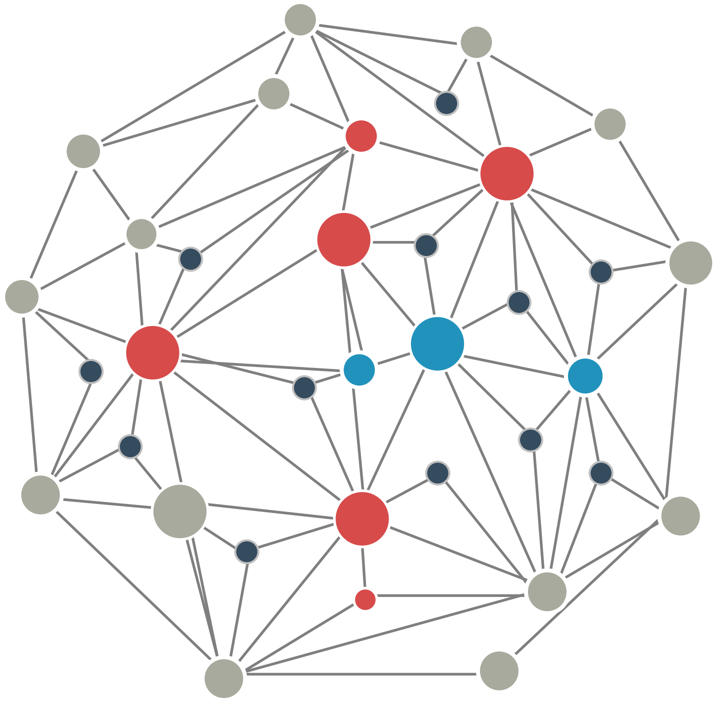
DST2 – Week 11

Database design and E-R model

Zhaoyuan Fang

zhaoyuanfang@intl.zju.edu.cn



2021/12/7

Week 11 Learning Objectives

- Understand why database design is important
- Demonstrate Entity-Relationship (E-R) modeling
  - What is Entity?
  - What is Relationship?
- Learn how to draw ERD and how to interpret ERD
- Classify various forms of attributes
- Distinguish unary, binary, ternary relationships

Database requirements

- **Database requirements** is the first and most critical step in the process of developing a database
  - They are **statements** that define the **details and constraints** of the data and metadata
  - They can be represented in a **conceptual database model**, such as **ER models**

- **Entity-Relationship (ER) models**
  - a widely used conceptual database modeling method
  - a **technique** that enables the **structuring and organizing** of the requirements
  - provides a way to graphically represent the requirements

Basic ER modeling constructs

- ER modeling involves three components
  - 1) **entities**: objects, e.g. persons, books
  - 2) **relationships**: associations among several entities, e.g. persons *reads* books
  - 3) **attributes**: features of an object, e.g. names of persons. An attribute takes its values from a domain, e.g. name domain = {“Smith”, “Joe”, “Tylor”, …}

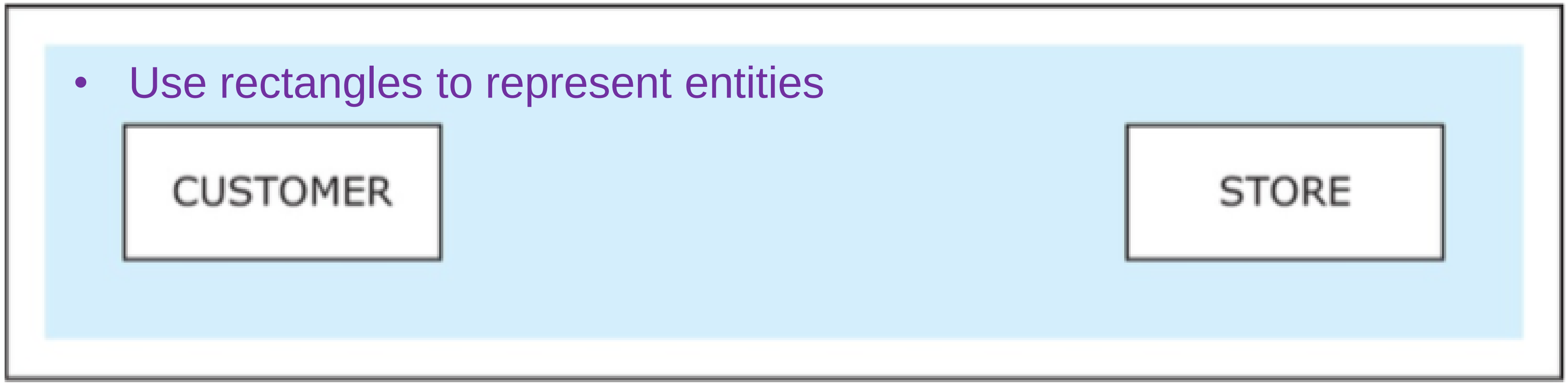
- The result of ER modeling is an **ER diagram (ERD)**
  - A graphical representation of the overall logical structure of a database

- Notations of ERD
  - many ER notations are in use, no universally adopted one
  - We will use a modified version of **Peter Chen’s notation** (more popular)
  - E-R can be represented differently, but they always **have the same meaning**

Entities

- **Entities** represent various **real-world notion**, such as people, places, objects, events, items etc.
  - e.g. an ERD of a retail company may contain entities such as CUSTOMER, STORE, ...

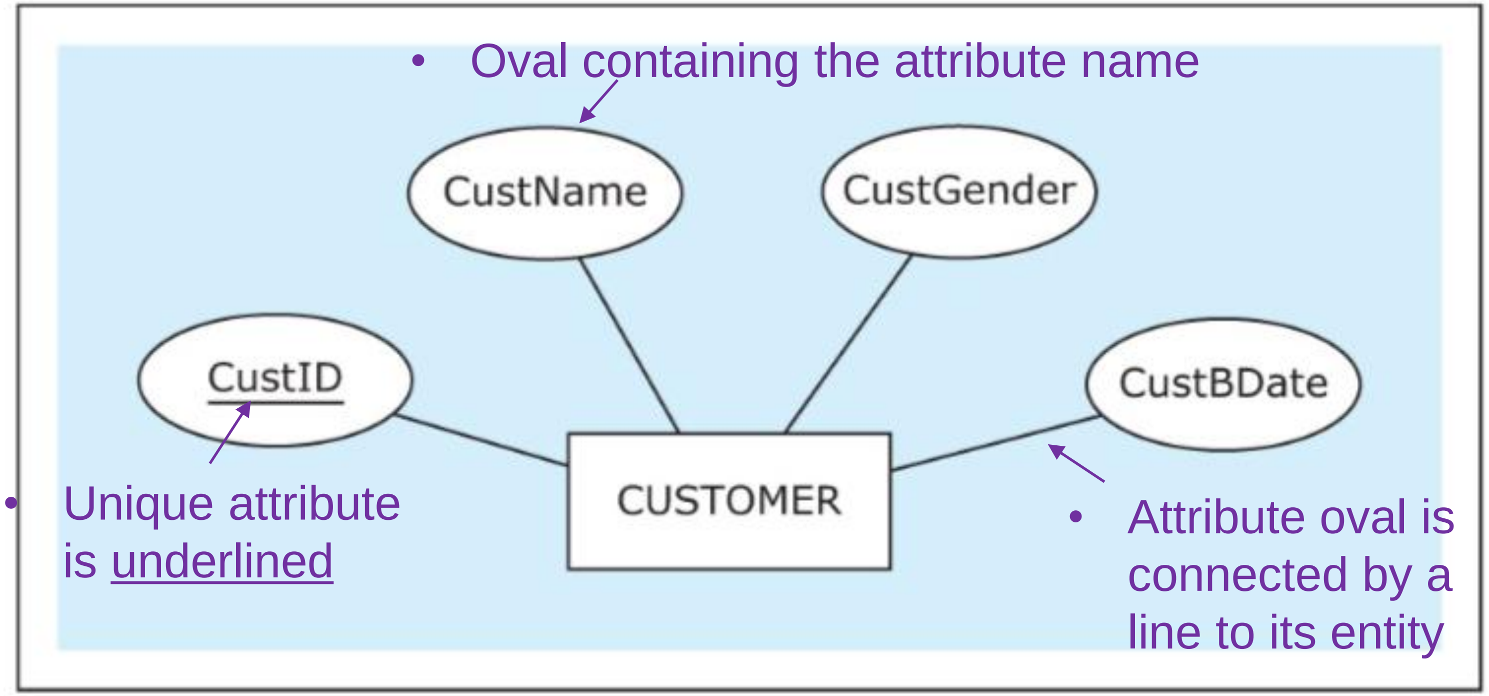
- Each entity is a **set**, containing a number of **entity instances (entity members)**
  - e.g. entity CUSTOMER = {Joe, Sue, Pat, …}



- Use rectangles to represent entities

Attributes

- Each entity has some **attribute**(s).
- An attribute tells us something about the entity: some characteristic, some information, some property, some feature, …
- Attributes represent the **details** that will be recorded for **each entity instance**
- e.g. entity CUSTOMER has attributes: CustID, CustName, CustBDate, CustGender.

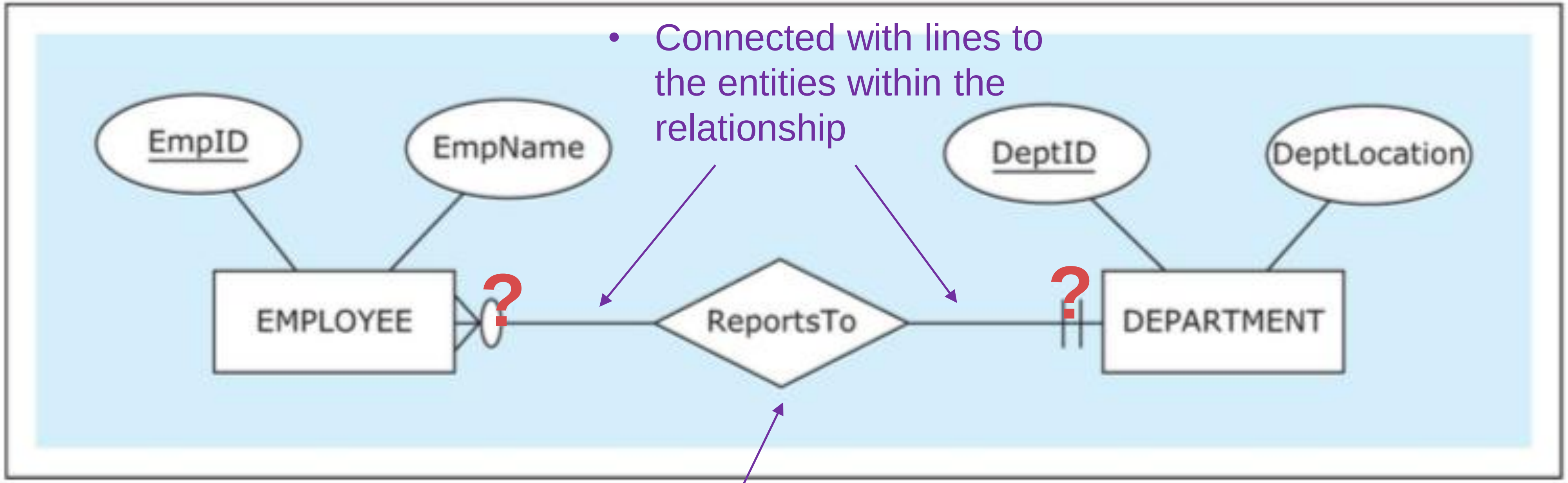


- Oval containing the attribute name
- Unique attribute is underlined
- Attribute oval is connected by a line to its entity

- **Unique attribute:** usually necessary for a regular entity (**at least one** unique attribute)

Relationships

- Entities are connected with each other. No isolated entities.
- Such relation is called a **relationship**.



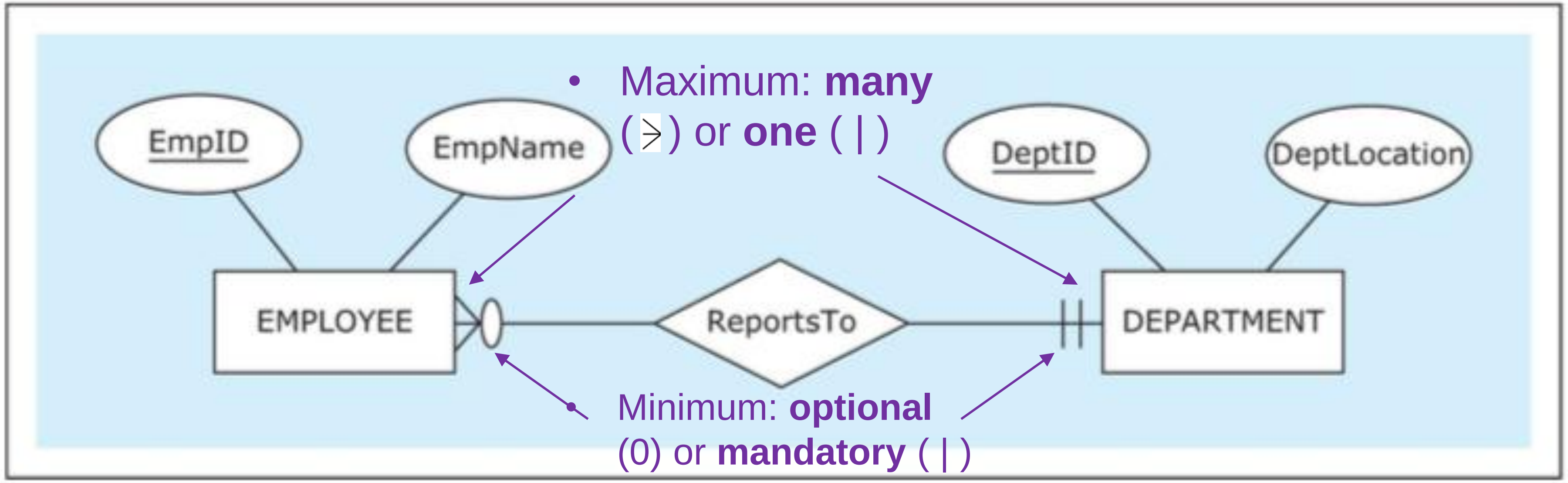
- Connected with lines to the entities within the relationship

- Shown as a diamond
- With a word or phrase naming the relationship inside

Cardinality Constraints

- **Cardinality constraints -** depicts how many instances of one entity can be associated with instances of another entity
  - **Maximum cardinality**
    - One (straight bar: I) or Many (crow's foot symbol: ⋗)
  - **Minimum cardinality (participation)**
    - Optional (circular symbol: 0) or Mandatory (straight bar: I)

Each employee reports to exactly one department. A department may have many employees reporting to it, but it does not have to have any.



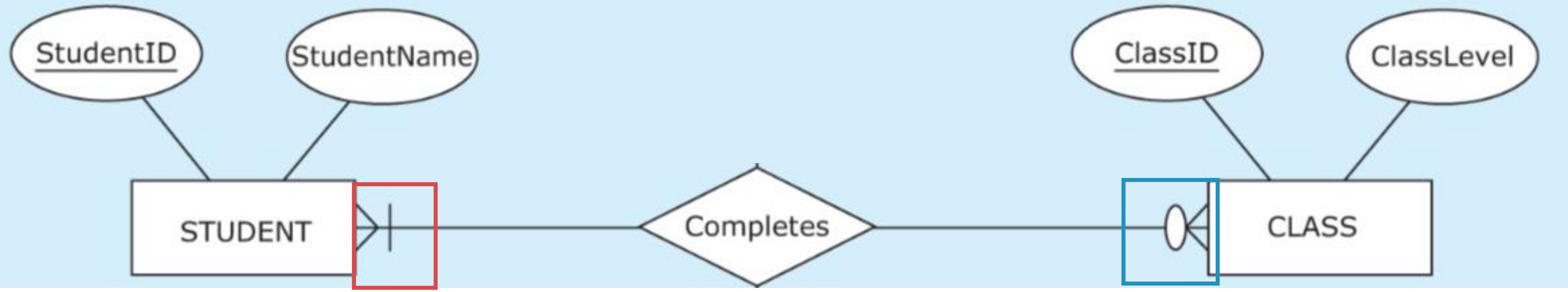
- Maximum: **many** (⋗) or **one** (|)

Minimum: **optional** (0) or **mandatory** (|)

Cardinality Constraints - task

Task 1 :

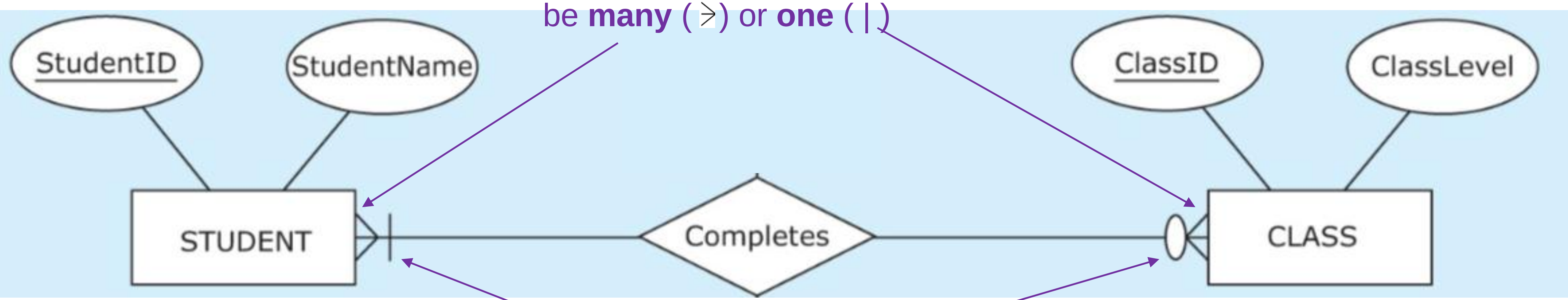
1. How many entities in this ERD?
2. What is the relationship?
3. What are the attributes?
4. What are the unique attributes?
5. What does the red box indicate? What is the meaning?
6. What does the blue box indicate? What is the meaning?



Cardinality Constraints - task answer

- *This database will keep track of students and class*
- *For each student, we will keep track of his or her unique student ID and name*
- *For each class, we will keep track of the unique class ID and class level*
- *Each students may have completed many class, but he/she does not have to complete any. Each class must have at least one student completed it and can have many student.*

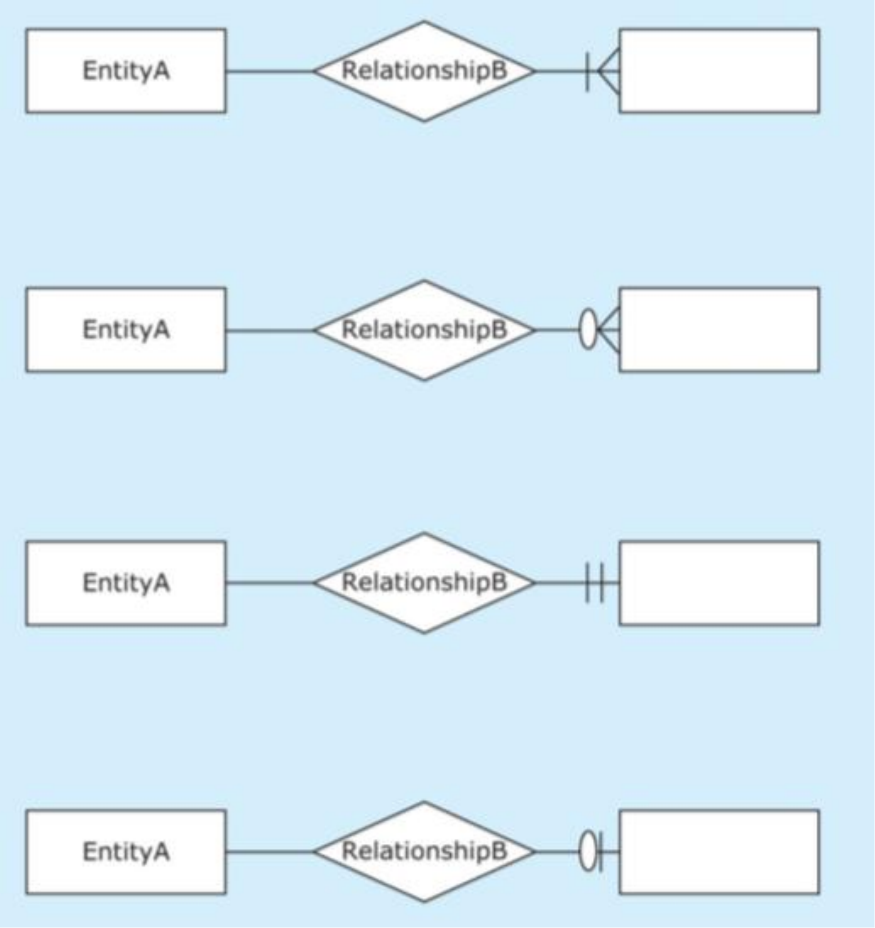
- Maximum cardinality can either be **many** (>) or **one** (|)



- Minimum can either be **mandatory** (|) or **optional** (0)

Cardinality Constraints - task

Task 2 : Connect the left diagrams to the right terms



Mandatory one

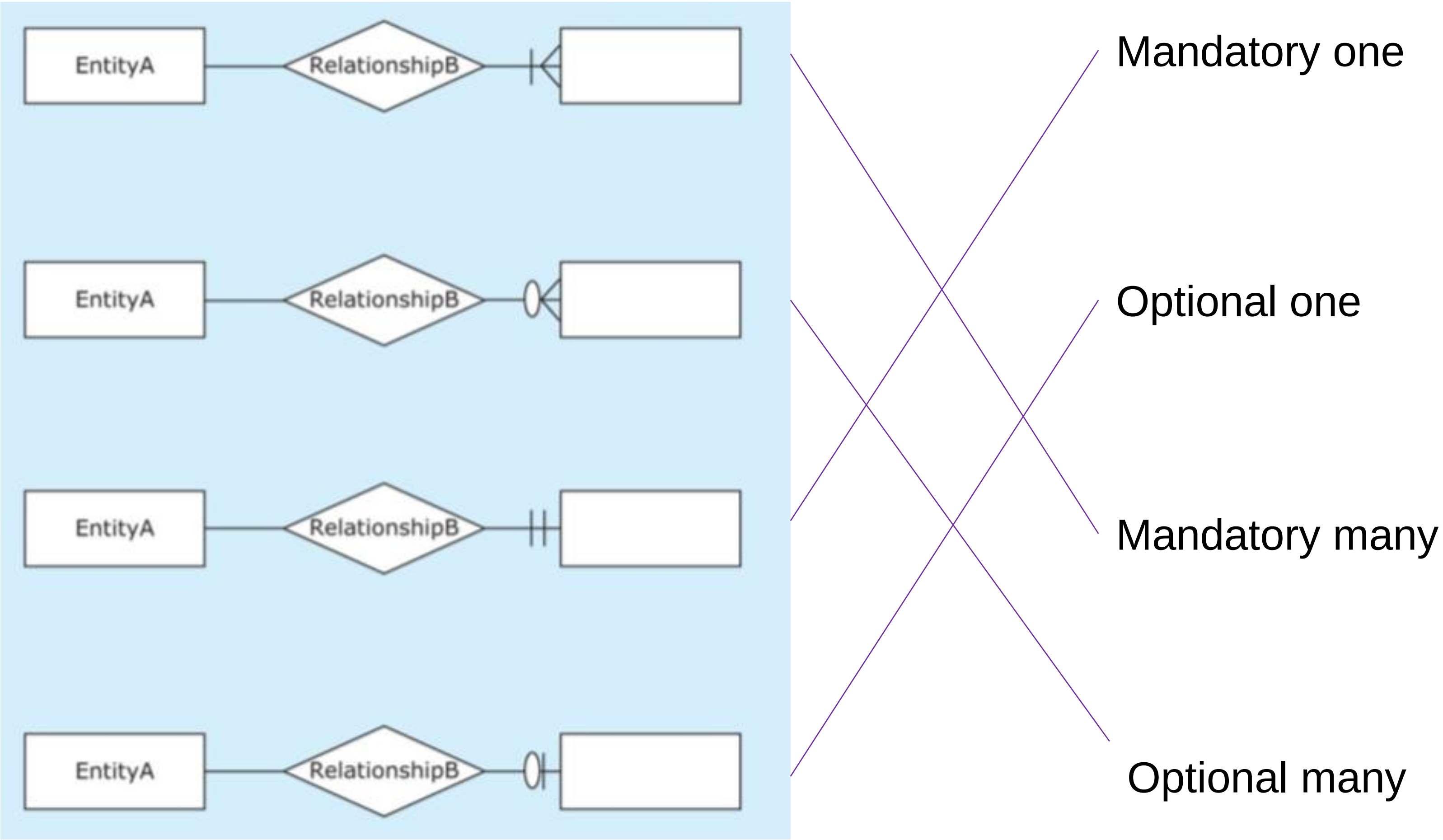
Optional one

Mandatory many

Optional many

Cardinality Constraints - task answer

Task 2 : Connect them



Mandatory one

Optional one

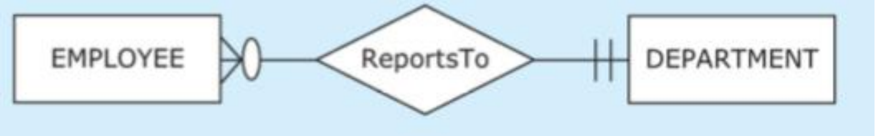
Mandatory many

Optional many

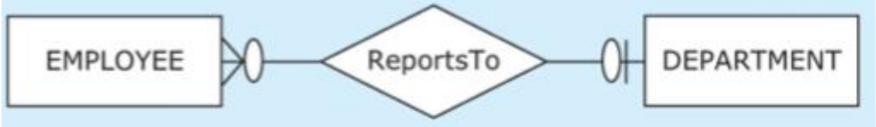
Cardinality Constraints - task

Task 3 : Draw out the ERD for the following description.

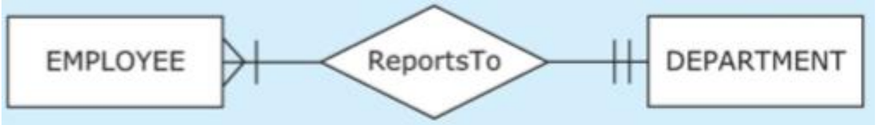
- Each employee reports to exactly one department. A department may have many employees reporting to it, but it does not have to have any.



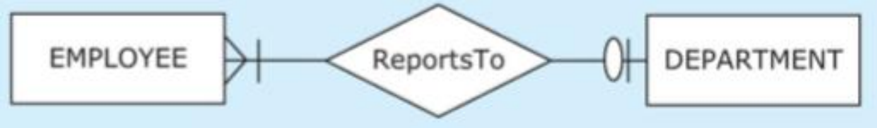
- An employee can report to one department or to no departments at all. A department may have many employees reporting to it, but it does not have to have any.



- Each employee reports to exactly one department. A department must have at least one employee reporting to it, but it may have many employees reporting to it.

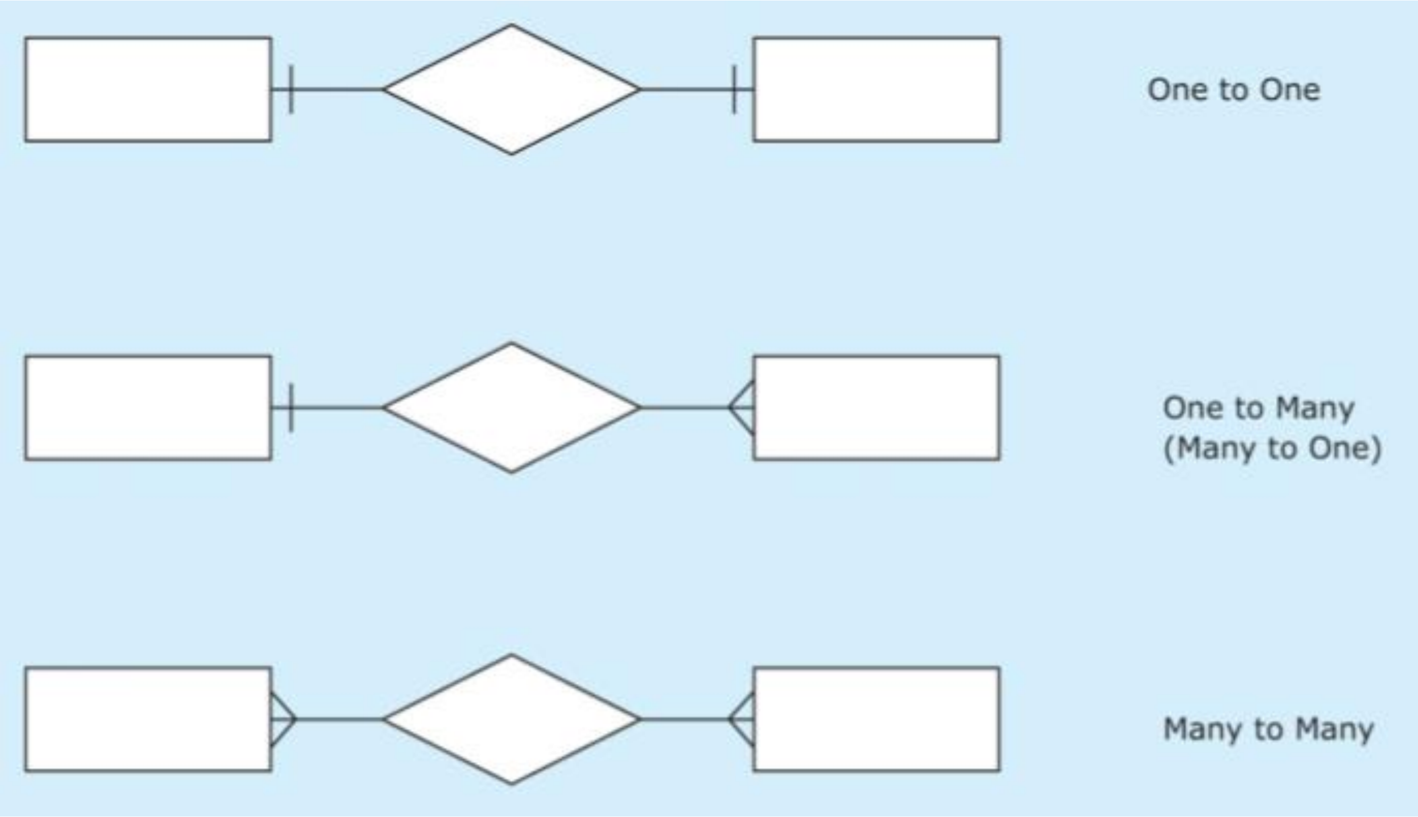


- An employee can report to one department or to no department at all. A department must have at least one employee reporting to it, but it many have many employees reporting to it.



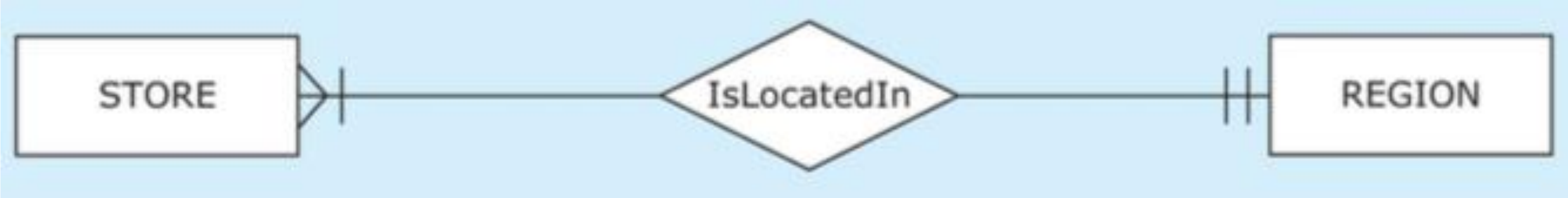
Types of relationships (Maximum cardinality-wise)

- The **maximum** cardinality on either side of the relationship can be either one or many.
- If we consider **maximum** cardinality without the minimum, we can classify each relationship between two entities into:
  - One-to-one relationship (1:1)
  - One-to-Many relationship (1:M)
  - Many-to-One relationship (M:1)
  - Many-to-Many relationship (M:N)

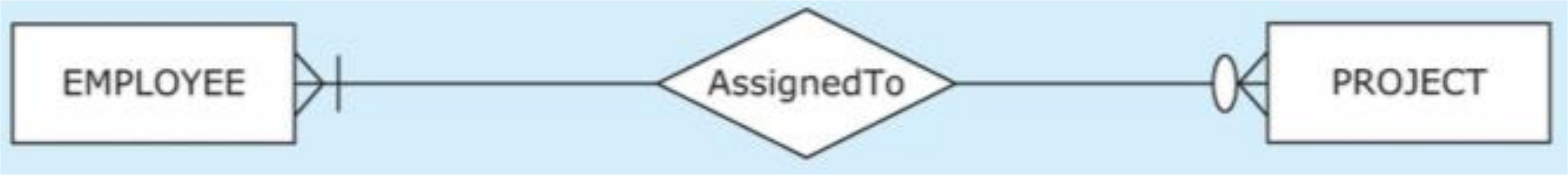


Types of relationships (Maximum cardinality-wise)

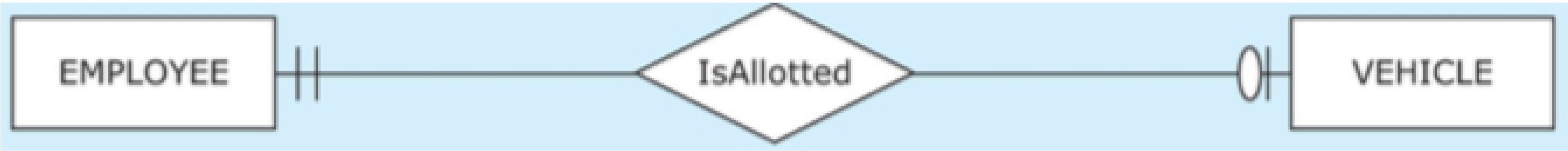
- Each store is located in exactly one region. Each region must have at least one store located in it, but it may have many stores located in it.



- An employee may be assigned to a number of projects, but he or she can also be assigned to none of the projects. A project must have at least one employee assigned to it, but it may have many employees assigned to it.

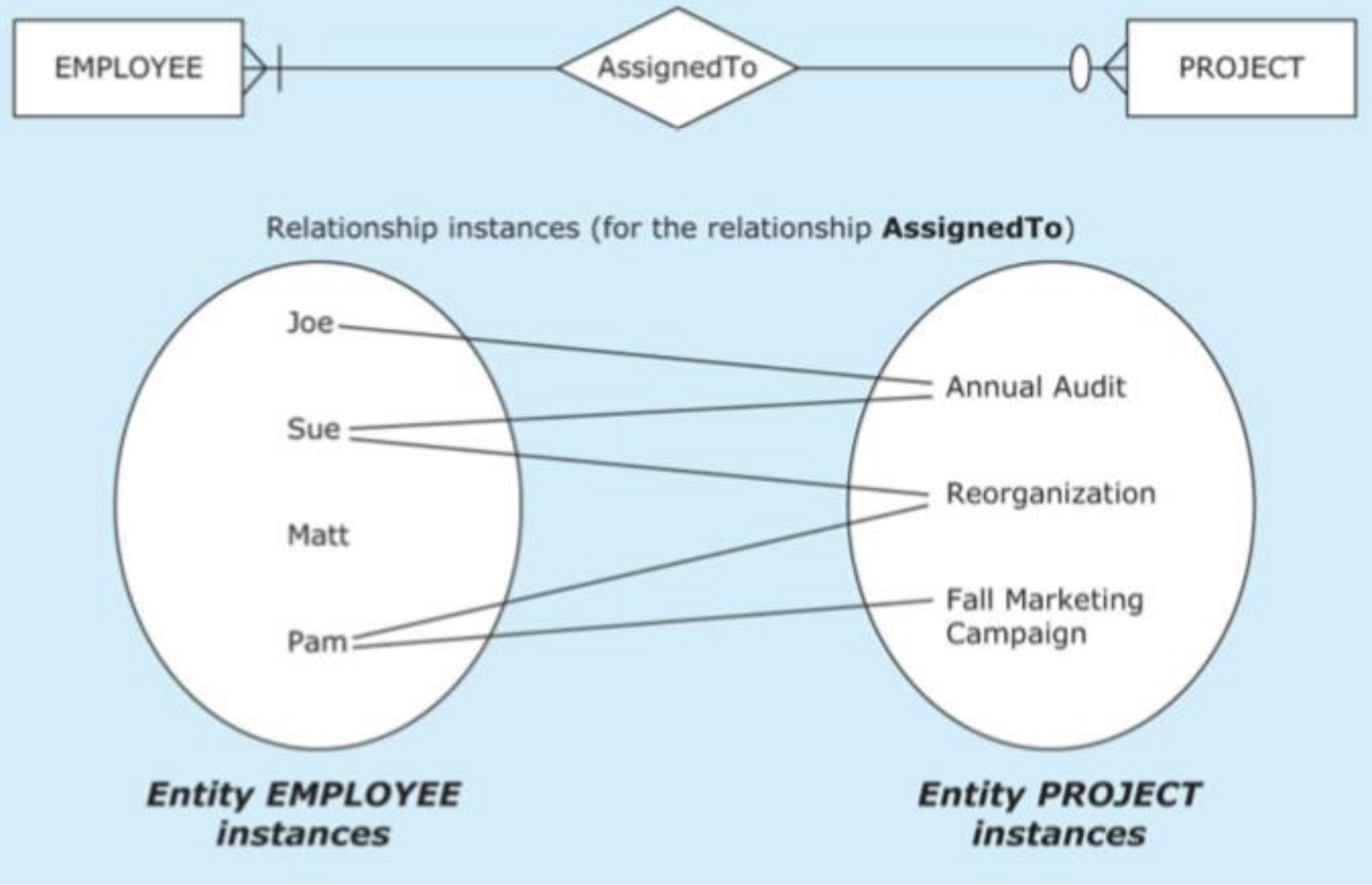


- Each employee is allotted either one vehicle or no vehicles. Each vehicle is allotted to exactly one employee.



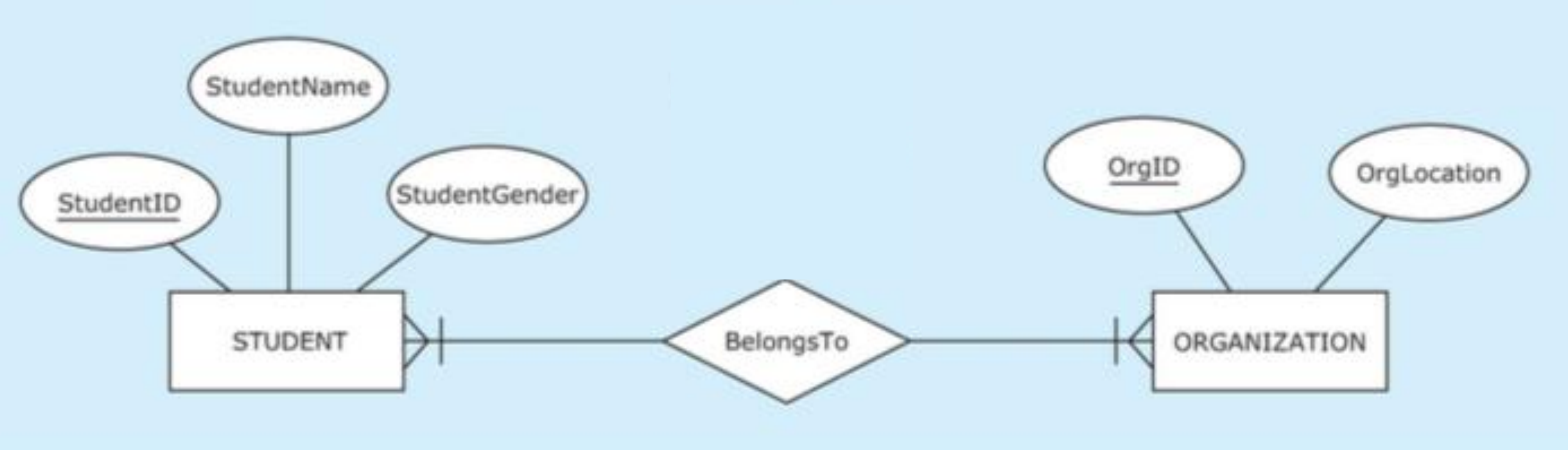
Relationships and relationship instances

- Relationship instances occur when an instance of one entity is related to an instance of another entity via a relationship.
- **Relationship** are depicted in the ERD, but relationship instances are not in ERD. Instead, **relationship instances** are recorded in the database based on the ERD. (Just like entity vs. entity instances)

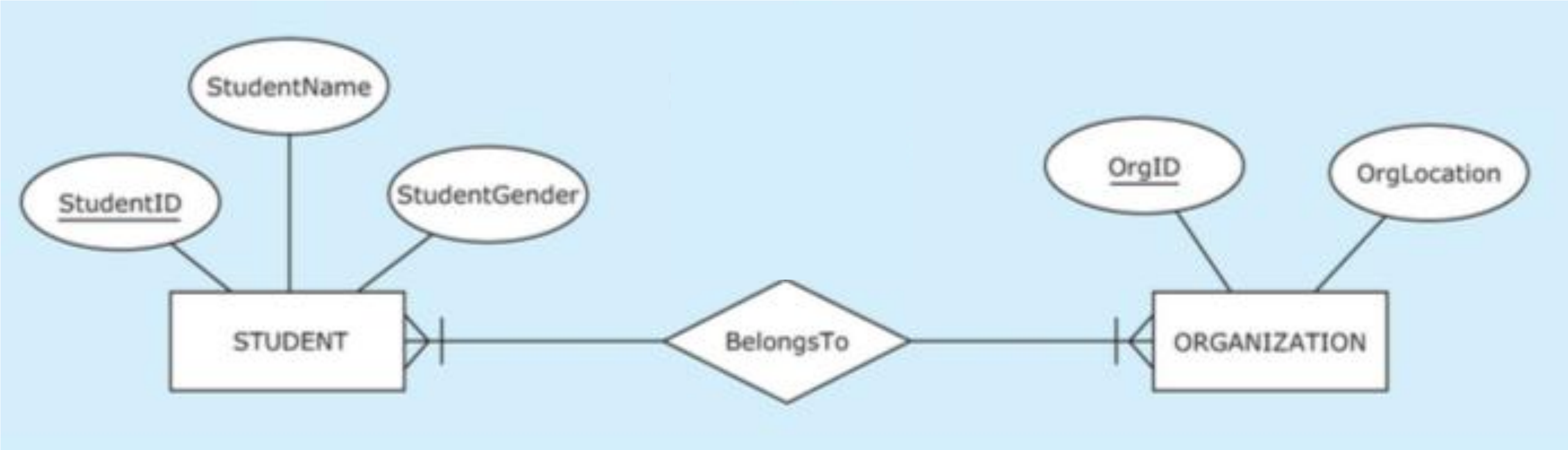

EMPLOYEE
AssignedTo
PROJECT
Relationship instances (for the relationship **AssignedTo**)
Joe
Sue
Matt
Pam
Annual Audit
Reorganization
Fall Marketing Campaign
Entity EMPLOYEE instances
Entity PROJECT instances

Relationship attributes

- *The database will keep track of students and campus organizations.*
- *For each student, we will keep track of his or her unique student ID, and his or her name and gender.*
- *For each organization, we will keep track of its unique organization ID and the location.*
- *Each student in the database belongs to at least one organization and can belong to multiple organizations.*
- *Each organization in the database has at least one student belonging to it and can have multiple students.*



Relationship attributes

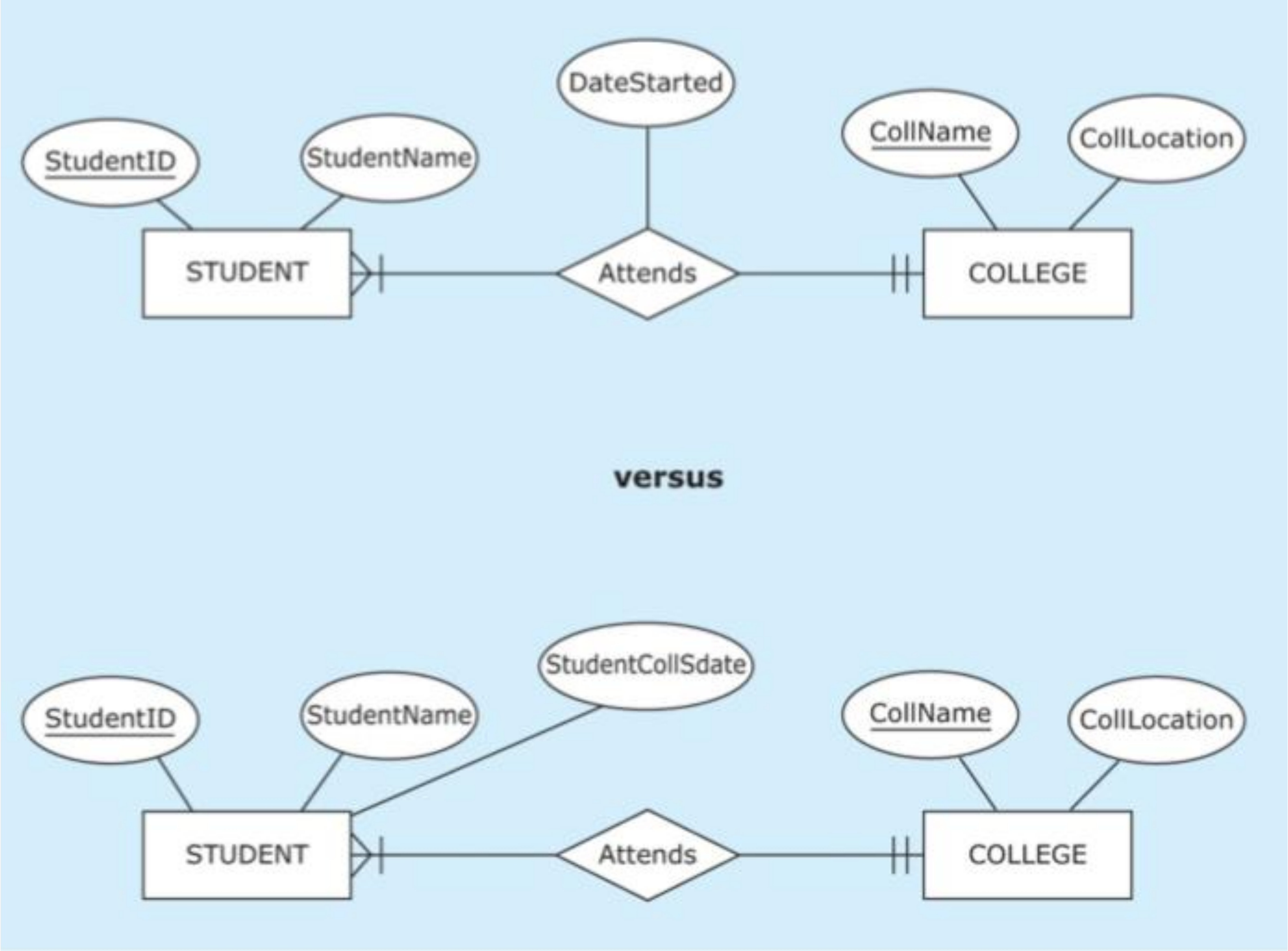


Task 4 : Note a student can belong to many organizations, and thus can have different roles (e.g. president, vice president, etc.) in each organization. Then if we want to record the students' roles in organizations, where should we put the 'role' attribute?

- In such cases, a relationship can have its own attributes (**relationship attributes**)
- It is useful for **many-to-many relationships** like this example

Whether 1:1 or 1:M relationships can have attributes?

Task 5 : Is it **necessary** for 1:1 or 1:M relationship to have an attribute?



No. It is only necessary for some M:N relationships.

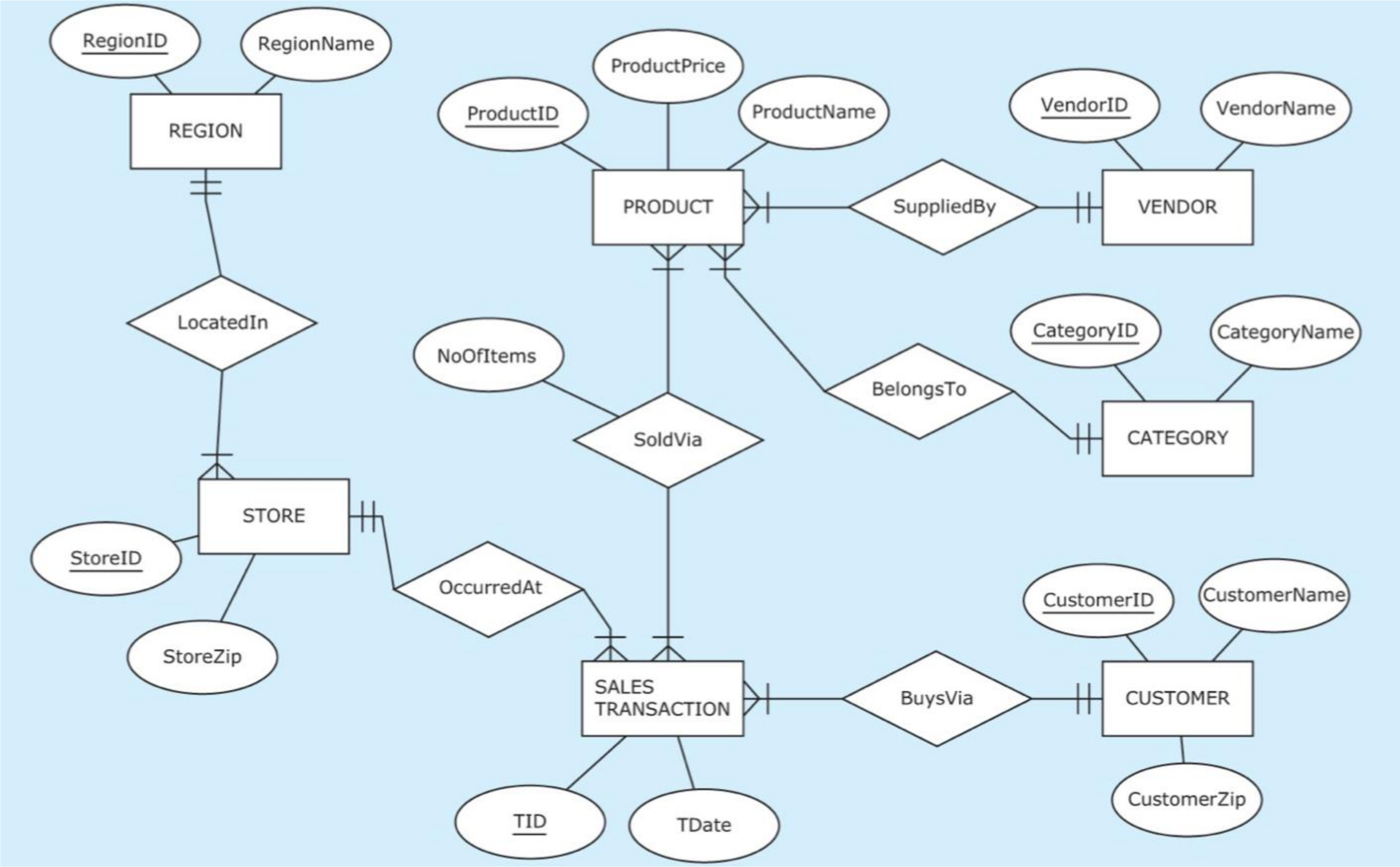
Draw the ERD - task

Task 6 : Draw the ERD based on the following database requirements (20min)

1. *For each **product** being sold: a product ID (unique), product name, and price*
2. *For each **category** of product: category ID (unique) and category name*
3. *For each **vendor**: vendor ID (unique) and vendor name*
4. *For each **store**: store ID (unique) and zip code*
5. *For each **region**: region ID (unique) and region name*
6. *For each **sales transaction**: transaction ID (unique) and date of transaction*
7. *For each **customer**: customerID (unique), customerName, customerZip*

1. *Each **product** is supplied by exactly one vendor; Each **vendor** supplies one or more products.*
2. *Each **product** belongs to exactly one category; Each **category** contains one or more products.*
3. *Each **store** is located in exactly one region; Each **region** contains one or more stores.*
4. *Each **sales transaction** occurs at one store; Each **store** has one or more transactions occurring at it.*
5. *Each **customer** buys via one or more sales transactions; Each **sales transaction** involves exactly one customer.*
6. *Each **product** is sold via one or more sales transactions. Each **sales transaction** includes one or more products.*
7. *For each instance of a product being sold via a sales transaction, the number of items being sold is recorded.*

Draw the ERD - task answer



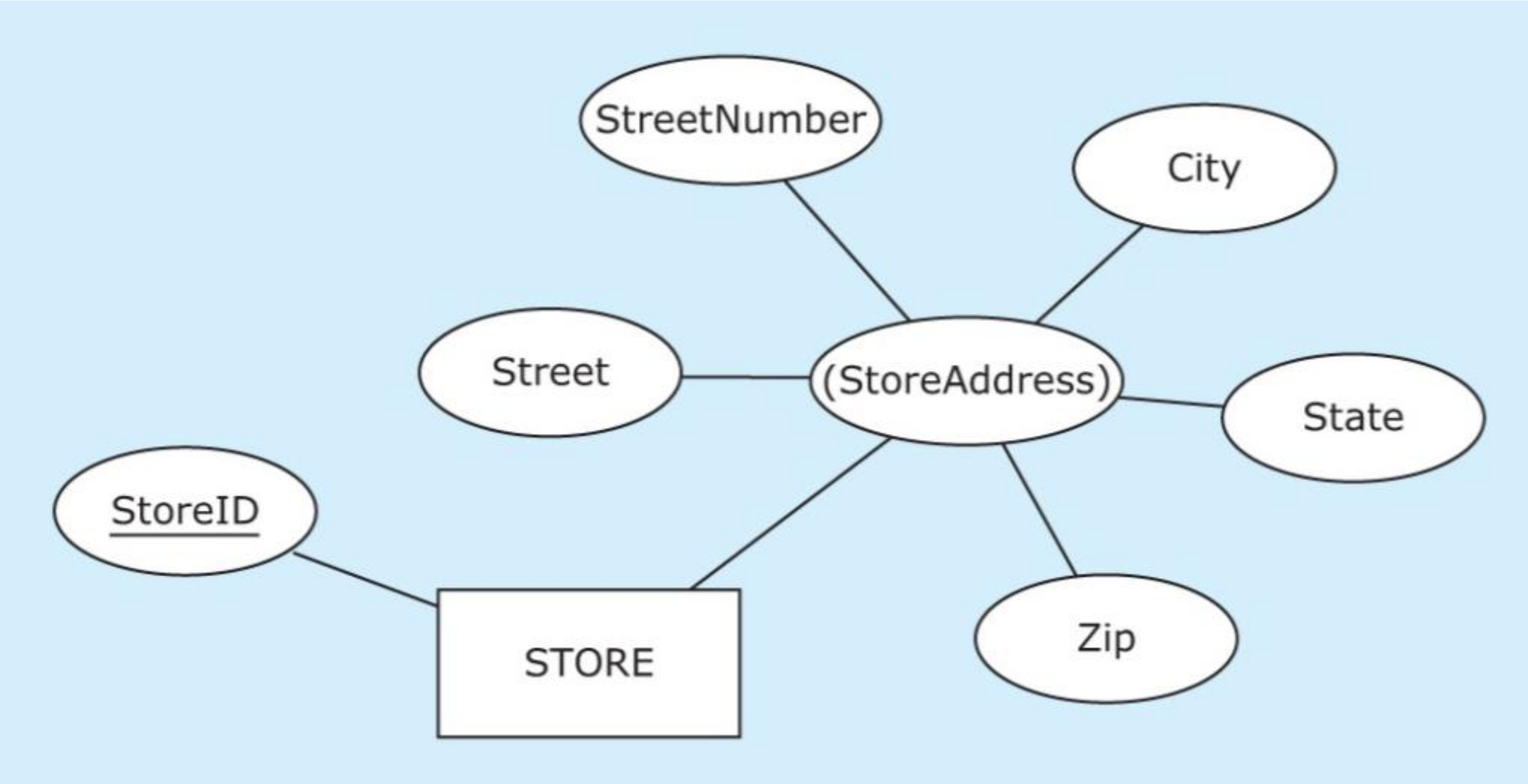
Draw the ERD - follow up questions

Task 7 : Follow up questions for the ERD we just draw.

1. What in particular would you like to record for each customer?
   - Customer ID, name, zip code
2. Is customer ID value unique for each customer?
   - Yes
3. Is customer name value unique for each customer?
   - No
4. Does each product belong to one or multiple categories?
   - One
5. Does each product come from only one vendor?
   - Yes
6. Would you ever record a vendor who does not supply any products?
   - No
7. Do you keep track of customers that did not buy anything yes (i.e. customers that were not involved in any sales transitions yet)?
   - No

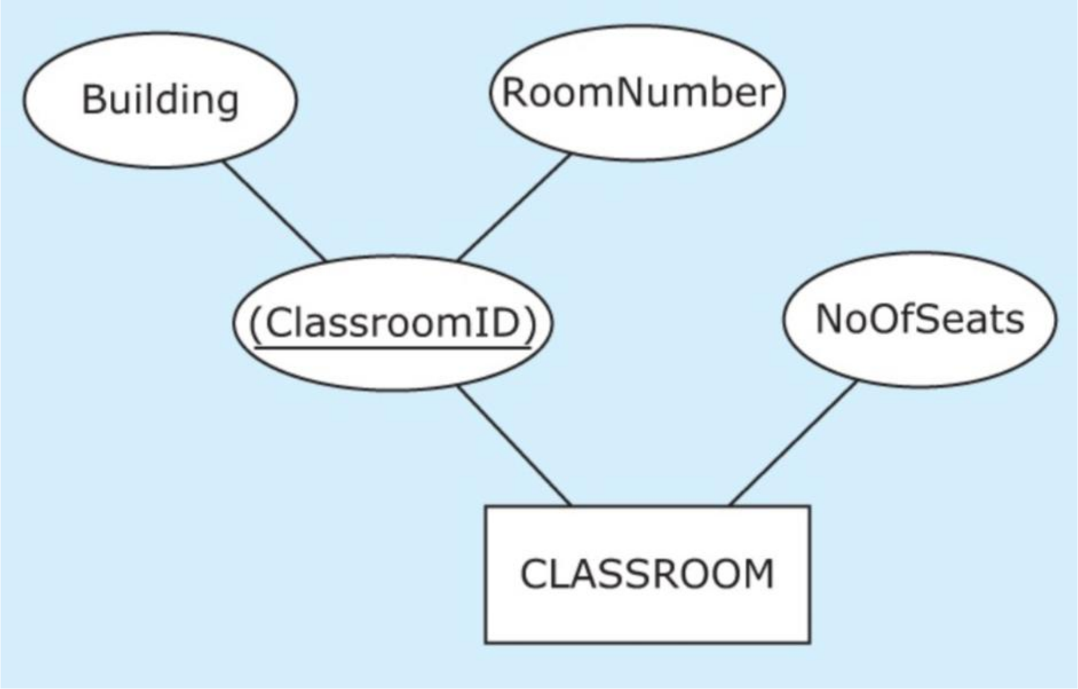
Composite attributes

- Other than the regular attributes shown before, ERD can depict several other types of attributes.
- **Composite attribute**: an attribute that is composed of serval attributes (those connected to it in ERD).



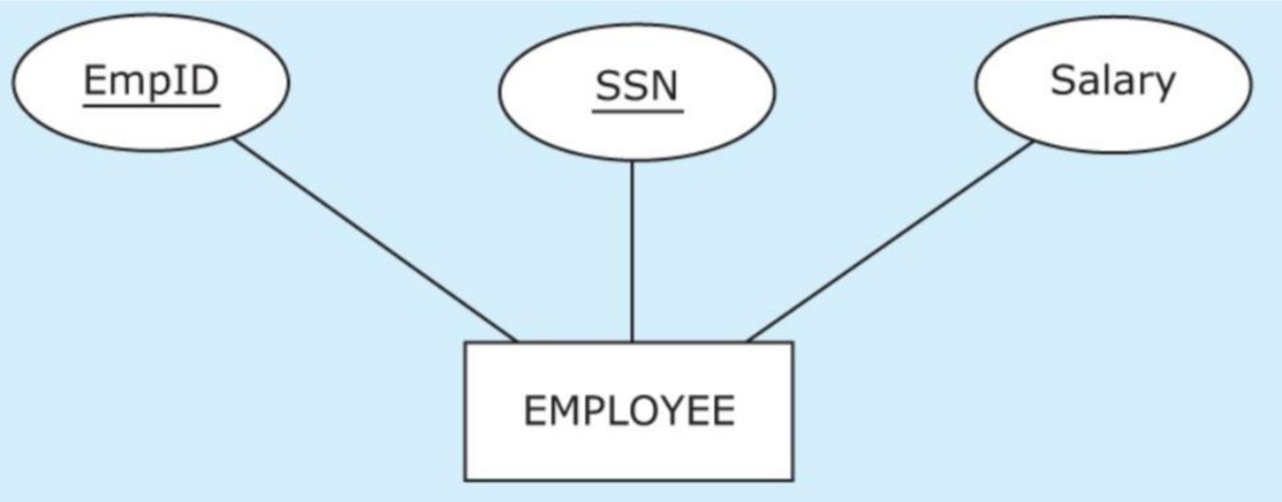
Composite attributes: unique

- CLASSROOM (shown right) entity has three attributes, **none** is **unique**
  - The *Building* attribute can be the same for many classrooms located in the same building, e.g. 201, 301 in A
  - *Room Number* can be the same across buildings, e.g. 301 in A or B
  - The *No. Of Seats* can be the same across classrooms
- The combination of "*Building + Room Number*" is **unique**
  - e.g. *B-410* (yes, this is exactly how we label this classroom)



Multiple unique attributes (Candidate keys)

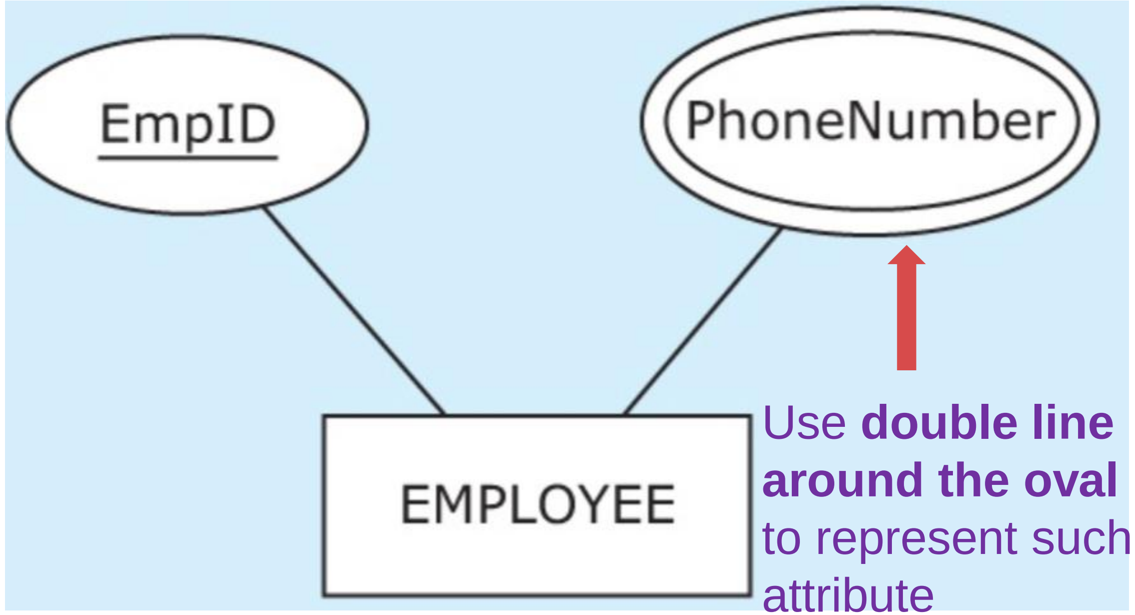
- Entity can have more than one unique attributes
- In this case, each unique attribute is called **candidate key**, i.e. candidates of primary key.



Multivalued attributes

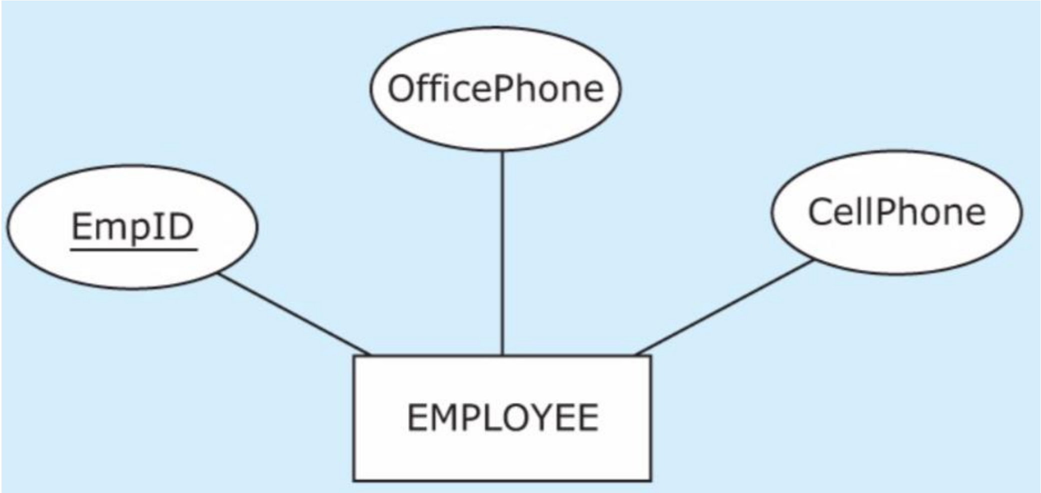
- **Multivalued attribute**: where instances of an entity can have multiple values for the same attribute.
- Used in cases in which there is a **variable number of values** that can be assigned to the particular attribute of the entity.

We can have >1 phone numbers



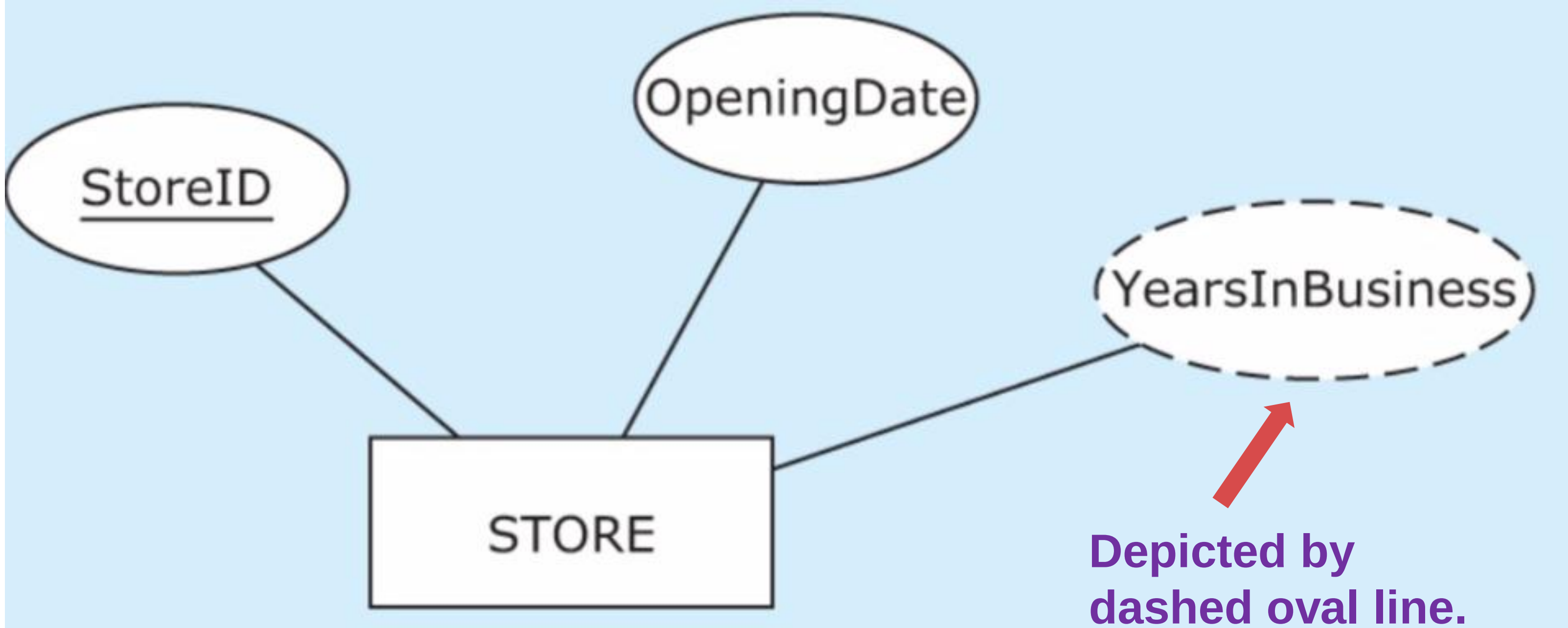
Use **double line around the oval** to represent such attribute

If only 2 different phone numbers, they may be split as **two attributes.**



Derived attributes

- **Derived attribute** is an attribute whose value will not be permanently stored in a database.
- The value of a derived attribute will be **calculated** from the stored values of other attributes and/or additional data.

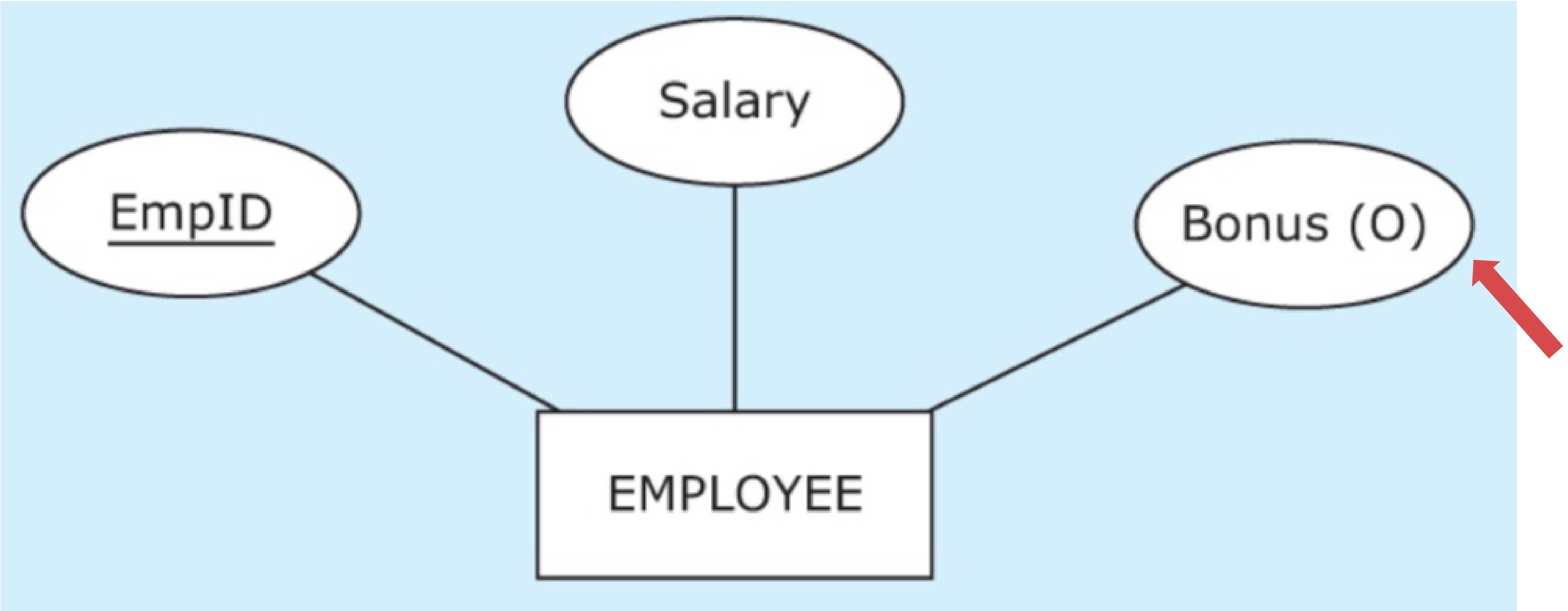


Depicted by dashed oval line.

Optional attributes

- **Optional attribute**: Such attributes are allowed to not have a value.
- e.g. each employee has salary, but not necessarily a *bonus*

  (otherwise it is not *bonus* - “extra reward”)



Depicted by a capital letter O in the parentheses, at the end of the attribute’s name

Task – various attributes

Task 8 :

A regular entity must have at least one of the following attributes:

[✓ A. Unique attribute]

B. Composite attribute

C. Multivalued attribute

D. Derived attribute

Task – various attributes

Task 9 :

The value of which attribute in the following are NOT permanently stored in a database

A. Unique attribute

B. Composite attribute

C. Multivalued attribute

[✓ D. Derived attribute]

Task – various attributes

Task 10 :

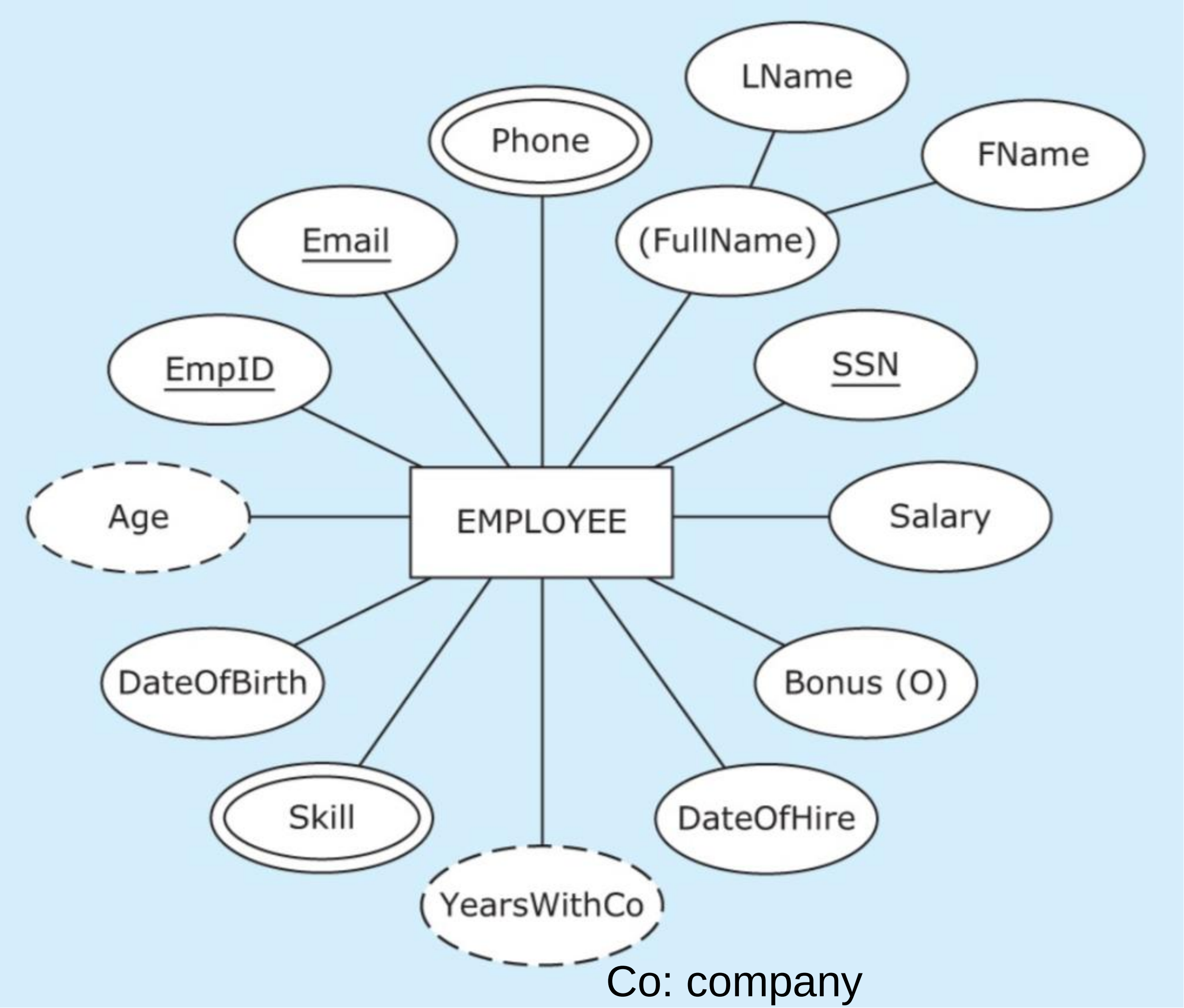
1. Find out all **multivalued attributes** and explain its meaning.

2. Find out all **optional attributes** and explain its meaning.

3. Find out all **derived attributes** and explain its meaning.

4. Find out all **multiple unique attributes** and explain its meaning.

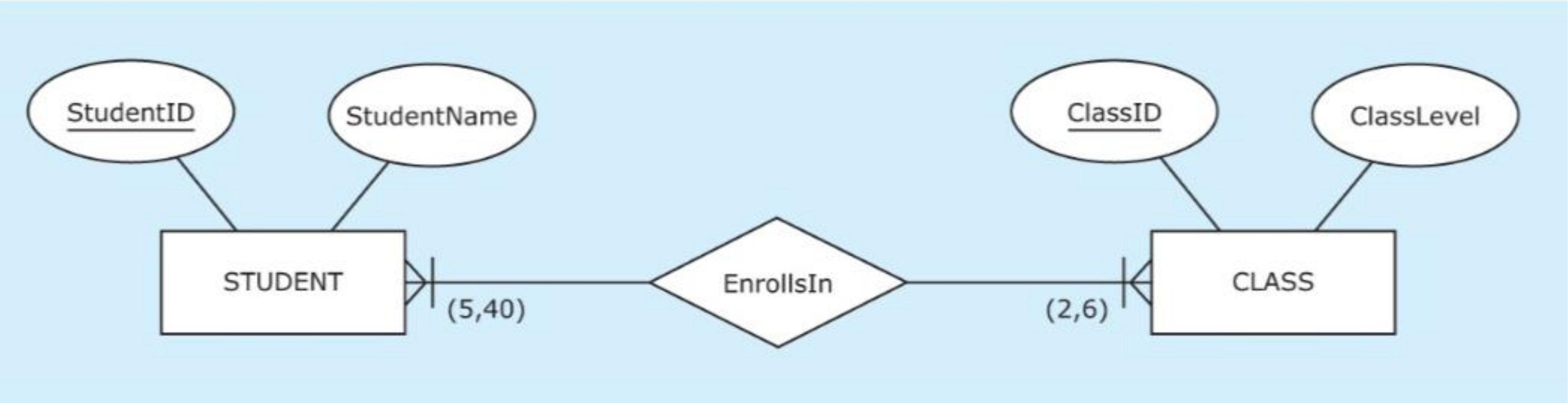
5. Find out all **composite attributes** and explain its meaning.



Co: company

Exact minimum and maximum cardinality in relationships

- Sometimes we know the exact minimum and/or maximum cardinality in the relationships
- Notation: (min, max), placed on the relationship lines
- e.g. each student must enroll in at least 2 and at most 6 classes, while a class must have at least 5 and at most 40 students enrolled.



Task – maximum and minimum cardinality

Task 11 :

Which of the following is a legitimate exact minimum and maximum cardinality?

A. (20,10)

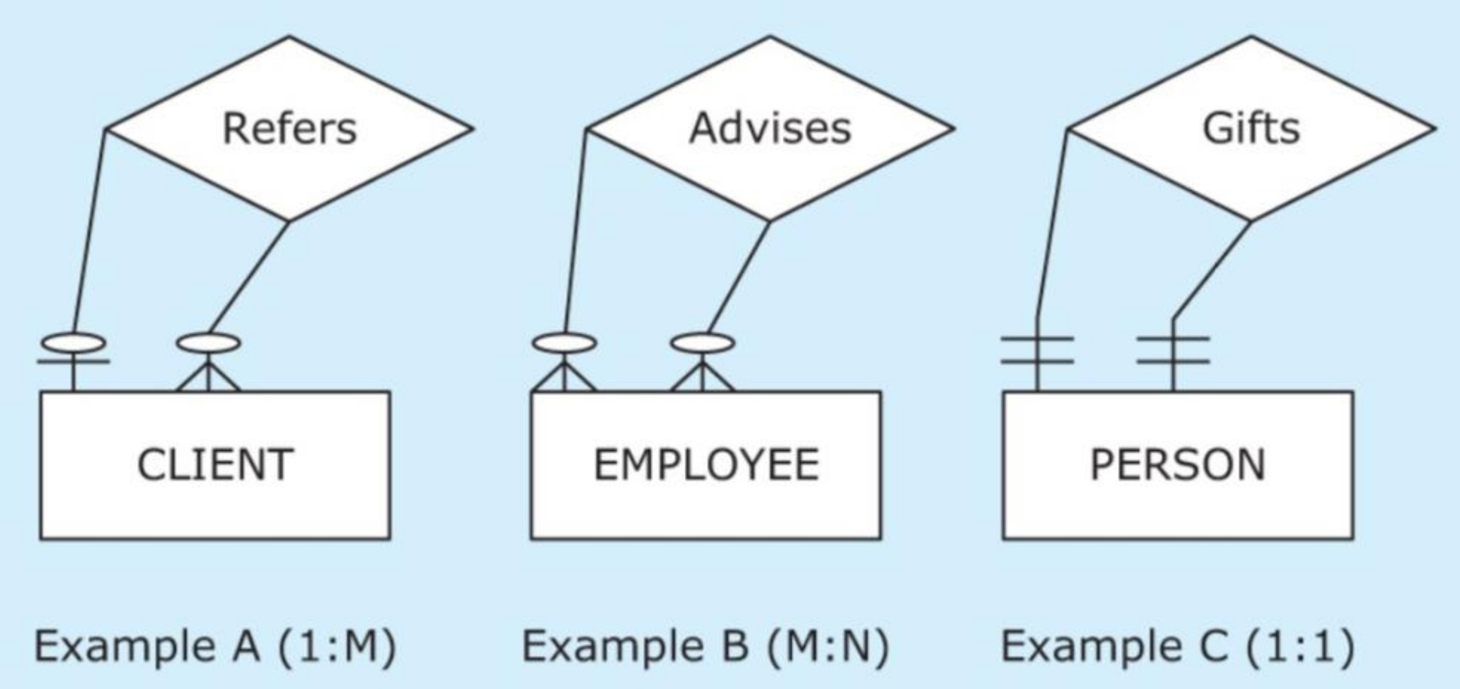
B. (5,0)

[✓ C. (5,10)]

D. (5,4)

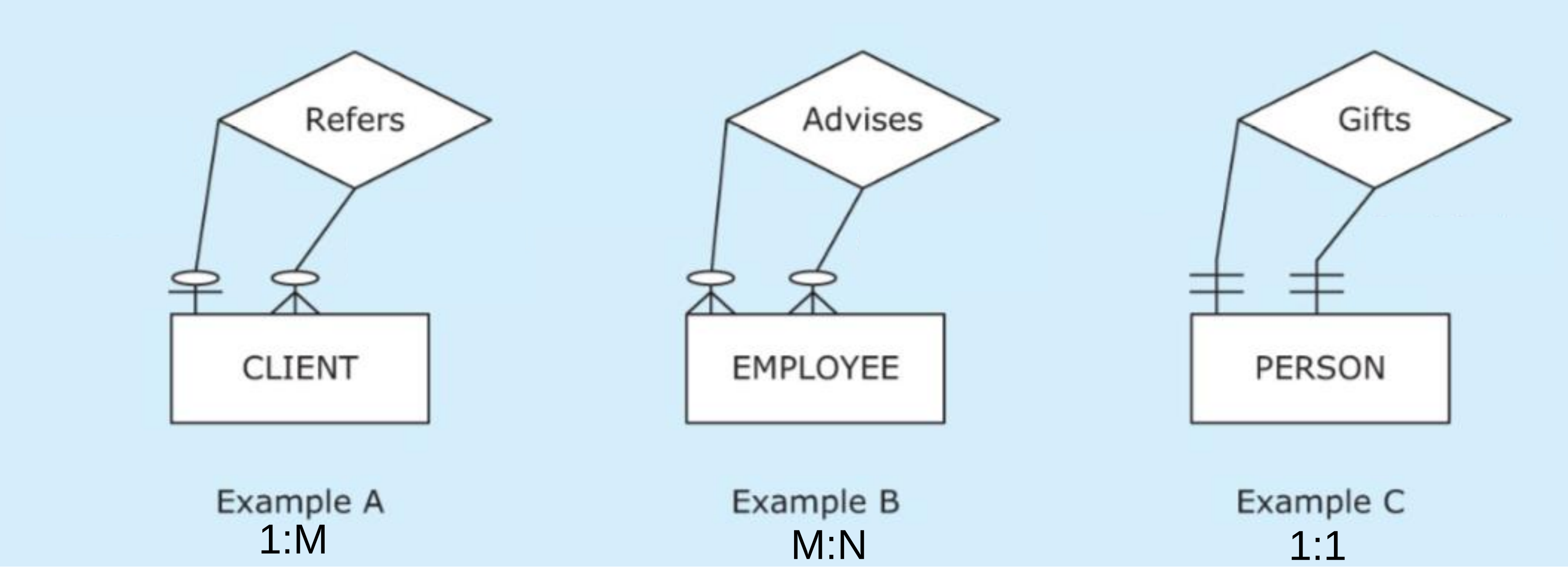
Degree of relationships and unary relationship

- The **degree of a relationship**: indicates **how many entities** are involved in a **relationship**.
- **binary relationship**: a relationship of **degree of 2**.
  - So far, every relationship we have examined involved two entities.
  - Vast majority of relationships in business-related ERD are binary.
- **unary relationship** (**recursive relationship**): A relationship of **degree 1**, occurs when an entity is involved in a relationship with itself. 1:M, M:N, 1:1



Unary relationships and relationship role

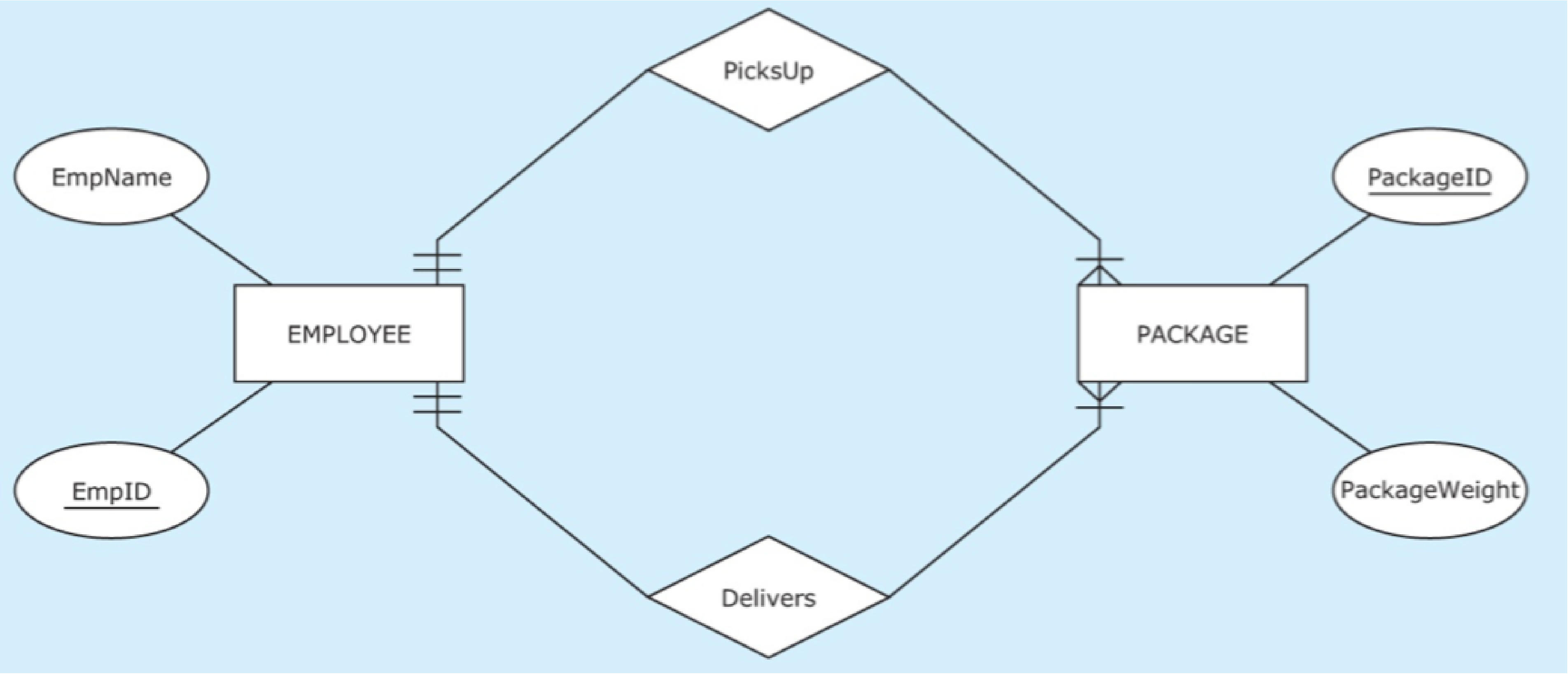
- **Relationship role:** further classification of entity instances in a relationship.
- e.g. client -> “referring entity” + “referred entity”
- So useful in unary relationships.



- A: a client can refer (recommend) many clients but does not have to refer any. Each client is either referred by one other client or is not referred at all.
- B: An employee can advise many employees but does not have to advise any. An employee can be advised by many employees but does not have to be advised by any.
- C: In a database for a gift-giving event (secret santa), each person gives a gift to exactly one person and each person receives a gift from exactly one person.

Multiple relationships between two entities

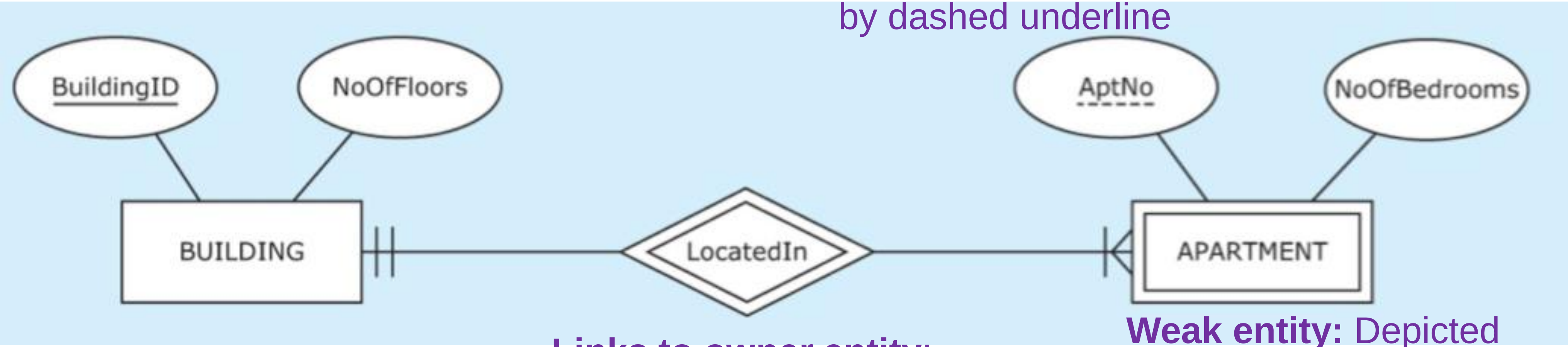
- It is not unusual: two entities in an ERD can have more than one relationship.



Weak entity

- **Weak entity**: do not have a unique attribute (NOTE, a regular entity must have unique attributes)
  - must be associated with its **owner entity** via an identifying relationship
  - e.g. APARTMENT is associated with BUILDING via “LocatedIn” relationship

Partial key, depicted by dashed underline

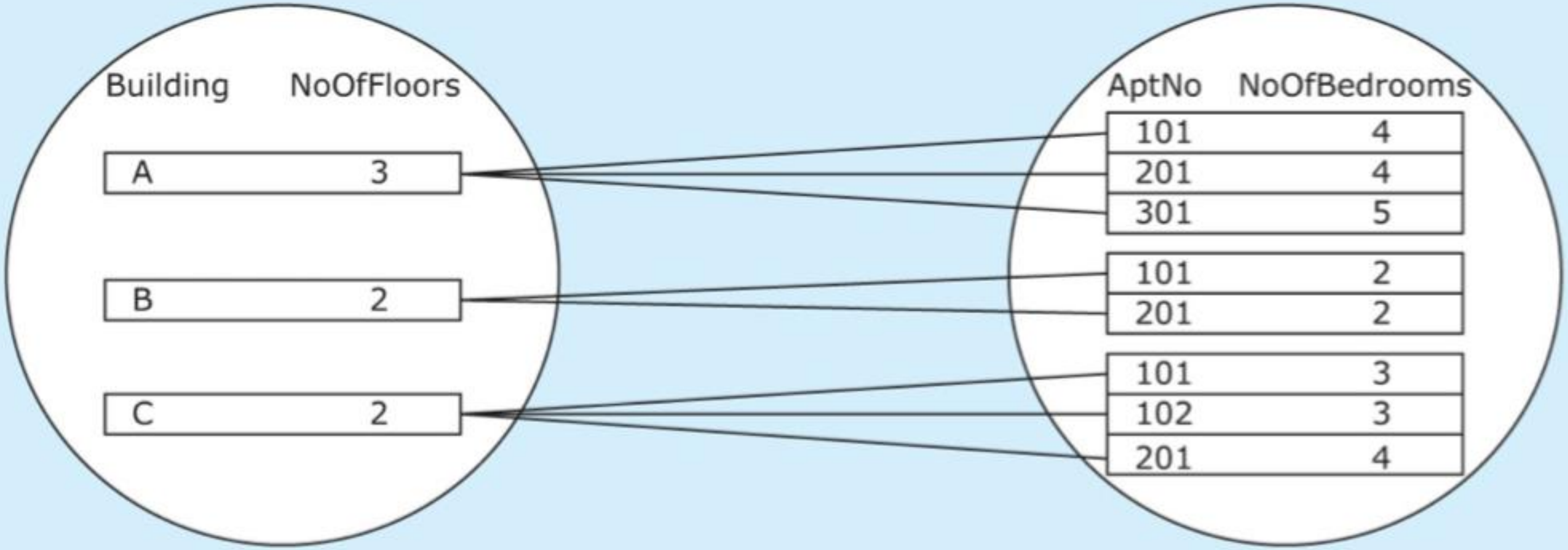


Links to owner entity: Depicted by double line diamond.

Weak entity: Depicted by double line rectangle.

- ***Partial key*** is unique within the **owner entity**’s unique attributes
  - e.g. Apartment No. is not unique (e.g. 301 in two buildings), but unique within the same <u>Building</u>

Weak entity



Building	NoOfFloors
A	3
B	2
C	2

AptNo	NoOfBedrooms
101	4
201	4
301	5
101	2
201	2
101	3
102	3
201	4

Entity BUILDING instances

Relationship instances (for the relationship **LocatedIn**)

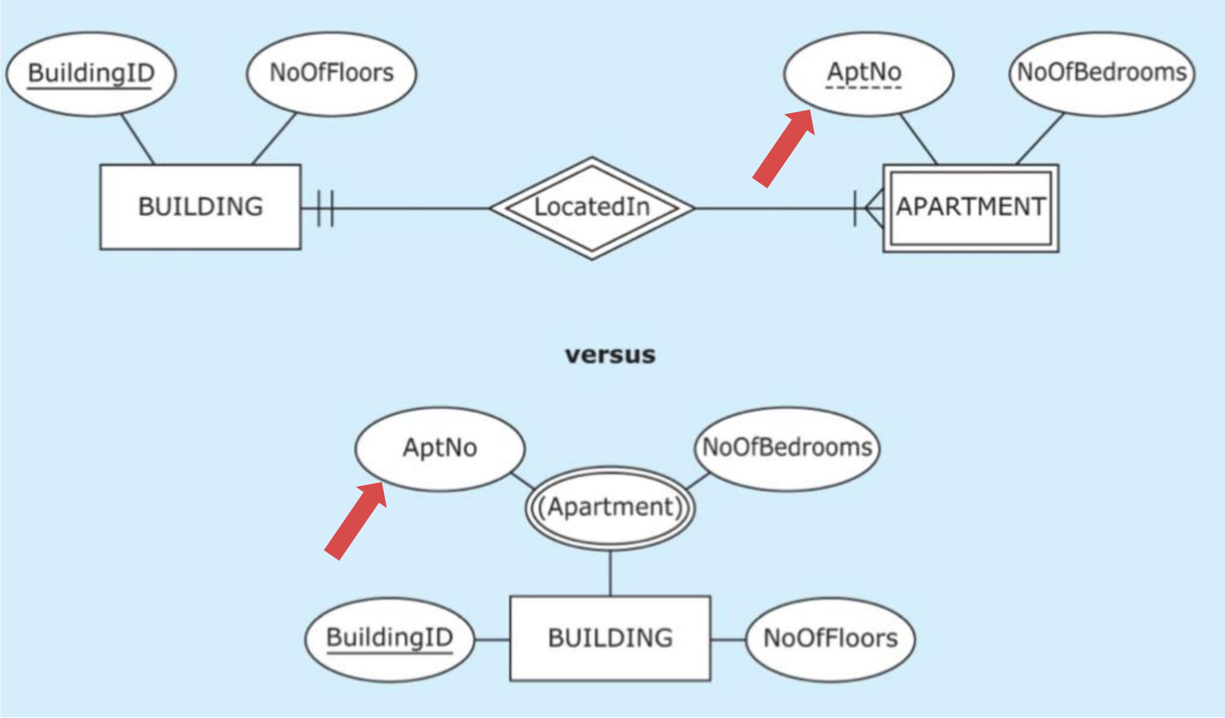
Entity APARTMENT instances

Weak entity vs. multivalued composite attribute

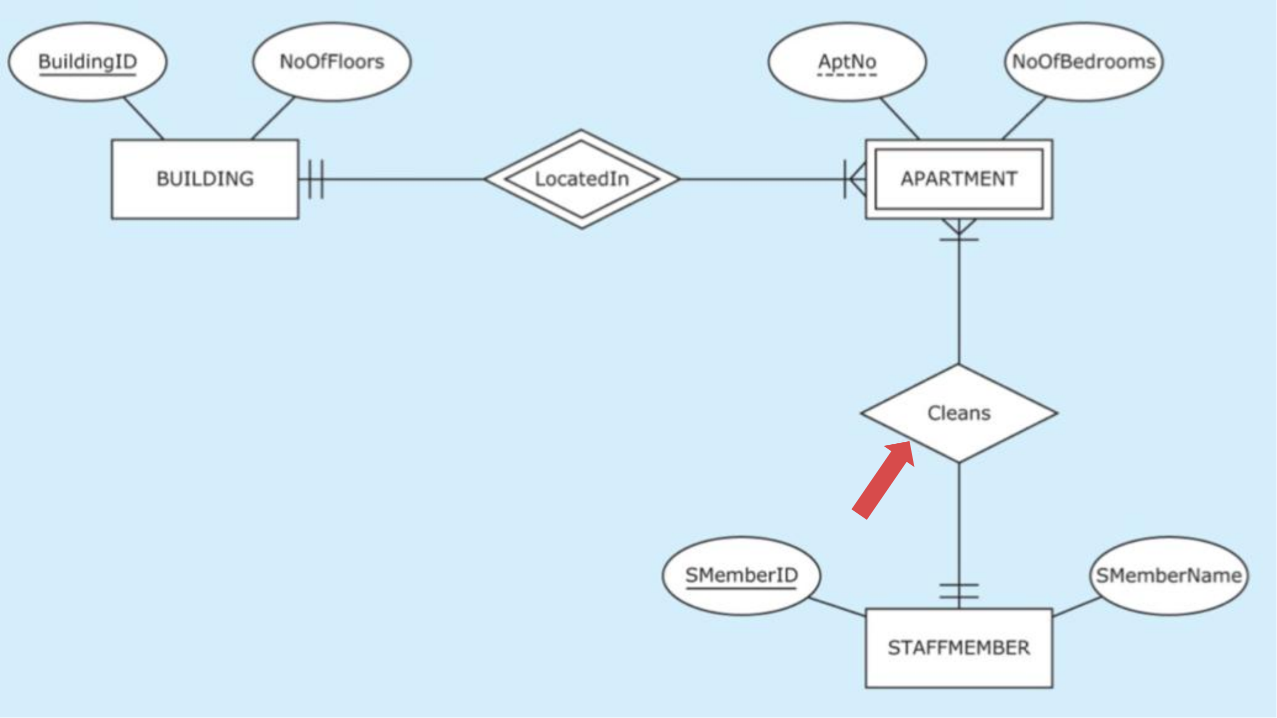
Weak entities can be replaced by **multivalued composite attributes** (since both represent similar requirements):
e.g. "LocatedIn" relationship to owner entity -> an attribute of entity

However, there are **two meaningful differences**:

- 1) A weak entity can have a *partial key*



- 2) A weak entity can have regular *relationships* with other entities



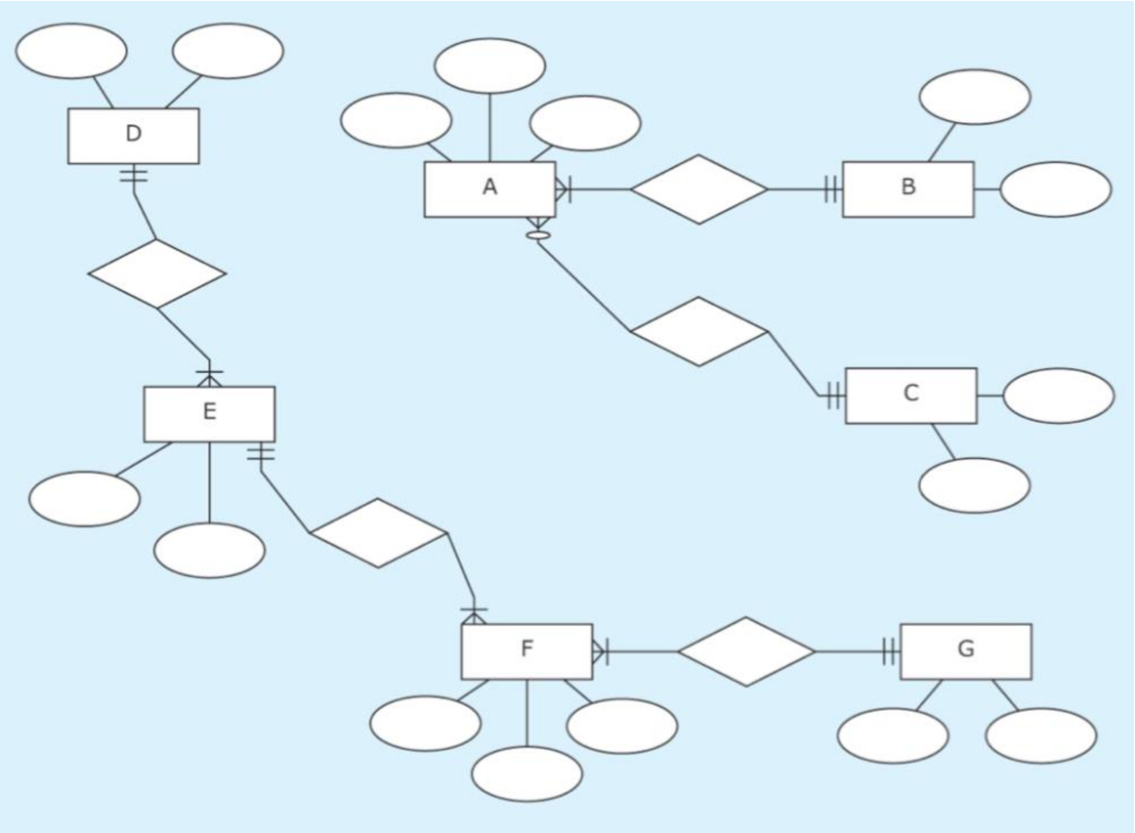
Naming Conventions

- Entity names: singular (rather than plural) nouns; (UPPER CASE)
  - *STUDENT*, *STORE* (good)
  - *STUDENTS*, *STORES* (bad)
- Relationship names: verbs or verb phrases rather than nouns
  - *Inspects*, *Manages*, *BelongsTo* (good)
  - *Inspection*, *Management*, *Belonging* (bad)
- Attribute names: no duplications
  - both EMPLOYEE and CUSTOME have NAME, so use EmpName and CustName to distinguish them
- As brief as possible
  - *UNVIERSITY STUDENT* -> *STUDENT*
  - Number of Floors -> *NoOfFloor*

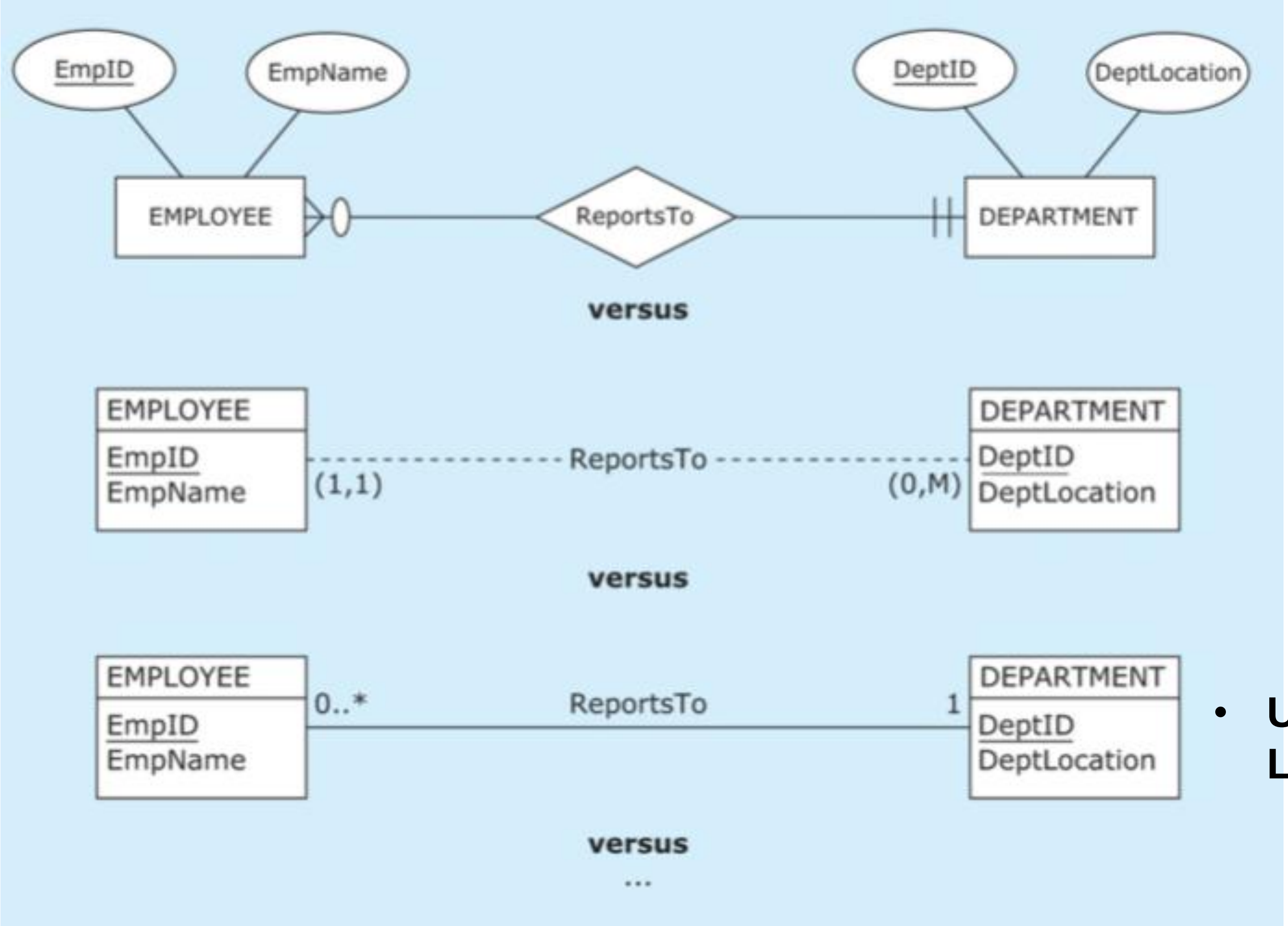
Connectivity and multiple ER Diagrams

- Connectivity in an ERD
  - each entity is always connected to all of the other entities, either via a direct relationship or indirectly via other entities and relationships
  - there is **a path** from one entity to every other entity

- If an ER schema contains entities that are **not connected** to other entities in the schema:
  - this becomes **multiple ER diagrams** (i.e. multiple separate databases)



Various ER Notation



- **UML (unified Modeling Language)**

Various format, but all depict the same concepts: entities, relationships, and attributes.

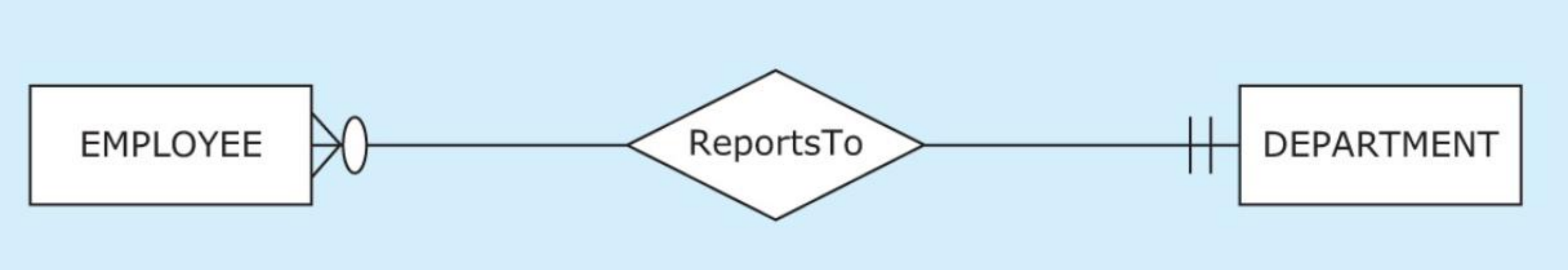
Task - database requirement and ERD

Task 12 :

Company X keeps track of its department and employees who report to those departments.

- Each employee reports to exactly one department.
- Each department can have many employees reporting to it but does not have to have any.

1. Draw an ERD yourself .

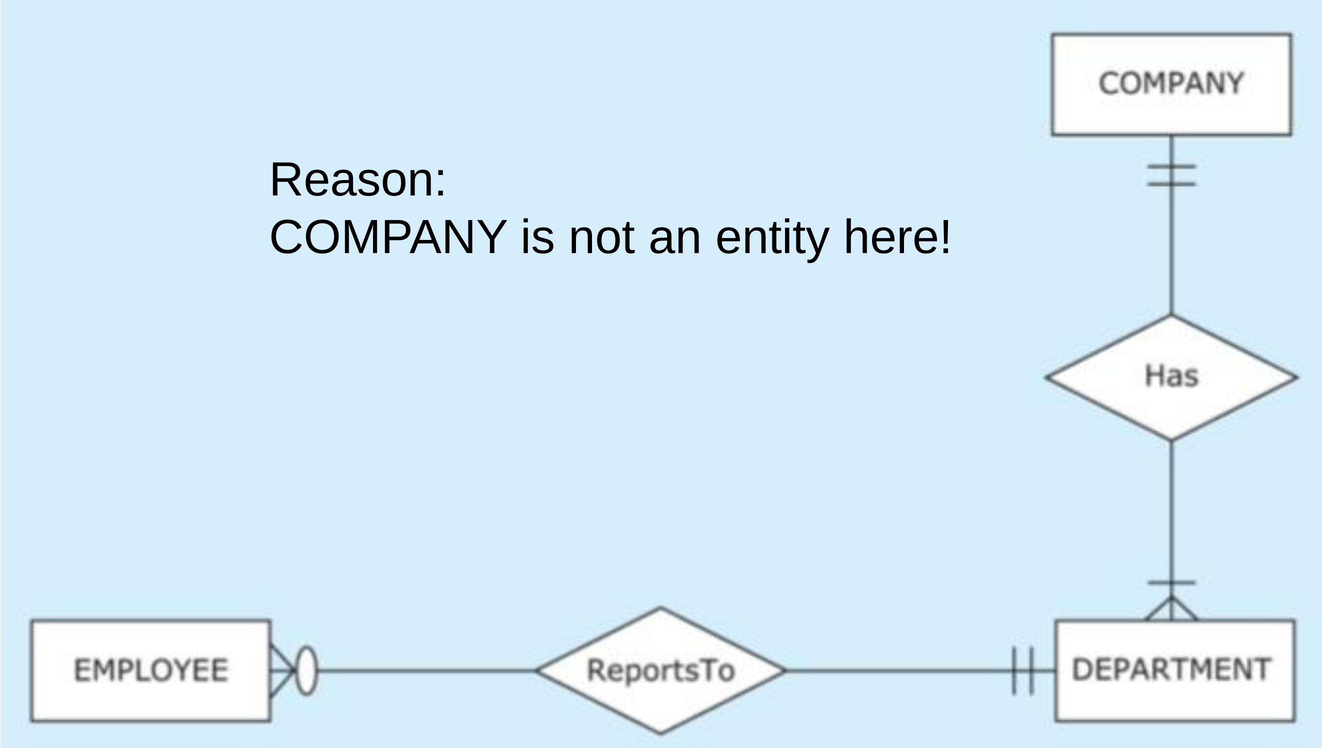


Task - database requirement and ERD

Task 12:
Company X keeps track of its department and employees who report to those departments.

- Each employee reports to exactly one department.
- Each department can have many employees reporting to it but does not have to have any.

2. Is the following ERD correct? (yes or no)



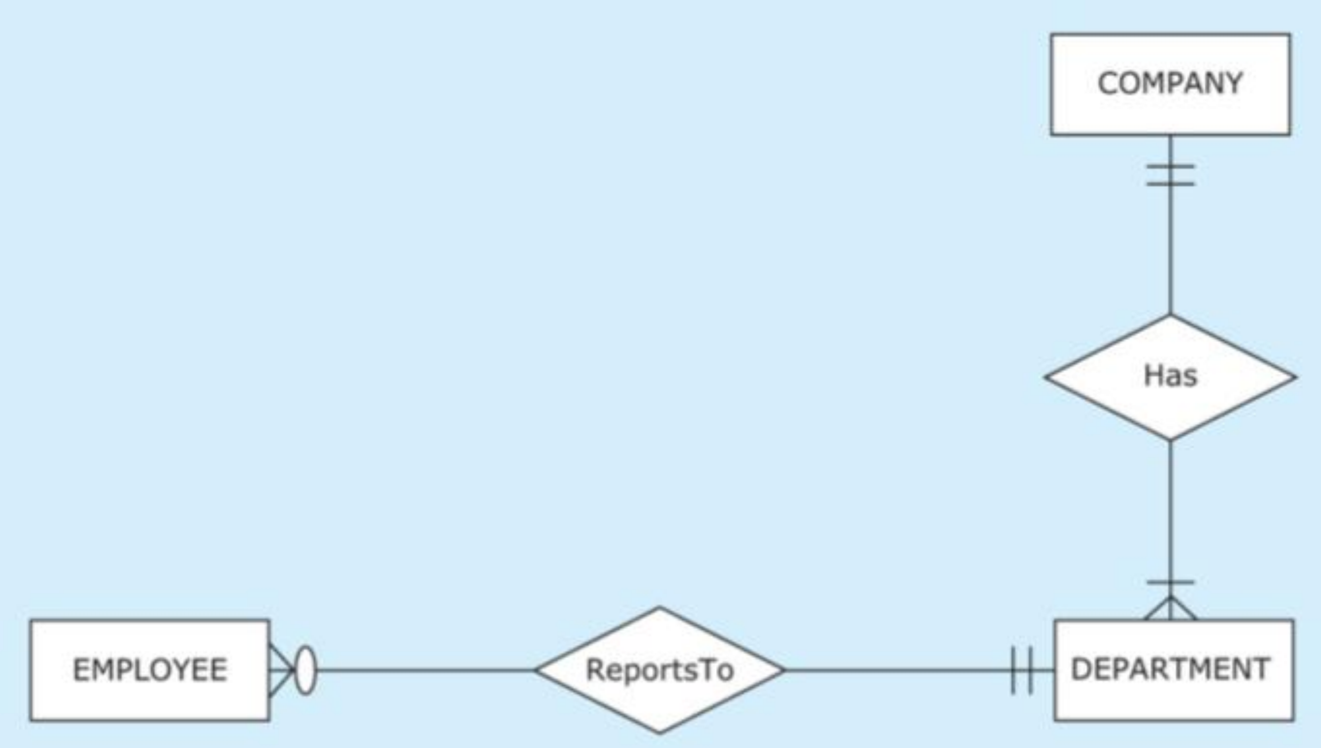
Reason:
COMPANY is not an entity here!

Task - database requirement and ERD

Task 12:
Company X keeps track of its department and employees who report to those departments.

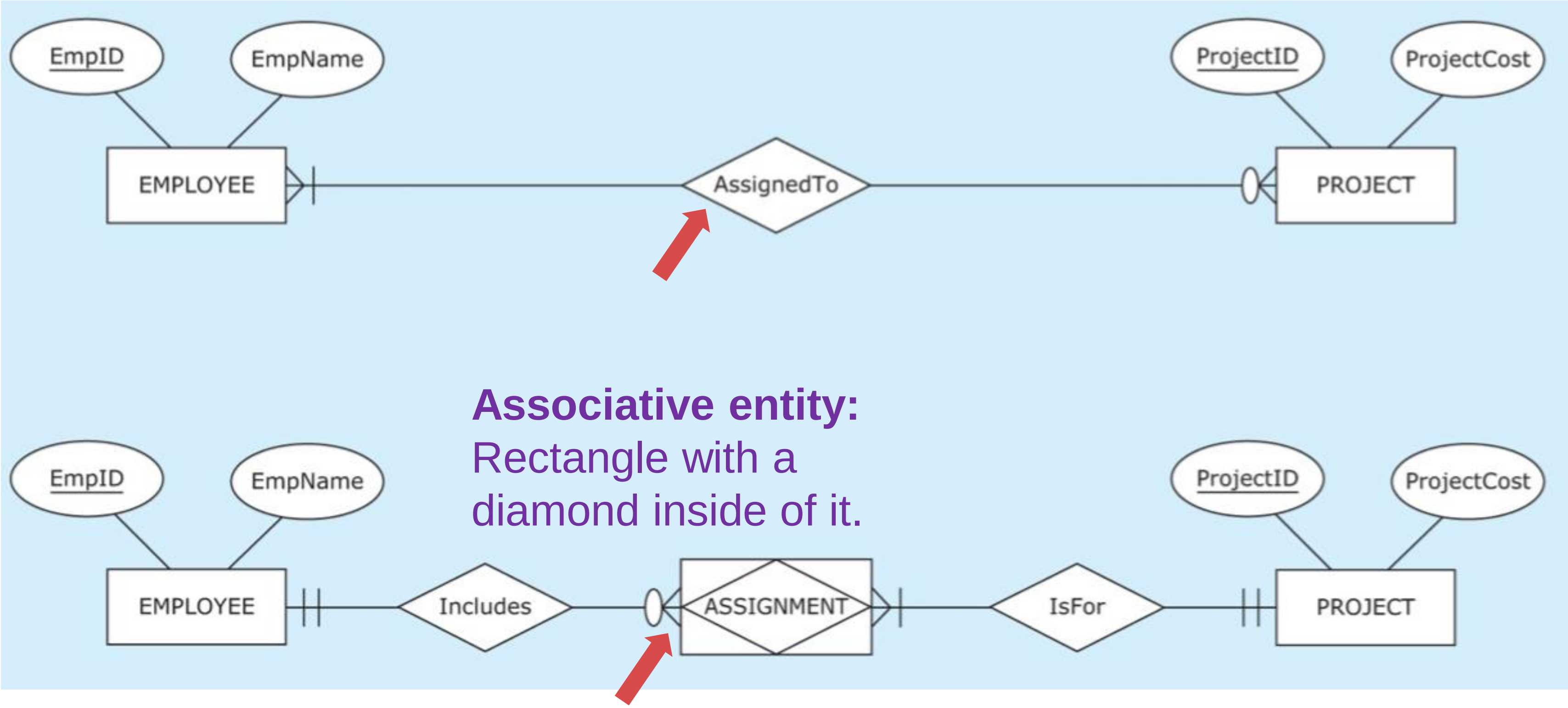
- Each employee reports to exactly one department.
- Each department can have many employees reporting to it but does not have to have any.

3. How should we correct it?



Associative Entities

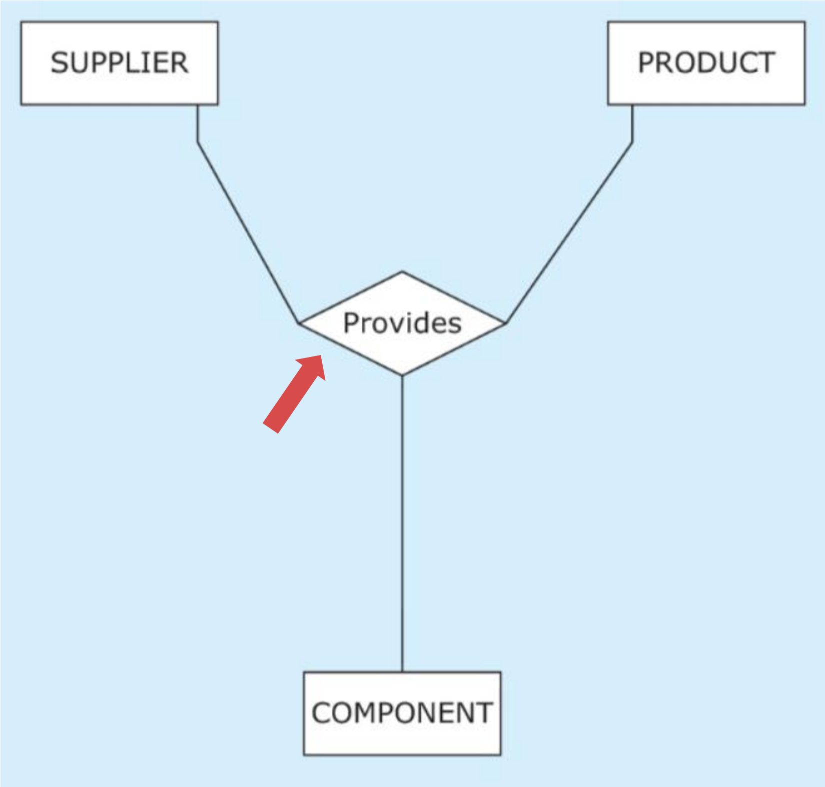
- An **associative entity** is *an alternative way* of depicting M:N relationships. Often **NO attributes** at all.
- For binary or unary relationship, associative entities are not necessary. But for relationship with a degree higher than 2, associative entity is useful to eliminate potential ambiguities in ERD (see next slide).

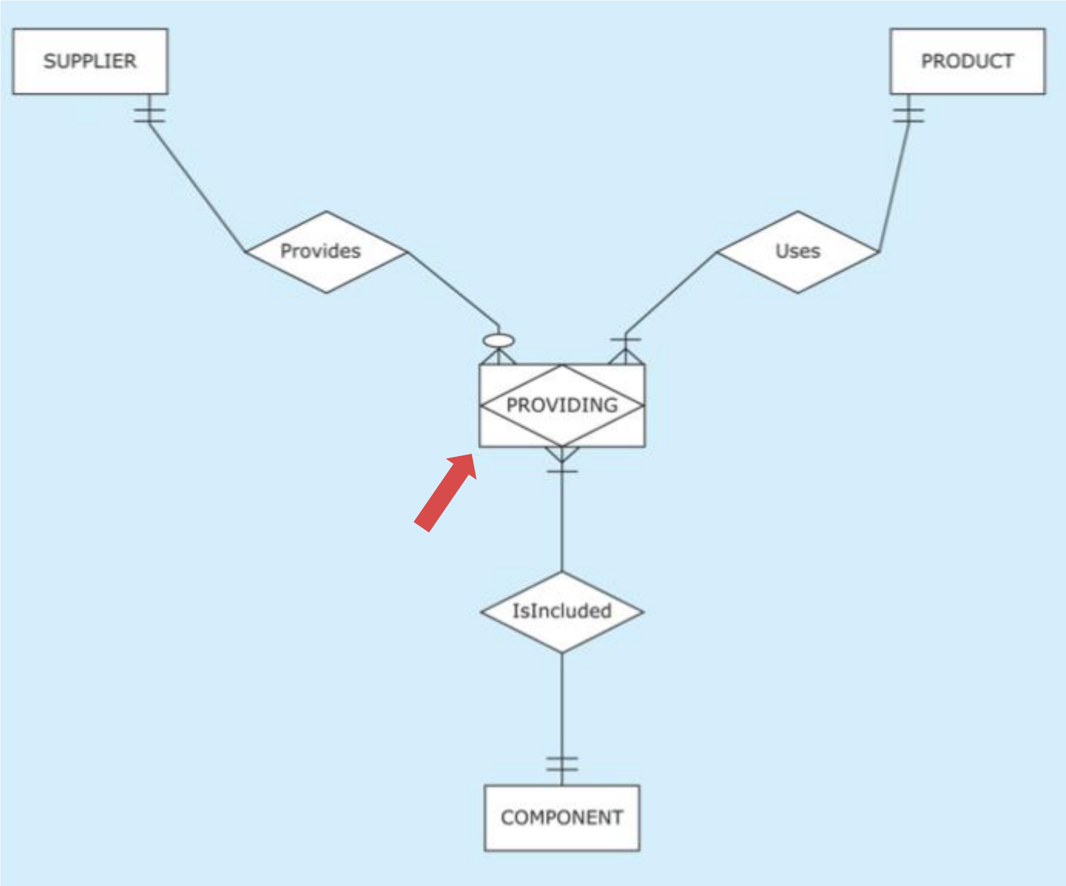


Associative entity:
Rectangle with a diamond inside of it.

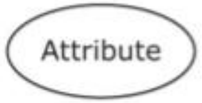
Ternary (and higher degree) relationships

- **Ternary relationship**: a relationship involving three entities
- Problem: where to label the necessary **cardinality constraints** unambiguously? Impossible.
- Solution: use an **associative entity**

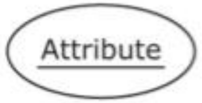




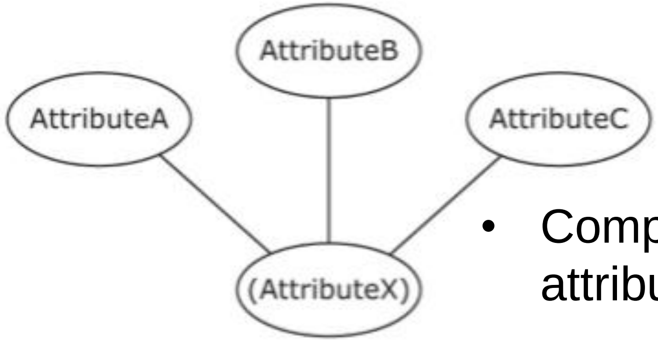
Summary



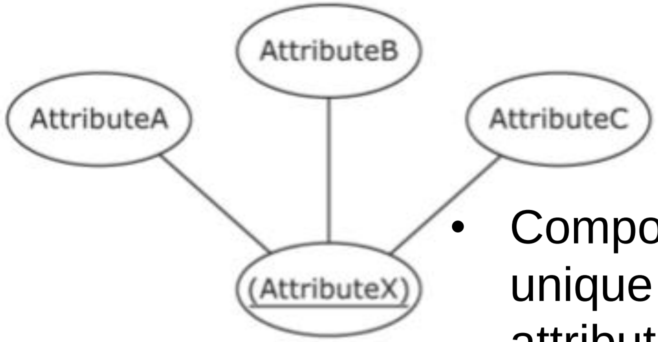
- Regular attribute



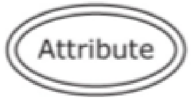
- Unique attribute



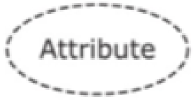
- Composite attribute



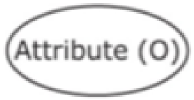
- Composite unique attribute



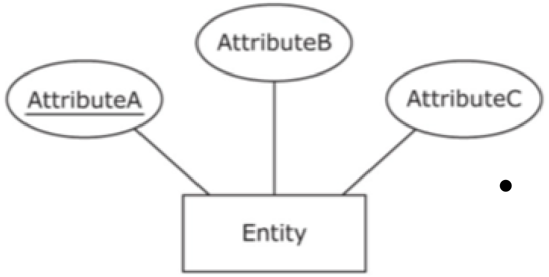
- Multivalued attribute



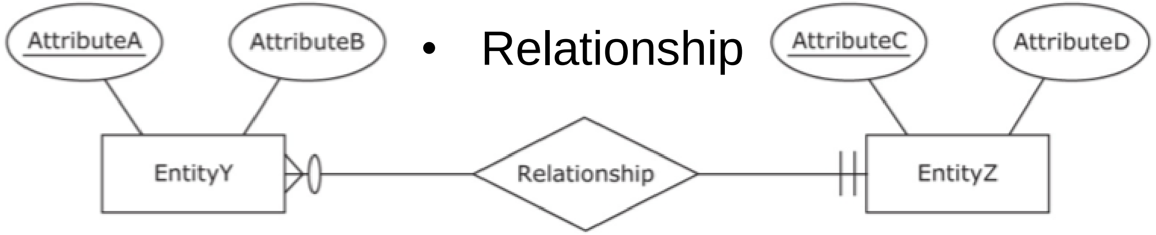
- Derived attribute



- Optional attribute



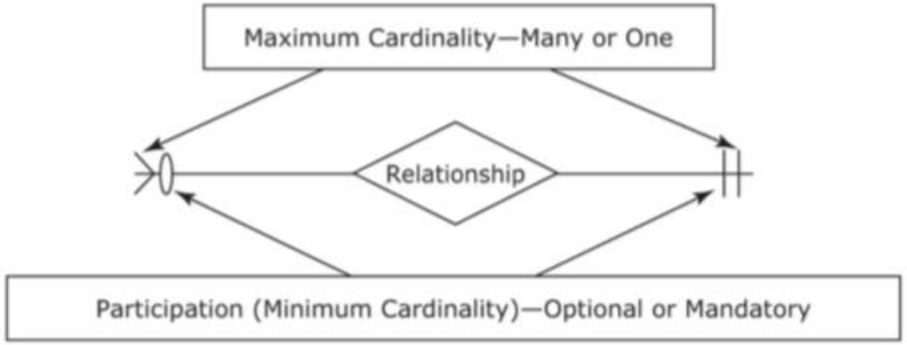
- Entity with attributes



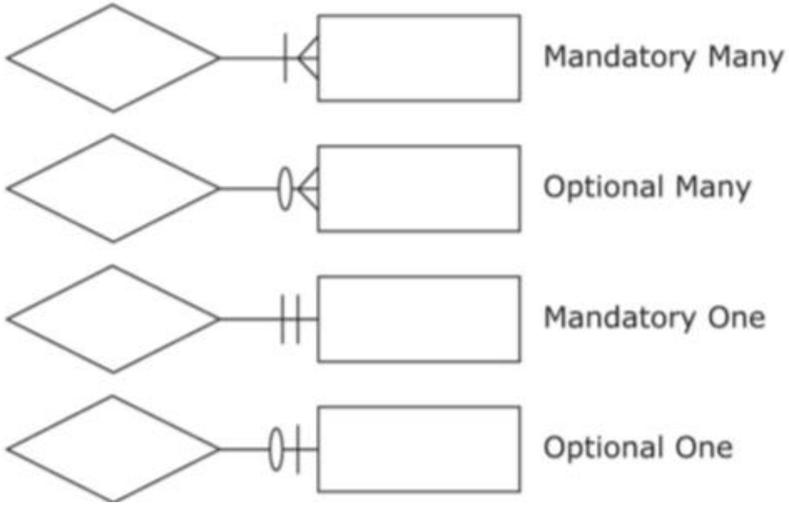
- Relationship

Summary

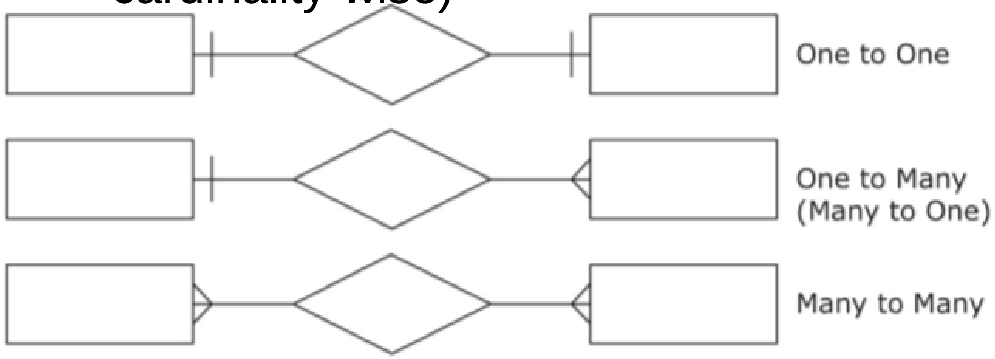
- Cardinality Constraints


Maximum Cardinality—Many or One
Relationship
Participation (Minimum Cardinality)—Optional or Mandatory

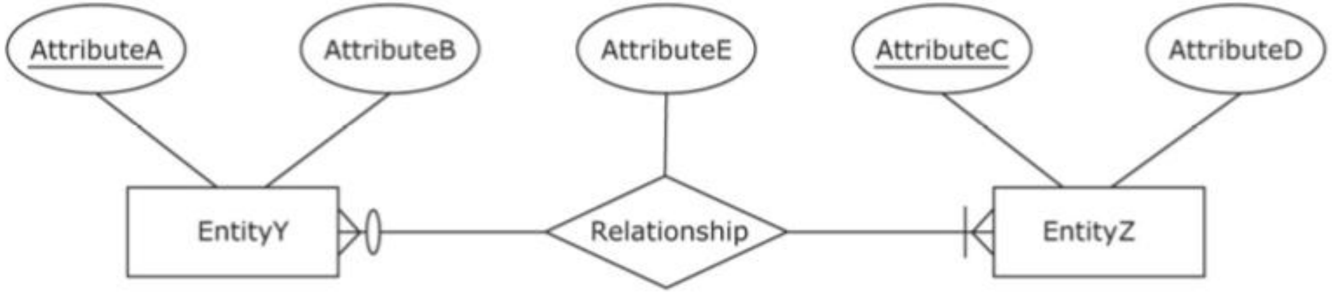
- Four possible cardinality constraints



- Three types of relationships (max cardinality-wise)

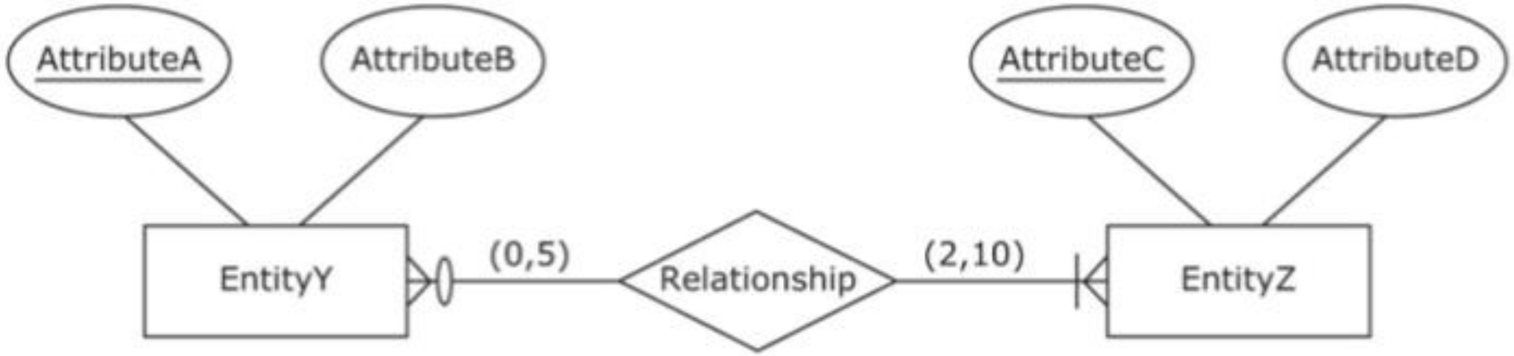


- Relationship with an attribute

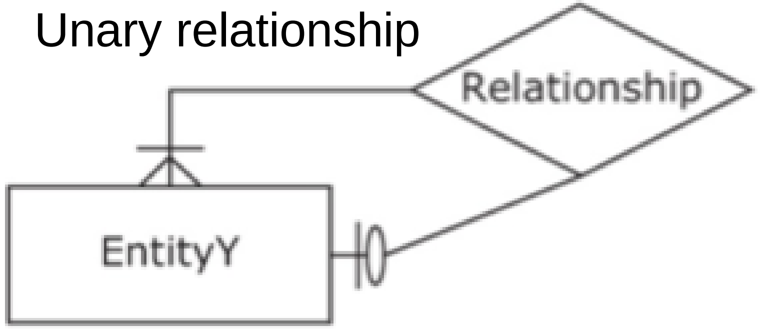


Summary

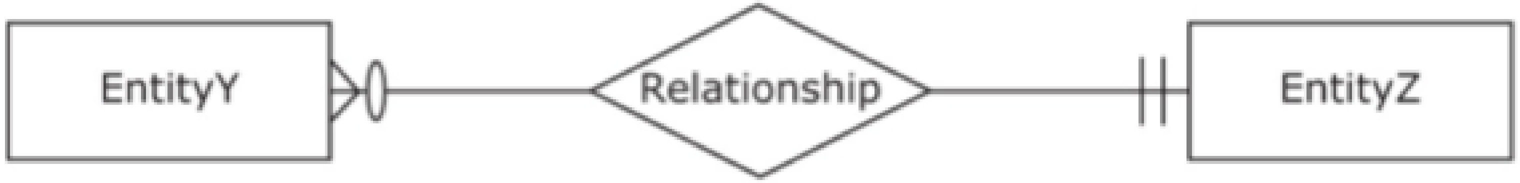
- Relationship with specific minimum and maximum cardinalities



- Unary relationship

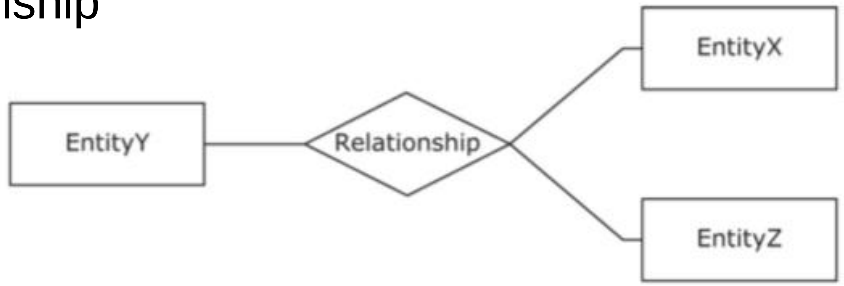


- Binary relationship

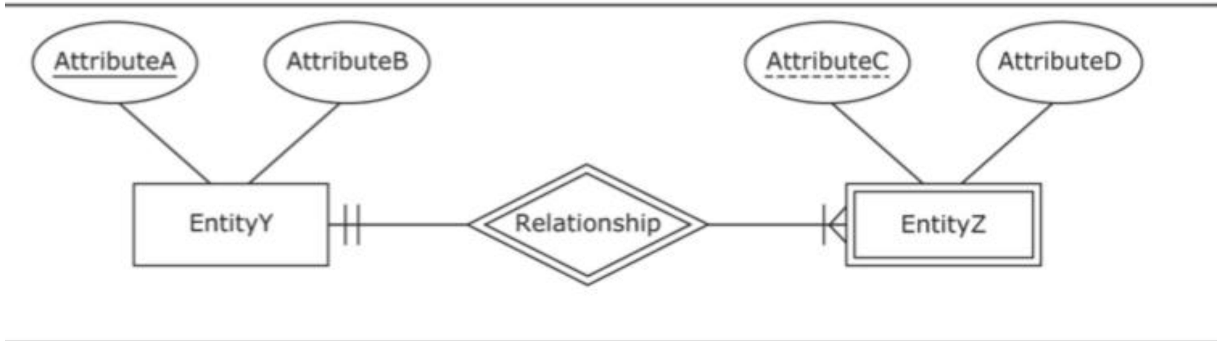


Summary

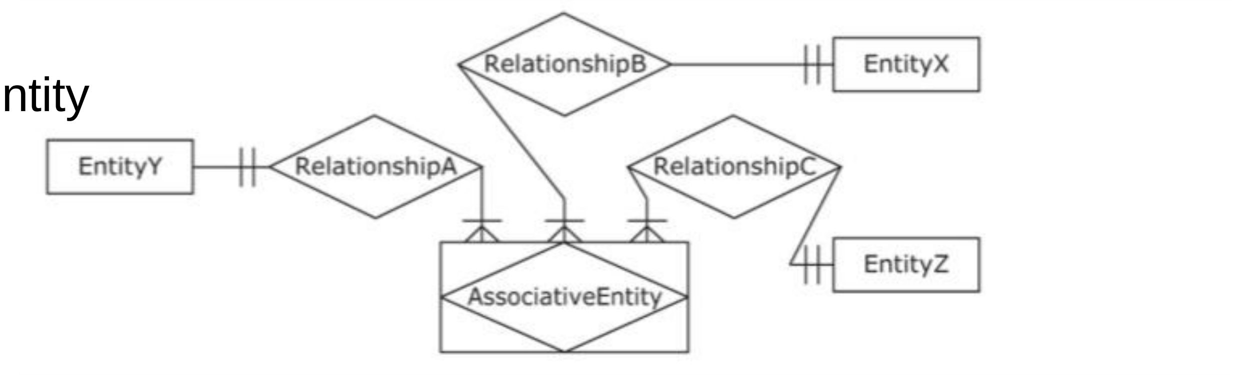
- Ternary relationship



- Weak entity



- Associative entity



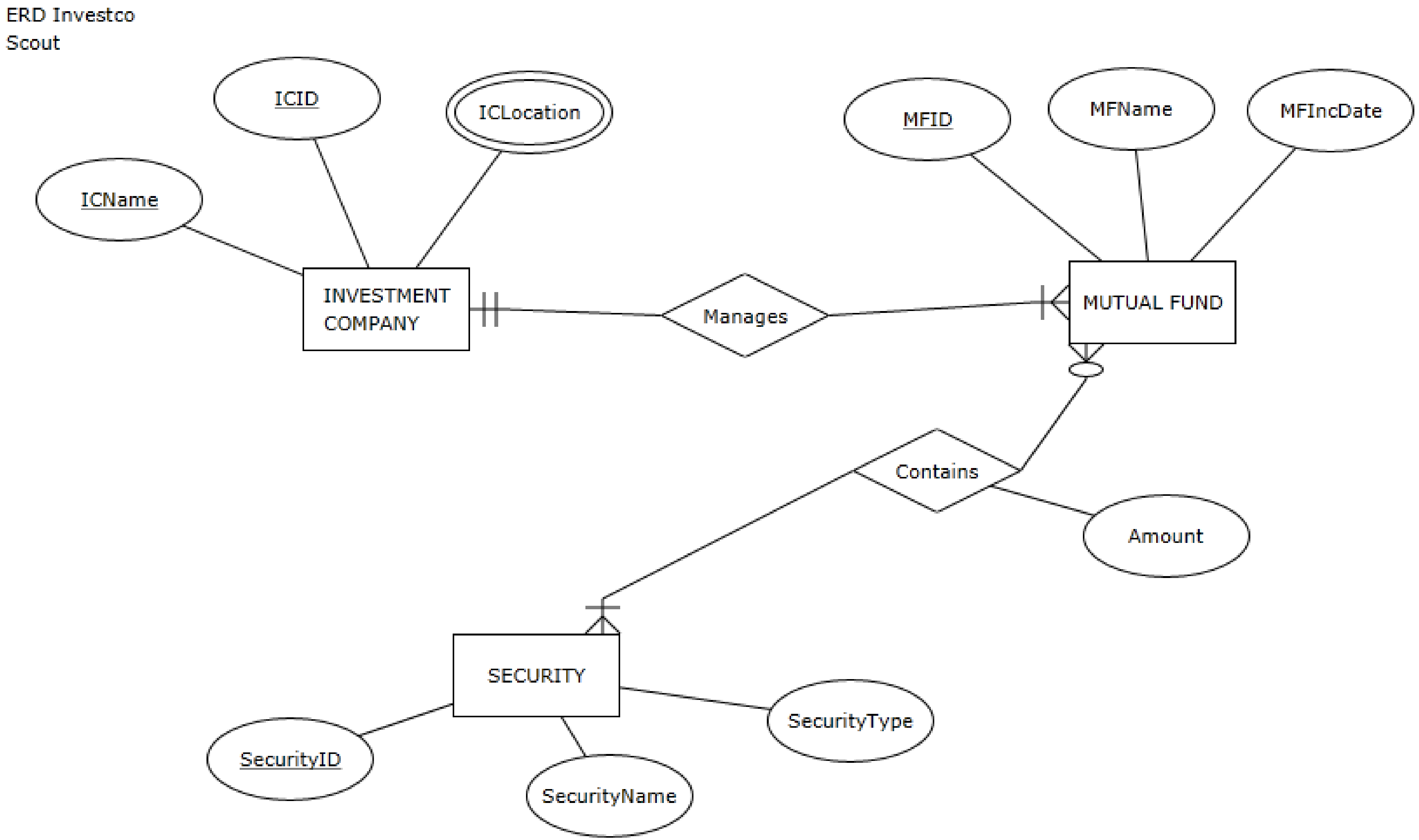
Optional Task – Draw the ERD

Task 6 : Investco Scout is an investment research company. Create the ERD for the Investco Scout Funds Database, based on the following requirements.

1. *The Investco Scount Founds Database will keep track of investment companies, the mutual funds they manage, and secruities contained in the mutual funds.*
2. *For each investment company, investco Scout will keep track of a unique investment company identifier and a unique investment company name as well as the names of multiple locations of the investment company.*
3. *For each mutual fund, Investco Scout will keep track of a unique mutual fund identifier as well as the mutual fund name and inception date.*
4. *For each security, Investco Scout will keep track of a unique security identifier as well as the security name and type*

1. *Investment companies can manage multiple mutual funds. Investco Scout does not keep track of investment companies that do not manage any mutual funds. A mutual fund is managed by one investment company.*
2. *A mutual fund contains one or many securities. A security can be included in many mutual funds. Investco Scout keeps track of securities that are not included in any mutual funds.*
3. *For each instance of a security included in a mutual fund, Investco Scout keeps track of the amount included.*

Optional Task – Draw the ERD


ERD Investco
Scout
ICID
ICLocation
ICName
INVESTMENT COMPANY
Manages
MFID
MFName
MFIncDate
MUTUAL FUND
Contains
Amount
SECURITY
SecurityID
SecurityName
SecurityType