# Title: Implement SCD Type 2

To implement SCD type-2 in the Customers table we need to track the historical changes for each customer.

## **Step 1: Create Tables**

### **Create the** Customers\_anu  **Table**:

This table will store the main customers data.

CREATE TABLE Customers\_anu (

CustomerID INT ,

FirstName VARCHAR(50),

LastName VARCHAR(50),

Email VARCHAR(100),

PhoneNumber VARCHAR(20),

Address VARCHAR(200),

StartDate DATE,

EndDate Date,

CurrentFlag CHAR(1),

Primary key(CustomerID,StartDate)

);

CREATE TABLE NewCustomers(

CustomerID INT ,

FirstName VARCHAR(50),

LastName VARCHAR(50),

Email VARCHAR(100),

PhoneNumber VARCHAR(20),

Address VARCHAR(200),

StartDate DATE,

EndDate Date,

CurrentFlag CHAR(1),

Primary key(CustomerID,StartDate)

);

## **Step 2: Insert Initial Data into** Customers\_anu

Let’s insert some sample customer records into the Customers\_anu  table.

When inserting data for the first time, we will set the StartDate to the date of insertione, the EndDate to NULL.

INSERT INTO Customers\_anu (CustomerID, FirstName, LastName, Email, PhoneNumber, Address,EffectiveDate,EndDate,CurrentFlag)

VALUES

(1, 'John', 'Doe', 'john.doe@example.com', '123-456-7890', '123 Elm St','2024-01-01',NULL,'Y'),

(2, 'Jane', 'Smith', 'jane.smith@example.com', '234-567-8901', '456 Oak St','2024-01-02',NULL,'Y'),

(3, 'Mary', 'Johnson', 'mary.johnson@example.com', '345-678-9012', '789 Pine St','2024-01-03',NULL,'Y'),

(4, 'Michael', 'Brown', 'michael.brown@example.com', '456-789-0123', '101 Maple St','2024-01-04',NULL,'Y'),

(5, 'David', 'Williams', 'david.williams@example.com', '567-890-1234', '202 Birch St','2024-01-05',NULL,'Y'),

(6, 'Sarah', 'Jones', 'sarah.jones@example.com', '678-901-2345', '303 Cedar St','2024-01-06',NULL,'Y'),

(7, 'Emily', 'Miller', 'emily.miller@example.com', '789-012-3456', '404 Ash St','2024-01-07',NULL,'Y'),

(8, 'James', 'Davis', 'james.davis@example.com', '890-123-4567', '505 Redwood St','2024-01-08',NULL,'Y'),

(9, 'Robert', 'Garcia', 'robert.garcia@example.com', '901-234-5678', '606 Fir St','2024-01-09',NULL,'Y'),

(10, 'Linda', 'Martinez', 'linda.martinez@example.com', '012-345-6789', '707 Pine St','2024-01-10',NULL,'Y');

## **Insert Initial Data into** NewCustomers

INSERT INTO NewCustomers (CustomerID, FirstName, LastName, Email, PhoneNumber, Address,StartDate,EndDate,CurrentFlag)

VALUES

(1, 'John', 'Doe', 'john.doe@example.com', '987-654-321', '123 New Elm St','2024-12-19',NULL,'Y'),

(2, 'Jane', 'Smith', 'jane.smith@example.com', '970-356-2946', '456 New Oak St','2024-12-19',NULL,'Y'),

(3, 'Mary', 'Johnson', 'mary.johnson@example.com', '630-931-2083', '789 New Pine St','2024-12-19',NULL,'Y'),

(4, 'Michael', 'Brown', 'michael.brown@example.com', '345-567-9867', '101 New Maple St','2024-12-19',NULL,'Y'),

(5, 'David', 'Williams', 'david.williams@example.com', '890-346-9345', '202 New Birch St','2024-12-19',NULL,'Y'),

(6, 'Sarah', 'Jones', 'sarah.jones@example.com', '124-287-9023', '303 New Cedar St','2024-12-19',NULL,'Y'),

(7, 'Emily', 'Miller', 'emily.miller@example.com', '235-653-1909', '404 New Ash St','2024-12-19',NULL,'Y'),

(8, 'James', 'Davis', 'james.davis@example.com', '288-999-0030', '505 New Redwood St','2024-12-19',NULL,'Y'),

(9, 'Robert', 'Garcia', 'robert.garcia@example.com', '944-777-8888', '606 New Fir St','2024-12-19',NULL,'Y'),

(10, 'Linda', 'Martinez', 'linda.martinez@example.com', '012-388-1111', '707 New Pine St','2024-12-19',NULL,'Y');

Merge Operation to handle changes from Customer\_anu into New Customer(Historical data)

MERGE INTO Customers\_anu AS target

USING (SELECT \* FROM NewCustomers) AS source

ON (target.CustomerID = source.CustomerID AND target.CurrentFlag = 'Y')

-- When there are changes in the data (such as address or phone number)

WHEN MATCHED AND (

target.FirstName != source.FirstName

OR target.LastName != source.LastName

OR target.Email != source.Email

OR target.PhoneNumber != source.PhoneNumber

OR target.Address != source.Address) THEN

-- Update the old record by marking it as inactive

UPDATE SET target.EndDate = Current\_Date, target.CurrentFlag = 'N'

-- When no matching record is found in the target table, insert the new record

WHEN NOT MATCHED BY TARGET THEN

INSERT (CustomerID, FirstName, LastName, Email, PhoneNumber, Address, StartDate, EndDate, CurrentFlag)

VALUES (source.CustomerID, source.FirstName, source.LastName, source.Email, source.PhoneNumber, source.Address, Current\_Date, NULL, 'Y');

### **Explanation**:

Matching Condition: The MERGE looks for records where the CustomerID \_id matches between Customers\_anu and NewCustomers and only considers active records from Customers\_anu.

When Matched and Changed: If there are any changes (such as first name, last name, email, or phone number), the old record in NewCustomers is closed by setting end\_date and marking active = FALSE.

When Not Matched by Target: If no matching record is found in NewCustomers, a new historical record is inserted into NewCustomers .

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

Query to View the Data

To check the updated and merged data:

View Current Active Records

sql

Copy code

SELECT \*

FROM Customers\_anu

WHERE CurrentFlag = ‘Y’;

View Historical Records (Merged Data in customers\_2)

sql

Copy code

SELECT \*

FROM NewCustomers

ORDER BY start\_date;

**Summary of Changes:**

Create Tables: We created Customers\_anu for current customer data and NewCustomers for historical data.

Insert Records: Inserted 10 records into Customers\_anu to simulate customer data.

Update Records: Updated customer information (e.g., email changes) and marked the old records as inactive.

Insert New Records: Inserted new customer records into Customers\_anu.

Merge Operation: Used the MERGE statement to move the data from Customers\_anu into NewCustomers, tracking changes and preserving the history.

This approach ensures that we maintain both current and historical customer data while implementing NewCustomers to track changes over time.

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

If you want to reset your tables for further testing:

TRUNCATE TABLE Customers\_anu;

TRUNCATE TABLE NewCustomers;