HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

IT3090E - Databases

Chapter 6: Database modelling – Entity / Relationship Model

Muriel VISANI
murielv@soict.hust.edu.vn

Outline

- Introduction
- How to create an ERD
- Mapping from ERD to relational schema

Objectives

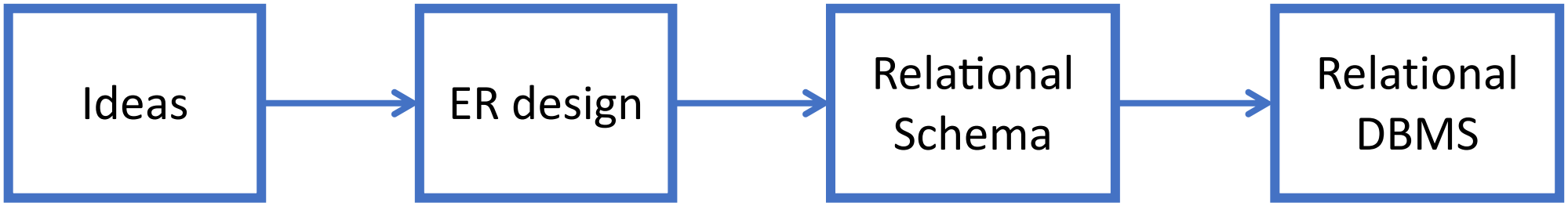
- Upon completion of this lesson, students will be able to:
  - Know what the entity relationship model is
  - Know how to create an ERD from a real-world problem
  - Transform from ERD into relational schema

1. Introduction

- Introduction
- Entity sets
- Attributes
- Relationships
- Key
- Different possible notations

1.1. Introduction

- Two approaches to DB designing
  - Bottom – up: Functional Dependencies and Normalization (chapter 5)
  - Top - down: Entity Relationship model (ER: this chapter)
- ER model is used in DB design

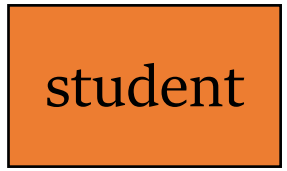


1.1. Introduction

- ERD is used to represent the structure of data graphically
- Three principal types of elements:
  - Entity sets
  - Attributes
  - Relationships

1.2. Entity sets

- Entity
  - is a “thing” (object) in the real world with an independent existence
  - An entity may be:
    - an object with a physical existence (a particular person, car, house, or employee), or
    - an object with a conceptual existence (a company, a job, or a university course)
- Entity sets
  - a collection of similar entities forms an entity set.
  - In ERD, rectangular boxes represent for entity sets



1.2. Entity sets

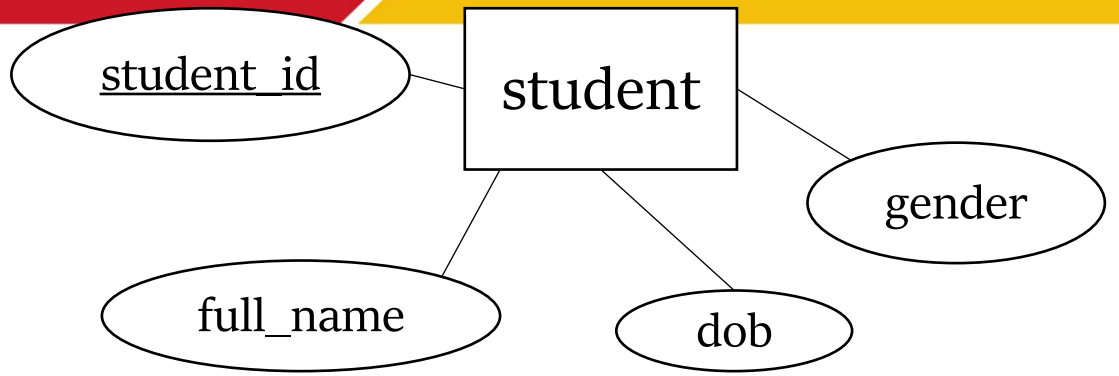
- Weak entity type vs. strong entity type
  - a **srong entity** set is an entity set that can be uniquely identified by its attributes.
  - a **weak entity** set is an entity set that cannot be uniquely identified by its attributes alone.
    - Therefore, it must use a foreign key (possibly in conjunction with its own attributes) to create a primary key.
    - The foreign key is typically a primary key of an entity it is related to
    - **Weak entities** are represented with **double rectangular** box in the ER Diagram

1.2. Entity sets

- Example: let's consider the following schema (from a previous lecture)
  - Which are the strong entity sets?
  - Which are the weak entity sets?

SHOPS (no_shop, city, name_manager)
CUSTOMERS (no_customer, name, country, city, type)
ITEMS(no_item, name, weight, color, qty_stock, BuyingPrice, SellingPrice, no_supplier#)
DELIVERY (no_delivery, date_del, no_customer#, no_shop#)
DELIVERY_DETAILS (no_delivery#, no_item#, qty, no_order#)
ORDERS (no_order, date, no_customer#, no_shop#)
ORDER_DETAILS (no_order#, no_item#, qty, no_delivery#, total_amount)
SUPPLIERS (no_supplier, supplier_name)

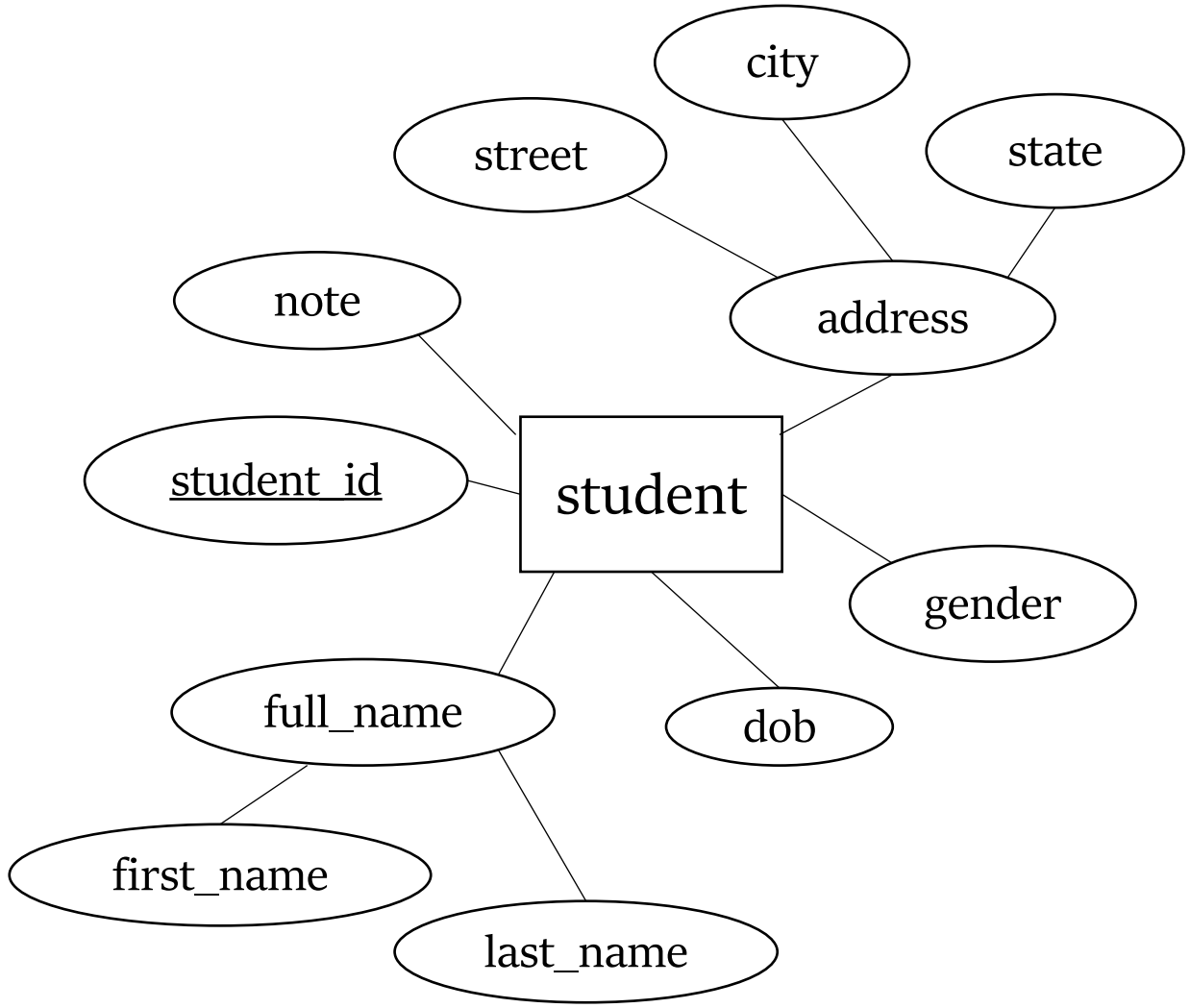
1.3. Attributes



- Attributes
  - Entity sets have associated attributes, which are properties of the entities in that set.
  - For instance, each entity "student" has some properties such as student_id, first_name, last_name, dob, gender, address, and so on.
  - In ERD, ovals (ellipses) are used to represent attributes
- Value domain of an attribute
  - Each simple attribute of an entity set is associated with a value set (or value domain).
  - For example: domain(gender) = {male, female}; domain(dob) = {date}; domain(last_name) = {char(30)}.

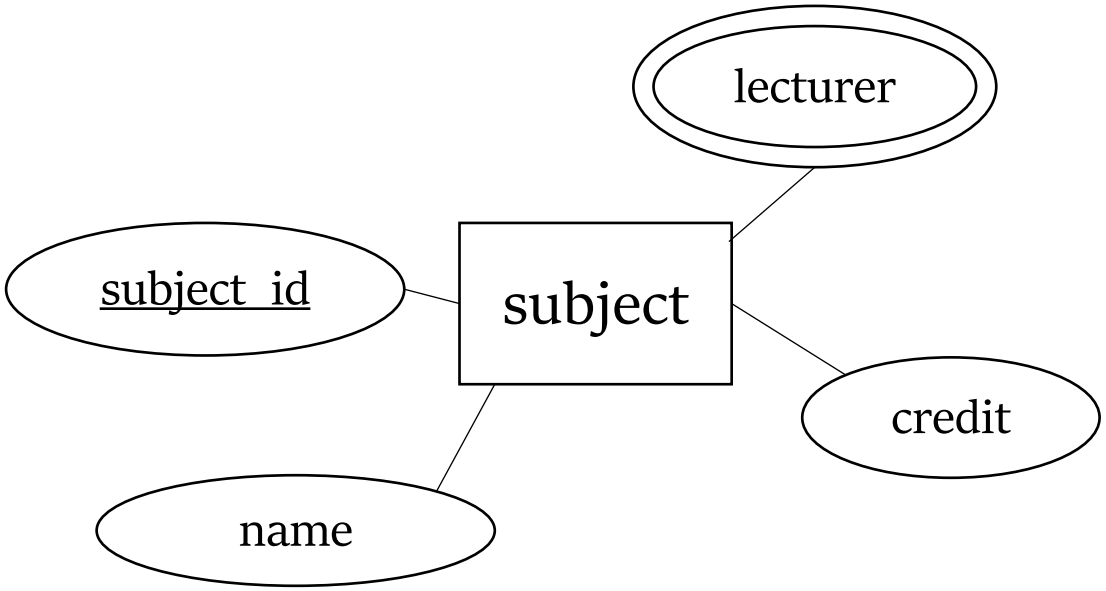
1.3. Attributes

- Some types of attributes
  - Simple/atomic attributes: Attributes that are not divisible.
  - Composite attributes: attributes that can be divided into smaller subparts, which represent more basic attributes with independent meanings.



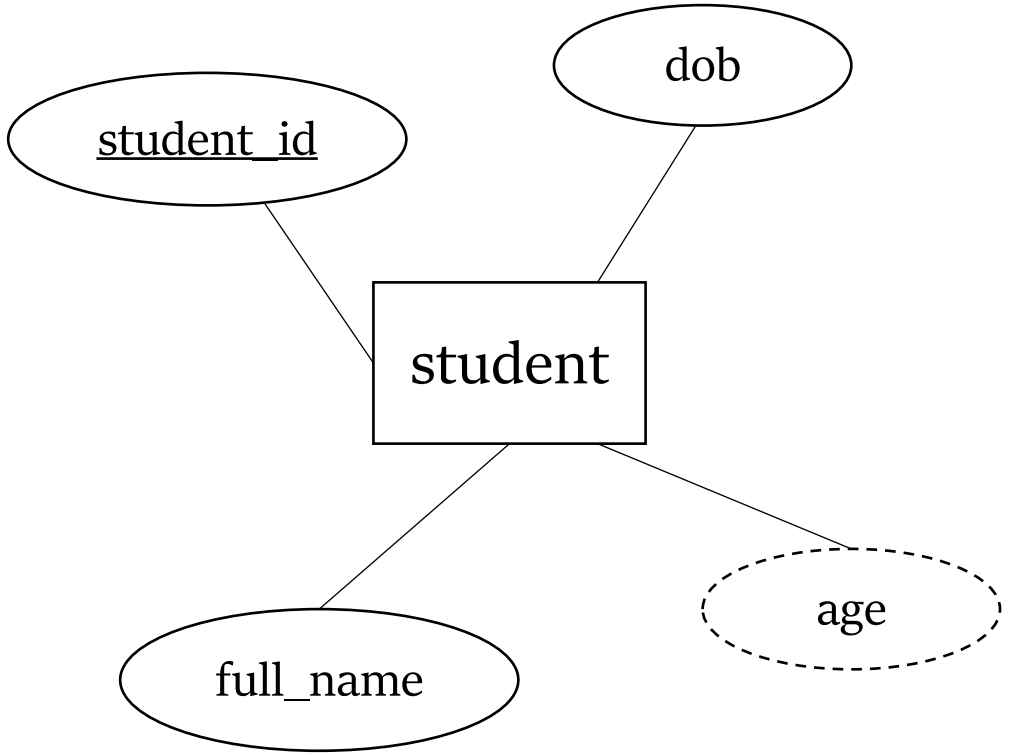
1.3. Attributes

- Some types of attributes
  - Single-valued attributes: have a single value for a particular entity
  - Multi-valued attributes: can have different numbers of values
    - Are denoted by double ovals in the ER diagram



1.3. Attributes

- Some types of attributes
  - Stored attributes vs. Derived attributes:
    - age attribute is called a derived attribute and is said to be derivable from the dob attribute (and the current date).
      - Derived attributes are denoted by dashed ovals in the ER diagram
    - The dob attribute is, on the other hand, a stored attribute.

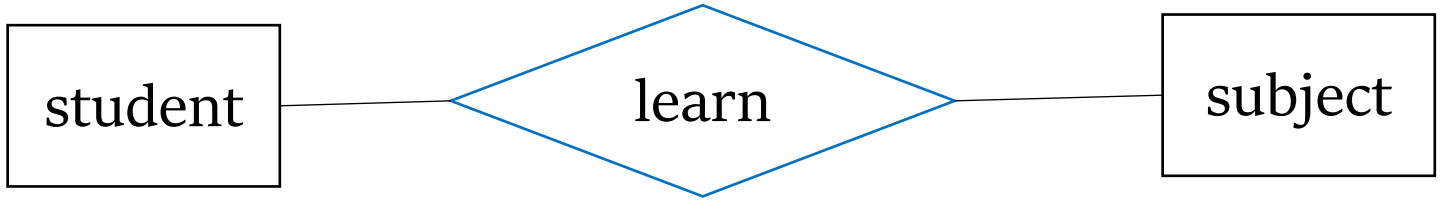


1.3. Attributes

- Comments:
  - A relation in 1st Normal Form (1NF) is defined formally as a relation where:
    - the attributes are simple/atomic
    - the attributes are single-valued
    - because we consider that avoiding redundancy is one of the main objectives of normalization, most French authors also consider that the 1NF also entitles that there are only stored attributes in the relation (no derived attributes)
      - Not a part of the formal definition, more a semantic normalization that we apply from the beginning
  - The ER model can be used to represent entity sets (implemented as relations in the database) which do not follow any normal form

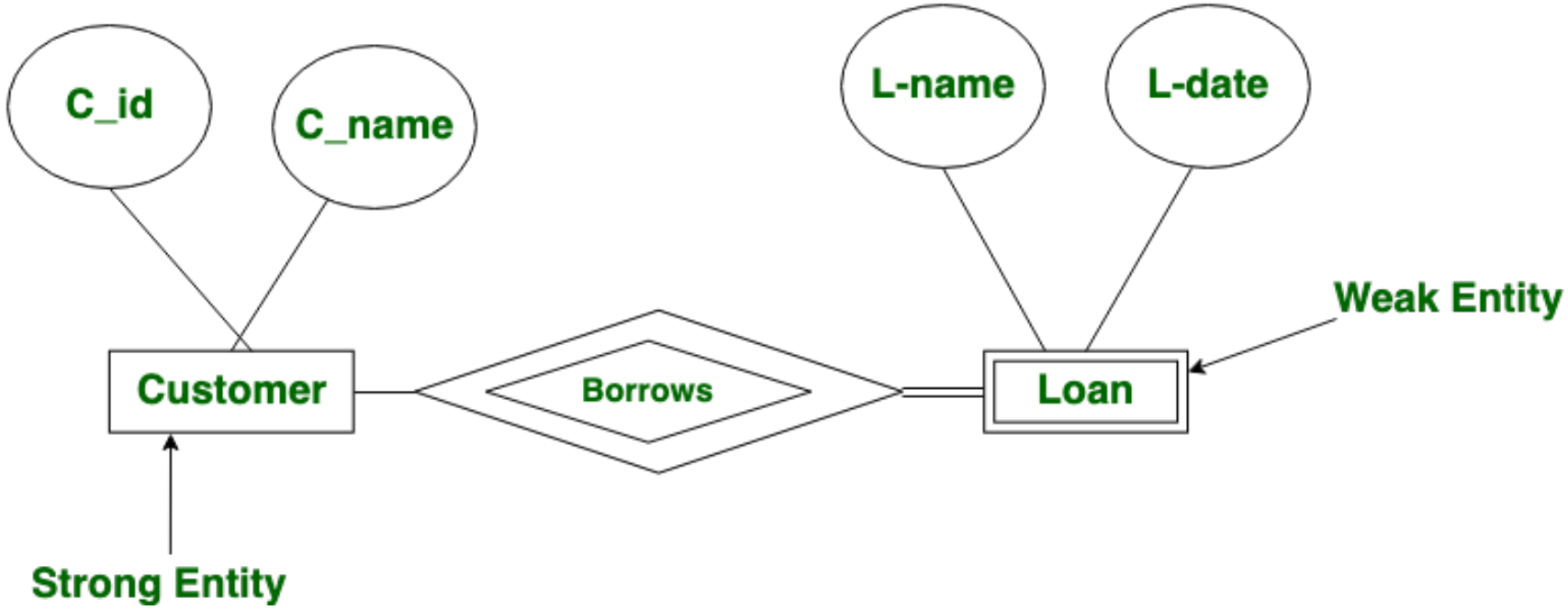
1.4. Relationships

- Relationships are connections among two or more entity sets.
- In ER diagrams, relationships are displayed as diamond-shaped boxes.
  - which are connected by straight lines to the rectangular boxes representing the participating entity types.
  - The relationship name is displayed in the diamond-shaped box.



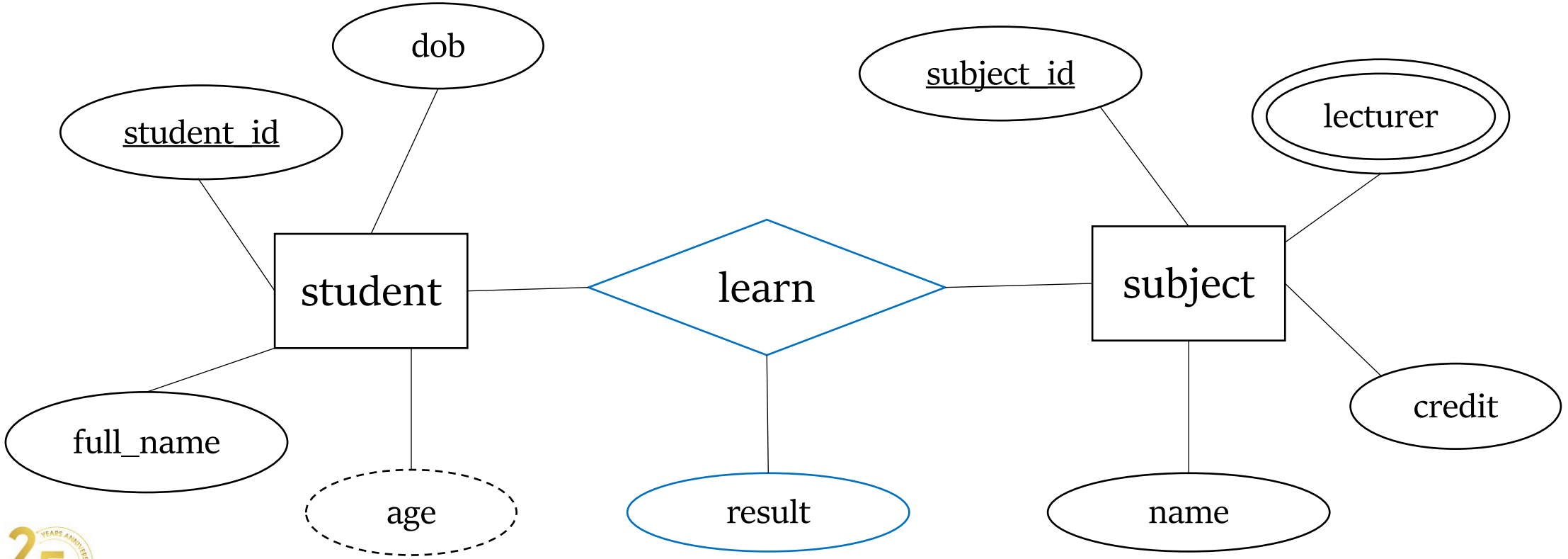
1.4. Relationships

- Identifying relationships for **weak entities** are relationship(s) between a weak entity set and its corresponding strong entity set(s)
  - Represented with **double diamond** in the ER Diagram



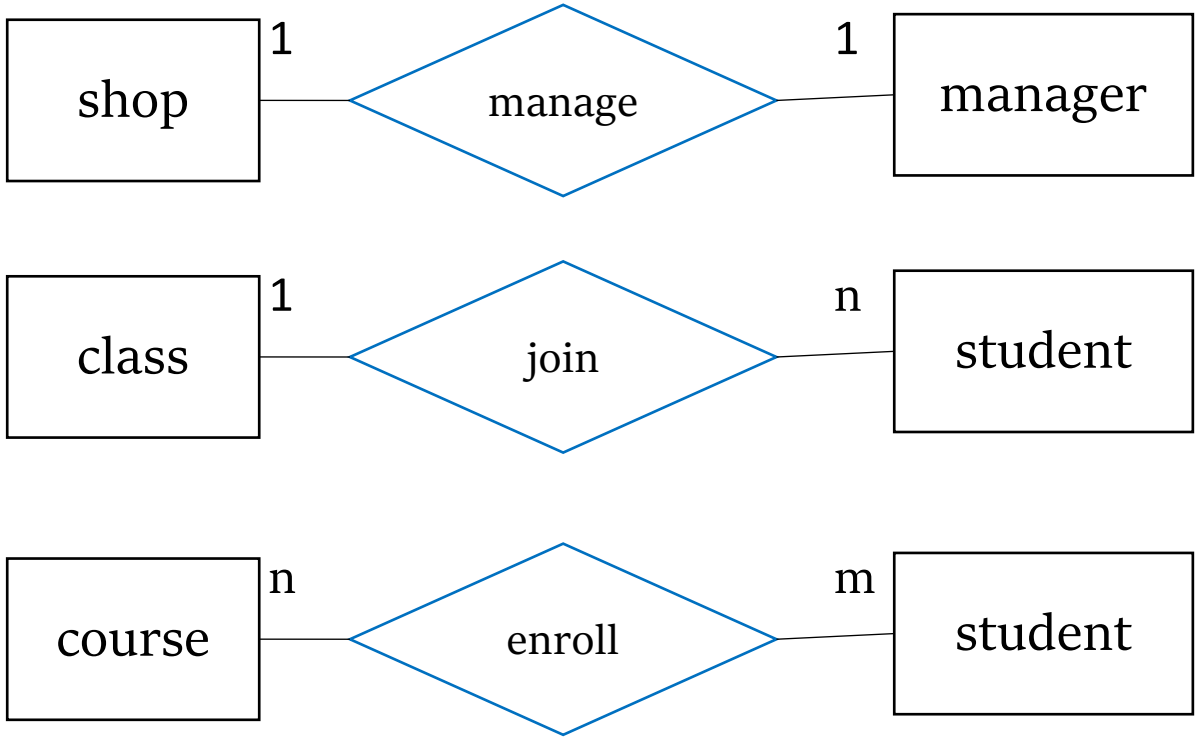
1.4. Relationships

- Relationships can have attributes of their own



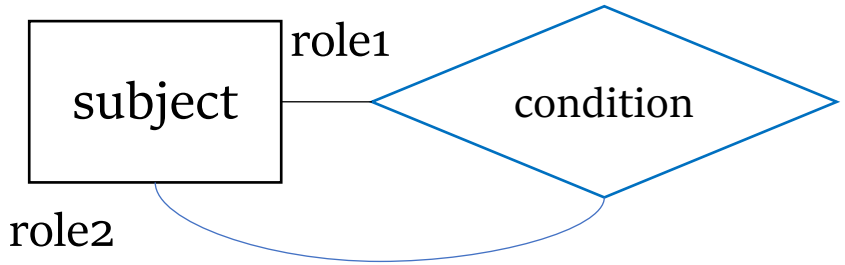
1.4. Relationships

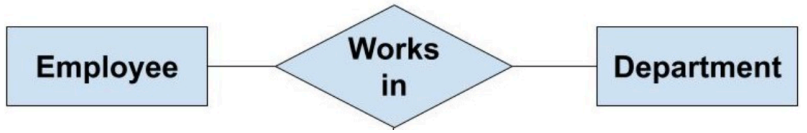
- Type of relationships
  - 1 – 1
  - 1 – n
  - n – m

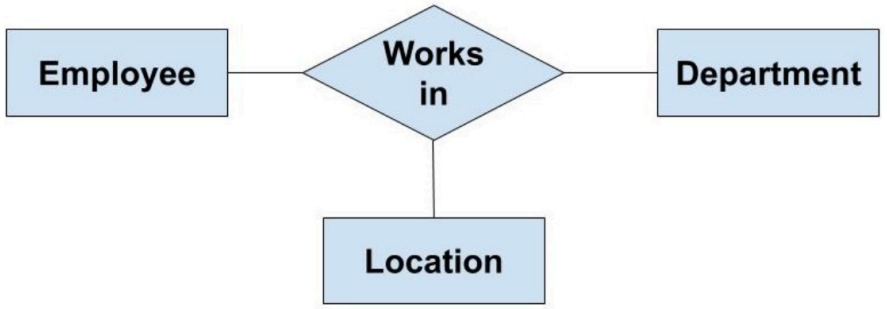


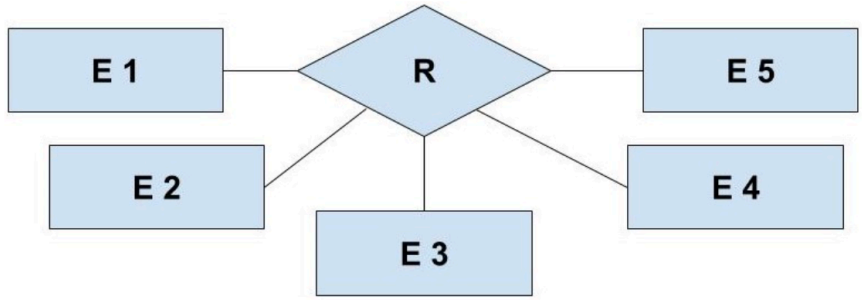
1.4. Relationships

- Degree of relationships
  - Unary (recursive)
  - Binary (the most usual)
    - Each employee works in a department
  - Ternary
    - An employee works in a department, an employee works at a particular location
  - … n-ary



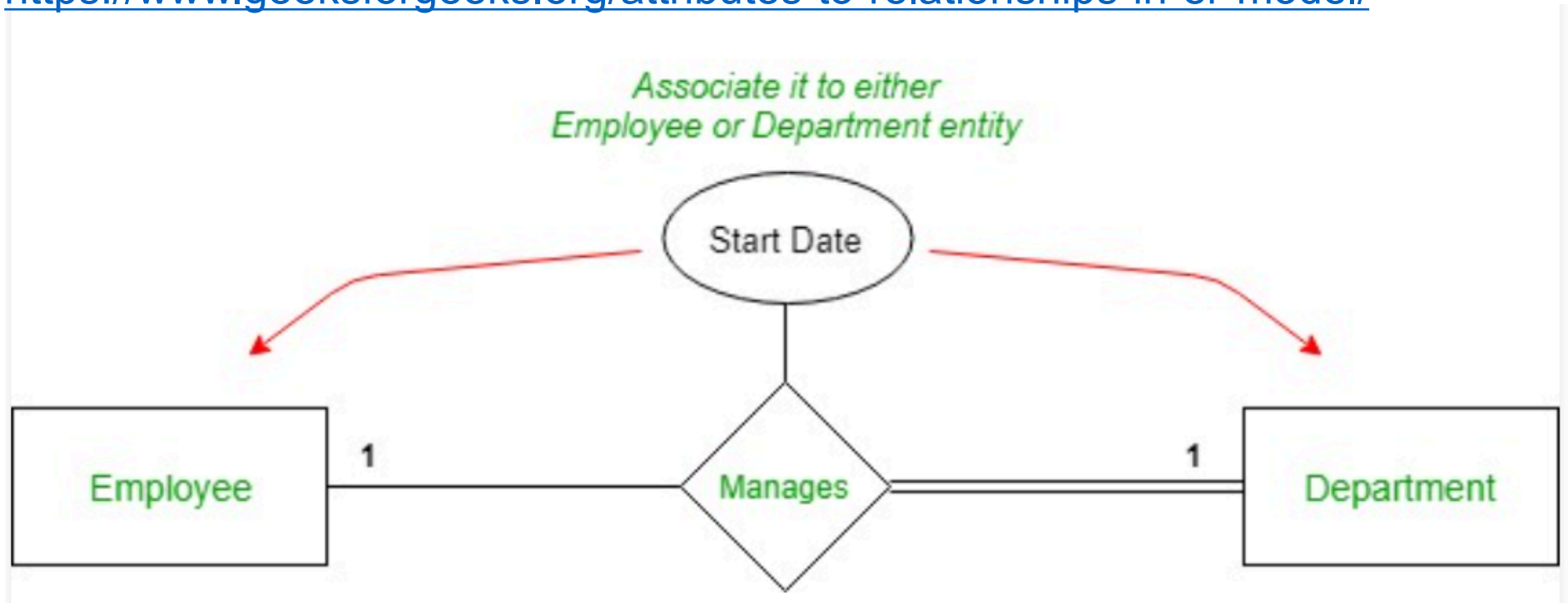






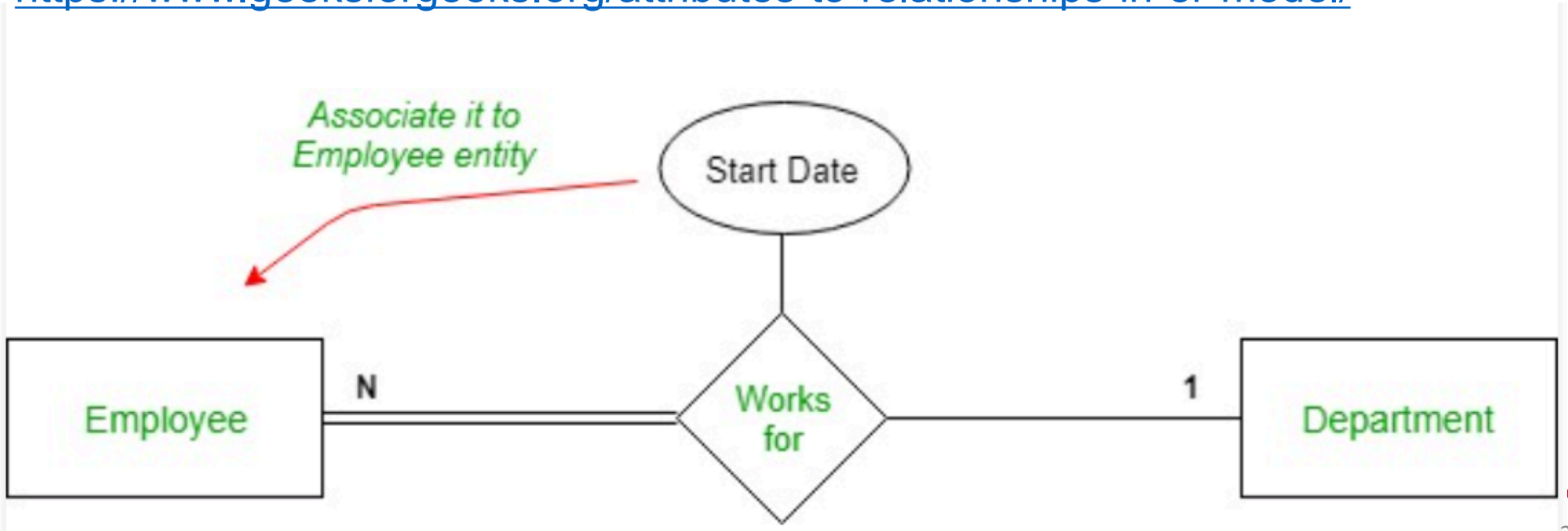
1.4. Relationships

- Just like entity sets, relationships can also have attributes…
  - … But, for 1 – 1 relationships, we prefer to associate the attribute to one of the entity sets in the relationship:
    - https://www.geeksforgeeks.org/attributes-to-relationships-in-er-model/



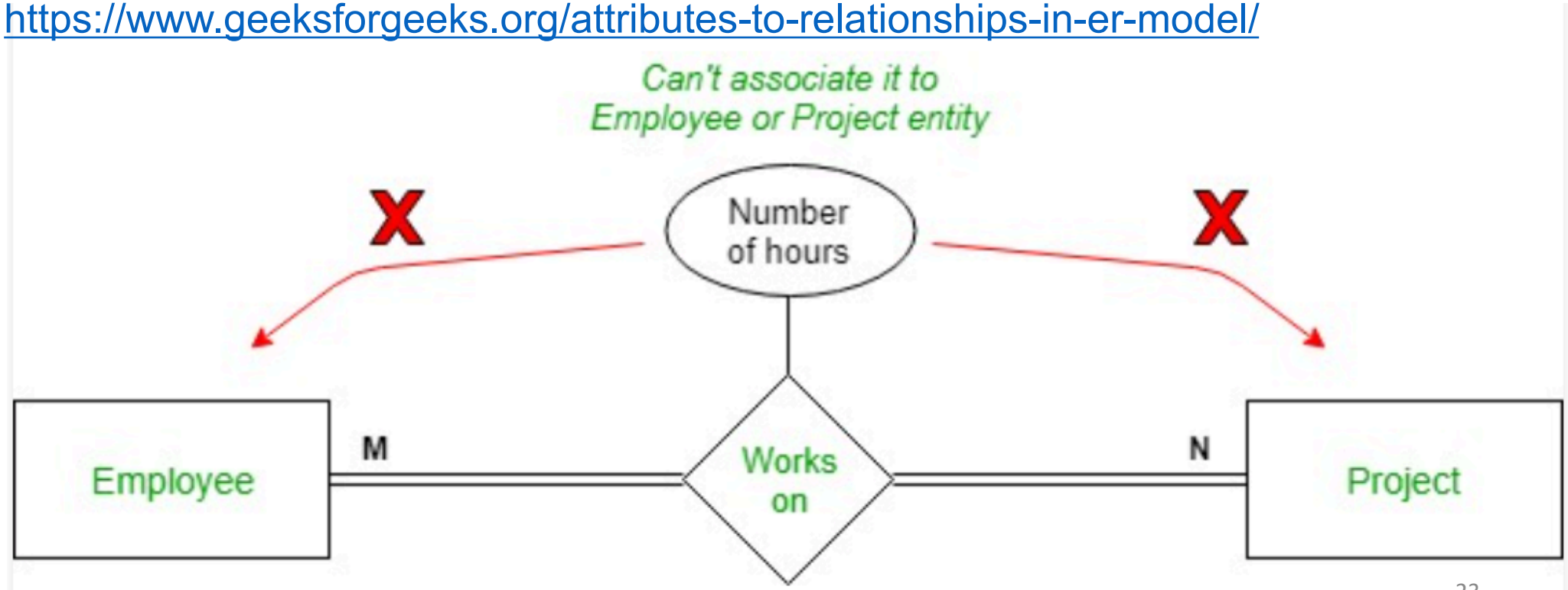
1.4. Relationships

- Just like entity sets, relationships can also have attributes…
  - … But, for 1 – n relationships, we prefer to associate the attribute to the entity sets on the “n side” of the relationship:
    - https://www.geeksforgeeks.org/attributes-to-relationships-in-er-model/



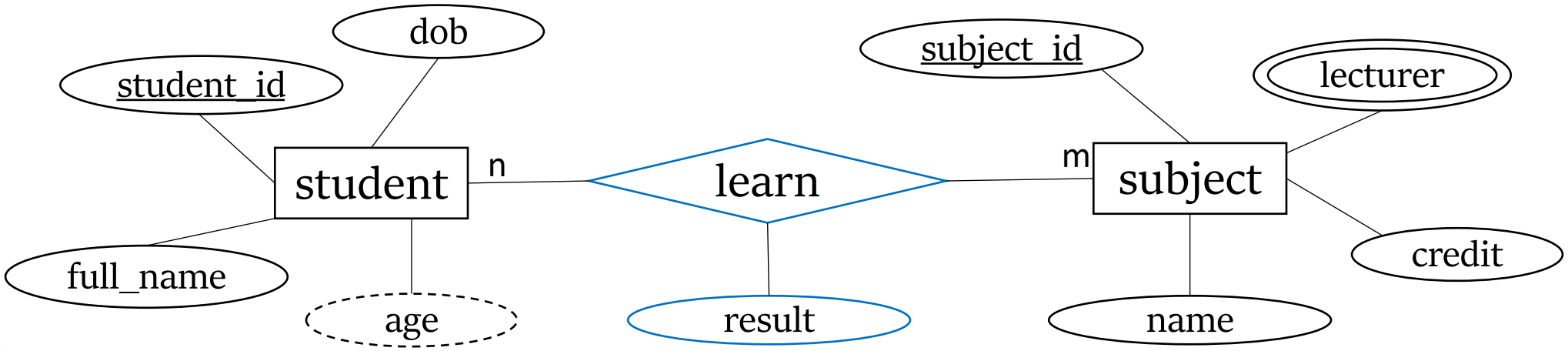
1.4. Relationships

- Just like entity sets, relationships can also have attributes…
  - … mostly for n – m relationships finally!
    - https://www.geeksforgeeks.org/attributes-to-relationships-in-er-model/



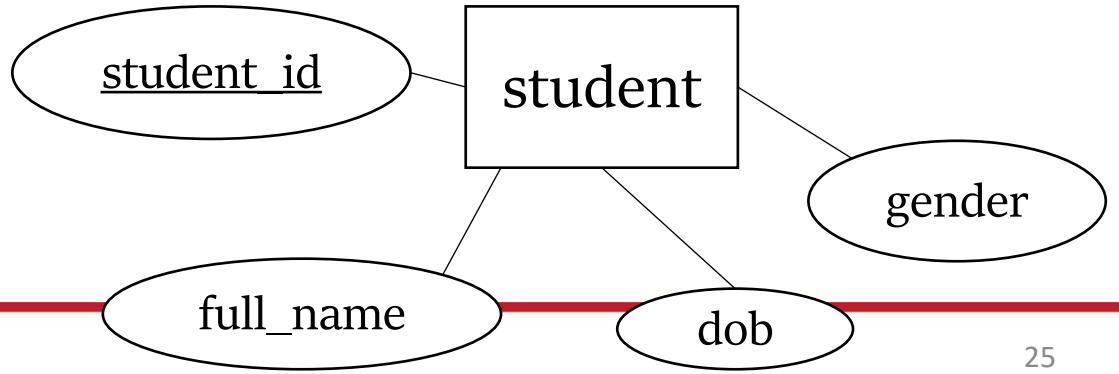
1.4. Relationships

- Just like entity sets, relationships can also have attributes…
  - But mostly for n – m relationships finally!
    - Another example:



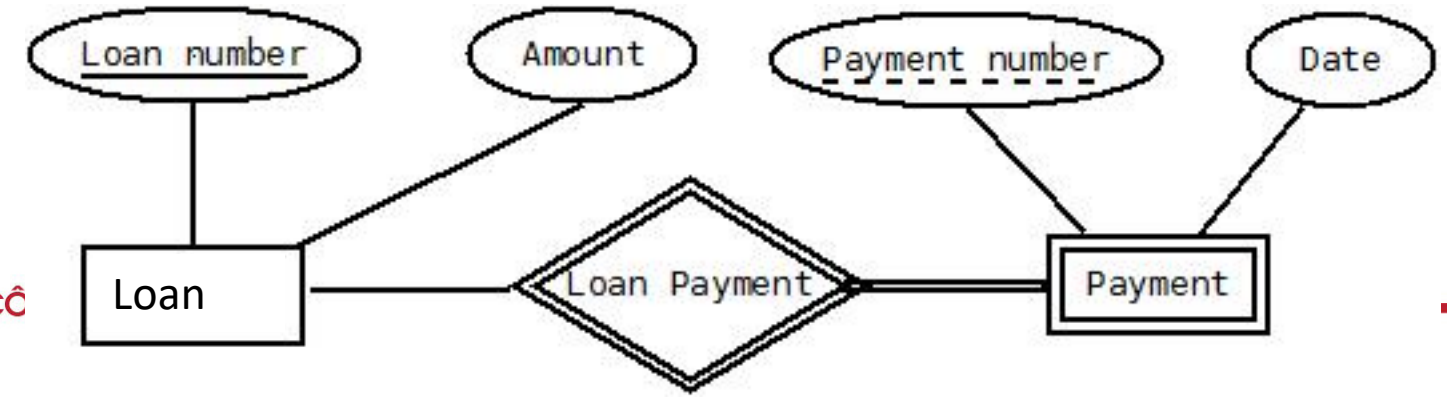
1.5. Key

- A key attribute is composed of one or more attributes which values are distinct for each individual entity in the entity set *(can be super-key, candidate key, primary key, alternate key…)*
- Each entity can have some keys. We choose one of them to be primary key.
- In ER diagrammatic notation, each key attribute has its name underlined inside the oval.



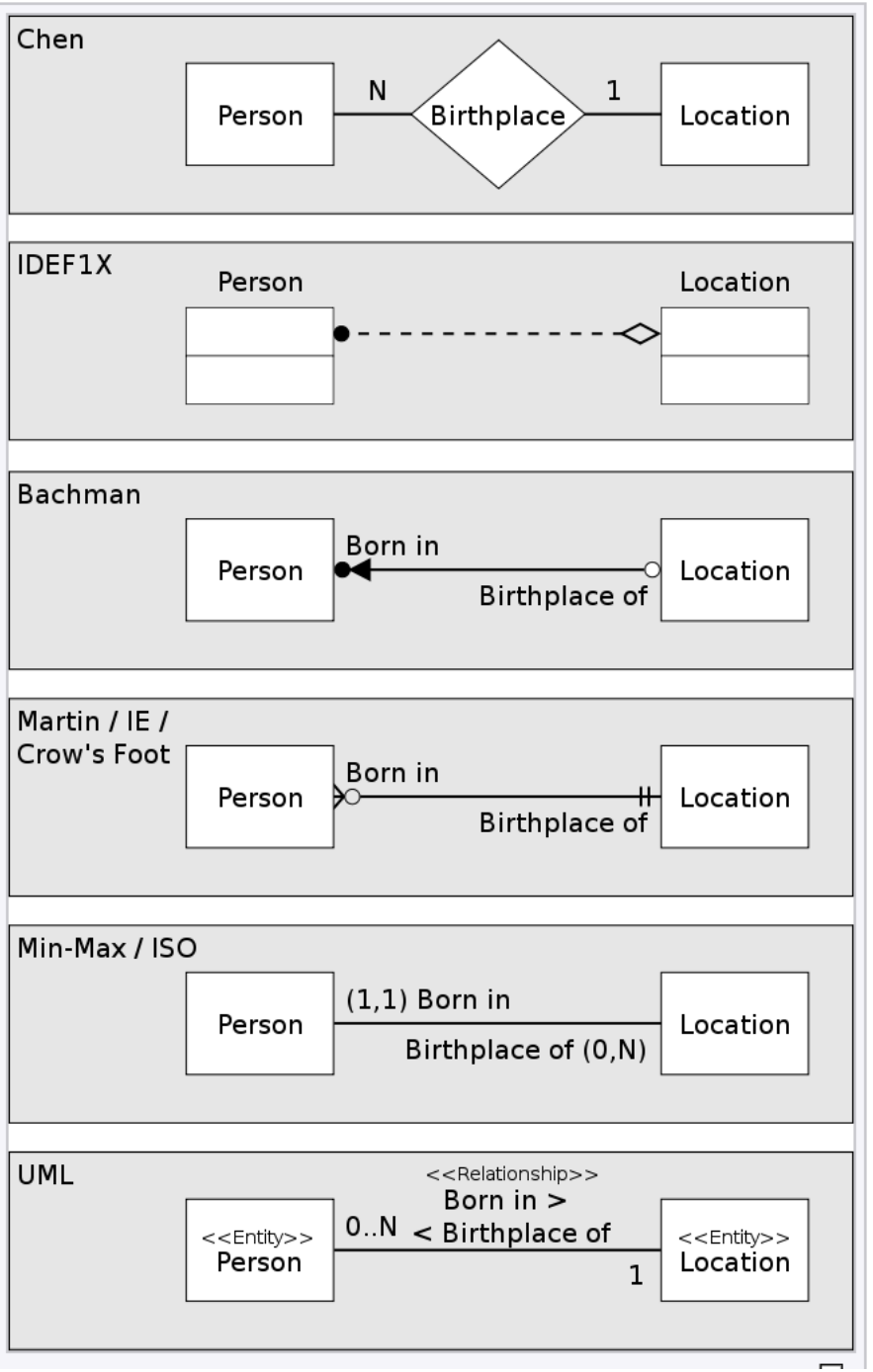
1.5. Key

- In the below ER Diagram:
  - 'Payment' is a weak entity set and loan is the corresponding strong entity set
  - 'Loan' is the corresponding strong entity set
  - 'Loan Payment' is the identifying relationship
  - 'Payment Number' is the partial key of the weak entity set, underlined with dotted lines in the ER diagram.
  - The PK of the Loan, along with the partial key, will be used to identify the payment



1.6. Different possible notations

- Many notations exist (most based on Chen' 76):
- Goals:
  - Graphical description of a database schema
  - Independent from actual database realization (network, RDBMS…)
  - Concise and readable description of a database
- One must be able to read/understand all the possible notations

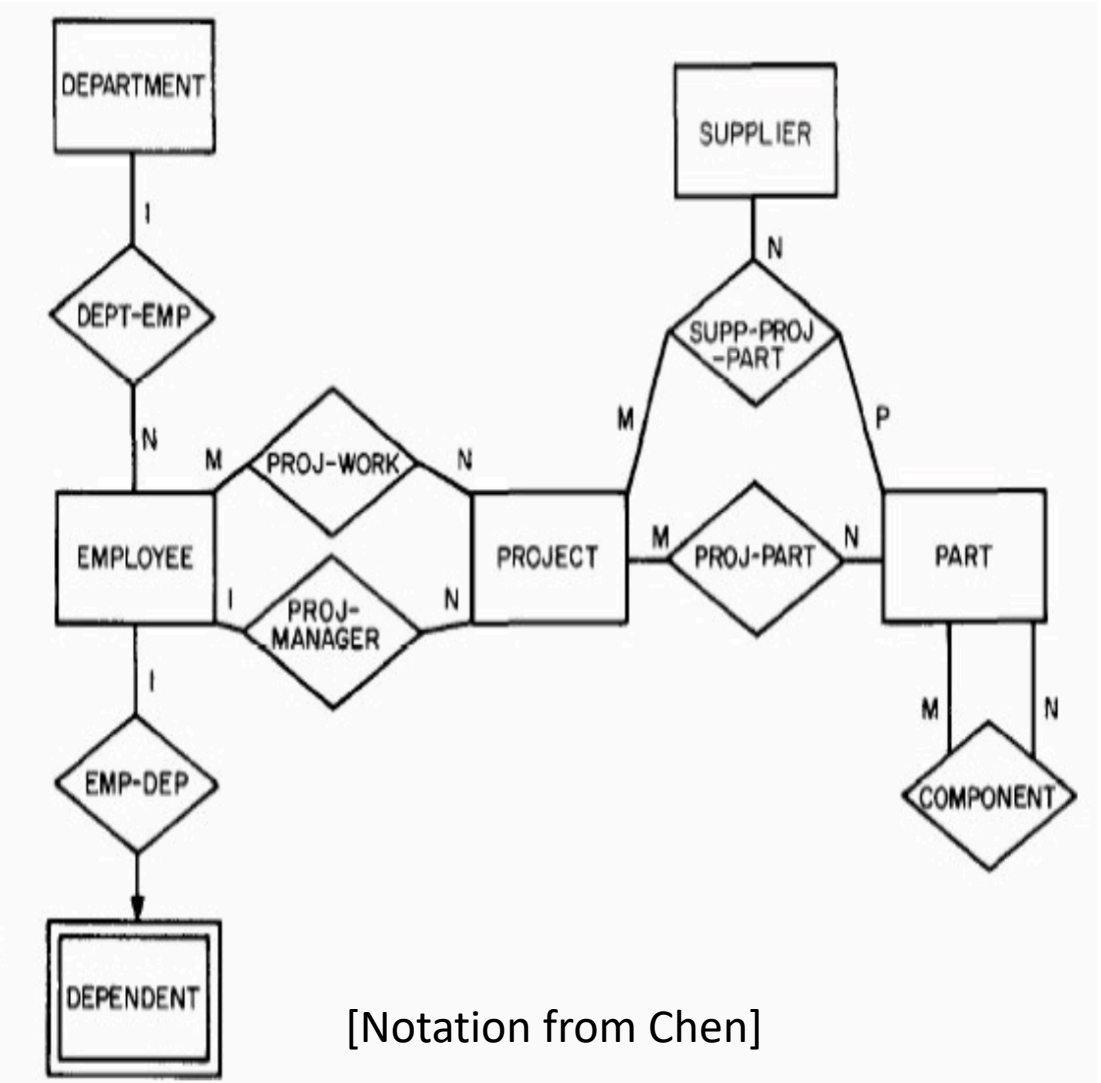


[Wikipedia]

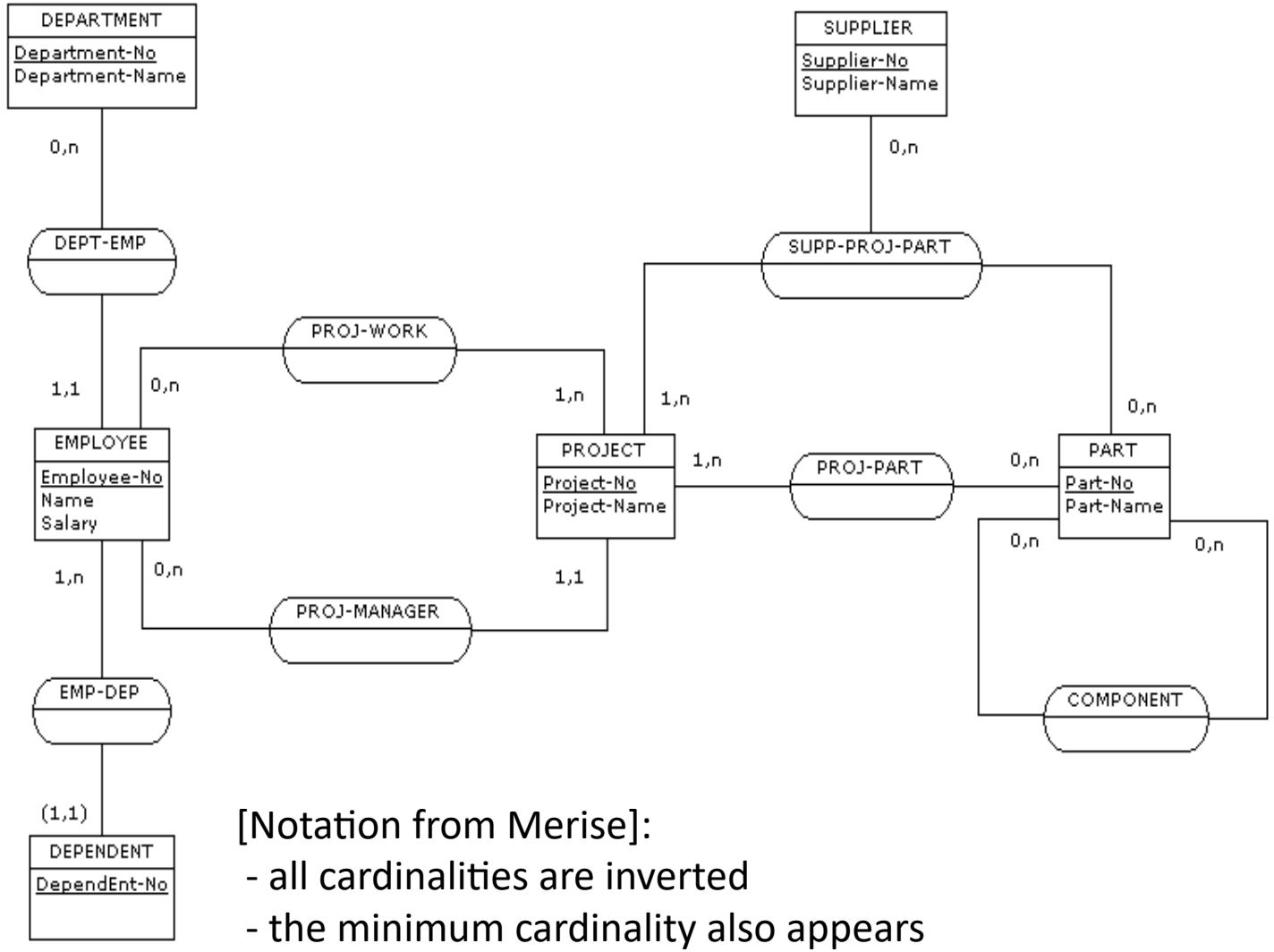
1.6. Different possible notations

- Mastering all possible notations is not so easy…
- Example of MERISE:
  - Widely used in France (and in Europe) because it has been introduced approximately around the same time as Chen's model (1976)
  - Cardinalities are **inverted** compared to Chen notations
  - In Merise, we also write the **minimum** cardinality (like in UML, but inverted)

1.6. Different possible notations



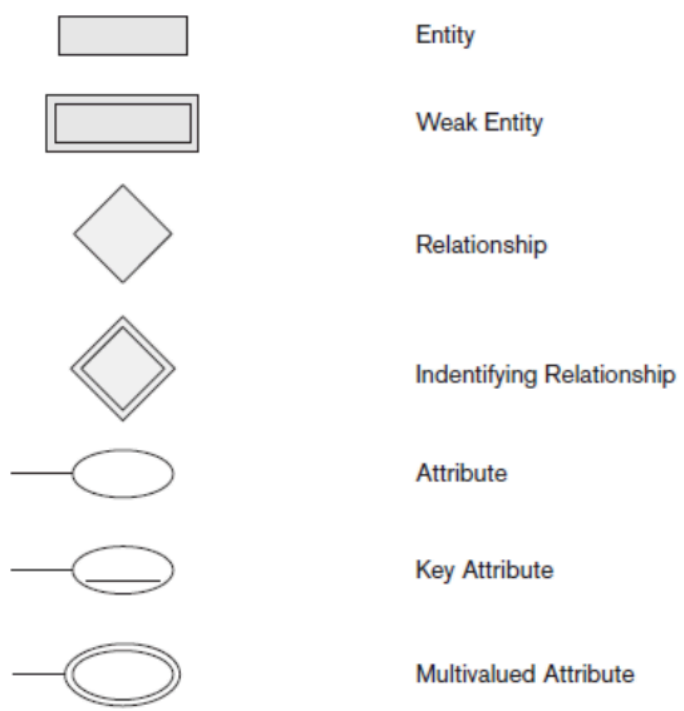
[Notation from Chen]

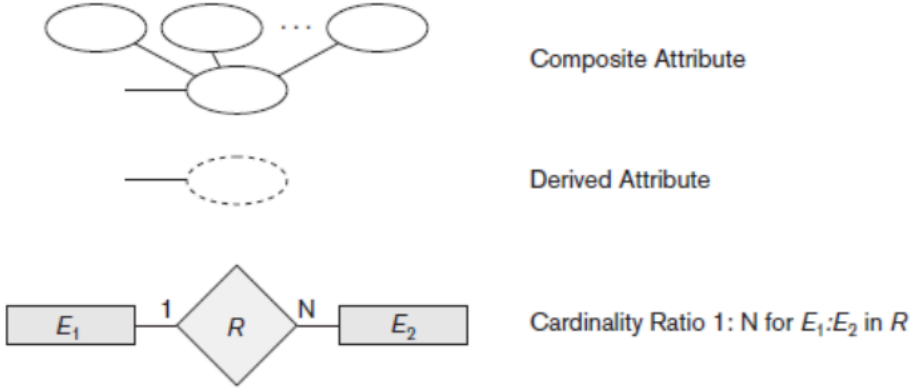


[Notation from Merise]:
- all cardinalities are inverted
- the minimum cardinality also appears

1.6. Different possible notations

- In this course, we'll mainly use Chen's notations defined as follows *(and sometimes, the UML notation)*





2. How to create an ERD

- ERD process
- Example

2.1. ERD process

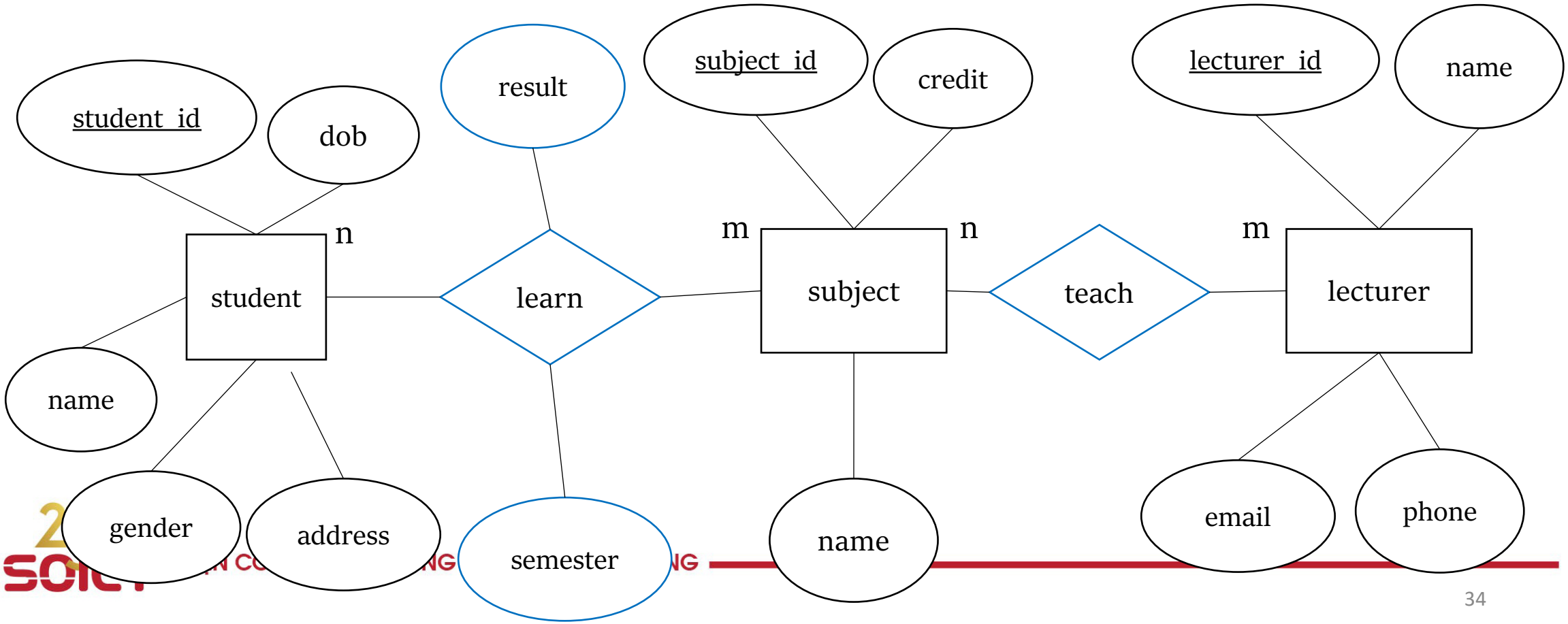
- Step 1: Identify all entity sets
  - Notice concepts, nouns
- Step 2: Identify all relationships among entity sets
  - Notice verbs
  - Type and degree of relationships

2.2. An example

- Read carefully the following scenario:
  - The information about students includes student identification (uniquely identifying each student), name, gender, date of birth and address.
  - During the education time at school, students must study a lot of subjects. A subject can be learnt by students. A subject should contain information such as subject identification, name and credit.
  - A lecturer can teach some subjects, and a subject can be taught by a group of lecturers. The information about lecturers should include lecturer identification, name, phone, email.
  - Students can learn subjects during several semesters, and their results (grades) should be stored.

2.2. An example

- We can draw this ER diagram

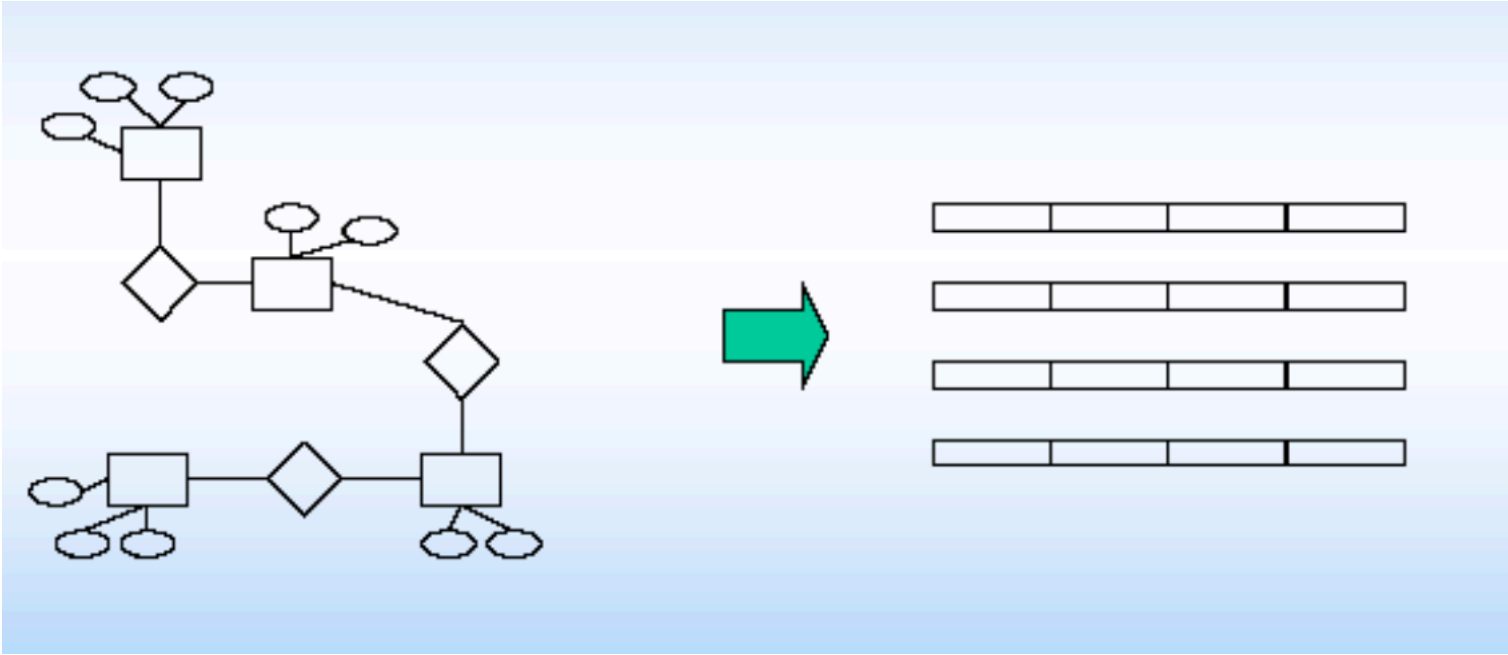


2.3. Some tips to draw good ER diagrams

- Make sure that each entity set only appears once per diagram
- Name every entity set, relationship, and attribute on your diagram
- Examine relationships between entity sets closely.
  - Are they necessary?
  - Are there any relationships missing?
  - Eliminate any redundant relationships.
- Don't connect relationships to each other.
- Use colors to highlight important portions of your diagram
- N.B. Usually, foreign key attributes do not appear in the ER diagram (they are implied by the cardinalities of the relationships)

3. Mapping from ERD to relational schema

- Mapping process
- Example



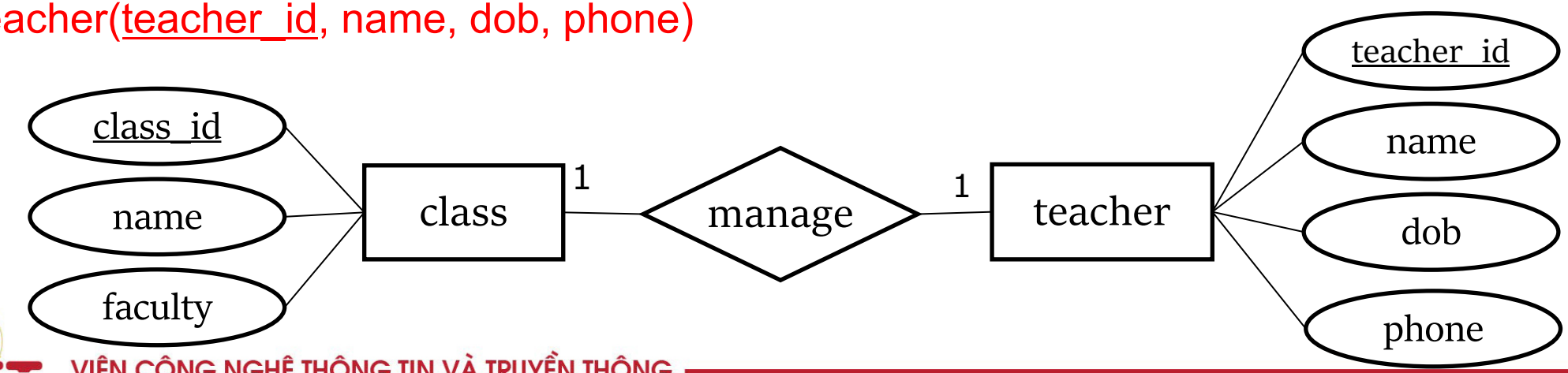
3.1. Mapping process

1- Mapping of strong entity sets

- For each strong entity set, create a relation that includes all the **simple** attributes of that entity set.
- The PK of the strong entity set becomes PK of the relation

class(class_id, name, faculty)

teacher(teacher_id, name, dob, phone)

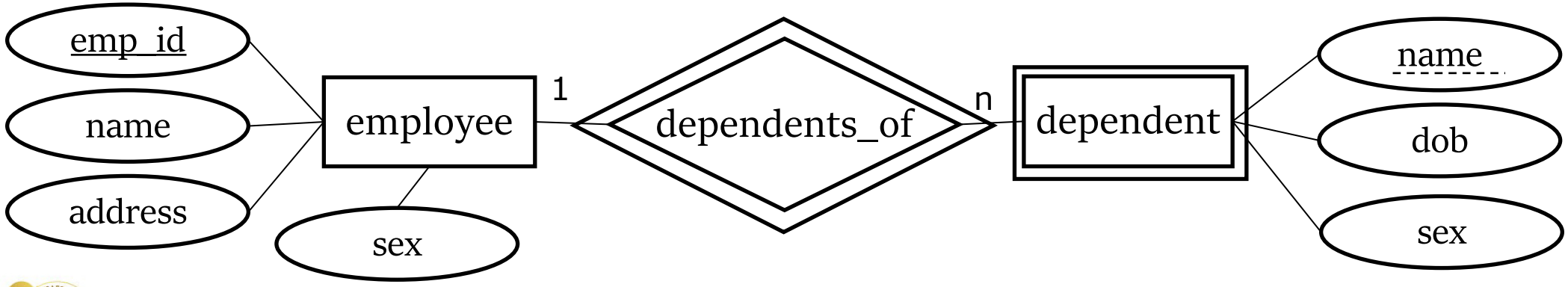


3.1. Mapping process

2- Mapping of weak entity sets

- For each weak entity set, create a relation that includes all the simple attributes of that entity set.
- The PK of the related strong entity set(s) should be included in PK of the relation corresponding to the weak entity (here dependent = child)

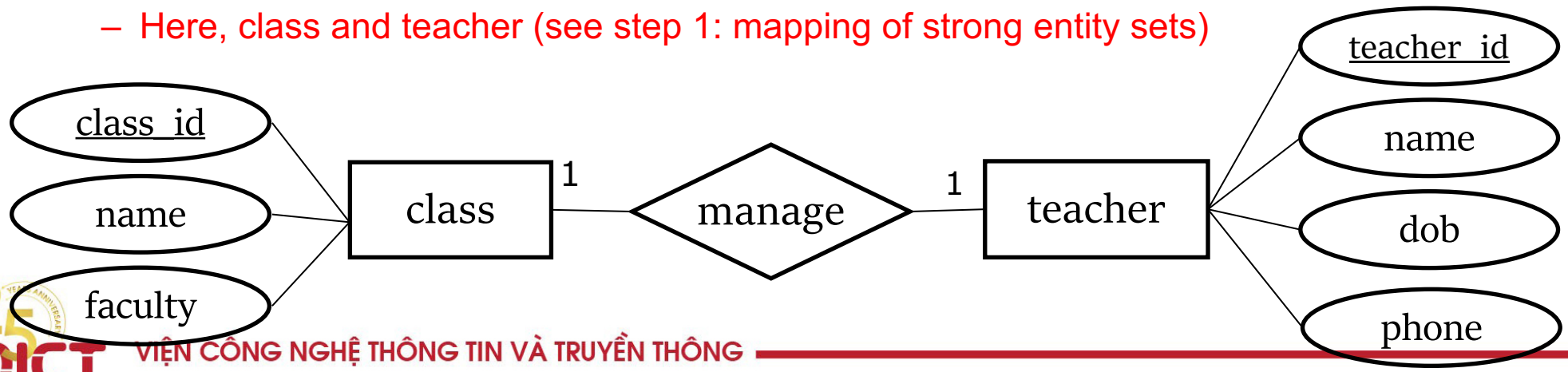
dependent(emp_id#, name, dob, sex)



3.1. Mapping process

3- Mapping of 1 - 1 relationships

- 1-1 relationships **should be rare** in any relational database design.
  - In many cases, it indicates that two entities actually belong in the same table…
- To map 1-1 relationships into a relational schema, we:
  - Create a new relation which has all prime-attributes of both entity sets
    - manage(class_id#, teacher_id#)
  - Use foreign keys to refer to the relations created from the entities
    - Here, class and teacher (see step 1: mapping of strong entity sets)



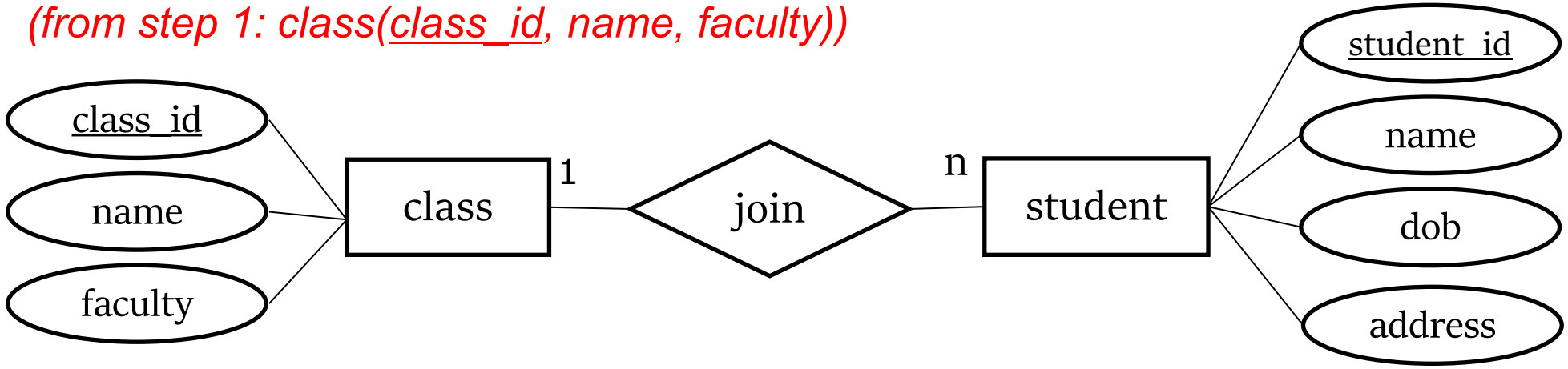
3.1. Mapping process

4- Mapping of 1 - n relationships

– Add 1 attribute/foreign key in the relation of the entity set on the “n side” of the relationship

student(student_id, name, dob, address, **class_id#**)

(from step 1: class(class_id, name, faculty))

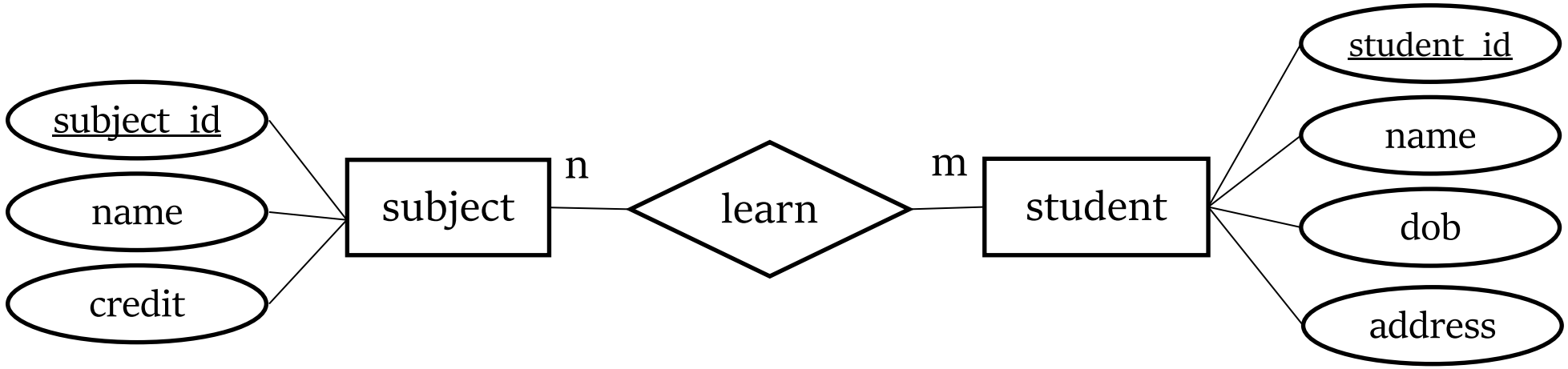


3.1. Mapping process

5 - Mapping of n - m relationships

– Create a new relation which has all PK attributes of both entity sets…

learn(subject_id#, student_id#)



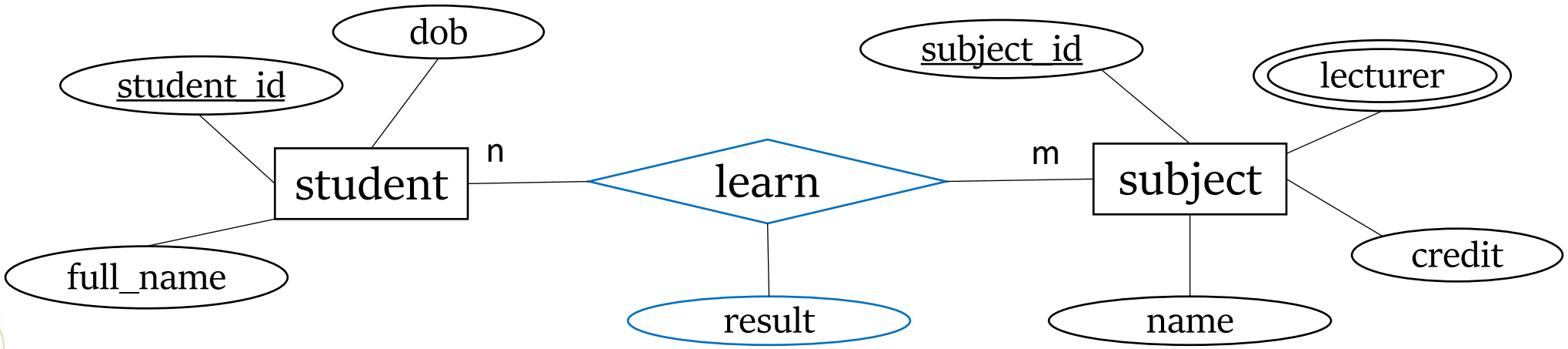
3.1. Mapping process

5 - Mapping of n - m relationships

– Create a new relation which has all PK attributes of both entity sets…

… plus, possibly, its own relationship attributes!

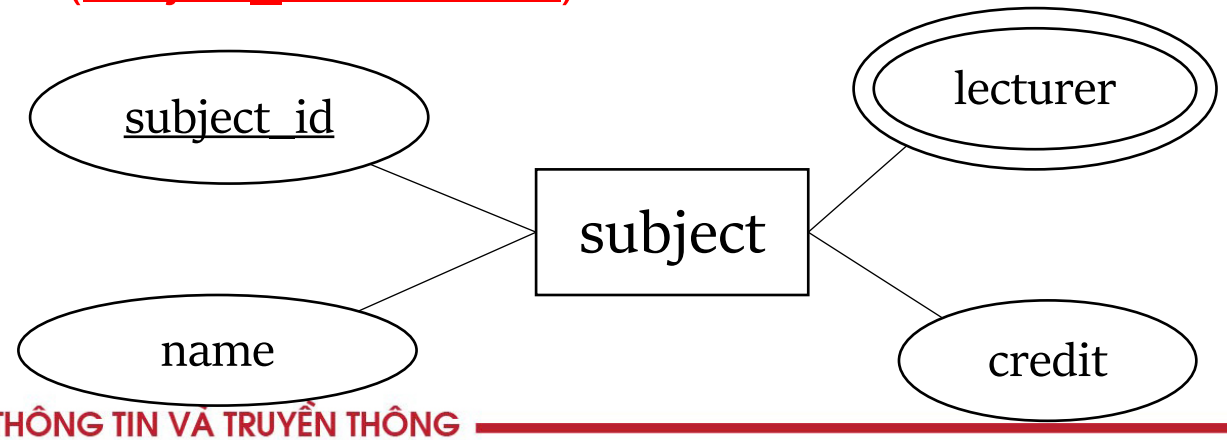
learn(subject_id#, student_id#, result)



3.1. Mapping process

6- Mapping of multivalued attributes

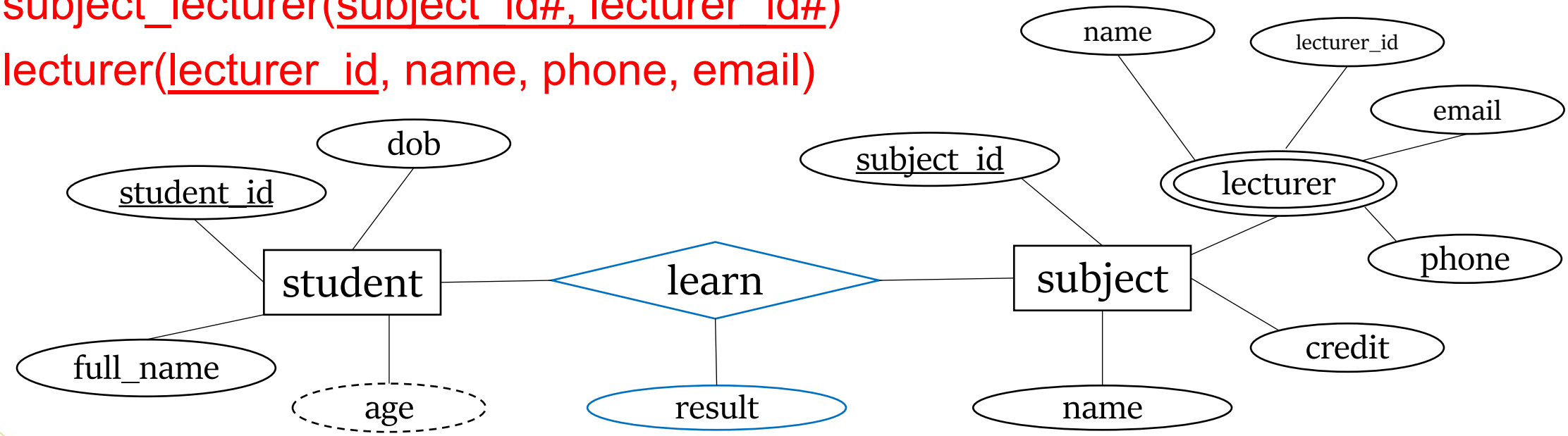
- For each multivalued attribute A, create a new relation R including an attribute corresponding to A, plus the primary key attribute K (as a foreign key in R) of the corresponding entity set
- The primary key of R is the combination of A and K.
  - subject_lecturer(subject_id#, lecturer)



3.1. Mapping process

7- Mapping of composite attributes

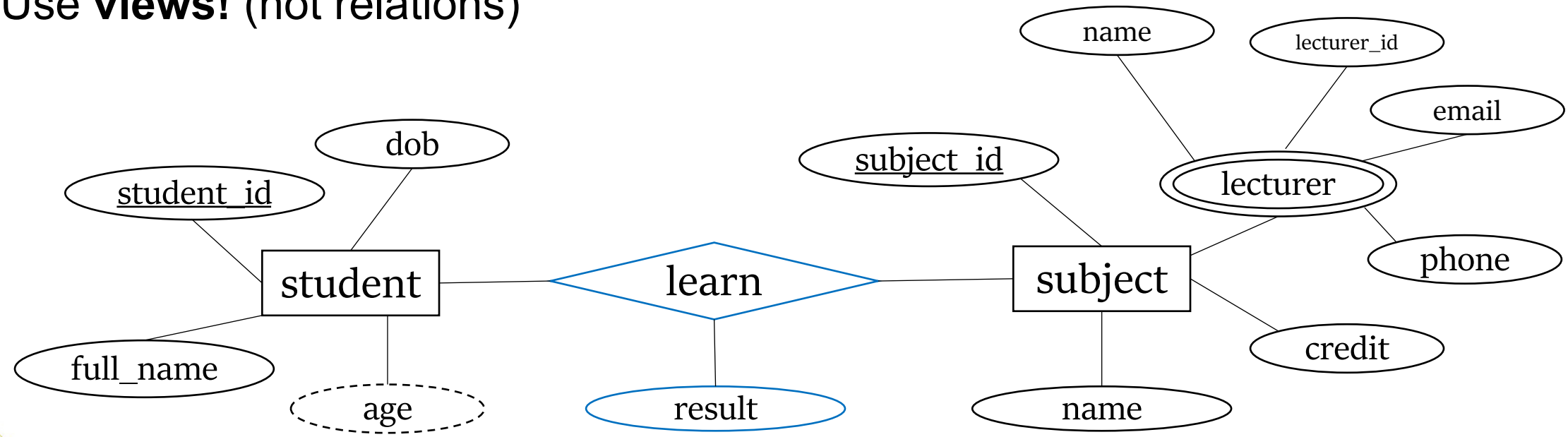
- Add a separate relation, and refer to it in the main relation
- subject_lecturer(subject_id#, lecturer_id#)
- lecturer(lecturer_id, name, phone, email)



3.1. Mapping process

8- Mapping of derived attributes

- Here, the attribute age is derived
- Use **views!** (not relations)



3.2. Example

- student(student_id, full_name, dob)
- subject(subject_id, name, credit)
- lecturer(lecturer_id, name, phone, email)
- learn(student_id#, subject_id#, result)
- teach(lecturer_id#, subject_id#)

3.3. Exercises

- **Exercise 1:** The HR of a company wants to manage the employees.
- We want to be able to know the name, function, date of entry, salary, commission (percentage on a sale) of each employee, and the identifier of the department in which each employee works (each employee works in a single department)
- The HR also wants to know the name of the department in which each employee works.
- As the company is spread out in several cities, the departments are characterized by their name and by their city (one department is located in only one city)

1. Give the ER diagram corresponding to this data
2. Map it into a relational schema

3.3. Exercises

- **Exercise 2:** same problem as before, but in another company where:
  - Each employee might belong to multiple departments
    - We need to store the percentage of employment of each employee in each department
  - One department might be located in multiple cities. Therefore, an employee who belongs to that department might also have to work in multiple cities, but each employee is officially based in only one city.

1. Give the ER diagram corresponding to this data
2. Map it into a relational schema

Summary

- Introduction
  - ERD in DB designing, and its components
- How to create an ERD
  - discover entity sets, attributes and relationships among entity sets
- Mapping from ERD to relational schema
  - transform from ERD into a set of tables

ĐẠI HỌC BÁCH KHOA

25 YEARS ANNIVERSARY SOICT

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Thank you for your attention!

soict.hust.edu.vn/ fb.com/groups/soict