**MongoDB Performance Analysis Report**

**Overview**This report outlines the methodologies and findings from a comparative analysis of query execution times between MySQL and MongoDB (NoSQL). The primary objective was to evaluate MongoDB's performance under various data extraction scenarios using complex querying techniques.**Methodology**Two collections, named p2 and project1, were created within the admin and config databases, respectively. The dataset (final\_dataset.csv) in consideration was initially uploaded to the p2 collection of the admin database through MongoDB Compass.To facilitate a comprehensive analysis, the dataset was also inserted into the project1 collection of the config database. This enabled the execution and timing of insertion, deletion, and update operations using Python scripts.In the subsequent phase, the dataset was divided into four subsets and loaded into four distinct collections: Drug\_product\_final, FDA, Drugs, and Manufacturers. This division was aimed at dissecting the aggregation pipeline's performance, particularly when utilizing the $lookup operator to merge data across multiple collections.All queries were constructed and executed using MongoDB's aggregation pipelines, with tests run both in Compass and a Python editor. A connection between MongoDB and Python was established to facilitate this process.**Findings**The execution times for each query were meticulously recorded, emphasizing the system's responsiveness to different data extraction methods. It is noteworthy that due to the considerable size of the dataset, the results of the queries were not printed. Instead, only the execution times were displayed to provide a clear measure of performance. However, the code to print the results is available and can be found in the comment section (#) at the end of each respective .py file.**Conclusions**The execution time analysis provides valuable insights into MongoDB's capabilities in handling large datasets and complex queries. It also offers a comparative perspective against traditional SQL databases like MySQL. The findings from this study will serve as a useful reference for database performance optimization and selection.**Documentation**The code for the various data extraction processes is documented across several files:Files **DW** through **DW8** include code for data extraction from multiple collections.Files **dwm1** through **dwm8** encompass code for data extraction from a single collection.

**Execution times of code involving single collection :**

**dwm1.py :** runfile('D:/Project DW Mongo/dwm1.py', wdir='D:/Project DW Mongo')

The execution time of the pipeline is 2.4246811866760254 seconds.

**dwm2.py** : runfile('D:/Project DW Mongo/dwm2.py', wdir='D:/Project DW Mongo')

The execution time of the pipeline is 3.6467363834381104 seconds.

**dwm3.py** : runfile('D:/Project DW Mongo/dwm3.py', wdir='D:/Project DW Mongo')

The execution time of the pipeline is 3.3315622806549072 seconds.

**dwm4.py** : runfile('D:/Project DW Mongo/dwm4.py', wdir='D:/Project DW Mongo')

The execution time of the pipeline 1.9538829326629639 seconds.

**dwm5.py** : runfile('D:/Project DW Mongo/dwm5.py', wdir='D:/Project DW Mongo')

The execution time of the pipeline is 3.3854403495788574 seconds.

**dwm6.py** : runfile('D:/Project DW Mongo/dwm6.py', wdir='D:/Project DW Mongo')

The execution time of the pipeline is 2.325939655303955 seconds.

**dwm7.py** : runfile('D:/Project DW Mongo/dwm7.py', wdir='D:/Project DW Mongo')

The execution time of the pipeline is 3.9536590576171875 seconds.

**dwm8.py** : runfile('D:/Project DW Mongo/dwm8.py', wdir='D:/Project DW Mongo')

The execution time of the pipeline is 2.805009126663208 seconds.

**Execution times of code involving multiple collections**

**DW.py** : runfile('C:/Users/hp/Downloads/DW.py', wdir='C:/Users/hp/Downloads')

The execution time of the pipeline is 1.1473450660705566 seconds.

**DW2.py** : runfile('C:/Users/hp/Downloads/DW2.py', wdir='C:/Users/hp/Downloads')

The execution time of the pipeline is 286.9734880924225 seconds.

**DW3.py** : runfile('C:/Users/hp/Downloads/DW3.py', wdir='C:/Users/hp/Downloads')

The execution time of the pipeline is 288.9863336086273 seconds.

**DW4.py** : runfile('C:/Users/hp/Downloads/DW4.py', wdir='C:/Users/hp/Downloads')

The execution time of the pipeline is 332.98991084098816 seconds.

**DW5.py** : runfile('C:/Users/hp/Downloads/DW5.py', wdir='C:/Users/hp/Downloads')

The execution time of the pipeline is 105.63162279129028 seconds.

**DW6.py** : runfile('C:/Users/hp/Downloads/DW6.py', wdir='C:/Users/hp/Downloads')

The execution time of the pipeline is 347.2062292098999 seconds

**DW7.py** : runfile('C:/Users/hp/Downloads/DW7.py', wdir='C:/Users/hp/Downloads')

The execution time of the pipeline is 96.32694220542908 seconds.

**DW8.py** : runfile('C:/Users/hp/Downloads/DW8.py', wdir='C:/Users/hp/Downloads')

The execution time of the pipeline is 101.5340096950531 seconds

**Insertion, Updation and Deletion execution times**

**Insert.py** : runfile('C:/Users/hp/Downloads/insert.py', wdir='C:/Users/hp/Downloads')

Import time: 123.19409918785095 seconds

**Update1.py :**  “ runfile('C:/Users/hp/Downloads/Update1.py', wdir='C:/Users/hp/Downloads')

Matched 1487411 documents and modified 1487411 documents.

The execution time of the update operation is 66.30523347854614 seconds.

**Delete.py** : runfile('C:/Users/hp/Downloads/delete.py', wdir='C:/Users/hp/Downloads')

1487411 documents deleted successfully.

Deletion time: 40.03939485549927 seconds

**Note : execution times may vary based on each system’s capabilities.**