Carleton University
COMP3005 – B

Final Project v2.0:
Health and Fitness Club Management System (30%)

John Tronciu
101193240

Due: April 13th, 2024

Repository Link

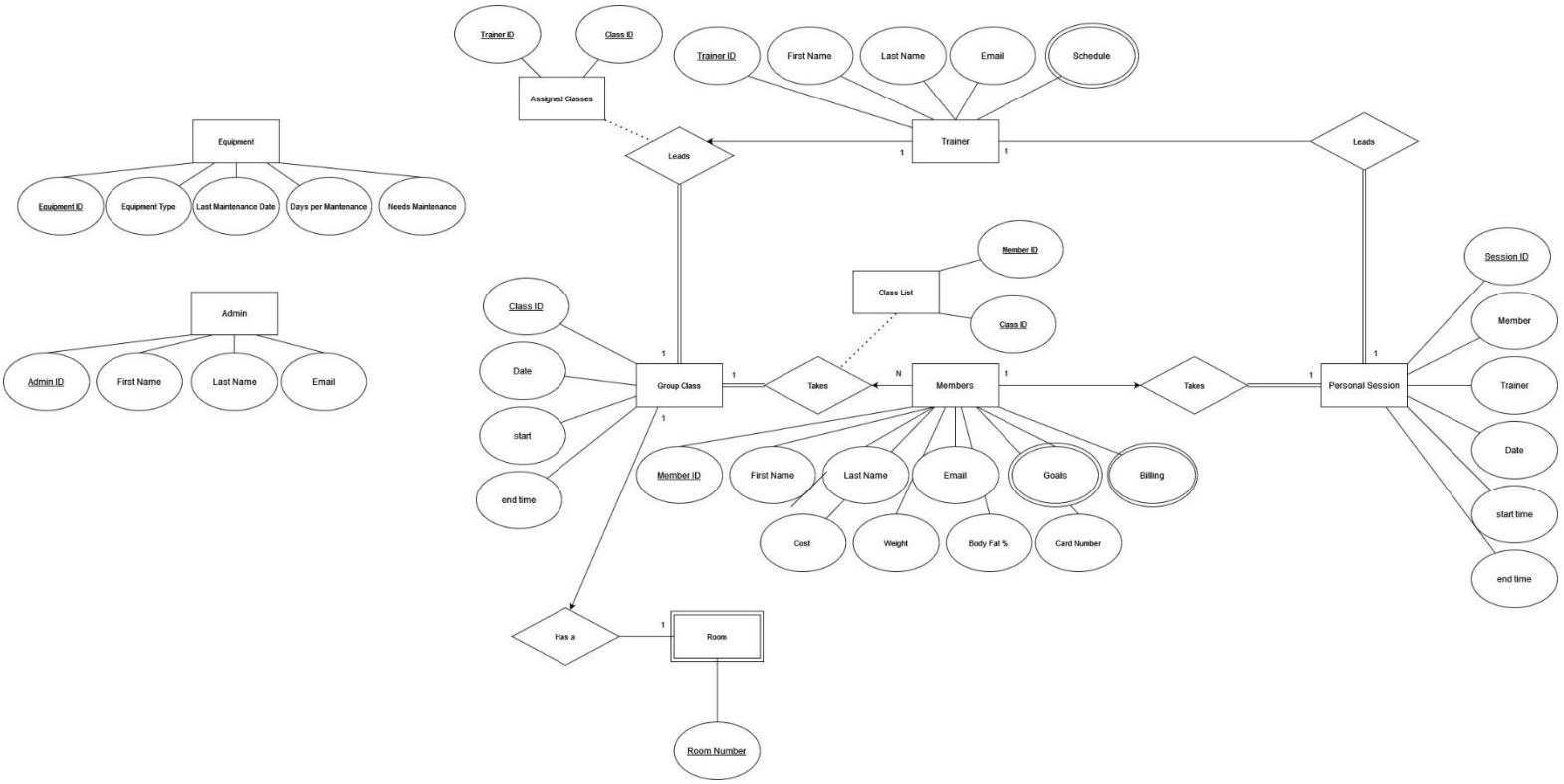
https://github.com/johnctronciu/W2024_3005_Project_v2

Conecptual Design

The conceptual design of the database is shown via. an ER diagram, the following assumptions were made about the project specifications to form the design:

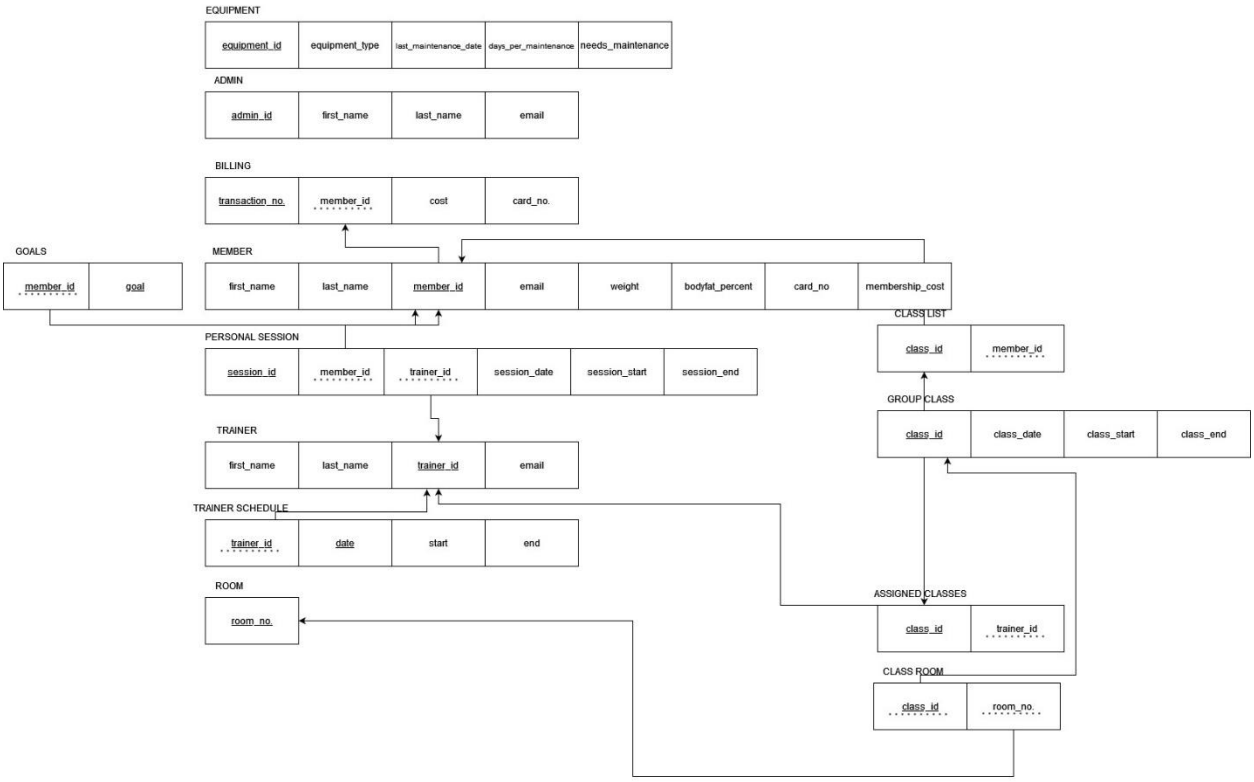
Requirement	Assumption	Representation in ER Model
Members can schedule, reschedule, or cancel personal training sessions	Personal training sessions will be in open weight room i.e., a public section of the gym but is one-on-one time with a trainer.	There is no room attribute for personal training sessions
Members should be able to register for group fitness classes	Members can register for group classes which contain a room and lead trainer all centrally managed by admins	Group class table related to member and trainer tables since there are intermediate tables to indicate which trainers and admins take the class, however admins are not connected to any table directly as they do not have any related data and simply oversee the operations.
Trainers should have the ability to manage their schedules and view member profiles.	Trainers have the prior abilities but are treated as contractors who simply set their availability and then are selected by admins to lead classes or by members for personal sessions. Trainers do not personally have the ability to schedule any event themselves.	Trainers do not have attributes or relationships to classes or sessions that suggest they have control of booking.
No mention of payment for admins or trainers	Admins and trainers are either volunteers or paid by a supervising database as admins have full access to the database and if handled in this database would be able to control their pay	No payroll attributes.
Administrative Staff should be equipped with features to manage room bookings	Only group classes use rooms and therefore the booking system is simply used to assign a room to a class ID.	“Has a” relationship between group class and room.
	Administrators don’t have specified responsibilities in	No direct relationship from admin to other classes or vice

	terms of specific rooms, classes etc. all administrators working are responsible for any work that comes up.	versa as there is no need to map class to specified admins as all admins are responsible for all classes etc.
Members have goals and bills to pay for membership	Goals and billing tables are made are made but cannot be defined by themselves	Marked as attributes with dependent relationship to owners
Group fitness classes	Classes can be planned without trainers assigned at the moment of assignment but need one to be conducted / exist	Assigned classes table that has trainer ID(s) and the class ID(s) they are responsible for.
Trainer can set the time for which they are available.	Trainers have multiple days they work, and, in a schedule, table would have repeated trainer IDs. Therefore, trainers can't have two shifts in the same day	Trainer schedule is a dependent relationship, and its primary key is trainer ID and date.
Equipment	Equipment is not assigned to a specific room or trainer or member and instead may be moved at any time and used by any person without documentation.	Equipment does not have any direct relation to other tables.
Personal session	Since trainers cannot schedule anything on their own the member schedules the session and picks the trainer therefore a personal session cannot exist without a trainer or member	Doubled lines to represent total participation.



(ER diagram, also in GitHub repository)

Reduction to Relation Schemas



(Relation Schema, also in GitHub repository)

Each title represent a unique table in the relation database. Primary keys are underlined, foreign keys are underlined with a dotted line and point to the value they reference.

DDL File

The DDL file is contained in the GitHub repository as outlined in the specifications.

DML File

The DML file is contained in the GitHub repository as outlined in the specifications.

Implementation

This project was implemented using a relational database, namely PostgreSQL, using pgadmin 4 to setup the database and run the DML and DDL files. The script to connect to the database and run the various functions requited by the project outline was written using Python, namely the extension psycopg2. Code uploaded to GitHub with video demo link going thorough the implementation and functionality.