

E-Commerce Order Management with Salesforce CRM

Phase 1: Problem Understanding & Industry Analysis

This is where business needs are understood, pain points are identified, and goals are defined.

- **Requirement Gathering:**

- Customer registration & profile management.
- Shopping cart & checkout workflows.
- Order placement, cancellation, and return processes.
- Payment status updates.
- Salesforce sync for leads, accounts, opportunities, and orders.

- **Stakeholder Analysis:**

- *Customers* → Smooth purchase and real-time order tracking.
- *Sales Team* → Visibility into customer purchase history, ability to follow up.
- *Management* → Dashboards for revenue, top products, conversion trends.

- **Business Process Mapping:**

- Map the order lifecycle: Customer Signup → Browse Products → Add to Cart → Checkout → Order → Salesforce Sync.
- Capture where CRM must intervene (lead creation, order tracking, upsell opportunity).

- **Industry Analysis:**

- Most eCommerce systems lack native CRM → sales data and customer support often remain fragmented.
- Salesforce adds **customer 360° view**, enabling better cross-sell, upsell, and marketing campaigns.

- **AppExchange Exploration:**

- Shipping & logistics connectors.
- Payment gateway integrations.
- Marketing automation (Salesforce Marketing Cloud).

Phase 2: Org Setup & Configuration

Salesforce Editions

- Using Salesforce Developer Edition Org (free, permanent).
- Supports custom objects, flows, Apex, and dashboards.

Company Profile Setup

- Setup → Company Information → Edit.
- Company Name: Modern store
- Default Locale: English (India)
- Time Zone: Asia/Kolkata (IST) Currency: INR

The screenshot shows the 'Company Information' setup page in Salesforce. The page title is 'Company Information' with a sub-header 'Modern Store'. Below the title, it says 'The organization's profile is below.' There are links for 'User Licenses (10)', 'Permission Set Licenses (10)', 'Feature Licenses (11)', and 'Usage-based Entitlements (10)'. The 'Organization Detail' section is expanded, showing various settings. The 'Organization Name' is 'Modern Store'. The 'Primary Contact' is 'OrgFarm EPIC'. The 'Division' is 'Near noble school'. The 'Address' is 'Madanapalle 517325, Andhra Pradesh, India'. The 'Fiscal Year Starts In' is 'January'. The 'Default Time Zone' is '(GMT+05:30) India Standard Time (Asia/Kolkata)'. The 'Currency Locale' is 'English (India) - INR'. The 'Used Data Space' is '469 KB (9%)'. The 'Used File Space' is '156 KB (1%)'. The 'API Requests, Last 24 Hours' is '0 (15,000 max)'. The 'Streaming API Events, Last 24 Hours' is '0 (10,000 max)'. The 'Restricted Logins, Current Month' is '0 (0 max)'. The 'Salesforce.com Organization ID' is '00DgK00000BaXAa'. There are 'Save' and 'Cancel' buttons at the bottom.

Organization Detail	
Organization Name	Modern Store
Primary Contact	OrgFarm EPIC
Division	Near noble school
Address	Madanapalle 517325 Andhra Pradesh India
Fiscal Year Starts In	January
Activate Multiple Currencies	<input type="checkbox"/>
Enable Data Translation	<input type="checkbox"/>
Newsletter	<input checked="" type="checkbox"/>
Admin Newsletter	<input checked="" type="checkbox"/>
Hide Notices About System Maintenance	<input type="checkbox"/>
Hide Notices About System Downtime	<input type="checkbox"/>
Locale Formats	ICU
Phone	
Fax	
Default Locale	English (India)
Default Language	English
Default Time Zone	(GMT+05:30) India Standard Time (Asia/Kolkata)
Currency Locale	English (India) - INR
Used Data Space	469 KB (9%) View
Used File Space	156 KB (1%) View
API Requests, Last 24 Hours	0 (15,000 max)
Streaming API Events, Last 24 Hours	0 (10,000 max)
Restricted Logins, Current Month	0 (0 max)
Salesforce.com Organization ID	00DgK00000BaXAa

- Saved changes for org-wide consistency

Business Hours & Holidays

- Setup → Business Hours → New Business Hours.
- Working Hours: 9:00 AM – 6:00 PM (Mon–Sat).
- Setup → Holidays → Add national holidays (e.g., 15th August – Independence Day).
- Ensures workflows/approvals don't trigger during holidays.


Fiscal Year Settings

- Setup → Fiscal Year → Standard (Jan–Dec).
- Useful for revenue and sales reporting.

The screenshot shows the 'Change Fiscal Year Period' dialog box. It has a title bar with 'Save' and 'Cancel' buttons. The 'Name' field is 'Modern Store'. The 'Fiscal Year Start Month' is set to 'January' with a dropdown arrow. The 'Fiscal Year is Based On' section has two radio buttons: 'The ending month' (selected) and 'The starting month'. There are 'Save' and 'Cancel' buttons at the bottom.

Change Fiscal Year Period	
Name	Modern Store
Fiscal Year Start Month	January
Fiscal Year is Based On	<input checked="" type="radio"/> The ending month <input type="radio"/> The starting month

Business hours

 **SETUP Business Hours**

Business Hours Edit Save Cancel

Step 1. Business Hours Name

Business Hours Name

Use these business hours as the default ☒

Active ☒

Step 2. Time Zone

Time Zone

Step 3. Business Hours

Sunday	<input type="text" value="12:00 AM"/>	to	<input type="text" value="12:00 AM"/>	<input checked="" type="checkbox"/> 24 hours
Monday	<input type="text" value="9:00 AM"/>	to	<input type="text" value="5:30 PM"/>	<input type="checkbox"/> 24 hours
Tuesday	<input type="text" value="9:00 AM"/>	to	<input type="text" value="5:30 PM"/>	<input type="checkbox"/> 24 hours
Wednesday	<input type="text" value="9:00 AM"/>	to	<input type="text" value="5:30 PM"/>	<input type="checkbox"/> 24 hours
Thursday	<input type="text" value="9:00 AM"/>	to	<input type="text" value="5:30 PM"/>	<input type="checkbox"/> 24 hours
Friday	<input type="text" value="9:00 AM"/>	to	<input type="text" value="5:30 PM"/>	<input type="checkbox"/> 24 hours
Saturday	<input type="text" value="12:00 AM"/>	to	<input type="text" value="12:00 AM"/>	<input checked="" type="checkbox"/> 24 hours

Save Cancel

Dev Org Setup

The development environment was prepared to facilitate project development.

- A Salesforce Developer Edition org was provisioned for building the application.
- A GitHub repository was initialized to manage source code versions.
- VS Code and SFDX were set up to develop Apex classes and future Lightning Web Components.

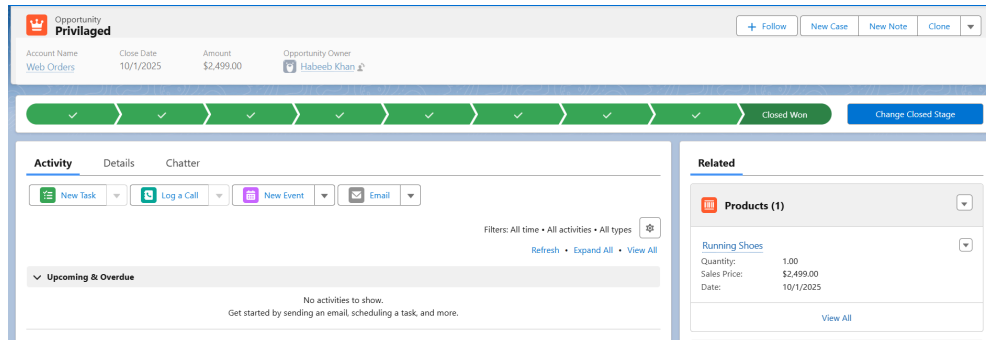
Phase 3: Data Modeling & Relationships

This phase defines the **CRM data architecture**.

- **Standard Objects Used:**
 - **Lead** → Captures potential customers signing up or leaving abandoned carts.
 - **Account** → Stores customer companies (for B2B).
 - **Contact** → Stores individual customers (for B2C).
 - **Opportunity** → Represents a potential deal (e.g., big cart order).
 - **Product** → Stores product catalog.
 - **Order & OrderItem** → Stores purchase details.

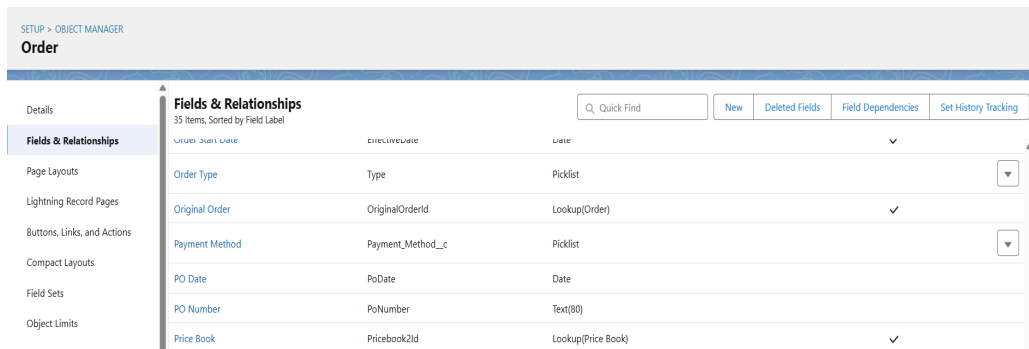
- **Custom Objects:**

- **Cart__c** → Temporary object for shopping cart.
- **Payment__c** → Tracks payment gateway status.



- **Relationships:**

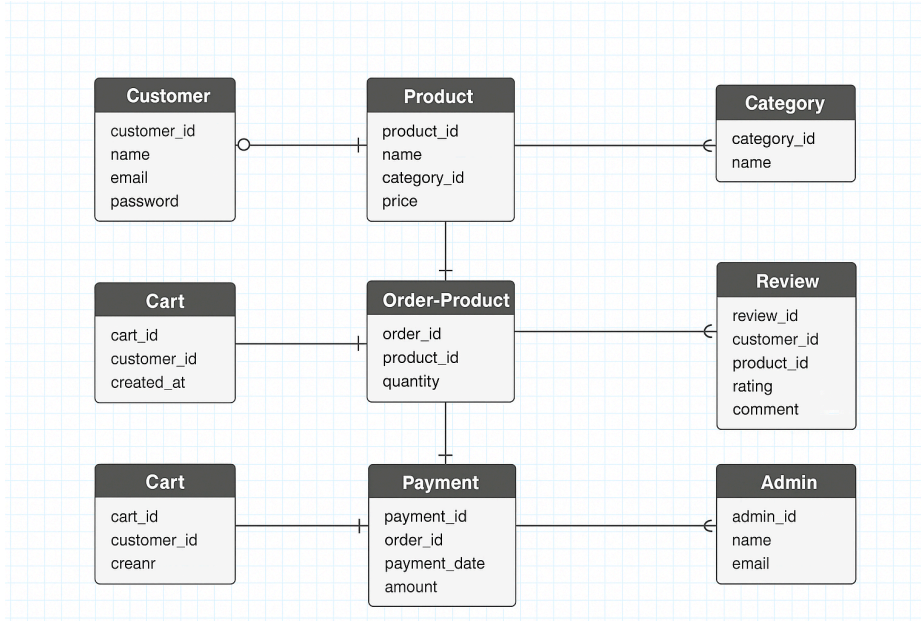
- One Account → Many Orders.
- One Order → Many Products (via OrderItem).
- Lookup from Payment__c → Order.



- **Record Types:**

- **Lead Types:** Website Lead, Social Media Lead.
- **Order Types:** Normal Order, Bulk Order, Return Order.

- **Schema Builder:** Use Salesforce Schema Builder to visually map eCommerce flow.

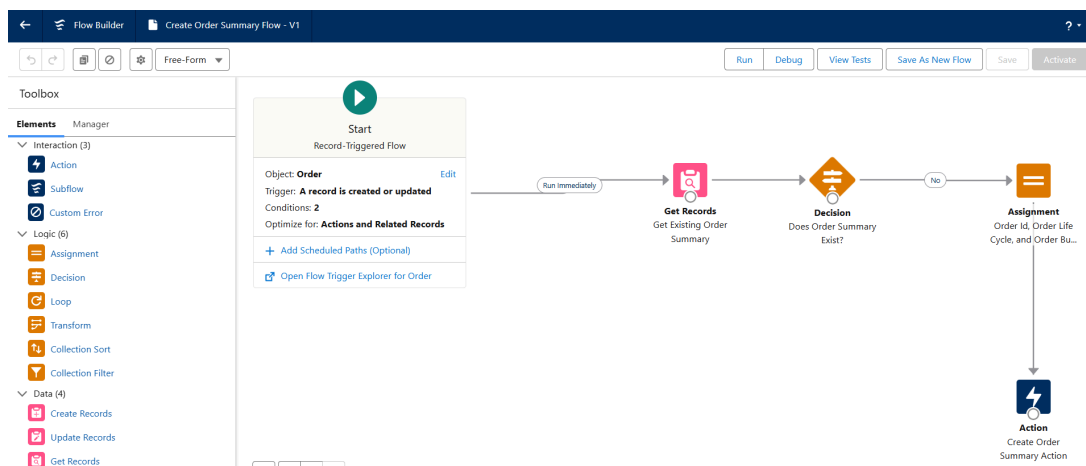


Phase 4: Process Automation (Admin)

Automation ensures **customer experience is fast and accurate**.

- **Validation Rules:**
 - Prevent negative product quantity.
 - Order cannot be closed unless payment is confirmed.
- **Workflow Rules:**
 - Send order confirmation email.
 - Notify manager when discount > 20%.
- **Process Builder:**
 - Auto-convert a Lead into Contact + Account when first purchase happens.
 - Update Opportunity Stage → “Won” when Order is placed.
- **Approval Process:**
 - Discount Approval workflow.
 - Refund Approval for returns.
- **Flow Builder:**
 - *Record-Triggered Flow:* Auto-create Opportunity when cart > \$500.

- *Screen Flow*: Order cancellation form.
- *Scheduled Flow*: Daily sync of abandoned carts into Leads.
- **Other Actions:**
 - *Email Alerts*: Order confirmation, shipment, refund.
 - *Tasks*: Assign follow-up calls for high-value customers.
 - *Custom Notifications*: Mobile push notifications for agents.



Phase 5: Apex Programming (Developer)

Custom code for scenarios beyond point-and-click tools.

- **Classes & Objects**: Handle order allocation and discount calculation.
- **Apex Triggers**:
 - Before Insert: Validate cart stock.
 - After Insert: Create Salesforce Order when payment received.
 - After Update: Update opportunity stage if order is shipped.
- **SOQL & SOSL**: Fetch customer purchase history.
- **Collections (List, Set, Map)**: Store bulk product IDs during checkout.
- **Batch Apex**: Bulk sync orders every night.
- **Queueable Apex**: Payment reconciliation with external gateway.
- **Scheduled Apex**: Daily email report of new orders to managers.

```
File Edit Debug Test Workspace Help < >
EcomHandler.apxc
Code Coverage: None API Version: 64
1 public class EcomHandler {
2     public static Decimal calcDiscount(Decimal amt) {
3         return (amt > 1000) ? amt * 0.1 : 0;
4     }
5     trigger CartBeforeInsert on Cart__c (before insert) {
6         for (Cart__c c : Trigger.new)
7             if (c.Total_Items__c > 100)
8                 c.addError('Stock limit exceeded!');
9     }
10    trigger PaymentAfterInsert on Payment__c (after insert) {
11        List<Order__c> ords = new List<Order__c>();
12        for (Payment__c p : Trigger.new)
13            if (p.Status__c == 'Paid')
14                ords.add(new Order__c(Customer__c=p.Customer__c, Amount__c=p.Amount__c));
15        insert ords;
16    }
17    trigger OrderAfterUpdate on Order__c (after update) {
18        List<Opportunity> opps = new List<Opportunity>();
19        for (Order__c o : Trigger.new)
20            if (o.Status__c == 'Shipped')
21                opps.add(new Opportunity(Id=o.Opportunity__c, StageName='Closed Won'));
22        update opps;
23    }
24    public static List<Order__c> getHistory(Id custId) {
25        return [SELECT Id, Amount__c FROM Order__c WHERE Customer__c = :custId];
26    }
}
```

- **Future Methods:** Async calls to external APIs (shipping, payments).

Phase 6: User Interface Development

Front-end design inside Salesforce.

- **Lightning App Builder:** Create “ECommerce CRM App.”
- **Custom Tabs:** Products, Orders, Leads, Payments.
- **Record Pages:** Customize Order page → show Product Items + Payment.

- **Utility Bar: Quick order search.**

Lightning App Builder | App Settings | Pages | Ecommerce_crm

App Settings

- App Details & Branding
- App Options
- Utility Items (Desktop Only)
- Navigation Items
- User Profiles

App Details & Branding
Give your Lightning app a name and description. Upload an image and choose the highlight color for its navigation bar.

App Details

*App Name ⓘ
Ecommerce_crm

*Developer Name ⓘ
Ecommerce_crm

Description ⓘ
Enter a description...

App Branding

Image ⓘ
[Green circle with white 'H']

Primary Color Hex Value ⓘ
#0070D2

Clear

Org Theme Options
☐ Use the app's image and color instead of the org's custom theme

App Launcher Preview

[Green circle with white 'H'] Ecommerce_crm

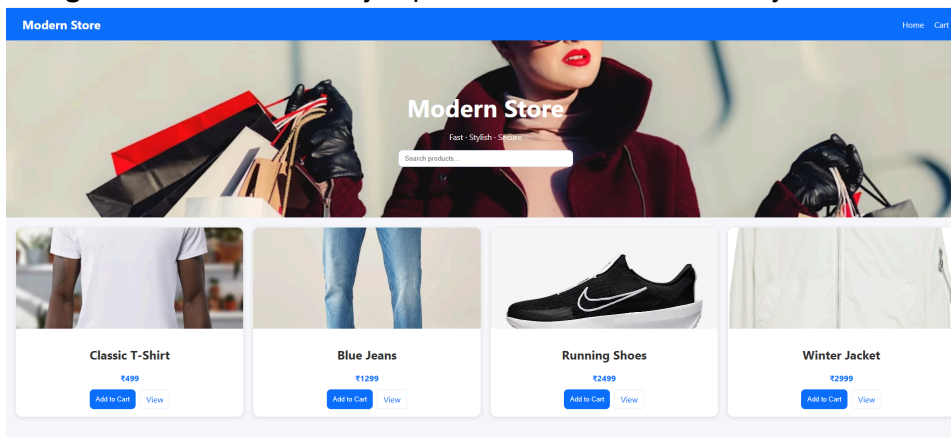
- **LWC (Lightning Web Components):**

- Product Search & Filter.
- Cart component inside Salesforce.
- Live order tracking view.

- **LWC + Apex:** Show “Recommended Products” (cross-sell/upsell).

- **Events & Wire Adapters:** Fetch stock availability.

- **Navigation Service:** Quick jump from Lead → Order → Payment.



Your Cart

Running Shoes	₹2499	1	Remove
---------------	-------	---	--------

Checkout

Contact Information

Billing Address

Shipping Address

☒ Shipping address same as billing address

Payment Information

Select Payment Method: ...Please choose an option...

Place Order

Phase 7: Integration & External Access

Bridge between Flask eCommerce App and Salesforce.

- **Named Credentials:** Store API keys securely.
- **External Services:** Integrate shipping APIs.

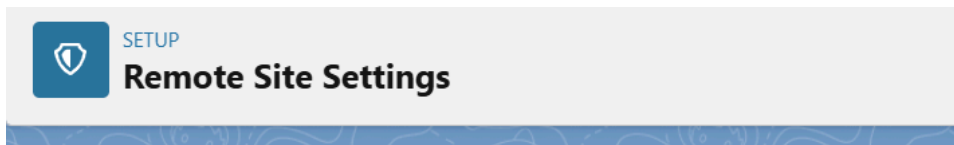
```

app.py 1 X
E-Commerce-Order-Management-with-Salesforce-CRM > app.py > ...
1 from flask import Flask, render_template, request, jsonify
2 from simple_salesforce import Salesforce, SalesforceAuthenticationFailed, SalesforceMalformedRequest
3 import os
4 from datetime import date
5 import random
6 import string
7 import requests # Import requests
8 from requests.packages.urllib3.exceptions import InsecureRequestWarning # Import InsecureRequestWarning
9
10 # Disable SSL warnings (use with caution in production)
11 requests.packages.urllib3.disable_warnings(InsecureRequestWarning)
12
13 # ----- SALESFORCE CREDENTIALS -----
14 SF_USERNAME = os.getenv("SF_USERNAME", "22691a3141147@agentforce.com")
15 SF_PASSWORD = os.getenv("SF_PASSWORD", "Habeet@123")
16 SF_SECURITY_TOKEN = os.getenv("SF_SECURITY_TOKEN", "W1ZH5gc1wD71Im9NqY9ivTs")
17 # -----
18
19 app = Flask(__name__)
20
21 # Connect to Salesforce once at startup
22 try:
23     # Create a session with SSL verification disabled
24     session = requests.Session()
25     session.verify = False
26
27     sf = Salesforce(
28         username=SF_USERNAME,
29         password=SF_PASSWORD,
30         security_token=SF_SECURITY_TOKEN,
31         session=session # Pass the session with verify=False
32     )

```

- **Web Services:**
 - REST → Sync orders & customers from Flask app.
 - SOAP → For older ERP integrations.
- **Callouts:** Payment gateway verification.

- **Platform Events:** Notify sales team when big order placed.
- **Change Data Capture:** Real-time sync for inventory.
- **Salesforce Connect:** If external product catalog is huge.
- **API Limits:** Monitor governor limits to avoid failures.
- **OAuth & Authentication:** Secure customer login.
- **Remote Site Settings:** Allow Flask app domain for API calls.



Remote Site Details

Remote Site Detail		Edit	Delete	Clone
Remote Site Name	Modern_store			
Remote Site URL	http://localhost:5000			
Disable Protocol Security	<input type="checkbox"/>			
Description				
Active	<input checked="" type="checkbox"/>			
Created By	Habeeb Khan, 10/9/2025, 7:59 AM	Edit	Delete	Clone

```
PS C:\Users\Patan Davood Khan\Videos\ecommerce_crm> python app.py
[+] Connected to Salesforce successfully!
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with watchdog (windowsapi)
[+] Connected to Salesforce successfully!
* Debugger is active!
* Debugger PIN: 671-768-980
```

Phase 8: Data Management & Deployment

Ensures clean data & smooth migration.

- **Data Import Wizard:** Import small CSVs of products.
- **Data Loader:** Bulk import 10,000+ orders.

Setup

Home

Object Manager

Cancel job

Choose data

Edit mapping

Start Import

Review & Start Import

Review your import information and click Start Import.

[Help for this page](#)

Your selections:

Details ✓

Add new records ✓

ecommerce_products_200.csv ✓

Your import **will include**:

Mapped fields

4

Your import **will not include**:

Unmapped fields

0

Cancel

Previous

Start Import

Congratulations, your import has started!
Click OK to view your import status on the Bulk Data Load Job page.

OK

- **Duplicate Rules:** Prevent duplicate Leads & Accounts.

SETUP

Duplicate Rules

Detail Duplicate Rule

User duplicate rule

Duplicate Rule Detail

Edit

Delete

Clone

Activate

Rule Name	User duplicate rule	Order	1 of 1	<div>Reorder</div>
Description				
Object	Detail			
Record-Level Security	Enforce sharing rules			
Action On Create	Allow	Operations On Create	<input type="checkbox"/> Alert	<input type="checkbox"/> Report
Action On Edit	Allow	Operations On Edit	<input type="checkbox"/> Alert	<input type="checkbox"/> Report
Alert Text				
Active	<input type="checkbox"/>			
Matching Rule	✔ Standard Contact Matching Rule ✔ Mapped		Matching Criteria	Matching rule for contact records. More info
Conditions	Current User: Email NOT EQUAL TO Email			
Created By	Habeeb Khan, 10/9/2025, 9:22 AM		Modified By	Habeeb Khan, 10/9/2025, 9:24 AM

Edit

Delete

Clone

Activate

- **Data Export & Backup:** Weekly scheduled backup.
- **Change Sets:** Deploy flows, objects, and layouts.
- **Packages:** Managed for reusable components, unmanaged for internal.
- **ANT Migration Tool:** For DevOps CI/CD pipelines.
- **VS Code & SFDX:** Modern development & deployment.

Phase 9: Reporting, Dashboards & Security Review

For management insights.

- **Reports:**
 - Tabular → All orders placed today.
 - Summary → Sales by price.
 - Matrix → Orders by Region vs Sales Rep.
 - Group by → Shipping cities.

Reports

Created by Me

1 item

REPORTS

Recent

Created by Me

Private Reports

Public Reports

All Reports

FOLDERS

All Folders

Created by Me

Shared with Me

Report Name

Description

Folder

Created By

Created On

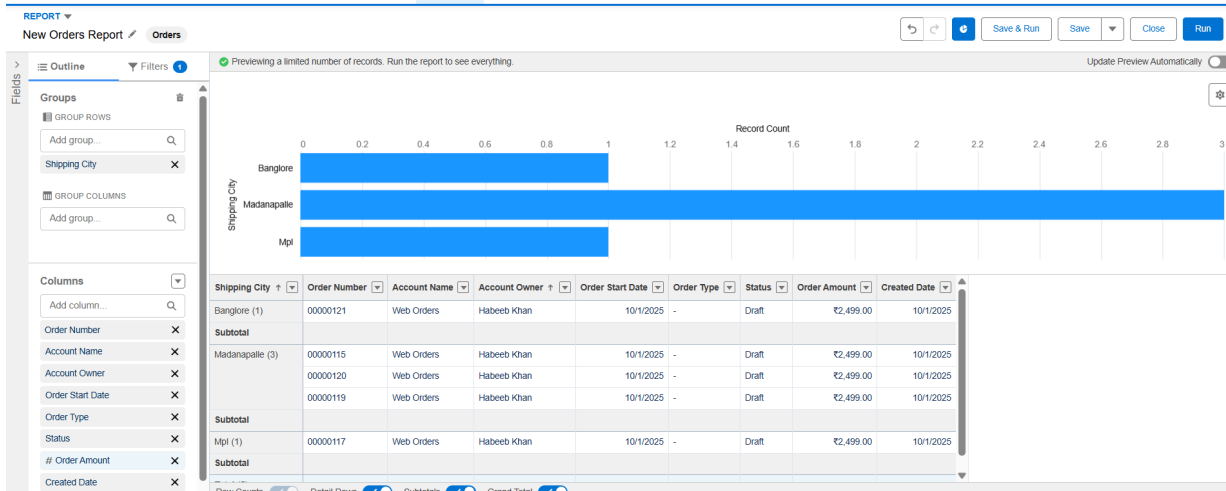
Subscribed

New Orders Report

Private Reports

Habeeb Khan

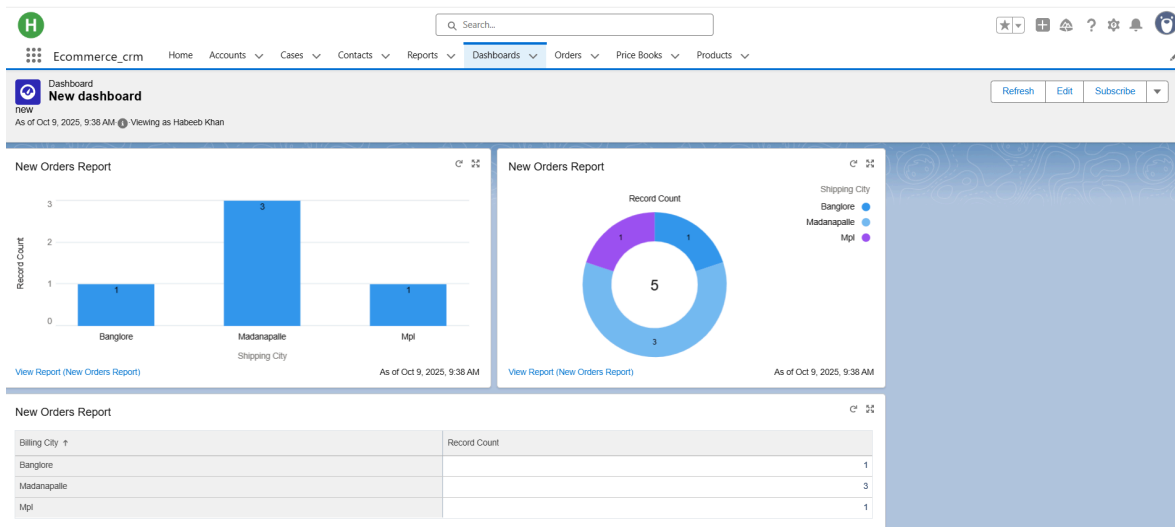
10/9/2025, 9:27 AM



- **Dashboards:**
Visualize key metrics for quick insights.

Steps:

- Dashboards → New Dashboard.
- Add components (charts, tables) using saved reports.
- Configure filters and display options. Save & run in a public folder.



- **Dynamic Dashboards:** Different KPIs for Manager vs Rep.
- **Security Settings:**
 - Field-Level Security for payment info.
 - Session Settings for MFA.
 - Login IP Ranges for secure access.



SETUP

Network Access

Network Access

The list below contains IP address ranges from sources that your organization trusts. Users logging in to salesforce.com with a browser from

Trusted IP Ranges			New
Action	Start IP Address ↑	End IP Address	
Edit Del	192.168.29.1	192.168.29.200	

- Audit Trail → Track admin changes.

Phase 10: Final Presentation & Demo Day

Key Features of the System:

1. Custom Web Application

- Centralized platform for managing products, orders, customers, and returns.

2. Product & Inventory Management

- Easy addition, update, and tracking of products, categories, and stock levels.

3. Role-Based Access

- Secure access for **Admin**, **Seller**, and **Customer** roles with appropriate permissions.

4. Order & Payment Handling

- Tracks customer orders from placement to delivery and integrates multiple payment options.

5. Complaint & Return Management

- Manages customer complaints, returns, and refunds efficiently to improve service.

6. Data Management

- Import, export, and maintain product, order, and customer data seamlessly using built-in database tools.

7. Automated Notifications

- Sends email/SMS alerts for **order confirmation**, **shipping updates**, and **promotions**.

8. Reports & Analytics

- Provides detailed reports on sales, top-selling products, customer activity, and revenue trends.

9. Dashboards

- Offers real-time insights into **business performance**, **inventory levels**, and **sales analytics** through interactive dashboards.