**Travel Company-: Geetanjali Gupta**

**I. SCENARIO**

This database was designed for a travel company to organize travel itineraries and packages for travelers. This database will allow the company to keep track of its traveler’s information, payments, and trip bookings. The traveler is able to also make bookings for accommodation and transportation at their chosen destination. They may decide to travel solo or as part of a group and also choose a tour package and tour guide that can further structure their trip and activities.

**II. MODEL WALKTHROUGH**

**a. The Central Event**

The booking event is central to the model. The relationships between a traveler and flights, accommodations, tour guide and destination are established through the booking event. From this booking event, a selection is made for a trip destination, flight, accommodations and tours.

**b. The Three Specializations**

The three different kinds of specializations are displayed in this model. Separate subtype tables were used for the different transportation options related to the booking: flight and car rental. This specialization was favored for this scenario to highlight the two different transportation options available to the traveler. The attributes of each mean of transportation, namely flight and car rental, are different.

The second specialization is the combination of subtype tables and one supertype table. This was displayed in the relationships stemming from the traveler entity, which serves as the supertype table. The subtypes are traveler passport, visa, and traveler insurance. The traveler supertype table stores all the common attributes related to the traveler while each subtype table present unique attributes.

The third specialization is the use of one table with the absence of subtype tables. The accommodation table characterizes this form of specialization. Four of the columns of the accommodation table are labeled “Hotel”, “Motel”, “Resort” and “Airbnb”. Depending on the subtype of accommodation, one of these columns will have a value while the others will be null. All the other attributes in the accommodation table are the same for all types of accommodations.

**c. The Limitation**

* Initially, this model assumed that each traveler made only one booking but could have multiple flights and accommodations. I modified it so that travelers could travel in groups and we wanted to ensure that each traveler was documented in the database and billing information for the group showed even payment distributions rather than duplicated amounts. However, this modified model does not seem to account for infants, specifically for making a flight booking for them.
* This model also does not account for travelers with dual citizenship or those with more than one passport. The database will document passport expiration dates and can replace old passport number for new ones if the traveler gets a new one. However, it will not accept more than one active or unexpired passport number for a given traveler.

**IV. REPORT QUERIES & DESCRIPTIONS**

For all queries:

**USE hinfcom\_gupta;**

1. How many travelers are traveling to Greece? Obtain a list of these travelers:

\*This will help us keep track of travelers traveling to specific destinations. The same query can be applied to different destinations as long as we specify the destination in the “where” condition:

**SELECT \* FROM Traveler t**

**JOIN Traveler\_Group tg ON t.Traveler\_ID= tg.Traveler\_ID**

**JOIN Booking b ON tg.Group\_ID=b.Group\_ID**

**JOIN Flight\_Booking fb ON b.Booking\_ID= fb.Booking\_ID**

**JOIN Flight f ON fb.Flight\_ID= f.Flight\_ID**

**WHERE To\_Destination\_Country= 'Greece';**

1. List the names and ages of all travelers:

**SELECT First\_Name, Last\_Name,**

**FLOOR (datediff(now(), Date\_of\_birth)/365.25) AS Age**

**FROM Traveler**

**ORDER BY First\_Name ASC;**

1. List all accommodations booked after 2020-10-01:

\* This will specifically select bookings made for specific accommodations after a given date:

**SELECT Ac.Name, Ac.CheckIn\_Date FROM Accommodation Ac**

**JOIN Accommodation\_Booking BAc ON BAc.Accommodation\_ID = Ac.Accommodation\_ID**

**JOIN Booking bk ON bk.Booking\_ID = BAc.Booking\_ID**

**WHERE bk.Booking\_Date > '2020-10-01';**

1. Determine the total amount of money collected by the travel company

**\*** This can give an overview for the total amount of money currently available in the company’s balance:

**SELECT sum(Paid\_Amount) FROM Billing**

**JOIN Booking bk ON bk.Booking\_ID = Billing.Booking\_ID;**

1. List the flight number, traveler ID, traveler name and email address of all travelers

\*This will allow the company to obtain information on travelers, their email addresses and flight number. Travelers will be able to receive an email with reminders or any details related to their upcoming flight:

**SELECT f.Flight\_Number, t.Traveler\_ID, t.First\_Name, t.Last\_Name, t.Email FROM Traveler t**

**JOIN Traveler\_Group tg ON t.Traveler\_ID= tg.Traveler\_ID**

**JOIN Booking b ON tg.Group\_ID=b.Group\_ID**

**JOIN Flight\_Booking fb ON b.Booking\_ID= fb.Booking\_ID**

**JOIN Flight f ON fb.Flight\_ID= f.Flight\_ID;**

1. List traveler(s) who have more than one booking:

\*This will help us identify travelers in the database who have made more than one booking:

**SELECT t.Traveler\_ID, t.First\_Name, t. Last\_Name, t.Email FROM Traveler t**

**JOIN Traveler\_Group tg ON t.Traveler\_ID= tg.Traveler\_ID**

**JOIN Booking b ON tg.Group\_ID=b.Group\_ID**

**HAVING count(Booking\_ID)>1;**

1. List traveler(s) and their unpaid balances:

\*This will determine which travelers still have payments that are pending and need to made. This will allow the company to communicate this information to the traveler by sending reminder emails:

**SELECT t.Traveler\_ID, t.First\_Name, t.Last\_Name, t.Email,**

**bill.Total\_Cost-Paid\_Amount AS UnpaidBalance**

**FROM Traveler t**

**JOIN Traveler\_Group tg ON t.Traveler\_ID= tg.Traveler\_ID**

**JOIN Booking bkg ON tg.Group\_ID= bkg.Group\_ID**

**JOIN Billing bill ON bkg.Booking\_ID= bill.Booking\_ID**

**ORDER BY First\_Name ASC;**

1. List traveler names, booking IDs, city they will visit as well as the total cost of their trip

\* This will provide a summary for the traveler’s trip including the billing (total cost of the trip):

**SELECT b.Booking\_ID,t.First\_Name,t.Last\_Name, dest.City,bill.Total\_Cost**

**FROM Booking b**

**JOIN**

**(SELECT d.City, db.Booking\_ID**

**FROM Destination\_Booking db**

**JOIN Destination d**

**ON db.Destination\_ID = d.Destination\_ID) dest**

**ON dest.Booking\_ID = b.Booking\_ID**

**JOIN Traveler\_Group tg ON b.Group\_ID= tg.Group\_ID**

**JOIN Traveler t ON tg.Traveler\_ID=t.Traveler\_ID**

**JOIN Billing bill**

**ON bill.Booking\_ID = b.Booking\_ID;**

1. Determining the number of tour guides booked by the traveler for a particular destination:

**SELECT dest.City, COUNT(tguide.Booking\_ID)**

**FROM Booking b**

**JOIN**

**(SELECT d.City, db.Booking\_ID**

**FROM Destination\_Booking db**

**JOIN Destination d**

**ON db.Destination\_ID = d.Destination\_ID) dest**

**ON dest.Booking\_ID = b.Booking\_ID**

**JOIN Tour\_Guide\_Booking tguide**

**ON tguide.Booking\_ID = b.Booking\_ID**

**WHERE city= 'Tulum';**