**Chapter 2 - Review Questions**

**3. How do you translate business rules into data model components?**

* As a general rule, a noun in a business rule will translate into an entity in the model, and a verb (active or passive) associating nouns will translate into a relationship among the entities. For example, the business rule "a customer may generate many invoices" contains two nouns (customer and invoice) and a verb ("generate") that associates them.

**8. Why is an object said to have greater semantic content than an entity?**

* An object has greater semantic content because it embodies both data and behavior. That is, the object contains, in addition to data, also the description of the operations that may be performed by the object.

**13. What is a relationship, and what three types of relationships exist?**

* A relationship is an association among (two or more) entities. Three types of relationships exist:
  1. One-to-one (1:1)
  2. One-to-many (1:M)
  3. Many-to-many (M:N or M:M.)

**15. What is a table, and what role does it play in the relational model??**

* Strictly speaking, the relational data model bases data storage on relations. These relations are based on algebraic set theory. However, the user perceives the relations to be tables. In the relational database environment, designers and users perceive a table to be a matrix consisting of a series of row/column intersections. Tables, also called relations, are related to each other by sharing a common entity characteristic. For example, an INVOICE table would contain a customer number that points to that same number in the CUSTOMER table. This feature enables the RDBMS to link invoices to the customers who generated them. Tables are especially useful from the modeling and implementation perspectives. Because tables are used to describe the entities they represent, they provide an easy way to summarize entity characteristics and relationships among entities. And, because they are purely conceptual constructs, the designer does not need to be concerned about the physical implementation aspects of the database design.

**Chapter 2 - Problems**

**17. United Broke Artist (UBA) is a broker for not-so-famous painters. UBA maintains a small network database to track painters, paintings, and galleries. Using PAINTER, PAINTING, and GALLERY, write the network structure and identify appropriate sets within the UBA database. (*Hint 1*: A painting is painted by a particular artist, and that painting is exhibited in a particular gallery, *Hint 2*: A gallery can exhibit many paintings, but each painting can be exhibited in only one gallery. Similarly, a painting is painted by a single painter, but each painter can paint many paintings.)**

UBA Network structure

PAINTER

GALLERY

1:M

1:M

Artist set

Exhibition set

PAINTING

**18. If you decide to convert the network database in Problem 17 to a relational database:**

**a. What tables would you create and what would the table components be?**

* + The three tables would be

|  |  |
| --- | --- |
| Table Name | Table components |
| PAINTER | PTR\_NUMBER, PTR\_NAME, PTR\_PHONE |
| PAINTING | PTG\_NUMBER, PTG\_TITLE, PTR\_NUMBER, GAL\_NUM |
| GALLERY | GAL\_NUM, GAL\_NAME, GAL\_ADDRESS |

**b. How might the (independent) tables be related to one another?**

* + The PAINTING table will be related to both the GALLERY and PAINTER tables. The PAINTING table will contain the attribute PTR\_NUMBER, which will relate it to the PAINTER table. The PAINTING table will also contain the GAL\_NUM attribute, which will relate it to the GALLERY where the painting is being shown.

**19. Using a Crow’s Foot ERD, convert the network database model in Figure 2.2 into a design for a relational database model. Show all entities and relationships.**

* The network database model based in Figure 2.2 converted into a Crow’s Foot ERD would look like:

Generates

Writes

Is recorded in

Includes

Makes

PRODUCT

SALESREP

INV\_LINE

PAYMENT

INVOICE

CUSTOMER