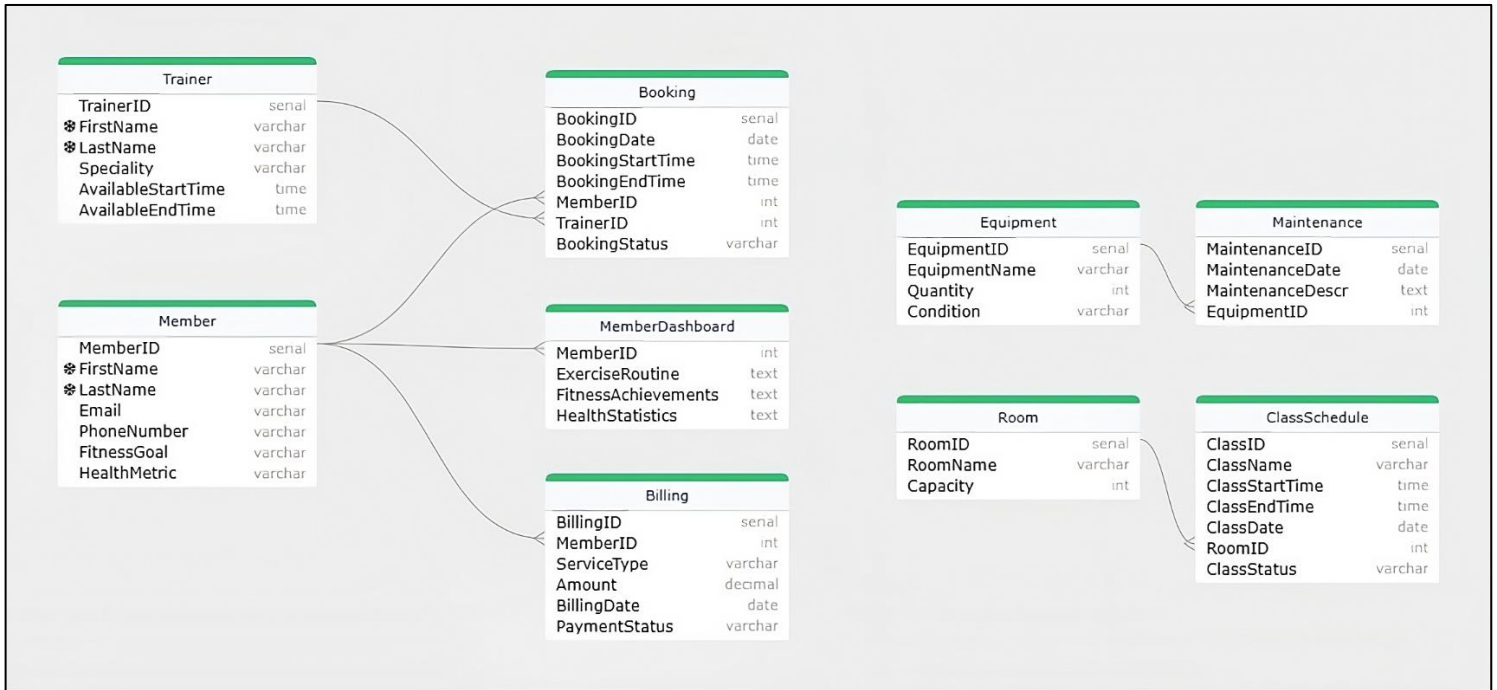
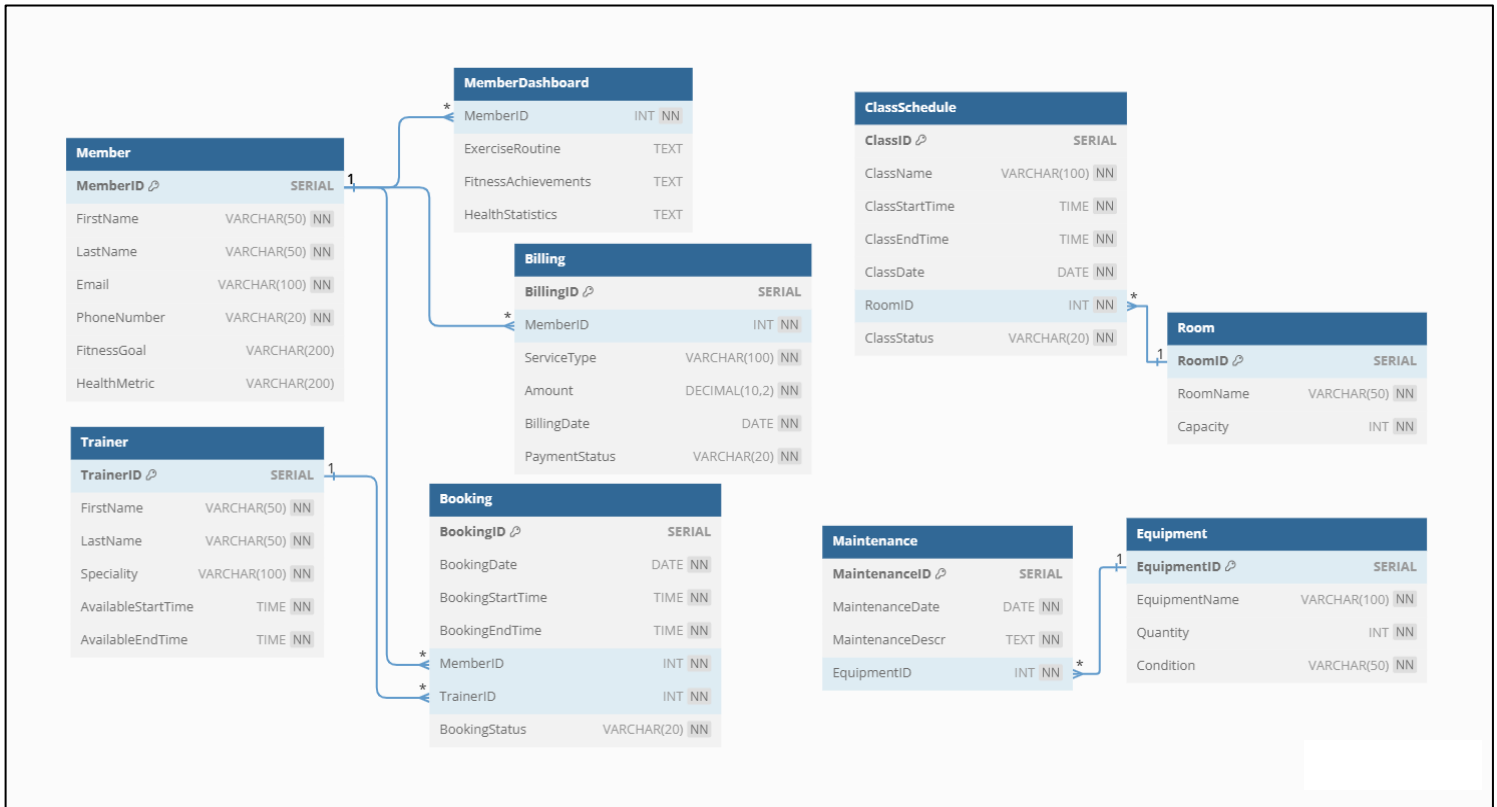


2.1 - Conceptual Design



Assumption Made (Topic)	Assumption Description
Members	A member can have zero or more bookings, but each booking is associated with only one member. This assumption relates to personal training sessions.
Fitness Equipment	I assumed that each maintenance record is associated with only one piece of equipment.
Member Booking for Personal Training Sessions	Bookings have one of three statuses: Scheduled, Rescheduled, or Cancelled.
Class Booking Status	Class schedules can have one of two statuses: Scheduled or Cancelled.
Billing	Each billing record is associated with only one member, meaning that your bill comes in one required payment.
Fitness Goals and Health Metrics	Members can set their own fitness goals and health metrics, there is no specific guideline or required information here.

2.2 - Reduction to Relation Schemas



Strong Entity Sets: Member, Trainer, Billing, Room, Equipment, Maintenance

Weak Entity Sets: Booking, ClassSchedule, MemberDashboard

Relationship Sets:

There is a relationship set between the Member and Trainer entities. Members book training sessions, and Trainers conduct booked training sessions, so both entities have a relationship there. This is a many-to-many relationship, as a member can book a session with different trainers, and a trainer can have multiple bookings from different members.

There is a relationship set between the Room and ClassSchedule entities. Class Schedules contain classes, and Rooms host the classes based on Class Schedules, so both entities have a relationship there. This is a many-to-one relationship, as a room can host multiple classes from the Class Schedules, but each class in the Class Schedule is only associated with one room.

There is a relationship set between the Member and MemberDashboard entities. Members create a Member Dashboard, and each MemberDashboard represents the personal dashboard information for each member, so both entities have a relationship there. This is a one-to-one relationship, as each member has exactly one member dashboard.

2.3 - DDL File, 2.4 - DML File, 2.5 - Implementation

All three of these parts are in my GitHub Repository. I used Python for the implementation, and it runs using a command-line interface.

2.6 - Bonus Features, 2.7 - GitHub Repository

The Bonus Features here is unapplicable here because I used a command-line interface is the basic answer without bonus marks (according to marking scheme).

My GitHub Repository is: <https://github.com/double-o-3/COMP-3005-Final-Project>