INTRODUCTION

With the rising complexity of database applications, the basic concepts of Entity-Relationship (ER) modeling (as originally developed by Chen, 1976) were no longer sufficient. So the basic ER model was extended to include generalizations and specializations (Bagui & Earp, 2003; Elmasri & Navathe, 2003) and the concept of categories (Elmasri, Weeldreyer & Hevner, 1985). An ER model which includes all the concepts of the original ER model and the additional concepts of generalizations/specializations and categories is often referred to as the Extended ER (EER) model (Elmasri & Navathe, 2003). In this short paper, we shed some light on these relationship concepts, concepts that database designers often find difficult to model directly (Engels et al., 1992/1993). We also discuss the mapping rules for generalizations/specializations and categories. Important contributions in this area are also reported in (Elmasri et al., 1985; Markowitz & Shoshani, 1992; Dey, Storey & Barron, 1999; Wiederhold & Elmasri, 1980). Song, Evans, and Park (1995) also give a good comparative analysis of other types of ER diagrams and their notations.

Due to the short nature of this paper, we will keep the discussion focused on implementing generalizations and specializations in relational databases; their parallel implementation in objects will not be covered. Also, the discussion of the concept of inheritance will center around generalizations/specializations and categories in EER diagrams, without getting into an almost equivalent notion in object-oriented (OO) theory, ORM (Object-Role Modeling), and UML (Unified Modeling Language) class diagrams.

BACKGROUND

A generalization/specialization relationship models a superclass/subclass type of relationship. A generalization is an abstraction mechanism that allows for viewing of entity-sets as a single generic entity-set. The attributes and associations which are common to the entity-sets are associated with the generic (generalized) entity-set. The inverse of generalization is called specialization.

GENERALIZATION/RELATIONSHIPS

If we are modeling a hospital database, and we want to store information about the hospital’s nurses, technicians, and physician assistants, we could create separate entities such as NURSE, TECHNICIAN, and PHYSICIAN ASSISTANT. But, these three entities would also have a lot of fields in common, for example, name, social security number, address, phone, and so forth. So, it would be a good idea to have an entity set called EMPLOYEE containing these common fields, and entity subsets, NURSE, TECHNICIAN and PHYSICIAN ASSISTANT, that could inherit this information from the EMPLOYEE entity set. In this case, the EMPLOYEE entity set would be called the superclass. This superclass is a generalization of the entity subsets, NURSE, TECHNICIAN, and PHYSICIAN ASSISTANT. The NURSE, TECHNICIAN, and PHYSICIAN ASSISTANT would be called the subclasses. The subclasses are specializations of the superclass, EMPLOYEE, and inherit from the superclass. Several specializations can be defined for the same entity type (or superclass).

The subclass, denoted by a separate entity rectangle in the EER diagram, is considered to be a part of the superclass entity set, EMPLOYEE. Although it will have attributes that distinguish it from other subclasses, it is considered only a subset of the EMPLOYEE entity set. That is, all nurses are employees, but the reverse is not true—not all employees are nurses. Likewise, all techni-
cians or physician assistants are employees, but all employees are not technicians or physician assistants.

Figure 1 shows this generalization/specialization example. We use Elmasri and Navathe’s (2003) diagrammatic notations for the EER diagrams. The subset symbol, “⊂”, indicates the direction of the superclass/subclass or parent-child, inheritance relationship. This superclass/subclass relationship is also often referred to as a IS-A or IS-PART-OF relationship (Sanders, 1995).

Constraints on Generalization/Specialization Relationships

Generalizations and specializations can have two types of constraints: (1) the disjoint/overlap relationship constraint and (2) participation constraints – total or partial. The combinations of these constraints can be (1) disjoint and total participation; (2) disjoint and partial participation; (3) overlap and total participation; or (4) overlap and partial participation. First we will discuss disjoint/overlap relationship constraints, and then we will discuss participation constraints, giving examples of combinations of the constraints along the way.

Disjoint/Overlap Relationship Constraints

Generalization/specialization relationships may be disjoint or they may overlap. A disjoint relationship is shown by a “d” in the circle attaching the superclass to the subclass or subclasses (as shown in Figure 1). A disjoint relationship means that an entity from the superclass can belong to only one of the subclasses (can be of only one specialization). For example, according to Figure 1, an EMPLOYEE can be at most a member of only one of the subclasses – PHYSICIAN ASSISTANT, NURSE, or TECHNICIAN. An employee cannot be a physician assistant as well as a nurse, or cannot be a nurse as well as a technician.

An overlap relationship is shown by an “o” in the circle attaching the superclass to the subclass or subclasses (as shown in Figure 4). Overlap means that an entity from the superclass can belong to more than one subclass (specialization). For example, according to Figure 4, a computer must be either a laptop or a desktop, or both a laptop and a desktop.

Participation Constraints

The second type of constraint on generalization/specialization relationships is participation constraints, which may be total (or full) or partial. As in the ER model (Bagui & Earp, 2003; Elmasri & Navathe, 2003), we will show a full or total participation between the superclass and subclass entities by double lines, and a partial participation between the superclass and subclass entities by single lines. Partial participation is shown in Figure 1. Figure 1 can be read as:

An EMPLOYEE may either be a PHYSICIAN ASSISTANT, NURSE, or TECHNICIAN.

Figure 1 shows a disjoint, partial participation relationship. The “may” means partial participation between the EMPLOYEE superclass and the respective subclasses entities. That is, not all employees of the EMPLOYEE entity set belong one of the subclasses, PHYSICIAN ASSISTANT, NURSE, or TECHNICIAN. One may ask, why? Or how? To answer this, we will extend Figure 1 to include another subclass, as shown in Figure 2. We now have an Employee from the EMPLOYEE entity set who may also belong to the HEAD subclass. Once again, the “may” is inferred from the single line between the superclass (or generalization) entity, EMPLOYEE, and the subclass (or specialization) entity, HEAD. Figure 2 can be read as:

An Employee may be a HEAD or PHYSICIAN ASSISTANT or NURSE or TECHNICIAN. Or, an Employee may be both a HEAD and a PHYSICIAN ASSISTANT, or both a HEAD and a NURSE, or both a HEAD and a TECHNICIAN.

An example of total or full participation is shown in Figure 3. We can read Figure 3 as:

An EMPLOYEE must either be a PHYSICIAN ASSISTANT, NURSE, or TECHNICIAN.

The “must” means total participation. So, an EMPLOYEE must belong to one of the subclasses. That is, all employees of the EMPLOYEE entity set must belong to one of the subclasses. But although there is total participation in Figure 3, the employee cannot belong to more than one subclass because of the “d” or disjoint relationship. Figure 3 shows a disjoint, full participation relationship, and Figure 4 shows an overlap, full participation relationship.

Mapping Generalizations and Specializations to a Relational Database

Rules to map generalizations/specializations to a relational database depend on the constraints on the generali-