**Tutorial 4 Assessment**

**Question 1**

Our work for EasyLife continues.

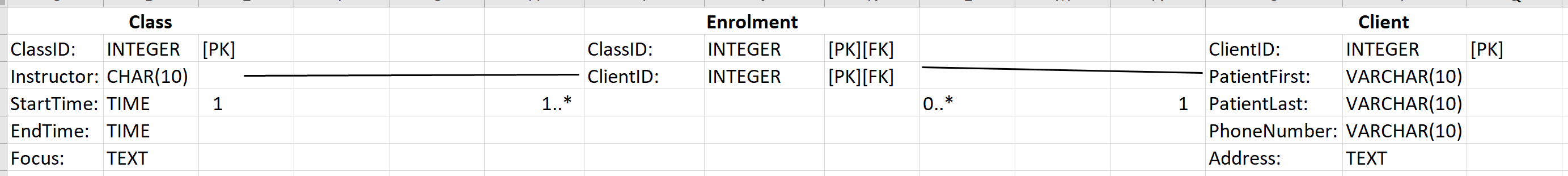
*They also run classes for their clients, where groups of people with a similar injury do exercises together under the instruction of a physiotherapist. EasyLife need the data structures to store these classes and their participants. The participants are clients. Clients can have individual appointments - which we have already modelled - and classes or group sessions.*

*The classes (or group sessions) have an instructor (physiotherapists Bruno or Deepika), a start time, an end time and a focus (spine, lower leg, elbow, etc. that determines the exercises).*

You have to create a model that enables **several clients to take the same class** and the **same client to participate in several classes.**

Model the entities you think are necessary for the datastore to be able to store this scenario. Determine an appropriate primary key for all entities involved. You may use a **natural** or **surrogate** **key.** Comment about the assumptions for your primary key - e.g. can two spine classes run at the same time?

Develop a UML diagram with attributes that show which of the attributes are part of the primary key. Show the relationships between the entities you have identified. Mark the attributes which are part of the foreign key.



Commenting on the assumptions of my primary key, using surrogate keys for the Class and Client entities guarantees a unique key. The relationships between the entity is:

* A "Class" can have multiple "Enrolments," and each "Enrolment" belongs to one "Class" (one-to-many relationship).
* A "Client" can have zero or multiple "Enrolments," and each "Enrolment" belongs to one "Client" (zero-to-many relationship).

**Question 2**

For every UML entity you have created for Question 1, make a data table to test whether your model works well. Make your own test data and check for problems of duplication (except where allowed, i.e. the foreign keys).

A screenshot of a computer

Description automatically generated

There are no duplications in “Class” since there’s a unique identifier “ClassID.” There are no duplications in “Client” since there’s a unique identifier “ClientID.” There are no duplications in “Enrolment” since there’s a unique identifier due to the surrogate key, and there are no duplications in "Appointment" since there’s a unique identifier “AppointmentID.”

There can be multiple appointments with the same client participating in different classes and vice versa.