Grenze International Journal of Engineering and Technology, Jan Issue

Data Modeling Practices for E-Commerce

Drushti Patel[1] and Monika Shah[2]
[1-2]Computer Science and Engineering Department, Institute of Technology, Nirma University, Ahmedabad, Gujarat, India
Email: drashti142000@gmail.com, monika.shah@nirmauni.ac.in

***Abstract*--In modern times, large amounts of data with diverse characteristics, such as volume, variety, etc., are generated daily. For enterprises such as E-commerce, which is escalating nowadays, massive amounts of unstructured data have increased significantly, making it essential to manage high data availability and consistency in conventional databases. To establish extreme data availability and consistency, opting for the most effective data model is necessary to ensure that data accuracy and consistency are achieved and can be easily accessed and manipulated by the intended users. The development of data modelling approaches has made it possible to use massive quantities of data without degrading performance or loosening the conventional ACID properties. This paper emphasizes the importance of efficient data modelling. It provides practical guidelines for structuring RDBMS data models and for NoSQL data models, considering the types of data access patterns and highlighting their advantages and limitations in terms of e-commerce enterprises.**

***Index Terms*— Data modeling, guidelines, Nosql, RDBMS, access pattern**

I. INTRODUCTION

The outbreak of interest from e-commerce enterprises in big data is increasing daily. Big data is classified into four categories: volume, variety, velocity, and veracity. E-commerce systems eventually generate big data, including massive volumes of customer data, products, categories, carts, orders, etc. E-commerce data is generated in real-time, meaning it must handle the velocity of the generated data. Also, it generates data of considerable variety, such as click stream data (b), image data (c), and video data (d), and such data can be incomplete, noisy, or inaccurate, and that's when veracity comes into the picture [1]. E-commerce sites and platforms face the challenge of maintaining consistent availability, performance, and throughput across all channels.
For an e-commerce application that generates and handles highly unstructured data with volume, variety, velocity, and variability, a superior database design becomes a must to maintain high data availability and consistency. For this reason, standard data modelling guidelines have become a must-have. Using relational databases proves to need improvement for such applications. In this context, NoSQL databases offer new storage solutions in large-scale environments, replacing many traditional database management systems. However, there are no clear boundaries between SQL and NoSQL, as it is becoming a trend that NoSQL databases can achieve strong consistency. At the same time, SQL has started to embrace some NoSQL characteristics.
This exploration highlights the importance of data modelling and its types. It shows a survey on the adaption of RDBMS/NoSQL data models in research precisely chosen for an e-commerce application. The content discusses data modelling guidelines used in RDBMS and NoSQL databases.
The rest of this paper is structured as follows: Section 2 shows the paper selection that was carried out through the survey for this paper. Section 3 briefs E-commerce life cycle and. Section 4 summarises data modelling

Grenze ID: 01.GIJET.10.1.174
© Grenze Scientific Society, 2024

Practices for E-commerce. Section 5 presents the concept of data modelling. Section 6 lists data modelling guidelines for relational databases. Section 7 lists data modelling guidelines for the NoSQL database. Section 8 lists challenges and scope for both RDBMS and NoSQL.

II. Research Paper Selecton

The method for choosing current research that was used for this survey is shown in Fig. 1. We began reading articles on data modelling for e-commerce and quickly saw that the data models varied greatly. Some people only use relational data models, while others use both relational and non-relational data models. Document store is a common NoSQL database for e-commerce. In relational databases and document-store databases, we have seen variances in database design. In order to compare with suggested E-commerce data models, we investigated relational and document store data modelling principles.

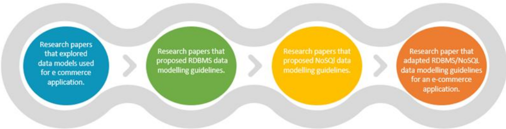

Fig. 1. Paper Selection Process

III. Background

A. Growth of the e-commerce industry in Big Data

According to [7], the mass of data in the digital universe will increase by 61 percent by 2025, reaching 175 zettabytes. Fig 2 shows the revenue growth of the E-commerce industries in India by 2026 [8].

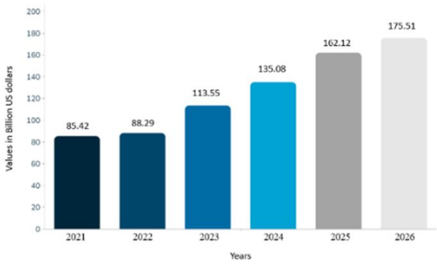

Fig. 2. Growth of e-commerce in India by 2026

B. E-commerce Life Cycle

Figure 3 shows life cycle of E-commerce. It mainly includes user signin, order placement, payment, order delivery to their stored shipping address by the vendors.

Fig. 3. E-commerce Life Cycle

E-commerce application generates and handles highly unstructured data with volume, variety, velocity, and variability. A superior database design becomes a must to maintain high data availability and consistency. For this reason, standard data modeling guidelines have become a must-have.

ER diagram shown in Fig 5 describe how various entities of E-commerce are related to each other.

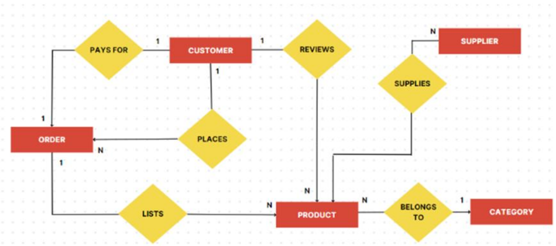

Fig. 5. ER diagram for E-commerce

Table 1 in this section enumerates few important functionalities of e-commerce application identified from relationships presented in ER diagram.

TABLE. I. DATA RELATIONSHIPS IN E-COMMERCE

Relation	Cardinality	Description
F1	1:M / 1:S	Customer Places Order
F2	M:1	Orders referring to product
F3	M:1	Products Belongs to Category
F4	M:N/ S:S	Supplier Supplies Products
F5	1:1	Customer makes payment

IV. DATA MODELING

Data modelling use data models to describe how data will be entered, captured, and modified in a database operation system [9]. Data modelling is a process of designing a data organization in the database and schema describes this database design. A well-designed data model helps ensure that the data is accurate, consistent, and complete and can be easily accessed and manipulated by the intended users. Some key benefits include increased efficiency, flexibility, and scalability. Fig. 4 describes the phases of data modelling at various abstraction level.

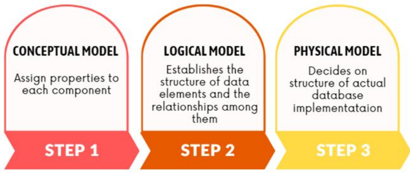

Fig. 4. Phases of data modeling

A database design comprises three main phases: a conceptual data model, a logical data model, and a physical data model.

- *Conceptual level data modeling*: The conceptual level seeks to demonstrate ideas about the system's domain to the end user by defining entities and interrelations within the model's bounds.
- *Logical level data modeling*: Unlike the previous level, the logical level focuses on creating designs with more specific ideas about the applications domain. This level is also known as the schema.
- *Physical level data modeling*: The physical level of a database management system describes implementation of logical schema using suitable data structures, file organization along with indexes to speed data access

V. RELATED WORK

Table 1 summarizes data models proposed by various researchers for E-commerce at conceptual, logical, and physical level of database design. It shows high popularity of NoSQL for E-commerce. It may because of increasing unstructured and semi-structured big-data usage in E-commerce. But, E-commerce application has critical need of consistency for order placement and payment, which suggest use of relational databases. Some developers prefer relational for complex data processing also. A very limited article like [22] discuss E-commerce design with relational databases. On other side, [5],[9],[13],[14],[15],and [21] uses document store; [16],[20] uses graph databases.

NoSQL databases are known for its flexible schema feature and the document store is most flexible NoSQL. But, the empirical results given at [2-4] shows that database design matters even in document oriented NoSQL for performance, energy, and storage requirement.

TABLE. II. DATA MODELS FOR E-COMMERCE

Ref	Year	Data Source	Conceptual	Logical	Physical
[12]	2022	Real data	ER	RDBMS, Document	PostgreSQL, MongoDB
[13]	2022	Real data	ER	Document	MongoDB
[14]	2022	Real data	UML	Key-Value, Column, Document	Oracle NoSQL, Cassandra, MongoDB
[15]	2021	Sample data	-	RDBMS, Column	SQL, Cassandra
[16]	2019	Sample data	Graph	Graph	Neo4j
[17]	2018	Real data	Generic	Column, Document	Cassandra, MongoDB
[18]	2017	Sample data	UML	Document	-
[19]	2017	Sample data	ER	Column	Cassandra
[20]	2016	Real data	UML, graph	Graph	Neo4j
[21]	2016	Sample data	ER	Document	MongoDB
[22]	2016	Sample data	UML, ER	-	MySQL
[23]	2016	Sample data	-	Document	MongoDB
[24]	2015	Real data	-	RDBMS, Document	MS SQL, MongoDB
[25]	2015	Real data	ER	Document	MongoDB
[26]	2013	Sample data	ER	Document	Oracle, MongoDB

VI. RELATIONAL DATA MODEL

A. Cardinalities

Cardinalities of relationships play a crucial role in the development of relational databases. One-to-one (1:1), many-to-many (1:M), and many-to-many (M:N) relationships are the only cardinalities considered by relational data models.

B. Relational data modeling guidelines

Entity participation also plays important role in relational data models. This section briefs various entity participation and suggest data modeling accordingly.

i. Total participation from E1

Total participation from E1 implies that every entity of E1 is associated in the relationship set. Here, Order entity is totally participated in Deliver_At relationship.

Fig. 6. Total participation of Order entity in Deliver_At

Table 3 shows the relational data modeling guidelines in case of total participation from E1, for each cardinality. E1 and E2 are the entities and PK is primary key.

TABLE. III. DATA MODELING GUIDELINES (TOTAL PARTICIPATION FROM E1)

cardinality	Total participation from E1
1:1	1. Combine table for E1 and E2 2. PK of E1 3. Table at-least in 3NF
1:M	1. Create separate table E1 and E2 2. PK of E1 in E1, PK of E2 in E2 3. PK of E2 use as FK in E1
M:N	1. Create 3 tables: E1, E2, R 2. Store PK's of E1 and E2 in R along with relationship attributes. 3. Normalize table in 3NF

ii. Total participation from E2

Total participation from E2 implies that every entity occurrence in E1 is compulsory and must be connected with a corresponding entity occurrence in E2.

Fig. 7. Pattern for total participation from E2

Table 4 shows the relational data modeling guidelines in case of total participation from E2, for each cardinality.

TABLE. IV. DATA MODELING GUIDELINES (TOTAL PARTICIPATION FROM E2)

cardinality	Total participation from E2
1:1	1. Create combine table for E1 and E2 2. PK of E2 3. Table in 3NF
1:M	1. Create separate tables E1 and E2 2. PK of E1 in E1, PK of E2 in E2 3. PK of E1 used as FK in E2
M:N	1. Create 3 tables: E1, E2, R 2. PK of E1 in E1 and PK of E2 in E2 3. Store PK's of E1 and E2 in E3 with relation of E2

iii. Partial Participation of E1 and E2

Partial participation of E1 and E2" refers to a situation where there are some instances in either E1 or E2 that are unrelated to any instances in the other entity set.

An e-commerce database can exhibit partial participation from E1 and E2 through the connection between the "Customer" and "Product" entities. It is not necessary for each customer in the "Customer" entity set to have made a purchase from the "Product" entity set, and similarly, each product may not necessarily be linked to a customer.

Fig. 8. Pattern for partial participation from E1 and E2

Table 5 shows the relational data modeling guidelines in case of partial participation from E1 and E2, for each cardinality.

TABLE. V. DATA MODELING GUIDELINES(PARTIAL PARTICIPATION OF E1/E2)

rdinality	Partial participation from E1 and E2
1:1	1. Create separate tables for E1 and E2 2. Create another table with FK of E1 and E2 3. Normalize table in 3NF

1:M	1. Create separate tables for E1 and E2 2. PK of E1 in E1, PK of E2 in E2 3. Use both PK of E1 and E2 as FK in another
M:N	1. Create separate tables for E1 and E2 2. Create another table with FK of E1 and E2 3. Storing relations of both E1 and E2 in E3

VII. NON-RELATIONAL DATA MODEL

Fig. 9 shows the growth in the CAGR of NoSQL in the global market. According to research published in [11], while the market size of RDBMS is more significant than that of NoSQL, the CAGR of NoSQL's growth rate is much higher in the coming years than that of SQL. The CAGR of SQL is only 12 percent from 2021–2029, and for NoSQL, it is around 30 percent.

Fig. 9. Growth of NoSQL in global market

A. Categorization of NoSQL data models

NoSQL, also called "Not Only SQL", facilitates the storage and retrieval of data in a non-relational database format. The widespread adoption of these databases increased after the 2000s due to the decreased storage cost and the increasing data processing requirements.

Fig 10 shows the categorization of the NoSQL database, which is categorized into four types: Document store, Key-Value store, Column-oriented, and Graph-oriented. Article [28] shows which of these NoSQL data models is maximumly used.

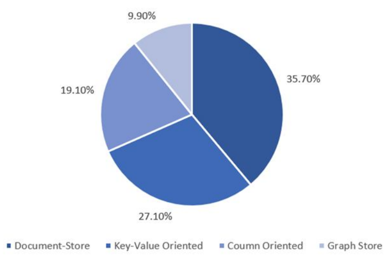

Fig. 10. Categorization of NoSQL data models

From the above graph, it is clear that the frequently adapted non-relational data model for big data is a document-oriented database.

A. Cardinalities

Unlike relational database, document oriented NoSQL uses following seven type of cardinalities for a relationship.

1. One-to-One (1:1)

A one-to-one relationship implies that a single document is exclusively linked to another document. For instance, in a document-oriented database, a customer document could have an embedded address document with a one-to-one correspondence.

2. One-to-Few (1:F)

As an instance, a document for a blog post can include comments embedded inside it, although there could be a maximum count constraint on the number of comments permitted.

3. One-to-Many (1:M)

An instance of a customer document can be associated with several order documents, for instance.

4. One-to-Squillion (1:S)

To describe a one-to-many relationship in NoSQL databases where the "many" side is extremely large, the term 1:Squillion cardinality is used. This relationship pertains to a situation where one document is linked to a large number of other documents. An example of this type of relationship would be a user document in a web application that has millions of log documents associated with it.

5. Few-to-Few (F:F)

Few:Few cardinality in NoSQL databases refers to a relationship where both entities involved have a small number of connections with each other. To illustrate, on a social media platform, users may have only a few friends, and those friends may also have only a few friends in common.

6. Many-to-Many (M:N)

In NoSQL databases, the Many:Many cardinality refers to a situation where two entities have a many-to-many relationship with each other. As an illustration, in a document-oriented database, a blog post document could be associated with multiple tag documents, and each tag document could also be associated with multiple blog posts.

7. Squillion-to-Squillion (S:S)

Squillion:Squillion cardinality in NoSQL databases refers to a relationship between two entities where both entities have an enormous number of relationships with each other. An example of this could be an e-commerce platform where a product document is related to millions of orders, and an order document is related to millions of product items.

B. NoSQL data modeling guidelines

As seen in section VII(B), the most adapted NoSQL data model in the case of Big Data is Document oriented databases. Hence, here the focus is on the data modeling guidelines of a document-oriented data model according to the data access pattern used.
In this section, Table 6 shows the different approach adapted in reviewed articles for an e-commerce based entities, here R refers to reference and E referes to embedding, given F1, F2, F3, F4, and F5 are the functionalities from Table 2 , with E1 and E2 referring to entities of the functionality, <E> referring to Embedding and <R> referring to Referencing.

TABLE. VI. DATA MODELING APPROACHES IN E-COMMERCE

Ref	F1	F2	F3	F4	F5
[12]	E1 <R> E2	E2 <R> E1	-	E1 <R> E2	-
[17]	E2 <E> E1	E2 <R> E1	E2 <E> E1	E1 <E> E2	E2 <E> E1
[18]	E1 <R> E2	E2 <R> E1	E2 <E> E1	-	E1 <R> E2
[24]	E1 <R> E2	E2 <R> E1	E2 <R> E1	E1 <R> E2	-
[25]	E2 <E> E1	E2 <R> E1	E2 <R> E1	E1 <R> E2	E2 <R> E1
[26]	E1 <R> E2	E2 <R> E1	E2 <E> E1	-	-

Below is the enlisted standard data modeling guidelines based on observations observed from carious researches for access patterns such as single access query, predicate on e1, and predicate on e2.

1. Single access query

A query for a single access retrieves only one document by using a unique identifier. For instance, in an e-commerce platform, the order ID can serve as the primary key for the order collection, and a single access query can retrieve the document for a specific order using that ID.

Table 7 shows the data modeling guidelines in case of a single access query data access pattern.

TABLE. VII DATA MODELING GUIDELINES FOR SINGLE QUERY ACCESS

1:1	Embed sub-doc in parent document or Reference parent document in related child document
1:f	Prefer embedding the few side within the one side of the document or Use indexing on both documents
1:m	Store the "many" relatd entities as an array of nested objects withi the one side or "Many" side have a separate collection with unique identifier of primary document.
1:s	Use referencing with data being partitioned or sharded or Reference with data cached in memory
f:f	Embed related child document in parent document or Index both documents or Reference primary document in related child document
m:m	Store array of embedded sub documents or Store documents in separate collection with each document having array of referenced ID of primary document
s:s	Prefer referencing with sharding or Reference with Indexing

2. Predicate on E1

To retrieve all documents that meet a specific condition on a particular field within the document, a predicate on E1 is used. For instance, in an e-commerce system, a predicate on E1 for the order collection could be used to retrieve all orders placed by a specific customer.
Table 8 shows the data modeling guidelines in case of predicate on entity 1.

TABLE. VIII DATA MODELING GUIDELINES FOR PREDICATE ON E1

1:1	Store both entities E1 and E2 within single document or E1 and E2 will have separate collection with E1 having reference of E2
1:f	Store E2 documents as an array within E1 document or E1 will have references of E2 and each E2 have reference to corresponding E1
1:m	Store references of E2 in primary document E1 or Store array of embedded document E2 in E1
1:s	Store references of multiple E2 documents within E1 or Reference with denormalized E2 data or E2 documents are stored as a large array within E1
f:f	Each E1 document contains array of embedded E2 documents or Each E1 document contains references to its associated E2 documents
m:m	Use referencing with array of references of E2 in E1 or Create a separate collection to store relationships between E1 and E2
s:s	Prefer indexing on both documents or Use hybrid method of using both embedding and referencing

3. Predicate on E2

To retrieve all documents that meet a condition on a specific field in a related document, a predicate on E2 is used. For instance, in an e-commerce system, a predicate on E2 for the order and product collections could be used to obtain all orders that contain a particular product.
Table 9 shows the data modeling guidelines in case of predicate on entity 2.

TABLE. IX DATA MODELING GUIDELINES FOR PREDICATE ON E2

1:1	Prefer embedding E1 within E2 document or Separate collection of E1 and E2 with E1 referenced in E2
1:f	Reference E1 from E2 document or Embed array of E1 instances within the E2 document
1:m	Prefer referencing E1 document in E2 document or Embed array of E1 within E2
1:s	Store references E1 within E2 or Embed E1 in E2
f:f	Use indexing Embed E1 in E2 or
m:m	E3 will have E1 and E2 ref or Store array of references of E1 in E2
s:s	E2 will have an array of sub-doc for each E1 or Prefer referencing E1 in E2

VIII. CHALLENGES AND SCOPE OF RELATIONAL AND NOSQL DATABASE

A. Challenges faced by Relational database

- *Scalability:* As the volume of data in an e-commerce application increases, it can be challenging to maintain consistent performance and response times. Scaling a relational database can be complicated and costly.
- *Security*: E-commerce apps store sensitive customer information, including credit card information, and therefore vital to ensure that the database is secure. Security breaches can significantly affect customer confidence and loyalty.
- *Complexity*: E-commerce applications can be complex, with multiple data sources and complex inter-entity relations. Dealing with these relationships in a relational database can be difficult.

B. Scope of Relational databases

- *Data Integrity:* Relational databases are intended to maintain data integrity and ensure consistency and accuracy. This is especially important for e-commerce applications, where errors in data may result in loss of sales and revenue.
- *Transaction Support:* Relational databases are built to process transactions, ensuring the modifications made to the database are consistent and reliable. It is vital in e-commerce applications, where transactions must be recorded accurately and securely.
- *Data Consistency*: Relational databases provide a coherent way to organize data, facilitating data consistency maintenance across the e-commerce application. It ensures that errors and inconsistencies in the user experience are avoided.

C. Challenges faced by NoSQL database

- *Lack of ACID compliance*: NoSQL databases prioritize scalability and availability over consistency, leading to consistent data with strong consistency guarantees.
- *Limited query capabilities*: NoSQL databases generally have limited query capabilities relative to relational databases, making data extraction for analysis more difficult.
- *Data Modeling complexity*: NoSQL databases often require a different data modeling approach from relational databases, which may require further expertise.

D. Scope of NoSQL databases

- *Scalability*: NoSQL databases are designed to process large amounts of data and are highly scalable, making them an ideal choice for e-commerce applications that require high uptime and performance.
- *Flexibility*: NoSQL databases can handle unstructured or semistructured data, allowing greater flexibility in the data types that can be stored.
- *High availability*: NoSQL databases are created to have a high level of availability, with built-in replication and aggregation features that guarantee data accessibility even in the event of a system failure.
- *Performance*: NoSQL databases can deliver high performance, particularly for heavy read workloads, due to their ability to move horizontally across multiple nodes.

IX. CONCLUSION

The exponential growth of data sources, data volume, and data variety from digital data all necessitates selecting the most efficient data model to guarantee data accuracy and consistency. Which strategies should be used and whether or not an RDBMS or NoSQL should be used is still up in the air. This study seeks to show the benefits and drawbacks of both relational and document-oriented NoSQL databases in an effort to convince the reader of the necessity of efficient data modelling. Data retrieval via complicated queries is commonplace in large enterprises; as a result, data modelling principles are presented taking into account the sort of data access pattern required.

REFERENCES

[1] Akter, S., Wamba, S.F. Big data analytics in E-commerce: a systematic review and agenda for future research. Electron Markets 26, 173–194 (2016). https://doi.org/10.1007/s12525-016- 0219-0

[2] Shah, Monika, Amit Kothari, and Samir Patel. "A Comprehensive Survey on Energy Consumption Analysis for NoSQL." Scalable Computing: Practice and Experience 23.1 (2022): 35-50.

[3] Shah, Monika, Amit Kothari, and Samir Patel. "Influence of schema design in nosql document stores." Mobile Computing and Sustainable Informatics: Proceedings of ICMCSI 2021. Springer Singapore, 2022.

[4] Thakkar, Ankit, Kinjal Chaudhari, and Monika Shah. "A comprehensive survey on energy-efficient power management techniques." Procedia Computer Science 167 (2020): 1189-1199

[5] Ramesh, Dharavath, Ekaansh Khosla, and Shankar Nayak Bhukya. ”Inclusion of e-commerce workflow with NoSQL DBMS: MongoDB document store.” 2016 IEEE international conference on computational intelligence and computing research (ICCIC). IEEE, 2016.

[6] Rodrigues, Romulo Alceu, et al. "Integrating NoSQL, relational database, and the Hadoop ecosystem in an interdisciplinary project involving big data and credit card transactions." Information technology-new generations. Springer, Cham, 2018. 443-451.

[7] “7 Ways Big Data is Changing E-commerce”, url: https://www.talend.com/resources/big-data-ecommerce

[8] Shrivastava Devashish and Mahatara Anga, “The Future of Ecommerce Industry in India”, url: https://startuptalky.com/ecommerce-industry-in-india/, 2022.

[9] Muhammad, Ananda Yulizar, and Fazat Nur Azizah. ”Conversion of Entity-Relationship Model to NoSQL Document-Oriented Database Logical Model Using Workload Information and Entity Update Frequency.” 2022 9th International Conference on Advanced Informatics: Concepts, Theory and Applications (ICAICTA). IEEE, 2022.

[10] Nakhare Disha, Hatekar Amit. “A Comparative study of SQL Databases and NoSQL Databases for E-Commerce”, Ijraset Journal For Research in Applied Science and Engineering Technology, 2021. url: https://www.ijraset.com/research-paper/sql-databases-and-nosql-databases-for-e-commerce

[11] “NoSQL Database Market: Global Industry Analysis and Forecast(2022-2029)”,url: https://www.maximizemarketresearch.com/market-report/global-nosql-database-market/97851/

[12] Shen, Wenbin. Performance comparison of NoSQL and SQL databases for different scales of e-commerce systems. Diss. Auckland University of Technology, 2022.

[13] A. Y. Muhammad and F. N. Azizah, ”Conversion of Entity-Relationship Model to NoSQL Document-Oriented Database Logical Model Using Workload Information and Entity Update Frequency,” 2022 9th International Conference on Advanced Informatics: Concepts, Theory and Applications (ICAICTA), Tokoname, Japan, 2022, pp. 1-6, doi: 10.1109/ICAICTA56449.2022.9932986.

[14] Chen, Liu, Ali Davoudian, and Mengchi Liu. ”A workload-driven method for designing aggregate-oriented NoSQL databases.” Data & Knowledge Engineering 142 (2022): 102089.

[15] Chen, Jeang-Kuo, and Wei-Zhe Lee. ”The Transformation of RDB to NoSQL DB.” 2021 International Conference on Technologies and Applications of Artificial Intelligence (TAAI). IEEE, 2021. 18

[16] Hristoski, Ilija, Tome Dimovski, and Violeta Manevska. ”Developing a B2C e-Commerce graph Database Model from a Relational Schema.” (2019): 179-182.

[17] Vega, Alfonso de la, et al. ”Mortadelo: A model-driven framework for NoSQL database design.” International Conference on Model and Data Engineering. Springer, Cham, 2018.

[18] Shin, Kwangchul, Chulhyun Hwang, and Hoekyung Jung. "NoSQL database design using UML conceptual data model based on Peter Chen's framework." International Journal of Applied Engineering Research 12.5 (2017): 632-636.

[19] Mior, Michael Joseph, et al. ”NoSE: Schema design for NoSQL applications.” IEEE Transactions on Knowledge and Data Engineering 29.10 (2017): 2275-2289.

[20] Daniel, Gwendal, Gerson Suny'e, and Jordi Cabot. "UMLtoGraphDB: mapping conceptual schemas to graph databases." International Conference on Conceptual Modeling. Springer, Cham, 2016.

[21] Lima, Cl´audio, and Ronaldo Santos Mello. ”On proposing and evaluating a NoSQL document database logical approach.” International Journal of Web Information Systems (2016).

[22] Guba, A., Ivanova Tz, and I. Batchkova. ”MODEL-DRIVEN DEVELOPMENT OF ONLINE SYSTEM FOR E-COMMERCE.” Science. Business. Society. 1.3 (2016): 58-61.

[23] Karnitis, Girts, and Guntis Arnicans. ”Migration of relational database to document-oriented database: Structure denormalization and data transformation.” 2015 7th International Conference on Computational Intelligence, Communication Systems and Networks. IEEE, 2015.

[24] Aboutorabi[a], Seyyed Hamid, et al. ”Performance evaluation of SQL and MongoDB databases for big e-commerce data.” 2015 International Symposium on Computer Science and Software Engineering (CSSE). IEEE, 2015. 16

[25] de Lima, Claudio, and Ronaldo dos Santos Mello. ”A workload-driven logical design approach for NoSQL document databases.” Proceedings of the 17th International Conference on Information Integration and Web-based Applications & Services. 2015.

[26] Ploetz, Aaron. ”A Study of MongoDB and Oracle in an E-Commerce Environment.” (2013).

[27] Hasan, Wordh Ul. ”Exploring NoSQL (Key-Value) DB vs Relational Database Management System.”

[28] “NoSQL Database Market Share, Size & Growth – Industry Report”, https://dataintelo.com/report/nosql-database-market/, 2026

[29] Gavin Fong, “6 Practical Steps to Create a Highly Efficiency Cassandra NoSQL Data Model for eCommerce Systems”, url: https://blog.devgenius.io/unleash-the-power-of-nosql-cassandra-with-query-first-datamodeling-97f1e53e8d1d, 2022.

[30] E. J. Yannakoudakis, ”Foundations of databases”, The Architectural Logic of Database Systems, pp. 1-16, 1988. 15