



Database Design for Fitness Center ABC



Tran Ha Trang
DATE: JULY 15, 2025

1. Executive Summary

Fitness Center ABC aims to increase revenue and member satisfaction but needs to know where to focus its operational efforts. This report analyzes member data to identify the most popular classes among members in the premium tiers. The key findings:

The solutions:

This design provides a structured, efficient, and scalable foundation for managing this fitness center and enhancing the membership services.

2. The Business Question

This report aims to answer the core question: "Which classes are most popular among members in the premium tiers, and how can the center use these insights to boost revenue and engagement?"

To answer this questions, the report will break it down to smaller parts:

- **Identify** the most valuable member segment based on membership tier revenue.
- **Analyze** the class attendance and preferences of this specific segment.
- **Come up** with data-driven recommendations to increase engagement and attract similar high-value members.

3. Database Schema

a. Entity Definitions

- **Members:** This entity registers personal information of every individual who has joined the gym. The attributes include:
 - MemberID (Primary Key, Integer)
 - Firstname (Varchar 50)
 - Lastname (Varchar 50)
 - Phonenummer (Varchar 20)
 - MembershipTierID (Varchar 50)
 - Joineddate (Date)
- **MembershipTier:** This entity stores information about different membership tiers and their prices. Its attributes include:
 - MembershipTierID (Primary key, Integer)
 - Tiername (Varchar 50)
 - Price (Decimal)
- **Class:** This entity keeps track of the class schedule, information of members who registered, and its trainers. The attributes are:
 - ClassID (Primary Key, Integer)

- Classname (Varchar 50)
- Classtime (Date time)
- Description (Varchar)
- TrainerID (Integer)
- **Trainer:** This entity records information about its trainers for every class the fitness center holds. The attributes include:
 - TrainerID (Primary Key, Integer)
 - Firstname (Varchar 50)
 - Lastname (Varchar 50)
 - Phonenumber (Varchar 20)
 - Specialization (Varchar 50)
- **Reservation:** This is the junction table between class and member (M:N relationship). The attributes are:
 - ReservationID (Primary key, Integer)
 - Reservation Time (Date time)
 - ClassID (Integer)
 - MemberID (Integer)

b. Entity-Relationship Diagram (ERD)

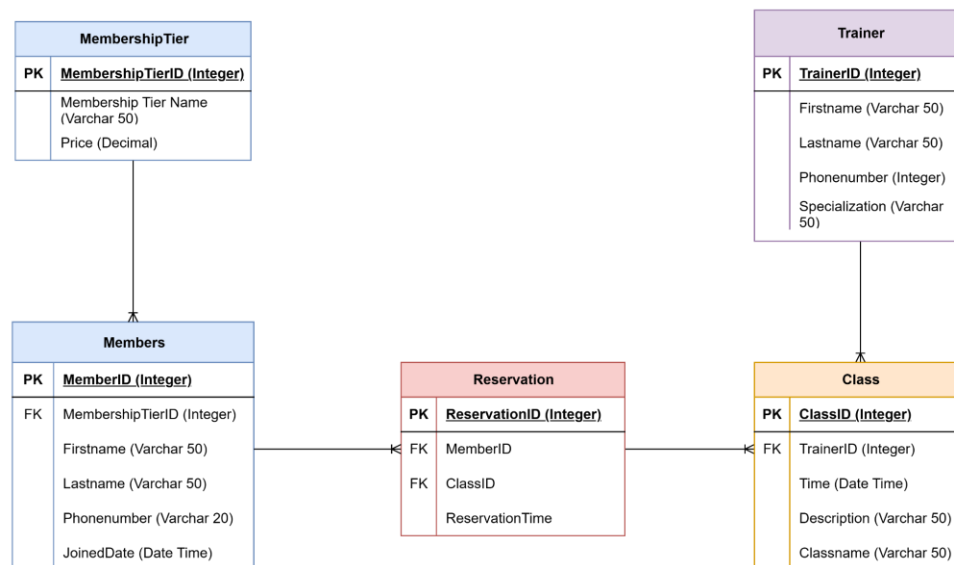


Figure 1: Entity-Relationship Diagram (ERD)

- **MembershipTier - Members (1-to-many):** One member can belong to only one membership tier but one membership tier can have many members.

- **Members - Reservation (1-to-many):** One member can make many reservations, but one reservation can only belongs to one member.
- **Reservation - Class (1-to-many):** One reservation can be made for one class, but one class can receive many reservations.
- **Trainer - Class (1-to-many):** One trainer can teach different classes but in one class, there can only be one trainer.

4. Analytical Findings

a. The Most Valuable Segment

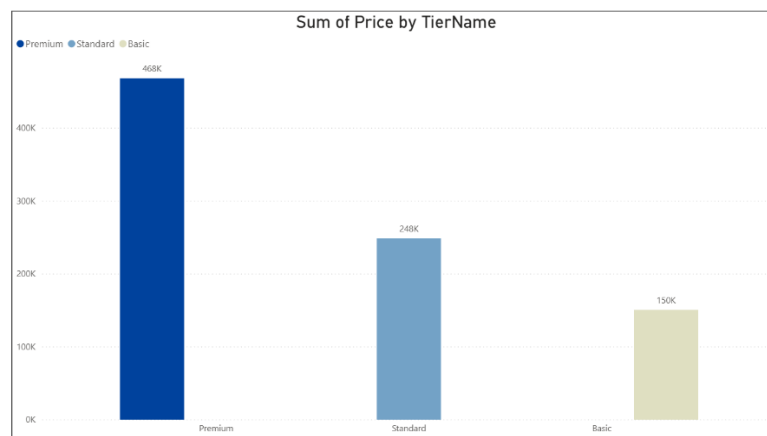


Figure 2: Sum of Price by TierName

As shown in Figure 2, the Premium tier members contribute the most revenue to the Fitness Center, with 468,000 JPY, significantly higher than that of the Standard and Basic tiers. This graph shows that the Premium tier is the most valuable segment, making their class preferences a critical focus for the remainder of the analysis.

b. Favorite Classes of the Premium Tier



Figure 3: Premium Member Reservations by ClassName

Figure 3 clearly depicted the most popular classes among premium members. It shows a preference for yoga and pilates, with 3 out of the 5 most popular classes are Morning Yoga, Restorative Yoga, and Core&More Pilates.

5. Recommendations

Based on the analysis, Fitness Center ABC should review the class schedules to optimize it for the premium members. This includes adding more yoga and pilates classes during the peak hours (evening and weekend) and reducing the frequency of less popular classes to allocate trainer resources in the most efficient way. The center can also start a marketing campaign to target the current members who look like the premium members. Use specific resources (photos, posters, etc) to highlight the gym's strength in these specific disciplines to attract a high-value clientele and increase premium-tier sign-ups."

6. Conclusion

The report has outlined a comprehensive and efficient relational database design for Fitness Center ABC. The proposed schema, centered around key entities such as **members**, **class**, and **reservation**, directly addresses all specified business requirements, from member subscription tracking to class scheduling. By establishing clear relationships and ensuring data integrity, this design provides a scalable and reliable foundation for the gym's daily operations. The SQL code used can be found in the Appendix.

Appendix:

SQL Code:

```
-- MembershipTier Table
CREATE TABLE MembershipTier (
    MembershipTierID SERIAL PRIMARY KEY,
    TierName VARCHAR(50) NOT NULL,
    Price DECIMAL(10, 2) NOT NULL
);

-- Trainer Table
CREATE TABLE Trainer (
    TrainerID SERIAL PRIMARY KEY,
    FirstName VARCHAR(50) NOT NULL,
    LastName VARCHAR(50) NOT NULL,
    PhoneNumber VARCHAR(20),
    Specialization VARCHAR(50)
);

-- Class Table
CREATE TABLE Class (
    ClassID SERIAL PRIMARY KEY,
    ClassName VARCHAR(50) NOT NULL,
    Description VARCHAR(255),
    ClassTime TIMESTAMP NOT NULL,
    TrainerID INTEGER,
    FOREIGN KEY (TrainerID) REFERENCES Trainer(TrainerID)
);

-- Members Table
CREATE TABLE Members (
    MemberID SERIAL PRIMARY KEY,
    FirstName VARCHAR(50) NOT NULL,
    LastName VARCHAR(50) NOT NULL,
    PhoneNumber VARCHAR(20),
    JoinedDate DATE NOT NULL,
    MembershipTierID INTEGER,
    FOREIGN KEY (MembershipTierID) REFERENCES
MembershipTier(MembershipTierID)
);

-- Reservation Table (Junction Table)
CREATE TABLE Reservation (
    ReservationID SERIAL PRIMARY KEY,
    MemberID INTEGER NOT NULL,
    ClassID INTEGER NOT NULL,
    ReservationTime TIMESTAMP NOT NULL,
    FOREIGN KEY (MemberID) REFERENCES Members(MemberID),
    FOREIGN KEY (ClassID) REFERENCES Class(ClassID)
);
```

```
SELECT
    mt.TierName,
    COUNT(m.MemberID) AS NumberOfMembers,
    SUM(mt.Price) AS TotalRevenue
FROM
```

```
        Members m
JOIN
    MembershipTier mt ON m.MembershipTierID = mt.MembershipTierID
GROUP BY
    mt.TierName
ORDER BY
    TotalRevenue DESC;

SELECT
    c.ClassName,
    COUNT(r.ReservationID) AS NumberOfReservations
FROM
    Reservation r
JOIN
    Class c ON r.ClassID = c.ClassID
JOIN
    Members m ON r.MemberID = m.MemberID
JOIN
    MembershipTier mt ON m.MembershipTierID = mt.MembershipTierID
WHERE
    mt.TierName = 'Premium'
GROUP BY
    c.ClassName
ORDER BY
    NumberOfReservations DESC
LIMIT 5;
```