ER diagram + Mapping + Normalization

1. Entity-Relationship Model

Entities and Attributes

1.1 Member

Description: Represents club members who use fitness services

Attributes:

- **member_id** (PK): Unique identifier
- email: Unique email address
- password: Account password
- first_name: Member's first name
- last_name: Member's last name
- date_of_birth: Birth date
- gender: Gender identity
- phone: Contact number
- address: Residential address
- registration_date: Date joined club

Constraints:

- Email must be unique and valid format
- Date of birth must be valid date

1.2 Trainer

Description: Fitness professionals who conduct sessions and classes

Attributes:

- **trainer_id** (PK): Unique identifier
- email: Unique email address
- password: Account password
- first_name: Trainer's first name
- last_name: Trainer's last name
- specialization: Area of expertise (e.g., "Strength Training")
- phone: Contact number
- hire_date: Employment start date

1.3 AdminStaff

Description: Administrative personnel managing operations

Attributes:

- **admin_id** (PK): Unique identifier
- email: Unique email address
- password: Account password
- first_name: Admin's first name
- last_name: Admin's last name
- role: Administrative role (e.g., "Manager")
- phone: Contact number

1.4 Room

Description: Physical spaces for activities

Attributes:

- **room_id** (PK): Unique identifier
- room_name: Display name (e.g., "Studio A")
- capacity: Maximum occupancy
- room_type: Purpose (e.g., "Group Class", "Personal Training")

Constraints:

- Capacity must be positive

1.5 Equipment

Description: Gym equipment and machines

Attributes:

- **equipment_id** (PK): Unique identifier
- room_id (FK): Location room
- equipment_name: Item name
- purchase_date: Acquisition date
- status: Current state (Operational/Under Maintenance/Out of Service)
- last_maintenance_date: Most recent service
- maintenance_notes: Service details

1.6 FitnessGoal

Description: Member fitness objectives

Attributes:

- **goal_id** (PK): Unique identifier
- member_id (FK): Goal owner
- goal_type: Goal category (e.g., "Weight Loss")

- target_value: Desired measurement
- current_value: Present measurement
- target_date: Deadline
- created_date: Goal creation date
- status: Progress state (Active/Achieved/Abandoned)

1.7 HealthMetric

Description: Historical health measurements

Attributes:

- **metric_id** (PK): Unique identifier
- member_id (FK): Data owner
- recorded_date: Measurement timestamp
- weight: Body weight (lbs)
- height: Height (inches)
- heart_rate: Heart rate (bpm)
- blood_pressure: Blood pressure reading
- body_fat_percentage: Body fat %
- notes: Additional observations

Design Note: This table maintains complete history - no updates, only inserts

1.8 TrainerAvailability

Description: Trainer working schedules

Attributes:

- **availability_id** (PK): Unique identifier
- trainer_id (FK): Schedule owner
- day_of_week: Weekday (Monday-Sunday)
- start_time: Shift start
- end_time: Shift end

Constraints:

- No overlapping time slots per trainer per day
- Start time must be before end time

1.9 Class

Description: Group fitness sessions

Attributes:

- **class_id** (PK): Unique identifier

- class_name: Activity name (e.g., "Morning Yoga")
- trainer_id (FK): Instructor
- room_id (FK): Location
- schedule_date: Class date
- start_time: Class start
- end_time: Class end
- capacity: Maximum participants
- current_enrollment: Current registrations
- status: State (Scheduled/Completed/Cancelled)

Constraints:

- Current enrollment ≤ capacity
- Start time < end time
- Capacity must be positive

1.10 PersonalTrainingSession

Description: One-on-one training appointments

Attributes:

- **session_id** (PK): Unique identifier
- member_id (FK): Client
- trainer_id (FK): Instructor
- room_id (FK): Location
- session_date: Appointment date
- start_time: Session start
- end_time: Session end
- status: State (Scheduled/Completed/Cancelled)
- notes: Session details

1.11 ClassRegistration

Description: Member enrollments in classes

Attributes:

- **registration_id** (PK): Unique identifier
- member_id (FK): Enrolled member
- class_id (FK): Target class
- registration_date: Enrollment timestamp
- status: State (Registered/Attended/Cancelled)

Constraints:

- Unique combination of member and class

1.12 Bill

Description: Financial invoices

Attributes:

- **bill_id** (PK): Unique identifier
- member_id (FK): Debtor
- bill_date: Invoice date
- due_date: Payment deadline
- total_amount: Total charges
- amount_paid: Payments received
- status: State (Pending/Paid/Overdue/Cancelled)
- description: Bill details

1.13 Payment

Description: Payment transactions

Attributes:

- **payment_id** (PK): Unique identifier
- bill_id (FK): Associated invoice
- payment_date: Transaction timestamp
- amount: Payment value
- payment_method: Method used (Cash/Credit Card/etc.)
- transaction_reference: External reference

2. Relationships

2.1 Member - FitnessGoal (1:N)

- **Cardinality**: One member can have many fitness goals
- **Participation**:
  - Member: Partial (not all members set goals)
  - FitnessGoal: Total (every goal belongs to a member)
- **Mapping**: Foreign key `member_id` in FitnessGoal

2.2 Member - HealthMetric (1:N)

- **Cardinality**: One member can have many health metric entries
- **Participation**:
  - Member: Partial (not all members log metrics)
  - HealthMetric: Total (every metric belongs to a member)
- **Mapping**: Foreign key `member_id` in HealthMetric

2.3 Trainer - TrainerAvailability (1:N)

- **Cardinality**: One trainer can have many availability slots
- **Participation**:
    - Trainer: Partial (new trainers may not have availability set)
    - TrainerAvailability: Total (every slot belongs to a trainer)
- **Mapping**: Foreign key `trainer_id` in TrainerAvailability

2.4 Room - Equipment (1:N)

- **Cardinality**: One room can contain many equipment items
- **Participation**:
    - Room: Partial (not all rooms have equipment)
    - Equipment: Partial (some equipment may not be assigned)
- **Mapping**: Foreign key `room_id` in Equipment with NULL allowed

2.5 Member - PersonalTrainingSession (1:N)

- **Cardinality**: One member can book many sessions
- **Participation**:
    - Member: Partial (not all members book PT)
    - Session: Total (every session has a member)
- **Mapping**: Foreign key `member_id` in PersonalTrainingSession

2.6 Trainer - PersonalTrainingSession (1:N)

- **Cardinality**: One trainer conducts many sessions
- **Participation**:
    - Trainer: Partial (not all trainers have sessions)
    - Session: Total (every session has a trainer)
- **Mapping**: Foreign key `trainer_id` in PersonalTrainingSession

2.7 Room - PersonalTrainingSession (1:N)

- **Cardinality**: One room hosts many sessions
- **Participation**: Both partial
- **Mapping**: Foreign key `room_id` in PersonalTrainingSession

2.8 Trainer - Class (1:N)

- **Cardinality**: One trainer teaches many classes
- **Participation**: Both partial
- **Mapping**: Foreign key `trainer_id` in Class

2.9 Room - Class (1:N)

- **Cardinality**: One room hosts many classes
- **Participation**: Both partial
- **Mapping**: Foreign key `room_id` in Class

2.10 Member - ClassRegistration - Class (M:N)

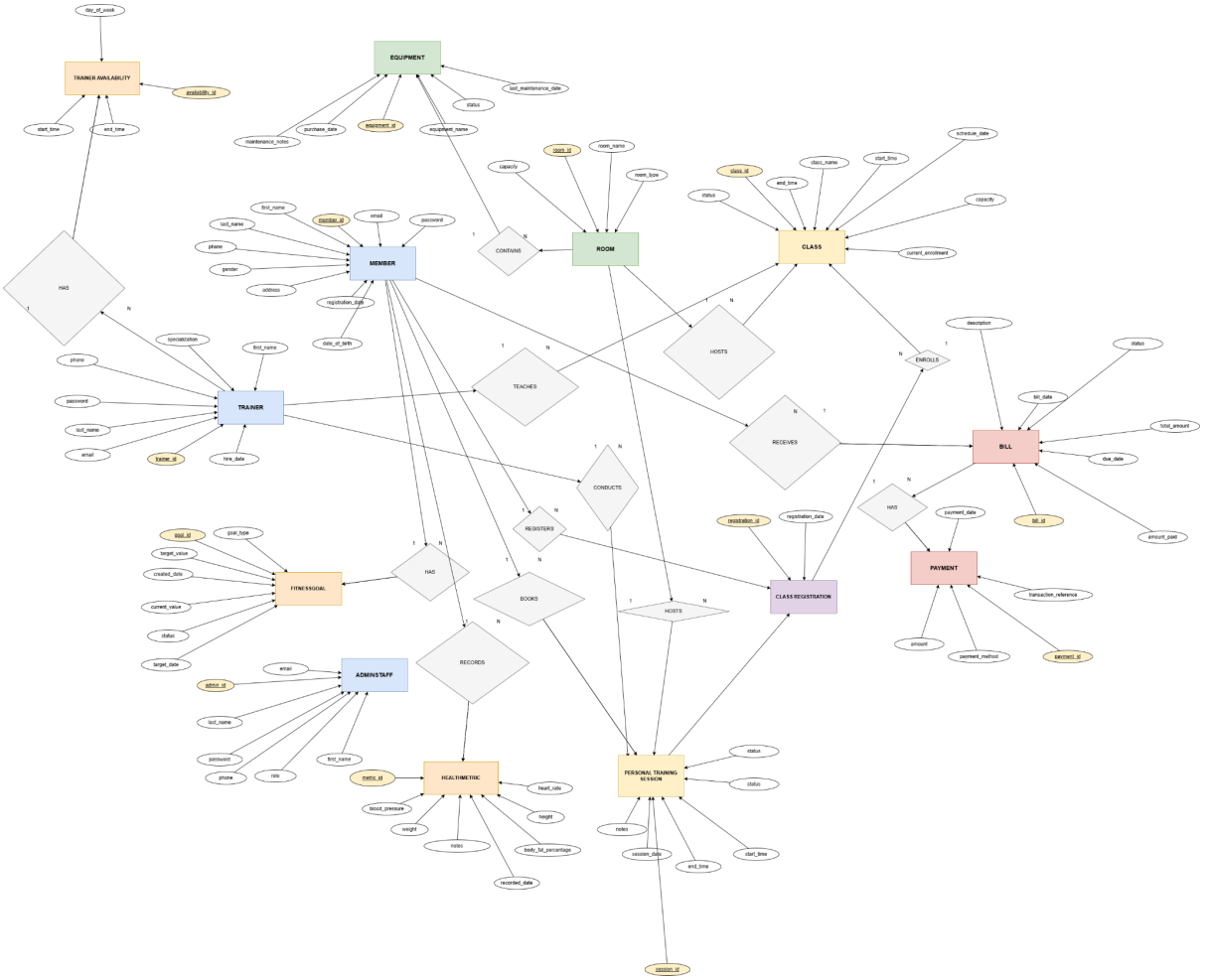
- **Cardinality**: Many members can register for many classes
- **Participation**: Both partial
- **Mapping**: Junction table `ClassRegistration` with foreign keys to both Member and Class

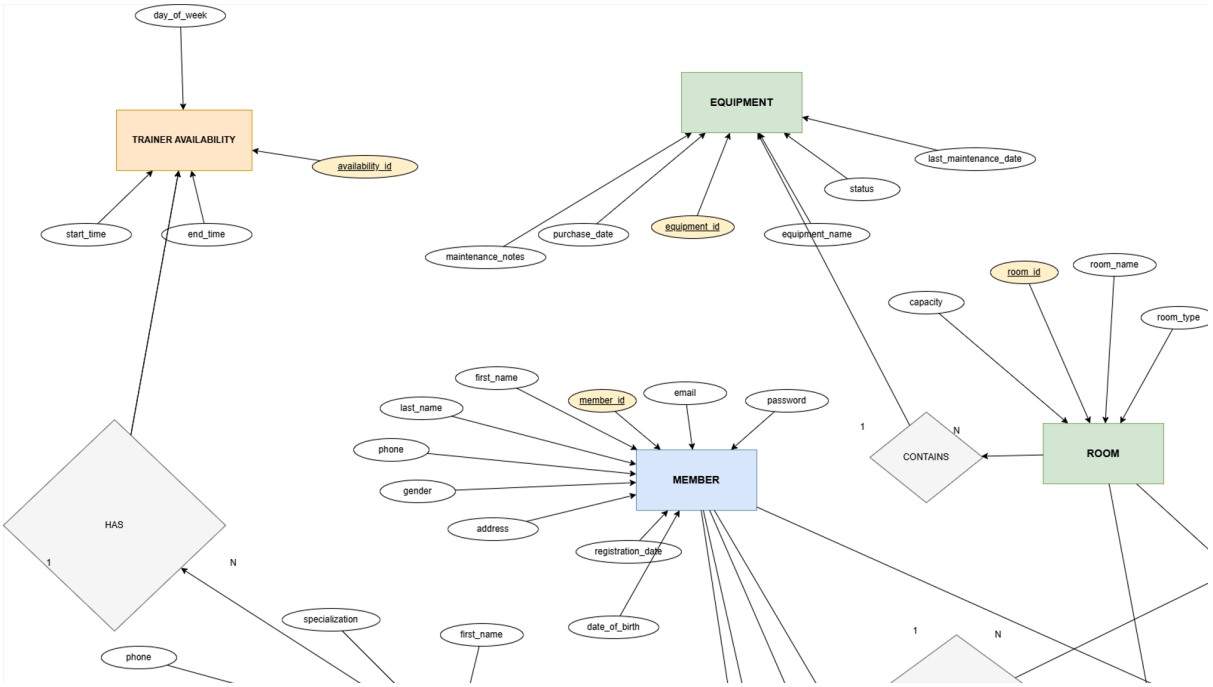
2.11 Member - Bill (1:N)

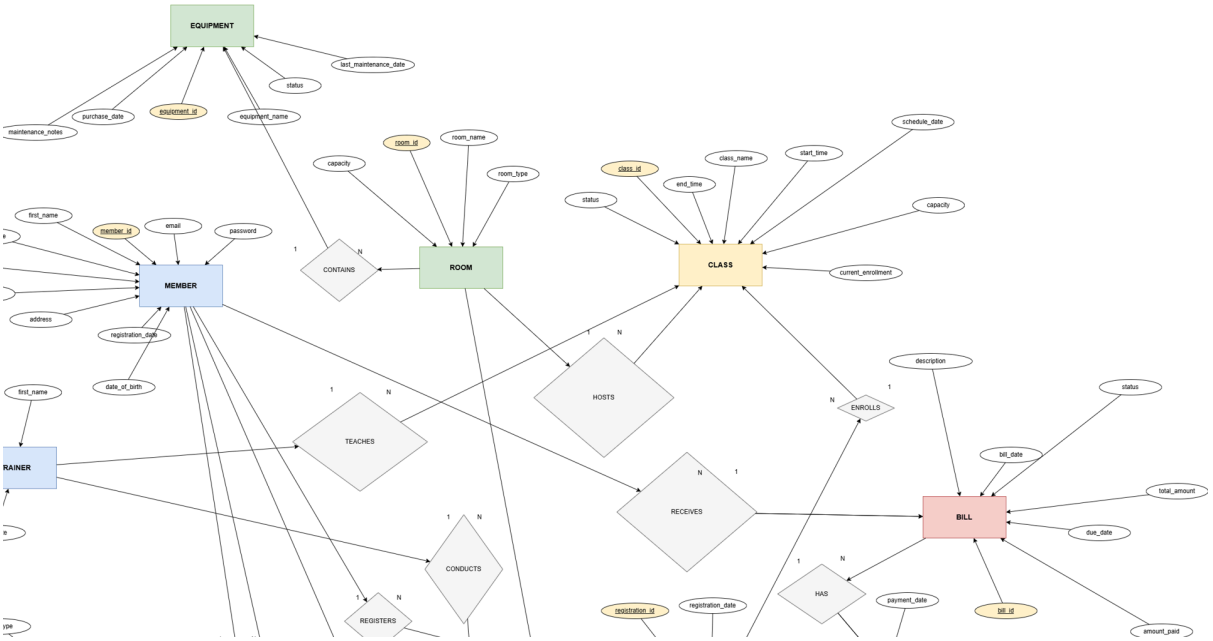
- **Cardinality**: One member can have many bills
- **Participation**:
    - Member: Partial
    - Bill: Total
- **Mapping**: Foreign key `member_id` in Bill

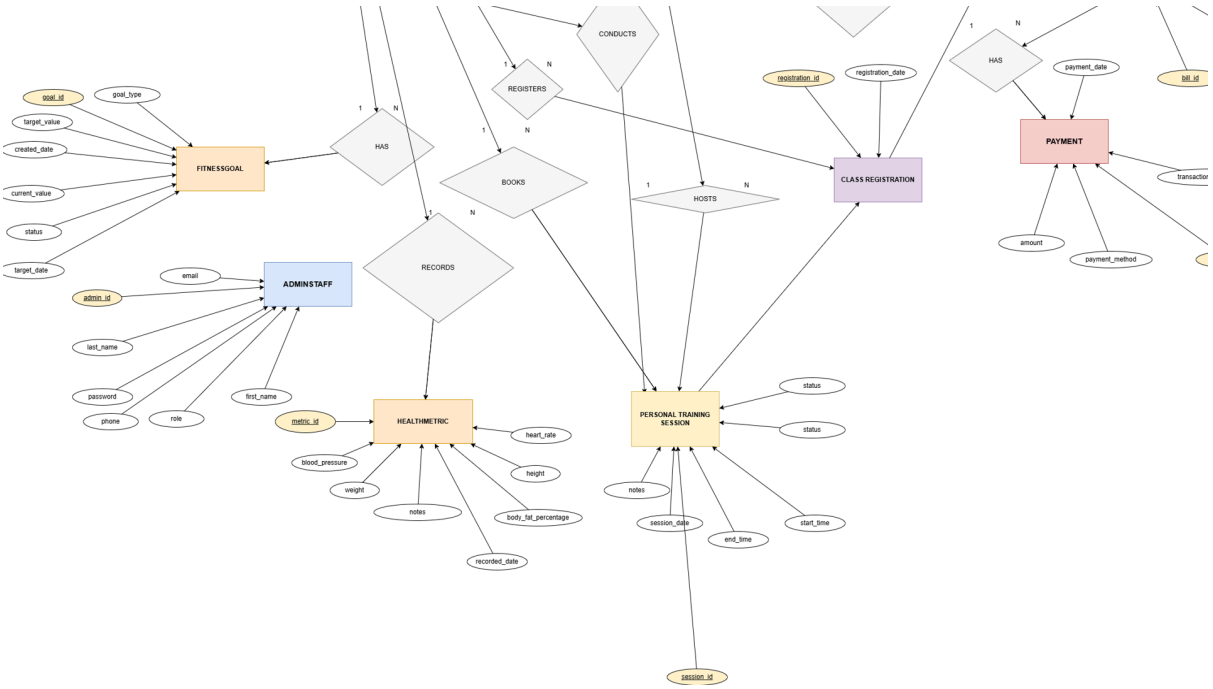
2.12 Bill - Payment (1:N)

- **Cardinality**: One bill can have many payments
- **Participation**:
    - Bill: Partial (not all bills paid yet)
    - Payment: Total
- **Mapping**: Foreign key `bill_id` in Payment


day_of_week
TRAINER AVAILABILITY
availability_id
start_time
end_time
EQUIPMENT
last_maintenance_date
status
maintenance_notes
purchase_date
equipment_id
equipment_name
capacity
room_id
room_name
room_type
CONTAINS
ROOM
class_id
class_name
start_time
schedule_date
status
end_time
capacity
current_enrollment
CLASS
first_name
member_id
email
password
last_name
phone
gender
address
registration_date
MEMBER
HAS
HOSTS
ENROLLS
description
specialization
first_name
phone
password
last_name
email
trainer_id
hire_date
TRAINER
TEACHES
date_of_birth
RECEIVES
BILL
status
bill_date
total_amount
due_date
amount_paid
CONDUCTS
REGISTERS
registration_id
registration_date
HAS
payment_date
bill_id
PAYMENT
transaction_reference
goal_id
goal_type
target_value
created_date
current_value
status
target_date
FITNESSGOAL
HAS
BOOKS
HOSTS
CLASS REGISTRATION
amount
payment_method
payment_id
email
admin_id
last_name
password
phone
role
first_name
ADMINSTAFF
RECORDS
metric_id
HEALTHMETRIC
heart_rate
height
body_fat_percentage
recorded_data
notes
weight
blood_pressure
PERSONAL TRAINING SESSION
status
status
notes
session_date
end_time
start_time
session_id


day_of_week
TRAINER AVAILABILITY
availability_id
start_time
end_time
EQUIPMENT
last_maintenance_date
status
maintenance_notes
purchase_date
equipment_id
equipment_name
room_id
room_name
capacity
room_type
first_name
member_id
email
password
last_name
phone
gender
address
registration_date
date_of_birth
MEMBER
HAS
CONTAINS
ROOM
1
N
specialization
phone
first_name


EQUIPMENT
last_maintenance_date
status
maintenance_notes
purchase_date
equipment_id
equipment_name
room_id
room_name
capacity
room_type
schedule_date
class_id
class_name
start_time
end_time
status
capacity
current_enrolment
first_name
member_id
email
password
address
registration_date
date_of_birth
MEMBER
CONTAINS
ROOM
CLASS
HOSTS
TEACHES
ENROLLS
description
status
bill_date
total_amount
due_date
RECEIVES
BILL
CONDUCTS
REGISTERS
HAS
payment_date
registration_id
registration_date
bill_id
amount_paid
1
N


CONDUCTS
REGISTERS
1
N
1
N
goal_id
goal_type
target_value
created_date
current_value
status
target_date
FITNESSGOAL
HAS
1
N
BOOKS
1
N
HOSTS
1
N
registration_id
registration_date
CLASS REGISTRATION
1
N
HAS
payment_date
bill_id
PAYMENT
transaction
amount
payment_method
email
admin_id
last_name
password
phone
role
ADMINSTAFF
first_name
RECORDS
metric_id
HEALTHMETRIC
heart_rate
blood_pressure
height
weight
notes
body_fat_percentage
recorded_date
status
status
PERSONAL TRAINING SESSION
notes
session_date
end_time
start_time
session_id

4. Mapping ER Model to Relational Schema

4.1 Strong Entities → Tables

Each strong entity becomes a table with its attributes:

sql
Member(member_id, email, password, first_name, last_name, date_of_birth,
gender, phone, address, registration_date)

Trainer(trainer_id, email, password, first_name, last_name, specialization,
phone, hire_date)

AdminStaff(admin_id, email, password, first_name, last_name, role, phone)

Room(room_id, room_name, capacity, room_type)

4.2 Weak Entities → Tables with Foreign Keys

Entities dependent on others include foreign keys:

sql
FitnessGoal(goal_id, member_id*, goal_type, target_value, current_value,
target_date, created_date, status)

HealthMetric(metric_id, member_id*, recorded_date, weight, height,
heart_rate, blood_pressure, body_fat_percentage, notes)

Equipment(equipment_id, room_id*, equipment_name, purchase_date, status,
last_maintenance_date, maintenance_notes)

4.3 1:N Relationships → Foreign Keys

One-to-many relationships add foreign key to "many" side:

TrainerAvailability(availability_id, trainer_id*, day_of_week,start_time, end_time)

PersonalTrainingSession(session_id, member_id*, trainer_id*, room_id*,
session_date, start_time, end_time, status, notes)

Class(class_id, class_name, trainer_id*, room_id*, schedule_date,
start_time, end_time, capacity, current_enrollment, status)

Bill(bill_id, member_id*, bill_date, due_date, total_amount, amount_paid, status,
description)

Payment(payment_id, bill_id*, payment_date, amount, payment_method,
transaction_reference)

4.4 M:N Relationships → Junction Tables

Many-to-many relationships create junction table:

ClassRegistration(registration_id, member_id*, class_id*, registration_date, status)

Rationale: Members and Classes have M:N relationship - a member can register for multiple classes, and a class can have multiple members.

5. Normalization Analysis

First Normal Form (1NF)

All tables satisfy 1NF:

- All attributes contain atomic values
- No repeating groups
- Each column has a unique name
- Order of rows is irrelevant

Example: HealthMetric stores individual measurements (weight, heart_rate) as separate columns, not as a list.

Second Normal Form (2NF)

All tables satisfy 2NF:

- Already in 1NF
- All non-key attributes fully depend on entire primary key
- No partial dependencies

Example: ClassRegistration has composite candidate (member_id, class_id). The attribute status depends on both - which member in which class.

Third Normal Form (3NF)

All tables satisfy 3NF:

- Already in 2NF
- No transitive dependencies
- All non-key attributes depend only on primary key

Example: Member table has no transitive dependencies. Email, name, phone all directly describe the member, not each other.

Normalization Examples

Member Table (3NF)

member_id → {email, first_name, last_name, date_of_birth, ...}

No transitive dependencies exist.

PersonalTrainingSession (3NF)

session_id → {member_id, trainer_id, room_id, session_date, start_time, ...}

All attributes directly describe the session.

Bill and Payment (3NF)

Instead of storing all payment details in Bill table:

Bill(bill_id, total_amount, payment1_amount, payment1_date, payment2_amount, payment2_date, ...) NOT 3NF

We separate into two tables:

Bill(bill_id, total_amount, amount_paid, status) = 3NF

Payment(payment_id, bill_id, amount, payment_date) = 3NF

6. Design Decisions and Assumptions

6.1 Historical Data

Decision: HealthMetric table maintains complete history

- **Rationale**: Track member progress over time
- **Implementation**: No UPDATE operations, only INSERT
- **Benefit**: Enables trend analysis and progress visualization

6.2 Automatic Enrollment Tracking

Decision: Use trigger to update Class.current_enrollment

- **Rationale**: Ensure consistency between ClassRegistration and Class
- **Implementation**: Trigger on INSERT/UPDATE/DELETE of ClassRegistration
- **Benefit**: Prevents manual synchronization errors

6.3 Bill Payment Status

Decision: Auto-update bill status when payments received

- **Rationale**: Maintain accurate financial records
- **Implementation**: Trigger on Payment INSERT
- **Benefit**: Real-time status updates

6.4 Cascade Deletions

Decision: ON DELETE CASCADE for most member relationships

- **Rationale**: When member leaves, remove their data
- **Implementation**: CASCADE on FitnessGoal, HealthMetric, etc.
- **Exception**: ON DELETE SET NULL for trainer/room in sessions (preserve history)

6.5 Availability Slots

Decision: Store recurring weekly schedule in TrainerAvailability

- **Rationale**: Trainers typically have consistent weekly schedules
- **Alternative Considered**: Individual date-based availability (more flexible but complex)
- **Trade-off**: Current design simpler but less flexible for one-off changes

6.6 Room Assignment

Decision: Automatic room assignment in application code

- **Rationale**: Complex business logic for finding available rooms
- **Alternative**: Could use database stored procedure
- **Trade-off**: Application-level gives more flexibility

7. Constraints and Business Rules

Database-Level Constraints

Unique email per user type

UNIQUE(email) on Member, Trainer, AdminStaff

Valid email format

CHECK (email ~* '^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}$')

Positive values

CHECK (capacity > 0)

CHECK (total_amount >= 0)

Time validation

CHECK (start_time < end_time)

Capacity enforcement

CHECK (current_enrollment <= capacity)

Valid status values

CHECK (status IN ('Operational', 'Under Maintenance', 'Out of Service'))

Application-Level Business Rules

1. **Trainer Availability**: Must be available before booking session
2. **Room Conflicts**: No double-booking of rooms
3. **Trainer Conflicts**: No overlapping sessions for same trainer
4. **Class Capacity**: Enforce maximum enrollment
5. **Payment Validation**: Cannot exceed remaining bill amount