

10 - Full-text searches

- Busca de dados não-estruturados

OVERVIEW

- Relevância relacionado a precisão e recall
 - Precisão:
 - Densidade de bons resultados
 - Recall:
 - Completude da busca
 - Trade-off entre as duas medidas

THE MATCH_ALL QUERY

- Busca todos os documentos do index
- Aceita boost ou nenhum argumento

Listing 10.2 Boosting the query with a predefined score

```
GET books/_search
{
  "query": {
    "match_all": {
      "boost": 2
    }
  }
}
```

◦

The match_all query sets a score of 2 for all returned documents.

THE MATCH_NONE QUERY

- Retorna nenhum resultado
- Útil aliado a alguma lógica na aplicação
 - Por exemplo, para retornar nada durante uma manutenção, em vez de dar erro.

THE MATCH QUERY

- O mais comum
 - Podemos procurar por um texto direto no field
 - Ou então passar um key:value com os parâmetros adicionais ao campo.

```
GET books/_search
{
  "query": {
    "match": {
      "FIELD": {
        "query": "<SEARCH TEXT>",
        "<parameter>": "<MY_PARAM>",
      }
    }
  }
}
```

◦

Declares FIELD as an object with additional parameters

The query attribute holds the search text.

The parameter (analyzer, operator, prefix_length, fuzziness, etc.) expects a value to be set.

- Termos procurados são analyzed.
 - A princípio, pelo menos analyzer da indexação.
 - Por default, passa por um token filter de lowercase.
 - Podemos passar várias palavras
 - Termos são procurados com operador OR
 - Embora mais termos traga score maior.
 - Pode ser alterado para AND
 - Também podemos manter o OR mas estabelecer um número mínimo de palavras para dar match com o parâmetro *"minimum_should_match"*
 - Parâmetro fuzziness (distância de levenshtein) também é possível de configurar, valores 0, 1, 2 ou AUTO.

THE MATCH_PHRASE QUERY

- Retorna o match da phrase exata, na mesma sequência
 - Diferente do Match com AND
- Parâmetro slop permite flexibilizar a busca e settar um número de palavras que permite-se pular na busca.
 - Número inteiro, número de palavras que permite-se pular e ainda retornar resultados.

THE MATCH_PHRASE_PREFIX QUERY

- Igual match_phrase mas apenas o prefixo, não o campo inteiro.
- e.g. "Concepts and found" retorna resultados tipo "Concepts and foundations"
- Também suporta slop

THE MULTI_MATCH QUERY

- Procura um string em vários fields ao mesmo tempo.
 - Passamos um array de fields

Listing 10.14 Searching multiple fields using multi_match

```
GET books/_search
{
  "_source": false,
  "query": {
    "multi_match": {
      "query": "Java",
      "fields": [
        "title",
        "synopsis",
        "tags"
      ]
    }
  },
  "highlight": {
    "fields": {
      "title": {},
      "tags": {}
    }
  }
}
```

Suppresses the source document from appearing in the results

Specifies the multi_match query

Defines the search criterion as the word "Java"

Searches across multiple fields provided in an array

Highlights the matches returned in the results

- Tem várias formas de scoring:
 - Parâmetro "type"
 - Default é "best_fields", score é o field que tem o melhor score, ignorando os demais.
 - outros: cross_fields, most_fields, phrase, phrase prefix
- Usa query tipo dis_max por trás
 - disjunction max query
 - Separa cada field para uma query separada

Listing 10.16 Disjunction max (dis_max) query in use

```
GET books/_search
{
  "_source": false,
  "query": {
    "dis_max": {
      "queries": [
        { "match": { "title": "Design Patterns" } },
        { "match": { "synopsis": "Design Patterns" } }
      ]
    }
  }
}
```

Specifies the dis_max query type

Defines the set of queries to include in a dis_max query block

Specifies a match query

- Podemos definir tie_breakers
 - Usa score dos outros campos não-best-field para o desempate de relevância
 - float que multiplica score dos não-best-fields
- Podemos dar um boost no score de certos campos
 - e.g. aumentar score para "título" para priorizar sobre outros campos.

THE QUERY_STRING QUERY

- Query com operadores booleanos.
- Feita como a busca direto por URI

Listing 10.20 Creating a query string with operators

```
GET books/_search
{
  "query": {
    "query_string": {
      "query": "author:Bert AND edition:2 AND release_date>=2000-01-01"
    }
  }
}
```

Specifies the query_string query

Provides the search criteria in the query

-
- Útil para criar templates para query do usuário.
- Também podemos definir se os termos são buscados como OR ou AND
- Ou então buscar por frases.
 - Permite slop também

FUZZY QUERIES

- Utilizamos "~" no sufixo do termo da query para definir fuzzy

THE SIMPLE_QUERY_STRING QUERY

- versão simplificada de query_string
- permite só alguns operadores:
 - e.g. "Java + Cay" busca por Java AND Cay
- Não retorna erros mesmo com erro de sintaxe

Summary

- Elasticsearch is big on searching unstructured data using full-text queries. Full-text queries yield relevancy, meaning the documents matched and returned have a positive relevancy score.
- Elasticsearch provides a `_search` API for querying purposes.
- Several kinds of `match` queries work for various use cases when searching full text with relevance. The most common query is the `match` query.
- The `match` query searches on text fields for search criteria and scores documents using the relevance algorithm.
- Match all (`match_all`) queries search across all indexes and do not require a body.
- To search for a phrase, we use a `match_phrase` query or its variant, `match_phrase_prefix`. Both types of queries let us search for specific words in a defined order. Additionally, we can use the `phrase_slop` parameter if the phrase is missing words.
- Searching for user criteria across multiple fields is enabled by using a `multi_match` query.
- A query string (`query_string`) query uses logical operators such as AND, OR, and NOT. However, the `query_string` query is strict on syntax, so we receive an exception if the input syntax is incorrect.
- If we need Elasticsearch to be less strict about the query string syntax, then instead of using a `query_string` query, we can use a `simple_query_string` query. With that query type, all syntactical errors are suppressed by the engine.