Proceedings of the 2023 International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/43/20230799

Investigation and designing a comprehensive supply chain database for Walmart

Jinheng Li[1,*], Yilin Lin[2], JiaXuan Ren[3,4]

[1]Department of Computer Science, University of Illinois at Urbana-Champaign, Champaign, United States
[2]AIEN Institute, Shanghai Ocean university, Shanghai, China
[3]University of Utah

[4] u1521444@umail.utah.edu
[*] Jinheng2@illinois.edu

Abstract. In light of the rapid progression of digital technology, fierce competition, and the ever-shifting demands of consumers, the contemporary retail sector is continually evolving. This underscores the critical need to embrace digital solutions. This study seeks to aid Walmart, one of the world's largest retailers, in improving supply chain efficiency, optimizing inventory management, and enhancing the overall customer experience by creating a comprehensive supply chain database. Specifically, we employ the core entity method to identify Walmart's key entity, which is the "product," and ascertain its associated entities. Subsequently, we establish relational models based on the attributes and relationships of each table, considering real-world scenarios, such as one-to-many and many-to-many relationships. We employ normalization techniques to ensure the proper utilization of functional dependencies and candidate keys for each entity. This guarantees that the designed tables adhere to the principles of the First, Second, and Third Normal Forms, preserving data integrity while minimizing redundancy. Furthermore, we create a set of logical database commands, illustrated through flowcharts, and rigorously test their functionality and efficiency. Lastly, we explore potential directions for future research. The successful implementation of this design relies on robust data management practices and a reliable database system tailored to meet Walmart's specific needs. This results in a logically structured and standardized supply chain database. The research findings underscore the vital importance of a well-conceived supply chain database in addressing the challenges faced by large-scale retail operations like Walmart supermarkets.

Keywords: Walmart, Supply Chain Management, Database Design, Core Entity.

1. Introduction

With the rapid proliferation of digital technology, diversification of consumer demands, and intensifying market competition, the retail industry faces ongoing pressure to enhance efficiency. As one of the world's largest retailers, Walmart has consistently served customers through its extensive physical retail network [1]. As of 2021, Walmart achieved an annual sales revenue of $555 billion and established an extensive global distribution network for goods and services, with its supply chain system serving as a critical pillar of its success [1]. However, the vast scale and complexity of Walmart's operations present challenges in supply chain management. In the digital era, conventional decision-making methods are no longer adequate, emphasizing the pivotal role of database systems in data management, where data-driven decisions have taken precedence [2].

This paper aims to assist Walmart in more effectively managing inventory, enhancing supply chain efficiency, and improving customer experiences by designing a supply chain database. This paper will delve into database design, including entity model, relational schema, normalization, and the creation and testing of database commands. Furthermore, a comprehensive analysis of the designed database system will be conducted, and future research directions will be explored. This research not only benefits Walmart but also provides a database solution for other enterprises to enhance data management and business performance.

2. Method

2.1. Entity-relationship diagram method

2.1.1. Theoretical issue

When confronted with significant changes in the database design of Walmart supermarkets in recent years, addressing theoretical challenges in handling vast amounts of data and the complex structure of Walmart supermarket Entity-Relationship (ER) diagrams remains daunting. These challenges encompass reducing the complexity of the ER diagram structure associated with Walmart supermarkets and efficiently managing extensive Walmart supermarket data [3].

2.1.2. Resolution methods

To address the aforementioned theoretical issues, our approach involved the identification of the core entities within Walmart supermarkets, specifically, the "product," and the determination of its associated entities. The database design method known as the "core entity" approach entails the identification of a central entity that serves as the foundation for the entire structure of the Walmart Supermarket system [4]. This core entity is crucial to modeling and establishing relationships with other associated entities within the business.

Firstly, it is imperative to determine the core entity of Walmart Supermarket, which in this case is represented by its products. All operations, such as supply management, sales, inventory control, promotions, and customer interactions revolve around these various products offered for sale. Therefore, this essay focuses on entities closely connected to products and their interactions within Walmart Supermarket's supplier chain. In order to achieve this objective effectively: In the second step, we can conduct research on the entities related to the 'product' entity and the relationship between the entities in Walmart Supermarket's supplier chain.

2.2. Normalization method

To ensure that Walmart's supply chain database complies with normalization standards, this section will approach the design of the relational database tables by addressing theoretical issues.

2.2.1. Theoretical issues

2.2.1.1. Partial dependency and transitive dependency

The relationships of dependency between attributes dictate the extent to which one attribute is influenced by others. Eessaar [5] pointed out that when certain attributes depend only on a subset of the candidate key's attributes rather than the entire candidate key, this type of dependency falls under partial dependency. Transitive dependency occurs when an attribute depends on other non-candidate key attributes. Both partial dependency and transitive dependency can lead to data redundancy [4]. Figure 1 and Figure 2 show the examples of partial dependency and transitive dependency.

DOI: 10.54254/2755-2721/43/20230799

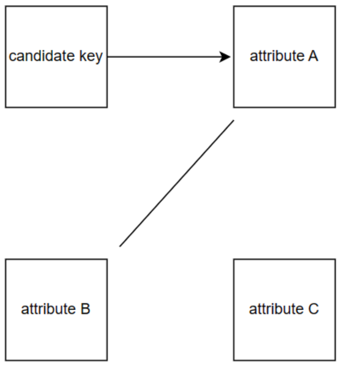

Figure 1. Partial Dependency Example.

(Photo/Picture credit: Original)

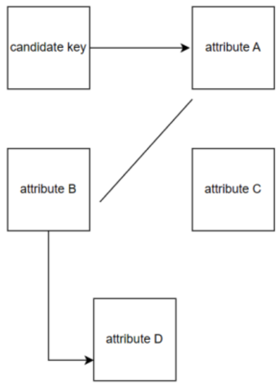

Figure 2. Transitive Dependency Example.

(Photo/Picture credit: Original)

2.2.1.2. Multi-Valued Attributes

Multi-valued attributes shown in Figure 3 are a common issue in database design, often resulting in data redundancy and consistency problems [5]. Using supply chain management as an example, a supplier may be associated with multiple departments, leading to the presence of multiple department IDs within supplier records, thus increasing data complexity.

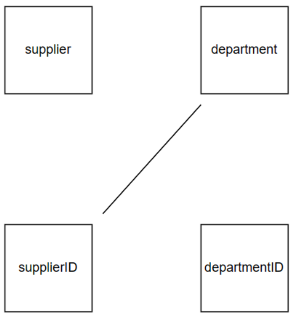

Figure 3. Multi-Valued Attribute Example (Photo/Picture credit: Original).

2.2.2. Resolution methods

To address the aforementioned theoretical issues, normalization provides a set of solutions and principles. These solutions encompass various normal forms, such as the First Normal Form (1NF), Second Normal Form (2NF), and Third Normal Form (3NF).

Determining Functional Dependencies (FD) [6, 7] and Candidate Keys (CK) [8, 9]:

In the process of database normalization, the primary task is to determine FD and CK. This step is critically important for ensuring the conformity of the database since it aids in identifying which attributes are interrelated and which attributes can uniquely identify each record [10].

We conducted an in-depth analysis of various entities within the database to identify their Functional Dependencies (FD) and Candidate Keys (CK) shown in Figure 4. In the Supplier entity, SupplierID was confirmed as the candidate key that uniquely identifies suppliers, and the existence of the functional dependency SupplierID → SupplierName was established. Through similar analyses of other entities, the team discovered that the key attributes of each entity uniquely determine the values of all other attributes within the relationship.

DOI: 10.54254/2755-2721/43/20230799

Supplier:		
Attribute	Functional Dependency	Candidate Key
SupplierID	SupplierID → supplierName, contactName, contactEmail, phoneNumber	Yes
procureDepartmentID	procureDepartmentID → departmentName, phoneNumber, location	No
DistributionCenter:		
Attribute	Functional Dependency	Candidate Key
CenterID	CenterID → centerName, address, capacity	Yes
manageDepartmentID	manageDepartmentID → departmentName, phoneNumber, location	No
ProductCategory:		
Attribute	Functional Dependency	Candidate Key
CategoryID	CategoryID → categoryName, description, SupplierID	Yes
SupplierID		No
Product:		
Attribute	Functional Dependency	Candidate Key
ProductID	ProductID → productName, description, price	Yes
CategoryID	CategoryID → categoryName, description	No
TransactionID	TransactionID → Date and Time, totalAmount, paymentMethod, SupplierID, CenterID	No
InventoryID	InventoryID → quantity, location, unitPrice, lastUpdated, CenterID, StoreID	No
Inventory:		
Attribute	Functional Dependency	Candidate Key
InventoryID	InventoryID → quantity, location, unitPrice, lastUpdated	Yes
CenterID	CenterID → centerName, address, capacity	No
StoreID	StoreID → storeName, location, openingHours	No
Order:		
Attribute	Functional Dependency	Candidate Key
OrderID	OrderID → orderDate, orderStatus, shipAddress, quantityOrdered, StoreID, CustomerID	Yes
StoreID		No
CustomerID		No
Store:		
Attribute	Functional Dependency	Candidate Key
StoreID	StoreID → storeName, location, openingHours	Yes
Customer:		
Attribute	Functional Dependency	Candidate Key
CustomerID	CustomerID → firstName, lastName, email, phoneNumber, loyalty, StoreID	Yes
StoreID		NO
Promotional Campaign:		
Attribute	Functional Dependency	Candidate Key

DOI: 10.54254/2755-2721/43/20230799

CampaignID	CampaignID → campaignName, campaignType, Holiday, description	Yes
CustomerID		No
marketDepartmentID		No
Transaction:		
Attribute	Functional Dependency	Candidate Key
TransactionID	TransactionID → Date and Time, totalAmount, paymentMethod	Yes
SupplierID		No
CenterID		No
Procurement Department:		
Attribute	Functional Dependency	Candidate Key
procurementDepartmentID	procurementDepartmentID → departmentName, phoneNumber, location	Yes
Management Department:		
Attribute	Functional Dependency	Candidate Key
manageDepartmentID	manageDepartmentID → departmentName, phoneNumber, location	Yes
Marketing Department:		
Attribute	Functional Dependency	Candidate Key
marketDepartmentID	marketDepartmentID → departmentName, phoneNumber, location	Yes

Figure 4. FD and CK for Each Table in the Walmart Database Designed by this study (Photo/Picture credit: Original).

3. Results

3.1. Entity-relationship diagram result

3.1.1. Entity model result

In creating the 13 entities, several key aspects were considered:

Supplier-Product Chain: The establishment of a one-to-many relationship between Supplier and Product serves as a cornerstone for monitoring the diverse array of products sourced from various suppliers. This relationship plays a vital role in facilitating effective inventory management and product categorization.

Distribution Center Role: The engagement of Distribution Centers in transactions and their role in replenishing inventory highlight their central position within the supply chain. This design element enables streamlined logistics management.

Store Ownership: The implementation of a one-to-one relationship between Store and Inventory signifies a direct ownership link. This simplifies inventory oversight and enhances the tracking of sales at individual store locations.

3.1.2. Relational schema result

Since our project involves a total of 13 relational models, it would be redundant to provide an extensive listing in this study. Hence, this section elucidates the three most crucial relational models in the subsequent sections. Figure 5 presents the Relationship description and corresponding information.

Supplier. The Supplier Model is for all the suppliers that cooperate with Walmart's retail department. Because there are a vast number of suppliers that work the Walmart, we will give each supplier a unique ID such that we can classify each company from each other no matter what their original names are. Still, we will include the supplier's original name in the attributes section, so the user can access the real naming after finding the desired supplier. Next, this study also includes the contact's name, email, and

DOI: 10.54254/2755-2721/43/20230799

phone number as an attribute in the Supplier Model, since it will be convenient if the user wants to communicate with a representative of the supplier. Moving on to the foreign key, according to our ER diagram, “each procurement department can be contracted with one or more suppliers”, so we have to include the procurement department’s primary key as a foreign key in the Supplier Model.

Transaction. The Transaction Model acts like a “midair transfer” between the Supplier Model, and the Distribution Model. This model will use the unique transaction ID as the primary key to classify all the transactions. The first attribute that this study includes is the data and time, i.e. the time stamp, of the purchase. Next, this study includes the total amount of money involved in the transaction, and its payment method: cash, credit card, etc. Finally, we need to add foreign keys to the Transaction Table. According to our ER diagram, “each supplier can receive one or more transactions”, and “each distribution center can be associated with one or more transactions”. Therefore, we have to include the primary key from the Supplier Model and the Distribution Model as the foreign keys in the Transaction Model.

Distribution Center. The Distribution Center Model contains all the distribution centers that Walmart has. Each distribution has been given a distribution center ID for identity validation. This study includes the center name as an attribute, so users can find out the name details after they zoom in to a specific distribution center ID. Sometimes, the distribution center’s name is not enough for locating the distribution’s physical location, so including the address information will help. Moreover, attributes like max capacity can come in handy when scheduling delivery, so shipment size does not exceed the total amount of storage space. Last but not least, the Distribution Center contains a foreign key from the Manage Department Model. According to our ER diagram, “each manage department can manage one or more distribution centers”. Thus, we have to include the primary key from the Manage Department Model to the Distribution Model as the foreign key.

Entities and Their Primary Keys	Relationship Type
Supplier (PK: SupplierID) - Product Category (PK: CategoryID)	One-to-Many
Supplier (PK: SupplierID) - Transaction (PK: TransactionID)	One-to-Many
Transaction (PK: TransactionID) - Product (PK: ProductID)	One-to-Many
Product Category (PK: CategoryID) - Product (PK: ProductID)	One-to-Many
Product (PK: ProductID) - Inventory (PK: InventoryID)	One-to-One
Transaction (PK: TransactionID) - Distribution Center (PK: CenterID)	One-to-Many
Distribution Center (PK: CenterID) - Inventory (PK: InventoryID)	One-to-Many
Procurement Department (PK: procureDepartmentID) - Supplier (PK: SupplierID)	One-to-Many
Management Department (PK: manageDepartmentID) - Distribution Center (PK: CenterID)	One-to-Many
Marketing Department (PK: marketDepartmentID) - Promotional Campaign (PK: CampaignID)	One-to-Many
Store (PK: StoreID) - Customer (PK: CustomerID)	One-to-Many
Store (PK: StoreID) - Order (PK: OrderID)	One-to-Many
Customer(PK: customerID) - Promotional Campaign (PK: CampaignID)	One-to-Many

Figure 5. Relationship description and corresponding information (Photo/Picture credit: Original).

Based on the above relationships, we can establish the entity relational diagram shown in Figure 6.

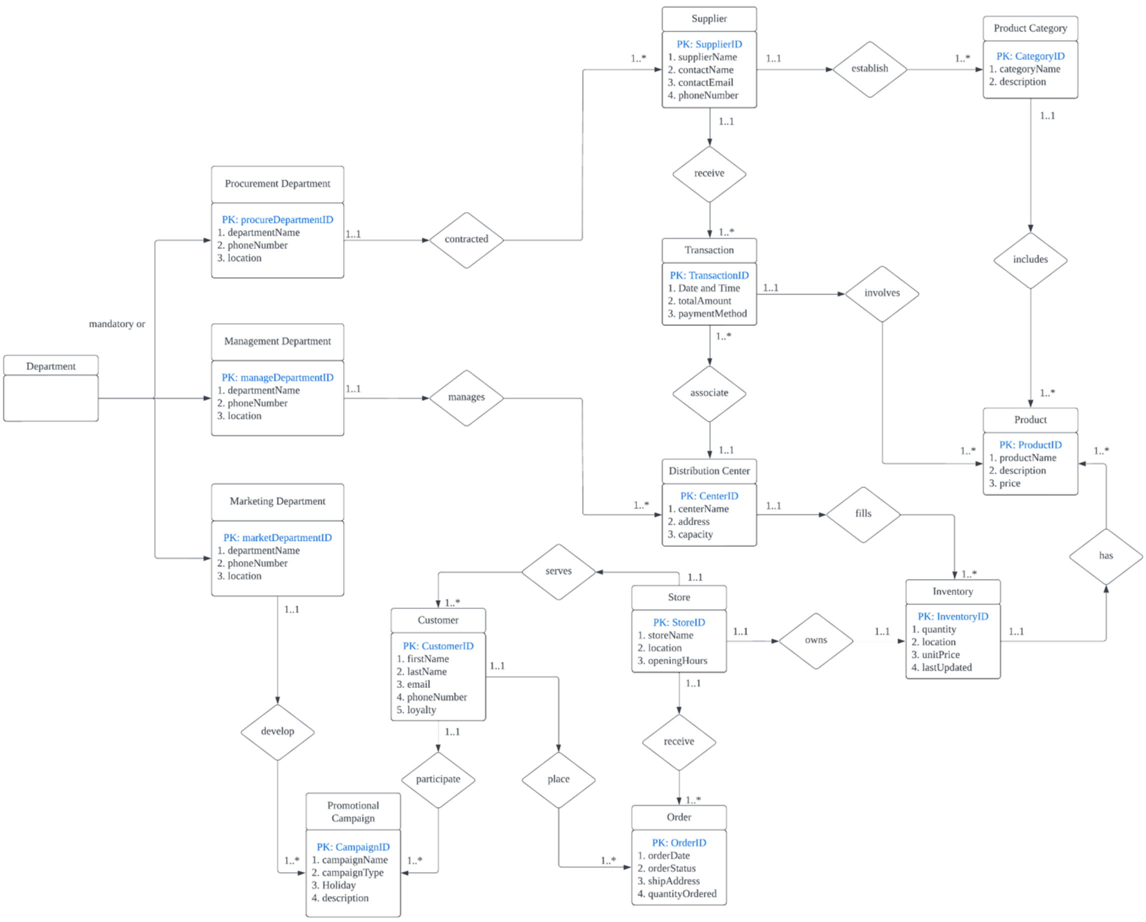

Figure 6. Entity relational diagram (Photo/Picture credit: Original).

3.2. Normalization result

By identifying FD and CK, it can be affirmed that every table within the database adheres to the prerequisites of the First Normal Form (1NF). This is due to the absence of multi-valued attributes and the presence of atomic values within each attribute. Furthermore, since the analysis of FD and CK did not reveal any partial dependency or transitive dependency relationships, each table also satisfies the requirements of the Second Normal Form (2NF) and Third Normal Form (3NF). This indicates that the structure of the designed Walmart database tables has been effectively normalized shown in Figure 7, eliminating issues of redundant data or data inconsistency.

Table Name	Attributes	Normal Formal
Product	ProductID, productName, description, price	1st 2nd 3rd normal form
ProductCategory	CategoryID, categoryName, description	1st 2nd 3rd normal form
Transaction	TransactionID, DateAndTime, totalAmount, paymentMethod	1st 2nd 3rd normal form
Inventory	InventoryID, quantity, location, unitPrice, lastUpdated	1st 2nd 3rd normal form
Order	OrderID, orderDate, orderStatus, shipAddress, quantityOrdered	1st 2nd 3rd normal form
Store	StoreID, storeName, location, openingHours	1st 2nd 3rd normal form
Customer	CustomerID, firstName, lastName, email, phoneNumber, loyalty	1st 2nd 3rd normal form
PromotionalCampaign	CampaignID, campaignName, campaignType, Holiday, description	1st 2nd 3rd normal form
Supplier	SupplierID, supplierName, contactName, contactEmail, phoneNumber, procureDepartmentID	1st 2nd 3rd normal form
DistributionCenter	CenterID, centerName, address, capacity, manageDepartmentID	1st 2nd 3rd normal form
ProcurementDepartment	procurementDepartmentID, departmentName, phoneNumber, location	1st 2nd 3rd normal form
ManagementDepartment	manageDepartmentID, departmentName, phoneNumber, location	1st 2nd 3rd normal form
MarketingDepartment	marketDepartmentID, departmentName, phoneNumber, location	1st 2nd 3rd normal form

Figure 7. Normalized Database Tables for Walmart (Photo/Picture credit: Original).

This normalization approach ensures the rationality and efficiency of the structure of Walmart's supply chain database, contributing to improved data management and query efficiency, while reducing the risk of data issues.

3.3. SQL Process

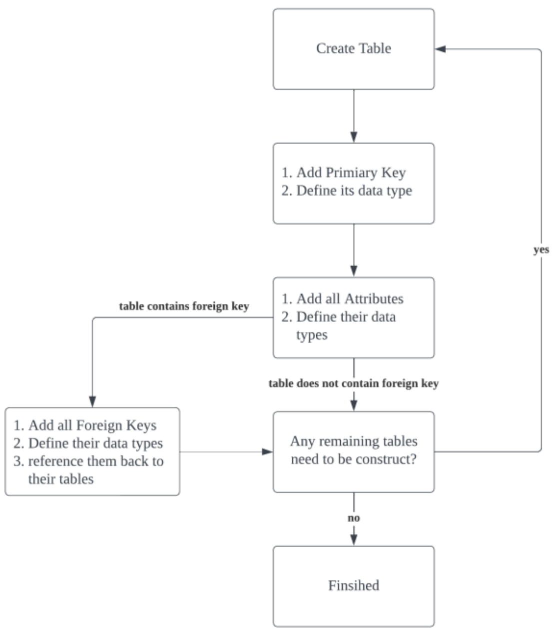

Figure 8. The process of implementing SQL (Photo/Picture credit: Original).

The above Figure 8 depicts the overall process flow for creating tables through SQL commands in this study. To maintain clarity, comprehensive code explanations are avoided. Instead, the focus shifts to elucidating the Supplier Table by referencing the accompanying chart in the following paragraph.

Supplier. Following the process outlined in the flowchart, the table for the supplier is initially created. Subsequently, the primary key for the table is declared, with the supplier ID serving as the primary key, utilizing a character data type with a size of 6. Moving on to the attributes of the table, all attributes including supplier name, contact name, contact email, and phone number are represented as character data types, with variations in their respective sizes. For instance, email addresses are typically around

25 characters long, while phone numbers are generally fixed at a size of 10, assuming a United States phone number format.

After incorporating these attributes, consideration is given to whether the table incorporates foreign keys. In this instance, the Supplier Table includes a foreign key referencing the Procurement Department Table. Consequently, a foreign key is added to the Supplier Table, employing the same character data type with a size of 5 as used in the Procurement Department Table, where the primary key was declared as such.

4. Conclusion

In conclusion, the ER diagram effectively captures the complex interactions within Walmart Supermarket's supplier chain and inventory management system. Its clear and well-defined relationships facilitate efficient data handling and provide valuable insights for optimizing inventory, logistics, and supplier relationships. However, it is important to note that successful implementation of this design will also depend on robust data management practices and a reliable database system. Overall, this ER diagram serves as a valuable blueprint for enhancing the efficiency and accuracy of Walmart Supermarket's operations within its supply chain and inventory management domains.

Authors Contribution

All the authors contributed equally, and their names were listed in alphabetical order.

References

[1] Banker S 2021 Walmart’s Massive Investment In A Supply Chain Transformation Forbes

[2] Pillai A R and Jolhe D A 2020 Market Basket Analysis: Case Study of a Supermarket. In Advances in Mechanical Engineering (pp. 727–734) Springer Singapore. https://doi.org/10.1007/978-981-15-3639-7_87

[3] Parra X et al. 2022 Chronological evolution of the information-driven decision-making process (1950–2020). J. Knowl. Econ. DOI: 10.1007/s13132-022-00917-y

[4] Wang T J et al 2017 Accounting For the Benefits Of Database Normalization Journal of Information Systems Education 28(1) 1-12

[5] Machines IJCSNS International Journal of Computer Science and Network Security 21(9)

[6] Lee M L et al 2002 Designing functional dependencies for XML. In Advances in Database Technology—EDBT 2002: 8th International Conference on Extending Database Technology Prague, Czech Republic, March 25–27, 2002 Proceedings 8 (pp. 124-141). Springer Berlin Heidelberg.

[7] Yao H et al 2002 FD/spl I. bar/Mine: discovering functional dependencies in a database using equivalences. In 2002 IEEE International Conference on Data Mining, 2002. Proceedings. (pp. 729-732). IEEE.

[8] Arman N 2006 Normalizer: A case tool to normalize relational database schemas.

[9] Lucchesi C L and Osborn S L 1978 Candidate keys for relations. Journal of Computer and System Sciences, 17(2), 270-279.

[10] Eessaar E 2016 The Database Normalization Theory and the Theory of Normalized Systems: Finding a Common Ground