

Project Title – “Smart Subscription Tracker”

Phase 4: Process Automation (Admin)

Validation Rules: EndDate After StartDate

Subscription Validation Rule

[Back to Subscription](#)

Validation Rule Detail

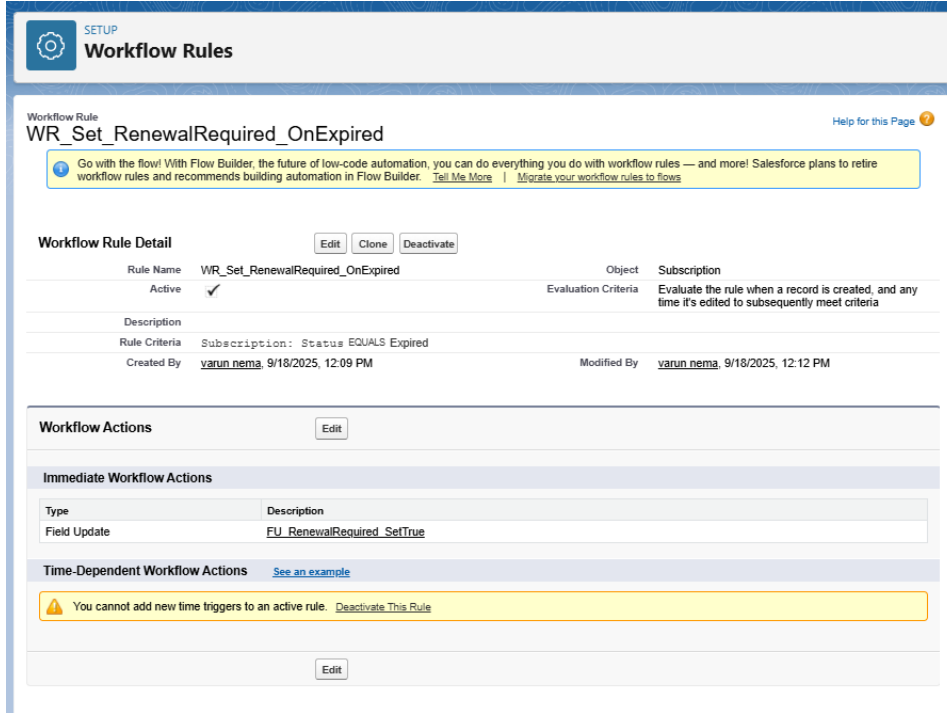
EditClone

Rule Name	EndDate_After_StartDate	Active	✓
Error Condition Formula	AND(NOT(ISBLANK(Start_Date__c)), NOT(ISBLANK(End_Date__c)), End_Date__c <= Start_Date__c)		
Error Message	End Date must be after Start Date.	Error Location	End Date
Description	Prevents saving subscriptions where End Date is on or before Start Date.		
Created By	varun.nema, 9/13/2025, 12:37 PM	Modified By	varun.nema, 9/13/2025, 12:37 PM

EditClone

- 1. **Purpose:** Ensures End Date is always after Start Date in subscriptions.
- 2. **Formula:** Checks both dates are entered and prevents saving if End Date ≤ Start Date.
- 3. **Error Message:** “End Date must be after Start Date.” (shown on End Date field).
- 4. **Impact:** Improves data accuracy, avoids invalid subscription periods, prevents billing/customer issues.
- 5. **Best Practices:** Clear rule name, simple error message, documented description.

WorkFlow rule : WR_Set_RenewalRequired_OnExpired



The screenshot shows the Salesforce Setup page for Workflow Rules. The rule name is 'WR_Set_RenewalRequired_OnExpired' and it is active. The object is 'Subscription'. The rule criteria is 'Subscription: Status EQUALS Expired'. The rule has one immediate action: 'Field Update' with the description 'FU_RenewalRequired_SetTrue'. A warning message states: 'You cannot add new time triggers to an active rule. Deactivate This Rule'.

Workflow Rule Detail

Rule Name	Object
WR_Set_RenewalRequired_OnExpired	Subscription

Workflow Actions

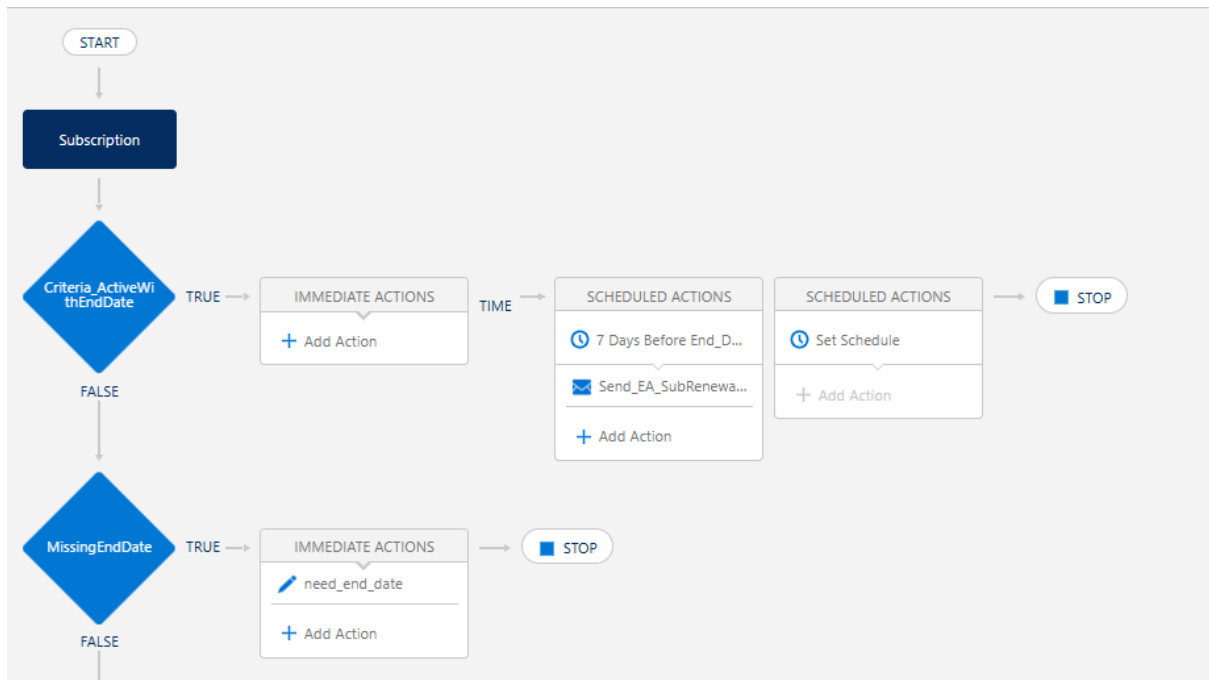
Type	Description
Field Update	FU_RenewalRequired_SetTrue

Time-Dependent Workflow Actions

You cannot add new time triggers to an active rule. [Deactivate This Rule](#)

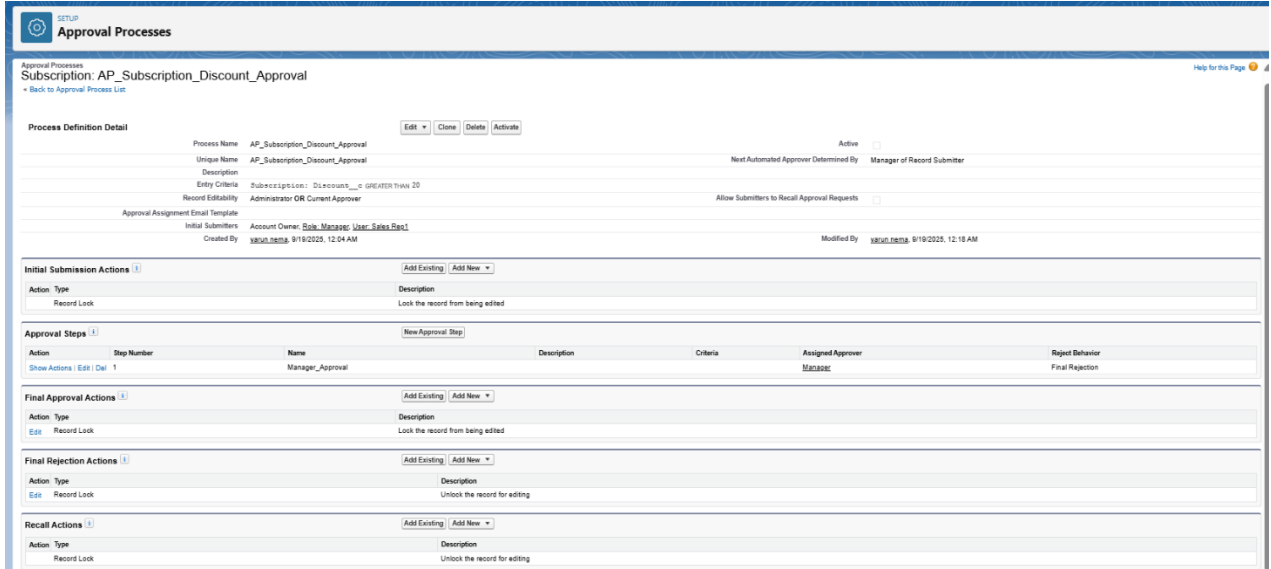
1. **Purpose:** Automates renewal tracking by marking subscriptions as requiring renewal when their status becomes *Expired*.
2. **Rule Criteria:** Subscription: Status = Expired.
3. **Action:** Immediate field update → sets *RenewalRequired* = *True*.
4. **Impact:** Saves manual effort, ensures expired subscriptions are flagged for renewal, supports timely follow-ups.
5. **Best Practices:** Clear rule name, automated update, improves subscription lifecycle management.

Process Builder



1. Purpose: Automates subscription renewal reminders by sending an email 7 days before the End Date.
2. Object: Subscription (Subscription__c), triggered when a record is created or edited.
3. Criteria: Runs when:
4. End_Date__c is not blank,
5. Status__c = Active,
6. Reminder_Sent__c = False.
7. Actions:
8. Scheduled (7 days before End Date): Send email alert *EA_SubRenewal_7d*.
9. (Optional) Update record → set Reminder_Sent__c = True.
10. Impact: Ensures timely renewal reminders, reduces missed renewals, and improves customer retention.

Approval Process: AP Subscription Discount Approval



The screenshot shows the 'Approval Processes' setup page for a process named 'AP_Subscription_Discount_Approval'. The page includes a 'Process Definition Detail' section with fields for Process Name, Unique Name, Description, Entry Criteria, Record Editability, Approval Assignment Email Template, Initial Submitters, and Created/Modified dates. Below this are sections for 'Initial Submission Actions', 'Approval Steps', 'Final Approval Actions', 'Final Rejection Actions', and 'Recall Actions', each with a table for defining actions and their descriptions.

Process Definition Detail

Field	Value
Process Name	AP_Subscription_Discount_Approval
Unique Name	AP_Subscription_Discount_Approval
Description	Subscription: Discount__c GREATER than 20
Entry Criteria	Subscription: Discount__c GREATER than 20
Record Editability	Administrator OR Current Approver
Approval Assignment Email Template	
Initial Submitters	Account Owner: Sales Manager; User: Sales Reps
Created By	9/18/2025, 12:04 AM
Modified By	9/18/2025, 12:18 AM

Initial Submission Actions

Action	Type	Description
Record Lock		Lock the record from being edited

Approval Steps

Action	Step Number	Name	Description	Criteria	Assigned Approver	Reject Behavior
Manager Approval	1	Manager Approval			Manager	Final Rejection

Final Approval Actions

Action	Type	Description
Record Lock		Lock the record from being edited

Final Rejection Actions

Action	Type	Description
Record Lock		Unlock the record for editing

Recall Actions

Action	Type	Description
Record Lock		Unlock the record for editing

1. Objective: To control subscription discounts exceeding 20% by requiring manager approval, ensuring compliance with pricing policies.
2. Entry Criteria: Discount__c > 20.
3. Initial Submitters: Account Owner, Managers, Sales Reps.
4. Initial Submission Action:
 - a. Record is locked to prevent edits during review.
 - b. (Optional) Status set to *Pending Approval* and email sent to approver.
5. Approval Steps:
 - a. Step 1: Manager approval is required.
 - b. If approved → moves to *Final Approval Actions*.
 - c. If rejected → moves to *Final Rejection Actions*.
6. Final Approval Actions:
 - a. Status updated to *Approved*.
 - b. Record remains locked (or unlocked if configured).
 - c. Notification can be sent to submitter.
7. Final Rejection Actions:
 - a. Status updated to *Rejected*.
 - b. Record unlocked for further editing if needed.
 - c. Recall Actions:

- d. If recalled, record is unlocked and editable again.
- e. Impact:
- f. Prevents unauthorized or excessive discounts.
- g. Automates approval routing to managers.
- h. Enhances compliance, accountability, and audit tracking through Approval History.

Flow builder

I have implemented Flow Builder to configure, Custom Notifications, and Task automation. The respective screenshots have been included for reference.

Email alerts

Email Template

SubRenewal_Reminder_7d

EditCloneDelete

Details

Related

Information

Email Template NameSubRenewal_Reminder_7d

Description

Made in Email Template Builder☐

Related Entity TypeSubscription

FolderPublic Email Templates

Message Content

SubjectEnhanced Letterhead

HTML Value

Subject: Reminder: Subscription {{{Subscription_c__c.Name}}} expires on {{{Subscription_c__c.End_Date__c}}}

Hi ,

Your subscription {{{Subscription_c__c.Name}}} for {{{Subscription_c__c.Account__c}}} expires on {{{Subscription_c__c.End_Date__c}}}

Please review and renew: [Open Subscription]({{{Subscription_c__c.Record_Link_c__c}}})

Thanks,

Sales Team

Additional Information

Created Byvarun nema, 9/17/2025, 12:06 PM

Last Modified Byvarun nema, 9/17/2025, 12:06 PM

All Email Alerts

[Help for this Page](#) ?

Email alerts are used to send emails from a flow or other automation.

View: All Email Alerts [Create New View](#)

A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | Other **All**

New Email Alert				
Action	Description ↑	Email Template Name	Object	Last Modified Date
Edit Del	expiration	Expiration Notice	Subscription	9/18/2025
Edit Del	Payment receipt	Expiration Notice	Subscription	9/18/2025
Edit Del	SubRenewal 7d	SubRenewal Reminder 7d	Subscription	9/18/2025

A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | Other **All**

1. Email Alert Configuration

To automate renewal reminders, an **Email Alert** was created in Salesforce. This reusable action links an email template to recipients and can be triggered by Flows or other automation tools.

- **Alert Name:** EA_SubRenewal_7d
- **Object:** Subscription__c
- **Email Template:** SubRenewal_Reminder_7d (Lightning Email Template)
- **Sender Address:** Org-Wide Email Address (ensures consistent communication)
- **Recipients:** Subscription Owner and related Primary Contact (custom field).
- **Purpose:** Ensures renewal reminders are sent in a standardized, timely manner.

2. Integration with Record-Triggered Flow

A **Record-Triggered Flow** was built to send reminders seven days before the subscription End Date.

- **Trigger:** Subscription__c created or updated.

- **Entry Conditions:** End_Date__c is not null and Status__c = 'Active'.
- **Scheduled Path:** Executes 7 days before End_Date__c.
- **Action:** Calls the Email Alert (EA_SubRenewal_7d), passing the Subscription record Id.
- **Duplicate Prevention:** Reminder_Sent__c flag is updated to TRUE after sending.

This ensures that each active subscription automatically generates a renewal reminder exactly one week before expiry.

3. Integration with Scheduled-Triggered Flow

A **Scheduled-Triggered Flow** was also developed to handle bulk processing of reminders.

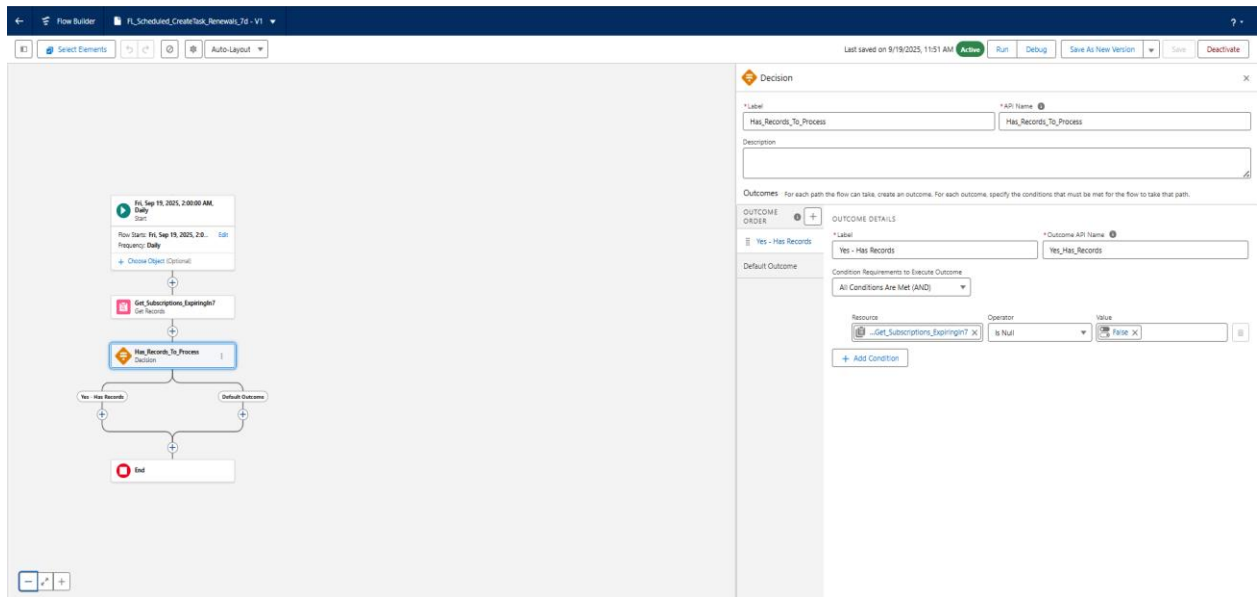
- **Schedule:** Runs daily at a fixed time.
- **Query Logic:** Retrieves all Subscription__c records where:
 - End_Date__c = Today + 7
 - Status__c = Active
 - Reminder_Sent__c = FALSE
- **Loop Logic:** For each qualifying record:
 - Sends the Email Alert EA_SubRenewal_7d.
 - Updates Reminder_Sent__c = TRUE to avoid duplicates.
- **Bulk Update:** Records are updated in a collection for efficiency.

This approach complements the record-triggered flow by ensuring no subscription is missed, even if End Dates were updated historically.

Field Updates

Field Updates were not used because they are **deprecated**, have **limited flexibility** (can only set simple values), and are being replaced by **Flows**, which provide more advanced, scalable, and future-proof automation.

Tasks



To automate reminders for upcoming subscription renewals, I implemented a **Scheduled Flow** on the Subscription__c object. This flow automatically creates **Tasks** for subscription owners 7 days before the subscription end date and ensures duplicate reminders are not sent.

Pre-Setup

Before building the flow, the following fields were confirmed/created on Subscription__c:

- **End_Date__c** (Date) – Subscription expiry date.
- **OwnerId** (standard owner field).
- **Primary_Contact__c** (Lookup to Contact) – optional but useful for linking.
- **Reminder_Sent__c** (Checkbox, default = False) – prevents duplicate reminders.
- **Record_Link__c** (Formula field) – optional hyperlink to the record for task description.

Flow Configuration Steps

1. Create Scheduled Flow

- Setup → Flows → New Flow → Scheduled-Triggered Flow.

- Frequency: **Daily** | Start Time: **2:00 AM**.

2. Get Records

- Retrieve Subscription__c records where:
 - End_Date__c = Today + 7
 - Status__c = Active
 - Reminder_Sent__c = False
- Store all records in a collection.

3. Decision Element

- Check if any records were retrieved.
- If none → end flow.
- If yes → proceed to loop.

4. Loop Through Subscriptions

- Iterate through each subscription record in the collection.

5. Create Task (inside loop)

- **Subject:** Call customer about subscription renewal.
- **OwnerId:** Subscription Owner.
- **WhatId:** Subscription record Id.
- **Whold:** Primary Contact (if available).
- **Due Date:** Today (7 days before expiry).
- **Status:** Not Started.
- **Priority:** High.
- **Description:** Includes Subscription Name, End Date, and Record Link.

6. Update Subscription (inside loop)

- Update field Reminder_Sent__c = True to prevent duplicate reminders.

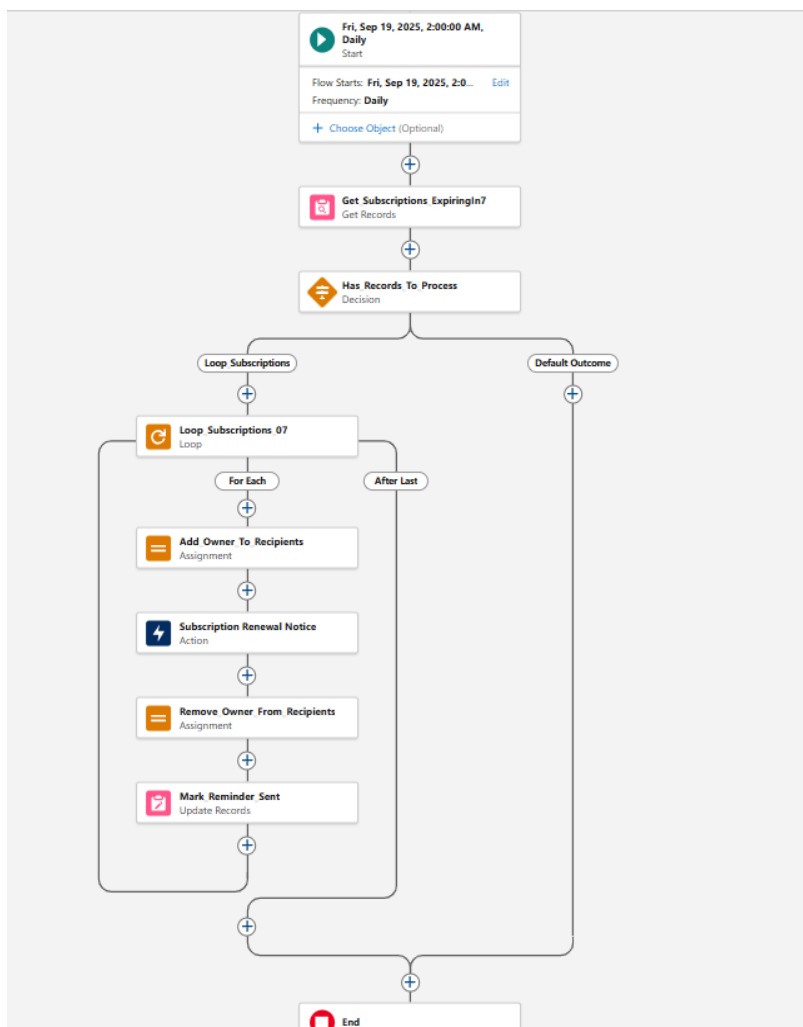
7. End Loop & Activate Flow

- Save as FL_Scheduled_CreateTask_Renewals_7d.
- Activate flow.

Outcome

This flow ensures that subscription owners receive a **high-priority Task reminder** exactly 7 days before the subscription expires, improving renewal management and reducing missed follow-ups.

Custom notification



To enhance subscription management, I created a **Custom Notification Type** and a **Scheduled Flow** to automatically notify subscription owners 7 days before a subscription's End Date. This ensures proactive follow-up and improves renewal efficiency.

Step 1: Create a Custom Notification Type

- **Navigation:** Setup → Notification Builder → Notification Types → New.
- **Details:**
 - Label: *Subscription Renewal Notice*
 - API Name: *Subscription_Renewal_Notice*
 - Supported Channels: Desktop, Mobile, and In-App (bell icon).
 - Description: Notification for upcoming subscription renewals.
- **Result:** Notification type is now available for use in Flows and Approval Processes.

Step 2: Build the Scheduled Flow

1. Flow Setup

- Setup → Flows → New Flow → *Scheduled-Triggered Flow*.
- Frequency: **Daily** | Start Time: **2:00 AM**.

2. Get Subscriptions to Notify

- Object: *Subscription__c*.
- Criteria:
 - End_Date__c = Today + 7
 - Status__c = Active
 - Reminder_Sent__c = False
- Store all matching records in a collection.

3. Decision Check

- If records exist → continue.
- If none → end flow.

4. Loop Through Records

- Iterate through each subscription in the collection.

5. Recipient Collection Setup

- Create a **Text Collection Variable** (RecipientIDs).
- Add Subscription Owner (OwnerId) into this collection.

6. Send Custom Notification (inside loop)

- **Notification Type:** Subscription Renewal Notice.
- **Title:** *Subscription expiring in 7 days: {!Subscription__c.Name}*
- **Body:** *Ends on {!Subscription__c.End_Date__c}. Open subscription for details.*
- **Target Id:** Subscription record Id.
- **Recipient Ids:** RecipientIDs collection.

7. Cleanup Recipients

- Remove the owner from RecipientIDs after each loop to avoid carry-over.

8. Update Subscription

- Set Reminder_Sent__c = True for each processed subscription to prevent duplicates.

9. Activate Flow

- Saved as **FL_Sched_CustomNotification_Renewal_7d** and activated.

Outcome

This automation sends a **real-time Salesforce in-app and mobile notification** to subscription owners 7 days before renewal, helping them take timely action. The system also updates each subscription to avoid duplicate notifications.
