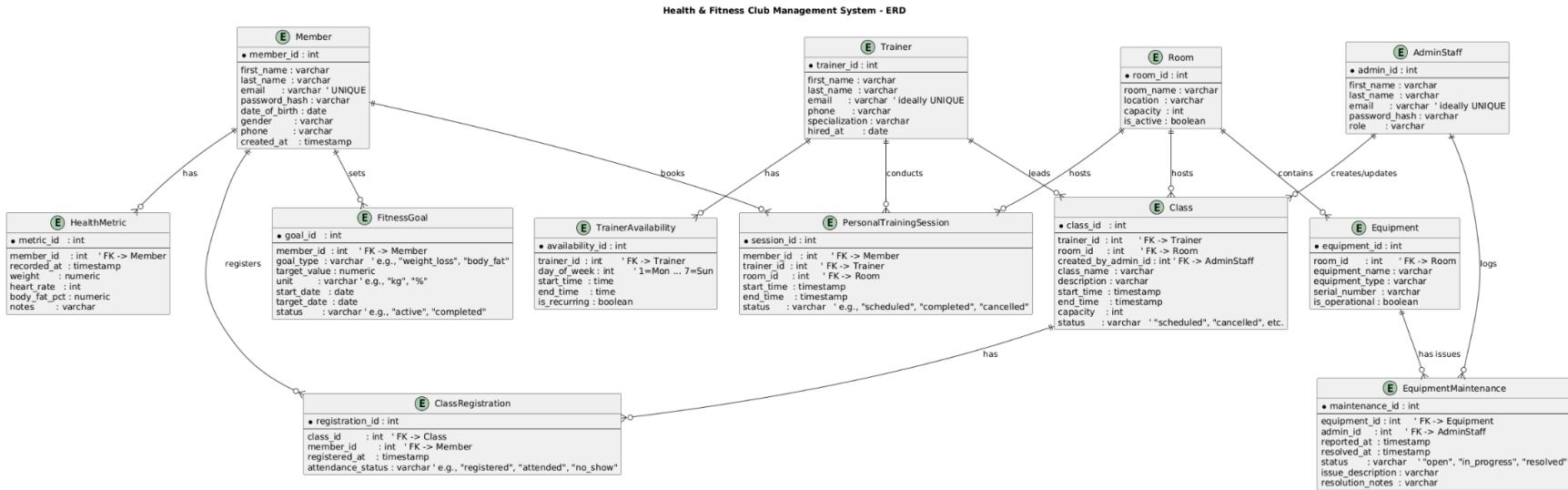


1. ERD Diagram



See the picture more clearly (allows for zoom): [Here](#)

2. Relational Schema

2.1 Member

```
Member(  
    member_id      INT PRIMARY KEY,  
    first_name     VARCHAR NOT NULL,  
    last_name      VARCHAR NOT NULL,  
    email          VARCHAR NOT NULL UNIQUE,  
    password_hash  VARCHAR NOT NULL,  
    date_of_birth  DATE,  
    gender         VARCHAR,  
    phone          VARCHAR,  
    created_at     TIMESTAMP NOT NULL  
)
```

- Keys: member_id PK, email UNIQUE
- No redundancy: personal info is here only.

2.2 Trainer

```
Trainer(  
    trainer_id    INT PRIMARY KEY,  
    first_name    VARCHAR NOT NULL,  
    last_name     VARCHAR NOT NULL,  
    email         VARCHAR NOT NULL UNIQUE,  
    phone         VARCHAR,  
    specialization VARCHAR,  
    hired_at      DATE  
)
```

- Separate from Member & AdminStaff because roles are conceptually different so no weak entity here.

2.3 AdminStaff

```
AdminStaff(  
    admin_id      INT PRIMARY KEY,  
    first_name    VARCHAR NOT NULL,  
    last_name     VARCHAR NOT NULL,  
    email         VARCHAR NOT NULL UNIQUE,  
    password_hash VARCHAR NOT NULL,  
    role          VARCHAR  
)
```

2.4 Room

```
Room(  
    room_id    INT PRIMARY KEY,  
    room_name  VARCHAR NOT NULL,  
    location   VARCHAR,  
    capacity   INT NOT NULL,  
    is_active  BOOLEAN NOT NULL  
)
```

- Rooms are referenced by Class, PersonalTrainingSession, Equipment.

2.5 Fitness Goal

```
FitnessGoal(  
    goal_id      INT PRIMARY KEY,  
    member_id    INT NOT NULL,  
    goal_type    VARCHAR NOT NULL,          (weight_loss, body_fat)  
    target_value NUMERIC NOT NULL,  
    unit         VARCHAR NOT NULL,          (kg, %)  
    start_date   DATE NOT NULL,  
    target_date  DATE,  
    status       VARCHAR NOT NULL,          (active, completed)  
CONSTRAINT fk_fitnessgoal_member  
    FOREIGN KEY (member_id) REFERENCES Member(member_id)  
    ON DELETE CASCADE  
)
```

- Relationship: 1 Member : many Goals
- Participation: member_id NOT NULL so each goal must belong to a member (total participation of FitnessGoal in Member).

2.6 HealthMetric

```
HealthMetric(  
    metric_id    INT PRIMARY KEY,  
    member_id    INT NOT NULL,  
    recorded_at  TIMESTAMP NOT NULL,  
    weight       NUMERIC,  
    heart_rate   INT,  
    body_fat_pct NUMERIC,  
    notes        VARCHAR,  
    CONSTRAINT fk_healthmetric_member  
        FOREIGN KEY (member_id) REFERENCES Member(member_id)  
        ON DELETE CASCADE  
)
```

- Tracks history, not overwriting: many rows per member, different recorded_at.

2.7 PersonalTrainingSession

```
PersonalTrainingSession(  
    session_id  INT PRIMARY KEY,  
    member_id   INT NOT NULL,  
    trainer_id  INT NOT NULL,  
    room_id     INT NOT NULL,  
    start_time  TIMESTAMP NOT NULL,  
    end_time    TIMESTAMP NOT NULL,  
    status      VARCHAR NOT NULL,          (scheduled, completed, cancelled)  
    CONSTRAINT fk_ptsession_member  
        FOREIGN KEY (member_id) REFERENCES Member(member_id),  
    CONSTRAINT fk_ptsession_trainer  
        FOREIGN KEY (trainer_id) REFERENCES Trainer(trainer_id),  
    CONSTRAINT fk_ptsession_room  
        FOREIGN KEY (room_id) REFERENCES Room(room_id)  
)
```

- Represents booking between one member, one trainer, one room.

2.8 Class

```
Class(
    class_id      INT PRIMARY KEY,
    trainer_id    INT NOT NULL,
    room_id       INT NOT NULL,
    created_by_admin_id INT NOT NULL,
    class_name    VARCHAR NOT NULL,
    description   VARCHAR,
    start_time    TIMESTAMP NOT NULL,
    end_time      TIMESTAMP NOT NULL,
    capacity      INT NOT NULL,
    status        VARCHAR NOT NULL,          (scheduled, cancelled)
    CONSTRAINT fk_class_trainer
        FOREIGN KEY (trainer_id) REFERENCES Trainer(trainer_id),
    CONSTRAINT fk_class_room
        FOREIGN KEY (room_id) REFERENCES Room(room_id),
    CONSTRAINT fk_class_admin
        FOREIGN KEY (created_by_admin_id) REFERENCES AdminStaff(admin_id)
)
```

- Admins set up classes (room + trainer), used for room booking and trainer schedule.

2.9 ClassRegistration (many to many Member Class)

```
ClassRegistration(  
    registration_id  INT PRIMARY KEY,  
    class_id        INT NOT NULL,  
    member_id       INT NOT NULL,  
    registered_at   TIMESTAMP NOT NULL,  
    attendance_status VARCHAR NOT NULL,          (registered, attended, no_show)  
    CONSTRAINT fk_classreg_class  
        FOREIGN KEY (class_id) REFERENCES Class(class_id)  
        ON DELETE CASCADE,  
    CONSTRAINT fk_classreg_member  
        FOREIGN KEY (member_id) REFERENCES Member(member_id)  
        ON DELETE CASCADE,  
    CONSTRAINT uq_class_member UNIQUE (class_id, member_id)  
)
```

- This handles the many-to-many relationship (one member can register for many classes, one class can have many members).
- UNIQUE(class_id, member_id) prevents duplicate registrations.

2.10 TrainerAvailability

```
TrainerAvailability(  
    availability_id  INT PRIMARY KEY,  
    trainer_id       INT NOT NULL,  
    day_of_week      INT NOT NULL,          (1 to 7)  
    start_time       TIME NOT NULL,  
    end_time         TIME NOT NULL,  
    is_recurring     BOOLEAN NOT NULL,  
    CONSTRAINT fk_avail_trainer  
        FOREIGN KEY (trainer_id) REFERENCES Trainer(trainer_id)  
        ON DELETE CASCADE  
)
```

- Used for “Set Availability” and to validate PT session booking times.

2.11 Equipment

```
Equipment(  
    equipment_id  INT PRIMARY KEY,  
    room_id       INT NOT NULL,  
    equipment_name VARCHAR NOT NULL,  
    equipment_type VARCHAR,  
    serial_number VARCHAR,  
    is_operational BOOLEAN NOT NULL,  
    CONSTRAINT fk_equipment_room  
        FOREIGN KEY (room_id) REFERENCES Room(room_id)  
)
```

- Admin can see equipment per room.

2.12 EquipmentMaintenance

```
EquipmentMaintenance(  
    maintenance_id    INT PRIMARY KEY,  
    equipment_id     INT NOT NULL,  
    admin_id         INT NOT NULL,  
    reported_at      TIMESTAMP NOT NULL,  
    resolved_at      TIMESTAMP,  
    status           VARCHAR NOT NULL,          (open, in_progress, resolved)  
    issue_description VARCHAR NOT NULL,  
    resolution_notes VARCHAR,  
    CONSTRAINT fk_maint_equipment  
        FOREIGN KEY (equipment_id) REFERENCES Equipment(equipment_id),  
    CONSTRAINT fk_maint_admin  
        FOREIGN KEY (admin_id) REFERENCES AdminStaff(admin_id)  
)
```

- Supports “Equipment Maintenance” so admins log & resolve issues.

3. Normalization and Notes

The ER model contains only strong entities. There are no weak entities in this design. All relationships are implemented using foreign keys (1 to many) or a separate junction table for many to many (such as ClassRegistration). The relational schema derived from the ER model adheres to Third Normal Form (3NF):

- No repeating groups or multivalued attributes
 - All tables have atomic values: names, numbers, timestamps.
- All non-key attributes fully depend on the primary key
 - For example, In FitnessGoal(goal_id PK, target_value, target_date, ...), all attributes describe the goal itself and depend on goal_id.
- No transitive dependencies
 - Member contact info is stored only in Member, not duplicated in other tables.
 - Trainer info is only in Trainer.
 - Room details only in Room.
 - Class registrations store only member_id and class_id (foreign keys), not member or class attributes.
- Many-to-many relationships are represented using junction tables
 - ClassRegistration(class_id, member_id) replaces any redundancy in storing enrolled members inside Class.
- No derived / computed values stored
 - Metrics are logged historically, not overwritten.
 - Member dashboard information is computed at query-time, not stored redundantly.

Therefore, the schema avoids redundancy, updates remain consistent, and all tables meet 3NF requirements.

