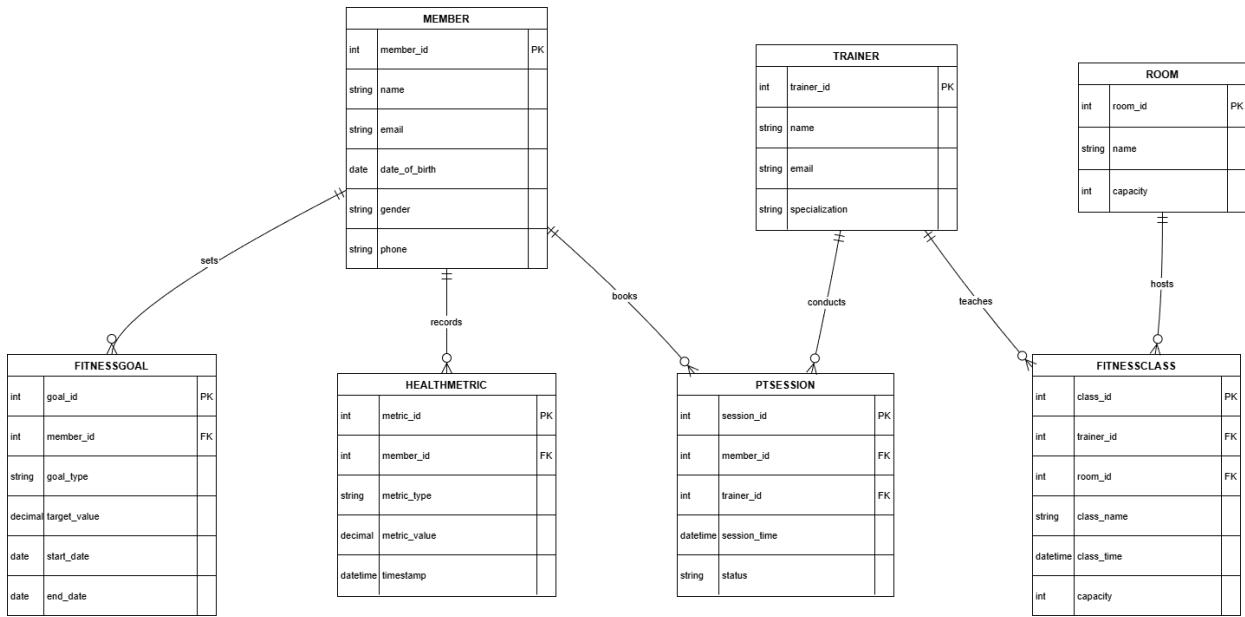


## 1. ER diagram



## 2. Mapping

Table: **Member**

Columns:

- member\_id – Primary key
- name
- email – unique
- date\_of\_birth
- gender
- phone

Represents someone registered at the club.

Table: **Trainer**

Columns:

- trainer\_id – Primary key
- name

- email -- unique
- Specialization

Represents a trainer who can teach classes or do personal training.

#### Table: **Admin**

Columns:

- admin\_id – Primary key
- username – unique
- password\_hash
- name
- email – unique

Represents a admin

#### Table: **FitnessGoal**

Columns:

- goal\_id – Primary key
- member\_id – Foreign key -> Member.member\_id
- goal\_type
- target\_value
- start\_date
- end\_date

Represents a specific goal for a member like losing weight or lowering body fat. One member can have multiple goals.

#### Table: **HealthMetric**

Columns:

- metric\_id – Primary key
- member\_id – foreign key -> Member.member\_id
- metric\_type
- metric\_value
- timestamp

Represents measurement recorded for a member at a certain time.

#### Table: **Room**

Columns:

- room\_id – primary key
- name
- Capacity -- must have capacity > 0

Represents a physical room in the club.

Table: **Class**

Columns:

- class\_id – primary key
- trainer\_id – foreign keys -> Trainer.trainer\_id
- class\_name
- class\_time
- Capacity – must have capacity > 0

Represents a scheduled group class taught by one trainer in one room.

Table: **PTSession**

Columns:

- session\_id – Primary key
- member\_id – foreign keys -> Member.member\_id
- trainer\_id – foreign key -> Trainer.trainer\_id
- session\_time
- status

Represents a one on one training session between a member and a trainer.

## 2.2 Relationship Implementation/Constraints

The 1 to many relationships from the ER diagram are implemented using foreign keys on the “many” side

FitnessGoal.member\_id -> Member.member\_id

HealthMetric.member\_id -> Member.member\_id

PTSession.member\_id -> Member.member\_id

PTSession.trainer\_id -> Trainer.trainer\_id

Class.trainer\_id -> Trainer.trainer\_id

Class.room\_id -> Room.room\_id

### 3. Normalization

All tables are designed to meet at least Third Normal Form. Keys are synthetic, every column is atomic, and every non key attribute depends only on the primary key.

Since all primary keys are single columns, 2NF is automatically satisfied. The main thing is checking for transitive dependencies.

#### Normal form checks by table

##### Member

- Primary key: member\_id
- Everything depends on member\_id
- Email is also unique but does not create transitive dependencies
- Member is in 3NF.

##### Trainer

- Primary key: trainer\_id
- All attributes depend on trainer\_id and no transitive dependencies
- Trainer is in 3NF

##### FitnessGoal

- Primary key: goal\_id
- Everything depends on goal\_id
- No transitive dependencies
- 3NF

##### HealthMetric

- Primary key: metric\_id
- Everything depends on metric\_id
- No transitive dependencies
- 3NF

### **Room**

- Primary key: room\_id
- Everything depends on room\_id.
- Name is unique but does not break 3NF
- 3NF

### **Class**

- Primary key, class\_id
- Everything depends on class\_id
- The unique pair room\_id and class\_time is just a business rule and does not affect 3NF
- 3NF

### **PTSession**

- Primary key: session\_id
- Everything depends on session\_id
- The unique constraint on trainer\_id and session\_time does not create a transitive dependency
- 3NF

### **Summary of normalization**

All tables are in 1NF, 2NF, and 3NF. No repeating groups no partial dependencies and no transitive dependencies. The schema meets the normalization requirements for the project.