Modeling Date in the organization

（shown in the Relationship & Entity cell）between course,not a multivalued attribute of COURSE .Representing prerequisites of a course and finding the course for which a course is prerequisite both deal with relationships between entity types, When a prerequisite is a multivalued attribute of COURSE ,finding the courses for which a course is a prerequisite means looking for a specific value for a prerequisite across all COURSE instances.as was shown in Figure 2-13a, such a situation could also be modeled as a unary relationship among instance of the COURSE entity type .In Visio, this specific situation requires creating the equivalent of an associative entity (see the Relationship & Entity cell in figure 2-15a;Visio does not use the rectangle with rounded corners symbol).By creating the associative entity, it is now easy to add characteristics to the relationship, such as a minimum grade required. Also note that Visio shows the identifier(in this case compound)with a PK stereotype symbol and boldface on the component attribute names, signifying these are required attributes.

In Figure 2-15b, employees potentially have multiple skills(shown in the Attribute cell),but skill could be viewed instead as an entity type(shown in the Relationship & Entity cell as the equivalent of an associative entity)about which the organization wants to maintain data(the unique code to identify each skill,a descriptive title,and the type of skill,for example, technical or managerial ).An employee has skills, which are not viewed as attributes, but rather as instances of related entity type, In the cases via a relationship with another entity type may, in the view of some people, simplify the diagram. On the other hand, the right-hand drawings in the these figures are closer to the system, the most popular type of DBMS in use today. Although we are not concerned with implementation during conceptual data modeling, there in some logic for keeping the conceptual and logical data models similar. Further, as we will see in the next example, there are times when an attribute, whether simple, composite, or multivalued, should be in a separate entity.

So, when should an attribute be linked to an entity type via a relationship? The answer is when the attribute is the identifier or some other characteristic of an entity type in the data model and multiple entity instances need to share these same attributes. Figure 2-15c represents an example of this rule. In this example, EMPOLYEE has a composite attribute of Department. Because Department is a concept of the business, and multiple employees will share the same department data, department data could be represented(nonredundantly) in a DEPARTMENT entity type, with attributes for the data about departments that all other related entity instances need to know. With this approach, not only can different employees share the storage of the same department data, but projects(which are assigned to a department) and organizational units(which are composed of departments) also can share the storage of this same department data.

**Cardinality Constraints**

There is one more important data modeling notation for representing common and important business rules. Suppose there are two entity types, A and B, that are connected by a relationship. A **cardinality constraint** specifies the number of instances of entity B that can (or must)be associated with each instance of entity A. For example, consider a video store that rents DVDs of movies. Because the store may stock more than one DVD for each movie, this is intuitively a one-to-many relationship, as shown in Figure 2-16a. Yet, it is also true that the store may not have any DVDs of a given movie in stock at a particular time (e.g., all copies may be checked out). We need a more precise notation to indicate the range of cardinalities for a relationship. This notation was introduced in Figure 2-2, which you may want to review at this time.

**MINIMUM CARDINALITY**

The **minimum cardinality** of a relationship is the minimum number of instances of entity B that my be associated with each instance of entity A. In our DVD example, the minimum number of DVDs for a movie is zero. When the minimum number of participants is zero , we say that entity type B is an optional participant in the relationship. In this example ,DVD(a weak entity type) is an optional participant in the Is stocked As relationship . This fact is indicated by the symbol zero through the line near the DVD entity in Figure 2-16b.

**MAXIMUM CARDINALITY**

The **maximum cardinality** of a relationship is the maximum number of instances of entity B that may be associated with each instance of entity A. in the video example, the maximum cardinality for the DVD entity type is “many”-that is, an unspecified number greater than one .this is indicated by the “crow’s foot “ symbol on the line next to the DVD entity symbol in Figure 2-16b.(You might find interesting the explanation of the origin of the crow’s foot notation found in the Wikipedia entry about the entity-relationship model ;this entry also shows the wide variety of notations used to represent cardinality ; see [http://en.wikipedia.org/wiki/Entity-relationship-mode](http://en.wikipedia.org/wiki/entity-relationship-modle)l.)

A relationship is, of course,bidirectional,so there is also cardinality notation next to the MOVE entity. Notice that the minimum and maximum are both one(see Figure2-16b). This is called a mandatory one cardinality.In other words,each DVD of a movie must be a copy of exactly one movie.In general, participation in a relationship may be optional or mandatory for the entities involved. If the minimum cardinality is zero,participation is optional;if the minimum cardinality is one, participation is mandatory.

In Figure 2-16b,some attributes have been added to each of the entity types.Notice that DVD is represented as a weak entity. This is because a DVD cannot exist unless the owner movie also exists. The identifier of MOVIE is Movie name.DVD does not have a unique identifier.however,copy number is a partial identifier, which together with movie name would uniquely identify an instance of DVD.

**Some examples of relationships and their cardinalities**

Examples of three relationships that show all possible combinations of minimum and maximum cardinalities appears in figure 2-17.each example states the business rule for each cardinality constraint and shows the associated E-R notation.each example also shows some relationship instances to clarify the nature of the relationship. You should study each of these example carefully. Following are the business rules for each of the example in Figure 2-17:

1. **PATIENT Has Recorded PATIENT HISTORY(Figure 2-17a)**

Each patient has one or more patient histories. ( The initial patient visit is always recorded as an instance of PATIENT HISTORY.) Each instance of PATIENT HISTORY”belongs to “ exactly one

PATIENT.

1. **employee is assigned to project (figure 2-17b)**

Each project has at least one EMPLOYEE assigned to it .(some projects have more than one .)each employee may or (optionally)may not be assigned to any existing project (e.g, employee Pete )or may be assigned to one or more projects.

3.person is married to person this is an optional zero or one cardinality in both direction ,because a person may or not be married at a given point in time.

It is possible for the maximum cardinality to be fixed number ,not an arbitrary many value .for example suppose corporate policy states that an employee may work on at most five project at the same time .we could show this business rule by placing a 5 above or below the crow s foot next to the project entity in figure 2-17b.

**A ternary relationship**

We showed the ternary relationship with the associative entity type SUPPLY SCHEDULE in Figure 2-14. Now let’s add cardinality constraints to this diagram, based on the business rules for this situation. The E-R diagram, with the relevant business rules, is shown in Figure 2-18. Notice that PART and WAREHOUSE must relate to some SUPPLY SCHEDULE instance, and a VENDOR optionally may not participate. The cardinality at each of the partcipating entities is a mandatory one, because each SUPPLY SCHEDULE instance must be related to exactly one instance of each of these participating entity types. (Remember, SUPPLY SCHEDULE is an associative entity.)

As note earlier, a ternary relationship is not equivalent to three binary relationships. Unfortunately, you are not able to draw ternary relationships with many CASE tools; instead, you are forced to represent ternary relationships as the three binaries(i.e.,