**Database Project (SWE3033) (Fall 2023)**

**Homework #8 (50pts, Due date: 11/22)**

**Student ID**: 2020315798

**Student Name**: Choi Jin Woo

**Instruction:** The objective of this homework is to construct a Python Elasticsearch client class than can be used to interact with an Elasticsearch cluster. Implement ‘**ElasticsearchClient’** class in ‘elasticsearch\_client.py’ and use the correct function for the annotation in ‘main.ipynb’.

We have provided the datasets and all the Python files you need. The dataset is about movie reviews and is provided in the form of JSON file. Please write your code to get the correct result. **If you edit anything other than EDIT HERE, you may be penalized.**

Submit two files as follows:

* ‘DBP\_Homework8\_STUDENTID.zip’
* Code:
  + main.ipynb
  + elasticsearch\_client.py
* Document: DBP\_Homework8\_STUDENTID.pdf

1. **[40pts]** Implement the various functions for common Elasticsearch operations at **‘elasticsearch\_client.py’** and report the result at **‘main.ipynb’**.
   1. **[30pts]** Implement the following functions.  
        
      Answer: Enter your **codes** here.

|  |
| --- |
| [def create\_index] |
| [def insert\_one\_document] |
| [def get\_document] |
| [def update\_document\_by\_id] |
| [def delete\_index] |
| [def delete\_document] |
| [def search] |
| [def count] |
| [def scan\_index] |
| [def bulk\_request] |

* 1. **[10pts]** Use the correct function to get the correct result. Enter your **codes** and **results** here. You must take a screenshot of the result. If there is no return value, you don’t need to report the result.

|  |
| --- |
| [delete index] |
| [create index] |
| [insert only one document] |
| [Delete one document] |
| [insert all document using bulk] |
| [get document] |
| [get count of all documents in the index] |
| [search document] |

1. **[10pts]** Compare the difference in execution time between basic and helper operations and report the result and your findings.

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

* 1. **[5pts]** Compare the difference in execution time between performing ‘bulk’ and ‘insert\_one\_document’ multiple times and explain the reason.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | method | w/o bulk | w/ bulk | | Time consumed | 1418.2433474063873 seconds | 23.01433706283569 seconds |   [Execution time report] |
| [explanation]  Using bulk API was significantly faster than inserting data without bulk. This is because of reduced overhead when inserting documents by sending multiple documents in a single request, rather than sending one by one without using bulk. |

* 1. **[5pts]** Compare the difference in execution time between performing ‘scan’ and ‘search’ multiple times and explain the reason.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [Execution time report: search and scroll]   |  |  |  | | --- | --- | --- | | Query | Query 1 | Query 2 | | count | 100789 | 6781 | | Time consumed | 6.6975953578949 seconds | 6.07894802093506 seconds | |
| [Execution time report: scan]   |  |  |  |  | | --- | --- | --- | --- | | Query | | Query 1 | Query 2 | | count | | 87571(10pages), 100789(1000pages) | 6781(10pages), 6781(1000pages) | | Page size | 10 | 17.140448570251465 seconds | 1.296100616455078 seconds | | 1000 | 0.8979315757751465 seconds | 0.120013952255249 seconds | |
| [explanation]   1. Compare between scan and search & scroll   Scan() is faster than the other, because scan() is designed for efficient retrieval of large result sets. It is optimized for scrolling through large numbers of document efficiently. Using scan() makes Elasticsearch prefetch the next batch of result to make faster work.   1. Compare between page size in scan()   If page size is larger, it is both faster and memory-intensive. A large page size lead to fast retrieval of documents because more documents are fetched in each batch. Also, large page size require more memory to store the retrieved document in memory before they are returned to the client. |