Database Engineering

Lecture #3

Entity-Relationship Model

Presented By:
Dr. Suvasini Panigrahi
Associate Professor, Department of CSE,
VSSUT, Burla

CONTENTS

- E-R Modeling
- Constraints
- E-R Diagram
- Design Issues
- Weak Entity Sets
- Extended E-R Features

E-R Modeling

- A *database* can be modeled as:
  - a collection of entities
  - relationship among entities
- An **entity** is an object that exists and is distinguishable from other objects.
  - Example: specific person, company, event, plant, student, employee, etc.
- Entities have *attributes*
  - Example: people have *names* and *addresses*
- An **entity set** is a set of entities of the same type that share the same properties.
  - Example: set of all persons, companies, trees, holidays, etc.

Entity Sets: *customer* and *loan*

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

Relationship Sets

- A **relationship** is an association among several entities

  Example:

  Hayes *depositor* A-102

  customer entity relationship set *account* entity

- A **relationship set** is a mathematical relation among $n \geq 2$ entities, each taken from one entity set

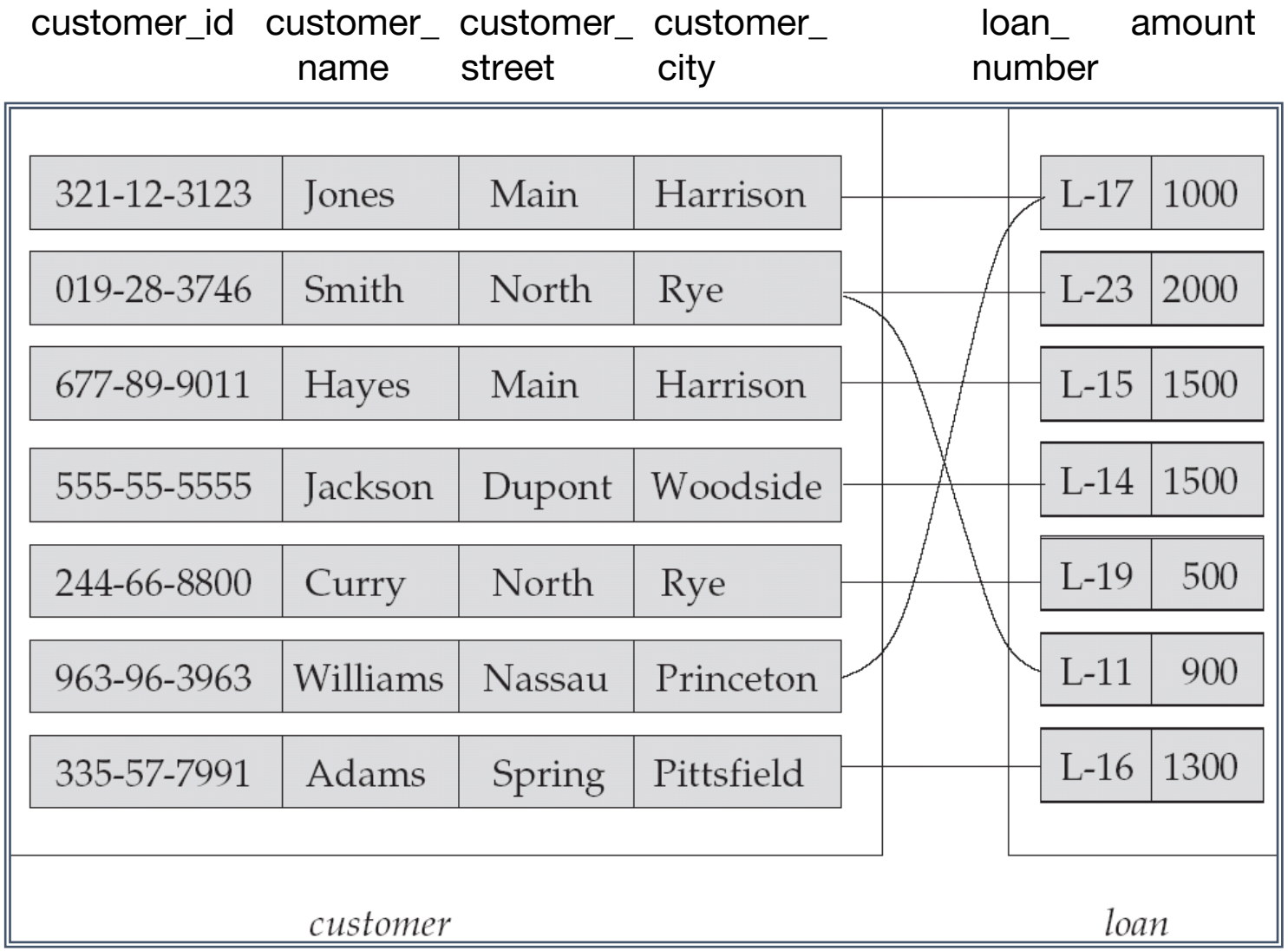
$$\{(e_1, e_2, \ldots e_n) \mid e_1 \in E_1, e_2 \in E_2, \ldots, e_n \in E_n\}$$

  where $(e_1, e_2, \ldots, e_n)$ is a relationship

  - Example:

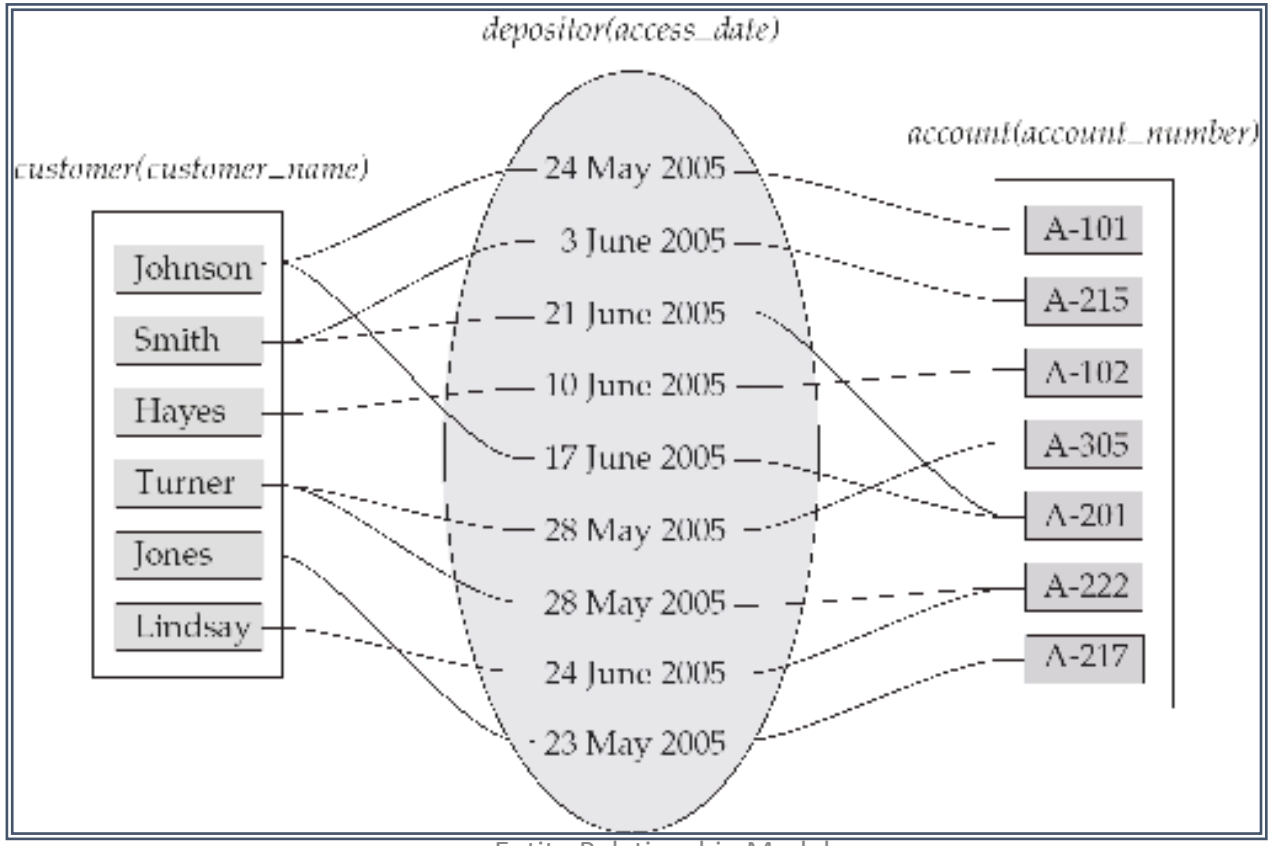
    (Hayes, A-102) $\in$ *depositor*

Relationship Set - *borrower*



Relationship Sets (Cont.)

- An **attribute** can also be property of a relationship set.
- For instance, the *depositor* relationship set between entity sets *customer* and *account* may have the attribute *access-date*



Degree of a Relationship Set

- Degree refers to the number of entity sets that participate in a relationship set.
- Relationship sets that involve two entity sets are **binary** (or degree two).
- Generally, most relationship sets in a database system are binary.
- Relationship sets may involve more than two entity sets.
    4. 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*
- Relationships between more than two entity sets are rare. Most relationships are binary.

Attributes

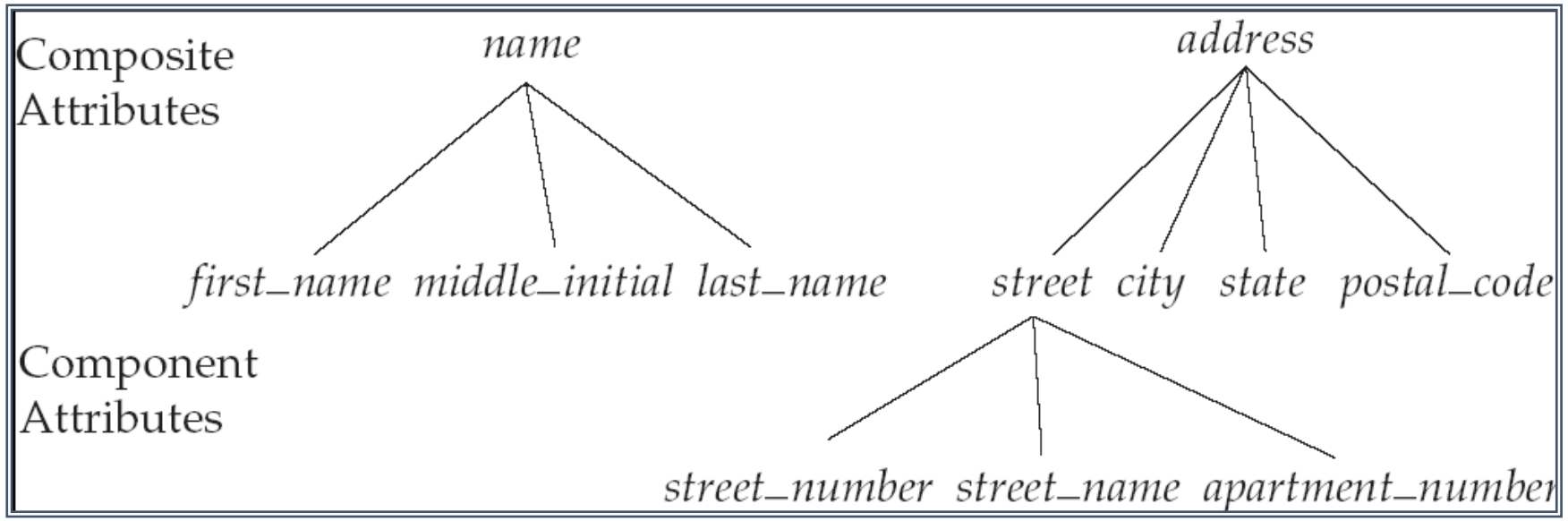
- An entity is represented by a set of attributes, that is descriptive properties possessed by all members of an entity set.

  Example:

  customer = (*customer_id, customer_name,*
  customer_street, customer_city)
  loan = (*loan_number, amount*)

- **Domain** – the set of permitted values for each attribute
- **Attribute types:**
  - ***Simple*** and ***composite*** **attributes**
  - ***Single-valued*** and ***multi-valued*** **attributes**
    - Example: multivalued attribute: *phone_number*
  - ***Base Attribute*** and ***Derived*** **attributes**
    - Derived attribute can be computed from another base attribute
    - Example: age (derived attribute) given date_of_birth (base attribute)

Composite Attributes

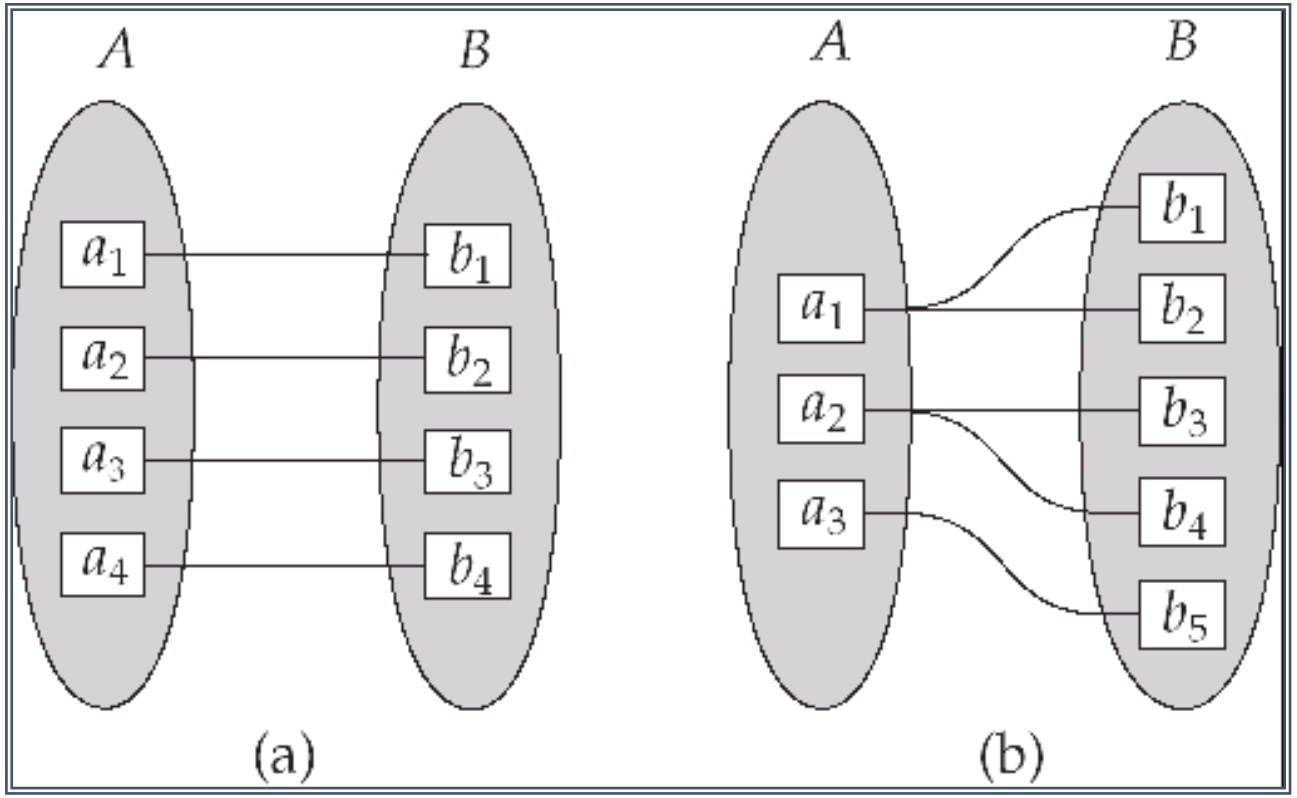


Mapping Cardinality Constraints

- In a database, the **mapping cardinality/cardinality ratio/ cardinality constraints** expresses the number of entities to which another entity set can be linked through a certain relationship set
- It is most useful in describing binary relationship sets.
- For a binary relationship set, the mapping cardinality must be one of the following types:
  - One-to-one
  - One-to-many
  - Many-to-one
  - Many-to-many

Mapping Cardinalities

One-to-one: In this type of cardinality mapping, an entity in A is connected to at most one entity in B.



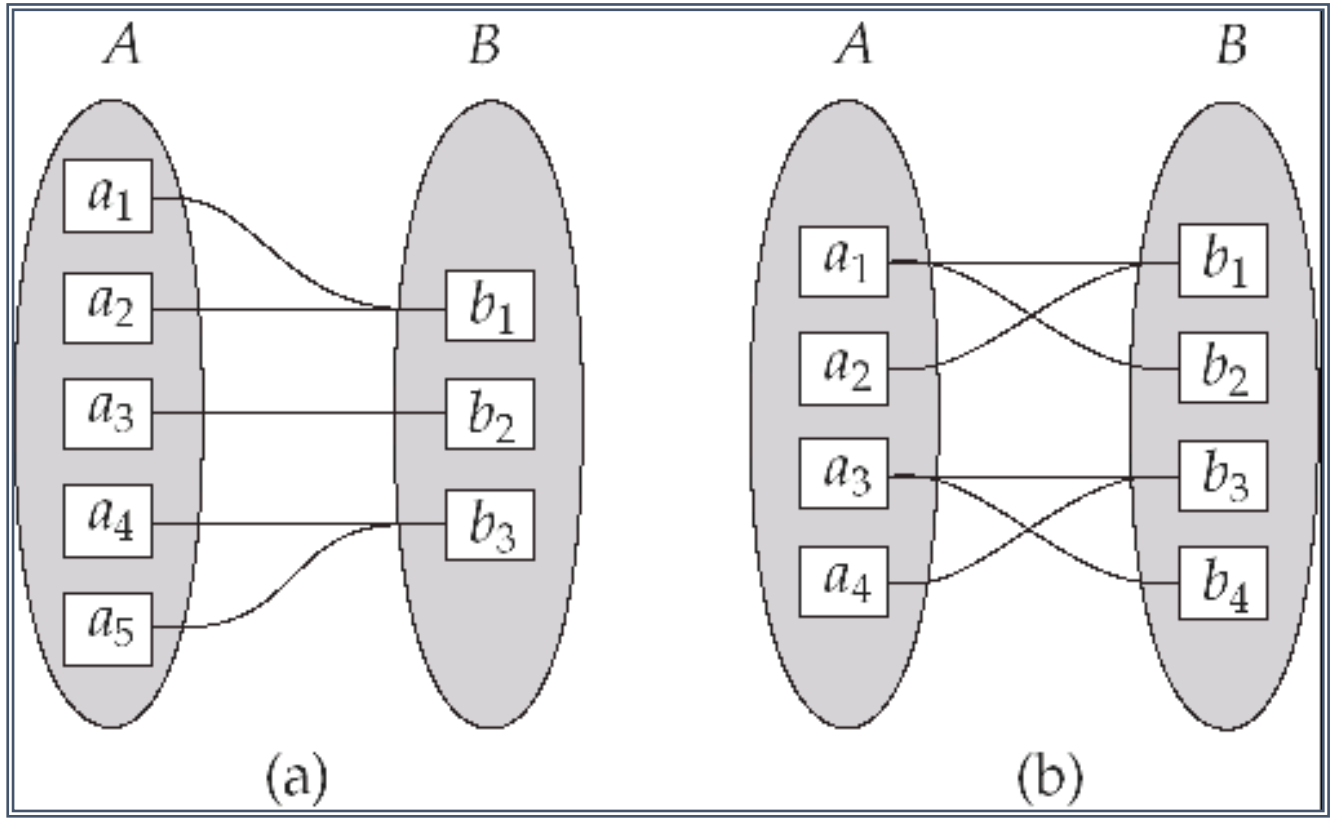
One-to-many: In this type of cardinality mapping, an entity in A is associated with any number of entities in B.

One-to-one **One-to-many**

Note: Some elements in *A* and *B* may not be mapped to any elements in the other set

Mapping Cardinalities

Many-to-one: In this type of cardinality mapping, an entity in A is connected to at most one entity in B. But, an entity in B can be associated with any number (zero or more) of entities in A.



Many-to-many: In this type of cardinality mapping, an entity in A is associated with any number of entities in B, and an entity in B is associated with any number of entities in A.

Many-to-one **Many-to-many**

Note: Some elements in A and B may not be mapped to any elements in the other set

Keys

- A **super key** of an entity set is a set of one or more attributes whose values uniquely determine each entity.
- For example, in a "Students" table, a super key could be a combination of the "Student ID" and "Email" columns, as it guarantees uniqueness for each row.
- A **candidate key** of an entity set is a minimal super key
  - *customer_id* is candidate key of *customer*
  - *account_number* is candidate key of *account*
- Although several candidate keys may exist, one of the candidate keys is selected to be the **primary key**.

Keys for Relationship Sets

- The combination of primary keys of the participating entity sets forms a super key of a relationship set.
  - (*customer_id, account_number*) is the super key of *depositor*
  - *NOTE: this means a pair of entity sets can have at most one relationship in a particular relationship set.*
    - Example: if we wish to track all access_dates to each account by each customer, we cannot assume a relationship for each access.
- We must consider the mapping cardinality of the relationship set when deciding what are the candidate keys
- Need to consider semantics of relationship set in selecting the *primary key*, in case where more than one candidate key is there

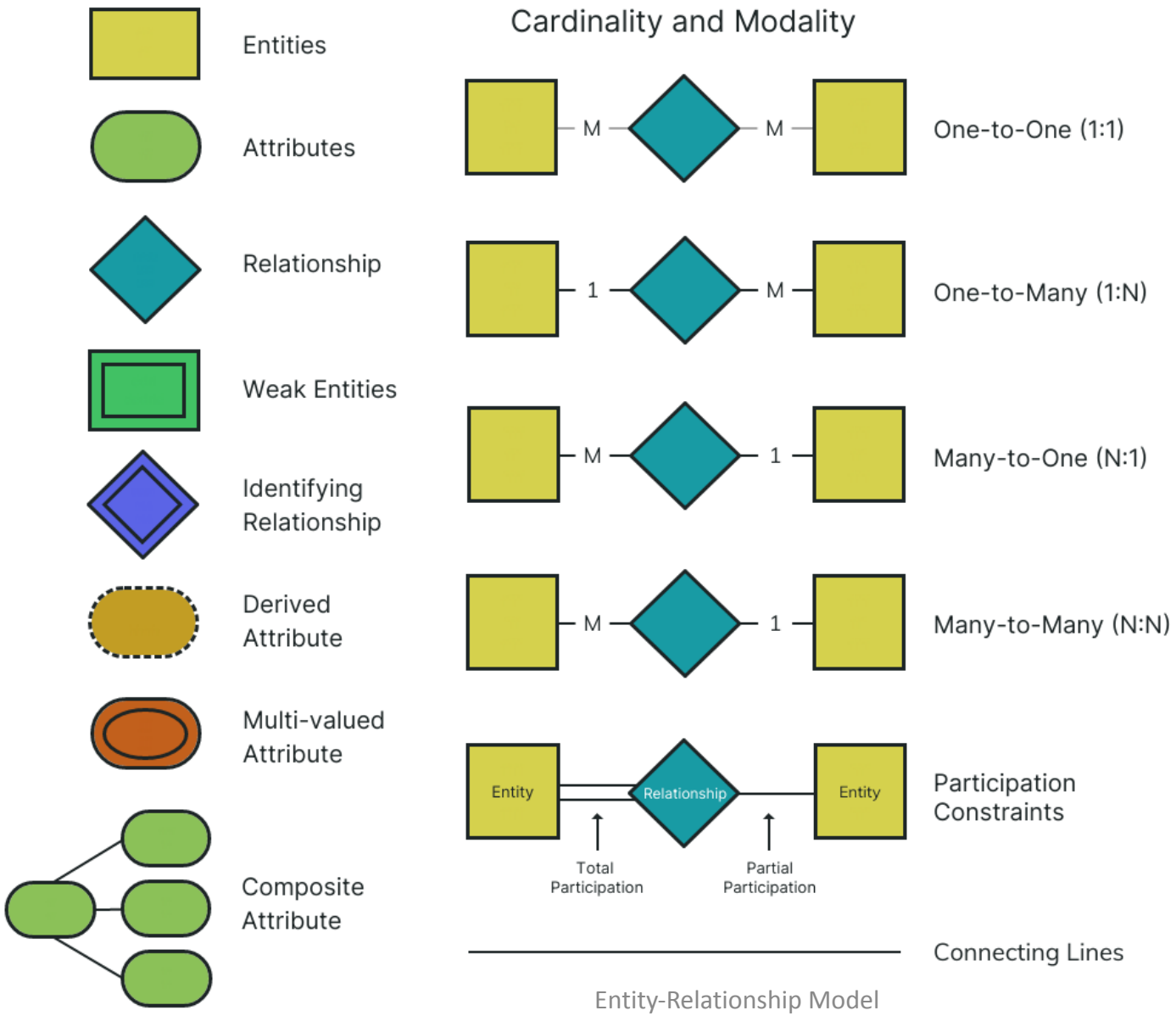
E-R (Entity-Relationship) Diagram

- The Entity-Relational Model is a model for identifying entities to be represented in the database and representation of how those entities are related
- The Entity-Relationship Diagram (ERD) is a diagrammatic representation of the real-world objects/entities like a person, a car, or a company, their attributes, the relation between these real-world objects, etc.

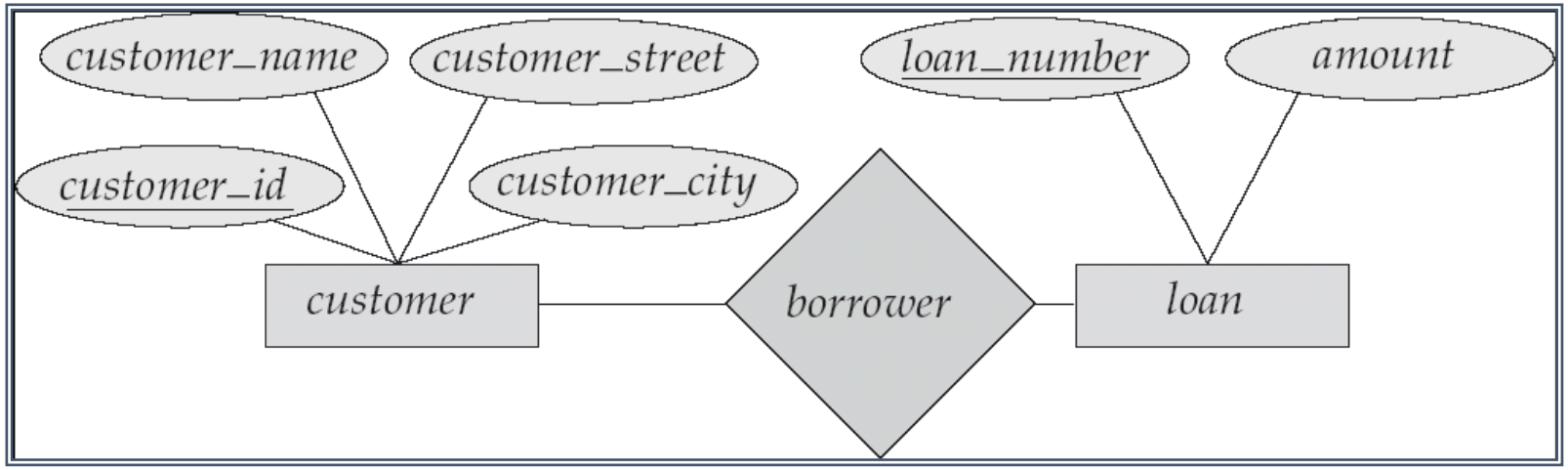
Symbols Used in ER Model

- Rectangles represent entity sets.
  - Double rectangles represent weak entity set.
- Diamonds represent relationship sets.
  - Double diamond represent identifying relationship sets.
- Ellipses represent attributes
  - Double ellipses represent multivalued attributes.
  - Dashed ellipses denote derived attributes.
- Lines link attributes to entity sets and entity sets to relationship sets.
- Underline below an attribute(s) indicates primary key attributes

ERD Symbols and Notations

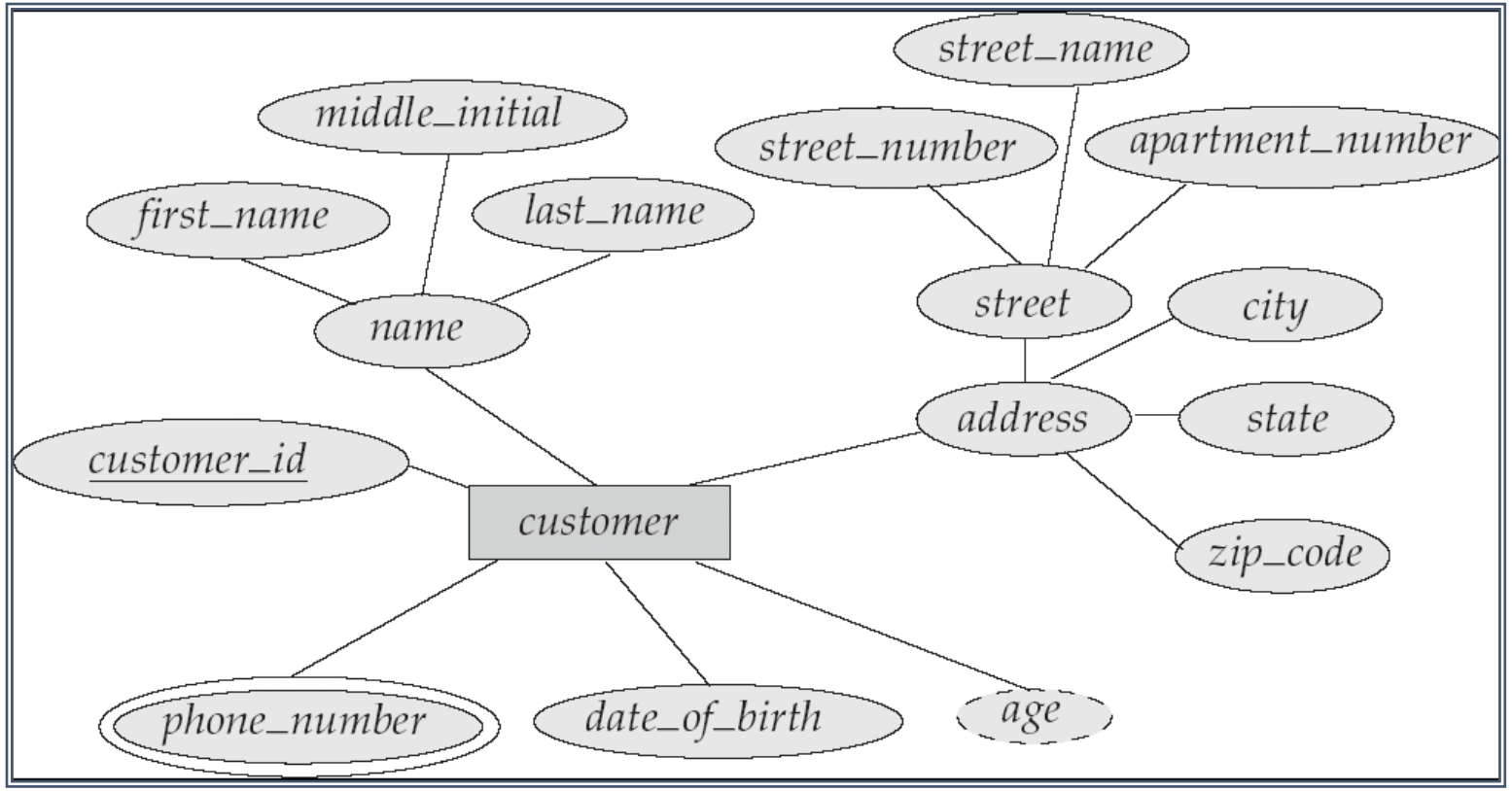


E-R diagram corresponding to entity sets customer and loan



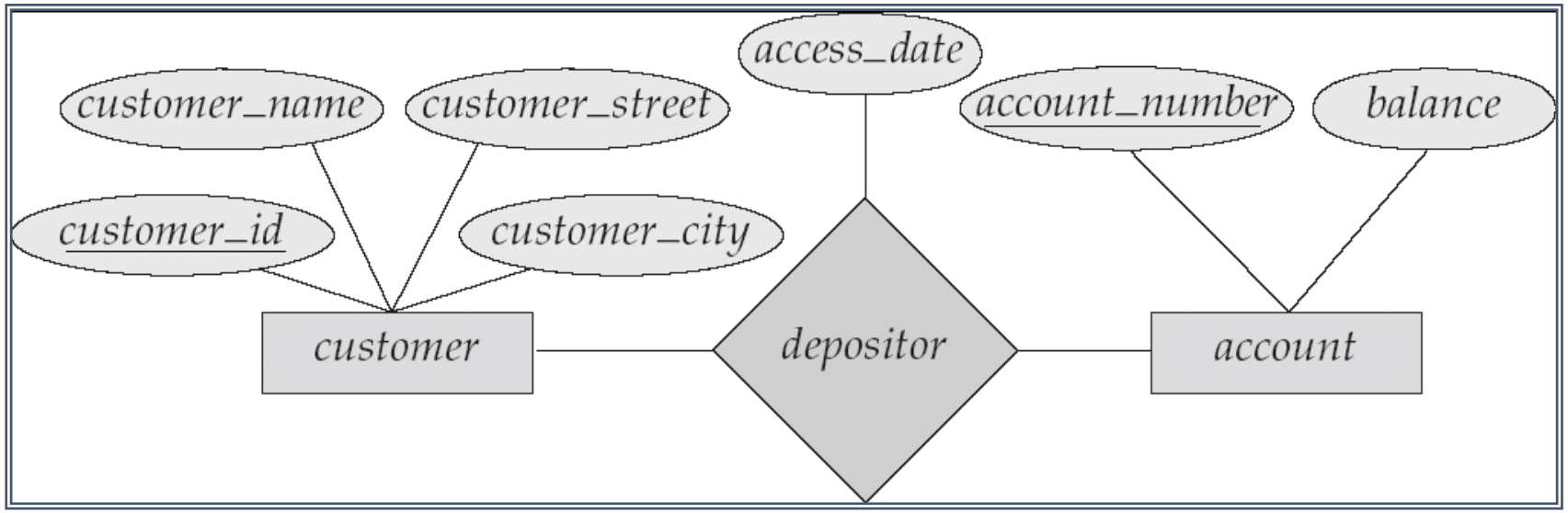
A customer entity set is associated with a loan entity set via the relationship set *borrower*

E-R Diagram With Composite, Multivalued, and Derived Attributes



In E-R model, entities have attributes which can be of various types like single-valued, multi-valued, simple, composite, stored, derived, etc.

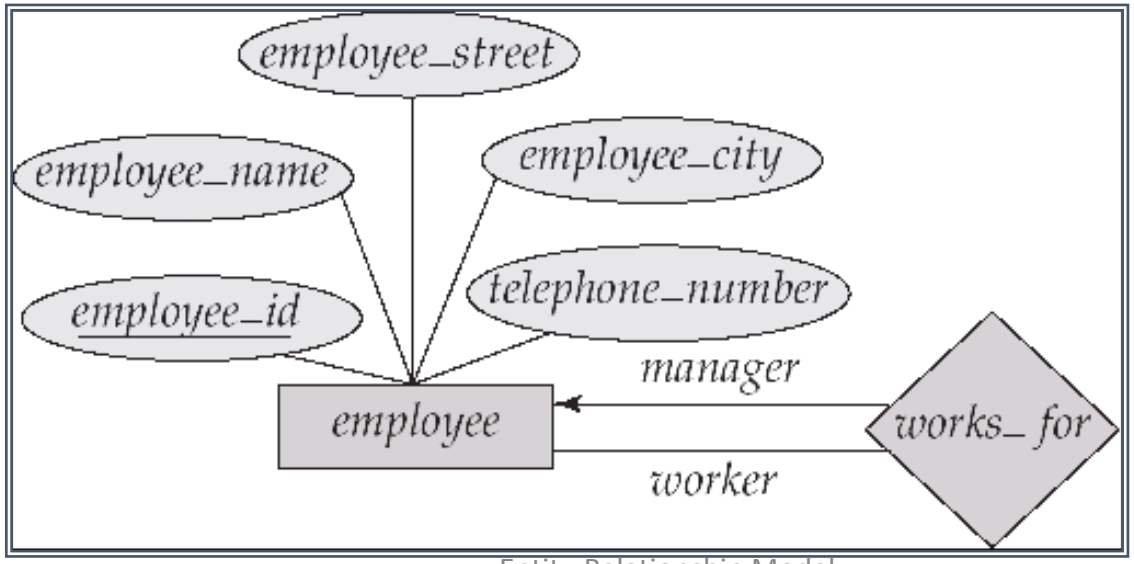
E-R Diagram having Relationship Set with Attribute(s)



Relationship set can also have one or more attributes associated with them.

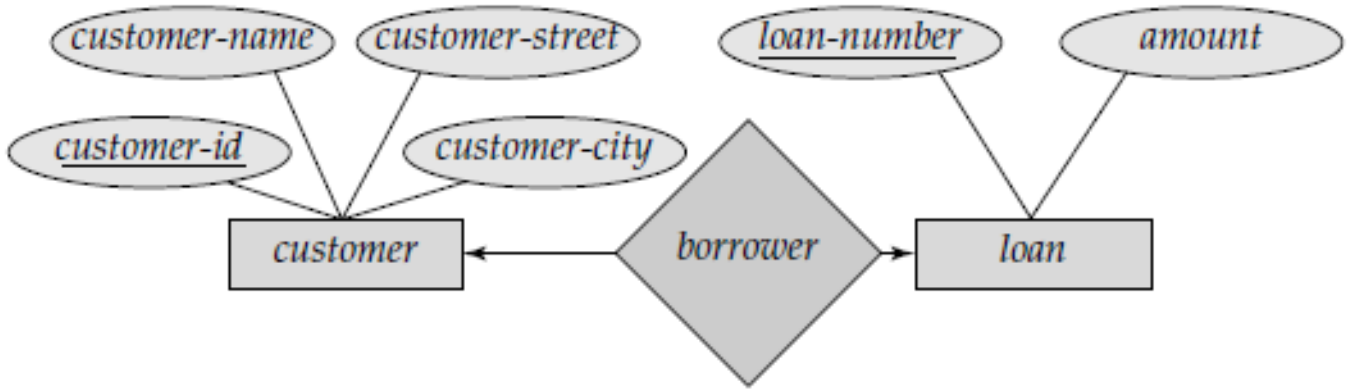
E-R Diagram having Roles

- **Role** is a function that specify how an entity is associated with a relationship
- Roles are indicated in E-R diagrams by labeling the lines that connect rectangle (entity set) to diamond (relationship set).
- Role labels are optional, and are used to clarify semantics of the relationship.
- For Example, the relationship set *works_for* defines an *employee as* manager or worker. The labels “manager” and “worker” are called **roles**; they specify how employee entities are related with the *works_for* relationship set.



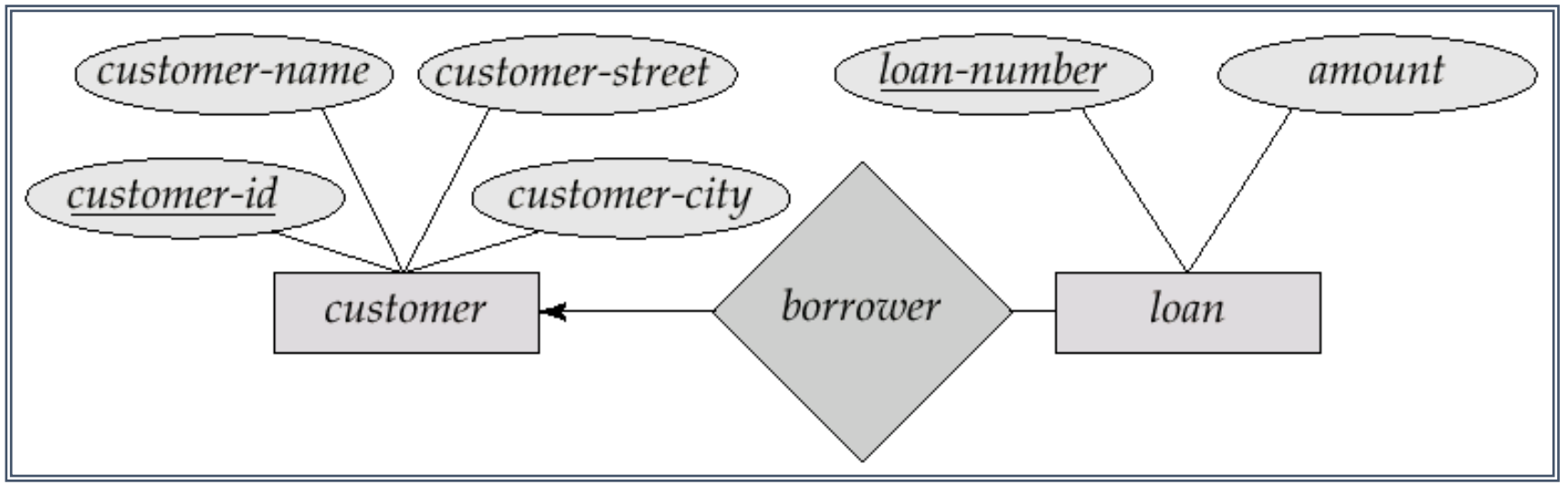
Cardinality Constraints

- We express cardinality constraints by drawing either a **directed line** (→), signifying "**one**," or an **undirected line** (—), signifying "**many**," between the relationship set and the entity set.
- One-to-one relationship:
  - In the one-to-one relationship, a customer is associated with at most one loan via the relationship *borrower*
  - A loan is also associated with at most one customer via *borrower*



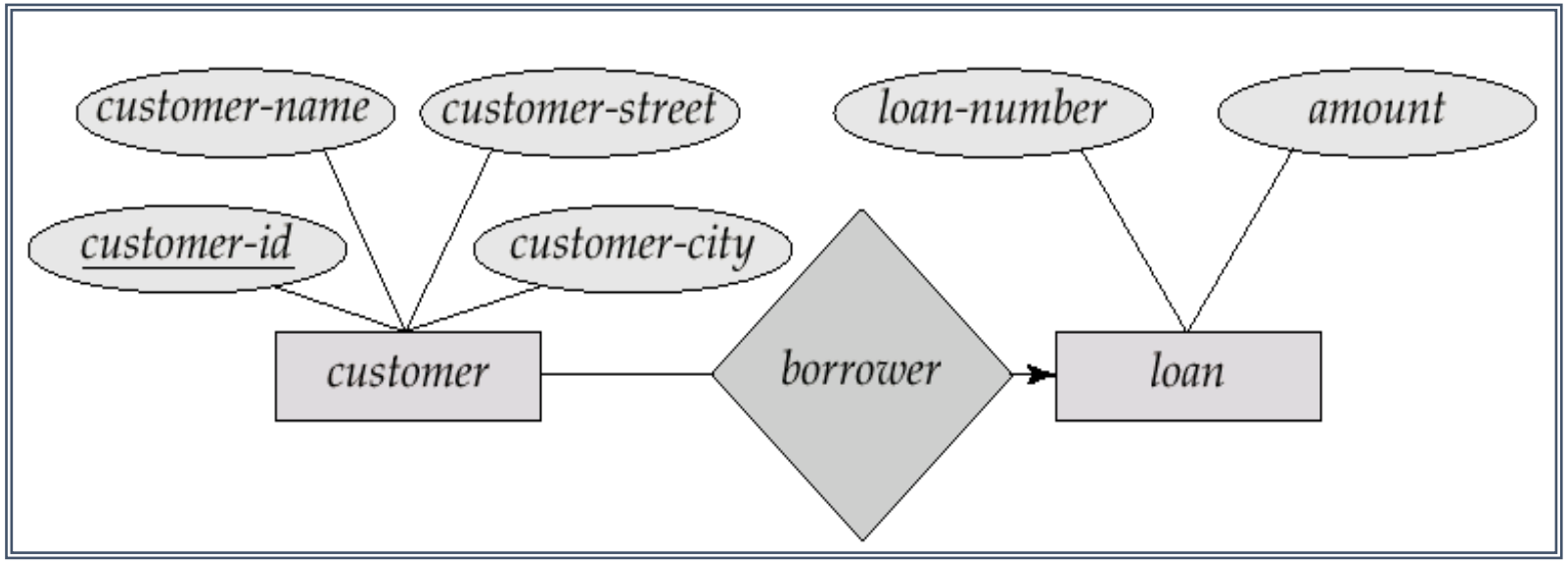
One-To-Many Relationship

- In the one-to-many relationship, a loan is associated with at most one customer via *borrower* but a customer is associated with several (including 0) loans via *borrower*



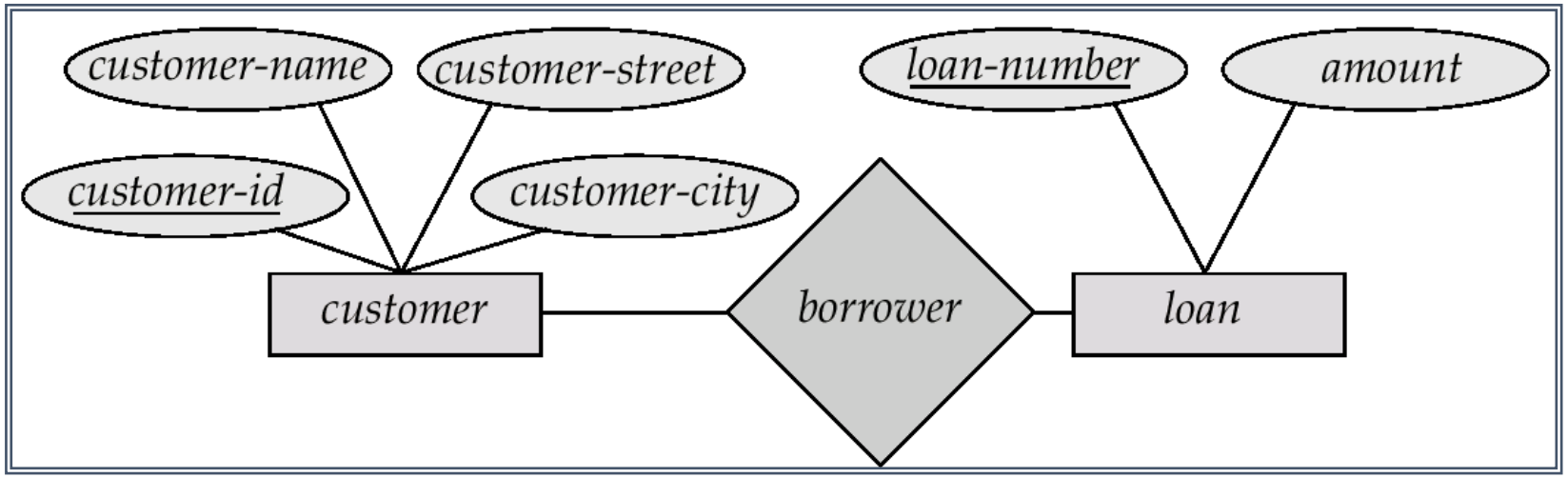
Many-To-One Relationships

- In a many-to-one relationship, a loan is associated with several (including 0) customers via *borrower*, a customer is associated with at most one loan via *borrower*



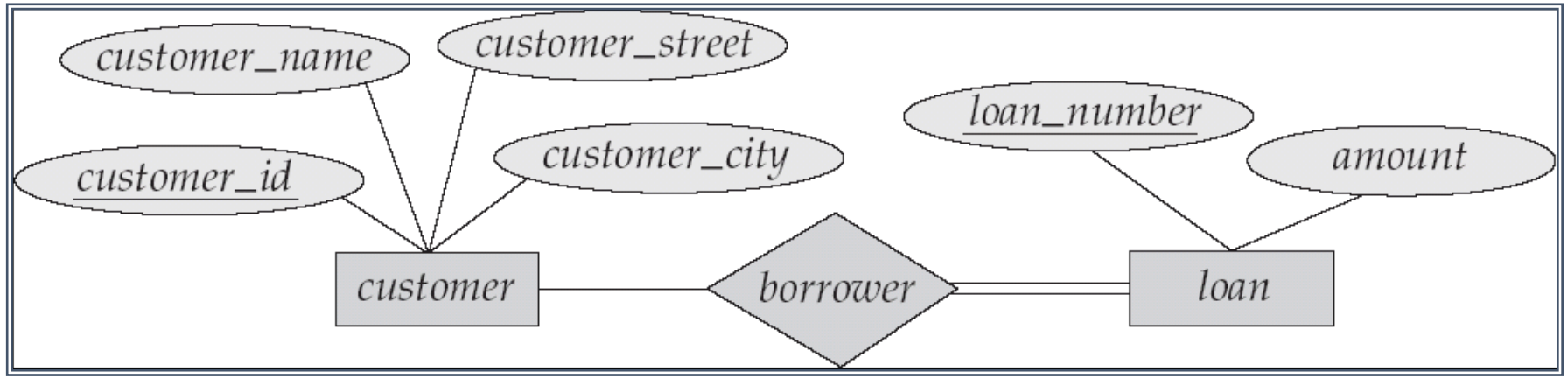
Many-To-Many Relationship

- A many-to-many relationship is a combination of one-to-many as well as many-to-one relationship
- A customer is associated with several (possibly 0) loans via borrower
- A loan is associated with several (possibly 0) customers via borrower



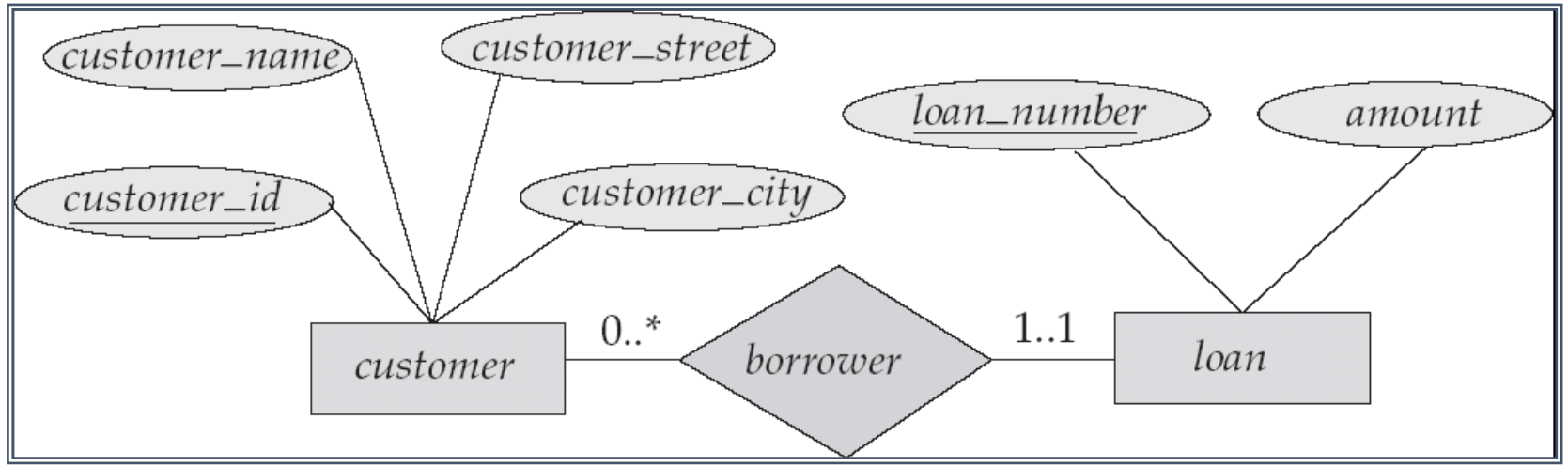
Participation of an Entity Set in a Relationship Set

- **Total participation** (indicated by double line): Every entity in the entity set participates in at least one relationship in the relationship set
  - E.g. participation of loan in borrower is total
    - every loan must have a customer associated to it via borrower
- **Partial participation** (indicated by single line): Some entities may not participate in any relationship in the relationship set
  - E.g.: participation of customer in borrower is partial

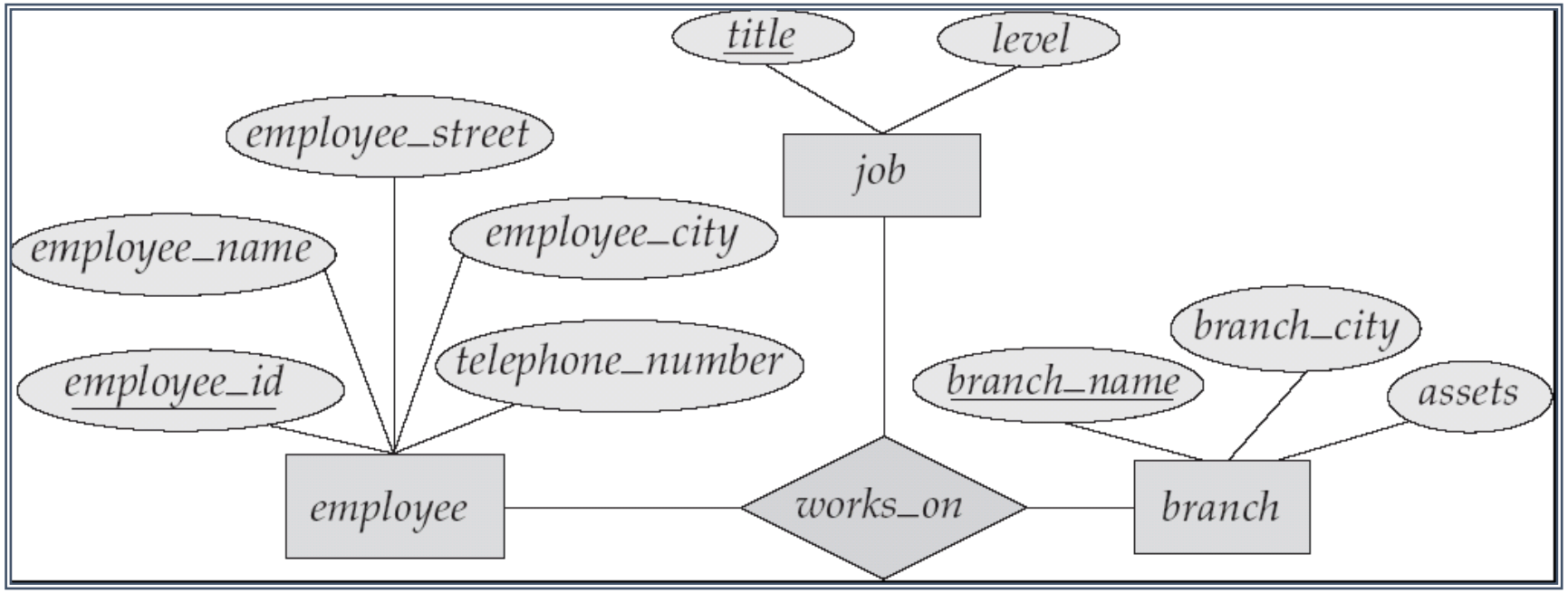


Alternative Notation for Cardinality Limits

- Cardinality limits and participation constraints can also be expressed in an E-R diagram in the form of cardinality limits in the form of l..h, where l is the minimum cardinality and h is the maximum cardinality
- An edge between an entity set and a binary relationship set can have an associated label l..h that indicates the minimum and maximum number of times an entity participates in a relationship set



E-R Diagram with a Ternary Relationship



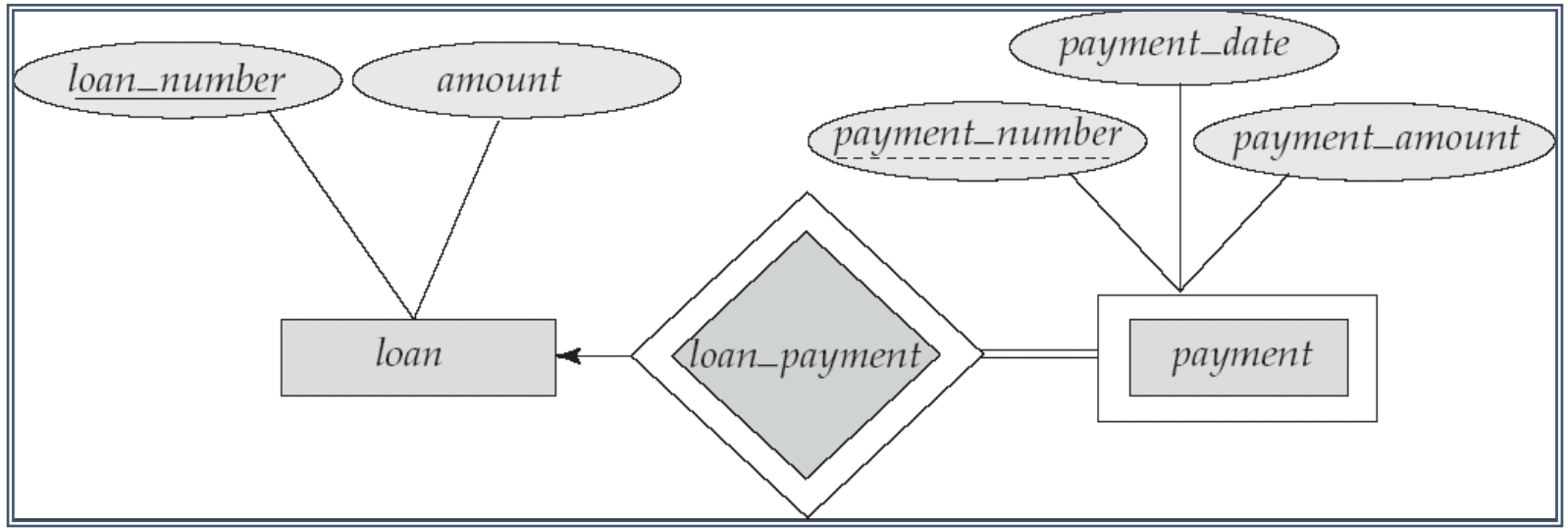
A ternary relationship is an association among three entities. The ternary relationship associates a single relationship set connected to three entity sets as shown in the above figure.

Weak Entity Sets

- An entity set that does not have a primary key is referred to as a weak entity set.
- The existence of a weak entity set depends on the existence of another identifying/owner entity set
- Identifying relationship is depicted using a double diamond
- The discriminator (*or partial key*) of a weak entity set is the set of attributes that distinguishes among all the entities of a weak entity set.
- The primary key of a weak entity set is formed by the *primary key of the strong entity set on which the weak entity set is dependent, plus the weak entity set's discriminator*.

Weak Entity Sets (Cont.)

- We depict a weak entity set by a double rectangle.
- We underline the discriminator of a weak entity set with a dashed line.
- payment_number – discriminator of the *payment* entity set
- Primary key for the weak entity set *payment* – (*loan_number, payment_number*)
- The weak entity set is related to the identifying entity set via a total, one-to-many relationship set from the identifying entity set to the weak entity set



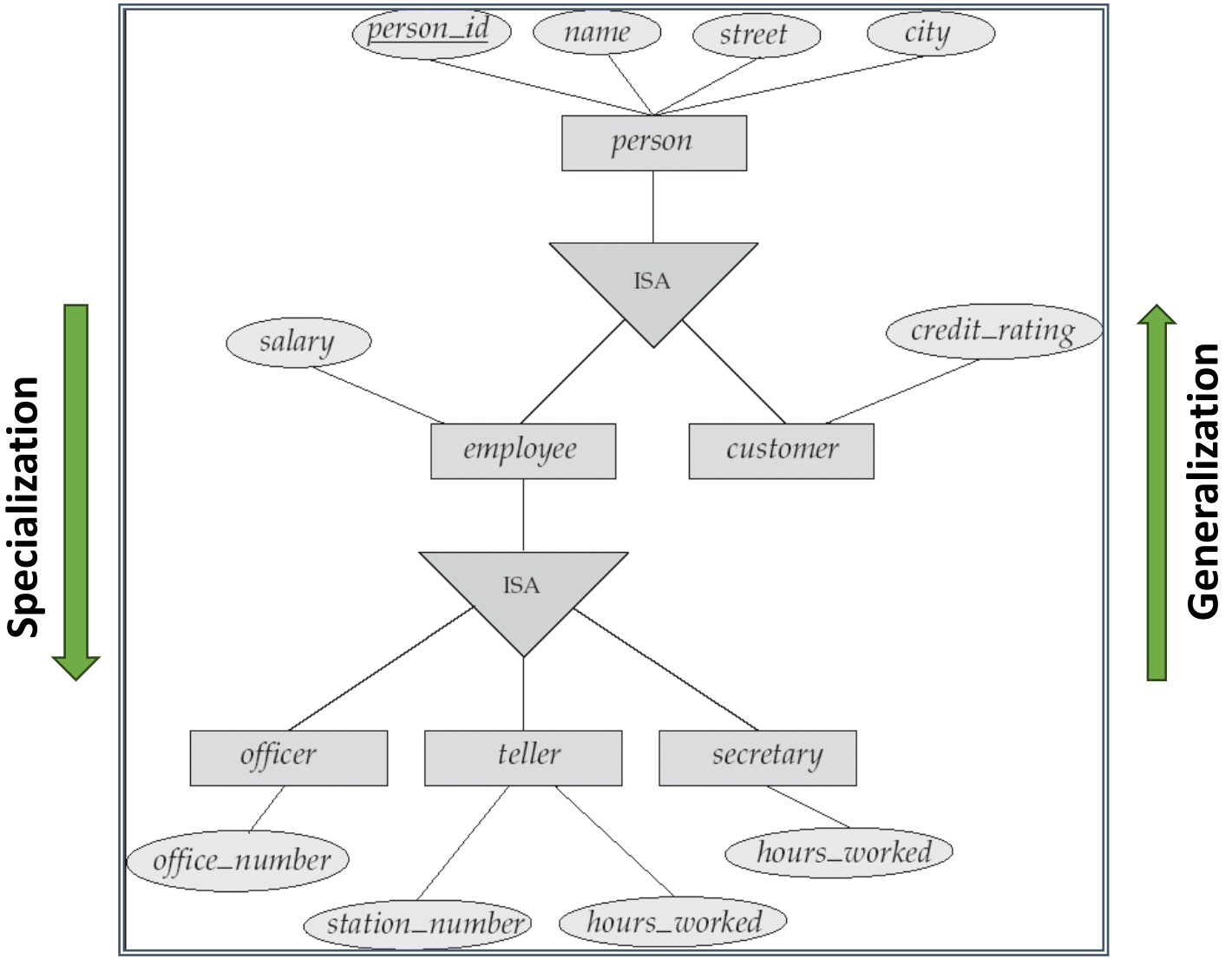
Extended E-R Features: Specialization

- Specialization is a top-down design process where the higher-level entity is specialized into two or more lower-level entities.
- In specialization, we designate sub-groupings (sub-entities) within an entity set that are distinctive from other entities in the set based on their characteristics.
- The process of designating sub-groupings within an entity set is called specialization.
- These subgroupings become lower-level entity sets that have new attributes that do not apply to the higher-level entity set.
- Specialization is depicted by a *triangle* component labeled ISA (E.g. *customer* "is a" *person*).

Extended ER Features: Generalization

- Generalization is a bottom-up design approach in which two or more entities can be generalized to a higher-level entity if they have some attributes in common. For Example, a STUDENT and FACULTY can be generalized to a higher-level entity called PERSON
- It is the process of extracting common properties from a set of entities and creating a generalized entity from it.
- It combines a number of lower-level entity sets that share the same features into a higher-level entity set. This is known as generalization.
- Specialization and generalization are simply inverse of each other.
- They are represented in an E-R diagram in the same way.

Specialization and Generalization Example

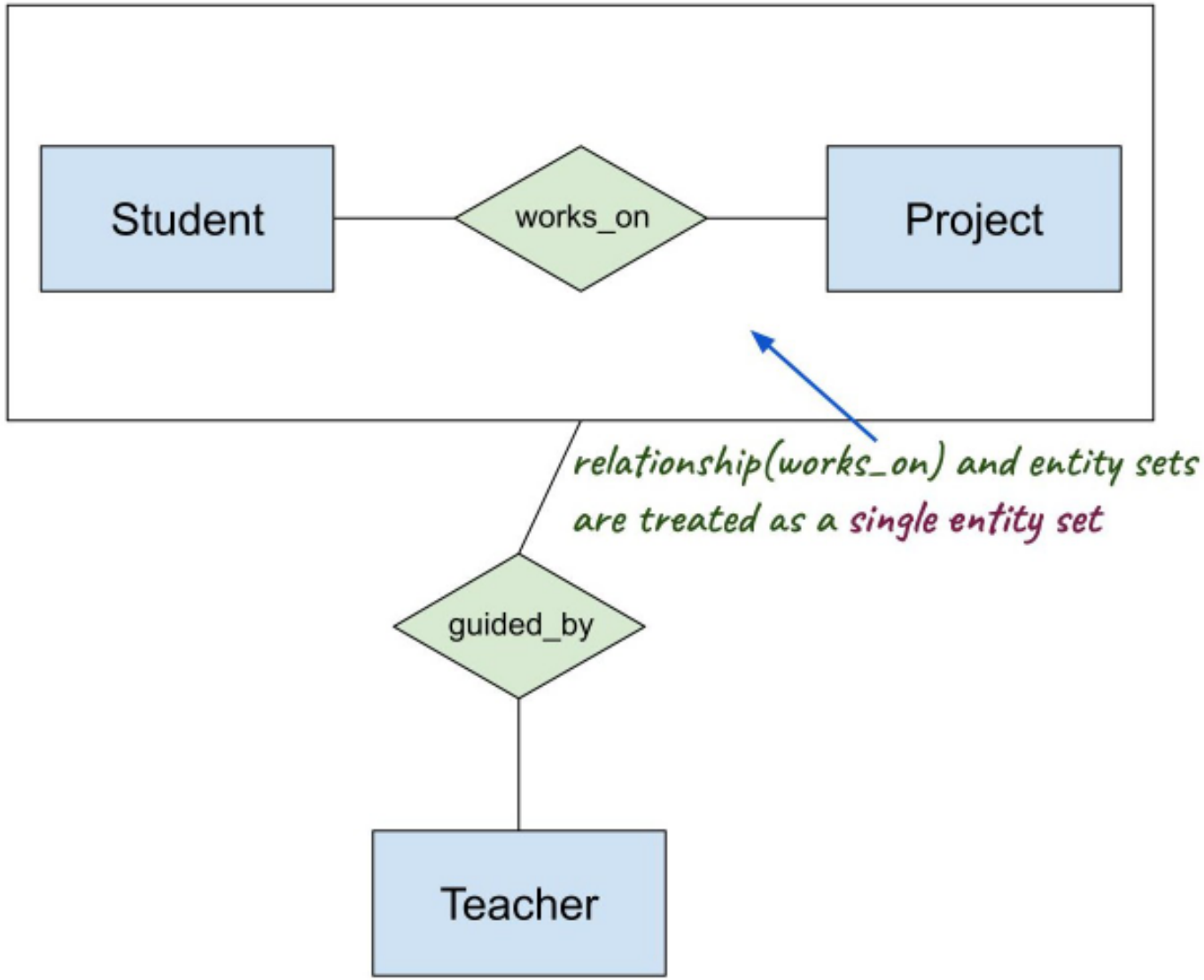


- The ISA relationship also referred to as superclass-subclass relationship
- Attribute inheritance – a lower-level entity set inherits all the attributes and relationship participation of the higher-level entity set to which it is linked.

Aggregation

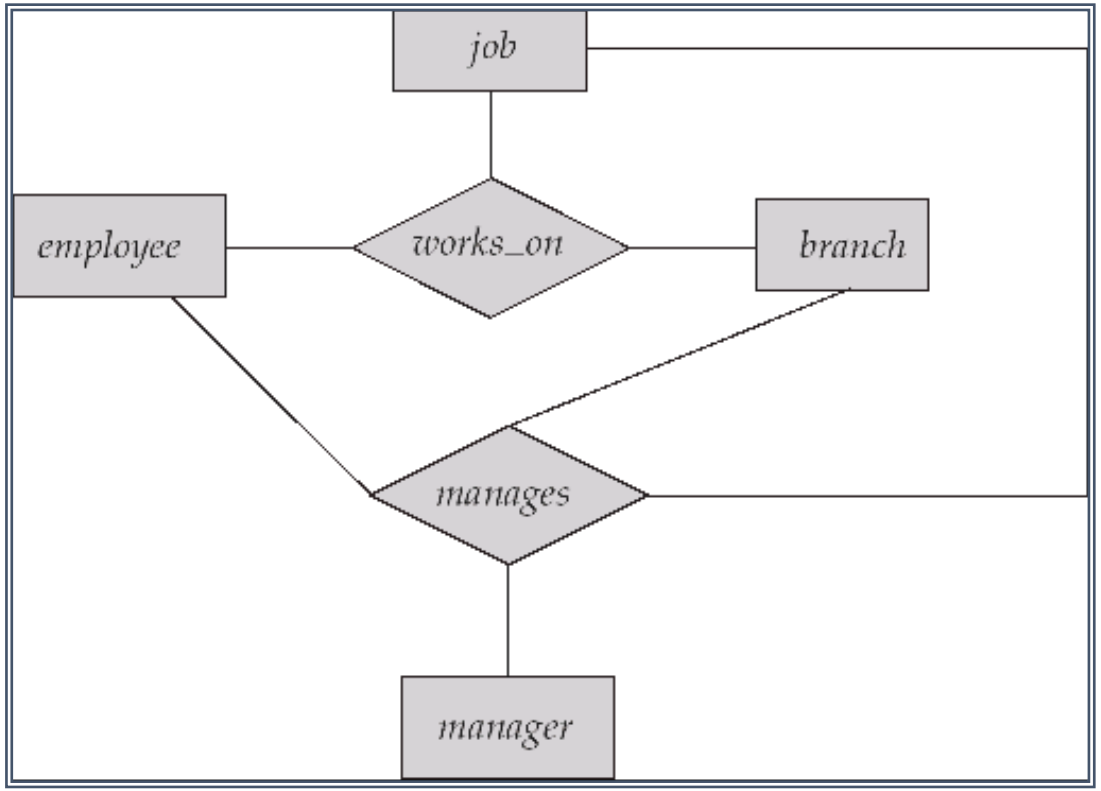
- The basic ER model lacks the feature to establish a relationship between two (or more) 'relationship sets' which may be required in some scenarios.
- Thus, aggregation was introduced in Extended E-R Model to support such situations.
- In those cases, a relationship with its corresponding entities is aggregated into a higher-level entity.
- Aggregation is an abstraction through which we can represent relationship between entity sets as higher-level entity sets.

Aggregation Example1

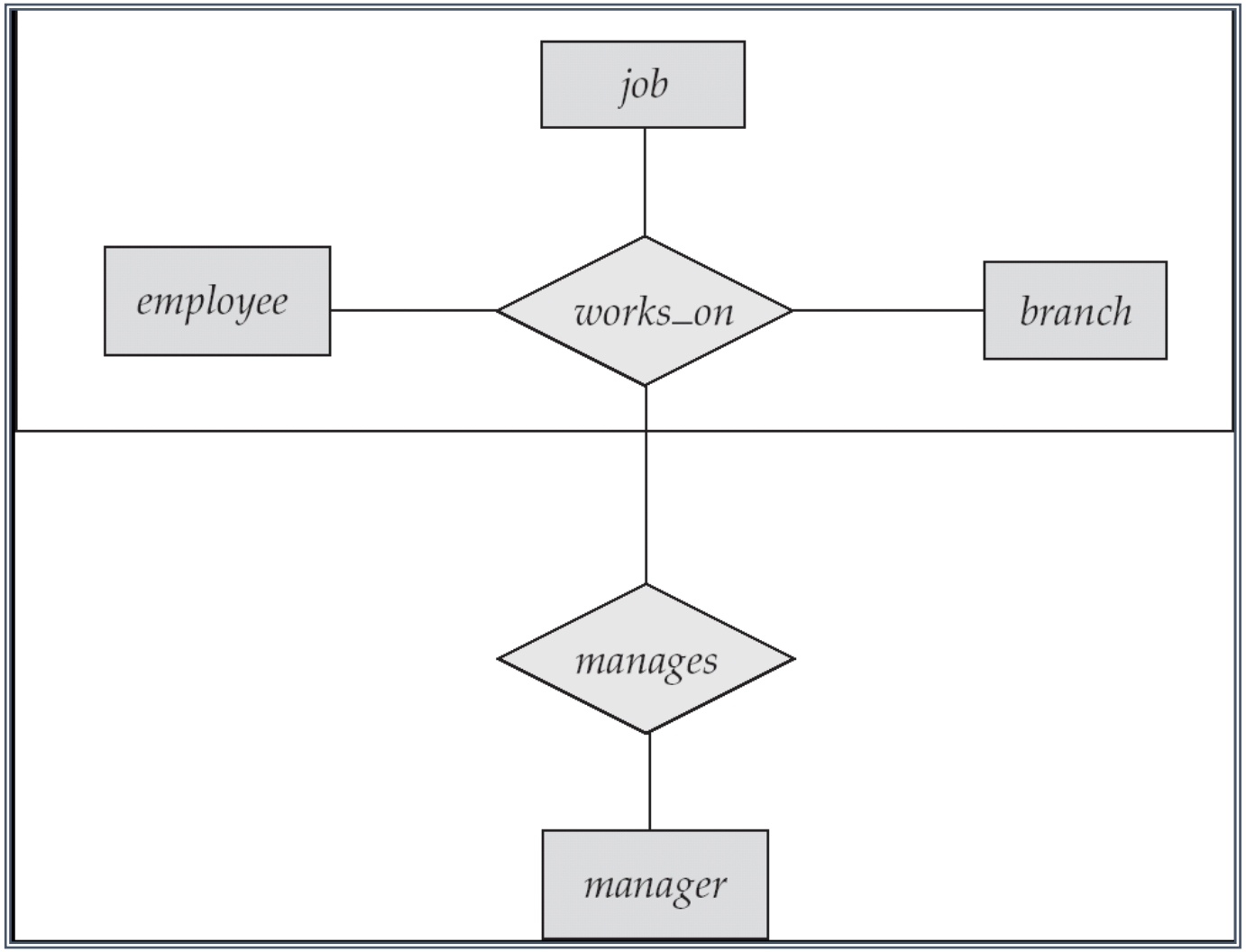


Aggregation Example2

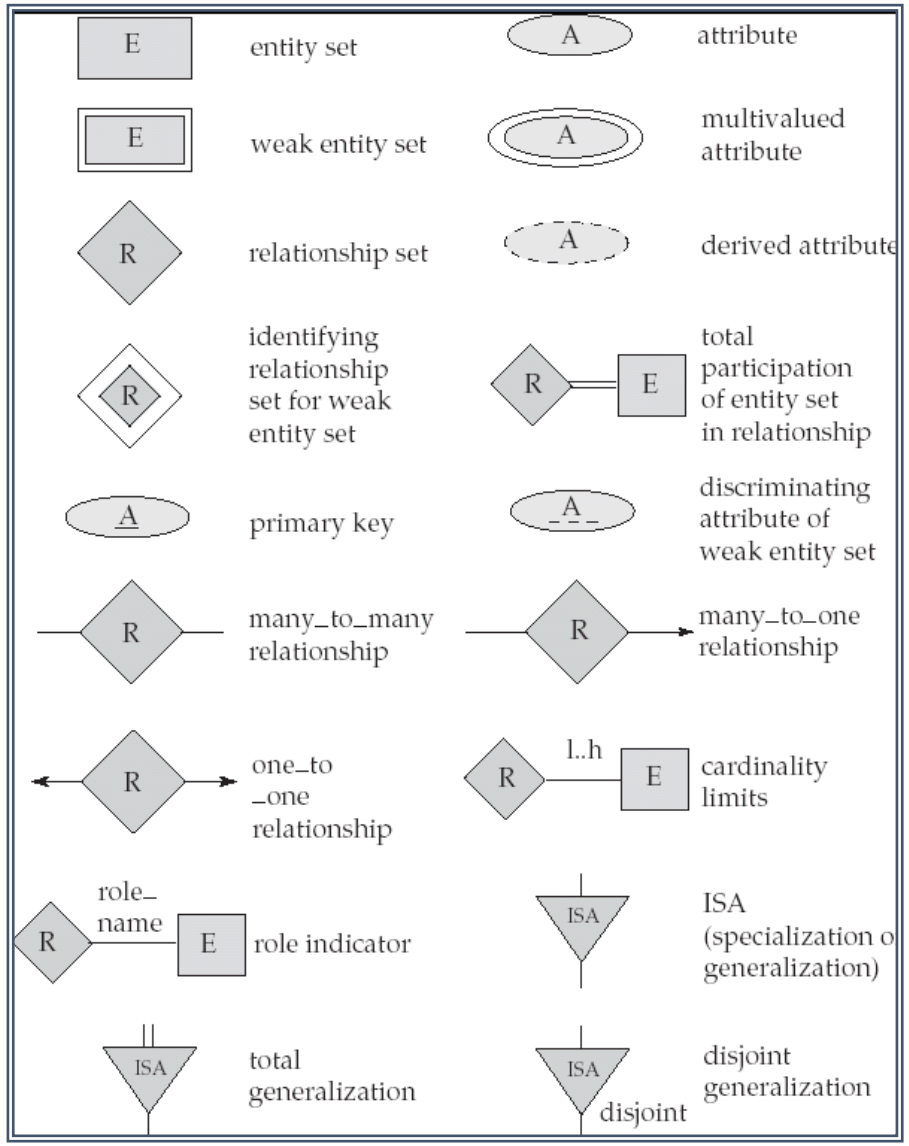
- Consider the ternary relationship *works_on*, which we saw earlier
- Suppose we want to employ managers for recording tasks performed by an employee at a branch



Aggregation Example2



Summary of Symbols Used in E-R Diagrams



ANY DOUBTS?

THANK YOU!