

# Database Project (SWE3033) (Fall 2023)

## Homework #7 (50pts, Due date: 11/15)

Student ID: 2020315798

Student Name: Choi Jin Woo

**Instruction:** We provided datasets and Jupyter notebook file. Data is data about film reviews and is provided in the form of JSON file. Please write your code to get correct result. **If you edit the "do not edit here" part, you will not receive any points.**

Submit two files as follows:

- 'DBP\_Homework7\_STUDENTID.zip'
  - Code: DBP\_Homework7\_STUDENTID.ipynb
  - Document: DBP\_Homework7\_STUDENTID.pdf

1. **[20pts]** Using the given **movie\_data.json** file, write the code of index mapping, write and execute the code to create index and documents to elastic search, and report the image that include the number of hits and the data view visualized through Kibana.

**Answer:** Enter your **code and result** here. You must show your result, either an image or text.  
**Hints:** Write your index mapping to work with the queries below.

**[5pts] [Index mapping]**

▼ **[5pts]** Write mapping for the index

```
[6]: INDEX_NAME = "movie_review"
##### EDIT HERE #####
mapping = {
  "movieId": {
    "type": "integer" # Fill in the blank
  },
  "title": {
    "type": "text"
  },
  "genres": {
    "type": "text"
  },
  "imdbId": {
    "type": "integer"
  },
  "tmdbId": {
    "type": "integer"
  },
  "userId": {
    "type": "integer"
  },
  "rating": {
    "type": "float"
  },
  "timestamp": {
    "type": "date"
  }
}
##### EDIT HERE #####
```

**[10pts] [create index & documents]**

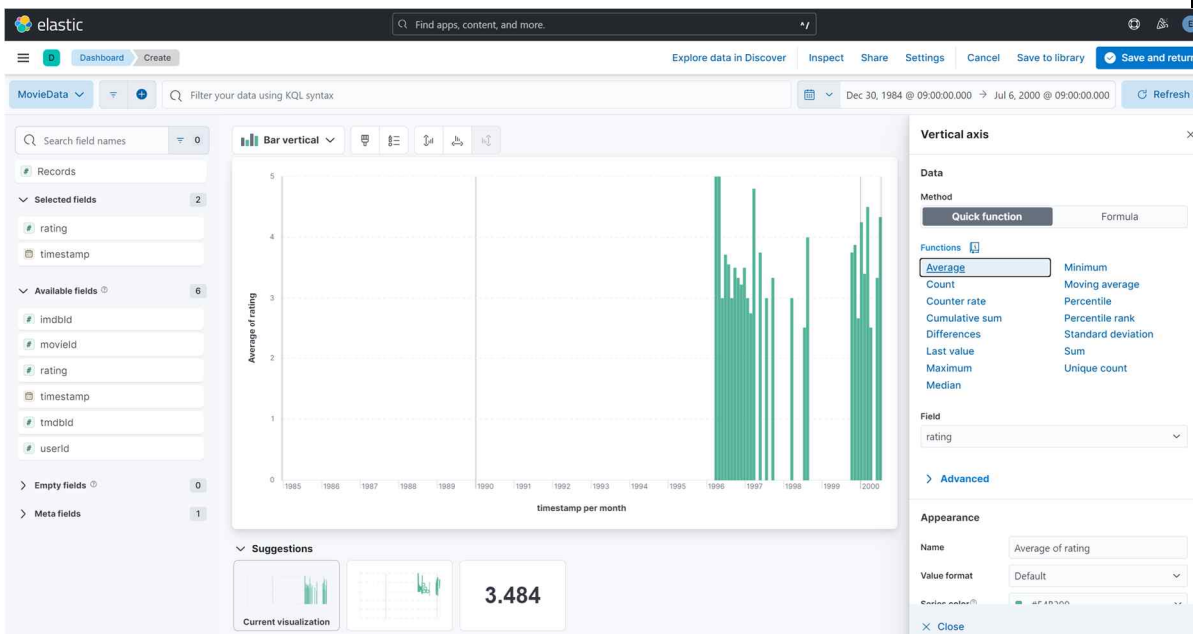
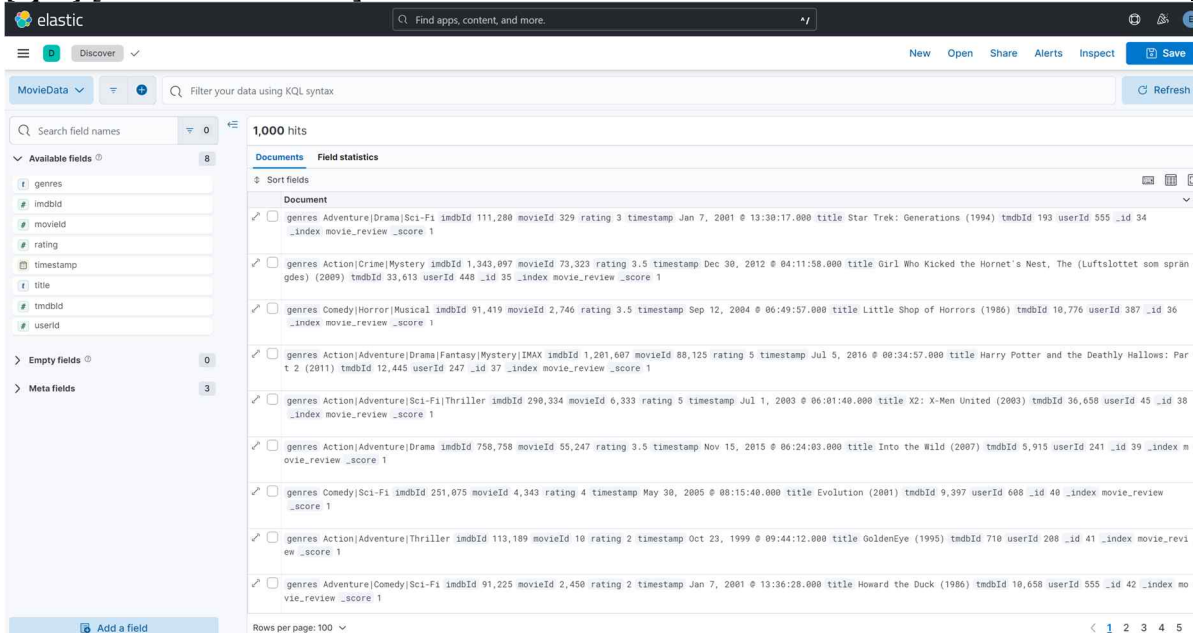
### [5pts] Create Index

```
[13]: if not client.indices.exists(index=INDEX_NAME):
      ##### EDIT HERE #####
      client.indices.create(index=INDEX_NAME, body={"mappings":{"properties":mapping}})
      ##### EDIT HERE #####
```

### [5pts] Insert documents

```
[14]: with open('./movie_data.json', 'r') as f:
      movie_json = json.load(f)
      ##### EDIT HERE #####
      for id_doc, document in enumerate(movie_json):
          client.index(index=INDEX_NAME, id=id_doc, body=document)
      ##### EDIT HERE #####
```

### [5pts] [elastic search result]



2. [30pts] Write a query code for the given questions and report the results.

Answer: Enter your **code and result** here. You must show your result, either an image or text.

- A. [10pts] What is the total **number of reviews** for films that contain the **both Action and Crime** genres?

[query code]

```
query = {
  "size": 1000,
  "query": {
    "bool": {
      "filter": [
        {"match": {"genres": "Action"}},
        {"match": {"genres": "Crime"}}
      ]
    }
  }
}
```

[result]

- [10pts] What is the total number of reviews for the films that contain the both "Action" and "Crime" genres?

```
[30]: query = {
      ##### EDIT HERE #####
      "size": 1000,
      "query": {
        "bool": {
          "filter": [
            {"match": {"genres": "Action"}},
            {"match": {"genres": "Crime"}}
          ]
        }
      }
      ##### EDIT HERE #####
    }

    ##### DO NOT EDIT HERE #####
    response = client.search(index=INDEX_NAME, body=query)
    print(len(response['hits']['hits']))
    ##### DO NOT EDIT HERE #####

77
```

The screenshot shows the Elasticsearch Kibana interface. The search bar at the top contains the query: `genres: Action & genres: Crime`. The results page shows 77 hits. The left sidebar displays the 'Available fields' section with fields like `genres`, `imdbid`, `movied`, `rating`, `timestamp`, `title`, `tmbid`, and `userid`. The main content area shows a list of documents with columns for `Document` and `Field statistics`. The first document is for the movie 'One for the Money' (2012) with a rating of 1. The second document is for 'Looper' (2012) with a rating of 3.5. The third document is for 'Sin City' (2005) with a rating of 4. The fourth document is for 'Dark Knight, The' (2008) with a rating of 4. The fifth document is for 'RoboCop 2' (1990) with a rating of 3. The sixth document is for 'Punisher, The' (2004) with a rating of 5. The seventh document is for 'Léon: The Professional' (1994) with a rating of 3. The eighth document is for 'Net, The' (1995) with a rating of 4. The bottom of the page shows 'Rows per page: 100'.

- B. [10pts] What is the total number of **reviews** for films that contain the **Sci-fi** genre and have a **rating 4.0 or more but 5.0 or less** and report the list of the movie.

[query code]

```
query = {  
  "size": 1000,  
  "query": {  
    "bool": {  
      "must": [  
        {"match": {"genres": "Sci-Fi"}},  
        {"range": {"rating": {"gte": 4.0, "lte": 5.0}}}]  
      }  
    }  
  }  
}
```

[result]

**[10pts]** What is the total number of reviews for the films that contain the "Sci-Fi" genres and have a rating 4.0 or more but 5.0 or less and report the list of the films.

```
[35]: query = {
    ##### EDIT HERE #####
    "size": 1000,
    "query": {
        "bool": {
            "must": [
                {"match": {"genres": "Sci-Fi"}},
                {"range": {"rating": {"gte": 4.0, "lte": 5.0}}}
            ]
        }
    }
    ##### EDIT HERE #####
}

##### DO NOT EDIT HERE #####
response = client.search(index=INDEX_NAME, body=query)
print("Numbers of result:", response['hits']['total']['value'])

movie_set = set()
for res in response['hits']['hits']:
    movie_set.add(res['_source']['title'])
for movie in movie_set:
    print(movie)
##### DO NOT EDIT HERE #####
```

```
Numbers of result: 72
Robots (2005)
Truman Show, The (1998)
Doctor Strange (2016)
Unbreakable (2000)
V for Vendetta (2006)
Star Wars: Episode V - The Empire Strikes Back (1980)
Pandorum (2009)
Ghost in the Shell 2.0 (2008)
Aliens (1986)
Island, The (2005)
Interstellar (2014)
2010: The Year We Make Contact (1984)
X-Men (2000)
Blob, The (1958)
Hollow Man (2000)
Piranha (1978)
Stepford Wives, The (1975)
The Hunger Games: Mockingjay - Part 1 (2014)
Star Trek: First Contact (1996)
Altered States (1980)
Avatar (2009)
RoboCop (1987)
Blood of Heroes, The (Salute of the Jugger, The) (1989)
Matrix Revolutions, The (2003)
Final Fantasy: The Spirits Within (2001)
Inception (2010)
Battlestar Galactica (2003)
Flash Gordon (1980)
Bill & Ted's Excellent Adventure (1989)
Cube (1997)
Superman Returns (2006)
Splice (2009)
Dune (1984)
Ex Machina (2015)
The Hunger Games (2012)
Blade Runner (1982)
Back to the Future (1985)
World's End, The (2013)
WALL-E (2008)
Thor: Ragnarok (2017)
Spider-Man (2002)
Demolition Man (1993)
Mad Max: Fury Road (2015)
```

```
Brazil (1985)
Iron Man 2 (2010)
Iron Giant, The (1999)
Star Wars: Episode VI - Return of the Jedi (1983)
K-PAX (2001)
Matrix Reloaded, The (2003)
X2: X-Men United (2003)
Total Recall (1990)
Outbreak (1995)
I Am Legend (2007)
Star Trek VI: The Undiscovered Country (1991)
Dune (2000)
Evolution (2001)
Back to the Future Part II (1989)
Twelve Monkeys (a.k.a. 12 Monkeys) (1995)
```

The screenshot shows the Elasticsearch Kibana interface. At the top, there's a search bar with the query: `genres: Sci-Fi & ("range":{"rating":{"gte":"4.0","lte":"5.0"}})`. Below the search bar, there's a sidebar with field statistics and a list of fields. The main area displays 72 hits, with a list of movie documents on the right and field statistics on the left. The documents are sorted by rating in ascending order.

C. [10pts] Find the 10 lowest rated reviews between 2005 and 2010, in order.

```
[query code]
query = {
  "size": 10,
  "query": {
    "bool": {
      "filter": [
        {"range": {"timestamp": {"gte": "2005-01-01", "lte": "2010-12-31"}}}
      ]
    }
  },
  "sort": [
    {"rating": {"order": "asc"}}
  ]
}
```

[result]

```
[39]: query = {
    ##### EDIT HERE #####
    "size": 10,
    "query": {
        "bool": {
            "filter": [
                {"range": {"timestamp": {"gte": "2005-01-01", "lte": "2010-12-31"}}}
            ]
        }
    },
    "sort": [
        {"rating": {"order": "asc"}}
    ]
}

##### DO NOT EDIT HERE #####
response = client.search(index=INDEX_NAME, body=query)
for hit in response['hits']['hits']:
    print(hit['_source']['title'], hit['_source']['rating'])
##### DO NOT EDIT HERE #####
```

```
Raising Cain (1992) 0.5
Titanic (1997) 0.5
Client, The (1994) 0.5
Darkness (2002) 0.5
Sex, Lies, and Videotape (1989) 0.5
Killer Shrews, The (1959) 0.5
Alien: Resurrection (1997) 1.0
Shrek (2001) 1.0
Pee-wee's Big Adventure (1985) 1.0
Legally Blonde (2001) 1.0
```

The screenshot shows the Elasticsearch Kibana interface. At the top, there's a search bar with 'Find apps, content, and more.' Below it, the 'Discover' tab is active, showing a table of search results. The table has two columns: 'rating' and 'title'. The results are sorted by 'rating' in ascending order. The first few rows are:

rating	title
0.5	Raising Cain (1992)
0.5	Titanic (1997)
0.5	Client, The (1994)
0.5	Darkness (2002)
0.5	Sex, Lies, and Videotape (1989)
0.5	Killer Shrews, The (1959)
1.0	Alien: Resurrection (1997)
1.0	Shrek (2001)

On the left side, there's a sidebar with 'Selected fields' (rating, title) and 'Available fields' (genres, imdbid, movied, rating, timestamp, title, imdbid, userid). At the bottom, there's a 'Add a field' button.