

**Phase 3: Data Modeling & Relationships**, written in a **professional, academic + project-report style**, covering **every object, field, relationship, justification, and structural diagram explanation**.

---

## PHASE 3: Data Modeling & Relationships (Detailed Version)

### **Goal:**

To design and implement a scalable and efficient Salesforce data architecture to support the Pharmacy Delivery CRM system.

---

### **3.1 Overview of Data Model**

The Pharmacy Delivery CRM requires a data model that can:

- Capture **patients, pharmacies, delivery agents, and delivery records**
- Maintain clear relationships between the entities
- Support automation, reporting, dashboards, and workflows
- Scale easily with future requirements (ex: agent routes, vehicle tracking, pharmacy branches, etc.)

To achieve this, both Salesforce **standard objects** and **custom objects** were utilized.

---

### **3.2 Standard & Custom Objects**

#### **3.2.1 Standard Object Used**

##### **Contact (renamed as Patient)**

The Contact object stores patient information because:

- Contacts already have fields like email, phone, address
- Salesforce best practice is to avoid recreating customer objects
- Contacts support Activities (emails, tasks, etc.)

##### **Used As:**

✓ Patient receiving the delivery

---

#### **3.2.2 Custom Objects Created**

##### **A) Delivery**

This is the **core object** that stores all delivery-related data.

**Purpose:**

Track the lifecycle of each medicine delivery request.

**Key Deliverables:**

- Delivery status
  - Delivery date
  - Pharmacy
  - Patient
  - Assigned agent
  - Payment info
  - Address
- 

**B) Delivery Agent**

This object stores information related to each agent responsible for delivering medicines.

**Purpose:**

Maintain a central list of agents with their contact and operational details.

---

**3.3 Fields in Each Object****3.3.1 Delivery Object Fields**

Field Label	API Name	Data Type	Purpose
Delivery Name	Name	Auto Number	Unique identifier (ex: DEL-0001)
Patient	Patient__c	Lookup(Contact)	Link to patient receiving delivery
Pharmacy	Pharmacy__c	Lookup(Account)	Linked pharmacy issuing medicines
Delivery Date	Delivery_Date__c	Date	Scheduled date
Delivery Address	Delivery_Address__c	Long Text	Customer address
Status	Status__c	Picklist	New, Scheduled, Out for Delivery, Delivered, Late, Failed
Amount	Amount__c	Currency	Total cost of medicines

Field Label	API Name	Data Type	Purpose
Payment Method	Payment_Method__c	Picklist	Cash on Delivery / Online
Delivery Agent	Delivery_Agent__c	Lookup(Delivery Agent)	Assigned agent
Alert Sent	Alert_Sent__c	Checkbox	Used in automation
Created By	CreatedById	Lookup(User)	Audit
Last Modified By	LastModifiedById	Lookup(User)	Audit

---

### 3.3.2 Delivery Agent Object Fields

Field Label	API Name	Data Type	Purpose
Delivery Agent Name	Name	Text	Name of the agent
Email	Email__c	Email	Used for notifications
Phone	Phone__c	Phone	Contact number
Vehicle Type	Vehicle_Type__c	Picklist	Bike / Scooter / Car
Active	Active__c	Checkbox	Indicates agent availability
Owner	OwnerId	Lookup(User)	User ownership

---

## 3.4 Relationship Settings

The system requires clean, flexible relationships. Therefore, only **Lookup relationships** were used.

### 3.4.1 Delivery → Patient (Contact)

- A patient can have multiple deliveries.
- Deliveries should not get deleted when the patient is deleted.
- Activities/logs should remain independent.

**Relationship Type:** Lookup

**Cardinality:** 1 Patient → Many Deliveries

---

### 3.4.2 Delivery → Pharmacy (Account)

Pharmacy is linked through Accounts for:

- Reusability
- Integration with Email-to-Case or future features

**Relationship Type:** Lookup

✓ Also allows multiple deliveries from the same pharmacy.

---

### 3.4.3 Delivery → Delivery Agent

Each delivery is handled by one agent.

**Relationship Type:** Lookup

**Reason:**

- Agent can change mid-way
  - Deleting an agent should not delete deliveries
  - Agents are not “owners” of deliveries
- 

## 3.5 Why Lookup Instead of Master-Detail?

**Reasons Lookup Was Chosen:**

1. **No strict dependency:**  
Deliveries must not be deleted automatically if patient/agent is deleted.
2. **Flexible ownership:**  
Users (not agents) remain owners of deliveries.
3. **Better automation support:**  
Flows and process builders work well with Lookups.
4. **Reporting flexibility:**  
Custom report types allow:
  - Deliveries with patients
  - Deliveries with agents
  - Deliveries grouped by pharmacy

**Why Hierarchical Not Used?**

Hierarchical relationships apply *only to User object*, so not suitable.

---

## 3.6 Record Types (future scalability)

Even though not required now, the model supports adding:

## **Potential Record Types:**

1. **Normal Delivery**
2. **Express Delivery**
3. **Medical Emergency Delivery**

Record Types make the system scalable for future versions.

---

## **3.7 Page Layouts**

### **3.7.1 Delivery Page Layout**

Contains:

- Basic delivery info (all fields)
- Status & Automation fields
- Related lists:
  - Delivery Agent
  - Activities
  - Notes & Attachments

#### **Purpose:**

Make it easy for users to view:

- ✓ Who ordered
  - ✓ Who delivers
  - ✓ Status
  - ✓ Payment
- 

### **3.7.2 Delivery Agent Layout**

Contains:

- Contact info
- Vehicle details
- Active status
- Related deliveries in a list

Ensures supervisors can track:

- ✓ Workload per agent
- ✓ Agent status
- ✓ Deliveries completed

---

### **3.8 Compact Layouts**

#### **Delivery Compact Layout (for highlights panel)**

Shows:

- Delivery Name
- Status
- Delivery Date
- Patient
- Amount

#### **Delivery Agent Compact Layout**

Shows:

- Agent Name
- Phone
- Email
- Vehicle

#### **Purpose:**

Better visibility in mobile and Lightning UI.

---

### **3.9 Schema Builder Representation**

Schema Builder visually shows the connections:

[Contact] ----(Lookup)----> [Delivery] ----(Lookup)----> [Delivery Agent]



(Lookup)

[Account (Pharmacy)]

#### **Schema Benefits:**

- Easy visualization
- Quick debugging
- Helps in documentation
- Shows all objects + their fields + relationships

---

### **3.10 Junction Objects (Not Required)**

A junction object is needed **only when many-to-many relationships exist**.

**Not needed because:**

- One delivery → one agent
- One delivery → one patient
- One delivery → one pharmacy

Possible future cases:

If a delivery needs **multiple agents**, then we create:

👉 *DeliveryAssignment\_c* (junction)

But not needed now.

---

### 3.11 External Objects (Optional)

Could integrate with:

- Pharmacy inventory system
- External delivery GPS tracking
- Medicine database

This phase is reserved for future expansion.

---

### 📌 PHASE 3 Summary

You built a **strong, scalable data model** containing:

- ✓ 2 Custom Objects
- ✓ 1 Standard Object
- ✓ All necessary relationships
- ✓ Lookup-based flexible design
- ✓ Page & compact layouts
- ✓ Automation-ready structure
- ✓ Schema Builder visualization

This data model fully supports:

- Deliveries
- Agents
- Pharmacies
- Patients
- Reporting

- Dashboards
  - Automations
- 

## Photos screenshots

SETUP > OBJECT MANAGER  
**Delivery**

**Validation Rules**  
3 Items, Sorted by Rule Name

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
Delivery_Date_Not_Past	Delivery Date	Delivery Date cannot be in the past.	✓	S.V.S.N. TEJA SWARUP, 12/7/2025, 5:00 AM
Require_Address_When_Delivered	Top of Page	Delivery Address is required when status is Delivered.	✓	S.V.S.N. TEJA SWARUP, 12/7/2025, 5:01 AM
Require_Agent_When_Out_For_Delivery	Delivery Agent	Delivery Agent is required when status is Out for Delivery.	✓	S.V.S.N. TEJA SWARUP, 12/7/2025, 7:40 AM

SETUP > OBJECT MANAGER  
**Delivery Agent**

**Fields & Relationships**  
8 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Active	Active__c	Checkbox		
Created By	CreatedBy	Lookup(User)		
Delivery Agent Name	Name	Text(80)		✓
Email	Email__c	Email		
Last Modified By	LastModifiedBy	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Phone	Phone__c	Phone		
Vehicle Type	Vehicle_Type__c	Picklist		

SETUP > OBJECT MANAGER  
**Delivery**

**Fields & Relationships**  
14 Items, Sorted by Field Label

FIELD NAME	FIELD LABEL	DATA TYPE	CONTROLLING FIELD	INDEXED
Delivery_Address	Delivery_Address__c	Long Text Area(32768)		
Delivery_Agent	Delivery_Agent__c	Lookup(Delivery Agent)		✓
Delivery_Date	Delivery_Date__c	Date		
Delivery_Name	Name	Auto Number		✓
LastModifiedBy	LastModifiedBy	Lookup(User)		
OwnerId	OwnerId	Lookup(User,Group)		✓
Patient	Patient__c	Lookup(Contact)		✓
Payment_Method	Payment_Method__c	Picklist		
Pharmacy	Pharmacy__c	Lookup(Account)		✓

Setup Home Object Manager

SETUP > OBJECT MANAGER Order

**Fields & Relationships**

33 items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Account Name	AccountId	Lookup(Account)		✓
Account Number	AccountNumber	Text(40)		
Activated By	ActivatedById	Lookup(User)		
Activated Date	ActivatedDate	Date/Time		✓
Bill To Contact	BillToContactId	Lookup(Contact)		✓
Billing Address	BillingAddress	Address		
Company Authorized By	CompanyAuthorizedById	Lookup(User)		
Company Authorized Date	CompanyAuthorizedDate	Date		
Contract End Date	ContractEndDate	Date		
Contract Name	ContractName	Text(80)		

Contract Number

Contract Number

SETUP > OBJECT MANAGER Delivery

**Page Layouts**

1 items, Sorted by Page Layout Name

PAGE LAYOUT NAME	CREATED BY	MODIFIED BY
Delivery Layout	S.V.S.N. TEJA SWARUP, 12/7/2025, 3:47 AM	S.V.S.N. TEJA SWARUP, 12/7/2025, 10:06 PM

Setup Home Object Manager

SETUP > OBJECT MANAGER Delivery

**Flow Triggers**

4 items • Sorted by Flow Label

Flow Label	Process Type	Trigger	Active	Last Modified By	Last Modified Date
Auto_Fill_Delivery_Address	Autolaunched Flow	Record—Run Before Save	✓	S.V.S.N. TEJA SWARUP	12/7/2025, 06:42 PM
Delivery Agent Route Notification	Autolaunched Flow	Record—Run After Save	✓	S.V.S.N. TEJA SWARUP	12/8/2025, 10:56 AM
Delivery Status Automation	Autolaunched Flow	Record—Run After Save	✓	S.V.S.N. TEJA SWARUP	12/8/2025, 11:28 AM
Late Delivery Alert Flow	Autolaunched Flow	Record—Run After Save	✓	S.V.S.N. TEJA SWARUP	12/8/2025, 09:43 AM

Related Lookup Filters

https://orgfarm-12af98cb1d-dev-ed.develop.my.salesforce-setup.com/one/one.app#/setup/ObjectManager/011gL000003M7Hf/RelatedLookupFilters/view

- 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
- Restriction Rules

**Validation Rules**

3 Items, Sorted by Rule Name

[New](#)

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
Delivery_Date_Not_Past	Delivery Date	Delivery Date cannot be in the past.	✓	S.V.S.N. TEJA SWARUP, 12/7/2025, 5:00 AM
Require_Address_When_Delivered	Top of Page	Delivery Address is required when status is Delivered.	✓	S.V.S.N. TEJA SWARUP, 12/7/2025, 5:01 AM
Require_Agent_When_Out_For_Delivery	Delivery Agent	Delivery Agent is required when status is Out for Delivery.	✓	S.V.S.N. TEJA SWARUP, 12/7/2025, 7:40 AM