

FINAL PROJECT

Non-SQL Database

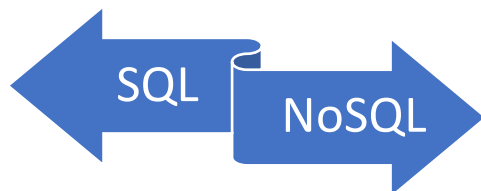
By ALLEN RICHARDS

Introduction

Every business could easily recognize the future predictions only with database especially historical one. Once we learn to understand the pattern or relationship between one data to another then we can clear the difficulties in the business forecasting process. Likewise in this study I investigate whether the non-relational database MongoDB is appropriate for **Macy's, Inc.**, a well-known company in USA. It tackles Macy's data management needs, emphasizing the difficulties brought on by dynamic e-commerce data. The scalability, flexibility, and real-time processing of MongoDB make it a preferred option for solution. To manage Macy's data, the paper presents a MongoDB prototype. It shows how effective this system is at handling large amounts of unstructured data, which makes it useful for businesses with comparable needs and to fulfil the requirements of business.

Business Overview

Macy's is a prominent online retailer with over around 42 million customer records. Primarily functioning as an e-commerce platform, Macy's acts as a middleman between suppliers and customers, allowing the former to sell their goods and the latter to browse and buy the same item from multiple vendors. Macy's has attracted a sizable number of online shoppers because of its affordability, ease of use, and regular sales. Because of this, companies such as Macy's need a strong platform that can store and manage a big and varied dataset that includes user data, transaction history, and real-time analytics. Only with such a system can companies run labour and time efficiently.

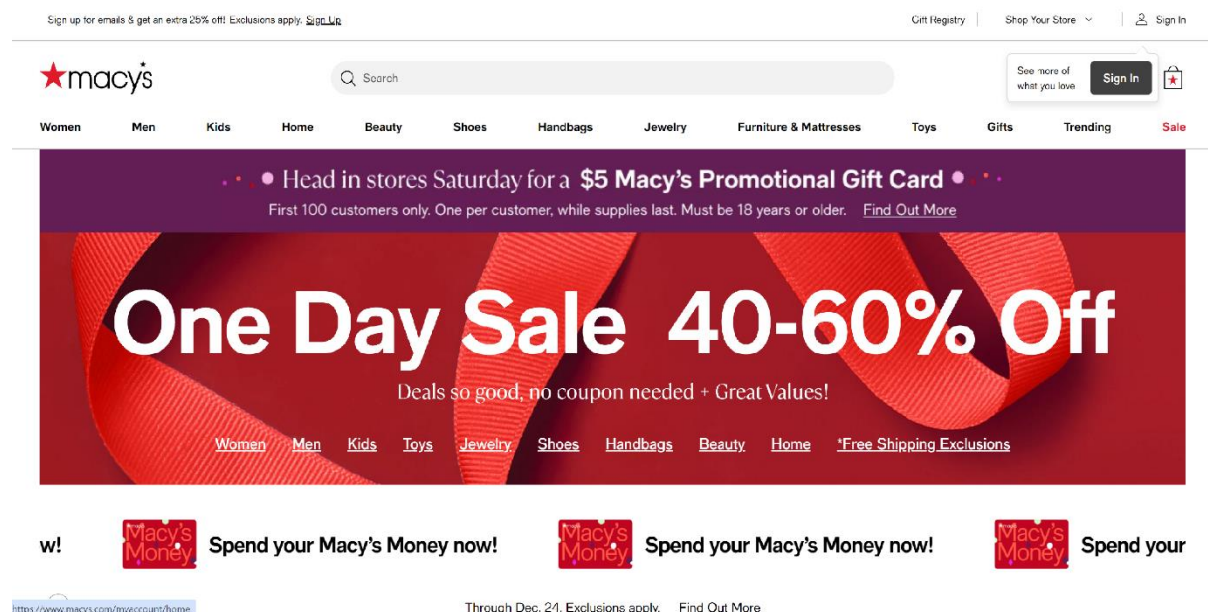


The Macy's business model of today is not what it first appears to be. The truth is that Macy's founded in 1858 by Rowland H and is headquartered in New York, NY. Since it is a very old company but equipped with advanced technology like online portals, website etc. There are many competitor companies struggling hard to improve their business I considered Macy's one of the best e-commerce companies which deals with bunch of customers.

Reason for considering NoSQL DB

- Macy's is a giant firm with millions of customers data. When it comes to managing a particular scalability, flexibility, and performance requirements, a NoSQL database can provide several benefits.
- It can easily scale out by adding more servers or nodes to handle increased data and traffic because they are made for horizontal scalability. Considering the volume of data produced by its e-commerce platform and other services, this is especially crucial for Macy's.
- Also, it enables a flexible schema, which eliminates the need for a preset schema and makes it simpler to manage a variety of dynamic data structures. This adaptability helps when Macy's rolls out new product categories or upgrades old features.
- For managing unstructured or semi-structured data, NoSQL databases like MongoDB provide essential flexibility, meeting Macy's dynamic environment.

This company generalized their online platform like others. Also, they are maintaining each customer profile and suggesting products as per their "Search". They use association rules and collaborating filter.



The "Sign IN" option helps to access customers profile and they gather data by this feature. It depends on in-the-moment user behaviour, product popularity, and market trends analysis. NoSQL databases are excellent at real-time analytics, giving Macy's the means to quickly derive useful information. This feature enables Macy's to optimize offerings, make data-driven decisions, and improve the user experience.

Macy's - NoSQL (Solution)

- MongoDB is a simpler way to get and manipulate data.
- It helps to intricate data retrieval requirements such as user profiles, product histories and transaction entries.
- It helps to evaluate customer's behaviour, spot best-selling items, and quickly respond to industry developments.
- It is document-oriented architecture analytics processing could be completed more quickly by strong and retrieving data.

Let's try Barbone Prototype Implementation

In implementation process we need to take various steps like organizing or categorize the data. Each company products are defined in various groups. For examples: clothes, electronics and accessories. There are main categories which plays a major role

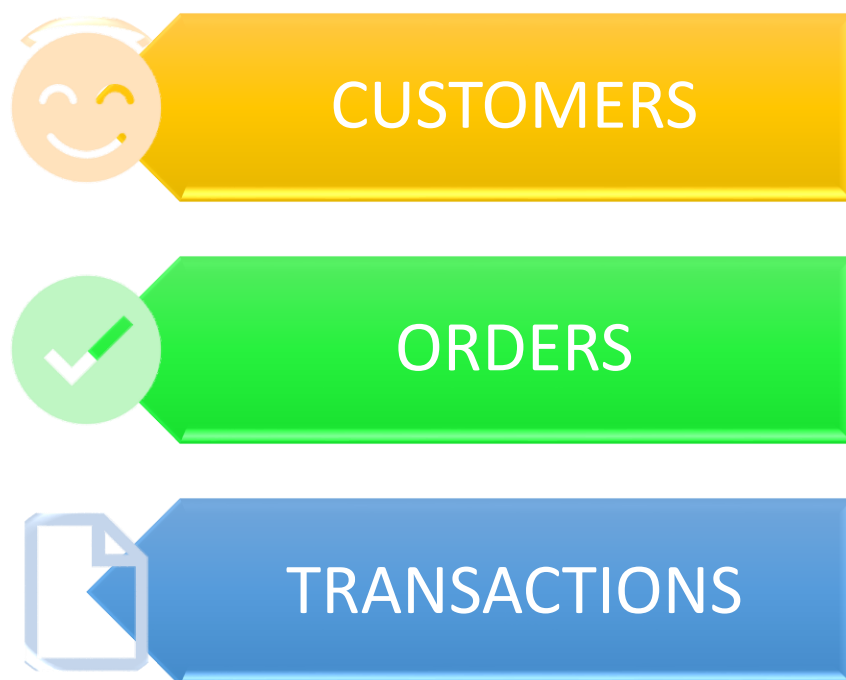


Fig 1.1

Every E-commerce company prioritizes these 3 parts likewise we need to define the data in this way. Each part connects different levels.

Let's Elaborate little further.

Customer: -

- NAME
- CUSTOMER ID (UNIQUE)
- ADDRESS
- PHONE NUMBER
- EMAIL

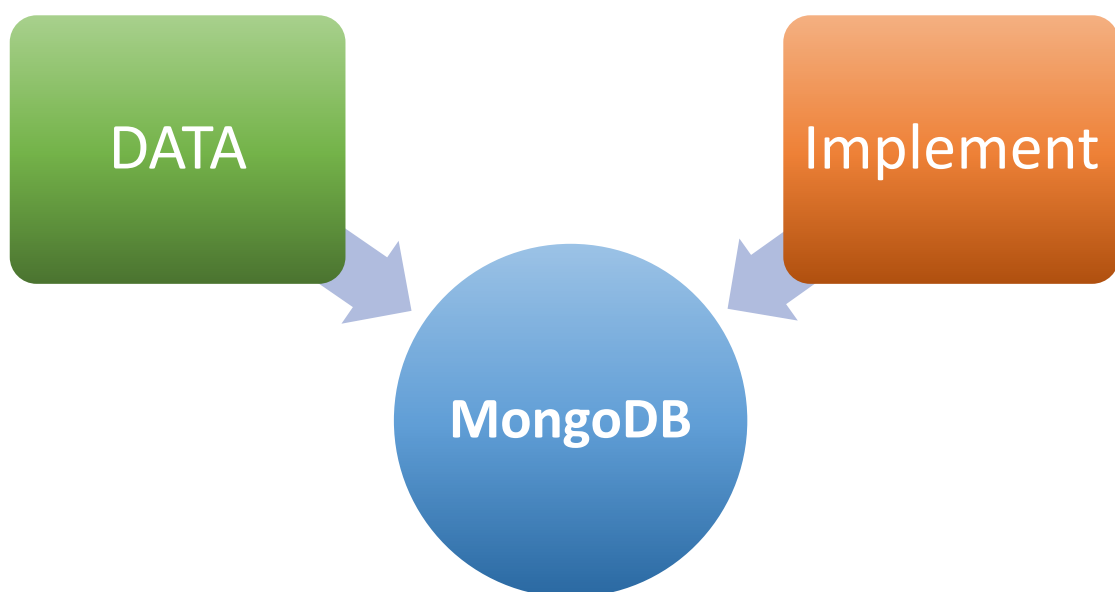
Orders: -

- PRODUCT NAME
- SUPPLIER ID
- MANUFACTURER NAME
- DESCRIPTION
- PRICE

Transactions: -

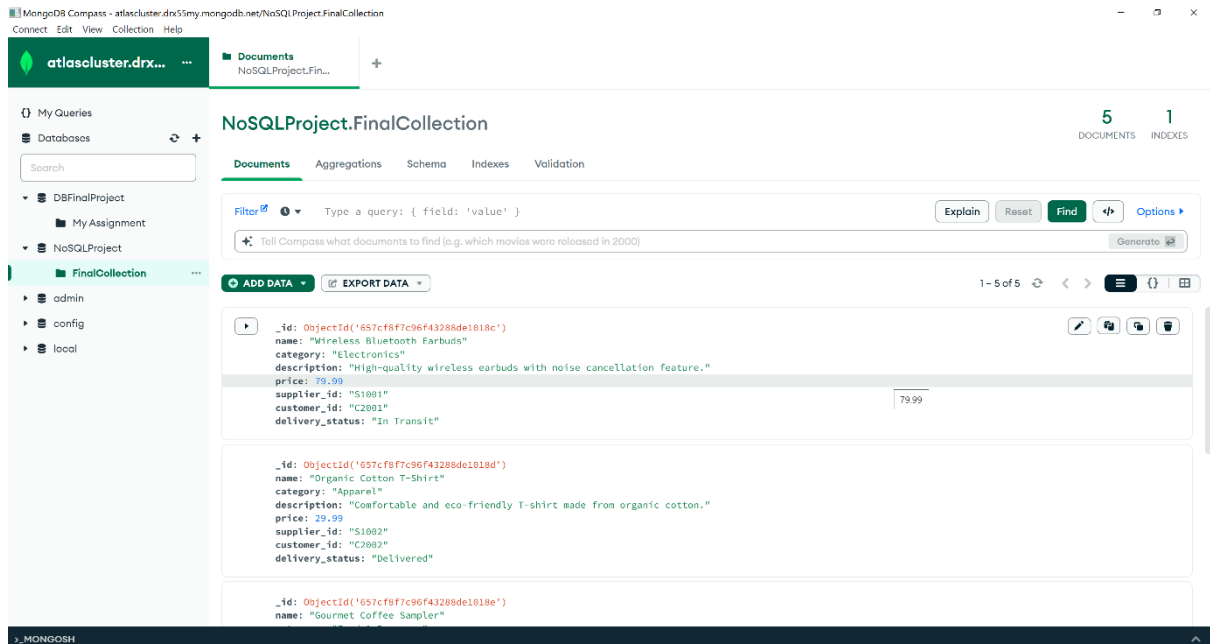
- PAYMENT METHODS
- DELIVERY STATUS
- TRANSACTION ID / NUMBER
- REFERENCE NUMBER
- BILL VALIDITY

These above details are essential to complete a 'SALE' but these are the important information we have to define in data before we insert into MongoDB.

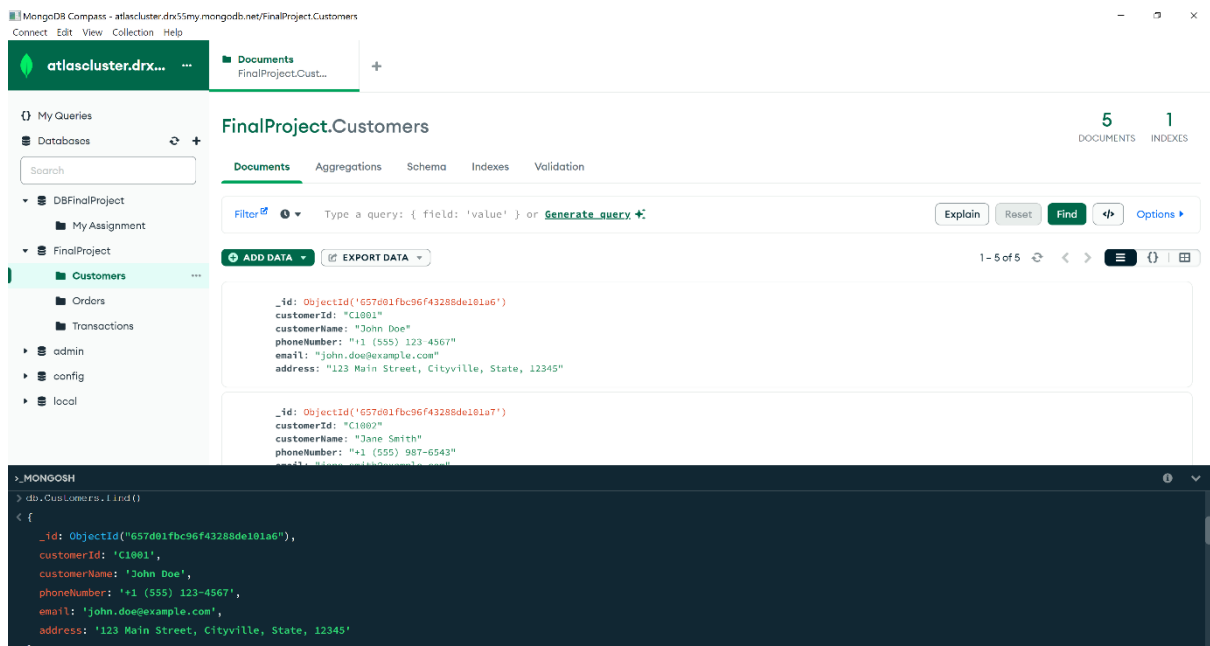


Now Adding data into MongoDB

As new database “NoSQLProject” with new folder “FinalCollection”. A new data from Macy’s with different headings.



The data was inserted into MongoDB now we can manipulate by “>_MONGOSH”



In this above screenshot the code we used “db.Customers.find()”

Now it extracts all information which mentioned in “Customers”

The next step is to extract like same but “db.Orders.find()”

The screenshot shows the MongoDB Compass interface for the 'FinalProject.Customers' collection. The left sidebar shows the database structure with 'Customers' selected. The main area displays a list of documents. Below the list, a MONGODB terminal window shows the command 'db.Orders.find()' and its output, which includes product details like 'Classic Denim Jacket'.

The 3rd step is to extract the last folder by “db.Transactions.find()”

The screenshot shows the MongoDB Compass interface for the 'FinalProject.Customers' collection. The left sidebar shows the database structure with 'Customers' selected. The main area displays a list of documents. Below the list, a MONGODB terminal window shows the command 'db.Transactions.find()' and its output, which includes transaction details like 'orderId', 'customerId', and 'products'.

Now all the categories were merged and shows the connectivity of each information. So we can now easily identify For ex: status of delivery and payment.

The first document that meets the given criteria in the Transaction collection of the current database will be retrieved by the command also other command predicts something which is related to that

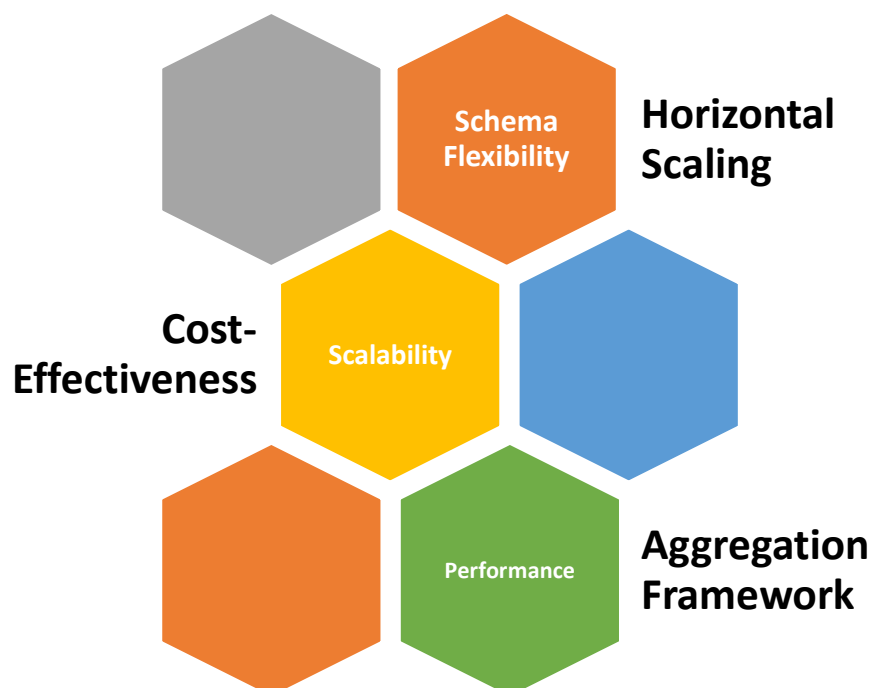
```
"db.Transactions.findOne({"deliveryStatus":"Shipped"})." &  
"db.Orders.findOne({"Price":"100"})
```

Implement Explanation

Since, the customers, orders, and transactions have the relationship between the products. The code “db.Customers.find” extracts the details of users and the second code “db.Orders.find” and the last one “db.Transaction.findone” finds the status of product. I consider this advantage of MongoDB which best suit for these kinds of data. While NoSQL databases are perfect fit for some hypothetical business problem but could not guarantee that it solves all the time.

In the MongoDB, I consider this as best solution for the problem and I mentioned in this implementation, I extracted the function which comes in MongoDB. Sample data was collected from Kaggle and Github also modified with some fake information. I have imported that data into the MongoDB compass and saved as text documents for reference. Later this data might be used on various devices such as smartphones, tablets etc.

Some advantages in MongoDB



Conclusion

In my opinion, the expanding e-commerce sector necessitates skilled data management, and NoSQL databases like MongoDB are essential to this flexibility and effectiveness. The key commands in MongoDB are essential for storing and retrieving data. The **Macy's** scenario highlights how crucial database management is to large e-commerce platforms, and as technology develops, we expect more systems to come online to meet changing needs. The rapid growth of technology we could improvement of development in DATABASE systems.

Reference

DATA: <https://github.com/bigbite/macy.js/blob/master/package.json>

(Modified according to the essay requirements with the help of Database Illuminated Chapter 14 Data Organization in MongoDB)

BOOK = 4th Edition - **DATABASE ILLUMINATED**

Author: **Catherine M. Ricardo, Susan D. Urban, Karen C. Davis.**