

Membership Info and Payment System

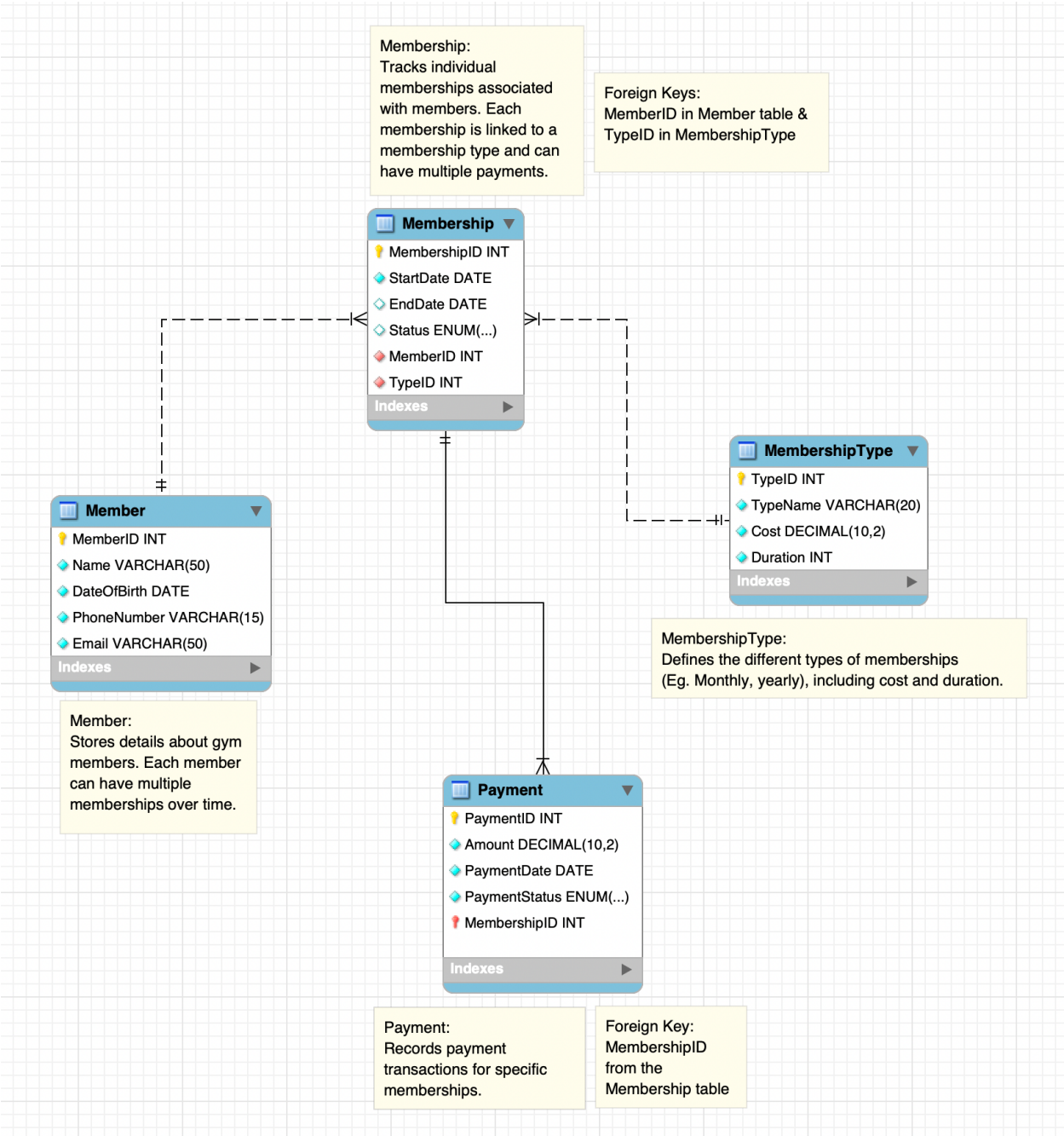
This system is designed to efficiently manage gym member information, membership details, and payment records. It enables administrators to track membership types, statuses, and durations while processing and monitoring payment transactions. The goal is to streamline the gym’s operations: enhance organization and improve daily function for gym management and its members.

Database Column Specifications

Table	Column	Data Type	Constraints
Member	MemberID	INT	PRIMARY KEY, AUTO INCREMENT
	Name	VARCHAR(50)	NOT NULL
	DateOfBirth	DATE	NOT NULL
	PhoneNumber	VARCHAR(45)	NOT NULL, UNIQUE
	Email	VARCHAR(50)	NOT NULL, UNIQUE
Membership	MembershipID	INT	PRIMARY KEY, AUTO INCREMENT
	MemberID	INT	FOREIGN KEY REFERENCES Member(MemberID)
	StartDate	DATE	NOT NULL
	EndDate	DATE	NULLABLE
	Status	ENUM('active', 'expired')	NOT NULL
	MembershipTypeID	INT	FOREIGN KEY REFERENCES MembershipType(TypeID)
Payment	PaymentID	INT	PRIMARY KEY, AUTO INCREMENT
	MembershipID	INT	FOREIGN KEY REFERENCES Membership(MembershipID)
	Amount	DECIMAL(10, 2)	NOT NULL
	PaymentDate	DATE	NOT NULL
	PaymentStatus	ENUM('paid', 'pending', 'failed')	NOT NULL
MembershipType	TypeID	INT	PRIMARY KEY, AUTO INCREMENT

	TypeName	VARCHAR(20)	NOT NULL
	Cost	DECIMAL(10, 2)	NOT NULL
	Duration	INT	NOT NULL

EER Diagram



Examples of how our system can be useful:

1. Subquery: showing which members are enrolled in a membership

```
30 # Subquery to find members who are enrolled in a membership:
31 SELECT Name, PhoneNumber, Email
32 FROM Member
33 WHERE MemberID IN (SELECT MemberID FROM Membership);
34
```

100% 1:1

Result Grid Filter Rows: Search Export:

	Name	PhoneNumber	Email
	Isabel Van Der Veen	123-456-7891	vandervi1@newpaltz.edu
	Ava Allen	567-891-2938	allena17@newpaltz.edu

2. Join: showing all of the 'active' memberships and the details for each

```
35 # Listing all of the active memberships with the individual members and membership types
36 SELECT m.Name, mt.TypeName, ms.StartDate, ms.EndDate
37 FROM Membership ms
38 JOIN Member m ON ms.MemberID = m.MemberID
39 JOIN MembershipType mt ON ms.TypeID = mt.TypeID
40 WHERE ms.Status = 'active';
41
```

100% 28:40

Result Grid Filter Rows: Search Export:

	Name	TypeName	StartDate	EndDate
	Isabel Van Der Veen	Annual	2023-01-01	2023-12-31
	Ava Allen	Monthly	2023-11-01	2023-11-30

3. SUM function: calculating the total revenue from all 'paid' memberships

```
42 # Calculating the total revenue from memberships
43 SELECT SUM(Amount) AS TotalRevenue
44 FROM Payment
45 WHERE PaymentStatus = 'paid';
```

100% 30:45

Result Grid Filter Rows: Search Export:

	TotalRevenue
	550.00

4. GroupBy: Counting the number of memberships and separating them by status

```
47 # Finding the COUNT for memberships for each 'status (active, pending, expired)
48 • SELECT Status, COUNT(MembershipID) AS MembershipCount
49 FROM Membership
50 GROUP BY Status;
```

100% 17:50

Result Grid Filter Rows: Search Export:

Status	MembershipCount
active	2

5. Filter: Finding all the emails that end in edu.

```
52 # Filtering by Emails ending in .edu
53 • SELECT Name, Email
54 FROM Member
55 WHERE Email LIKE '%.edu';
```

100% 26:55

Result Grid Filter Rows: Search

Name	Email
Isabel Van Der Veen	vandervi1@newpaltz.edu
Ava Allen	allena17@newpaltz.edu

6. Multiple Condition Combination: Finding all active memberships that started on or after 1/1/2023 and that are longer than 6 months.

```
57 # Listing all active members that started on or after '2023-01-01' and have a duration of longer than 6 months
58 • SELECT m.Name, ms.StartDate, mt.TypeName, mt.Duration
59 FROM Membership ms
60 JOIN Member m ON ms.MemberID = m.MemberID
61 JOIN MembershipType mt ON ms.TypeID = mt.TypeID
62 WHERE ms.Status = 'active'
63 AND ms.StartDate >= '2023-01-01'
64 AND mt.Duration > 6;
```

100% 42:38

Result Grid Filter Rows: Search Export:

Name	StartDate	TypeName	Duration
Isabel Van Der Veen	2023-01-01	Annual	12