

Phase 4: Process Automation (RetailHub)

This phase focused on implementing the **core business logic** of the RetailHub CRM application through **Process Automation**. Salesforce's declarative tools were leveraged to automate complex processes, enforce data quality, and create a **dynamic and responsive user experience**.

Phase 4.1: Validation Rules

A critical component of process automation is **enforcing data integrity and business rules** at the point of user entry. For the RetailHub CRM project, **Validation Rules** were used to ensure that only **valid data** is entered into the system, thereby maintaining data quality and preventing common user errors.

A **Validation Rule** is a declarative mechanism that evaluates a formula or expression. If the formula evaluates to **TRUE**, the rule is triggered, preventing the record from being saved and displaying a **custom error message** to the user.

Inventory Check Validation Rule

One of the key validation rules implemented addressed a critical business need: **preventing the sale of out-of-stock items**.

Object: `Purchase_Line_Item__c`

Purpose: Ensure that sales representatives cannot create or update a purchase line item for a quantity that exceeds available stock.

Evaluation Criteria:

- The rule triggers **whenever a Purchase Line Item record is created or updated**.
- This ensures real-time stock validation whenever a product is added or modified in a sale.

Error Condition Formula:

`Quantity__c > Product__r.Stock_Quantity__c`

- This formula compares the quantity entered on the line item with the **Stock_Quantity__c** field of the related parent product.
- If the quantity requested exceeds the available stock, the rule prevents the record from being saved.

The screenshot shows the Salesforce Setup interface for a Validation Rule. The breadcrumb trail is 'SETUP > OBJECT MANAGER'. The page title is 'Purchase Line Item'. The left sidebar contains a navigation menu with the following items: Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, List View Button Layout, and Restriction Rules. The main content area is titled 'Purchase Line Item Validation Rule' with a 'Back to Purchase Line Item' link. Below the title is a 'Validation Rule Detail' section with 'Edit' and 'Clone' buttons. The details are as follows:

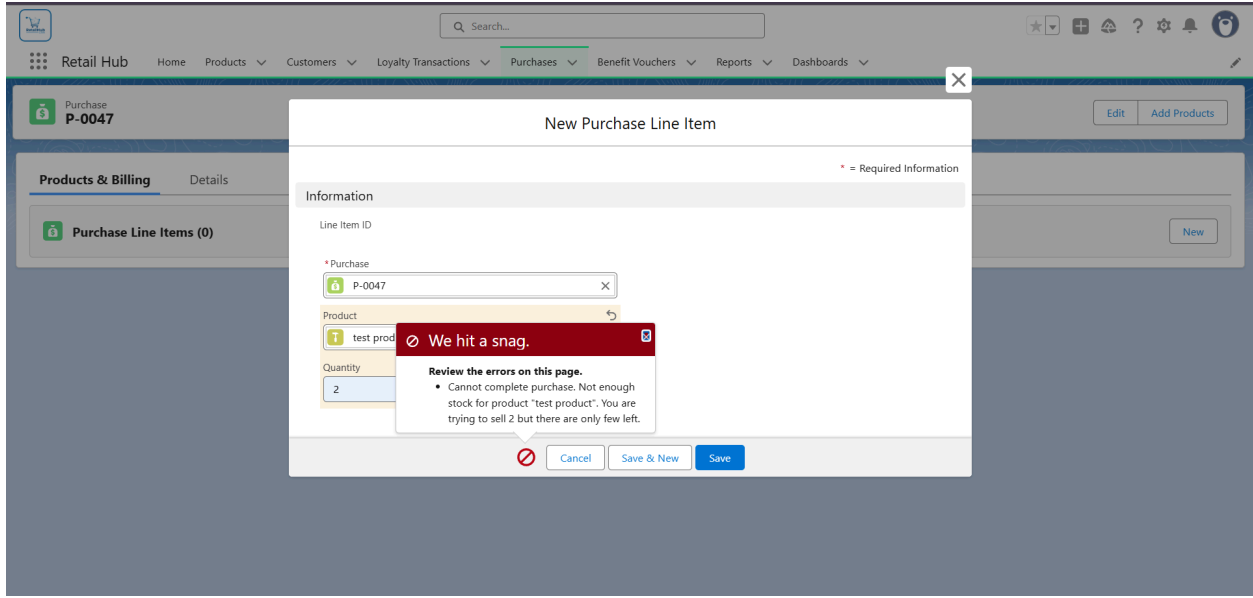
Rule Name	Prevent_Sale_of_Out_of_Stock_Items	Active	<input type="checkbox"/>
Error Condition Formula	Quantity__c > Product__r.Stock_Quantity__c		
Error Message	"You cannot sell this quantity. There are only items left in stock."		Error Location: Top of Page
Description			
Created By	Amod khurasiya, 9/23/2025, 9:10 PM	Modified By	Amod khurasiya, 9/24/2025, 7:01 AM

At the bottom of the details section, there are 'Edit' and 'Clone' buttons.

Dynamic Error Message:

```
"You cannot sell this quantity. " &
IF(
    Product__r.Stock_Quantity__c > 0,
    "There are only " & TEXT(Product__r.Stock_Quantity__c) & " items
left in stock.",
    "This item is currently out of stock."
)
```

- The message provides a **clear, actionable explanation** to the user.
- It dynamically informs the sales representative of the **exact quantity available** or indicates that the item is completely out of stock.



Benefits:

- Maintains **data accuracy** and prevents overselling.
- Provides a **user-friendly, informative error message** to guide immediate corrective action.
- Reduces **inventory discrepancies** and supports accurate reporting and customer satisfaction.

Phase 4.2: Flow Builder

The **Flow Builder** was the primary tool used to implement complex, multi-step automation in the RetailHub CRM. Flow Builder was selected over legacy automation tools due to its **superior flexibility, efficiency, and ability to handle sophisticated logic**.

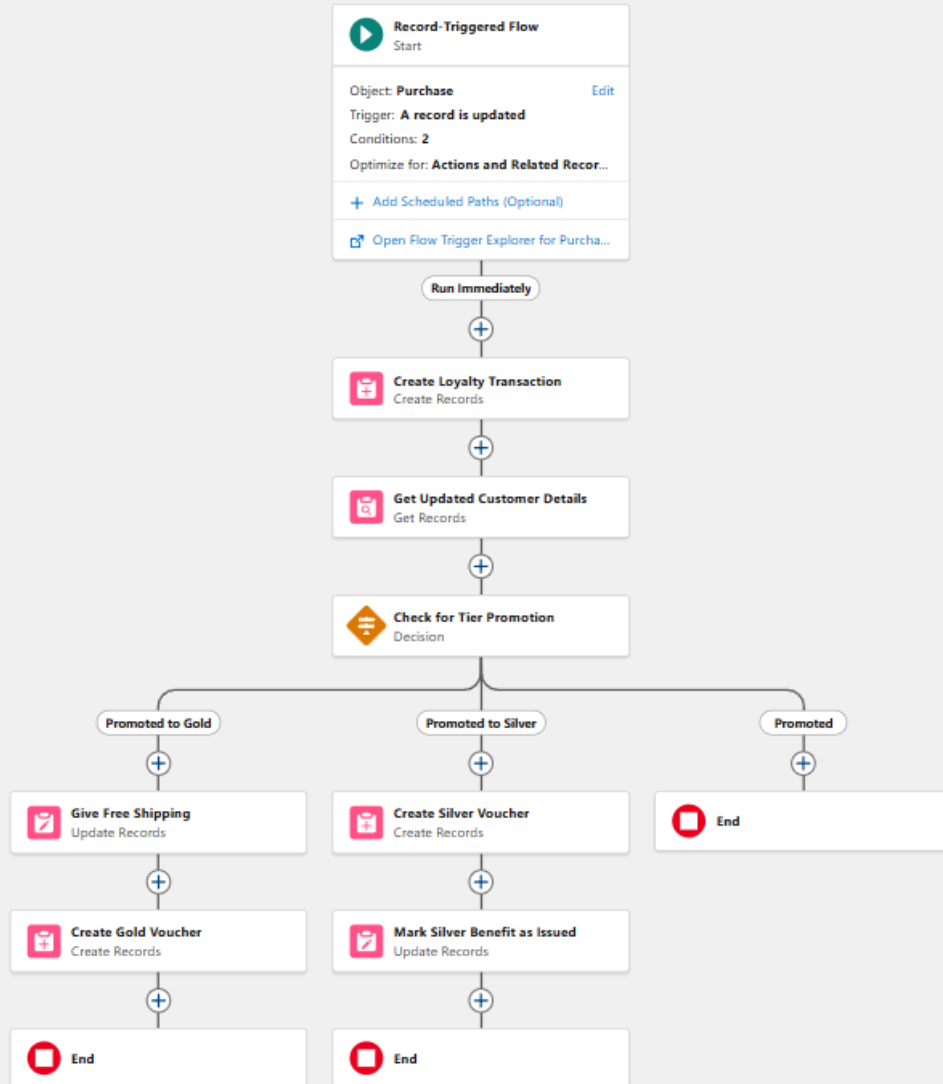
The project utilized:

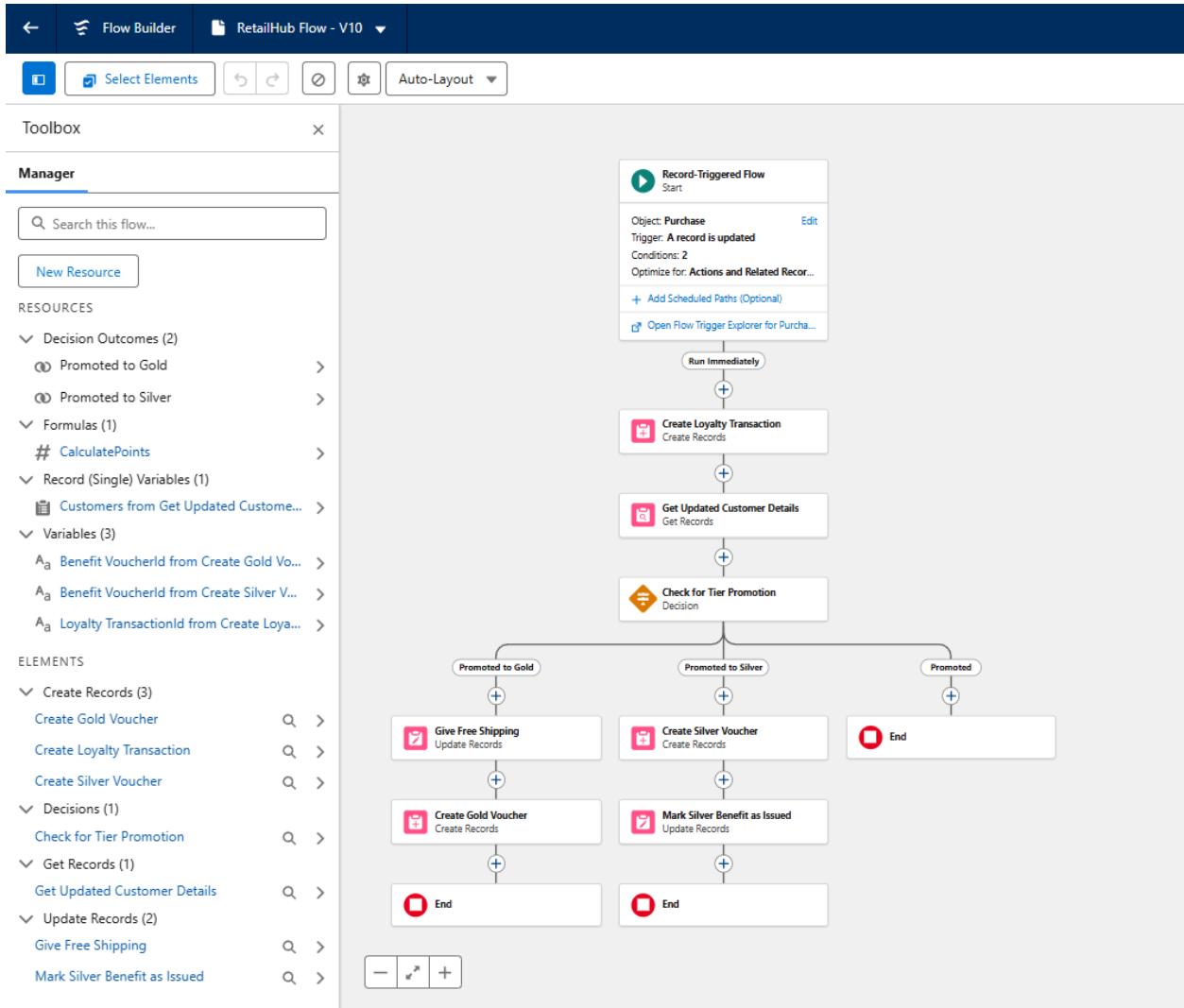
- **Record-Triggered Flows** – Run automatically in the background based on changes to records.
 - **Screen Flows** – Provide interactive user interfaces for end-users.
-

4.2.1 Flow: Set Price on Purchase Line Item

Objective: Ensure accurate pricing at the moment a product is added to a purchase.

- **Type:** Record-Triggered Flow (Optimized for Fast Field Updates)
- **Trigger:** Runs automatically **before a Purchase Line Item record is created**.
- **Logic:**
 - Retrieves the **Price__c** from the related parent **Product** record.
 - Updates the **Unit_Price__c** field on the new Purchase Line Item.
 - Ensures the product price at the time of sale is **accurately and permanently recorded** on the transaction line.





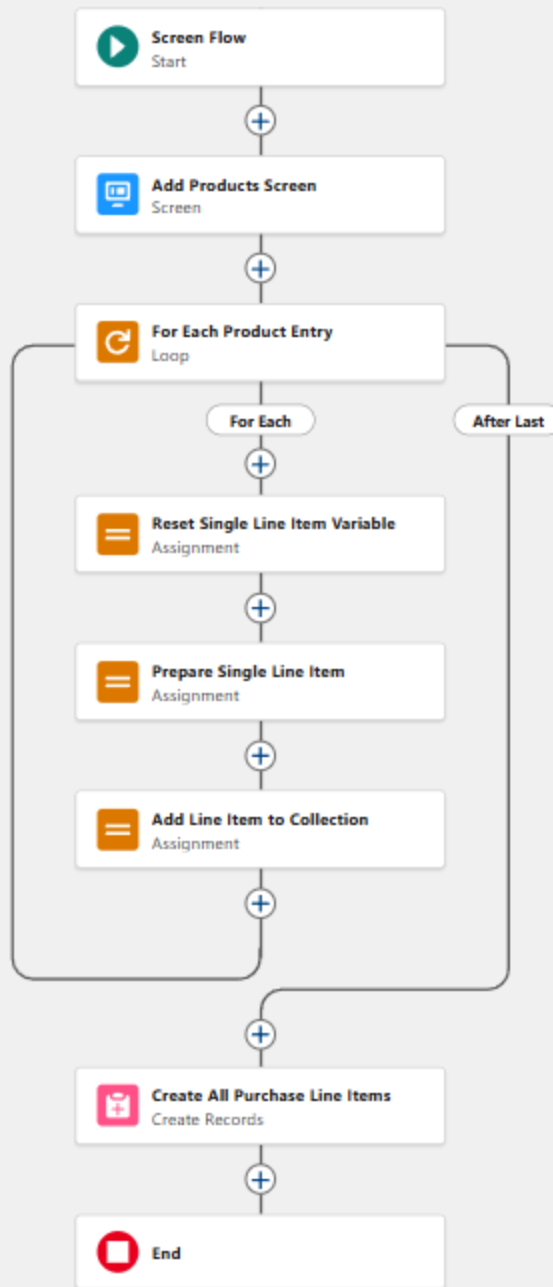
4.2.2 Flow: Add Multiple Products

Objective: Streamline the process of adding multiple products to a purchase.

- **Type:** Screen Flow
- **User Interface:**
 - Launched from a custom “**Add Products**” action on the Purchase record page.
 - Presents a single, interactive screen with a **Repeater component**, allowing sales reps to add multiple product-quantity rows in one step.

- **Logic:**

- Uses a **Loop** to iterate through each product entry submitted by the user.
- **Assignment elements** prepare each Purchase Line Item record in memory.
- A **Create Records** element inserts the entire collection at once, ensuring an **efficient and bulk-safe operation**.



Toolbox

Search this flow...

New Resource

RESOURCES

Loops (1)

Current Item from Loop For Each Pr... >

Record (Single) Variables (1)

singleLineItem >

Record Collection Variables (1)

purchaseLineItemsToCreate >

Screen Components (3)

Product_List >

Product >

Quantity >

Variables (1)

recordId >

ELEMENTS

Assignments (3)

Add Line Item to Collection 🔍 >

Prepare Single Line Item 🔍 >

Reset Single Line Item Variable 🔍 >

Create Records (1)

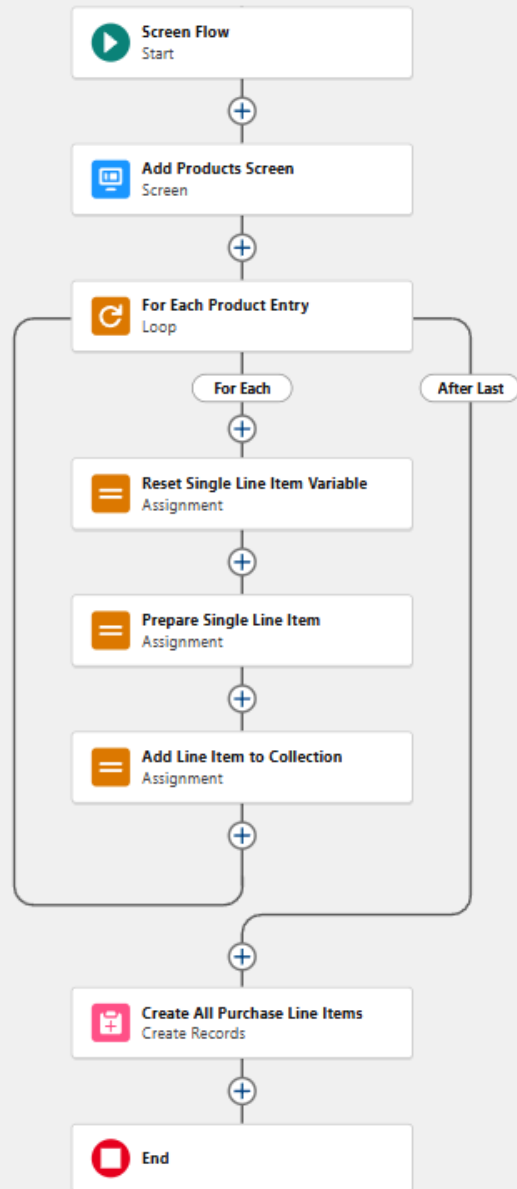
Create All Purchase Line Items 🔍 >

Loops (1)

For Each Product Entry 🔍 >

Screens (1)

Add Products Screen 🔍 >



4.2.3 Flow: Main RetailHub Flow

Objective: Automate the post-sale business lifecycle, including inventory updates, loyalty point calculation, and benefits management.

- **Type:** Record-Triggered Flow
- **Trigger:** Runs **after a Purchase record is updated**, specifically when **Status__c** changes from "Draft" to "Completed".

- **Logic:** The flow executes the following actions in sequence:

1. Inventory Deduction

- Retrieves all related **Purchase Line Items**.
- Loops through each item and updates the corresponding **Product.Stock_Quantity__c** to subtract the sold quantity.

2. Loyalty Points Calculation

- Creates a **Loyalty Transaction** record for the customer.
- Uses a formula within the flow to calculate points based on the customer's **current Loyalty Tier**, granting higher points for more loyal customers.

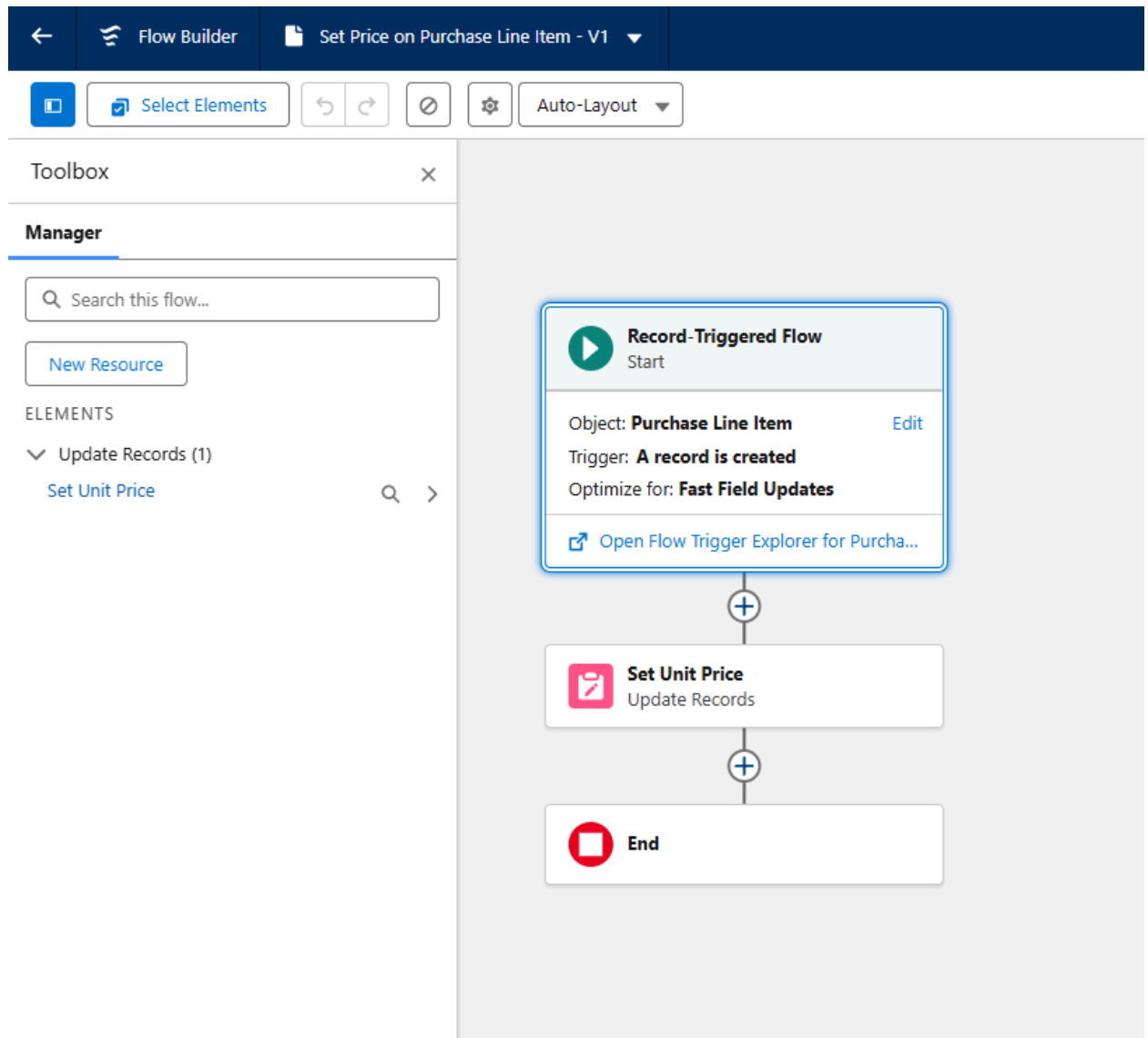
3. Tier Promotion & Benefits

- Evaluates the customer's updated point total using a **Decision element**.
- If a promotion is earned, the flow:
 - Creates the appropriate **Benefit Voucher** record.
 - Updates the **Customer** record to grant ongoing perks such as **free shipping**.

Key Benefits of Flow Automation

- Automates critical business processes with **minimal manual effort**.

- Ensures **data accuracy** in pricing, inventory, and loyalty calculations.
- Provides a **seamless and interactive experience** for sales representatives.
- Supports **scalable and bulk-safe operations**, reducing errors and processing time.



Toolbox

Manager

Search this flow...

New Resource

RESOURCES

Formulas (1)

newQuantityOnHand

ELEMENTS

Update Records (1)

Deduct Product Quantity

Record-Triggered Flow
Start

Object: **Purchase Line Item** [Edit](#)

Trigger: **A record is created**

Optimize for: **Actions and Related Recor...**

[+ Add Scheduled Paths \(Optional\)](#)

[Open Flow Trigger Explorer for Purcha...](#)

Run Immediately



Deduct Product Quantity

Update Records



End