NC STATE UNIVERSITY

INTEGRATED DATA MANAGEMENT AND ANALYSIS OF TARGET STORES IN BRAZIL: SALES TRENDS THROUGH MS ACCESS AND POWER BI

ISE 519 Fall 2023 Final Project Report

Prabhaker Manikyam

December 4, 2023

Table of Contents

PROJECT OVERVIEW

Context

Target is a globally renowned brand and a prominent retailer in the United States. Target makes itself a preferred shopping destination by offering outstanding value, inspiration, innovation and an exceptional guest experience that no other retailer can deliver.

This project focuses on the operations of Target in Brazil and provides insightful information about 100,000 orders placed between 2016 and 2018. The dataset offers a comprehensive view of various dimensions including the order status, price, payment and freight performance, customer location, product attributes, and customer reviews.

By analyzing this extensive dataset, it becomes possible to gain valuable insights into Target's operations in Brazil. The information can shed light on various aspects of the business, such as order processing, pricing strategies, payment and shipping efficiency, customer demographics, product characteristics, and customer satisfaction levels.

Objective:

The Project deals with Target Corporation’s data from Brazil and our objective is to create a Secure application on MS Access that allows the sales head to analyze sales trends by region, time period, product categories etc.,along with the privileges to update some details in the database.

About the Dataset:

This publicly available dataset from GitHub is generated by Target Corporation in Brazil between 01/01/2016 to 08/29/2018. This business case has information of approx. 100k orders made at Target in Brazil.
The data is available in 8 csv files:

1.customers.csv
2.sellers.csv
3.order_items.csv
4.geolocation.csv
5.payments.csv
6.reviews.csv
7.orders.csv
8.products.csv

Tables and Contents used for the Application:

1 The **customers.csv** contain following features:

Features	**Description**
customer_id	ID of the consumer who made the purchase
customer_unique_id	Unique ID of the consumer
customer_zip_code_prefix	Zip Code of consumer’s location
customer_city	Name of the City from where order is made
customer_state	State Code from where order is made (Eg. são paulo - SP)

2 The **sellers.csv** contains following features:

Features	**Description**
seller_id	Unique ID of the seller registered
seller_zip_code_prefix	Zip Code of the seller’s location
seller_city	Name of the City of the seller
seller_state	State Code (Eg. são paulo - SP)

3 The **order_items.csv** contain following features:

Features	**Description**
order_id	A Unique ID of order made by the consumers
order_item_id	A Unique ID given to each item ordered in the order
product_id	A Unique ID given to each product available on the site
seller_id	Unique ID of the seller registered in Target
price	Actual price of the products ordered

4 The **payments.csv** contain following features:

Features	**Description**
order_id	A Unique ID of order made by the consumers
payment_sequential	Sequences of the payments made in case of EMI
payment_type	Mode of payment used (Eg. Credit Card)
payment_installments	Number of installments in case of EMI purchase
payment_value	Total amount paid for the purchase order

5 The **orders.csv** contain following features:

Features	**Description**
order_id	A Unique ID of order made by the consumers
customer_id	ID of the consumer who made the purchase
order_purchase_timestamp	Timestamp of the purchase
order_delivered_customer_date	Date at which customer got the product

6 The **reviews.csv** contain following features:

Features	Description
review_id	ID of the review given on the product ordered by the order id
order_id	A Unique ID of order made by the consumers
review_score	Review score given by the customer for each order on a scale of 1-5
review_creation_date	Timestamp of the review when it is created
review_answer_timestamp	Timestamp of the review answered

7 The **products.csv** contain following features:

Features	Description
product_id	A Unique identifier for the proposed project.
product_category_name	Name of the product category

IDEF DIAGRAMS

Figure: NODE TREE

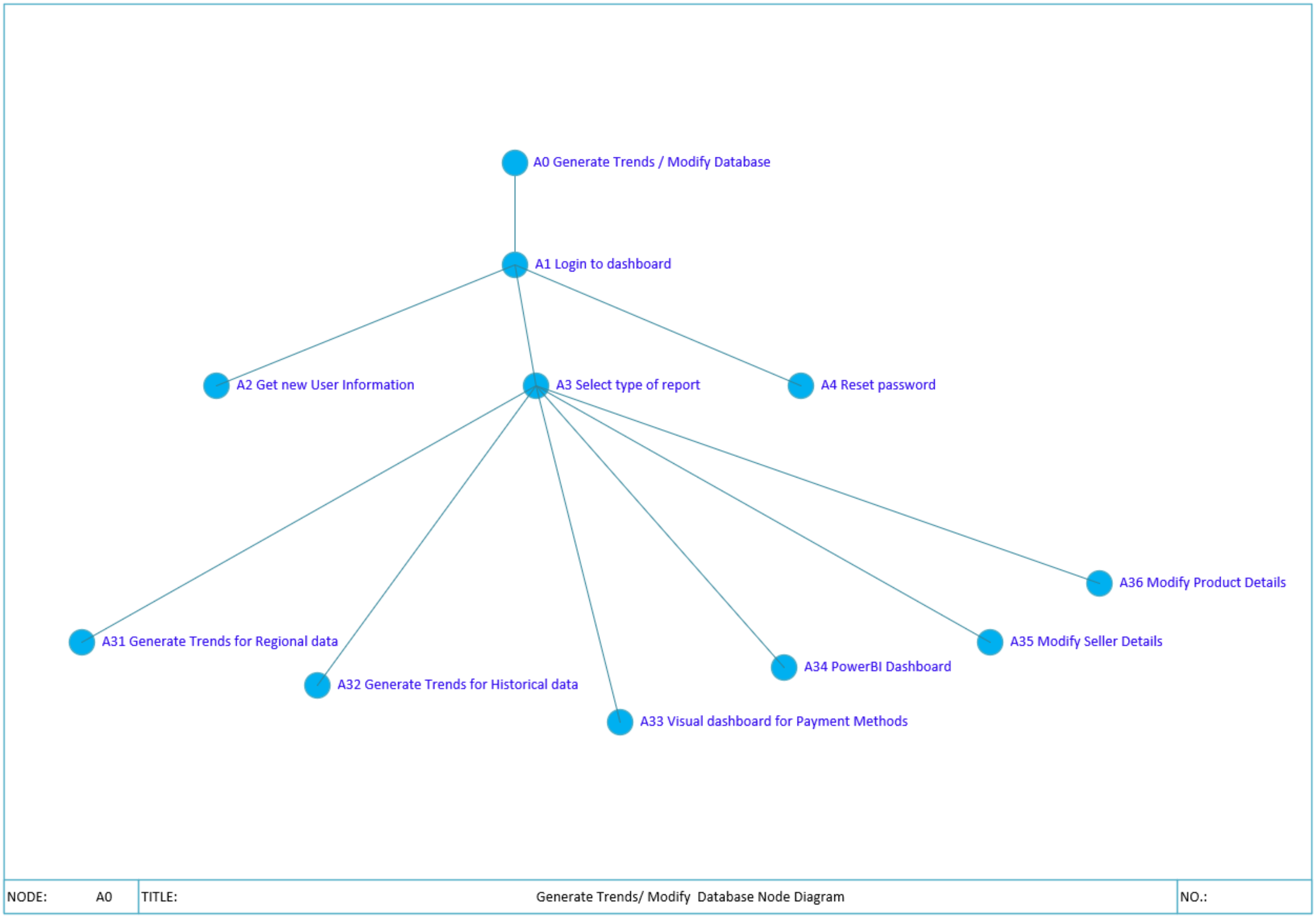
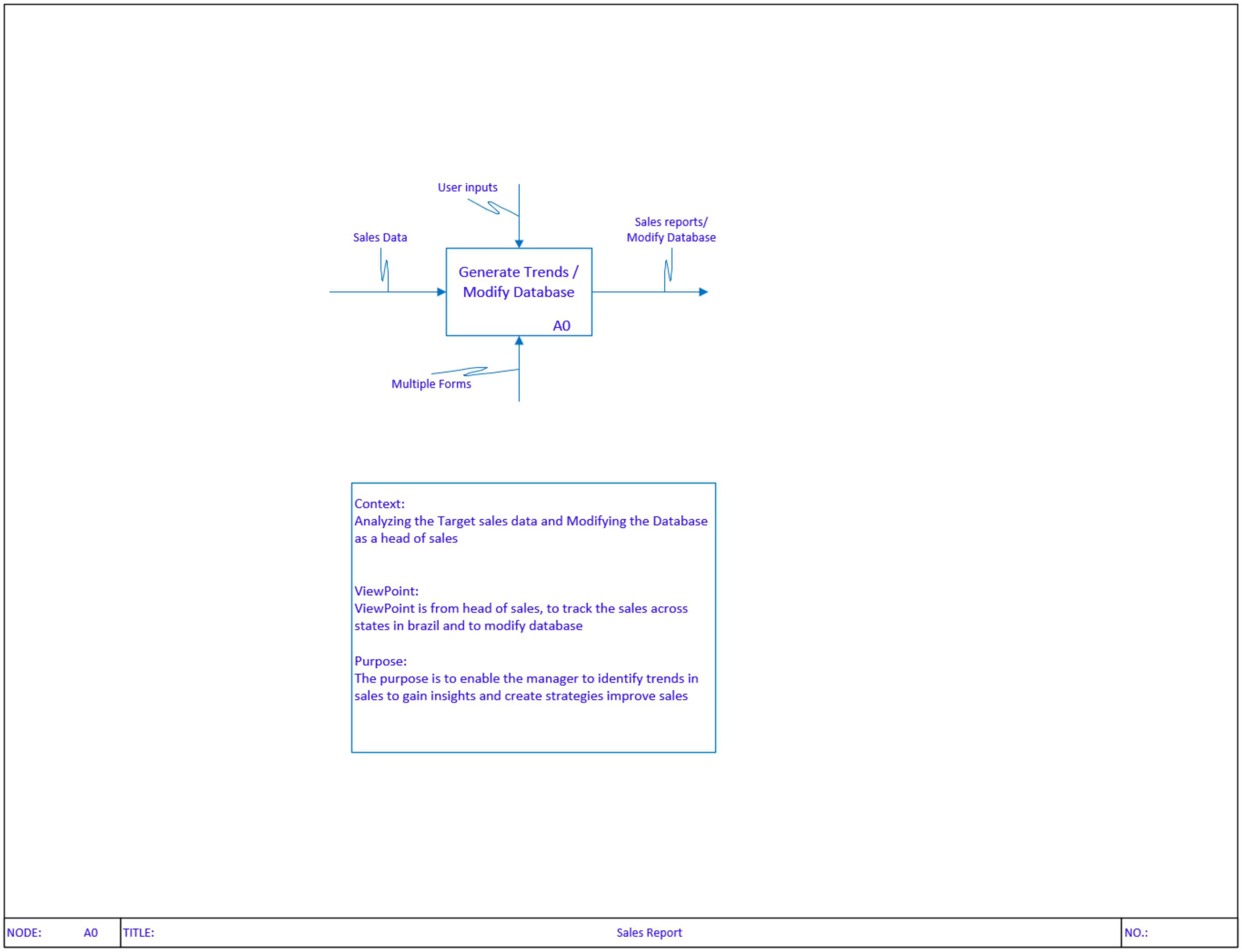

A0 Generate Trends / Modify Database
A1 Login to dashboard
A2 Get new User Information
A3 Select type of report
A4 Reset password
A36 Modify Product Details
A31 Generate Trends for Regional data
A35 Modify Seller Details
A34 PowerBI Dashboard
A32 Generate Trends for Historical data
A33 Visual dashboard for Payment Methods
NODE: A0
TITLE: Generate Trends/ Modify Database Node Diagram
NO.:

Figure: IDEF 0 TOP LEVEL



Context:
Analyzing the Target sales data and Modifying the Database as a head of sales

ViewPoint:
ViewPoint is from head of sales, to track the sales across states in brazil and to modify database

Purpose:
The purpose is to enable the manager to identify trends in sales to gain insights and create strategies improve sales

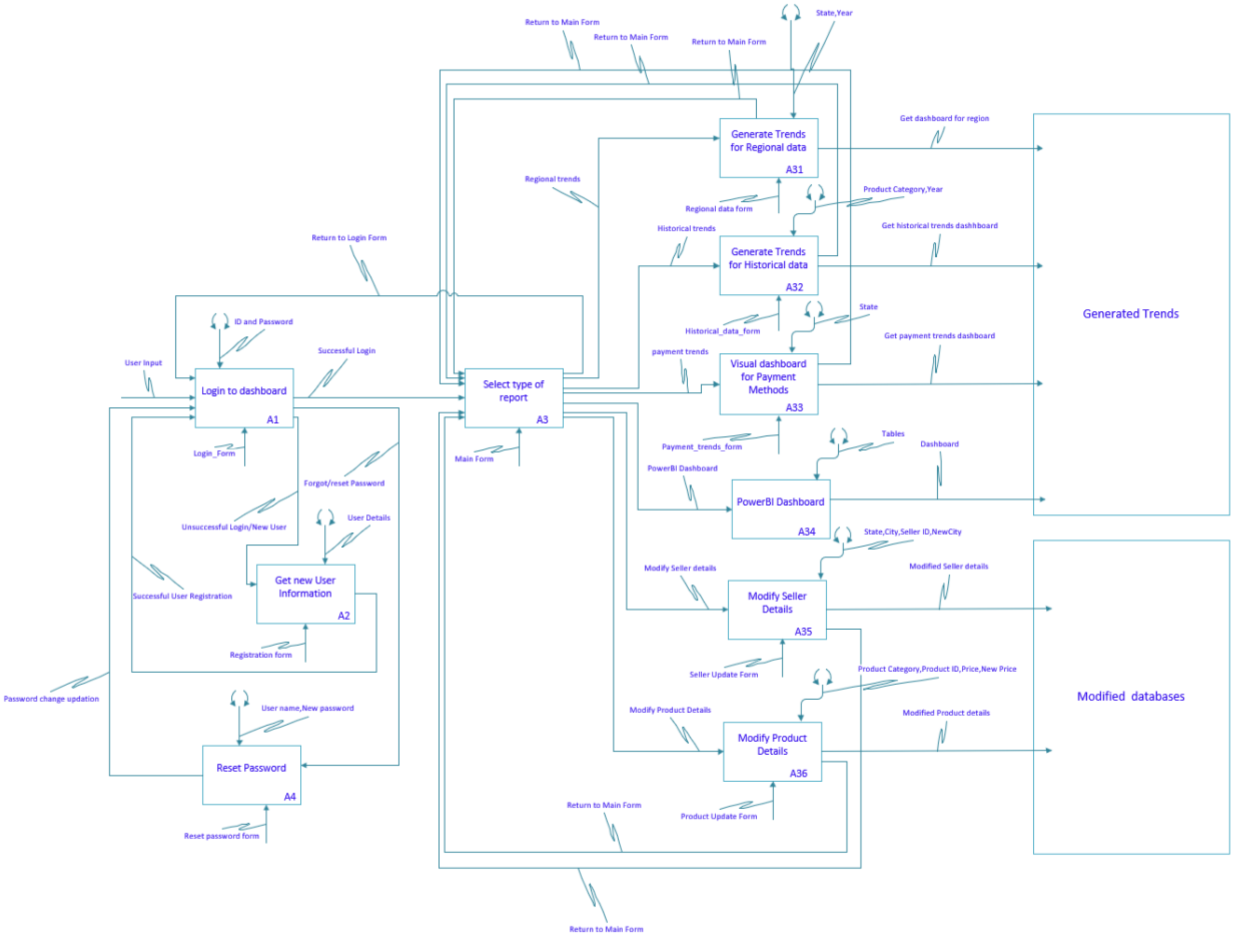


Figure: IDEF 0

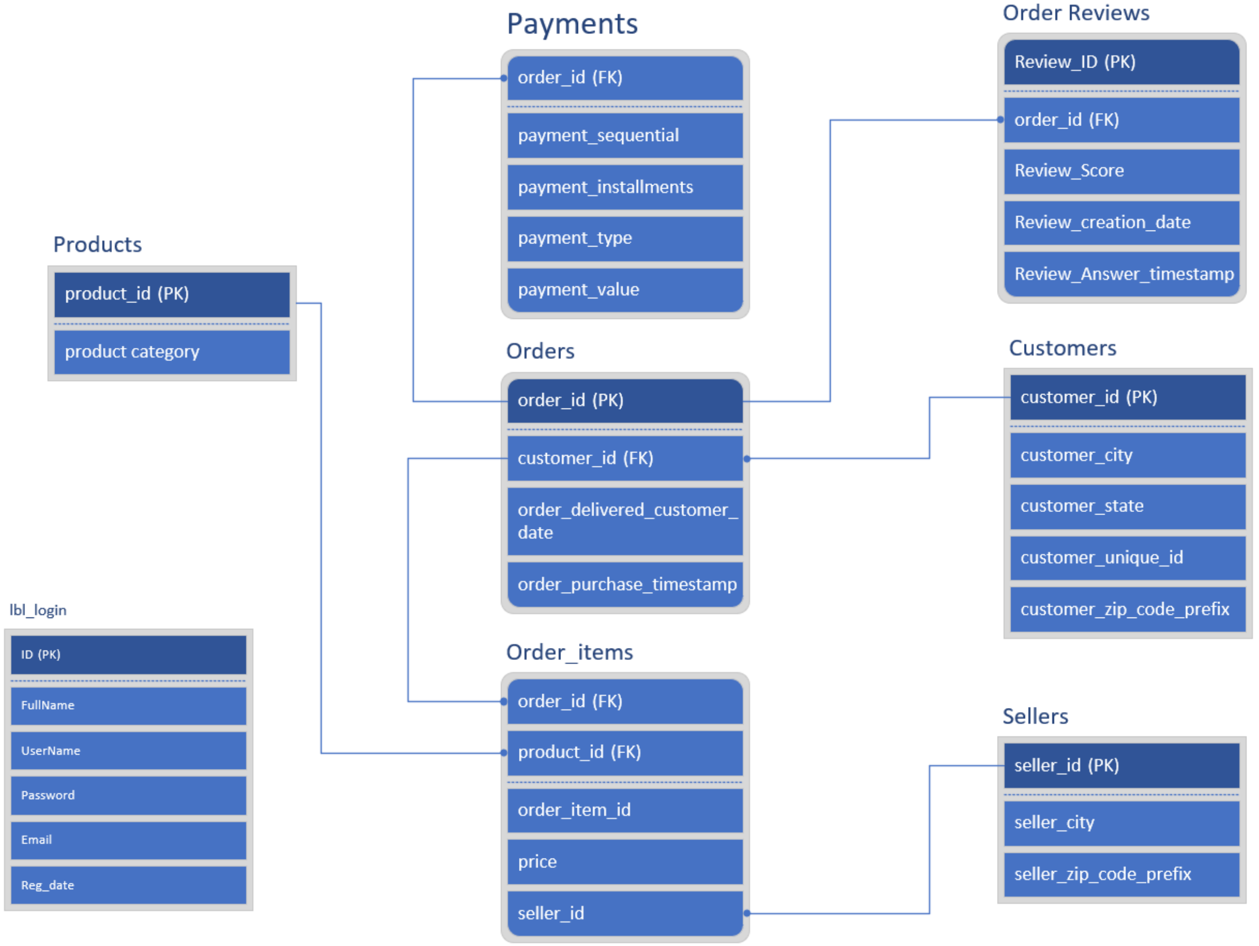


Figure: IDEF 1X

APPLICATION PROCEDURE

1. The application starts with a 'Login_Form' form that lets the User login with their credentials. There is also a Register New User option where the user will be directed to another form, to create the log in credentials for a new user. There is also a Forgot/Reset Password option where the user will be directed to another form, to create a new password.

2. In the 'reset_password_form' form, the User can enter the current username, new password and confirm the password. After entering the details in the above fields, the user can save the password. The new password is updated to 'lbl_login' table.

3. In the 'reg_form' form, the User can enter the ID, Username, password, e-mail. The details are updated to 'lbl_login' table after clicking the 'Register' button. Also, there are button options to 'Return to login' and to enter a 'New Entry'.

4. After successful login, the User is led to the 'Main' form from where they can go to either 'Regional trends' form, 'Historical data' form, 'Payment trends' form, 'Product Price Update' form, 'Seller Update' form or they can go to the 'PowerBI dashboard' form or they can end the application by clicking 'END' button.

5. In the 'Regional trends' form, the User can select state and/or year to get specific trends generated in charts. The User can choose to go back to the 'Main' form as well.

6. In the 'Historical data' form, the User can select Product category and/or year to get specific trends generated in the charts. The User can choose to go back to the 'Main' form as well.

7. In the 'Payment trends' form, the User can select state to view the payment methods to analyze the preferred payment methods in that state. The User can choose to go back to the 'Main' form as well.

8. In the 'Product Price Update' form, the User can select Product category and productID and the existing product price is displayed. After the user enters a new price, the price gets updated to 'Products' table against the selected ProductID. The User can choose to go back to the 'Main' form as well.

9. In the 'Seller Update' form, the User can select the existing sellers state, city and ID and can update the new city for that seller. After clicking the update seller button, the seller city is updated to 'sellers' table. The user also has an option to delete the selected seller from database. The User can choose to go back to the 'Main' form as well.

10. From the main form, the user choose can choose to navigate to a PowerBI dashboard by clicking on the "PowerBI dashborad' button. In the PowerBI dashboard, the user has slicer options to choose state wise, year wise and quarter wise trends for different product categories, sellers and payment methods and order volumes etc.

FORMS SPECIFICATION

00 Login Form

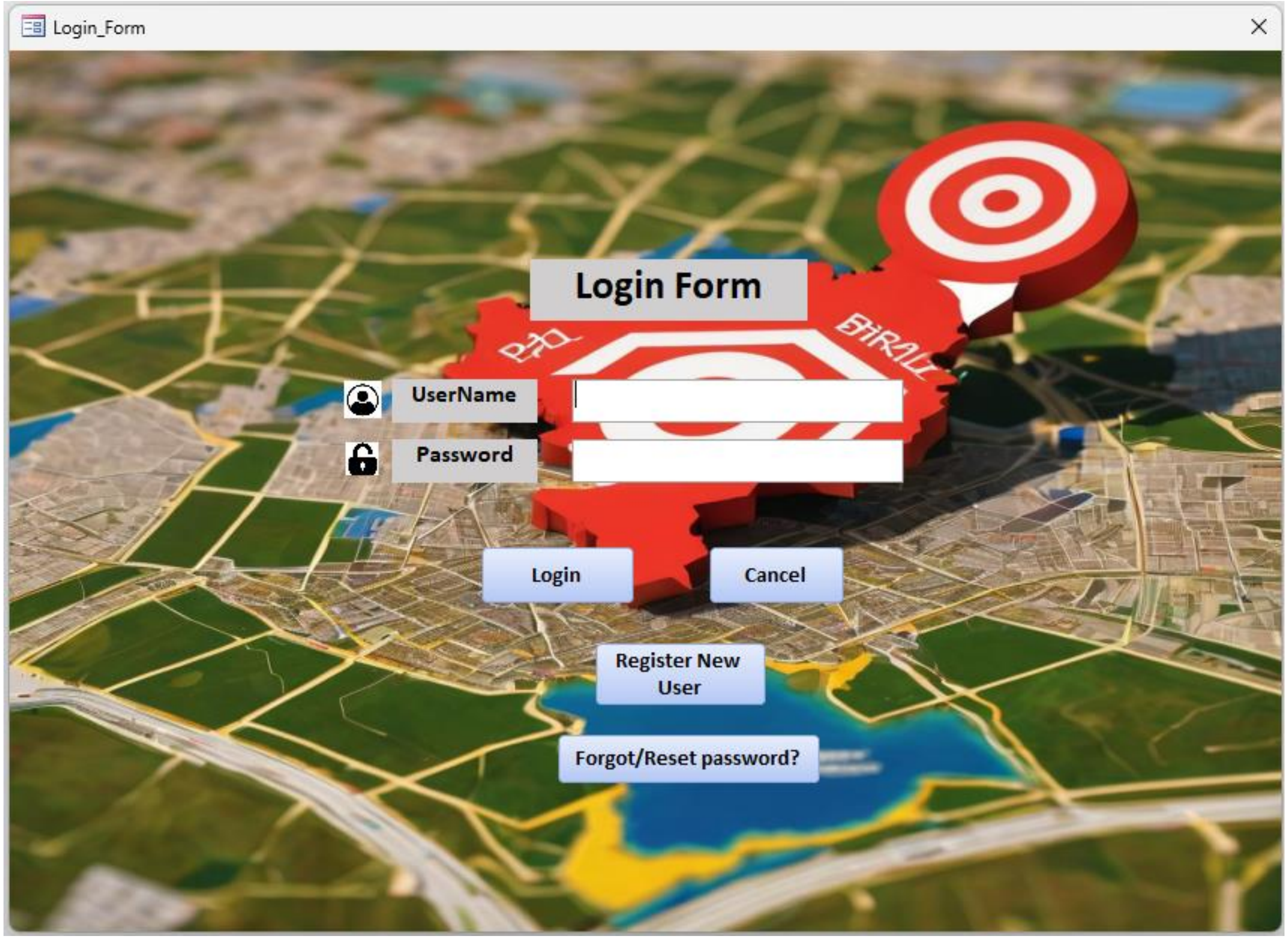
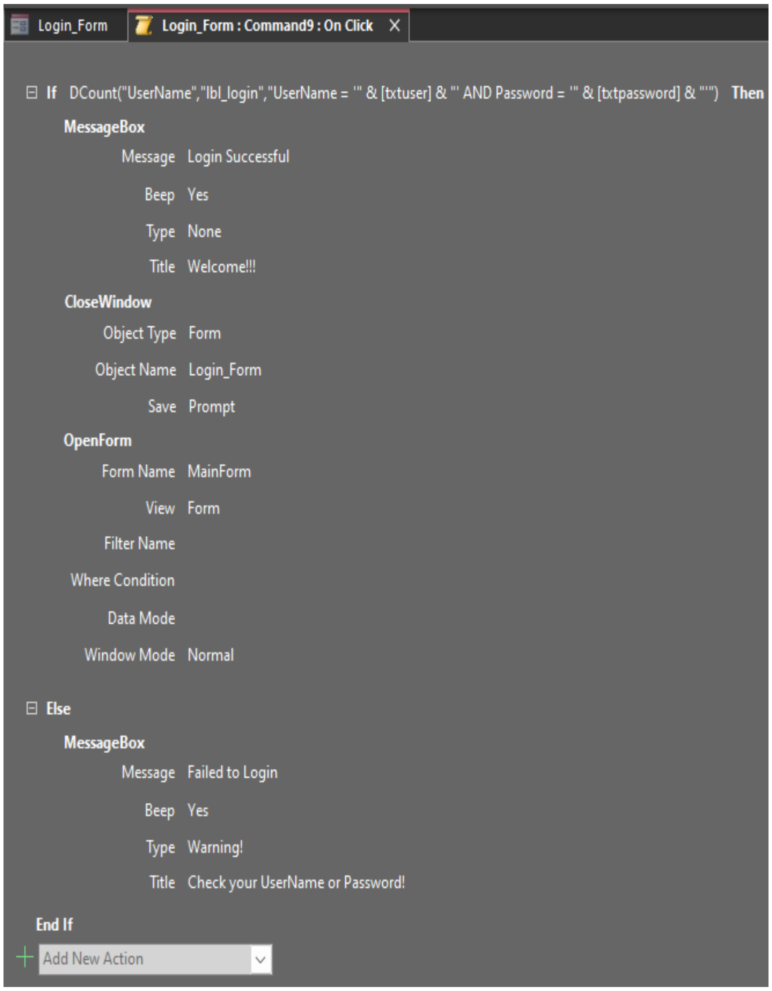
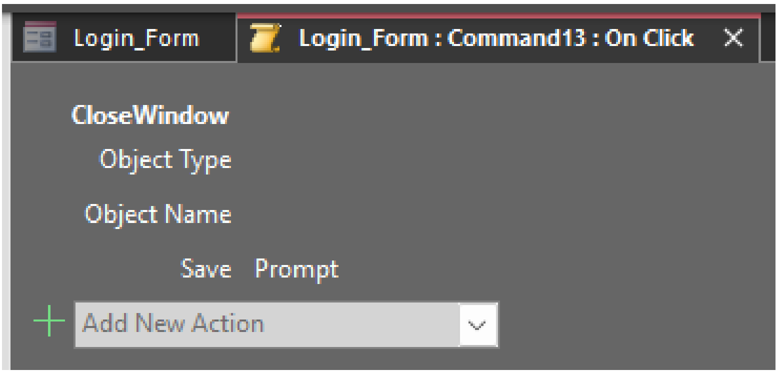


Figure:0.1 Login Page

Label/Title	Object Name	Attached Queries	Attached Macros
UserName	txtuser	NA	NA
Password	txtpassword	NA	NA
Login	Command9	[Embedded Macro]	 If DCount("UserName","lbl_login","UserName = '" & [txtuser] & "' AND Password = '" & [txtpassword] & "'") Then MessageBox Message Login Successful Beep Yes Type None Title Welcome!!! CloseWindow Object Type Form Object Name Login_Form Save Prompt OpenForm Form Name MainForm View Form Filter Name Where Condition Data Mode Window Mode Normal Else MessageBox Message Failed to Login Beep Yes Type Warning! Title Check your UserName or Password! End If
Cancel	Command13	[Embedded Macro]	 CloseWindow Object Type Object Name Save Prompt

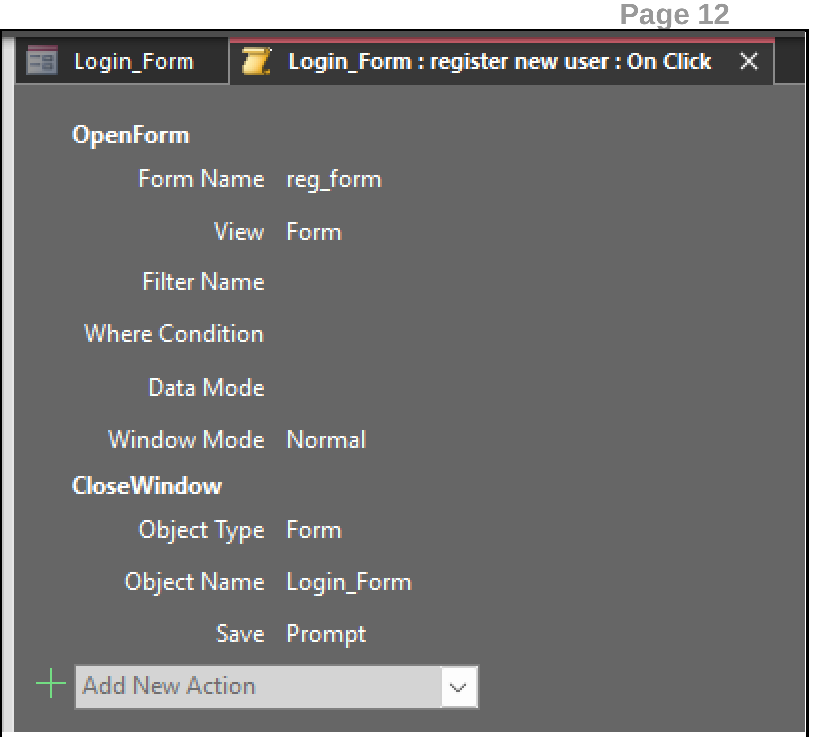
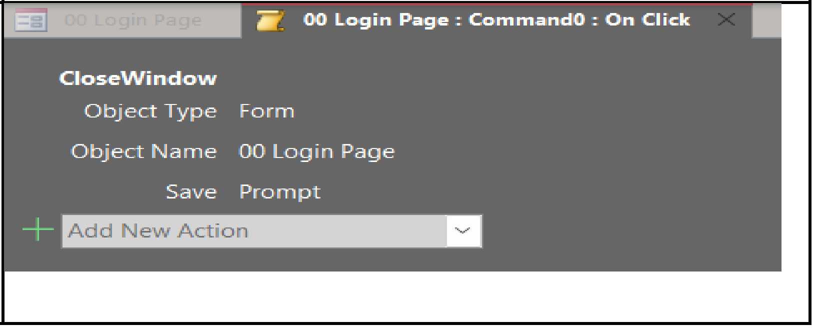
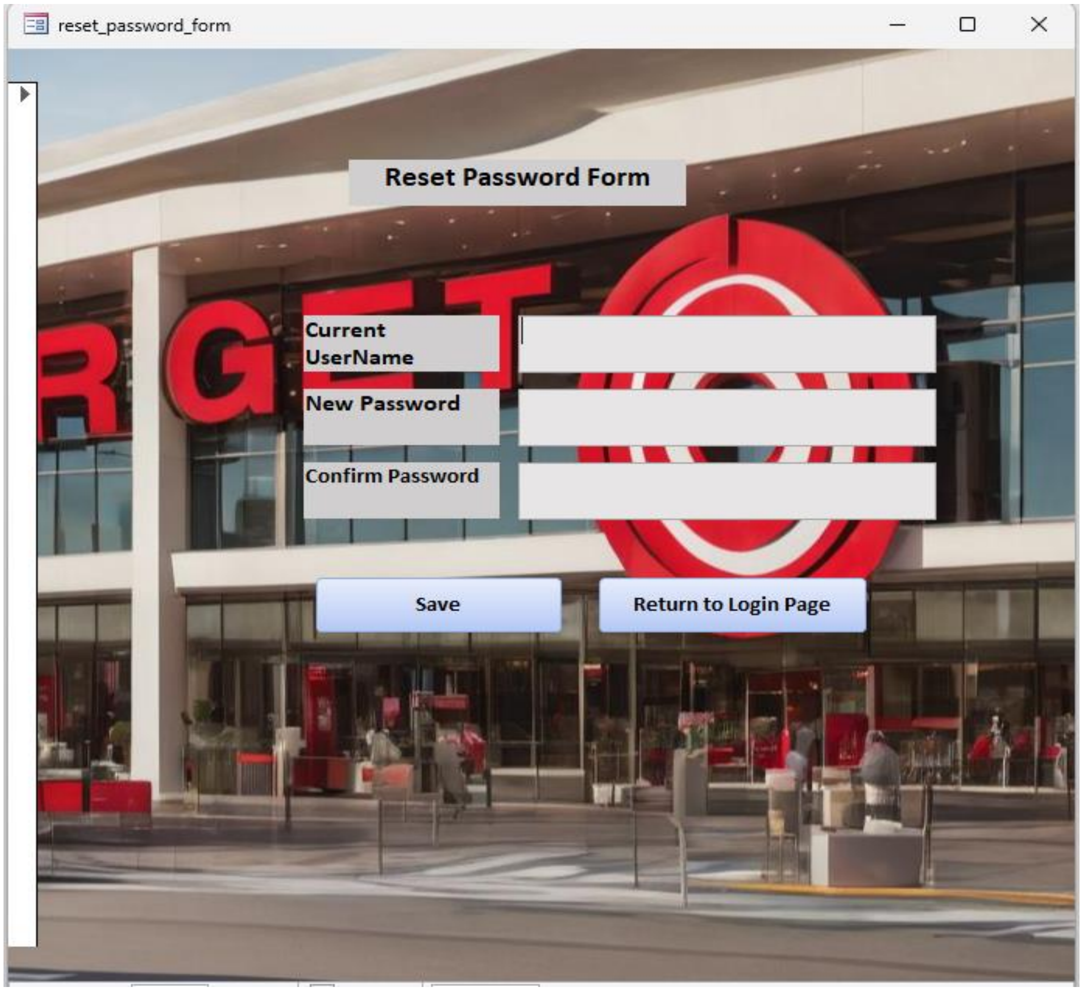
Register New User	register new user	[Embedded Macro]	 Login_Form Login_Form : register new user : On Click OpenForm Form Name reg_form View Form Filter Name Where Condition Data Mode Window Mode Normal CloseWindow Object Type Form Object Name Login_Form Save Prompt Add New Action
Forgot/Reset password?	forgot_password	[Embedded Macro]	 00 Login Page 00 Login Page : Command0 : On Click CloseWindow Object Type Form Object Name 00 Login Page Save Prompt Add New Action

Table:0.1 Login Page Details

Label/Title	**Object Name**	**Event Property**	**Event Action**
UserName	txtuser	N/A	Enter the username
Password	txtpassword	N/A	Enter the passwaord
Login	Command9	On Click	If entered Correct Username & Password, “Main Menu” will Open,else it will display the message “Failed to login”
Cancel	Command13	On Click	Closes the login window
Register New User	Register new user	On Click	Takes user to Registration form
Forgot/Reset password?	forgot_password	On Click	Takes the user to Forgot/Reset password form

Table:0.2 Login Page Actions

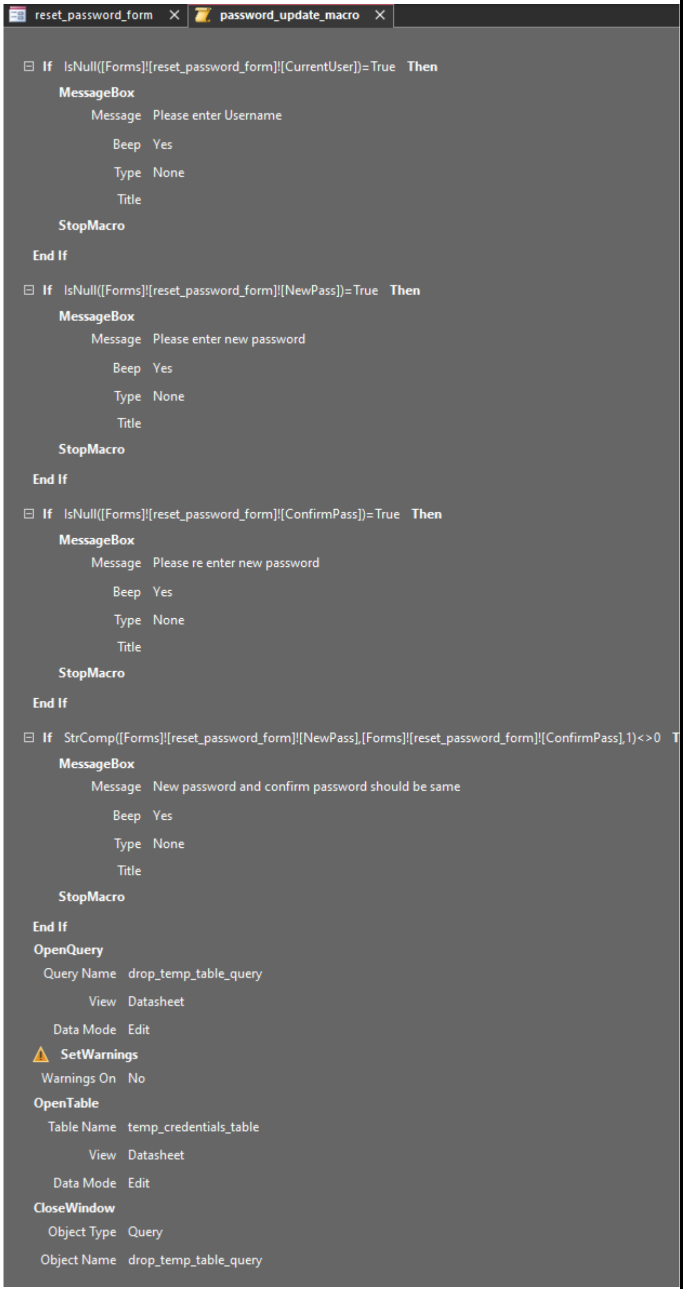
01 Forget / Reset Password



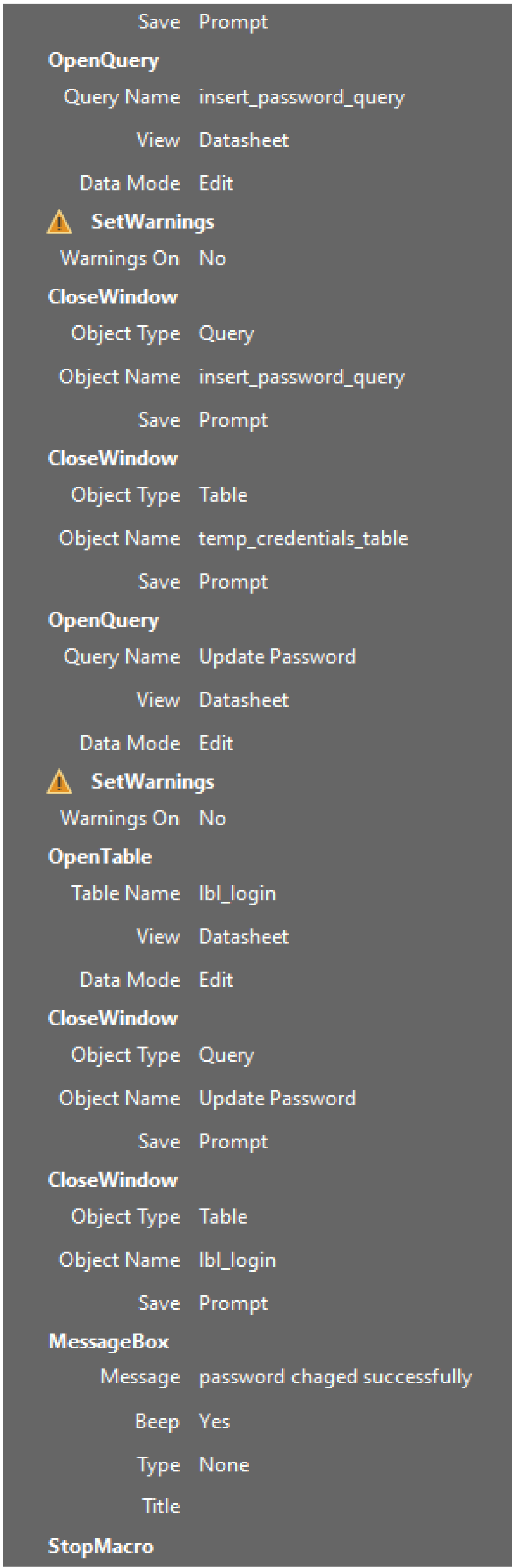
Reset Password Form

Current UserName	
New Password	
Confirm Password	

Figure:1.1 Forget/Reset Password

Label/Title	Object Name	Attached Queries	Attached Macros
Current UserName	CurrentUser	N/A	N/A
New Password	NewPass	N/A	N/A
Confirm Password	ConfirmPass	N/A	N/A
Save	validate	password_update_macro	

```
⊟ If   IsNull([Forms]![reset_password_form]![CurrentUser])=True   Then
        MessageBox
              Message   Please enter Username
                  Beep   Yes
                  Type   None
                   Title
        StopMacro
   End If
⊟ If   IsNull([Forms]![reset_password_form]![NewPass])=True   Then
        MessageBox
              Message   Please enter new password
                  Beep   Yes
                  Type   None
                   Title
        StopMacro
   End If
⊟ If   IsNull([Forms]![reset_password_form]![ConfirmPass])=True   Then
        MessageBox
              Message   Please re enter new password
                  Beep   Yes
                  Type   None
                   Title
        StopMacro
   End If
⊟ If   StrComp([Forms]![reset_password_form]![NewPass],[Forms]![reset_password_form]![ConfirmPass],1)<>0   T
        MessageBox
              Message   New password and confirm password should be same
                  Beep   Yes
                  Type   None
                   Title
        StopMacro
   End If
   OpenQuery
     Query Name   drop_temp_table_query
           View   Datasheet
      Data Mode   Edit
   ⚠ SetWarnings
    Warnings On   No
   OpenTable
     Table Name   temp_credentials_table
           View   Datasheet
      Data Mode   Edit
   CloseWindow
    Object Type   Query
    Object Name   drop_temp_table_query
```

			 Save Prompt **OpenQuery** Query Name insert_password_query View Datasheet Data Mode Edit ⚠ **SetWarnings** Warnings On No **CloseWindow** Object Type Query Object Name insert_password_query Save Prompt **CloseWindow** Object Type Table Object Name temp_credentials_table Save Prompt **OpenQuery** Query Name Update Password View Datasheet Data Mode Edit ⚠ **SetWarnings** Warnings On No **OpenTable** Table Name lbl_login View Datasheet Data Mode Edit **CloseWindow** Object Type Query Object Name Update Password Save Prompt **CloseWindow** Object Type Table Object Name lbl_login Save Prompt **MessageBox** Message password chaged successfully Beep Yes Type None Title **StopMacro**

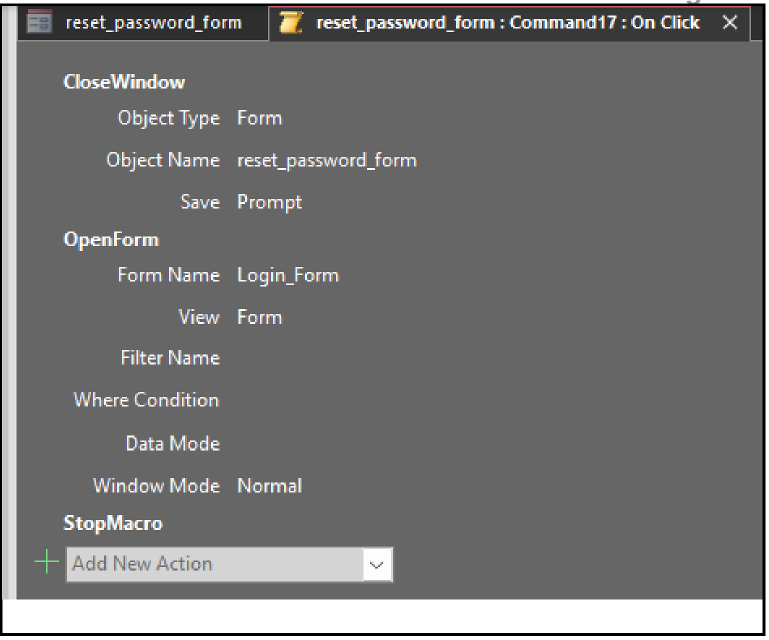
Return to Login Page	Command17	N/A	 reset_password_form reset_password_form : Command17 : On Click CloseWindow Object Type Form Object Name reset_password_form Save Prompt OpenForm Form Name Login_Form View Form Filter Name Where Condition Data Mode Window Mode Normal StopMacro

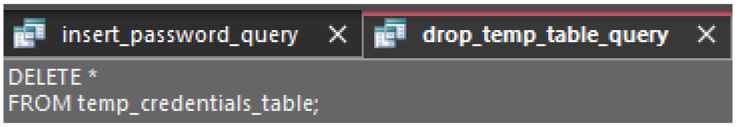
Table:1.1 Forget Password Details

Label/Title	Object Name	Event Property	Event Action
Current UserName	CurrentUser	N/A	Enter the username
New Password	NewPass	N/A	Enter the password
Confirm Password	ConfirmPass	N/A	Re-enter the password
Save	validate	On Click	If entered Password, and confirm password are matching the password is updated
Return to Login Page	Command17	On Click	Takes user to Registration form

Table:1.2 Forget Password Actions

Queries for password update macro:

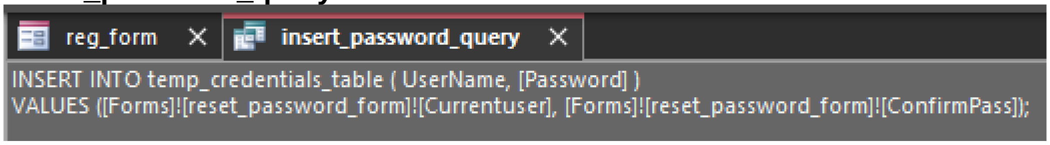
drop_temp_table_query:



```
DELETE *
FROM temp_credentials_table;
```

Figure 1.2: drop temp table query.

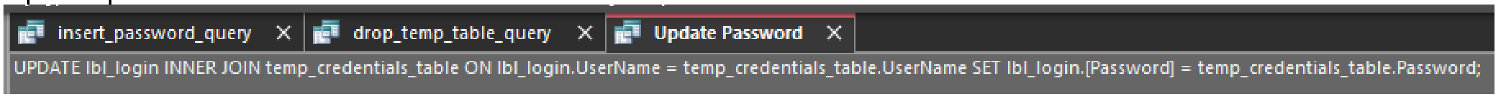
Insert_password_query:



```
INSERT INTO temp_credentials_table ( UserName, [Password] )
VALUES ([Forms]![reset_password_form]![Currentuser], [Forms]![reset_password_form]![ConfirmPass]);
```

Figure 1.3: insert password query.

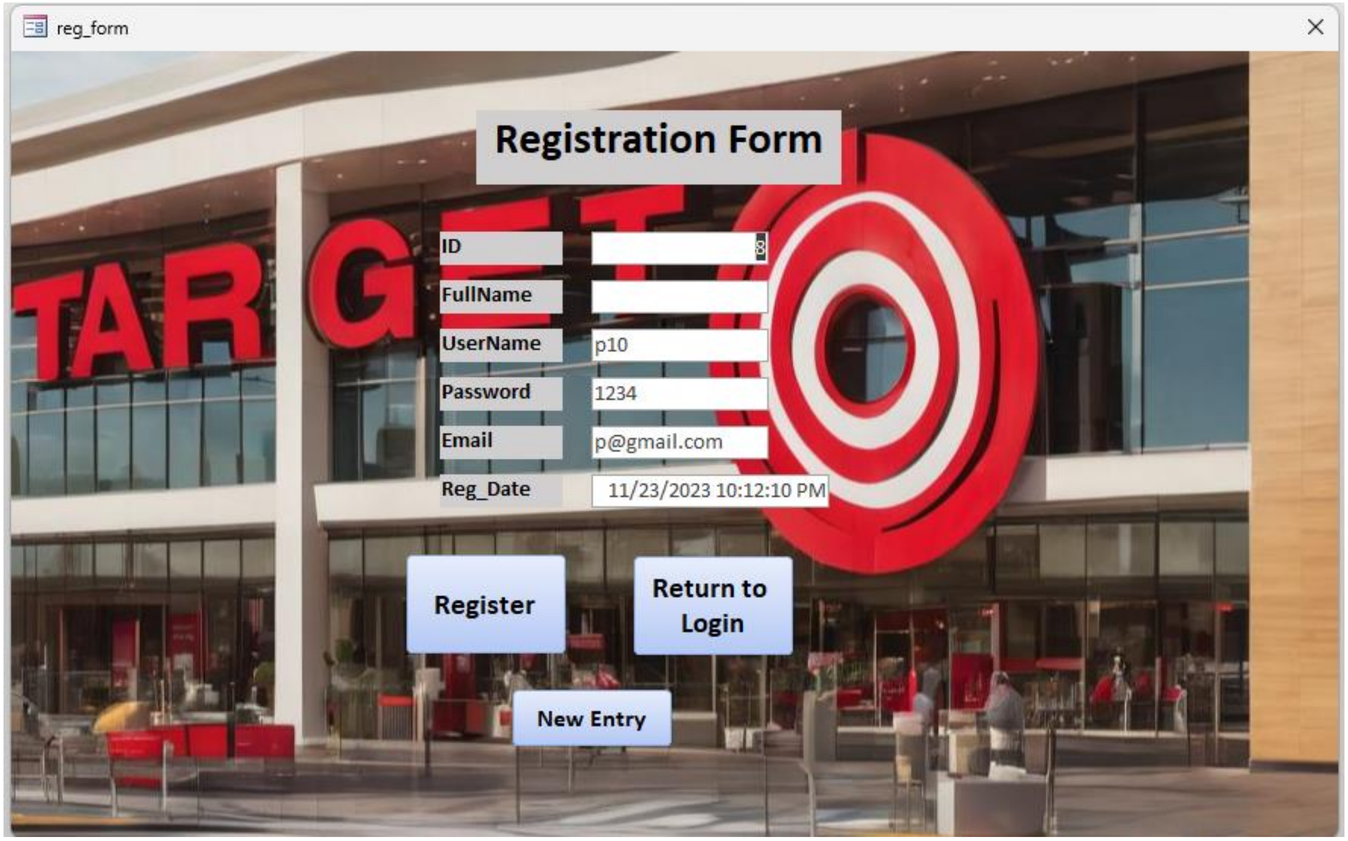
Update password:



```
UPDATE lbl_login INNER JOIN temp_credentials_table ON lbl_login.UserName = temp_credentials_table.UserName SET lbl_login.[Password] = temp_credentials_table.Password;
```

Figure 1.4: update password query.

02 New User Registration Form

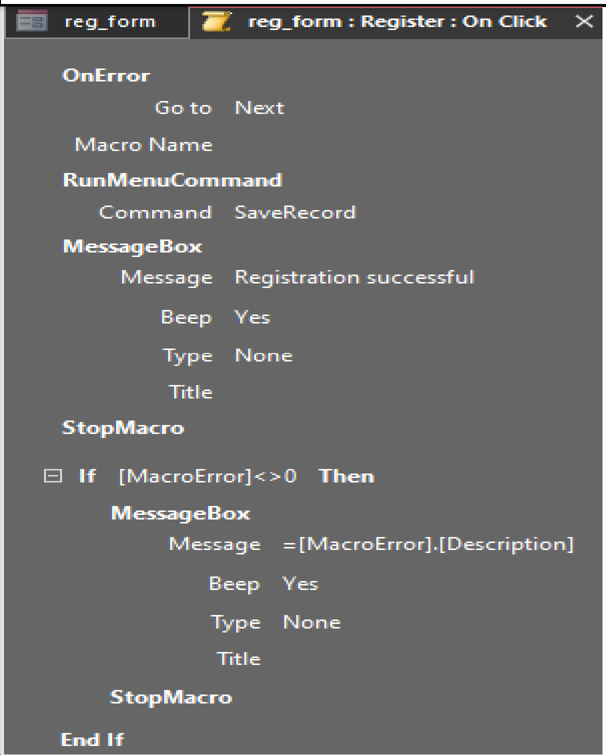
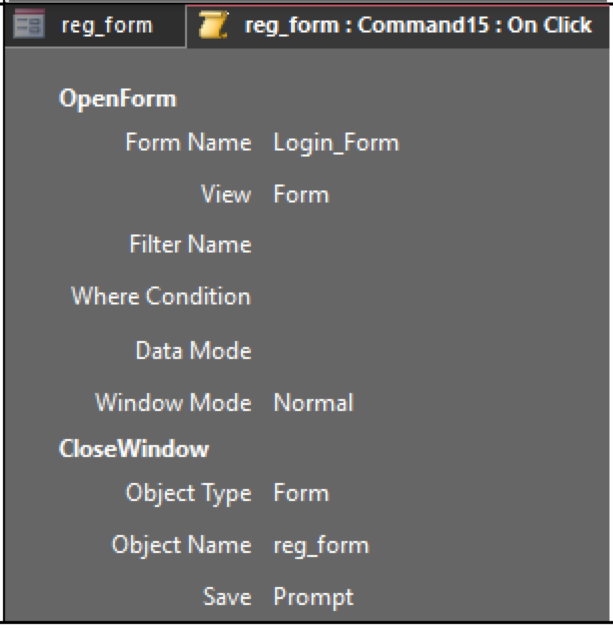


reg_form

Registration Form

ID	8
FullName	
UserName	p10
Password	1234
Email	p@gmail.com
Reg_Date	11/23/2023 10:12:10 PM

Figure: 2.1 New User Registration Form

Label/Title	Object Name	Attached Queries	Attached Macros
ID	ID	N/A	N/A
FullName	FullName	N/A	N/A
UserName	UserName	N/A	N/A
Password	Password	N/A	N/A
Email	Email	N/A	N/A
Reg_Date	Reg_Date	N/A	N/A
Register	Register	[Embedded Macro]	 reg_form reg_form : Register : On Click OnError Go to Next Macro Name RunMenuCommand Command SaveRecord MessageBox Message Registration successful Beep Yes Type None Title StopMacro If [MacroError]<>0 Then MessageBox Message =[MacroError].[Description] Beep Yes Type None Title StopMacro End If
Return to Login	Command15	[Embedded Macro]	 reg_form reg_form : Command15 : On Click OpenForm Form Name Login_Form View Form Filter Name Where Condition Data Mode Window Mode Normal CloseWindow Object Type Form Object Name reg_form Save Prompt

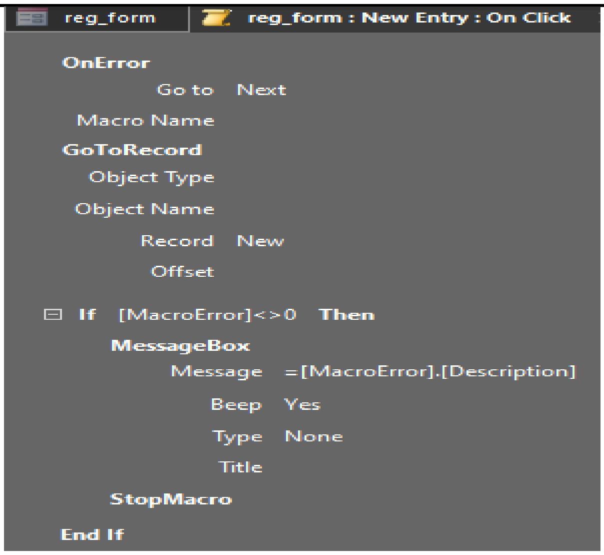
New Entry	New Entry		 reg_form reg_form : New Entry : On Click OnError Go to Next Macro Name GoToRecord Object Type Object Name Record New Offset If [MacroError]<>0 Then MessageBox Message =[MacroError].[Description] Beep Yes Type None Title StopMacro End If

Table:2.1 Registration Form Details

Label/Title	**Object Name**	**Event Property**	**Event Action**
ID	ID	N/A	Generates new ID+1 greater than the table ID row
FullName	FullName	N/A	Enter User Full Name
UserName	UserName	N/A	Enter UserName
Password	Password	N/A	Enter password
Email	Email	N/A	Enter Email ID
Reg_Date	Reg_Date	N/A	Pick a date
Register	Register	OnClick	Saves user details in lbl_login table and provides a popup as user successfully registered
Return to Login	Command15	OnClick	Takes user to login form
New Entry	New Entry	OnClick	Clears the fields in the form

Table:2.2 Registration Form Actions

03 Main Form

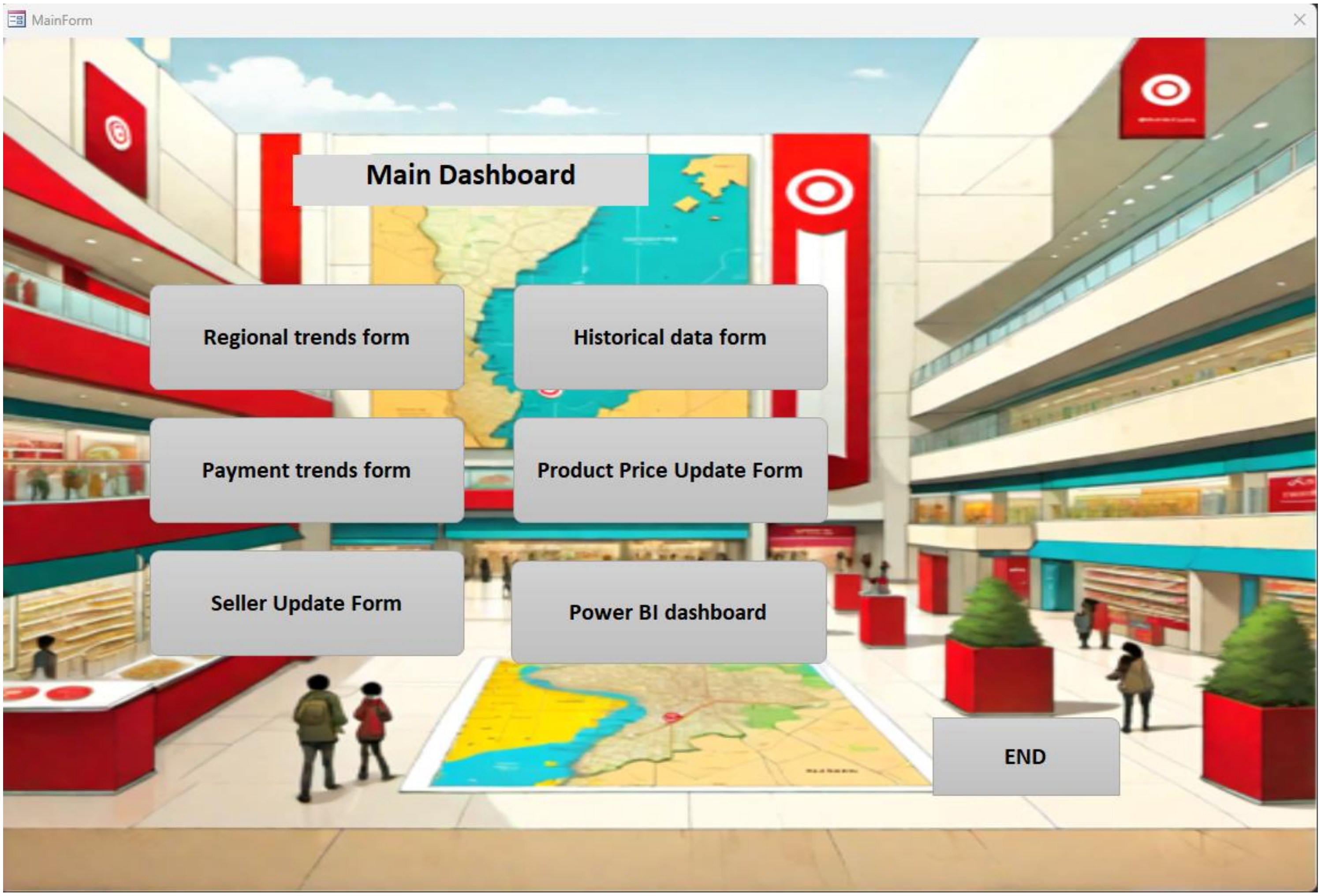
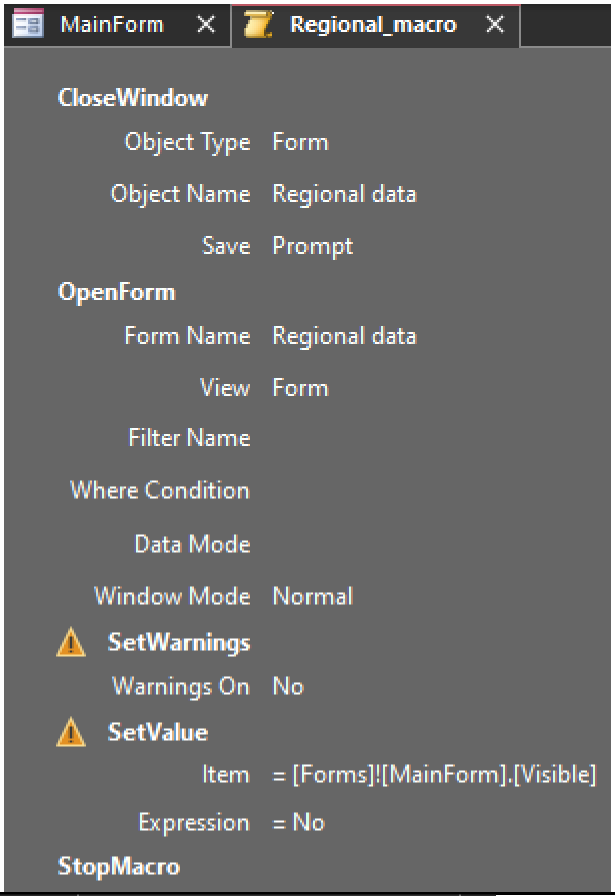
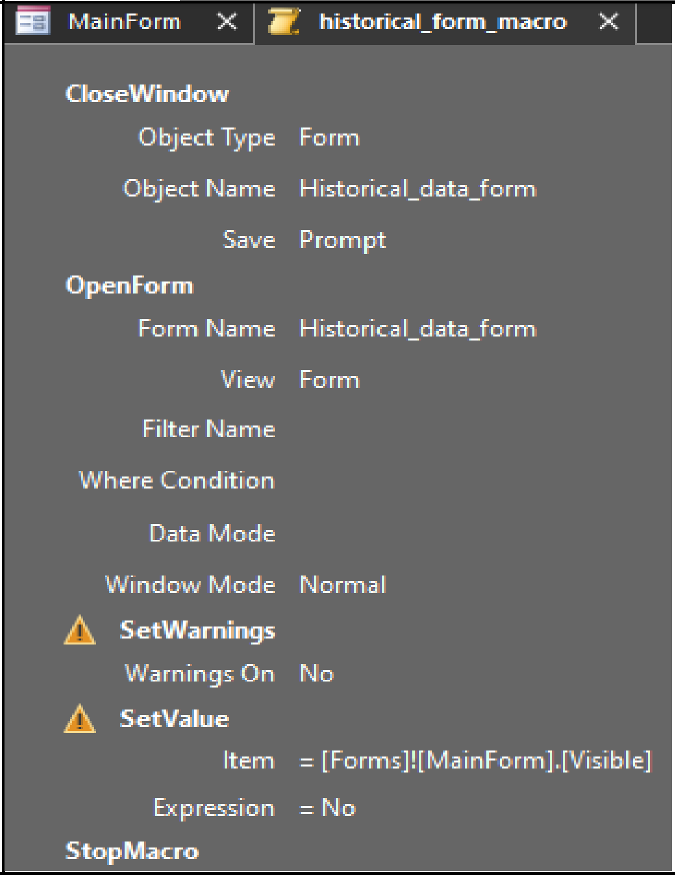
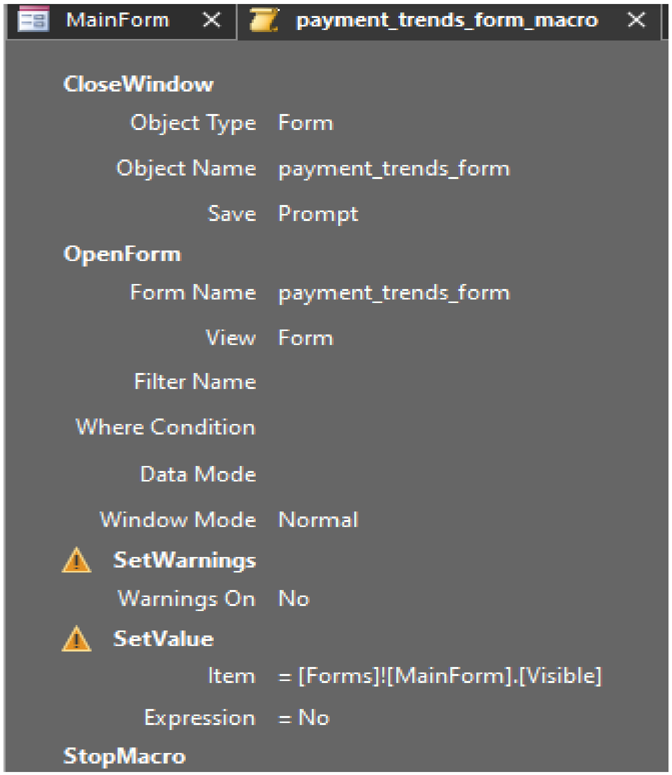
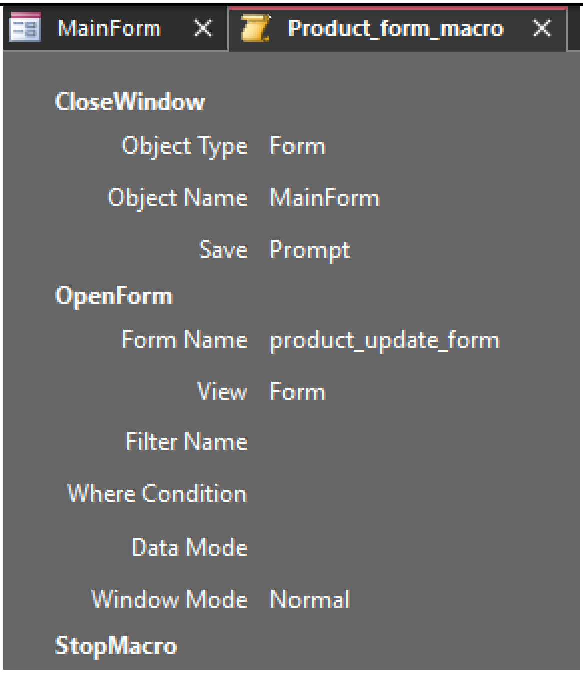
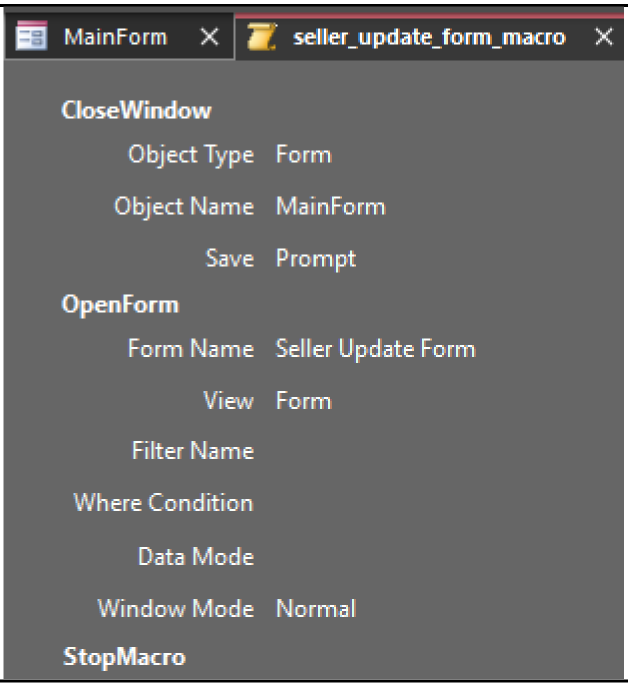


Figure:3.1 Main Dashboard

Label/Title	Object Name	Attached Macros
Regional trends form	Command3	 **CloseWindow** Object Type Form Object Name Regional data Save Prompt **OpenForm** Form Name Regional data View Form Filter Name Where Condition Data Mode Window Mode Normal ⚠ **SetWarnings** Warnings On No ⚠ **SetValue** Item = [Forms]![MainForm].[Visible] Expression = No **StopMacro**
Historical Data Form	Command4	 **CloseWindow** Object Type Form Object Name Historical_data_form Save Prompt **OpenForm** Form Name Historical_data_form View Form Filter Name Where Condition Data Mode Window Mode Normal ⚠ **SetWarnings** Warnings On No ⚠ **SetValue** Item = [Forms]![MainForm].[Visible] Expression = No **StopMacro**

Payment trends form	Command5	 MainForm payment_trends_form_macro CloseWindow Object Type Form Object Name payment_trends_form Save Prompt OpenForm Form Name payment_trends_form View Form Filter Name Where Condition Data Mode Window Mode Normal ⚠ SetWarnings Warnings On No ⚠ SetValue Item = [Forms]![MainForm].[Visible] Expression = No StopMacro
Product Price Update Form	Command8	 MainForm Product_form_macro CloseWindow Object Type Form Object Name MainForm Save Prompt OpenForm Form Name product_update_form View Form Filter Name Where Condition Data Mode Window Mode Normal StopMacro

Seller Update Form	Command7	 CloseWindow Object Type Form Object Name MainForm Save Prompt OpenForm Form Name Seller Update Form View Form Filter Name Where Condition Data Mode Window Mode Normal StopMacro
Power BI dashboard	power_bi	VBA code: Private Sub power_bi_Click() Application.FollowHyperlink "https://app.powerbi.com/groups/me/reports/0c990988-7bc8-4334-b3f9-9e3f3142faf7/ReportSection0e70e3053650a7dd1520?experience=power-bi" End Sub

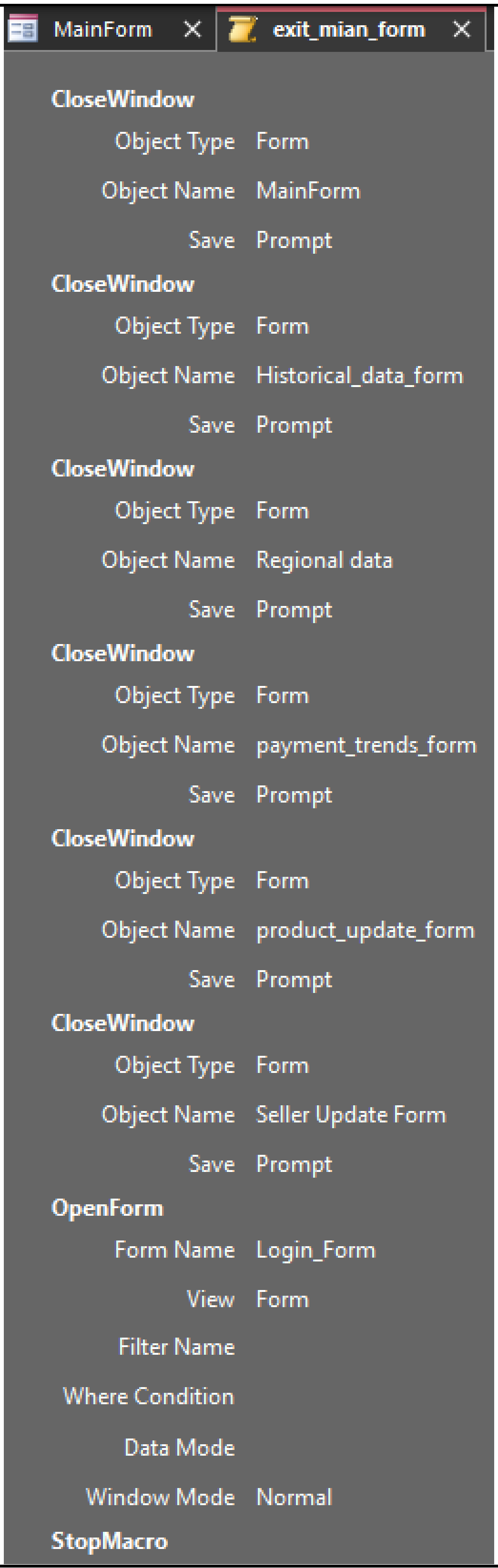
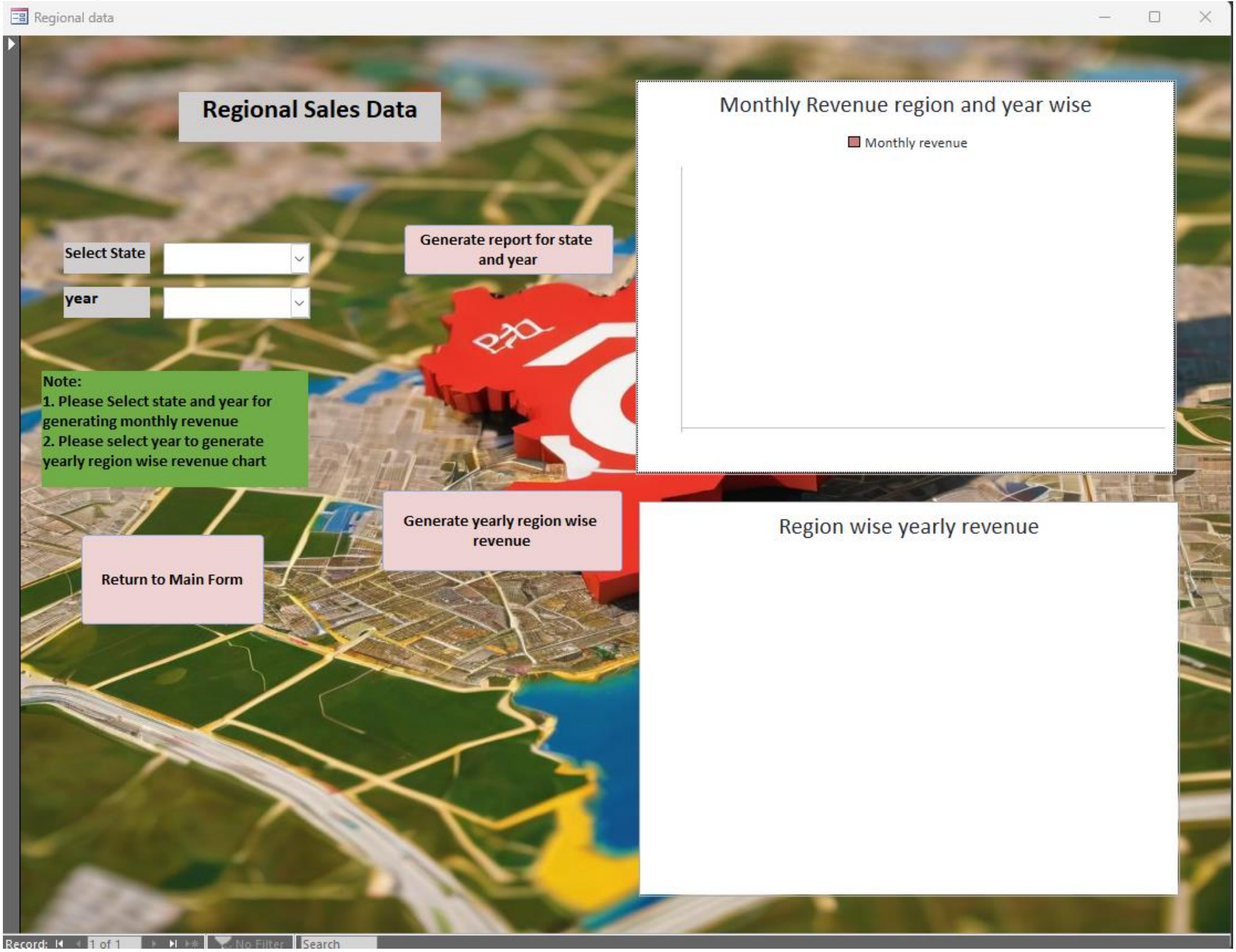
End	Command9	 CloseWindow Object Type Form Object Name MainForm Save Prompt CloseWindow Object Type Form Object Name Historical_data_form Save Prompt CloseWindow Object Type Form Object Name Regional data Save Prompt CloseWindow Object Type Form Object Name payment_trends_form Save Prompt CloseWindow Object Type Form Object Name product_update_form Save Prompt CloseWindow Object Type Form Object Name Seller Update Form Save Prompt OpenForm Form Name Login_Form View Form Filter Name Where Condition Data Mode Window Mode Normal StopMacro

Table:3.1 Main Form Details

Label/Title	Object Name	Event Property	Event Action
Regional Data form	Command3	On Click	Opens Regional Data form
Historical Data Form	Command4	On Click	Opens historical Data form
Payment trends form	Command5	On Click	Opens Payment Trend form
Product Price Update Form	Command8	On Click	Opens product update form
Seller Update Form	Command7	On Click	Opens Seller update form
Power BI dashboard	power_bi	On Click	Opens PowerBI dashboard in google chrome
END	Command9	On Click	Closes main form linked forms

Table:3.2 Main Form Actions

04Regional Data Form


Regional data
Regional Sales Data
Select State
year
Generate report for state and year
Generate yearly region wise revenue
Return to Main Form
Monthly Revenue region and year wise
🟥 Monthly revenue
Region wise yearly revenue
Record: 1 of 1
No Filter
Search

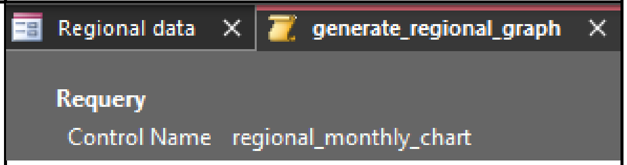
Note:
1. Please Select state and year for generating monthly revenue
2. Please select year to generate yearly region wise revenue chart

Figure:4.1 Regional Data Form

Label/Title	Object Name	Attached Queries	Attached Macros

Select State	state_list	SELECT DISTINCT customer_state
		FROM customers;
			N/A

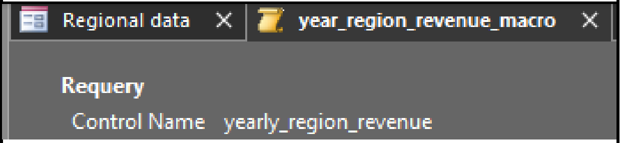
year	year_list	SELECT DISTINCT year(order_purchase_timestamp) AS [YEAR]
		FROM orders;
			N/A

Generate report for state and year	Command5	N/A	 Requery Control Name regional_monthly_chart

regional_monthly_chart	regional_monthly_chart	SELECT MONTH(orders.order_purchase_timestamp) AS [Month], SUM(order_items.price) AS TotalPrice
		FROM (orders INNER JOIN customers ON customers.customer_id = orders.customer_id) INNER JOIN order_items ON orders.order_id = order_items.order_id
		WHERE customers.customer_state = Forms![Regional data]![state_list].Value
		AND YEAR(orders.order_purchase_timestamp) = [Forms]![Regional data]![year_list].Value
		GROUP BY MONTH(orders.order_purchase_timestamp);	

yearly_region_revenue	yearly_region_revenue	SELECT DISTINCT customers.customer_state, SUM(order_items.price) AS TotalPrice
		FROM (orders INNER JOIN customers ON customers.customer_id = orders.customer_id) INNER JOIN order_items ON orders.order_id =	

		order_items.order_id
		WHERE YEAR(orders.order_purchase_timestamp) = [Forms]![Regional data]![year_list].Value
		GROUP BY customers.customer_state;	

Generate yearly region wise revenue	Command7	N/A	 Requery Control Name yearly_region_revenue

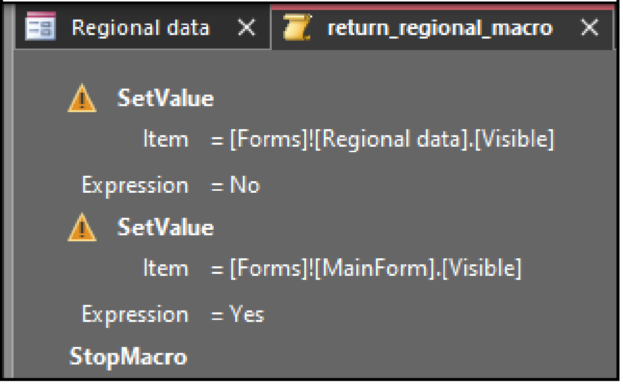
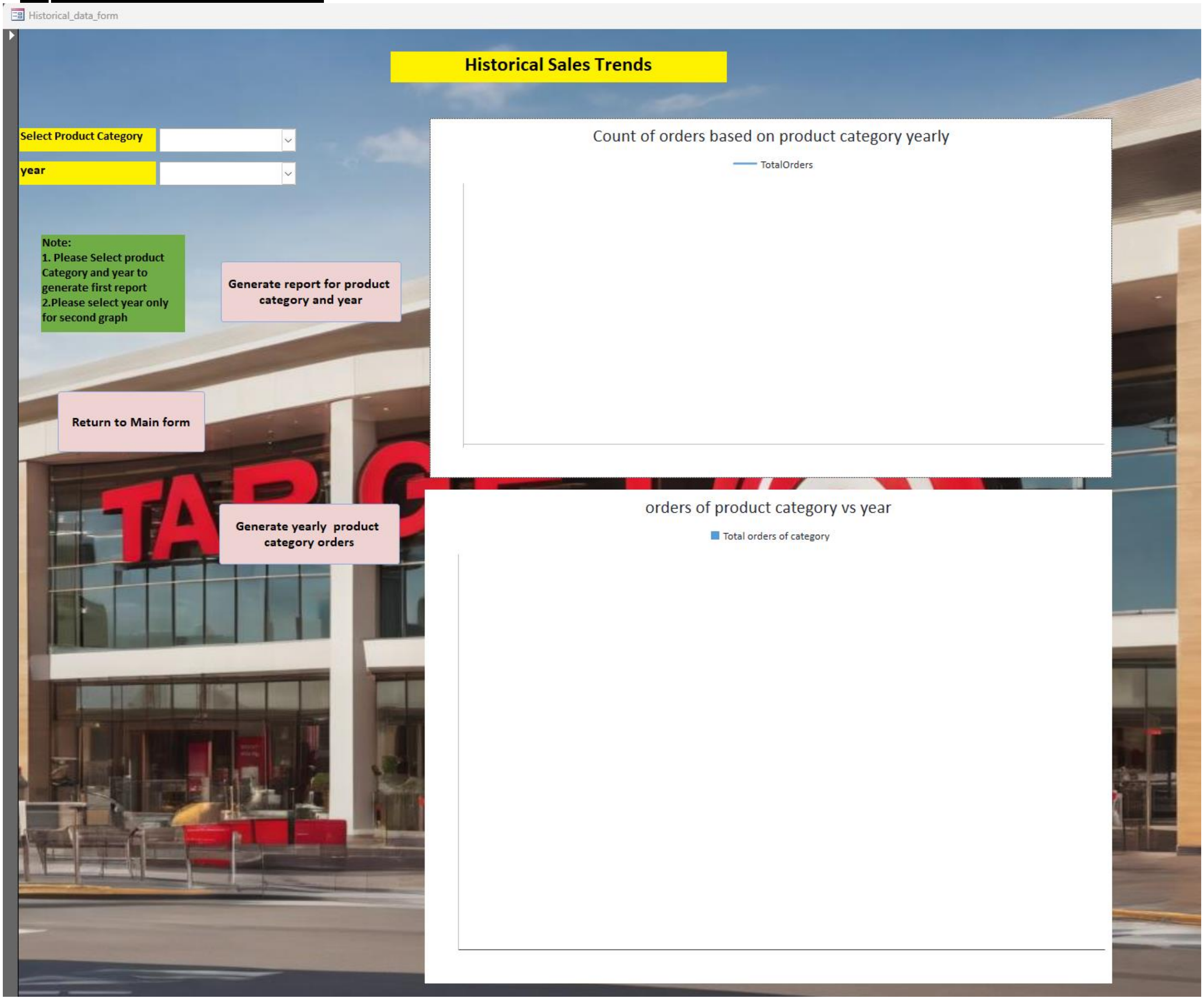
Return to Main Form	Command8	N/A	
			SetValue
			Item = [Forms]![Regional data].[Visible]
			Expression = No
			SetValue
			Item = [Forms]![MainForm].[Visible]
			Expression = Yes
			StopMacro

Table:4.1 Regional Data Details

Label/Title	Object Name	Event Property	Event Action
Select State	state_list	N/A	Select State in the combo box
year	year_list	N/A	Select year in the combo box
Generate report for state and year	Command5	On Click	Generate monthly revenue bar chart based on state and year
Generate yearly region wise revenue	Command7	On Click	Generate yearly revenue pie chart for states
Return to Main Form	Command8	On Click	Return to main Form

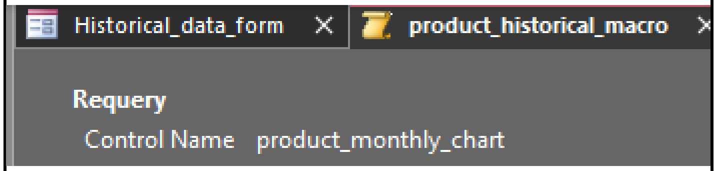
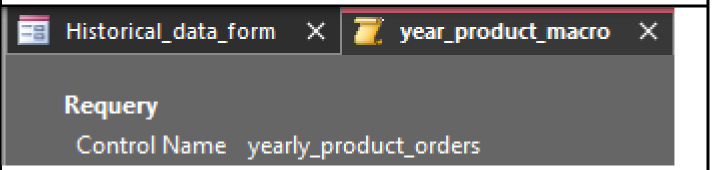
Table:4.2 Regional Data Form Actions

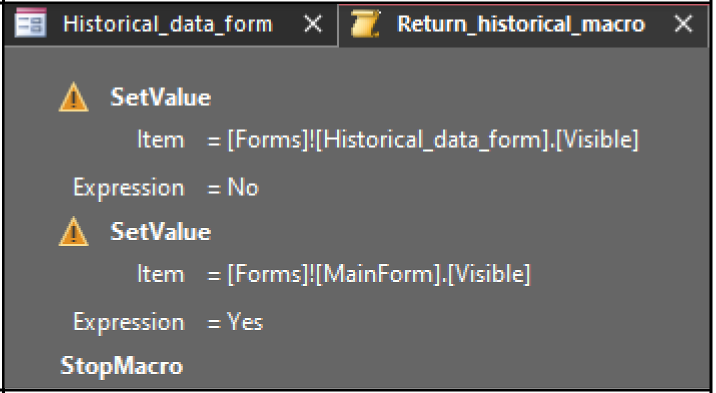
05 Historical Data Form


Historical_data_form
Historical Sales Trends
Select Product Category
year
Generate report for product category and year
Return to Main form
Generate yearly product category orders
Count of orders based on product category yearly
TotalOrders
orders of product category vs year
Total orders of category

Note:
1. Please Select product Category and year to generate first report
2.Please select year only for second graph

Figure:5.1 historical Data form

Label/Title	Object Name	Attached Queries/ Row Source	Attached Macros
Select Product Category	product_category_list	SELECT DISTINCT [product category] FROM products;	N/A
year	year_list	SELECT DISTINCT year(order_purchase_timestamp) AS [YEAR] FROM orders;	N/A
Generate report for product category and year	Command5	N/A	 Requery Control Name product_monthly_chart
Generate yearly product category orders	Command7	N/A	 Requery Control Name yearly_product_orders

Return to Main form	Command8	N/A	
			SetValue
			Item = [Forms]![Historical_data_form].[Visible]
			Expression = No
			SetValue
			Item = [Forms]![MainForm].[Visible]
			Expression = Yes
			StopMacro

product_monthly_chart	product_monthly_chart	SELECT MONTH(orders.order_purchase_timestamp) AS [Month], COUNT(orders.order_id) AS TotalOrders FROM (orders INNER JOIN order_items ON orders.order_id = order_items.order_id) INNER JOIN products ON products.product_id = order_items.product_id WHERE products.[product category] =	N/A

		Forms![Historical_data_form]![product_category_list].Value AND YEAR(orders.order_purchase_timestamp) = [Forms]![Historical_data_form]![year_list].Value GROUP BY MONTH(orders.order_purchase_timestamp), products.[product category];	

yearly_product_orders	yearly_product_orders	SELECT products.[product category], COUNT(orders.order_id) AS TotalOrders
		FROM (orders INNER JOIN order_items ON orders.order_id = order_items.order_id) INNER JOIN products ON products.product_id = order_items.product_id
		WHERE YEAR(orders.order_purchase_timestamp) = [Forms]![Historical_data_form]![year_list].Value
		GROUP BY products.[product category];	

Table:5.1 Historical Data form Details

Label/Title	Object Name	Event Property	Event Action
Select Product Category	product_category_list	N/A	Select Product category in the combo box
year	year_list	N/A	Select year in the combo box
Generate report for product category and year	Command5	On Click	Generate total no of order monthly line chart in a year for product category
Generate yearly product category orders	Command7	On Click	Generate yearly total orders for all product categories bar chart for states
Return to Main form	Command8	On Click	Return to main Form

Table:5.2 Historical Data form Actions

06 Payment Trends Form

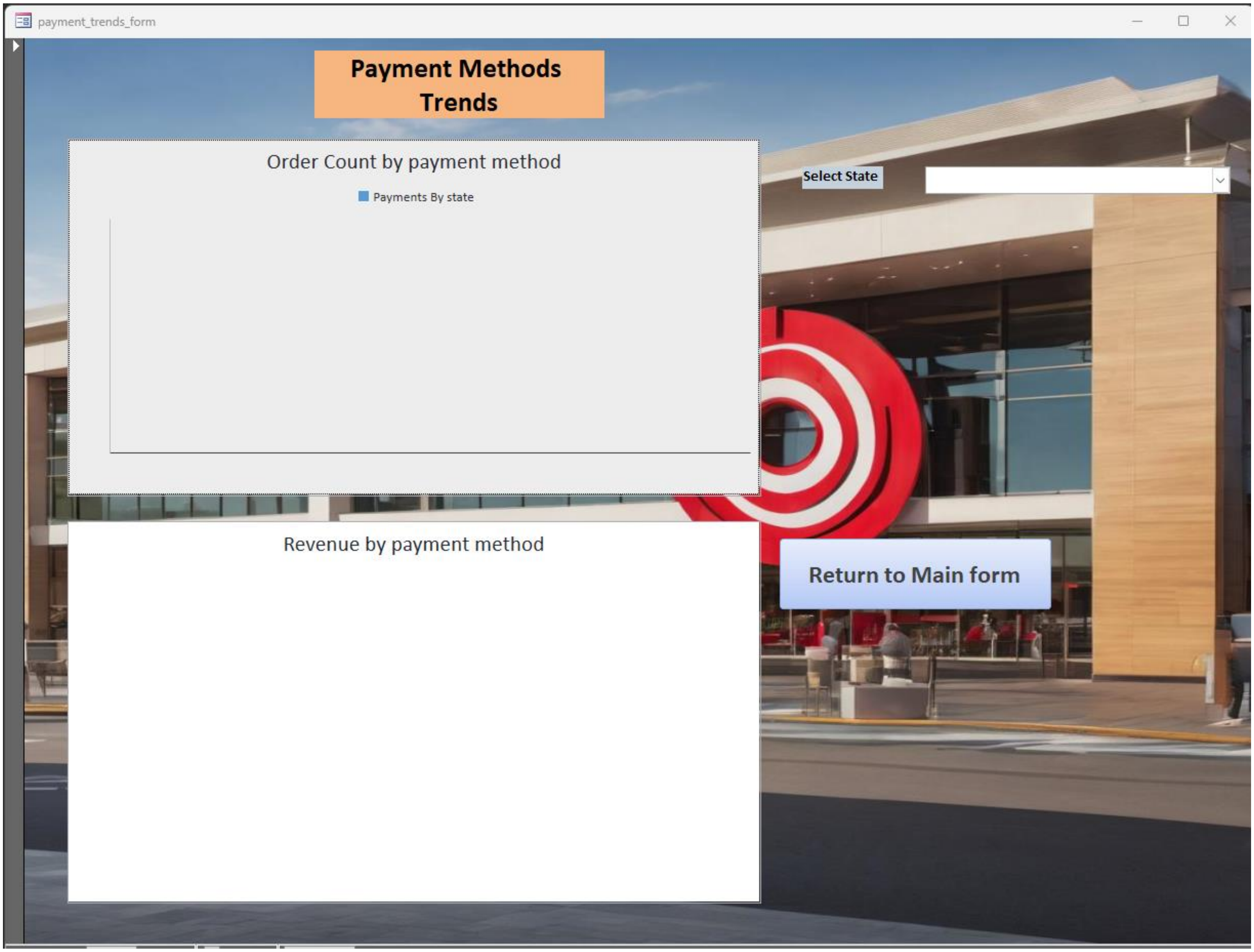
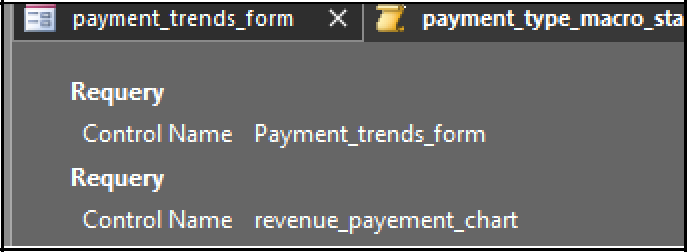


Figure:6.1 Payment Trends Form

Label/Title	Object Name	Attached Queries/ Row Source	Attached Macros
Select State	state_list	SELECT DISTINCT customer_state FROM customers;	 payment_trends_form × payment_type_macro_sta Requery Control Name Payment_trends_form Requery Control Name revenue_payement_chart

Order count by payment method - chart	Payment_trends_form	SELECT DISTINCT payments.payment_type , count(orders.order_id)
		FROM payments, customers, orders
		WHERE customers.customer_id = orders.customer_id and payments.order_id = orders.order_id and customers.customer_state = [Forms]![payment_trends_form]![state_list].Value GROUP BY payments.payment_type;
			N/A

Revenue by payment method	revenue_payement_chart	SELECT DISTINCT payments.payment_type, sum(payments.payment_value) AS TotalPrice
		FROM payments, customers, orders
		WHERE (((customers.customer_id)=orders.customer_id) And ((payments.order_id)=orders.order_id) And ((customers.customer_state)=Forms!payment_trends_form!state_list.Value)) GROUP BY payments.payment_type;
			N/A

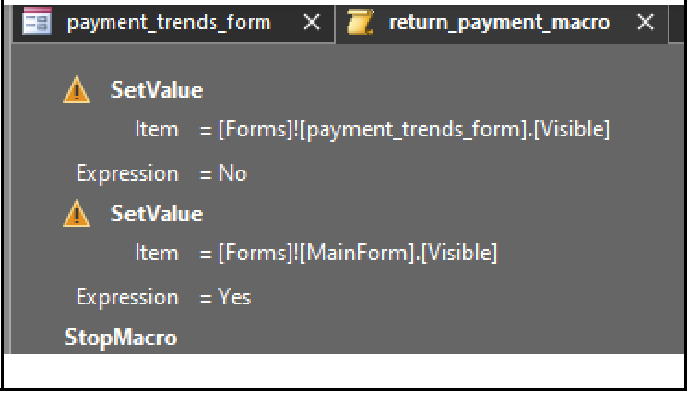
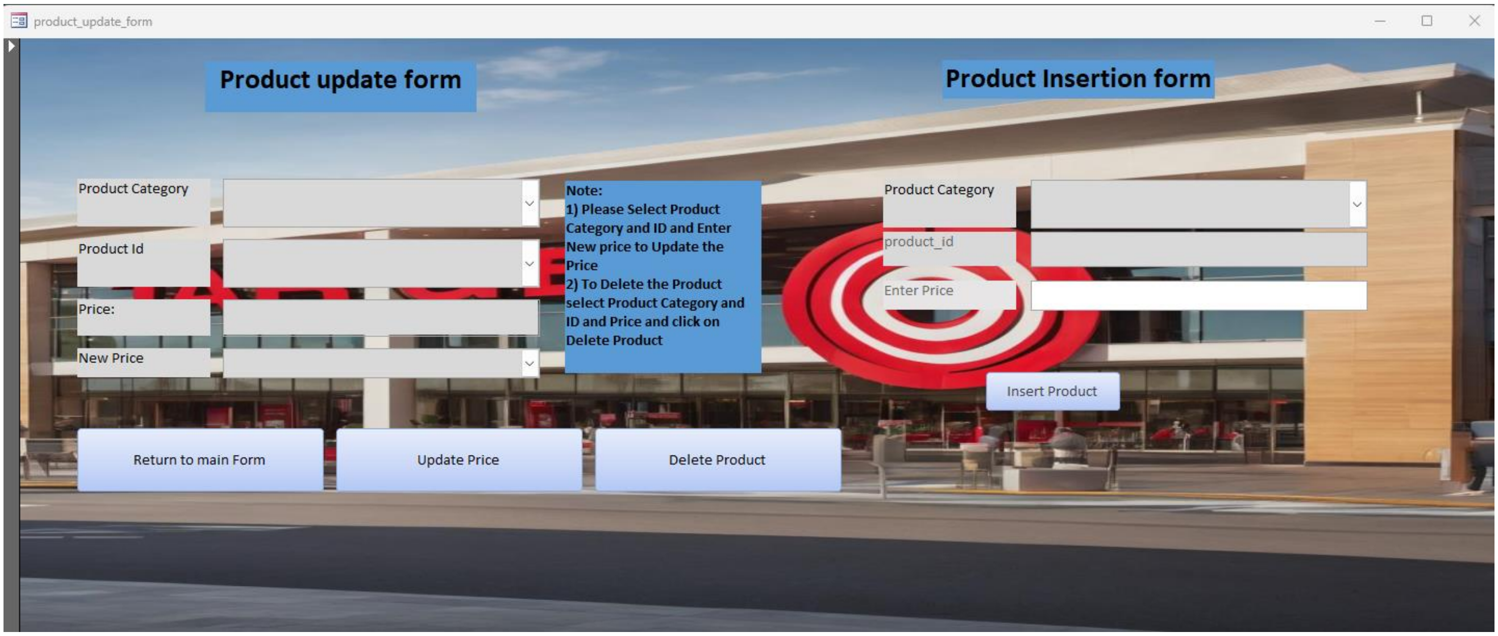
Return to Main form	Command4	N/A	
			
			SetValue
			Item = [Forms]![payment_trends_form].[Visible]
			Expression = No
			SetValue
			Item = [Forms]![MainForm].[Visible]
			Expression = Yes
			StopMacro

Table: ‘04 Payment trend form Details

Label/Title	Object Name	Event Property	Event Action
Select Product Category	product_category_list	N/A	Select state in the combo box- it automatically generates the graphs for order count and revenue by different payment methods in bar graph and pie chart
Return to Main form	Command8	On Click	Return to main Form

Table:6.1 Payment trend form Action

07 Product Update Form



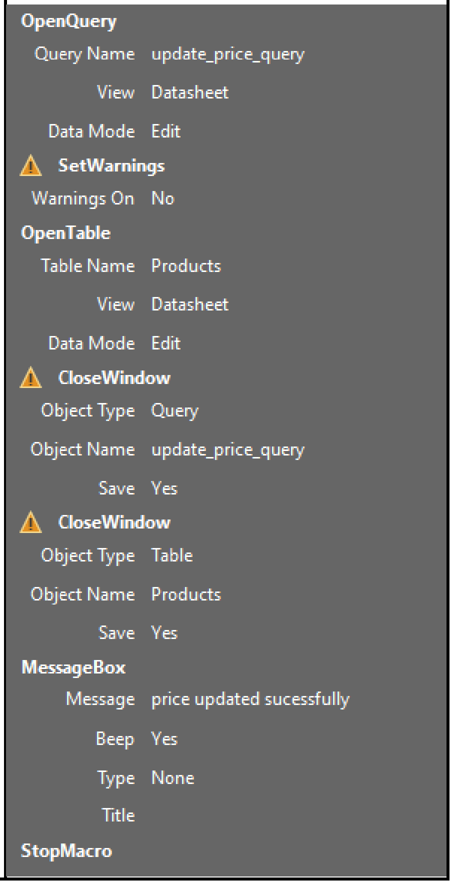
Note:
1) Please Select Product Category and ID and Enter New price to Update the Price
2) To Delete the Product select Product Category and ID and Price and click on Delete Product

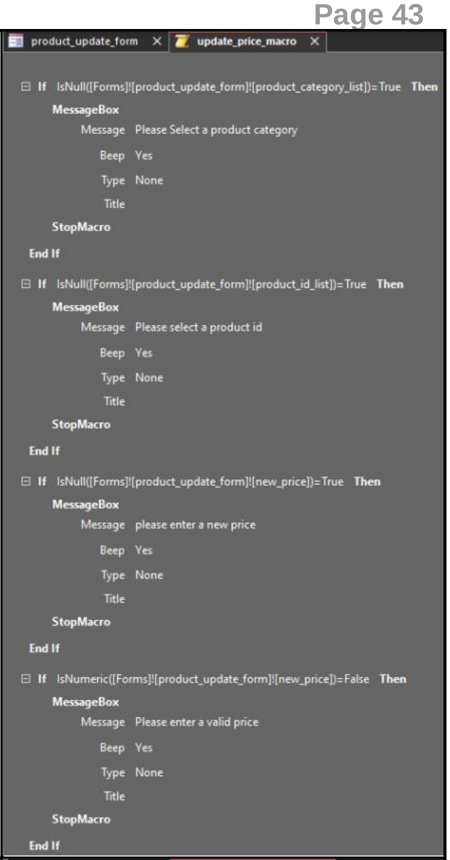
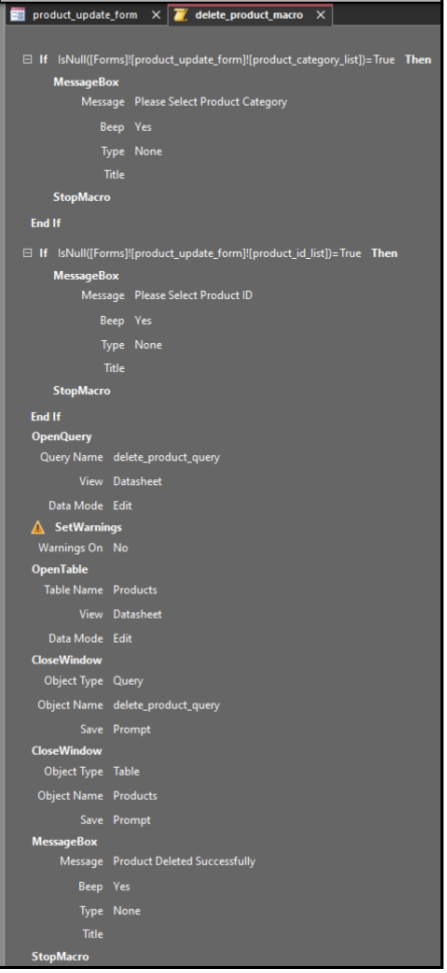
Figure:7.1 Product Update Form

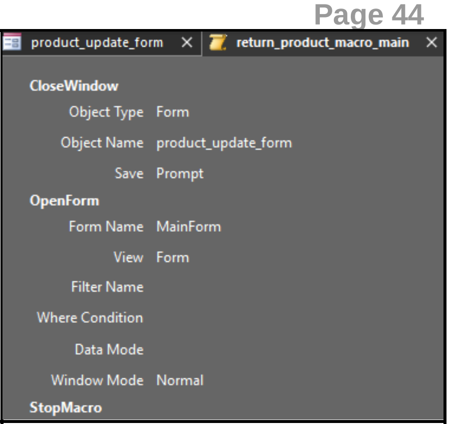
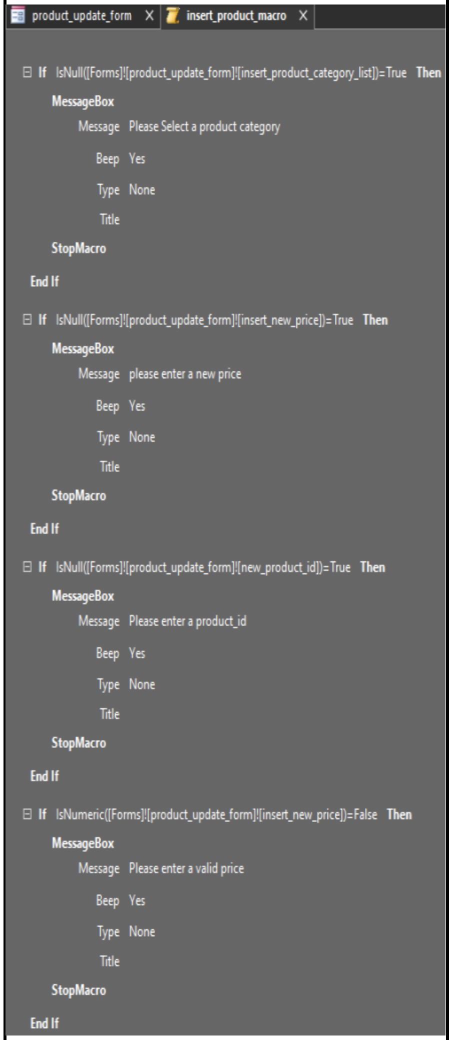
Label/Title	Object Name	Attached Queries/ Row Source	Attached Macros
Product Category	product_category_list	SELECT DISTINCT [product category] FROM products;	N/A

Product Id:	product_id_list	SELECT DISTINCT product_id
		FROM products
		WHERE [product category] = [Forms]![product_update_form]![product_category_list].Value;
			N/A

Price:	price	SELECT DISTINCT Price
		FROM products
		WHERE product_id = [Forms]![product_update_form]![product_id_list].Value;
			N/A

New Price	new_price	N/A	N/A
Update Price	Command18	N/A	 OpenQuery Query Name update_price_query View Datasheet Data Mode Edit ⚠ SetWarnings Warnings On No OpenTable Table Name Products View Datasheet Data Mode Edit ⚠ CloseWindow Object Type Query Object Name update_price_query Save Yes ⚠ CloseWindow Object Type Table Object Name Products Save Yes MessageBox Message price updated sucessfully Beep Yes Type None Title StopMacro

			 product_update_form update_price_macro If IsNull([Forms]![product_update_form]![product_category_list])=True Then MessageBox Message Please Select a product category Beep Yes Type None Title StopMacro End If If IsNull([Forms]![product_update_form]![product_id_list])=True Then MessageBox Message Please select a product id Beep Yes Type None Title StopMacro End If If IsNull([Forms]![product_update_form]![new_price])=True Then MessageBox Message please enter a new price Beep Yes Type None Title StopMacro End If If IsNumeric([Forms]![product_update_form]![new_price])=False Then MessageBox Message Please enter a valid price Beep Yes Type None Title StopMacro End If
Delete Product	Command30	N/A	 product_update_form delete_product_macro If IsNull([Forms]![product_update_form]![product_category_list])=True Then MessageBox Message Please Select Product Category Beep Yes Type None Title StopMacro End If If IsNull([Forms]![product_update_form]![product_id_list])=True Then MessageBox Message Please Select Product ID Beep Yes Type None Title StopMacro End If OpenQuery Query Name delete_product_query View Datasheet Data Mode Edit SetWarnings Warnings On No OpenTable Table Name Products View Datasheet Data Mode Edit CloseWindow Object Type Query Object Name delete_product_query Save Prompt CloseWindow Object Type Table Object Name Products Save Prompt MessageBox Message Product Deleted Successfully Beep Yes Type None Title StopMacro

Return to Main form	Command21	N/A	 product_update_form X return_product_macro_main X CloseWindow Object Type Form Object Name product_update_form Save Prompt OpenForm Form Name MainForm View Form Filter Name Where Condition Data Mode Window Mode Normal StopMacro
Product Category	insert_product_category_list	SELECT DISTINCT [product category] FROM products;	N/A
product_id	new_product_id	N/A	N/A
Enter Price	insert_new_price	N/A	N/A
Insert Product	Command67	N/A	 product_update_form X insert_product_macro X ⊟ If IsNull([Forms]![product_update_form]![insert_product_category_list])=True Then MessageBox Message Please Select a product category Beep Yes Type None Title StopMacro End If ⊟ If IsNull([Forms]![product_update_form]![insert_new_price])=True Then MessageBox Message please enter a new price Beep Yes Type None Title StopMacro End If ⊟ If IsNull([Forms]![product_update_form]![new_product_id])=True Then MessageBox Message Please enter a product_id Beep Yes Type None Title StopMacro End If ⊟ If IsNumeric([Forms]![product_update_form]![insert_new_price])=False Then MessageBox Message Please enter a valid price Beep Yes Type None Title StopMacro End If

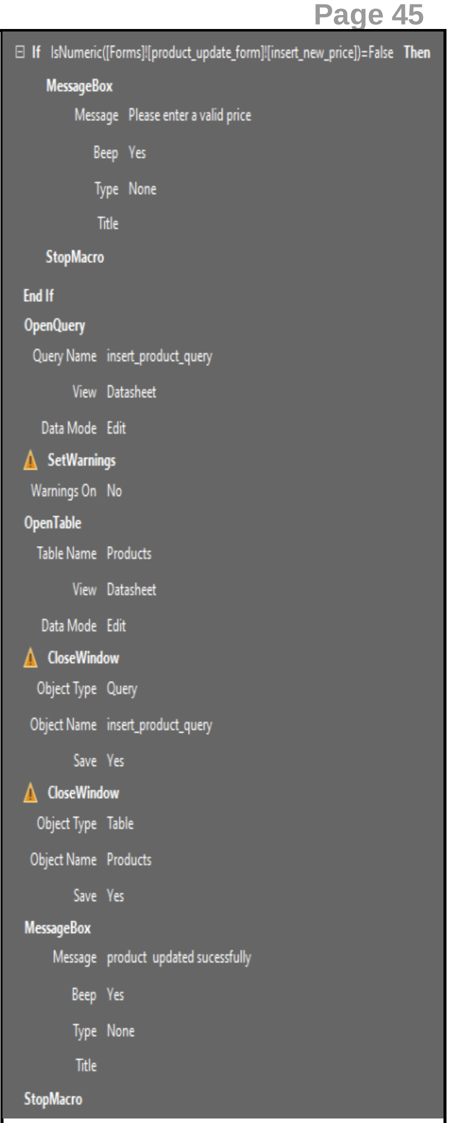
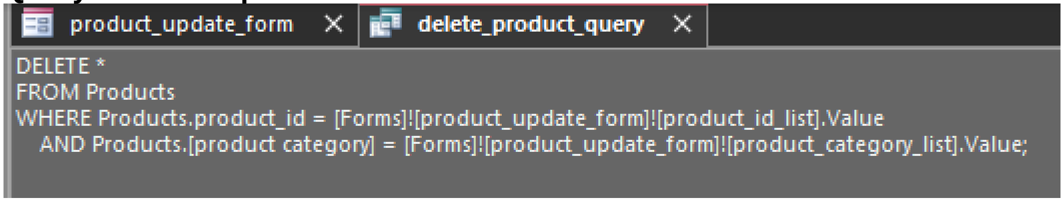
			 ⊟ **If** IsNumeric([Forms]![product_update_form]![insert_new_price])=False **Then** **MessageBox** Message Please enter a valid price Beep Yes Type None Title **StopMacro** **End If** **OpenQuery** Query Name insert_product_query View Datasheet Data Mode Edit ⚠ **SetWarnings** Warnings On No **OpenTable** Table Name Products View Datasheet Data Mode Edit ⚠ **CloseWindow** Object Type Query Object Name insert_product_query Save Yes ⚠ **CloseWindow** Object Type Table Object Name Products Save Yes **MessageBox** Message product updated sucessfully Beep Yes Type None Title **StopMacro**

Table:7.1 Product Update and Insertion Form Details

Label/Title	Object Name	Event Property	Event Action
Product Category	product_category_list	N/A	Select Product category in the combo box from shown product categories list
Product Id:	product_id_list	N/A	Select Product id in the combo box from shown product Id list
Price:	price	N/A	Shows the existing price of the product
New Price	new_price	N/A	Enter new Price to update the price of a product
Update Price	Command18	On Click	Updates the price of a product in products table based on product category, product id and price
Delete Product	Command30	On Click	Deletes the product in products table based on product category and product id
Return to Main M	Command21	On Click	Closes current form and opens Main Form
Product Category	insert_product_category_list	N/A	Select product category in the combo box in the shown list
product_id	new_product_id	N/A	Enter new product id which is guid, generate random guid and enter
Enter Price	insert_new_price	N/A	Enter price for the product that we are inserting

Table:7.2 Product Update and Insertion form Actions

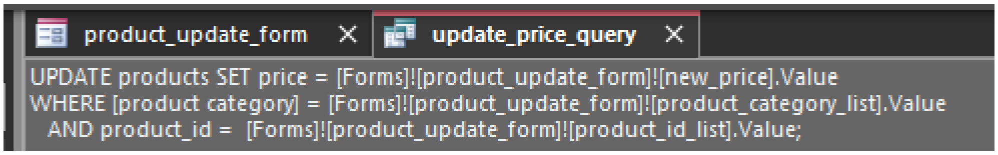
Query for Delete product macro:



```
DELETE *
FROM Products
WHERE Products.product_id = [Forms]![product_update_form]![product_id_list].Value
   AND Products.[product category] = [Forms]![product_update_form]![product_category_list].Value;
```

Figure 7.2: delete product query.

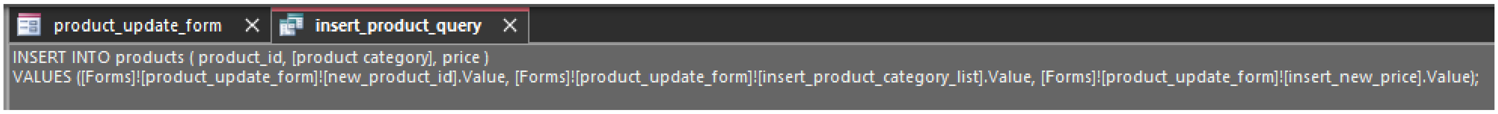
Query for Update product macro:



```
UPDATE products SET price = [Forms]![product_update_form]![new_price].Value
WHERE [product category] = [Forms]![product_update_form]![product_category_list].Value
  AND product_id =  [Forms]![product_update_form]![product_id_list].Value;
```

Figure 7.3: update product price query

Query for Insert product macro:



```
INSERT INTO products ( product_id, [product category], price )
VALUES ([Forms]![product_update_form]![new_product_id].Value, [Forms]![product_update_form]![insert_product_category_list].Value, [Forms]![product_update_form]![insert_new_price].Value);
```

Figure 7.4: Insert product query

08 Seller Update Form

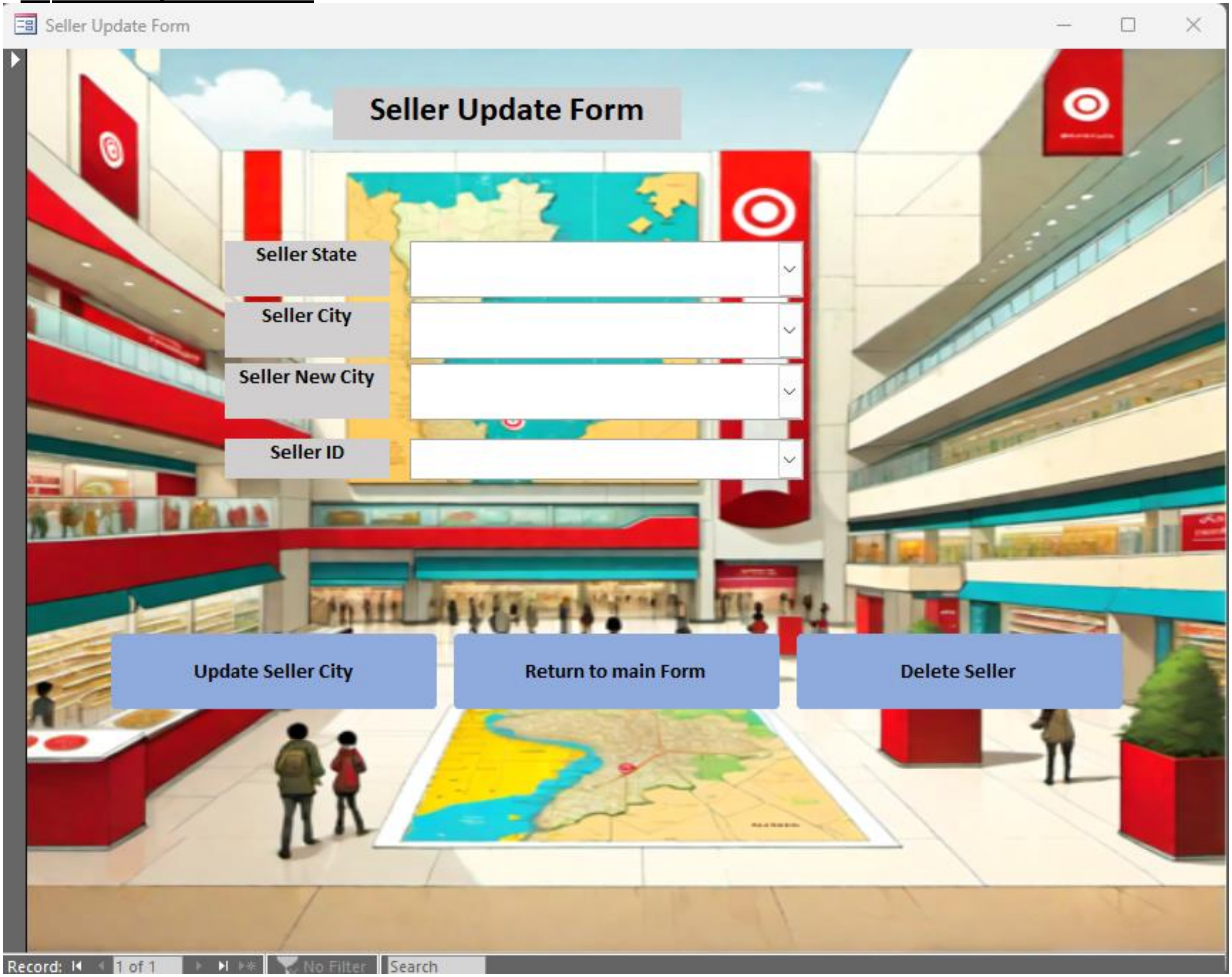
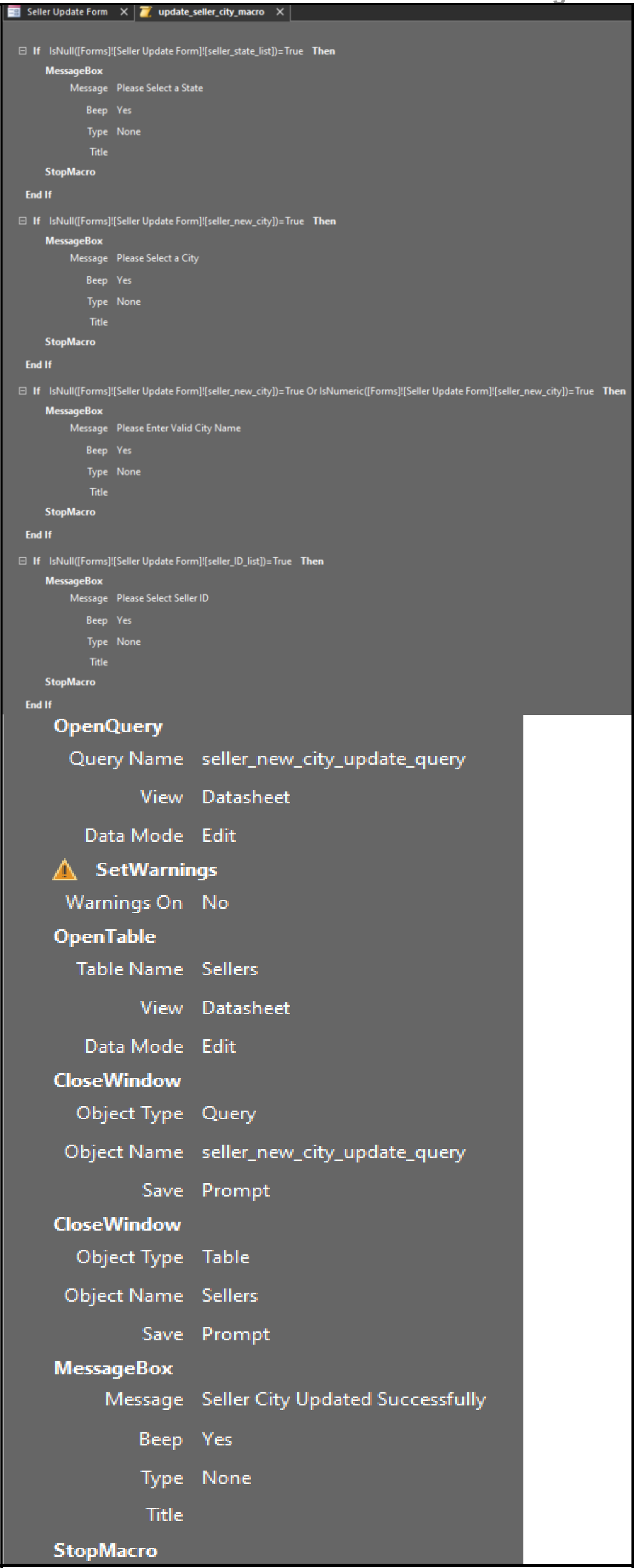
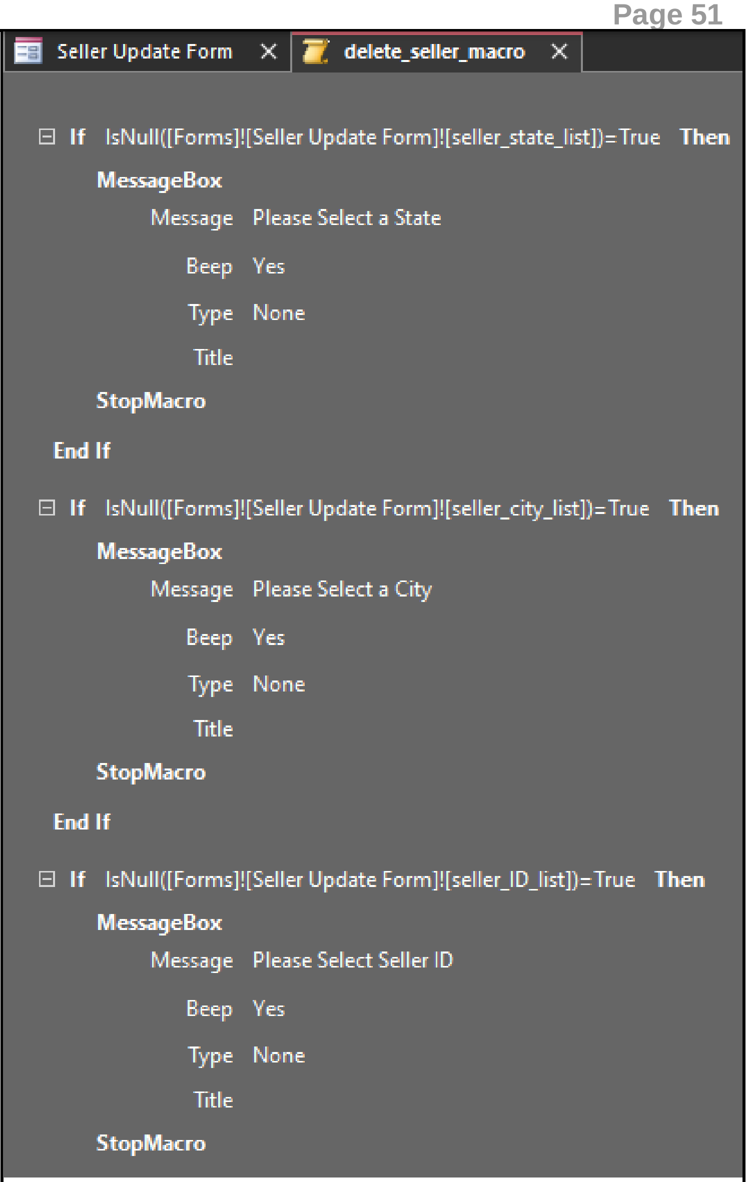


Figure:8.1 Seller Update Form

Label/Title	Object Name	Attached Queries	Attached Macros
Seller State	seller_state_list	SELECT DISTINCT seller_state FROM Sellers;	N/A
Seller City	seller_city_list	SELECT DISTINCT seller_city FROM Sellers WHERE seller_state = [Forms]![Seller Update Form]![seller_state_list].Value;	N/A
Seller New City	seller_new_city	N/A	N/A
Seller ID	seller_ID_list	SELECT DISTINCT seller_id FROM Sellers WHERE seller_state = [Forms]![Seller Update Form]![seller_state_list].Value and seller_city = [Forms]![Seller Update Form]![seller_city_list].Value;	N/A

Update Seller City	Command18	N/A	 Seller Update Form update_seller_city_macro If IsNull([Forms]![Seller Update Form]![seller_state_list])=True Then MessageBox Message Please Select a State Beep Yes Type None Title StopMacro End If If IsNull([Forms]![Seller Update Form]![seller_new_city])=True Then MessageBox Message Please Select a City Beep Yes Type None Title StopMacro End If If IsNull([Forms]![Seller Update Form]![seller_new_city])=True Or IsNumeric([Forms]![Seller Update Form]![seller_new_city])=True Then MessageBox Message Please Enter Valid City Name Beep Yes Type None Title StopMacro End If If IsNull([Forms]![Seller Update Form]![seller_ID_list])=True Then MessageBox Message Please Select Seller ID Beep Yes Type None Title StopMacro End If OpenQuery Query Name seller_new_city_update_query View Datasheet Data Mode Edit SetWarnings Warnings On No OpenTable Table Name Sellers View Datasheet Data Mode Edit CloseWindow Object Type Query Object Name seller_new_city_update_query Save Prompt CloseWindow Object Type Table Object Name Sellers Save Prompt MessageBox Message Seller City Updated Successfully Beep Yes Type None Title StopMacro
Delete Seller	Command59	N/A	

Seller Update Form ✕ delete_seller_macro ✕

```
⊟ If   IsNull([Forms]![Seller Update Form]![seller_state_list])=True   Then
        MessageBox
              Message   Please Select a State
                 Beep   Yes
                 Type   None
                 Title
        StopMacro
   End If
⊟ If   IsNull([Forms]![Seller Update Form]![seller_city_list])=True   Then
        MessageBox
              Message   Please Select a City
                 Beep   Yes
                 Type   None
                 Title
        StopMacro
   End If
⊟ If   IsNull([Forms]![Seller Update Form]![seller_ID_list])=True   Then
        MessageBox
              Message   Please Select Seller ID
                 Beep   Yes
                 Type   None
                 Title
        StopMacro
```

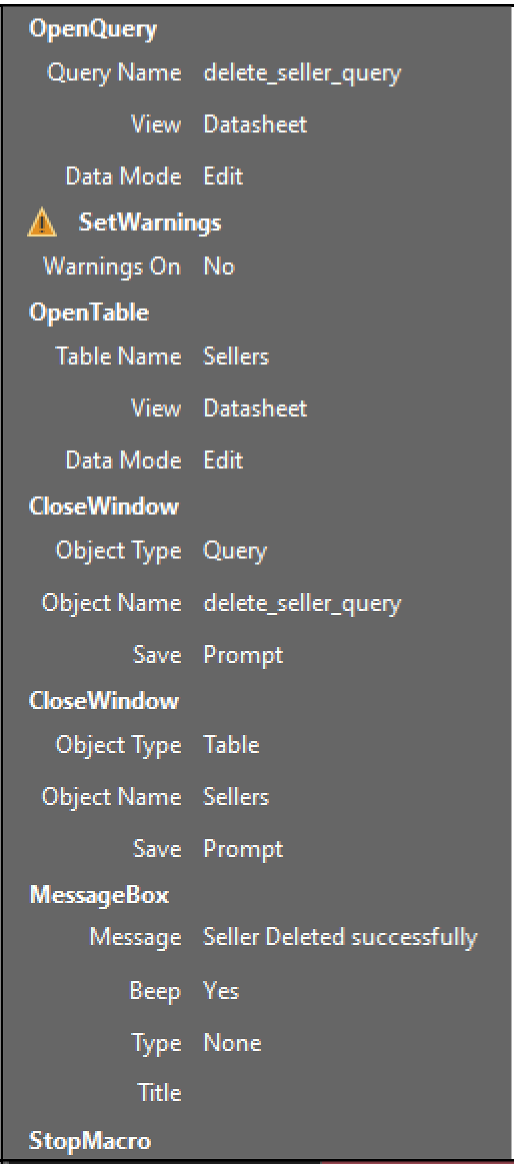
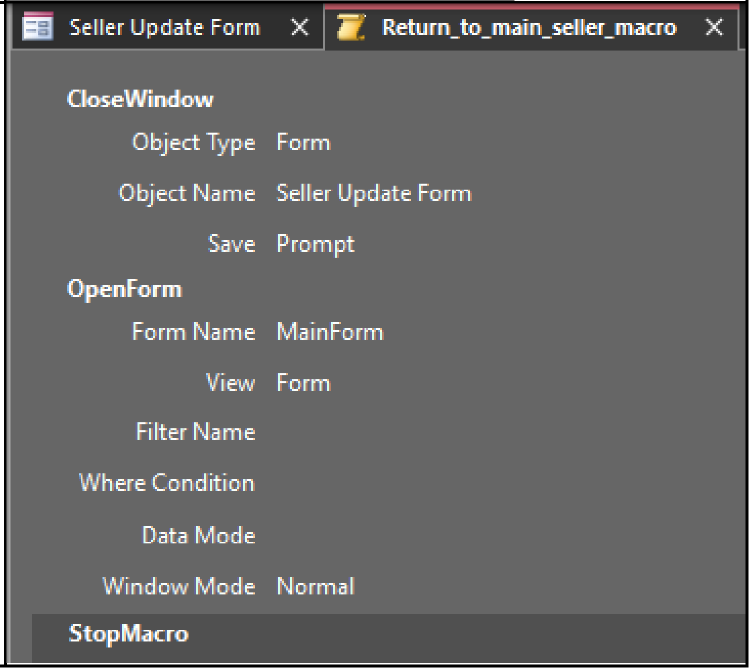
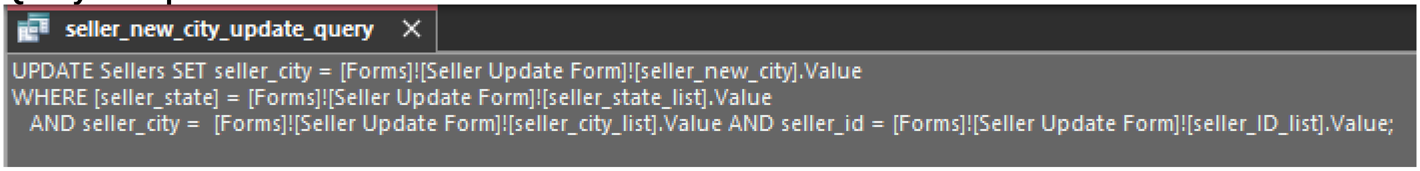
			 **OpenQuery** Query Name delete_seller_query View Datasheet Data Mode Edit ⚠ **SetWarnings** Warnings On No **OpenTable** Table Name Sellers View Datasheet Data Mode Edit **CloseWindow** Object Type Query Object Name delete_seller_query Save Prompt **CloseWindow** Object Type Table Object Name Sellers Save Prompt **MessageBox** Message Seller Deleted successfully Beep Yes Type None Title **StopMacro**
Return to main Form	Command21	N/A	 Seller Update Form ✕ Return_to_main_seller_macro ✕ **CloseWindow** Object Type Form Object Name Seller Update Form Save Prompt **OpenForm** Form Name MainForm View Form Filter Name Where Condition Data Mode Window Mode Normal **StopMacro**

Table:8.1 Seller Update Form Details

Label/Title	**Object Name**	**Event Property**	**Event Action**
Seller State	seller_state_list	N/A	Select Seller State
Seller City	seller_city_list	N/A	Select Seller City
Seller New City	seller_new_city	N/A	Enter New Seller New City
Seller ID	seller_ID_list	N/A	Select Seller ID
Update Seller City	Command18	OnClick	Updates seller city based on seller state, seller id, seller city
Delete Seller	Command59	OnClick	Deletes a seller based on seller state, seller id, seller city
Return to Main Form	Command21	OnClick	Takes user to main form

Table:8.2 Seller Update Form Actions

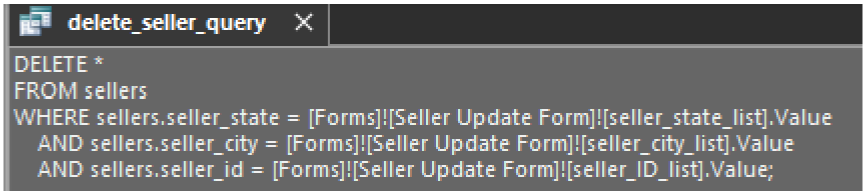
Query for Update seller macro:



```
UPDATE Sellers SET seller_city = [Forms]![Seller Update Form]![seller_new_city].Value
WHERE [seller_state] = [Forms]![Seller Update Form]![seller_state_list].Value
  AND seller_city =  [Forms]![Seller Update Form]![seller_city_list].Value AND seller_id = [Forms]![Seller Update Form]![seller_ID_list].Value;
```

Figure 8.2: update seller city query

Query for delete seller macro:



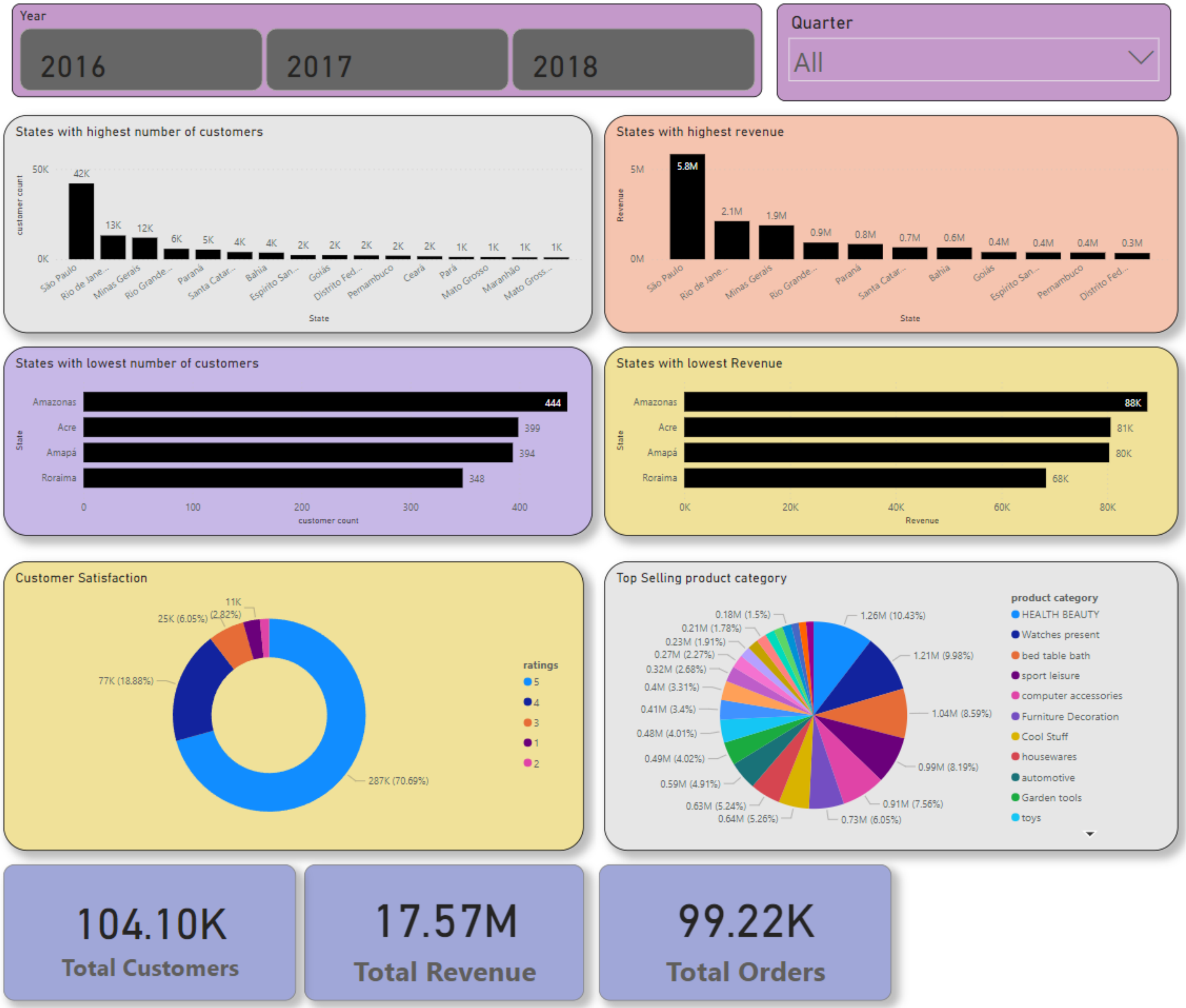
```
DELETE *
FROM sellers
WHERE sellers.seller_state = [Forms]![Seller Update Form]![seller_state_list].Value
   AND sellers.seller_city = [Forms]![Seller Update Form]![seller_city_list].Value
   AND sellers.seller_id = [Forms]![Seller Update Form]![seller_ID_list].Value;
```

Figure 8.3: delete seller query

09 Power BI Dashboard

PowerBI dashboard, the user has slicer options to choose state wise, year wise and quarter wise trends for different product categories, sellers and payment methods and order volumes etc.

Figure 9.1: Power BI Dashboard



Year

2016 2017 2018

Quarter

All

104.10K Total Customers

17.57M Total Revenue

99.22K Total Orders

FUTURE SCOPE OF PROJECT

There are multiple ways in which we can improve the application to provide more functionalities for the client, such as by.

- Analyze trends based on ordering times during a day (dawn, morning, afternoon, night)
- Explore opportunities to expand the marketplace by onboarding additional sellers or partnering with popular brands. Increasing the variety of products available can attract a broader customer base and drive higher sales.
- Refine customer segmentation based on purchasing behavior, demographics, and geography. Tailor marketing efforts and promotions to specific customer segments, increasing the likelihood of conversion and maximizing revenue from different customer groups.
- Regularly analyze market trends, customer feedback, and competitor activities. Being agile and adaptable to changing market dynamics is crucial for long-term success.
- Continue to emphasize a data-driven approach to decision-making. Regularly analyze sales data, customer feedback, and market trends to make informed decisions on pricing, promotions, and product offerings.

Project Report Summary

Objective: Develop a secure MS Access application for analyzing Target Corporation's operations in Brazil.

Authentication System:
Login_Form: Secure user login with options for new user registration and password reset.
reset_password_form: Password update functionality stored in 'lbl_login' table.
reg_form: New user registration with details updated to 'lbl_login' table.

Main Form Navigation:
Users directed to 'Main' form post-login, with options to navigate to various analysis forms and the PowerBI dashboard.
Forms include 'Regional trends,' 'Historical data,' 'Payment trends,' 'Product Price Update,' and 'Seller Update.'

Analysis Forms:
Regional trends Form: Explore trends by state and/or year through interactive charts.
Historical data Form: Analyze historical trends by product category and/or year.
Payment trends Form: Insights into preferred payment methods in selected states.
Product Price Update Form: Securely update product prices in the 'Products' table.
Seller Update Form: Update seller information and delete sellers from the 'sellers' table.

PowerBI Dashboard:
Access dynamic PowerBI dashboard with slicer options for state-wise, year-wise, and quarter-wise trends.
Trends include product categories, sellers, payment methods, and order volumes.

Conclusion: Comprehensive, user-friendly interface design with robust security measures enhances sales trend analysis capabilities, making it a powerful decision-making resource.