

ENTITY<sub>1</sub> and ENTITY<sub>2</sub> having the attributes ( $Atr_{11}, \dots, Atr_{1n}$ ) and ( $Atr_{21}, \dots, Atr_{2n}$ ), respectively. The attributes of the relationship are ( $Atr_{R1}, \dots, Atr_{Rn}$ ). The relationship ENROLLMENT in Figure 2.21 is many to many. In Figure 2.22, the relationship MARRIAGE is one-to-one and REPORTS\_TO is one-to-many.

Before discussing the E-R model in more detail, we reexamine the two components of the E-R model: entities and relationships.

### 2.4.1 Entities

As discussed in Chapter 1, an entity is an object that is of interest to an organization. Objects of similar types are characterized by the same set of attributes or properties. Such similar objects form an entity set or entity type. Two objects are mutually distinguishable and this fact is represented in the entity set by giving them unique identifiers.

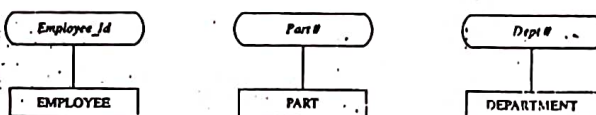
Consider an organization such as a hotel. Some of the objects of concern to it are its employees, rooms, guests, restaurants, and menus. These collections of similar entities form the entity sets, EMPLOYEE, ROOM, GUEST\_LIST, RESTAURANT, MENUS.

Given an entity set, we can determine whether or not an object belongs to it. An object may belong to more than one entity set. For example, an individual may be part of the entity set STUDENT, the entity set PART\_TIME\_EMPLOYEE, and the entity set PERSON. Entities interact with each other to establish relationships of various kinds.

Objects are represented by their attributes and, as objects are indistinguishable, a subset of these attributes forms a primary key or key for uniquely identifying an instance of an entity. Entity types that have primary keys are called strong entities. The entity set EMPLOYEE discussed in Section 2.2 would qualify as a strong entity because it has an attribute *Employee\_Id* that uniquely identifies an instance of the entity EMPLOYEE; no two instances of the entity have the same value for the attribute *Employee\_Id*. Figure 2.18 shows some examples of strong entities. Only the attributes that form the primary keys are shown.

Entities may not be distinguished by their attributes but by their relationship to another entity. Recall the representation of the entity EMPLOYEE wherein the 1:M association involving the attributes (*Dependent\_Name*, *Relationship\_to\_Employee*) is removed as a separate entity, DEPENDENTS. We then establish a relationship, DE-DUCTIONS, between the modified entity EMPLOYEE and DEPENDENTS as

Figure 2.18 Strong entities.



Existence Dependencies - If the existence of entity X depends on the existence of entity Y, then X is said to be existence dependent on

Y. (Here Y is the dominant entity and X is the subordinate entity)

Ex. <sup>dominant</sup> account and <sup>subordinate entity</sup> transaction sets and a relationship log between them (1:M)  
If account entity is deleted, its associated transaction entity must also be deleted.

Figure 2.19 Converting an attribute association to a relationship.



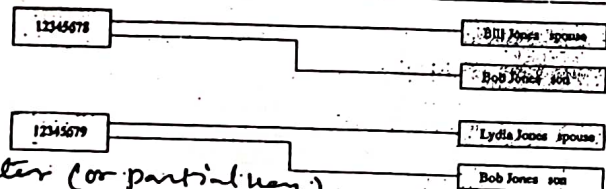
shown in Figure 2.19. In this case, the instances of the entity from the set DEPENDENTS are distinguishable only by their relationship with an instance of an entity from the entity set EMPLOYEE. The relationship set DEDUCTIONS is an example of an identifying relationship and the entity set DEPENDENTS is an example of a weak entity.

Instances of a weak entity set are distinguished by the values of the attributes of the strong entity set. For instance, the EMPLOYEE 12345678 (the Jones) in Figure 2.7 has two DEPENDENTS, Bill Jones, spouse and Bob Jones, son. These are distinct and can be distinguished from each other. The organization could have another Jones in its employ (with given name Jim and Employee-id = 12345679), who has dependents Lydia Jones, spouse and Bob Jones, son. This is illustrated in Figure 2.20. Note also that by adding attributes such as Social Security Number of the dependent to the weak entity it can be converted into a strong entity set. However, there may be no need to do so in a given application if there is an identifying relationship.

The two instances (Bob Jones, son) of the weak entity set DEPENDENTS associated with different instances of the strong entity set EMPLOYEE are not distinguishable from each other. They are nonetheless distinct because they are associated with different instances of the strong entity set EMPLOYEE. The primary key of a weak entity set is thus formed by using the primary key of the strong entity set to which it is related, along with the discriminator of the weak entity. We rule out the case where a dependent such as Bob Jones is the son of two different employees, namely his mother and father, since only one of them will claim him as a dependent!

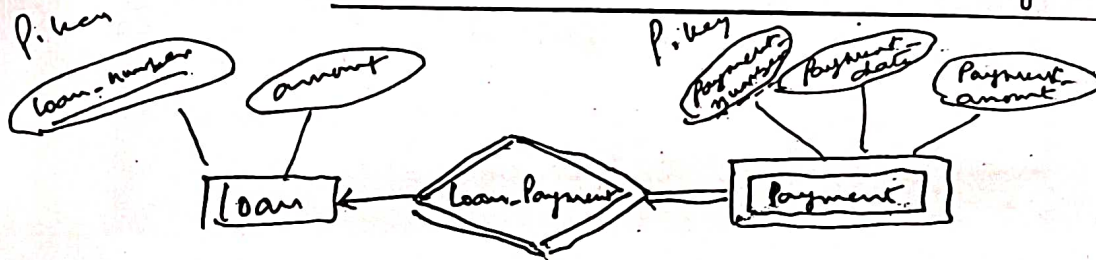
An entity set that does not have a primary key is referred to as a weak entity set.

Figure 2.20 Instances of a 1:M converted relationship.



The discriminator (or partial key) of a weak entity set is the set of attributes that distinguishes among all the entities of a weak entity set.

We depict a weak entity set by double rectangles.



Payment-number - discriminator of the payment entity set  
we underline the discriminator of a weak entity set with a dashed line.

The P. key of a W.E.S. is formed by the P. key of the S.E.S. on which the W.E.S. is existence dependent, plus the W.E.S.'s discriminator.



If loan-number were explicitly stored, payment could be made a strong entity, but then the relationship between payment and loan would be duplicated by an implicit relationship defined by the attribute loan-number common to payment and loan.