HEALTH AND FITNESS CLUB MANAGEMENT SYSTEM REPORT

I. Requirements (Unstructured Text)

1. Application User:

- The users will be organized into 3 different types, which are **members**, **trainers** and **administrative staff**.
- Users will have to **register and log in** using **a unique email** and **a password**.
- Users will also have to provide information, such as **their name**, **DOB**, **address (address, city, province, postal code)**, and **contact details (phone).**

2. Club Members:

- Each member should have a unique identifier.
- Members' profiles should include personal information (name, DOB, contact details…), and health metrics (height, weight, heart rate…).
- Each member may have several fitness goals.
- Each member may take multiple classes.

3. Administrative Staff

- Administrative staff are the ones who manage the club. Staff will have access to a lot of information.
- Staff will have their information such as DOB, address, contact details, and their managers.
- Staff can organize events, such as workshops, and classes.
- Staff should keep track of fitness equipment maintenance.

4. Events

- Everything that happens in a club will be an event. There can even be workshops and more.
- Each event will have its description, start date, end date.
- Each event will have one main head organizer and many other people to help organize the event.

5. Rooms:

- There are rooms in the clubs.
- The Rooms will have their name, purposes, and capacity.

6. Trainers:

- A trainer can have more than 1 certificate.
- Trainer will also have to provide their SIN number, for billing purposes.
- Trainers can manage multiple personal training sessions.
- Each trainer can teach multiple classes.

7. **Personal Training Sessions:**
- Each personal training session may have multiple exercises, and each member can choose to have multiple personal training sessions with different trainers.
- Each trainer can have multiple sessions to train members, but each session has only one member and one trainer (since it is a personal training session).

8. **Financial transactions and billing:**
- Every payment record includes details of the transactions, such as payment methods, member accounts, amounts, status, and payment date.
- The system manages billing for services like membership fees.

9. **Membership plan**
- Each plan will have a plan name, description, duration, price and the benefits that the plan offers

10. **Member subscription:**
- Each member can subscribe to plans to receive benefits
- Each subscription will have a start and end date
- Each subscription will have a price as well.
- Each member can have only one subscription.

11. **Loyalty Program:**
- There are loyalty programs members can take part in.
- Members can participate in the loyalty program to earn or redeem points to get discounts, gift cards, free events, etc.

12. **Equipment:**
- There is equipment at the gyms that need to be maintained.
- Equipment will have their name, type, purchase date and purchase price.
- Equipment has to be maintained in order to be in good condition

13. **Exercises:**
- There are exercises in the app that members can choose, to keep track of their progress.
- User will create
- It will have properties such as name, calories burned, duration, dates and description.

14. **Member fitness goal:**
- Members can have multiple fitness goals.

15. Dashboard:

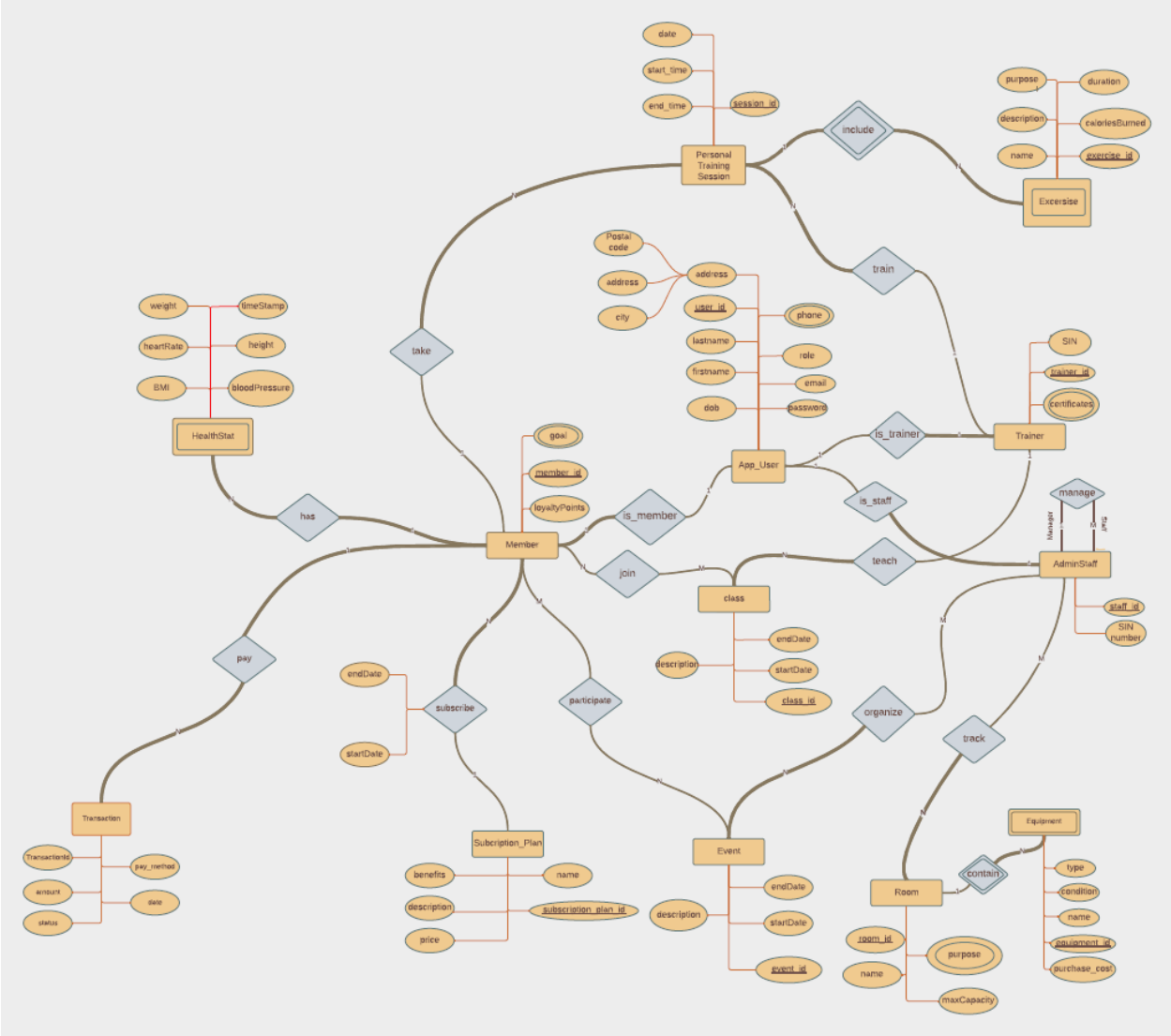
- Health stats (Exercises done, calories burned, weight changes, …).
- The information for the dashboard can be queried in the database

.

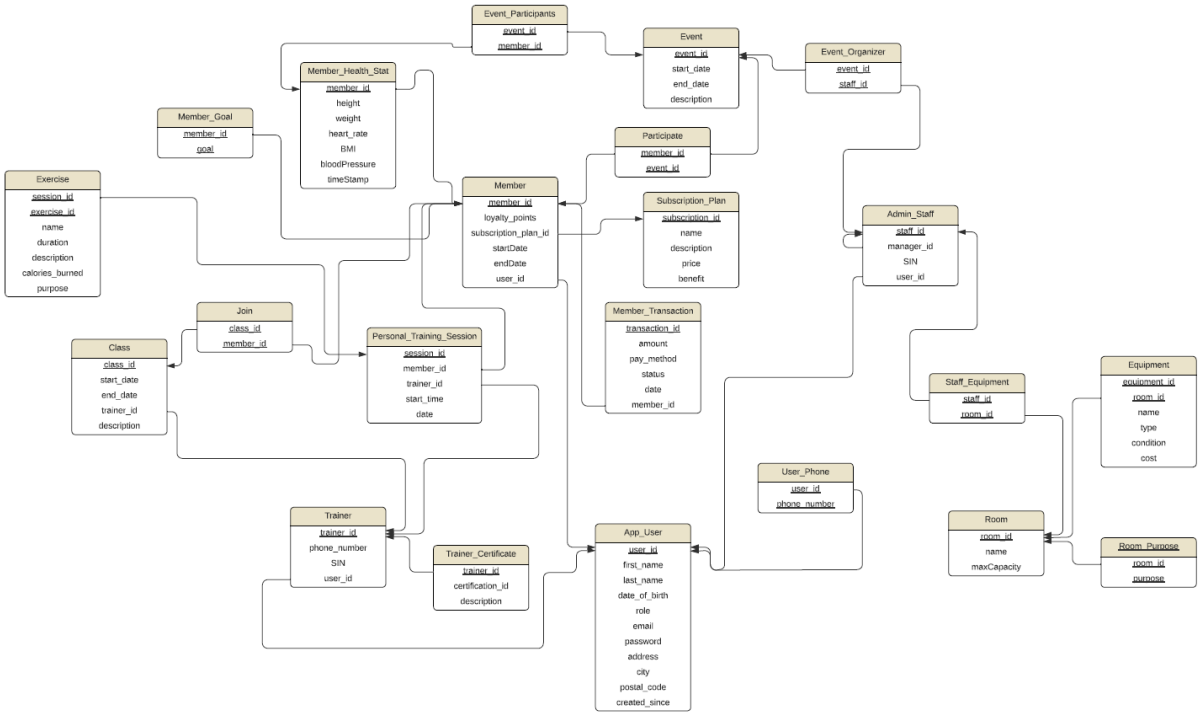
16. Class:

- Each class can have multiple members, and each member can take multiple classes.
- Each class can only have one trainer.

II. Creating an ER model



III. Converting the ER model into a database structure



IV. Normalization

Decomposition to 2NF:

- **Identify any partial dependencies in the relation**
  1. **Member_Goal (<u>member_id</u>, <u>goal_id</u>, description)** has the primary key as a combination of **“member_id”** and **“goal_id”**.

     This table is already in 2NF since each **“description”** is dependent on the entire primary key **“(member_id, goal_id)”** and not on “member_id” or “goal_id” alone.

  2. **Equipment (<u>equipment_id</u>, <u>room_id</u>, name, type, condition, cost)** has the primary key as a combination of **“equipment_id”** and **“room_id”**.

     This table is already in 2NF since each **“name”, “type”, “condition”, “cost”** are dependent on the entire primary key **“(equipment_id, room_id)”** and not on **“equipment_id”** or **“room_id”** alone.

  3. **Exercise (<u>session_id</u>, <u>exercise_id</u>, name, description, duration, caloriesBurned, purpose)** has the primary key as a combination of **“session_id”** and **“exercise_id”**.

     This table is already in 2NF since each **“name”, “duration”, “description”, “caloriesBurned”, “purpose”** are dependent on the entire primary key **“(<u>session_id</u>, <u>exercise_id</u>)”** and not on **“session_id”** or **“exercise_id”** alone.

Decomposition to 3NF:

- **Identify any transitive dependencies in the 2NF relations**
  1. **Exercise (<u>session_id</u>, <u>exercise_id</u>, name, description, duration, caloriesBurned, purpose)** has the primary key as a combination of **“session_id”** and **“exercise_id”**.

Below are transitive dependencies:

name -> caloriesBurned, purpose is a transitive dependency as **“name”** indirectly determines **“caloriesBurned”** and **“purpose”**

2. **Subscription_Plan (subcription_id, name, description, price, benefit)** has a primary key of **“subscription_id”**

   Below are transitive dependencies:

   name -> price, benefit is a transitive dependency as **“name”** indirectly determines **“price”** and **“benefit”**

- **Decompose the 2NF relations into the 3NF relations.**
  1. Decompose table **Exercise**:

     Exercise: {session_id, exercise_id} -> {name, description, duration}

     Exercise_Name: name -> {caloriesBurned, purpose}

     New relation: name -> {caloriesBurned, purpose}

  2. Decompose table **Subscription_Plan:**

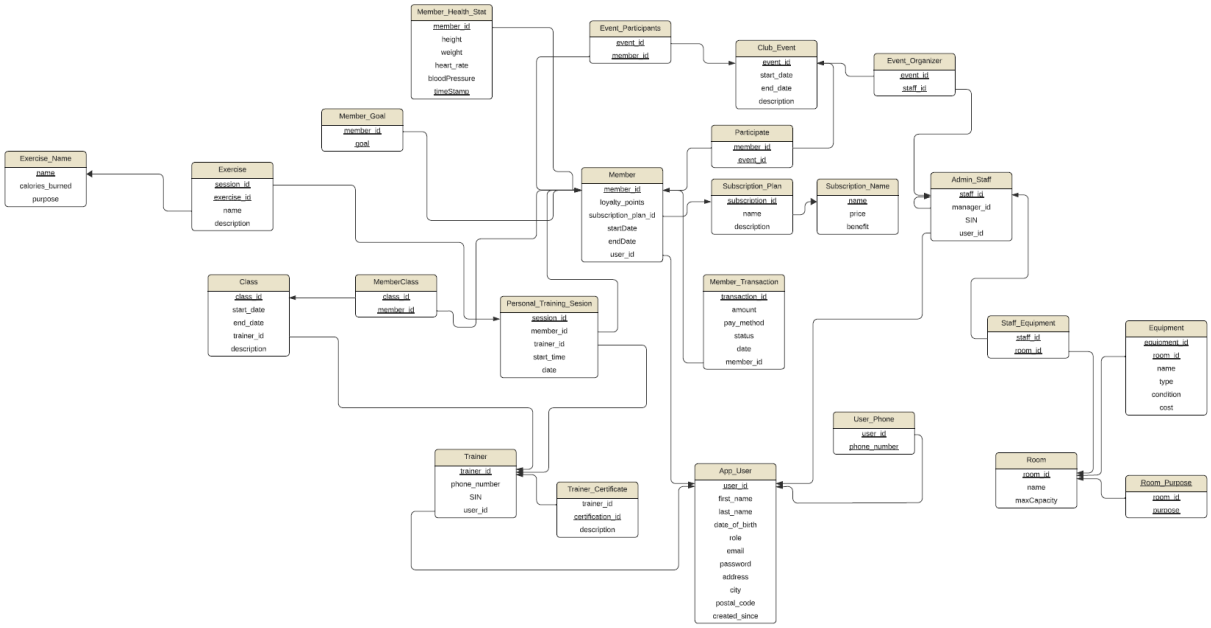
     Subscription_Plan: subscription_id -> name, description

     Subscription_Name: name -> {price, benefit}

     New relation: name -> {price, benefit}

For all other tables, they are already in 3NF, as none of the attributes in the tables depend on other non-key attributes.

V. Redrawing the database structure



VI. Writing DDL commands

```
DROP TABLE IF EXISTS Exercise;

DROP TABLE IF EXISTS Exercise_Name;

DROP TABLE IF EXISTS Personal_Training_Session;

DROP TABLE IF EXISTS Participate;

DROP TABLE IF EXISTS Event_Organizer;

DROP TABLE IF EXISTS Event_Participant;

DROP TABLE IF EXISTS Member_Health_Stat;

DROP TABLE IF EXISTS Club_Event;

DROP TABLE IF EXISTS Member_Goal;

DROP TABLE IF EXISTS Member_Class;

DROP TABLE IF EXISTS Class;

DROP TABLE IF EXISTS Member_Transaction;

DROP TABLE IF EXISTS Staff_Equipment;

DROP TABLE IF EXISTS Admin_Staff;

DROP TABLE IF EXISTS Trainer_Certificate;

DROP TABLE IF EXISTS Trainer;

DROP TABLE IF EXISTS Equipment;

DROP TABLE IF EXISTS Room_Purpose;

DROP TABLE IF EXISTS Room;

DROP TABLE IF EXISTS Member;

DROP TABLE IF EXISTS Subscription_Plan;

DROP TABLE IF EXISTS Subscription_Name;

DROP TABLE IF EXISTS User_Phone;

DROP TABLE IF EXISTS App_User;
```

```
CREATE TABLE App_User(
    user_id SERIAL PRIMARY KEY,
    first_name VARCHAR(255) NOT NULL,
    last_name VARCHAR(255) NOT NULL,
    date_of_birth DATE,
    role VARCHAR(255) NOT NULL,
    email VARCHAR(255) NOT NULL UNIQUE,
    password VARCHAR(255) NOT NULL,
    address VARCHAR(255),
    city VARCHAR(255),
    postal_code VARCHAR(255),
    created_since timestamp DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE User_Phone(
    user_id INT,
    phone_number VARCHAR(10),
    PRIMARY KEY (user_id, phone_number),
    FOREIGN KEY (user_id) REFERENCES App_User(user_id)
);

CREATE TABLE Subscription_Name(
    name VARCHAR(255) PRIMARY KEY,
    price FLOAT(2) NOT NULL,
    benefit TEXT NOT NULL
```

```
);

CREATE TABLE Subscription_Plan(
    subscription_id SERIAL PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    description TEXT,
    FOREIGN KEY (name) REFERENCES Subscription_Name(name)
);

CREATE TABLE Member(
    member_id SERIAL PRIMARY KEY,
    loytalty_points INT,
    subscription_id INT,
    start_date DATE,
    end_date DATE,
    user_id INT,
    FOREIGN KEY (user_id) REFERENCES App_User(user_id)
);

CREATE TABLE Room(
    room_id SERIAL PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    maxCapacity INT NOT NULL
);

CREATE TABLE Room_Purpose(
    room_id INT ,
```

```
    purpose VARCHAR(255) NOT NULL,
    PRIMARY KEY (room_id, purpose),
    FOREIGN KEY (room_id) REFERENCES Room(room_id)
);

CREATE TABLE Equipment(
    equipment_id SERIAL,
    room_id INT,
    name VARCHAR(255) NOT NULL,
    type VARCHAR(255) NOT NULL,
    condition VARCHAR(255) NOT NULL,
    cost FLOAT(2) NOT NULL
);

CREATE TABLE Trainer(
    trainer_id SERIAL PRIMARY KEY,
    user_id INT,
    SIN VARCHAR(9) NOT NULL,
    FOREIGN KEY (user_id) REFERENCES App_User(user_id)
);

CREATE TABLE Trainer_Certificate(
    trainer_id INT,
    certificate_id SERIAL,
    description TEXT,
    PRIMARY KEY (certificate_id),
    FOREIGN KEY (trainer_id) REFERENCES Trainer(trainer_id)
```

```
);

CREATE TABLE Admin_Staff(
    staff_id SERIAL PRIMARY KEY,
    manager_id INT,
    user_id INT,
    SIN VARCHAR(9) NOT NULL,
    FOREIGN KEY (user_id) REFERENCES App_User(user_id),
    FOREIGN KEY (manager_id) REFERENCES Admin_Staff(staff_id)
);

CREATE TABLE Staff_Equipment(
    staff_id INT,
    room_id INT,
    PRIMARY KEY (staff_id, room_id),
    FOREIGN KEY (staff_id) REFERENCES Admin_Staff(staff_id),
    FOREIGN KEY (room_id) REFERENCES Room(room_id)
);

CREATE TABLE Member_Transaction(
    transaction_id SERIAL PRIMARY KEY,
    amount FLOAT(2) NOT NULL,
    pay_method VARCHAR(255) NOT NULL,
    status VARCHAR(255) NOT NULL,
    date DATE NOT NULL,
    member_id INT
);
```

```
CREATE TABLE Class(
    class_id SERIAL PRIMARY KEY,
    start_date DATE NOT NULL,
    end_date DATE NOT NULL,
    trainer_id INT,
    description TEXT,
    FOREIGN KEY (trainer_id) REFERENCES Trainer(trainer_id)
);


CREATE TABLE Member_Class(
    class_id INT,
    member_id INT,
    PRIMARY KEY (class_id, member_id),
    FOREIGN KEY (class_id) REFERENCES Class(class_id),
    FOREIGN KEY (member_id) REFERENCES Member(member_id)
);

CREATE TABLE Member_Goal(
    member_id INT,
    goal_id TEXT,
    PRIMARY KEY (member_id, goal_id),
    FOREIGN KEY (member_id) REFERENCES Member(member_id)
);

CREATE TABLE Club_Event(
```

```
    event_id SERIAL PRIMARY KEY,
    start_date DATE NOT NULL,
    end_date DATE NOT NULL,
    description TEXT
);

CREATE TABLE Member_Health_Stat(
    member_id INT,
    height FLOAT(2) NOT NULL,
    weight FLOAT(2) NOT NULL,
    heart_rate INT NOT NULL,
    blood_pressure INT NOT NULL,
    timestamp timestamp DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY (member_id, timestamp)
);

CREATE TABLE Event_Participant(
    event_id INT,
    member_id INT,
    PRIMARY KEY (event_id, member_id),
    FOREIGN KEY (event_id) REFERENCES Club_Event(event_id),
    FOREIGN KEY (member_id) REFERENCES Member(member_id)
);

CREATE TABLE Event_Organizer(
    event_id INT,
    staff_id INT,
```

```
    PRIMARY KEY (event_id, staff_id),
    FOREIGN KEY (event_id) REFERENCES Club_Event(event_id),
    FOREIGN KEY (staff_id) REFERENCES Admin_Staff(staff_id)
);

CREATE TABLE Participate(
   member_id INT,
   event_id INT,
    PRIMARY KEY (member_id, event_id),
    FOREIGN KEY (member_id) REFERENCES Member(member_id),
    FOREIGN KEY (event_id) REFERENCES Club_Event(event_id)
);

CREATE TABLE Personal_Training_Session(
    session_id SERIAL PRIMARY KEY,
    member_id INT,
    trainer_id INT,
    start_time TIME NOT NULL,
    date DATE NOT NULL,

    FOREIGN KEY (member_id) REFERENCES Member(member_id),
    FOREIGN KEY (trainer_id) REFERENCES Trainer(trainer_id)
);

CREATE TABLE Exercise_Name(
    name VARCHAR(255) PRIMARY KEY,
    calories_burned FLOAT(2) NOT NULL,
```

```
    purpose TEXT NOT NULL
);

CREATE TABLE Exercise(
    session_id INT,
    exercise_id INT,
    name VARCHAR(255) NOT NULL,
    duration_in_seconds INT,
    FOREIGN KEY (name) REFERENCES Exercise_Name(name)
);
```

VII. Writing DML commands

```
TRUNCATE TABLE Exercise CASCADE;
TRUNCATE TABLE Exercise_Name CASCADE;
TRUNCATE TABLE Personal_Training_Session CASCADE;
TRUNCATE TABLE Participate CASCADE;
TRUNCATE TABLE Event_Organizer CASCADE;
TRUNCATE TABLE Event_Participant CASCADE;
TRUNCATE TABLE Member_Health_Stat CASCADE;
TRUNCATE TABLE Club_Event CASCADE;
TRUNCATE TABLE Member_Goal CASCADE;
TRUNCATE TABLE Member_Class CASCADE;
TRUNCATE TABLE Class CASCADE;
TRUNCATE TABLE Member_Transaction CASCADE;
TRUNCATE TABLE Staff_Equipment CASCADE;
TRUNCATE TABLE Admin_Staff CASCADE;
TRUNCATE TABLE Trainer_Certificate CASCADE;
```

```
TRUNCATE TABLE Trainer CASCADE;

TRUNCATE TABLE Equipment CASCADE;

TRUNCATE TABLE Room_Purpose CASCADE;

TRUNCATE TABLE Room CASCADE;

TRUNCATE TABLE Member CASCADE;

TRUNCATE TABLE Subscription_Plan CASCADE;

TRUNCATE TABLE Subscription_Name CASCADE;

TRUNCATE TABLE User_Phone CASCADE;

TRUNCATE TABLE App_User CASCADE;


INSERT INTO App_User (first_name, last_name, date_of_birth, role, email, password,
address, city, postal_code)

VALUES

   ('Tung', 'Tran', DATE '2077-05-16', 'Staff', 'tung@tran.com', '123456789','1234 Five St',
'City', 'N7T8N9'),

   ('Truc', 'Le', DATE '2021-05-14', 'Staff', 'truc@le.com', '456789123','5678 Fue St', 'City',
'E7U3O5'),

   ('Anthony', 'Lincoln', DATE '1875-06-09', 'Staff', 'anthony.lincoln@example.com',
'anthony123', '987 Lark St', 'City', '12345'),

   ('Alina', 'Mann', DATE '1798-03-15', 'Staff', 'alina.mann@example.com', 'alina123', '789
Odd St', 'City', '67890'),

   ('Matthew', 'Taylor', DATE '1987-07-20', 'Staff', 'matthew.taylor@example.com',
'passwordjkl', '741 Cedar St', 'City', '97531')

;


INSERT INTO App_User (first_name, last_name, date_of_birth, role, email, password,
address, city, postal_code)

VALUES

   ('John', 'Doe', DATE '1990-01-01', 'Member', 'john.doe@example.com', 'password123',
'123 Main St', 'City', '12345'),
```

```
    ('Jane', 'Smith', DATE '1995-02-15', 'Member', 'jane.smith@example.com', 'password456',
'456 Elm St', 'City', '67890'),

    ('Michael', 'Johnson', DATE '1985-03-30', 'Member', 'michael.johnson@example.com',
'password789', '789 Oak St', 'City', '54321'),

    ('Emily', 'Brown', DATE '1992-04-10', 'Member', 'emily.brown@example.com',
'passwordabc', '321 Pine St', 'City', '09876'),

    ('David', 'Wilson', DATE '1988-05-25', 'Member', 'david.wilson@example.com',
'passworddef', '654 Maple St', 'City', '13579')

;


INSERT INTO App_User (first_name, last_name, date_of_birth, role, email, password,
address, city, postal_code)

VALUES

    ('Jew', 'Trainer', DATE '1990-01-01', 'Trainer', 'jew.trainer@example.com', 'password123',
'123 Main St', 'City', '12345'),

    ('Doe', 'Trainer', DATE '1995-02-15', 'Trainer', 'doe.trainer@example.com', 'password456',
'456 Elm St', 'City', '67890'),

    ('Olivia', 'Martinez', DATE '1994-08-12', 'Trainer', 'olivia.martinez@example.com',
'passwordmno', '852 Walnut St', 'City', '86420'),

    ('Daniel', 'Harris', DATE '1989-09-28', 'Trainer', 'daniel.harris@example.com',
'passwordpqr', '963 Pineapple St', 'City', '75319'),

    ('Sophia', 'Clark', DATE '1991-10-08', 'Trainer', 'sophia.clark@example.com', 'passwordstu',
'159 Orange St', 'City', '95173')

;


INSERT INTO User_Phone (user_id, phone_number) VALUES

(1, '1234567890'),

(2, '1234567891'),

(3, '1234567892'),

(4, '1234567893'),

(5, '1234567894'),
```

```
(6, '1234567895'),
(7, '1234567896'),
(8, '1234567897'),
(9, '1234567898'),
(10, '1234567899'),
(11, '1234567809'),
(12, '1234567819'),
(13, '1234567829'),
(14, '1234567839'),
(15, '1234567849')
;

INSERT INTO Subscription_Name (name, price, benefit) VALUES
('Basic', 19.99, 'Access to gym and basic classes'),
('Premium', 29.99, 'Access to gym, all classes, and sauna'),
('Gold', 39.99, 'All club facilities with free personal training session'),
('Platinum', 49.99, 'All club facilities, unlimited personal training, and nutrition plans'),
('Diamond', 59.99, 'All club benefits plus guest access and special discounts')
;

INSERT INTO Subscription_Plan (name, description) VALUES
('Basic', 'This is a basic subscription plan'),
('Premium', 'This is a premium subscription plan'),
('Gold', 'This is a gold subscription plan'),
('Platinum', 'This is a platinum subscription plan'),
('Diamond', 'This is a diamond subscription plan')
;
```

```
INSERT INTO Member (loytalty_points, subscription_id, start_date, end_date, user_id)
VALUES

(100, 1, '2021-01-01', '2022-01-01', 6),

(200, 2, '2021-02-01', '2022-02-01', 7),

(300, 3, '2021-03-01', '2022-03-01', 8),

(400, 4, '2021-04-01', '2022-04-01', 9),

(500, 5, '2021-05-01', '2022-05-01', 10)

;


INSERT INTO Room (name, maxCapacity) VALUES

('Aerobics Room', 25),

('Spin Room', 20),

('Yoga Studio', 15),

('Weight Room', 50),

('Cardio Room', 30)

;


INSERT INTO Room_Purpose (room_id, purpose) VALUES

(1, 'Group Fitness Classes'),

(2, 'Cycling Classes'),

(3, 'Yoga and Meditation'),

(4, 'Strength Training'),

(5, 'Treadmills and Ellipticals')

;


INSERT INTO Equipment (room_id, name, type, condition, cost) VALUES
```

```
(1, 'Treadmill', 'Cardio', 'New', 6000.00),
(2, 'Stationary Bike', 'Cardio', 'Used', 1500.00),
(3, 'Yoga Mat', 'Accessory', 'New', 50.00),
(4, 'Dumbbell Set', 'Weights', 'Used', 750.00),
(5, 'Rowing Machine', 'Cardio', 'New', 1200.00)
;

INSERT INTO Trainer (user_id, SIN) VALUES
(11, '123456789'),
(12, '987654321'),
(13, '234567891'),
(14, '876543219'),
(15, '345678912')
;

INSERT INTO Trainer_Certificate (trainer_id, description) VALUES
(1, 'Certified Personal Trainer'),
(2, 'Certified Nutrition Specialist'),
(3, 'Certified Yoga Instructor'),
(4, 'Certified Strength and Conditioning Specialist'),
(5, 'Certified Group Fitness Instructor')
;

INSERT INTO Admin_Staff (manager_id, user_id, SIN) VALUES
(NULL, 3, '123123123'),
(1, 4, '321321321'),
(1, 2, '213213213'),
```

```
(2, 1, '132132132'),
(3, 5, '231231231')
;

INSERT INTO Staff_Equipment (staff_id, room_id) VALUES
(1, 1),
(2, 2),
(3, 3),
(4, 4),
(5, 5)
;

INSERT INTO Member_Transaction (amount, pay_method, status, date, member_id) VALUES
(19.99, 'Credit Card', 'Completed', '2021-01-01', 1),
(29.99, 'Debit Card', 'Completed', '2021-02-01', 2),
(39.99, 'Credit Card', 'Pending', '2021-03-01', 3),
(49.99, 'Cash', 'Completed', '2021-04-01', 4),
(59.99, 'Check', 'Cancelled', '2021-05-01', 5)
;

INSERT INTO Class (start_date, end_date, trainer_id, description) VALUES
('2021-06-01', '2021-06-30', 1, 'Body Pump'),
('2021-07-01', '2021-07-31', 2, 'Cycling'),
('2021-08-01', '2021-08-31', 3, 'HIIT'),
('2021-09-01', '2021-09-30', 4, 'Yoga'),
('2021-10-01', '2021-10-31', 5, 'Zumba')
;
```

```
INSERT INTO Member_Class (class_id, member_id) VALUES
(1, 1),
(2, 2),
(3, 3),
(4, 4),
(5, 5)
;

INSERT INTO Member_Goal (member_id, goal_id) VALUES
(1, 'Weight Loss'),
(2, 'Muscle Gain'),
(3, 'Flexibility'),
(4, 'Endurance'),
(5, 'Wellness')
;

INSERT INTO Club_Event (start_date, end_date, description) VALUES
('2021-11-01', '2021-11-02', 'Marathon Prep Workshop'),
('2021-12-01', '2021-12-02', 'Healthy Eating Seminar'),
('2022-01-01', '2022-01-02', 'New Year Fitness Challenge'),
('2022-02-01', '2022-02-02', 'Valentine Day Couples Yoga'),
('2022-03-01', '2022-03-02', 'Spring Into Fitness Bootcamp')
;

INSERT INTO Member_Health_Stat (member_id, height, weight, heart_rate,
blood_pressure) VALUES
```

(1, 1.75, 75.0, 60, 120),

(2, 1.80, 80.0, 65, 121),

(3, 1.65, 65.0, 70, 122),

(4, 1.70, 70.0, 75, 123),

(5, 1.85, 85.0, 80, 124)

;

INSERT INTO Event_Participant (event_id, member_id) VALUES

(1, 1),

(2, 2),

(3, 3),

(4, 4),

(5, 5)

;

INSERT INTO Event_Organizer (event_id, staff_id) VALUES

(1, 1),

(2, 2),

(3, 3),

(4, 4),

(5, 5)

;

INSERT INTO Participate (member_id, event_id) VALUES

(1, 1),

(2, 2),

(3, 3),

```
(4, 4),
(5, 5)
;

INSERT INTO Personal_Training_Session(member_id, trainer_id, start_time, date)
VALUES
   (1, 1, TIME '10:00:00', DATE '2021-05-14'),
   (2, 1, TIME '11:00:00', DATE '2021-05-14'),
   (3, 1, TIME '12:00:00', DATE '2021-05-14'),
   (4, 1, TIME '13:00:00', DATE '2021-05-14'),
   (5, 1, TIME '14:00:00', DATE '2021-05-14')
;

INSERT INTO Exercise_Name (name, calories_burned, purpose) VALUES
('Push-up', 100, 'Strength'),
('Sit-up', 150, 'Core Stability'),
('Squat', 200, 'Lower Body Strength'),
('Burpee', 250, 'Full Body Conditioning'),
('Plank', 50, 'Core Endurance')
;

INSERT INTO Exercise (session_id, exercise_id, name, duration_in_seconds) VALUES
(1, 1, 'Push-up', 60),
(2, 2, 'Sit-up', 90),
(3, 3, 'Squat', 120),
(4, 4, 'Burpee', 30),
(5, 5, 'Plank', 180)
```

;

VIII. Writing SQL commands for testing

```
-- Retrieve all staffs and their details
SELECT
   au.user_id,
   au.first_name,
   au.last_name,
   ast.staff_id,
   ast.SIN,
   ast.manager_id
FROM App_User au
JOIN Admin_Staff ast ON au.user_id = ast.user_id;

-- Retrieve all members and their details
SELECT
   au.user_id,
   au.first_name,
   au.last_name,
   m.member_id,
   m.loytalty_points,
   m.subscription_id
FROM App_User au
JOIN Member m ON au.user_id = m.user_id;

-- Retrieve all trainers and their details
SELECT
```

```
    au.user_id,
    au.first_name,
    au.last_name,
    t.trainer_id,
    t.SIN
FROM App_User au
JOIN Trainer t ON au.user_id = t.user_id;

-- List all personal training sessions with their corresponding member and trainer details
SELECT
    p.session_id,
    p.date,
    m.member_id,
    t.trainer_id
FROM Personal_Training_Session p
JOIN Member m ON p.member_id = m.member_id
JOIN Trainer t ON p.trainer_id = t.trainer_id;

-- Retrieve all Personal Training Sessions and the Exercises included in each session
SELECT
    pts.session_id,
    pts.start_time,
    pts.date,
    e.name,
    e.duration_in_seconds,
    en.calories_burned,
    en.purpose
```

```
FROM Personal_Training_Session pts
LEFT JOIN Exercise e ON pts.session_id = e.session_id
LEFT JOIN Exercise_Name en ON e.name = en.name;

-- Retrieve all classes along with the assigned trainer's information
SELECT
   c.class_id,
   c.description,
   t.trainer_id,
   t.SIN
FROM Class c
JOIN Trainer t ON c.trainer_id=t.trainer_id;

-- Retrieve all Equipment and their respective room destails
SELECT
   e.equipment_id,
   e.name,
   e.type,
   e.condition,
   e.cost,
   r.name AS room_name,
   r.maxCapacity
FROM Equipment e
LEFT JOIN Room r ON e.room_id = r.room_id;

-- Retrieve all Events and their participants
SELECT
```

```
    ev.event_id,
    ev.description,
    ep.member_id
FROM Club_Event ev
LEFT JOIN Event_Participant ep ON ev.event_id = ep.event_id;

-- Get the phone numbers of all app users
SELECT
    a.first_name,
    a.last_name,
    u.phone_number
FROM App_User a
JOIN User_Phone u ON a.user_id = u.user_id;

-- Count the number of members subscribed to each subscription plan
SELECT
    s.name,
    COUNT(m.subscription_id) AS Number_Of_Members
FROM Subscription_Plan s
JOIN Member m ON s.subscription_id = m.subscription_id
GROUP by s.name;

-- Find all trainers and the number of personal training sessions they have conducted
SELECT
    t.trainer_id,
    COUNT(p.session_id) AS Number_Of_Sessions
FROM Trainer t
```

```
LEFT JOIN Personal_Training_Session p ON t.trainer_id = p.trainer_id
GROUP BY t.trainer_id;

-- Select all members who have not participated in any event
SELECT
   m.member_id
FROM Member m
LEFT JOIN Event_Participant e ON m.member_id = e.member_id
WHERE e.event_id IS NULL;

-- Retrieve the latest health statistics for each member
SELECT
   member_id,
   MAX(timestamp) AS Lastest_Time
FROM Member_Health_Stat
GROUP BY member_id;

-- Find the total revenue generated from the subscription plans
SELECT
   SUM(t.amount) AS Total_Revenue
FROM Member_Transaction t

-- List all exercise names along with their calories burned and purpose
SELECT
   e.name,
   e.calories_burned,
   e.purpose
```

```
FROM Exercise_Name e;
```