Instructor

```sql
CREATE TABLE Instructor (instructorId INTEGER PRIMARY KEY AUTOINCREMENT, name varchar (50), specialty varchar (50), phone varchar (15), email varchar (100) NOT NULL);
```

Class

```sql
CREATE TABLE class (classId INTEGER PRIMARY KEY AUTOINCREMENT, className varchar (50), classType varchar (20) CHECK (ClassType == "Yoga" OR ClassType == "Zumba" OR ClassType == "HIIT" OR ClassType == "Weights"), duration INTEGER NOT NULL, classCapacity INTEGER NOT NULL, instrucotorId INTEGER REFERENCES Instructor (instructorId), gymID INTEGER REFERENCES gym_facility (gym_id));
```

Gym Facility

```sql
CREATE TABLE gym_facility (
   gym_id INTEGER PRIMARY KEY AUTOINCREMENT,
   location VARCHAR(100),
   phone VARCHAR(50),
   manager VARCHAR(50)
);
```

Membership Plan

```sql
CREATE TABLE membershipPlan(
   planId INTEGER PRIMARY KEY AUTOINCREMENT,
   planType VARCHAR(20) CHECK(planType =='Monthly' OR planType =='Annual'),
   cost NUMERIC NOT NULL
);
```

Attends

```sql
CREATE TABLE attends (
   memberId INTEGER NOT NULL,
   classId INTEGER NOT NULL,
   attendanceDate DATE NOT NULL,
   PRIMARY KEY (memberId, classId, attendanceDate),
   FOREIGN KEY(memberId) REFERENCES member(memberId),
   FOREIGN KEY(classId) REFERENCES class(classId)
);
```

Equipment

```sql
CREATE TABLE equipment (
  equipment_id 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(30),
  gym_id INTEGER,
  FOREIGN KEY(gym_id) REFERENCES gym_facility(gym_id)
);
```

Member

```sql
CREATE TABLE member(
  memberId INTEGER PRIMARY KEY AUTOINCREMENT,
  name VARCHAR(50),
  email VARCHAR(50) NOT NULL,
  phone VARCHAR(15),
  address VARCHAR(100),
  age INTEGER CHECK (age > 14),
  membershipStartDate DATE NOT NULL,
  membershipEndDate DATE NOT NULL CHECK (membershipEndDate > membershipStartDate)
);
```

Payment

```sql
CREATE TABLE payment(
  paymentId INTEGER PRIMARY KEY AUTOINCREMENT,
  memberId INTEGER,
  planId INTEGER,
  amountPaid REAL NOT NULL,
  paymentDate DATE NOT NULL,
  FOREIGN KEY(memberId) REFERENCES member(memberId),
  FOREIGN KEY(planId) REFERENCES membershipPlan(planId)
);
```