3005 Project Outline

Functional Requirements:

- Member Functions:
  - User Registration
  - Profile Management
  - Dashboard
  - PT Session Scheduling
  - Group Class Registration
- Trainer Functions:
  - Set Availability
  - Schedule View
- Administrative Staff Functions:
  - Room Booking
  - Equipment Maintenance
  - Class Management

Entities:

- UserAccount
- Member
- Trainer
- FitnessGoal
- HealthMetric
- GroupClass
- PTSession
- Room
- Equipment
- MaintenanceTicket

Relations:

- Member -> isMemberAcc -> UserAccount
- Trainer -> isTrainerAcc -> UserAccount
- Member -> sets -> FitnessGoal
- Member -> sets -> HealthMetric

- Member -> sessionRegistration -> PTSession
- Member -> ClassRegistration -> GroupClass
- Trainer -> Teaches -> GroupClass
- Trainer -> assigned -> PTSession
- Room -> hosts -> PTSession
- Room -> hosts -> GroupClasses
- Room -> contains -> Equipment

Database Schema:

```
CREATE TABLE UserAccount (
   user_id      INT GENERATED ALWAYS AS IDENTITY,
   email        VARCHAR(255) NOT NULL UNIQUE,
   password     VARCHAR(255) NOT NULL,
   role_type    TEXT NOT NULL,
   is_active    BOOLEAN NOT NULL DEFAULT TRUE,
   PRIMARY KEY (user_id),
   CHECK (role_type IN ('MEMBER', 'TRAINER', 'ADMIN'))
);


CREATE TABLE Member (
   member_id         INT PRIMARY KEY,
   name              VARCHAR(255) NOT NULL,
   dob               DATE,
   gender            TEXT,
   phone             VARCHAR(30),
   address           VARCHAR(255),
   registration_date  DATE NOT NULL DEFAULT CURRENT_DATE,
   FOREIGN KEY (member_id) REFERENCES UserAccount(user_id)
);
```

```
CREATE TABLE Trainer (
    trainer_id   INT PRIMARY KEY,
    name         VARCHAR(255) NOT NULL,
    start_time   TIMESTAMP NOT NULL,
    end_time     TIMESTAMP NOT NULL,
    FOREIGN KEY (trainer_id) REFERENCES UserAccount(user_id),
    CHECK (end_time > start_time)
);

CREATE TABLE FitnessGoal (
        goal_id             INT GENERATED ALWAYS AS IDENTITY,
        member_id           INT NOT NULL,
        goal_type           VARCHAR(255),
        target_value        NUMERIC,
        start_date          DATE NOT NULL,
        end_date            DATE NOT NULL,
        PRIMARY KEY         (goal_id),
        FOREIGN KEY         (member_id) REFERENCES Member(member_id)
);

CREATE TABLE HealthMetric (
        measured_at         TIMESTAMP NOT NULL DEFAULT NOW(),
        member_id           INT NOT NULL,
        height              NUMERIC,
        weight              NUMERIC,
        bfp                     NUMERIC,
        heart_rate              INT,
        PRIMARY KEY             (member_id, measured_at),
        FOREIGN KEY             (member_id) REFERENCES Member(member_id)
);

CREATE TABLE Room (
        room_id                         INT GENERATED ALWAYS AS IDENTITY,
        name                    VARCHAR(255) NOT NULL,
        capacity                INT NOT NULL,
        PRIMARY KEY             (room_id)
);

CREATE TABLE Equipment (
        room_id                         INT NOT NULL,
        equipment_no                    INT NOT NULL,
        name                    VARCHAR(255) NOT NULL,
        type                    VARCHAR(255) NOT NULL,
        PRIMARY KEY             (room_id, equipment_no),
        FOREIGN KEY             (room_id) REFERENCES Room(room_id)
);

CREATE TABLE MaintenanceTicket (
        ticket_id               INT GENERATED ALWAYS AS IDENTITY,
        room_id                         INT,
        equipment_no                    INT,
        issue                   VARCHAR(255) NOT NULL,
        priority        VARCHAR(50) NOT NULL,
        status                  VARCHAR(50) NOT NULL DEFAULT 'OPEN',
        PRIMARY KEY             (ticket_id),
        FOREIGN KEY             (room_id) REFERENCES Room(room_id),
        FOREIGN KEY             (room_id, equipment_no) REFERENCES Equipment(room_id,
equipment_no),
        CHECK (
                (room_id IS NOT NULL AND equipment_no IS NULL)
                OR
                (room_id IS NOT NULL AND equipment_no IS NOT NULL)
        )
);

CREATE TABLE GroupClass (
        class_id                INT GENERATED ALWAYS AS IDENTITY,
        class_name              VARCHAR(255) NOT NULL,
        trainer_id              INT NOT NULL,
        room_id                         INT NOT NULL,
        scheduled_at            TIMESTAMP NOT NULL,
        capacity                INT NOT NULL,
        duration_minutes        INT NOT NULL,
        PRIMARY KEY             (class_id),
        FOREIGN KEY             (trainer_id) REFERENCES Trainer(trainer_id),
        FOREIGN KEY             (room_id) REFERENCES Room(room_id)
);
```

```
CREATE TABLE ClassRegistration (
        registration_id     INT GENERATED ALWAYS AS IDENTITY,
        class_id            INT NOT NULL,
        member_id           INT NOT NULL,
        PRIMARY KEY         (registration_id),
        FOREIGN KEY         (class_id) REFERENCES GroupClass(class_id),
        FOREIGN KEY         (member_id) REFERENCES Member(member_id),
        UNIQUE                      (class_id, member_id)
);
```

```
CREATE TABLE PTSession (
        session_id          INT GENERATED ALWAYS AS IDENTITY,
        member_id           INT NOT NULL,
        trainer_id          INT NOT NULL,
        room_id                     INT NOT NULL,
        session_at          TIMESTAMP NOT NULL,
        duration_minutes    INT NOT NULL,
        PRIMARY KEY         (session_id),
        FOREIGN KEY         (member_id) REFERENCES Member(member_id),
        FOREIGN KEY         (trainer_id) REFERENCES Trainer(trainer_id),
        FOREIGN KEY         (room_id) REFERENCES Room(room_id)
);
```

Normalization Justification:

1. UserAccount

Attributes / key

- Key: user_id
- Other: email, password, role_type, is_active

Functional dependencies

- user_id -> email, password, role_type, is_active
- email -> user_id, password, role_type, is_active (email is a candidate key)

Normalization

- 2NF: PK is a single attribute = no partial dependencies.
- 3NF: Every non-key attribute depends on a key (either user_id or email), and no non-key attribute determines another non-key attribute as a non-key determinant.

2. Member

   Attributes / key

   - Key: member_id (also FK to UserAccount(user_id))
   - Other: name, dob, gender, phone, address, registration_date

   FDs

   - member_id -> name, dob, gender, phone, address, registration_date

   Normalization

   - 2NF: Single-attribute PK = no partial dependencies.
   - 3NF: All non-key attributes depend directly on member_id, and there are no non-key → non-key FDs inside this table.

3. Trainer

   Attributes / key

   - Key: trainer_id (also FK to UserAccount(user_id))
   - Other: name, start_time, end_time, created_at

   FDs

   - trainer_id -> name, start_time, end_time, created_at

Normalization

- 2NF: Single-attribute PK = no partial dependencies.
- 3NF: All non-key attributes depend on the key; no non-key attribute determines another non-key attribute.

4. FitnessGoal

Attributes / key

- Key: goal_id
- Other: member_id, goal_type, target_value, start_date, end_date

FDs

- goal_id -> member_id, goal_type, target_value, start_date, end_date

Normalization

- 2NF: Single-attribute PK = no partial dependencies.
- 3NF: All non-key attributes depend on goal_id; there are no non-key determinants (e.g., we do not enforce member_id or (member_id, goal_type, start_date) as a key here).

5. HealthMetric

Attributes / key

- Key: (member_id, measured_at)
- Other: height, weight, bfp, heart_rate

FDs

- (member_id, measured_at) -> height, weight, bfp, heart_rate

2NF (partial dependency)

- height, weight, bfp, heart_rate do not depend solely on member_id (a member has many measurements).
- They also do not depend solely on measured_at (different members can share timestamps).
- All non-key attributes depend on the full composite key, not on a subset.

3NF (transitive)

- Only determinant for non-key attributes is the composite key (member_id, measured_at).
- No non-key attribute determines another non-key attribute.

6. Room

Attributes / key

- Key: room_id
- Other: name, capacity, location

FDs

- room_id -> name, capacity, location

Normalization

- 2NF: Single-attribute PK = no partial dependencies.
- 3NF: No non-key attribute is modeled as a determinant; all depend on room_id.

7. Equipment

Attributes / key

- Key: (room_id, equipment_no)
- Other: name, type

FDs

- (room_id, equipment_no) → name, type

2NF

- name and type do not depend on room_id alone or equipment_no alone (equipment_no is only unique within a room).
- Non-key attributes depend on the full composite key.

3NF

- No non-key attribute determines another non-key attribute.

8. MaintenanceTicket

Attributes / key

- Key: ticket_id
- Other: room_id, equipment_no, issue, priority, status

FDs

- ticket_id -> room_id, equipment_no, issue, priority, status

Normalization

- 2NF: Single-attribute PK = no partial dependencies.
- 3NF: All non-key attributes depend on ticket_id; no non-key determinants.

9. GroupClass

Attributes / key

- Key: class_id
- Other: class_name, trainer_id, room_id, scheduled_at, capacity, duration_minutes

FDs

- class_id -> class_name, trainer_id, room_id, scheduled_at, capacity, duration_minutes

Normalization

- 2NF: Single-attribute PK = no partial dependencies.
- 3NF: Non-key attributes are not used as determinants inside this table; any extra semantics (e.g., trainer_id -> trainer_name) live in other tables.

10. ClassRegistration

Attributes / key

- Key: registration_id
- Other: class_id, member_id
- Additional constraint: UNIQUE (class_id, member_id)

FDs

- registration_id -> class_id, member_id

- (class_id, member_id) -> registration_id (because of UNIQUE; this is a candidate key)

Normalization

- 2NF: Chosen PK is single attribute registration_id = no partial dependencies w.r.t. that key.
- 3NF: Determinants are keys (registration_id or (class_id, member_id)); there are no non-key determinants.

11. PTSession

Attributes / key

- Key: session_id
- Other: member_id, trainer_id, room_id, session_at, duration_minutes

FDs

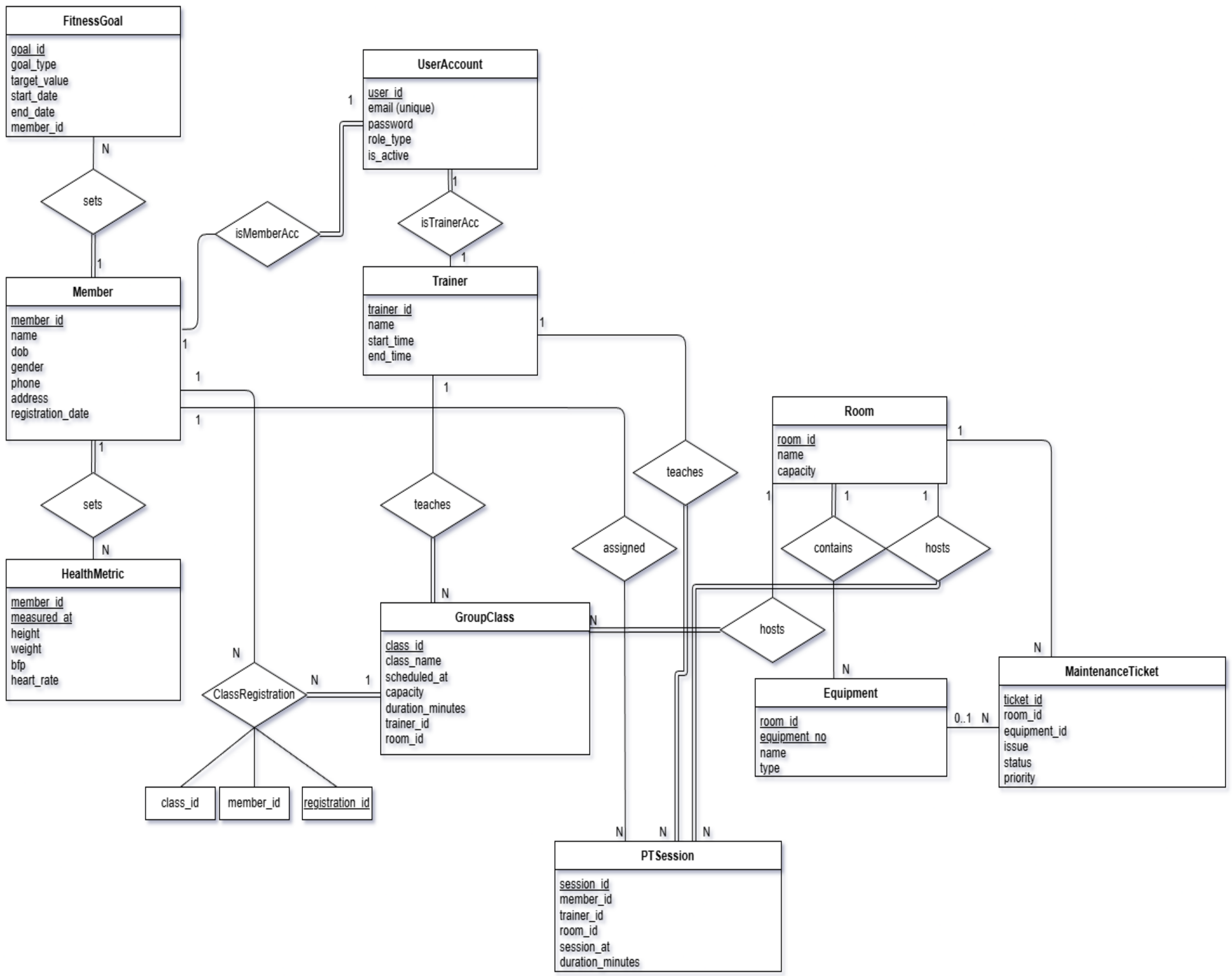
- session_id -> member_id, trainer_id, room_id, session_at, duration_minutes

Normalization

- 2NF: Single-attribute PK = no partial dependencies.
- 3NF: Non-key attributes all depend on the key, and there are no additional non-keys -> non-key FDs defined in this relation.

Overall, given the assumed functional dependencies, all relations in the schema satisfy Second Normal Form and Third Normal Form. There are no partial dependencies on a proper subset of any candidate key, and there are no non-key attributes that functionally determine other non-key attributes within a relation.

ER-Model:


FitnessGoal
goal_id
goal_type
target_value
start_date
end_date
member_id
N
sets
1
UserAccount
user_id
email (unique)
password
role_type
is_active
1
isMemberAcc
1
isTrainerAcc
1
Member
member_id
name
dob
gender
phone
address
registration_date
1
1
1
1
Trainer
trainer_id
name
start_time
end_time
1
1
sets
N
HealthMetric
member_id
measured_at
height
weight
bfp
heart_rate
teaches
N
teaches
assigned
Room
room_id
name
capacity
1
1
1
1
contains
hosts
GroupClass
class_id
class_name
scheduled_at
capacity
duration_minutes
trainer_id
room_id
N
hosts
N
ClassRegistration
N
1
class_id
member_id
registration_id
N
Equipment
room_id
equipment_no
name
type
0..1
N
N
MaintenanceTicket
ticket_id
room_id
equipment_id
issue
status
priority
N
N
N
PTSession
session_id
member_id
trainer_id
room_id
session_at
duration_minutes