

Improving the SQL code quality with SQLFLUFF rules

Author: Cristina Bocan

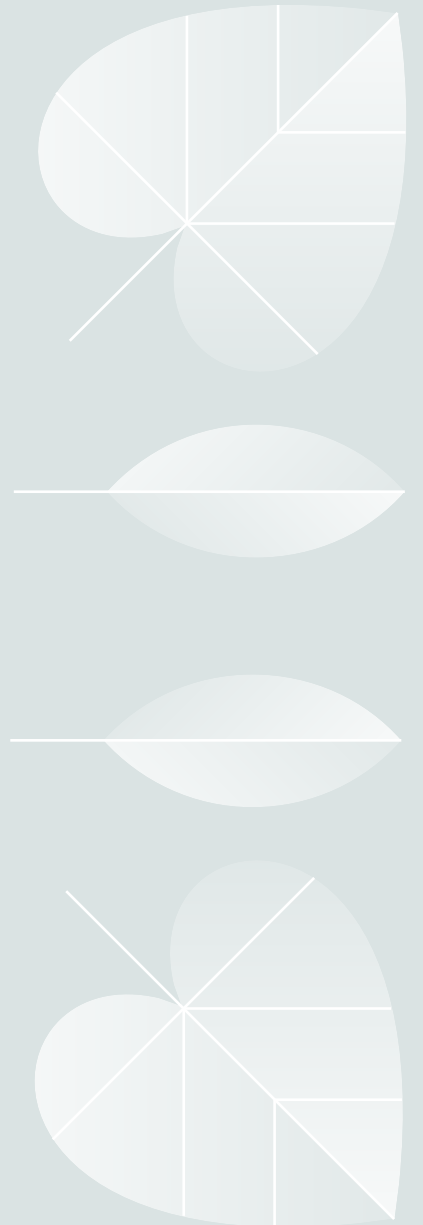


**REVIEWS 10 LINES OF CODE: FINDS
10 ISSUES.**

**REVIEWS 500 LINES OF CODE:
“LOOKS GOOD TO ME”**

What is SQLFLUFF ?

- Sqlfluff is a tool whose purpose is SQL code improvement by reviewing the code and identifying problems
- It is called linter
- It is a linter implemented in Python
- It is open source and dialect-flexible, meaning that it supports multiple sql dialects (e.g. Postgres, DataBricks, Redshift, Db2 etc.).




A decorative graphic on the left side of the slide, consisting of two clusters of stylized, overlapping leaves. The leaves are light gray with white outlines, arranged in a fan-like pattern.

SQLFLUFF purposes

- Code review time reduced: identifies issues related to form and convention, so the code reviewer can focus on the code functionality
- Fixing issues time reduced: has the capability to fix the issues found
- Custom rules: allows the developer to create their own rules according to the business requirements



How SQLFLUFF works?


1. SEGMENT -> SQLFLUFF splits the file into raw segments that are attributed a classification, such as statement segment or newline segment. The segment has 3 components: the type, the anchor, the raw value. Segment example: < KeywordSegment> L :3, P:5 “AS”
 2. RULE -> SQLFLUFF applies its rules against each segment. The rules are grouped in multiple categories, such as “capitalization” or “layout”. Each time a rule is violated SQLFLUFF displays the position of the segment, the violated rule code and description of the rule
- 



SQLFLUFF custom rule implementation


The custom rules should be created as Python packages and transformed into plugins attached to SQLFLUFF.

Steps for creating new rules:

- Create folder containing the setup.py file. This file contains the name for the future plugin and implements the setup() function
 - Create subfolder containing the __init__.py file of the package and the rule.py file
 - Implement the new rule in the rule.py file
 - Run the command `pip install -e "folder_name"`, which detects the setup.py, calls the setup function inside it and creates a plugin out of the rule
- 




SQLFLUFF custom rule implementation details

1. Rule Name Pattern: “Rule_PluginName_LLNN” in case of plugins because the rule inherits from the BaseRule class that inherits from the class RuleMetaclass that contains the protected method called _populate_code_and_description(). This method assigns the code to the rule as being the PluginName + 2 Letters + 2 Digits
 2. Documentation pattern: Presents how an incorrect and correct SQL files should look like
- 



SQLFLUFF rules main elements

1. Name
 2. Groups
 3. Targets_templated
 4. Crawl_behaviour
 5. Is_fix_compatible
 6. Context = the whole file
 7. `_eval()` function -> this is the rule main function because it applies the rule code upon the segments
- 



Thank you