Fitness Center Membership Database – Final Project Report

Name: Joshua Villarojo

Class: INFO 4707 Data Modeling

## 1. Introduction

This final report details the implementation of the Fitness Center Membership Database, designed to support gym operations through efficient management of members, bookings, payments, trainers, and equipment. The report includes a full description of the system design and implementation, along with example queries to demonstrate its utility in real-world use cases.

## 2. Project Design

### 2.1 Description

This database is intended to centralize the operations of a fitness center. It handles tasks related to membership tracking, class and session booking, trainer assignments, and payment processing. It provides interfaces for both administrative users and gym members, improving access, organization, and efficiency.

### 2.2 Objectives

* Create and manage gym member records with personal and plan data.
* Enable scheduling for classes and training sessions.
* Track trainer assignments and class logistics.
* Manage billing and payment transactions securely.
* Provide reports to guide administrative decisions and analytics.

### 2.3 Scope

* Used by fitness centers for internal operations.
* Supports administrative and member-level access.
* Manages scheduling, billing, and equipment tracking.
* Can scale for use across multiple fitness centers.
* Enables role-based access for data entry and review.

### 2.4 Project Requirements

Operating System: Windows

Database: MySQL

Applications: MySQL Workbench, Lucidchart, Microsoft Word

### 2.5 User Requirements

* Members can manage their profiles and bookings.
* Staff can assign trainers to classes and track attendance.
* Payments are recorded and linked to members and plans.
* Role-based access ensures security of sensitive data.
* Data is stored for reporting and operational planning.

### 2.6 Database Requirements

* Member
* MembershipPlan
* ClassSchedule
* Trainer
* Booking
* Payment
* Equipment

### 2.7 Business Rules

* Each member is assigned one membership plan.
* Trainers may lead multiple classes but not at overlapping times.
* Members may book multiple sessions, subject to availability.
* Equipment must be scheduled for regular maintenance.
* Payments are required for continued access to facilities.

### 2.8 Entity-Relationship Diagram (ERD)

## 3. Data Dictionary

The data dictionary outlines the attributes of each table, including data types, formats, and constraints. It ensures consistency and clarity across the database system and serves as a key resource during development and maintenance.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Table | Field | Description | Type | Format | Required | Key |
| Member | Member\_ID | Primary identifier for member | INT | 10000-99999 | Yes | PK |
| Member | Plan\_ID | Membership type reference | INT | 1-999 | Yes | FK |
| MembershipPlan | Plan\_ID | Unique membership plan ID | INT | 1-999 | Yes | PK |
| Booking | Booking\_ID | Unique booking record ID | INT | 1-999999 | Yes | PK |
| Booking | Member\_ID | Linked to member | INT | 10000-99999 | Yes | FK |
| Booking | Class\_ID | Scheduled class reference | INT | 1-9999 | Yes | FK |
| ClassSchedule | Class\_ID | Unique class ID | INT | 1-9999 | Yes | PK |
| ClassSchedule | Trainer\_ID | Assigned trainer | INT | 1-999 | Yes | FK |
| Payment | Payment\_ID | Payment transaction record | INT | 1-9999999 | Yes | PK |
| Payment | Member\_ID | Paying member ID | INT | 10000-99999 | Yes | FK |
| Equipment | Equipment\_ID | Fitness equipment ID | INT | 1-9999 | Yes | PK |

## 4. Database Implementation

### 4.1 Entity Generation and Data Entry

The following SQL statements were used to create the core entities of the fitness center database. Each main entity contains at least 10 records, while supporting tables include a minimum of 5 records. Below are the statements and a short description of each table.

#### 4.1.1 CREATE TABLE Statements

* Table: Member

CREATE TABLE Member (  
 Member\_ID INT PRIMARY KEY,  
 FirstName VARCHAR(50),  
 LastName VARCHAR(50),  
 Email VARCHAR(100),  
 Phone VARCHAR(15),  
 Plan\_ID INT,  
 FOREIGN KEY (Plan\_ID) REFERENCES MembershipPlan(Plan\_ID)  
);

* Table: MembershipPlan

CREATE TABLE MembershipPlan (  
 Plan\_ID INT PRIMARY KEY,  
 Name VARCHAR(50),  
 Price DECIMAL(6,2),  
 DurationMonths INT  
);

* Table: Trainer

CREATE TABLE Trainer (  
 Trainer\_ID INT PRIMARY KEY,  
 FirstName VARCHAR(50),  
 LastName VARCHAR(50),  
 Specialty VARCHAR(100)  
);

* Table: ClassSchedule

CREATE TABLE ClassSchedule (  
 Class\_ID INT PRIMARY KEY,  
 ClassName VARCHAR(100),  
 StartTime DATETIME,  
 Trainer\_ID INT,  
 Capacity INT,  
 FOREIGN KEY (Trainer\_ID) REFERENCES Trainer(Trainer\_ID)  
);

* Table: Booking

CREATE TABLE Booking (  
 Booking\_ID INT PRIMARY KEY,  
 Member\_ID INT,  
 Class\_ID INT,  
 BookingDate DATE,  
 FOREIGN KEY (Member\_ID) REFERENCES Member(Member\_ID),  
 FOREIGN KEY (Class\_ID) REFERENCES ClassSchedule(Class\_ID)  
);

* Table: Payment

CREATE TABLE Payment (  
 Payment\_ID INT PRIMARY KEY,  
 Member\_ID INT,  
 Amount DECIMAL(6,2),  
 PaymentDate DATE,  
 FOREIGN KEY (Member\_ID) REFERENCES Member(Member\_ID)  
);

* Table: Equipment

CREATE TABLE Equipment (  
 Equipment\_ID INT PRIMARY KEY,  
 Name VARCHAR(100),  
 LastMaintenance DATE  
);

#### 4.1.2 INSERT INTO Statements and Screenshots

Each table has been populated using SQL INSERT INTO statements. Below are example statements and descriptions. Screenshots of the data inserted successfully in MySQL Workbench should be included here.

[Insert INSERT INTO statements and corresponding screenshots here. Include a caption for each screenshot.]

### 4.2 Data Retrieval and Reports

The following are sample queries designed to answer practical business questions. Each includes a plain-English description, the corresponding SQL SELECT statement, and expected results.

#### Query 1

Plain English: List all members in the Premium membership plan.

SQL Query:

SELECT FirstName, LastName FROM Member   
JOIN MembershipPlan ON Member.Plan\_ID = MembershipPlan.Plan\_ID   
WHERE MembershipPlan.Name = 'Premium';

#### A screenshot of a computer AI-generated content may be incorrect.

#### Query 2

Plain English: Show upcoming classes and assigned trainers.

SQL Query:

SELECT ClassName, StartTime, Trainer.FirstName FROM ClassSchedule   
JOIN Trainer ON ClassSchedule.Trainer\_ID = Trainer.Trainer\_ID   
WHERE StartTime > NOW();

#### A screenshot of a computer AI-generated content may be incorrect.

#### Query 3

Plain English: Display how many classes each member has booked.

SQL Query:

SELECT Member\_ID, COUNT(\*) AS BookingCount FROM Booking   
GROUP BY Member\_ID;

#### A screenshot of a computer AI-generated content may be incorrect.

#### Query 4

Plain English: List all equipment scheduled for maintenance this month.

SQL Query:

SELECT Equipment\_ID, Name, LastMaintenance FROM Equipment   
WHERE MONTH(LastMaintenance) = MONTH(CURDATE());

## A screenshot of a computer AI-generated content may be incorrect.

## 5. Conclusion

The implemented database meets the operational needs of a fitness center by supporting core functionalities such as member tracking, class booking, trainer management, and financial records. The system design adheres to relational database principles and demonstrates efficiency in data retrieval and reporting. Future work may include integration with a front-end system for online member access.