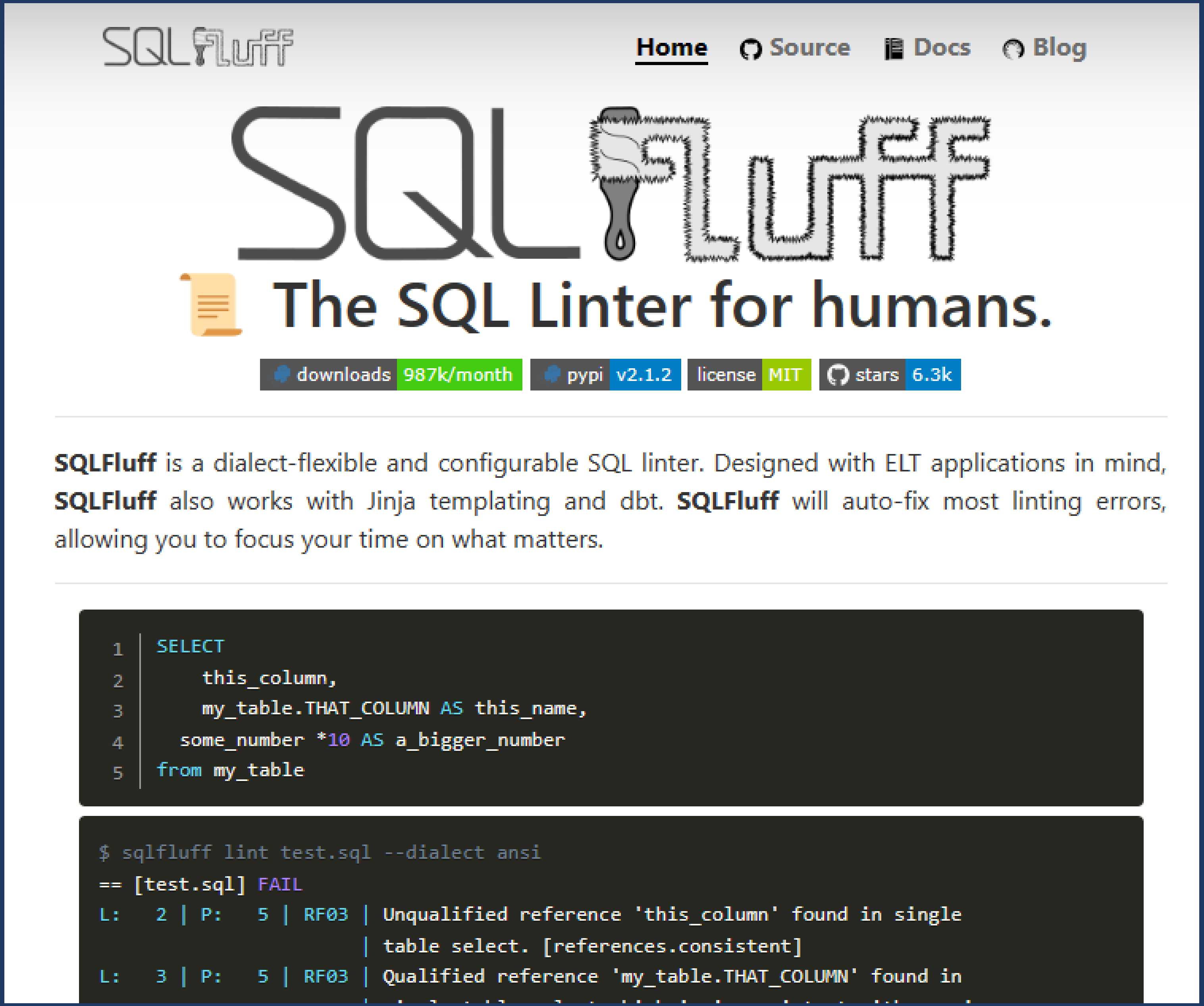
Use SQLFluff in VS Code

Owen Price
@flexyourdata

SQLFluff is a SQL Linter



Home Source Docs Blog

The SQL Linter for humans.

downloads 987k/month | pypi v2.1.2 | license MIT | stars 6.3k

SQLFluff is a dialect-flexible and configurable SQL linter. Designed with ELT applications in mind, **SQLFluff** also works with Jinja templating and dbt. **SQLFluff** will auto-fix most linting errors, allowing you to focus your time on what matters.

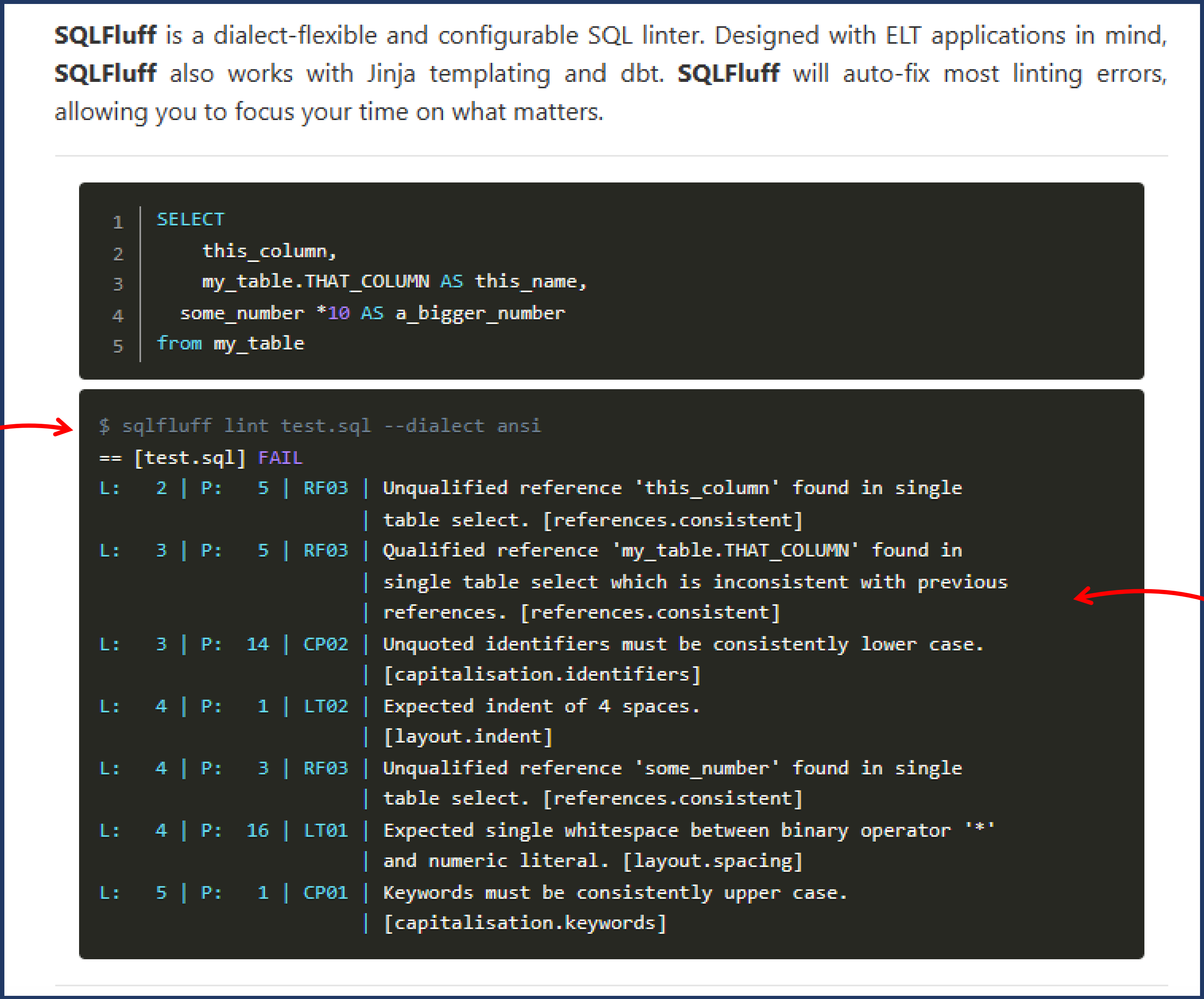
```
SELECT
    this_column,
    my_table.THAT_COLUMN AS this_name,
  some_number *10 AS a_bigger_number
from my_table
```

```
$ sqlfluff lint test.sql --dialect ansi
== [test.sql] FAIL
L:   2 | P:   5 | RF03 | Unqualified reference 'this_column' found in single
                       | table select. [references.consistent]
L:   3 | P:   5 | RF03 | Qualified reference 'my_table.THAT_COLUMN' found in
```

Owen Price
@flexyourdata

It checks your code for *lint*

i.e. fluff / unnecessary fabric which can be removed



SQLFluff is a dialect-flexible and configurable SQL linter. Designed with ELT applications in mind, **SQLFluff** also works with Jinja templating and dbt. **SQLFluff** will auto-fix most linting errors, allowing you to focus your time on what matters.

```sql
SELECT
    this_column,
    my_table.THAT_COLUMN AS this_name,
  some_number *10 AS a_bigger_number
from my_table
```

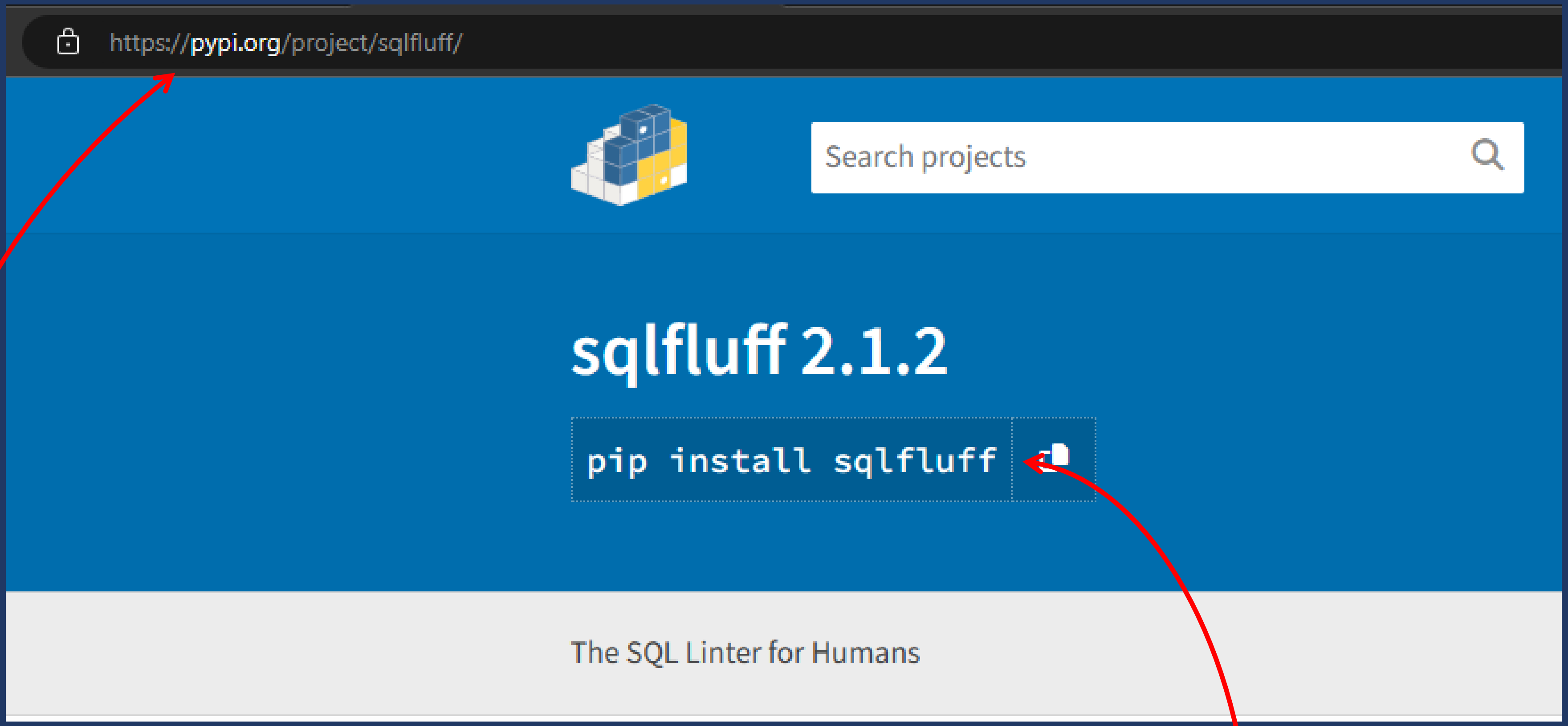
```
$ sqlfluff lint test.sql --dialect ansi
== [test.sql] FAIL
L:   2 | P:    5 | RF03 | Unqualified reference 'this_column' found in single
                        | table select. [references.consistent]
L:   3 | P:    5 | RF03 | Qualified reference 'my_table.THAT_COLUMN' found in
                        | single table select which is inconsistent with previous
                        | references. [references.consistent]
L:   3 | P:   14 | CP02 | Unquoted identifiers must be consistently lower case.
                        | [capitalisation.identifiers]
L:   4 | P:    1 | LT02 | Expected indent of 4 spaces.
                        | [layout.indent]
L:   4 | P:    3 | RF03 | Unqualified reference 'some_number' found in single
                        | table select. [references.consistent]
L:   4 | P:   16 | LT01 | Expected single whitespace between binary operator '*'
                        | and numeric literal. [layout.spacing]
L:   5 | P:    1 | CP01 | Keywords must be consistently upper case.
                        | [capitalisation.keywords]
```

We can use it to check the quality of our SQL

We can define many rules to enforce and check our SQL automatically against each one

Owen Price
@flexyourdata

It’s a Python package


https://pypi.org/project/sqlfluff/

Search projects

sqlfluff 2.1.2

```
pip install sqlfluff
```

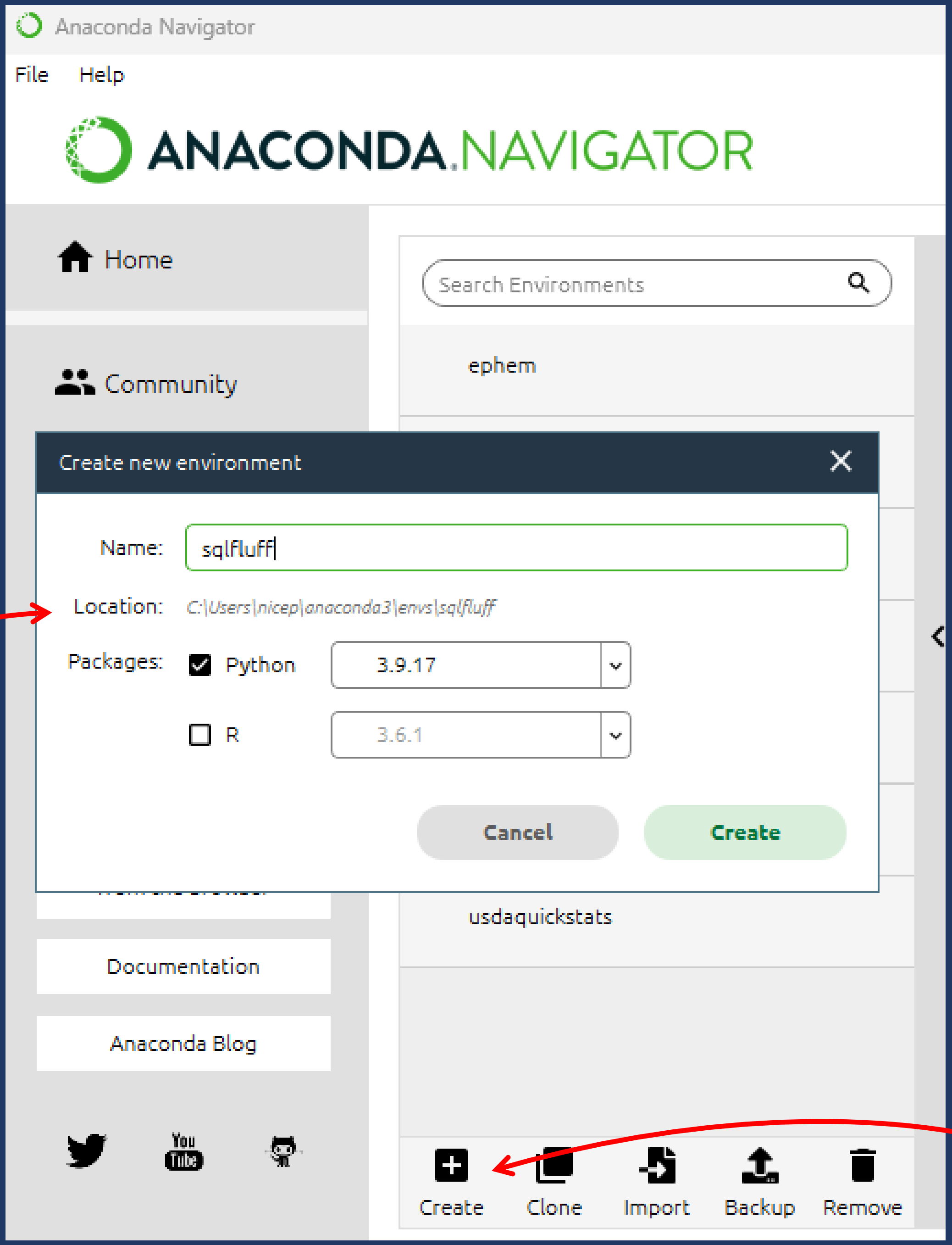
The SQL Linter for Humans

It’s available on PyPI – the official Python Package Index

We can install it into a Python environment using pip

Owen Price
@flexyourdata

For example, create a new environment in Anaconda



Name: sqlfluff

Location: C:\Users\nicep\anaconda3\envs\sqlfluff

Packages: ☑ Python 3.9.17

☐ R 3.6.1

2 Enter a name and select the version of Python to install

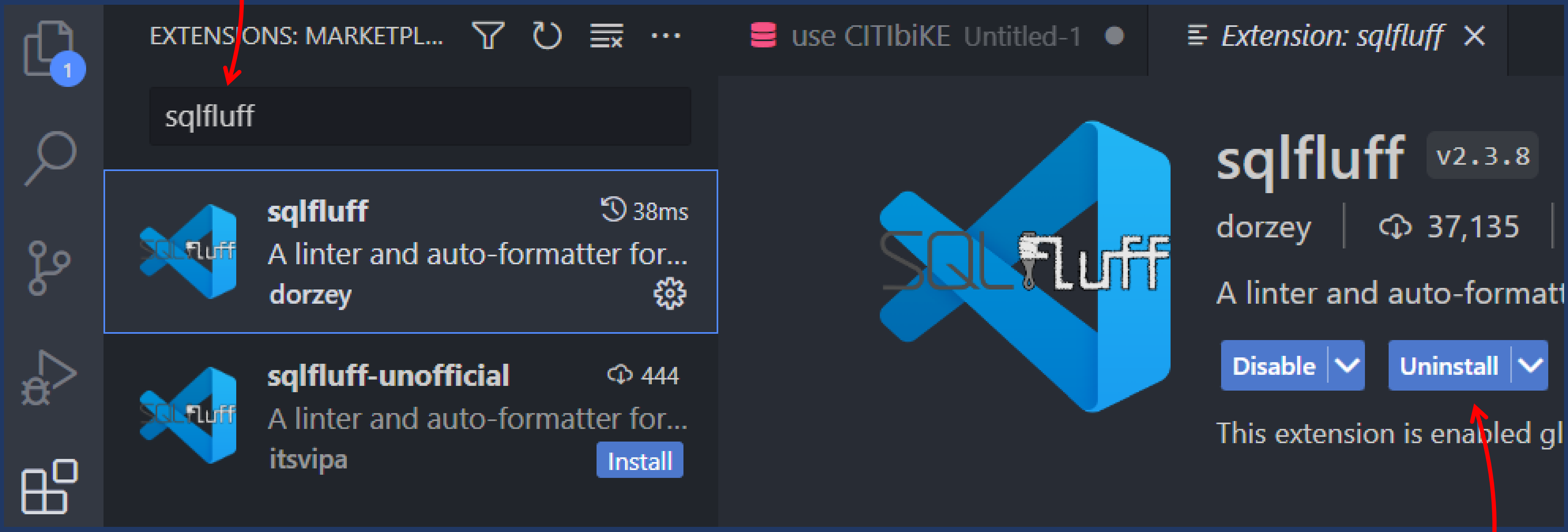
i This is just one way of creating an environment. Other ways are available or if you prefer, you can work without an environment (not recommended)

1 From the Environments tab, hit Create

Owen Price
@flexyourdata

Similarly, install the sqlfluff extension

2 Search for sqlfluff



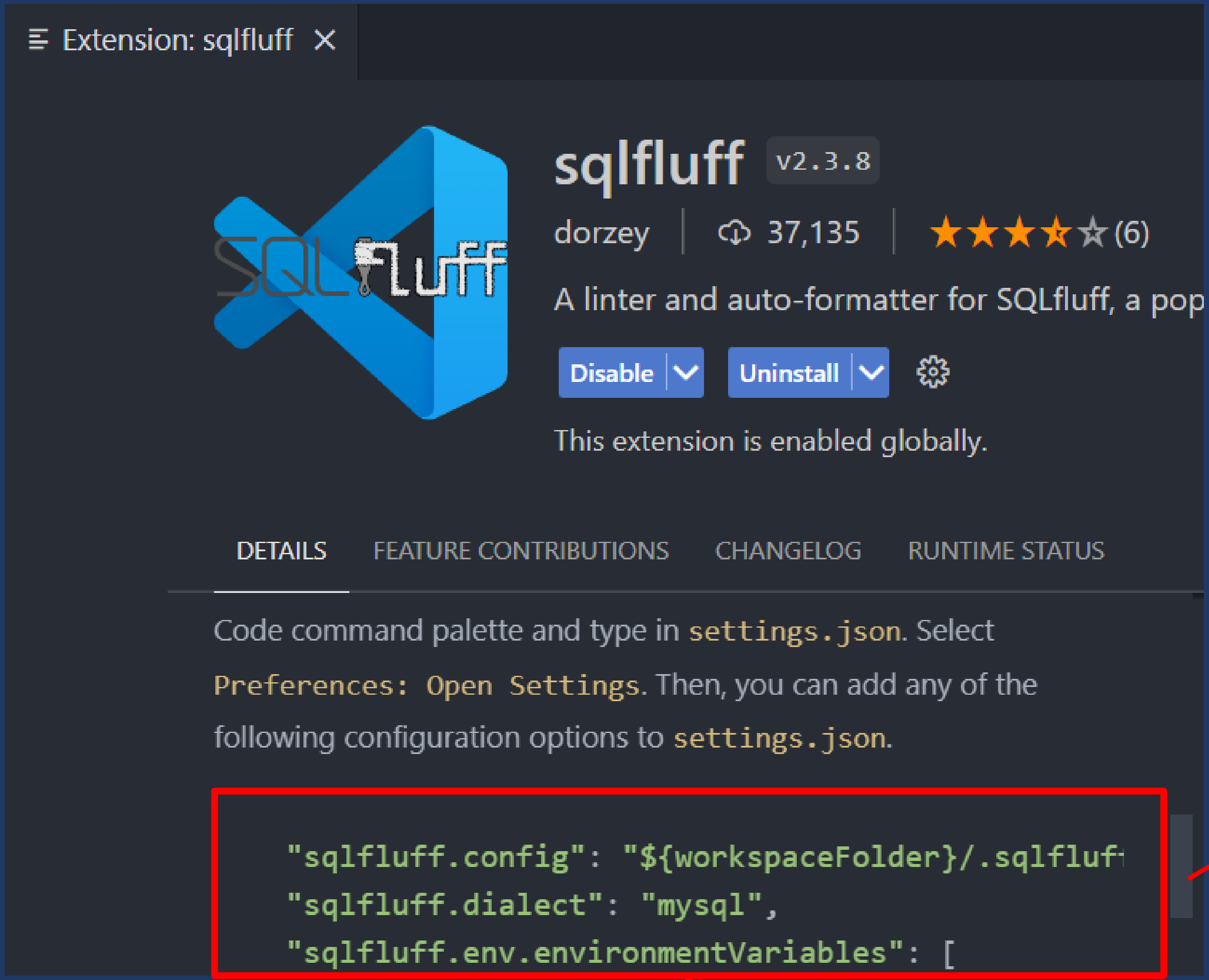
1 Select the extensions tab

3 Hit this button to install the extension (it will say ***Install*** when not installed)

Owen Price
@flexyourdata

Edit *settings.json* in your project folder

Create the file if it doesn't exist



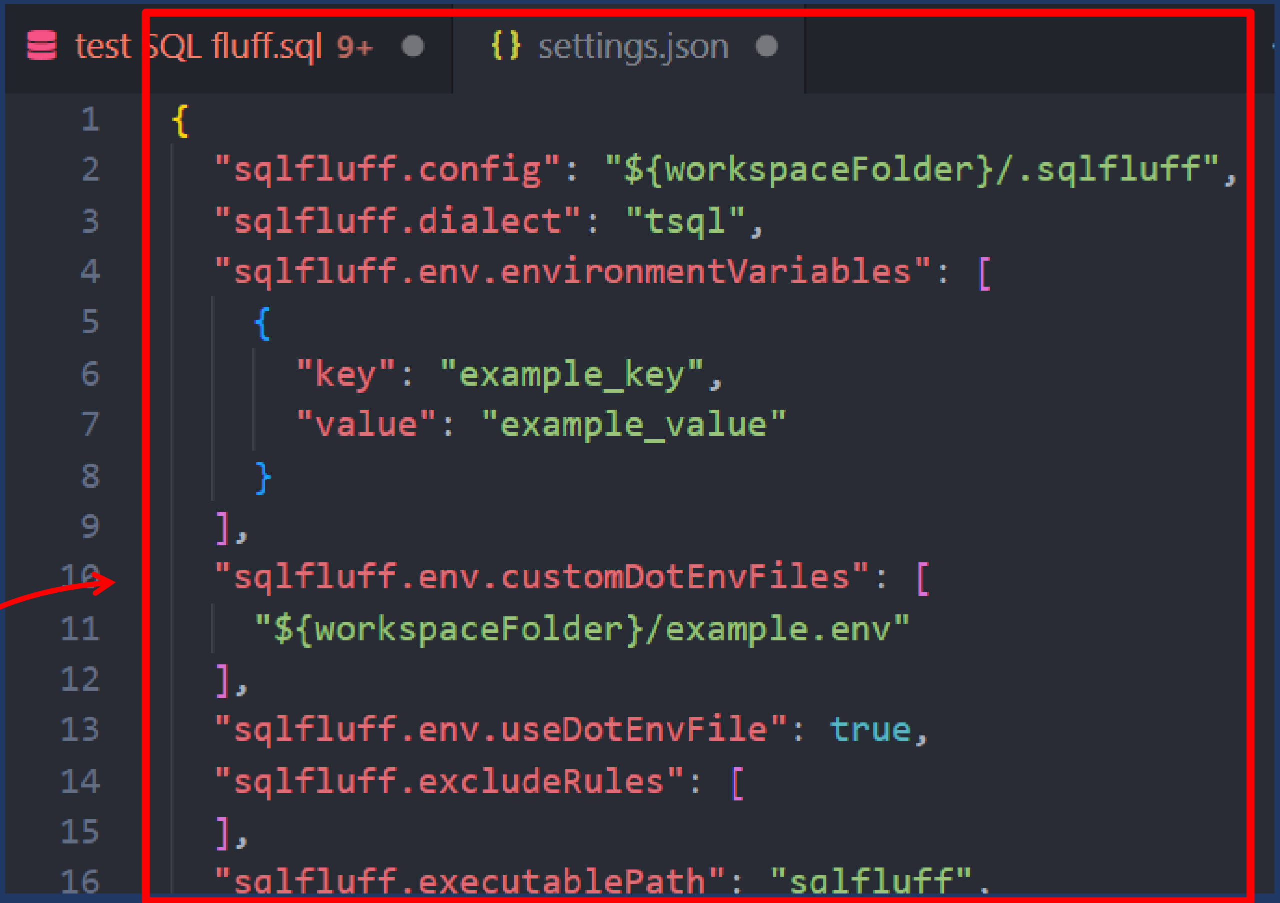
A linter and auto-formatter for SQLfluff, a pop

This extension is enabled globally.

Code command palette and type in `settings.json`. Select `Preferences: Open Settings`. Then, you can add any of the following configuration options to `settings.json`.

```
  "sqlfluff.config": "${workspaceFolder}/.sqlfluf
  "sqlfluff.dialect": "mysql",
  "sqlfluff.env.environmentVariables": [
```

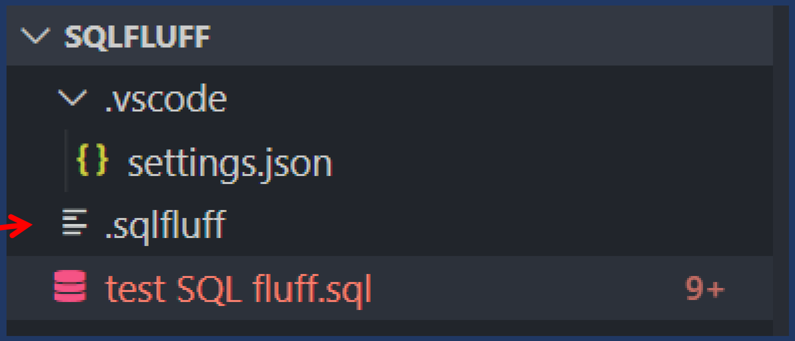
Scroll down on the extension page, then copy the supplied JSON into your settings.json file



```
{
  "sqlfluff.config": "${workspaceFolder}/.sqlfluff",
  "sqlfluff.dialect": "tsql",
  "sqlfluff.env.environmentVariables": [
    {
      "key": "example_key",
      "value": "example_value"
    }
  ],
  "sqlfluff.env.customDotEnvFiles": [
    "${workspaceFolder}/example.env"
  ],
  "sqlfluff.env.useDotEnvFile": true,
  "sqlfluff.excludeRules": [
  ],
  "sqlfluff.executablePath": "sqlfluff",
```

Owen Price
@flexyourdata

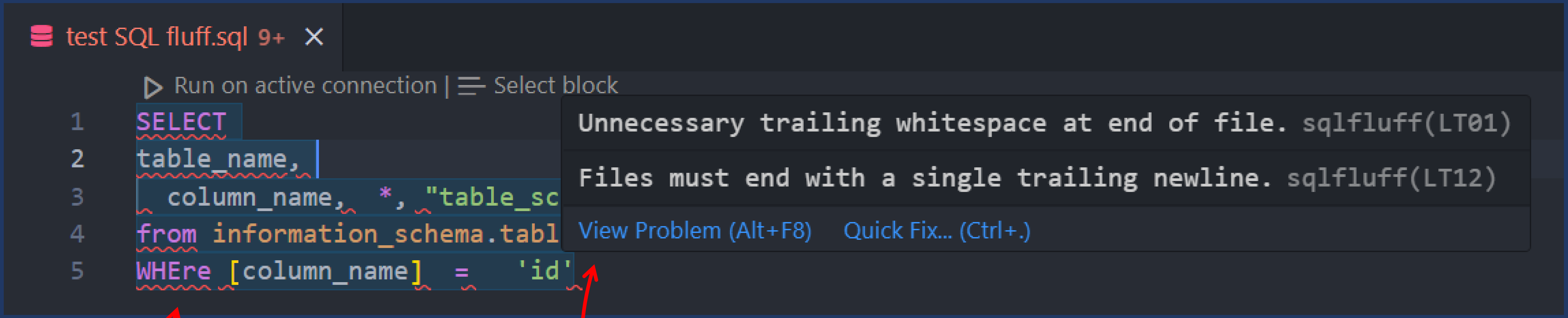
Create an empty text file called *.sqlfluff* in your project folder


SQLFLUFF
.vscode
settings.json
.sqlfluff
test SQL fluff.sql 9+

You can leave the file empty for now

Owen Price
@flexyourdata

Create a new .sql file and type some ugly SQL



```
SELECT
table_name,
  column_name,  *, "table_sc
from information_schema.tabl
WHEre [column_name]  =   'id'
```

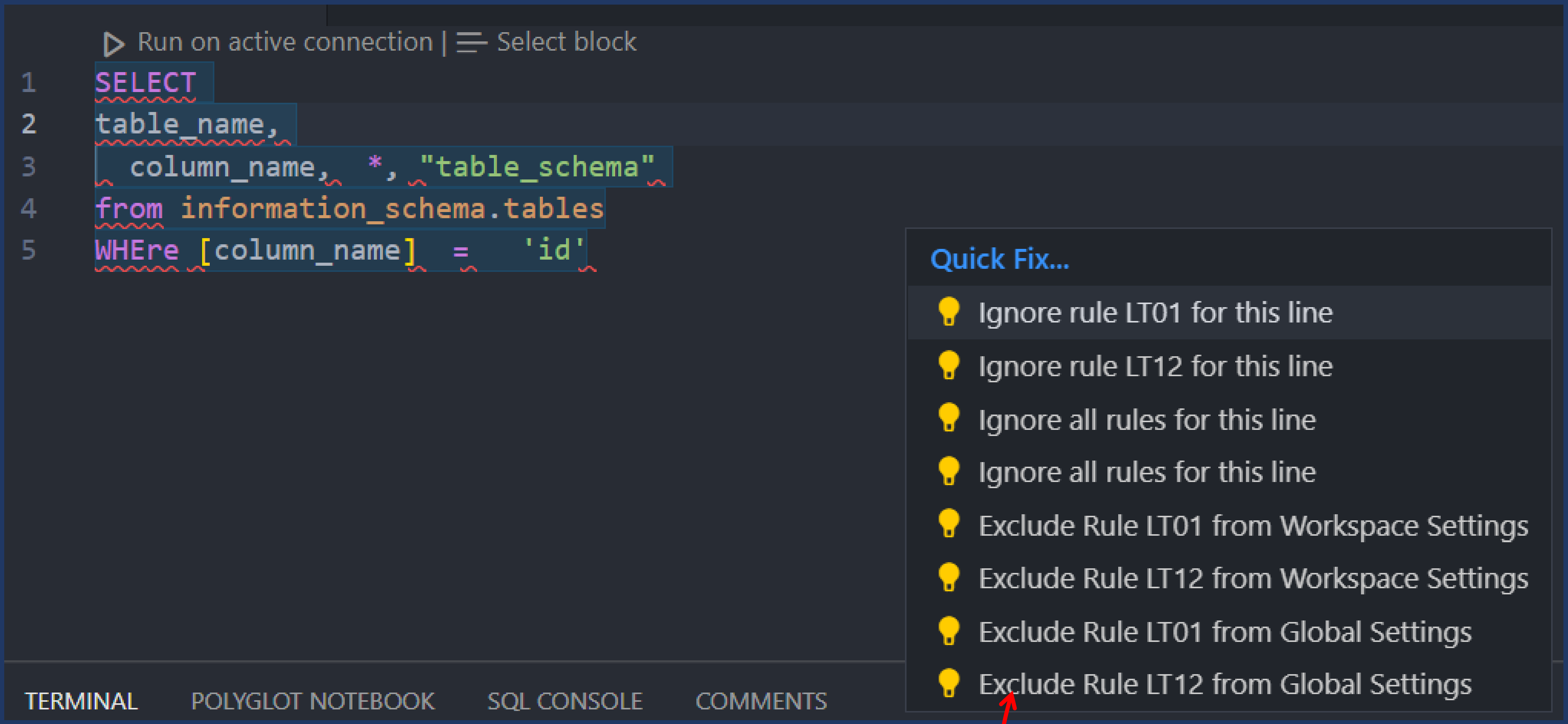
```
Unnecessary trailing whitespace at end of file. sqlfluff(LT01)
Files must end with a single trailing newline. sqlfluff(LT12)
```

SQL which violates the rules will be underlined

Hovering over an underline will show a pop-up describing which rule(s) have been violated

Owen Price
@flexyourdata

Optionally, hover over a rule and click "Quick Fix"



```
SELECT
table_name,
  column_name,  *, "table_schema"
from information_schema.tables
WHEre [column_name]  =   'id'
```

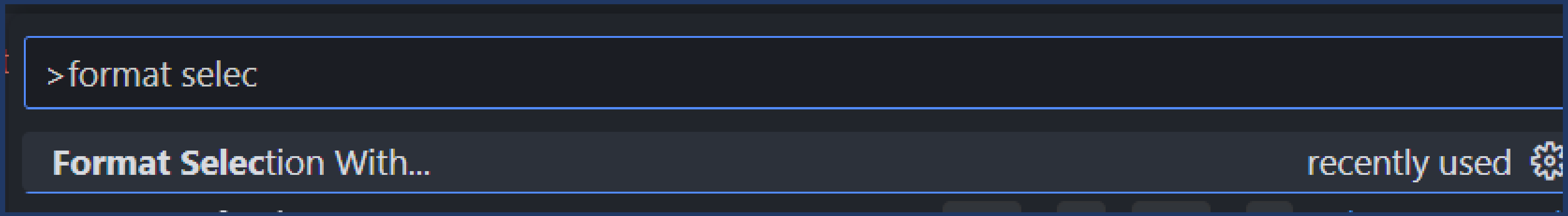
Run on active connection | Select block
Quick Fix...
Ignore rule LT01 for this line
Ignore rule LT12 for this line
Ignore all rules for this line
Ignore all rules for this line
Exclude Rule LT01 from Workspace Settings
Exclude Rule LT12 from Workspace Settings
Exclude Rule LT01 from Global Settings
Exclude Rule LT12 from Global Settings
TERMINAL
POLYGLOT NOTEBOOK
SQL CONSOLE
COMMENTS

Optionally, select an option to remediate the problem.

Owen Price
@flexyourdata

Alternatively, we can auto-fix the problems - 1

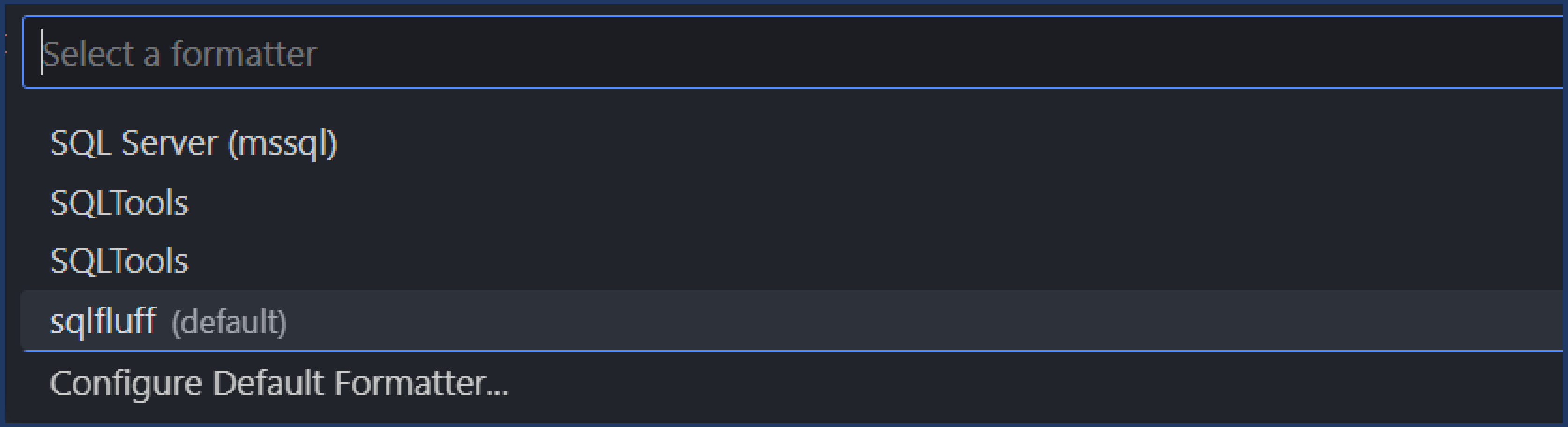
Ctrl+Shift+P and search for "Format selection with…"



>format selec

Format Selection With... recently used

Ensure sqlfluff is configured as the default formatter



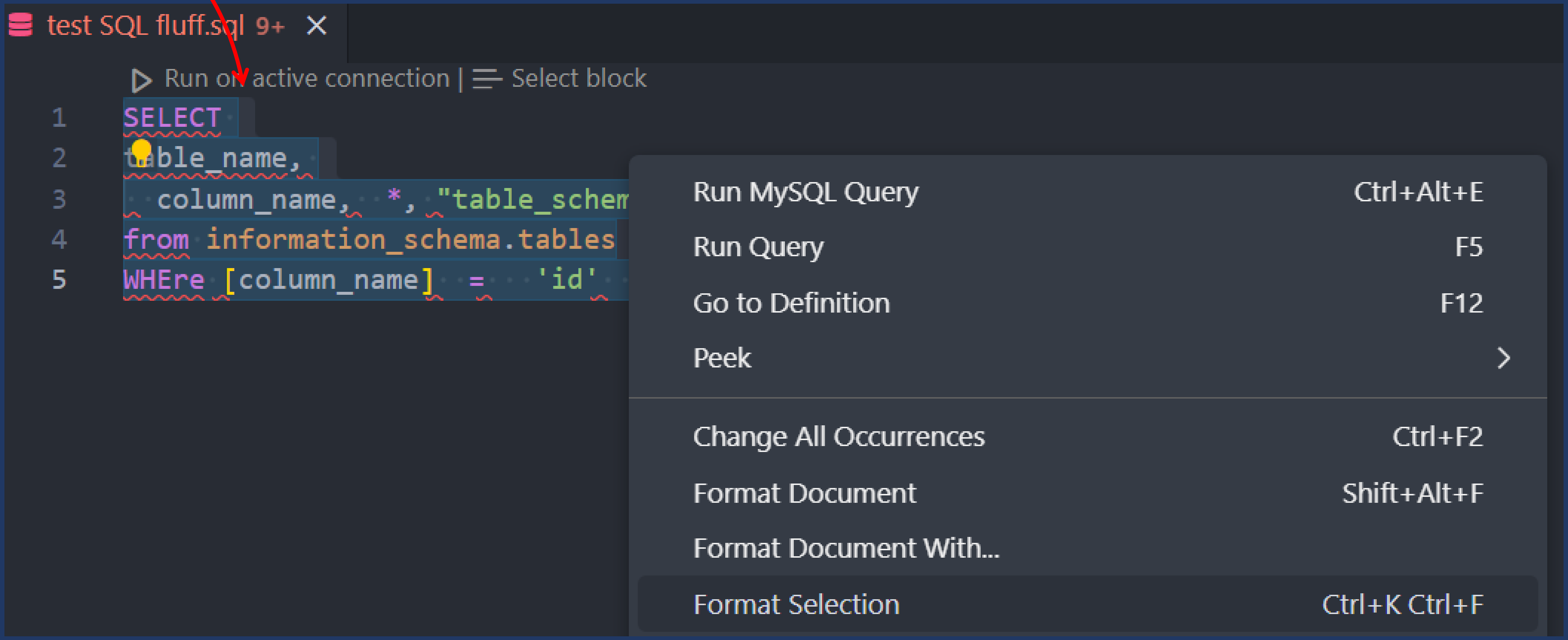
Select a formatter

SQL Server (mssql)
SQLTools
SQLTools
sqlfluff (default)
Configure Default Formatter...

Owen Price
@flexyourdata

Alternatively, we can auto-fix the problems - 2

Select the code you want to fix



test SQL fluff.sql 9+

Run on active connection | Select block

```
SELECT
table_name,
  column_name,  *, "table_schem
from information_schema.tables
WHEre [column_name]  =   'id'
```

Run MySQL Query Ctrl+Alt+E
Run Query F5
Go to Definition F12
Peek
Change All Occurrences Ctrl+F2
Format Document Shift+Alt+F
Format Document With...
Format Selection Ctrl+K Ctrl+F

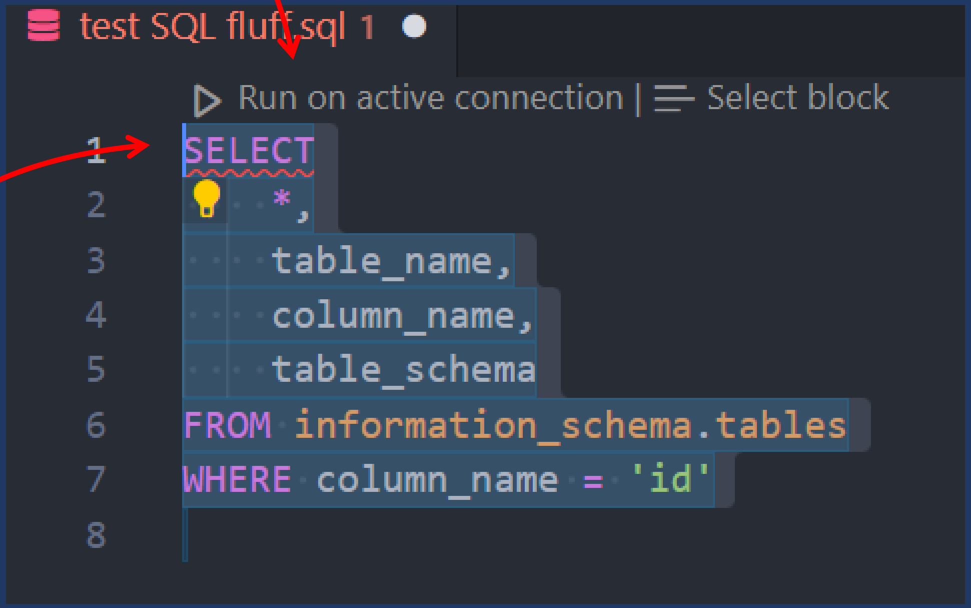
Right-click and use “Format Selection” (alternatively, Ctrl+K Ctrl+F)

Owen Price
@flexyourdata

Alternatively, we can auto-fix the problems - 3

After formatting, the code is reformatted and *most* rules are fixed

Some rules such as use of asterisk in SELECT can't be fixed automatically



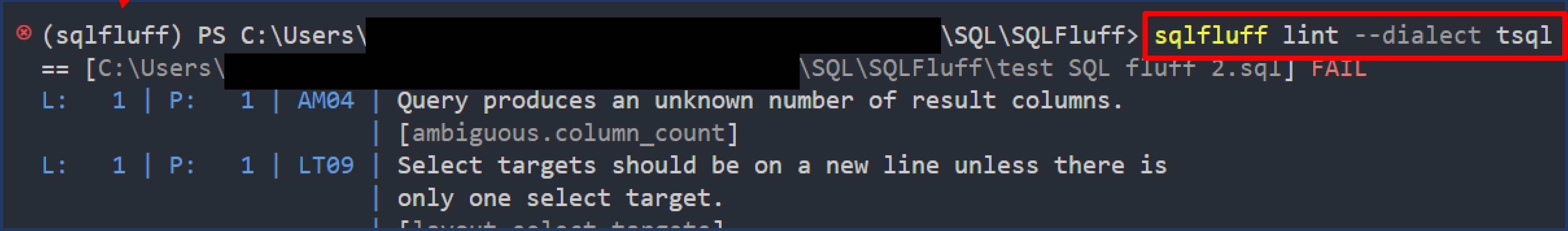
```sql
SELECT
    *,
    table_name,
    column_name,
    table_schema
FROM information_schema.tables
WHERE column_name = 'id'
```

Owen Price
@flexyourdata

You can check your entire project using the terminal

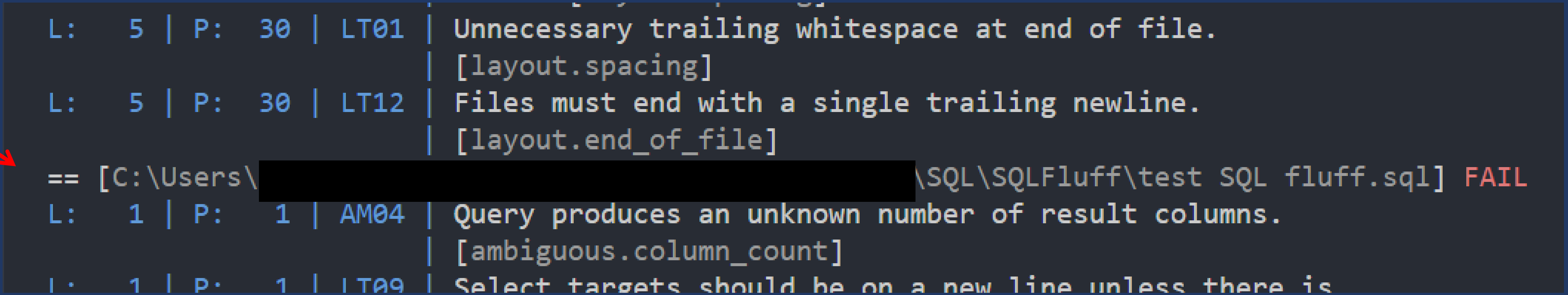
Ensure your env. is activated. If not, use ***conda activate envname***

Use the ***sqlfluff lint*** command to check the project



```
⊗ (sqlfluff) PS C:\Users\[redacted]\SQL\SQLFluff> sqlfluff lint --dialect tsql
  == [C:\Users\[redacted]\SQL\SQLFluff\test SQL fluff 2.sql] FAIL
  L:   1 | P:    1 | AM04 | Query produces an unknown number of result columns.
                          | [ambiguous.column_count]
  L:   1 | P:    1 | LT09 | Select targets should be on a new line unless there is
                          | only one select target.
```

Each file is checked against all rules in turn

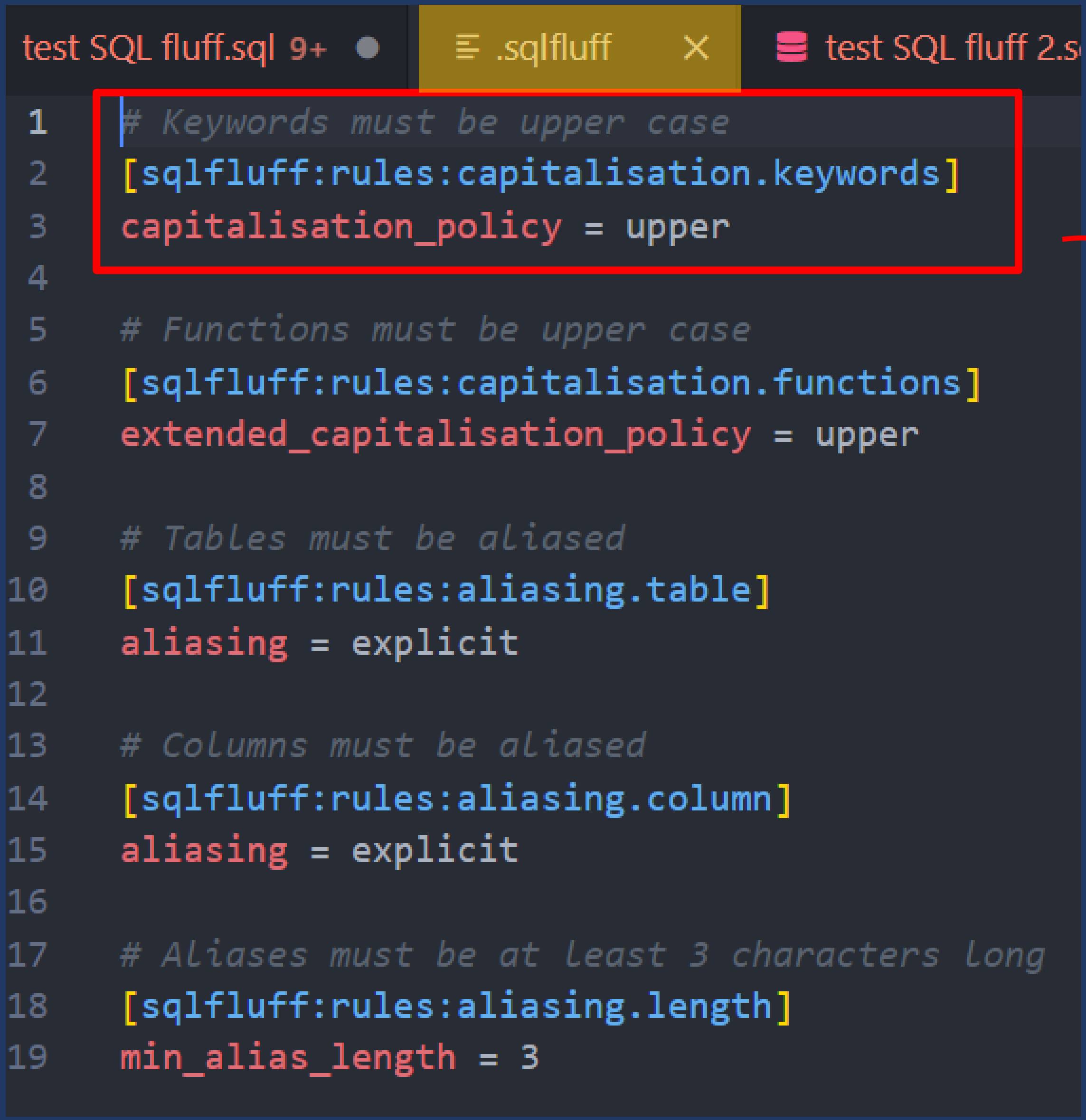


```
L:   5 | P:   30 | LT01 | Unnecessary trailing whitespace at end of file.
                        | [layout.spacing]
L:   5 | P:   30 | LT12 | Files must end with a single trailing newline.
                        | [layout.end_of_file]
== [C:\Users\[redacted]\SQL\SQLFluff\test SQL fluff.sql] FAIL
L:   1 | P:    1 | AM04 | Query produces an unknown number of result columns.
                        | [ambiguous.column_count]
L:   1 | P:    1 | LT09 | Select targets should be on a new line unless there is
```

sqlfluff fix will attempt to fix all problems in the proejct

Owen Price
@flexyourdata

Use .sqlfluff to configure the rules for your project



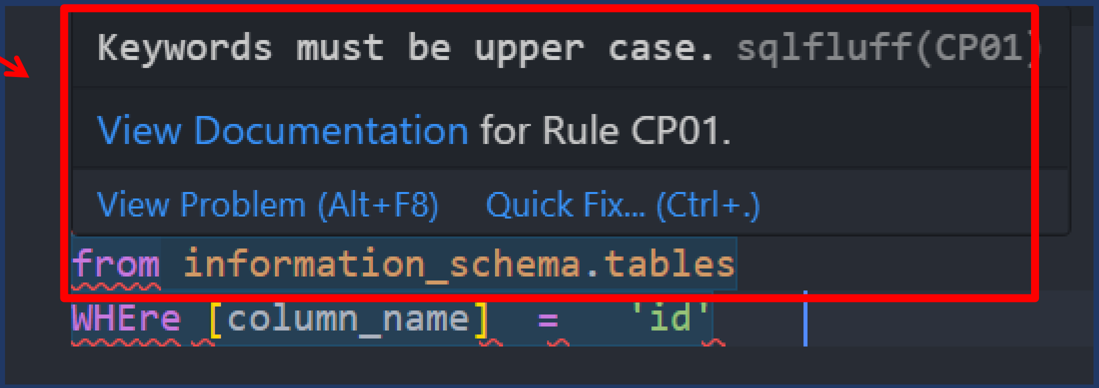
```
# Keywords must be upper case
[sqlfluff:rules:capitalisation.keywords]
capitalisation_policy = upper

# Functions must be upper case
[sqlfluff:rules:capitalisation.functions]
extended_capitalisation_policy = upper

# Tables must be aliased
[sqlfluff:rules:aliasing.table]
aliasing = explicit

# Columns must be aliased
[sqlfluff:rules:aliasing.column]
aliasing = explicit

# Aliases must be at least 3 characters long
[sqlfluff:rules:aliasing.length]
min_alias_length = 3
```



Keywords must be upper case. sqlfluff(CP01)

View Documentation for Rule CP01.

```
from information_schema.tables
WHEre [column_name] =  'id'
```

https://docs.sqlfluff.com/en/stable/configuration.html

Owen Price
@flexyourdata

Takeaways:

1. SQLFluff is used to enforce SQL style and formatting rules
2. We can lint a single query or an entire project (or anywhere between)
3. We can configure which rules are used on a per-project basis

Hi!
I’m Owen and I want to help you flex your data! 💪

I have 20 years’ experience solving tricky data problems.

I have C-suite experience leading global data and programming teams for data products and a host of data-focused technologies in my toolkit (SQL, Python, Advanced Excel, R, Tableau, Power BI, Athena, Glue, Spark, RDS… to name a few).

Let’s connect!

LinkedIn:	linkedin.com/in/owenhprice
YouTube:	@flexyourdata
Instagram:	@flexyourdata
Blog:	flexyourdata.com