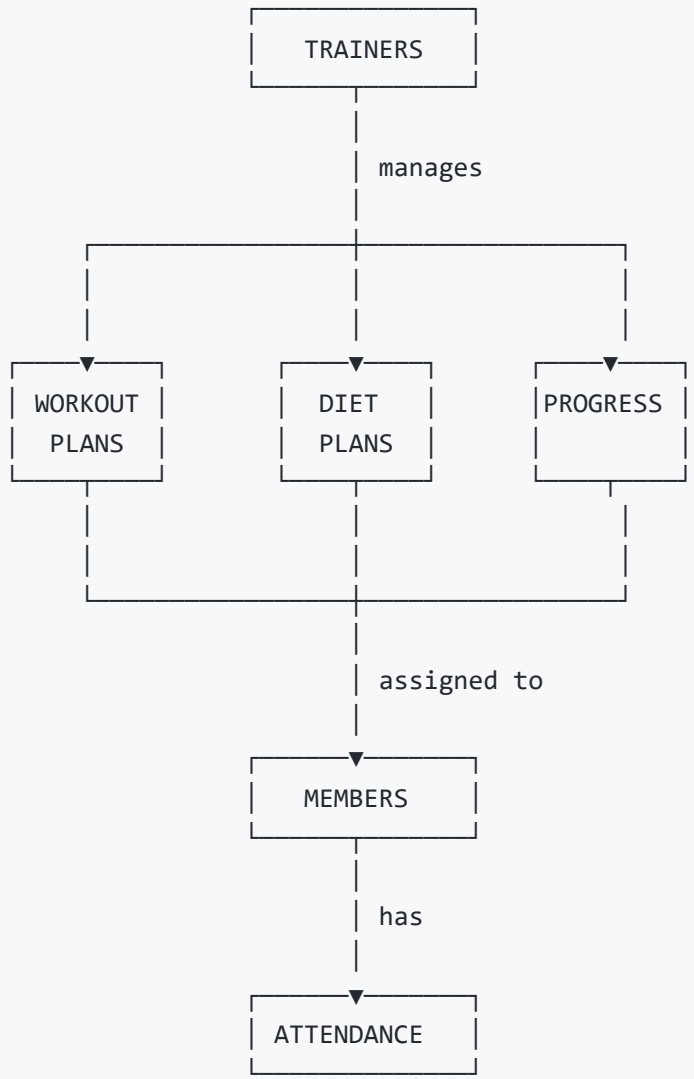
Entity-Relationship Diagrams

Gym Management System

---

1. HIGH-LEVEL ER DIAGRAM



---

2. DETAILED ER DIAGRAM WITH ATTRIBUTES

Entity: MEMBERS

```
├──────────────────────────────────────┤
│ PK: member_id (UUID)                 │
│                                      │
│ • name (VARCHAR)                     │
│ • email (VARCHAR) [UNIQUE]           │
│ • password (VARCHAR) [HASHED]        │
│ • age (INTEGER)                      │
│ • gender (VARCHAR)                   │
│ • phone (VARCHAR)                    │
│ • join_date (TIMESTAMP)              │
│ • status (VARCHAR) DEFAULT 'active'  │
└──────────────────────────────────────┘
```

Entity: TRAINERS

```
┌──────────────────────────────────────┐
│            TRAINERS                  │
├──────────────────────────────────────┤
│ PK: trainer_id (UUID)                │
│                                      │
│ • name (VARCHAR)                     │
│ • email (VARCHAR) [UNIQUE]           │
│ • password (VARCHAR) [HASHED]        │
│ • specialization (VARCHAR)           │
└──────────────────────────────────────┘
```

Entity: WORKOUT_PLANS

```
┌──────────────────────────────────────┐
│         WORKOUT_PLANS                │
├──────────────────────────────────────┤
│ PK: plan_id (UUID)                   │
│ FK: member_id → MEMBERS              │
│ FK: trainer_id → TRAINERS            │
│                                      │
│ • plan_details (TEXT)                │
│ • created_at (TIMESTAMP)             │
└──────────────────────────────────────┘
```

Entity: DIET_PLANS

```
┌──────────────────────────────────────┐
│          DIET_PLANS                  │
├──────────────────────────────────────┤
│ PK: diet_id (UUID)                   │
│ FK: member_id → MEMBERS              │
│ FK: trainer_id → TRAINERS            │
```

```
│                                          │
│ • diet_details (TEXT)                    │
│ • created_at (TIMESTAMP)                 │
└──────────────────────────────────────────┘
```

Entity: ATTENDANCES

```
┌──────────────────────────────────────────┐
│         ATTENDANCES                      │
├──────────────────────────────────────────┤
│ PK: attendance_id (UUID)                 │
│ FK: member_id → MEMBERS                  │
│                                          │
│ • date (TIMESTAMP)                       │
│ • status (VARCHAR)                       │
└──────────────────────────────────────────┘
```

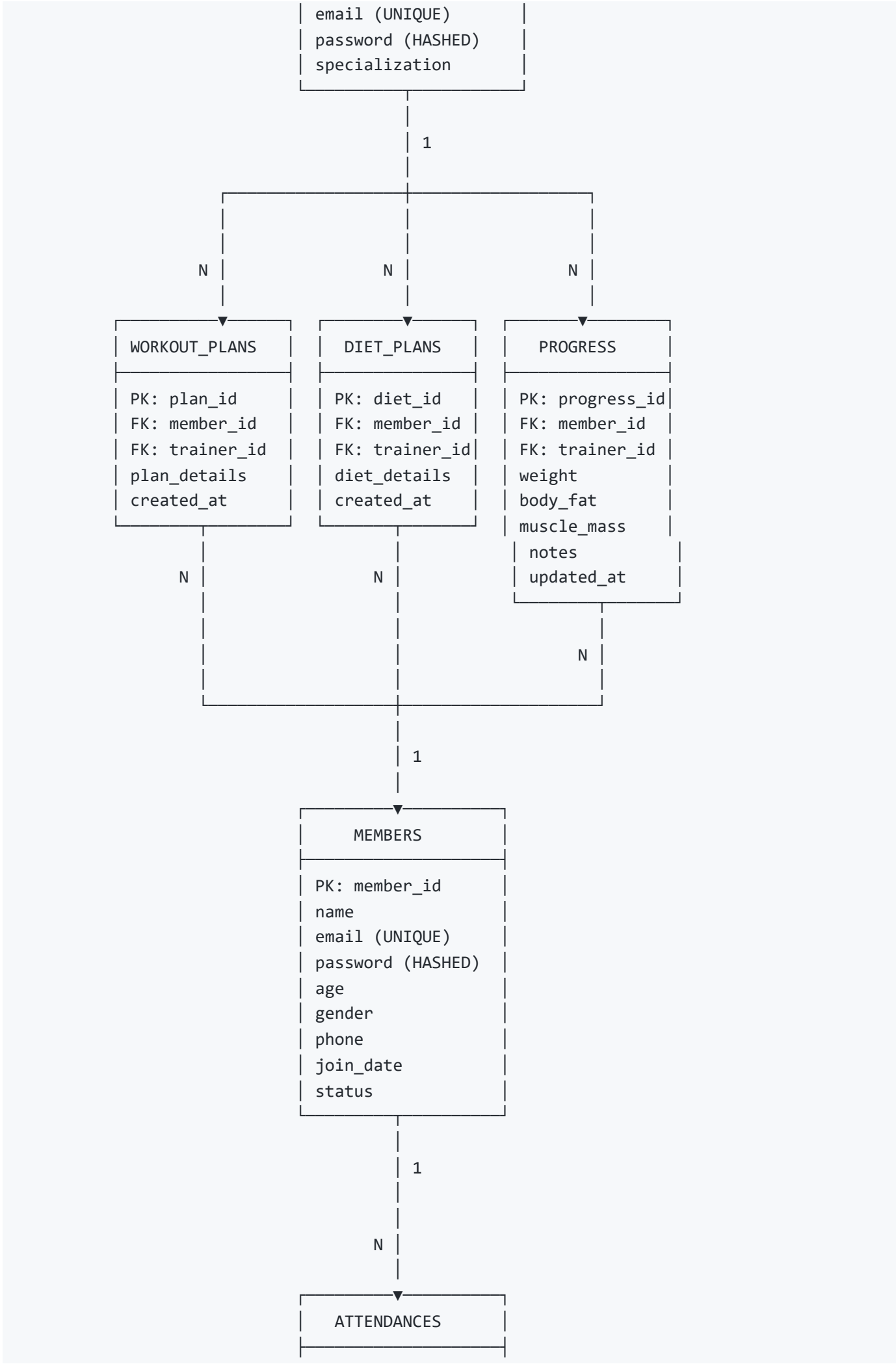
Entity: PROGRESS

```
┌──────────────────────────────────────────┐
│           PROGRESS                       │
├──────────────────────────────────────────┤
│ PK: progress_id (UUID)                   │
│ FK: member_id → MEMBERS                  │
│ FK: trainer_id → TRAINERS                │
│                                          │
│ • weight (FLOAT)                         │
│ • body_fat (FLOAT)                       │
│ • muscle_mass (FLOAT)                    │
│ • notes (TEXT) [NULLABLE]                │
│ • updated_at (TIMESTAMP)                 │
└──────────────────────────────────────────┘
```

---

3. COMPLETE ER DIAGRAM WITH RELATIONSHIPS

```
┌──────────────────────────────────────────────────────────────────────────────┐
│                    GYM MANAGEMENT SYSTEM - ER DIAGRAM                        │
└──────────────────────────────────────────────────────────────────────────────┘


                          ┌──────────────────────┐
                          │     TRAINERS         │
                          ├──────────────────────┤
                          │ PK: trainer_id       │
                          │ name                 │
```


email (UNIQUE)
password (HASHED)
specialization
1
N
N
N
WORKOUT_PLANS
PK: plan_id
FK: member_id
FK: trainer_id
plan_details
created_at
DIET_PLANS
PK: diet_id
FK: member_id
FK: trainer_id
diet_details
created_at
PROGRESS
PK: progress_id
FK: member_id
FK: trainer_id
weight
body_fat
muscle_mass
notes
updated_at
N
N
N
1
MEMBERS
PK: member_id
name
email (UNIQUE)
password (HASHED)
age
gender
phone
join_date
status
1
N
ATTENDANCES

```
                              | PK: attendance_id  |
                              | FK: member_id      |
                              | date               |
                              | status             |
                              └────────────────────┘
```

---

4. RELATIONSHIP DESCRIPTIONS

4.1 TRAINERS ↔ WORKOUT_PLANS

- **Relationship Type**: One-to-Many (1:N)
- **Description**: One trainer can create multiple workout plans
- **Cardinality**:
  - Trainer: 1 (minimum 0, maximum unlimited)
  - Workout Plan: N (each plan has exactly 1 trainer)
- **Participation**:
  - Trainer: Partial (trainer can exist without creating plans)
  - Workout Plan: Total (every plan must have a trainer)

4.2 TRAINERS ↔ DIET_PLANS

- **Relationship Type**: One-to-Many (1:N)
- **Description**: One trainer can create multiple diet plans
- **Cardinality**:
  - Trainer: 1 (minimum 0, maximum unlimited)
  - Diet Plan: N (each plan has exactly 1 trainer)
- **Participation**:
  - Trainer: Partial (trainer can exist without creating plans)
  - Diet Plan: Total (every plan must have a trainer)

4.3 TRAINERS ↔ PROGRESS

- **Relationship Type**: One-to-Many (1:N)
- **Description**: One trainer can record multiple progress entries
- **Cardinality**:
  - Trainer: 1 (minimum 0, maximum unlimited)
  - Progress: N (each entry has exactly 1 trainer)
- **Participation**:
  - Trainer: Partial (trainer can exist without recording progress)
  - Progress: Total (every entry must have a trainer)

4.4 MEMBERS ↔ WORKOUT_PLANS

- **Relationship Type**: One-to-Many (1:N)
- **Description**: One member can have multiple workout plans
- **Cardinality**:
  - Member: 1 (minimum 0, maximum unlimited)
  - Workout Plan: N (each plan belongs to exactly 1 member)
- **Participation**:
  - Member: Partial (member can exist without workout plans)
  - Workout Plan: Total (every plan must belong to a member)

4.5 MEMBERS ↔ DIET_PLANS

- **Relationship Type**: One-to-Many (1:N)
- **Description**: One member can have multiple diet plans
- **Cardinality**:
  - Member: 1 (minimum 0, maximum unlimited)
  - Diet Plan: N (each plan belongs to exactly 1 member)
- **Participation**:
  - Member: Partial (member can exist without diet plans)
  - Diet Plan: Total (every plan must belong to a member)

4.6 MEMBERS ↔ ATTENDANCES

- **Relationship Type**: One-to-Many (1:N)
- **Description**: One member can have multiple attendance records
- **Cardinality**:
  - Member: 1 (minimum 0, maximum unlimited)
  - Attendance: N (each record belongs to exactly 1 member)
- **Participation**:
  - Member: Partial (member can exist without attendance records)
  - Attendance: Total (every record must belong to a member)

4.7 MEMBERS ↔ PROGRESS

- **Relationship Type**: One-to-Many (1:N)
- **Description**: One member can have multiple progress records
- **Cardinality**:
  - Member: 1 (minimum 0, maximum unlimited)
  - Progress: N (each record belongs to exactly 1 member)
- **Participation**:

- Member: Partial (member can exist without progress records)
- Progress: Total (every record must belong to a member)

---

5. CARDINALITY MATRIX

Entity 1	Relationship	Entity 2	Cardinality	Participation
TRAINERS	creates	WORKOUT_PLANS	1:N	Partial:Total
TRAINERS	creates	DIET_PLANS	1:N	Partial:Total
TRAINERS	records	PROGRESS	1:N	Partial:Total
MEMBERS	has	WORKOUT_PLANS	1:N	Partial:Total
MEMBERS	has	DIET_PLANS	1:N	Partial:Total
MEMBERS	has	ATTENDANCES	1:N	Partial:Total
MEMBERS	has	PROGRESS	1:N	Partial:Total

---

6. CONSTRAINTS AND BUSINESS RULES

6.1 Entity Constraints

1. **MEMBERS.email**: Must be unique across all members
2. **TRAINERS.email**: Must be unique across all trainers
3. **MEMBERS.password**: Must be hashed using bcrypt
4. **TRAINERS.password**: Must be hashed using bcrypt
5. **MEMBERS.status**: Default value is 'active'
6. **MEMBERS.join_date**: Auto-set to current timestamp

6.2 Referential Integrity Constraints

1. **ON DELETE CASCADE**: When a member is deleted, all related records (workout plans, diet plans, attendances, progress) are automatically deleted
2. **ON DELETE CASCADE**: When a trainer is deleted, all related records (workout plans, diet plans, progress) are automatically deleted
3. **ON UPDATE CASCADE**: When primary keys are updated, foreign keys are automatically updated

6.3 Business Rules

1. A member must register before being assigned any plans
2. Only trainers can create workout plans and diet plans
3. Only trainers can record member attendance
4. Only trainers can update member progress
5. A workout plan must be assigned to exactly one member by exactly one trainer
6. A diet plan must be assigned to exactly one member by exactly one trainer
7. An attendance record must belong to exactly one member
8. A progress record must belong to exactly one member and be recorded by exactly one trainer

6.4 Data Type Constraints

- **UUID**: Used for all primary keys (member_id, trainer_id, plan_id, etc.)
- **VARCHAR**: Used for text fields (name, email, specialization, status)
- **TEXT**: Used for large text fields (plan_details, diet_details, notes)
- **INTEGER**: Used for age
- **FLOAT**: Used for weight, body_fat, muscle_mass
- **TIMESTAMP**: Used for all date/time fields

---

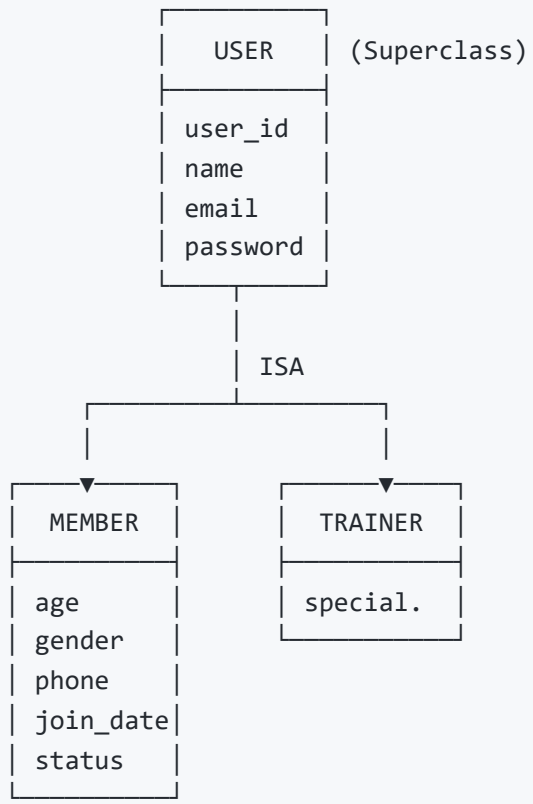
7. ER DIAGRAM NOTATION LEGEND

```
┌─────────────────┐
│   ENTITY    │  = Strong Entity
└─────────────────┘

PK = Primary Key
FK = Foreign Key
[UNIQUE] = Unique Constraint
[HASHED] = Hashed Value
[NULLABLE] = Can be NULL
DEFAULT = Default Value

1:N = One-to-Many Relationship
1:1 = One-to-One Relationship
N:M = Many-to-Many Relationship

│ = Relationship Line
┼ = Connection Point
▼ = Relationship Direction
```

---

8. EXTENDED ER FEATURES

8.1 Specialization/Generalization



Note: The current implementation uses separate tables for members and trainers (no inheritance), but this diagram shows how it could be modeled with specialization.

8.2 Weak Entities

In this system, there are no weak entities. All entities have their own unique identifiers (UUIDs) and can exist independently, though their relationships may be enforced through foreign key constraints.

Document Version: 1.0
Date: November 22, 2025