```
USE [SalesAnalytics];
GO

-- 2. Create the staging table with columns matching your Google Sheet/Python DataFrame
CREATE TABLE Sales_Dataset (
    [OrderID] VARCHAR(50) NOT NULL, -- Assuming OrderID is a unique identifier
    [OrderDate] DATE NULL,
    [CustomerID] VARCHAR(50) NULL,
    [CustomerSegment] VARCHAR(100) NULL,
    [Region] VARCHAR(50) NULL,
    [Country] VARCHAR(100) NULL,
    [ProductCategory] VARCHAR(100) NULL,
    [ProductSubCategory] VARCHAR(100) NULL,
    [ProductName] VARCHAR(255) NULL,
    [SalesAmount] DECIMAL(18, 2) NULL,
    [Quantity] INT NULL,
    [Discount] DECIMAL(5, 2) NULL, -- Assuming discount is a percentage or factor (e.g., 0.15)
    [Profit] DECIMAL(18, 2) NULL,
    [PaymentMode] VARCHAR(50) NULL,
    [ShippingDays] INT NULL,
    [DeliveryStatus] VARCHAR(50) NULL
);
GO

-- Optional: Verify the table was created successfully
SELECT 'Sales_Dataset table created successfully.' AS Status;
SELECT * FROM Sales_Dataset;
GO
```

```
USE [SalesAnalytics];
GO

-- 1. Create the Staging Table with the correct columns
IF OBJECT_ID('Live_Data_Staging', 'U') IS NOT NULL
    DROP TABLE Live_Data_Staging;
GO

CREATE TABLE Live_Data_Staging (
    [OrderID] VARCHAR(50) NOT NULL,
    [OrderDate] DATE NULL,
    [CustomerID] VARCHAR(50) NULL,
    [CustomerSegment] VARCHAR(100) NULL,
    [Region] VARCHAR(50) NULL,
    [Country] VARCHAR(100) NULL,
    [ProductCategory] VARCHAR(100) NULL,
    [ProductSubCategory] VARCHAR(100) NULL,
    [ProductName] VARCHAR(255) NULL,
    [SalesAmount] DECIMAL(18, 2) NULL,
    [Quantity] INT NULL,
    [Discount] DECIMAL(5, 2) NULL,
    [Profit] DECIMAL(18, 2) NULL,
    [PaymentMode] VARCHAR(50) NULL,
    [ShippingDays] INT NULL,
    [DeliveryStatus] VARCHAR(50) NULL
);
GO

SELECT 'Live_Data_Staging created successfully.' AS Status;
```

```
USE [SalesAnalytics];
GO

-- Drop and recreate the procedure
IF OBJECT_ID('Process_Live_Sales_Data', 'P') IS NOT NULL
   DROP PROCEDURE Process_Live_Sales_Data;
GO

CREATE PROCEDURE Process_Live_Sales_Data
AS
BEGIN
   SET NOCOUNT ON;

   MERGE [SalesAnalytics].[dbo].[Sales_Dataset] AS T  -- Target (Final Table)
   USING [SalesAnalytics].[dbo].[Live_Data_Staging] AS S    -- Source (Staging Table)
   ON (T.[OrderID] = S.[OrderID])

   WHEN MATCHED THEN
      UPDATE SET
         -- --- START FULL UPDATE LIST ---
         T.[OrderDate]          = S.[OrderDate],
         T.[CustomerID]         = S.[CustomerID],
         T.[CustomerSegment]    = S.[CustomerSegment],
         T.[Region]             = S.[Region],
         T.[Country]            = S.[Country],
         T.[ProductCategory]    = S.[ProductCategory],
         T.[ProductSubCategory]  = S.[ProductSubCategory],
         T.[ProductName]        = S.[ProductName],
         T.[SalesAmount]        = S.[SalesAmount],
         T.[Quantity]           = S.[Quantity],
         T.[Discount]           = S.[Discount],
         T.[Profit]             = S.[Profit],
         T.[PaymentMode]        = S.[PaymentMode],
         T.[ShippingDays]       = S.[ShippingDays],
         T.[DeliveryStatus]     = S.[DeliveryStatus]
         -- --- END FULL UPDATE LIST ---

   WHEN NOT MATCHED BY TARGET THEN
      INSERT (
         [OrderID], [OrderDate], [CustomerID], [CustomerSegment], [Region], [Country],
         [ProductCategory], [ProductSubCategory], [ProductName], [SalesAmount],
[Quantity],
         [Discount], [Profit], [PaymentMode], [ShippingDays], [DeliveryStatus]
      )
      VALUES (
         S.[OrderID], S.[OrderDate], S.[CustomerID], S.[CustomerSegment], S.[Region],
S.[Country],
         S.[ProductCategory], S.[ProductSubCategory], S.[ProductName],
S.[SalesAmount], S.[Quantity],
         S.[Discount], S.[Profit], S.[PaymentMode], S.[ShippingDays], S.[DeliveryStatus]
      );
END
GO
```