# SQLFluff SQL linter and formatter

## CONCEPTS

Always set a dialect (--dialect or in .sqlfluff); without one SQLFluff cannot lint most real SQL.

**Template** jinja / dbt
↓ render
**Lex + parse** per dialect
↓ rules
**Lint** violations
↓ fix
**Fixed SQL** written back

## LINT

`sqlfluff lint <path>`
Lint a file or directory

`sqlfluff lint .`
Lint the current directory

`sqlfluff lint -d <dialect> <path>`
Set the dialect

`sqlfluff lint -r <rules> <path>`
Check only these rules

`sqlfluff lint -e <rules> <path>`
Skip these rules

`sqlfluff lint -f json <path>`
Machine-readable output

`sqlfluff lint --nofail <path>`
Always exit 0

## FIX AND FORMAT

`sqlfluff fix <path>`
Fix violations in place

`sqlfluff format <path>`
Autoformat in place

`sqlfluff fix -r <rules> <path>`
Fix only these rules

## INSPECT

`sqlfluff parse <file>`
Print the parse tree

`sqlfluff render <file>`
Show templated SQL

`sqlfluff rules`
List all rules

`sqlfluff dialects`
List all dialects

## OPTIONS

`--templater / -t <name>`
raw, jinja, python, placeholder

## CONFIG

`[sqlfluff]`
Core settings section

`dialect = <name>`
Default dialect

`exclude_rules = <list>`
Rules to skip

`max_line_length = <n>`
Line length limit (LT05)

`[sqlfluff:rules:<rule>]`
Per-rule settings

`.sqlfluffignore`
Paths to skip

## IGNORING CODE

`-- noqa`
Ignore all rules on this line

`-- noqa: <rules>`
Ignore listed rules on this line

## COMMON RULES

`LT01`
Spacing

`LT02`
Indentation

`LT05`
Line too long

`CP01`
Keyword capitalisation

## INTEGRATION

`pip install sqlfluff`
Install

## RECIPES

**Minimal .sqlfluff**

```
[sqlfluff]
dialect = postgres
exclude_rules = LT05, ST07
max_line_length = 100

[sqlfluff:indentation]
tab_space_size = 4
```