

Robert Pavlik

D597 Data Management

MKN1 – Task2: Non-Relational Database Design and Implementation

Scenario 2

## Part 1: Design Documentation

A.

### 1. Business Problem:

EcoMart faces challenges with managing a diverse and constantly changing product inventory. With new items being added, existing ones updated, and seasonal products rotating regularly, the company requires a flexible data structure to handle varying product attributes. Traditional relational databases struggle with this complexity, especially when dealing with unstructured or semi-structured data like product descriptions, certifications, or user reviews, leading to inefficiencies and potential scalability issues.

### 2. NoSQL Solution:

A NoSQL database, particularly a document-based solution, addresses the flexibility needed for EcoMart's dynamic product catalog. By storing product data in JSON-like documents, NoSQL databases enable the platform to handle varying attributes and nested data structures without rigid schema constraints or relational databases. This flexibility is critical as EcoMart's product data grows and changes over time. Additionally, NoSQL solutions are well-suited for horizontal scaling, which is necessary to manage increasing traffic and data volume as EcoMart expands.

### 3. NoSQL Database Type:

A document-based NoSQL database, such as MongoDB, is ideal for EcoMart. MongoDB allows the storage of product data as JSON-like documents, supporting flexible schema designs. This enables EcoMart to easily store and query products with diverse attributes like descriptions, pricing, availability, and user reviews while accommodating sustainability certification, which may vary from product to product.

#### 4. Business Data Use:

The product data stored in the NoSQL database will be used to power various features of the EcoMart platform:

- **Product Listings:** Products with their descriptions, prices, availability, and sustainability certifications will be queried and displayed on the marketplace.
- **User Reviews:** Customer feedback and reviews, which can vary in structure, will be stored and retrieved dynamically for each product.
- **Search and Filter:** The NoSQL database will support searching products based on attributes like sustainability certifications, price ranges, or user ratings.
- **Inventory Management:** Availability and stock levels will be updated frequently as products are purchased or new stock arrives

#### B. Scalability Strategies

The proposed NoSQL database design will incorporate several strategies to ensure EcoMart's platform scales efficiently.

- **Horizontal Scalability**
  - Document-based NoSQL databases like MongoDB are designed for horizontal scaling, which allows for distributing data across multiple servers (Sharding). As EcoMart grows, additional servers can be added to handle larger datasets and increase user traffic without affecting performance.
- **Sharding**
  - By portioning the database into smaller, more manageable pieces, each shard can be stored and processed on separate servers. Sharding will enable EcoMart to scale out easily as product categories and user base increase.
- **Replication**
  - Data replication will be implemented to ensure high availability and fault tolerance. Multiple copies of the database will be stored on different servers, ensuring that even in case of hardware failure, the platform remains operational with minimal downtime
- **Caching**
  - To enhance query performance, caching solutions like Redis can be integrated. Frequently accessed data, such as popular product listings or user reviews, will be cached to reduce the load on the database.

### C. Privacy and Security Measures

To protect sensitive customer data and comply with regulations such as GDPR and CCPA, the following privacy and security measures will be implemented

- Encryption
  - Both data-at-rest (stored data) and data-in-transit (data being transmitted between servers and users) will be encrypted using strong encryption protocols.
- Access Control
  - Role-based access control (RBAC) will be employed to limit access to the database based on user roles. This will ensure that only authorized personnel can access sensitive customer data.
- Audit Logging
  - Detailed logging of all database access and changes will be maintained. Audit logs will track who accessed the data, when and what actions were taken, which will help detect any suspicious or unauthorized activity.
- Backup and Recovery
  - Regular backups will be automated to ensure that EcoMart can recover quickly from data loss or corruption. These backups will be stored securely and tested for integrity.
- Compliance and Regulations
  - The database will be designed with GDPR and CCPA compliance in mind, including the ability to handle requests for data deletion and protection of personal customer information.

By adopting a NoSQL document-based database, EcoMart will have the flexibility, scalability, and security required to support its expanding marketplace while maintaining a commitment to data privacy and performance optimization.

## Part 2: Implementation

### D. Implement the proposed database design

#### 1. Create the database

×

## Create Database

Database Name

D597\_Task2

Collection Name

cosmetics

☐ Time-Series

Time-series collections efficiently store sequences of measurements over a period of time. [Learn More](#)

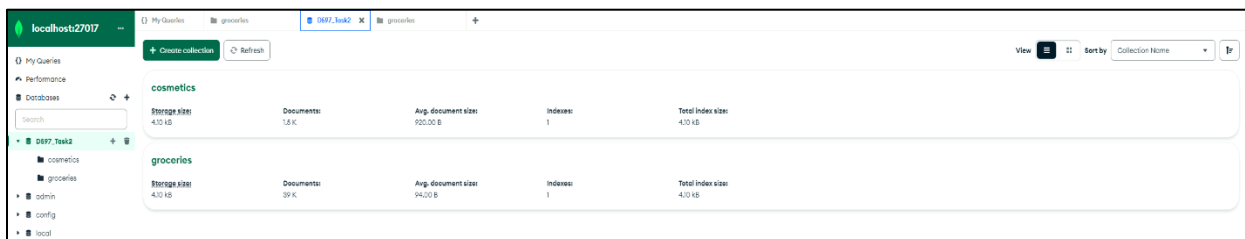
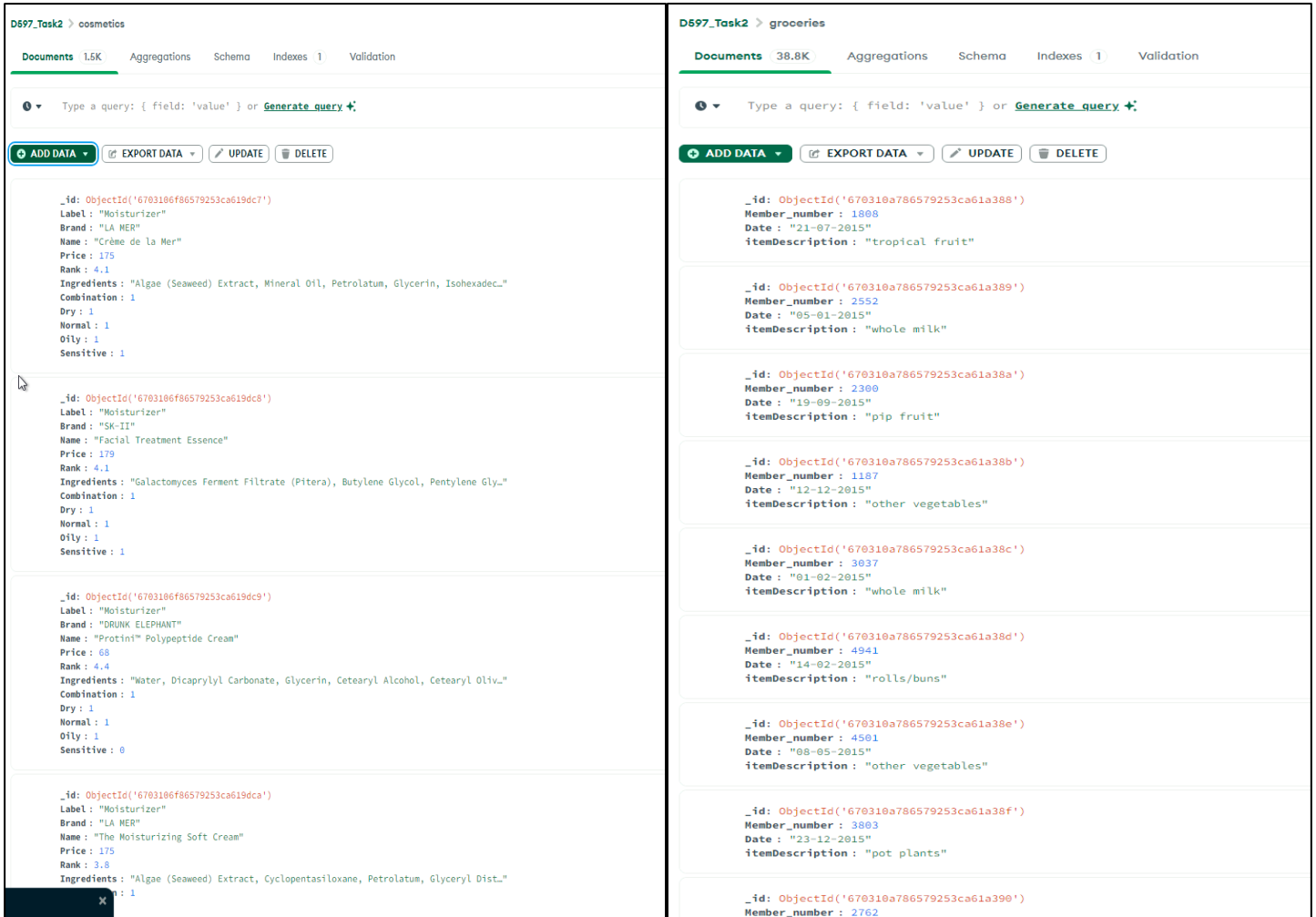
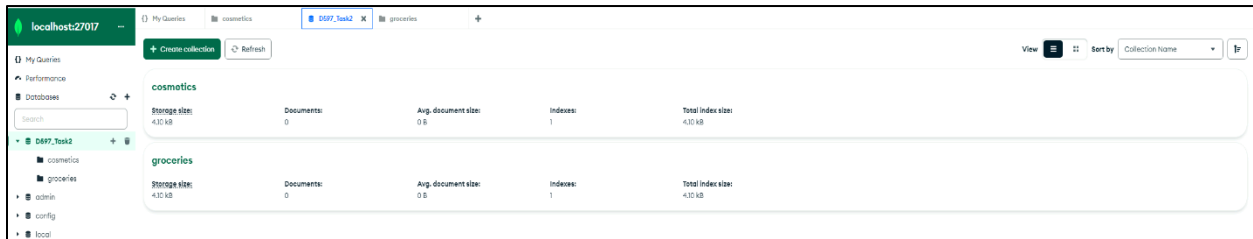
➤ Additional preferences

(e.g. Custom collation, Capped, Clustered collections)

Cancel

Create Database

## 2. Create a Script to insert data records from the chosen scenario JSON file



### 3. Create 3 Queries to retrieve specific information from the database

- Query 1

The screenshot displays the MongoDB Compass interface. At the top, the database is 'D597\_Task2' and the collection is 'cosmetics'. The query entered is `{ ("Rank": { $gt: 4 }) }`. Below the query, there are buttons for 'ADD DATA', 'EXPORT DATA', 'UPDATE', and 'DELETE'. The results section shows four documents, each representing a cosmetic product. Each document includes fields for \_id, Label, Brand, Name, Price, Rank, Ingredients, Combination, Dry, Normal, Oily, and Sensitive. The 'Rank' field for all four documents is greater than 4, as specified in the query. Below the results, there is an 'Explain Plan' section. It shows the query execution plan, including a 'COLLSCAN' operation. The 'CollSCAN' operation has a 'Returned' count of 4 and an 'Execution Time' of 0 ms. The 'Documents Examined' count is 1472. To the right of the explain plan, there is a 'Query Performance Summary' section. It lists the following metrics: 1038 documents returned, 1472 documents examined, 1 ms execution time, 16 MB sorted in memory, 0 index keys examined, and a warning that no index is available for this query.

Documents

Aggregations

Schema

Indexes

Validation

Query: `{ ("Rank": { $gt: 4 }) }`

ADD DATA EXPORT DATA UPDATE DELETE

Document 1:

```
{
  "_id": "ObjectID('6703196f86570233ca19dc7')",
  "Label": "Moisturizer",
  "Brand": "LA MER",
  "Name": "Crème de la Mer",
  "Price": 175,
  "Rank": 4.2,
  "Ingredients": "Algae (Seaweed) Extract, Mineral Oil, Petrolatum, Glycerin, Isohexadec-",
  "Combination": 1,
  "Dry": 1,
  "Normal": 1,
  "Oily": 1,
  "Sensitive": 1
}
```

Document 2:

```
{
  "_id": "ObjectID('6703196f86570233ca19dc8')",
  "Label": "Moisturizer",
  "Brand": "SK-II",
  "Name": "Facial Treatment Essence",
  "Price": 179,
  "Rank": 4.1,
  "Ingredients": "Galactomyces Ferment Filtrate (Pitera), Butylene Glycol, Pentylene Gly-",
  "Combination": 1,
  "Dry": 1,
  "Normal": 1,
  "Oily": 1,
  "Sensitive": 1
}
```

Document 3:

```
{
  "_id": "ObjectID('6703196f86570233ca19dc9')",
  "Label": "Moisturizer",
  "Brand": "DRUNK ELEPHANT",
  "Name": "Protini™ Polypeptide Cream",
  "Price": 68,
  "Rank": 4.4,
  "Ingredients": "Water, Dicaprylyl Carbonate, Glycerin, Cetearyl Alcohol, Cetearyl Oliv-",
  "Combination": 1,
  "Dry": 1,
  "Normal": 1,
  "Oily": 1,
  "Sensitive": 0
}
```

Document 4:

```
{
  "_id": "ObjectID('6703196f86570233ca19ddb')",
  "Label": "Moisturizer",
  "Brand": "IT COSMETICS",
  "Name": "Your Skin But Better™ CC+ Cream with SPF 50+",
  "Price": 38,
  "Rank": 4.3,
  "Ingredients": "Water, Snail Secretion Filtrate, Phenyl Trimethicone, Dimethicone, But-",
  "Combination": 1,
  "Dry": 1,
  "Normal": 1,
  "Oily": 1,
  "Sensitive": 0
}
```

Explain Plan

Explain provides key execution metrics that help diagnose slow queries and optimize index usage. [Learn more](#)

Visual Tree Raw Output

COLLSCAN

Returned: 4 Execution Time: 0 ms

Documents Examined: 1472

Query Performance Summary

- 1038 documents returned
- 1472 documents examined
- 1 ms execution time
- 16 MB sorted in memory
- 0 index keys examined
- No index available for this query.

This query uses the (\$gt) operator which stands for “greater than”. This query filters the documents in the cosmetic collection, returning all cosmetics that have a Rank greater than 4. This is used to find products with better customer ratings.

This query helps EcoMart identify the top-rated products. As a marketplace focused on eco-conscious products, EcoMart can feature products in promotions or recommend them to new customers

By quickly fetching high-rated products, EcoMart can ensure that customers are seeing the best, most trusted items, increasing satisfaction and trust in the platform.

- Query 2

The screenshot displays the MongoDB Query Shell interface. At the top, the database 'D597\_Task2' and collection 'groceries' are selected. The query filter is `{ "Member_number": 1808 }`. Below the query, there are buttons for 'ADD DATA', 'EXPORT DATA', 'UPDATE', and 'DELETE'. The results show a list of grocery items purchased by member 1808, including tropical fruit, long life bakery product, meat, sugar, rolls/buns, semi-finished bread, whole milk, and citrus fruit.

Below the results, the 'Explain Plan' section is visible. It shows the query execution plan, which is a COLLSCAN (Collection Scan). The plan indicates that 1472 documents were examined and 756 documents were returned. The execution time was 2 ms, and the query was not sorted in memory. The plan also shows that 8 index keys were examined, but no index was available for this query.

**Explain Plan**  
Explain provides key execution metrics that help diagnose slow queries and optimize index usage. [Learn more](#)

Visualize Row Output

**> COLLSCAN**  
Returned: 756 Execution Time: 2 ms  
Documents Examined: 1472

**Query Performance Summary**

- 756 documents returned
- 1472 documents examined
- 2 ms execution time
- is not sorted in memory
- 8 index keys examined
- No index available for this query.

This query retrieves all grocery items purchased by the member with Member\_number 1808. It scans the grocery collection and filters records where Member\_number matches 1808.

By retrieving the purchasing history of a specific user, EcoMart can tailor product recommendations to that individual's buying patterns. For example, if the user regularly buys organic fruits, EcoMart can recommend eco-friendly or organic products.

This helps EcoMart send personalized offers, promotions, and loyalty rewards based on individual purchase history, improving customer engagement and retention.

## • Query 3

My Queries

DBPT\_Task2

cosmetics

✕

+

DBPT\_Task2 > cosmetics

Documents 1,0K
Aggregations
Schema
Indexes 1
Validation

▼

{"Sensitive": 1 }

ADD DATA

EXPORT DATA

UPDATE

DELETE

.\_id: ObjectId("6780380f80079233ca648dc7")

Label: "Moisturizer"

Brand: "A.R.D"

Name: "Crème de la Mer"

Price: 175

Rank: 4.1

Ingredients: "Algae (Seaweed) Extract, Mineral Oil, Petrolatum, Glycerin, Isohexadec..."

Combination: 1

Dry: 1

Normal: 1

Oily: 1

Sensitive: 1

.\_id: ObjectId("6780380f80079233ca648dc8")

Label: "Moisturizer"

Brand: "SK-II"

Name: "Facial Treatment Essence"

Price: 170

Rank: 4.1

Ingredients: "Galactomyces Ferment Filtrate (Pitera), Butylene Glycol, Pentylene Gly..."

Combination: 1

Dry: 1

Normal: 1

Oily: 1

Sensitive: 1

.\_id: ObjectId("6780380f80079233ca648dc9")

Label: "Moisturizer"

Brand: "A.R.D"

Name: "The Moisturizing Soft Cream"

Price: 175

Rank: 3.8

Ingredients: "Algae (Seaweed) Extract, Cyclopentasiloxane, Petrolatum, Glyceryl Dist..."

Combination: 1

Dry: 1

Normal: 1

Oily: 1

Sensitive: 1

.\_id: ObjectId("6780380f80079233ca648dd0")

Label: "Moisturizer"

Brand: "T1 COSMETICS"

Name: "Your Skin But Better™ CC+ Cream with SPF 36+"

Price: 30

Rank: 4.1

Ingredients: "Water, Snail Secretion Filtrate, Phenyl Trimethicone, Dimethicone, But..."

Combination: 1

Dry: 0

Explain Plan

Explain provides key execution metrics that help diagnose slow queries and optimize index usage. [Learn more](#)

Visualize

Raw Output

COLLSCAN

Returned 10

Execution Time 1.4

Documents Examined 38768

Query Performance Summary

10 documents returned

38768 documents examined

16 ms execution time

Not stored in memory

8 index keys examined

No index available for this query.

This query retrieves all cosmetics that are suitable for people with Sensitive skin. In this case, the Sensitive field is a binary value (1 for True, and 0 for False), and the query is filtering for products where Sensitive is 1 (True).

EcoMart can use this query to recommend skin products specifically for customers with sensitive skin. By segmenting products based on skin type, EcoMart can offer more accurate and helpful suggestions.

By recommending products that are suitable for sensitive skin, EcoMart improves the likelihood of repeat purchases and builds trust with consumers who are concerned about their specific skin care needs.



## 4. Apply optimization techniques

0897\_Task2 > cosmetics

Documents 1.5K Aggregations Schema **Indexes 1** Validation

Create Index Refresh

Name and Definition	Type	Size	Usage	Properties
> _id	REGULAR	26.7 KB	6 (since Sun Oct 06 2024)	UNIQUE
> Sensitive_1	REGULAR	24.6 KB	0 (since Sun Oct 06 2024)	
> Rank_1	REGULAR	26.7 KB	0 (since Sun Oct 06 2024)	

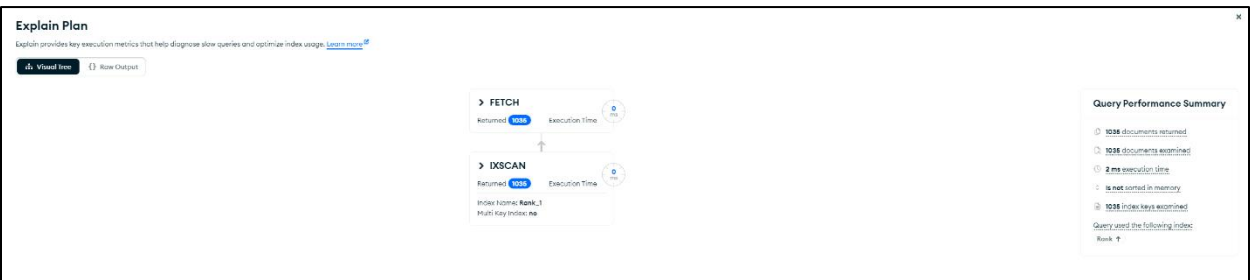
0897\_Task2 > groceries

Documents 38.8K Aggregations Schema **Indexes 1** Validation

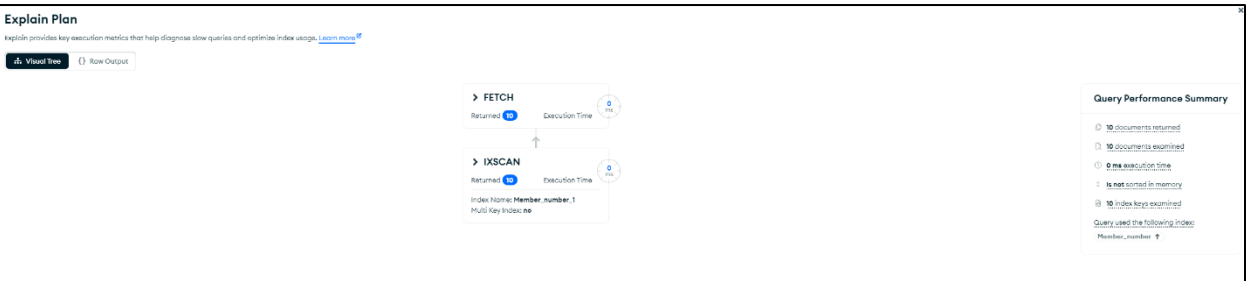
Create Index Refresh

Name and Definition	Type	Size	Usage	Properties
> _id	REGULAR	401.4 KB	7 (since Sun Oct 06 2024)	UNIQUE
> Member_number_1	REGULAR	241.7 KB	0 (since Sun Oct 06 2024)	

- Optimized Query 1



- Optimized Query 2



- Optimized Query 3

