CHAPTER 7

ENTITY-RELATIONSHIP DATA MODEL

CHAPTER OBJECTIVES

- Study entity-relationship (E-R) data modeling as another technique to create a semantic model
- Understand the principles and concepts of E-R modeling
- Discuss the components of the E-R data model
- Explore special types of business entities and relationships
- Examine an entity-relationship diagram

Database practitioners and experts refer to a data model that captures the true meaning of real-world information requirements as a semantic data model. Such a model truly represents the semantics or real meaning and implications of what you need to model from the relevant real-world information. We have studied the object-based data modeling technique as a technique for creating a semantic data model. Now we want to study another popular technique for building a semantic data model: entity-relationship (E-R) data modeling.

As you review the concepts, principles, and components of the E-R data model, you will notice a number of similarities with the object-based data model. Both achieve the same purpose; both are intended to produce similar results; both reflect the true meaning of information requirements. As we proceed with our discussion, wherever essential, we will compare the two techniques and point out the correspondences.

Because of the similarities, we need not go over again the basic principles and examples of the mapping of real-world information to the components of the model.

Database Design and Development: An Essential Guide for IT Professionals by Paulraj Ponniah
ISBN 0-471-21877-4

These are basically the same for both modeling techniques. We will therefore concentrate on presenting the various components and their notations used to draw the E-R data model diagram. Because the two data modeling methods were developed independently, the notations or symbols used for the components are somewhat different. However even in the notations, you will note many resemblances.

Study this chapter for the purpose of reinforcing the concepts already learned in the discussion of object-based data modeling. Note the variations in certain notions. Also, observe how the E-R data model includes additional notations to represent individual types within a particular component.

INTRODUCTION TO E-R MODEL

First of all, note that the E-R data model is another generic model similar to the object-based data model. You will recall how a generic model is free from the rules and restrictions of conventional models such as hierarchical, network, and relational models. As you know, these conventional models call for information requirements to be perceived in certain given ways and their relationships to be established according to defined conventions. Being a generic model, the E-R data model can apply to all types of nuances in information requirements. Again, because it is a generic data model, the E-R model is amenable to being transformed to any particular conventional model based on the implementation of your database system.

The E-R data modeling technique is a widely used method. You have a choice of many data modeling tools to create the model. Because of its popularity and widespread use, the notations and symbols are more standardized. This modeling technique provides a wider range of notations to represent variations within each component.

Basic Concepts

By way of introduction to the basic concepts and principles of E-R modeling technique, note the following points:

Mainstream approach. Most data modelers and practitioners seem to prefer the E-R modeling technique, probably because of the availability of several CASE tools to support the technique.

Things and associations. Examining any set of information requirements, you discover that these are really about things and their associations—things that are of interest to the business and associations that are relevant to the business processes. The E-R modeling technique recognizes this underlying premise.

Entities and relationships. As the name of the technique signifies, the entity-relationship modeling technique focuses on entities and relationships as natural concepts. Entities are the things the business is interested in, and relationships are the associations among the things.

Description of information. The data modeling technique describes real-world information in terms of entities, their attributes, and the relationships among the entities.

Graphics and rules. You draw an entity-relationship diagram (ERD) according to appropriate rules by using well-defined graphical symbols.

Theoretical foundation. The modeling technique rests on a solid foundation on the principles of entities, attributes, and relationships.

Means for communication. An E-R model proves to be a useful and easily understood means for communication among the IT professionals on the database project as well as with user representatives.

Data Modeling Process

E-R data modeling seems to be a natural way of looking at real-world information without being particularly guided by business processes. Following up on this realization, Figure 7-1 illustrates how E-R data modeling derives the components, one by one, as a result of natural observation of real-world information.

The figure essentially illustrates the process of creating an E-R model. Carefully note the boxes on the left side of the figure showing how each observation seems to follow naturally from the previous one. Let us see how this process of creating an E-R model works from one observation to the next. This is how the model is created from observations made of the information requirements.

Consider a typical organization without getting into too much detail. Let us record the flow of observations and the derivation of the components.

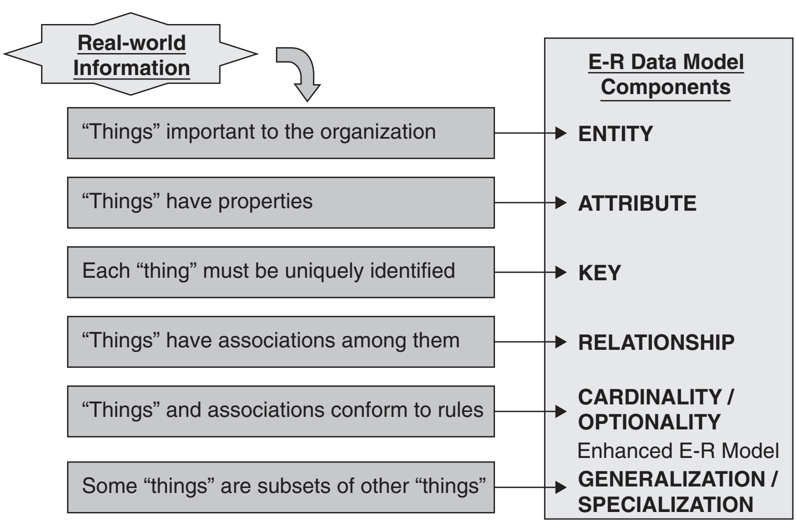


Figure 7-1 Deriving E-R data model components.

Certain "things" are important to the organization. List these things and derive these as entities of DEPARTMENT, EMPLOYEE, CUSTOMER, ORDER, and INVOICE.

Each of these "things" has properties. Mark these properties or characteristics and derive these as attributes of each of the entities DEPARTMENT, EMPLOYEE, CUSTOMER, ORDER, and INVOICE. For example, note that EMPLOYEE has characteristics such as EmployeeName, EmployeeAddress, and SocSecNumber. Derive these as attributes of EMPLOYEE.

Each case or occurrence of a "thing" must be uniquely identified. Select one or more attributes in each entity so that the values of these attributes may be used to uniquely identify an occurrence of an entity. Select SocSecNumber as the identifier or key for EMPLOYEE.

"Things" have associations among them. Derive relationships among entities from the associations. CUSTOMER places ORDER. Derive a relationship between the two entities from this association.

"Things" and associations conform to rules. One customer may place one or more orders. There may some customers who have not placed any orders. Rules govern the association between CUSTOMER and ORDER. From such rules, derive the cardinality and optionality of the relationship.

Some "things" are subsets of other "things." Individuals and institutions may be customers to the organization. From this concept, derive INDIVIDUAL and INSTITUTION as subsets of CUSTOMER.

In earlier versions of E-R modeling, the natural sequence had stopped with the component "Cardinality/Optionality" mentioned above. Deeper observation of real-world information requirements reveals the existence of supersets and subsets as special types of entities. Therefore, the E-R data model was enhanced later to include generalization and specialization.

Major Components

The components of the E-R model are practically the same as the object-based data model. Both being techniques for building a semantic data model, you would almost expect the building blocks to be the same. Therefore, as we describe the components here or below in the chapter, you will note the special features of the E-R model. Over the years, more precise notations have been included to indicate special aspects of real-world information requirements.

Figure 7-2 presents the graphical notations or symbols for the various components.

We mentioned above that E-R data model provides representations for different variations within a component. For example, real-world information contains variations in attributes. Sometimes, an attribute may be a composite of other attributes. Mostly, an attribute has a single value for one occurrence of an entity. However, you also come across an attribute having multiple values for one occurrence of an

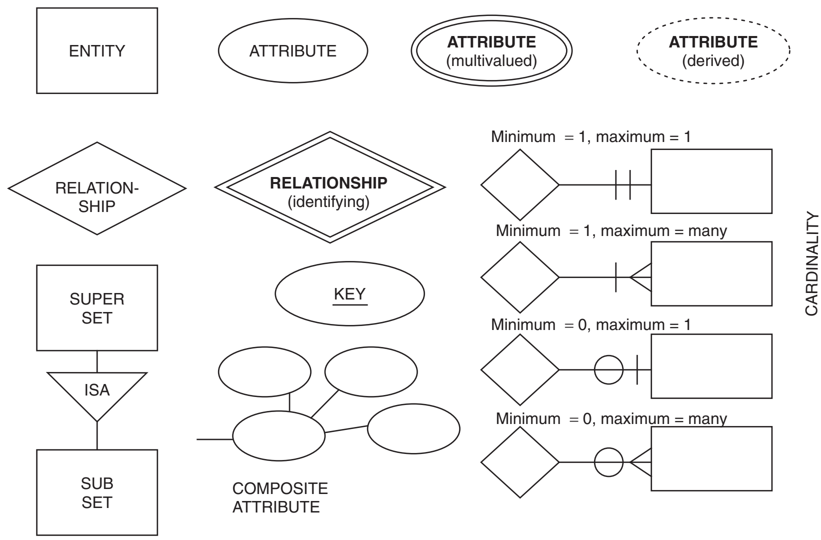


Figure 7-2 E-R data model: graphical notations or symbols.

entity. Note the different notations in the figure to denote single-valued, multivalued, derived, and composite attributes. Also, observe the notations for the two types of relationship. You will also note that symbols for supersets, subsets, and cardinality indicators are different from the ones used in the object-based data model covered in Chapter 6.

ENTITIES

Consider a banking institution. For this organization, real-world things exist that are of interest. The bank is interested in customers; it renders banking services to each customer. The real-world thing called CUSTOMER is of interest to the bank. Customers open bank accounts. These bank accounts may be checking, savings, or loan accounts. As such, real-world things such as CHECKING ACCOUNT, SAVINGS ACCOUNT, and LOAN ACCOUNT are also of interest to the organization. If you examine the bank's operations, you will notice other real-world things that are of interest to the bank.

In creating an E-R data model, you start by observing real-world things that are of interest to the organization in the course of its operations. You are not really examining individual processes; you are just listing the things, both tangible and intangible. These real-world things are the entities that need to be included in the data model. This is similar to recognizing business objects and including them in an object-based data model.

While studying entities in the E-R data model, observe the similarities with the object-based data model. Observe how entity types and entities match up with object sets and object instances.

Entity Definition

Entities are real-world "things" of interest to an organization. Note the following businesses and a few of the things of interest to the particular organizations:

Airline: AIRCRAFT, AIRCRAFT TYPE, MANUFACTURER, PASSENGER, FLIGHT

Pharmacist: CUSTOMER, PRODUCT, PRESCRIPTION, SUPPLIER, BILL

Car Wash: WASH TYPE, WASH, EQUIPMENT

Farm Market: PRODUCE, SUPPLIER, SALE

Orchard: TREE, VARIETY, SPECIES, ORCHARD, CUSTOMER

Observe the following features of an entity:

Independent existence. Notice that in each case, an entity in the real world has independent existence. That means an entity does not depend on the presence of another entity for its existence. The entity PASSENGER exists irrespective of whether it is related to another entity called FLIGHT. There could potentially be prospective passengers who have not been associated with any flights at all. Still, the entity exists in the database system.

Distinguishable. In the real world, one entity is distinguishable from another. That means that one passenger named Tom James is distinguishable from another passenger named Joe Rawlins. Each entity is unique, and you can differentiate one entity from another.

Physical and Conceptual Entities. When you review the examples of entities listed above, you observe that some of the things may be seen or touched—things that are tangible. When you examine the others on the list, you realize that some of the other things cannot be really seen or experienced. These are not tangible. Nevertheless, both categories are things that are of interest to an organization. Make sure you include both types as indicated below in your data model.

Things with physical existence. An employee, a specific automobile, a particular building, an explicit machine, and a particular aircraft are examples of this type.

Things with conceptual existence. Examples include a university course, a visit to the doctor, an invoice sent to a customer, a bank account, and a type of aircraft.

Entity Types

So far in our discussion we have indicated one particular thing of interest to an organization as an entity. An entity is a single thing. One customer is a "thing" that an organization is interested in. Of course, there would be thousands of customers for an organization. So database practitioners make a distinction and refer to the notion that refers to the collection of entities as an entity type. Entities are individual "things," whereas entity types refer to the group of entities.

Figure 7-3 illustrates the two notions of entity types and entities.

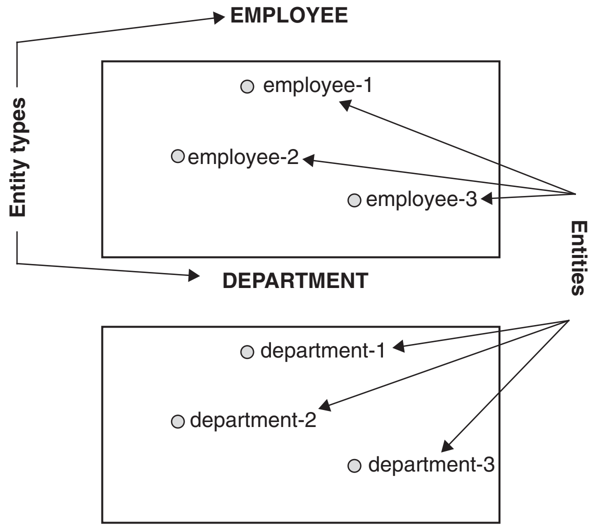


Figure 7-3 Entity types and entities.

Note the two examples shown in the figure:

Entity Types—EMPLOYEE, DEPARTMENT
Entity—Individual occurrences of each entity type

Entity Sets

You now understand the meaning of the two terms "entity" and "entity type." Take the example of the entity type EMPLOYEE. Assume that there are 5000 employees in the database at a given time. This is the set of employee entities in the database at that time. "Entity set" refers to the entire group of entities of an entity type in the database at a given time.

Figure 7-4 explains the concept of entity set.

Note the sets of employee and department entities shown in the figure. In practice, the name of the entity type also refers to the entity set. See how the entity type EMPLOYEE refers to the set of employee entities. A rectangular box denotes an entity type in the E-R data model.

Weak Entity Types

Consider the real-world situation of customers placing orders and orders containing one or more detail lines. Each detail line in an order contains the data about a product ordered. For this small set of real-world information requirements you can come up with three entity types as shown in Figure 7-5.

Now think about these three entity types, especially about the existence of entities of these types. An entity of CUSTOMER can exist whether a corresponding entity of ORDER exists or not. Each customer entity is uniquely identified inde-

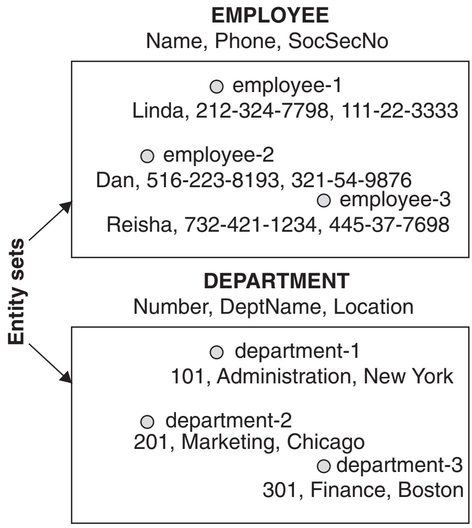


Figure 7-4 Entity sets.

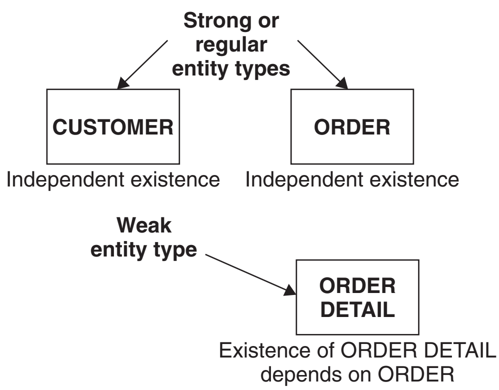


Figure 7-5 Strong and weak entity types.

pendently. Similarly, each order entity is uniquely identified by an order number and therefore, can exist without dependence on entities of other entity types.

Contrast this with the case of the entity type ORDER DETAIL. An order detail is not uniquely identifiable unless it is related to the corresponding order to which it is an order detail. That is, an order detail entity has no independent existence apart from an order entity. The ORDER DETAIL entity type depends on the ORDER entity type for its existence.

ORDER DETAIL is called a weak entity type. CUSTOMER and ORDER are known as regular or strong entity types. A weak entity type depends on another regular entity type for its existence. Weak entities depend on the corresponding strong entities for identification. We will review weak entities further in later sections.

ATTRIBUTES

In the discussion on entities and entity types, you must have observed that a group of entities were put together and called an entity type because these entities are similar. Customer entities are grouped together as the entity type CUSTOMER for the reason that the entities share common characteristics. If one customer entity possesses the characteristics of name, address, and phone, then another customer entity also has these characteristics; yet another customer entity also shares these characteristics. Common characteristics or attributes determine whether a number of entities or “things” may be grouped together as one type.

The E-R data model represents inherent characteristics of entity types as attributes in the same way as the object-based data model indicates properties of object sets. Therefore, our discussion of attributes can be brief, just highlighting the representation and serving as a refresher.

Attribute Specification

Inherent characteristics or basic properties of an entity type are defined as attributes for that entity type. What is the function of attributes? Attributes describe an entity type. Consider the attributes for the entity type CUSTOMER shown in Figure 7-6.

The figure presents four attributes for CUSTOMER. Look at the following values of the attributes for one customer or a single entity:

CustomerName:	John A. Harreld
CustomerAddress:	2401 Merry Boulevard, Branchburg, NJ 08810
CustomerPhone:	(908) 722-8419
CreditCode:	AAB

The values indicate the particular customer entity you are interested in. The values describe the specific customer and distinguish that customer from others.

Similar to the attributes represented in an object-based data model, the attributes in an E-R data model also possess the following features:

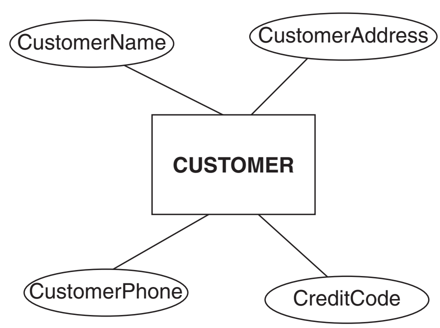


Figure 7-6 Attributes for CUSTOMER entity type.

Unique values. Each entity within an entity type has a unique set of values for the attributes. The values for CustomerName, CustomerAddress, CustomerPhone, and CreditCode for John A. Harreld form a distinct set that pertains to this customer entity. This, however, does not mean that a value for any attribute is not repeated for other customers. Other customers may also have the value "AAB" for CreditCode.

Changeable values. As mentioned for the object-based data model, the values of a particular customer entity may change over time. John A. Harreld may change his phone number to (908) 887-6123.

Null values. An attribute may have a null value when the actual value is unknown, unavailable, or missing.

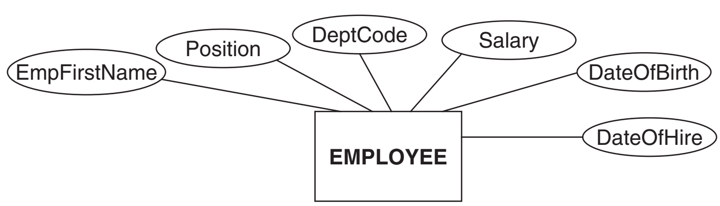
Values and Domains

As you already know, each attribute of the various entities of an entity type gets its values from a set of allowable values. This set of legal values forms the domain of values for the attribute. Figure 7-7 displays examples of attributes and respective domains for the entity type EMPLOYEE.

Note how the domain for each attribute consists of a set of allowable values. Notice how the two attributes DateOfBirth and DateOfHire share the same domain. A domain of values may apply to several attributes of an entity type. However, each attribute takes its values from a single domain.

Attribute Types

In most cases, a particular attribute of a single entity has only one value at any given time. For example, the attribute ProjectDuration for a single entity of the entity type



ATTRIBUTE	VALUE	DOMAIN
EmpFirstName	Susan	Character: size 25
Position	Manager	Character: size 30 (Manager, Programmer, Analyst, etc.)
DeptCode	D62	Character: size 3, range D01–D99
Salary	95000	Numeric: 6 digits, range 0-995000
DateOfBirth	01JAN1955	Valid date, range 01JAN1920–
DateOfHire	16SEP1986	Valid date, range 01JAN1920–

Figure 7-7 Attributes and domains for EMPLOYEE entity type.

PROJECT has a value of 90 days. At any given time, this is the only value for that attribute. However, in real-world situations, you will come across attributes that may have more than one value at the same time. You will also notice other types of variations in attributes.

In this subsection, we describe such variations in the types of attributes. As you know, a semantic data model such as the E-R data model must reflect real-world information correctly. The E-R model provides for representation of different attribute types. Let us go over a few examples.

Single-Valued and Multivalued Attributes Note the following examples of single-valued and multivalued attributes. Observe the values of attributes for a single entity. These are the values at a given point in time.

Single-valued

Entity type:	EMPLOYEE
Attribute:	EmployeeJob
Attribute value for single entity:	Salesperson
Entity type:	EMPLOYEE
Attribute:	EmployeeDOB
Attribute value for single entity:	24JAN1975

Multivalued

Entity type:	AUTOMOBILE
Attribute:	ExteriorColor
Attribute values for single entity:	Beige, Gold (two-tone color)
Entity type:	CUSTOMER
Attribute:	CustomerPhone
Attribute values for single entity:	732-888-1234, 732-888-3456, 732-889-5566

Figure 7-8 illustrates how single-valued and multivalued attributes are represented in a data model diagram with different notations for the two types.

Simple and Composite Attributes This is another variation in attribute types. In real-world information, you will notice that some attributes may be divided further into smaller units. The smaller units are known as simple or atomic attributes, whereas the larger units are called composite attributes. Most of the attributes in real-world information, however, are simple attributes. Your data model has to represent these variations. Note the examples presented below.

Composite

Entity type:	CUSTOMER
Composite attribute:	CustomerAddress
Component simple attributes:	Street, City, State, Zip

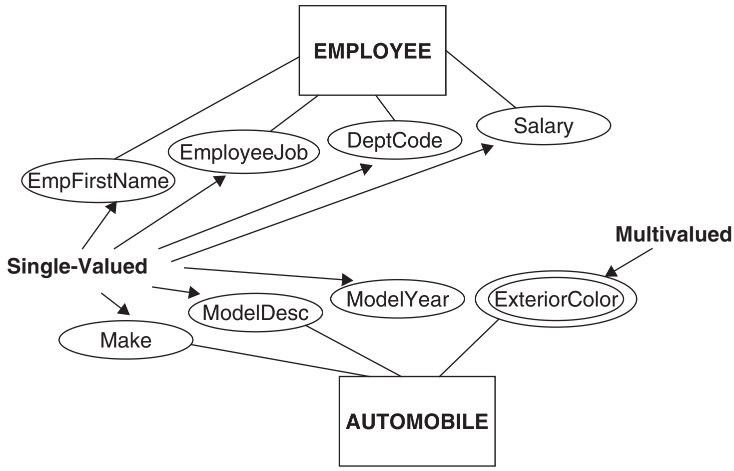


Figure 7-8 Single-valued and multivalued attributes.

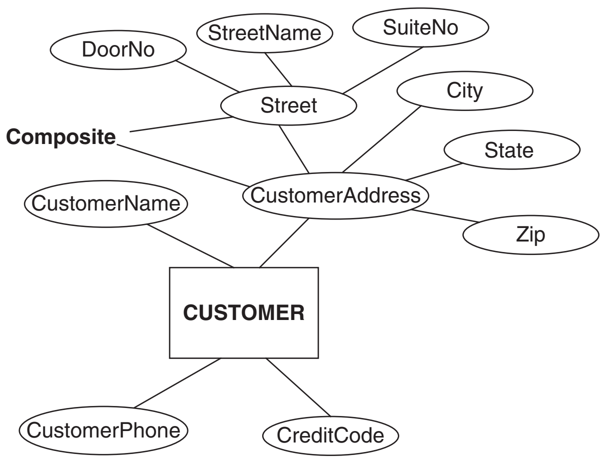


Figure 7-9 Composite attribute.

Entity type:	EMPLOYEE
Composite attribute:	EmployeeName
Component simple attributes:	FirstName, LastName, MidInitial

Simple

Entity type:	LOAN ACCOUNT
Simple attribute:	LoanAmount
Entity type:	EMPLOYEE
Simple attribute:	SocialSecNumber

Figure 7-9 shows the notations used for representing composite attributes. Note how the composite attribute branches into the simple attributes that are its components. Especially observe how the attribute Street may be broken down further into simple attributes.

Attributes with Stored and Derived Values In later phases of database development, you will transform the data model into the physical structure of how data gets stored in the database. From our discussion of attributes, you must have realized that the physical data stored in the database consist of values of the attributes of the complete set of all the entities. Stored data are really values of the attributes. If you have a CUSTOMER entity type with CustomerName, CustomerAddress, and Phone as the attributes, then your database stores the values of these attributes for all the customer entities in your organization. These are attributes whose values are stored in the database.

Sometimes you would want to calculate and derive values from the values of one or more attributes and to store the derived values in separate attributes. These are attributes containing derived values. Look at the following examples.

Attributes with derived values

Entity type:	EMPLOYEE
Derived attribute:	LengthOfEmployee
Attribute derived from:	EmployeeStartDate (and today's date)
Entity type:	PRODUCT
Derived attribute:	ProfitMargin
Attributes derived from:	UnitPrice and UnitCost

Figure 7-10 illustrates how a derived attribute is represented in an E-R data model.

Candidate and Primary Keys

In the discussion of object sets and object instances of object-based data model, we studied how certain attributes of an object set may be used to uniquely identify the instances. These were called identifiers. In the same manner, entity-relationship model has provisions to represent key attributes. All the points discussed on keys in Chapter 6 apply to the E-R model as well. You also know that underlining the

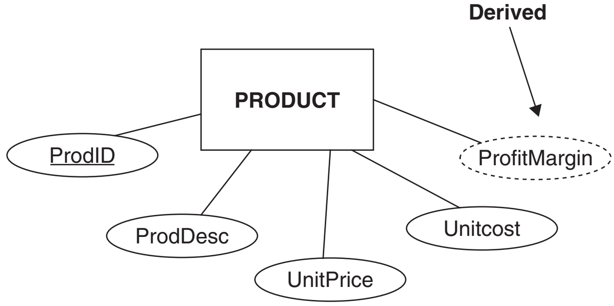


Figure 7-10 Derived attribute.

attribute name in a data model diagram indicates that the attribute is the primary key. Here is a brief summary of the main points:

Candidate key. One or more attributes whose values uniquely identify individual entities.

Primary key. One of the candidate keys selected to serve as the unique identifier for the entity type.

Composite key. A candidate key that consists of two or more attributes.

Considerations for Choosing Primary Key

Attribute length. Choose the shortest of the candidate keys.

Number of attributes. Prefer a key with a single attribute to a composite key.

Certainty of uniqueness. From the candidate keys, select the one that is more likely to contain unique values during the life of the database system.

Built-in meanings. Avoid any candidate key where values of parts of the attribute have built-in meanings, e.g., the attribute ProductNumber, where the values of part number indicate the production batch in the manufacturing plant.

RELATIONSHIPS

Let us reconsider information requirements for a university. Note the obvious fact that students register for courses. You observe that a particular student registers for several courses. Student Karolyn Stone registers for the courses Database Design, Data Warehousing, and Web Page Design. This is a relationship of that particular student Karolyn Stone with the three courses.

On the flip side of the relationship, you also note that many students register for one particular course. You find that for the course on Database Design, other students such as Samuel Wang, Tabitha Peters, and Monika Gonzales have also registered. So you notice the relationship between the course Database Design and the four students.

We have been through relationships and their features in sufficient detail while discussing the object-based data model. You know how instances of one object set may have relationships with instances of one or more object sets. Now let us highlight some key points in the way the E-R data model deals with relationships. Note the slight differences in the notations used in the E-R data model.

Association Between Entities

Go back to the above example of students and course. Let us review the example in the context of the E-R data model and use the terms that apply to this model.

Entity type:	STUDENT
Entities:	Karolyn Stone, Samuel Wang, Tabitha Peters, and Monika Gonzales
Entity type:	COURSE
Entities:	Database Design, Data Warehousing, Web Page Design
Associations:	Karolyn Stone with Database Design Karolyn Stone with Data Warehousing Karolyn Stone with Web Page Design
Associations:	Database Design with Karolyn Stone Database Design with Samuel Wang Database Design with Tabitha Peters Database Design with Monika Gonzales

Note the above associations and express these in E-R model terminology. Carefully examine the two sets of associations. What is the observation? In the first set of associations, an individual entity Karolyn Stone of entity type STUDENT associates with other individual entities Database Design, Data Warehousing, and Web Page Design of the other entity type, COURSE. If entities of one entity type associate with entities of another entity type, you express these associations in a E-R data model diagram by joining the rectangular boxes representing the two entity types with a relationship line. Although the two entity types are linked together to express the relationship, remember that the associations are actually between individual entities of these two entity types.

Figure 7-11 indicates how the above relationship is represented in an E-R diagram. Look at the associations between individual entities illustrated in the figure. As you know from our study of object-based data modeling, this is a many-to-many relationship. Observe the notation in the figure to denote a many-to-many relationship. This notation has acquired the name "crow's feet." Compare this notation with the notation used in object-based data model for a many-to-many relationship. Usage of a specific notation is just a matter of convention and acceptance. In practice, you will come across different sets of notations in common usage. Appendix C covers diagramming conventions and symbols.

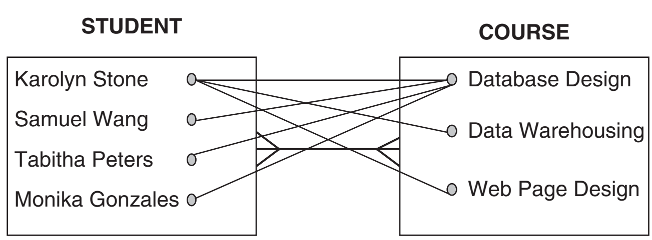


Figure 7-11 Relationship between entity types.

Degree of a Relationship

The degree of a relationship in the E-R data model is defined in the same way as in object-based data model. The degree of a relationship refers to the number of entity types that participate in the relationship. A three-way relationship is a ternary relationship, two-way a binary relationship, and one-way a unary relationship. A unary relationship is also known as a recursive relationship, because the entities of the same entity type associate with one another.

Figure 7-12 illustrates the concept of relationship degrees. It presents relationships whose degrees are three, two, and one. You may visualize relationships with higher degrees, although they are not common in real-world information requirements. Binary relationships are the most common relationships in the real world.

Cardinality in Relationships

It may be worthwhile to refer back to our detailed discussion of cardinality in relationship expressed in the object-based data model. The discussion covered one-to-one, one-to-many, and many-to-many relationships. Also, the notions of maximum and minimum cardinalities should be reviewed. The same principles and concepts apply to the E-R data model as well. We do not propose to repeat the discussions here. However, let us note the notations used for representing these two relationship types in the E-R data model.

Figure 7-13 shows the notations for representing these relationship types. Note the minimum and maximum cardinality indicators placed near each entity type.

Optional and Mandatory Conditions

Recall how the minimum cardinality indicator expresses the optional or mandatory nature of a relationship. Placing a “0” as the minimum cardinality indicator near an

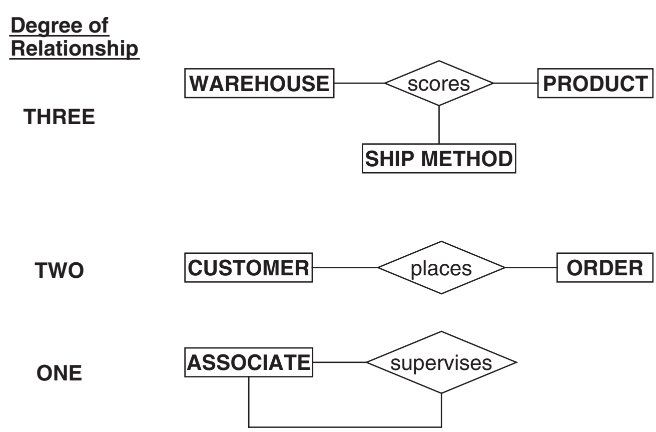


Figure 7-12 Degrees of relationships.

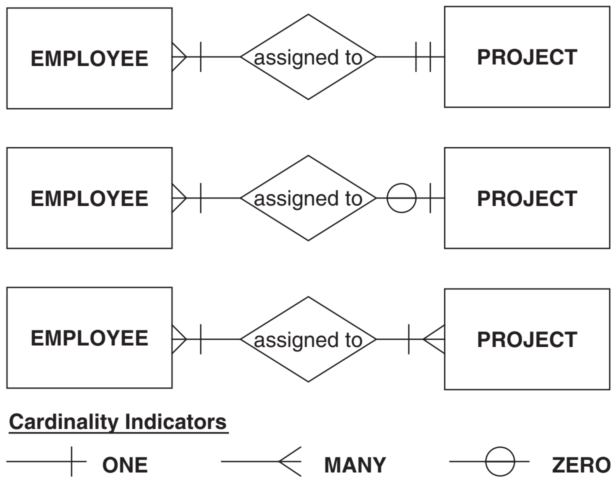


Figure 7-13 Cardinality indicators.

entity type indicates that some of the entities of that entity type may not participate in the relationship. That means that the relationship is optional for that entity type. Let us explore the notion of optional and mandatory nature of relationships further.

Consider a real-world situation of employees working in departments. Normally, every employee is assigned to a department and every department will have employees. But this may not always be true. Newer employees may not be assigned to a department yet; some employees on special assignments may not be part of the conventional departments. On the other hand, some special departments may have been created for costing raw materials and equipment with no human resources being part of the departments. In this case, such departments do not have employees associated with them. Your data model must be able to represent such exceptional conditions found in real-world information requirements. Follow along to learn how these conditions are represented in the data model.

Four cases arise based on the exception conditions. Figure 7-14 illustrates the first two cases and Figure 7-15 the next two cases. Note the minimum and maximum cardinality indicators shown in each case. Also, note each dotted line indicating that the relationship on that side is optional.

Let us review these four cases in some detail. As you will observe, the minimum cardinality indicator denotes the optional or mandatory nature of the relationship—whether it is partial participation or full participation in the relationship by the specific entity type. Pay special attention to the figure while we discuss the four cases in detail.

Case 1

A department must have at least one employee, but it may have many employees. Note minimum cardinality indicator 1 and maximum cardinality indicator * near EMPLOYEE entity type.

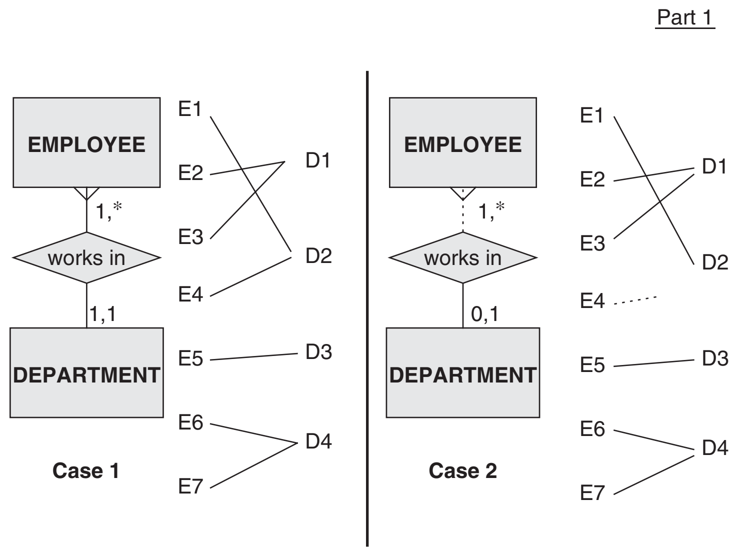


Figure 7-14 Relationship: optional and mandatory conditions—Part 1.

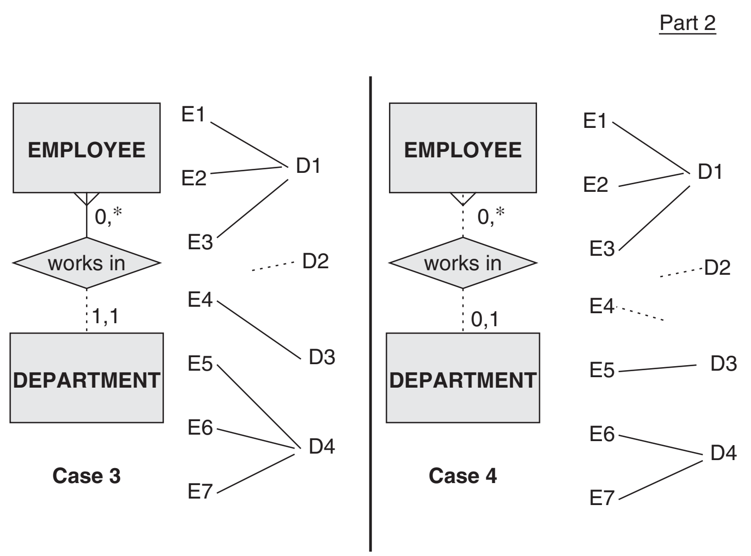


Figure 7-15 Relationship: optional and mandatory conditions—Part 2.

Meaning of minimum cardinality indicator 1 Each department entity is associated with at least one employee entity. That is, the minimum number of employee instances associated with one department entity is 1. Note the solid relationship line next to DEPARTMENT indicating that the relationship on this side is mandatory and that every department entity participates in the relationship.

Meaning of maximum cardinality indicator * A department entity may be associated with many instances of employee.

An employee must be assigned to at least one department, and he or she can be assigned to only one department. Note minimum cardinality indicator 1 and maximum cardinality indicator 1 near DEPARTMENT entity type.

Meaning of minimum cardinality indicator 1 Each employee entity is associated with at least one department entity. That is, the minimum number of department instances associated with one employee entity is 1. Notice the solid relationship line next to EMPLOYEE indicating that the relationship on this side is mandatory and that every employee entity participates in the relationship.

Meaning of maximum cardinality indicator 1 An employee entity can be associated with one instance of department at most.

Case 2

A department must have at least one employee, but it may have many employees. Note minimum cardinality indicator 1 and maximum cardinality indicator * near EMPLOYEE entity type.

Meaning of minimum cardinality indicator 1 Each department entity is associated with at least one employee entity. That is, the minimum number of employee instances associated with one department entity is 1. Notice the solid relationship line next to DEPARTMENT indicating that the relationship on this side is mandatory and that every department entity participates in the relationship.

Meaning of maximum cardinality indicator * A department entity may be associated with many employee entities.

Every employee may not be assigned to a department; if assigned, he or she can be assigned to only one department. Note minimum cardinality indicator 0 and maximum cardinality indicator 1 near DEPARTMENT entity type.

Meaning of minimum cardinality indicator 0 Some employee entities may not be associated with any department entities. Not every employee entity is associated with a department entity. That is, the minimum number of department instances associated with one employee entity is 0. Not every employee entity participates in the relationship. Notice the dotted or broken relationship line next to EMPLOYEE entity type, indicating that the relationship on this side is optional and that not every employee entity participates in the relationship. A broken or dotted line denotes partial participation in the relationship.

Meaning of maximum cardinality indicator 1 An employee entity can be associated with one instance of department at most.

Case 3

Every department may not have employees; if a department has employees, it can have many employees. Note minimum cardinality indicator 0 and maximum cardinality indicator * near EMPLOYEE entity type.

Meaning of minimum cardinality indicator 0 Some department entities may not be associated with any employee entities. Not every department entity is associated with employee entities. That is, the minimum number of employee instances associated with one department entity is 0. Not every department entity participates in the relationship. Note the dotted or broken relationship line next to the DEPARTMENT entity type, indicating that the relationship on this side is optional and that not every department entity participates in the relationship. A broken or dotted line denotes partial participation in the relationship.

*Meaning of maximum cardinality indicator ** A department entity may be associated with many employee entities.

An employee must be assigned to at least one department, and he or she can be assigned to only one department. Note minimum cardinality indicator 1 and maximum cardinality indicator 1 near DEPARTMENT entity type.

Meaning of minimum cardinality indicator 1 Each employee entity is associated with at least one department entity. That is, the minimum number of department instances associated with one employee entity is 1. Note the solid relationship line next to EMPLOYEE, indicating that the relationship on this side is mandatory and that every employee entity participates in the relationship.

Meaning of maximum cardinality indicator 1 An employee entity can be associated with one instance of department at most.

Case 4

Every department may not have employees; if a department has employees, it can have many employees. Note minimum cardinality indicator 0 and maximum cardinality indicator * near EMPLOYEE entity type.

Meaning of minimum cardinality indicator 0 Some department entities may not be associated with any employee entities. Not every department entity is associated with employee entities. That is, the minimum number of employee instances associated with one department entity is 0. Not every department entity participates in the relationship. Note the dotted or broken relationship line next to DEPARTMENT entity type, indicating that the relationship on this side is optional and that not every department entity participates in the relationship. A broken or dotted line denotes partial participation in the relationship.

Every employee may not be assigned to a department; if assigned, he or she can be assigned to only one department. Note minimum cardinality indicator 0 and maximum cardinality indicator 1 near DEPARTMENT entity type.

Meaning of minimum cardinality indicator 0 Some employee entities may not be associated with any department entities. Not every employee entity is associated with a department entity. That is, the minimum number of department instances associated with one employee entity is 0. Not every employee entity participates in the relationship. Note the dotted or broken relationship line next to EMPLOYEE entity type indicating that the relationship on this side is optional and that not every employee entity participates in the relationship. Broken or dotted line denotes partial participation in the relationship.

Meaning of maximum cardinality indicator 1 An employee entity can be associated with one instance of department at most.

SPECIAL CASES

In our discussion on object-based data modeling, we have already covered supersets and subsets. The concept of generalization and specialization was added to the E-R model at a later date. The enhanced entity-relationship (EER) model includes the concept of superentity and subentity types.

We will now devote some time to special considerations of the E-R data model. We have already touched on weak entity types; we will now elaborate on this. What about the concept of aggregate object sets in the object-based data model? Is there anything comparable in the E-R data model? What is a gerund? What is the place for a gerund in data modeling? Let us discuss this and other such special cases.

Modeling Time-Dependent Components

Think of the values of attributes stored in the database. Normally, the values stored are the current values. What values do you find for the Address attribute of the CUSTOMER entity type? The current address of each customer. As a customer's address changes, the new address replaces the old address in the database. In the same way, what values will be stored in the ProductUnitCost and ProductUnitPrice attributes of the PRODUCT entity type?

Assume that just the current costs and current prices are stored in the database for the products. What happens if an old invoice to a customer has to be recalculated and reprinted? What happens when you want to calculate the profit margin for the past quarter using the unit cost and unit price? If there had been no intervening price or cost changes, you can perform these tasks without any problems. What you notice is that ProductUnitCost and ProductUnitPrice are time-dependent components and that your data model must include historical entity types to keep track of the changes.

Figure 7-16 indicates how historical entity types are included in the data model. Note why ProductNo and EffectiveDate are used together to form the primary key for the historical entity types.

Identifying and Nonidentifying Relationships

Refer back to Figure 7-5 describing a weak or dependent entity type. Let us now explain the relationship among the entity types shown in that figure. ORDER DETAIL was shown to be a weak entity type depending on ORDER entity type

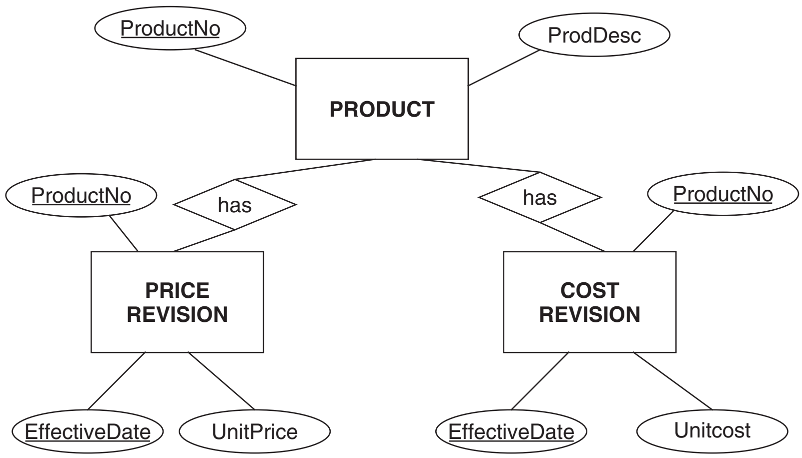


Figure 7-16 Time-dependent model components.

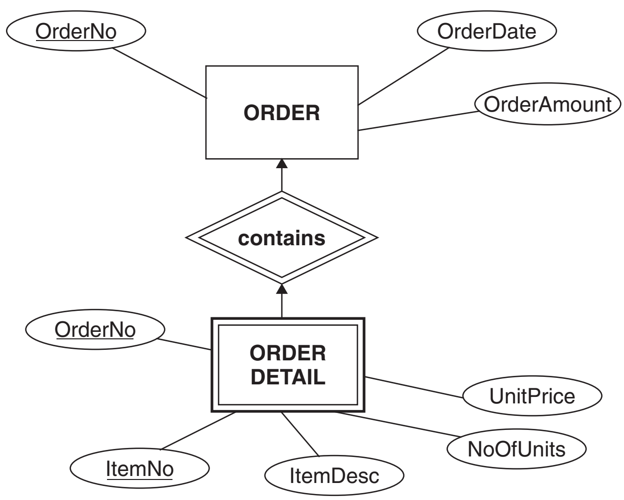


Figure 7-17 Identifying relationship.

for its existence. This implies that you cannot have an order detail entity unless a corresponding order entity exists. Without an order entity an order detail entity cannot exist in the database. It is meaningless to think of stray and loose order detail entities existing in the database. Figure 7-17 shows the relationships between two entity types presented in Figure 7-5. Notice the notations used in the diagram to distinguish the components.

Note the following points about the two entity types:

Weak entity type. An entity type that is existence-dependent on another entity type. ORDER DETAIL is a weak entity type. It depends on ORDER entity.

Identifying entity type. A weak entity is identified by being related to another entity type called the identifying or owner entity type. ORDER is the identifying entity type.

Primary key. A weak entity does not have a primary key by itself. Usually, weak entities are identified by partial identifiers or discriminators just to distinguish them within the owner entity. For ORDER DETAIL, ItemNo is a partial identifier. The full primary key of a weak entity type usually consists of the concatenation of the primary key of the owner entity type with the partial identifier of the weak entity type. Note the primary key for ORDER DETAIL—OrderNo and ItemNo concatenated together.

Identifying relationship. The relationship between a weak entity type and its identifying or owner entity type is known as an identifying relationship. Observe the relationship notation shown between ORDER DETAIL and ORDER.

Attributes of Relationship Types

In the discussion of the object-based data model, we studied aggregate object sets. Recall that relationships between objects sets give rise to aggregate object sets. Such aggregate object sets behave in the same way as regular object sets. An aggregate object set may have attributes and relationships with other regular or aggregate object sets.

This notion of aggregate object sets is also found in the E-R data model. A relationship object type in E-R modeling corresponds to an aggregate object set in object-based modeling. Figure 7-18 shows an example of relationship object type. Note the attributes for the relationship object type WORKSON shown in the diagram.

When to Use a Gerund

What is a gerund? The relationship entity type described above is sometimes called a gerund. In English grammar, the word “writing” is a gerund; it is derived from the verb “to write” but behaves like a noun. Similarly, if a relationship behaves like an entity type, that relationship may be termed as a gerund.

Consider a three-way many-to-many relationship among three entity types: CUSTOMER, PRODUCT, and WAREHOUSE. This relationship arises out of the real-world situation of shipping products to customers from different supply warehouses. A diamond symbol normally represents this relationship. However, in reality, the relationship appears to be an entity type. It is a gerund. When do you, as a data modeler, represent a relationship as a gerund? You need to represent a relationship as an entity type if the relationship truly has specific attributes or the relationship, by itself, has relationships with other entity types.

Figure 7-19 illustrates the gerund SHIPMENT, based on the relationships among CUSTOMER, PRODUCT, and WAREHOUSE.

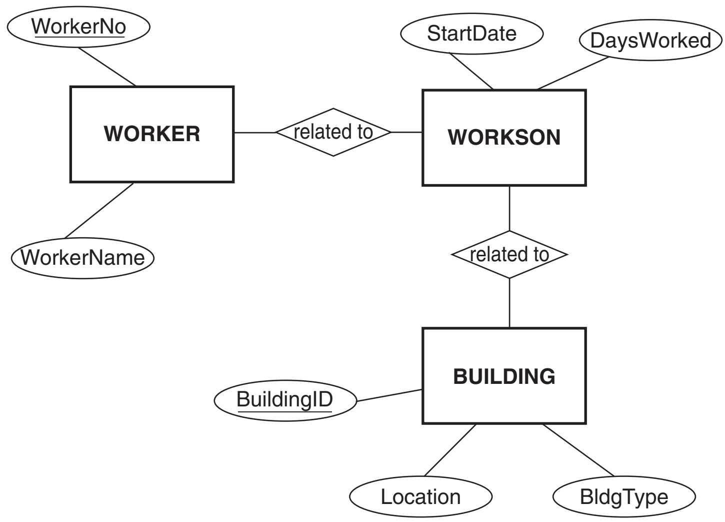


Figure 7-18 Relationship entity type.

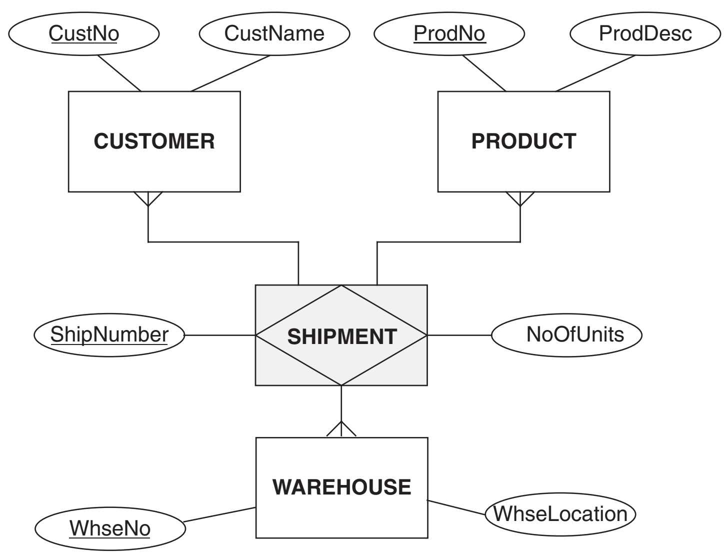


Figure 7-19 Representing a gerund in E-R model.

Generalization and Specialization

All the points covered in our discussion of generalization and specialization in the object-based data model apply to the E-R data model. Instead of repeating the discussion here, it will be worthwhile to refer back to Chapter 6.

We just want to highlight the difference in the notations. Figure 7-20 contains a partial E-R data model showing an example of generalization and specialization. Note the notations used according to E-R data model conventions.

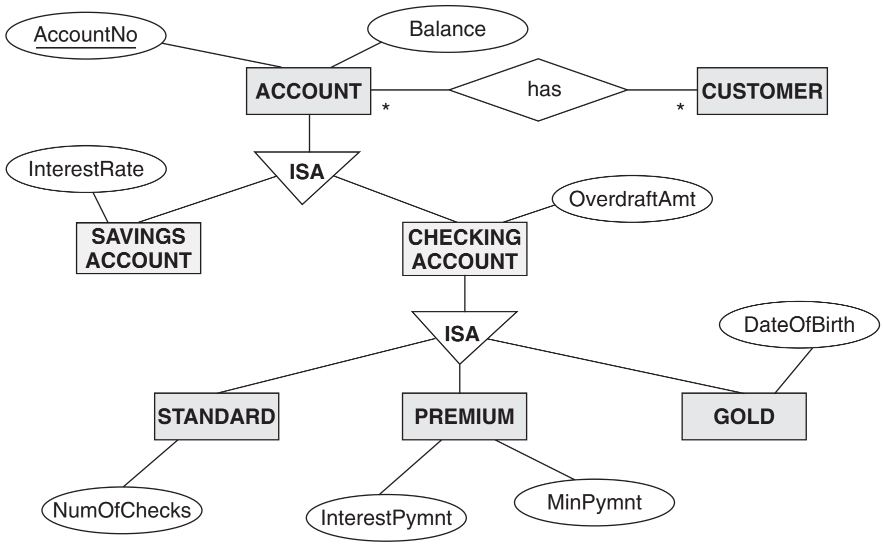


Figure 7-20 E-R model: generalization and specialization.

THE ENTITY-RELATIONSHIP DIAGRAM

This section pulls together everything we have covered on E-R data modeling and summarizes the topic with a comprehensive entity relationship diagram (ERD). Before we review the diagram and note the components, let us revisit the components and their notations. As mentioned above, in practice, you will observe some minor variations in the notations used by different data modelers. Which set will be your standard set? You would want to conform to the notations adopted by a CASE tool you are required to use in your database project.

Review of Components and Notations

Figure 7-21 highlights many of the standard components and notations used in entity-relationship modeling. Carefully review the notations or symbols.

The following is a recapitulation of components:

Entity Type
- Strong
- Weak

Attribute
- Simple, single-valued
- Composite
- Multivalued
- Derived

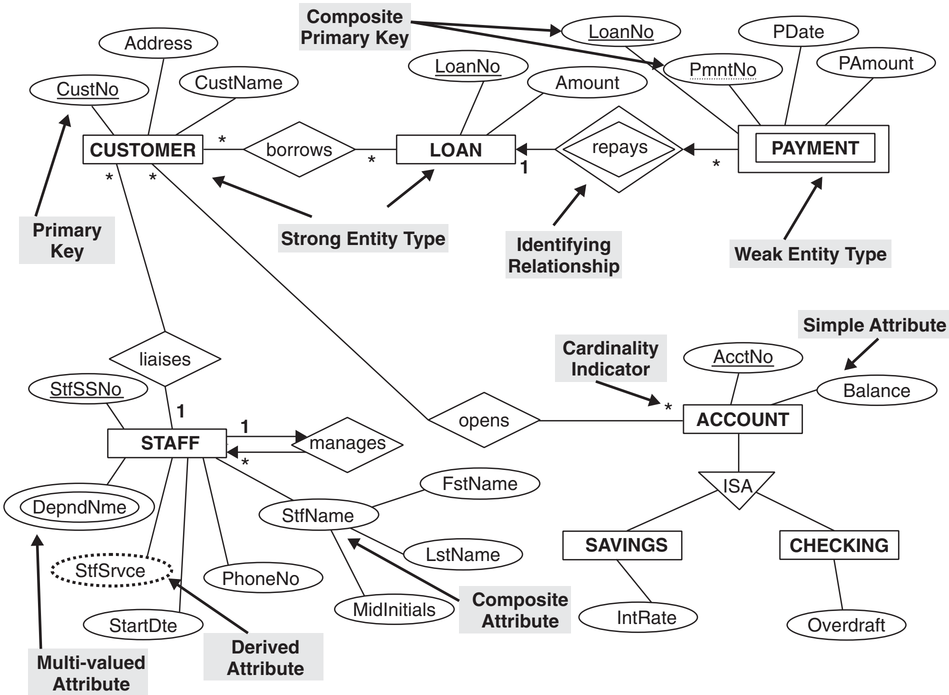


Figure 7-21 E-R data model: recapitulation of components and notations.

Primary Key
- Single attribute
- Composite

Relationship
- Names
- Cardinality indicators
  - Maximum
  - Minimum
- Identifying and nonidentifying
- Optional and mandatory conditions
- Gerund

Sample ERD

Let us bring all the principles and concepts of E-R data modeling together and create a data model for the information requirements for a university. Assume that the data model has to represent only the teaching aspects of the university. Ignore other information relating to receivables, payables, salaries, administration, employees, and so on.

Consider the following requirements to be included in the model:

1. Each course offered may be taught in different classes. For example, the Database Design course may have a Tuesday night class and a Thursday night class.
2. A faculty member may teach one or more classes.
3. Different faculty members may be assigned to teach different classes of the same course.
4. Each faculty member decides on the number of exams for the class he or she teaches. Student scores are assigned for each type of examination or test.
5. A specific textbook is assigned to a course.
6. A student may be enrolled in one or more courses.
7. Each class meets in the same room during a semester.

Figure 7-22 presents an ERD for the given information requirements. Note that the diamond-shaped relationship notation is not shown in this ERD. This is an acceptable practice. Although the relationship notation is not explicitly shown, the implied relationships are expressed by the lines connecting entity type boxes.

Highlights of Sample ERD

Study the ERD carefully. Note all the entity types and observe the relationships. What do the cardinality indicators represent? Are there any composite attributes? Go back to the list of components and notations shown in Figure 7-21. How many of these can you identify in Figure 7-22?

Compare the information requirements with the ERD. Does the ERD reflect the information requirements adequately?

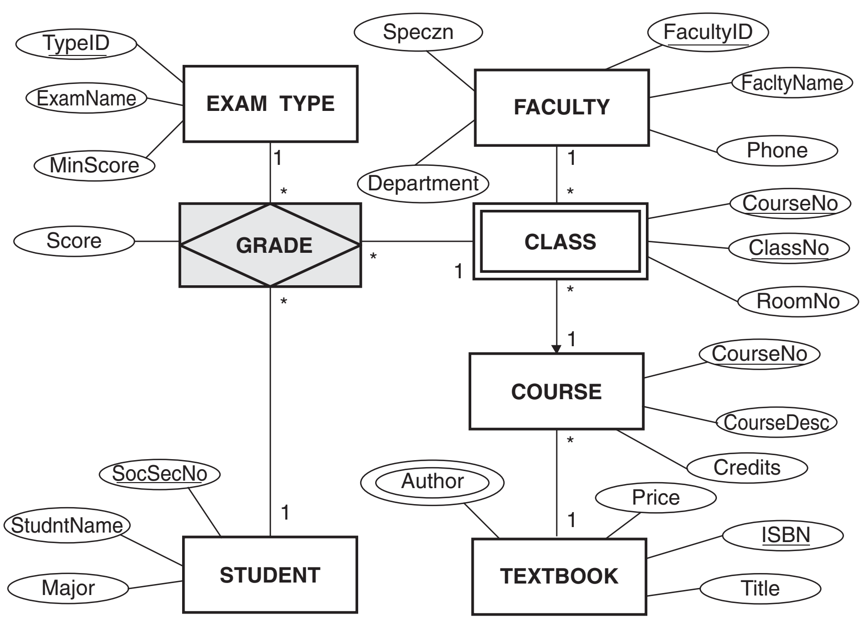


Figure 7-22 ERD: university teaching aspects.

Let us highlight certain points about the ERD to indicate how it matches with the stated information requirements.

- For each course, there may exist one or more classes. Note the one-to-many relationship between COURSE and CLASS.
- The relationship between COURSE and CLASS is an identifying relationship. CLASS depends on COURSE for its existence. CLASS is a weak entity type.
- Observe the one-to-many relationship between TEXTBOOK and COURSE. Although a specific textbook is assigned to a course, the relationship allows the possibility of the same textbook being used for more than one course.
- Why is the attribute *Author* shown as a multivalued attribute? This is to provide for multiple authors for the same textbook.
- The entity type EXAM TYPE includes examinations such as Mid-Term, Quizzes, Assignments, Laboratory Work, Final, and so on.
- The entity type GRADE represents the three-way relationship among CLASS, STUDENT, and EXAM TYPE.
- Although faculty members teach specific courses, the teaching assignments are for teaching particular classes of these courses. Note the direct relationship between FACULTY and CLASS.
- The attribute *Score* refers to the score for a given student, in a certain class, and for a particular examination type.

CHAPTER SUMMARY

- Like object-based modeling, entity-relationship modeling is another generic data modeling technique. E-R data modeling seems to be a natural way of looking at real-world information.
- Many data modelers prefer E-R data modeling because several CASE tools support it.
- Components or building blocks of E-R data model: entity type, attribute, primary key, relationship, cardinality indicators, generalization/specialization. The E-R data model provides additional notations to indicate variations in the components.
- An entity type represents all similar entities; on the flip side, an entity is an instance or occurrence of an entity type.
- Attribute types: single-valued and multivalued, simple and composite, derived.
- One of the candidate keys is chosen as the primary key for an entity type. A composite key consists of more than one attribute.
- Relationship between two entity types expresses the associations between entities of one entity type and entities of the second entity type. The cardinality indicators denote how many instances of one entity type may be associated with one instance of the other entity type.
- The minimum cardinality indicator represents optional or mandatory nature of a relationship.

- A weak entity type depends on another entity type for its existence.
- The notion of generalization and specialization in the E-R model is similar to the same concept in the object-based data model.

REVIEW QUESTIONS

1. "The E-R data modeling technique focuses on entities and relationships as fundamental model components." Explain.
2. Define the terms entity, entity type, and entity set.
3. What are physical and conceptual entities? Give two examples for each.
4. Describe attributes and value domains with three examples.
5. What is a composite attribute? Can a composite attribute itself be part of another attribute? If so, can you think of an example?
6. Describe the considerations for choosing the primary key for an entity type.
7. What is the degree of a relationship? Give two examples.
8. Give an example of a situation that needs modeling time-dependent components.
9. What is an identifying relationship? Explain briefly with an example.
10. What is a gerund? Why would you use a gerund in a data model? Give an example.

EXERCISES

1. Match the columns:

1. domain	A. three-way relationship
2. composite attribute	B. one or more simple attributes
3. relationship	C. time-dependent
4. ternary relationship	D. not absolutely essential in a model
5. minimum cardinality	E. no independent primary key
6. historical entity type	F. unary relationship
7. weak entity type	G. has distinct notation in E-R model
8. recursive relationship	H. association between entities
9. derived attribute	I. allowable values for attribute
10. primary key	J. indicates optionality of relationship

2. As a data modeler for a car dealership, identity the major entity types. List the attributes and primary keys for two of the entity types.
3. The health care coverage for employees in your company extends to employee dependents. Create a partial E-R data model to represent the health insurance coverage. (Hint: use identifying and nonidentifying relationships.)
4. Consider the relationship between the entity types WORKER and BUILDING. Assume variations in business rules with optional and manda-

tory conditions governing the relationship. Describe the different cases that might arise. In each case, show the minimum and maximum cardinality indicators for the relationship.

5. You are a senior data modeler for an airlines business. Consider all the major business processes. Create an E-R data model to represent the information requirements. Use proper notations to indicate the various components.