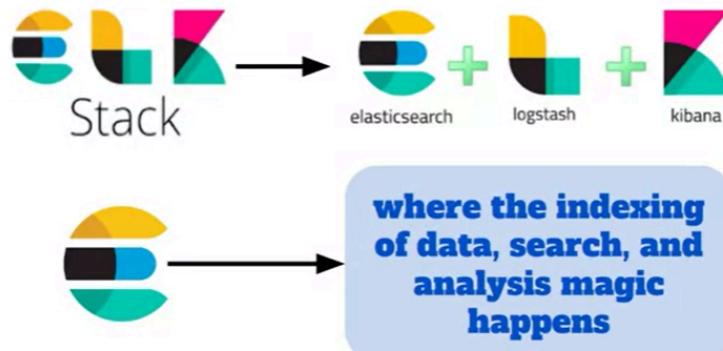
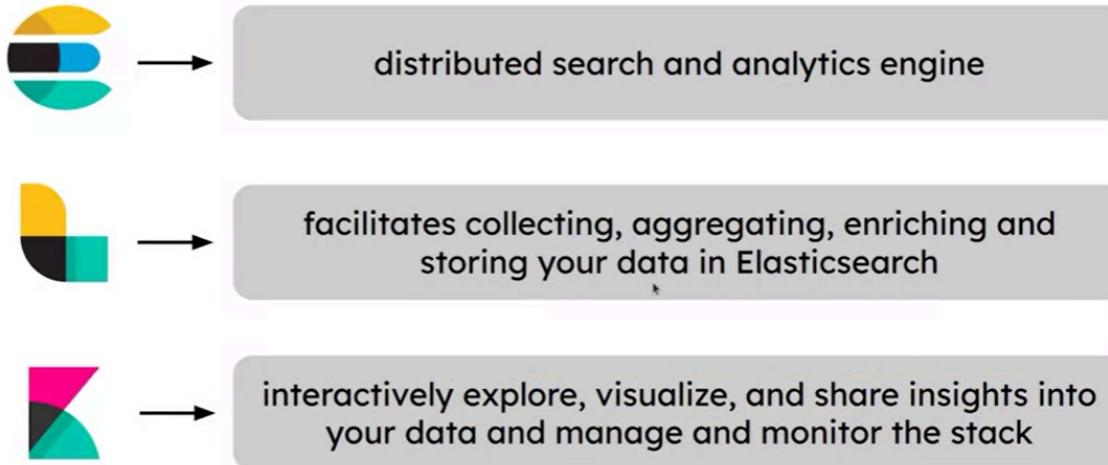


What is Elasticsearch?

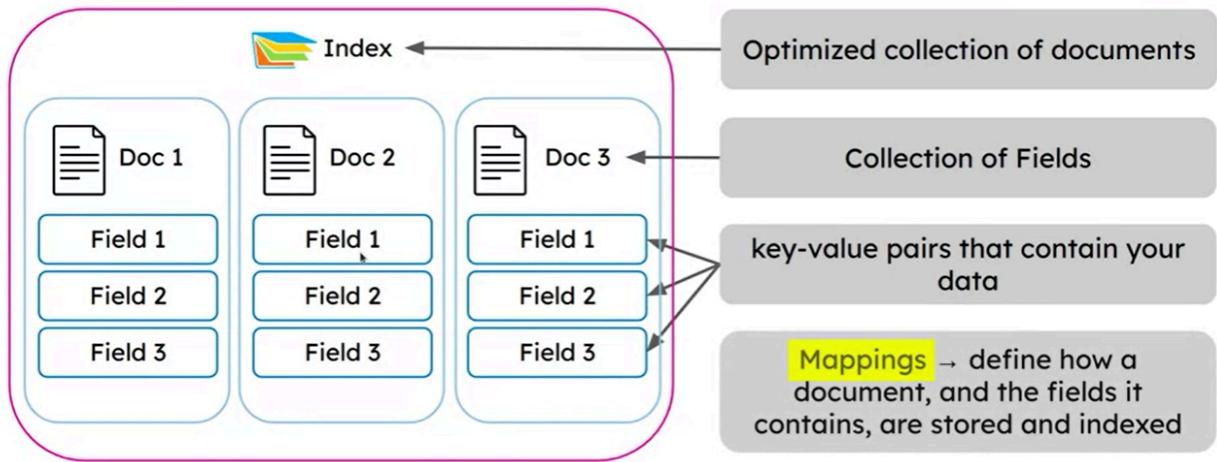
Elasticsearch is the distributed search and analytics engine at the heart of the Elastic Stack



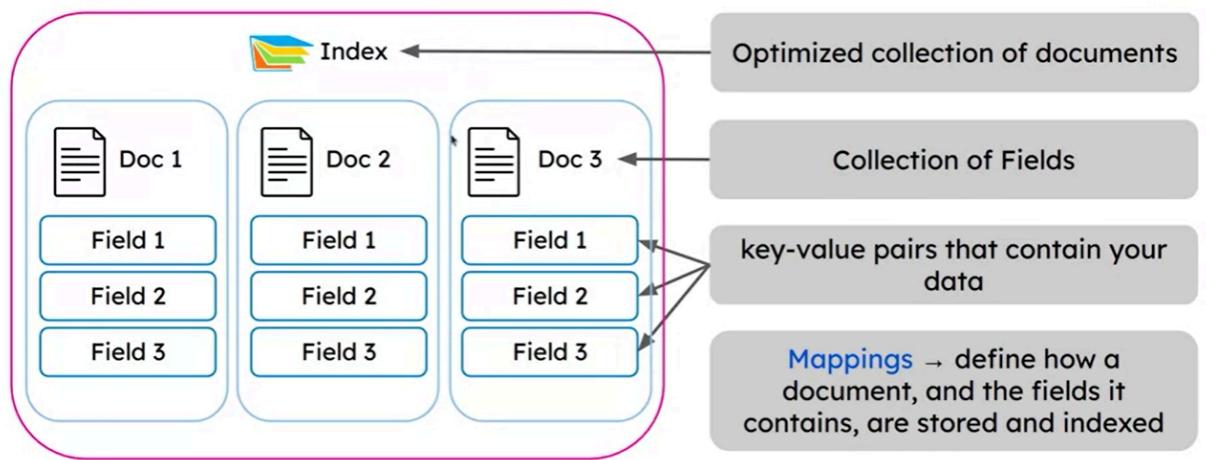
ELK Stack



How Elasticsearch stores data



How Elasticsearch stores data



Not a relational database



Schema flexibility

→ add or remove fields to documents at any time without having to modify the underlying schema

Data Model

→ document-oriented → stores data in a hierarchical JSON-like format



Not a relational database

Scalability

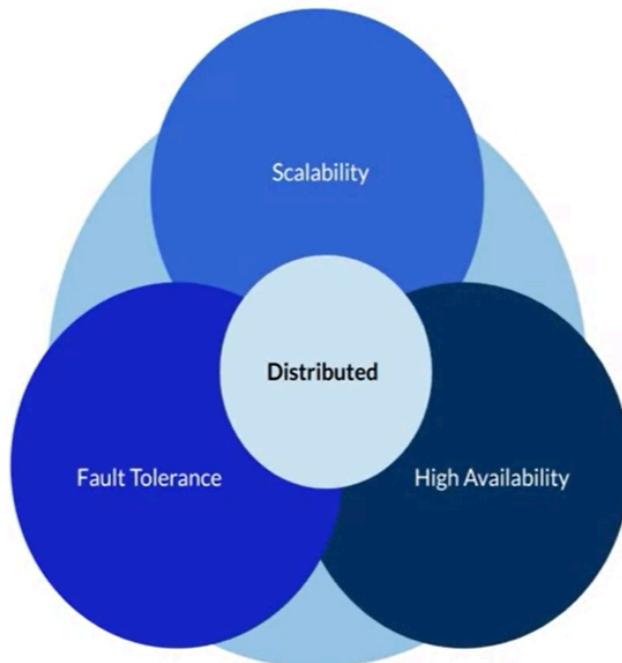
→ distribute data across multiple nodes and handle large volumes of data and queries in real-time

Different Search functionality/Use

→ optimized for search and has powerful search capabilities, including support for full-text search, faceted search, and complex queries

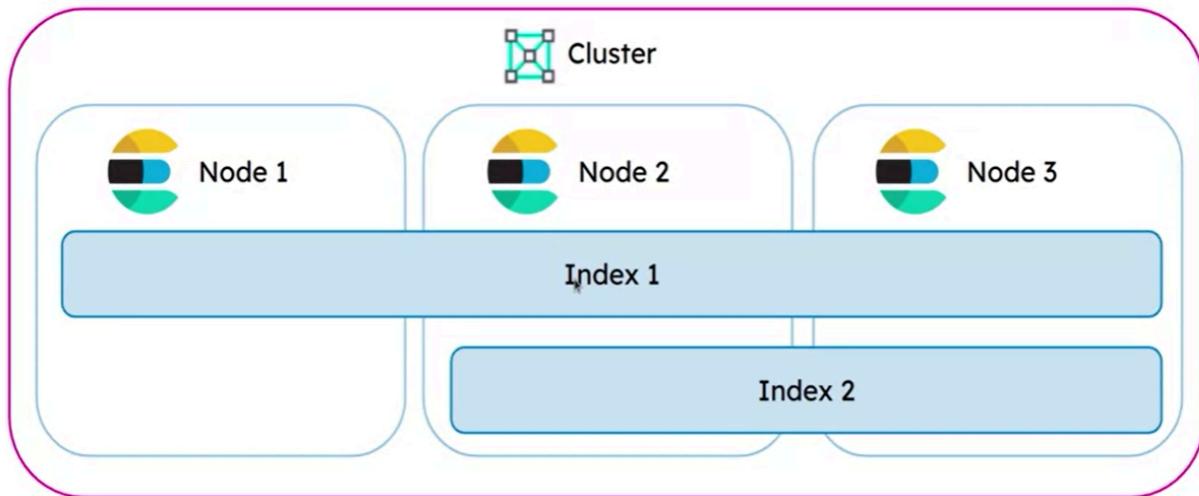


Architecture of Elasticsearch

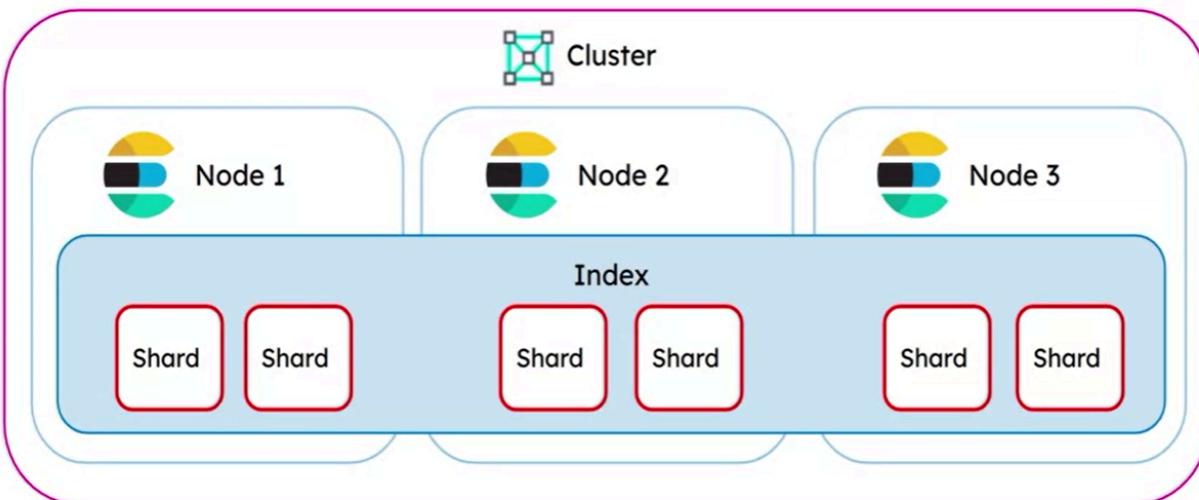


clusters
Nodes
Indices
Shards

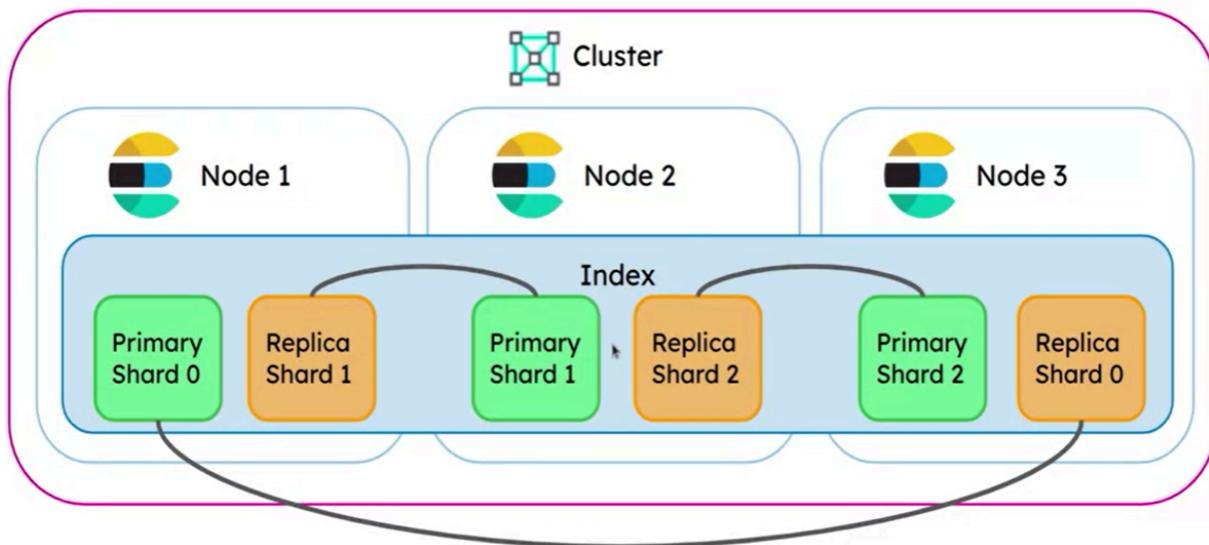
Architecture of Elasticsearch



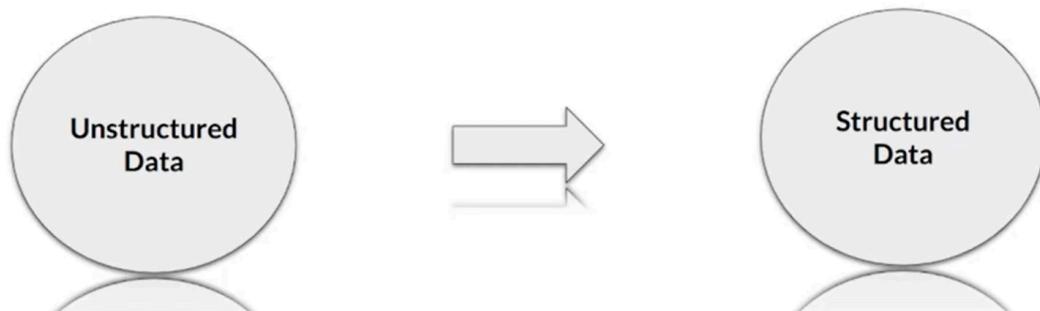
Architecture of Elasticsearch



Architecture of Elasticsearch



Text Analysis



Common use cases

- Build a search engine
- Mine unstructured data
- Fine tune search for a specific language
- Perform lexicographic or linguistic research

Basic Concepts of Text Analysis

- **Tokenization**

– breaking a text down into smaller chunks, called tokens.

the quick brown fox → ["the", "quick", "brown", "fox"]

- **Normalization**

– allows you to match tokens that are not exactly the same as the search terms, but similar enough to still be relevant

Quick → quick (lowercased)

Foxes → fox (stemmed)

jump, leap → jump (synonyms)

How do Text Analyzers work?

- Character filters

- transforms a stream of characters by adding, removing, or changing characters.

Hindu-Arabic numerals (۱۲۳۴۵۶۷۸۹) – Arabic-Latin equivalents (۱۲۳۴۵۶۷۸۹)

strip HTML elements like from the stream.

- An analyzer may have zero or more character filters, which are applied in order.

- Tokenizer

- receives a stream of characters, breaks it up into individual tokens (usually individual words), and outputs a stream of token

a "whitespace" tokenizer would convert "Quick brown fox!" into the terms [Quick, brown, fox!].

- An analyzer must have exactly one tokenizer

- Token Filters

- receives the token stream and may add, remove, or change tokens

"lowercase" token filter → converts all tokens to lowercase

"stop" token filter - removes common words (stop words) like the from the token stream

The screenshot shows the Elasticsearch Dev Tools interface with the 'Console' tab selected. The URL bar indicates the current endpoint is `/_index/hotel_data`. The main content area displays a large JSON object representing the configuration for the 'hotel_data' index. The JSON structure includes fields for 'mappings', 'settings', and various properties like 'category', 'city', and 'property_id'. The code editor on the left shows the following code:

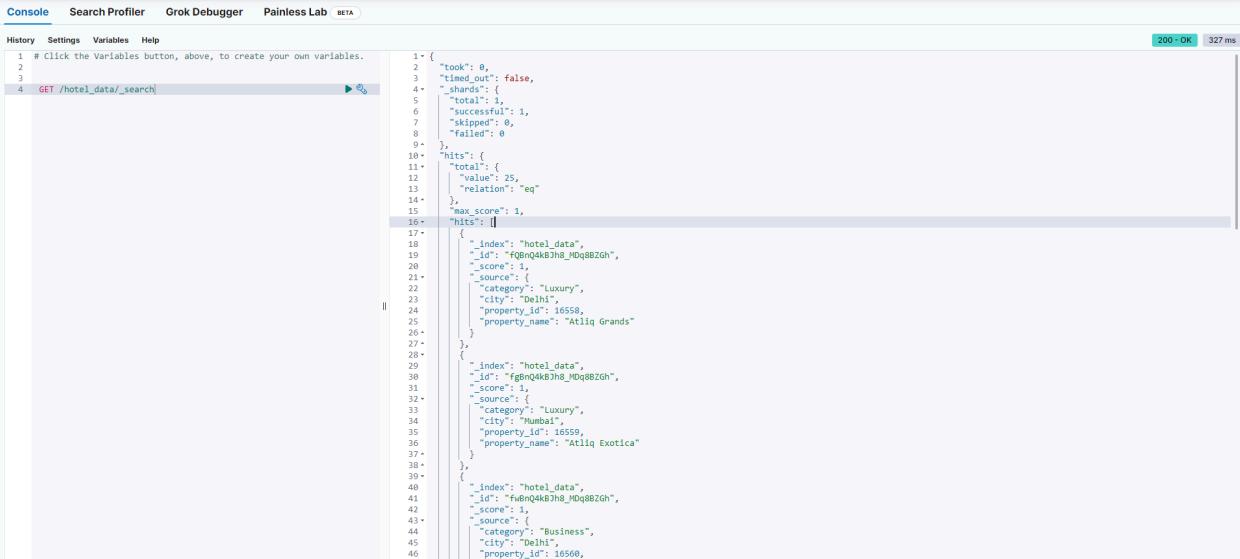
```
1 # Click the Variables button, above, to create your own variables.  
2 GET /hotel_data
```

At the top right, there are status indicators: '200 - OK' and '356 ms'.

```
1 {  
2   "mappings": {  
3     "properties": {  
4       "category": {  
5         "type": "keyword"  
6       },  
7       "city": {  
8         "type": "keyword"  
9       },  
10      "property_id": {  
11        "type": "long"  
12      },  
13      "property_name": {  
14        "type": "keyword"  
15      }  
16    }  
17  },  
18  "settings": {  
19    "index": {  
20      "routing": {  
21        "allocation": {  
22          "include": {  
23            "tier_preference": "data_content"  
24          }  
25        }  
26      }  
27    },  
28    "number_of_shards": "1",  
29    "provided_name": "hotel_data",  
30    "creation_date": "1689052971541",  
31    "number_of_replicas": "1",  
32    "uuid": "3Kq4g5t1rnB8PChupYwW",  
33    "version": {  
34      "created": "8080299"  
35    }  
36  }  
37}  
38}  
39}  
40}  
41}  
42}  
43}
```

Searching in the overall index

GET /hotel_data/_search



The screenshot shows the Elasticsearch Dev Tools interface with the 'Console' tab selected. The URL bar shows 'GET /hotel_data/_search'. The response body is displayed as a JSON object with the following structure:

```
1 * {  
2     "took": 0,  
3     "timed_out": false,  
4     "_shards": {  
5         "total": 1,  
6         "successful": 1,  
7         "skipped": 0,  
8         "failed": 0  
9     },  
10    "hits": {  
11        "total": {  
12            "value": 25,  
13            "relation": "eq"  
14        },  
15        "max_score": 1,  
16        "hits": [  
17            {  
18                "_index": "hotel_data",  
19                "_id": "fgBnQ4k8Jh8_MDq88ZGh",  
20                "_score": 1,  
21                "_source": {  
22                    "category": "Luxury",  
23                    "city": "Delhi",  
24                    "property_id": 16558,  
25                    "property_name": "Atiliq Grands"  
26                }  
27            },  
28            {  
29                "_index": "hotel_data",  
30                "_id": "fgBnQ4k8Jh8_MDq88ZGh",  
31                "_score": 1,  
32                "_source": {  
33                    "category": "Luxury",  
34                    "city": "Mumbai",  
35                    "property_id": 16559,  
36                    "property_name": "Atiliq Exotica"  
37                }  
38            },  
39            {  
40                "_index": "hotel_data",  
41                "_id": "fgBnQ4k8Jh8_MDq88ZGh",  
42                "_score": 1,  
43                "_source": {  
44                    "category": "Business",  
45                    "city": "Delhi",  
46                    "property_id": 16560,  
47                }  
48            }  
49        ]  
50    }  
51}
```

The status bar at the top right indicates '200 - OK' and '327 ms'.

Searching from the first 100 documents in the index

GET /hotel_data/_search

```
{  
    "size": "100",  
    "from": 0  
}
```

Console Search Profiler Grok Debugger Painless Lab BETA

History Settings Variables Help

```

1 # Click the Variables button, above, to create your own variables.
2
3
4 GET /hotel_data/_search
5 {
6   "size": "100",
7   "from": 0
8 }

```

```

1+ {
2   "took": 0,
3   "timed_out": false,
4+ "_shards": {
5     "total": 1,
6     "successful": 1,
7     "skipped": 0,
8     "failed": 0
9   },
10+ "hits": {
11   "total": {
12     "value": 25,
13     "relation": "eq"
14   },
15   "max_score": 1,
16   "hits": [
17     {
18       "_index": "hotel_data",
19       "_id": "fgbnq4k8Jhd_MQgBZGh",
20       "_score": 1,
21       "_source": {
22         "category": "Luxury",
23         "city": "Delhi",
24         "property_id": 16558,
25         "property_name": "Atiliq Grands"
26       }
27     },
28     {
29       "_index": "hotel_data",
30       "_id": "fgbnq4k8Jhd_MQgBZGh",
31       "_score": 1,
32       "_source": {
33         "category": "Luxury",
34         "city": "Mumbai",
35         "property_id": 16559,
36         "property_name": "Atiliq Exotica"
37       }
38     },
39     {
40       "_index": "hotel_data",
41       "_id": "fu6nq4k8Jhd_MQgBZGh",
42       "_score": 1,
43       "_source": {
44         "category": "Business",
45         "city": "Delhi",
46         "property_id": 16560,
47       }
48     }
49   ]
50 }

```

200 - OK 338 ms

Searching in the next 100 documents of the index

GET /hotel_data/_search

```
{
  "size": "100",
  "from": 100
}
```

Console Search Profiler Grok Debugger Painless Lab BETA

History Settings Variables Help

```
1 # Click the Variables button, above, to create your own variables.
2
3 GET /hotel_data/_search
4 {
5   "size":100,
6   "from":100
7 }
8 ]
```

1+ [
2 "took": 0,
3 "timed_out": false,
4 "_shards": {
5 "total": 1,
6 "successful": 1,
7 "skipped": 0,
8 "failed": 0
9 },
10 "hits": {
11 "total": {
12 "value": 25,
13 "relation": "eq"
14 },
15 "max_score": 1,
16 "hits": []
17 }
18]

200 - OK 433 ms

To create new index into the cluster

PUT /sample_index2

Console Search Profiler Grok Debugger Painless Lab BETA

History Settings Variables Help

```
1 # Click the Variables button, above, to create your own variables.
2
3
4 PUT /sample_index2
5 ]
```

1+ [
2 "acknowledged": true,
3 "shards_acknowledged": true,
4 "index": "sample_index2"
5]

200 - OK 969 ms

Create new document into the index

Console Search Profiler Grok Debugger Painless Lab BETA

History Settings Variables Help

```
1 # Click the Variables button, above, to create your own variables.
2
3
4 PUT /sample_index2
5
6 POST /sample_index2/_doc
7 {
8   "name": "Surya Mahesh Kolisetty",
9   "Age": 20,
10  "Height": 5.8,
11  "Weight": 56
12 }
```

1+ {
2 "index": "sample_index2",
3 "_id": "sqwadG5uvtPefBbt",
4 "_version": 1,
5 "_result": "created",
6 "_shards": {
7 "total": 2,
8 "successful": 2,
9 "failed": 0
10 },
11 "_seq_no": 0,
12 "_primary_term": 1
13 }

201 - Created 389 ms

☰ D Dev Tools Console

Console Search Profiler Grok Debugger Painless Lab BETA

History Settings Variables Help

```
1 # Click the Variables button, above, to create your own variables.
2
3
4 PUT /sample_index2/_doc/_id_1
5 {
6   "name": "surya",
7   "age": 20
8 }
```

1+ {
2 "index": "sample_index2",
3 "_id": "sqwadG5uvtPefBbt",
4 "_version": 1,
5 "_result": "created",
6 "_shards": {
7 "total": 2,
8 "successful": 2,
9 "failed": 0
10 },
11 "_seq_no": 1,
12 "_primary_term": 1
13 }

Searching for the document inside the index after putting into it

☰ D Dev Tools Console

Console Search Profiler Grok Debugger Painless Lab BETA

History Settings Variables Help

```

1 # Click the Variables button, above, to create your own variables.
2
3 PUT /sample_index2/_doc/id_1
4 {
5   "name": "surya",
6   "age": 20
7 }
8
9 GET /sample_index2/_search

```

```

1 {
2   "took": 103,
3   "timed_out": false,
4   "_shards": {
5     "total": 1,
6     "successful": 1,
7     "skipped": 0,
8     "failed": 0
9   },
10  "hits": {
11    "total": {
12      "value": 2,
13      "relation": "eq"
14    },
15    "max_score": 1,
16    "hits": [
17      {
18        "_index": "sample_index2",
19        "_id": "AQNZQ4kB5wKvtPeFbIbt",
20        "_score": 1,
21        "_source": {
22          "name": "Surya Mahesh Kolisetty",
23          "age": 20,
24          "height": 5.8,
25          "weight": 56
26        }
27      },
28      {
29        "_index": "sample_index2",
30        "_id": "id_1",
31        "_score": 1,
32        "_source": {
33          "name": "surya",
34          "age": 20
35        }
36    ]
37  }
38}
39

```

Delete documents with id inside the document

☰ D Dev Tools Console Search Profiler Grok Debugger Painless Lab BETA

History Settings Variables Help

```

1 # Click the Variables button, above, to create your own variables.
2
3 PUT /sample_index2/_doc/id_1
4 {
5   "name": "surya",
6   "age": 20
7 }
8
9 GET /sample_index2/_search
10
11 DELETE /sample_index2/_doc/AQNZQ4kB5wKvtPeFbIbt

```

```

1 {
2   "took": 0,
3   "timed_out": false,
4   "_shards": {
5     "total": 1,
6     "successful": 1,
7     "skipped": 0,
8     "failed": 0
9   },
10  "hits": {
11    "total": {
12      "value": 1,
13      "relation": "eq"
14    },
15    "max_score": 1,
16    "hits": [
17      {
18        "_index": "sample_index2",
19        "_id": "id_1",
20        "_score": 1,
21        "_source": {
22          "name": "surya",
23          "age": 20
24        }
25      ]
26    }
27  }
28

```

Delete the index in th cluster

The screenshot shows the Elasticsearch Painless Lab interface. At the top, there are tabs for Console, Search Profiler, Grok Debugger, and Painless Lab (BETA). Below the tabs, a menu bar includes History, Settings, Variables, and Help. The main area contains a code editor with the following Painless script:

```

1 # Click the Variables button, above, to create your own variables.
2
3 PUT /sample_index2/_doc/_id_1
4 {
5   "name": "surya",
6   "age": 20
7 }
8
9 GET /sample_index2/_search
10
11 DELETE /sample_index2/_doc/AQNzQ4k85wKvtPeFbIbt
12
13 DELETE /sample_index2

```

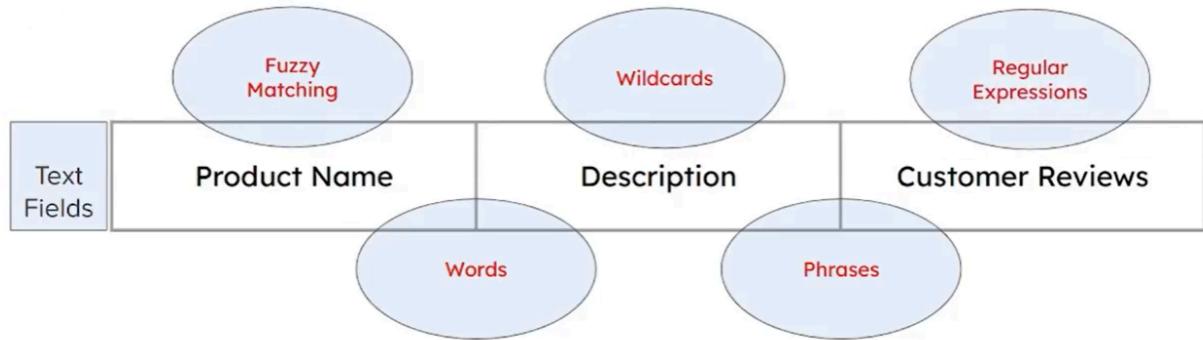
To the right of the code editor, a results panel displays the response from the last DELETE request:

```

1 { "acknowledged": true
2
3 }

```

Full Text Queries



Full Text Queries: A brief Example

Intervals Query

- returns documents based on the order and proximity of matching terms.
- uses matching rules, constructed from a small set of definitions



elastic

Find apps, content, and more.

☰ D Dev Tools Console

Console Search Profiler Grok Debugger Painless Lab BETA

History Settings Variables Help

```
1 #sony camera or sony player
2
3 GET /product_catalog/_search
4 {
5   "query": {
6     "intervals": {
7       "name": {
8         "all_of": {
9           "intervals": [
10             {
11               "match": {
12                 "query": "Sony"
13               }
14             },
15             {
16               "any_of": {
17                 "intervals": [
18                   {
19                     "match": {
20                       "query": "Camera"
21                     }
22                   },
23                   {
24                     "match": {
25                       "query": "Player"
26                     }
27                   }
28                 ]
29               }
30             }
31           ]
32         }
33       }
34     }
35   }
36 }
```

Console Search Profiler Grok Debugger Painless Lab BETA 200 - OK 314 ms

```
History Settings Variables Help
1 $sony camera or sony player
2
3 GET /product_catalog/_search
4 {
5   "query": {
6     "bool": {
7       "name": {
8         "all_of": [
9           {
10             "match": {
11               "query": "Sony"
12             }
13           }
14         ],
15         "any_of": [
16           {
17             "intervals": [
18               {
19                 "match": {
20                   "query": "Camera"
21                 }
22               },
23               {
24                 "match": {
25                   "query": "Player"
26                 }
27               }
28             ]
29           }
30         ]
31       }
32     }
33   }
34 }
35 }
```

10+ "hits": {
11+ "total": {
12+ "value": 32,
13+ "relation": "eq"
14+ },
15+ "max_score": 0.33333333,
16+ "hits": [
17+ {
18+ "index": "product_catalog",
19+ "id": "ZB0wvocuH7HSig2XzP4",
20+ "score": 0.33333333,
21+ "source": {
22+ "name": "Sony Soft Camera Carrying Case - LCSMX100",
23+ "description": "Sony Soft Camera Carrying Case - LCSMX100/ Stain Resistant Nylon/ Colored Interior/ Movable Partition/ Build-In Modular Interface/ Black Finish",
24+ "id": "34198"
25+ }
26+ },
27+ {
28+ "index": "product_catalog",
29+ "id": "B0BwvocuH7HSig2XzL3",
30+ "score": 0.19999999,
31+ "source": {
32+ "name": "Sony 5 Disc CD Player - CDCE375",
33+ "description": "Sony 5 Disc CD Player- CDCE375/ 5 Disc Changer/ Variable Line Output/ CD-R/RW Playback Capability/ 20 Track Basic Calendar/ Digital Servo Control/ Remote Commander Remote Control",
34+ "id": "5644"
35+ }
36+ },
37+ {
38+ "index": "product_catalog",
39+ "id": "SHDwvocuH7HSig2Xz14",
40+ "score": 0.19999999,
41+ "source": {
42+ "name": "Sony DSCQ Soft Carrying Camera Case - Black Finish - LSCSH",
43+ "description": "Sony DSCQ Soft Carrying Black Camera Case - LSCSH/ Polypropylene Construction/ Extra Compartments For Accessory Storage/ Can Hold Many Sony Or Other Manufacturers Digital Cameras/ Black Finish",
44+ "id": "24922"
45+ }
46+ },
47+ {
48+ "index": "product_catalog",
49+ "id": "SHDwvocuH7HSig2XzD4",
50+ "score": 0.19999999,
51+ "source": {
52+ "name": "Sony DSLR-A350 Digital Camera And 18-70mm Zoom Lens - DSCRA350K/ 14.2 Megapixel/ Tiltable LCD Screen/ Smart Teleconverter 2X Zoom/ Blonz Image Processor/ Super SteadyShot In-Camera Image Stabilization/ Anti-Dust Technology/ Eye-Start Autofocus System/ Auto Pop-Up Flash/ Black Finish",
53+ "id": "34024"
54+ }
55+ }
56+ }
57+ }
58+ }

Console Search Profiler Grok Debugger Painless Lab BETA 200 - OK 313 ms

```
History Settings Variables Help
1 $sony camera or sony player
2
3 GET /product_catalog/_search
4 {
5   "query": {
6     "bool": {
7       "name": {
8         "all_of": [
9           {
10             "intervals": [
11               {
12                 "match": {
13                   "query": "Sony"
14                 }
15               }
16             ],
17             "any_of": [
18               {
19                 "intervals": [
20                   {
21                     "match": {
22                       "query": "Camera"
23                     }
24                   },
25                   {
26                     "match": {
27                       "query": "Player"
28                     }
29                   }
30                 ]
31               }
32             ]
33           }
34         ]
35       }
36     }
37   }
38 }
```

1+ {
2+ "took": 3,
3+ "timed_out": false,
4+ "shards": {
5+ "total": 1,
6+ "successful": 1,
7+ "skipped": 0,
8+ "failed": 0
9+ },
10+ "hits": {
11+ "total": {
12+ "value": 1,
13+ "relation": "eq"
14+ },
15+ "max_score": 0.33333333,
16+ "hits": [
17+ {
18+ "index": "product_catalog",
19+ "id": "ZB0wvocuH7HSig2XzP4",
20+ "score": 0.33333333,
21+ "source": {
22+ "name": "Sony Soft Camera Carrying Case - LCSMX100",
23+ "description": "Sony Soft Camera Carrying Case - LCSMX100/ Stain Resistant Nylon/ Colored Interior/ Movable Partition/ Build-In Modular Interface/ Black Finish",
24+ "id": "34198"
25+ }
26+ },
27+ {
28+ "index": "product_catalog",
29+ "id": "B0BwvocuH7HSig2XzL3",
30+ "score": 0.19999999,
31+ "source": {
32+ "name": "Sony 5 Disc CD Player - CDCE375",
33+ "description": "Sony 5 Disc CD Player- CDCE375/ 5 Disc Changer/ Variable Line Output/ CD-R/RW Playback Capability/ 20 Track Basic Calendar/ Digital Servo Control/ Remote Commander Remote Control",
34+ "id": "5644"
35+ }
36+ },
37+ {
38+ "index": "product_catalog",
39+ "id": "SHDwvocuH7HSig2Xz14",
40+ "score": 0.19999999,
41+ "source": {
42+ "name": "Sony DSCQ Soft Carrying Camera Case - Black Finish - LSCSH",
43+ "description": "Sony DSCQ Soft Carrying Black Camera Case - LSCSH/ Polypropylene Construction/ Extra Compartments For Accessory Storage/ Can Hold Many Sony Or Other Manufacturers Digital Cameras/ Black Finish",
44+ "id": "24922"
45+ }
46+ },
47+ {
48+ "index": "product_catalog",
49+ "id": "SHDwvocuH7HSig2XzD4",
50+ "score": 0.19999999,
51+ "source": {
52+ "name": "Sony DSLR-A350 Digital Camera And 18-70mm Zoom Lens - DSCRA350K/ 14.2 Megapixel/ Tiltable LCD Plus Screen/ Smart Teleconverter 2X Zoom/ Blonz Image Processor/ Super SteadyShot In-Camera Image Stabilization/ Anti-Dust Technology/ Eye-Start Autofocus System/ Auto Pop-Up Flash/ Black Finish",
53+ "id": "34024"
54+ }
55+ }
56+ }
57+ }
58+ }

Introduction to Elasticsearch/Basics of Elasticsearch

Elasticsearch is a distributed, free and open search and analytics engine for all types of data, including textual, numerical, geospatial, structured, and unstructured. Elasticsearch is built on Apache Lucene and was first released in 2010 by Elasticsearch N.V. (now known as Elastic). Known for its simple REST APIs, distributed nature, speed, and scalability, Elasticsearch is the central component of the Elastic Stack, a set of free and open tools for data ingestion, enrichment, storage, analysis, and

visualization. Commonly referred to as the ELK Stack (after Elasticsearch, Logstash, and Kibana), the Elastic Stack now includes a rich collection of lightweight shipping agents known as Beats for sending data to Elasticsearch.

Elasticsearch architecture

Elasticsearch is a distributed search engine used for full-text search. In this section, you will understand the physical architecture of Elasticsearch. In which we will see how documents are distributed across the physical or virtual machine. Along with it, we will also see how machines work together to form a cluster.

In Elasticsearch architecture, node and cluster play an important role. These are the center of Elasticsearch architecture. Each node in a cluster handles the HTTP request for a client who wants to send the request to the cluster.

Full Text Search

The full text queries enable you to search analyzed text fields such as the body of an email. The query string is processed using the same analyzer that was applied to the field during indexing.

Basic search operations with Elasticsearch API

This API is used to search content in Elasticsearch. A user can search by sending a get request with query string as a parameter or they can post a query in the message body of post request. Mainly all the search APIS are multi-index, multi-type.

Full Text Queries and Intervals Query

The intervals query uses matching rules, constructed from a small set of definitions. These rules are then applied to terms from a specified field.

The definitions produce sequences of minimal intervals that span terms in a body of text. These intervals can be further combined and filtered by parent sources.

What is full text search FTS?

Full Text Search (FTS) refers to the process of searching and retrieving information from a body of text, such as documents, articles, web pages, or databases. It involves querying the text content and returning results based on the relevance and context of the search terms.

FTS systems are designed to handle natural language queries and provide accurate and comprehensive search results. They go beyond simple keyword matching and consider factors such as word frequency, stemming, synonyms, relevance ranking, and contextual analysis to deliver more meaningful and relevant search results.

Key components and features of Full Text Search include:

1. **Indexing:** FTS systems create an index of the text content, which allows for efficient searching and retrieval of relevant documents. The index stores information about the terms and their locations within the documents.

2. **Tokenization:** Text is divided into individual words or tokens, which are then processed and indexed. Tokenization involves splitting the text based on whitespace, punctuation, or other language-specific rules.
3. **Stemming and Normalization:** Stemming reduces words to their base or root form to account for variations like plurals or verb tenses. Normalization involves transforming words to a consistent format (e.g., converting to lowercase) to ensure accurate matching.
4. **Ranking and Relevance:** FTS systems often employ algorithms to rank the search results based on their relevance to the query. Various factors, such as term frequency, document length, and term proximity, may influence the ranking.
5. **Advanced Querying:** FTS systems typically support advanced querying capabilities, including Boolean operators (AND, OR, NOT), phrase searching, wildcard or fuzzy matching, and proximity searches to refine search criteria.

Full Text Search is widely used in search engines, document management systems, content management systems, e-commerce platforms, and database systems to provide powerful and efficient text-based search functionality. It helps users find relevant information quickly and effectively by analyzing and matching the content of the documents rather than relying solely on metadata or simple keyword matching.

Interval search query

An Interval Query, also known as an Interval Search or Interval Tree Query, is a type of query that focuses on finding intervals or ranges that overlap with or contain a specified range of values. It is commonly used in scenarios where data is organized in intervals or segments and retrieval of overlapping or intersecting intervals is required.

In an Interval Query, intervals are defined as pairs of values, typically representing a start point and an end point. Each interval has a specific range that can represent time intervals, numeric ranges, or any other type of interval-based data.

The main objective of an Interval Query is to identify intervals that intersect or overlap with a given query interval. The query can specify a range of values, and the Interval Query algorithm determines which intervals in the dataset have a non-empty intersection with the query range.

Interval Query algorithms are often based on efficient data structures called Interval Trees or Segment Trees. These data structures allow for quick searching and retrieval of intervals that overlap with a given range by recursively partitioning the intervals and maintaining appropriate indexes.

Interval Queries have various applications in different domains, such as scheduling systems, calendar applications, time-series data analysis, computational geometry, and database systems. They enable efficient retrieval of relevant intervals based on their

overlap or intersection, providing valuable insights and supporting decision-making processes.

ELASTIC SEARCH

WHAT IS ELASTIC SEARCH???

index
search engine
analytics database
kind of like google

{

Examples of Analytical Databases

Transactional data – Historical transactions that can include purchasing patterns for improved marketing.

Sensor data – Historical data from sensors that monitor situations like the weather.

Natural language data – Study of social media posts for research purposes

}

nosql database communicated through json requests

[elastic search = distributed nosql database in the form json documents]

json :- java script object notation -basically data representation or data format used in web development
used to send data to and from the serve.

search engines based on the "APACHE LUCEBE LIBRARY"

- used for log analytics,full text serach,security intelligence,business analytics and operational intelligence

1. What is Full Text Search (FTS)?

- A query language used for retrieving data from relational databases.
- A method used for searching and matching exact strings in a text document.
- A feature in databases that allows searching and matching of text documents based on relevance and ranking.
- A technique used for performing searches based on the similarity of text documents.

2. What is an Interval Query?

- A query that retrieves data based on a specific time range.
- A query that retrieves data based on overlapping or intersecting intervals.
- A query that retrieves data based on the length of intervals.
- A query that retrieves data based on the position of intervals within a text document.

2. What is an Interval Query?

1 point

- A query that retrieves data based on a specific time range.
- A query that retrieves data based on overlapping or intersecting intervals.
- A query that retrieves data based on the length of intervals.
- A query that retrieves data based on the position of intervals within a text document.

3. Which of the following is a common use case for Full Text Search?

1 point

- Searching for exact matches of keywords in a text document.
- Searching for relevant results based on the relevance and ranking of text documents.
- Searching for data based on specific time ranges.
- Searching for data based on overlapping or intersecting intervals.

3. Which of the following is a common use case for Full Text Search?

1 point

- Searching for exact matches of keywords in a text document.
- Searching for relevant results based on the relevance and ranking of text documents.
- Searching for data based on specific time ranges.
- Searching for data based on overlapping or intersecting intervals.

4. Which of the following is NOT a common feature of Full Text Search?

1 point

- Case sensitivity in matching search terms.
- Ranking of search results based on relevance.
- Support for word stemming and synonym matching.
- Search across multiple fields or attributes.

4. Which of the following is NOT a common feature of Full Text Search?

1 point

- Case sensitivity in matching search terms.
- Ranking of search results based on relevance.
- Support for word stemming and synonym matching.
- Search across multiple fields or attributes.

5. What is the purpose of an Interval Query in databases?

1 point

- To retrieve data based on exact matches of intervals.
- To retrieve data based on the position of intervals within a text document.
- To retrieve data based on the length of intervals.
- To retrieve data based on overlapping or intersecting intervals.

1. What is Full Text Search (FTS)?

1 / 1 point

- A query language used for retrieving data from relational databases.
- A method used for searching and matching exact strings in a text document.
- A feature in databases that allows searching and matching of text documents based on relevance and ranking.
- A technique used for performing searches based on the similarity of text documents.

 **Correct**

Correct. Full Text Search (FTS) is a feature in databases that allows searching and matching of text documents based on relevance and ranking. It is commonly used in applications such as search engines, content management systems, and data analytics.

2. What is an Interval Query?

1 / 1 point

- A query that retrieves data based on a specific time range.
- A query that retrieves data based on overlapping or intersecting intervals.
- A query that retrieves data based on the length of intervals.
- A query that retrieves data based on the position of intervals within a text document.

 **Correct**

Correct. An Interval Query is a query that retrieves data based on overlapping or intersecting intervals. It is commonly used in applications where data is organized into time intervals, such as scheduling, event management, and time-series data analysis.

3. Which of the following is a common use case for Full Text Search?

1 / 1 point

- Searching for exact matches of keywords in a text document.
- Searching for relevant results based on the relevance and ranking of text documents.
- Searching for data based on specific time ranges.
- Searching for data based on overlapping or intersecting intervals.

 **Correct**

Correct. Full Text Search is commonly used for searching and retrieving relevant results based on the relevance and ranking of text documents. It allows users to search for relevant information even if the exact keywords are not present in the document, by considering factors such as word proximity, term frequency, and document relevance.

4. Which of the following is NOT a common feature of Full Text Search?

1 / 1 point

- Case sensitivity in matching search terms.
- Ranking of search results based on relevance.
- Support for word stemming and synonym matching.
- Search across multiple fields or attributes.

 **Correct**

Correct. Full Text Search typically includes options for case-insensitive matching of search terms. Case sensitivity in matching search terms is not a common feature of Full Text Search, as it may lead to inconsistent and incomplete search results.

5. What is the purpose of an Interval Query in databases?

1 / 1 point

- To retrieve data based on exact matches of intervals.
- To retrieve data based on the position of intervals within a text document.
- To retrieve data based on the length of intervals.
- To retrieve data based on overlapping or intersecting intervals.

 **Correct**

Correct. The purpose of an Interval Query in databases is to retrieve data based on overlapping or intersecting intervals. It is commonly used in applications where data is organized into time intervals or ranges, such as scheduling, event management, and time-series data analysis.

Match Query

- the standard query for performing a full-text search, including options for fuzzy matching.
- Returns documents that match a provided text, number, date or boolean value

```
1 GET /product_catalog/_search
2 {
3   "query": {
4     "match": {
5       "name": "Panasonic"
6     }
7   }
8 }
```

Interval Query - Matching Rules

- prefix
- wildcard
- match
- all_of
- any_of
- fuzzy

Prefix query

Console Search Profiler Grok Debugger Painless Lab BETA

History Settings Variables Help

```
1 #Painless
2
3 GET /product_catalog/_search
4
5 "query": {
6   "intervals": {
7     "name": {
8       "prefix": {
9         "prefix": "PL"
10      }
11    }
12  }
13 }
```

200 OK 303 ms

```
1+ "hits": {
2+   "total": {
3+     "value": 73,
4+     "relation": "eq"
5+   },
6+   "max_score": 0.6666666,
7+   "hits": [
8+     {
9+       "_index": "product_catalog",
10+      "_id": "VdxwvocBuTHIgDXzL4",
11+      "_score": 0.6666666,
12+      "_source": {
13+        "name": "Frigidaire 24' PLD4375RFC Built-In Stainless Steel Dishwasher - PLD437555",
14+        "description": "Frigidaire 24' PLD4375RFC Built-In Stainless Steel Dishwasher - PLD437555/ Power Dry Drying System/ GraniteGrey Interior/ 5-Level Precision Direct Wash System/ Self-Cleaning Filter/ Smart Soil Sensor/ Stainless Steel Finish",
15+        "id": "2599"
16+      }
17+    },
18+    {
19+      "_index": "product_catalog",
20+      "_id": "VdxwvocBuTHIgDXzL4",
21+      "_score": 0.6666666,
22+      "_source": {
23+        "name": "Frigidaire Professional 24' PLD4555RFC Built-In Stainless Steel Dishwasher - PLD455555",
24+        "description": "Frigidaire Professional 24' PLD4555RFC Built-In Stainless Steel Dishwasher - PLD455555/ 12 Wash Cycles/ Intellisense/ Dry Intelligence/ Sabbath System/ Slimline Sound Insulation Package/ 12 Easy Clean Electronic Touchpads Digital Display/ 5-Level Precision Direct Wash System With AquaSurge And Variable Washing Pressure/ Smart Soil Sensor/ Stainless Steel Finish",
25+        "id": "27389"
26+      }
27+    },
28+    {
29+      "_index": "product_catalog",
30+      "_id": "VdxwvocBuTHIgDXzL4",
31+      "_score": 0.6666666,
32+      "_source": {
33+        "name": "Frigidaire Professional 24' PLD4555RFC Built-In Stainless Steel Dishwasher - PLD455555",
34+        "description": "Frigidaire Professional 24' PLD4555RFC Built-In Stainless Steel Dishwasher - PLD455555/ 12 Wash Cycles/ Intellisense/ Dry Intelligence/ Sabbath System/ Slimline Sound Insulation Package/ 12 Easy Clean Electronic Touchpads Digital Display/ 5-Level Precision Direct Wash System With AquaSurge And Variable Washing Pressure/ Smart Soil Sensor/ Stainless Steel Finish",
35+        "id": "27389"
36+      }
37+    },
38+    {
39+      "_index": "product_catalog",
40+      "_id": "VdxwvocBuTHIgDXzL3",
41+      "_score": 0.5,
42+      "_source": {
43+        "name": "Sony 5 Disc CD Player - CDPCE375",
44+        "description": "Sony 5 Disc CD Player - CDPCE375/ 5 Disc Changer/ Variable Line Output/ CD-R/RW Playback Capability/ 20 Track Music Calendar/ Digital Servo Control/ Remote Commander Remote Control",
45+        "id": "5644"
46+      }
47+    },
48+    {
49+      "_index": "product_catalog",
50+      "_id": "VdxwvocBuTHIgDXzL3",
51+      "_score": 0.5,
52+    }
53+  ]
54+ }
```

Wild card query

Console Search Profiler Grok Debugger Painless Lab BETA

History Settings Variables Help

```
1 #Charger or Changer
2
3 GET /product_catalog/_search
4
5 "query": {
6   "intervals": {
7     "name": {
8       "wildcard": {
9         "pattern": "Charger"
10      }
11    }
12  }
13 }
```

200 OK 313 ms

```
1+ {
2+   "took": 2,
3+   "timed_out": false,
4+   "shards": 1,
5+   "total": 1,
6+   "successful": 1,
7+   "skipped": 0,
8+   "failed": 0
9+ },
10+ "hits": {
11+   "total": {
12+     "value": 13,
13+     "relation": "eq"
14+   },
15+   "max_score": 0.5,
16+   "hits": [
17+     {
18+       "_index": "product_catalog",
19+       "_id": "CdxwvocBuTHIgDXzL3",
20+       "_score": 0.5,
21+       "_source": {
22+         "name": "Sony 30 Disc CD Changer - CDPCK355",
23+         "description": "Sony 30 Disc CD Changer - CDPCK355/ MegaStorage Control/ CD-R/CD-RW Playback/ 32 Step Program Play/ Control At 11/ Custom File Memo/ CD Text/ Keyboard Input/ 2 Jog Shuts/ Remote Control/ Black Finish",
24+         "id": "8552"
25+       }
26+     },
27+     {
28+       "_index": "product_catalog",
29+       "_id": "CdxwvocBuTHIgDXzL3",
30+       "_score": 0.5,
31+       "_source": {
32+         "name": "Sony 30 Disc MegaStorage CD Changer - CDPCK455",
33+         "description": "Sony 30 Disc MegaStorage CD Changer - CDPCK455/ MP3 Playback Capability/ CD-R/CD-RW Playback/ Twin Jog Dial For Easy Disc Access/ MegaChanger Control/ Optical Digital Output/ Keyboard Input/ Remote Commander/ Remote Control",
34+         "id": "9071"
35+       }
36+     }
37+   ]
38+ }
```

Wild card way-2

Console Search Profiler Grok Debugger Painless Lab BETA

History Settings Variables Help

```

1 #Charger or Changer
2
3 GET /product_catalog/_search
4 {
5   "query": {
6     "intervals": {
7       "name": {
8         "wildcard": {
9           "pattern": "charger"
10          }
11        }
12      }
13    }
14  }

```

200 - OK 348 ms

```

48   "score": 0.5,
49   "index": "Sony Xplod 10-Disc Add-On CD/MP3 Changer - DCK5650KF",
50   "id": "B00vocu01H6lgi2XfL3",
51   "source": {
52     "name": "Onkyo 6 Disc CD Player Changer - DCK390B",
53     "description": "Onkyo 6 Disc CD Player Changer - DCK390B/ Multi Single-Bit DAC/ 6 Repeat Modes/ 40 Random Selections/ Direct Track Access/ 96 db S/N Ratio/ 5 Hz-20 kHz Frequency Response/ Remote Control Included/ Black Finish",
54     "id": "13945"
55   }
56 }
57
58   "index": "product_catalog",
59   "id": "B00vocu01H6lgi2XfL4",
60   "score": 0.5,
61   "source": {
62     "name": "Olympus Olympus NL-MH Quick Charger And Battery Set - B96SU",
63     "description": "Olympus Olympus NL-MH Quick Charger And Battery Set - B96SU/ Ultra-Compact Size And Light Weight/ AC Cable/ Charges Four AA Batteries In Around 230 Minutes/ 4 x 1300mAh AA Batteries Included",
64     "id": "16774"
65   }
66
67   "index": "product_catalog",
68   "id": "B00vocu01H6lgi2XfL4",
69   "score": 0.5,
70   "source": {
71     "name": "Sony 400-Disc DVD/SA-CD/Digital Mega Changer - DVPCX995V",
72     "description": "Sony 400-Disc DVD, SACD & CD Mega Changer - DVPCX995V/ Super Audio CD Playback/ DVD-Rw/DVD-R/ CD-Rw/ CD-R/ MP3 Playback/ Dolby Digital And Dts/ Advanced SmoothScan Technology/ Instant Replay/ Back Ground Graphics/ Black Finish",
73     "id": "19672"
74   }
75
76   "index": "product_catalog",
77   "id": "B00vocu01H6lgi2XfL4",
78   "score": 0.5

```

Console Search Profiler Grok Debugger Painless Lab BETA

History Settings Variables Help

```

1 GET /product_catalog/_search
2 {
3   "query": {
4     "intervals": {
5       "description": {
6         "match": {
7           "query": "up to",
8           "ordered": true
9         }
10       }
11     }
12   }
13 }
14
15 GET /product_catalog/_search
16 {
17   "query": {
18     "match": {
19       "description": {
20         "query": "Black Finish",
21         "operator": "or"
22       }
23     }
24   }
25 }
26
27 GET /product_catalog/_search
28 {
29   "size": 500,
30   "query": {
31     "intervals": {
32       "description": {
33         "match": {
34           "query": "Black Finish",
35           "max_pager": 8
36         }
37       }
38     }
39   }
40 }

```

200 - OK 421 ms

```

1 {
2   "took": 1,
3   "timed_out": false,
4   "_shards": {
5     "total": 1,
6     "successful": 1,
7     "failed": 0
8   },
9   "hits": {
10     "total": {
11       "value": 833,
12       "relation": "eq"
13     },
14     "max_score": 1.4548919,
15     "hits": [
16       {
17         "index": "product_catalog",
18         "id": "B00vocu01H6lgi2XfP4",
19         "score": 1.4548919,
20         "source": {
21           "name": "Canon Deluxe Soft Black Camera Case - 07168001",
22           "description": "Canon Deluxe Soft Black Camera Case - 07168001/ Compatible with Canon PowerShot S2 IS/ Black Finish",
23           "id": "36928"
24         }
25       },
26       {
27         "index": "product_catalog",
28         "id": "B00vocu01H6lgi2XfP4",
29         "score": 1.426652,
30         "source": {
31           "name": "Nikon COOLPIX Leather Camera Case in Black - 5811",
32           "description": "Nikon COOLPIX Leather Camera Case in Black - 5811/ Compatible With Selected S Series COOLPIX cameras/ Black Finish",
33           "id": "29832"
34         }
35       },
36       {
37         "index": "product_catalog",
38         "id": "B00vocu01H6lgi2XfP4",
39         "score": 1.426652,
40         "source": {
41           "name": "Sony Black Soft Carrying Case - LCSX38",
42           "description": "Sony Black Soft Carrying Case - LCSX38",
43           "id": "34156"
44         }
45       }
46     ]
47   }
48 }
49
50   "index": "product_catalog",
51   "id": "B00vocu01H6lgi2XfP4",
52   "score": 1.426652,
53   "source": {
54     "name": "Transcend 4GB Micro Secure Digital Memory Card - TS4GUSDHCS",
55     "description": "Transcend 4GB Micro Secure Digital Memory Card - TS4GUSDHCS/ 4GB MicroSD Card With Adapter/ Compatible With Cell Phones And Mobile Devices With MicroSD Input/ Mechanical Write Protection Switch/ Includes Adapter To Fit SD Card Slot For Data Transfer/ Compliant With Secure Digital Music Initiative/ Black Finish",
56     "id": "34156"
57   }
58
59   "index": "product_catalog",
60   "id": "B00vocu01H6lgi2XfP4",
61   "score": 1.426652,
62   "source": {
63     "name": "Samsung Stereo Bluetooth Headset In Black - SBH500",
64     "description": "Samsung Stereo Bluetooth Headset - SBH500/ Integrated Microphone/ Receive Inbound Calls/ Answer/ End Calls/ Volume Control/ 10 Hours Talk Time/ 11 Hours Play Time/ 205 Hours Of Standby Time/ Send & End Function/ 3-way Calling/ Wireless Stereo Music Streaming/ Black Finish",
65     "id": "36127"
66   }
67
68   "index": "product_catalog",
69   "id": "B00vocu01H6lgi2XfP4",
70   "score": 1.426652

```

Console Search Profiler Grok Debugger Painless Lab BETA

History Settings Variables Help

```

1 GET /product_catalog/_search
2 {
3   "query": {
4     "intervals": {
5       "description": {
6         "all": {
7           "intervals": [
8             {
9               "match": {
10                 "query": "Black Finish",
11                 "ordered": true
12               }
13             },
14             {
15               "match": {
16                 "query": "Music"
17               }
18             }
19           ]
20         }
21       }
22     }
23   }
24 }
25
26 GET /product_catalog/_search
27 {
28   "query": {
29     "intervals": {
30       "description": {
31         "match": {
32           "query": "Black Finish Music",
33           "ordered": true
34         }
35       }
36     }
37   }
38 }

```

200 - OK 745 ms

```

1 {
2   "took": 2,
3   "timed_out": false,
4   "_shards": {
5     "total": 1,
6     "successful": 1,
7     "failed": 0,
8     "skipped": 0,
9     "failed": 0
10   },
11   "hits": {
12     "total": {
13       "value": 17,
14       "relation": "eq"
15     },
16     "max_score": 0.3333333,
17     "hits": [
18       {
19         "index": "product_catalog",
20         "id": "B00vocu01H6lgi2XfP4",
21         "score": 0.3333333,
22         "source": {
23           "name": "Transcend 4GB Micro Secure Digital Memory Card - TS4GUSDHCS",
24           "description": "Transcend 4GB Micro Secure Digital Memory Card - TS4GUSDHCS/ 4GB MicroSD Card With Adapter/ Compatible With Cell Phones And Mobile Devices With MicroSD Input/ Mechanical Write Protection Switch/ Includes Adapter To Fit SD Card Slot For Data Transfer/ Compliant With Secure Digital Music Initiative/ Black Finish",
25           "id": "34156"
26         }
27       },
28       {
29         "index": "product_catalog",
30         "id": "B00vocu01H6lgi2XfP4",
31         "score": 0.3333333,
32         "source": {
33           "name": "Samsung Stereo Bluetooth Headset In Black - SBH500",
34           "description": "Samsung Stereo Bluetooth Headset - SBH500/ Integrated Microphone/ Receive Inbound Calls/ Answer/ End Calls/ Volume Control/ 10 Hours Talk Time/ 11 Hours Play Time/ 205 Hours Of Standby Time/ Send & End Function/ 3-way Calling/ Wireless Stereo Music Streaming/ Black Finish",
35           "id": "36127"
36         }
37       }
38     ]
39   }
40 }

```

```

History Settings Variables Help
1 GET /product_catalog/_search
2 {
3   "query": {
4     "intervals": {
5       "operator": "any_of",
6       "intervals": [
7         {
8           "match": {
9             "query": "Black Finish",
10            "ordered": true
11          }
12        },
13        {
14          "match": {
15            "query": "Music"
16          }
17        }
18      ]
19    }
20  }
21 }
22 }
23 }
24 }
25
26 GET /product_catalog/_search
27 {
28   "query": {
29     "match": {
30       "operator": "or",
31       "query": "Black Finish Music",
32     }
33   }
34 }
35 }
36 }
37 }

2 * 1
3   "took": 1,
4   "timed_out": false,
5   "_shards": {
6     "total": 1,
7     "successful": 1,
8     "skipped": 0,
9     "failed": 0
10  }
11  "hits": {
12    "total": {
13      "value": 535,
14      "relation": "eq"
15    },
16    "max_score": 4.715847,
17    "hits": [
18      {
19        "_index": "product_catalog",
20        "_id": "wH0vooBuTHSiq2XzI4",
21        "_score": 4.715847,
22        "_source": {
23          "name": "Sony Bluetooth Adaptor/Portable Transmitter - THRBT10",
24          "description": "Sony Bluetooth Adaptor/Portable Transmitter - THRBT10/ Listen Up To 11 Hours Of Music/ Connects Wirelessly From Digital Music Player To Compatible Devices/ Bluetooth Technology/ Gold Plated Plug/ Rechargeable",
25          "id": "32822"
26        }
27      },
28      {
29        "_index": "product_catalog",
30        "_id": "wH0vooBuTHSiq2XzI4",
31        "_score": 4.2340217,
32        "_source": {
33          "name": "Sony Mylo Black Personal Communicator - COMBLACK",
34          "description": "Sony Mylo Black Personal Communicator - COMBLACK/ 3.5\" MVA LCD Touch Screen/ 1.3 Megapixel Camera/ Face Calling With Skype/ IEEE 802.11b/g Wireless LAN/ Music And Video Player/ Memory Stick Duo /PRO Duo Expansion Slot/ 1GB Internal Memory/ Built-in Microphone/ Black Finish",
35          "id": "32561"
36        }
37      }
38      {
39        "_index": "product_catalog",
40        "_id": "wH0vooBuTHSiq2XzI4",
41        "_score": 4.2340217,
42        "_source": {
43          "name": "Yamaha YAS-71 Black Front Surround System - YAS71BK",
44          "id": "32561"
45        }
46      }
47    ]
48  }
49 }
50 }
51 }

2 * 1
3   "took": 1,
4   "timed_out": false,
5   "_shards": {
6     "total": 1,
7     "successful": 1,
8     "skipped": 0,
9     "failed": 0
10  }
11  "hits": {
12    "total": {
13      "value": 4,
14      "relation": "eq"
15    },
16    "max_score": 1.6860131,
17    "hits": [
18      {
19        "_index": "test_index_1",
20        "_id": "7Qbx4cB519qWVzAB5a",
21        "_score": 1.6860131,
22        "_source": {
23          "id": 2,
24          "test_string": "The big wolf"
25        }
26      },
27      {
28        "_index": "test_index_1",
29        "_id": "wBVax4cBuTHSiq2g8r",
30        "_score": 1.498163,
31        "_source": {
32          "id": 1,
33          "test_string": "The big bad wolf"
34        }
35      },
36      {
37        "_index": "test_index_1",
38        "_id": "7XkxzobUThSiq2JA-Q",
39        "_score": 1.498163,
40        "_source": {
41          "id": 10,
42          "test_string": "The big lonely wolf"
43        }
44      },
45      {
46        "_index": "test_index_1",
47        "_id": "88VzkocBuThSiq2Pg-l",
48        "_score": 1.498163,
49        "_source": {
50          "id": 11,
51          "test_string": "The big good wolf"
52        }
53      }
54    ]
55  }
56 }
57 }

200 - OK 2219 ms

```

Search query with filters

Big and wolf together into the test strings of the documents into the indices

```

History Settings Variables Help
1 GET /test_index_1/_search
2
3 GET /test_index_1/_search
4 {
5   "query": {
6     "match": {
7       "test_string": {
8         "query": "big wolf",
9         "operator": "and"
10      }
11    }
12  }
13 }
14
15 GET /test_index_1/_search
16 {
17   "query": {
18     "intervals": {
19       "test_string": {
20         "match": {
21           "query": "big wolf",
22           "max_gaps": 0
23         }
24       }
25     }
26   }
27 }
28
29 GET /test_index_1/_search
30 {
31   "query": {
32     "intervals": {
33       "test_string": {
34         "match": {
35           "query": "big wolf",
36           "filter": {
37             "not_containing": {
38               "match": {
39                 "query": "bad"
40               }
41             }
42           }
43         }
44       }
45     }
46   }
47 }
48
49 GET /test_index_1/_search
50 {
51   "query": {
52     "intervals": {
53       "test_string": {
54         "match": {
55           "query": "bad"
56         }
57       }
58     }
59   }
60 }

2 * 1
3   "took": 1,
4   "timed_out": false,
5   "_shards": {
6     "total": 1,
7     "successful": 1,
8     "skipped": 0,
9     "failed": 0
10  }
11  "hits": {
12    "total": {
13      "value": 4,
14      "relation": "eq"
15    },
16    "max_score": 1.6860131,
17    "hits": [
18      {
19        "_index": "test_index_1",
20        "_id": "7Qbx4cB519qWVzAB5a",
21        "_score": 1.6860131,
22        "_source": {
23          "id": 2,
24          "test_string": "The big wolf"
25        }
26      },
27      {
28        "_index": "test_index_1",
29        "_id": "wBVax4cBuTHSiq2g8r",
30        "_score": 1.498163,
31        "_source": {
32          "id": 1,
33          "test_string": "The big bad wolf"
34        }
35      },
36      {
37        "_index": "test_index_1",
38        "_id": "7XkxzobUThSiq2JA-Q",
39        "_score": 1.498163,
40        "_source": {
41          "id": 10,
42          "test_string": "The big lonely wolf"
43        }
44      },
45      {
46        "_index": "test_index_1",
47        "_id": "88VzkocBuThSiq2Pg-l",
48        "_score": 1.498163,
49        "_source": {
50          "id": 11,
51          "test_string": "The big good wolf"
52        }
53      }
54    ]
55  }
56 }
57 }

2 * 1
3   "took": 1,
4   "timed_out": false,
5   "_shards": {
6     "total": 1,
7     "successful": 1,
8     "skipped": 0,
9     "failed": 0
10  }
11  "hits": {
12    "total": {
13      "value": 4,
14      "relation": "eq"
15    },
16    "max_score": 1.6860131,
17    "hits": [
18      {
19        "_index": "test_index_1",
20        "_id": "7Qbx4cB519qWVzAB5a",
21        "_score": 1.6860131,
22        "_source": {
23          "id": 2,
24          "test_string": "The big wolf"
25        }
26      },
27      {
28        "_index": "test_index_1",
29        "_id": "wBVax4cBuTHSiq2g8r",
30        "_score": 1.498163,
31        "_source": {
32          "id": 1,
33          "test_string": "The big bad wolf"
34        }
35      },
36      {
37        "_index": "test_index_1",
38        "_id": "7XkxzobUThSiq2JA-Q",
39        "_score": 1.498163,
40        "_source": {
41          "id": 10,
42          "test_string": "The big lonely wolf"
43        }
44      },
45      {
46        "_index": "test_index_1",
47        "_id": "88VzkocBuThSiq2Pg-l",
48        "_score": 1.498163,
49        "_source": {
50          "id": 11,
51          "test_string": "The big good wolf"
52        }
53      }
54    ]
55  }
56 }
57 }

200 - OK 2219 ms

```

Searches the strings with big wolf but not includes the bad word in the test string the documents

```

History Settings Variables Help
1 GET /test_index_1/_search
2
3 GET /test_index_1/_search
4 {
5   "query": {
6     "match": {
7       "test_string": {
8         "query": "big wolf",
9         "operator": "and"
10      }
11    }
12  }
13 }
14 GET /test_index_1/_search
15 {
16   "query": {
17     "intervals": {
18       "test_string": {
19         "match": {
20           "query": "big wolf",
21           "max_gaps": 0
22         }
23       }
24     }
25   }
26 }
27 }
28 GET /test_index_1/_search
29 {
30   "query": {
31     "intervals": {
32       "test_string": {
33         "match": {
34           "query": "big wolf"
35         }
36       }
37     }
38   }
39 }
40 }
41 }
42 }
43 }
44 }
45 }
46 }
47 }
48 }
49 GET /test_index_1/_search
50 {
51   "query": {
52     "intervals": {
53       "test_string": {
54         "match": {
55           "query": "hot"
56         }
57       }
58     }
59   }
60 }
61 }
62 }
63 }
64 }
65 }
66 }
67 }

1+ {
2  "took": 0,
3  "timed_out": false,
4  "_shards": {
5    "total": 1,
6    "successful": 1,
7    "skipped": 0,
8    "failed": 0
9  },
10 }
11 }
12 }
13 }
14 }
15 }
16 }
17 }
18 }
19 }
20 }
21 }
22 }
23 }
24 }
25 }
26 }
27 }
28 }
29 }
30 }
31 }
32 }
33 }
34 }
35 }
36 }
37 }
38 }
39 }
40 }
41 }
42 }
43 }
44 }
45 }
46 }
47 }

1+ [
2   {
3     "took": 0,
4     "timed_out": false,
5     "_shards": {
6       "total": 1,
7       "successful": 1,
8       "skipped": 0,
9       "failed": 0
10     },
11     "hits": [
12       {
13         "total": {
14           "value": 3,
15           "relation": "eq"
16         },
17         "max_score": 0.5,
18         "hits": [
19           {
20             "_index": "test_index_1",
21             "_id": "7gbxv4cB519qDvzAB5a",
22             "_score": 0.5,
23             "_source": {
24               "id": 3,
25               "test_string": "The big wolf"
26             }
27           }
28         ]
29       }
30     ]
31   }
32 }
33 }
34 }
35 }
36 }
37 }
38 }
39 }
40 }
41 }
42 }
43 }
44 }
45 }
46 ]

```

Documents with test strings

With match of “hot” keyword before the “soup” key word of the documents in the index

```

History Settings Variables Help
32+   "test_soup": {
33   "match": {
34     "query": "big wolf",
35     "filter": {
36       "not_containing": {
37         "match": {
38           "query": "bad"
39         }
40       }
41     }
42   }
43 }
44 }
45 }
46 }
47 }
48 }
49 GET /test_index_1/_search
50 {
51   "query": {
52     "intervals": {
53       "test_string": {
54         "match": {
55           "query": "hot",
56           "filter": [
57             {
58               "before": {
59                 "match": {
60                   "query": "soup"
61                 }
62               }
63             }
64           ]
65         }
66       }
67     }
68   }
69 }

1+ [
2   {
3     "took": 1,
4     "timed_out": false,
5     "_shards": {
6       "total": 1,
7       "successful": 1,
8       "skipped": 0,
9       "failed": 0
10     },
11     "hits": [
12       {
13         "total": {
14           "value": 1,
15           "relation": "eq"
16         },
17         "max_score": 0.5,
18         "hits": [
19           {
20             "_index": "test_index_1",
21             "_id": "98bDzocBu7HSig2lQ8n",
22             "_score": 0.5,
23             "_source": {
24               "id": 14,
25               "test_string": "hot salty soup"
26             }
27           }
28         ]
29       }
30     ]
31   }
32 }
33 }
34 }
35 }
36 }
37 }
38 }
39 }
40 }
41 }
42 }
43 }
44 }
45 }
46 }
47 }

1+ [
2   {
3     "took": 1,
4     "timed_out": false,
5     "_shards": {
6       "total": 1,
7       "successful": 1,
8       "skipped": 0,
9       "failed": 0
10     },
11     "hits": [
12       {
13         "total": {
14           "value": 1,
15           "relation": "eq"
16         },
17         "max_score": 0.5,
18         "hits": [
19           {
20             "_index": "test_index_1",
21             "_id": "98bDzocBu7HSig2lQ8n",
22             "_score": 0.5,
23             "_source": {
24               "id": 14,
25               "test_string": "hot salty soup"
26             }
27           }
28         ]
29       }
30     ]
31   }
32 }
33 }
34 }
35 }
36 }
37 }
38 }
39 }
40 }
41 }
42 }
43 }
44 }
45 }
46 }
47 ]

```

Overlapping parameter of the filter query

Containing “salty” keyword match with the overlapping keywords of the “cold porridge”

The screenshot shows the Elasticsearch Painless Lab interface. The top navigation bar includes 'Console', 'Search Profiler', 'Grok Debugger', 'Painless Lab (BETA)', 'History', 'Settings', 'Variables', and 'Help'. The main area has two panes: a code editor on the left and a results panel on the right.

```

22 *     "test_string": {
23 *       "match": {
24 *         "query": "big wolf",
25 *         "filter": {
26 *           "not_containing": {
27 *             "match": {
28 *               "query": "bad"
29 *             }
30 *           }
31 *         }
32 *       }
33 *     }
34 *   }
35 * }
36 *
37 * }
38 * }
39 * }
40 * }
41 * }
42 * }
43 * }
44 * }
45 * }
46 * }
47 * }
48 *
49 GET /test_index_1/_search
50 {
51   "query": {
52     "intervals": {
53       "test_string": {
54         "match": {
55           "query": "hot",
56           "filter": {
57             "before": {
58               "match": {
59                 "query": "soup"
60               }
61             }
62           }
63         }
64       }
65     }
66   }
67 }
68
69 GET /test_index_1/_search
70 {
71   "query": {
72     "intervals": {
73       "test_string": {
74         "match": {
75           "query": "salty",
76           "filter": {
77             "overlapping": {
78               "match": {
79                 "query": "cold porridge"
80               }
81             }
82           }
83         }
84       }
85     }
86   }
87 }

```

The results panel displays the response from the search query:

```

1- {
2-   "took": 0,
3-   "timed_out": false,
4-   "_shards": {
5-     "total": 1,
6-     "successful": 1,
7-     "skipped": 0,
8-     "failed": 0
9-   },
10-  "hits": {
11-    "total": {
12-      "value": 1,
13-      "relation": "eq"
14-    },
15-    "max_score": 0.5,
16-    "hits": [
17-      {
18-        "index": "test_index_1",
19-        "id": "BqDzocBS19qUVvzDR60",
20-        "score": 0.5,
21-        "source": {
22-          "id": 13,
23-          "test_string": "cold salty porridge"
24-        }
25-      }
26-    ]
27-  }
28- }

```

Reading comprehension

Match Query and its implementation

Returns documents that match a provided text, number, date or boolean value. The provided text is analyzed before matching.

The match query is the standard query for performing a full-text search, including options for fuzzy matching.

[Read more](#)

Intervals query rules

The intervals query uses matching rules, constructed from a small set of definitions. These rules are then applied to terms from a specified field.

The definitions produce sequences of minimal intervals that span terms in a body of text. These intervals can be further combined and filtered by parent sources.

[Read more](#)

Implementing Intervals Query Using Filters

A filter in Elasticsearch is all about applying some conditions inside the query that are used to narrow down the matching result set.

1. What is the purpose of implementing Match and Intervals Query in a database?

1 / 1 point

- To retrieve data based on exact matches of intervals.
- To retrieve data based on the position of intervals within a text document.
- To retrieve data based on the length of intervals.
- To retrieve data based on overlapping or intersecting intervals.

 **Correct**

Correct. The purpose of implementing Match and Intervals Query in a database is to retrieve data based on overlapping or intersecting intervals. This allows for advanced querying capabilities when dealing with time intervals, ranges, or other types of data that require complex matching logic.

2. How can Intervals Query be implemented using filters in a database?

1 / 1 point

- By using regular expressions to filter data.
- By using range queries to filter data.
- By using custom filters or libraries that support interval querying.
- By using full-text search to filter data.

 **Correct**

Correct. Intervals Query can be implemented using custom filters or libraries that provide support for interval querying. These filters or libraries may offer specific functions or methods to filter data based on overlapping or intersecting intervals.

3. What is the purpose of implementing Minimization in interval-based queries?

1 / 1 point

- To reduce the storage space required for interval data.
- To reduce the number of intervals that need to be considered in queries.
- To reduce the number of queries required to retrieve interval data.
- To reduce the complexity of interval-based queries.

 **Correct**

Correct. Minimization in interval-based queries aims to reduce the number of intervals that need to be considered during querying, which can improve query performance and reduce computational overhead.

4. What are some common challenges in implementing Match and Intervals Query in a database?

1 / 1 point

- Managing large datasets.
- Ensuring efficient query performance.
- Dealing with inconsistent or incomplete data.
- Ensuring data security and privacy.

 **Correct**

Correct. Efficient query performance can be a common challenge in implementing Match and Intervals Query, as dealing with overlapping or intersecting intervals may require complex querying logic and optimization techniques to achieve efficient performance.

5. Which of the following is a potential benefit of implementing Match and Intervals Query in a database?

1 / 1 point

- Advanced querying capabilities for overlapping or intersecting intervals.
- Improved query performance for interval-based data.
- Simplified data management for interval-based data.
- Enhanced data security and privacy for interval-based data.

 **Correct**

Correct. Implementing Match and Intervals Query in a database can provide advanced querying capabilities for overlapping or intersecting intervals, allowing for more sophisticated data retrieval and analysis.

1. What is an Intervals Query in Elasticsearch?

1 point

- A search technique that retrieves only the metadata of documents.
- A query that retrieves documents based on their relevance score.
- A query that retrieves documents based on the occurrence of terms within specific intervals.
- A query that retrieves documents based on their publication date.

2. Which Elasticsearch API can be used to implement an Intervals Query?

1 point

- Search API
- Aggregations API
- Analysis API
- Ingest API

3. What does the "match" query do in Elasticsearch?

1 point

- Retrieves documents that match a specific field value exactly.
- Retrieves documents that contain a specific term in a specific field.
- Retrieves documents that have the highest relevance score.
- Retrieves documents that were indexed most recently.

4. What is the purpose of the "slop" parameter in the "match" query in Elasticsearch?

1 point

- It specifies the maximum number of terms that can be skipped in a phrase match.
- It specifies the minimum number of terms that must match in a phrase query.
- It specifies the maximum distance between terms in a phrase match.
- It specifies the maximum number of documents that can be retrieved in a single search query.

5. What is the purpose of implementing Minimization in Elasticsearch?

1 point

- To reduce the storage space used by Elasticsearch.
- To improve the indexing speed of documents in Elasticsearch.
- To optimize the performance of search queries in Elasticsearch.
- To enhance the security features of Elasticsearch.

6. What is an example use case for implementing Intervals Query in Elasticsearch?

1 point

- Searching for documents that have a specific field value.
- Searching for documents that contain a specific phrase with variable term positions.
- Searching for documents that were indexed within a specific time range.
- Searching for documents that have a specific relevance score.

7. Which type of query in Elasticsearch can be used to implement an Intervals Query with strict ordering of terms?

1 point

- Bool Query
- Match Query
- Phrase Query
- Range Query

8. Which Elasticsearch filter can be used to implement an Intervals Query?

1 point

- Match Filter
- Range Filter
- Interval Filter
- Exists Filter

9. What is the purpose of the "max_gaps" parameter in the Interval Filter in Elasticsearch?

1 point

- It specifies the maximum number of gaps allowed between terms in an interval.
- It sets the minimum number of gaps allowed between terms in an interval.
- It defines the range of values for the interval.
- It sets the boost factor for the interval.

9. What is the purpose of the "max_gaps" parameter in the Interval Filter in Elasticsearch?

1 point

- It specifies the maximum number of gaps allowed between terms in an interval.
- It sets the minimum number of gaps allowed between terms in an interval.
- It defines the range of values for the interval.
- It sets the boost factor for the interval.

10. What is the main advantage of using an Interval Filter over an Interval Query in Elasticsearch?

1 point

- Interval Filters are faster in terms of query execution time.
- Interval Filters are more flexible in terms of term positions within the interval.
- Interval Filters provide better relevancy scoring for search results.
- Interval Filters can be used as a standalone filter, while Interval Queries need to be combined with other queries.

11. Which Elasticsearch query can be used to implement a search query with multiple Intervals Queries combined with boolean logic?

1 point

- Range Query
- Match Query
- Phrase Query
- Bool Query

11. Which Elasticsearch query can be used to implement a search query with multiple Intervals Queries combined with boolean logic?

1 point

- Range Query
- Match Query
- Phrase Query
- Bool Query

12. What is the purpose of the "use_field" parameter in the Intervals Query in Elasticsearch?

1 point

- It specifies the name of the field to be used for the intervals.
- It sets the boost factor for the Intervals Query.
- It defines the range of values for the intervals.
- It specifies how the intervals should be constructed.

13. Which Elasticsearch query can be used to search for documents that contain terms in close proximity to each other?

1 point

- Prefix Query
- Wildcard Query
- Term Query
- Span Query

14. Which Elasticsearch query can be used to search for documents that contain terms that have the same prefix?

1 point

- Fuzzy Query
- Match Query
- Wildcard Query
- Prefix Query

15. Which Elasticsearch query can be used to search for documents that contain terms that occur within a specific range?

1 point

- Match Query
- Wildcard Query
- Term Query
- Range Query

16. What is the purpose of implementing the "match" query in Elasticsearch?

1 point

- To search for documents that contain an exact term in a specific field.
- To search for documents that contain terms that match the query string, using a full-text search approach.
- To search for documents that contain terms that have the same prefix.
- To search for documents that contain terms in close proximity to each other.

17. Which Elasticsearch query can be used to search for documents that contain terms that match a specified pattern, using regular expressions?

1 point

- Wildcard Query
- Regexp Query
- Fuzzy Query
- Prefix Query

18. What is the purpose of implementing the "intervals" query in Elasticsearch?

1 point

- To search for documents that contain terms in close proximity to each other.
- To search for documents that contain terms that match a specified pattern based on exact word positions.
- To search for documents that contain terms that have the same prefix.
- To search for documents that contain terms that match a specified pattern based on a combination of character filters, token filters, and terms.

19. What is the purpose of implementing the "minimization" technique in Elasticsearch?

1 point

- To reduce the size of the Elasticsearch index by removing unnecessary metadata.
- To optimize the performance of Elasticsearch queries by caching frequently used queries.
- To improve the relevancy of Elasticsearch search results by boosting certain terms or fields.
- To optimize the performance of Elasticsearch queries by simplifying complex queries and reducing unnecessary processing.

18. What is the purpose of implementing the "intervals" query in Elasticsearch?

1 point

- To search for documents that contain terms in close proximity to each other.
- To search for documents that contain terms that match a specified pattern based on exact word positions.
- To search for documents that contain terms that have the same prefix.
- To search for documents that contain terms that match a specified pattern based on a combination of character filters, token filters, and terms.

19. What is the purpose of implementing the "minimization" technique in Elasticsearch?

1 point

- To reduce the size of the Elasticsearch index by removing unnecessary metadata.
- To optimize the performance of Elasticsearch queries by caching frequently used queries.
- To improve the relevancy of Elasticsearch search results by boosting certain terms or fields.
- To optimize the performance of Elasticsearch queries by simplifying complex queries and reducing unnecessary processing.

20. Which Elasticsearch query can be used to search for documents that contain terms that match a specified query string, but with the ability to specify different levels of fuzziness?

1 point

- Match Query
- Wildcard Query
- Fuzzy Query
- Query String Query

1. What is an Intervals Query in Elasticsearch?

1 / 1 point

- A search technique that retrieves only the metadata of documents.
- A query that retrieves documents based on their relevance score.
- A query that retrieves documents based on the occurrence of terms within specific intervals.
- A query that retrieves documents based on their publication date.

 **Correct**

Correct. An Intervals Query in Elasticsearch is used to search for documents based on the occurrence of terms within specific intervals.

2. Which Elasticsearch API can be used to implement an Intervals Query?

1 / 1 point

- Search API
- Aggregations API
- Analysis API
- Ingest API

 **Correct**

Correct. The Search API in Elasticsearch is used to perform searches, including Intervals Queries.

3. What does the "match" query do in Elasticsearch?

1 / 1 point

- Retrieves documents that match a specific field value exactly.
- Retrieves documents that contain a specific term in a specific field.
- Retrieves documents that have the highest relevance score.
- Retrieves documents that were indexed most recently.

 **Correct**

Correct. The "match" query in Elasticsearch is used to search for documents that contain a specific term in a specific field.

4. What is the purpose of the "slop" parameter in the "match" query in Elasticsearch?

0 / 1 point

- It specifies the maximum number of terms that can be skipped in a phrase match.
- It specifies the minimum number of terms that must match in a phrase query.
- It specifies the maximum distance between terms in a phrase match.
- It specifies the maximum number of documents that can be retrieved in a single search query.

 **Incorrect**

Incorrect. The "slop" parameter in the "match" query specifies the maximum number of positions that terms can be moved to match a phrase query, not the number of terms that can be skipped.

5. What is the purpose of implementing Minimization in Elasticsearch?

0 / 1 point

- To reduce the storage space used by Elasticsearch.
- To improve the indexing speed of documents in Elasticsearch.
- To optimize the performance of search queries in Elasticsearch.
- To enhance the security features of Elasticsearch.

 **Incorrect**

Incorrect. Minimization in Elasticsearch is not related to reducing storage space, but rather optimizing the performance of search queries.

6. What is an example use case for implementing Intervals Query in Elasticsearch?

1 / 1 point

- Searching for documents that have a specific field value.
- Searching for documents that contain a specific phrase with variable term positions.
- Searching for documents that were indexed within a specific time range.
- Searching for documents that have a specific relevance score.

 **Correct**

Correct. An example use case for implementing Intervals Query in Elasticsearch is to search for documents that contain a specific phrase with variable term positions, such as searching for "quick brown fox" with "brown" and "fox" having one or more terms in between.

7. Which type of query in Elasticsearch can be used to implement an Intervals Query with strict ordering of terms?

1 / 1 point

- Bool Query
- Match Query
- Phrase Query
- Range Query

 **Correct**

Correct. Phrase Query in Elasticsearch can be used to implement an Intervals Query with strict ordering of terms, as it allows specifying the terms with exact positions in a phrase.

8. Which Elasticsearch filter can be used to implement an Intervals Query?

0 / 1 point

- Match Filter
- Range Filter
- Interval Filter
- Exists Filter

 **Incorrect**

Incorrect. Match Filter in Elasticsearch is used to filter documents based on a specific term in a specific field, and does not specifically support Intervals Query.

9. What is the purpose of the "max_gaps" parameter in the Interval Filter in Elasticsearch?

1 / 1 point

- It specifies the maximum number of gaps allowed between terms in an interval.
- It sets the minimum number of gaps allowed between terms in an interval.
- It defines the range of values for the interval.
- It sets the boost factor for the interval.

 **Correct**

Correct. The "max_gaps" parameter in the Interval Filter in Elasticsearch is used to specify the maximum number of gaps allowed between terms in an interval. This helps in controlling the flexibility of term positions within the specified interval.

10. What is the main advantage of using an Interval Filter over an Interval Query in Elasticsearch?

1 / 1 point

- Interval Filters are faster in terms of query execution time.
- Interval Filters are more flexible in terms of term positions within the interval.
- Interval Filters provide better relevancy scoring for search results.
- Interval Filters can be used as a standalone filter, while Interval Queries need to be combined with other queries.

 **Correct**

Correct. One advantage of using an Interval Filter over an Interval Query in Elasticsearch is that Interval Filters can be used as a standalone filter, while Interval Queries need to be combined with other queries, such as Bool Query or Dis Max Query, to form a complete search query.

11. Which Elasticsearch query can be used to implement a search query with multiple Intervals Queries combined with boolean logic?

1 / 1 point

- Range Query
- Match Query
- Phrase Query
- Bool Query

 **Correct**

Correct. Bool Query in Elasticsearch is a compound query that combines multiple queries with boolean logic, and can be used to implement a search query with multiple Intervals Queries combined with boolean logic.

12. What is the purpose of the "use_field" parameter in the Intervals Query in Elasticsearch?

0 / 1 point

- It specifies the name of the field to be used for the intervals.
- It sets the boost factor for the Intervals Query.
- It defines the range of values for the intervals.
- It specifies how the intervals should be constructed.

 **Incorrect**

Incorrect. The "use_field" parameter in the Intervals Query in Elasticsearch is not related to specifying the name of the field to be used for the intervals, but rather specifying the type of interval to be used.

13. Which Elasticsearch query can be used to search for documents that contain terms in close proximity to each other?

1 / 1 point

- Prefix Query
- Wildcard Query
- Term Query
- Span Query

 **Correct**

Correct. Span Query in Elasticsearch is a type of query that allows searching for terms in close proximity to each other, using various span queries such as Span Near Query, Span Term Query, Span First Query, etc.

14. Which Elasticsearch query can be used to search for documents that contain terms that have the same prefix?

1 / 1 point

- Fuzzy Query
- Match Query
- Wildcard Query
- Prefix Query

 **Correct**

Correct. Prefix Query in Elasticsearch is used to search for documents that contain terms with a specific prefix, and can be used to search for terms that have the same prefix.

15. Which Elasticsearch query can be used to search for documents that contain terms that occur within a specific range?

1 / 1 point

- Match Query
- Wildcard Query
- Term Query
- Range Query

 **Correct**

Correct. Range Query in Elasticsearch is used to search for documents that contain terms within a specific range, such as numeric values or date ranges. It allows specifying the lower and upper bounds for the range and can be used to search for terms that occur within that range.

16. What is the purpose of implementing the "match" query in Elasticsearch?

1 / 1 point

- To search for documents that contain an exact term in a specific field.
- To search for documents that contain terms that match the query string, using a full-text search approach.
- To search for documents that contain terms that have the same prefix.
- To search for documents that contain terms in close proximity to each other.

 **Correct**

Correct. The "match" query in Elasticsearch is used to search for documents that contain terms that match the query string, using a full-text search approach. It takes into account the analysis process applied to the terms in the specified field, such as tokenization, normalization, and stemming, to match the terms in the query string.

17. Which Elasticsearch query can be used to search for documents that contain terms that match a specified pattern, using regular expressions?

1 / 1 point

- Wildcard Query
- Regexp Query
- Fuzzy Query
- Prefix Query

 **Correct**

Correct. Regexp Query in Elasticsearch is used to search for documents that contain terms that match a specified pattern, using regular expressions. It allows specifying a regular expression pattern as the query string, and documents containing terms that match that pattern will be returned in the search results.

18. What is the purpose of implementing the "intervals" query in Elasticsearch?

0 / 1 point

- To search for documents that contain terms in close proximity to each other.
- To search for documents that contain terms that match a specified pattern based on exact word positions.
- To search for documents that contain terms that have the same prefix.
- To search for documents that contain terms that match a specified pattern based on a combination of character filters, token filters, and terms.

 **Incorrect**

Incorrect. The "intervals" query in Elasticsearch is not specifically used to search for terms in close proximity to each other, but rather to search for documents that contain terms that match a specified pattern based on intervals within the text.

19. What is the purpose of implementing the "minimization" technique in Elasticsearch?

0 / 1 point

- To reduce the size of the Elasticsearch index by removing unnecessary metadata.
- To optimize the performance of Elasticsearch queries by caching frequently used queries.
- To improve the relevancy of Elasticsearch search results by boosting certain terms or fields.
- To optimize the performance of Elasticsearch queries by simplifying complex queries and reducing unnecessary processing.

 **Incorrect**

Incorrect. The "minimization" technique in Elasticsearch is not used to reduce the size of the index by removing metadata, but rather to reduce the size of the query by removing unnecessary clauses and filters.

20. Which Elasticsearch query can be used to search for documents that contain terms that match a specified query string, but with the ability to specify different levels of fuzziness?

0 / 1 point

- Match Query
- Wildcard Query
- Fuzzy Query
- Query String Query

 **Incorrect**

Incorrect. Fuzzy Query in Elasticsearch does provide the ability to specify different levels of fuzziness, but it does not match a specified query string, but rather searches for documents that contain terms that are similar to a specific term based on a fuzzy matching algorithm.

Fuzziness



- match approximate search terms in addition to exact matches

color → colour



- useful for accommodating spelling mistakes, variations, or typos in search terms

Eiffel Tover → Eiffel Tower

```

History Settings Variables Help
1 #Fuzzy searching using match query
2
3 #fuzziness, fuzzy_transpositions, prefix_length and fuzzy_rewrite.
4
5 GET /product_catalog/_search
6 {
7   "size": 100,
8   "query": {
9     "match": {
10       "name": {
11         "query": "Hnome",
12         "fuzziness": 1,
13         "fuzzy_transpositions": "true"
14       }
15     }
16   }
17 }
18
19 GET /product_catalog/_search
20 {
21   "size": 100,
22   "query": {
23     "intervals": {
24       "name": {
25         "term": "Hnome",
26         "fuzziness": 1,
27         "transpositions": true
28       }
29     }
30   }
31 }
32
33
34
35 #Intervals -> Boolean similiy score -> matching Intervals -> higher
36
37 #match -> fuzzy algorithm -> single character edits -> Insertions, deletions, substitutions or even transpositions. The relevance score -> TF/IDF algorithm. -> term frequency-inverse document frequency -> frequency of the search term in the document + as well as in the collection/index
38
39
40 Intervals -> term frequency -> in the document
41 Match -> term frequency -> in the document + as well in the index
42
43

```

```

1 {
2   "took": 1,
3   "timed_out": false,
4   "shards": [
5     {"total": 1,
6      "successful": 1,
7      "failed": 0
8    }
9  ],
10 "hits": [
11   {
12     "total": {
13       "value": 25,
14       "relation": "eq"
15     },
16     "max_score": 2.9582854,
17     "hits": [
18       {
19         "index": "product_catalog",
20         "id": "R9WvocBu7HSig2Xp4",
21         "score": 2.9582854,
22         "source": {
23           "name": "SIRIUS External Home Antenna - 1424B",
24           "description": "SIRIUS External Home Antenna - 1424B/ Adjustable Design/ Durable UV-Coated Weather Resistant Construction/ Included Is A 30 Cable And Complete Mounting Hardware/ Compatible With Single Input SIRIUS Home Tuners And Plug & Play Receivers",
25           "id": "28963"
26         }
27       },
28       {
29         "index": "product_catalog",
30         "id": "R9WvocBu7HSig2Xp4",
31         "score": 2.789672,
32         "source": {
33           "name": "Sony BRAVIA Home Theater System - DAVHDX275",
34           "description": "Sony BRAVIA Home Theater System - DAVHDX275/ 5.1 Channel Surround/ 1080p Upscaling Via HDMI/ 1000Watts 5-Disc DVD/CD Player/ Multi-Room S-AIR Technology/ BRAVIA Sync/ iPod Cradle/ Digital Media Port/ Digital Cinema Auto Calibration/ Multi Speaker Formation/ Black and Silver Finish",
35           "id": "33998"
36         }
37       },
38       {
39         "index": "product_catalog",
40         "id": "R9WvocBu7HSig2Xp4",
41         "score": 2.6456435,
42         "source": {
43           "name": "Sony Home Entertainment Server In Black - HESV1000",
44           "description": "Sony Home Entertainment Server In Black - HESV1000/ 200 BD/DVD/CD Changer With Full HD 1080P Video Output/ 500GB Hard-Drive/ Store Up To 137 Hours of Video, 2000 Music Albums, Or 20,000 Photos/ XMB"
45         }
46       }
47     ]
48   }
49 }

```

Fuzzy Searching using Intervals Query

The fuzzy rule matches terms that are similar to the provided term, within an edit distance defined by Fuzziness. If the fuzzy expansion matches more than 128 terms, Elasticsearch returns an error.

[Read more](#)

Fuzzy Searching using Match Query

Match Query with fuzziness parameter is more preferable than Fuzzy Query. The analyzer in the query will analyze your query before searching it into the Inverted Index.

[Read more](#)

Special Parameter for the Search Field

The `search_fields` parameter restricts a query to search only specific fields.

Restricting fields will result in faster queries, especially for schemas with many text fields.

[Read more](#)

1. What is Query Analytics?

1 / 1 point

- A method for optimizing query performance in a database.
- A technique for encrypting query data to enhance data security.
- A process for identifying and fixing errors in database queries.
- A practice of analyzing and understanding query patterns, usage, and effectiveness for improving search functionality.

 **Correct**

Correct. Query analytics involves analyzing and understanding query patterns, usage, and effectiveness for improving search functionality, which can include techniques like fuzziness, Levenshtein Edit Distance, and special parameters for search fields.

2. What is fuzziness in the context of query analytics?

1 / 1 point

- A technique for optimizing query performance in a database.
- A concept that allows for approximate matches in search queries, accounting for variations in spelling, word order, etc.
- A process for identifying and fixing errors in database queries.
- A technique for encrypting query data to enhance data security.

 **Correct**

Correct. Fuzziness in query analytics involves allowing for approximate matches in search queries, accounting for variations in spelling, word order, and other similar factors to improve search functionality.

3. What is Levenshtein Edit Distance?

1 / 1 point

- A measure of the similarity between two strings, defined as the minimum number of single-character edits (insertions, deletions, or substitutions) required to change one string into the other.
- A technique for optimizing query performance in a database.
- A process for identifying and fixing errors in database queries.
- A technique for encrypting query data to enhance data security.

 **Correct**

Correct. Levenshtein Edit Distance is a measure of the similarity between two strings, defined as the minimum number of single-character edits (insertions, deletions, or substitutions) required to change one string into the other. It is often used in query analytics to calculate the similarity between a search query and a database entry, allowing for approximate matches.

4. What is a special parameter for the search field in the context of query analytics?

1 / 1 point

- A parameter that allows for optimizing query performance in a database.
- A parameter that encrypts query data to enhance data security.
- A parameter that identifies and fixes errors in database queries.
- A parameter that allows for customizing the behavior of the search field, such as specifying search modes, boosting certain fields, or setting query-time parameters.

 **Correct**

Correct. A special parameter for the search field in query analytics refers to a parameter that allows for customizing the behavior of the search field, such as specifying search modes, boosting certain fields, or setting query-time parameters. This can help improve the search functionality and relevance of search results.

5. What is the main purpose of implementing intervals query using filters in query analytics?

1 / 1 point

- To optimize query performance by reducing the number of search queries.
- To encrypt query data for enhanced data security.
- To identify and fix errors in database queries.
- To refine search results by filtering them based on specific intervals or ranges of values.

 **Correct**

Correct. The main purpose of implementing intervals query using filters in query analytics is to refine search results by filtering them based on specific intervals or ranges of values. This can help narrow down search results and provide more precise and relevant search results based on specific criteria or ranges.

Relevance Score

Relevance score is a measure of how well a document matches a particular search query

Relevance Score

How does Elasticsearch calculate the score that is used to order the documents returned by a query?

Relevance Score

BM25/Best Match 25

Term Frequency

More number of times → Relevance Score

Inverse document frequency

More number of times → Relevance Score

Field-length

Longer the field → Relevance Score

```
1 PUT /demo_idx/_alias
2
3 PUT demo_idx/_doc/1
4 {
5   "content": "Distributed nature, simple REST APIs, speed, and scalability"
6 }
7 PUT demo_idx/_doc/2
8 {
9   "content": "Known for its simple REST APIs, distributed nature, speed, and scalability, Elasticsearch is the central
10 component of the Elastic Stack, a set of open source tools for data ingestion, enrichment, storage, analysis, and
11 visualization."
12 }
13
14 }
15
16 GET demo_idx/_search
17 {
18   "query": {
19     "match": {
20       "content": {
21         "query": "simple rest apis distributed nature",
22         "operator": "and"
23       }
24     }
25 }
26 }
```

```
1+ { "took": 0,
2+ "timed_out": false,
3+ "_shards": {
4+   "total": 1,
5+   "successful": 1,
6+   "skipped": 0,
7+   "failed": 0
8+ },
9+ "hits": {
10+   "total": {
11+     "value": 3,
12+     "relation": "eq"
13+   },
14+   "max_score": 1.2689934,
15+   "hits": [
16+     {
17+       "_index": "demo_idx",
18+       "_id": "1",
19+       "_score": 1.2689934,
20+       "_source": {
21+         "content": "Distributed nature, simple REST APIs, speed, and scalability"
22+       }
23+     },
24+     {
25+       "_index": "demo_idx",
26+       "_id": "2",
27+       "_score": 0.6970792,
28+       "_source": {
29+         "content": "Known for its simple REST APIs, distributed nature, speed, and scalability, Elasticsearch is the
30+         central component of the Elastic Stack, a set of open source tools for data ingestion, enrichment, storage,
31+         analysis, and visualization."
32+     }
33+   ]
34+ },
35+ "version": 6,
36+ "source": {
37+   "content": "Known for its simple REST APIs, distributed nature, speed, and scalability, Elasticsearch is the
38+         central component of the Elastic Stack, a set of open source tools for data ingestion, enrichment, storage,
39+         analysis, and visualization."
40+ }
41+ }
42+ }
43+ }
```

```
1 PUT /demo_idx/_alias
2
3 PUT demo_idx/_doc/1
4 {
5   "content": "Distributed nature, simple REST APIs, speed, and scalability"
6 }
7 PUT demo_idx/_doc/2
8 {
9   "content": "Known for its simple REST APIs, distributed nature, speed, and scalability, Elasticsearch is the central
10 component of the Elastic Stack, a set of open source tools for data ingestion, enrichment, storage, analysis, and
11 visualization."
12 }
13
14 }
15
16 GET demo_idx/_search
17 {
18   "query": {
19     "match_phrase": {
20       "content": {
21         "query": "simple rest apis distributed nature"
22       }
23     }
24 }
25 }
```

```
1+ { "took": 0,
2+ "timed_out": false,
3+ "_shards": {
4+   "total": 1,
5+   "successful": 1,
6+   "skipped": 0,
7+   "failed": 0
8+ },
9+ "hits": {
10+   "total": {
11+     "value": 1,
12+     "relation": "eq"
13+   },
14+   "max_score": 0.6961101,
15+   "hits": [
16+     {
17+       "_index": "demo_idx",
18+       "_id": "3",
19+       "_score": 0.6961101,
20+       "_source": {
21+         "content": "Known for its simple REST APIs, distributed nature, speed, and scalability, Elasticsearch is the
22+           central component of the Elastic Stack, a set of open source tools for data ingestion, enrichment, storage,
23+           analysis, and visualization."
24+     }
25+   ]
26+ }
```

```

1 PUT /demo_idx/_alias
2
3 PUT demo_idx/_doc/1
4 {
5   "content": "distributed nature, simple REST APIs, speed, and scalability"
6 }
7
8 PUT demo_idx/_doc/2
9 {
10   "content": "distributed nature, simple APIs, speed, and scalability"
11 }
12
13 PUT demo_idx/_doc/3
14 {
15   "content": "Known for its simple REST APIs, distributed nature, speed, and scalability, Elasticsearch is the central component of the Elastic Stack, a set of open source tools for data ingestion, enrichment, storage, analysis, and visualization."
16 }
17
18 GET demo_idx/_search
19 {
20   "query": {
21     "bool": {
22       "should": [
23         {
24           "match": {
25             "content": {
26               "query": "simple rest apis distributed nature"
27             }
28           }
29         },
30         {
31           "match": {
32             "content": {
33               "query": "simple rest apis distributed nature",
34               "operator": "and"
35             }
36           }
37         },
38         {
39           "match_phrase": {
40             "content": {
41               "query": "simple rest apis distributed nature"
42             }
43           }
44         }
45       ]
46     }
47   }
48 }
```

Explain API

why a specific document matches (or doesn't match) a query

computes a score explanation for a query and a specific document

This can give useful feedback why a document matches or doesn't match a specific query

Further reading on Query Optimization

Understanding Query Optimization in Elasticsearch

Elasticsearch can query, filter and aggregate in many ways. Often there are several ways to solve the same problem – and possibly with very different performance characteristics.

[Read more](#)

Using the Explain API to Identify Query Performance Bottlenecks

The explain API computes a score explanation for a query and a specific document. This can give useful feedback on whether a document matches or didn't match a specific query.

[Read more](#)

Improving Query Performance with Filters and Caching in Elasticsearch

Elasticsearch can query, filter and aggregate in many ways. Often there are several ways to solve the same problem – and possibly with very different performance characteristics. This article will cover some important optimizations that can buy you a lot of performance.

[Read more](#)

Boosting Fields to Improve Search Result Relevance in Elasticsearch

Out of the box, App Search provides quality search relevance. Built on-top of Elasticsearch, App Search is a managed, expertly crafted distillation of its finest points.

It provides tools to help you further tune the search experience to optimize for your own needs.

Relevance Tuning has two core components: Weights and Boosts. Before we get into them, we shall take a quick dive into the basics of fields.

[Read more](#)

```

7+ PUT demo_idx/_doc/_id3
8+ {
9+   "content": "Distributed nature, simple APIs, speed, and scalability"
10+ }
11+ PUT demo_idx/_doc/3
12+ {
13+   "content": "Known for its simple REST APIs, distributed nature, speed, and scalability, Elasticsearch is the central component of the Elastic Stack, a set of open source tools for data ingestion, enrichment, storage, analysis, and visualization."
14+ }
15+ POST demo_idx/_explain/2
16+ {
17+   "query": {
18+     "bool": {
19+       "should": [
20+         {
21+           "match": {
22+             "content": {
23+               "query": "simple rest apis distributed nature"
24+             }
25+           }
26+         },
27+         {
28+           "match": {
29+             "content": {
30+               "query": "simple rest apis distributed nature",
31+               "operator": "and"
32+             }
33+           }
34+         },
35+         {
36+           "match_phrase": {
37+             "content": {
38+               "query": "simple rest apis distributed nature",
39+               "boost": 2
40+             }
41+           }
42+         }
43+       ]
44+     }
45+   }
46+ }
47+
48+ GET demo_idx/_explain/2
49+ {
50+   "query": {
51+     "match": {
52+       "content": {
53+         "query": "simple rest apis distributed nature",
54+         "operator": "and"
55+       }
56+     }
57+   }
58+ }
59+

```

The screenshot shows a code editor with a syntax-highlighted JSON document. The code consists of several lines of JSON, primarily focused on a search query and its explanation. The JSON includes fields like 'content' and 'operator', and uses nested structures for 'match' and 'match_phrase' queries. The code is annotated with line numbers from 1 to 59. A vertical scroll bar is visible on the right side of the editor window.

1. What is the purpose of using the Explain API in Elasticsearch to identify query performance bottlenecks?

1 / 1 point

- To optimize query performance by reducing the number of search queries.
- To encrypt query data for enhanced data security.
- To identify and fix errors in database queries.
- To provide detailed information about how a query is scored and executed, helping identify potential performance bottlenecks and optimize query performance.

 **Correct**

Correct. The main purpose of using the Explain API in Elasticsearch is to provide detailed information about how a query is scored and executed, helping identify potential performance bottlenecks and optimize query performance. This can help in improving the overall search performance of Elasticsearch queries.

2. How can query performance be improved with filters and caching in Elasticsearch?

1 / 1 point

- By reducing the number of search queries.
- By encrypting query data for enhanced data security.
- By storing frequently used results in memory for faster retrieval and filtering documents based on specific criteria to reduce the number of documents that need to be considered for a query.

 **Correct**

Correct. Query performance can be improved with filters and caching in Elasticsearch by storing frequently used results in memory for faster retrieval and filtering documents based on specific criteria to reduce the number of documents that need to be considered for a query. This can help in speeding up query execution and improving overall search performance.

3. What is the main purpose of Elasticsearch's scoring algorithm in optimizing search result relevance?

1 / 1 point

- To optimize query performance by reducing the number of search queries.
- To encrypt query data for enhanced data security.
- To identify and fix errors in database queries.
- To determine the relevance of search results based on various factors such as term frequency, inverse document frequency, and field length, helping optimize search result relevance.

 **Correct**

Correct. The main purpose of Elasticsearch's scoring algorithm is to determine the relevance of search results based on various factors such as term frequency, inverse document frequency, and field length. This helps in optimizing search result relevance by ranking the search results based on their relevance to the query, allowing users to find the most relevant results for their search queries.

4. What is the purpose of boosting fields in Elasticsearch to improve search result relevance?

1 / 1 point

- To optimize query performance by reducing the number of search queries.
- To encrypt query data for enhanced data security.
- To identify and fix errors in database queries.
- To assign higher importance or relevance to certain fields in the search results, helping improve search result relevance by influencing the ranking of search results.

 **Correct**

Correct. The main purpose of boosting fields in Elasticsearch is to assign higher importance or relevance to certain fields in the search results, helping improve search result relevance by influencing the ranking of search results. This allows users to control the relevance of different fields in the search results and prioritize certain fields over others.

5. What is the Levenshtein Edit Distance used for in Elasticsearch?

1 / 1 point

- To optimize query performance by reducing the number of search queries.
- To encrypt query data for enhanced data security.
- To identify and fix errors in database queries.
- To measure the similarity between two strings based on the minimum number of edit operations (insertions, deletions, substitutions) required to transform one string into another, allowing for fuzzy matching and approximate string matching in search queries.

 **Correct**

Correct. The main purpose of the Levenshtein Edit Distance in Elasticsearch is to measure the similarity between two strings based on the minimum number of edit operations (insertions, deletions, substitutions) required to transform one string into another. This allows for fuzzy matching and approximate string matching in search queries, helping improve the search functionality by considering slight variations or errors in search terms.

```

History Settings Variables Help
1 GET /test_index_1/_search
2 {
3   "query": {
4     "match_phrase": {
5       "test_string": {
6         "query": "big wolf",
7         "slop": 1
8       }
9     }
10   }
11 }

1  {
2   "took": 1,
3   "timed_out": false,
4   "shards": [
5     {
6       "total": 1,
7       "successful": 1,
8       "skipped": 0,
9       "failed": 0
10      }
11    ]
12  }
13  {
14    "hits": {
15      "total": {
16        "value": 4,
17        "relation": "eq"
18      },
19      "max_score": 1.6860131,
20      "hits": [
21        {
22          "_index": "test_index_1",
23          "_id": "7qpbx4cB519qUVvtAB5a",
24          "_score": 1.6860131,
25          "_source": {
26            "id": 2,
27            "test_string": "The big wolf"
28          }
29        },
30        {
31          "_index": "test_index_1",
32          "_id": "NBVax4cBu17H5Ig26g8r",
33          "_score": 0.9520669,
34          "_source": {
35            "id": 1,
36            "test_string": "The big bad wolf"
37          }
38        },
39        {
40          "_index": "test_index_1",
41          "_id": "7xvkzocBu17H5Ig234-0",
42          "_score": 0.9520669,
43          "_source": {
44            "id": 10,
45            "test_string": "The big lonely wolf"
46          }
47        },
48        {
49          "_index": "test_index_1",
50          "_id": "88VzkzocBu17H5Ig2Pg-l",
51          "_score": 0.9520669,
52          "_source": {
53            "id": 11,
54            "test_string": "The big good wolf"
55          }
56        }
57      ]
58    }
59  }

```

SLOP → doesn't care about how many terms it came in between our search terms

While executing with the match_phrase query

Match_phrase_prefix combination of match_phrase and prefix query , searches for the match_phrase and searches for the last term as the prefix of the search

The screenshot shows the Elasticsearch Dev Tools interface. At the top, there are tabs for History, Settings, Variables, and Help. Below the tabs, a code editor displays a GET request to /test_index_1/_search with the following JSON query:

```

1 GET /test_index_1/_search
2 {
3   "query": {
4     "match_phrase": {
5       "test_string": {
6         "query": "big wolf",
7         "slop": 1
8       }
9     }
10  }
11

```

The results section shows the response from the search request. It includes the took time, total hits, and a list of four documents (hits) with their _index, _id, _score, and _source fields. The documents are:

- _index: "test_index_1", _id: "7qpbx4cB519qUVvtAB5a", _score: 1.6860131, _source: { _id: 2, "test_string": "The big wolf" }
- _index: "test_index_1", _id: "NBVax4cBu17H5Ig26g8r", _score: 0.9520669, _source: { _id: 1, "test_string": "The big bad wolf" }
- _index: "test_index_1", _id: "7xvkzocBu17H5Ig234-0", _score: 0.9520669, _source: { _id: 10, "test_string": "The big lonely wolf" }
- _index: "test_index_1", _id: "BBVvkzocBu17H5Ig2Pg-l", _score: 0.9520669, _source: { _id: 11, "test_string": "The big good wolf" }

Multi match query

Searching the term “camera” in different fields both in “name” and “description”

Multi-Match Query

builds on the match query to allow multi-field queries

The screenshot shows the Elasticsearch Dev Tools interface. A code editor displays a GET request to /product_catalog/_search with the following JSON query:

```

1 GET /product_catalog/_search
2 {
3   "query": {
4     "multi_match": {
5       "query": "camera",
6       "fields": [ "name", "description" ]
7     }
8   }

```

Types of Multi-Match Query

best_fields

most_fields

cross_fields

phrase

phrase_prefix

bool_prefix

Searching for some term in multiple different fields

Multi_match query (search)

```
History Settings Variables Help
1 GET /product_catalog/_search
2 {
3   "query": {
4     "multi_match": {
5       "query": "Black Finish",
6       "fields": ["name", "description"]
7     }
8   }
9 }
```

200 - OK 376 ms

```
1+ { "took": 3,
2+   "timed_out": false,
3+   "_shards": {
4+     "total": 1,
5+     "successful": 1,
6+     "skipped": 0,
7+     "failed": 0
8+   }
9+   "hits": {
10+     "total": {
11+       "value": 546,
12+       "relation": "eq"
13+     },
14+     "max_score": 5.8369797,
15+     "hits": [
16+       {
17+         "_index": "product_catalog",
18+         "_id": "10WwocBuI7HSig2Xl4",
19+         "_score": 5.8369797,
20+         "_source": {
21+           "name": "Sony Lightweight Tripod - Black Finish - VCTR640",
22+           "description": "Sony Lightweight Tripod - VCTR640/ Adjustable Height And Three-Way Head Allows You To Remove Your Camera Quickly And Easily/ Black Finish",
23+           "id": 23778
24+         }
25+       },
26+       {
27+         "_index": "product_catalog",
28+         "_id": "ERWwocBuI7HSig2Xl3",
29+         "_score": 4.7779246,
30+         "_source": {
31+           "name": "Garmin Deluxe Carrying Case - Black Finish - 0101023101",
32+           "description": "Garmin Deluxe Carrying Case - 0101023101/ Designed For Street Pilot III 2610/2650 GPS System/ Holds Your GPS System, Mounting Bracket, Cables, Batteries And Data Cards",
33+           "id": 14033
34+         }
35+       },
36+     ],
37+     "aggregations": {
38+       "Index": {
39+         "_index": "product_catalog",
40+         "_id": "QDwWwocBuI7HSig2Xl4",
41+         "_score": 4.5424037,
42+         "_source": {
43+           "name": "Panasonic Lithium Ion Rechargeable Battery - Black Finish - CGR5006A1B",
44+           "description": "Panasonic Lithium Ion Rechargeable Battery - CGR5006A1B/ Compatible With DMCFZ7 And DMCFZ30 Digital Cameras",
45+           "id": 24346
46+         }
47+       }
48+     }
49+   }
50+ }
```

History Settings Variables Help

```
1 GET /product_catalog/_search
2 {
3   "query": {
4     "multi_match": {
5       "query": "Black Finish",
6       "fields": ["name", "description"]
7     }
8   }
9 }
10 }
```

200 - OK 376 ms

```
1 { "took": 3,
2 "timed_out": false,
3 "_shards": {
4   "total": 1,
5   "successful": 1,
6   "skipped": 0,
7   "failed": 0
8 },
9 "hits": {
10   "total": {
11     "value": 546,
12     "relation": "eq"
13   },
14   "max_score": 5.0369797,
15   "hits": [
16     {
17       "_index": "product_catalog",
18       "_id": "UhdmcnBuTHGIpGxZ14",
19       "_score": 5.0369797,
20       "_source": {
21         "name": "Sony Lightweight Tripod - Black Finish - VCTR840",
22         "description": "Sony Lightweight Tripod - VCTR840/ Adjustable Height And Three-Way Head Allow For Creative Camera Angles/ Quick-Release Head Allows You To Remove Your Camera Quickly And Easily/ Black Finish",
23         "id": "25776"
24       }
25     },
26     {
27       "_index": "product_catalog",
28       "_id": "QdewocBuTHGIpGxZ13",
29       "_score": 4.7769246,
30       "_source": {
31         "name": "Garmin Deluxe Carrying Case - Black Finish - 0101083101",
32         "description": "Designed For Street Pilot III 2610/2650 GPS Systems/Holds Your GPS System, Mounting Bracket, Cables, Batteries And Data Cards",
33         "id": "14833"
34       }
35     },
36     {
37       "_index": "product_catalog",
38       "_id": "QdewocBuTHGIpGxZ14",
39       "_score": 5.0369797,
40       "_source": {
41         "name": "Panasonic Lithium Ion Rechargeable Battery - Black Finish - COR5086A1B",
42         "description": "Panasonic Lithium Ion Rechargeable Battery - COR5086A1B/ Compatible With DMCFZ7 And DMCFZ30 Digital Cameras",
43         "id": "24346"
44       }
45     }
46   ]
47 }
```

```
History Settings Variables Help
1 GET /name_index/_search
2 {
3     "query": {
4         "multi_match": {
5             "query": "Will Smith",
6             "fields": ["first_name", "last_name"]
7         }
8     }
9 }
```

I

```
1 {
2     "took": 2,
3     "timed_out": false,
4     "_shards": {
5         "total": 1,
6         "successful": 1,
7         "skipped": 0,
8         "failed": 0
9     },
10    "hits": {
11        "total": {
12            "value": 8,
13            "relation": "eq"
14        },
15        "max_score": 1.2809337,
16        "hits": [
17            {
18                "_index": "name_index",
19                "_id": "8e9d31c8519q0vzbB40",
20                "_score": 1.2809337,
21                "_source": {
22                    "first_name": "Will",
23                    "last_name": "Smith"
24                }
25            },
26            {
27                "_index": "name_index",
28                "_id": "866d31c8519q0VzzXx6X",
29                "_score": 1.2809337,
30                "_source": {
31                    "first_name": "Patrick",
32                    "last_name": "Will"
33                }
34            },
35            {
36                "_index": "name_index",
37                "_id": "K0g9d31c8519q0vz0h7f",
38                "_score": 1.2809337,
39                "_source": {
40                    "first_name": "will",
41                    "last_name": "Will"
42                }
43            },
44            {
45                "_index": "name_index",
46                "_id": "A9Hd31c817MSg29RF5",
47                "_score": 0.3254224,
48                "_source": {
49                    "first_name": "John",
50                    "last_name": "Smith"
51                }
52            },
53        ]
54    }
55    "total": {
56        "value": 8,
57        "relation": "eq"
58    }
59 }
```

The screenshot shows the Elasticsearch Dev Tools interface with two search requests and their results.

Top Search Request:

```

1 GET /name_index/_search
2 {
3   "query": {
4     "multi_match": {
5       "query": "Will Smith",
6       "fields": ["first_name", "last_name"],
7       "type": "cross_fields"
8     }
9   }
10 }

```

Result (Top Hit):

```

1 {
2   "took": 2,
3   "timed_out": false,
4   "_shards": {
5     "total": 1,
6     "successful": 1,
7     "skipped": 0,
8     "failed": 0
9   }
10 }
11 {
12   "hits": [
13     "total": {
14       "value": 8,
15       "relation": "eq"
16     },
17     "max_score": 1.6663561,
18     "hits": [
19       {
20         "_index": "name_index",
21         "_id": "9aq3lC8519qUvzb848",
22         "_score": 1.6663561,
23         "_source": [
24           {
25             "first_name": "Will",
26             "last_name": "Smith"
27           }
28         ],
29         "_index": "name_index",
30         "_id": "96p0lC8519qUvzbX6X",
31         "_score": 1.2889337,
32         "_source": [
33           {
34             "first_name": "Patrick",
35             "last_name": "Will"
36           }
37         ],
38         "_index": "name_index",
39         "_id": "Kqg3lC8519qUvzb0h7f",
40         "_score": 1.2889337,
41         "_source": [
42           {
43             "first_name": "Will",
44             "last_name": "Will"
45           }
46         ],
47         "_index": "name_index",
48         "_id": "AHm3lC8519qUTh5ig29RFs",
49         "_score": 0.3254224,
50         "_source": [
51           {
52             "first_name": "John",
53             "last_name": "Smith"
54           }
55         ],
56         "_index": "name_index",
57         "_id": "8uN3lC8519qUTh5ig29RFs"
58       }
59     ]
60   }
61 }

```

Bottom Search Request:

```

1 GET /example_index_3/_search
2 {
3   "query": {
4     "multi_match": {
5       "query": "the b",
6       "fields": ["title", "description"],
7       "type": "phrase_prefix"
8     }
9   }
10 }

```

Result (Top Hit):

```

1 {
2   "took": 1,
3   "timed_out": false,
4   "_shards": {
5     "total": 1,
6     "successful": 1,
7     "skipped": 0,
8     "failed": 0
9   }
10 }
11 {
12   "hits": [
13     "total": {
14       "value": 3,
15       "relation": "eq"
16     },
17     "max_score": 1.8898406,
18     "hits": [
19       {
20         "_index": "example_index_3",
21         "_id": "CBqg3lC8519qUvzb7Ed",
22         "_score": 1.8898406,
23         "_source": [
24           {
25             "title": "the big wolf",
26             "description": "There's a big wolf in the forest"
27           }
28         ],
29         "_index": "example_index_3",
30         "_id": "-aqC3lC8519qUvzb85Q",
31         "_score": 1.6844232,
32         "_source": [
33           {
34             "title": "The big bad wolf",
35             "description": "there's a big bad wolf hiding in the forest"
36           }
37         ],
38         "_index": "example_index_3",
39         "_id": "ghm3lC8519qUTh5ig2PMGw",
40         "_score": 1.6844232,
41         "_source": [
42           {
43             "title": "The big lonely wolf",
44             "description": "there's a big lonely wolf alone in the wild"
45           }
46         ]
47       }
48     ]
49   }
50 }

```

Further Reading on Multi-match Querying Mechanism

Match Phrase & Match Phrase Prefix

Returns documents that contain the words of a provided text, in the same order as provided. The last term of the provided text is treated as a prefix, matching any words that begin with that term.

[Read more](#)

Introduction to Multi-Match Query

The multi_match query builds on the match query to allow multi-field queries

[Read more](#)

Relevance Score

The relevance score is a positive floating point number, returned in the _score metadata field of the search API. The higher the _score, the more relevant the document. While each query type can calculate relevance scores differently, score calculation also depends on whether the query clause is run in a query or filter context.

[Read more](#)

Implementation of Multi-match Query Patterns

The multi-match (multi_match) query, as the name suggests, searches the query across multiple fields. For example, if we want to search for the word Java across the three fields title, synopsis, and tags, then the multi_matchquery is the answer. The following listing shows a query that searches for Java across these three fields.

[Read more](#)

1. What is the main difference between "Phrase" and "Phrase Prefix" in Elasticsearch's multi-match querying?

1 / 1 point

- "Phrase" queries search for exact matches of the entire search phrase, while "Phrase Prefix" queries search for partial matches of the search phrase.
- "Phrase" queries search for partial matches of the search phrase, while "Phrase Prefix" queries search for exact matches of the entire search phrase.
- "Phrase" queries search for exact matches of the entire search phrase, while "Phrase Prefix" queries search for exact matches of the search phrase.
- "Phrase" queries search for exact matches of the entire search phrase, while "Phrase Prefix" queries search for exact matches of the search phrase followed by additional terms, allowing for partial matches.

 **Correct**

Correct. The main difference between "Phrase" and "Phrase Prefix" in Elasticsearch's multi-match querying is that "Phrase" queries search for exact matches of the entire search phrase, while "Phrase Prefix" queries search for exact matches of the search phrase followed by additional terms, allowing for partial matches.

2. What is the relevance score in Elasticsearch?

1 / 1 point

- The score assigned to a query based on the total number of terms matched.
- The score assigned to a query based on the number of fields matched.
- The score assigned to a query based on the query execution time.
- The score assigned to a query based on the relevance and importance of the matching terms in the search results.

 **Correct**

Correct. The relevance score in Elasticsearch is assigned to a query based on the relevance and importance of the matching terms in the search results. It takes into account various factors such as term frequency, inverse document frequency, field length, and other relevancy metrics, to determine the relevance of a document to a given search query.

3. How can multi-match query patterns be implemented in Elasticsearch?

1 / 1 point

- By using phrase queries with the multi-match query.
- By using phrase prefix queries with the multi-match query.
- By using wildcard queries with the multi-match query.
- By specifying the desired patterns and settings in the multi-match query using the appropriate options.

 **Correct**

Correct. Multi-match query patterns in Elasticsearch can be implemented by specifying the desired patterns and settings in the multi-match query using the appropriate options. This allows for customization and fine-tuning of the query to match specific search requirements.

4. What can be done to boost fields and improve search result relevance in Elasticsearch?

1 / 1 point

- Using filters and caching in the query.
- Using the Explain API to analyze query performance.
- Using the Levenshtein edit distance for fuzzy matching.
- Using field boosting techniques such as boosting query clauses, boosting field values, or applying custom boost functions.

 **Correct**

Correct. Field boosting techniques such as boosting query clauses, boosting field values, or applying custom boost functions can be used to boost fields and improve search result relevance in Elasticsearch. These techniques allow for fine-tuning the relevance of fields in the search results, making them more prominent or influential in the scoring process.

Graded Quiz

1. Which Elasticsearch feature is used to improve query performance by filtering results based on specific criteria? 1 point

- Filters
- Boosting
- Multi-match Querying
- Relevance Score

2. What is the purpose of caching in Elasticsearch? 1 point

- To store query results for faster retrieval in subsequent requests
- To improve the relevance score of search results
- To optimize the indexing process
- To filter out irrelevant data from search results

3. How does Elasticsearch's scoring algorithm determine the relevance of search results? 1 point

- By assigning a relevance score to each document based on query match criteria
- By sorting documents alphabetically based on query match criteria
- By filtering out irrelevant documents based on query match criteria
- By boosting certain documents based on query match criteria

4. Which Elasticsearch feature is used to improve search result relevance by boosting certain fields?

1 point

- Filters
- Caching
- Multi-match Querying
- Boosting Fields

5. What is the difference between a phrase query and a phrase prefix query in Elasticsearch?

1 point

- Phrase query matches exact phrases, while phrase prefix query matches partial phrases.
- Phrase query matches partial phrases, while phrase prefix query matches exact phrases.
- Phrase query matches documents with higher relevance scores, while phrase prefix query matches documents with lower relevance scores.
- Phrase query is used for filtering, while phrase prefix query is used for boosting.

6. What is the purpose of the relevance score in Elasticsearch?

1 point

- To filter out irrelevant documents from search results
- To sort documents in ascending order based on relevance
- To rank documents in descending order based on relevance
- To group documents into different categories based on relevance

7. Which Elasticsearch feature is used to search for a query string across multiple fields?

1 point

- Filters
- Caching
- Multi-match Querying
- Boosting Fields

8. Which Elasticsearch feature is used to assign higher importance or relevance to certain fields during query execution?

1 point

- Boosting Fields
- Filters
- Caching
- Multi-match Querying

9. What is the purpose of phrase query in Elasticsearch?

1 point

- To match exact phrases in documents
- To match partial phrases in documents
- To filter out irrelevant documents from search results
- To sort documents based on relevance

10. What is the purpose of phrase prefix query in Elasticsearch?

1 point

- To match partial phrases in documents
- To match exact phrases in documents
- To filter out irrelevant documents from search results
- To sort documents based on relevance

11. Which Elasticsearch feature is used to improve query performance by storing query results in memory for faster retrieval?

1 point

- Filters
- Multi-match Querying
- Boosting Fields
- Caching

12. What is the purpose of using filters in Elasticsearch?

1 point

- To narrow down search results based on specific criteria
- To sort documents based on relevance
- To store query results in memory for faster retrieval
- To assign higher importance or relevance to certain fields during query execution

13. Which Elasticsearch feature is used to search for a query string that matches multiple fields with different boosting factors?

1 point

- Filters
- Multi-match Querying
- Boosting Fields
- Caching

14. What is the purpose of using caching in Elasticsearch?

1 point

- To narrow down search results based on specific criteria
- To sort documents based on relevance
- To assign higher importance or relevance to certain fields during query execution
- To store query results in memory for faster retrieval

15. What is the purpose of boosting fields in Elasticsearch?

1 point

- To narrow down search results based on specific criteria
- To sort documents based on relevance
- To store query results in memory for faster retrieval
- To assign higher importance or relevance to certain fields during query execution

16. What is the main difference between phrase and phrase prefix queries in Elasticsearch?

1 point

- Phrase queries require an exact match of the query string, while phrase prefix queries match partial phrases.
- Phrase queries match partial phrases, while phrase prefix queries require an exact match of the query string.
- Both phrase queries and phrase prefix queries require an exact match of the query string.
- Both phrase queries and phrase prefix queries match partial phrases.

17. What is the relevance score in Elasticsearch?

1 point

- A score assigned to each document indicating its importance or relevance to the query
- A score assigned to each field indicating its importance or relevance to the query
- A score assigned to each query indicating its effectiveness in retrieving relevant documents
- A score assigned to each search result indicating its popularity or user engagement

Query String query

SYNTAX BASED

VERSATILE

USES OPERATORS

ALLOWS FOR COMPLEX SEARCHES

all_of,any_of

The screenshot shows a search interface with two code snippets side-by-side.

Left Snippet (all_of):

```
1 GET /product_catalog/_search
2 {
3   "size": 300,
4   "query": {
5     "query_string": {
6       "default_field": "description",
7       "query": "Sony AND (Camera OR Player)"
8     }
9   }
10 }
11
12 GET /product_catalog/_search
13 {
14   "query": {
15     "intervals": {
16       "description": {
17         "all_of": [
18           "intervals": [
19             {
20               "match": {
21                 "query": "Sony",
22                 "max_gaps": 1
23               }
24             }
25           ],
26           "any_of": [
27             "intervals": [
28               {
29                 "match": {
30                   "query": "Camera"
31                 }
32               }
33             ]
34           ]
35         }
36       }
37     }
38 }
```

Right Snippet (any_of):

```
1 {
2   "took": 2,
3   "timed_out": false,
4   "shards": [
5     {
6       "total": 1,
7       "successful": 1,
8       "failed": 0
9     }
10   ],
11   "hits": [
12     {
13       "total": {
14         "value": 50,
15         "relation": "eq"
16       },
17       "max_score": 0.3333333,
18       "hits": [
19         {
20           "index": "product_catalog",
21           "id": "2B0wv0cBuTHS1gqXZp4",
22           "score": 0.3333333,
23           "source": {
24             "brand": "Sony",
25             "model": "Sony DSCQ2 Soft Camera Carrying Case - LCDPK100",
26             "description": "Sony Soft Camera Carrying Case - LCSM100/ Stain Resistant Nylon/ Colored Interior/ Movable Partition/ Build-In Modular Interface/ Black Finish",
27             "id": "34150"
28           }
29         },
30         {
31           "index": "product_catalog",
32           "id": "2B0wv0cBuTHS1gqXZp4",
33           "score": 0.21686745,
34           "source": {
35             "brand": "Sony",
36             "model": "Sony DSCQ2 Soft Carrying Camera Case - Black Finish - LCSM100",
37             "description": "Sony DSCQ2 Soft Carrying Black Camera Case - LCSM1/ Polypropylene Construction/ Extra Compartments For Accessory Storage/ Can Hold Many Sony Or Other Manufacturers Digital Cameras/ Black Finish",
38             "id": "24921"
39           }
40         },
41         {
42           "index": "product_catalog",
43           "id": "8B0wv0cBuTHS1gqXZp4",
44           "score": 0.19999999,
45           "source": {
46             "brand": "Sony",
47             "model": "Sony 5 Disc CD Player - CDCE375",
48             "description": "Sony 5 Disc CD Player- CDCE375/ 5 Disc Changer/ Variable Line Output/ CD-R/RW Playback Capability/ 20 Track Music Calendar/ Digital Servo Control/ Remote Commander Remote Control",
49             "id": "5644"
50           }
51         },
52         {
53           "index": "product_catalog",
54           "id": "9B0wv0cBuTHS1gqXZp4",
55           "score": 0.19999999,
```

With max_gaps parameters

```

History Settings Variables Help
1 GET /product_catalog/_search
2 {
3   "size": 300,
4   "query": {
5     "query_string": {
6       "default_field": "description",
7       "query": "Sony AND (Camera OR Player)"
8     }
9   }
10 }
11
12 GET /product_catalog/_search
13 {
14   "query": {
15     "intervals": {
16       "description": {
17         "all_of": [
18           {"max_gaps": 3,
19            "intervals": [
20              {
21                "match": {
22                  "query": "Sony"
23                }
24              },
25              {
26                "any_of": {
27                  "intervals": [
28                    {
29                      "match": {
30                        "query": "Camera"
31                      }
32                    },
33                    {
34                      "match": {
35                        "query": "Player"
36                      }
37                    }
38                  ]
39                }
40              }
41            ]
42          }
43        }
44      }
45    }
46  }
47 }
48
49 
```

200 OK 339 ms

```

1 {
2   "took": 2,
3   "timed_out": false,
4   "_shards": {
5     "total": 1,
6     "successful": 1,
7     "skipped": 0,
8     "failed": 0
9   },
10 },
11 {
12   "hits": {
13     "total": 4,
14     "value": 4,
15     "relation": "eq",
16     "max_score": 0.33333333,
17     "hits": [
18       {
19         "_index": "product_catalog",
20         "_id": "ZB0wvocBuTH5Ig2XzP4",
21         "_score": 0.33333333,
22         "_source": {
23           "name": "Sony Soft Camera Carrying Case - LCSMX100",
24           "description": "Sony Soft Camera Carrying Case - LCSMX100/ Stain Resistant Nylon/ Colored Interior/ Movable Partition/ Build-in Modular Interface/ Black Finish",
25           "id": "34158"
26         }
27     },
28     {
29       "_index": "product_catalog",
30       "_id": "mB0wvocBuTH5Ig2XzI3",
31       "_score": 0.19999999,
32       "_source": {
33         "name": "Sony 5 Disc CD Player - CDPCE375",
34         "description": "Sony 5 Disc CD Player- CDPCE375/ 5 Disc Changer/ Variable Line Output/ CD-R/RW Playback Capability/ 20 Track Music Calendar/ Digital Servo Control/ Remote Commander Remote Control",
35         "id": "56447"
36       }
37     },
38     {
39       "_index": "product_catalog",
40       "_id": "mB0wvocBuTH5Ig2XzP4",
41       "_score": 0.19999999,
42       "_source": {
43         "name": "Sony DSLR-A350 Digital Camera And 18-70mm Zoom Lens - DSLRA350K",
44         "description": "Sony DSLR-A350 Digital Camera And 18-70mm Zoom Lens - DSLRA350K/ 14.2 Megapixel/ Tilttable 2.7 LCD Plus Screen/ Smart Teleconverter 2X Zoom/ Bionz Image Processor/ Super Steadyshot In-Camera Image Stabilization/ Anti-Dust Technology/ Eye-Start Autofocus System/ Auto Pop-up Flash/ Black Finish",
45         "id": "34824"
46       }
47     },
48     {
49       "_index": "product_catalog",
50       "_id": "ZB0wvocBuTH5Ig2XzI4",
51       "_score": 0.19999999,
52     }
53   }
54 }
```

Using query_string , string based search

```

History Settings Variables Help
1 GET /product_catalog/_search
2 {
3   "size": 300,
4   "query": {
5     "query_string": {
6       "default_field": "description",
7       "query": "(\\"Sony Camera\\")-3 OR (\\"Sony Player\\")-3"
8     }
9   }
10 }
11
12 GET /product_catalog/_search
13 {
14   "query": {
15     "intervals": {
16       "description": {
17         "all_of": [
18           {"max_gaps": 3,
19            "intervals": [
20              {
21                "match": {
22                  "query": "Sony"
23                }
24              },
25              {
26                "any_of": {
27                  "intervals": [
28                    {
29                      "match": {
30                        "query": "Camera"
31                      }
32                    },
33                    {
34                      "match": {
35                        "query": "Player"
36                      }
37                    }
38                  ]
39                }
40              }
41            ]
42          }
43        }
44      }
45    }
46  }
47 }
48
49 
```

200 OK 348 ms

```

1 {
2   "took": 2,
3   "timed_out": false,
4   "_shards": {
5     "total": 1,
6     "successful": 1,
7     "skipped": 0,
8     "failed": 0
9   },
10 },
11 {
12   "hits": {
13     "total": 4,
14     "value": 4,
15     "relation": "eq",
16     "max_score": 3.226815,
17     "hits": [
18       {
19         "_index": "product_catalog",
20         "_id": "ZB0wvocBuTH5Ig2XzP4",
21         "_score": 3.226815,
22         "_source": {
23           "name": "Sony Soft Camera Carrying Case - LCSMX100",
24           "description": "Sony Soft Camera Carrying Case - LCSMX100/ Stain Resistant Nylon/ Colored Interior/ Movable Partition/ Build-in Modular Interface/ Black Finish",
25           "id": "34158"
26         }
27     },
28     {
29       "_index": "product_catalog",
30       "_id": "mB0wvocBuTH5Ig2XzI3",
31       "_score": 1.3982547,
32       "_source": {
33         "name": "Sony 5 Disc CD Player - CDPCE375",
34         "description": "Sony 5 Disc CD Player- CDPCE375/ 5 Disc Changer/ Variable Line Output/ CD-R/RW Playback Capability/ 20 Track Music Calendar/ Digital Servo Control/ Remote Commander Remote Control",
35         "id": "56447"
36       }
37     },
38     {
39       "_index": "product_catalog",
40       "_id": "ZB0wvocBuTH5Ig2XzI4",
41       "_score": 1.3897247,
42       "_source": {
43         "name": "Sony Blu-Ray Disc Player - BDPSS50",
44         "description": "Sony Blu-Ray Disc Player- BDPSS50/ Decode Dolby TrueHD And dts HD Internally/ Full HD 1080 Video Output/ 24p True Cinema/ DVD Upscaling To 1080p Via HDMI/ BRAVIA Theatre Sync (HDMI-CEC)/ BD/DVD/CD Playback And AVOID With x.v. Color Output/ Dolby Digital Plus Decoding/ Backlit Remote/ Glass Front Finish",
45         "id": "37539"
46       }
47     },
48     {
49       "_index": "product_catalog",
50       "_id": "mB0wvocBuTH5Ig2XzP4",
51       "_score": 0.19999999,
52     }
53   }
54 }
```

Query string syntax

Query String Syntax

Wildcards →	qu?ck bro*
Regular Expressions →	name:/joh?n(ath[oa]n)/
Fuzziness →	quikc~ brwn~ foks~
Proximity Searches →	"fox quick"~5
Boosting →	quick^2 fox OR "john smith"^2 OR (foo bar)^4
Boolean Operators →	((quick AND fox) OR (brown AND fox))

Query string syntax ranges

Query String Syntax: Ranges

Range queries can be specified for date, numeric and string fields.

- **date:**[2012-01-01 TO 2012-12-31]
- **count:**[1 TO 5]
- **tag:**{alpha TO omega}
- **count:**[10 TO *]
- **date:**{* TO 2012-01-01}

Curly and square brackets can be combined:

- **count:**[1 TO 5]

```

History Settings Variables Help
1 GET /product_catalog/_search
2 {
3   "query": {
4     "query_string": {
5       "query": "description:black"
6     }
7   }
8 }

1+ {
2   "took": 2,
3   "timed_out": false,
4   "_shards": {
5     "total": 1,
6     "successful": 1,
7     "skipped": 0,
8     "failed": 0
9   },
10  "hits": {
11    "total": {
12      "value": 335,
13      "relation": "eq"
14    },
15    "max_score": 1.1422418,
16    "hits": [
17      {
18        "_index": "product_catalog",
19        "_id": "H0wocBuTHS1g2XzP4",
20        "_score": 1.1422418,
21        "_source": {
22          "name": "Canon Deluxe Soft Black Camera Case - 07160001",
23          "description": "Canon Deluxe Soft Black Camera Case - 07160001/ Compatible with Canon PowerShot S2 IS/ Black Finish",
24          "id": "36028"
25        }
26      },
27      {
28        "_index": "product_catalog",
29        "_id": "H0wocBuTHS1g2XzI4",
30        "_score": 1.1226186,
31        "_source": {
32          "name": "Nikon COOLPIX Leather Camera Case In Black - 5811",
33          "description": "Nikon COOLPIX Leather Camera Case In Black - 5811/ Compatible With Selected S Series COOLPIX Cameras/ Black Finish",
34          "id": "29932"
35        }
36      },
37      {
38        "_index": "product_catalog",
39        "_id": "ch0wocBuTHS1g2XzP4",
40        "_score": 1.1226186,
41        "_source": {
42          "name": "Sony Black Soft Carrying Case - LCSK30",
43          "description": "Sony Black Soft Carrying Case - LCSK30/ Stain Resistant Nylon Case/ Increased Interior Capacity/ Modular Interface/ Black Finish",
44          "id": "34282"
45        }
46      },
47      {
48        "_index": "product_catalog",
49        "_id": "ch0wocBuTHS1g2XzI4"
50      }
51    ]
52  }
53 }

200 - OK 363 ms

```

```

History Settings Variables Help
1 GET /product_catalog/_search
2 {
3   "query": {
4     "query_string": {
5       "query": "description:(black or music)"
6     }
7   }
8 }

1+ {
2   "took": 2,
3   "timed_out": false,
4   "_shards": {
5     "total": 1,
6     "successful": 1,
7     "skipped": 0,
8     "failed": 0
9   },
10  "hits": {
11    "total": {
12      "value": 367,
13      "relation": "eq"
14    },
15    "max_score": 6.007123,
16    "hits": [
17      {
18        "_index": "product_catalog",
19        "_id": "ER0wocBuTHS1g2Xz4",
20        "_score": 6.007123,
21        "_source": {
22          "name": "Home iPod & iPhone Bluetooth Alarm Clock Radio System - IP47",
23          "description": "Home iPod & iPhone Bluetooth Alarm Clock Radio System - IP47/ Play And Charge Your iPod Or iPhone/ Wirelessly Stream Music From Bluetooth Devices/ Auto-Set Clock/ Speakerphone With Built-In Microphone/ 4 Speaker Stereo/ Dual Alarm/ 12 AM/FM Station Memory Presets/ Full-Function Remote Included/ Compatible With Most iPods/ Black Finish (iPhone Not Included)",
24          "id": "36308"
25        }
26      },
27      {
28        "_index": "product_catalog",
29        "_id": "H0wocBuTHS1g2XzI4",
30        "_score": 5.5848183,
31        "_source": {
32          "name": "Sony Home Entertainment Server In Black - HESV1000",
33          "description": "Sony Home Entertainment Server In Black - HESV1000/ 200 BD/DVD/CD Changer With Full HD 1080P Upconversion/ 1000GB Hard Drive Storage To 130 Hours Of Video Or 20,000 Music Albums/ Or 20,000 Photos/ 7MB (CrossMediaBar) Memory/ BRAVIA Theatre Sync (HDMI-CEC)/ AVC-HD With x.v.Color Output/ Dolby Digital And dts 5.1 Channel Output/ x-Pict Story HD And x-Sceabook Photo Slide Show Applications/ Remote Control/ Black Finish",
34          "id": "36006"
35        }
36      },
37      {
38        "_index": "product_catalog",
39        "_id": "ch0wocBuTHS1g2XzI4",
40        "_score": 4.7224116,
41        "_source": {
42          "name": "Sony Bluetooth Adaptor/Portable Transmitter - THBBT10",
43          "description": "Sony Bluetooth Adaptor/Portable Transmitter - THBBT10/ Listen Up To 11 Hours Of Music/ Transmit Audio Wirelessly From Digital Music Player To Compatible Devices/ Bluetooth Technology/ Gold Plated Plug/ Rechargeable",
44          "id": "32022"
45        }
46      }
47    ]
48  }
49 }

200 - OK 364 ms

```

```

History Settings Variables Help
1 GET /product_catalog/_search
2 {
3   "query": {
4     "query_string": {
5       "query": "description:('Black Finish')"
6     }
7   }
8 }

1 "took": 2,
2 "timed_out": false,
3 "_shards": [
4   {"total": 1,
5   "successful": 1,
6   "skipped": 0,
7   "failed": 0
8 },
9 ],
10 "hits": {
11   "total": {
12     "value": 305,
13     "relation": "eq"
14   },
15   "max_score": 1.3498285,
16   "hits": [
17     {
18       "_index": "product_catalog",
19       "_id": "5B0vovBu17HSig2XZp4",
20       "_score": 1.3498285,
21       "_source": {
22         "name": "Peerless Flat Panel Display Mount Black Finish - SGLB01",
23         "description": "Peerless Flat Panel Display Mount Black Finish - SGLB01/ Can Hold Between 32' To 50' Screens /Vertical Screen Adjustment Feature for Optimal Viewing Height/ Internal Cable Management/ Adjustable Glass Shelving/ Black Finish",
24         "id": "32984"
25     },
26   },
27   {
28     "_index": "product_catalog",
29     "_id": "h0vovBu17HSig2XZp4",
30     "_score": 1.2721126,
31     "_source": {
32       "name": "Canon Battery Charger - 11338001",
33       "description": "Canon Battery Charger - 11338001/ Compatible With Canon NB-SL Lithium Battery/ Black Finish",
34       "id": "36027"
35     }
36   },
37   {
38     "_index": "product_catalog",
39     "_id": "4h0vovBu17HSig2XZp4",
40     "_score": 1.2339892,
41     "_source": {
42       "name": "Garmin 010-10723-02 Carrying Case - 0101072302",
43       "description": "Garmin 010-10723-02 Carrying Case - 0101072302/ Compatible With Garmin Nuvi 350/ Black Finish",
44       "id": "32875"
45     }
46   },
47   {
48     "_index": "product_catalog",
49     "_id": "h0vovBu17HSig2XZp4"
50   }
51 }

1 GET /fox-index/_search
2 {
3   "query": {
4     "query_string": {
5       "query": "\'quick fox\'~-5"
6     }
7   }
8 }

10 GET /product_catalog
11
12
13
14
15
16
17

1 "took": 1,
2 "timed_out": false,
3 "_shards": [
4   {"total": 1,
5   "successful": 1,
6   "skipped": 0,
7   "failed": 0
8 },
9 ],
10 "hits": {
11   "total": {
12     "value": 2,
13     "relation": "eq"
14   },
15   "max_score": 1.2039728,
16   "hits": [
17     {
18       "_index": "fox-index",
19       "_id": "9Kob3lCB519qUVzrBSR",
20       "_score": 1.2039728,
21       "_source": {
22         "title": "quick fox"
23       }
24     },
25     {
26       "_index": "fox-index",
27       "_id": "86obb3lCB519qUVzJRSy",
28       "_score": 0.4604298,
29       "_source": {
30         "title": "the quick brown fox"
31       }
32     }
33   ]
34 }

1 GET /fox-index/_search
2 {
3   "query": {
4     "query_string": {
5       "query": "\'quick fox\'~-5"
6     }
7   }
8 }

10 GET /product_catalog/_search
11 {
12   "query": {
13     "query_string": {
14       "query": "price:[399 TO 1099]"
15     }
16   }
17 }

1 "took": 1,
2 "timed_out": false,
3 "_shards": [
4   {"total": 1,
5   "successful": 1,
6   "skipped": 0,
7   "failed": 0
8 },
9 ],
10 "hits": {
11   "total": {
12     "value": 2,
13     "relation": "eq"
14   },
15   "max_score": 1.2039728,
16   "hits": [
17     {
18       "_index": "fox-index",
19       "_id": "9Kob3lCB519qUVzrBSR",
20       "_score": 1.2039728,
21       "_source": {
22         "title": "quick fox"
23       }
24     },
25     {
26       "_index": "fox-index",
27       "_id": "86obb3lCB519qUVzJRSy",
28       "_score": 0.4604298,
29       "_source": {
30         "title": "the quick brown fox"
31       }
32     }
33   ]
34 }

```

[Learn more about String Based Search](#)

Introduction to String Based Search

The search API allows you to execute a search query and get back search hits that match the query. The query can either be provided using a simple query string as a parameter, or using a request body.

[Read more](#)

Custom Attributes for Query Based Search

The query string is parsed into a series of terms and operators. A term can be a single word — quick or brown — or a phrase, surrounded by double quotes — "quick brown" — which searches for all the words in the phrase, in the same order.

Operators allow you to customize the search.

[Read more](#)

Implementation of Query String

You can use the query_string query to create a complex search that includes wildcard characters, searches across multiple fields, and more. While versatile, the query is strict and returns an error if the query string includes any invalid syntax.

[Read more](#)

1. What is the purpose of custom attributes in query-based search?

1 / 1 point

- To specify additional parameters for sorting the search results.
- To define custom filters for refining the search results.
- To provide custom scoring rules for ranking the search results.
- To specify custom language settings for tokenizing the search query.

 **Correct**

Custom attributes can be used to specify custom language settings for tokenizing the search query, allowing for better query parsing and tokenization.

2. How is the implementation of a query string query done in Elasticsearch?

1 / 1 point

- By using the "_query_string" field in the query DSL.
- By using the "query_string" query type in the query DSL.
- By using the "query_string" parameter in the search API.
- By using the "query_string" field in the index mapping.

 **Correct**

The "query_string" parameter is used in the search API to implement a query string query in Elasticsearch.

3. What are the properties of a common terms query in Elasticsearch?

1 / 1 point

- It searches for terms with a minimum match ratio.
- It includes both stop words and non-stop words in the search.
- It boosts terms with low term frequency to improve search relevance.
- It allows specifying a custom list of stop words to be excluded from the search.

 **Correct**

Common terms query allows specifying a custom list of stop words to be excluded from the search, improving search relevance.

4. What is the purpose of the "phrase" vs "phrase_prefix" option in multi-match querying?

1 / 1 point

- To specify the maximum number of allowed phrase prefixes in the search results.
- To define the type of phrase matching to be used in the multi-match query.
- To set the minimum length of phrases to be considered in the search results.
- To specify the maximum number of allowed phrase matches in the search results.

 **Correct**

The "phrase" vs "phrase_prefix" option in multi-match querying is used to define the type of phrase matching to be used, allowing for more precise search results.

5. What is the purpose of the "explain" API in Elasticsearch for query optimization?

1 / 1 point

- To provide a detailed explanation of how scoring is calculated for search results.
- To analyze the performance of the search query and identify performance bottlenecks.
- To optimize search result relevance by adjusting scoring factors.
- To enable caching of search results for faster query performance.

 **Correct**

The "explain" API in Elasticsearch is used to analyze the performance of the search query and identify performance bottlenecks, helping in query optimization.

Join Field Type

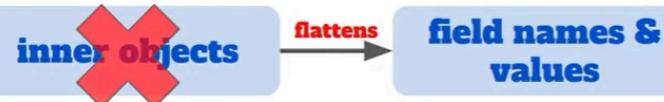
creates parent/child relation within documents of the same index



BOARD

Nested Field Type

allows arrays of objects to be indexed in a way that they can be queried independently of each other.



maintains the independence of each object in the array

each object is indexed in the array as a separate hidden document

can be queried independently of the others with the nested query

☰ D Dev Tools Console

Console Search Profiler Grok Debugger Painless Lab BETA

History Settings Variables Help

```

1 PUT my-index-000001/_doc/1
2 {
3   "group": "fans",
4   "user": [
5     {
6       "first": "John",
7       "last": "Smith"
8     },
9     {
10       "first": "Alice",
11       "last": "White"
12     }
13   ]
14 }
```

▶ ⚙️

```

1 {
2   "acknowledged": true,
3   "shards_acknowledged": true,
4   "index": "my-index-000001"
5 }
```

Console Search Profiler Grok Debugger Painless Lab BETA

History Settings Variables Help

```

1 GET my-index-000001/_search
2 {
3   "query": {
4     "nested": {
5       "path": "user",
6       "query": {
7         "bool": {
8           "must": [
9             {
10               "match": { "user.first": "Alice" }
11             },
12             {
13               "match": { "user.last": "Smith" }
14             }
15           ]
16         }
17       }
18     }
19   }
20 }
```

▶ ⚙️

```

1 {
2   "took": 3,
3   "timed_out": false,
4   "_shards": {
5     "total": 1,
6     "successful": 1,
7     "skipped": 0,
8     "failed": 0
9   },
10  "hits": {
11    "total": {
12      "value": 0,
13      "relation": "eq"
14    },
15    "max_score": null,
16    "hits": []
17  }
18 }
```

History Settings Variables Help

```

1 GET my-index-000001/_search
2 {
3   "query": {
4     "nested": {
5       "path": "user",
6       "query": {
7         "bool": {
8           "must": [
9             {
10               "match": {
11                 "user.first": "Alice"
12               }
13             },
14             {
15               "match": {
16                 "user.last": "White"
17               }
18             }
19           ]
20         }
21       }
22     }
23 }
```

▶ ⚙️

```

1 {
2   "took": 1,
3   "timed_out": false,
4   "_shards": {
5     "total": 1,
6     "successful": 1,
7     "skipped": 0,
8     "failed": 0
9   },
10  "hits": {
11    "total": {
12      "value": 1,
13      "relation": "eq"
14    },
15    "max_score": 1.3862942,
16    "hits": [
17      {
18        "_index": "my-index-000001",
19        "_id": "1",
20        "_score": 1.3862942,
21        "_source": {
22          "group": "fans",
23          "user": [
24            {
25              "first": "John",
26              "last": "Smith"
27            },
28            {
29              "first": "Alice",
30              "last": "White"
31            }
32          ]
33        }
34      }
35    ]
36  }
37 }
```

Further Reading on Implementation of Advanced Techniques

Nested Queries in Elasticsearch: Searching Inside Nested Objects

The nested query searches nested field objects as if they were indexed as separate documents. If an object matches the search, the nested query returns the root parent document.

[Read more](#)

Script Queries in Elasticsearch: Customizing Query Behavior with Scripts

Filters documents based on a provided script. The script query is typically used in a filter context.

[Read more](#)

Join Queries in Elasticsearch: Searching Across Related Documents

Performing full SQL-style joins in a distributed system like Elasticsearch is prohibitively expensive. Instead, Elasticsearch offers two forms of join which are designed to scale horizontally.

[Read more](#)

1. What is the purpose of using Nested Queries in Elasticsearch?

1 / 1 point

- To search for documents within nested objects.
- To customize the query behavior with scripts.
- To search across related documents.
- To optimize query performance by caching search results.

 **Correct**

Yes, Nested Queries are used to search for documents within nested objects in Elasticsearch.

2. What is the purpose of using Script Queries in Elasticsearch?

1 / 1 point

- To search for documents within nested objects.
- To customize the query behavior with scripts.
- To search across related documents.
- To optimize query performance by caching search results.

 **Correct**

Yes, Script Queries are used to customize the query behavior with scripts in Elasticsearch.

3. What is the purpose of using Join Queries in Elasticsearch?

1/1 point

- To search for documents within nested objects.
- To customize the query behavior with scripts.
- To search across related documents.
- To optimize query performance by caching search results.



Correct

Yes, Join Queries are used to search across related documents in Elasticsearch.

4. What is the purpose of using Script Queries in Elasticsearch?

1/1 point

- To customize the query behavior with scripts.
- To search for documents within nested objects.
- To search across related documents.
- To optimize query performance by caching search results.



Correct

Yes, Script Queries are used to customize the query behavior with scripts in Elasticsearch.

Week -3 Graded Quiz

1. What is the primary purpose of a string-based search?

1 point

- To search for specific data types in a database.
- To search for files with specific file extensions.
- To search for web pages based on their URLs.
- To search for specific strings in a text document.

2. What is the difference between a query-based search and a string-based search?

1 point

- Query-based searches are more accurate than string-based searches.
- String-based searches are faster than query-based searches.
- Query-based searches use custom attributes to refine the search results.

3. What is the purpose of custom attributes in a query-based search?

1 point

- To define the type of input field.
- To provide a unique identifier for the input field.
- To refine the search results based on specific criteria.

4. What is a nested query in Elasticsearch?

1 point

- A query that searches for related documents in multiple indexes.
- A query that searches for documents within a specified range of values.
- A query that searches for documents based on a specific field or value.
- A query that searches for documents within nested objects.

5. What is a script query in Elasticsearch?

1 point

- A query that searches for documents with a specific script tag.
- A query that searches for documents based on a specific scripting language.
- A query that customizes the behavior of a query using a script.

6. What is a join query in Elasticsearch?

1 point

- A query that searches for documents based on a specific field or value.
- A query that searches for documents within a specified range of values.
- A query that searches for related documents in multiple indexes.
- A query that searches across related documents within the same index.

7. What is the purpose of a query string in Elasticsearch?

1 point

- To define the type of input field for a search query.
- To provide a unique identifier for a search query.
- To search for documents based on a specified string of text.

8. What is a nested object in Elasticsearch?

1 point

- An object that is stored within a specific field in a document.
- An object that is shared across multiple documents in an index.
- An object that is embedded within another object in a document.

9. What is a filter in Elasticsearch?

1 point

- A query that searches for documents with a specific filter tag.
- A query that searches for documents based on a specific field or value.
- A query that narrows down the search results based on specific criteria.

10. What is the purpose of a boosting query in Elasticsearch?

1 point

- To reduce the relevance of certain documents in the search results.
- To increase the relevance of certain documents in the search results.
- To randomly select certain documents to include in the search results.
- To sort the search results in ascending or descending order.

11. What is a common use case for a custom attribute in a query-based search?

1 point

- To specify the type of input field for a search query.
- To provide a unique identifier for a search query.
- To filter or sort search results based on a specific attribute of a document.
- To search for documents across multiple indexes.

12. What is the purpose of a nested aggregation in Elasticsearch?

1 point

- To combine the results of multiple search queries into a single result set.
- To search for documents within a specified range of values.
- To customize the behavior of a query using a script.
- To summarize data within nested objects in a document.

13. What is the purpose of the "type" attribute in an HTML form input element?

1 point

- To specify the label for the input field.
- To define the type of input field.
- To provide a unique identifier for the input field.
- To set a default value for the input field.

14. What is the purpose of a nested object in Elasticsearch?

1 point

- To customize the behavior of a query using a script.
- To filter or sort search results based on a specific attribute of a document.
- To search for documents within nested objects in a document.
- To organize and structure complex data within a document.

15. What is the purpose of a custom score query in Elasticsearch?

1 point

- To combine the results of multiple search queries into a single result set.
- To filter or sort search results based on a specific attribute of a document.
- To search for documents within nested objects in a document.
- To customize the relevance score of search results based on specific criteria;

16. What is the purpose of a function score query in Elasticsearch?

1 point

- To combine the results of multiple search queries into a single result set.
- To filter or narrow down search results based on specific criteria.
- To search for documents based on a specific query string.
- To customize the relevance score of search results based on specific criteria;