

A MINI-PROJECT REPORT ON
“GYM MANAGEMENT SYSTEM”

SUBMITTED TO SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE IN
THE PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE AWARD OF THE DEGREE

Bachelor of Engineering
in
Information Technology

Class: S.E

BY

Name :	Roll No./Exam Seat No. :
Prathamesh Gadekar	15
Jagdish Kalokhe	25
Parjanya Bhuse	07
Labina Malik	33

Under the guidance of
Ms. Jaitee Bankar



Sinhgad Institutes

DEPARTMENT OF INFORMATION TECHNOLOGY

RMD SINHGAD SCHOOL OF ENGINEERING

WARJE, PUNE-411058

A.Y: 2024 - 25



Sinhgad Institutes
DEPARTMENT OF INFORMATION TECHNOLOGY
RMD SINHGAD SCHOOL OF ENGINEERING

WARJE, PUNE-411058

CERTIFICATE

This is to certify that the Mini-Project Report entitled

“Gym Management System”

Submitted by

Name:	Roll No./Exam Seat No. :
Prathamesh Gadekar	15
Jagdish Kalokhe	25
Parjanya Bhuse	07
Labina Malik	33

is a Bonafide work carried out by him/her under the supervision of **Ms. Jaitee Bankar** and it is submitted towards the partial fulfillment of the requirement for S.E (Information Technology) – 2019 course of Savitribai Phule Pune University, Pune in the academic year 2024-2025.

(Ms. Jaitee Bankar)

Guide

Department of Information Technology

(Prof. Sweta Kale)

Head,

Department of Information Technology

Place:

Date:

(Dr. V.V. Dixit)

Principal,

RMD Sinhgad School of Engineering Pune – 58

ACKNOWLEDGEMENT

Apart from the efforts of team, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project. The completion efforts of several sources of knowledge.

The valuable support was given by our HOD Prof. Sweta Kale . We are eternally grateful to our guide Ms.Jaitee Bankar for her even willingness to give us valuable advice and direction under which we executed this project. Her constant guidance and willingness to share her vast knowledge made us understand this project and its manifestations in great depths and helped us to complete the assigned tasks.

The contribution and support was received from all the team members including Prathamesh Gadekar, Jagdish Kalokhe , Parjanya Bhuse, Labina Malik is vital. The team spirit shown by all has made a project report work successful.

Date: _____

NAME OF THE STUDENTS

- 1) Prathamesh Gadekar
- 2) Jagdish Kalokhe
- 3) Parjanya Bhuse
- 4) Labina Malik

CONTENTS

Sr.No.	Contents	Page No.
1.	Abstract	1
2.	Introduction	3
3.	Data Types	6
4.	Data Requirements 4.1 Requirements Collection and Analysis 4.2 Entities 4.3 Attributes and Keys 4.4 Relationships- Cardinality	11
5.	E-R Diagram	19
6.	Schema Diagram	20
7	Relational Database Design	21
8.	Creating database using MySQL	27
9.	Test Case Queries	32
10.	Conclusion	50
11.	References	51

1. ABSTRACT

The project Gym Management System developed using MYSQL for database management the system provides the interface for all the members where they can see gym related information but only the gym admin can access the website where he can perform various operations.

Various operations are: -

- SEARCH members.
- ADD new members.
- DELETE old members.
- UPDATE members information.

The project aims to provide an efficient and user-friendly system for gym management. The Gym Management System is a comprehensive software solution designed to streamline and automate the daily operations of a fitness center. This system enables efficient handling of member registrations, attendance tracking, subscription management, trainer scheduling, and payment processing. By replacing manual and paper-based processes with a digital platform, the system improves accuracy, reduces administrative workload, and enhances member experience. The application offers an intuitive user interface for both administrators and users, ensuring easy access to essential features. Additionally, the system includes reporting tools for performance analysis and business insights. Overall, the Gym Management System aims to increase operational efficiency and provide a scalable solution for gyms and fitness centers of all sizes.

2. INTRODUCTION

2.1. Problem Statement: To Develop a MySQL-based Gym Management System to efficiently manage members, trainers, sessions, and payments by automating tasks like attendance tracking, scheduling, and payment management.

2.2. Motivation

The motivation behind this project is to streamline gym operations by creating a centralized database system. Automating tasks such as tracking member details, managing attendance, scheduling workout sessions, and assigning trainers will greatly reduce manual work. Additionally, maintaining accurate payment histories and membership records will help ensure transparency and efficiency. Overall, the system aims to simplify management for gym staff while enhancing the experience for members.

The motivation behind developing the Gym Management System is to:

- To streamline gym operations through a centralized database.
-
- To automate routine tasks like tracking member details and attendance.
-
- To efficiently manage workout session scheduling and trainer assignments.
-
- To maintain accurate payment histories and membership records.
-
- To improve overall efficiency for gym staff and enhance the member experience.

2.3. Objectives:

1. To develop a user-friendly system for managing gym operations efficiently.
2. To automate member registration, attendance tracking, subscription management, and payment handling.
3. To reduce manual work and human error by digitizing records and processes.
4. To provide real-time access to member and trainer data for better decision-making.
5. To enhance customer satisfaction through timely updates and seamless communication.
6. To design and implement a normalized relational database to store and manage gym-related data securely.
7. To develop modules for CRUD (Create, Read, Update, Delete) operations on members, trainers, schedules, and payments.
8. To ensure data integrity, consistency, and security through proper constraints and access control.
9. To use SQL queries for efficient data manipulation and reporting.
10. To build a responsive interface that interacts smoothly with the backend database.
11. To implement authentication and role-based access for administrators, trainers, and members.

3. DATA TYPES

Data Types used in Gym Management System project, aligned with the entities and their attributes from the database:

1. Users

user_id – INT

name – VARCHAR(50)

age – INT

gender – VARCHAR(50)

phone – VARCHAR(20)

mail – VARCHAR(30)

role – enum(Trainer , Member) not null

joins_date – DATE

specialization – VARCHAR(100)

2. Plans

plan_id – INT auto_incrmenet primary key

plan_name – VARCHAR(50)

duration – VARCHAR(50)

price – DECIMAL(10,2)

Gym Management System

3. *Payments*

payment_id – INT auto_incrmenet primary key

user_id – INT

plan_id – INT

amount – DECIMAL (10,2)

payment_date – DATE

4. *Attendance*

attendance_id – INT auto_incrmenet primary key

user_id – INT

Date – DATE

4.DATA REQUIREMENTS

4.1 Requirements Collection and Analysis

The system is designed to manage day-to-day gym operations efficiently.

It requires the storage and processing of key information including:

- User information (members and trainers)
- Plan details (membership types and services)
- Payment transactions (fee collection and billing)
- Attendance logs (daily check-ins and session tracking)

All data must be:

- Well-structured
- Relational
- Maintained with integrity for accurate reporting and smooth functionality

4.2 Entity Types, Entity Sets

The Gym Management System is structured around several core entities to efficiently manage operations. The primary entity types in the system include User, Plan, Payment, and Attendance. Each of these entities represents a crucial aspect of gym functionality. Corresponding to these types, the system maintains organized entity sets such as Users, Plans, Payments, and Attendance. The Users' entity set contains all registered members and staff, storing their personal and contact information. The Plans entity set includes details of various membership packages offered by the gym. Payments track all financial transactions related to memberships and services, while the Attendance entity set logs daily member check-ins and participation in sessions. These structured entity sets enable the system to ensure smooth data management, retrieval, and reporting, contributing to overall operational efficiency.

4.3 Attributes, and Keys

Attributes and Keys of Each Entity:

1. Users

Attributes : user_id(PK) , name , age , gender , phone , mail , role , join_date , specialization

Primary Key – user_id

2. Plans

Attributes : plan_id(PK) , plan_name , duration, price,

Primary Key – plan_id

3. Attendance

Attributes : attendance_id(PK) , user_id, date

Primary Key – attendance_id

4. Payments

Attributes : payment_id(PK) , user_id, plan_id , amount , payment_date

Primary Key – payment_id

4.4 Relationships- Cardinality

User to Payment Relationship

- One user can make multiple payments.
- This represents a one-to-many (1:N) relationship.

User to Attendance Relationship

- One user can have multiple attendance records.
- This also represents a one-to-many (1:N) relationship.

Plan to User Relationship

- A single plan can be purchased by multiple users.
- This is another one-to-many (1:N) relationship.

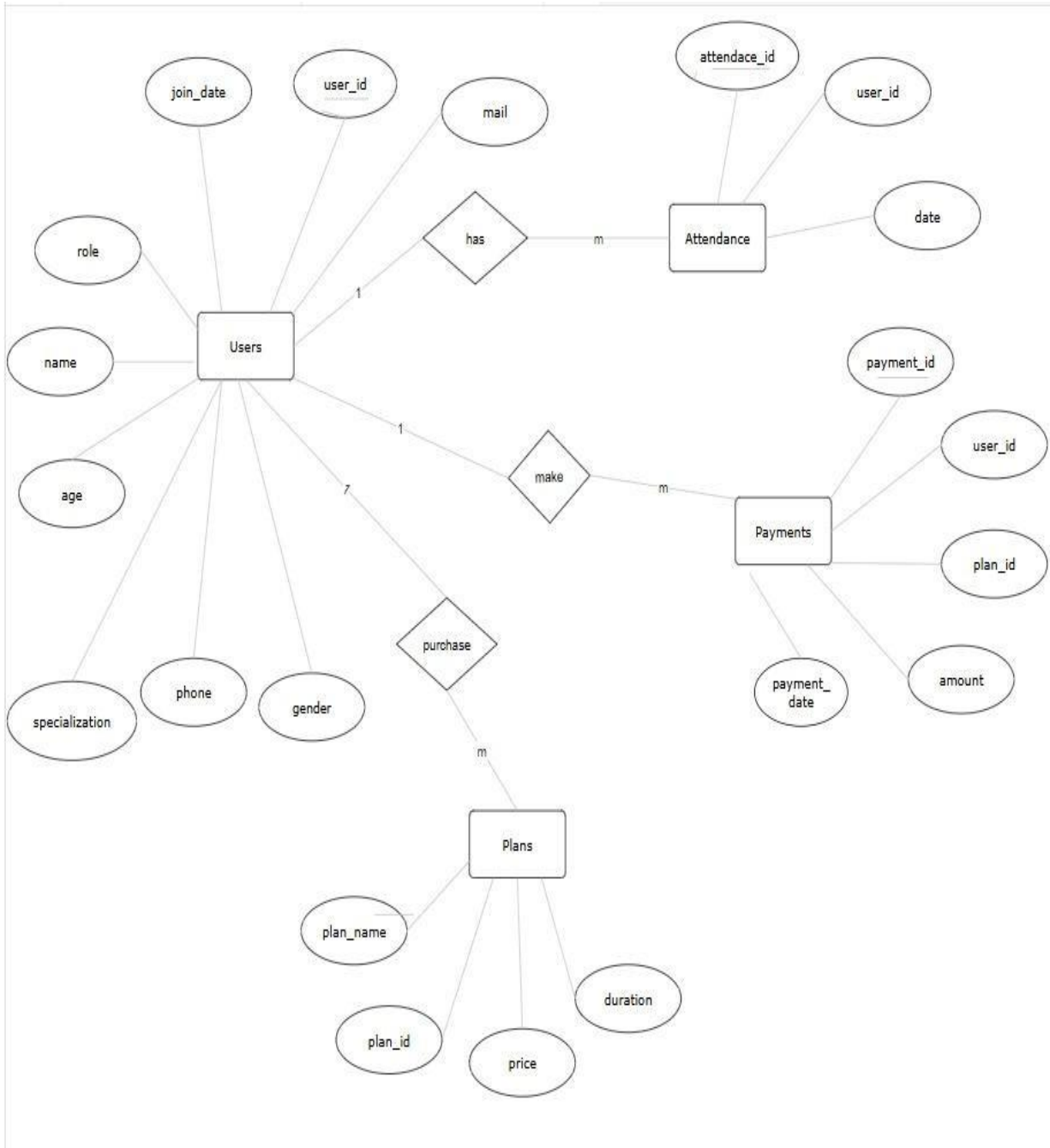
User to Role Relationship

- Each user is assigned exactly one role (e.g., member or trainer).
- This is a one-to-one (1:1) relationship.

Trainer to Specialization Relationship

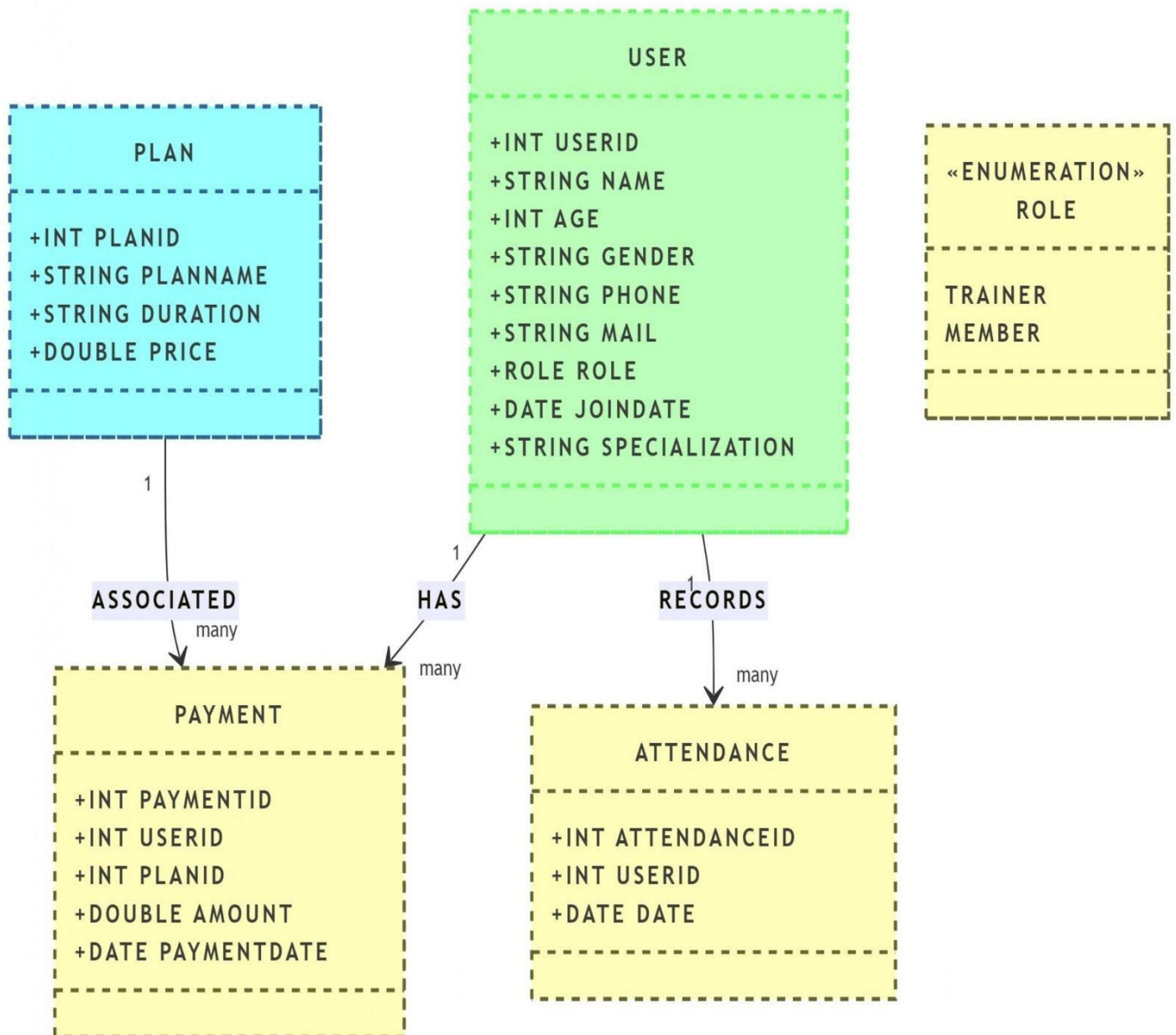
- Trainers may optionally have a specialization.
- This is a one-to-one (1:1) conditional relationship.

5. E-R Diagram



E-R Diagram of Movie Ticket Booking Management System

6. SCHEMA DIAGRAM



Schema Diagram of Gym Management System

7. RELATIONAL DATABASE DESIGN

Relational Database Design of the Gym Management System involves organizing data into structured tables based on the ER diagram, ensuring data integrity, minimizing redundancy, and maintaining consistency. The design includes tables such as Users, Plans, Payments, Attendance, and Trainers, each with appropriate relationships and constraints. For example, the Users table stores member and trainer information, while the Plans table outlines membership options. The Payments table records transaction history, and the Attendance table logs daily check-ins. By implementing a relational structure, the system ensures efficient data retrieval, supports accurate reporting, and maintains the integrity of gym operations.

Relational Database Design :-

The Gym Management System is designed using the principles of relational database design. It consists of multiple interrelated tables—such as Users, Plans, Payments, Attendance, and Trainers—each representing a specific entity or relationship within the system, as derived from the ER diagram. The database schema is normalized to at least the Third Normal Form (3NF) to eliminate redundancy, maintain data consistency, and ensure data integrity. This structured design allows for efficient data storage, seamless updates, and accurate reporting across all gym operations.

1. Key Requirements of the System

To design a relational database for a Gym Management System, we need to identify core entities and their relationships. Major system components include:

- Members
- Membership Plans
- Staff (trainers and other employees)
- Equipment
- Workout Plans
- Payments
- Attendance

Gym Management System

2. Entity-Relationship (ER) Diagram Overview

Before translating into tables, an ER diagram is typically used to visualize relationships.

Major Entities:

- Member
- Trainer
- Membership
- Workout Plan
- Equipment
- Payment
- Session

Relationships:

- A Member is assigned a Membership.
- A Member may be assigned to one or more Workout Plans.
- A Trainer may guide multiple Members.
- A Payment is linked to a member.
- A Session links a Member with a Trainer at a specific time.

Gym Management System

3. Table Design (Schema)

Here's a relational schema for the database with sample fields.

Table: Members

Field Name	Data Type	Description
member_id	INT (PK)	Unique ID for member
name	VARCHAR(100)	Full name
email	VARCHAR(100)	Email address
phone	VARCHAR(15)	Contact number
address	TEXT	Residential address
join_date	DATE	Membership start date
membership_id	INT (FK)	Linked to Memberships table

Table: Memberships

Field Name	Data Type	Description
membership_id	INT (PK)	Unique ID
type	VARCHAR(50)	Plan name (Basic, Premium, etc.)
duration_months	INT	Duration in months
price	DECIMAL(10,2)	Cost of the membership

Gym Management System

Table: Trainers

Field Name	Data Type	Description
trainer_id	INT (PK)	Unique trainer ID
name	VARCHAR(100)	Trainer's full name
specialization	VARCHAR(100)	Area of expertise
phone	VARCHAR(15)	Contact number
email	VARCHAR(100)	Email address

Table: Workout_Plans

Field Name	Data Type	Description
plan_id	INT (PK)	Unique ID for workout plan
name	VARCHAR(100)	Plan name
description	TEXT	Plan details
duration_weeks	INT	Duration of the plan

Gym Management System

Table: Equipment

Field Name	Data Type	Description
equipment_id	INT (PK)	Unique ID
name	VARCHAR(100)	Name of equipment
category	VARCHAR(50)	Type (cardio, strength, etc.)
purchase_date	DATE	Date of purchase
status	VARCHAR(20)	Working, Under Maintenance

Table: Payments

Field Name	Data Type	Description
payment_id	INT (PK)	Unique payment ID
member_id	INT (FK)	Linked to Members table
amount	DECIMAL(10,2)	Payment amount
payment_date	DATE	Date of payment
method	VARCHAR(50)	Cash, Card, Online

Gym Management System

Table: Sessions

Field Name	Data Type	Description
session_id	INT (PK)	Unique session ID
member_id	INT (FK)	Linked to Members
trainer_id	INT (FK)	Linked to Trainers
session_date	DATE	Date of session
start_time	TIME	Start time
end_time	TIME	End time

4. Normalization

The database design follows normalization rules to:

- Eliminate redundancy
- Ensure data integrity
- Improve query performance

8. CREATING DATABASE USING MYSQL

```
CREATE DATABASE GYM_MANAGEMENT_SYSTEM;
```

```
USE GYM_MANAGEMENT_SYSTEM;
```

```
-- Users Table
```

```
CREATE TABLE users (  
    user_id INT AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(100),  
    age INT,  
    gender VARCHAR(10),  
    phone VARCHAR(15),  
    mail VARCHAR(100),  
    role ENUM('trainer', 'member') NOT NULL,  
    join_date DATE,  
    specialization VARCHAR(100)  
);
```

```
-- Plans Table
```

```
CREATE TABLE plans (  
    plan_id INT AUTO_INCREMENT PRIMARY KEY,  
    plan_name VARCHAR(100),  
    duration VARCHAR(50),  
    price DECIMAL(10, 2)  
);
```

```
-- payments Table
```

```
CREATE TABLE payments (  
    payment_id INT AUTO_INCREMENT PRIMARY KEY,  
    user_id INT,  
    plan_id INT,  
    amount DECIMAL(10, 2),  
    payment_date DATE,  
    FOREIGN KEY (user_id) REFERENCES users(user_id),  
    FOREIGN KEY (plan_id) REFERENCES plans(plan_id)  
);
```

```
-- attendance Table
```

```
CREATE TABLE attendance (  
    attendance_id INT AUTO_INCREMENT PRIMARY KEY,  
    user_id INT,  
    date DATE,  
    FOREIGN KEY (user_id) REFERENCES users(user_id)  
);
```

8. Test Queries (Minimum 25 Queries)

```
mysql> -- 1. Get all users over the age of 30
```

```
mysql> SELECT * FROM users WHERE age > 30;
```

user_id	name	age	gender	phone	mail	role	join_date	specialization
4	Olivia Davis	32	Female	9876543213	olivia.davis@example.com	trainer	2022-11-20	Yoga
7	Noah Moore	35	Male	9876543216	noah.moore@example.com	member	2023-05-15	Pilates
9	Ethan Anderson	31	Male	9876543218	ethan.anderson@example.com	member	2023-06-01	Kickboxing
11	Lucas Jackson	34	Male	9876543220	lucas.jackson@example.com	trainer	2023-01-10	Strength Training
17	James Martinez	33	Male	9876543226	james.m@example.com	member	2023-05-20	Yoga
19	Benjamin Clark	36	Male	9876543228	ben.clark@example.com	trainer	2023-03-11	Strength Training
21	Alexander Lewis	31	Male	9876543230	alex.lewis@example.com	trainer	2023-01-25	General Fitness
25	Matthew Allen	35	Male	9876543234	matt.allen@example.com	member	2023-04-12	Weight Loss
29	David Wright	32	Male	9876543238	david.w@example.com	member	2023-02-14	Bodybuilding
34	Grace Adams	31	Female	9876543243	grace.a@example.com	trainer	2023-04-27	Weight Loss
37	Jack Carter	34	Male	9876543246	jack.c@example.com	trainer	2023-02-20	Cardio
44	Nathan Parker	32	Male	9876543253	nathan.p@example.com	trainer	2023-01-11	Weight Loss
48	Mason Bryant	31	Male	9876543257	mason.b@example.com	trainer	2023-05-11	Kickboxing

```
13 rows in set (0.00 sec)
```

```
mysql> -- 2. List all female members
```

```
mysql> SELECT * FROM users WHERE gender = 'Female' AND role = 'member';
```

user_id	name	age	gender	phone	mail	role	join_date	specialization
2	Emma Johnson	25	Female	9876543211	emma.j@example.com	member	2023-02-10	Cardio
10	Isabella Thomas	26	Female	9876543219	isabella.thomas@example.com	member	2023-03-18	General Fitness
12	Mia White	23	Female	9876543221	mia.white@example.com	member	2023-01-12	Cardio
14	Charlotte Martin	28	Female	9876543223	charlotte.m@example.com	member	2023-02-25	Bodybuilding
18	Harper Robinson	22	Female	9876543227	harper.r@example.com	member	2023-06-10	Cardio
20	Evelyn Rodriguez	30	Female	9876543229	evelyn.r@example.com	member	2023-04-18	Zumba
22	Abigail Lee	26	Female	9876543231	abigail.lee@example.com	member	2023-02-07	CrossFit
24	Emily Hall	24	Female	9876543233	emily.h@example.com	member	2023-04-01	Yoga
30	Ella Lopez	26	Female	9876543239	ella.l@example.com	member	2023-01-19	Cardio
32	Scarlett Scott	23	Female	9876543241	scarlett.s@example.com	member	2023-02-18	Yoga
36	Chloe Nelson	27	Female	9876543245	chloe.n@example.com	member	2023-06-05	Aerobics
38	Lily Mitchell	26	Female	9876543247	lily.m@example.com	member	2023-03-03	General Fitness
40	Aria Roberts	24	Female	9876543249	aria.r@example.com	member	2023-04-16	Bodybuilding
43	Layla Campbell	25	Female	9876543252	layla.c@example.com	member	2023-02-13	CrossFit
45	Zoe Price	27	Female	9876543254	zoe.p@example.com	member	2023-02-22	Pilates
47	Nora Hughes	26	Female	9876543256	nora.h@example.com	member	2023-04-19	Yoga
49	Stella Rivera	24	Female	9876543258	stella.r@example.com	member	2023-06-08	Zumba

```
17 rows in set (0.00 sec)
```

```
mysql> -- 4. List users who joined in 2023
```

```
mysql> SELECT * FROM users
```

```
  -> WHERE YEAR(join_date) = 2022;
```

user_id	name	age	gender	phone	mail	role	join_date	specialization
4	Olivia Davis	32	Female	9876543213	olivia.davis@example.com	trainer	2022-11-20	Yoga
8	Ava Taylor	24	Female	9876543217	ava.taylor@example.com	trainer	2022-12-30	Aerobics

```
2 rows in set (0.00 sec)
```

```
mysql> -- 3. Get count of trainers per specialization
```

```
mysql> SELECT specialization, COUNT(*) AS total_trainers
```

```
  -> FROM users
```

```
  -> WHERE role = 'trainer'
```

```
  -> GROUP BY specialization;
```

specialization	total_trainers
Strength Training	3
Yoga	1
Zumba	2
Aerobics	2
Weight Loss	3
Pilates	3
General Fitness	1
CrossFit	1
Cardio	1
Kickboxing	2
Bodybuilding	1

```
11 rows in set (0.02 sec)
```

```
mysql> -- 5. Get top 3 oldest users
```

```
mysql> SELECT * FROM users
```

```
  -> ORDER BY age DESC
```

```
  -> LIMIT 3;
```

user_id	name	age	gender	phone	mail	role	join_date	specialization
19	Benjamin Clark	36	Male	9876543228	ben.clark@example.com	trainer	2023-03-11	Strength Training
25	Matthew Allen	35	Male	9876543234	matt.allen@example.com	member	2023-04-12	Weight Loss
7	Noah Moore	35	Male	9876543216	noah.moore@example.com	member	2023-05-15	Pilates

```
3 rows in set (0.00 sec)
```

```
mysql> -- 7. Get average amount paid by users
```

```
mysql> SELECT AVG(amount) AS avg_payment FROM payments;
```

avg_payment
1385.666667

```
1 row in set (0.00 sec)
```



```
mysql> -- 6. List all payment transactions made in the last 30 days
mysql> SELECT * FROM payments
-> WHERE payment_date >= CURDATE() - INTERVAL 500 DAY;
```

payment_id	user_id	plan_id	amount	payment_date
1	1	1	499.00	2024-01-10
2	2	2	899.00	2024-02-15
3	3	3	1299.00	2024-03-20
4	4	1	499.00	2024-01-25
5	5	4	1799.00	2024-04-05
6	6	2	899.00	2024-03-01
7	7	5	2999.00	2024-05-10
8	8	1	499.00	2024-06-01
9	9	3	1299.00	2024-02-08
10	10	4	1799.00	2024-03-11
11	11	2	899.00	2024-02-28
12	12	1	499.00	2024-01-20
13	13	5	2999.00	2024-04-22
14	14	3	1299.00	2024-03-05
15	15	2	899.00	2024-02-12
16	16	1	499.00	2024-01-08
17	17	4	1799.00	2024-04-17
18	18	5	2999.00	2024-05-18
19	19	3	1299.00	2024-03-23
20	20	2	899.00	2024-02-05
21	21	1	499.00	2024-01-13
22	22	4	1799.00	2024-04-03
23	23	3	1299.00	2024-03-29
24	24	2	899.00	2024-02-17
25	25	5	2999.00	2024-05-02
26	26	1	499.00	2024-01-31
27	27	4	1799.00	2024-04-10
28	28	2	899.00	2024-02-21
29	29	3	1299.00	2024-03-08
30	30	5	2999.00	2024-05-25

30 rows in set (0.00 sec)

```
mysql> -- 8. Find the user who made the highest payment
mysql> SELECT u.name, p.amount FROM payments p
  -> JOIN users u ON p.user_id = u.user_id
  -> ORDER BY p.amount DESC
  -> LIMIT 1;
```

name	amount
Noah Moore	2999.00

1 row in set (0.00 sec)

```
mysql> -- 9. List users with payments above ₹5000
mysql> SELECT u.name, p.amount FROM payments p
  -> JOIN users u ON p.user_id = u.user_id
  -> WHERE p.amount > 1000;
```

name	amount
Mike Brown	1299.00
Liam Wilson	1799.00
Noah Moore	2999.00
Ethan Anderson	1299.00
Isabella Thomas	1799.00
Logan Harris	2999.00
Charlotte Martin	1299.00
James Martinez	1799.00
Harper Robinson	2999.00
Benjamin Clark	1299.00
Abigail Lee	1799.00
Daniel Walker	1299.00
Matthew Allen	2999.00
Henry Hernandez	1799.00
David Wright	1299.00
Ella Lopez	2999.00

16 rows in set (0.00 sec)

```
mysql> -- 10. Count number of payments per plan
mysql> SELECT plan_id, COUNT(*) AS count FROM payments
      -> GROUP BY plan_id;
```

plan_id	count
1	7
2	7
3	6
4	5
5	5

5 rows in set (0.00 sec)

```
mysql> -- 11. Get all plans cheaper than ₹2000
mysql> SELECT * FROM plans WHERE price < 2000;
```

plan_id	plan_name	duration	price
1	Basic Monthly	1 Month	999.00
2	Standard Monthly	1 Month	1299.00
3	Premium Monthly	1 Month	1599.00
18	Weekend Warrior	1 Month	799.00
19	Yoga Only	3 Months	1999.00
20	Zumba Blast	3 Months	1999.00

6 rows in set (0.00 sec)

```
mysql> -- 12. List all plans sorted by price descending
mysql> SELECT * FROM plans
      -> ORDER BY price DESC;
```

plan_id	plan_name	duration	price
16	Family Pack	12 Months	24999.00
15	Couple Plan	12 Months	16999.00
12	Premium Yearly	12 Months	11999.00
11	Standard Yearly	12 Months	9999.00
17	Corporate Plan	6 Months	9999.00
10	Basic Yearly	12 Months	8999.00
14	Senior Citizen	12 Months	7499.00
9	Premium Half-Yearly	6 Months	6999.00
8	Standard Half-Yearly	6 Months	5999.00
7	Basic Half-Yearly	6 Months	4999.00
6	Premium Quarterly	3 Months	4199.00
13	Student Special	6 Months	3999.00
5	Standard Quarterly	3 Months	3499.00
4	Basic Quarterly	3 Months	2799.00
19	Yoga Only	3 Months	1999.00
20	Zumba Blast	3 Months	1999.00
3	Premium Monthly	1 Month	1599.00
2	Standard Monthly	1 Month	1299.00
1	Basic Monthly	1 Month	999.00
18	Weekend Warrior	1 Month	799.00

20 rows in set (0.00 sec)

Gym Management System

```
mysql> -- 13. Find plans with duration longer than 3 months
mysql> SELECT * FROM plans
      -> WHERE duration LIKE '%month%' AND CAST(SUBSTRING_INDEX(duration, ' ', 1) AS UNSIGNED) > 3;
```

plan_id	plan_name	duration	price
7	Basic Half-Yearly	6 Months	4999.00
8	Standard Half-Yearly	6 Months	5999.00
9	Premium Half-Yearly	6 Months	6999.00
10	Basic Yearly	12 Months	8999.00
11	Standard Yearly	12 Months	9999.00
12	Premium Yearly	12 Months	11999.00
13	Student Special	6 Months	3999.00
14	Senior Citizen	12 Months	7499.00
15	Couple Plan	12 Months	16999.00
16	Family Pack	12 Months	24999.00
17	Corporate Plan	6 Months	9999.00

11 rows in set (0.00 sec)

```
mysql> -- 14. Get total number of plans
mysql> SELECT COUNT(*) FROM plans;
```

COUNT(*)
20

1 row in set (0.00 sec)

```
mysql> -- 15. Find the most expensive plan
mysql> SELECT * FROM plans
      -> ORDER BY price DESC
      -> LIMIT 1;
```

plan_id	plan_name	duration	price
16	Family Pack	12 Months	24999.00

1 row in set (0.00 sec)

```
mysql> -- 16. List attendance of all users on a specific date
mysql> SELECT * FROM attendance
-> WHERE date = '2025-04-01';
```

attendance_id	user_id	date
1	1	2025-04-01
2	2	2025-04-01
3	3	2025-04-01
4	4	2025-04-01
5	5	2025-04-01
11	6	2025-04-01
12	7	2025-04-01
13	8	2025-04-01
14	9	2025-04-01
15	10	2025-04-01
31	1	2025-04-01
32	2	2025-04-01
33	3	2025-04-01
34	4	2025-04-01
35	5	2025-04-01
41	6	2025-04-01
42	7	2025-04-01
43	8	2025-04-01
44	9	2025-04-01
45	10	2025-04-01

20 rows in set (0.00 sec)

```
mysql> -- 17. Find total attendance records in the system
mysql> SELECT COUNT(*) AS total_attendance FROM attendance;
```

total_attendance
60

1 row in set (0.00 sec)

```
mysql> -- 18. Find users with at least 10 attendance records
mysql> SELECT u.name, COUNT(a.attendance_id) AS attendance_count
-> FROM users u
-> JOIN attendance a ON u.user_id = a.user_id
-> GROUP BY u.user_id
-> HAVING attendance_count >= 2;
```

name	attendance_count
John Smith	4
Emma Johnson	4
Mike Brown	4
Olivia Davis	4
Liam Wilson	4
Sophia Miller	4
Noah Moore	4
Ava Taylor	4
Ethan Anderson	4
Isabella Thomas	4
Lucas Jackson	4
Mia White	4
Logan Harris	4
Charlotte Martin	4
Jacob Thompson	4

15 rows in set (0.00 sec)

```
mysql> -- 20. Find users who attended in the last week
mysql> SELECT DISTINCT u.name FROM attendance a
-> JOIN users u ON a.user_id = u.user_id
-> WHERE a.date >= CURDATE() - INTERVAL 7 DAY;
Empty set (0.00 sec)
```

```
mysql> -- 19. Get attendance records for trainers only
mysql> SELECT a.* FROM attendance a
      -> JOIN users u ON a.user_id = u.user_id
      -> WHERE u.role = 'trainer';
```

attendance_id	user_id	date
1	1	2025-04-01
4	4	2025-04-01
6	1	2025-04-02
9	4	2025-04-02
11	6	2025-04-01
13	8	2025-04-01
16	6	2025-04-02
18	8	2025-04-02
21	11	2025-04-03
23	13	2025-04-03
26	11	2025-04-04
28	13	2025-04-04
31	1	2025-04-01
34	4	2025-04-01
36	1	2025-04-02
39	4	2025-04-02
41	6	2025-04-01
43	8	2025-04-01
46	6	2025-04-02
48	8	2025-04-02
51	11	2025-04-03
53	13	2025-04-03
56	11	2025-04-04
58	13	2025-04-04

```
24 rows in set (0.00 sec)
```


Gym Management System

```
mysql> -- 21. List users with no attendance or payment
mysql> SELECT * FROM users
-> WHERE user_id NOT IN (SELECT user_id FROM payments)
-> AND user_id NOT IN (SELECT user_id FROM attendance);
```

user_id	name	age	gender	phone	mail	role	join_date	specialization
31	Joseph Hill	30	Male	9876543240	joseph.h@example.com	trainer	2023-01-07	CrossFit
32	Scarlett Scott	23	Female	9876543241	scarlett.s@example.com	member	2023-02-18	Yoga
33	Samuel Green	28	Male	9876543242	sam.green@example.com	member	2023-03-22	Strength Training
34	Grace Adams	31	Female	9876543243	grace.a@example.com	trainer	2023-04-27	Weight Loss
35	Owen Baker	29	Male	9876543244	owen.b@example.com	member	2023-05-14	Zumba
36	Chloe Nelson	27	Female	9876543245	chloe.n@example.com	member	2023-06-05	Aerobics
37	Jack Carter	34	Male	9876543246	jack.c@example.com	trainer	2023-02-20	Cardio
38	Lily Mitchell	26	Female	9876543247	lily.m@example.com	member	2023-03-03	General Fitness
39	Sebastian Perez	28	Male	9876543248	seb.perez@example.com	trainer	2023-01-30	Kickboxing
40	Aria Roberts	24	Female	9876543249	aria.r@example.com	member	2023-04-16	Bodybuilding
41	Daniela Turner	30	Female	9876543250	daniela.t@example.com	trainer	2023-05-09	Pilates
42	Jayden Phillips	29	Male	9876543251	jayden.p@example.com	member	2023-06-07	Yoga
43	Layla Campbell	25	Female	9876543252	layla.c@example.com	member	2023-02-13	CrossFit
44	Nathan Parker	32	Male	9876543253	nathan.p@example.com	trainer	2023-01-11	Weight Loss
45	Zoe Price	27	Female	9876543254	zoe.p@example.com	member	2023-02-22	Pilates
46	Isaac Bennett	30	Male	9876543255	isaac.b@example.com	trainer	2023-03-16	Bodybuilding
47	Nora Hughes	26	Female	9876543256	nora.h@example.com	member	2023-04-19	Yoga
48	Mason Bryant	31	Male	9876543257	mason.b@example.com	trainer	2023-05-11	Kickboxing
49	Stella Rivera	24	Female	9876543258	stella.r@example.com	member	2023-06-08	Zumba

19 rows in set (0.00 sec)

```
mysql> -- 23. Compare number of trainers vs members
mysql> SELECT role, COUNT(*) AS count FROM users
-> GROUP BY role;
```

role	count
trainer	20
member	29

2 rows in set (0.00 sec)

```
mysql> -- 22. Find users who joined recently and made a payment
mysql> SELECT DISTINCT u.name FROM users u
      -> JOIN payments p ON u.user_id = p.user_id
      -> WHERE u.join_date > CURDATE() - INTERVAL 3000 DAY;
```

name
John Smith
Emma Johnson
Mike Brown
Olivia Davis
Liam Wilson
Sophia Miller
Noah Moore
Ava Taylor
Ethan Anderson
Isabella Thomas
Lucas Jackson
Mia White
Logan Harris
Charlotte Martin
Jacob Thompson
Amelia Garcia
James Martinez
Harper Robinson
Benjamin Clark
Evelyn Rodriguez
Alexander Lewis
Abigail Lee
Daniel Walker
Emily Hall
Matthew Allen
Elizabeth Young
Henry Hernandez
Sofia King
David Wright
Ella Lopez

```
30 rows in set (0.00 sec)
```

```
mysql> -- 24. Top 3 users who made the most payments
mysql> SELECT u.name, COUNT(p.payment_id) AS payment_count
  -> FROM users u
  -> JOIN payments p ON u.user_id = p.user_id
  -> GROUP BY u.user_id
  -> ORDER BY payment_count DESC
  -> LIMIT 3;
```

name	payment_count
John Smith	1
Emma Johnson	1
Mike Brown	1

3 rows in set (0.00 sec)

```
mysql> -- 25. List members who joined but never attended
mysql> SELECT * FROM users
  -> WHERE role = 'member'
  -> AND user_id NOT IN (SELECT user_id FROM attendance);
```

user_id	name	age	gender	phone	mail	role	join_date	specialization
17	James Martinez	33	Male	9876543226	james.m@example.com	member	2023-05-20	Yoga
18	Harper Robinson	22	Female	9876543227	harper.r@example.com	member	2023-06-10	Cardio
20	Evelyn Rodriguez	30	Female	9876543229	evelyn.r@example.com	member	2023-04-18	Zumba
22	Abigail Lee	26	Female	9876543231	abigail.lee@example.com	member	2023-02-07	CrossFit
24	Emily Hall	24	Female	9876543233	emily.h@example.com	member	2023-04-01	Yoga
25	Matthew Allen	35	Male	9876543234	matt.allen@example.com	member	2023-04-12	Weight Loss
27	Henry Hernandez	29	Male	9876543236	henry.h@example.com	member	2023-06-02	Kickboxing
29	David Wright	32	Male	9876543238	david.w@example.com	member	2023-02-14	Bodybuilding
30	Ella Lopez	26	Female	9876543239	ella.l@example.com	member	2023-01-19	Cardio
32	Scarlett Scott	23	Female	9876543241	scarlett.s@example.com	member	2023-02-18	Yoga
33	Samuel Green	28	Male	9876543242	sam.green@example.com	member	2023-03-22	Strength Training
35	Owen Baker	29	Male	9876543244	owen.b@example.com	member	2023-05-14	Zumba
36	Chloe Nelson	27	Female	9876543245	chloe.n@example.com	member	2023-06-05	Aerobics
38	Lily Mitchell	26	Female	9876543247	lily.m@example.com	member	2023-03-03	General Fitness
40	Aria Roberts	24	Female	9876543249	aria.r@example.com	member	2023-04-16	Bodybuilding
42	Jayden Phillips	29	Male	9876543251	jayden.p@example.com	member	2023-06-07	Yoga
43	Layla Campbell	25	Female	9876543252	layla.c@example.com	member	2023-02-13	CrossFit
45	Zoe Price	27	Female	9876543254	zoe.p@example.com	member	2023-02-22	Pilates
47	Nora Hughes	26	Female	9876543256	nora.h@example.com	member	2023-04-19	Yoga
49	Stella Rivera	24	Female	9876543258	stella.r@example.com	member	2023-06-08	Zumba

20 rows in set (0.00 sec)

```
mysql> -- 29. Total revenue earned from each type of plan duration
mysql> SELECT pl.duration, SUM(p.amount) AS total_revenue
-> FROM plans pl
-> JOIN payments p ON p.plan_id = pl.plan_id
-> GROUP BY pl.duration;
```

duration	total_revenue
1 Month	17580.00
3 Months	23990.00

```
2 rows in set (0.00 sec)
```

```
mysql> -- 27. Find average age of trainers and members
mysql> SELECT role, AVG(age) AS average_age
-> FROM users
-> GROUP BY role;
```

role	average_age
trainer	29.8500
member	27.5517

```
2 rows in set (0.00 sec)
```

```
mysql> -- 28. Plans not chosen by anyone
mysql> SELECT pl.plan_name
      -> FROM plans pl
      -> LEFT JOIN payments p ON pl.plan_id = p.plan_id
      -> WHERE p.plan_id IS NULL;
```

plan_name
Premium Quarterly
Basic Half-Yearly
Standard Half-Yearly
Premium Half-Yearly
Basic Yearly
Standard Yearly
Premium Yearly
Student Special
Senior Citizen
Couple Plan
Family Pack
Corporate Plan
Weekend Warrior
Yoga Only
Zumba Blast

```
15 rows in set (0.00 sec)
```

```
mysql> -- 26. Show each plan's name with the number of users subscribed
mysql> SELECT pl.plan_name, COUNT(pm.user_id) AS users_count
-> FROM plans pl
-> LEFT JOIN payments pm ON pl.plan_id = pm.plan_id
-> GROUP BY pl.plan_name;
```

plan_name	users_count
Basic Monthly	7
Standard Monthly	7
Premium Monthly	6
Basic Quarterly	5
Standard Quarterly	5
Premium Quarterly	0
Basic Half-Yearly	0
Standard Half-Yearly	0
Premium Half-Yearly	0
Basic Yearly	0
Standard Yearly	0
Premium Yearly	0
Student Special	0
Senior Citizen	0
Couple Plan	0
Family Pack	0
Corporate Plan	0
Weekend Warrior	0
Yoga Only	0
Zumba Blast	0

```
20 rows in set (0.00 sec)
```

9. Conclusion

The Gym Management System is designed to streamline and automate the daily operations of a gym, including member registration, workout and diet planning, trainer assignments, attendance tracking, and payment management. By implementing this system, gyms can reduce manual errors, enhance operational efficiency, and provide a better experience for both members and staff. The use of a relational database ensures that all data is well-organized, consistent, and easily accessible.

With the ability to manage memberships, schedule sessions, monitor equipment usage, and track payments, the system supports both administrative needs and business growth.

In conclusion, this Gym Management System not only helps maintain accurate records and optimize resources but also plays a vital role in supporting the health and fitness goals of its members through a more structured and personalized approach.

10. References

1. Database Management Systems – Raghu Ramakrishnan, Johannes Gehrke
2. Fundamentals of Database Systems – Ramez Elmasri, Shamkant B. Navathe
3. Head First Java – Kathy Sierra, Bert Bates
4. Java: The Complete Reference – Herbert Schildt
5. MySQL Documentation – <https://dev.mysql.com/doc/>

