

sqlparse



A Short Introduction
Python User Group Berlin, 2021-04-08, Andi Albrecht

sqlparse - Where It Began

- CrunchyFrog, a homegrown SQL GUI
- Split SQL statements in editor window
- SQL statements are (syntactically) incorrect most of the time
- Highlighting works, so why not use it!
- First sqlparse release: Apr 8, 2009 (Happy Birthday! 🎉)

sqlparse - Main Goals

Dialect agnostic

Should be able to parse different SQL dialects

No strict parsing

Should be able to parse even broken SQLs, no grammar

Splitting

Should be able to identify separate SQL statements in a text

Formatting

Should be able to format SQL statements (to identify errors easily)

sqlparse - Basic Concepts

Tokenize

Tokenizes a text stream based on a set of regular expression.

Base for tokenizing was taken from pygments.

Analyze

Classify tokens

Group them together in a tree structure

Filter Chain

Apply filters for removing white-space, formatting, etc.

sqlparse - Examples

```
>>> import sqlparse

>>> # Split a string containing two SQL statements:
>>> raw = 'select * from foo; select * from bar;'
>>> statements = sqlparse.split(raw)
>>> statements
['select * from foo;', 'select * from bar;']

>>> # Format the first statement and print it out:
>>> first = statements[0]
>>> print(sqlparse.format(first, reindent=True, keyword_case='upper'))
SELECT *
FROM foo;

>>> # Parsing a SQL statement:
>>> parsed = sqlparse.parse('select * from foo')[0]
>>> parsed.tokens
[<DML 'select' at 0x7f22c5e15368>, <Whitespace ' ' at 0x7f22c5e153b0>, <Wildcard '*' ... ]
>>>
```

sqlparse - Current Limitations (and future developments)

- Conflicting keywords in different dialects
- Analyzer needs (heavy) improvements (functions, CTEs, ...)
- Recursion / performance issues for some SQL statements (weird ones)
- No official API for inspecting or manipulating the tree structure
- Still some issues with parsing
- ...

Thanks!

Project

<https://github.com/andialbrecht/sqlparse/>

Contact

albrecht.andi@gmail.com

Questions?