DATA70141 Assignment-2

Individual Report

Student ID : 11356488

PART - 1

A. Post group formation and initial group meeting , the Amazone project was strategically divided amongst the group members. The members worked on different parts of the project simultaneously before we consolidated our findings. To begin with this section, I will discuss my individual contributions to the project and key outcomes of my efforts.

1. My primary task involved understanding the problem statement and designing the NOSQL schema for 'Amazone'. Database design task involved me going through the given schema, analysing the collections based on additional requirements such as adding a 'Partners' collection to assign delivery partners for fresh category orders based on the partner location. Overall schema creation was not finished in 1 attempt, but several modifications were made by me to accommodate the various requirements that the database and our background python scripts required. One such schema modification that was added by me was a 'Payments' collection to capture all payment details.
2. After designing the NoSQL schema, I was responsible for creating the Entity relationship diagram and for generating the diagram I used an online tool LucidChart [1].
3. In the Database populating stage, our group decided to automate the process by utilising custom python scripts. I was assigned to create and populate 'customers', 'stores' and 'warehouses' item collections with Manchester-specific details. I wrote simple python scripts integrating with MongoDB, which made the insertion process efficient ensuring that it met all necessary minimum requirements.
4. I developed specific features for updating different collections. I created a function for calculating the average rating for each product and updating them in the 'products' collection. To perform this, I built a python script which utilises the MongoDB aggregation pipeline to group ratings by productID and then calculate the average rating for each product from the 'ratings' collection.
5. Additionally, my task was to develop the code which would generate product recommendations for each user. I designed the code with a strategy where the items a customer has not purchased are sorted based on average rating and the top-rated unrated product item is recommended to the customer. This recommendation is updated in the 'Customer' collection.
6. Developed a query representing a customer ordering a product, adding it to the cart and making a payment.

7. Developed a managerial query to visualise the sales performance. The python code utilises the past orders to find the total sales price on each day. The performance graph is visualised using pandas and matplotlib. Maximum sales for each fresh and other category.

B. This section talks about my project learnings and challenges that I faced during the project stages:

Learnings gained :

- The project provided an excellent opportunity to delve into the NoSQL database system, offering hands-on experience with tools like MongoDB Compass.
- The project facilitated a comprehensive understanding of the design process for a NoSQL schema, demonstrating how to effectively tailor the schema to accommodate an evolving business model and new requirements and iterating through the documents efficiently.
- A key learning through the project was understanding the integration of database techniques with python using the ‘pymongo’ library that connects mongoDB terminal with our python code. This provided a fresh perspective on dealing with real-world requirements.
- Learning effective project management and task prioritisation emerged as significant lessons from the project.
- Collaborating within a team was essential in honing teamwork skills which is crucial in the tech industry. Regular team meetings helped in effective communication and exchanging ideas which generated strategies to design the database in the most efficient manner. Responsibilities like dividing tasks,engaging in team discussions to explore potential methodologies and providing assistance to teammates during challenges played a crucial role in fostering personal and professional growth.

Challenges faced :

- Coming from an SQL background, adapting to NoSQL techniques, the syntax of queries using MongoDB Compass was an initial challenge.
- Data modelling was being done on an evolving schema and hence it was complex to manage an evolving schema with collections like customers, partners,payments etc. especially as business requirements evolved. This required careful planning to avoid data migration issues.
- A specific challenge arose when we were working on generating product recommendations for customers. Initial approach was to create a mechanism similar to trigger, which would automatically calculate and store the recommendations as an array in the customer collection. However, due to technical difficulties and lack of trigger-like functionality in MongoDB, we had to revise our approach and opt for a python-based function. This function reads data from MongoDB and then applies the logic to generate customer specific product recommendations.

- We faced a few challenges to combine our individual findings and reach to one common design. This was an initial team coordination issue, which improved with time.

PART - 2

A. To expand operations, the company has decided to set up an additional data centre in Europe. We must consider various factors like data synchronisation, data volume, data availability, disaster recovery and business expectations to ensure cost-effectiveness before we can decide on a replication strategy.

   According to my understanding, the ideal replication strategy would be a **"partially replicated distributed system"**. This strategy would be especially useful as we are dealing with huge volumes of data including millions of customers and products and more to come as the business expands in the Europe region. Fully replicated dataset is not the right approach as it will lead to slower writing process on the database and increase query lags. Also, as mentioned in the project brief, some of the collections are not getting queried much like the past orders data, so storing these data in specific data centres would be more realistic.

   On the other hand, some features associated with average customer rating and product recommendations need to be queried and updated frequently, here non-replicated systems are not feasible. This is because in case of a node failure, it would lead to an increase in reading time consequently affecting the duration of background calculations. Therefore, an optimal solution would involve adopting a partially replicated database system using sharding techniques which will help us to maintain data availability, wherein the frequently queried collections are replicated on each data centre , while the least frequently accessed collections are stored in designated data centres UK or Europe in our case.

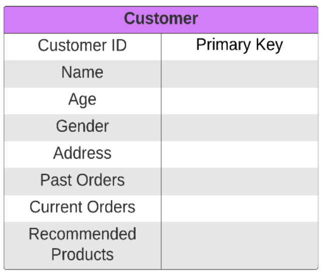
   Apache Kafka, a third party integration for the replication process can be very beneficial.It connects to mongodb using its native connector namely, "MongoDB connector for Apache Kafka." Through this connection customers can stream real-time data between the two data centres. For advanced analytics and replicating data we can also integrate Apache Hadoop and Apache Spark.

B. This part will discuss the suggested partition and allocation strategy based on the same factors discussed above.

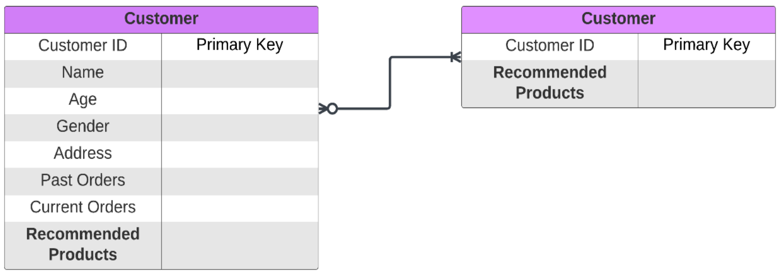
   Starting with partition, based on our business scenario both vertical and horizontal partitioning techniques would be beneficial. Considering the 2 data centres in the UK and Europe, a suggested method would be to horizontally partition and store the customer data in their specific region only. Another collection that can be partitioned horizontally is past orders (if exists) as they are rarely queried; they can be stored in their specific regions. However, the 'products' collection will not follow this because we assume that a

UK customer can order European products , hence product horizontal partitioning would be infeasible.

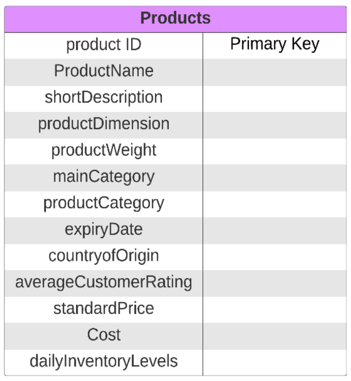
Now, another method that can be employed is vertical partitioning which involves separating features of each collection. For instance, average ratings and recommendations require constant updates and need to be readily available. To update ratings, data from the Orders and Products item collection is required, these details are rarely queried. Therefore a strategic vertical partition can be made to isolate only the items and their corresponding ratings in a new collection, separating them from less important features. Similarly, vertical partitions can be applied to Customers, Warehouse, partners and payments collections to optimise background processes.



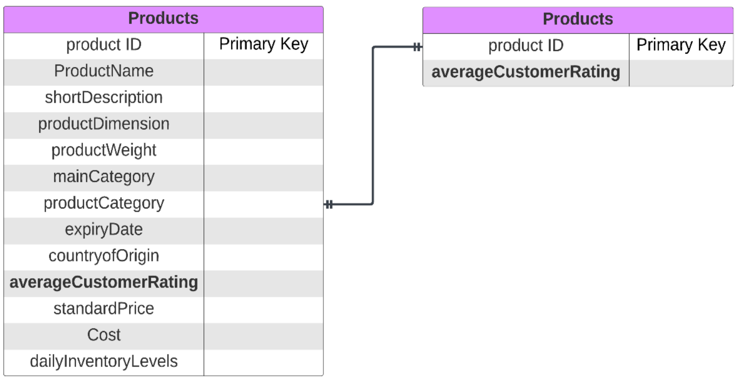
Customers schema before partitioning



Proposed schema after partitioning



Products schema before partitioning



Proposed schema after partitioning

For allocation strategy, we will consider sharding which means data distribution across regions. For sharding, we need to prioritise which data collections are most important and frequently accessed for a smooth business process. For example, data related to products and their associated ratings , as well as their average product ratings should undergo fully replication. On the other hand, customer data , warehouse and store data specific to regions like the UK or Europe, should remain non-replicated and present in data centres dedicated to their region.

C. According to me, the expansion of business in Europe requires database design modification. Most of the changes will be because we want to incorporate data partitioning and allocation strategies for efficient real-time data processing and high business operations performance. Some changes will happen to also adapt and include replication strategies to ensure disaster recovery.

   Changes made due to partitioning will require rearrangement of features. Notable modifications can be observed in the structure of Customers, Products and Order items collections. Below is an example of the changes made in Orders collection and products collection, emphasising the impact on the schema

   Finally, some changes will be due to applied replication strategies which happens because we are setting up a new data centre in Europe. Collections such as Customers, Warehouse, grocery stores and certain region-specific products will require the inclusion of a “location” feature. This feature would include geographic distinction between UK and Europe specific data items.

References

1. https://www.lucidchart.com/
2. https://pypi.org/project/pymongo/
3. https://docs.mongodb.com/drivers/pymongo/
4. https://matplotlib.org/
5. https://pandas.pydata.org/