**Executive Summary**

**Unit 11 – End of Module Assignment**

**Course:** Deciphering Big Data July 2024

**Instructor:** Dr Ali Zazala

**Date:** 14th October 2024

Contents

[Introduction 3](#_Toc179798520)

[Critical Evaluation of the Database Model 4](#_Toc179798521)

[SQL and No-SQL Options 6](#_Toc179798522)

[Compliance requirements 8](#_Toc179798523)

[Recommendations 10](#_Toc179798524)

[Conclusion 12](#_Toc179798525)

[References 13](#_Toc179798526)

# Introduction

The discussion in this report will provide the key points that led the team to the design of the online bookstore inventory database management system. The first section summarises the work of the report of the database model made. Furthermore, the database model is evaluated, and critical points, mainly normalisation, are evaluated further. In addition, hosting on the cloud will be discussed.

The second section will evaluate and discuss the choice of SQL or NoSQL database management system. In this section, the discussion will revolve around the advantages and disadvantages of both database systems, the flexibility of a NoSQL database, and the efficiency of a SQL database (Bjeladinovic, Marjanovic & Babarogic, 2020). Both systems have a part to play in creating an ideal database management system with the right features to increase revenue streams for the online bookstore.

The third section discusses compliance. Compliance mainly involves GDPR and other related data laws that enhance the right to privacy and personal data protection (Shastri, et al., 2020). The discussion goes through how the database system must comply with these laws to protect both the personal rights of individuals and protect the organisation from fines and lawsuits.

The last section will give recommendations that could add value to the online inventory database management system. Specifically, recommender systems which can increase revenue by making it easier for users to find likeable products (Mazeh & Shmueli, 2020). This section will include the different compliance requirements of recommender systems along with the database design changes that they involve.

# Critical Evaluation of the Database Model

The database model that was designed for an online bookstore's inventory management system started by considering the user requirements: inventory updates after purchases and sales of books, search and analytics by features like ISBNs, publisher, and genre, amongst others, analytics on inventory turnover and sales trends, and accounting software integration for revenue management, cost management, and inventory valuation (Zapka, Singh, & Caruana, 2024).

IncomingOrderItem 
ISBN (PK) (varchar (255)) 
Purchase ID (FK) (integer (10)) 
Supplier ID (FK) (Integer (10) 
Quantity (Integer (10)) 
Supplier 
Supplier ID (PK) (integer (10)) 
Name (varchar (255)) 
Supplier contact narne (varchar (255)) 
Address (varchar (255)) 
Email (varchar (255)) 
Phone (unique) (varchar) 
Website (varchar (255)) 
IncomingOrder 
Purchase ID (PK) (integer (10)) 
Supplier ID (FK) (integer (10)) 
ISBN (FK) (varchar (255)) 
Date placed (datetime) 
Purchase status (varchar (255)) 
Outgoingorder 
Order ID (PK) (integer (10)) 
ISBN (FK) (varchar (255)) 
Customer ID (FK) (Integer (10)) 
Date placed (datetime) 
Shipping address (varchar (255)) 
Order status (varchar (255)) 
ISBN (PR) (varchar (255)) 
Title (varchar (255)) 
Author (varchar (255)) 
Publisher (varchar (255)) 
Publication Date (datetime) 
Genre (varchar (255)) 
Language (varchar (255)) 
Price (numeric (19. O)) 
Quantity in stock (integer (10) 
Ouantity on order (integer (10) 
ISBN (PK) (varchar (255)) 
Order ID (FK) (integer (10)) 
Customer D (FK) (Integer 
(10) 
Quantity (Integer (10)) 
Customer D (PK) (integer (10)) 
Name (varchar (255)) 
Address (varchar (255)) 
Email (varchar (255)) 
Phone (unique) (varchar) 
Password (varchar (255)) 

Figure 1: ERD of the proposed database design (Zapka, Singh, & Caruana, 2024)

Adding to the user requirements, Figure 1 details the inventory database management system's Entity Relationship Diagram (ERD), created to identify the database elements and relationships. Furthermore, according to Figure 1, the database model has one-to-many relationships between the following entities: Supplier to IncomingOrder, Customer to OutgoingOrder, Book to IncomingOrderItem, Book to OutgoingOrderItem, IncomingOrder to IncomingOrderItem, and OutgoingOrder to OutgoingOrderItem. The database model also has many-to-many relationships between the following entities: Customer to Book and Supplier to Book (Zapka, Singh, & Caruana, 2024).

The organisation and efficiency of this ERD diagram will reduce latency and duplicate storage. Wong Ting Yan and Fong (2021) state that normalisation eliminates inefficiencies that will improve data redundancy and storage costs. Database normalisation is essential in having an efficient database that will reduce inconsistencies. As an example, if the database is normalised correctly when a supplier address needs to be changed, this will only happen in the supplier table, not in each order the supplier has. Furthermore, database normalisation comes in forms. In this design, the normalisation takes place for practicality until the third normal form. Other forms exist, but up to third-form database normalisation is enough for the proper functionality. To comply with the first normal form, each set of related data must have a primary key, which must be unique for each row. Each set of data must be in a separate table, and each record of data must not have more than one entry. For example, each OutgoingOrder row cannot have more than one Customer ID. Furthermore, the database model is also in second normal form, which implies that a separate table must be created for data that appear in different tables. For example, the supplier address in IncomingOrder must be in a separate Supplier table and linked with a foreign key. The third normal form states that any key that does not depend on the primary key should be kept in a separate table. This overview of database normalisation gives further information on the database model’s efficient design according to client requirements (Microsoft, 2024).

The hosting platform, cloud or private network, is another vital point in our data architecture. In recent years, cloud computing has proliferated among businesses that use this technology for its flexibility and scalability. There are several advantages of cloud computing, including reduced cost due to less capital investment, reduced cost due to the ‘use what you need’ concept, and scalability, which is essential in the case of an online bookstore. On days when sales increase drastically, cloud computing has the resources to increase storage and processing capacity as needed (Maresova, Sobeslav & Krejcar, 2017).

# SQL and No-SQL Options

Another vital step in designing the right online database management system is choosing the database type. SQL was the right fit for the client's requirements in the online bookstore case. Having said this, NoSQL database management can add significant advantages to an online bookstore by increasing essential features that could increase revenue streams for the organisation. The advantage of an SQL database is managing data in an organised, standardised and efficient manner. When it comes to querying, SQL has an advantage when complex joins are needed, as it is very efficient in these types of queries because of its structure. On the other hand, different NoSQL databases do not have a standardised language. Therefore, the integrity of data in NoSQL can be compromised due to the unstandardised structure (Bjeladinovic, Marjanovic & Babarogic, 2020).

Data consistency is an important element in our database type. SQL’s advantage is aligned with this, hence why it is ideal for online transaction processing (OLTP) systems. The SQL database type adheres to the four principles of atomism, consistency, isolation, and durability, more formerly known as ACID properties.

On the other hand, NoSQL is ideal for big data analytics processes. In this case, NoSQL does not adhere to ACID principles but to CAP. CAP stands for consistency, availability, and partition tolerance; the theory suggests that only two of these principles can be adhered to by the same database in a NoSQL database. NoSQL databases are ideal for scalability, while SQL databases are limited when it comes to the scalability feature (Khan et al., 2023).

As discussed in the previous section, cloud computing has several advantages that make it the right choice for hosting the online bookstore management system. Furthermore, the cloud can host the SQL  and NoSQL databases if any changes to the structure of the inventory management system are needed. In the recommendations section, a recommender system will be discussed in which a NoSQL database is a better solution. In this case, a hybrid architecture can satisfy both storing and processing unstructured and structured data requirements.

A screenshot of a computer

Description automatically generated

Figure 2 SQL and NoSQL structure (Shareef, Shareef & Rashid, 2022)

Figure 2 shows the structures of SQL and NoSQL databases, which take four forms: column, key-value, document, and graph. Amazon Redshift is an example of a column NoSQL database that uses columns instead of rows to store data. Redis is an example of a key-value database whose storage method is self-explanatory. In addition, graph databases use nodes and connections with their entities to store data. An example is AllegroGraph. The last structure is the document database structure, which mainly stores data in JSON format. An example is MongoDB (Shareef, Shareef & Rashid, 2022).

# Compliance requirements

Compliance in our inventory database management system is a crucial part that starts from the design and continues in each phase of the database system. Data compliance mainly begins with compliance with regulations like GDPR, which protects the right to privacy and security of individuals' data. Having said this, GDPR compliance starts from the database model and data architecture design. Principles like data processing purpose declaration, minimal data capture, consent for data collection, and anonymisation of private data must all be incorporated into the database design (Cormack, 2021).

In the online bookstore database management system, client personal and payment details are the data at the highest privacy risk. Measures to anonymise this data, along with suitable storage barriers, should keep this data safe from cyber-attacks and mitigate sensitive data breaches that might lead to fines and court cases for the organisation. Furthermore, to align with the GDPR requirements, an organisation must store metadata and private data separately from other data. This requirement might increase the workload and processing requirements for the database management system. Therefore, these requirements must be accounted for when deciding on the processing power of your database management system (Shastri et al., 2020).

A screenshot of a computer

Description automatically generated

Figure 3 Requirements of key GDPR articles into database system attributes and actions (Shastri et al., 2020)

Figure 3 gives an overview of each GDPR article or clause that will affect or impact the database design or model. These requirements might impact storage and cost, hindering standard database optimisation goals of efficiency and cost reduction. Having said this, an organisation must comply with these requirements today, as hefty fines can be given for not complying with GDPR (Shastri et al., 2020).

Security design is a relevant issue in the case of security in NoSQL databases, as security was not considered initially in the NoSQL database design. However, security has become more of a concern lately, especially in the latest versions of MongoDB, where security features have been added. Lately, organisations are tackling security issues at the design stage to mitigate security issues later on in the stages of the database management systems (Blanco et al., 2022).

# Recommendations

A screenshot of a computer

Description automatically generated

Figure 4 Recommender system building methods (Onokoy & Lavendels, 2019)

In addition to the user requirements, a suggested recommender system can allow the online bookstore to add another feature that could help increase revenue by suggesting similar books that clients might purchase. There are several types of recommender systems. Figure 4 gives an overview of the kinds of modern recommender systems. In the bookstore case, collaborative filtering methods would be ideal as they are built to recommend similar purchases to those other buyers have made previously. Memory-based methods give recommendations from the accumulated data that is stored by the system. The choice further aligns depending on whether the user’s data is user-based and product data is item-based or a combination of both. The recommendation will be decided on the system that is used. The accuracy and simplicity of this system, in comparison to others, is an advantage. On the other hand, this system might use more storage, and the forecasts regarding new products are inaccurate (Onokoy & Lavendels, 2019).

A diagram of a process

Description automatically generated

Figure 5 Lambda Architecture (Sanla & Numnonda, 2019)

A white rectangular box with black text

Description automatically generated

Figure 6 Kappa Architecture (Sanla & Numnonda, 2019)

Recommender systems need a different database type than SQL, which leads us to the NoSQL database. Having said this, the architecture of the cloud will also need to be different. For these instances, two kinds of architectures are evaluated for this situation: Lambda Architecture, as shown in Figure 5, and Kappa Architecture, as shown in Figure 6. The Lambda architecture consists of three different layers: the batch layer that sends data in batches, which is ideal for SQL databases, and the speed layer, which is the layer that streams data where NoSQL databases are ideal. The combined data views can be explored further in the serving layer, where the merged layers can be analysed. In the Kappa architecture, there is only a streaming and serving layer. The cost of the Lambda architecture comes from the maintenance of all layers, but its high accuracy is a benefit of this type of architecture. Flexibility is the positive aspect of these architectures, where we can combine types of databases to extract the most benefit for our online bookstore database. However, we must keep the budget in mind as Lambda architecture can cost double Kappa architecture (Sanla & Numnonda, 2019).

# Conclusion

This section concludes this report. From the requirements analysis, this report gives an overview of what is needed and requested to satisfy the client. In addition to satisfying the client’s requirement, a recommended recommender system was added to help the client increase revenue by using the system's accumulated data. This difference from the client’s initial requirements requires us to change the database type and architecture. Despite this, there are several benefits of recommender systems, from providing service to customers by automatically recommending products for them to the reduction in cost of labour due to the proactive selling of products by machine learning (Onokoy & Lavendels, 2019).

In conclusion, this report has addressed the client's requirements with a robust database management system and architecture that is ready for growth and flexible for any required changes.

# References

Blanco, C. et al. (2022) Security policies by design in NoSQL document databases. Journal of Information Security and Applications, 65: 103120-. DOI: <https://doi.org/10.1016/j.jisa.2022.103120>

Bjeladinovic, S., Marjanovic, Z. & Babarogic, S. (2020) A proposal of architecture for integration and uniform use of hybrid SQL/NoSQL database components. The Journal of Systems and Software 168: 110633-. DOI: <https://doi.org/10.1016/j.jss.2020.110633>

Cormack, A. (2021) Thinking with GDPR: A guide to better system design. Information Services & Use 41(1–2): 61–69. DOI: <https://doi.org/10.3233/ISU-210107>

Khan, W., Kumar, T., Zhang, C., Raj, K., Roy, A. M. & Luo, B. (2023) SQL and NoSQL Database Software Architecture Performance Analysis and Assessments—A Systematic Literature Review. Big Data and Cognitive Computing 7(2): 97-. DOI: <https://doi.org/10.3390/bdcc7020097>

Maresova, P., Sobeslav, V. & Krejcar, O. (2017) Cost-benefit analysis - evaluation model of cloud computing deployment for use in companies. Applied Economics, 49(6): 521–533. DOI: <https://doi.org/10.1080/00036846.2016.1200188>

Mazeh, I. & Shmueli, E. (2020) A personal data store approach for recommender systems: enhancing privacy without sacrificing accuracy. Expert Systems with Applications 139: 112858-. DOI: <https://doi.org/10.1016/j.eswa.2019.112858>

Microsoft (2024) Description of the database normalization basics. Available from: <https://learn.microsoft.com/en-us/office/troubleshoot/access/database-normalization-description> [Accessed 11 October 2024]

Onokoy, L. & Lavendels, J. (2019) Modern Approaches to Building Recommender Systems for Online Stores. Applied Computer Systems 24(1): 18–24. DOI: <https://doi.org/10.2478/acss-2019-0003>

Sanla, A. & Numnonda, T. (2019) A Comparative Performance of Real-time Big Data Analytic Architectures. 2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC) 1–5. DOI: <https://doi.org/10.1109/ICEIEC.2019.8784580>

Shareef, T. H., Shareef, K. H. & Rashid, B. N. (2022) A Survey of Comparing Different Cloud Database Performance: SQL and NoSQL. Passer Journal of Basic and Applied Sciences 4(1): 45–57. DOI: <https://doi.org/10.24271/psr.2022.301247.1104>

Shastri, S., Banakar, V., Wasserman, M., Kumar, A. & Chidambaram, V. (2020) Understanding and benchmarking the impact of GDPR on database systems. Proceedings of the VLDB Endowment 13(7): 1064–1077. DOI: <https://doi.org/10.14778/3384345.3384354>

Wong Ting Yan, K. & Fong, J. S. P. (2021) Data Normalization. In *Information Systems Reengineering, Integration and Normalization.* Springer International Publishing AG. DOI: <https://doi.org/10.1007/978-3-030-79584-9_8>

Zapka, B., Singh, D. & Caruana, A. (2024) Development Team Project: Project Report *DBD July 2024.* Report submitted to the University of Essex Online.