

E-commerce Database Management System

DBMS Technical Report: House Depot

Course: CPS 510 - Data Systems I

Assignment 1

Jairo Mendoza - 501039006

Arooran Sivakarunai - 501183635

Daniel Mielnik - 501118927

Sep 16, 2024

1. Abstract

This report details the design and implementation of a Database Management System (DBMS) for "House Depo", which will serve as a tool, material, and home improvement E-commerce platform. The DBMS will aim to facilitate the operations of "House Depo", providing the backbone to the E-commerce venture. This DBMS will handle the storage, retrieval, and relationships between products, customers, vendors (sellers), transactions, and reviews. The following report will cover the database's architecture and design choices, normalization choices, and the SQL queries set in place to interact with the system. This report will also highlight the non-functional aspects of the DBMS, such as detailing performance, scalability, security, and data integrity. Finally, a table representation of the DBMS schema is given to clarify the scope and entity relationships concerning the data used for the "House Depo" e-commerce platform.

2. Introduction

"House Depo" is an E-commerce platform for tools, materials, and home improvement products. To efficiently manage data related to products, customers, vendors, transactions, reviews, and inventory, a robust Database Management System (DBMS) is essential. The primary purpose of creating this DBMS is to provide a scalable

foundation for organizing and retrieving critical information, ensuring a seamless and reliable online shopping experience for all users.

This project aims to provide all of the key features a thriving e-commerce business requires, including the following primary objectives:

1. Designing a scalable, efficient, and reliable database structure that effectively organizes data related to products, orders, customers, vendors, and reviews.
2. Ensuring seamless data management and information retrieval while minimizing redundancy and maintaining data integrity.
3. Implementing an optimal database system to support the smooth operation of the "House Depo" E-commerce platform.

In addition to these key objectives, it is important that we are able to manage and organize critical entity-relationships that are fundamental to our e-commerce platform.

The scope of such key entities are list below as follows:

1. Products: Data related to product listings, categories, pricing, and availability.
2. Customers: Information about registered customers, such as contact details, shipping addresses, and preferences.
3. Vendors: Details about sellers, including product inventories, ratings, and contact information.
4. Transactions: Order details, including purchase history, payment status, shipping information, and returns/refunds.

5. Inventory: Management of product quantities, stock levels, and tracking of inventory for vendors.

3. System Requirements

As a functioning e-commerce platform, there is a set of absolute tasks required to facilitate, sustain, and carry-out operations that are carried out by a DBMS. These requirements are as follows:

1. Store and retrieve product data (name, price, category, stock)
2. Handle customer information (name, contact, shipping details)
3. Process orders (order details, payment, status)
4. Manage reviews (product ratings, feedback)
5. Keep detailed records of inventory (stock, price, and shipment information)

Alongside these properties, using a DBMS also provides unique benefits which can help provide a better experience for developers, vendors and customers, such as:

1. Performance: Fast retrieval of product and order information.
2. Scalability: Ability to handle growing customer and product base.
3. Security: Secure customer and payment information.
4. Data Integrity: Ensure accuracy and consistency of order records.
5. Analytics: Using data stored to make basic insights and suggestions.

4. Database Design

Customers Table

Field	Data Type	Description
customer_id	Int	Unique identifier for each customer.
name	String	Customer's full name.
phone_number	String	Customer's contact number.
email	String	Customer's email address
recovery_email	String	Optional secondary email for account recovery.
address	String (For Now)	Customer's shipping and billing address.
date_of_birth	Date	For birthday promotions
user_role	Enum String	“Customer” or “admin”
password_hash	String	For security user login credentials

Seller table

Field	Data Type	Description
seller_id	Int	Unique Identifier for each seller
name	String	Seller's name.
phone_number	String	Seller's contact number.
email	String	Seller's email address.
rating	Float	Aggregated rating for the seller based on customer reviews.
account_status	String Enum	“Active”, “suspended”, to

		manage seller accounts
--	--	------------------------

Product Table

Field	Data Type	Description
product_id	Int	Unique identifier for each product.
product_name	String	Name of the product.
product_description	String	Detailed description of the product.
product_price	Float	Price of the product
product_brand	String	Product's brand
product_weight	Float	Product's weight to help with the shipping calculations
categories	String	Categories to which the product belongs.
seller_id	Int	The seller offering the product.

Inventory Table

Field	Data Type	Description
product_id	Int	Unique identifier of the product .
seller_id	Int	Seller offering the product.
in_stock	Int	Current stock available for the product.
minimum_stock_level	Int	To trigger restock alerts
next_deliver_date	Date	The expected date for the next delivery of the

		product.
next_shipment_amout	Int	Quantity expected in the next shipment.

Order_History Table

Field	Data Type	Description
transaction_id	Int	Unique identifier for each transaction.
payment_method	String Enum	'Credit Card', 'debit', 'paypal'. Method use to pay the order
order_total	Float	Total cost of the order
shipping_address	String (For Now)	Address where the order will be delivered.
user_id	Int	Reference to the customer placing the order.
seller_id	Int	Seller providing the product.
status	String	Order status (e.g., "shipped", "delivered").
product_id	Int	Product related to the transaction.
order_date	Date	Date the order was placed.

RELATIONSHIPS:

- CUSTOMER is related to ORDER_HISTORY via customer_id

- A CUSTOMER can have many ORDER_HISTORY, an ORDER_HISTORY can only have one CUSTOMER
- SELLER is related to PRODUCT via seller_id
 - A SELLER can have many PRODUCT, a PRODUCT can only have one seller
- SELLER is related to ORDER_HISTORY via seller_id
 - A SELLER can have many ORDER_HISTORY, an ORDER_HISTORY has only one seller
- SELLER is related to INVENTORY via seller_id
 - A SELLER can have many INVENTORY, an INVENTORY can only have one SELLER
- PRODUCT is related to INVENTORY via product_id
 - A PRODUCT can only have one inventory, an INVENTORY can only have one PRODUCT
- PRODUCT is related to ORDER_HISTORY via product_id
 - A PRODUCT can have many ORDER_HISTORY, AN ORDER_HISTORY can have many PRODUCT

5. Conclusion

In conclusion, the development of a comprehensive Database Management System (DBMS) for the "House Depo" E-commerce platform will address the need for efficient data organization and retrieval. The system manages critical information related to products, customers, vendors, transactions, reviews, and inventory, ensuring a seamless and reliable shopping experience for all users. Key features of the DBMS include scalable architecture, optimized data management, and streamlined information retrieval processes. The successful implementation of this robust Database Management System (DBMS) will not only address "House Depo's" need for efficient data management but also demonstrate the powerful impact of utilizing advanced database technology to optimize e-commerce operations and enhance user experiences.