COMP3005 Final Project Report
Health and Fitness App

By: Waleed Gabr and Mohammad Saad

Introduction:

The health and fitness club app is a ruled based data-driven application to design the support and support the operations of a modern Fitness Center the system supports three user rules;

- Members: profile management, fitness tracking and registrations
- Trainers: ability to define availability and look up member data
- Administrators: manage overall operations such as management of classes and rooms

The system uses PostgreSQL as its backend database and python as its application layer. The library used to connect the python application to the SQL Server is psycopg2. The application enforces data consistency, normalization, constraints such as trainer and availability validation while also preventing conflicts for both personal training sessions and scheduling of classes.

Conceptual Assumptions for Entities:

1. Trainers or members log in using their email and password only not a username or any other personal information
2. Every personal training session must occur within an existing trainer availability time range that is not booked
3. Trainer availability may not overlap with other availability blocks created for the same trainer
4. Class capacity must be respected and will be enforced by a database trigger
5. Only administrators have the ability to create and assign classes and rooms
6. Rooms may be reused across different classes as long as the schedules do not overlap
7. A new table will be created to represent the M:N relationship found between members and classes

Entities:

There are a total of eight entities that exist within the database.

1. Admin
2. Member
3. Trainer
4. Room
5. HealthMetric
6. TrainerAvailability
7. PersonalTrainingSession
8. Class

Relationships, Explanation and Assumptions:

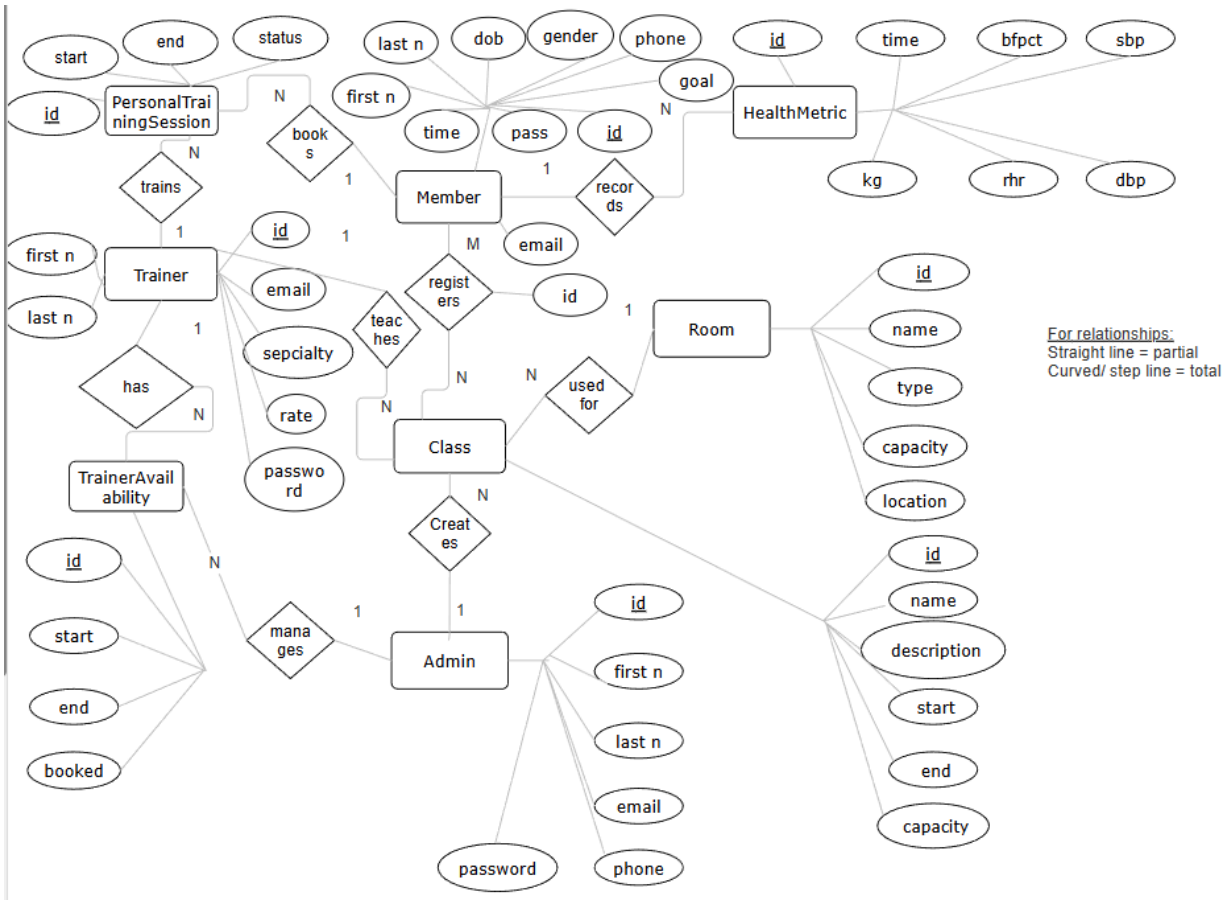

Figure 1: ER diagram for reference showing the different entities and their corresponding relationships

Relationship	**Name in ERD**	**Cardinality**	**Participation**	**Explanation/Assumption**
Member to HealthMetric	Records	1:N	Member: partial HealthMetric: total	A member may or may not have health metrics but every Health metric record must belong to exactly one member
Member to PersonalTrainingSession	Books	1:N	Member: Partial PTS: Total	A member May schedule a PT

				session but a PT session must have a member involved
Trainer to PTS	Trains	1:N	Trainer: Partial PTS: Total	A trainer may have a personal training session coming up but a personal training session must have a trainer teaching it
Trainer to TrainerAvailability	Has	1:N	Trainer: Partial TA: Total	A trainer must have created an availability block but an availability cannot exist without the trainer
Admin to TA	Manages	1:N	Partial on both	An admin can optionally create availabilities but not every availability needs to have an admin associated with it because trainers can manage their own availability if they want to
Trainer to Class	Teaches	1:N	Trainer: Partial Class: Total	A trainer May teach many classes but every class must be assigned to a trainer.
Member to Class	Registers	M:N	Member: Total Class: Total	Members can register for many classes and classes have

				many members . ClassRegistration implemented as table in DDL
Room to Class	Used for	1:N	Room: Partial Class: Total	A room can be used for many classes but every class must be assigned to exactly one room and a room cannot exist without hosting any classes

Normalization:

All relations are in 3NF:

Example: Class

PK: class_id
All non-key attributes depend on only PK. Therefore, no partial dependencies
No attribute depends on a monkey attribute. Therefore, no transitive dependencies

Example for relationship: ClassRegistration

Composite PK: (member_id, class_id)
No other non-key attributes

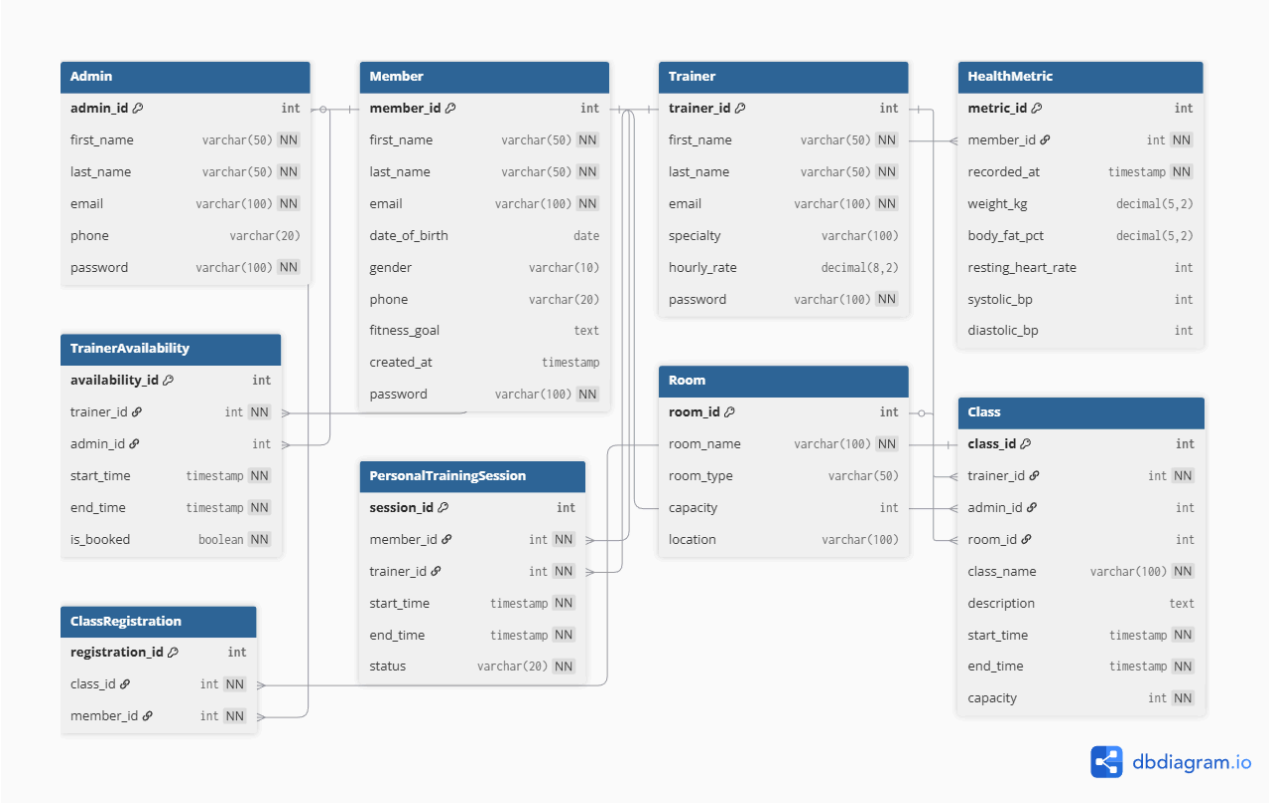

Figure 2: Resulting schema after consideration of assumptions and normalization