

SQL comment

SQL line comment (--) or block comment (/* */).

```
(-- [^\n]*|\/\/*[\\s\S]*?\\*\/)
```

Token-by-token breakdown

Token	Meaning
(	start of a capturing group
--	literal text "--"
[^\n]*	zero or more: any character except "n"
	OR — try the alternative
\/	literal "/"
*	literal "*"
[\\s\S]*?	zero or more (lazy — as few as possible): any of: whitespace, non-whitespace
*	literal "*"
\/	literal "/"
)	end of group

Quick usage in 7 languages

JavaScript	<pre>const re = /(-- [^\n]* \/\/*[\\s\S]*?*\/)/; re.test(value);</pre>
Python	<pre>import re re.match(r"-- [^\n]* \/\/*[\\s\S]*?*\/", value)</pre>
Java	<pre>Pattern.compile("-- [^\n]* \/\/*[\\s\S]*?*\/").matcher(value).matches();</pre>
C# / .NET	<pre>Regex.IsMatch(value, @"(-- [^\n]* \/\/*[\\s\S]*?*\/)");</pre>
Go	<pre>regexp.MustCompile(`(-- [^\n]* \/\/*[\\s\S]*?\\*\/)`).MatchString(value)</pre>
Ruby	<pre>/(-- [^\n]* \/\/*[\\s\S]*?*\/)/.match?(value)</pre>
PHP	<pre>preg_match('~(-- [^\n]* \/\/*[\\s\S]*?*\/)~', \$value);</pre>

Common pitfalls

- **Not anchored.** Without ^ and \$ this can match a substring anywhere in the input — add anchors if you need the whole value to conform.
- **Escape it correctly per language.** In Java and JavaScript strings each backslash must be doubled (\\d); in Python, Go, and C# use raw/verbatim strings so the backslashes survive.
- **Validate beyond format.** Matching the format doesn't guarantee the value is real. Confirm the sql comment against a source of truth (database, API, or checksum) where it matters.