Module 2.2 – Extended ER (EER) Model

2. DB Modeling

- 2.1 Entity Relationship (ER) Model
    - Entities, Relationships, Constraints
- 2.2 Extended Entity Relationship (EER) Model
    - Specialization, Subtype, Supertype, Subtype Discriminator
- 2.3 Translation of EER Models to Relations

2.1 ER Model

Learning Outcomes

- EERM
  - Understand and Explain Extended ER modeling concepts
    - Inheritance, Subtypes, supertypes, specialization hierarchy, subtype discriminator; Overlapping vs disjoint, mandatory vs optional completeness
  - Understand that various standards are used in ER modeling (Chen, **Crow's foot**, UML, …)
  - Explain and use Associative entities
  - Create ER diagrams from user requirements
- Selecting Primary Keys & Flexible Design
  - Understand the selection process for primary keys and when to use surrogate keys
  - Explain: implementing 1:1 relationships, fan-traps, redundant relationships
  - Design for maintaining historical data

Textbook Readings

- Chap 5

Testing*

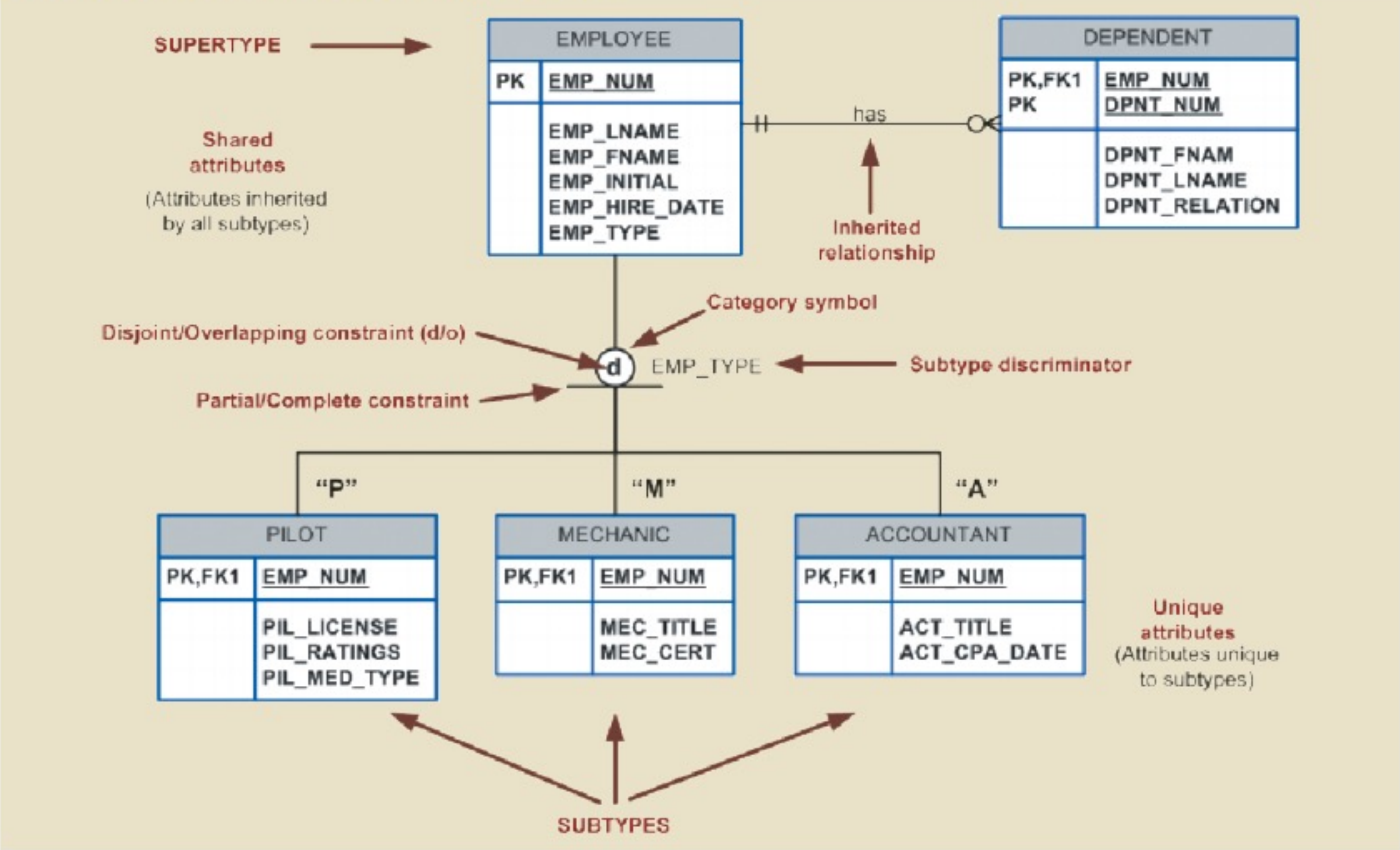
*Main (but not the only ones) sections of the textbook used for testing are identified in parentheses

- Entity Relationship modeling (5.1, 5.3, 5.4)

EER: Supertypes, Subtypes, Specialization Hierarchy

- Supertype
- Subtype
- Specialization
- Attribute Inheritance
- Subtype Discriminator

FIGURE 5.2 A SPECIALIZATION HIERARCHY



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
    - There must be different, identifiable kinds of the entity in the user's environment
    - The different kinds of instances should each have one or more attributes that are unique to that kind of instance

Specialization Hierarchy - Attribute Inheritance

- 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
  - Subtype exists within the context of a supertype
  - Every subtype has one supertype to which it is directly related
  - Supertype can have many subtypes

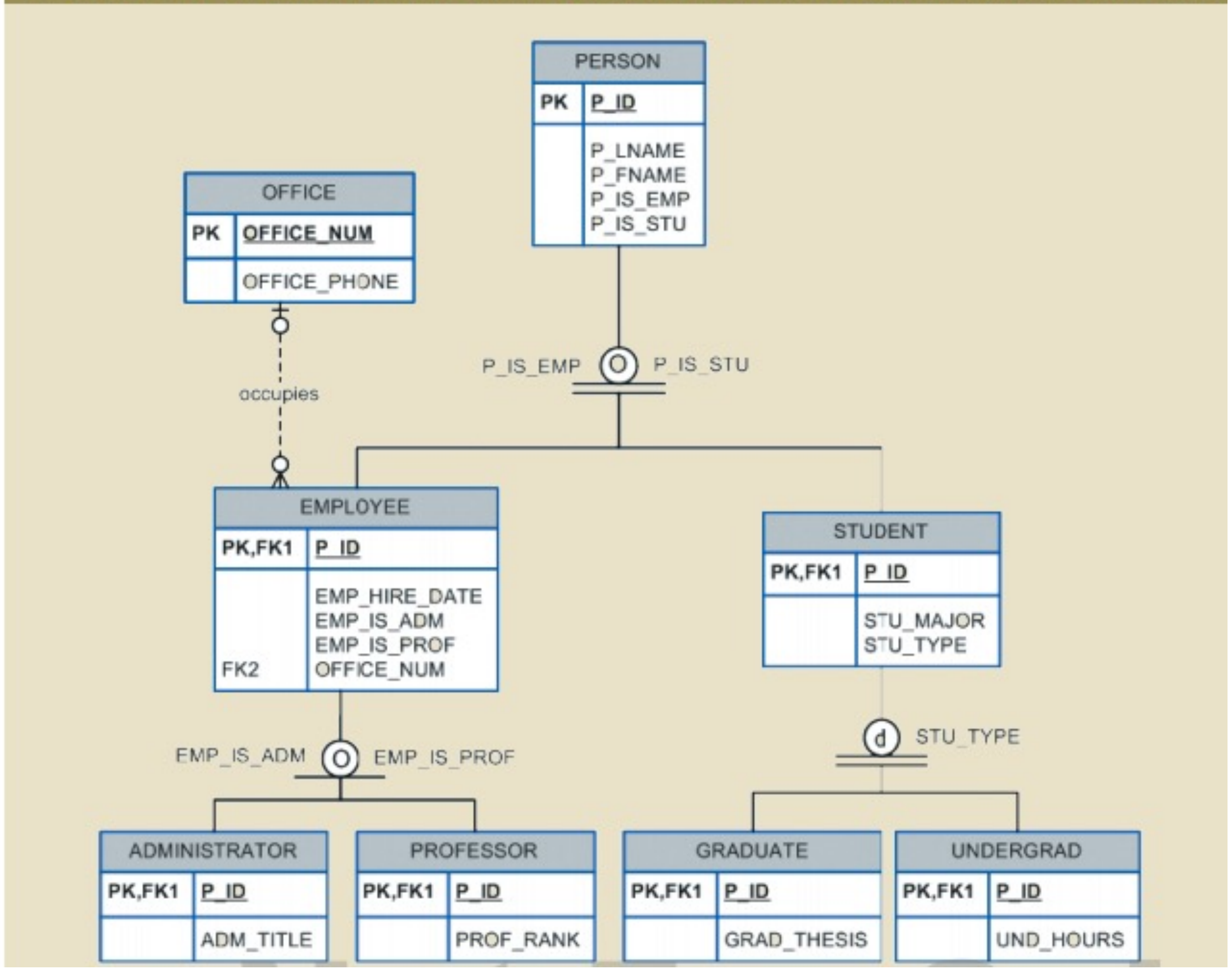
- A *specialization* hierarchy provides the means to:
  - Support *attribute inheritance*
  - Define a special supertype attribute known as the subtype *discriminator*
  - Define *disjoint* or *overlapping* constraints and *complete* or *partial* constraints

Inheritance

- Enables an entity subtype to inherit attributes and relationships of the supertype
  - All entity subtypes inherit their primary key attribute from their supertype
  - At the implementation level, supertype and its subtype(s) maintain a 1:1 relationship
  - Entity subtypes inherit all relationships in which supertype entity participates
  - Lower-level subtypes inherit all attributes and relationships from its upper-level supertypes

Inheritance … continued

FIGURE 5.4 SPECIALIZATION HIERARCHY WITH OVERLAPPING SUBTYPES



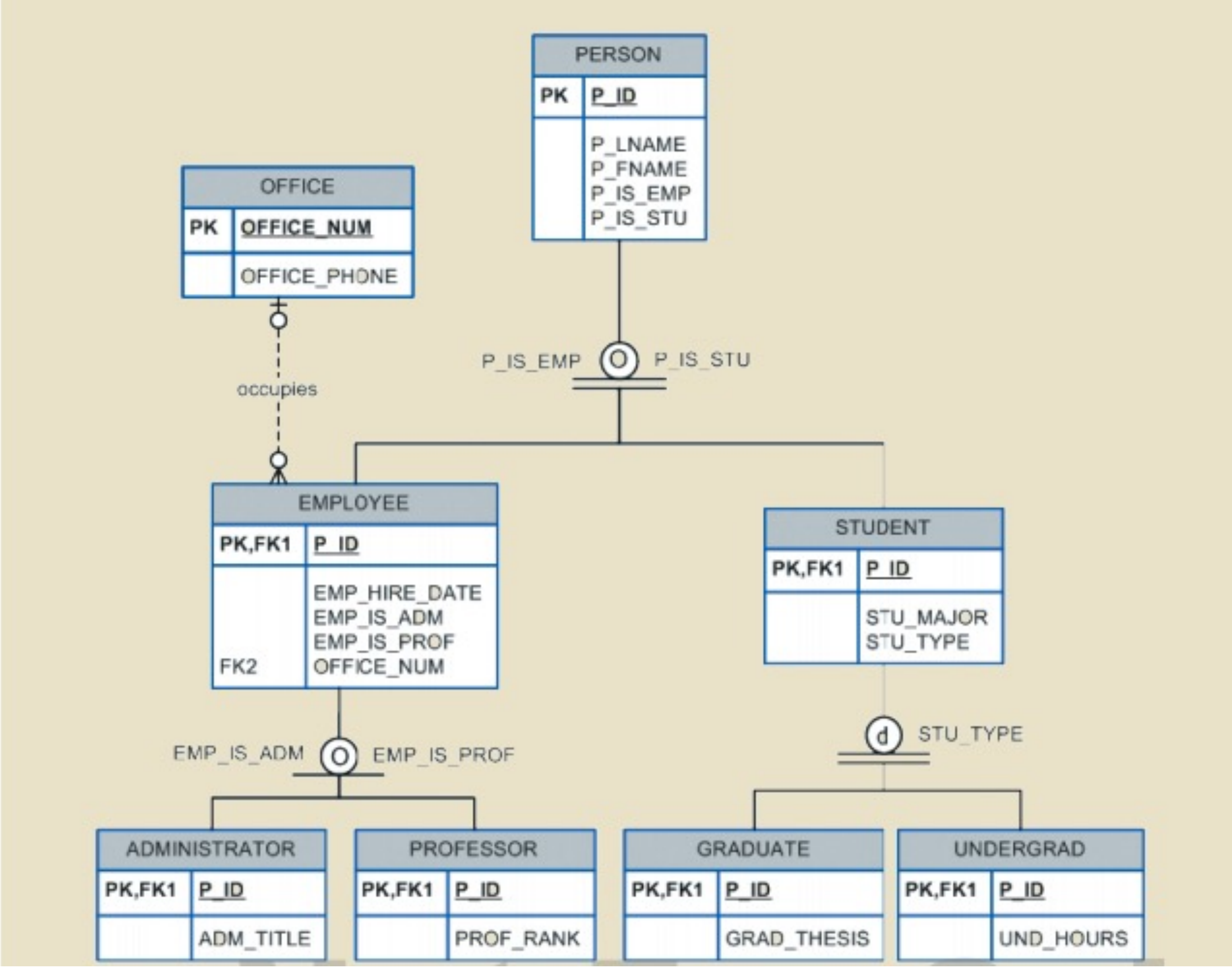
Subtype Discriminator

- Subtype Discriminator: Attribute in the supertype entity that determines to which entity subtype the supertype occurrence is related
  - Default comparison condition is the equality comparison
  - In some situations, the subtype discriminator is not necessarily based on an equality comparison

Disjoint and Overlapping Constraints … continued

- 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
- Overlapping subtypes
    - Contain nonunique subsets of the supertype entity set
    - Implementation requires the *use of one discriminator attribute for each subtype*

FIGURE 5.4 SPECIALIZATION HIERARCHY WITH OVERLAPPING SUBTYPES



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 (do not worry about)
  - 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 Integrity: Selecting Primary Keys

- Entity Integrity: Selecting Primary Keys
  - Primary keys: single attribute or a combination of attributes
    - Uniquely identifies each entity instance
    - Guarantees entity integrity => NULL cannot appear in the primary key
    - Works with foreign keys to implement relationships
- Natural Keys and Primary Keys
  - Natural key or natural identifier: real-world identifier used to uniquely identify real-world objects
    - Familiar to end users and forms part of their day-to-day business vocabulary
    - Used as the primary key of the entity being modeled

Primary Key Guidelines

- Desirable primary key characteristics
  - Non intelligent
    - Intelligent contains some information about an entity instance
      - E.g., intelligent invoice number that contains date: 20190116-CO1743C1d
  - No change over time
  - Preferably single-attribute
  - Preferably numeric
  - Security-compliant

Use of Composite Primary Keys

- Identifiers of composite associative entities
  - Each primary key combination is allowed once in M:N relationship
- Identifiers of weak entities
  - Strong identifying relationship with the parent entity
  - Represents a real-world object that is existence-dependent on another real-world object
  - Represented in the data model as two separate entities in a strong identifying relationship

Use of Composite Primary Keys … continued

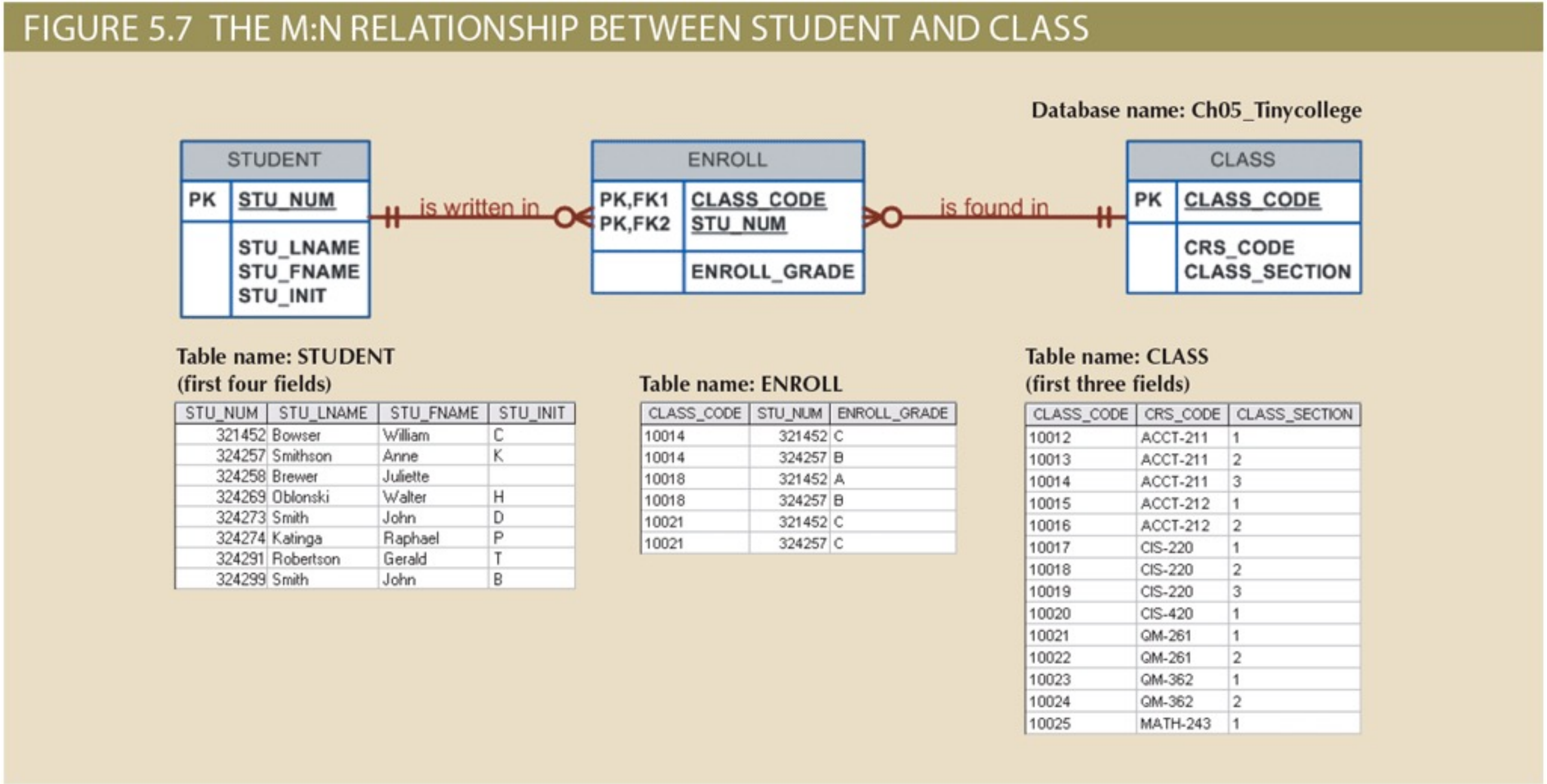


FIGURE 5.7 THE M:N RELATIONSHIP BETWEEN STUDENT AND CLASS

Database name: Ch05_Tinycollege

STUDENT

PK	STU_NUM
	STU_LNAME STU_FNAME STU_INIT

is written in

ENROLL

PK,FK1 PK,FK2	CLASS_CODE STU_NUM
	ENROLL_GRADE

is found in

CLASS

PK	CLASS_CODE
	CRS_CODE CLASS_SECTION

Table name: STUDENT
(first four fields)

STU_NUM	STU_LNAME	STU_FNAME	STU_INIT
321452	Bowser	William	C
324257	Smithson	Anne	K
324258	Brewer	Juliette	
324269	Oblonski	Walter	H
324273	Smith	John	D
324274	Katinga	Raphael	P
324291	Robertson	Gerald	T
324299	Smith	John	B

Table name: ENROLL

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
(first three fields)

CLASS_CODE	CRS_CODE	CLASS_SECTION
10012	ACCT-211	1
10013	ACCT-211	2
10014	ACCT-211	3
10015	ACCT-212	1
10016	ACCT-212	2
10017	CIS-220	1
10018	CIS-220	2
10019	CIS-220	3
10020	CIS-420	1
10021	QM-261	1
10022	QM-261	2
10023	QM-362	1
10024	QM-362	2
10025	MATH-243	1

Use of Surrogate Primary Keys

- Primary key used to simplify the identification of entity instances
  - Useful when there is no natural key
  - Helpful if selected candidate key has embedded semantic contents or is too long
- Require ensuring that the candidate key of entity in question performs properly
  - Use “unique index” and “not null” constraints

Use of Surrogate Primary Keys … continued

Table 5.4: Data Used to Keep Track of Events					
DATE	TIME_START	TIME_END	ROOM	EVENT_NAME	PARTY_OF
6/17/2018	11:00 a.m.	2:00 p.m.	Allure	Burton Wedding	60
6/17/2018	11:00 a.m.	2:00 p.m.	Bonanza	Adams Office	12
6/17/2018	3:00 p.m.	5:30 p.m.	Allure	Smith Family	15
6/17/2018	3:30 p.m.	5:30 p.m.	Bonanza	Adams Office	12
6/18/2018	1:00 p.m.	3:00 p.m.	Bonanza	Boy Scouts	33
6/18/2018	11:00 a.m.	2:00 p.m.	Allure	March of Dimes	25
6/18/2018	11:00 a.m.	12:30 p.m.	Bonanza	Smith Family	12

Flexible Design: Implementing 1:1 Relationships

- Foreign keys work with primary keys to properly implement relationships in relational model
  - Place primary key of the parent entity on the dependent entity as foreign key
- Options for selecting and placing the foreign key
  - Place a foreign key in both entities
  - Place a foreign key in one of the entities

Design Case 1: Implementing 1:1 Relationships … continued

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

Design Case 1: Implementing 1:1 Relationships … continued

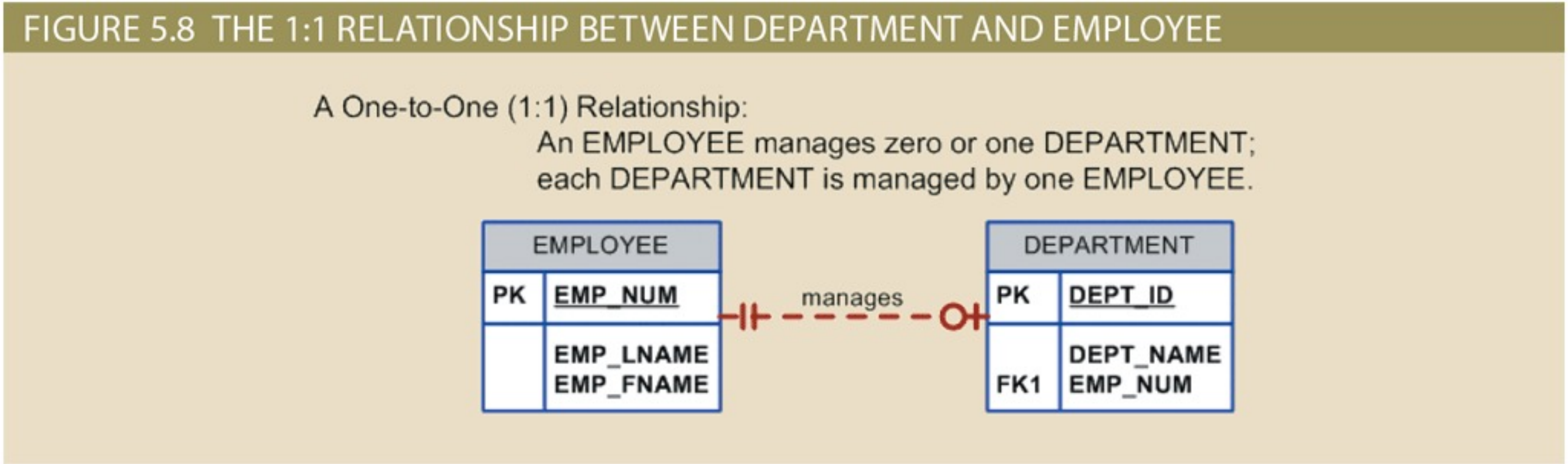


FIGURE 5.8 THE 1:1 RELATIONSHIP BETWEEN DEPARTMENT AND EMPLOYEE

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

EMPLOYEE	
PK	EMP_NUM
	EMP_LNAME EMP_FNAME

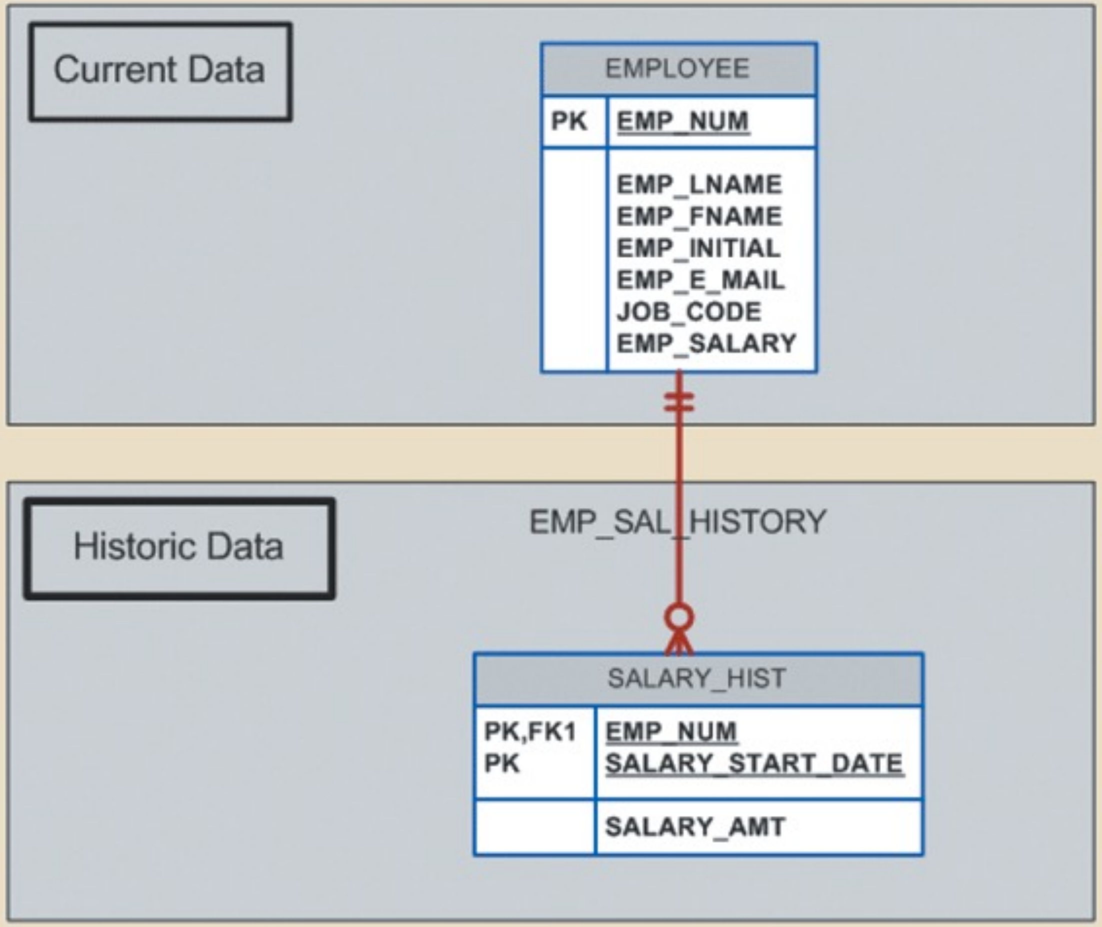
DEPARTMENT	
PK	DEPT_ID
 FK1	DEPT_NAME EMP_NUM

Design Case 2: Maintaining 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

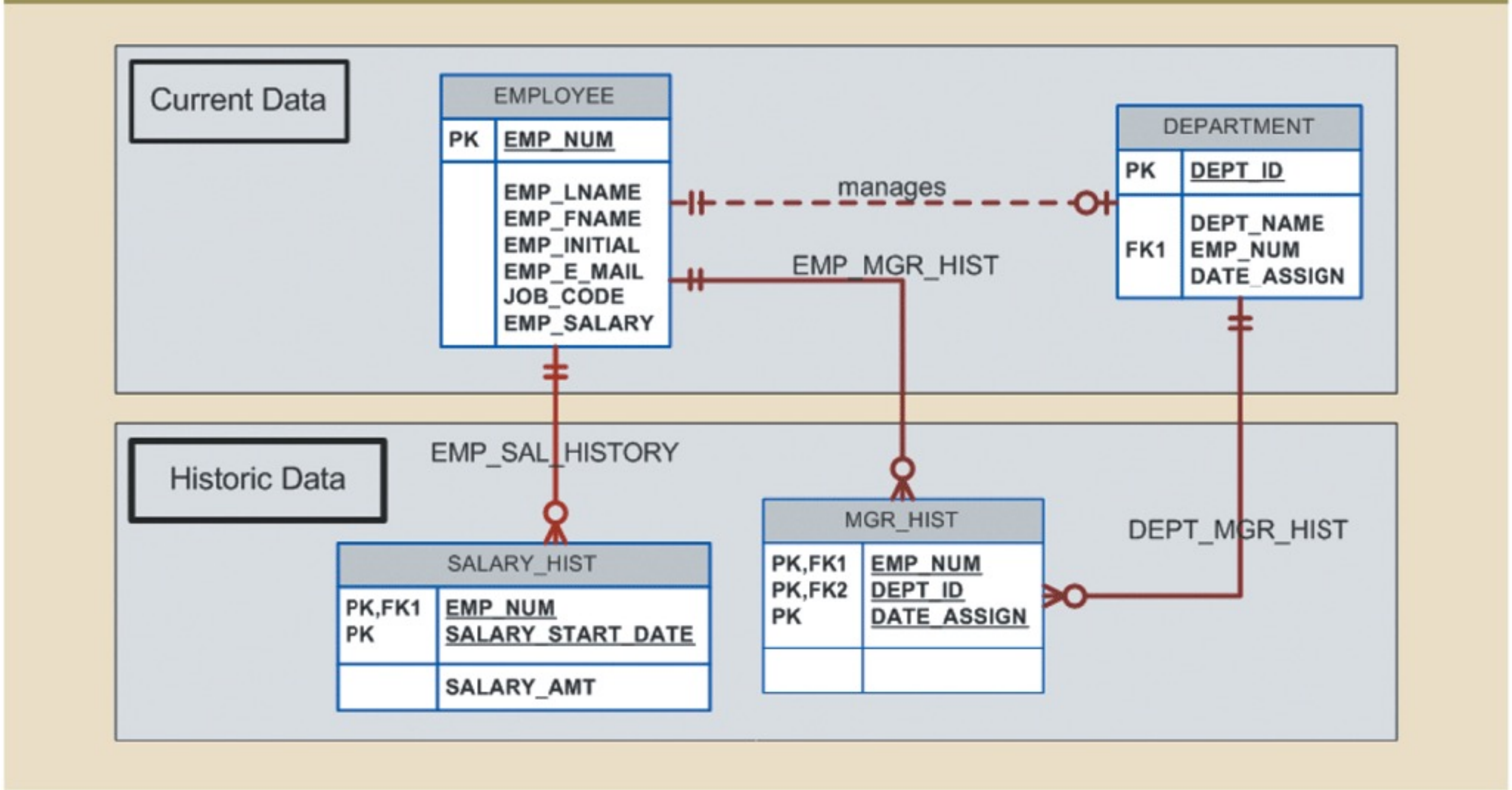
Maintaining History of Time-Variant Data … continued

FIGURE 5.9 MAINTAINING SALARY HISTORY



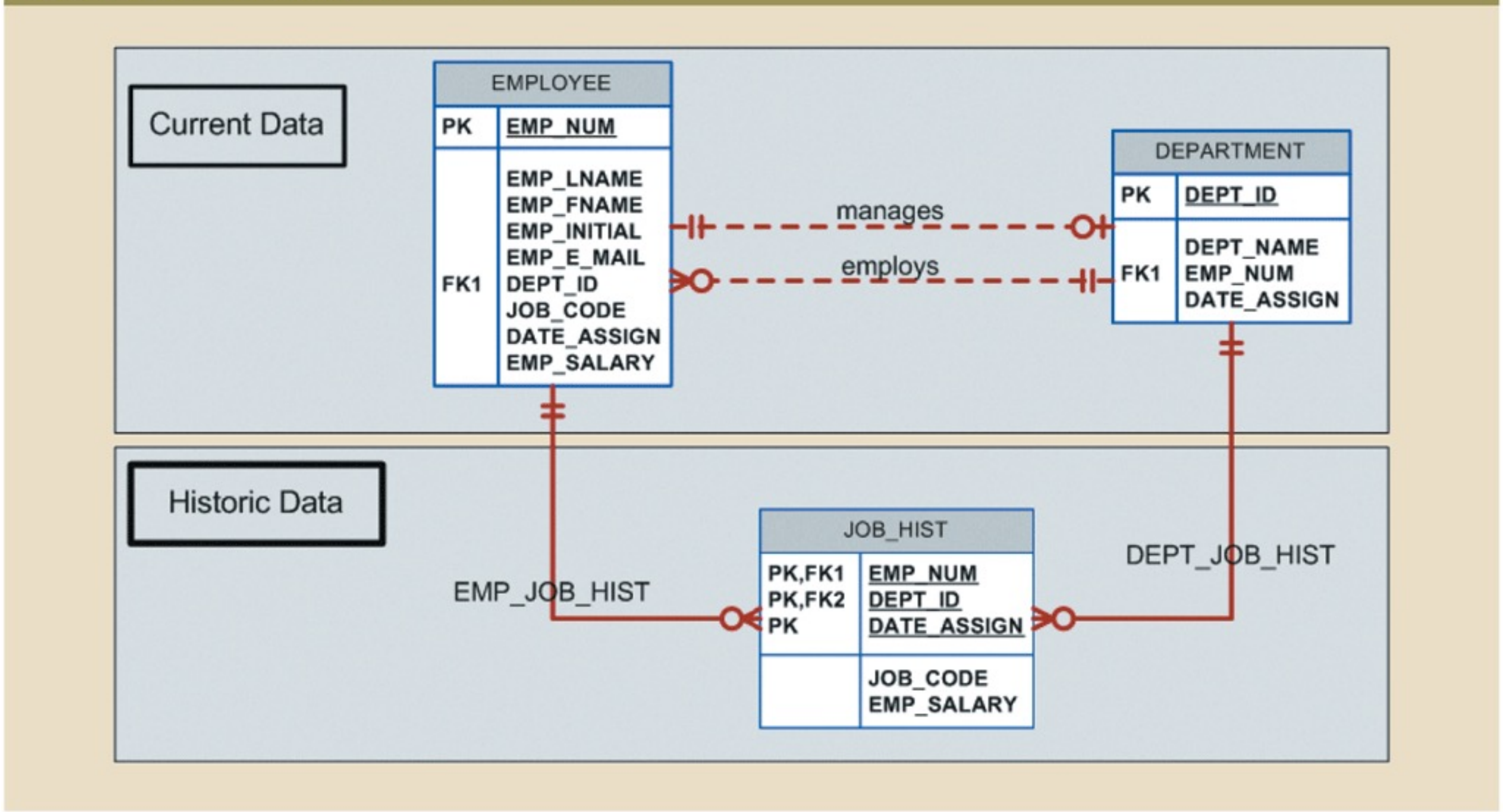
Maintaining History of Time-Variant Data … continued

FIGURE 5.10 MAINTAINING MANAGER HISTORY



Maintaining History of Time-Variant Data … continued

FIGURE 5.11 MAINTAINING JOB HISTORY



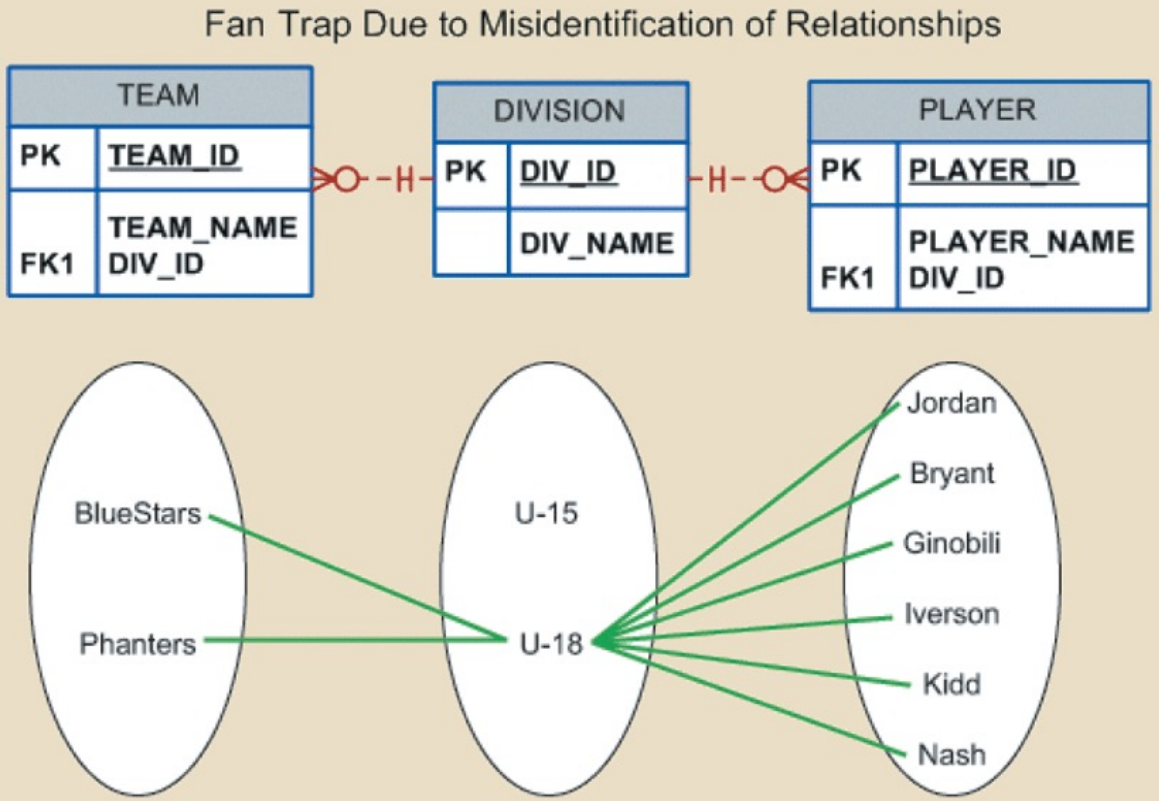
Fan Traps

Fan trap

Occurs when one entity is in two 1:M relationships to other entities

Produces an association among other entities not expressed in the model

FIGURE 5.12 INCORRECT ERD WITH FAN TRAP PROBLEM



Fan Trap Due to Misidentification of Relationships

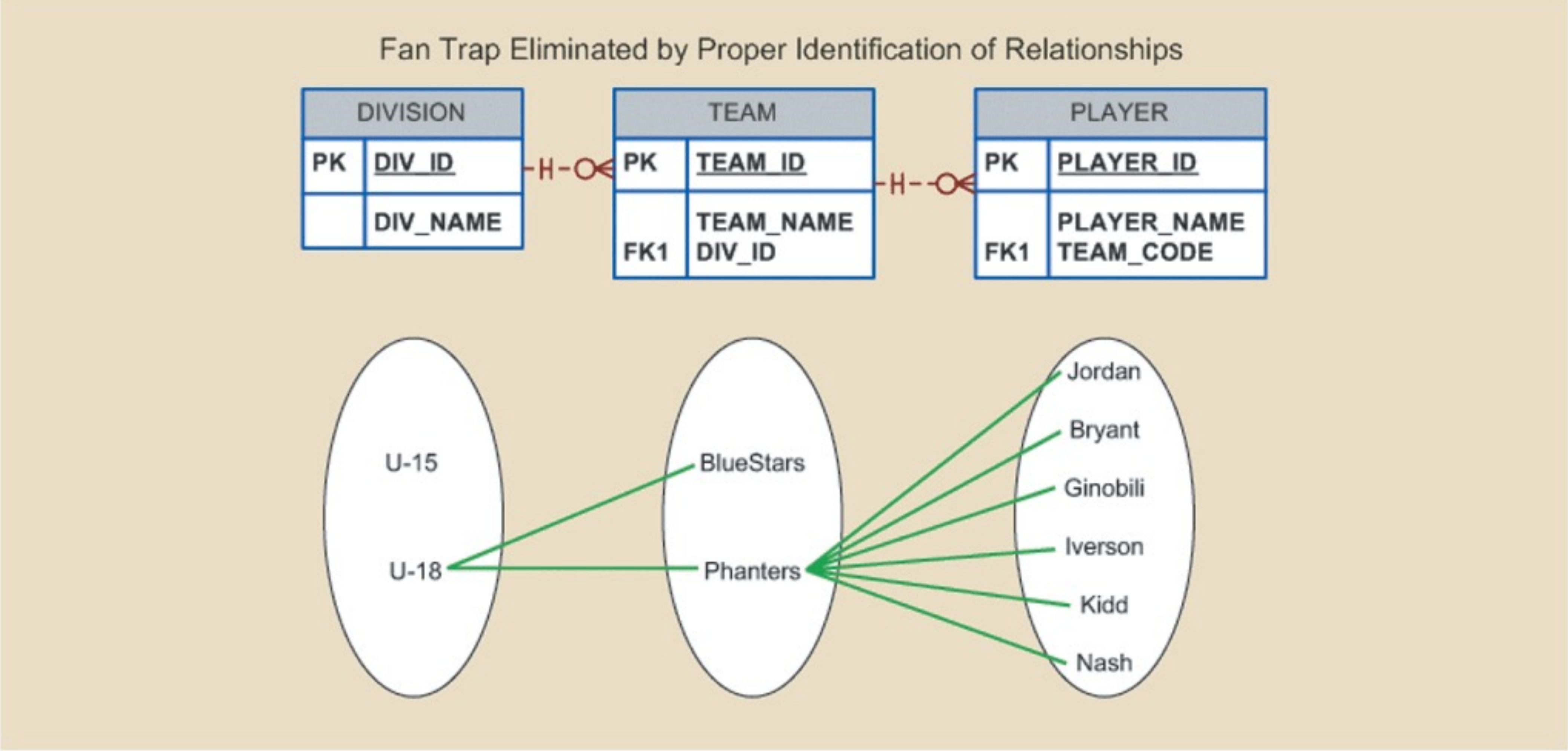
TEAM	
PK	TEAM_ID
FK1	TEAM_NAME DIV_ID

DIVISION	
PK	DIV_ID
	DIV_NAME

PLAYER	
PK	PLAYER_ID
FK1	PLAYER_NAME DIV_ID

Fan Traps … continued

FIGURE 5.13 CORRECTED ERD AFTER REMOVAL OF THE FAN TRAP



Fan Trap Eliminated by Proper Identification of Relationships

DIVISION	
PK	<u>DIV_ID</u>
	DIV_NAME

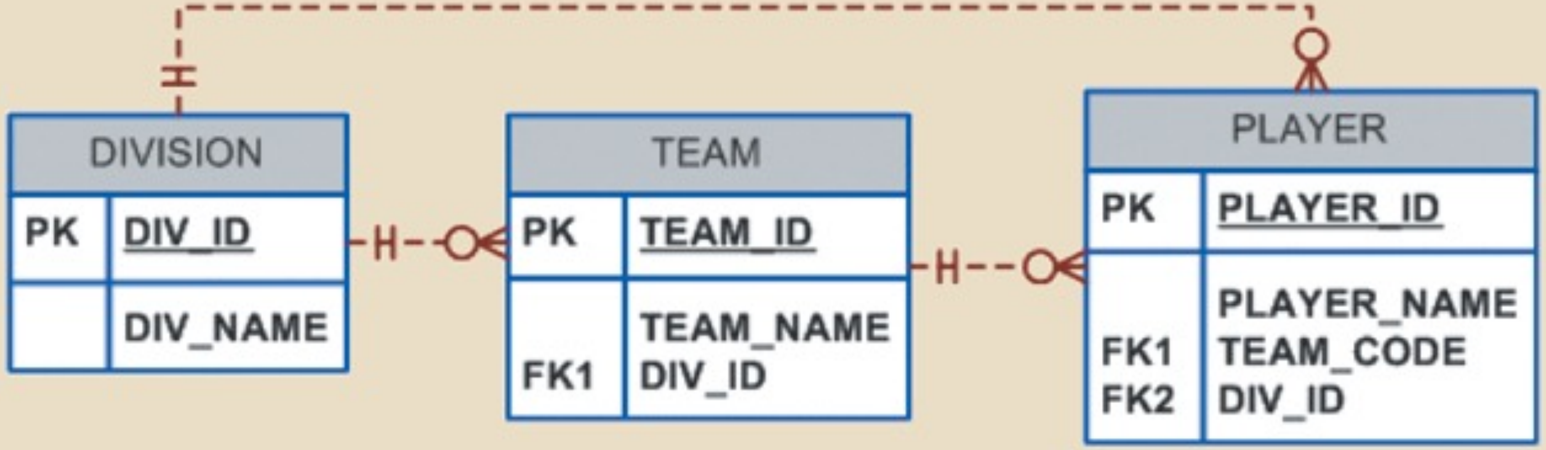
TEAM	
PK	<u>TEAM_ID</u>
 FK1	TEAM_NAME DIV_ID

PLAYER	
PK	<u>PLAYER_ID</u>
 FK1	PLAYER_NAME TEAM_CODE

Design Case 4: Redundant Relationships

- Occur when there are multiple relationship paths between related entities
  - Must remain consistent across the model
  - Help simplify the design

FIGURE 5.14 A REDUNDANT RELATIONSHIP



DIVISION

PK	DIV_ID
	DIV_NAME

TEAM

PK	TEAM_ID
	TEAM_NAME
FK1	DIV_ID

PLAYER

PK	PLAYER_ID
	PLAYER_NAME
FK1	TEAM_CODE
FK2	DIV_ID

Summary

- The extended entity relationship (EER) model adds semantics to the ER model via entity supertypes, subtypes, and clusters
  - A specialization hierarchy depicts the arrangement and relationships between entity supertypes and entity subtypes
  - Natural keys are identifiers that exist in the real world
  - Composite keys are useful to represent M:N relationships and weak (strong identifying) entities
  - Surrogate primary keys are useful when there is no natural key that makes a suitable primary key, when the primary key is a composite primary key with multiple data types, or when the primary key is too long to be usable
  - Time-variant data refers to data whose values change over time and MAY require that you keep a history of data changes
  - A fan trap occurs when you have one entity in two 1:M relationships to other entities, and there is an association among the other entities not expressed in the model

Questions and Answers (Q/A)