**Dynamic VBA Solution for Annual Sales, Delivery, and Payment Analysis**

Document Title: User Guide for VBA Macro Automation

Date: 28/11/2024

Prepared By: Tommy O’Gorman

**1. Introduction**

* **Purpose:**  
  This guide provides step-by-step instructions for using and interpreting the VBA macros developed for the company’s annual reporting.
* **Audience:**  
  This document is intended for employees and analysts who will run, review, or modify the macros to generate the annual report.

**2. Overview of the VBA Automation**

The VBA macros automate data consolidation, feature engineering, and analysis, producing key metrics and insights for stakeholders. The automated steps include:

* **Data Consolidation:** Importing and merging raw data from various sources (customers.csv, orders.csv, etc.) into a single workbook.
* **Feature Engineering:** Adding calculated columns such as timeFrameForDeliveries, Revenue, ProfitPercentage, and others to enrich the dataset.
* **SLA Compliance Analysis:** Assesses adherence to delivery timeframes (within 6 days ) by location.
* **Comparative Revenue Trends:** Summarizing monthly sales, payments, and probabilities (based on a relative frequency approach).
* **Dashboard Creation:** A summary tab for high-level insights.

**3. Accessing the Macros**

* **Location:**  
  The Excel workbook containing the macros is saved at [annual\_internal\_report.xlsm](https://tusmm-my.sharepoint.com/:x:/r/personal/k00278245_student_tus_ie/Documents/Group4/working%20files_Macros/VBA%20Macros/annual_internal_report.xlsm?d=w287ee417438e49b88a09560f97c58eb1&csf=1&web=1&e=rclIWF)
* **Files Required:**  
  Ensure all raw data files (customers.csv, orders.csv, etc.) are in the specified folder before running the macros.

**3. Accessing the Report**

* **Location:** [annual\_internal\_report.xlsm](https://tusmm-my.sharepoint.com/:x:/r/personal/k00278245_student_tus_ie/Documents/Group4/working%20files_Macros/VBA%20Macros/annual_internal_report.xlsm?d=w287ee417438e49b88a09560f97c58eb1&csf=1&web=1&e=rclIWF)

**Workbook and Worksheet Setup Requirements**

**Why This Matters:**  
The macros rely on specific sheet names and structure to run properly. If anything is off—like a tab being named wrong or missing—it could break the automation.

**Naming the Workbook**  
Make sure the workbook is named **annual\_internal\_report** and saved as a macro-enabled workbook.

**Tab Requirements**  
Here’s the list of tabs you’ll need. These names are case-sensitive, so double-check for typos or extra spaces.

1. **Refresh Report**
2. **Summary Dashboard**
3. **SLA\_Compliance**
4. **SLA\_Compliance\_Analysis**
5. **Sales Analysis**
6. **Payments Analysis**
7. **Master Sheet**
8. **Payments Master Sheet**
9. **Products & Profit Master Sheet**
10. **Orders**
11. **Customers**
12. **Employees**
13. **Offices**
14. **OrderDetails**
15. **Payments**
16. **ProductsLine**
17. **Products**

If the tabs aren’t set up exactly as listed, the macros won’t work properly.

**Conditional Formatting Rules**

In the **SLA Compliance Analysis**, **Sales Analysis**, and **Payments Analysis** tabs, conditional formatting is used to make key trends stand out. Here's how it works:

1. **Top 2 Values in Each Column**
   * Highlighted in green to show the best-performing offices or months. This helps you quickly spot what's working well and where strategies are succeeding.
2. **0% or Lowest Values**
   * Highlighted in red to flag problem areas. For example, if SLA compliance is 0% in a month, it’s a sign of a potential cashflow issue. Low values in Absolute Percentage Deviation are formatted as red, and this indicate a small difference between historical and most recent data, suggesting that comparisons between locations are similar over time, so this is a positive.

**Purpose and Use**  
The goal is to make the data easier to interpret at a glance. The green highlights show benchmarks or success stories you can learn from, while the red highlights pinpoint areas that need attention.

**Troubleshooting Common Issues**

If you’re having trouble running the macros or interpreting the results, here are a few tips:

1. **Macros Not Running**:
   * Double-check that all tab names match the required list.
   * Make sure the workbook is saved as .xlsm and macros are enabled.
2. **Data Not Loading**:
   * Check that your raw data files are saved in the correct folder. Look for typos in file names like customers.csv or orders.csv.
3. **Incorrect Year Assignment**:
   * If the macro isn’t assigning historical years correctly, look at cells A1:E2 in the Master Sheet tab. You might need to manually edit the years or rerun the year assignment macro (Macro\_06).

A screenshot of a computer

Description automatically generated

* + Editing the years is a completely acceptable approach. Just remember to proceed from macro 7 if entering the years manually. Historical years are defined as a year where we have a full 12 months of data. Current year is defined as where we have less then 12 months of data.

1. **Formatting Looks Off**:
   * If charts or tables don’t look right, try rerunning the relevant macro. For example, if the SLA visuals aren’t working, rerun Macro\_10. However, macro if macro 10 is the macro causing issues, feel free to skip it as what it creates is not essential to the report overall.
2. **Unexpected Errors**:
   * Go to the Developer tab and step through the macro to see where it’s breaking. Most issues will be related to missing data or incorrectly named tabs.

**Step-by-Step Guide to Running the Report**

Once everything is set up (correct tabs, workbook name, and raw data files), go to the tab titled *Refresh Report*: and click on each step sequentially.

**A screenshot of a graph

Description automatically generated**

* (**Note:** Button titled Step 3C: SLA Visuals can be finnicky. If it is not working for you, feel free to skip it as what it creates is not essential to the report overall.)
* **Note**: You will get a confirmation message after running each of the macros to let you know that each has been ran successfully.

**Explainer of what’s behind each button:**

1. **Step 1: Load Data**
   * This step runs **Macro\_01\_LoadData**. It pulls raw data from external CSV files (“orders.csv”, “customers.csv”, “offices.csv”, “orderdetails.csv”, “payments.csv”, “productlines.csv”, and “products.csv.”) into an “annual\_internal\_report.xlsm”.
   * What it does:
     + The code is automated to detect the folder path where the files are located. You must ensure that the CSV files are in the same folder as the workbook “annual\_internal\_report.xlsm” and the file names match with those specified in the macro. If CSV files are not in same folder it will pop up a error.
     + Clears any old data from the relevant tabs to avoid duplication.
     + Loads fresh data from the source files.
     + Confirms when the process is complete so you can move to the next step.
2. **Step 2: Data Preparation**
   * This step runs **Macro\_99\_DataPreparation**, which includes several sub-macros:
     + **Macro\_02\_Populate\_Master\_Data\_tab**: Structures the raw data in the Master Sheet.
     + **Macro\_03\_PopulatePaymentsMaster**: Prepares payment data for analysis.
     + **Macro\_04\_PopulateProductsAndProfitMaster**: Organises product and profit data.
     + **Macro\_04A\_FilterStatuses**: Cleans up any invalid or unnecessary data entries.
   * Why it matters: This step ensures all data is cleaned and ready for deeper analysis.
3. **Step 3A: SLA Compliance Data Preparation**
   * Runs **Macro\_99A\_SLAComplianceDataPreparation**, which:
     + Populates the SLA Compliance tab with raw data.
     + Assigns year labels based on the month of the order.
     + Adds formulas for SLA compliance percentages and other calculations.
4. **Step 3B: SLA Compliance Analysis**
   * Runs **Macro\_08\_SLAComplianceAnalysisStructure** and **Macro\_09\_SLAComplianceAnalysis** to:
     + Build and populate the SLA Compliance Analysis tab.
     + Calculate SLA compliance percentages for each office.
     + Highlight deviations and trends for quick insights.
5. **Step 3C: SLA Visuals**
   * Runs **Macro\_10\_SLAComplianceVisuals**, which:
     + Creates dynamic charts showing historical and recent SLA compliance trends.
     + Automatically adjusts chart layouts for clarity.
6. **Step 4: Payments Analysis**
   * Runs **Macro\_99\_PaymentsAnalysis**, including:
     + **Macro\_11\_CreatePaymentsAnalysisStructure**: Sets up the Payments Analysis tab.
     + **Macro\_12\_Populate\_Payments\_Analysis**: Fills in the tab with detailed payment data and trends.
7. **Step 5: Sales Analysis**
   * Runs:
     + Structures the Sales Analysis tab.
     + Populates it with metrics like sales volume, revenue, and profitability.
8. **Step 6: Summary Dashboard**
   * Runs:
     + **Macro\_15\_Create\_Summary\_Dashboard\_Structure**: Lays out the dashboard framework.
     + **Macro\_16\_Populate\_All\_Charts\_In\_Summary\_Dashboard**: Fills the dashboard with visuals and summaries.
     + **Macro\_17\_Apply\_All\_Conditional\_Formatting**: Highlights key metrics for easy interpretation.

If you have issues with any of the above, go to the developer tab as follows and click on each macro (1-17 ) individually:

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Important Notes:**

* **Do not run macros with "99" in their names individually.** These macros are specifically linked to the buttons on the *Refresh Report* page and are designed to run from there only. Running them independently may cause errors or incomplete execution.
* **No need to worry about pre-existing formulas or formatting.** Each macro is designed to fully recreate the relevant tabs, including all formulas, formatting, and charts where applicable. Everything required is embedded directly in the code, ensuring a clean and consistent output every time the macros are run.

**End of Main Instructions.**

1. **Additional Resources**

**Explainer for SLA Compliance Analysis Tab**

The **SLA Compliance Analysis Tab** is designed to evaluate how well each office meets Service Level Agreements (SLAs), focusing on on-time deliveries and their contribution to the overall operational performance. Here’s what each section means:

**Sections and Metrics:**

1. **Overall Historical SLA Compliance by Office:**
   * **Total Deliveries:** The number of total deliveries made by each office historically.
   * **On-time Deliveries:** The number of deliveries made within the SLA timeframe (6 days).
   * **SLA Compliance (%):** The percentage of deliveries made on time relative to total deliveries.
   * **Contribution to Total On-Time Deliveries (%):** Each office’s share of the total on-time deliveries across all offices historically.
2. **Recent SLA Compliance by Office:**
   * Similar to the historical section but focuses on the most recent year’s data, offering insights into current performance trends.
3. **SLA Compliance Comparison - Summary:**
   * This section compares historical and recent SLA compliance metrics and highlights **absolute percentage deviations** to show performance improvements or declines.
   * Contribution deviations show shifts in the role each office plays in on-time deliveries over time.
4. **Monthly SLA Performance Matrix:**
   * Displays historical SLA compliance data for each office by month to identify seasonal trends or challenges.
5. **Recent SLA Compliance Matrix:**
   * Recent performance trends for each office by month, useful for pinpointing areas needing immediate attention.
6. **Monthly Error Analysis:**
   * Highlights the months and offices with significant deviations from expected SLA performance, helping identify specific problem areas.

**Rules and Interpretation:**

* **Compliance Threshold:** Deliveries made within 6 days are considered "on-time."
* **High SLA Compliance:** Offices with consistently high percentages (e.g., Boston) indicate effective processes and resource management.
* **Low SLA Compliance:** Offices with low percentages (e.g., Tokyo or Sydney) need further investigation into operational challenges.
* **Monthly Trends:** Patterns of poor performance in certain months (e.g., July or August) suggest seasonal issues that need preemptive solutions.

**Explainer for Sales Analysis Tab**

The **Sales Analysis Tab** tracks office-wise and month-wise sales activities, revenue, and profitability. This data helps assess market performance and identify key areas for growth.

**Sections and Metrics:**

1. **Sales Activity:**
   * Represents the total number of sales transactions completed by each office during a given month.
   * This measures the operational efficiency and customer engagement of each location.
2. **Total Revenue:**
   * Calculated as the total monetary value of items sold (price × quantity).
   * Indicates how much revenue each office generates and helps highlight top-performing offices.
3. **Profitability:**
   * Derived as profit per item multiplied by the total items sold.
   * Provides insights into the efficiency of each office in generating profit.

**Rules and Interpretation:**

* **High Sales Volume with Low Profitability:** Indicates potential issues with pricing strategies or operational inefficiencies.
* **Low Sales Volume with High Profitability:** Suggests premium product sales or cost-effective processes.
* **Seasonal Trends:** Monitor patterns like increased sales in December for London, indicating opportunities for future planning.

**Explainer for Payments Analysis Tab**

The **Payments Analysis Tab** focuses on payment activities, emphasising the operational efficiency of collecting payments and their monetary value.

**Sections and Metrics:**

1. **Payment Activity:**
   * Tracks the total number of payments received by each office during a given month.
   * Indicates how active each office is in collecting payments and managing customer relationships.
2. **Payment Value:**
   * Represents the monetary value of the payments received.
   * Highlights the financial contribution of each office and helps assess the quality of the payments.

**Rules and Interpretation:**

* **High Payment Activity with Low Value:** Indicates smaller transactions or outstanding large payments.
* **Low Payment Activity with High Value:** Suggests fewer but larger payments, which might carry a higher risk of delayed payments.
* **Seasonal Trends:** Months like November and December often see spikes in payment activity, especially in high-performing offices like London and Boston.

Each tab serves as a crucial component of the annual report, providing insights into performance, trends, and areas needing attention, all tied to the company’s operational goals.

**Conditional Formatting Rules for SLA Compliance, Sales, and Payments Tabs**

In the **SLA Compliance Analysis**, **Sales Analysis**, and **Payments Analysis** tabs, conditional formatting has been applied to highlight key insights for easy interpretation. Here's an explanation of how the rules work and what they mean:

**Conditional Formatting Rules:**

1. **Top 2 Values in Each Column (Green):**
   * The top 2 performing values in each column are highlighted in **green**.
   * This is used to quickly identify the offices or months with the best results, whether it’s high SLA compliance, sales, revenue, profitability, or payment performance.
   * For example, in the SLA Compliance tab, Boston and London often appear in green due to their consistently strong SLA compliance rates.
2. **0% or Lowest Values (Red):**
   * Values of **0%** or the lowest performers in the column are highlighted in **red**.
   * This rule is designed to immediately flag underperforming months or offices that require attention.
   * For instance, in the Payments Analysis tab, any office with 0% payment activity is shown in red, indicating no payments received for that period.

**Purpose and Interpretation:**

* **Green Highlights:** These are the benchmarks and reflect the best-performing offices or months. They’re useful for identifying strategies or practices that are working well and can potentially be replicated across other locations.
* **Red Highlights:** These act as warning signals, showing areas that need urgent intervention. For example, an office with a red value in SLA compliance might indicate inefficiencies in their delivery processes or customer service.
* **Comparisons:** These conditional formats make it easy to compare performances across offices or months at a glance, helping management focus on strengths and weaknesses without manually reviewing every number.

The use of green and red formatting makes the data visually intuitive, helping stakeholders immediately understand what’s working and where improvements are needed. These formatting rules are consistent across all tabs to ensure a uniform approach to analysis.

**Explainer of individual macros:**

**1. Macro\_01\_LoadData**

This macro is the foundation of the automation process. It imports data from external CSV files into the workbook. Each dataset is assigned to its corresponding worksheet (e.g., Orders, Customers, Payments). It clears any existing data on these sheets and reloads fresh data, ensuring you're always working with the most up-to-date information.

**2. Macro\_02\_Populate\_Master\_Data\_tab**

This macro consolidates and structures data in the **Master Data** tab. It creates a unified dataset from multiple sources, which is critical for downstream analysis. Think of it as laying the groundwork for all the analyses to follow.

**3. Macro\_03\_PopulatePaymentsMaster**

This macro focuses specifically on preparing the **Payments Master** sheet. It aggregates payment-related data, ensuring the necessary calculations and summaries are ready for further processing.

**4. Macro\_04\_PopulateProductsAndProfitMaster**

Here, data related to products and profit is compiled and calculated. This macro pulls together details such as product categories, sales, and profits into a well-structured format for easy analysis.

**5. Macro\_04A\_FilterStatuses**

This macro cleans and filters statuses in your datasets. It standardizes values and removes any unnecessary or invalid entries, ensuring data accuracy and consistency.

**6. Macro\_05\_SLAMacro1\_PopulateSLACompliance**

This macro is the first step in the SLA Compliance Analysis. It populates the **SLA Compliance Summary** tab with raw data, setting the stage for more advanced analyses.

**7. Macro\_06\_SLAMacro2\_IdentifyYearsBasedOnMonths**

This macro helps categorize data into yearly buckets based on month information. It ensures that SLA compliance data is correctly grouped and organized by year.

**8. Macro\_07\_SLAMacro3\_PopulateFormulasinSLACompliance**

This macro fills in the formulas for SLA compliance metrics. It automates calculations for compliance percentages, deviations, and other critical indicators.

**9. Macro\_08\_SLAMacro4\_SetComplianceAnalysisStructure**

This macro creates the structure for the **SLA Compliance Analysis** tab. It sets up the necessary columns, headings, and formats to prepare for deeper analysis.

**10. Macro\_09\_SLAMacro5\_PopulateSLAComplianceAnalysis**

This macro populates the compliance analysis with data and formulas. It ensures that all metrics are calculated and displayed accurately in the **SLA Compliance Analysis** tab.

**11. Macro\_10\_SLAMacro6\_CreateSLAVisuals**

This macro is dedicated to creating visualizations for SLA Compliance. It generates two dynamic charts:

1. Historical On-Time Delivery Rates by Office
2. Historical vs. Recent SLA Compliance.

These charts are added to the **SLA Compliance Analysis** sheet for clear, actionable insights.

**12. Macro\_11\_PaymentsAnalysisMacro1\_CreatePaymentsAnalysisStructure**

This macro sets up the **Payments Analysis** tab, creating headers and a structure for monthly payment data. It prepares the space for subsequent population of data.

**13. Macro\_12\_Populate\_Payments\_Analysis**

This macro fills in the **Payments Analysis** tab with calculated payment metrics, including totals and trends over time.

**14. Macro\_13\_Macro\_SalesAnalysis\_CreateStructure**

This macro structures the **Sales Analysis** tab. It creates headers and organizes the layout to display revenue, profit, and sales trends.

**15. Macro\_14\_Populate\_Sales\_Analysis**

This macro populates the **Sales Analysis** tab with sales data and formulas. It automates calculations such as monthly revenue, product sales, and profit percentages.

**16. Macro\_15\_Create\_Summary\_Dashboard\_With\_Chart\_Structures**

This macro designs the **Summary Dashboard**, setting up the structure, headers, and placeholders for all key charts and summaries.

**17. Macro\_16\_Populate\_All\_Charts\_In\_Summary\_Dashboard**

This macro fills the Summary Dashboard with data-driven charts. It ensures all summaries, trends, and KPIs are visualized effectively.

**18. Macro\_17\_Apply\_All\_Conditional\_Formatting**

This macro applies conditional formatting across the workbook. It highlights important values, such as SLA compliance deviations and top performers, using color-coded rules to improve readability.

**19. Macro Code** (also available in notepad document provided):

Sub Macro\_01\_LoadData()

Dim sourceFiles As Variant

Dim targetSheets As Variant

Dim i As Integer

' Disable screen updating to prevent flickering

Application.ScreenUpdating = False

' Arrays with source file names and corresponding target sheets

sourceFiles = Array("orders.csv", "customers.csv", "employees.csv", "offices.csv", \_

"orderdetails.csv", "payments.csv", "productlines.csv", "products.csv")

targetSheets = Array("Orders", "Customers", "Employees", "Offices", \_

"OrderDetails", "Payments", "ProductsLine", "Products")

' Loop through each file and its corresponding sheet

For i = LBound(sourceFiles) To UBound(sourceFiles)

' Activate destination workbook and clear the corresponding sheet

Windows("annual\_internal\_report.xlsm").Activate

Sheets(targetSheets(i)).Select

Cells.ClearContents

' Open the source file, this code is to open the files located in the same folder

Workbooks.Open Filename:=ThisWorkbook.Path & "\" & sourceFiles(i)

' Activate source file, copy data, and close it

Workbooks(sourceFiles(i)).Activate

Cells.Select

Selection.Copy

' Return to destination workbook and paste values

Windows("annual\_internal\_report.xlsm").Activate

Sheets(targetSheets(i)).Select

Cells.Select

Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks:=False, Transpose:=False

' Clear copy mode to optimize memory

Application.CutCopyMode = False

' Close the CSV file

Workbooks(sourceFiles(i)).Close SaveChanges:=False

Next i

' Return user to the "Refresh Report" tab

Sheets("Refresh Report").Activate

' Re-enable screen updating

Application.ScreenUpdating = True

' Display a message box to notify the user

MsgBox "Data loading complete.", vbInformation, "Load Data Complete"

End Sub

Sub Macro\_02\_Populate\_Master\_Data\_tab()

Application.Calculation = xlCalculationManual

Application.ScreenUpdating = False

Application.EnableEvents = False

Application.DisplayStatusBar = False

Dim wsOrderDetails As Worksheet, wsOrders As Worksheet, wsCustomers As Worksheet

Dim wsEmployees As Worksheet, wsOffices As Worksheet, wsPayments As Worksheet, wsProducts As Worksheet

Dim wsMaster As Worksheet

Dim lastRowMaster As Long, lastRowOrders As Long, lastRowCustomers As Long

Dim lastRowEmployees As Long, lastRowOffices As Long, lastRowPayments As Long, lastRowProducts As Long

' Define worksheets

Set wsMaster = ThisWorkbook.Sheets("Master Sheet")

Set wsOrderDetails = ThisWorkbook.Sheets("OrderDetails")

Set wsOrders = ThisWorkbook.Sheets("Orders")

Set wsCustomers = ThisWorkbook.Sheets("Customers")

Set wsEmployees = ThisWorkbook.Sheets("Employees")

Set wsOffices = ThisWorkbook.Sheets("Offices")

Set wsPayments = ThisWorkbook.Sheets("Payments")

Set wsProducts = ThisWorkbook.Sheets("Products")

' Clear the Master Sheet before running the macro

wsMaster.Cells.Clear

' Set headers for the Master Sheet

wsMaster.Range("A1:AC1").Value = Array("orderNumber", "productCode", "quantityOrdered", "priceEach", "orderLineNumber", \_

"orderDate", "requiredDate", "shippedDate", "status", "customerNumber", \_

"salesRepEmployeeNumber", "creditLimit", "city", "officeCode", "officeCity", \_

"paymentDate", "amount", "productLine", "quantityInStock", "MSRP", \_

"timeFrameforDeliveries", "internalStatus", "Year", "Month", "buyPrice", "Revenue", "ProfitPercentage", "Profit>Median", "Profit")

' Copy OrderDetails data to Master Sheet including the following columns in this order: orderNumber, productCode, quantityOrdered, priceEach, orderLineNumber

lastRowMaster = wsOrderDetails.Cells(wsOrderDetails.Rows.Count, "A").End(xlUp).Row

wsOrderDetails.Range("A2:E" & lastRowMaster).Copy wsMaster.Range("A2")

' Get last row of Master Sheet after copying

lastRowMaster = wsMaster.Cells(wsMaster.Rows.Count, "A").End(xlUp).Row

' Apply VLOOKUP formulas to columns in the Master Sheet

lastRowOrders = wsOrders.Cells(wsOrders.Rows.Count, "A").End(xlUp).Row

wsMaster.Range("F2:F" & lastRowMaster).formula = "=IFERROR(VLOOKUP(A2,Orders!A:B,2,FALSE),"""")" ' orderDate

wsMaster.Range("G2:G" & lastRowMaster).formula = "=IFERROR(VLOOKUP(A2,Orders!A:C,3,FALSE),"""")" ' requiredDate

wsMaster.Range("H2:H" & lastRowMaster).formula = "=IFERROR(VLOOKUP(A2,Orders!A:D,4,FALSE),"""")" ' shippedDate

wsMaster.Range("I2:I" & lastRowMaster).formula = "=IFERROR(VLOOKUP(A2,Orders!A:E,5,FALSE),"""")" ' status

wsMaster.Range("J2:J" & lastRowMaster).formula = "=IFERROR(VLOOKUP(A2,Orders!A:G,7,FALSE),"""")" ' customerNumber

lastRowCustomers = wsCustomers.Cells(wsCustomers.Rows.Count, "A").End(xlUp).Row

wsMaster.Range("K2:K" & lastRowMaster).formula = "=IFERROR(VLOOKUP(J2,Customers!A:L,12,FALSE),"""")" ' salesRepEmployeeNumber

wsMaster.Range("L2:L" & lastRowMaster).formula = "=IFERROR(VLOOKUP(J2,Customers!A:M,13,FALSE),"""")" ' creditLimit

wsMaster.Range("M2:M" & lastRowMaster).formula = "=IFERROR(VLOOKUP(J2,Customers!A:H,8,FALSE),"""")" ' city

lastRowEmployees = wsEmployees.Cells(wsEmployees.Rows.Count, "A").End(xlUp).Row

wsMaster.Range("N2:N" & lastRowMaster).formula = "=IFERROR(VLOOKUP(K2,Employees!A:F,6,FALSE),"""")" ' officeCode

wsMaster.Range("O2:O" & lastRowMaster).formula = "=IFERROR(VLOOKUP(N2,Offices!A:B,2,FALSE),"""")" ' officeCity

lastRowPayments = wsPayments.Cells(wsPayments.Rows.Count, "A").End(xlUp).Row

wsMaster.Range("P2:P" & lastRowMaster).formula = "=IFERROR(VLOOKUP(J2,Payments!A:C,3,FALSE),"""")" ' paymentDate

lastRowProducts = wsProducts.Cells(wsProducts.Rows.Count, "A").End(xlUp).Row

wsMaster.Range("Q2:Q" & lastRowMaster).formula = "=IFERROR(VLOOKUP(J2,Payments!A:D,4,FALSE),"""")" ' Amount

wsMaster.Range("R2:R" & lastRowMaster).formula = "=IFERROR(VLOOKUP(B2,Products!A:C,3,FALSE),"""")" ' ProductLine

wsMaster.Range("S2:S" & lastRowMaster).formula = "=IFERROR(VLOOKUP(B2,Products!A:G,7,FALSE),"""")" ' QuantityInStock

wsMaster.Range("T2:T" & lastRowMaster).formula = "=IFERROR(VLOOKUP(B2,Products!A:I,9,FALSE),"""")" ' MSRP

wsMaster.Range("Y2:Y" & lastRowMaster).formula = "=IFERROR(VLOOKUP(B2,Products!A:I,8,FALSE),"""")" ' BuyPrice

' Calculated Columns

wsMaster.Range("U2:U" & lastRowMaster).formula = "=G2-H2" ' TimeFrameForDeliveries

wsMaster.Range("V2:V" & lastRowMaster).formula = "=IF(OR(I2=""Shipped"",I2=""Resolved""),IF(U2<=6,""Yes"",""No""),"""")" ' InternalStatus

wsMaster.Range("W2:W" & lastRowMaster).formula = "=YEAR(F2)" ' Year

wsMaster.Range("X2:X" & lastRowMaster).formula = "=MONTH(F2)" ' Month

wsMaster.Range("Z2:Z" & lastRowMaster).formula = "=C2\*D2" ' Revenue

wsMaster.Range("AA2:AA" & lastRowMaster).formula = "=((D2-Y2)/Y2)\*100" ' ProfitPercentage

wsMaster.Range("AB2:AB" & lastRowMaster).formula = "=IF(AA2>=67, 1,0)" ' Profit>Median

wsMaster.Range("AC2:AC" & lastRowMaster).formula = "=C2\*(D2-Y2)" ' Profit

' Format columns F, G, H, and P as dates

wsMaster.Columns("F").NumberFormat = "mm/dd/yyyy"

wsMaster.Columns("G").NumberFormat = "mm/dd/yyyy"

wsMaster.Columns("H").NumberFormat = "mm/dd/yyyy"

wsMaster.Columns("P").NumberFormat = "mm/dd/yyyy"

' Autofit columns for better visibility

wsMaster.Columns("A:X").AutoFit

Application.Calculation = xlCalculationAutomatic

Application.ScreenUpdating = True

Application.EnableEvents = True

Application.DisplayStatusBar = True

End Sub

Sub Macro\_03\_PopulatePaymentsMaster()

Application.Calculation = xlCalculationManual

Application.ScreenUpdating = False

Application.EnableEvents = False

Application.DisplayStatusBar = False

Dim wsPayments As Worksheet

Dim wsMaster As Worksheet

Dim lastRowPayments As Long

Dim lastRowMaster As Long

' Set references to the worksheets

On Error Resume Next

Set wsPayments = ThisWorkbook.Sheets("Payments")

Set wsMaster = ThisWorkbook.Sheets("Payments Master Sheet")

On Error GoTo 0

If wsPayments Is Nothing Or wsMaster Is Nothing Then

MsgBox "One or more required sheets ('Payments' or 'Payments Master Sheet') are missing.", vbCritical

Exit Sub

End If

' Determine the last row of data in the Payments sheet

lastRowPayments = wsPayments.Cells(wsPayments.Rows.Count, "A").End(xlUp).Row

' Clear the Payments Master Sheet before pasting new data

wsMaster.Cells.Clear

' Copy data from Payments sheet to Payments Master Sheet

wsPayments.Range("A1:D" & lastRowPayments).Copy Destination:=wsMaster.Range("A1")

' Add headers for Year, Month, and OfficeCity

With wsMaster

.Range("E1").Value = "Year"

.Range("F1").Value = "Month"

.Range("G1").Value = "OfficeCity"

' Determine the last row of data in Payments Master Sheet

lastRowMaster = .Cells(.Rows.Count, "A").End(xlUp).Row

' Add Year formula in column E

.Range("E2:E" & lastRowMaster).formula = "=YEAR(C2)"

' Add Month formula in column F

.Range("F2:F" & lastRowMaster).formula = "=MONTH(C2)"

' Add VLOOKUP formula for OfficeCity in column G

' Assumes the lookup table is on the 'Master Sheet' with key in column D and values in column I

.Range("G2:G" & lastRowMaster).formula = "=VLOOKUP(A2,'Master Sheet'!J:O,6,FALSE)"

' Format the date column (Column C)

.Columns("C:C").NumberFormat = "dd/mm/yyyy;@"

' Auto-fit all columns

.Columns("A:G").EntireColumn.AutoFit

Application.Calculation = xlCalculationAutomatic

Application.ScreenUpdating = True

Application.EnableEvents = True

Application.DisplayStatusBar = True

End With

End Sub

Sub Macro\_04\_PopulateProductsAndProfitMaster()

Application.Calculation = xlCalculationManual

Application.ScreenUpdating = False

Application.EnableEvents = False

Application.DisplayStatusBar = False

' Declare variables

Dim wsOrderDetails As Worksheet

Dim wsMaster As Worksheet

Dim wsOrders As Worksheet

Dim lastRowOrderDetails As Long

Dim lastRowMaster As Long

' Set worksheets

Set wsOrderDetails = Sheets("OrderDetails")

Set wsMaster = Sheets("Products & Profit Master Sheet")

Set wsOrders = Sheets("Orders")

' Find the last row in OrderDetails

lastRowOrderDetails = wsOrderDetails.Cells(wsOrderDetails.Rows.Count, "A").End(xlUp).Row

' Copy data from OrderDetails to the master sheet

wsOrderDetails.Range("A1:E" & lastRowOrderDetails).Copy

wsMaster.Range("A1").PasteSpecial Paste:=xlPasteValues

Application.CutCopyMode = False ' Clear clipboard

' Add headers for additional columns

wsMaster.Range("F1").Value = "Order Date"

wsMaster.Range("G1").Value = "Status"

wsMaster.Range("H1").Value = "buyPrice"

wsMaster.Range("I1").Value = "totalRevenue"

wsMaster.Range("J1").Value = "totalProfit"

wsMaster.Range("K1").Value = "Year"

wsMaster.Range("L1").Value = "Month"

wsMaster.Range("M1").Value = "officeCity"

' Find the last row in the master sheet

lastRowMaster = wsMaster.Cells(wsMaster.Rows.Count, "A").End(xlUp).Row

' Populate the 'Order Date' column using VLOOKUP with absolute references

wsMaster.Range("F2:F" & lastRowMaster).formula = \_

"=VLOOKUP($A2,Orders!$A:$B,2,FALSE)"

' Populate the 'Status' column using VLOOKUP with absolute references

wsMaster.Range("G2:G" & lastRowMaster).formula = \_

"=VLOOKUP($A2,Orders!$A:$E,5,FALSE)"

' Populate the 'buyPrice' column using VLOOKUP with absolute references

wsMaster.Range("H2:H" & lastRowMaster).formula = \_

"=VLOOKUP(B2,Products!A:H,8)"

' Populate the 'Total Revenue' column

wsMaster.Range("I2:I" & lastRowMaster).formula = \_

"=C2\*D2"

' Populate the 'Total Profit' column

wsMaster.Range("J2:J" & lastRowMaster).formula = \_

"=(D2-H2)\*C2"

' Populate the 'Year' column

wsMaster.Range("K2:K" & lastRowMaster).formula = \_

"=year(F2)"

' Populate the 'Month' column

wsMaster.Range("L2:L" & lastRowMaster).formula = \_

"=month(F2)"

' Populate the 'officeCity' column

wsMaster.Range("M2:M" & lastRowMaster).formula = \_

"=VLOOKUP(A2,'Master Sheet'!A:O,15)"

' Format the 'Order Date' column as date

wsMaster.Columns("F:F").NumberFormat = "dd/mm/yyyy;@"

' Autofit columns for better readability

wsMaster.Columns("A:G").AutoFit

Application.Calculation = xlCalculationAutomatic

Application.ScreenUpdating = True

Application.EnableEvents = True

Application.DisplayStatusBar = True

End Sub

Sub Macro\_04\_PopulateProductsAndProfitMaster()

Application.Calculation = xlCalculationManual

Application.ScreenUpdating = False

Application.EnableEvents = False

Application.DisplayStatusBar = False

' Declare variables

Dim wsOrderDetails As Worksheet

Dim wsMaster As Worksheet

Dim wsOrders As Worksheet

Dim lastRowOrderDetails As Long

Dim lastRowMaster As Long

' Set worksheets

Set wsOrderDetails = Sheets("OrderDetails")

Set wsMaster = Sheets("Products & Profit Master Sheet")

Set wsOrders = Sheets("Orders")

' Find the last row in OrderDetails

lastRowOrderDetails = wsOrderDetails.Cells(wsOrderDetails.Rows.Count, "A").End(xlUp).Row

' Copy data from OrderDetails to the master sheet

wsOrderDetails.Range("A1:E" & lastRowOrderDetails).Copy

wsMaster.Range("A1").PasteSpecial Paste:=xlPasteValues

Application.CutCopyMode = False ' Clear clipboard

' Add headers for additional columns

wsMaster.Range("F1").Value = "Order Date"

wsMaster.Range("G1").Value = "Status"

wsMaster.Range("H1").Value = "buyPrice"

wsMaster.Range("I1").Value = "totalRevenue"

wsMaster.Range("J1").Value = "totalProfit"

wsMaster.Range("K1").Value = "Year"

wsMaster.Range("L1").Value = "Month"

wsMaster.Range("M1").Value = "officeCity"

' Find the last row in the master sheet

lastRowMaster = wsMaster.Cells(wsMaster.Rows.Count, "A").End(xlUp).Row

' Populate the 'Order Date' column using VLOOKUP with absolute references

wsMaster.Range("F2:F" & lastRowMaster).formula = \_

"=VLOOKUP($A2,Orders!$A:$B,2,FALSE)"

' Populate the 'Status' column using VLOOKUP with absolute references

wsMaster.Range("G2:G" & lastRowMaster).formula = \_

"=VLOOKUP($A2,Orders!$A:$E,5,FALSE)"

' Populate the 'buyPrice' column using VLOOKUP with absolute references

wsMaster.Range("H2:H" & lastRowMaster).formula = \_

"=VLOOKUP(B2,Products!A:H,8)"

' Populate the 'Total Revenue' column

wsMaster.Range("I2:I" & lastRowMaster).formula = \_

"=C2\*D2"

' Populate the 'Total Profit' column

wsMaster.Range("J2:J" & lastRowMaster).formula = \_

"=(D2-H2)\*C2"

' Populate the 'Year' column

wsMaster.Range("K2:K" & lastRowMaster).formula = \_

"=year(F2)"

' Populate the 'Month' column

wsMaster.Range("L2:L" & lastRowMaster).formula = \_

"=month(F2)"

' Populate the 'officeCity' column

wsMaster.Range("M2:M" & lastRowMaster).formula = \_

"=VLOOKUP(A2,'Master Sheet'!A:O,15)"

' Format the 'Order Date' column as date

wsMaster.Columns("F:F").NumberFormat = "dd/mm/yyyy;@"

' Autofit columns for better readability

wsMaster.Columns("A:G").AutoFit

Application.Calculation = xlCalculationAutomatic

Application.ScreenUpdating = True

Application.EnableEvents = True

Application.DisplayStatusBar = True

End Sub

Sub Macro\_04A\_FilterStatuses()

Dim ws As Worksheet

Dim lastRow As Long

Dim keepStatuses As Variant

Dim i As Long

Dim statusColumn As String

' Set the worksheet

Set ws = ThisWorkbook.Sheets("Products & Profit Master Sheet")

' Determine the last row in the dataset

lastRow = ws.Cells(ws.Rows.Count, "A").End(xlUp).Row

' Define the statuses to keep

keepStatuses = Array("Shipped", "Resolved")

' Define the column containing the statuses (adjust if not column G)

statusColumn = "G"

' Loop through rows from bottom to top to avoid skipping rows during deletion

For i = lastRow To 2 Step -1

' Check if the status in the row is NOT in the list of statuses to keep

If IsError(Application.Match(ws.Cells(i, statusColumn).Value, keepStatuses, 0)) Then

' If the status is not in the keepStatuses array, delete the row

ws.Rows(i).Delete

End If

Next i

End Sub

Sub Macro\_05\_SLAMacro1\_PopulateSLACompliance()

On Error GoTo ErrorHandler

Application.Calculation = xlCalculationManual

Application.ScreenUpdating = False

Application.EnableEvents = False

Application.DisplayStatusBar = False

' Define office cities

Dim officeCities As Variant

officeCities = Array("Boston", "San Francisco", "NYC", "Paris", "Tokyo", "Sydney", "London")

' Define months

Dim months As Variant

months = Array("January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December")

' Define worksheets

Dim wsCompliance As Worksheet, wsMaster As Worksheet

Set wsCompliance = ThisWorkbook.Sheets("SLA\_Compliance")

Set wsMaster = ThisWorkbook.Sheets("Master Sheet")

' Clear all data and formatting on the SLA\_Compliance sheet

wsCompliance.Cells.Clear

' Format columns G, M, and S as percentage

wsCompliance.Columns("G").NumberFormat = "0%"

wsCompliance.Columns("M").NumberFormat = "0%"

wsCompliance.Columns("S").NumberFormat = "0%"

' Align columns G, M, and S to the right

wsCompliance.Columns("G").HorizontalAlignment = xlRight

wsCompliance.Columns("M").HorizontalAlignment = xlRight

wsCompliance.Columns("S").HorizontalAlignment = xlRight

' Define variables

Dim currentRow As Long, dataStartRow As Long

Dim lastRow As Long, orderYear As Long

Dim uniqueYears As Object, yearKey As Variant

Dim city As Variant, monthName As Variant

' Initialize

currentRow = 9 ' Start populating from row 9

dataStartRow = 9 ' Data starts from row 9

Set uniqueYears = CreateObject("Scripting.Dictionary")

' Add section heading at row 7

wsCompliance.Cells(7, 1).Value = "Deliveries per year"

wsCompliance.Cells(7, 1).Font.Bold = True ' Bold the section heading

' Add column headings for the main section

wsCompliance.Cells(8, 1).Value = "Office City"

wsCompliance.Cells(8, 2).Value = "Year"

wsCompliance.Cells(8, 3).Value = "Month"

wsCompliance.Cells(8, 4).Value = "Total Deliveries"

wsCompliance.Cells(8, 5).Value = "On-time Deliveries"

wsCompliance.Cells(8, 6).Value = "Late Deliveries"

wsCompliance.Cells(8, 7).Value = "SLA Compliance Percentage"

' Add column headings for additional sections

wsCompliance.Cells(8, 9).Value = "Office City"

wsCompliance.Cells(8, 10).Value = "Month"

wsCompliance.Cells(8, 11).Value = "Total Deliveries"

wsCompliance.Cells(8, 12).Value = "On-time Deliveries"

wsCompliance.Cells(8, 13).Value = "SLA Compliance Percentage"

wsCompliance.Cells(8, 15).Value = "Office City"

wsCompliance.Cells(8, 16).Value = "Month"

wsCompliance.Cells(8, 17).Value = "Total Deliveries"

wsCompliance.Cells(8, 18).Value = "On-time Deliveries"

wsCompliance.Cells(8, 19).Value = "SLA Compliance Percentage"

' Add section titles above column headings

wsCompliance.Cells(7, 9).Value = "Average % of On-Time Deliveries (Historical Data)"

wsCompliance.Cells(7, 15).Value = "Average % of On-Time Deliveries (Most Recent Data)"

' Bold all headings

wsCompliance.Rows(7 & ":" & 8).Font.Bold = True

' Move to data rows

currentRow = 9

' Get unique years from Master Sheet

lastRow = wsMaster.Cells(wsMaster.Rows.Count, "F").End(xlUp).Row ' Order date column (F)

Dim i As Long

For i = 2 To lastRow ' Start at row 2 to skip headers

If IsDate(wsMaster.Cells(i, "F").Value) Then

orderYear = year(wsMaster.Cells(i, "F").Value)

If Not uniqueYears.exists(orderYear) Then

uniqueYears.Add orderYear, True

End If

End If

Next i

' Populate data for main section starting from row 9

For Each city In officeCities

For Each yearKey In uniqueYears.Keys

For Each monthName In months

wsCompliance.Cells(currentRow, 1).Value = city ' Office City

wsCompliance.Cells(currentRow, 2).Value = yearKey ' Year

wsCompliance.Cells(currentRow, 3).Value = monthName ' Month

currentRow = currentRow + 1

Next monthName

Next yearKey

Next city

' Populate data for Historical Data (columns I and J)

Dim officeRow As Long

officeRow = dataStartRow ' Adjusted to avoid gaps

For Each city In officeCities

For Each monthName In months

wsCompliance.Cells(officeRow, 9).Value = city ' Office City in column I

wsCompliance.Cells(officeRow, 10).Value = monthName ' Month in column J

officeRow = officeRow + 1

Next monthName

Next city

' Populate data for Most Recent Data (columns O and P)

Dim monthRow As Long

monthRow = dataStartRow ' Adjusted to avoid gaps

For Each city In officeCities

For Each monthName In months

wsCompliance.Cells(monthRow, 15).Value = city ' Office City in column O

wsCompliance.Cells(monthRow, 16).Value = monthName ' Month in column P

monthRow = monthRow + 1

Next monthName

Next city

' Autofit all columns

wsCompliance.Columns("A:S").AutoFit

Exit Sub

ErrorHandler:

MsgBox "An error occurred: " & Err.Description, vbCritical

Application.Calculation = xlCalculationAutomatic

Application.ScreenUpdating = True

Application.EnableEvents = True

Application.DisplayStatusBar = True

End Sub

Sub Macro\_06\_SLAMacro2\_IdentifyYearsBasedOnMonths()

On Error GoTo ErrorHandler

Application.Calculation = xlCalculationManual

Application.ScreenUpdating = False

Application.EnableEvents = False

Application.DisplayStatusBar = False

' Define worksheets

Dim wsMaster As Worksheet, wsCompliance As Worksheet

Set wsMaster = ThisWorkbook.Sheets("Master Sheet")

Set wsCompliance = ThisWorkbook.Sheets("SLA\_Compliance")

' Define variables

Dim lastRow As Long

Dim yearMonthsDict As Object

Dim firstFullYear As Variant, secondFullYear As Variant, incompleteYear As Variant

Dim year As Long

Dim month As String

Dim cell As Range

Dim monthsRequired As Variant

Dim currentYearValue As Variant

' Initialize dictionary and variables

Set yearMonthsDict = CreateObject("Scripting.Dictionary")

monthsRequired = Array("January", "February", "March", "April", "May", "June", \_

"July", "August", "September", "October", "November", "December")

' Find the last row in column F (Master Sheet)

lastRow = wsMaster.Cells(wsMaster.Rows.Count, "F").End(xlUp).Row

' Loop through column F to gather year and month data

For Each cell In wsMaster.Range("F2:F" & lastRow)

If IsDate(cell.Value) Then

year = VBA.year(cell.Value)

month = VBA.Format(cell.Value, "mmmm") ' Get full month name

If Not yearMonthsDict.exists(year) Then

yearMonthsDict.Add year, CreateObject("Scripting.Dictionary")

End If

If Not yearMonthsDict(year).exists(month) Then

yearMonthsDict(year).Add month, True

End If

End If

Next cell

' Determine full years and incomplete years

For Each currentYearValue In yearMonthsDict.Keys

If yearMonthsDict(currentYearValue).Count = 12 Then

If IsEmpty(firstFullYear) Then

firstFullYear = currentYearValue

ElseIf IsEmpty(secondFullYear) Then

secondFullYear = currentYearValue

End If

Else

incompleteYear = currentYearValue ' The year with less than 12 months

End If

Next currentYearValue

' Output results to SLA\_Compliance sheet

wsCompliance.Cells(1, 1).Value = "Historical Year 1:"

wsCompliance.Cells(1, 2).Value = firstFullYear

wsCompliance.Cells(2, 1).Value = "Historical Year 2:"

wsCompliance.Cells(2, 2).Value = secondFullYear

wsCompliance.Cells(1, 4).Value = "Current Year:"

wsCompliance.Cells(1, 5).Value = incompleteYear

Exit Sub

ErrorHandler:

MsgBox "An error occurred: " & Err.Description, vbCritical

Application.Calculation = xlCalculationAutomatic

Application.ScreenUpdating = True

Application.EnableEvents = True

Application.DisplayStatusBar = True

End Sub

Sub Macro\_07\_SLAMacro3\_PopulateFormulasinSLACompliance()

On Error GoTo ErrorHandler

Application.Calculation = xlCalculationManual

Application.ScreenUpdating = False

Application.EnableEvents = False

Application.DisplayStatusBar = False

Dim wsCompliance As Worksheet

Dim lastRowMain As Long, lastRowHistorical As Long, lastRowRecent As Long

Dim nonBoldRange As Range

' Set the worksheet

Set wsCompliance = ThisWorkbook.Sheets("SLA\_Compliance")

' Define the last rows dynamically for each section

lastRowMain = wsCompliance.Cells(wsCompliance.Rows.Count, "A").End(xlUp).Row ' Last row for main data

lastRowHistorical = wsCompliance.Cells(wsCompliance.Rows.Count, "I").End(xlUp).Row ' Last row for historical data

lastRowRecent = wsCompliance.Cells(wsCompliance.Rows.Count, "O").End(xlUp).Row ' Last row for recent data

' Insert formulas dynamically for the main section

With wsCompliance

' D9: Total Deliveries

.Range("D9:D" & lastRowMain).formula = "=E9+F9"

' E9: On-time Deliveries

.Range("E9:E" & lastRowMain).formula = "=IFERROR(" & \_

"COUNTIFS('Master Sheet'!$V:$V, ""Yes""," & \_

"'Master Sheet'!$W:$W, B9," & \_

"'Master Sheet'!$X:$X, MATCH(C9, {""January"",""February"",""March"",""April"",""May"",""June"",""July"",""August"",""September"",""October"",""November"",""December""}, 0)," & \_

"'Master Sheet'!$O:$O, A9)," & \_

"0)"

' F9: Late Deliveries

.Range("F9:F" & lastRowMain).formula = "=IFERROR(" & \_

"COUNTIFS('Master Sheet'!$V:$V, ""No""," & \_

"'Master Sheet'!$W:$W, B9," & \_

"'Master Sheet'!$X:$X, MATCH(C9, {""January"",""February"",""March"",""April"",""May"",""June"",""July"",""August"",""September"",""October"",""November"",""December""}, 0)," & \_

"'Master Sheet'!$O:$O, A9)," & \_

"0)"

' G9: SLA Compliance Percentage

.Range("G9:G" & lastRowMain).formula = "=IFERROR(E9/D9,""N/A"")"

' Columns K, L, M for Historical Data

.Range("K9:K" & lastRowHistorical).formula = "=SUMIFS(D:D, B:B, $B$1, A:A, I9, C:C, J9) + SUMIFS(D:D, B:B, $B$2, A:A, I9, C:C, J9)"

.Range("L9:L" & lastRowHistorical).formula = "=SUMIFS(E:E, B:B, $B$1, A:A, I9, C:C, J9) + SUMIFS(E:E, B:B, $B$2, A:A, I9, C:C, J9)"

.Range("M9:M" & lastRowHistorical).formula = "=IFERROR(L9/K9,""NA"")"

' Columns Q, R, S for Most Recent Data

.Range("Q9:Q" & lastRowRecent).formula = "=SUMIFS(D:D, B:B, $E$1, A:A, O9, C:C, P9)"

.Range("R9:R" & lastRowRecent).formula = "=SUMIFS(E:E, B:B, $E$1, A:A, O9, C:C, P9)"

.Range("S9:S" & lastRowRecent).formula = "=IFERROR(R9/Q9,""NA"")"

' Ensure R column onwards (R9 downwards) is NOT bold

Set nonBoldRange = .Range("R9:S" & lastRowRecent)

nonBoldRange.Font.Bold = False

End With

' Exit the subroutine successfully

Application.Calculation = xlCalculationAutomatic

Application.ScreenUpdating = True

Application.EnableEvents = True

Application.DisplayStatusBar = True

Exit Sub

ErrorHandler:

MsgBox "An error occurred: " & Err.Description, vbCritical

End Sub

Sub Macro\_08\_SLAMacro4\_SetComplianceAnalysisStructure()

'

' SLA\_Compliance\_Analysis\_Structure Macro

'

Dim ws As Worksheet

Dim obj As Object

Application.Calculation = xlCalculationManual

Application.ScreenUpdating = False

Application.EnableEvents = False

Application.DisplayStatusBar = False

' Set worksheet

Set ws = ThisWorkbook.Sheets("SLA\_Compliance\_Analysis") ' Ensure you're on the correct sheet

' Remove all charts and embedded objects

For Each obj In ws.Shapes

obj.Delete

Next obj

' Clear the sheet for a clean start (cell contents)

ws.Range("A1:XFD80").ClearContents '

' 1. Overall Historical SLA Compliance by Office

ws.Range("A1").Value = "Overall Historical SLA Compliance by Office"

ws.Range("A2").Value = "Office"

ws.Range("B2").Value = "Total Deliveries"

ws.Range("C2").Value = "On-time Deliveries"

ws.Range("D2").Value = "Historical data - SLA Compliance (%)"

ws.Range("E2").Value = "Historical Contribution to Total On-Time Deliveries (%)"

ws.Range("A3:A9").Value = Application.Transpose(Array("Boston", "San Francisco", "NYC", "Paris", "Tokyo", "Sydney", "London"))

' Merge and center A1:C1

With ws.Range("A1:C1")

.Merge

.HorizontalAlignment = xlLeft

.VerticalAlignment = xlCenter

.Font.Size = 14 ' Set the font size to 14

End With

' Formatting for Section 1

ws.Range("A1:E2").Font.Bold = True

ws.Range("A2:E9").Borders.LineStyle = xlContinuous

' 2. Recent SLA Compliance by Office

ws.Range("A12").Value = "Recent SLA Compliance by Office"

ws.Range("A13").Value = "Office"

ws.Range("B13").Value = "Total Deliveries"

ws.Range("C13").Value = "On-time Deliveries"

ws.Range("D13").Value = "Recent data - SLA Compliance (%)"

ws.Range("E13").Value = "Recent Contribution to Total On-Time Deliveries (%)"

ws.Range("A14:A20").Value = Application.Transpose(Array("Boston", "San Francisco", "NYC", "Paris", "Tokyo", "Sydney", "London"))

' Merge and center A12:C12

With ws.Range("A12:C12")

.Merge

.HorizontalAlignment = xlLeft

.VerticalAlignment = xlCenter

.Font.Size = 14 ' Set the font size to 14

End With

' Formatting for Section 2

ws.Range("A12:E13").Font.Bold = True

ws.Range("A13:E20").Borders.LineStyle = xlContinuous

' 3. Recent SLA Compliance by Office with Deviations (Moved Up)

ws.Range("A23").Value = "SLA Compliance Comparison by Office - Summary"

ws.Range("A24").Value = "Office"

ws.Range("B24").Value = "Historical data - SLA Compliance (%)"

ws.Range("C24").Value = "Most recent data - SLA Compliance (%)"

ws.Range("D24").Value = "Absolute Percentage Deviation"

ws.Range("E24").Value = "Historical Contribution to Total On-Time Deliveries (%)"

ws.Range("F24").Value = "Recent Contribution to Total On-Time Deliveries (%)"

ws.Range("G24").Value = "Absolute Percentage Deviation in Contribution"

ws.Range("A25:A31").Value = Application.Transpose(Array("Boston", "San Francisco", "NYC", "Paris", "Tokyo", "Sydney", "London"))

' Merge and center A23:C23

With ws.Range("A23:C23")

.Merge

.HorizontalAlignment = xlLeft

.VerticalAlignment = xlCenter

.Font.Size = 14 ' Set the font size to 14

End With

' Formatting for Section 3

ws.Range("A23:G24").Font.Bold = True

ws.Range("A24:G31").Borders.LineStyle = xlContinuous

' 4. Historical SLA Performance Matrix by Office and by Month

ws.Range("A34").Value = "Historical SLA Performance Matrix - by office and by month"

ws.Range("A35").Value = "Month"

ws.Range("B35:H35").Value = Array("Boston", "San Francisco", "NYC", "Paris", "Tokyo", "Sydney", "London")

ws.Range("A36:A47").Value = Application.Transpose(Array("January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December"))

' Merge and center A34:C34

With ws.Range("A34:C34")

.Merge

.HorizontalAlignment = xlLeft

.VerticalAlignment = xlCenter

.Font.Size = 14 ' Set the font size to 14

End With

' Formatting for Section 4

ws.Range("A34:H35").Font.Bold = True

ws.Range("A35:H47").Borders.LineStyle = xlContinuous

' 5. Recent SLA Compliance by Office and by Month

ws.Range("A50").Value = "Recent SLA Compliance Matrix - most recent data by office and by month"

ws.Range("A51").Value = "Month"

ws.Range("B51:H51").Value = Array("Boston", "San Francisco", "NYC", "Paris", "Tokyo", "Sydney", "London")

ws.Range("A52:A63").Value = Application.Transpose(Array("January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December"))

' Merge and center A50:D50

With ws.Range("A50:D50")

.Merge

.HorizontalAlignment = xlLeft

.VerticalAlignment = xlCenter

.Font.Size = 14 ' Set the font size to 14

End With

' Formatting for Section 5

ws.Range("A50:H51").Font.Bold = True

ws.Range("A51:H63").Borders.LineStyle = xlContinuous

' 6. Monthly Error Analysis (Matrix by Month and Office)

ws.Range("A66").Value = "Monthly Error Analysis (Matrix by Month and Office)"

ws.Range("A67").Value = "Month"

ws.Range("B67:H67").Value = Array("Boston", "San Francisco", "NYC", "Paris", "Tokyo", "Sydney", "London")

ws.Range("A68:A79").Value = Application.Transpose(Array("January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December"))

' Merge and center A66:C66

With ws.Range("A66:C66")

.Merge

.HorizontalAlignment = xlLeft

.VerticalAlignment = xlCenter

.Font.Size = 14 ' Set the font size to 14

End With

' Formatting for Section 6

ws.Range("A66:H67").Font.Bold = True

ws.Range("A67:H79").Borders.LineStyle = xlContinuous

' Set all columns to a maximum width of 26.71

Dim col As Range

For Each col In ws.Columns("A:H")

col.ColumnWidth = 26.71

Next col

' Wrap text in all cells

ws.Cells.WrapText = True

Application.Calculation = xlCalculationAutomatic

Application.ScreenUpdating = True

Application.EnableEvents = True

Application.DisplayStatusBar = True

End Sub

Sub Macro\_09\_SLAMacro5\_PopulateSLAComplianceAnalysis()

Application.Calculation = xlCalculationManual

Application.ScreenUpdating = False

Application.EnableEvents = False

Application.DisplayStatusBar = False

Dim ws As Worksheet

Set ws = ThisWorkbook.Sheets("SLA\_Compliance\_Analysis") ' Replace with your sheet name if different

' 1. Populate "Overall Historical SLA Compliance by Office"

ws.Range("B3:B9").formula = "=SUMIF(SLA\_Compliance!I:I,A3,SLA\_Compliance!K:K)" ' Total Deliveries

ws.Range("C3:C9").formula = "=SUMIF(SLA\_Compliance!I:I,A3,SLA\_Compliance!L:L)" ' On-time Deliveries

ws.Range("D3:D9").formula = "=C3/B3" ' SLA Compliance %

ws.Range("E3:E9").formula = "=C3/SUM($C$3:$C$9)" ' Historical Contribution to Total On-Time Deliveries (%)

ws.Range("D3:E9").NumberFormat = "0%" ' Format as percentage with no decimal places

' 2. Populate "Recent SLA Compliance by Office"

ws.Range("B14:B20").formula = "=SUMIF(SLA\_Compliance!O:O,A14,SLA\_Compliance!Q:Q)" ' Total Deliveries

ws.Range("C14:C20").formula = "=SUMIF(SLA\_Compliance!O:O,A14,SLA\_Compliance!R:R)" ' On-time Deliveries

ws.Range("D14:D20").formula = "=C14/B14" ' SLA Compliance %

ws.Range("E14:E20").formula = "=C14/SUM($C$14:$C$20)" ' Recent Contribution to Total On-Time Deliveries (%)

ws.Range("D14:E20").NumberFormat = "0%" ' Format as percentage with no decimal places

' 3. Populate "SLA Compliance Comparison"

ws.Range("B25:B31").formula = "=D3" ' Historical SLA Compliance (%)

ws.Range("C25:C31").formula = "=D14" ' Most recent SLA Compliance (%)

ws.Range("D25:D31").formula = "=ABS(B25-C25)" ' Absolute Percentage Deviation

ws.Range("E25:E31").formula = "=E3" ' Historical Contribution to Total On-Time Deliveries (%)

ws.Range("F25:F31").formula = "=E14" ' Recent Contribution to Total On-Time Deliveries (%)

ws.Range("G25:G31").formula = "=ABS(E25-F25)" ' Absolute Percentage Deviation in Contribution

ws.Range("B25:G31").NumberFormat = "0%" ' Format as percentage with no decimal places

' 4. Populate "Historical SLA Performance Matrix by Office and by Month"

ws.Range("B36").FormulaArray = \_

"=INDEX(SLA\_Compliance!$M:$M,MATCH(1,(SLA\_Compliance!$I:$I=$B$35)\*(SLA\_Compliance!$J:$J=A36),0))"

ws.Range("B36:B47").FillDown ' Dynamically fill the range

ws.Range("C36").FormulaArray = \_

"=INDEX(SLA\_Compliance!$M:$M,MATCH(1,(SLA\_Compliance!$I:$I=$C$35)\*(SLA\_Compliance!$J:$J=A36),0))"

ws.Range("C36:C47").FillDown ' Dynamically fill the range

ws.Range("D36").FormulaArray = \_

"=INDEX(SLA\_Compliance!$M:$M,MATCH(1,(SLA\_Compliance!$I:$I=$D$35)\*(SLA\_Compliance!$J:$J=A36),0))"

ws.Range("D36:D47").FillDown ' Dynamically fill the range

ws.Range("E36").FormulaArray = \_

"=INDEX(SLA\_Compliance!$M:$M,MATCH(1,(SLA\_Compliance!$I:$I=$E$35)\*(SLA\_Compliance!$J:$J=A36),0))"

ws.Range("E36:E47").FillDown ' Dynamically fill the range

ws.Range("F36").FormulaArray = \_

"=INDEX(SLA\_Compliance!$M:$M,MATCH(1,(SLA\_Compliance!$I:$I=$F$35)\*(SLA\_Compliance!$J:$J=A36),0))"

ws.Range("F36:F47").FillDown ' Dynamically fill the range

ws.Range("G36").FormulaArray = \_

"=INDEX(SLA\_Compliance!$M:$M,MATCH(1,(SLA\_Compliance!$I:$I=$G$35)\*(SLA\_Compliance!$J:$J=A36),0))"

ws.Range("G36:G47").FillDown ' Dynamically fill the range

ws.Range("H36").FormulaArray = \_

"=INDEX(SLA\_Compliance!$M:$M,MATCH(1,(SLA\_Compliance!$I:$I=$H$35)\*(SLA\_Compliance!$J:$J=A36),0))"

ws.Range("H36:H47").FillDown ' Dynamically fill the range

' Format B36:H47 as percentage with no decimal places and align to the right

With ws.Range("B36:H47")

.NumberFormat = "0%" ' Set format to percentage with no decimals

.HorizontalAlignment = xlRight ' Align to the right

End With

' 5. Populate "Recent SLA Compliance by Office - most recent data by office and by month"

ws.Range("B52").FormulaArray = \_

"=INDEX(SLA\_Compliance!S:S,MATCH(1,(SLA\_Compliance!O:O=$B$51)\*(SLA\_Compliance!P:P=A52),0))"

ws.Range("B52:B63").FillDown ' Dynamically fill the range

ws.Range("C52").FormulaArray = \_

"=INDEX(SLA\_Compliance!S:S,MATCH(1,(SLA\_Compliance!O:O=$C$51)\*(SLA\_Compliance!P:P=A52),0))"

ws.Range("C52:C63").FillDown ' Dynamically fill the range

ws.Range("D52").FormulaArray = \_

"=INDEX(SLA\_Compliance!S:S,MATCH(1,(SLA\_Compliance!O:O=$D$51)\*(SLA\_Compliance!P:P=A52),0))"

ws.Range("D52:D63").FillDown ' Dynamically fill the range

ws.Range("E52").FormulaArray = \_

"=INDEX(SLA\_Compliance!S:S,MATCH(1,(SLA\_Compliance!O:O=$E$51)\*(SLA\_Compliance!P:P=A52),0))"

ws.Range("E52:E63").FillDown ' Dynamically fill the range

ws.Range("F52").FormulaArray = \_

"=INDEX(SLA\_Compliance!S:S,MATCH(1,(SLA\_Compliance!O:O=$F$51)\*(SLA\_Compliance!P:P=A52),0))"

ws.Range("F52:F63").FillDown ' Dynamically fill the range

ws.Range("G52").FormulaArray = \_

"=INDEX(SLA\_Compliance!S:S,MATCH(1,(SLA\_Compliance!O:O=$G$51)\*(SLA\_Compliance!P:P=A52),0))"

ws.Range("G52:G63").FillDown ' Dynamically fill the range

ws.Range("H52").FormulaArray = \_

"=INDEX(SLA\_Compliance!S:S,MATCH(1,(SLA\_Compliance!O:O=$H$51)\*(SLA\_Compliance!P:P=A52),0))"

ws.Range("H52:H63").FillDown ' Dynamically fill the range

' Format B52:H63 as percentage with no decimal places and align to the right

With ws.Range("B52:H63")

.NumberFormat = "0%" ' Set format to percentage with no decimals

.HorizontalAlignment = xlRight ' Align to the right

End With

' 6. Populate "Monthly Error Analysis (Matrix by Month and Office)"

ws.Range("B68").FormulaArray = \_

"=IFERROR(ABS(B36-B52),""NA"")"

ws.Range("B68:B79").FillDown ' Dynamically fill the range

ws.Range("C68").FormulaArray = \_

"=IFERROR(ABS(C36-C52),""NA"")"

ws.Range("C68:C79").FillDown ' Dynamically fill the range

ws.Range("D68").FormulaArray = \_

"=IFERROR(ABS(D36-D52),""NA"")"

ws.Range("D68:D79").FillDown ' Dynamically fill the range

ws.Range("E68").FormulaArray = \_

"=IFERROR(ABS(E36-E52),""NA"")"

ws.Range("E68:E79").FillDown ' Dynamically fill the range

ws.Range("F68").FormulaArray = \_

"=IFERROR(ABS(F36-F52),""NA"")"

ws.Range("F68:F79").FillDown ' Dynamically fill the range

ws.Range("G68").FormulaArray = \_

"=IFERROR(ABS(G36-G52),""NA"")"

ws.Range("G68:G79").FillDown ' Dynamically fill the range

ws.Range("H68").FormulaArray = \_

"=IFERROR(ABS(H36-H52),""NA"")"

ws.Range("H68:H79").FillDown ' Dynamically fill the range

' Format B68:H79 as percentage with no decimal places and align to the right

With ws.Range("B68:H79")

.NumberFormat = "0%" ' Set format to percentage with no decimals

.HorizontalAlignment = xlRight ' Align to the right

End With

' Apply Conditional Formatting to SLA Compliance Columns

' Conditional formatting for the first range

Dim rng1 As Range

Set rng1 = ws.Range("D3:D9, D14:D20, B36:H47, B52:H63, D68:D79, B25:C31, E3:E9, E14:E20, E25:E31, F25:F31")

With rng1

.FormatConditions.AddDatabar

With .FormatConditions(.FormatConditions.Count)

.ShowValue = True

.SetFirstPriority

.MinPoint.Modify newtype:=xlConditionValueNumber, newvalue:=0

.MaxPoint.Modify newtype:=xlConditionValueNumber, newvalue:=1

.BarColor.Color = RGB(0, 176, 80)

End With

End With

' Conditional formatting for the second range

Dim rng2 As Range

Set rng2 = ws.Range("D25:D31, B68:H79, G25:G31")

With rng2

.FormatConditions.AddDatabar

With .FormatConditions(.FormatConditions.Count)

.ShowValue = True

.SetFirstPriority

.MinPoint.Modify newtype:=xlConditionValueNumber, newvalue:=0

.MaxPoint.Modify newtype:=xlConditionValueNumber, newvalue:=1

.BarColor.Color = RGB(255, 0, 0)

End With

End With

Application.Calculation = xlCalculationAutomatic

Application.ScreenUpdating = True

Application.EnableEvents = True

Application.DisplayStatusBar = True

End Sub

Sub Macro\_10\_SLAMacro6\_CreateSLAVisuals()

Application.Calculation = xlCalculationManual

Application.ScreenUpdating = False

Application.EnableEvents = False

Application.DisplayStatusBar = False

Dim ws As Worksheet

Dim chart1 As ChartObject, chart2 As ChartObject

Dim chartRange1 As Range, chartRange2 As Range

' Set worksheet reference

Set ws = ThisWorkbook.Sheets("SLA\_Compliance\_Analysis")

' Validate the chart ranges

Set chartRange1 = ws.Range("A2:D9")

Set chartRange2 = ws.Range("A24:D31")

If WorksheetFunction.CountA(chartRange1) = 0 Or WorksheetFunction.CountA(chartRange2) = 0 Then

MsgBox "The data range for one or both charts is empty or invalid.", vbExclamation, "Chart Creation Error"

Exit Sub

End If

' Remove existing chart objects

Dim chartObj As ChartObject

For Each chartObj In ws.ChartObjects

chartObj.Delete

Next chartObj

' Create the first chart

Set chart1 = ws.ChartObjects.Add(Left:=ws.Columns("L").Left + 10, Width:=500, Top:=50, Height:=250)

With chart1.Chart

.ChartType = xlColumnClustered

.SetSourceData Source:=chartRange1

.HasTitle = True

.ChartTitle.Text = "Historical On-Time Delivery Rates by Office"

.ChartTitle.Format.TextFrame2.TextRange.Font.Size = 16

.SeriesCollection(3).AxisGroup = xlSecondary

.Axes(xlValue, xlPrimary).HasTitle = True

.Axes(xlValue, xlPrimary).AxisTitle.Text = "Deliveries"

.Axes(xlValue, xlPrimary).AxisTitle.Format.TextFrame2.TextRange.Font.Size = 10.5

.Axes(xlValue, xlSecondary).HasTitle = True

.Axes(xlValue, xlSecondary).AxisTitle.Text = "Percentage SLA Compliance"

.Axes(xlValue, xlSecondary).AxisTitle.Format.TextFrame2.TextRange.Font.Size = 10.5

With .SeriesCollection(3)

.ChartType = xlLine

.Format.Line.ForeColor.RGB = RGB(200, 200, 200)

.Format.Line.Weight = 2

End With

.ChartStyle = 328

End With

' Create the second chart

Set chart2 = ws.ChartObjects.Add(Left:=ws.Columns("L").Left + 10, Width:=500, Top:=350, Height:=250)

With chart2.Chart

.ChartType = xlColumnClustered

.SetSourceData Source:=chartRange2

.HasTitle = True

.ChartTitle.Text = "Historical vs Recent SLA Compliance"

.ChartTitle.Format.TextFrame2.TextRange.Font.Size = 16

.SeriesCollection(3).AxisGroup = xlSecondary

.Axes(xlValue, xlPrimary).HasTitle = True

.Axes(xlValue, xlPrimary).AxisTitle.Text = "SLA Compliance (%)"

.Axes(xlValue, xlSecondary).HasTitle = True

.Axes(xlValue, xlSecondary).AxisTitle.Text = "Absolute Percentage Deviation"

With .SeriesCollection(3)

.ChartType = xlLine

.Format.Line.ForeColor.RGB = RGB(200, 200, 200)

.Format.Line.Weight = 2

End With

.ChartStyle = 328

End With

' Restore application settings

Application.Calculation = xlCalculationAutomatic

Application.ScreenUpdating = True

Application.EnableEvents = True

Application.DisplayStatusBar = True

End Sub

Sub Macro\_11\_PaymentsAnalysisMacro1\_CreatePaymentsAnalysisStructure()

Application.Calculation = xlCalculationManual

Application.ScreenUpdating = False

Application.EnableEvents = False

Application.DisplayStatusBar = False

' Declare variables

Dim ws As Worksheet

Dim col As Range

' Check if "Payments Analysis" sheet exists, and create it if not

On Error Resume Next

Set ws = ThisWorkbook.Sheets("Payments Analysis")

On Error GoTo 0

If ws Is Nothing Then

Set ws = ThisWorkbook.Sheets.Add(After:=ThisWorkbook.Sheets(ThisWorkbook.Sheets.Count))

ws.Name = "Payments Analysis"

Else

ws.Range("A1:XFD52").ClearContents '

' Clear the sheet if it exists

End If

' Set values for A1, A2, and A3 in the "Payments Analysis" tab

ws.Range("A1").Value = "Year 1"

ws.Range("A2").Value = "Year 2"

ws.Range("A3").Value = "Current Year"

' Copy the values from the "SLA\_Compliance" sheet to the "Payments Analysis" sheet

On Error Resume Next

If Not ThisWorkbook.Sheets("SLA\_Compliance") Is Nothing Then

' Copy value from B1 in "SLA\_Compliance" to B1 in "Payments Analysis"

ws.Range("B1").Value = ThisWorkbook.Sheets("SLA\_Compliance").Range("B1").Value

' Copy value from B2 in "SLA\_Compliance" to B2 in "Payments Analysis"

ws.Range("B2").Value = ThisWorkbook.Sheets("SLA\_Compliance").Range("B2").Value

' Copy value from E1 in "SLA\_Compliance" to B3 in "Payments Analysis"

ws.Range("B3").Value = ThisWorkbook.Sheets("SLA\_Compliance").Range("E1").Value

Else

MsgBox "Sheet 'SLA\_Compliance' not found. B1, B2, and B3 in 'Payments Analysis' not updated.", vbExclamation

End If

On Error GoTo 0

' Create "Monthly Payment Activity - Year 1 Matrix" Section

ws.Range("A6").Value = "Monthly Payment Activity - Year 1 - by office and by Month"

ws.Range("A7").Value = "Month"

ws.Range("B7:H7").Value = Array("Boston", "San Francisco", "NYC", "Paris", "Tokyo", "Sydney", "London")

ws.Range("A8:A19").Value = Application.Transpose(Array("January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December"))

' Merge and format the section header

With ws.Range("A6:E6")

.Merge

.HorizontalAlignment = xlLeft

.VerticalAlignment = xlCenter

.Font.Size = 14

End With

' Apply formatting

ws.Range("A6:H7").Font.Bold = True

ws.Range("A7:H19").Borders.LineStyle = xlContinuous

' Create "Monthly Payment Activity - Year 2 Matrix" Section

ws.Range("A22").Value = "Monthly Payment Activity - Year 2 - by office and by Month"

ws.Range("A23").Value = "Month"

ws.Range("B23:H23").Value = Array("Boston", "San Francisco", "NYC", "Paris", "Tokyo", "Sydney", "London")

ws.Range("A24:A35").Value = Application.Transpose(Array("January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December"))

' Merge and format the section header

With ws.Range("A22:E22")

.Merge

.HorizontalAlignment = xlLeft

.VerticalAlignment = xlCenter

.Font.Size = 14

End With

' Apply formatting

ws.Range("A22:H23").Font.Bold = True

ws.Range("A23:H35").Borders.LineStyle = xlContinuous

' Create "Monthly Payment Activity - Current Year Matrix" Section

ws.Range("A38").Value = "Current Year Payments Performance Matrix by Office and by Month"

ws.Range("A39").Value = "Month"

ws.Range("B39:H39").Value = Array("Boston", "San Francisco", "NYC", "Paris", "Tokyo", "Sydney", "London")

ws.Range("A40:A51").Value = Application.Transpose(Array("January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December"))

' Merge and format the section header

With ws.Range("A38:E38")

.Merge

.HorizontalAlignment = xlLeft

.VerticalAlignment = xlCenter

.Font.Size = 14

End With

' Apply formatting

ws.Range("A38:H39").Font.Bold = True

ws.Range("A39:H51").Borders.LineStyle = xlContinuous

' Create "Year 1 - Monthly Payment Value" Section

ws.Range("J6").Value = "Year 1 Monthly Payment Value - by office and by Month"

ws.Range("J7").Value = "Month"

ws.Range("K7:Q7").Value = Array("Boston", "San Francisco", "NYC", "Paris", "Tokyo", "Sydney", "London")

ws.Range("J8:J19").Value = Application.Transpose(Array("January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December"))

' Merge and format the section header

With ws.Range("J6:N6")

.Merge

.HorizontalAlignment = xlLeft

.VerticalAlignment = xlCenter

.Font.Size = 14

End With

' Apply formatting

ws.Range("J6:Q7").Font.Bold = True

ws.Range("J7:Q19").Borders.LineStyle = xlContinuous

' Create "Year 2 - Monthly Payment Value" Section

ws.Range("J22").Value = "Year 2 Payment Value - by office and by Month"

ws.Range("J23").Value = "Month"

ws.Range("K23:Q23").Value = Array("Boston", "San Francisco", "NYC", "Paris", "Tokyo", "Sydney", "London")

ws.Range("J24:J35").Value = Application.Transpose(Array("January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December"))

' Merge and format the section header

With ws.Range("J22:N22")

.Merge

.HorizontalAlignment = xlLeft

.VerticalAlignment = xlCenter

.Font.Size = 14

End With

' Apply formatting

ws.Range("J22:Q23").Font.Bold = True

ws.Range("J23:Q35").Borders.LineStyle = xlContinuous

' Create "Current Year Payment Value Matrix - by office and by Month" Section

ws.Range("J38").Value = "Current Year Monthly Payment Value - by office and by Month"

ws.Range("J39").Value = "Month"

ws.Range("K39:Q39").Value = Array("Boston", "San Francisco", "NYC", "Paris", "Tokyo", "Sydney", "London")

ws.Range("J40:J51").Value = Application.Transpose(Array("January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December"))

' Merge and format the section header

With ws.Range("J38:N38")

.Merge

.HorizontalAlignment = xlLeft

.VerticalAlignment = xlCenter

.Font.Size = 14

End With

' Apply formatting

ws.Range("J38:Q39").Font.Bold = True

ws.Range("J39:Q51").Borders.LineStyle = xlContinuous

' Set all columns to match width of SLA\_Compliance\_Analysis

For Each col In ws.Columns("A:H")

col.ColumnWidth = 17.57 ' Match width

Next col

' Wrap text in all cells

ws.Cells.WrapText = True

Application.Calculation = xlCalculationAutomatic

Application.ScreenUpdating = True

Application.EnableEvents = True

Application.DisplayStatusBar = True

End Sub

Sub Macro\_12\_Populate\_Payments\_Analysis()

Application.Calculation = xlCalculationManual

Application.ScreenUpdating = False

Application.EnableEvents = False

Application.DisplayStatusBar = False

'

' Populate\_Payments\_Analysis Macro

' Populates payment data, applies conditional formatting, and ensures formulas are applied correctly.

'

Dim ws As Worksheet

Set ws = ThisWorkbook.Sheets("Payments Analysis") ' Reference the Payment Analysis sheet

' --- Populate the first matrix (B8:H19) ---

ws.Range("B8").formula = \_

"=COUNTIFS('Payments Master Sheet'!$E:$E,$B$1,'Payments Master Sheet'!$F:$F,MATCH($A8,{""January"",""February"",""March"",""April"",""May"",""June"",""July"",""August"",""September"",""October"",""November"",""December""},0),'Payments Master Sheet'!$G:$G,B$7) / " & \_

"COUNTIFS('Payments Master Sheet'!$E:$E,$B$1,'Payments Master Sheet'!$G:$G,B$7)"

ws.Range("B8:B19").FillDown ' Drag formula down

ws.Range("B8:H19").FillRight ' Drag formula across

' --- Populate the second matrix (B24:H35) ---

ws.Range("B24").formula = \_

"=COUNTIFS('Payments Master Sheet'!$E:$E,$B$2,'Payments Master Sheet'!$F:$F,MATCH($A24,{""January"",""February"",""March"",""April"",""May"",""June"",""July"",""August"",""September"",""October"",""November"",""December""},0),'Payments Master Sheet'!$G:$G,B$23) / " & \_

"COUNTIFS('Payments Master Sheet'!$E:$E,$B$2,'Payments Master Sheet'!$G:$G,B$7)"

ws.Range("B24:B35").FillDown ' Drag formula down

ws.Range("B24:H35").FillRight ' Drag formula across

' --- Populate the third matrix (B40:H51) ---

ws.Range("B40").formula = \_

"=COUNTIFS('Payments Master Sheet'!$E:$E,$B$3,'Payments Master Sheet'!$F:$F,MATCH($A40,{""January"",""February"",""March"",""April"",""May"",""June"",""July"",""August"",""September"",""October"",""November"",""December""},0),'Payments Master Sheet'!$G:$G,B$39) / " & \_

"((COUNTIFS('Payments Master Sheet'!$E:$E,$B$1,'Payments Master Sheet'!$G:$G,B$39) + " & \_

"COUNTIFS('Payments Master Sheet'!$E:$E,$B$2,'Payments Master Sheet'!$G:$G,B$39)) / 2)"

ws.Range("B40:B51").FillDown ' Drag formula down

ws.Range("B40:H51").FillRight ' Drag formula across

' --- Populate the fourth matrix (K8:Q19) ---

ws.Range("K8").formula = \_

"=SUMIFS('Payments Master Sheet'!$D:$D,'Payments Master Sheet'!$E:$E,$B$1,'Payments Master Sheet'!$F:$F,MATCH($J8,{""January"",""February"",""March"",""April"",""May"",""June"",""July"",""August"",""September"",""October"",""November"",""December""},0),'Payments Master Sheet'!$G:$G,K$7) / " & \_

"SUMIFS('Payments Master Sheet'!$D:$D,'Payments Master Sheet'!$E:$E,$B$1,'Payments Master Sheet'!$G:$G,K$7)"

ws.Range("K8:K19").FillDown ' Drag formula down

ws.Range("K8:Q19").FillRight ' Drag formula across

' --- Populate the fifth matrix (K24:Q35) ---

ws.Range("K24").formula = \_

"=SUMIFS('Payments Master Sheet'!$D:$D,'Payments Master Sheet'!$E:$E,$B$2,'Payments Master Sheet'!$F:$F,MATCH($J24,{""January"",""February"",""March"",""April"",""May"",""June"",""July"",""August"",""September"",""October"",""November"",""December""},0),'Payments Master Sheet'!$G:$G,K$23) / " & \_

"SUMIFS('Payments Master Sheet'!$D:$D,'Payments Master Sheet'!$E:$E,$B$2,'Payments Master Sheet'!$G:$G,K$23)"

ws.Range("K24:K35").FillDown ' Drag formula down

ws.Range("K24:Q35").FillRight ' Drag formula across

' --- Populate the sixth matrix (K40:Q51) ---

ws.Range("K40").formula = \_

"=SUMIFS('Payments Master Sheet'!$D:$D,'Payments Master Sheet'!$E:$E,$B$3,'Payments Master Sheet'!$F:$F,MATCH($J40,{""January"",""February"",""March"",""April"",""May"",""June"",""July"",""August"",""September"",""October"",""November"",""December""},0),'Payments Master Sheet'!$G:$G,K$39) / " & \_

"((SUMIFS('Payments Master Sheet'!$D:$D,'Payments Master Sheet'!$E:$E,$B$1,'Payments Master Sheet'!$G:$G,K$39) + " & \_

"SUMIFS('Payments Master Sheet'!$D:$D,'Payments Master Sheet'!$E:$E,$B$2,'Payments Master Sheet'!$G:$G,K$39)) / 2)"

ws.Range("K40:K51").FillDown ' Drag formula down

ws.Range("K40:Q51").FillRight ' Drag formula across

' --- Apply Conditional Formatting ---

Dim rng As Range

Dim col As Range

' Define ranges for conditional formatting

Set rng = Union(ws.Range("B8:H19"), ws.Range("K8:Q19"), ws.Range("B24:H35"), ws.Range("K24:Q35"), ws.Range("B40:H45"), ws.Range("K40:Q45"))

' Apply rules for each column

For Each col In rng.Columns

' Highlight top two values

With col

.FormatConditions.AddTop10

With .FormatConditions(.FormatConditions.Count)

.SetFirstPriority

.TopBottom = xlTop10Top

.Rank = 2

.Percent = False

.Interior.Color = RGB(198, 239, 206) ' Light green

End With

End With

' Highlight zero values

With col

.FormatConditions.Add Type:=xlCellValue, Operator:=xlEqual, formula1:="=0"

With .FormatConditions(.FormatConditions.Count)

.SetFirstPriority

.Interior.Color = RGB(255, 199, 206) ' Light red

End With

End With

Next col

' --- Format All Ranges as Percentage ---

rng.NumberFormat = "0%" ' Apply percentage format to all specified ranges

Application.Calculation = xlCalculationAutomatic

Application.ScreenUpdating = True

Application.EnableEvents = True

Application.DisplayStatusBar = True

End Sub

Sub Macro\_13\_Macro\_SalesAnalysis\_CreateStructure()

Application.Calculation = xlCalculationManual

Application.ScreenUpdating = False

Application.EnableEvents = False

Application.DisplayStatusBar = False

' Declare variables

Dim ws As Worksheet

Dim col As Range

' Check if "Sales Analysis" sheet exists, and create it if not

On Error Resume Next

Set ws = ThisWorkbook.Sheets("Sales Analysis")

On Error GoTo 0

If ws Is Nothing Then

Set ws = ThisWorkbook.Sheets.Add(After:=ThisWorkbook.Sheets(ThisWorkbook.Sheets.Count))

ws.Name = "Sales Analysis"

Else

ws.Range("A1:XFD52").ClearContents ' Clear the sheet if it exists

End If

' Set values for A1, A2, and A3 in the "Sales Analysis" tab

ws.Range("A1").Value = "Year 1"

ws.Range("A2").Value = "Year 2"

ws.Range("A3").Value = "Current Year"

' Copy the values from the "SLA\_Compliance" sheet to the "Payments Analysis" sheet

On Error Resume Next

If Not ThisWorkbook.Sheets("SLA\_Compliance") Is Nothing Then

' Copy value from B1 in "SLA\_Compliance" to B1 in "Payments Analysis"

ws.Range("B1").Value = ThisWorkbook.Sheets("SLA\_Compliance").Range("B1").Value

' Copy value from B2 in "SLA\_Compliance" to B2 in "Payments Analysis"

ws.Range("B2").Value = ThisWorkbook.Sheets("SLA\_Compliance").Range("B2").Value

' Copy value from E1 in "SLA\_Compliance" to B3 in "Payments Analysis"

ws.Range("B3").Value = ThisWorkbook.Sheets("SLA\_Compliance").Range("E1").Value

Else

MsgBox "Sheet 'SLA\_Compliance' not found. B1, B2, and B3 in 'Payments Analysis' not updated.", vbExclamation

End If

On Error GoTo 0

' Create "Monthly Sales Activity - Year 1 Matrix" Section

ws.Range("A6").Value = "Sales Activity - Year 1 - by Office and by Month"

ws.Range("A7").Value = "Month"

ws.Range("B7:H7").Value = Array("Boston", "San Francisco", "NYC", "Paris", "Tokyo", "Sydney", "London")

ws.Range("A8:A19").Value = Application.Transpose(Array("January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December"))

' Merge and format the section header

With ws.Range("A6:E6")

.Merge

.HorizontalAlignment = xlLeft

.VerticalAlignment = xlCenter

.Font.Size = 14

End With

' Apply formatting

ws.Range("A6:H7").Font.Bold = True

ws.Range("A7:H19").Borders.LineStyle = xlContinuous

' Create "Monthly Sales Activity - Year 2 Matrix" Section

ws.Range("A22").Value = "Sales Activity - Year 2 - by Office and by Month"

ws.Range("A23").Value = "Month"

ws.Range("B23:H23").Value = Array("Boston", "San Francisco", "NYC", "Paris", "Tokyo", "Sydney", "London")

ws.Range("A24:A35").Value = Application.Transpose(Array("January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December"))

' Merge and format the section header

With ws.Range("A22:E22")

.Merge

.HorizontalAlignment = xlLeft

.VerticalAlignment = xlCenter

.Font.Size = 14

End With

' Apply formatting

ws.Range("A22:H23").Font.Bold = True

ws.Range("A23:H35").Borders.LineStyle = xlContinuous

' Create "Monthly Sales Activity - Current Year Matrix" Section

ws.Range("A38").Value = "Sales Activity - Current Year - by Office and by Month"

ws.Range("A39").Value = "Month"

ws.Range("B39:H39").Value = Array("Boston", "San Francisco", "NYC", "Paris", "Tokyo", "Sydney", "London")

ws.Range("A40:A51").Value = Application.Transpose(Array("January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December"))

' Merge and format the section header

With ws.Range("A38:E38")

.Merge

.HorizontalAlignment = xlLeft

.VerticalAlignment = xlCenter

.Font.Size = 14

End With

' Apply formatting

ws.Range("A38:H39").Font.Bold = True

ws.Range("A39:H51").Borders.LineStyle = xlContinuous

' Create "Year 1 - Monthly Sales Value" Section

ws.Range("J6").Value = "Total Revenue - Year 1 - by Office and by Month"

ws.Range("J7").Value = "Month"

ws.Range("K7:Q7").Value = Array("Boston", "San Francisco", "NYC", "Paris", "Tokyo", "Sydney", "London")

ws.Range("J8:J19").Value = Application.Transpose(Array("January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December"))

' Merge and format the section header

With ws.Range("J6:N6")

.Merge

.HorizontalAlignment = xlLeft

.VerticalAlignment = xlCenter

.Font.Size = 14

End With

' Apply formatting

ws.Range("J6:Q7").Font.Bold = True

ws.Range("J7:Q19").Borders.LineStyle = xlContinuous

' Create "Year 2 - Monthly Sales Value" Section

ws.Range("J22").Value = "Total Revenue - Year 2 - by Office and by Month"

ws.Range("J23").Value = "Month"

ws.Range("K23:Q23").Value = Array("Boston", "San Francisco", "NYC", "Paris", "Tokyo", "Sydney", "London")

ws.Range("J24:J35").Value = Application.Transpose(Array("January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December"))

' Merge and format the section header

With ws.Range("J22:N22")

.Merge

.HorizontalAlignment = xlLeft

.VerticalAlignment = xlCenter

.Font.Size = 14

End With

' Apply formatting

ws.Range("J22:Q23").Font.Bold = True

ws.Range("J23:Q35").Borders.LineStyle = xlContinuous

' Create "Current Year Sales Value Matrix - by office and by Month" Section

ws.Range("J38").Value = "Total Revenue - Current Year - by Office and by Month"

ws.Range("J39").Value = "Month"

ws.Range("K39:Q39").Value = Array("Boston", "San Francisco", "NYC", "Paris", "Tokyo", "Sydney", "London")

ws.Range("J40:J51").Value = Application.Transpose(Array("January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December"))

' Merge and format the section header

With ws.Range("J38:N38")

.Merge

.HorizontalAlignment = xlLeft

.VerticalAlignment = xlCenter

.Font.Size = 14

End With

' Apply formatting

ws.Range("J38:Q39").Font.Bold = True

ws.Range("J39:Q51").Borders.LineStyle = xlContinuous

' Create "Year 1 - Monthly Profitability" Section

ws.Range("S6").Value = "Profitability - Year 1 - by Office and by Month"

ws.Range("S7").Value = "Month"

ws.Range("T7:Z7").Value = Array("Boston", "San Francisco", "NYC", "Paris", "Tokyo", "Sydney", "London")

ws.Range("S8:S19").Value = Application.Transpose(Array("January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December"))

' Merge and format the section header

With ws.Range("S6:V6")

.Merge

.HorizontalAlignment = xlLeft

.VerticalAlignment = xlCenter

.Font.Size = 14

End With

' Apply formatting

ws.Range("S6:Z7").Font.Bold = True

ws.Range("S7:Z19").Borders.LineStyle = xlContinuous

' Create "Year 2 - Monthly Profitability" Section

ws.Range("S22").Value = "Profitability - Year 2 - by Office and by Month"

ws.Range("S23").Value = "Month"

ws.Range("T23:Z23").Value = Array("Boston", "San Francisco", "NYC", "Paris", "Tokyo", "Sydney", "London")

ws.Range("S24:S35").Value = Application.Transpose(Array("January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December"))

' Merge and format the section header

With ws.Range("S22:V22")

.Merge

.HorizontalAlignment = xlLeft

.VerticalAlignment = xlCenter

.Font.Size = 14

End With

' Apply formatting

ws.Range("S22:Z23").Font.Bold = True

ws.Range("S23:Z35").Borders.LineStyle = xlContinuous

' Create "Current Year - Monthly Profitability" Section

ws.Range("S38").Value = "Profitability - Current Year - by Office and by Month"

ws.Range("S39").Value = "Month"

ws.Range("T39:Z39").Value = Array("Boston", "San Francisco", "NYC", "Paris", "Tokyo", "Sydney", "London")

ws.Range("S40:S51").Value = Application.Transpose(Array("January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December"))

' Merge and format the section header

With ws.Range("S38:W38")

.Merge

.HorizontalAlignment = xlLeft

.VerticalAlignment = xlCenter

.Font.Size = 14

End With

' Apply formatting

ws.Range("S38:Z39").Font.Bold = True

ws.Range("S39:Z51").Borders.LineStyle = xlContinuous

' Set all columns to match width

For Each col In ws.Columns("A:Z")

col.ColumnWidth = 17.57 ' Match width

Next col

' Wrap text in all cells

ws.Cells.WrapText = True

Application.Calculation = xlCalculationAutomatic

Application.ScreenUpdating = True

Application.EnableEvents = True

Application.DisplayStatusBar = True

End Sub

Sub Macro\_14\_Populate\_Sales\_Analysis()

Application.Calculation = xlCalculationManual

Application.ScreenUpdating = False

Application.EnableEvents = False

Application.DisplayStatusBar = False

'

' Populate\_Sales\_Analysis Macro

' Populates sales data, applies conditional formatting, and ensures formulas are applied correctly.

'

Dim ws As Worksheet

Set ws = ThisWorkbook.Sheets("Sales Analysis") ' Reference the Sales Analysis sheet

' --- Populate the first matrix (B8:H19) ---

ws.Range("B8").formula = \_

"=COUNTIFS('Products & Profit Master Sheet'!$K:$K,$B$1,'Products & Profit Master Sheet'!$L:$L,MATCH($A8,{""January"",""February"",""March"",""April"",""May"",""June"",""July"",""August"",""September"",""October"",""November"",""December""},0),'Products & Profit Master Sheet'!$M:$M,B$7) / " & \_

"COUNTIFS('Products & Profit Master Sheet'!$K:$K,$B$1,'Products & Profit Master Sheet'!$M:$M,B$7)"

ws.Range("B8:B19").FillDown ' Drag formula down

ws.Range("B8:H19").FillRight ' Drag formula across

' --- Populate the second matrix (B24:H35) ---

ws.Range("B24").formula = \_

"=COUNTIFS('Products & Profit Master Sheet'!$K:$K,$B$2,'Products & Profit Master Sheet'!$L:$L,MATCH($A24,{""January"",""February"",""March"",""April"",""May"",""June"",""July"",""August"",""September"",""October"",""November"",""December""},0),'Products & Profit Master Sheet'!$M:$M,B$23) / " & \_

"COUNTIFS('Products & Profit Master Sheet'!$K:$K,$B$2,'Products & Profit Master Sheet'!$M:$M,B$23)"

ws.Range("B24:B35").FillDown ' Drag formula down

ws.Range("B24:H35").FillRight ' Drag formula across

' --- Populate the third matrix (B40:H51) ---

ws.Range("B40").formula = \_

"=COUNTIFS('Products & Profit Master Sheet'!$K:$K,$B$3,'Products & Profit Master Sheet'!$L:$L,MATCH($A40,{""January"",""February"",""March"",""April"",""May"",""June"",""July"",""August"",""September"",""October"",""November"",""December""},0),'Products & Profit Master Sheet'!$M:$M,B$39) / " & \_

"((COUNTIFS('Products & Profit Master Sheet'!$K:$K,$B$1,'Products & Profit Master Sheet'!$M:$M,B$39) + " & \_

"COUNTIFS('Products & Profit Master Sheet'!$K:$K,$B$2,'Products & Profit Master Sheet'!$M:$M,B$39)) / 2)"

ws.Range("B40:B51").FillDown ' Drag formula down

ws.Range("B40:H51").FillRight ' Drag formula across

' --- Populate the fourth matrix (K8:Q19) ---

ws.Range("K8").formula = \_

"=SUMIFS('Products & Profit Master Sheet'!$I:$I,'Products & Profit Master Sheet'!$K:$K,$B$1,'Products & Profit Master Sheet'!$L:$L,MATCH($J8,{""January"",""February"",""March"",""April"",""May"",""June"",""July"",""August"",""September"",""October"",""November"",""December""},0),'Products & Profit Master Sheet'!$M:$M,K$7) / " & \_

"SUMIFS('Products & Profit Master Sheet'!$I:$I,'Products & Profit Master Sheet'!$K:$K,$B$1,'Products & Profit Master Sheet'!$M:$M,K$7)"

ws.Range("K8:K19").FillDown ' Drag formula down

ws.Range("K8:Q19").FillRight ' Drag formula across

' --- Populate the fifth matrix (K24:Q35) ---

ws.Range("K24").formula = \_

"=SUMIFS('Products & Profit Master Sheet'!$I:$I,'Products & Profit Master Sheet'!$K:$K,$B$2,'Products & Profit Master Sheet'!$L:$L,MATCH($J24,{""January"",""February"",""March"",""April"",""May"",""June"",""July"",""August"",""September"",""October"",""November"",""December""},0),'Products & Profit Master Sheet'!$M:$M,K$23) / " & \_

"SUMIFS('Products & Profit Master Sheet'!$I:$I,'Products & Profit Master Sheet'!$K:$K,$B$2,'Products & Profit Master Sheet'!$M:$M,K$23)"

ws.Range("K24:K35").FillDown ' Drag formula down

ws.Range("K24:Q35").FillRight ' Drag formula across

' --- Populate the sixth matrix (K40:Q51) ---

ws.Range("K40").formula = \_

"=SUMIFS('Products & Profit Master Sheet'!$I:$I,'Products & Profit Master Sheet'!$K:$K,$B$3,'Products & Profit Master Sheet'!$L:$L,MATCH($J40,{""January"",""February"",""March"",""April"",""May"",""June"",""July"",""August"",""September"",""October"",""November"",""December""},0),'Products & Profit Master Sheet'!$M:$M,K$39) / " & \_

"((SUMIFS('Products & Profit Master Sheet'!$I:$I,'Products & Profit Master Sheet'!$K:$K,$B$1,'Products & Profit Master Sheet'!$M:$M,K$39) + " & \_

"SUMIFS('Products & Profit Master Sheet'!$I:$I,'Products & Profit Master Sheet'!$K:$K,$B$2,'Products & Profit Master Sheet'!$M:$M,K$39)) / 2)"

ws.Range("K40:K51").FillDown ' Drag formula down

ws.Range("K40:Q51").FillRight ' Drag formula across

' --- Populate the seventh matrix (T8:Z19) ---

ws.Range("T8").formula = \_

"=SUMIFS('Products & Profit Master Sheet'!$J:$J,'Products & Profit Master Sheet'!$K:$K,$B$1,'Products & Profit Master Sheet'!$L:$L,MATCH($S8,{""January"",""February"",""March"",""April"",""May"",""June"",""July"",""August"",""September"",""October"",""November"",""December""},0),'Products & Profit Master Sheet'!$M:$M,T$7) / " & \_

"SUMIFS('Products & Profit Master Sheet'!$J:$J,'Products & Profit Master Sheet'!$K:$K,$B$1,'Products & Profit Master Sheet'!$M:$M,T$7)"

ws.Range("T8:T19").FillDown ' Drag formula down

ws.Range("T8:Z19").FillRight ' Drag formula across

' --- Populate the eighth matrix (T24:Z35) ---

ws.Range("T24").formula = \_

"=SUMIFS('Products & Profit Master Sheet'!$J:$J,'Products & Profit Master Sheet'!$K:$K,$B$2,'Products & Profit Master Sheet'!$L:$L,MATCH($S24,{""January"",""February"",""March"",""April"",""May"",""June"",""July"",""August"",""September"",""October"",""November"",""December""},0),'Products & Profit Master Sheet'!$M:$M,T$23) / " & \_

"SUMIFS('Products & Profit Master Sheet'!$J:$J,'Products & Profit Master Sheet'!$K:$K,$B$2,'Products & Profit Master Sheet'!$M:$M,T$23)"

ws.Range("T24:T35").FillDown ' Drag formula down

ws.Range("T24:Z35").FillRight ' Drag formula across

' --- Populate the ninth matrix (T40:Z51) ---

ws.Range("T40").formula = \_

"=SUMIFS('Products & Profit Master Sheet'!$J:$J,'Products & Profit Master Sheet'!$K:$K,$B$3,'Products & Profit Master Sheet'!$L:$L,MATCH($S40,{""January"",""February"",""March"",""April"",""May"",""June"",""July"",""August"",""September"",""October"",""November"",""December""},0),'Products & Profit Master Sheet'!$M:$M,T$39) / " & \_

"((SUMIFS('Products & Profit Master Sheet'!$J:$J,'Products & Profit Master Sheet'!$K:$K,$B$1,'Products & Profit Master Sheet'!$M:$M,T$39) + " & \_

"SUMIFS('Products & Profit Master Sheet'!$I:$I,'Products & Profit Master Sheet'!$K:$K,$B$2,'Products & Profit Master Sheet'!$M:$M,T$39)) / 2)"

ws.Range("T40:T51").FillDown ' Drag formula down

ws.Range("T40:Z51").FillRight ' Drag formula across

' --- Apply Conditional Formatting ---

Dim rng As Range

Dim col As Range

' Define ranges for conditional formatting

Set rng = Union(ws.Range("B8:H19"), ws.Range("K8:Q19"), ws.Range("B24:H35"), ws.Range("K24:Q35"), ws.Range("B40:H45"), ws.Range("K40:Q45"), ws.Range("T8:Z19"), ws.Range("T24:Z35"), ws.Range("T40:Z45"))

' Apply rules for each column

For Each col In rng.Columns

' Highlight top two values

With col

.FormatConditions.AddTop10

With .FormatConditions(.FormatConditions.Count)

.SetFirstPriority

.TopBottom = xlTop10Top

.Rank = 2

.Percent = False

.Interior.Color = RGB(198, 239, 206) ' Light green

End With

End With

' Highlight zero values

With col

.FormatConditions.Add Type:=xlCellValue, Operator:=xlEqual, formula1:="=0"

With .FormatConditions(.FormatConditions.Count)

.SetFirstPriority

.Interior.Color = RGB(255, 199, 206) ' Light red

End With

End With

Next col

' --- Format All Ranges as Percentage ---

rng.NumberFormat = "0%" ' Apply percentage format to all specified ranges

Application.Calculation = xlCalculationAutomatic

Application.ScreenUpdating = True

Application.EnableEvents = True

Application.DisplayStatusBar = True

End Sub

Sub Macro\_15\_Create\_Summary\_Dashboard\_With\_Chart\_Structures()

'

' Create\_Summary\_Dashboard\_With\_Chart\_Structures Macro

' Dynamically generates the updated structure of the Summary Dashboard with office and month chart structures.

'

Dim ws As Worksheet

Dim lastUpdated As String

Dim months As Variant

Dim offices As Variant

Dim i As Integer, rowStart As Integer

Dim rng As Range

' Create or activate the Summary Dashboard sheet

On Error Resume Next

Set ws = ThisWorkbook.Sheets("Summary Dashboard")

If ws Is Nothing Then

Set ws = ThisWorkbook.Sheets.Add(After:=ThisWorkbook.Sheets(ThisWorkbook.Sheets.Count))

ws.Name = "Summary Dashboard"

End If

On Error GoTo 0

ws.Cells.Clear ' Clear existing content

' Add explanatory note in row 3

ws.Range("A3:N3").Merge

ws.Range("A3:N3").Value = "All data is based on empirical probability (relative frequency)."

ws.Range("A3:N3").Font.Italic = True

ws.Range("A3:N3").Font.Size = 10

ws.Range("A3:N3").HorizontalAlignment = xlCenter

' --- Title Section ---

ws.Range("A1:N1").Merge

ws.Range("A1:N1").Value = "Annual Internal Report – Summary Dashboard"

ws.Range("A1:N1").Font.Bold = True

ws.Range("A1:N1").Font.Size = 16

ws.Range("A1:N1").HorizontalAlignment = xlCenter

ws.Range("A2:N2").Merge

ws.Range("A2:N2").Value = "Comparative Analysis of Sales, Deliveries, and Payments"

ws.Range("A2:N2").Font.Italic = True

ws.Range("A2:N2").Font.Size = 12

ws.Range("A2:N2").HorizontalAlignment = xlCenter

' Add Last Updated Timestamp

ws.Range("M4").Value = "Last Updated: " & Now()

ws.Range("M4").Font.Size = 10

ws.Range("M4").Font.Italic = True

' --- SLA Compliance Section ---

ws.Range("A5:N5").Merge

ws.Range("A5:N5").Value = "SLA Compliance Summary"

ws.Range("A5:N5").Font.Bold = True

ws.Range("A5:N5").Font.Size = 14

' Add column headers for SLA Compliance

ws.Range("A6").Value = "Office"

ws.Range("B6").Value = "Historical Compliance (%)"

ws.Range("C6").Value = "Recent Compliance (%)"

ws.Range("D6").Value = "Deviation (%)"

ws.Range("E6").Value = "Historical Contribution to Total (%)"

ws.Range("F6").Value = "Recent Contribution to Total (%)"

ws.Range("G6").Value = "Deviation in Contribution (%)"

ws.Range("A6:G6").Font.Bold = True

' Populate office names in SLA Compliance

offices = Array("Boston", "San Francisco", "NYC", "Paris", "Tokyo", "Sydney", "London")

For i = LBound(offices) To UBound(offices)

ws.Range("A7").Offset(i, 0).Value = offices(i)

Next i

' Set background color for SLA Compliance Summary

ws.Range("A7:G13").Interior.Color = RGB(242, 242, 242) ' Light gray for SLA Compliance table

' Add borders to SLA Compliance table

Set rng = ws.Range("A6:G13")

With rng.Borders

.LineStyle = xlContinuous

.Weight = xlThin

.ColorIndex = 1 ' Black border color

End With

' Apply alternating row colors to SLA Compliance table

For i = 7 To 13

If i Mod 2 = 0 Then

ws.Range("A" & i & ":G" & i).Interior.Color = RGB(255, 255, 255) ' White

Else

ws.Range("A" & i & ":G" & i).Interior.Color = RGB(242, 242, 242) ' Light gray

End If

Next i

' --- Define Months ---

months = Array("January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December")

' --- Add Sections for Each Chart ---

Dim sections As Variant

sections = Array( \_

"Year 1 Monthly Payments Received (Money Coming In) – by Office and Month", \_

"Year 2 Payments Received (Money Coming In) – by Office and Month", \_

"Current Year Payments Received (Money Coming In) – by Office and Month", \_

"Year 1 Total Sales Revenue (Money Going Out) – by Office and Month", \_

"Year 2 Total Sales Revenue (Money Going Out) – by Office and Month", \_

"Current Year Total Sales Revenue (Money Going Out) – by Office and Month" \_

)

rowStart = 15 ' Start below the SLA Compliance Summary

For i = LBound(sections) To UBound(sections)

' Add section title

ws.Range("A" & rowStart & ":N" & rowStart).Merge

ws.Range("A" & rowStart).Value = sections(i)

ws.Range("A" & rowStart).Font.Bold = True

ws.Range("A" & rowStart).Font.Size = 14

' Add Office Names as headers

Dim k As Integer

For k = 0 To UBound(offices)

ws.Cells(rowStart + 1, 2 + k).Value = offices(k) ' Start from column B

Next k

ws.Range("B" & (rowStart + 1) & ":H" & (rowStart + 1)).Font.Bold = True

' Add Months to the first column

Dim j As Integer

For j = 0 To UBound(months)

ws.Range("A" & (rowStart + 2 + j)).Value = months(j)

Next j

' Format table background

ws.Range("A" & (rowStart + 2) & ":H" & (rowStart + 13)).Interior.Color = RGB(242, 242, 242) ' Light gray for the table structure

' Add borders to the table

Set rng = ws.Range("A" & (rowStart + 1) & ":H" & (rowStart + 13))

With rng.Borders

.LineStyle = xlContinuous

.Weight = xlThin

.ColorIndex = 1 ' Black border color

End With

' Apply alternating row colors

For j = 0 To 11

If j Mod 2 = 0 Then

ws.Range("A" & (rowStart + 2 + j) & ":H" & (rowStart + 2 + j)).Interior.Color = RGB(255, 255, 255) ' White

Else

ws.Range("A" & (rowStart + 2 + j) & ":H" & (rowStart + 2 + j)).Interior.Color = RGB(242, 242, 242) ' Light gray

End If

Next j

' Update rowStart for next section

rowStart = rowStart + 15 ' Ensure enough spacing between sections

Next i

' Autofit columns for readability

ws.Columns("A:N").AutoFit

End Sub

Sub Macro\_16\_Populate\_All\_Charts\_In\_Summary\_Dashboard()

Dim ws As Worksheet

Dim rng As Range

Dim i As Integer, j As Integer

' Reference the Summary Dashboard sheet

Set ws = ThisWorkbook.Sheets("Summary Dashboard")

' --- SLA Compliance Summary ---

With ws

' Add formulas

.Range("B7").formula = "=SLA\_Compliance\_Analysis!B25"

.Range("B7:B13").FillDown ' Drag formulas down

.Range("B7:G13").FillRight ' Drag formulas across

' Reapply formatting

.Range("A7:G13").Borders.LineStyle = xlContinuous ' Add borders

.Range("B7:G13").HorizontalAlignment = xlCenter ' Center align text

' Set background color

.Range("A7:G13").Interior.Color = RGB(242, 242, 242) ' Light gray for SLA Compliance table

' Add alternating row colors

For i = 7 To 13

If i Mod 2 = 0 Then

.Range("A" & i & ":G" & i).Interior.Color = RGB(255, 255, 255) ' White

Else

.Range("A" & i & ":G" & i).Interior.Color = RGB(242, 242, 242) ' Light gray

End If

Next i

End With

' --- Chart Sections ---

Dim sections As Variant

Dim rowStart As Integer

sections = Array( \_

"Year 1 Monthly Payments Received (Money Coming In)", \_

"Year 2 Payments Received (Money Coming In)", \_

"Current Year Payments Received (Money Coming In)", \_

"Year 1 Total Sales Revenue (Money Going Out)", \_

"Year 2 Total Sales Revenue (Money Going Out)", \_

"Current Year Total Sales Revenue (Money Going Out)" \_

)

Dim formulas As Variant

formulas = Array( \_

"='Payments Analysis'!K8", \_

"='Payments Analysis'!K24", \_

"='Payments Analysis'!K40", \_

"='Sales Analysis'!K8", \_

"='Sales Analysis'!K24", \_

"='Sales Analysis'!K40" \_

)

rowStart = 17

For i = LBound(sections) To UBound(sections)

With ws

' Add formulas

.Range("B" & rowStart).formula = formulas(i)

.Range("B" & rowStart & ":B" & (rowStart + 11)).FillDown ' Drag formulas down

.Range("B" & rowStart & ":H" & (rowStart + 11)).FillRight ' Drag formulas across

' Reapply formatting

.Range("A" & rowStart & ":H" & (rowStart + 11)).Borders.LineStyle = xlContinuous ' Add borders

.Range("B" & rowStart & ":H" & (rowStart + 11)).HorizontalAlignment = xlCenter ' Center align text

.Range("A" & rowStart & ":A" & (rowStart + 11)).Font.Bold = True ' Bold months

' Set background color

.Range("A" & rowStart & ":H" & (rowStart + 11)).Interior.Color = RGB(242, 242, 242) ' Light gray for table structure

' Add alternating row colors

For j = 0 To 11

If j Mod 2 = 0 Then

.Range("A" & (rowStart + j) & ":H" & (rowStart + j)).Interior.Color = RGB(255, 255, 255) ' White

Else

.Range("A" & (rowStart + j) & ":H" & (rowStart + j)).Interior.Color = RGB(242, 242, 242) ' Light gray

End If

Next j

End With

' Update rowStart for the next chart

rowStart = rowStart + 15 ' Move down for next section

Next i

End Sub

Sub Macro\_17\_Apply\_All\_Conditional\_Formatting()

Dim ws As Worksheet

Dim rngChart As Range

Dim col As Range

Dim sections As Variant

Dim i As Integer

Dim rngGreen As Range, rngRed As Range

' Reference the Summary Dashboard sheet

Set ws = ThisWorkbook.Sheets("Summary Dashboard")

' --- Conditional Formatting for SLA Compliance Summary ---

' Green Conditional Formatting

Set rngGreen = ws.Range("B7:C13, E7:F13") ' Compliance Columns

With rngGreen

.FormatConditions.AddDatabar

With .FormatConditions(.FormatConditions.Count)

.ShowValue = True

.SetFirstPriority

.MinPoint.Modify newtype:=xlConditionValueNumber, newvalue:=0

.MaxPoint.Modify newtype:=xlConditionValueNumber, newvalue:=1

.BarColor.Color = RGB(0, 176, 80) ' Green

End With

End With

' Red Conditional Formatting

Set rngRed = ws.Range("D7:D13, G7:G13") ' Deviation Columns

With rngRed

.FormatConditions.AddDatabar

With .FormatConditions(.FormatConditions.Count)

.ShowValue = True

.SetFirstPriority

.MinPoint.Modify newtype:=xlConditionValueNumber, newvalue:=0

.MaxPoint.Modify newtype:=xlConditionValueNumber, newvalue:=1

.BarColor.Color = RGB(255, 0, 0) ' Red

End With

End With

' --- Conditional Formatting for Charts ---

' Define the chart ranges for all sections

sections = Array( \_

"B17:H28", \_

"B32:H43", \_

"B47:H52", \_

"B62:H73", \_

"B77:H88", \_

"B92:H97" \_

)

' Loop through each section

For i = LBound(sections) To UBound(sections)

Set rngChart = ws.Range(sections(i))

' Apply light red fill for 0% values

With rngChart

.FormatConditions.Add Type:=xlCellValue, Operator:=xlEqual, formula1:="=0"

With .FormatConditions(.FormatConditions.Count)

.Interior.Color = RGB(255, 204, 204) ' Light red

End With

End With

' Loop through each column in the range to apply top 2 green formatting

For Each col In rngChart.Columns

With col

.FormatConditions.AddTop10

With .FormatConditions(.FormatConditions.Count)

.TopBottom = xlTop10Top

.Rank = 2 ' Top 2 values

.Percent = False

.Interior.Color = RGB(198, 239, 206) ' Light green

End With

End With

Next col

Next i

End Sub

Sub Macro\_99\_DataPreparation()

Application.ScreenUpdating = False

Call Macro\_02\_Populate\_Master\_Data\_tab

Call Macro\_03\_PopulatePaymentsMaster

Call Macro\_04\_PopulateProductsAndProfitMaster

Call Macro\_04A\_FilterStatuses

Application.ScreenUpdating = True

MsgBox "Data Preparation Completed!", vbInformation, "Step 2 Complete"

End Sub

Sub Macro\_99\_PaymentsAnalysis()

Application.ScreenUpdating = False

Call Macro\_11\_PaymentsAnalysisMacro1\_CreatePaymentsAnalysisStructure

Call Macro\_12\_Populate\_Payments\_Analysis

Application.ScreenUpdating = True

MsgBox "Payments Analysis Completed!", vbInformation, "Step 4 Complete"

End Sub

Sub Macro\_99\_SalesAnalysis()

Application.ScreenUpdating = False

Call Macro\_13\_Macro\_SalesAnalysis\_CreateStructure

Call Macro\_14\_Populate\_Sales\_Analysis

Application.ScreenUpdating = True

MsgBox "Sales Analysis Completed!", vbInformation, "Step 5 Complete"

End Sub

Sub Macro\_99\_SummaryDashboardSetup()

Application.ScreenUpdating = False

Call Macro\_15\_Create\_Summary\_Dashboard\_With\_Chart\_Structures

Call Macro\_16\_Populate\_All\_Charts\_In\_Summary\_Dashboard

Call Macro\_17\_Apply\_All\_Conditional\_Formatting

Application.ScreenUpdating = True

MsgBox "Summary Dashboard Setup Completed!", vbInformation, "Step 6 Complete"

End Sub

Sub Macro\_99A\_SLAComplianceDataPreparation()

Application.ScreenUpdating = False

Call Macro\_05\_SLAMacro1\_PopulateSLACompliance

Call Macro\_06\_SLAMacro2\_IdentifyYearsBasedOnMonths

Call Macro\_07\_SLAMacro3\_PopulateFormulasinSLACompliance

Application.ScreenUpdating = True

MsgBox "SLA Compliance Data Preparation Completed!", vbInformation, "Step 3A Complete"

End Sub

Sub Macro\_99B\_SLAComplianceAnalysisAndVisuals()

Application.ScreenUpdating = False

Call Macro\_08\_SLAMacro4\_SetComplianceAnalysisStructure

Call Macro\_09\_SLAMacro5\_PopulateSLAComplianceAnalysis

Application.ScreenUpdating = True

MsgBox "SLA Compliance Analysis Completed!", vbInformation, "Step 3B Complete"

End Sub