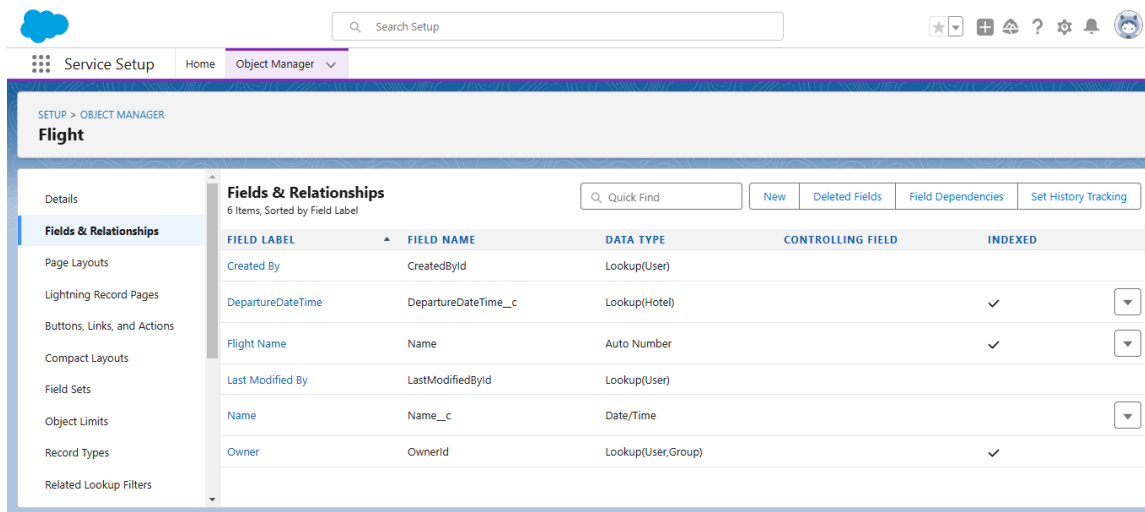


# PROJECT TITLE: TRIPADVISOR E-MANAGEMENT

## TEAM MEMEBERS :

1. SARANYA G
2. DEEPA C
3. KEERTHANA T
4. RAMYA P

## Create Object:



The screenshot displays the Salesforce Setup interface for the 'Flight' object. The left sidebar shows navigation options: Details, Fields & Relationships (selected), Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, and Related Lookup Filters. The main content area is titled 'Flight' and shows the 'Fields & Relationships' section with 6 items, sorted by Field Label. A 'Quick Find' search bar is present. The table lists the following fields:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
DepartureDateTime	DepartureDateTime__c	Lookup(Hotel)		✓
Flight Name	Name	Auto Number		✓
Last Modified By	LastModifiedById	Lookup(User)		
Name	Name__c	Date/Time		
Owner	OwnerId	Lookup(User,Group)		✓

At the top of the setup page, there is a 'Search Setup' bar and a navigation menu with 'Service Setup', 'Home', and 'Object Manager' (selected). The top right corner contains various utility icons.

Service Setup Home Object Manager

Search Setup

SETUP > OBJECT MANAGER

### Customer

Details

**Fields & Relationships**  
8 Items, Sorted by Field Label

Quick Find New Deleted Fields Field Dependencies Set History Tracking

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Customer Name	Customer_Name__c	Text(50)		
Customer Name	Name	Text(80)		✓
Discount Amount	Discount_Amount__c	Currency(18, 0)		
Discount Percent	Discount_Percent__c	Percent(18, 0)		
Email__c	Email_c__c	Email		
Last Modified By	LastModifiedById	Lookup(User)		

## Food Option :

Service Setup Home Object Manager

Search Setup

SETUP > OBJECT MANAGER

### Food Option

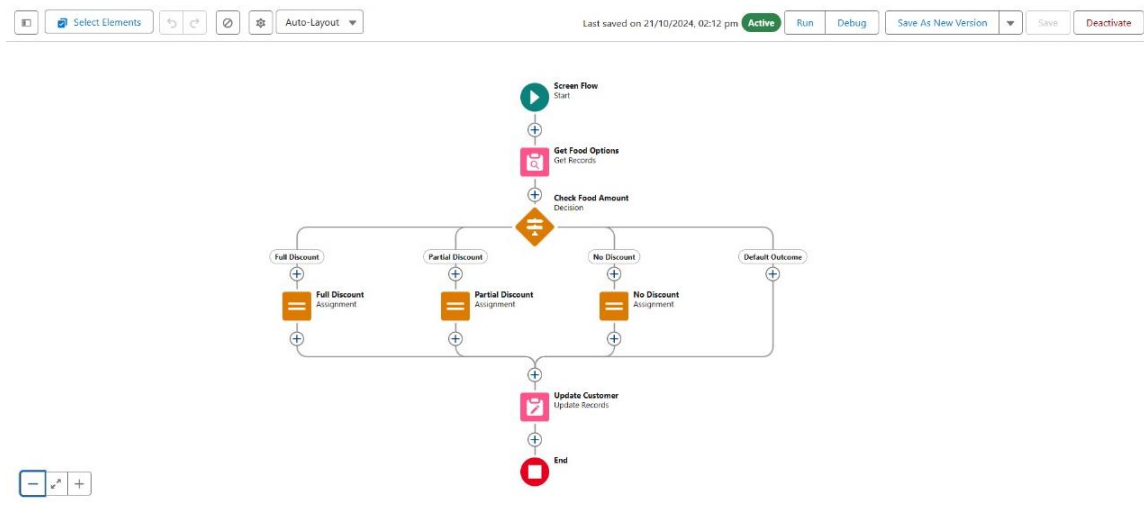
Details

**Fields & Relationships**  
7 Items, Sorted by Field Label

Quick Find New Deleted Fields Field Dependencies Set History Tracking

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Food Amount	Food_Amount__c	Currency(18, 0)		
Food Option Name	Name	Auto Number		✓
Hotel	Hotel__c	Lookup(Hotel)		✓
Last Modified By	LastModifiedById	Lookup(User)		
Name	Name__c	Text(50)		
Owner	OwnerId	Lookup(User,Group)		✓

## Create Flow :



## Apex Triggers :

```

1 trigger FoodOptionTrigger on Food_Option__c (after insert, after update, after delete, after undelete) {
2
3     if (Trigger.isAfter) {
4         if (Trigger.isInsert || Trigger.isUpdate || Trigger.isUndelete) {
5             FoodOptionTriggerHandler.updateHotelInformation(Trigger.new);
6         }
7         if (Trigger.isDelete) {
8             FoodOptionTriggerHandler.updateHotelInformation(Trigger.old);
9         }
10    }
11 }
  
```

The screenshot shows the Apex Trigger code in the IDE. The trigger is named 'FoodOptionTrigger' and is triggered on the 'Food\_Option\_\_c' object. It is an 'after' trigger that fires on 'insert', 'update', 'delete', and 'undelete' events. The trigger logic is as follows:

- If the trigger is an 'after' trigger, check if it is an 'insert', 'update', or 'undelete' event. If so, call the 'updateHotelInformation' method of the 'FoodOptionTriggerHandler' class with 'Trigger.new' as the argument.
- If the trigger is a 'delete' event, call the 'updateHotelInformation' method of the 'FoodOptionTriggerHandler' class with 'Trigger.old' as the argument.

```

1 public class FoodOptionTriggerHandler {
2
3     // Method to update hotel information based on food options
4     public static void updateHotelInformation(List<Food_Option__c> foodOptions) {
5
6         Set<Id> hotelIdsToUpdate = new Set<Id>();
7
8         // Collect unique Hotel Ids affected by food options changes
9         for (Food_Option__c foodOption : foodOptions) {
10             hotelIdsToUpdate.add(foodOption.Hotel__c);
11         }
12
13         // Aggregate the total food options count for each hotel in one query
  
```

The screenshot shows the Apex TriggerHandler code in the IDE. The class is named 'FoodOptionTriggerHandler' and contains a static method 'updateHotelInformation' that takes a list of 'Food\_Option\_\_c' objects as input. The method logic is as follows:

- Initialize a 'Set<Id>' named 'hotelIdsToUpdate'.
- Iterate over each 'Food\_Option\_\_c' object in the 'foodOptions' list. For each object, add its 'Hotel\_\_c' value to the 'hotelIdsToUpdate' set.
- Aggregate the total food options count for each hotel in one query.

```
File • Edit • Debug • Test • Workspace • Help • < >
FoodOptionTriggerHandler.apxc
Code Coverage: None API Version: 62 Go To
14 Map<Id, Integer> hotelFoodOptionCountMap = new Map<Id, Integer>();
15
16 for (AggregateResult ar : [
17     SELECT Hotel__c, COUNT(Id) totalFoodOptions
18     FROM Food_Option__c
19     WHERE Hotel__c IN :hotelIdsToUpdate
20     GROUP BY Hotel__c
21 ]) {
22     hotelFoodOptionCountMap.put((Id) ar.get('Hotel__c'), (Integer) ar.get('totalFoodOptions'));
23 }
24
25 // Fetch hotels and update TotalFoodOptions__c based on the aggregated count
26 List<Hotel__c> hotelsToUpdate = [
27     SELECT Id, TotalFoodOptions__c
28     FROM Hotel__c
29     WHERE Id IN :hotelIdsToUpdate
30 ];
31
32 for (Hotel__c hotel : hotelsToUpdate) {
33     Integer totalFoodOptions = hotelFoodOptionCountMap.get(hotel.Id);
34     hotel.TotalFoodOptions__c = totalFoodOptions != null ? totalFoodOptions : 0;
35 }
36
37 // Update hotels with new total food options count
38 update hotelsToUpdate;
39 }
40 }
```

Name	Line	Problem
------	------	---------

```
File • Edit • Debug • Test • Workspace • Help • < >
FoodOptionTriggerHandler.apxc
Code Coverage: None API Version: 62 Go To
28 FROM Hotel__c
29 WHERE Id IN :hotelIdsToUpdate
30 ];
31
32 for (Hotel__c hotel : hotelsToUpdate) {
33     Integer totalFoodOptions = hotelFoodOptionCountMap.get(hotel.Id);
34     hotel.TotalFoodOptions__c = totalFoodOptions != null ? totalFoodOptions : 0;
35 }
36
37 // Update hotels with new total food options count
38 update hotelsToUpdate;
39 }
40 }
```

Name	Line	Problem
------	------	---------

Apex Schedule :

