SVKM’s NMIMS
School of Technology Management & Engineering, Navi Mumbai
A.Y. 2023 - 24
Course: Database Management Systems

Project Report

Program	B.tech Computer Engineering	
Semester	4	
Name of the Project:	Gym Database Management System	
Details of Project Members		
Batch	Roll No.	Name
B2	A099	Aarushi Jain
Date of Submission: 02-04-2024		

Contribution of each project Members:

Roll No.	Name:	Contribution
A099	Aarushi Jain	Full Project

Github link of your project: https://github.com/aarushij978/DBMS-project

Note:

1. Create a readme file if you have multiple files
2. All files must be properly named (Example:R004_DBMSProject)
3. Submit all relevant files of your work (Report, all SQL files, Any other files)
4. **Plagiarism is highly discouraged (Your report will be checked for plagiarism)**

Rubrics for the Project evaluation:

First phase of evaluation: Innovative Ideas (5 Marks) Design and Partial implementation (5 Marks)	10 marks
Final phase of evaluation Implementation, presentation and viva, Self-Learning and Learning Beyond classroom	10 marks

SVKM'S NMIMS Deemed to be UNIVERSITY | NAVI MUMBAI

PROJECT REPORT

GYM DATABASE MANAGEMENT SYSTEM

By

Aarushi Jain, Roll No.: A099

Course: DBMS

AY:2023-24

Table of Contents

Sr no.	Topic	Page no.
1	Storyline	4
2	Components of Database Design	4
3	Entity Relationship Diagram	8
4	Relational Model	9
5	Normalization	10
6	SQL Queries	11
7	Learning from the Project	26
8	Project Demonstration	27
9	Self-learning beyond classroom	27
10	Learning from the project	27
8	Challenges faced	27
9	Conclusion	27

I. Storyline

FlexFit is a modern gym aiming to provide top-notch fitness services to its members. It's a vibrant hub for individuals of all fitness levels to come together and achieve their health goals with state-of-the-art equipment, expert trainers, and a supportive atmosphere. To streamline operations and enhance member experience, FlexFit plans to implement a comprehensive gym management system. The system, named FlexFit Gym Management System, will efficiently handle membership plans, member records, class schedules, and various administrative tasks.

II. Components of Database Design

Entities: plans, Members, BasicPlan, StandardPlan, PremiumPlan, CouplePlan, StudentPlan, Trainers, Classes, Payments

1. plans:

plans
<u>PlanID</u>
PlanName

- Cardinality: 5

2. Members:

Members
<u>MemberID</u>
FirstName
LastName
PhoneNo
Age
Gender
PlanName
PlanID

- Cardinality: 50

3. Trainers:

Members
<u>TrainerID</u>
Name
PhoneNo
PlanName
PlanID
Specialization
Age
Gender

- Cardinality: 9

4. Classes:

Classes
<u>ClassID</u>
ClassName
TrainerID
Duration

- Cardinality: 9

5. Payments:

Payments
<u>PaymentID</u>
MemberID
PaymentDate
PaymentStatus
PlanID

- Cardinality: 50

6. BasicPlan:

Members
<u>MemberID</u>
FirstName
LastName
PhoneNo
Age
Gender
PlanName
PlanID

- Cardinality: 20

7. StandardPlan:

Members
<u>MemberID</u>
FirstName
LastName
PhoneNo
Age
Gender
PlanName
PlanID

- Cardinality: 13

8. PremiumPlan:

Members
<u>MemberID</u>
FirstName
LastName
PhoneNo
Age
Gender
PlanName
PlanID

- Cardinality: 12

9. CouplePlan:

Members
<u>MemberID</u>
FirstName
LastName
PhoneNo
Age
Gender
PlanName
PlanID
PartnerName
PartnerAge
PartnerGender

- Cardinality: 1

10. StudentPlan:

Members
<u>MemberID</u>
FirstName
LastName
PhoneNo
Age
Gender
PlanName
PlanID

- Cardinality: 4

Relationships:

1. Members-Plans (One-to-Many):

   - Each member can have only one plan, but each plan can have multiple members.

2. Trainers - Plans (One-to-Many):

   - Each trainer can be associated with only one plan, but each plan can have multiple trainers.

3. Classes - Trainers (Many-to-One):

   - Each class is conducted by one trainer, but a trainer can conduct multiple classes.

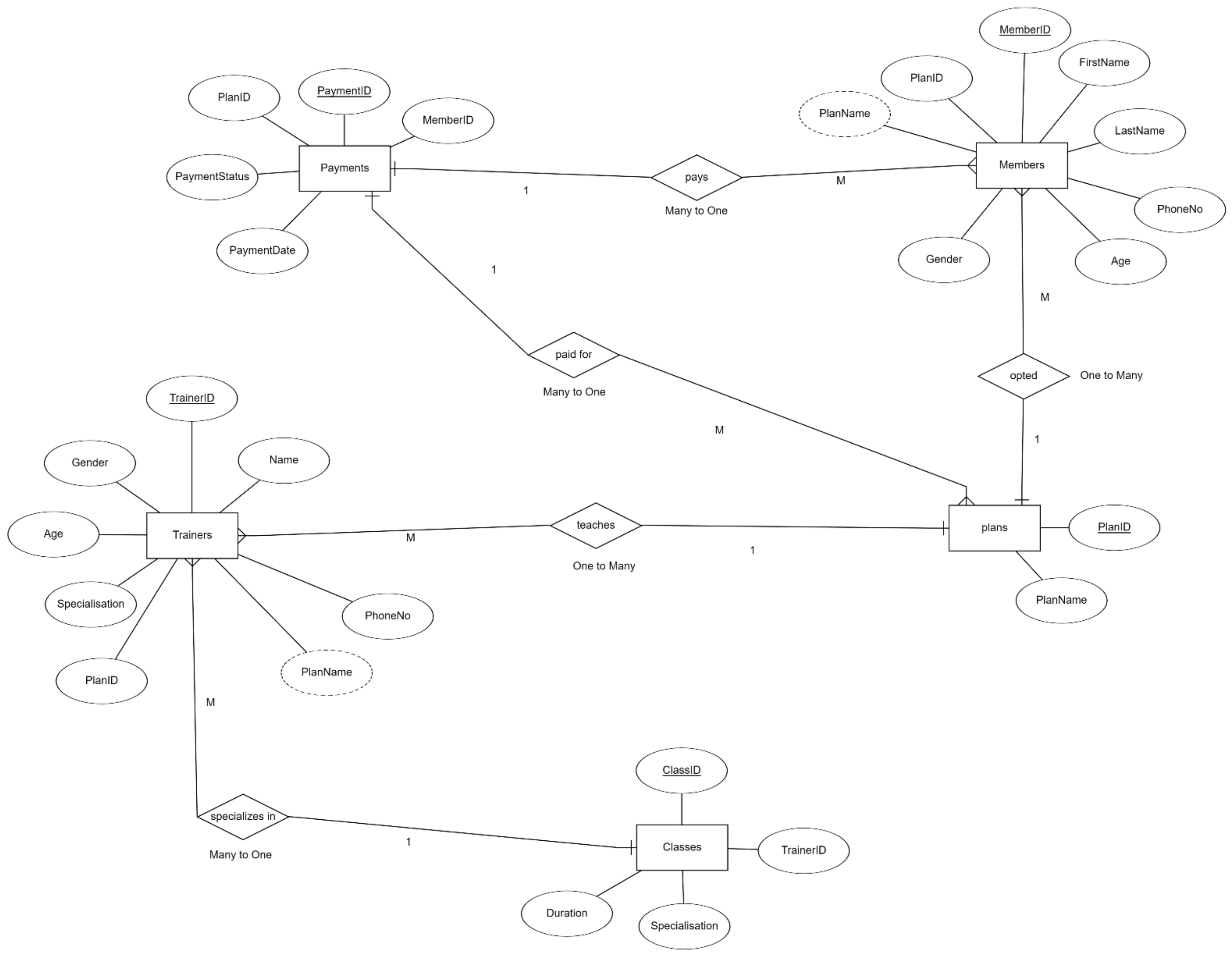
4. Payments - Members (Many-to-One):

   - Each payment is made by one member, but a member can have multiple payments.

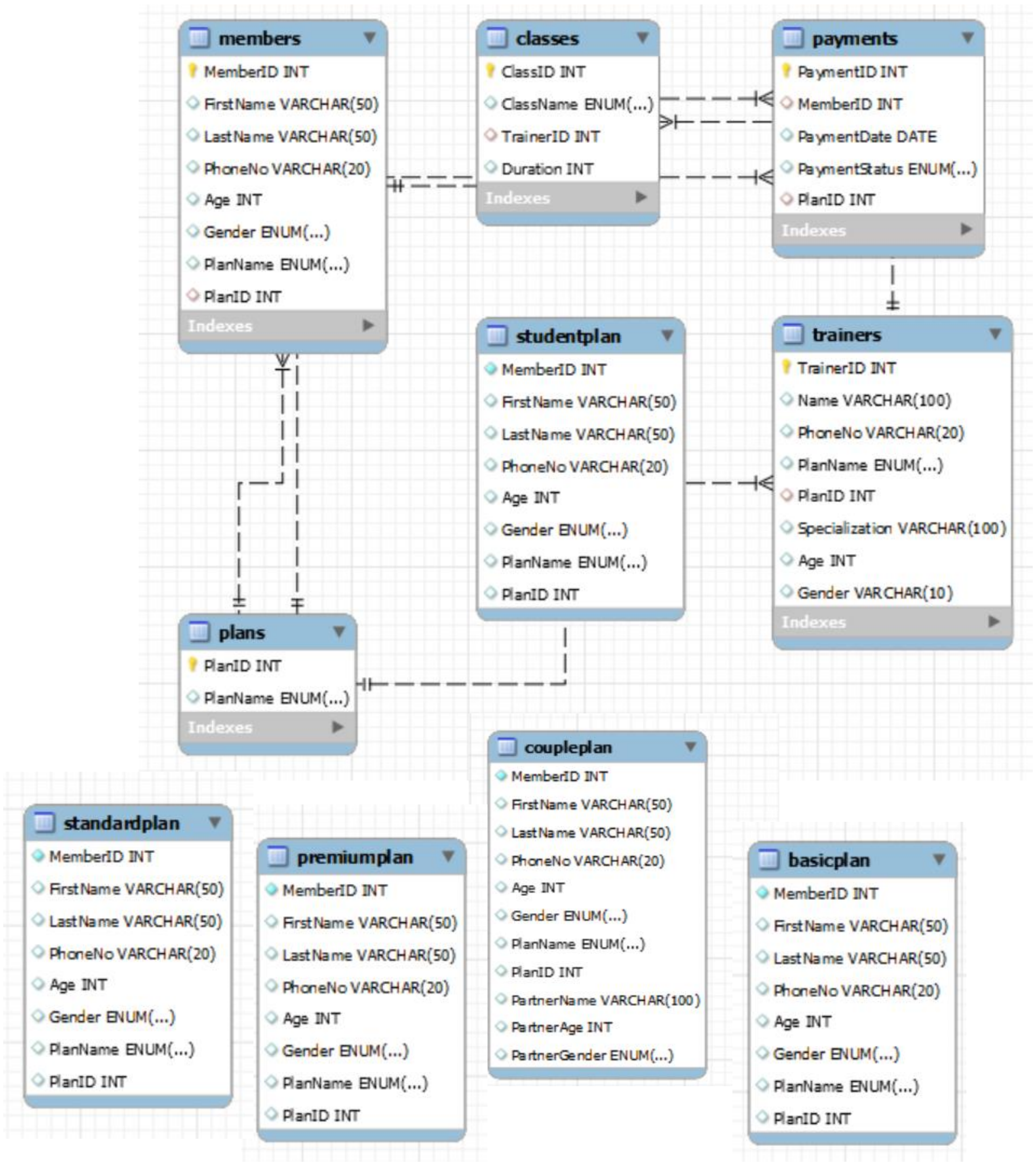
5. Payments - Plans (Many-to-One):

   - Each payment is associated with one plan, but a plan can have multiple payments.

III. Entity Relationship Diagram



IV. Relational Model



V. Normalization

1. plans:

- This table meets all normal forms as it has:
  - Single valued attributes (1NF).
  - Primary key (PlanID) that uniquely identifies each plan (1NF).
  - No partial dependencies on the primary key (2NF).
  - No transitive dependencies (3NF).
  - The determinant (PlanID) is the whole primary key itself (BCNF).

2. Members :

- Single valued attributes (1NF).
- Primary key (PlanID) that uniquely identifies each plan (1NF).
- No partial dependencies on the primary key (2NF).

3. Classes:

- Single valued attributes (1NF).
- Primary key (PlanID) that uniquely identifies each plan (1NF).
- No partial dependencies on the primary key (2NF).

4. Trainers:

- Single valued attributes (1NF).
- Primary key (PlanID) that uniquely identifies each plan (1NF).
- No partial dependencies on the primary key (2NF).
- No transitive dependencies (3NF).

5. Payments:

- Single valued attributes (1NF).
- Primary key (PlanID) that uniquely identifies each plan (1NF).
- No partial dependencies on the primary key (2NF).
- No transitive dependencies (3NF).

VI. SQL Queries

Creating the tables and Populating the tables:

```
create database FlexFit;
use FlexFit;
show databases;

CREATE TABLE plans (
   PlanID INT PRIMARY KEY AUTO_INCREMENT,
   PlanName ENUM('Basic', 'Standard', 'Premium', 'Couple', 'Student')
);
INSERT INTO plans (PlanName) VALUES
('Basic'),
('Standard'),
('Premium'),
('Couple'),
('Student');
SELECT * FROM plans;



CREATE TABLE Members (
   MemberID INT PRIMARY KEY,
   FirstName VARCHAR(50),
LastName VARCHAR(50),
   PhoneNo VARCHAR(20),
   Age INT,
   Gender ENUM('Male', 'Female', 'Other'),
   PlanName ENUM('Basic', 'Standard', 'Premium', 'Couple', 'Student'),
   PlanID INT,
   FOREIGN KEY (PlanID) REFERENCES plans(PlanID)
);
INSERT INTO Members (MemberID, FirstName, LastName, PhoneNo, Age, Gender, PlanName, PlanID)
VALUES
(1, 'John', 'Doe', '1234567890', 30, 'Male', 'Basic', 1),
(2, 'Jane', 'Smith', '9876543210', 25, 'Female', 'Standard', 2),
(3, 'Michael', 'Johnson', '5683723727', 40, 'Male', 'Premium', 3),
(4, 'Emily', 'Davis', '8053083678', 35, 'Female', 'Standard', 2),
(5, 'Christopher', 'Brown', '5049444119', 28, 'Male', 'Basic', 1),
(6, 'Jessica', 'Wilson', '9344541826', 33, 'Female', 'Premium', 3),
(7, 'Matthew', 'Taylor', '7196402739', 45, 'Male', 'Standard', 2),
(8, 'Amanda', 'Martinez', '5864139699', 22, 'Female', 'Basic', 1),
(9, 'David', 'Anderson', '9778816901', 29, 'Male', 'Student', 5),
(10, 'Jennifer', 'Thomas', '7751815220', 31, 'Female', 'Basic', 1),
(11, 'James', 'Jackson', '9000228385', 27, 'Male', 'Standard', 2),
(12, 'Melissa', 'White', '6512792946', 32, 'Female', 'Premium', 3),
(13, 'Ryan', 'Harris', '8768380253', 26, 'Male', 'Basic', 1),
(14, 'Sarah', 'Clark', '2870815315', 24, 'Female', 'Standard', 2),
(15, 'Daniel', 'Lewis', '2251652275', 38, 'Male', 'Couple', 4),
(16, 'Laura', 'Turner', '6115114112', 36, 'Female', 'Basic', 1),
(17, 'Kevin', 'Martin', '8097248122', 34, 'Male', 'Standard', 2),
(18, 'Kimberly', 'Lee', '4665700669', 23, 'Female', 'Premium', 3),
(19, 'Justin', 'Perez', '7720334133', 41, 'Male', 'Basic', 1),
(20, 'Ashley', 'Nguyen', '1234567876', 39, 'Female', 'Premium', 3),
(21, 'Brandon', 'Robinson', '9876543456', 37, 'Male', 'Student', 5),
(22, 'Stephanie', 'Hall', '9872323575', 20, 'Female', 'Basic', 1),
(23, 'Andrew', 'Allen', '8765432345', 21, 'Male', 'Standard', 2),
(24, 'Nicole', 'King', '3456833467', 42, 'Female', 'Basic', 1),
(25, 'Robert', 'Scott', '7654322345', 43, 'Male', 'Premium', 3),
(26, 'Christina', 'Green', '6543234567', 44, 'Female', 'Standard', 2),
(27, 'William', 'Adams', '2345676543', 46, 'Male', 'Basic', 1),
(28, 'Megan', 'Baker', '3264786978', 47, 'Female', 'Premium', 3),
(29, 'Nicholas', 'Nelson', '6534354676', 48, 'Male', 'Basic', 1),
(30, 'Kayla', 'Rivera', '1237893456', 49, 'Female', 'Standard', 2),
(31, 'Zachary', 'Carter', '9876556789', 50, 'Male', 'Basic', 1),
(32, 'Vanessa', 'Torres', '1900956789', 51, 'Female', 'Premium', 3),
(33, 'Cody', 'Evans', '5798833961', 52, 'Male', 'Student', 5),
(34, 'Rebecca', 'Hughes', '3253267065', 53, 'Female', 'Basic', 1),
(35, 'Tyler', 'Long', '7226178674', 54, 'Male', 'Standard', 2),
(36, 'Brittany', 'Foster', '3758325044', 55, 'Female', 'Premium', 3),
(37, 'Austin', 'Diaz', '9647943412', 56, 'Male', 'Basic', 1),
(38, 'Hannah', 'Griffin', '6571131003', 57, 'Female', 'Standard', 2),
(39, 'Dylan', 'Russell', '7037798274 ', 58, 'Male', 'Basic', 1),
(40, 'Samantha', 'Diaz', '5201559505', 59, 'Female', 'Premium', 3),
(41, 'Jordan', 'Ward', '8243730048', 60, 'Male', 'Basic', 1),
(42, 'Maria', 'Bell', '8748020948', 61, 'Female', 'Standard', 2),
(43, 'Joshua', 'Price', '1804013581', 62, 'Male', 'Basic', 1),
(44, 'Lauren', 'Murphy', '3623178417', 63, 'Female', 'Premium', 3),
(45, 'Alex', 'Campbell', '5678323358', 64, 'Male', 'Basic', 1),
(46, 'Olivia', 'Foster', '8385367973', 65, 'Female', 'Standard', 2),
(47, 'Ethan', 'Richardson', '3279566109', 66, 'Male', 'Premium', 3),
(48, 'Grace', 'Diaz', '6158343963', 67, 'Female', 'Basic', 1),
(49, 'Noah', 'Bryant', '3210322449', 68, 'Male', 'Student', 5),
(50, 'Chloe', 'Sanders', '8001463499', 69, 'Female', 'Basic', 1);
SELECT * FROM Members;

CREATE TABLE BasicPlan AS
SELECT *
FROM members
WHERE PlanName = 'Basic';
SELECT * FROM BasicPlan;


CREATE TABLE StandardPlan AS
SELECT *
FROM members
WHERE PlanName = 'Standard';
SELECT * FROM StandardPlan;


CREATE TABLE PremiumPlan AS
SELECT *
FROM members
WHERE PlanName = 'Premium';
SELECT * FROM PremiumPlan;


CREATE TABLE CouplePlan AS
SELECT *
FROM members
WHERE PlanName = 'Couple';


CREATE TABLE StudentPlan AS
SELECT *
FROM members
```

```
WHERE PlanName = 'Student';
SELECT * FROM StudentPlan;

Show Tables;


CREATE TABLE Trainers (
    TrainerID INT PRIMARY KEY,
    Name VARCHAR(100),
    PhoneNo VARCHAR(20),
    PlanName ENUM('Standard', 'Premium', 'Couple', 'Student'),
    PlanID INT,
    Specialization VARCHAR(100),
    Age INT,
    Gender VARCHAR(10),
    FOREIGN KEY (PlanID) REFERENCES plans(PlanID)
);

INSERT INTO Trainers (TrainerID, Name, PhoneNo, PlanName, PlanID, Specialization, Age, Gender)
VALUES
(1, 'Sonia Dailey', '7048571096', 'Standard', 2, 'Zumba', 30, 'Female'),
(2, 'David Miller', '9276356499', 'Premium', 3, 'Yoga', 38, 'Male'),
(3, 'Sophia Garcia', '6758014676', 'Premium', 3, 'Pilates', 32, 'Female'),
(4, 'Daniel Martinez', '2664245272', 'Premium', 3, 'Boot camp', 40, 'Male'),
(5, 'Emily Brown', '3100894721', 'Premium', 3, 'Pilates', 40, 'Female'),
(6, 'Michael Davis', '9381389554', 'Premium', 3, 'Strength training', 45, 'Male'),
(7, 'Jessica Wilson', '9636032615', 'Premium', 3, 'Boot camp', 35, 'Female'),
(8, 'Nichole Lewis', '8122409578', 'Couple', 4, 'HIIT', 35, 'Male'),
(9, 'Alex Johnson', '9231475436', 'Student', 5, 'Yoga', 28, 'Male');
SELECT * FROM Trainers;

CREATE TABLE Classes (
    ClassID INT PRIMARY KEY AUTO_INCREMENT,
    Specialization ENUM('Pilates', 'Boot camp', 'Strength training', 'HIIT', 'Yoga', 'Zumba'),
    TrainerID INT,
    Duration INT,
    FOREIGN KEY (TrainerID) REFERENCES Trainers(TrainerID)
);
INSERT INTO Classes (Specialization, TrainerID, Duration)
VALUES
```

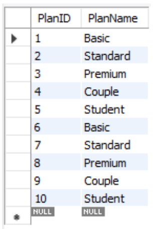
```
('Zumba', 1, 45),
('Yoga', 2, 60),
('Pilates', 3, 90),
('Boot camp', 4, 90),
('Pilates', 5, 90),
('Strength training', 6, 60),
('Boot camp', 7, 90),
('HIIT', 8, 60),
('Yoga', 9, 60);
SELECT * FROM Classes;


CREATE TABLE Payments (
    PaymentID INT PRIMARY KEY AUTO_INCREMENT,
    MemberID INT,
    PaymentDate DATE,
    PaymentStatus ENUM('Paid', 'Unpaid'),
    PlanID INT,
    FOREIGN KEY (MemberID) REFERENCES members(memberID),
    FOREIGN KEY (PlanID) REFERENCES plans(PlanID)
);

INSERT INTO Payments (MemberID, PaymentDate, PaymentStatus, PlanID)
VALUES
(1, '2024-01-23', 'Paid', 1),
(2, CURDATE(), 'Unpaid', 2),
(3, '2024-04-14', 'Paid', 3),
(4, '2024-03-01', 'Paid', 2),
(5, CURDATE(), 'Unpaid', 1),
(6, '2024-02-25', 'Paid', 3),
(7, CURDATE(), 'Unpaid', 2),
(8, '2024-02-12', 'Paid', 1),
(9, '2024-01-18', 'Paid', 5),
(10, '2024-01-10', 'Paid', 1),
(11, '2024-02-04', 'Paid', 2),
(12, CURDATE(), 'Unpaid', 3),
(13, CURDATE(), 'Unpaid', 1),
(14, '2024-03-02', 'Paid', 2),
(15, '2024-03-05', 'Paid', 4),
(16, '2024-03-17', 'Paid', 1),
(17, '2024-02-16', 'Paid', 2),
(18, '2024-02-04', 'Paid', 3),
(19, '2024-01-05', 'Paid', 1),
(20, '2024-01-09', 'Paid', 3),
(21, CURDATE(), 'Unpaid', 5),
(22, '2024-01-15', 'Paid', 1),
(23, '2024-01-26', 'Paid', 2),
(24, CURDATE(), 'Unpaid', 1),
(25, CURDATE(), 'Unpaid', 3),
(26, '2024-02-20', 'Paid', 2),
(27, '2024-04-23', 'Paid', 1),
(28, '2024-03-17', 'Paid', 3),
(29, '2024-02-06', 'Paid', 1),
(30, '2024-01-08', 'Paid', 2),
(31, '2024-02-16', 'Paid', 1),
(32, '2024-01-18', 'Paid', 3),
(33, '2024-02-11', 'Paid', 5),
(34, CURDATE(), 'Unpaid', 1),
(35, '2024-03-19', 'Paid', 2),
(36, '2024-02-04', 'Paid', 3),
(37, '2024-01-27', 'Paid', 1),
(38, CURDATE(), 'Unpaid', 2),
(39, '2024-01-25', 'Paid', 1),
(40, '2024-01-20', 'Paid', 3),
(41, '2024-02-22', 'Paid', 1),
(42, CURDATE(), 'Unpaid', 2),
(43, '2024-04-28', 'Paid', 1),
(44, '2024-03-01', 'Paid', 3),
(45, '2024-02-17', 'Paid', 1),
(46, '2024-01-18', 'Paid', 2),
(47, '2024-01-10', 'Paid', 3),
(48, '2024-01-17', 'Paid', 1),
(49, CURDATE(), 'Unpaid', 5),
(50, '2024-02-01', 'Paid', 1);
SELECT * FROM Payments;
```

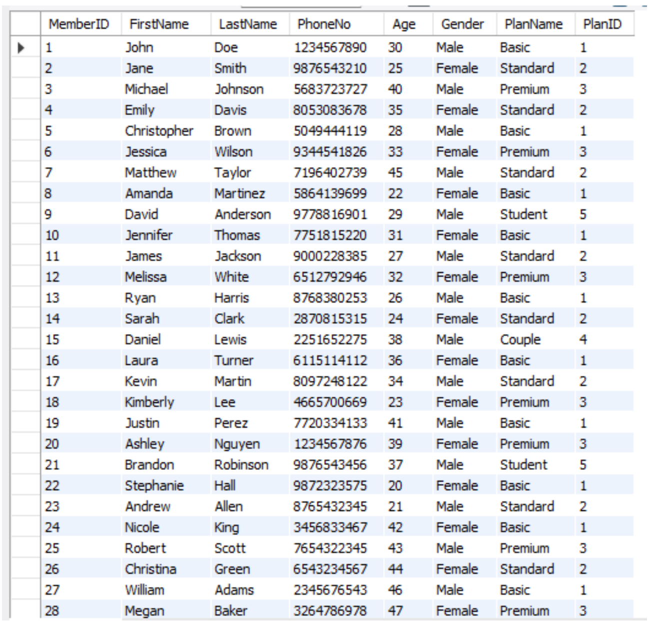
Output of Tables:

plans:

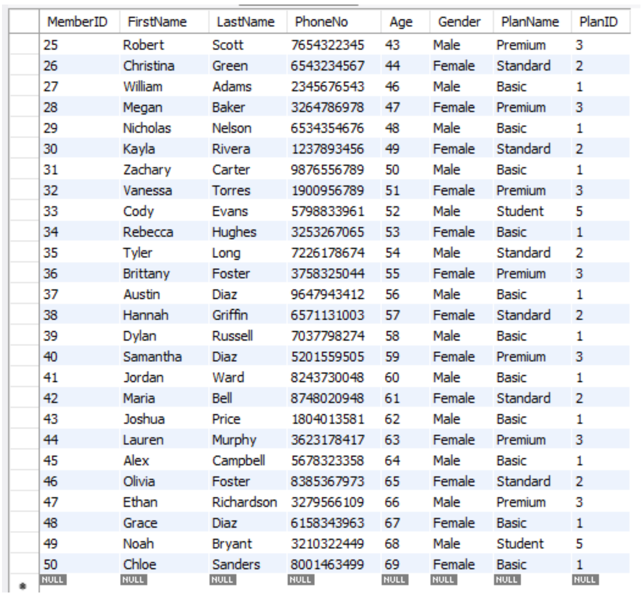


PlanID	PlanName
1	Basic
2	Standard
3	Premium
4	Couple
5	Student
6	Basic
7	Standard
8	Premium
9	Couple
10	Student
NULL	NULL

Members:

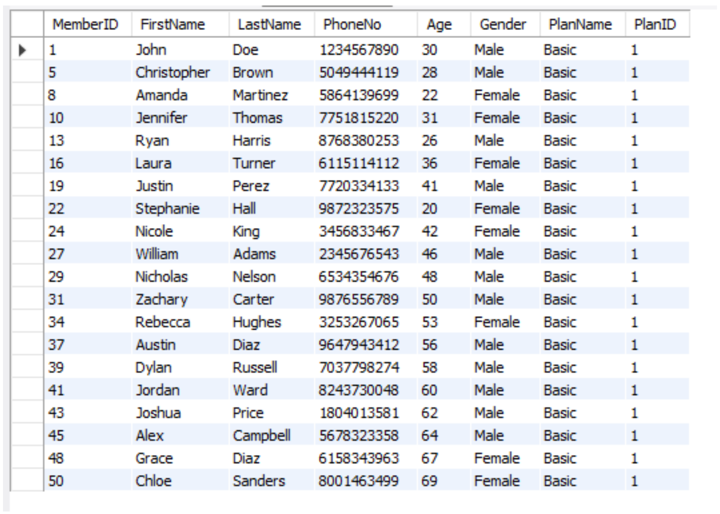


MemberID	FirstName	LastName	PhoneNo	Age	Gender	PlanName	PlanID
1	John	Doe	1234567890	30	Male	Basic	1
2	Jane	Smith	9876543210	25	Female	Standard	2
3	Michael	Johnson	5683723727	40	Male	Premium	3
4	Emily	Davis	8053083678	35	Female	Standard	2
5	Christopher	Brown	5049444119	28	Male	Basic	1
6	Jessica	Wilson	9344541826	33	Female	Premium	3
7	Matthew	Taylor	7196402739	45	Male	Standard	2
8	Amanda	Martinez	5864139699	22	Female	Basic	1
9	David	Anderson	9778816901	29	Male	Student	5
10	Jennifer	Thomas	7751815220	31	Female	Basic	1
11	James	Jackson	9000228385	27	Male	Standard	2
12	Melissa	White	6512792946	32	Female	Premium	3
13	Ryan	Harris	8768380253	26	Male	Basic	1
14	Sarah	Clark	2870815315	24	Female	Standard	2
15	Daniel	Lewis	2251652275	38	Male	Couple	4
16	Laura	Turner	6115114112	36	Female	Basic	1
17	Kevin	Martin	8097248122	34	Male	Standard	2
18	Kimberly	Lee	4665700669	23	Female	Premium	3
19	Justin	Perez	7720334133	41	Male	Basic	1
20	Ashley	Nguyen	1234567876	39	Female	Premium	3
21	Brandon	Robinson	9876543456	37	Male	Student	5
22	Stephanie	Hall	9872323575	20	Female	Basic	1
23	Andrew	Allen	8765432345	21	Male	Standard	2
24	Nicole	King	3456833467	42	Female	Basic	1
25	Robert	Scott	7654322345	43	Male	Premium	3
26	Christina	Green	6543234567	44	Female	Standard	2
27	William	Adams	2345676543	46	Male	Basic	1
28	Megan	Baker	3264786978	47	Female	Premium	3



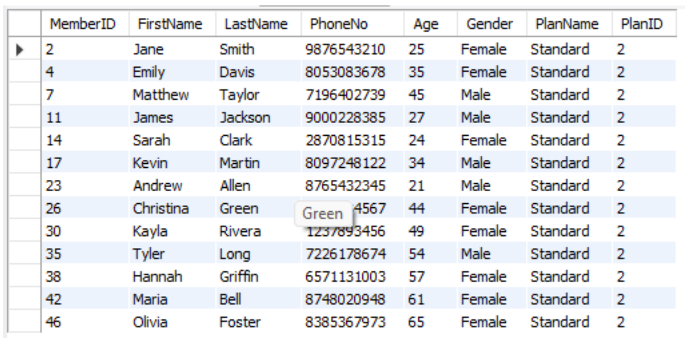
MemberID	FirstName	LastName	PhoneNo	Age	Gender	PlanName	PlanID
25	Robert	Scott	7654322345	43	Male	Premium	3
26	Christina	Green	6543234567	44	Female	Standard	2
27	William	Adams	2345676543	46	Male	Basic	1
28	Megan	Baker	3264786978	47	Female	Premium	3
29	Nicholas	Nelson	6534354676	48	Male	Basic	1
30	Kayla	Rivera	1237893456	49	Female	Standard	2
31	Zachary	Carter	9876556789	50	Male	Basic	1
32	Vanessa	Torres	1900956789	51	Female	Premium	3
33	Cody	Evans	5798833961	52	Male	Student	5
34	Rebecca	Hughes	3253267065	53	Female	Basic	1
35	Tyler	Long	7226178674	54	Male	Standard	2
36	Brittany	Foster	3758325044	55	Female	Premium	3
37	Austin	Diaz	9647943412	56	Male	Basic	1
38	Hannah	Griffin	6571131003	57	Female	Standard	2
39	Dylan	Russell	7037798274	58	Male	Basic	1
40	Samantha	Diaz	5201559505	59	Female	Premium	3
41	Jordan	Ward	8243730048	60	Male	Basic	1
42	Maria	Bell	8748020948	61	Female	Standard	2
43	Joshua	Price	1804013581	62	Male	Basic	1
44	Lauren	Murphy	3623178417	63	Female	Premium	3
45	Alex	Campbell	5678323358	64	Male	Basic	1
46	Olivia	Foster	8385367973	65	Female	Standard	2
47	Ethan	Richardson	3279566109	66	Male	Premium	3
48	Grace	Diaz	6158343963	67	Female	Basic	1
49	Noah	Bryant	3210322449	68	Male	Student	5
50	Chloe	Sanders	8001463499	69	Female	Basic	1
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

BasicPlan:



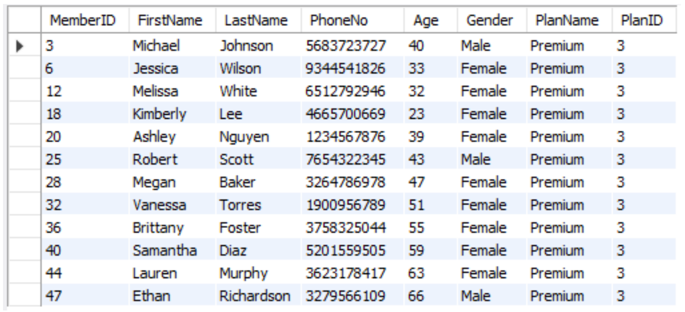
MemberID	FirstName	LastName	PhoneNo	Age	Gender	PlanName	PlanID
1	John	Doe	1234567890	30	Male	Basic	1
5	Christopher	Brown	5049444119	28	Male	Basic	1
8	Amanda	Martinez	5864139699	22	Female	Basic	1
10	Jennifer	Thomas	7751815220	31	Female	Basic	1
13	Ryan	Harris	8768380253	26	Male	Basic	1
16	Laura	Turner	6115114112	36	Female	Basic	1
19	Justin	Perez	7720334133	41	Male	Basic	1
22	Stephanie	Hall	9872323575	20	Female	Basic	1
24	Nicole	King	3456833467	42	Female	Basic	1
27	William	Adams	2345676543	46	Male	Basic	1
29	Nicholas	Nelson	6534354676	48	Male	Basic	1
31	Zachary	Carter	9876556789	50	Male	Basic	1
34	Rebecca	Hughes	3253267065	53	Female	Basic	1
37	Austin	Diaz	9647943412	56	Male	Basic	1
39	Dylan	Russell	7037798274	58	Male	Basic	1
41	Jordan	Ward	8243730048	60	Male	Basic	1
43	Joshua	Price	1804013581	62	Male	Basic	1
45	Alex	Campbell	5678323358	64	Male	Basic	1
48	Grace	Diaz	6158343963	67	Female	Basic	1
50	Chloe	Sanders	8001463499	69	Female	Basic	1

StandardPlan:



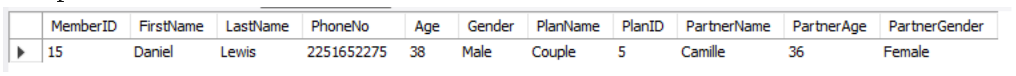
MemberID	FirstName	LastName	PhoneNo	Age	Gender	PlanName	PlanID
2	Jane	Smith	9876543210	25	Female	Standard	2
4	Emily	Davis	8053083678	35	Female	Standard	2
7	Matthew	Taylor	7196402739	45	Male	Standard	2
11	James	Jackson	9000228385	27	Male	Standard	2
14	Sarah	Clark	2870815315	24	Female	Standard	2
17	Kevin	Martin	8097248122	34	Male	Standard	2
23	Andrew	Allen	8765432345	21	Male	Standard	2
26	Christina	Green	[illegible]4567	44	Female	Standard	2
30	Kayla	Rivera	1237893456	49	Female	Standard	2
35	Tyler	Long	7226178674	54	Male	Standard	2
38	Hannah	Griffin	6571131003	57	Female	Standard	2
42	Maria	Bell	8748020948	61	Female	Standard	2
46	Olivia	Foster	8385367973	65	Female	Standard	2

PremiumPlan:



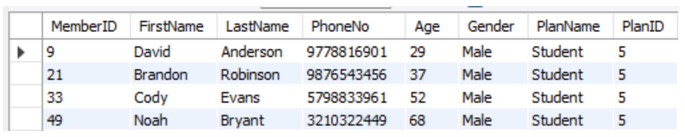
MemberID	FirstName	LastName	PhoneNo	Age	Gender	PlanName	PlanID
3	Michael	Johnson	5683723727	40	Male	Premium	3
6	Jessica	Wilson	9344541826	33	Female	Premium	3
12	Melissa	White	6512792946	32	Female	Premium	3
18	Kimberly	Lee	4665700669	23	Female	Premium	3
20	Ashley	Nguyen	1234567876	39	Female	Premium	3
25	Robert	Scott	7654322345	43	Male	Premium	3
28	Megan	Baker	3264786978	47	Female	Premium	3
32	Vanessa	Torres	1900956789	51	Female	Premium	3
36	Brittany	Foster	3758325044	55	Female	Premium	3
40	Samantha	Diaz	5201559505	59	Female	Premium	3
44	Lauren	Murphy	3623178417	63	Female	Premium	3
47	Ethan	Richardson	3279566109	66	Male	Premium	3

CouplePlan:



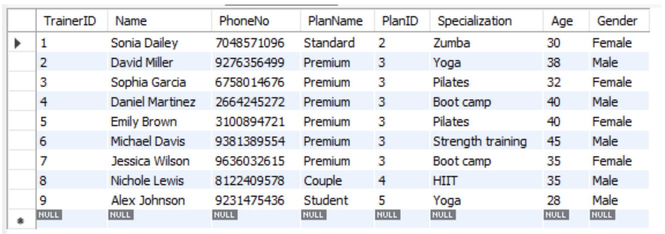
MemberID	FirstName	LastName	PhoneNo	Age	Gender	PlanName	PlanID	PartnerName	PartnerAge	PartnerGender
15	Daniel	Lewis	2251652275	38	Male	Couple	5	Camille	36	Female

StudentPlan:



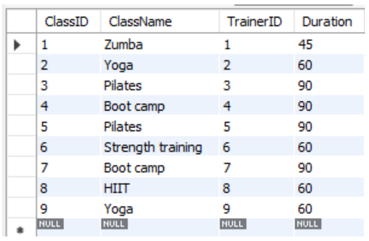
MemberID	FirstName	LastName	PhoneNo	Age	Gender	PlanName	PlanID
9	David	Anderson	9778816901	29	Male	Student	5
21	Brandon	Robinson	9876543456	37	Male	Student	5
33	Cody	Evans	5798833961	52	Male	Student	5
49	Noah	Bryant	3210322449	68	Male	Student	5

Trainers:



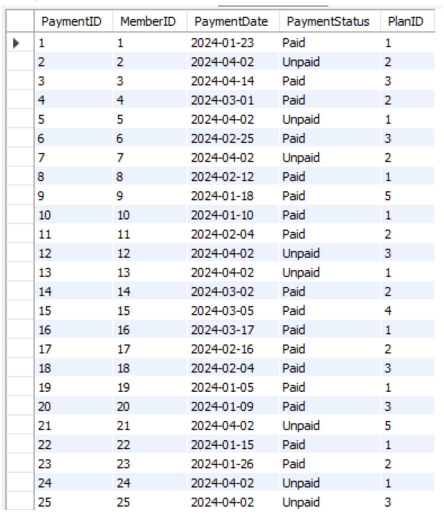
TrainerID	Name	PhoneNo	PlanName	PlanID	Specialization	Age	Gender
1	Sonia Dailey	7048571096	Standard	2	Zumba	30	Female
2	David Miller	9276356499	Premium	3	Yoga	38	Male
3	Sophia Garcia	6758014676	Premium	3	Pilates	32	Female
4	Daniel Martinez	2664245272	Premium	3	Boot camp	40	Male
5	Emily Brown	3100894721	Premium	3	Pilates	40	Female
6	Michael Davis	9381389554	Premium	3	Strength training	45	Male
7	Jessica Wilson	9636032615	Premium	3	Boot camp	35	Female
8	Nichole Lewis	8122409578	Couple	4	HIIT	35	Male
9	Alex Johnson	9231475436	Student	5	Yoga	28	Male
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Classes:

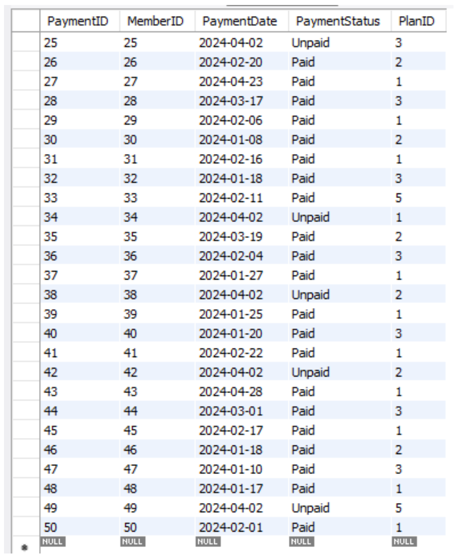


ClassID	ClassName	TrainerID	Duration
1	Zumba	1	45
2	Yoga	2	60
3	Pilates	3	90
4	Boot camp	4	90
5	Pilates	5	90
6	Strength training	6	60
7	Boot camp	7	90
8	HIIT	8	60
9	Yoga	9	60
NULL	NULL	NULL	NULL

Payments:



PaymentID	MemberID	PaymentDate	PaymentStatus	PlanID
1	1	2024-01-23	Paid	1
2	2	2024-04-02	Unpaid	2
3	3	2024-04-14	Paid	3
4	4	2024-03-01	Paid	2
5	5	2024-04-02	Unpaid	1
6	6	2024-02-25	Paid	3
7	7	2024-04-02	Unpaid	2
8	8	2024-02-12	Paid	1
9	9	2024-01-18	Paid	5
10	10	2024-01-10	Paid	1
11	11	2024-02-04	Paid	2
12	12	2024-04-02	Unpaid	3
13	13	2024-04-02	Unpaid	1
14	14	2024-03-02	Paid	2
15	15	2024-03-05	Paid	4
16	16	2024-03-17	Paid	1
17	17	2024-02-16	Paid	2
18	18	2024-02-04	Paid	3
19	19	2024-01-05	Paid	1
20	20	2024-01-09	Paid	3
21	21	2024-04-02	Unpaid	5
22	22	2024-01-15	Paid	1
23	23	2024-01-26	Paid	2
24	24	2024-04-02	Unpaid	1
25	25	2024-04-02	Unpaid	3



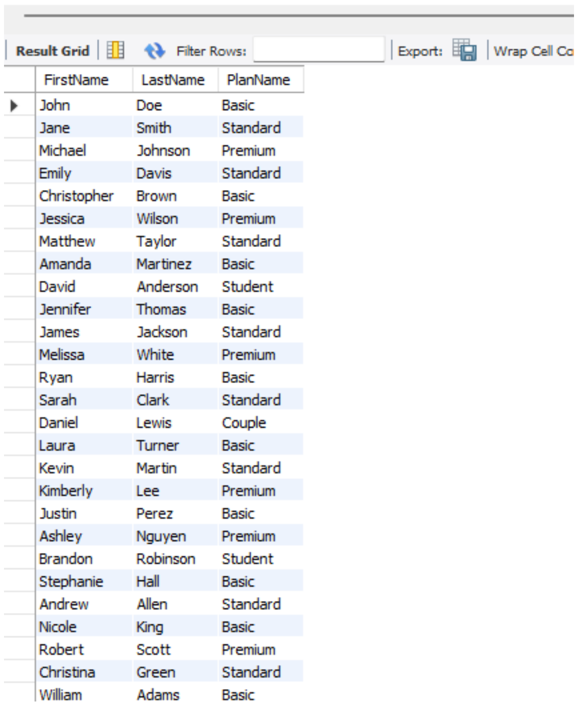
PaymentID	MemberID	PaymentDate	PaymentStatus	PlanID
25	25	2024-04-02	Unpaid	3
26	26	2024-02-20	Paid	2
27	27	2024-04-23	Paid	1
28	28	2024-03-17	Paid	3
29	29	2024-02-06	Paid	1
30	30	2024-01-08	Paid	2
31	31	2024-02-16	Paid	1
32	32	2024-01-18	Paid	3
33	33	2024-02-11	Paid	5
34	34	2024-04-02	Unpaid	1
35	35	2024-03-19	Paid	2
36	36	2024-02-04	Paid	3
37	37	2024-01-27	Paid	1
38	38	2024-04-02	Unpaid	2
39	39	2024-01-25	Paid	1
40	40	2024-01-20	Paid	3
41	41	2024-02-22	Paid	1
42	42	2024-04-02	Unpaid	2
43	43	2024-04-28	Paid	1
44	44	2024-03-01	Paid	3
45	45	2024-02-17	Paid	1
46	46	2024-01-18	Paid	2
47	47	2024-01-10	Paid	3
48	48	2024-01-17	Paid	1
49	49	2024-04-02	Unpaid	5
50	50	2024-02-01	Paid	1
NULL	NULL	NULL	NULL	NULL

SQL Queries:

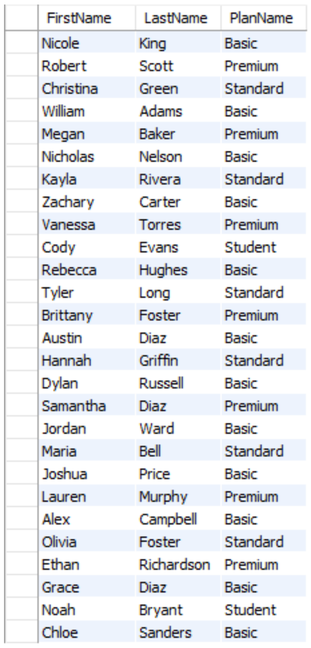
1. Select all members' first names and last names along with their plan names:

Query&Output:

```sql
SELECT FirstName, LastName, PlanName FROM Members;
```



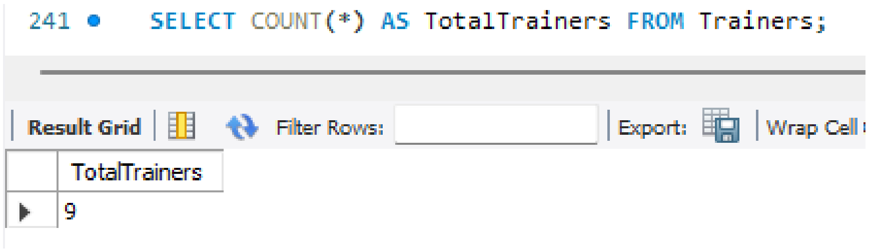
FirstName	LastName	PlanName
John	Doe	Basic
Jane	Smith	Standard
Michael	Johnson	Premium
Emily	Davis	Standard
Christopher	Brown	Basic
Jessica	Wilson	Premium
Matthew	Taylor	Standard
Amanda	Martinez	Basic
David	Anderson	Student
Jennifer	Thomas	Basic
James	Jackson	Standard
Melissa	White	Premium
Ryan	Harris	Basic
Sarah	Clark	Standard
Daniel	Lewis	Couple
Laura	Turner	Basic
Kevin	Martin	Standard
Kimberly	Lee	Premium
Justin	Perez	Basic
Ashley	Nguyen	Premium
Brandon	Robinson	Student
Stephanie	Hall	Basic
Andrew	Allen	Standard
Nicole	King	Basic
Robert	Scott	Premium
Christina	Green	Standard
William	Adams	Basic



FirstName	LastName	PlanName
Nicole	King	Basic
Robert	Scott	Premium
Christina	Green	Standard
William	Adams	Basic
Megan	Baker	Premium
Nicholas	Nelson	Basic
Kayla	Rivera	Standard
Zachary	Carter	Basic
Vanessa	Torres	Premium
Cody	Evans	Student
Rebecca	Hughes	Basic
Tyler	Long	Standard
Brittany	Foster	Premium
Austin	Diaz	Basic
Hannah	Griffin	Standard
Dylan	Russell	Basic
Samantha	Diaz	Premium
Jordan	Ward	Basic
Maria	Bell	Standard
Joshua	Price	Basic
Lauren	Murphy	Premium
Alex	Campbell	Basic
Olivia	Foster	Standard
Ethan	Richardson	Premium
Grace	Diaz	Basic
Noah	Bryant	Student
Chloe	Sanders	Basic

2. Count the total number of trainers:

Query&Output:



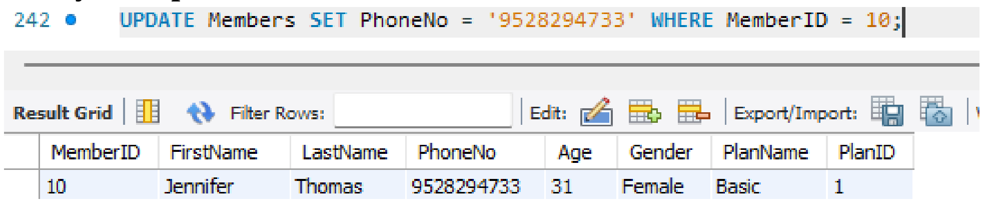
```
SELECT COUNT(*) AS TotalTrainers FROM Trainers;
```

Result Grid | Filter Rows: | Export: | Wrap Cell

TotalTrainers
9

3. Update the phone number of a member with MemberID 10:

Query&Output:



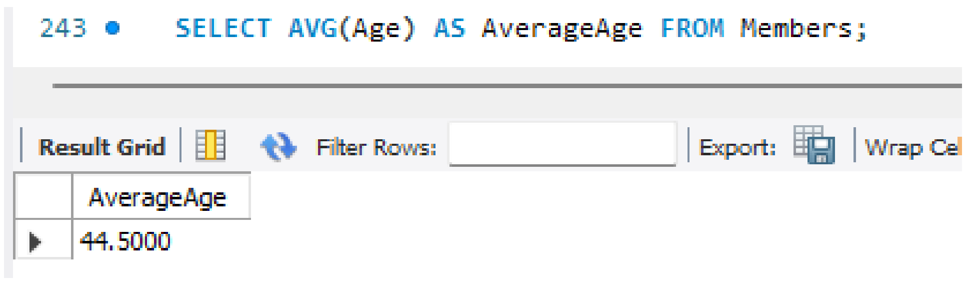
```
UPDATE Members SET PhoneNo = '9528294733' WHERE MemberID = 10;
```

Result Grid | Filter Rows: | Edit: | Export/Import:

MemberID	FirstName	LastName	PhoneNo	Age	Gender	PlanName	PlanID
10	Jennifer	Thomas	9528294733	31	Female	Basic	1

4. Find the average age of members:

Query&Output:

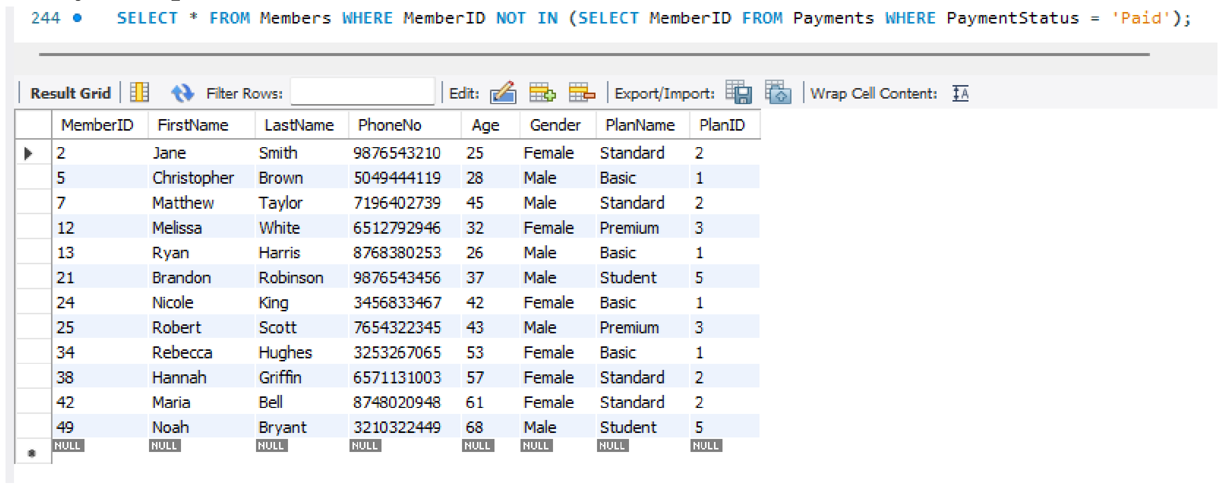


```sql
SELECT AVG(Age) AS AverageAge FROM Members;
```

AverageAge
44.5000

5. List all members who have not paid yet:

Query&Output:

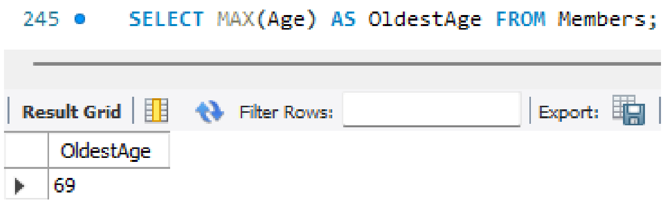


```sql
SELECT * FROM Members WHERE MemberID NOT IN (SELECT MemberID FROM Payments WHERE PaymentStatus = 'Paid');
```

MemberID	FirstName	LastName	PhoneNo	Age	Gender	PlanName	PlanID
2	Jane	Smith	9876543210	25	Female	Standard	2
5	Christopher	Brown	5049444119	28	Male	Basic	1
7	Matthew	Taylor	7196402739	45	Male	Standard	2
12	Melissa	White	6512792946	32	Female	Premium	3
13	Ryan	Harris	8768380253	26	Male	Basic	1
21	Brandon	Robinson	9876543456	37	Male	Student	5
24	Nicole	King	3456833467	42	Female	Basic	1
25	Robert	Scott	7654322345	43	Male	Premium	3
34	Rebecca	Hughes	3253267065	53	Female	Basic	1
38	Hannah	Griffin	6571131003	57	Female	Standard	2
42	Maria	Bell	8748020948	61	Female	Standard	2
49	Noah	Bryant	3210322449	68	Male	Student	5
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

6. Find the oldest member's age:

Query&Output:

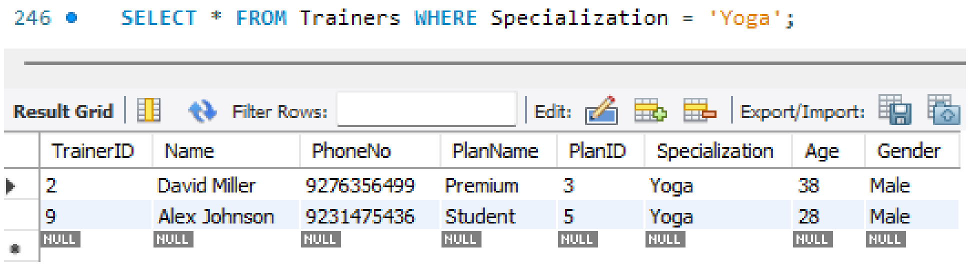


```sql
SELECT MAX(Age) AS OldestAge FROM Members;
```

OldestAge
69

7. List all trainers who specialize in Yoga:

Query&Output:

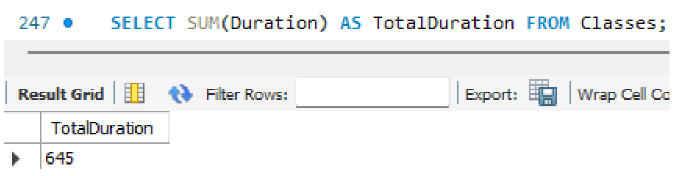


```sql
SELECT * FROM Trainers WHERE Specialization = 'Yoga';
```

TrainerID	Name	PhoneNo	PlanName	PlanID	Specialization	Age	Gender
2	David Miller	9276356499	Premium	3	Yoga	38	Male
9	Alex Johnson	9231475436	Student	5	Yoga	28	Male
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

8. Calculate the total duration of all classes:

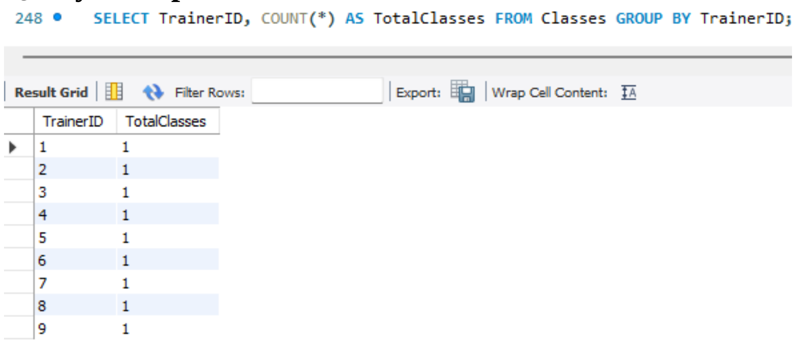
Query&Output:



```sql
SELECT SUM(Duration) AS TotalDuration FROM Classes;
```

```
TotalDuration
645
```

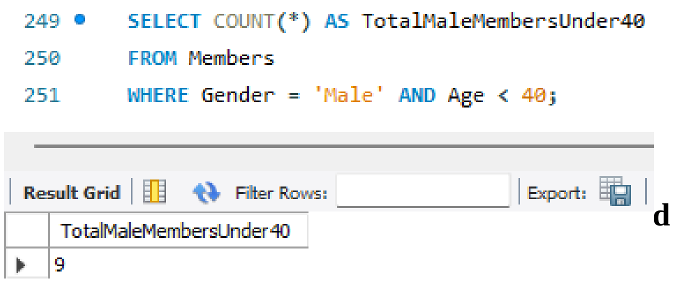
9. Find the total number of classes each trainer conducts:
Query&Output:



```sql
SELECT TrainerID, COUNT(*) AS TotalClasses FROM Classes GROUP BY TrainerID;
```

```
TrainerID  TotalClasses
1          1
2          1
3          1
4          1
5          1
6          1
7          1
8          1
9          1
```

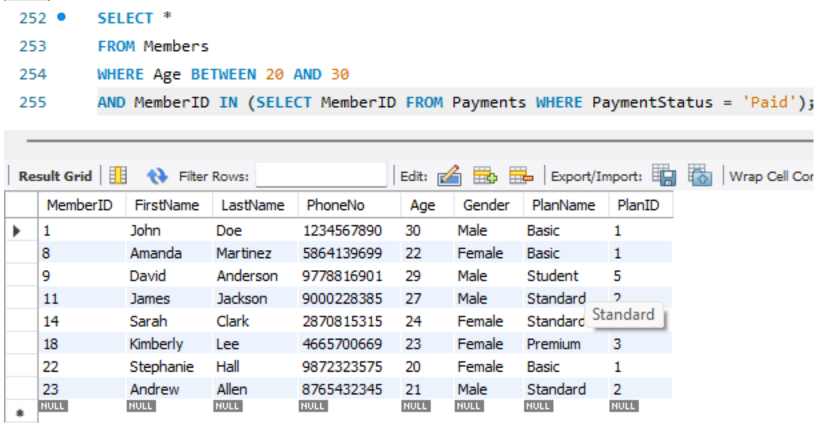
10. Find the total number of male members aged below 40:
Query&Output:



```sql
SELECT COUNT(*) AS TotalMaleMembersUnder40
FROM Members
WHERE Gender = 'Male' AND Age < 40;
```

```
TotalMaleMembersUnder40
9
```

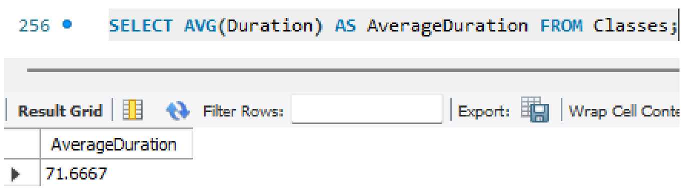
d between 20 and 30:



```sql
SELECT *
FROM Members
WHERE Age BETWEEN 20 AND 30
AND MemberID IN (SELECT MemberID FROM Payments WHERE PaymentStatus = 'Paid');
```

```
MemberID  FirstName  LastName  PhoneNo     Age  Gender  PlanName  PlanID
1         John       Doe       1234567890  30   Male    Basic     1
8         Amanda     Martinez  5864139699  22   Female  Basic     1
9         David      Anderson  9778816901  29   Male    Student   5
11        James      Jackson   9000228385  27   Male    Standard  2
14        Sarah      Clark     2870815315  24   Female  Standard  [illegible]
18        Kimberly   Lee       4665700669  23   Female  Premium   3
22        Stephanie  Hall      9872323575  20   Female  Basic     1
23        Andrew     Allen     8765432345  21   Male    Standard  2
NULL      NULL       NULL      NULL        NULL NULL    NULL      NULL
```

12. Find the average duration of classes:
Query&Output:

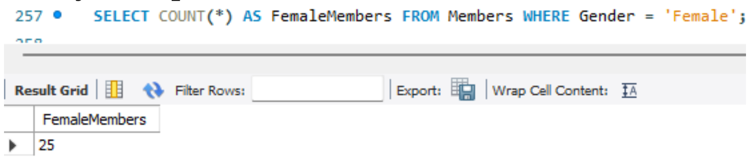


```sql
SELECT AVG(Duration) AS AverageDuration FROM Classes;
```

```
AverageDuration
71.6667
```

13. Count the number of female members:

Query&Output:

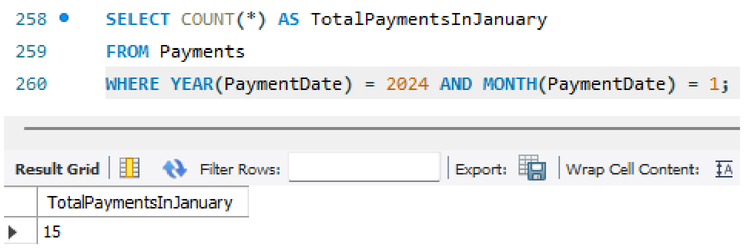


```sql
SELECT COUNT(*) AS FemaleMembers FROM Members WHERE Gender = 'Female';
```

```
FemaleMembers
25
```

14. Find the total number of payments made in January 2024:

Query&Output:

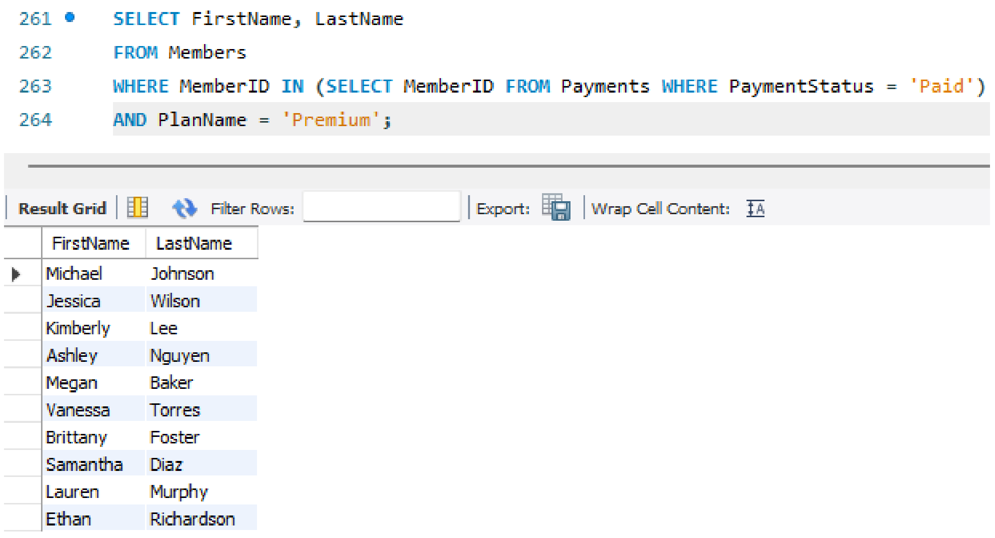


```sql
SELECT COUNT(*) AS TotalPaymentsInJanuary
FROM Payments
WHERE YEAR(PaymentDate) = 2024 AND MONTH(PaymentDate) = 1;
```

```
TotalPaymentsInJanuary
15
```

15. Find the names of members who have paid and are part of the PremiumPlan:

Query&Output:

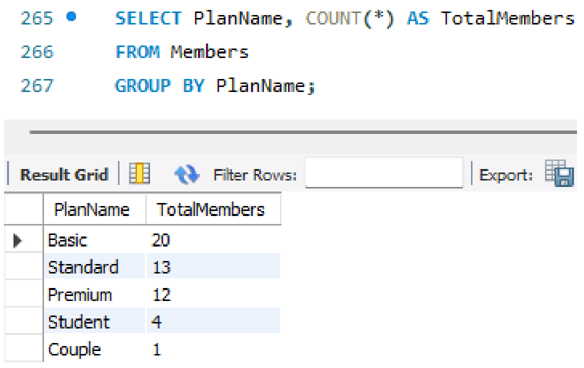


```sql
SELECT FirstName, LastName
FROM Members
WHERE MemberID IN (SELECT MemberID FROM Payments WHERE PaymentStatus = 'Paid')
AND PlanName = 'Premium';
```

```
FirstName   LastName
Michael     Johnson
Jessica     Wilson
Kimberly    Lee
Ashley      Nguyen
Megan       Baker
Vanessa     Torres
Brittany    Foster
Samantha    Diaz
Lauren      Murphy
Ethan       Richardson
```

16. Calculate the total number of members enrolled in each plan:

Query&Output:

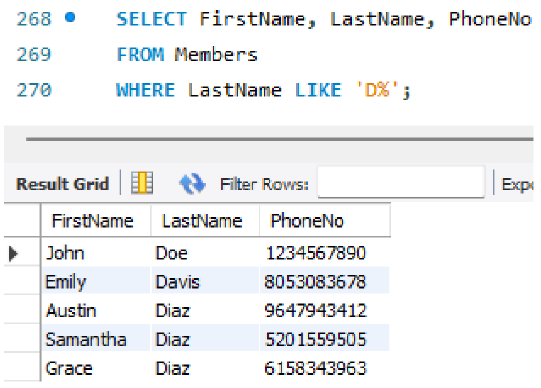


```
SELECT PlanName, COUNT(*) AS TotalMembers
FROM Members
GROUP BY PlanName;
```

PlanName	TotalMembers
Basic	20
Standard	13
Premium	12
Student	4
Couple	1

17. Find the names and phone numbers of embers whose last names start with "D":

Query&Output:

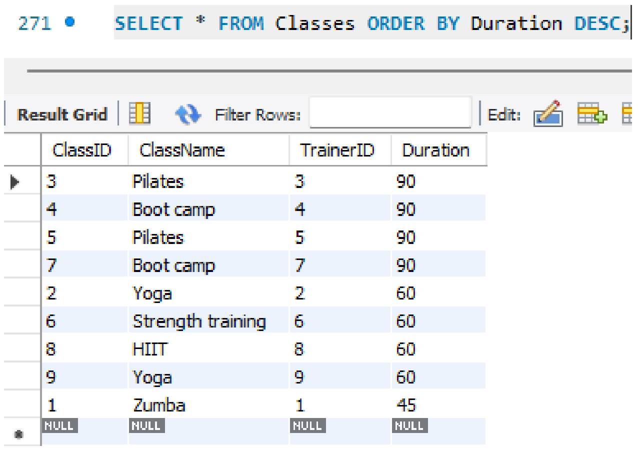


```
SELECT FirstName, LastName, PhoneNo
FROM Members
WHERE LastName LIKE 'D%';
```

FirstName	LastName	PhoneNo
John	Doe	1234567890
Emily	Davis	8053083678
Austin	Diaz	9647943412
Samantha	Diaz	5201559505
Grace	Diaz	6158343963

18. List all classes sorted by duration in descending order:

Query&Output:

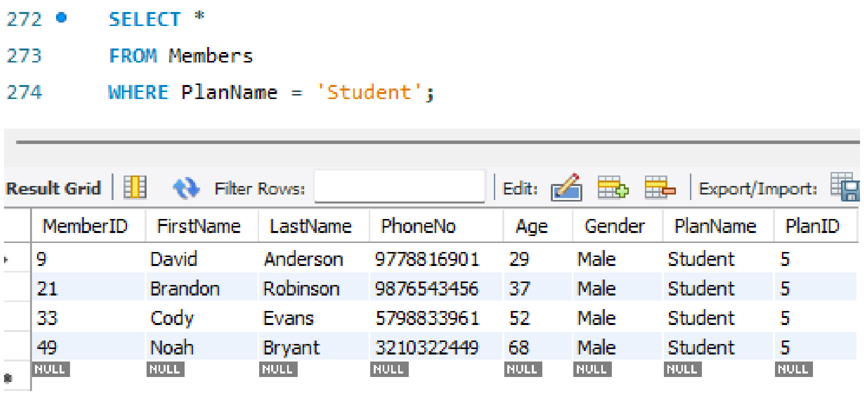


```
SELECT * FROM Classes ORDER BY Duration DESC;
```

ClassID	ClassName	TrainerID	Duration
3	Pilates	3	90
4	Boot camp	4	90
5	Pilates	5	90
7	Boot camp	7	90
2	Yoga	2	60
6	Strength training	6	60
8	HIIT	8	60
9	Yoga	9	60
1	Zumba	1	45
NULL	NULL	NULL	NULL

19. List all members who are a part of the StudentPlan:

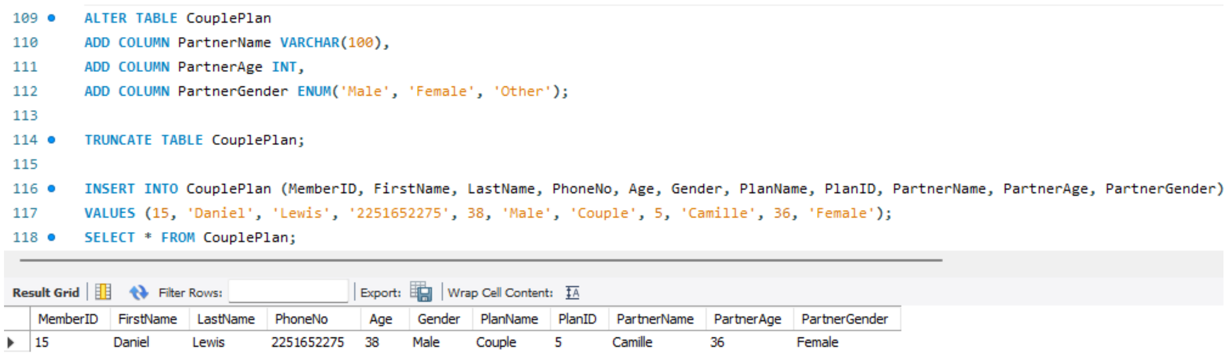
Query&Output:



```sql
SELECT *
FROM Members
WHERE PlanName = 'Student';
```

```
MemberID  FirstName  LastName  PhoneNo     Age   Gender  PlanName  PlanID
9         David      Anderson  9778816901  29    Male    Student   5
21        Brandon    Robinson  9876543456  37    Male    Student   5
33        Cody       Evans     5798833961  52    Male    Student   5
49        Noah       Bryant    3210322449  68    Male    Student   5
NULL      NULL       NULL      NULL        NULL  NULL    NULL      NULL
```

20.Add Partner's Name, Age and Gender to the Table CouplePlan and insert values to it: Query&Output:



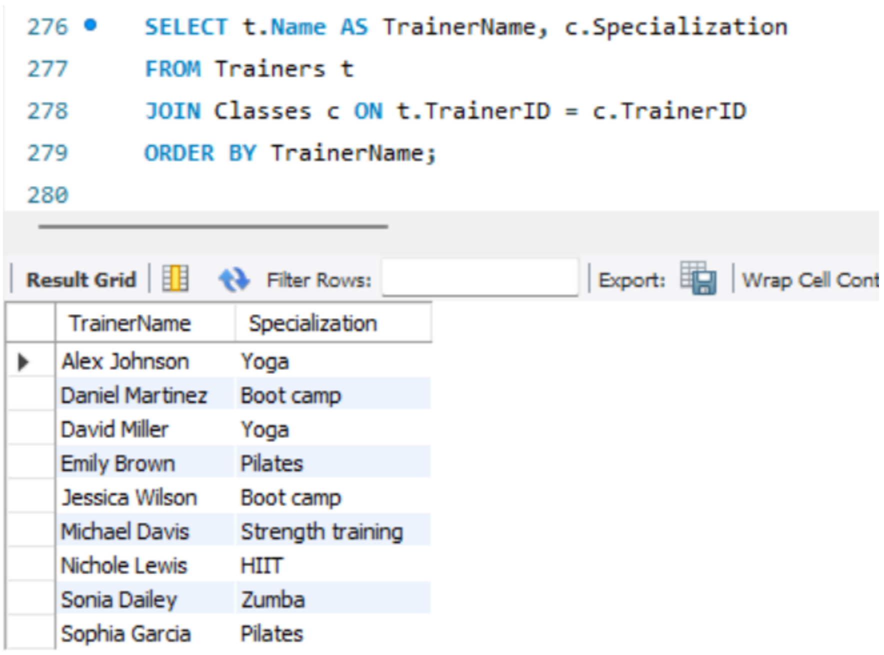
```sql
ALTER TABLE CouplePlan
ADD COLUMN PartnerName VARCHAR(100),
ADD COLUMN PartnerAge INT,
ADD COLUMN PartnerGender ENUM('Male', 'Female', 'Other');

TRUNCATE TABLE CouplePlan;

INSERT INTO CouplePlan (MemberID, FirstName, LastName, PhoneNo, Age, Gender, PlanName, PlanID, PartnerName, PartnerAge, PartnerGender)
VALUES (15, 'Daniel', 'Lewis', '2251652275', 38, 'Male', 'Couple', 5, 'Camille', 36, 'Female');
SELECT * FROM CouplePlan;
```

```
MemberID  FirstName  LastName  PhoneNo     Age  Gender  PlanName  PlanID  PartnerName  PartnerAge  PartnerGender
15        Daniel     Lewis     2251652275  38   Male    Couple    5       Camille      36          Female
```

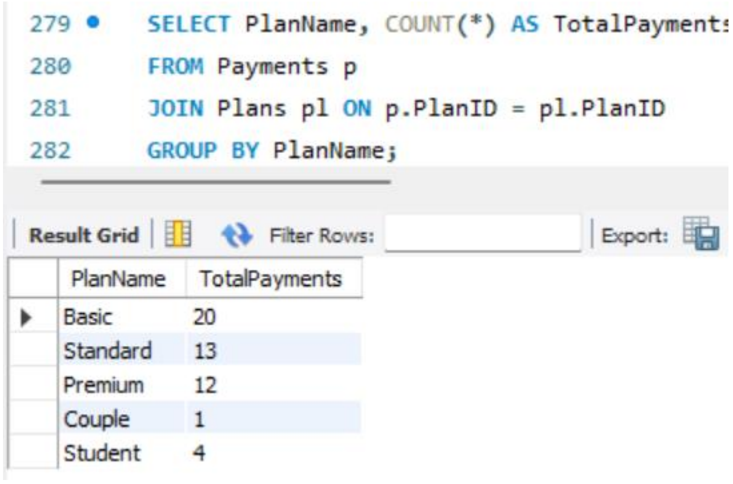
21.Display the names of trainers and the classes they conduct, sorted by trainer's name: Query&Output



```
SELECT t.Name AS TrainerName, c.Specialization
FROM Trainers t
JOIN Classes c ON t.TrainerID = c.TrainerID
ORDER BY TrainerName;
```

TrainerName	Specialization
Alex Johnson	Yoga
Daniel Martinez	Boot camp
David Miller	Yoga
Emily Brown	Pilates
Jessica Wilson	Boot camp
Michael Davis	Strength training
Nichole Lewis	HIIT
Sonia Dailey	Zumba
Sophia Garcia	Pilates

22.Calculate total number of payments made per plan:
Query&Output



```
SELECT PlanName, COUNT(*) AS TotalPayments
FROM Payments p
JOIN Plans pl ON p.PlanID = pl.PlanID
GROUP BY PlanName;
```

PlanName	TotalPayments
Basic	20
Standard	13
Premium	12
Couple	1
Student	4

VI. Project demonstration

- I have used MySQL for this Project.

VII. Self -Learning beyond classroom:

- My self-learning experience beyond the classroom involved a blend of theoretical knowledge application, problem-solving and practical implementation. This enriched my understanding of DBMS subject in the real-world context.

VIII. Learning from the Project

- In summary, this project provided a hands-on learning experience that bridged theoretical knowledge with practical skills, fostering a deeper understanding of DBMS concepts.

IX. Challenges Faced

- Initially I was trying to connect this FlexFit database to it's front-end website that I created, but faced problems with mysql and html connection using nodejs.
- Understanding the structure and relationships within the database required thorough analysis and interpretation of the provided tables and data.
- Identifying and resolving normalization issues, such as transitive dependencies and redundancy, proved challenging due to the complexity of the data model and relationships.

X. Conclusion

- Overall, the FlexFit database normalization project served as a valuable learning experience that bridged theoretical knowledge with practical skills, preparing for future challenges and opportunities in database management and beyond.