Phase 4: Process Automation (Admin)

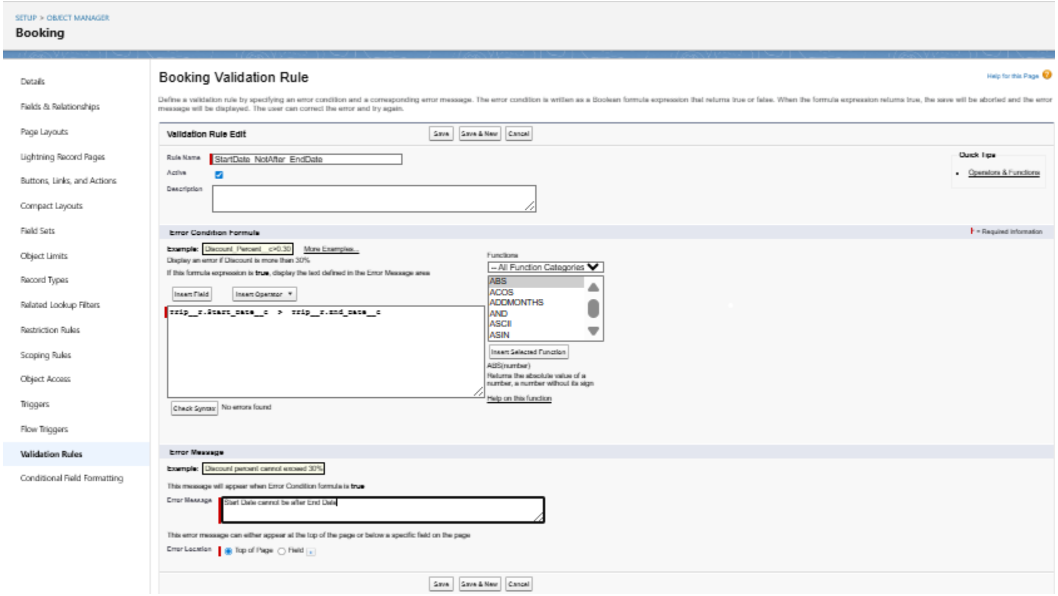
1. Validation Rules:

Use validation rules to stop bad data.

Path: Setup → Object Manager → Booking → Validation Rules → New

Rule 1 — Start date cannot be after End date

- Rule Name: StartDate_NotAfter_EndDate
- Formula: Trip__r.Start_Date__c > Trip__r.End_Date__c
- Error Message: Start Date cannot be after End Date.
- Error Location: Top of Page (or field Start_Date__c) → Save.



Booking Validation Rule

Define a validation rule by specifying an error condition and a corresponding error message. The error condition is written as a Boolean formula expression that returns true or false. When the formula expression returns true, the save will be aborted and the error message will be displayed. The user can correct the error and try again.

Validation Rule Edit

Rule Name: StartDate_NotAfter_EndDate

Active: ☑

Description:

Error Condition Formula

Example: Discount_Percent__c>0.30

Display an error if Discount is more than 30%

If this formula expression is **true**, display the text defined in the Error Message area

```
trip__r.Start_Date__c  >  trip__r.End_Date__c
```

No errors found

ABS(number)

Returns the absolute value of a number, a number without its sign

Error Message

Example: Discount percent cannot exceed 30%

This message will appear when Error Condition formula is **true**

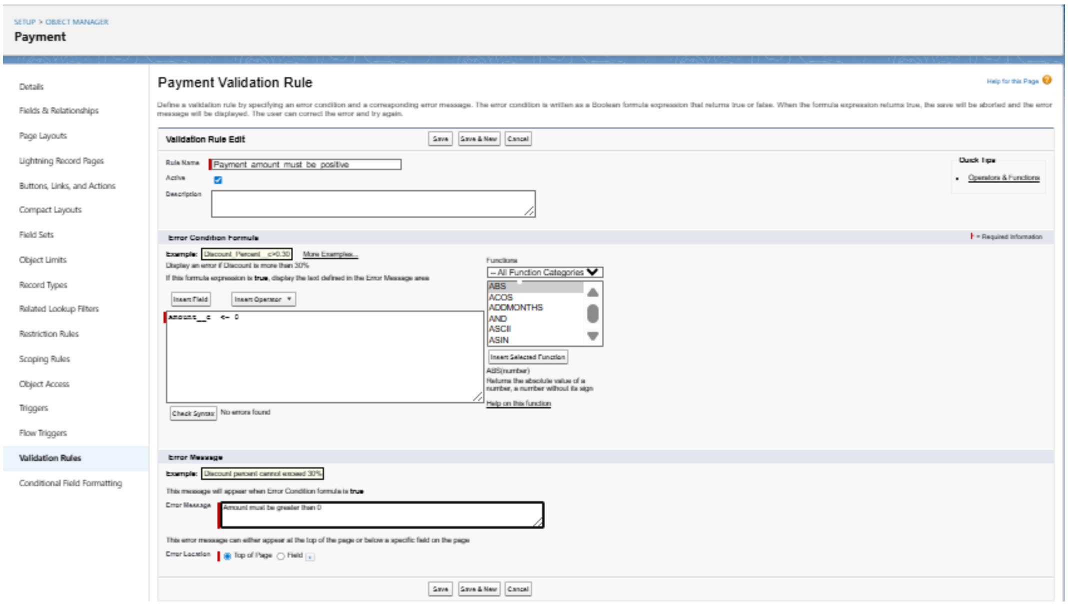
Error Message: Start Date cannot be after End Date

This error message can either appear at the top of the page or below a specific field on the page

Error Location: ● Top of Page ○ Field

Rule 2 — Payment amount must be positive (on Payment object)

- Path: Payment → Validation Rules → New
- Formula: Amount__c <= 0
- Error Message: Amount must be greater than 0. → Save.



Payment Validation Rule

Define a validation rule by specifying an error condition and a corresponding error message. The error condition is written as a Boolean formula expression that returns true or false. When the formula expression returns true, the save will be aborted and the error message will be displayed. The user can correct the error and try again.

Validation Rule Edit

Rule Name: Payment_amount_must_be_positive

Active: ☑

Description:

Error Condition Formula

| = Required Information

Example: Discount_Percent__c>0.30

Display an error if Discount is more than 30%

If this formula expression is **true**, display the text defined in the Error Message area

```
amount__c <= 0
```

No errors found

ABS(number)
Returns the absolute value of a number, a number without its sign

Error Message

Example: Discount percent cannot exceed 30%

This message will appear when Error Condition formula is **true**

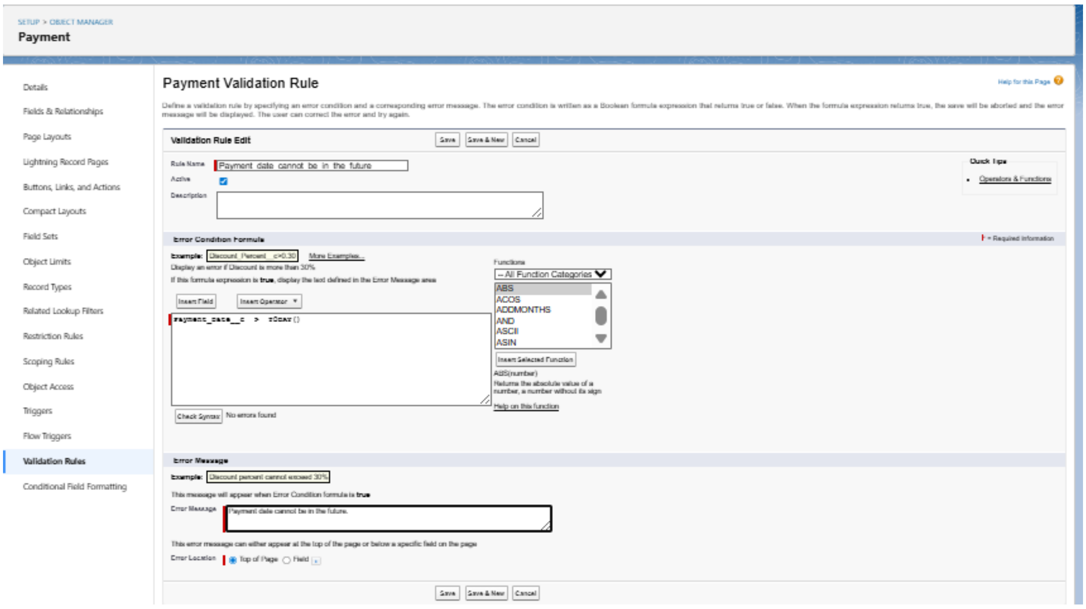
Error Message: Amount must be greater than 0

This error message can either appear at the top of the page or below a specific field on the page

Error Location: ● Top of Page ○ Field

Rule 3 — Payment date cannot be in the future

- On Payment → Validation Rules → New
- Formula: Payment_Date__c > TODAY ()
- Error: Payment date cannot be in the future. → Save.



Payment Validation Rule

Define a validation rule by specifying an error condition and a corresponding error message. The error condition is written as a Boolean formula expression that returns true or false. When the formula expression returns true, the save will be aborted and the error message will be displayed. The user can correct the error and try again.

Validation Rule Edit

Rule Name: Payment_date_cannot_be_in_the_future

Active: ☑

Description:

Error Condition Formula

| = Required Information

Example: Discount_Percent__c>0.30

Display an error if Discount is more than 30%

If this formula expression is **true**, display the text defined in the Error Message area

```
payment_date__c > TODAY()
```

No errors found

ABS(number)
Returns the absolute value of a number, a number without its sign

Error Message

Example: Discount percent cannot exceed 30%

This message will appear when Error Condition formula is **true**

Error Message: Payment date cannot be in the future.

This error message can either appear at the top of the page or below a specific field on the page

Error Location: ● Top of Page ○ Field

2. Flow Builder

1) Record-Triggered Flow — Auto-confirm Booking when Payment marked paid.

Setup → Flows → New Flow → choose **Record-Triggered Flow → Create**.

Object: Payment → Trigger the Flow **When a record is updated**.

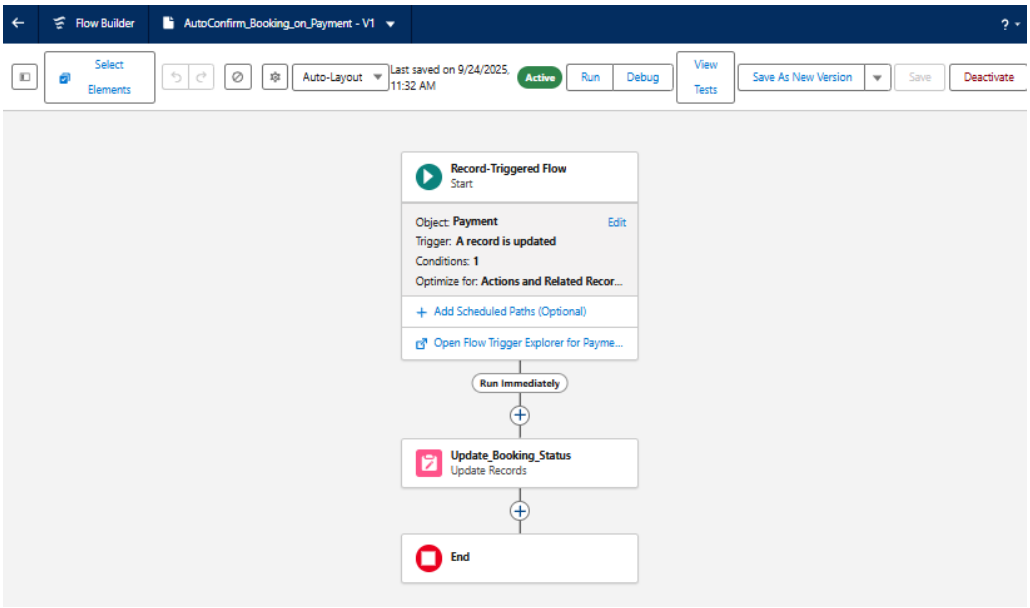
Condition Requirements:

- Entry Condition: Payment_Status__c Equals Paid

Add Element → Update Records

- Drag an **Update Records** element into the Flow.
- Label it: Update_Booking_Status.
- **How to Find Records to Update** → choose:
  "Specify conditions to identify records, and set fields individually".
- In **Object**: choose **Booking**.
- In **Filter Conditions**:
- Field: **Id**
- Operator: Equals
- Value: $Record.Booking__c
  (This means: update the Booking that is related to this Payment.)
- In **Set Field Values for the Booking Record**:
- Field = Status__c
- Value = Confirmed.

Save name: AutoConfirm_Booking_on_Payment → **Activate**.



Object: Payment

Trigger: A record is updated

Conditions: 1

Optimize for: Actions and Related Recor...

2) Scheduled-Triggered Flow:

1. Setup -> Quick Find ->Flows

1. Click New Flow
2. Choose Schedule-Triggered Flow -> Create

2. Configure the schedule

In the Start element (opened automatically):

- Start Date: pick Today (or a date in the past so it runs daily from now)
- Start Time: e.g., 01:00 AM
- Frequency: Daily
- Click Done

3. Create a Date formula resource for “2 days from today”

1. In Flow Builder → Manager tab -> New Resource
2. Resource Type: Formula
3. API Name: varDatePlus2
4. Data Type: Date
5. Formula: $Flow.CurrentDate + 2
6. Save.

4. Create collection & temp record variables (for bulk Task creation)

1. New Resource -> Resource Type: Variable
    - API Name: varTasksToCreate
    - Data Type: Record
    - Object: Task
    - Check Allow multiple values (collection), Save
2. New Resource -> Resource Type: Variable
    - API Name: varTempTask
    - Data Type: Record
    - Object: Task
    - Do NOT allow multiple values (single record), Save

5. Get matching bookings

1. Get Records
2. Label: Get Bookings in 2 Days
3. Object: Booking__c
4. Filter Conditions:
    - Travel_Date__c Equals varDatePlus2 (select the formula resource you created)
    - Status__c Equals Confirmed
5. How Many Records to Store: All records
6. How to Store Record Data: Automatically store all fields
7. Done.

6. Add a Loop over the bookings

1. Loop
2. Label: Loop — Bookings
3. Collection Variable: the output from "Get Bookings in 2 Days" (e.g. {! Get_Bookings_in_2_Days})
4. Direction: First to Last, Done

7. Add Outcome.

1. Label: Confirmed
2. Condition:
    - Resource → {!Loop.CurrentItem.Status__c}
      (choose it from the resource picker inside the loop)
    - Operator → Equals
    - Value → type manually:
    - Confirmed

8. Default Outcome

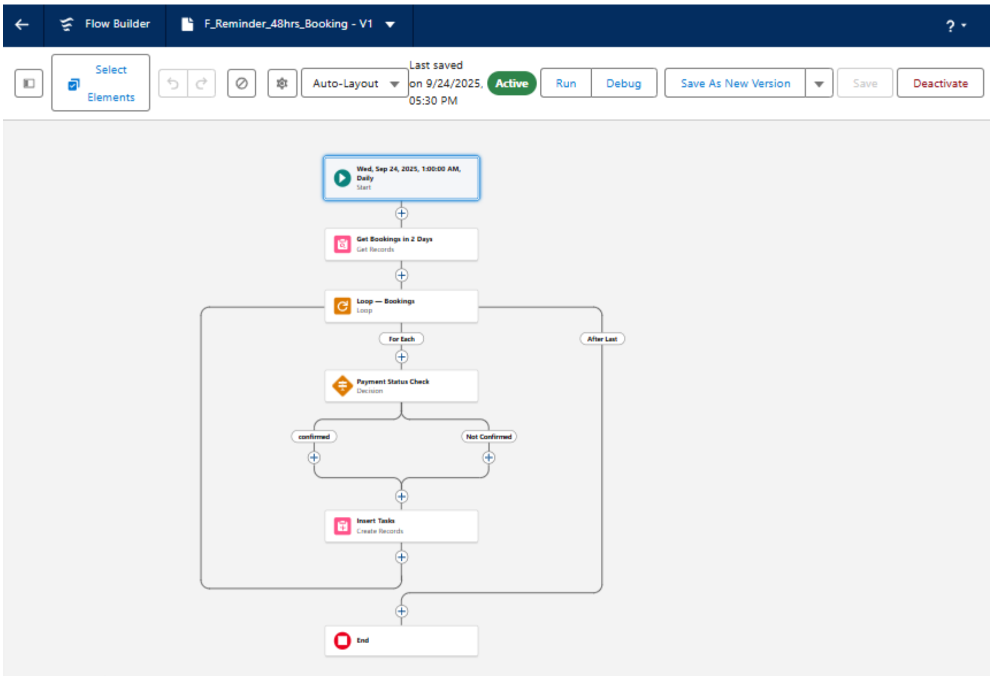
- Leave the Default Outcome as is.
- Rename it → Not Confirmed
- This will cover Enquiry, Completed, Canceled.

9. After the Loop — bulk insert all Tasks

1. Create Records
2. Label: Insert Tasks
3. How Many Records to Create: Use separate resources, and literal values? → Use collection variable
    - Select the collection: varTasksToCreate
4. Done.

10. Save, Activate

1. Click Save — give it a name: F_Reminder_48hrs_Booking and API Name F_Reminder_48hrs_Booking
2. click Activate



3. Email Alerts:

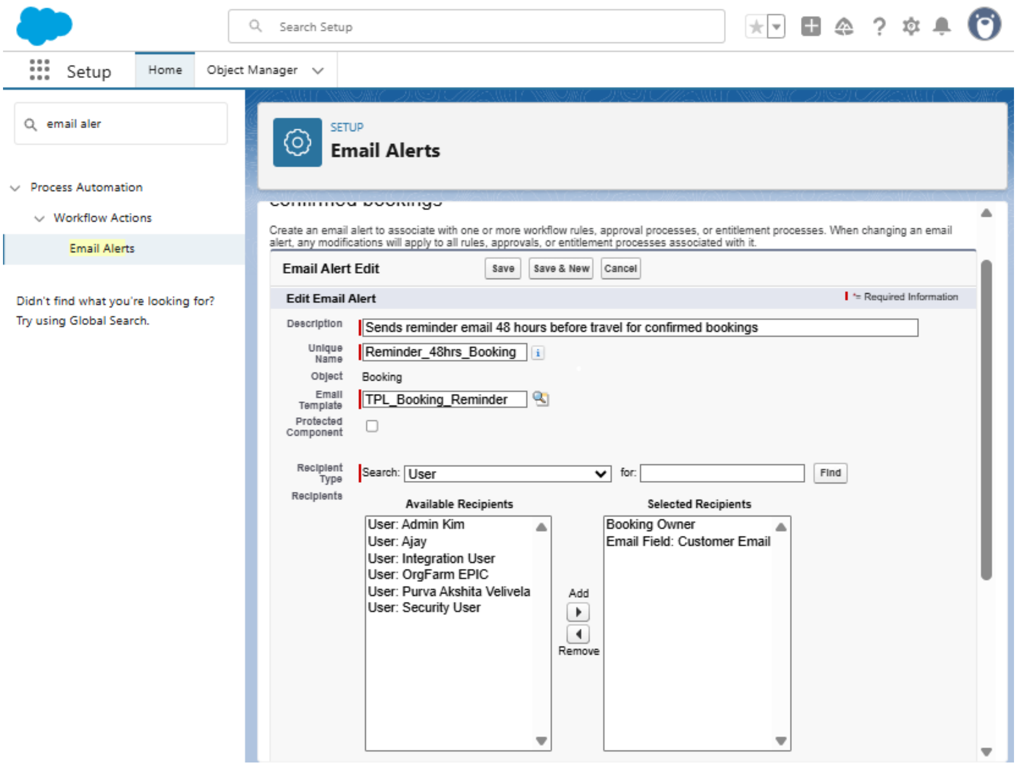
1. Setup ->Quick Find ->Email Alerts
2. Click New Email Alert

- Description: Sends reminder email 48 hours before travel for confirmed bookings
- Unique Name: Reminder_48hrs_Booking (auto-filled)
- Object: Booking__c
- Email Template: select TPL_Booking_Reminder (or your template name)
- From Email Address: choose the Org-Wide Address if you created one, otherwise leave default (the running user)

Select Recipients

- Click Email Field (or expand it) ▶ pick Booking__c.Customer_Email__c (or whatever your API name is) ▶ click Add.

- Notify the Booking Owner → Add.
- Click Save.



Create an email alert to associate with one or more workflow rules, approval processes, or entitlement processes. When changing an email alert, any modifications will apply to all rules, approvals, or entitlement processes associated with it.

Email Alert Edit

Edit Email Alert

| = Required Information

Description	Sends reminder email 48 hours before travel for confirmed bookings
Unique Name	Reminder_48hrs_Booking
Object	Booking
Email Template	TPL_Booking_Reminder
Protected Component	□
Recipient Type	Search: User for:

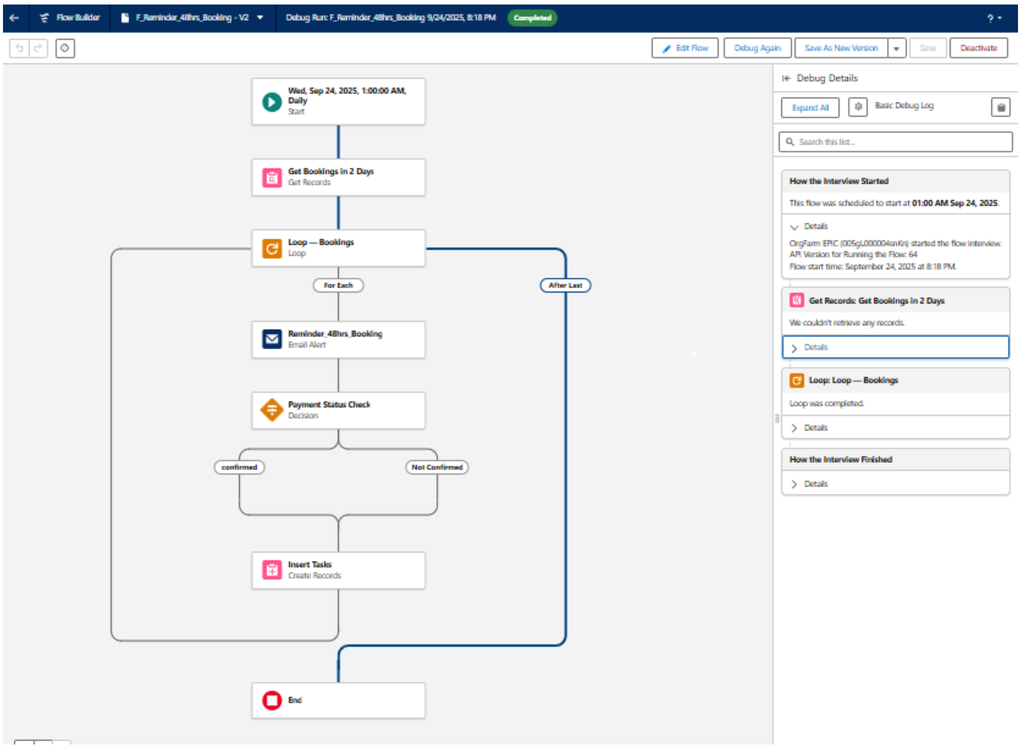
Recipients

Available Recipients	Selected Recipients
User: Admin Kim	Booking Owner
User: Ajay	Email Field: Customer Email
User: Integration User	
User: OrgFarm EPIC	
User: Purva Akshita Velivela	
User: Security User	

Use the Email Alert inside your Flow (exact clicks)

1. Open Setup -> Flows -> open your Scheduled Flow -> Edit.
2. On the canvas, inside the Loop where you iterate bookings, click the + where you want to send the email.
3. In the Elements panel select Action (or click the + and choose **Action**).
4. In the Action search box type the name of your email alert: Reminder_48hrs_Booking.
5. Select the action that represents your Email Alert (label will match the Email Alert). Click it.Configure the action inputs
6. Set the Loop.CurrentItem.Id
7. Click Done.

9. Save the flow, Debug, Activate.



How the Interview Started

This flow was scheduled to start at **01:00 AM Sep 24, 2025**.

Details

OrgFarm EPIC (005gL000004snKn) started the flow interview.
API Version for Running the Flow: 64
Flow start time: September 24, 2025 at 8:18 PM.

Get Records: Get Bookings in 2 Days

We couldn't retrieve any records.

Details

Loop: Loop — Bookings

Loop was completed.

Details

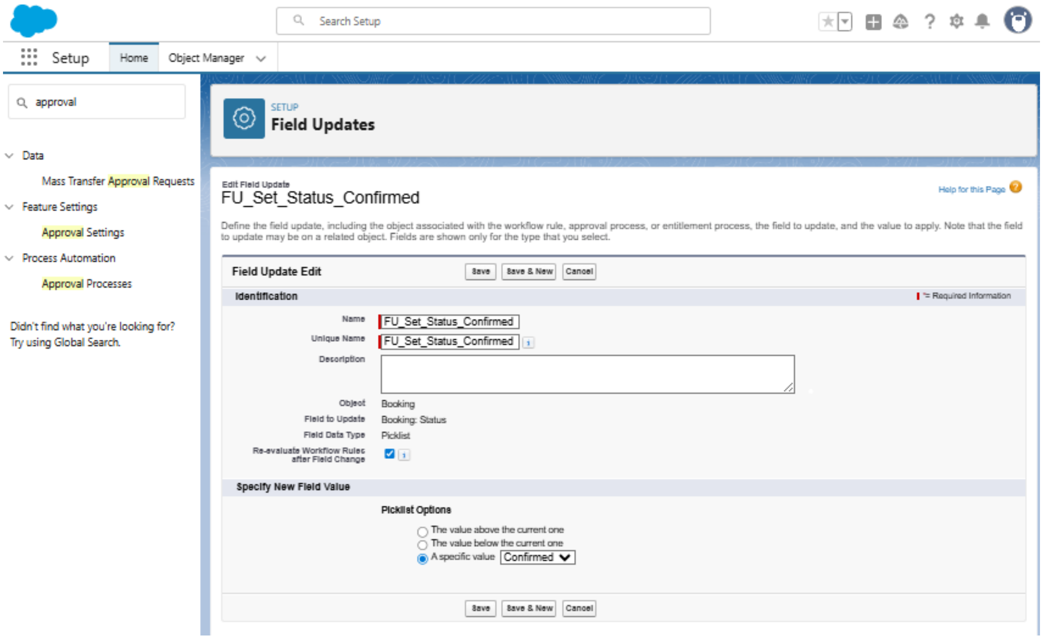
How the Interview Finished

Details

4. Field Updates:

When Booking is confirmed, set Booking.Confirmed_Date__c = Today and Booking.Status__c = 'Confirmed'.

- In your Record-Triggered Flow (After Save) add Update Records element
  - Update the triggering record: use {!$Record.Id}
  - Set fields Confirmed_Date__c = $Flow.CurrentDate and Status__c = 'Confirmed'.



Edit Field Update

FU_Set_Status_Confirmed

Define the field update, including the object associated with the workflow rule, approval process, or entitlement process, the field to update, and the value to apply. Note that the field to update may be on a related object. Fields are shown only for the type that you select.

Field Update Edit

Identification | = Required Information

Name	FU_Set_Status_Confirmed
Unique Name	FU_Set_Status_Confirmed
Description	
Object	Booking
Field to Update	Booking: Status
Field Data Type	Picklist
Re-evaluate Workflow Rules after Field Change	☑

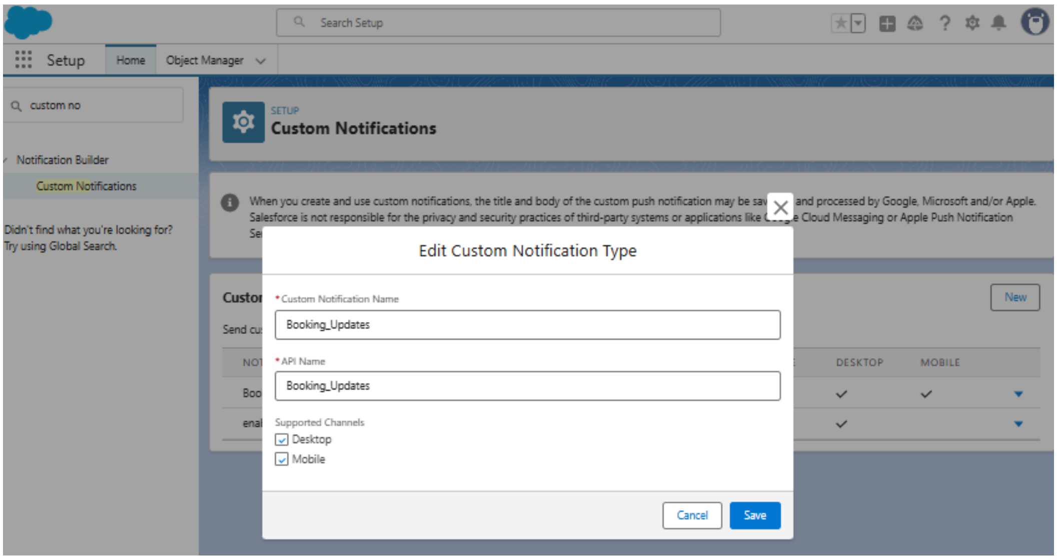
Specify New Field Value

Picklist Options

○ The value above the current one
○ The value below the current one
● A specific value Confirmed

5. Custom Notifications:

- Setup -> Quick Find -> Notification Builder -> Custom Notifications
- Click New
- Enter Name: Booking_Updates and API Name Booking_Updates
- Choose supported channels: Desktop, Mobile.
- Save



When you create and use custom notifications, the title and body of the custom push notification may be sav and processed by Google, Microsoft and/or Apple. Salesforce is not responsible for the privacy and security practices of third-party systems or applications like e Cloud Messaging or Apple Push Notification

Edit Custom Notification Type

* Custom Notification Name: Booking_Updates

* API Name: Booking_Updates

Supported Channels

☑ Desktop

☑ Mobile

Cancel | Save