Part 2 - Schema Screenshots

Member

```
sqlite> .schema Member
CREATE TABLE Member (
    memberId INTEGER PRIMARY KEY AUTOINCREMENT,
    name VARCHAR(50) NOT NULL,
    email VARCHAR(50) UNIQUE NOT NULL,
    phone VARCHAR(15),
    address VARCHAR(100),
    age INTEGER CHECK (age >= 15),
    membershipStartDate DATE NOT NULL,
    membershipEndDate DATE NOT NULL CHECK(membershipEndDate > membershipStartDate)
);
sqlite> pragma table_info('Member')
   ...> ;
0|memberId|INTEGER|0||1
1|name|VARCHAR(50)|1||0
2|email|VARCHAR(50)|1||0
3|phone|VARCHAR(15)|0||0
4|address|VARCHAR(100)|0||0
5|age|INTEGER|0||0
6|membershipStartDate|DATE|1||0
7|membershipEndDate|DATE|1||0
sqlite> SElECT * FROM Instructor
   ...> ;
1|Jacob Smith|Yoga|1112223333|js@mail.com
2|Mary Johnson|Zumba|2223334444|mj@mail.com
3|Tom Jacobson|HIIT|3334445555|tj@mail.com
4|Joseph Gonzales|Weights|4445556666|jg@mail.com
5|Bob Brown|Weights|5556667777|bb@mail.com
sqlite>
```

Class

```
sqlite> .schema Class
CREATE TABLE Class (
    classId INTEGER PRIMARY KEY AUTOINCREMENT,
    className VARCHAR(50),
    classType VARCHAR(20) CHECK(classType IN ('Yoga', 'Zumba', 'HIIT', 'Weights')),
    duration INTEGER CHECK (duration > 0),
    classCapacity INTEGER CHECK(classCapacity > 0),
    instructorId INTEGER,
    gymId INTEGER,
    FOREIGN KEY (instructorId) REFERENCES Instructor(instructorId),
    FOREIGN KEY (gymId) REFERENCES GymFacility(gymId)
);
sqlite> pragma table_info('Class');
0|classId|INTEGER|0||1
1|className|VARCHAR(50)|0||0
2|classType|VARCHAR(20)|0||0
3|duration|INTEGER|0||0
4|classCapacity|INTEGER|0||0
5|instructorId|INTEGER|0||0
6|gymId|INTEGER|0||0
sqlite> SElECT * FROM Class;
1|Morning Yoga|Yoga|60|20|1|1
2|Zumba Dance|Zumba|45|25|2|2
3|HIIT Blast|HIIT|30|15|3|3
4|Strength Training|Weights|75|10|4|4
5|Afternoon Yoga|Yoga|60|20|5|5
sqlite>
```

Instructor

```
sqlite> .schema Instructor
CREATE TABLE Instructor (
    instructorId INTEGER PRIMARY KEY AUTOINCREMENT,
    name VARCHAR(50) NOT NULL,
    specialty VARCHAR(50),
    phone VARCHAR(15),
    email VARCHAR(100) UNIQUE NOT NULL
);
sqlite> pragma table_info('Instructor');
0|instructorId|INTEGER|0||1
1|name|VARCHAR(50)|1||0
2|specialty|VARCHAR(50)|0||0
3|phone|VARCHAR(15)|0||0
4|email|VARCHAR(100)|1||0
sqlite> SElECT * FROM Instructor;
1|Jacob Smith|Yoga|1112223333|js@mail.com
2|Mary Johnson|Zumba|2223334444|mj@mail.com
3|Tom Jacobson|HIIT|3334445555|tj@mail.com
4|Joseph Gonzales|Weights|4445556666|jg@mail.com
5|Bob Brown|Weights|5556667777|bb@mail.com
sqlite>
```

GymFacility

```
sqlite> .schema GymFacility
CREATE TABLE GymFacility (
    gymId INTEGER PRIMARY KEY AUTOINCREMENT,
    location varchar(100),
    phone varchar(30),
    manager varchar(50)
);
sqlite> pragma table_info('GymFacility');
0|gymId|INTEGER|0||1
1|location|varchar(100)|0||0
2|phone|varchar(30)|0||0
3|manager|varchar(50)|0||0
sqlite> SELECT * FROM GymFacility;
1|Portales NM|4441112222|Thomas Pritchett
2|Clovis NM|5552223333|Kayla Stevenson
3|Roswell NM|6663332222|Caroline Wall
4|Ruidoso NM|7774445555|Corny Friesen
5|Melrose NM|8885556666|Peter Froese
sqlite>
```

Equipment

```
sqlite> .schema Equipment
CREATE TABLE Equipment (
    equipmentId INTEGER PRIMARY KEY AUTOINCREMENT,
    name VARCHAR(50) NOT NULL,
    type VARCHAR(30) CHECK(type = 'Cardio' OR type = 'Strength' OR type = 'Flexibility' OR type = 'Recovery'),
    quantity INTEGER CHECK (quantity >= 0),
    gymId INTEGER,
    FOREIGN KEY (gymId) REFERENCES GymFacility(gymId)
);
sqlite> pragma table_info('Equipment');
0|equipmentId|INTEGER|0||1
1|name|VARCHAR(50)|1||0
2|type|VARCHAR(30)|0||0
3|quantity|INTEGER|0||0
4|gymId|INTEGER|0||0
sqlite> SELECT * FROM Equipment;
1|Treadmills|Cardio|5|1
2|Dumbbells|Strength|20|2
3|Resistance Bands|Flexibility|15|3
4|Foam Rollers|Recovery|10|4
5|Elliptical|Cardio|3|5
sqlite>
```

MembershipPlan

```
sqlite> .schema MembershipPlan
CREATE TABLE MembershipPlan (
    planId INTEGER PRIMARY KEY AUTOINCREMENT,
    planType VARCHAR(20) CHECK(planType IN('Monthly', 'Annual')),
    cost NUMERIC CHECK (cost >= 0)
);
sqlite> pragma table_info('MembershipPlan');
0|planId|INTEGER|0||1
1|planType|VARCHAR(20)|0||0
2|cost|NUMERIC|0||0
sqlite> SELECT * FROM MembershipPlan;
1|Monthly|50
2|Annual|500
3|Monthly|25
4|Annual|250
5|Annual|750
sqlite>
```

Payment

```
sqlite> .schema Payment
CREATE TABLE Payment (
    paymentId INTEGER PRIMARY KEY AUTOINCREMENT,
    memberId INTEGER,
    planId INTEGER,
    amountPaid REAL NOT NULL CHECK(amountPaid >= 0),
    paymentDate DATE NOT NULL,
    FOREIGN KEY (memberId) REFERENCES Member(memberId),
    FOREIGN KEY (planId) REFERENCES MembershipPlan(planId)
);
sqlite> pragma table_info('Payment');
0|paymentId|INTEGER|0||1
1|memberId|INTEGER|0||0
2|planId|INTEGER|0||0
3|amountPaid|REAL|1||0
4|paymentDate|DATE|1||0
sqlite> SELECT * FROM Payment;
1|1|1|50.0|2025-03-01
2|2|2|500.0|2025-03-01
3|3|3|25.0|2025-03-01
4|4|4|250.0|2025-03-01
5|5|5|750.0|2025-03-01
sqlite>
```

Attends

```
sqlite> .schema Attends
CREATE TABLE Attends (
    memberId INTEGER,
    classId INTEGER,
    attendanceDate DATE NOT NULL,
    PRIMARY KEY (memberId, classId, attendanceDate),
    FOREIGN KEY (memberId) REFERENCES Member(memberId),
    FOREIGN KEY (classId) REFERENCES Class(classId)
);
sqlite> pragma table_info('Attends');
0|memberId|INTEGER|0||1
1|classId|INTEGER|0||2
2|attendanceDate|DATE|1||3
sqlite> SELECT * FROM Attends;
1|2|2025-03-02
2|1|2025-03-02
3|4|2025-03-02
4|3|2025-03-02
5|2|2025-03-02
sqlite>
```