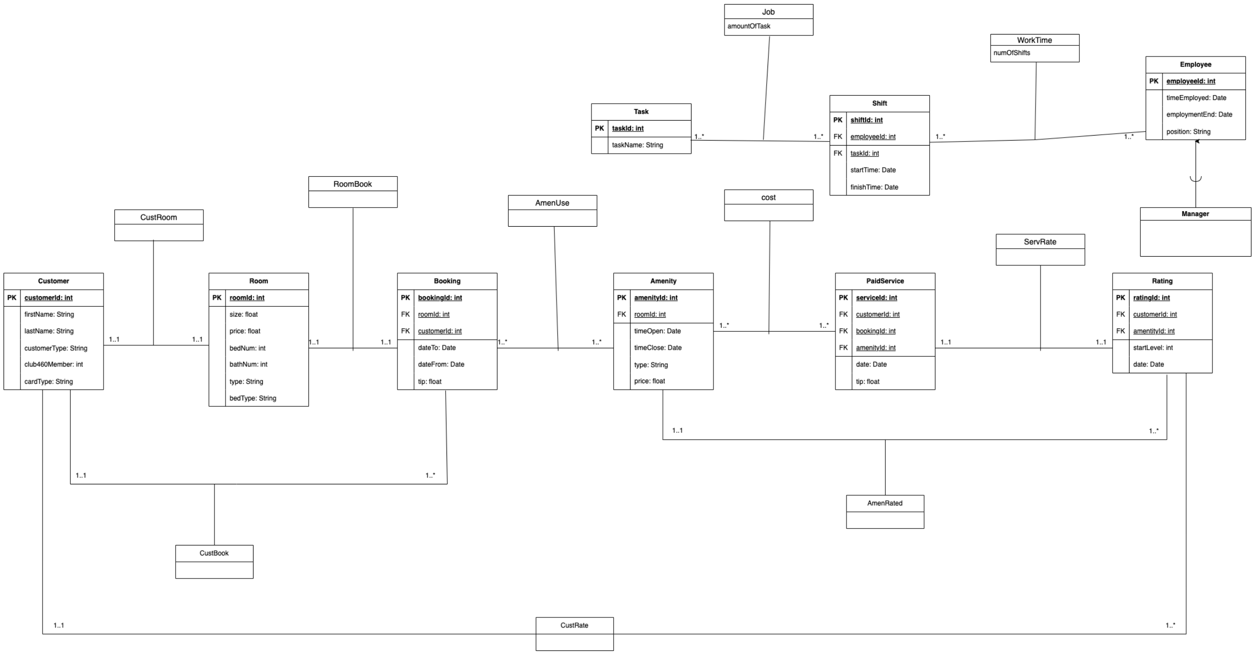
i. Conceptual database design: Your final E–R diagram along with your design rationale and any necessary high–level text description of the data model (e.g., constraints, or anything you were not able to show in the E–R diagram but that is necessary to help people understand your database design).



Manager - Employee is our EER.

ii. Logical database design: The conversion of your E–R schema into a relational database schema. Provide the schemas of the tables resulting from this step.

```
Customer(Customer_Id, First_Name, Last_Name, Customer_Type,
Club460_Member, CardType)
Room(Room_Id, Room_Size, Price, Bed_Num, Bath_Num, Type, Bed_Type)
Booking(Booking_Id, Room_Id, Customer_Id, Date_To, Date_From, Tip)
Amenity(Amenity_Id, Room_Id, Time_Open, Time_Close, Type, Price)
Paid_Service(Service_Id, Customer_Id, Booking_Id, Amenity_Id,
Service_Date, Tip)
Rating(Rating_Id, Customer_Id, Amenity_Id, Star_Level, Rating_Date)
Employee(Employee_Id, Time_Employed, Employment_End, Position)
Shift(Shift_Id, Employee_Id, Task_Id, Shift_Date, Start_Time,
Finish_Time)
Task(Task_Id, Task_Name)
```

iii. Normalization analysis: For each of your entity sets (tables), provide all of the FDs of the table and justify why your table adheres to 3NF / BCNF.

Customer:

- Customer_Id → First_Name
- Customer_Id → Last_Name

- Customer_Id → Customer_Type
- Customer_Id → Club460_Member
- Customer_Id → CardType

Satisfy 3NF because all FDs are non-trivial and Customer_Id is a superkey of Customer.

Room:

- Room_Id → Room_Size
- Room_Id → Price
- Room_Id → Bed_Num
- Room_Id → Bath_Num
- Room_Id → Type
- Room_Id → Bed_Type

Satisfy 3NF because all FDs are non-trivial and Room_Id is a superkey of Room.

Booking:

- Booking_Id → Room_Id
- Booking_Id → Customer_Id
- Booking_Id → Date_To
- Booking_Id → Date_From
- Booking_Id → Tip

Satisfy 3NF because all FDs are non-trivial and Booking_Id is a superkey of Booking.

Amenity:

- Amenity_Id → Room_Id
- Amenity_Id → Time_Open
- Amenity_Id → Time_Close
- Amenity_Id → Type
- Amenity_Id → Price

Paid_Service:

- Service_Id → Customer_Id
- Service_Id → Booking_Id
- Service_Id → Amenity_Id
- Service_Id → Service_Date
- Service_Id → Tip

Satisfy 3NF because all FDs are non-trivial and Service_Id is a superkey of Paid_Service.

Rating:

- Rating_Id → Customer_Id
- Rating_Id → Amenity_Id
- Rating_Id → Star_Level
- Rating_Id → Rating_Date

Satisfy 3NF because all FDs are non-trivial and Rating_Id is a superkey of Rating.

Employee:

- Employee_Id → Time_Emplyed
- Employee_Id → Employment_End
- Employee_Id → Position

Satisfy 3NF because all FDs are non-trivial and Employee_Id is a superkey of Employee.

Shift:

- Shift_Id → Employee_Id
- Shift_Id → Task_Id
- Shift_Id → Shift_Date
- Shift_Id → Start_Time
- Shift_Id → Finish_Time

Satisfy 3NF because all FDs are non-trivial and Shift_Id is a superkey of Shift.

Task:

- Task_Id → Task_Name

Satisfy 3NF because all FDs are non-trivial and Task_Id is a superkey of Task.

iv. Query description: Describe your self–designed query. Specifically, what question is it answering, and what is the utility of including such a query in the system?

The query that we included is to look up all the amenities a given customer never used before. The user will provide the customer’s ID. This would be helpful to populate recommendations for the user to try during their stay.