# Phase 4: Process Automation

Step 1: Validation Rules

Validation Rules prevent wrong or incomplete data from being saved.

A. CarbonFootprint – Prevent Negative CO₂ Saved

- Goal: Ensure users don't enter negative values.

- Setup → Object Manager → CarbonFootprint__c → Validation Rules → New.

- Rule Name: Prevent_Negative_CO2

- Formula:

  $$ISBLANK(CO2\_Saved\_\_c) \,||\, CO2\_Saved\_\_c < 0$$

- Error Message: "CO₂ Saved cannot be negative."

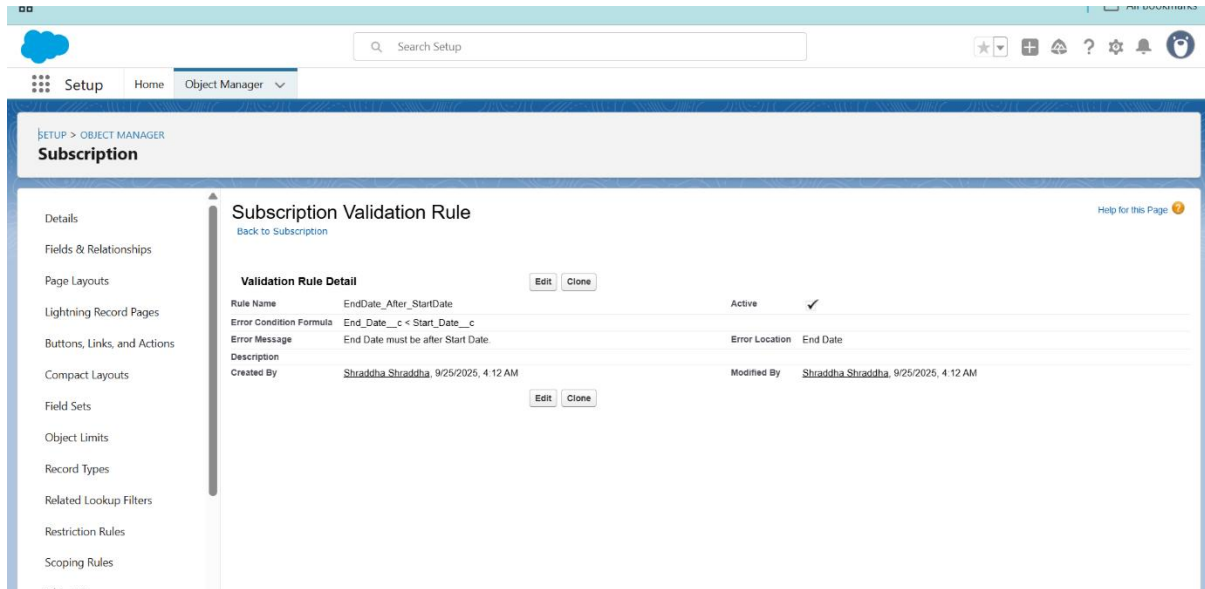- Error Location: Field → CO2_Saved__c.

- Save.
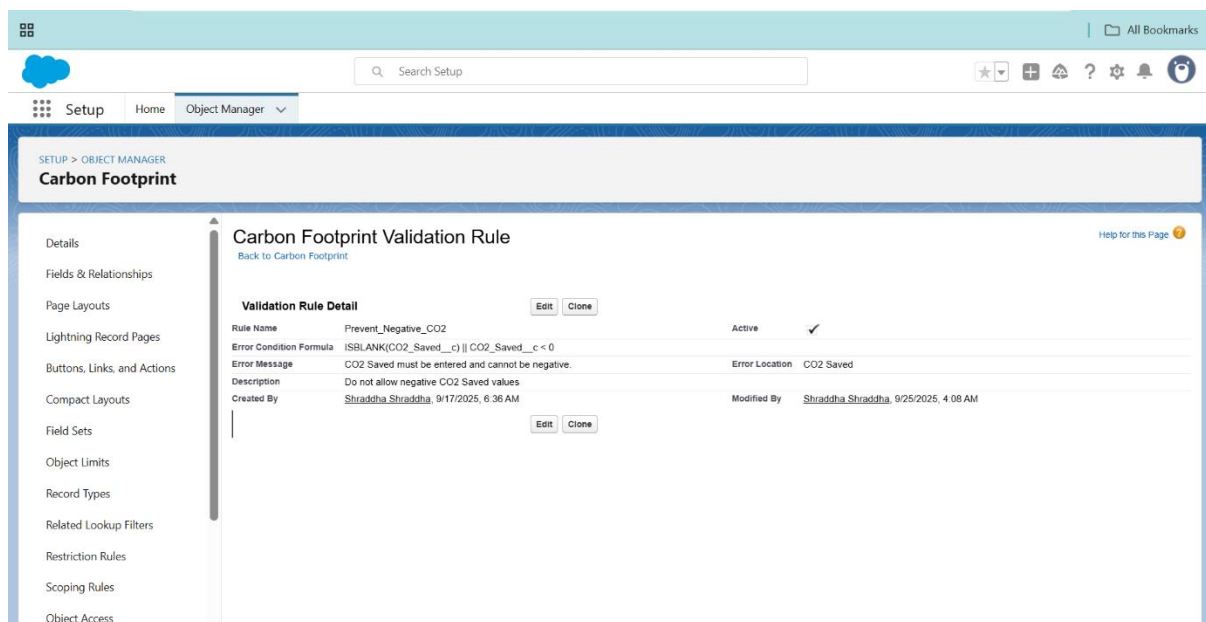
B. Subscription – End Date Must Be After Start Date

Goal: Ensure that a subscription's End Date is always after the Start Date.

- Setup → Object Manager → Subscription__c → Validation Rules → New.

- Rule Name: EndDate_After_StartDate

- Formula:

- End_Date__c < Start_Date__c

- Error Message:
  "End Date must be after Start Date."

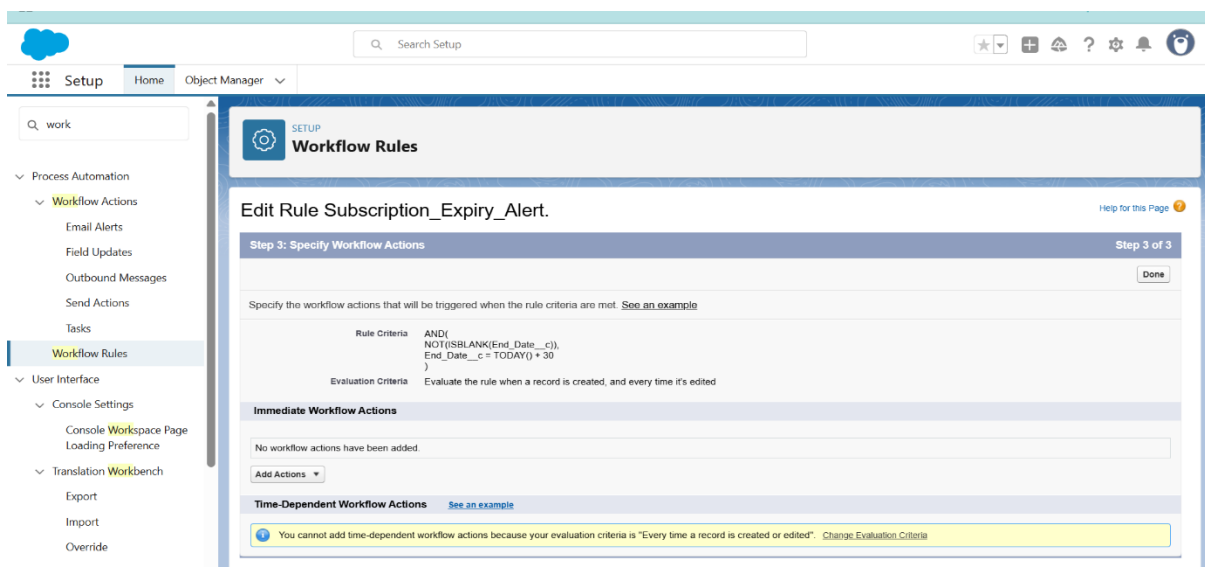- Error Location: Field → End_Date__c.

Step 2: Workflow Rule + Email Alert

Workflow Rule: Notify when a Subscription is expiring.

- Setup → Workflow Rules → New Rule.

- Object: Subscription__c.

- Rule Name: Subscription_Expiry_Alert.

- Rule Criteria:

 AND(NOT(ISBLANK(End_Date__c)),End_Date__c = TODAY() + 30)

Create Email Template

- Go to **Setup → Classic Email Templates → New**.

- **Folder:** Unfiled Public.

- **Template Name:** Subscription_CO2_Update

- **Subject:** CO_2 Update Notification

- **Body:**

  > Hello {!Subscription__c.Contact__c},
  > Your subscription "{!Subscription__c.Name}" has a new CO_2 update.
  > Current CO_2: {!Subscription__c.CO2__c} kg.
  > Thank you,
  > GreenEnergyTracker Team

- Save the template.
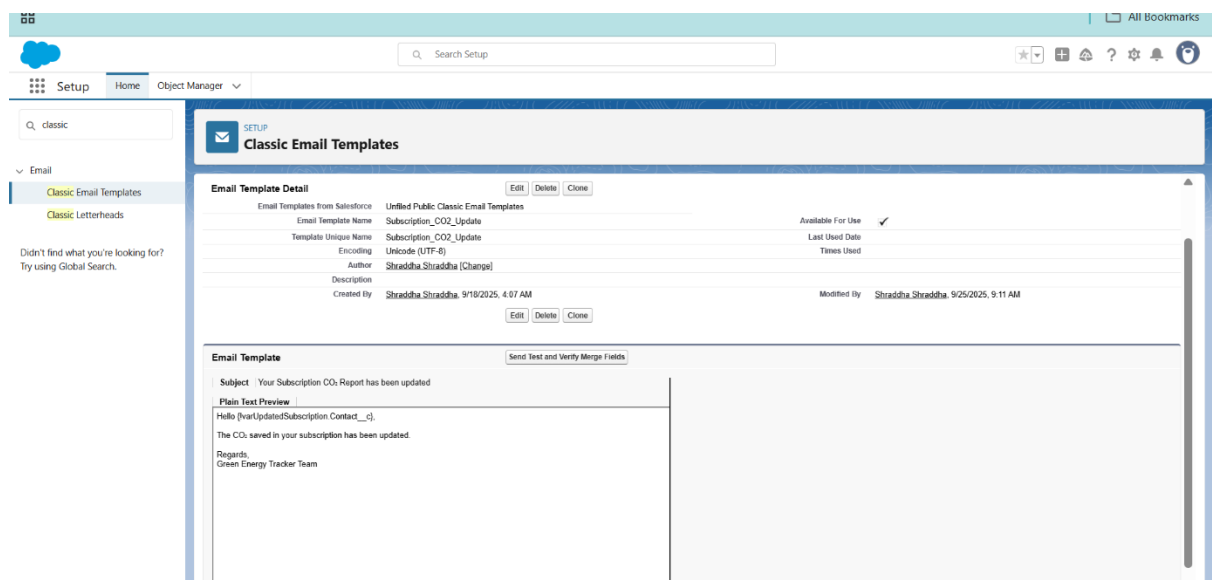
Step 3: Process Builder

Automation Process Builder automates record updates or actions.

Start the Process

- **Object:** Subscription__c
- **Start the process:** when a record is created or edited.

Define Criteria

Use "Formula Evaluates to True"

1. Criteria Name: CO2_Value_Updated
2. Criteria for Executing Actions: Conditions are met → Choose "Formula evaluates to true"
3. Formula:
4. ISCHANGED([Subscription__c].CO2__c)
5. Do not select the advanced option "Only when specified changes are made to the record."

This allows ISCHANGED to work properly.
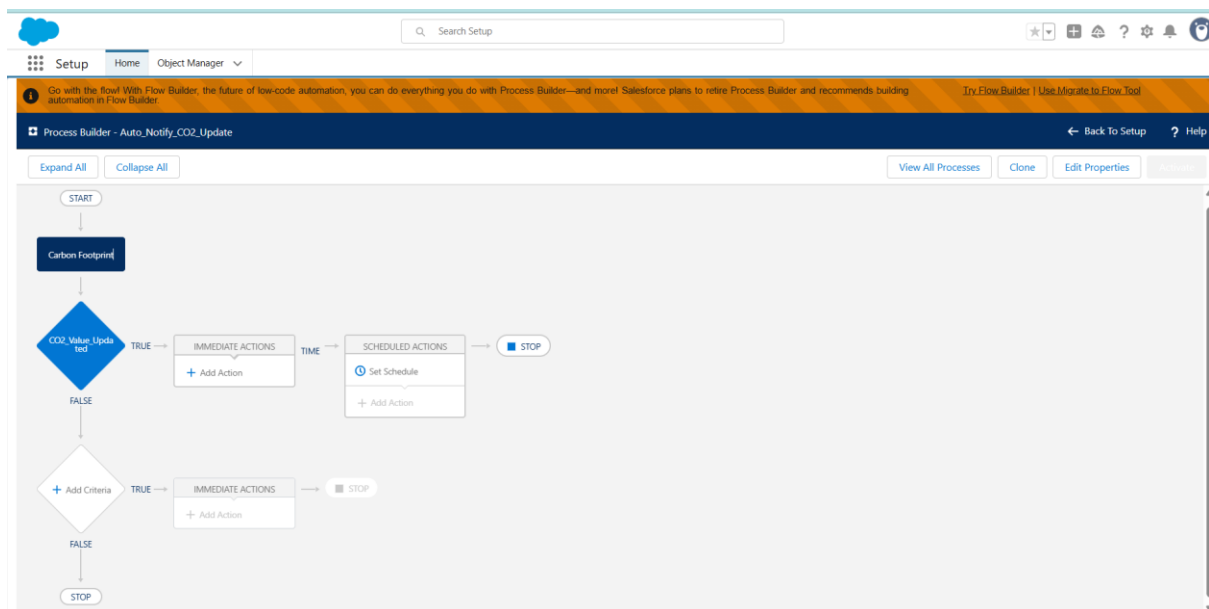
Step 4: Record-Triggered Flow

Flows are more advanced automation.

Example: Update_Total_CO2_Subscription

Create a New Flow

1. Go to **Setup → Flows → New Flow**.

2. Choose **Record-Triggered Flow**.

3. Click **Create**

Configure Trigger

1. **Object:** Carbon_Footprint__c

2. **Trigger:** A record is created or updated

   o Optionally: Run **only if a record is updated to meet condition** (e.g., CO_2 value is not null).

3. **Run the Flow:** After the record is saved (so you can update related subscription records).

Add Flow Logic

1. **Get Records:** (Optional, if you need all CO_2 records for the subscription)

   o Object: Carbon_Footprint__c

   o Filter: Subscription__c = {!$Record.Subscription__c}

   o Store all records → All Records

2. **Assignment:** Calculate total CO_2

   o Loop through all Carbon_Footprint__c records (if using a collection)

   o Sum the CO_2 values → store in a variable varTotalCO2

3. **Update Records:**

   o **Record to Update:** Subscription__c related to the Carbon_Footprint__c record ($Record.Subscription__c)

   o **Field:** Total_CO2__c = {!varTotalCO2}

Save & Activate
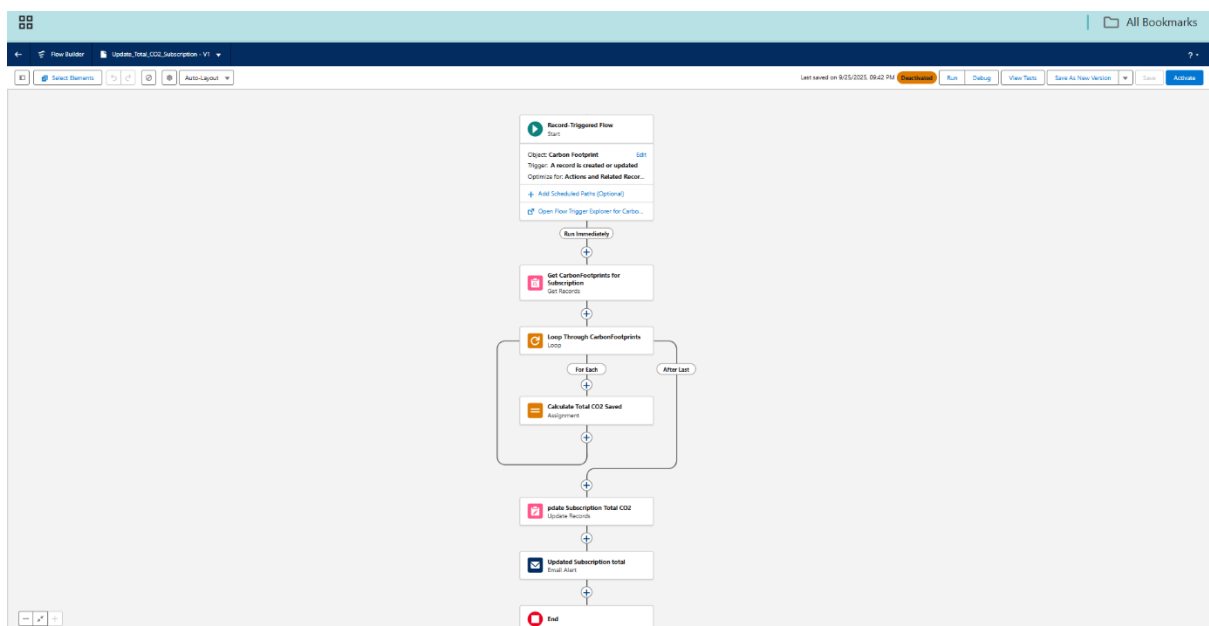
Step 5: Approval Process

Approval Processes let managers approve/reject records.

Create Approval Process

1. Go to Setup → Approval Processes → New Approval Process.

2. Choose Use Standard Setup Wizard.

3. Object: Subscription__c

4. Name: High_Cost_Subscription_Approval

5. Unique Name: High_Cost_Subscription_Approval

6. Description: Approval process for subscriptions with cost greater than 500.