UTS:ENGINEERING AND INFORMATION TECHNOLOGY

UTS
UNIVERSITY OF TECHNOLOGY, SYDNEY

lecture 3: Data Modeling Part II

Main reference:

Modern Database Management, 11th Edition
Chapter 2: Modeling Data in the Organization

Subject Coordinator and Instructor:
Dr. Danna (Fahimeh) Ramezani

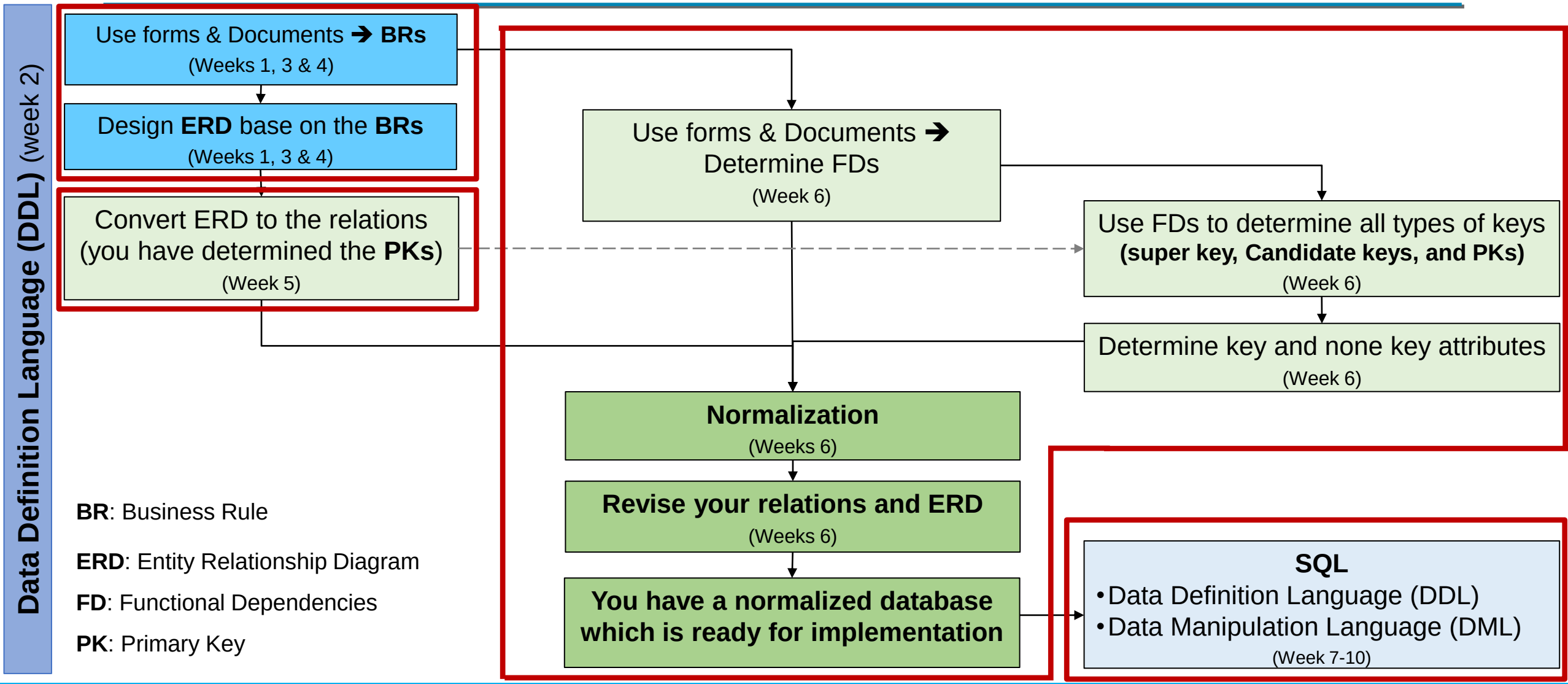
Participations and Discussions

The DF lecture are designed and elaborated to create a collaborative learning environment and engage students in concepts via class activities and discussions.

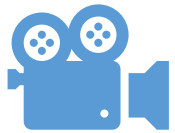
If you have any question and you don't want to share it in class, send it to us via Discussion Board on ED.

However, it is better to speak out in class ☺

Subject Flowchart



Subject Overview

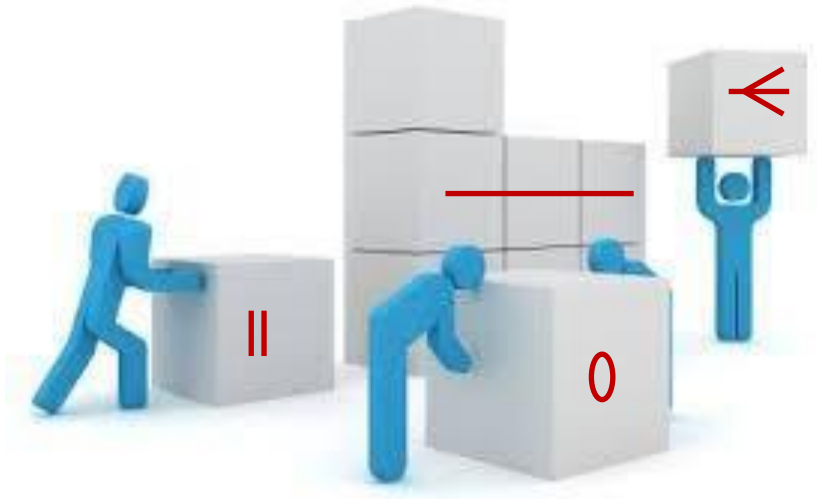


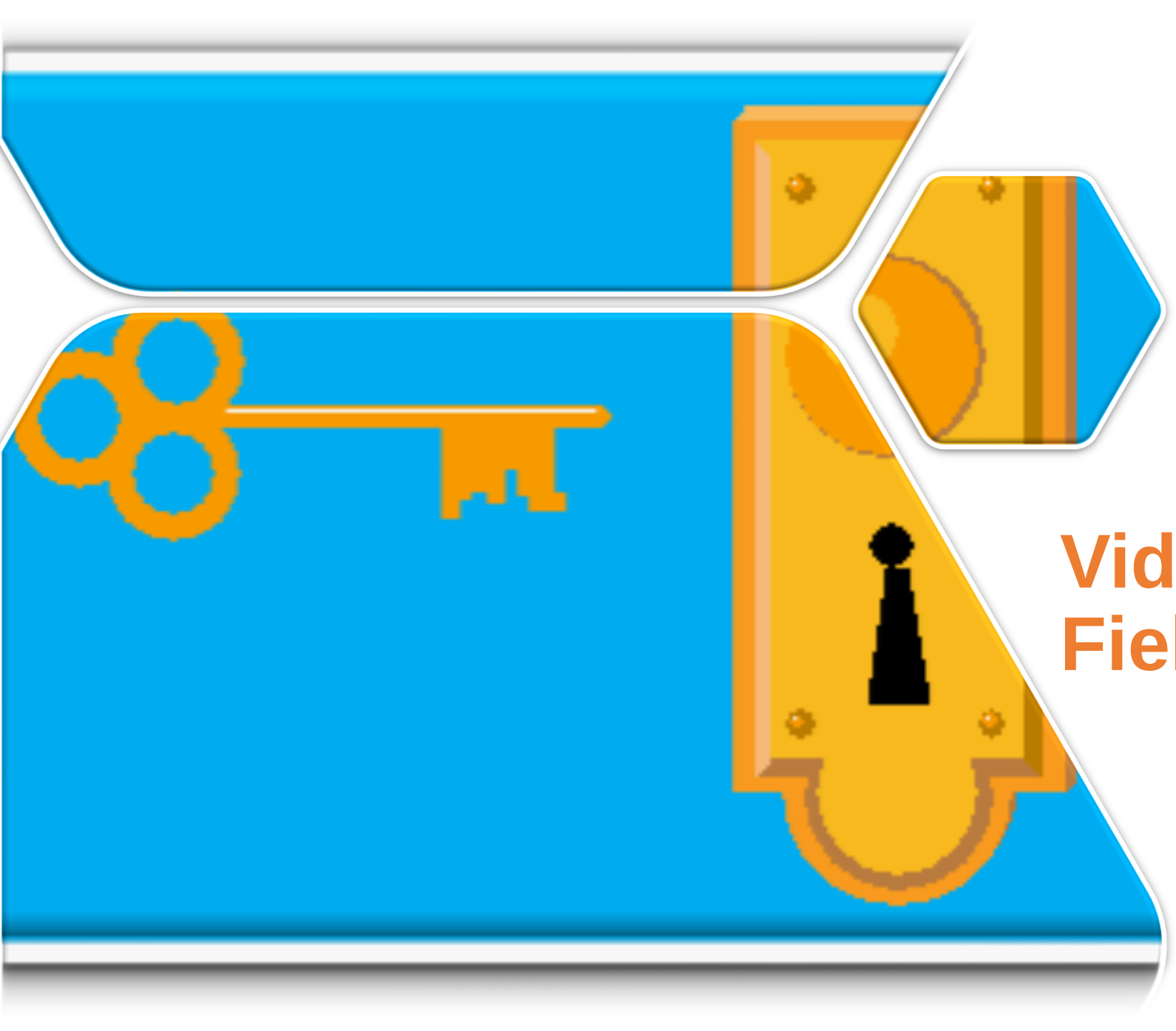
- **Design Entity Relationship Diagram (ERD)**
  - **Week 1: Data Modelling I (Conceptual Level):** Entity, Attributes, PK, FK, …
  - **Week 2: Data Definition Language (DDL):** Create tables, constraints, insert, …
  - **Week 3: Data Modelling II (Conceptual Level):** Associative, Weak, …
  - **Week 4: Data Modelling III (Conceptual Level):** Subtype/Supertype
  - **Week 5: Convert ERD to Relations (Logical Level)**
  - **Week 6: Functional Dependencies, and Normalization**
- **Data manipulation**
  - **Week 7: Simple Query**
  - **Week 8: Multiple Table Queries**
  - **Week 9: Subquery**
  - **Week 10: Correlated Subquery**

Objectives

- ✓ Business Rules
- ✓ E-R Model Constructs:
  - ✓ Entities
  - ✓ Attributes / Identifiers (Keys)

1. **Modeling Relationships:**
   - **1.1.** Relationship Types vs. Relationship Instances
   - **1.2.** Degree of Relationships
   - **1.3.** Cardinality of Relationships
   - **1.4.** Multiple Relationships Between Entities
   - **1.5.** Relationships with Attribute(s)
   - **1.6. Associative Entity– Combination of Relationship and Entity**
   - **1.7.** Multivalued Attributes Can be Represented as Relationships
   - **1.8. Weak and Strong Entities– Identifying Relationship**
2. **Notations**





Video 2.1: Key Fields

PK

Customer_ID	Customer_Name	Customer_Street	Customer_City	Customer_State	CustomerPostal_Code
1	Contemporary Casuals	1355 S Hines Blvd	Gainesville	FL	32601-2871
2	Value Furnitures	15145 S.W. 17th St.	Plano	TX	75094-7743
3	Home Furnishings	1900 Allard Ave	Albany	NY	12209-1125
4	Eastern Furniture	1925 Beltline Rd.	Carteret	NJ	07008-3188
5	Impressions	5585 Westcott Ct.	Sacramento	CA	94206-4056
6	Furniture Gallery	325 Flatiron Dr.	Boulder	CO	80514-4432
7	Eastern Furniture	Palace Ave	Farmington	NM	NULL

Primary keys (PKs) are **unique** identifiers of the relation (table). PK guarantees that all rows are unique.

Foreign keys (FKs) are attribute(s) in a relation (table) whose value match a primary key in another relation (table).

PK

Order_ID	Order_Date	Customer_ID
1001	8/09/2009	4
1002	4/10/2009	3
1003	19/07/2009	1
1004	1/11/2009	6
1005	28/07/2009	4
1006	27/08/2009	4

FK

Keys can be *simple* (a single field) or *composite* (more than one field).

Simple PK

PK

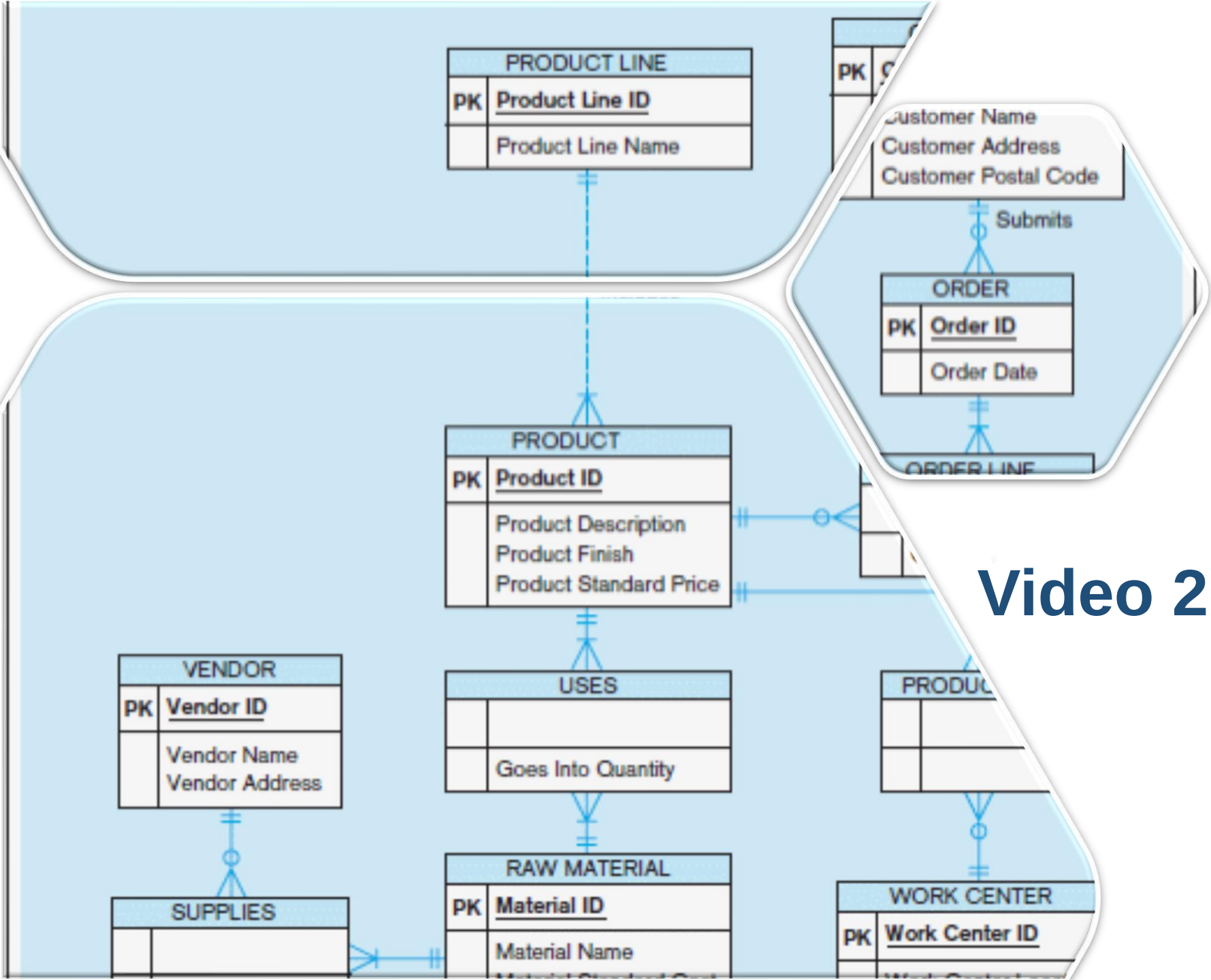
Order_ID	Order_Date	Customer_ID
1001	8/09/2009	4
1002	4/10/2009	3
1003	19/07/2009	1
1004	1/11/2009	6
1005	28/07/2009	4
1006	27/08/2009	4

Composite PK

Employee ID	Course ID	Date Completed
1234587	C68	1/1/2017
3459087	A57	1/2/2016
3459087	C68	5/10/2016

Explore In Class (EIC)

Go to → Lecture 2_EIC_2_PK and FK



Video 2.2:Relationships

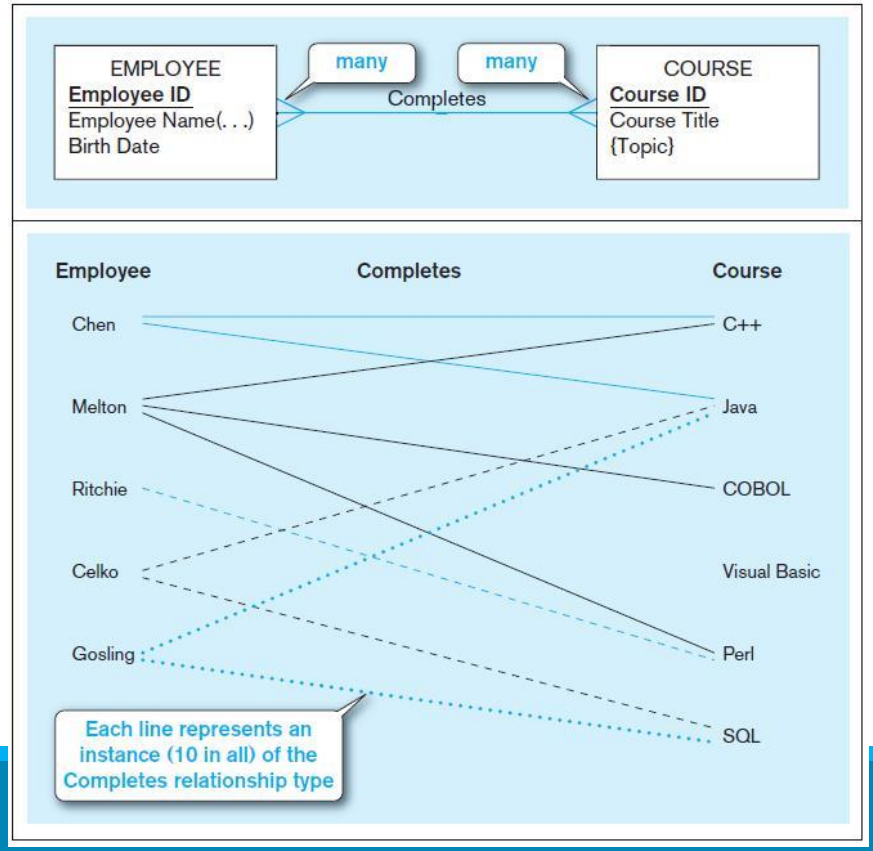
1.1. Relationship: Relationship types and instances (Figure 2-10)

a) The relationship type is **modeled** as lines between entity types (corresponds to primary key-foreign key equivalencies in related tables)

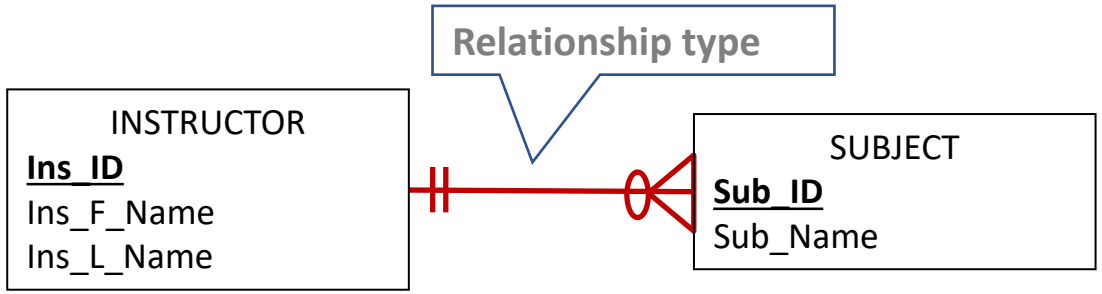
b) The relationship instance is between specific entity instances

a) Relationship type (Completes)

b) Relationship instances



1.1. Relationship: Relationship types and instances



Instructor

Ins_ID	Ins_F_Name	Ins_L_Name
12548	Fahimeh	Ramezani
45476	Luke	Brown
14475	Jack	Cooper

Subject FK

Sub_ID	Sub_Name	Ins_ID
31271	Database fundamentals	12548
31253	Database Programming	12548
41114	Software Development Studio	

Relationship instances

Note: any entity instances is a row in a table

Cardinality refers to the **relationship** between a instance of an entity (*a row of one table*) and instance(s) of another entity (*row(s) of another table*).

1.3. Relationship: Cardinality of Relationships

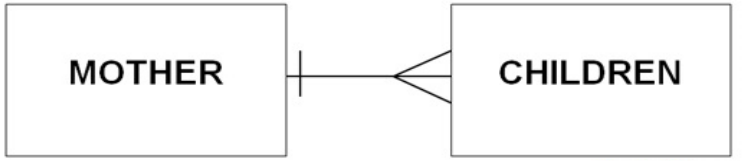
Cardinality refers to the **relationship** between a instance of an entity (*a row of one table*) and instance(s) of another entity (*row(s) of another table*).

➢ One-to-One

- Each entity instance in the relationship will have exactly **one related entity instance.**

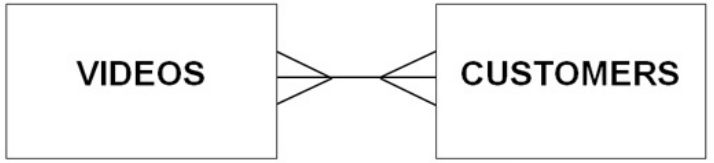
➢ One-to-Many

- An entity instance on one side of the relationship can have **many related entity instances**, but an entity on the other side will have **a maximum of one related entity instance.**



➢ Many-to-Many

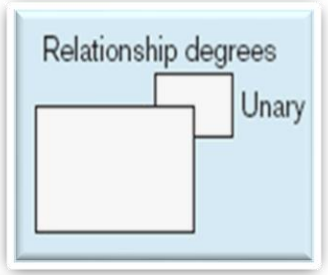
- Entities on both sides of the relationship can have **many related entity instances** on the other side.



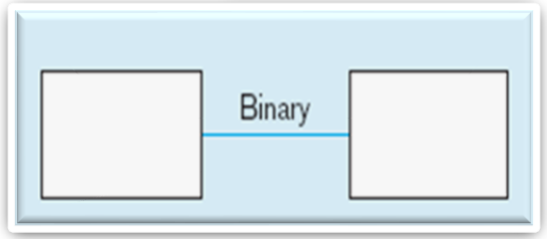
1.2. Relationship: Degree of Relationships

➢ **Degree of a relationship** is **the number of entity types** that participate in it.

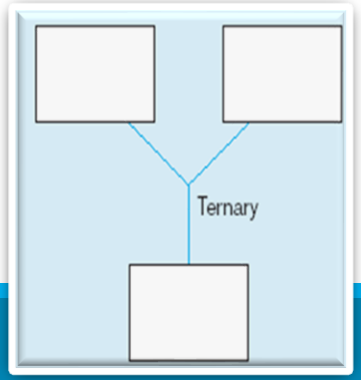
- **Unary Relationship** : A relationship between instances of the same entity. Also called a **recursive relationship**.



- **Binary Relationship :** A relationship between instances of two entities.

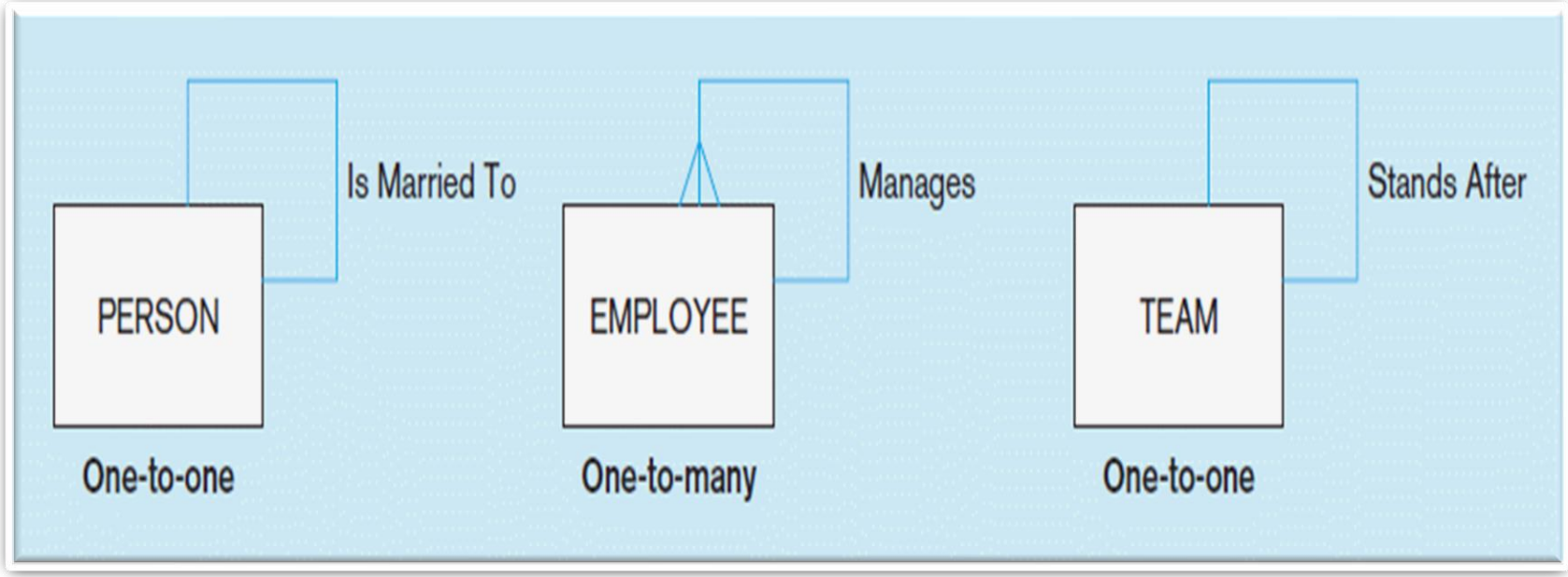


- **Ternary Relationship :** A relationship among instances of three entities.



1.2. Relationship: Examples of Relationships of Different Degrees (Figure 2-12)

a) Unary relationships



PK FK

Employee

Employee ID	Employee Name	Employment Date	Manager ID
1234587	Jack	1/1/2011	3459087
3459087	Michael	5/2/2010	
2387468	Sara	5/8/2010	3459087
5745321	Fahimeh	9/2/2012	1234587
8743836	Ricky	5/11/2010	3459087

Explore in Class 2.0:

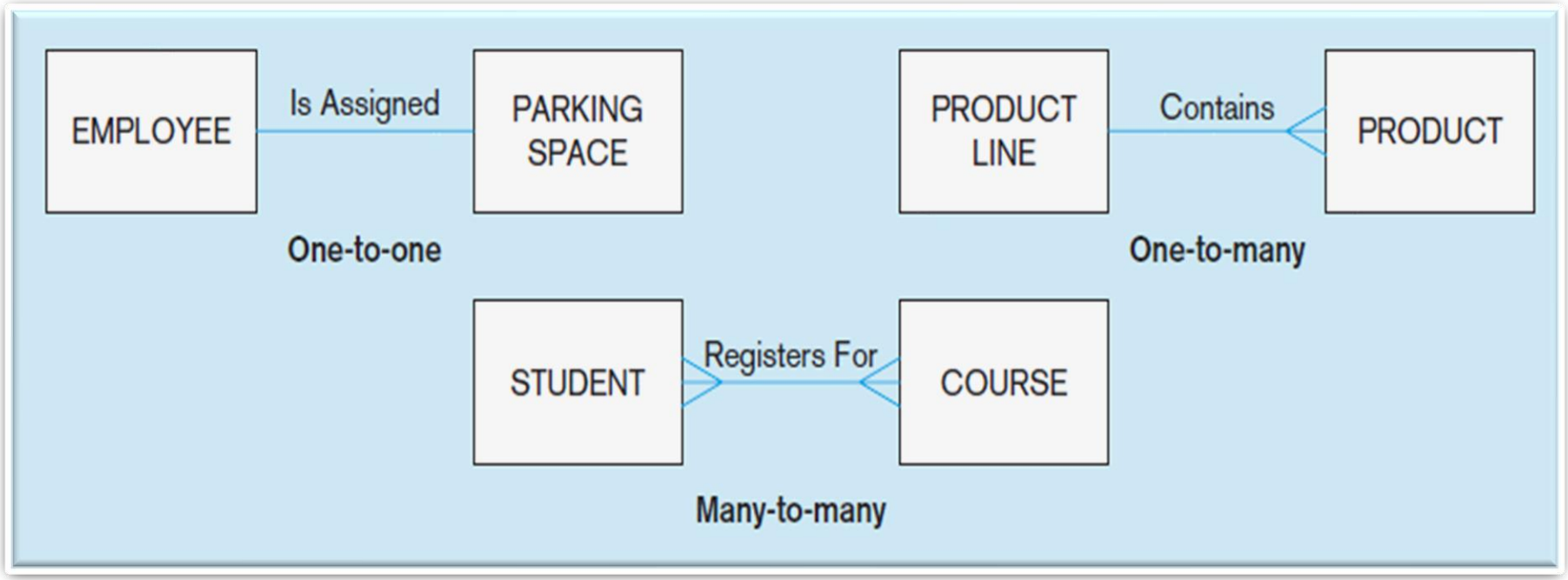
Who is Jack's manager?

Explore In Class (EIC)

Go to → Lecture 2_EIC_2.0_Recursive_Relationship

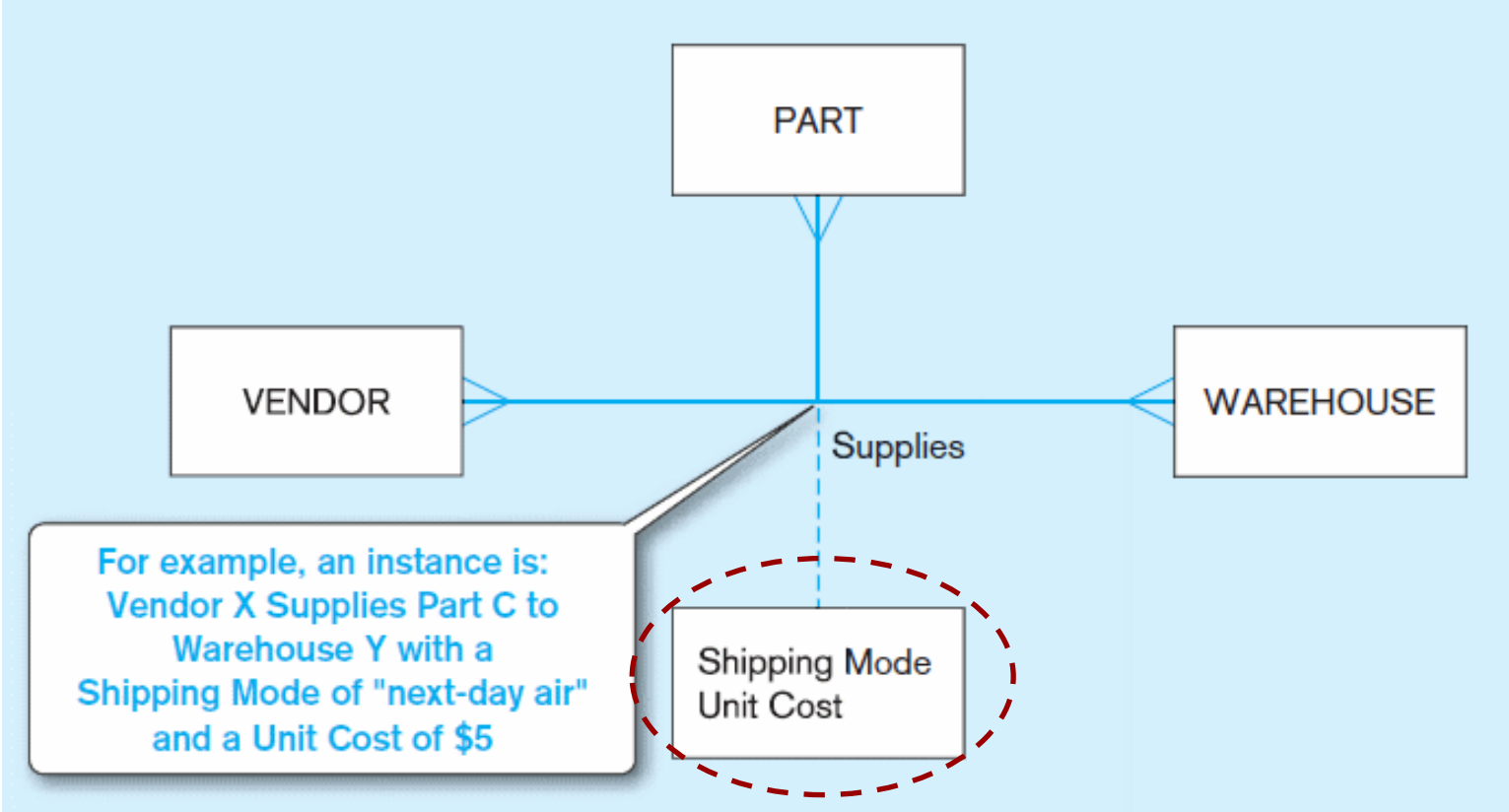
1.2. Relationship: Examples of Relationships of Different Degrees (Figure 2-12) (cont.)

b) Binary relationships



1.2. Relationship: Examples of Relationships of Different Degrees (Figure 2-12) (cont.)

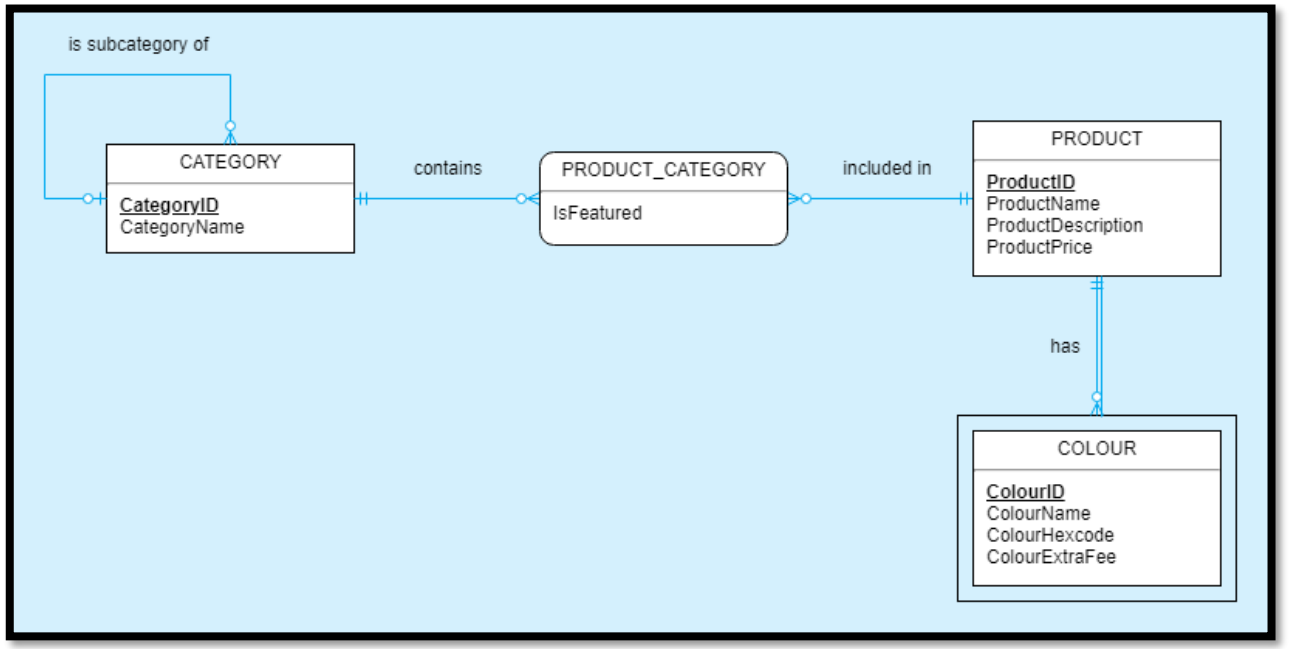
c) Ternary relationship



Note: a relationship can have attributes of its own

Class Activity 2.1: Determine the degree of relationships in the Clothing Shop ERD.

A clothing store has a variety products which they sell online. To make things easier for customers to navigate the online store, the products are separated into categories with popular products being featured in each. Each product can come in different colours which each include additional fees.



Solution: Degree of Relationships

Class Activity 2.2: Provide an example for One-to-One relationship (4 minutes).

Definition: Each entity instance in the relationship will have exactly **one related entity instance.**

Class Activity 2.2: Provide an example for One-to-Many relationship (2 minutes)

Definition: An entity instance on one side of the relationship can have **many related entity instances**, but an entity on the other side will have **a maximum of one related entity instance.**

Class Activity 2.2: Provide an example for Many-to-Many relationship.

Definition: Entities on both sides of the relationship can have **many related entity instances** on the other side.

Pause 

Video 2.3: The Rule

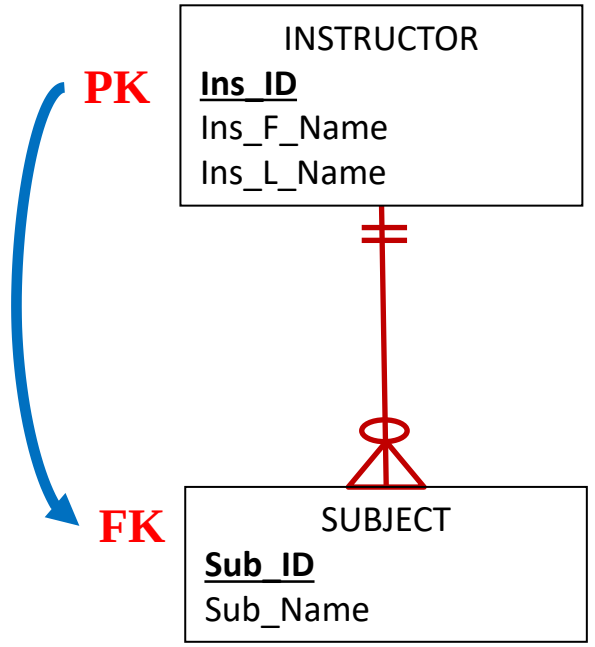
Explore in Class 2.1:

The Rule: in each **one-to-many** relationships,

PK of the entity on one side

is

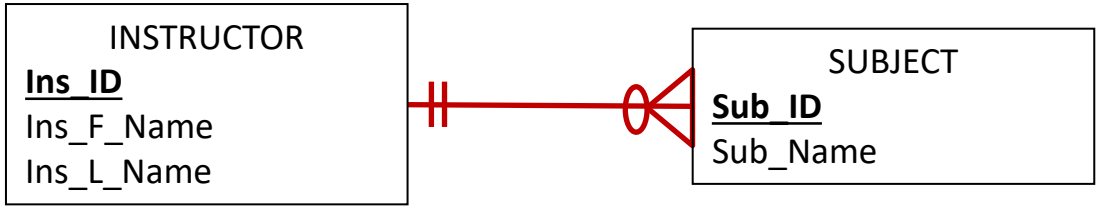
FK of the entity on the many side.



Explore in Class 2.1:

Why in each one-to-many relationships, PK of the entity on one side is FK of the entity on the many side?

BR: One instructor **can** teach **many** subjects, but one subject **needs to** be taught by **one** instructor.



Store data using the following designs to show Fahimeh coordinates "database Fundamentals" and "Database Programming" subjects.

incorrect choice of PK and FK.

Instructor

Ins_ID	Ins_F_Name	Ins_L_Name	Sub_ID
12548	Fahimeh	Ramezani	
45476	Luke	Brown	
14475	Jack	Cooper	

FK PK

Subject

Sub_ID	Sub_Name
31271	Database fundamentals
31253	Database Programming
41114	Software Development Studio

Correct choice of PK and FK.

PK **Instructor**

Ins_ID	Ins_F_Name	Ins_L_Name
12548	Fahimeh	Ramezani
45476	Luke	Brown
14475	Jack	Cooper

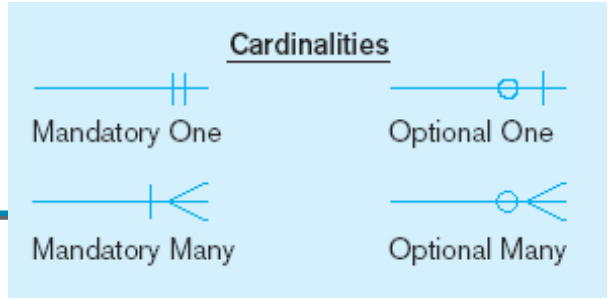
Subject FK

Sub_ID	Sub_Name	Ins_ID
31271	Database fundamentals	
31253	Database Programming	
41114	Software Development Studio	

Explore In Class (EIC)

Go to → Lecture 2_EIC_2.1_PK and FK Rule

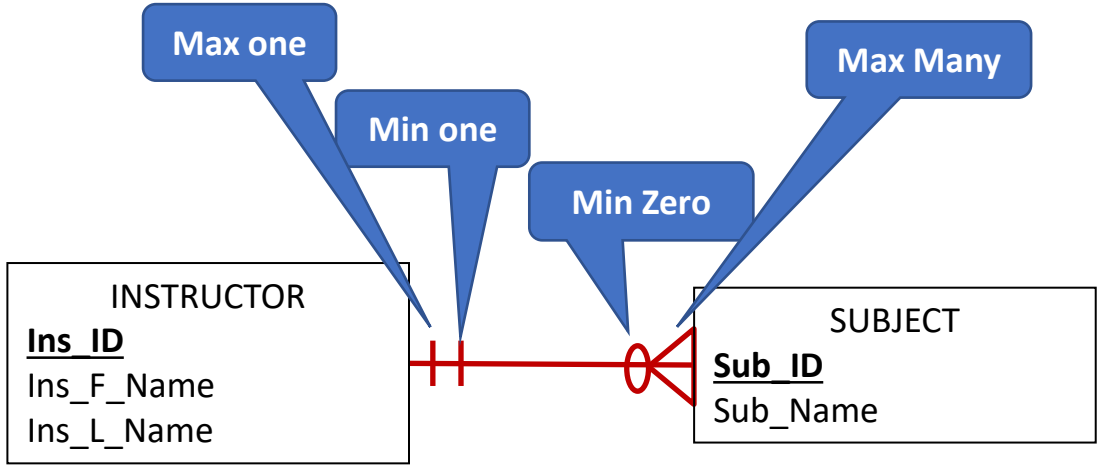
Video 2.4: Cardinality Constraints



- **Cardinality Constraints—** the number of instances of one entity that **can** (Optional) or **must** (mandatory) be associated with each instance of another entity

- **Minimum Cardinality**
  - If zero, then optional
  - If one or more, then mandatory

- **Maximum Cardinality**
  - The maximum number



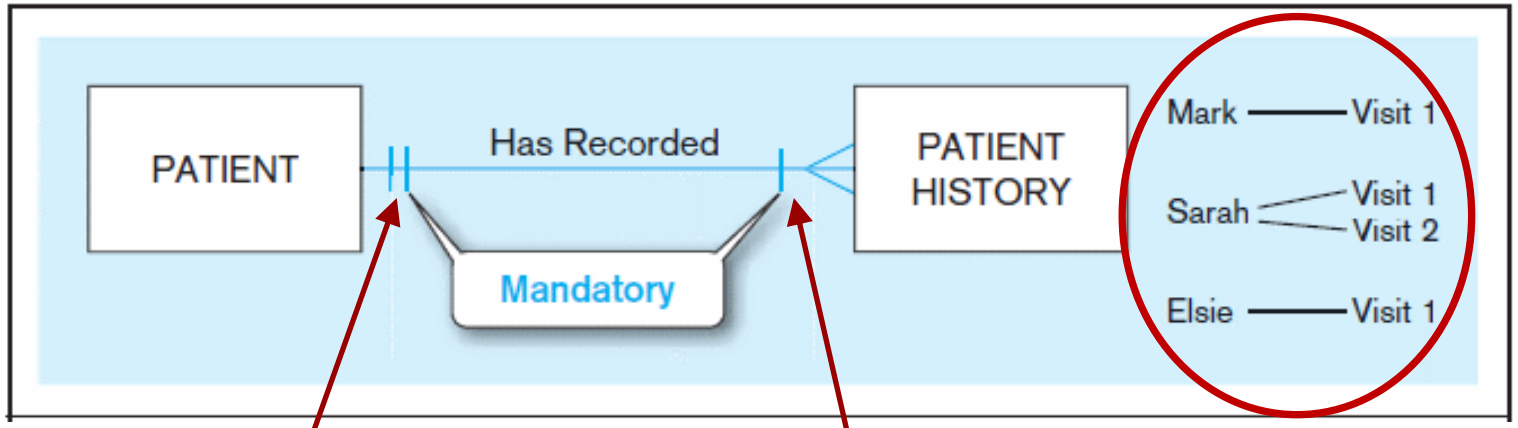
Business rules:

BR1: a subject **must** have one and only one instructor.

BR2: an instructor **can** teach any number of subjects.

1.3. Relationship: Examples of Cardinality Constraints (Figure 2-17)

a) Mandatory cardinalities



BR: A patient history is recorded for one and only one patient.

BR: A patient must have recorded at least one history, and can have many

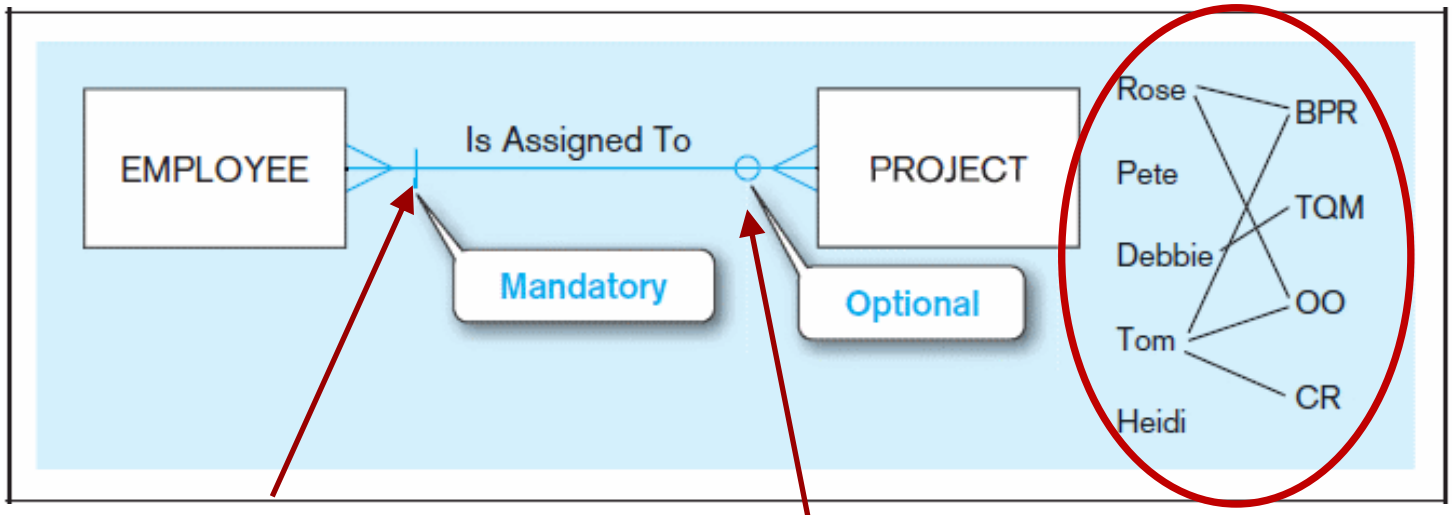
Patient

P_ID	P_F_Name	P_L_Name	…
123	Mark	Romanous	…
124	Sarah	Ramezani	…
125	Elsie	Cooper	…

Patient-History

P_ID	P_Chart_ID	Doctor_ID	Nurse_ID	…
123	1	x	y	…
124	1	m	y	…
124	2	m	z	…
125	1	w	c	…

b) One optional, one mandatory

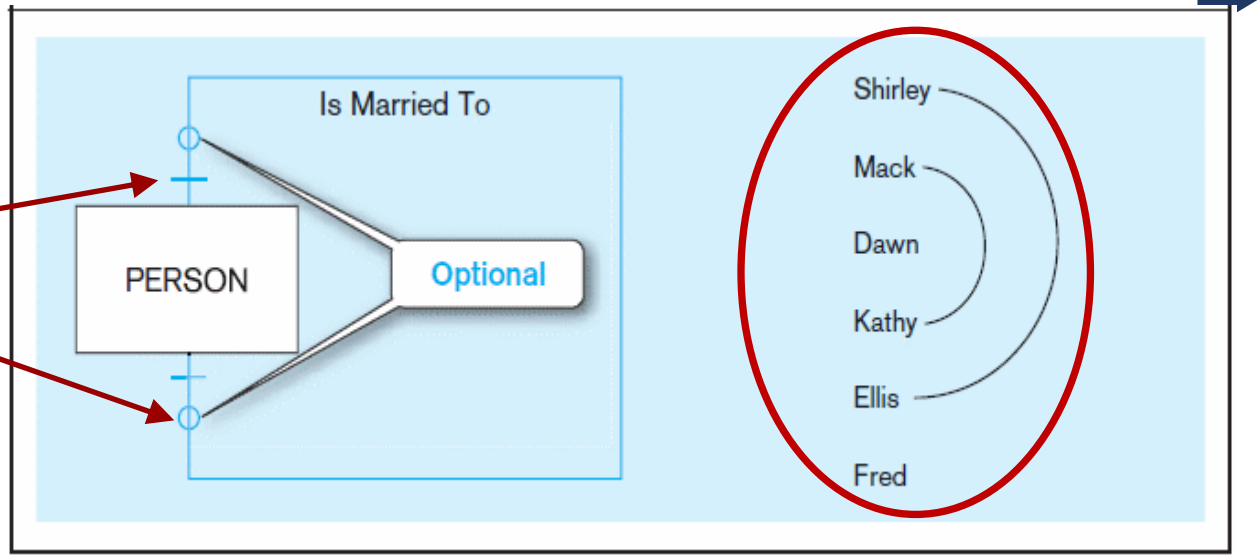


BR: A project must be assigned to at least one employee, and may be assigned to many

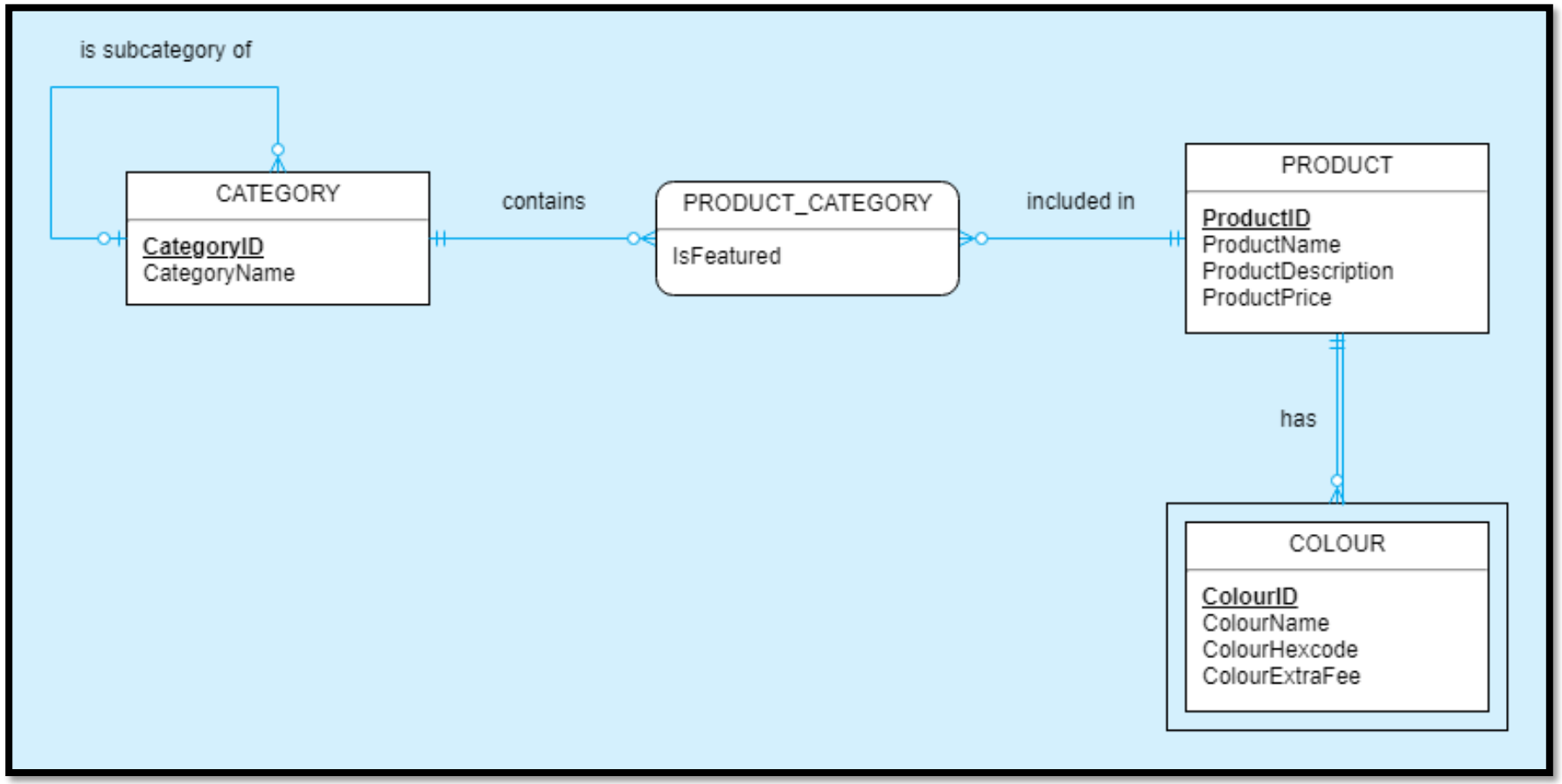
BR: An employee can be assigned to any number of projects, or may not be assigned to any at all

Class Activity 2.3: Draw the related table to this ERD (4 minutes)

BR: A person is married to at most one other person or may not be married at all.



Class Activity 2.4: Determine the cardinalities in the Clothing Shop ERD. (4 minutes)



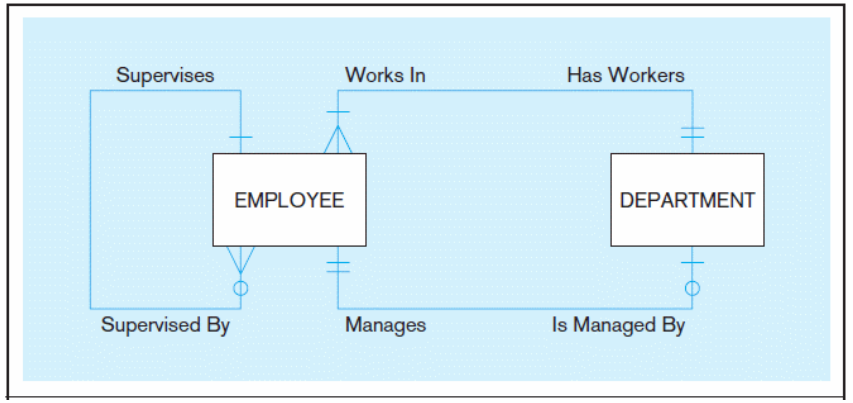
Solution- Cardinalities

1.4. Relationship: Multiple Relationships Between Entities

- Two entities can have more than one type of relationship between them (multiple relationships)

a) Figure 2-21: Employees and departments

Entities can be related to one another in more than one way

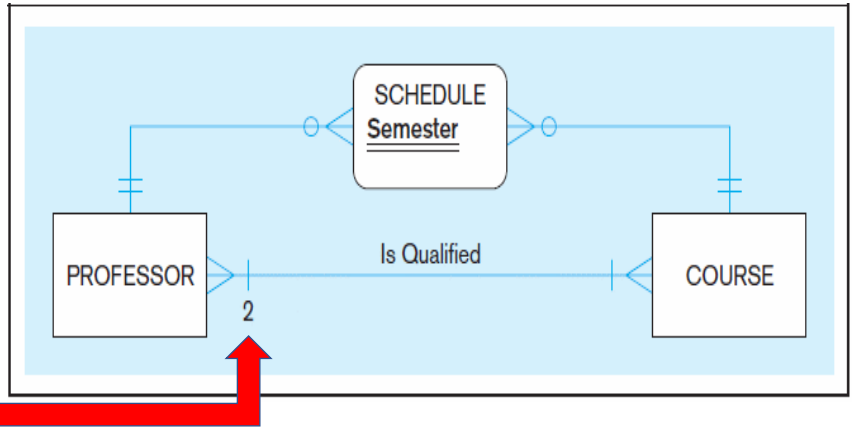


b) Figure 2-21: Professors and courses (fixed lower limit constraint)

Here, min cardinality constraint is **2**.

BR: At least two professors must be qualified to teach each course.

BR: Each professor must be qualified to teach at least one course.



1.5. Relationship: Relationships Can Have Attributes

- Attributes of a relationship describe features pertaining to the **association between** the entities in the relationship

Here, the **date completed** attribute pertains specifically to the employee's completion of a course … it is an attribute of the *relationship*.

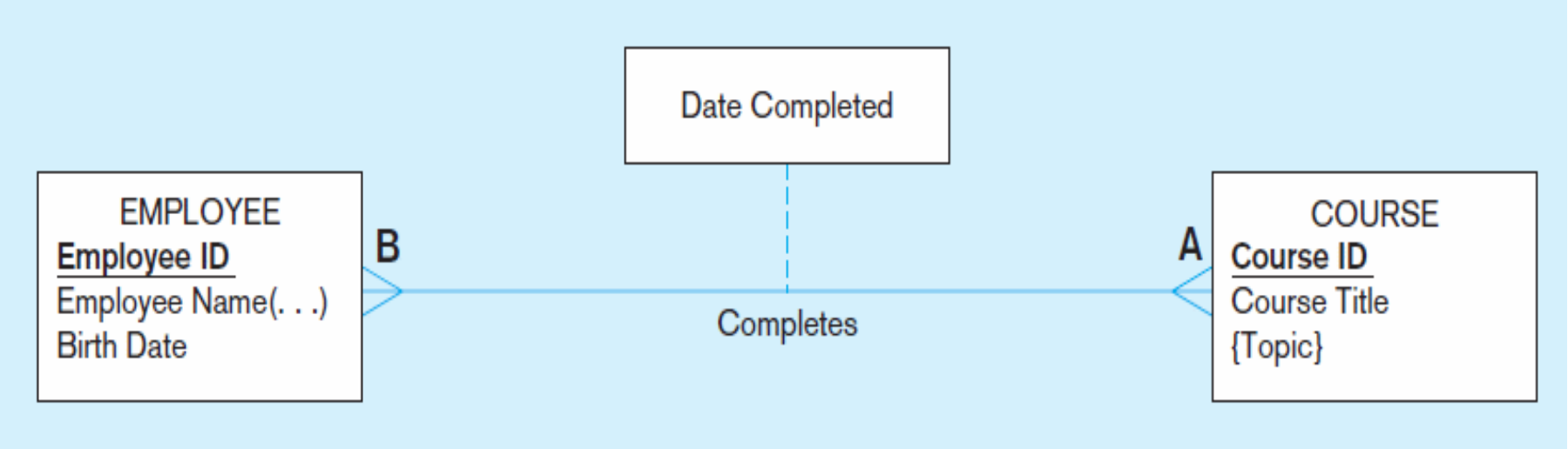


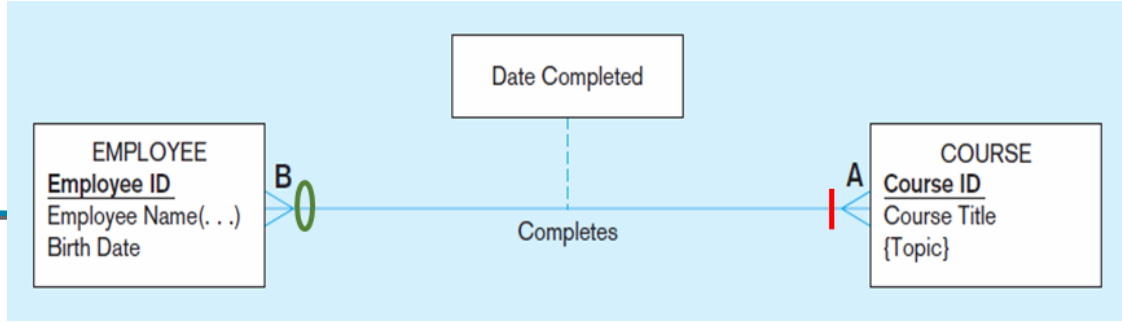
Figure 2-11a A binary relationship with an attribute

Video 2.5: Associative Entities

An associative entity **links** entities together

A Many-To-Many Relationship

BR1: Each employee should complete at least one course.
BR2: Any course can be completed by any number of employees and can be taken by none.
BR3: The completion date of the courses need to be stored.



Business data that need to be stored in the database:
- Jack has completed course ID C68
- Danna has completed course IDs A57 and C68

Question: How to store these data and show which employee has completed which courses?

PK FK PK

Employee

Employee ID	Employee Name	Birth Date	Course ID	Date Completed
1234587	Jack	1/1/1960	C68	1/1/2017
3459087	Danna	5/2/1985	A57 , C68	1/2/2016 , 5/10/2016

Course

Course ID	Course Title	Topic
A57	Knowledge Management	….
C68	Project Management	….

PK PK FK

Employee

Employee ID	Employee Name	Birth Date
1234587	Jack	1/1/1960
3459087	Danna	5/2/1985

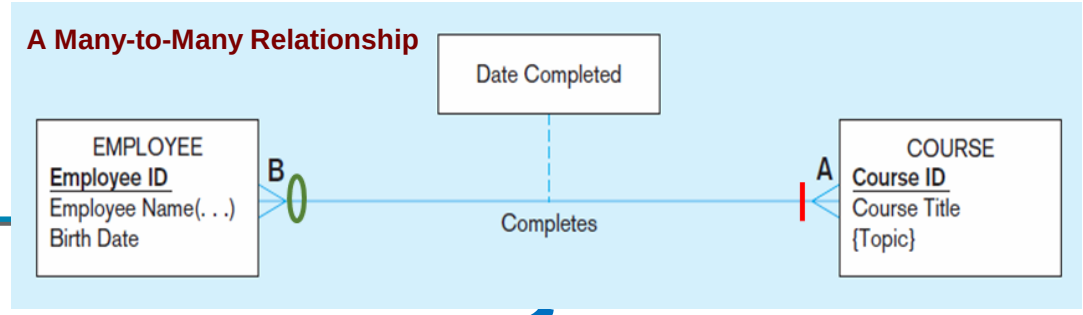
Course

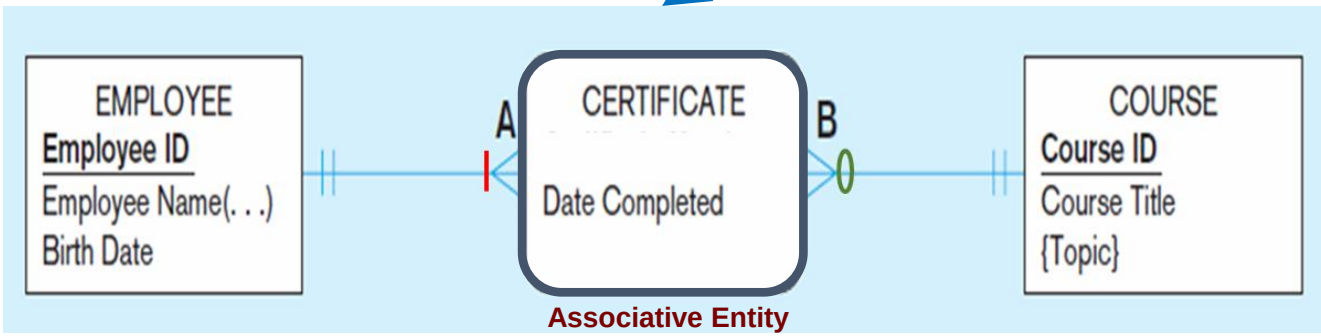
Course ID	Course Title	Topic	Employee ID	Date Completed
A57	Knowledge Management	….	3459087	1/2/2016
C68	Project Management	….	3459087 , 1234587	5/10/2016 , 1/1/2017

An associative entity (Figure 2-11b)

BR1: Each employee should complete at least one course.
BR2: Any course can be completed by any number of employees and can be taken by none.
BR3: The completion date of the courses need to be stored.

Question: Which Employee has completed which courses?





Remember the Rule: in each one-to-many relationships, PK of the entity on one side is FK of the entity on the many side.

PK

PK

FK FK

PK

Employee

Employee ID	Employee Name	Birth Date
1234587	Jack	1/1/1960
3459087	Danna	5/2/1985

Certificate

Employee ID	Course ID	Date Completed
1234587	C68	1/1/2017
3459087	A57	1/2/2016
3459087	C68	5/10/2016

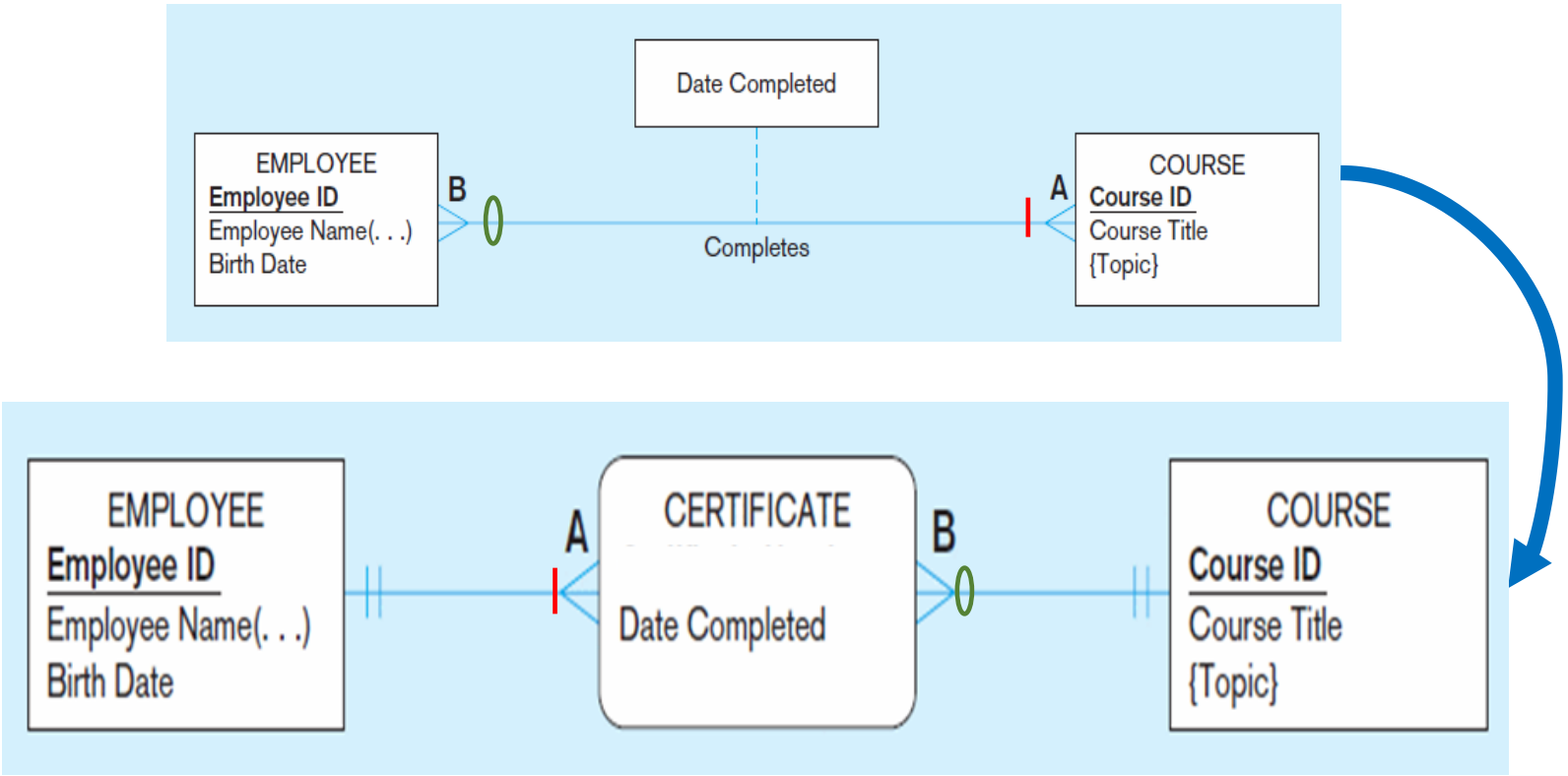
Course

Course ID	Course Title	Topic
A57	Knowledge Management	….
C68	Project Management	….

1.6. Cardinality constraints of an associative entity (Figure 2-11b)

BR1: Each employee should complete at least one course.
BR2: Any course can be completed by any number of employees and can be taken by none.
BR3: The completion date of the courses need to be stored.

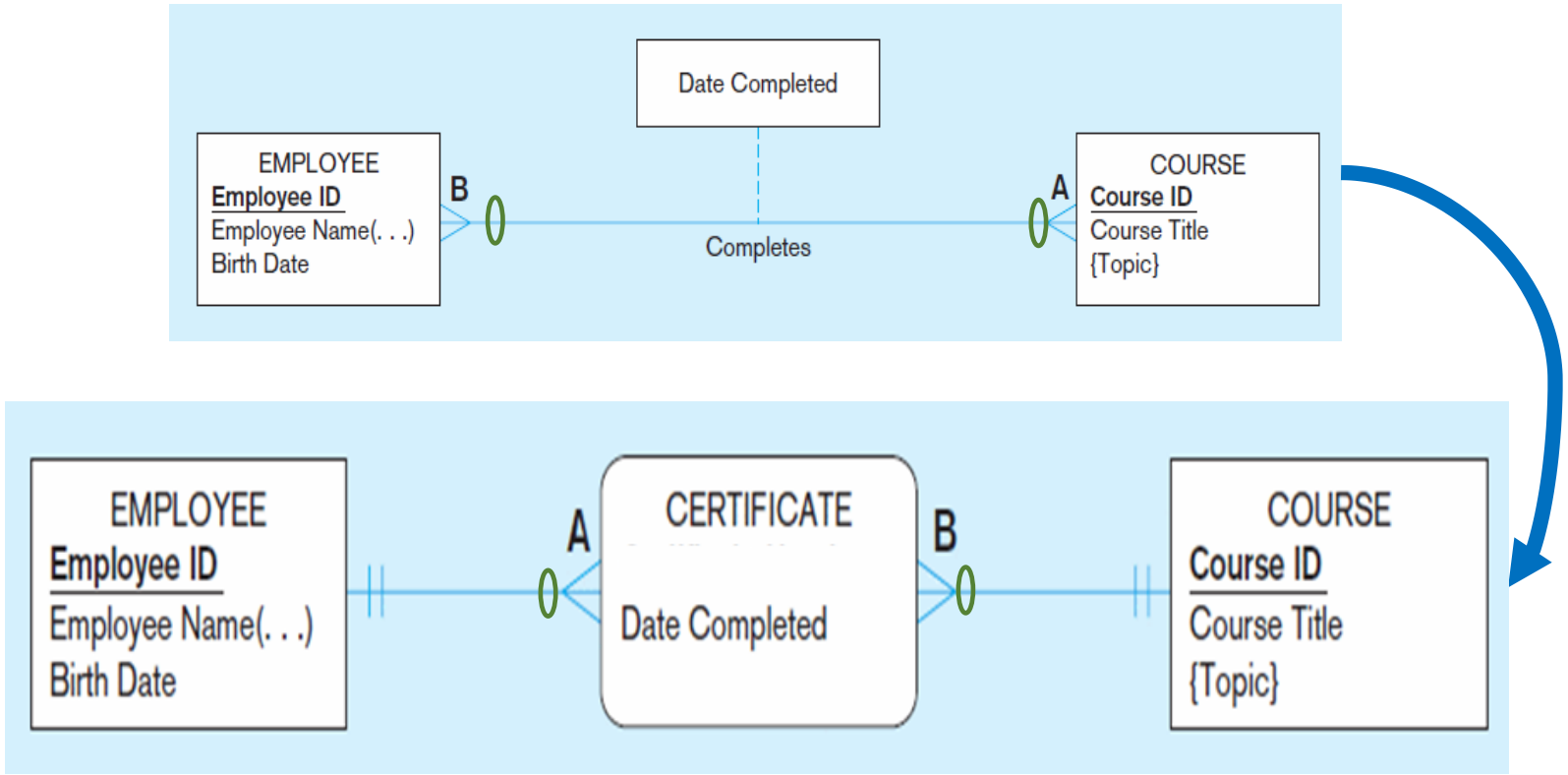
- We need to create a new relation for every Many-to-Many relationship.
- How to determine the cardinalities for the new relation (the associative entity)?



1.6. Cardinality constraints of an associative entity (Figure 2-11b)

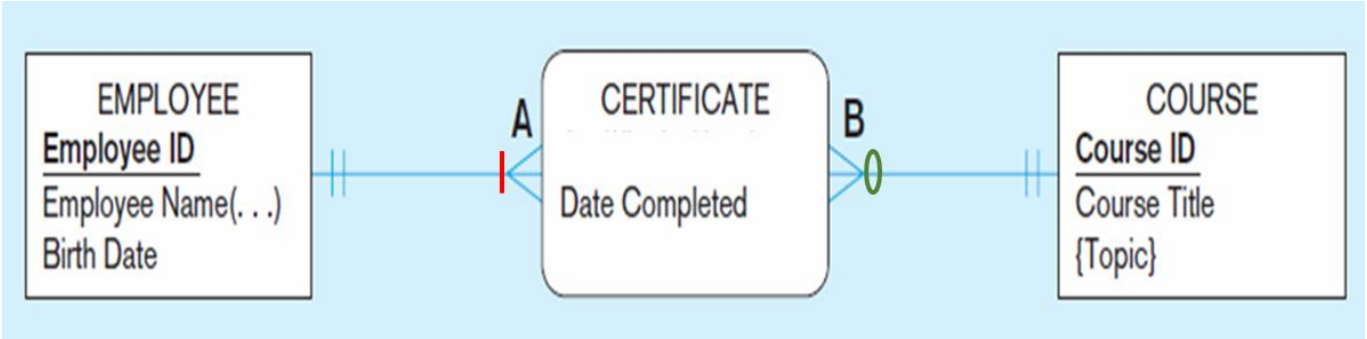
BR1: Each employee can complete any number of courses and can complete none.
BR2: Any course can be completed by any number of employees and can be taken by none.
BR3: The completion date of the courses need to be stored.

- We need to create a new relation for every Many-to-Many relationship.
- How to determine the cardinalities for the new relation (the associative entity)?

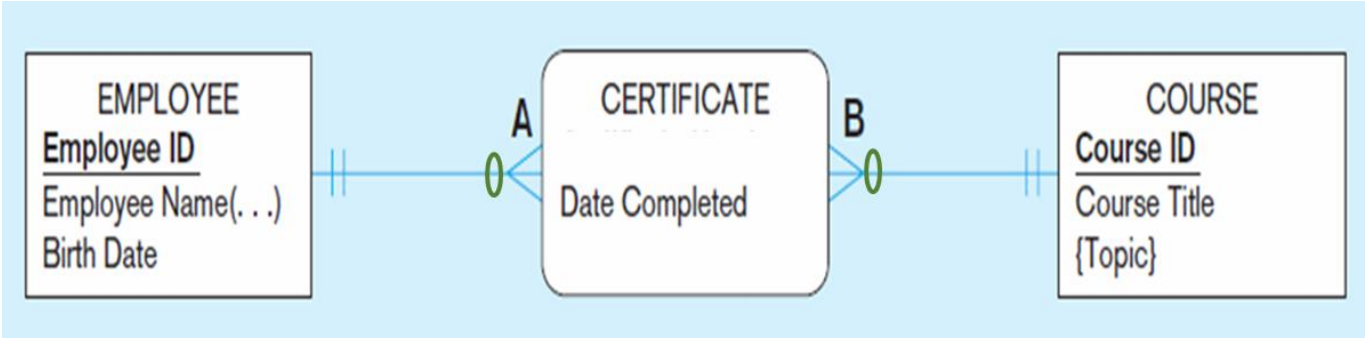


See the difference…

BR1: Each employee should complete at least one course.
BR2: Any course can be completed by any number of employees and can be taken by none.
BR3: The completion date of the courses need to be stored.



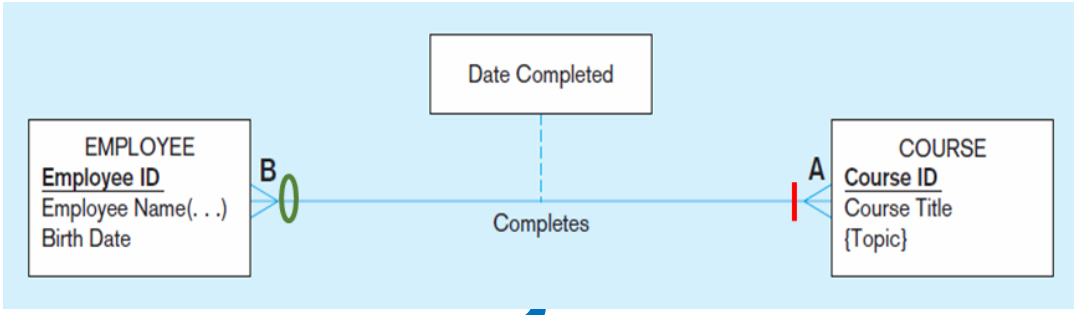
BR1: Each employee can complete any number of courses and can complete none.
BR2: Any course can be completed by any number of employees and can be taken by none.
BR3: The completion date of the courses need to be stored.

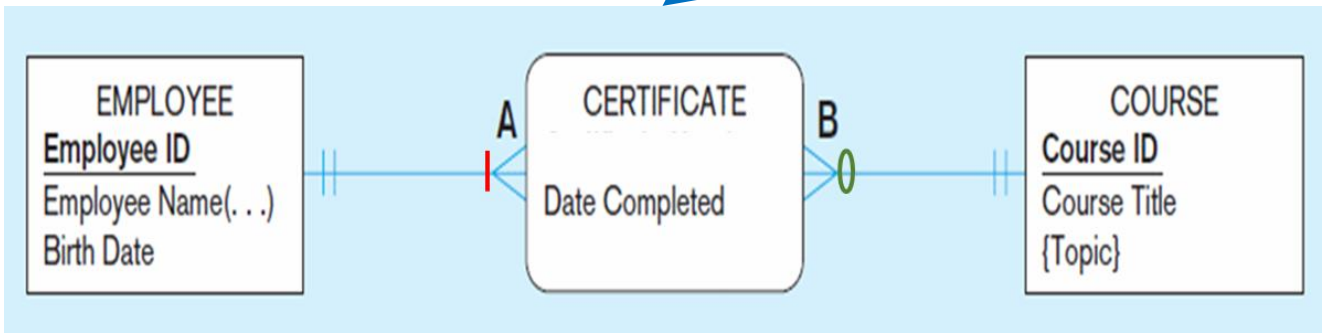


1.6. The PK of an Associative entity

Explore in Class 2.2:

Let's jump to Ed and write some query to learn more about the PK of an associative entities …





PK

Employee

Employee ID	Employee Name	Birth Date
1234587	Jack	1/1/1960
3459087	Danna	5/2/1985

PK

FK FK

Certificate

Employee ID	Course ID	Date Completed
1234587	C68	1/1/2017
3459087	A57	1/2/2016
3459087	C68	5/10/2016

PK

Course

Course ID	Course Title	Topic
A57	Knowledge Management	….
C68	Project Management	….

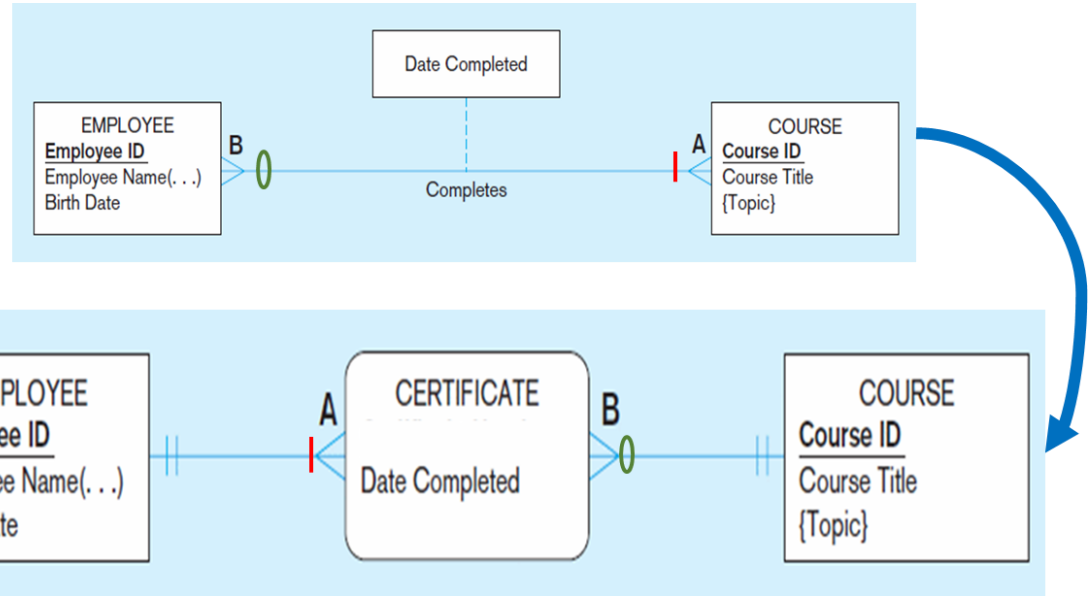
Remember the Rule: in each one-to-many relationships, PK of the entity on one side is FK of the entity on the many side

Explore In Class (EIC)

Go to → Lecture 2_EIC_2.2_Associative_Entity

1.6. An associative entity (Figure 2-11b): Notes

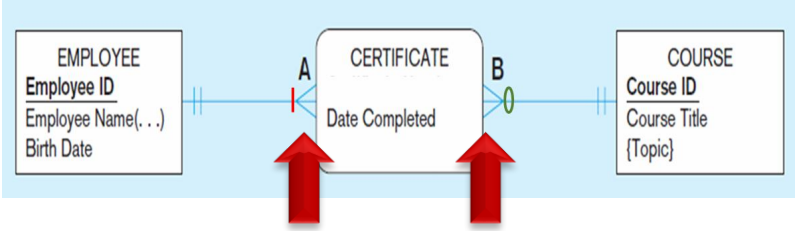
The many-to-many relationship between entities in the first figure is replaced by two one-to-many relationships with the associative entity (CERTIFICATE).



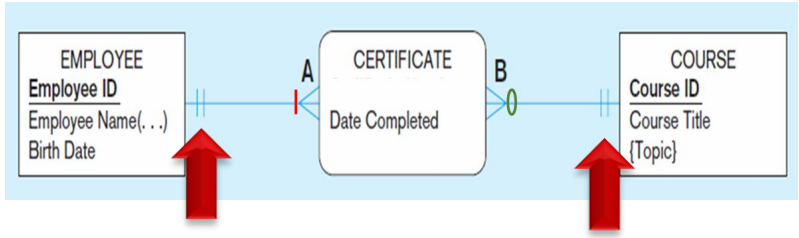
- An associative entity is like a relationship with an attribute, but it is also considered to be an entity in its own right.
  - An associative Entity is an entity and has attributes.
  - An associative Entity is a relationship that links entities together.
- Note that we **do not show FKs** in the ERD.

1.6. Associative Entities (or a linking table): Notes

- Whenever we have a Many-to-Many or a Ternary relationship, we need to convert the relationship to an associative entity.
- The relationship between **the related entities** and **the associative entity** should have maximum many cardinality.

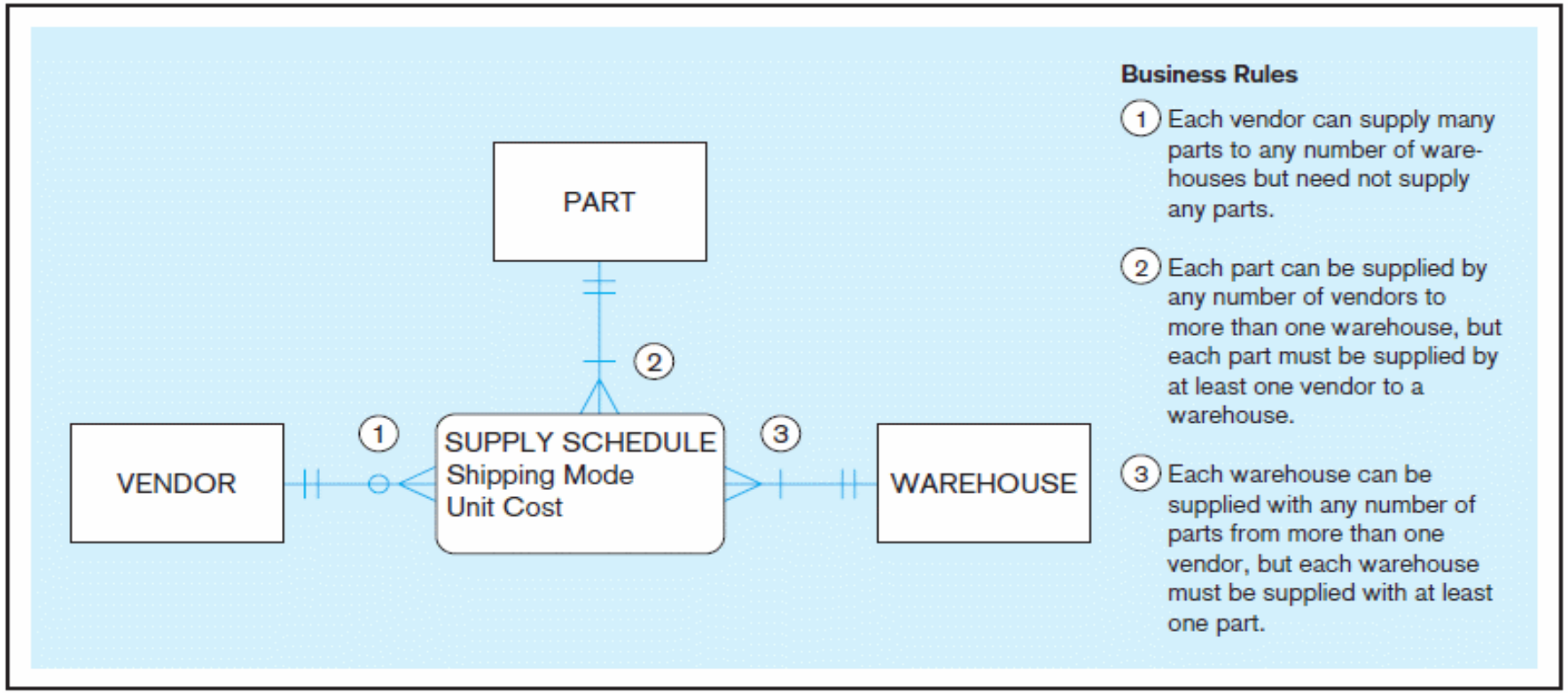


- The relationship between **the associative entity** and **the related entities** should be mandatory one (minimum one and maximum one cardinality).



- The associative entity has a composite identifier and can also have other attributes.
- The associative entity may participate in other relationships other than the linked entities.

1.6. Cardinality constraints in a ternary relationship (Figure 2-18)



Vendor_ID	Part_ID	Warehouse_ID	Shipping _Mode	Unit_Cost
123	P23e	w1	Truck	1000
345	P23e	w2	Air	500
123	P23e	w1	Rail	300
123	P34b	w5	…	…

Pause 

Class Activity 2.6: (10 minutes)

Consider a book rental system in a store. When a customer borrows or returns a book, the shop-keeper needs to mark down the transaction or update the corresponding record on the transaction book. Any customer can borrow many books for many times.

a. Create a list of business rules. (2 minutes)

b. Draw an ERD for this book rental system while: (3 minutes)
   - Identifying the type of relationship between the entities.
   - Identifying the cardinalities of the relationships.
   - Designing some attributes for each entity.

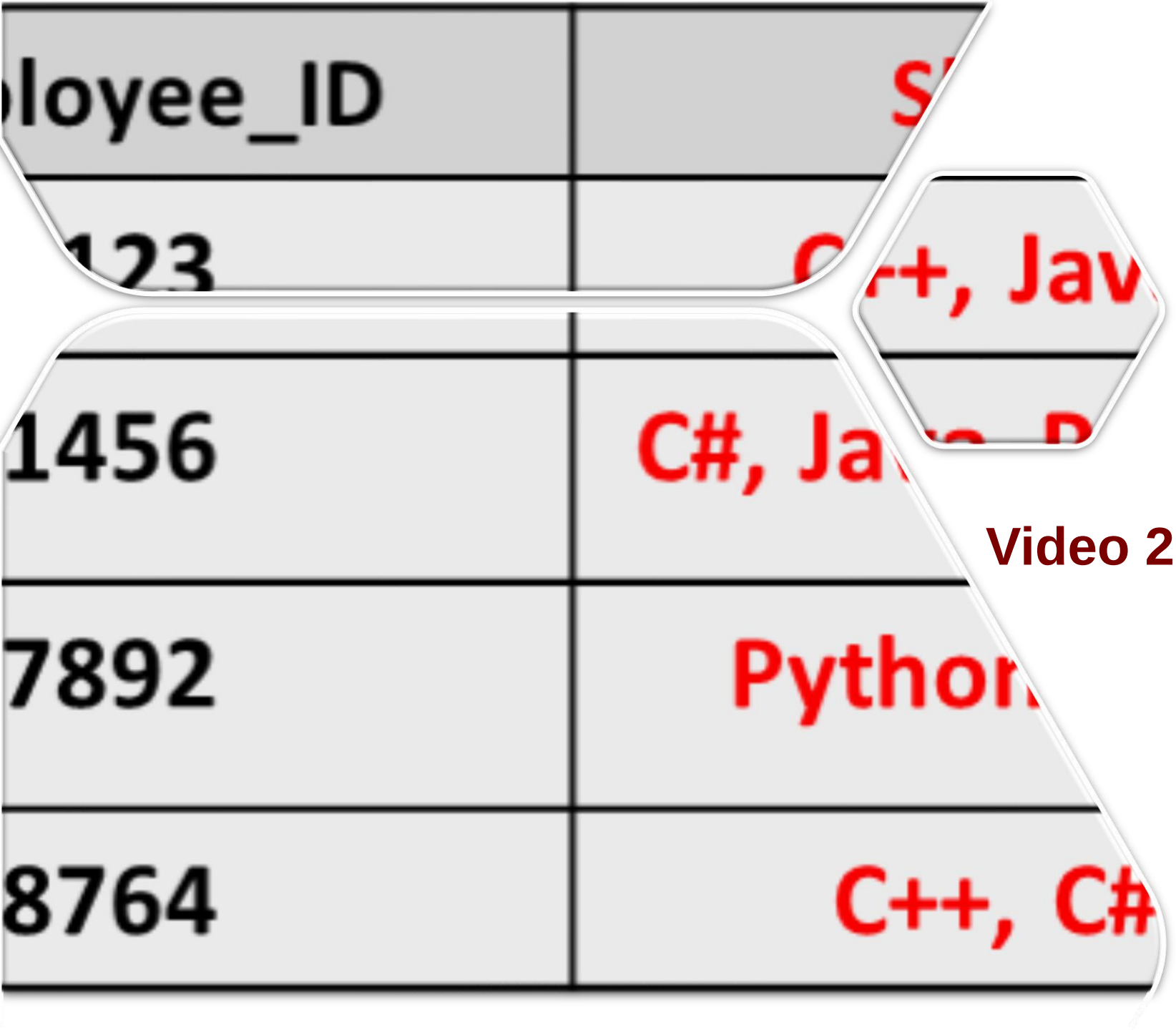
c. Determine the degree of a relationship in your ERD (Unary, Binary, or Ternary Relationship) (1 minutes)

d. Draw the correspond table to each entity with some sample data that shows how the data of each table are related to the data of the other tables by considering their PK and FK(s). Each table should have at least 3 rows of data. (4 minutes)

Solution to Class Activity 2.7

Solution to Class Activity 2.7 (cont.)

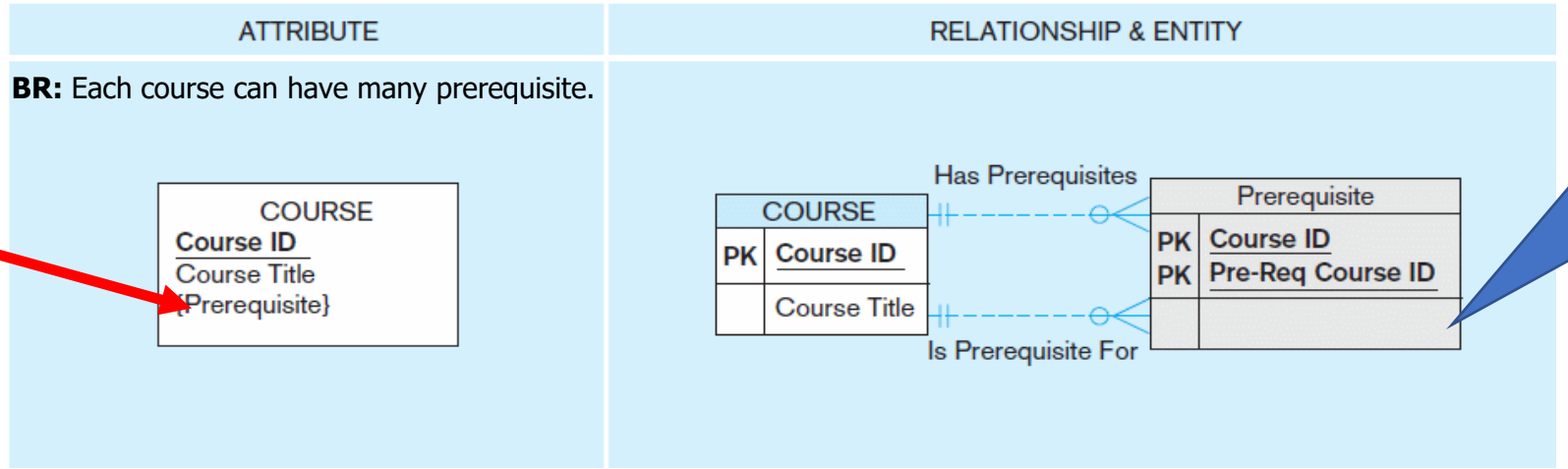
Solution to Class Activity 2.7 (cont.)



Video 2.6: Multivalued attributes

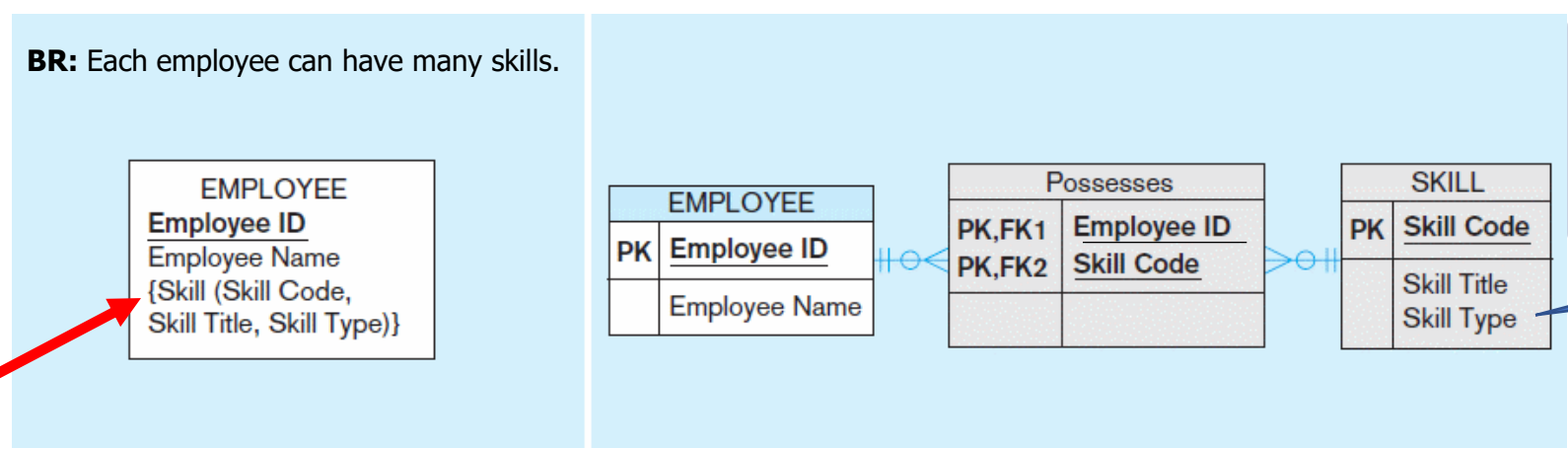
1.7. Relationship: Multivalued attributes can be represented as relationships (Figure 2-15a and 2-15b)

Prerequisite is a simple multivalued attribute



Here there is no other attribute for Prerequisite to be stored in the database

Skill is a composite Multivalued attribute



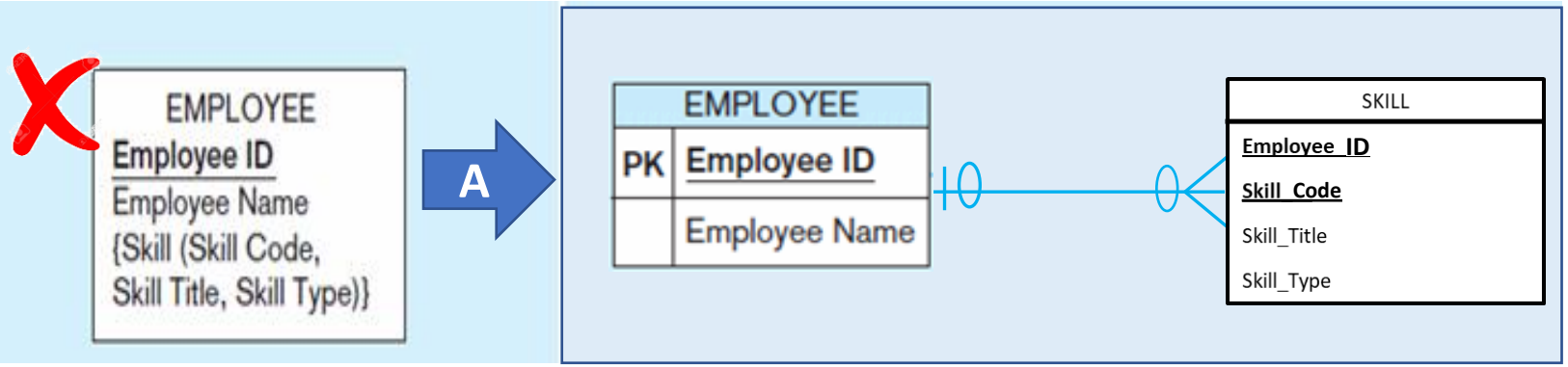
Here we have **Skills' attributes** that need to be stored in the database

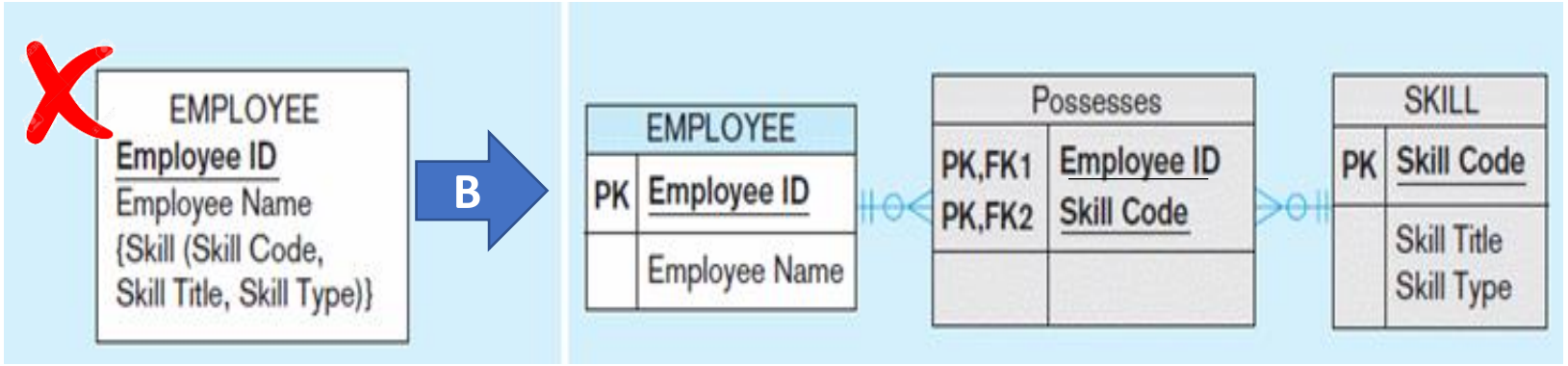
1.7. Relationship: Multivalued attributes can be represented as relationships.

EMPLOYEE
Employee_ID
Employee_F_Name
Employee_L_Name
{Skills (Skill Code, Skill Title, Skill Type)}

BR: Each employee can have many skills.

The ERD is redesigned. Let's see which solution (A or B) is practical.

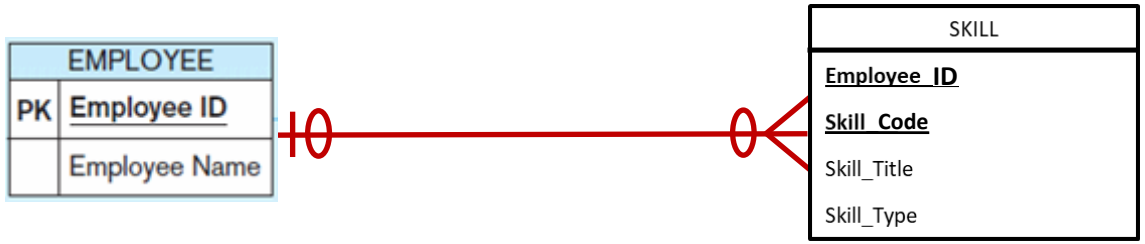




Redesign ERD to solve the problem with a multi-valued attribute

EMPLOYEE
Employee ID
Employee Name
{Skill (Skill Code, Skill Title, Skill Type)}

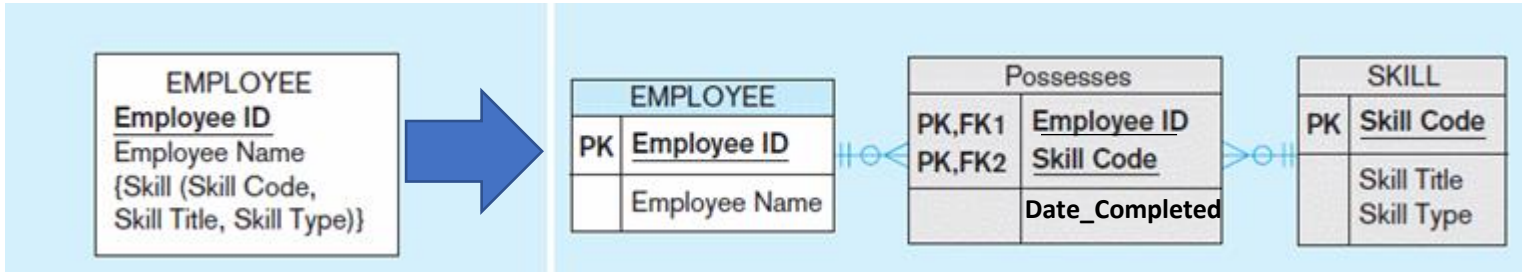
Employee_ID	Employee_F_Name	Employee_L_Name	Skill_Code	Skill_Tittle	Skill_Type
1123	Sara	Brown	B86, A23	C++, Java	PL, PL
1456	Jake	Cooper	C55, A23, C45	C#, Java, Python	PL, PL, PL
7892	Fahimeh	Ramezani	C45, B86	Python, C++	PL, PL
8764	Ricky	Romanous	B86, C55	C++, C#	PL, PL



Here we have repeated skill tittle and type.

Employee_ID	Employee_F_Name	Employee_L_Name
1123	Sara	Brown
1456	Jake	Cooper
7892	Fahimeh	Ramezani
8764	Ricky	Romanous

Employee_ID	Skill_Code	Skill_Title	Skill_Type
1123	A23	Java	PL
1123	B86	C++	PL
1456	C55	C#	CL
1456	A23	Java	PL
1456	C45	Python	PL
7892	C45	Python	PL
…	…	…	…



Explore in Class 2.3:
Let's jump to Ed and create this ERD.

Employee_ID	Employee_F_Name	Employee_L_Name	Skill_Code	Skill_Tittle	Skill_Type
1123	Sara	Brown	B86, A23	C++, Java	PL, PL
1456	Jake	Cooper	C55, A23, C45	C#, Java, Python	PL, PL, PL
7892	Fahimeh	Ramezani	C45, B86	Python, C++	PL, PL
8764	Ricky	Romanous	B86, C55	C++, C#	PL, PL

Employee_ID	Employee_F_Name	Employee_L_Name
1123	Sara	Brown
1456	Jake	Cooper
7892	Fahimeh	Ramezani
8764	Ricky	Romanous

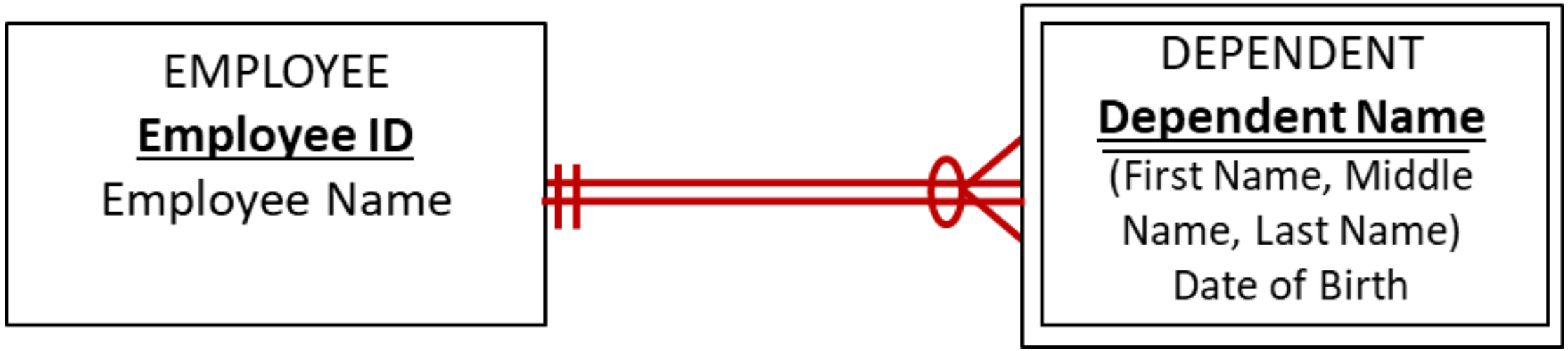
Employee_ID	Skill_Code	Date_Completed
1123	B86	1/1/2004
1123	A23	2/4/2007

Skill_Code	Skill_title	Skill_Type
A23	Java	PL
B86	C++	PL
C55	C#	PL
C45	Python	PL

Explore In Class (EIC)

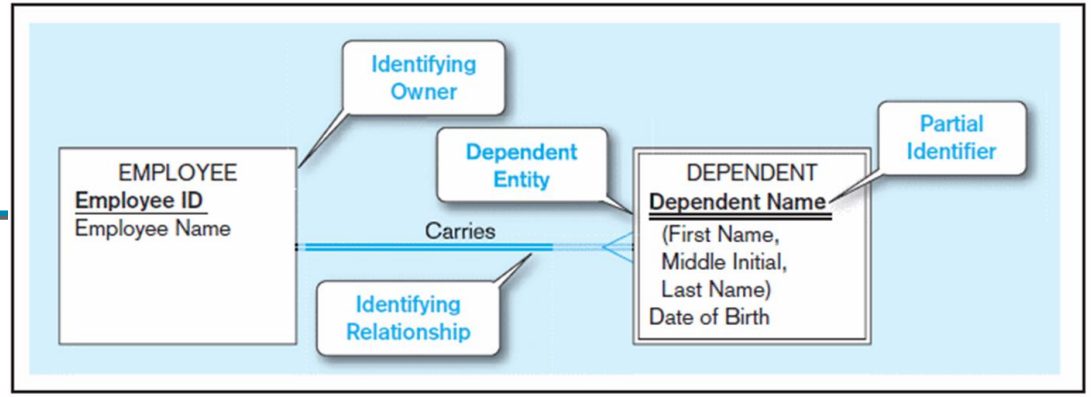
Go to → Lecture 2_EIC_2.3_Multi-Valued_Attribute

•



Video 2.7: Identifying Relationships
Strong and Weak Entities

1.8. Identifying Relationships: Strong vs. Weak Entities



Strong entity Weak entity

- **Strong entity**
  - exists independently of other types of entities
  - has its own unique identifier
    - identifier underlined with single line

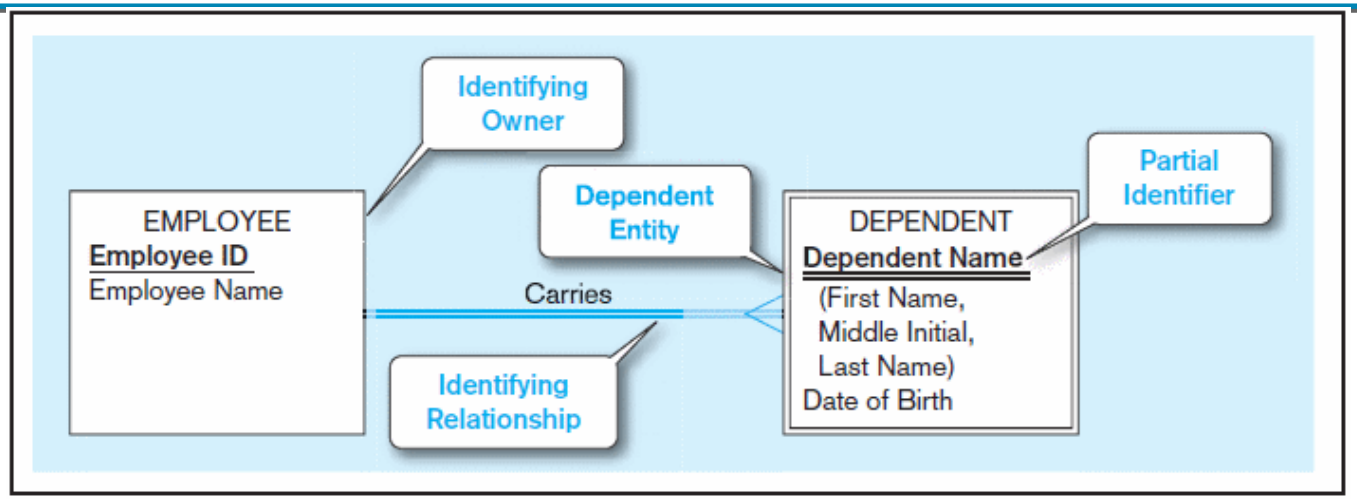
- **Weak entity**
  - dependent on a strong entity (identifying owner) and cannot exist on its own.
  - does not have a unique identifier (only has a **partial identifier**).
    - entity box has double lines (book notation).
    - partial identifier underlined with double lines (book notation).
  - Has a composite key of **its partial identifier** and **the PK of the strong entity.**

- **Identifying relationship**
  - links strong entities to weak entities,
    - shown by double lines (book notation).

Note: We call this Identifying relationship because the weak entity requires the key of the strong entity to be included as part of its own key.

1.8. PK of weak entity

Explore in Class 2.4:
Let's jump to Ed and create table **Employee.**



PK of the weak entity (Dependent)

PK

FK

Employee

Employee ID	Employee Name	Birth Date
1234587	Jack	1/1/1960
3459087	Michael	5/2/1985

Dependent

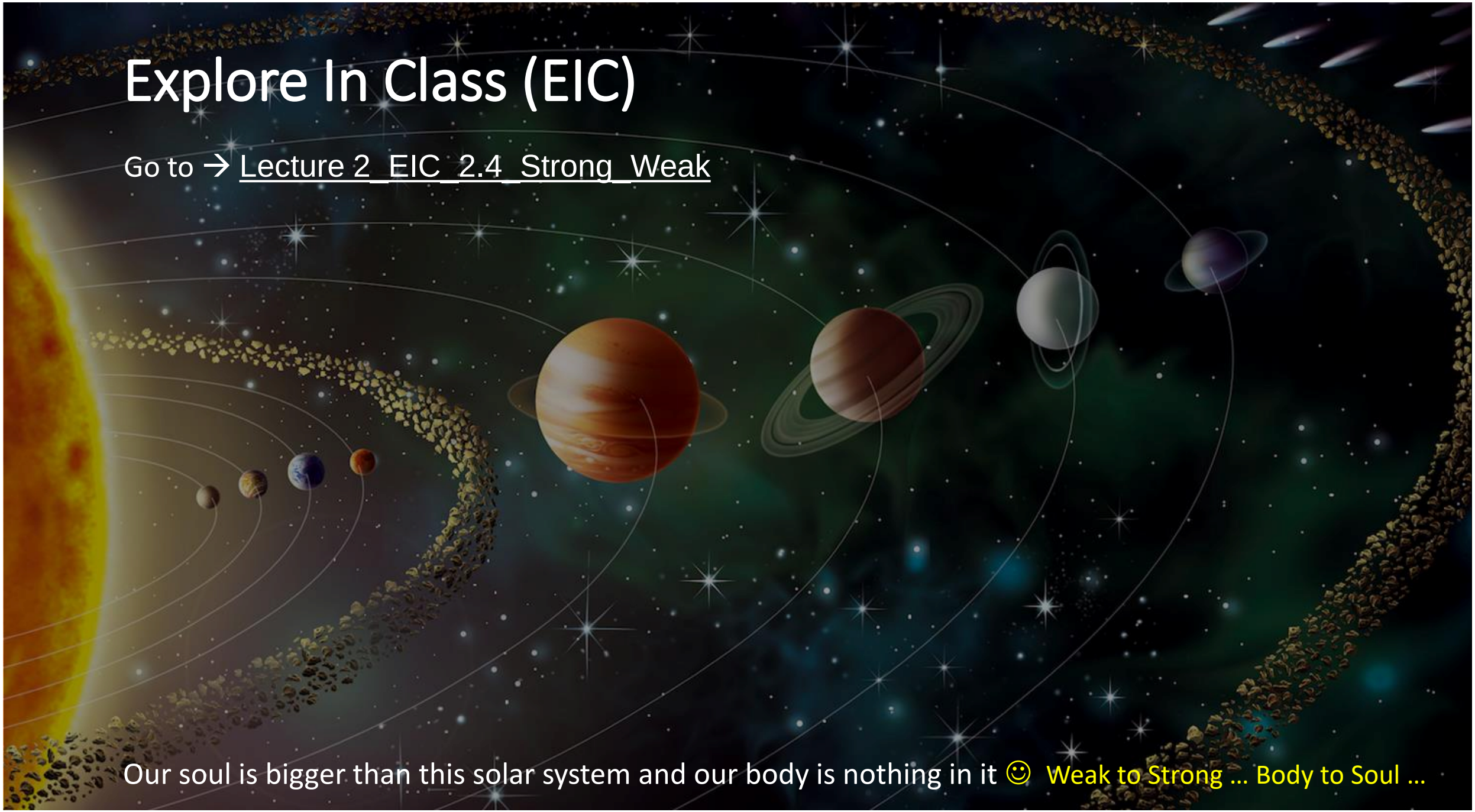
Employee ID	Dependent Name	Date of Birth
1234587	John Harris	1/3/2004
1234587	Jennifer Harris	5/7/2006
3459087	John Harris	1/1/2015

Important Rule:

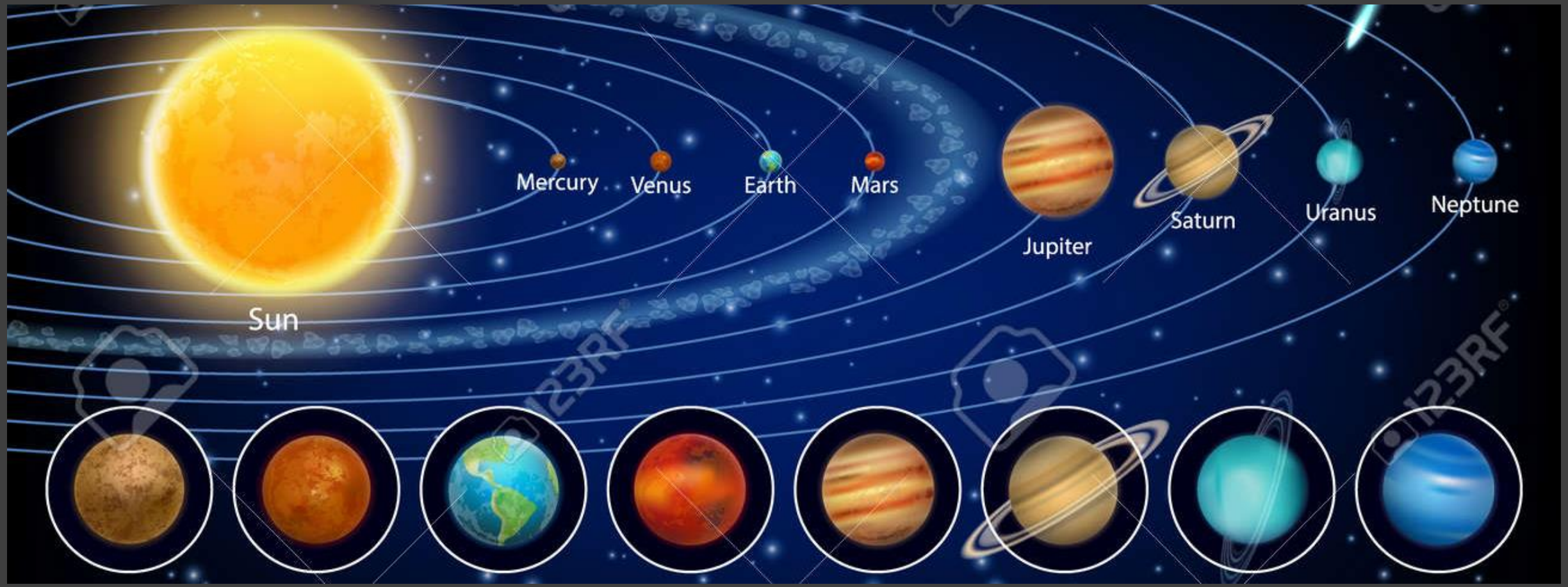
In each one-to-many relationships, PK of the entity on one side is FK of the entity on the many side

Explore In Class (EIC)

Go to → Lecture 2_EIC_2.4_Strong_Weak



Our soul is bigger than this solar system and our body is nothing in it ☺ Weak to Strong … Body to Soul …

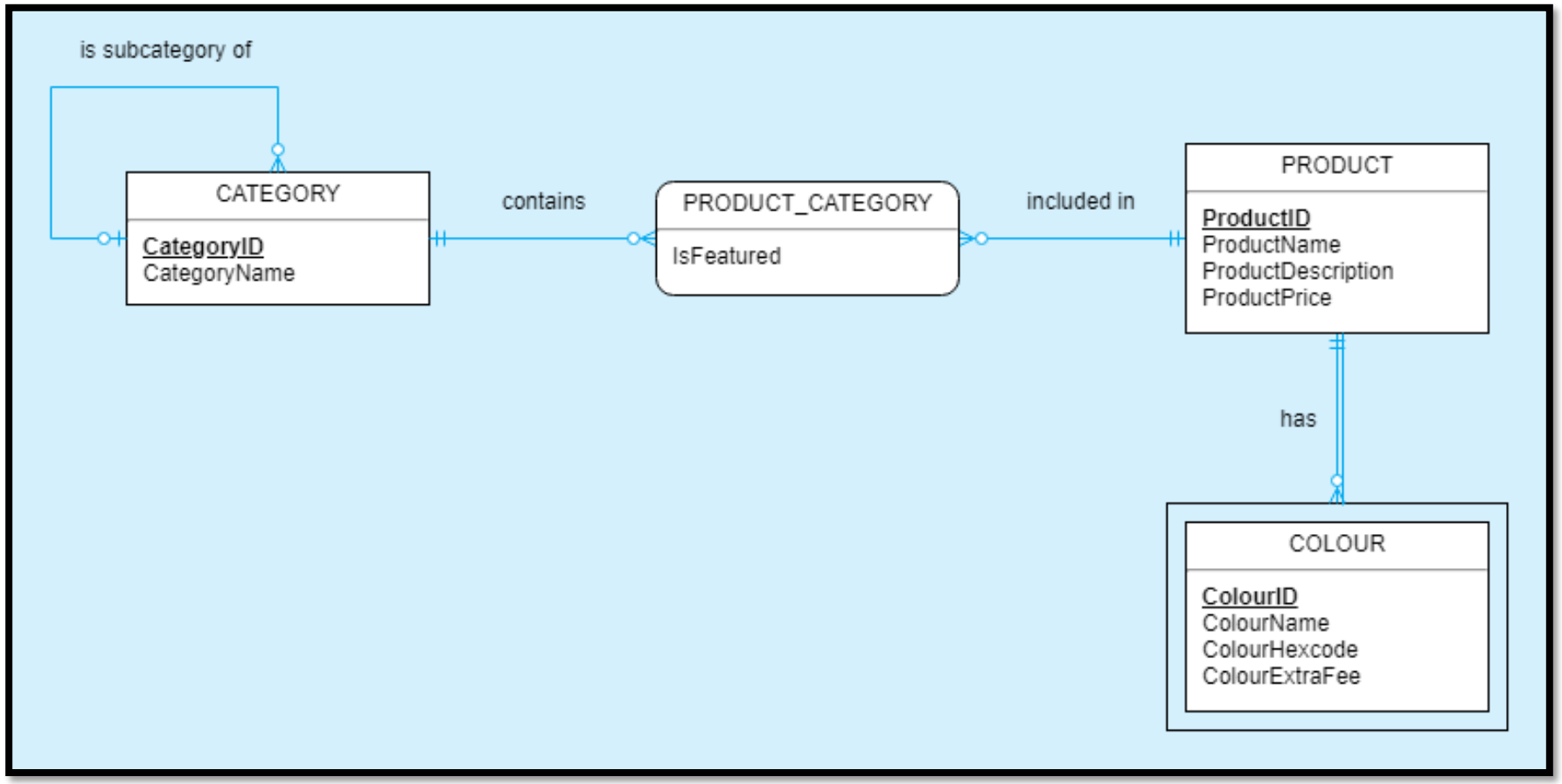


Now …

Can you design a database to store data related to planets, their information (name, size, etc.) and the distance between them?

Explore this page: https://www.britannica.com/science/solar-system

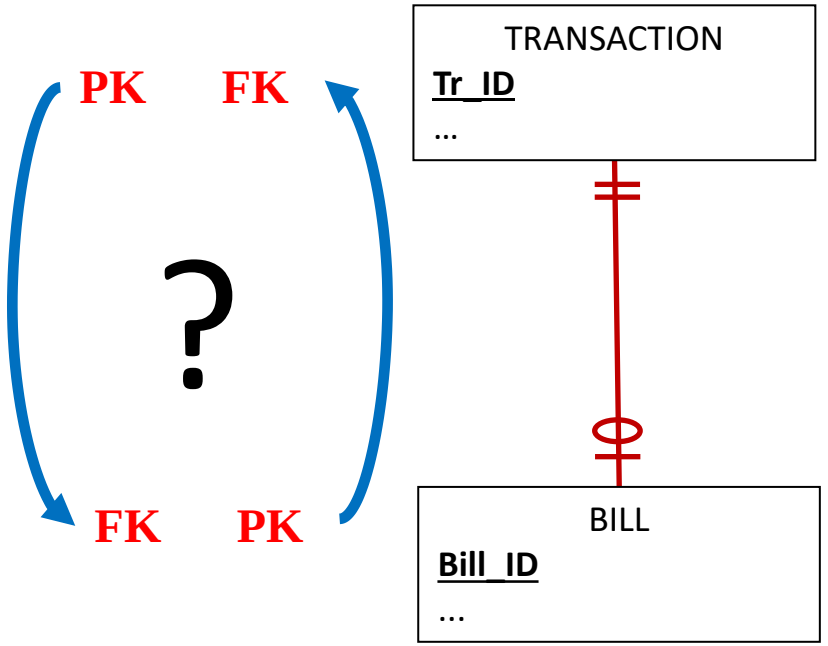
Class Activity 2.7: Determine the entity types in the Clothing Shop ERD. (2 minutes)

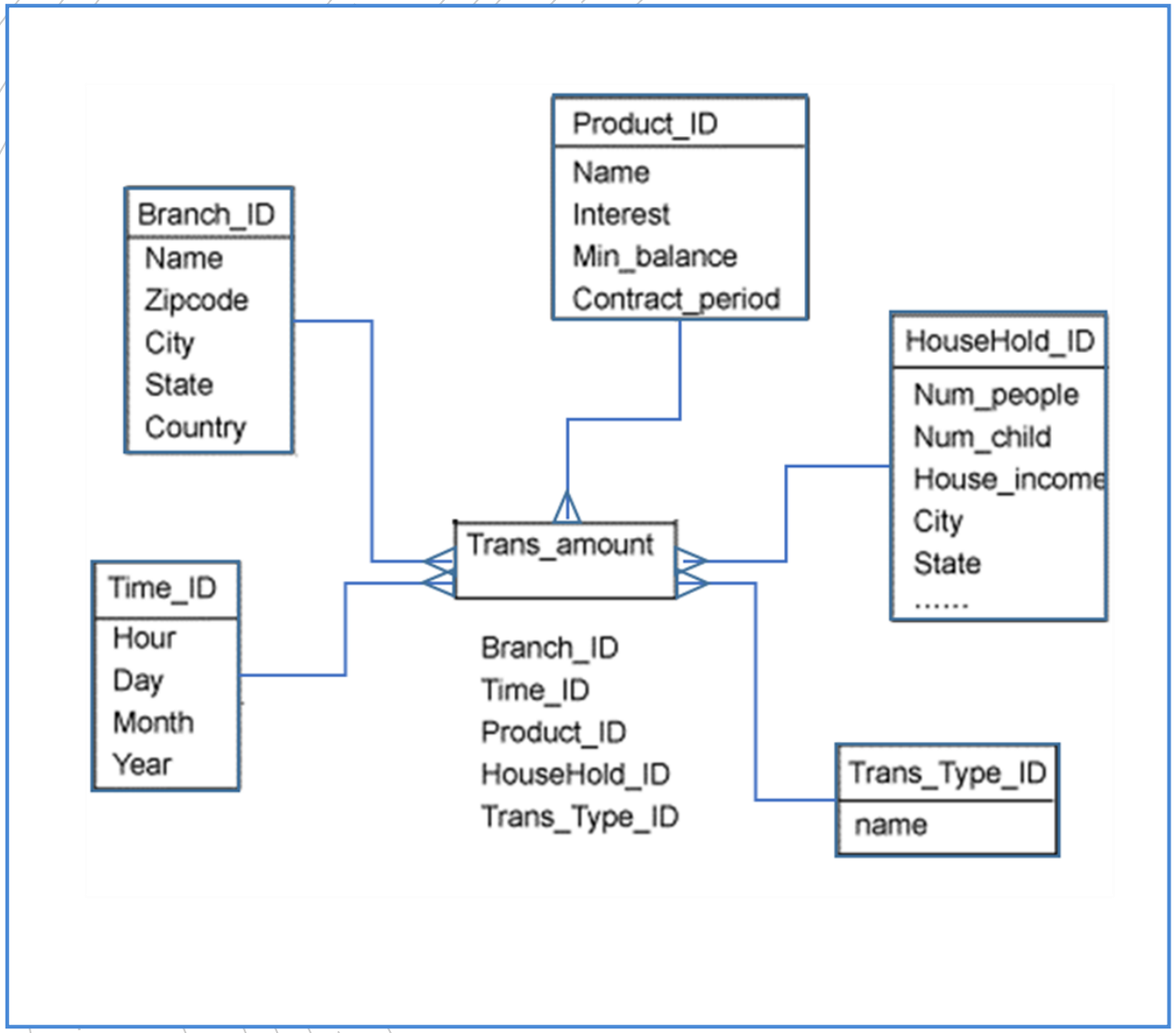


Solution- Entity Types

Assessment Challenge 1.2:

When we have **one-to-one** relationship, with **one optional** and **one mandatory** minimum cardinality, PK of which entity should be FK in another entity?



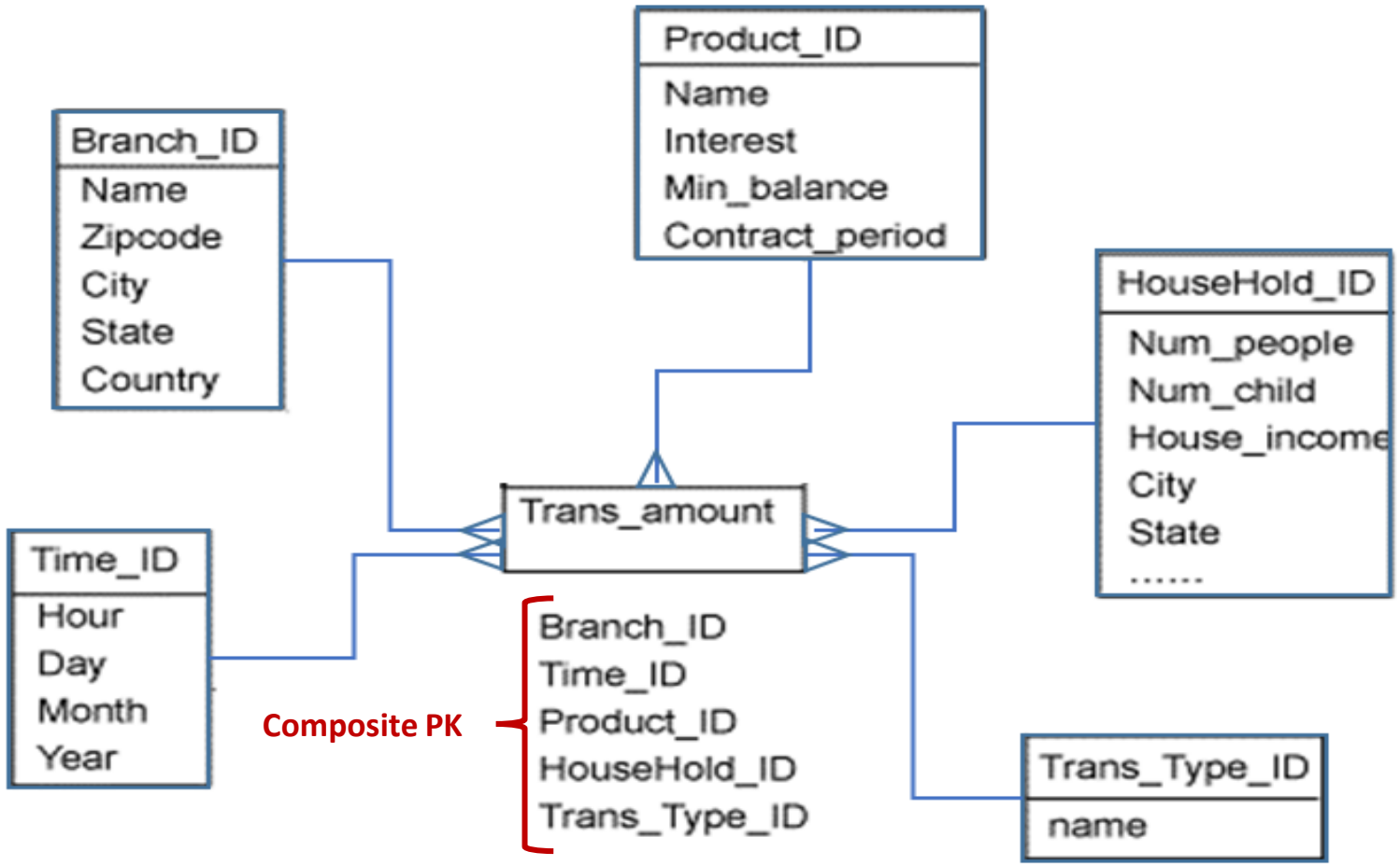


Video 2.8: Surrogate Key

Note: Learning about Surrogate Key is optional

Note: Learning about Surrogate Key is optional.

In Transaction table, we store the transaction amounts related to a product and a household, and a branch with specific transaction type and time.



Explore in Class 2.5:

Let's move to Ed and create **Transaction** table.

Review the rule: In each one-to-many relationships, **PK** of the entity on **one side** is **FK** of the entity on the **many side**.

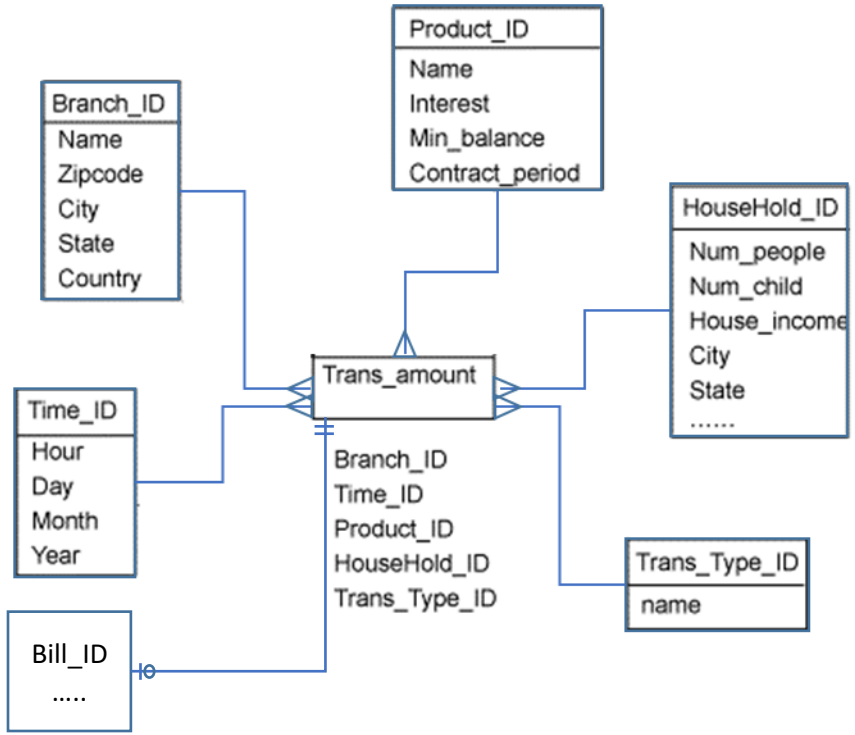
Surrogate Key for Transaction entity

The combination of "BranchID, TimeID, ProductID, HouseHoldID, TransTypeID" is the composite PK of the Transaction entity/table,

We can keep the combination of "BranchID, TimeID, ProductID, HouseHoldID and TransTypeID" unique and use system generated ID (Surrogate key) as well.

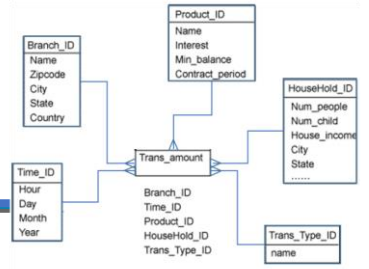
The combination of "BranchID, TimeID, ProductID, HouseHoldID, TransTypeID" which is the composite PK of the Transaction entity/table, will be the FK in Bill entity.

See the examples in the next slide to see what will be the FK in BILL if you use a surrogate key.



"BranchID, TimeID, ProductID, HouseHoldID, TransTypeID" is the composite PK of the Transaction table, so the FK in Bill table will be "BranchID, TimeID, ProductID, HouseHoldID, TransTypeID"

Key example: B11, T34, P87, H875, TT12



Transaction Table

PK

BranchID	TimeID	ProductID	HouseHoldID	TransTypeID	Trans_amount
B11	T34	P87	H875	TT12	50
B12	T38	P89	H900	TT18	200

Bill Table

FK

BranchID	TimeID	ProductID	HouseHoldID	TransTypeID	Bill_ID
B11	T34	P87	H875	TT12	14256
B12	T38	P89	H900	TT18	45712

Using Surrogate Key

"BranchID, TimeID, ProductID, HouseHoldID, TransTypeID" is the composite key of the Transaction table.
If you use surrogate key in Transaction Table instead of the composite PK, then the FK in Bill table will be the "Surrogate key" NOT the combination of "BranchID, TimeID, ProductID, HouseHoldID, TransTypeID".

Key example: 1

Transaction Table

Unique

PK

TA_Surrogate_Key	BranchID	TimeID	ProductID	HouseHoldID	TransTypeID	....
1	B11	T34	P87	H875	TT12	
2	B12	T38	P89	H900	TT18	

Bill Table

FK

TA_Surrogate_Key	Bill_ID
1	14256
2	45712

Important Note:

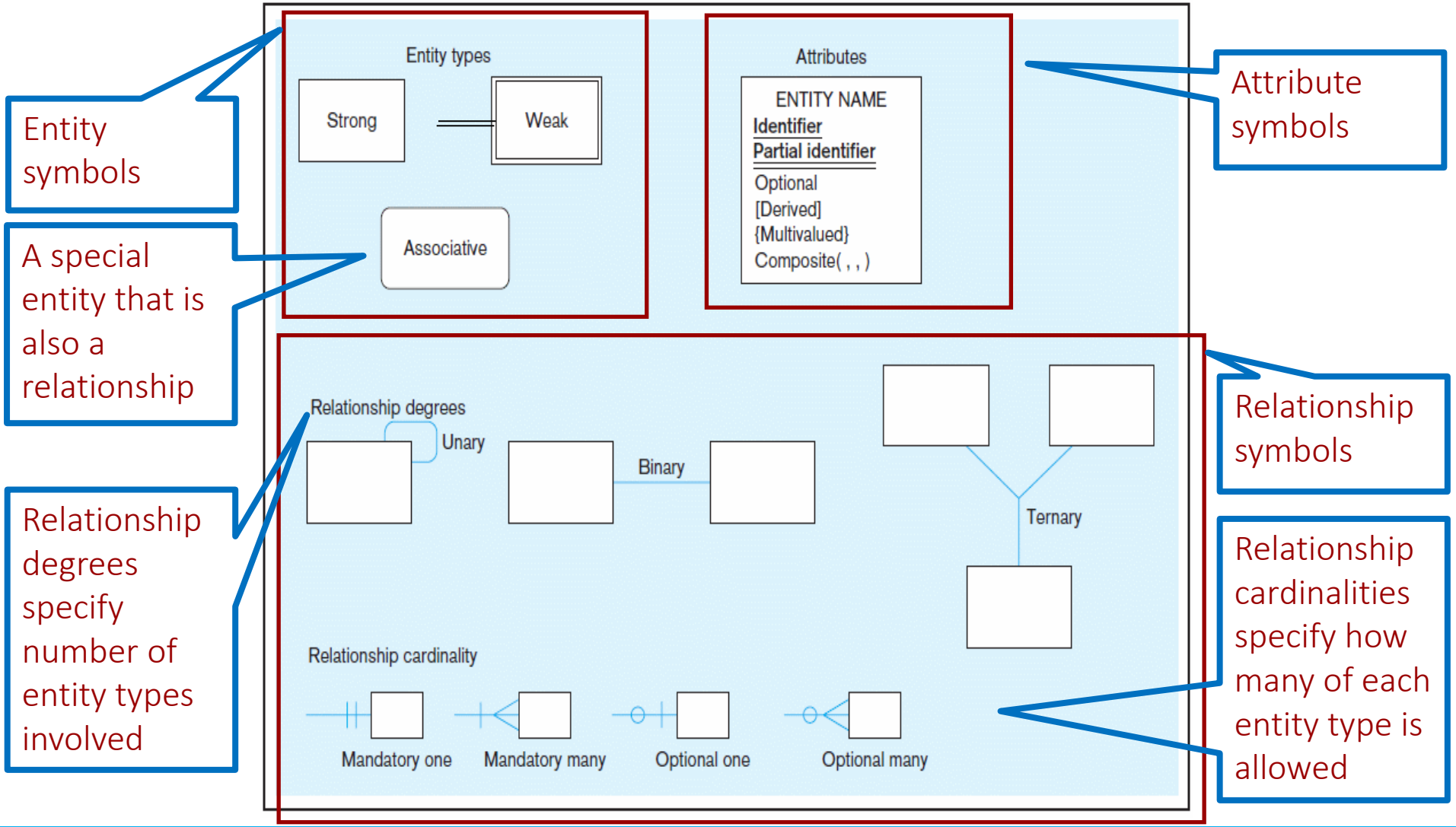
Using surrogate Key is a key skill for your future …

For now …

Do not use surrogate key in your assignments.

2. Notations (Crow's Foot)

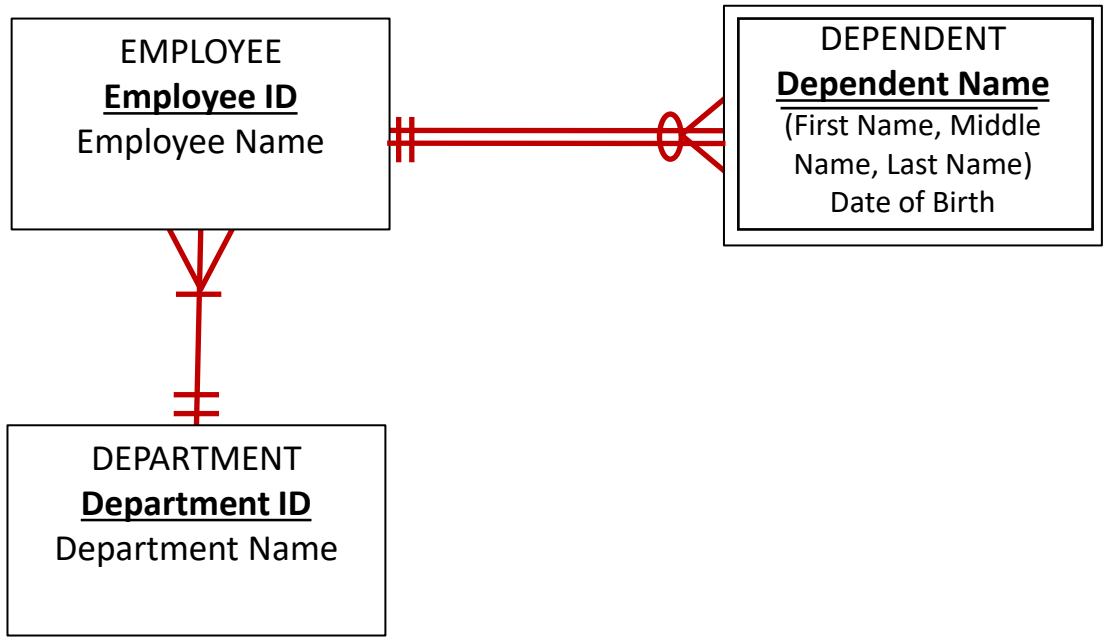
2.1. Basic Entity Relationship (ER) Notation (Figure 2-2)



2.2. Notation: Lines

- **Double Line:**
  - Between Weak and Strong entities

- **Solid Line:**
  - Between other entities



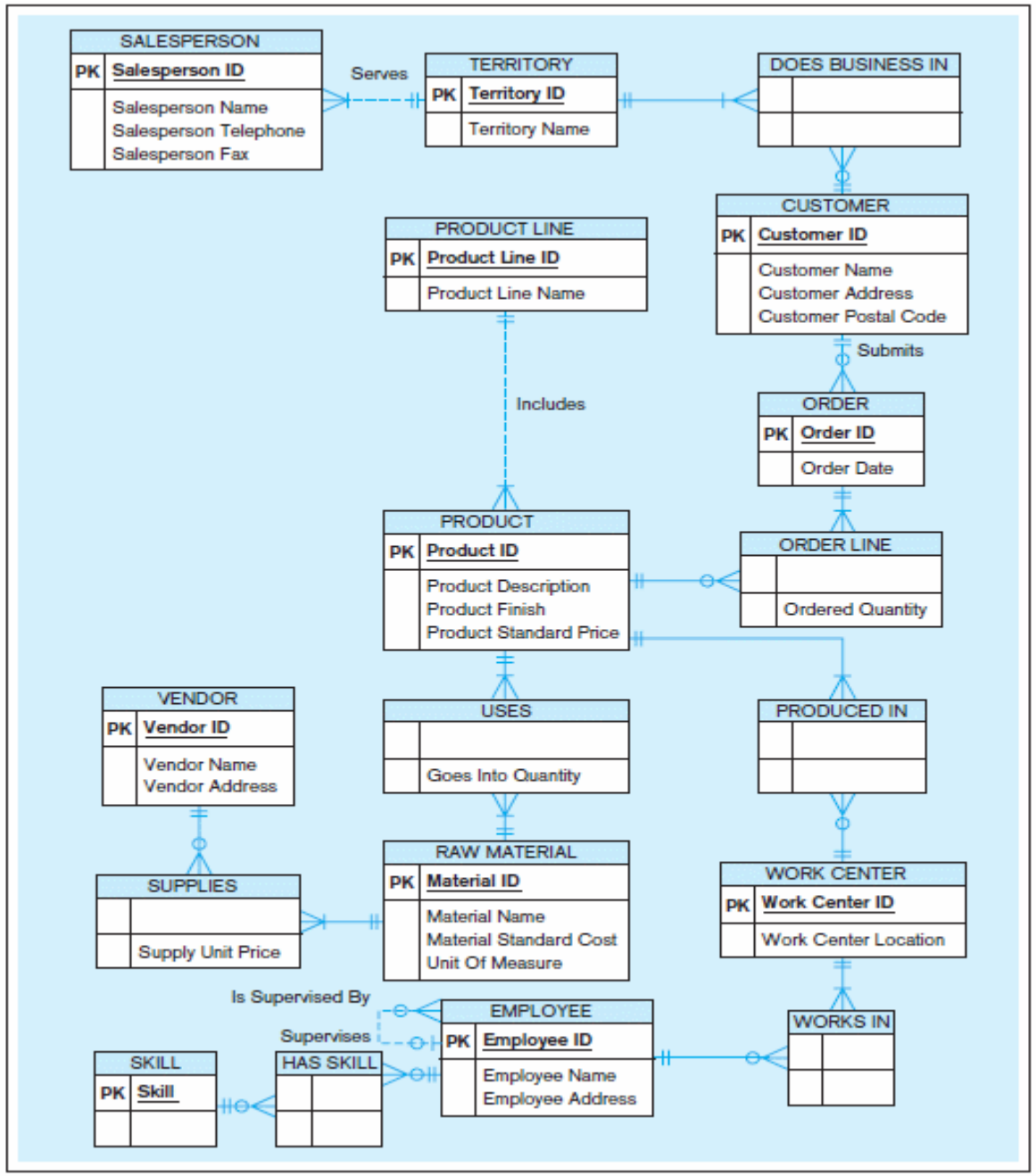


Figure 2-22
Data model for Pine Valley Furniture Company in Microsoft Visio notation

Different modeling software tools may have different notation for the same constructs.

Note: Please be aware that you are not allowed to use Visio notation in your assignment as this notation doesn't have a clear notation for weak and associative entities.

Summary

- ✓ Define terms
- ✓ Understand importance of data modeling
- ✓ Distinguish unary, binary, and ternary relationships
- ✓ Model different types of attributes, entities, relationships, and cardinalities
- ✓ Draw E-R diagrams for common business situations
- ✓ Convert many-to-many relationships to associative entities
- ✓ Distinguish weak and strong entities (Identifying Relationship)
- ✓ Notations

Next Lecture…

1. **Supertype/Subtype Relationships**
2. **Relationships and Subtypes**
3. **Generalization and Specialization**
4. **Constraints in Supertype/Subtype Relationships**
   - 4.1. Completeness Constraints
   - 4.2. Disjointness Constraints
   - 4.3. Subtype Discriminator