ELECTRONIC RETAIL AND EMI MANAGEMENT DATABASE

SQL PROJECT

Amrita Chinnam
02133164
20-Dec-2023

TABLE OF CONTENTS

LIST OF FIGURES

EXECUTIVE SUMMARY

The primary objective of our database project is to establish an efficient and comprehensive system for managing electronic retail operations along with EMI management. The key goals include the management of customer data, order tracking, product inventory, payment processing, EMI management. The system allows customers to place an order, select products, choose payment methods while providing real time order tracking. With inventory control, payment options tracking, the database contributes enhanced customer experience. The system provides customer behaviour, trends, financial usage, inventory management. Ultimately, the designed database system serves as a pivotal tool for the organization's electronic retail operations, promoting efficiency, accuracy, and customer satisfaction.

PROJECT DESCRIPTION

Apple, Inc. has been used for this project. Apple, Inc. is a widely recognised electronics gadgets, and accessories retail company. Considering the high influx of orders for their products due to their demand, our primary objective is to develop a system that seamlessly captures and processes order details, ensuring efficient inventory management, accurate payment processing, and usage patterns of different payment transactions. The database system encompasses features to tailor inventory planning, and financial strategies to support the various preferred payment methods of the diverse customer base. The payment methods include the full payment, Equated Monthly Installments (EMI). The targeted users include customer service representatives, inventory holders, and the finance team. By addressing these objectives, our database system aims to significantly enhance Apple, Inc.’s order fulfillment efficiency, elevate customer satisfaction, and contribute to overall operational excellence, as well as track the EMI payments and analyse the payment data to thrive in electronic retail industry.

DATABASE DESIGN: ER DIAGRAM

ATTRIBUTES, DATA TYPES, AND CONSTRAINTS

There are six tables in the database. All the tables have the following attributes, data types, and constraints as mentioned below.

1. **Customers_Online Table:** Includes all the customers data.

- CustomerID (Primary Key, INT, Not Null, Auto_Increment)
- First_Name (VARCHAR(255), Not Null)
- Last_Name (VARCHAR(255), Not Null)
- Gender (CHAR(1), Not Null)
- Address (VARCHAR(255), Not Null)
- Postal_Code (VARCHAR(50), Not Null)
- Country (VARCHAR(255), Not Null)
- Phone (VARCHAR(12))
- FULLNAME (UNIQUE Constraint on Last_Name, First_Name)

The full name constraint ensures uniqueness of combination of last and first name. This ensures that no two customers can have the same combination of last and first name.

2. **Orders_Online Table:** Includes all the orders data.

- OrderID (Primary Key, INT, Not Null, Auto Increment)
- OrderPlaced_Date (DATE, Not Null)
- CustomerID (Foreign Key, INT) (ON UPDATE CASCADE)
- Amount (DECIMAL (20,2))

3. **Products_Online Table:** Includes all the products available online.

- ProductID (Primary Key, INT, Not Null, Auto Increment)

- Product_Description (VARCHAR(255), UNIQUE, Not Null)
- Stock (INT, Not Null)
- Category (VARCHAR(255), Not Null)
- Price (DECIMAL(10,2), Not Null)

Product description has a unique constraint as every product is unique in the database.

4. **OrdersAndProducts Table:** Bridges the orders and products.

- OPID (Primary Key, INT, Not Null, Auto Increment)
- Quantity (DECIMAL(10), Not Null)
- OrderID (Foreign Key, INT, Not Null) (ON UPDATE CASCADE)
- ProductID (Foreign Key, INT, Not Null) (ON UPDATE CASCADE)

5. **Payments Table:** Includes all the payment details.

- PaymentID (Primary Key, INT, Not Null, Auto Increment)
- Payment_Date ((DATE), Not Null)
- Payment_Method (VARCHAR(255), Not Null)
- Payment_Option (VARCHAR(255), Not Null)
- OrderID (Foreign Key, INT, Not Null) (ON UPDATE CASCADE)

6. **EMI/Equated Monthly Installment Table:** Includes all the data required to analyse the EMI payments.

- EMI_Number_ID (Primary Key, INT, Not Null, Auto Increment)
- Tenure (INT)
- Monthly_Installment_Amount (DECIMAL(20,2))
- EMI_Completed (CHAR(1), Not Null, Default ‘N’)
- Payment_ID (Foreign Key, Int, Not Null) (ON UPDATE CASCADE)

Default N has been added to EMI_Completed as initially any EMI/Equated Monthly Installment is not completed by default.

On update cascade is a constraint added in all foreign keys to mention that every change in primary key/parent table to occur in foreign key/child table.

ENTITY RELATIONSHIPS

The below are the entity relationships between tables.

1. **Customers_Online Table:**

One-to-Many with *Orders_Online*: A customer can place multiple orders.

2. **Orders_Online Table:**

One-to-Many with *OrdersAndProducts:* One order can include multiple products.

One-to-One with *Payments*: Each order has a corresponding payment.

3. **Products_Online Table:**

One-to-Many with *OrdersAndProducts:* One product can be part of multiple orders.

4. **Payments Table:**

One-to-One with *EMI*: Each payment can have an associated EMI.

The ER Diagram for the above created database has been provided in the Appendix.

NORMALIZATION

For Normalization/BCNF - Boyce-Codd Normal Form, we need to first check the tables and their dependencies.

Customers_Online (CustomerID **PK**, First_Name, Last_Name, Gender, Address, Postal_Code, Country, Phone, **FULLNAME UNIQUE** (Last_Name, First_Name))

Orders_Online (OrderID **PK**, OrderPlaced_Date, Amount, CustomerID **FK**)

WHERE Orders_Online.CustomerID MUST EXIST IN Customers_Online.CustomerID

Products_Online (ProductID **PK**, Product_Description **UNIQUE**, Stock, Category, Price)

OrdersAndProducts (OPID **PK** Auto_Increment**,** Quantity, OrderID **FK**, ProductID **FK**)

WHERE OrdersAndProducts.OrderID = Orders_Online.OrderID

WHERE OrdersAndProducts.ProductID = Products_Online.ProductID

Payments (PaymentID **PK**, Payment_Date, Payment_Method, Payment_Option, OrderID **FK**)

WHERE Payments.OrderID MUST EXIST IN Orders_Online.OrderID

EMI (EMI_Number_ID **PK**, Tenure, Monthly_Installment_Amount, EMI_Completed **DEFAULT 'N'**, PaymentID **FK**)

WHERE EMI.PaymentID MUST EXIST IN Payments.PaymentID

The determinants in tables are candidate keys. There are no dependencies between the attributes in tables and all referential integrity has been maintained among the tables. Hence, all the tables are in BCNF, and database implementation can be further continued.

DATABASE IMPLEMENTATION

DDL

As the database is in BCNF, the database '**database_records'** is created. This database consists of six tables including **customers_online** (for customers data), **orders_online** (for orders made online), **products_online** (for products available online), **ordersandproducts**

(for all the orders and products), **payments** (for all types of payments on orders), **EMI** (for all EMI type of transactions).

'**Create database, use database**' function is used to create and use the database consisting of the tables. '**Create table'** function is used to create the table along with the constraints in the database.

TRIGGERS

1. *Amount* column in **OrdersAndProducts** table

OrdersAndProducts table consists of all orderID and productID. To include the total amount of all orders based on their products, a trigger has been created.

```
DELIMITER //
CREATE TRIGGER Update_Order_Amount
AFTER INSERT ON OrdersAndProducts
FOR EACH ROW
BEGIN
    UPDATE Orders_Online SET Amount = New.Quantity * (SELECT price FROM Products_Online WHERE ProductId = New.ProductId )
    WHERE OrderId = New.OrderId;
END;
//
DELIMITER ;
```

Figure 1 Amount column in OrdersAndProducts table.

This trigger automatically populates the *Amount* column in the **OrdersAndProducts** table to correspond the order and product. The *quantity* is taken from the **orders_online** table, while the *price* of the products is taken from the **products_online** table to create the *Amount* column in **OrdersAndProducts** table. This eliminates the need for manual calculations.

2. **EMI** Table

A customer chooses to make payments in the form of full payment or monthly Installment payments. Since the **EMI** table from the database depends on the *payments method* chosen from the **payments** table, we have created a trigger in the code.

```
DELIMITER //
CREATE TRIGGER Insert_EMI_Record
AFTER INSERT ON Payments
FOR EACH ROW
BEGIN
IF NEW.Payment_Method = 'EMI' THEN
    INSERT INTO EMI (Tenure, Monthly_Installment_Amount, EMI_Completed, PaymentID) VALUES(NULL , NULL,'N',New.PaymentID);
END IF;
END;
//
DELIMITER ;
```

Figure 2 EMI Table.

The following is the trigger that inserts a column into **EMI** table stating that a new payment of EMI method has been initiated by a customer. When a new row is inserted into **payments** table, the trigger is activated. The trigger is used to update the **EMI** table based on the **payment method** selected by the customer.

3. **Monthly Installment Amount** in **EMI** table

The trigger is designed to automatically update the *Monthly_Installment_Amount* column in the **EMI** table. This trigger aims to recalculate and adjust the monthly installment amount based on changes in the Tenure and associated payment details.

```
DELIMITER //
CREATE TRIGGER Update_EMI_Monthly_installment_Amount
BEFORE UPDATE ON EMI
FOR EACH ROW
BEGIN
IF NEW.Tenure IS NOT NULL THEN
    SET NEW.Monthly_Installment_Amount = (SELECT oo.Amount FROM Orders_Online as oo, Payments as p, EMI as e
    WHERE e.PaymentID = p.PaymentID AND p.OrderID = oo.OrderID AND e.PaymentID = New.PaymentID)/ NEW.Tenure;
END IF;
END;
//
DELIMITER ;
```

Figure 3 Monthly Installment Amount in EMI table.

If the *Tenure* in the updated row is not null, then the trigger calculates the new *Monthly_Installment_Amount* by dividing the total order amount (from the **Orders_Online**) by the *Tenure* value in the updated row. The result is then set as the new value for

Monthly_Installment_Amount in the **EMI** table. This helps in adjusting the installment amount based on the tenure chosen for payment, providing an up to date of the payment schedule.

After the creation of tables, and triggers, values are inserted into the tables using the '**INSERT INTO (TABLE) VALUES ()'** function.

QUERY

1. Query for Total Amount spent by each customer -

This query helps in finding the total amount spent by each customer for the products available. In this output we see that Christopher Steve has spent the highest amount of 4,495 USD, while Faisa Ali has spent the lowest amount of 1,096 USD in purchases.

CustomerID	First_Name	Last_Name	TotalAmount_in_USD
4	Christopher	Steve	4,495.00
1	Annie	Doe	3,497.00
3	Bella	Johnson	2,697.00
2	Angel	Smith	2,635.06
5	Dove	Stem	1,528.95
6	Faisa	Ali	1,096.00

Figure 4 Query for Total Amount spent on each customer.

2. Query for the most sold product in the company -

This query shows the most sold product and its category. The most sold product is the iPhone 15 256GB from the iPhones category with a highest total quantity sold of 7. While, Laptops, and Accessories have been sold less in quantity. This shows the demand for the category of products that the company can focus on for profits.

ProductID	Product_Description	TotalQuantity	Category	Price
2	iPhone 15 256GB	7	iPhones	899.00
5	Airpods Pro 2nd generation	6	Airpods	249.00
7	iPad Pro	4	iPad	899.00
4	Apple Watch Series 9	3	Watches	499.00
1	iPhone 15 Pro 256GB	2	iPhones	1499.00
9	iPhone 15 Pro Silicone Case	2	Accessories	49.00
3	MacBook Pro 14 Inch	1	Laptops	1599.00
6	Waterproof Case for Airpods	1	Accessories	34.95
8	Apple Pencil 2nd generation	1	Accessories	137.06
10	Apple Watch Sport Band	1	Accessories	49.00

Figure 5 Query for most sold product.

3. Query for the most chosen payment method -

This query shows the most chosen payment method by customers. Here, we see the highest payment method chosen is the EMI with a value of 12,517.06. This shows that the customers prefer this payment method widely.

Payment_Method	OptionCount	TotalAmount
EMI	10	12517.06
Full Payment	3	3431.95

Figure 6 Query for most chosen payment method.

4. Query for payment method chosen for category -

The most chosen payment method by customers is an EMI for mostly all categories of products except Laptops. Some Accessories, iPads are paid in full amounts. This helps in understanding the category of products that are preferred with EMI option.

Category	Payment_Method	ProductIds	OrderIds
Accessories	EMI	9,9,10,8	10,11,11,13
Accessories	Full Payment	6	8
Airpods	EMI	5	9
iPad	EMI	7	3
iPad	Full Payment	7	5
iPhones	EMI	1,2,2,2	1,3,6,7
Laptops	Full Payment	3	4
Watches	EMI	4,4	2,12

Figure 7 Query for payment method chosen as per category.

5. Query to alter and update EMI table -

Altering *EMI* table for adding columns (Total_amount, Tenure_Completed, Tenure_remaining) to get overall picture of the details on EMI payments made.

```
/* Alter EMI table and add tenure completed and tenure remaining column */
ALTER TABLE EMI
ADD COLUMN Tenure_Completed DECIMAL(20,2);
UPDATE EMI
SET Tenure_Completed = 2;

ALTER TABLE EMI
ADD COLUMN Tenure_Remaining DECIMAL(20,2);
UPDATE EMI
SET Tenure_Remaining = Tenure - Tenure_Completed;

/* Updating Tenure remaining in EMI table */
UPDATE EMI
SET EMI_Completed = CASE WHEN Tenure_Remaining = 0 THEN 'Y' ELSE 'N' END;
```

Figure 8 Altering and Updating EMI table.

Final output of EMI table –

EMI_Number_ID	Tenure	Monthly_Installment_Amount	EMI_Completed	PaymentID	Total_Amount	Tenure_Completed	Tenure_Remaining
1	24	124.92	N	1	2998.08	2.00	22.00
2	36	13.86	N	2	498.96	2.00	34.00
3	24	37.46	N	3	899.04	2.00	22.00
4	36	24.97	N	6	898.92	2.00	34.00
5	24	187.29	N	7	4494.96	2.00	22.00
6	24	62.25	N	9	1494.00	2.00	22.00
7	2	24.50	Y	10	49.00	2.00	0.00
8	2	24.50	Y	11	49.00	2.00	0.00
9	36	27.72	N	12	997.92	2.00	34.00
10	3	45.69	N	13	137.07	2.00	1.00
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

6. Query for list of all EMI payments –

Payment_Date	Payment_Method	Tenure	Monthly_Installment_Amount	Total_Amount
2023-01-01	EMI	24	124.92	2998.08
2023-02-01	EMI	36	13.86	498.96
2023-02-01	EMI	24	37.46	899.04
2023-11-28	EMI	36	24.97	898.92
2023-03-21	EMI	24	187.29	4494.96
2023-11-09	EMI	24	62.25	1494.00
2023-05-11	EMI	2	24.50	49.00
2023-08-08	EMI	2	24.50	49.00
2023-08-08	EMI	36	27.72	997.92
2023-03-01	EMI	3	45.69	137.07

Figure 9 Query for list of all EMI payments.

7. Query for list of all completed EMIs for the products purchased –

The customer Faisa Ali has completed her EMIs for the product iPhone 15 Pro Silicone Case, and Apple watch sport band.

CustomerID	First_Name	Last_Name	EMI_Completed	Product_Description
6	Faisa	Ali	Y	iPhone 15 Pro Silicone Case
6	Faisa	Ali	Y	iPhone 15 Pro Silicone Case
6	Faisa	Ali	Y	Apple Watch Sport Band

Figure 10 Query for list of completed EMIs.

8. Query for list of active EMIs –

The top 3 Active EMIs have 22 months tenure remaining for the products iPhone 15, iPhone 15 Pro, Airpods pro 2nd generation. It is important that the finance department keeps a track of these active EMIs.

CustomerID	First_Name	Last_Name	Product_Description	EMI_Completed	Tenure_Remaining	Monthly_Installment_Amount	Total_Amount
4	Christopher	Steve	iPhone 15 256GB	N	22.00	187.29	4494.96
1	Annie	Doe	iPhone 15 Pro 256GB	N	22.00	124.92	2998.08
5	Dove	Stem	Airpods Pro 2nd generation	N	22.00	62.25	1494.00

Figure 11 Query for list of active EMIs.

9. Query for list of repeat customers and their payment method chosen –

We see that eventually the payment method chosen by every repeat customer is a shift to EMI payment method irrespective of the amount on the order. Hence, the company can make focus on providing more EMI options to every new and existing customer to create better customer satisfaction.

First_Name	Last_Name	Payment_Method	Amount	OrderPlaced_Date
Angel	Smith	Full Payment	1599.00	2023-02-01
Angel	Smith	EMI	137.06	2023-03-01
Angel	Smith	EMI	899.00	2023-04-01
Annie	Doe	EMI	2998.00	2023-01-01
Annie	Doe	EMI	499.00	2023-02-01
Bella	Johnson	Full Payment	1798.00	2023-01-15
Bella	Johnson	EMI	899.00	2023-11-28
Christopher	Steve	EMI	4495.00	2023-03-21
Dove	Stem	EMI	1494.00	2023-11-09
Dove	Stem	Full Payment	34.95	2023-12-31
Faisa	Ali	EMI	49.00	2023-05-11
Faisa	Ali	EMI	49.00	2023-08-08
Faisa	Ali	EMI	998.00	2023-08-08

Figure 12 Query for list of repeat customers and their payment method.

CONCLUSION

Considering the wide take on EMI options as the ease payment option for customers. The company can promote EMI options available to customers during the checkout process. The company could expand EMI offerings by collaborating with financial institutions. They could also focus on optimising the electronic purchase experience for customers opting for EMI option. It is also important to strategically plan the inventory and sales forecasts based on the increased usage of EMI payment method. Finally, the company could focus on monitoring customer behaviour regarding payment methods to identify evolving trends and adapt strategies.

APPENDIX

1. The below is the ER Diagram created in MYSQL for the database – ‘database_records’.

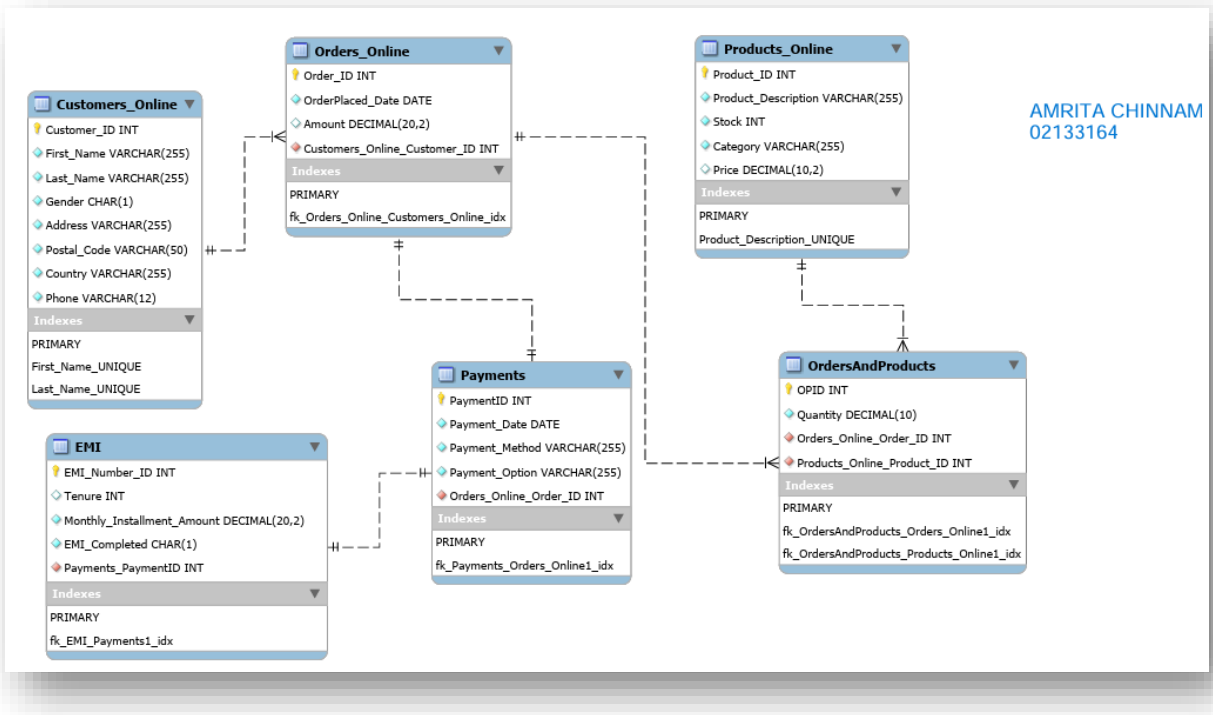

Figure 13 Entity Relationship Diagram - Database_records.

REFERENCES

1. Malte Krueger, KM, 2001 August, The Future of M-payments–Business Options and Policy Issues.

Link - Potential involvement of telcos in the payment system (psu.edu)

2. Gustavo du Mortier, MG, 2023 April, Ecommerce Database Design: ER Diagram for Online Shopping.

Link - Ecommerce Database Design: ER Diagram for Online Shopping | Vertabelo Database Modeler