

12 - Práce s relačními daty

Předpokládejme, že máme v databázi dvě provázané tabulky — *order* a *order_product*:

- **order:**

id	email	date	status
1	john.s@icloud.com	2022-02-20	delivered
2	jorge.c@gmail.com	2022-02-21	new

- **order_product:**

id	order_id	name	price
24	1	OttLite Achieve Desk Lamp	49.97
25	1	TOPS Pen Pal Flexible Pen Holders	3.36
26	2	Pen+Gear Plastic Storage Box	1.24

V Elasticsearch bohužel není možné provést dotaz s JOIN, který by spojil výsledky z dvou indexů.

Řešením je provést spojení ještě předtím, než se data do Elasticsearch uloží a ukládat je tak v denormalizované podobě.

Pro uložení takového dokumentu do Elasticsearch by bylo nutné vytvořit následující JSON:

```
{
  "email": "john.s@icloud.com",
  "date": "2022-02-20",
  "status": "delivered",
  "products": [
    {
      "name": "OttLite Achieve Desk Lamp",
      "price": 49.97
    },
    {
      "name": "TOPS Pen Pal Flexible Pen Holders",
      "price": 3.36
    }
  ]
}
```

```
}
```

Přestože lze tento dokument uložit do Elasticsearch tak jak je, v některých případech se bude chovat neočekávaně.

Například, pokud zkusíme vyhledat objednávky, které obsahují produkt, jehož `name` začíná `OttLite` a zároveň je jeho `price` menší než `10`.

In this case, such a document would be found, even though the first condition is met only in the first product, while the second condition applies only to the second (different) product.

To fix it, a mapping for a field `products` must be set to a type `nested`:

```
PUT order_nested
{
  "mappings": {
    "properties": {
      "status": {
        "type": "keyword"
      },
      "products": {          // <= array of nested objects
        "type": "nested",
        "properties": {
          "price": {
            "type": "half_float"
          }
        }
      }
    }
  }
}
```

Then, when searching in a nested object, you have to wrap your query into a `nested` query:

```
GET order_nested/_search
{
  "query": {
    "nested": {
      "path": "products",
      "query": {
        "range": {
          "products.nested": {
            "lt": 10
          }
        }
      }
    }
  }
}
```

The same applies to aggregations — you should use a `nested` aggregation as a wrapper for other queries.