

COMP 3005 Final Project

Winter 2024

Health and Fitness Club Management System

By

Tomas Teixeira (101244889)

April 10th, 2024

Conceptual Design

For the ER-diagram, I represented all people as either a Member, a Trainer, or an Admin entity. The Admin entity is not part of any relationships, however it is important for storing information about any non-trainer staff that works at the gym, as well as their email and password for logging in to the system.

The Member entity is part of relationships with HealthMetric, Membership, FitnessGoal, and Routine. These are all one-to-many relationships where each of the entities listed can only have one member linked to it but any given member may be linked to multiple of the listed entities. This allows the member to have a history of their health measurements and achievements in the database as well as store all their routines and any memberships they are currently subscribed to.

Next, the Member entity is also part of relationships with PersonalSession and GroupSession entities. As expected, the relationship with PersonalSession is one-to-many where a PersonalSession may only be linked to one member but a member can be registered to many PersonalSessions, while the GroupSession is many-to-many as, once again, a member can be registered to many GroupSessions but this time the GroupSessions contains multiple members. Additionally, both GroupSessions and PersonalSessions have a many-to-one relationship with Trainers. An assumption is made here that each training session may only have one trainer leading it, but trainers can be signed up to lead multiple sessions.

Regarding the previous paragraph, this also doubles as a scheduling system for trainers. The way this works is trainers will create PersonalSessions and GroupSessions in line with their availability. This will then allow members to view all sessions with space and allow them to then register for the session.

Finally, PersonalSession and GroupSession entities are part of a many-to-one relationship with Room entities, as each session can only take place in a single room, but rooms can host many sessions throughout the week. Furthermore, these rooms will have a many-to-one relationship with Equipment, allowing trainers and admins to track maintenance/cleanliness of rooms and equipment as well as availabilities of specific rooms.

I've included a PNG of the ER-Diagram on the next page (Figure 1) to give a visual representation of the database as well as allow you to view the properties of the entities I did not discuss in this section.

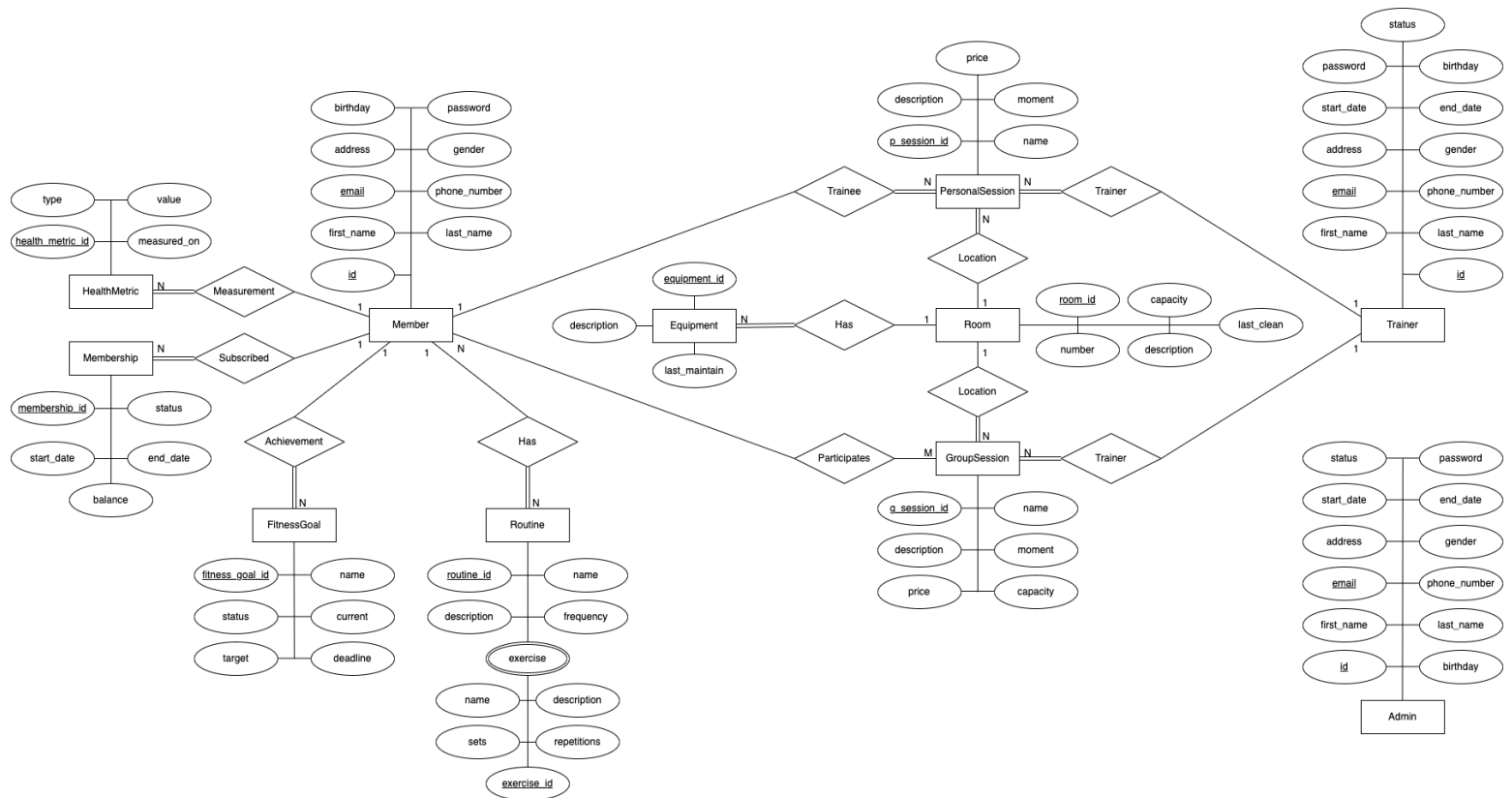


Figure 1: ER-Diagram

Reduction to Relation Schemas



Figure 2: Relation Schemas

DDL File

The data definition language statements is included in a file in the GitHub repo called “DDL.sql” under the “SQL” directory.

DML File

The data manipulation language statements is included in a file in the GitHub repo called “DML.sql” under the “SQL” directory.

Implementation

The Health and Fitness Club Management System program I've implemented takes the form of a command-line interface program written in JAVA. All source code is included under the "Program" directory in the GitHub repository.

The database I used for this program was PostgreSQL. The database name was "HealthFitnessClub". In order to update the login conditions when you go to run it, you must modify the "URL", "USER", and "PASSWORD" variables stored under the "Database.java" class.

To start, there is a "Database" class which serves as the path to the database for all other classes in the program. It is a singleton class which connects to the database on startup and allows all other classes to access the database through the public "getConnection" method.

Next, all entities in the database have their own class (Member, Trainer, FitnessGoal, etc.) that store the properties of the entity in variables. Each class additionally has behaviour that allows the class to not only give information about itself, but also modify/delete itself from the database (deleteFromDB method, for example).

Lastly, each class has an associated tracker class (except for Member, Trainer, and Admin). These classes store all instances of a specific class (FitnessGoalTracker stores all FitnessGoals) associated to the user currently logged in. Since, this is a command-line program, these tracker classes also serve as the control flow of the program, prompting users to make decisions on what they want to do. Access to the trackers depends on who the user is. For example, members are given access to RoutineTracker, SessionTracker, FitnessGoalTracker, and MetricTracker while trainers are given access only to MemberViewer and SessionTracker.

Github Repository

The link for the GitHub repository is: <https://github.com/SunflowerEdition/COMP3005FINAL>

The link for the video presentation is located on the GitHub readme.