Chapter 3: Data Modeling Using the Entity- Relationship Model

3.1 The Building Blocks of an Entity-Relationship Diagram

ER diagram is a graphical modeling tool to standardize ER modeling. The modeling can be carried out with the help of pictorial representation of entities, attributes, and relationships. The basic building blocks of Entity- Relationship diagram are Entity, Attribute and Relationship.

3.1.1 Entity

An entity is an object that exists and is distinguishable from other objects. In other words, the entity can be uniquely identified.
The examples of entities are:

- A particular person, for example Dr. A.P.J. Abdul Kalam is an entity.
- A particular department, for example Electronics and Communication Engineering Department.
- A particular place, for example Coimbatore city can be an entity.

3.1.2 Entity Type

An entity type or entity set is a collection of similar entities. Some examples of entity types are:

- All students in KUD, say STUDENT.
- All courses in KUD, say COURSE.
- All departments in KUD, say DEPARTMENT.

An entity may belong to more than one entity type. For example, a staff working in a particular department can pursue higher education as part-time. Hence the same person is a LECTURER at one instance and STUDENT at another instance.

3.1.3 Relationship

A relationship is an association of entities where the association includes one entity from each participating entity type whereas relationship type is a meaningful association between entity types.
The examples of relationship types are:

- Teaches is the relationship type between LECTURER and STUDENT.
- Buying is the relationship between VENDOR and CUSTOMER.
- Treatment is the relationship between DOCTOR and PATIENT.

3.1.4 Attributes

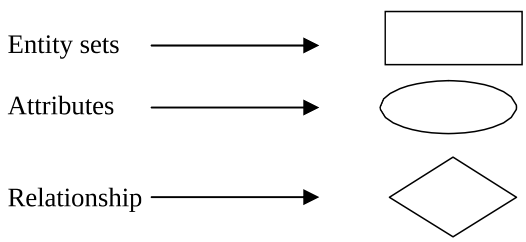
Attributes are properties of entity types. In other words, entities are described in a database by a set of attributes.
The following are example of attributes:

- Brand, cost, and weight are the attributes of CELLPHONE.
- Roll number, name, and grade are the attributes of STUDENT.
- Data bus width, address bus width, and clock speed are the attributes of MICROPROCESSOR.

3.1.5 ER Diagram

The ER diagram is used to represent database schema. In ER diagram:

- A rectangle represents an entity set.
- An ellipse represents an attribute.
- A diamond represents a relationship.
- Lines represent linking of attributes to entity sets and of entity sets to relationship sets.



Example of ER diagram

Let us consider a simple ER diagram as shown in Fig. 3.1.

In the ER diagram the two entities are STUDENT and CLASS. Two simple attributes which are associated with the STUDENT are Roll number and the name. The attributes associated with the entity CLASS are Subject Name and Hall Number. The relationship between the two entities STUDENT and CLASS is Attends.

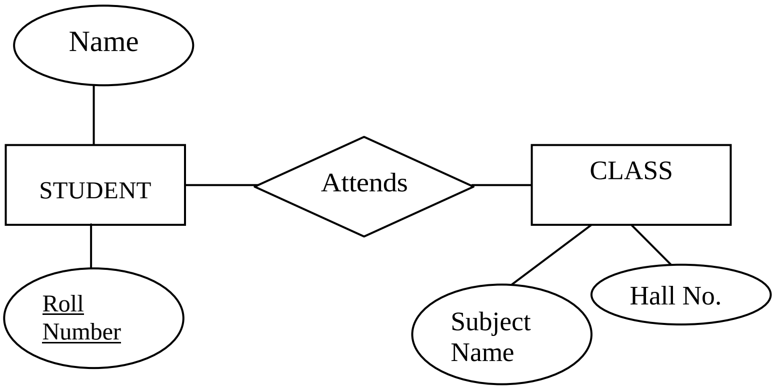
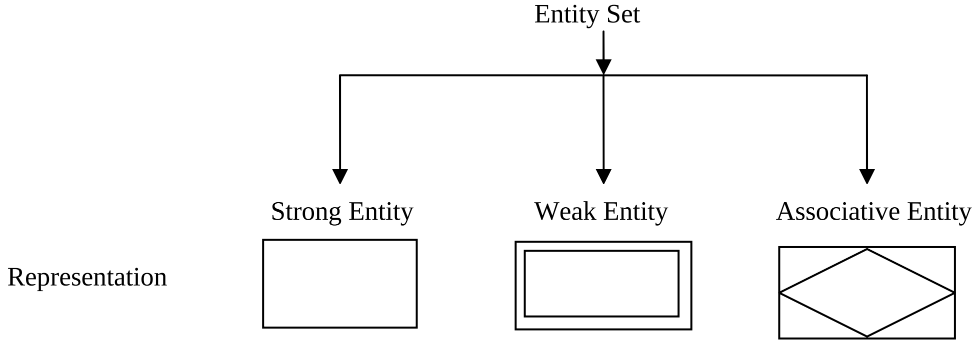


Figure 3.1 ER diagram

3.2 Classification of Entity Sets

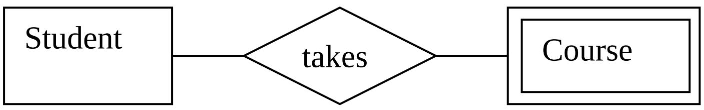
Entity sets can be broadly classified into:

1. Strong entity.
2. Weak entity.
3. Associative entity



3.2.1 Strong Entity

Strong entity is one whose existence does not depend on other entity.
Example: Consider the example, student takes course. Here student is a strong entity.

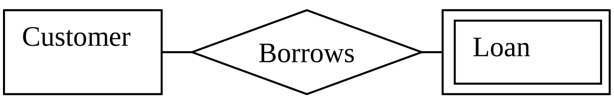


In this example, course is considered as weak entity because, if there are no students to take a particular course, then that course cannot be offered. The COURSE entity depends on the STUDENT entity.

3.2.2 Weak Entity

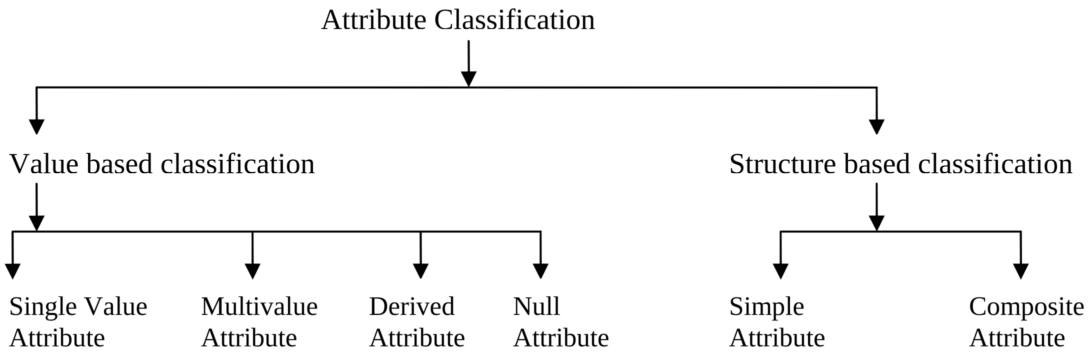
Weak entity is one whose existence depends on other entity. In many cases, weak entity does not have primary key.

Example: Consider the example, customer borrows loan. Here loan is a weak entity. For every loan, there should be at least one customer. Here the entity loan depends on the entity customer hence loan is a weak entity.



3.3 Attribute Classification

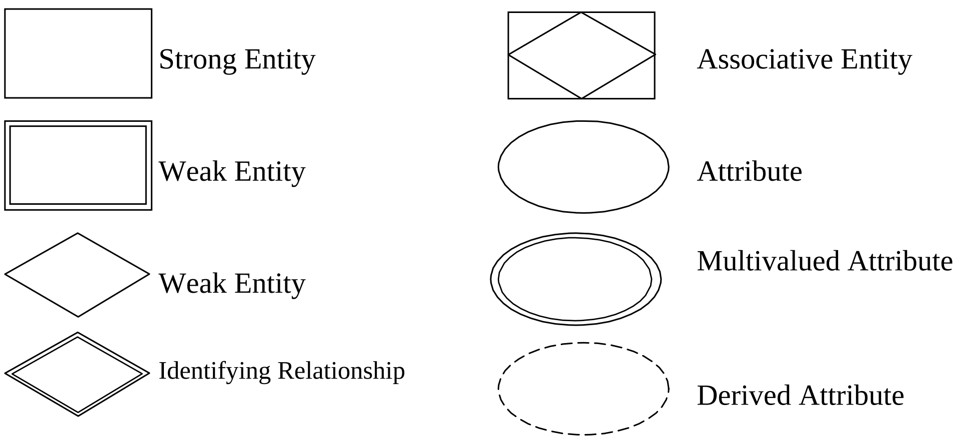
Attribute is used to describe the properties of the entity. This attribute can be broadly classified based on value and structure. Based on value the attribute can be classified into single value, multivalue, derived, and null value attribute. Based on structure, the attribute can be classified as simple and composite attribute.



3.3.1 Symbols Used in ER Diagram

The elements in ER diagram are Entity, Attribute, and Relationship. The different types of entities like strong, weak, and associative entity, different types of attributes like multivalued and derived attributes and identifying relationship and their corresponding symbols are shown below:

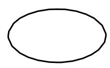
Basic Symbols



Single Value Attribute

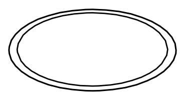
Single value attribute means, there is only one value associated with that attribute.

Example: The examples of single value attribute are age of a person, Roll number of the student, Registration number of a car, etc.

Representation of Single Value Attribute in ER Diagram 

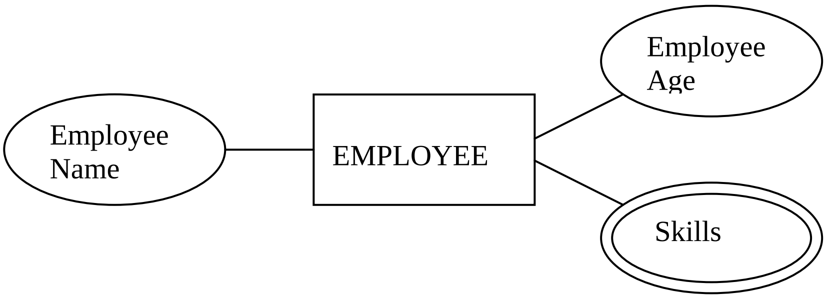
Multivalued Attribute

In the case of multivalue attribute, more than one value will be associated with that attribute.

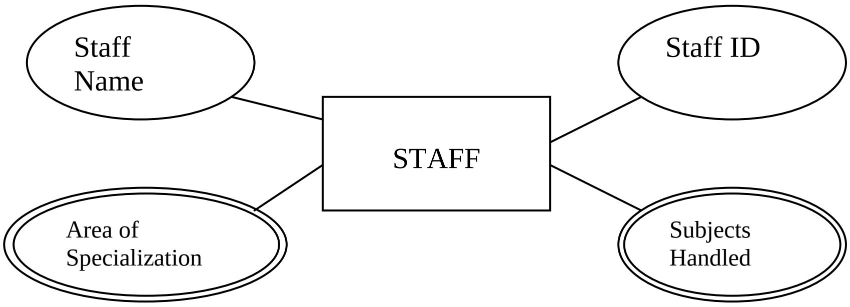
Representation of Multivalued Attribute in ER Diagram 

Examples of Multivalued Attribute

1. Consider an entity EMPLOYEE. An Employee can have many skills; hence skills associated to an employee are a multivalue attribute.



2. Subjects handled by a staff. A staff can handle more than one subject in a particular semester; hence it is an example of multivalue attribute.

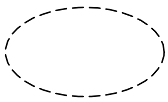


Moreover a staff can be an expert in more than one area, hence area of specialization is considered as multivalued attribute.

Derived Attribute

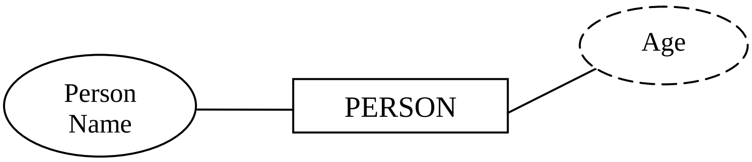
The value of the derived attribute can be derived from the values of other related attributes or entities. In ER diagram, the derived attribute is represented by dotted ellipse.

Representation of Derived Attribute in ER Diagram



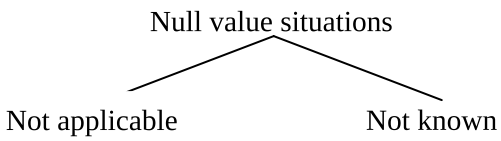
Example of Derived Attribute

1. Age of a person can be derived from the date of birth of the person. In this example, age is the derived attribute.



2. Experience of an employee in an organization can be derived from date of joining of the employee.

Null Value Attribute: In some cases, a particular entity may not have any applicable value for an attribute. For such situation, a special value called null value is created.

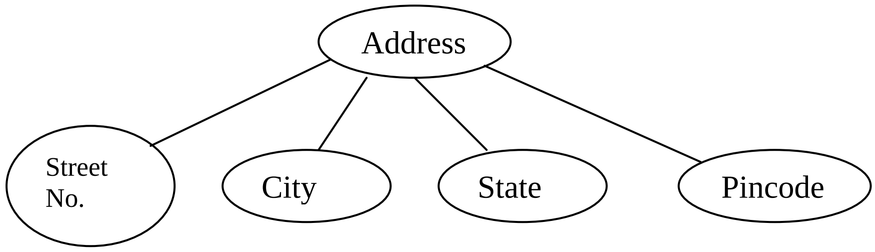


Example: In application forms, there is one column called phone no. if a person do not have phone then a null value is entered in that column.

Composite Attribute

Composite attribute is one which can be further subdivided into simple at- tributes.

Example: Consider the attribute "address" which can be further subdivided into Street name, City, and State.

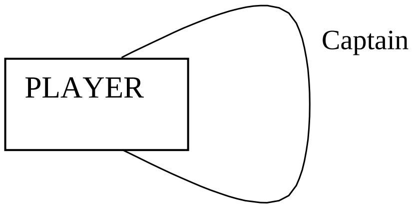


3.4 Relationship Degree

Relationship degree refers to the number of associated entities. The relationship degree can be broadly classified into unary, binary, and ternary relationship.

3.4.1 Unary Relationship

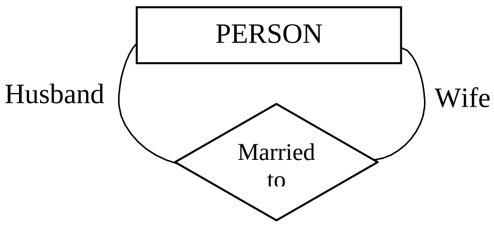
The unary relationship is otherwise known as recursive relationship. In the unary relationship the number of associated entity is one. An entity related to itself is known as recursive relationship.



Roles and Recursive Relation

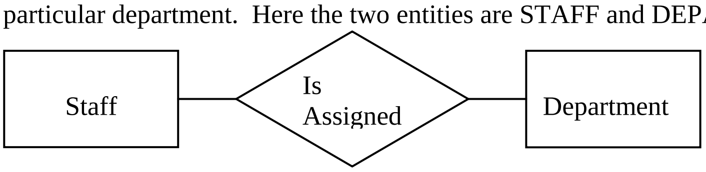
When an entity sets appear in more than one relationship, it is useful to add labels to connecting lines. These labels are called as roles.

Example: In this example, Husband and wife are referred as roles



3.4.2 Binary Relationship

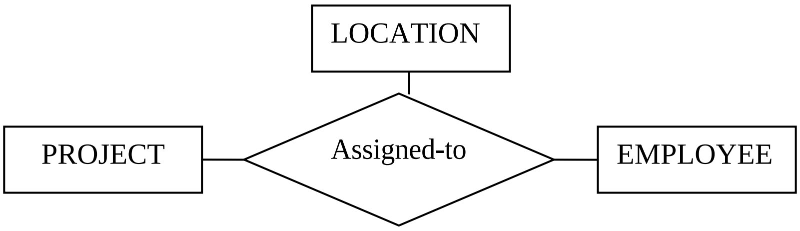
In a binary relationship, two entities are involved. Consider the example; each staff will be assigned to a particular department. Here the two entities are STAFF and DEPARTMENT.



3.4.3 Ternary Relationship

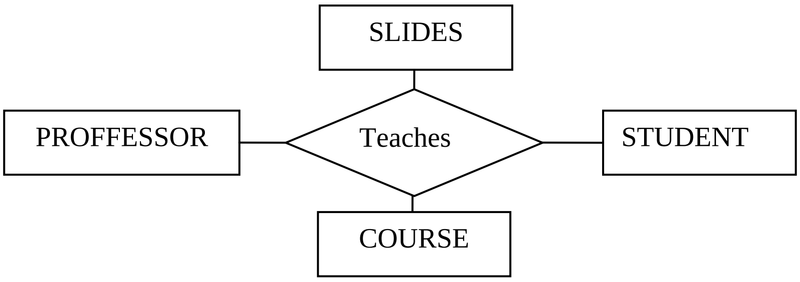
In a ternary relationship, three entities are simultaneously involved. Ternary relationships are required when binary relationships are not sufficient to accurately describe the semantics of an association among three entities.

Example: Consider the example of employee assigned a project. Here we are considering three entities EMPLOYEE, PROJECT, and LOCATION. The relationship is "assigned-to." Many employees will be assigned to one project hence it is an example of one-to-many relationship.



3.4.4 Quaternary Relationships

Quaternary relationships involve four entities. The example of quaternary relationship is "A professor teaches a course to students using slides." Here the four entities are PROFESSOR, SLIDES, COURSE, and STUDENT. The relationships between the entities are "Teaches."



3.5 Relationship Classification

Relationship is an association among one or more entities. This relationship can be broadly classified into one-to-one relation, one-to-many relation, many- to-many relation and recursive relation.

3.5.1 One-to-Many Relationship Type

The relationship that associates one entity to more than one entity is called one-to-many relationship. Example of one-to-many relationship is Country having states. For one country there can be more than one state hence it is an example of one-to-many relationship. Another example of one-to-many relationship is parent–child relationship. For one parent there can be more than one child. Hence it is an example of one-to-many relationship.

3.5.2 One-to-One Relationship Type

One-to-one relationship is a special case of one-to-many relationship. True one-to-one relationship is rare. The relationship between the President and the country is an example of one-to-one relationship. For a particular country there will be only one President. In general, a country will not have more than one President hence the relationship between the country and the President is an example of one-to-one relationship. Another example of one-to-one relationship is House to Location. A house is obviously in only one location.

3.5.3 Many-to-Many Relationship Type

The relationship between EMPLOYEE entity and PROJECT entity is an example of many-to-many relationship. Many employees will be working in many projects hence the relationship between employee and project is many- to-many relationship.

Table 2.1 Relationship types

Relationship type	Representation	Example
One-to-one		PRESIDENT — COUNTRY
One-to-many		DEPARTMENT — EMPLOYEES
Many -to- many		EMPLOYEE — PROJECT
Many-to-one		EMPLOYEE — DEPARTMENT

3.5.4 Many-to-One Relationship Type

The relationship between EMPLOYEE and DEPARTMENT is an example of many-to-one relationship. There may be many EMPLOYEES working in one DEPARTMENT. Hence relationship between EMPLOYEE and DEPARTMENT is many-to-one relationship. Four relationship types are summarized and shown in Table 2.1.

3.6 Constraints

An E-R enterprise schema may define certain constraints to which the contents of a database must conform. In this section, we examine mapping cardinalities and participation constraints, which are two of the most important types of constraints.

3.6.1 Mapping Cardinalities: Mapping cardinalities, or cardinality ratios, express the number of entities to which another entity can be associated via a relationship set.

Mapping cardinalities are most useful in describing binary relationship sets, al- though they can contribute to the description of relationship sets that involve more than two entity sets. In this section, we shall concentrate on only binary relationship sets.

For a binary relationship set R between entity sets A and B, the mapping cardinal- ity must be one of the following:

- One to one. An entity in A is associated with at most one entity in B, and an entity in B is associated with at most one entity in A. (See Figure 3.1a.)
- One to many. An entity in A is associated with any number (zero or more) of entities in B. An entity in B, however, can be associated with at most one entity in A. (See Figure 3.1b.)
- Many to one. An entity in A is associated with at most one entity in B. An entity in B, however, can be associated with any number (zero or more) of entities in A. (See Figure 3.2a.)
- Many to many. An entity in A is associated with any number (zero or more) of entities in B, and an entity in B is associated with any number (zero or more) of entities in A. (See Figure 3.2b.)

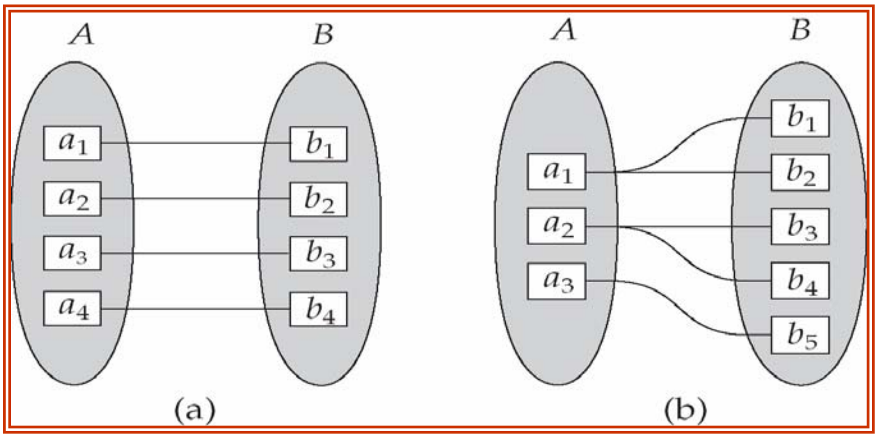


Figure 3.1 Mapping cardinalities. (a) One to one. (b) One to many.

The appropriate mapping cardinality for a particular relationship set obviously depends on the real-world situation that the relationship set is modeling.
As an illustration, consider the borrower relationship set. If, in a particular bank, a loan can belong to only one customer, and a customer can have several loans, then the relationship set from customer to loan is one to many. If a loan can belong to several customers (as can loans taken jointly by several business partners), the relationship set is many to many. Figure 3.3 depicts this type of relationship.

3.6.2 Participation Constraints

The participation of an entity set E in a relationship set R is said to be total if every entity in E participates in at least one relationship in R. If only some entities in E participate in relationships in R, the participation of entity set E in relationship R is said to be partial. For example, we expect every loan entity to be related to at least one customer through the borrower relationship. Therefore the participation of loan in the relationship set borrower is total. In contrast, an individual can be a bank customer whether or not she has a loan with the bank. Hence, it is possible that only some of the customer entities are related to the loan entity set through the borrower relationship, and the participation of customer in the borrower relationship set is therefore partial.

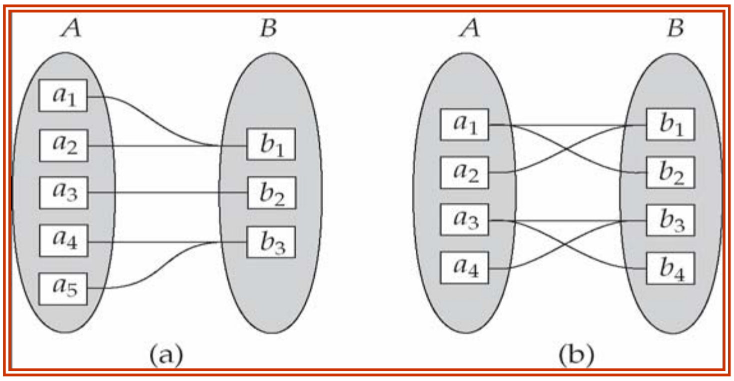
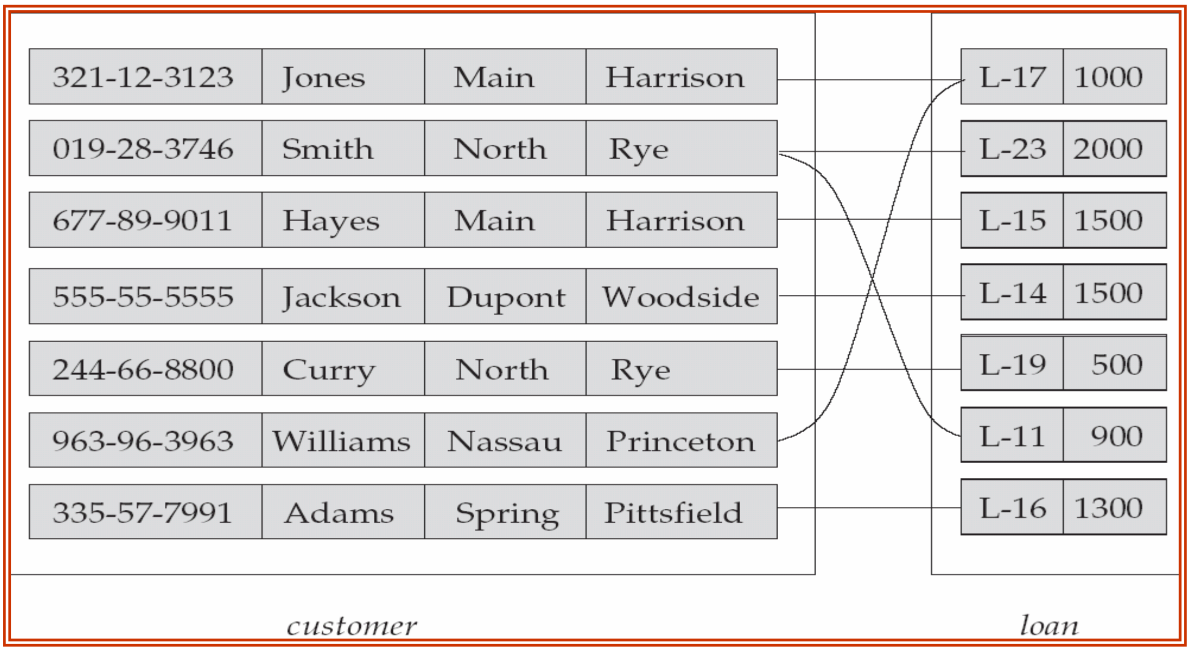


Figure 3.2 Mapping cardinalities. (a) Many to one. (b) Many to many



customer **loan**

Figure 3.3 Relationship set borrower.

3.7 Extended E-R Features

Although the basic E-R concepts can model most database features, some aspects of a database may be more aptly expressed by certain extensions to the basic E-R model. In this section, we discuss the extended E-R features of specialization, generalization, higher- and lower-level entity sets, attribute inheritance, and aggregation.

3.7.1 Specialization

An entity set may include subgroupings of entities that are distinct in some way from other entities in the set. For instance, a subset of entities within an entity set may have attributes that are not shared by all the entities in the entity set. The E-R model provides a means for representing these distinctive entity groupings.

Consider an entity set person, with attributes name, street, and city. A person may be further classified as one of the following:

- *customer*
- *employee*

Each of these person types is described by a set of attributes that includes all the at- tributes of entity set person plus possibly additional attributes. For example, customer entities may be described further by the attribute customer-id, whereas employee entities may be described further by the attributes employee-id and salary. The process of designating subgroupings within an entity set is called specialization. The specialization of person allows us to distinguish among persons according to whether they are employees or customers.

As another example, suppose the bank wishes to divide accounts into two categories, checking account and savings account. Savings accounts need a minimum balance, but the bank may set interest rates differently for different customers, offering better rates to favored customers. Checking accounts have a fixed interest rate, but offer an overdraft facility; the overdraft amount on a checking account must be recorded.

The bank could then create two specializations of account, namely savings-account and checking-account. As we saw earlier, account entities are described by the at- tributes account-number and balance. The entity set savings-account would have all the attributes of account and an additional attribute interest-rate. The entity set checking- account would have all the attributes of account, and an additional attribute overdraft-amount.

We can apply specialization repeatedly to refine a design scheme. For instance, bank employees may be further classified as one of the following:

- *officer*
- *teller*
- *secretary*

Each of these employee types is described by a set of attributes that includes all the attributes of entity set employee plus additional attributes. For example, officer entities may be described further by the attribute office-number, teller entities by the attributes station-number and hours-per-week, and secretary entities by the attribute hours-per- week. Further, secretary entities may participate in a relationship secretary-for, which identifies which employees are assisted by a secretary.

An entity set may be specialized by more than one distinguishing feature. In our example, the distinguishing feature among employee entities is the job the employee performs. Another, coexistent, specialization could be based on whether the person is a temporary (limited-term) employee or a permanent employee, resulting in the entity sets temporary-employee and permanent-employee. When more than one specialization is formed on an entity set, a particular entity may belong to multiple specializations. For instance, a given employee may be a temporary employee who is a secretary.

In terms of an E-R diagram, specialization is depicted by a triangle component labeled ISA, as Figure 3.4 shows. The label ISA stands for “is a” and represents, for example, that a customer “is a” person. The ISA relationship may also be referred to as a superclass-subclass relationship. Higher- and lower-level entity sets are depicted as regular entity sets - that is, as rectangles containing the name of the entity set.

3.7.2 Generalization

The refinement from an initial entity set into successive levels of entity subgroupings represents a top-down design process in which distinctions are made explicit. The design process may also proceed in a bottom-up manner, in which multiple entity sets are synthesized into a higher-level entity set on the basis of common features. The database designer may have first identified a customer entity set with the attributes name, street, city, and customer-id, and an employee entity set with the attributes name, street, city, employee-id, and salary.

There are similarities between the customer entity set and the employee entity set in the sense that they have several attributes in common. This commonality can be expressed by generalization, which is a containment relationship that exists between a higher-level entity set and one or more lower-level entity sets. In our example, person is the higher-level entity set and customer and employee are lower-level entity sets. Higher- and lower-level entity sets also may be designated by the terms superclass and subclass, respectively. The person entity set is the superclass of the customer and employee subclasses.

For all practical purposes, generalization is a simple inversion of specialization. We will apply both processes, in combination, in the course of designing the E-R schema for an enterprise. In terms of the E-R diagram itself, we do not distinguish between specialization and generalization. New levels of entity representation will be distinguished (specialization) or synthesized (generalization) as the design schema comes to express fully the database application and the user requirements of the database. Differences in the two approaches may be characterized by their starting point and overall goal.

Specialization stems from a single entity set; it emphasizes differences among entities within the set by creating distinct lower-level entity sets. These lower-level entity sets may have attributes, or may participate in relationships, that do not apply to all the entities in the higher-level entity set. Indeed, the reason a designer applies specialization is to represent such distinctive features. If customer and employee neither have attributes that person entities do not have nor participate in different relationships than those in which person entities participate, there would be no need to specialize the person entity set.

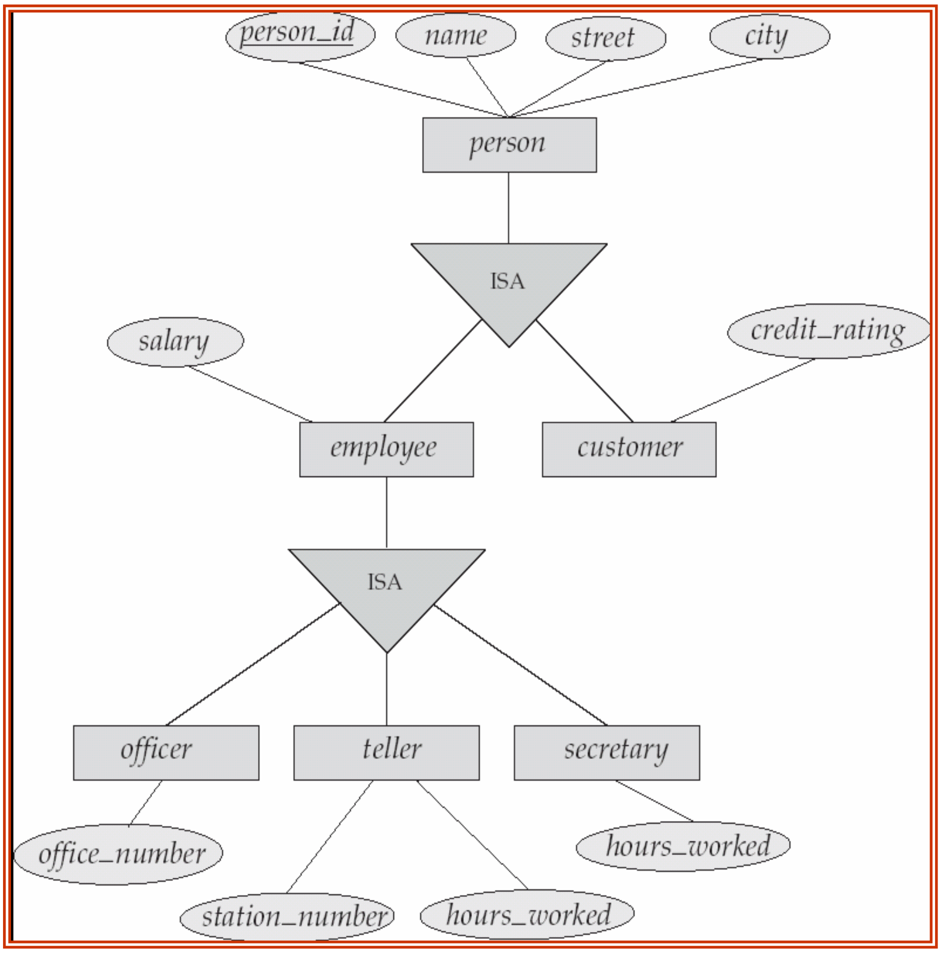


Figure 3.4 Specialization and generalization.

Generalization proceeds from the recognition that a number of entity sets share some common features (namely, they are described by the same attributes and participate in the same relationship sets). On the basis of their commonalities, generalization synthesizes these entity sets into a single, higher-level entity set. Generalization is used to emphasize the similarities among lower-level entity sets and to hide the differences; it also permits an economy of representation in that shared attributes are not repeated.

Summary of Symbols Used in E-R Notation

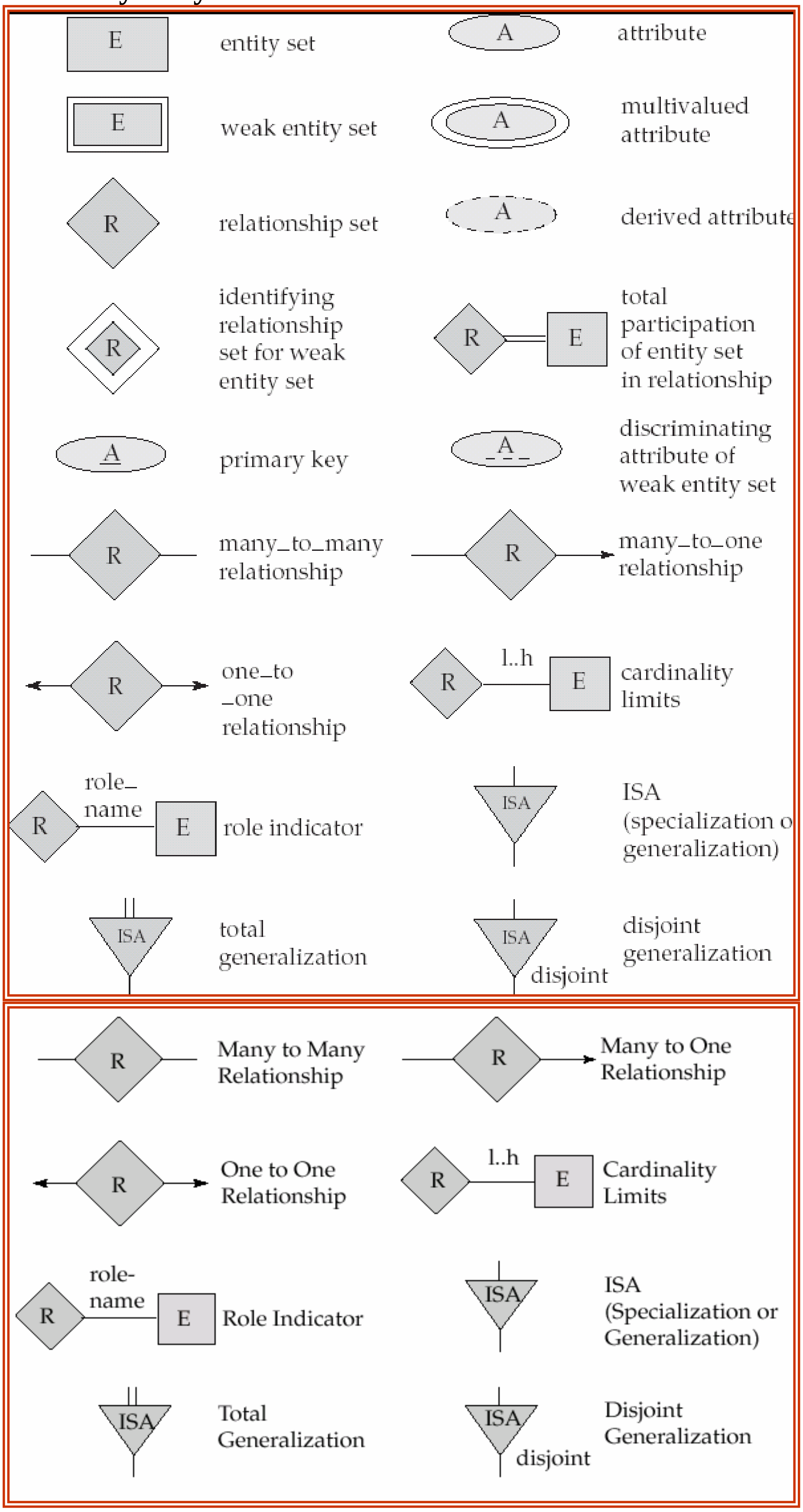


Figure 3.5 Symbols used in the E-R notation.

3.7.3 Database Design for Banking Enterprise

We now look at the database-design requirements of a banking enterprise in more detail, and develop a more realistic, but also more complicated, design than what we have seen in our earlier examples. However, we do not attempt to model every aspect of the database-design for a bank; we consider only a few aspects, in order to illustrate the process of database design.

account_number	*branch_name*	*balance*
A-101	Downtown	500
A-102	Perryridge	400
A-201	Brighton	900
A-215	Mianus	700
A-217	Brighton	750
A-222	Redwood	700
A-305	Round Hill	350

Figure 3.6 The *account* relation.

branch_name	*branch_city*	*assets*
Brighton	Brooklyn	7100000
Downtown	Brooklyn	9000000
Mianus	Horseneck	400000
North Town	Rye	3700000
Perryridge	Horseneck	1700000
Pownal	Bennington	300000
Redwood	Palo Alto	2100000
Round Hill	Horseneck	8000000

Figure 3.7 The *branch* relation.

customer_name	*account_number*
Hayes	A-102
Johnson	A-101
Johnson	A-201
Jones	A-217
Lindsay	A-222
Smith	A-215
Turner	A-305

Figure 3.8 The *depositor* Relation

customer_name	*loan_number*
Adams	L-16
Curry	L-93
Hayes	L-15
Jackson	L-14
Jones	L-17
Smith	L-11
Smith	L-23
Williams	L-17

Figure 3.9 The *borrower* relation

customer_name	*customer_street*	*customer_city*
Adams	Spring	Pittsfield
Brooks	Senator	Brooklyn
Curry	North	Rye
Glenn	Sand Hill	Woodside
Green	Walnut	Stamford
Hayes	Main	Harrison
Johnson	Alma	Palo Alto
Jones	Main	Harrison
Lindsay	Park	Pittsfield
Smith	North	Rye
Turner	Putnam	Stamford
Williams	Nassau	Princeton

Figure 3.8 The *customer* Relation

loan_number	*branch_name*	*amount*
L-11	Round Hill	900
L-14	Downtown	1500
L-15	Perryridge	1500
L-16	Perryridge	1300
L-17	Downtown	1000
L-23	Redwood	2000
L-93	Mianus	500

Figure 3.10 The *loan* relation

3.7.3.1 Data Requirements: The initial specification of user requirements may be based on interviews with the database users, and on the designer's own analysis of the enterprise. The description that arises from this design phase serves as the basis for specifying the conceptual structure of the database. Here are the major characteristics of the banking enterprise.

- The bank is organized into branches. Each branch is located in a particular city and is identified by a unique name. The bank monitors the assets of each branch.
- Bank customers are identified by their customer-id values. The bank stores each customer's name, and the street and city where the customer lives. Customers may have accounts and can take out loans. A customer may be associated with a particular banker, who may act as a loan officer or personal banker for that customer.
- Bank employees are identified by their employee-id values. The bank administration stores the name and telephone number of each employee, names of the employee's dependents, and the employee-id number of the employee's manager. The bank also keeps track of the employee's start date and, thus, length of employment.
- The bank offers two types of accounts — savings and checking accounts. Ac- counts can be held by more than one customer, and a customer can have more than one account. Each account is assigned a unique account number. The bank maintains a record of each account's balance, and the most recent date on which the account was accessed by each customer holding the account. In addition, each savings account has an interest rate, and overdrafts are recorded for each checking account.
- A loan originates at a particular branch and can be held by one or more customers. A loan is identified by a unique loan number. For each loan, the bank keeps track of the loan amount and the loan payments. Although a loan- payment number does not uniquely identify a particular payment among those for all the bank's loans, a payment number does identify a particular payment for a specific loan. The date and amount are recorded for each payment.

In a real banking enterprise, the bank would keep track of deposits and withdrawals from savings and checking accounts, just as it keeps track of payments to loan accounts. Since the modeling requirements for that tracking are similar, and we would like to keep our example application small, we do not keep track of such de- posits and withdrawals in our model.

3.7.3.2 Entity Sets Designation

Our specification of data requirements serves as the starting point for constructing a conceptual schema for the database. From the characteristics listed in Section 3.8.3.1, we begin to identify entity sets and their attributes:

- The branch entity set, with attributes branch-name, branch-city, and assets.
- The customer entity set, with attributes customer-id, customer-name, customer - street; and customer-city. A possible additional attribute is banker-name.
- The employee entity set, with attributes employee-id, employee-name, telephone- number, salary, and manager. Additional descriptive features are the multivalued attribute dependent-name, the base attribute start-date, and the derived at- tribute employment-length.
- Two account entity sets — savings-account and checking-account — with the common attributes of account-number and balance; in addition, savings-account has the attribute interest-rate and checking-account has the attribute overdraft-amount.
- The loan entity set, with the attributes loan-number, amount, and originating- branch.
- The weak entity set loan-payment, with attributes payment-number, payment- date, and payment-amount.

3.7.3.3 Relationship Sets Designation

We now return to the rudimentary design scheme of Section 3.8.3.2 and specify the following relationship sets and mapping cardinalities. In the process, we also refine some of the decisions we made earlier regarding attributes of entity sets.

- borrower, a many-to-many relationship set between customer and loan.

- loan - branch, a many-to-one relationship set that indicates in which branch a loan originated. Note that this relationship set replaces the attribute originating- branch of the entity set loan.
- loan - payment, a one-to-many relationship from loan to payment, which documents that a payment is made on a loan.
- depositor, with relationship attribute access-date, a many-to-many relationship set between customer and account, indicating that a customer owns an account.
- cust - banker, with relationship attribute type, a many-to-one relationship set expressing that a customer can be advised by a bank employee, and that a bank employee can advise one or more customers. Note that this relationship set has replaced the attribute banker-name of the entity set customer.
- works - for, a relationship set between employee entities with role indicators manager and worker; the mapping cardinalities express that an employee works for only one manager and that a manager supervises one or more employees. Note that this relationship set has replaced the manager attribute of employee.

2.7.3.4 E-R Diagram

Drawing on the discussions in Section 3.7.3.3, we now present the completed E-R diagram for our example banking enterprise. Figure 3.6 depicts the full representation of a conceptual model of a bank, expressed in terms of E-R concepts. The diagram includes the entity sets, attributes, relationship sets, and mapping cardinalities arrived at through the design processes of Sections 3.7.3.1 and 3.7.3.2, and refined in Section 3.7.3.3.

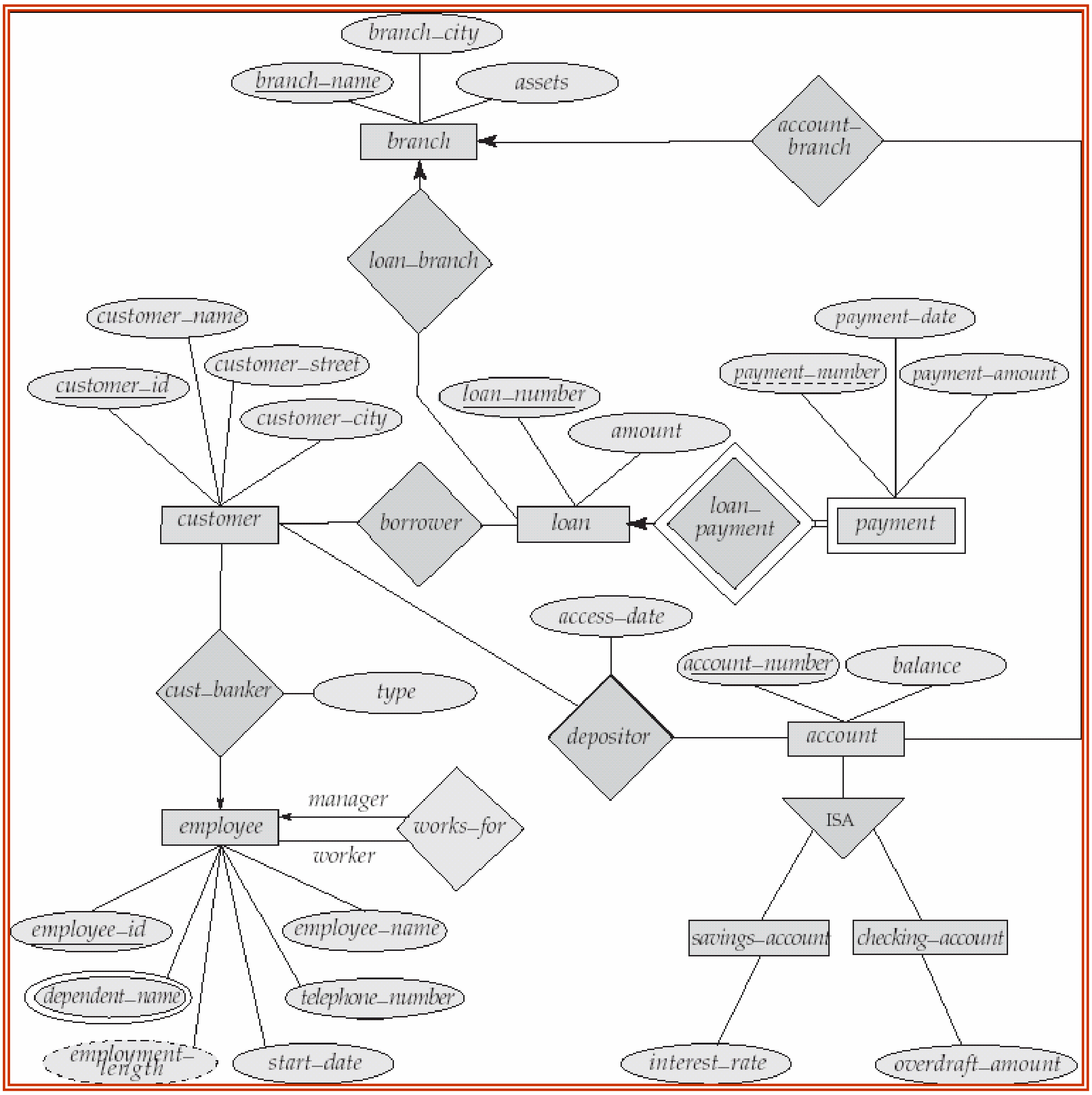


Figure 3.6 E-R diagram for a banking enterprise.

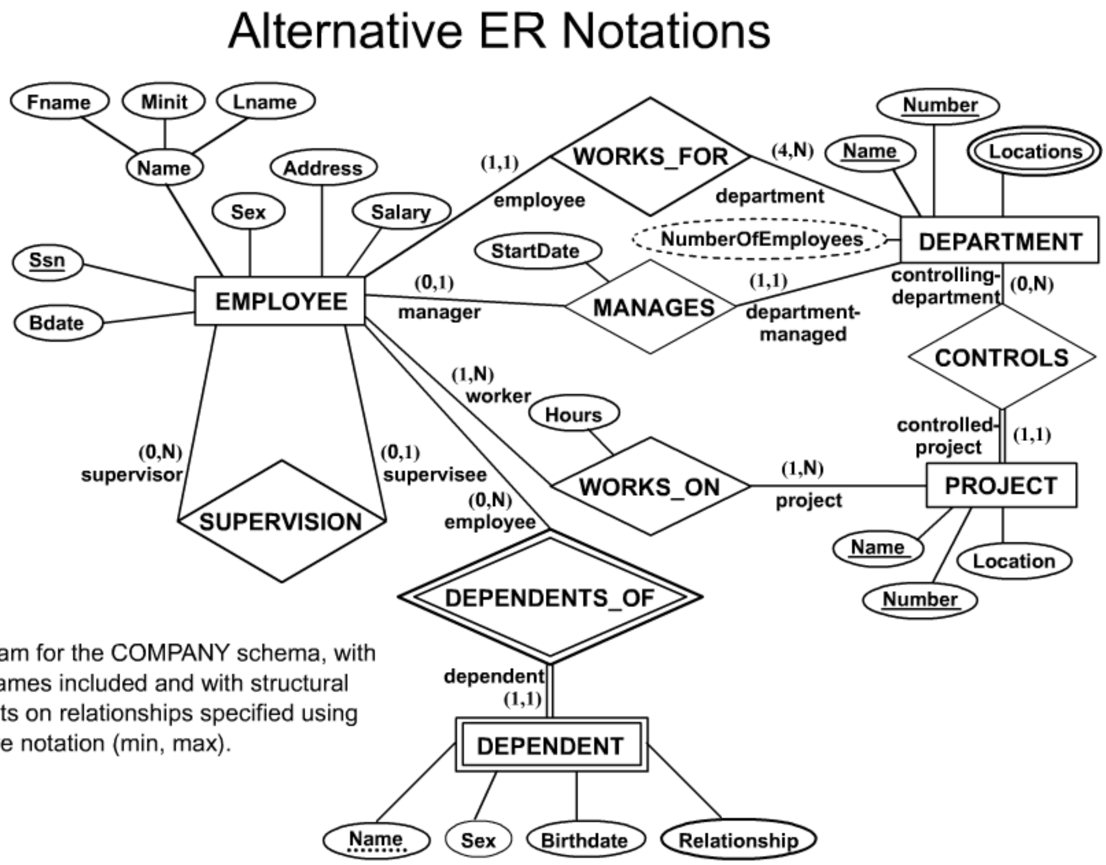
3.8 ER Diagram for COMPANY schema

Summary of Notation for ER Diagrams

Entity types such as EMPLOYEE, DEPARTMENT, and PROJECT are shown in rectangular boxes. Relationship types such as WORKS_FOR, MANAGES, CONTROLS, and WORKS_ON are shown in diamond-shaped boxes attached to the participating entity types with straight lines. Attributes are shown in ovals, and each attribute is attached by a straight line to its entity type or relationship type. Component attributes of a composite attribute are attached to the oval representing the composite attribute, as illustrated by the Name attribute of EMPLOYEE. Multivalued attributes are shown in double ovals, as illustrated by the Locations attribute of DEPARTMENT. Key attributes have their names underlined. Derived attributes are shown in dotted ovals, as illustrated by the NumberOfEmployees attribute of DEPARTMENT.

Weak entity types are distinguished by being placed in double rectangles and by having their identifying relationship placed in double diamonds, as illustrated by the DEPENDENT entity type and the DEPENDENTS_OF identifying relationship type. The partial key of the weak entity type is underlined with a dotted line.

The cardinality ratio of DEPARTMENT: EMPLOYEE in MANAGES is 1:1, whereas it is 1:N for DEPARTMENT:EMPLOYEE in WORKS_FOR, and it is M:N for WORKS_ON. The participation constraint is specified by a single line for partial participation and by double lines for total participation (existence dependency).



ER diagram for the COMPANY schema, with all role names included and with structural constraints on relationships specified using alternative notation (min, max).

Figure 3.7 COMPANY ER Schema Diagram using (min, max) notation