Tour Management System with Travel Package Customization and Vendor Integration

Team 17:

P2 Comment:

Rethink how "services" is done. If a customer wants to book a specific room, they can't because there is no relationship to Accommodation, same with Transportation and Meal. Perhaps think about generalization for these services, and categorization to solve the problem.

In our Logical ERD design, We have addressed the concern about booking specific services by using a Generalization-Specialization approach. A generalized 'Service' entity captures common attributes such as Service_ID, Service_Name,Service_price and, while specialized sub-entities like Accommodation, Transportation, Meal, and Activity inherit these features and contain specific attributes. For example, Accommodation includes attributes like Room_Type, Price_per_night and while Transportation specifies Vehicle_Type. This setup ensures that each specialized service can be uniquely defined and selected based on its specific characteristics.

To link these services to customer bookings, We have implemented a 'Service_Booking' associative entity, which connects bookings to individual services. This entity allows for capturing details like the number of rooms, meals, or activities selected and the total cost associated with them. With this structure, customers can directly choose specific services, such as a Deluxe Room or a private Car, while the system accurately tracks and calculates the total cost based on the selected options. Incorporating total specialization ,means that every instance of the **Service** entity must belong to one of the subtypes (specific service types), ensuring that all services are categorized.

Improvements and Changes:

Generalization of Services Entity:

Created a generalized **Service** entity with common attributes (Service_ID, Service_Name, Service_Type, Service_Price, Service_Availability).

Established **specialized sub-entities** (Accommodation, Transportation, Meal, Activity) that inherit from the generalized Service entity.

This approach uses **Generalization-Specialization** to group common features while accommodating the specific characteristics of each type of service.

Defining Specialized Sub-Entities:

1. Accommodation

- **AccommodationID**
- **Room_Type** (e.g., Deluxe Room, Suite, Standard Room)

2. Transportation

- **TransportationID**
- **Vehicle_Type** (e.g., Car, Van, Bus)
- **Vehicle_capacity**

3. Meal

- **MealID**
- **Meal_Type** (e.g., Breakfast, Lunch, Dinner)
- **Cuisine_Type** (e.g., Italian, Indian, Continental)

4. Activity

- **ActivityID**
- **Activity_Name** (e.g., Sightseeing, Hiking, Water Sports)
- **Activity_Type** (e.g., Adventure, Relaxation, Educational)
- **Activity_Duration** (e.g., 2 hours, Full day)

Revised Relationships:

Established relationships between the generalized **Service** entity and the specialized entities (Accommodation, Transportation, Meal, Activity) using inheritance (generalization-specialization).

Updated the **Booking_Service** associative entity to reference the generalized **Service** entity, allowing customers to book specific rooms, transport services, meals, or activities.

Booking_ID: A foreign key referencing the unique identifier of the associated booking.

Service_ID: A foreign key referencing the unique identifier of the selected service.

Quantity: The number of units of the selected service included in the booking.

Price: The price for the service

Total_Cost: The calculated total cost for the selected service based on quantity and service pricing.

Check in Date: Hotel Checkin

Check out Date: Hotel checkout

For more understanding:

Booking_ID	*Service_ID*	*Quantity*	*Service_Price*	*Total_Service_Cost*
101	*1*	*5 nights*	*$150 per night*	*$750*
101	*2*	*5 days*	*$50 per day*	*$250*
101	*3*	*10 meals*	*$20 per meal*	*$200*
101	*4*	*1 tour*	*$100 per person*	**$100**

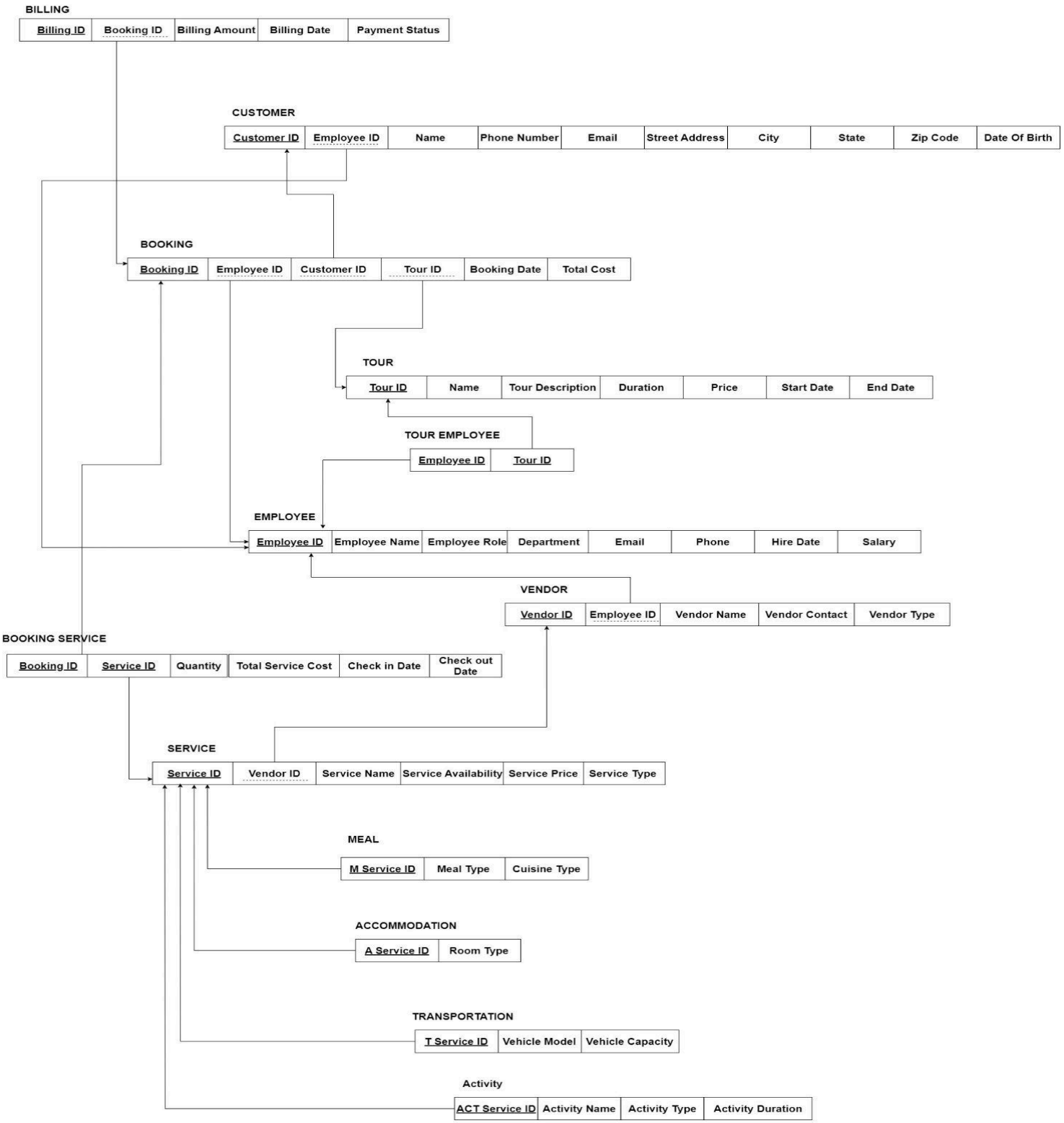
Explanation:

- The customer booked **5 nights** in a deluxe room (Accommodation) at $150 per night → Total: **$750**.
- The customer booked **5 days** of private car service (Transportation) at $50 per day → Total: **$250**.
- The customer selected **10 meals** (2 meals per day x 5 days) (Meal) at $20 per meal → Total: **$200**.
- The customer selected a **1-person sightseeing tour** (Activity) at $100 → Total: **$100**.

Normalization:

All tables in the database design for the Tour Management System have been thoroughly normalized to ensure data integrity and reduce redundancy. Each entity is structured to meet the criteria for Third Normal Form (3NF), meaning that all attributes are atomic, fully functionally dependent on their primary keys, and free from transitive dependencies. The composite primary key in the Booking Service table effectively captures the many-to-many relationship between bookings and services, while each entity's attributes are clearly defined and directly related to their respective primary keys. This normalization process enhances the overall efficiency of the database, facilitates accurate data retrieval, and simplifies future modifications.

RELATIONAL SCHEMA



Removal of Composite Attributes in Customer Table:

Customer Entity had an Address Field containing composite attributes(Street Address,city,state, zipcode) , all these attributes were converted into separate columns to remove the composite attribute address.

Logical Entity Relationship Diagram