# Report: Completion of Phase 3 - Data Modeling & Relationships

This document provides a detailed, step-by-step summary of all data modeling tasks completed for Phase 3 of the Salon Management System project. The objective of this phase was to build the foundational data structure, or "skeleton," of the application.

1. Standard & Custom Objects

- **Purpose:** To establish the core "filing cabinets" for our data. We used Standard Objects for common business entities (people and employees) and created Custom Objects for salon-specific information (services and bookings).
- **Steps Followed:**
  1. **Standard Objects Utilized:** The standard `Contact` object was designated to store client information, and the standard `User` object was used for salon staff (Stylists).
  2. **Custom Object `Service__c` Created:**
     - Navigated to `Setup > Object Manager > Create > Custom Object`.
     - Set the **Label** to `Service` and **Plural Label** to `Services`.
     - Changed the standard "Name" field to an **Auto-Number** with the format `SERV-{0000}`.
     - Enabled the option to **"Launch New Custom Tab Wizard"** and selected the "Scissors" icon for the tab.
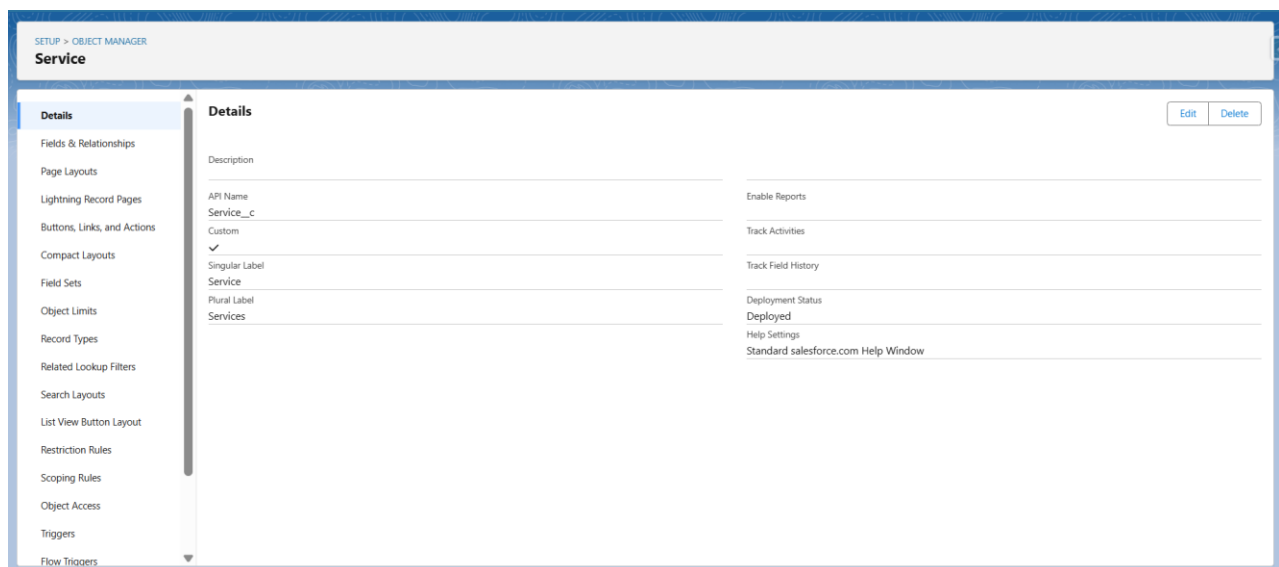

  3. **Custom Object `Appointment__c` Created:**
     - Navigated to `Setup` > Object Manager > Create > Custom `Object`.
     - Set the **Label** to `Appointment` and **Plural Label** to `Appointments`.
     - Set the standard "Name" field to an **Auto-Number** with the format `APP-{0000}`.
     - Enabled the option to **"Launch New Custom Tab Wizard"** and selected the "Calendar" icon for the tab.
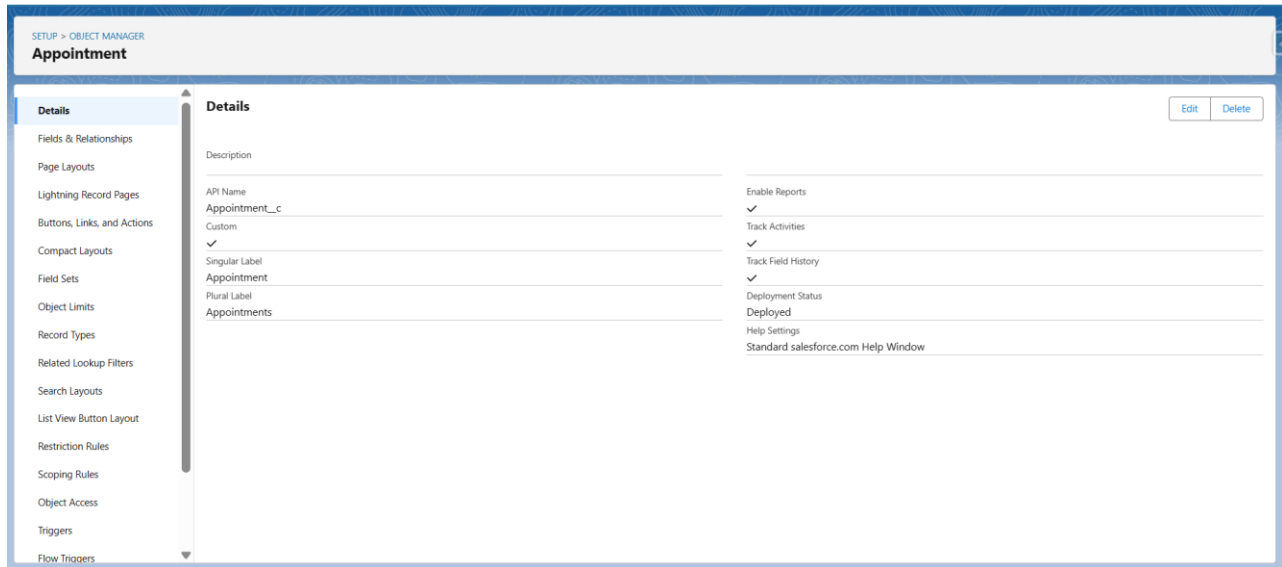
2. Fields & Relationships

- **Purpose:** To define the specific pieces of information (the "fields" in the filing cabinet) for each object and to create the crucial links (relationships) between them.
- **Steps Followed:**
    1. **On the `Service__c` Object:**
        - Created a new `Picklist` field named **`Service Name`** with values like "Men's Haircut," "Full Color," etc. This was made **Required**.
        - Created a new `Currency` field named **`Price`**.
        - Created a new `Number` field named **`Duration (Minutes)`**.
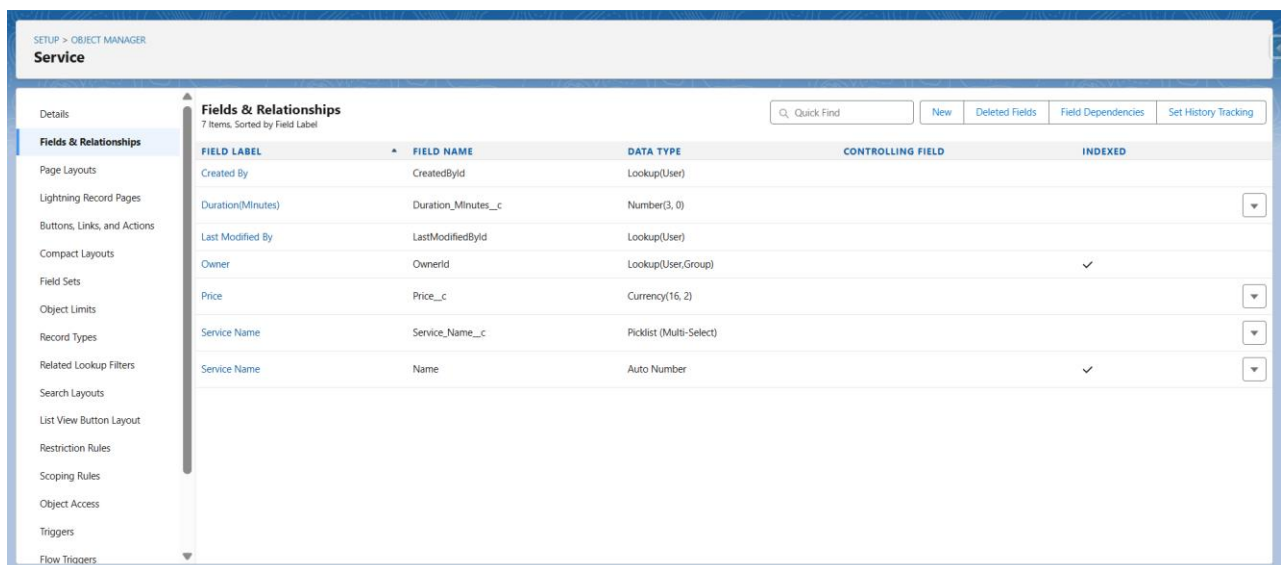
    2. **On the `Appointment__c` Object:**
        - Created a **`Lookup Relationship`** to the `Contact` object, with the Field Label **`Client`**. This field was made **Required**.

- Created a **Lookup Relationship** to the `Service` object, with the Field Label **`Service`**. This field was made **Required**.
- Created a **Lookup Relationship** to the `User` object, with the Field Label **`Stylist`**. This field was made **Required**.
- Created a new `Date/Time` field named **`Appointment Date & Time`**.
- Created a new `Picklist` field named **`Status`** with the values: `Scheduled`, `Completed`, `Cancelled`.

3. Record Types

- **Purpose:** Record Types are used to create different business processes and page layouts for the same object.
- **Steps Followed:** No Record Types were created in this phase. The current business process is standardized for all appointments, so a single page layout is sufficient. This feature could be used in the future to create different types of appointments (e.g., "In-Salon" vs. "Home Visit").

4. Page Layouts

- **Purpose:** To organize the fields, buttons, and related lists on a record's detail page for a clean and efficient user experience.
- **Steps Followed:**
  1. Navigated to `Setup` > Object Manager > Appointment > `Page Layouts`.
  2. Clicked **Edit** next to the "Appointment Layout."
  3. In the layout editor, dragged and dropped the key fields (`Client`, `Stylist`, `Service`, `Appointment Date & Time`, and `Status`) to the top of the "Information" section for high visibility.
  4. Removed unnecessary fields (like the default `Owner` field) from the layout by dragging them back to the top panel.
 5. Clicked **Save**.
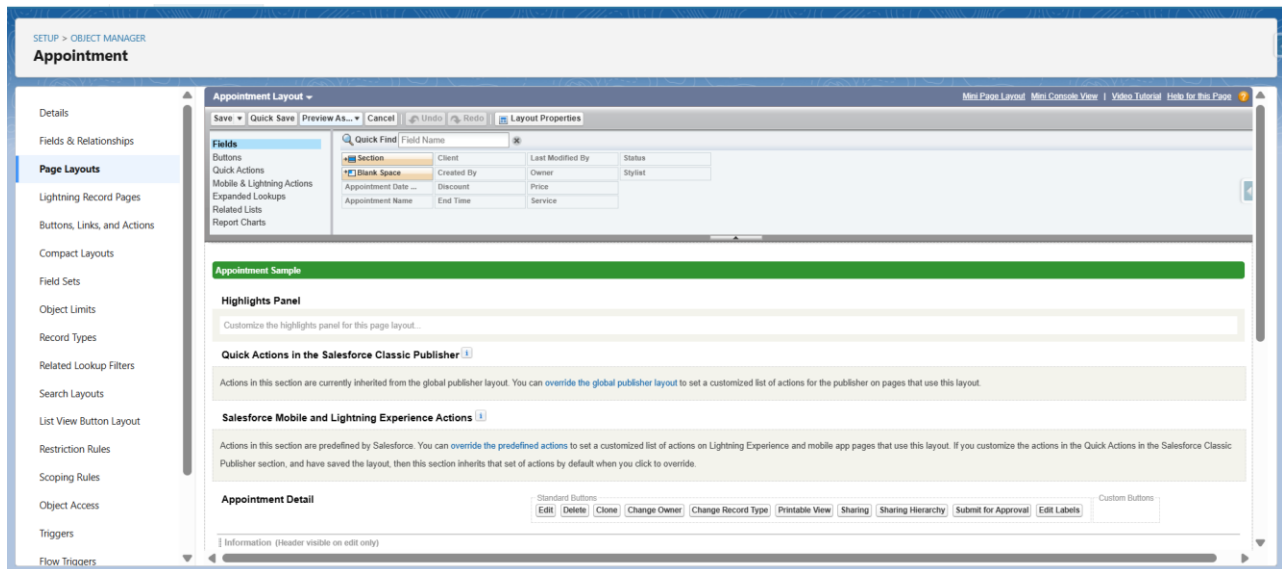
5. Compact Layouts

- **Purpose:** To control which key fields appear in the "at-a-glance" highlights panel at the top of a record page and in the Salesforce mobile app.
- **Steps Followed:**
    1. Navigated to `Setup > Object Manager > Appointment > Compact Layouts`.
    2. Clicked **New** and created a layout named `Appointment Highlights`.
    3. Added the `Client`, `Stylist`, `Service`, and `Appointment Date & Time` fields to the "Selected Fields" list.
    4. Clicked **Save**.
    5. Clicked the **Compact Layout Assignment** button and assigned the new `Appointment Highlights` layout as the Primary Compact Layout for the object.
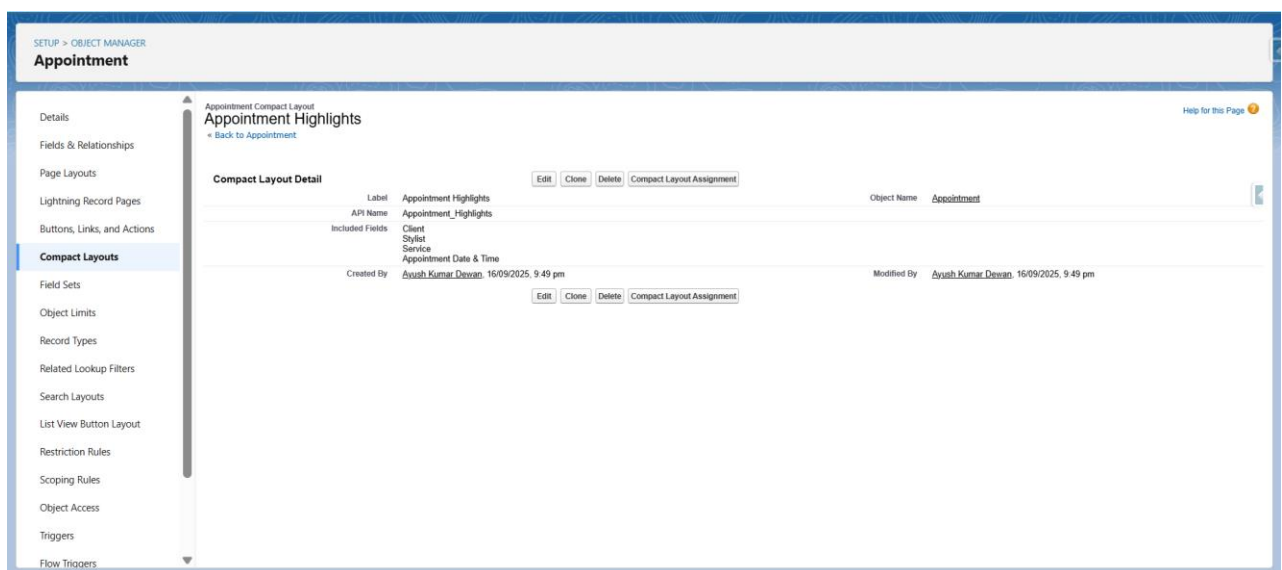
6. Schema Builder

- **Purpose:** To visually review the entire data model and confirm that all objects and relationships have been built correctly.
- **Steps Followed:**
    1. Navigated to `Setup > Schema Builder`.
    2. Cleared the canvas of all default objects.
    3. Selected the four key objects for this project: `Contact`, `User`, `Service`, and `Appointment`.
    4. Used the **Auto-Layout** feature to generate a clean, visual diagram of the data model, confirming that the `Appointment` object correctly links the other three.
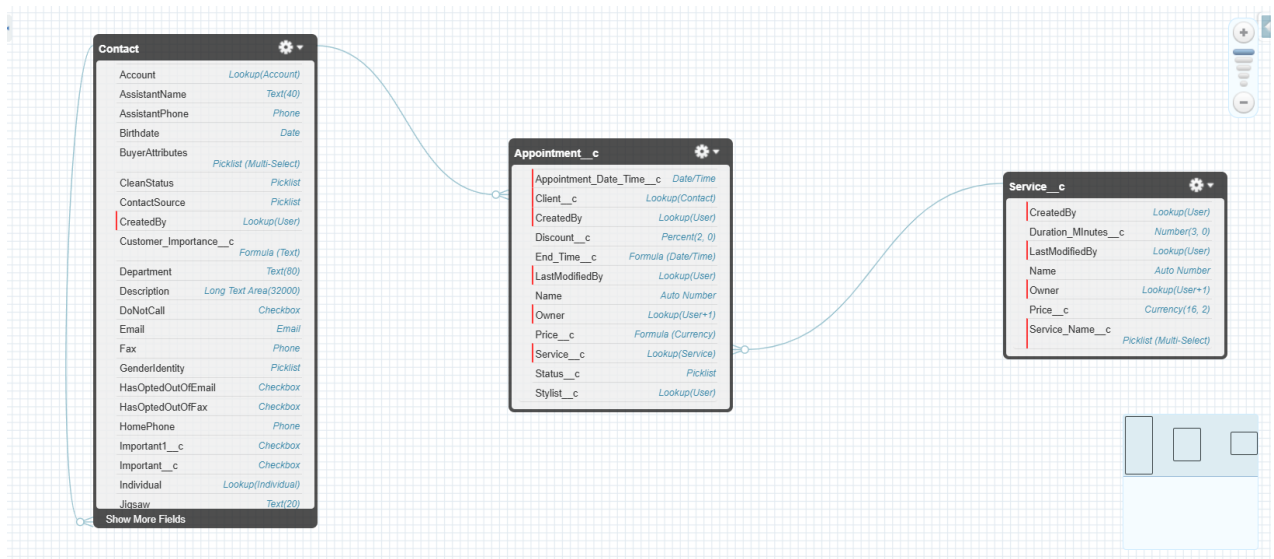
7. Lookup vs Master-Detail vs Hierarchical Relationships

- **Purpose:** To choose the correct type of relationship for our business needs, ensuring the right behavior for data deletion and security.
- **Steps Followed:**
    - **Lookup Relationships** were intentionally chosen to connect `Appointments` to `Contacts`, `Services`, and `Users`. This ensures that if a client or service is ever deleted, the old appointment records are kept for historical reporting, which is a key business requirement.
    - A **Hierarchical Relationship** was not directly created in this phase, but the concept is used on the `User` object (via the `Manager` field), which is driven by the Role Hierarchy built in Phase 2.

8. Junction Objects

- **Purpose:** Junction Objects are used to create a many-to-many relationship between two objects.
- **Steps Followed:** No Junction Objects were created. The current business model requires one primary service per appointment, which is a one-to-many relationship (one service can be on many appointments). This feature could be used in the future to allow for multiple "add-on" services for a single appointment.

9. External Objects

- **Purpose:** External Objects are used to view and interact with data from an external system without importing it into Salesforce.
- **Steps Followed:** No External Objects were created,