Luis Medina

ID: 012020203

Data Management – D597

Task 2: Non-relational Database Design and Implementation

Part 1: Design Document

A1. Business Problem

The cosmetics industry is facing a challenging problem right now with customers with varying skin types, concerns, and budgets, making it difficult to provide personal product recommendations. In addition to the vast majority of products available to customers, it’s easy for them to feel overwhelmed and often buy products that are not for their needs. This will lead to the customers being dissatisfied with their purchases and will cause a return and reduce customer loyalty. Developing a personal database for business that can help customers with products based on criteria can lead to great customer satisfaction.

A2. Database Justification

A NoSQL is the ideal solution for the business problem from the previous section and here are several reasons why. Since the data that will be included will vary between different ingredient lists and skin-type compatibility flags, we need a database to store the documents that is flexible with data types. By looking at the JSON files I can see that the data has nested fields, so the perfect way to go about this is with a document-based model like MongoBD. Lastly, using a NoSQL database will be scalable, so we will be able to filter our search products easily without performance degradation.

A3. Database Type

The best type of NoSQL database to use for this type of data provided is going to be a document-oriented database, such as MongoDB. Using this would easily store data in JSON format, making it perfect for the data we have. MongoDB, would easily be able to handle nested fields like ingredients or skin-type compatibility, making it another reason to use this type of database.

A4. Data Usage

Here I am going to describe how the business will make use of the data. First, it will be with personalized recommendations, meaning that when a user selects their skins type, the database can query and return the products that best suit their skin type. Another usage with the data is that they will have an easy way to filter data, that way customers can search or filter by brand, price range, product rank, or ingredients. Then, it can be used for analytics for business to better understand trends in customers. By understanding trends in customers, it will help with inventory systems as they won’t have to worry about running out of popular products because they will know what to order.

B. Scalability

Next, for scalability, which will be a critical part of the database, as it grows there will be more demand, and we would not want any problems to come up. With more customers, more product data, and more queries, we want to be able to handle all this, so it doesn’t slow anything down. First, its horizontal scaling, MongoDB supports partitioning data across multiple servers. This is helpful because as more product is added into the database, then it will grow horizontally instead of vertically. Next, I will use indexing on fields that are frequently used in search queries. This will reduce the query latency, and this will be important when the database increases in records, as the company will get new customers and new products. This will also help in the future if the company requires any data analysis done on the data. Another useful quality of MongoDB is that it allows the document to have different set of attributes. Meaning that if we need to add new products or categories, we can do so without redesigning the entire schema.

C. Privacy and Security

Here we have one of the most important parts when designing and implementing a schema. It is super important to protect customer and product data, therefore there are going to be some measures that must be followed. First, not everyone has access to the database, I will use authorized users to give certain people certain access to the database. For example, an admin would have the most access compared to an analyst who would have access to pull data. This way only authorized people will be able to access the sensitive data. Next, I will use encryption to protect the data from being intercepted or leaked while in storage or transmission. Lastly, I will create a schedule for regular backups of the database to make sure nothing gets lost. There will be tested to check the back up data is up and correct, to minimize the risk of losing data in case of a hardware failure or accidental deletions.

D1. Database Instance