Search by name...



- Secrets
- **ABAP**
- Apex
- C++
- CloudFormation
- COBOL
- C#
- CSS
- Flex

Go

- Java
- JavaScript

HTML

- Kotlin
- Kubernetes
- Objective C
- PL/I

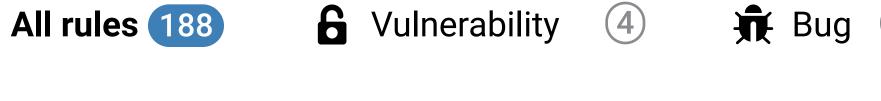
PL/SQL

- 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

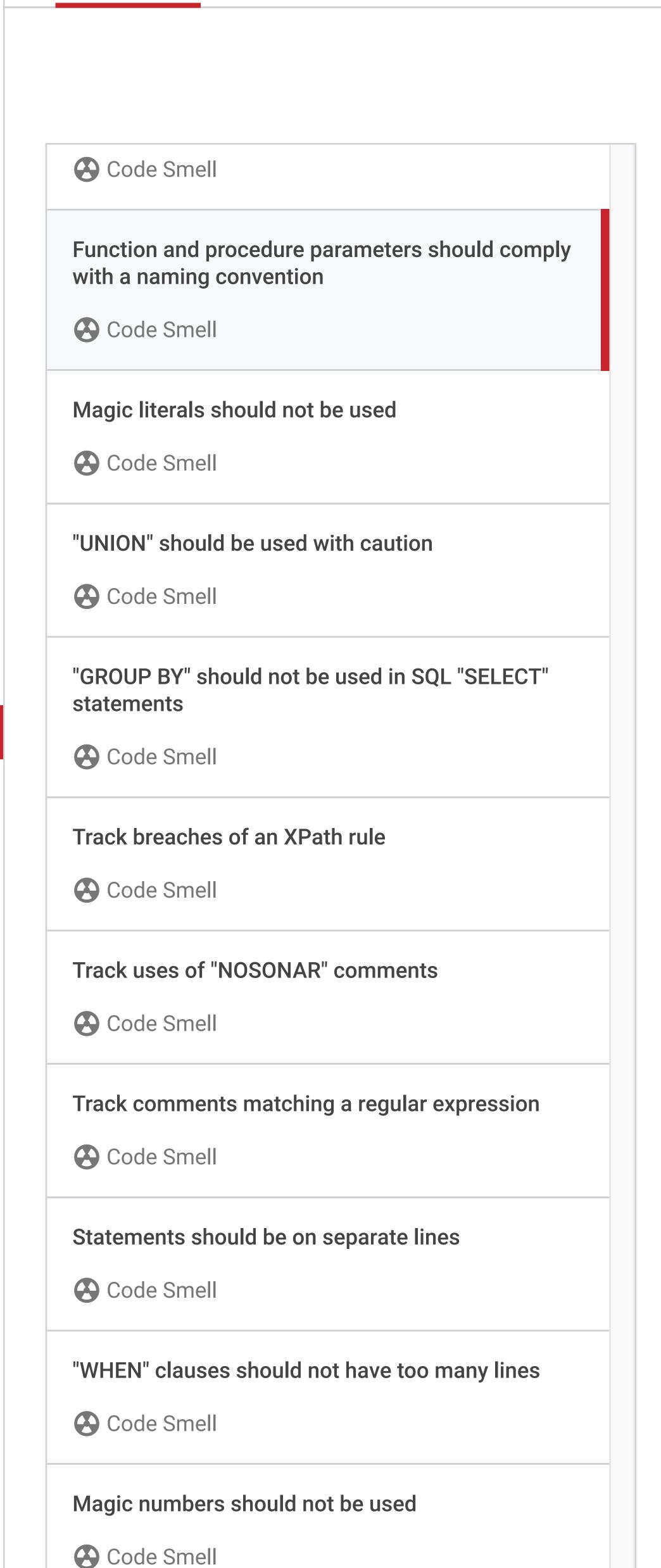






(2)





Files should not have too many lines of code

Code Smell

Code Smell

Lines should not be too long

```
Function and procedure parameters should comply with a naming
                                                                           Analyze your code
convention
Each function and procedure parameter name must match a given regular expression.
Noncompliant Code Example
 DECLARE
   FUNCTION myfunction2(parameter_ PLS_INTEGER) RETURN PLS_INTEGER; -- Noncompliant
   PROCEDURE myprocedure2(parameter_ PLS_INTEGER); -- Noncompliant
   FUNCTION myfunction2(parameter_ PLS_INTEGER) RETURN PLS_INTEGER AS -- Noncompliant
   BEGIN
     RETURN 42;
   END;
   PROCEDURE myprocedure2(parameter_ PLS_INTEGER) AS -- Noncompliant
   BEGIN
     NULL;
   END;
 BEGIN
   NULL;
 END;
```

Tags

Compliant Solution

```
DECLARE
 FUNCTION myfunction1(parameter PLS_INTEGER) RETURN PLS_INTEGER; -- Compliant
  PROCEDURE myprocedure1(parameter PLS_INTEGER); -- Compliant
  FUNCTION myfunction1(parameter PLS_INTEGER) RETURN PLS_INTEGER AS -- Compliant
  BEGIN
   RETURN 42;
  END;
 PROCEDURE myprocedure1(parameter PLS_INTEGER) AS -- Compliant
  BEGIN
   NULL;
  END;
BEGIN
 NULL;
END;
```

Available In:

sonarlint sonarcloud sonarqube Developer Edition