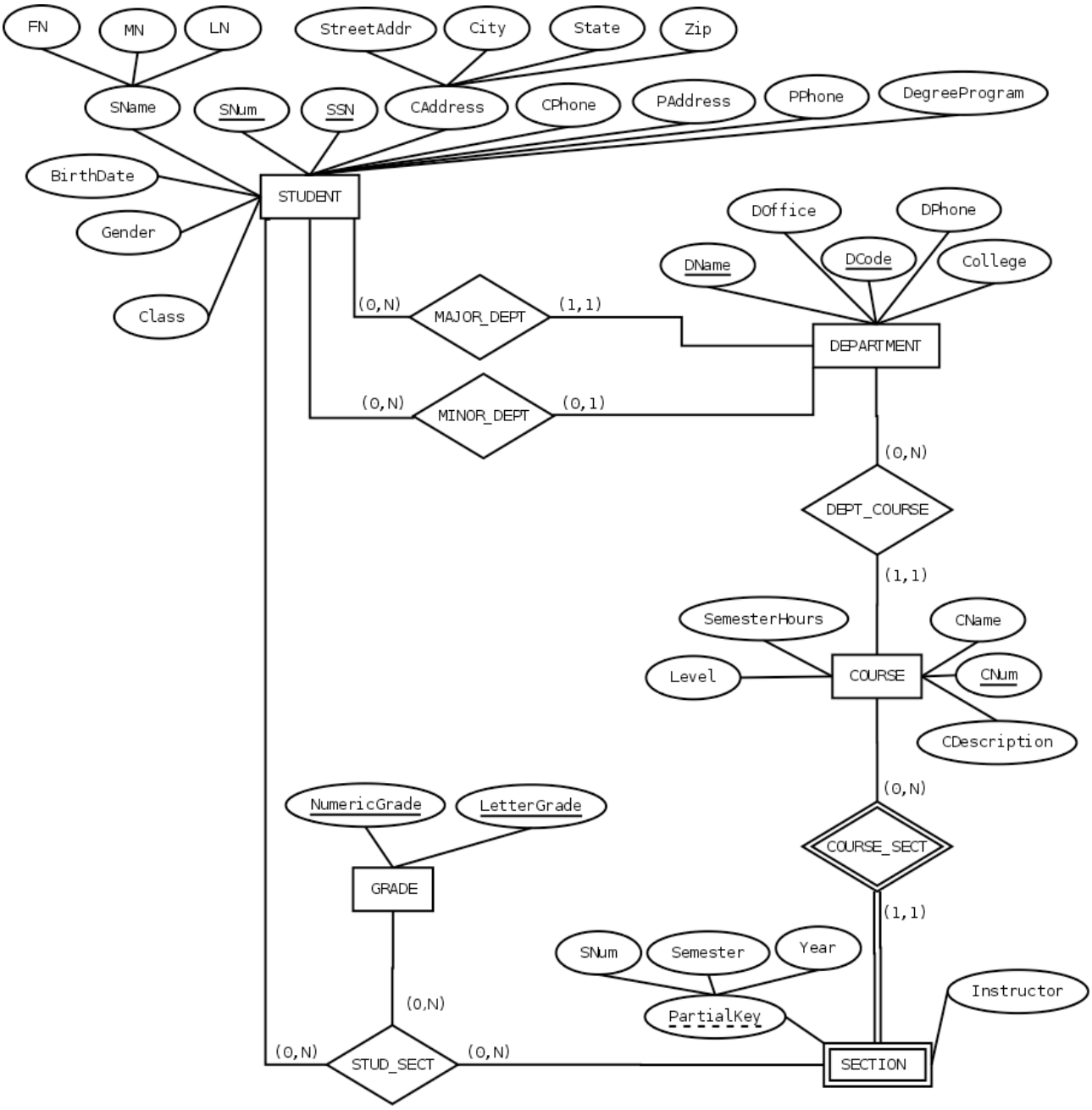
MODEL ANSWERS :: Examples Clinic 2

Entity-Relationship Modelling

T1. *Here is a sample ER diagram derived from the given textual requirements. Note that it uses the alternative notation (i.e., pairs, rather than single/double lines) for expressing the structural constraints on the relationships:*



Do not worry if you used the alternative notation for cardinality ratio and participation constraints but use this opportunity to learn how to use either. In particular notice that, in this alternative notation, there are variants that swap the side in which the pairs are placed.

Also, note that your choice of names for entity types, attributes and relationship types may be different and is immaterial provided that it is sensible and consistently applied.

Note, furthermore, that the answer above is rather lax regarding naming relationship types with verbs: we hope you have done better than that!

Finally, the overall layout does not matter very much but, again, try to follow the left-to-right, top-to-bottom legibility guideline.

Some remarks about the conceptual model above now follow for you to reflect on.

- *Notice how SName and CAddress are modelled as composite attributes. Many of you may have felt the inclination to consider them as entity types, but one reason why this does not seem a good idea is because for a concept to be modelled as an entity type, the entities in its entity set (i.e., in its possible stored extents) must be of interest as entities. In a university database, this does not seem the case for an address, but, e.g., for the Royal Mail, it may well be. In fact, when you think about it, the postcode used in the UK is precisely an artificially created key attribute resulting from the interest by Royal Mail in managing information about addresses irrespective of their occupants.*
- *Note also that entity types may have more than one key attribute (e.g., GRADE does, as does STUDENT). Be careful not to think about this in relational terms (i.e., referring to the level of logical schemas). In ER terms (i.e., at level of conceptual schemas), all that is needed from the designer is (a) to satisfy herself that there is a key attribute, and (b) if there is more than one key attribute, to model this fact.*
- *Note that STUD_SECT is a natural ternary relationship, i.e., a STUDENT achieves a GRADE for a particular SECTION. A good, domain-independent, example of a natural ternary relationship is BETWEEN, because the concept is that, say, X is BETWEEN Y and Z, i.e., there are three arguments to the relationship.*
- *Note that SECTION is a weak entity type because it only has a partial (or weak) key. A section entity depends for its existence on a course, which is to say, if we were to refer to, say, Section 1 of Semester 2 of 2012, we will be asked: 'Of which COURSE?'. In other words, there has to be an identifying relationship with COURSE.*
- *Note, finally, that we must have two relationship types between STUDENT and DEPARTMENT: one for the case of a major and one for the case of a minor. The crucial reason is because the constraints are different. Whenever we have such distinct constraints, we must split in order to accurately represent the domain.*
- *To reinforce your understanding of the notation, in the case above, let's take the MAJOR_DEPT relationship type between the STUDENT and DEPARTMENT entity types. The relevant part of the above diagram, given the pairs (0,N) on the STUDENT side and (1,1) on the DEPARTMENT side should be read as follows: a STUDENT must have one, and only one, DEPARTMENT as her MAJOR_DEPT, whereas a DEPARTMENT may have between none and many STUDENTs for which it is the MAJOR_DEPT.*
- *Now, let's take the MINOR_DEPT relationship type. The pairs are (0,N) on the STUDENT side and (0,1) on the DEPARTMENT side. We read it as follows: a STUDENT may have no DEPARTMENT as her MINOR_DEPT or she may have exactly one, whereas a DEPARTMENT may have between none and many STUDENTs for which it is the MINOR_DEPT.*

T2. *Here an example of what the textual form of the requirements that led to the given ER diagram might have looked like*:

I. **The database represents each AIRPORT, keeping its unique AirportCode, the AIRPORT Name, and the City and State in which the AIRPORT is located.**

II. **Each airline FLIGHT has a unique number, the Airline for the FLIGHT, and the Weekdays on which the FLIGHT is scheduled (for example, every day of the week except Sunday can be coded as 'MTWTFS-').**

III. **A FLIGHT is composed of one or more FLIGHT LEGs (for example, flight number CO1223 from New York to Los Angeles may have two FLIGHT LEGs: leg 1 from New York to Houston and leg 2 from Houston to Los Angeles). A FLIGHT LEG must be associated with a specific FLIGHT but is merely weakly identified by a leg number. Each FLIGHT LEG has a unique DEPARTURE AIRPORT and Scheduled Departure Time, and a unique ARRIVAL AIRPORT and Scheduled Arrival Time. Of course, airports have many arriving and departing flight legs.**

IV. **A FLIGHT carries passengers on different FAREs, for which the Code, the Amount in dollars that it costs and the Restrictions associated must be stored. The Code is a weak identifier and it must be bound to a FLIGHT to be properly stored.**

V. **A LEG INSTANCE is an instance of a FLIGHT LEG on a specific Date (for example, CO1223 leg 1 on July 30, 1989). The actual Departure Time and Arrival Time at AIRPORTs are recorded for each flight leg after the flight leg instance has been concluded. The Number of available seats and the AIRPLANE used in the LEG INSTANCE are also kept.**

VI. **The data stored for each RESERVATION on each LEG INSTANCE consists of the Customer Name, Phone, and Seat Number.**

VII. **Information on AIRPLANEs and AIRPLANE TYPEs is also kept. For each AIRPLANE TYPE (for example, DC-10), the TypeName, manufacturing Company, and Maximum Number of Seats are kept. The AIRPORTs in which planes of this type CAN LAND are kept in the database. For each AIRPLANE, the AirplaneId, Total number of seats, and TYPE are kept.**

The above sample answer is realistic, but not a model answer. A truly exemplary specification would be much more meticulous, comprehensive and precise about the structural constraints, and, in particular, about the unusually large number of weak entity types. Therefore, take this sample answer as being the result of a first approximation, one that needs to be further refined via more contact time with the users to prise out all the details involved.

If you were more meticulous than the sample answer above, well done! One aspiration is to adopt a kind of controlled English, one that follows a template format that we strive to reach by refinement, one in which we record entity types, then their attributes, then the relationship types and, for each, the structural constraints, all in a rather stilted, structured kind of English.

Note also that the diagram given is not very consistent in using verbal forms to name relationship types. Again, if you could write your answer trying to do so (which would then count as corrections to the diagram you were given to work on), that would be a good exercise in rigour and consistency.