**Task 2: Non-Relational Database Design and Implementation**

Joseph Emmanuel Cayetano

Western Governors University

D597: Data Management Task 2

September 20, 2024

**A. Select one of the provided scenarios**

I chose scenario 2, which is about an online marketplace named EcoMart that sells a diverse range of environmentally friendly products to people that want to help the planet Earth.

**A1. Describe a business problem that can be solved with a database solution that aligns with scenario 2**

Since EcoMart is an emerging online marketplace that offers a diverse range of eco-friendly products such as cosmetics and groceries, it is possible that they could get overwhelmed with the increasing amount of data they have to handle. For example, if they were to use a traditional relational database, it would be difficult for them to adjust their database to accommodate new product types or attributes because relational databases do not have flexible schemas. Also, creating separate tables for different products can be time-consuming and can degrade database performance considering many tables will need to be joined when performing complex queries. This business problem can be solved with the use of MongoDB’s document-based structure. Using a MongoDB database solution would allow EcoMart to manage their diverse and growing eco-friendly product catalog efficiently.

**A2. Justify why a NoSQL database solution will solve the business problem**

A NoSQL database solution like MongoDB will solve the business problem that EcoMart could potentially face. MongoDB’s flexible and dynamic schema would allow EcoMart to accommodate the evolving attributes of their eco-friendly products. For example, MongoDB allows the use of arrays for fields, which makes it possible to store related data together within the same document without needing to create separate tables. This is useful regarding the use of both EcoMart’s datasets, where, for instance, you could easily add or modify an ingredient in a cosmetic product with the help of arrays. Also, arrays simplify data modeling, making it easier to retrieve data since there is no need for complex joins. Additionally, eliminating the need for joins means faster data access. Using a MongoDB database solution would be a perfect fit for EcoMart’s database needs.

**A3. Identify a NoSQL database type to solve the business problem**

The NoSQL database type that will be used to solve EcoMart’s business problem is a documented-oriented MongoDB. MongoDB would fit nicely for an e-commerce platform like EcoMart because it has flexible schema, which allows EcoMart to store increasing number of products with varying attributes without being limited by a fixed schema that can be seen in relational databases. For instance, both the groceries and cosmetics collections have their own different structure with no constraints between them. This allows EcoMart to efficiently manage their dynamic and evolving product catalog without any problem. Another reason why MongoDB is a perfect choice for EcoMart is its ability to handle growing amounts of data and traffic very well as the e-commerce platform expands.

**A4. Explain how the business data will be used in the database solution**

Here is how EcoMart’s business data will be used within the database solution. First, the cosmetics collection contains detailed information about individual cosmetic products. It has several fields in it such as id, label, brand, name, price, rank, ingredients, and skin types. The data in the cosmetics collection can be used to find cosmetic products that are within a certain price range, contains a specific ingredient, highest-ranked, etc. Second, the groceries collection contains information related to individual grocery purchases. It has four fields, and they are id, member number, date, and item description. The data in the groceries collection can be used to identify trends in customer purchasing behavior, such as frequently bought products at specific times of the year.

**B. Discuss how the database design addresses scalability concerns for scenario 2**

The proposed NoSQL database design will solve EcoMart’s scalability concerns in multiple ways. First, having a documented-oriented database solution like MongoDB allows EcoMart to store diverse product data like cosmetics and grocery items without requiring rigid schemas. It makes it easy for EcoMart to handle new product attributes and categories as the product catalog evolves without requiring major schema changes. Second, MongoDB supports powerful indexing techniques that enables fast query response times, even as the data volume increases. Third, MongoDB supports Sharding, which is “a technique used to distribute a database horizontally across multiple nodes or servers” (Tonete, 2024, par. 2). This would help EcoMart manage their large amounts of data and increasing customer traffic.

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

There are security and privacy measures that should be implemented in the proposed database design for EcoMart. The first measure is TLS, which is Transport Layer Security. It is important that EcoMart uses TLS to encrypt the data transmitted between applications and the database. Another measure that EcoMart should implement is Multi-Factor Authentication (MFA). MFA improves security by requiring multiple forms of verification before gaining access to data. Backing up the data is also important in case of system failures.

**D1. Write script to create a database instance named “D597 Task 2” based on your design in Part 1. Provide a screenshot.**

A screenshot of a computer

Description automatically generated

Writing a script to create a database requires the tool called mongoimport. However, mongoimport is not installed on the virtual machine, so I used MongoDB Compass to create the “D597\_Task2” database.

A screenshot of a computer

Description automatically generated

This is what the “D597\_Task2” database looks like after creating it.

A screenshot of a computer

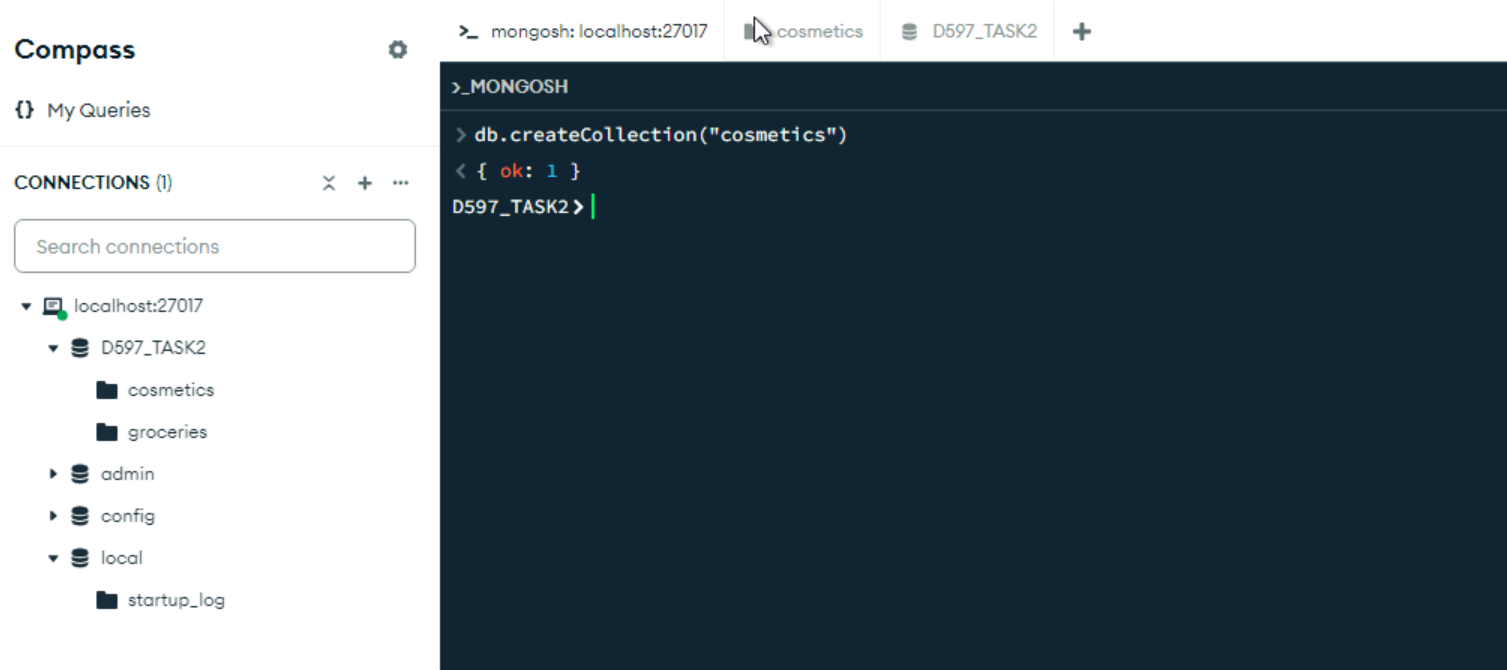
Description automatically generated

This is the script that I used to create the Groceries collection.

A screenshot of a computer

Description automatically generated

This is what the Groceries collection looks like after creating it. It currently has no data.



This is the script I used to create the Cosmetics collection.

A screenshot of a computer

Description automatically generated

This is what the Cosmetics collection looks like after creating it. It currently has no data.

**D2. Write script to insert data from the chosen scenario JSON files into the database instance. Provide a screenshot.**

A screenshot of a computer

Description automatically generated

This is the script to import the data from D597\_Groceries.json into the Groceries collection.

A screenshot of a computer

Description automatically generated

This is what the Groceries collection looks like after the import. There are about 38,000 documents in the collection and each document has fields such as id, member number, date, and item description. I will be converting the date format into yyyy-m-d for query purposes.

A screen shot of a computer

Description automatically generated

This is the script to import the data from D597\_Cosmetics.json into the Cosmetics collection.

A screenshot of a computer

Description automatically generated

This is what the Cosmetics collection looks like after the import. There are about 1,400 documents in the collection and each document has fields such as id, label, brand, name, price, rank, ingredients, combination, dry, normal, oily, and sensitive. I will be creating a new field called “SkinTypes” and store the individual skin type fields into a single array field. I will also convert the ingredients field into an array field.

**D3. Write script for three queries to retrieve specific information from the database that help solve the business problem. Provide a screenshot.**

A screenshot of a computer program

Description automatically generated

This is the first query. The query returns the brand and name of two cleansers that contain the ingredient “Allantoin” and are suitable for all skin types.

A screenshot of a computer

Description automatically generated

This is the second query. The query returns the label, brand, name, price, and rank of the highest ranked moisturizer between the price 50 and 200.

A screenshot of a computer program

Description automatically generated

This is the third query. The query returns the item description and count for the top five most bought items during the month of December in 2017.

**D4. Apply optimization techniques to improve the run time of your queries from part D3. Provide a screenshot.**

Before applying indexes to improve query performance, here are the totalDocsExamined counts for all three queries. TotalDocsExamined indicates the total number of documents examined to fulfill the query. Essentially, a higher totalDocsExamined count indicates the query is inefficient, and we want to reduce the count with the use of indexes.

A computer screen shot of a program

Description automatically generated

Before applying the indexes, query #1 has a totalDocsExamined value of 321.

A screenshot of a computer

Description automatically generated

Before applying the indexes, query #2 has a totalDocsExamined value of 1472.

A screenshot of a computer program

Description automatically generated

Before applying the indexes. Query #3 has a totalDocsExamined value of 38765.

A screenshot of a computer program

Description automatically generated

These are the indexes I created for both the Groceries and Cosmetics collections. In the Cosmetics collection, I created two indexes on the ingredients and price fields. In the Groceries collection, I created an index on the date field. These indexes improve query performance.

A screenshot of a computer

Description automatically generated

After applying the indexes, query #1’s totalDocsExamined value went from 321 to 24. The query’s performance significantly improved.

A screenshot of a computer screen

Description automatically generated

After applying the indexes, query #2’s totalDocsExamined value went from 1472 to 569. The query’s performance significantly improved.

A screenshot of a computer program

Description automatically generated

After applying the indexes. Query #3 totalDocsExamined value went from 38765 to 1536. The query’s performance significantly improved.

**E. Panopto Presentation**

Here is the link to my Panopto presentation:

<https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=a6d95330-daba-4a6b-9848-b1f1017d96cc>

**References**

Tonete, A. (2024, September 13*). A tutorial on mongodb sharding with best practices & when to enable it*. Percona Database Performance Blog. https://www.percona.com/blog/when-should-i-enable-mongodb-sharding/