Lab Session 11

11. NoSQL Indexing

Aim:

The aim of this experiment is to improve query performance and data retrieval efficiency by creating data structures that allow for faster search and retrieval of information. Indexing enables MongoDB to locate and retrieve specific documents more efficiently, reducing the need to scan the entire collection.

Description:

Indexing in MongoDB is a mechanism that enhances query performance and data retrieval efficiency. It involves creating data structures, called indexes, that provide a way to access and retrieve specific information from a MongoDB collection quickly. When indexing is applied to a field or a combination of fields, MongoDB creates a separate data structure that organizes the indexed field(s) in a sorted manner. This structure, known as the index, contains references to the actual documents in the collection.

Pre-Requisites:

- Choose a NoSQL database to work with (e.g., MongoDB, Cassandra, Redis)
- Install the database software on your local machine or use a cloud-based service.
- Set up a development environment with appropriate drivers or libraries for your chosen NoSQL database.
- Access the database through a command-line interface or a graphical user interface.

Pre-Lab:

1. What is the role of the Wired Tiger storage engine in MongoDB?

WiredTiger is MongoDB's default storage engine that provides high performance, compression, and concurrency for read/write operations.

2. What is the purpose of the \$graphLookup operator in MongoDB	aggregation?
---	--------------

Used in aggregation to perform recursive searches (like traversing parent-child or tree structures within a collection).

- 3. What is the difference between a TTL index and a regular index in MongoDB?
- TTL (Time-To-Live) index automatically deletes expired documents.
- Regular index only improves query performance and doesn't delete data.

4. What is the purpose of the \$replaceRoot operator in MongoDB aggregation?

Replaces the current document with a specified embedded document, often used to reshape output in pipelines.

5. What is the role of the mongodump command in MongoDB?

mongodump creates a binary backup of a MongoDB database or collection for data backup and restore.

IN LAB:

To demonstrate the optimization of queries using indexes, we have to use a fairly large database where query execution times are longer. For the following examples, we'll use a dataset of daily NASDAQ summaries. To follow along, you'll need this data locally.

Develop a query to demonstrate Text search using catalog data collection for a given word.

To demonstrate Text Search using indexes on a large dataset like NASDAQ daily summaries, we first need to:

- 1. Create a collection (e.g., catalog).
- 2. Insert sample NASDAQ summary documents into it.
- 3. Create a text index.
- 4. Run a query that performs a full-text search using a specific word.

Step-by-step

1. Sample NASDAQ Summary Document (in catalog collection)

```
{
  "symbol": "AAPL",
  "date": "2024-04-17",
  "summary": "Apple Inc. saw a strong performance with increased trading volume and positive investor sentiment."
}
```

2. Create Text Index

To enable full-text search on the summary field:

```
db.catalog.createIndex({ summary: "text" })
```

3. Perform a Text Search Query

Let's say we want to find all documents with the word "performance" in the summary.

```
db.catalog.find({ $text: { $search: "performance" } })
```

4. Use Explain to Show Optimization

To demonstrate the performance impact and optimization:

db.catalog.find({ \$text: { \$search: "performance" } }).explain("executionStats")

This will show how efficiently the query is executed using the text index.

1. What is the purpose of indexing in MongoDB?

To speed up query performance by avoiding full collection scans.

2. How does indexing improve query performance in MongoDB?

It lets MongoDB quickly locate data using the index instead of scanning every document.

3. Explain the concept of an index in MongoDB.

4. What are the benefits of creating indexes in frequently queried fields?

- Faster queries
- Reduced server load
- Better read performance

5. Can you create indexes on multiple fields in MongoDB?

```
js

db.collection.createIndex({ field1: 1, field2: -1 })
```

Post Lab:

MongoDB Indexing using a MongoDB Driver Objective: To create and use indexes in MongoDB for improved query performance using a MongoDB driver (e.g., pymongo for Python).

Materials: MongoDB installed and running (follow the installation steps mentioned earlier)

• MongoDB driver for your programming language (e.g., pymongo for Python)

Procedure:

- 1. Install the MongoDB driver for your chosen programming language (e.g., pymongo for Python).
- 2. Open a text editor or IDE and create a new script file.
- 3. Import the required MongoDB driver module in your script.
- 4. Establish a connection to the MongoDB server using the driver's connection methods.
- 5. Switch to the desired database using the driver's methods (e.g., client.get_database() in pymongo).
- 6. Identify the collection on which you want to create an index. For example, consider a collection named mycollection.
- 7. To create an index on a specific field, use the appropriate method provided by the driver (e.g., db.collection_name.create_index() in pymongo). For example, to create an index on the "name" field, use the following code: db.collection_name.create_index([("name", pymongo.ASCENDING)]) This code creates an ascending index on the "name" field. You can use pymongo.DESCENDING for a descending index.
- 8. Once the index is created, perform query operations on the collection and observe the improved query performance.
- 9. Optionally, you can use the appropriate method provided by the driver (e.g., db.collection_name.index_information() in pymongo) to view the list of indexes created on a collection.

print(collection.index_information())

Students Signature

(For Evaluator's use only)

Comment of the Evaluator (if Any)	Evaluator's Observation Marks Secured: out of
	Full Name of the Evaluator:
	Signature of the Evaluator Date of Evaluation: