SIT103/SIT772 Database Fundamentals

Week 3

Entity Relationship Diagram (ERD)

Dr Iynkaran Natgunanathan,

email: iynkaran.natgunanathan@deakin.edu.au,
Phone: +61 3 924 68825.

Last Week

- Database design – Conceptual and logical design
- Relational model – Entity, Attribute, Relationships, and constraints
- Keys (composite, super, candidate, primary, natural, surrogate, foreign, secondary)
- Integrity Rules – Entity and Referential Integrity
- Entity Relationship Diagram

Last Week's OnTrack Task

- Task 2.1P Database Modelling Tools

  - Basics of relational database modelling

  - To familiarise you with modelling tools

    - LucidChart and MS Visio

Questions?

Any questions/comments so far

Last week's content

OnTrack tasks

Anything in general about the unit

This week

- More on conceptual and logic design

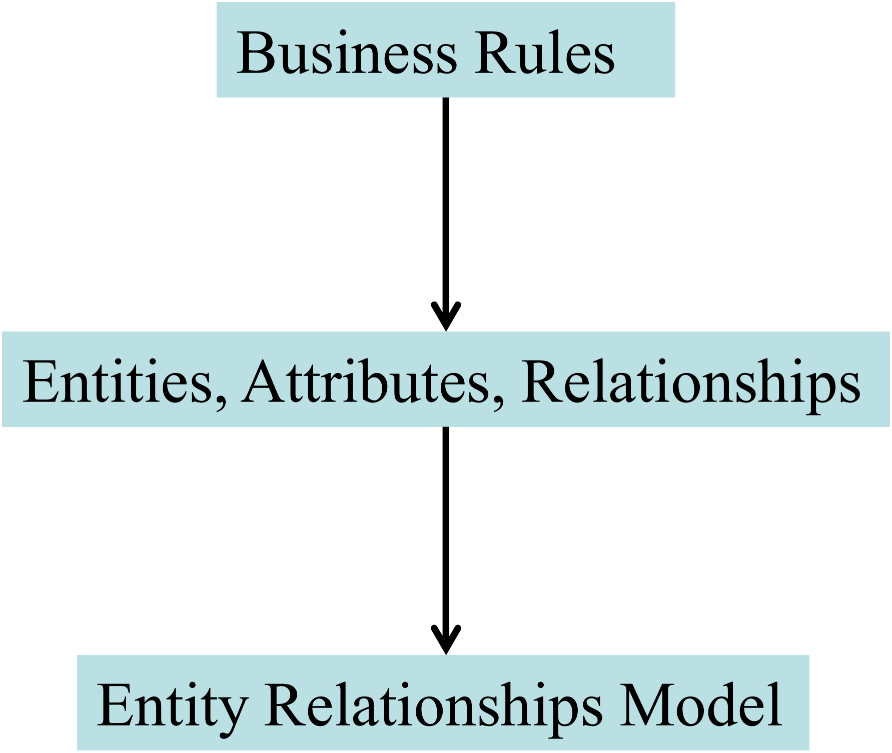
  - Entity Relationship Diagram (ERD)

- Some case studies

Entity Relationships Model (Recap)

- **Entity**: person, place, thing, or event about which data will be collected and stored
  - e.g., Student, Course, Product, Order, Transaction, etc.
- **Attribute**: characteristic of an entity
  - e.g., ID, Name, DoB, Address, etc.
- **Relationship**: association among entities
  - One-to-one (1:1 OR 1..1)
  - One-to-many (1:M OR 1..*)
  - Many-to-many (M:N OR *..*)
- **Constraint**: restriction placed on data
  - Ensures data integrity, e.g., Unique, Not NULL, etc.

Entity Relationships Model (2)



Business Rules

Entities, Attributes, Relationships

Entity Relationships Model

Attributes and domain

- **Required attribute**: must have a value (NOT NULL)
- **Optional attribute**: may be left empty (NULL is allowed)
- **Domain**: set of possible values for an attribute
  - States of Australia - {ACT, NT, NSW, QLD, SA, TAS, VIC, WA}
  - Height, Weight – decimal numbers
  - Age – integer numbers

Types of Attributes

- **Composite attribute** – ***can*** be further ***subdivided***

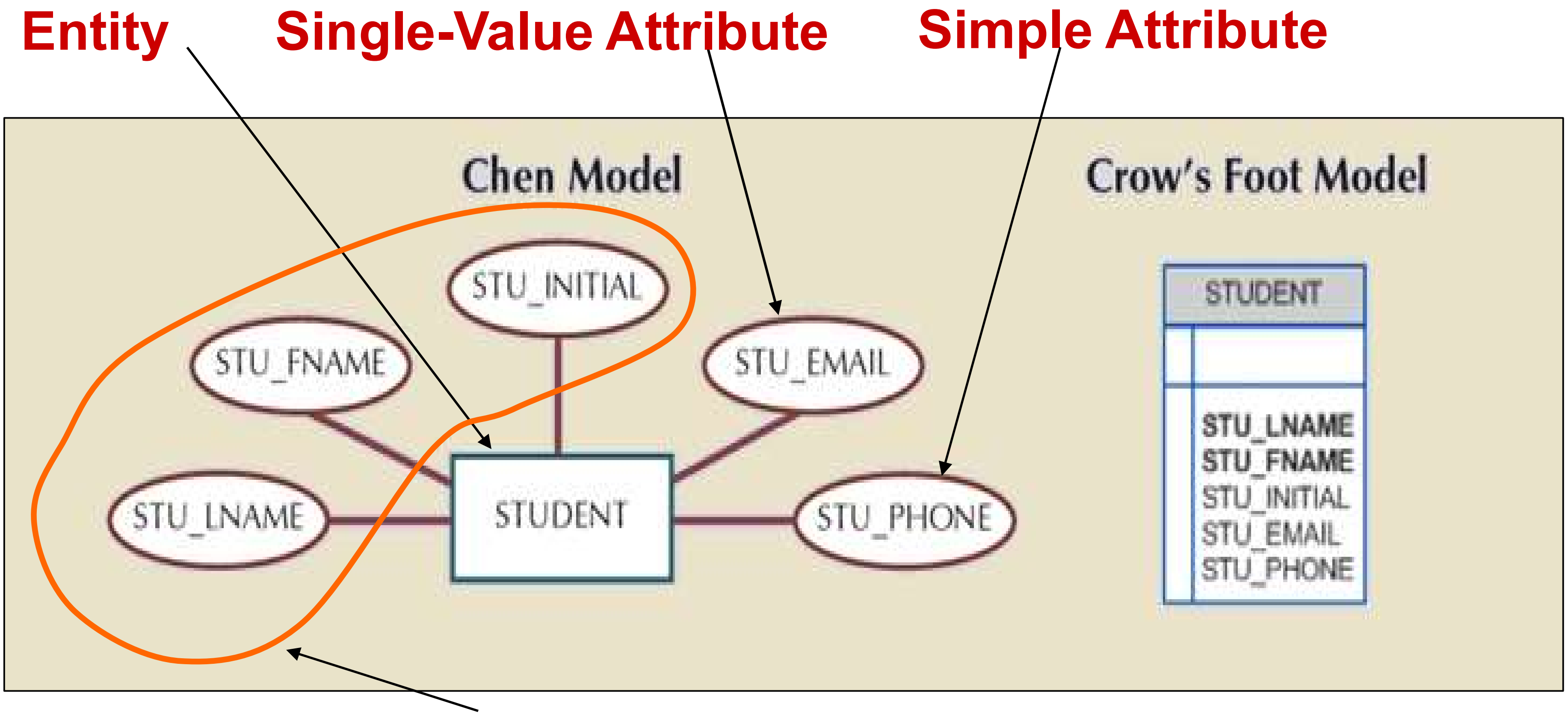
- **Simple attribute –** ***cannot*** be further ***subdivided***

- **Single-valued attribute** – can only have a ***single value*** at a particular instance of time

  e.g., a person has one weight

- **Multi-valued attribute** – can have ***many values***, *e.g.*, a person can have several aliases, or multiple contact numbers or several skills, or several qualifications, and so on

Types of Attributes (2)



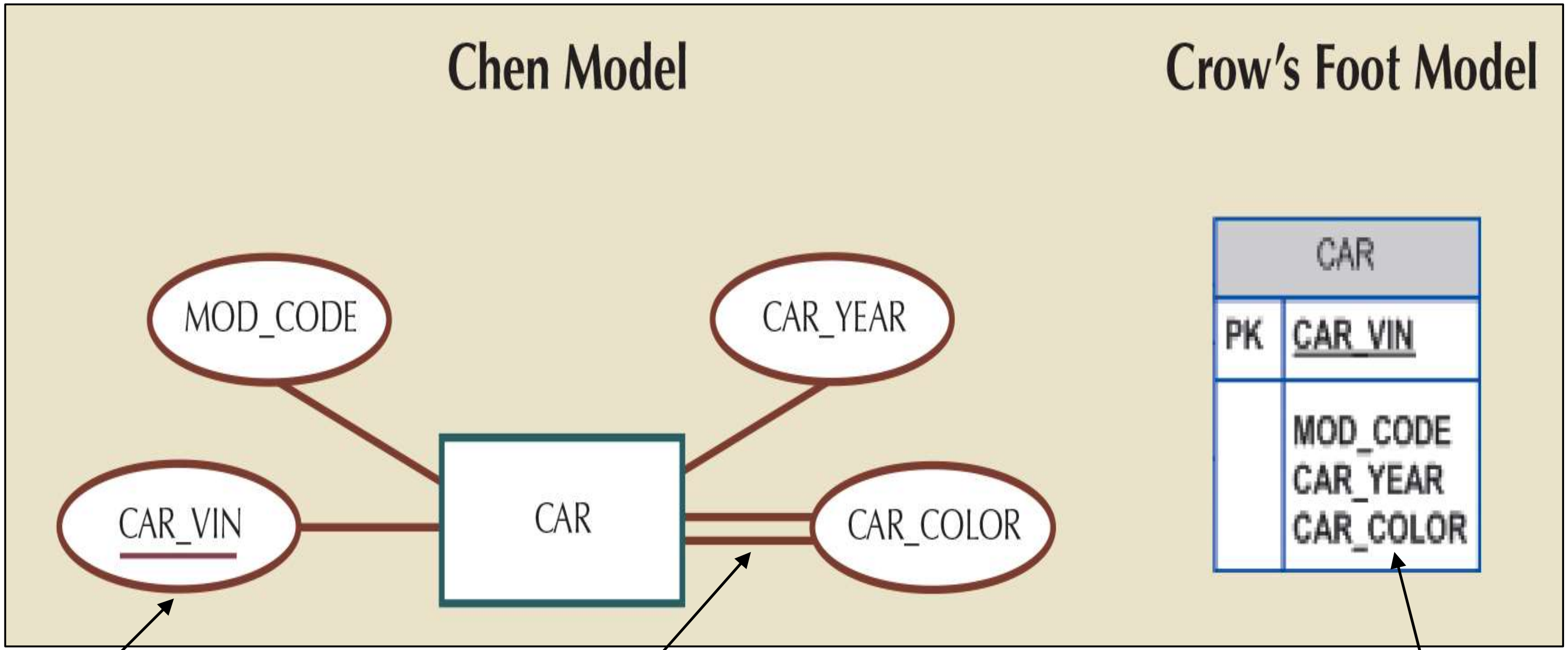
Composite Attribute STU_NAME can be subdivided into these three simple attributes

Textbook
Figure 4.1

Multi-valued Attribute

Car's color – body color, roof color, and trim color

Textbook Figure 4.3



Primary Key

A double line denotes a multi-valued attribute

Multi-valued attribute is not differentiated

Multi-valued Attribute (2)

- Problematic – we should avoid them

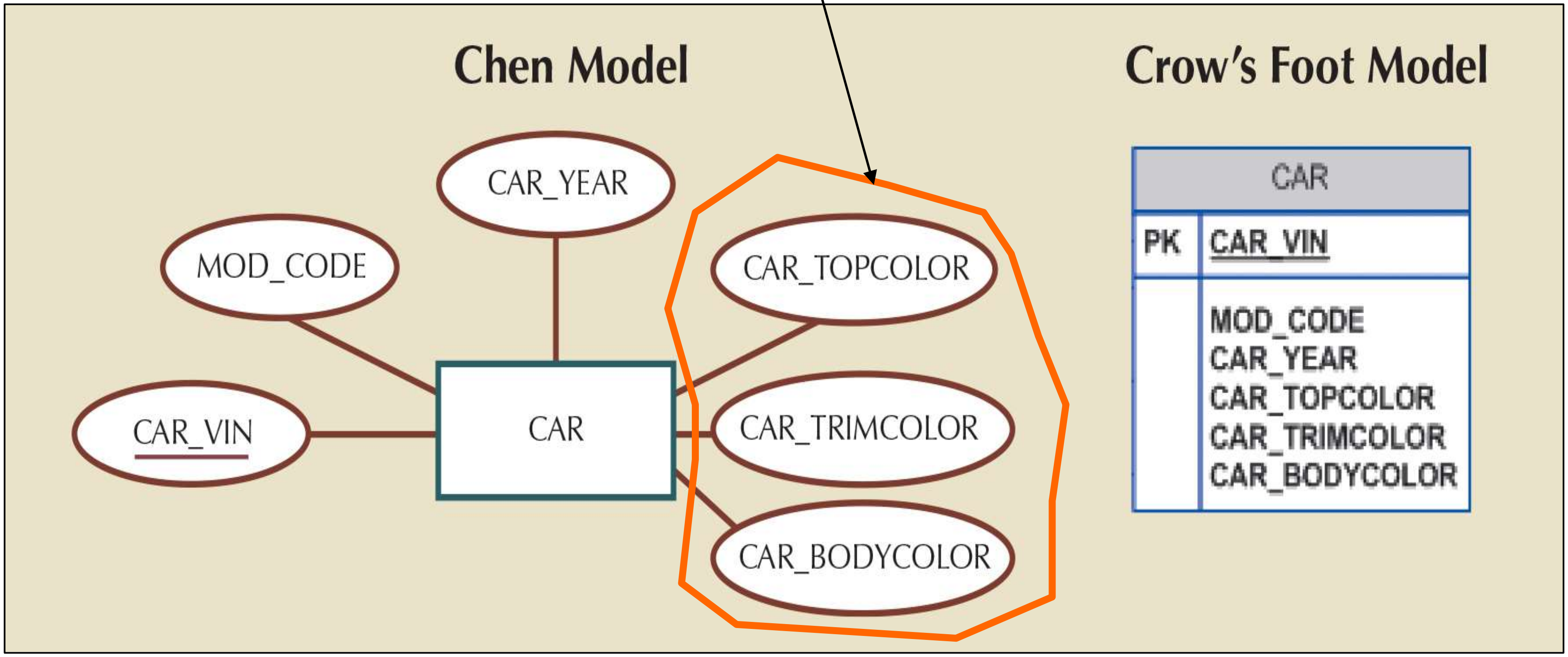
  - Remember, in relational model

  "Each row and column intersection must represent a single data value"

- Two possible solutions
  - **Create new attributes** one for each of the original multi-valued attribute's components
  - **Create a new entity** composed of original multi-valued attribute's components

Multi-valued Attribute (3)

Splitting the multi-valued attribute (CAR_COLOR) into three attributes



CAR	
PK	CAR_VIN
	MOD_CODE CAR_YEAR CAR_TOPCOLOR CAR_TRIMCOLOR CAR_BODYCOLOR

Textbook
Figure 4.4

Multi-valued Attribute (4)

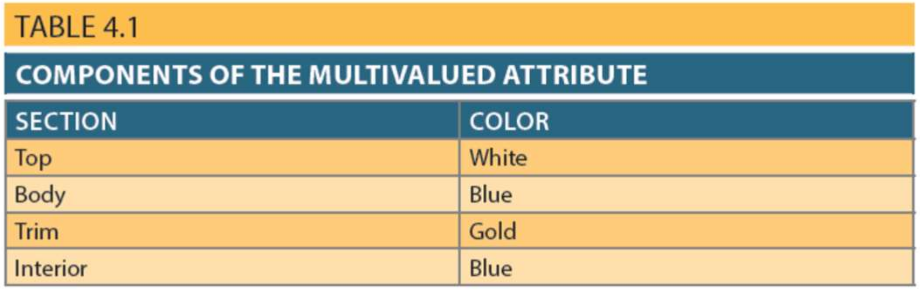
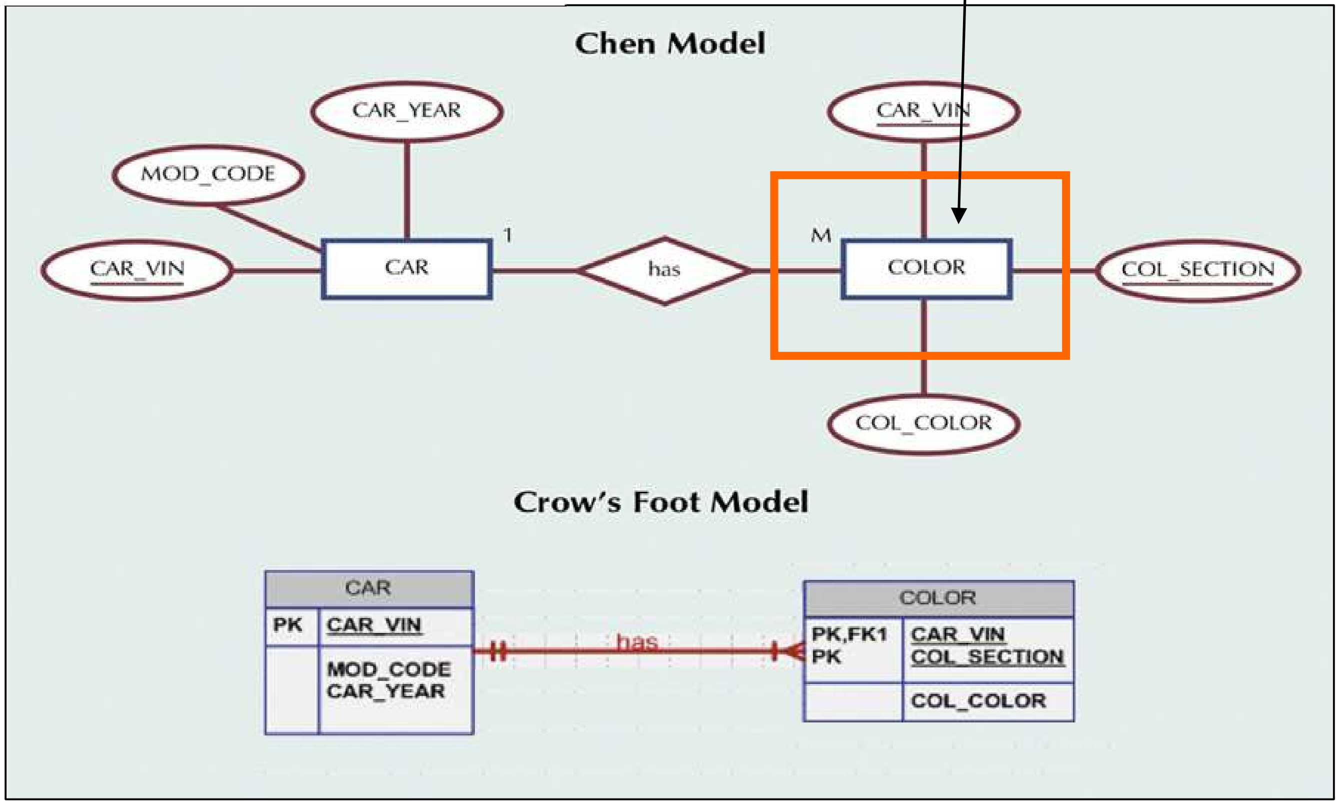


TABLE 4.1	
COMPONENTS OF THE MULTIVALUED ATTRIBUTE	
SECTION	COLOR
Top	White
Body	Blue
Trim	Gold
Interior	Blue

Textbook Table 4.1

A new entity from a multi-valued attribute



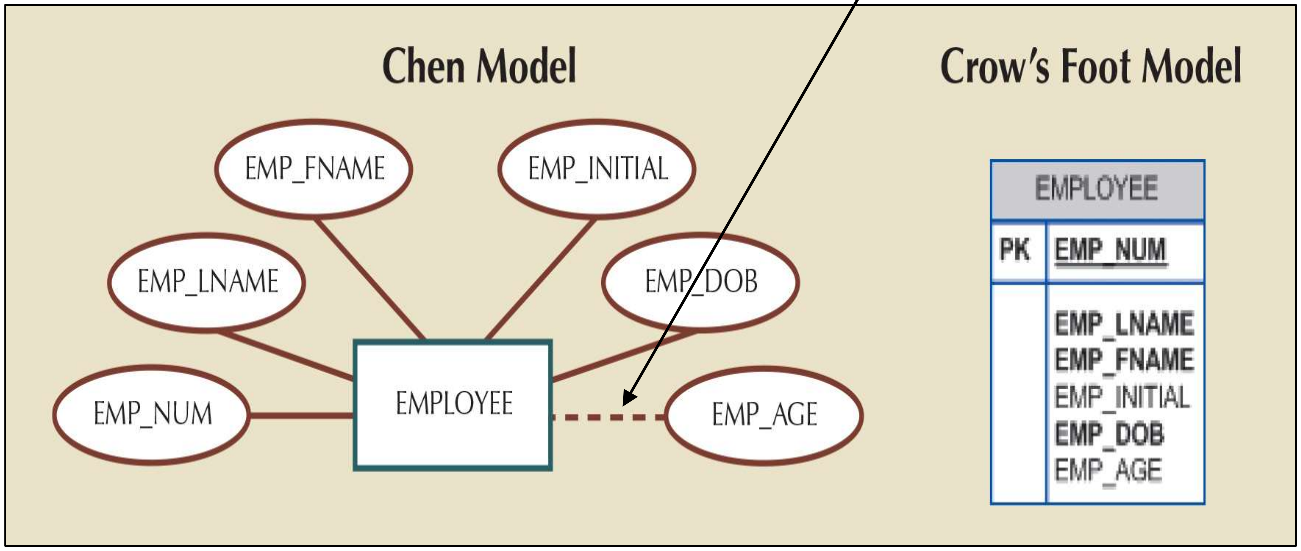
Textbook Figure 4.5 (bottom figure)

Derivable Attributes

- An attribute whose value may be calculated (derived) from other attributes
  - **<u>Need not be physically stored</u>** in the DB
  - **<u>Can be derived</u>** when needed

A dashed line also depicts a derived attribute, alternative symbol in Chen's notation

Textbook Figure 4.6



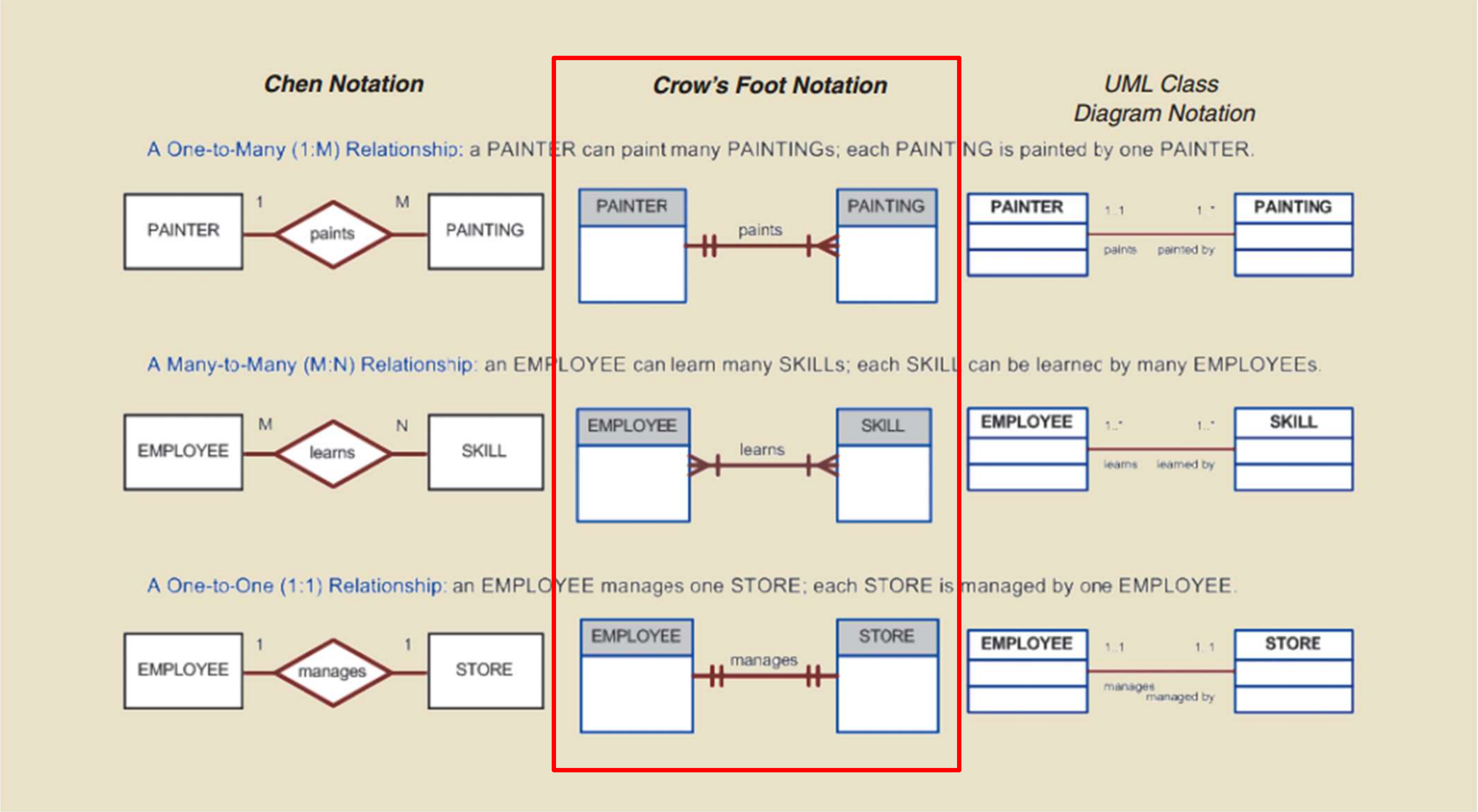
Should we store derived attributes?

Table 4.2	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

Textbook Table 4.2

ERD Notations (Recap)

FIGURE 2.3 THE ER MODEL NOTATIONS



A One-to-Many (1:M) Relationship: a PAINTER can paint many PAINTINGs; each PAINTING is painted by one PAINTER.

A Many-to-Many (M:N) Relationship: an EMPLOYEE can learn many SKILLs; each SKILL can be learned by many EMPLOYEEs.

A One-to-One (1:1) Relationship: an EMPLOYEE manages one STORE; each STORE is managed by one EMPLOYEE.

Textbook Figure 2.3

ERD: Relationships

- **Mandatory vs Optional relationships**

Textbook Table 4.3

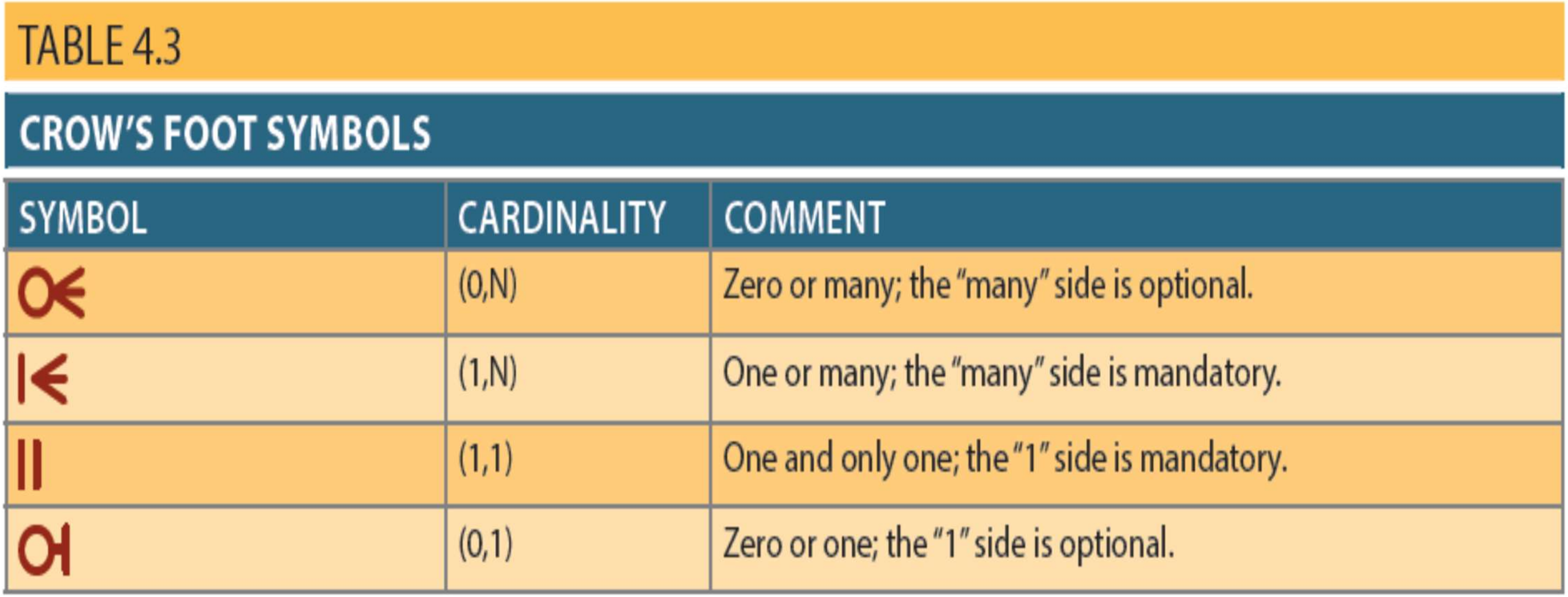
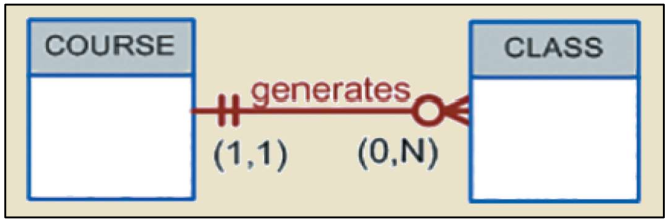


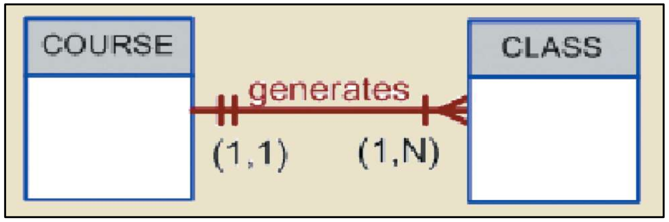
TABLE 4.3

CROW'S FOOT SYMBOLS

SYMBOL	CARDINALITY	COMMENT
O<	(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.
O|	(0,1)	Zero or one; the "1" side is optional.

Textbook Figure 4.13





Textbook Figure 4.14

ERD: Relationship Strength

- **Weak (non-identifying) relationship**
  - Primary key of the related entity does not contain a primary key component of the other entity
  - The relationship is denoted using dashed line in the ERD
- **Strong (identifying) relationships**
  - Primary key of the related entity contains a primary key component of the other entity
  - A relationship that occurs when two entities are existence dependent
  - The relationship is denoted using solid line in the ERD

Weak Relationship

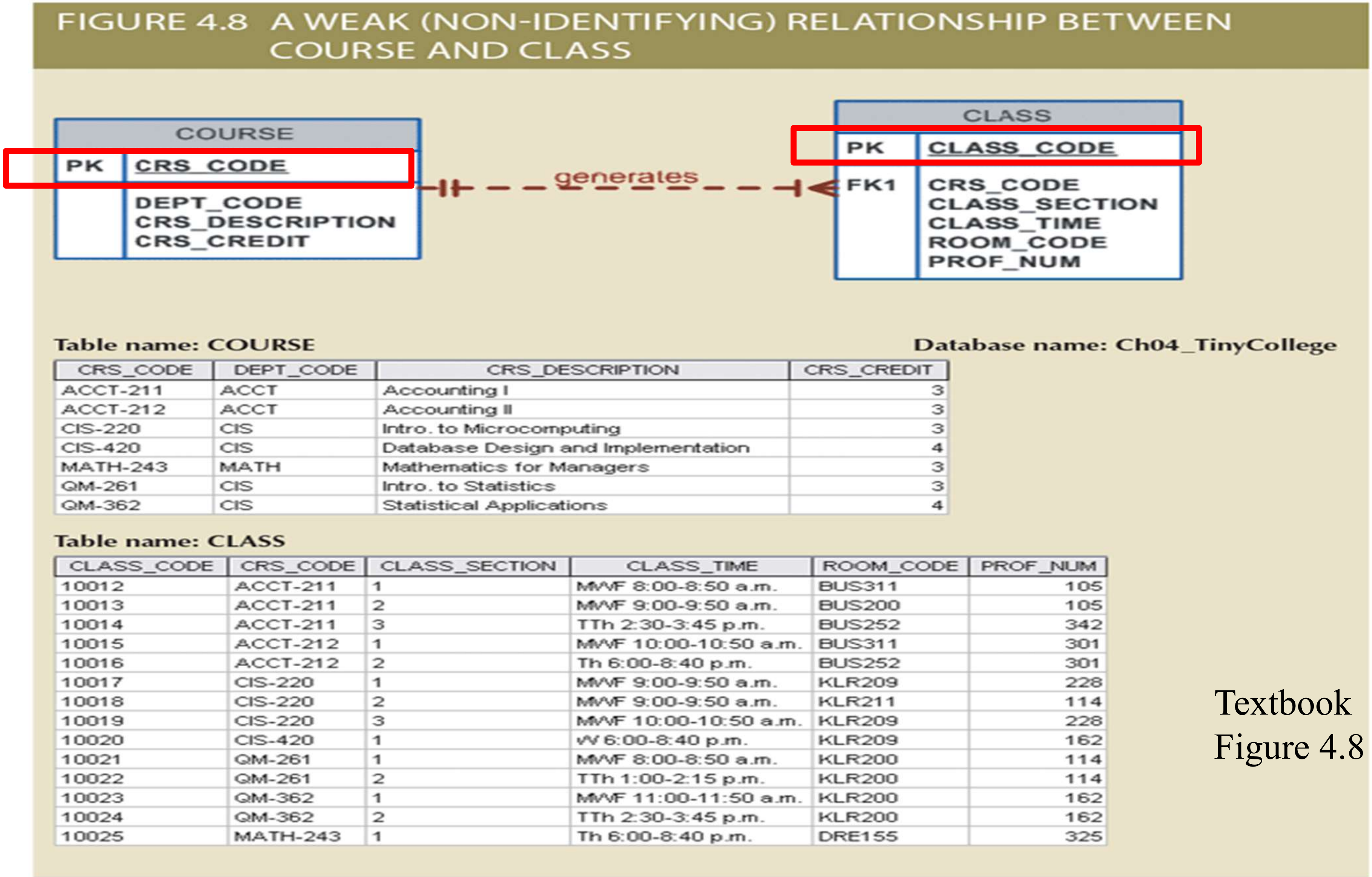


FIGURE 4.8 A WEAK (NON-IDENTIFYING) RELATIONSHIP BETWEEN COURSE AND CLASS

COURSE

PK CRS_CODE

DEPT_CODE
CRS_DESCRIPTION
CRS_CREDIT

generates

CLASS

PK CLASS_CODE

FK1 CRS_CODE
CLASS_SECTION
CLASS_TIME
ROOM_CODE
PROF_NUM

Table name: COURSE **Database name: Ch04_TinyCollege**

CRS_CODE	DEPT_CODE	CRS_DESCRIPTION	CRS_CREDIT
ACCT-211	ACCT	Accounting I	3
ACCT-212	ACCT	Accounting II	3
CIS-220	CIS	Intro. to Microcomputing	3
CIS-420	CIS	Database Design and Implementation	4
MATH-243	MATH	Mathematics for Managers	3
QM-261	CIS	Intro. to Statistics	3
QM-362	CIS	Statistical Applications	4

Table name: CLASS

CLASS_CODE	CRS_CODE	CLASS_SECTION	CLASS_TIME	ROOM_CODE	PROF_NUM
10012	ACCT-211	1	MWF 8:00-8:50 a.m.	BUS311	105
10013	ACCT-211	2	MWF 9:00-9:50 a.m.	BUS200	105
10014	ACCT-211	3	TTh 2:30-3:45 p.m.	BUS252	342
10015	ACCT-212	1	MWF 10:00-10:50 a.m.	BUS311	301
10016	ACCT-212	2	Th 6:00-8:40 p.m.	BUS252	301
10017	CIS-220	1	MWF 9:00-9:50 a.m.	KLR209	228
10018	CIS-220	2	MWF 9:00-9:50 a.m.	KLR211	114
10019	CIS-220	3	MWF 10:00-10:50 a.m.	KLR209	228
10020	CIS-420	1	W 6:00-8:40 p.m.	KLR209	162
10021	QM-261	1	MWF 8:00-8:50 a.m.	KLR200	114
10022	QM-261	2	TTh 1:00-2:15 p.m.	KLR200	114
10023	QM-362	1	MWF 11:00-11:50 a.m.	KLR200	162
10024	QM-362	2	TTh 2:30-3:45 p.m.	KLR200	162
10025	MATH-243	1	Th 6:00-8:40 p.m.	DRE155	325

Textbook Figure 4.8

Strong Relationship

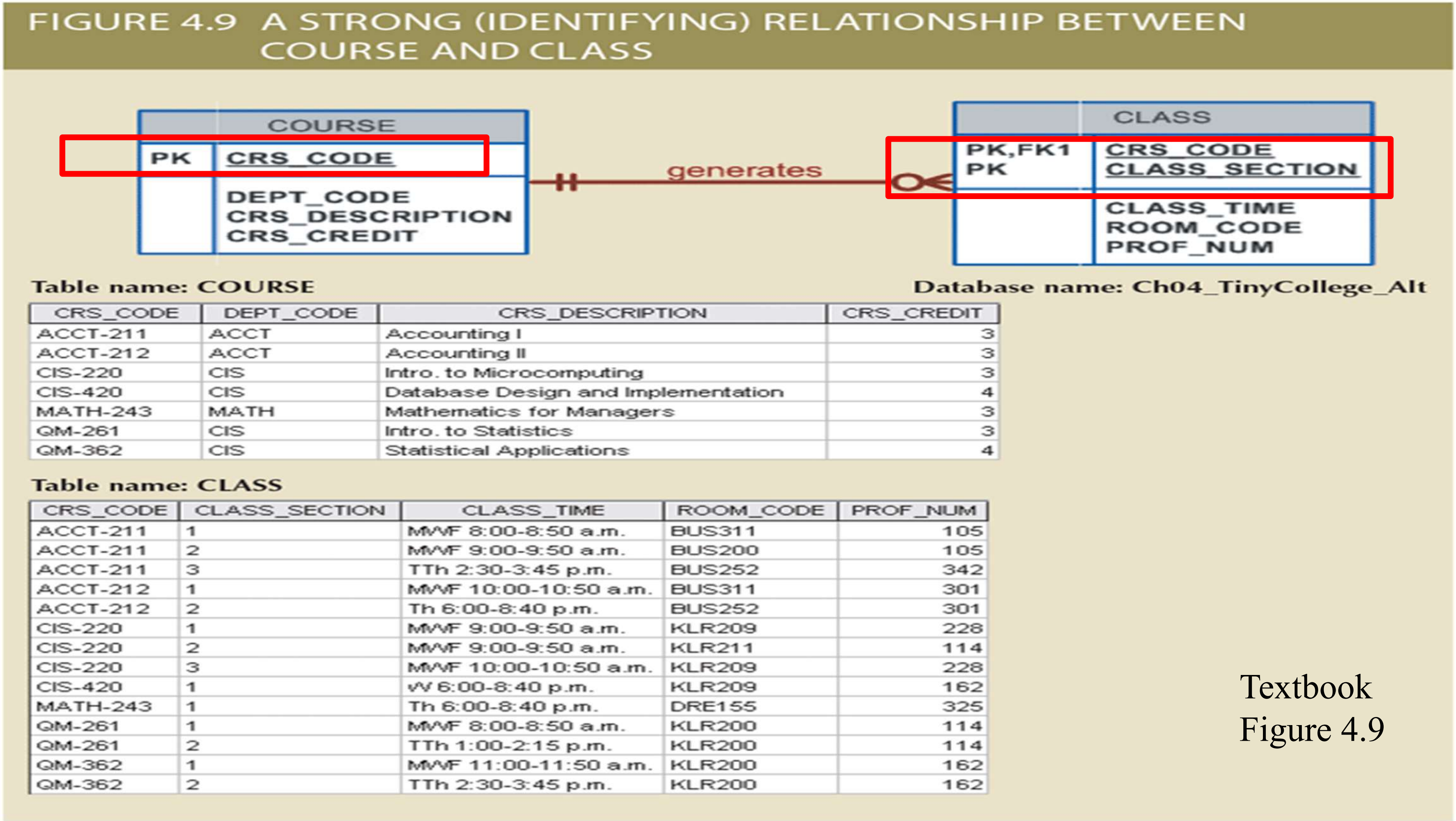


FIGURE 4.9 A STRONG (IDENTIFYING) RELATIONSHIP BETWEEN COURSE AND CLASS

Table name: COURSE

Database name: Ch04_TinyCollege_Alt

CRS_CODE	DEPT_CODE	CRS_DESCRIPTION	CRS_CREDIT
ACCT-211	ACCT	Accounting I	3
ACCT-212	ACCT	Accounting II	3
CIS-220	CIS	Intro. to Microcomputing	3
CIS-420	CIS	Database Design and Implementation	4
MATH-243	MATH	Mathematics for Managers	3
QM-261	CIS	Intro. to Statistics	3
QM-362	CIS	Statistical Applications	4

Table name: CLASS

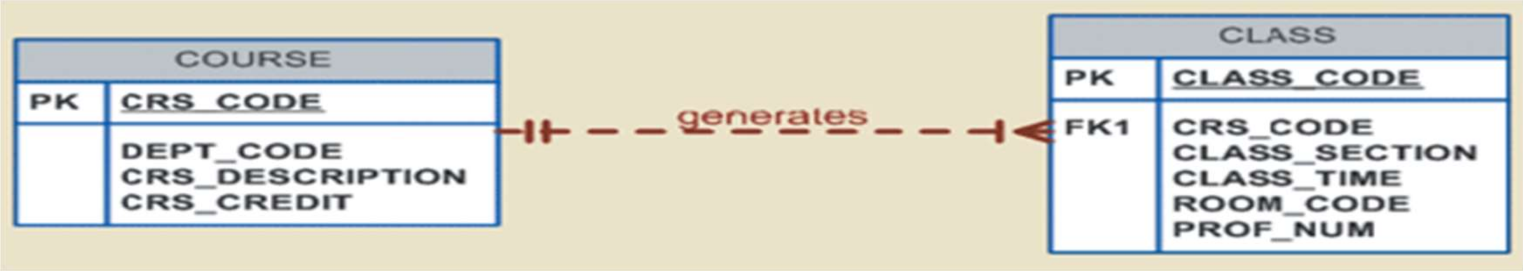
CRS_CODE	CLASS_SECTION	CLASS_TIME	ROOM_CODE	PROF_NUM
ACCT-211	1	MWF 8:00-8:50 a.m.	BUS311	105
ACCT-211	2	MWF 9:00-9:50 a.m.	BUS200	105
ACCT-211	3	TTh 2:30-3:45 p.m.	BUS252	342
ACCT-212	1	MWF 10:00-10:50 a.m.	BUS311	301
ACCT-212	2	Th 6:00-8:40 p.m.	BUS252	301
CIS-220	1	MWF 9:00-9:50 a.m.	KLR209	228
CIS-220	2	MWF 9:00-9:50 a.m.	KLR211	114
CIS-220	3	MWF 10:00-10:50 a.m.	KLR209	228
CIS-420	1	W 6:00-8:40 p.m.	KLR209	162
MATH-243	1	Th 6:00-8:40 p.m.	DRE155	325
QM-261	1	MWF 8:00-8:50 a.m.	KLR200	114
QM-261	2	TTh 1:00-2:15 p.m.	KLR200	114
QM-362	1	MWF 11:00-11:50 a.m.	KLR200	162
QM-362	2	TTh 2:30-3:45 p.m.	KLR200	162

Textbook
Figure 4.9

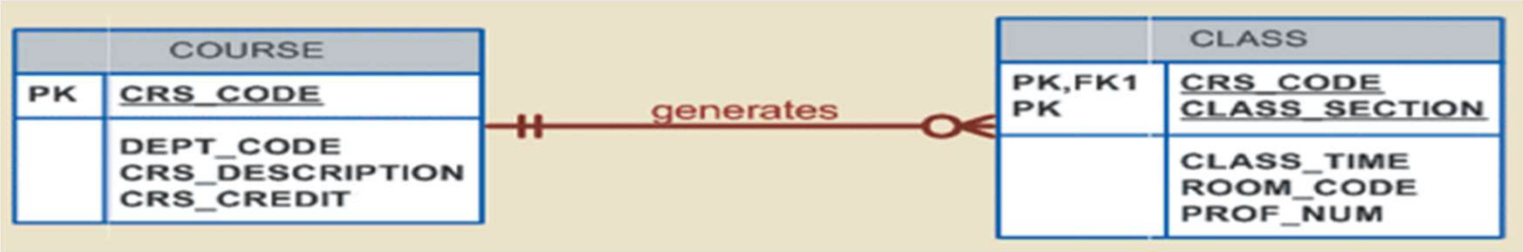
Strong vs Weak Relationships

Textbook
Figure 4.8

Weak Relationship



Strong Relationship



Textbook
Figure 4.9

- **Entity and Referential Integrity**

  - Primary key: NOT NULL

  - Foreign key: Can be NULL

Relationship can be strong/weak depending on how primary keys are defined in entities

A class record can be created without course information – Weak relationship

A class record can’t be created without course information – Strong relationship

Existence Dependence

- **Existence dependence**
  - Entity exists in the database only when it is associated with another related entity occurrence
- **Existence independence**
  - Entity exists apart from all of its related entities
  - Referred to as **a strong entity or regular entity**

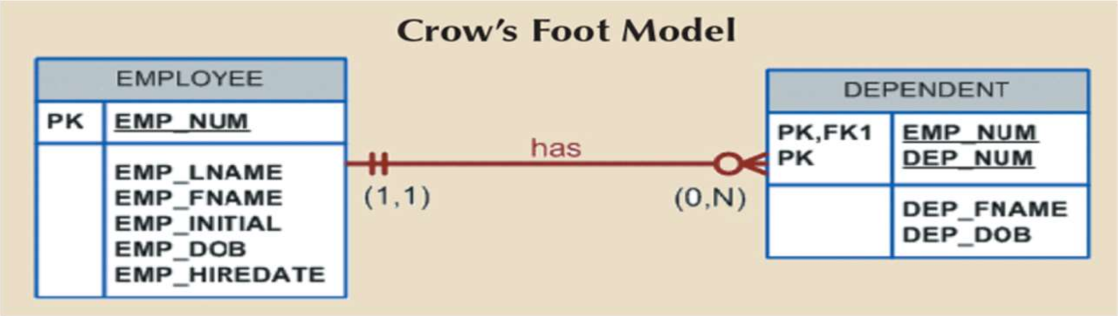
Weak Entities

- **Conditions of a weak entity**
  - Existence-dependent
  - Has a primary key that is partially or totally derived from parent entity in the relationship
  - **Condition of strong relationships**

Weak entity has a strong relationships with another entity

Weak Entities

- **A company insurance policy insures an employee and any dependents**



Textbook
Figure 4.10

FIGURE 4.11 A WEAK ENTITY IN A STRONG RELATIONSHIP

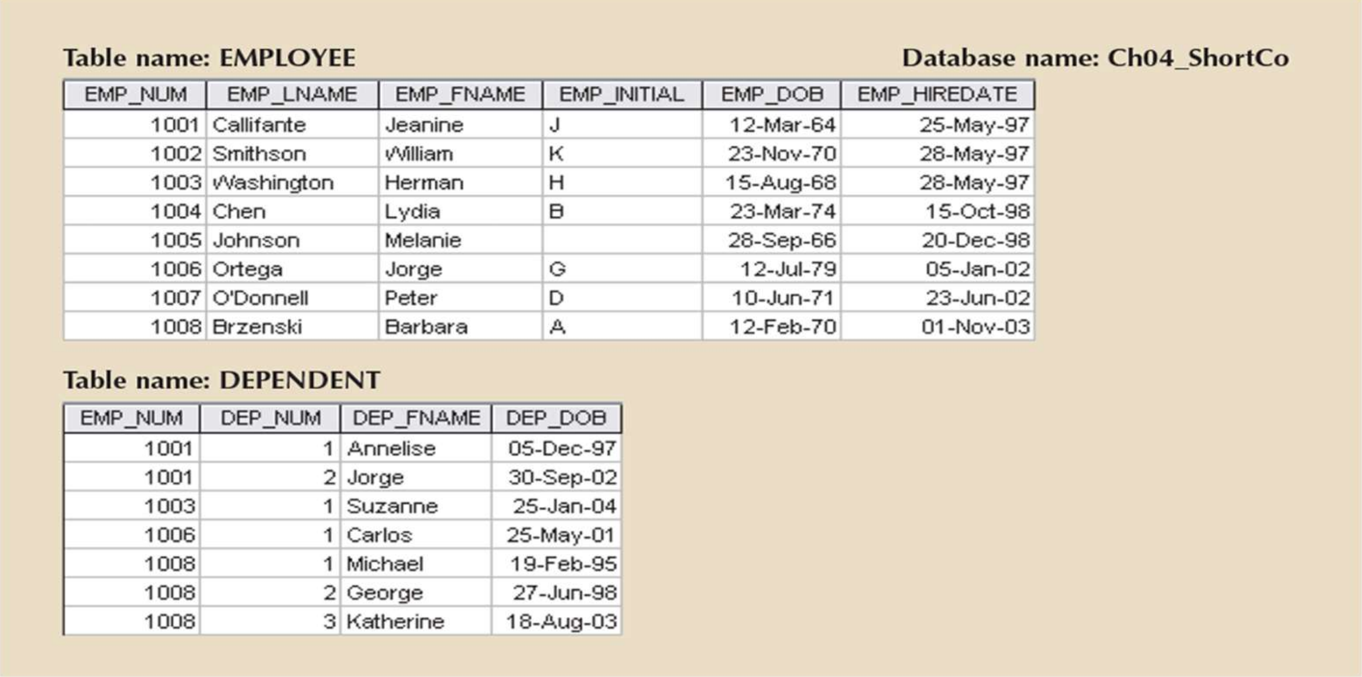


Table name: EMPLOYEE **Database name: Ch04_ShortCo**

EMP_NUM	EMP_LNAME	EMP_FNAME	EMP_INITIAL	EMP_DOB	EMP_HIREDATE
1001	Callifante	Jeanine	J	12-Mar-64	25-May-97
1002	Smithson	William	K	23-Nov-70	28-May-97
1003	Washington	Herman	H	15-Aug-68	28-May-97
1004	Chen	Lydia	B	23-Mar-74	15-Oct-98
1005	Johnson	Melanie		28-Sep-66	20-Dec-98
1006	Ortega	Jorge	G	12-Jul-79	05-Jan-02
1007	O'Donnell	Peter	D	10-Jun-71	23-Jun-02
1008	Brzenski	Barbara	A	12-Feb-70	01-Nov-03

Table name: DEPENDENT

EMP_NUM	DEP_NUM	DEP_FNAME	DEP_DOB
1001	1	Annelise	05-Dec-97
1001	2	Jorge	30-Sep-02
1003	1	Suzanne	25-Jan-04
1006	1	Carlos	25-May-01
1008	1	Michael	19-Feb-95
1008	2	George	27-Jun-98
1008	3	Katherine	18-Aug-03

Textbook
Figure 4.11

Implementing Relationships

1:1 Relationship:

PK of one entity as a FK in another entity

Textbook
Figure 5.8

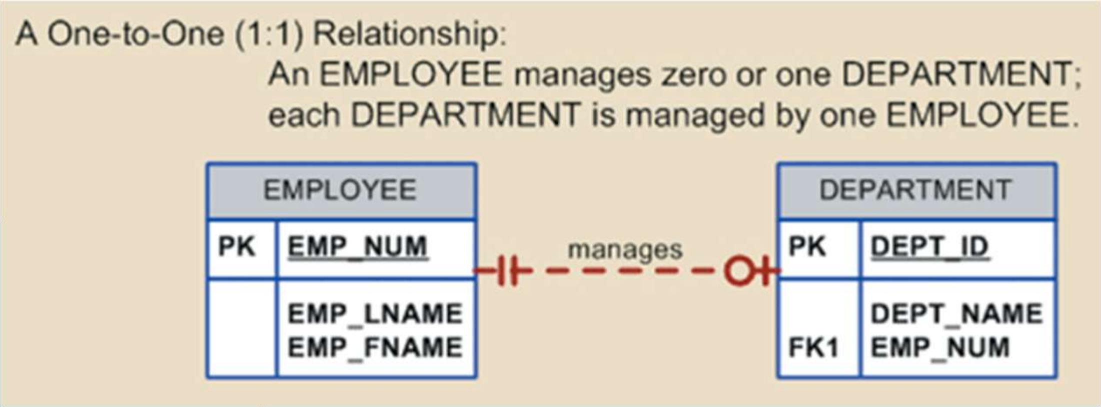
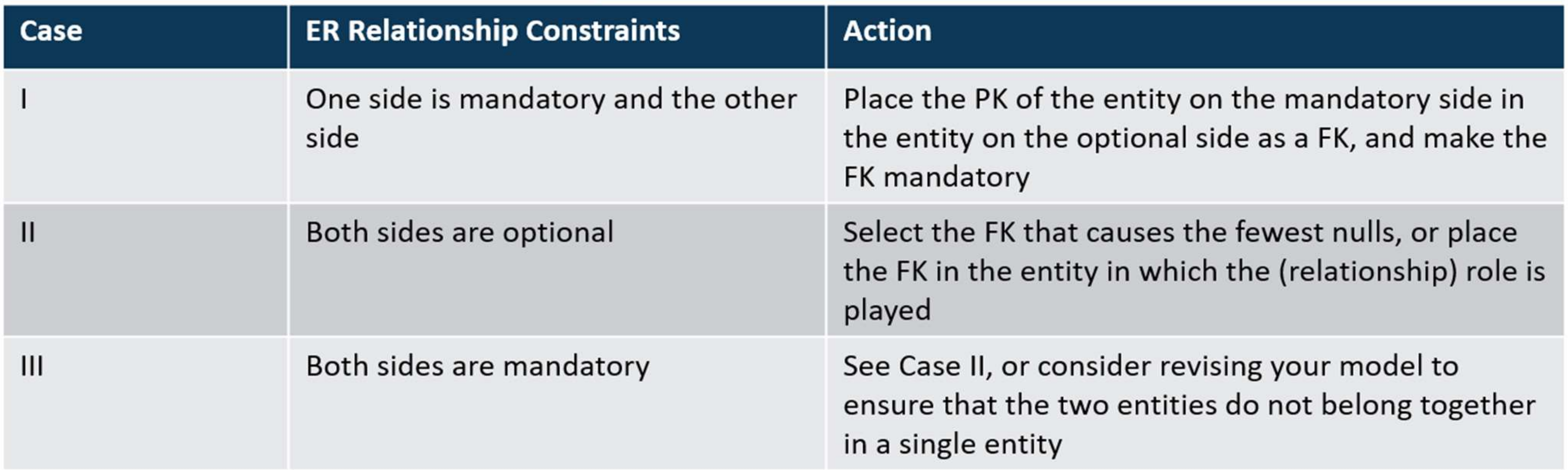

A One-to-One (1:1) Relationship:
An EMPLOYEE manages zero or one DEPARTMENT;
each DEPARTMENT is managed by one EMPLOYEE.

Table 5.5: Selection of Foreign Key in a 1:1 Relationship



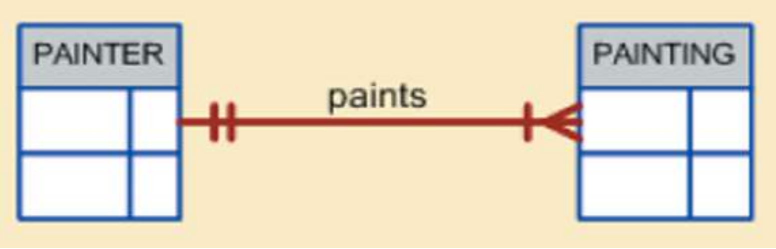
Case	ER Relationship Constraints	Action
I	One side is mandatory and the other side	Place the PK of the entity on the mandatory side in the entity on the optional side as a FK, and make the FK mandatory
II	Both sides are optional	Select the FK that causes the fewest nulls, or place the FK in the entity in which the (relationship) role is played
III	Both sides are mandatory	See Case II, or consider revising your model to ensure that the two entities do not belong together in a single entity

Implementing Relationships (2)

1:M Relationship:

PK of the "1" side in the table of the "M" side as a FK

Textbook Figure 3.17



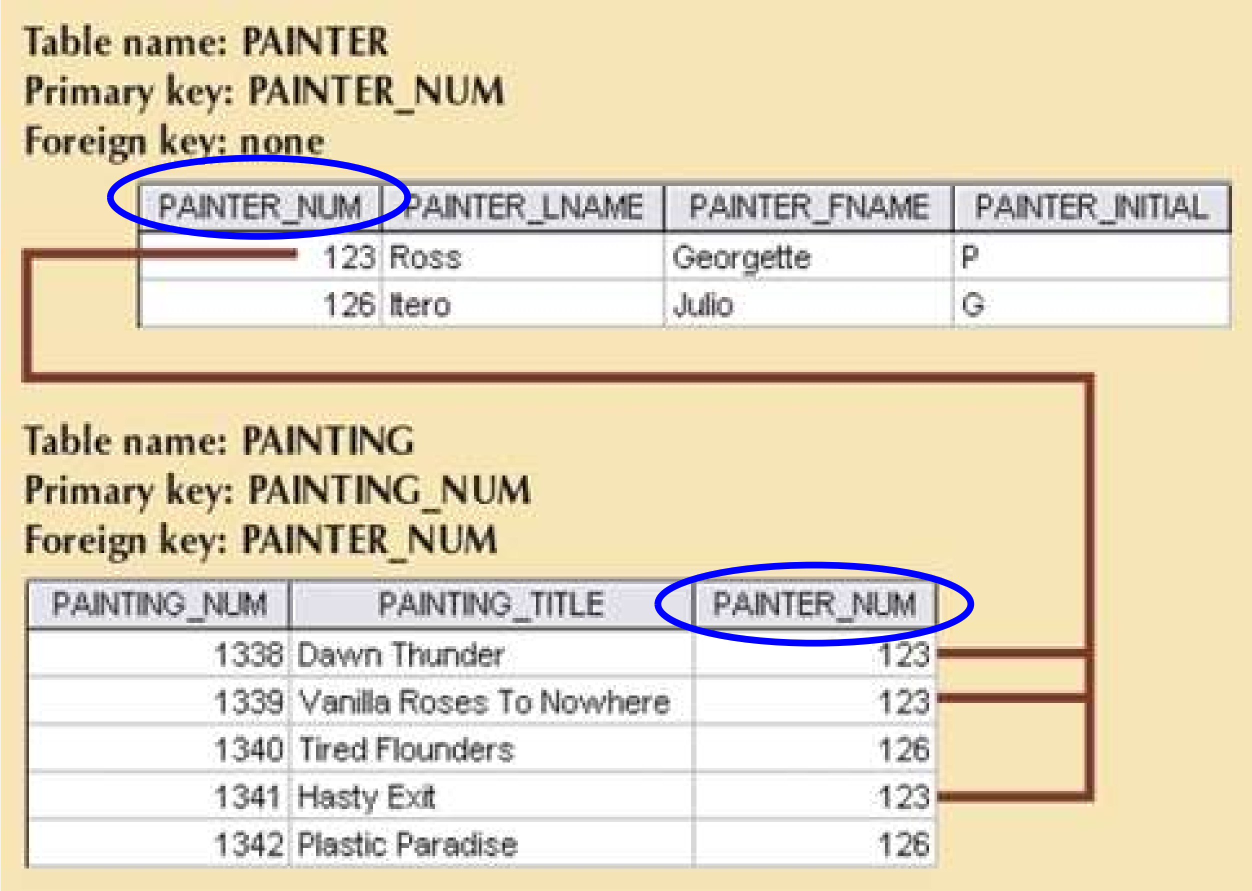


Table name: PAINTER
Primary key: PAINTER_NUM
Foreign key: none

PAINTER_NUM	PAINTER_LNAME	PAINTER_FNAME	PAINTER_INITIAL
123	Ross	Georgette	P
126	Itero	Julio	G

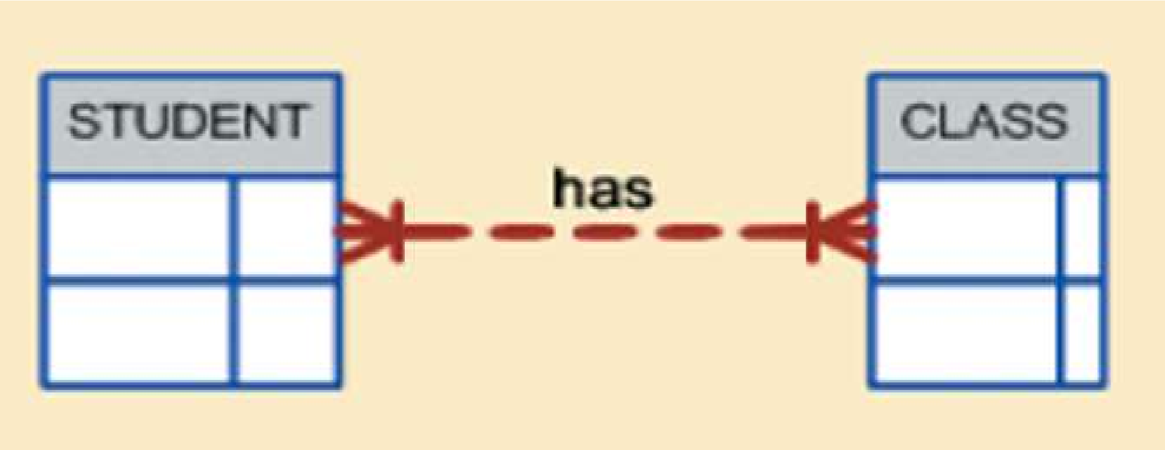
Table name: PAINTING
Primary key: PAINTING_NUM
Foreign key: PAINTER_NUM

PAINTING_NUM	PAINTING_TITLE	PAINTER_NUM
1338	Dawn Thunder	123
1339	Vanilla Roses To Nowhere	123
1340	Tired Flounders	126
1341	Hasty Exit	123
1342	Plastic Paradise	126

Textbook Figure 3.18

Implementing Relationships (3)

M:N Relationship:



Textbook
Figure 3.23

Any idea how can we implement it?

Do you think it can be implemented just using PK/FK?

It is little bit tricky!

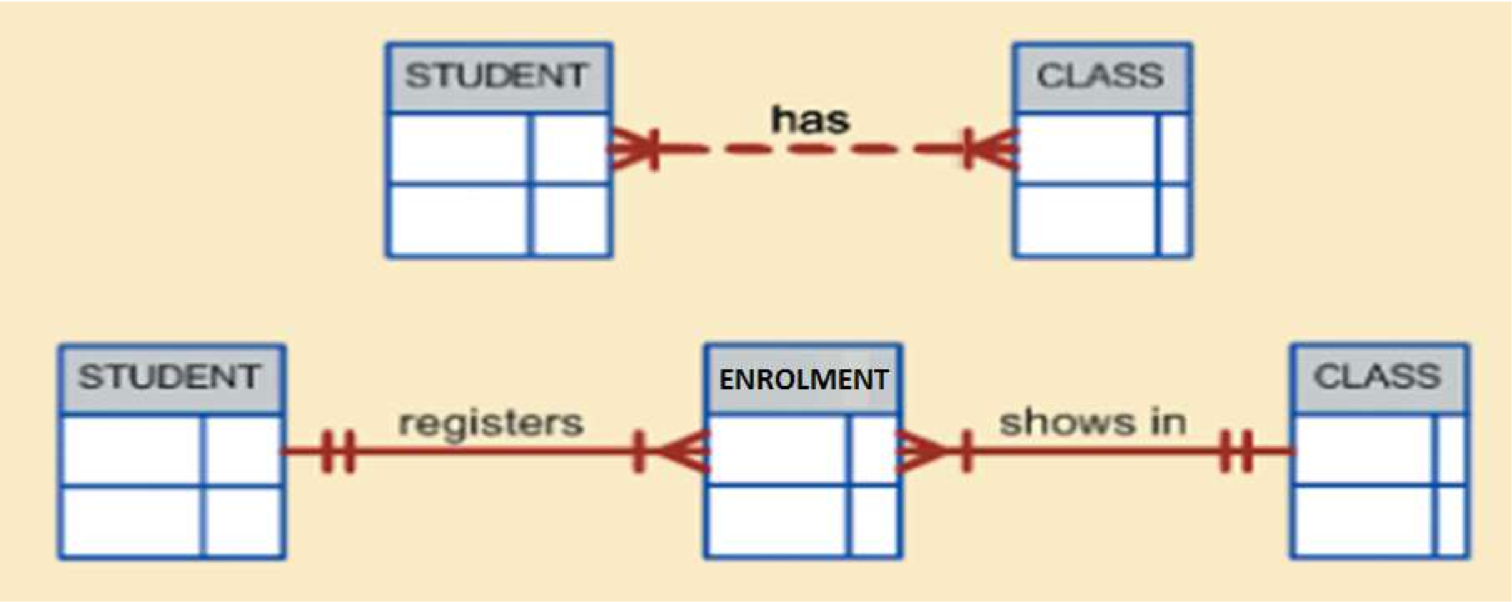
Implementing M:N Relationship

Replacing it with two 1:M relationships by creating a new composite table

- bridge table or associative entity

- Including foreign keys based on the primary keys of the 2 tables
- Assigning additional attributes as needed to the composite table

Textbook Figure 3.26



M:N Relationship Example

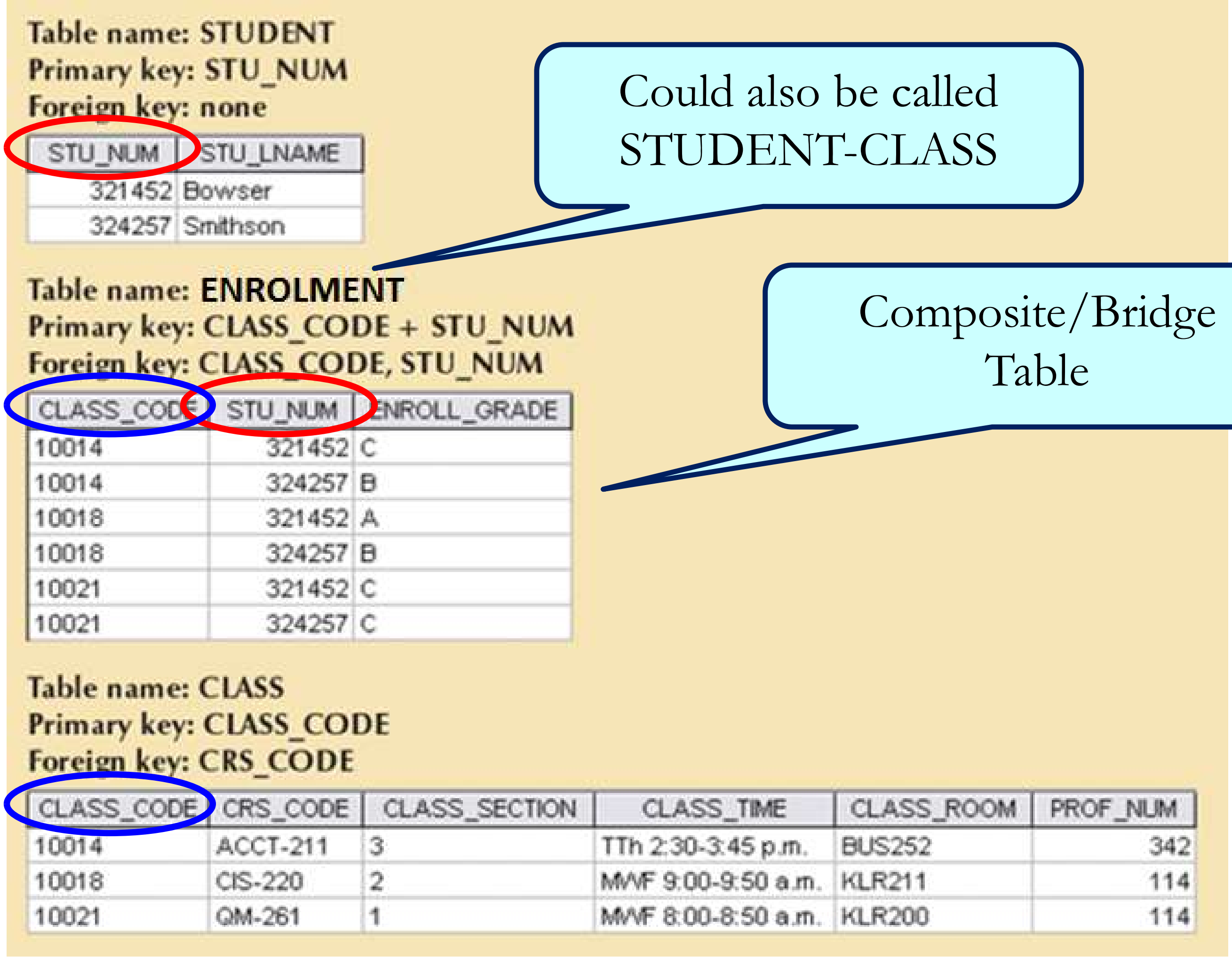


Table name: STUDENT
Primary key: STU_NUM
Foreign key: none

STU_NUM	STU_LNAME
321452	Bowser
324257	Smithson

Table name: ENROLMENT
Primary key: CLASS_CODE + STU_NUM
Foreign key: CLASS_CODE, STU_NUM

CLASS_CODE	STU_NUM	ENROLL_GRADE
10014	321452	C
10014	324257	B
10018	321452	A
10018	324257	B
10021	321452	C
10021	324257	C

Table name: CLASS
Primary key: CLASS_CODE
Foreign key: CRS_CODE

CLASS_CODE	CRS_CODE	CLASS_SECTION	CLASS_TIME	CLASS_ROOM	PROF_NUM
10014	ACCT-211	3	TTh 2:30-3:45 p.m.	BUS252	342
10018	CIS-220	2	MWF 9:00-9:50 a.m.	KLR211	114
10021	QM-261	1	MWF 8:00-8:50 a.m.	KLR200	114

Could also be called STUDENT-CLASS

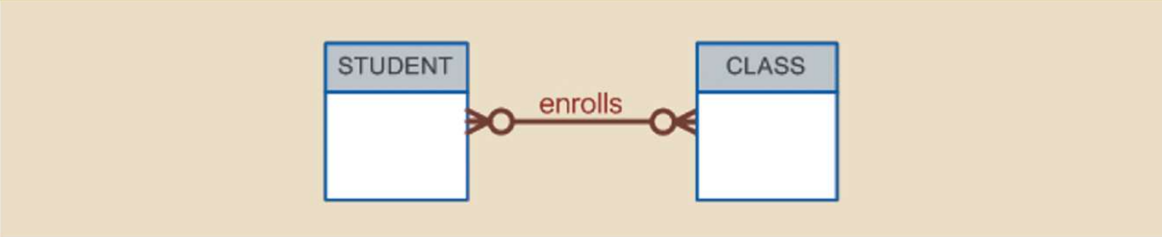
Composite/Bridge Table

Textbook
Figure 3.25

Associative Entity (Revisit)

- Used to implement an M:N relationship between two or more entities
- Composed of the primary key attributes of each parent entity
- May also contain additional attributes that play no role in connective process

FIGURE 4.24 THE M:N RELATIONSHIP BETWEEN STUDENT AND CLASS

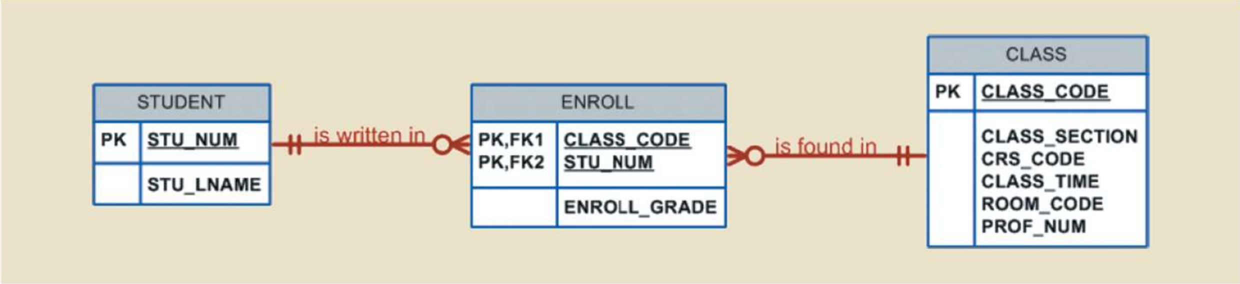


Textbook Figure 4.24

Business case

Implemented case

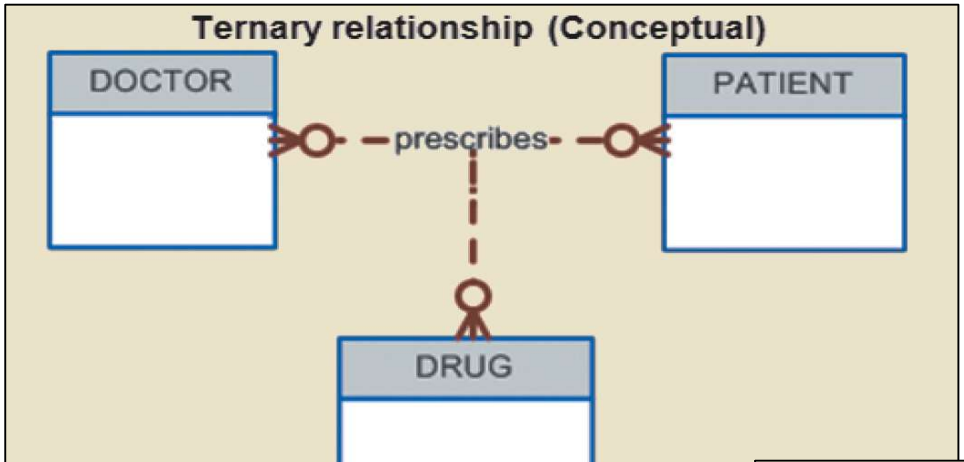
FIGURE 4.25 A COMPOSITE ENTITY IN AN ERD



Textbook Figure 4.25

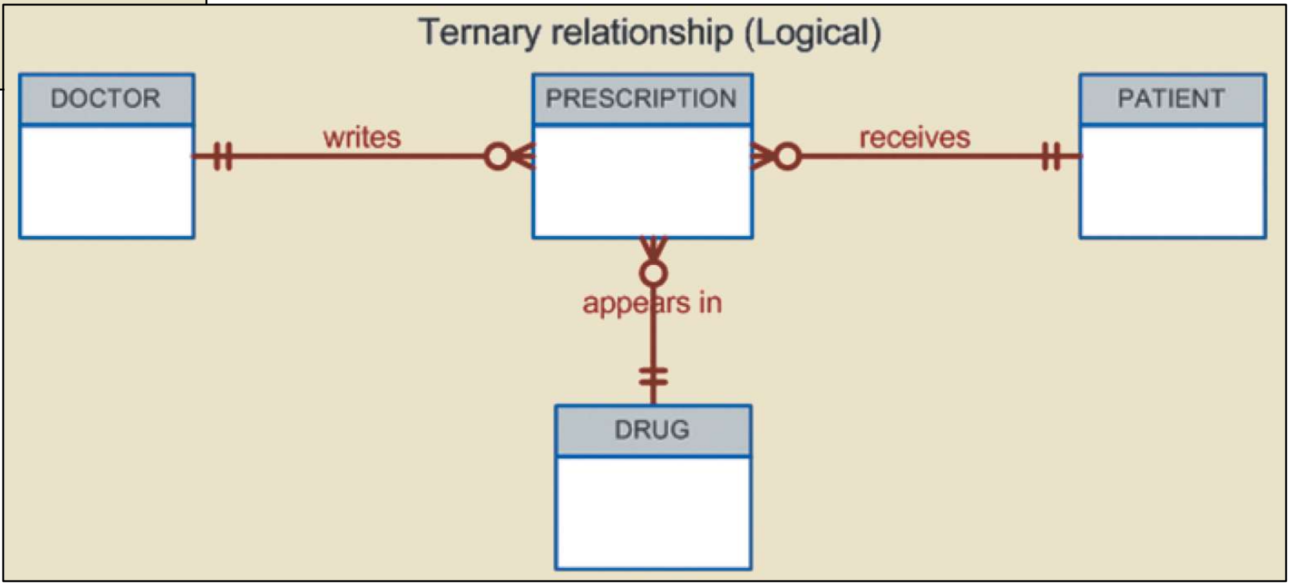
Associative Entity (Revisit) (2)

- DOCTORs prescribe DRUGs for PATIENTs.



Textbook
Figure 4.15

Textbook
Figure 4.15



Extended ER Model

- Advanced data modelling
- Result of adding more semantic constructs to ER model

  - modelling data requirements in complex real-world applications

  - Entity supertypes

  - Entity subtypes

  - Entity clustering

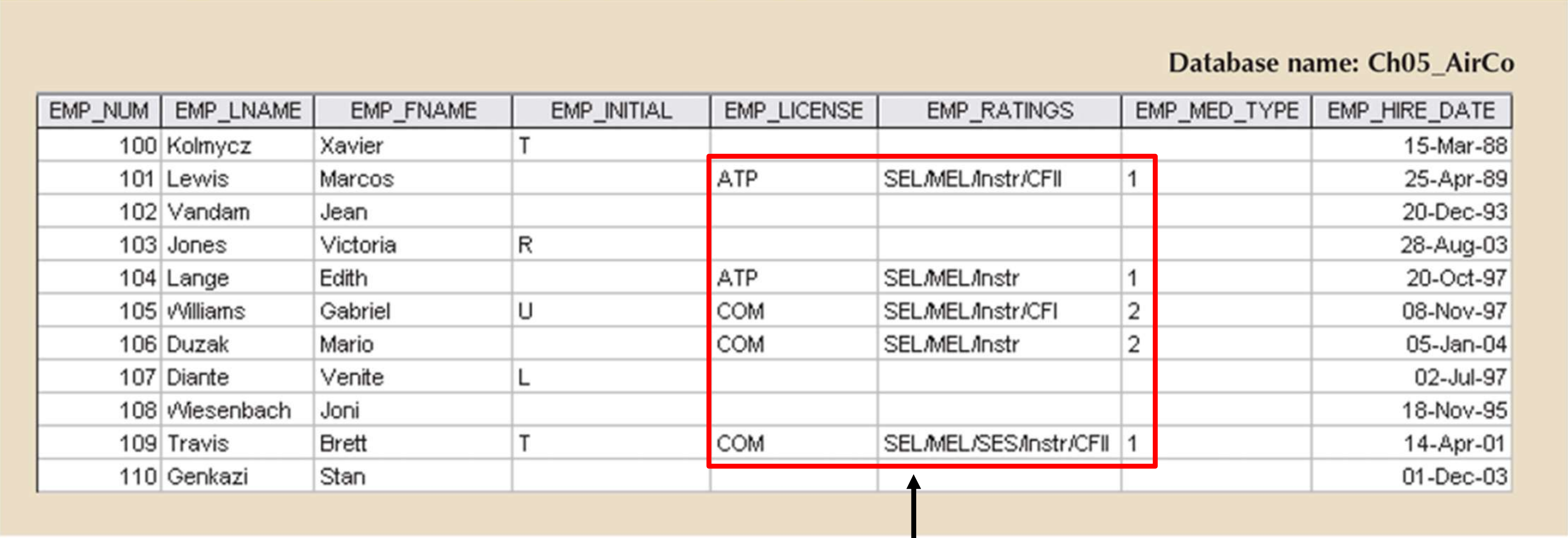
Extended ER Model (2)

- Consider a scenario, in which most employees possess a wide range of skills and special qualifications,
- Database designer must find a variety of ways to group employees based on their characteristics.
- For instance, a retail company could group employees as salaried and hourly,
- while a university could group employees as faculty, and admin staff
- The grouping of employees into various types provides two important benefits:
  - It avoids unnecessary NULLs in attributes when some employees have characteristics that are not shared by other employees.
- It enables a particular employee type to participate in relationships that are unique to that employee type.

Issues of NULLs

Textbook
Figure 5.1

FIGURE 5.1 NULLS CREATED BY UNIQUE ATTRIBUTES



Database name: Ch05_AirCo

EMP_NUM	EMP_LNAME	EMP_FNAME	EMP_INITIAL	EMP_LICENSE	EMP_RATINGS	EMP_MED_TYPE	EMP_HIRE_DATE
100	Kolmycz	Xavier	T				15-Mar-88
101	Lewis	Marcos		ATP	SEL/MEL/Instr/CFII	1	25-Apr-89
102	Vandam	Jean					20-Dec-93
103	Jones	Victoria	R				28-Aug-03
104	Lange	Edith		ATP	SEL/MEL/Instr	1	20-Oct-97
105	Williams	Gabriel	U	COM	SEL/MEL/Instr/CFI	2	08-Nov-97
106	Duzak	Mario		COM	SEL/MEL/Instr	2	05-Jan-04
107	Diante	Venite	L				02-Jul-97
108	Wiesenbach	Joni					18-Nov-95
109	Travis	Brett	T	COM	SEL/MEL/SES/Instr/CFII	1	14-Apr-01
110	Genkazi	Stan					01-Dec-03

These attributes are applicable to certain types of Employees only

Entity Supertypes and Subtypes

- **Entity supertype**
  - Generic entity type related to one or more entity subtypes
  - Contains common characteristics
- **Entity subtype**
  - Contains unique characteristics of each entity subtype
- Criteria to determine usage
  - The different kinds of instances should each have one or more attributes that are unique to that kind of instance
  - Define a special **Supertype attribute** known as the **Subtype discriminator**.
  - Define **disjoint** or **overlapping** constraints and **complete** or **partial** constraints.

Specialization Hierarchy

- **Entity supertypes and subtypes are organized in a specialization hierarchy**
  - Depicts arrangement of higher-level entity supertypes and lower-level entity subtypes
  - Relationships are described in terms of **"is-a" relationships**
  - Every subtype has one supertype to which it is directly related
  - Supertype can have many subtypes

E.g. Full-time employee is an Employee

Part-time employee is an Employee

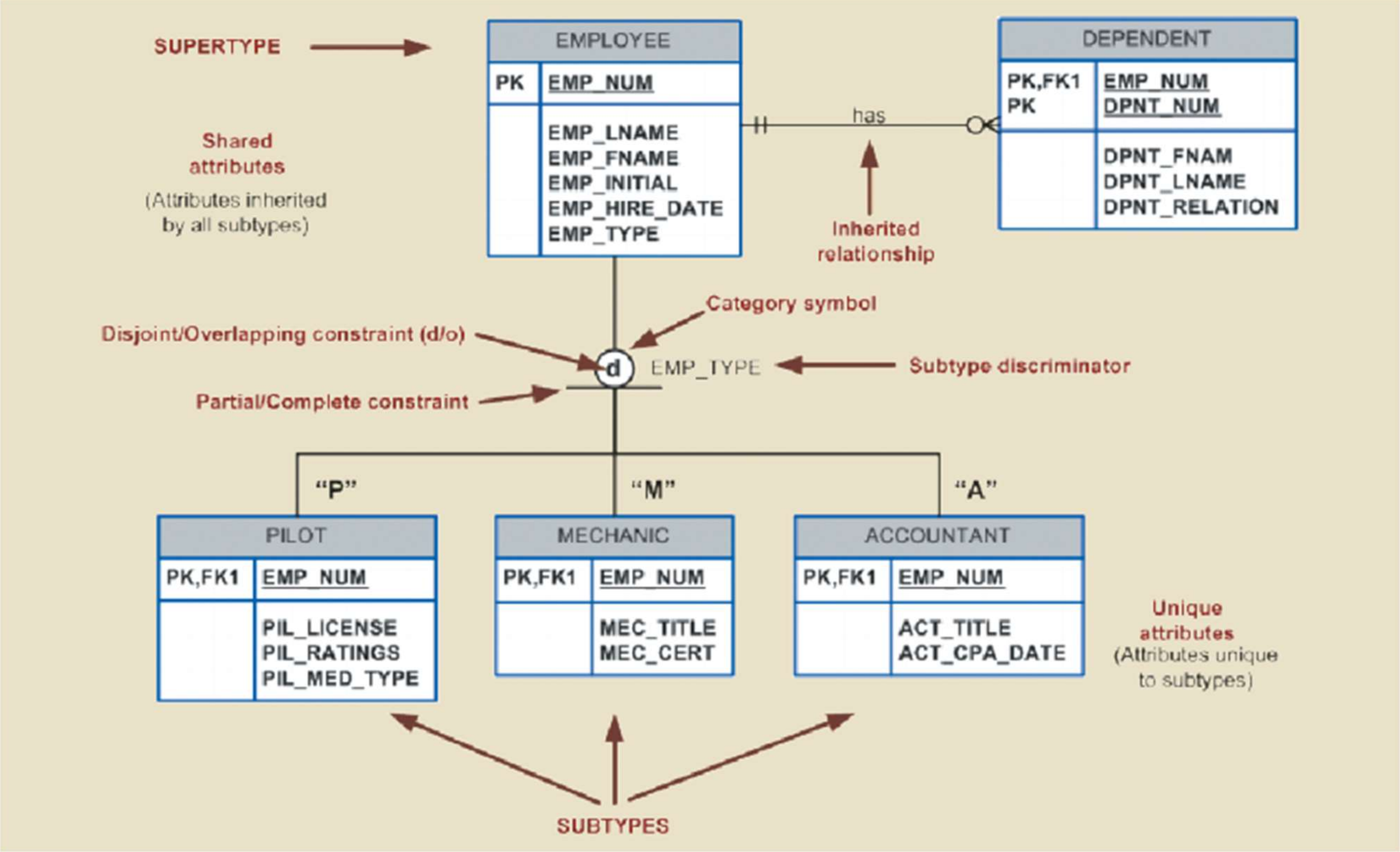
Subtype (special case)

Supertype (general case)

Specialization Hierarchy (2)

Textbook
Figure 5.2

FIGURE 5.2 A SPECIALIZATION HIERARCHY



SUPERTYPE

Shared attributes
(Attributes inherited by all subtypes)

EMPLOYEE
- PK: EMP_NUM
- EMP_LNAME
- EMP_FNAME
- EMP_INITIAL
- EMP_HIRE_DATE
- EMP_TYPE

has

DEPENDENT
- PK,FK1: EMP_NUM
- PK: DPNT_NUM
- DPNT_FNAM
- DPNT_LNAME
- DPNT_RELATION

Inherited relationship

Category symbol

Disjoint/Overlapping constraint (d/o)

d EMP_TYPE

Subtype discriminator

Partial/Complete constraint

"P" "M" "A"

PILOT
- PK,FK1: EMP_NUM
- PIL_LICENSE
- PIL_RATINGS
- PIL_MED_TYPE

MECHANIC
- PK,FK1: EMP_NUM
- MEC_TITLE
- MEC_CERT

ACCOUNTANT
- PK,FK1: EMP_NUM
- ACT_TITLE
- ACT_CPA_DATE

Unique attributes
(Attributes unique to subtypes)

SUBTYPES

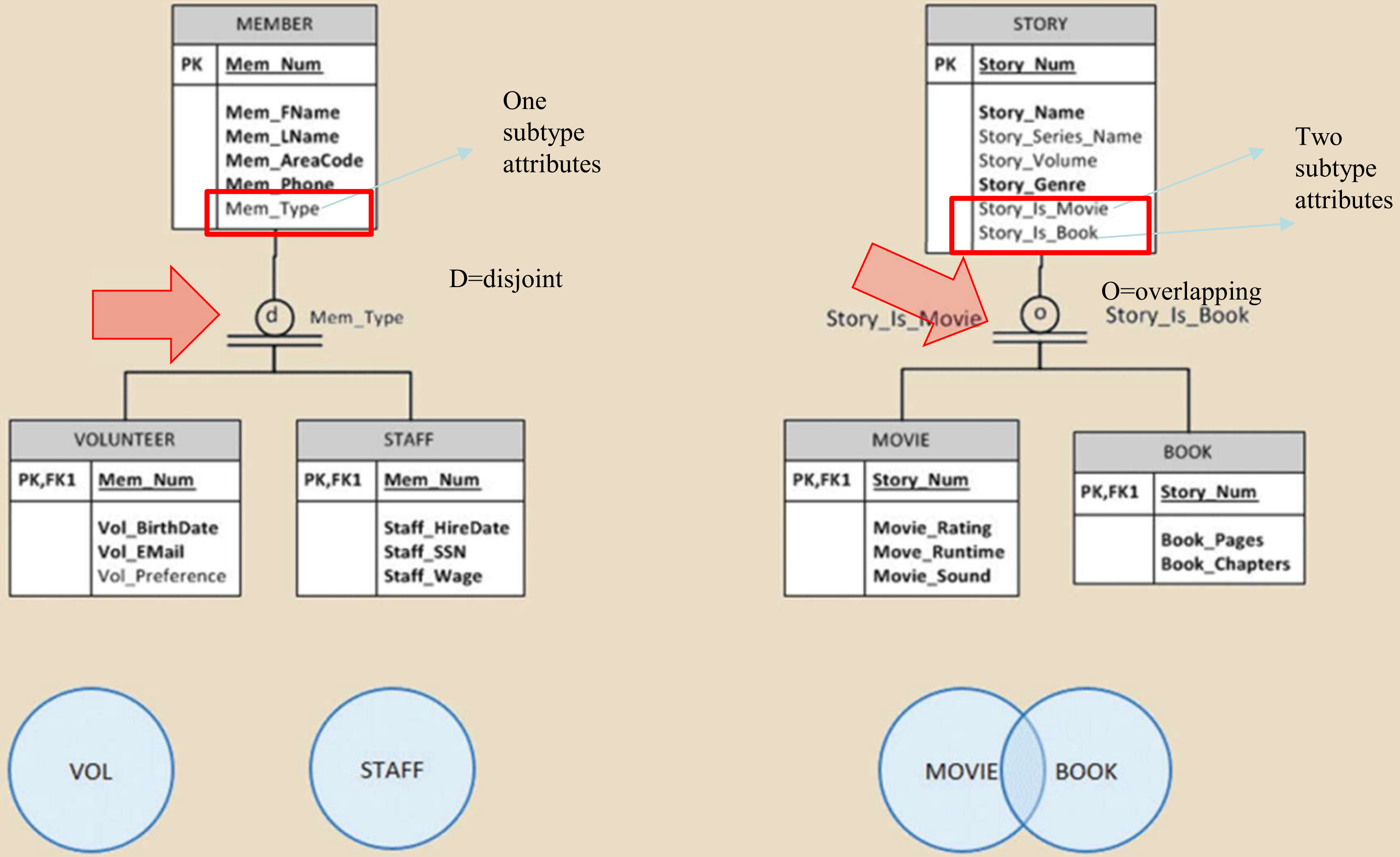
Disjoint and Overlapping Constraints

- **Disjoint subtypes**: contain a unique subset of the supertype entity set
  - Known as nonoverlapping subtypes
  - Implementation is based on the value of the subtype discriminator attribute in the supertype (in the parent entity)
- **Overlapping subtypes**: contain nonunique subsets of the supertype entity set
  - Implementation requires the use of one discriminator attribute for each subtype (in the parent entity)

Disjoint and Overlapping Constraints (2)

Textbook
Figure 5.5

FIGURE 5.5 DISJOINT AND OVERLAPPING SUBTYPES



One subtype attributes

D=disjoint

Two subtype attributes

O=overlapping

disjoint

A MEMBER can be in VOLUNTEER or STAFF, but not both

overlapping

A STORY can be in MOVIE, BOOK, or both

Completeness constraints

- Specifies whether each supertype occurrence must also be a member of at least one subtype
  - **Partial completeness**: not every supertype occurrence is a member of a subtype
  - **Total completeness**: every supertype occurrence must be a member of at least one subtypes

Textbook Figure 5.2

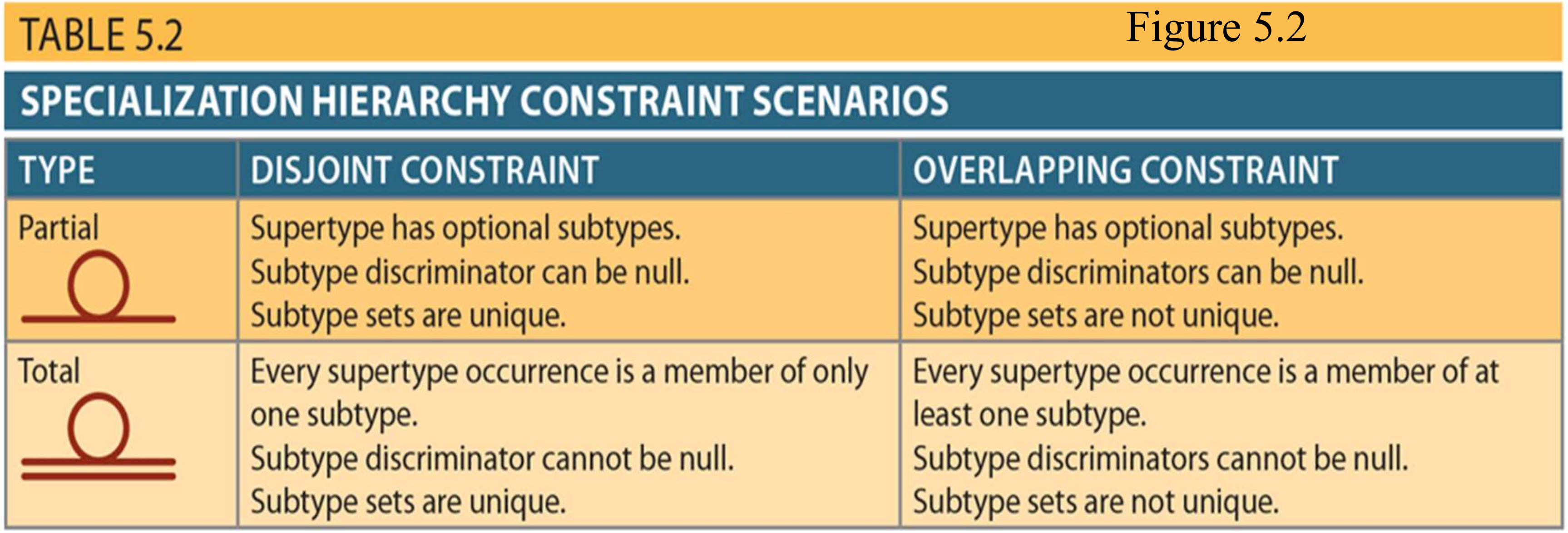


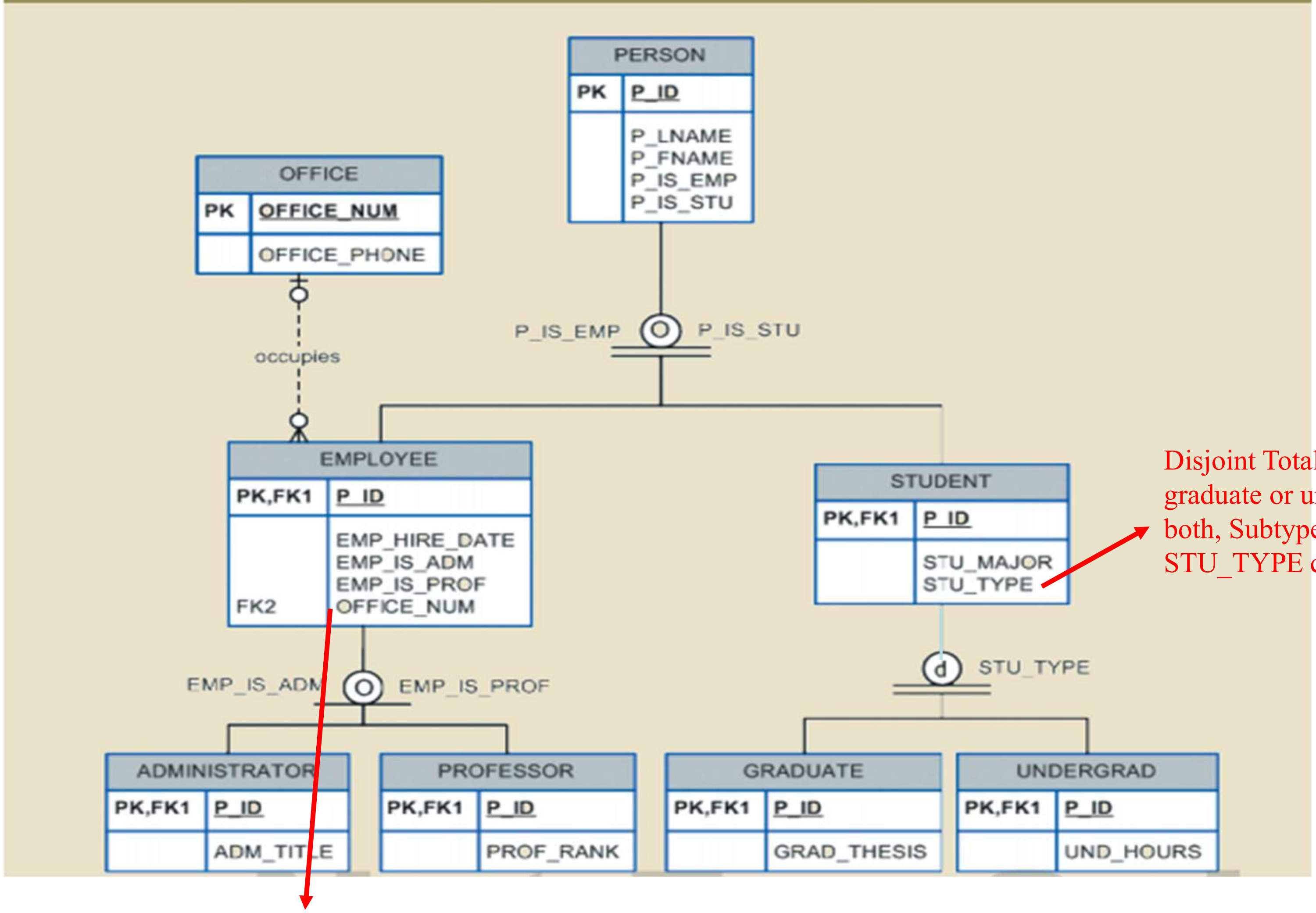
TABLE 5.2

SPECIALIZATION HIERARCHY CONSTRAINT SCENARIOS

TYPE	DISJOINT CONSTRAINT	OVERLAPPING CONSTRAINT
Partial	Supertype has optional subtypes. Subtype discriminator can be null. Subtype sets are unique.	Supertype has optional subtypes. Subtype discriminators can be null. Subtype sets are not unique.
Total	Every supertype occurrence is a member of only one subtype. Subtype discriminator cannot be null. Subtype sets are unique.	Every supertype occurrence is a member of at least one subtype. Subtype discriminators cannot be null. Subtype sets are not unique.

Levels of hierarchy

FIGURE 5.4 SPECIALIZATION HIERARCHY WITH OVERLAPPING SUBTYPES



Textbook Figure 5.4

Disjoint Total: Student must be either graduate or undergard, but cannot be both, Subtype discriminator: STU_TYPE cannot be NULL

Overlapping partial: All employee may not be (either administrator or a professor)
Subtype discriminator: EMP_IS_ADM,EMP_IS_PROFC for this can be NULL

Specialization and Generalization

Specialization

- Top-down process
- Identifies lower-level, more specific entity subtypes from a higher-level entity supertype
- Based on grouping unique characteristics and relationships of the subtypes

Generalization

- Bottom-up process
- Identifies a higher-level, more generic entity supertype from lower-level entity subtypes
- Based on grouping common characteristics and relationships of the subtypes

Entity Clustering

"Virtual/Abstract" entity type used to represent multiple entities and relationships in ERD

- To simply complex ERD

  - in some problems you may have hundreds of entities

- Formed by combining multiple interrelated entities into a single, abstract entity object

- **General rule**: avoid the display of attributes to eliminate complications that result when the inheritance rules change

- Not an actual entity from the business rule

- Not implemented

Entity Clustering (2)

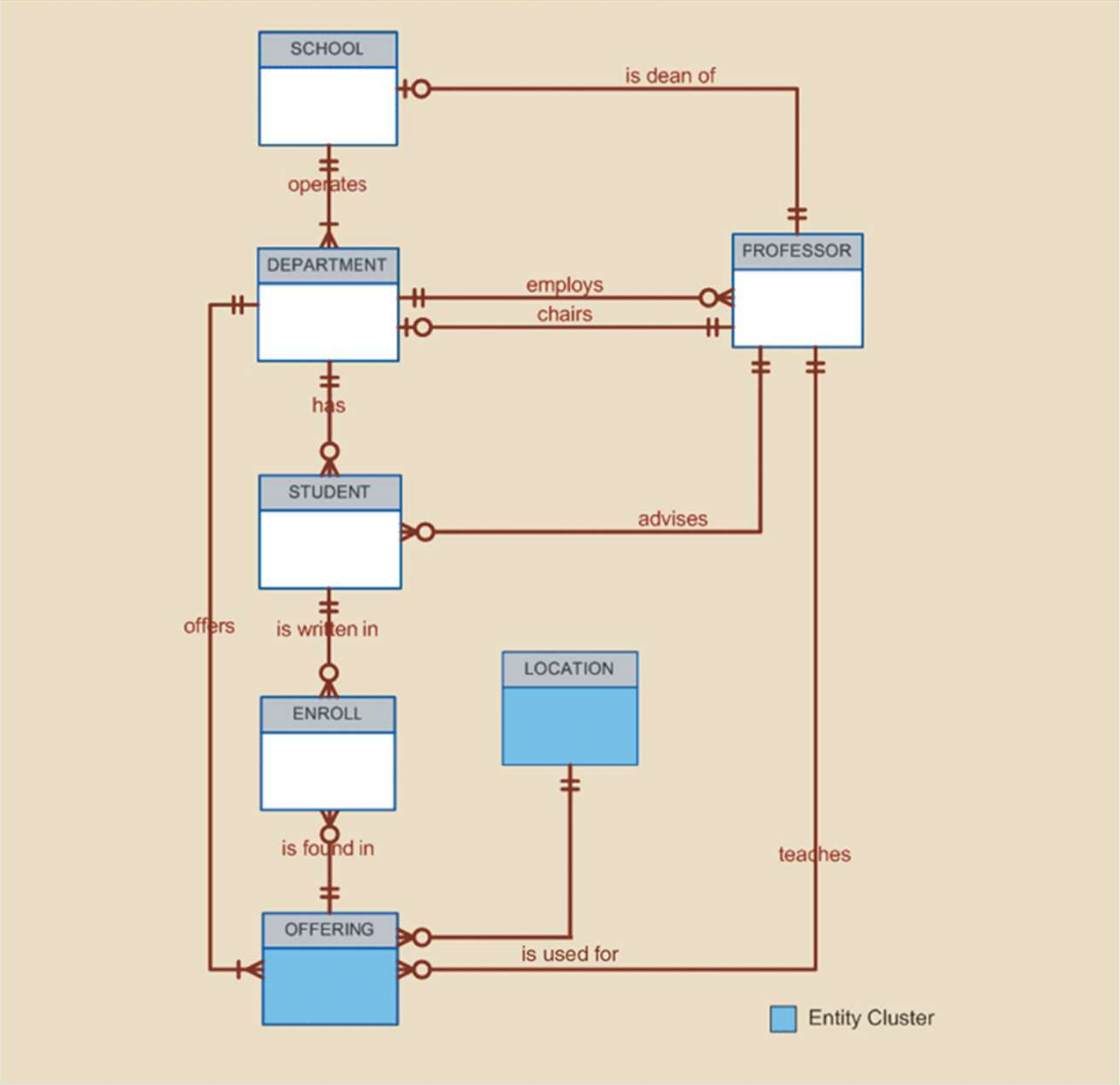
Textbook
Figure 5.6

OFFERING groups SEMESTER, COURSE, and CLASS

LOCATION groups BUILDING and ROOM

Relationships of Entities within the Entity cluster is represented in a separate ERD

FIGURE 5.6 TINY COLLEGE ERD USING ENTITY CLUSTERS



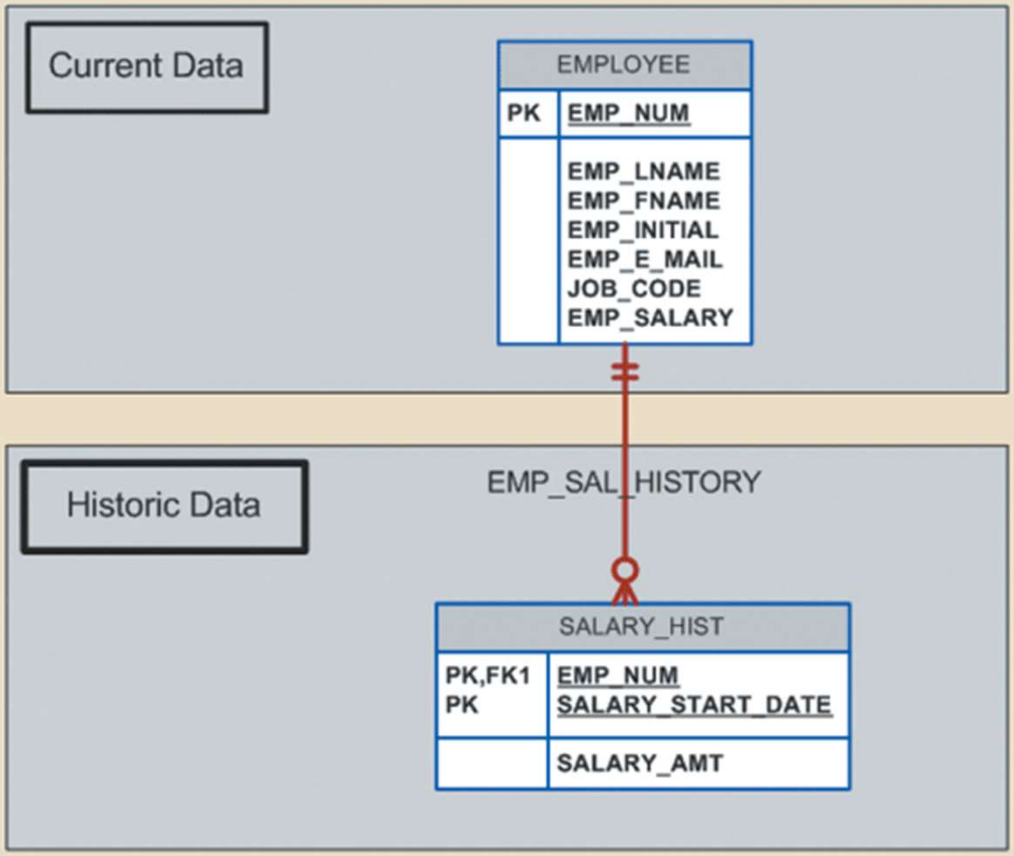
History of Time-variant data

- **Time-variant data**: data whose values change over time and for which a history of the data changes must be retained
  - Requires creating a new entity in a 1:M relationship with the original entity
  - New entity contains the new value, date of the change, and any other pertinent attribute

E.g., tracking salary histories of employees

Maintaining data history

FIGURE 5.9 MAINTAINING SALARY HISTORY



Current Data

EMPLOYEE	
PK	EMP_NUM
	EMP_LNAME EMP_FNAME EMP_INITIAL EMP_E_MAIL JOB_CODE EMP_SALARY

EMP_SAL_HISTORY

Historic Data

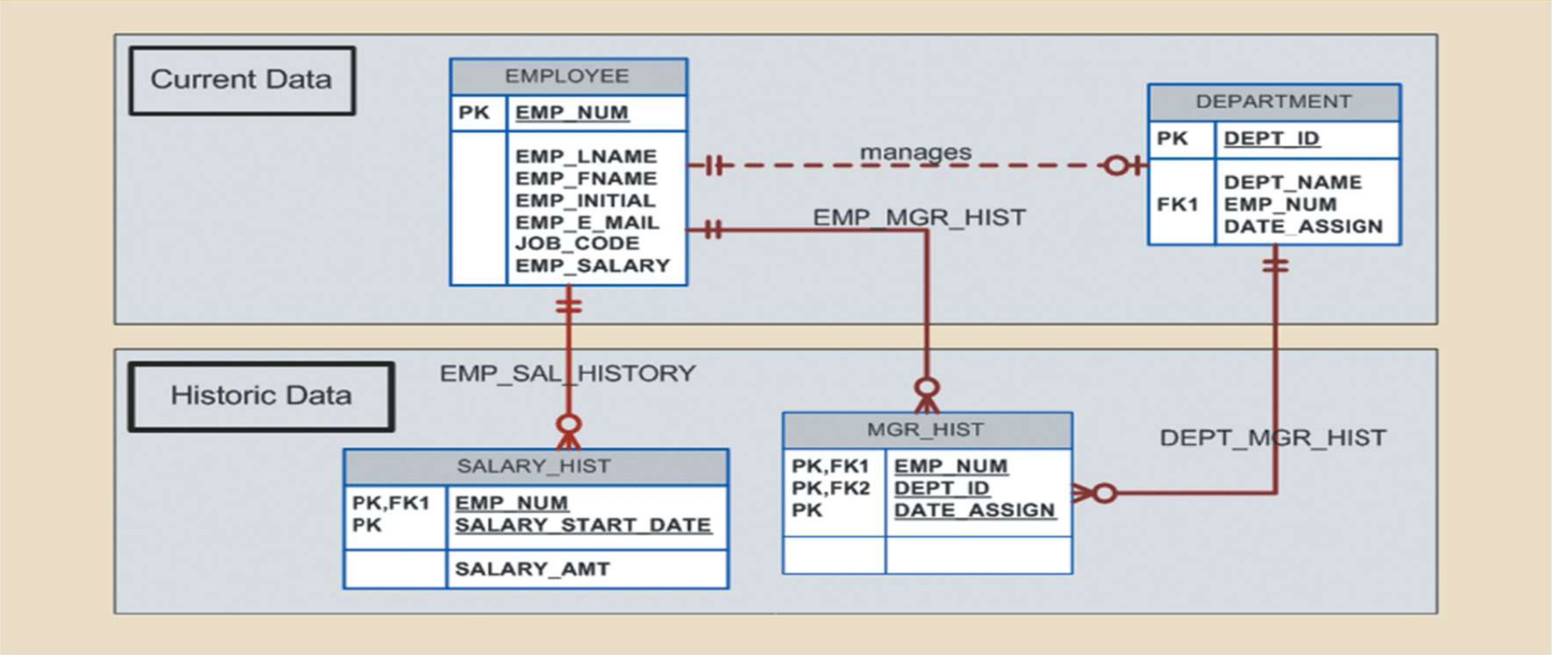
SALARY_HIST	
PK,FK1 PK	EMP_NUM SALARY_START_DATE
	SALARY_AMT

Textbook Figure 5.9

Maintaining data history (2)

- An employee could be the manager of many different departments over time,
- A department could have different managers over time.
- Because you are recording time-variant data, you must store the DATE_ASSIGN attribute in the MGR_HIST entity to provide the date that the employee (EMP_NUM) became the department manager.

FIGURE 5.10 MAINTAINING MANAGER HISTORY



Textbook Figure 5.10

ERD Exercise: Case 1

Draw an ER diagram for the following description

The HEG has twelve instructors and can handle up to thirty trainees per class. HEG offers five "advanced technology" courses, each of which may generate several classes. If a class has fewer than ten trainees in it, it will be cancelled. It is, therefore, possible for a course not to generate any classes during a session. Each class is taught by one instructor. Each instructor may teach up to two classes or may be assigned to do research only. Each trainee may take up to two classes per session.

ERD Exercise: Case 1 (2)

Determine Entities

The HEG has twelve instructors and can handle up to thirty trainees per class. HEG offers five "advanced technology" courses, each of which may generate several classes. If a class has fewer than ten trainees in it, it will be cancelled. It is, therefore, possible for a course not to generate any classes during a session. Each class is taught by one instructor. Each instructor may teach up to two classes or may be assigned to do research only. Each trainee may take up to two classes per session.

ERD Exercise: Case 1 (3)

Determine Relationships

- One instructor → Many classes (*teach*)
- One class → One instructor (*taught by*)
- Some instructors MAY NOT teach any classes

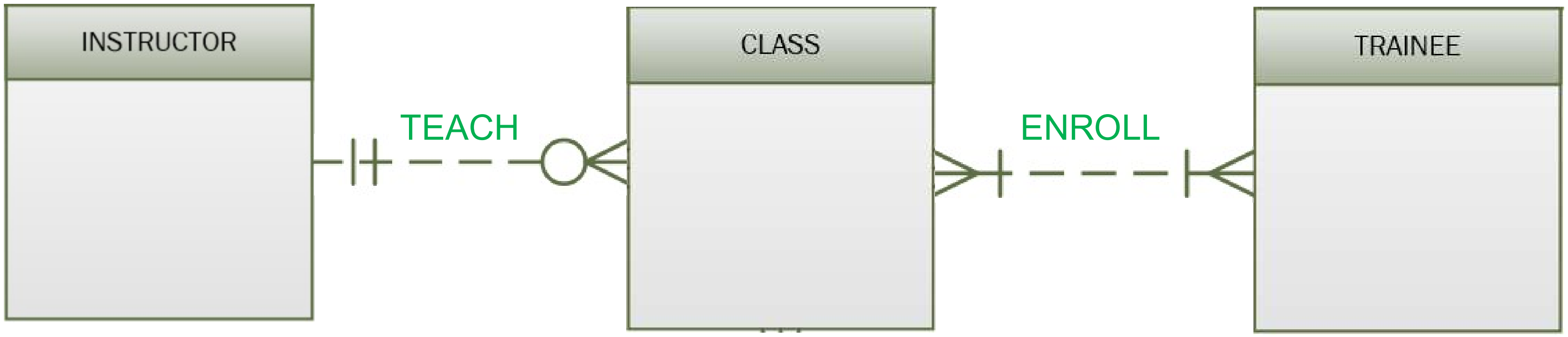
- One course → Many classes (*generate*)
- One class → One course (*generated by*)
- Some courses MAY NOT generate any classes

- One trainee → Many classes (*enroll*)
- One class → Many trainees (*enrolled by*)

ERD Exercise: Case 1 (4)

HEG has twelve **instructors** and can handle up to thirty **trainees** per **class**. …

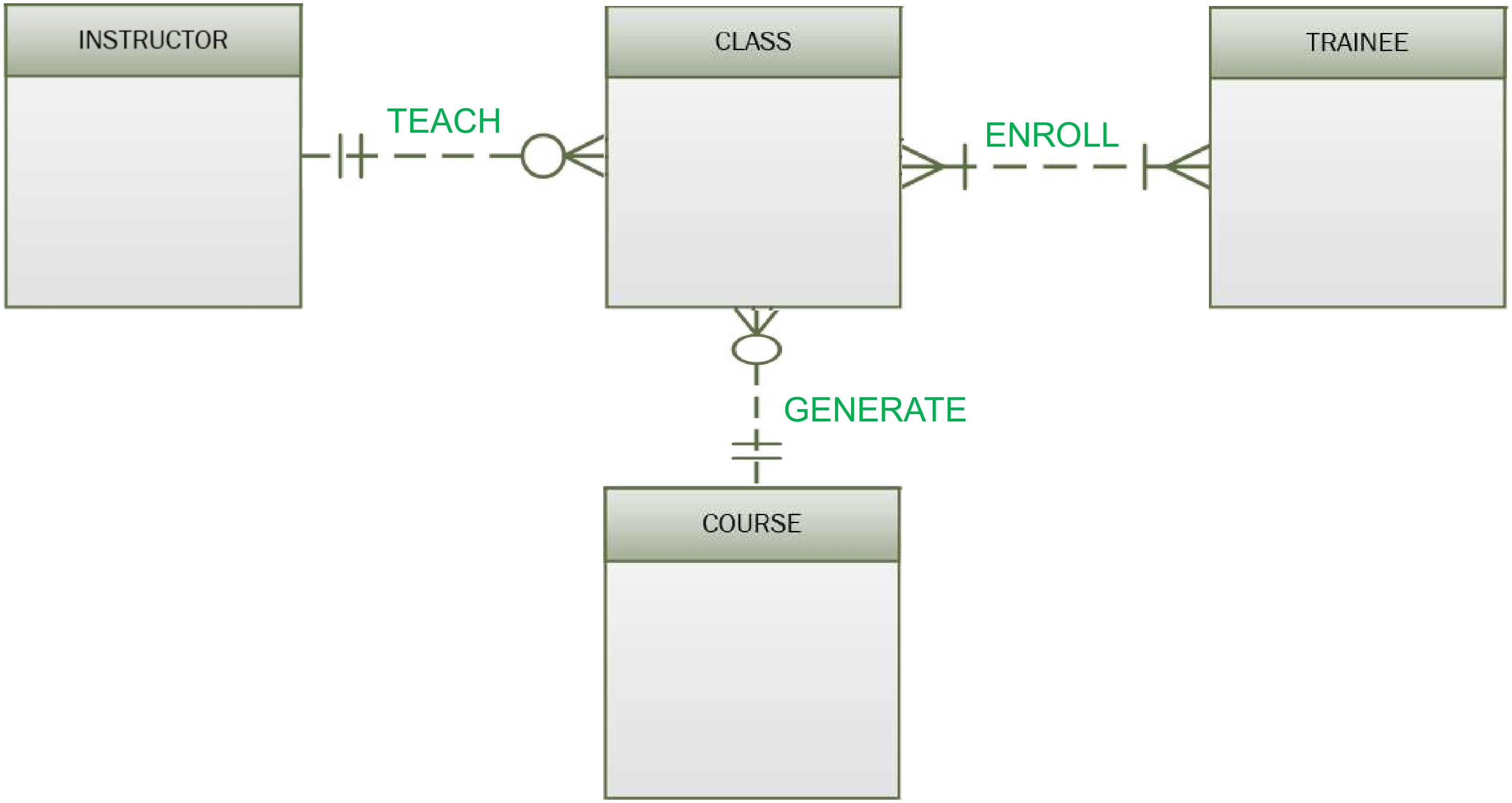
Also consider the business rules about: Instructor, Class, Trainee



ERD Exercise: Case 1 (4)

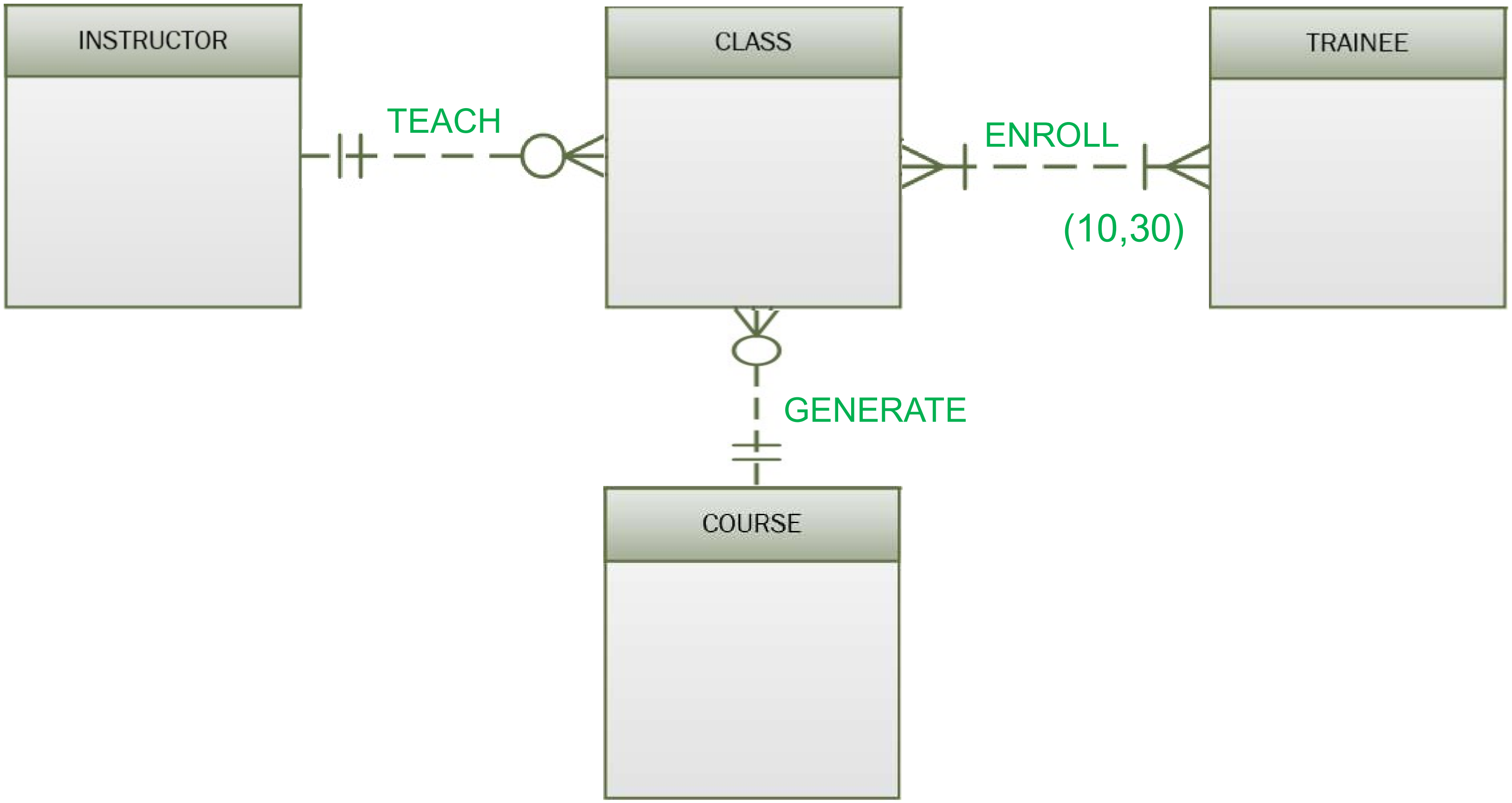
… HEG offers five "advanced technology" ***courses****, each of which may* generate *several* ***classes****. …*

Also consider the business rules about: Class, Course



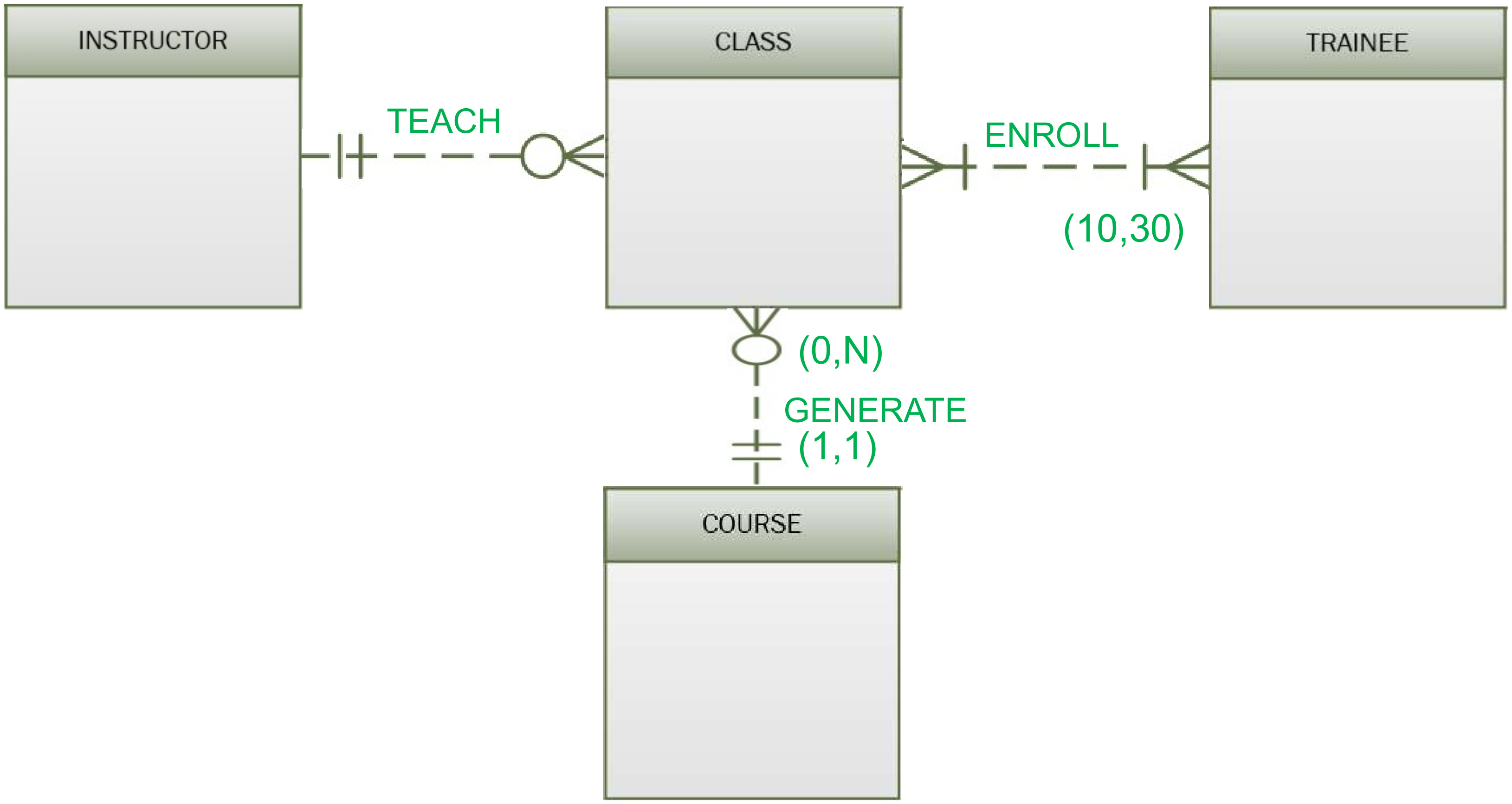
ERD Exercise: Case 1 (5)

*… twelve instructors can handle up to **thirty trainees per class**. … . If a class has fewer than **ten trainees** in it, it will be cancelled.*



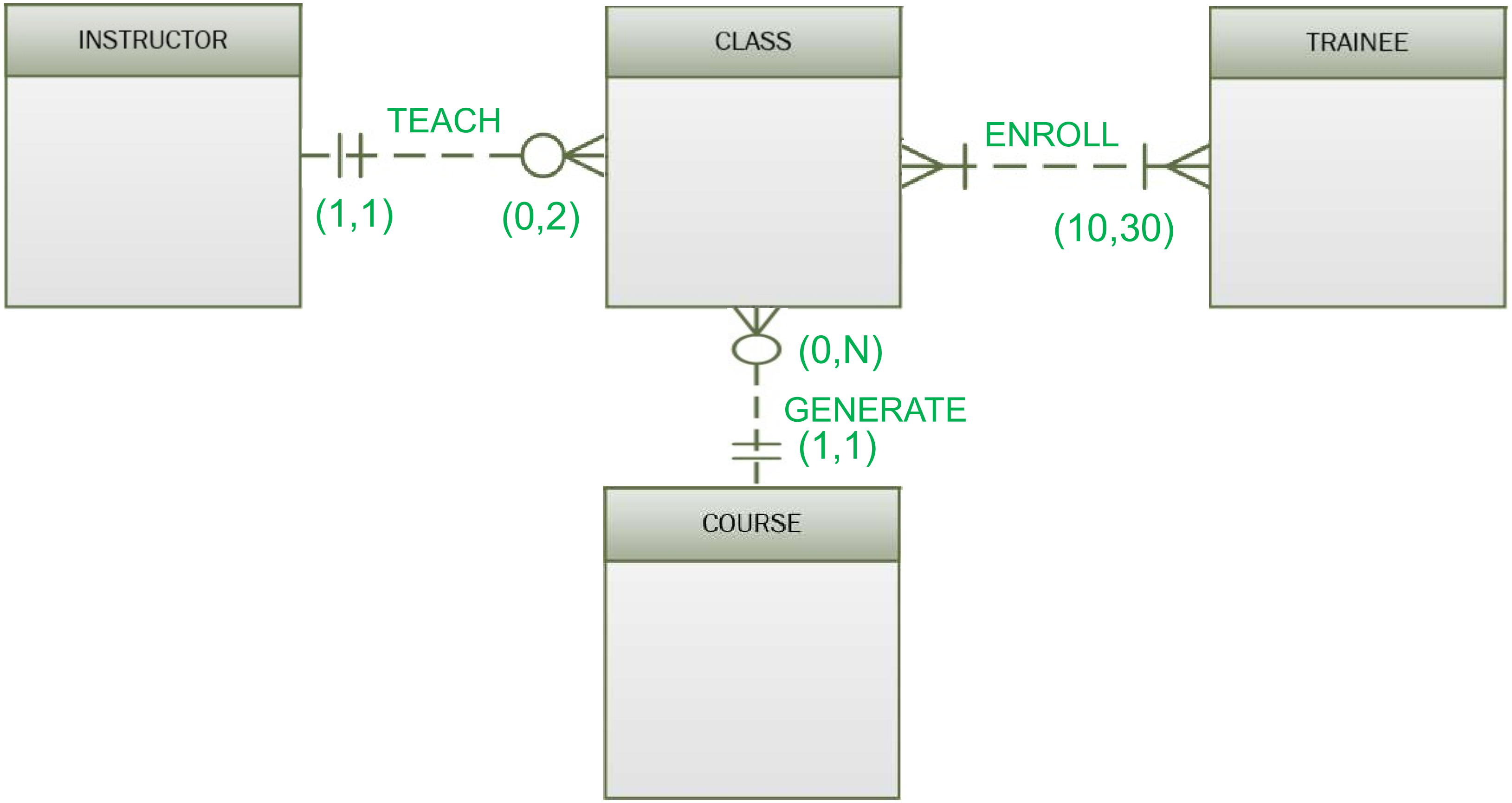
ERD Exercise: Case 1 (6)

… each course may generate ***<u>several</u>*** *classes. … It is, therefore, possible for a course* ***<u>not to generate any classes</u>*** *during a session.*



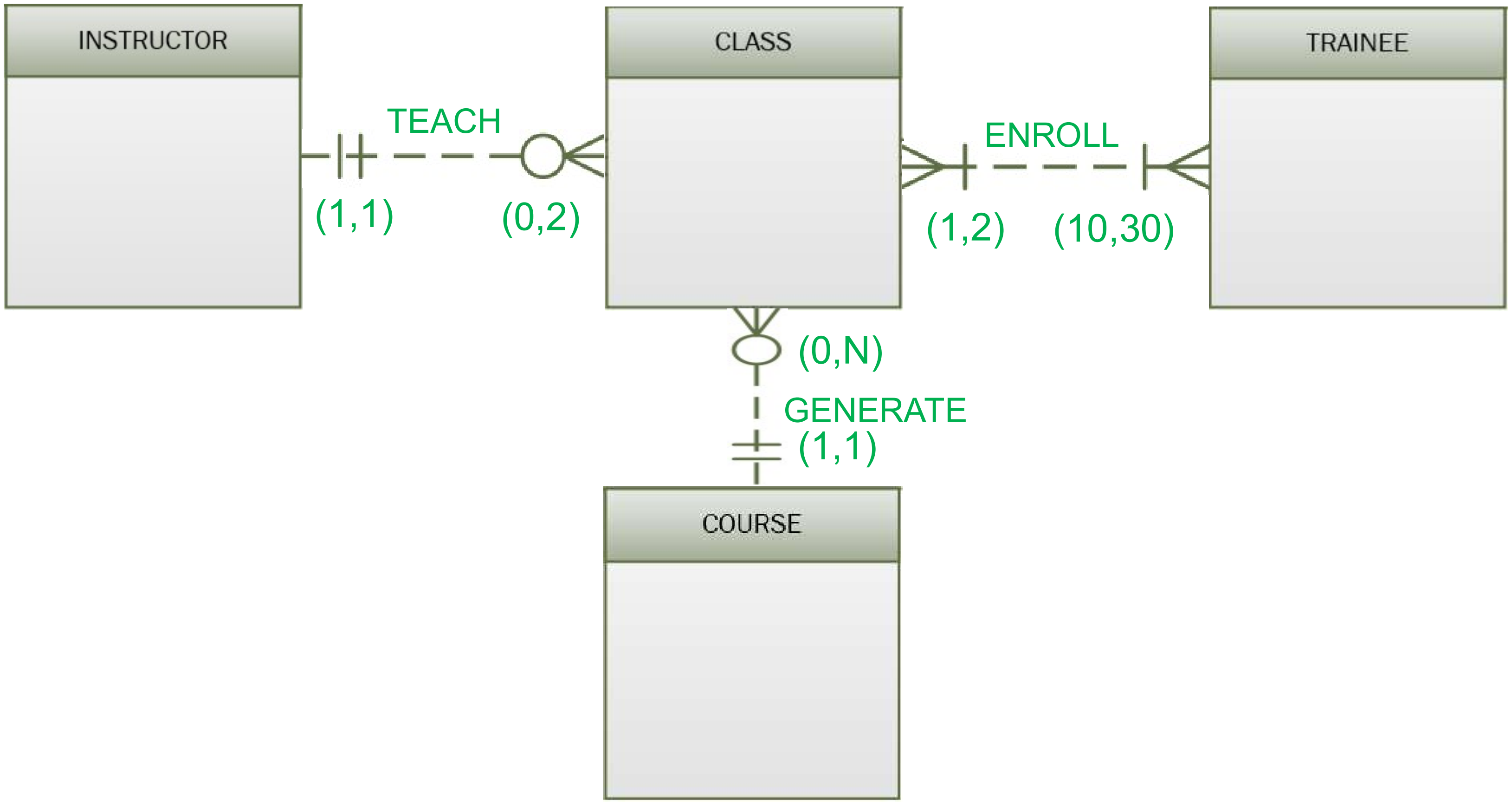
ERD Exercise: Case 1 (7)

*… . Each class is taught by **<u>one instructor</u>**. Each instructor may teach **<u>up to two classes</u>** or may be assigned to **<u>do research only</u>**.*



ERD Exercise: Case 1 (8)

… Each trainee may take up to ***two classes*** *per session.*



ERD Exercise: Case 1 (3)

Note that we **did not** consider **Attributes** here

What next:

- Attributes
- Primary keys
- Maintaining relationships

  - using FK

  - using bridge/associative entity (M:N relationships)

This is a homework for you!

ERD Exercise: Case 2

Draw an E-R diagram for the following description

The Jonesburgh County Basketball Conference (JCBC) is an amateur basketball association. Each city in the county has one team that represents it. Each team has a maximum of twelve players and a minimum of nine players. Each team also has up to three coaches (offensive, defensive and PT coaches). Each team plays two games (home and visitor) against each of the other teams during the season.

ERD Exercise: Case 2 (2)

Determine Entities

The Jonesburgh County Basketball Conference (JCBC) is an amateur basketball association. Each city in the county has one team that represents it. Each team has a maximum of twelve players and a minimum of nine players. Each team also has up to three coaches (offensive, defensive and PT coaches). Each team plays two games (home and visitor) against each of the other teams during the season.

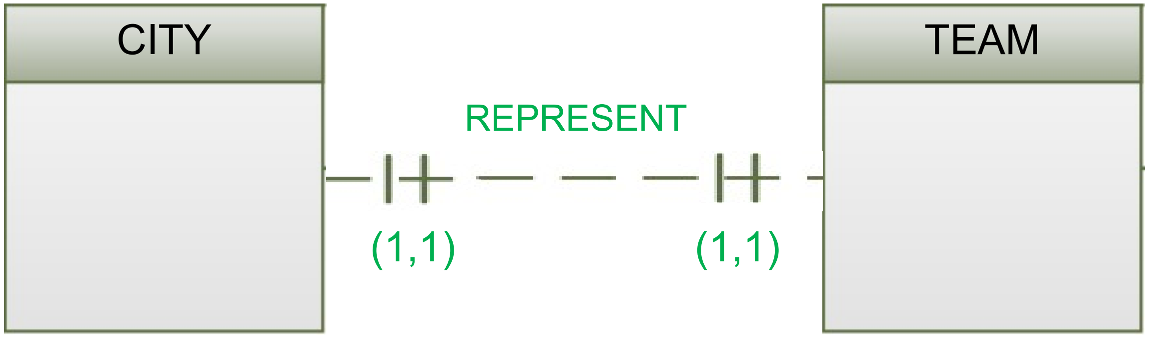
ERD Exercise: Case 2 (3)

Determine Relationships

- One city → One team (*posses*)
- One team → One city (*represent*)
- One team → Many players (*has*)
- One player → One team (*belong to*)
- One team → Many coaches (*employ*)
- One coach → One team (*employed by*)
- One team → Many teams (*play against*)
- One team → Many teams (*play against*)

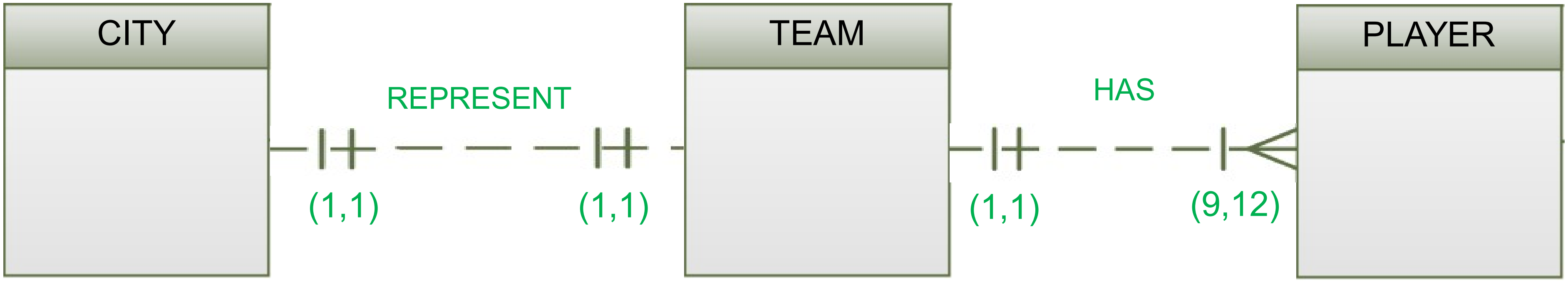
ERD Exercise: Case 2 (4)

… Each **city** in the county has one **team** that represents it. …



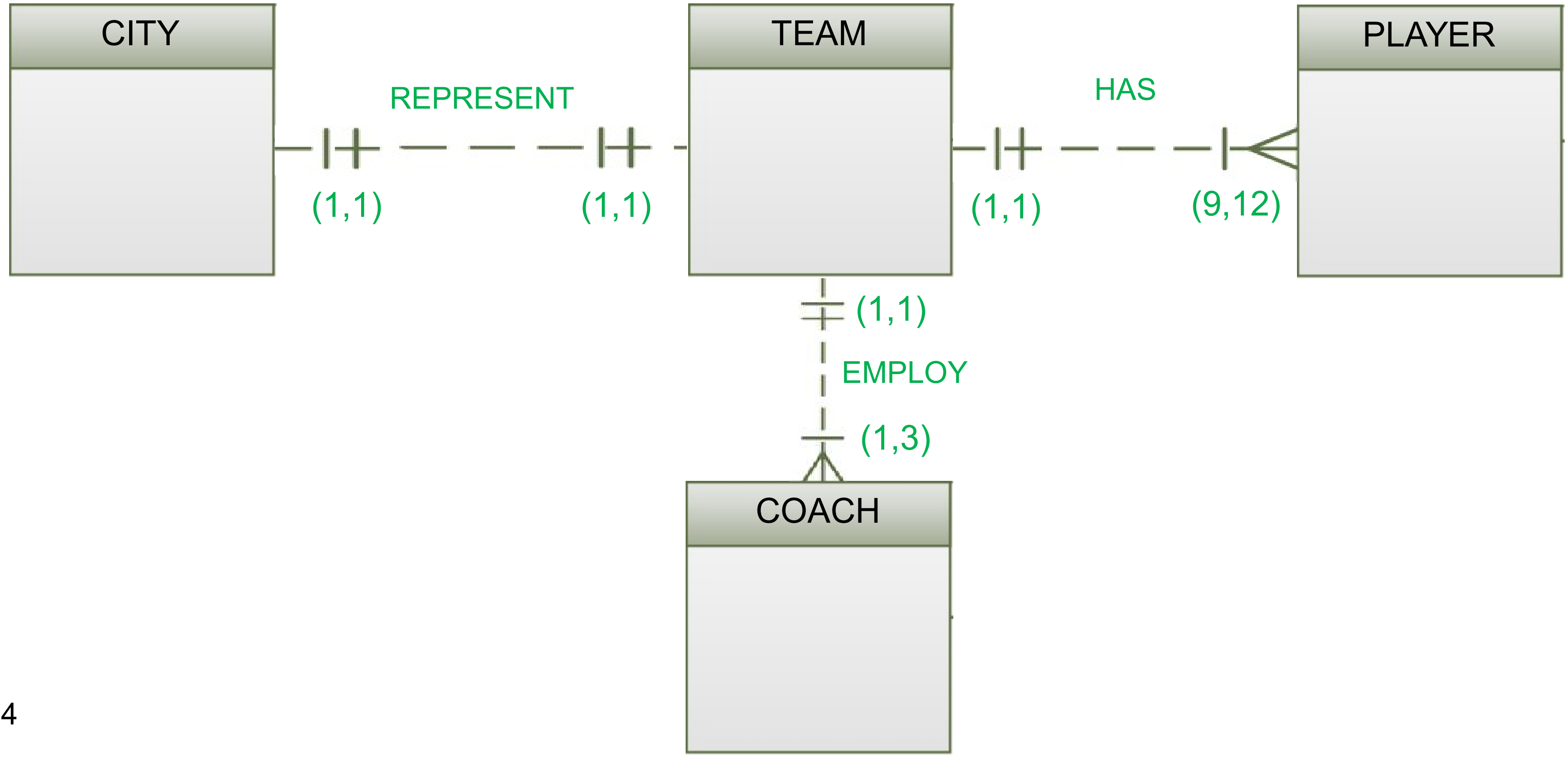
ERD Exercise: Case 2 (5)

…. Each **team** has a minimum of nine **players** and a maximum of twelve players.….



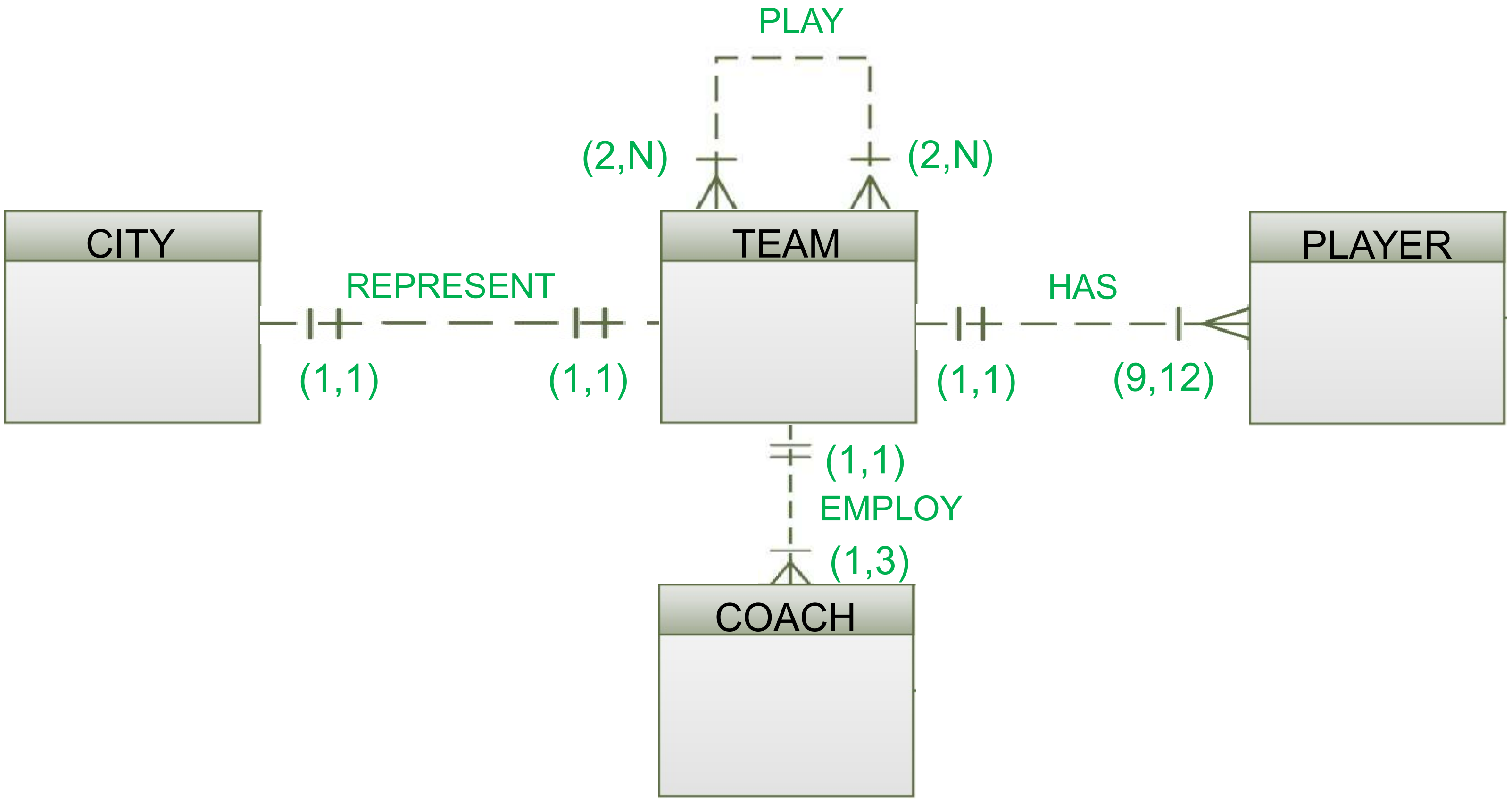
ERD Exercise: Case 2 (6)

… Each **team** also has up to three **coaches** …



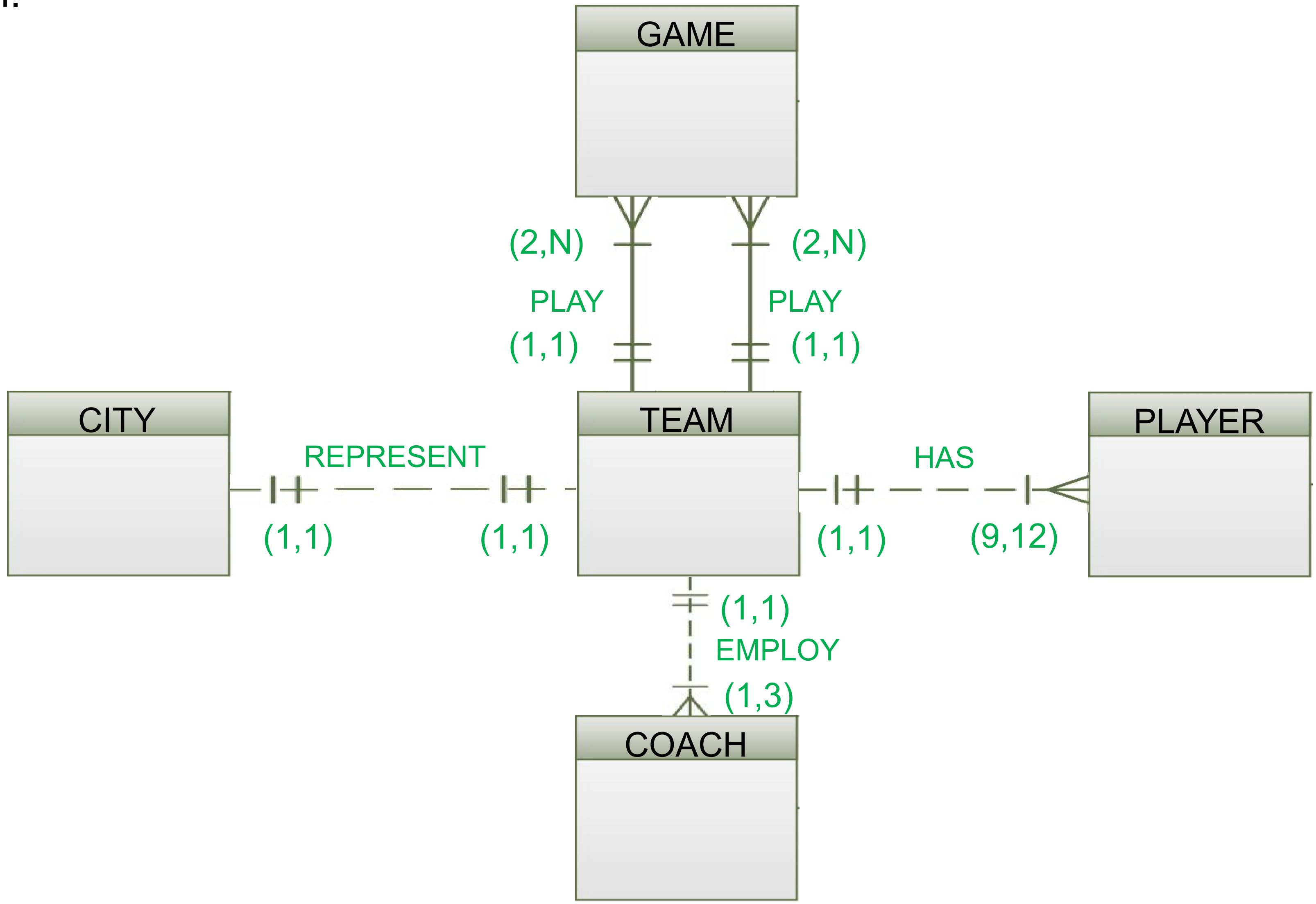
ERD Exercise: Case 2 (7)

…. Each **team** plays two games (home and visitor) against each of the other **teams** during the season.



ERD Exercise: Case 2 (8)

…. Each **team** plays two games (home and visitor) against each of the other **teams** during the season.



Summary

- Types of Attributes
- Strengths of Relationships
- Types of Entities
- Implementing Relationships
- Extended/Advanced ERD concepts
- Modelling historical time-variant data
- Case studies

This Week's OnTrack Tasks

- 3.1P Modelling database for a given business scenario in terms of ERD

  - Using Lucid Chart or MS Visio

- 3.2HD Research Report and Presentation

  - On a topic of your interest related to database and/or data management

  - Select one from the given list or propose your own (discuss with us)

  - Due on Friday of Week 9 (16 Sept 2022)

- Please check the task sheets and start working on them

Next Week

- Normalisation

Thank you

See you next week

Any questions/comments?

Readings and References:

- Chapters 2-5 and 9

Database Systems : Design, Implementation, & Management 13TH EDITION, by Carlos Coronel, Steven Morris