Mookit: Move-In Essentials Inventory and Order Management System

Meet our Team! - Group 5

Dhanvardini Rajendran

Kunal Kale

Kalyan Venkata Swamy Karnati

Sai Kumar Katakam

Kalyan Satwik Adabala

Mookit's Mission: Simplifying and Sustaining

1

Customer Focus

Mookit prioritizes customer satisfaction by offering affordable, adaptable solutions to cater to a wide range of needs.

2

Environmental Responsibility

Through our furniture buyback program, Mookit promotes sustainability by reducing waste and extending the lifespan of pre-owned furniture.

3

Market Expansion

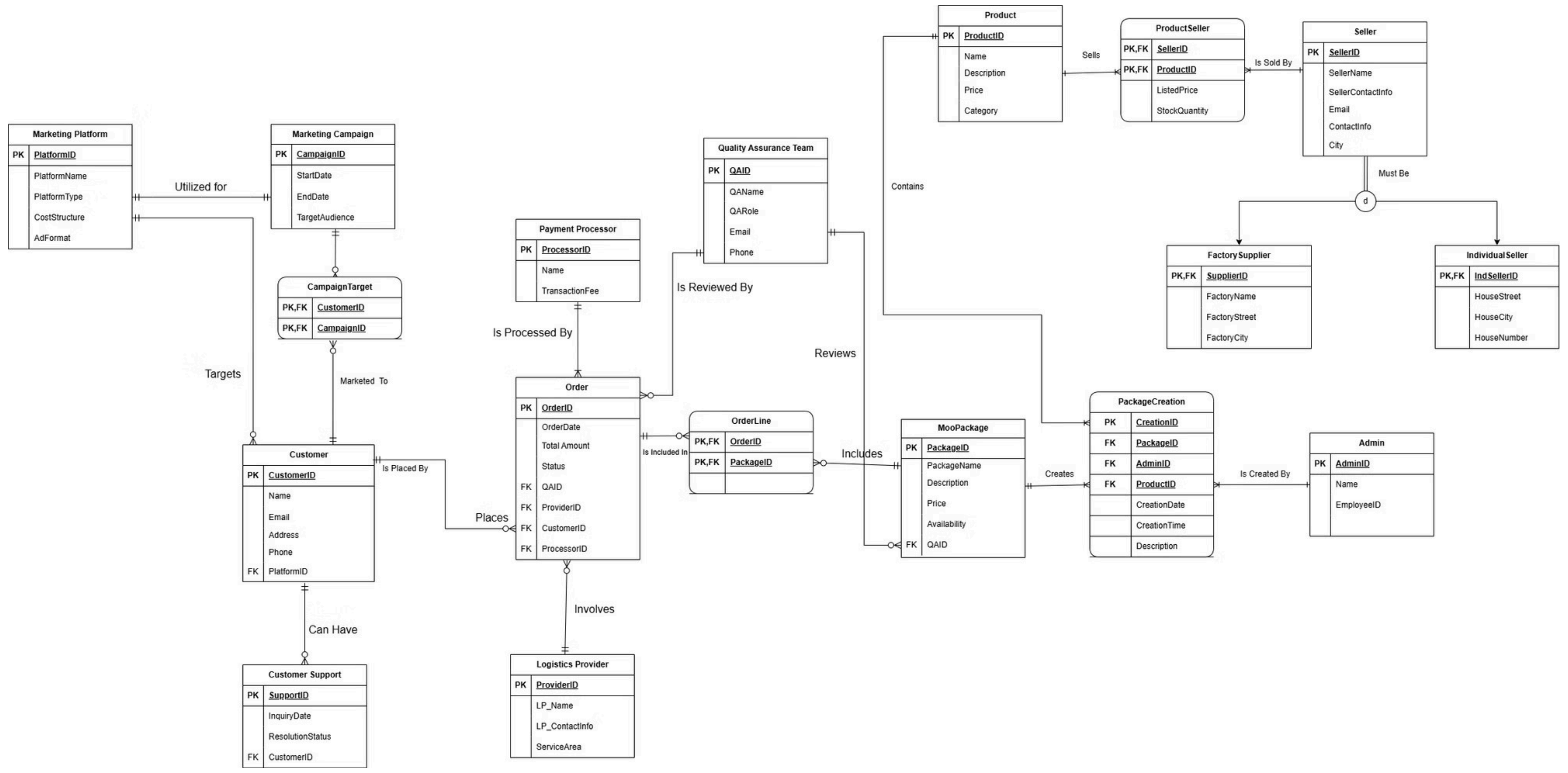
Mookit aims to expand its reach to serve a larger demographic of movers, including families and young professionals.

4

Operational Efficiency

Mookit streamlines logistics and procurement processes to maintain competitive pricing and ensure quality service.

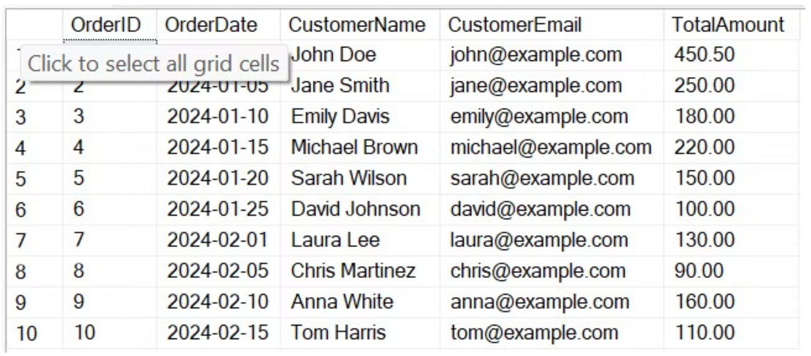
ERD Diagram



Marketing Platform
PK PlatformID
PlatformName
PlatformType
CostStructure
AdFormat
Utilized for
Marketing Campaign
PK CampaignID
StartDate
EndDate
TargetAudience
CampaignTarget
PK,FK CustomerID
PK,FK CampaignID
Marketed To
Targets
Customer
PK CustomerID
Name
Email
Address
Phone
FK PlatformID
Is Placed By
Places
Can Have
Customer Support
PK SupportID
InquiryDate
ResolutionStatus
FK CustomerID
Payment Processor
PK ProcessorID
Name
TransactionFee
Is Processed By
Order
PK OrderID
OrderDate
Total Amount
Status
FK QAID
FK ProviderID
FK CustomerID
FK ProcessorID
Involves
Logistics Provider
PK ProviderID
LP_Name
LP_ContactInfo
ServiceArea
Quality Assurance Team
PK QAID
QAName
QARole
Email
Phone
Is Reviewed By
Reviews
OrderLine
PK,FK OrderID
PK,FK PackageID
Is Included In
Includes
MooPackage
PK PackageID
PackageName
Description
Price
Availability
FK QAID
Creates
PackageCreation
PK CreationID
FK PackageID
FK AdminID
FK ProductID
CreationDate
CreationTime
Description
Is Created By
Admin
PK AdminID
Name
EmployeeID
Contains
Product
PK ProductID
Name
Description
Price
Category
Sells
ProductSeller
PK,FK SellerID
PK,FK ProductID
ListedPrice
StockQuantity
Is Sold By
Seller
PK SellerID
SellerName
SellerContactInfo
Email
ContactInfo
City
Must Be
d
Factory Supplier
PK,FK SupplierID
FactoryName
FactoryStreet
FactoryCity
Individual Seller
PK,FK IndSellerID
HouseStreet
HouseCity
HouseNumber

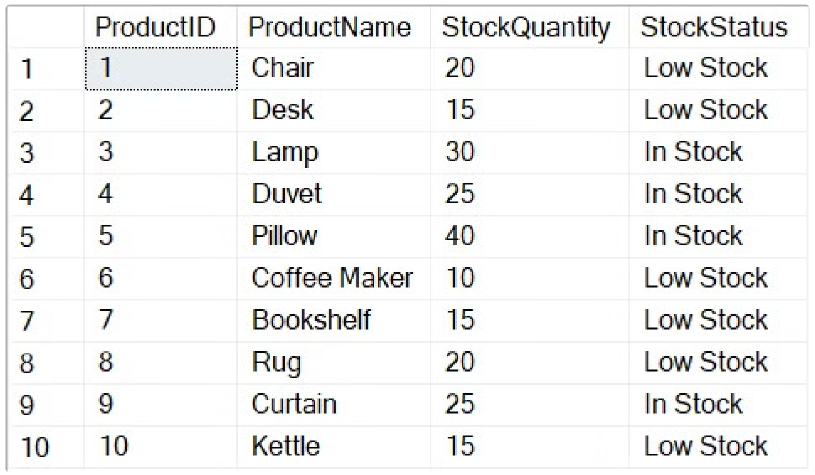
VIEWS

1. The vw_OrderDetailsWithCustomer view efficiently joins the Order and Customer tables to present order details alongside customer information using the CustomerID.
2. The vw_InventoryStatus view aggregates Product and ProductSeller data, categorizing stock levels with a CASE statement based on StockQuantity.
3. Both views optimize data retrieval by encapsulating complex joins and logic, reducing query complexity for reporting and analysis.


Click to select all grid cells

	OrderID	OrderDate	CustomerName	CustomerEmail	TotalAmount
1	[illegible]	[illegible]	John Doe	john@example.com	450.50
2	2	2024-01-05	Jane Smith	jane@example.com	250.00
3	3	2024-01-10	Emily Davis	emily@example.com	180.00
4	4	2024-01-15	Michael Brown	michael@example.com	220.00
5	5	2024-01-20	Sarah Wilson	sarah@example.com	150.00
6	6	2024-01-25	David Johnson	david@example.com	100.00
7	7	2024-02-01	Laura Lee	laura@example.com	130.00
8	8	2024-02-05	Chris Martinez	chris@example.com	90.00
9	9	2024-02-10	Anna White	anna@example.com	160.00
10	10	2024-02-15	Tom Harris	tom@example.com	110.00

Order Details with Customer Info

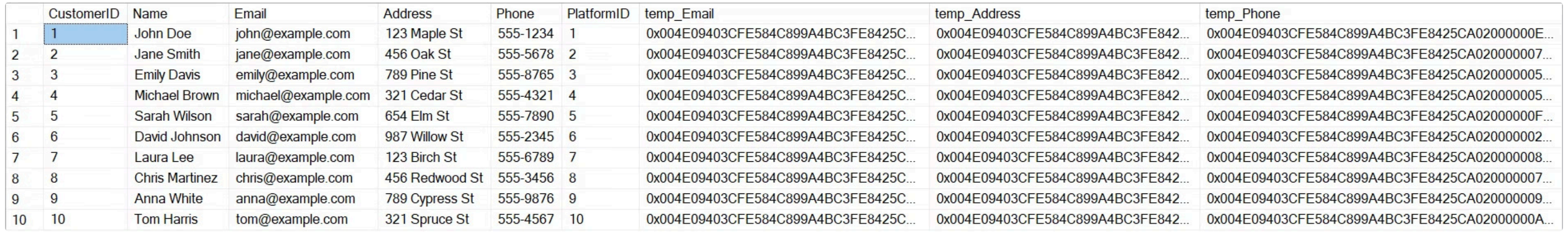


	ProductID	ProductName	StockQuantity	StockStatus
1	1	Chair	20	Low Stock
2	2	Desk	15	Low Stock
3	3	Lamp	30	In Stock
4	4	Duvet	25	In Stock
5	5	Pillow	40	In Stock
6	6	Coffee Maker	10	Low Stock
7	7	Bookshelf	15	Low Stock
8	8	Rug	20	Low Stock
9	9	Curtain	25	In Stock
10	10	Kettle	15	Low Stock

Inventory Status Report

Made with Gamma

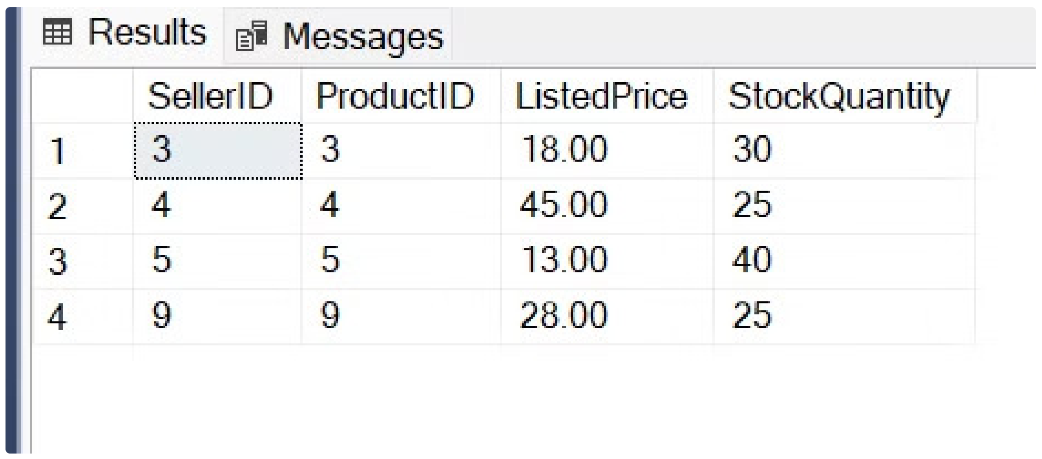
ENCRYPTION



	CustomerID	Name	Email	Address	Phone	PlatformID	temp_Email	temp_Address	temp_Phone
1	1	John Doe	john@example.com	123 Maple St	555-1234	1	0x004E09403CFE584C899A4BC3FE8425C...	0x004E09403CFE584C899A4BC3FE842...	0x004E09403CFE584C899A4BC3FE8425CA02000000E...
2	2	Jane Smith	jane@example.com	456 Oak St	555-5678	2	0x004E09403CFE584C899A4BC3FE8425C...	0x004E09403CFE584C899A4BC3FE842...	0x004E09403CFE584C899A4BC3FE8425CA020000007...
3	3	Emily Davis	emily@example.com	789 Pine St	555-8765	3	0x004E09403CFE584C899A4BC3FE8425C...	0x004E09403CFE584C899A4BC3FE842...	0x004E09403CFE584C899A4BC3FE8425CA020000005...
4	4	Michael Brown	michael@example.com	321 Cedar St	555-4321	4	0x004E09403CFE584C899A4BC3FE8425C...	0x004E09403CFE584C899A4BC3FE842...	0x004E09403CFE584C899A4BC3FE8425CA020000005...
5	5	Sarah Wilson	sarah@example.com	654 Elm St	555-7890	5	0x004E09403CFE584C899A4BC3FE8425C...	0x004E09403CFE584C899A4BC3FE842...	0x004E09403CFE584C899A4BC3FE8425CA02000000F...
6	6	David Johnson	david@example.com	987 Willow St	555-2345	6	0x004E09403CFE584C899A4BC3FE8425C...	0x004E09403CFE584C899A4BC3FE842...	0x004E09403CFE584C899A4BC3FE8425CA020000002...
7	7	Laura Lee	laura@example.com	123 Birch St	555-6789	7	0x004E09403CFE584C899A4BC3FE8425C...	0x004E09403CFE584C899A4BC3FE842...	0x004E09403CFE584C899A4BC3FE8425CA020000008...
8	8	Chris Martinez	chris@example.com	456 Redwood St	555-3456	8	0x004E09403CFE584C899A4BC3FE8425C...	0x004E09403CFE584C899A4BC3FE842...	0x004E09403CFE584C899A4BC3FE8425CA020000007...
9	9	Anna White	anna@example.com	789 Cypress St	555-9876	9	0x004E09403CFE584C899A4BC3FE8425C...	0x004E09403CFE584C899A4BC3FE842...	0x004E09403CFE584C899A4BC3FE8425CA020000009...
10	10	Tom Harris	tom@example.com	321 Spruce St	555-4567	10	0x004E09403CFE584C899A4BC3FE8425C...	0x004E09403CFE584C899A4BC3FE842...	0x004E09403CFE584C899A4BC3FE8425CA02000000A...

1. A **Master Key** is created and used to secure the encryption hierarchy, followed by a **Certificate** (MookitCert) to encrypt a **Symmetric Key** (Mookit) with the AES_256 algorithm for secure data encryption.
2. Sensitive data in the Customer table (Email, Address, Phone) is converted to VARBINARY(MAX) and encrypted using the ENCRYPTBYKEY function with the symmetric key, ensuring data confidentiality.
3. Data is decrypted by opening the symmetric key, using DECRYPTBYKEY to retrieve the original values, and then casting the decrypted binary data back to its respective data types.

INDEXES, UDFs, TRIGGERS & STORED PROCEDURES

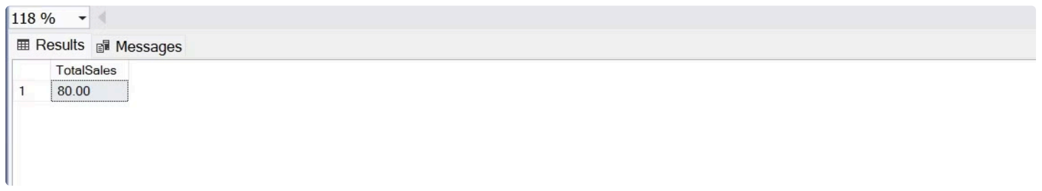


Results | Messages

	SellerID	ProductID	ListedPrice	StockQuantity
1	3	3	18.00	30
2	4	4	45.00	25
3	5	5	13.00	40
4	9	9	28.00	25

On Stock Quantity

1. The IX_ProductStock nonclustered index on the ProductSeller table is dropped to remove any previous indexing on the StockQuantity column.
2. A new **nonclustered index** is created on StockQuantity, improving lookup efficiency for queries that filter by stock levels.
3. The index enhances query performance by reducing the cost of scanning ProductSeller when filtering rows with StockQuantity > 20, leveraging efficient index seeks.



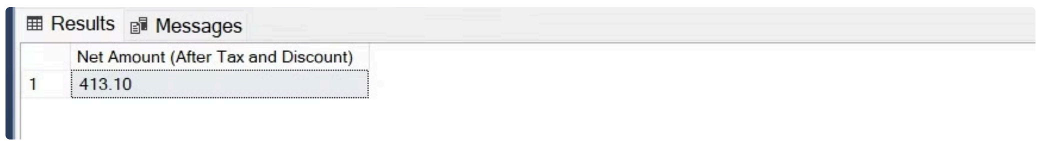
118 %

Results | Messages

	TotalSales
1	80.00

Calculating Total Sales

1. The usp_CalculateTotalSales stored procedure calculates the total sales for a specific provider by summing the TotalAmount from the [Order] table based on the provided ProviderID.
2. It includes input validation to ensure the ProviderID is not NULL, and raises an error if this condition is violated, while returning the total sales in the @TotalSales output parameter.
3. The procedure handles null results by setting @TotalSales to 0 if no sales are found for the given ProviderID, ensuring robustness in the output.

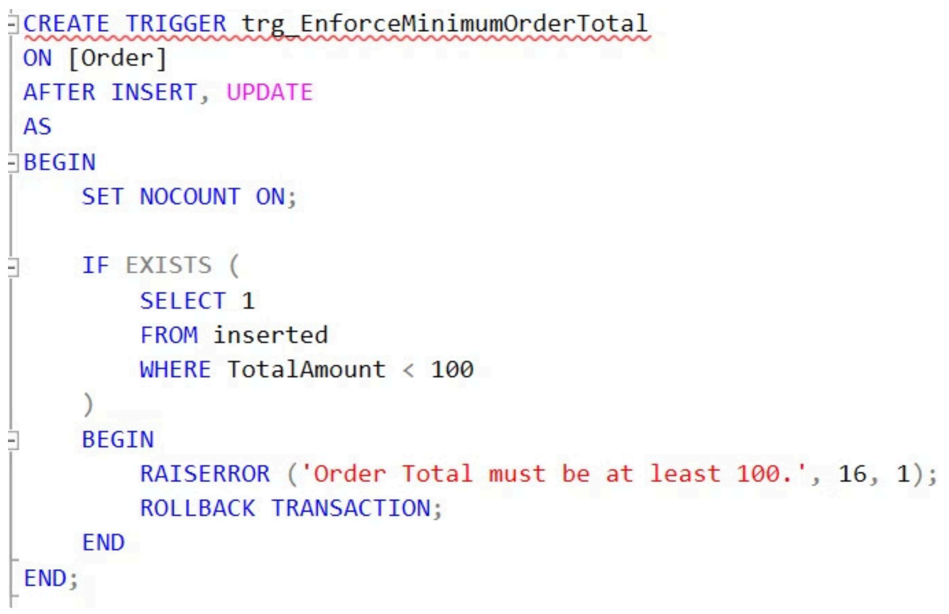


Results | Messages

	Net Amount (After Tax and Discount)
1	413.10

Calculate Net Amount (After Tax and Discount)

1. The CalculateNetAmount function computes the net amount after applying a discount and tax, by first calculating the discount and tax amounts using input parameters @TotalAmount, @DiscountRate, and @TaxRate.
2. The function then returns the final amount, subtracting the discount and adding the tax based on the given rates, using precise DECIMAL(18, 2) calculations.
3. Permissions are granted for the function to be executed by a specified user (S) after ensuring the function's existence and checking the required access permissions.



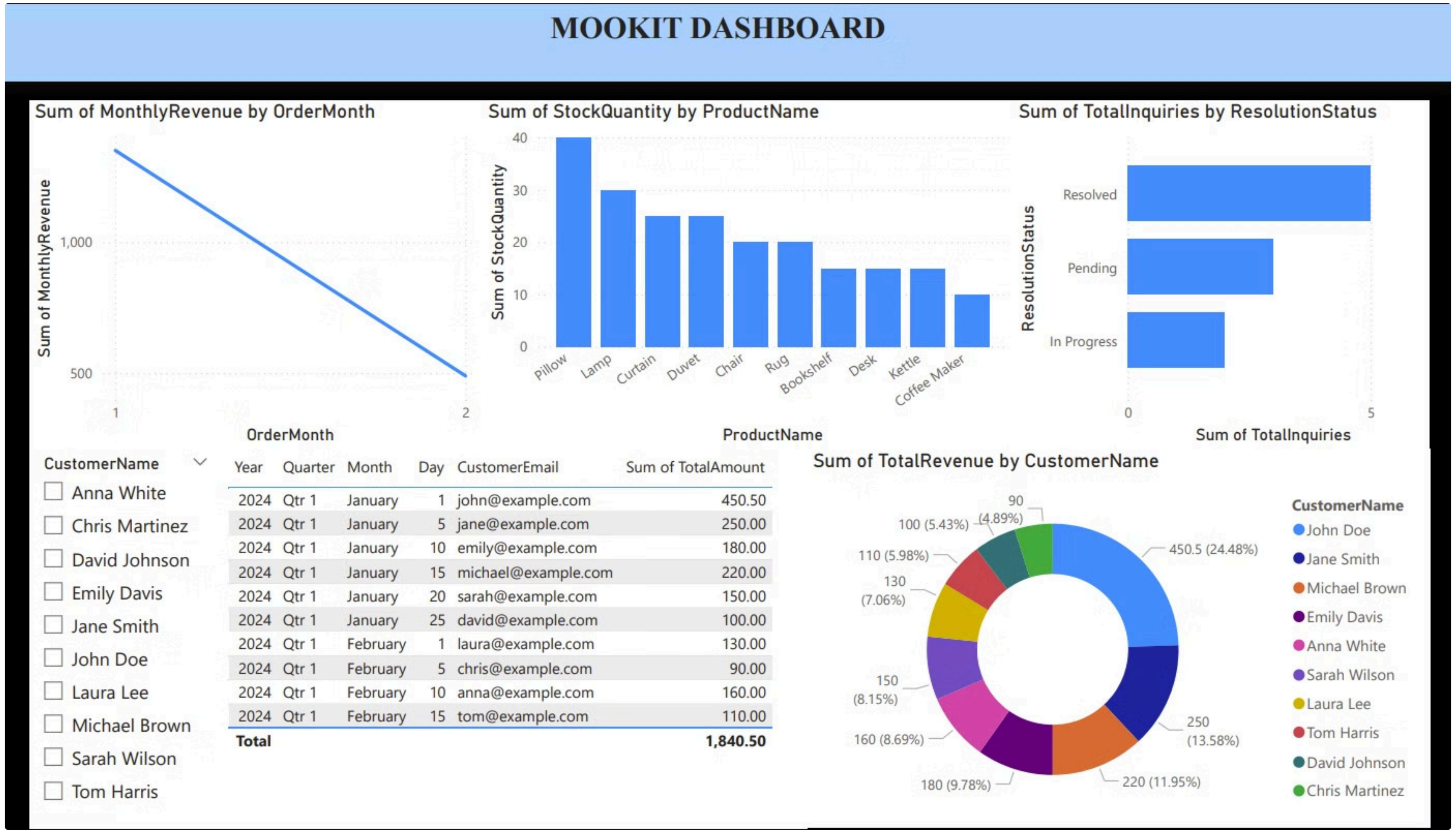
```
CREATE TRIGGER trg_EnforceMinimumOrderTotal
ON [Order]
AFTER INSERT, UPDATE
AS
BEGIN
    SET NOCOUNT ON;

    IF EXISTS (
        SELECT 1
        FROM inserted
        WHERE TotalAmount < 100
    )
    BEGIN
        RAISERROR ('Order Total must be at least 100.', 16, 1);
        ROLLBACK TRANSACTION;
    END
END;
```

Enforce Minimum Order Total

1. The trg_EnforceMinimumOrderTotal trigger is defined to fire **AFTER INSERT** and **AFTER UPDATE** events on the [Order] table to enforce a minimum order total of 100.
2. The trigger checks if the TotalAmount in the inserted pseudo-table is less than 100; if so, it raises an error with RAISERROR and rolls back the transaction.
3. The trigger ensures data integrity by preventing orders with a total amount below 100, thus enforcing business rules on order creation or updates.

PowerBI Dashboard



CustomerName

- ☐ Anna White
- ☐ Chris Martinez
- ☐ David Johnson
- ☐ Emily Davis
- ☐ Jane Smith
- ☐ John Doe
- ☐ Laura Lee
- ☐ Michael Brown
- ☐ Sarah Wilson
- ☐ Tom Harris

Year	Quarter	Month	Day	CustomerEmail	Sum of TotalAmount
2024	Qtr 1	January	1	john@example.com	450.50
2024	Qtr 1	January	5	jane@example.com	250.00
2024	Qtr 1	January	10	emily@example.com	180.00
2024	Qtr 1	January	15	michael@example.com	220.00
2024	Qtr 1	January	20	sarah@example.com	150.00
2024	Qtr 1	January	25	david@example.com	100.00
2024	Qtr 1	February	1	laura@example.com	130.00
2024	Qtr 1	February	5	chris@example.com	90.00
2024	Qtr 1	February	10	anna@example.com	160.00
2024	Qtr 1	February	15	tom@example.com	110.00
Total					**1,840.50**

Made with Gamma

Conclusion: Empowering the Move-In Experience

1

Convenience

Mookit simplifies the move-in process by offering customizable kits of essential items, eliminating the need for extensive shopping and sourcing.

2

Affordability

Mookit's pricing tiers cater to a wide range of budgets, making it accessible to students, young professionals, and families.

3

Sustainability

Mookit promotes environmental responsibility by incentivizing the return and reuse of pre-owned furniture, reducing waste and promoting a circular economy.