Elasticsearch Hands-on Session

Given below are the sample queries used in the Elasticsearch Hands-on session. These queries can be directly executed in the DevTools console in Kibana

1. Sample Search Queries

<table>
<tr>
<td><b>Bool Query</b></td>
<td>

```
GET /amazon-reviews/_search
{
 "query": {
   "bool" : {
     "must" : {
       "match" : { "review_body" : { "query" : "customer support" }}
     },
     "filter": {
       "term" : { "language" : "sin" }
     },
     "must_not" : {
       "range" : {
         "stars" : { "gte" : 7 }
       }
     },
     "should" : [
       { "term" : { "product_category" : "wireless" } }
     ],
     "minimum_should_match" : 1,
     "boost" : 1.0
   }
 }
}
```

</td>
</tr>
<tr>
<td><b>Match Query</b></td>
<td>

```
GET /amazon-reviews/_search
{
 "query": {
   "match": {
     "review_body.case_insensitive_and_inflections": {
       "query": "poor quality",
     }
   }
 }
}
```

</td>
</tr>
</table>

Match Phrase Query

```
GET /amazon-reviews/_search
{
 "query": {
   "match_phrase": {
     "review_body.case_insensitive_and_inflections": "manufacturing defect"
   }
 }
}
```

Match Boolean Prefix Query

```
GET /amazon-reviews/_search
{
 "query": {
   "match_bool_prefix" : {
     "review_body" : "missing struct"
   }
 }
}
```

Multi Match Query

```
GET /_search
{
 "query": {
   "multi_match" : {
     "query":      "order not received",
     "type":       "most_fields",
     "fields":     [ "review_body", "review_title"]
   }
 }
}
```

Custom Script Query

```
GET /amazon-reviews/_search
{
 "query": {
   "bool": {
     "filter": {
       "script": {
         "script": """
           double rating = doc['stars'].value;
           if (doc['product_category'].value == 'wireless') {
             rating *= -1;
           }
           return rating > 5;
         """
       }
     }
   }
 }
}
```

Range Query	GET /amazon-reviews/_search
	{
	 "query": {
	  "range": {
	   "stars": {
	    "gte": 4,
	    "lte": 6
	   }
	  }
	 }
	}

	GET /amazon-reviews/_search
	{
	 "query": {
	  "range": {
	   "published_date": {
	    "gte": "01/01/2022",
	    "lt": "04/04/2022"
	   }
	  }
	 }
	}

Term Query	GET /amazon-reviews/_search
	{
	 "query": {
	  "term": {
	   "product_category": {
	    "value": "kitchen"
	   }
	  }
	 }
	}

Wildcard Query	GET /_search
	{
	 "query": {
	  "wildcard": {
	   "product_category": {
	    "value": "*home*"
	    }
	  }
	 }
	}

Interval Query

```
GET /amazon-reviews/_search
{
 "_source": ["review_body"],
 "query": {
  "bool": {
   "must": {
    "intervals" : {
     "review_body.case_insensitive_and_inflections" : {
      "all_of" : {
       "ordered" : false,
       "max_gaps": 10,
       "intervals" : [
        {
         "match" : {
          "query" : "Manufacturer defect",
          "ordered" : true,
          "max_gaps": 0
         }
        },
        {
         "match" : {
          "query" : "broken"
         }
        }
       ]
      }
     }
    }
   }
  }
 },
 "highlight": {
  "fields": {
   "review_body.case_insensitive_and_inflections": {}
  },
  "pre_tags": "<em>",
  "post_tags": "</em>",
  "fragment_size": 200
 }
}
```

2. Sample Aggregation Queries

<table>
<tr>
<td><b>Date Range Aggregation</b></td>
<td>

```
GET /amazon-reviews/_search
{
  "aggs": {
    "rating_ranges": {
      "range": {
        "field": "stars",
        "ranges": [
          { "to": 4 },
          { "from": 4, "to": 8 },
          { "from": 8 }
        ]
      }
    }
  }
}
```

</td>
</tr>
<tr>
<td><b>Terms Aggregation</b></td>
<td>

```
GET /amazon-reviews/_search
{
  "aggs": {
    "product": {
      "terms": { "field": "product_category" }
    }
  }
}
```

</td>
</tr>
<tr>
<td><b>Metrics Aggregation</b></td>
<td>

```
POST /amazon-reviews/_search
{
  "aggs": {
    "rating_stats": { "stats": { "field": "stars" } }
  }
}
```

</td>
</tr>
</table>

3. Other major functionalities

Paginating search results

```
GET /amazon-reviews/_search
{
  "from": 0,
  "size" : 2,
  "query": {
     "multi_match" : {
       "query":     "money",
       "type":      "most_fields",
       "fields":    [ "review_body.case_insensitive_and_inflections",
"review_title"]
    }
  }
}
```

Highlighting search results

```
GET /amazon-reviews/_search
{
  "_source": {
          "includes": [ "review_body", "review_title" ]
       },

  "query": {
    "multi_match" : {
      "query":     "order not received",
      "type":      "most_fields",
      "fields":    ["review_body.case_insensitive","review_title"]
    }
  },
  "highlight": {
    "fields": {
      "review_body.case_insensitive": {}
    },
    "pre_tags": "<em>",
    "post_tags": "</em>",
    "fragment_size": 200
  }
}
```

Search templates and it’s Execution

```
PUT _scripts/search-mode-1
{
  "script": {
    "lang": "mustache",
    "source": {
      "query": {
        "intervals" : {
          "review_body.case_insensitive_and_inflections" : {
          "all_of" : {
            "ordered" : "{{order}}",
```

	"max_gaps": "{{distance}}",
	       "intervals" : [
	        {
	         "match" : {
	          "query" : "{{query_string_1}}",
	          "ordered" : true,
	          "max_gaps": 0
	         }
	        },
	        {
	         "match" : {
	          "query" : "{{query_string_2}}"
	         }
	        }
	       ]
	      }
	     }
	    }
	   }
	  }
	 }
	}

	GET /amazon-reviews/_search/template
	{
	 "id": "search-mode-1",
	 "params": {
	  "query_string_1": "Manufacturer defect",
	  "query_string_2": "broken",
	  "order": false,
	  "distance": 10
	 }
	}

Calculating query document result score	GET /amazon-reviews/_search?explain=true
	{
	 "query": {
	  "multi_match" : {
	   "query": "order not received",
	   "type": "most_fields",
	   "fields": ["review_body", "review_title"]
	  }
	 }
	}

Insight on how the query is executed

```
GET /amazon-reviews/_search
{
  "profile": "true",
  "query": {
    "multi_match" : {
      "query":      "order not received",
      "type":       "most_fields",
      "fields":     [ "review_body", "review_title"]
    }
  }
}
```