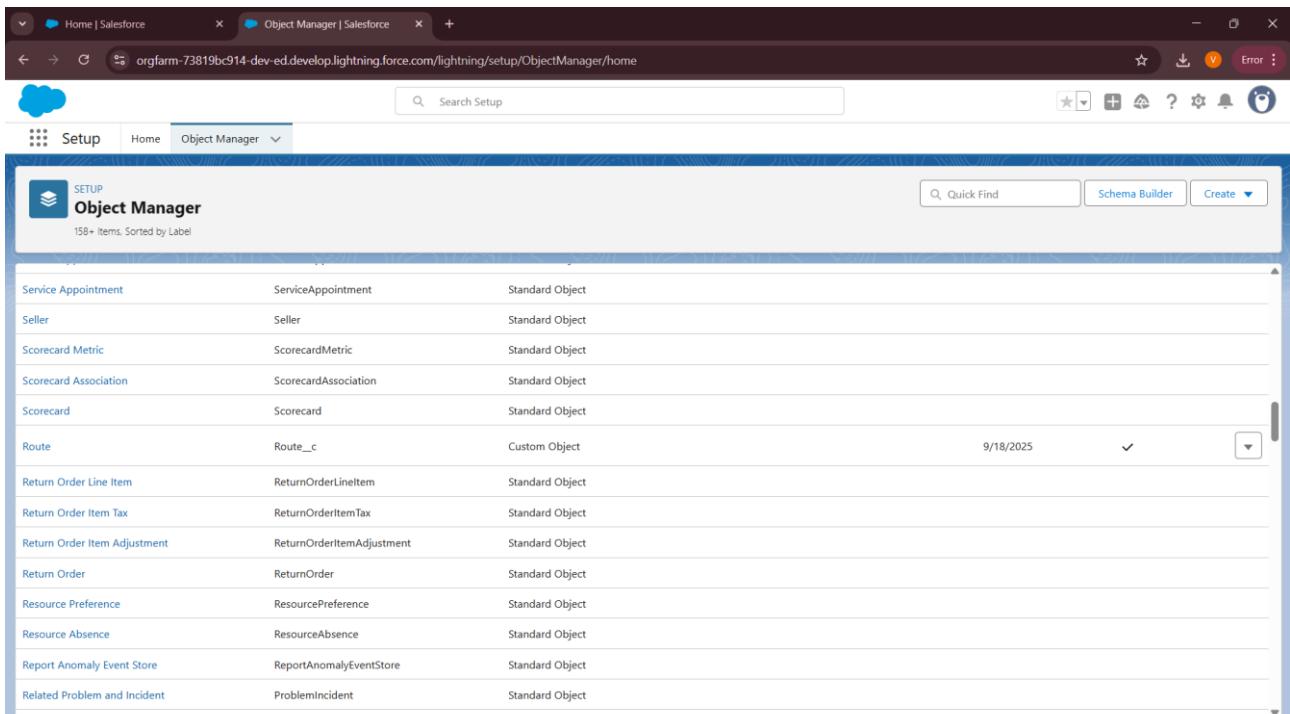


Train Ticket Booking Management System

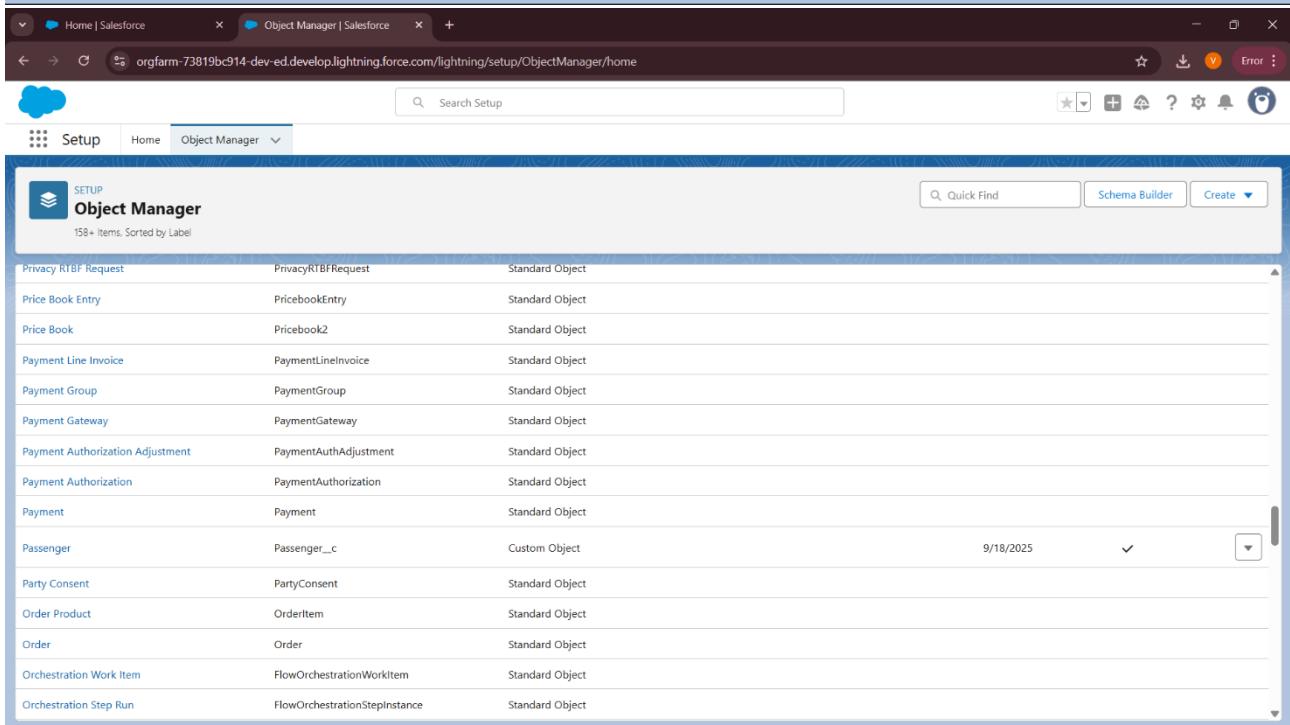
In this phase, we design and configure the data model for the **Train Ticket Booking System**. This involves creating custom objects, fields, relationships, record types, layouts, and schema visualization.

◆ 1. Standard & Custom Objects

- **Standard Objects Used:** User (for Agent assignment), Contact (optional alternative to Passenger).
- **Custom Objects Created:**
 - **Train__c** → Stores train details.
 - **Passenger__c** → Stores passenger/customer details.
 - **Ticket__c** → Stores ticket information (PNR, journey, status).
 - **Booking__c** (optional junction) → Groups passenger & ticket if many-to-many is required.
 - **Route__c** → Stores source, destination, and station details.



The screenshot shows the Salesforce Object Manager interface. The top navigation bar includes links for Home, Object Manager, and a search bar labeled "Search Setup". Below the header, there's a "SETUP" button and tabs for "Object Manager" and "Schema Builder". A "Quick Find" search bar is also present. The main content area displays a table titled "Object Manager" with 158+ items, sorted by Label. The table lists various objects such as Service Appointment, Seller, Scorecard Metric, Scorecard Association, Scorecard, Route, Return Order Line Item, Return Order Item Tax, Return Order Item Adjustment, Return Order, Resource Preference, Resource Absence, Report Anomaly Event Store, and Related Problem and Incident. Each row shows the object name, its API name, and its type (Standard Object or Custom Object). A date field "9/18/2025" and a dropdown arrow are visible in the last column.



This screenshot shows the same Salesforce Object Manager interface, but the list of objects has changed. It now includes Privacy RTBF Request, Price Book Entry, Price Book, Payment Line Invoice, Payment Group, Payment Gateway, Payment Authorization Adjustment, Payment Authorization, Payment, Passenger, Party Consent, Order Product, Order, Orchestration Work Item, and Orchestration Step Run. The table structure remains the same, with columns for object name, API name, type, and a date/time field.

2. Fields

Example Fields Created:

- **Train__c** → Train_No, Train_Name, Source, Destination, Class_Type, Total_Seats, Available_Seats.
- **Passenger__c** → Name, Age, Gender, Email, Phone, Wallet_Balance, ID_Proof, ID_Number.
- **Ticket__c** → PNR (Auto Number), Journey_Date, Status, Seat_No, Price, Booking_Date, Passenger (Lookup), Train (Lookup), Assigned Agent (Lookup to User), Payment_Status, Refund_Status.
- **Route__c** → Route_Name, Source_Station, Destination_Station, Station_List, Departure_Time, Arrival_Time.

Passenger

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Age	Age_c	Number(3, 0)		
Created By	CreatedById	Lookup(User)		
Email_p	Email_p_c	Email		
Gender	Gender_c	Picklist		
ID_Number	ID_Number_c	Text(20) (External ID)		✓
ID_Proof	ID_Proof_c	Picklist		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User, Group)		✓
Passenger Name	Name	Text(80)		✓
Phone_p	Phone_p_c	Phone		

Booking_no

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Booking_no Name	Name	Text(80)		✓
BookingNumber	BookingNumber_c	Auto Number		
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User, Group)		✓
Passenger	Passenger_c	Lookup(Passenger)		✓
Refund_Status	Refund_Status_c	Picklist		
Ticket	Ticket_c	Lookup(Ticket)		✓

The image displays two screenshots of the Salesforce Object Manager interface, showing the 'Fields & Relationships' section for the 'Route' and 'Ticket' objects.

Route Object Fields & Relationships:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Arrival_time	Arrival_time_c	Time		
Created By	CreatedBy	Lookup(User)		
Departure_time	Departure_time_c	Time		
Destination_station	Destination_station_c	Text(20)		
Last Modified By	LastModifiedBy	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Route Name	Name	Text(80)		✓
Route_name	Route_name_c	Text(20)		
Source_station	Source_station_c	Text(20)		
Train	Train_c	Lookup(Train)		✓

Ticket Object Fields & Relationships:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedBy	Lookup(User)		
Journey_Date	Journey_Date_c	Date		
Last Modified By	LastModifiedBy	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Passenger	Passenger_c	Lookup(Passenger)		✓
PNR	PNR_c	Auto Number		
Price	Price_c	Currency(18, 0)		
Seat_No	Seat_No_c	Text(3)		
Status	Status_c	Picklist		
Ticket Name	Name	Text(80)		✓
Train	Train_c	Lookup(Train)		✓

3. Record Types

Created record types on **Ticket__c** to manage business processes:

- Booked
- Cancelled
- Waitlisted
- Pending Payment

The screenshot shows the Salesforce Setup interface with the URL <https://orgfarm-73819bc914-dev-ed.lightning.force.com/lightning/setup/ObjectManager/01lgL000002HxhZ/RecordTypes/view>. The page title is "Ticket | Salesforce". The left sidebar under "Ticket" has "Record Types" selected. The main content area displays a table titled "Record Types" with three items: "Booked", "cancelled", and "Waitlisted". The table includes columns for "RECORD TYPE LABEL", "DESCRIPTION", "ACTIVE", and "MODIFIED BY". A "Quick Find" search bar and "New" and "Page Layout Assignment" buttons are at the top right of the table.

RECORD TYPE LABEL	DESCRIPTION	ACTIVE	MODIFIED BY
Booked		✓	Vasavi Kota, 9/22/2025, 6:05 AM
cancelled			Vasavi Kota, 9/22/2025, 6:05 AM
Waitlisted			Vasavi Kota, 9/22/2025, 6:06 AM

◆ 4. Page Layouts

- **Ticket — Passenger Layout:** Only booking details, no refund or agent fields.
- **Ticket — Agent Layout:** Includes Assigned Agent, Refund Status, Payment Status.
- **Ticket — Admin Layout:** Full access to all fields.

Layouts are assigned to profiles based on record type.

◆ 5. Compact Layouts

Compact layouts were configured for quick view in Lightning header & mobile:

- **Train__c** → Train Name, Train No, Available Seats.
- **Ticket__c** → PNR, Status, Journey Date, Train.

The image displays two separate screenshots of the Salesforce Setup Object Manager interface, both titled "Compact Layout Detail".

Screenshot 1 (Top): Train Compact Layout

- Title:** Train Compact Layout
Train_compact_layout
- Object Name:** Train
- Compact Layout Detail:**
 - Label:** Train_compact_layout
 - API Name:** Train_compact_layout
 - Included Fields:** Available_seats, Class_type, Train_Name, Train_Number, seat_no, Destination.
 - Created By:** Vasavi.Kota, 9/22/2025, 6:10 AM
 - Modified By:** Vasavi.Kota, 9/22/2025, 6:10 AM
- Left Sidebar (Compact Layouts section):**
 - Details
 - Fields & Relationships
 - Page Layouts
 - Lightning Record Pages
 - Buttons, Links, and Actions
 - Compact Layouts** (selected)
 - Field Sets
 - Object Limits
 - Record Types
 - Related Lookup Filters
 - Restriction Rules
 - Scoping Rules
 - Object Access
 - Triggers
 - Flow Triggers

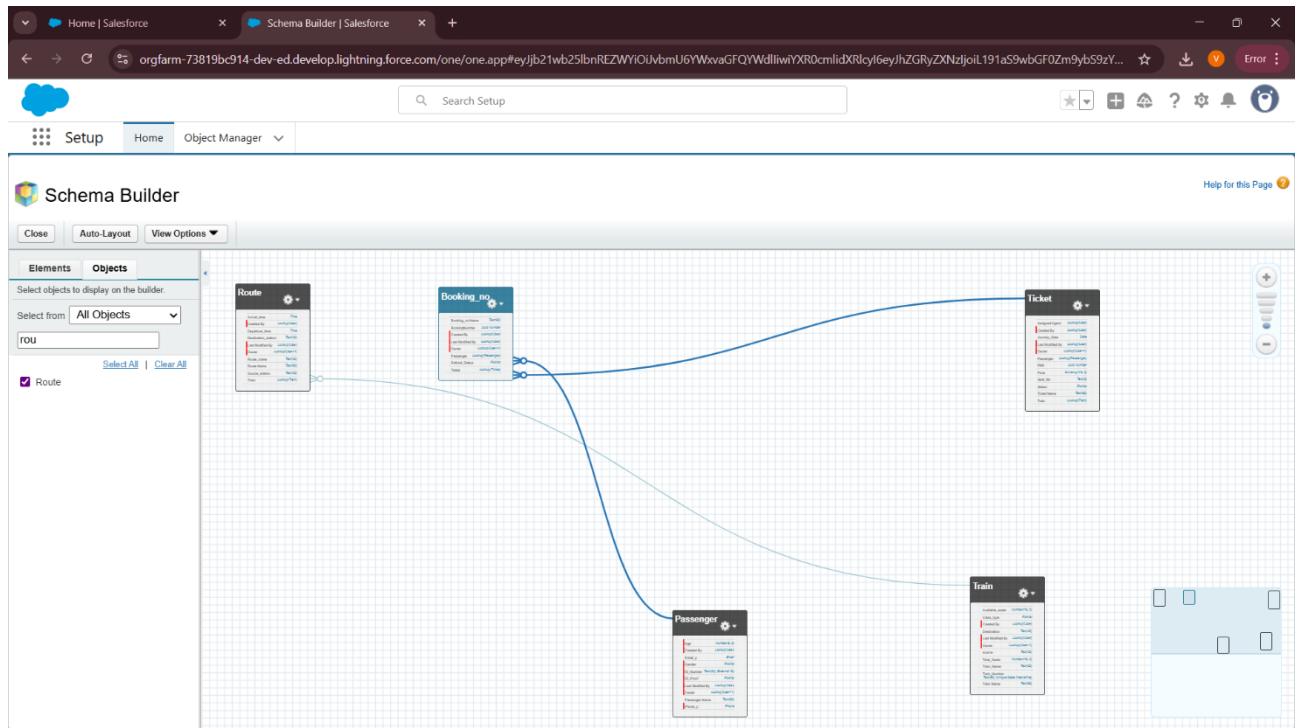
Screenshot 2 (Bottom): Ticket Compact Layout

- Title:** Ticket Compact Layout
Ticket_compact_layout
- Object Name:** Ticket
- Compact Layout Detail:**
 - Label:** Ticket_compact_layout
 - API Name:** Ticket_compact_layout
 - Included Fields:** PNR, Passenger, Train, Seat_No, Price, Status, Journey_Date.
 - Created By:** Vasavi.Kota, 9/22/2025, 6:09 AM
 - Modified By:** Vasavi.Kota, 9/22/2025, 6:09 AM
- Left Sidebar (Compact Layouts section):**
 - Details
 - Fields & Relationships
 - Page Layouts
 - Lightning Record Pages
 - Buttons, Links, and Actions
 - Compact Layouts** (selected)
 - Field Sets
 - Object Limits
 - Record Types
 - Related Lookup Filters
 - Restriction Rules
 - Scoping Rules
 - Object Access
 - Triggers
 - Flow Triggers

❖ 6. Schema Builder

- Verified relationships using **Schema Builder**.
- Relationships configured:
 - **Ticket → Train** = Lookup (1 Train : Many Tickets).
 - **Ticket → Passenger** = Lookup (1 Passenger : Many Tickets).
 - **Ticket → Assigned Agent** = Lookup(User).
 - **Booking → Ticket & Passenger** = optional Master-Detail (junction).

- **Route → Train** = Lookup.
- Schema Builder used to visualize and confirm links between objects.



◆ 7. Relationship Choices

- **Lookup** used where we want flexibility (Train ↔ Ticket, Passenger ↔ Ticket).
- **Master-Detail** used in junction (Booking ↔ Ticket, Booking ↔ Passenger) if grouping is required.
- **Hierarchical Relationship** is only for User object (not applied here).
- This design ensures data safety (Tickets not auto-deleted if Train is deleted)