TREINAMENTO ELASTICSEARCH

Apostila com os comandos das vídeoaulas
Uso do Dev Tools

Utilização dos comandos
Apostila criada por Grimaldo Oliveira

Curso ELASTICSEARCH
Prof. Grimaldo Oliveira
grimaldo_lopes@hotmail.com

Conheça o Professor Grimaldo Oliveira

Sou professor das pós-graduações das universidades **UNIFACS, CATÓLICA DO SALVADOR** e **ISL Wyden**. **Mestre** pela **Universidade do Estado da Bahia** (UNEB) no Curso de Mestrado Profissional Gestão e Tecnologias Aplicadas à Educação (GESTEC). Possuo Especialização em Análise de Sistemas pela Faculdade Visconde de Cairu e **Bacharelado** em **Estatística** pela Universidade Federal da Bahia. Atuo profissionalmente como consultor há mais de **15 anos nas áreas de Data Warehouse, Mineração de Dados, Ferramentas de Tomada de Decisão e Estatística**.

BI PRO

Idealizador do treinamento online **BI PRO** com + de 10 módulos contendo todas as disciplinas para formação completa na área de dados. Quem participa do **BI PRO** tem acesso gratuito: todos os meus cursos de dados da Udemy, + ebook BI COMO DEVE SER - O Guia Definitivo, espaço de mentoria para retirada de dúvidas, respostas das atividades. Acesse www.bipro.com.br

Autor do eBook **BI COMO DEVE SER - O Guia Definitivo,** com ele você poderá entender os conceitos e técnicas utilizados para o desenvolvimento de uma solução BI, tudo isso de forma objetiva e prática, com linguagem acessível tanto para técnicos quanto gestores e analista de negócio. Acesse www.bicomodeveser.com.br

Site de **cupons** do prof. Grimaldo, com desconto de todos os seus cursos de dados da Udemy, atualizado diariamente com diversas promoções, incluindo cursos gratuitos. Acesse https://is.gd/CUPOMCURSOSPROFGRIMALDO

Curso ELASTICSEARCH
Prof. Grimaldo Oliveira
grimaldo_lopes@hotmail.com

Idealizador do Blog **BI COM VATAPÁ** reúne informações diversas sobre a área de dados com detalhes sobre o mundo de Business Intelligence, Big Data, Ciência de dados, Mineração de dados e muitos outros. Acesse http://bicomvatapa.blogspot.com/

Aula – comandos

Esta apostila contém os comandos utilizados no DEV TOOL para interação com o ELASTICSEARCH

- Consultando todos os registros cujo o campo (term) **Gender** é igual a M.

```
GET /financeiro/_search
{
    "query": {
        "match": {
            "Gender": {
            "query": "M"
            }
        }
    }
}
```

- Quantidade de registros importados pelo logstash.

```
GET /financeiro/_count
```

- Pesquisa pelas expressões Blue e Graduate no campo (term) **message**.

```
GET financeiro/_search
    {
      "query": {
        "match": {
          "message": {
            "query": "Blue Graduate",
            "operator": "and"
          }
        }
      }
    }
```

- Busca em mais de um campo (term) **message** e **Gender** com a expressão $120K.

```
GET financeiro/_search
    {
      "query": {
        "multi_match": {
          "query": "$120K",
          "fields": [
            "message",
            "Gender"
          ]
        }
      }
    }
```

- Pesquisa em um campo (term) **message** mesmo se o conteúdo estiver digitado errado, por exemplo a expressão Singre para que encontremos as variações da expressão (Single).

```
GET financeiro/_search
    {
      "_source": "message",
      "query": {
        "match": {
          "message": {
            "query" : "Singre",
            "fuzziness": "auto"
          }
        }
      }
    }
```

• Pesquisando todos os registros cujo **Education_Level** é Graduate e **Gender** é igual a M e **Card_Category** não é Blue.

```
GET /financeiro/_search
{
  "query": {
    "bool": {
      "must": [
        {
          "match": {
            "Education_Level": "Graduate"
          }
        }
      ],
      "should": [
        {
          "match": {
            "Gender": "M"
          }
        }
      ],
      "must_not": [
        {
          "match": {
            "Card_Category": "Blue"
          }
        }
      ]
    }
  }
}
```

- Pesquisa se o campo **Avg_Open_To_Buy** tem um valor maior que 5550.

gte – maior do que igual a

gt – maior do que

Lte – menos do que igual a

lt – menos do que

```
POST /financeiro/_search
{
   "query":{
      "range":{
         "Avg_Open_To_Buy":{
            "gte":5550
         }
      }
   }
}
```

- Deleta registro onde **CLIENTNUM** é igual a `719942883`. Depois pesquisa-se para comprovar a eliminação.

```
POST financeiro/_delete_by_query
{
  "query": {
    "match": {
      "CLIENTNUM": "719942883"
    }
  }
}

POST /financeiro/_search
{
   "query":{
      "term":{"CLIENTNUM":"719942883"}
   }
}
```

- Query em SQL para exibição dos campos do índice.

```
POST /_sql?format=txt
{
  "query":"DESCRIBE financeiro"
}
```

- Query em SQL para busca de todos os registros.

```
POST /_sql?format=txt
{
  "query": "SELECT * FROM financeiro"
}
```

- Query em SQL para verificar a quantidade de registros com a situação de **Card_Category** igual a Blue.

```
POST /_sql?format=txt
{
  "query": "SELECT count(*) FROM financeiro where
Card_Category='Blue'"
}
```

- Query em formato SQL que retorna os 20 primeiros registros de acordo com a consulta.

```
POST /_sql?format=json
{
  "query": "SELECT * FROM financeiro where
Card_Category='Silver'",
  "fetch_size": 20,
  "columnar": true
}
```

- Query em SQL que retorna alguns campos selecionados em formato **txt**.

```
POST /_sql?format=txt
{
  "query": "SELECT  CLIENTNUM, Customer_Age, Gender FROM
financeiro where Customer_Age > 40 ORDER BY Customer_Age"
}
```

- Query em SQL que retorna alguns campos selecionados em formato **csv**.

```
POST /_sql?format=csv
{
  "query": "SELECT  CLIENTNUM, Customer_Age, Gender FROM
financeiro where Customer_Age > 40 ORDER BY Customer_Age"
}
```

- Query em SQL que retorna alguns campos selecionados em formato **json**.

```
POST /_sql?format=json
{
  "query": "SELECT  CLIENTNUM, Customer_Age, Gender FROM
financeiro where Customer_Age > 40 ORDER BY Customer_Age"
}
```

- Query em SQL e com a linguagem padrão do Elasticsearch para listar os registros que possuem **Total_Trans_Ct** entre 100 e 200, exibindo apenas os primeiros 5 registros (fetch_size 5)

```
POST /_sql?format=txt
{
  "query": "SELECT  CLIENTNUM, Customer_Age,
Gender,Total_Trans_Ct FROM financeiro",
  "filter": {
    "range": {
      "Total_Trans_Ct": {
        "gte" : 100,
        "lte" : 200
      }
    }
  },
  "fetch_size": 5
}
```

- Query em SQL para a passagem de parâmetros, neste caso são passados dois parâmetros **Customer_Age** e **Gender**.

```
POST /_sql?format=txt
{
    "query": "SELECT  CLIENTNUM, Customer_Age,
Gender,Total_Trans_Ct FROM financeiro where Customer_Age = ?
and Gender = ?",
    "params": [33, "M"]
}
```

- Deleta o cabeçalho do index. Primeiro pesquisa um campo que exiba a situação do cabeçalho e depois executa a query de delete.

```
POST /_sql?format=txt
{
  "query": "SELECT CLIENTNUM,Gender, Marital_Status,
Customer_Age from financeiro where Customer_Age =0 order by
CLIENTNUM desc",
  "fetch_size": 5
}

POST financeiro/_delete_by_query
{
  "query": {
    "match": {
      "Customer_Age": 0
    }
  }
}
```

- Agregação que recupera o total de documentos por **Gender**, o size=3 informa para retornar os 3 maiores.

```
GET financeiro/_search
{
 "size": 0,
 "aggs": {
   "Total Documentos por Categoria de Renda": {
     "terms": {
       "field": "Income_Category.keyword",
       "size": 3
     }
   }
 }
}
```

- Geração de todas as estatísticas de um campo numérico (**Total_Trans_Amt**).

```
GET financeiro/_search
{
    "aggregations": {
          "Montante": {
              "extended_stats": {
                          "field": "Total_Trans_Amt"
                                }
                      }
                  },
     "size": 0
}
```

- Agregação com os resultados de todos os registros cujo **Gender** é F, com a métrica **Total_Trans_Amt**, gerando a média do montante.

```
GET financeiro/_search
{
"size" : 0,
 "aggs" : {
        "Sexo Feminino" : {
            "filter" : { "term": { "Gender.keyword": "F"} },
            "aggs" : {
                "Media" : { "avg" : { "field" :
"Total_Trans_Amt" }    }
            }
        }
    }
}
```