11e

Chapter 6
Normalization of Database Tables

Learning Objectives

- In this chapter, students will learn:
  - What normalization is and what role it plays in the database design process
  - About the normal forms 1NF, 2NF, 3NF and 4NF
  - How normal forms can be transformed from lower normal forms to higher normal forms
  - That normalization and ER modeling are used concurrently to produce a good database design
  - That some situations require denormalization to generate information efficiently

Normalization

- Process of evaluating and correcting table structures to minimize data redundancies
  - Reduces data anomalies
- After the initial design is complete, the designer can use normalization to analyze the relationships among the attributes within each entity and determine if the structure can be improved through normalization.
- Works through a series of stages called normal forms:
  - First normal form (1NF)
  - Second normal form (2NF)
  - Third normal form (3NF)

Normalization

- Structural point of view of normal forms
  - Higher normal forms are better than lower normal forms (i.e., 2NF is better than 1NF; 3NF is better than 2NF)
- For most business database design purposes, 3NF is as high as we need to go in normalization process
  - Properly designed 3NF structures meet the requirement of fourth normal form (4NF)
- **Denormalization**: Produces a lower normal form
  - Results in increased performance and greater data redundancy

Need for Normalization

- Used while designing a new database structure
  - Analyzes the relationship among the attributes within each entity
  - Determines if the structure can be improved
- Improves the existing data structure and creates an appropriate database design

Normalization Process

- Each project has its own project number, name, assigned employees, and so on.
- Each employee has an employee number, name, and job classification, such as engineer or computer technician.

Normalization Process

PROJECT NUMBER	PROJECT NAME	EMPLOYEE NUMBER	EMPLOYEE NAME	JOB CLASS	CHARGE/ HOUR	HOURS BILLED	TOTAL CHARGE
15	Evergreen	103	June E. Arbough	Elec. Engineer	$ 84.50	23.8	$ 2,011.10
		101	John G. News	Database Designer	$105.00	19.4	$ 2,037.00
		105	Alice K. Johnson *	Database Designer	$105.00	35.7	$ 3,748.50
		106	William Smithfield	Programmer	$ 35.75	12.6	$ 450.45
		102	David H. Senior	Systems Analyst	$ 96.75	23.8	$ 2,302.65
				Subtotal			**$10,549.70**
18	Amber Wave	114	Annelise Jones	Applications Designer	$ 48.10	24.6	$ 1,183.26
		118	James J. Frommer	General Support	$ 18.36	45.3	$ 831.71
		104	Anne K. Ramoras *	Systems Analyst	$ 96.75	32.4	$ 3,134.70
		112	Darlene M. Smithson	DSS Analyst	$ 45.95	44.0	$ 2,021.80
				Subtotal			**$ 7,171.47**
22	Rolling Tide	105	Alice K. Johnson	Database Designer	$105.00	64.7	$ 6,793.50
		104	Anne K. Ramoras	Systems Analyst	$ 96.75	48.4	$ 4,682.70
		113	Delbert K. Joenbrood *	Applications Designer	$ 48.10	23.6	$ 1,135.16
		111	Geoff B. Wabash	Clerical Support	$ 26.87	22.0	$ 591.14
		106	William Smithfield	Programmer	$ 35.75	12.8	$ 457.60
				Subtotal			**$13,660.10**
25	Starflight	107	Maria D. Alonzo	Programmer	$ 35.75	24.6	$ 879.45
		115	Travis B. Bawangi	Systems Analyst	$ 96.75	45.8	$ 4,431.15
		101	John G. News *	Database Designer	$105.00	56.3	$ 5,911.50
		114	Annelise Jones	Applications Designer	$ 48.10	33.1	$ 1,592.11
		108	Ralph B. Washington	Systems Analyst	$ 96.75	23.6	$ 2,283.30
		118	James J. Frommer	General Support	$ 18.36	30.5	$ 559.98
		112	Darlene M. Smithson	DSS Analyst	$ 45.95	41.4	$ 1,902.33
				Subtotal			**$17,559.82**
				Total			**$48,941.09**

Note: A * indicates the project leader.

Normalization Process

Table name: RPT_FORMAT

PROJ_NUM	PROJ_NAME	EMP_NUM	EMP_NAME	JOB_CLASS	CHG_HOUR	HOURS
15	Evergreen	103	June E. Arbough	Elect. Engineer	84.50	23.8
		101	John G. News	Database Designer	105.00	19.4
		105	Alice K. Johnson *	Database Designer	105.00	35.7
		106	William Smithfield	Programmer	35.75	12.6
		102	David H. Senior	Systems Analyst	96.75	23.8
18	Amber Wave	114	Annelise Jones	Applications Designer	48.10	24.6
		118	James J. Frommer	General Support	18.36	45.3
		104	Anne K. Ramoras *	Systems Analyst	96.75	32.4
		112	Darlene M. Smithson	DSS Analyst	45.95	44.0
22	Rolling Tide	105	Alice K. Johnson	Database Designer	105.00	64.7
		104	Anne K. Ramoras	Systems Analyst	96.75	48.4
		113	Delbert K. Joenbrood *	Applications Designer	48.10	23.6
		111	Geoff B. Wabash	Clerical Support	26.87	22.0
		106	William Smithfield	Programmer	35.75	12.8
25	Starflight	107	Maria D. Alonzo	Programmer	35.75	24.6
		115	Travis B. Bawangi	Systems Analyst	96.75	45.8
		101	John G. News *	Database Designer	105.00	56.3
		114	Annelise Jones	Applications Designer	48.10	33.1
		108	Ralph B. Washington	Systems Analyst	96.75	23.6
		118	James J. Frommer	General Support	18.36	30.5
		112	Darlene M. Smithson	DSS Analyst	45.95	41.4

Normalization Process

- The project number (PROJ_NUM) contains nulls.
- The table entries invite data inconsistencies.

Table name: RPT_FORMAT

PROJ_NUM	PROJ_NAME	EMP_NUM	EMP_NAME	JOB_CLASS	CHG_HOUR	HOURS
15	Evergreen	103	June E. Arbough	Elect. Engineer	84.50	23.8
		101	John G. News	Database Designer	105.00	19.4
		105	Alice K. Johnson *	Database Designer	105.00	35.7
		106	William Smithfield	Programmer	35.75	12.6
		102	David H. Senior	Systems Analyst	96.75	23.8
18	Amber Wave	114	Annelise Jones	Applications Designer	48.10	24.6
		118	James J. Frommer	General Support	18.36	45.3
		104	Anne K. Ramoras *	Systems Analyst	96.75	32.4
		112	Darlene M. Smithson	DSS Analyst	45.95	44.0
22	Rolling Tide	105	Alice K. Johnson	Database Designer	105.00	64.7
		104	Anne K. Ramoras	Systems Analyst	96.75	48.4
		113	Delbert K. Joenbrood *	Applications Designer	48.10	23.6
		111	Geoff B. Wabash	Clerical Support	26.87	22.0
		106	William Smithfield	Programmer	35.75	12.8
25	Starflight	107	Maria D. Alonzo	Programmer	35.75	24.6
		115	Travis B. Bawangi	Systems Analyst	96.75	45.8
		101	John G. News *	Database Designer	105.00	56.3
		114	Annelise Jones	Applications Designer	48.10	33.1
		108	Ralph B. Washington	Systems Analyst	96.75	23.6
		118	James J. Frommer	General Support	18.36	30.5
		112	Darlene M. Smithson	DSS Analyst	45.95	41.4

Normalization Process

- The table displays data redundancies that yield the following anomalies:
  - Update anomalies. Modifying the JOB_CLASS for employee number 105 requires many potential alterations, one for each EMP_NUM=105
  - Insertion anomalies. Just to complete a row definition, a new employee must be assigned to a project. If the employee is not yet assigned, a phantom project must be created to complete the employee data entry.
  - Deletion anomalies. Suppose that only one employee is associated with a given project. If that employee leaves the company and the employee data are deleted, the project information will also be deleted.

Normalization Process

- Objective is to ensure that each table conforms to the concept of well-formed relations
  - Each table represents a single subject (A COURSE table will contain only data the directly pertain to courses.)
  - No data item will be unnecessarily stored in more than one table (ensure that the data are updated in only one place.)
  - All nonprime attributes in a table are dependent on the primary key
  - Each table is void of insertion, update, and deletion anomalies

Normalization Process

- Ensures that all tables are in at least 3NF
- Higher forms are not likely to be encountered in business environment
- Works one relation at a time
- Starts by:
  - Identifying the dependencies of a relation (table)
  - Progressively breaking the relation into new set of relations

Table 6.2 - Normal Forms

NORMAL FORM	CHARACTERISTIC	SECTION
First normal form (1NF)	Table format, no repeating groups, and PK identified	6.3.1
Second normal form (2NF)	1NF and no partial dependencies	6.3.2
Third normal form (3NF)	2NF and no transitive dependencies	6.3.3
Boyce-Codd normal form (BCNF)	Every determinant is a candidate key (special case of 3NF)	6.6.1
Fourth normal form (4NF)	3NF and no independent multivalued dependencies	6.6.2

Cengage Learning © 2015

Functional Dependence Concepts

Concept	Definition
Functional dependence	The attribute B is fully functionally dependent on the attribute A if each value of A determines one and only one value of B.
Functional dependence (Generalized definition)	Attribute A determines attribute B if all of the rows in the table that agree in value for attribute A also agree in value for attribute B.
Fully functional dependence (composite key)	If attribute B is functionally dependent on a composite key A but not on any Subset of that composite key, the attribute B is fully functionally dependent on A.

Types of Functional Dependencies

- **Partial dependency**: Functional dependence in which the determinant is only part of the primary key
  - If (A,B)→(C,D), B→C, and (A,B) is the primary key. Then, the functional dependence B→C is a partial dependency because only part of the primary key (B) is needed to determine the values of C.
- **Transitive dependency**: An attribute functionally depends on another nonkey attribute.
  - X→Y, Y→Z, and X is the primary. Then, X→Z is a transitive dependency because X determines the value of Z via Y.

Conversion to First Normal Form

- **Repeating group**: Derives its name from the fact that a group of multiple entries of same type can exist for any single key attribute occurrence
  - Existence proves the presence of data redundancies
- Enable reducing data redundancies
- Normalization is three-step procedure
  - Eliminate the repeating groups
  - Identify the primary key
  - Identify all dependencies

Conversion to First Normal Form (continued)

- Step 1: Eliminate the Repeating Groups
  - Present data in tabular format, where each cell has single value and there are no repeating groups
  - Eliminate repeating groups, eliminate nulls by making sure that each repeating group attribute contains an appropriate data value

Conversion to First Normal Form (continued)

Table name: DATA_ORG_1NF

PROJ_NUM	PROJ_NAME	EMP_NUM	EMP_NAME	JOB_CLASS	CHG_HOUR	HOURS
15	Evergreen	103	June E. Arbough	Elect. Engineer	84.50	23.8
15	Evergreen	101	John G. News	Database Designer	105.00	19.4
15	Evergreen	105	Alice K. Johnson *	Database Designer	105.00	35.7
15	Evergreen	106	William Smithfield	Programmer	35.75	12.6
15	Evergreen	102	David H. Senior	Systems Analyst	96.75	23.8
18	Amber Wave	114	Annelise Jones	Applications Designer	48.10	24.6
18	Amber Wave	118	James J. Frommer	General Support	18.36	45.3
18	Amber Wave	104	Anne K. Ramoras *	Systems Analyst	96.75	32.4
18	Amber Wave	112	Darlene M. Smithson	DSS Analyst	45.95	44.0
22	Rolling Tide	105	Alice K. Johnson	Database Designer	105.00	64.7
22	Rolling Tide	104	Anne K. Ramoras	Systems Analyst	96.75	48.4
22	Rolling Tide	113	Delbert K. Joenbrood *	Applications Designer	48.10	23.6
22	Rolling Tide	111	Geoff B. Wabash	Clerical Support	26.87	22.0
22	Rolling Tide	106	William Smithfield	Programmer	35.75	12.8
25	Starflight	107	Maria D. Alonzo	Programmer	35.75	24.6
25	Starflight	115	Travis B. Bawangi	Systems Analyst	96.75	45.8
25	Starflight	101	John G. News *	Database Designer	105.00	56.3
25	Starflight	114	Annelise Jones	Applications Designer	48.10	33.1
25	Starflight	108	Ralph B. Washington	Systems Analyst	96.75	23.6
25	Starflight	118	James J. Frommer	General Support	18.36	30.5
25	Starflight	112	Darlene M. Smithson	DSS Analyst	45.95	41.4

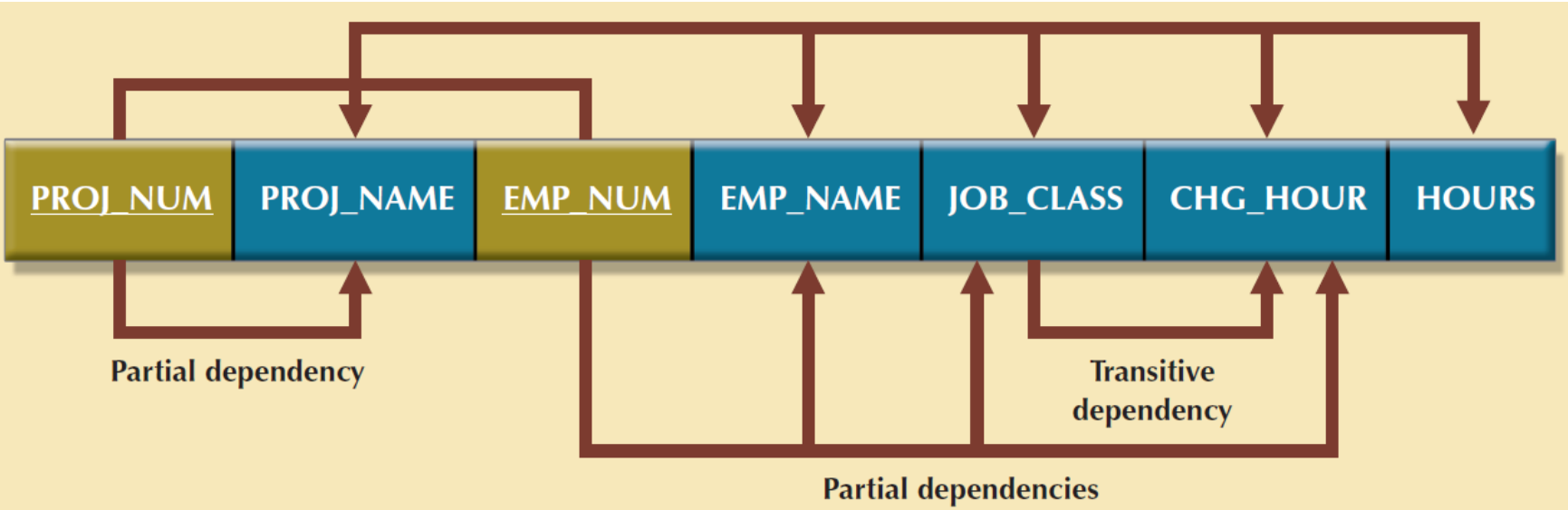
Conversion to First Normal Form (continued)

- Step 2: Identify the Primary Key
  - Primary key must uniquely identify attribute value
  - New key must be composed

Conversion to First Normal Form (continued)

- Step 3: Identify All Dependencies
  - Dependencies can be depicted with help of a diagram
  - Dependency diagram:
    - Depicts all dependencies found within given table structure
    - Helpful in getting bird's-eye view of all relationships among table's attributes
    - Makes it less likely that will overlook an important dependency

Figure 6.3 - First Normal Form (1NF) Dependency Diagram



1NF (PROJ_NUM, EMP_NUM, PROJ_NAME, EMP_NAME, JOB_CLASS, CHG_HOURS, HOURS)

PARTIAL DEPENDENCIES:
(PROJ_NUM → PROJ_NAME)
(EMP_NUM → EMP_NAME, JOB_CLASS, CHG_HOUR)

TRANSITIVE DEPENDENCY:
(JOB_CLASS → CHG_HOUR)

Cengage Learning © 2015

Conversion to First Normal Form

- 1NF describes tabular format in which:
  - All key attributes are defined
  - There are no repeating groups in the table
  - All attributes are dependent on the primary key
- All relational tables satisfy 1NF requirements
- Some tables contain partial dependencies
  - Dependencies based on only part of the primary key
  - Sometimes used for performance reasons, but should be used with caution
  - Subject to data redundancies and various anomalies

Conversion to Second Normal Form

- Relational database design can be improved by converting the database into second normal form (2NF)
- Occurs only when the 1NF has a composite primary key. If the 1NF has a single-attribute primary key, then the table is automatically in 2NF.
- Two Steps
  - Make new tables to eliminate partial dependencies
  - Reassign corresponding dependent attributes

Conversion to Second Normal Form (continued)

- Step 1: Write Each Key Component on a Separate Line
  - Write each key component on separate line, then write original (composite) key on last line
    - Ex: PROJ_NUM

      EMP_NUM

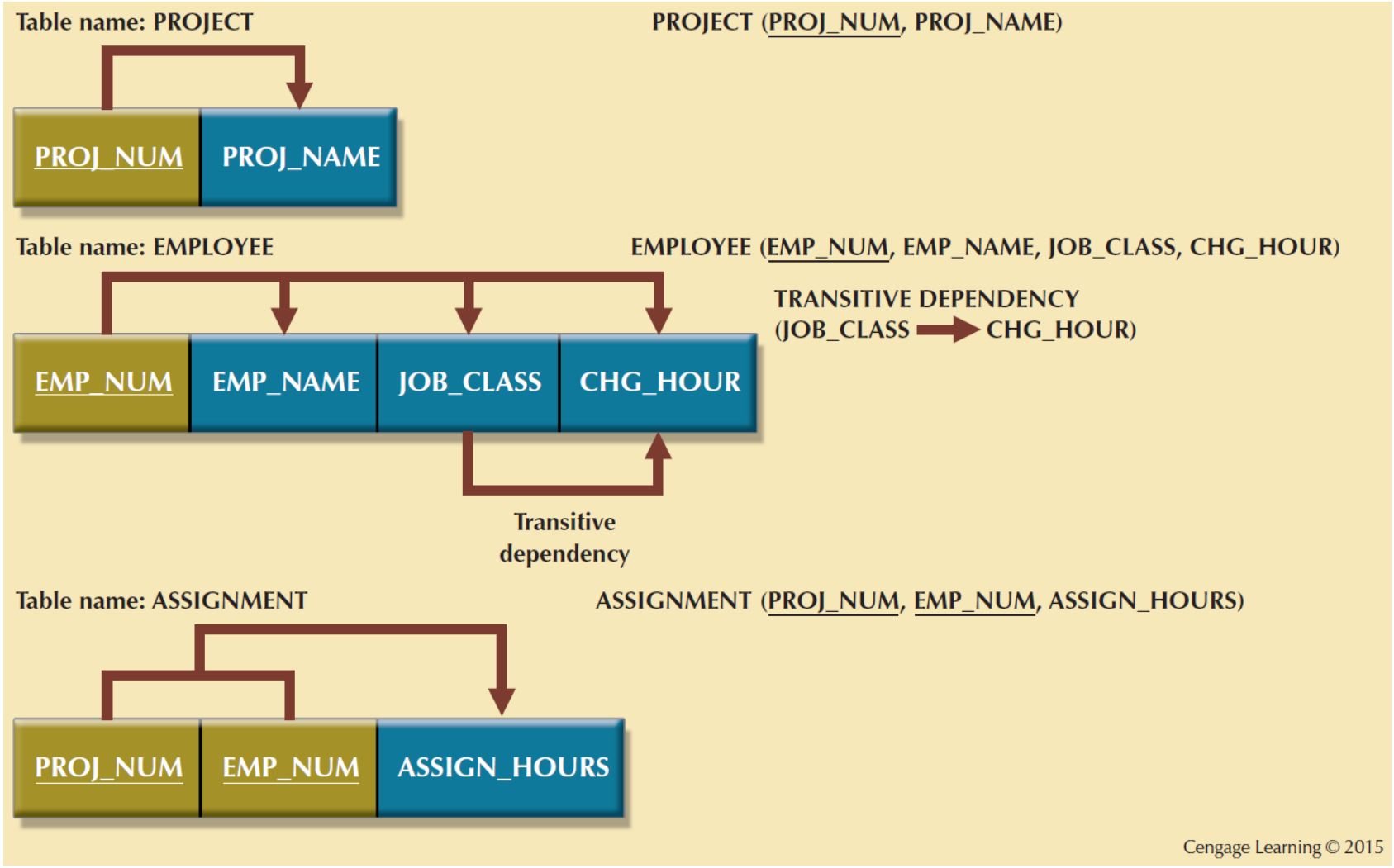
      PROJ_NUM EMP_NUM
  - Each component will become key in new table.
    - Ex: The original table is now divided into three tables

      (PROJECT, EMPLOYEE, ASSIGNMENT)

Conversion to Second Normal Form (continued)

- Step 2: Assign Corresponding Dependent Attributes
  - Determine those attributes that are dependent on other attributes (functional dependencies)
  - The attributes that are dependent in a partial dependency are removed from the original table and placed in the new table.
  - Any attributes that are not dependent in a partial dependency will remain in the original table.
  - At this point, most anomalies have been eliminated

Figure 6.4 - Second Normal Form (2NF) Conversion Results



Conversion to Second Normal Form (continued)

- Table is in 2NF when it:
  - Is in 1NF
  - Includes no partial dependencies
    - No attribute is dependent on only portion of primary key
- A table in 2NF can exhibit transitive dependency.

Conversion to Third Normal Form

- Make new tables to eliminate transitive dependencies
- Data anomalies created are easily eliminated by completing three steps
- Step 1: Identify Each New Determinant
  - For every transitive dependency, write its determinant as PK for new table
    - Determinant
      - Any attribute whose value determines other values within a row

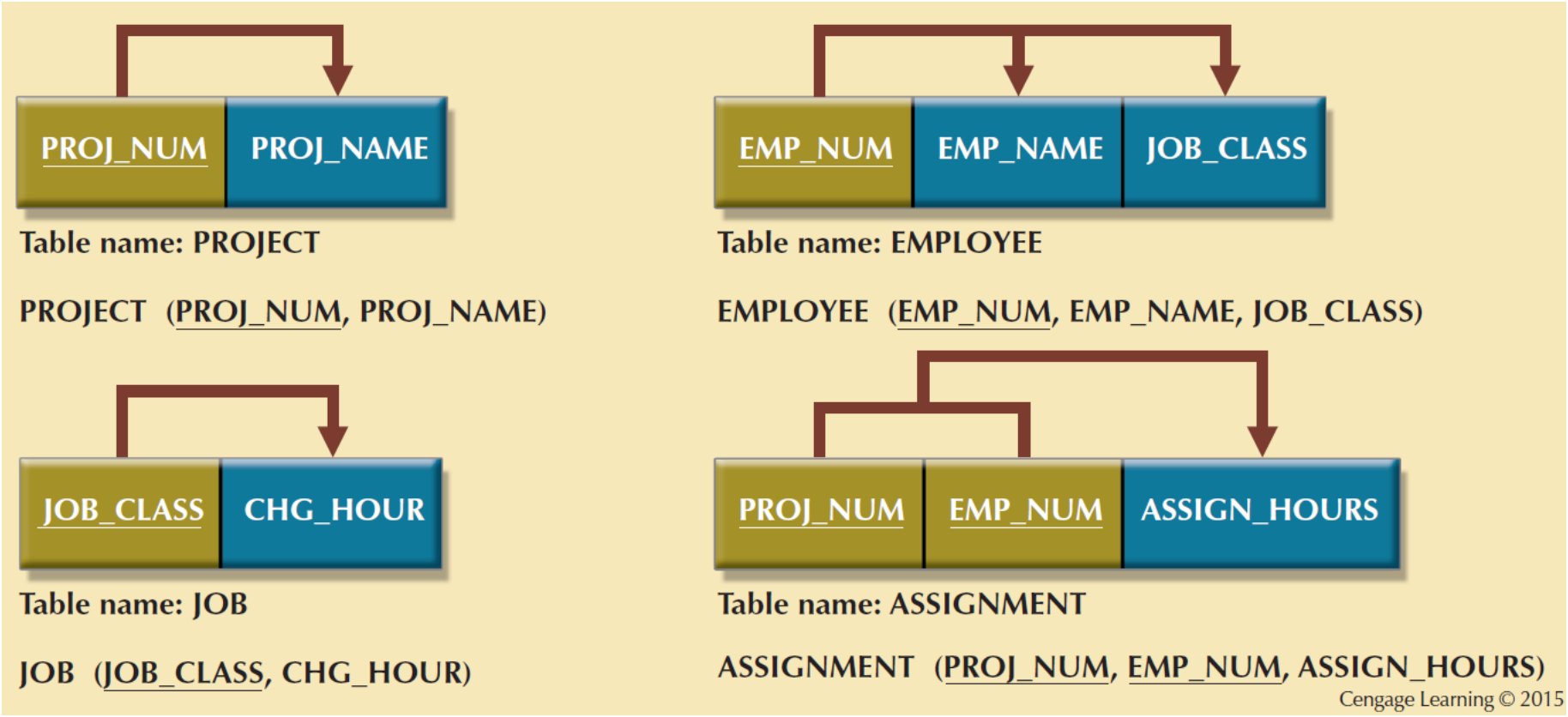
Conversion to Third Normal Form (continued)

- Step 2: Identify the Dependent Attributes
  - Identify attributes dependent on each determinant identified in Step 1 and identify dependency
  - Name table to reflect its contents and function

Conversion to Third Normal Form (continued)

- Step 3: Remove the Dependent Attributes from Transitive Dependencies
  - Eliminate all dependent attributes in transitive relationship(s) from each of the tables that have such a transitive relationship
  - Draw new dependency diagram to show all tables defined in Steps 1–3
  - Check new tables as well as tables modified in Step 3 to make sure that each table has determinant and that no table contains inappropriate dependencies

Figure 6.5 - Third Normal Form (3NF) Conversion Results



Conversion to Third Normal Form (continued)

- A table is in third normal form (3NF) when both of the following are true:
  - It is in 2NF
  - It contains no transitive dependencies

Improving the Design

- Table structures are cleaned up to eliminate troublesome initial partial and transitive dependencies
- Normalization cannot, by itself, be relied on to make good designs
- It is valuable because its use helps eliminate data redundancies

Requirements for Good Normalized Set of Tables

- Evaluate PK assignments and naming conventions
- Refine attribute atomicity
  - **Atomic attribute**: Cannot be further subdivided
  - **Atomicity**: Characteristic of an atomic attribute
- Identify new attributes and new relationships
- Refine primary keys as required for data granularity
  - **Granularity**: Level of detail represented by the values stored in a table's row
- Maintain historical accuracy and evaluate using derived attributes

Figure 6.6 - The Completed Database

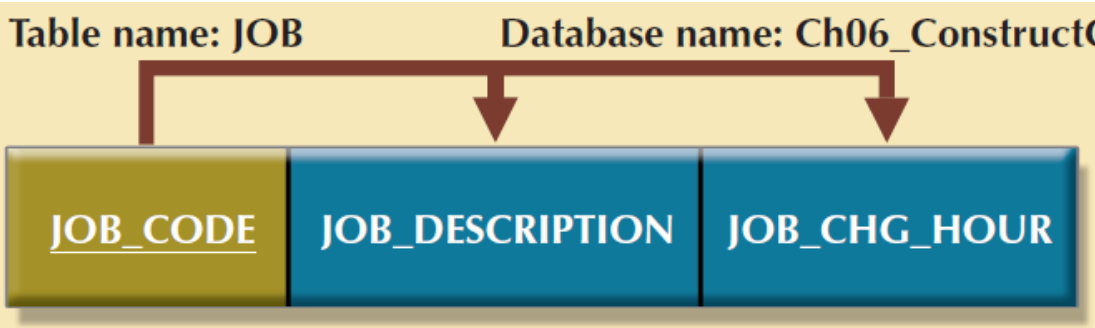


Table name: JOB

JOB_CODE	JOB_DESCRIPTION	JOB_CHG_HOUR
500	Programmer	35.75
501	Systems Analyst	96.75
502	Database Designer	105.00
503	Electrical Engineer	84.50
504	Mechanical Engineer	67.90
505	Civil Engineer	55.78
506	Clerical Support	26.87
507	DSS Analyst	45.95
508	Applications Designer	48.10
509	Bio Technician	34.55
510	General Support	18.36

Cengage Learning © 2015

- Entering DB designer instead of Database Designer for the JOB_CLASS attribute in the EMPLOYEE table can lead to referential integrity violations.

Figure 6.6 - The Completed Database

Table name: EMPLOYEE **Database name: Ch06_ConstructCo**

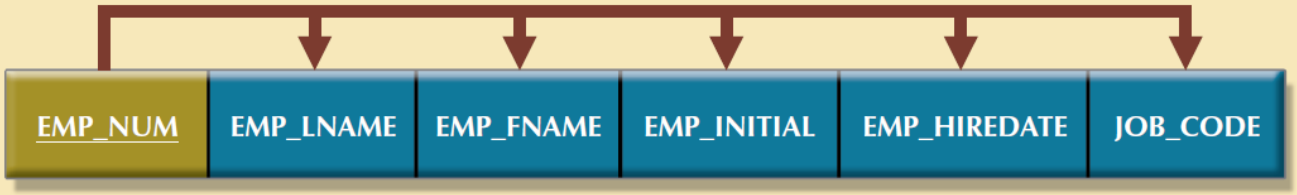
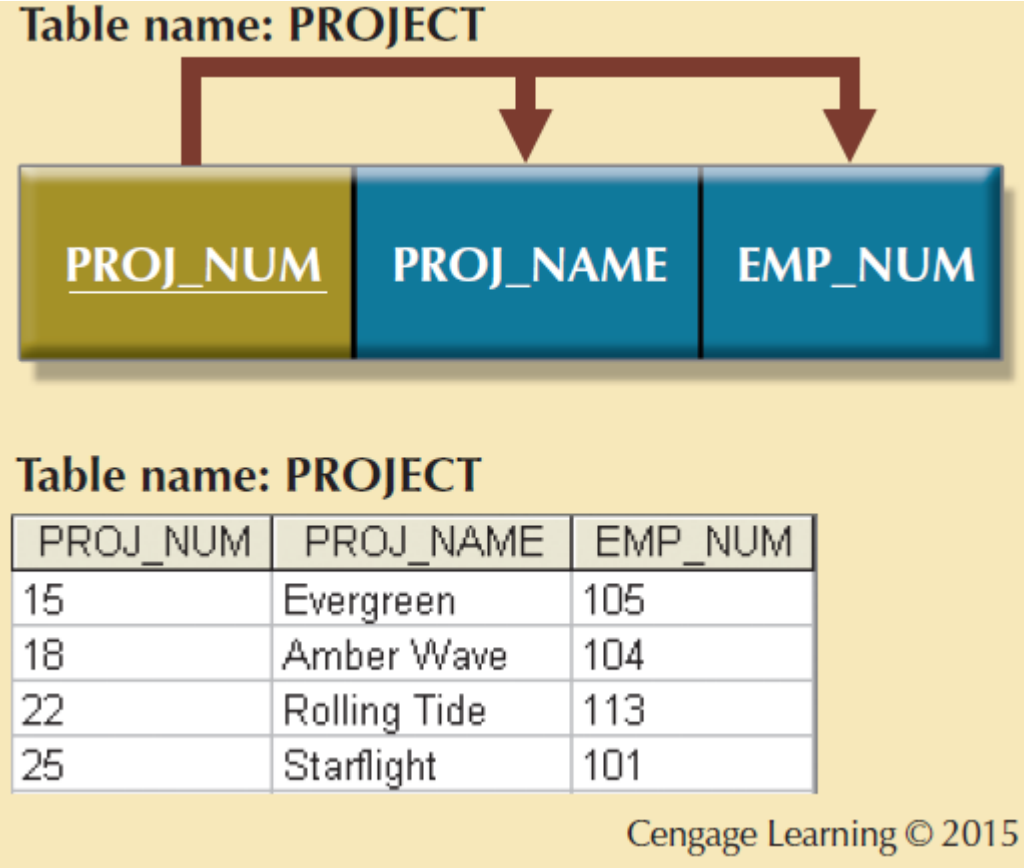


Table name: EMPLOYEE

EMP_NUM	EMP_LNAME	EMP_FNAME	EMP_INITIAL	EMP_HIREDATE	JOB_CODE
101	News	John	G	08-Nov-00	502
102	Senior	David	H	12-Jul-89	501
103	Arbough	June	E	01-Dec-97	503
104	Ramoras	Anne	K	15-Nov-88	501
105	Johnson	Alice	K	01-Feb-94	502
106	Smithfield	William		22-Jun-05	500
107	Alonzo	Maria	D	10-Oct-94	500
108	Washington	Ralph	B	22-Aug-89	501
109	Smith	Larry	W	18-Jul-99	501
110	Olenko	Gerald	A	11-Dec-96	505
111	Wabash	Geoff	B	04-Apr-89	506
112	Smithson	Darlene	M	23-Oct-95	507
113	Joenbrood	Delbert	K	15-Nov-94	508
114	Jones	Annelise		20-Aug-91	508
115	Bawangi	Travis	B	25-Jan-90	501
116	Pratt	Gerald	L	05-Mar-95	510
117	Williamson	Angie	H	19-Jun-94	509
118	Frommer	James	J	04-Jan-06	510

Cengage Learning © 2015

Figure 6.6 - The Completed Database



- Each project has only one manager.
- Use the EMP_NUM as a foreign key in PROJECT.

Figure 6.6 - The Completed Database

Table name: ASSIGNMENT

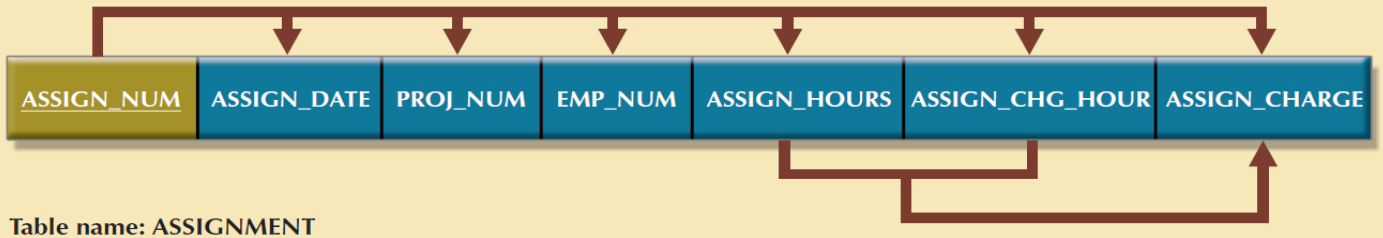


Table name: ASSIGNMENT

ASSIGN_NUM	ASSIGN_DATE	PROJ_NUM	EMP_NUM	ASSIGN_HOURS	ASSIGN_CHG_HOUR	ASSIGN_CHARGE
1001	04-Mar-14	15	103	2.6	84.50	219.70
1002	04-Mar-14	18	118	1.4	18.36	25.70
1003	05-Mar-14	15	101	3.6	105.00	378.00
1004	05-Mar-14	22	113	2.5	48.10	120.25
1005	05-Mar-14	15	103	1.9	84.50	160.55
1006	05-Mar-14	25	115	4.2	96.75	406.35
1007	05-Mar-14	22	105	5.2	105.00	546.00
1008	05-Mar-14	25	101	1.7	105.00	178.50
1009	05-Mar-14	15	105	2.0	105.00	210.00
1010	06-Mar-14	15	102	3.8	96.75	367.65
1011	06-Mar-14	22	104	2.6	96.75	251.55
1012	06-Mar-14	15	101	2.3	105.00	241.50
1013	06-Mar-14	25	114	1.8	48.10	86.58
1014	06-Mar-14	22	111	4.0	26.87	107.48
1015	06-Mar-14	25	114	3.4	48.10	163.54
1016	06-Mar-14	18	112	1.2	45.95	55.14
1017	06-Mar-14	18	118	2.0	18.36	36.72
1018	06-Mar-14	18	104	2.6	96.75	251.55
1019	06-Mar-14	15	103	3.0	84.50	253.50
1020	07-Mar-14	22	105	2.7	105.00	283.50
1021	08-Mar-14	25	108	4.2	96.75	406.35
1022	07-Mar-14	25	114	5.8	48.10	278.98
1023	07-Mar-14	22	106	2.4	35.75	85.80

- Using a surrogate primary key such as ASSIGN_NUM provides lower granularity and yields greater flexibility.

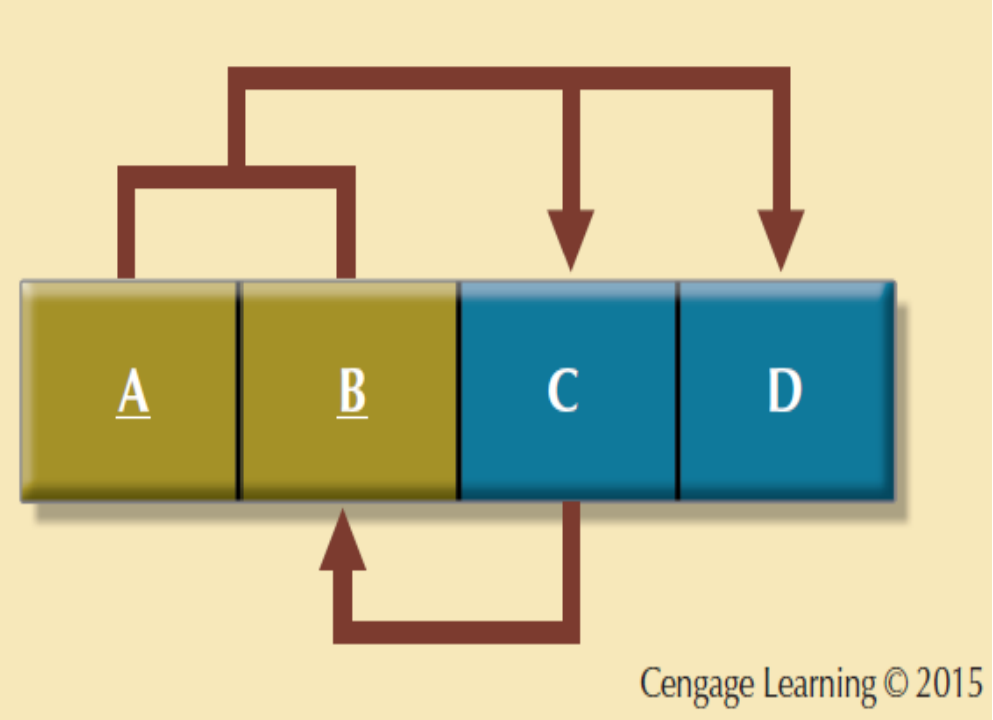
Surrogate Keys

- Used by designers when the primary key is considered to be unsuitable
- System-defined attribute
- Created and managed via the DBMS
- Have a numeric value which is automatically incremented for each new row
  - However, can create the case where duplicate records are created, without violation of either entity integrity or referential integrity

The Boyce-Codd Normal Form (BCNF)

- Every determinant in the table should be a candidate key
  - Candidate key - Same characteristics as primary key but not chosen to be the primary key
- Equivalent to 3NF when the table contains only one candidate key
- Violated only when the table contains more than one candidate key
- Considered to be a special case of 3NF

Figure 6.7 - A Table That is in 3NF and not in BCNF



- A+B→ C, D
- A+C→ B, D
- C→ B

Figure 6.8 - Decomposition to BCNF

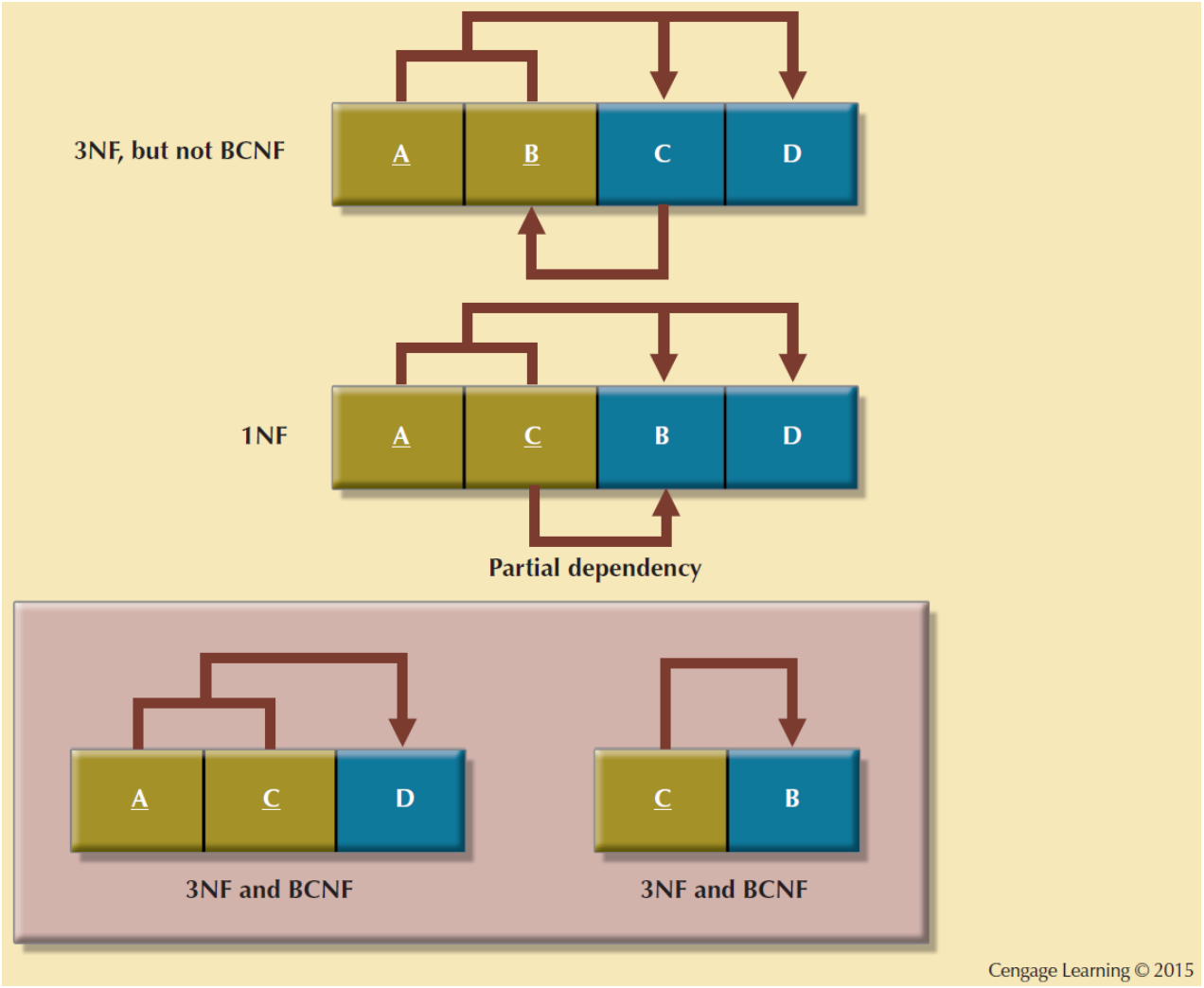


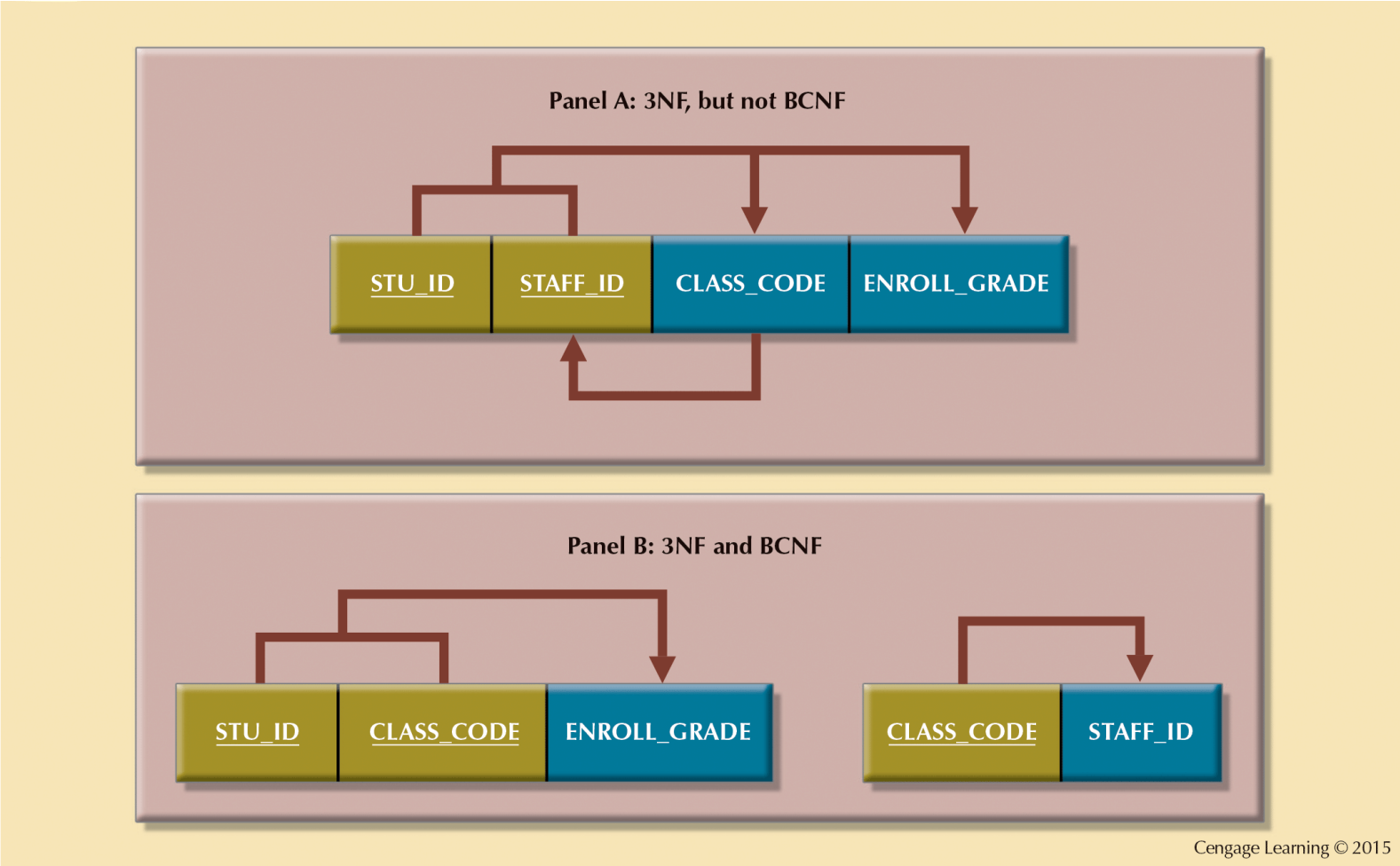
Table 6.5 - Sample Data for a BCNF Conversion

STU_ID	STAFF_ID	CLASS_CODE	ENROLL_GRADE
125	25	21334	A
125	20	32456	C
135	20	28458	B
144	25	27563	C
144	20	32456	B

Cengage Learning © 2015

- If a different staff member is assigned to teach class 32456, two rows will require updates→ update anomaly.
- If student 135 drops class 28458, information about who taught that class is lost→ deletion anomaly.

Figure 6.9 - Decomposition to BCNF



Fourth Normal Form (4NF)

- Table is in 4NF when it:
  - Is in 3NF
  - Has no multivalued dependencies
- Rules
  - All attributes must be dependent on the primary key, but they must be independent of each other
  - No row may contain two or more multivalued facts about an entity

Figure 6.10 - Tables with Multivalued Dependencies

Database name: Ch06_Service

Table name: VOLUNTEER_V1

EMP_NUM	ORG_CODE	ASSIGN_NUM
10123	RC	1
10123	UW	3
10123		4

Table name: VOLUNTEER_V2

EMP_NUM	ORG_CODE	ASSIGN_NUM
10123	RC	
10123	UW	
10123		1
10123		3
10123		4

Table name: VOLUNTEER_V3

EMP_NUM	ORG_CODE	ASSIGN_NUM
10123	RC	1
10123	RC	3
10123	UW	4

Cengage Learning © 2015

- V3: One employee can have many service entries and many assignment entries. Therefore, one EMP_NUM can determine multiple values of ORG_CODE and multiple values of ASSIGN_NUM;

Figure 6.11 - A Set of Tables in 4NF

Database name: CH06_Service

Table name: PROJECT

PROJ_CODE	PROJ_NAME	PROJ_BUDGET
1	BeThere	1023245.00
2	BlueMoon	20198608.00
3	GreenThumb	3234456.00
4	GoFast	5674000.00
5	GoSlow	1002500.00

Table name: ASSIGNMENT

ASSIGN_NUM	EMP_NUM	PROJ_CODE
1	10123	1
2	10121	2
3	10123	3
4	10123	4
5	10121	1
6	10124	2
7	10124	3
8	10124	5

Table name: EMPLOYEE

EMP_NUM	EMP_LNAME
10121	Rogers
10122	O'Leery
10123	Panera
10124	Johnson

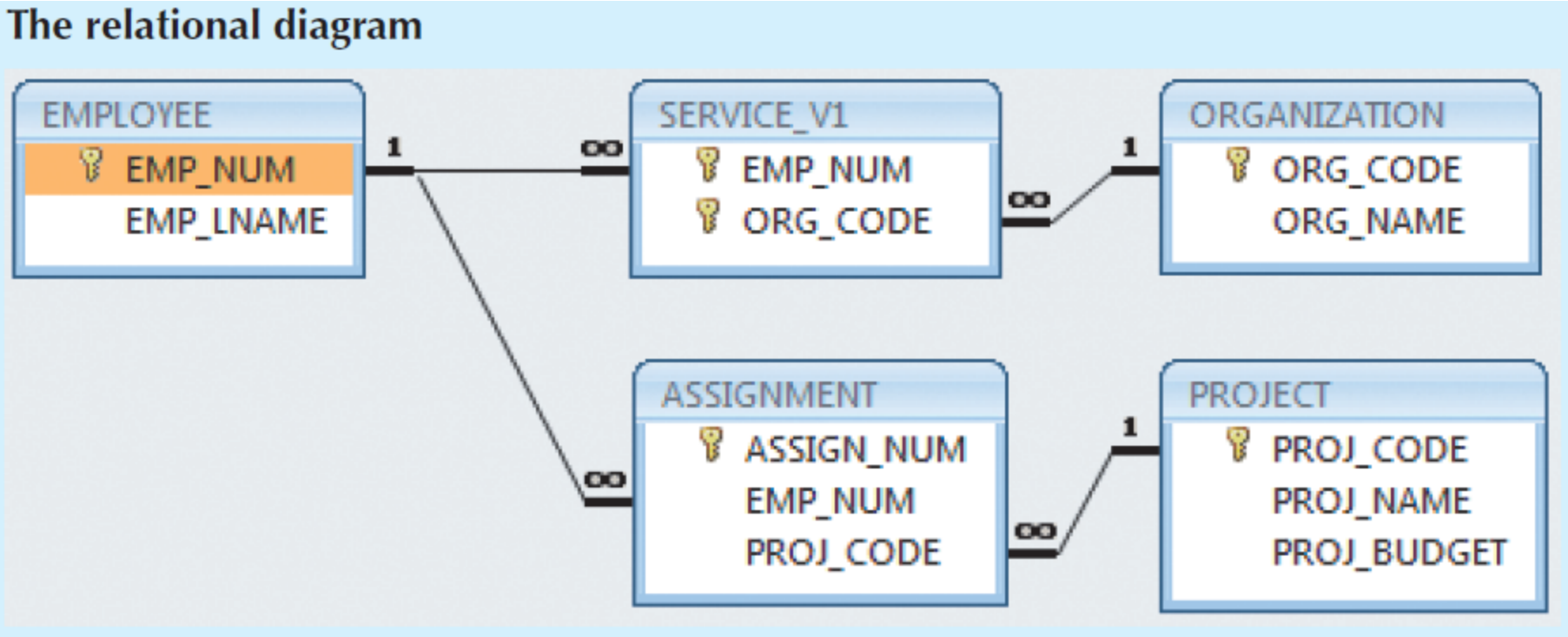
Table name: ORGANIZATION

ORG_CODE	ORG_NAME
RC	Red Cross
UW	United Way
WF	Wildlife Fund

Table name: SERVICE_V1

EMP_NUM	ORG_CODE
10123	RC
10123	UW
10123	WF

Figure 6.11 - A Set of Tables in 4NF



Cengage Learning © 2015

Normalization and Database Design

- Normalization should be part of the design process
- Proposed entities must meet the required normal form before table structures are created
- Principles and normalization procedures to be understood to redesign and modify databases
  - ERD is created through an iterative process
  - Normalization focuses on the characteristics of specific entities

Normalization and Database Design (continued)

- ER diagram
  - Provides big picture, or macro view, of an organization's data requirements and operations
  - Created through an iterative process
    - Identifying relevant entities, their attributes and their relationship
    - Use results to identify additional entities and attributes

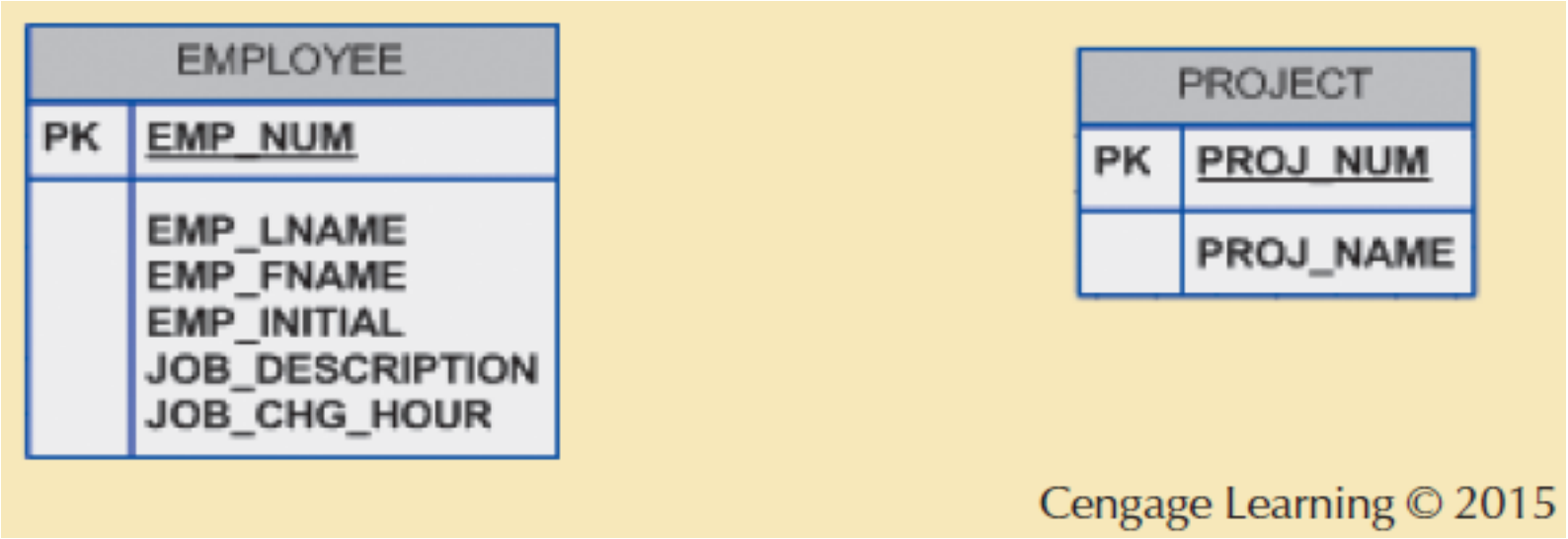
Normalization and Database Design (continued)

- Normalization procedures
  - Focus on characteristics of specific entities
  - Represents micro view of entities within ER diagram
- Difficult to separate normalization process from ER modeling process.
- Two techniques should be used concurrently

Normalization in the Design Process

- The company manages many projects
- Each project requires the services of many employees.
- An employee may be assigned to several different projects. Some employees are not assigned to a project and perform duties not specifically related to a project. Some employees are part of a labor pool, to be shared by all project teams.
- Each employee has a single primary job classification, which determines the hourly billing rate.
- Many employees can have the same job classification.

Figure 6.12 - Initial Contracting Company ERD



- JOB_DESCRIPTION defines job classifications such as Systems Analyst, Database Designer and Programmer.
- Those classifications determine the billing rates, JOB_CHG_HOUR.
- EMPLOYEE contains a transitive dependency.

Figure 6.13 - Modified Contracting Company ERD

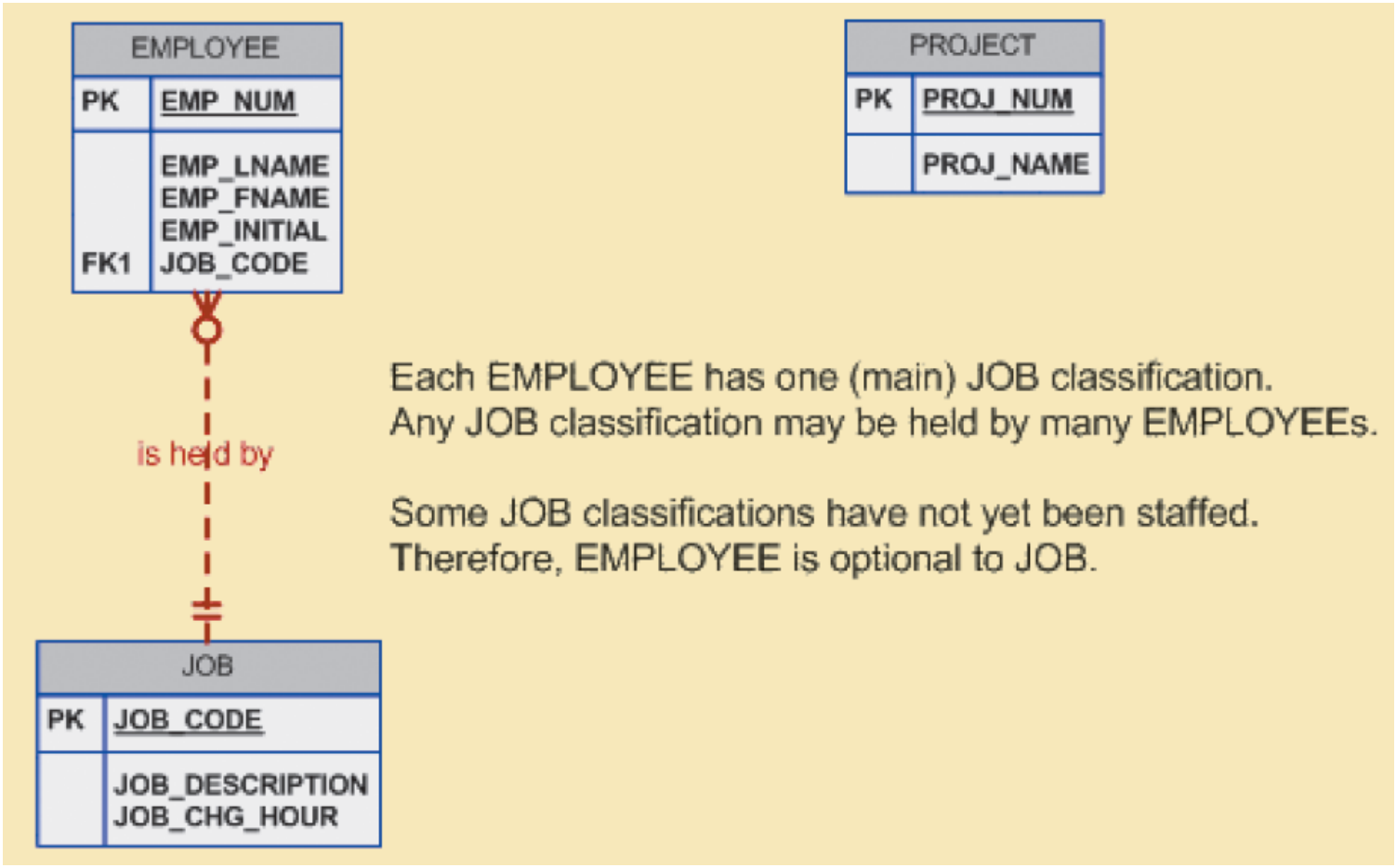
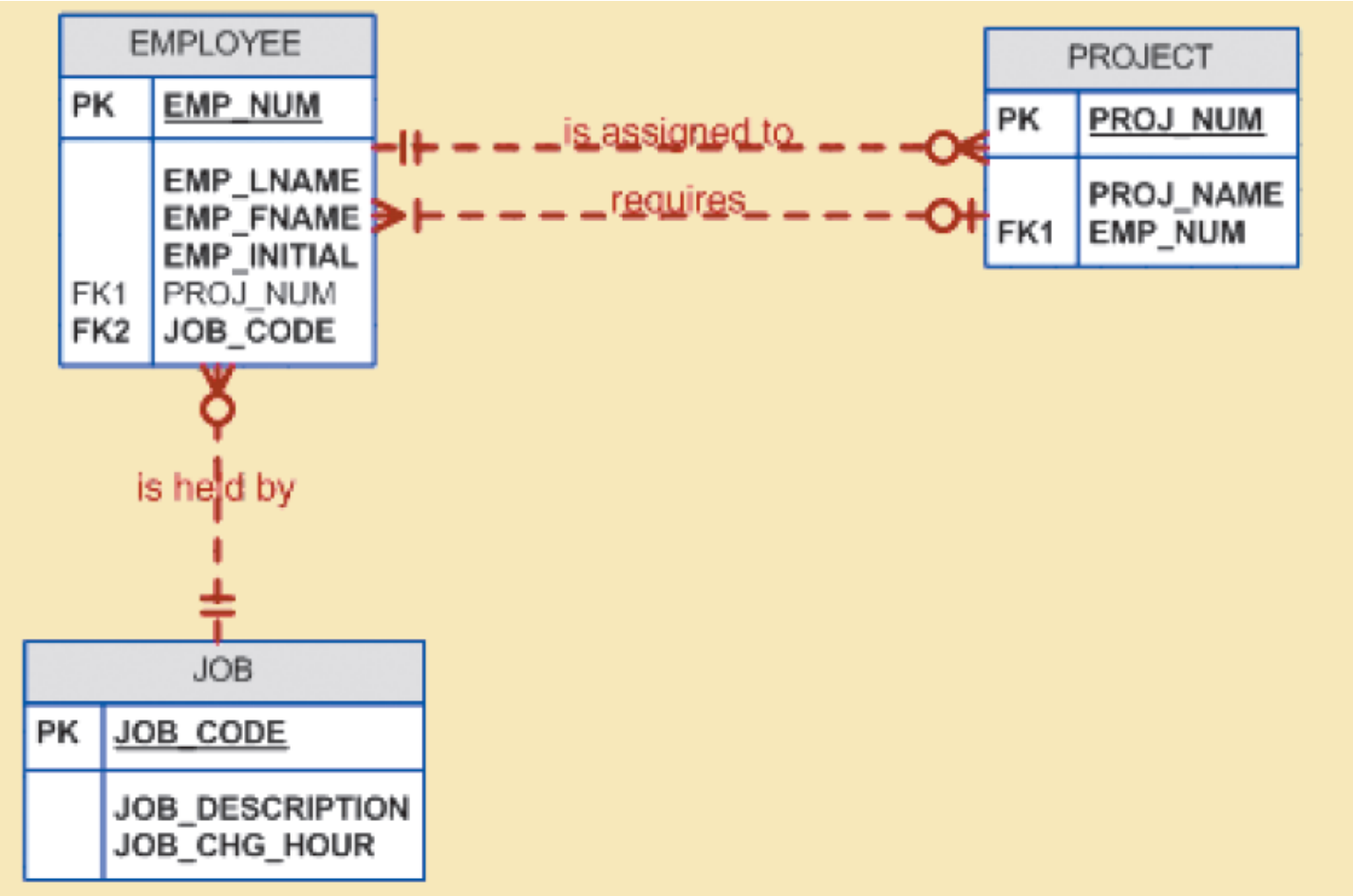
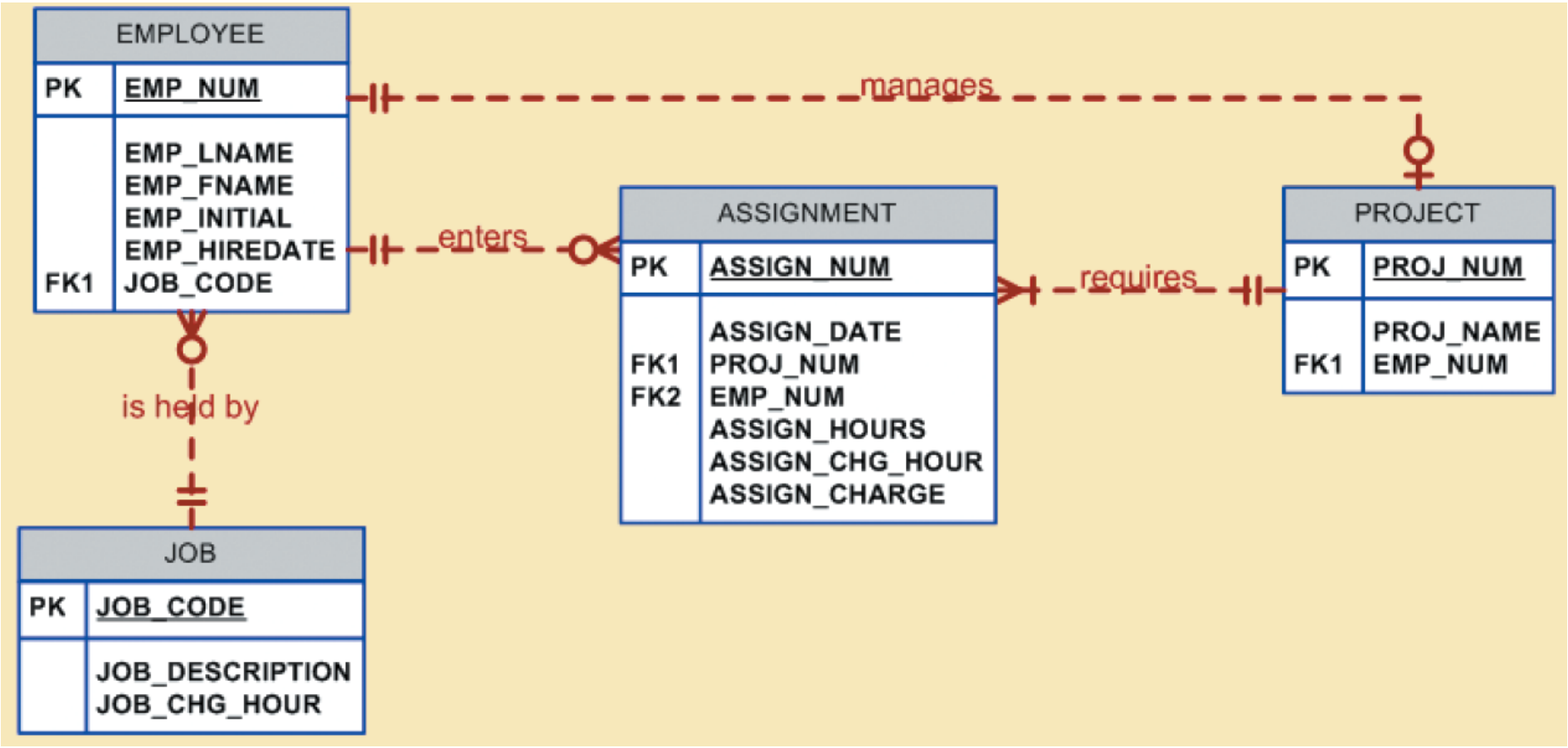


Figure 6.14 - Incorrect M:N Relationship Representation



Cengage Learning © 2015

Figure 6.15 - Final Contracting Company ERD



Cengage Learning © 2015

Figure 6.16 - The Implemented Database

Table name: EMPLOYEE

Database name: Ch06_ConstructCo

EMP_NUM	EMP_LNAME	EMP_FNAME	EMP_INITIAL	EMP_HIREDATE	JOB_CODE
101	News	John	G	08-Nov-00	502
102	Senior	David	H	12-Jul-89	501
103	Arbough	June	E	01-Dec-97	503
104	Ramoras	Anne	K	15-Nov-88	501
105	Johnson	Alice	K	01-Feb-94	502
106	Smithfield	William		22-Jun-05	500
107	Alonzo	Maria	D	10-Oct-94	500
108	Washington	Ralph	B	22-Aug-89	501
109	Smith	Larry	W	18-Jul-99	501
110	Olenko	Gerald	A	11-Dec-96	505
111	Wabash	Geoff	B	04-Apr-89	506
112	Smithson	Darlene	M	23-Oct-95	507
113	Joenbrood	Delbert	K	15-Nov-94	508
114	Jones	Annelise		20-Aug-91	508
115	Bawangi	Travis	B	25-Jan-90	501
116	Pratt	Gerald	L	05-Mar-95	510
117	Williamson	Angie	H	19-Jun-94	509
118	Frommer	James	J	04-Jan-06	510

Table name: JOB

JOB_CODE	JOB_DESCRIPTION	JOB_CHG_HOUR
500	Programmer	35.75
501	Systems Analyst	96.75
502	Database Designer	105.00
503	Electrical Engineer	84.50
504	Mechanical Engineer	67.90
505	Civil Engineer	55.78
506	Clerical Support	26.87
507	DSS Analyst	45.95
508	Applications Designer	48.10
509	Bio Technician	34.55
510	General Support	18.36

Table name: PROJECT

PROJ_NUM	PROJ_NAME	EMP_NUM
15	Evergreen	105
18	Amber Wave	104
22	Rolling Tide	113
25	Starflight	101

Table name: ASSIGNMENT

ASSIGN_NUM	ASSIGN_DATE	PROJ_NUM	EMP_NUM	ASSIGN_HOURS	ASSIGN_CHG_HOUR	ASSIGN_CHARGE
1001	04-Mar-14	15	103	2.6	84.50	219.70
1002	04-Mar-14	18	118	1.4	18.36	25.70
1003	05-Mar-14	15	101	3.6	105.00	378.00
1004	05-Mar-14	22	113	2.5	48.10	120.25
1005	05-Mar-14	15	103	1.9	84.50	160.55
1006	05-Mar-14	25	115	4.2	96.75	406.35
1007	05-Mar-14	22	105	5.2	105.00	546.00
1008	05-Mar-14	25	101	1.7	105.00	178.50
1009	05-Mar-14	15	105	2.0	105.00	210.00
1010	06-Mar-14	15	102	3.8	96.75	367.65
1011	06-Mar-14	22	104	2.6	96.75	251.55
1012	06-Mar-14	15	101	2.3	105.00	241.50
1013	06-Mar-14	25	114	1.8	48.10	86.58
1014	06-Mar-14	22	111	4.0	26.87	107.48
1015	06-Mar-14	25	114	3.4	48.10	163.54
1016	06-Mar-14	18	112	1.2	45.95	55.14
1017	06-Mar-14	18	118	2.0	18.36	36.72
1018	06-Mar-14	18	104	2.6	96.75	251.55
1019	06-Mar-14	15	103	3.0	84.50	253.50
1020	07-Mar-14	22	105	2.7	105.00	283.50
1021	08-Mar-14	25	108	4.2	96.75	406.35
1022	07-Mar-14	25	114	5.8	48.10	278.98
1023	07-Mar-14	22	106	2.4	35.75	85.80

Cengage Learning © 2015

Denormalization

- Design goals
  - Creation of normalized relations
  - Processing requirements and speed
- Number of database tables expands when tables are decomposed to conform to normalization requirements
- Joining a larger number of tables:
  - Takes additional input/output (I/O) operations and processing logic
  - Reduces system speed

Denormalization (continued)

- Use denormalization cautiously
- Understand why—under some circumstances—unnormalized tables are better choice

Denormalization (continued)

CASE	EXAMPLE	RATIONALE AND CONTROLS
Redundant data	Storing ZIP and CITY attributes in the AGENT table when ZIP determines CITY (see Figure 2.2)	• Avoid extra join operations • Program can validate city (drop-down box) based on the zip code
Derived data	Storing STU_HRS and STU_CLASS (student classification) when STU_HRS determines STU_CLASS (see Figure 3.28)	• Avoid extra join operations • Program can validate classification (lookup) based on the student hours
Preaggregated data (also derived data)	Storing the student grade point average (STU_GPA) aggregate value in the STUDENT table when this can be calculated from the ENROLL and COURSE tables (see Figure 3.28)	• Avoid extra join operations • Program computes the GPA every time a grade is entered or updated • STU_GPA can be updated only via administrative routine
Information requirements	Using a temporary denormalized table to hold report data; this is required when creating a tabular report in which the columns represent data that are stored in the table as rows (see Figures 6.17 and 6.18)	• Impossible to generate the data required by the report using plain SQL • No need to maintain table • Temporary table is deleted once report is done • Processing speed is not an issue

Cengage Learning © 2015

Table 6.7 - Data-Modeling Checklist

BUSINESS RULES
• Properly document and verify all business rules with the end users. • Ensure that all business rules are written precisely, clearly, and simply. The business rules must help identify entities, attributes, relationships, and constraints. • Identify the source of all business rules, and ensure that each business rule is justified, dated, and signed off by an approving authority.

Cengage Learning © 2015

Table 6.7 - Data-Modeling Checklist

DATA MODELING

Naming conventions: All names should be limited in length (database-dependent size).

- Entity names:
  - Should be nouns that are familiar to business and should be short and meaningful
  - Should document abbreviations, synonyms, and aliases for each entity
  - Should be unique within the model
  - For composite entities, may include a combination of abbreviated names of the entities linked through the composite entity
- Attribute names:
  - Should be unique within the entity
  - Should use the entity abbreviation as a prefix
  - Should be descriptive of the characteristic
  - Should use suffixes such as _ID, _NUM, or _CODE for the PK attribute
  - Should not be a reserved word
  - Should not contain spaces or special characters such as @, !, or &
- Relationship names:
  - Should be active or passive verbs that clearly indicate the nature of the relationship

Cengage Learning © 2015

Table 6.7 - Data-Modeling Checklist

DATA MODELING

Entities:

- Each entity should represent a single subject.
- Each entity should represent a set of distinguishable entity instances.
- All entities should be in 3NF or higher. Any entities below 3NF should be justified.
- The granularity of the entity instance should be clearly defined.
- The PK should be clearly defined and support the selected data granularity.

Attributes:

- Should be simple and single-valued (atomic data)
- Should document default values, constraints, synonyms, and aliases
- Derived attributes should be clearly identified and include source(s)
- Should not be redundant unless this is required for transaction accuracy, performance, or maintaining a history
- Nonkey attributes must be fully dependent on the PK attribute

Table 6.7 - Data-Modeling Checklist

DATA MODELING
Relationships: • Should clearly identify relationship participants • Should clearly define participation, connectivity, and document cardinality **ER model:** • Should be validated against expected processes: inserts, updates, and deletions • Should evaluate where, when, and how to maintain a history • Should not contain redundant relationships except as required (see attributes) • Should minimize data redundancy to ensure single-place updates • Should conform to the minimal data rule: All that is needed is there, and all that is there is needed.

Cengage Learning © 2015

Summary

- Normalization is technique used to design tables in which data redundancies are minimized
- First three normal forms (1NF, 2NF, and 3NF) are most commonly encountered
- Table is in 1NF when all key attributes are defined and when all remaining attributes are dependent on primary key

Summary (continued)

- Table is in 2NF when it is in 1NF and contains no partial dependencies
- Table is in 3NF when it is in 2NF and contains no transitive dependencies
- Table that is not in 3NF may be split into new tables until all of the tables meet 3NF requirements
- Normalization is important part—but only part—of design process