convention

	Code Smell
117.	
	Cursors should follow a naming convention Code Smell
118.	
	Types should follow a naming convention Code Smell
119.	
	Cursor parameters should follow a naming convention Code Smell
120.	
	Exceptions should follow a naming convention Code Smell
121.	
	Exceptions should not be ignored Code Smell
122.	
	Variables should not be initialized with "NULL" Code Smell
123.	
	Boolean checks should not be inverted Code Smell
124.	
	Unused local variables should be removed Code Smell
125.	
	Package names should comply with a naming convention Code Smell
126.	
	Variables should comply with a naming convention Code Smell
127.	
	Return of boolean expressions should not be wrapped into an "if-then-else" statement Code Smell
128.	
	Boolean literals should not be redundant Code Smell
129.	
	Comments should not be nested Code Smell
130.	
	"DBMS_UTILITY.FORMAT_ERROR_STACK" and "FORMAT_ERROR_BACKTRACE" should be used together Code Smell
131.	
	Procedures should not contain "RETURN" statements Code Smell
132.	
	Track uses of "TODO" tags Code Smell
133.	

	Neither DES (Data Encryption Standard) nor DESede (3DES) should be used Vulnerability
134.	"SYNCHRONIZE" should not be used Bug
135.	Global public variables should not be defined Code Smell
136.	A primary key should be specified during table creation Code Smell
137.	Track lack of copyright and license headers Code Smell
138.	SHA-1 and Message-Digest hash algorithms should not be used in secure contexts Vulnerability
139.	Sensitive "SYS" owned functions should not be used Vulnerability
140.	"FORMS_DDL('COMMIT')" and "FORMS_DDL('ROLLBACK')" should not be used Bug
141.	"DBMS_OUTPUT.PUT_LINE" should not be used Code Smell
142.	"WHEN OTHERS" should not be the only exception handler Code Smell
143.	"INSERT" statements should explicitly list the columns to be set Code Smell
144.	"%TYPE" and "%ROWTYPE" should not be used in package specification Code Smell
145.	Functions and procedures should not be too complex Code Smell
146.	"IF ... ELSEIF" constructs should end with "ELSE" clauses Code Smell
147.	"WHEN OTHERS" clauses should be used for exception handling Code Smell
148.	"CREATE_TIMER" should not be used Code Smell
149.	"TO_DATE" and "TO_TIMESTAMP" should be used with a datetime format model Code Smell
150.	

	Features deprecated in Oracle 12 should not be used Code Smell
151.	Tables should be aliased Code Smell
152.	"SIMPLE_INTEGER" should be used instead of "PLS_INTEGER" Code Smell
153.	Queries should not "SELECT" too many columns Code Smell
154.	"ROWID" and "UROWID" data types should not be used Code Smell
155.	"RETURN" should not be used from within a loop Code Smell
156.	"NUMBER" variables should be declared with precision Code Smell
157.	Nested subqueries should be avoided Code Smell
158.	Nested loops should be labeled Code Smell
159.	Nested blocks should be labeled Code Smell
160.	"RESULT_CACHE" should not be used Code Smell
161.	Columns should be aliased Code Smell
162.	Track parsing failures Code Smell
163.	Files should not be too complex Code Smell
164.	Function and procedure parameters should comply with a naming convention Code Smell
165.	Magic literals should not be used Code Smell
166.	"UNION" should be used with caution Code Smell
167.	

	"GROUP BY" should not be used in SQL "SELECT" statements Code Smell
168.	Track breaches of an XPath rule Code Smell
169.	Track uses of "NOSONAR" comments Code Smell
170.	Track comments matching a regular expression Code Smell
171.	Statements should be on separate lines Code Smell
172.	"WHEN" clauses should not have too many lines Code Smell
173.	Magic numbers should not be used Code Smell
174.	Files should not have too many lines of code Code Smell
175.	Lines should not be too long Code Smell
176.	Explicitly opened cursors should be closed Bug
177.	Identifiers should be written in lower case Code Smell
178.	"PLS_INTEGER" types should be used Code Smell
179.	Reserved words should be written in upper case Code Smell
180.	Parameter "IN" mode should be specified explicitly Code Smell
181.	Lines in a multiline comment should start with "*" Code Smell
182.	CASE should be used for sequences of simple tests Code Smell
183.	SQL tables should be joined with the "JOIN" keyword Code Smell
184.	

	Constraint names should comply with a naming convention <u>Code Smell</u>
185.	
	Reserved words should be written in lower case <u>Code Smell</u>
186.	
	Comments should not be located at the end of lines of code <u>Code Smell</u>
187.	
	Lines should not end with trailing whitespaces <u>Code Smell</u>
188.	
	The "RELIES_ON" clause should not be used <u>Code Smell</u>