

Conversor de consulta SQL a DSL para Elasticsearch

Introducción

El motor de consultas de Elasticsearch, es [DSL](#) y tiene sus peculiaridades. No recuerdo en que version fue, pero se incorporó a Elasticsearch la herramienta [SQL Translate API](#) en mi opinion una de las mejoras cosas que ocurrio para evitar el uso de librerias de terceros, por el miedo escénico que produce el lenguaje DSL.

Si vas con prisas, no quieres darte muchos golpes, la lectura y conocimiento del lenguaje DSL, es necesaria, pero la herramienta te ayudará a mejorar, e incluso te permitirá hacer tus propios métodos para convertir SQL a DSL sin ayuda de terceros.

Ejemplo

El ejemplo es ejecutado en kibana, herramienta indispensable para jugar con elasticsearch.

Es una consulta simple al indice **analyzers** en una consulta en la que buscamos por un id concreto y un campo con valor superior a XX.

```
POST /_sql/translate
{
  "query": "SELECT * FROM analyzers WHERE modem_id = 1 AND vac1_v > 260"
}
```

Resultado

```
{
  "size": 1000,
  "query": {
    "bool": {
      "must": [
        {
```

```
      "term": {
        "modem_id": {
          "value": 1
        }
      },
    },
    {
      "range": {
        "vac1_v": {
          "gt": 260,
          "boost": 1
        }
      }
    }
  ],
  "boost": 1
},
"_source": false,
"fields": [
  {
    "field": "cos1_a"
  },
  {
    "field": "cos1_m"
  },
  {
    "field": "cos2_a"
  },
  {
    "field": "cos2_m"
  },
  {
    "field": "cos3_a"
  },
  {
    "field": "cos3_m"
  },
  {
    "field": "datetime",
```

```
    "format": "strict_date_optional_time_nanos"
  },
  {
    "field": "eneact1_kwh"
  },
  {
    "field": "eneact2_kwh"
  },
  {
    "field": "eneact3_kwh"
  },
  {
    "field": "eneact_kwh"
  },
  {
    "field": "eneapa1_kvah"
  },
  {
    "field": "eneapa2_kvah"
  },
  {
    "field": "eneapa3_kvah"
  },
  {
    "field": "enereal_kvarh"
  },
  {
    "field": "enerea2_kvarh"
  },
  {
    "field": "enerea3_kvarh"
  },
  {
    "field": "enerea_kvarh"
  },
  {
    "field": "freq_hz"
  },
  {
    "field": "iac1_a"
```

```
},
{
  "field": "iac1_a_a"
},
{
  "field": "iac1_m_a"
},
{
  "field": "iac2_a"
},
{
  "field": "iac2_a_a"
},
{
  "field": "iac2_m_a"
},
{
  "field": "iac3_a"
},
{
  "field": "iac3_a_a"
},
{
  "field": "iac3_m_a"
},
{
  "field": "ip"
},
{
  "field": "log_id"
},
{
  "field": "lvac1_v"
},
{
  "field": "lvac2_v"
},
{
  "field": "lvac3_v"
},
}
```

```
{
  "field": "message"
},
{
  "field": "mod_bus_error"
},
{
  "field": "modem_id"
},
{
  "field": "pa1_a_w"
},
{
  "field": "pa1_m_w"
},
{
  "field": "pa1_w"
},
{
  "field": "pa2_a_w"
},
{
  "field": "pa2_m_w"
},
{
  "field": "pa2_w"
},
{
  "field": "pa3_a_w"
},
{
  "field": "pa3_m_w"
},
{
  "field": "pa3_w"
},
{
  "field": "pf"
},
{
```

```
    "field": "pf1"
  },
  {
    "field": "pf2"
  },
  {
    "field": "pf3"
  },
  {
    "field": "powapa_va"
  },
  {
    "field": "powrea_var"
  },
  {
    "field": "pp1_va"
  },
  {
    "field": "pp2_va"
  },
  {
    "field": "pp3_va"
  },
  {
    "field": "pr1_a_var"
  },
  {
    "field": "pr1_m_var"
  },
  {
    "field": "pr1_var"
  },
  {
    "field": "pr2_a_var"
  },
  {
    "field": "pr2_m_var"
  },
  {
    "field": "pr2_var"
```

```
},
{
  "field": "pr3_a_var"
},
{
  "field": "pr3_m_var"
},
{
  "field": "pr3_var"
},
{
  "field": "provider"
},
{
  "field": "response_time"
},
{
  "field": "status_code"
},
{
  "field": "v_event"
},
{
  "field": "vac1_a_v"
},
{
  "field": "vac1_m_v"
},
{
  "field": "vac1_v"
},
{
  "field": "vac2_a_v"
},
{
  "field": "vac2_m_v"
},
{
  "field": "vac2_v"
},
},
```

```
{
  "field": "vac3_a_v"
},
{
  "field": "vac3_m_v"
},
{
  "field": "vac3_v"
}
],
"sort": [
  {
    "_doc": {
      "order": "asc"
    }
  }
],
"track_total_hits": -1
}
```

Bien, la consulta para nuestro propósito sería

```
GET /analyzers/_search
{
  "query": {
    "bool": {
      "must": [
        {
          "term": {
            "modem_id": {
              "value": 1
            }
          }
        },
        {
          "range": {
            "vac1_v": {
              "gt": 260,
              "boost": 1
            }
          }
        }
      ]
    }
  }
}
```



```
    }  
  }  
],  
  "boost": 1  
}  
}  
}
```

“ Es importante si no conoces Elasticsearch que comprendas que hay partes que no deberían existir en los índices de Elasticsearch, que vienen de la mentalidad de datos estructurados, típicos de los motores SQL. Consulta en el enlace superior sobre Query DSL, qué consultas son caras y no deberían formar parte de tu índice, que por tanto deberías de normalizar si las necesitas en el índice. Buena suerte

Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).

Revision #1

Created 16 November 2022 08:19:55 by Abkrim

Updated 16 November 2022 08:32:39 by Abkrim