# Title: Implement SCD Type 1

## **Step 1: Create Tables**

### **Create the Sales\_Dimension Table**:

This table will store the main Sales dimension data.

CREATE TABLE Sales\_Dimension\_Priya (

    SalesID INT PRIMARY KEY,

    CustomerID INT,

    ProductID INT,

    SalesAmount DECIMAL(10, 2),

    SalesDate DATE

);

### **Create the Incoming\_Sales\_Data Table**:

This table simulates the incoming updates (new or changed data).

CREATE TABLE IncomingSales\_Dimension\_Priya (

    SalesID INT PRIMARY KEY,

    CustomerID INT,

    ProductID INT,

    SalesAmount DECIMAL(10, 2),

    SalesDate DATE

);

## **Step 2: Insert Initial Data into Sales\_Dimension**

Let’s insert some sample Sales records into the Sales\_Dimension table.

INSERT INTO Sales\_Dimension\_Priya (SalesID, CustomerID, ProductID, SalesAmount, SalesDate) VALUES

(1, 101, 1001, 250.75, '2024-01-15'),

(2, 102, 1002, 150.50, '2024-02-20'),

(3, 103, 1003, 300.00, '2024-03-10'),

(4, 104, 1004, 450.25, '2024-04-05'),

(5, 105, 1005, 500.00, '2024-05-25');

## **Step 3: Insert Updated Data into Incoming\_Sales\_Data**

Simulate incoming updated Sales data. Some records will have changes, and others will remain the same.

INSERT INTO IncomingSales\_Dimension\_Priya (SalesID, CustomerID, ProductID, SalesAmount, SalesDate) VALUES

(3, 103, 1003, 570.00, '2024-03-18'),-- changed salesamount , date

(4, 104, 1004, 678.25, '2024-04-25'), -- changed salesamount , date

(5, 105, 1005, 789.00, '2024-05-30'); -- changed salesamount , date

## **Step 4: Implement SCD Type 1 Logic**

We will use the **MERGE** statement to compare the Incoming\_Sales\_Data table with the Sales\_Dimension table.

* If a **match** is found based on SalesID, the existing data will be **overwritten**.
* If there’s a new record (not matched), it will be **inserted**.

### **SCD Type 1 MERGE Statement**:

MERGE INTO [dbo].[Sales\_Dimension\_Priya] AS target

USING [dbo].[IncomingSales\_Dimension\_Priya] As source

ON target.SalesID = source.SalesID

WHEN MATCHED THEN

    UPDATE SET

        target.SalesAmount = source.SalesAmount,

        target.SalesDate = source.SalesDate

    WHEN NOT MATCHED THEN

    INSERT (SalesID, CustomerID, ProductID, SalesAmount, SalesDate)

    VALUES(Source.SalesID, Source.CustomerID, Source.ProductID, Source.SalesAmount, Source.SalesDate);

### **Explanation**:

1. **MERGE INTO**: Targets the Sales\_Dimension table.
2. **USING**: Specifies the Incoming\_Sales\_Data table as the source.
3. **ON**: Matches records based on the SalesID.
4. **WHEN MATCHED**: If a match is found, updates the record in the Sales\_Dimension table.
5. **WHEN NOT MATCHED**: If no match is found, inserts the record as new.

## **Step 5: Validate the Results**

Run the following query to check the updated data in the Sales\_Dimension table:

select \* from Sales\_Dimension\_Priya

**Expected Output** (Final Updated Table):

A screenshot of a computer

Description automatically generated

After Merge :

A screenshot of a computer

Description automatically generated

## **Summary of Changes**:

1. SalesID **2**:
   * Email updated to bob.j@email.com
   * City updated to San Francisco
2. SalesID **3**:
   * Email updated to charlie.b@email.com
3. SalesID **4**:
   * City updated to Dallas
4. SalesID **1**:
   * Remained unchanged as there was no incoming update.

## **Clean-up (Optional)**:

If you want to reset your tables for further testing:

TRUNCATE TABLE Sales\_Dimension;

TRUNCATE TABLE Incoming\_Sales\_Data;

### **Key Notes**:

* **MERGE** is the most efficient way to implement SCD Type 1 in SQL.
* Azure SQL Database fully supports the MERGE statement for such operations.
* You can integrate these scripts into stored procedures or pipelines in Azure Data Factory for automation.

Let me know if you need further assistance or enhancements! 😊