(1) CREATE THE MEMBERS TABLE

```
CREATE TABLE Members (
  MemberID INT PRIMARY KEY AUTO_INCREMENT,
  FirstName VARCHAR(100) NOT NULL,
   LastName VARCHAR(100) NOT NULL,
   ContactDetails VARCHAR(255),
   Email VARCHAR(100),
  MembershipTypeID INT,
  JoinDate DATE,
  FOREIGN KEY (MembershipTypeID) REFERENCES
  MembershipTypes(MembershipTypeID)
);
```

(2) CREATE THE MEMBERSHIPTYPES TABLE

```
CREATE TABLE MembershipTypes (
  MembershipTypeID INT PRIMARY KEY AUTO_INCREMENT,
  MembershipTypeName VARCHAR(100) NOT NULL,
  Description TEXT,
   Price DECIMAL(10, 2)
);
```

(3) CREATE THE TRAINERS TABLE

```
CREATE TABLE Trainers (
   TrainerID INT PRIMARY KEY AUTO_INCREMENT,
   FirstName VARCHAR(100) NOT NULL,
  LastName VARCHAR(100) NOT NULL,
  Qualifications TEXT,
  Specialties TEXT,
  Availability VARCHAR(255)
```

```
);
```

(4) CREATE THE CLASSES TABLE

```
CREATE TABLE Classes (
   ClassID INT PRIMARY KEY AUTO_INCREMENT,
   ClassName VARCHAR(100) NOT NULL,
   Description TEXT,
   Date DATE,
  Time TIME,
  TrainerID INT,
  FOREIGN KEY (TrainerID) REFERENCES Trainers(TrainerID)
);
```

(5) CREATE THE CLASSREGISTRATIONS TABLE

```
CREATE TABLE ClassRegistrations (
  RegistrationID INT PRIMARY KEY AUTO_INCREMENT,
  MemberID INT,
  ClassID INT,
  Attendance BOOLEAN,
  Feedback TEXT,
  FOREIGN KEY (MemberID) REFERENCES Members(MemberID),
  FOREIGN KEY (ClassID) REFERENCES Classes(ClassID)
);
```

(6) CREATE THE PAYMENTS TABLE

```
CREATE TABLE Payments (
  PaymentID INT PRIMARY KEY AUTO_INCREMENT,
  MemberID INT,
  PaymentDate DATE,
```

```
  Amount DECIMAL(10, 2),
  PaymentType VARCHAR(50),
  FOREIGN KEY (MemberID) REFERENCES Members(MemberID)
);
```

(7) CREATE THE EQUIPMENT TABLE

```
CREATE TABLE Equipment (
  EquipmentID INT PRIMARY KEY AUTO_INCREMENT,
  EquipmentName VARCHAR(100) NOT NULL,
  Description TEXT,
  PurchaseDate DATE,
  Condition VARCHAR(50)
);
```

(8) CREATE THE EQUIPMENTUSAGE TABLE

```
CREATE TABLE EquipmentUsage (
   UsageID INT PRIMARY KEY AUTO_INCREMENT,
   EquipmentID INT,
  MemberID INT,
  UsageDate DATE,
  Duration INT, -- Duration in minutes
  FOREIGN KEY (EquipmentID) REFERENCES Equipment(EquipmentID),
  FOREIGN KEY (MemberID) REFERENCES Members(MemberID)
);
```

(9) CREATE THE PROMOTIONS TABLE

```
CREATE TABLE Promotions (
  PromotionID INT PRIMARY KEY AUTO_INCREMENT,
  MemberID INT,
```

```
  PromotionDate DATE,
  PromotionDetails TEXT,
  FOREIGN KEY (MemberID) REFERENCES Members(MemberID)
);
```

(10)CREATE THE REPORTS TABLE

```
CREATE TABLE Reports (
  ReportID INT PRIMARY KEY AUTO_INCREMENT,
  ReportType VARCHAR(100),
  GenerationDate DATE,
  Details TEXT
);
```

SOLUTION OF THE GIVEN QUESTION

1. **Find the class with the highest attendance**

```
SELECT
  C.ClassID,
  C.ClassName,
  COUNT(CR.Attendance) AS TotalAttendance
FROM
  Classes C
JOIN
  ClassRegistrations CR
ON
  C.ClassID = CR.ClassID
WHERE
  CR.Attendance = 1
GROUP BY
  C.ClassID, C.ClassName
ORDER BY
  TotalAttendance DESC
LIMIT 1;
```

2. CALCULATE THE AVERAGE NUMBER OF CLASSES ATTENDED PER MEMBER

```
SELECT
  AVG(ClassAttendance.AttendedClasses) AS AverageClassesAttended
FROM
  (
    SELECT
      cr.MemberID,
      COUNT(cr.Attendance) AS AttendedClasses
    FROM
      ClassRegistrations cr
    WHERE
      cr.Attendance = 1 -- Assuming Attendance is stored as 1 for attended and 0
for not attended
    GROUP BY
      cr.MemberID
  ) AS ClassAttendance;
```

3. RETRIEVE THE MINIMUM NUMBER OF PARTICIPANTS IN ANY CLASS

```
SELECT MIN(ParticipantsCount) AS MinParticipants
FROM (
  SELECT ClassID, COUNT(MemberID) AS ParticipantsCount
  FROM ClassRegistrations
  GROUP BY ClassID
) AS Subquery;
```

4. FIND THE TRAINER WITH THE HIGHEST NUMBER OF CLASSES TAUGHT

```
    SELECT

        t.TrainerID,

        t.FirstName,

        t.LastName,

        COUNT(c.ClassID) AS TotalClassesTaught

    FROM

        Trainers t

    JOIN

        Classes c

    ON
```

```
    t.TrainerID = c.TrainerID
GROUP BY
    t.TrainerID,
    t.FirstName,
    t.LastName
ORDER BY
    TotalClassesTaught DESC
LIMIT 1;
```

5. LIST THE TOP 5 MOST POPULAR MEMBERSHIP TYPES

```
SELECT
  MT.MembershipTypeName,
  COUNT(M.MemberID) AS NumberOfMembers
FROM
  MembershipTypes MT
JOIN
  Members M ON MT.MembershipTypeID = M.MembershipTypeID
GROUP BY
  MT.MembershipTypeName
ORDER BY
  NumberOfMembers DESC
LIMIT 5;
```