## PropertyCare CRM Project

**Phase 1: Problem Understanding & Industry Analysis**

1. **Requirement Gathering**

* Engage with key stakeholders such as the Real Estate CEO, Sales Agents, Property Managers, and Prospective Buyers to identify core needs.
* The system requirements include:
* Automate lead capture from website/social media.
* Track property inventory with price, location, and status.
* Allow scheduling and tracking of property visits.
* Generate sales funnel reports and revenue dashboards.

**2. Stakeholder Analysis**

* Admin – Setup & configuration.
* Sales Agents – Manage leads, visits, and deals.
* Property Manager – Manage property listings.
* Manager/CEO – Monitor dashboards, approvals.
* Customers – Receive visit confirmations.

**3. Business Process Mapping**

* Flow Example:

Customer Inquiry → Lead Captured → Lead Qualification → Assigned to Agent → Visit Scheduled → Property Manager Updates Availability → Manager Reviews → Deal Closed → Dashboard Updated

**4. Industry-specific Use Case Analysis**

* Real Estate industry requires handling large volumes of inquiries, high-value deals, and accurate property availability.
* Lead qualification and scoring.
* Visit scheduling with notifications.
* Tracking buyer preferences and purchase history.
* Dashboards for sales performance and revenue trends.

**5. AppExchange Exploration:** Search for “Real Estate CRM” apps. Some apps exist, but to learn and build end-to-end, we’ll develop a tailored solution with PropertyCare CRM.

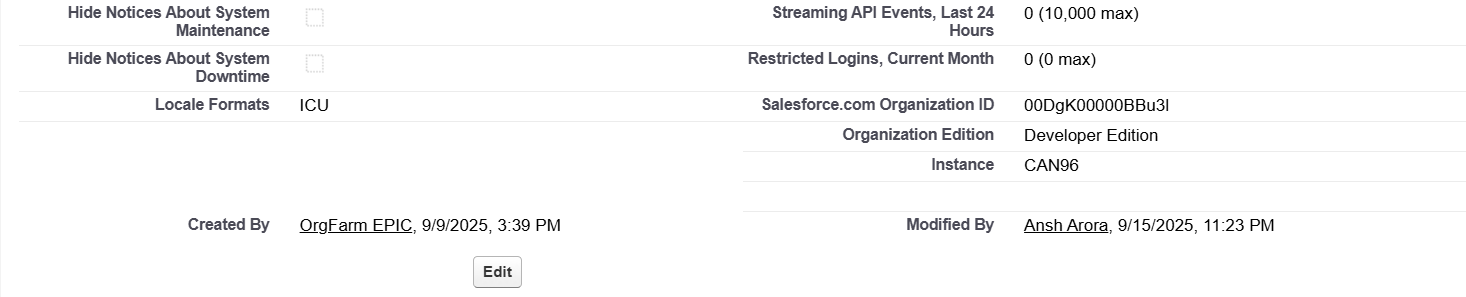
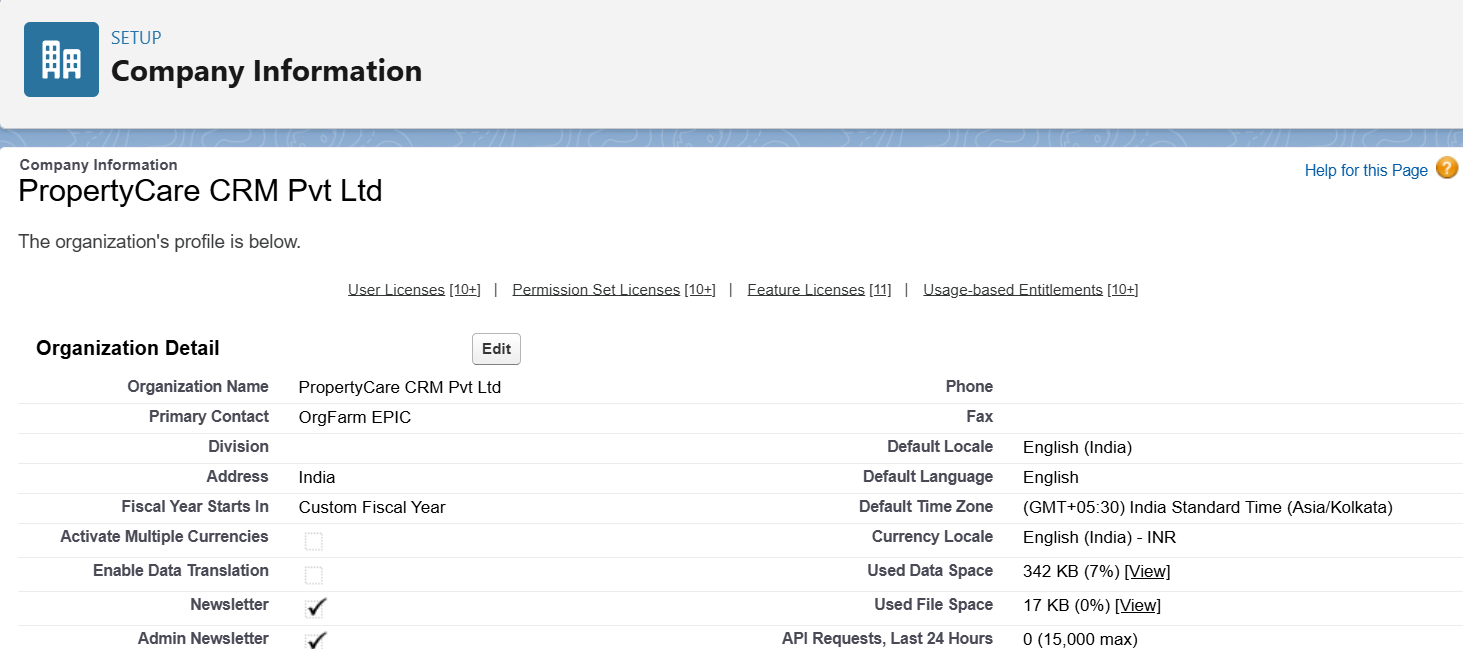
**Phase 2: Org Setup & Configuration**

**1. Salesforce Editions**

* Use **Salesforce Developer Edition** for this project.
* This edition provides full CRM features with free licenses for testing.

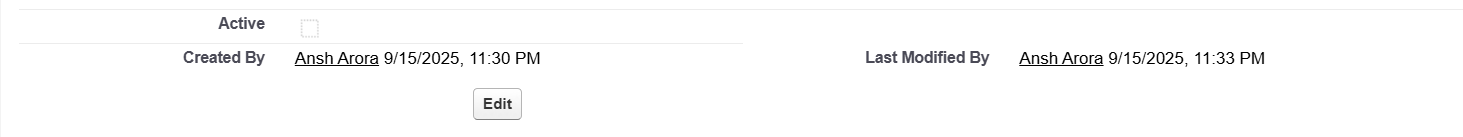
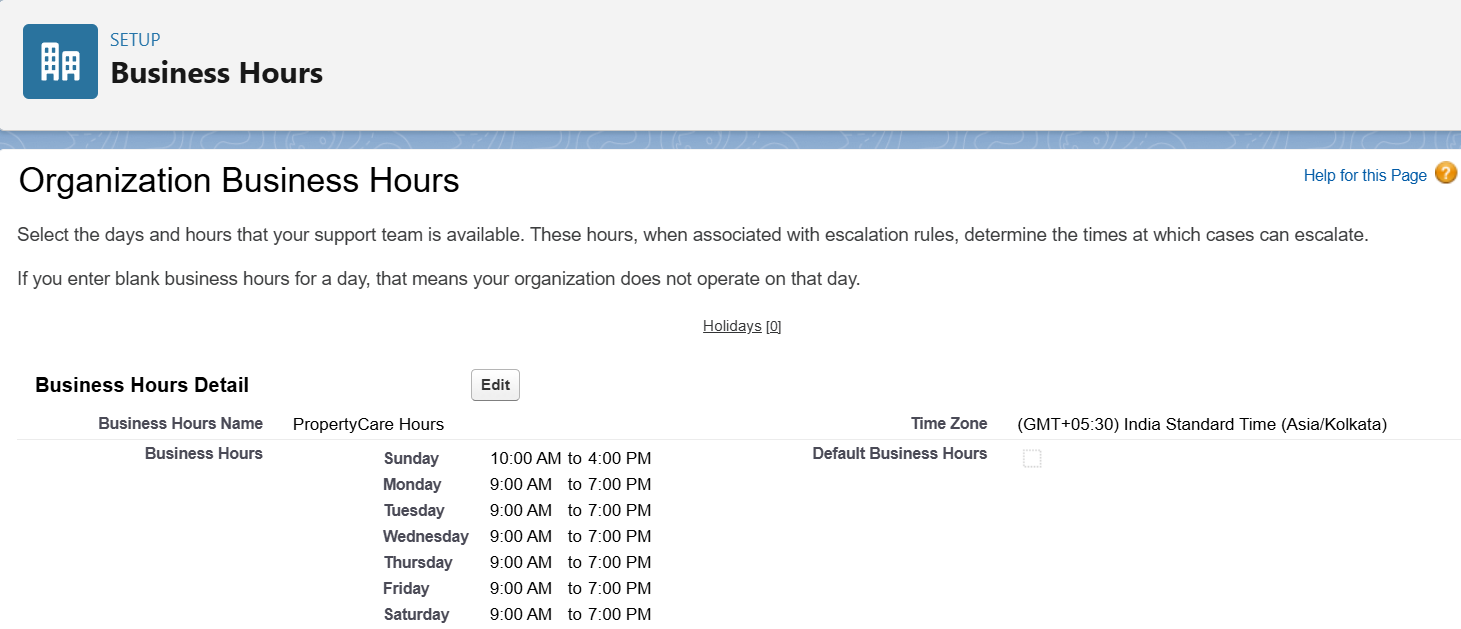
**2. Company Profile Setup**

* Navigate: **Setup → Company Information**.
* Configure:
  + **Organization Name** = PropertyCare CRM Pvt. Ltd.
  + **Default Time Zone** = (GMT+05:30) India Standard Time.
  + **Currency** = INR.



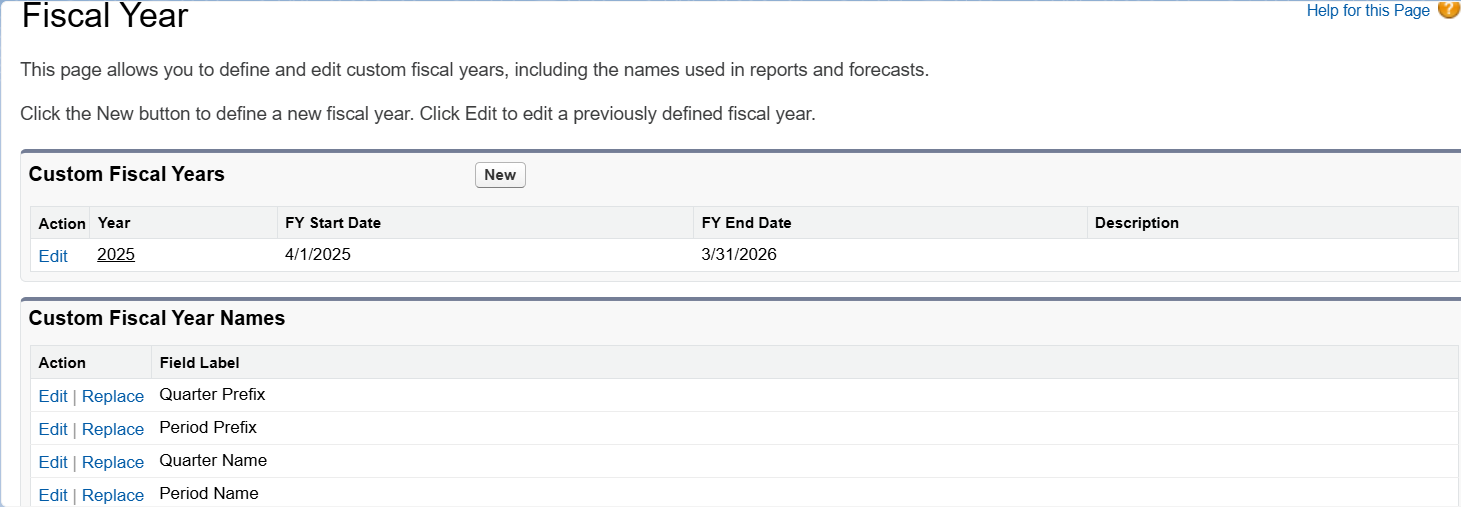
**3. Business Hours Setup**

* Setup → **Business Hours** → Add:
  + Default Hours: Mon–Sun (9:00 AM – 7:00 PM).
* **Name**: PropertyCare Hours
* **Time Zone:** (GMT+05:30) India Standard Time (Asia/Kolkata)



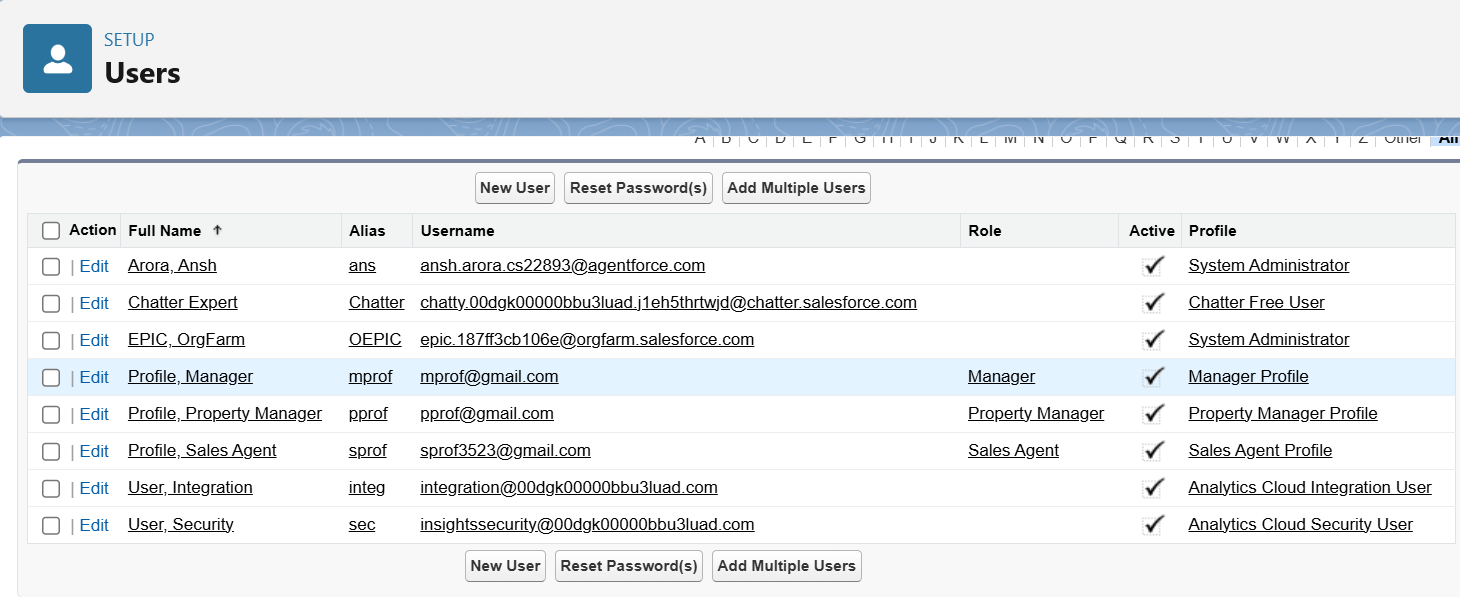
**4. Fiscal Year Settings**

* Setup → **Fiscal Year**.
* Enable **Custom Fiscal Year** (April–March cycle).
* Quarters:
  + Q1: Apr–Jun
  + Q2: Jul–Sep
  + Q3: Oct–Dec
  + Q4: Jan–Mar



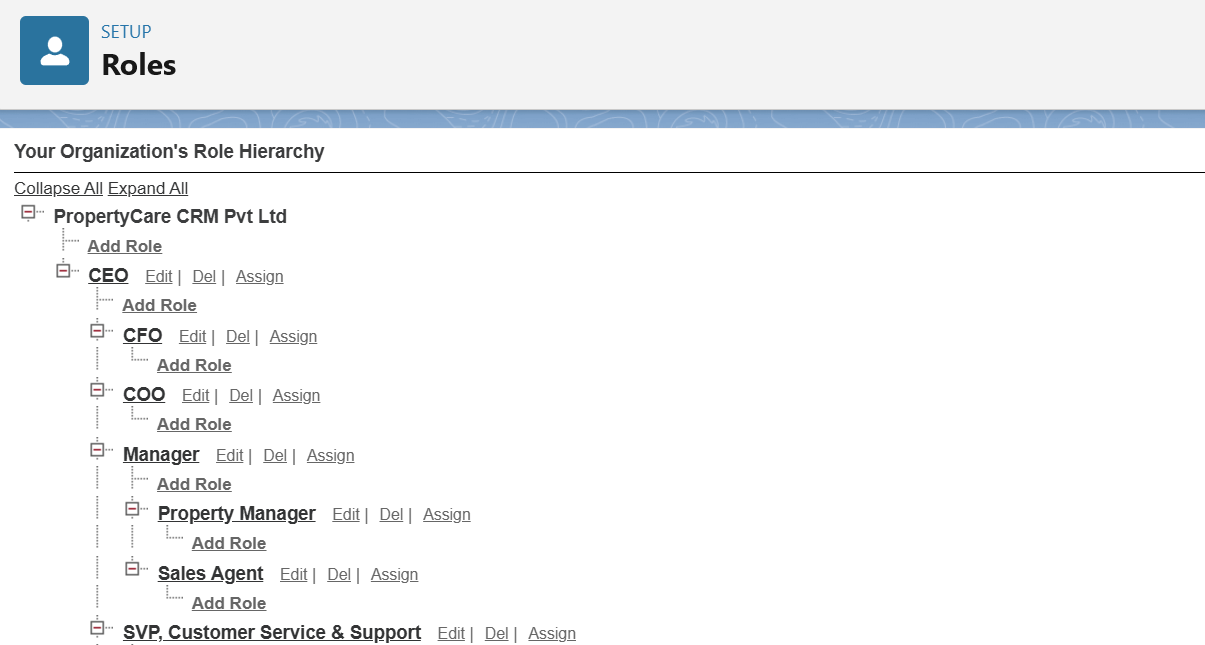
**5. User Setup & Licenses**

* Create 3 Users:
  + **Manager** → Salesforce License → Manager Profile
  + **Property Manager** → Salesforce Platform License → Property Manager Profile
  + **Sales Agent** → Salesforce License → Sales Agent Profile



**6. Roles**

* Setup → Roles → Set Up Roles.
* Create hierarchy:
  + CEO (top)
    - Manager
      * Property Manager
      * Sales Agent



**7. Profiles**

 Clone **Standard User** profile to create: Sales Agent, Property Manager, Manager Profiles.

 Assign object permissions based on role:

* Sales Agent → Leads, Opportunities, Properties
* Property Manager → Properties, Events
* Manager → Full access to all CRM objects

 Configure field-level security and app access as needed.

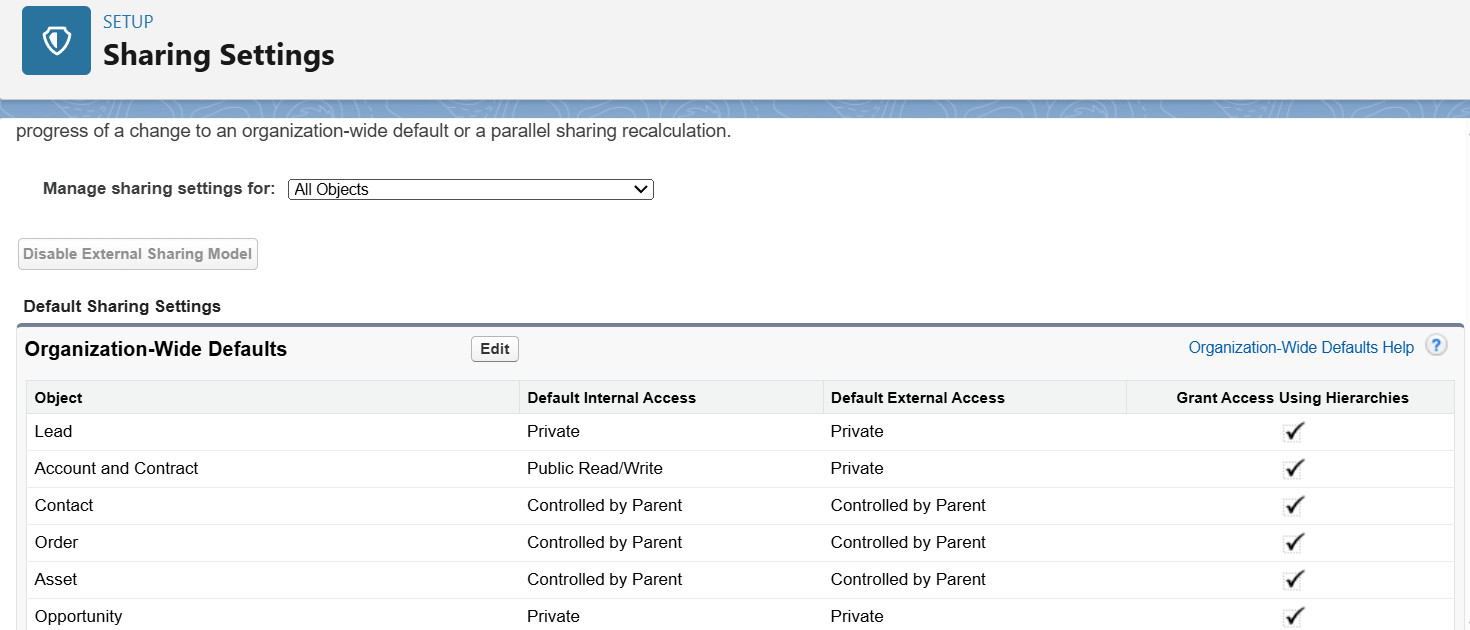
 Save and verify permissions.

**8. Permission Sets**

* Setup → **Permission Sets → New**.
* Name: *Visit Scheduling Access*.
* Add permissions for **Events** (Read & Edit) and **Calendar.**
* Assign the Permission Set to Sales Agents.

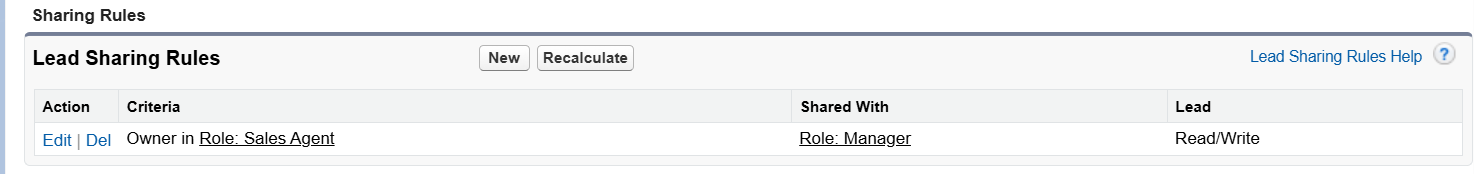
**9. Org-Wide Defaults (OWD)**

* Navigate: **Setup → Sharing Settings → Organization-Wide Defaults**
* Configure:
  + **Leads** → Private (Internal & External)
  + **Properties (Custom Object)** → Public Read Only (Internal & External)
  + Opportunities = Private (Internal & External)



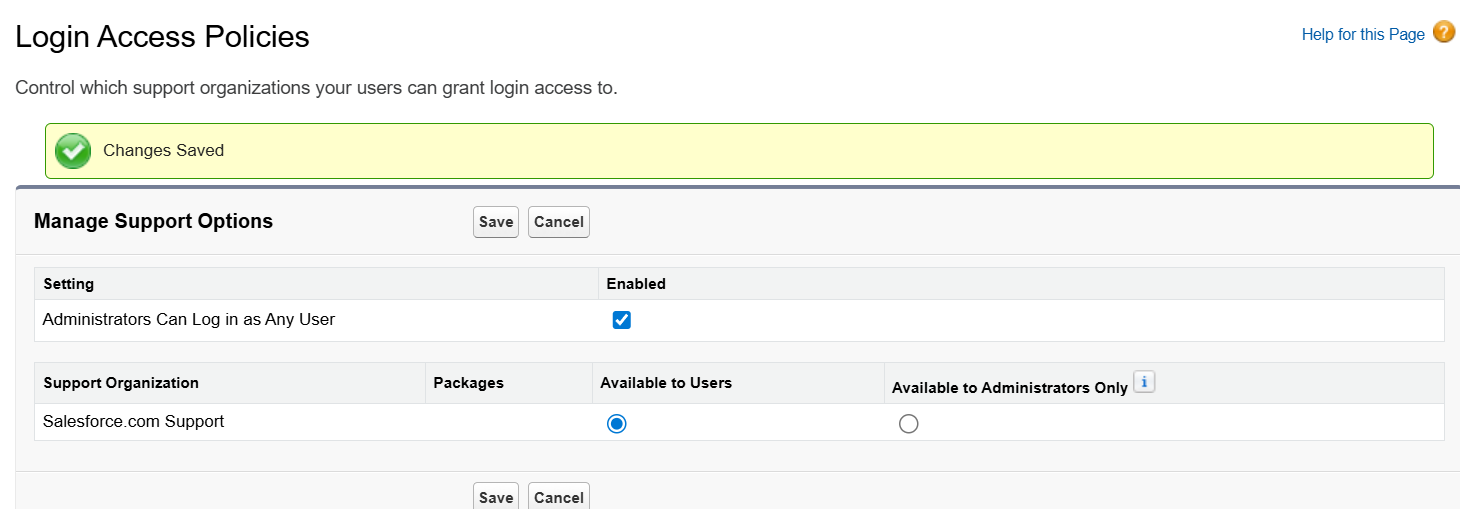
**10. Sharing Rules**

* Navigate: **Setup → Sharing Settings → Sharing Rules → New**
* Created rule for **Lead** object.
* Criteria: **Owner in Role = Sales Agent**.
* Shared With: **Role = Manager**.
* Access Level: **Read/Write**.
* This allows Managers to view and update Leads owned by Sales Agents.



**11. Login Access Policies**

* Setup → **Login Access Policies**.
* Enable: *Admins can log in as any user*.



**12. Dev Org & Sandbox**

* Developer Org will act as sandbox.
* For advanced deployment → enable sandbox (Enterprise Edition only).

**13. Deployment Basics**

* Setup → **Outbound Change Sets** (for metadata migration).
* Installed **VS Code + Salesforce Extensions** → connected with SFDX.

**Phase 3: Data Modeling & Relationships**

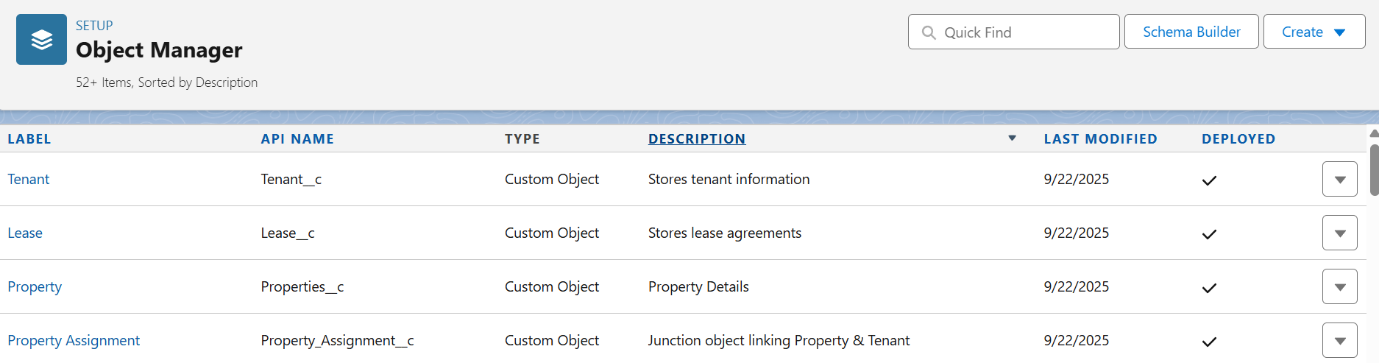
1. **Create Custom Objects**

**Objects Created:**

1. **Property** – Stores property details. Record Name: Property Name (Text)
2. **Tenant** – Stores tenant information. Record Name: Tenant Name (Text)
3. **Lease** – Stores lease agreements. Record Name: Lease Number (Auto Number: L - {0000}
4. **Property Assignment** – Junction object linking Property & Tenant. Record Name: Assignment Number (Auto Number: PA- {0000})

**Procedure:**

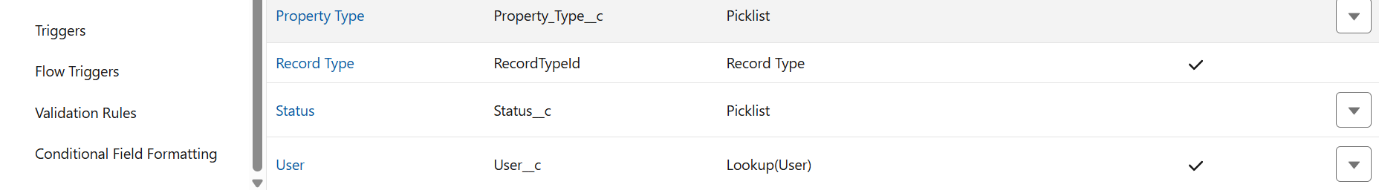
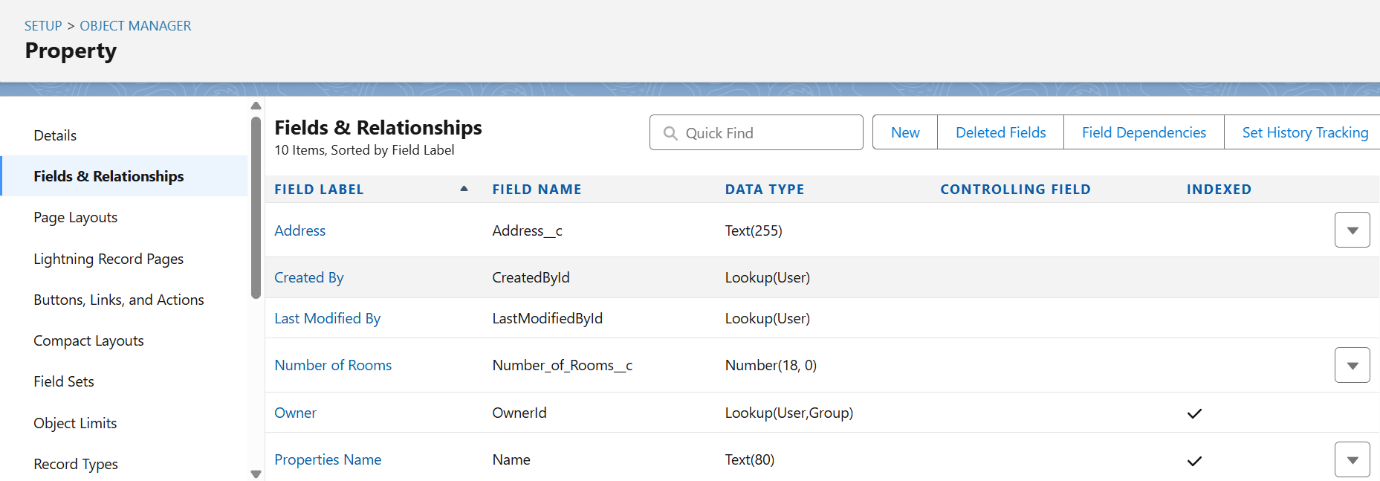
* Setup → Object Manager → Create → Custom Object
* Fill Label, Plural Label, Record Name, Data Type
* Enable **Allow Reports** and **Allow Activities**
* Set Deployment Status = Deployed
* Save and repeat for all objects



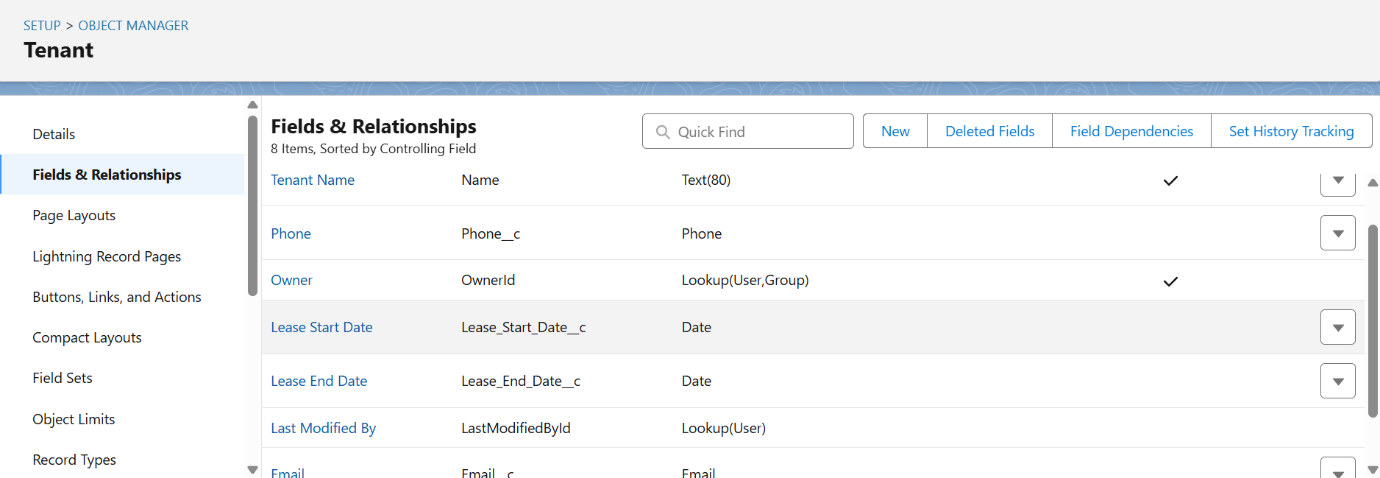
### ****Add Fields to Objects****

Added fields for all objects to capture business data. Examples:

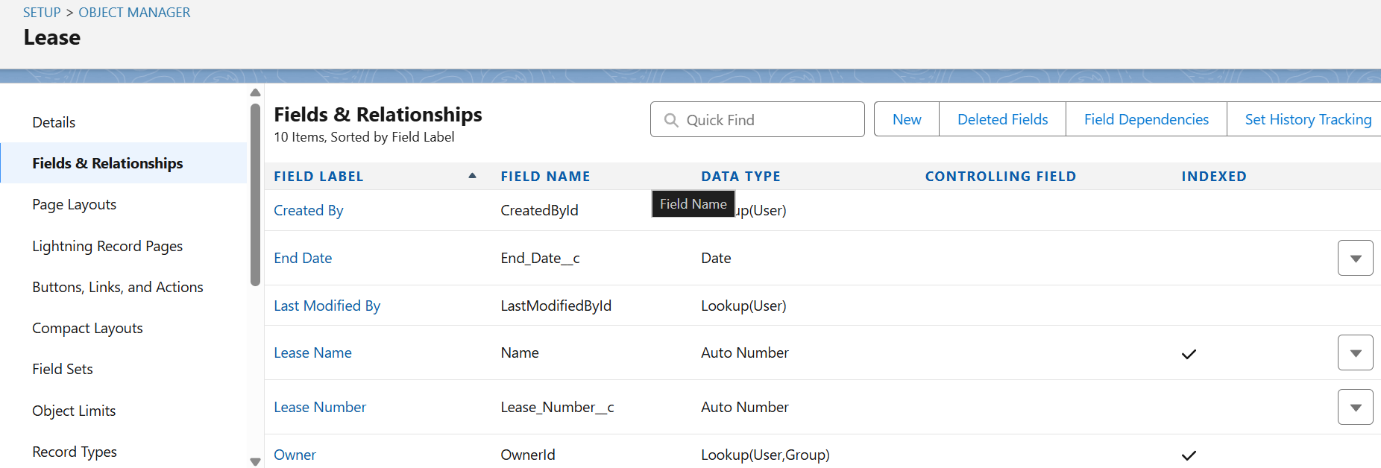
* **Property:** Property Type (Picklist), Address (Text), Manager (Lookup to User), Status (Picklist), Number of Rooms (Number).



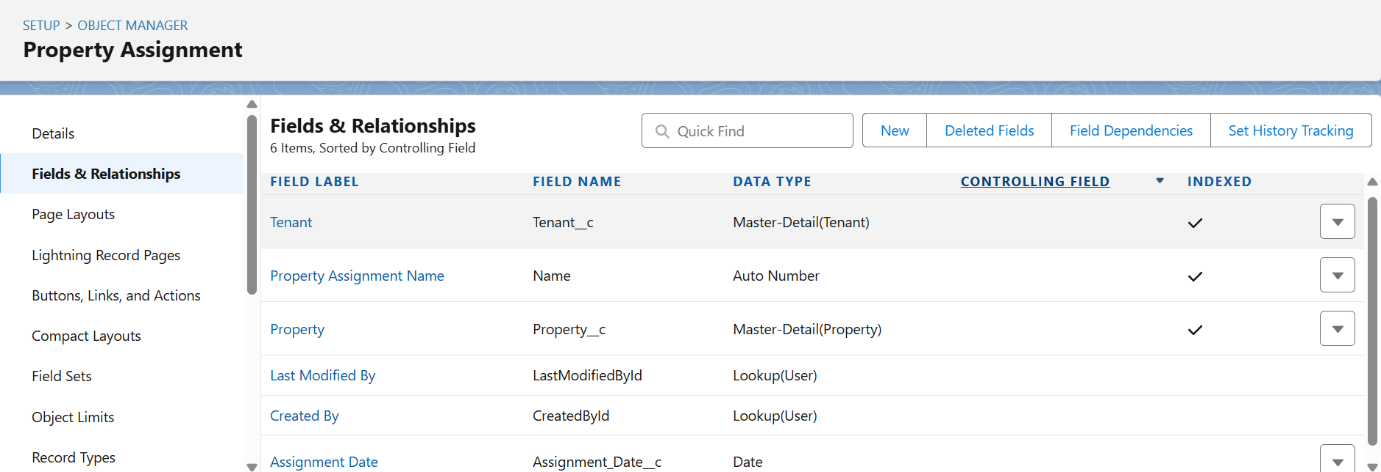
* **Tenant:** Tenant Name, Email, Phone, Lease Dates.



* **Lease:** Lease Number (Auto Number), Property (Lookup), Tenant (Lookup), Start/End Dates, Rent Amount.



* **Property Assignment:** Property (Master-Detail), Tenant (Master-Detail), Assignment Date, Assignment Number.



1. **Record Types**

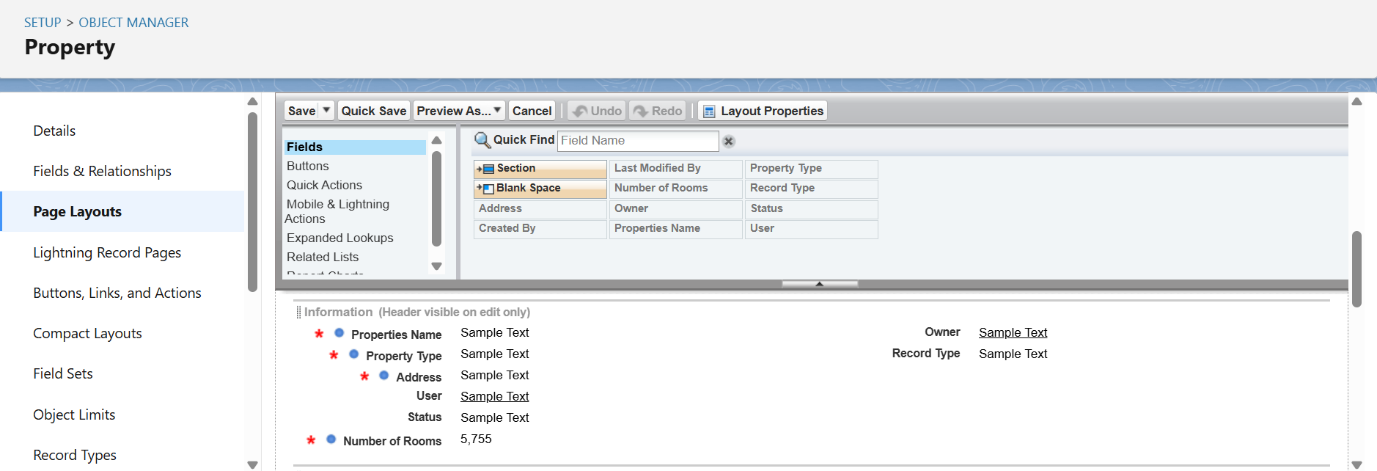
Record Types were configured to categorize records and control layouts:

* **Property:** Residential and Commercial types with different page layouts.
* **Lease:** Active and Expired types.
* **Tenant & Property Assignment:** Master record type only.
* This ensures different layouts and picklist values are shown based on record type.

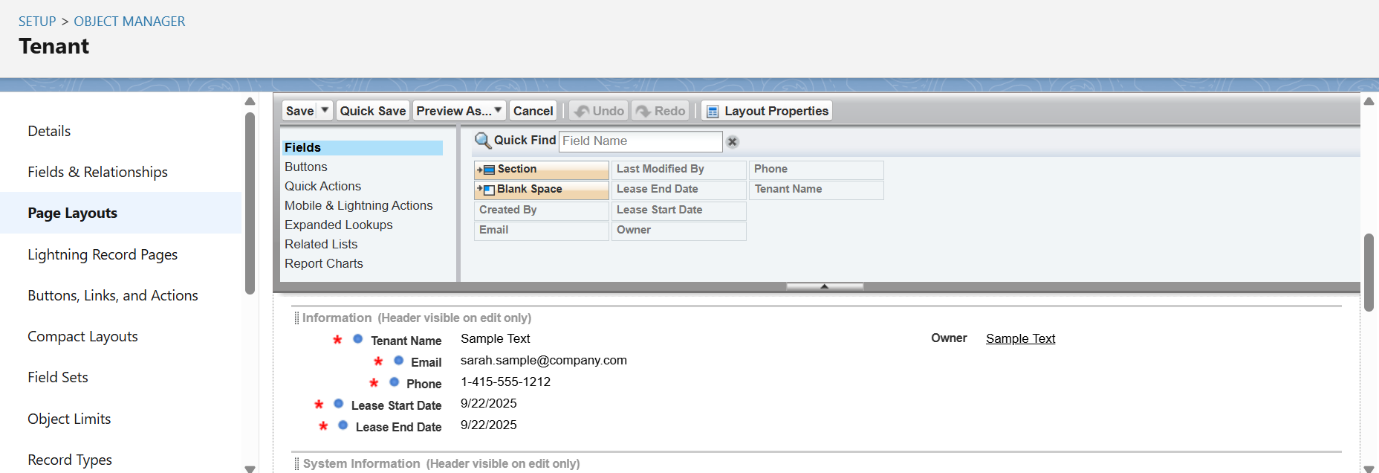
1. **Page Layouts**

Page layouts were designed to display fields and related lists in an organized manner.

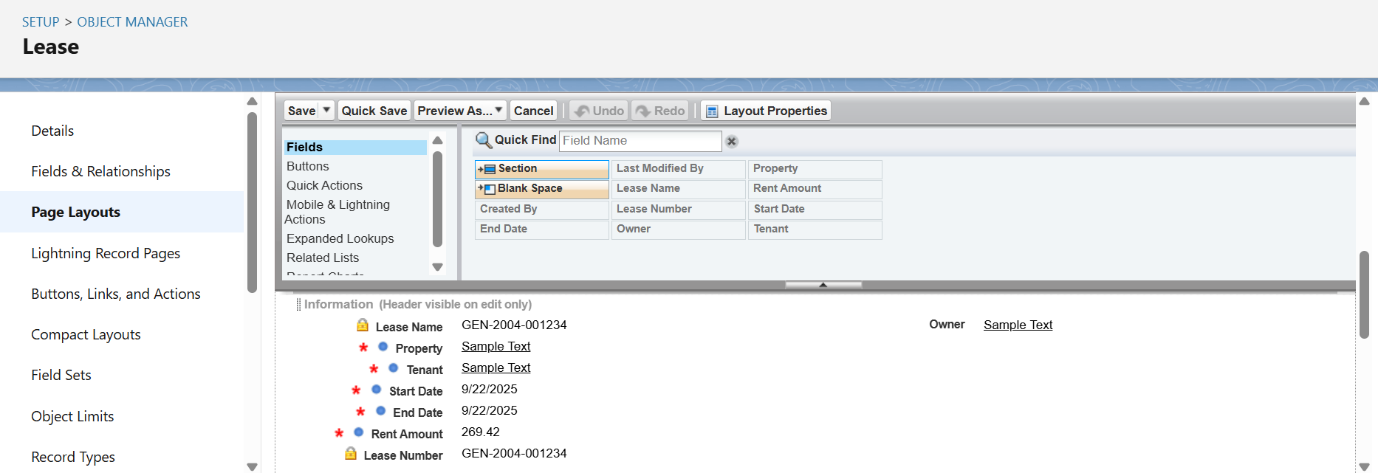
* **Property Section:** Property Name, Property Type, Address, Owner, Status, Number of Rooms



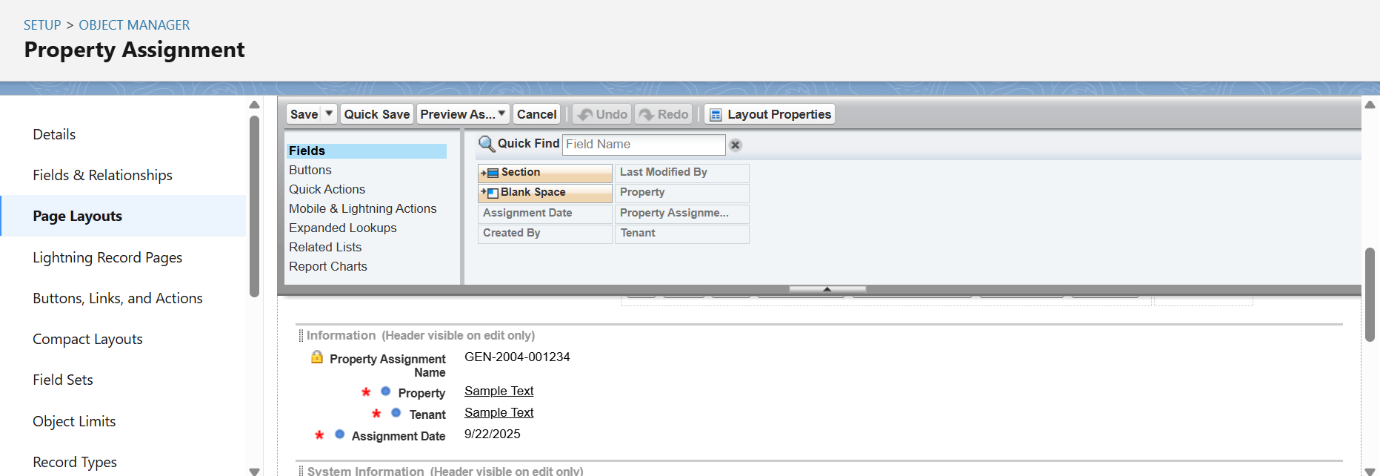
* **Tenant Section:** Tenant Name, Contact Number, Email, Address, Status



* **Lease Section:** Lease ID, Property (Lookup), Tenant (Lookup), Start Date, End Date, Rent Amount, Status



* **Property Assignment Section:** Property (Lookup), Tenant (Lookup), Assignment Date, Role

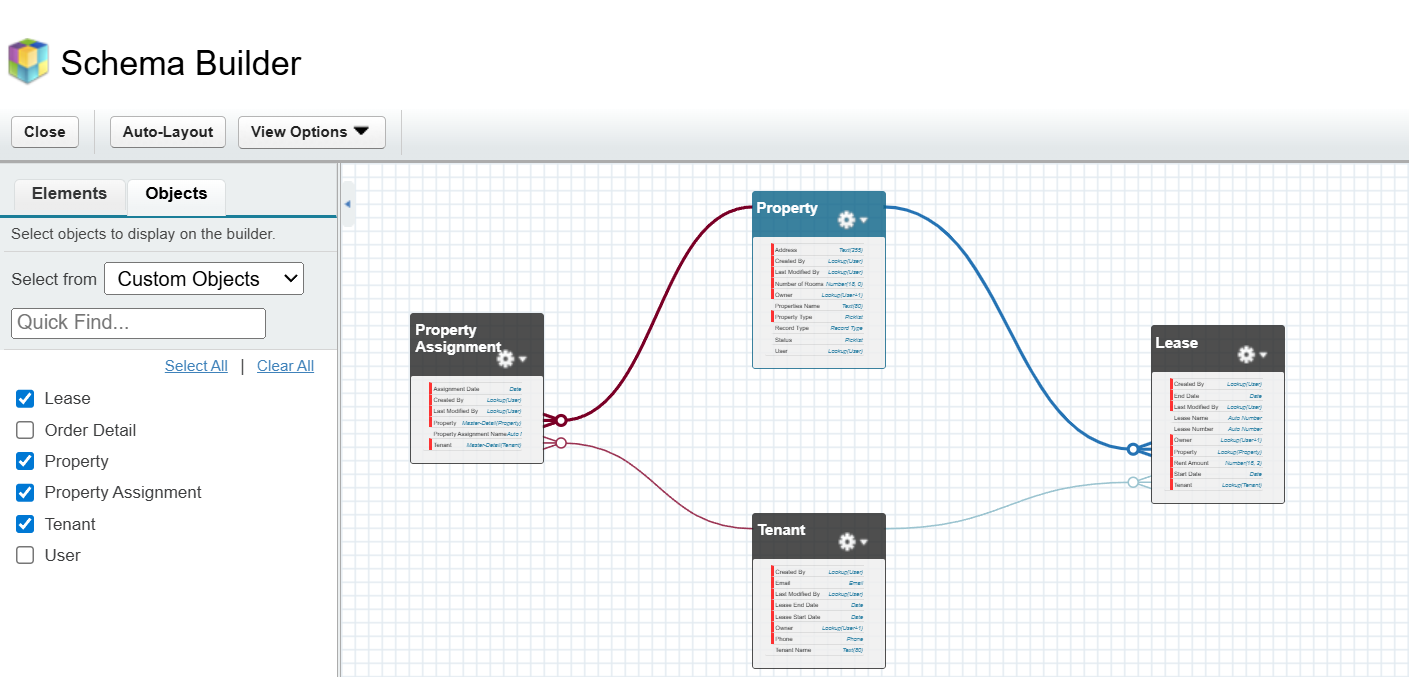


1. **Compact Layouts**

* Compact layouts display the key fields in the highlights panel for quick reference. For the **Property** object, fields include Property Name, Property Type, Address, and Status. For the **Tenant** object, fields include Tenant Name, Contact Number, Email, and Status. For **Lease** and **Property Assignment**, key fields are Lease ID, Property, Tenant, Start/End Date, Assignment Date, and Role.

## ****Schema Builder****

* Schema Builder provides a visual representation of objects, fields, and relationships in Salesforce.
* Selected objects like Property, Tenant, Lease, and Property Assignment are displayed on the canvas.
* Relationships such as lookup, master-detail, and junction are clearly shown.
* New fields or relationships can be added via drag-and-drop for quick configuration.
* Changes are saved and reflected in Object Manager and page layouts.
* It helps verify that all objects and relationships are correctly linked and structured.



## ****Lookup, Master-Detail & Hierarchical Relationships****

* **Lookup Relationship:** Connects two objects independently; deleting the parent does not affect the child.
  + Example: Lease → Property, Tenant → Property.
* **Master-Detail Relationship:** Strong dependency; parent controls ownership, sharing, and deletion of the child.
  + Example: Property Assignment → Property, Property Assignment → Tenant.
* **Hierarchical Relationship:** Special lookup for User objects to define reporting hierarchies.
  + Label: Manager
  + Child Relationship Name: Subordinates
  + Allows viewing all subordinates reporting to a manager.
* Implemented to ensure proper object linkage, data integrity, and enforcement of business rules.

## ****Junction Objects****

* Used to create many-to-many relationships between objects.
* Custom object connects two parent objects via master-detail relationships.
* Example: **Property Assignment** links **Property** and **Tenant**.
* Supports complex assignments and enforces business rules.

## ****External Objects****

* Allow Salesforce to access data stored outside the platform without importing it.
* Configured via **Salesforce Connect** with mapped fields and relationships.
* Example: Access external property databases for reporting and reference.
* Enables real-time integration and keeps data up-to-date.