ER Modeling II

ER-Relationship Degree

Relationship degree refers to the number of associated entities.

The relationship degree can be broadly classified into unary, binary, and ternary relationship

Unary Relationship

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

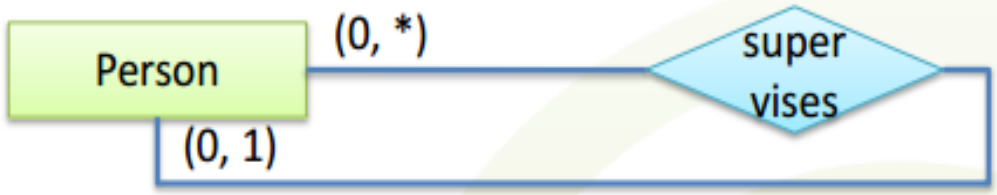
An entity related to itself is known as recursive relationship

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.

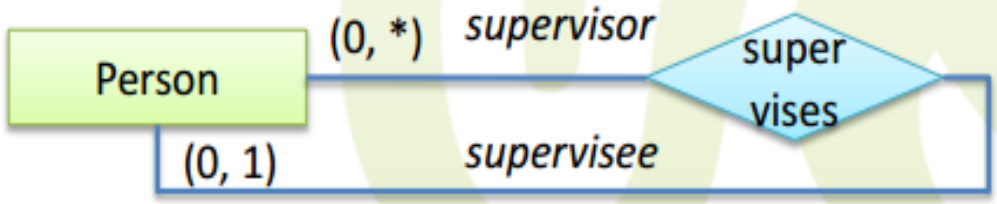
ER-Relationship Degree

– e.g. relationship *supervises*



– what is meant? Who is the supervisor? Who is the supervised person?

– roles can be annotated on the relationship lines



ER-Relationship Degree

Binary Relationship

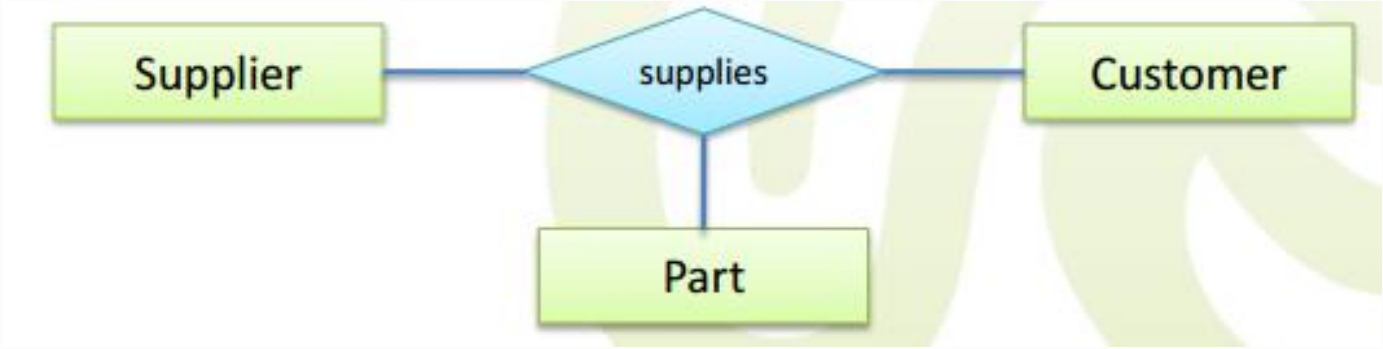
In a binary relationship, two entities are involved.

ER-Relationship Degree

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.



Participation

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

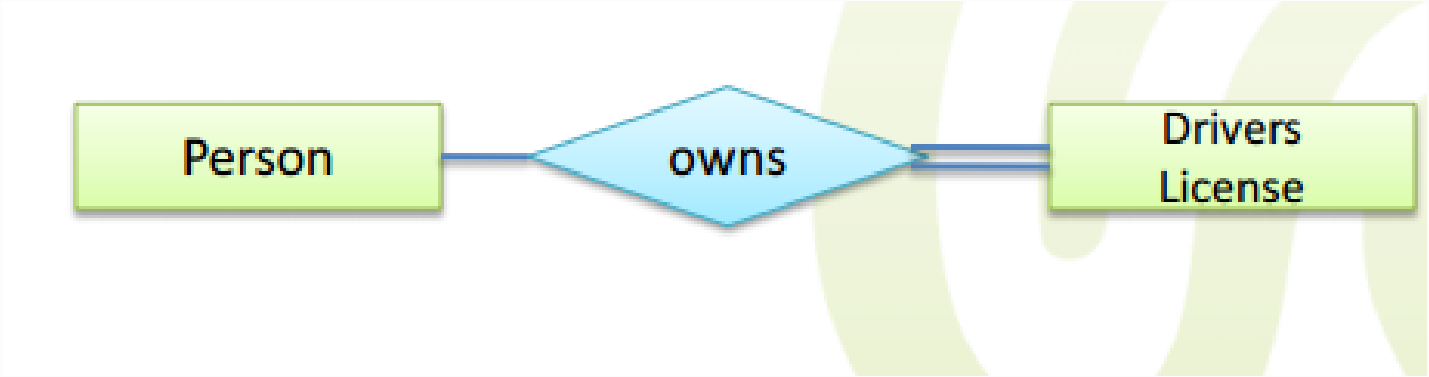
E.g. participation of customer in borrower is partial

Participation

To express that all entities of an entity type appear in a certain relationship set, the concept of total participation can be used

– the entity type which is totally participating is indicated by a double line

– e.g. Each driver's license must belong to an exact person



Weak Entity

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 a strong entity set; it must relate to the strong set via a one-to-many relationship set.

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.

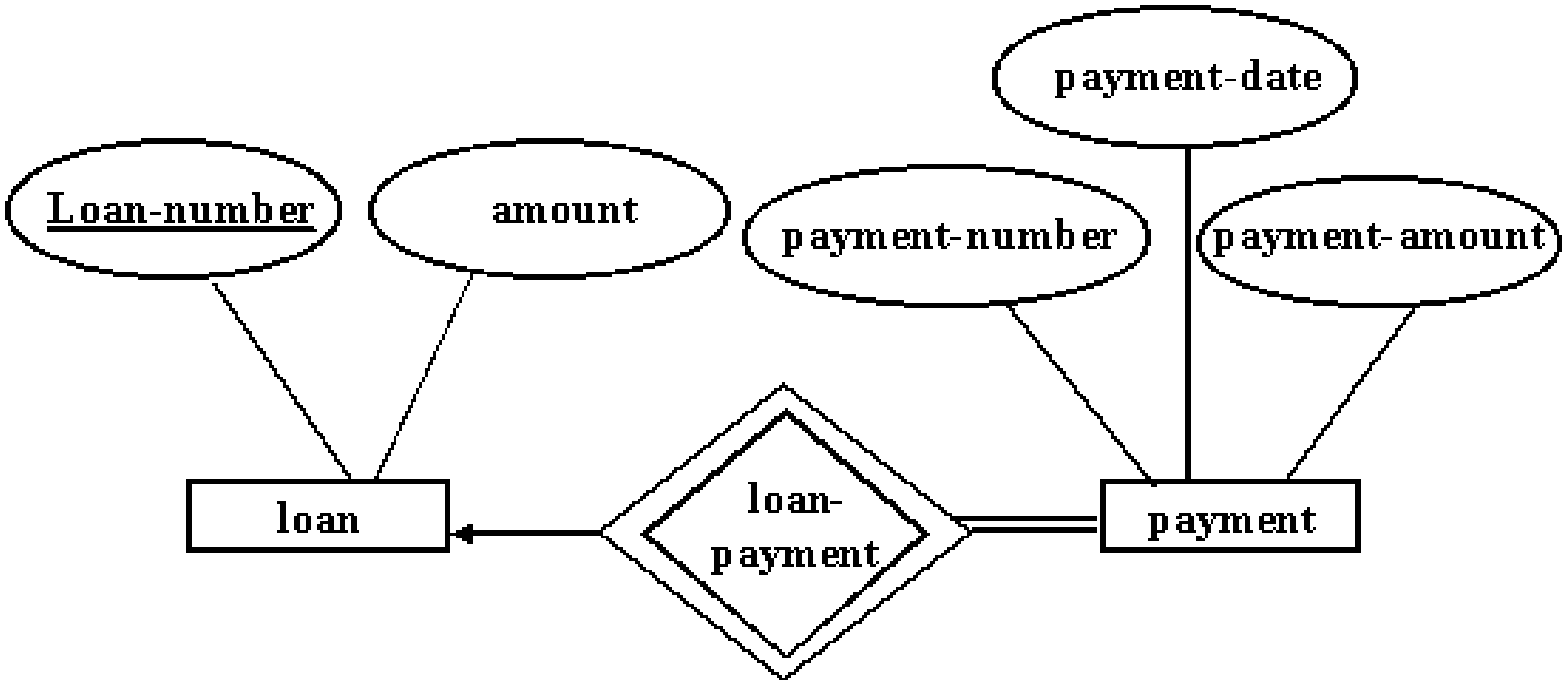
Weak Entity

A weak entity set represented by double rectangles.

The discriminator of a weak entity set represented with a dashed line.

- payment-number– discriminator of the payment entity set
- Primary key for payment–(loan-number, payment-number)

WEAK ENTITY SET

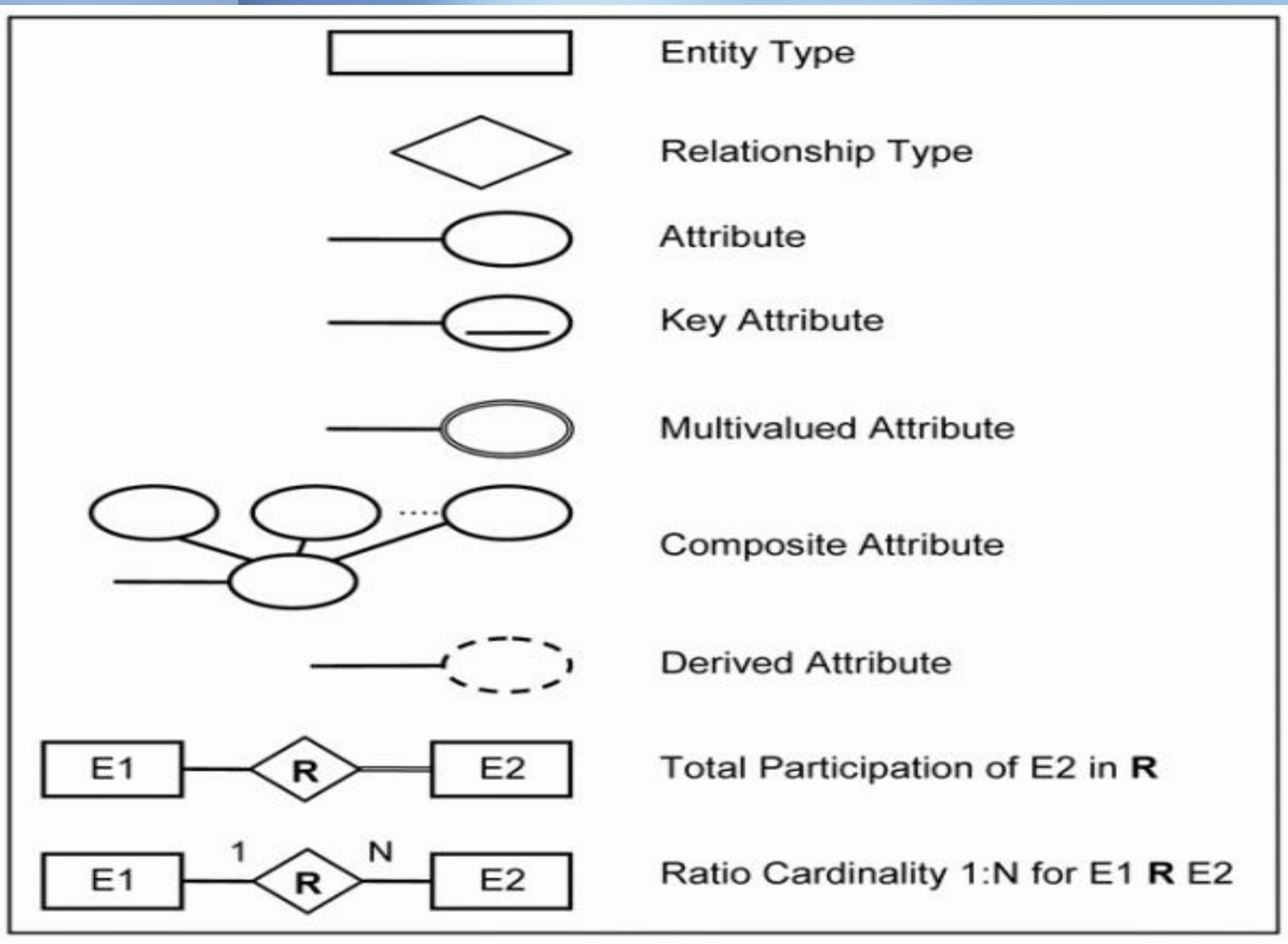


Weak Entity

– Example

- *An online shopping order contains several order items.*

- an *order item* can only exit within an *order*
- each *order item* can be identified by the *order no* of it's owning *order* **and** its *item line*


Entity Type
Relationship Type
Attribute
Key Attribute
Multivalued Attribute
Composite Attribute
Derived Attribute
E1
R
E2
Total Participation of E2 in R
1
N
E1
R
E2
Ratio Cardinality 1:N for E1 R E2

ER Structure

English grammar structure ER structure

Common noun Entity type

Proper noun Entity

Transitive verb Relationship type

Intransitive verb Attribute type

Adjective Attribute for entity

Adverb Attribute for relationship

Associative Entity

- An **associative entity** is an element of the entity–relationship model. The database relational model does not offer direct support to many-to-many relationships, even though such relationships happen frequently in normal usage. The solution to this problem is the creation of another table to hold the necessary information for this relationship. This new table is called an **associative entity**.

- An associative entity can be thought of as both an entity and a relationship since it encapsulates properties from both. It is a relationship since it is serving to join two or more entities together, but it is also an entity since it may have its own properties. The associative entity must have the primary keys of both adjoining tables as identifiers, but may also contain its own unique identifier and other information about the relationship

Associative Entity

Example I

- A college contains many departments
- Each department can offer any number of courses
- Many instructors can work in a department
- An instructor can work only in one department
- For each department there is a Head
- An instructor can be head of only one department
- Each instructor can take any number of courses
- A course can be taken by only one instructor
- A student can enroll for any number of courses
- Each course can have any number of students

Example I

Identify the Entities

What are the entities here?

From the statements given, the entities are

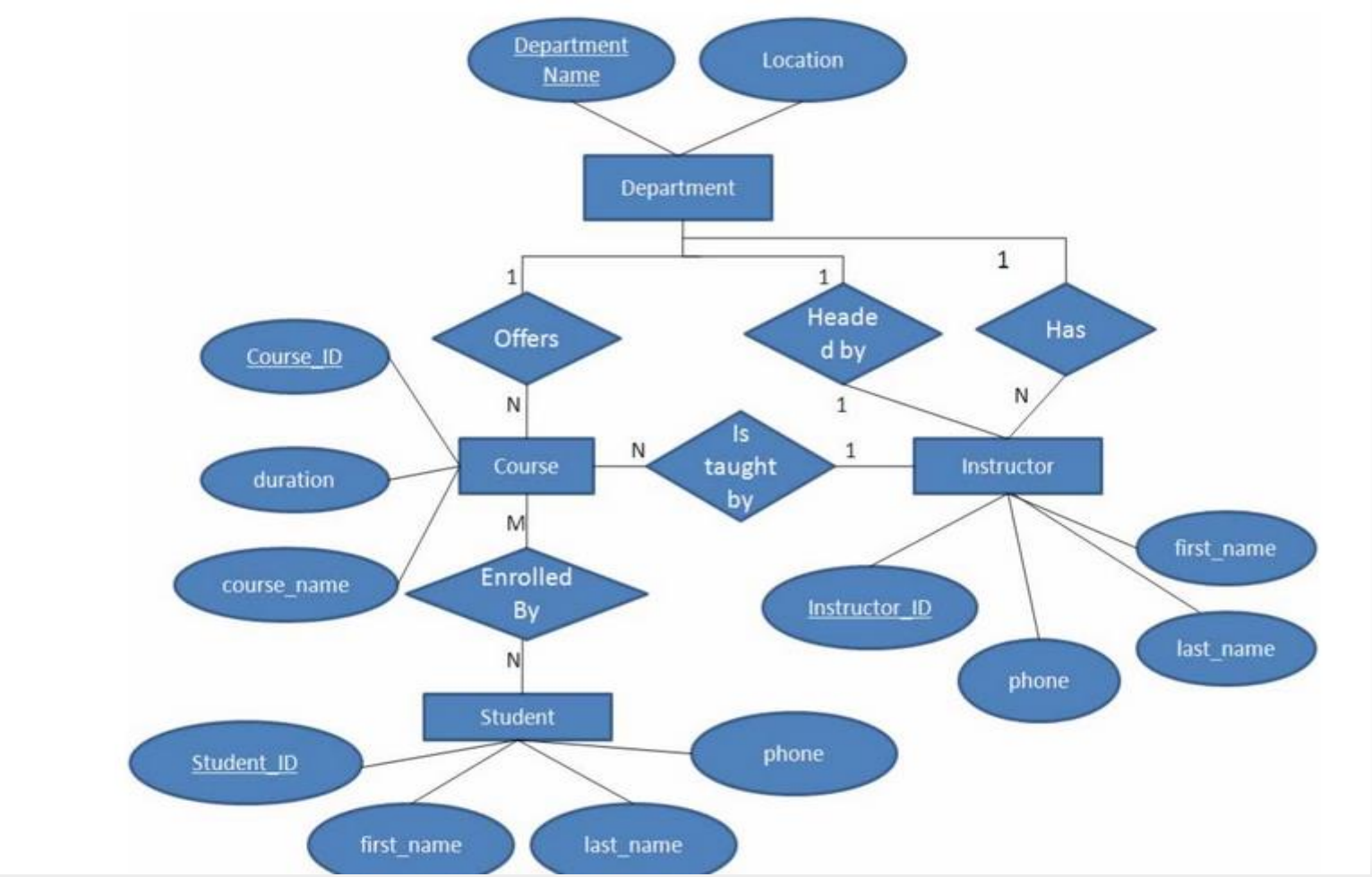
- Department
- Course
- Instructor
- Student

Example I

Identify the relationships

- One department offers many courses. But one particular course can be offered by only one department. hence the cardinality between department and course is One to Many (1:N)
- One department has multiple instructors . But instructor belongs to only one department. Hence the cardinality between department and instructor is One to Many (1:N)
- One department has only one head and one head can be the head of only one department. Hence the cardinality is one to one. (1:1)
- One course can be enrolled by many students and one student can enroll for many courses. Hence the cardinality between course and student is Many to Many (M:N)
- One course is taught by only one instructor. But one instructor teaches many courses. Hence the cardinality between course and instructor is Many to One (N :1)

ER-Model-I



Example II

- A company database needs to store information about employees (identified by *ssn, with salary and phone)departments (identified by dno,* with *dname and budget), and children of employees (with name and age*).

- Employees *work in departments; each department is managed by an* employee; a child must be identified uniquely by *name when the parent (who is an* employee; assume that only one parent works for the company) is known. We are not interested in information about a child once the parent leaves the company.

Example II

- A company database needs to store information about employees (identified by *ssn, with salary and phone), departments (identified by dno,* with *dname and budget), and children of employees (with name and age*).

- Employees *work in departments; each department is managed by an* employee; a child must be identified uniquely by *name when the parent (who is an* employee; assume that only one parent works for the company) is known. We are not interested in information about a child once the parent leaves the company.

Example II

- A company database needs to store information about employees (identified by *ssn, with salary and phone), departments (identified by dno,* with *dname and budget as attributes), and children of employees (with name and age*).

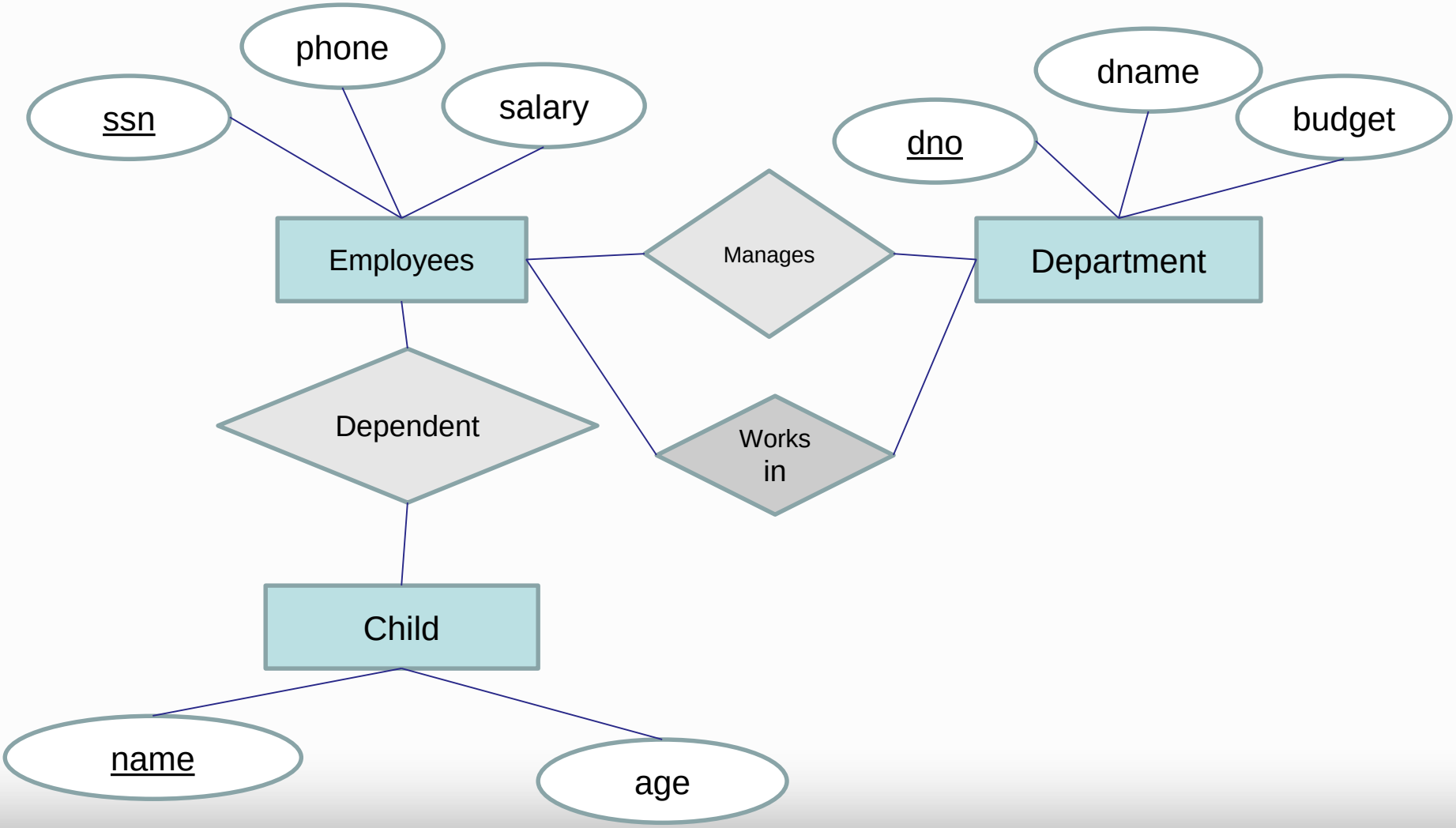
- Employees *work in departments; each department is managed by an* employee; a child must be identified uniquely by *name when the parent (who is an* employee; assume that only one parent works for the company) is known. We are not interested in information about a child once the parent leaves the company.

Example II

- A company database needs to store information about employees (identified by *ssn, with salary and phone), departments (identified by dno,* with *dname and budget as attributes), and children of employees (with name and age*).

- Employees *work in departments; each department is managed by an* employee; a child must be identified uniquely by *name when the parent (who is an* employee; assume that only one parent works for the company) is known. We are not interested in information about a child once the parent leaves the company.

ER-Model-II



Example III

- Galleries keep information about artists, their names (which are unique), birthplaces, age, and style of art. For each piece of artwork, the artist, the year it was made, its unique title, its type of art (e.g., painting, sculpture, photograph), and its price must be stored. Pieces of artwork are also classified into groups of various kinds, for example, portraits works by Picasso, or works of the 19th century; a given piece may belong to more than one group.
- Each group is identified by a name (like those just given) that describes the group. Finally, galleries keep information about customers. For each customer, galleries keep that person's unique name, address, total amount of dollars spent in the gallery (very important!), and the artists and groups of art that the customer tends to like.
- Draw the ER diagram for the database.

Example III- Entites

- Galleries keep information about artists, their names (which are unique), birthplaces, age, and style of art. For each piece of artwork, the artist, the year it was made, its unique title, its type of art (e.g., painting, sculpture, photograph), and its price must be stored. Pieces of artwork are also classified into groups of various kinds, for example, portraits works by Picasso, or works of the 19th century; a given piece may belong to more than one group.
- Each group is identified by a name (like those just given) that describes the group. Finally, galleries keep information about customers. For each customer, galleries keep that person's unique name, address, total amount of dollars spent in the gallery (very important!), and the artists and groups of art that the customer tends to like.
- Draw the ER diagram for the database.

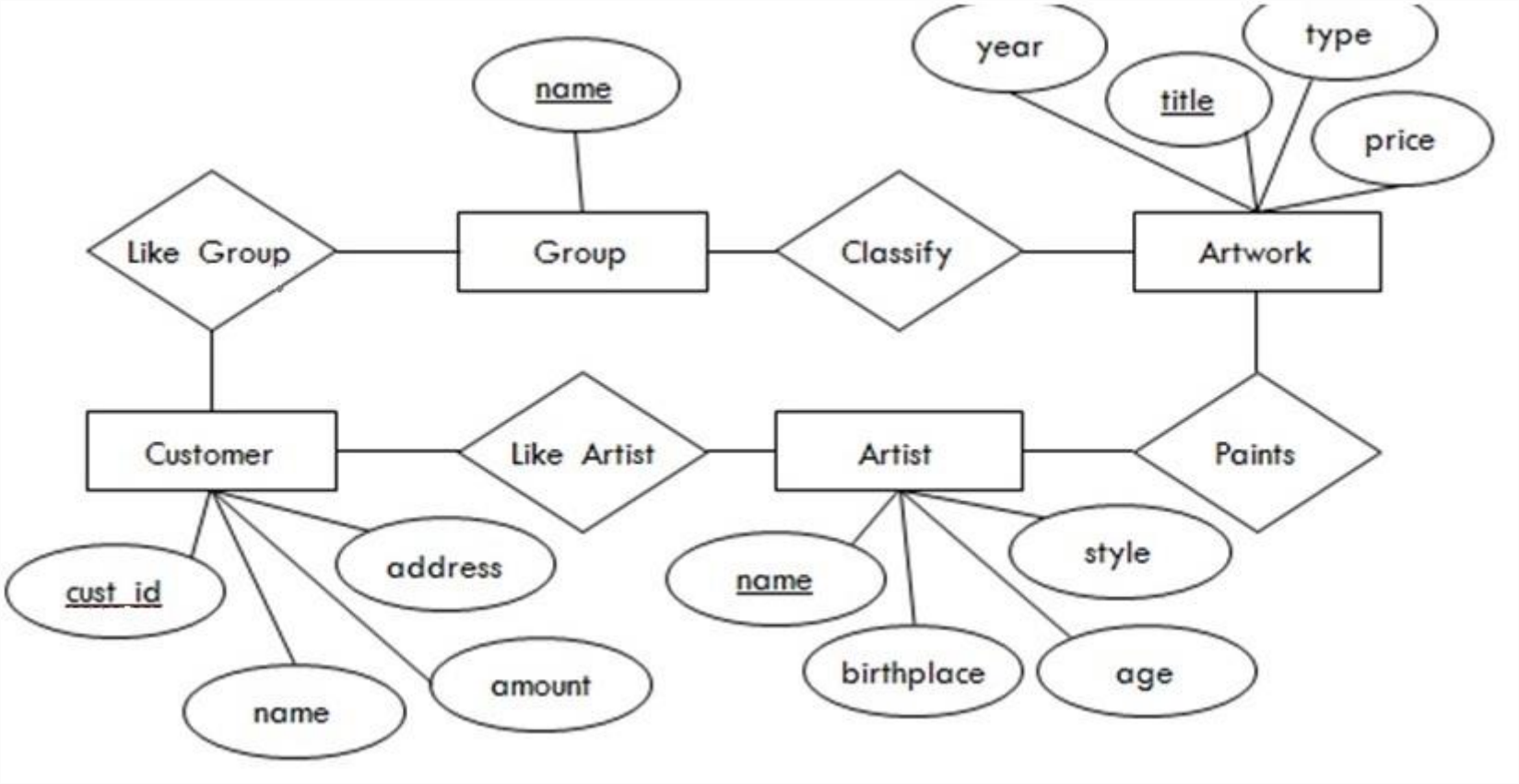
Example III- Attributes

- Galleries keep information about artists, their names (which are unique), birthplaces, age, and style of art. For each piece of artwork, the artist, the year it was made, its unique title, its type of art (e.g., painting, sculpture, photograph), and its price must be stored. Pieces of artwork are also classified into groups of various kinds, for example, portraits works by Picasso, or works of the 19th century; a given piece may belong to more than one group.
- Each group is identified by a name (like those just given) that describes the group. Finally, galleries keep information about customers. For each customer, galleries keep that person's unique id, name, address, total amount of dollars spent in the gallery (very important!), and the artists and groups of art that the customer tends to like.
- Draw the ER diagram for the database.

Example III- Attributes

- Galleries keep information about artists, their names (which are unique), birthplaces, age, and style of art. For each piece of artwork, the artist, the year it was made, its unique title, its type of art (e.g., painting, sculpture, photograph), and its price must be stored. Pieces of artwork are also classified into groups of various kinds, for example, portraits works by Picasso, or works of the 19th century; a given piece may belong to more than one group.
- Each group is identified by a name (like those just given) that describes the group. Finally, galleries keep information about customers. For each customer, galleries keep that person's unique id,name, address, total amount of dollars spent in the gallery (very important!), and the artists and groups of art that the customer tends to like.
- Draw the ER diagram for the database.

ER-Model-III



Case Study I-*DIY*

In our database, we have students. They have a name, a registration number, and a course of study. The university offers lectures. Each lecture may be part of some course of study in a certain semester. Lectures may have other lectures as prerequisites. They have a title, provide a specific number of credits and have a unique ID Each year, some of these lectures are offered by a professor at a certain day at a fixed time in a specific room. Students may register for that lecture. Professors have a name and are member of a specific department

Case Study II-*DIY*

The craft trading website ABC is setting up a database to record sellers and their products.

This requires recording the following information:

- For each seller, their name, contact email, and postal address.
- For each product, its name, price, and number available.
- Which product is from which seller.
- A unique id number for each product.

Draw an entity-relationship (ER) diagram that represents this information.

Make sure to capture the constraints on the relationships involved, and designate appropriate primary keys for the entities.