Customer (**CustID**, CustName, CustPassportID, CustPhoneNo, CustAddress, InsuranceAvalibility)

Companion (**CompID**, CompPassportID, CompName, CompPhoneNo, CompAddress, **CustID**, InsuranceAvalibility)

BookingRecord (**RecordID**, **CustID**, **TripID**, BookDate)

Trip (**TripID**, StartDate, EndDate, Cost, **InsuranceID**)

OutwardFlight (**FlightID,** **TripID**, FlightNo, DeparturePoint, Destination, TakeOffTime, LandingTime)

ReturnFlight (**FlightID**, **TripID**, FlightNo, DeparturePoint, Destination, TakeOffTime, LandingTime)

HotelInfo (**HotelID**, **TripID**, HotelName, HotelAddress)

CustomerHotelInfo (**CustHotelID**, **RecordID**, RoomNo, CheckInDate, CheckOutDate)

PaymentInfo(**PaymentID**, **RecordID**, PaymentDate, PaymentAmount, PaymentMethod, PaymentStatus)

CustomerService (**StaffID**, StaffName, StaffPhoneNo, **TripID**, ShiftSchedule, Rating)

TourGuide (**StaffID**, StaffName, StaffPhoneNo, **TripID**, LanguagesSpoken, TourCount, CertificationLevel)

Insurance (**InsuranceID**, PolicyNo, Price, CoverageAmount, ExpirationDate, Provider)

**Task 1**

**b) Conceptual Database Design / Model**

**What the database should contain? Show how you form models of the entities and their interrelationships.**

Based on the scenario and our design, only the customers and the travel company staff are users of the travel agency’s database system. We need to have a Customer table and a Companion table to store information such as their names, addresses, phone numbers, passport numbers, and other relevant details.

Since the travel agency offers various trips, the system includes a Trip table to store specific information about each trip. One customer can browse multiple trips, and one trip can have many customer reservations, so the system also requires a Booking table to record the relevant booking details.

Each trip comes with flight and hotel arrangements. Therefore, the system should include a Flight table and Hotel tables. The Hotel table contains general information such as the hotel's name and address, while another table records each customer's specific details, such as their room number.

The travel agency hires staff, such as customer service representatives and tour guides, to provide the best possible service to customers for each trip. Thus, we need the Staff tables.

For managing fees, the system has a Payment table to record all financial transactions. Additionally, the travel agency offers insurance services for each trip, allowing customers to choose whether to purchase insurance. Therefore, we build an Insurance table to store insurance-related details.

The Companion table is used to store information about the companions who travel with the customers, and it is linked to the Customer table through `CustID`, indicating who brought the companions on the trip.

Each trip offered by the travel agency is stored in the Trip table. The `InsuranceID` is used to link the trip with the Insurance table.

The CustomerHoteLInfo table is used to record the specific hotel stay information for each customer, including room numbers, check-in, and check-out dates, and it is linked to the BookingRecord table by `RecordID`.

The booking records for customers are stored in the BookingRecord table, which captures details of the customer's trip reservations. It is linked to the Customer and Trip tables through `CustID` and `TripID`, establishing the relationship between the customer and the trip.

The Insurance table stores detailed information about insurance policies. Customers can choose to purchase insurance for their trips, and the insurance information is linked to each trip by `InsuranceID`.

The travel agency's staff are divided into CustomerService (customer service representatives) and TourGuide (tour guides), with separate tables storing detailed information for each role. Each staff member is linked to the trips they are responsible for via `TripID`, indicating which trip they are assigned to. The CustomerService and TourGuide tables also store additional staff-related information, such as certification levels, languages spoken, tour counts, shift schedules, and ratings.

Relationships:

* One to one:

*BookingRecord : PaymentInfo* -- Each booking corresponds to one payment record.

* One mandatory to many optional:

*Customer : Companion* -- A customer can bring multiple companions, but a companion is optional for each customer.

*Trip : BookingRecord* -- One trip may not have any booking. Sad…

*Trip : Insurance* -- The travel agency offers only one insurance plan for each trip.

* One to many:

*Customer : BookingRecord* -- A customer can make multiple bookings for different trips.

*Trip : FlightInfo* – A trip can involve multiple flights.

*Trip : HotelInfo* -- A trip can involve multiple hotels, as the trip may span various locations.

*BookingRecord : CustomerHotelInfo* -- A booking can involve multiple hotels, and each hotel stay has a room number.

*Trip : Staff* -- The travel agency assigns multiple staff members to each trip(one tour guide and several customer service representatives).

Relationships:

* One to one:

*BookingRecord : PaymentInfo* -- Each booking corresponds to one payment record.

* One to one optional:

*Trip : Insurance* -- The travel agency offers only one insurance plan for each trip.

* One mandatory to many optional:

*Customer : Companion* -- A customer can bring multiple companions, but a companion is optional for each customer.

*Trip : BookingRecord* -- One trip may have many bookings or none at all. Sad…

* One to many:

*Customer : BookingRecord* -- A customer can make multiple bookings for different trips.

*Trip : FlightInfo* – A trip can involve multiple flights.

*Trip : HotelInfo* -- A trip can involve multiple hotels, as the trip may span various locations.

*BookingRecord : CustomerHotelInfo* -- A booking can involve multiple hotels, and each hotel stay has a room number.

*Trip : Staff* -- The travel agency assigns multiple staff members to each trip(one tour guide and several customer service representatives).