

ERM - Entity Relationship Modeling

- ERD depicts the conceptual model of a database
 - Entities - Entity is a set of objects, not a single instance or occurrence
 - Attributes - Characteristics of an entity
 - Relationships - Association between entities

Attributes

- Domain – Set of possible values for a given attribute
- Types (not to be confused with data types)
 - Required Must have a value, cannot be null
 - Optional Does not require a value, can be null
 - Simple Cannot be subdivided
 - Composite Can be subdivided to yield additional attributes
 - Single-valued Has only a single value (not to be confused with domain)
 - Multivalued Can have many values (not to be confused with domain)
 - Derived Value is calculated from other attributes

Attributes Types (Example)

- STUDENT entity (some possibilities)

• CWID	required	simple	single-valued
• FIRST NAME	required	simple	single-valued
• LAST NAME	required	simple	single-valued
• MIDDLE INIT	optional	simple	single-valued
• ADDRESSES	required	composite	multivalued
• PHONES	optional	simple	multivalued
• GPA	required	simple	single-valued
• DOB	required	simple	single-valued
• AGE	required	simple	single-valued
			derived

Attributes Types (Example)

- ADDRESSES as single valued and simple

- PERMANENT

• PERM_LINE1	required	simple	single-valued
• PERM_LINE2	optional	simple	single-valued
• PERM_STATE	required	simple	single-valued
• PERM_ZIP	required	simple	single-valued

- LOCAL

• LOCAL_LINE1	optional	simple	single-valued
• LOCAL_LINE2	optional	simple	single-valued
• LOCAL_STATE	optional	simple	single-valued
• LOCAL_ZIP	optional	simple	single-valued

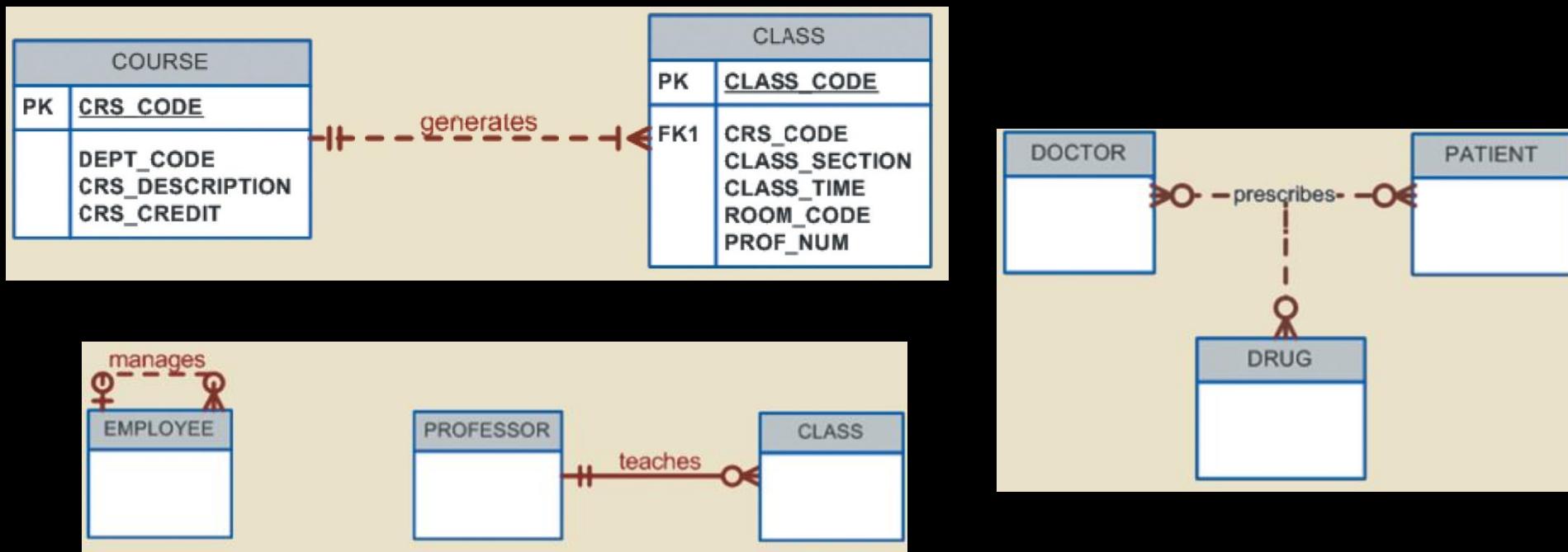
Attributes Types

- AGE as a derived attribute, calculated from the DOB

ADVANTAGES AND DISADVANTAGES OF STORING DERIVED ATTRIBUTES		
	DERIVED ATTRIBUTE	
	STORED	NOT STORED
Advantage	Saves CPU processing cycles Saves data access time Data value is readily available Can be used to keep track of historical data	Saves storage space Computation always yields current value
Disadvantage	Requires constant maintenance to ensure derived value is current, especially if any values used in the calculation change	Uses CPU processing cycles Increases data access time Adds coding complexity to queries

Entities

- Participants – Entities participating in a relationship

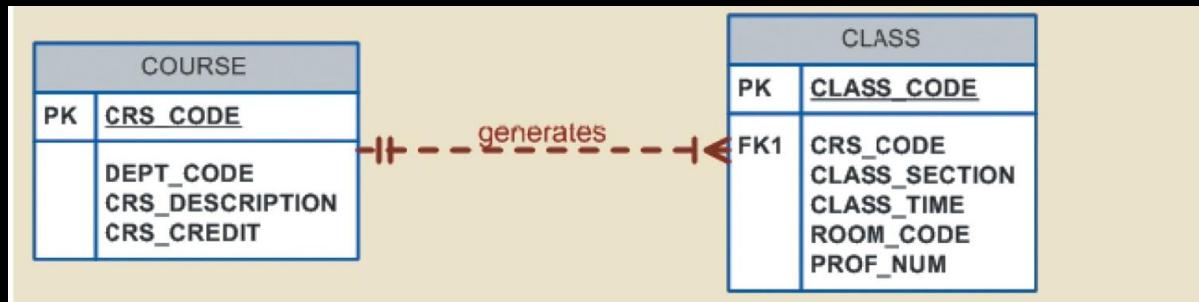


Entities

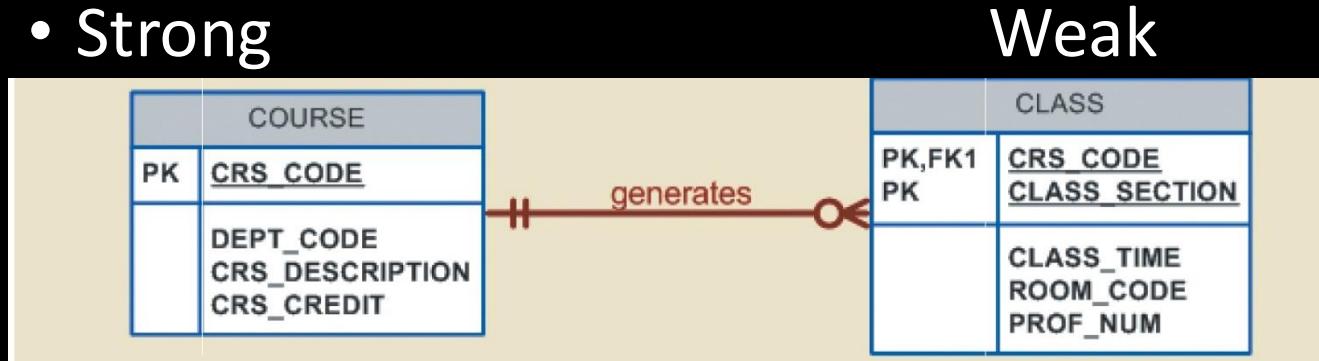
- Entity strength – Determined by property of existence dependence
 - Existence-dependent (weak) – exists only when it is associated with another related entity occurrence
 - It cannot exist without the related entity
 - PK is partially or totally derived from its related entity
 - It has a required foreign key
 - Example: employees having dependents
 - Existence-independent (strong) – exists apart from its related entities
 - It can exist without any related entities
 - PK is not derived from its related entity
 - It does not have any required foreign keys
 - Example: company produce some parts while also getting some parts from a vendor

Entity Strength

- Strong



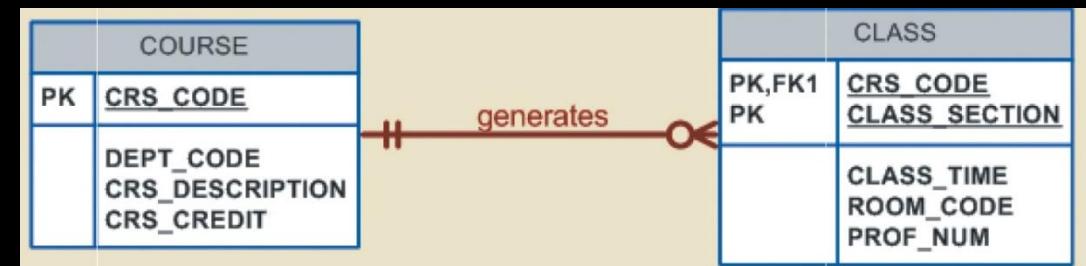
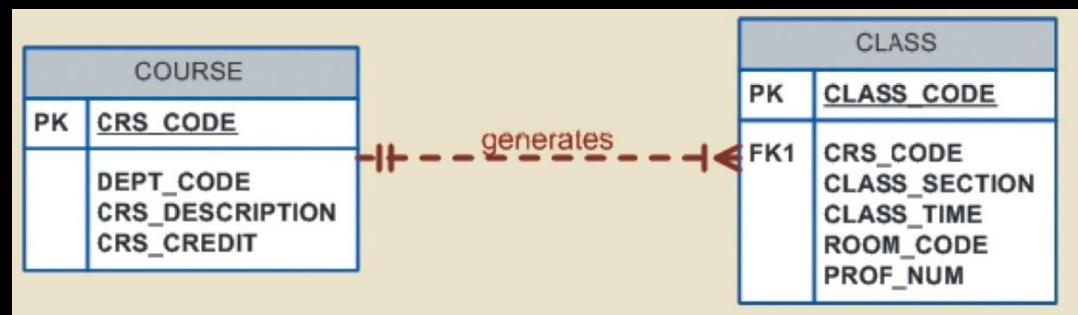
- Strong



- Do not confuse this with relationship strength!

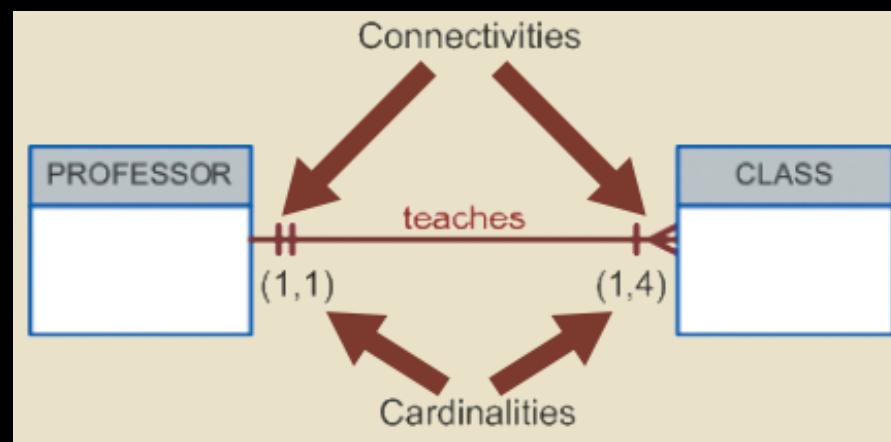
Relationships

- Relationship strength – based on the PK of a related entity
 - Non-identifying (weak) – PK of the related entity does not contain a PK component of the related entity
 - Identifying (strong) – PK of the related entity contains a PK component of the related entity
 - Do not confuse this with entity strength!



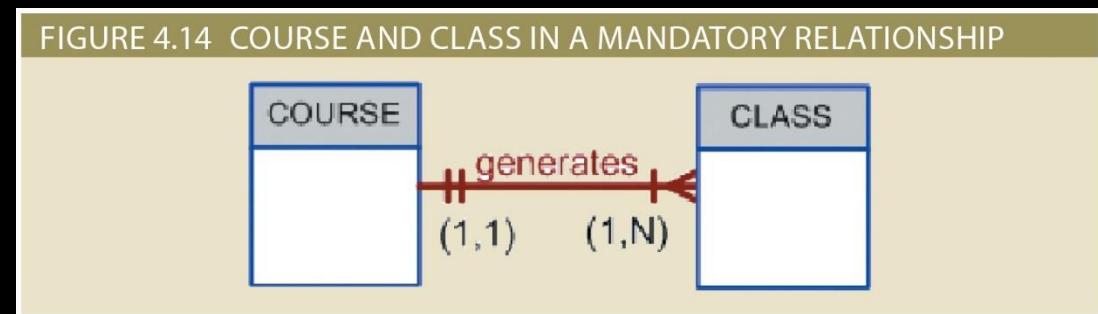
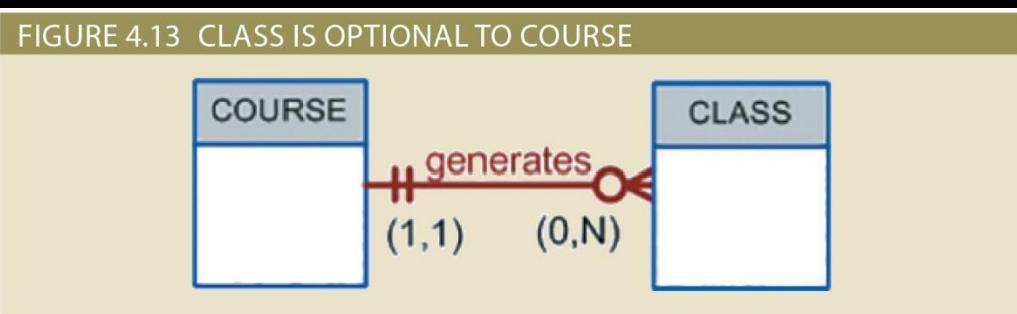
Relationships

- Connectivity – describes the relationship classification
 - See Chapter 3 – (1:1), (1:M), (M:N)
- Cardinality – expresses the min and max number of entity occurrences associated with one occurrence of the related entity



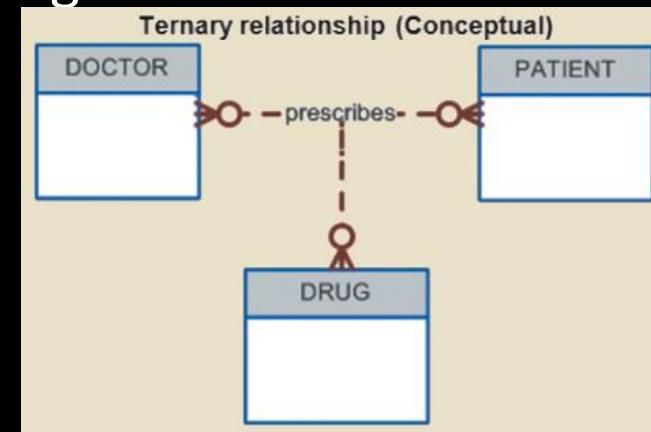
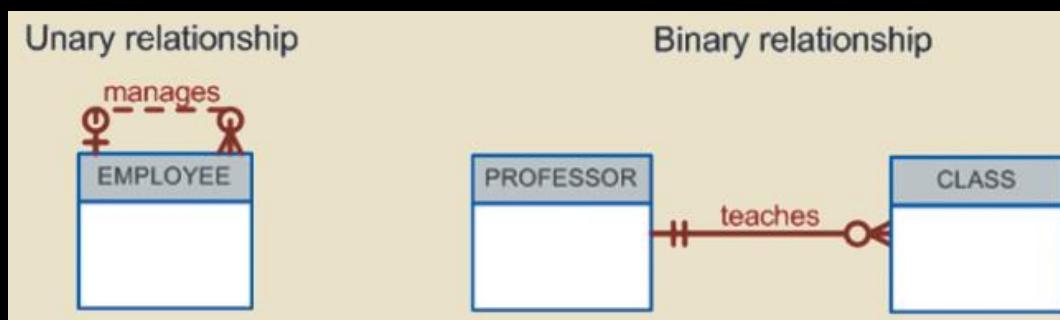
Relationships

- Participation
 - Optional – One entity occurrence does not require a corresponding entity occurrence in the related entity
 - Mandatory - One entity occurrence requires a corresponding entity occurrence in the related entity

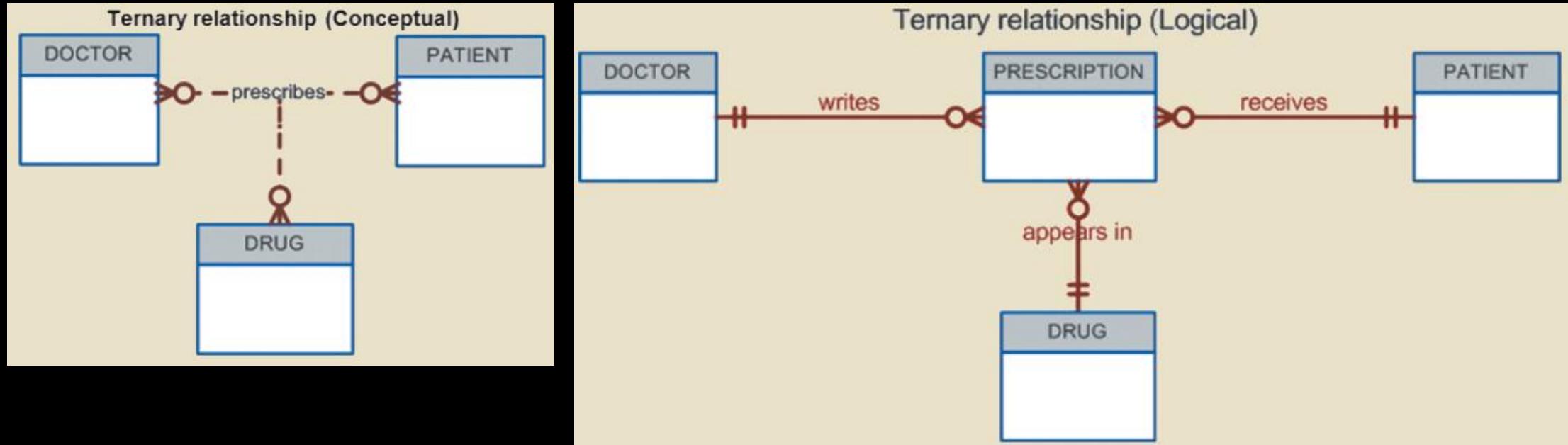


Relationships

- Degree – number of participants associated with a relationship
 - Unary (or recursive) – relationship maintained within a single entity
 - Relationship exists between occurrences of the same entity
 - Binary – relationship maintained between two entities
 - Ternary – relationship maintained among three entities
 - Higher order – relationship maintained among more than three entities

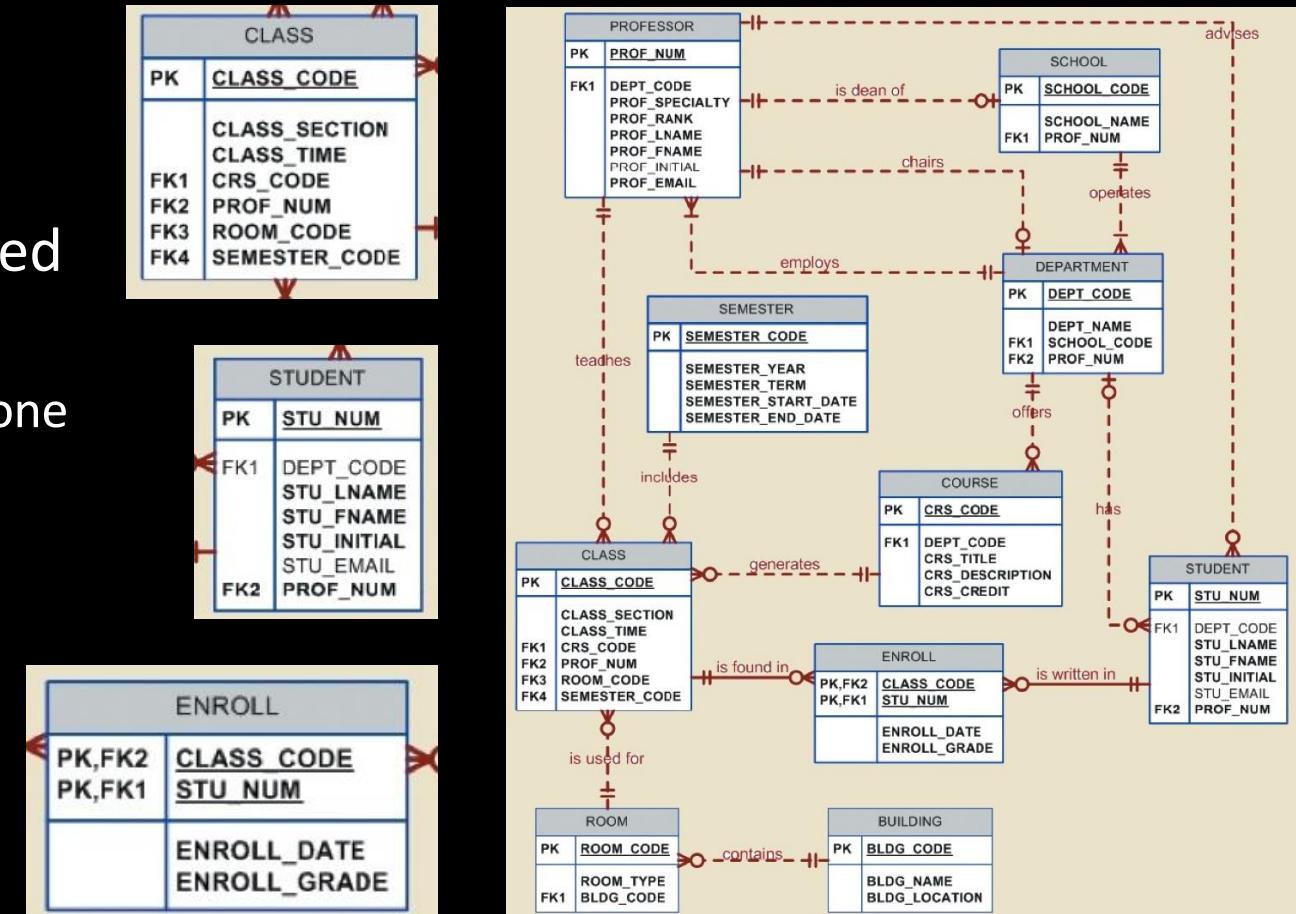


Relationships



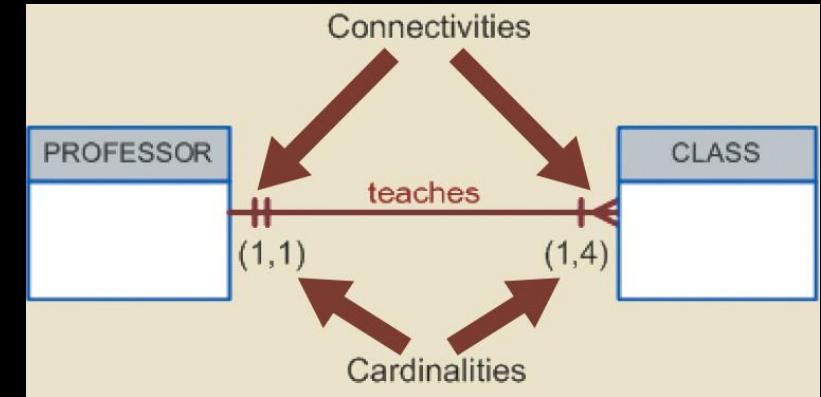
Our ERD Requirements - Entities

- Name at the top
- Attributes
 - PK – PK label, bold, underlined
 - FK
 - FK label with #, if more than one
 - Required – bold
 - Optional – not bold
 - PK component underlined
- Other attributes
 - Required – bold
 - Optional – not bold



Our ERD Requirements - Relationships

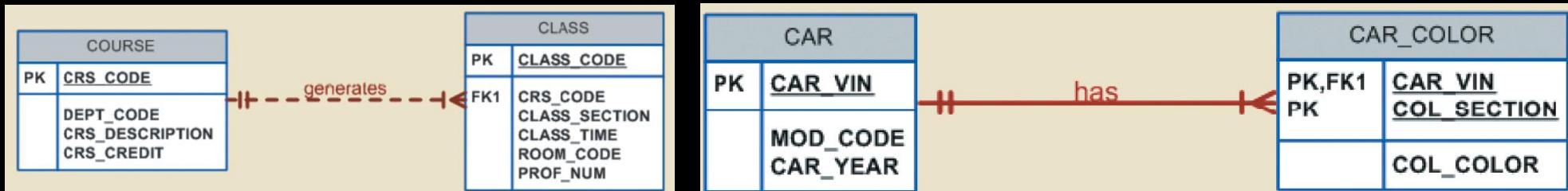
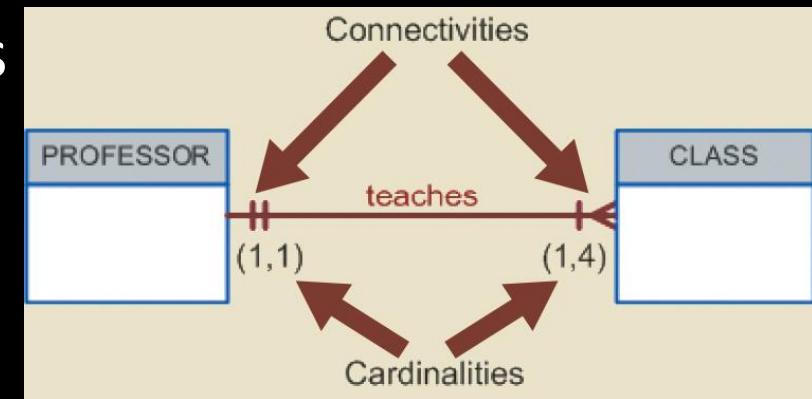
- Labels – verb (typically), concise, intuitive
- Connectivity - O:1, 1:1, O:M, 1:M
 - Must use the correct symbol on both sides
- Participation – Optional or mandatory
 - Must use the correct symbol on both sides



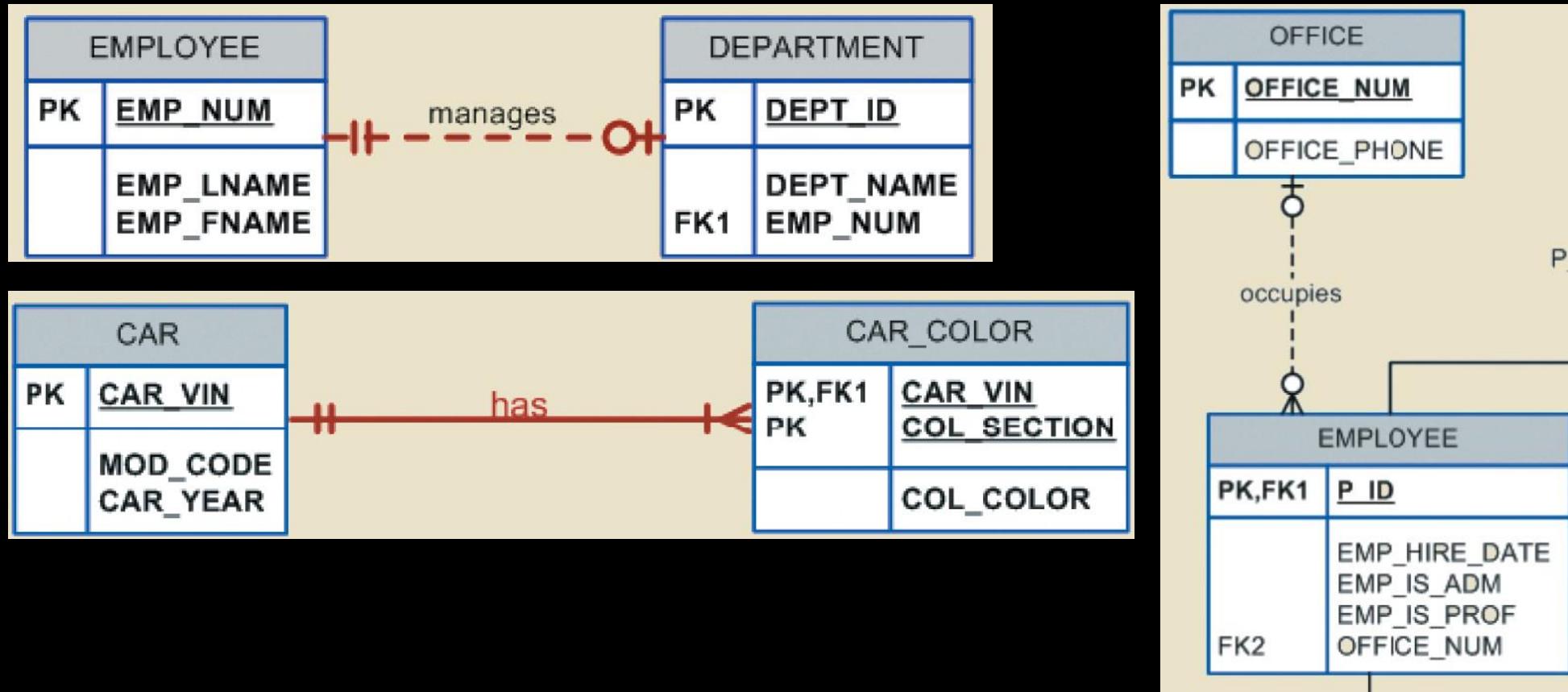
SYMBOL	CARDINALITY	COMMENT
○ ←	(0,N)	Zero or many; the “many” side is optional.
←	(1,N)	One or many; the “many” side is mandatory.
	(1,1)	One and only one; the “1” side is mandatory.
○ ↓	(0,1)	Zero or one; the “1” side is optional.

Our ERD Requirements - Relationships

- Cardinality – number of related occurrences
 - You must have the cardinality, except for those that are covered by the connectivity symbols
- Strength
 - Weak - dashed line
 - Strong - solid line



Our ERD Requirements - Relationships



Our Relational Schema Notation

- Name – typically uppercase
- Parenthesis – holds the attribute list
- Attributes – comma-separated list, typically lowercase
 - PK – bold, underlined
 - FK – italicized and:
 - Required – bold
 - Optional – not bold
- Other attributes
 - Required – bold
 - Optional – not bold

Our Relational Schema Notation

EMPLOYEE(emp_num, emp_name, *emp_dept*)

DEPARTMENT(dept_num, dept_name, *mgr_emp_num*)

CLASS(crs_code, class_section, time, *prof_emp_num*)

COURSE(crs_code, *dept_num*, crs_descr, crs_credit)

- Note that these examples do not necessarily correspond to an ERD from the book or other class material.

Our Relational Schema Notation

- Remember: bold and underline all attributes in a composite PK
- Not all FKs are bolded—it will depend on design specifications
- FKs involved with a composite PK are always bolded—a PK using a FK requires that FK to have a value
- This is an extended notation beyond the book's examples. You may also see other forms of it, but this is what we will use in this class.