Project: Health and Fitness Club Management System

COMP 3005 – Database Management Systems
Carleton University

By: Rama Alkhouli(101198025)

2.1 Conceptual Design

Assumptions regarding cardinalities and participations::

Members Entity:

1. Each member must have a PersonalizedDashboard where they can access their exercise routines, fitness goals, achievement, health metrics and health statistics.Every member has 1 PersonalizedDashboardand each PersonalizedDashboard is made for a member, hence the 1 to 1 relationship and the Total participation from both sides
2. All trainers can view registered members information
3. A member can take one personal training session during a certain time slot.
4. All members have to pay Gym Membership fees, hence why the participation is total
5. Exercise routines are already populated by third party software when member registers with the app
6. A member can subscribe in many classes but does not necessarily have to subscribe in a class hence the partial participation

Trainer Entity:

1. A member can only be coached by 1 personal trainer, but a trainer can coach multiple members
2. Every training session has to have a Trainer to train the session, hence the total participation between PersonalTrainingSession and Trains
3. Not every Trainer is necessarily training a PersonalTrainingSession, hence the partial participation between Trainer and Trains.
4. A Trainer can give multiple classes but not necessarily has to give a class hence the partial participation
5. A class must only have one Trainer Giving it hence the total participation and the one 1 to many relationship
6. Trainers can view all registered members profiles but they don't necessarily have to do it, hence partial participation.
7. Assume Trainers availability is for all business days of the week. If the trainer chooses 9 am, then it means the trainer is available at 9 am from Monday to Sunday.

Administrative Staff:

1. Assume administrative staff issues bills for members

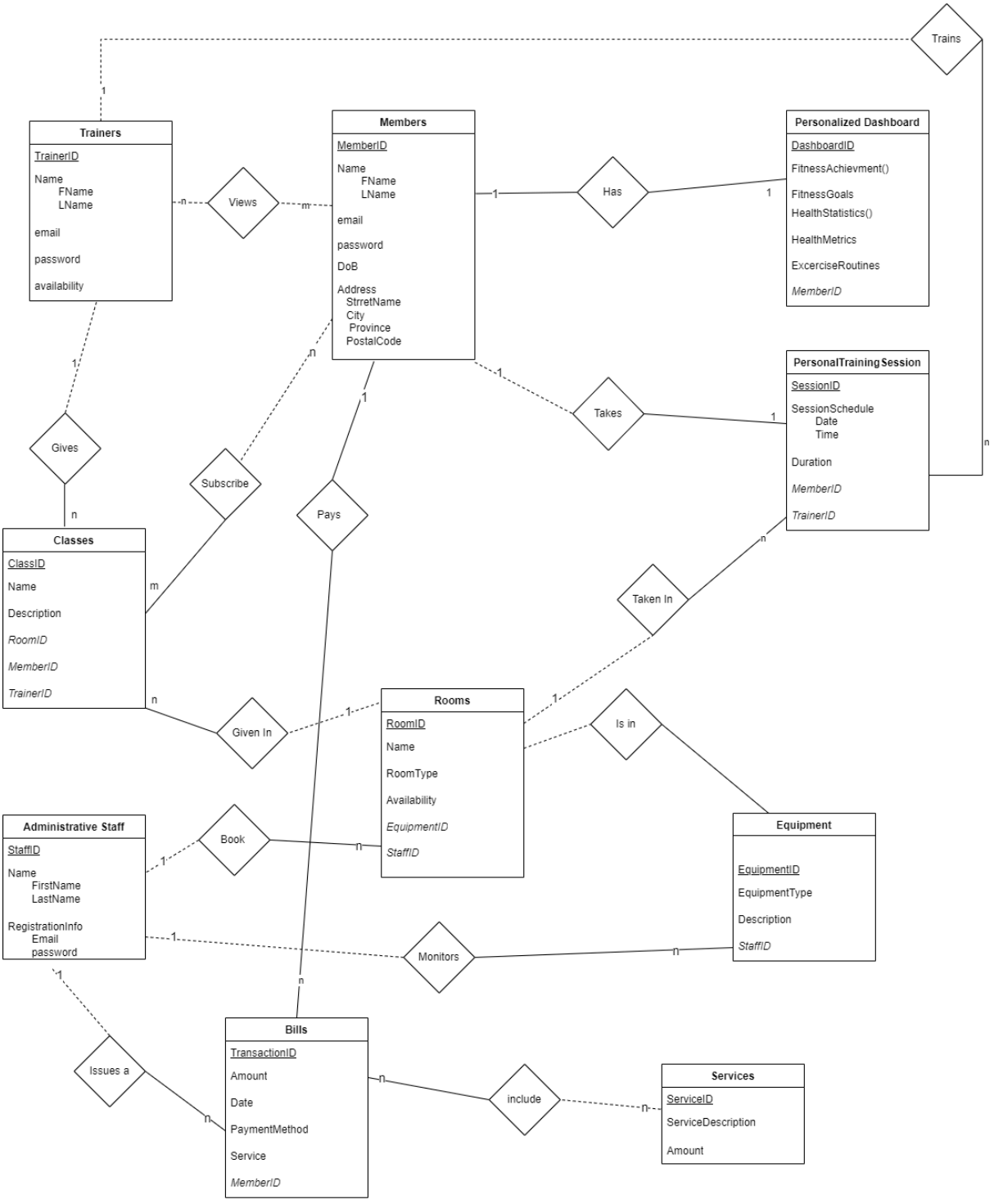
Others:

1. PaymentMethod in Billing can be either money (credit card, PayPal or ApplePay)
2. Classes must be given in a room
3. A PersonalTrainingSession can only have 1 coach to train it
4. A class can have many members subscribed in and needs at least 1 member to be created, hence the full participation between class and subscribe
5. Since a class can have many members subscribe in it and a member can subscribe in multiple classes, hence a many to many relationship
6. A room can have multiple classes given in it (at different times) while a class can be given in only 1 room, hence the 1 to many relationship and each class needs to be given in a room, hence the full participation but not all rooms have to have a class given in it. hence the partial participation. Same logic applies between room and personal Trainings sessions

7. Fitness classes and rooms have no capacity, any number of members can register to a class and can fit in a room

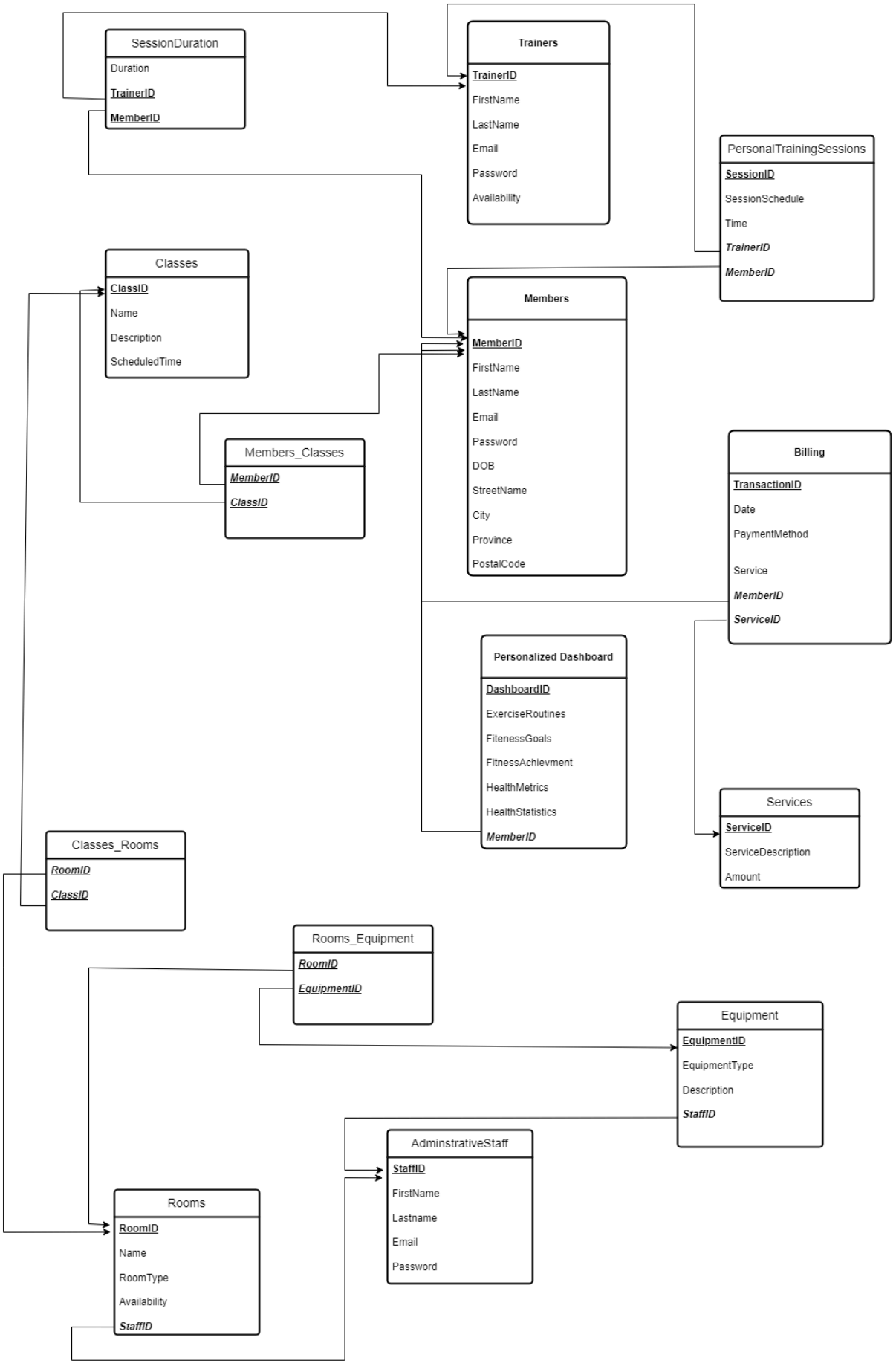
ER Diagram:

Dashed line represents partial participation
Solid line represents total participation
Underlined -> primary key
Italicized -> foreign key



2.2 Reduction to Relational Schemas

Underlined: Primary Key
Italic: Foreign Key


SessionDuration
Duration
TrainerID
MemberID
Trainers
TrainerID
FirstName
LastName
Email
Password
Availability
PersonalTrainingSessions
SessionID
SessionSchedule
Time
TrainerID
MemberID
Classes
ClassID
Name
Description
ScheduledTime
Members
MemberID
FirstName
LastName
Email
Password
DOB
StreetName
City
Province
PostalCode
Members_Classes
MemberID
ClassID
Billing
TransactionID
Date
PaymentMethod
Service
MemberID
ServiceID
Personalized Dashboard
DashboardID
ExerciseRoutines
FitenessGoals
FitnessAchievment
HealthMetrics
HealthStatistics
MemberID
Services
ServiceID
ServiceDescription
Amount
Classes_Rooms
RoomID
ClassID
Rooms_Equipment
RoomID
EquipmentID
Equipment
EquipmentID
EquipmentType
Description
StaffID
AdminstrativeStaff
StaffID
FirstName
Lastname
Email
Password
Rooms
RoomID
Name
RoomType
Availability
StaffID

2.3 DDL File

Transform your Relational Database Schema into a data definition language (DDL) statements file with a '.sql' extension.

```
-- Table for Members
CREATE TABLE Members (
   MemberID SERIAL PRIMARY KEY,
   FirstName VARCHAR(255),
   LastName VARCHAR(255),
   Email VARCHAR(255) UNIQUE,
   Password VARCHAR(255),
   DoB TEXT,
   StreetName VARCHAR(255),
   City VARCHAR(255),
   Province VARCHAR(255),
   PostalCode VARCHAR(255)
);


-- Table for Personalized Dashboard
CREATE TABLE PersonalizedDashboards (
   DashboardID SERIAL PRIMARY KEY,
   ExerciseRoutine TEXT,
   FitnessGoals TEXT,
   FitnessAchievements Text,
   HealthMetrics DOUBLE PRECISION,
   HealthStatistics TEXT,
   MemberID INT UNIQUE,
   FOREIGN KEY (MemberID) REFERENCES Members(MemberID)
);


- - Table for Trainers
CREATE TABLE Trainers (
   TrainerID INT AUTO_INCREMENT PRIMARY KEY,
   FirstName VARCHAR(255),
   LastName VARCHAR(255),
   Email VARCHAR(255) UNIQUE,
   Password VARCHAR(255),
   Availability TEXT
);


-- Table for Classes
CREATE TABLE Classes (
   ClassID SERIAL PRIMARY KEY,
   Name VARCHAR(255),
   Description TEXT,
   ScheduledTime Time
);
```

```sql
-- Table for Administrative Staff
CREATE TABLE AdministrativeStaff (
   StaffID SERIAL PRIMARY KEY,
   FirstName VARCHAR(255),
   LastName VARCHAR(255),
   Email VARCHAR(255) UNIQUE,
   Password VARCHAR(255)
);

-- Table for Rooms
CREATE TABLE Rooms (
   RoomID SERIAL PRIMARY KEY,
   Name VARCHAR(255),
   RoomType VARCHAR(255),
   Availability BOOLEAN,
   StaffID INT,
   FOREIGN KEY (StaffID) REFERENCES AdministrativeStaff(StaffID)
);

-- Table for Equipment
CREATE TABLE Equipment (
   EquipmentID SERIAL PRIMARY KEY,
   EquipmentType VARCHAR(255),
   Description TEXT,
   StaffID INT,
   FOREIGN KEY (StaffID) REFERENCES AdministrativeStaff(StaffID)
);



-- Table for Personal Training Sessions
CREATE TABLE PersonalTrainingSessions (
   SessionID SERIAL PRIMARY KEY,
   SessionSchedule DATE,
   Time TIME,
   MemberID INT,
   FOREIGN KEY (MemberID) REFERENCES Members(MemberID),
   TrainerID INT,
   FOREIGN KEY (TrainerID) REFERENCES Trainers(TrainerID)
);

-- Table for Services
CREATE TABLE Services (
   ServiceID SERIAL PRIMARY KEY,
   ServiceDescription VARCHAR(255),
   Amount INT
);

-- Table for Billing
CREATE TABLE Billing (
   TransactionID SERIAL PRIMARY KEY,
   Date DATE,
   PaymentMethod VARCHAR(255),
```

```
    Service VARCHAR(255),
    MemberID INT,
    FOREIGN KEY (MemberID) REFERENCES Members(MemberID),
    ServiceID INT,
    FOREIGN KEY (ServiceID) REFERENCES Services(ServiceID)
);

-- Table for SessionDuration
CREATE TABLE SessionDuration (
    Duration INT,
    MemberID INT,
    FOREIGN KEY (MemberID) REFERENCES Members(MemberID),
    TrainerID INT,
    FOREIGN KEY (TrainerID) REFERENCES Trainers(TrainerID)
);

-- Associative table for the many-to-many relationship between Members and
Classes
CREATE TABLE Members_Classes (
    MemberID INT,
    ClassID INT,
    PRIMARY KEY (MemberID, ClassID),
    FOREIGN KEY (MemberID) REFERENCES Members(MemberID),
    FOREIGN KEY (ClassID) REFERENCES Classes(ClassID)
);

-- Associative table for the relationship between Classes and Rooms
CREATE TABLE Classes_Rooms (
    ClassID INT,
    RoomID INT,
    PRIMARY KEY (ClassID, RoomID),
    FOREIGN KEY (ClassID) REFERENCES Classes(ClassID),
    FOREIGN KEY (RoomID) REFERENCES Rooms(RoomID)
);

-- Associative table for the relationship between Rooms and Equipment
CREATE TABLE Rooms_Equipment (
    RoomID INT,
    EquipmentID INT,
    PRIMARY KEY (RoomID, EquipmentID),
    FOREIGN KEY (RoomID) REFERENCES Rooms(RoomID),
    FOREIGN KEY (EquipmentID) REFERENCES Equipment(EquipmentID)
);
```

2.4 DML File

The DML sql file is in a folder called SQL in the project folder. The file contains the following queries.

```
----- Populating Tables -----------

-- Populate the Members table
```

```
INSERT INTO Members (FirstName, LastName, Email, Password, DoB, StreetName,
City, Province, PostalCode)
VALUES
('SpongeBob', 'SquarePants', 'spongebob@bikinibottom.net', 'pineapple123',
'1986-07-14', '124 Conch Street', 'Bikini Bottom', 'Ocean', '12345'),
('Patrick', 'Star', 'patrick@bikinibottom.net', 'rockstar123', '1984-02-26',
'120 Conch Street', 'Bikini Bottom', 'Ocean', '12345'),
('Squidward', 'Tentacles', 'squidward@bikinibottom.net', 'clarinet123',
'1972-10-09', '122 Conch Street', 'Bikini Bottom', 'Ocean', '12345');

-- Populate the Trainers table
INSERT INTO Trainers (FirstName, LastName, Email, Password, Availability)
VALUES
('Sandy', 'Cheeks', 'sandy@bikinibottom.net', 'karate123', '{"09:00",
"11:00", "15:00"}'),
('Larry', 'Lobster', 'larry@bikinibottom.net', 'weights123', '{"10:00",
"12:00", "16:00"}');

-- Populate the AdministrativeStaff table
INSERT INTO AdministrativeStaff (FirstName, LastName, Email, Password)
VALUES
('Pearl', 'Krabs', 'pearl@krustykrab.net', 'shopping123'),
('Karen', 'Plankton', 'karen@chum.bucket', 'computer123');

-- Populate the Classes table
INSERT INTO Classes (Name, Description, ScheduledTime)
VALUES
('Spinning', 'Intense spinning class to boost your endurance.', '09:00'),
('Yoga', 'Relaxing yoga session to improve flexibility.', '10:00'),
('Pilates', 'A class aimed at strengthening the body with an emphasis on core
strength.', '11:00'),
('Boxing', 'High-intensity class focusing on strength, cardio, and
technique.', '12:00'),
('Kickboxing', 'A class combining boxing with elements of karate, especially
kicks.', '16:00');

-- Populate the Rooms table
INSERT INTO Rooms (Name, RoomType, Availability, StaffID)
VALUES
('Room A', 'Spinning', TRUE, 1),
('Room B', 'Yoga', TRUE, 2);

-- Populate the Equipment table
INSERT INTO Equipment (EquipmentType, Description, StaffID)
VALUES
('Treadmill', 'High-speed treadmill for cardio workouts.', 1),
('Yoga Mats', 'Eco-friendly yoga mats.', 2),
('Elliptical', 'Well-maintained and functional', 1),
('Stationary Bike', 'Newly serviced, works perfectly', 1),
('Dumbbells', 'Complete set, varying weights, all in good condition', 1),
('Kettlebells', 'Recently acquired, no signs of wear', 1);

-- populate Services Table
```

```sql
INSERT INTO Services (ServiceDescription, Amount) VALUES
('Gym Membership - Monthly', 89),
('Personal Training - Single Session', 30),
('Yoga Class - Drop-in', 25),
('Boxing Workshop', 25),
('Pilates Class Package', 100),
('Nutrition Consultation', 80),
('Health Assessment', 60);

-- populate PersonalizedDashboards Table
INSERT INTO PersonalizedDashboards (ExerciseRoutine, FitnessGoals,
FitnessAchievements, HealthMetrics, HealthStatistics, MemberID)
VALUES ('Monday: Cardio, Tuesday: Arms, Wednesday: Abs and Obliques,
Thursday: Lower Body, Friday: Cardio, Saturday and Sunday: Rest', 'Run a
marathon', 'Goal is in progress', 75.0, 'No data yet', 1);


---- Sample Queries used in the project -----------

--  register a new member
INSERT INTO Members (FirstName, LastName, Email, Password, DoB, StreetName,
City, Province, PostalCode)
VALUES ('New', 'Member', 'newmember@bikinibottom.net', 'newpassword123',
'1990-01-01', '100 New Street', 'Bikini Bottom', 'Ocean', '12349');

-- update a member's profile information
UPDATE Members
SET Email = 'updatedemail@bikinibottom.net', City = 'New Bikini Bottom'
WHERE MemberID = 1;

-- update a member’s fitness goals in the PersonalizedDashboards table
UPDATE PersonalizedDashboards
SET FitnessGoals = 'Run a 10 km marathon'
WHERE MemberID = 1;

-- Update Health Metrics in PersonalizedDashboards
UPDATE PersonalizedDashboards
SET HealthMetrics = 70.0
WHERE DashboardID = 1;

-- Update Member's ExcerciseRoutine. Assume the routine is generated by third
part service
UPDATE PersonalizedDashboards
SET ExerciseRoutine = 'Monday: Cardio, Tuesday: Arms, Wednesday: Abs and
Obliques, Thursday: upper Body, Friday: Cardio & swimming, Saturday and
Sunday: Rest'
WHERE DashboardID = 1;

-- Display member's dashboard
SELECT * FROM PersonalizedDashboards
WHERE MemberID = 1;

-- Member books a personal training session
```

```sql
INSERT INTO PersonalTrainingSessions (SessionSchedule, Time, MemberID,
TrainerID)
VALUES ('2024-04-20', '10:00', 1, 1);

-- Schedule Group Fitness Classes
INSERT INTO Members_Classes (MemberID, ClassID)
VALUES (1, (SELECT ClassID FROM Classes WHERE Name = 'Yoga'));

-- Trainer Setting Availability - updating the availability for a trainer
UPDATE Trainers
SET Availability = ARRAY['10:00:00'::TIME, '12:00:00'::TIME,
'14:00:00'::TIME]
WHERE TrainerID = 1;

-- Trainers views a member's profile by searching Member's FirstName
SELECT * FROM Members
WHERE FirstName = 'Patrick';

-- AdministrativeStaff: book a room for a class
UPDATE Rooms
SET Availability = FALSE
WHERE RoomID = 1;

-- AdministrativeStaff- Equipment Maintenance Monitoring: updating the
equipment status
UPDATE Equipment
SET Description = 'out of order'
WHERE EquipmentID = 1;

-- Administrative Staff : Class Schedule Updating
UPDATE Classes
SET ScheduledTime = '15:00'
WHERE ClassID = 1;

-- Admin Issuing a bill for a member
INSERT INTO Billing (Date, PaymentMethod, Service, MemberID, ServiceID)
VALUES (
CURRENT_DATE,
   'Credit Card',
   (SELECT ServiceDescription FROM Services WHERE ServiceID = 1),
   1,
   1);

--Find the availability of all rooms:
SELECT RoomID, Name, Availability
FROM Rooms
WHERE Availability = TRUE;
```

2.5 Implementation

- **Project Description**

The project utilised a command-line interface with an application's architecture built on Java, interfacing with a PostgreSQL relational database via JDBC. It supports user roles like Members, Trainers, and Administrative Staff, each with specific functionalities encapsulated in separate Java classes. Operations such as user registration, updates, and session scheduling are managed through prepared SQL statements, ensuring security and efficiency. A central DBConnection class handles all database connections, optimizing resource use and maintenance. This setup is ideal for environments requiring simple, robust, and easily maintainable interfaces.For more Information about the project and how to set it up, check the README file.

2.6 Video Demo Link

Sorry the Video is a bit long but I didn't want to keep on re-recording. Please watch at 1.5 - 2x the speed.

https://share.vidyard.com/watch/ZS8dyoXqVZ8VsSZxAXDay3?