**Phase 3: Data Modeling & Relationships**

**1. Standard & Custom Objects:**

**i) Standard Objects:**

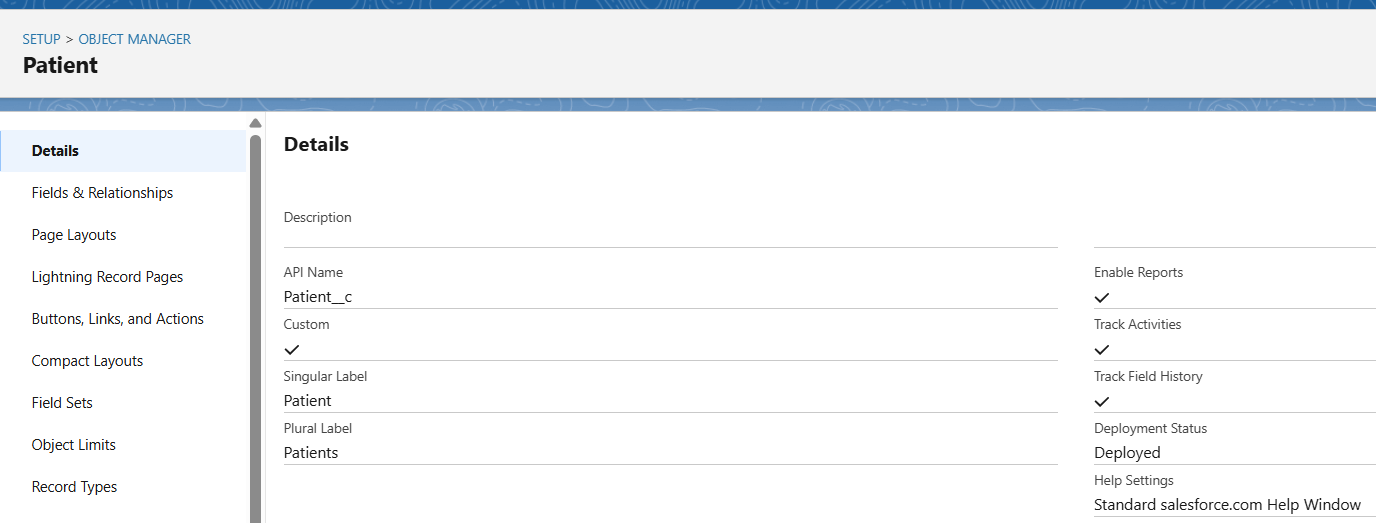
🔹 Standard Objects Used

* User → To manage system users (Doctors, Nurses, Receptionists, Billing Officers, etc.)
* Account / Contact (optional if used for Patients/Doctors)
* Report, Dashboard → For analytics

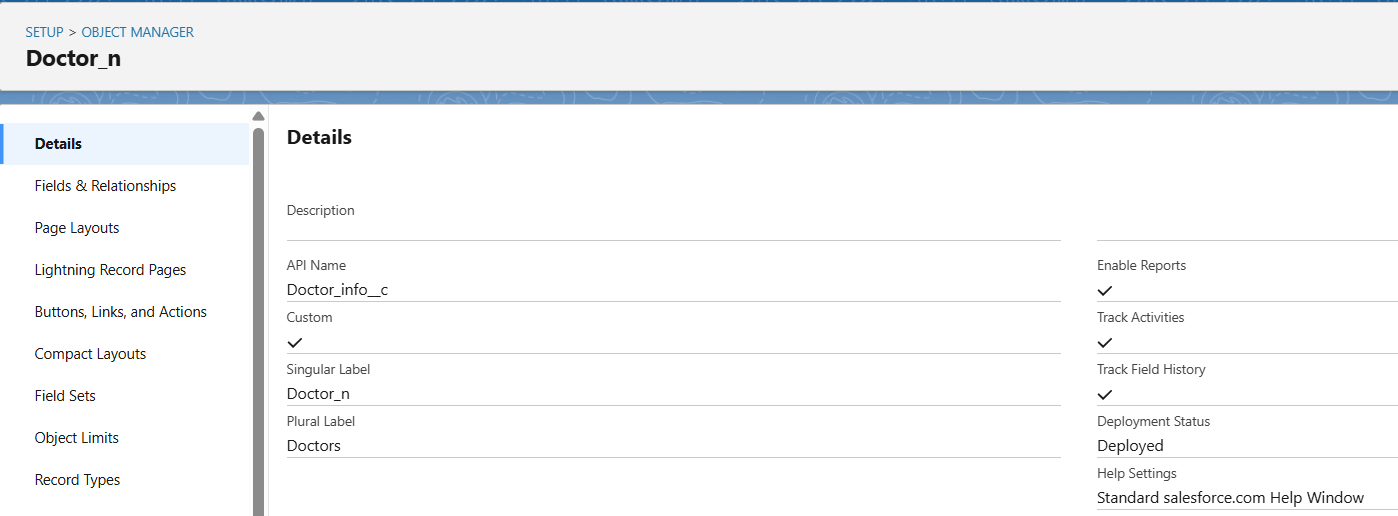
**ii) Custom Objects:**

🔹 Custom Objects Created in MediConnect CRM

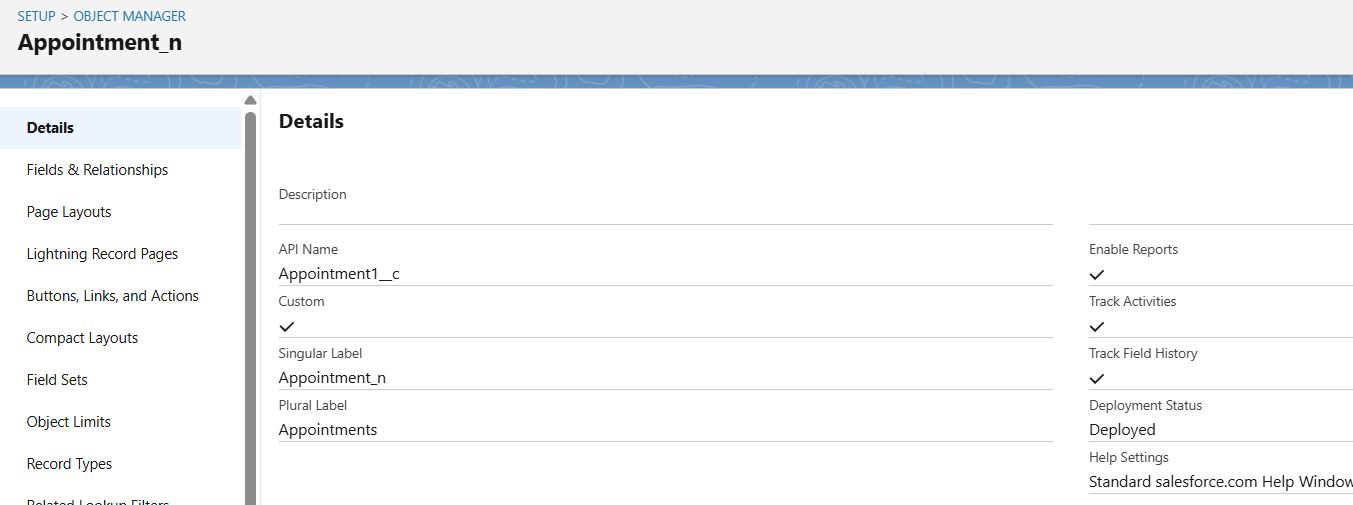
1. **Patient\_\_c**
   * Stores patient details (Name, Age, Gender, Contact, Medical History).
   * *Reason*: Hospitals need centralized patient information for appointments, billing, and treatment tracking.



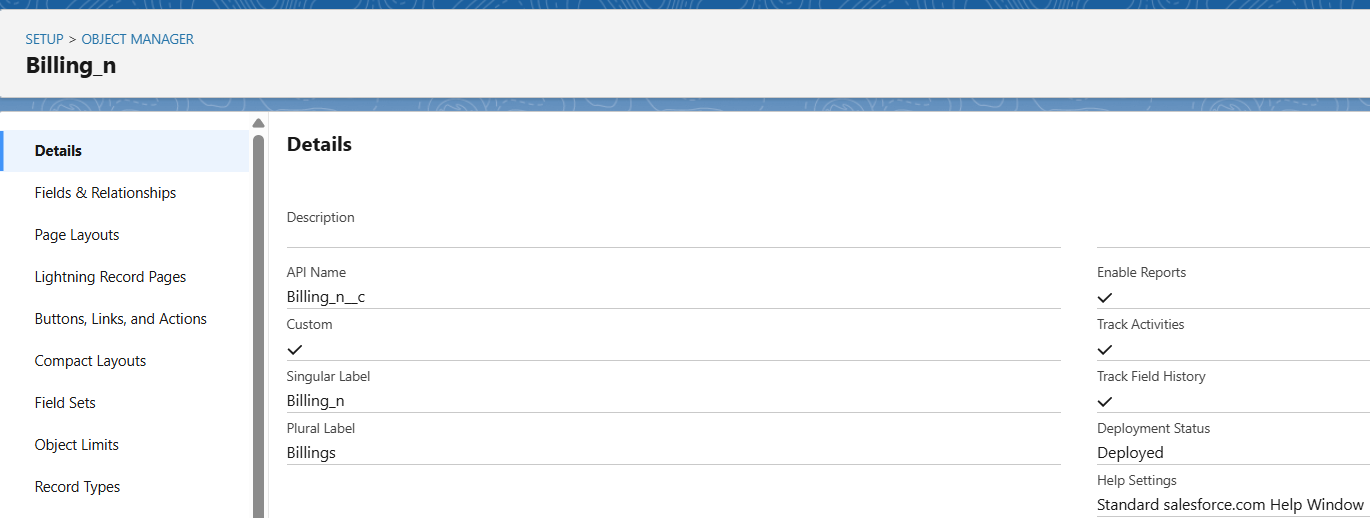
1. **Doctor\_\_c**
   * Stores doctor details (Name, Specialization, Contact, Availability).
   * *Reason*: To manage doctors working in hospital and link them to patients/appointments.



1. **Appointment\_\_c**
   * Stores appointment details (Date, Time, Patient, Doctor, Status).
   * *Reason*: To schedule and track appointments between patients and doctors.



1. **Billing\_\_c**
   * Stores billing details (Bill ID, Amount, Payment Mode, Bill Date, Patient).
   * *Reason*: For financial management and tracking payments of patients.



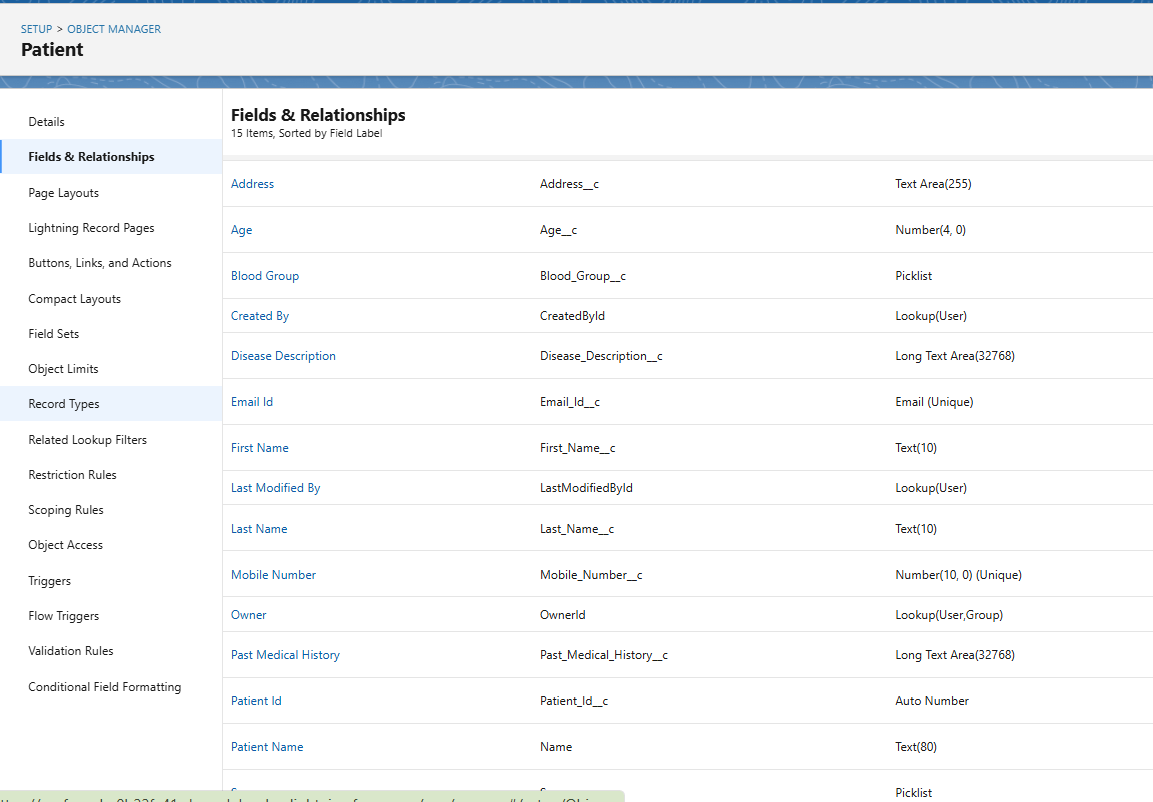
✅ **Why Custom Objects?**  
Because Salesforce standard CRM is more sales-focused (Leads, Opportunities, Cases). For a **Hospital CRM**, we need domain-specific objects like Patient, Doctor, Appointment, Billing.

**2.Fields:**

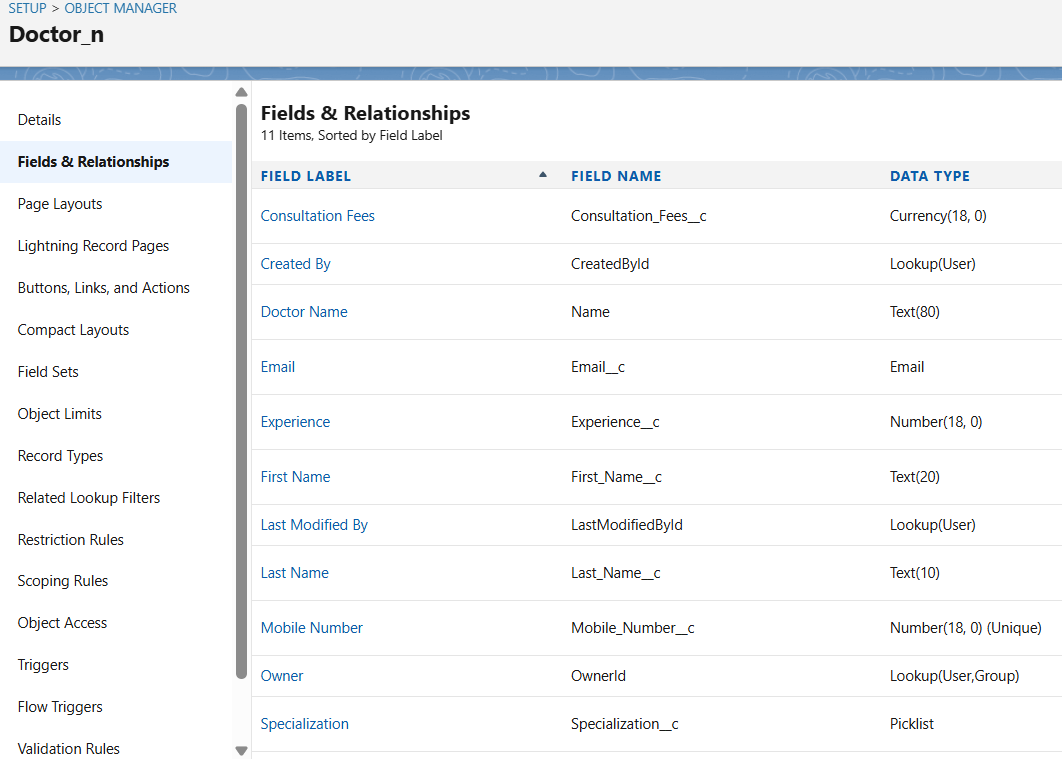
Each object has both **standard fields** (Name, Owner, Created By) and **custom fields** we added.

**🔹 Example Fields**

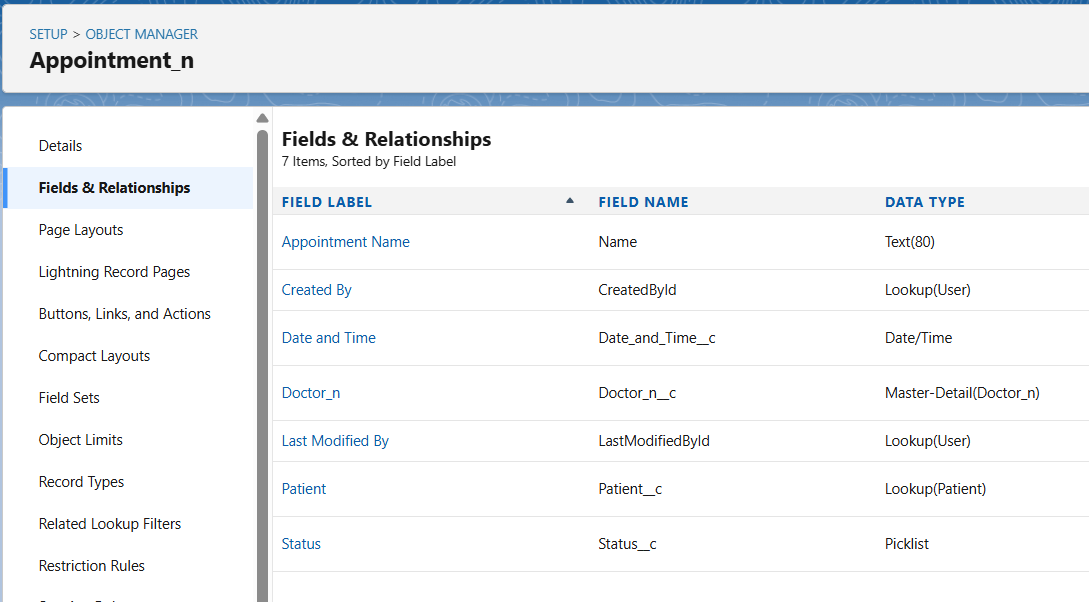
* **Patient\_\_c** → Patient ID (Auto Number), Name, Age, Gender, Phone, Address, Blood Group (picklist), Disease Discription (Long Text).



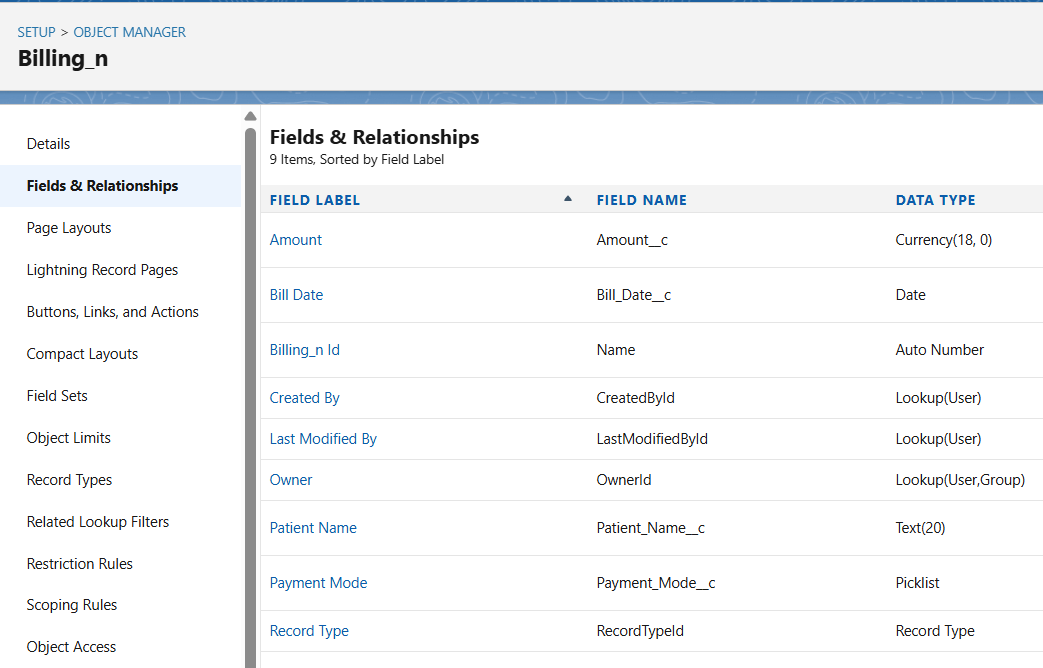
* **Doctor\_\_c** → Doctor ID, Name, Specialization (Picklist), Contact, Experience



* **Appointment\_\_c** → Appointment ID, Date, Time, Status (Picklist), Patient (MDR), Doctor (MDR).



* **Billing\_\_c** → Billing ID, Bill Date, Amount (Currency), Payment Mode (Picklist), Patient (Lookup).



* **Treatment\_\_c** → Doctor (MDR), Patient (MDR), Notes, Medicine, Treatment Date.

**3. Record Types:**

Purpose: Record Types allow us to offer different business processes, picklist values, and page layouts for the same object.

**Record Types in Billing\_n:**

**Purpose**

Record Types allow different **business processes** and **picklist values** for the same object.  
In hospitals, billing can happen in two different ways:

1. **Self Billing** – Patient pays directly (Cash/UPI/Card).
2. **Insurance Billing** – Payment is made via Insurance.

So we created **two Record Types** in the **Billing\_n object**.

**🔹 Record Type 1: Self Billing**

* **Fields**:
  + Billing Id
  + Patient Name
  + Amount
  + Bill Date
  + Payment Mode → (Cash, UPI, Debit Card, Credit Card, Net Banking)
* **Use Case**: Direct payment by patient at hospital counter.
* **Page Layout**: Shows **all payment options** for flexibility.

**🔹 Record Type 2: Insurance Billing**

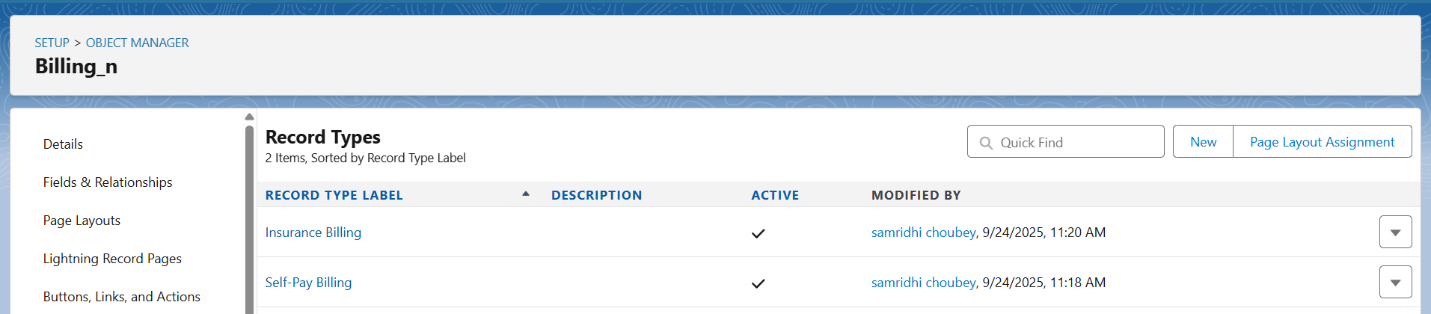
* **Fields**:
  + Billing Id
  + Patient Name
  + Amount
  + Bill Date
  + Payment Mode → Only **Insurance**
  + Insurance Provider (new field)
  + Policy Number (new field)
* **Use Case**: For patients covered under health insurance.
* **Page Layout**: Simplified – only insurance-related fields visible.

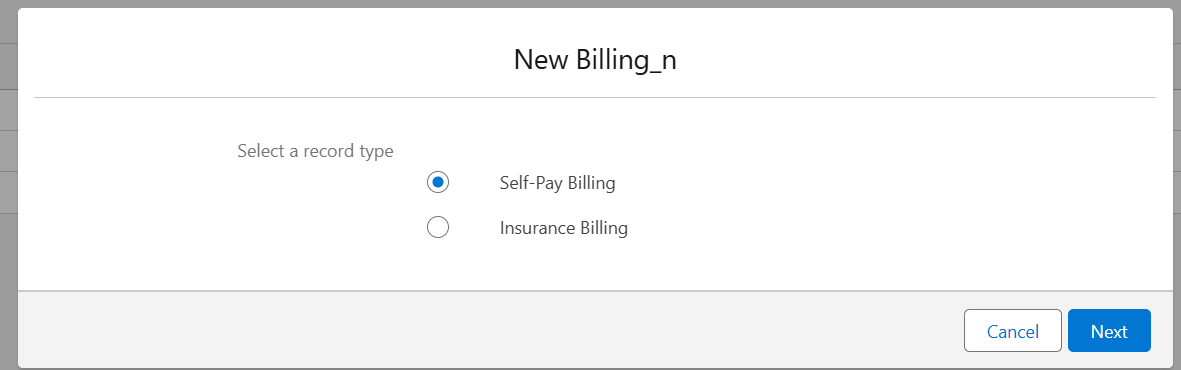
**🔹 Why Record Types Here?**

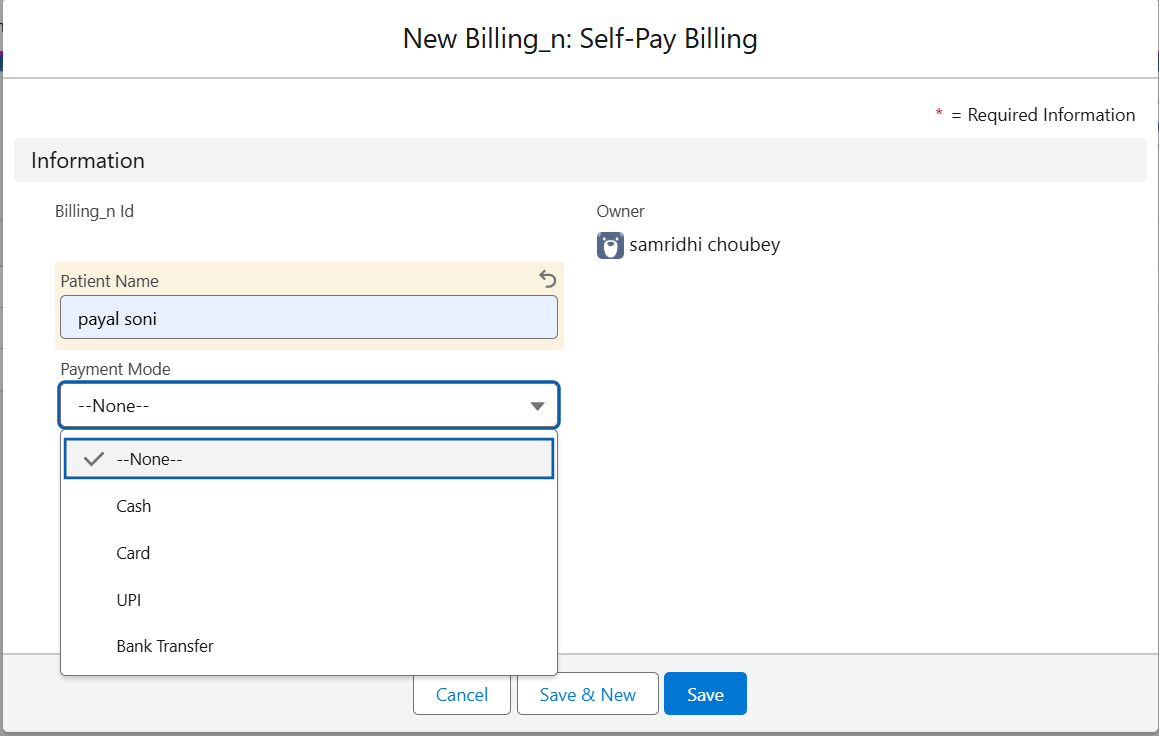
* **Different Payment Processes** need different field visibility and picklist options.
* Doctors/Receptionists see only relevant fields, avoiding confusion.
* Improves **data accuracy**: No chance of entering “Cash” for Insurance Billing.

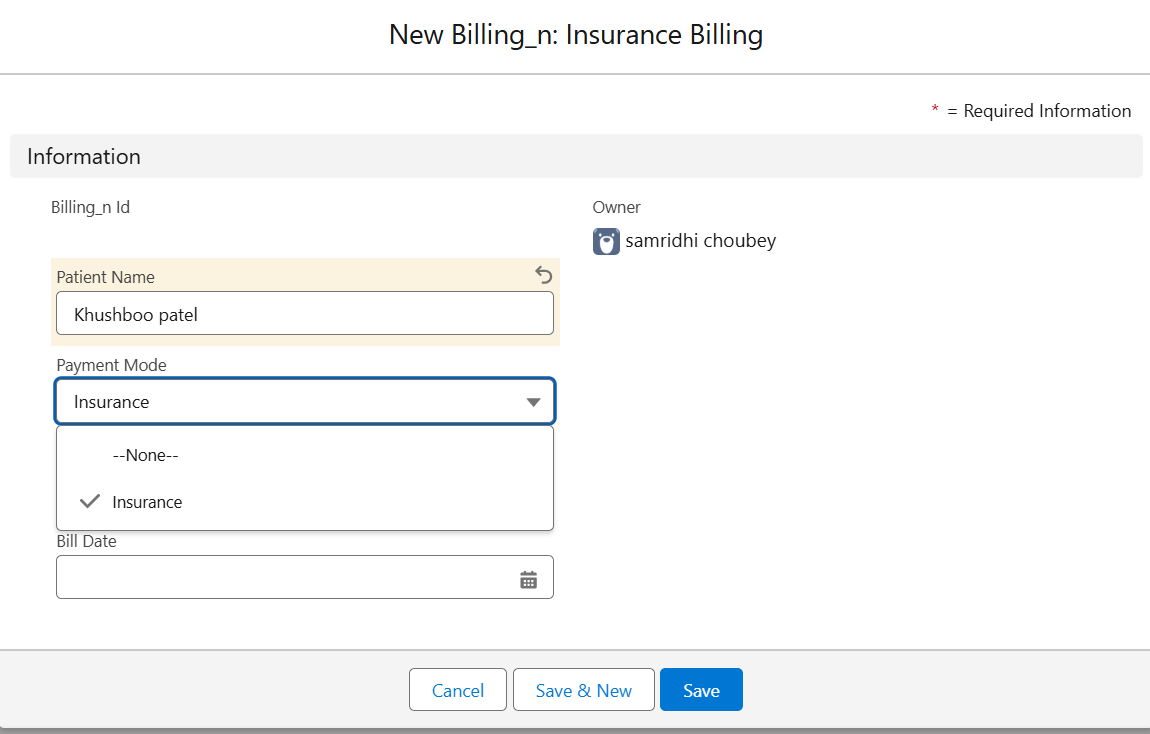
**Example**:

* Patient “Payal Soni” pays in Cash → **Self Billing record type**.
* Patient “Khusboo patel” pays through ICICI Lombard Insurance → **Insurance Billing record type**.

****

****

****

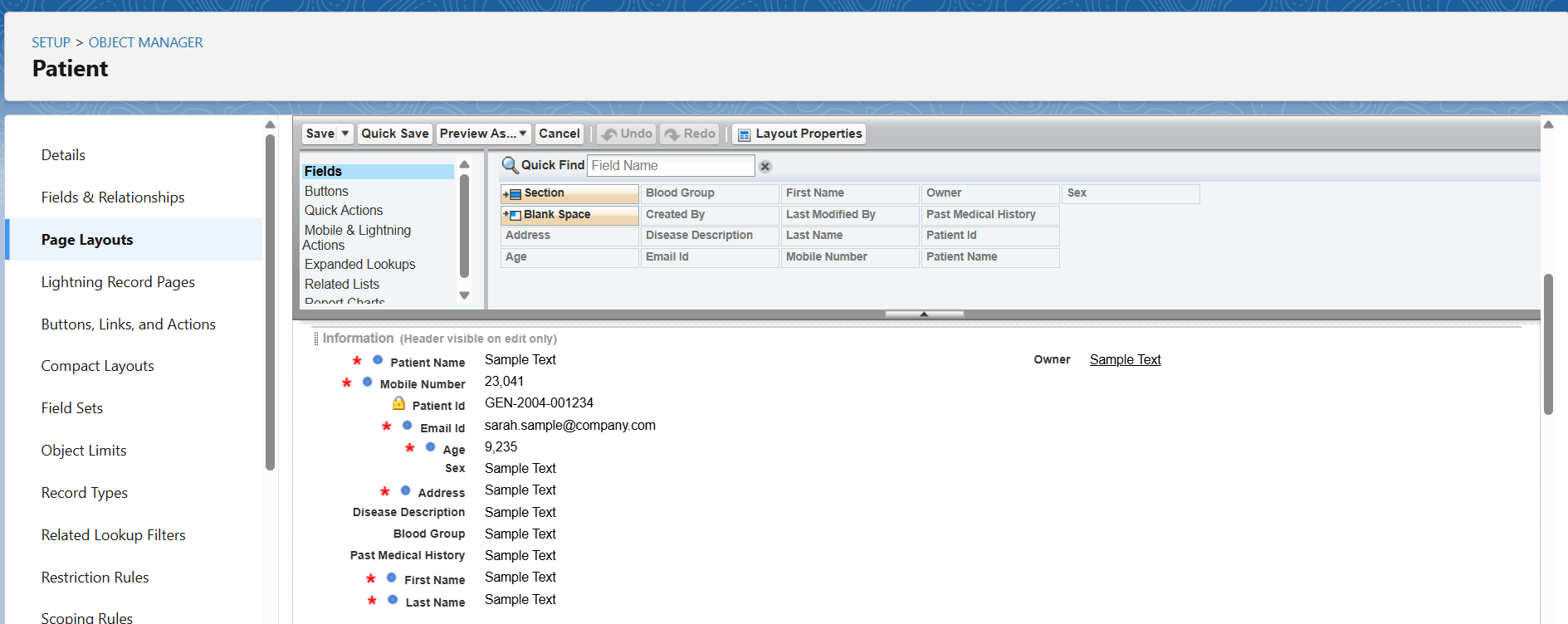
****

**4. Page Layouts:**

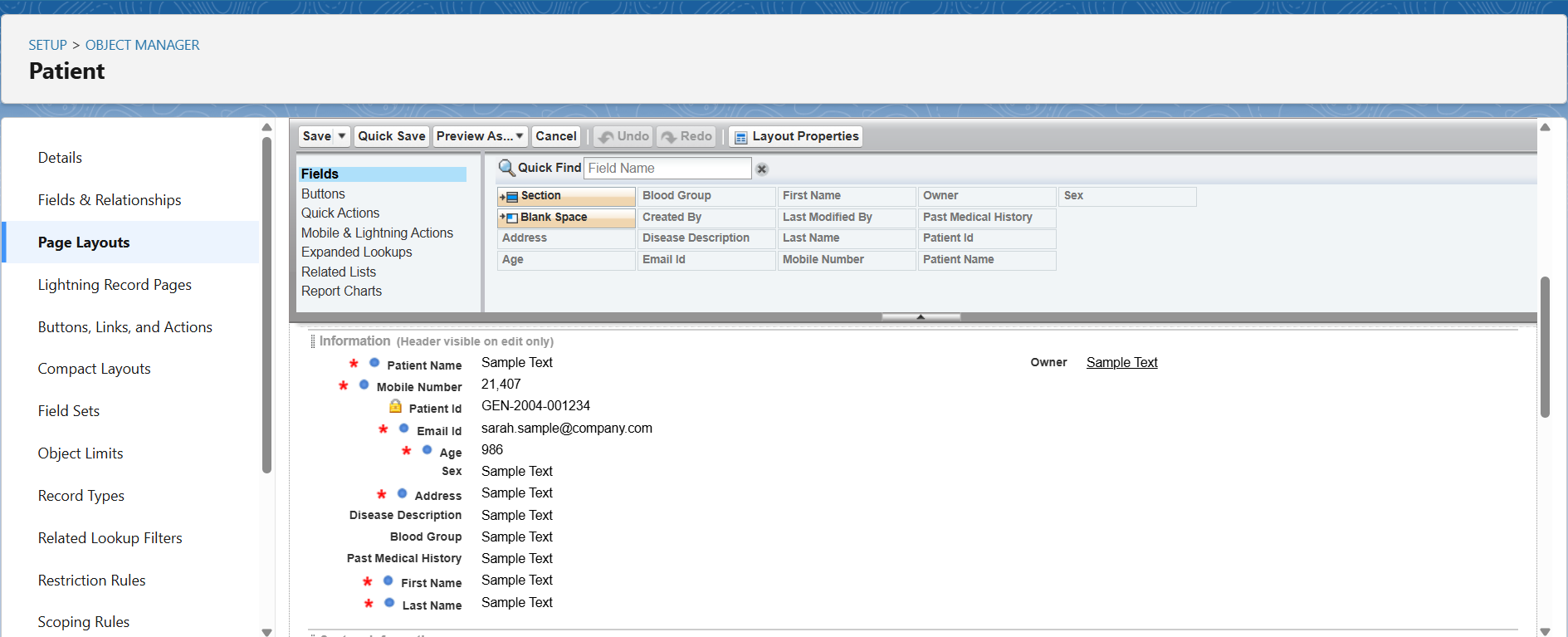
Page Layouts define which fields are visible/editable on a record page.

**Layouts in MediConnect:**

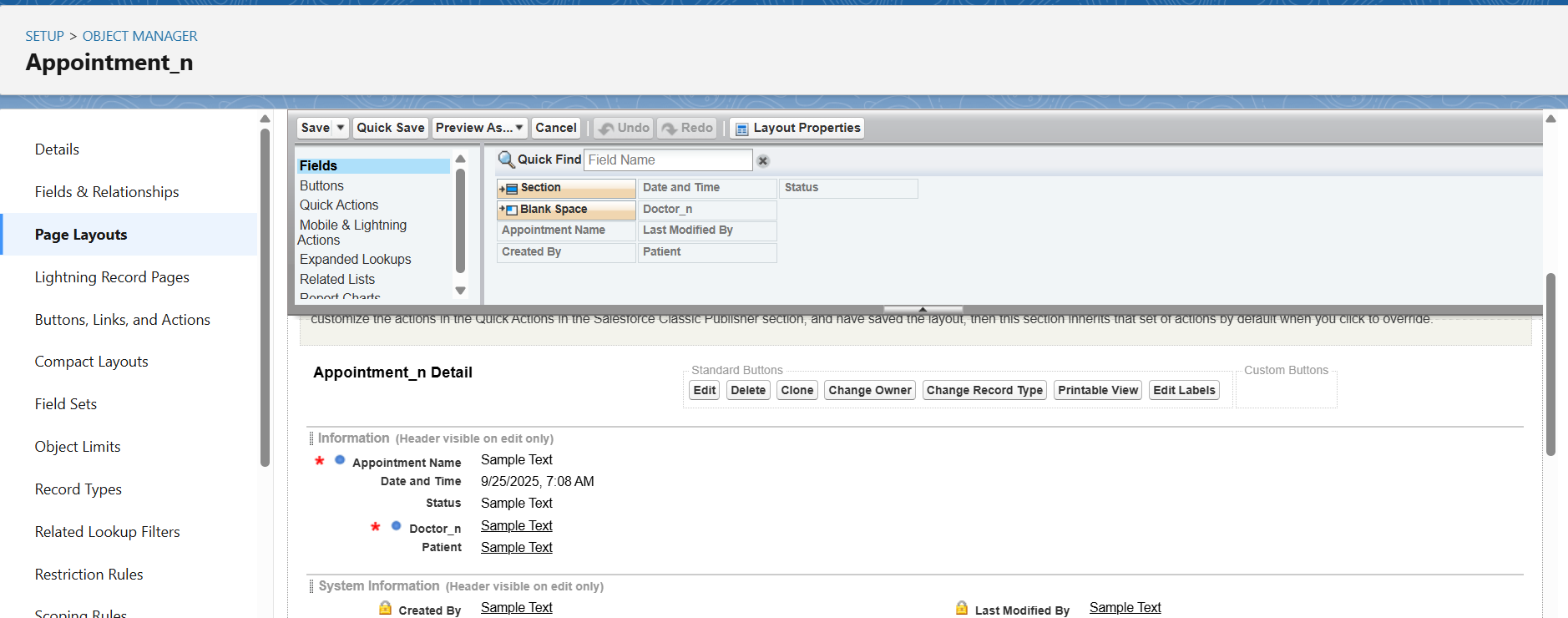
* **Patient Layout** → Shows demographics + medical history.



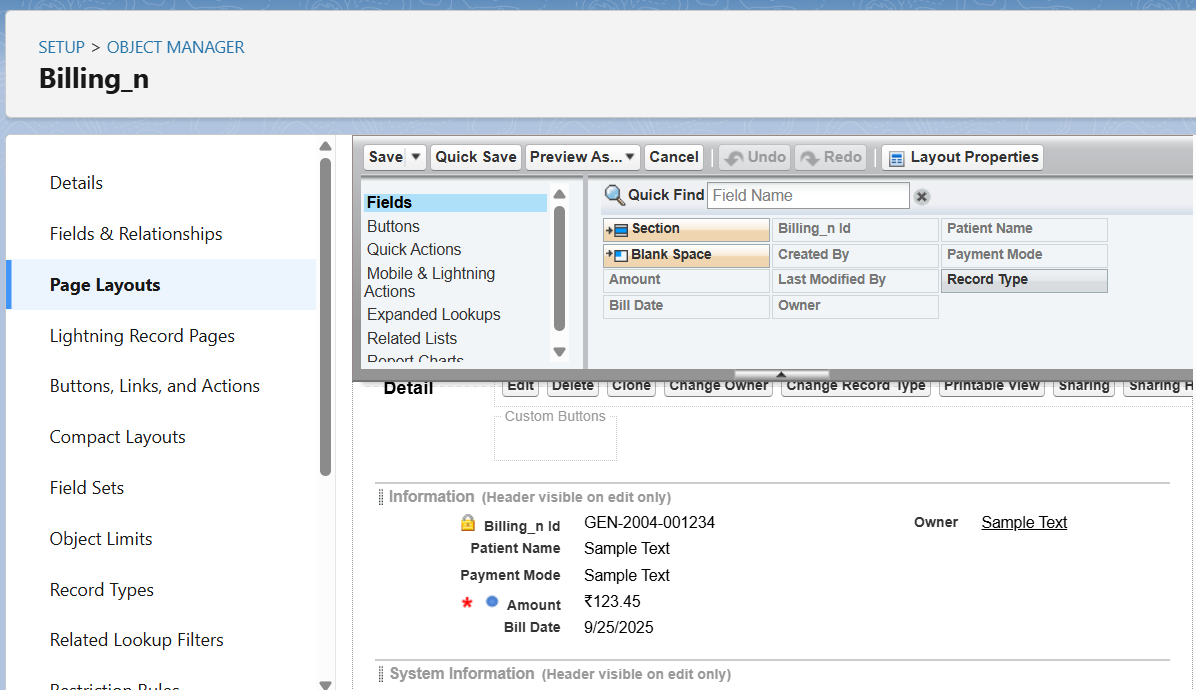
* **Doctor Layout** → Shows specialization + contact info.



* **Appointment Layout** → Date, Time, Status, Linked Patient & Doctor.



* **Billing Layout** → Payment details and patient link.

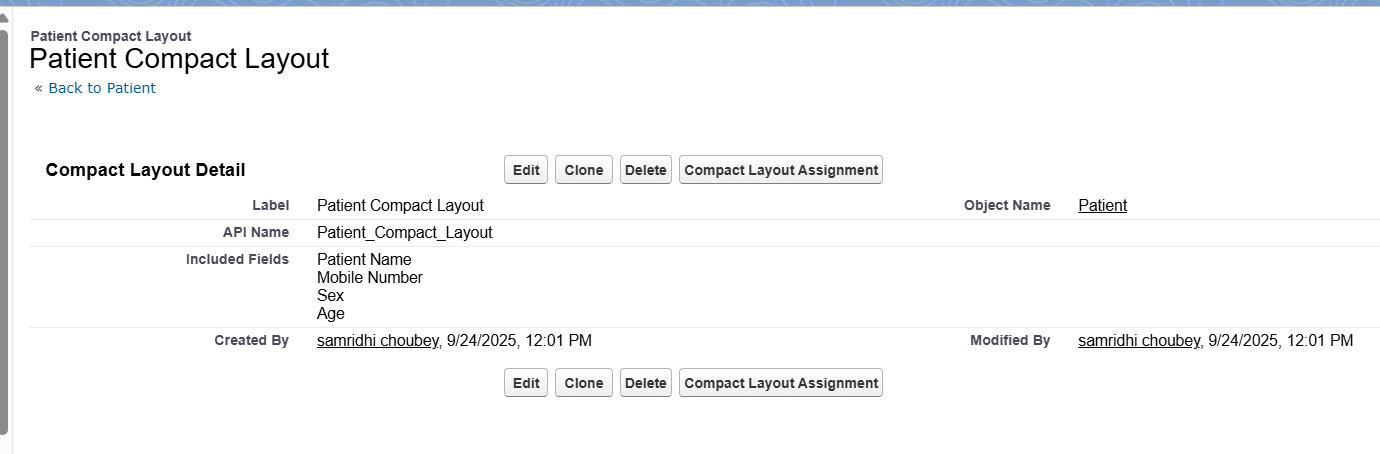


**5. Compact Layouts:**

Compact Layouts show key fields in the record header (highlight panel).

**Example Compact Layouts:**

* **Patient** → Name, Age, Gender, Phone.



*Reason*: Provides a quick snapshot for users (Doctors see Patient Name + Age immediately).

**6. Schema Builder**

* **Schema Builder** visually represents objects and their relationships.
* Building two custom objects Nurse and Treatment using Schema Builder,

**1. Nurse (Custom Object)**

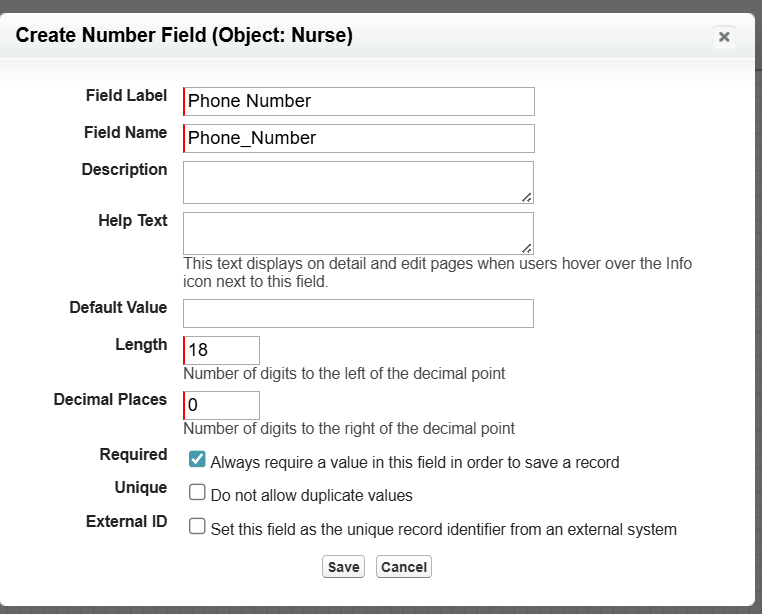
* **Record Name**: Nurse Name (Text)
* **Additional Fields**:
  + Nurse ID (Auto Number: NUR-{0000})
  + Contact Number (Phone)
  + Shift (Picklist: Morning, Evening, Night)

**🔹 Relationships:**

* **Lookup → Doctor**
  + Reason: Each Nurse works under a Doctor. (One Doctor can have many Nurses).
* **Lookup → Patient (optional)**
  + Reason: If Nurses are directly assigned to Patients for care.

**Use Case in MediConnect**:

* Helps track which Doctor supervises which Nurse.
* Enables assigning specific Nurses to patients for ongoing treatment or monitoring.

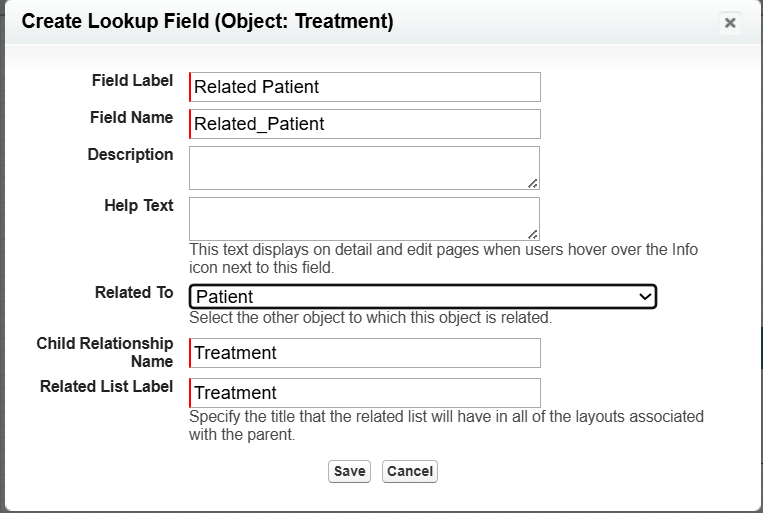


**2. Treatment (Custom Object)**

* **Record Name**: Treatment ID (Auto Number: TRT-{0000})
* **Additional Fields**:
  + Treatment Name (Text → e.g., Chemotherapy, Physiotherapy)
  + Treatment Date (Date)
  + Description (Long Text Area)
  + Cost (Currency)

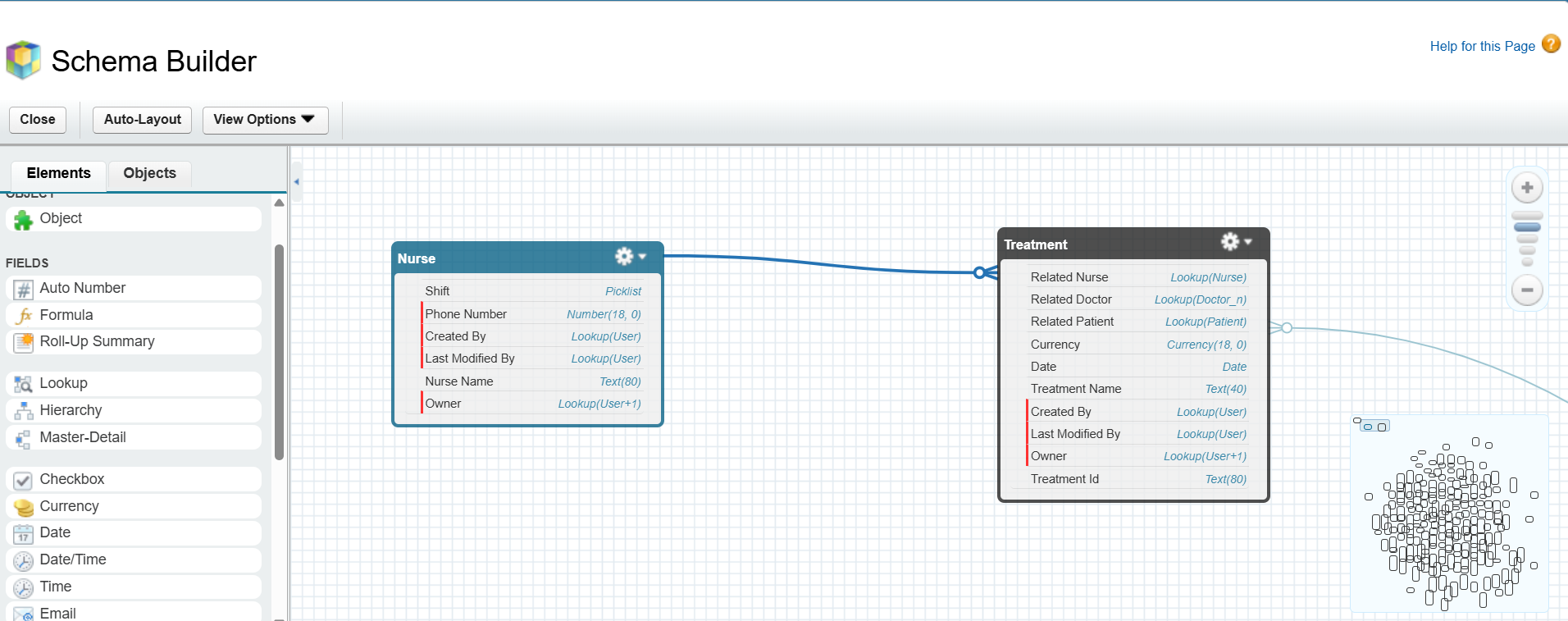
**🔹 Relationships:**

* **Lookup → Appointment**
  + Reason: A Treatment is linked to an Appointment (so we know when and why treatment happened).
* **Lookup → Nurse**
  + Reason: A Nurse may administer the treatment (one Nurse can handle many Treatments).



**Use Case in MediConnect**:

* Records exact treatment given to a Patient during an Appointment.
* Links the treatment to the Nurse for accountability and tracking.
* Nurse and Treatment (Custom Objects)



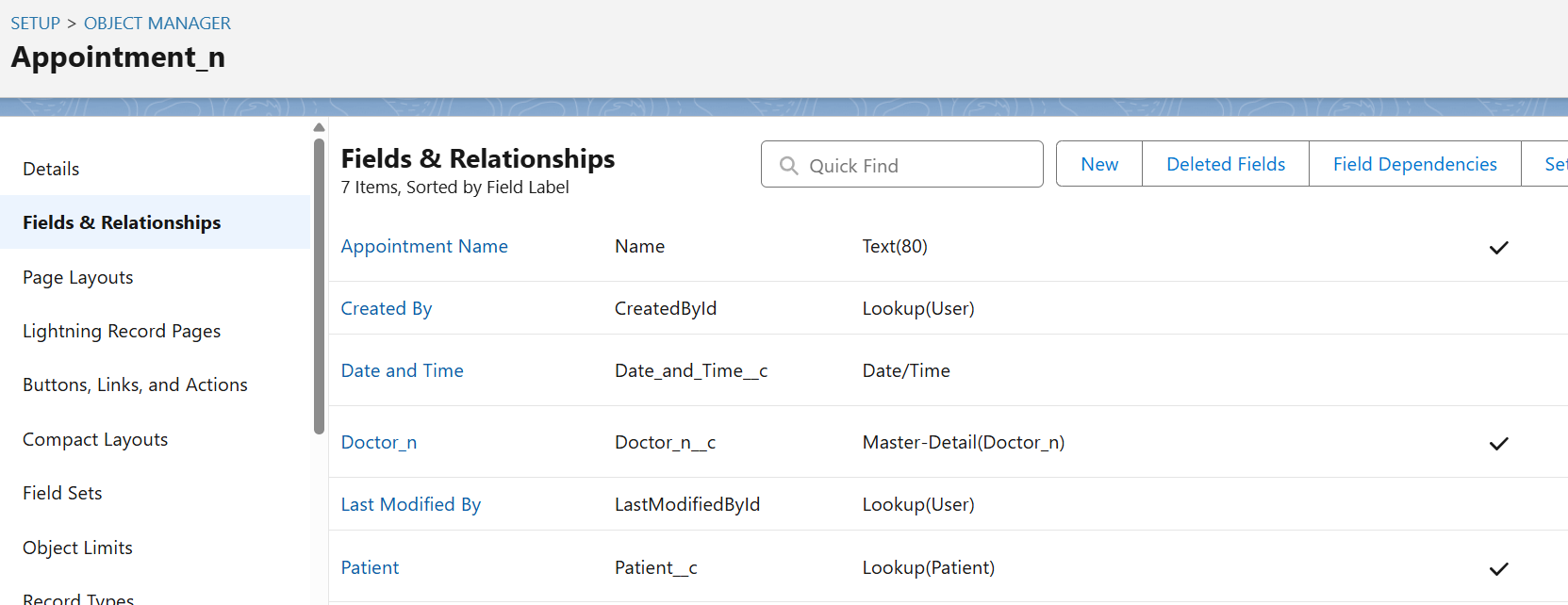
**7. Lookup vs Master-Detail vs Hierarchical Relationships**

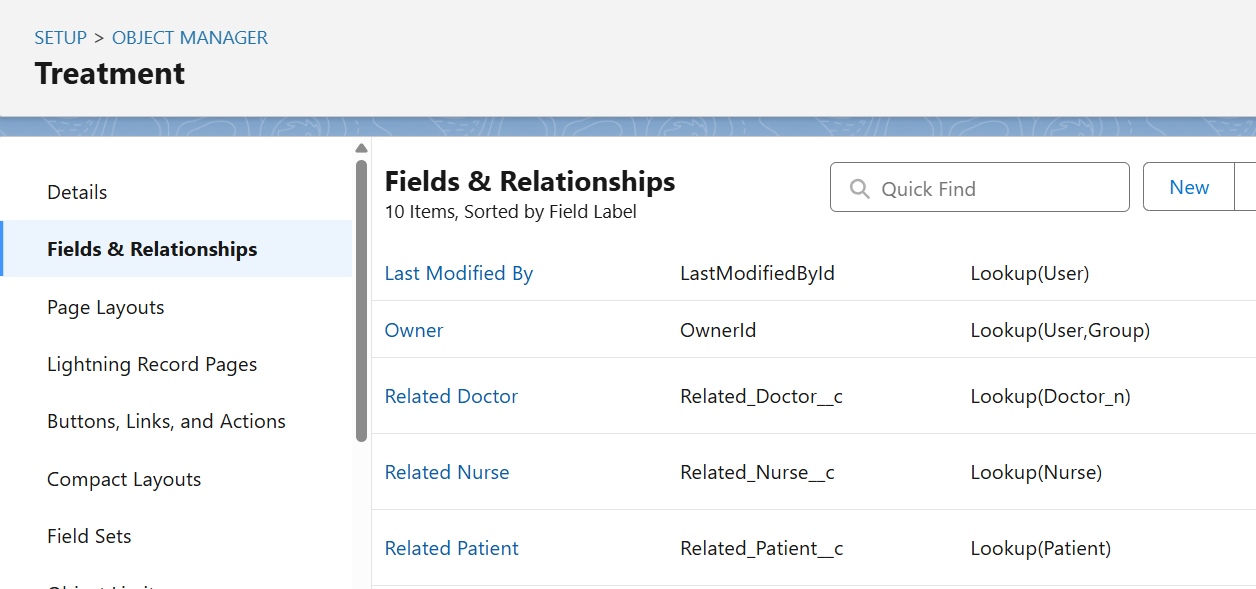
**🔹 Lookup Relationship**

* Loosely connects records (child can exist without parent).
* Example: Billing ↔ Patient (Lookup).
* Billing Object – Patient Link Explanation
* **Current Setup:** Billing has a Lookup to **Patient**.
* **Optional Alternative:** If a **Master-Detail Relationship (MDR) is created from Billing → Appointment, the Patient can be accessed through the Appointment.**

**Reason/Use Case:**

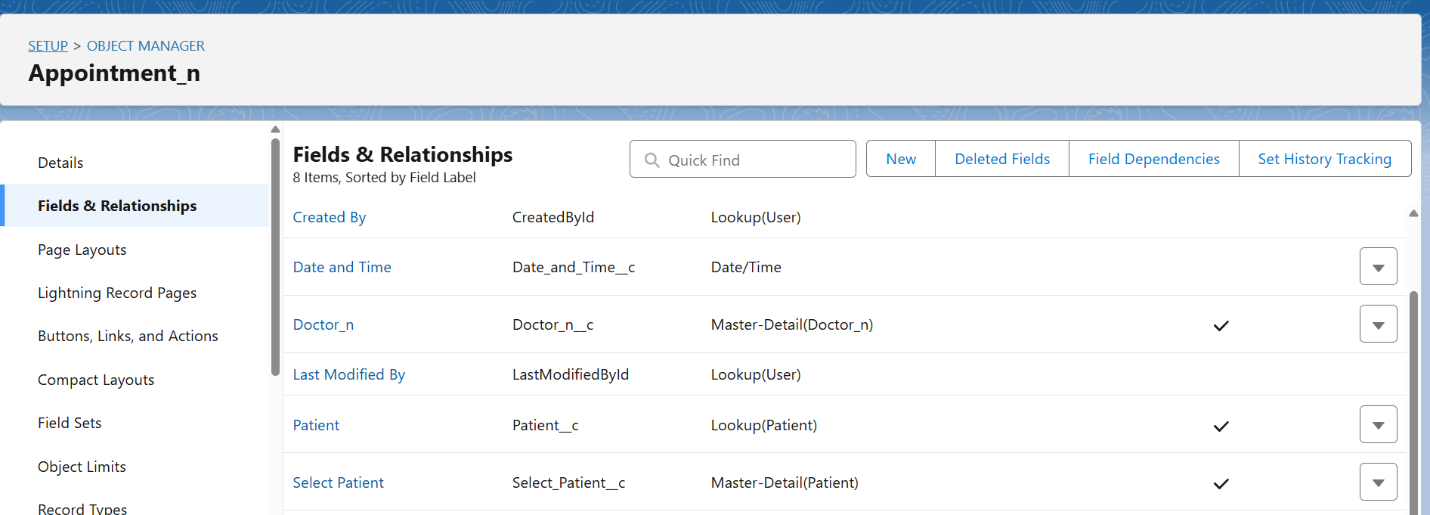
* Even without a direct Patient Lookup in Billing, you can still know which Patient the bill belongs to.
* Keeping the Patient Lookup is optional, but it can make **direct access and reporting easier**.
* This setup also allows flexibility if multiple Billing record types exist (e.g., Self Billing vs Insurance Billing).
* A bill can exist even if the patient record is archived, for financial records.





**🔹 Master-Detail Relationship (MDR)**

* Strong dependency, child record cannot exist without parent.
* Example: Appointment ↔ Patient (MDR), Appointment ↔ Doctor (MDR).
* *Reason*: Appointment makes no sense without a Patient or Doctor.



**🔹 Hierarchical Relationship**

* Only available for **User object**.
* Used for manager-subordinate setup.
* Example: Hospital staff reporting structure (Doctor reports to Head Doctor).

**8. Junction Object**:

**Junction Object – Appointment**

**Purpose**

The **Appointment** object acts as a **junction object** in the MediConnect CRM. It connects **Patients** and **Doctors**, enabling a **many-to-many relationship**:

* **Many Patients** can have appointments with **many Doctors**.
* **Many Doctors** can see **many Patients** through different appointment records.

This allows the hospital to **track all appointments accurately** and provides a foundation for billing, treatment tracking, and reporting.

**Use Cases**

1. **Patient View:** See all upcoming and past appointments.
2. **Doctor View:** View schedule and assigned patients.
3. **Reporting:** Generate reports such as:
   * “All appointments for a patient”
   * “All appointments handled by a doctor”
4. **Automation:** Workflow rules can trigger notifications for upcoming appointments or missed appointments.

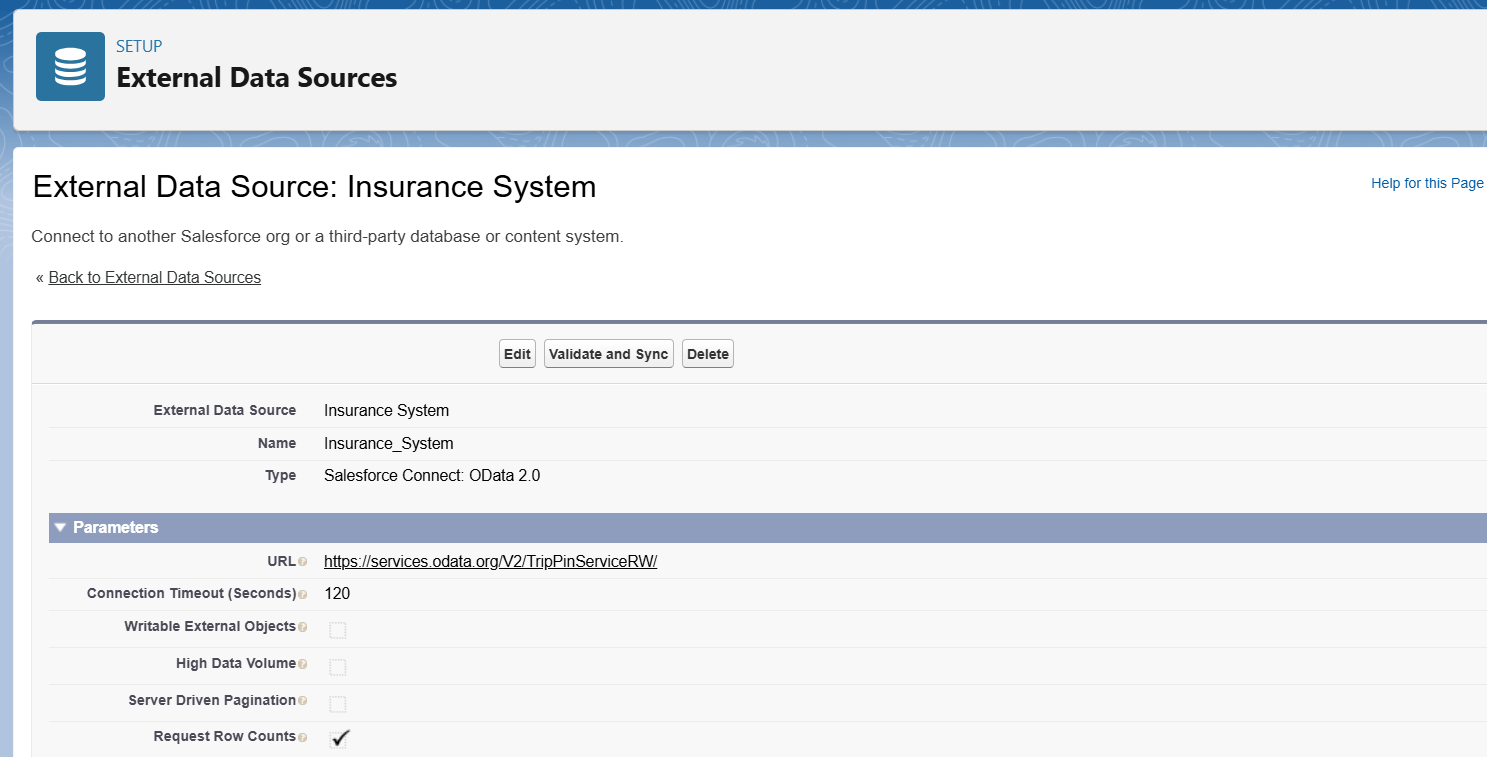
**9.External Objects in MediConnect**

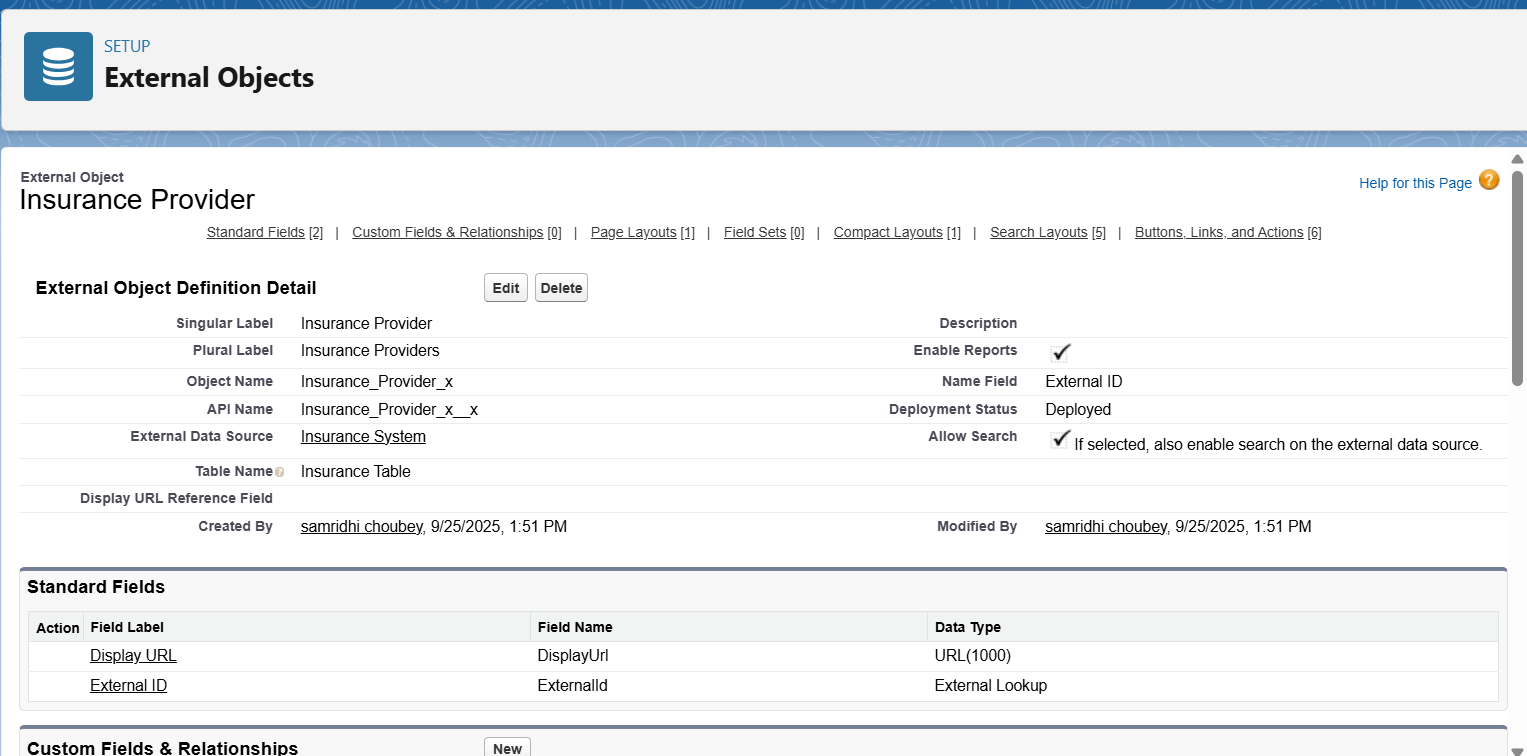
* **External Objects** are Salesforce objects that **map to data stored outside Salesforce**.
* They are used in **Salesforce Connect** to access external data **in real-time** without storing it in Salesforce.
* They look and behave like standard or custom objects inside Salesforce, but the data resides externally (e.g., in **another database, ERP, or API service**).

In MediConnect CRM, we might have external data like:

* **Insurance Provider Records** stored in an external insurance system

Instead of importing all this data, you can create **External Objects** in Salesforce to **view and interact with it directly**.

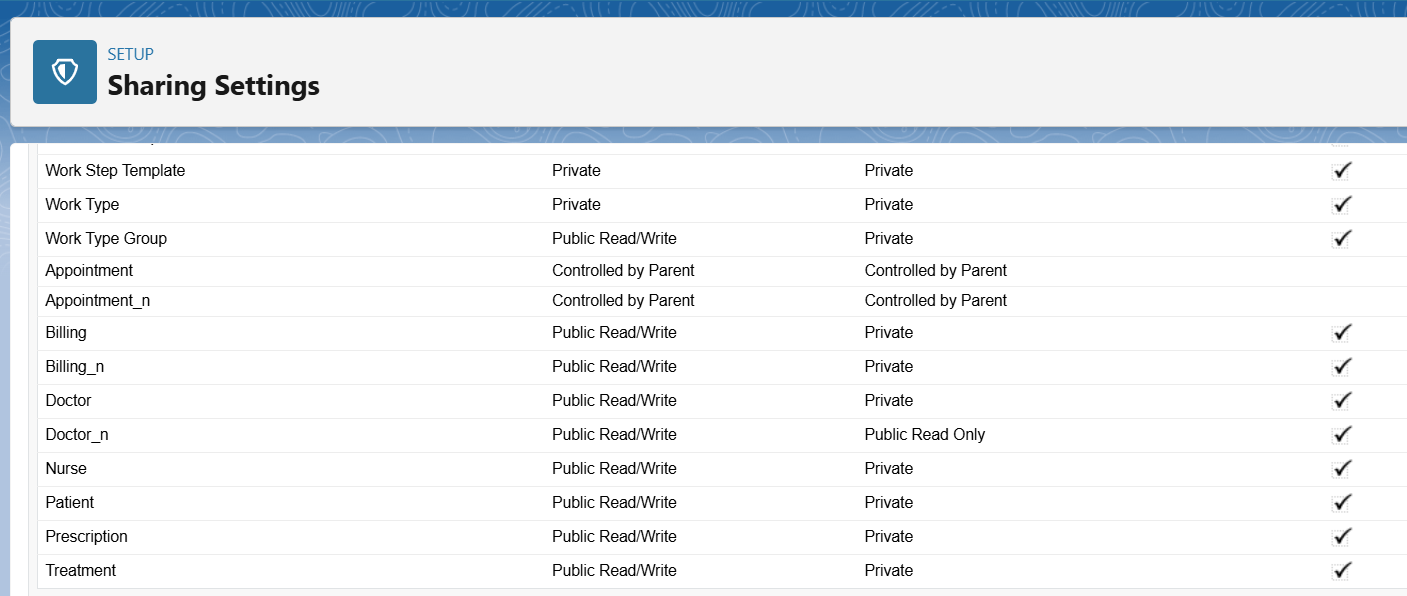




**NOTE: Remaining steps of phase 2 that can only be done after making custom objects.**

**Organization-wide Defaults (OWD):**

* OWD controls the default access to records for all users in the org.
* It defines the baseline level of access (Read / Read-Write / Private) for objects.

****

**Sharing Rule:**

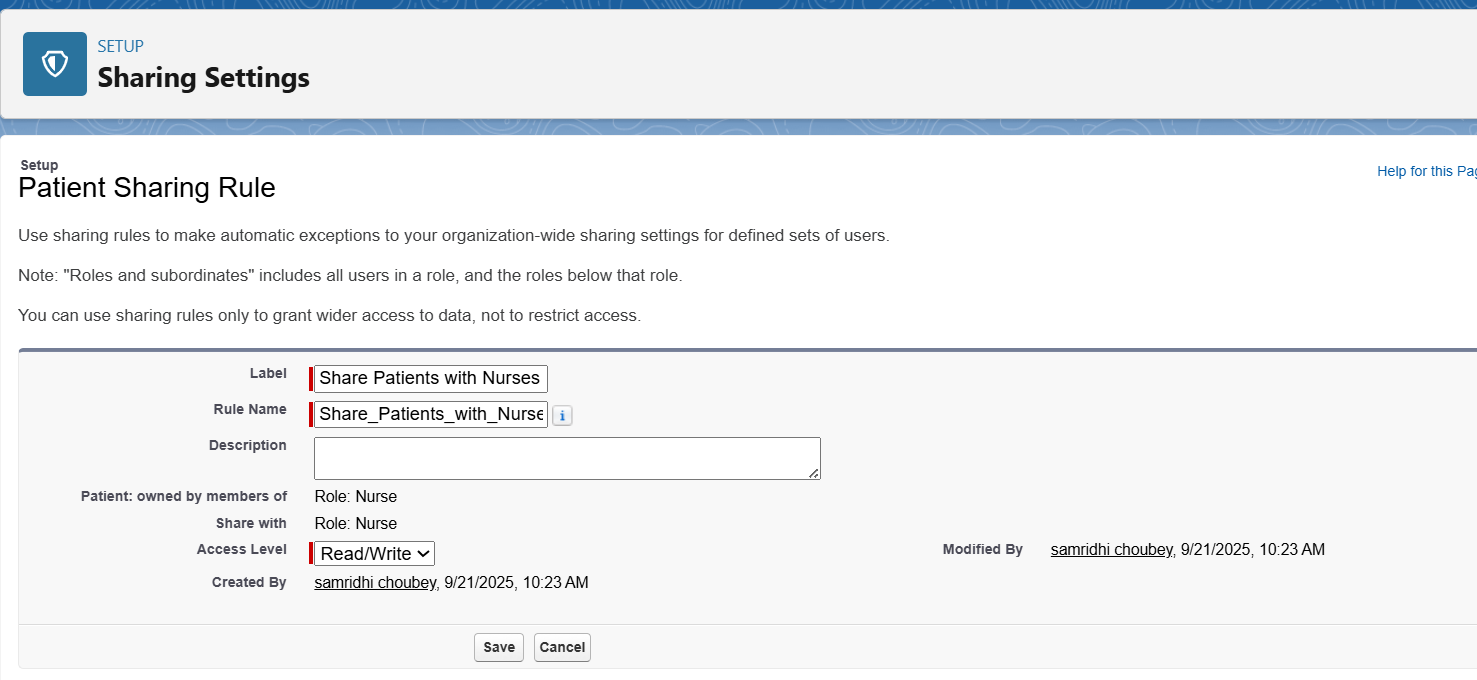
* Sharing Rules are used to grant additional access beyond OWD.
* They are record-level rules based on criteria or ownership.
* Allows: Read-Only or Read/Write access to groups of users.

**Sharing Rule – Patient with Nurse**

**Ex 1:**

* **OWD (Patient):** Private → by default, only the record owner and admins can see patient records.
* **Requirement:** Nurses need access to the **patients assigned to their doctors**.

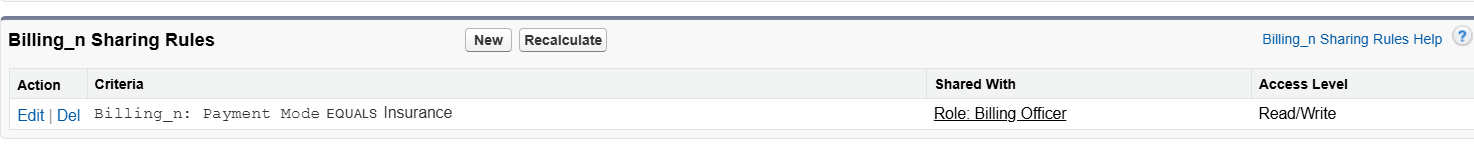


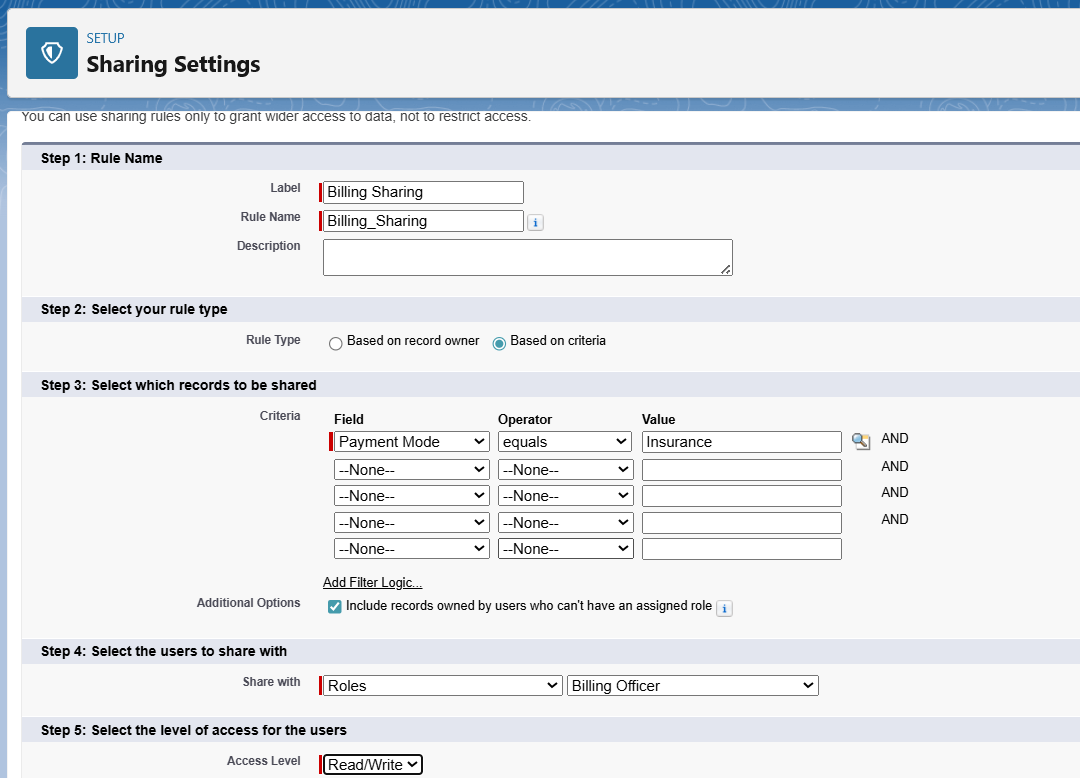


**Ex 2:**

**Billing Sharing Rule**

* OWD for Billing = Private
* Billing Officer needs access to billing records for processing if *Payment mode == insurance*





**Use Case**: Protects sensitive billing information; only authorized staff can access and update it.