Project Title: Jewellery Store Management

Problem Statement:

Jewellery stores like Jewellery shop face challenges in managing customer relationships, tracking sales, and offering personalized services. Customers often do not receive timely updates on new collections, exclusive offers, or important services such as repairs and maintenance. Sales staff may not have access to complete customer profiles or purchase history, making it difficult to deliver personalized recommendations or upsell effectively. Inventory tracking and post-sale services like resizing or repair updates are often handled manually, resulting in inefficiencies and delays. There is no single system that connects customer preferences, sales data, inventory, and marketing efforts—making it harder to deliver a seamless and memorable shopping experience.

Proposed Salesforce Solution:

- Customers create profiles with their preferences, past purchases, anniversaries, and style interests.
- Sales team records interactions and purchase history directly in the CRM.
- Salesforce sends automated notifications to customers about:
  - New arrivals ○ Personalized offers ○ Service reminders

  (e.g., maintenance, cleaning, or repair updates)
- Inventory is tracked in real-time, showing availability across collections and categories.
- Appointments for in-store visits or virtual consultations can be scheduled and tracked inside Salesforce.
- Dashboards provide real-time insights to store managers showing:
  - Top-selling products and collections
  - Customer buying trends
  - Inventory levels and reorder alerts
  - Customer engagement and service history

PHASE 1: PROBLEM UNDERSTANDING & INDUSTRY ANALYSIS

Project Title: JewelleryStore Management CRM – Elevating Jewelry Retail with Smart Customer Engagement

1. REQUIREMENT GATHERING

Goal: Understand the business needs and pain points.

- Talk to store managers, sales staff, marketing, and inventory teams.
- Identify current issues like manual tracking, poor customer communication, and lack of personalization.
- Gather real examples of what users want Salesforce to do.

2. STAKEHOLDER ANALYSIS

Goal: Identify who will use or benefit from the CRM.

Internal Users:

- Sales Team: Needs customer history and inventory info.
- Managers: Want reports and dashboards.
- Marketing Team: Needs customer data for targeted campaigns.
- Inventory Staff: Needs live stock updates and alerts.

External Users:

- Customers: Want timely updates, offers, and personalized service.

3. BUSINESS PROCESS MAPPING

Goal: Understand how work is currently done and where Salesforce can help.

Key Areas:

- Customer onboarding is manual.
- No system to track customer purchases.
- Inventory is tracked separately, often manually.
- No proper tracking of post-sale services like repairs or resizing.
- Marketing messages are not personalized.

How Salesforce Helps:

- Automates data entry and updates.
- Centralizes sales and customer info.
- Tracks inventory in real-time.

- Sends automatic reminders and personalized offers.

4. INDUSTRY USE CASE ANALYSIS

Goal: Understand jewelry industry needs to design a better CRM.

Jewelry Retail Needs:

- Personalized customer service (preferences, occasions).
- Follow-ups for high-value purchases.
- Regular reminders for cleaning or repairs.
- Building long-term customer loyalty.

Salesforce Solutions:

- Custom fields for preferences, ring size, etc.
- Service Cloud for tracking repairs.
- Marketing automation for events like birthdays or anniversaries.
- Dashboards to track trends and performance.

5. APPEXCHANGE EXPLORATION

Goal: Find ready-to-use Salesforce apps that add value.

Helpful App Types:

- Appointment booking
- Loyalty and rewards management
- Inventory and repair tracking

Action Steps:

- Explore top-rated apps on AppExchange.
- Check for compatibility with Sales or Service Cloud.
- Choose between using apps or custom development.

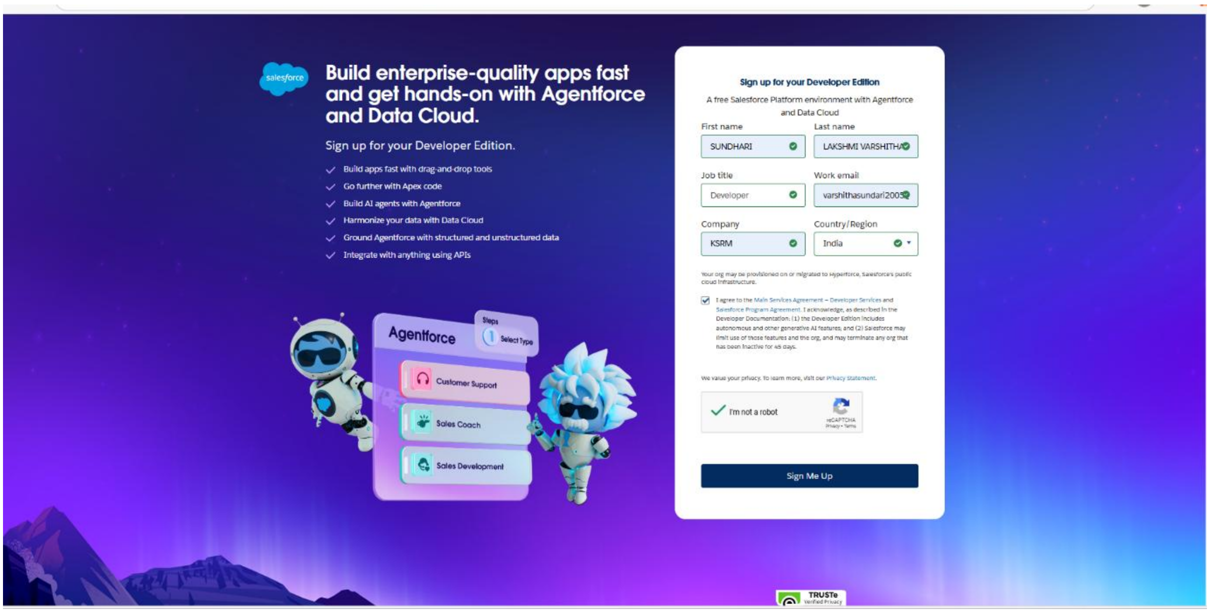
Phase 2: Org Setup & Configuration

Project : GoldenEra Enterprises CRM

This document summarizes the work performed in **Phase 2** of the capstone: setting up the Salesforce Developer Org and configuring the resources required for the Intelligent Case Routing project. It includes step-by-step actions completed and a screenshot of the custom object & fields created for routing configuration.

Step 1- Sign up

sign up for your developer edition



Build enterprise-quality apps fast and get hands-on with Agentforce and Data Cloud.

Sign up for your Developer Edition.

- ✓ Build apps fast with drag-and-drop tools
- ✓ Go further with Apex code
- ✓ Build AI agents with Agentforce
- ✓ Harmonize your data with Data Cloud
- ✓ Ground Agentforce with structured and unstructured data
- ✓ Integrate with anything using APIs

Sign up for your Developer Edition

A free Salesforce Platform environment with Agentforce and Data Cloud

First name: SUNDHARI ✓	Last name: LAKSHMI VARSHITHA ✓
Job title: Developer ✓	Work email: varshithasundari2005 ✓
Company: KSRM ✓	Country/Region: India ✓

Your org may be provisioned on or migrated to Hyperforce, Salesforce's public cloud infrastructure.

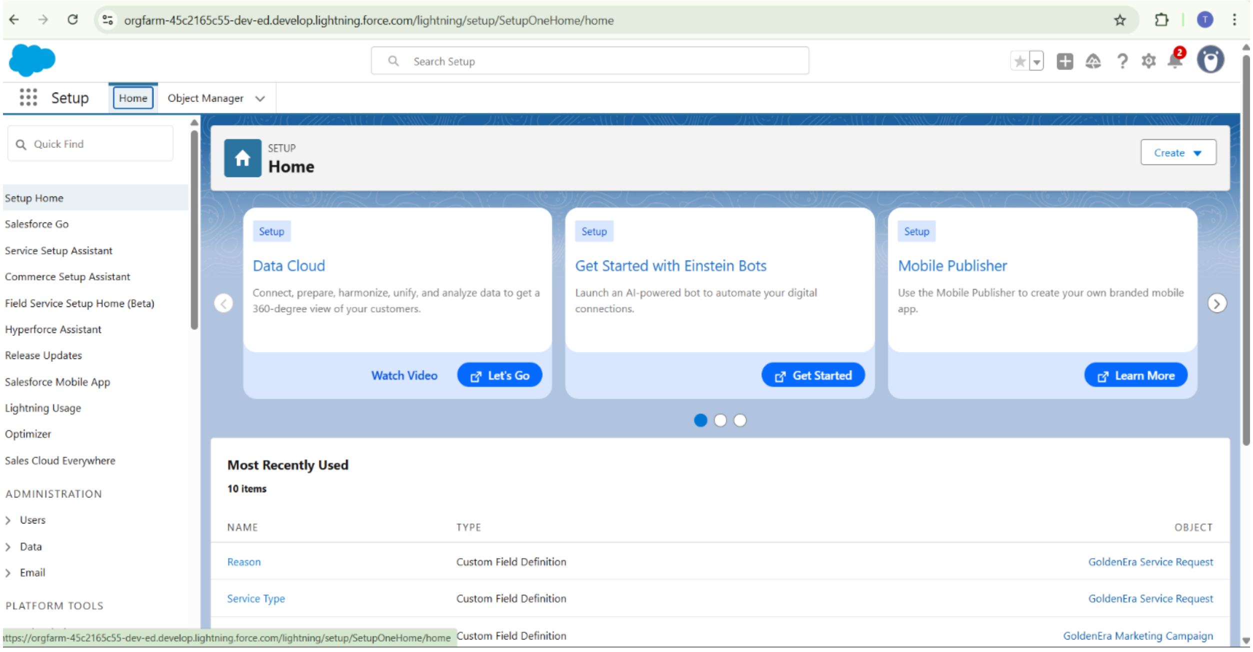
☑ I agree to the Main Services Agreement – Developer Services and Salesforce Program Agreement. I acknowledge, as described in the Developer Documentation: (1) the Developer Edition includes autonomous and other generative AI features; and (2) Salesforce may limit use of those features and the org, and may terminate any org that has been inactive for 45 days.

We value your privacy. To learn more, visit our Privacy Statement.

✓ I'm not a robot

Step 2: Open Setup

1. Login to Salesforce Lightning.
2. Click the **Gear icon** in the top-right → select **Setup**.



orgfarm-45c2165c55-dev-ed.develop.lightning.force.com/lightning/setup/SetupOneHome/home

Setup — Data Cloud: Connect, prepare, harmonize, unify, and analyze data to get a 360-degree view of your customers. [Watch Video] [Let's Go]

Setup — Get Started with Einstein Bots: Launch an AI-powered bot to automate your digital connections. [Get Started]

Setup — Mobile Publisher: Use the Mobile Publisher to create your own branded mobile app. [Learn More]

Most Recently Used

10 items

NAME	TYPE	OBJECT
Reason	Custom Field Definition	GoldenEra Service Request
Service Type	Custom Field Definition	GoldenEra Service Request
	Custom Field Definition	GoldenEra Marketing Campaign

ttps://orgfarm-45c2165c55-dev-ed.develop.lightning.force.com/lightning/setup/SetupOneHome/home

Step 3: Update Company Information

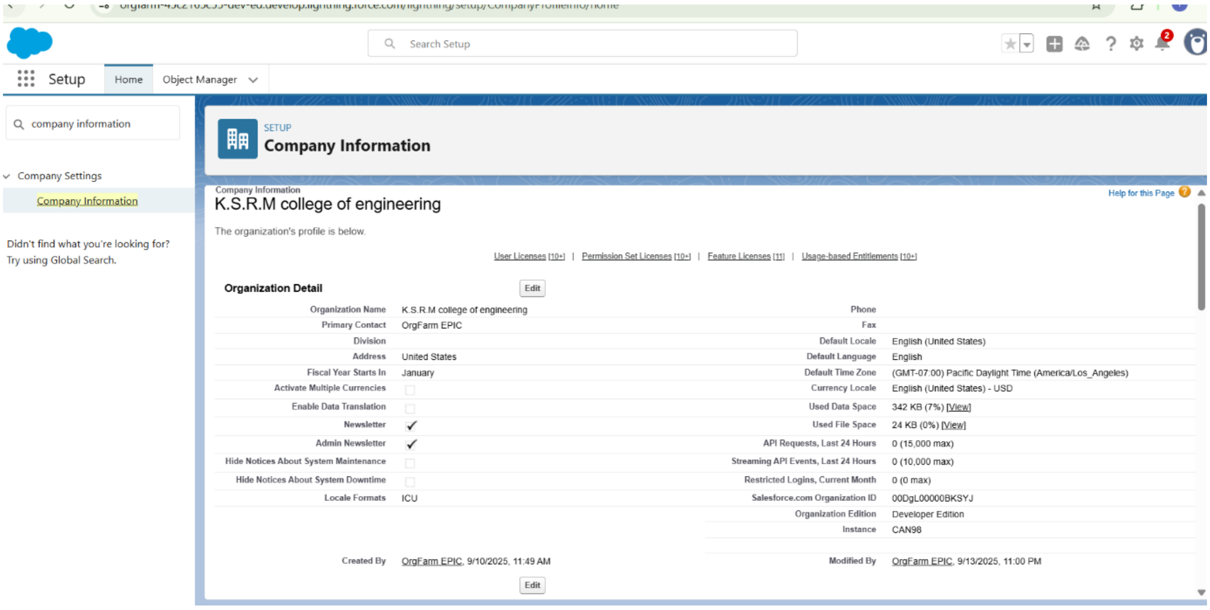
1. In Setup, use **Quick Find** → type **Company Information** → open it.

2. Click **Edit**.

3. Update:

   - **Organization Name:** KSRM College of Engineering
   - **Default Time Zone:** (09:00 – 18:00)

4. Click **Save**.



Company Information

K.S.R.M college of engineering

The organization's profile is below.

Organization Detail

Organization Name	K.S.R.M college of engineering	Phone	
Primary Contact	OrgFarm EPIC	Fax	
Division		Default Locale	English (United States)
Address	United States	Default Language	English
Fiscal Year Starts In	January	Default Time Zone	(GMT-07:00) Pacific Daylight Time (America/Los_Angeles)
Activate Multiple Currencies	□	Currency Locale	English (United States) - USD
Enable Data Translation	□	Used Data Space	342 KB (7%) [View]
Newsletter	✓	Used File Space	24 KB (0%) [View]
Admin Newsletter	✓	API Requests, Last 24 Hours	0 (15,000 max)
Hide Notices About System Maintenance	□	Streaming API Events, Last 24 Hours	0 (10,000 max)
Hide Notices About System Downtime	□	Restricted Logins, Current Month	0 (0 max)
Locale Formats	ICU	Salesforce.com Organization ID	00DgL00000BKSYJ
		Organization Edition	Developer Edition
		Instance	CAN98
Created By	OrgFarm EPIC, 9/10/2025, 11:49 AM	Modified By	OrgFarm EPIC, 9/13/2025, 11:00 PM

Step 4: Set Business Hours

1. Quick Find → **Business Hours** → click **New**.

2. Fill in:

   Name: Default Hours

   Hours: 09:00 – 18:00 (or your actual business hours)

3. Click **Save**.

Why: Defines working hours for workflows, notifications, and approval processes

---

Step 5: Profiles

Profiles define what users **can do** . For Golden Era Enterprises, you'd create these Salesforce profiles:

a) System Administrator

- Full access to all standard/custom objects, settings, and configuration.

- Used by IT/Admins

Step 6: Standard and Custom Objects

1. **GoldenEra Customer**
   - Stores all customer information (profiles, preferences, history).
   - Central object for connecting orders, service requests, and campaigns.
2. **GoldenEra Orders**
   - Tracks all jewelry purchases made by customers.
   - Connects with products and updates inventory.
3. **GoldenEra Products**
   - Holds product catalog details (rings, necklaces, bridal sets, etc.).
   - Links with inventory and orders.
4. **GoldenEra Inventory**
   - Manages stock availability and locations.
   - Supports reorder alerts and stock insights.
5. **GoldenEra Marketing Campaign**
   - Manages campaigns (festive sales, offers, events).
   - Tracks customer engagement and ROI.
6. **GoldenEra Service Request**
   - Handles after-sales services (repairs, resizing, cleaning).
   - Connects with customers and their orders.

Phase 3: Data Modeling & Relationships

Step 1: Standard & Custom Objects

Six custom objects were created to store business-critical data:

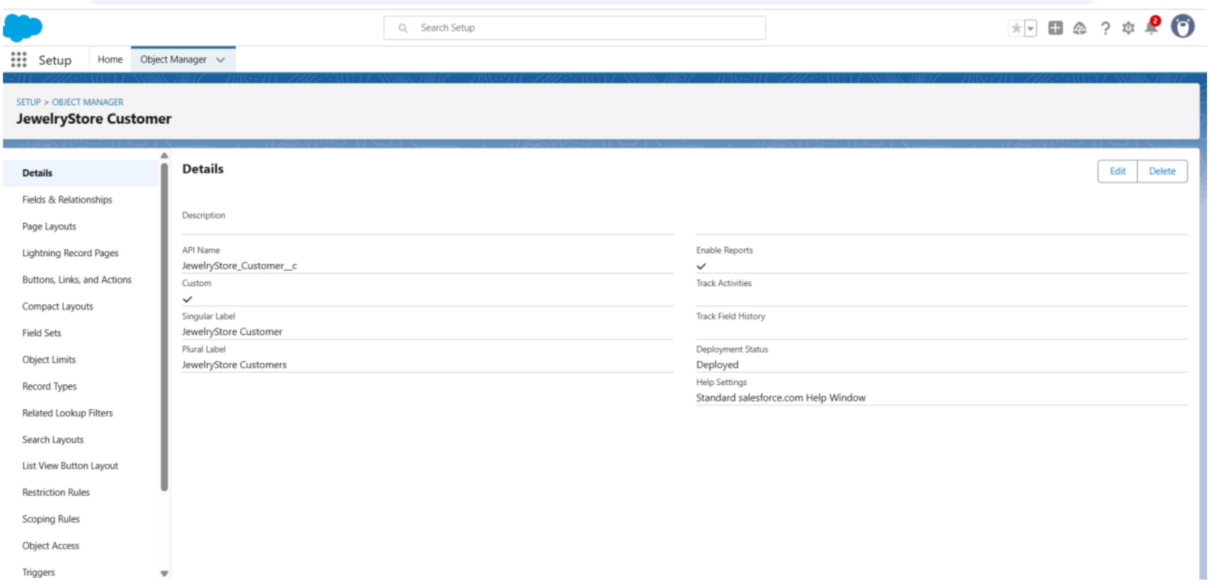
- **JewelleryStore Customer** – Stores customer details such as preferences, anniversaries, and purchase history.
- **JewelleryStore Product** – Stores product catalog details such as category, material, price, and SKU.
- **JewelleryStore Order** – Stores orders placed by customers, including order date, payment status, and delivery status.
- **JewelleryStore Inventory** – Tracks stock availability, reorder levels, and store location.
- **JewelleryStore Marketing Campaign** – Stores details of promotional campaigns such as campaign name, offers, and status.
- **JewelleryStore Repire Request** – Stores after-sales service requests such as repairs, resizing, cleaning, and their status.

Steps followed:

• Navigated to Setup → Object Manager → Create → Custom Object

- Provided label, name, and enabled reports/search

- Saved and created Tabs for each object



SETUP > OBJECT MANAGER

JewelryStore Customer

Details

Description

API Name: JewelryStore_Customer__c

Custom: ✓

Singular Label: JewelryStore Customer

Plural Label: JewelryStore Customers

Enable Reports: ✓

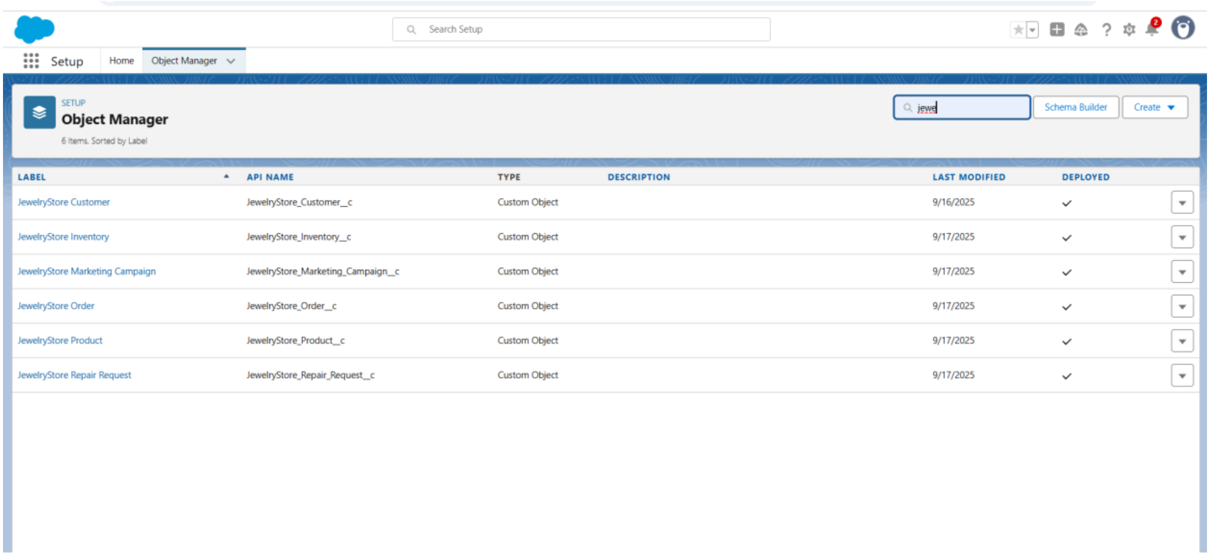
Track Activities

Track Field History

Deployment Status: Deployed

Help Settings: Standard salesforce.com Help Window

Same steps followed for all six custom object creation



SETUP

Object Manager

6 Items, Sorted by Label

LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
JewelryStore Customer	JewelryStore_Customer__c	Custom Object		9/16/2025	✓
JewelryStore Inventory	JewelryStore_Inventory__c	Custom Object		9/17/2025	✓
JewelryStore Marketing Campaign	JewelryStore_Marketing_Campaign__c	Custom Object		9/17/2025	✓
JewelryStore Order	JewelryStore_Order__c	Custom Object		9/17/2025	✓
JewelryStore Product	JewelryStore_Product__c	Custom Object		9/17/2025	✓
JewelryStore Repair Request	JewelryStore_Repair_Request__c	Custom Object		9/17/2025	✓

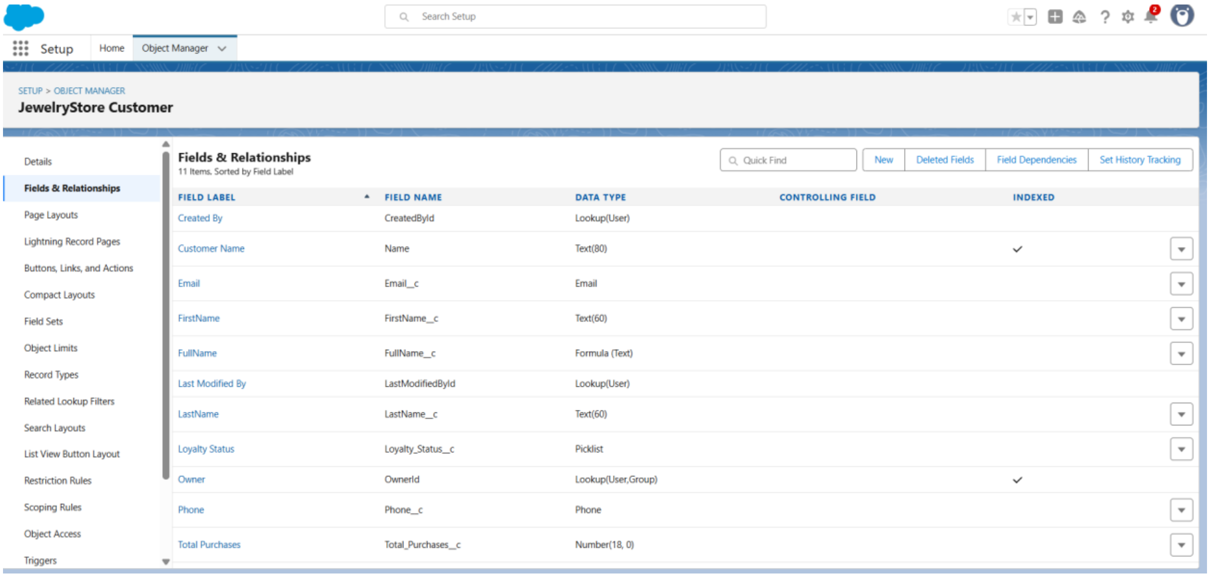
Step 2: Fields

JewelleryStore Customer

- **JewelleryStore Customer Name** (Text, 80) – Standard Name field

- **FirstName** (Text, 60)

- **LastName** (Text, 60)
- **FullName** (Formula, Text → FirstName + LastName)
- **Email** (Email)
- **Phone** (Phone)
- **Loyalty Status** (Picklist: Gold, Silver, Bronze, Platinum)
- **Total Purchases** (Number, 18,0)
- **Owner** (Lookup → User/Group)
- **Created By** (Lookup → User)
- **Last Modified By** (Lookup → User)



SETUP > OBJECT MANAGER

JewelryStore Customer

Fields & Relationships

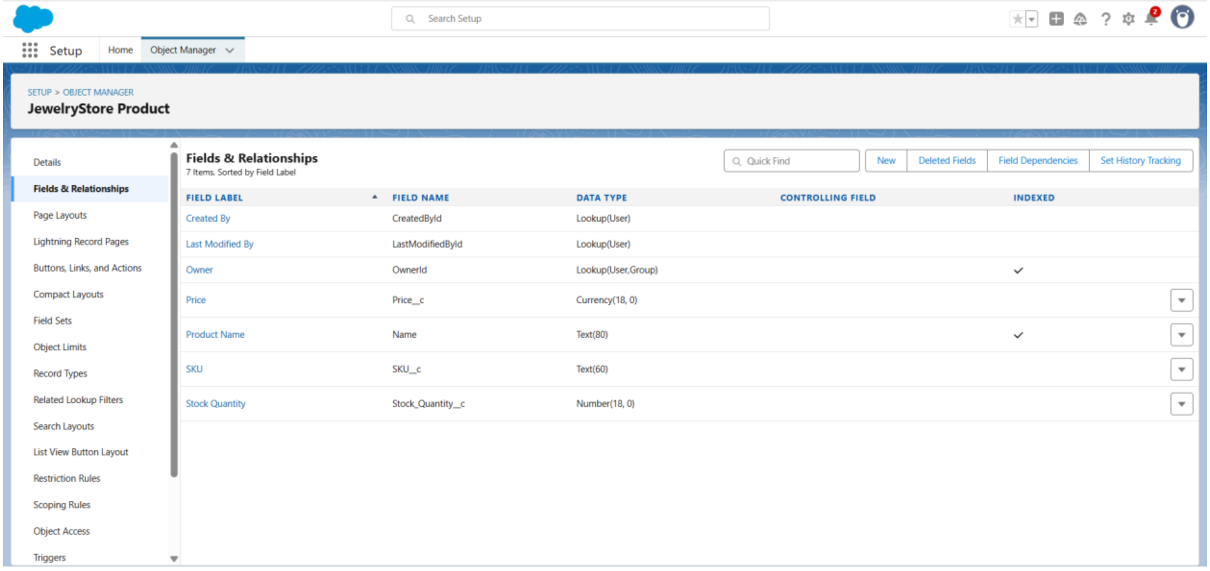
11 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Customer Name	Name	Text(80)		✓
Email	Email__c	Email		
FirstName	FirstName__c	Text(60)		
FullName	FullName__c	Formula (Text)		
Last Modified By	LastModifiedById	Lookup(User)		
LastName	LastName__c	Text(60)		
Loyalty Status	Loyalty_Status__c	Picklist		
Owner	OwnerId	Lookup(User,Group)		✓
Phone	Phone__c	Phone		
Total Purchases	Total_Purchases__c	Number(18, 0)		

JewelleryStore Product

- **JewelleryStore Product Name** – Name of the jewelry product (e.g., Diamond Ring, Gold).
- **Price** – Selling price of the product.
- **SKU** – Unique identifier/code for the product.
- **Stock Quantity** – Number of items available in inventory.

- **Owner** – Salesforce user or group responsible for managing the product.
- **Created By** – User who created the product record.
- **Last Modified By** – User who last updated the product record.



Fields & Relationships
7 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Price	Price__c	Currency(18, 0)		
Product Name	Name	Text(80)		✓
SKU	SKU__c	Text(60)		
Stock Quantity	Stock_Quantity__c	Number(18, 0)		

JewelleryStore Order

- **Customer** (Lookup **JewelleryStore** Customer)
- **Product** (Lookup → **JewelleryStore** Product)
- **Order Date** (Date)
- **Payment Status** (Picklist: Pending, Paid, Refunded)
- **Delivery Status** (Picklist: Pending, Shipped, Delivered)

JewelleryStore Inventory

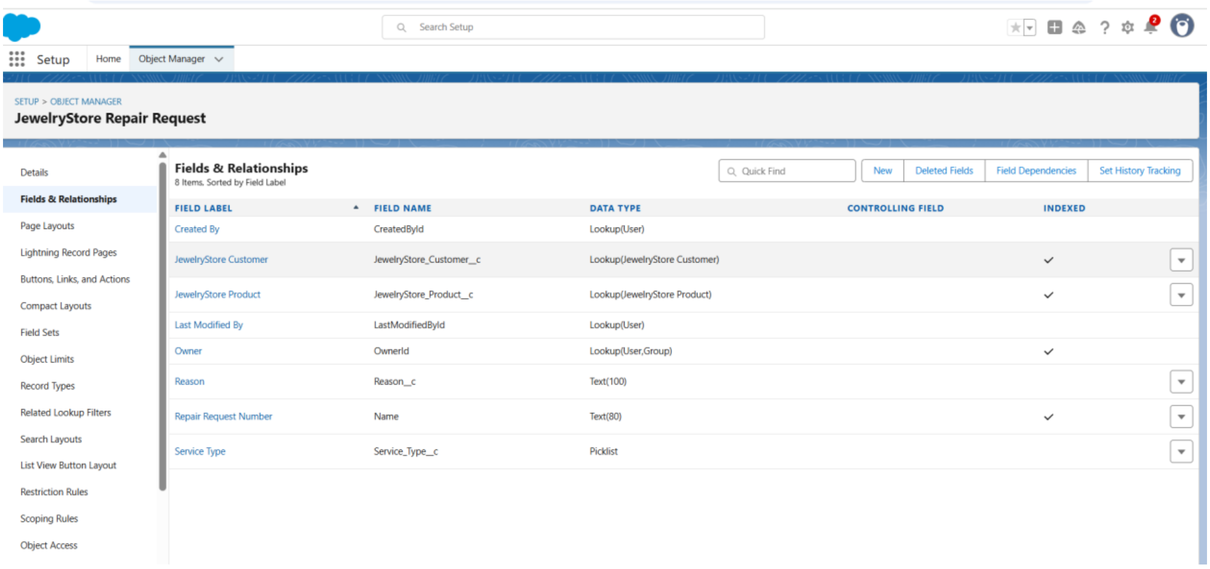
- **Product** (Lookup → **JewelleryStore** Product)
- **Available Quantity** (Number)
- **Reorder Level** (Number)
- **Store Location** (Text)
- **Stock Status** (Formula → IF(Available_Quantity__c > Reorder_Level__c, "Available", "Low Stock"))

JewelleryStore Marketing Campaign

- **Campaign Name** (Text)
- **Start Date** (Date)
- **End Date** (Date)
- **Offer Type** (Picklist: Discount %, Free Gift, Exclusive Access)
- **Campaign Status** (Picklist: Planned, Active, Completed)

JewelleryStore Repair Request

- **Customer** (Lookup → **JewelleryStore** Customer)
- **Product** (Lookup → **JewelleryStore** Product)
- **Request Type** (Picklist: Repair, Cleaning, Resizing, Engraving)
- **Request Status** (Picklist: New, In Progress, Completed)
- **Service Date** (Date)



JewelryStore Repair Request

Fields & Relationships

8 Items, Sorted by Field Label

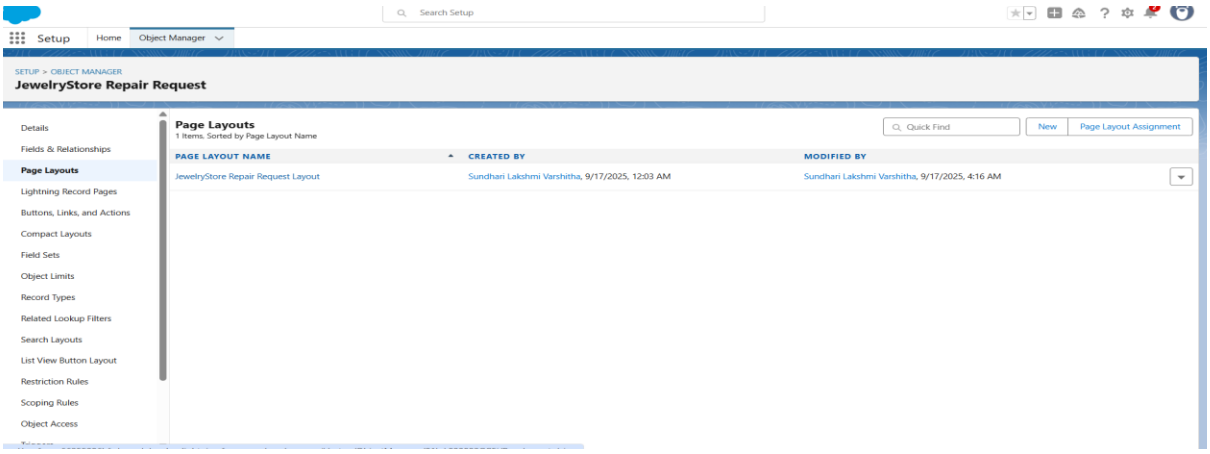
FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
JewelryStore Customer	JewelryStore_Customer__c	Lookup(JewelryStore Customer)		✓
JewelryStore Product	JewelryStore_Product__c	Lookup(JewelryStore Product)		✓
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Reason	Reason__c	Text(100)		
Repair Request Number	Name	Text(80)		✓
Service Type	Service_Type__c	Picklist		

3. Record Types

- Used to **differentiate business processes** if needed.
- Example:
  - **JewelleryStore Repair Request** → Record Types: *Repair Request*, *Cleaning Request*, *Resizing Request*
  - **JewelleryStore Order** → Record Types: *Online Order*, *In-Store Order*

4. Page Layouts

- Each object has **custom page layouts** to show only relevant fields to different users.
- Example:
  - **Sales team** sees Customer Name, Email, Phone, Orders, Total Purchases.
  - **Service team** sees Service Requests, Product details, Request Status.



JewelryStore Repair Request

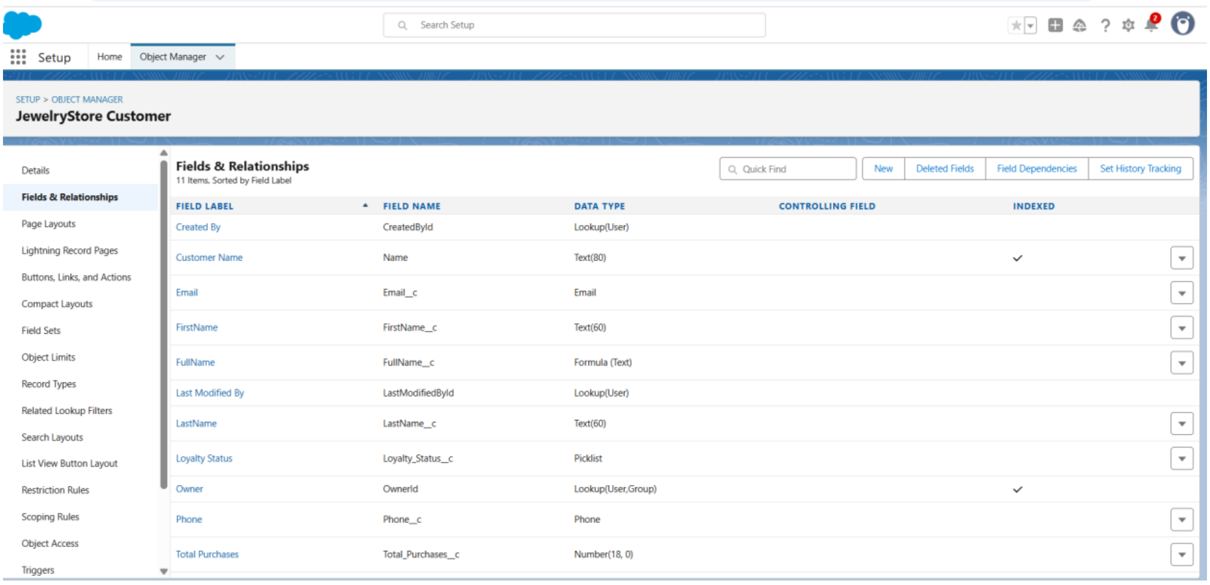
Page Layouts
1 Items, Sorted by Page Layout Name

PAGE LAYOUT NAME	CREATED BY	MODIFIED BY
JewelryStore Repair Request Layout	Sundhari Lakshmi Varshitha, 9/17/2025, 12:03 AM	Sundhari Lakshmi Varshitha, 9/17/2025, 4:16 AM

5. Compact Layouts

- Provides **summary info on record highlights** (used in mobile view or hover cards).
- Example:
  - **JewelleryStore Customer Compact Layout:** Full Name, Email, Phone, Loyalty Status, Total Purchases.
  - **JewelleryStore Product Compact Layout:** Product Name, Price, Stock Quantity.

But I used Compact Layout as System Default.



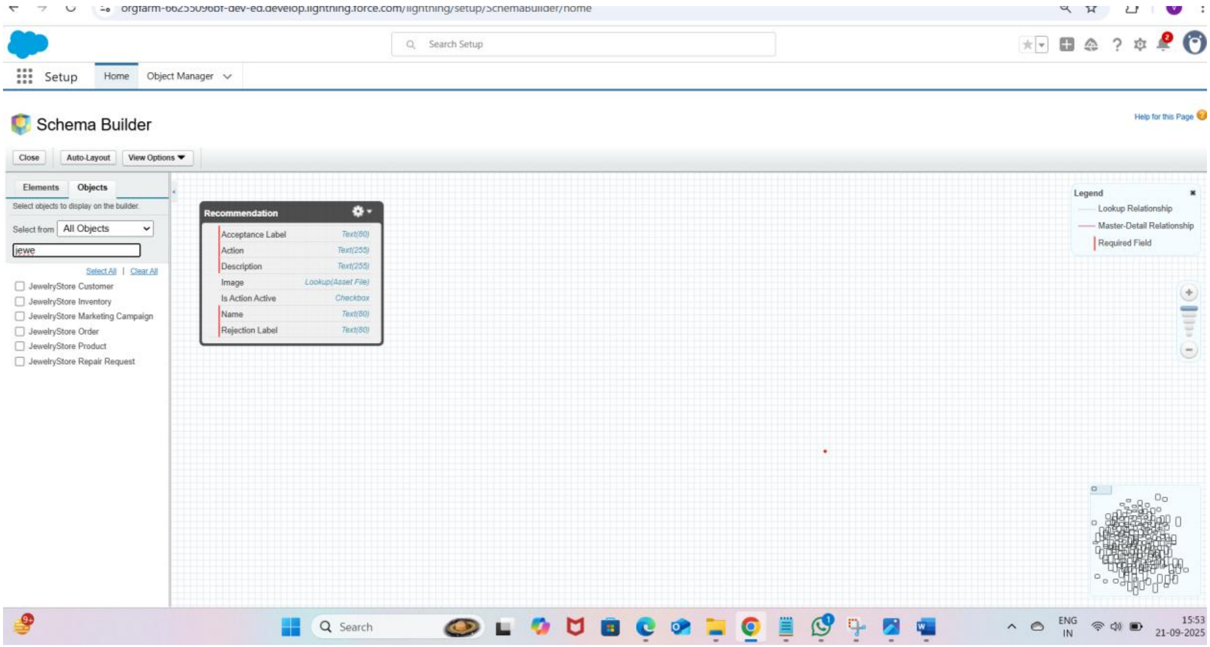
JewelryStore Customer

Fields & Relationships
11 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Customer Name	Name	Text(80)		✓
Email	Email__c	Email		
FirstName	FirstName__c	Text(60)		
FullName	FullName__c	Formula (Text)		
Last Modified By	LastModifiedById	Lookup(User)		
LastName	LastName__c	Text(60)		
Loyalty Status	Loyalty_Status__c	Picklist		
Owner	OwnerId	Lookup(User,Group)		✓
Phone	Phone__c	Phone		
Total Purchases	Total_Purchases__c	Number(18, 0)		

6. Schema Builder

- Used to **visualize all objects and their relationships**.
- Shows **custom and standard objects**, field types, and lookup/master-detail links.
- Helps stakeholders understand **data flow and connections**.



Schema Builder

Select objects to display on the builder.

Select from: All Objects

jewe

Select All | Clear All

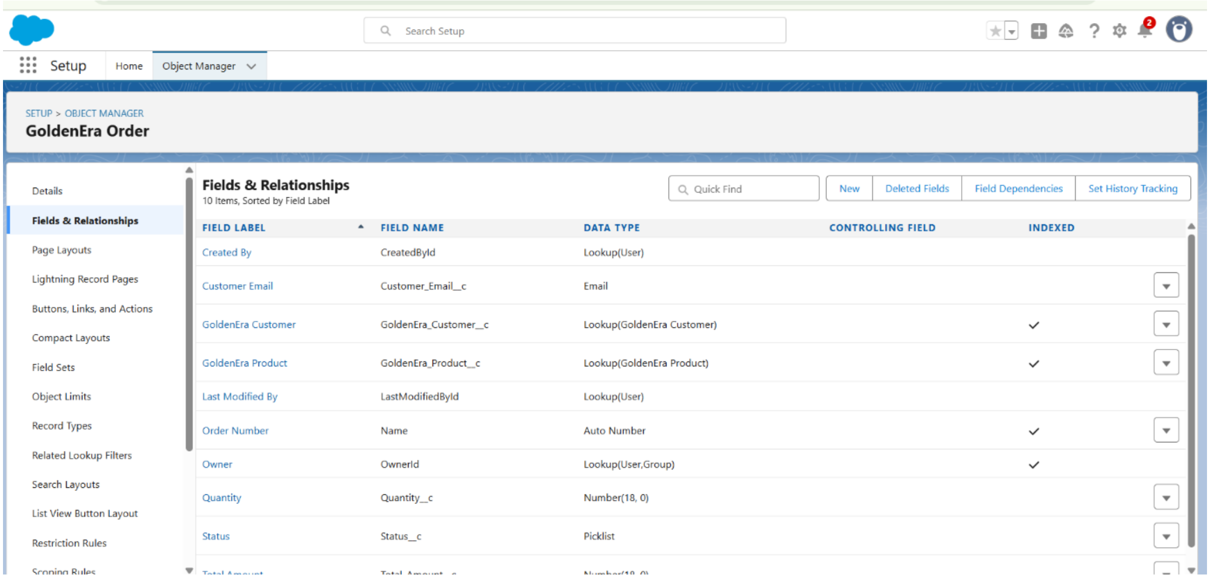
- □ JewelryStore Customer
- □ JewelryStore Inventory
- □ JewelryStore Marketing Campaign
- □ JewelryStore Order
- □ JewelryStore Product
- □ JewelryStore Repair Request

Recommendation

Acceptance Label	Text(80)
Action	Text(255)
Description	Text(255)
Image	Lookup(Asset File)
Is Action Active	Checkbox
Name	Text(80)
Rejection Label	Text(80)

7. Lookup vs Master-Detail vs Hierarchical Relationships

- **Lookup Relationship:**
  - Simple connection between two objects.
  - Example: → JewelleryStore Customer
- **Master-Detail Relationship:**
  - Strong dependency, parent controls child record behavior.
  - Example: JewelleryStore Inventory → Product (Product is parent, Inventory is child)
- **Hierarchical Relationship:**
  - Special relationship for **User object** only.
  - Example: Manager → Employee (not used heavily in this project)



GoldenEra Order

Fields & Relationships

10 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Customer Email	Customer_Email__c	Email		
GoldenEra Customer	GoldenEra_Customer__c	Lookup(GoldenEra Customer)		✓
GoldenEra Product	GoldenEra_Product__c	Lookup(GoldenEra Product)		✓
Last Modified By	LastModifiedById	Lookup(User)		
Order Number	Name	Auto Number		✓
Owner	OwnerId	Lookup(User,Group)		✓
Quantity	Quantity__c	Number(18, 0)		
Status	Status__c	Picklist		
Total Amount	Total_Amount__c	Number(18, 0)		

8. Junction Objects

- Used for **many-to-many relationships**.
- Example (optional for future enhancement):
    - **JewelleryStore Customer ↔ JewelleryStore Marketing Campaign**
    - A single customer can belong to multiple campaigns, and a campaign can target multiple customers.

Phase 4: Process Automation (Admin)

Project Title: Jewellery Store Management CMR

1• Validation Rules

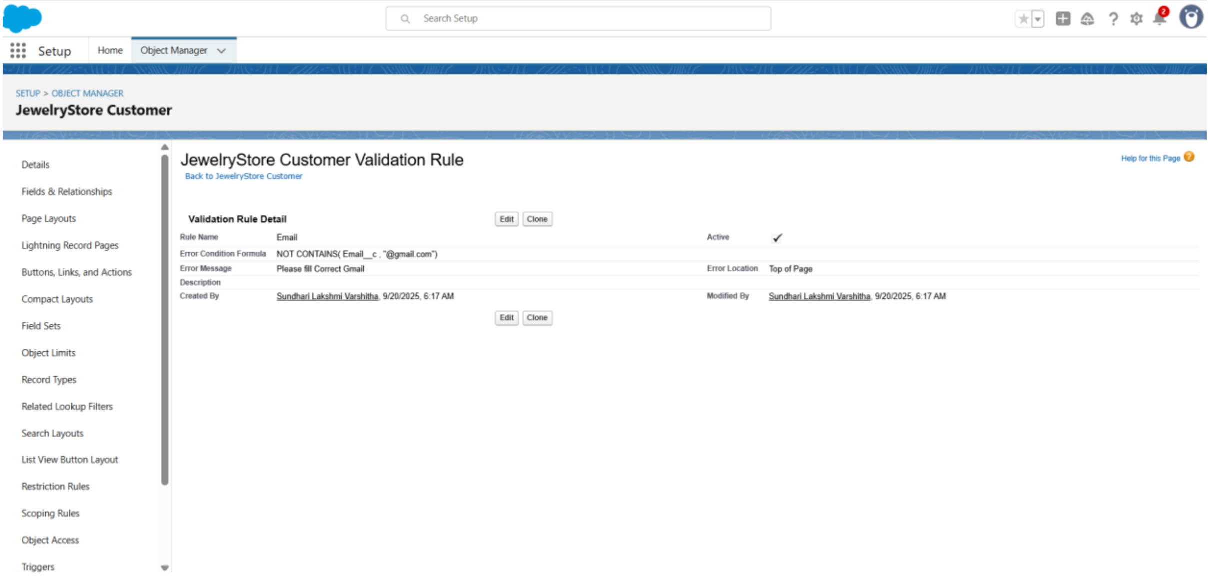
Validation Rules ensure that the data entered into Salesforce records meets specific business criteria. They prevent incorrect or inconsistent data from being saved.

Examples for Jewellery Store:

- JewelleryStore Order__c: Total_Amount__c <= 0
    - Error Message: "Please Enter Correct Amount"
    - Ensures orders have a positive total amount.
- Inventory__c: Stock_Quantity__c <= 0
    - Error Message: "Inventory count cannot be less than zero"
    - Prevents stock from being negative.
- JewelleryStore Customer__c: NOT(CONTAINS(Email, "@gmail.com"))
    - Error Message: "Please fill Correct Gmail"
    - Ensures customer email follows a valid format.

Steps to create a validation rule:

1. Setup → Object Manager → Select the Object → Validation Rules → New.
2. Enter Rule Name, Error Condition Formula, and Error Message.
3. Choose error location (Field or Top of Page) → Save.



JewelryStore Customer Validation Rule

Validation Rule Detail

Rule Name	Email	Active	✓
Error Condition Formula	NOT CONTAINS(Email__c , "@gmail.com")		
Error Message	Please fill Correct Gmail	Error Location	Top of Page
Description			
Created By	Sundhari Lakshmi Varshitha, 9/20/2025, 6:17 AM	Modified By	Sundhari Lakshmi Varshitha, 9/20/2025, 6:17 AM

2. Workflow Rules

Workflow Rules automate standard internal processes. They can trigger Email Alerts, Field Updates, Tasks, or Outbound Messages when record conditions are met.

Example for Jewelry Store:

- When Inventory__c.Stock_Quantity__c < 5, send an email alert to the Inventory Manager.
- When JewelryStore_Order__c.Status__c = Confirmed, update a field Order_Confirmed__c to true.

3.Process Builder

Process Builder allows multi-step automation beyond what workflow rules can do. It can:

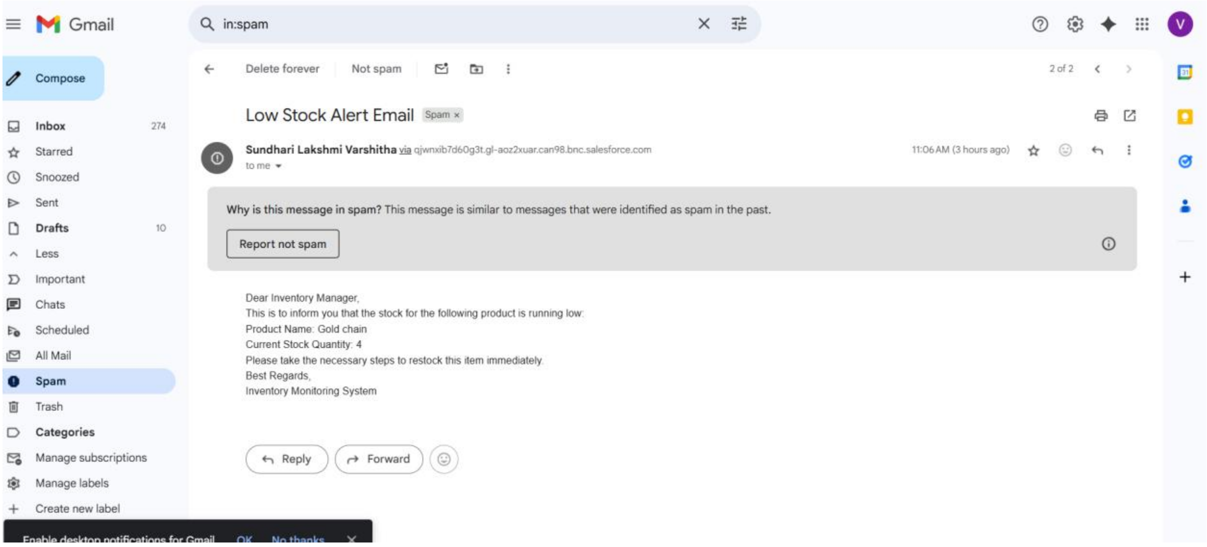
- Update related records
- Post to Chatter
- Launch Flows or Apex
- Send Email Alerts

Example for Jewelry Store:

- When Total_Purchases__c of a customer exceeds 1000, update Loyalty_Status__c to Gold.
- When JewelryStore_Order__c.Status__c = Rejection, notify the Sales team automatically.

Steps:

1. Setup → Process Builder → New → Name your process → Choose Object.
2. Define criteria → Add Immediate or Scheduled Actions.
3. Save & Activate.



Low Stock Alert Email Spam ×

Sundhari Lakshmi Varshitha via qjwnxib7d60g3t.gl-aoz2xuar.can98.bnc.salesforce.com
to me
11:06 AM (3 hours ago)

Why is this message in spam? This message is similar to messages that were identified as spam in the past.

Dear Inventory Manager,
This is to inform you that the stock for the following product is running low:
Product Name: Gold chain
Current Stock Quantity: 4
Please take the necessary steps to restock this item immediately.
Best Regards,
Inventory Monitoring System

4. Approval Process

Approval Processes manage record approvals in stages.
Examples:

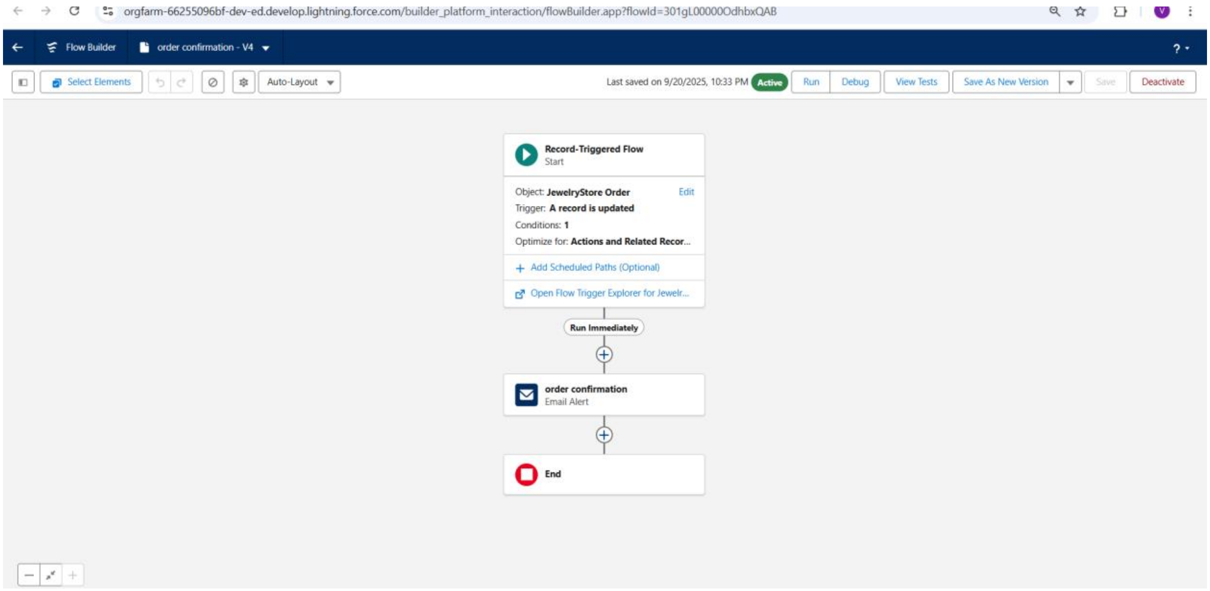
- Orders above a certain amount require manager approval.
- Automated actions: update order status, send notification emails to approvers.

Useful for enforcing business policies and checks.

5.Flow Builder

Flow Implementations

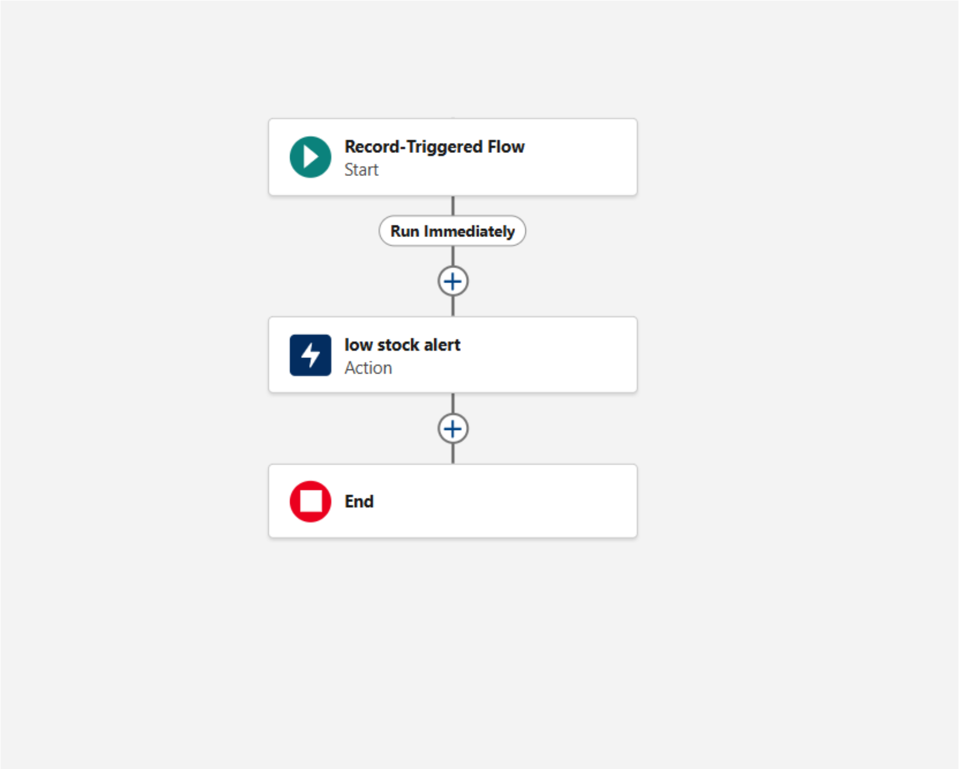
a. **Order Confirmation Flow**
• Triggered when an order is updated to Confirmed. • Sends an Order Confirmation email to the related customer.


orgfarm-66255096bf-dev-ed.develop.lightning.force.com/builder_platform_interaction/flowBuilder.app?flowId=301gL00000OdhbxQAB
Flow Builder
order confirmation - V4
Select Elements
Auto-Layout
Last saved on 9/20/2025, 10:33 PM
Active
Run
Debug
View Tests
Save As New Version
Save
Deactivate
Record-Triggered Flow
Start
Edit
Add Scheduled Paths (Optional)
Open Flow Trigger Explorer for Jewelr...
Run Immediately
order confirmation
Email Alert
End

Object: JewelryStore Order
Trigger: A record is updated
Conditions: 1
Optimize for: Actions and Related Recor...

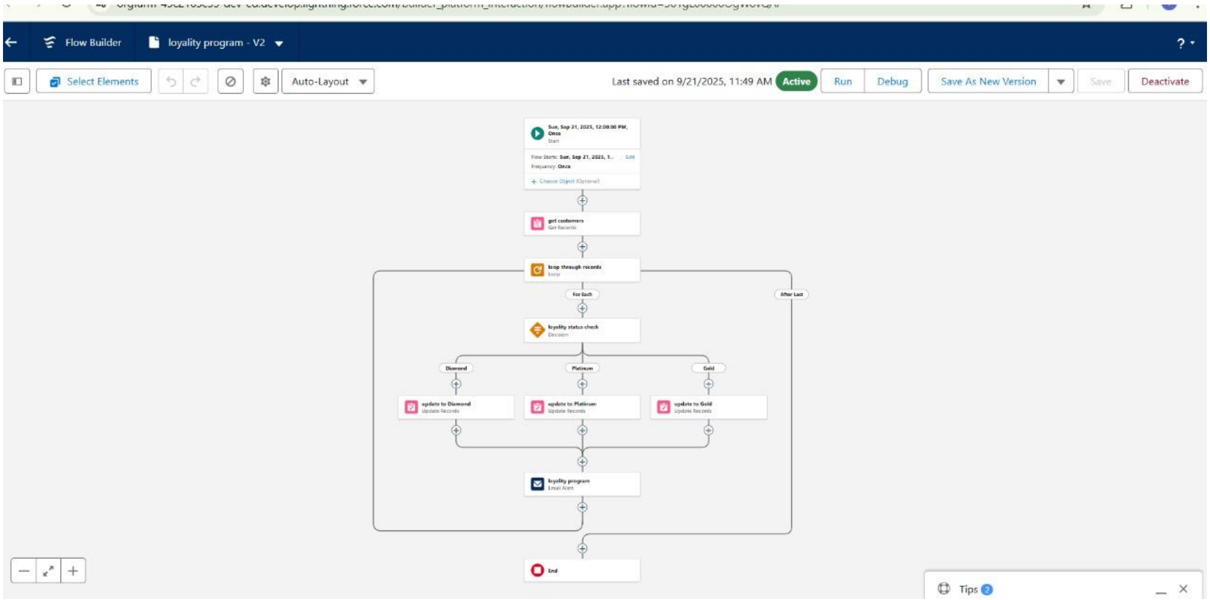
b. Stock Alert Flow

• Triggered when Inventory stock drops below 5. • Sends Low Stock email to Inventory Manager.



c. Scheduled Flow:

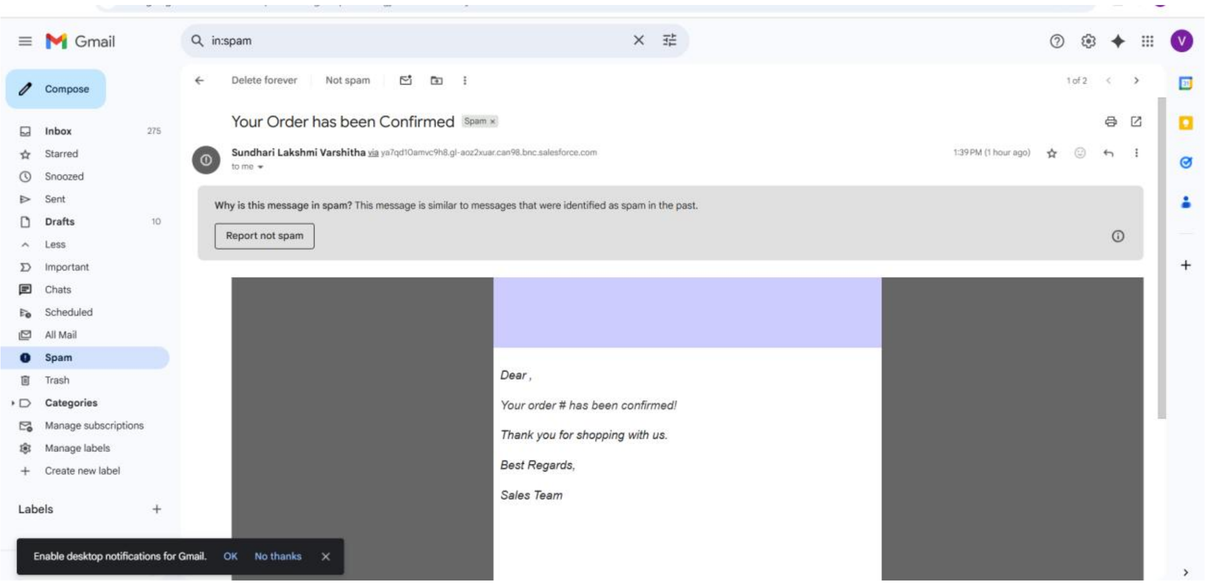
Loyalty Update • Runs daily at midnight. • Loops through customers and updates their Loyalty Status based on total purchases.



6. Email Alerts

Send automated emails based on workflows, processes, or flows.
Examples:

- Order Confirmation Email to customer.



Your Order has been Confirmed Spam ×

Sundhari Lakshmi Varshitha via ya7qd10amvc9h8.gl-aoz2xuar.can98.bnc.salesforce.com
to me
1:39 PM (1 hour ago)

Why is this message in spam? This message is similar to messages that were identified as spam in the past.

Dear ,

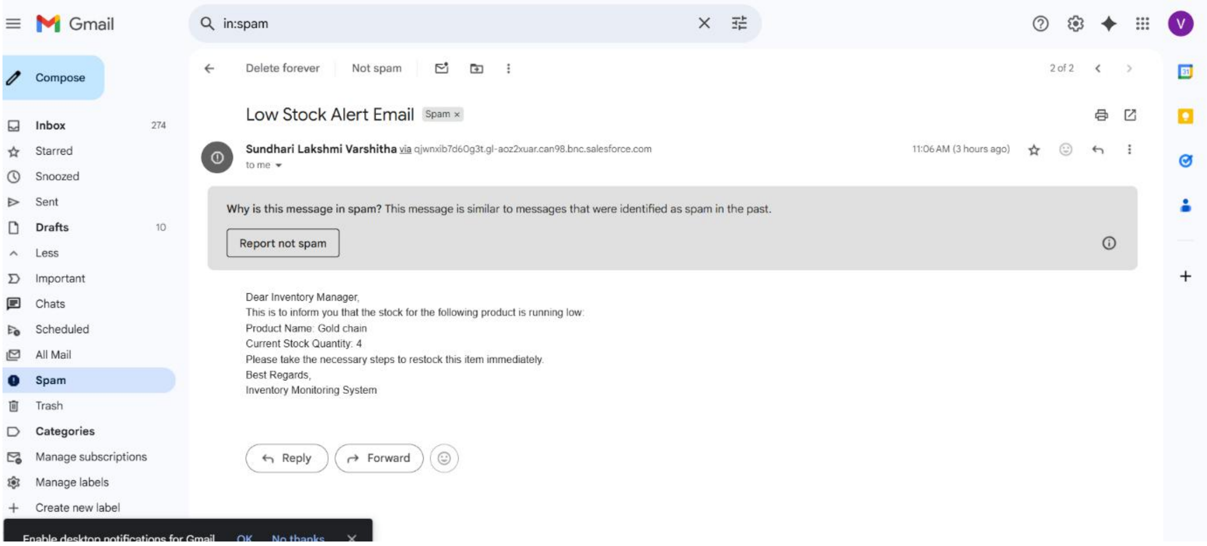
Your order # has been confirmed!

Thank you for shopping with us.

Best Regards,

Sales Team

- Low Stock Alert to Inventory Manager.



Low Stock Alert Email Spam ×

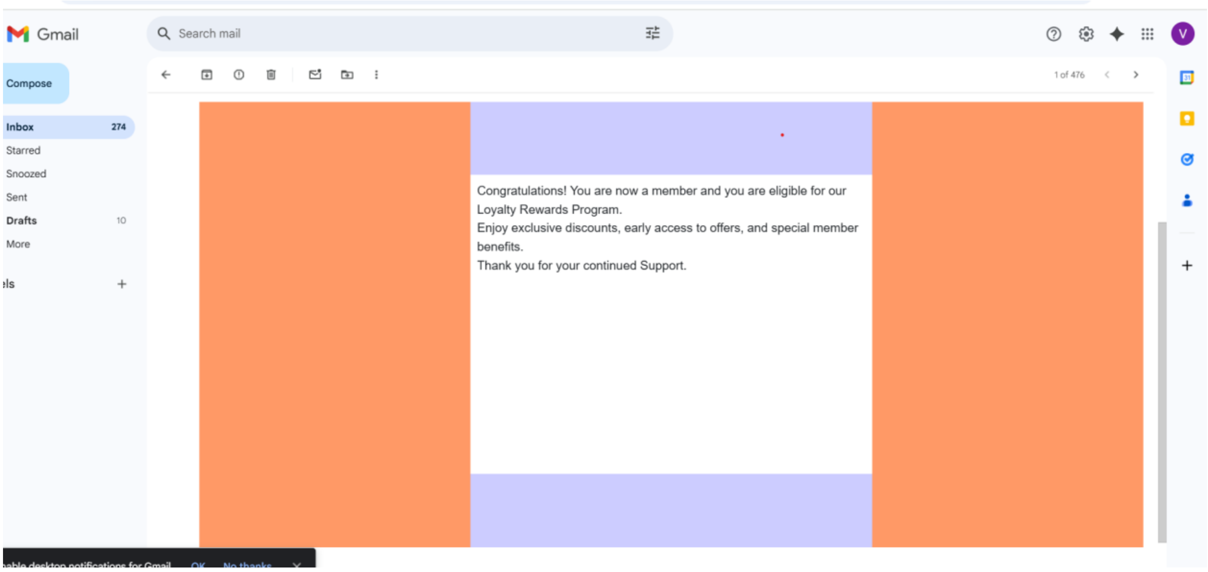
Sundhari Lakshmi Varshitha via qjwnxib7d60g3t.gl-aoz2xuar.can98.bnc.salesforce.com
to me
11:06 AM (3 hours ago)

Why is this message in spam? This message is similar to messages that were identified as spam in the past.

Dear Inventory Manager,
This is to inform you that the stock for the following product is running low:
Product Name: Gold chain
Current Stock Quantity: 4
Please take the necessary steps to restock this item immediately.
Best Regards,
Inventory Monitoring System

-

- Loyalty Program Email to qualifying customers.

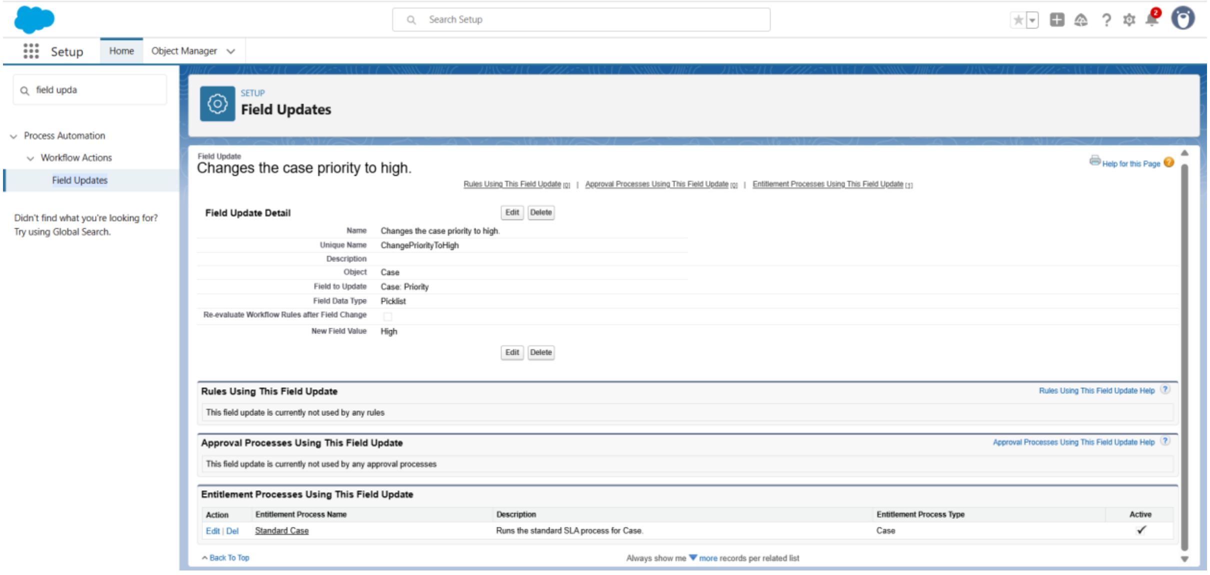


Congratulations! You are now a member and you are eligible for our Loyalty Rewards Program.
Enjoy exclusive discounts, early access to offers, and special member benefits.
Thank you for your continued Support.

7. Field Updates

Automatically update field values when criteria are met.
Examples:

- Mark Order_Confirmed__c = true when an order is confirmed.
- Update loyalty status based on total purchases.



Field Update
Changes the case priority to high.

Rules Using This Field Update [0] | Approval Processes Using This Field Update [0] | Entitlement Processes Using This Field Update [1]

Field Update Detail

Name	Changes the case priority to high.
Unique Name	ChangePriorityToHigh
Description	
Object	Case
Field to Update	Case: Priority
Field Data Type	Picklist
Re-evaluate Workflow Rules after Field Change	□
New Field Value	High

Rules Using This Field Update — Rules Using This Field Update Help

This field update is currently not used by any rules

Approval Processes Using This Field Update — Approval Processes Using This Field Update Help

This field update is currently not used by any approval processes

Entitlement Processes Using This Field Update

Action	Entitlement Process Name	Description	Entitlement Process Type	Active
Edit | Del	Standard Case	Runs the standard SLA process for Case.	Case	✓

Always show me more records per related list

8. Custom Notifications

Send real-time notifications to users on desktop or mobile.
Examples:

- Alert sales managers when high-value orders are placed.
- Alert inventory managers when stock is critically low.

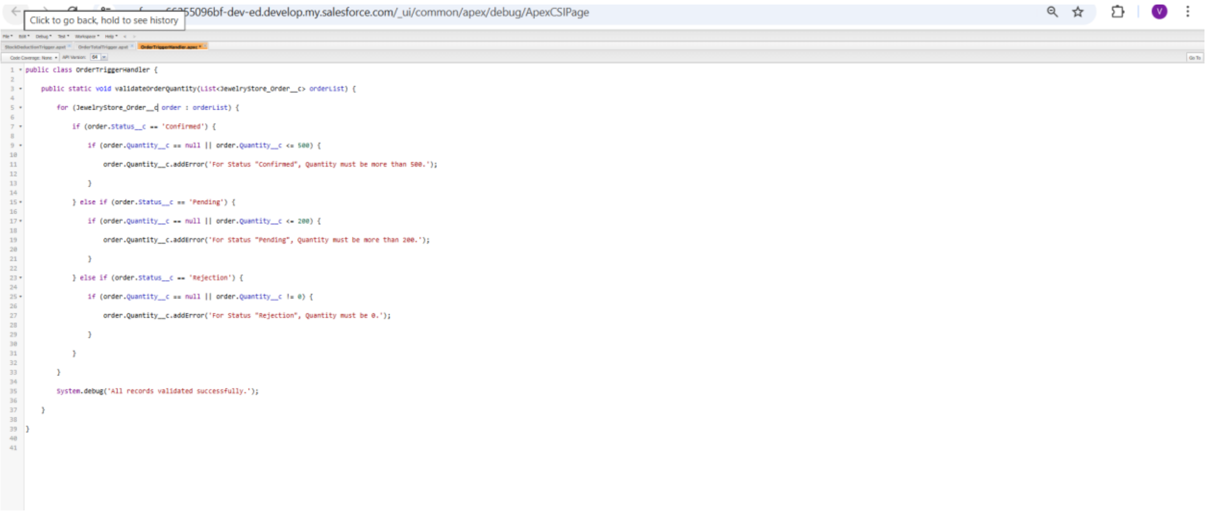
Phase 5: Apex Programming (Developer)

Project:Jewellery Store Management CRM

Step 1: Classes & Objects

□ **Apex is object-oriented**: It allows the creation of **classes** (blueprints) and **objects** (instances) to organize and reuse business logic.

□ In this project, a **Trigger Handler Class** approach was implemented to follow best practices.



Click to go back, hold to see history
…55096bf-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
OrderTriggerHandler.apxc *

```
public class OrderTriggerHandler {

    public static void validateOrderQuantity(List<JewelryStore_Order__c> orderList) {

        for (JewelryStore_Order__c order : orderList) {

            if (order.Status__c == 'Confirmed') {

                if (order.Quantity__c == null || order.Quantity__c <= 500) {

                    order.Quantity__c.addError('For Status "Confirmed", Quantity must be more than 500.');

                }

            } else if (order.Status__c == 'Pending') {

                if (order.Quantity__c == null || order.Quantity__c <= 200) {

                    order.Quantity__c.addError('For Status "Pending", Quantity must be more than 200.');

                }

            } else if (order.Status__c == 'Rejection') {

                if (order.Quantity__c == null || order.Quantity__c != 0) {

                    order.Quantity__c.addError('For Status "Rejection", Quantity must be 0.');

                }

            }

        }

        System.debug('All records validated successfully.');

    }

}
```

Source Code:

```
public class OrderTriggerHandler {

  public static void validateOrderQuantity(List<JewelryStore_Order__c> orderList) {

    for (JewelryStore_Order__c order : orderList) {

      if (order.Status__c == 'Confirmed') {

        if (order.Quantity__c == null || order.Quantity__c <= 500) {

          order.Quantity__c.addError('For Status "Confirmed", Quantity must be more than 500.');

        }

      } else if (order.Status__c == 'Pending') {

        if (order.Quantity__c == null || order.Quantity__c <= 200) {

       order.Quantity__c.addError('For Status "Pending", Quantity must be more than 200.');

       }

      } else if (order.Status__c == 'Rejection') {
```

```
                if (order.Quantity__c == null || order.Quantity__c != 0) {
                    order.Quantity__c.addError('For Status "Rejection", Quantity must be 0.');
                }
            }
        }
        System.debug('All records validated successfully.');
    }
}
```

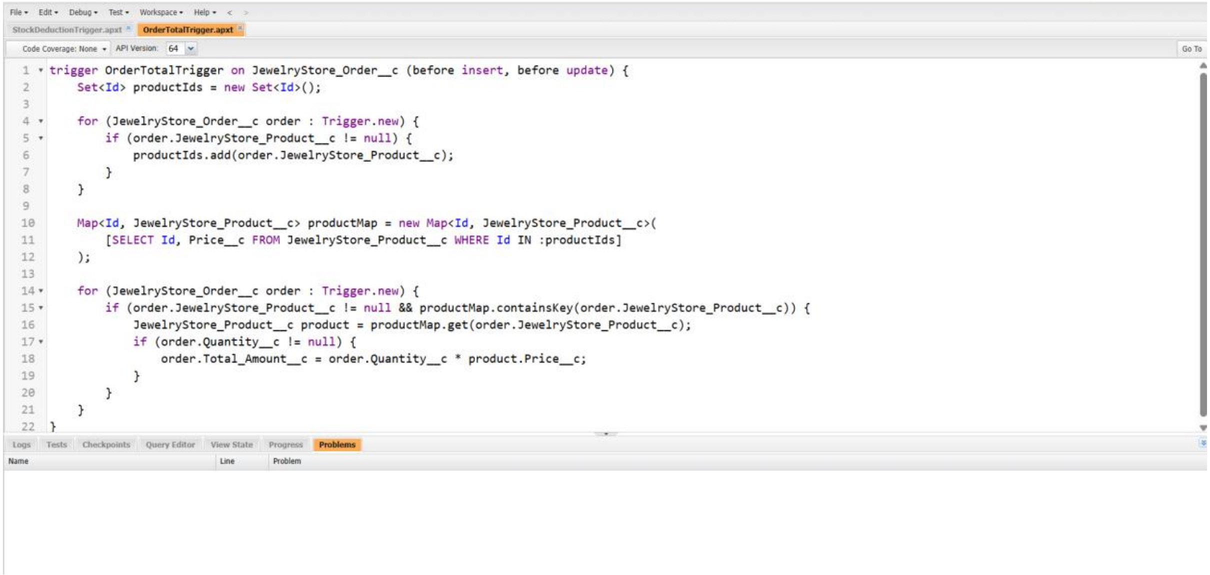
Step 2:Apex Triggers (before/after insert/update/delete)

In this project, two custom Apex triggers were implemented to automate critical processes in the Golden Era Enterprises CRM:

a) Order Total Trigger

Purpose:

- Automatically calculate the **Total Price** for each order.
- Formula: **Quantity** × **Product Price**



```
trigger OrderTotalTrigger on JewelryStore_Order__c (before insert, before update) {
    Set<Id> productIds = new Set<Id>();

    for (JewelryStore_Order__c order : Trigger.new) {
        if (order.JewelryStore_Product__c != null) {
            productIds.add(order.JewelryStore_Product__c);
        }
    }

    Map<Id, JewelryStore_Product__c> productMap = new Map<Id, JewelryStore_Product__c>(
        [SELECT Id, Price__c FROM JewelryStore_Product__c WHERE Id IN :productIds]
    );

    for (JewelryStore_Order__c order : Trigger.new) {
        if (order.JewelryStore_Product__c != null && productMap.containsKey(order.JewelryStore_Product__c)) {
            JewelryStore_Product__c product = productMap.get(order.JewelryStore_Product__c);
            if (order.Quantity__c != null) {
                order.Total_Amount__c = order.Quantity__c * product.Price__c;
            }
        }
    }
}
```

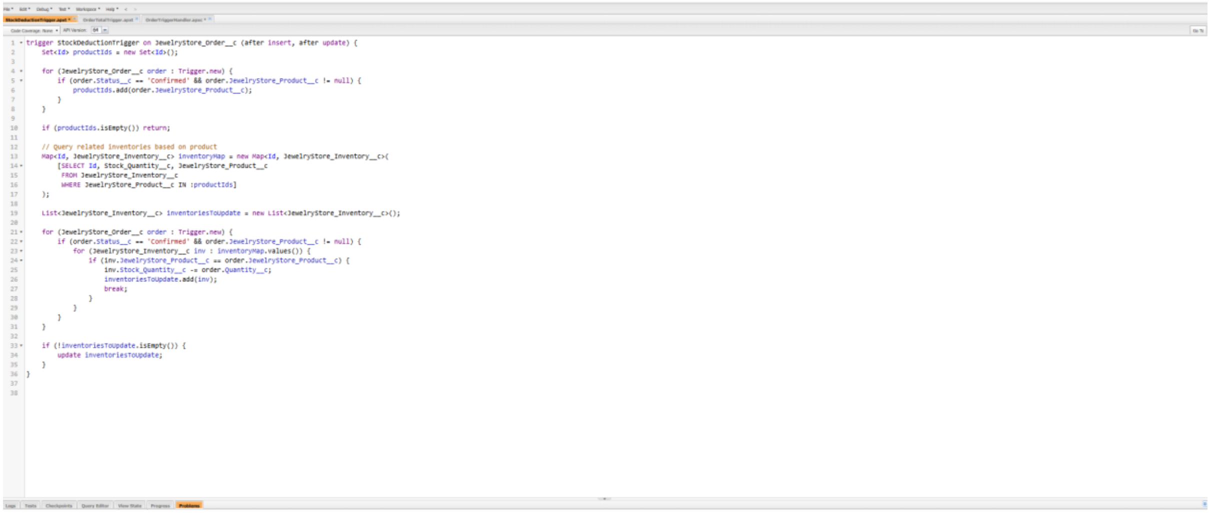
Source Code:

```
trigger OrderTotalTrigger on JewelryStore_Order__c (before insert, before update) {
    Set<Id> productIds = new Set<Id>();
    for (JewelryStore_Order__c order : Trigger.new) {
        if (order.JewelryStore_Product__c != null) {
            productIds.add(order.JewelryStore_Product__c);
        }
    }
    Map<Id, JewelryStore_Product__c> productMap = new Map<Id, JewelryStore_Product__c>(
        [SELECT Id, Price__c FROM JewelryStore_Product__c WHERE Id IN :productIds]
    );
    for (JewelryStore_Order__c order : Trigger.new) {
        if (order.JewelryStore_Product__c != null && productMap.containsKey(order.JewelryStore_Product__c)) {
            JewelryStore_Product__c product = productMap.get(order.JewelryStore_Product__c);
            if (order.Quantity__c != null) {
                order.Total_Amount__c = order.Quantity__c * product.Price__c;
            }
        }
    }
}
```

b) Stock Deduction Trigger

Purpose:

- Deduct stock from **Inventory** (or Product Stock Quantity) based on the order quantity.
- Example: If customer orders 2 rings, Inventory decreases by 2.



```
trigger StockDeductionTrigger on JewelryStore_Order__c (after insert, after update) {
    Set<Id> productIds = new Set<Id>();

    for (JewelryStore_Order__c order : Trigger.new) {
        if (order.Status__c == 'Confirmed' && order.JewelryStore_Product__c != null) {
            productIds.add(order.JewelryStore_Product__c);
        }
    }

    if (productIds.isEmpty()) return;

    // Query related inventories based on product
    Map<Id, JewelryStore_Inventory__c> inventoryMap = new Map<Id, JewelryStore_Inventory__c>(
        [SELECT Id, Stock_Quantity__c, JewelryStore_Product__c
         FROM JewelryStore_Inventory__c
         WHERE JewelryStore_Product__c IN :productIds]
    );

    List<JewelryStore_Inventory__c> inventoriesToUpdate = new List<JewelryStore_Inventory__c>();

    for (JewelryStore_Order__c order : Trigger.new) {
        if (order.Status__c == 'Confirmed' && order.JewelryStore_Product__c != null) {
            for (JewelryStore_Inventory__c inv : inventoryMap.values()) {
                if (inv.JewelryStore_Product__c == order.JewelryStore_Product__c) {
                    inv.Stock_Quantity__c -= order.Quantity__c;
                    inventoriesToUpdate.add(inv);
                    break;
                }
            }
        }
    }

    if (!inventoriesToUpdate.isEmpty()) {
        update inventoriesToUpdate;
    }
}
```

Source Code:

```
trigger StockDeductionTrigger on JewelryStore_Order__c (after insert, after update) {

   Set<Id> productIds = new Set<Id>();

   for (JewelryStore_Order__c order : Trigger.new) {

      if (order.Status__c == 'Confirmed' && order.JewelryStore_Product__c != null) {

         productIds.add(order.JewelryStore_Product__c);

      }

   }

   if (productIds.isEmpty()) return;

   // Query related inventories based on product

   Map<Id, JewelryStore_Inventory__c> inventoryMap = new Map<Id,
JewelryStore_Inventory__c>(

      [SELECT Id, Stock_Quantity__c, JewelryStore_Product__c

       FROM JewelryStore_Inventory__c

       WHERE JewelryStore_Product__c IN :productIds]
```

```
  );

  List<JewelryStore_Inventory__c> inventoriesToUpdate = new
List<JewelryStore_Inventory__c>();

  for (JewelryStore_Order__c order : Trigger.new) {

    if (order.Status__c == 'Confirmed' && order.JewelryStore_Product__c != null) {

      for (JewelryStore_Inventory__c inv : inventoryMap.values()) {

        if (inv.JewelryStore_Product__c == order.JewelryStore_Product__c) {

          inv.Stock_Quantity__c -= order.Quantity__c;

          inventoriesToUpdate.add(inv);

          break;

        }

      }

    }

  }

  if (!inventoriesToUpdate.isEmpty()) {

    update inventoriesToUpdate;

  }

}
```

Phase 6: User Interface Development (Jewelry Store Management)

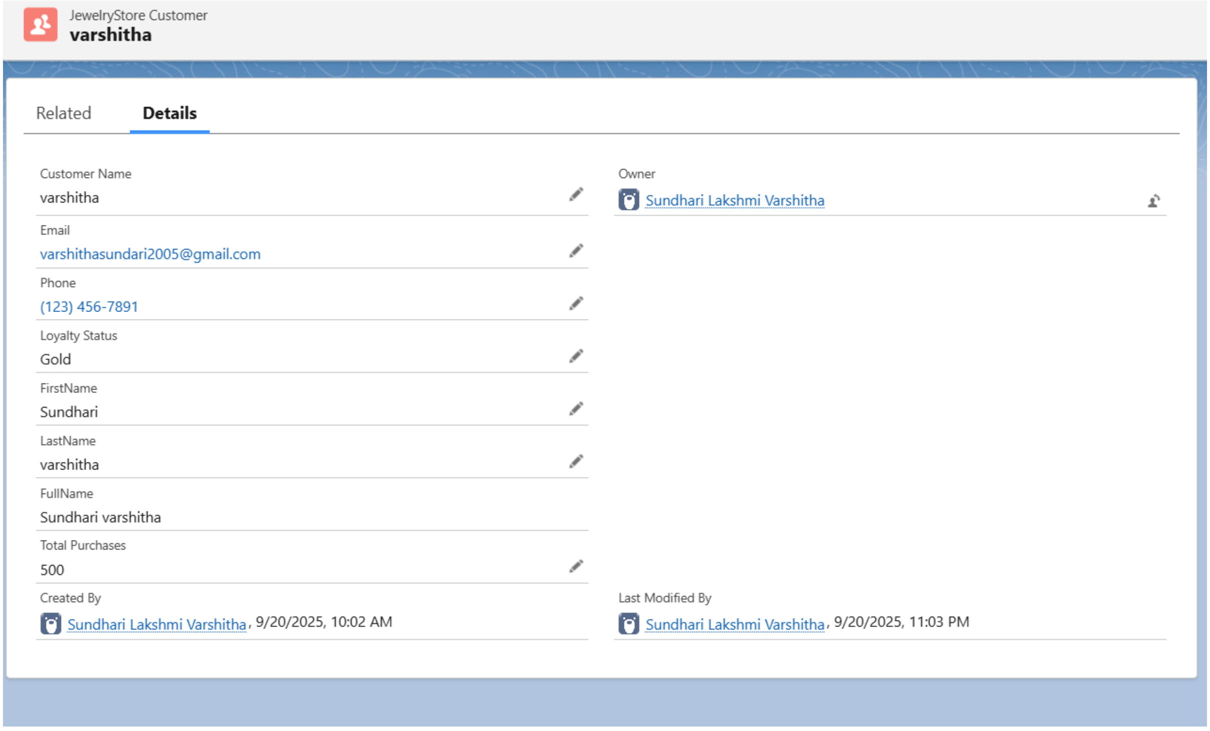
This phase focuses on creating a user-friendly interface that allows jewelry store employees, managers, and customers to easily interact with Salesforce.

1. Lightning App Builder

- A drag-and-drop tool used to build custom applications without writing code.
- Admins can design apps by combining standard and custom components.
- **Jewelry Store Example:** Build a “Jewelry Store Management App” that combines customers, orders, inventory, and loyalty programs into one central app for staff.

2. Record Pages

- Custom layouts for specific object records (Customer, Order, Inventory, Product).
- You can decide what fields, related lists, and components appear.
- **Jewelry Store Example:**
  - **Customer Record Page:** Show customer details, loyalty points, past purchases.



JewelryStore Customer
varshitha

Customer Name
varshitha

Email
varshithasundari2005@gmail.com

Phone
(123) 456-7891

Loyalty Status
Gold

FirstName
Sundhari

LastName
varshitha

FullName
Sundhari varshitha

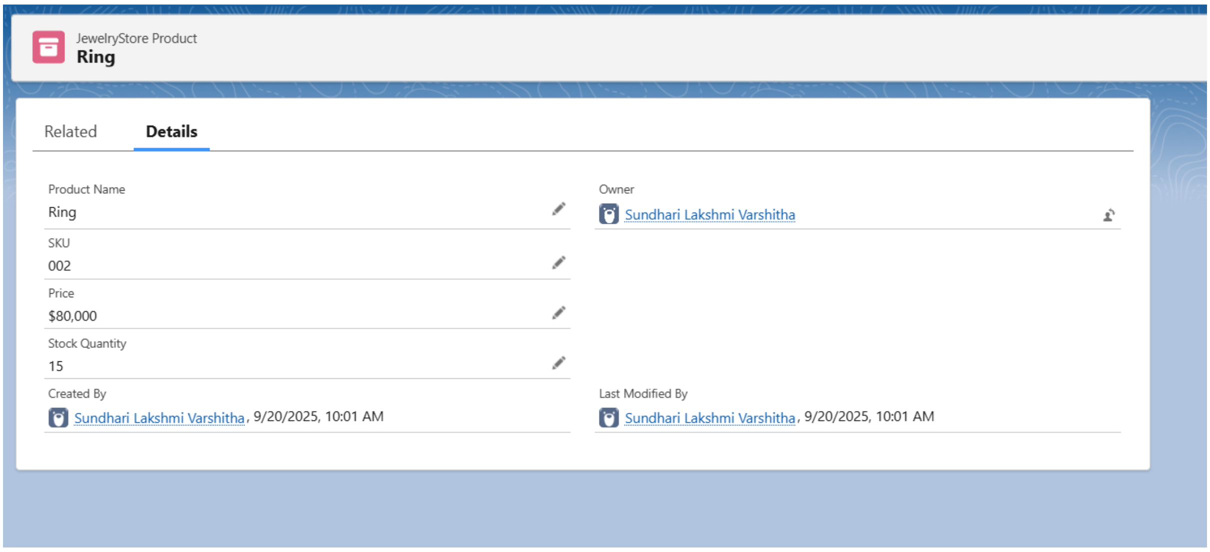
Total Purchases
500

Created By
Sundhari Lakshmi Varshitha, 9/20/2025, 10:02 AM

Owner
Sundhari Lakshmi Varshitha

Last Modified By
Sundhari Lakshmi Varshitha, 9/20/2025, 11:03 PM

  - **Product Record Page:** Show product details, stock quantity, and pricing.



JewelryStore Product
Ring

Product Name
Ring

SKU
002

Price
$80,000

Stock Quantity
15

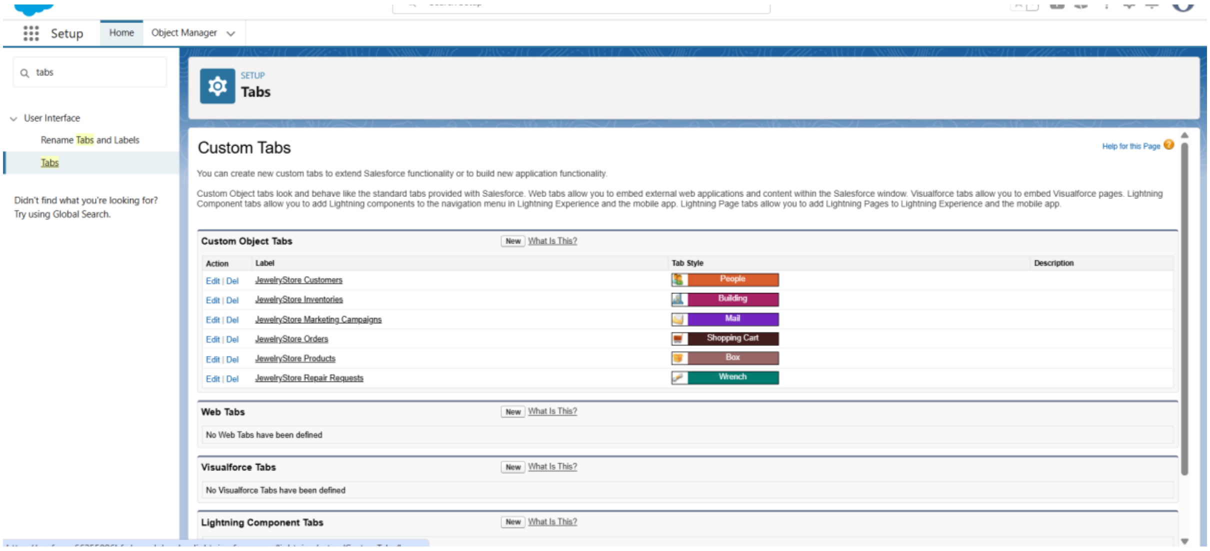
Created By
Sundhari Lakshmi Varshitha, 9/20/2025, 10:01 AM

Owner
Sundhari Lakshmi Varshitha

Last Modified By
Sundhari Lakshmi Varshitha, 9/20/2025, 10:01 AM

3. Tabs

- Tabs allow easy navigation across different objects in Salesforce.
- **Jewelry Store Example:** Create separate tabs for Customers, Orders, Products, Inventory, and Loyalty Programs so store staff can quickly switch between them.



Custom Tabs

You can create new custom tabs to extend Salesforce functionality or to build new application functionality.

Custom Object tabs look and behave like the standard tabs provided with Salesforce. Web tabs allow you to embed external web applications and content within the Salesforce window. Visualforce tabs allow you to embed Visualforce pages. Lightning Component tabs allow you to add Lightning components to the navigation menu in Lightning Experience and the mobile app. Lightning Page tabs allow you to add Lightning Pages to Lightning Experience and the mobile app.

Custom Object Tabs New What Is This?

Action	Label	Tab Style	Description
Edit | Del	JewelryStore Customers	People	
Edit | Del	JewelryStore Inventories	Building	
Edit | Del	JewelryStore Marketing Campaigns	Mail	
Edit | Del	JewelryStore Orders	Shopping Cart	
Edit | Del	JewelryStore Products	Box	
Edit | Del	JewelryStore Repair Requests	Wrench	

Web Tabs New What Is This?

No Web Tabs have been defined

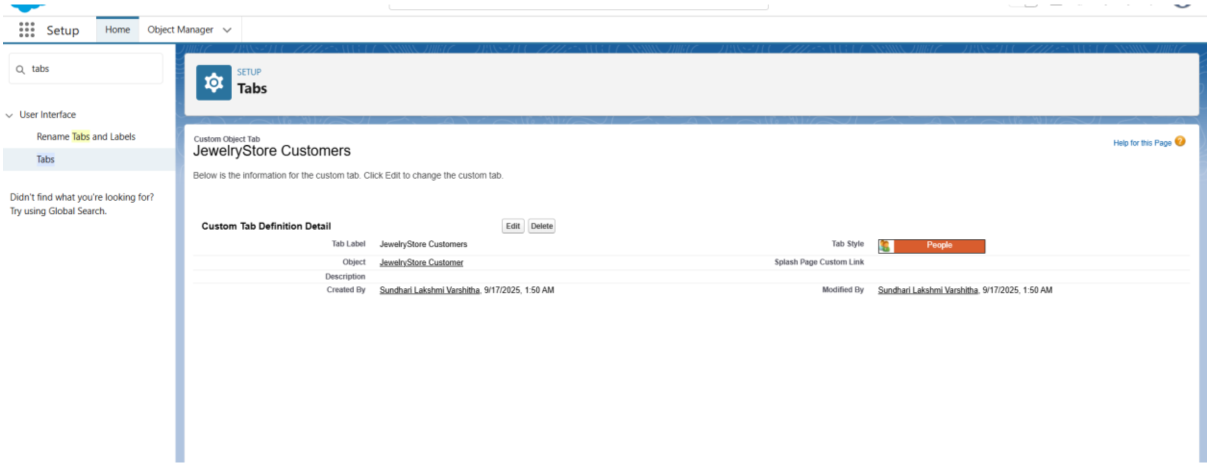
Visualforce Tabs New What Is This?

No Visualforce Tabs have been defined

Lightning Component Tabs New What Is This?

JewelryStore Customers:

This tab allows staff to view, create, and manage customer records, including details such as name, email, phone number, and loyalty status.



Custom Object Tab
JewelryStore Customers

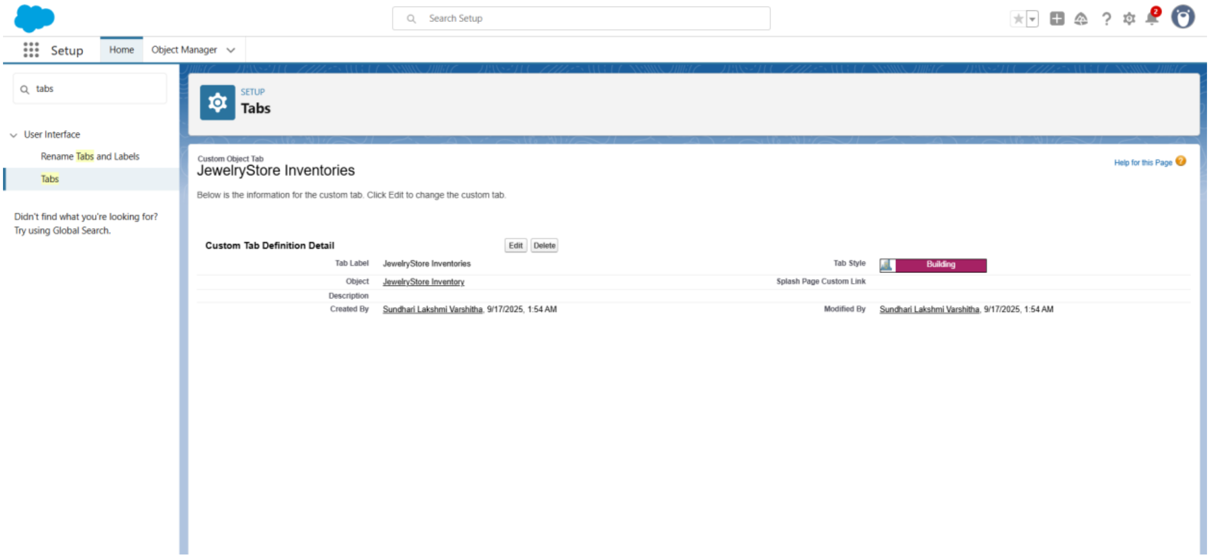
Below is the information for the custom tab. Click Edit to change the custom tab.

Custom Tab Definition Detail Edit Delete

Tab Label	JewelryStore Customers	Tab Style	People
Object	JewelryStore Customer	Splash Page Custom Link	
Description			
Created By	Sundhari Lakshmi Varshitha, 9/17/2025, 1:50 AM	Modified By	Sundhari Lakshmi Varshitha, 9/17/2025, 1:50 AM

JewelryStore Inventories:

Staff can use this tab to check and manage the stock levels of jewelry products, helping to prevent stock shortages and ensuring smooth operations.



Custom Object Tab

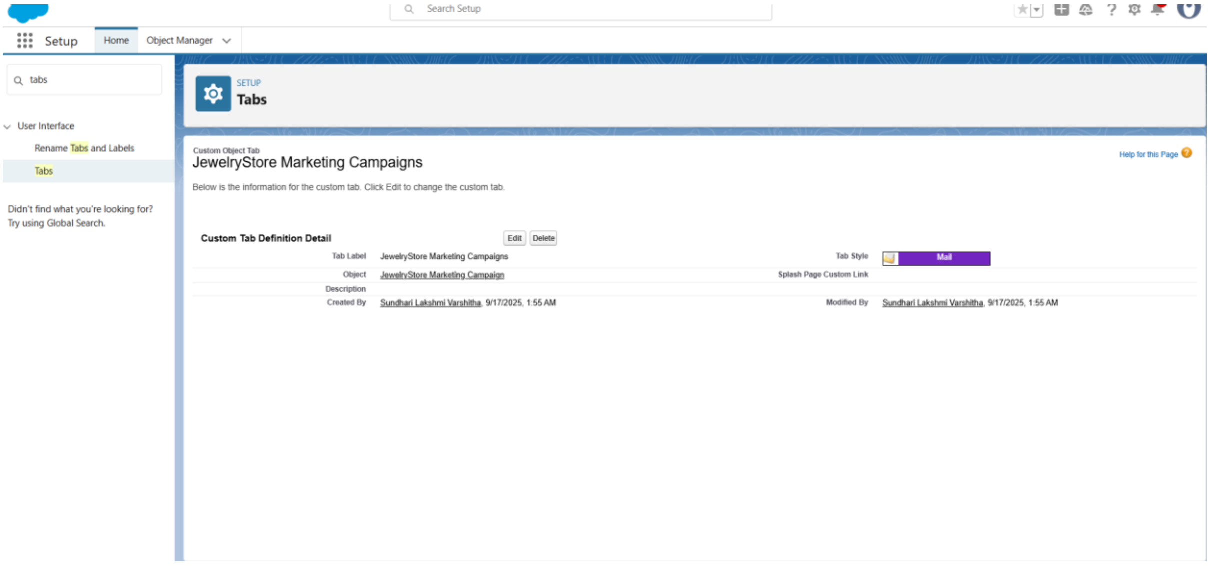
JewelryStore Inventories

Below is the information for the custom tab. Click Edit to change the custom tab.

Custom Tab Definition Detail

Tab Label	JewelryStore Inventories	Tab Style	Building
Object	JewelryStore Inventory	Splash Page Custom Link	
Description			
Created By	Sundhari Lakshmi Varshitha, 9/17/2025, 1:54 AM	Modified By	Sundhari Lakshmi Varshitha, 9/17/2025, 1:54 AM

JewelryStore Marketing Campaigns: This tab is used by the marketing team to manage promotional campaigns, track their progress, and send loyalty program emails to customers.



Custom Object Tab

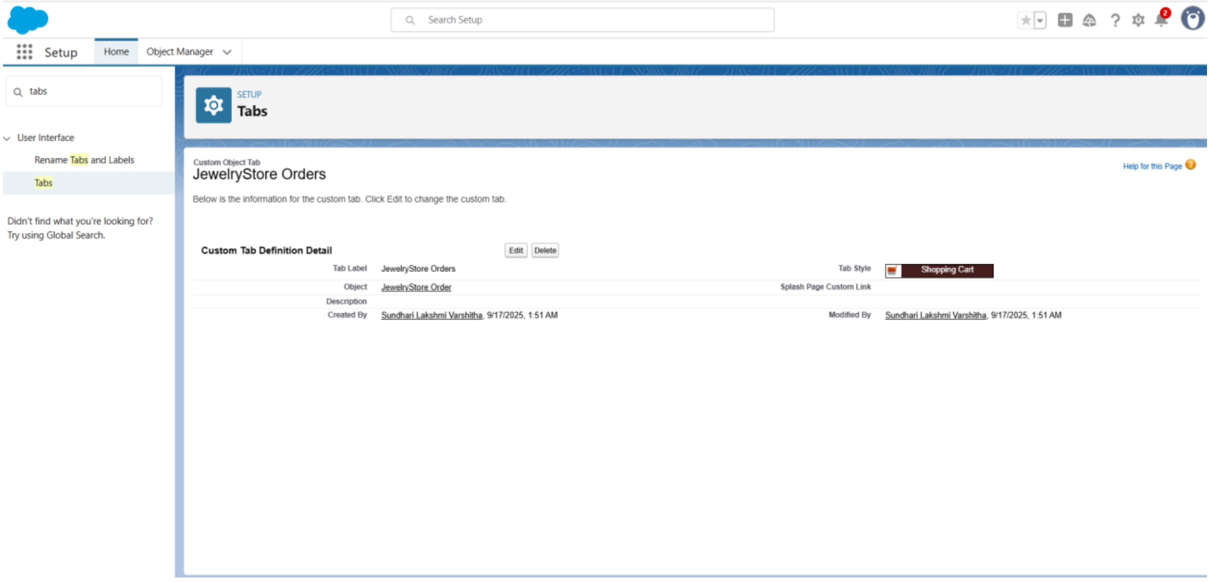
JewelryStore Marketing Campaigns

Below is the information for the custom tab. Click Edit to change the custom tab.

Custom Tab Definition Detail

Tab Label	JewelryStore Marketing Campaigns	Tab Style	Mail
Object	JewelryStore Marketing Campaign	Splash Page Custom Link	
Description			
Created By	Sundhari Lakshmi Varshitha, 9/17/2025, 1:55 AM	Modified By	Sundhari Lakshmi Varshitha, 9/17/2025, 1:55 AM

JewelryStore Orders: Through this tab, sales staff can track all customer orders, including Pending, Confirmed, and Rejected statuses. It also allows easy creation and management of new orders.



Custom Object Tab

JewelryStore Orders

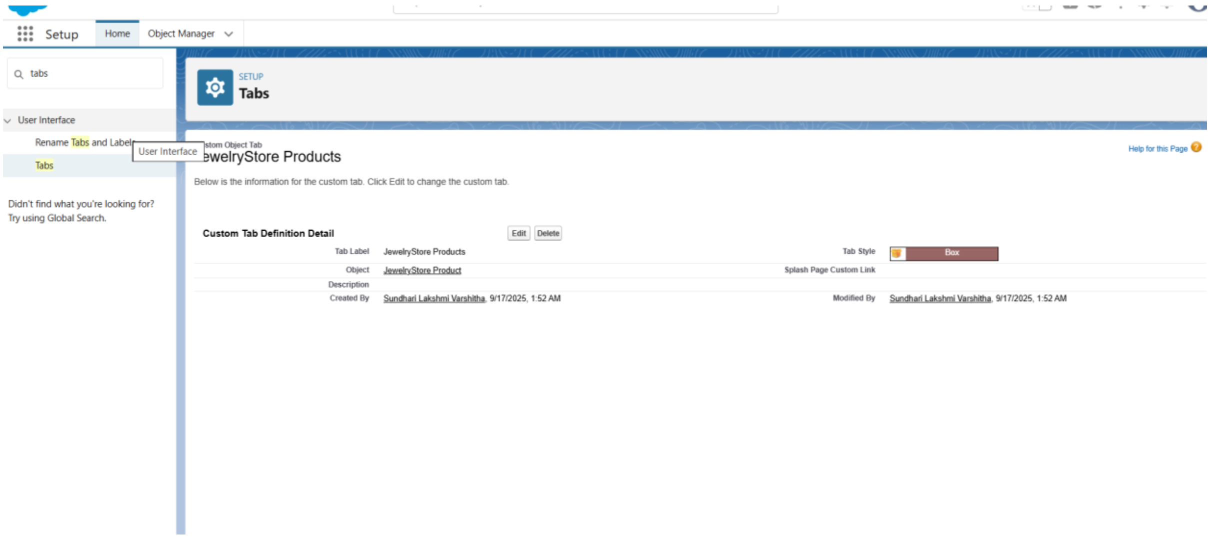
Below is the information for the custom tab. Click Edit to change the custom tab.

Custom Tab Definition Detail

Tab Label	JewelryStore Orders	Tab Style	Shopping Cart
Object	JewelryStore Order	Splash Page Custom Link	
Description			
Created By	Sundhari Lakshmi Varshitha, 9/17/2025, 1:51 AM	Modified By	Sundhari Lakshmi Varshitha, 9/17/2025, 1:51 AM

JewelryStore Products:

This tab contains the complete product catalog, allowing staff to view jewelry items, their prices, and stock information.



stom Object Tab

ewelryStore Products

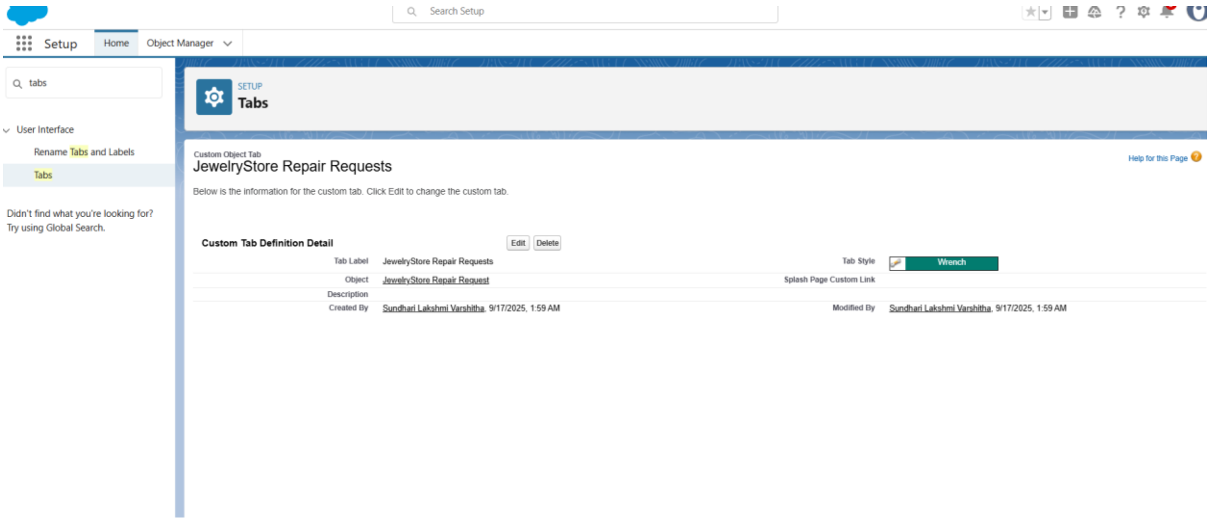
Below is the information for the custom tab. Click Edit to change the custom tab.

Custom Tab Definition Detail

Tab Label	JewelryStore Products	Tab Style	Box
Object	JewelryStore Product	Splash Page Custom Link	
Description			
Created By	Sundhari Lakshmi Varshitha, 9/17/2025, 1:52 AM	Modified By	Sundhari Lakshmi Varshitha, 9/17/2025, 1:52 AM

JewelryStore Repair Requests:

This tab is designed for logging and tracking jewelry repair requests, ensuring after-sales services are handled efficiently and customers are updated on the progress.



Custom Object Tab

JewelryStore Repair Requests

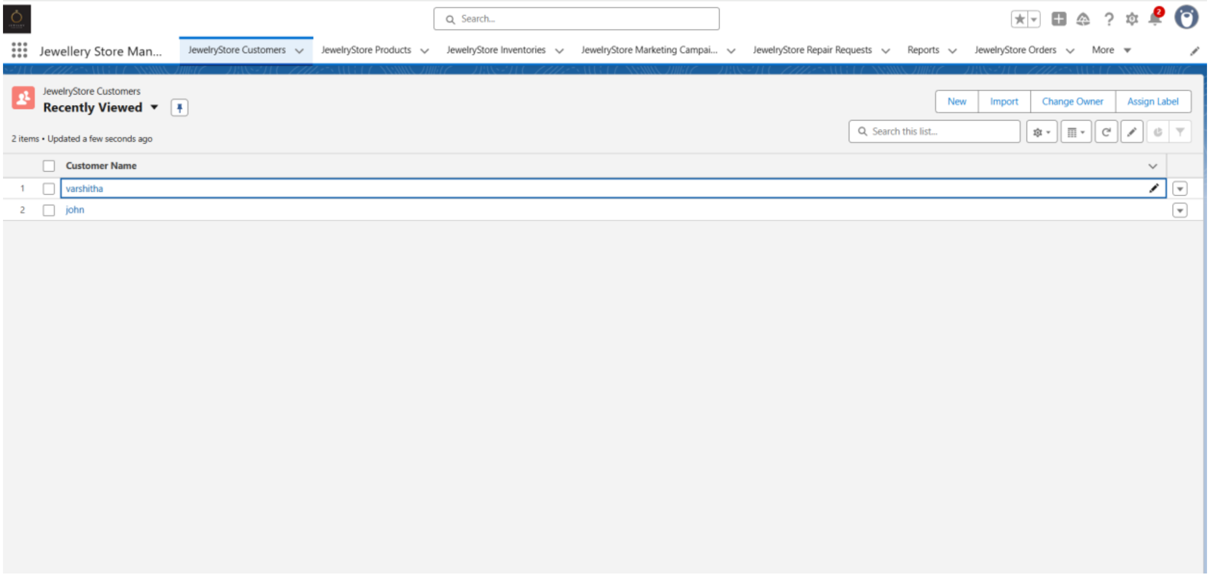
Below is the information for the custom tab. Click Edit to change the custom tab.

Custom Tab Definition Detail

Tab Label	JewelryStore Repair Requests	Tab Style	Wrench
Object	JewelryStore Repair Request	Splash Page Custom Link	
Description			
Created By	Sundhari Lakshmi Varshitha, 9/17/2025, 1:59 AM	Modified By	Sundhari Lakshmi Varshitha, 9/17/2025, 1:59 AM

4. Home Page Layouts

- The home page can be customized for different profiles (e.g., Sales Rep, Manager).
- Provides dashboards, tasks, and quick actions.
- **Jewelry Store Example:**
  - **For Sales Reps:** Show today's appointments, new leads, and pending orders.
  - **For Managers:** Display KPIs like daily revenue, low stock alerts, and customer complaints.



JewelryStore Customers

Recently Viewed

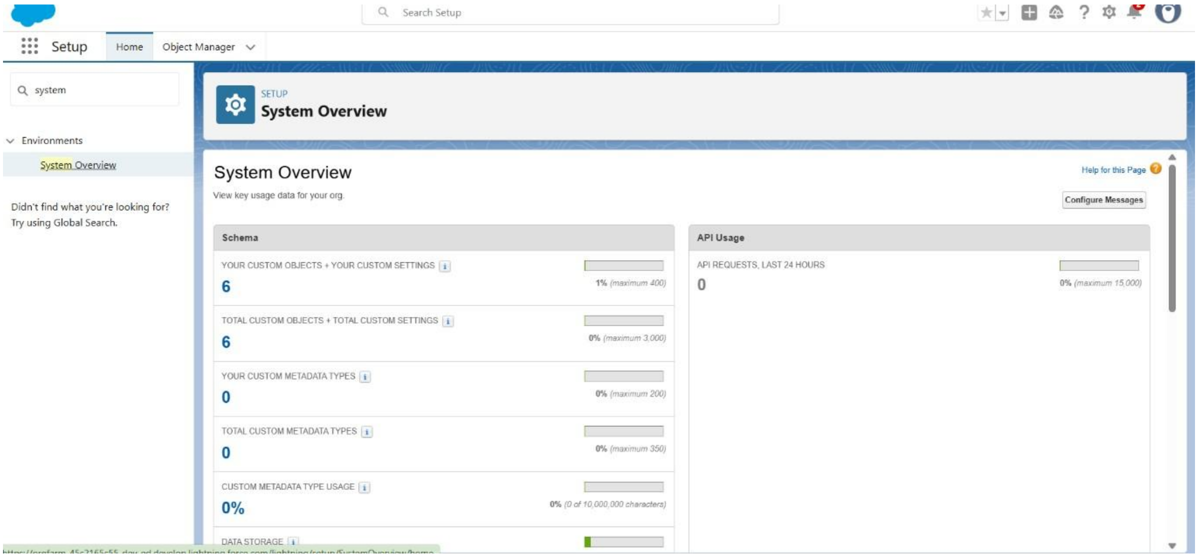
2 items • Updated a few seconds ago

	Customer Name
1	varshitha
2	john

Phase 7: Integration & External Access

1. API Limits

- Salesforce tracks the number of **API calls** made by external systems or integrations.
- In **Golden Era Enterprises CRM**, all major actions (Customer registration, Orders, Inventory tracking, Marketing Campaigns, Service Requests) are handled **inside Salesforce**.
- No external APIs are currently being used in this project, so **API limits are not a concern**.
- You can monitor API usage in **Setup → System Overview**, but no special configuration is required.



System Overview

View key usage data for your org.

Schema

YOUR CUSTOM OBJECTS + YOUR CUSTOM SETTINGS
6
1% (maximum 400)

TOTAL CUSTOM OBJECTS + TOTAL CUSTOM SETTINGS
6
0% (maximum 3,000)

YOUR CUSTOM METADATA TYPES
0
0% (maximum 200)

TOTAL CUSTOM METADATA TYPES
0
0% (maximum 350)

CUSTOM METADATA TYPE USAGE
0%
0% (0 of 10,000,000 characters)

DATA STORAGE

API Usage

API REQUESTS, LAST 24 HOURS
0
0% (maximum 15,000)

Phase 8: Data Management & Deployment

Step 1:Duplicate Management

Prevent duplicate **GoldenEra Customer** records (by Email/Phone/Name) and surface potential duplicates during data entry or import.

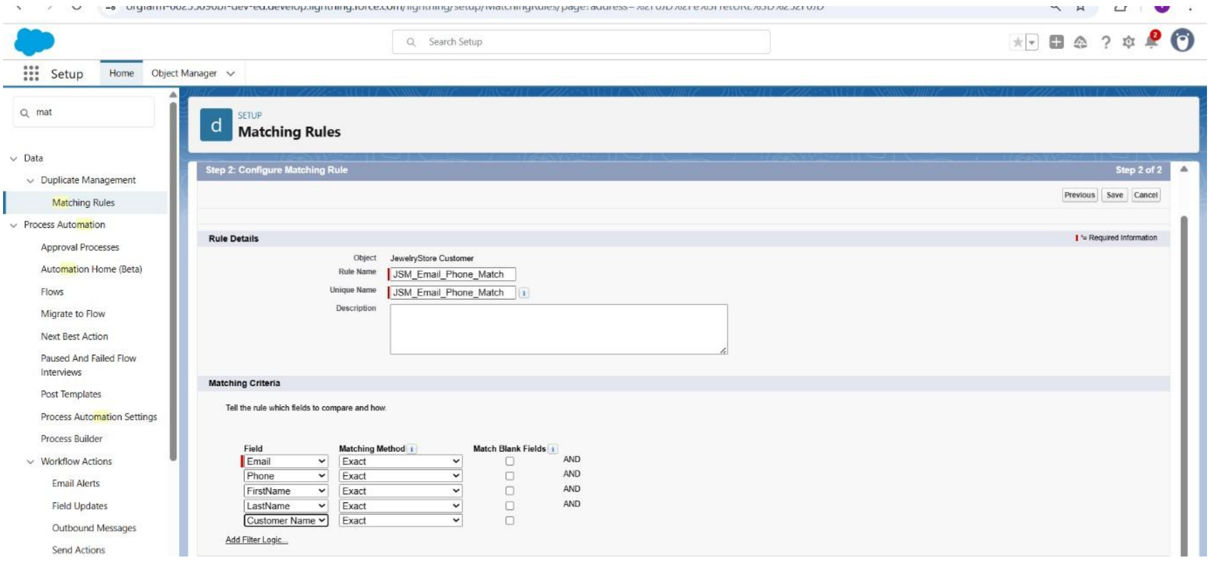
1) Create the Matching Rule

1. Setup → Quick Find → **Matching Rules** → **New Rule**.
2. **Object:** `GoldenEra Customer` (Jewelrystore_Customer__c).
3. **Rule Name:** `GEC_Email_Phone_Match` (or any clear name).
4. **Add Matching Criteria:**

o Field = **Email** → Matching Method = **Exact.** o Click **Add Row** → Field = **Phone** → Matching Method = **Exact**.

5. **Save** the rule.
6. Click **Activate** (only active rules can be used by Duplicate Rules).

7.



SETUP
Matching Rules

Step 2: Configure Matching Rule — Step 2 of 2

Rule Details — | = Required Information

Object	JewelryStore Customer
Rule Name	JSM_Email_Phone_Match
Unique Name	JSM_Email_Phone_Match
Description	

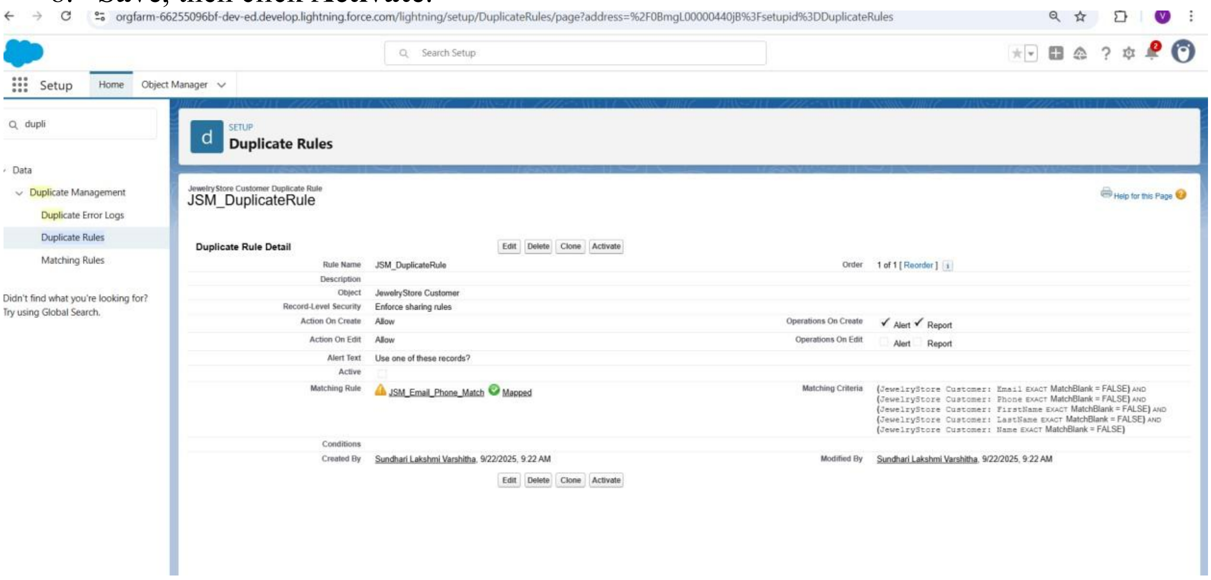
Matching Criteria

Tell the rule which fields to compare and how.

Field	Matching Method	Match Blank Fields	
Email	Exact	☐	AND
Phone	Exact	☐	AND
FirstName	Exact	☐	AND
LastName	Exact	☐	AND
Customer Name	Exact	☐	

2) Create the Duplicate Rule

1. Setup → Quick Find → **Duplicate Rules** → **New Rule**.
2. **Object:** `Jewelrystore Customer`.
3. **Rule Label:** `JSM_DuplicateRule`.
4. Under **Matching Rules**, click **Add** and select the `JSM_Email_Phone_Match` matching rule you just activated.
5. **Action on Create:** choose **Alert** (start in Alert mode while testing).
6. **Action on Edit:** choose **Alert**.
7. (Optional) Scope: set record types or profiles if you want the rule to apply only to some users.
8. **Save**, then click **Activate**.


orgfarm-66255096bf-dev-ed.develop.lightning.force.com/lightning/setup/DuplicateRules/page?address=%2F0BmgL00000440jB%3Fsetupid%3DDuplicateRules
Search Setup
Setup
Home
Object Manager
dupli
Data
Duplicate Management
Duplicate Error Logs
Duplicate Rules
Matching Rules
Didn't find what you're looking for? Try using Global Search.
Help for this Page
Edit
Delete
Clone
Activate

SETUP
Duplicate Rules

JewelryStore Customer Duplicate Rule
JSM_DuplicateRule

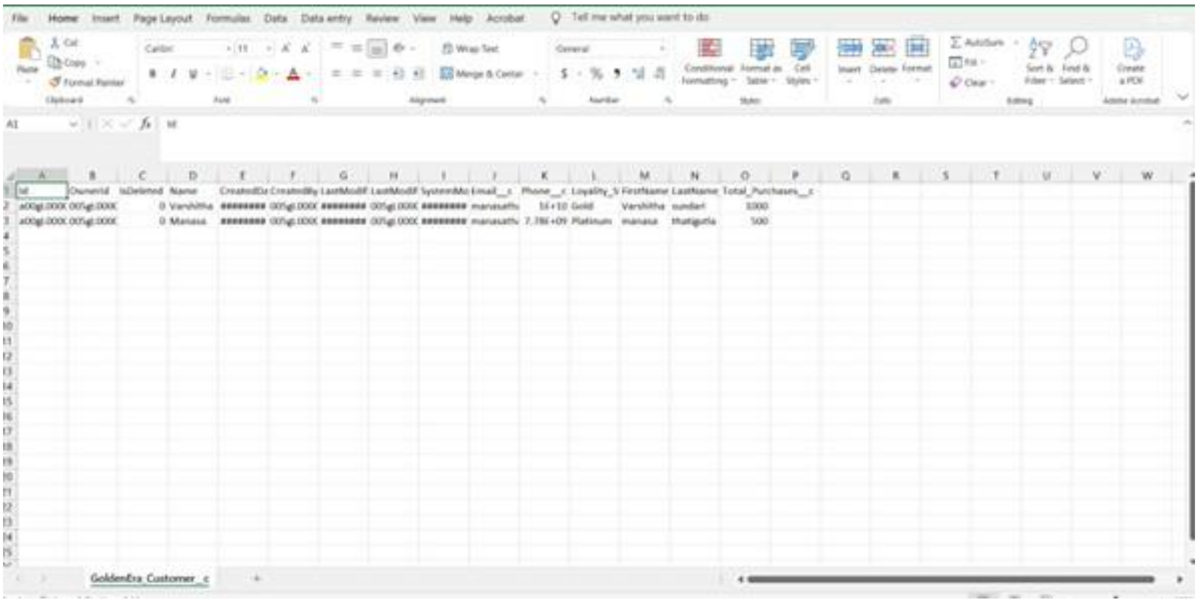
Duplicate Rule Detail

Rule Name	JSM_DuplicateRule	Order	1 of 1 [Reorder]
Description			
Object	JewelryStore Customer		
Record-Level Security	Enforce sharing rules		
Action On Create	Allow	Operations On Create	✓ Alert ✓ Report
Action On Edit	Allow	Operations On Edit	☐ Alert ☐ Report
Alert Text	Use one of these records?		
Active	☐		
Matching Rule	JSM_Email_Phone_Match ✓ Mapped	Matching Criteria	(JewelryStore Customer: Email EXACT MatchBlank = FALSE) AND (JewelryStore Customer: Phone EXACT MatchBlank = FALSE) AND (JewelryStore Customer: FirstName EXACT MatchBlank = FALSE) AND (JewelryStore Customer: LastName EXACT MatchBlank = FALSE) AND (JewelryStore Customer: Name EXACT MatchBlank = FALSE)
Conditions			
Created By	Sundhari Lakshmi Varshitha, 9/22/2025, 9:22 AM	Modified By	Sundhari Lakshmi Varshitha, 9/22/2025, 9:22 AM

Step 2:Data Backup

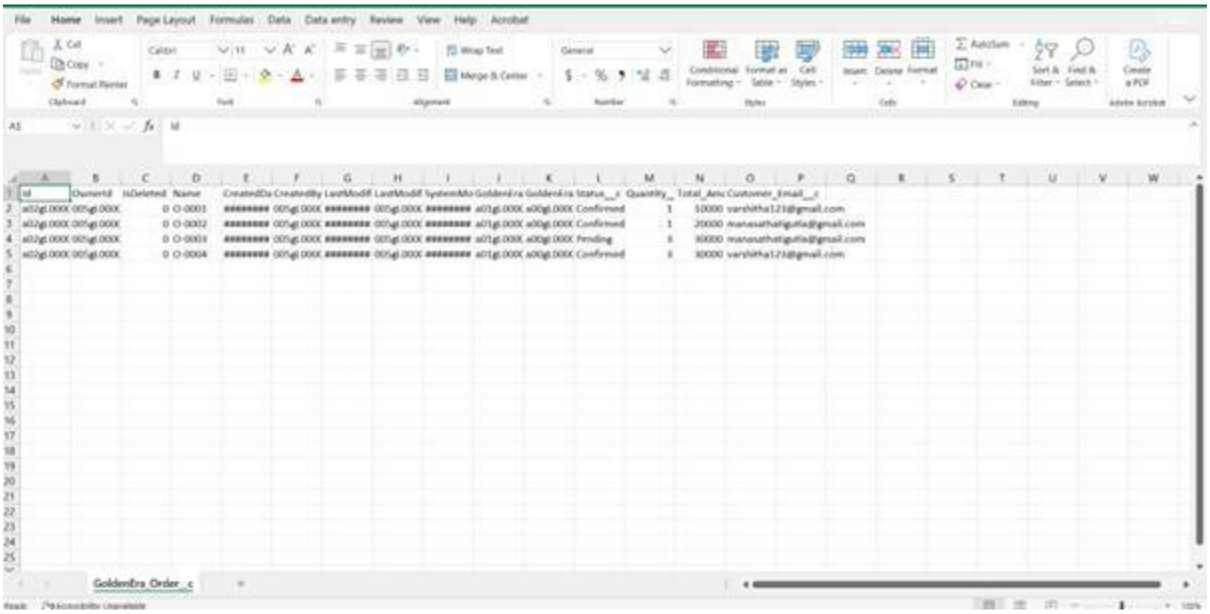
Steps:

- Go to **Setup**.
- In Quick Find, type **Data Export** → click **Data Export**.
- Choose one:
- **Export Now** → run a one-time backup.
- **Schedule Export** → set weekly/monthly backups.
- Select the objects you want:
- **JewelleryStore Customer, JewelleryStore Product, JewelleryStore Order, JewelleryStore Inventory, JewelleryStore Marketing Campaign, JewelleryStore Service Request**, and any standard objects you use (e.g., Users).
- Click **Start Export** (for immediate) or **Save** (for scheduled).
- Wait → Salesforce emails you when the backup is ready.
- Download the **.zip file** from the export page → extract CSV files.
- **Store the backup securely** (encrypted drive, company server, cloud storage).

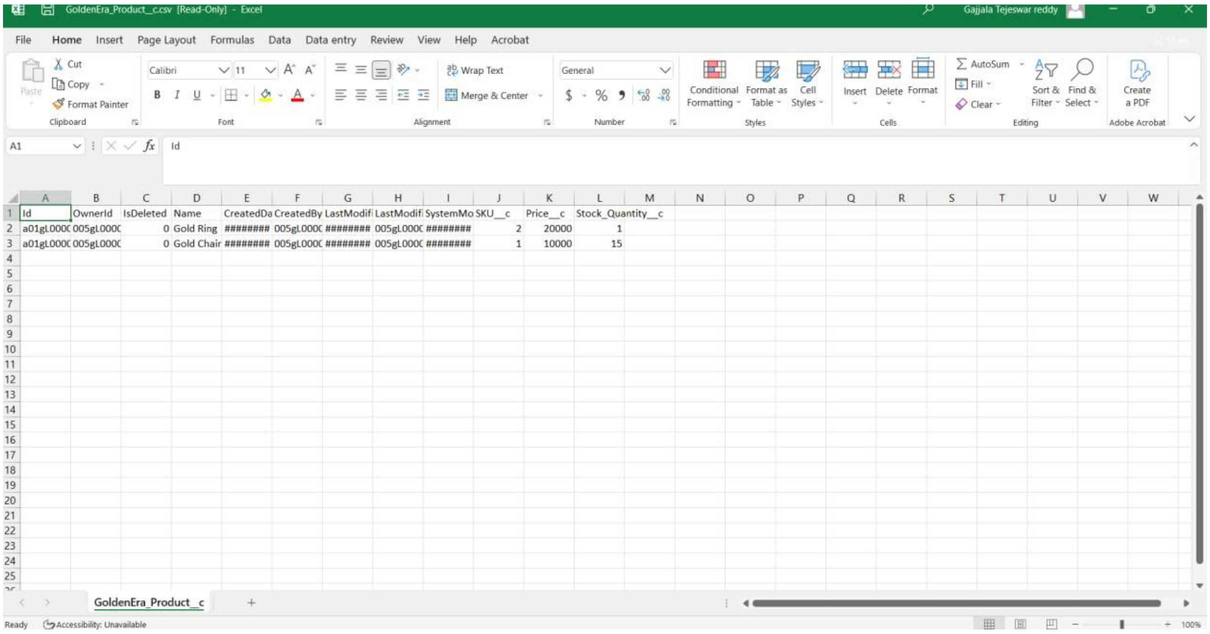


File Home Insert Page Layout Formulas Data Data entry Review View Help Acrobat
Tell me what you want to do
A1
Id
GoldenEra_Customer__c

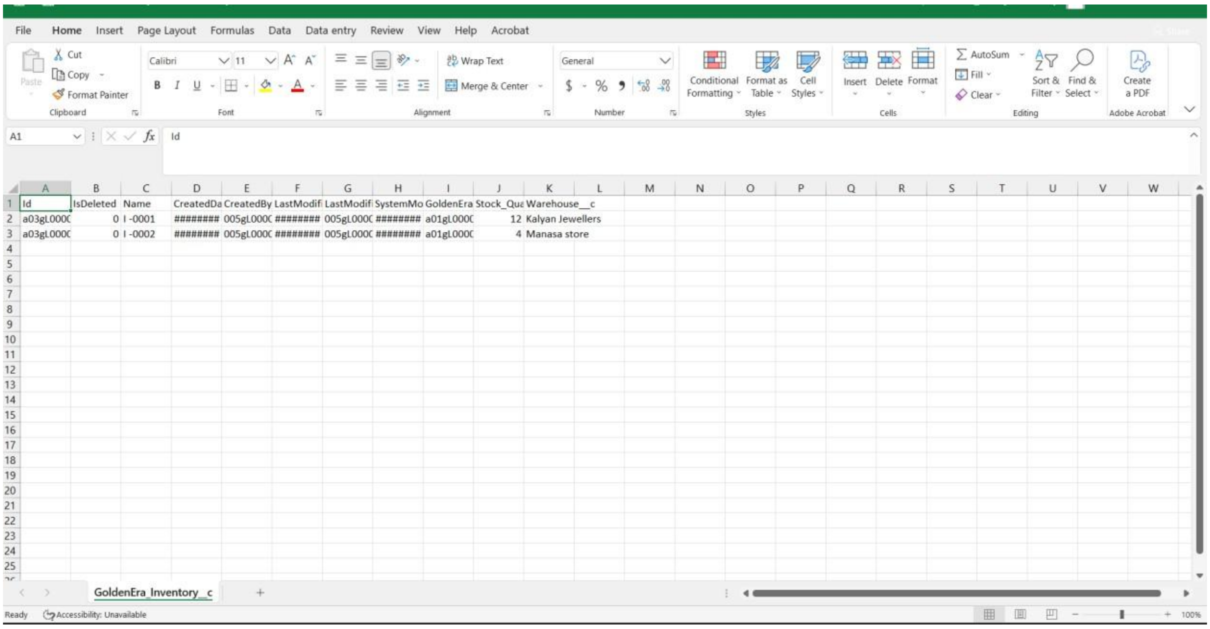
Id	OwnerId	IsDeleted	Name	CreatedDa	CreatedBy	LastModif	LastModif	SystemMo	Email__c	Phone__c	Loyalty_S	FirstName	LastName	Total_Purchases__c
a00gL000X	005gL000X	0	Varshitha	########	005gL000X	########	005gL000X	########	manasathi	1E+10	Gold	Varshitha	sundari	1000
a00gL000X	005gL000X	0	Manasa	########	005gL000X	########	005gL000X	########	manasathi	7.78E+09	Platinum	manasa	thatiguta	500



Id	OwnerId	IsDeleted	Name	CreatedDa	CreatedBy	LastModifi	LastModifi	SystemMo	GoldenEra	GoldenEra	Status__c	Quantity_	Total_Amo	Customer_Email__c
a02gL000C	005gL000C	0	O-0001	########	005gL000C	########	005gL000C	########	a01gL000C	a00gL000C	Confirmed	1	10000	varshitha123@gmail.com
a02gL000C	005gL000C	0	O-0002	########	005gL000C	########	005gL000C	########	a01gL000C	a00gL000C	Confirmed	1	20000	manasathatigutla@gmail.com
a02gL000C	005gL000C	0	O-0003	########	005gL000C	########	005gL000C	########	a01gL000C	a00gL000C	Pending	3	30000	manasathatigutla@gmail.com
a02gL000C	005gL000C	0	O-0004	########	005gL000C	########	005gL000C	########	a01gL000C	a00gL000C	Confirmed	3	30000	varshitha123@gmail.com



Id	OwnerId	IsDeleted	Name	CreatedDa	CreatedBy	LastModifi	LastModifi	SystemMo	SKU__c	Price__c	Stock_Quantity__c
a01gL000C	005gL000C	0	Gold Ring	########	005gL000C	########	005gL000C	########	2	20000	1
a01gL000C	005gL000C	0	Gold Chain	########	005gL000C	########	005gL000C	########	1	10000	15



Id	IsDeleted	Name	CreatedDa	CreatedBy	LastModifi	LastModifi	SystemMo	GoldenEra	Stock_Qua	Warehouse__c
a03gL000C	0	I -0001	########	005gL000C	########	005gL000C	########	a01gL000C	12	Kalyan Jewellers
a03gL000C	0	I -0002	########	005gL000C	########	005gL000C	########	a01gL000C	4	Manasa store

Phase 9: Reporting, Dashboards & Security Review

In Phase 9 of the Jewelry Store Management System project, the emphasis is placed on generating meaningful business insights, visualizing operational data, and enforcing robust data security protocols. This phase is crucial for enabling store managers and staff to monitor key business metrics such as sales performance, inventory status, and customer trends, while also protecting sensitive data such as customer information and financial records.

1. Reports

Reports help track **Products, Orders, and Sales** effectively in the Jewelry Store.

A. Create Reports

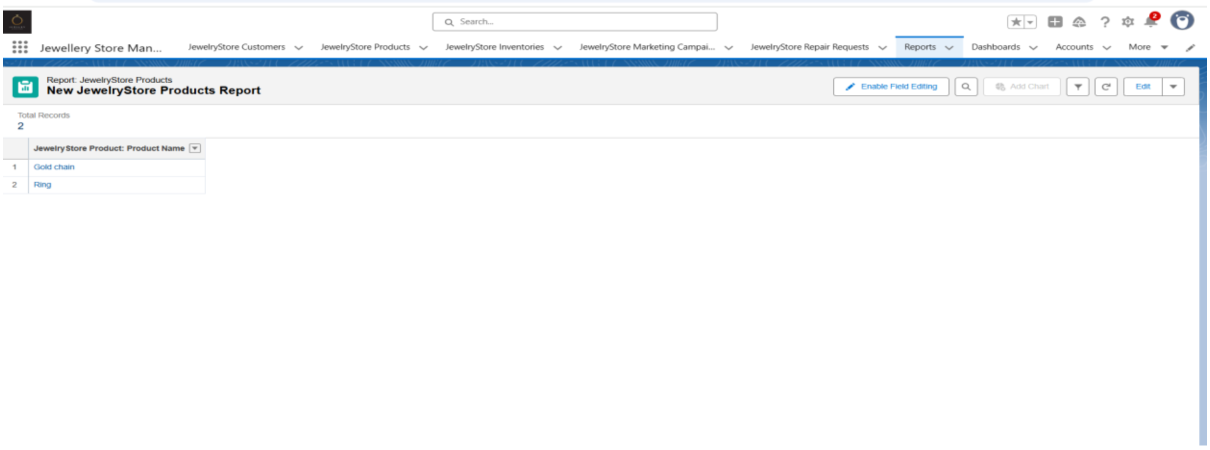
1. Go to **App Launcher → Reports → New Report**.
2. Select the report type: e.g., **Products** or **Orders**.
3. Add fields:
   - For Products Report:
     - Product Name
     - Category
     - Price
     - Quantity in Stock
     - Availability Status
   - For Orders Report:
     - Order ID
     - Customer Name

- Products Ordered
- Quantity
- Order Date
- Total Amount

4. Apply filters if needed (e.g., show only active products or recent orders).
5. Save and Run the report.

B. Useful Reports

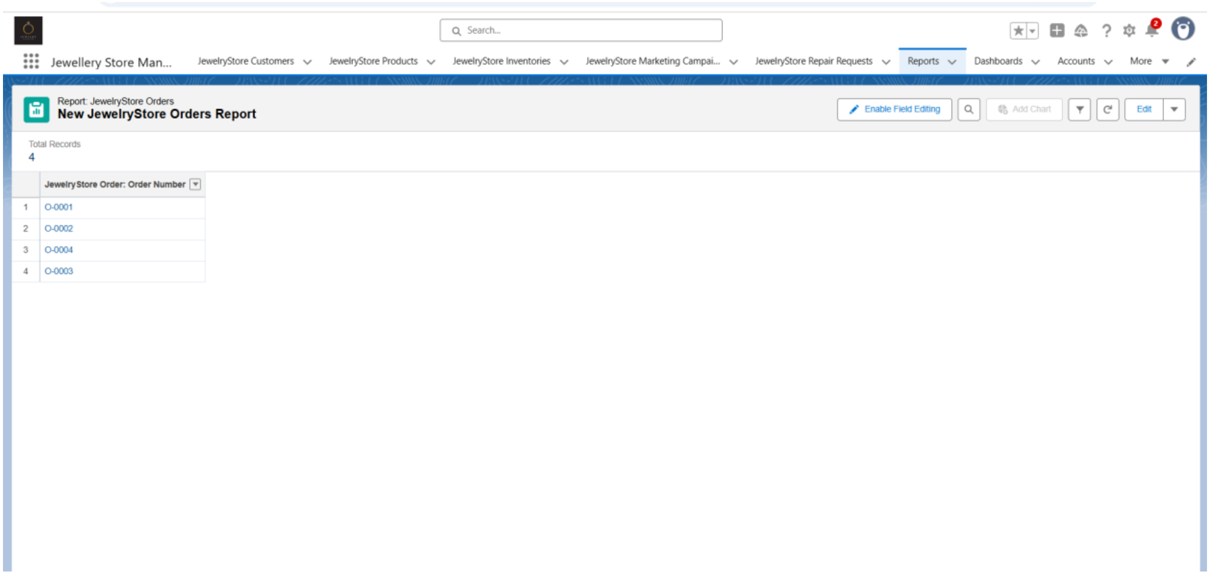
- **Products Inventory Report:** Shows product availability and stock levels.
- **Orders Summary Report:** Displays recent customer orders with quantities and amounts.
- **Sales by Category:** Groups sales data by jewelry category to identify best sellers.
- **Monthly Sales Trends:** Groups sales by month to track revenue changes over time.



Report: JewelryStore Products
New JewelryStore Products Report

Total Records
2

	JewelryStore Product: Product Name
1	Gold chain
2	Ring



Report: JewelryStore Orders
New JewelryStore Orders Report

Total Records
4

	JewelryStore Order: Order Number
1	O-0001
2	O-0002
3	O-0004
4	O-0003

2. Dashboard

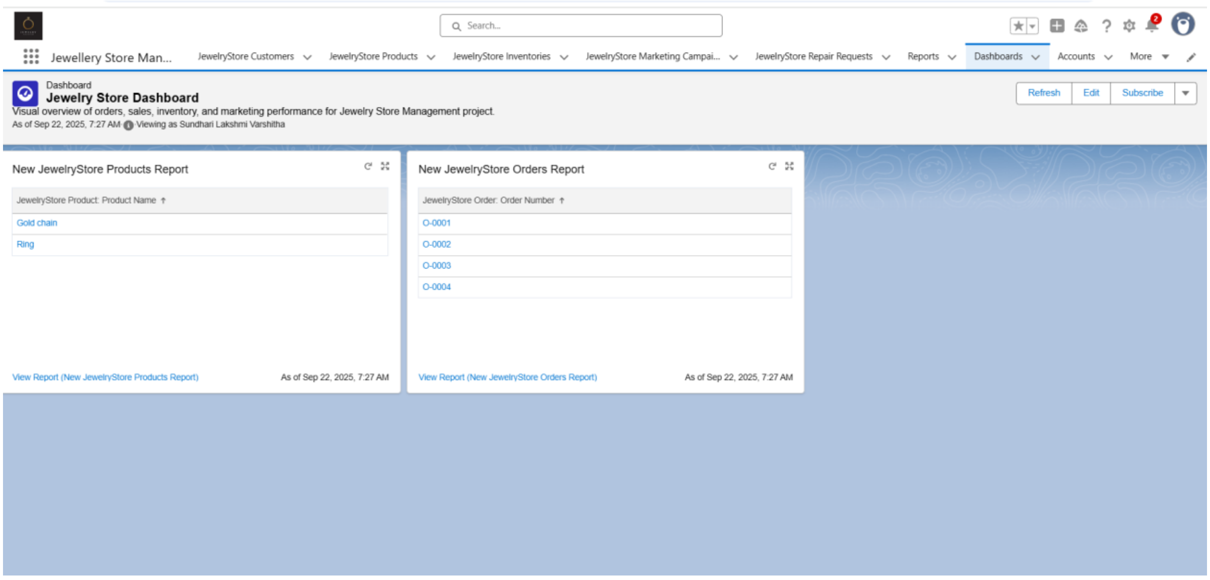
Dashboards provide visual insights into sales, inventory, and orders.

A. Create Dashboard

1. Go to App Launcher → Dashboards → New Dashboard.
2. Enter dashboard name, e.g., Jewelry Store Dashboard.
3. Add components using your reports as data sources (charts, tables, graphs).
4. Configure components to show key metrics:
    - Total Sales
    - Stock Levels
    - Recent Orders
5. Arrange components for clear visualization.
6. Save the dashboard.

B. Dashboard Benefits

- Visualizes key store performance metrics at a glance.
- Helps identify trends and stock issues quickly.
- Supports faster decision-making for store managers.



Dashboard
Jewelry Store Dashboard
Visual overview of orders, sales, inventory, and marketing performance for Jewelry Store Management project.
As of Sep 22, 2025, 7:27 AM · Viewing as Sundhari Lakshmi Varshitha

New JewelryStore Products Report

JewelryStore Product: Product Name ↑
Gold chain
Ring

View Report (New JewelryStore Products Report) As of Sep 22, 2025, 7:27 AM

New JewelryStore Orders Report

JewelryStore Order: Order Number ↑
O-0001
O-0002
O-0003
O-0004

View Report (New JewelryStore Orders Report) As of Sep 22, 2025, 7:27 AM