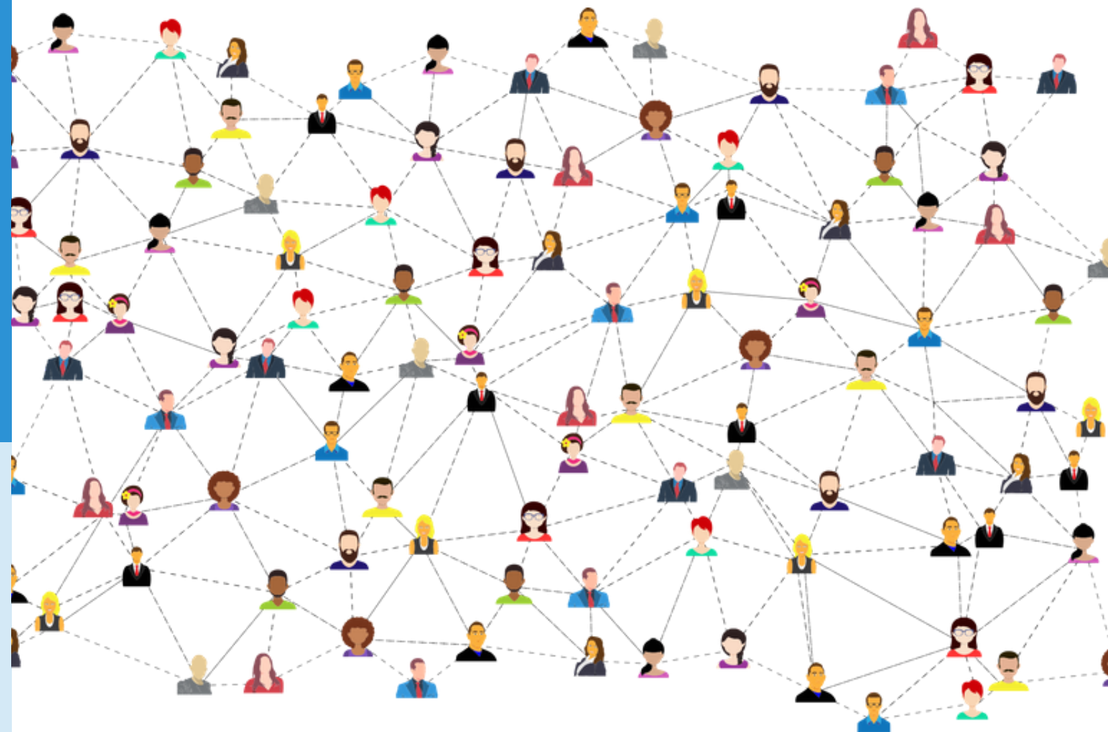
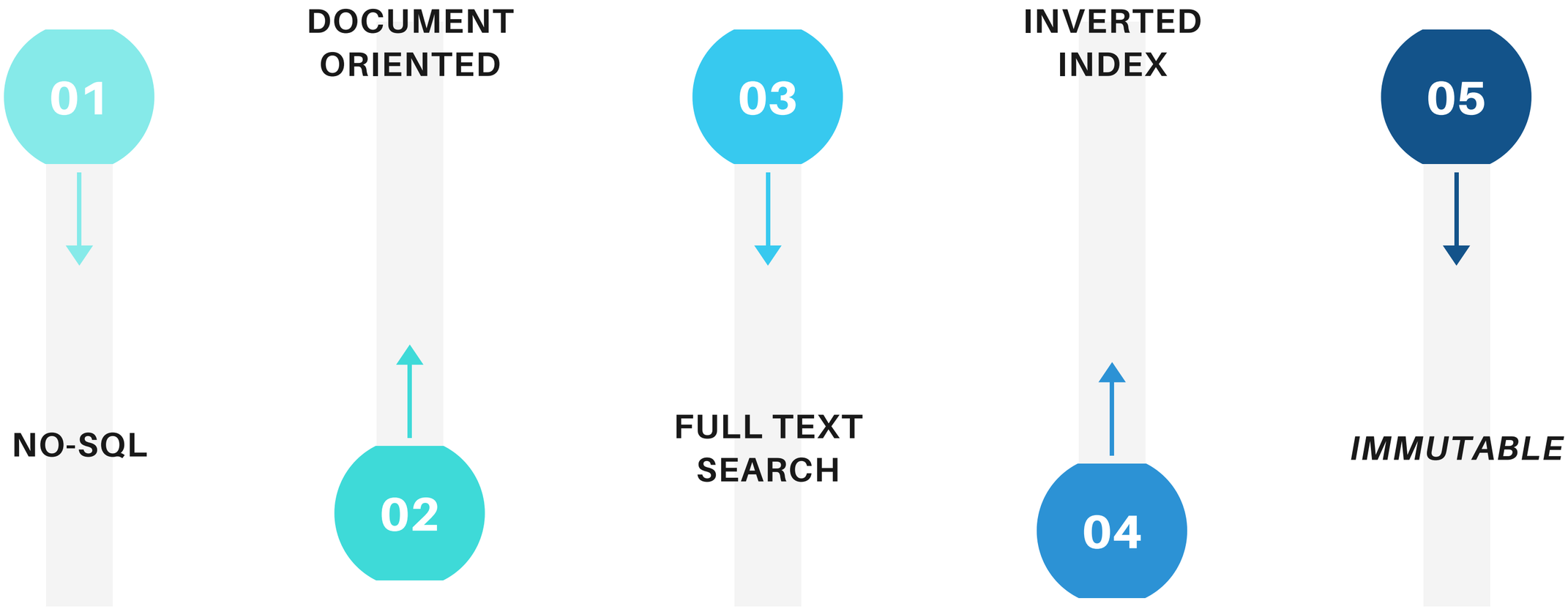
Effective Elasticsearch

SHANI COHEN



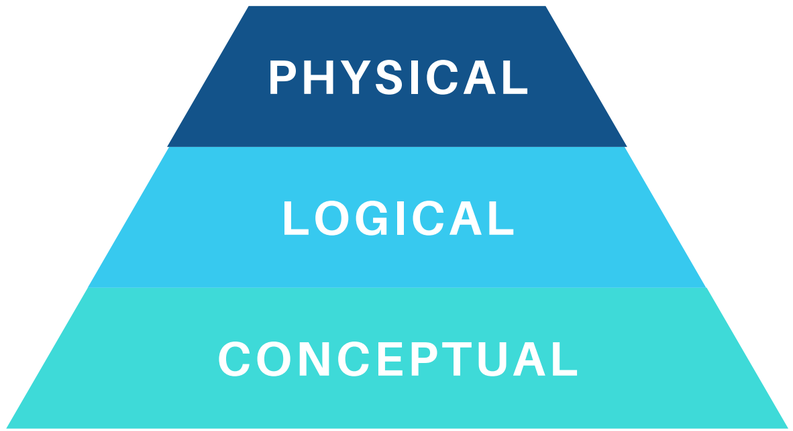
Powerful Elasticsearch Mapping

Elasticsearch



PROCESS OF CREATING A DATA MODEL

1. *Conceptual*
2. *Logical*
3. *Physical*

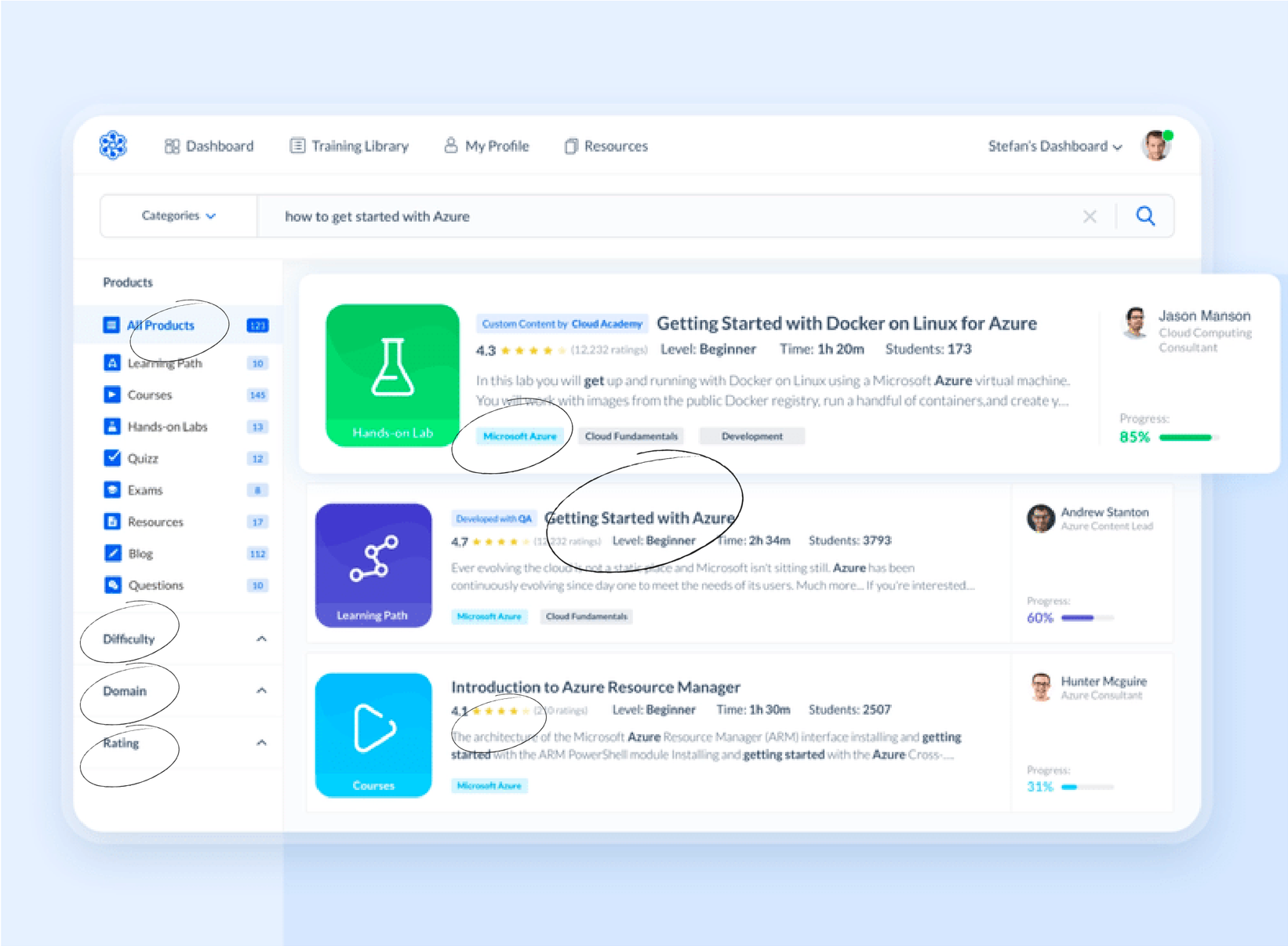


Concetual

Logical

Physical

CONCEPTUAL


Dashboard
Training Library
My Profile
Resources
Stefan's Dashboard
Categories
how to get started with Azure
Products
All Products 121
Learning Path 10
Courses 145
Hands-on Labs 13
Quizz 12
Exams 8
Resources 17
Blog 112
Questions 10
Difficulty
Domain
Rating
Hands-on Lab
Custom Content by Cloud Academy
Getting Started with Docker on Linux for Azure
4.3 (12,232 ratings) Level: Beginner Time: 1h 20m Students: 173
In this lab you will get up and running with Docker on Linux using a Microsoft Azure virtual machine. You will work with images from the public Docker registry, run a handful of containers,and create y...
Microsoft Azure
Cloud Fundamentals
Development
Jason Manson
Cloud Computing Consultant
Progress:
85%
Learning Path
Developed with QA
Getting Started with Azure
4.7 (13,232 ratings) Level: Beginner Time: 2h 34m Students: 3793
Ever evolving the cloud is not a static place and Microsoft isn't sitting still. Azure has been continuously evolving since day one to meet the needs of its users. Much more... If you're interested...
Microsoft Azure
Cloud Fundamentals
Andrew Stanton
Azure Content Lead
Progress:
60%
Courses
Introduction to Azure Resource Manager
4.1 (220 ratings) Level: Beginner Time: 1h 30m Students: 2507
The architecture of the Microsoft Azure Resource Manager (ARM) interface installing and getting started with the ARM PowerShell module Installing and getting started with the Azure Cross-...
Microsoft Azure
Hunter Mcguire
Azure Consultant
Progress:
31%

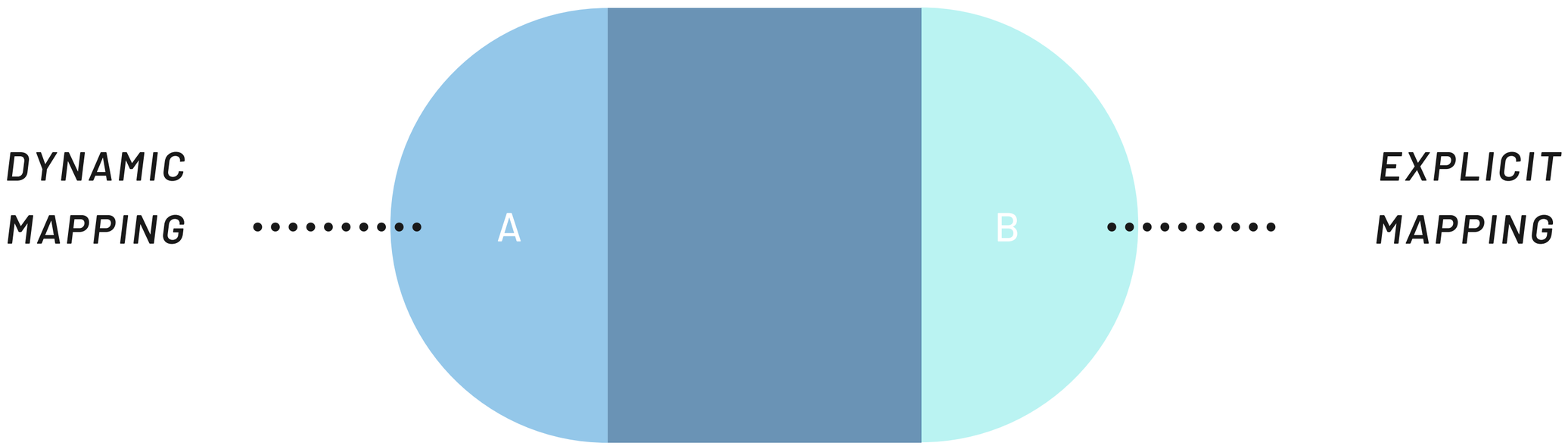
LOGICAL

```
PUT /MY-INDEX-000001
{
 "MAPPINGS": {
  "PROPERTIES": {
   "AGE":   { "TYPE": "INTEGER" },
   "EMAIL":  { "TYPE": "KEYWORD"  },
   "NAME":  { "TYPE": "TEXT"  }
  }
 }
}
```

“All models are wrong, but some are useful.”

George Box

SCHEMALESS DATA MODELING?

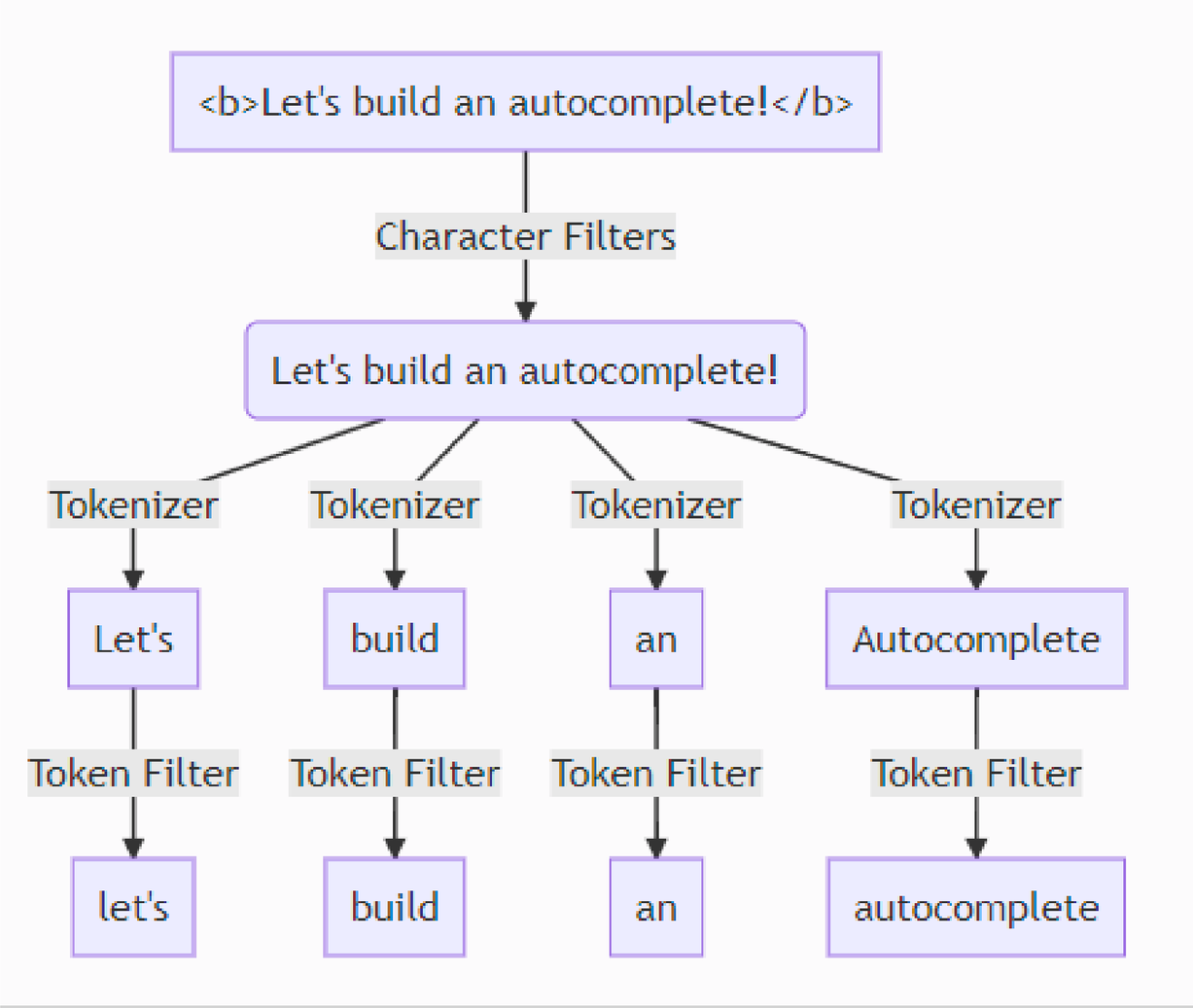


Schema changes affect?

LUCENE INVERTED INDEX

Term	Document #1	Document #2
best	X	
carbonara		X
delicious		X
pasta	X	X
pesto	X	
recipe	X	X
the	X	
with	X	

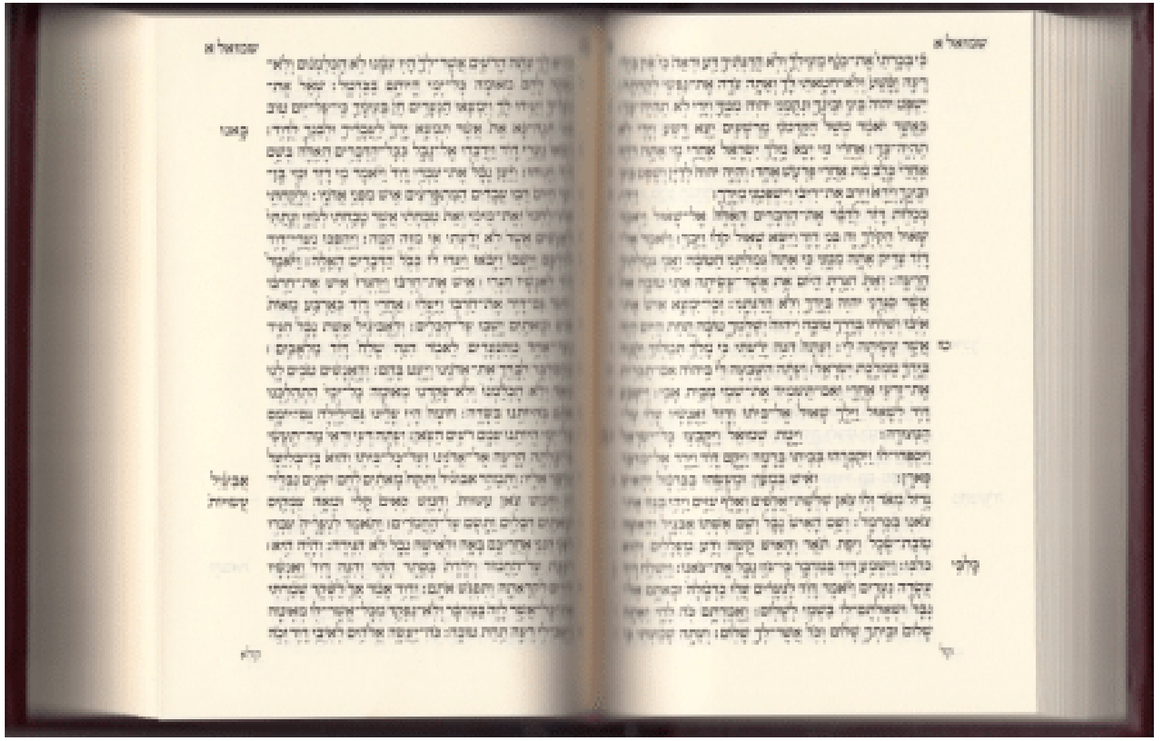
ANALYSIS


Let's build an autocomplete!
Character Filters
Let's build an autocomplete!
Tokenizer
Tokenizer
Tokenizer
Tokenizer
Let's
build
an
Autocomplete
Token Filter
Token Filter
Token Filter
Token Filter
let's
build
an
autocomplete

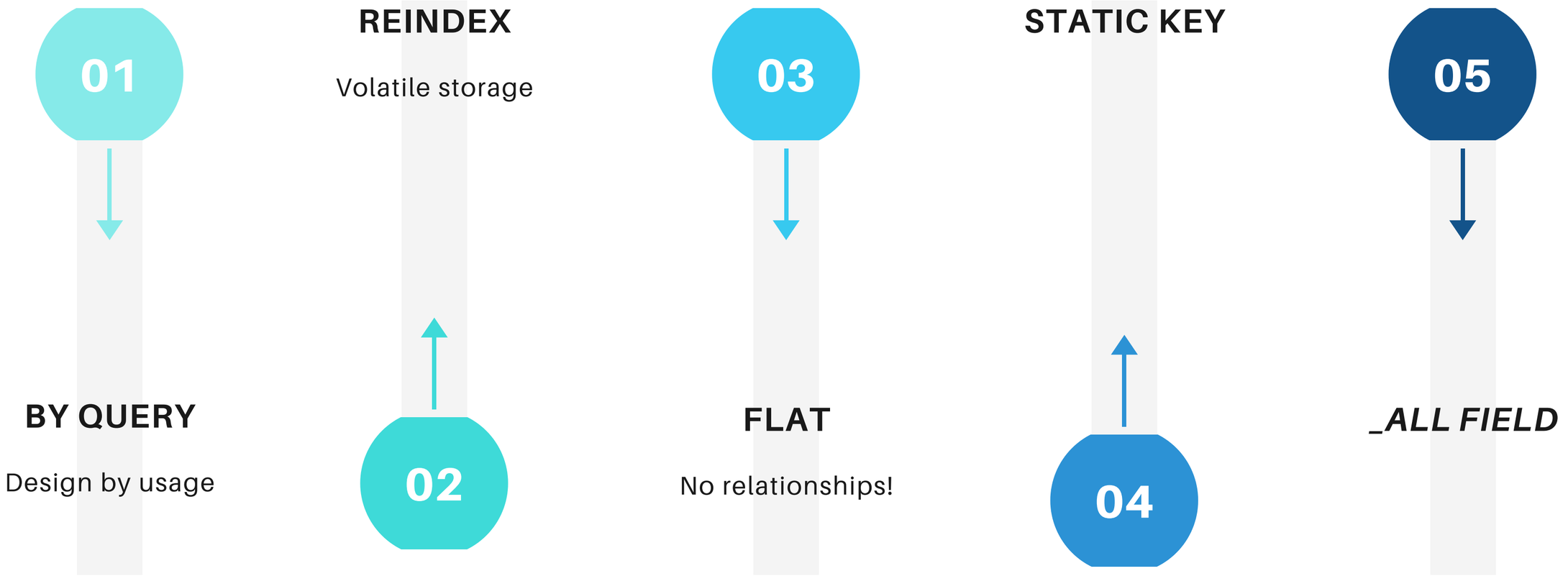
Core Field Types

Field Type	1 Searchable	2 Aggregable	3 Example
Number		●	45
Keyword		●	"u8ui-u78i-kki8-okih"
Text	●		You know...
Boolean		●	True

Single Source Of True



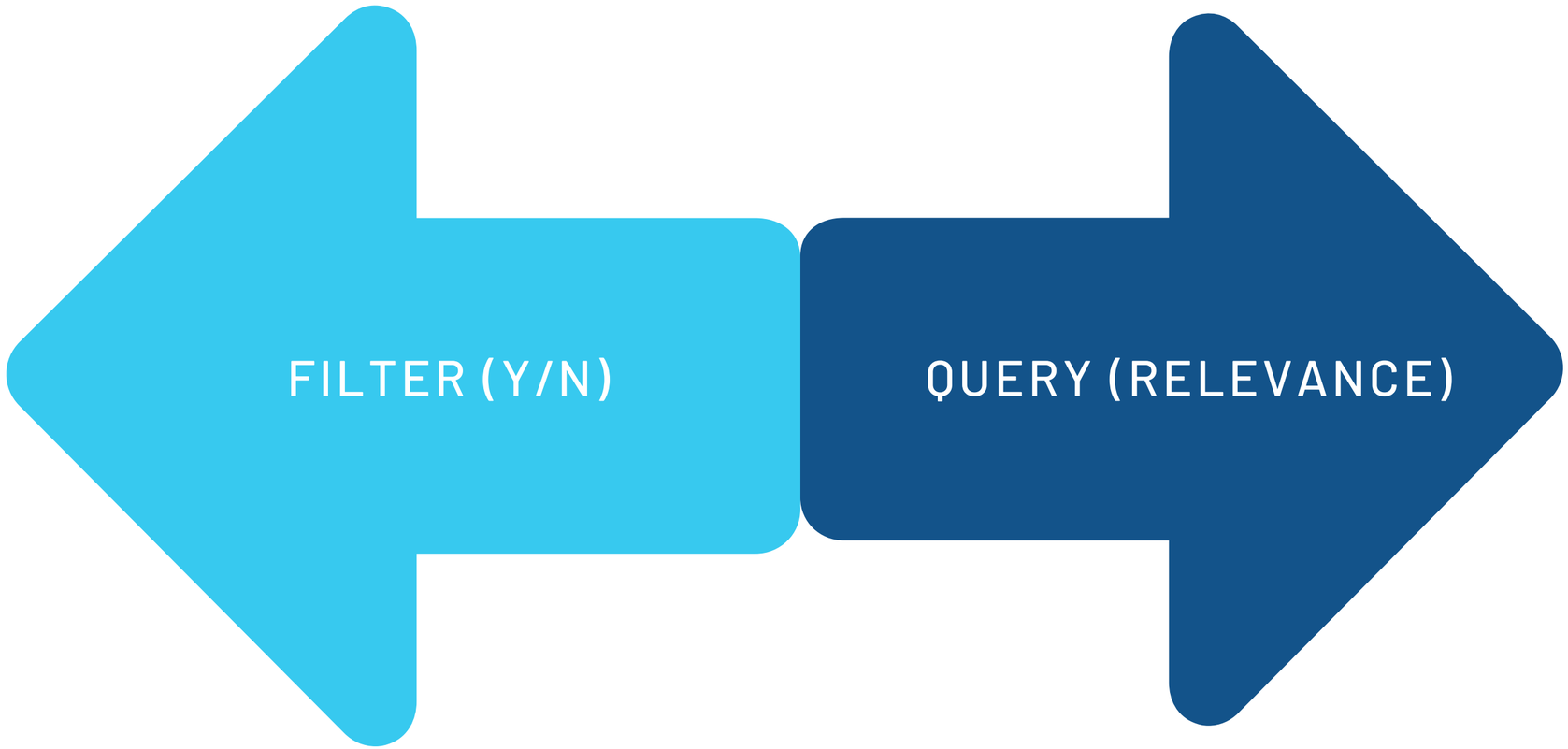
Mapping key notes



Effective Elasticsearch Querying

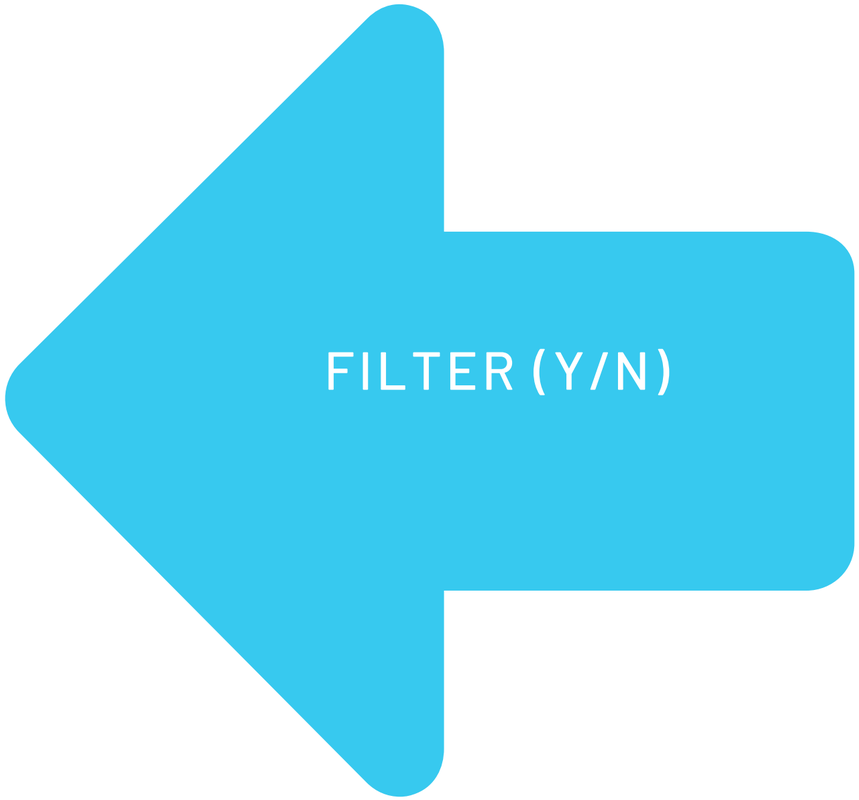
Do you really need to score your documents while querying?

QUERY LATENCY

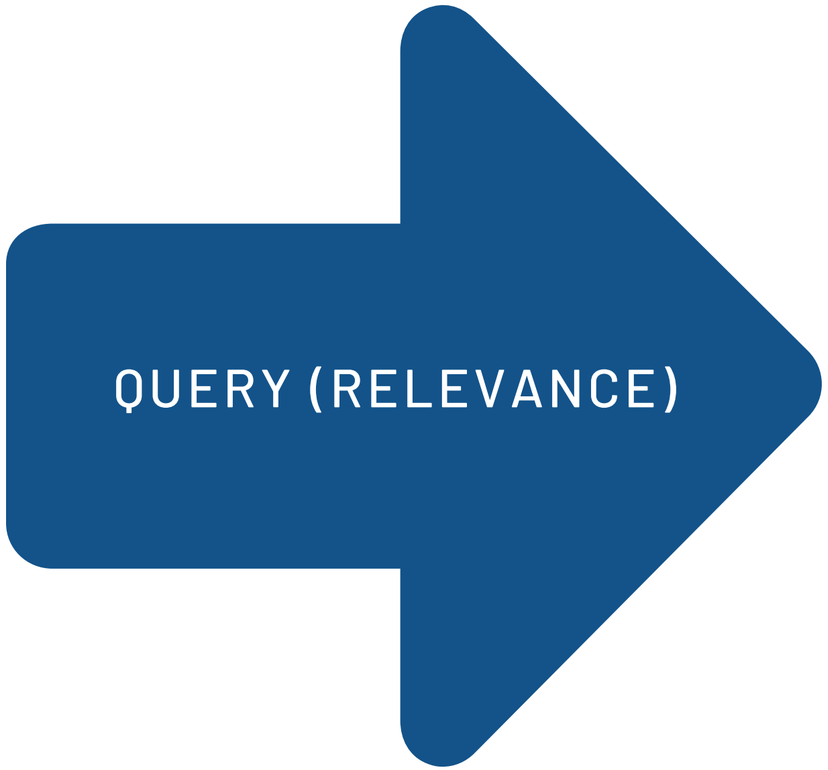


QUERY LATENCY

Filter out documents which do not match

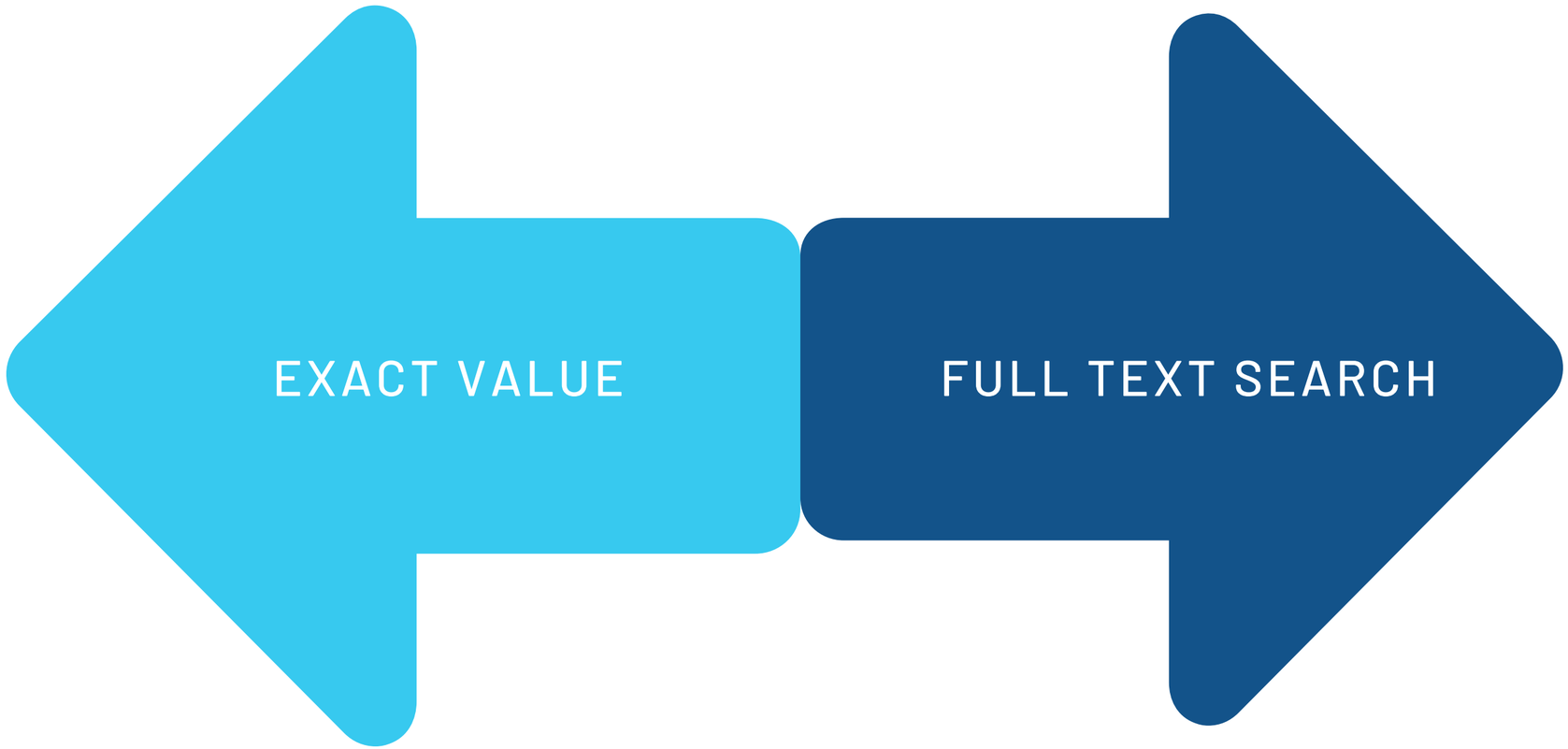


QUERY CONTEXT



How well does the document match

QUERY LATENCY



EXACT VALUE

- Term
- Range
- Regexp, wildcard, and prefix
- exists
- fuzzy

EXACT VALUE
TERM SEARCH

```
GET /_search
{
  "query": {
    "term": {
      "<field_name>": {
        "value": "<your_value>"
      }
    }
  }
}
```

- Exactly matches the criteria
- select * from table_name where column_name =...
- For keyword
- In query/filter

EXACT VALUE
TERM QUERY WITH FILTER

```
GET /_search
{
  "query": {
    "constant_score" : {
      "filter" : {
        "term" : {"<field_name>" : "<your_value>"}
      }
    }
  }
}
```

FULL TEXT SEARCH

Full-text queries take advantage of the analyzer

- Match
- Match Phrase
- Multi-Match

MATCH QUERY

```
GET /_search
{
 "query" : {
  "match" : {
   "<text_field>" {
    "query" : "<your_value>"
   }
  }
 }
}
```

MATCH_PHRASE QUERY

```
GET /_search
{
  "query": {
    "match_phrase" : {
      "<text_field>" : {
        "query" : "<your_value>",
        "slop" : "0"
      }
    }
  }
}
```

MULTY_MATCH QUERY

```
GET /_search
{
 "query": {
  "multi_match" : {
   "query":    "<your_value>",
   "fields": [ "<text_field1>", "<text_field2>" ]
  }
 }
}
```

COMPOUND QUERY

- Bool
  - Must
  - Should
  - Must not
  - Filter
- Boost
- Constant score query

BOOL QUERY

```
POST _search
{
 "query": {
  "bool" : {
   "must" : {
    "term" : { "user.id" : "kimchy" }
   },
   "filter": {
    "term" : { "tags" : "production" }
   },
   "must_not" : {
    "range" : {
     "age" : { "gte" : 10, "lte" : 20 }
    }
   },
   "should" : [
    { "term" : { "tags" : "env1" } },
    { "term" : { "tags" : "deployed" } }
   ]
  }
 }
```

DEBUGGING QUERY

- Explain
- Profile
- Named

EXPLAIN

```
GET /my-index-000001/_explain/0
{
 "query" : {
  "match" : { "message" : "elasticsearch" }
 }
}
```

PROFILE

```
GET /my-index-000001/_search
{
 "profile": true,
 "query" : {
  "match" : { "message" : "GET /search" }
 }
}
```

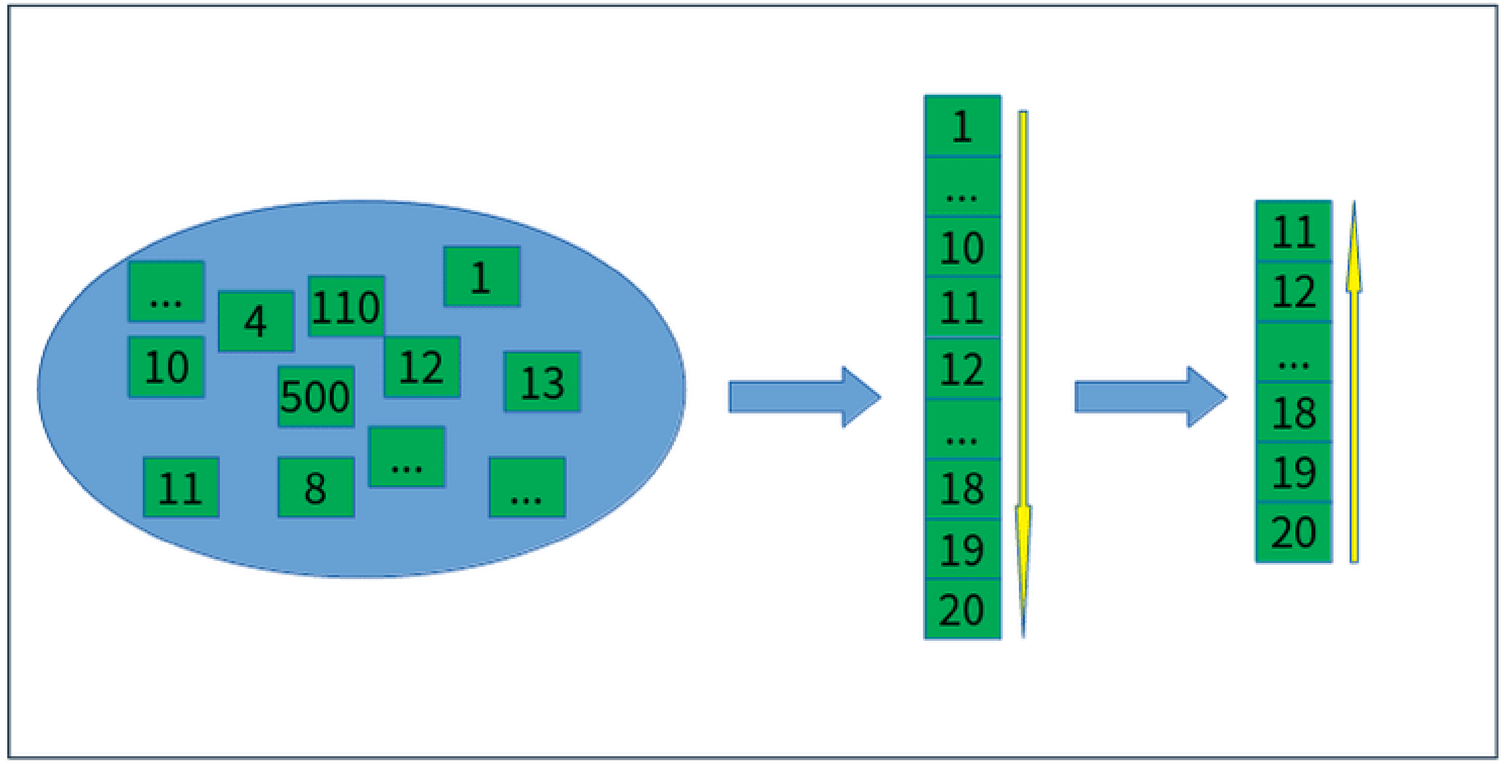
USEFUL QUERIES - NAMED

```
GET /_search
{
  "query": {
    "bool" : {
      "should" : [
        {"match" : { "name.first" : {"query" : "shay", "_name" : "first"} }},
        {"match" : { "name.last" : {"query" : "banon", "_name" : "last"} }}
      ],
      "filter" : {
        "terms" : {
          "name.last" : ["banon", "kimchy"],
          "_name" : "test"
        }
      }
    }
  }
}
```

PAGINATION

```
GET /_search
{
 "from": 5,
 "size": 20,
 "query": {
  "match": {
   "user.id": "kimchy"
  }
 }
}
```

DEEP PAGINATION



Questions?