POLITECNICO
MILANO 1863

Elasticsearch

Andrea Tocchetti
andrea.tocchetti@polimi.it

Elasticsearch - Dataset

You can download the dataset at the following LINK.

The dataset is a collection of employees with the following attributes and types.

- **Address** - Text
- **Age** - Integer
- **DateOfJoining** - Date
- **Designation** - Text
- **FirstName** - Text
- **Gender** - Text
- **Interests** - Text
- **LastName** - Text
- **MaritalStatus** - Text
- **Salary** - Integer

Elasticsearch - Exercises

Write a query to collect all the employees whose first name is “Bob”.

```
GET /employees50k/_search {
        “query”: {
                “match”: {
                        “FirstName”: “Bob”
                }
        }
}
```

Elasticsearch - Exercises

Write a query to collect all the employees whose salary is greater than 80’000 and less than 95’000.

```
GET /employees50k/_search {
    “query”: {
        “range”: {
            “Salary”: {
                “gt”: 80’000,
                “lt”: 95’000
            }
        }
    }
}
```

Elasticsearch - Exercises

Write a query to collect all the employees whose age is 31.

```
GET /employees50k/_search {
        “query”: {
                “term”: {
                        “Age”: 31
                }
        }
}
```

Disclaimer: This solution is not optimal (although it works). The best one would be a range query with gte and lte 31.

Elasticsearch - Exercises

Write a query to collect all the employees who joined later than 03/09/2015.

```
GET /employees50k/_search {
        “query”: {
                “range”: {
                        “DateOfJoining”: {
                                “gt”: “03-09-2015” → Any date format would be fine (based on the mapping)
                        }
                }
        }
}
```

Elasticsearch - Exercises

Write a query to collect all the employees whose gender is exactly “Male”.

```
GET /employees50k/_search {
        “query”: {
                “term”: {
                        “Gender”: “Male”
                }
        }
}
```

Elasticsearch - Exercises

Write a query to collect all the employees whose first name is “Moses” and whose last name is “Daschofsky”.

```
GET /employees50k/_search {
        “query”: {
                “bool”: {
                        “must”: [ {“match”: {“FirstName”: “Moses”}},
                                  {“match”: {“LastName”: “Daschofsky”}} ]
                }
        }
}
```

Elasticsearch - Exercises

Write a query to collect all the employees whose interests contains **one or more words** among “Paragliding”, “Kayaking”, and “Playing”. The more words are found, the higher the final score.

```
GET /employees50k/_search {
        “query”: {
                “match”: {
                        “Interest”: “Paragliding, Kayaking, Playing”,
                        “operator”: “OR”
                }
        }
}
```

Elasticsearch - Exercises

Write a query to collect all the employees whose name is “Elden”, assigning a higher score those whose designation is “Delivery Manager”.

```
GET /employees50k/_search {
        “query”: {
                “bool”: {
                        “must”: [ {“match”: {“FirstName”: “Elden”}} ],
                        “should”: [ {“match”: {“Designation”: “Delivery Manager”}} ]
                }
        }
}
```

Elasticsearch - Exercises

Write a query to collect all the employees whose last name is “Weatherly” and whose salary is greater than 50’000. The name should not affect the final score.

```
GET /employees50k/_search {
        “query”: {
                “bool”: {
                        “must”: [ {“range”: {“Salary”: {“gt”: 50’000} }} ]
                        “filter”: [ {“match”: {“LastName”: “Weatherly”}} ],
                }
        }
}
```

Elasticsearch - Exercises

Write a query to collect all the employees whose designation is “Manager”, assigning a higher score to those whose gender is “Female” and those who are “Married”.

```
GET /employees50k/_search {
        “query”: {
                “bool”: {
                        “must”: [ {“match”: {“Designation”: “Manager” }} ]
                        “should”: [ {“term”: {“Gender”: “Male”}}, {“term”: {“MaritalStatus”: “Married”}} ],
                }
        }
}
```

Elasticsearch - Exercises

Write a query to collect all the employees whose designation is “Manager” or “Delivery Manager” and whose salary is not higher than 150’000. Assign a higher score to those whose interests includes “Blogging”.

```
GET /employees50k/_search {
        “query”: {
                “bool”: {
                        “must”: [ {“match”: {“Designation”: “Delivery Manager”, “operator”: “OR” }},
                                    {“range”: {“Salary”: {“lte”: 150’000} }} ]
                        “should”: [ {“match”: {“Interest”: “Blogging”}} ]
                }
        }
}
```

Elasticsearch - Exercises

Write a query to count the number of employees based on their gender and designation **separately**.

```
GET /employees50k/_search {
        “query”: {
                “size”: 0,
                “aggs”: {
                        “employees_per_gender”: {“terms”: {“field”: “Gender”} },
                        “employees_per_designation”: {“terms”: {“field”: “Designation”} }
                }
        }
}
```

Elasticsearch - Exercises

Write a query to count the number of employees based on their gender. Then, compute the number of people based on their age for each gender.

```
GET /employees50k/_search {
    “query”: {
        “size”: 0,
        “aggs”: {
            “employees_per_gender”: {
                “terms”: {“field”: “Gender”}
                “aggs”: {
                    “employees_per_gender_per_age”: {
                        “terms”: {“field”: “Age”}
                }
            }
        }
    }
}
```