PHASE 1:

Objective: To lay the foundational groundwork for the Patient Appointment Management System by formalizing the project's purpose, identifying key players, and mapping the primary business workflow.

1. Project Charter

This document acts as the guiding star of the project, outlining its core purpose, scope, and intended outcomes.

- **Project Title:** Patient Appointment Management System
- **Industry:** Healthcare / Clinic Management
- **Problem Statement:**
  The current appointment scheduling in many small clinics is handled manually through registers or spreadsheets. This results in several operational challenges:
  - Receptionists struggle to track doctors' availability, often leading to overlapping bookings.
  - Patients receive no reminders or confirmations, causing missed appointments.
  - Doctors have no organized view of their daily schedules.
  - Administrators lack real-time reports on appointment trends, cancellations, and resource usage.
    These issues reduce clinic efficiency, increase patient wait times, and create overall dissatisfaction.
- **Proposed Solution:**
  Develop a custom Salesforce-based application called **Patient Appointment Management System** that centralizes all patient, doctor, and appointment data. The system will automate scheduling, prevent overlapping appointments, send confirmation/reminder notifications, and provide real-time dashboards to improve decision-making and operational efficiency.
- **Key Use Cases:**
  - Receptionist creates a new Patient record and books an appointment with a Doctor.
  - The system validates doctor availability and prevents overlapping bookings.

    - Patient automatically receives a confirmation email and a reminder 1 day before the appointment.
    - Doctor views their appointment schedule and marks completed appointments.
    - Admin views dashboards to analyze daily appointments, cancellations, and doctor workload.

2. Identified Stakeholders & Users

The following key users have been identified as the target audience for the application:

- **Receptionist:** Manages patient records, books appointments, and handles cancellations. Needs a fast and simple interface to manage schedules accurately.
- **Doctor:** Provides medical consultation and needs a clear view of upcoming appointments, patient details, and the ability to update appointment status.
- **Patient:** Receives confirmation and reminder communications and attends scheduled appointments.
- **System Administrator:** Configures and maintains the Salesforce system, manages security, profiles, permissions, and supports users.
- **Clinic Administrator:** Reviews reports and dashboards to track clinic performance, appointment trends, and resource allocation.

3. Core Business Process Map

The main workflow that the system will support is as follows:

1. **Patient Registration:** Receptionist registers a new Patient or selects an existing record.
2. **Appointment Booking:** Receptionist books an Appointment with a selected Doctor and time slot.
3. **Availability Validation:** The system checks for conflicts and prevents overlapping appointments for the same Doctor.
4. **Notification Trigger:** The system sends an appointment confirmation email immediately and a reminder email 1 day before the appointment.
5. **Consultation:** On the appointment date, the Doctor attends the patient and marks the Appointment as “Completed.”

6. **Performance Review:** The Administrator views dashboards to analyze the number of appointments, cancellations, and doctor workload trends.

PHASE 2:

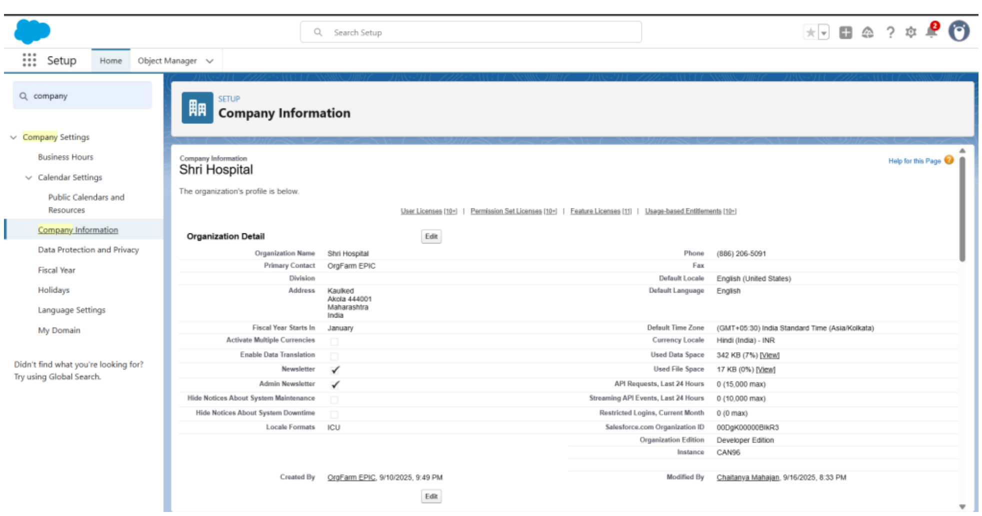
Objective: Prepare the Salesforce Developer Org, users, security model, and baseline settings for the appointment booking system.

1. Salesforce Edition

- Used **Developer Edition Org (free)** from developer.salesforce.com.
- Includes support for Custom Objects, Flows, Validation Rules, Lightning App Builder, and Email Templates.
- No Sandbox in Developer Edition, so all configuration is done directly in the Dev Org.

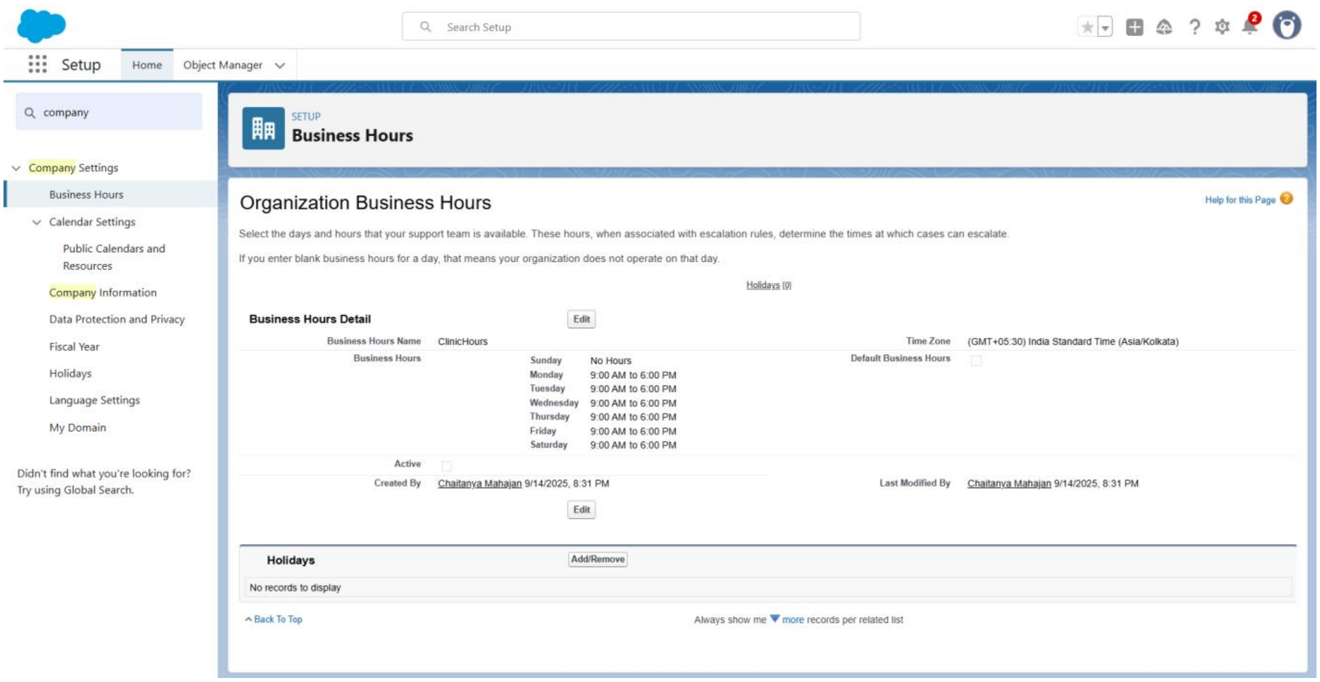
2. Company Profile Setup

- **Path:** Setup → Company Settings → Company Information → Edit
- Updated values:
    - Organization Name → Shri Hospital
    - Default Locale → English (India)
    - Currency → INR
    - Time Zone → IST (Asia/Kolkata)
- Purpose: ensures proper formatting of dates, numbers, currency in records and reports.



3. Business Hours & Holidays

- **Path:** Setup → Company Settings → Business Hours → New
    - Business Days → Monday–Saturday
    - Time → 9:00 AM – 6:00 PM



- **Path:** Setup → Company Settings → Holidays → New Holiday
    - Added holidays like Republic Day, Diwali, Independence Day.

4. Fiscal Year

- Left as default (Standard fiscal year starting January).

5. Users Setup & Licenses

- **Path:** Setup → Users → Users → New User
- Created 3 test users for role-based access testing:
    - **Admin (you)** — full system access
    - **Receptionist** — can create/manage appointments and patients
    - **Doctor** — can only view/edit their appointments
- License used: **Salesforce Platform** (suitable for custom-object-only users).

6. Profiles

- Cloned the **Standard Platform User** profile to create custom profiles:
  - **Admin Profile:** Full CRUD + Setup access
  - **Receptionist Profile:** CRED on Patient, Appointment; Read on Doctor
  - **Doctor Profile:** Read/Edit on Appointment; Read on Patient; Read on Doctor
- **Path:** Setup → Profiles → Clone → Configure Object Settings.

7. Roles

- Defined a role hierarchy to control record visibility:
  - **Admin → Receptionist → Doctor**
- **Path:** Setup → Users → Roles → Set Up Roles → New Role
- Purpose: ensures higher roles (Receptionist/Admin) can see records owned by lower roles (Doctor).

8. Permission Sets

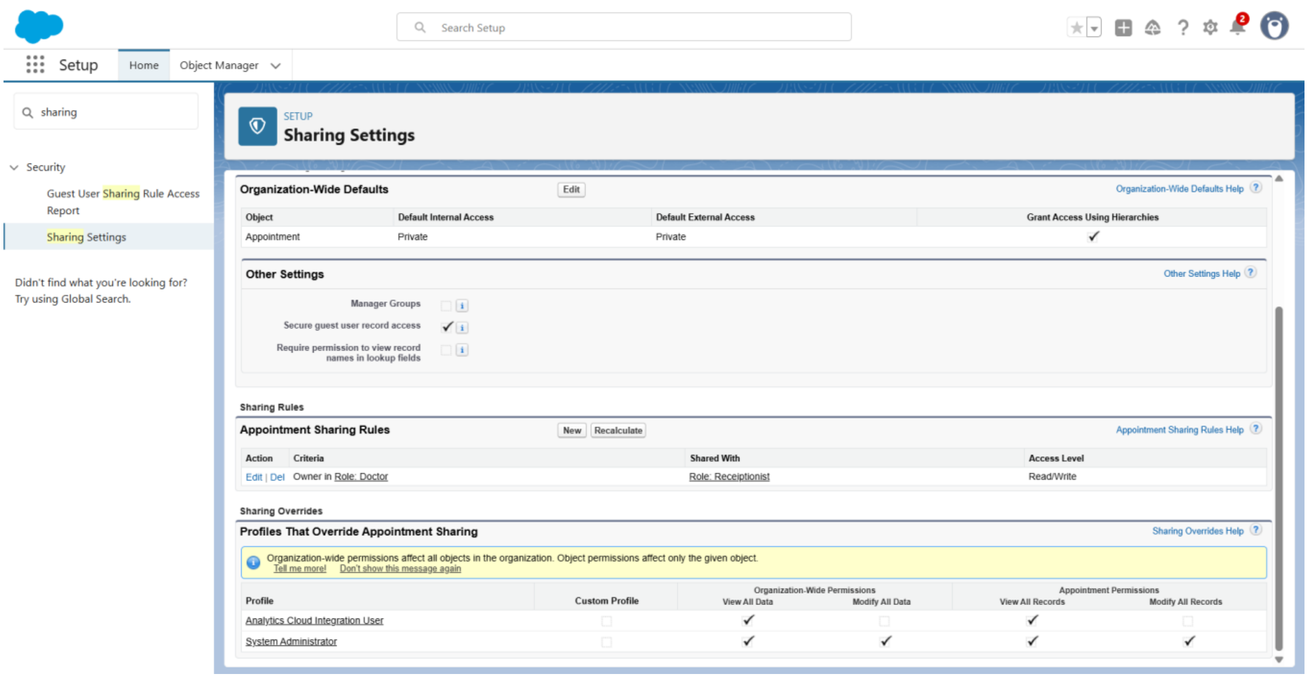
- (Optional) Created a **Flow User Permission Set** to give Flows-related permissions and assigned it to all test users.
- **Path:** Setup → Permission Sets → New.

9. Org-Wide Defaults (OWD)

- **Path:** Setup → Security → Sharing Settings → Edit
- Set sharing model:
  - Doctor__c = Public Read Only
  - Patient__c = Public Read Only
  - Appointment__c = Private
- Ensures records are only visible to their owners unless shared via role hierarchy or rules.

10. Sharing Rules

- Created a sharing rule to allow Receptionists to access Doctor and Appointment records owned by Doctors.
- **Path:** Setup → Security → Sharing Settings → Appointment Sharing Rules → New
  - **Owned By:** Role → Doctor
  - **Share With:** Role → Receptionist
  - **Access Level:** Read/Write.



11. Login Access Policies

- No changes made (defaults used).

PHASE 3:

Objective: Define objects, fields, relationships, and UI layouts for clinic data.

1. Custom Objects

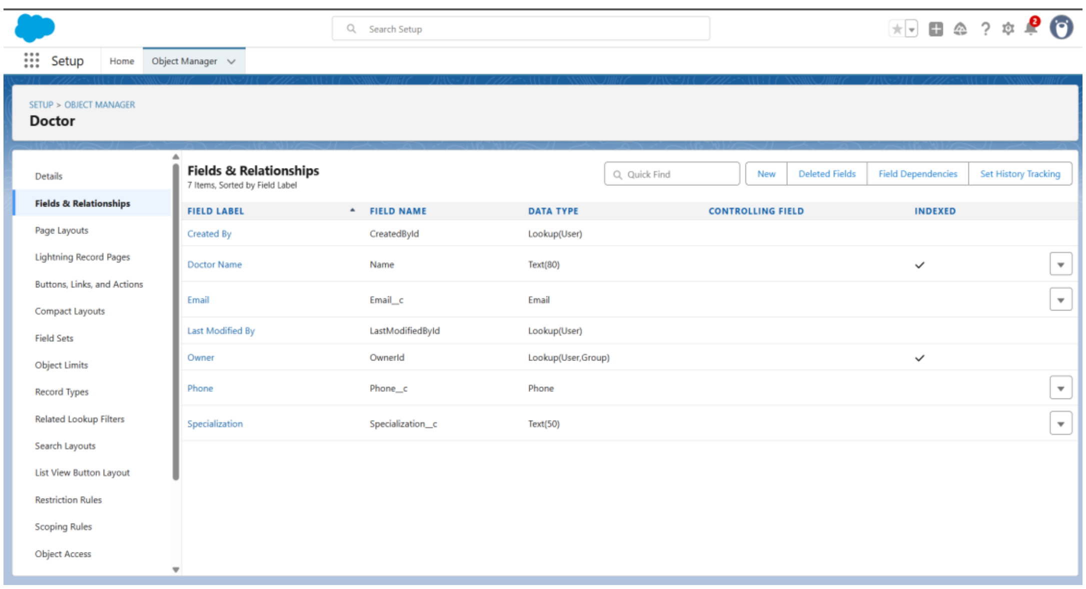
Path: Setup → Object Manager → Create → Custom Object**

- **Doctor__c**
  - Label: Doctor, Plural: Doctors, Record Name: Name (Text)
- **Patient__c**
  - Label: Patient, Plural: Patients, Record Name: Name (Text)
- **Appointment__c**
  - Label: Appointment, Plural: Appointments
  - Record Name: Auto Number → Format APT-{00000000}

2. Custom Fields

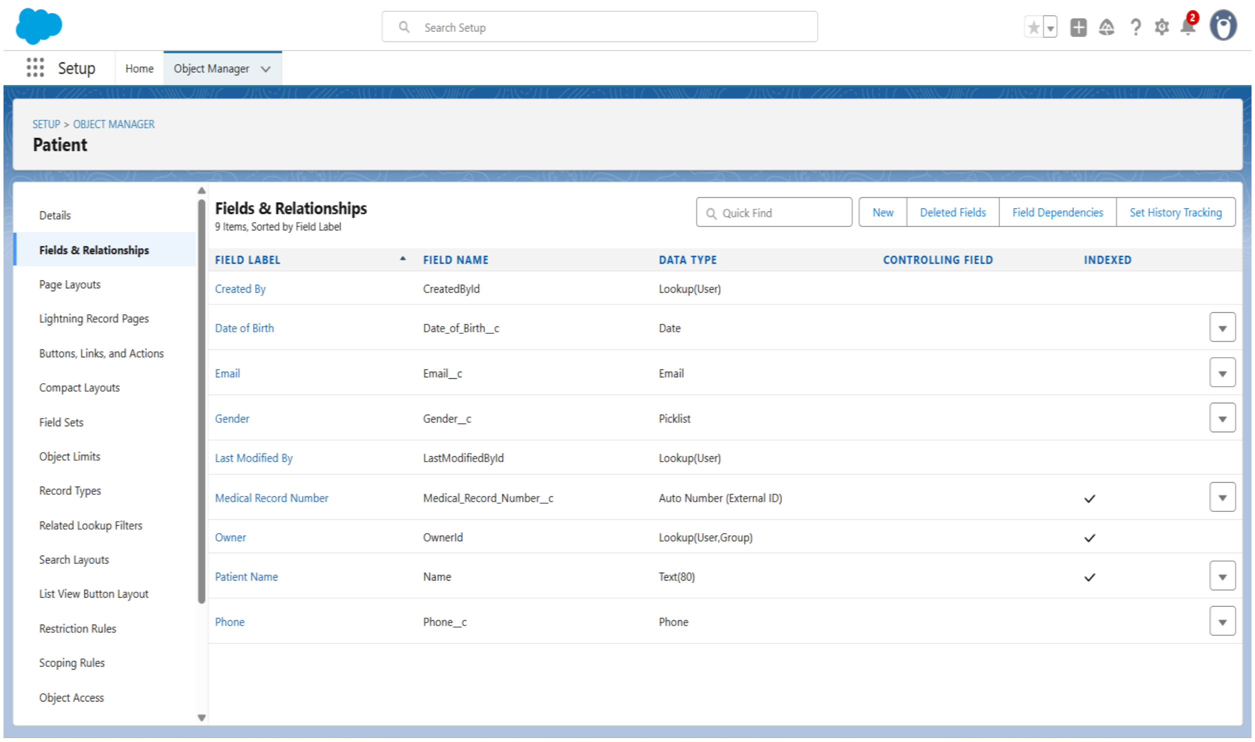
Doctor

- Specialization__c — Picklist (Dentist, Cardiologist, Orthopedic, General Physician, Pediatrician, Other)
- Contact_Number__c — Phone
- Email__c — Email



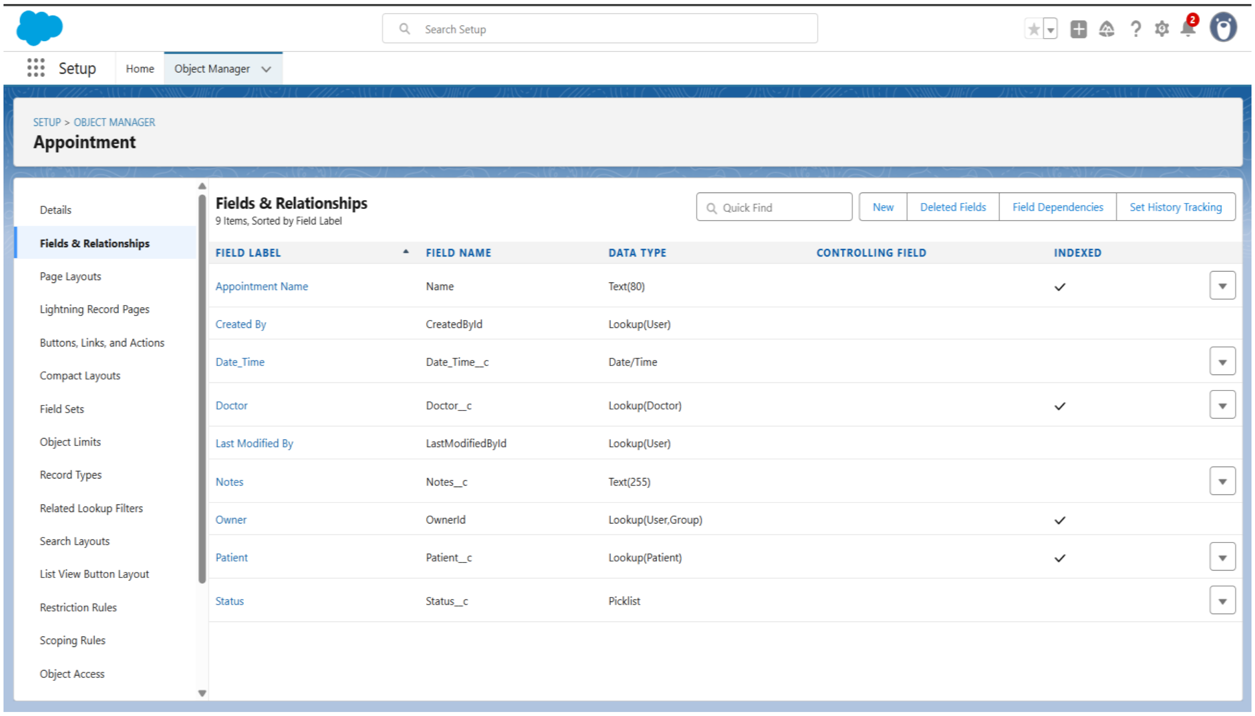
Patient

- Date_of_Birth__c — Date
- Gender__c — Picklist (Male, Female, Other)
- Contact_Number__c — Phone
- Email__c — Email



Appointment

- Appointment_Date_Time__c — Date/Time (Required)
- Status__c — Picklist (Scheduled (default), Completed, Cancelled)
- Doctor__c — Lookup (Doctor) (Required)
- Patient__c — Lookup (Patient) (Required)
- Notes__c — Long Text Area (32000)



3. Page Layouts & Related Lists

- Edited Appointment layout: placed Appointment Date/Time, Status, Doctor, Patient, Notes at top.
- Edited Doctor layout: added Specialization, Phone, Email, and Appointments related list.
- Edited Patient layout: added Age, Gender, Phone, Email, and Appointments related list.
- **Path:** Object Manager → [Object] → Page Layouts → Edit.

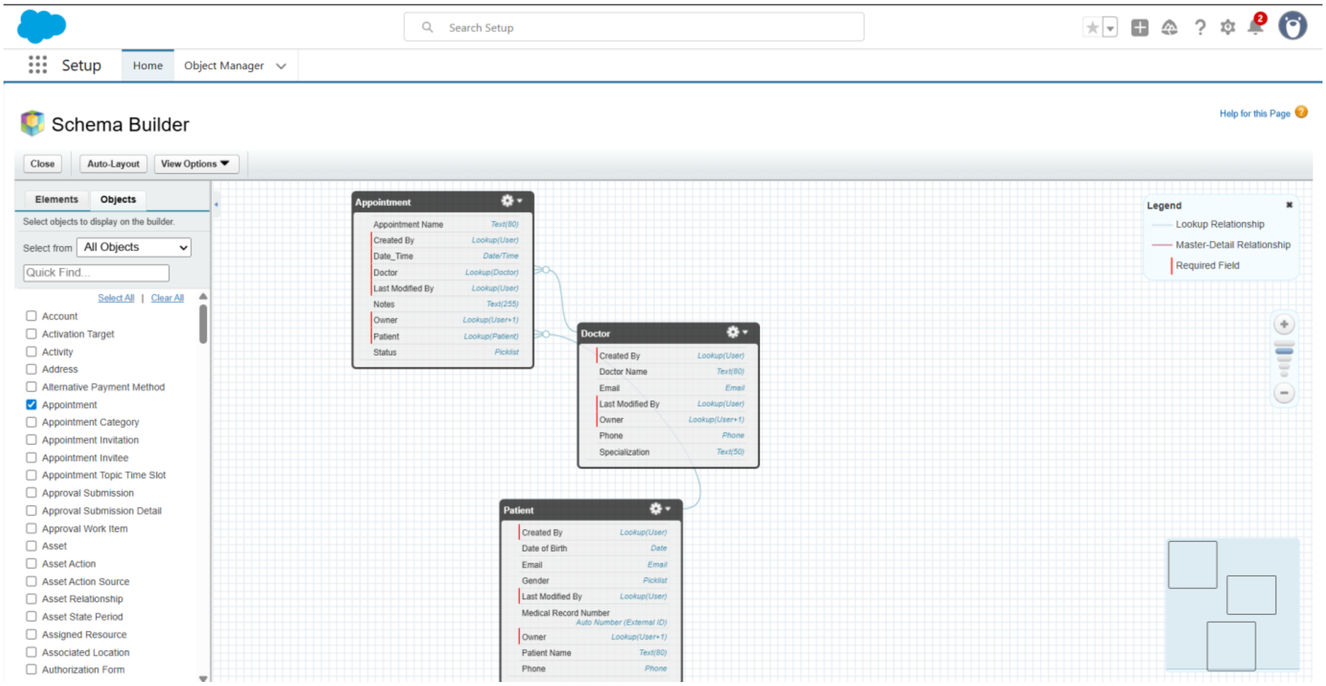
4. Compact Layouts

- Appointment: DateTime, Doctor, Patient, Status.
- Doctor: Name, Specialization, Phone.
- Patient: Name, Age, Phone.
- **Path:** Object Manager → [Object] → Compact Layouts → New → Assign.

5. Schema Builder

- **Path:** Setup → Schema Builder

- Dragged Doctor, Patient, Appointment objects to verify relationships visually as ER diagram.

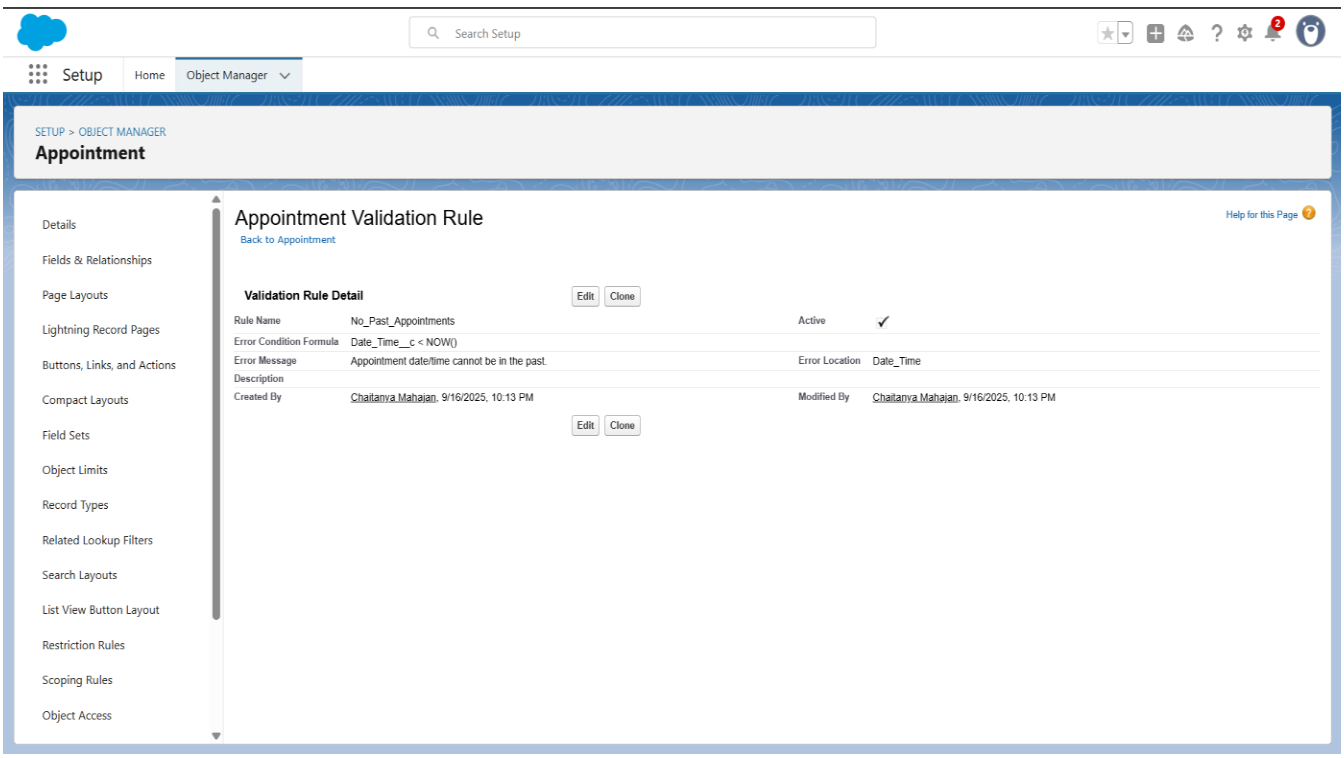


PHASE 4:

Objective: Prevent invalid data and send appointment confirmations and reminders.

A. Validation Rule — No Past Dates

- **Path:** Object Manager → Appointment → Validation Rules → New
- **Name:** No_Past_Appointments
- **Formula:**
- Appointment_Date_Time__c < NOW()
- **Error:** Appointment date/time cannot be in the past.



B. Prevent Overlapping Appointments — Before-Save Flow

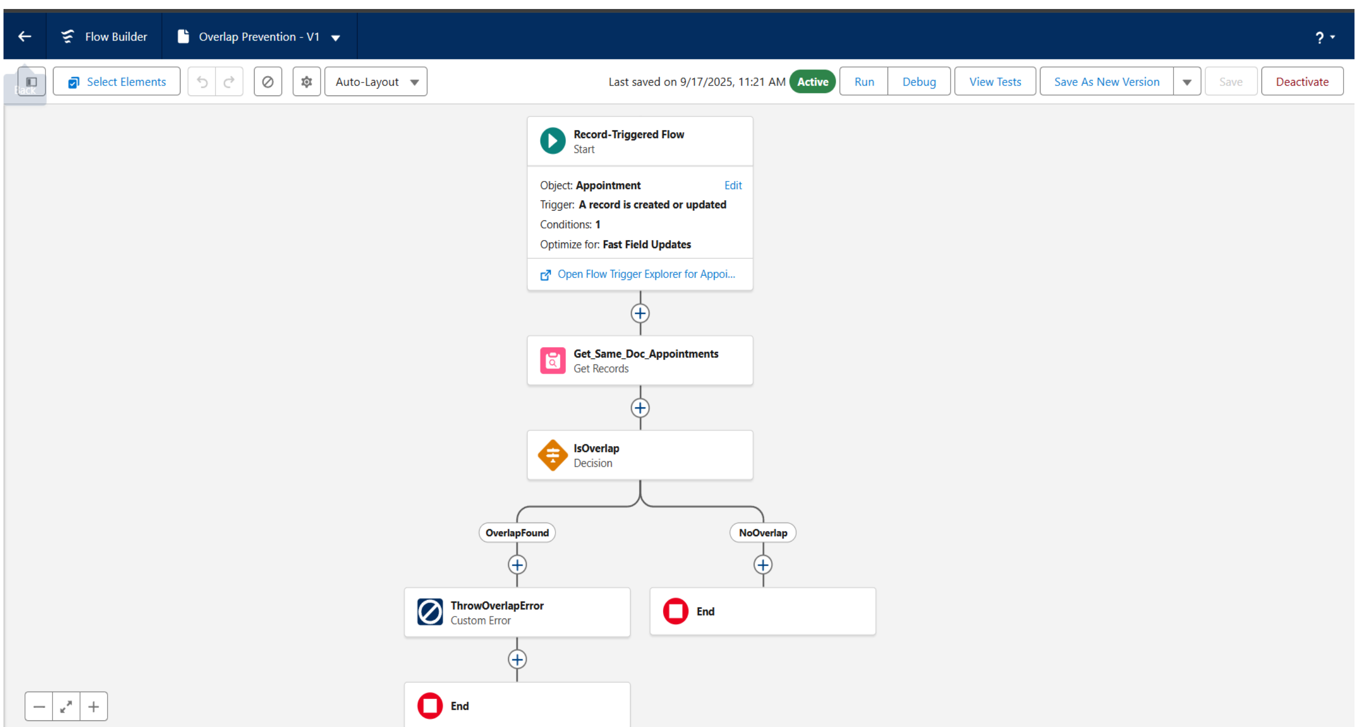
- **Flow:** Record-Triggered (Before Save) on Appointment
- Trigger: When record is created or updated

- Logic:
    1. **Get Records** — Appointment where
        - Doctor__c = $Record.Doctor__c
        - Appointment_Date_Time__c = $Record.Appointment_Date_Time__c
        - Id != $Record.Id
        - Status__c != 'Cancelled'
    2. **Decision** — if a record is found
    3. **Assignment** — $Record.Overlapping__c = TRUE
- Save as **FTB_Appointment_Overlap_Check** → Activate

Validation Rule to block save:

Overlapping__c = TRUE

→ Error: "This doctor already has an appointment at the selected date/time."



C. Appointment Confirmation Emails — After-Save Flow

- Created Lightning Email Template **Appointment_Confirmation_Template** with merge fields:
  - Patient Name, Doctor Name, Appointment DateTime, Notes
- **Flow:** Record-Triggered (After Save) on Appointment
- Trigger: When created
- Entry Condition: Status__c = 'Scheduled'
- Immediate Action: Send Email (Confirmation template) → Recipient: $Record.Patient__r.Email
- Scheduled Path: Reminder_1_Day_Before
  - Time Source: Appointment_Date_Time__c
  - Offset: -1 Day
  - Action: Send Email using Appointment_Reminder_Template
- Save as **AFT_Send_Appointment_Emails** → Activate