Caitlin Atkins
D597 – Data Management
September 3, 2024

Task 2 – Non-Relational Database Design and Implementation

Part 1: Design Document

A. Select one of the provided scenarios and complete the following:

Scenario 2

1. **Describe a business problem that can be solved with a database solution and is aligned with the chosen scenario.**

Now, EcoMart relies on a relational database to handle its product information. As the company expands and adds more product information, each with its own identifying factors, the database will need a way to keep up with the increasing complexity and variability. The unforgiving structure of a relational database makes it difficult to store all the information that EcoMart requires.

This is because relational databases must catch up with the additional information, leading to inefficiency and difficulty scaling. This limits EcoMart's ability to expand and scale its platform horizontally, optimize user experience performance, and handle increasing data and user traffic.

Without failure, the business must store and manage product details like descriptions, pricing, availability, sustainability, certifications, and user reviews. EcoMart needs a flexible and scalable solution to expand the platform.

2. **Justify why a NoSQL database solution will solve the identified business problem.**

Because EcoMart needs a flexible and scalable database to expand the platform horizontally and quickly store different kinds of data, a NoSQL database is perfect for the company's needs. NoSQL databases, like MongoDB, are ideal for flexibility in storing and manipulating various data types.

According to MongoDB's website, NoSQL databases are designed to move away from the strictness and inflexibility that relational databases hold to offer a flexible schema that can adapt to changes in the needs of businesses or data patterns (*NoSQL Database Design and Data Modeling, 2024).*

Based on these critical points of NoSQL, EcoMart will thrive most with a NoSQL database to hold space for its ever-changing product attributes and user demographics.

3. **Identify a NoSQL database type to solve the identified business problem.**

Of all the existing NoSQL databases, based on the information that needs to be stored and organized, EcoMart will be able to solve its identified business problem most efficiently with the help of MongoDB. This NoSQL database is document-based; this database style allows for various datatypes in any volume. Thus, MongoDB is excellent at "handling unstructured data and is particularly suitable for projects requiring agility and scalability" (Western Governors University, 2020).

MongoDB will allow all data, regardless of datatype, to be stored in one document. This makes it significantly easier to store, look up, and manage EcoMart's information, regardless of product numbers or users of the database.

4. Explain how the database solution will use the business data.

When using MongoDB, each product will be stored as a document; the product's relevant details, including pricing, user reviews, sustainability certifications, availability, and other specific information, will all be found in that document.

Before adapting to MongoDB, EcoMart's relational database spread the different aspects of each product into multiple tables. MongoDB will ensure that all information is found easily in one document. EcoMart can efficiently update, manage, and search for products through this strategy.

EcoMart can use this database solution to search for products. When all products are found in the same database, product retrieval based on price, ratings, keywords, or other identifications becomes more manageable and quicker. With this efficient retrieval, EcoMart can explore real-time insights to identify trending and poorly performing products. Understanding the insights to this degree makes inventory management significantly easier; EcoMart knows which products should have a higher inventory and where discounts should be applied. Finally, the database solution allows a more personalized customer service experience by allowing access to order histories and preferences.

For this task, queries will be made to explore basic data retrieval and product insight. First, two collections will be created to separate cosmetics and groceries. This is because the two data files will likely never interact with each other; it makes more sense for efficiency to keep the two files separate. After developing these collections as documents, two data retrieval queries demonstrate that keywords and brands can easily be recalled with MongoDB. Finally, product insights are presented by creating a query that shows which cosmetic brands have high ratings.

With this database in mind, the user experience is improved while the database is efficient during the company expansion and introduction of new products.

B. Discuss how the proposed database design addresses scalability concerns, including strategies that align with the chosen scenario.

When EcoMart uses a NoSQL database like MongoDB, there are no significant horizontal scalability concerns. These needs have already been met using a NoSQL database like MongoDB. This is because NoSQL databases, specifically MongoDB, are scalable designs already capable of handling more data and users by being document-based. By being

document-based, MongoDB can enable the database to perform the same way regardless of the products and users in the database.

While MongoDB allows the database to scale and run efficiently, faster performance might be needed. To ensure this need is met, EcoMart can use caching to reduce the load on the database.

C. Outline the privacy and security measures that should be implemented in the proposed database design.

When handling sensitive information related to EcoMart's business actions and storage, it is essential and critical to use privacy and security measures to protect the company and its user's information.

According to MongoDB's website, some techniques and measures can be used to ensure optimal security. Methods including authentications, encryptions, role-based authorizations, and regular backup and recovery systems can protect the company (*Security Checklist for Self-Managed Deployments*, 2024).

Authentication methods can be used to verify identities and ensure that the right people are accessing the sensitive data. Encryption should be used for all incoming and outgoing connections. This is to make sure that data remains unreadable in the case of being intercepted or accessed without authorization. Role-based authorizations allow users to follow the principle of least privilege by creating roles that define the exact access rights required. Finally, regular backups should occur to ensure that data can be restored during data loss or corruption.

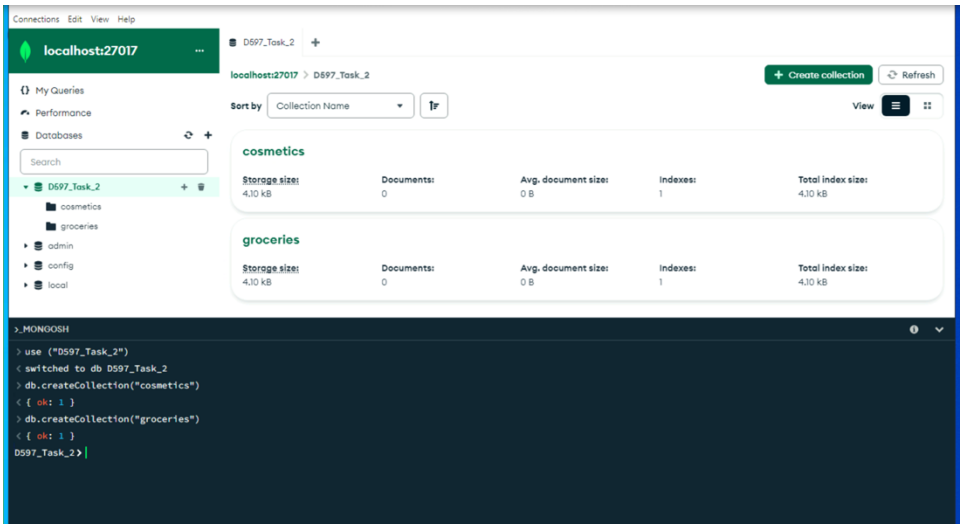
Part 2: Implementation

Note: The data files for each scenario are in a folder titled "D597 Datasets" on the desktop of the WGU Virtual Lab environment. Pull the files from "Task 2" related to your chosen scenario.

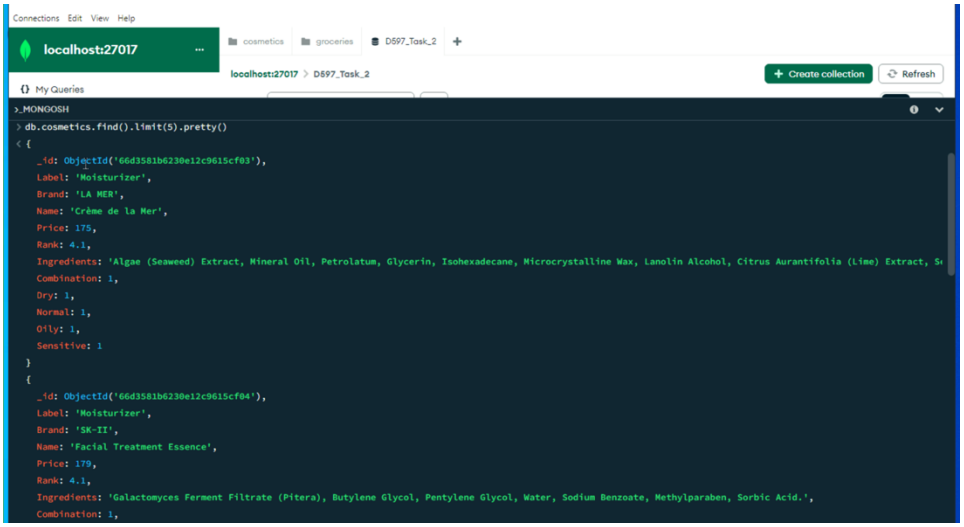
Note: Submit your screenshots from the WGU Virtual Lab for each prompt with your design document.

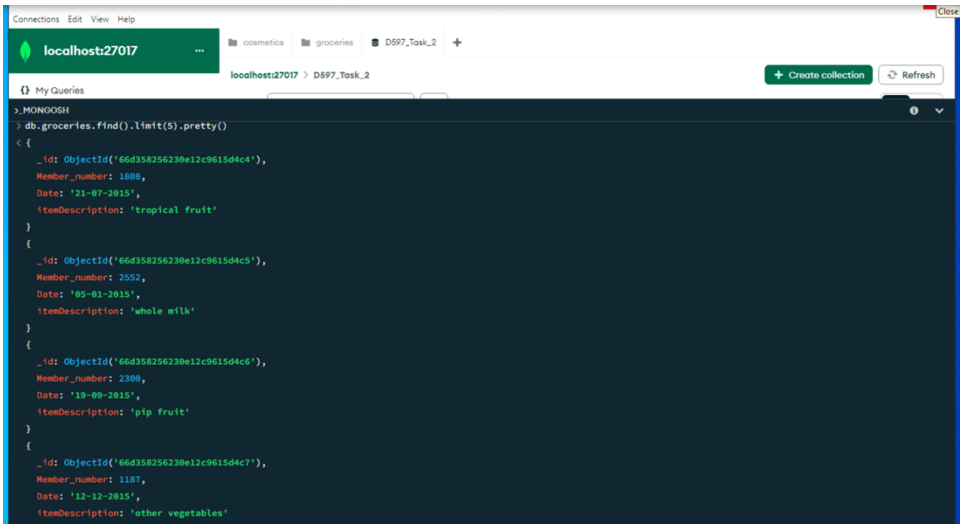
D. Implement the proposed database design in the WGU Virtual Lab environment by completing the following:

1. **Write a script to create a database instance named "D597 Task 2" using the appropriate query language based on your design in Part 1. Provide a screenshot showing the script and the database instance in the platform.**

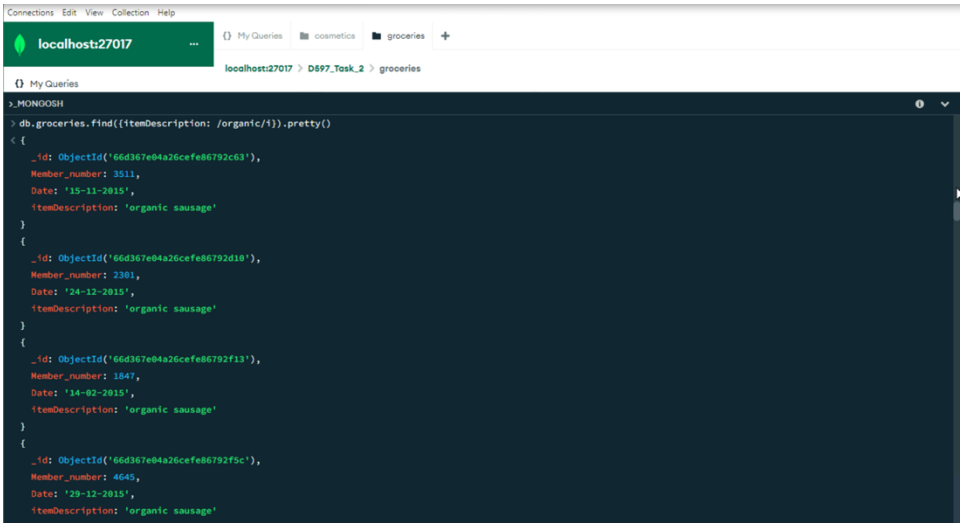


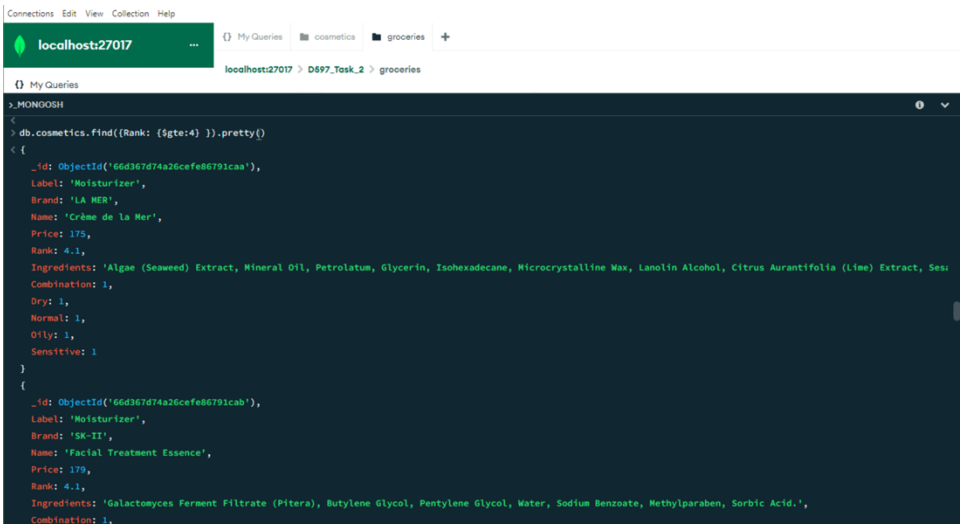
2. **Write a script to insert or map the data records from the chosen scenario JSON files into the database instance. Provide a screenshot showing the script and the data correctly inserted or mapped into the database.**

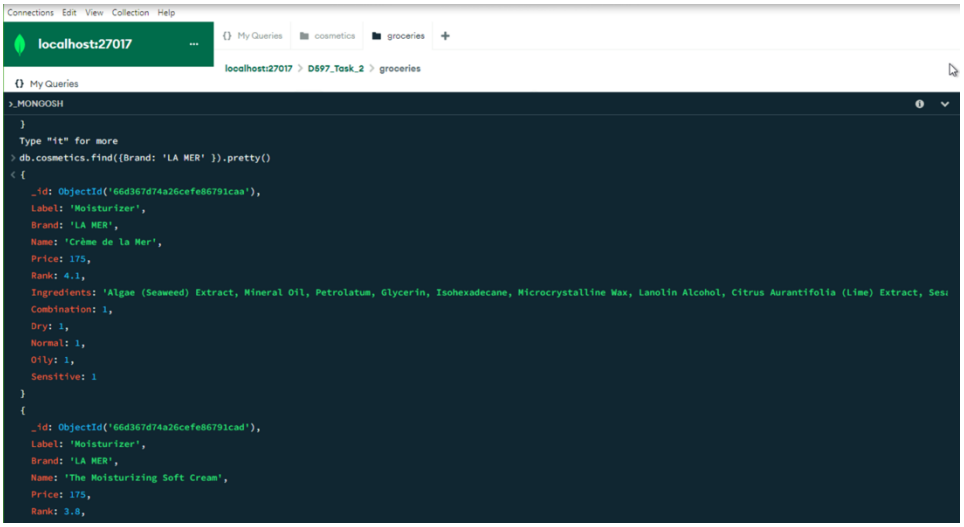




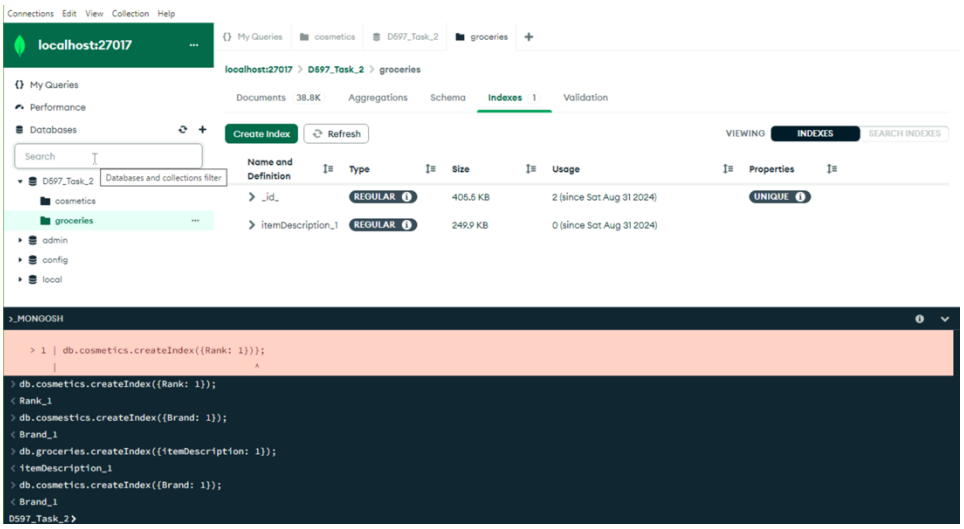
3. **Write a script for** three **queries to retrieve specific information from the database to help solve the identified business problem. Provide a screenshot showing the script for each query successfully executed.**

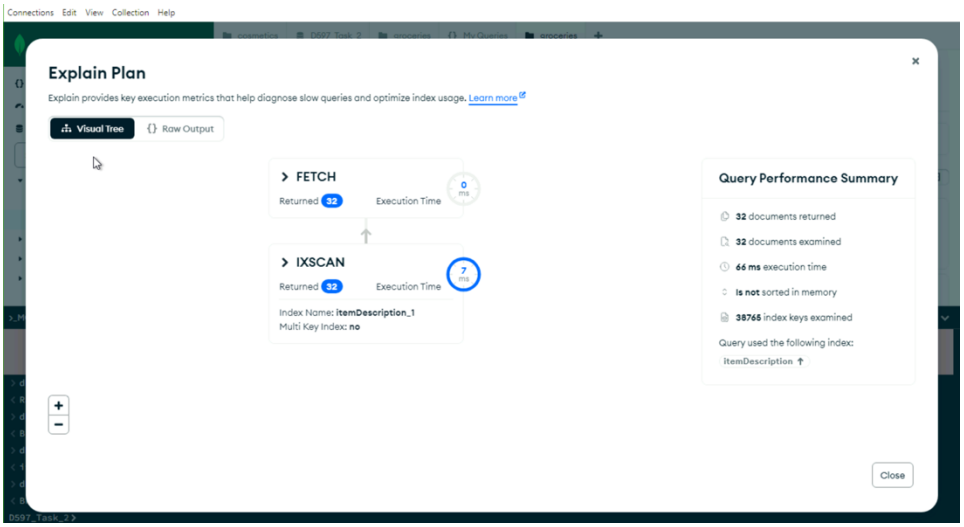


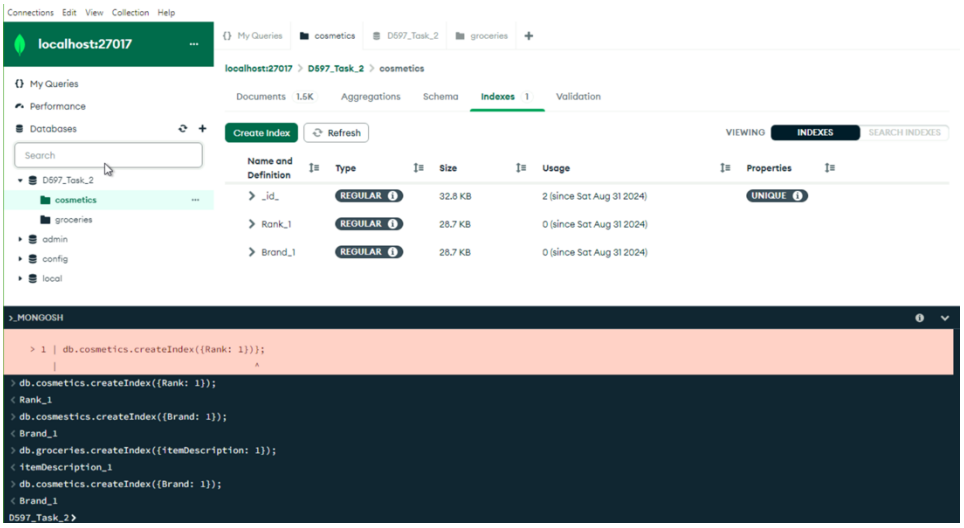


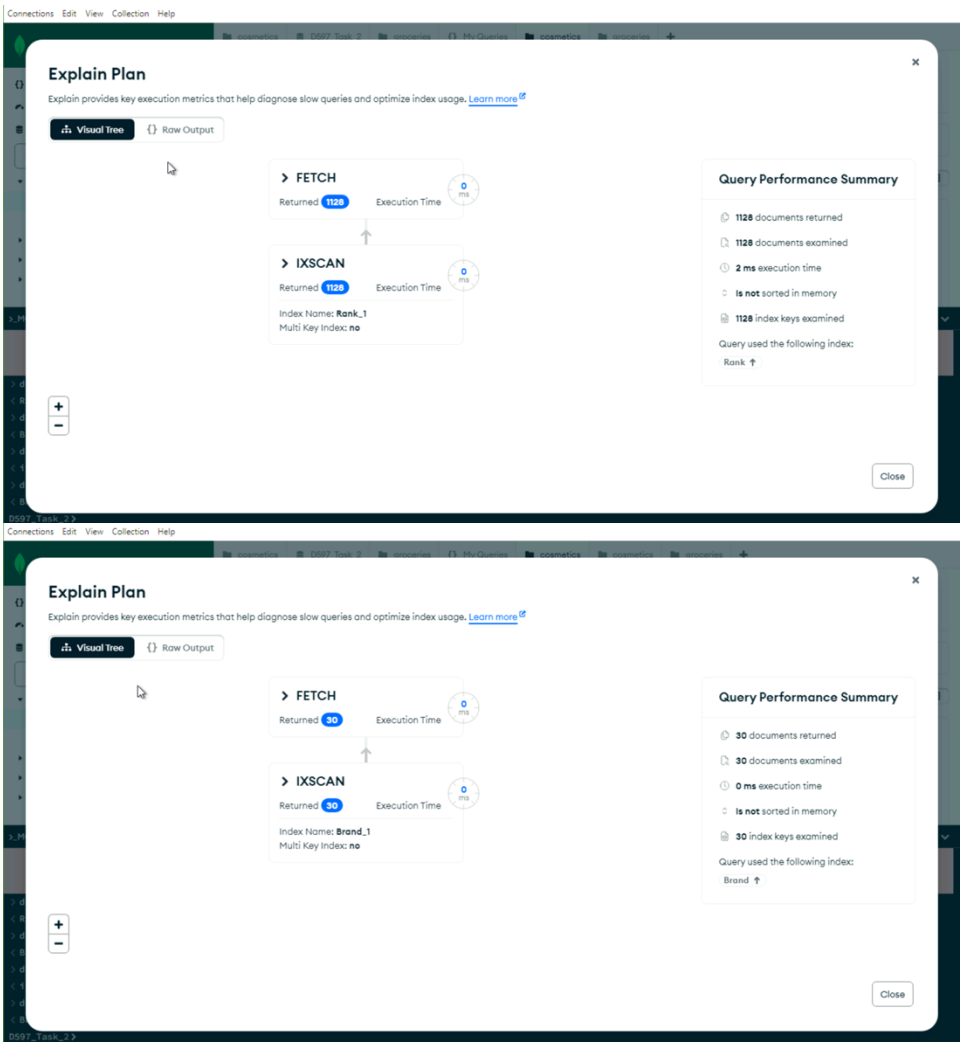


4. Apply optimization techniques to improve the run time of your queries from part D3, providing output results via a screenshot.


Connections Edit View Collection Help
localhost:27017
My Queries
Performance
Databases
Search
Databases and collections filter
D597_Task_2
cosmetics
groceries
admin
config
local
My Queries
cosmetics
D597_Task_2
groceries
localhost:27017 > D597_Task_2 > groceries
Documents 38.8K
Aggregations
Schema
Indexes 1
Validation
Create Index
Refresh
VIEWING
INDEXES
SEARCH INDEXES
Name and Definition
Type
Size
Usage
Properties
_id_
REGULAR
406.5 KB
2 (since Sat Aug 31 2024)
UNIQUE
itemDescription_1
REGULAR
249.9 KB
0 (since Sat Aug 31 2024)
>_MONGOSH
> 1 | db.cosmetics.createIndex({Rank: 1});
> db.cosmetics.createIndex({Rank: 1});
< Rank_1
> db.cosmestics.createIndex({Brand: 1});
< Brand_1
> db.groceries.createIndex({itemDescription: 1});
< itemDescription_1
> db.cosmetics.createIndex({Brand: 1});
< Brand_1
D597_Task_2 >


Connections Edit View Collection Help
Explain Plan
Explain provides key execution metrics that help diagnose slow queries and optimize index usage. Learn more
Visual Tree
Raw Output
FETCH
Returned 32
Execution Time
0 ms
IXSCAN
Returned 32
Execution Time
7 ms
Index Name: itemDescription_1
Multi Key Index: no
Query Performance Summary
32 documents returned
32 documents examined
66 ms execution time
is not sorted in memory
38765 index keys examined
Query used the following index:
itemDescription
Close


Connections Edit View Collection Help
localhost:27017
My Queries
Performance
Databases
Search
D597_Task_2
cosmetics
groceries
admin
config
local
My Queries
cosmetics
D597_Task_2
groceries
localhost:27017 > D597_Task_2 > cosmetics
Documents 1.5K
Aggregations
Schema
Indexes 1
Validation
Create Index
Refresh
VIEWING
INDEXES
SEARCH INDEXES
Name and Definition
Type
Size
Usage
Properties
_id_
REGULAR
32.8 KB
2 (since Sat Aug 31 2024)
UNIQUE
Rank_1
REGULAR
28.7 KB
0 (since Sat Aug 31 2024)
Brand_1
REGULAR
28.7 KB
0 (since Sat Aug 31 2024)
>_MONGOSH
> 1 | db.cosmetics.createIndex({Rank: 1});
> db.cosmetics.createIndex({Rank: 1});
< Rank_1
> db.cosmestics.createIndex({Brand: 1});
< Brand_1
> db.groceries.createIndex({itemDescription: 1});
< itemDescription_1
> db.cosmetics.createIndex({Brand: 1});
< Brand_1
D597_Task_2 >



E. Acknowledge sources, using in-text citations and references, for content that is quoted, paraphrased, or summarized.

NoSQL Database Design and Data Modeling. (2024). MongoDB. Retrieved August 29, 2024, from https://www.mongodb.com/resources/basics/databases/nosql-explained/data-modeling

Security Checklist for Self-Managed Deployments. (2024). MongoDB. Retrieved August 29, 2024, from https://www.mongodb.com/docs/manual/administration/security-checklist/

Western Governors University (2020). *NoSQL Database Overview.* https://apps.cgp-oex.wgu.edu/wgulearning/course/course-v1:WGUx+OEX0343+v01/block-v1:WGUx+OEX0343+v01+type@sequential+block@35aff5e9d6284e40a79af95a9b20bacb/block-v1:WGUx+OEX0343+v01+type@vertical+block@755b27dd08a34af9985fd4586c723728