# Entity-Relationship Model

# Entity Sets

- 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
- An entity has a set of properties, values of some properties uniquely identify an entity.
  - 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

# Key Attributes of an entity type

- A **key attribute** is a minimal set of attributes of an entity set, which uniquely identifies an entity in an entity set.

- An entity type usually has one or more attributes whose values are distinct for each individual entity in the entity set. Such an attribute is called a **key attribute**

- A key may be a single attribute or may be more than one attribute.

**Example:**

* For the student entity the RegNo can be the key attribute.
* For a person entity the SSN can be the key attribute.

- Some times key may be formed by the combination of several attributes – a composite attribute. (ie., the combination of the attribute values must be distinct for each individual entity.

**Example:** The registration no. for a vehicle with two two simple attributes, i.e., state number. and registration number

# Value Sets (Domains) of Attributes

## Value Sets (Domains) of Attributes:

- A set of values that may be assigned to the attributes of each individual entity, in an entity set is called the value set or domain.

## Example:

* For employee entity, if age limit is 20 to 58 then the value set (domain) of attribute age consists of integer from 20 to 58.
  Age: Domain [ 20 – 58]

* The value set for name attribute is a set of alphabets and some special characters.
  Name: Domain [a – z], [A – Z], blank space, dot

# 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*

# Attributes

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

  Example:

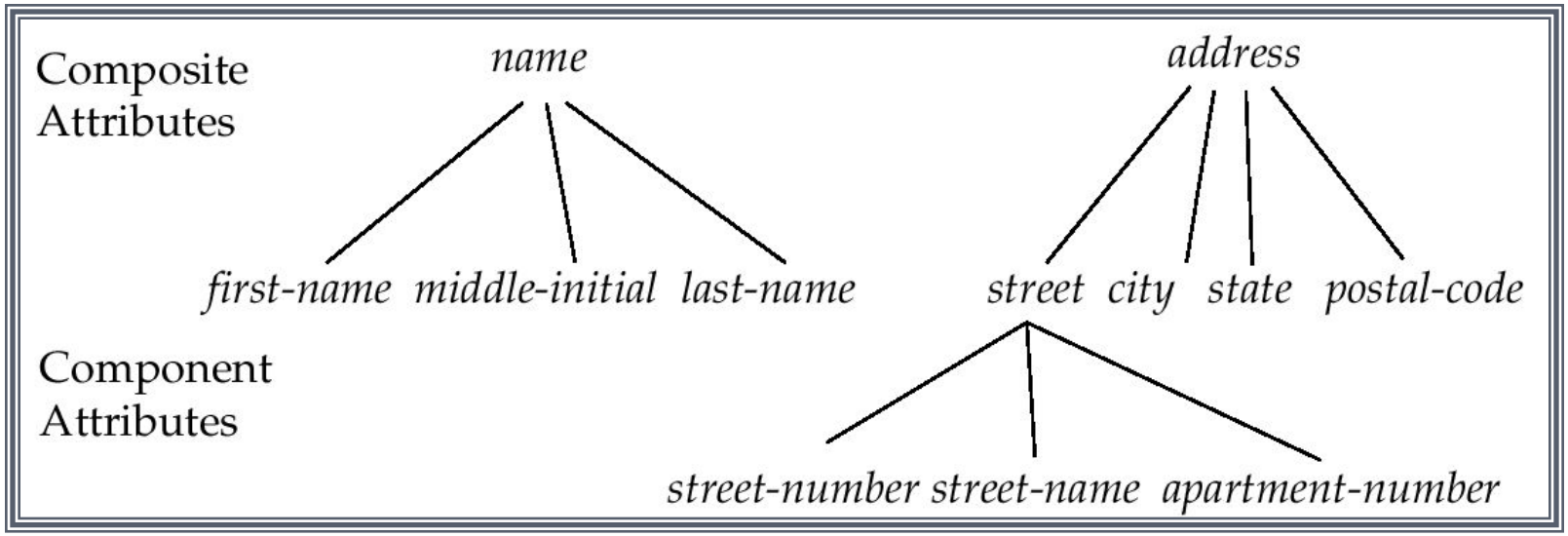
  *customer = (customer-id, customer-name, customer-street, customer-city)*
  *loan = (loan-number, amount)*

- *Domain* – the set of permitted values for each attribute
- Attribute types:
  - *Simple* and *composite* attributes.
  - *Single-valued* and *multi-valued* attributes
  - *Derived* attributes

- Simple attributes: Attributes which are not divided into sub parts.it is called atomic attributes
- Eg: age

- Composite attributes: Attributes which can be divided into subparts, helps us to group together related attributes.
- Eg: Name- First name, Middile initial, Last name

# Composite Attributes




Name, Address are the composite attributes.
First name, middle initial, last name,street,city,state,postal code are component attributes.
Street no, street name, appartment number are component attributes.

- Single valued attribute: Attributes having a single value for a particular entity.
- Eg: loan number attribute for a specific loan entity refers to only one loan number.
- Eg: Date of birth
- Multivalued attribute: Attributes having a set of values for a specific entity.
- Eg: an emp entity set with the attribute phno
- entity student can have multiple values for the hobby attribute- reading, listening music etc
- Derived attributes: Values for this type of attributes can be derived from the value of another other related attributes or entities.

- e.g. entity set customer has an attribute age, which indicates the customer's age. If the customer entity set also has an attribute date-of-birth. Age is calculated from date-of-birth and the current date. Thus age is a derived attribute.
- In this case dob may be refered to as base attribute or stored attribute.
- Strored attribute: the value of derived attribute is not stored but is computed when required.
- Null attribute: Attribute can be null.A null value is used when an entity does not have a value for an attribute.
- Null also indicates attribute value is unknown.

# Relationship Sets

- A relationship is an association among several entities

  Example: Book is published by a particular publisher.

  Relationship set is the set of all relationship of the same type.

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

- $E_1, E_2, \ldots, E_n$ are *entity sets then relationship set R is defined as*

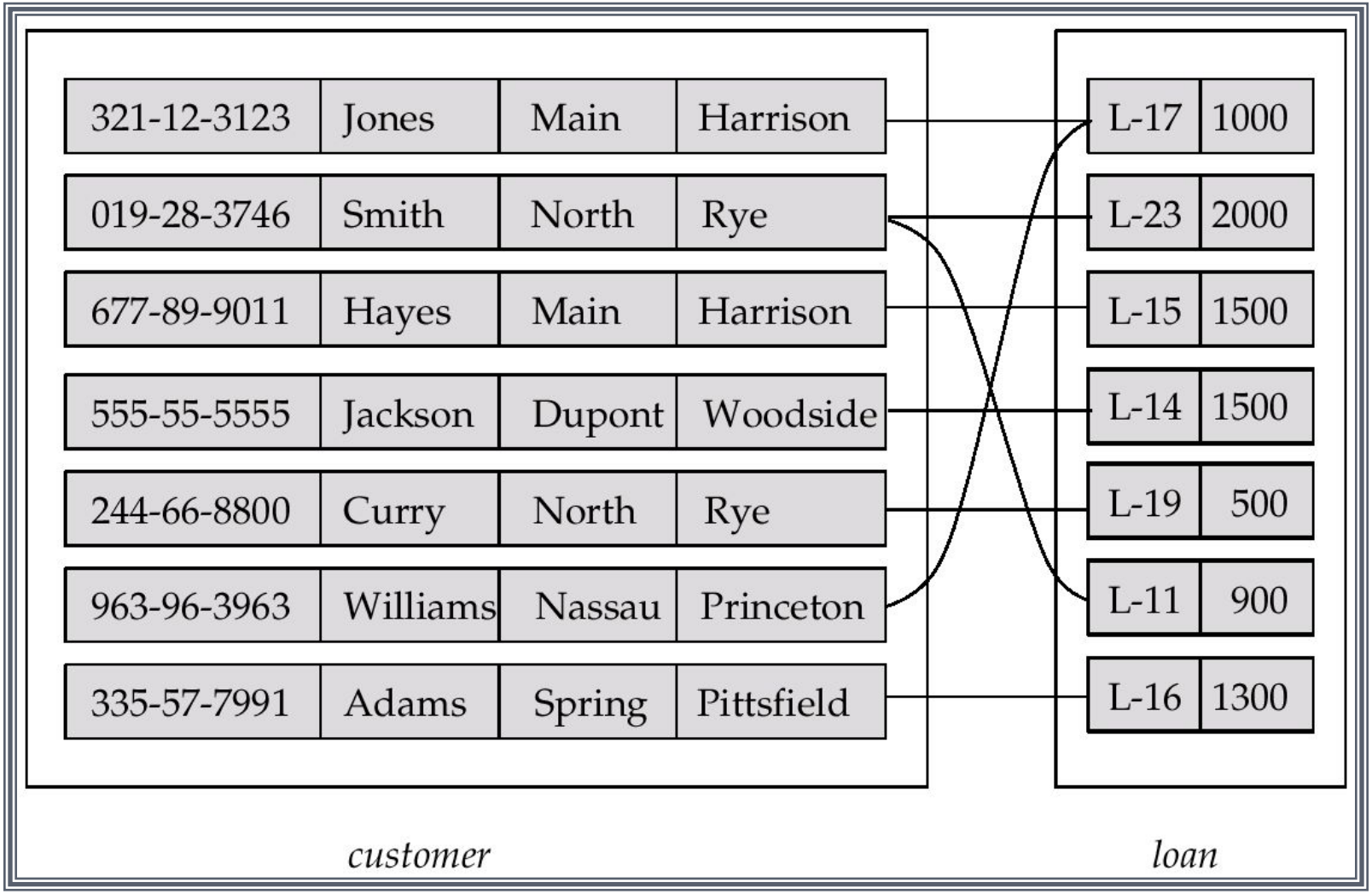
$$\{(e_1, e_2, \ldots e_n) \mid e_1 \in E_1, e_2 \in E_2, \ldots, e_n \in E_n\}$$

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

  - Example:

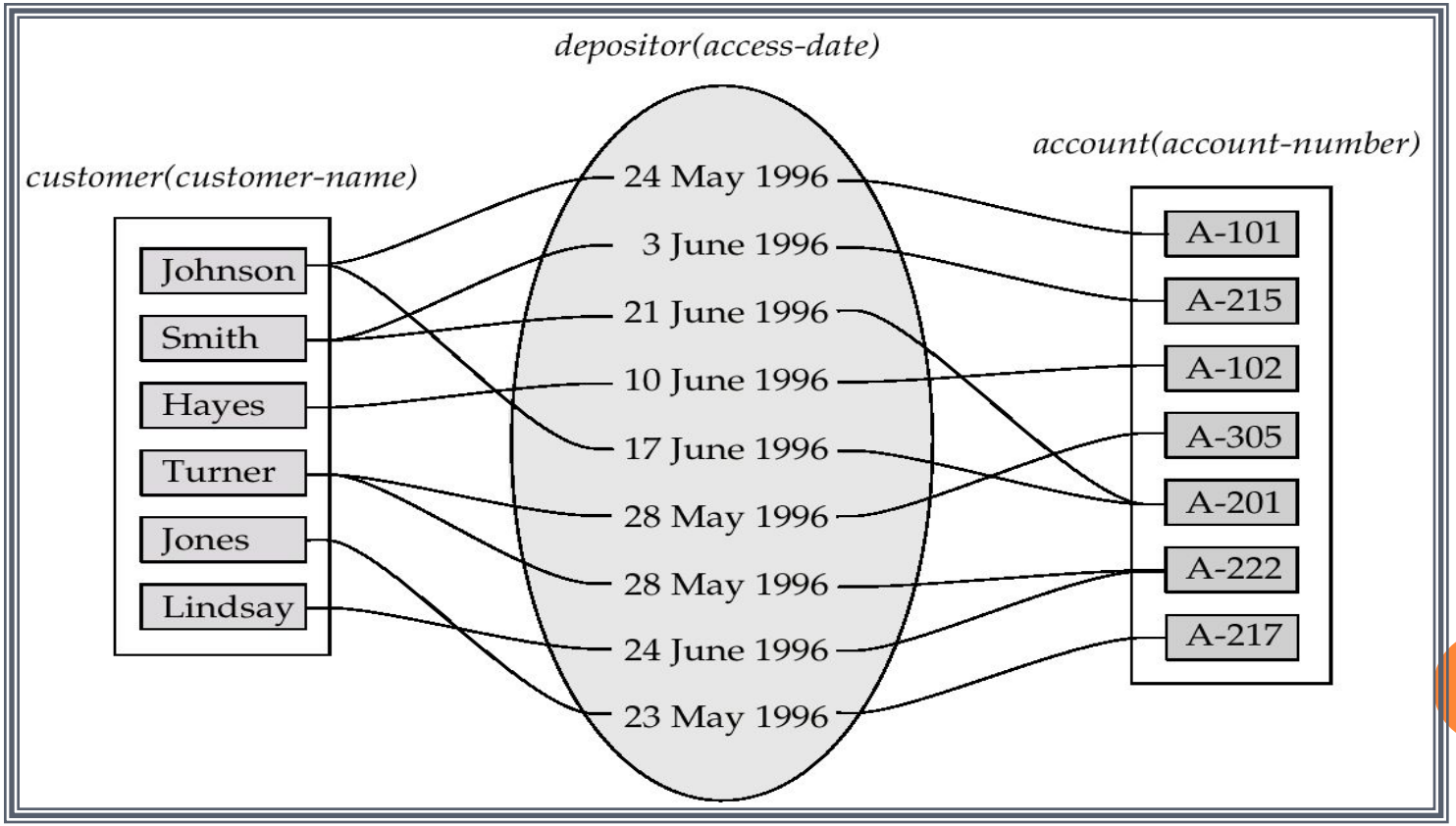
    (Hayes, A-102) $\in$ *depositor*

# Relationship Set *borrower*



# Relationship Sets (Cont.)

- A relationship may also have attribute called descriptive attribute.
- 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*

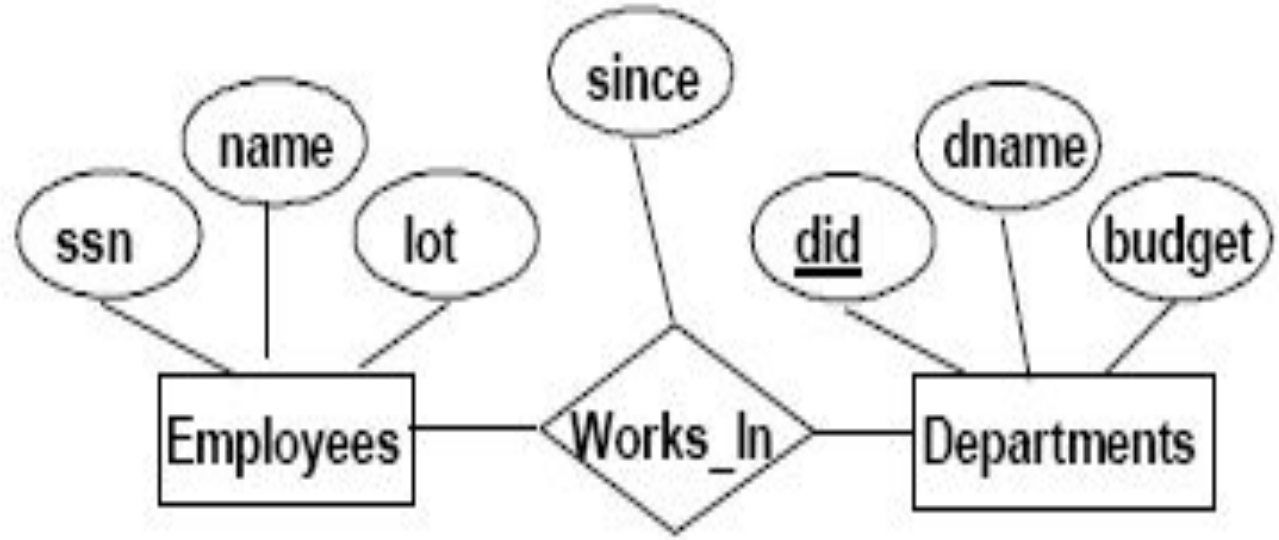


# Degree of a Relationship Set

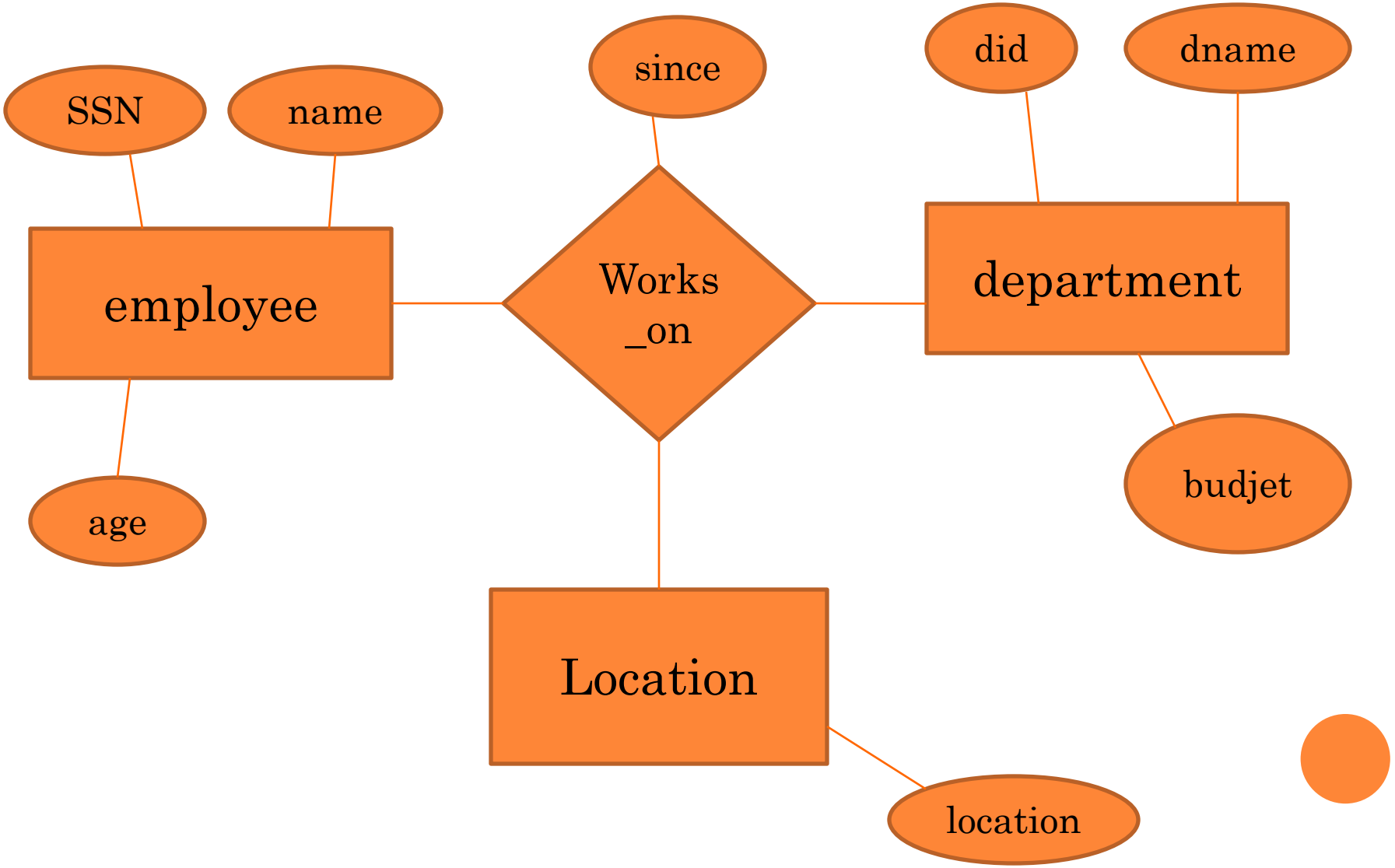
- Refers to number of entity sets that participate in a relationship set.
- Relationship sets that involve two entity sets are *binary* (or degree two). Generally, most relationship sets in a database system are binary.
- Relationship sets may involve more than two entity sets.
  - E.g. Suppose employees 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.
- Relationship types of degree 3 are called **ternary** and of degree n are called **n-ary**

# E.G. BINARY RELATIONSHIP



# E.G. TERNARY RELATIONSHIP



# ER Diagram

An entity-relationship diagram (ERD) is a data modeling technique that graphically illustrates an information system's entities and the relationships between those entities.

An ERD is a conceptual and representational model of data used to represent the entity framework infrastructure.

The elements of an ERD are:

Entities

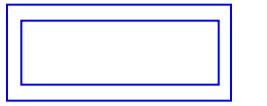
Relationships

Attributes

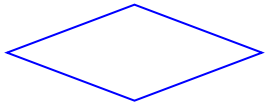
- **Rectangles** represent entity sets. entity written in upper case, where as the attribute name is written in lower case.
- **Diamonds** represent relationship sets.
- **Lines** link attributes to entity sets and entity sets to relationship sets.
  - **Underline** indicates primary key attributes
- **Ellipses** represent attributes
  - ✔ ***Double ellipses represent multivalued attributes.***
  - ✔ ***Dashed ellipses denote derived attributes.***

## ERD symbols

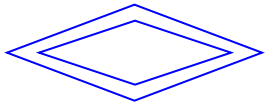
Entity



Weak entity



Relationship

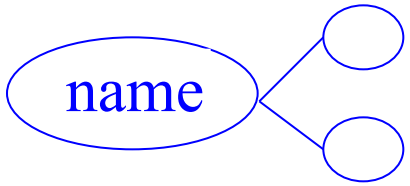


Identifying relationship


Attribute


Key attribute

## ERD symbols




Composite attribute


Derived attribute


Mutlti-valued attribute



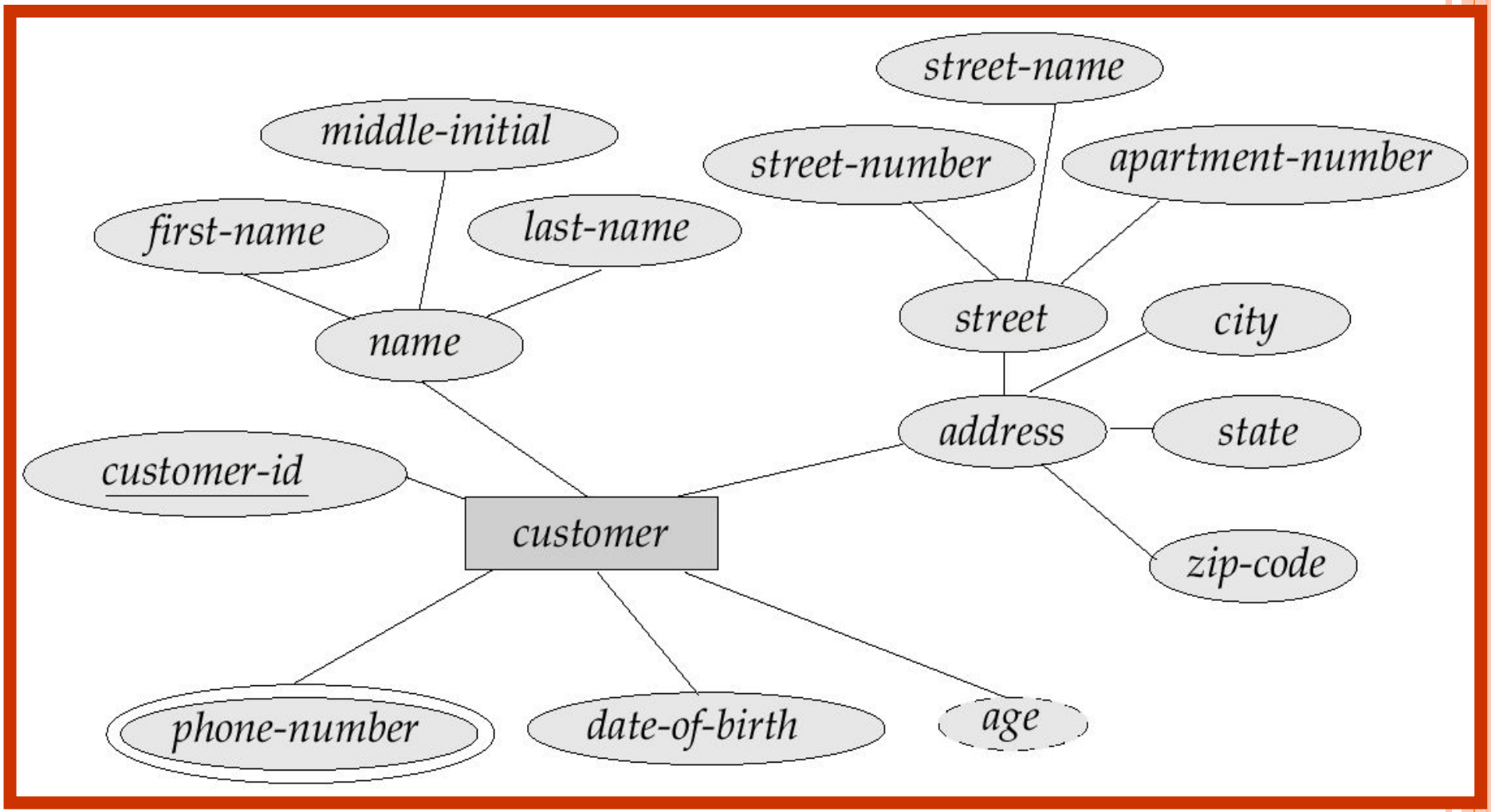
Partial participation

Total participation

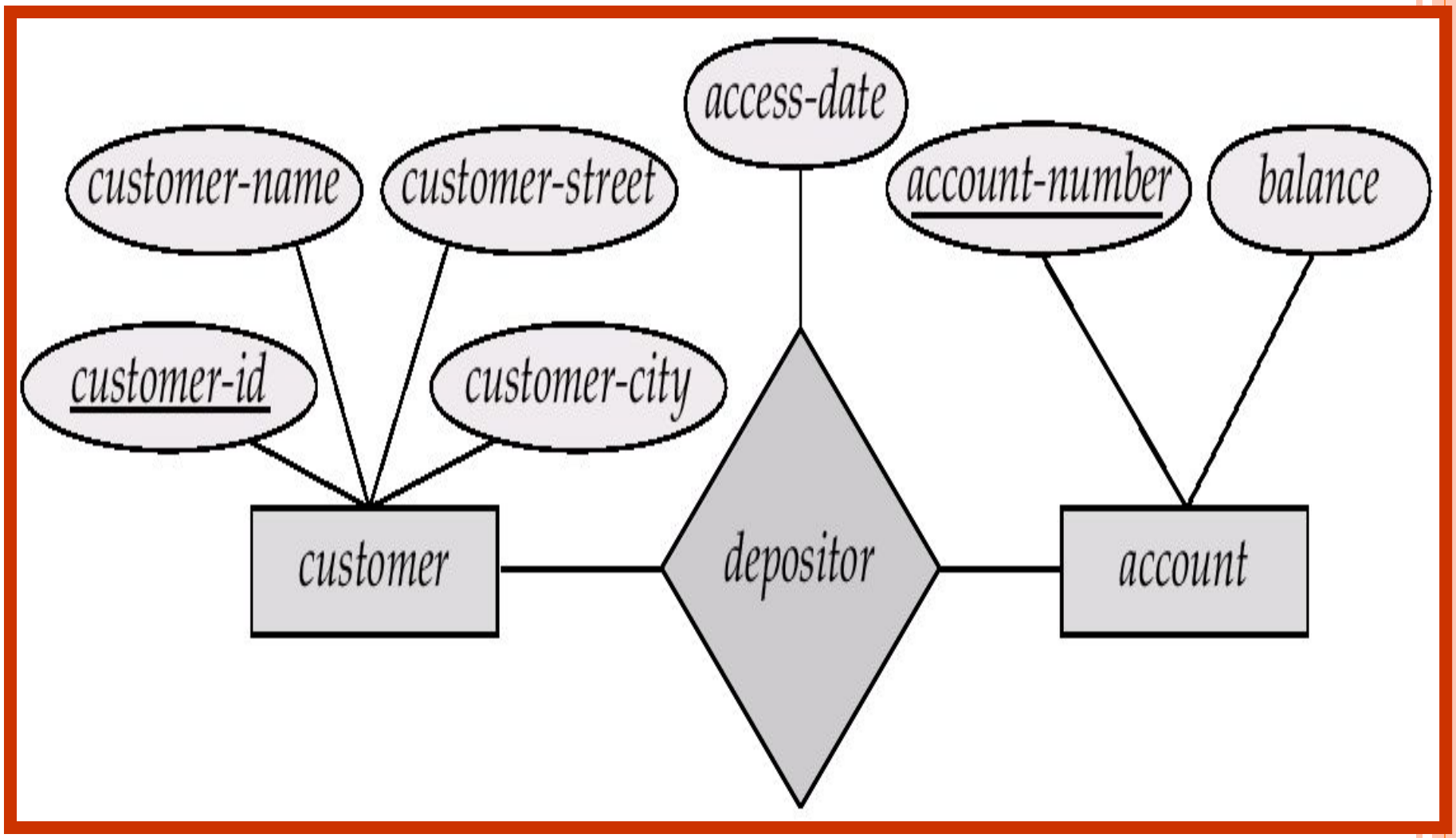
1 n N

Cardinality

# ER diagram with composite, Multivalued and Derived attributes



# Eg: Relationship sets with attributes



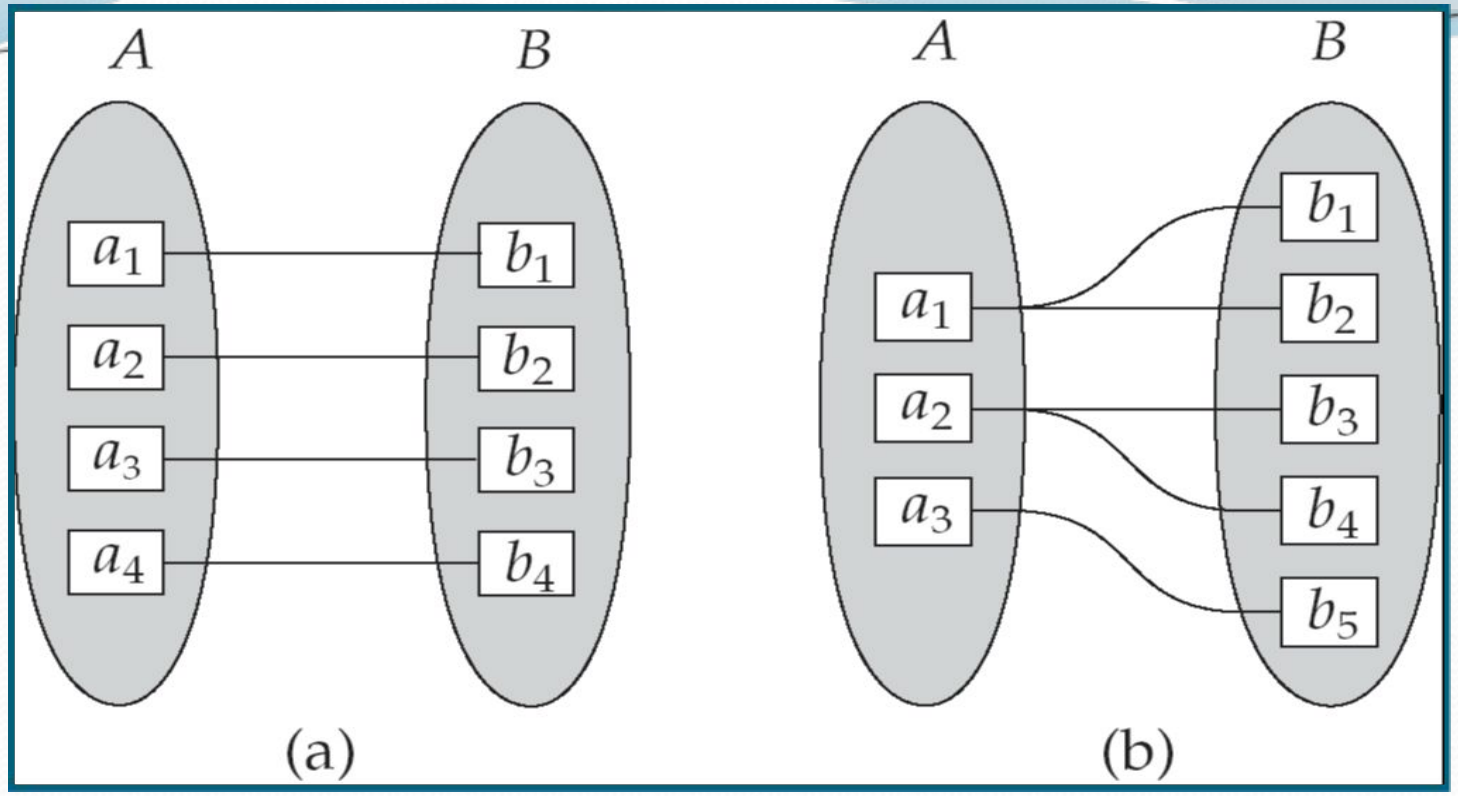
# Additional features of ER model

# 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.

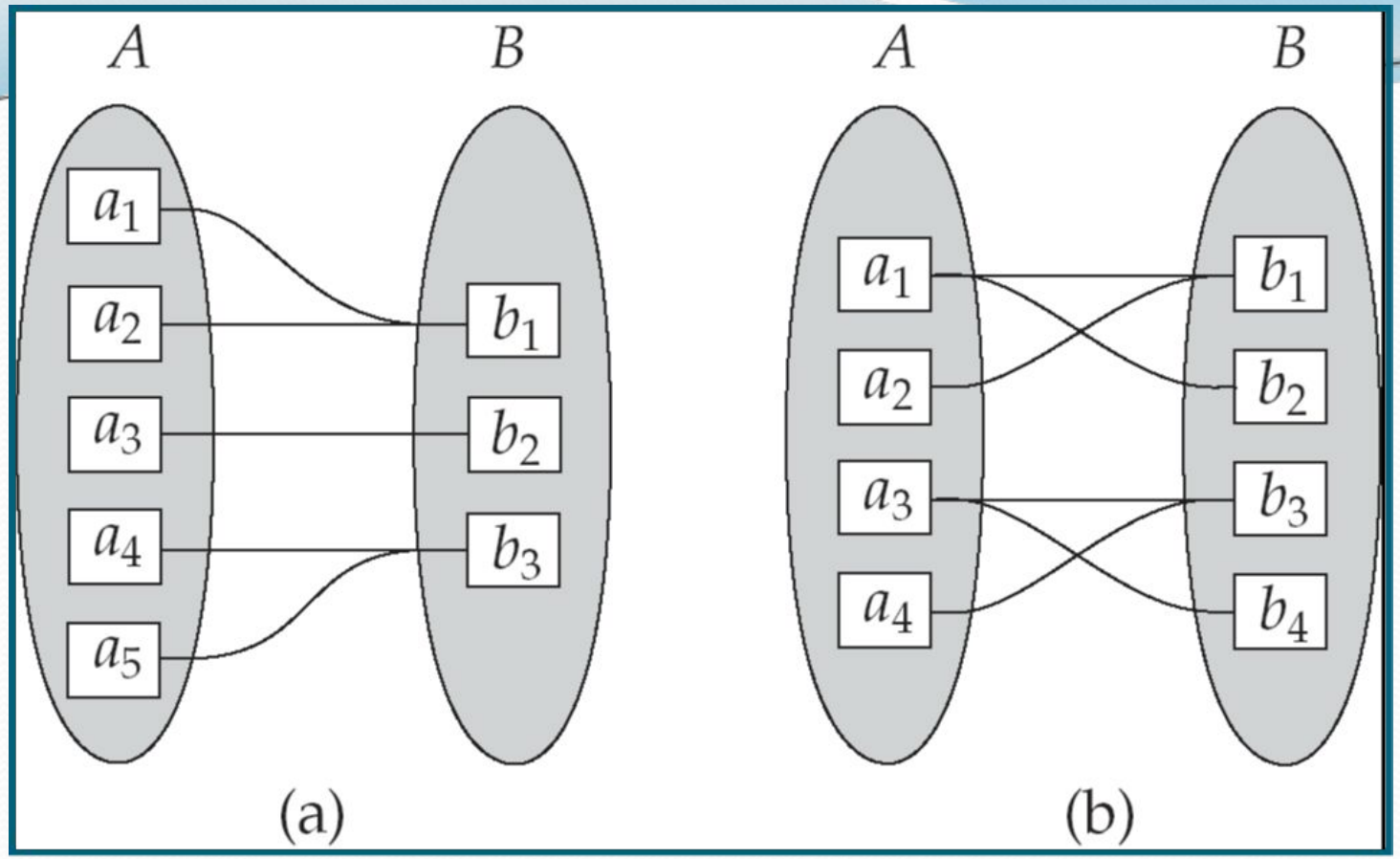
# Mapping Cardinalities

- Express the number of entities to which another entity can be associated via a relationship set.
- 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




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.

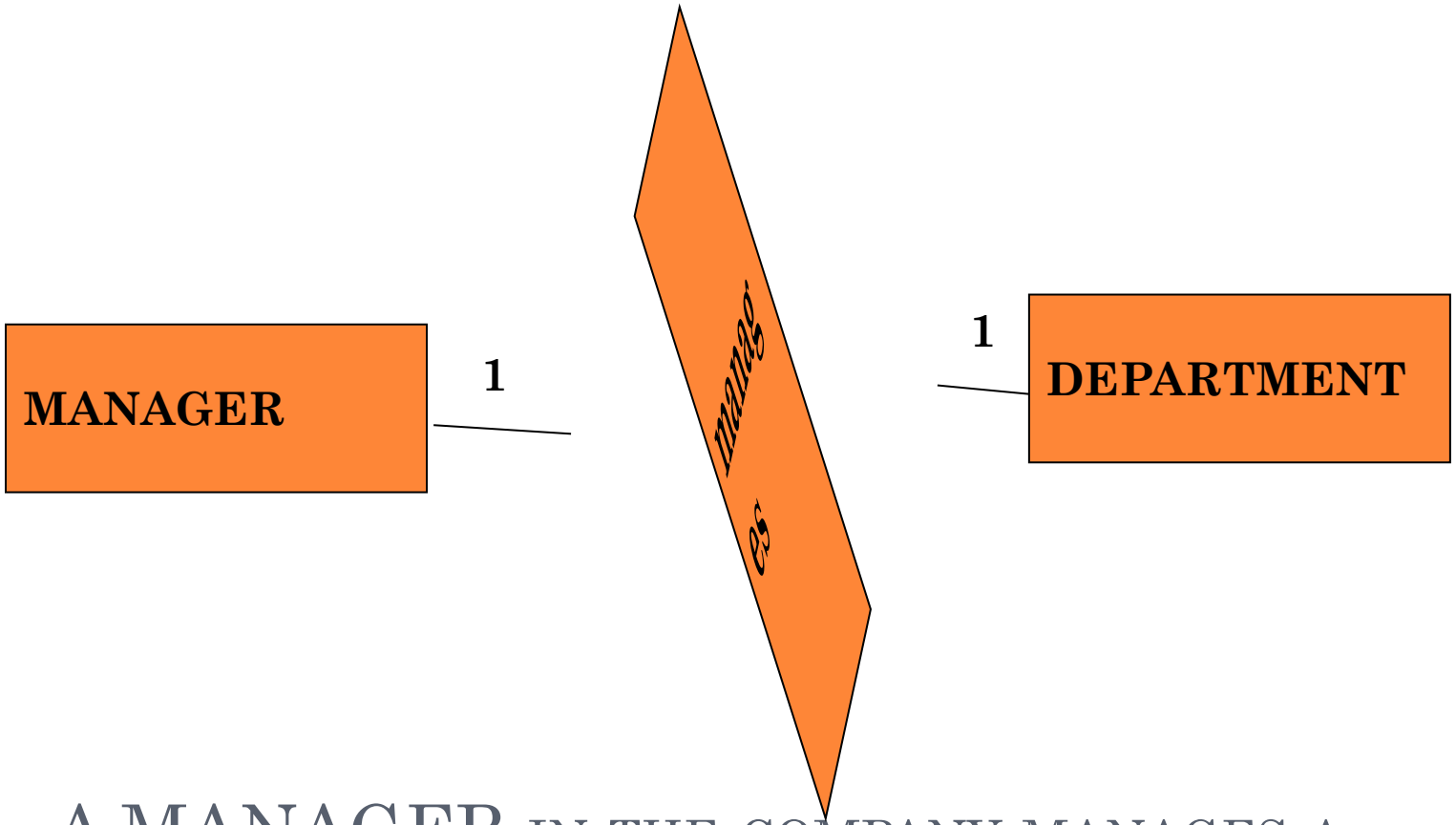
One to Many: An entity in A is associated with any number of entities in B. An entity in B, however can be associated with at most one entity in A.




Many to One: An entity in A is associated with at most one entity in B, and an entity in B is associated with any number of entities in A.

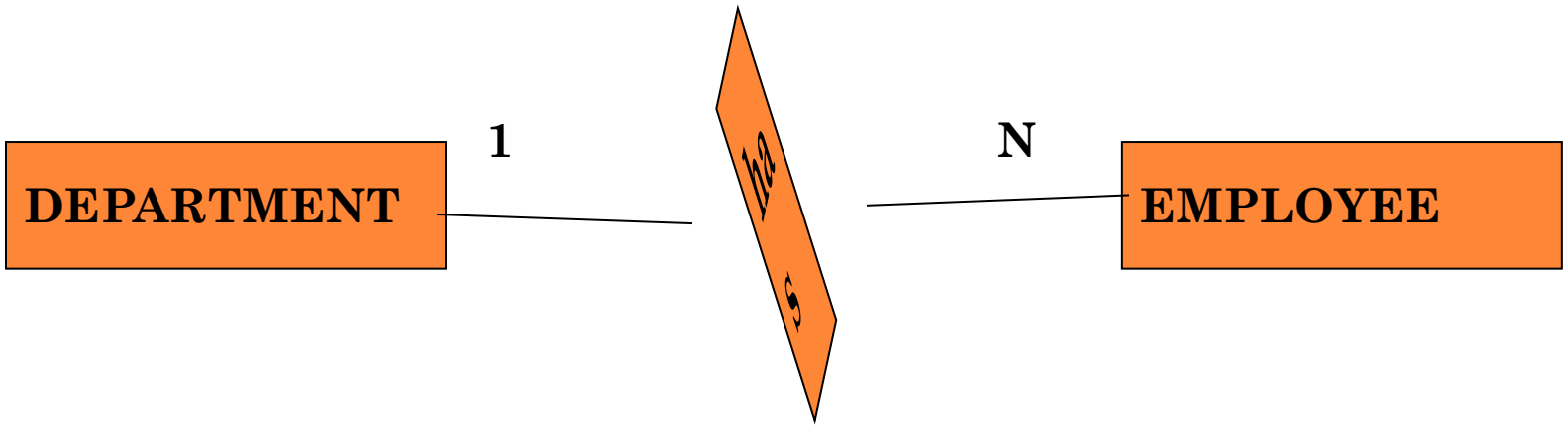
Many to Many: An entity in A is associated with any number of entities in B. and n entity in B, is associated with any number of entities in A.

# ONE TO ONE RELATIONSHIP



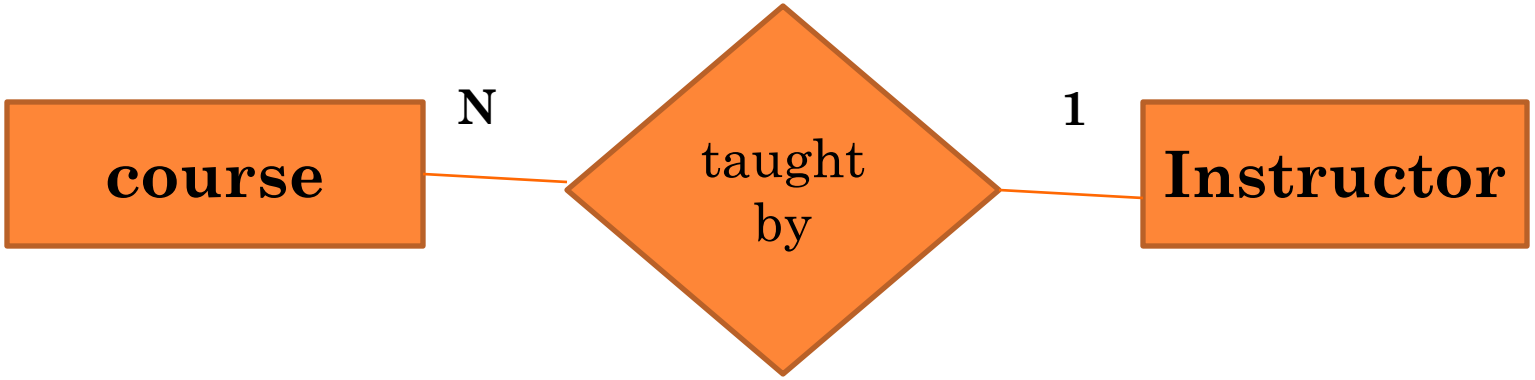

A MANAGER IN THE COMPANY MANAGES A SINGLE DEPARTMENT

# ONE TO MANY RELATIONSHIP



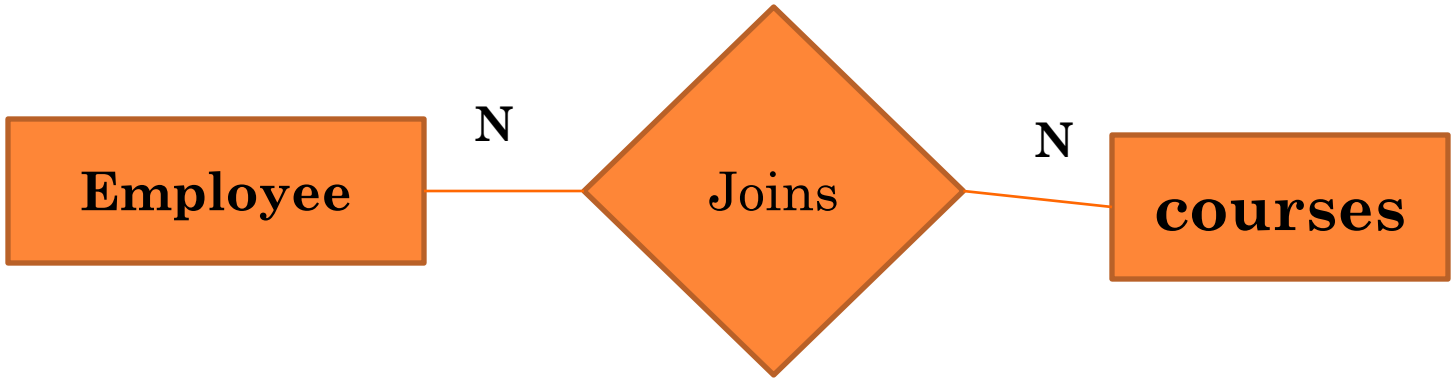

A DEPARTMENT CAN HAVE MORE THAN ONE EMPLOYEE

# Many to one relationship




N courses can be taught by one instructor

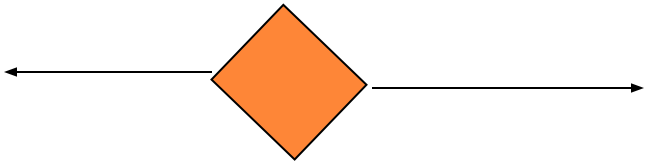
# Many to many relationship



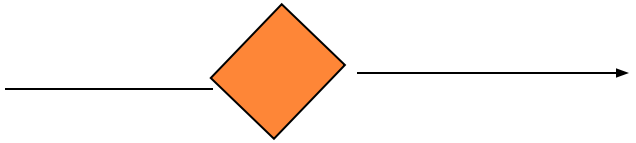

Each Employee can join for more than one course

# Mapping cardinality

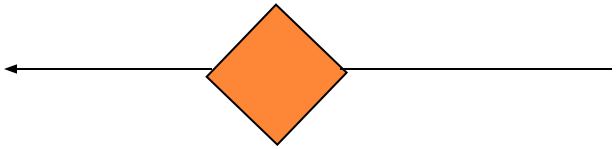
One to One



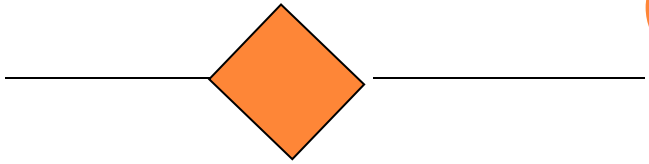
Many to one



One to many

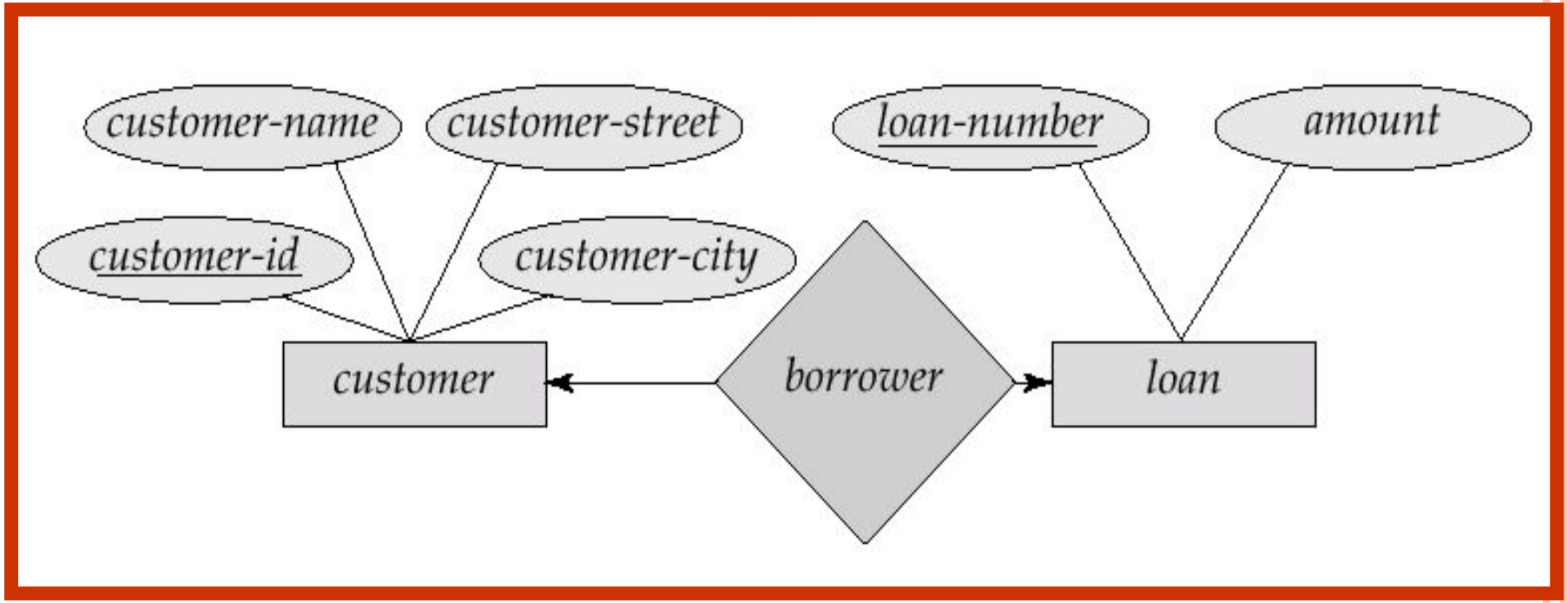


Many to many



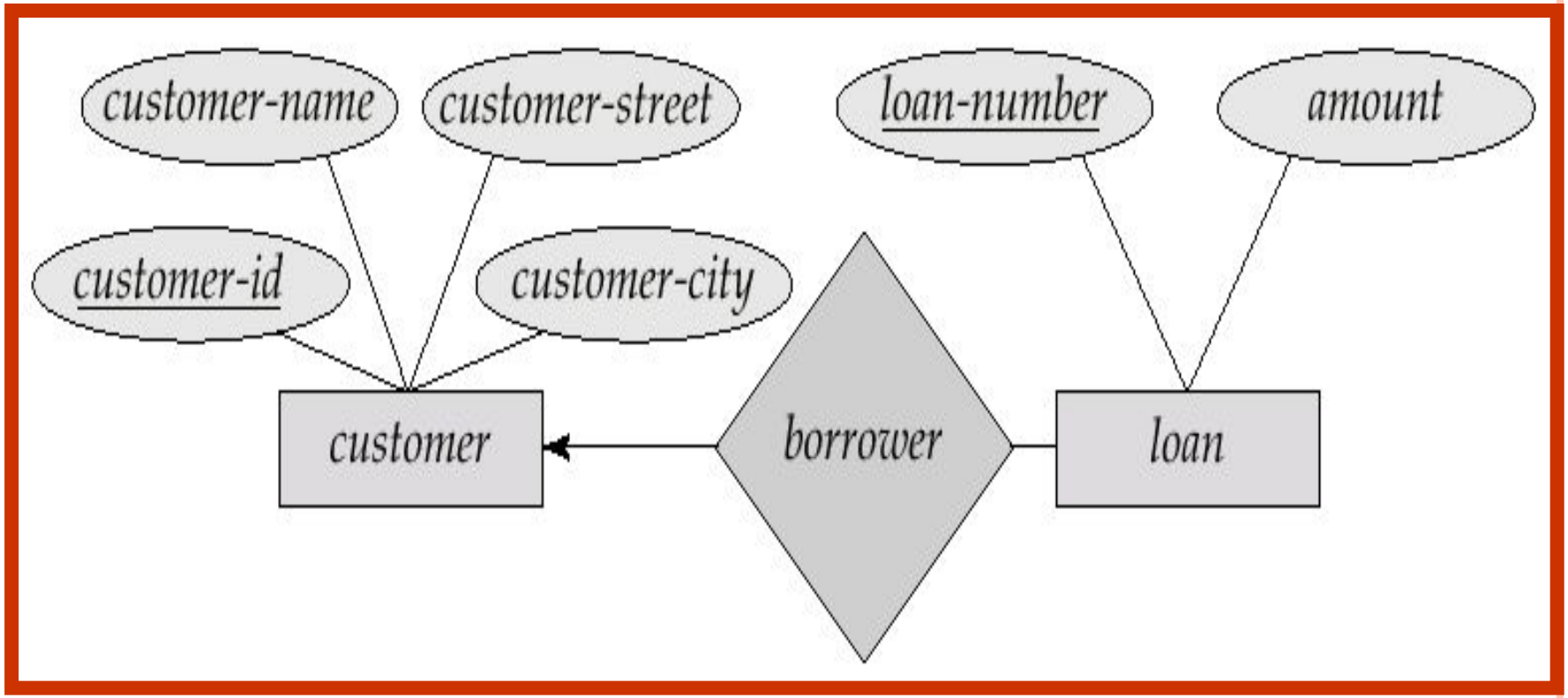
# One to one relationship

- A customer is associated with at most one loan via the relationship *borrower*
- *A loan is associated with at most one customer via borrower*



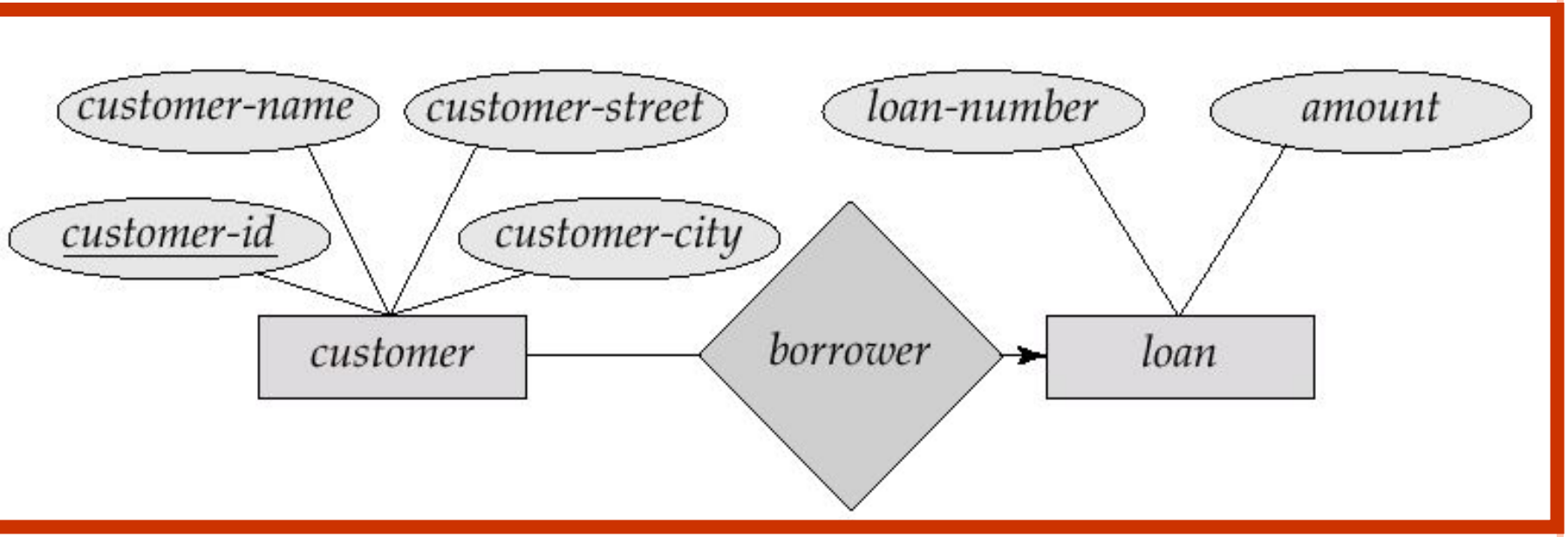
# One to many relationship

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



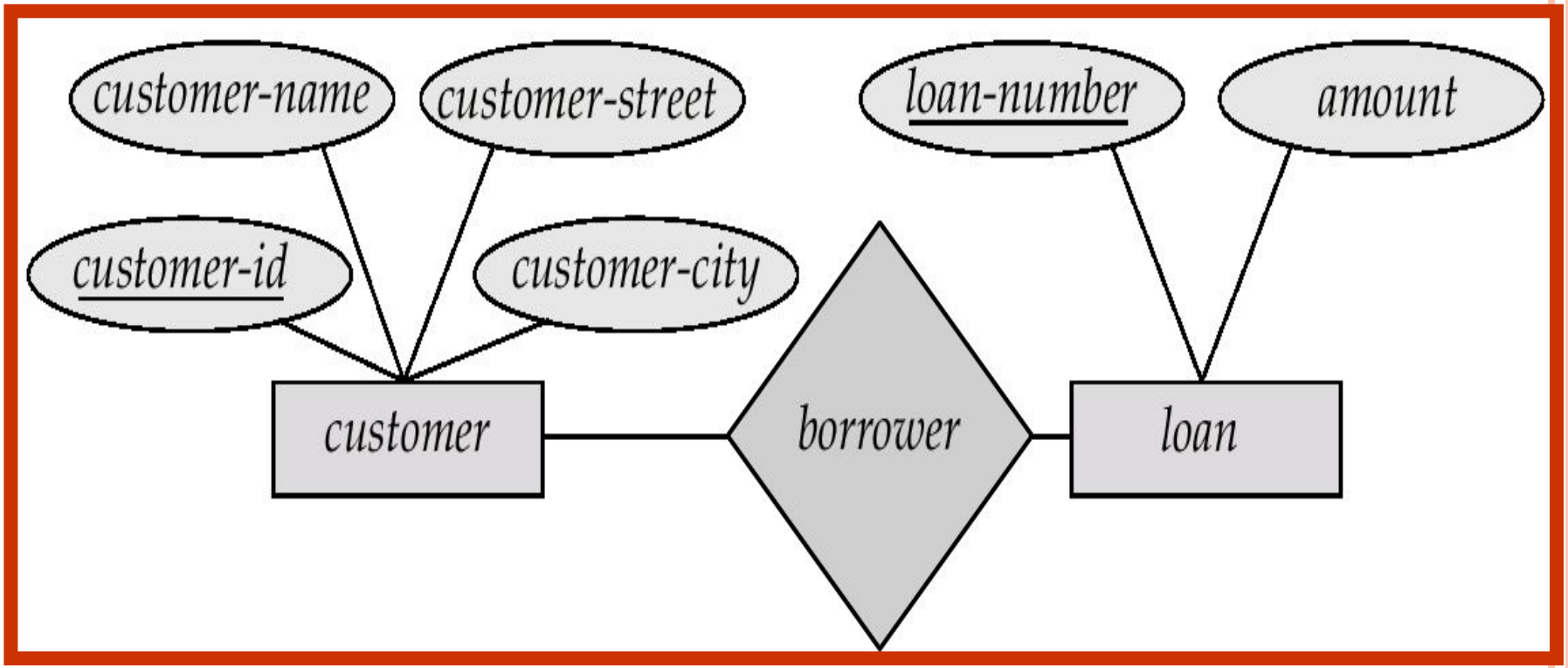
# Many to one relationship

- 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*



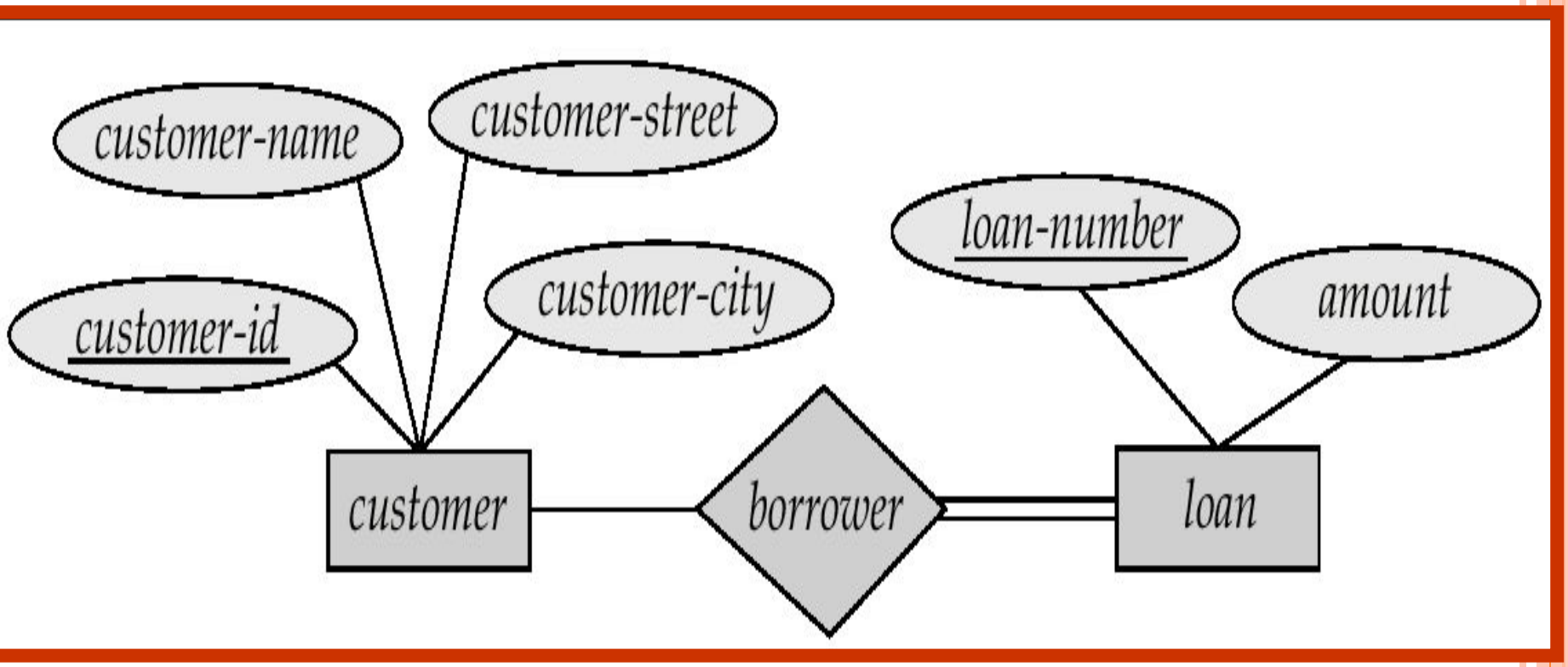
# Many to many relationship

- A customer is associated with several (possibly 0) loans via borrower and A loan is associated with several (possibly 0) customers via borrower

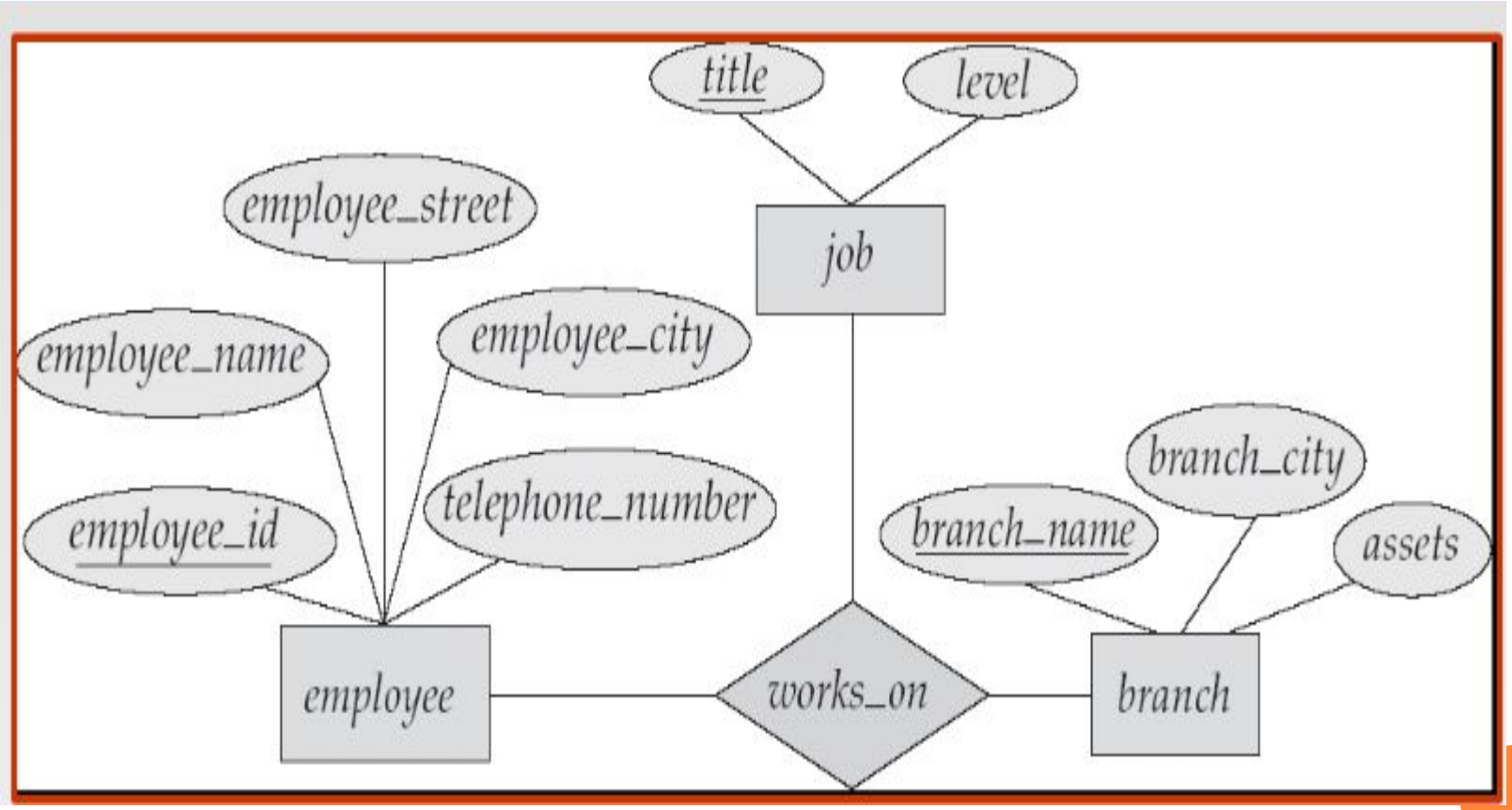


# Participation of entity set in a Relationship set

- **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** some entities may not participate in any relationship in the relationship set
  - ✔ ***E.g. participation of customer in borrower is partial***


customer-name
customer-street
customer-id
customer-city
loan-number
amount
customer
borrower
loan

# E-R diagram with a ternary relationship

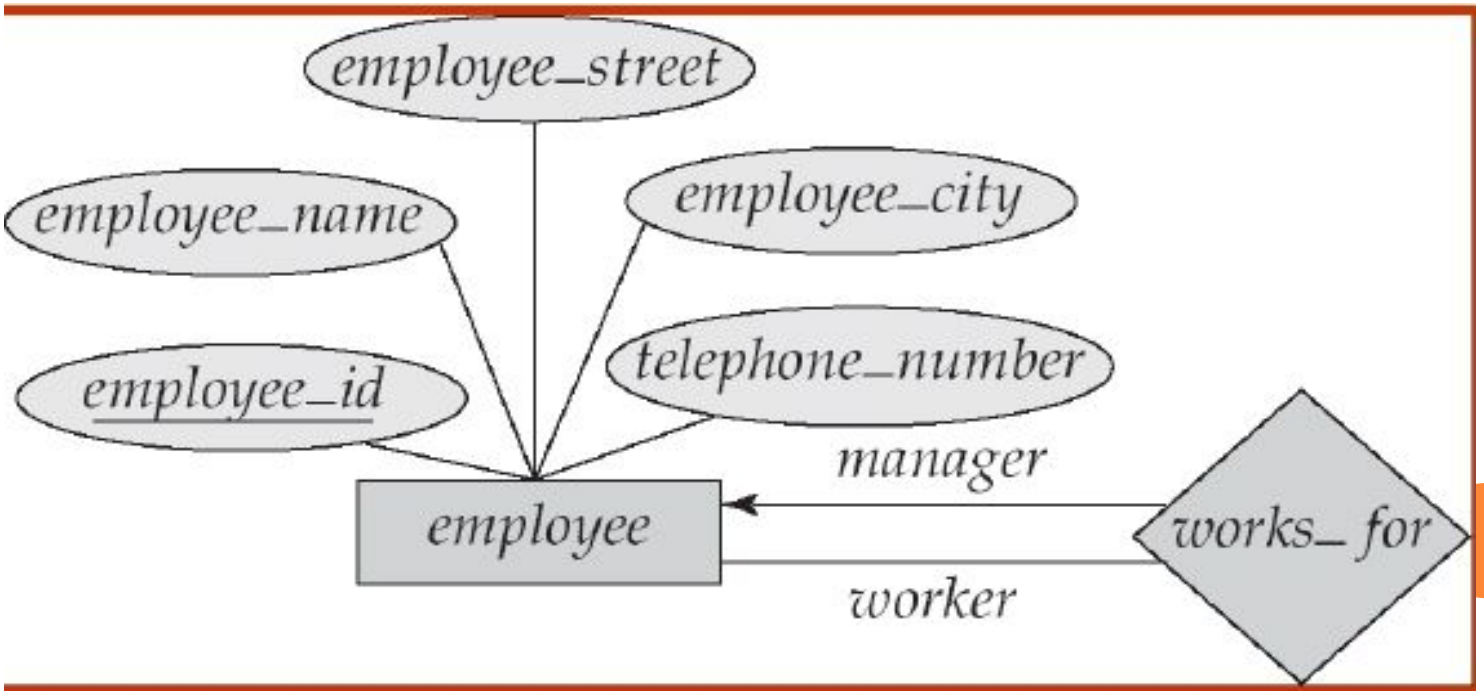


# Roles

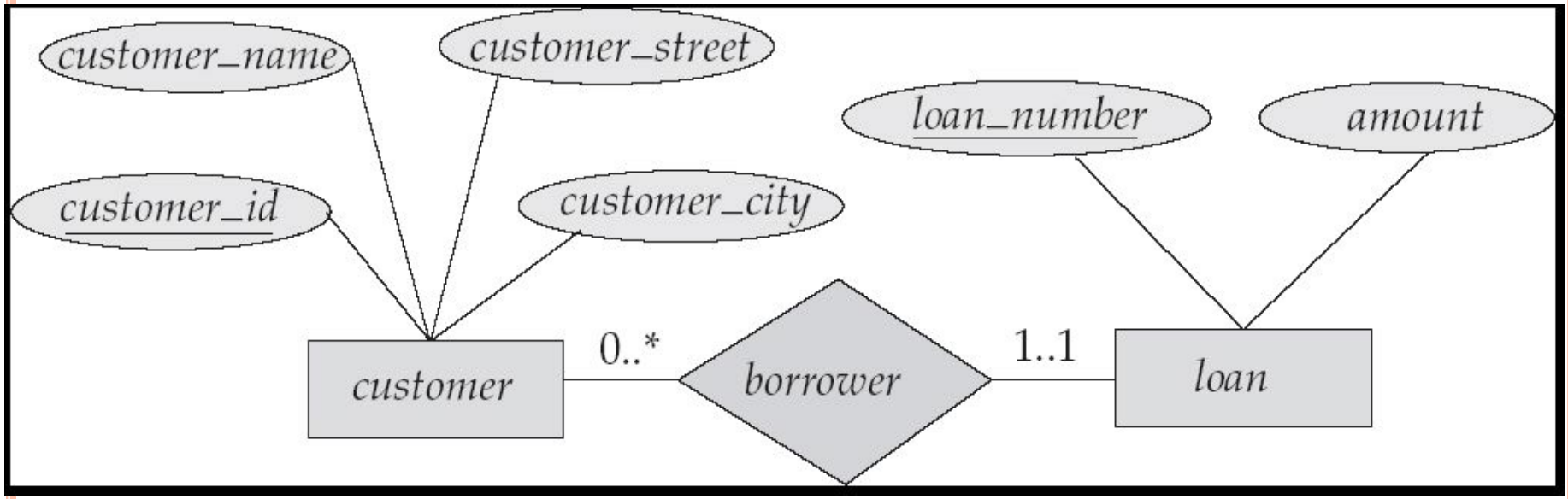
- A relationship can be between entities of one entity set; the two entities in such a relationship are in different **roles**.
  - E.g., Two employees in the employee entity set are involved in a works_for relationship. One has a role of "manager" and the other "worker".

# ER DIAGRAM WITH ROLE INDICATOR

- The labels "manager" and "worker" are called **roles; they specify how** employee entities interact via the works_for relationship set.
- Roles are indicated in ERdiagrams by labeling the lines that connect diamonds to rectangles.
- Role labels are optional, and are used to clarify semantics of the relationship



# Alternative Notation for Cardinallity Limits




An edge between entity set and relationship can have minimum and maximum cardinality shown in form of l..h , l minimum , h maximum

Minimum 1- indicates total participation

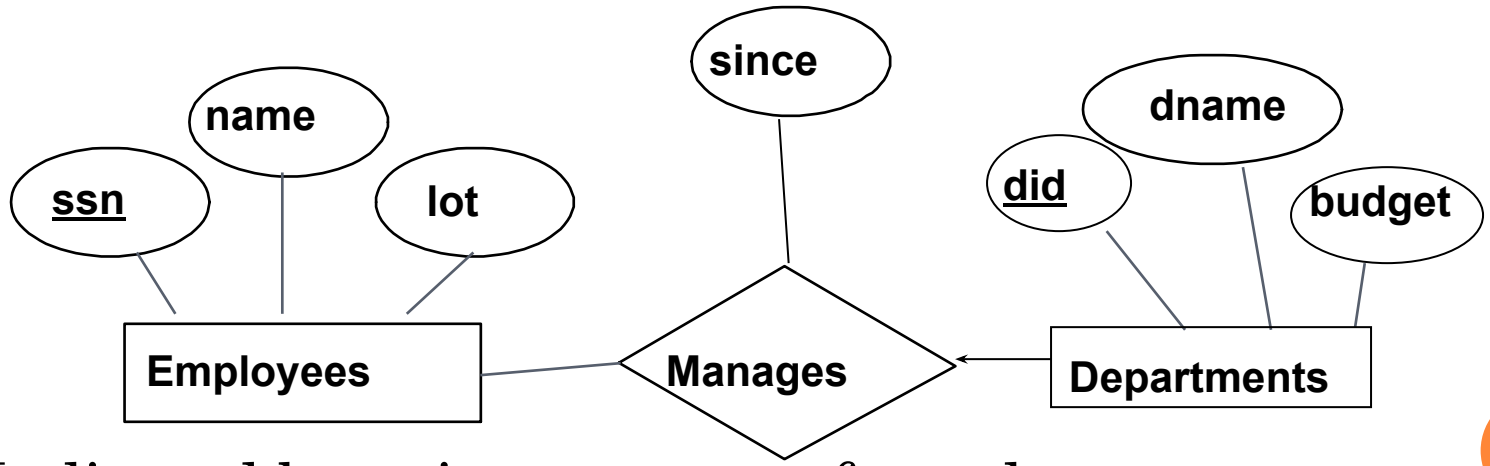
Maximum 1-indicates entity participate in at most 1 relationship

Maximum value * indicates no limit

# Key Constraint

- A key constraint between an entity set S and a relationship set restricts instances of the relationship set by requiring that each entity of S participate in at most one relationship

Eg: each dept has at most one manager, according to the *key constraint* on Manages.




Indicated by using an arrow from departments to manages

- Now consider relationship set called Manages between the Employees and Departments entity sets such that each department has at most one manager,
- Although a single employee is allowed to manage more than one department.
- The restriction that each department has at most one manager is an example of a **key constraint**, and it implies that each Departments entity appears in at most one Manages relationship in any allowable instance of Manages. This restriction is indicated in the ER diagram by using an arrow

# Weak entity set

- An entity set that does not have a primary key is referred to as a weak entity s.
- An entity set that has a primary key is termed strong entity set.

.

- The existence of a weak entity set depends on the existence of a Strong entity set
  - Identifying weak entity set is represented with double-line rectangle.
  - Identifying relationship is the relationship between Strong and weak entity sets and it is represented by double diamond.

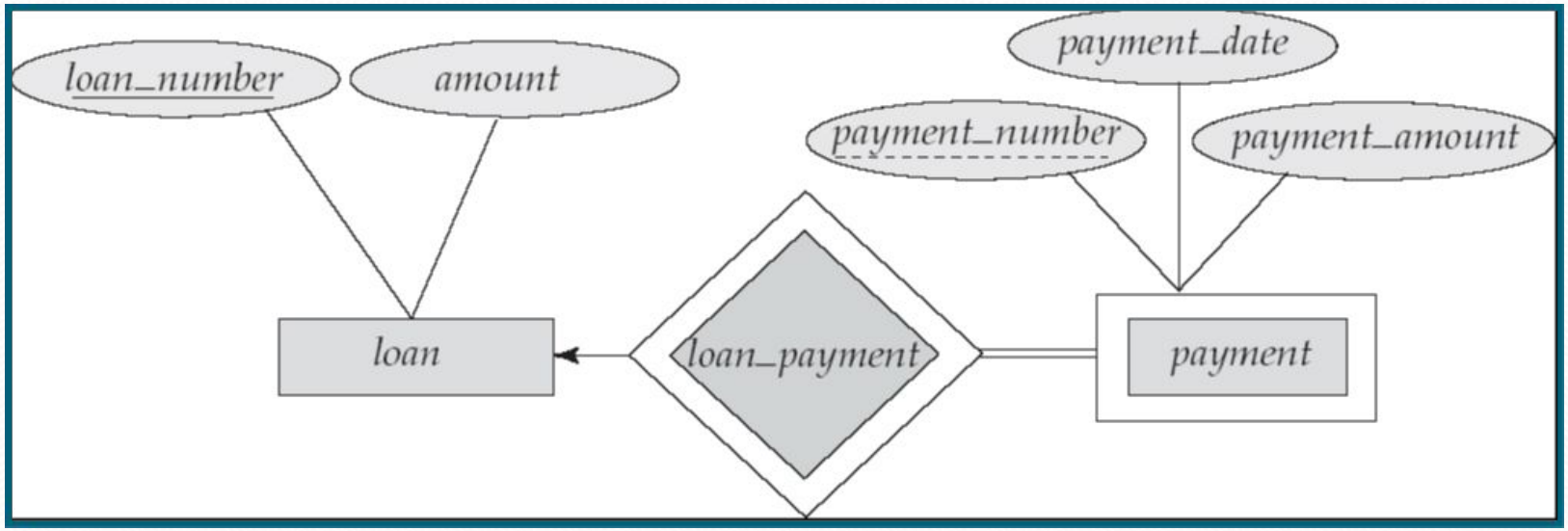
- The discriminator (*or partial key)* of a weak entity set is the set of attributes that distinguishes among all the entities of a weak entity set.

- 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 existence dependent, plus the weak entity set's discriminator.

# Identifying relationship

- We represent **weak entity sets by double rectangles**.
- We underline the discriminator of a weak entity set with a dashed line.

**payment_number – discriminator of the *payment* entity set**
**Primary key for *payment* – (*loan_number, payment_number*)**



# Additional features of ER model

# Class Hierarchies-Specialization & Generalization

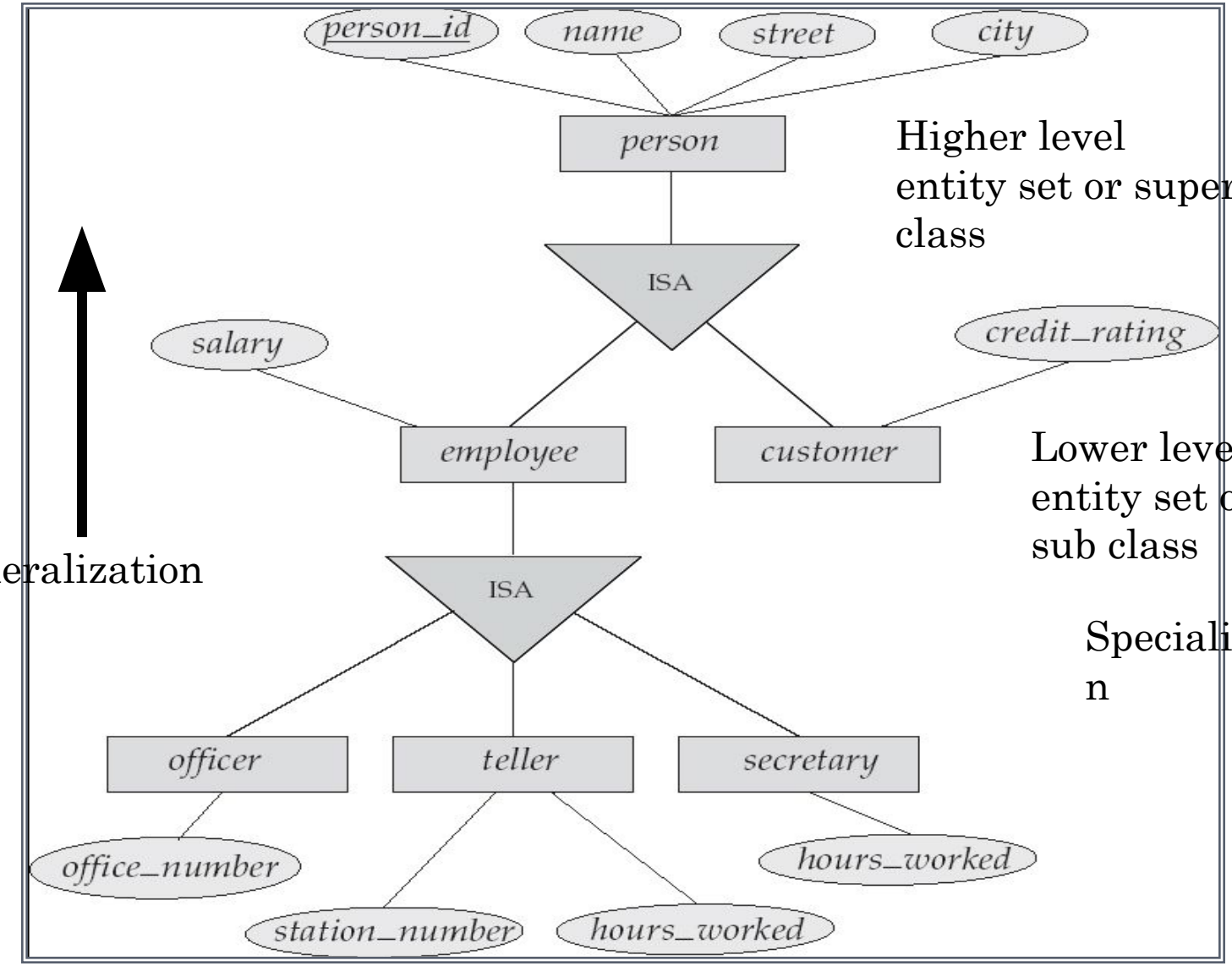
## Specialization:

An entity set may include sub groupings of entities that are distinct in some way from other entities. (i.e a subset of entities within an entity set may have attributes that are not shared by all entities in the entity set.)

## Generalization:

- **Generalization** is an abstraction mechanism, we can think of the reverse of the specialization process.
- Several classes with common features are generalized into a super class; original classes become its subclasses.
- **Example:** CAR, TRUCK generalized into VEHICLE; both CAR, TRUCK become subclasses of the super class VEHICLE.
  - We can view {CAR, TRUCK} as a specialization of VEHICLE
  - Alternatively, we can view VEHICLE as a generalization of CAR and TRUCK

# Specialization / generalization Example




Higher level entity set or super class

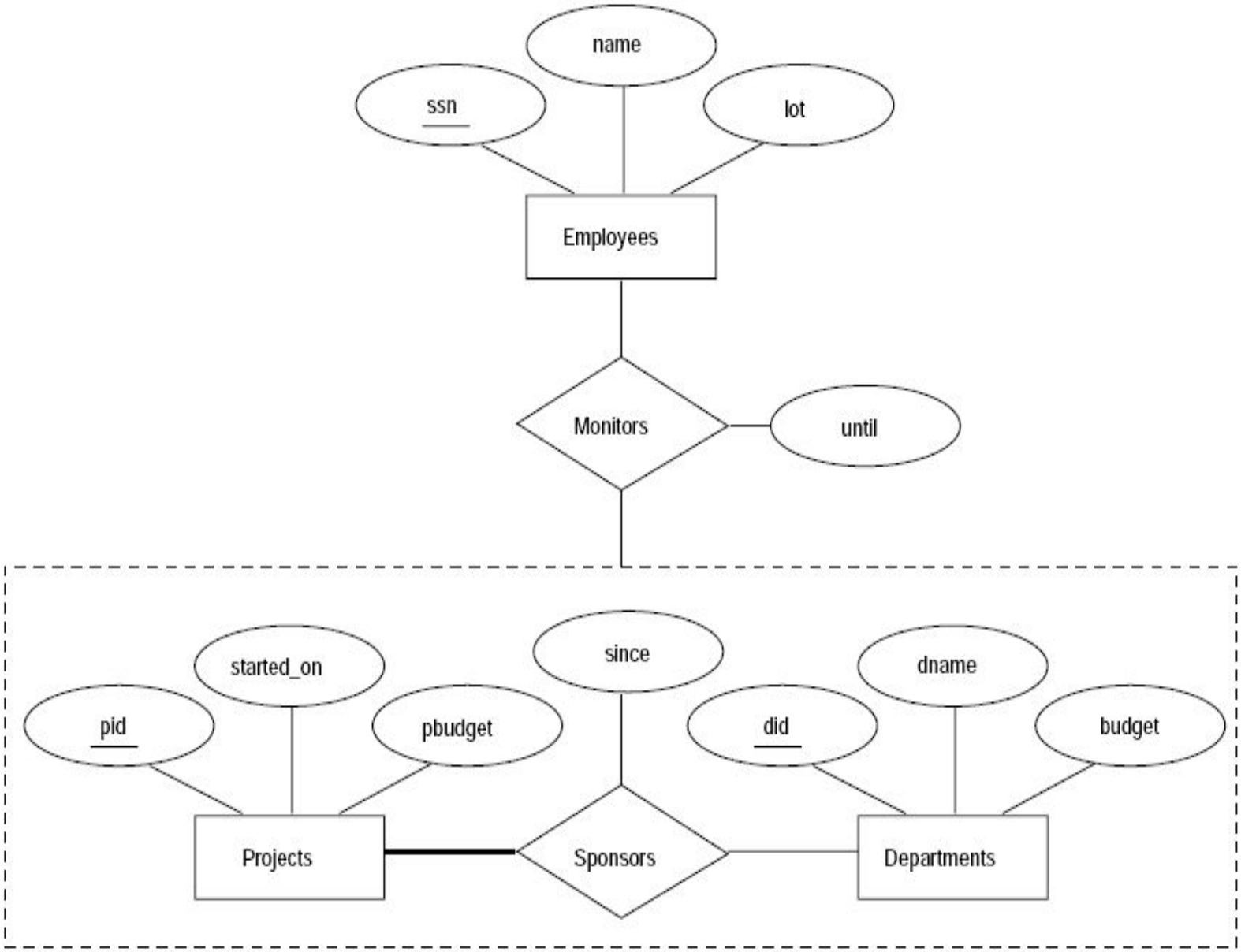
Lower level entity set or sub class

generalization

Specialization

# *Aggregation*

- A relationship set is an association between entity sets.
- Sometimes we have to model a relationship between a collection of entities and *relationships*.
- Suppose that we have an entity set called Projects and that each Projects entity is sponsored by one or more departments.
- The Sponsors relationship set captures this information.

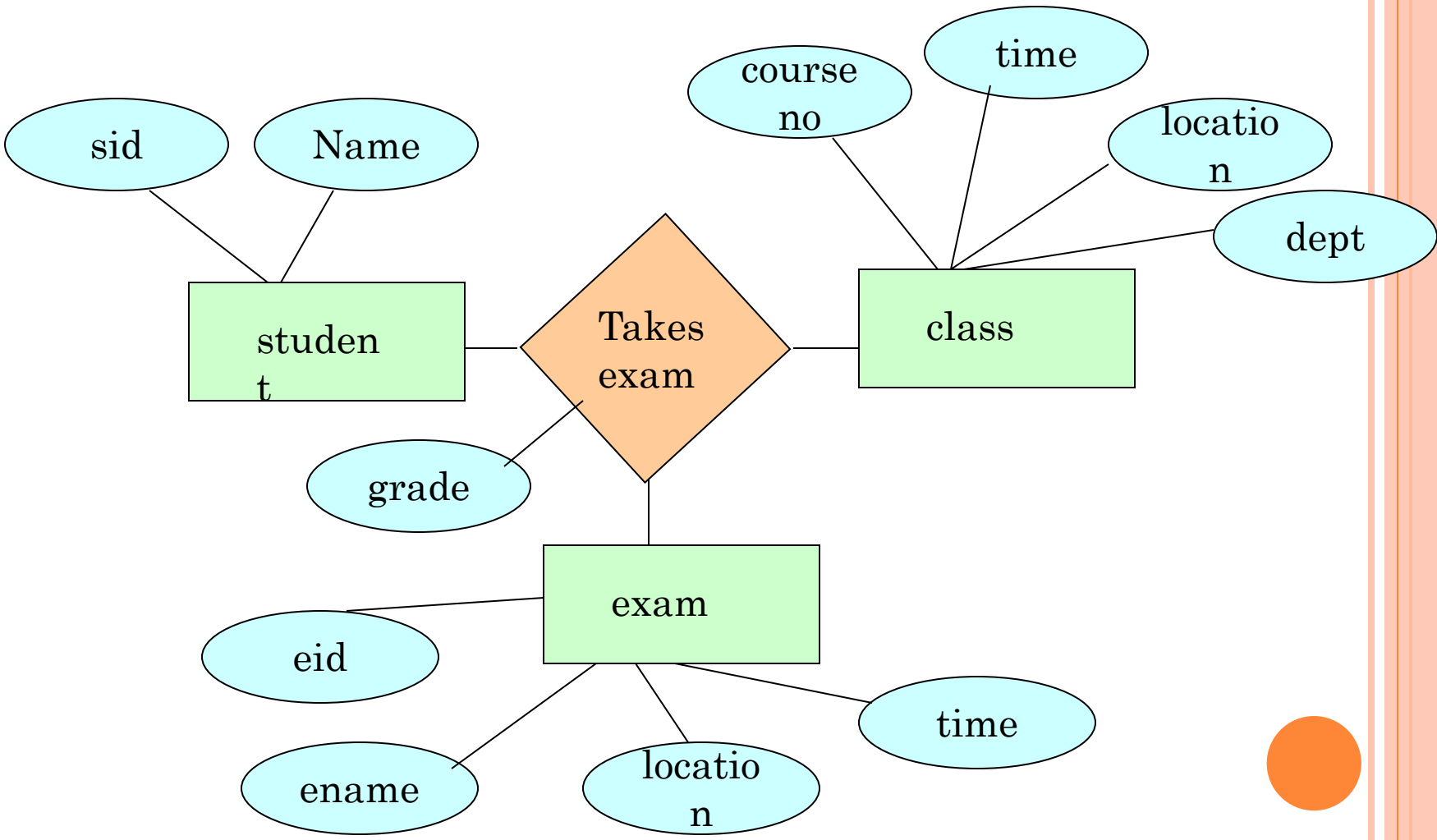

name
ssn
lot
Employees
Monitors
until
started_on
since
dname
pid
pbudget
did
budget
Projects
Sponsors
Departments

- A department that sponsors a project might assign employees to monitor the sponsorship.
- ie, Monitors should be a relationship set that associates a Sponsors relationship (rather than a Projects or Departments entity) with an Employees entity.
- However, we have defined relationships to associate two or more *entities*.

- In order to define a relationship set such as Monitors, we introduce a new feature of the ER model, called *aggregation.*
- A dashed box is used to illustrate aggregation.
- *Aggregation allows us to indicate that a relationship*set (identified through a dashed box) participates in another relationship set.

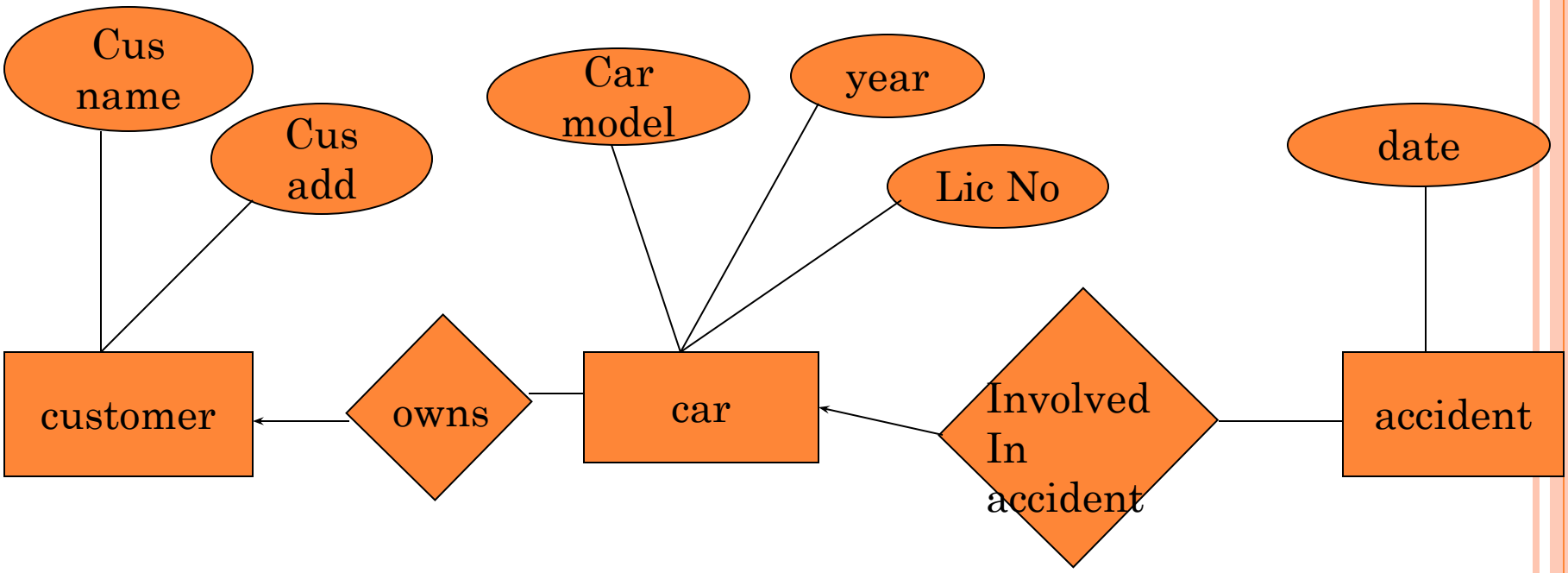
# ER diagram

Consider a database used to record the marks that students get in different exams of different course offerings. Construct an E-R diagram for the database modeling exams as entities and using a ternary relationship.

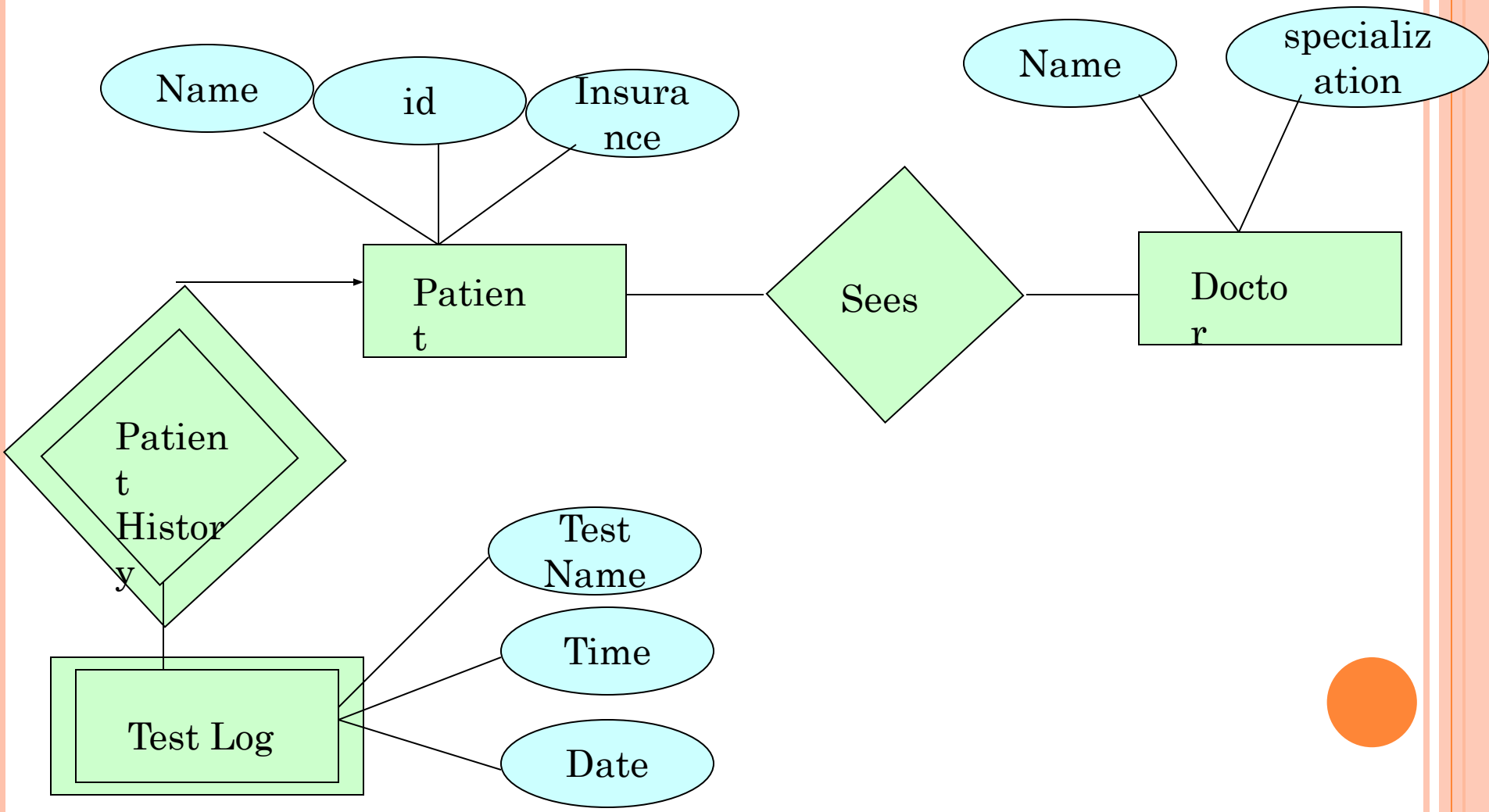

sid
Name
course no
time
location
dept
student
Takes exam
class
grade
exam
eid
time
ename
location

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.

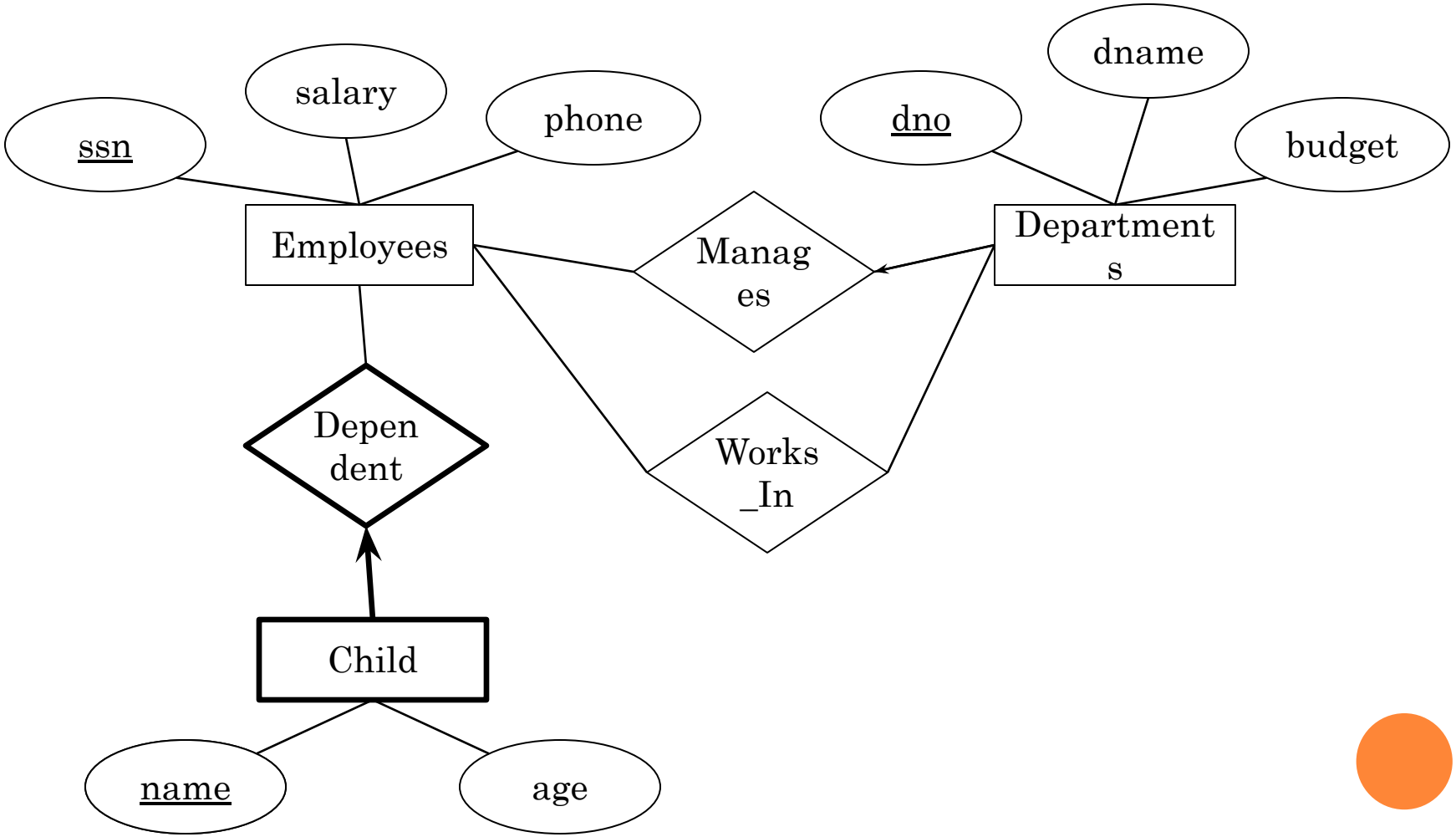
Construct an ER diagram for a car insurance company whose customers own one or more cars each. Each car has associated with it zero to any number of recorded accidents.



# Construct an E-R diagram for a hospital with a set of patients and a set of medical doctors. Associate with each patient a log of the various tests and examinations conducted.


Name
id
Insura
nce
Patien
t
Sees
Name
specializ
ation
Docto
r
Patien
t
Histor
y
Test Log
Test
Name
Time
Date

- A company database needs to store information about employees(identitfied by ssn with salary and phone attributes),departments(identified with dno with dname and budget as attributes) and children of employees(name,age).employee works in department.each department is *managed by* an employee.a child must identified uniquely by *name* when parent(who is an employeea ssume that only one parent works for the company) is known


dname
salary
dno
ssn
phone
budget
Employees
Manag
es
Department
s
Depen
dent
Works
_In
Child
name
age