Review questions:

- What is the difference between logical and physical models?

Logical shows what a system is, or does. It is implementation-independent. Physical shows what the system is, or does, but also shows how it is physically or technically implemented. Physical models are thus implementation-dependent.

- Why is it important to create an implementation independent model of a system?

An implementation-independent model of a system allows the design of the system to remain stable and usable even when the technology supporting the system changes.

- Why is it necessary to create an implementation dependent model of a system?

An implementation-dependent view of a system depicts available technology choices, and the limitations of those choices.

- What is an entity? What are entity instances?

Entity: It is a class of persons, places, objects, events, or concepts about which we need to capture and store data. (I.e. think of it as a noun, or a thing such as a 'student' or 'bird.').

- A relationship is a natural business association between entitles. What is the relationship between student and teacher? Does it depend on how many classes a student can take, or how many classes a teacher can teach?

A student is taught by a teacher, or conversely a teacher teaches a student. While the relationship itself is not affected by the number of classes taught or taken, the cardinality is affected by the number of classes a student can take (e.g. If a student may take only one class, the cardinality of the relationship is different than if the student can take many classes.).

- What is cardinality? Give an example?

Cardinality is the minimum and maximum number of occurrences of one entity that can be related to a single occurrence of another entity. An ex-ample would be: A student can have one or many teachers.

Problems and Exercises:

- What is a reasonable domain for the data attribute for a student's last name?

Text (12)

Note: Allowing more than 12 characters means that the database will catch the full last names of students with more than 12 characters in length. But, you must weigh the length of the average last name against the need to capture all of every last name. That is, increasing the domain size to say, 20 characters will allow students with names 19 characters long to be entered in their entirety. But, if the majority of last names have 5 characters, then there will be a lot of wasted space taken in the database (as the database must reserve that character length whether it is used, or not).

- What default value would you choose for a student's last name?

Required:

Note: No value in the last name domain is common enough to set as the default. Since this attribute would usually be required in a database, set the default to 'required.

- What default value would you choose for gender?

None given'

Note: This one is tricky. As a system designer, you must be sensitive to is-sues for which you may not be familiar. In the case of gender, a simple M or F is not permissible. A person may choose not to reveal their gender, or may have gender-identity issues that preclude them from answering. Thus, 'none given' or a similar default is best.

- The student table you are working with contains the attributes: STUDENT ID, ,NAME, PHONE NUMBER, and MAJOR. Normalize to 3NF ?

Table will contain: Student_ID, First_Name, Last_Name, Home_Phone, Work_Phone, Emergency_Phone, and Major.

Note: Name and Phone_Number were not atomic. Thus, they needed to be reduced to meet the requirements of 1st NF.

- What attributes would you have in a table to describe a movie?

Movie_Identifying_Number,

Title, Release_Date, Rating, Lead_Actor,

Lead_Actress, Producer

- A many to-many relationship (also called a nonspecific relationship) can and generally should he resolved into a pair of one.-to-many relationships with an associative entity. When is this not the case?

Many to many relationships are resolved using entities rather than associative entity, when the many to many relationship is caused by a forgotten entity in the data model.

- Give an example of a many-to-many relationship. Resolve using an entity or an associative entity. Which did you use? Why?

See page 305 for an example of resolving involving entities and associative entities. Remember to use the associative entity when all business entities have been correctly included in the diagram. Otherwise, use an entity. Re-call that some many to many relationships occur because an important entity has been forgotten in the model.

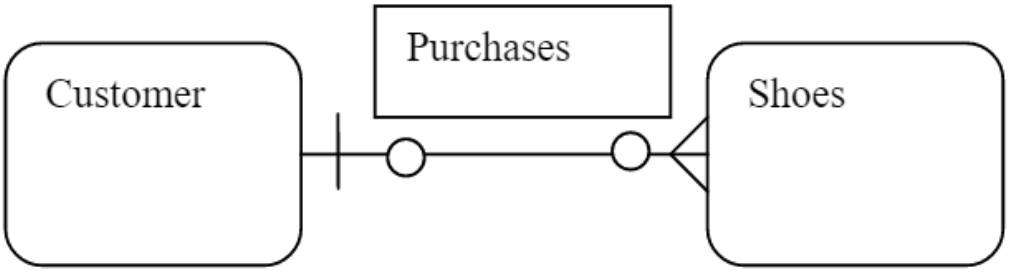
- Describe each of the first three normal forms. Give an example of each?

1st NF: All attributes must be atomic (no multivalued or composite attrib-utes) Example: Student(Name(First_Name, Last_Name), {phone_number}) Becomes: Student(First_Name, Last_Name, Home_Phone, Work_Phone, Cell_Phone)

2nd NF: 1st + all attributes must be fully functionally dependent Consider a table where there is a joint primary key of Student_Id and Class_Id for a student's identifying number and the identifying number of aclass at a college. An appropriate attribute in the table would be: Stu-dent_Grade: i.e. a student's grade in a particular class. Note that the at-tribute is both dependent upon the Student_Id and the particular class, or Class_Id. An attribute for Teacher_Name would not be appropriate, since the teacher of a particular class is dependent only upon the Class_Id and not the Student_Id

3rd NF: 2nd + no transitive dependencies Example: A table with Student_Id as the primary key has an attribute of Date (for entrance into the program). Also included in the table is an attrib-ute for Weather. Since weather is dependent on the Date, and Date is de-pendent on the primary key, there is a transitive dependency. Move Weather to another table with a primary key of Date.

- A customer goes to a shoe store and purchases several pairs of shoes. Diagram this relationship?



- Give an example each of ternary, identifying, and nonidentifying relationships ?

see pg. 279 for example of Identifying and Non-Identifying Relationships

And see pg. 277 for an example of a Ternary Relationship.

- On the surface. data modeling appears not to requiar much creativity. Why is this incorrect?

Data modeling is much like drawing a picture of a database. A creative person will create a model that incorporates entities and attributes that are advantageous to the company as well as non-intrusive to the customer. A good modeler will balance the need for information with the realities of constraints on getting, keeping and maintaining that information. All of this takes creativity.

- Can a well-designed database give a business a strategic advantage? How?

Absolutely, yes. A well designed database offers management strategic information about such things as product usage and marketing possibilities, as well as supply and inventory issues – this cuts costs as well as enables management to exploit new opportunities. For employees that interact directly with customers, a well designed database can enable the employee to offer superior customer service, individualized attention and enable them to resolve problems more easily. This increase in customer service will help the company keep returning customers.