Medicare Cloud — Phase 3: Data Modeling & Relationships

Goal: Build a scalable, secure, and well-documented data model for Medicare Cloud that supports patients, appointments, insurance claims, care plans, and integrations while ensuring proper relationships, permissions, and reporting.

---

1. Overview & Naming Conventions

- Use clear and consistent naming for objects, fields, and relationships.
- **Objects:** Patient__c, Appointment__c, Insurance_Claim__c, Care_Plan__c.
- **Fields:** CamelCase__c (e.g., Date_of_Birth__c, Primary_Doctor__c).
- **Record Names:** Choose meaningful options like Auto Number (Patient Number) or Text (Full Name).
- Maintain a **metadata spreadsheet** documenting all objects, fields, data types, default values, help text, and formulas.

---

2. Standard vs Custom Objects

- Identify entities that are standard (like Contact, Account) and custom (Patient__c, Appointment__c).
- **Steps to create:**
    1. Setup → Object Manager → Create → Custom Object.
    2. Fill Label, Plural Label, Record Name Type.
    3. Enable reports, activities, and track field history as needed.
- Document each object's purpose and relationship to other objects.

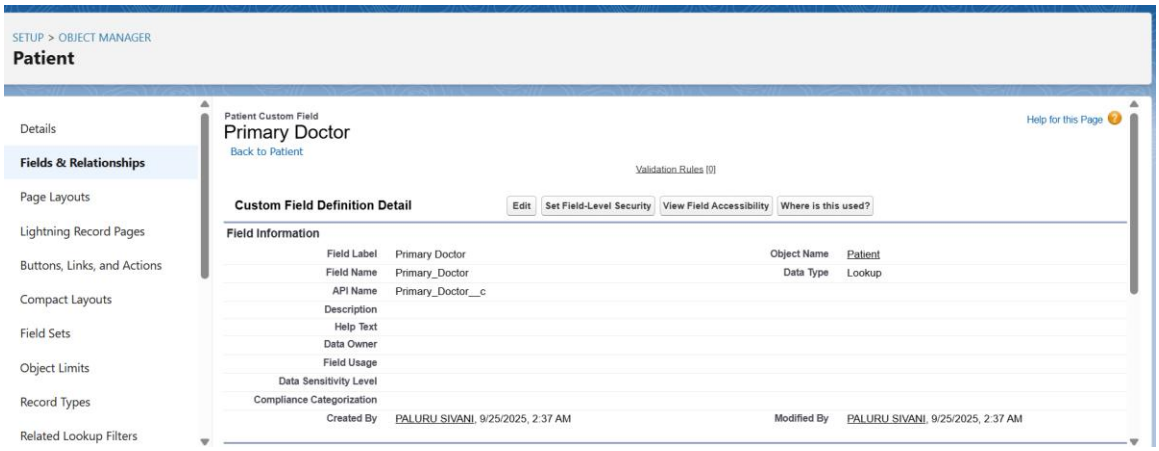
---

3. Fields & Field Types

- **Field Types:** Text, Number, Date, Date/Time, Currency, Formula, Picklist, Checkbox, Phone, Lookup, Master-Detail.
- **Examples:**

- Patient__c: Full_Name__c, Date_of_Birth__c, Gender__c, Phone__c, Primary_Doctor__c.
- Appointment__c: Start_Date__c, End_Date__c, Status__c, Duration_Minutes__c.
- Insurance_Claim__c: Claim_Amount__c, Claim_Status__c, Submission_Date__c.

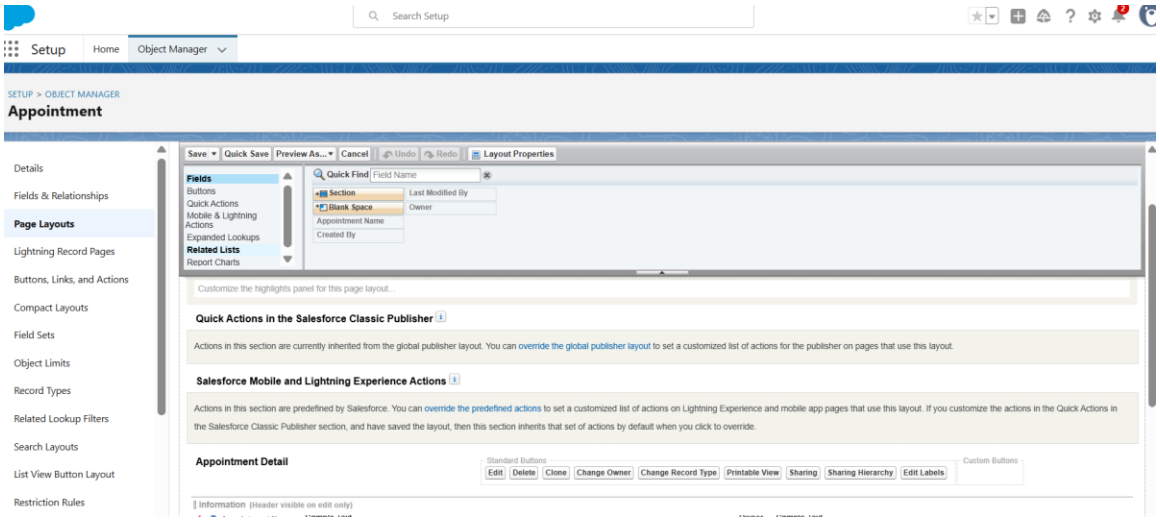
- Use **Formula Fields** for calculations (e.g., Appointment Duration = End_Date__c - Start_Date__c).
- Use **Picklists** for fixed options (e.g., Gender, Appointment Status, Claim Status).



---

4. Record Types & Page Layouts

- Record Types allow multiple business processes per object.
- Page Layouts control how fields and related lists appear to users.
- **Example:** Appointment__c can have “Consultation” vs “Follow-Up” record types with different fields.
- Ensure proper field-level security for sensitive data (PHI).



---

5. Compact Layouts

- Compact Layouts define which fields appear in mobile cards or highlights.
- Include key fields like Name, Status, and Appointment Date for mobile users.

---

6. Schema Builder

- Visualize objects, fields, and relationships in a drag-and-drop interface.
- Helps identify missing relationships or redundant objects.
- Supports creation of new fields and relationships directly in the visual model.

---

7. Lookup vs Master-Detail vs Hierarchical

- **Lookup:** Loose coupling; child can exist without parent. Use for independent records.
- **Master-Detail:** Strong coupling; child deletion cascades, sharing rolls up to parent. Use when child logically depends on parent.
- **Hierarchical:** Only available for User object; use for reporting lines.
- **Step:** Object Manager → Child Object → Fields & Relationships → New → Choose Relationship Type → Configure.

---

8. Junction Objects (Many-to-Many)

- Use when one object relates to multiple records of another object.
- Example: One Booking can include multiple Cars.
- Steps:
    1. Create Junction Object (Booking_Car__c).
    2. Add Master-Detail to Rental_Booking__c and Car__c.
    3. Add related lists on parent objects.
    4. Include formula or roll-up fields (e.g., Total Booking Amount).

---

9. External Objects

- Connect to external data sources (insurance databases, external EHR systems).
- Allows read/write access to external data without duplicating it in Salesforce.
- Steps: Setup → External Data Sources → New → Configure → Sync.

---

10. Data Import & Sample Data

- Import parent records first (Patients, Cars), then child records (Appointments, Claims).
- Use **Data Import Wizard** for small datasets or **Data Loader** for large datasets.
- Validate imported data via related lists and reports.

---

11. Testing & Validation

- Create sample records for each object.
- Verify related lists, roll-up summaries, formulas, picklists, and field validations.
- Test sharing rules, OWD, profiles, and permission sets.
- Validate mobile and compact layouts.

---

12. Performance & Best Practices

- Use **External IDs** and **unique fields** for efficient lookups.
- Avoid unnecessary Master-Detail relationships to prevent complex cascading deletes.
- Keep field-level security conservative to protect PHI.
- Document all changes in a metadata spreadsheet for governance.
- Maintain version control using **SFDX + Git** for deployments.

---

13. Deliverables

- Object & field inventory spreadsheet.
- Schema Builder or ERD diagram.
- Sample data CSVs.
- Test cases and UAT checklist.