TECHNICAL UNIVERSITY OF LIBEREC
Faculty of Mechatronics, Informatics
and Interdisciplinary Studies

ELASTICSEARCH

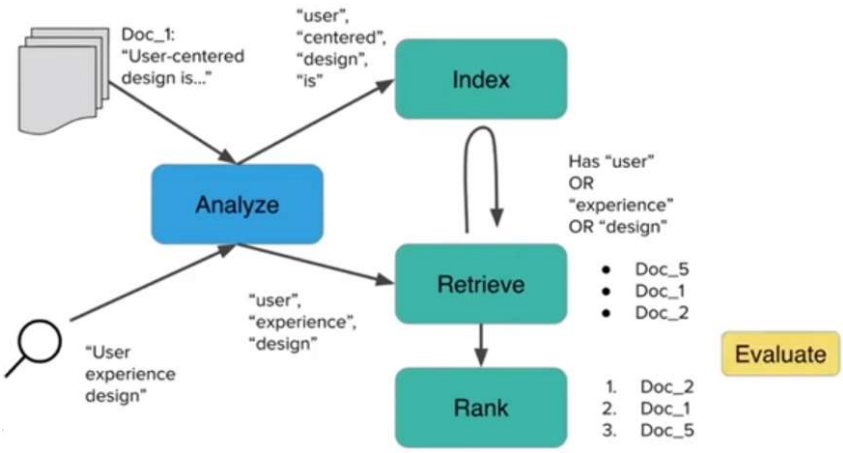
Lukáš Matějů

22.4.2024 | DPB

ČÁST I.:
OPAKOVÁNÍ

OPAKOVÁNÍ

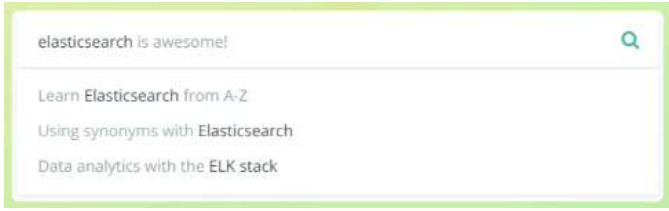
- search-engine databáze
  - NoSQL dokumentové databáze zaměřené na vyhledání obsahu
  - využívají indexování
    - kategorizace podobných vlastností mezi daty
    - ➢ urychlení vyhledávání
  - optimalizované pro práci s daty
    - velké množství dat
    - strukturovaná i nestrukturovaná data
      - volné schéma
  - poskytují speciální funkce
    - full-textové vyhledávání
    - složité vyhledávací výrazy
    - řazení výsledků
    - distribuované vyhledávání



https://www.youtube.com/watch?v=dqRDyeFJUvk

OPAKOVÁNÍ

- Elasticsearch
  - engine pro full-textové prohledávání a analýzu
  - open source
    - napsán v Javě
    - založený na Apache Lucene
  - dokumentová databáze
    - data ukládána do dokumentů s poli
    - využívá JSON
  - dotazování pomocí REST API
  - distribuovaný
    - vysoká škálovatelnost
    - ➢ rychlost prohledávání
  - součástí Elastic Stack
    - Kibana, Logstash, X-Pack, Beats

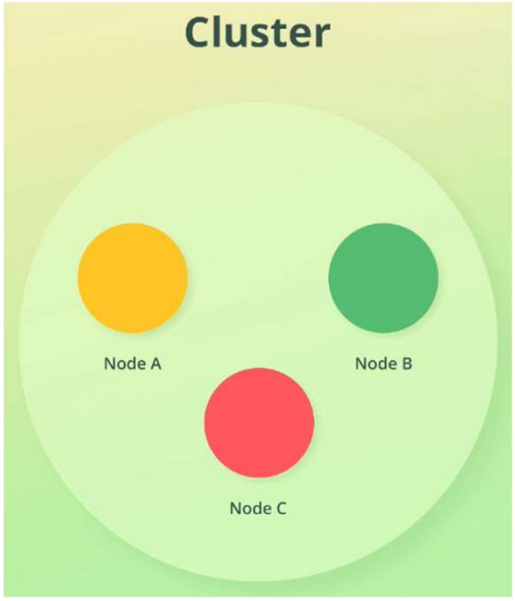


https://www.udemy.com/course/elasticsearch-complete-guide/

elasticsearch

OPAKOVÁNÍ

- Elasticsearch
  - engine pro full-textové prohledávání a analýzu
  - architektura
    - základem je uzel (node)
      - fyzický, virtuální, docker kontejner
    - každý uzel je součástí clusteru
    - základní jednotkou pro ukládání dat je dokument
      - JSON objekty obsahující data
    - data jsou organizována v indexech



https://www.udemy.com/course/elasticsearch-complete-guide/

OPAKOVÁNÍ

- komunikace s Elasticsearch REST API
  - využití HTTP žádostí pro přístup k datům
    - metody POST, GET, PUT a DELETE
    - odpovídá CRUD – vytvoření, čtení, update, smazání dat
    - libovolný HTTP klient
  - využití Kibany -> http://localhost:5601
    - vestavěná konzole
    - překládá uživatelské dotazy na HTTP žádosti
    - žádosti následně posílány do Elasticsearch
    - dotaz na stav clusteru

```
GET /_cluster/health
```

  - využití ovladačů pro přístup z aplikací

OPAKOVÁNÍ

```
PUT /products
```

```
{
  "acknowledged" : true,
  "shards_acknowledged" : tru
  "index" : "products"
}
```

```
PUT /products

POST /products/_doc
{
  "name" : "Coffee Maker",
  "price" : 64,
  "in_stock" : 10
}
```

```
{
  "_index" : "products",
  "_type" : "_doc",
  "_id" : "f_Ox23UBkaLlnoKa18am",
  "_version" : 1,
  "result" : "created",
  "_shards" : {
    "total" : 2,
    "successful" : 2,
    "failed" : 0
```

```
POST /products/_update/100
{
  "doc" : {
    "in_stock" : 3,
    "tags" : ["electronics"]
  }
}

GET /products/_doc/100
```

```
{
  "_index" : "products",
  "_type" : "_doc",
  "_id" : "100",
  "_version" : 3,
  "result" : "noop",
  "_shards" : {
    "total" : 0,
    "successful" : 0,
    "failed" : 0
  },
  "_seq_no" : 4,
  "_primary_term" : 1
}
```

```
DELETE /products/_doc/100

GET /products/_doc/100
```

```
{
  "_index" : "products",
  "_type" : "_doc",
  "_id" : "100",
  "_version" : 13,
  "result" : "deleted",
  "_shards" : {
    "total" : 2,
    "successful" : 2,
    "failed" : 0
  },
```

```
GET /products/_search
{
  "query": {
    "match": {
      "name": "Lobster"
    }
  }
}
```

```
  "hits" : {
    "total" : {
      "value" : 5,
      "relation" : "eq"
    },
    "max_score" : 6.035804,
    "hits" : [
      {
        "_index" : "products",
        "_type" : "_doc",
        "_id" : "19",
        "_score" : 6.035804,
        "_source" : {
          "name" : "Lobster - Live",
```

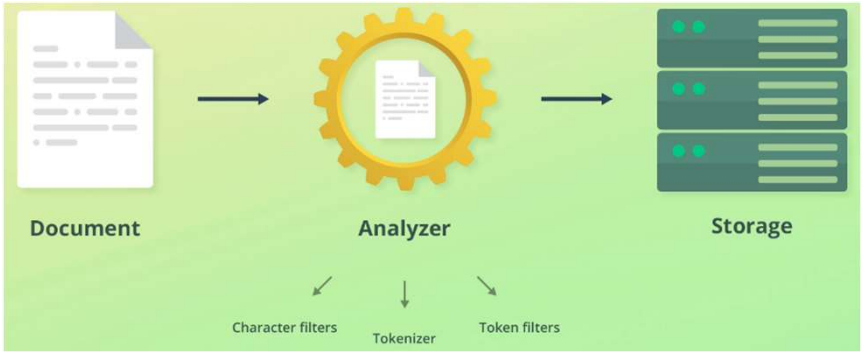
```
GET /products/_search
{
  "query": {
    "term": {
      "name": "lobster"
    }
  }
}
```

```
  "hits" : {
    "total" : {
      "value" : 5,
      "relation" : "eq"
    },
    "max_score" : 6.035804,
    "hits" : [
      {
        "_index" : "products",
        "_type" : "_doc",
        "_id" : "19",
        "_score" : 6.035804,
        "_source" : {
          "name" : "Lobster - Live",
```

ČÁST II.:
ELASTICSEARCH
ANALÝZA A MAPOVÁNÍ

ANALÝZA

- často také jako textová analýza
  - reálná aplikace jen na textová pole / hodnoty
- vstupní data dokumentu v poli _source
  - ta ale nejsou přímo používána pro prohledávání
    - např. dlouhý popis produktu nelze efektivně prohledávat bez předzpracování
- ➢ při indexaci jsou textová pole analyzována
  - o analýzu se stará analyzér skládající se ze tří komponent
    - znakový filtr (character filter)
    - tokenizér (tokenizer)
    - token filtr (token filter)
  - výsledky analýzy uloženy v polích efektivních pro prohledávání



https://www.udemy.com/course/elasticsearch-complete-guide/

ANALYZÉR

- znakové filtry (character filters)
  - transformují vstupní text úpravou znaků
    - přidání, změna, odstranění
  - nemusí být žádný, ale i několik
  - případně aplikovány v předem definovaném pořadí
  - např. filtr html_strip
    - filtr odstraňující HTML elementy a transformující HTML entity

Input: "I'm in a good mood - and I love açaí!"

Output: "I'm in a good mood - and I love açaí!"

https://www.udemy.com/course/elasticsearch-complete-guide/

  - např. filtr odstraňující diakritiku

ANALYZÉR

- tokenizér (tokenizer)
  - přesně jeden v analyzéru
  - stará se o tokenizaci
    - rozdělení řetězce na tokeny (token = nejmenší jednotka textu; většinou grafické slovo)
  - může odstraňovat znaky
    - interpunkce
  - ukládá také offset znaků
  - např. rozdělení řetězce podle bílého znaku
    - výsledkem jsou 4 tokeny

Input: `"I REALLY like beer!"`

Output: `["I", "REALLY", "like", "beer"]`

https://www.udemy.com/course/elasticsearch-complete-guide/

ANALYZÉR

- token filtry (token filters)
  - dostávají výstup z tokenizéru jako vstup
  - mohou přidávat, modifikovat nebo mazat tokeny
  - nemusí být žádný, ale i několik
  - případně aplikovány v předem definovaném pořadí
  - např. lowercase filtr
    - převedení všech tokenů na malá písmena

Input: `["I", "REALLY", "like", "beer"]`

Output: `["i", "really", "like", "beer"]`

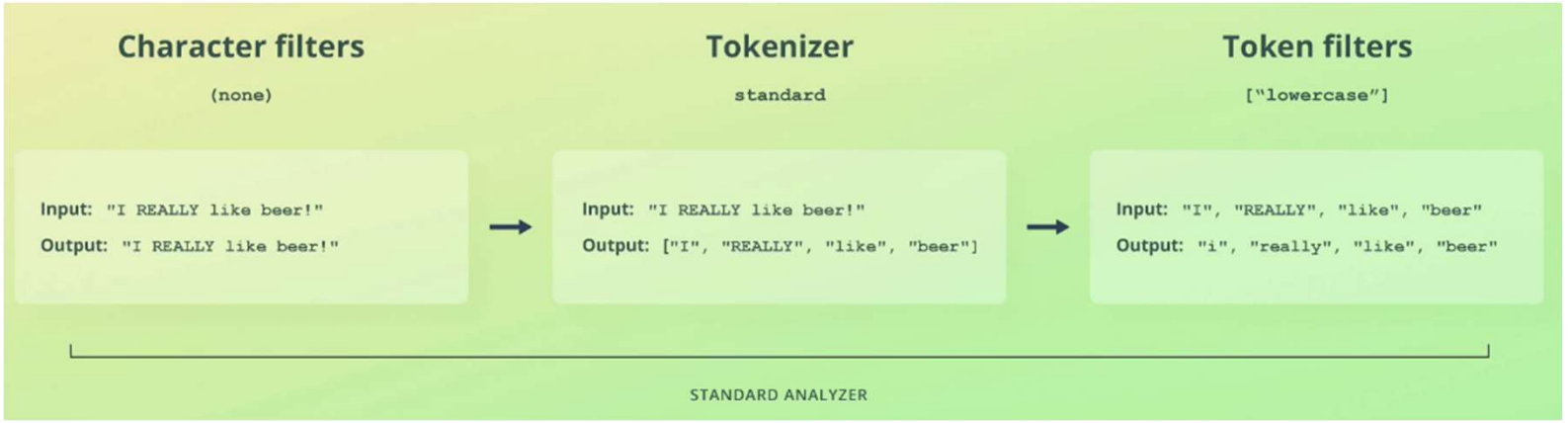
https://www.udemy.com/course/elasticsearch-complete-guide/

ANALÝZA

- k dispozici vestavěné analyzéry
  - znakové filtry, tokenizéry i filtry tokenů
  - možnost definovat i vlastní
- co se stane s textem v základním nastavení?
  - žádný znakový filtr není použit
  - tokenizer dělí řetězec na tokeny podle Unicode Segmentation algoritmu
    - víceméně rozděluje řetězec podle bílých znaků, pomlček a podobně
    - odstraňuje také interpunkci
  - lowercase token filter převádí tokeny na malá písmena
  - ➢ standardní analyzér
    - použit na všechna textová pole, pokud není specifikováno jinak
    - ➢ použit ve většině případů

ANALÝZA

- co se stane s textem v základním nastavení?
  - standardní analyzér



https://www.udemy.com/course/elasticsearch-complete-guide/

DATOVÉ STRUKTURY

- co se děje s výslednými tokeny po analýze?
- hodnoty polí jsou uloženy v jedné z několika datových struktur
  - použitá datová struktura závisí na datovém typu pole
- zajišťují elektivní přístup k datům
  - např. pro prohledávání
  - např. vyhledávání přesného termínu je obsluhováno jinak než agregace dat
- v úplném jádru se o datové struktury stará Apache Lucene
  - ne Elasticsearch

➢ (textové) tokeny jsou uloženy v datové struktuře invertovaný index

INVERTOVANÉ INDEXY

- mapování mezi výrazy a dokumenty, které je obsahují
  - mimo kontext analyzéru jsou tokeny označovány jako výrazy (terms)
  - výrazy jsou řazeny abecedně
  - dokumenty jsou odkazovány pomocí _id

TERM	DOCUMENT #1	DOCUMENT #2	DOCUMENT #3
2	X	X	X
a	X	X	
around			X
bar	X	X	
but	X		
ducks	X		X
guys	X	X	
into	X	X	
lake			X
the	X		X
third	X		
walk	X		X
went		X	

https://www.udemy.com/course/elasticsearch-complete-guide/

INVERTOVANÉ INDEXY

- efektivní vyhledání výrazu a dokumentů, kde se výraz vyskytuje
- proč invertovaný?
  - obrácené mapování logičtější
    - dokument odkazuje na výrazy, které obsahuje
    - to ale neumožňuje efektivní vyhledávání výrazů
    - invertovaný index
- v praxi obsahují invertované indexy další informace
  - např. informace pro skóre relevance

TERM	DOCUMENT #1	DOCUMENT #2	DOCUMENT #3
2	X	X	X
a	X	X	
around			X
bar	X	X	
but	X		
ducks	X		X
guys	X	X	
into	X	X	
lake			X
the	X		X
third	X		
walk	X		X
went		X	

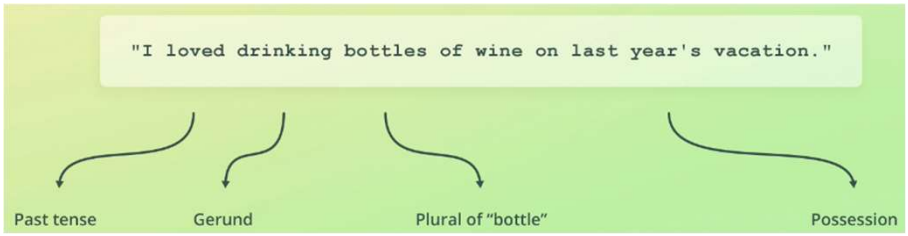
https://www.udemy.com/course/elasticsearch-complete-guide/

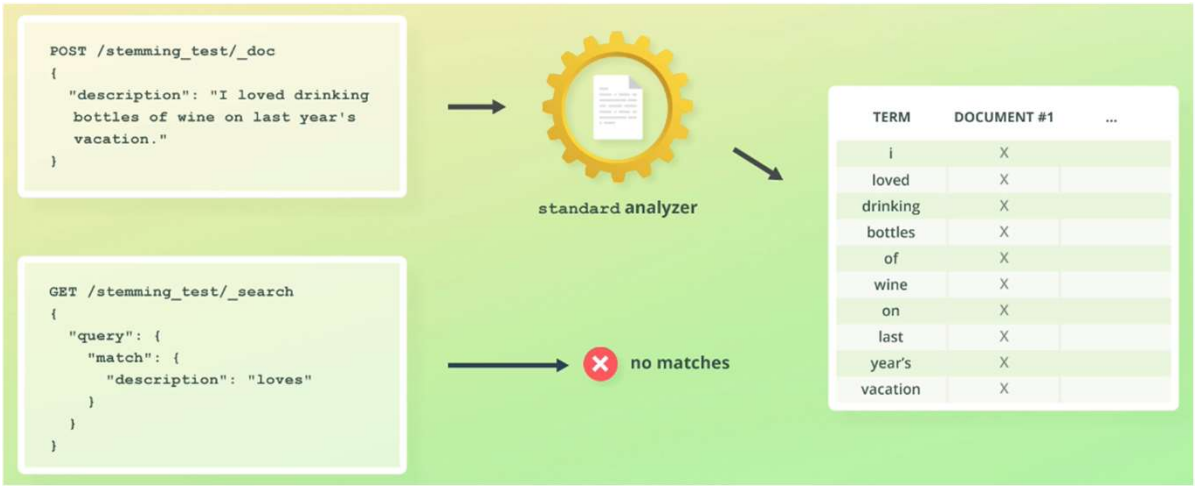
INVERTOVANÉ INDEXY

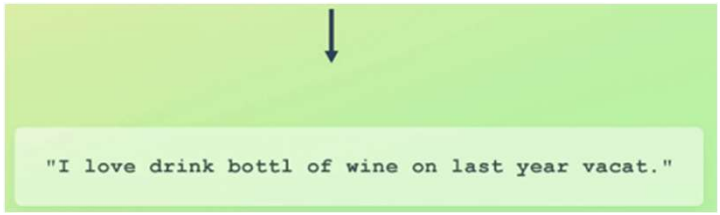
- vytvářeny pro každé textové pole
  - jeden invertovaný index = jedno textové pole
  - definovány na úrovní polí
  - pro netextová pole používány jiné datové struktury
    - např. BKD stromy pro číselné hodnoty a datumy

ANALYZÉRY

- spousta vestavěných analyzérů
  - standard, simple, whitespace, keyword, pattern, language (i pro češtinu)
  - možno upravovat i vytvářet vlastní
  - jazykové analyzéry často provádí stematizaci
    - proces redukující slova na jejich kmen

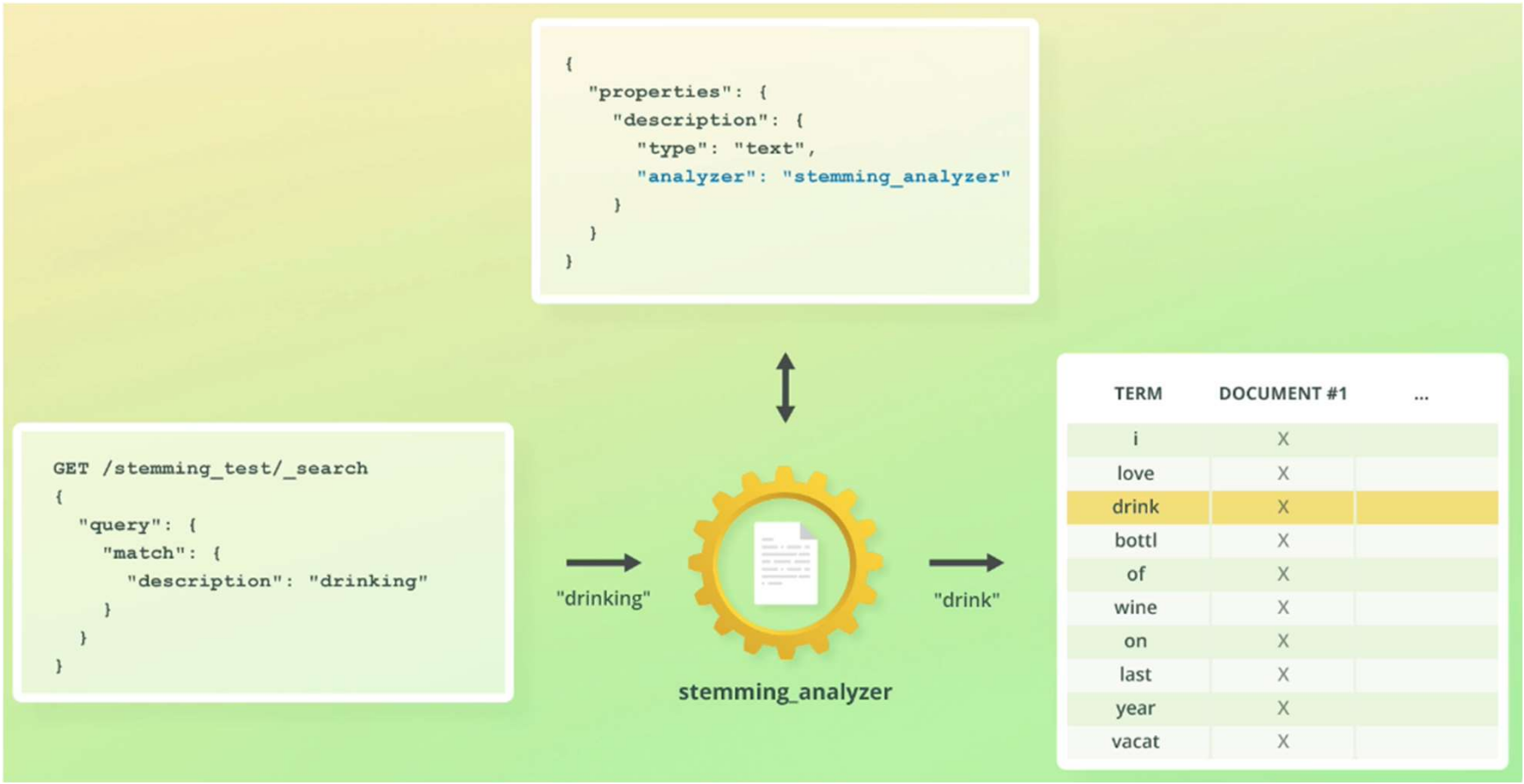






https://www.udemy.com/course/elasticsearch-complete-guide/

ANALYZÉRY



https://www.udemy.com/course/elasticsearch-complete-guide/

MAPOVÁNÍ

- definuje strukturu dokumentů, a jak jsou indexovány a ukládány
  - pole a jejich datové typy
- zjednodušeně se dá přirovnat k relaci v relačních databázích

https://www.udemy.com/course/elasticsearch-complete-guide/

MAPOVÁNÍ

- Elasticsearch podporuje dva typy mapování
  - explicitní
    - manuální definice polí a jejich datových typů
    - zpravidla při indexaci
  - dynamické
    - Elasticsearch se stará o mapování
    - podle dodané hodnoty vyřeší, jak by pole mělo být namapováno
    - např. vstupní hodnotou je řetězec, Elasticsearch přiřadí datový typ text
  - možno kombinovat
- povoluje chybějící pole u dokumentů

DATOVÉ TYPY

- obrovské množství podporovaných datových typů
  - boolean, short, integer, long, float, double, date, …
  - object
    - použitý pro všechny JSON objekty
    - objekty mohou být vnořené
  - nested
    - podobný object, ale uchovává vztahy mezi objekty
    - užitečné pro indexaci polí objektů
  - text
    - částečná shoda vyhledávání
    - fulltextové vyhledávání
  - keyword
    - přesná shoda vyhledávání

```
PUT /products
{
  "mappings": {
    "properties": {
      "name": { "type": "text" },
      "price": { "type": "double" },
      "in_stock": { "type": "short" },
      "is_active": { "type": "boolean" },
      "manufacturer": {
        "properties": {
          "name": { "type": "text" },
          "country": { "type": "text" }
        }
      }
    }
  }
}
```

https://www.udemy.com/course/elasticsearch-complete-guide/

KEYWORD TYP

- analyzován keyword analyzérem
  - jedná se o no-op analyzér
  - neprovádí žádnou operaci
  - jen vrací vstupní řetězec jako jeden token
  - ➢ určeno pro přesnou shodu s vyhledávaným řetězcem

```
POST /_analyze
{
  "text": "2 guys walk into   a bar, but the third... DUCKS! :-)",
  "analyzer": "keyword"
}
```

```
{
  "tokens" : [
    {
      "token" : "2 guys walk into   a bar, but the third... DUCKS! :-)",
      "start_offset" : 0,
      "end_offset" : 53,
      "type" : "word",
      "position" : 0
    }
  ]
}
```

TERM	DOCUMENT #1	DOCUMENT #2	DOCUMENT #3
2 ducks walk around the lake	X		
2 guys walk into a bar, but the third... DUCKS! :-)		X	
2 guys went into a bar			X

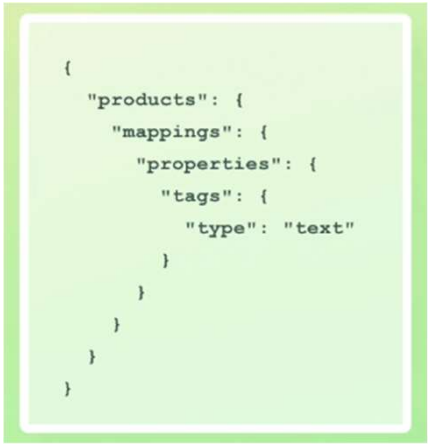
KEYWORD TYP

- analyzován keyword analyzérem
  - vhodné pro filtrování, agregace, řazení
  - např. emailové adresy
    - pozn. vhodné by ale bylo využít lowercase filtr

https://www.udemy.com/course/elasticsearch-complete-guide/

POLE (ARRAYS)

- datový typ pro pole (array) neexistuje
- každé pole (field) může obsahovat nula nebo více hodnot
  - mapování není potřeba
  - v případě řetězců jsou řetězce spojeny a tokeny uloženy do invertovaného indexu
  - v ostatních případech data nejsou analyzována



https://www.udemy.com/course/elasticsearch-complete-guide/

- musí obsahovat stejný datový typ
- vnořená pole jsou narovnána [1, [2, 3]] -> [1, 2, 3]
- typ nested pro pole objektů, která jsou dotazována nezávisle

EXPILICITNÍ MAPOVÁNÍ

- vytvoření explicitního mapování
  - možno při vytváření indexu ale i později
  - klíč mappings
  - všechny pole mapování uzavřeny v klíči properties
    - platí i pro vnořené objekty
    - následuje pole a datový typ v klíči type

```
PUT /reviews
{
  "mappings": {
    "properties" : {
      "rating": { "type": "float" },
      "content": { "type": "text"},
      "product_id": { "type": "integer" },
      "author": {
        "properties": {
          "first_name": { "type": "text" },
          "last_name": { "type": "text" },
          "email": { "type": "keyword" }
        }
      }
    }
  }
}
```

```
{
  "acknowledged" : true,
  "shards_acknowledged" : true,
  "index" : "reviews"
}
```

```
PUT /reviews/_doc/1
{
  "rating": 5.0,
  "content": "Outstanding course! Bo really
    taught me a lot about Elasticsearch!",
  "product_id": 123,
  "author": {
    "first_name": "John",
    "last_name": "Doe",
    "email": "johndoe123@example.com"
  }
}
```

```
{
  "_index" : "reviews",
  "_type" : "_doc",
  "_id" : "1",
  "_version" : 1,
  "result" : "created",
  "_shards" : {
    "total" : 2,
    "successful" : 2,
    "failed" : 0
  },
  "_seq_no" : 0,
  "_primary_term" : 1
}
```

EXPILICITNÍ MAPOVÁNÍ

- vytvoření explicitního mapování
  - možnost i přes tečkovou notaci
  - lze využít i při prohledávání

```
PUT /reviews
{
  "mappings": {
    "properties": {
      "rating": { "type": "float" },
      "content": { "type": "text" },
      "product_id": { "type": "integer" },
      "author": {
        "properties": {
          "first_name": { "type": "text" },
          "last_name": { "type": "text" },
          "email": { "type": "keyword" }
        }
      }
    }
  }
}
```

```
PUT /reviews_dot_notation
{
  "mappings": {
    "properties": {
      "rating": { "type": "float" },
      "content": { "type": "text" },
      "product_id": { "type": "integer" },
      "author.first_name": { "type": "text" },
      "author.last_name": { "type": "text" },
      "author.email": { "type": "keyword" }
    }
  }
}
```

- vytvoření až po definování indexu
  - Mapping API
  - properties jako hlavní klíč

```
PUT /reviews/_mapping
{
  "properties": {
    "created_at": { "type": "date" }
  }
}
```

```
{
  "acknowledged" : true
}
```

EXPILICITNÍ MAPOVÁNÍ

- přečtení mapování
  - Mapping API
  - i pro jednotlivá pole
  - vnořená pole přes tečkovou notaci

```
GET /reviews/_mapping

GET /reviews/_mapping/field/content

GET /reviews/_mapping/filed/author.email
```

```
{
  "reviews" : {
    "mappings" : {
      "properties" : {
        "author" : {
          "properties" : {
            "email" : {
              "type" : "keyword"
            },
            "first_name" : {
              "type" : "text"
            },
            "last_name" : {
              "type" : "text"
            }
          }
        },
        "content" : {
          "type" : "text"
        },
        "product_id" : {
          "type" : "integer"
        },
        "rating" : {
          "type" : "float"
        }
      }
    }
  }
}
```

DATUMY

- specifikovány v jednom ze tří formátu
  - formátovaný řetězec podle ISO 8601
    - datum bez času i datum s časem
    - podpora vlastního formátování
  - milisekundy od epochy (long)
    - 1. 1. 1970
  - sekundy od epochy (int)
- datumy ukládány jako long hodnoty
  - v UTC
  - stejná konverze je provedena při vyhledávání

https://www.udemy.com/course/elasticsearch-complete-guide/

PARAMETRY MAPOVÁNÍ

- velké množství parametrů mapování
- některé významnější
  - format, properties, nested, coerce, doc_values, norms, index, null_value, copy_to
- obecně nelze aktualizovat mapování polí
  - bylo by značně problematické pro existující dokumenty
  - lze jen přidávat nové
    - nelze tedy mapování ani odstranit
  - výjimka jen pro pár vybraných parametrů
    - např. ignore_above, field alias
  - ➢ řešením je přeindexování dokumentů do nového indexu
    - ➢ Reindex API

VÍCENÁSOBNÉ MAPOVÁNÍ

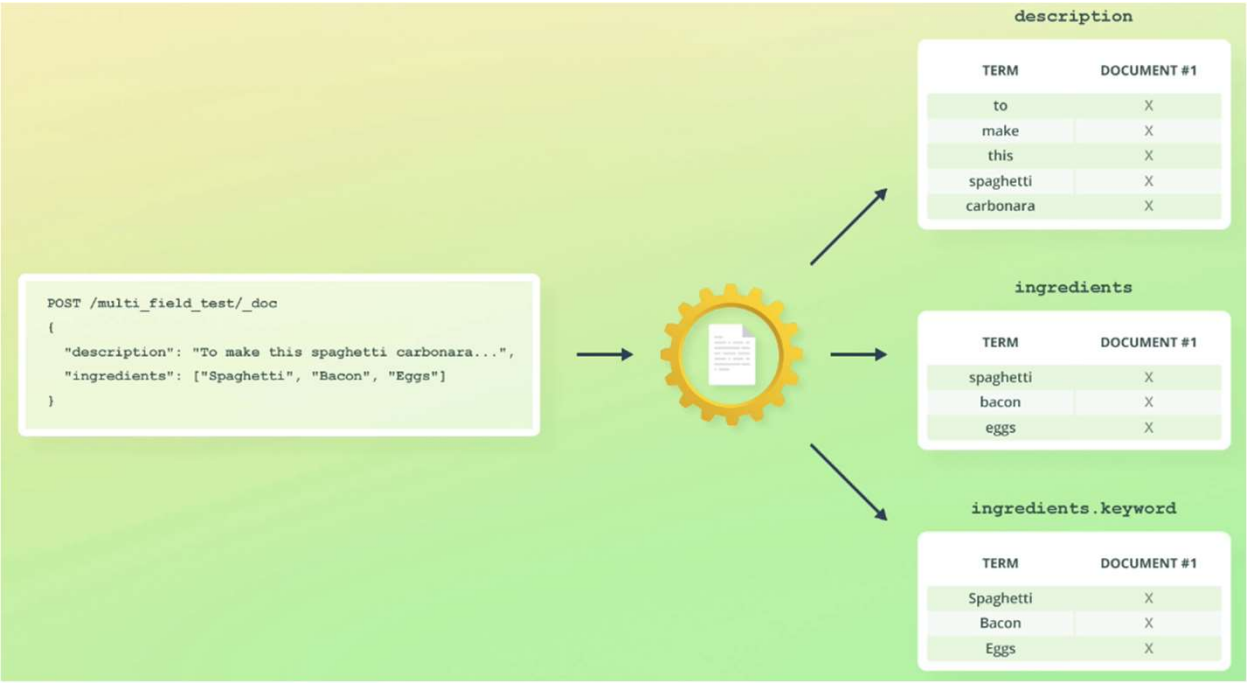
- na jedno pole je možné definovat více mapování
  - např. ingredience u receptu můžeme chtít vyhledávat fulltextově i jako klíčové slovo
  - klíč fields

```
PUT /multi_field_test
{
  "mappings": {
    "properties": {
      "description": {
        "type": "text"
      },
      "ingredients": {
        "type": "text",
        "fields": {
          "keyword": {
            "type": "keyword"
          }
        }
      }
    }
  }
}
```

```
POST /multi_field_test/_doc
{
  "description": "To make this spaghetti carbonara, you first need to...",
  "ingredients": ["Spaghetti", "Bacon", "Eggs"]
}
```

VÍCENÁSOBNÉ MAPOVÁNÍ

- textové pole bylo analyzováno a byly vytvořeny dva indexy
  - description a ingredients
- pole ingredients navíc analyzováno keyword analyzátorem
  - obsahuje nezměněné hodnoty



https://www.udemy.com/course/elasticsearch-complete-guide/

VÍCENÁSOBNÉ MAPOVÁNÍ

```
GET /multi_field_test/_search
{
  "query": {
    "match": {
      "ingredients": "Spaghetti"
    }
  }
}
```

```
{
  "took" : 638,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 1,
      "relation" : "eq"
    },
    "max_score" : 0.2876821,
    "hits" : [
      {
        "_index" : "multi_field_test",
        "_type" : "_doc",
        "_id" : "gPOT9XUBkaLlnoKaY8Zf",
        "_score" : 0.2876821,
        "_source" : {
          "description" : "To make this
          "ingredients" : [
            "Spaghetti",
            "Bacon",
            "Eggs"
          ]
        }
      }
    ]
  }
}
```

```
GET /multi_field_test/_search
{
  "query": {
    "term": {
      "ingredients.keyword":
        "Spaghetti"
    }
  }
}
```

```
{
  "took" : 185,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 1,
      "relation" : "eq"
    },
    "max_score" : 0.39556286,
    "hits" : [
      {
        "_index" : "multi_field_test",
        "_type" : "_doc",
        "_id" : "gPOT9XUBkaLlnoKaY8Zf",
        "_score" : 0.39556286,
        "_source" : {
          "description" : "To make this
          "ingredients" : [
            "Spaghetti",
            "Bacon",
            "Eggs"
          ]
        }
      }
    ]
  }
}
```

DYNAMICKÉ MAPOVÁNÍ

- usnadňuje práci s Elasticsearch
  - mapování vytváří sám Elasticsearch při prvním setkání s nenamapovaným polem dokumentu (vložení prvního dokumentu)

https://www.udemy.com/course/elasticsearch-complete-guide/

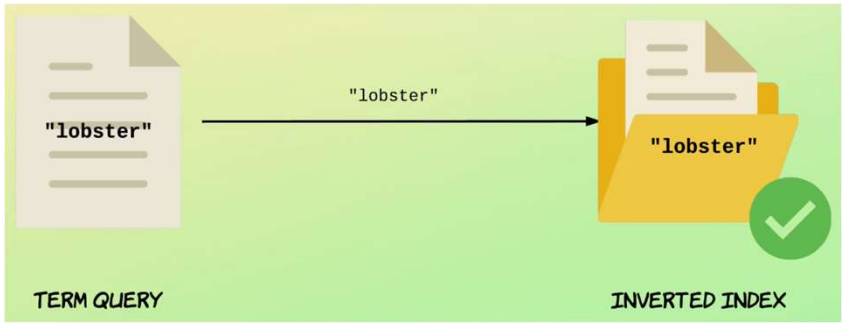
JSON	ELASTICSEARCH
string	One of the following: • `text` field with `keyword` mapping • `date` field • (`float` or `long` field)
integer	`long`
floating point number	`float`
boolean (`true` or `false`)	`boolean`
object	`object`
array	Depends on the first non-null value

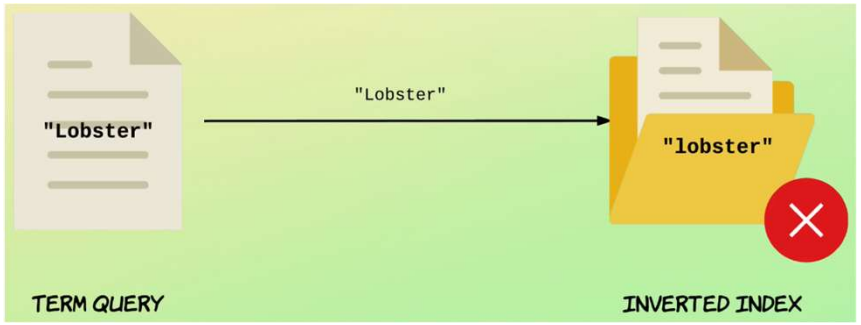
  - je možné konfigurovat i vypnout
  - ne vždy ideální
    - explicitní mapování (případně kombinace)

ČÁST III.:
ELASTICSEARCH
ZPĚT K PROHLEDÁVÁNÍ

ZPĚT K PROHLEDÁVÁNÍ

- dotazovací typy
  - dotazy na úrovni termínů (term-level queries)
    - vyhledávají přesnou hodnotu vůči invertovanému indexu
      - ne vůči přímo cílovému dokumentu
    - ➢ vyhledávaná hodnota není analyzována
    - vhodné např. pro datumy, čísla, statusy, …
      - obecně pro keyword pole
      - ne pro řetězce (text)





https://www.udemy.com/course/elasticsearch-complete-guide/

TERM-LEVEL QUERIES

- vyhledání podle výrazu
  - term query
  - vypsání produktů, které jsou v prodeji
    - pole is_active
- vyhledání podle více výrazů
  - terms query
  - pole hodnot
  - dokument je výsledkem, pokud obsahuje alespoň jednu z hodnot
  - vyhledání polévek a dortů podle tagu
    - pole tags.keyword

```
GET /products/_search
{
  "query": {
    "term": {
      "is_active": true
    }
  }
}

GET /products/_search
{
  "query": {
    "term": {
      "is_active": {
        "value": true
      }
    }
  }
}
```

```
{
  "took" : 29,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 487,
      "relation" : "eq"
    },
    "max_score" : 0.7194644,
    "hits" : [
      {
        "_index" : "products",
        "_type" : "_doc",
```

```
GET /products/_search
{
  "query": {
    "terms": {
      "tags.keyword": [
        "Soup",
        "Cake"
      ]
    }
  }
}
```

TERM-LEVEL QUERIES

- vyhledání podle id
  - vypsání dokumentů s ID 1-3

```
GET /products/_search
{
  "query": {
    "ids": {
      "values": [1,2,3]
    }
  }
}
```

```
{
  "took" : 3,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 3,
      "relation" : "eq"
    },
    "max_score" : 1.0,
```

- vyhledání podle rozsahu hodnot
  - vypsání produktů, které docházejí
    - in_stock 1 až 5

```
GET /products/_search
{
  "query": {
    "range": {
      "in_stock": {
        "gte": 1,
        "lte": 5
      }
    }
  }
}
```

  - vypsání produktů, které byly přidány v roce 2010

```
GET /products/_search
{
  "query": {
    "range": {
      "created": {
        "gte": "2010/01/01",
        "lte": "2010/12/31"
      }
    }
  }
}
```

- vyhledání produktů s alespoň jedním tagem
  - exists query

```
GET /products/_search
{
  "query": {
    "exists": {
      "field": "tags"
    }
  }
}
```

TERM-LEVEL QUERIES

- vyhledání podle prefixu
  - tagy začínající na Vege
    - Vegetables

```
GET /products/_search
{
  "query": {
    "prefix": {
      "tags.keyword": "Vege"
    }
  }
}
```

- vyhledání s využitím žolíků (wildcards)
  - * – sekvence znaků (včetně nulové sekvence)
  - ? – jeden znak

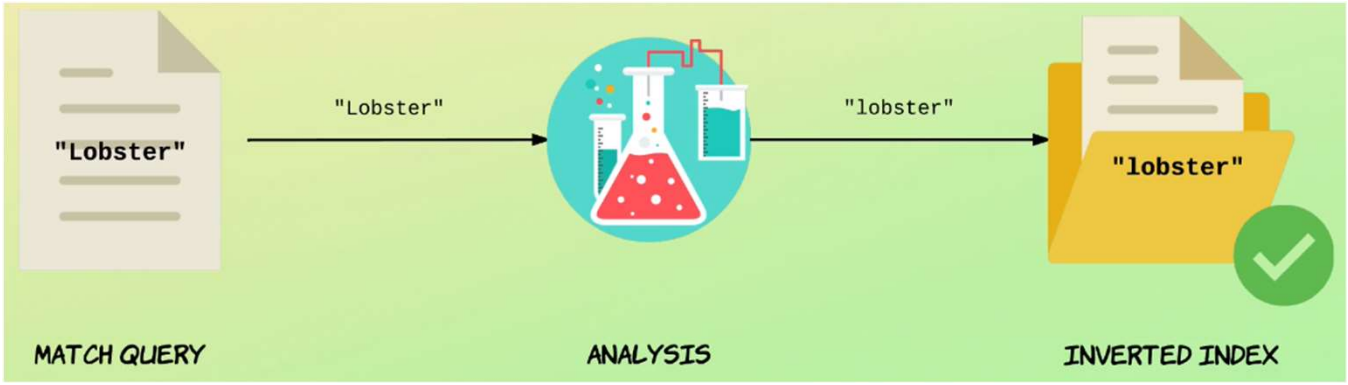
```
GET /products/_search
{
  "query": {
    "wildcard": {
      "tags.keyword": "V?g*ble"
    }
  }
}
```

- vyhledání s využitím regulárních výrazů
  - ne všechna funkcionalita je ale podporována
  - regexp query

```
GET /products/_search
{
  "query": {
    "regexp": {
      "tags.keyword": "Veget[a-zA-Z]?ble"
    }
  }
}
```

ZPĚT K PROHLEDÁVÁNÍ

- dotazovací typy
  - fulltextové dotazy (full-text queries)
    - vyhledávaná hodnota je analyzována stejným analyzérem jako invertovaný index
    - ➢ je možné najít jen hodnoty v invertovaném indexu
    - vhodné pro fulltextové vyhledávání
      - řetězce



https://www.udemy.com/course/elasticsearch-complete-guide/

FULL-TEXT QUERIES

- match query
  - vhodné např. pro dotazy zadané uživatelem (Google)
  - vyhledání názvů receptů podle zadané věty
    - hledá názvy obsahující výrazy ze vstupní věty
      - nemusí obsahovat všechny výrazy
    - probíhá analýza a pro výsledky počítáno skóre relevance

```
GET /recipes/_search
{
  "query": {
    "match": {
      "title": "Recipes with pasta or spaghetti"
    }
  }
}
```

```
"hits" : [
  {
    "_index" : "recipes",
    "_type" : "_doc",
    "_id" : "11",
    "_score" : 5.130475,
    "_source" : {
      "title" : "Spaghetti Puttanesca (Pasta or Spaghetti With Capers, Olives, and Anchovies)"
```

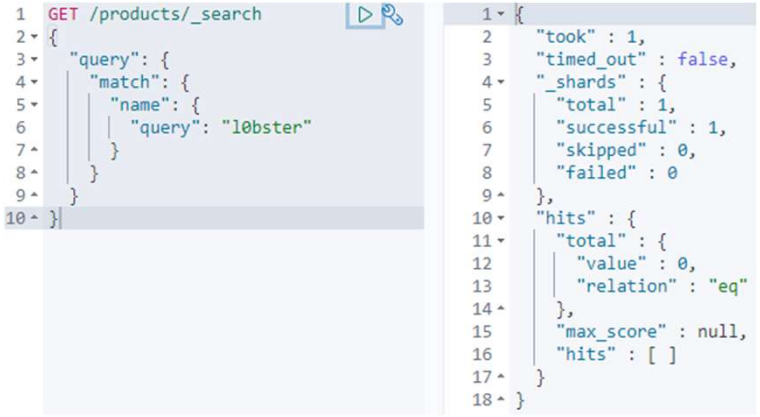
    - možnost definovat, aby název musel obsahoval všechny výrazy ze vstupní věty

```
GET /recipes/_search
{
  "query": {
    "match": {
      "title": {
        "query": "Recipes with pasta or spaghetti",
        "operator": "and"
      }
    }
  }
}
```

```
{
  "took" : 1,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 0,
```

FULL-TEXT QUERIES

- match query
  - vhodné např. pro dotazy zadané uživatelem (Google)
  - a co když uživatel udělá překlep?
    - řeší parametr fuzziness
    - počítá Levenštejnovu vzdálenost pro měření editační vzdálenosti mezi hledaným výrazem (na úrovni tokenů) a invertovanými indexy
    - pokud je práh menší než zadaná hodnota fuzziness, jedná se o shodu



FULL-TEXT QUERIES

- match_phrase query
  - u match query nezáleželo na pořadí výrazů v dotazu
  - ➢ match_phrase query umožňuje vyhledávání přesných frází
  - také umožňuje proximity search
    - hledání v blízkosti
    - parametr slop
    - určuje míru flexibility
    - udává počet povolených změn
    - ➢ s hodnotou dvě by obě zobrazená vyhledávání byla úspěšná

```
GET /recipes/_search
{
  "query": {
    "match_phrase": {
      "title": "puttanesca spaghetti"
    }
  }
}
```

```
{
  "took" : 1,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 0,
      "relation" : "eq"
    },
    "max_score" : null,
    "hits" : [ ]
  }
}
```

```
GET /recipes/_search
{
  "query": {
    "match_phrase": {
      "title": "spaghetti puttanesca"
    }
  }
}
```

```
"hits" : [
  {
    "_index" : "recipes",
    "_type" : "_doc",
    "_id" : "11",
    "_score" : 3.5470161,
    "_source" : {
      "title" : "Spaghetti Puttanesca (Pasta or Spaghetti With Capers, Olives, and Anchovies)",
```

FULL-TEXT QUERIES

- multi_match query
  - vyhledávání přes více polí

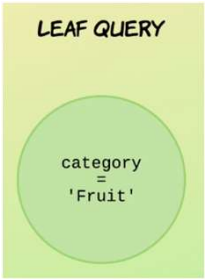
```
GET /recipes/_search
{
  "query": {
    "multi_match": {
      "query": "pasta",
      "fields": ["title", "description"]
    }
  }
}
```

```
"_source" : {
  "title" : "Vegan Carbonara Pasta",
  "description" : "Carbonara may be one of the
    most difficult recipes to vegan-ify, since
    every major ingredient in the sauce is off
    -limits. But by eating lots of the real
    deal and getting mighty crafty with an
    array of unlikely ingredients, I managed to
    create a vegan carbonara that captures the
    essence of the original like no other: It's
    silky and rich, unctuous, and studded with
    meaty bits, with the sharp, lactic tang of
    Pecorino Romano (but, of course, no actual
    Pecorino Romano).",
  "preparation_time_minutes" : 30,
```

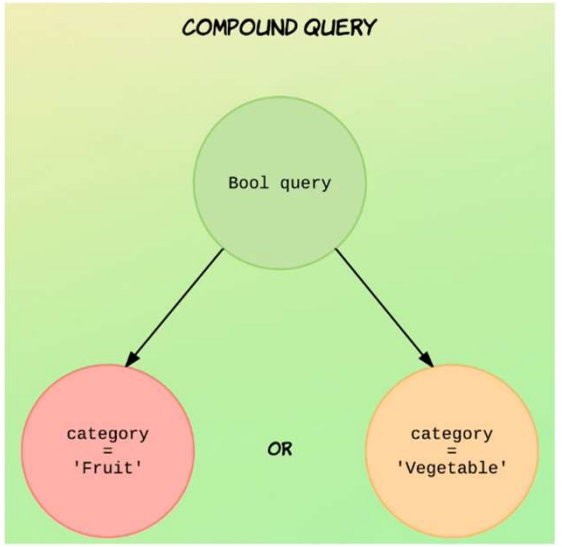
```
"_source" : {
  "title" : "Spaghetti Aglio e Olio Recipe",
  "description" : "One of the most basic pasta
    sauces, aglio e olio uses just garlic and
    olive oil (and maybe a pinch of red pepper
    flakes for heat). It sounds too simple to
    be good, but it's among the best.",
  "preparation_time_minutes" : 10,
```

DOTAZY S BOOLEOVSKOU LOGIKOU

- leaf queries
  - dotazy, se kterými jsme pracovali doteď
  - nejjednodušší dotazy provádějící jen jednu operaci
  - vyhledávají hodnotu v daném poli
  - např. term nebo match query
- compound queries
  - tvoří složitější dotazy zaobalením leaf nebo dalších compound queries
  - např. dvě leaf query spojené přes bool query
  - ➢ využívají booleovskou logiku



https://www.udemy.com/course/elasticsearch-complete-guide/



DOTAZY S BOOLEOVSKOU LOGIKOU

- bool query
  - umožňuje použití booleovské logiky v dotazování
  - dotazovací i filtrovací kontext
  - podobné WHERE v SQL
    - ale navíc i vyhodnotí skóre relevance
  - vyhledání receptů s ingrediencí parmezán a dobou přípravy maximálně 15 minut
    - klíč must specifikuje podmínky, které musí být splněny
    - obě položky použity k výpočtu skóre relevance
    - to ale nedává u doby přípravy smysl
      - otázka ano / ne
      - ➢ filtrovací kontext
      - klíč filter

```
GET /recipes/_search
{
  "query": {
    "bool": {
      "must": [
        {
          "match": {
            "ingredients.name": "parmesan"
          }
        },
        {
          "range": {
            "preparation_time_minutes": {
              "lte": 15
            }
          }
        }
      ]
    }
  }
}
```

DOTAZY S BOOLEOVSKOU LOGIKOU

- bool query
  - a co když nechci tuňáka?
    - klíč must_not
    - provedeno ve filtrovacím kontextu
      - není skórováno
  - a co když mám hodně rád petržel?
    - klíč should
    - petržel není podmínkou
    - ➢ ale pokud je součástí, zvyšuje skóre
    - ➢ preference
    - pokud není definován ani must ani filter klíč, alespoň jedna should podmínka musí platit

```
GET /recipes/_search
{
  "query": {
    "bool": {
      "must": [
        {
          "match": {
            "ingredients.name": "parmesan"
          }
        }
      ],
      "must_not": [
        {
          "match": {
            "ingredients.name": "tuna"
          }
        }
      ],
      "should": [
        {
          "match": {
            "ingredients.name": "parsley"
          }
        }
      ],
      "filter": [
        {
          "range": {
            "preparation_time_minutes": {
              "lte": 15
            }
          }
        }
      ]
    }
  }
}
```

SPOJOVÁNÍ DOTAZŮ

- ukládání dat v Elasticsearch odpovídá NoSQL databázím
  - ne relačním databázím

➢ dokumenty jsou denormalizovány
  - ➢ rychlost vyhledávání na úkor nároků na skladování
  - Elasticsearch ale většinou není používán jako primární úložiště
  - ➢ výkon > úložiště

➢ omezená podpora spojování (join)
  - jen jednoduché spojování
  - a velmi neefektivní

https://www.udemy.com/course/elasticsearch-complete-guide/

SPOJOVÁNÍ DOTAZŮ

- nested query
  - ve spojení s datovým typem nested
    - používaný pro pole objektů, kde je potřeba zachovat vztahy mezi vlastnostmi objektů
      - pokud není typ nested, vlastnosti objektů jsou smíchány při ukládání v Elasticsearch
    - vyhledání oddělení, kde pracují stážistky

```
PUT /departments
{
  "mappings": {
    "properties": {
      "name": {
        "type": "text"
      },
      "employees": {
        "type": "nested"
      }
    }
  }
}
```

```
PUT /departments/_doc/1
{
  "name": "Development",
  "employees": [
    {
      "name": "Eric Green",
      "age": 39,
      "gender": "M",
      "position": "Big Data Specialist"
    },
```

```
PUT /departments/_doc/2
{
  "name": "HR & Marketing",
  "employees": [
    {
      "name": "Patricia Lewis",
      "age": 42,
      "gender": "F",
      "position": "Senior Marketing
        Manager"
    },
```

```
GET /departments/_search
{
  "query": {
    "nested": {
      "path": "employees",
      "query": {
        "bool": {
          "must": [
            {
              "match": {
                "employees.position": "intern"
              }
            },
            {
              "term": {
                "employees.gender.keyword": {
                  "value": "F"
                }
              }
            }
          ]
        }
      }
    }
  }
}
```

```
{
    "persons": [
        { "name": "Bo Andersen", "age": 28 },
        { "name": "Fogell McLovin", "age": 20 }
    ]
}
```

is stored as

```
{
    "persons.name": [ "Bo Andersen", "Fogell McLovin" ],
    "persons.age": [ 20, 28 ]
}
```

https://www.udemy.com/course/elasticsearch-complete-guide/

SPOJOVÁNÍ DOTAZŮ

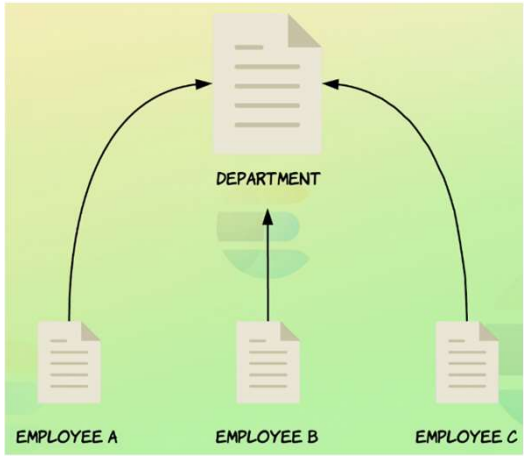
- nested inner hits
  - předchozí dotaz vrací oddělení, která vyhovují dotazu, ale nevrací konkrétní zaměstnance

```
GET /departments/_search
{
  "query": {
    "nested": {
      "path": "employees",
      "inner_hits": {},
      "query": {
        "bool": {
          "must": [
            {
              "match": {
                "employees.position": "intern"
              }
            },
            {
              "term": {
                "employees.gender.keyword": {
                  "value": "F"
                }
              }
            }
          ]
        }
      }
    }
  }
}
```

```
"inner_hits" : {
  "employees" : {
    "hits" : {
      "total" : {
        "value" : 1,
        "relation" : "eq"
      },
      "max_score" : 2.3905568,
      "hits" : [
        {
          "_index" : "departments",
          "_type" : "_doc",
          "_id" : "1",
          "_nested" : {
            "field" : "employees",
            "offset" : 3
          },
          "_score" : 2.3905568,
          "_source" : {
            "gender" : "F",
            "name" : "Julie Powell",
            "position" : "Intern",
            "age" : 26
          }
        }
      ]
    }
```

SPOJOVÁNÍ DOTAZŮ

- v minulém případě zaměstnanci součástí dokumentů oddělení
  - složitější aktualizace zaměstnanců
  - co když by byl dokumentem i zaměstnanec?
  - ➢ mapování vztahů dokumentů
    - ➢ join_field
    - ➢ typ join
    - oddělení je rodičem zaměstnance
      - při přidávání dokumentu je potřeba specifikovat vztah
    - možnost i víceúrovňových vztahů (vnořených)



https://www.udemy.com/course/elasticsearch-complete-guide/

```
PUT /department/_mapping
{
  "properties": {
    "join_field": {
      "type": "join",
      "relations": {
        "department": "employee"
      }
    }
  }
}
```

```
PUT /department/_doc/1
{
  "name": "Development",
  "join_field": "department"
}

PUT /department/_doc/2
{
  "name": "Marketing",
  "join_field": "department"
}
```

```
PUT /department/_doc/3?routing=1
{
  "name": "Bo Andersen",
  "age": 28,
  "gender": "M",
  "join_field": {
    "name": "employee",
    "parent": 1
  }
}
```

```
PUT /department/_doc/4?routing=2
{
  "name": "John Doe",
  "age": 44,
  "gender": "M",
  "join_field": {
    "name": "employee",
    "parent": 2
  }
}
```

SPOJOVÁNÍ DOTAZŮ

- vyhledání potomků na základě id rodiče
  - parent_id
- vyhledání potomků podle rodiče
  - has_parent
  - vyhledání zaměstnanců ve vývoji
- vyhledání rodiče podle potomků
  - has_child
  - vyhledání oddělení se zaměstnancem starším 50 let
- možnost použití inner_hits jako u klíče nested
- obecně velmi drahé a pomalé
  - denormalizace dokumentů je preferovaná

```
GET /department/_search
{
  "query": {
    "parent_id": {
      "type": "employee",
      "id": 1
    }
  }
}
```

```
GET /department/_search
{
  "query": {
    "has_parent": {
      "parent_type": "department",
      "query": {
        "term": {
          "name.keyword": "Development"
        }
      }
    }
  }
}
```

```
GET /department/_search
{
  "query": {
    "has_child": {
      "type": "employee",
      "query": {
        "range": {
          "age": {
            "gte": 50
          }
        }
      }
    }
  }
}
```

SPRÁVA VÝSLEDKŮ PROHLEDÁVÁNÍ

- formátované zobrazení
  - pretty
  - Kibana obstarává automaticky

```
GET /recipe/_search?pretty
{
    "query": {
      "match": { "title": "pasta" }
    }
}
```

- filtrování vrácených dokumentů
  - pole _source
    - v základním nastavení vrací celé dokumenty
    - nastavením na hodnotu false zůstane pole prázdné
    - nastavením na konkrétní název pole vrací dané pole

➢ snížení množství přenášených dat

```
GET /recipes/_search
{
  "_source": false,
  "query": {
    "match": { "title": "pasta" }
  }
}
```

```
{
  "took" : 1,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 9,
      "relation" : "eq"
    },
    "max_score" : 1.0835493,
    "hits" : [
      {
        "_index" : "recipes",
        "_type" : "_doc",
        "_id" : "8",
        "_score" : 1.0835493
      },
      {
        "_index" : "recipes",
        "_type" : "_doc",
        "_id" : "18",
        "_score" : 0.8755694
      },
```

```
GET /recipes/_search
{
  "_source": "created",
  "query": {
    "match": { "title": "pasta" }
  }
}
```

```
  "hits" : [
    {
      "_index" : "recipes",
      "_type" : "_doc",
      "_id" : "8",
      "_score" : 1.0835493,
      "_source" : {
        "created" : "2002/01/04"
      }
    },
```

```
GET /recipes/_search
{
  "_source": ["title", "created"],
  "query": {
    "match": { "title": "pasta" }
  }
}
```

```
GET /recipes/_search
{
  "_source": ["ingredients.*", "servings"],
  "query": {
    "match": { "title": "pasta" }
  }
}
```

SPRÁVA VÝSLEDKŮ PROHLEDÁVÁNÍ

- filtrování vrácených dokumentů
  - pole _source
    - možnost definovat i prvky, které nechceme vrátit
- maximální počet vrácených shod
  - parametr size
  - v základu 10
- zobrazení dalších výsledků
  - např. druhá strana vyhledávání
  - parametr offset
  - v základu 0

```
GET /recipes/_search
{
  "_source": {
    "includes": "ingredients.*",
    "excludes": "ingredients.name"
  },
  "query": {
    "match": { "title": "pasta" }
  }
}
```

```
GET /recipes/_search
{
  "_source": false,
  "size": 2,
  "query": {
    "match": {
      "title": "pasta"
    }
  }
}
```

```
{
  "took" : 1,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 9,
      "relation" : "eq"
    },
    "max_score" : 1.0835493,
    "hits" : [
      {
        "_index" : "recipes",
        "_type" : "_doc",
        "_id" : "8",
        "_score" : 1.0835493
      },
      {
        "_index" : "recipes",
        "_type" : "_doc",
        "_id" : "18",
        "_score" : 0.8755694
      }
    ]
  }
}
```

```
GET /recipes/_search
{
  "_source": false,
  "size": 2,
  "from": 2,
  "query": {
    "match": {
      "title": "pasta"
    }
  }
}
```

SPRÁVA VÝSLEDKŮ PROHLEDÁVÁNÍ

- řazení výsledků
  - parametr sort
  - v základu vzestupně
  - recepty podle doby přípravy od nejrychlejších
  - recepty od nejnovějších
    - potřeba definovat sestupné pořadí
  - řazení podle více polí

```
GET /recipes/_search
{
  "_source": [ "preparation_time_minutes", "created" ],
  "query": {
    "match_all": {}
  },
  "sort": [
    { "preparation_time_minutes": "asc" },
    { "created": "desc" }
  ]
}
```

  - řazení podle vícehodnotových polí
    - mód pro minimum, maximum, průměr, sumu, …

```
GET /recipes/_search
{
  "_source": false,
  "query": {
    "match_all": {}
  },
  "sort": [
    "preparation_time_minutes"
  ]
}
```

```
GET /recipes/_search
{
  "_source": "created",
  "query": {
    "match_all": {}
  },
  "sort": [
    { "created": "desc" }
  ]
}
```

```
GET /recipes/_search
{
  "_source": "ratings",
  "query": {
    "match_all": {}
  },
  "sort": [
    {
      "ratings": {
        "order": "desc",
        "mode": "avg"
      }
    }
  ]
}
```

SPRÁVA VÝSLEDKŮ PROHLEDÁVÁNÍ

- zvýraznění výsledků
  - parametr highlight
    - spousta možností konfigurace
    - vrací fragmenty

```
PUT /highlighting/_doc/1
{
  "description": "Let me tell you a
    story about Elasticsearch. It's a
    full-text search engine that is
    built on Apache Lucene. It's
    really easy to use, but also
    packs lots of advanced features
    that you can use to tweak its
    searching capabilities. Lots of
    well-known and established
    companies use Elasticsearch, and
    so should you!"
}
```

```
GET /highlighting/_search
{
  "_source": false,
  "query": {
    "match": { "description":
      "Elasticsearch story" }
  },
  "highlight": {
    "fields": {
      "description" : {}
    }
  }
}
```

```
{
  "took" : 208,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 1,
      "relation" : "eq"
    },
    "max_score" : 0.68324494,
    "hits" : [
      {
        "_index" : "highlighting",
        "_type" : "_doc",
        "_id" : "1",
        "_score" : 0.68324494,
        "highlight" : {
          "description" : [
            "Let me tell you a <em>story</em> about <em>Elasticsearch</em>.",
            "Lots of well-known and established companies use <em>Elasticsearch</em>, and so should you!"
          ]
        }
      }
    ]
  }
}
```

A PŘÍŠTĚ?

- Elasticsearch
  - agregace
  - shardování a replikace
  - Elastic Stack

Děkuji za pozornost.
Otázky?