In this schema, the diagram illustrates each entity and relationship required by Mrs. Patterson in the letter. I developed three entities – *Customer, Reservation* and *Room* which are connected by the relationships *Makes* and *Contains.*

The *Customer* entity’s identifying attribute is “CID,” which is defined by their place in the reservation queue (so the first customer to ever reserve a room would have a CID of 1, and so on). Following that, the attribute CName contains the customer’s name for identification purposes. Also, as requested, I added “PermAddress” as a composite attribute.

The *Reservation* entity contains the “ResID” attribute as an identifier and “ResStay” as a composite attribute that denotes the customer’s stay (with the values “CheckInDate” and “CheckOutDate”). Then, the derived attribute “DaysIn” subtracts the “CheckOutDate” from “CheckInDate” in order to generate a bill amount for the customer. The “BillAmt” derived attribute computes the product of “DaysIn” and “Room.RPrice” to generate the bill amount. The reason this was made into an associative entity is due to the fact that a reservation requires an ID attribute for itself, and that cannot be represented in a relationship.

The *Room* entity contains the “RoomNum” attribute as an identifier. The composite attribute RoomType contains the attributes RType and RPrice which keep track of the room type and its respective price. Then, the derived attribute “Popularity” shows which room type is the one that is reserved the most.

The *Makes* relationship does not possess any attributes because its respective attributes were placed in the Reservation associative entity, and its constraints are as follows: a customer can make zero to as many room reservations as they please, but a reservation can only be made by one customer only. Same applies to the *Contains* relationship: a room may have zero to many reservations, but a reservation can only apply to one room per instance.

