

**SQL / NOSQL BASED DATA ARCHITECTURES**

*CAPSTONE PROJECT**FITNESS CLUB MANAGEMENT*

**Pooja jaju - 24WUO202353**

**Konala Nikhil - 24WU0202330**

**Mutpuri Akhil Gupta - 24WU0202343**

**Suparna Nadendla - 24WU0202345**

**Shreya Sarojini - 24WU0202347**

**Satya Sri Priya - 24WU0202378**

***Fitness Club Management Project Report***

**Project Overview**

The Fitness Club Management project aims to streamline operations by creating a database to manage members, trainers, fitness classes, and attendance records. The primary goals are to:

* Centralize member and trainer data for better accessibility.
* Track attendance and class participation.
* Generate insights about membership trends, and class popularity.

Expected outcomes include improved administrative efficiency, actionable insights from data queries, and enhanced member experience through effective resource management.

**Purpose**

The primary purpose of this project is to create a centralized database system for efficiently managing key aspects of a fitness club, including members, trainers, classes, and attendance records. By transitioning from manual or disjointed data management methods to a structured relational database, the fitness club aims to improve operational efficiency, enhance user experience, and enable better decision-making through accurate and accessible data.

**Goals**

1. **Streamline Operations:**  
   Automate and optimize daily administrative tasks like member registration, class scheduling, trainer assignments.
2. **Data Integrity and Accessibility:**  
   Ensure accurate, consistent, and secure data storage that can be easily retrieved, updated, or analyzed by authorized personnel.
3. **Enhanced Member Engagement:**  
   Provide a system that allows for personalized member services, such as tailored class recommendations or tracking individual attendance.
4. **Performance Monitoring:**  
   Enable club managers to monitor the performance of trainers and the popularity of classes, facilitating informed decision-making and resource allocation.
5. **Scalability:**  
   Build a system capable of scaling with the club's growth, accommodating more members, trainers, and class offerings over time.

**Expected Outcomes**

1. **Improved Efficiency:**  
   Reduction in manual effort for administrative staff, allowing them to focus on enhancing the overall club experience.
2. **Accurate Data Management:**  
   Eliminate data redundancy and errors through a well-designed relational database that maintains data integrity.
3. **Better Insights:**  
   Generate meaningful reports on membership trends, trainer performance, and class attendance patterns for strategic planning.
4. **Enhanced Member Satisfaction:**  
   Offer a seamless experience for members through better-organized schedules, availability of classes, and transparent attendance records.
5. **Future-Ready System:**  
   A flexible and scalable system that can integrate with additional features like online booking systems, mobile apps, or fitness tracking tools in the future.

**Database Design**

The database consists of four primary tables: **Members**, **Trainers**, **Classes**, and **Attendance**, with their structures as follows:

DATABASE NAME -> FitnessClub

**1.Members Table**

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| MemberID | INT (Primary Key, Auto Increment) | Unique identifier for each member. |
| FirstName | VARCHAR(50) | Member's first name. |
| LastName | VARCHAR(50) | Member's last name. |
| MembershipType | VARCHAR(20) | Type of membership (e.g., Premium, Standard). |
| MembershipFee | DECIMAL(10, 2) | Fee associated with the membership. |
| JoinDate | DATE | Date the member joined. |

**2. Trainers Table**

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| TrainerID | INT (Primary Key, Auto Increment) | Unique identifier for each trainer. |
| FirstName | VARCHAR(50) | Trainer's first name. |
| LastName | VARCHAR(50) | Trainer's last name. |
| Specialty | VARCHAR(50) | Trainer's area of expertise. |
| HireDate | DATE | Date the trainer was hired. |

**3. Classes Table**

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| ClassID | INT (Primary Key, Auto Increment) | Unique identifier for each class. |
| ClassName | VARCHAR(50) | Name of the fitness class. |
| TrainerID | INT (Foreign Key) | Links the class to a specific trainer. |
| ClassTime | TIME | Scheduled time for the class. |
| Fee | DECIMAL(10, 2) | Fee for attending the class. |

**4. Attendance Table**

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| AttendanceID | INT (Primary Key, Auto Increment) | Unique identifier for attendance. |
| MemberID | INT (Foreign Key) | Links attendance to a specific member. |
| ClassID | INT (Foreign Key) | Links attendance to a specific class. |
| AttendanceDate | DATE | Date of attendance. |

**TABLE RELATIONSHIP:**

* **Members ↔ Attendance**: A one-to-many relationship where a member can attend multiple classes.
* **Trainers ↔ Classes**: A one-to-many relationship where a trainer can conduct multiple classes.
* **Classes ↔ Attendance**: A one-to-many relationship where a class can have multiple attendees.

**Insights and Analysis**

**Use Cases**

1. **Member Management:**
   * **Scenario:** A new member joins the fitness club. The database stores their details, including name, contact information, membership type, and start/end dates.
   * **Action:** Staff can query the Members table to retrieve member profiles, check membership validity, and update personal details as needed.
2. **Class Scheduling and Registration:**
   * **Scenario:** A member wants to enroll in a yoga class. The database checks the availability of the class and updates the member's registration.
   * **Action:** Queries on the Classes table determine class schedules, available slots, and trainer assignments.
3. **Trainer Assignment and Management:**
   * **Scenario:** A trainer is assigned to a new fitness class. The system links the trainer's details to the class in the Classes table.
   * **Action:** Managers can retrieve all classes conducted by a specific trainer to monitor their workload and performance.
4. **Membership Renewal and Expiry Alerts:**
   * **Scenario:** A member's membership is about to expire. The system flags the member for renewal notifications.
   * **Action:** Queries on the Members table identify memberships nearing expiration and trigger alerts.
5. **Class Popularity Analysis:**
   * **Scenario:** The club wants to analyze which classes have the highest attendance.
   * **Action:** Aggregate queries on the Attendance table reveal trends in class participation, helping optimize schedules and resources.

**Practical Insights**

1. **Optimized Scheduling:**
   * Data from the Classes and Attendance tables helps identify peak times for classes and allocate trainers efficiently, ensuring better utilization of resources.
2. **Personalized Member Engagement:**
   * Insights from attendance records allow the club to suggest classes or training programs tailored to individual preferences, improving member satisfaction.
3. **Performance Metrics for Trainers:**
   * By analyzing trainer-specific class data, management can assess trainers' effectiveness, identify areas for improvement, and reward high-performing staff.
4. **Membership Trends:**
   * Tracking active and inactive members helps the club devise retention strategies, such as special offers or targeted marketing campaigns.
5. **Improved Financial Planning:**
   * Membership and attendance data enable accurate revenue forecasting and resource allocation, ensuring sustainable growth.
6. **Scalability:**
   * The modular design of the database allows for the integration of advanced features like online member portals, fitness tracking, and mobile applications as the club expands.
7. **Compliance and Security:**
   * Centralized and secure data management ensures compliance with data protection regulations and safeguards sensitive member information.

**Conclusion:**

The Fitness Club Management database offers an efficient and comprehensive solution for managing the club's operations. By tracking member and trainer details, attendance records, and membership plans, the database enables management to make informed, data-driven decisions. It allows for personalized member engagement, optimized trainer assignments, and better resource allocation, leading to improved member satisfaction and retention.

Additionally, the database’s ability to generate advanced queries and reports provides actionable insights into class popularity, financial trends, and staffing needs. This supports strategic decision-making, enhances operational efficiency, and improves overall profitability. With streamlined administrative tasks, the system helps the club operate more smoothly while focusing on providing high-quality customer service.

**Queries and O/P Snippets**

SQL CODE:

-- Create the database

CREATE DATABASE FitnessClub;

USE FitnessClub;

-- Create the Members table

CREATE TABLE Members (

MemberID INT AUTO\_INCREMENT PRIMARY KEY,

FirstName VARCHAR(50),

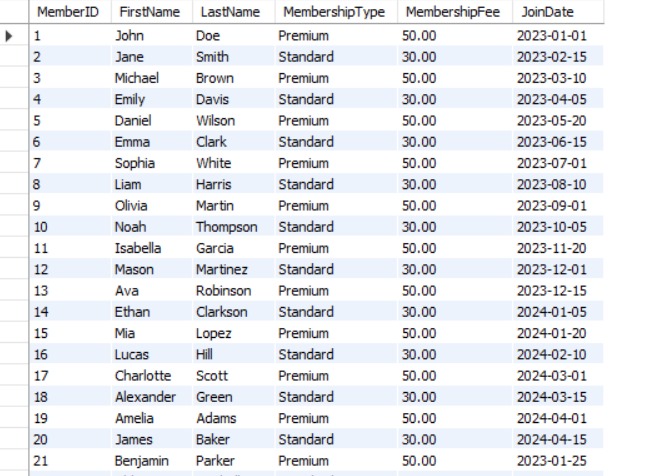
LastName VARCHAR(50),

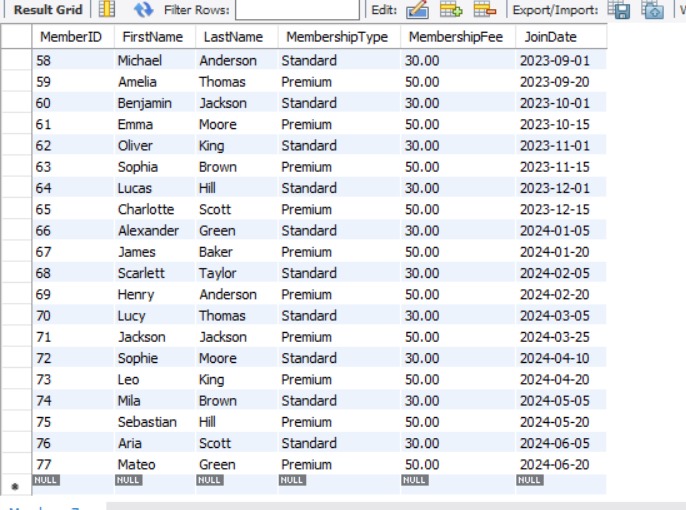
MembershipType VARCHAR(20),

MembershipFee DECIMAL(10,2),

JoinDate DATE

);





-- Create the Trainers table

CREATE TABLE Trainers (

TrainerID INT AUTO\_INCREMENT PRIMARY KEY,

FirstName VARCHAR(50),

LastName VARCHAR(50),

Specialty VARCHAR(50),

HireDate DATE

);



-- Create the Classes table

CREATE TABLE Classes (

ClassID INT AUTO\_INCREMENT PRIMARY KEY,

ClassName VARCHAR(50),

TrainerID INT,

ClassTime TIME,

Fee DECIMAL(10, 2),

FOREIGN KEY (TrainerID) REFERENCES Trainers(TrainerID)

);



-- Create the Attendance table

CREATE TABLE Attendance (

AttendanceID INT AUTO\_INCREMENT PRIMARY KEY,

MemberID INT,

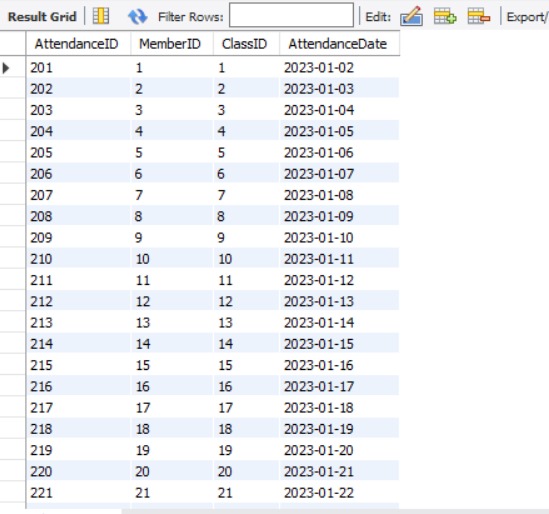
ClassID INT,

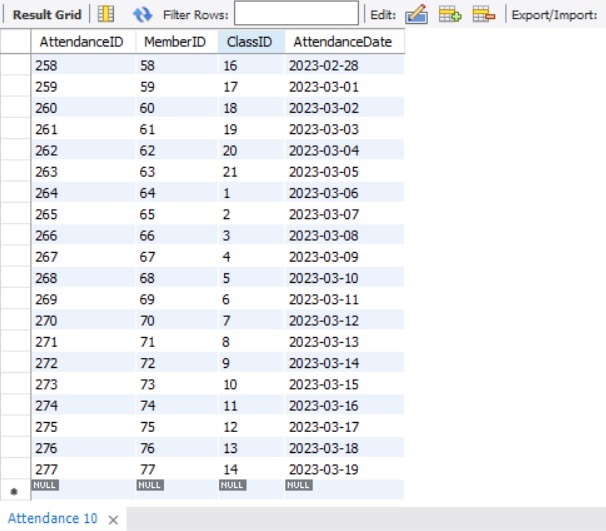
AttendanceDate DATE,

FOREIGN KEY (MemberID) REFERENCES Members(MemberID),

FOREIGN KEY (ClassID) REFERENCES Classes(ClassID)

);





-- Populate the Members table

INSERT INTO Members (FirstName, LastName, MembershipType, MembershipFee, JoinDate)

VALUES

('John', 'Doe', 'Premium', 50.00, '2023-01-01'),

('Jane', 'Smith', 'Standard', 30.00, '2023-02-15'),

('Michael', 'Brown', 'Premium', 50.00, '2023-03-10'),

('Emily', 'Davis', 'Standard', 30.00, '2023-04-05'),

('Daniel', 'Wilson', 'Premium', 50.00, '2023-05-20'),

('Emma', 'Clark', 'Standard', 30.00, '2023-06-15'),

('Sophia', 'White', 'Premium', 50.00, '2023-07-01'),

('Liam', 'Harris', 'Standard', 30.00, '2023-08-10'),

('Olivia', 'Martin', 'Premium', 50.00, '2023-09-01'),

('Noah', 'Thompson', 'Standard', 30.00, '2023-10-05'),

('Isabella', 'Garcia', 'Premium', 50.00, '2023-11-20'),

('Mason', 'Martinez', 'Standard', 30.00, '2023-12-01'),

('Ava', 'Robinson', 'Premium', 50.00, '2023-12-15'),

('Ethan', 'Clarkson', 'Standard', 30.00, '2024-01-05'),

('Mia', 'Lopez', 'Premium', 50.00, '2024-01-20'),

('Lucas', 'Hill', 'Standard', 30.00, '2024-02-10'),

('Charlotte', 'Scott', 'Premium', 50.00, '2024-03-01'),

('Alexander', 'Green', 'Standard', 30.00, '2024-03-15'),

('Amelia', 'Adams', 'Premium', 50.00, '2024-04-01'),

('James', 'Baker', 'Standard', 30.00, '2024-04-15'),

('Benjamin', 'Parker', 'Premium', 50.00, '2023-01-25'),

('Chloe', 'Mitchell', 'Standard', 30.00, '2023-02-10'),

('Elijah', 'Perez', 'Premium', 50.00, '2023-03-20'),

('Victoria', 'Roberts', 'Standard', 30.00, '2023-04-15'),

('Jacob', 'Turner', 'Premium', 50.00, '2023-05-05'),

('Hannah', 'Phillips', 'Standard', 30.00, '2023-06-01'),

('William', 'Campbell', 'Premium', 50.00, '2023-07-10'),

('Grace', 'Evans', 'Standard', 30.00, '2023-08-05'),

('Logan', 'Edwards', 'Premium', 50.00, '2023-09-15'),

('Ella', 'Collins', 'Standard', 30.00, '2023-10-25'),

('Charlotte', 'Wilson', 'Premium', 50.00, '2023-11-10'),

('James', 'Miller', 'Standard', 30.00, '2023-12-05'),

('Scarlett', 'Taylor', 'Premium', 50.00, '2023-12-20'),

('Henry', 'Anderson', 'Standard', 30.00, '2024-01-10'),

('Lucy', 'Thomas', 'Premium', 50.00, '2024-01-25'),

('Jackson', 'Jackson', 'Standard', 30.00, '2024-02-15'),

('Sophie', 'Moore', 'Premium', 50.00, '2024-02-20'),

('Leo', 'King', 'Standard', 30.00, '2024-03-05'),

('Mila', 'Brown', 'Premium', 50.00, '2024-03-25'),

('Sebastian', 'Hill', 'Standard', 30.00, '2024-04-10'),

('Aria', 'Scott', 'Premium', 50.00, '2024-04-20'),

('Mateo', 'Green', 'Standard', 30.00, '2023-01-15'),

('Layla', 'Adams', 'Premium', 50.00, '2023-02-01'),

('Daniel', 'Baker', 'Standard', 30.00, '2023-02-20'),

('Lily', 'Parker', 'Premium', 50.00, '2023-03-05'),

('Aiden', 'Mitchell', 'Standard', 30.00, '2023-03-15'),

('Emily', 'Perez', 'Premium', 50.00, '2023-03-30'),

('Owen', 'Roberts', 'Standard', 30.00, '2023-04-10'),

('Zoe', 'Turner', 'Premium', 50.00, '2023-04-25'),

('Ethan', 'Phillips', 'Standard', 30.00, '2023-05-10'),

('Harper', 'Campbell', 'Premium', 50.00, '2023-05-25'),

('Logan', 'Evans', 'Standard', 30.00, '2023-06-05'),

('Ella', 'Edwards', 'Premium', 50.00, '2023-06-20'),

('Levi', 'Collins', 'Standard', 30.00, '2023-07-05'),

('Luna', 'Wilson', 'Premium', 50.00, '2023-07-20'),

('Carter', 'Miller', 'Standard', 30.00, '2023-08-01'),

('Chloe', 'Taylor', 'Premium', 50.00, '2023-08-15'),

('Michael', 'Anderson', 'Standard', 30.00, '2023-09-01'),

('Amelia', 'Thomas', 'Premium', 50.00, '2023-09-20'),

('Benjamin', 'Jackson', 'Standard', 30.00, '2023-10-01'),

('Emma', 'Moore', 'Premium', 50.00, '2023-10-15'),

('Oliver', 'King', 'Standard', 30.00, '2023-11-01'),

('Sophia', 'Brown', 'Premium', 50.00, '2023-11-15'),

('Lucas', 'Hill', 'Standard', 30.00, '2023-12-01'),

('Charlotte', 'Scott', 'Premium', 50.00, '2023-12-15'),

('Alexander', 'Green', 'Standard', 30.00, '2024-01-05'),

('James', 'Baker', 'Premium', 50.00, '2024-01-20'),

('Scarlett', 'Taylor', 'Standard', 30.00, '2024-02-05'),

('Henry', 'Anderson', 'Premium', 50.00, '2024-02-20'),

('Lucy', 'Thomas', 'Standard', 30.00, '2024-03-05'),

('Jackson', 'Jackson', 'Premium', 50.00, '2024-03-25'),

('Sophie', 'Moore', 'Standard', 30.00, '2024-04-10'),

('Leo', 'King', 'Premium', 50.00, '2024-04-20'),

('Mila', 'Brown', 'Standard', 30.00, '2024-05-05'),

('Sebastian', 'Hill', 'Premium', 50.00, '2024-05-20'),

('Aria', 'Scott', 'Standard', 30.00, '2024-06-05'),

('Mateo', 'Green', 'Premium', 50.00, '2024-06-20');

select \* from Members;

-- Populate the Trainers table

INSERT INTO Trainers (FirstName, LastName, Specialty, HireDate)

VALUES

('David', 'Miller', 'Yoga', '2022-01-01'),

('Sophia', 'Taylor', 'Pilates', '2022-02-15'),

('Oliver', 'Anderson', 'Strength Training', '2022-03-10'),

('Ava', 'Thomas', 'Cardio', '2022-04-05'),

('Ethan', 'Jackson', 'HIIT', '2022-05-20'),

('Emma', 'Moore', 'Zumba', '2022-06-15'),

('Liam', 'King', 'Cycling', '2022-07-01'),

('Olivia', 'Brown', 'Dance Fitness', '2022-08-10'),

('Lucas', 'Smith', 'Aerobics', '2022-09-15'),

('Charlotte', 'Wilson', 'Bodybuilding', '2022-10-01'),

('Mason', 'Davis', 'Meditation', '2022-11-05'),

('Chloe', 'Garcia', 'Kickboxing', '2022-12-01'),

('Jacob', 'Evans', 'CrossFit', '2022-12-15'),

('Grace', 'Martinez', 'Barre', '2023-01-10'),

('Elijah', 'Harris', 'Functional Training', '2023-02-05'),

('Victoria', 'Thompson', 'Spin', '2023-02-20'),

('William', 'Green', 'Stretching', '2023-03-01'),

('Benjamin', 'Roberts', 'Core Stability', '2023-03-15'),

('Scarlett', 'Parker', 'Bootcamp', '2023-04-01'),

('Henry', 'Phillips', 'TRX Training', '2023-04-15'),

('Lily', 'Adams', 'Strength & Conditioning', '2023-05-01');

select \* from Trainers;

-- Populate the Classes table

INSERT INTO Classes (ClassName, TrainerID, ClassTime, Fee)

VALUES

('Morning Yoga', 1, '08:00:00', 10.00),

('Evening Yoga', 1, '18:00:00', 10.00),

('Pilates Basics', 2, '10:00:00', 15.00),

('Advanced Pilates', 2, '16:00:00', 15.00),

('Strength Training 101', 3, '09:00:00', 20.00),

('Cardio Blast', 4, '11:00:00', 12.00),

('HIIT Express', 5, '13:00:00', 18.00),

('Zumba Party', 6, '15:00:00', 14.00),

('Cycling Challenge', 7, '17:00:00', 15.00),

('Dance Fitness Fun', 8, '19:00:00', 12.00),

('Meditation Basics', 9, '08:30:00', 8.00),

('Kickboxing Advanced', 10, '09:30:00', 20.00),

('CrossFit Strength', 11, '10:30:00', 25.00),

('Barre Fundamentals', 12, '12:00:00', 10.00),

('Spin Power', 13, '14:00:00', 18.00),

('Core Stability Level 1', 14, '16:30:00', 12.00),

('Bootcamp Challenge', 15, '18:00:00', 22.00),

('TRX Training', 16, '07:30:00', 14.00),

('Stretch & Relax', 17, '19:30:00', 10.00),

('Functional HIIT', 18, '06:30:00', 18.00),

('Strength & Conditioning Elite', 19, '20:00:00', 25.00);

-- Populate the Attendance table

INSERT INTO Attendance (MemberID, ClassID, AttendanceDate)

VALUES

(1, 1, '2023-01-02'),

(2, 2, '2023-01-03'),

(3, 3, '2023-01-04'),

(4, 4, '2023-01-05'),

(5, 5, '2023-01-06'),

(6, 6, '2023-01-07'),

(7, 7, '2023-01-08'),

(8, 8, '2023-01-09'),

(9, 9, '2023-01-10'),

(10, 10, '2023-01-11'),

(11, 11, '2023-01-12'),

(12, 12, '2023-01-13'),

(13, 13, '2023-01-14'),

(14, 14, '2023-01-15'),

(15, 15, '2023-01-16'),

(16, 16, '2023-01-17'),

(17, 17, '2023-01-18'),

(18, 18, '2023-01-19'),

(19, 19, '2023-01-20'),

(20, 20, '2023-01-21'),

(21, 21, '2023-01-22'),

(22, 1, '2023-01-23'),

(23, 2, '2023-01-24'),

(24, 3, '2023-01-25'),

(25, 4, '2023-01-26'),

(26, 5, '2023-01-27'),

(27, 6, '2023-01-28'),

(28, 7, '2023-01-29'),

(29, 8, '2023-01-30'),

(30, 9, '2023-01-31'),

(31, 10, '2023-02-01'),

(32, 11, '2023-02-02'),

(33, 12, '2023-02-03'),

(34, 13, '2023-02-04'),

(35, 14, '2023-02-05'),

(36, 15, '2023-02-06'),

(37, 16, '2023-02-07'),

(38, 17, '2023-02-08'),

(39, 18, '2023-02-09'),

(40, 19, '2023-02-10'),

(41, 20, '2023-02-11'),

(42, 21, '2023-02-12'),

(43, 1, '2023-02-13'),

(44, 2, '2023-02-14'),

(45, 3, '2023-02-15'),

(46, 4, '2023-02-16'),

(47, 5, '2023-02-17'),

(48, 6, '2023-02-18'),

(49, 7, '2023-02-19'),

(50, 8, '2023-02-20'),

(51, 9, '2023-02-21'),

(52, 10, '2023-02-22'),

(53, 11, '2023-02-23'),

(54, 12, '2023-02-24'),

(55, 13, '2023-02-25'),

(56, 14, '2023-02-26'),

(57, 15, '2023-02-27'),

(58, 16, '2023-02-28'),

(59, 17, '2023-03-01'),

(60, 18, '2023-03-02'),

(61, 19, '2023-03-03'),

(62, 20, '2023-03-04'),

(63, 21, '2023-03-05'),

(64, 1, '2023-03-06'),

(65, 2, '2023-03-07'),

(66, 3, '2023-03-08'),

(67, 4, '2023-03-09'),

(68, 5, '2023-03-10'),

(69, 6, '2023-03-11'),

(70, 7, '2023-03-12'),

(71, 8, '2023-03-13'),

(72, 9, '2023-03-14'),

(73, 10, '2023-03-15'),

(74, 11, '2023-03-16'),

(75, 12, '2023-03-17'),

(76, 13, '2023-03-18'),

(77, 14, '2023-03-19');

select \* from Attendance;

-- Query to join members with trainers and classes

SELECT

Members.FirstName AS MemberFirstName,

Members.LastName AS MemberLastName,

Trainers.FirstName AS TrainerFirstName,

Trainers.LastName AS TrainerLastName,

Classes.ClassName,

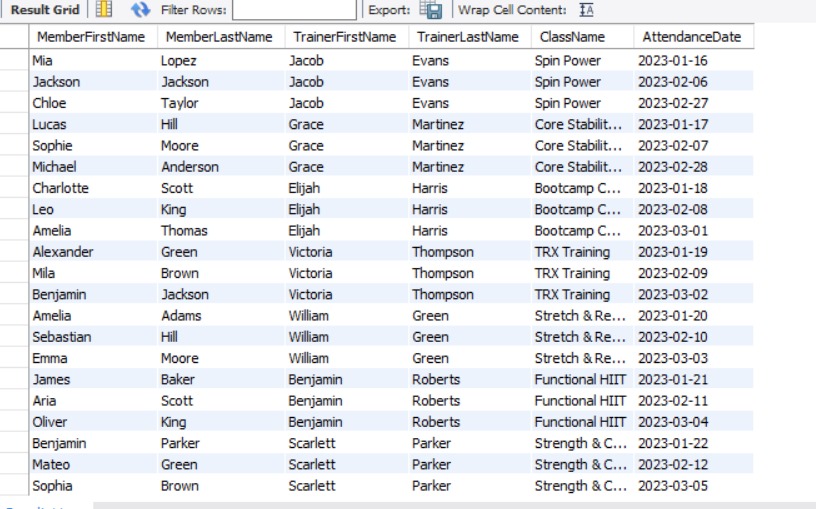
Attendance.AttendanceDate

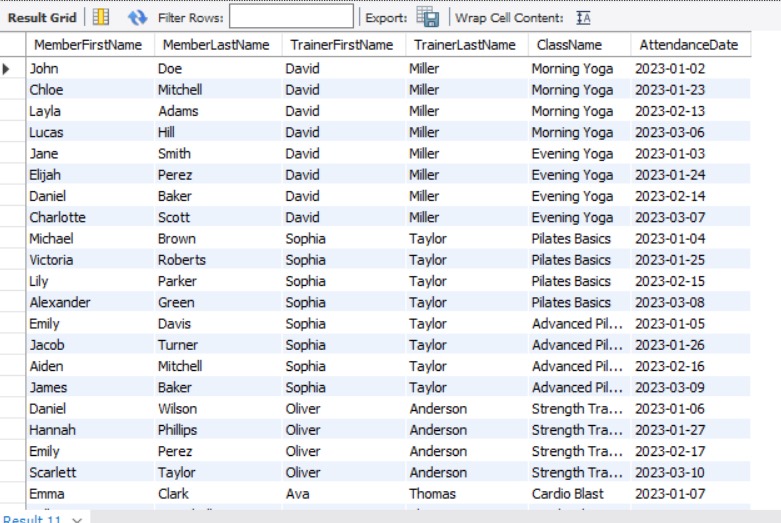
FROM Attendance

JOIN Members ON Attendance.MemberID = Members.MemberID

JOIN Classes ON Attendance.ClassID = Classes.ClassID

JOIN Trainers ON Classes.TrainerID = Trainers.TrainerID;





-- Query to calculate average attendance per class

SELECT

Classes.ClassName,

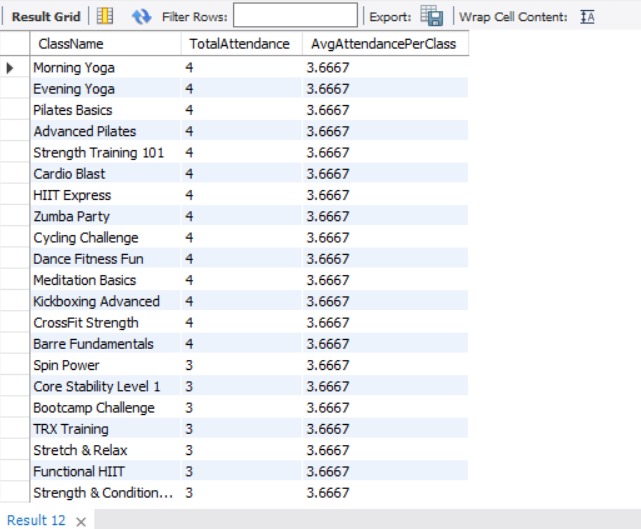
COUNT(Attendance.AttendanceID) AS TotalAttendance,

AVG(COUNT(Attendance.AttendanceID)) OVER() AS AvgAttendancePerClass

FROM Attendance

JOIN Classes ON Attendance.ClassID = Classes.ClassID

GROUP BY Classes.ClassID;



-- Query to calculate total revenue from memberships

SELECT

SUM(Members.MembershipFee) AS TotalMembershipRevenue

FROM Members;



-- Revenue Breakdown by Class

SELECT

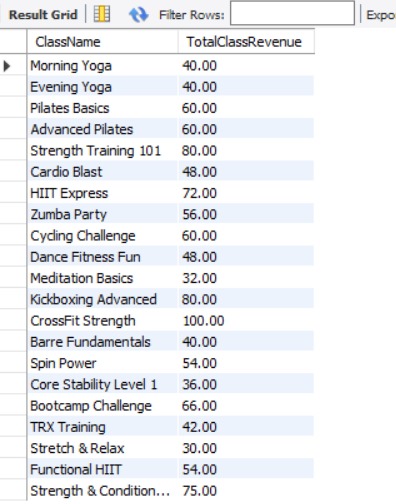
Classes.ClassName,

COUNT(Attendance.AttendanceID) \* Classes.Fee AS TotalClassRevenue

FROM Attendance

JOIN Classes ON Attendance.ClassID = Classes.ClassID

GROUP BY Classes.ClassID;



**--To find premium members from the table:**

SELECT FirstName, LastName, MembershiptypeFROM

Members

WHERE MembershipType = 'Premium';

A screenshot of a computer

Description automatically generated

--**List All Classes Offered by Each Trainer**

SELECT

Trainers.FirstName AS TrainerFirstName,

Trainers.LastName AS TrainerLastName,

Classes.ClassName

FROM Trainers

JOIN Classes ON Trainers.TrainerID = Classes.TrainerID

ORDER BY Trainers.FirstName, Classes.ClassName;

A screenshot of a computer

Description automatically generated

**--Find the Least Popular Class**

SELECT

Classes.ClassName,

COUNT(Attendance.AttendanceID) AS TotalAttendance

FROM Attendance

JOIN Classes ON Attendance.ClassID = Classes.ClassID

GROUP BY Classes.ClassID

ORDER BY TotalAttendance ASC

LIMIT 1;

A screenshot of a computer

Description automatically generated

--**Trainers Who Were Hired After a Specific Date**

SELECT FirstName, LastName, Specialty,

HireDate FROM Trainers

WHERE HireDate > '2023-01-01';

A screenshot of a computer

Description automatically generated