# USE CASE STUDY REPORT

E-Commerce Data Management System

**Group No**.: Group 23

**Student Names:** Akshita Singh and Amoolya Bagalkoti

## I. Introduction

**Executive Summary:**
E-commerce is a major source of data these days. Millions of orders are being place and delivered each day. The following project "E-Commerce Data Management System" can be looked at as a proof of concept (poc) about managing the e-commerce data and using it for analytical purposes at a large scale.

**Problem Statement:**

Prior to the development of the internet, customers had to physically visit a store to make purchases, but this is no longer the case. People can now purchase goods online from the comfort of their homes. This has resulted in the expansion of the e-commerce market. Due to this, tons of data are produced and stored every second. Modern information technology is required to manage this e-commerce system due to the growing number of people engaging in electronic commerce. The e-commerce sector depends heavily on databases, and in the current environment, an e-commerce company's performance is directly related to how well it has optimized its database and so, efficient data management is crucial. Along with it, identifying trends and relevant conclusions from the data is also very important to beat the competition. This problem can be resolved with the aid of an effective eCommerce database management system in which the data is organized in a proper format, which helps in accessing data more efficiently by e-commerce applications. Furthermore, Ecommerce Analytics helps you determine the volume of sales of a particular product and understand the buying trends of the customers/users which will be helpful in building a recommendation system. We can keep track of inventory, track the products that are going out of stock and restock them well in advance.

**Goal:**

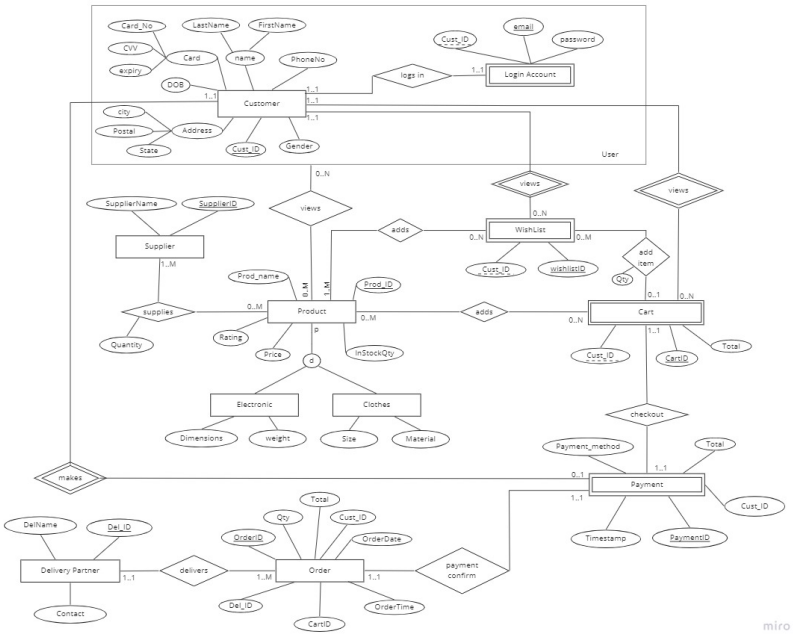
The goal of this E-Commerce Data Management system is to showcase managing of retail data and utilizing the data for analytical purposes.

**Requirements:**

- MySQL workbench
- MongoDB Compass
- Jupyter Notebook (Python)

# II. Conceptual Data Modeling

Enhanced Entity Relationship Model:




UML Class Diagram:

## III. Mapping Conceptual Model to Relational Model

E-Commerce Data Relational Model
*{ Primary Keys are underlined and Foriegn Keys are denoted with * }*

Customer ( CustID, FirstName, LastName, PhoneNumber, Gender, DOB)
CustomerAddress( CustID*, postalcode, city, state)
CustomerBilling( CardNo, CardExpiry,CVV, CustID*)
Login(Email, Password, CustID*)
Supplier ( SupplierID, SupplierName )
supplies ( SupplierID*, ProdID*, Quantity )
Product ( ProdID, ProdName, Price, Rating, InstockQty)
Electronic ( ProdID*, Dimensions, Weight)
Clothes ( ProdID*, size, material )
prodToCart ( ProdID*, CartID*, Qty)
Cart ( CartID, CustID*)
Wishlist( WishlistID, CustID*)
ProductToWishlist( WishListID*, ProdID*)
Payment( PaymentID, PaymentMethod, CustID*, Timestamp)
Orders ( OrderID, OrderDate, OrderTime, CartID*,CustID*, DelID*, PaymentID*)
DeliveryPartner ( DelID, DelName, Contact)

## IV. Implementation of Relation Model via MySQL and NoSQL

MySQL DDLs:

```
1   create schema ecommerce;
3   use ecommerce;
8   create table customer (
9       Custid INT NOT NULL primary key,
10      FirstName VARCHAR(50),
11      LastName VARCHAR(50),
12      PhoneNumber VARCHAR(50),
13      Gender VARCHAR(50),
14      DOB date
15  );
117 -- creating address table
118 create table CustomerAddress (
119     Custid INT NOT NULL,
120     postalCode VARCHAR(50) not null,
121     city VARCHAR(50),
122     state VARCHAR(50),
123     FOREIGN KEY(Custid) references customer(Custid)
124     ON DELETE CASCADE ON UPDATE CASCADE,
125     primary key (Custid,postalCode)
126 );
```

```sql
CREATE TABLE login(
    `email` varchar(255) NOT NULL Primary key,
    `password` varchar(255),
    `custID` int NOT NULL,
    FOREIGN KEY(Custid) references customer(Custid) ON DELETE CASCADE ON UPDATE CASCADE
) ;
CREATE TABLE Supplier(
    `SupplierID` int NOT NULL primary KEY,
    `SupplierName` varchar(255) default NULL
);
    -- deliverypartner table
    create table DeliveryPartner (
      delID INT NOT NULL primary key,
      delName VARCHAR(50),
      Contact VARCHAR(50)
);
CREATE TABLE CustomerBilling (
    `cardNo` varchar(255) NOT NULL,
    `CVV` varchar(255),
    `CardExpiry` varchar(255),
    `Custid` int not null,
    primary key (cardNo, Custid),
    FOREIGN KEY(Custid) references customer(Custid) ON DELETE CASCADE ON UPDATE CASCADE
);
create table product (
    ProdID INT NOT NULL Primary key,
    ProdName VARCHAR(80),
    Rating DECIMAL(2,1),
    Price DECIMAL(6,2),
    InStockQuantity INT
);
create table electronics (
    ProdID INT NOT NULL primary key,
    dimensions varchar(50),
    weight float,
    FOREIGN KEY(ProdID) references Product(ProdID) ON DELETE CASCADE ON UPDATE CASCADE
);
create table clothes(
    ProdID INT NOT NULL primary key,
    size varchar(50),
    material varchar(20),
    gender varchar (40),
    FOREIGN KEY(ProdID) references Product(ProdID) ON DELETE CASCADE ON UPDATE CASCADE
);
create table supplies (
    ProdID INT NOT NULL,
    SupplierID INT NOT NULL,
    Quantity INT,
    primary key (ProdID, SupplierID),
    FOREIGN KEY(SupplierID) references supplier(SupplierID) ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY(ProdID) references Product(ProdID) ON DELETE CASCADE ON UPDATE CASCADE
);
create table payment (
    paymentID INT not null primary key,
    paymentmethod VARCHAR(11),
    custID INT,
    timestamp DATE,
    FOREIGN KEY(custID) references customer(custID) ON DELETE CASCADE ON UPDATE CASCADE
);
```

```
743 • create table cart (
744       cartID INT not null primary key,
745       custID INT,
746       FOREIGN KEY(custID) references customer(custID) ON DELETE CASCADE ON UPDATE CASCADE
747   );
805 • create table wishlist (
806       wishlistID INT not null primary key,
807       custID INT,
808       FOREIGN KEY(custID) references customer(custID) ON DELETE CASCADE ON UPDATE CASCADE
809   );
837 • create table productTowishlist (
838      wishlistID INT,
839      prodID INT,
840      primary key(wishlistID,prodID),
841      FOREIGN KEY(prodID) references product(prodID) ON DELETE CASCADE ON UPDATE CASCADE,
842          FOREIGN KEY(prodID) references product(prodID) ON DELETE CASCADE ON UPDATE CASCADE,
843      FOREIGN KEY(wishlistID) references wishlist(wishlistID) ON DELETE CASCADE ON UPDATE CASCADE
844   );
890 • create table orders (
891       orderID INT not null primary key,
892       orderDate DATE,
893       orderTime VARCHAR(50),
894       cartID INT,
895       custID INT,
896       paymentID INT,
897       DelID INT,
898       FOREIGN KEY(custID) references customer(custID) ON DELETE CASCADE ON UPDATE CASCADE,
899       FOREIGN KEY(paymentID) references payment(paymentID) ON DELETE CASCADE ON UPDATE CASCADE,
900       FOREIGN KEY(delID) references deliverypartner(delID) ON DELETE CASCADE ON UPDATE CASCADE,
901       FOREIGN KEY(cartID) references cart(cartID) ON DELETE CASCADE ON UPDATE CASCADE
902   );
931 • create table prodToCart (
932       cartID INT not null,
933       prodID INT not null,
934       Qty INT,
935       primary key(cartID, prodID),
936       FOREIGN KEY(prodID) references product(prodID) ON DELETE CASCADE ON UPDATE CASCADE,
937       FOREIGN KEY(cartID) references cart(cartID) ON DELETE CASCADE ON UPDATE CASCADE
938   );
```

MySQL Queries:

1.Products where instock quantity is less than 10.

SELECT ProdName, ProdID , InstockQuantity
FROM Product
WHERE InstockQuantity < 10;

| ProdName | ProdID | InstockQuantity |
|---|---|---|
| calvin klein pants | 812 | 2 |
| Ponds defining cream | 813 | 8 |
| Apple Tv | 824 | 9 |
| Bath and bodyworks conditioner | 842 | 5 |
| Asus rog gaming laptop | 844 | 9 |

Analytical purpose: to help restock the items in time. The supplier can be notified about restocking when the instock quantity goes below 10.

2. Product with minimum rating.

```
SELECT Prodname, rating
FROM product
WHERE rating = (SELECT MIN(rating)
FROMproduct);
```

| | Prodname | rating |
|---|---|---|
| ▶ | samsung galaxy flip z4 | 2.1 |

Analytical purpose : Helps in sending in constructive feedback about the product to the supplier or manufacturer.

3. Creating a view on top of cart table to calculate the total no.of items in the cart and the total cart price. This view can be further used as a table.

```
CREATE VIEW cart_view AS (
SELECT c.cartID, c.custID , SUM(pc.qty) AS totalItemsInCart, SUM(pc.qty*p.price) AS total
FROM cart c
JOIN prodtocart pc ON c.cartID=pc.cartID
JOIN product p ON p.prodID=pc.prodID
GROUP BY c.custID);
SELECT * FROM cart_view;
```

| | cartID | custID | totalItemsInCart | total |
|---|---|---|---|---|
| ▶ | 302 | 42 | 3 | 85.47 |
| | 303 | 48 | 9 | 5861.81 |
| | 304 | 12 | 8 | 1812.00 |
| | 306 | 88 | 4 | 3999.36 |
| | 310 | 46 | 4 | 50.36 |
| | 311 | 5 | 6 | 115.59 |
| | 314 | 45 | 6 | 101.34 |
| | 315 | 92 | 3 | 3901.08 |
| | 317 | 91 | 2 | 2598.00 |
| | 318 | 73 | 4 | 430.36 |
| | 319 | 11 | 9 | 9228.68 |
| | 320 | 7 | 10 | 3168.23 |
| | 321 | 24 | 4 | 1824.72 |
| | 323 | 68 | 3 | 2999.52 |

Analytical Purpose : Since totalItemsInCart and the total price are both derived. This calculation in the view will be helpful.

NoSQL Queries:
The data from MySQL workbench was exported in json format and imported into MongoDB Compass.

1. To list distinct materials the clothes are offered in.

```
Ecommerce> db.clothes.distinct("material")
[
  'Fleece', 'Polyester',
  'cotton', 'courdrouy',
  'denim',  'fleece',
  'linen',  'polyester',
  'silk'
]
```

Analytical Purpose: Helpful in understanding the different variety of materials being offered by the system.

2. To find the Clothes that have clothes in plus Sizes.

```
Ecommerce> db.clothes.find({"size":/XXL/})
[
  {
    _id: ObjectId("638c17a6f402e4200e9a9bef"),
    ProdID: 821,
    size: 'XXL/XXXL',
    material: 'fleece',
    gender: 'Male'
  },
  {
    _id: ObjectId("638c17a6f402e4200e9a9bf2"),
    ProdID: 830,
    size: 'XXL/XXXL',
    material: 'courdrouy',
    gender: 'Unisex'
  },
  {
    _id: ObjectId("638c17a6f402e4200e9a9bf7"),
    ProdID: 846,
    size: 'M/L/XL/XXL',
    material: 'silk',
    gender: 'Female'
  }
]
```

Analytical Purpose: This will give insights on the brands that offer clothes in plus size that will be helpful in categorizing the brands within the plus size section of the website.

3. To find the how many items each customer has ordered.

```
Ecommerce> db.orders.aggregate([{$group:{_id:"$custID", TotalProductsOrdered:{$sum:1}}}]);
[
  { _id: 28, TotalProductsOrdered: 2 },
  { _id: 34, TotalProductsOrdered: 1 },
  { _id: 71, TotalProductsOrdered: 1 },
  { _id: 56, TotalProductsOrdered: 1 },
  { _id: 80, TotalProductsOrdered: 1 },
  { _id: 43, TotalProductsOrdered: 2 },
  { _id: 69, TotalProductsOrdered: 5 },
  { _id: 65, TotalProductsOrdered: 1 },
  { _id: 60, TotalProductsOrdered: 1 },
  { _id: 68, TotalProductsOrdered: 1 },
  { _id: 86, TotalProductsOrdered: 3 },
  { _id: 21, TotalProductsOrdered: 1 },
  { _id: 57, TotalProductsOrdered: 1 },
  { _id: 32, TotalProductsOrdered: 1 },
  { _id: 44, TotalProductsOrdered: 3 }
]
```

Analytical Purpose: Helpful in understanding which customers shops frequently. Push notifications with discounts and sale prices can be sent to these customers to increase the sales.

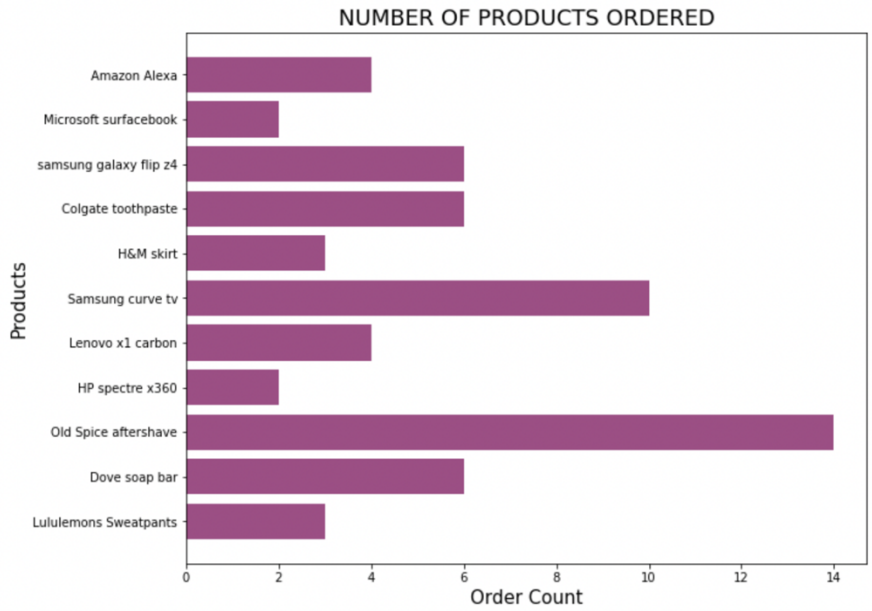
## V. Database Access via Python:

```
import mysql.connector
from mysql.connector import Error
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import matplotlib
```

```
connection = mysql.connector.connect(host='localhost',
                                     database='ecommerce',
                                     user='root',
                                     password='admin'
                                     )
```

### 1. Most sold products

```
#plt.hist(x=productdf['ProdName'], by=productdf['Order Count'] ,color='pink', edgecolor='black')
plt.figure(figsize=(10,8))
plt.barh(productdf['ProdName'], productdf['Order Count'],color='#9b4f85')
plt.title('NUMBER OF PRODUCTS ORDERED',fontsize=18)
plt.ylabel('Products',fontsize=15)
plt.xlabel('Order Count',fontsize=15)
```




Analytic Purpose: Helps in finding the best selling products. These best-selling products can be showed on the top of the search results to the customer which will help customer find good products without wasting much of their time.

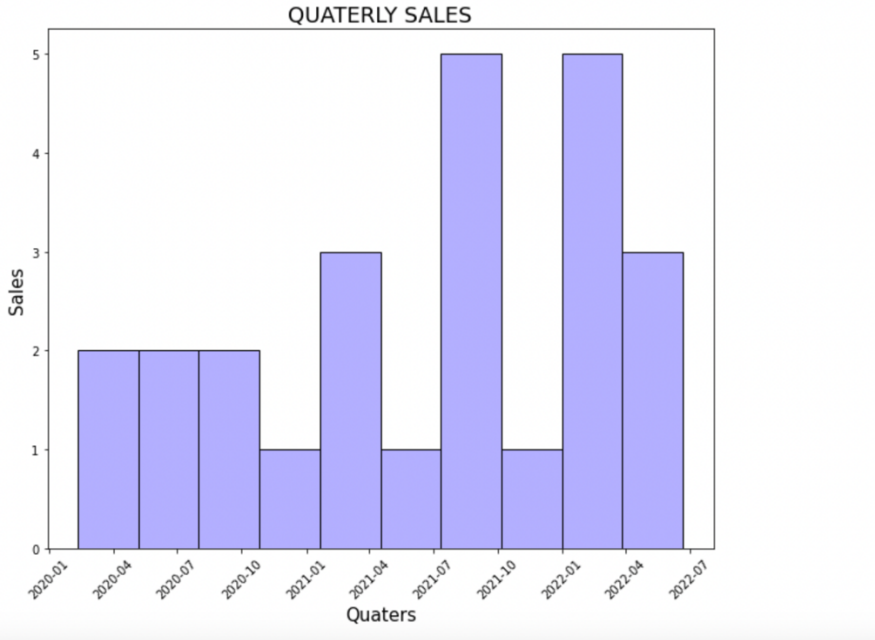
## 2. To project the Quarterly sales from 2020 to 2022.

```
select_sql_query= "SELECT * FROM orders;"
cursor = connection.cursor()
cursor.execute(select_sql_query)
records = cursor.fetchall()

df= pd.DataFrame(records)
df
```

```
plt.figure(figsize=(10,8))
plt.hist(df[1], edgecolor='black', color="#b3afff")
plt.title('QUATERLY SALES',fontsize=18)
plt.xticks(rotation=45)
plt.ylabel('Sales',fontsize=15)
plt.xlabel('Quaters',fontsize=15)
```

```
Text(0.5, 0, 'Quaters')
```




Analytic Purpose: This is useful in projecting the sales of the upcoming quarters so that the teams can accordingly plan for their future deliverables.

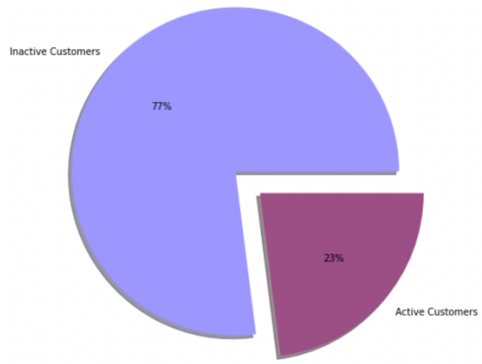
## 3. Identifying Active and Inactive Customers

```
select_sql_query="SELECT (SELECT COUNT(c.custid) FROM customer c WHERE c.custid NOT IN (SELECT DISTINCT o.custid FROM orders o)) AS InactiveCustomers, (SELECT count(c.cust
cursor = connection.cursor()
cursor.execute(select_sql_query)
records = cursor.fetchall()
df= pd.DataFrame(records, columns = ['Inactive','Active']).T
df
```

| | 0 |
|---|---|
| Inactive | 77 |
| Active | 23 |

```
mylabels = ["Inactive Customers", "Active Customers"]
myexplode = [0.2, 0]
mycolors = ["#9c97ff","#9b4f85"]

plt.figure(figsize=(8,8))
plt.pie(df[0], labels = mylabels, explode = myexplode, shadow = True, colors = mycolors, autopct='%.0f%%')
plt.show()
```




Analytic Purpose: We can collect the information about the inactive customers and send them push notifications regarding offers or price drops or even send discounts to encourage them to place orders.

## VI. Summary and recommendation

Customers no longer need to physically visit a store to make purchases, thanks to the emergence of the internet. Nowadays, consumers may shop online from the convenience of their homes. The market for online shopping has grown as a result. As a result, massive amounts of data are generated and stored every second. Due to the increasing number of people using electronic commerce, modern information technology is needed to handle this e-commerce system. The e-commerce industry relies significantly on databases; hence, effective data management is essential. In the current context, an e-commerce company's profitability is directly connected to how well it has optimized its database. Additionally, it involves seeing patterns and drawing pertinent inferences from the data.

Future Scope:

- The model can be extended with the usage of ML algorithms to significantly improve personalized recommendation of products to the customers.
- The smart image recognition system can be implemented with the help of image recognition techniques for searching of the products.