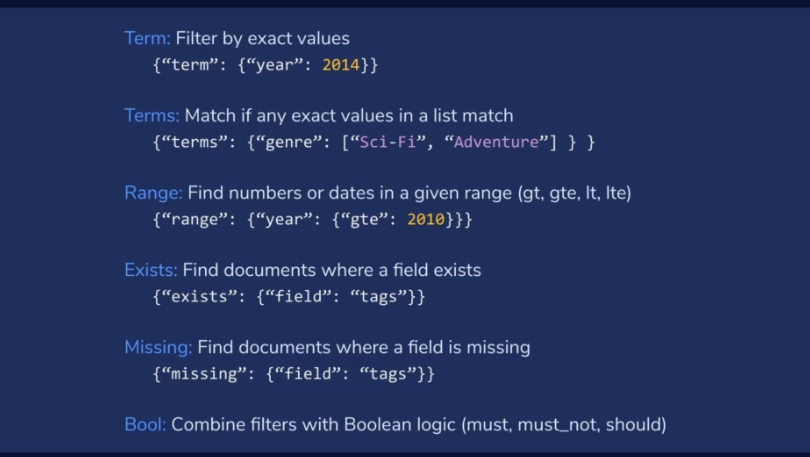
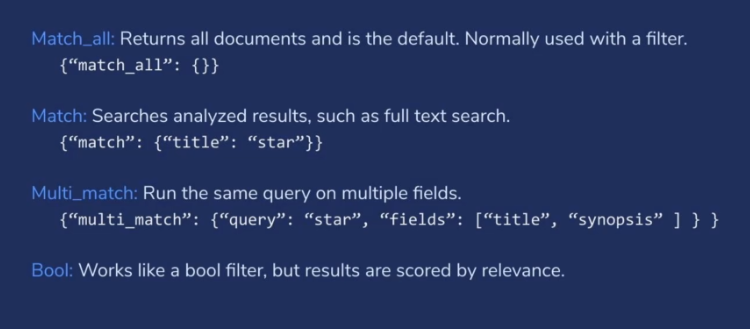
****

****

**==============Analyzers===============**

String => "The lazy fox jump over top"

Text ==> "The", "lazy", "fox", "jump", "over", "top"

Keyword ==> "The lazy fox jump over top"

when you want then stored 2 way text and keyword

Manufacturer -> text

Manufacturer.keyword -> text

**Analyzers Part**

1. char filter ==> used to replace text like :) <-> happy , (: <-> sad
2. tokenizer ==> help to break the sentence like we have some predefined tokenizer

"The 2 QUICK Brown-Foxes jumped over the lazy dog's bone."

* 1. standard (most common) => [ The, 2, QUICK, Brown, Foxes, jumped, over, the, lazy, dog's, bone ]
  2. letter (remove other then letter)=> [ The, QUICK, Brown, Foxes, jumped, over, the, lazy, dog, s, bone ]
  3. lowercase (remove other then letter and convert into lower case)=> [ the, quick, brown, foxes, jumped, over, the, lazy, dog, s, bone ]
  4. whitespace (based on whitespace) ==> [ The, 2, QUICK, Brown-Foxes, jumped, over, the, lazy, dog's, bone. ]
  5. classic (classic tokenizer is a grammar based tokenizer that is good for English language documents. This tokenizer has heuristics for special treatment of acronyms, company names, email addresses, and internet host names. ) ==> [ The, 2, QUICK, Brown, Foxes, jumped, over, the, lazy, dog's, bone ]
  6. keyword (stored same as it is) => "The 2 QUICK Brown-Foxes jumped over the lazy dog's bone."
  7. pattern => "The foo\_bar\_size's default is 5." => [ The, foo\_bar\_size, s, default, is, 5 ]
  8. uax\_url\_email => same as statrtand but undstand email and url

1. filter ==> apply filter on after tokenizer like lowercase, remove some word like 'i', 'am'
   1. it help in autosugget

"Brown"

1. Ngram -> (2, 5) -> “Br”, “ro”, “ow“, “wn”….
2. EdgerNgram 🡪 (2,5) 🡪 br, bro, brow, brown
   1. IT‘s is help to search like bike and get other bike without keyword

For testing create folder in Elasticsearch config folder and then file

* + 1. Synonyms

first close index

PUT my\_analyzers/\_settings

{

"analysis": {

"analyzer": {

"my\_custome\_analyze": {

"char\_filter": [

"emotions"

],

"tokenizer": "punctuation",

"filter": [

"lowercase",

"english\_stop"

]

}

},

"char\_filter": {

"emotions": {

"type": "mapping",

"mappings": [

":) => \_happy\_",

":( => \_sad\_"

]

}

},

"tokenizer": {

"punctuation": {

"type": "pattern",

"pattern": "[ ,'.!?]"

}

},

"filter": {

"english\_stop": {

"type": "stop",

"stopwords": "\_english\_"

}

}

}

}