**2 Project Report**

**Database Conceptual Design for Health and Fitness Club Management System**

**Introduction**

The Health and Fitness Club Management System is an application intended to streamline the operations of a health club by integrating member services, trainer schedules, and administrative functionalities. It forms a nexus for members to establish fitness goals, monitor progress, and manage schedules, for trainers to oversee their availability and client profiles, and for administrative staff to handle logistics and financial transactions.

**Conceptual Design**

The system's conceptual design is underpinned by a relational database that encompasses entities and relationships fundamental to the club's operations. This design is visually represented in an Entity-Relationship Diagram (ERD) and articulated in the accompanying Data Definition Language (DDL) file.

Entities and Their Relationships The entities encapsulated within the database schema reflect the real-world constituents of the club:

1. **Members:** This entity represents the club members, with attributes capturing personal information, credentials, and an administrative flag denoting whether a member is part of the staff. A one-to-many relationship is established between Members and FitnessGoals, HealthMetrics, FitnessAchievements, ClassRegistrations, and Billing, indicating that one member can have multiple entries in these entities.
2. **Trainers:** Trainers are also considered Members, evident from the foreign key reference to the Members table. Trainers can conduct multiple Classes and perform checks on Equipment, indicated by the foreign key relationships from Classes and Equipment to Trainers.
3. **Classes:** This entity signifies the classes offered by the club, capturing details such as the class name, schedule, and the boolean groupsession attribute, which differentiates between personal and group sessions. Classes are linked to Trainers and have registrations managed through the ClassRegistrations entity.
4. **FitnessGoals:** It captures the fitness aspirations of members, allowing the database to store varied goals along with their current and target values. This dynamic structure supports the tracking of different types of fitness goals.
5. **HealthMetrics:** This entity is designed to track various health metrics for members over time, providing a historical view of a member's health data.
6. **ClassRegistrations:** It maintains a log of members' registrations for classes, facilitating the management of class attendance and member engagement in club activities.
7. **Equipment:** The entity records the details of the equipment available at the club, alongside maintenance schedules.
8. **Billing:** It is integral to the financial aspect of the system, linking members to their financial transactions, including membership fees and payments for classes and personal training sessions.

**Cardinality and Participation Types**

* **Member-FitnessGoals:** Members can set multiple fitness goals, suggesting a one-to-many relationship between Members and FitnessGoals.
* **Member-HealthMetrics:** Members can record multiple health metrics, indicating a one-to-many relationship between Members and HealthMetrics.
* **Members to ClassRegistrations:** One-to-Many, as members can register for multiple classes.
* **Members to Billing:** One-to-Many, since members can have multiple billing records.
* **Trainers to Classes:** One-to-Many, indicating a trainer can conduct several classes.
* **Trainers to Equipment:** One-to-Many, as trainers are responsible for multiple pieces of equipment.
* **Member-ClassRegistrations:** Members can register for multiple classes, indicating a one-to-many relationship between Members and ClassRegistrations.
* **Class-ClassRegistrations**: Each class can have multiple registrants, indicating a one-to-many relationship between Classes and ClassRegistrations.
* **Trainer to Member:** A trainer can be a member, suggesting a one-to-one relationship between the Trainer and Member entities, but it's optional (zero or one to one).

**Assumptions the database schema assumes the following:**

* Each member has a unique identifier, and members can also be trainers or administrative staff.
* Each member can have multiple fitness goals and health metrics.
* Fitness goals are diverse and include various metrics for assessment.
* Health metrics are tracked over time for each member.
* Trainers manage their schedules and are tied to the classes they conduct.
* Group sessions and individual sessions are differentiated within the classes offered.
* Financial transactions are logged and associated with individual members.
* Equipment in a room is checked by a trainer following their class, so equipment maintenance logs are generated following each class.
* Billing is handled via the information in the database by some external party.

**Entities and their attributes:**

**Members**

* MemberID (PK)
* FirstName
* LastName
* Email
* Password
* DateOfBirth
* IsAdmin

**FitnessGoals**

* GoalID (PK)
* MemberID (FK)
* GoalType
* ExerciseType (optional)
* CurrentStrengthWeight (optional for strength goals)
* GoalStrengthWeight (optional for strength goals)
* CurrentWeight (optional for weight goals)
* DesiredWeight (optional for weight goals)

**HealthMetrics**

* MetricID (PK)
* MemberID (FK)
* MetricType
* MetricValue
* DateRecorded

**Trainers**

* TrainerID (PK)
* FirstName
* LastName
* Specialization
* MemberID (FK, nullable if not all trainers are members)

**Classes**

* ClassID (PK)
* ClassName
* RoomID
* TrainerID (FK)
* DateTime
* GroupSession (indicates whether the class is a group session or not)

**ClassRegistrations** (Associative Entity for many-to-many relationship)

* RegistrationID (PK)
* MemberID (FK)
* ClassID (FK)

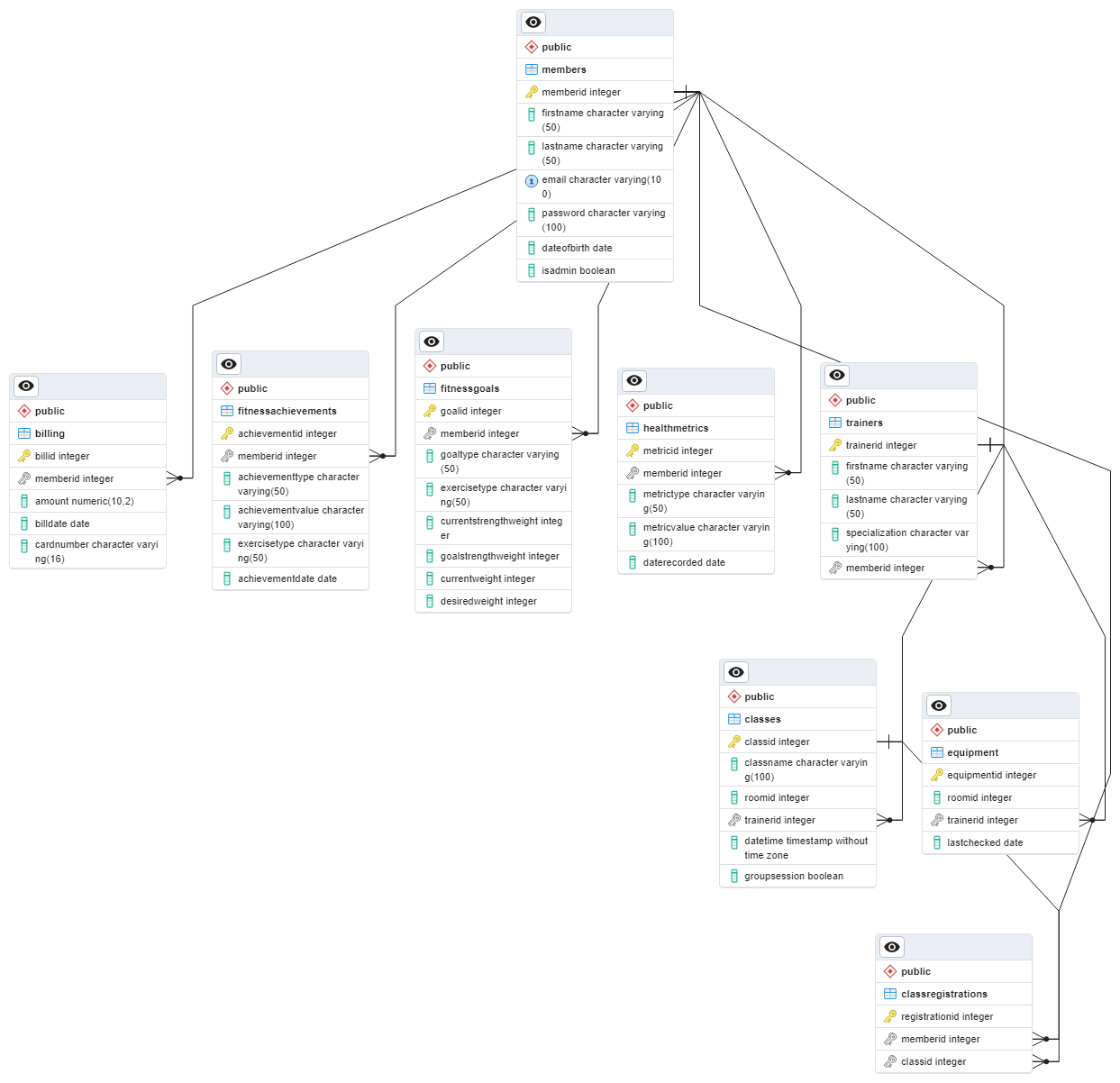
**Equipment**

* EquipmentID (PK)
* RoomID
* TrainerID (FK)
* LastChecked

**Billing**

* BillID (PK)
* MemberID (FK)
* Amount
* BillDate
* CardNumber (optional, assuming null if payment not made by card)

**DDL and DML files can be found in the SQL directory.**

**Entity-Relationship Diagram:**

**Requirements Checklist (Missing a feature? Check here):**  
Member Functions:  
1. User Registration – **Check**, users cannot login until they register and users who have registered can login with their user info.

2. Profile Management (Updating personal information, fitness goals, health metrics) – **Check**, personal information (name, birthday, etc), fitness goals (strength, or weight), health metrics (cholesterol, BMI, and blood pressure) can all be set and updated either from the menus or directly from the dynamic dashboard.

3. Dashboard Display (Displaying exercise routines, fitness achievements, health statistics), **Check,** There are 3 panels on the dashboard corresponding to each requirement. The dashboard includes bonus features, such as dynamic editing of profile management values and dynamic fitness achievements which are automatically committed to the database.

4. Schedule Management (Scheduling personal training sessions or group fitness classes. The system must ensure that the trainer is available), **Check,** members can register, drop, and reschedule their classes and attendance with a personal trainer based on what sessions appear on the schedule. Sessions will be marked group or not group, and users can decide which to register for accordingly. The schedules dynamic UI I implemented as a bonus feature.

Trainer Functions:  
1. Schedule Management (Trainer can set the time for which they are available.) **Check,** trainers can add and remove classes which are dynamically displayed on the calendar giving trainers complete control over managing their schedule and what classes appear.

2. Member Profile Viewing (Search by Member’s name) **Check,** trainers can use the member lookup button to view a member’s dashboard which displays their fitness, health, and achievement information.

Administrative Staff Functions:  
1. Room Booking Management, **Check**, admins have access to the admin panel which allows them to view room bookings via room ID and displays the time and date the bookings take place. They can also choose to manage these room bookings by cancelling a booking for a trainer that isn’t them.

2. Equipment Maintenance Monitoring, **Check,** every time a class occurs in a specific roomID, the trainer running the class is assumed to do an equipment maintenance check. Therefore, the system logs each check following a trainer’s class by identifying which trainer did the check, when it was, and what room had its equipment checked. All of this information is displayed on the admin panel for admins to monitor equipment status. This general approach allows the system to be applied to any fitness club and avoids being specific to one fitness club’s set of equipment which would have to be hard implemented into the database.

3. Class Schedule Updating**, Check,** Admins can alter class schedules by cancelling classes in roomIDs that are booked or being used for administrative purposes and these changes are reflected dynamically via the schedule. Otherwise, the schedule is self-updating.

4. Billing and Payment Processing (Your system should assume integration with a payment service  
[Note: Do not actually integrate with a payment service], **Check,** whenever a member registers for a class they are recorded in the billings table which could be processed by a payment service trivially.