Instructor

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

Class

```
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,
    instructorId INTEGER,
    gymID         INTEGER,
    FOREIGN KEY(instructorId) REFERENCES Instructor (instructorId),
    FOREIGN KEY(gymID) REFERENCES gymFacility (gymId)
);
```

Gym Facility

```
CREATE TABLE gymFacility (
  gymId INTEGER PRIMARY KEY AUTOINCREMENT,
  location VARCHAR(100),
  phone VARCHAR(50),
  manager VARCHAR(50)
);
```

Membership Plan

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

Attends

```
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

```
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(30),
  gymId INTEGER,
  FOREIGN KEY(gymId) REFERENCES gymFacility(gymId)
);
```

Member

```
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

```
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)
);
```