2
Entity-Relationship Model

- Introduction
- Basic Concepts
- Constraints
- Keys
- Entity-Relationship (ER) Diagram
- EER (Extended ER Modeling)
- Summary of Symbols in ER Diagram
- Alternative ER Notations
- Design Issues

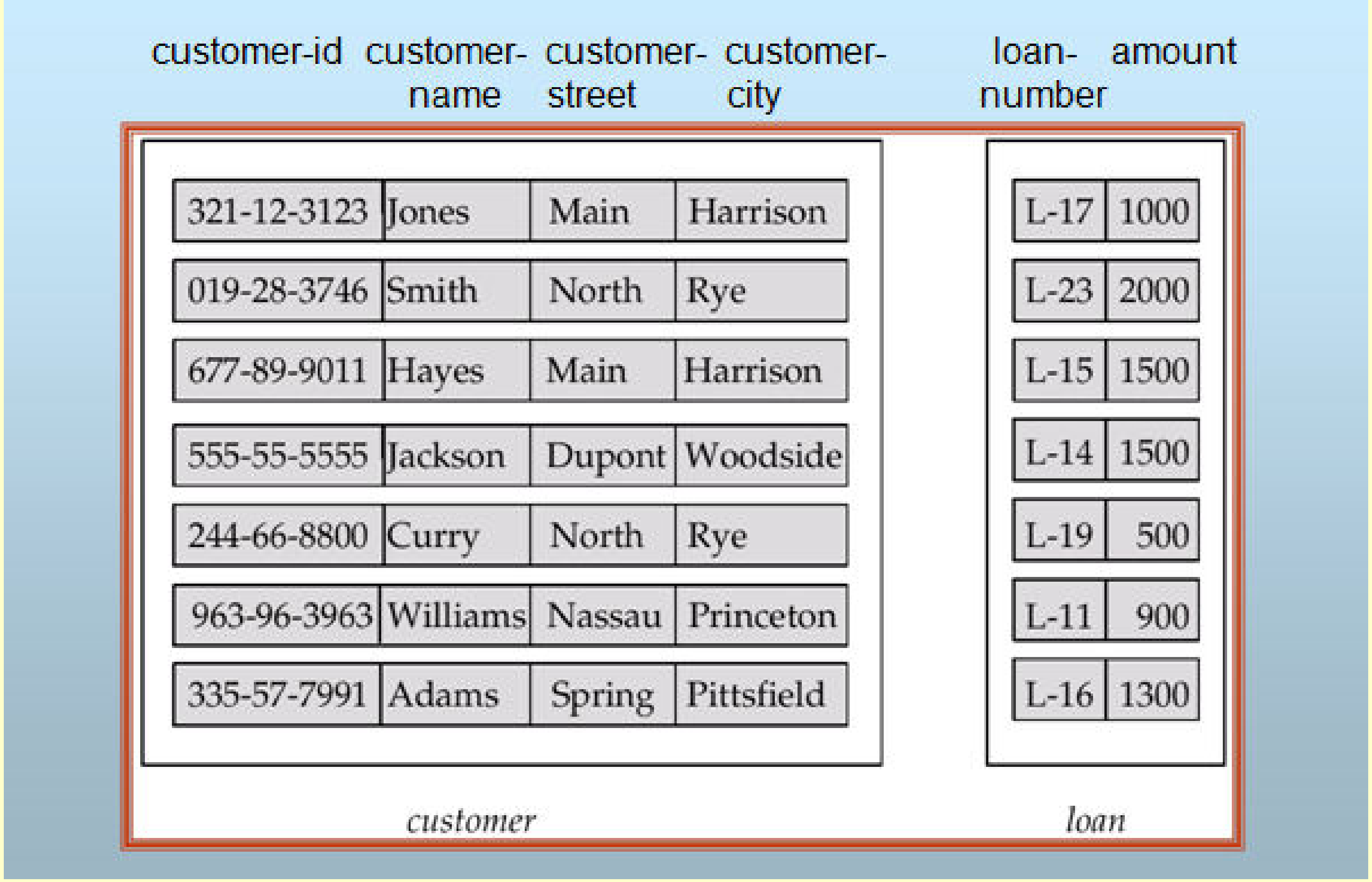
Introduction

- The **entity-relationship (E-R) model** is a high-level data model based on the perception of a real world that consists of a collection of basic objects, called **entities** and **relationships** among these entities.
- An **entity** is a thing or object in the real world that is distinguishable from other objects.
- A **relationship** is an association among several entities.
- Entities are described by a set of **attributes**.
- The set of all entities of the same type is called an **entity set** and the set of all relationships of the same type is called a **relationship set**.
- In this model, we use **E-R diagrams** to express overall logical structure of the database.

Basic Concepts

- The E-R model employs three basic notations: **entity sets**, **relationship sets**, and **attributes**.
- **Entity Sets:**
  - An **entity** is a thing or object in the real world that is distinguishable from all other objects. For example, specific person, company, event, plant etc.
  - An entity has a set of properties called **attributes**, and the values for some set of properties may uniquely identify an entity. For example, people have *names* and *addresses* as attributes.
  - An **entity set** is a set of entities of the same type that share the same properties, or attributes. For example, set of all persons, companies, trees, holidays etc.
  - The individual entities that constitute a set are said to be the **extension** of the entity set. Entity sets do not need to be disjoint.

Basic Concepts



customer-id	customer-name	customer-street	customer-city
321-12-3123	Jones	Main	Harrison
019-28-3746	Smith	North	Rye
677-89-9011	Hayes	Main	Harrison
555-55-5555	Jackson	Dupont	Woodside
244-66-8800	Curry	North	Rye
963-96-3963	Williams	Nassau	Princeton
335-57-7991	Adams	Spring	Pittsfield

customer

loan-number	amount
L-17	1000
L-23	2000
L-15	1500
L-14	1500
L-19	500
L-11	900
L-16	1300

loan

Fig: Entity sets **customer** and **loan**

Basic Concepts

- **Attributes:**
  - An entity is represented by a set of **attributes**. For example, a *customer* entity can have *customer-id, customer-name, customer-street,* and *customer-city* as attributes.
  - Attributes are the descriptive properties possessed by all members of an entity set. Each entity may have its own value for each attribute.
  - For each attribute, there is a set of permitted values, called the **domain** or **value set** of that attribute.
  - The E-R model uses the different types of attributes.
  - **Simple and Composite Attributes:**
    - **Simple attributes** cannot be divided into subparts. For example, *gender*.
    - **Composite attributes** on the other hand can be divided into subparts; that is, other attributes. A composite attribute may appear as hierarchy. For example, *name, address* etc.

Basic Concepts

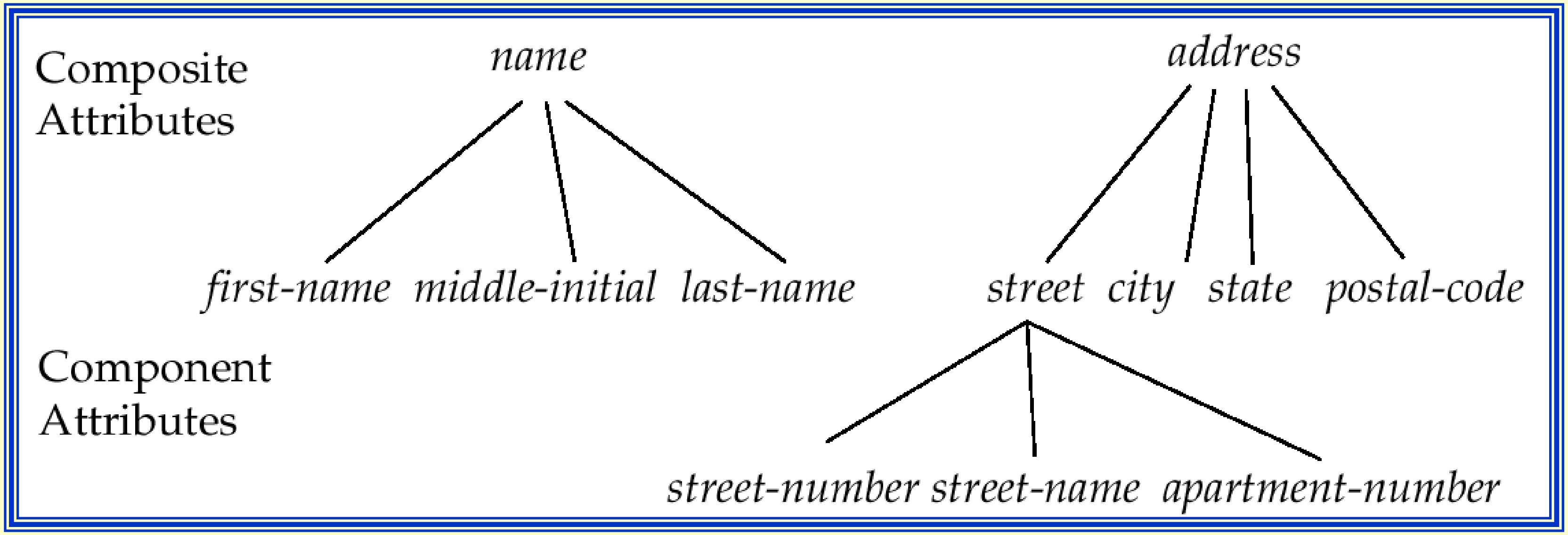


Fig: Composite attributes

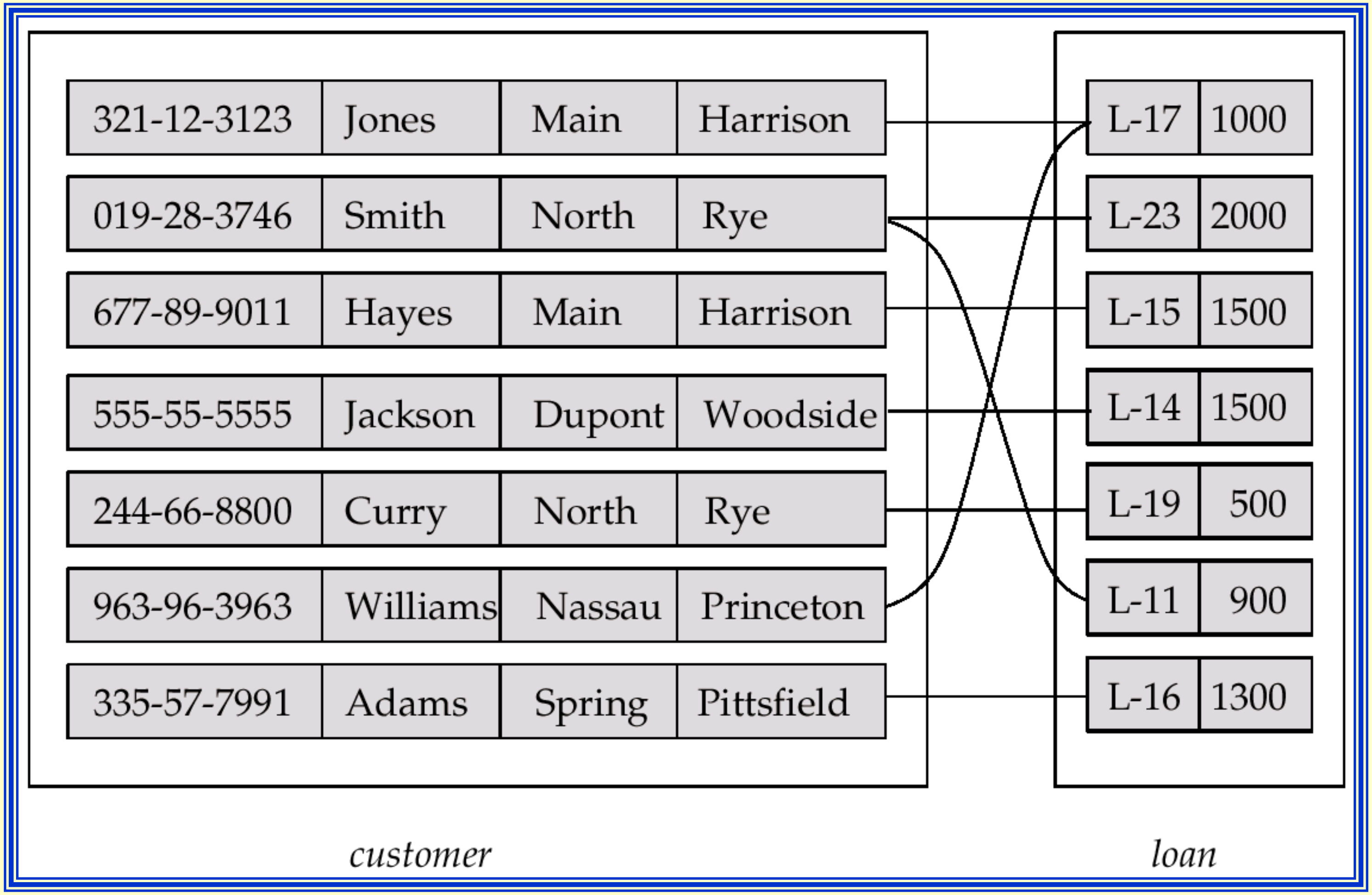
Basic Concepts

- **Single-valued and Multi-valued Attributes:**
  - An attribute having a single value for a particular entity is called the **single-valued attribute**. For example, *gender*.
  - An attribute having a set of values for a particular entity is called the **multi-valued attribute**. For example, *phone-numbers*.
- **Derived Attributes:**
  - The value of the **derived attributes** can be computed from other attributes. For example, *age*, given *date-of-birth*.
- All attributes take a **null** value when an entity does not have a value for it. The null value may indicate "not applicable", that is, the value does not exist for the entity.

Basic Concepts

- **Relationship Sets:**
  - A **relationship** is an association among several entities.
  - A **relationship set** is a set of relationships of the same type.
  - Formally, it is a mathematical relation on $n \geq 2$ entity sets. If E_1, E_2,…, E_n are entity sets, then a relationship set R is a subset of $\{(e_1, e_2,…, e_n) | e_1 \in E_1, e_2 \in E_2,…, e_n \in E_n\}$, where $(e_1, e_2,….., e_n)$ is a relationship. For example, (*Hayes, L-15*) ∈ *borrower* relationship set.
  - The association between entity sets is referred to as **participation**, that is, the entity sets E_1, E_2,…., E_n participate in relationship set R.
  - A **relationship instance** represents an association between the named entities.

Basic Concepts



321-12-3123	Jones	Main	Harrison
019-28-3746	Smith	North	Rye
677-89-9011	Hayes	Main	Harrison
555-55-5555	Jackson	Dupont	Woodside
244-66-8800	Curry	North	Rye
963-96-3963	Williams	Nassau	Princeton
335-57-7991	Adams	Spring	Pittsfield

customer

L-17	1000
L-23	2000
L-15	1500
L-14	1500
L-19	500
L-11	900
L-16	1300

loan

Fig: Relationship set **borrower**

Basic Concepts

- A relationship set may also have attributes called **descriptive attributes**. For example, the *depositor* relationship set between entity sets *customer* and *account* may have the attribute *access-date*. See figure on the next slide.
- A relationship instance in a given relationship set must be uniquely identifiable from other relationship instances, without using descriptive attributes.

Basic Concepts

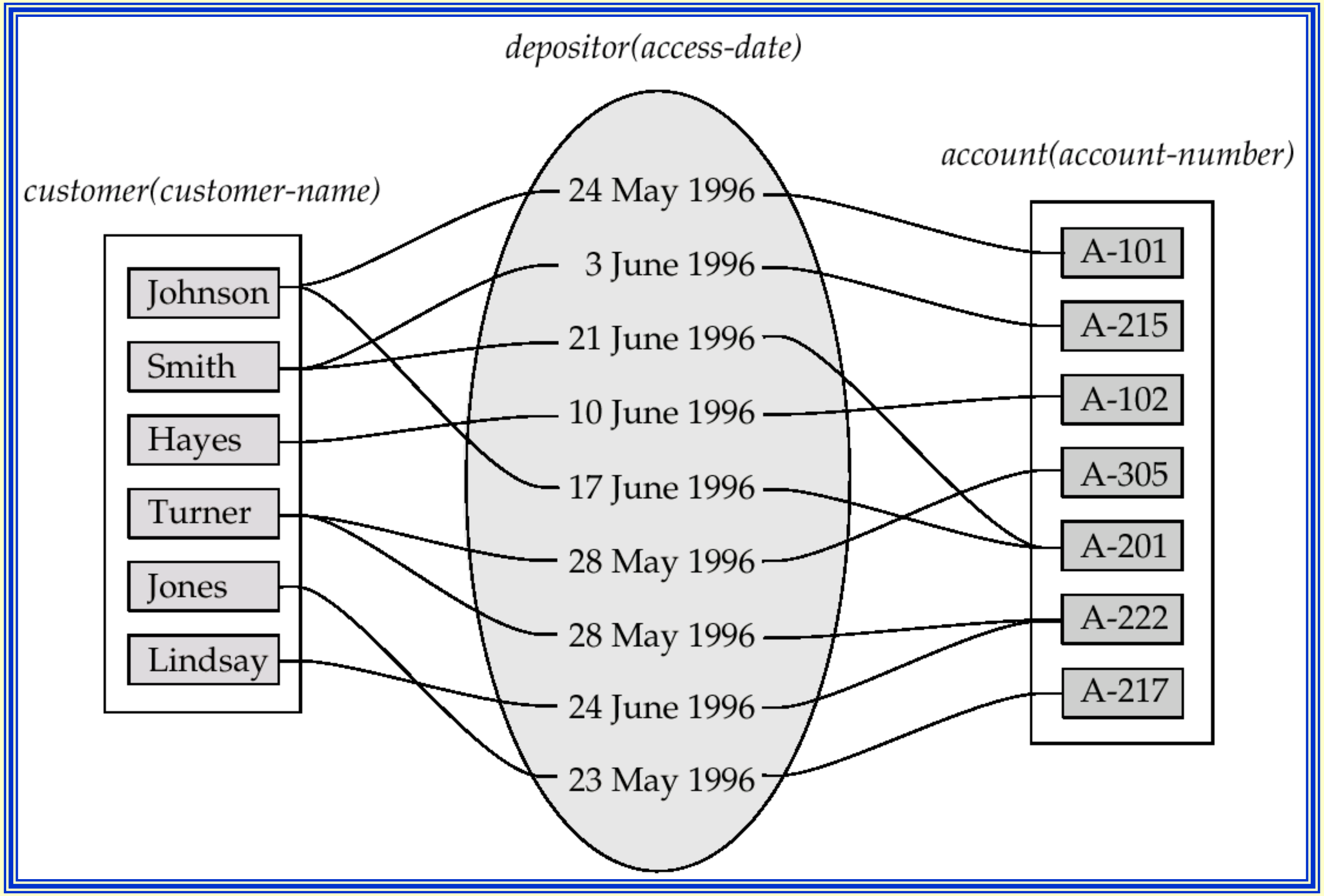


Fig: Descriptive attribute

Basic Concepts

- There can be more than one relationship set involving the same entity set.
- **Degree of a relationship set** refers to the number of entity sets that participate in a relationship set.
- Relationship sets that involve two entity sets are called **binary relationship** sets. Most relationship sets in a database system are binary.
- Relationship sets may involve more than two entity sets called **n-ary** relationship sets but are rare. For example, suppose employees of a bank may have jobs (responsibilities) at multiple branches, with different jobs at different branches. Then there is a ternary relationship set between entity sets *employee, job and branch.*

Basic Concepts

- The function that an entity plays in a relationship is called that entity's **role**.
- In case of distinct entity sets participating in a relationship set, roles are implicit and are not usually specified.
- When the entity sets of a relationship set are not distinct; that is, the same entity set participates in a relationship set more than once, in different roles, explicit role names are necessary to specify how an entity participates in a relationship instance.
- This type of relationship set is sometimes called a **recursive** relationship set.

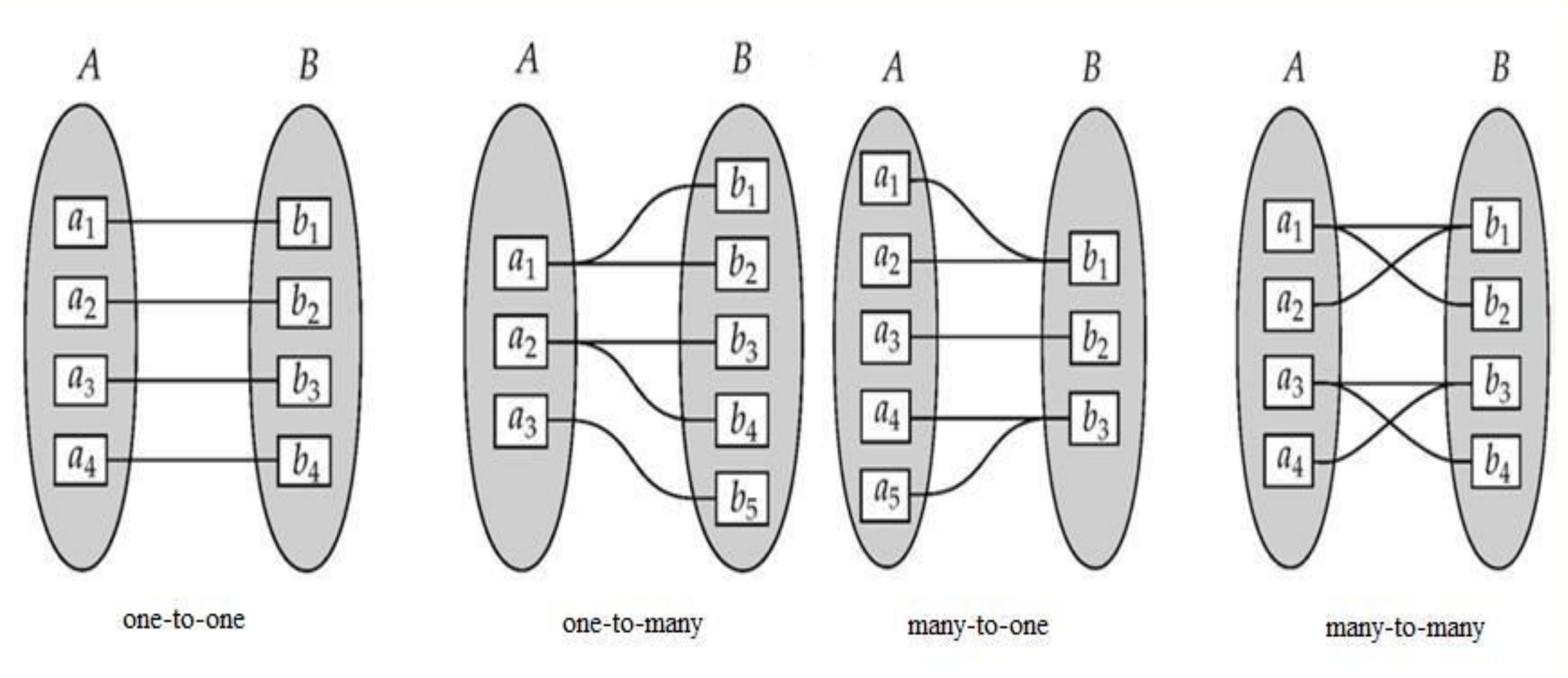
Constraints

- An entity relationship model may define certain constraints to which the contents of a database must conform. The most important constraints are: **mapping cardinalities** and **participation constraints**.
- **Mapping Cardinalities:**
  - These are also called **cardinalities ratios** and express the number of entities to which another entity can be associated via a relationship set.
  - These are most useful in describing binary relationship sets.
  - For binary relationship set **R** between entity sets **A** and **B**, there are four types of mapping cardinalities: *one-to-one*, *one-to-many*, *many-to-one*, and *many-to-many*.
  - **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**. For example, mapping cardinality between *departments* and *chairpersons*.

Constraints

- **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**. For example, mapping cardinality between *mothers* and *children*.
- **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**. For example, mapping cardinality between *children* and *mothers*.
- **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**. For example, mapping cardinality between *students* and *courses*.

Constraints



one-to-one

one-to-many

many-to-one

many-to-many

Constraints

- **Participation Constraints:**
  - The participation of an entity set **A** in a relationship set **R** is said to be **total** if every entity in **A** participates in at least one relationship in **R**. For example, consider *customer* and *account* entity sets in a banking system, and a relationship set *depositor* between them indicating that each customer must have an account. Then there is total participation of entity set *customer* in the relationship set *depositor*.
  - If only some entities in **A** participate in relationships in **R**, the participation of entity set **A** in relationship set **R** is said to be **partial**. For example, consider *customer* and *loan* entity sets in a banking system, and a relationship set *borrower* between them indicating that some customers have loans. Then there is partial participation of entity set *customer* in the relationship set *borrower*.

Keys

- Keys are used to distinguish the entities within a given entity set. Furthermore, a key allows us to identify a set of attributes that suffice to distinguish entities from each other. Keys also help to uniquely identify relationships.
- **Keys for Entity Sets:**
  - A **superkey** of an entity set is a set of one or more attributes whose values uniquely determine each entity. Any superset of a superkey is also a superkey.
  - A superkey may also contain extraneous attributes. A **candidate key** of an entity set is a minimal superkey; that is, superkey for which no proper subset is a superkey. For example, *customer-id* is candidate key of *customer* entity set.
  - An entity set may have several candidate keys. We use the term **primary key** to denote a candidate key that is chosen by the database designer for identifying entities within an entity set.

Keys

  - **Alternate keys** are the remaining candidate keys other than primary key.
  - So, there is always one and only one primary key in an entity set. A primary key of an entity set allows us to distinguish among various entities of the set.

- **Keys for Relationship Sets:**
  - The combination of primary keys of the participating entity sets forms a superkey for the relationship set. For example, (*customer-id, account-number*) is the super key of *depositor* relationship set.
  - Let R be a relationship set involving entities E_1, E_2,……, E_n. Let *primary-key* (E_i) denotes the set of attributes that forms the primary key for entity set E_i. Assume that the attribute names of all primary keys are unique, and each entity set participates only once in the relationship. The superkey for the relationship set can be defined as

    primary-key (E_1)U *primary-key* (E_2)U……U *primary-key* (E_n)

Entity-Relationship Diagram

- An E-R diagram expresses the overall logical structure of a database graphically. E-R diagram consists of the following major components:
  - **Rectangles** represent entity sets.
  - **Diamonds** represent relationship sets.
  - **Lines** link attributes to entity sets and entity sets to relationship sets.
  - **Ellipses** represent attributes
  - **Double ellipses** represent multivalued attributes.
  - **Dashed ellipses** denote derived attributes.
  - **Underline** indicates primary key attributes
  - **Double Lines** indicate total participation of an entity set in a relationship set.
  - **Double Rectangles** represent weak entity sets.
  - **Double Diamonds** represent identifying relationship sets.

Entity-Relationship Diagram

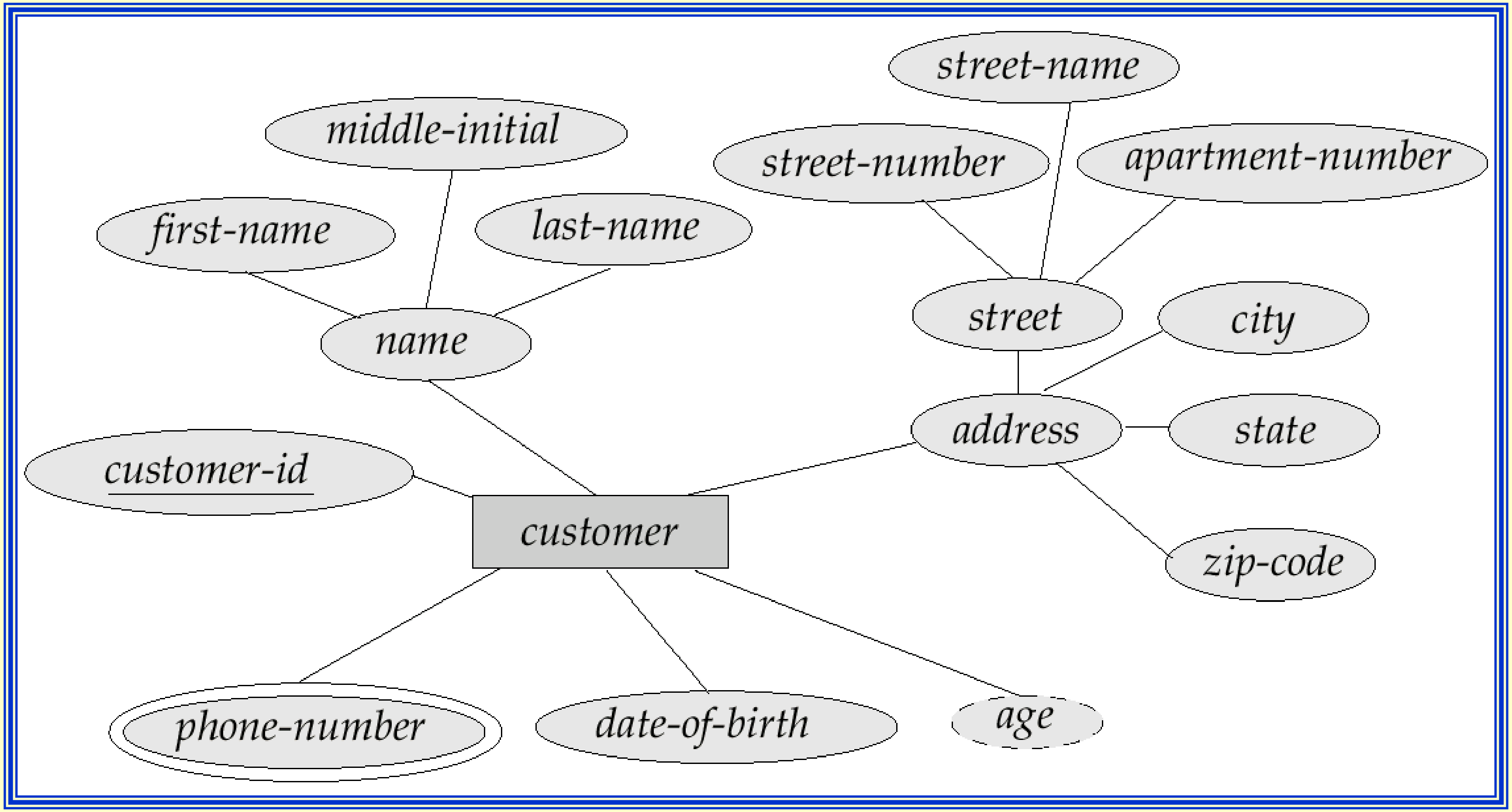


Fig: E-R diagram with *composite*, *multivalued*, and *derived* attributes

Entity-Relationship Diagram

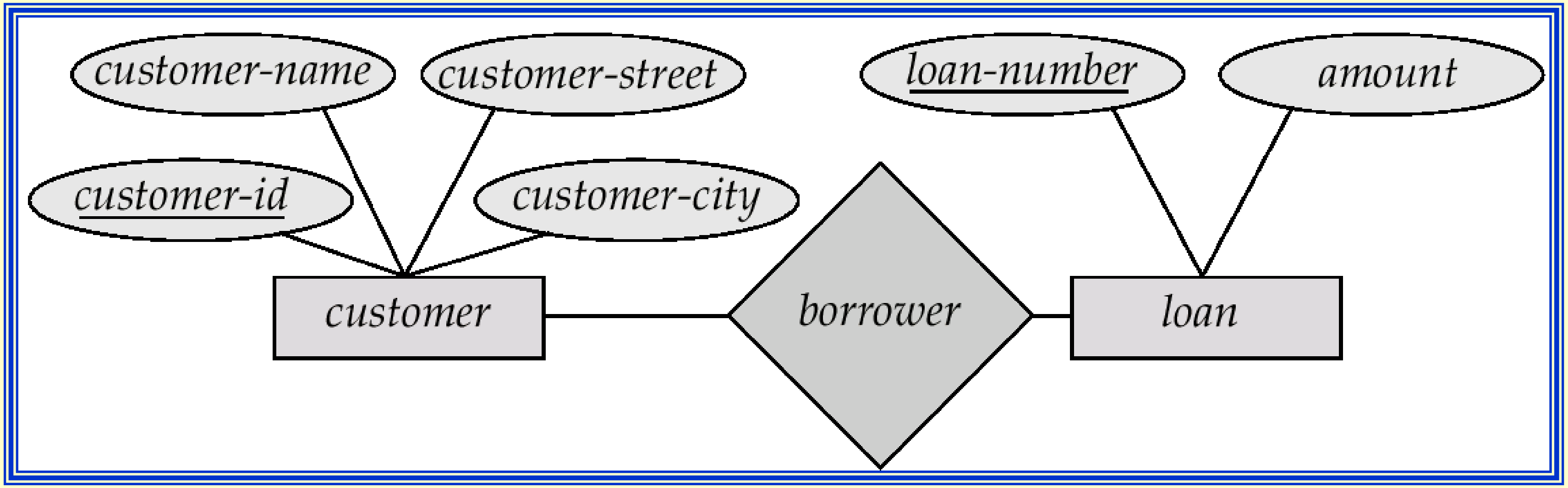
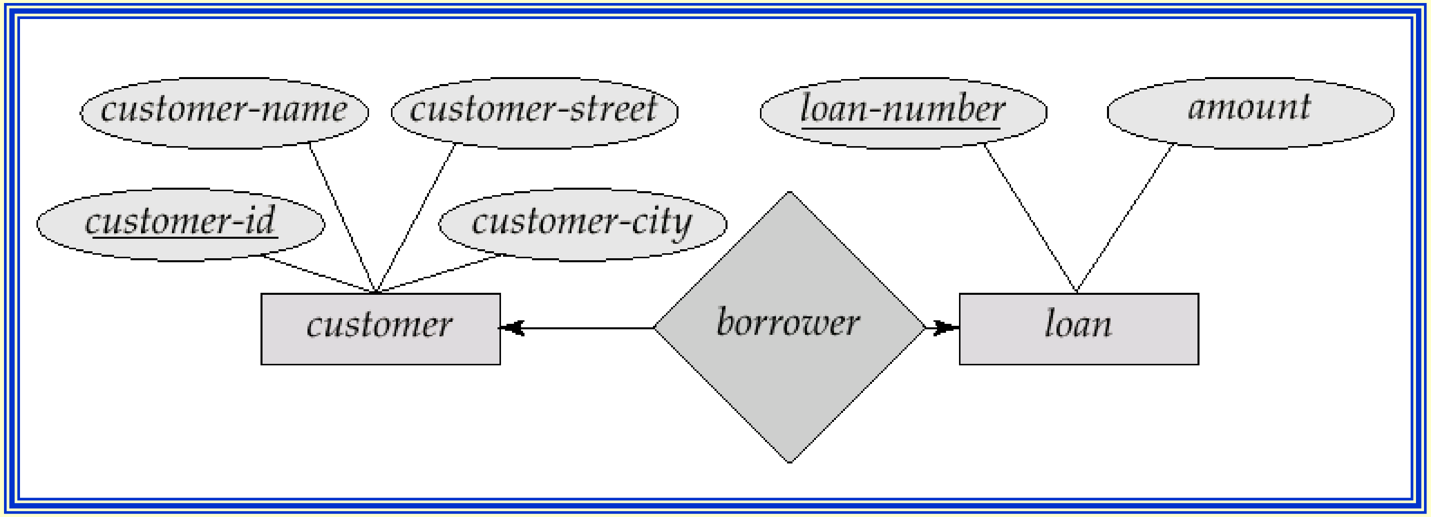


Fig: E-R diagram corresponding to *customers* and *loans*

- To distinguish the type of the relationships, we draw either a directed line (→) or an undirected line (—) between the relationship set and the entity set. Directed line indicates *one* and undirected line indicates *many*.
- Hence, the figure above has many-to-many relationship.

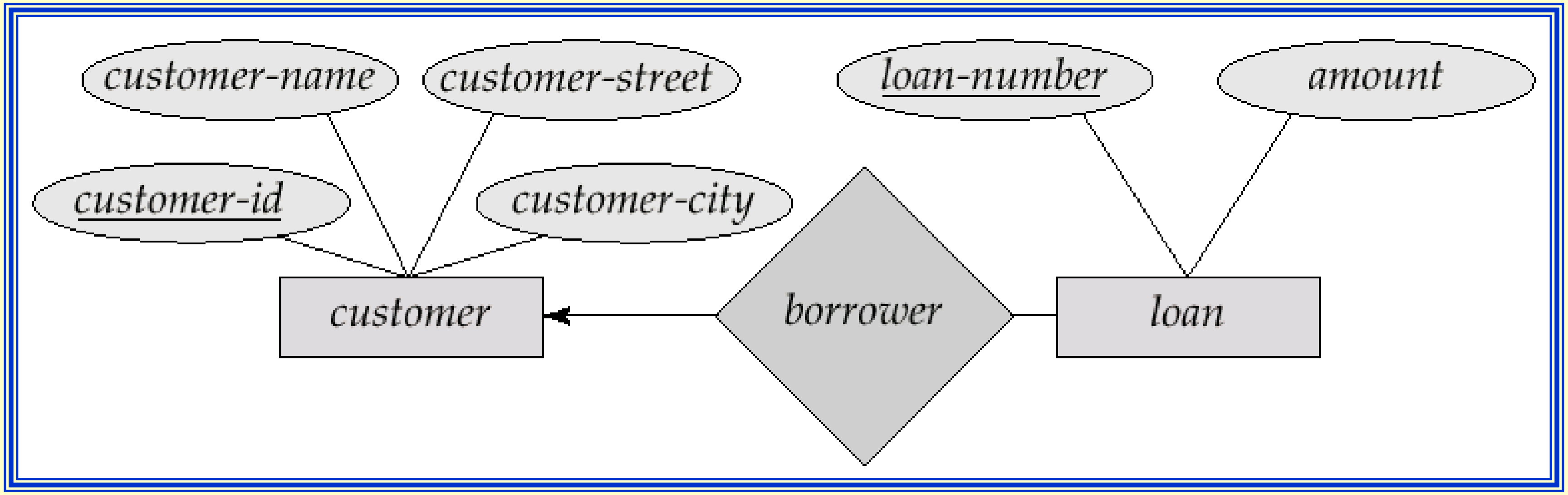
Entity-Relationship Diagram

- **One-to-one relationship:** Suppose a customer is associated with at most one loan and a loan is associated with at most one customer via the relationship set *borrower.*



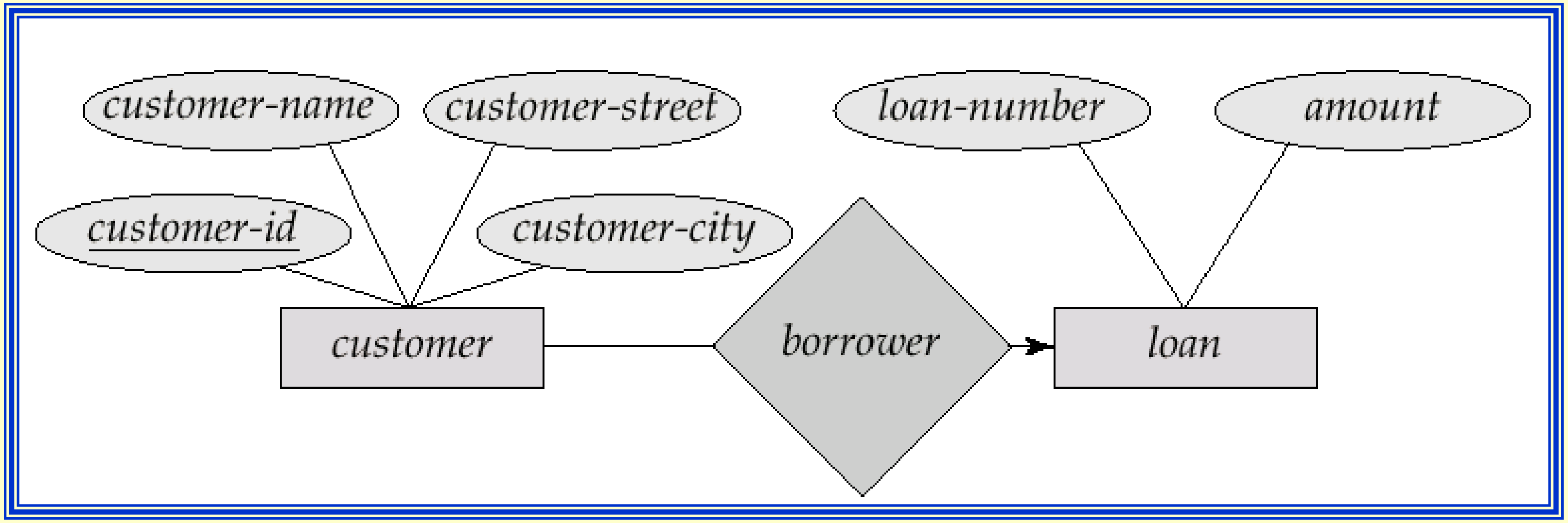
Entity-Relationship Diagram

- **One-to-many relationship:** Suppose a loan is associated with at most one customer and a customer is associated with several (including 0) loans via *borrower* relationship set.



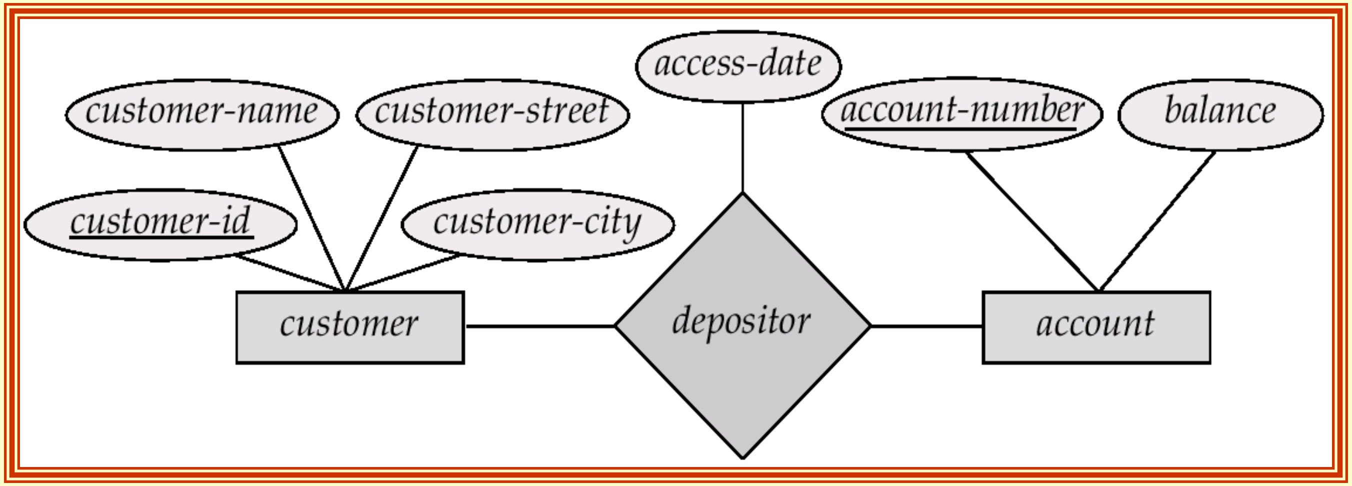
Entity-Relationship Diagram

- **One-to-many relationship:** Suppose a loan is associated with several (including 0) customers and a customer is associated with at most one loan via *borrower* relationship set.



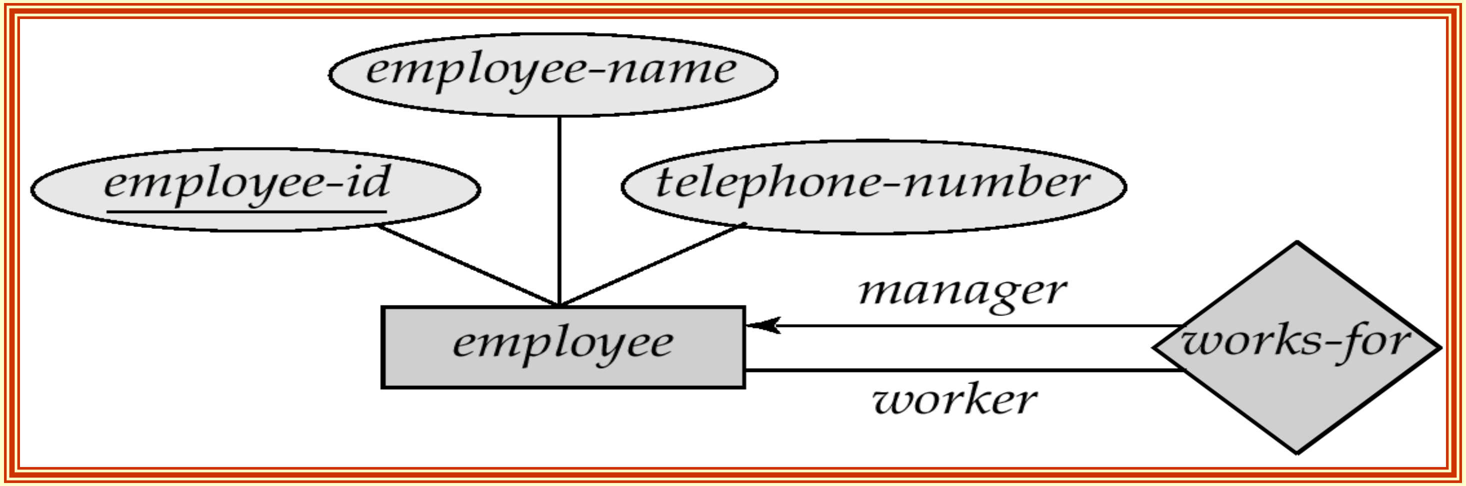
Entity-Relationship Diagram

- If a relationship set has also some attributes associated with it, then we link these attributes to that relationship set.



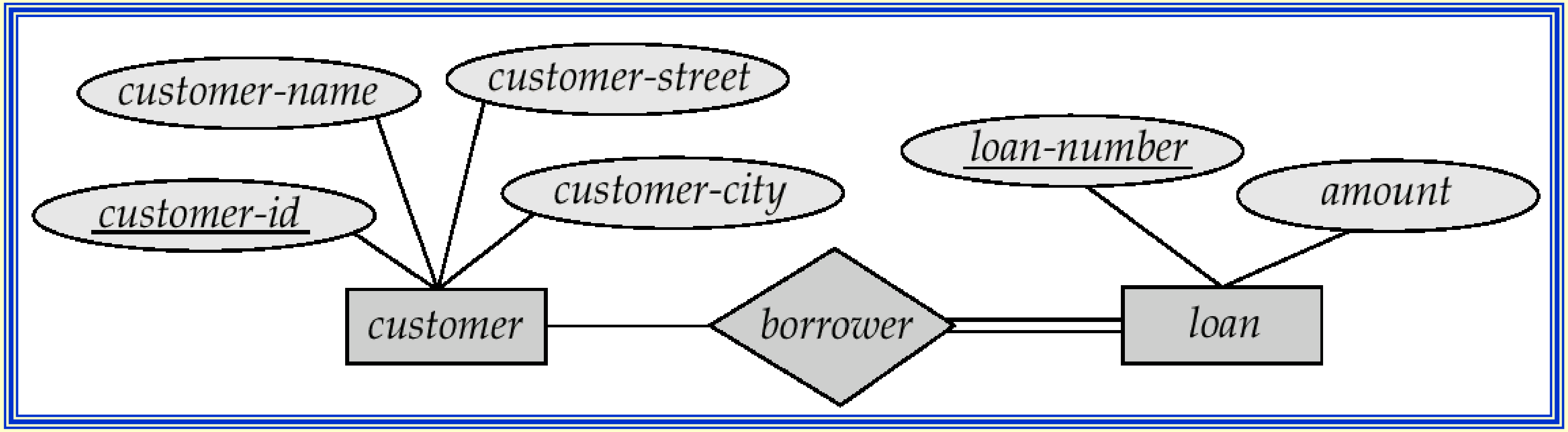
Entity-Relationship Diagram

- We indicate roles in an E-R diagram by labeling the lines that connect diamonds to rectangles.



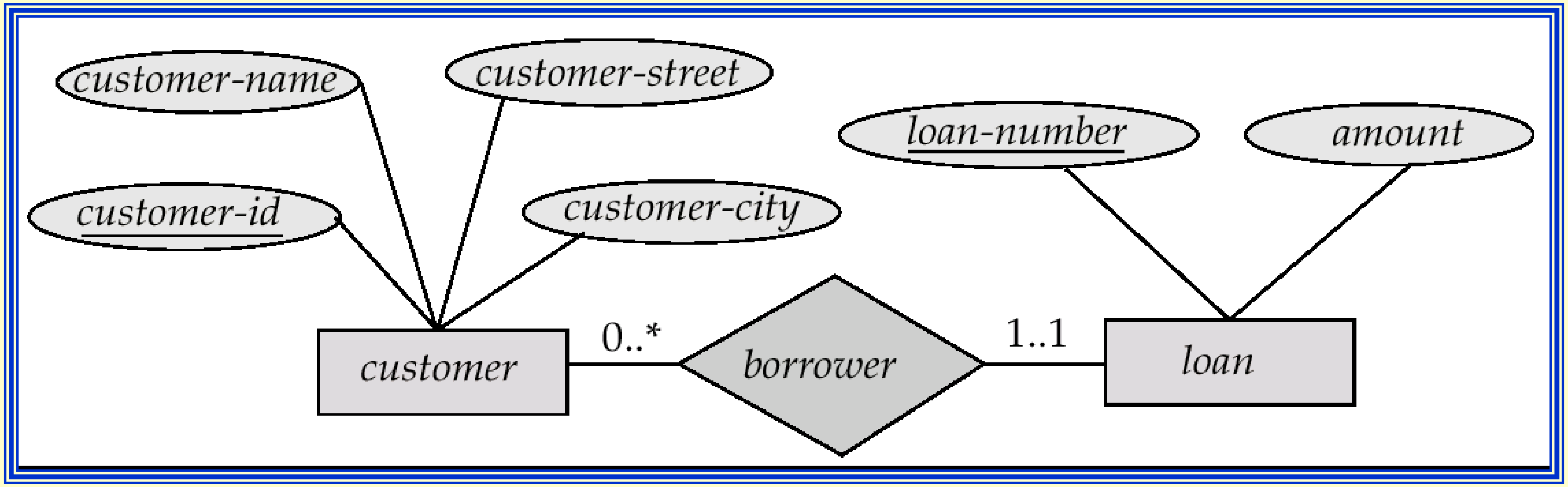
Entity-Relationship Diagram

- Double lines are used to indicate that the participation of an entity set in a relationship set is total; that is, each entity in the entity set occurs in at least one relationship in that relationship set.



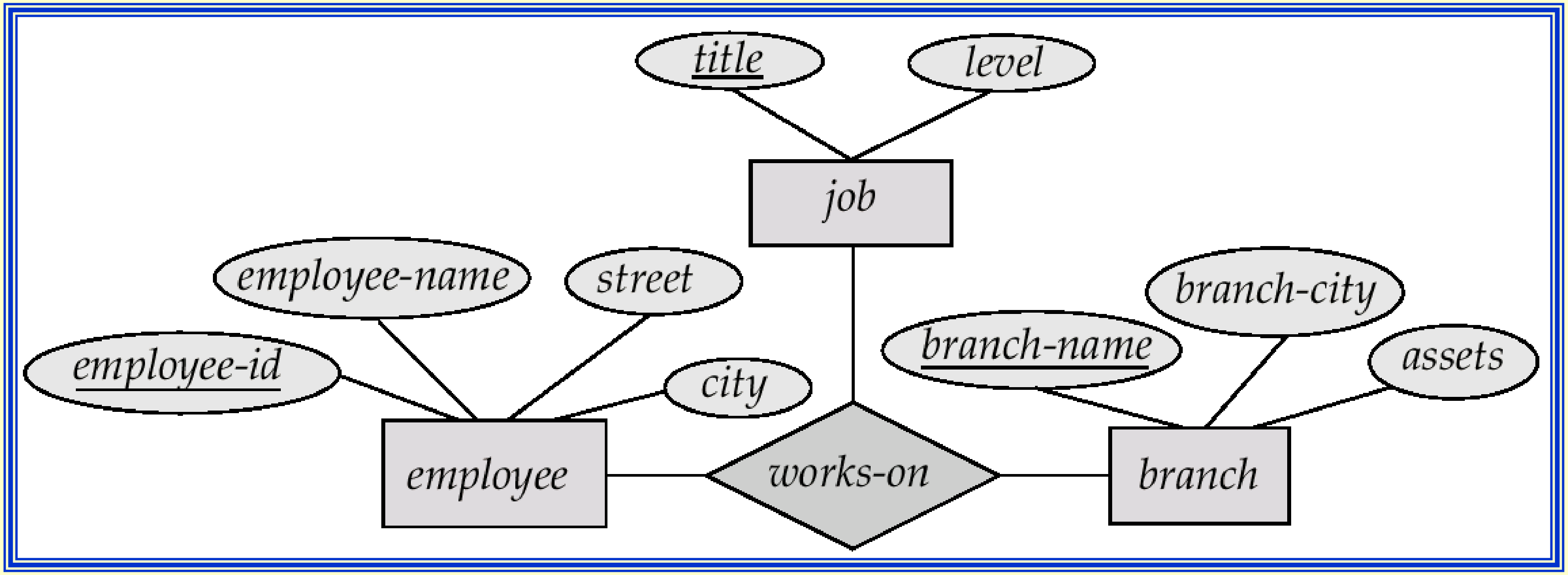
Entity-Relationship Diagram

- E-R diagram also provide a way to indicate more complex constraints on the number of times each entity participates in relationships in a relationship set. An edge between an entity set and a binary relationship set can have an associated minimum and maximum cardinality, shown in the form of L….H, where L is the minimum and H is maximum cardinality.



Entity-Relationship Diagram

- Non-binary relationship sets can also be specified easily in an E-R diagram.

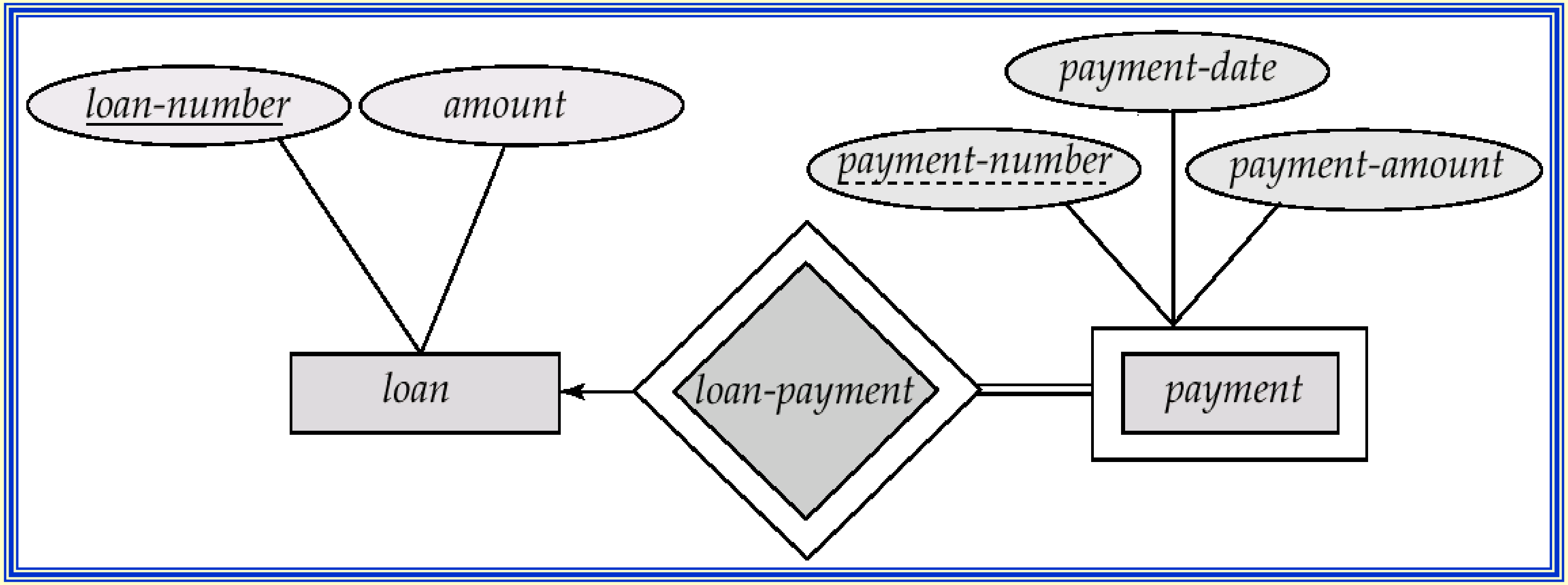


Entity-Relationship Diagram

- **Weak entity set:**
  - An entity set may not have sufficient attributes to form a primary key. Such an entity set is termed as a **weak entity set**. An entity set that has a primary key is termed as a **strong entity set**.
  - For a weak entity set to be meaningful, it must be associated with another entity set, called the **identifying** or **owner entity** set, using one of the key attribute of owner entity set. The weak entity set is said to be **existence dependent** on the identifying entity set. The relationship associating the weak entity set with the identifying entity set is called the **identifying relationship**. The identifying relationship is many-to-one form the weak entity set to the identifying entity set, and the participation of the weak entity set in the relationship set is total.
  - Although a weak entity set does not have a primary key, we use **discriminator** (or **partial key**) as a set of attributes that allows the distinction to be made among all the entities in the weak entity set.

Entity-Relationship Diagram

- In the figure below, *payment-number* is partial key and (*loan-number*, *payment-number*) is primary key for *payment* entity set.



EER(Extended ER) Modeling

- It includes all of the modeling concepts of the ER model, in addition it includes the concepts of **subclass** and **superclass** and the related concepts of **specialization** and **generalization**.
- **Subclass and Superclass:**
  - An entity set may have a number of sub-groupings of its entities that are meaningful. Here, the entity set is called *superclass* while the subgroupings are known as *subclasses* of the *superclass*, that is, a *subclass* **S** is a class whose entities must always be a subset of the entities in another class, called *superclass* **C**.
  - We call the relationship between a *superclass* and any one of its *subclass* a **superclass/subclass** or simply **class/subclass** relationship.
  - An entity that is a member of *subclass* inherits all attributes from its *superclass*. That is, there is type inheritance in subclass. The entity also inherits all the relationships in which the *superrclass* participates.

EER(Extended ER Modeling)

- **Specialization:**
  - It is a **top-down design** process. Here, we define a set of subclasses of a given, superclass entity set.
  - The set of sublcasses that form specialization is defined on the basis of some distinguishing characteristics of the entities in the superclass.
  - Specialization is depicted by a *triangle* component labeled **ISA**. For example, *customer* **"is a"** *person*.
  - The ISA relationship also referred to as **superclass-subclass** relationship.
  - Here, a lower-level entity set inherits all the attributes and relationship participation of the higher-level entity set to which it is linked.
  - The figure on the next slide shows specialization.

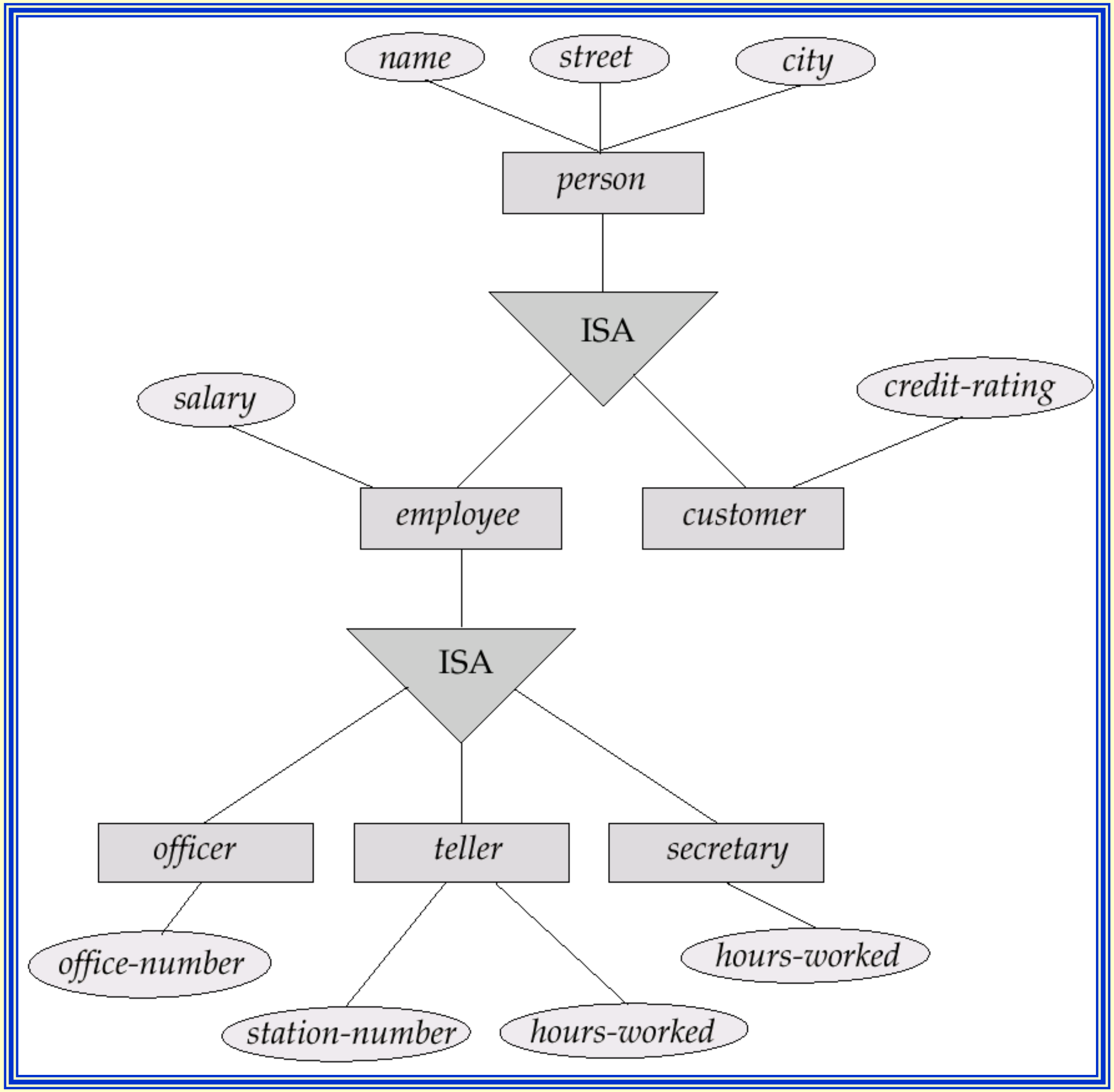


Fig: Specialization example

EER(Extended ER Modeling)

- **Generalization:**
  - It is a **bottom-up design** process. Here, we combine a number of entity sets that share the same features into a higher-level entity set.
  - The original classes become the *subclass* of the newly formed generalized *superclass*.
  - The reason, a designer applies generalization is to emphasize the similarities among the entity sets and hide their differences.
  - Specialization and generalization are simple inversions of each other; they are represented in an E-R diagram in the same way.
  - The terms specialization and generalization are used interchangeably.

EER(Extended ER Modeling)

- **Design Constraints on Specialization/Generalization:**
  - Constraint on which entities can be members of a given lower-level entity set:
    - **Condition-defined:** If we can determine exactly those entities that will become members of each subclass by a condition, the subclasses are called *predicate-defined* (or condition-defined) subclasses. Here, condition is a constraint that determines subclass members. For example, all customers over 65 years are members of *senior-citizen* entity set; *senior-citizen* ISA *person*.
    - **Attribute-defined:** If all subclasses have membership condition on the same attribute of the superclass, it is called an *attribute defined-*subclass. And, the attribute is called the *defining attribute*. Example: JobType is the defining attribute of {SECRETARY, TECHNICIAN, ENGINEER} of EMPLOYEE. It is similar to condition-defined.

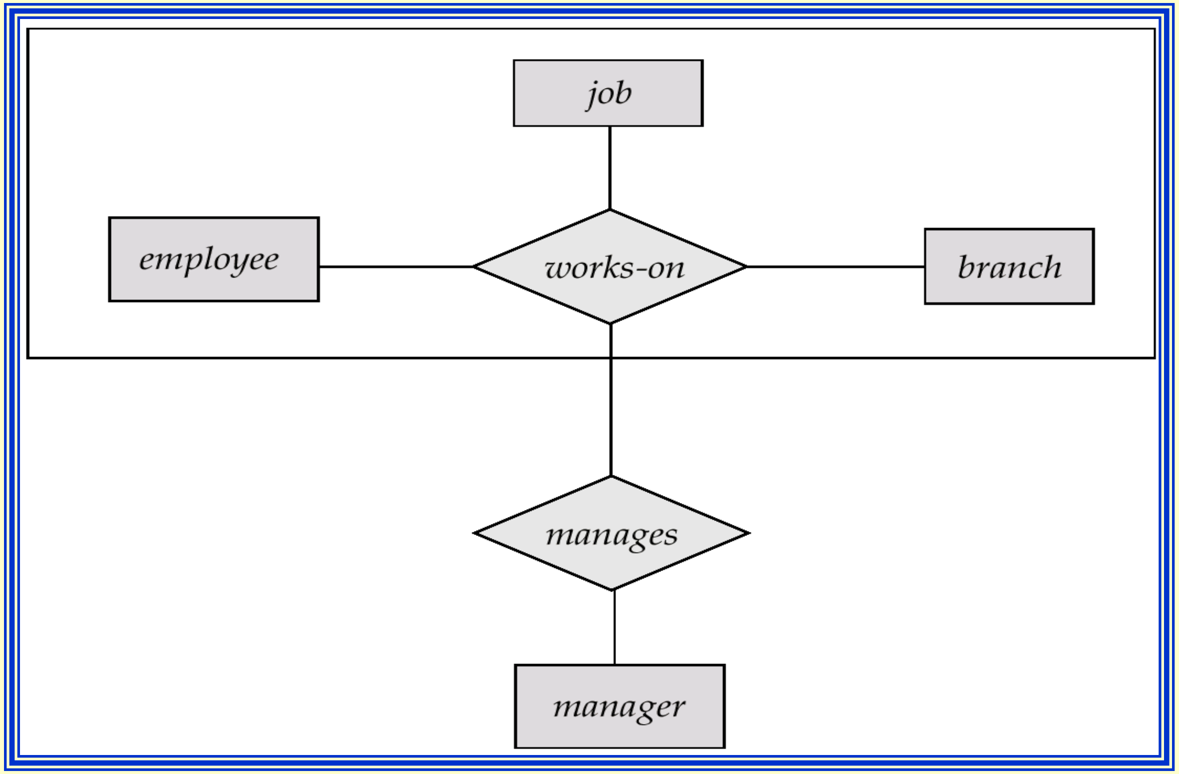
EER(Extended ER Modeling)

  - **User-defined:** If no condition determines membership, the subclass is called *user-defined.* Membership in a subclass is determined by the database users by applying an operation to add an entity to the subclass. Membership in the subclass is specified individually for each entity in the superclass by the user.

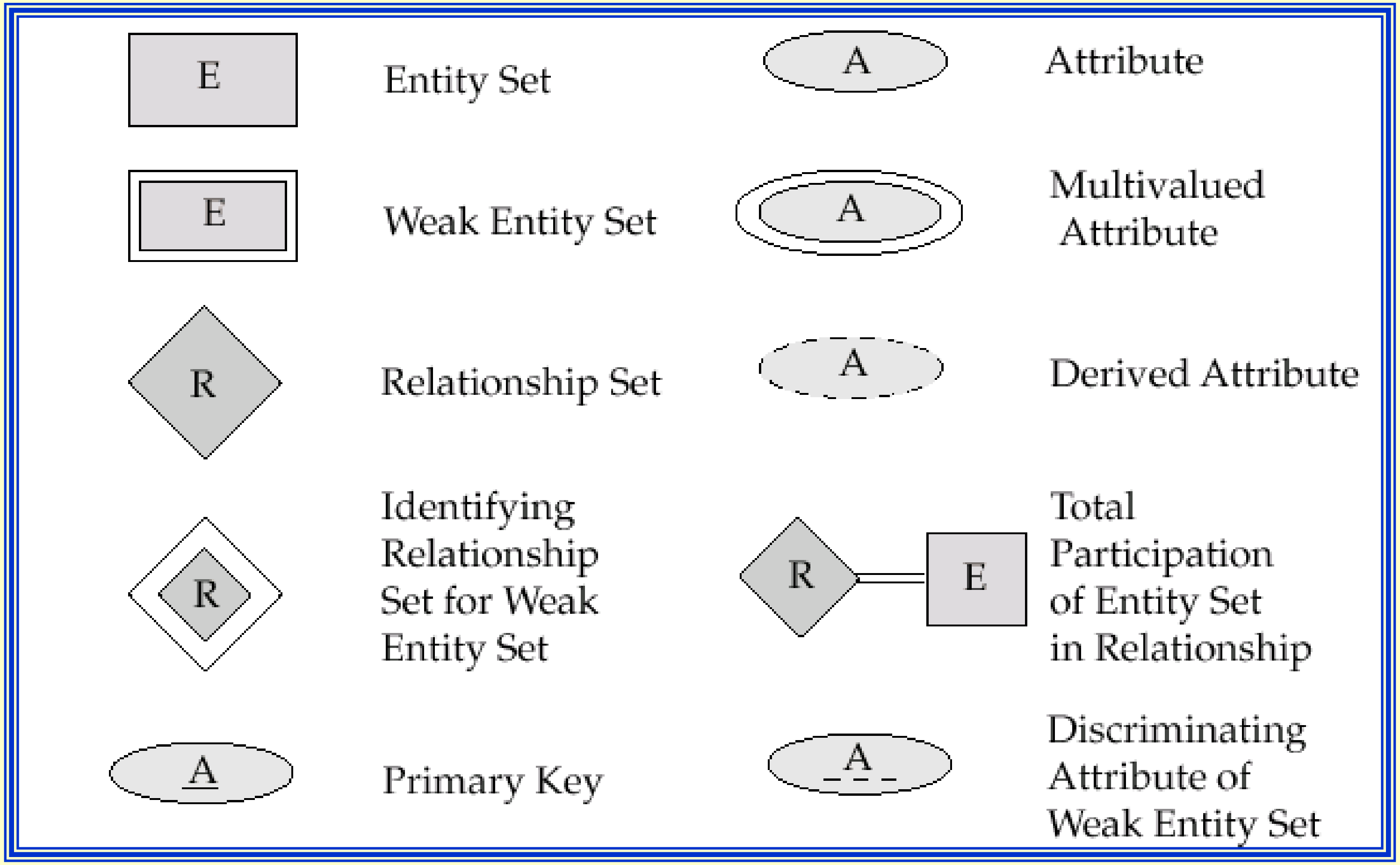
- Constraint on whether or not entities may belong to more than one lower-level entity set.
  - **Disjoint:** An entity can belong to only one lower-level entity set. Noted in E-R diagram by writing *disjoint* next to the ISA triangle.
  - **Overlapping:** An entity can belong to more than one lower-level entity set.
- Constraint that specifies whether or not an entity in the higher-level entity set must belong to at least one of the lower-level entity sets.
  - **Total** : An entity must belong to one of the lower-level entity sets
  - **Partial**: An entity need not belong to one of the lower-level entity sets.

EER(Extended ER Modeling)

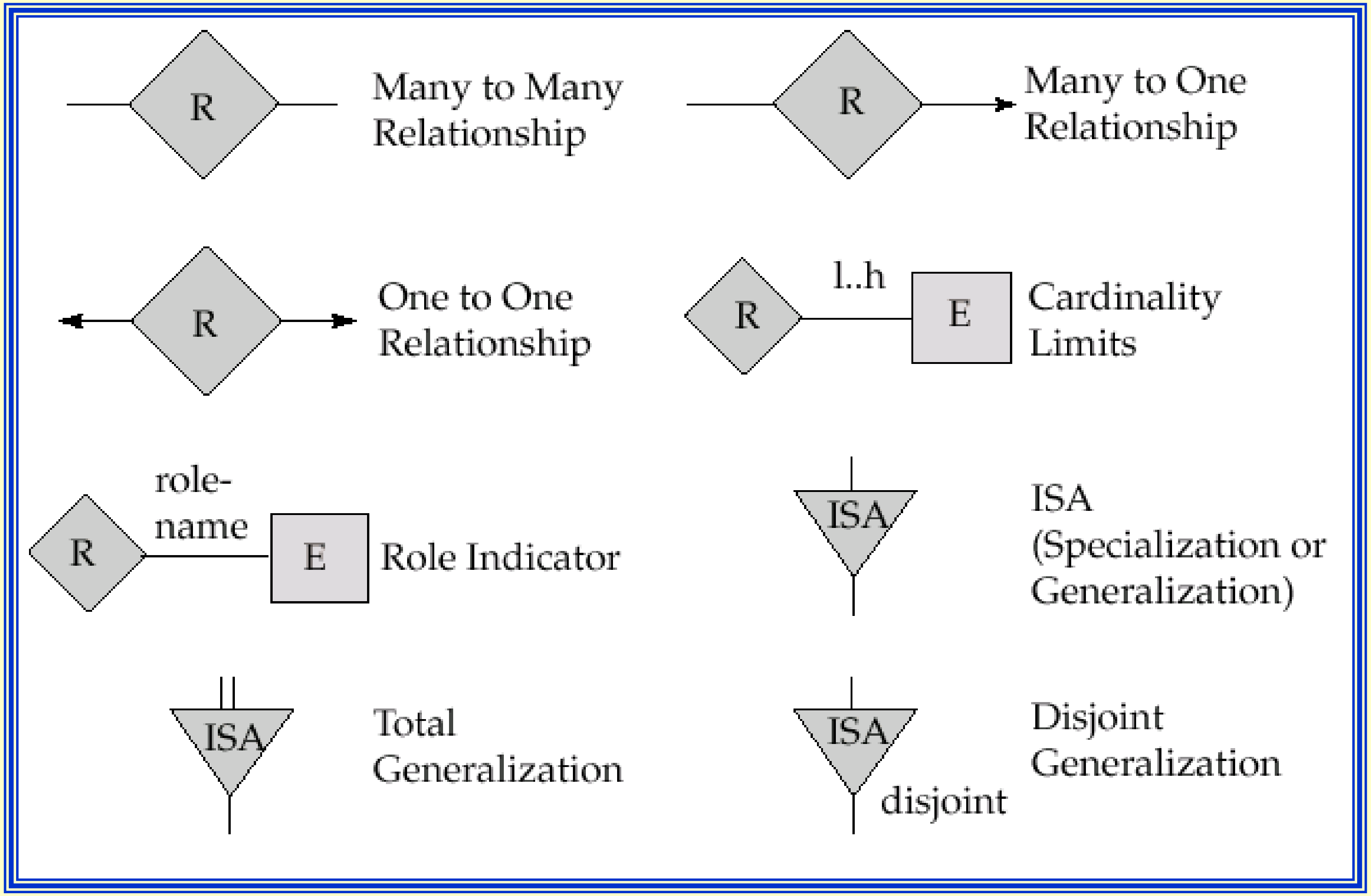
- **Aggregation**
  - Aggregation is an abstraction through which relationship sets are treated as high-level entity sets and can participate in relationship sets.
  - Allows relationships between relationships.



Summary of Symbols in ER Diagram


E
Entity Set
A
Attribute
E
Weak Entity Set
A
Multivalued Attribute
R
Relationship Set
A
Derived Attribute
R
Identifying Relationship Set for Weak Entity Set
R
E
Total Participation of Entity Set in Relationship
A
Primary Key
A
Discriminating Attribute of Weak Entity Set

Summary of Symbols in ER Diagram



Many to Many Relationship

Many to One Relationship

One to One Relationship

Cardinality Limits

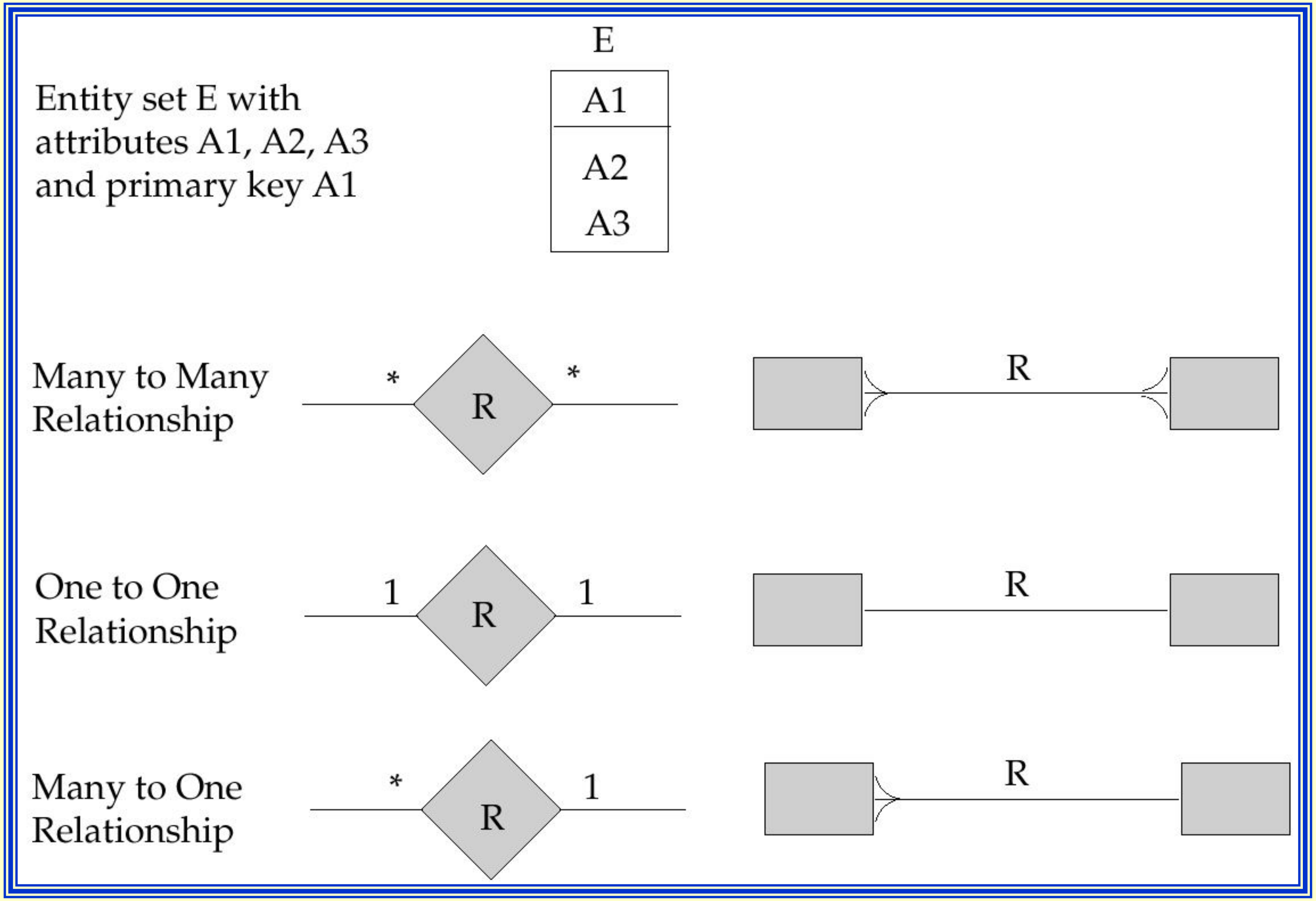
Role Indicator

ISA (Specialization or Generalization)

Total Generalization

Disjoint Generalization

Alternative ER Notations



Entity set E with attributes A1, A2, A3 and primary key A1

Many to Many Relationship

One to One Relationship

Many to One Relationship

Design Issues

- It is possible to define a set of entities and the relationships among them in a number of ways.

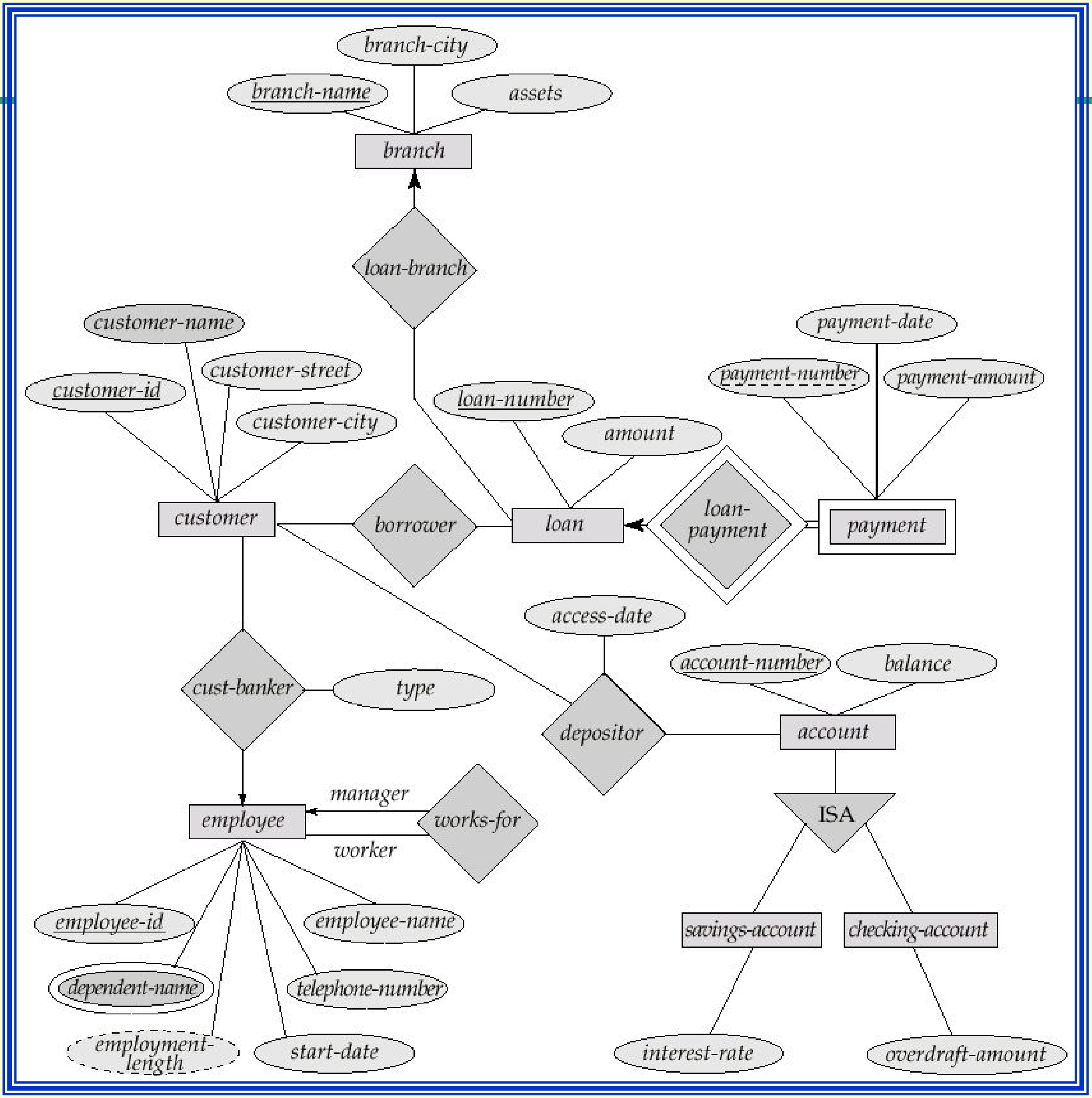
1. **Use of Entity Sets versus Attributes:** The choice of entity sets and attributes mainly depend on the structure of the real world enterprise being modeled, and on the semantics associated with the attribute in question. Some guidelines include:
   - Do not use the primary key on an entity set as an attribute of another entity set, instead of using a relationship.
   - Do not use the primary key attributes of the related entity sets as attributes of the relationship set.
2. **Use of Entity Sets versus Relationship Sets:** Possible guideline in determining whether to use an entity set or a relationship set is to designate a relationship set to describe an action that occurs between entities.

Design Issues

3. **Binary versus n-ary Relationship Sets:** Although it is possible to replace any nonbinary (n-ary, n>2) relationship set by a number of distinct binary relationship sets, a n-ary relationship set shows more clearly that several entities participate in a single relationship.

4. **Placement of Relationship Attributes:** The cardinality ratio of a relationship can affect the placement of relationship attributes. Thus, attributes of *one-to-one* or *one-to-many* relationship sets can be associated with one of the participating entity sets, rather than with the relationship set. Attributes of a *one-to-many* relationship set can be repositioned to only the entity set on the "many" side of the relationship. For *one-to-one* relationship sets, the relationship attribute can be associated with either one of the participating entity sets.

Example: ER Diagram for Banking Enterprise



Exercises

1. Construct an ER diagram for a car-insurance company whose customers own one or more cars each. Each car has associated with it zero to any number of recorded accidents.
2. Construct an ER diagram for a hospital with a set of patients and a set of doctors. Associate with each patient a log of the various tests and examinations conducted.
3. Construct an ER diagram of the library system in your college.
4. Construct an ER diagram to maintain data about students, instructors, semester, and courses in a college.
5. Construct an ERD to record the marks that students get in different exams of different course offerings.