Cthulhu Filter Specification

Cthulhu is a prefix notation syntax written in JSON for specifying filters on records (usually column-based records).

Purpose

Easily parses into an abstract syntax tree.

Glossary

any - any valid json basic value (number, string)

... - previous item in a list can repeat as many times as necessary

Operators

Name	Spec	Example	SQL Equivalent
eq	`["eq", <column: string>, <value: any>]`	`["eq", "x", 1]`	`x = 1`
gt	`["gt", <column: string>, <value: any>]`	`["gt", "x", 1]`	`x > 1`
gte	`["gte", <column: string>, <value: any>]`	`["gte", "x", 1]`	`x >= 1`
lt	`["lt", <column: string>, <value: any>]`	`["lt", "x", 1]`	`x < 1`

lte	["lte", <column: string>, <value: any>]	["lte", "x", 1]	x <= 1
like	["like", <column: string>, <value: regex>]	["like", "x", ".*foobar"]	x LIKE "%foobar"
between	["between", <column: string>, <lower: any>, <upper: any>]	["between", "x", 1, 2]	x BETWEEN 1 AND 2 x >= 1 AND x <= 2
In	["in", <column: string>, <value: any>, ...]	["in", "x", 1, 2, 3]	x IN (1,2,3)
and	["and", <operator>, ...]	["and", ["gt", "x", 1], ["lt", "x", 2]]	x > 1 AND x < 2

or	`["or", <operator>, ...]`	`["or", ["eq", "x", 1], ["eq", "x", 2]]`	`x = 1 OR x = 2`
not	`["not", <operator>]`	`["not", ["eq", "x", 1]]`	`x != 1`