1. **Клиент (Client)**
   * id (int, primary key)
   * first\_name (string)
   * last\_name (string)
   * email (string, unique)
   * phone (string, unique)
   * date\_of\_birth (date)
   * created\_at (timestamp)
   * updated\_at (timestamp)
2. **Тренер (Trainer)**
   * id (int, primary key)
   * first\_name (string)
   * last\_name (string)
   * email (string, unique)
   * phone (string, unique)
   * specialization (string)
   * created\_at (timestamp)
   * updated\_at (timestamp)
3. **Абонемент (Membership)**
   * id (int, primary key)
   * name (string)
   * description (text)
   * price (decimal)
   * duration\_months (int)
   * created\_at (timestamp)
   * updated\_at (timestamp)
4. **Посещение (Attendance)**
   * id (int, primary key)
   * client\_id (int, foreign key to Clients)
   * date (date)
   * created\_at (timestamp)
   * updated\_at (timestamp)
5. **Зал (Class)**
   * id (int, primary key)
   * name (string)
   * description (text)
   * trainer\_id (int, foreign key to Trainers)
   * schedule (string)
   * created\_at (timestamp)
   * updated\_at (timestamp)
6. **Запись в зал (ClassRegistration)**
   * id (int, primary key)
   * client\_id (int, foreign key to Clients)
   * class\_id (int, foreign key to Classes)
   * registration\_date (date)
   * created\_at (timestamp)
   * updated\_at (timestamp)

#### **Реляции/отношения между сущностями**

1. **Клиент - Абонемент (Client - Membership)**
   * Один клиент может иметь много абонементов (One-to-Many)
   * В таблице Membership будет client\_id (foreign key to Clients)
2. **Клиент - Посещение (Client - Attendance)**
   * Один клиент может иметь много посещений (One-to-Many)
   * В таблице Attendance будет client\_id (foreign key to Clients)
3. **Тренер - Класс (Trainer - Class)**
   * Один тренер может вести много классов (One-to-Many)
   * В таблице Class будет trainer\_id (foreign key to Trainers)
4. **Клиент - Зал (Client - Class) через ClassRegistration**
   * Много клиентов могут записаться во много заллов (Many-to-Many)
   * В таблице ClassRegistration будет client\_id (foreign key to Clients) и class\_id (foreign key to Classes)

#### **Обоснование выбора сущностей и связей**

* **Клиент**: Центральная сущность системы, представляющая человека, который посещает спортивный клуб.
* **Тренер**: Люди, которые проводят занятия и тренировки.
* **Абонемент**: Сущность для управления различными планами абонементов, которые клиенты могут приобретать.
* **Посещение**: Отслеживает посещаемость клиентов.
* **Зал:** Различные виды тренировок и занятий, проводимых в клубе.
* **Запись на класс**: Для управления записью клиентов на конкретные классы.