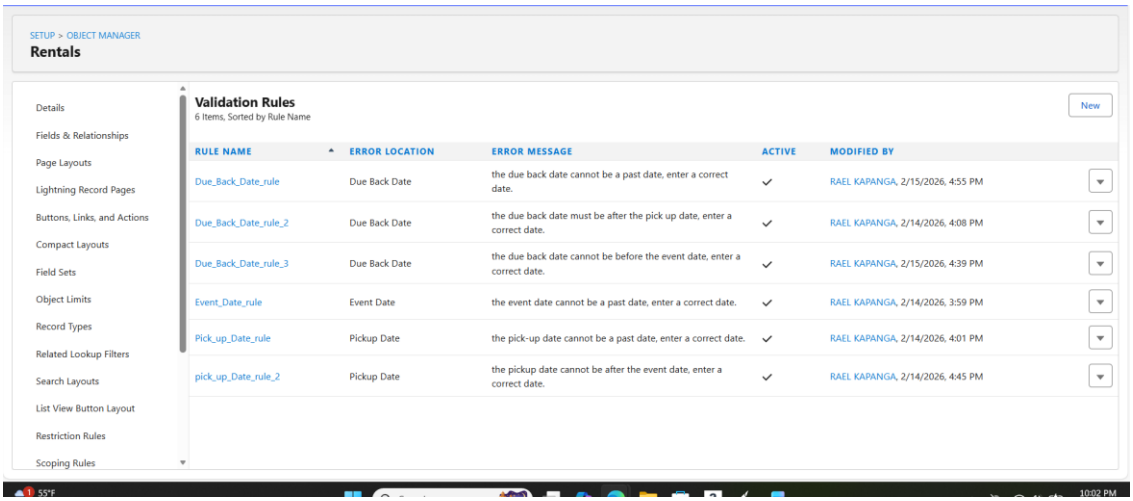
PHASE 4) Validation Rules Implementation

Implement Validation Rules on the Rental object to ensure data accuracy and prevent incomplete or incorrect record creation.



- Due back date rule 1

Rentals Validation Rule

Back to Rentals

Validation Rule Detail Edit Clone

Rule Name	Due_Back_Date_rule	Active	✓
Error Condition Formula	Due_Back_Date__c < TODAY()		
Error Message	the due back date cannot be a past date, enter a correct date.	Error Location	Due Back Date
Description	the due back date cannot be a past date		
Created By	RAEL KAPANGA, 2/14/2026, 4:03 PM	Modified By	RAEL KAPANGA, 2/14/2026, 4:03 PM

Edit Clone

- Due back date rule 2

Rentals Validation Rule

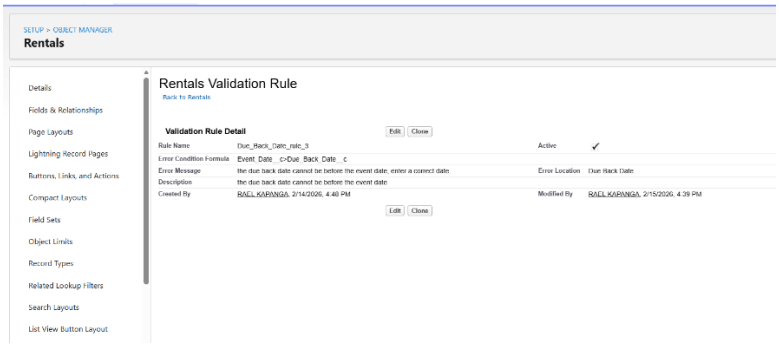
Back to Rentals

Validation Rule Detail Edit Clone

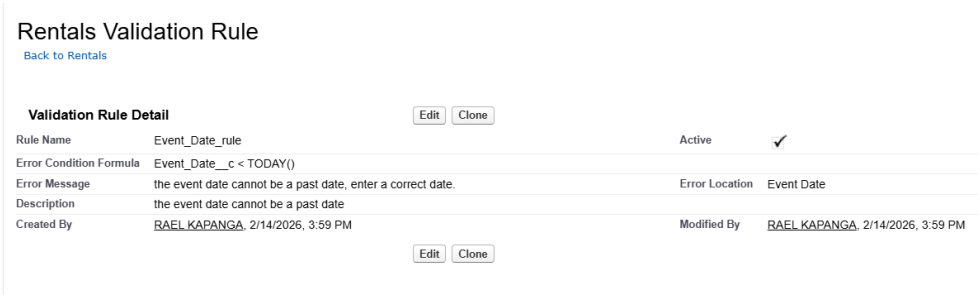
Rule Name	Due_Back_Date_rule_2	Active	✓
Error Condition Formula	Due_Back_Date__c <= Pickup_Date__c		
Error Message	the due back date must be after the pick up date, enter a correct date.	Error Location	Due Back Date
Description	the due back date must be after the pick up date		
Created By	RAEL KAPANGA, 2/14/2026, 4:08 PM	Modified By	RAEL KAPANGA, 2/14/2026, 4:08 PM

Edit Clone

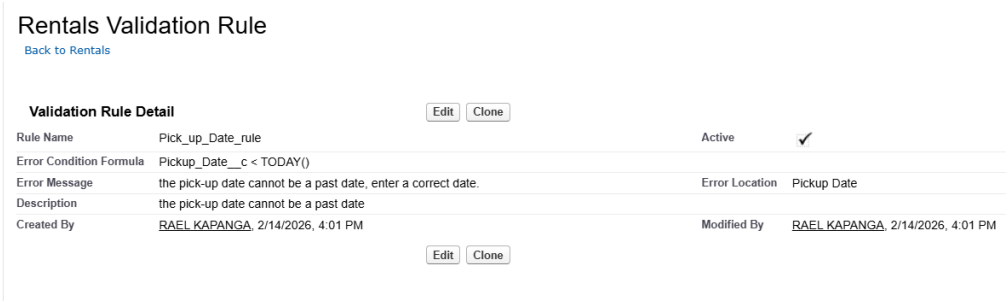
- Due back date rule 3



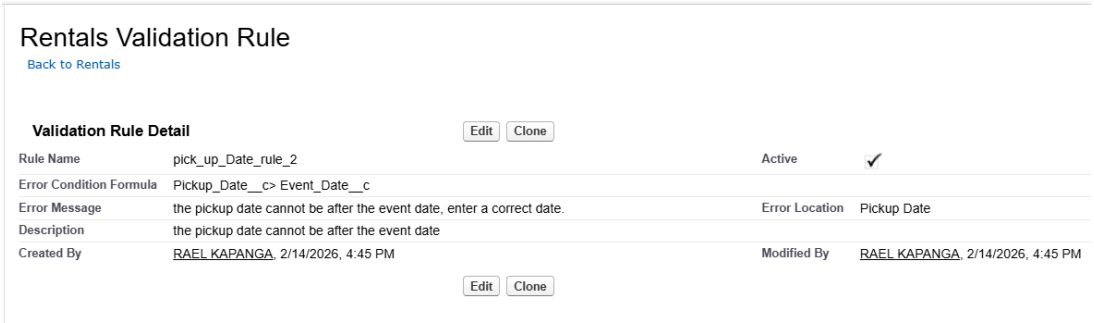
- Event date rule



- Pick up date rule 1



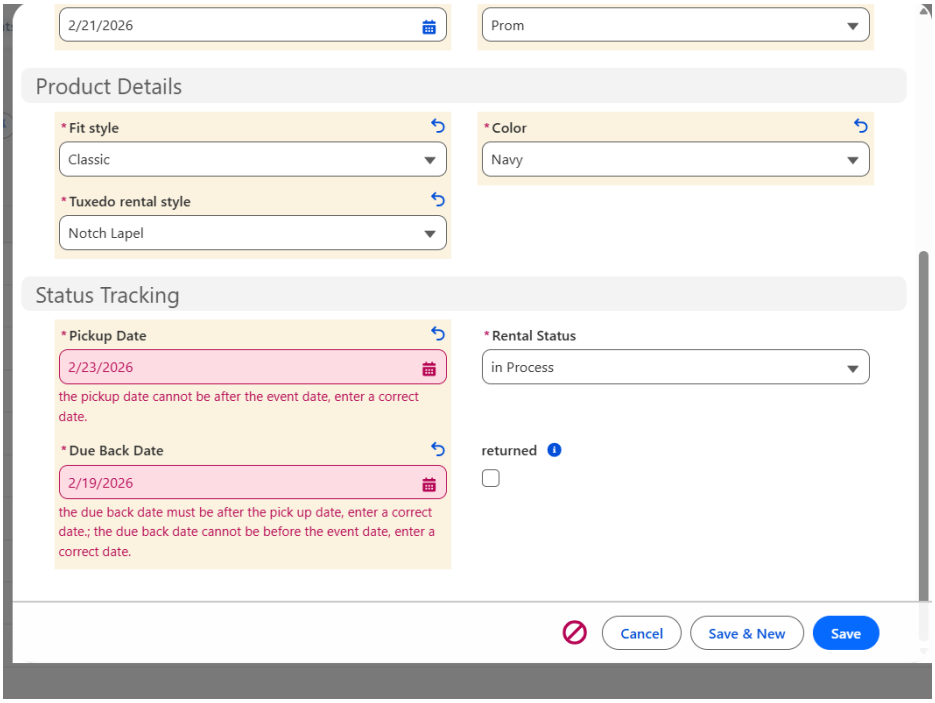
- Pick up date rule 2

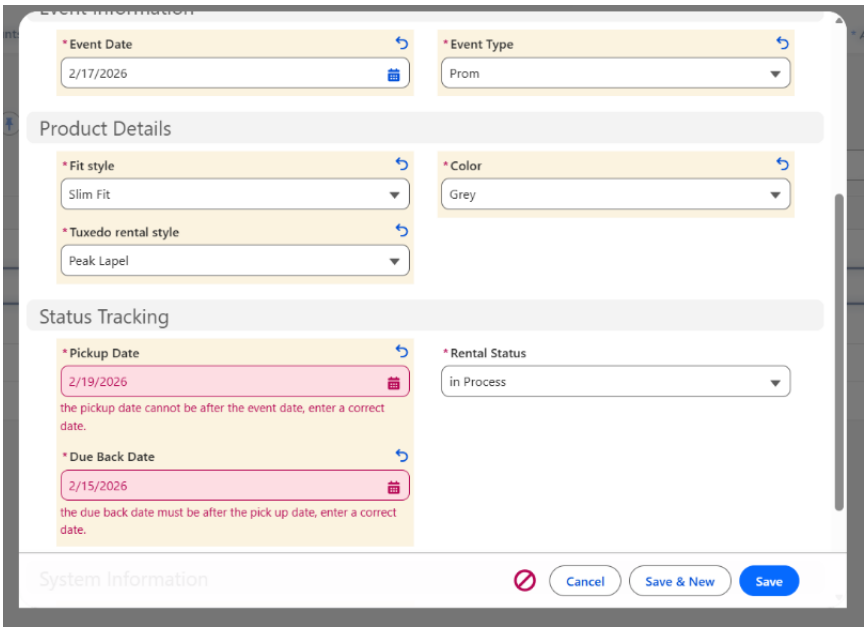


Testing & Verification

Each validation rule was tested by:

- Attempting to create invalid records
- Confirming error messages display correctly
- Verifying that valid records save successfully





Functional Outcome

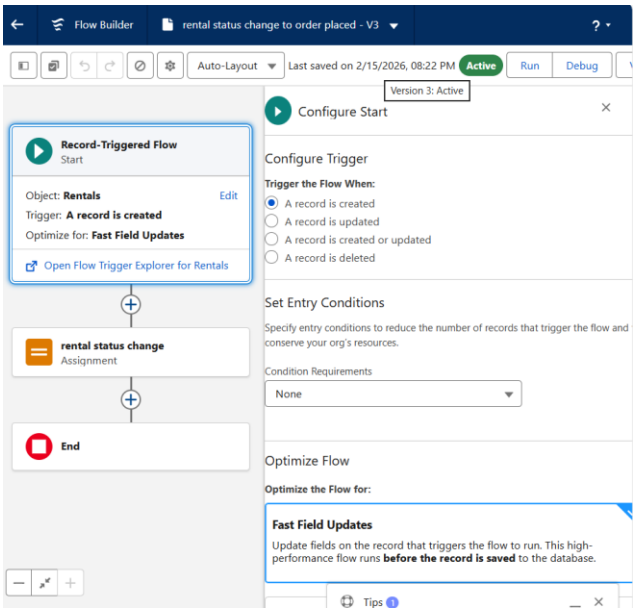
The system now prevents invalid rental date entries, ensures logical rental duration and protects data integrity. Users receive clear, immediate error messages before saving.

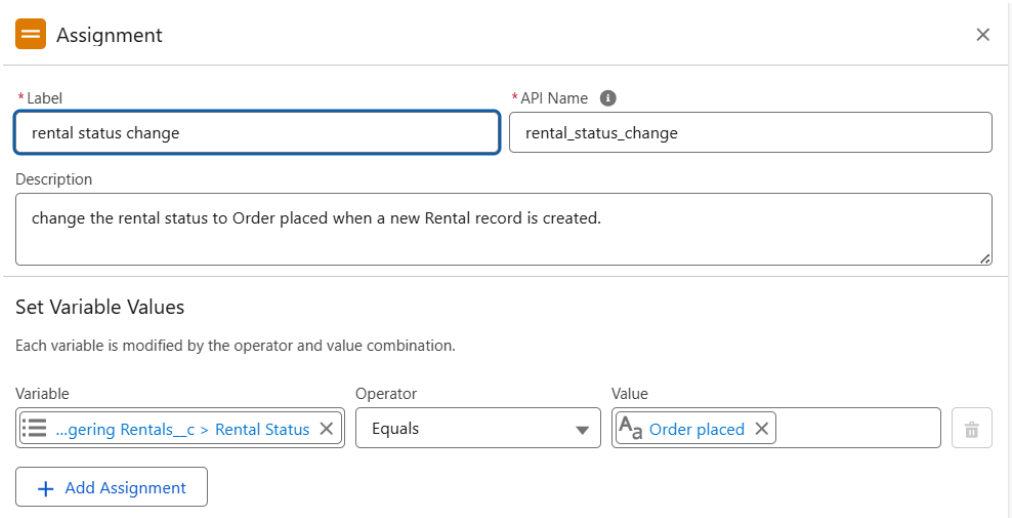
PHASE 5) Automation & Flow Implementation

FLOW 1: Rental Status Change to Order Placed

Flow type : Record- triggered flow.

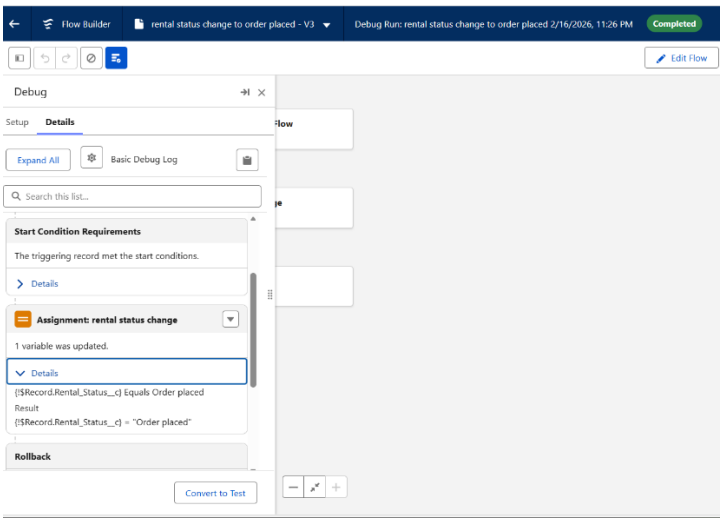
Used record trigger type, when a rental record is created, the system should automatically change the default status “in progress” of the rental to the status to Order Placed to reflect that the rental request has been successfully submitted.





This flow ensures consistent status tracking, eliminates manual updates and improves reporting accuracy.

Used the feature Debug to test the flow before activating it.



FLOW 2: Status Update to Past Due

Flow type: Record-Triggered Flow

Goal: If a rental has not been returned after the rental duration has passed, the system must automatically change the status to Past Due.

Trigger Configuration

- Object: Rentals
- Trigger: Record Updated, because a rental can't a past due status when it is created but only later on.
- Condition: Return Date < TODAY
- Schedule path: I used schedule path because I have to execute this part of a flow at a scheduled time after a triggering event.

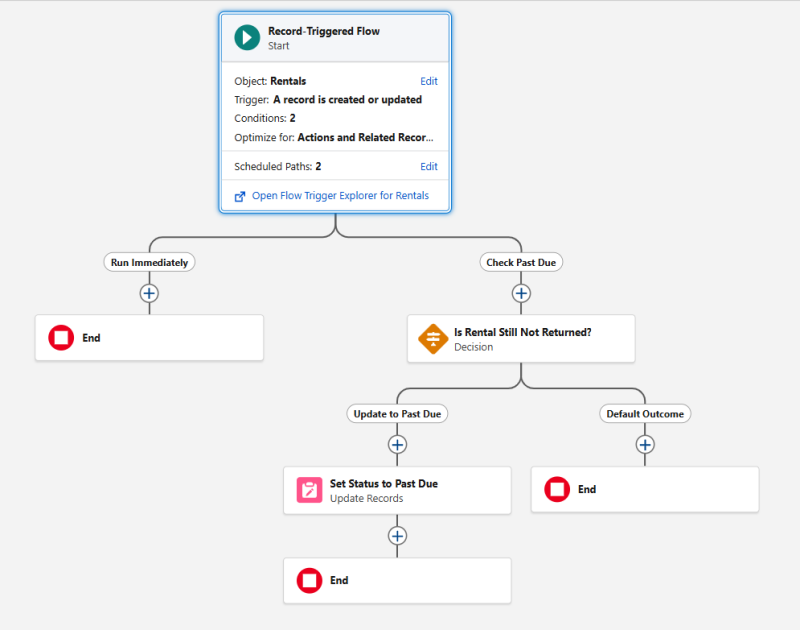
Flow Logic

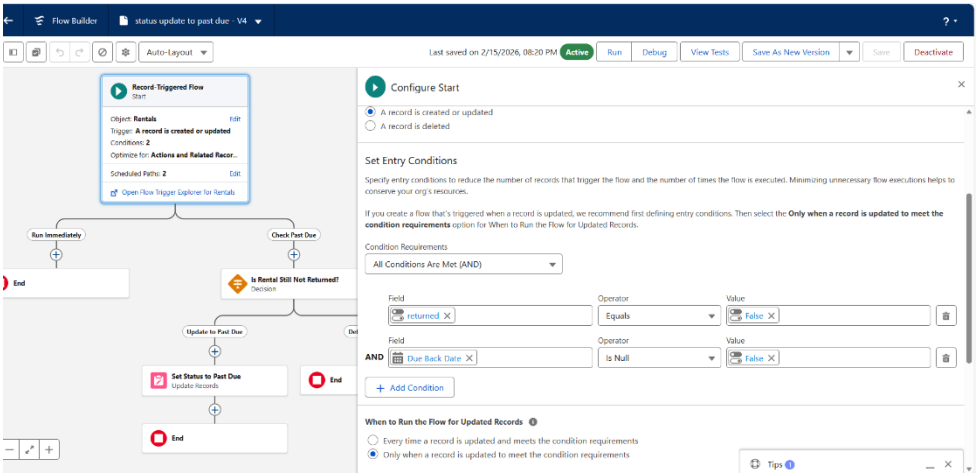
The flow evaluates whether:

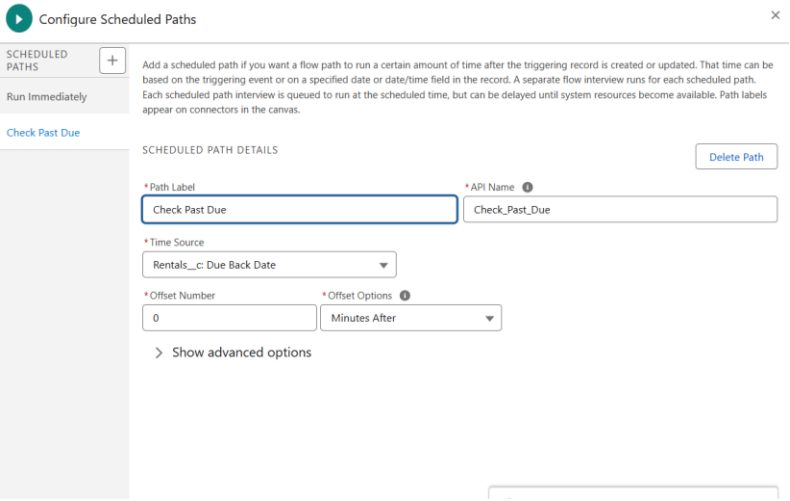
- Status ≠ Returned
- AND Due Back Date is before today

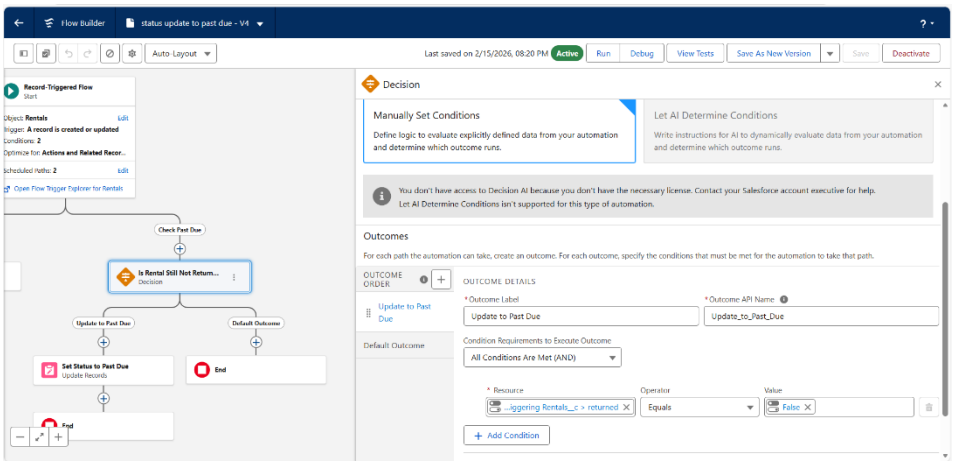
If true:

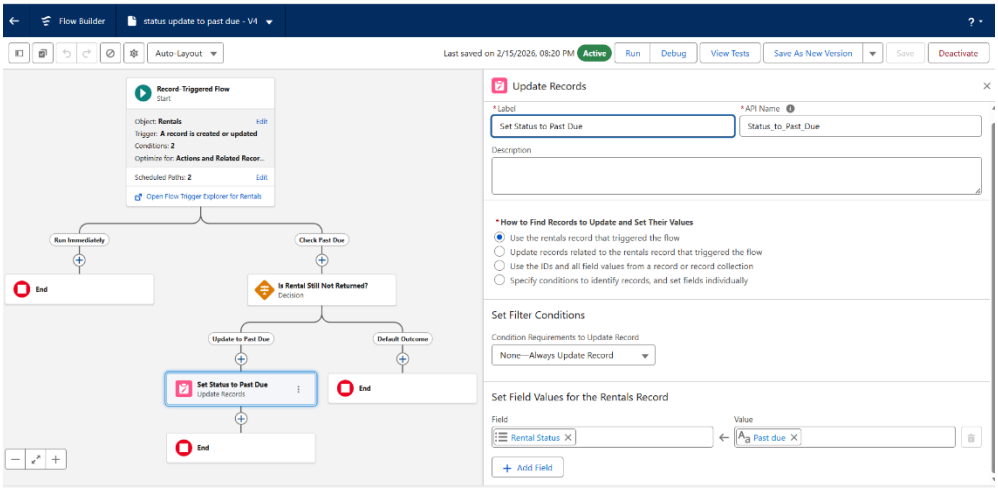
- Update Status → Past Due


Record-Triggered Flow
Start
Object: Rentals
Edit
Trigger: A record is created or updated
Conditions: 2
Optimize for: Actions and Related Recor...
Scheduled Paths: 2
Edit
Open Flow Trigger Explorer for Rentals
Run Immediately
Check Past Due
End
Is Rental Still Not Returned?
Decision
Update to Past Due
Default Outcome
Set Status to Past Due
Update Records
End
End


Flow Builder
status update to past due - V4
Auto-Layout
Last saved on 2/15/2026, 08:20 PM
Active
Run
Debug
View Tests
Save As New Version
Save
Deactivate
Configure Start
A record is created or updated
A record is deleted
Set Entry Conditions
Specify entry conditions to reduce the number of records that trigger the flow and the number of times the flow is executed. Minimizing unnecessary flow executions helps to conserve your org's resources.
If you create a flow that's triggered when a record is updated, we recommend first defining entry conditions. Then select the Only when a record is updated to meet the condition requirements option for When to Run the Flow for Updated Records.
Condition Requirements
All Conditions Are Met (AND)
Field
returned
Operator
Equals
Value
False
AND
Due Back Date
Is Null
Add Condition
When to Run the Flow for Updated Records
Every time a record is updated and meets the condition requirements
Only when a record is updated to meet the condition requirements
Tips


Configure Scheduled Paths
SCHEDULED PATHS
Run Immediately
Check Past Due
Add a scheduled path if you want a flow path to run a certain amount of time after the triggering record is created or updated. That time can be based on the triggering event or on a specified date or date/time field in the record. A separate flow interview runs for each scheduled path. Each scheduled path interview is queued to run at the scheduled time, but can be delayed until system resources become available. Path labels appear on connectors in the canvas.
SCHEDULED PATH DETAILS
Delete Path
*Path Label
Check Past Due
*API Name
Check_Past_Due
*Time Source
Rentals__c: Due Back Date
*Offset Number
0
*Offset Options
Minutes After
Show advanced options





FLOW 3: Set Status to Returned

Flow type: Record-Triggered Flow

When a rental item is returned, the system should automatically update the rental status to Returned**.**

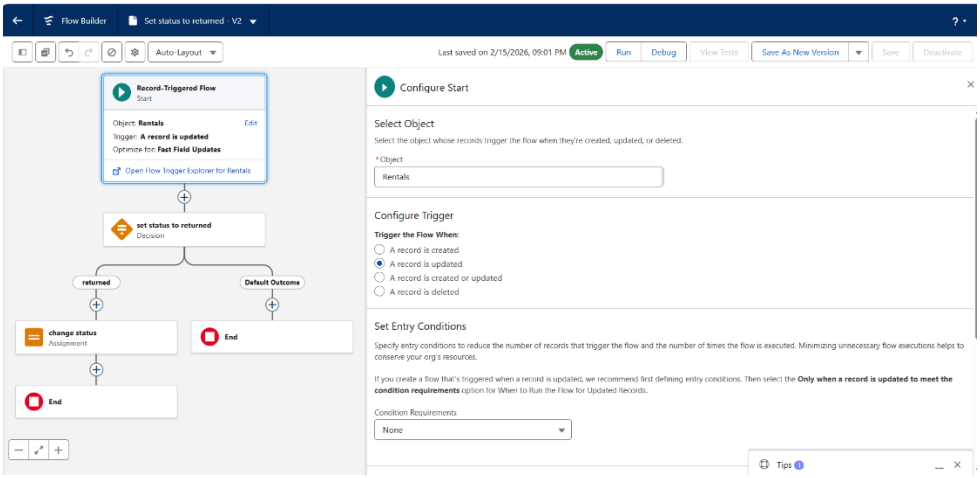
Trigger Configuration

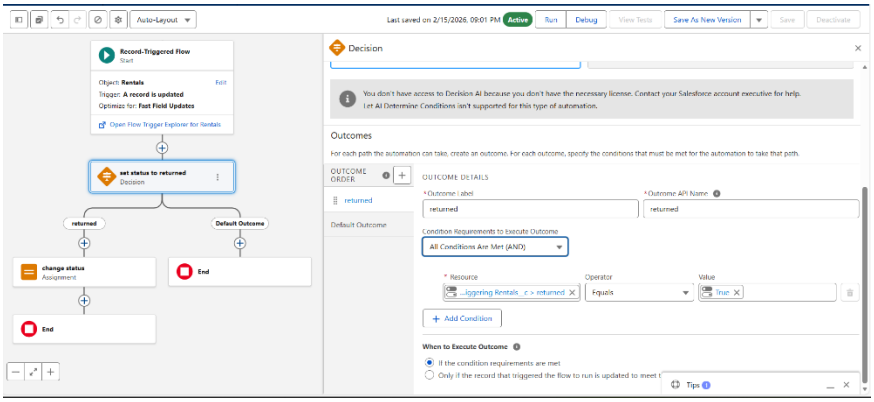
- Object: Rentals
- Trigger: Record Updated
- Condition: Return Checkbox = True, every time the return check box is checked the flow should automatically update the status to returned

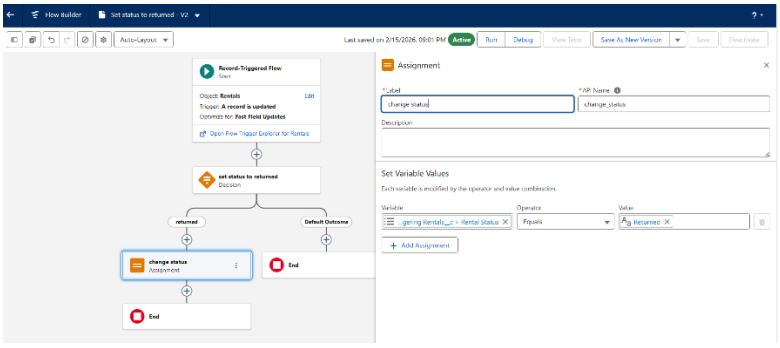
Flow Logic

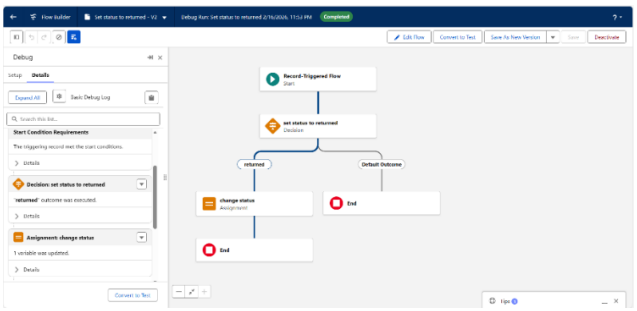
When return condition is met a want the flow to:

- Update Status to Returned









The goal of this flow is to maintain accurate tracking, prevents incorrect Past Due updates and supports accurate reports.

FLOW 4: Send Email When Rental Is Past Due

Flow Type: Record-Triggered Flow

When a rental status changes to **Past Due**, an automated email notification should be sent to the related Contact to inform them of the late return.

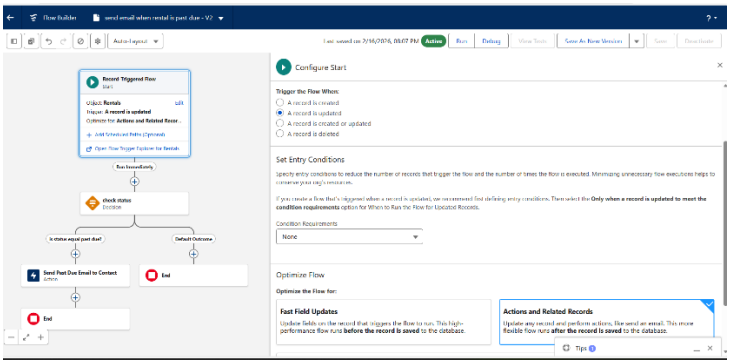
Trigger Configuration

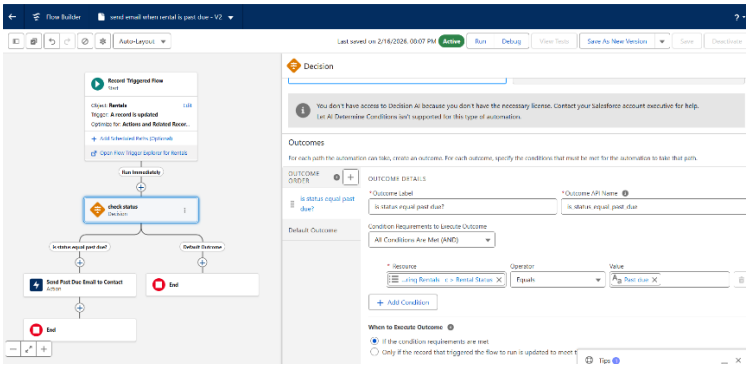
- Object: Rentals
- Trigger: this will be triggered when a record is updated
- Condition: when the rental status = Past Due

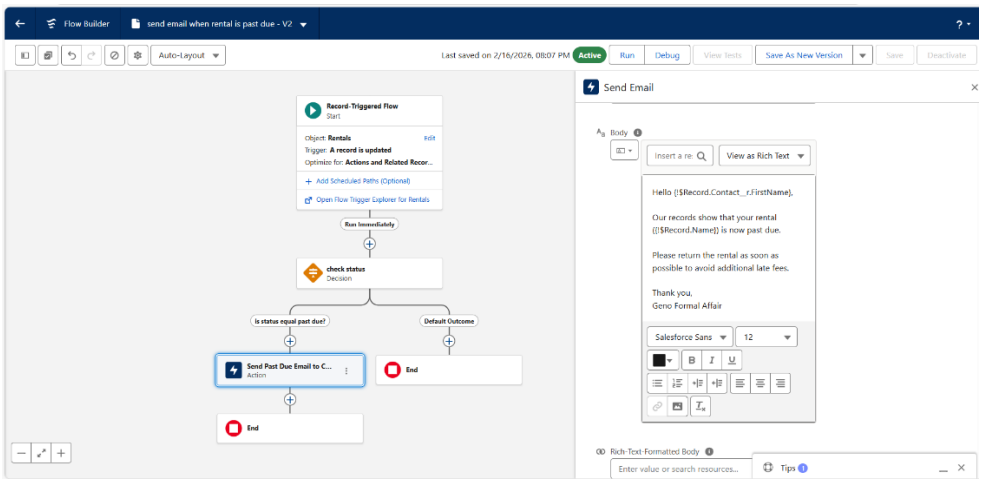
Flow Logic

When status changes to Past Due an email should be sent.

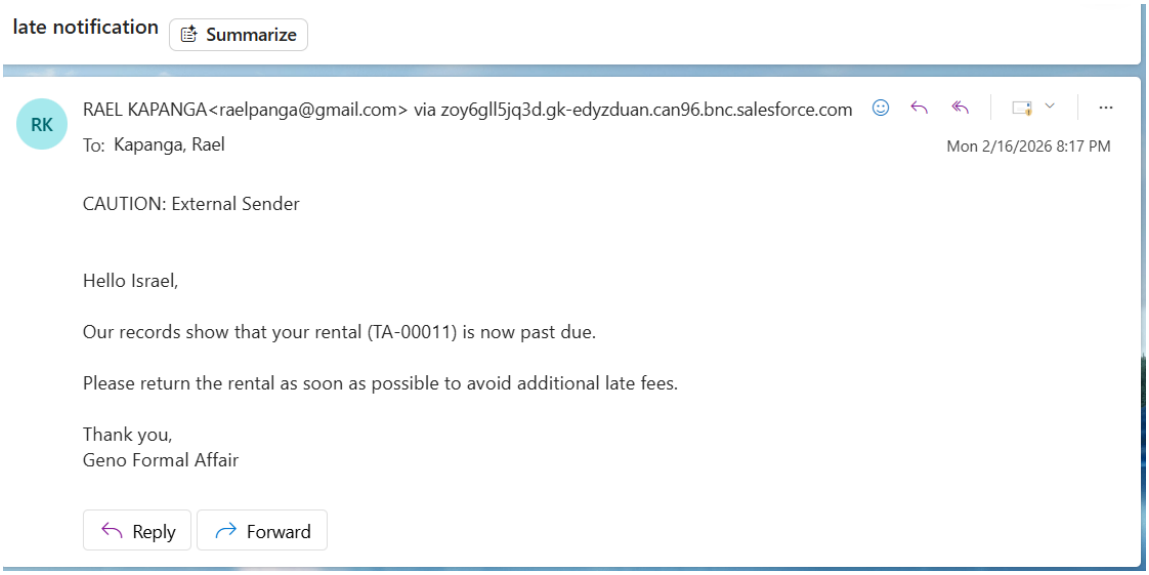
- I used the Send Email feature in the Action under interaction
- The email should be send to {!$Record.Contact__r.Email}, this formula indicates the email of the contact.







Output result



This flow improves customer communication and reduces manual follow-up.

The system was designed using event-driven automation principles.
Each flow handles a specific lifecycle stage of the Rental object, ensuring:

- Clear separation of logic
- Scalable automation design

  Minimal manual intervention