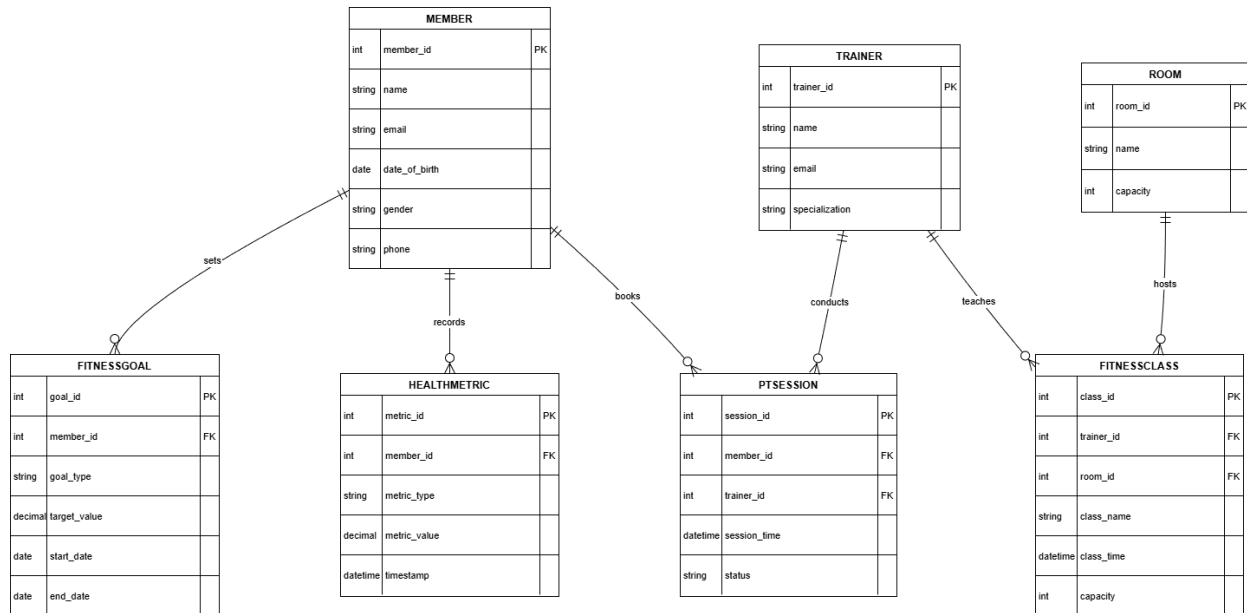


1. ER diagram



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.