Database Principles: Fundamentals of Design, Implementation, and Management

Tenth Edition

Chapter 8

Data Modeling Advanced Concepts

Objectives

- In this chapter, students will learn:
  - About the extended entity relationship (EER) model
  - How entity clusters are used to represent multiple entities and relationships
  - The characteristics of good primary keys and how to select them
  - How to use flexible solutions for special data modeling cases

The Extended Entity Relationship Model

- Result of adding more semantic constructs to original entity relationship (ER) model
- Diagram using this model is called an EER diagram (EERD)

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

FIGURE 8.1 Nulls created by unique attributes

Database name: Ch08 _AirCo

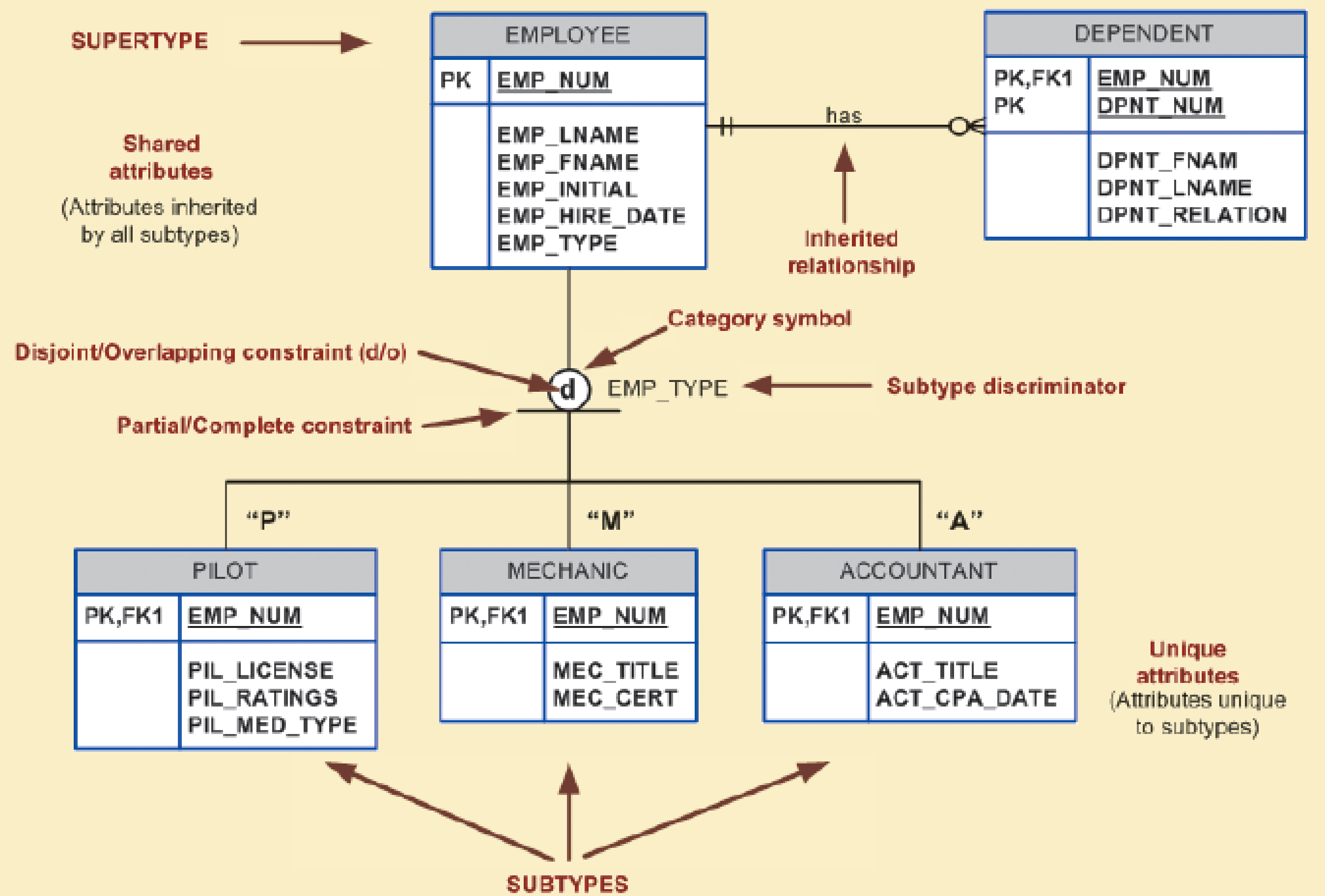
EMP_NUM	EMP_LNAME	EMP_FNAME	EMP_INITIAL	EMP_LICENSE	EMP_RATINGS	EMP_MED_TYPE	EMP_HIRE_DATE
100	Kolmycz	Xavier	T				15-Mar-88
101	Lewis	Marcos		ATP	SEL/MEL/Instr/CFII	1	25-Apr-89
102	Vandam	Jean					20-Dec-93
103	Jones	Victoria	R				28-Aug-03
104	Lange	Edith		ATP	SEL/MEL/Instr	1	20-Oct-97
105	Williams	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

SOURCE: Course Technology/Cengage Learning

Specialization Hierarchy

- Depicts arrangement of higher-level entity supertypes and lower-level entity subtypes
- Relationships described in terms of "IS-A" relationships
- Subtype exists only within context of supertype
- Every subtype has only one supertype to which it is directly related
- Can have many levels of supertype/subtype relationships

FIGURE 8.2 A specialization hierarchy



SOURCE: Course Technology/Cengage Learning

Inheritance

- Enables entity subtype to inherit attributes and relationships of supertype
- All entity subtypes inherit their primary key attribute from their supertype
- At 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 all upper-level supertypes

FIGURE 8.3 The EMPLOYEE-PILOT supertype-subtype relationship

Database name: Ch08_AirCo

Table name: EMPLOYEE

EMP_NUM	EMP_LNAME	EMP_FNAME	EMP_INITIAL	EMP_HIRE_DATE	EMP_TYPE
100	Kolmycz	Xavier	T	15-Mar-88	
101	Lewis	Marcos		25-Apr-89	P
102	Vandam	Jean		20-Dec-93	A
103	Jones	Victoria	R	28-Aug-03	
104	Lange	Edith		20-Oct-97	P
105	Williams	Gabriel	U	08-Nov-97	P
106	Duzak	Mario		05-Jan-04	P
107	Diante	Venite	L	02-Jul-97	M
108	Wiesenbach	Joni		18-Nov-95	M
109	Travis	Brett	T	14-Apr-01	P
110	Genkazi	Stan		01-Dec-03	A

Table name: PILOT

EMP_NUM	PIL_LICENSE	PIL_RATINGS	PIL_MED_TYPE
101	ATP	SEL/MEL/Instr/CFII	1
104	ATP	SEL/MEL/Instr	1
105	COM	SEL/MEL/Instr/CFI	2
106	COM	SEL/MEL/Instr	2
109	COM	SEL/MEL/SES/Instr/CFII	1

SOURCE: Course Technology/Cengage Learning

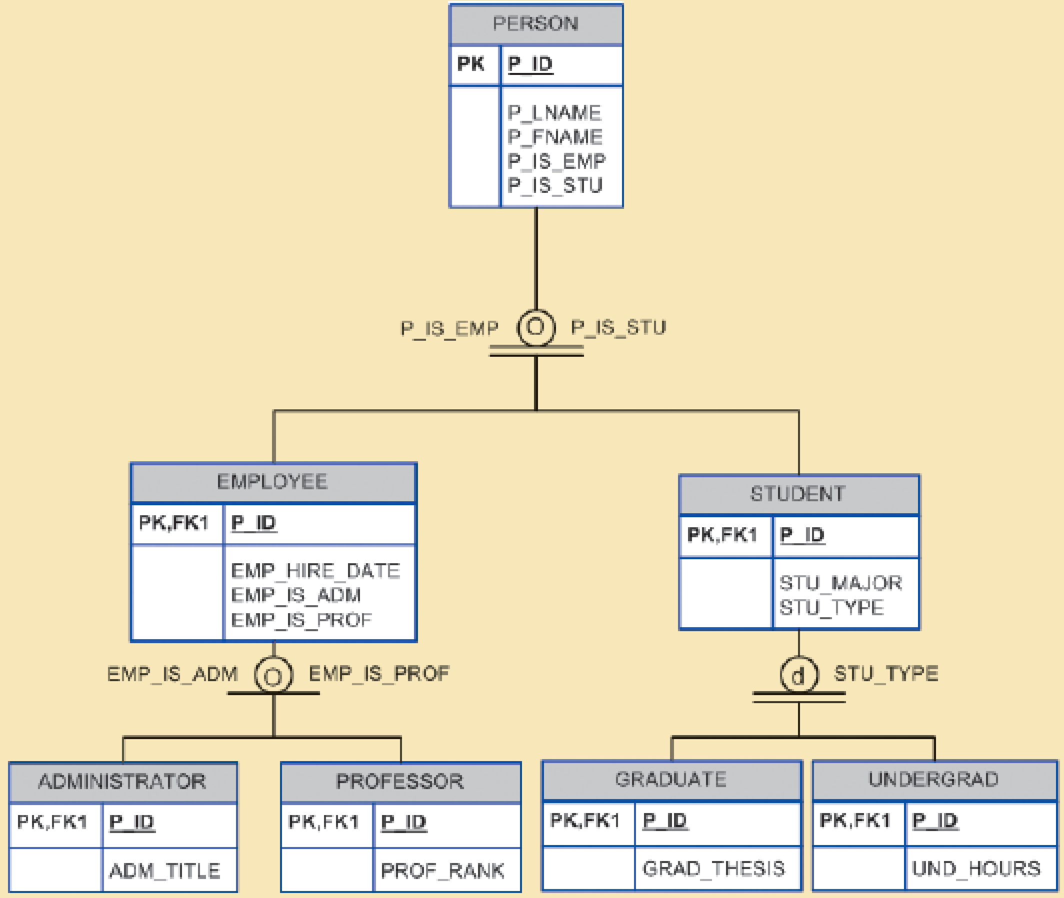
Subtype Discriminator

- Attribute in supertype entity
  - Determines to which entity subtype each supertype occurrence is related
- Default comparison condition for subtype discriminator attribute is equality comparison
- Subtype discriminator may be based on other comparison condition

Disjoint and Overlapping Constraints

- Disjoint subtypes
  - Also called nonoverlapping subtypes
  - Subtypes that contain unique subset of supertype entity set
- Overlapping subtypes
  - Subtypes that contain nonunique subsets of supertype entity set

FIGURE 8.4 Specialization hierarchy with overlapping subtypes



SOURCE: Course Technology/Cengage Learning

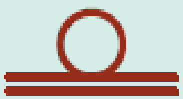
TABLE 8.1 Discriminator Attributes with Overlapping Subtypes

DISCRIMINATOR ATTRIBUTES		COMMENT
Professor	Administrator	
Y	N	The Employee is a member of the Professor subtype.
N	Y	The Employee is a member of the Administrator subtype.
Y	Y	The Employee is both a Professor and an Administrator.

Completeness Constraint

- Specifies whether entity supertype occurrence must be a member of at least one subtype
- Partial completeness
  - Symbolized by a circle over a single line
  - Some supertype occurrences are not members of any subtype
- Total completeness
  - Symbolized by a circle over a double line
  - Every supertype occurrence must be member of at least one subtype

TABLE 8.2 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.

Specialization and Generalization

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

Specialization and Generalization (cont'd.)

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

Entity Clustering

- “Virtual” entity type used to represent multiple entities and relationships in ERD
- Considered “virtual” or “abstract” because it is not actually an entity in final ERD
- Temporary entity used to represent multiple entities and relationships
- Eliminate undesirable consequences
  - Avoid display of attributes when entity clusters are used

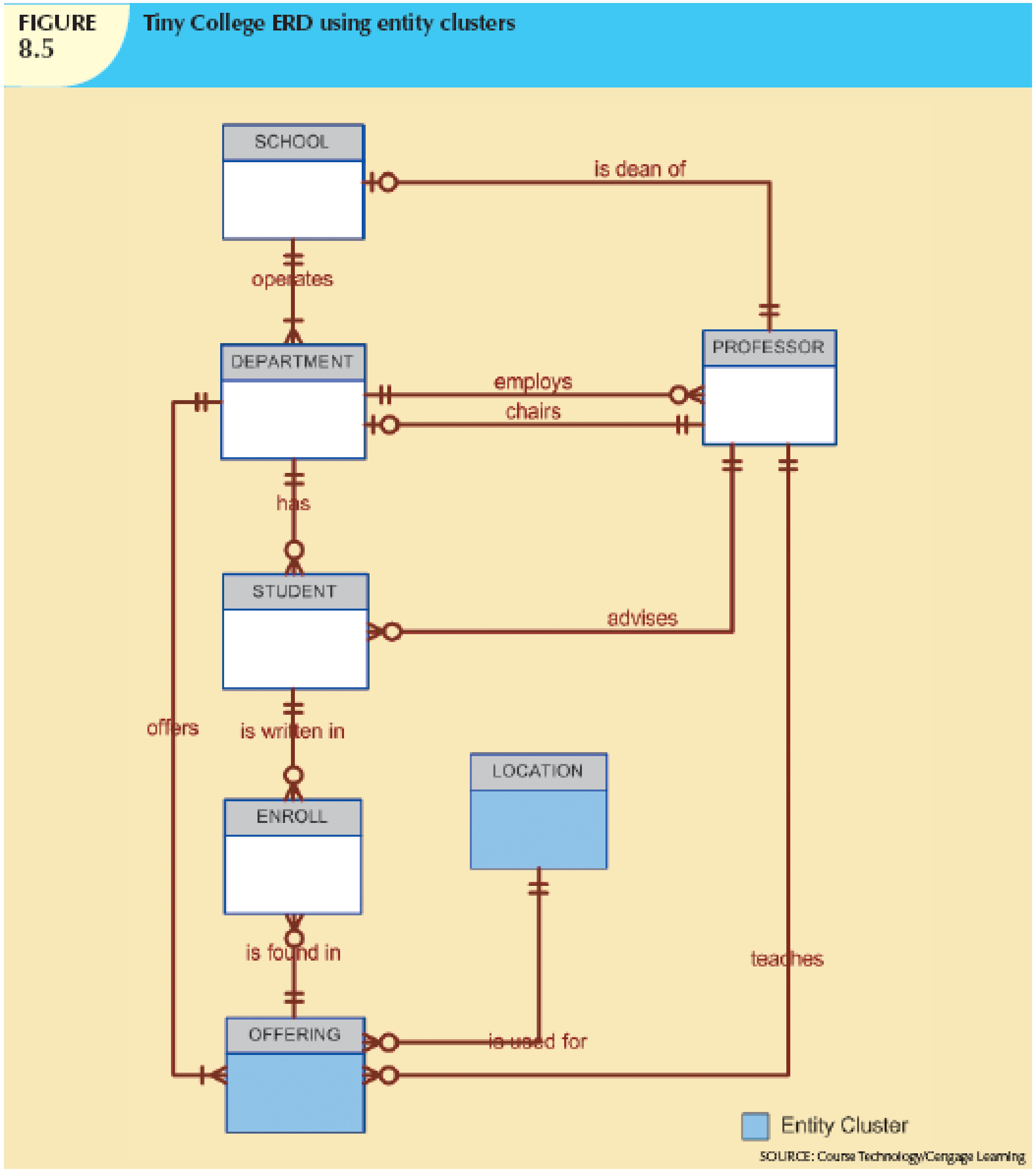


FIGURE 8.5
Tiny College ERD using entity clusters
SCHOOL
is dean of
operates
PROFESSOR
DEPARTMENT
employs
chairs
has
STUDENT
advises
offers
is written in
LOCATION
ENROLL
is found in
teaches
OFFERING
is used for
Entity Cluster
SOURCE: Course Technology/Cengage Learning

Entity Integrity: Selecting Primary Keys

- Primary key is the most important characteristic of an entity
  - Single attribute or some combination of attributes
- Primary key's function is to guarantee entity integrity
- Primary keys and foreign keys work together to implement relationships
- Properly selecting primary key has direct bearing on efficiency and effectiveness

Natural Keys and Primary Keys

- Natural key is a real-world identifier used to uniquely identify real-world objects
  – Familiar to end users and forms part of their day-to-day business vocabulary
- Generally, data modeler uses natural identifier as primary key of entity being modeled
- May instead use composite primary key or surrogate key

Primary Key Guidelines

- Attribute that uniquely identifies entity instances in an entity set
  – Could also be combination of attributes
- Main function is to uniquely identify an entity instance or row within a table
- Guarantee entity integrity, not to “describe” the entity
- Primary keys and foreign keys implement relationships among entities
  – Behind the scenes, hidden from user

TABLE 8.3 **Desirable Primary Key Characteristics**

PK CHARACTERISTIC	RATIONALE
Unique values	The PK must uniquely identify each entity instance. A primary key must be able to guarantee unique values. It cannot contain nulls.
Nonintelligent	The PK should not have embedded semantic meaning other than to uniquely identify each entity instance. An attribute with embedded semantic meaning is probably better used as a descriptive characteristic of the entity than as an identifier. For example, a student ID of 650973 would be preferred over *Smith, Martha L.* as a primary key identifier.
No change over time	If an attribute has semantic meaning, it might be subject to updates, which is why names do not make good primary keys. If *Vickie Smith* is the primary key, what happens if she changes her name when she gets married? If a primary key is subject to change, the foreign key values must be updated, thus adding to the database work load. Furthermore, changing a primary key value means that you are basically changing the identity of an entity. In short, the PK should be permanent and unchangeable.
Preferably single-attribute	A primary key should have the minimum number of attributes possible (irreducible). Single-attribute primary keys are desirable but not required. Single-attribute primary keys simplify the implementation of foreign keys. Having multiple-attribute primary keys can cause primary keys of related entities to grow through the possible addition of many attributes, thus adding to the database work load and making (application) coding more cumbersome.
Preferably numeric	Unique values can be better managed when they are numeric, because the database can use internal routines to implement a counter-style attribute that automatically increments values with the addition of each new row. In fact, most database systems include the ability to use special constructs, such as Autonumber in Microsoft Access, to support self-incrementing primary key attributes.
Security-compliant	The selected primary key must not be composed of any attribute(s) that might be considered a security risk or violation. For example, using a Social Security number as a PK in an EMPLOYEE table is not a good idea.

When to Use Composite Primary Keys

- Composite primary keys useful in two cases:
  - As identifiers of composite entities
    - In which each primary key combination is allowed once in M:N relationship
  - As identifiers of weak entities
    - In which weak entity has a strong identifying relationship with the parent entity
- Automatically provides benefit of ensuring that there cannot be duplicate values

FIGURE 8.6 The M:N relationship between STUDENT and CLASS

Database name: Ch08_Tinycollege

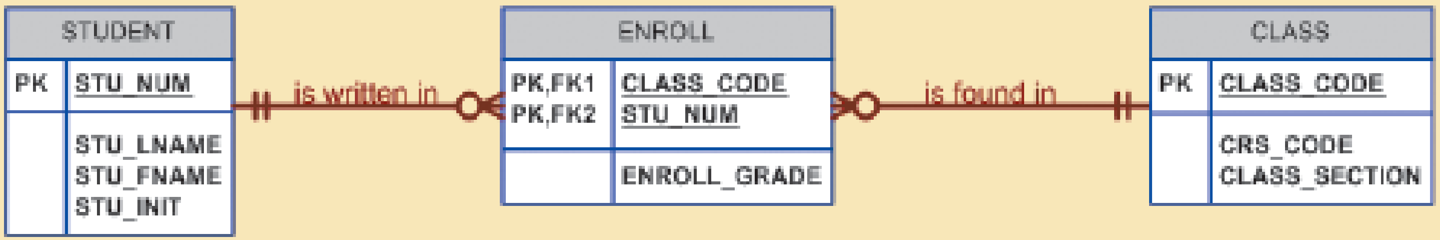


Table name: STUDENT
(first four fields)

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

SOURCE: Course Technology/Cengage Learning

When to Use Composite Primary Keys (cont'd.)

- When used as identifiers of weak entities normally used to represent:
  - Real-world object that is existent-dependent on another real-world object
  - Real-world object that is represented in data model as two separate entities in strong identifying relationship
- Dependent entity exists only when it is related to parent entity

When To Use Surrogate Primary Keys

- Especially helpful when there is:
  - No natural key
  - Selected candidate key has embedded semantic contents
  - Selected candidate key is too long or cumbersome

When To Use Surrogate Primary Keys (cont'd.)

- If you use surrogate key:
  - Ensure that candidate key of entity in question performs properly
  - Use "unique index" and "not null" constraints

TABLE 8.4 Data Used to Keep Track of Events

DATE	TIME_START	TIME_END	ROOM	EVENT_NAME	PARTY_OF
6/17/2012	11:00AM	2:00PM	Allure	Burton Wedding	60
6/17/2012	11:00AM	2:00PM	Bonanza	Adams Office	12
6/17/2012	3:00PM	5:30PM	Allure	Smith Family	15
6/17/2012	3:30PM	5:30PM	Bonanza	Adams Office	12
6/18/2012	1:00PM	3:00PM	Bonanza	Boy Scouts	33
6/18/2012	11:00AM	2:00PM	Allure	March of Dimes	25
6/18/2012	11:00AM	12:30PM	Bonanza	Smith Family	12

Design Cases: Learning Flexible Database Design

- Data modeling and design requires skills acquired through experience
- Experience acquired through practice
- Four special design cases that highlight:
  - Importance of flexible design
  - Proper identification of primary keys
  - Placement of foreign keys

Design Case 1: Implementing 1:1 Relationships

- Foreign keys work with primary keys to properly implement relationships in relational model
- Put primary key of the “one” side on the “many” side as foreign key
  - Primary key: parent entity
  - Foreign key: dependent entity

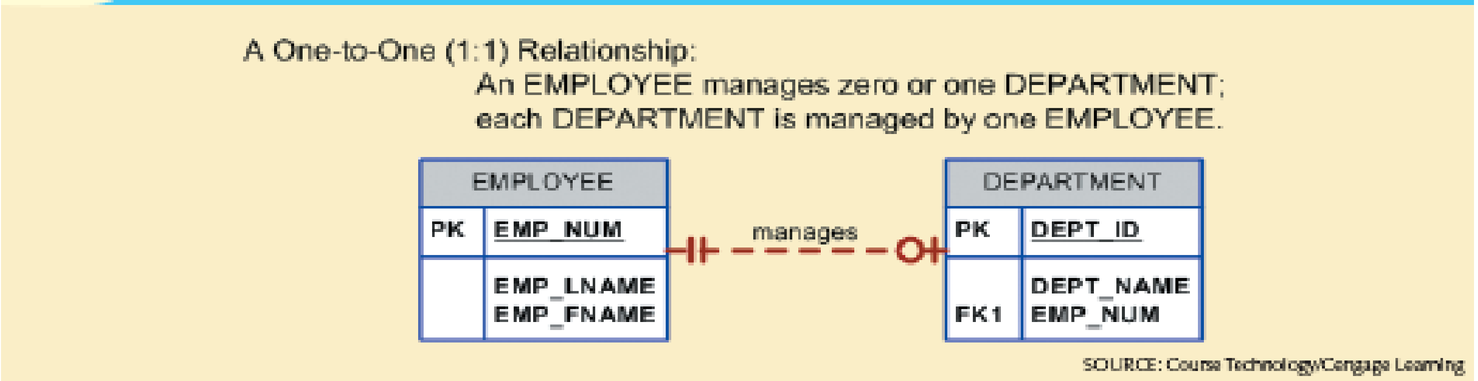
Design Case 1: Implementing 1:1 Relationships (cont'd.)

- In 1:1 relationship, there are two options:
  - Place a foreign key in both entities (not recommended)
  - Place a foreign key in one of the entities
    - Primary key of one of the two entities appears as foreign key of other

TABLE 8.5 Selection of Foreign Key in a 1:1 Relationship

CASE	ER RELATIONSHIP CONSTRAINTS	ACTION
I	One side is mandatory and the other side is optional.	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.

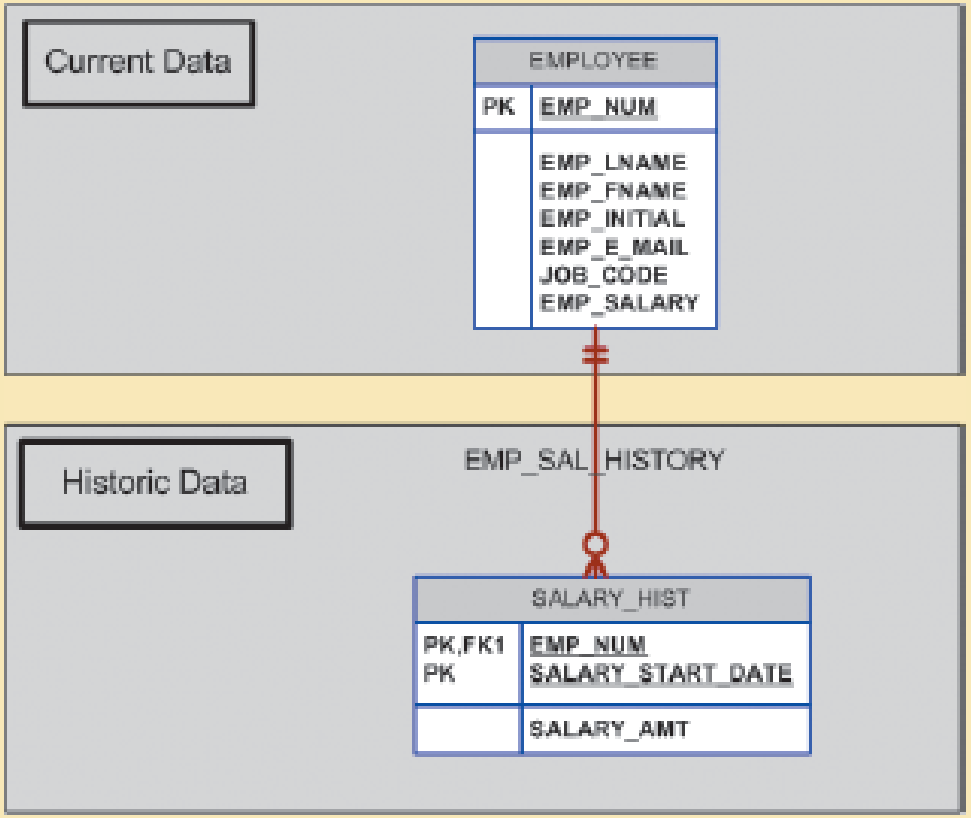
FIGURE 8.7 **The 1:1 relationship between DEPARTMENT and EMPLOYEE**



Design Case 2: Maintaining History of Time-Variant Data

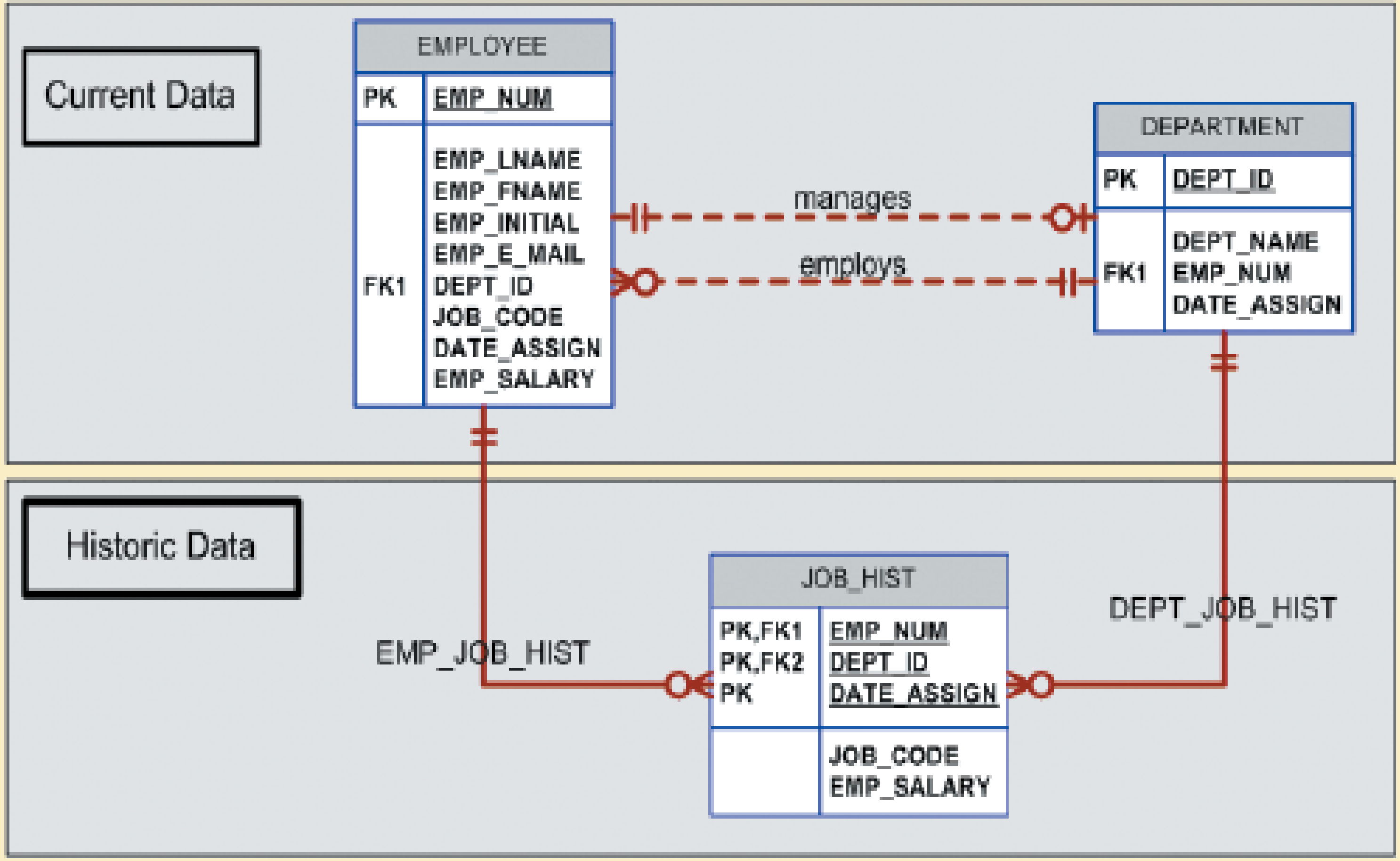
- Normally, existing attribute values are replaced with new value without regard to previous value
- Time-variant data:
  - Values change over time
  - Must keep a history of data changes
- Keeping history of time-variant data equivalent to having a multivalued attribute in your entity
- Must create new entity in 1:M relationships with original entity
- New entity contains new value, date of change

FIGURE 8.8 Maintaining salary history



SOURCE: Course Technology/Cengage Learning

FIGURE 8.10 Maintaining job history



SOURCE: Course Technology/Cengage Learning

Design Case 3: Fan Traps

- Design trap occurs when relationship is improperly or incompletely identified
  - Represented in a way not consistent with the real world
- Most common design trap is known as fan trap
- 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 8.11 Incorrect ERD with fan trap problem

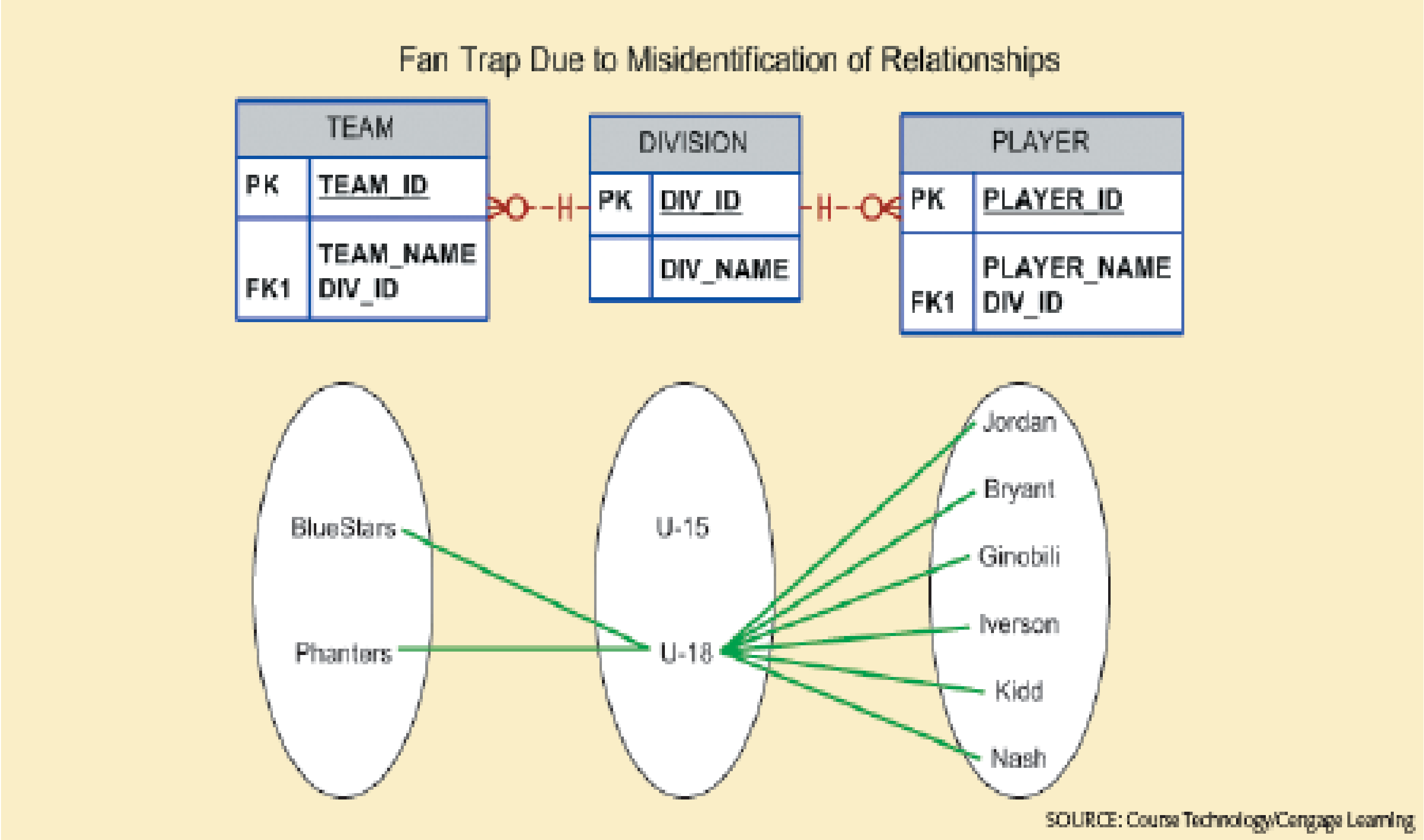
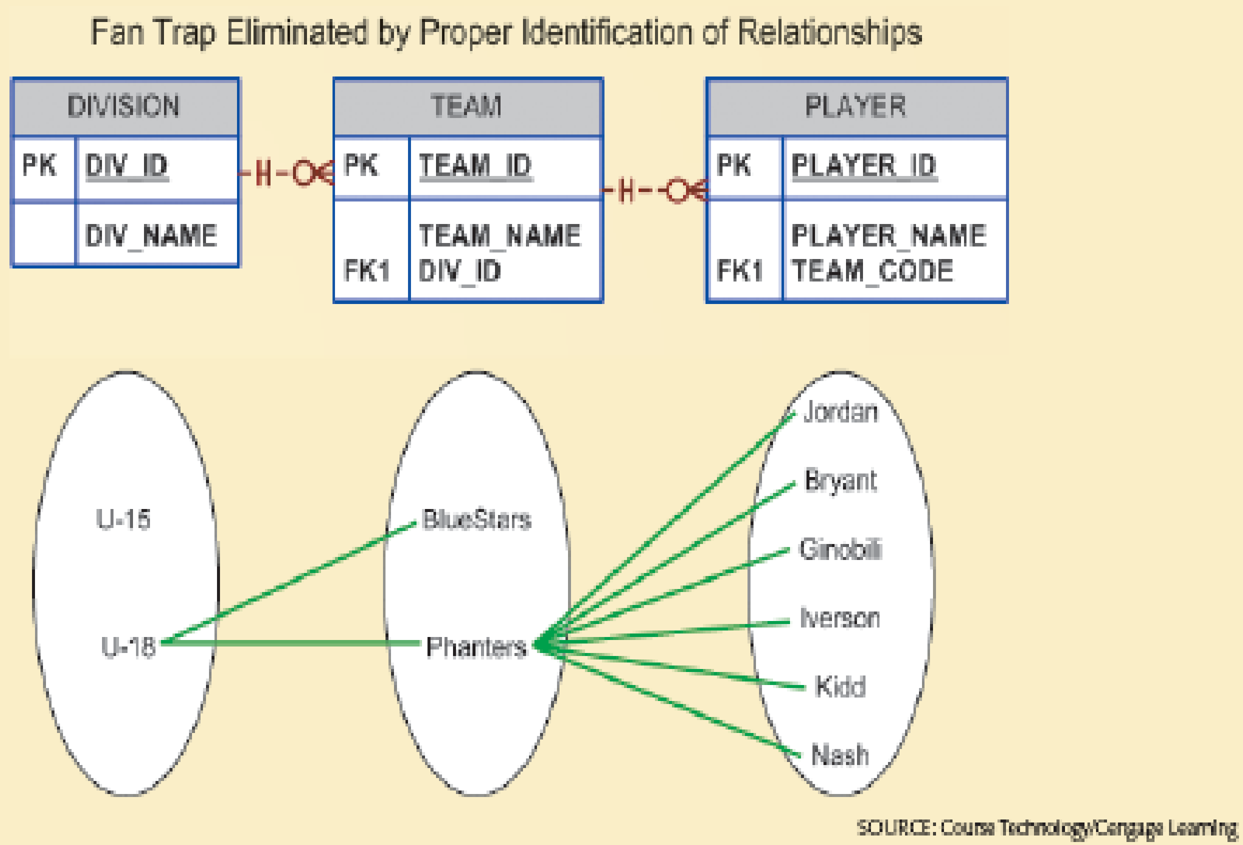


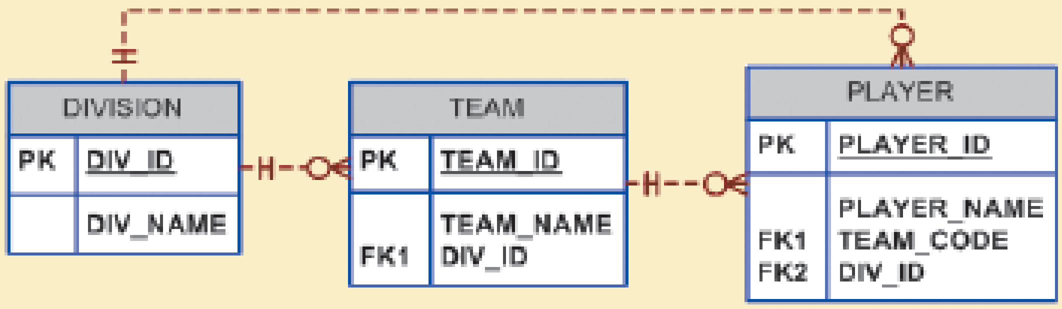
FIGURE 8.12 Corrected ERD after removal of the fan trap



Design Case 4: Redundant Relationships

- Redundancy is seldom a good thing in database environment
- Occurs when there are multiple relationship paths between related entities
- Main concern is that redundant relationships remain consistent across model
- Some designs use redundant relationships to simplify the design

FIGURE 8.13 A redundant relationship



SOURCE: Course Technology/Cengage Learning

Summary

- Extended entity relationship (EER) model adds semantics to ER model
  - Adds semantics via entity supertypes, subtypes, and clusters
  - Entity supertype is a generic entity type related to one or more entity subtypes
- Specialization hierarchy
  - Depicts arrangement and relationships between entity supertypes and entity subtypes
- Inheritance means an entity subtype inherits attributes and relationships of supertype

Summary (cont'd.)

- Subtype discriminator determines which entity subtype the supertype occurrence is related to:
  - Partial or total completeness
  - Specialization vs. generalization
- Entity cluster is "virtual" entity type
  - Represents multiple entities and relationships in ERD
  - Formed by combining multiple interrelated entities and relationships into a single object

Summary (cont'd.)

- Natural keys are identifiers that exist in real world
  - Sometimes make good primary keys
- Characteristics of primary keys:
  - Must have unique values
  - Should be nonintelligent
  - Must not change over time
  - Preferably numeric or composed of single attribute

Summary (cont'd.)

- Composite keys are useful to represent
  - M:N relationships
  - Weak (strong-identifying) entities
- Surrogate primary keys are useful when no suitable natural key makes primary key
- In a 1:1 relationship, place the PK of mandatory entity:
  - As FK in optional entity
  - As FK in entity that causes least number of nulls
  - As FK where the role is played

Summary (cont'd.)

- Time-variant data
  - Data whose values change over time
  - Requires keeping a history of changes
- To maintain history of time-variant data:
  - Create entity containing the new value, date of change, other time-relevant data
  - Entity maintains 1:M relationship with entity for which history maintained

Summary (cont'd.)

- Fan trap:
  - One entity in two 1:M relationships to other entities
  - Association among the other entities not expressed in model
- Redundant relationships occur when multiple relationship paths between related entities
  - Main concern is that they remain consistent across the model
- Data modeling checklist provides way to check that the ERD meets minimum requirements