

SEEM3430 – Tutorial 2

Entity Relationship Diagrams

Luyang Lin
lylin@se.cuhk.edu.hk
31/10/2019



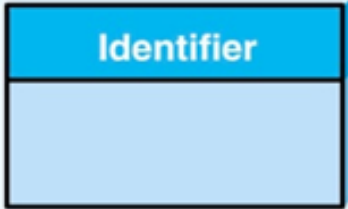

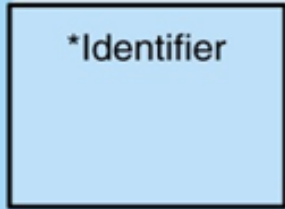
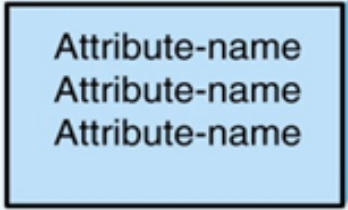

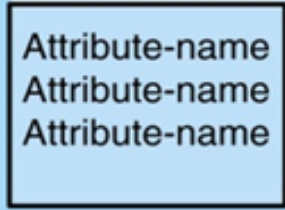

What we cover today?

- What is ERD?
 - Definition
 - Building blocks
- How to draw ERDs
 - An example
- Database Normalization

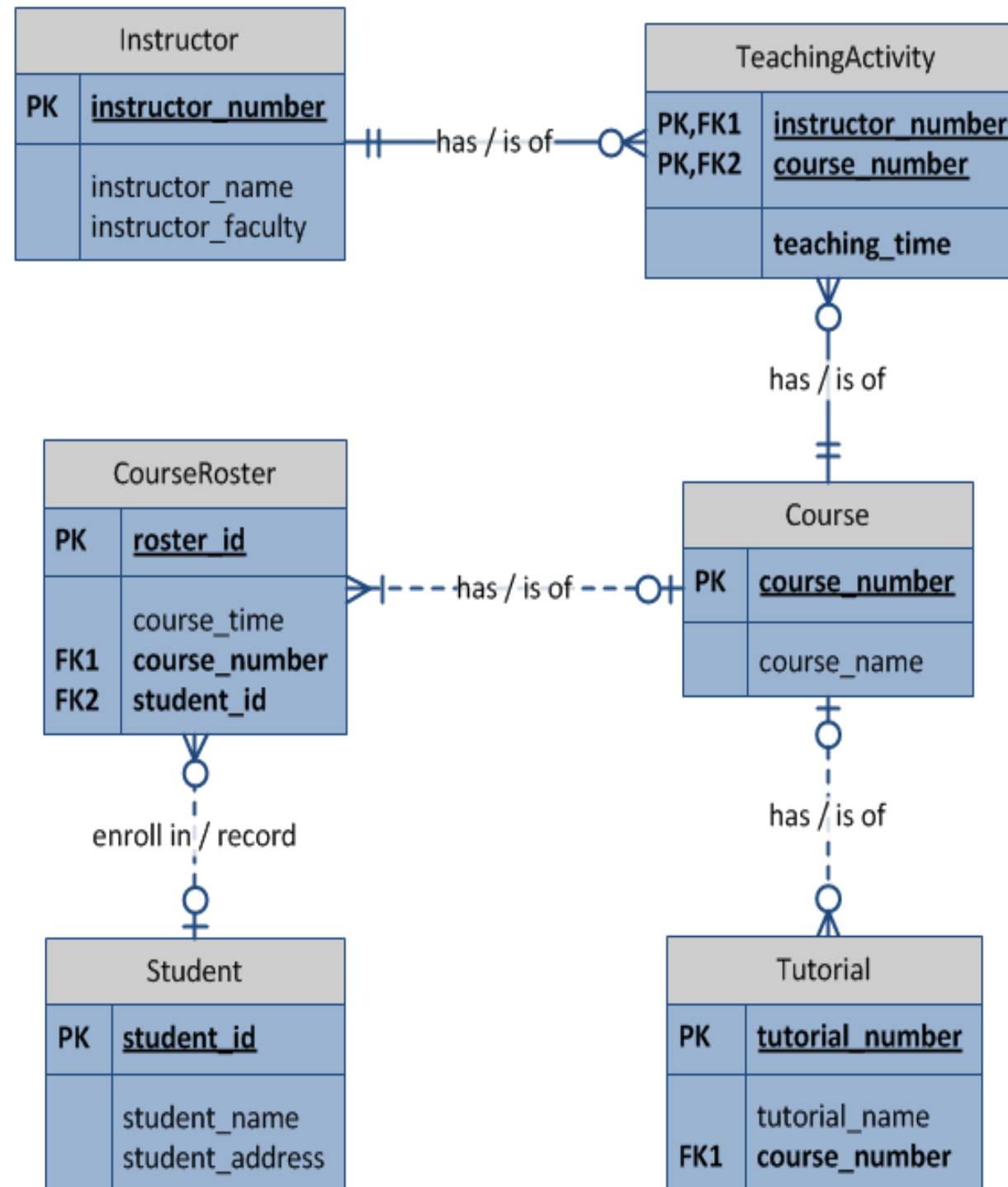
Definition

- An approach for data modeling
- A database is divided into two logical parts:
 - Entities
 - Relationships
- Diagrams created to design entities and relationships are called entity-relationship diagrams

Symbol Sets

	IDEF1X	Chen	Crow's Foot
An ENTITY ✓ is a person, place, or thing. ✓ has a singular name spelled in all capital letters. ✓ has an identifier. ✓ should contain more than one instance of data.	ENTITY-NAME 	ENTITY-NAME 	ENTITY-NAME 
An ATTRIBUTE ✓ is a property of an entity. ✓ should be used by at least one business process. ✓ is broken down to its most useful level of detail.	ENTITY-NAME 		ENTITY-NAME 
A RELATIONSHIP ✓ shows the association between two entities. ✓ has a parent entity and a child entity. ✓ is described with a verb phrase. ✓ has cardinality (1 : 1, 1 : N, or M : N). ✓ has modality (null, not null). ✓ is dependent or independent.	<u>Relationship-name</u>		<u>Relationship-name</u>

An ERD Example



Entity

- Real-world objects, e.g., student, book
- Entity instance: a particular entity, e.g., Harry Potter
- Represented by a named rectangle
- Attributes and identifiers

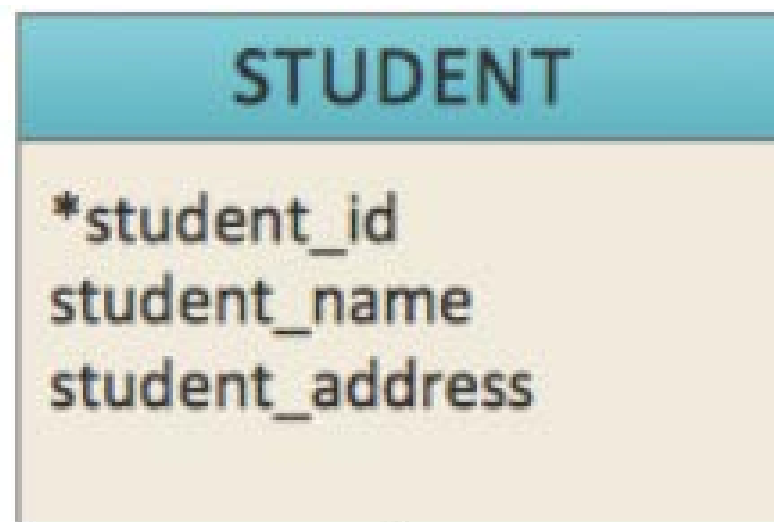
Instructor	
PK	<u>instructor_number</u>
	instructor_name instructor_faculty

Attribute

Attribute describes the characteristics of an entity.

- Primary key.
- A key attribute is unique, distinguishing characteristic of the entity.

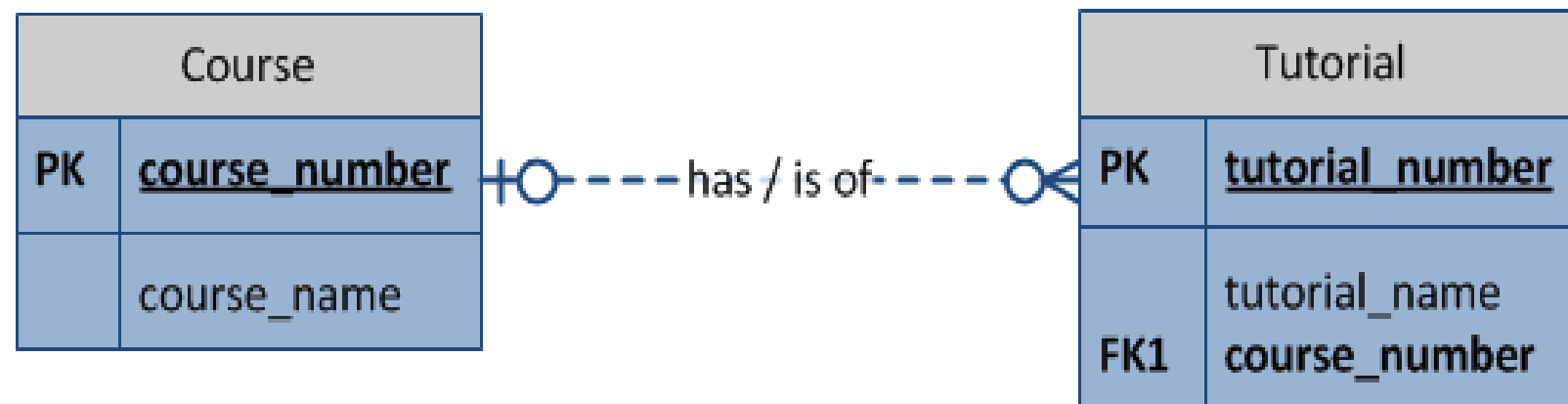
Identifier: One or more attributes that can uniquely identify one instance of an entity, which noted by an asterisk next to the attribute name.



Relationship

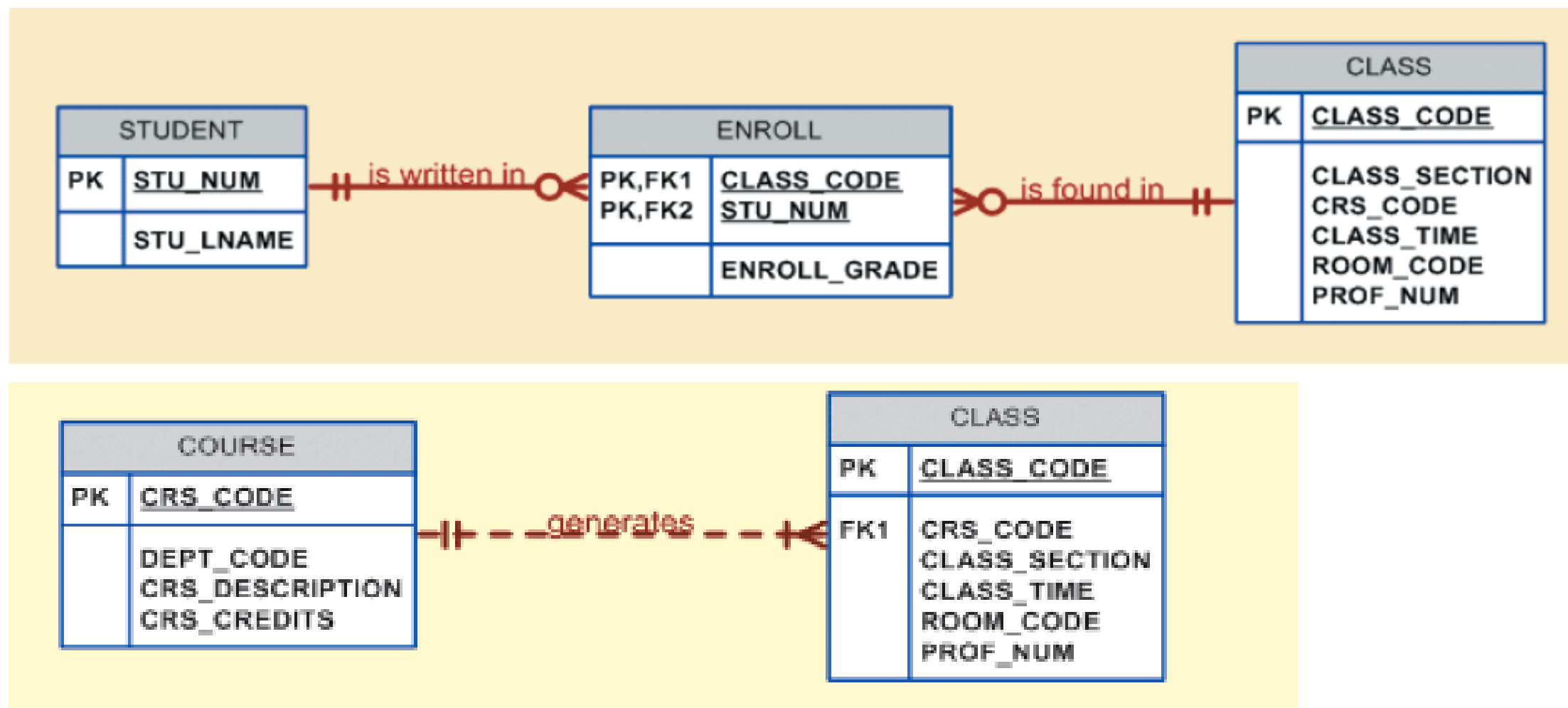
A relationship is an association among entities.

- Relationships can be thought of as verbs, linking two or more nouns.
- Examples: an “owns” relationship between a company and a computer, a “performs” relationship between an artist and a song.



Relationship

- Strong Relationships(solid lines): the FK (foreign key) of the related table is also involved in its PK (primary key), along with being the PK of another table.
- Weak Relationships(dashed line): the relationship is not strong. E.g. Sales had the Agent's ID (the Agent PK) as a FK, which is not its PK.







Cardinality

Cardinality specifies how many instances of an entity relate to one instance of another entity.

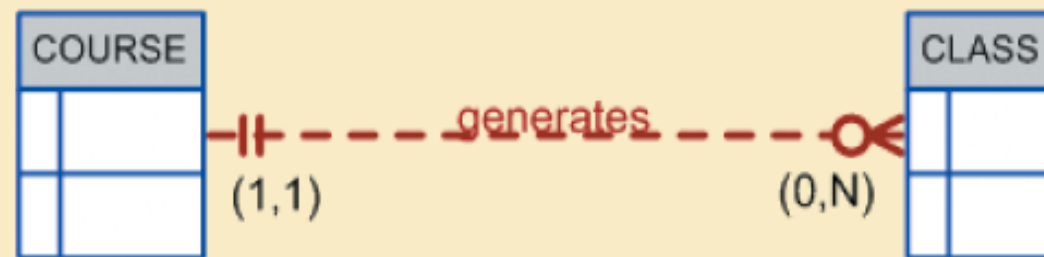
1:1 relationship: one instance of the parent entity is associated with one instance of the child entity.

1:N relationship: a single instance of a parent entity is associated with many instances of a child entity.

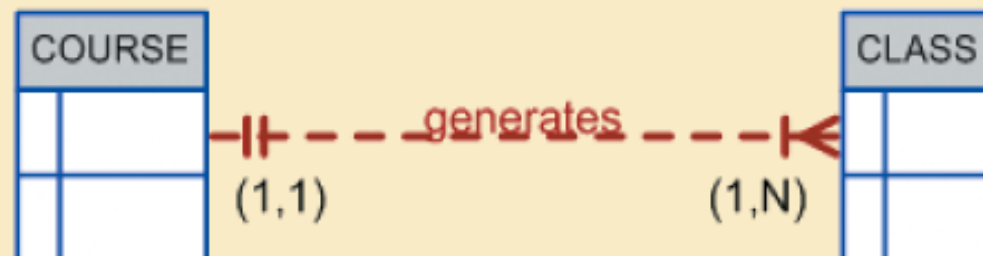
M:N relationship: many instances of a parent entity can relate to many instances of a child entity.

CROW'S FOOT SYMBOL	CARDINALITY	COMMENT
	(0,N)	Zero or many. Many side is optional.
	(1,N)	One or many. Many side is mandatory.
	(1,1)	One and only one. 1 side is mandatory.
	(0,1)	Zero or one. 1 side is optional.

Cardinality



One course can have multiple classes.
Some courses may not have classes.

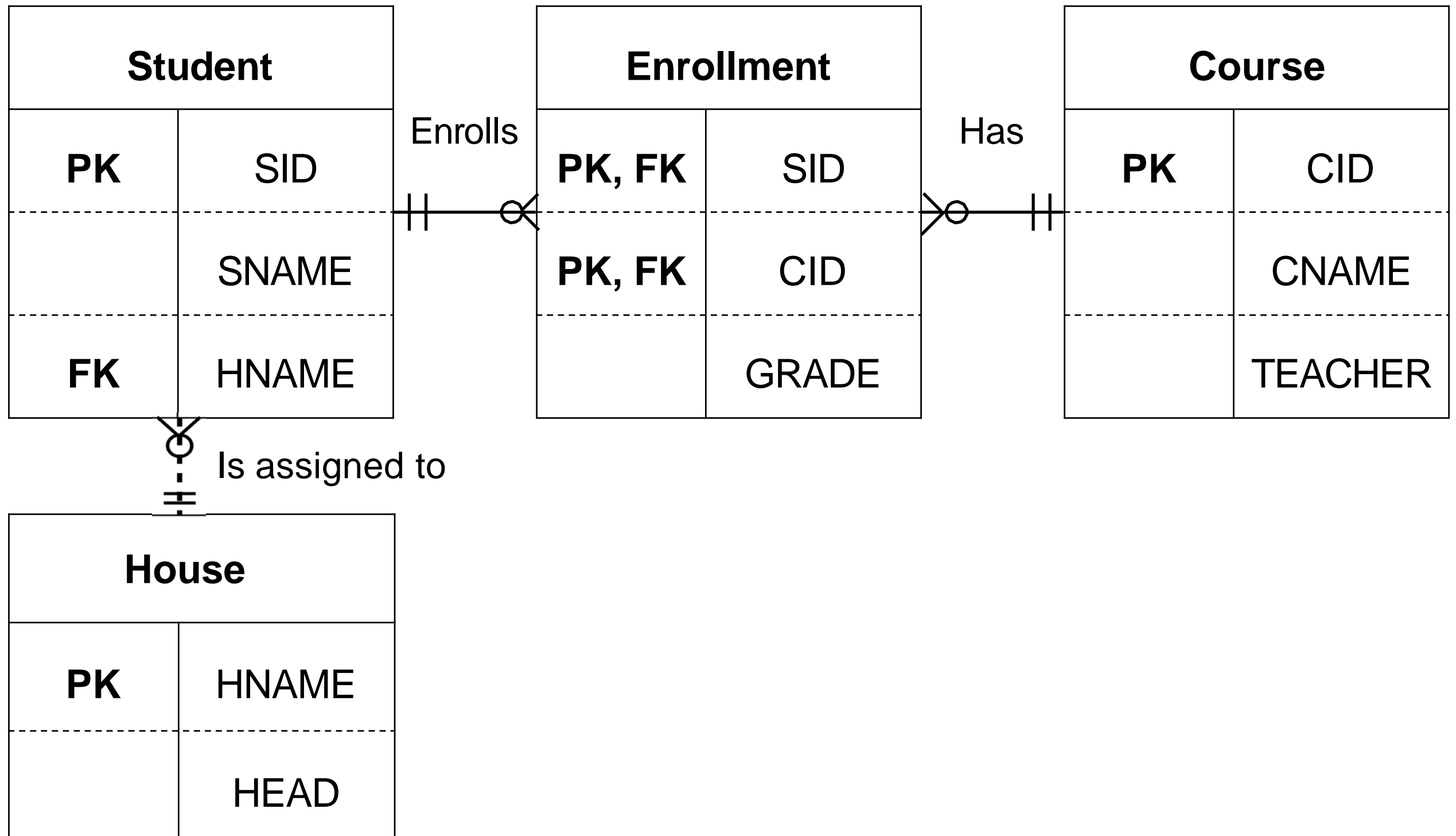


One course can have multiple classes.
Each course must have at least one class.

Example

- Prof. McGonagall found that the modern student information management system convenient
- She would like to build up one for students in Hogwarts too. But she has been in the magic world for too long. Can you help her draw the ERD?
- Basic entities:
 - Students (SID, SNAME, HOUSE, COURSE, GRADE)
 - Courses (CID, CNAME, TEACHER)
 - Houses (HNAME, HEAD)

Solution

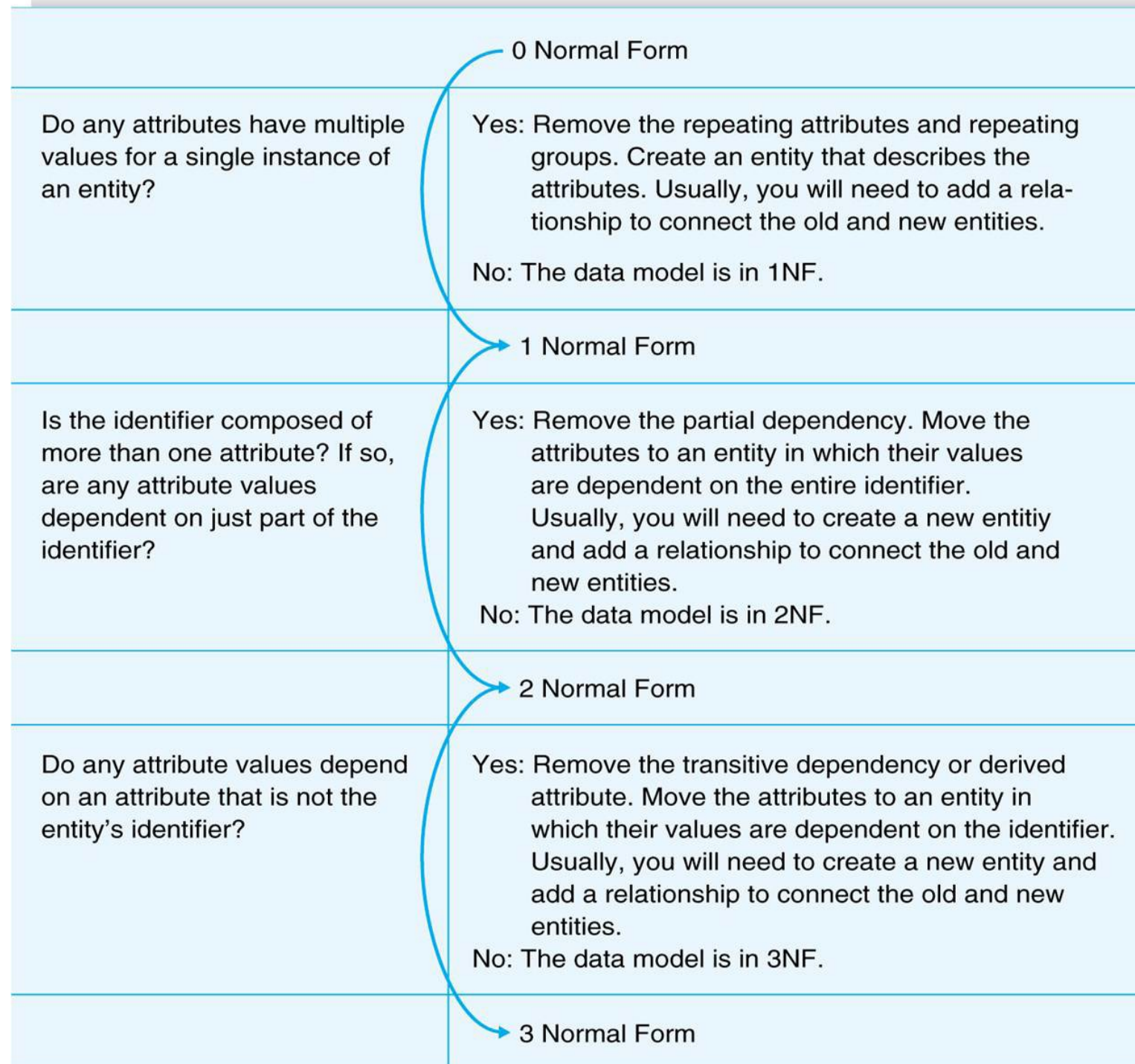


Database Normalization




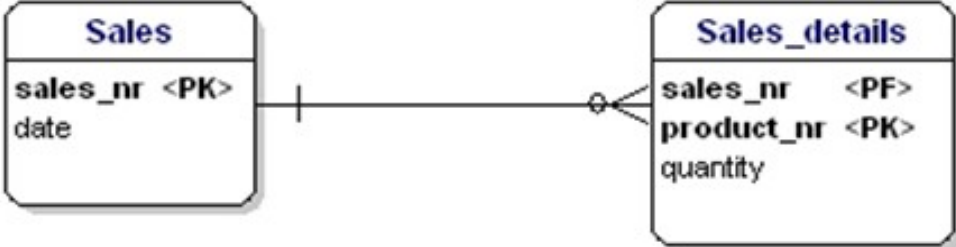

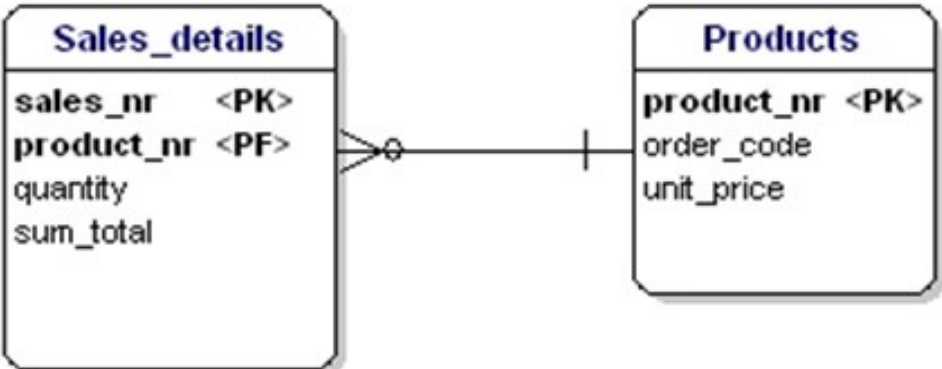
Definition:

Normalization is the process of organizing data in a database. This includes **creating tables and establishing relationships** between those tables according to rules designed both to protect the data and to make the database more flexible by **eliminating redundancy and inconsistent dependency**.

Normalization Process



Normalization Process






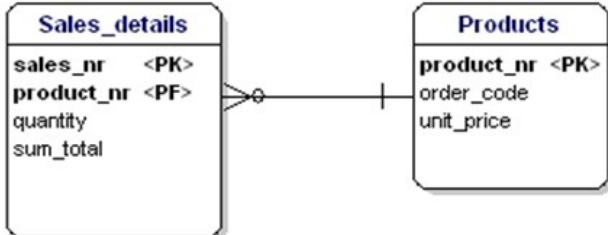
Normalization	Before	After
1st normal form	 <pre> erDiagram SALES --o{ SALES_DETAILS : "" SALES { string product1 string product2 string product3 float sum_total date date } SALES_DETAILS { float sales_nr PK string product_nr PF } </pre>	 <pre> erDiagram SALES --o{ SALES_DETAILS : "" SALES { float sales_nr PK float sum_total date date } SALES_DETAILS { float sales_nr PF string product_nr } </pre>
2nd normal form	 <pre> erDiagram SALES --o{ SALES_DETAILS : "" SALES { float sales_nr PK } SALES_DETAILS { float sales_nr PK string product_nr PK date date float quantity } </pre>	 <pre> erDiagram SALES --o{ SALES_DETAILS : "" SALES { float sales_nr PK date date } SALES_DETAILS { float sales_nr PF string product_nr PK float quantity } </pre>
3rd normal form	 <pre> erDiagram SALES --o{ PRODUCTS : "" SALES { float sales_nr PK string product_nr PK string order_code float unit_price float quantity float sum_total } PRODUCTS { string product_nr PK string order_code float unit_price } </pre>	 <pre> erDiagram SALES --o{ PRODUCTS : "" SALES { float sales_nr PK string product_nr PF float quantity float sum_total } PRODUCTS { string product_nr PK string order_code float unit_price } </pre>

Normalization Process

After 1st Norm, the repeated attributes (product1, product2, product3) was moved to a new table, Sales was created.

After 2nd Norm, the partial dependency of date on sales nr was removed and Sales table was created.

After 3rd Norm, the transitive dependency of order code and unit price on product nr was removed and Products table was created.

Normalization	Before	After
1st normal form		
2nd normal form		
3rd normal form		

Normalization Process

1NF: A single attribute is not allowed to contain multiple values; no repeating/duplicated attributes.

2NF: If an attribute depend on the primary key, then it must depend on the entire primary key (or super key), i.e. no partial dependencies on a concatenated primary key. But it does not ensure all attributes depend on the primary key.

3NF: All attributes must directly depend on the entire primary key (or super key), i.e. no dependencies on non-key attributes.

Summary

- Identify *Entities* for your data modeling problem
- Add *Attributes* for each entity and determine the primary key
- Establish *Relationships* between entities
- Check the *Cardinalities*
- Normalize your database via 1NF, 2NF, 3NF
- Review your E-R diagram and *Refine* it

Thank you!