

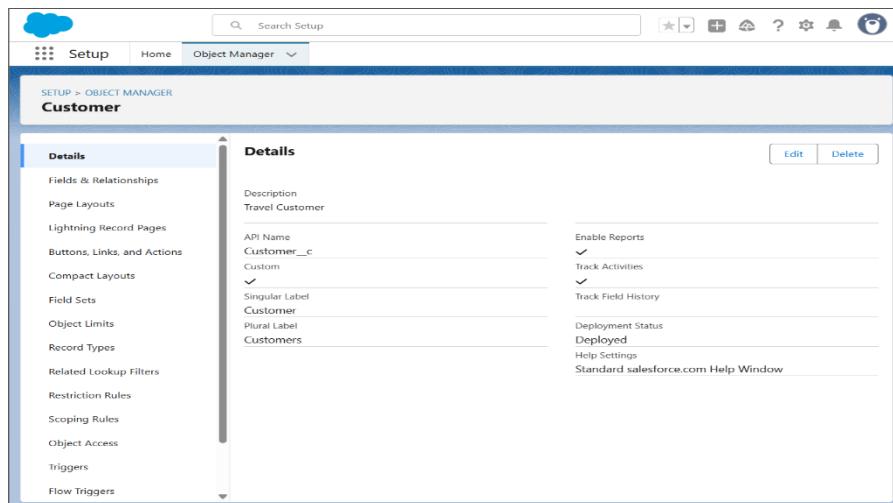
Phase 3 — Data Modeling & Relationships

Objective

To design the data structure of **Smart TravelConnect CRM** by creating custom objects, fields, and relationships that capture the travel agency's core processes: customers, travel packages, bookings, payments, and feedback.

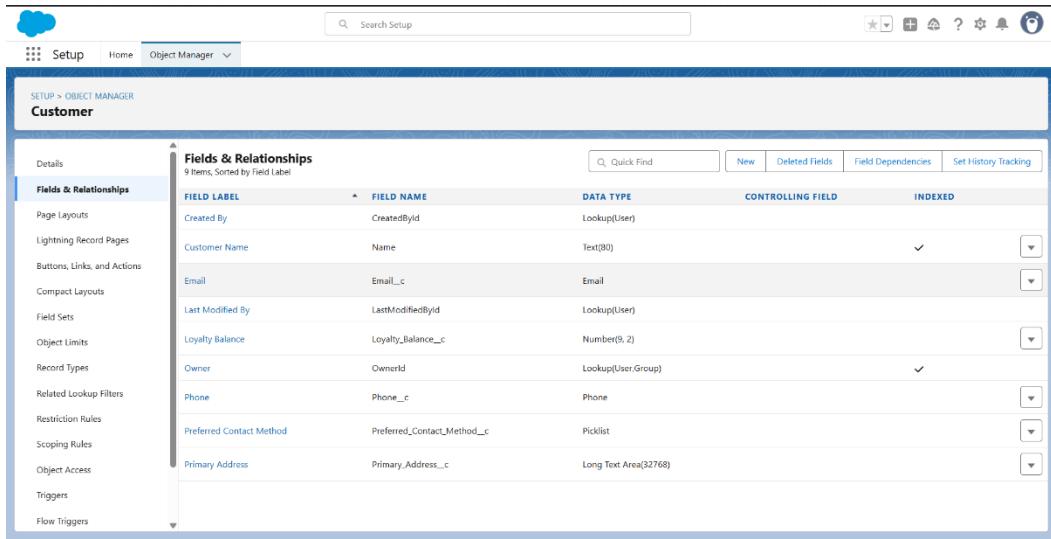
Step 1: Create the Customer Object

1. Go to **Setup** → **Object Manager** → **Create** → **Custom Object**.
2. Enter details:
 - o Label: Customer
 - o Plural Label: Customers
 - o Record Name: Customer Name
 - o Check **Allow Reports** and **Allow Activities**.
 - o Save.



3. Add fields:
 - o First Name → Text(40), Required

- Last Name → Text(80), Required
- Email → Email
- Phone → Phone
- Loyalty Points → Number(10,0), Default 0
- Address → Long Text Area (255)



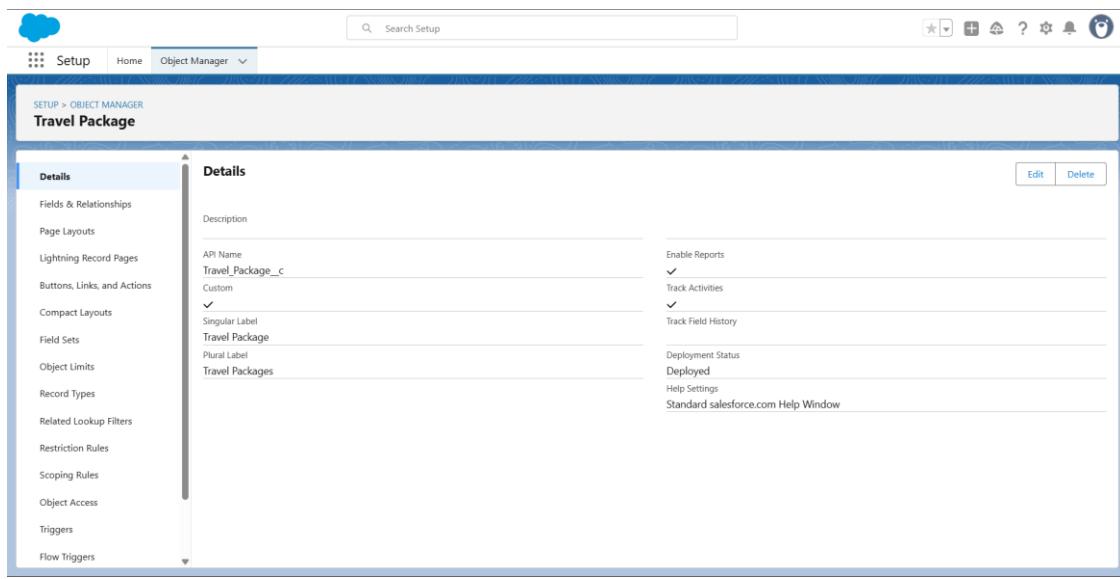
The screenshot shows the Salesforce Object Manager interface for the 'Customer' object. The left sidebar lists various setup options like Details, Fields & Relationships, Page Layouts, etc. The main area displays the 'Fields & Relationships' section with a table of fields:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		✓
Customer Name	Name	Text(80)		✓
Email	Email__c	Email		✓
Last Modified By	LastModifiedById	Lookup(User)		✓
Loyalty Balance	Loyalty_Balance__c	Number(9, 2)		✓
Owner	OwnerId	Lookup(User,Group)		✓
Phone	Phone__c	Phone		✓
Preferred Contact Method	Preferred_Contact_Method__c	Picklist		✓
Primary Address	Primary_Address__c	Long Text Area(32768)		✓

4. Configure Compact Layout to display: Full Name, Email, Phone.

Step 2: Create the Travel Package Object

1. Setup → Object Manager → Create → Custom Object.
2. Enter details:
 - Label: Travel Package
 - Plural: Travel Packages
 - Record Name: Package Name (Text).
 - Save.



3. Add fields:

- o Destination → Text(100)
- o Start Date → Date
- o End Date → Date
- o Address → Text(255)
- o Booked Count → Roll-Up Summary
- o Price → Currency(16,2)
- o Package Type → Picklist (Values: Standard, Premium, Customized)
- o Package Code → Auto Number
- o Status → Picklist
- o View Map → Formula(Text)
- o Active → Checkbox (Default: TRUE)

Step 3: Create the Booking Object

1. Create object:

- Label: Booking
- Plural: Bookings
- Record Name: Booking Number (Auto Number format BK-{0000}).

2. Add fields:

- Customer → Lookup(Customer)
- Booking Name → Text(80)

- Booking Number → Auto Number
- Travel Start Date → Date (Required)
- Travel End Date → Date
- Number of Passengers → Number(3,0), Default 1
- Booking Status → Picklist (Values: Pending, Confirmed, Cancelled, Completed, Refunded) → Default: Pending
- Total Amount → Currency
- Total Paid → Currency (Roll-Up from Payments)
- Balance Due → Formula (Currency) = Total Amount – Total Paid
- Total Booked Value → Roll Up Summary
- Agent → Lookup(User)
- Source → Picklist (Website, Walk-in, Phone, Social, Campaign)

Fields & Relationships			
15 Items, Sorted by Field Label			
Agent_c	Agent_c_c	Lookup(User)	▼
Balance Due	Balance_Due_c_c	Formula (Currency)	▼
Booking Name	Name	Text(80)	▼
Booking Number	Booking_Number_c_c	Auto Number	▼
Booking Status	Booking_Status_c_c	picklist	▼
Booking_Date_c	Booking_Date_c_c_c	Date	▼
Created By	CreatedBy	Lookup(User)	▼
Customer	Customer_c_c	Master-Detail(Customer)	▼
End_Date_c	End_Date_c_c_c	Date	▼
Last Modified By	LastModifiedBy	Lookup(User)	▼
Source	Source_c_c	picklist	▼
Start_Date_c	Start_Date_c_c_c	Date	▼
Total Amount	Total_Amount_c_c	Roll-Up Summary (SUM Booking Package)	▼
Total Booked Value	Total_Booked_Value_c_c	Roll-Up Summary (SUM Booking Package)	▼
Total Paid	Total_Paid_c_c	Roll-Up Summary (SUM Payment)	▼

3. Add Validation Rules:

- Travel Start Date cannot be earlier than today.
- Travel End Date must be on/after Travel Start Date.

The screenshot shows the Salesforce Setup interface under the Object Manager section for the 'Booking' object. On the left, a sidebar lists various configuration options like Details, Fields & Relationships, Page Layouts, and Lightning Record Pages. The main content area displays the 'Booking Validation Rule' configuration. The rule is named 'Travel_Date_Validation' and has the formula 'Start_Date_c < TODAY()'. It is active and associated with the field 'Start_Date_c'. The 'Description' field contains the message 'Travel date cannot be in the past.' The 'Created By' field shows 'Yash Dighade' with a timestamp of '9/20/2025, 9:23 PM'. The 'Modified By' field also shows 'Yash Dighade' with the same timestamp.

Step 4: Create the Payment Object

1. Create object:
 - o Label: Payment
 - o Plural: Payments

The screenshot shows the Salesforce Setup interface under the Object Manager section for the 'Payment' object. The sidebar on the left is identical to the previous screenshot. The main content area shows the 'Details' tab for the 'Payment' object. The 'API Name' is set to 'Payment__c'. The 'Custom' checkbox is checked. The 'Singular Label' is 'Payment' and the 'Plural Label' is 'Payments'. On the right side, there are sections for 'Enable Reports', 'Track Activities', 'Track Field History', 'Deployment Status' (set to 'Deployed'), 'Help Settings', and 'Standard salesforce.com Help Window'.

2. Create Master-Detail Relationship → Parent = Booking.
3. Add fields:

- o Amount → Currency(16,2), Required

- Payment Date → Date (Default = Today)
- Payment Mode → Picklist (Card, UPI, NetBanking, Cash, Wallet)
- Transaction ID → Text(80)
- Status → Picklist (Paid, Failed, Pending, Refunded)

4. Create Roll-Up Summary on Booking:

- Field Name: Total Paid
- Type: SUM of Payment Amount.

The screenshot shows the Salesforce Setup interface for the Payment object. The left sidebar has a tree view with 'Fields & Relationships' selected under 'Payment'. The main content area is titled 'Fields & Relationships' and lists 11 items, sorted by Field Label. Each item shows the field name, field label, and data type. For example, 'Amount' is a Currency(16, 2) field, and 'Booking' is a Master-Detail(Booking) field. Other fields include 'Created By' (Lookup(User)), 'Last Modified By' (Lookup(User)), 'Payment Date' (Date/Time), 'Payment Mode' (Picklist), 'Payment Name' (Text(80)), 'Payment Number' (Auto Number), 'Receipt Link' (URL(255)), 'Status' (Picklist), and 'Transaction Id' (Text(255) (External ID) (Unique Case Insensitive)).

Step 5: Create the Feedback Object

1. Create object:
 - Label: Feedback
 - Record Name: Feedback Number (Auto Number).
2. Add fields:
 - Booking → Lookup(Booking)
 - Customer → Lookup(Customer)
 - Rating → Picklist(1–5)
 - Comments → Long Text Area(32768)

The screenshot shows the Salesforce Setup interface with the 'Object Manager' selected. The main area displays the 'Fields & Relationships' section for the 'Feedback' object. On the left, a sidebar lists various setup categories like Details, Page Layouts, Lightning Record Pages, etc. The main table lists eight fields with their labels, names, data types, controlling fields, and indexing status.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Booking	Booking__c	Lookup(Booking)		✓
Comments	Comments__c	Long Text Area(32768)		✗
Created By	CreatedById	Lookup(User)		✗
Customer	Customer__c	Lookup(Customer)		✓
Feedback Name	Name	Text(80)		✓
Last Modified By	LastModifiedById	Lookup(User)		✗
Owner	OwnerId	Lookup(User,Group)		✓
Rating	Rating__c	Picklist		✗

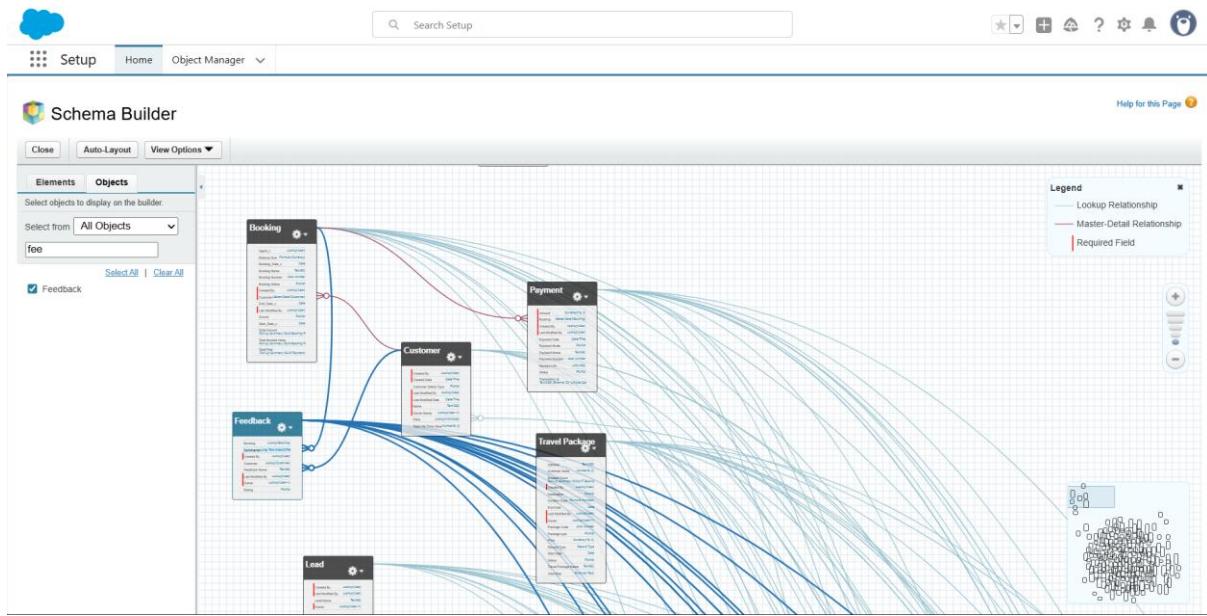
Step 6: Define Relationships

- **Customer → Booking:** One-to-Many (one customer can create many bookings).
- **Booking → Payment:** One-to-Many (one booking can have many payments).
- **Travel Package ↔ Booking:** Many-to-Many (using a junction object *Booking_Package* if needed).
- **Feedback → Booking & Customer:** Lookup to connect feedback with both customer and booking.

Step 7: Verify with Schema Builder

1. Go to **Setup → Schema Builder**.
2. Drag all created objects (Customer, Travel Package, Booking, Payment, Feedback).
3. Visually confirm the relationships:
 - Customer → Booking (Lookup).
 - Booking → Payment (Master-Detail).
 - Feedback → Customer + Booking.

4. Save schema diagram.



Outcome of Phase 3:

The data model of Smart TravelConnect CRM is successfully created with properly linked objects, fields, and relationships to manage customers, packages, bookings, payments, and feedback.