**1. What is the purpose of an E-R model?**

The notations in the sample *E-R models* in this chapter reflect only one way of diagramming entity

relationships. To design a database, the requirements of the database

—inputs, processes—outputs—must be identified first. Usually, the first question asked is, “What information, or output, must come from this database?” or “What questions should this database be able to answer?” By understanding the necessary output, the designer can then determine what information should be stored in the database. An E-R model is also called an entity-relationship diagram (ERD).

**2. What is an entity?**

In an E-R model, an entity is any person, place, or thing with characteristics or attributes that will be

included in the system.

An E-R model is a diagram that identifies the entities (customers, books, orders, and such) in the database, and it shows how the entities are related to one another. It serves as the logical representation of the physical system to be built.

3. **Give an example of three entities that might exist in a database for a medical office, and list some attributes that would be stored in a table for each entity**.

Doctor’s attributes, patient’s attributes, medicine’s attributes

**4. Define a one-to-many relationship.**

* One-to-one: In a one-to-one relationship, each occurrence of data in one entity is represented by only one occurrence of data in the other entity. For example, if each classroom is assigned to only one academic division, a one-to-one relationship is created between the classroom and division entities. This type of relationship is depicted in an E-R model as a simple straight line.
* One-to-many: In a one-to-many relationship, each occurrence of data in one entity can be represented by many occurrences of the data in the other entity. For example, a class has only one instructor, but an instructor might teach many classes. A one-to-many relationship is represented by a straight line with a crow’s foot at the “many” end.
* Many-to-many: In a many-to-many relationship, data can have multiple occurrences in both entities. For example, a class can consist of more than one student, and a student can take more than one class. A straight line with a crow’s foot at each end indicates a many-to-many relationship.

**5. Discuss the problems that can be caused by data redundancy.**

Database Normalization

Many people unfamiliar with database design principles often ask, “Why not just put all the data in one big table?” This single-table approach leads to problems of data redundancy (duplication) and data anomalies (data inconsistencies). To avoid these data issues, database normalization is used to create a design that reduces or eliminates data redundancy and, therefore, avoids data anomalies. In general, normalization helps database designers determine which attributes, or fields, belong to each entity. In turn, this information helps determine which fields belong in each table. Normalization is a multistage process that enables designers to take the raw data to be collected about an entity and develop the data into a structured, normalized form thatreduces the risks associated with data redundancy. Data redundancy poses a special problem in databases because storing the same data in different places can cause problems when updates or changes to data are required.

**6. Explain the role of a primary key.**

A primary key is used to uniquely identify each record. After each table has been normalized, make certain all relationships among tables have been established. When data that already exists, such as a book ISBN, is used as a primary key, it’s often referred to as an intelligent key or a natural key. At times, data serving as a primary key doesn’t exist, so a system generated unique value is used as a primary key.

**7. Describe how a foreign key is different from a primary key.**