**PROJECT REPORT**

Group 23

Student1 Manoghn Kandiraju

Student2 Rishwanth Reddy

857-423-5798 (Tel of Student 1)

737-881-2194 (Tel of Student 2)

[kandiraju.m@northeastern.edu](mailto:kandiraju.m@northeastern.edu)

[yadamakanti.r@northeastern.edu](mailto:yadamakanti.r@northeastern.edu)

**Percentage of Effort Contributed by Student1: 50%**

**Percentage of Effort Contributed by Student2: 50%**

**Signature of Student 1: Manoghn**

**Signature of Student 2: Rishwanth Reddy**

**Submission Date: 10th December, 2023 \_\_\_\_\_\_\_\_\_\_\_**

**EXECUTIVE SUMMARY:**

This project focuses on applying data management to enhance many facts about e-commerce online retail operations and data management principles. All issue statements will be discussed and show how data management and analysis can provide answers and insights.

The major data management that we considering is the profit that the e-commerce website is earning considering all the product costs and revenue generated which will be incurred to deliver the product to the customers.

By analysing past data of e-commerce, which includes product categories, which products are getting sold in high demand, product category performance, customer behaviour with the product he bought, and calculating the profit of the category of a product, Maximizing profit needs Gathering historical data on an e-commerce site that includes sales transactions (i.e., order date, product date, product ID, revenue generated, cost, and product categories)

We are going to compare all the categories and obtain the profit margin among the categories. We are going to compare the reviews obtained for each category and also find out the category which is in demand and also to indicate which stock to increase so that e-commerce could further increase the profit.

**I. INTRODUCTION**

Convenience is king in the age of digital technology today. Thanks to electronic commerce, or e-commerce, the internet, which was originally used exclusively for information and networking, has evolved into a humming marketplace. For people all across the world, this phenomenon has made daily life simpler.

E-commerce, which includes both retail and wholesale, is quickly increasing in popularity and is quickly becoming a crucial component of contemporary business. It gives customers a wide range of options, including a variety of brands, sizes, and product groups including electronics and housewares. Customers can easily examine these possibilities and make selections from many vendors through online apps, encouraging healthy competition.

E-commerce stands out for its ability to offer products at various prices to suit all price points, from budget-friendly to upscale. Online purchasing is becoming widely used as a result of this accessibility. As a result, companies are scrambling to build a strong online presence with transactional capabilities to satisfy customers' changing needs.

In conclusion, e-commerce has completely changed how businesses operate by providing unmatched ease and choice. Businesses are adjusting as it grow and using online platforms to meet the ever-changing demands of their clients.

**REQUIREMENTS:**

The system revolves around a product-centric approach, where each product is categorized and associated with various entities. The product is categorized under a specific category, ensuring that each product belongs to one and only one category. Conversely, a category can encompass one or more products.

* Products are supplied by vendors, and each product can be supplied by one or more vendors. On the other hand, vendors can supply multiple products, establishing a many-to-many relationship.
* The purchasing process involves users who can buy multiple products, and each product can be purchased by multiple users. This implies a many-to-many relationship between users and products.
* Additionally, products have a stock status, indicating the availability or quantity of the product. Optionally, each stock status is associated with a single product.
* Users can provide reviews for products, with each user capable of writing zero to multiple reviews. This relationship allows for user-generated feedback on products.
* Transactions involve users making purchases, creating a relationship where each user can have zero to multiple transactions. Furthermore, each transaction is associated with a specific payment method, and each payment method can be linked to multiple transactions.
* A distinct payment status is assigned to each transaction, signifying the state of payment completion. This ensures that each transaction has precisely one payment status.
* For efficient management, payment methods have a default payment status, providing a standardized initial state for transactions associated with that payment method. This default payment status can be associated with multiple payment methods, offering flexibility in defining default states.

In summary, the system is designed to handle the dynamic relationships between products, categories, vendors, users, reviews, transactions, payment methods, payment statuses, and stock statuses, facilitating a comprehensive and interconnected platform for product management and user transactions.

**II. CONCEPTUAL DATA MODELING**

**EER DIAGRAM**

**A diagram of a company

Description automatically generated**

Fig 1. EER Diagram of E-Commerce

**UML CLASS DIAGRAM**

**A diagram of a software application

Description automatically generated with medium confidence**

Fig 2. UML Class Diagram of E-Commerce

**III. MAPPING CONCEPTUAL MODEL TO RELATIONAL MODEL**

**Product(product\_id, product\_name, id\_user\_bought, category\_id, discounted\_price, actual\_price, discount\_percentage, rating, rating\_count, about\_product, vendor\_id)**

* product\_id: Primary key and NOT NULL
* id\_user\_bought: Foreign key (references User(user\_id))
* category\_id: Foreign key (references Category(category\_id))
* vendor\_id: Foreign key (references Vendor(vendor\_id))

**Category(category\_id, category\_name)**

* category\_id: Primary key and NOT NULL

**User(user\_id, user\_name, user\_gender, user\_age, user\_wishlist\_count)**

* user\_id: Primary key and NOT NULL

**Review(review\_id, product\_id, user\_id, review\_title, review\_content, review\_date)**

* review\_id: Primary key and NOT NULL
* product\_id: Foreign key (references Product(product\_id))
* user\_id: Foreign key (references User(user\_id))

**Vendor(vendor\_id, vendor\_name, vendor\_loc)**

* vendor\_id: Primary key and NOT NULL
* vendor\_name: NOT NULL
* vendor\_loc: NOT NULL

**Transaction(transaction\_id, user\_id, product\_id, purchase\_date)**

* transaction\_id: Primary key and NOT NULL
* user\_id: Foreign key (references User(user\_id))
* product\_id: Foreign key (references Product(product\_id))

**PaymentMethod(user\_id, payment\_method\_id, payment\_method\_name)**

* user\_id: Foreign key (references User(user\_id))
* payment\_method\_id: Primary key and NOT NULL

**ProductImageLink(product\_id, img\_link, product\_link)**

* product\_id: Primary key and Foreign key (references Product(product\_id))

**StockStatus(stock\_status, product\_id, stock\_quantity)**

* product\_id: Primary key and Foreign key (references Product(product\_id))

**PaymentStatus(transaction\_id, payment\_status)**

* transaction\_id: Foreign key (references Transaction(transaction\_id))

**IV. IMPLEMENTATION OF RELATION MODEL VIA MYSQL AND**

**NOSQL**

**MySQL Implementation:**

The database was created in MySQL and the following queries were performed:

***Query: -1***

**Retrieve the names of products along with their discounted prices and vendor names for products with a rating greater than 4**

SELECT p.product\_name, p.discounted\_price, v.vendor\_name FROM product\_table p  
JOIN vendor\_table v ON p.vendor\_id = v.vendor\_id  
WHERE p.rating > 4

GROUP BY p.product\_name, p.discounted\_price, v.vendor\_name LIMIT 0, 1000;

A screenshot of a computer

Description automatically generated

***Query: -2***

**List the products with a discount percentage greater than 20% and order them by the discounted price in descending order**

SELECT product\_name, discounted\_price, discount\_percentage

FROM product\_table WHERE discount\_percentage > 20

ORDER BY discounted\_price DESC;

A screenshot of a computer

Description automatically generated

***Query: -3***

**Retrieve the payment methods used by each user along with the count of transactions for each payment method**

SELECT u.user\_name, pm.payment\_method\_name, COUNT(t.transaction\_id) AS transaction\_count

FROM user\_table u LEFT JOIN transaction\_table t ON u.user\_id = t.user\_id

LEFT JOIN payment\_method\_table pm ON u.user\_id = pm.user\_id

GROUP BY u.user\_name, pm.payment\_method\_name;

A screenshot of a computer

Description automatically generated

***Query: -4***

**Retrieve the latest review for each product along with the reviewer's name and review date**

SELECT r.review\_id, p.product\_name, u.user\_name, r.review\_date

FROM review\_table r JOIN product\_table p ON r.product\_id = p.product\_id

JOIN user\_table u ON r.user\_id = u.user\_id

WHERE r.review\_date = (SELECT MAX(review\_date) FROM review\_table WHERE product\_id = p.product\_id);

A screenshot of a computer

Description automatically generated

***Query: -5***

**List the products with low stock quantities (less than 10) along with their current stock status:**

SELECT p.product\_name, s.stock\_quantity, ss.stock\_status

FROM product\_table p JOIN stockstatus\_table s ON p.product\_id = s.product\_id

JOIN stockstatus\_table ss ON s.product\_id = ss.product\_id WHERE s.stock\_quantity < 10;

A screenshot of a computer

Description automatically generated

***Query: -6***

**Total Revenue from Product Sales per Category**

SELECT category\_name, SUM(actual\_price - discounted\_price) AS total\_revenue

FROM product\_table

JOIN category\_table ON product\_table.category\_id = category\_table.category\_id

GROUP BY category\_name;

***A screenshot of a computer

Description automatically generated***

***Query: -7***

**Calculate the total revenue generated from product sales for each vendor:**

SELECT vendor\_name, SUM(actual\_price - discounted\_price) AS total\_revenue

FROM product\_table

JOIN vendor\_table ON product\_table.vendor\_id = vendor\_table.vendor\_id

GROUP BY vendor\_name;

***A screenshot of a computer

Description automatically generated***

***Query: -8***

**Find the total stock quantity for each product and its stock status:**

SELECT product\_name, stock\_status, SUM(stock\_quantity) AS total\_stock\_quantity

FROM product\_table

JOIN stockstatus\_table ON product\_table.product\_id = stockstatus\_table.product\_id

GROUP BY product\_name, stock\_status;

***A screenshot of a computer

Description automatically generated***

**NoSQL Implementation:**

Tables were imported to Mongodb. The queries are as follows:

***Query: - 1***

**Find all products with the categoryid 0, limiting to 3**

db.Product.find({ category\_id: 0 }).limit(3)

**A screenshot of a computer program

Description automatically generated**

***Query: - 2***

**Find the names and genders of users who purchased a product with a discount percentage greater than 50%.**

db.User.find({

"user\_id": {

$in: db.Transaction.distinct("user\_id", {

"product\_id": {

$in: db.Product.distinct("product\_id", {

"discount\_percentage": { $gt: 50 }

})

}

})

}

}, { "\_id": 0, "user\_name": 1, "user\_gender": 1 })

A screen shot of a computer

Description automatically generated

***Query: - 3***

**Calculate the average rating for each vendor.**

db.Review.aggregate([

{ $lookup: { from: "Product", localField: "product\_id", foreignField: "product\_id", as: "products" } },

{ $unwind: "$products" },

{ $lookup: { from: "Vendor", localField: "products.vendor\_id", foreignField: "vendor\_id", as: "vendors" } },

{ $group: { \_id: "$vendors.vendor\_name", average\_rating: { $avg: "$products.rating" } } },

{ $project: { \_id: 0, vendor\_name: "$\_id", average\_rating: 1 } },

{ $sort: { average\_rating: -1 } }

])

A screenshot of a computer

Description automatically generated**V. DATABASE ACCESS VIA R OR PYTHON**

Python is used to access the database, and the data visualizations that result from the analysis are displayed below. MySQL. connector is used to connect MySQL to Python. Data frames are then created for each table, and Matplotlib is used to plot the graphs for the analytics.

Connected this to Python application with the following code:-

import mysql.connector

#

from mysql.connector import Error

#

try:

connection = mysql.connector.connect(host='localhost',

database='DMA\_PJT',

user='root',

password='Mysql@2321',

auth\_plugin = 'mysql\_native\_password')

if connection.is\_connected():

db\_Info = connection.get\_server\_info()

print("Connected to MySQL Server version ", db\_Info)

cursor = connection.cursor()

cursor.execute("select database();")

record = cursor.fetchone()

print("Your connected to database: ", record)

#

sql\_select\_Query = "select DISTINCT(product\_name) from product\_table"

cursor = connection.cursor()

cursor.execute(sql\_select\_Query)

records = cursor.fetchall()

print("Data of Product names:\n")

for row in records:

print('Product Name =',row[0],"\n")

#

except Error as e:

print("Error while connecting to MySQL", e)

finally:

if (connection.is\_connected()):

cursor.close()

connection.close()

print("MySQL connection is closed")

**Total revenue generated from product sales for the top 7 vendors.**

A graph of different colored rectangular shapes

Description automatically generated with medium confidence

**Average rating for each vendor**

A colorful pie chart with text

Description automatically generated

**Top 5 products with the highest stock quantity and their stock status**

**A graph with a green line

Description automatically generated**

**VI. SUMMARY AND RECOMMENDATION**

**Market Concentration and Vendor Diversity:**

Graph 1 highlights a concentrated market share among the top 7 vendors, with a substantial portion of revenue attributed to the top 3 vendors (DOVICE, TOO, and job 10). The company should prioritize diversifying its vendor portfolio to mitigate risks associated with dependency on a few dominant players, ensuring a more resilient and sustainable supply chain.

**Optimized Inventory Management:**

Graph 2 showcases the top 5 products with the highest stock quantity, all of which are currently in stock. The company appears to have effectively managed its inventory, maintaining optimal stock levels for these high-quantity products. However, continuous monitoring of stock levels and demand patterns is recommended to avoid potential overstocking.

**Category-driven Sales Performance:**

Graph 3 emphasizes the sales performance across various product categories, with Electronics, Home and kitchen, and Clothing and accessories contributing significantly to overall revenue. The company should capitalize on the strong sales performance in these categories but also consider expanding its product offerings into under-represented categories to diversify revenue streams and capture a broader market share.

The overall insights suggest positive performance and ample inventory but also indicate potential risks related to market concentration and overstocking. A balanced approach is crucial. The company should leverage its strengths but actively address the identified risks through strategic measures to ensure sustained growth and mitigate potential challenges.

**Actionable Recommendations:**

Diversify vendor portfolio to reduce dependence on specific suppliers. Implement robust inventory monitoring systems to optimize stock levels based on demand. Explore opportunities in categories with lower representation to expand the product portfolio. Establish customer feedback systems to gather insights for continuous improvement in products and services.

By implementing these actionable recommendations, the company can enhance its market resilience, adapt to changing dynamics, and foster long-term success in a competitive landscape.