**Phase 3: Data Modeling & Relationships**

1. **Standard & Custom Objects**

* **Standard Objects**: Already available in Salesforce (Account, Contact, User, Opportunity, Case).
* **Custom Objects**: Created by us for project-specific needs.
* **Customer** – EV users
* **Charging Station** – location & station details
* **Booking** – reservation of a charging slot
* **Payment** – payments for bookings
* **Maintenance Ticket** – station breakdowns

1. **Fields**

* **Customer**
* Name
* Email
* Phone
* Vehicle Number
* Membership Type
* **Charging Station**
* Station Name (Text)
* Location (Text/Geolocation)
* Capacity (Number)
* Status (Picklist: Active, Inactive, Under Maintenance)
* **Booking**
* Booking Number (Auto Number)
* Customer (Lookup → Customer)
* Station (Lookup → Charging Station)
* Start Time (Date/Time)
* End Time (Date/Time)
* Status (Picklist: Pending, Confirmed, Cancelled, Completed)
* **Payment**
* Payment ID (Auto Number)
* Booking (Lookup → Booking)
* Amount (Currency)
* Method (Picklist: Card, UPI, NetBanking, Wallet)
* Status (Picklist: Success, Failed, Pending)
* **Maintenance Ticket**
* Ticket Number (Auto Number)
* Station (Lookup → Charging Station)
* Issue Description (Long Text)
* Reported By (Lookup → User)
* Status (Picklist: Open, In Progress, Resolved)

1. **Record Types**

* **Customer**: EV Individual / Corporate Customer
* **Station**: Urban / Highway station
* **Booking**: Regular Booking / Priority Booking (e.g., for premium members)

1. **Page Layouts**

* **Customer Layout →** Basic Info + Related List (Bookings)
* **Booking Layout** → Customer, Station, Status + Related List (Payments)

1. **Compact Layout**

* **Customer**: Name, Email, Phone
* **Booking**: Booking Number, Status, Start Time, End Time
* **Payment**: Payment ID, Amount, Status

1. **Schema Builder**

* Visually view object relationships.
* Drag & drop new fields.
* Confirm lookups and relationships between objects.

1. **Lookup vs Master-Detail vs Hierarchical**

* **Lookup** → Flexible link between objects (e.g., Booking → Customer).
* **Master-Detail** → Strong link; child record cannot exist without parent (e.g., Payment could be Master-Detail with Booking if we want auto-delete).
* **Hierarchical** → Special relationship only for **User** object.

1. **Junction Objects**

**If we want many-to-many relationships:**

* Example: A Customer can book multiple Stations, and a Station can serve multiple Customers.
* Create a **Junction Object** like *StationBooking* to map Customer ↔ Station with extra details (time, slot) and (Currently, Booking itself is acting like a junction object)

1. **External Objects**

* If we integrate with an external database (e.g., IoT system storing station health), we can use External Objects via Salesforce Connect.