

given json → check if valid

→ return filter value

## Query Modeler

structure: object for query?

↳ id, dataset.keys()

↳ where → parse into sub items

→ helpers EBNF

↳ option: columns (must)

```
QUERY ::= '{' BODY ' ', ' OPTIONS '}'

// Note: a BODY with no FILTER (i.e. WHERE:{}) matches all entries.
BODY ::= 'WHERE:{' FILTER? '}'

FILTER ::= LOGICCOMPARISON | MCOMPARISON | SCOMPARISON | NEGATION

LOGICCOMPARISON ::= LOGIC ':' [' FILTER_LIST ']'
MCOMPARISON ::= MCOMPARATOR ':' [' mkey ':' number ']'
SCOMPARISON ::= 'IS:{' skey ':' ' ' [*]? inputstring [*]? ' ' }' // Asterisks at the beginning or end of the inputstring should
act as wildcards.
NEGATION ::= 'NOT :{' FILTER '}'

FILTER_LIST ::= '{' FILTER '}' | '{' FILTER ' ', ' FILTER_LIST // comma separated list of filters containing at least one filter
LOGIC ::= 'AND' | 'OR'
MCOMPARATOR ::= 'LT' | 'GT' | 'EQ'

OPTIONS ::= 'OPTIONS:{' COLUMNS '}' | 'OPTIONS:{' COLUMNS ' ', ORDER:' key '}'

COLUMNS ::= 'COLUMNS:[' KEY_LIST ']'

KEY_LIST ::= key | key ' ', ' KEY_LIST // comma separated list of keys containing at least one key

key ::= mkey | skey
mkey ::= ' ' idstring ' ' mfield ' '
skey ::= ' ' idstring ' ' sfield ' '
mfield ::= 'avg' | 'pass' | 'fail' | 'audit' | 'year'
sfield ::= 'dept' | 'id' | 'instructor' | 'title' | 'uuid'
idstring ::= [^_]+ // One or more of any character, except underscore.
inputstring ::= [^']* // Zero or more of any character, except asterisk.
```

## Query Engine

Things to keep in mind:

### WHERE:

1. NOT { NOT { NOT { ... NOT { } } } } or/And { NOT { } } , NOT { } } )

2. And { And { And { } , And { } } }

3. or { or { } , or { } } }

MC [ check if length ≠ 1 x ] && [ check if field is not int x ] && [ mkey not in list x ]  
SC [ check if length ≠ 1 x ] && [ check if field is not string x ] && [ skey not in list x ]

input string validation.

check in not  
if length > 1 then x

check in And  
if length < 1 then x

check in or  
if length < 1 then x

value type

Recursive?

~~check id count if length > 1, throw~~

or

assign value to the 1st encountered id, compare, if ≠, throw

### Options:

1. check if columns.length < 1 x
2. check if columns.value not in list of mkey/skey.
3. check if order exist: if true: key in order must exist