Database Systems: Design, Implementation, and Management

Lesson 6

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
  - 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

Database Tables and Normalization

- **Normalization**
  - Process for evaluating and correcting table structures to minimize data redundancies
    - Reduces data anomalies
  - Series of stages called normal forms:
    - First normal form (1NF)
    - Second normal form (2NF)
    - Third normal form (3NF)

Database Tables and Normalization (cont'd.)

- Normalization (continued)
  - 2NF is better than 1NF; 3NF is better than 2NF
  - For most business database design purposes, 3NF is as high as needed in normalization
  - Highest level of normalization is not always most desirable
- **Denormalization** produces a lower normal form
  - Increased performance but greater data redundancy

The Need for Normalization

- Example: company that manages building projects
  - Charges its clients by billing hours spent on each contract
  - Hourly billing rate is dependent on employee’s position
  - Periodically, report is generated that contains information such as displayed in Table 6.1

FIGURE 6.1 Tabular representation of the report format

Table name: RPT_FORMAT **Database name: Ch06_ConstructCo**

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

TABLE 6.1 A Sample Report Layout

PROJECT NUMBER	PROJECT NAME	EMPLOYEE NUMBER	EMPLOYEE NAME	JOB CLASS	CHARGE/ HOUR	HOURS BILLED	TOTAL CHARGE
15	Evergreen	103	June E. Arbough	Elec. Engineer	$ 85.50	23.8	$ 2,034.90
		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			**$10,573.50**
18	Amber Wave	114	Annelise Jones	Applications Designer	$ 48.10	25.6	$ 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	45.0	$ 2,067.75
				Subtotal			**$ 7,265.52**
22	Rolling Tide	105	Alice K. Johnson	Database Designer	$105.00	65.7	$ 6,998.50
		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			**$13,765.10**
25	Starflight	107	Maria D. Alonzo	Programmer	$ 35.75	25.6	$ 915.20
		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			**$17,595.57**
				Total			**$49,199.69**

Note: * indicates project leader

The Need for Normalization (cont’d.)

- Structure of data set in Figure 6.1 does not handle data very well
- Table structure appears to work; report is generated with ease
- Report may yield different results depending on what data anomaly has occurred
- Relational database environment is suited to help designer avoid data integrity problems

The Normalization Process

- Each table represents a single subject
- No data item will be unnecessarily stored in more than one table
- All nonprime attributes in a table are dependent on the primary key
- Each table is void of insertion, update, deletion anomalies

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

The Normalization Process (cont'd.)

- Objective of normalization is to ensure that all tables are in at least 3NF
- Higher forms are not likely to be encountered in business environment
- Normalization works one relation at a time
- Progressively breaks table into new set of relations based on identified dependencies

TABLE 6.3

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*. Example: PROJ_NUM → PROJ_NAME (read as "PROJ_NUM functionally determines PROJ_NAME") In this case, the attribute PROJ_NUM is known as the "determinant" attribute, and the attribute PROJ_NAME is known as the "dependent" attribute.
Functional dependence (generalized definition)	Attribute *A* determines attribute *B* (that is, *B* is functionally dependent on *A*) 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*.

Conversion to First Normal Form

- **Repeating group**
  - Group of multiple entries of same type can exist for any single key attribute occurrence
- Relational table must not contain repeating groups
- Normalizing table structure will reduce data redundancies
- Normalization is three-step procedure

Conversion to First Normal Form (cont'd.)

- Step 1: Eliminate the Repeating Groups
  - Eliminate nulls: each repeating group attribute contains an appropriate data value
- Step 2: Identify the Primary Key
  - Must uniquely identify attribute value
  - New key must be composed
- Step 3: Identify All Dependencies
  - Dependencies are depicted with a diagram

FIGURE 6.2 A table in first normal form

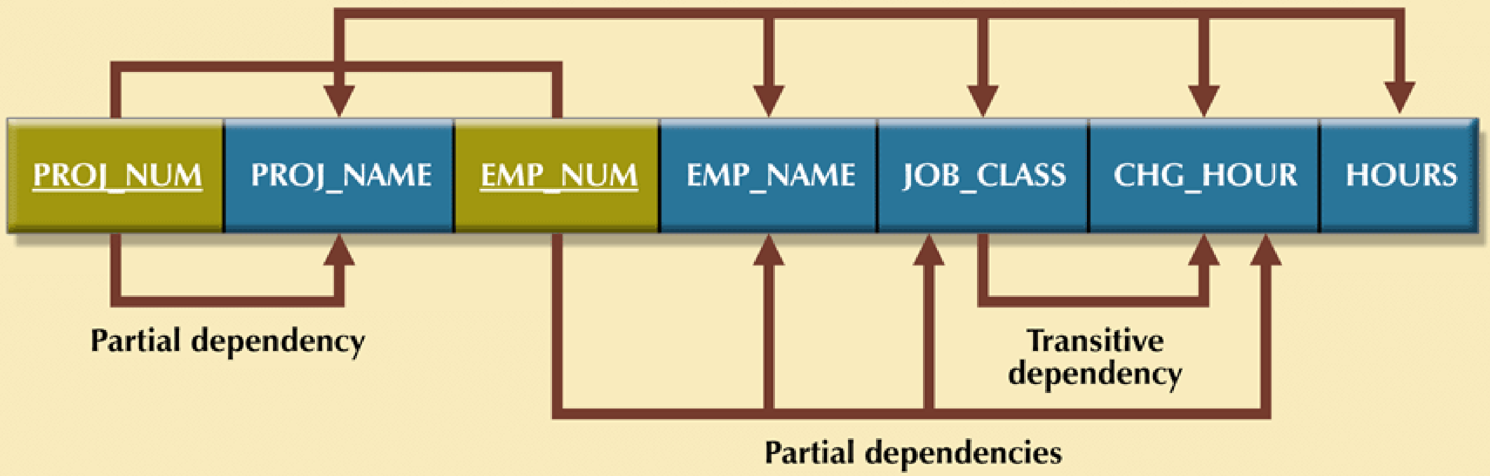
Table name: DATA_ORG_1NF **Database name: Ch06_ConstructCo**

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 (cont'd.)

- **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 you 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)

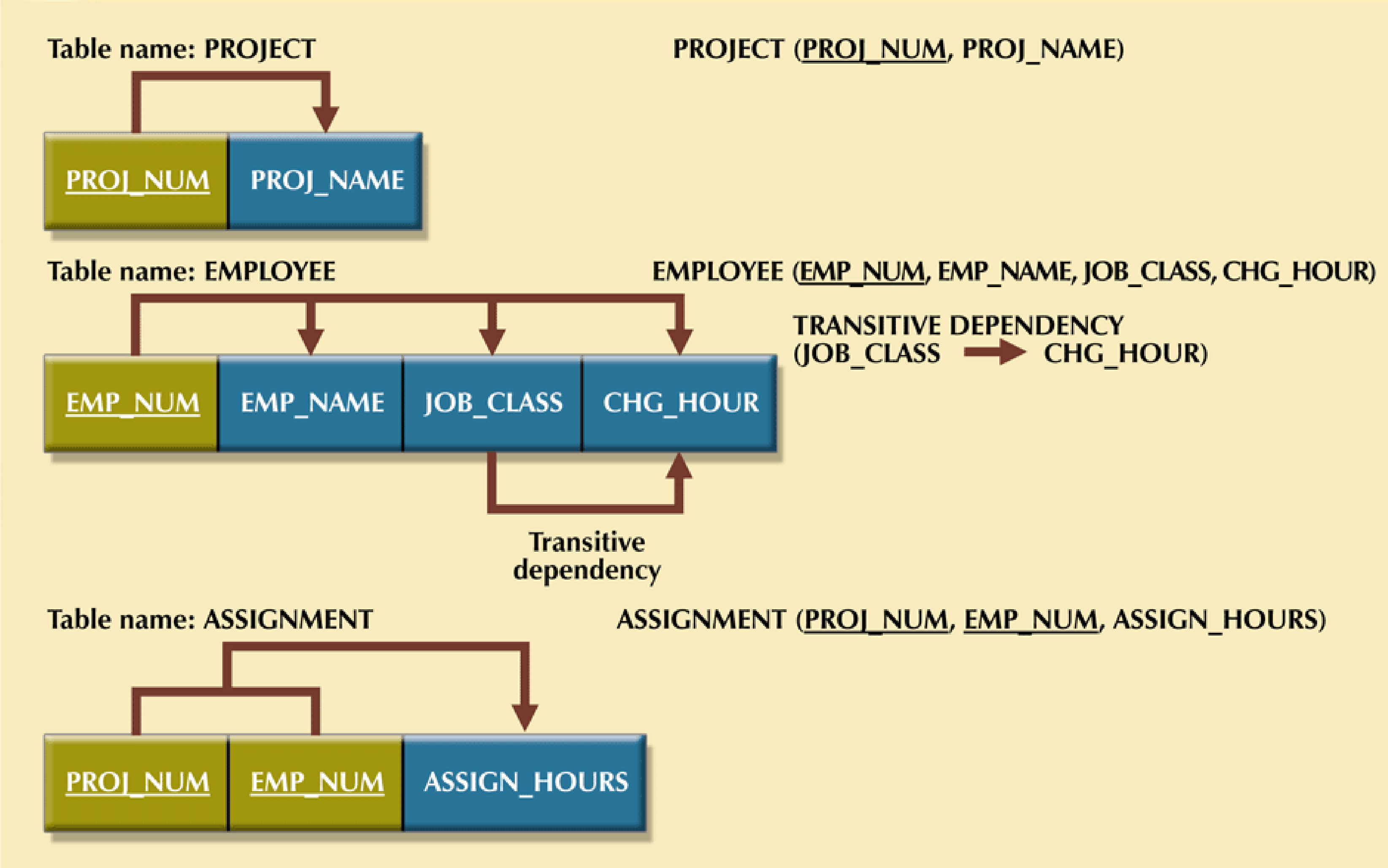
Conversion to First Normal Form (cont'd.)

- First normal form describes tabular format:
  - All key attributes are defined
  - No repeating groups in the table
  - All attributes are dependent on primary key
- All relational tables satisfy 1NF requirements
- Some tables contain partial dependencies
  - Dependencies are based on part of the primary key
  - Should be used with caution

Conversion to Second Normal Form

- Step 1: Make New Tables to Eliminate Partial Dependencies
  - Write each key component on separate line, then write original (composite) key on last line
  - Each component will become key in new table
- Step 2: Assign Corresponding Dependent Attributes
  - Determine attributes that are dependent on other attributes
  - At this point, most anomalies have been eliminated

FIGURE 6.4 Second normal form (2NF) conversion results



Conversion to Second Normal Form (cont'd.)

- Table is in second normal form (2NF) when:
  - It is in 1NF *and*
  - It includes no partial dependencies:
    - No attribute is dependent on only portion of primary key

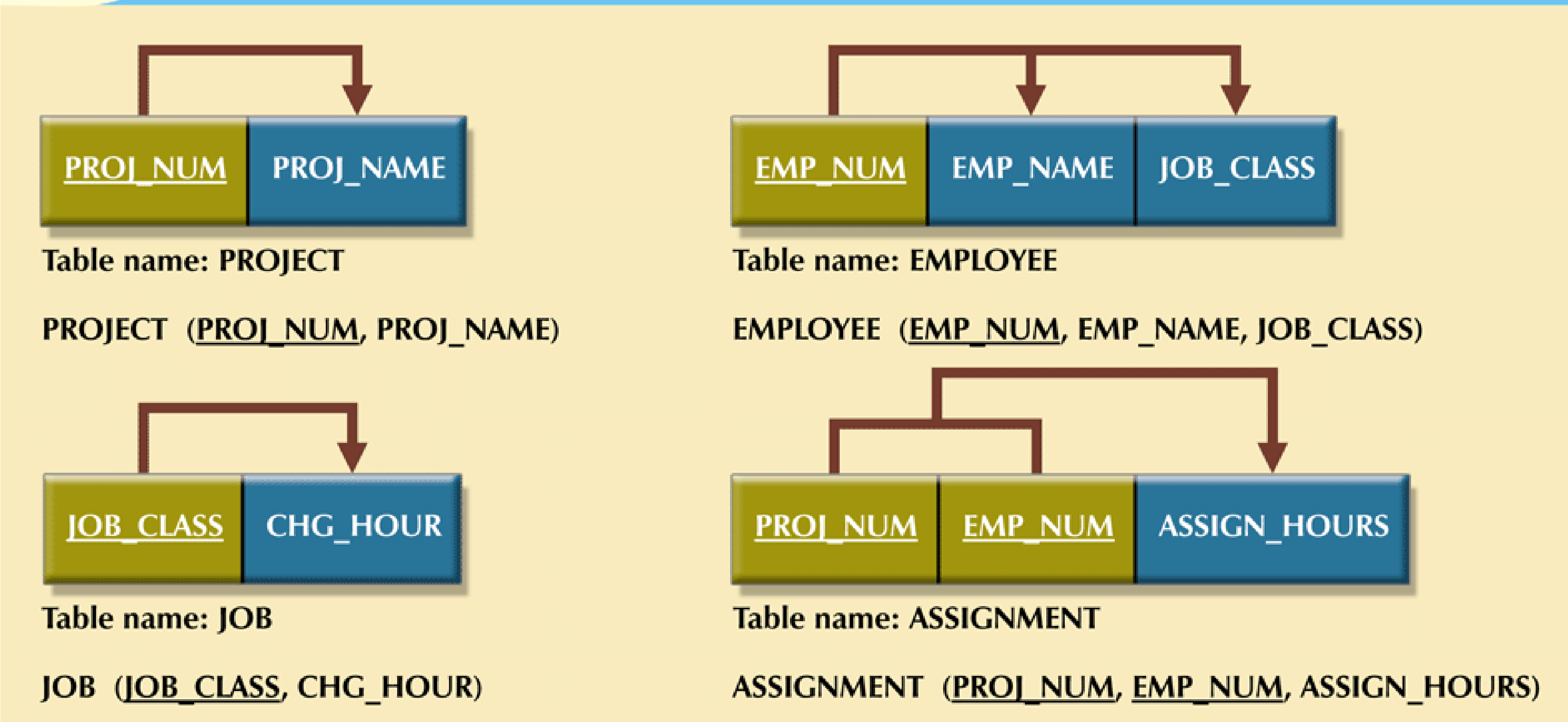
Conversion to Third Normal Form

- Step 1: Make New Tables to Eliminate Transitive Dependencies
  - 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 (cont'd.)

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

FIGURE 6.5 Third normal form (3NF) conversion results



Conversion to Third Normal Form (cont'd.)

- 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 should be cleaned up to eliminate initial partial and transitive dependencies
- Normalization cannot, by itself, be relied on to make good designs
- Valuable because it helps eliminate data redundancies

Improving the Design (cont'd.)

- Issues to address, in order, to produce a good normalized set of tables:
  - Evaluate PK Assignments
  - Evaluate Naming Conventions
  - Refine Attribute Atomicity
  - Identify New Attributes

Improving the Design (cont'd.)

- Issues to address, in order, to produce a good normalized set of tables (cont'd.):
  - Identify New Relationships
  - Refine Primary Keys as Required for Data Granularity
  - Maintain Historical Accuracy
  - Evaluate Using Derived Attributes

FIGURE 6.6 The completed database (continued)

Table name: EMPLOYEE

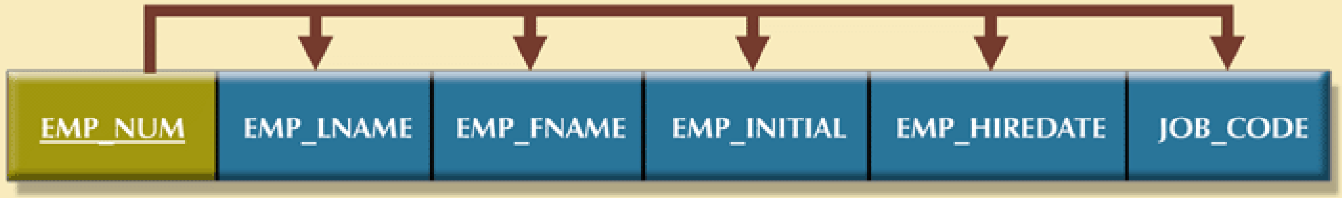


Table name: EMPLOYEE

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

Surrogate Key Considerations

- When primary key is considered to be unsuitable, designers use surrogate keys
- Data entries in Table 6.4 are inappropriate because they duplicate existing records
  - No violation of entity or referential integrity

TABLE 6.4 Duplicate Entries in the Job Table

JOB_CODE	JOB_DESCRIPTION	JOB_CHG_HOUR
511	Programmer	$35.75
512	Programmer	$35.75

Normalization and Database Design

- Normalization should be part of the design process
- Make sure that proposed entities meet required normal form before table structures are created
- Many real-world databases have been improperly designed or burdened with anomalies
- You may be asked to redesign and modify existing databases

Normalization and Database Design (cont'd.)

- ER diagram
  - Identify relevant entities, their attributes, and their relationships
  - Identify additional entities and attributes
- Normalization procedures
  - Focus on characteristics of specific entities
  - Micro view of entities within ER diagram
- Difficult to separate normalization process from ER modeling process

FIGURE 6.12 Initial contracting company ERD

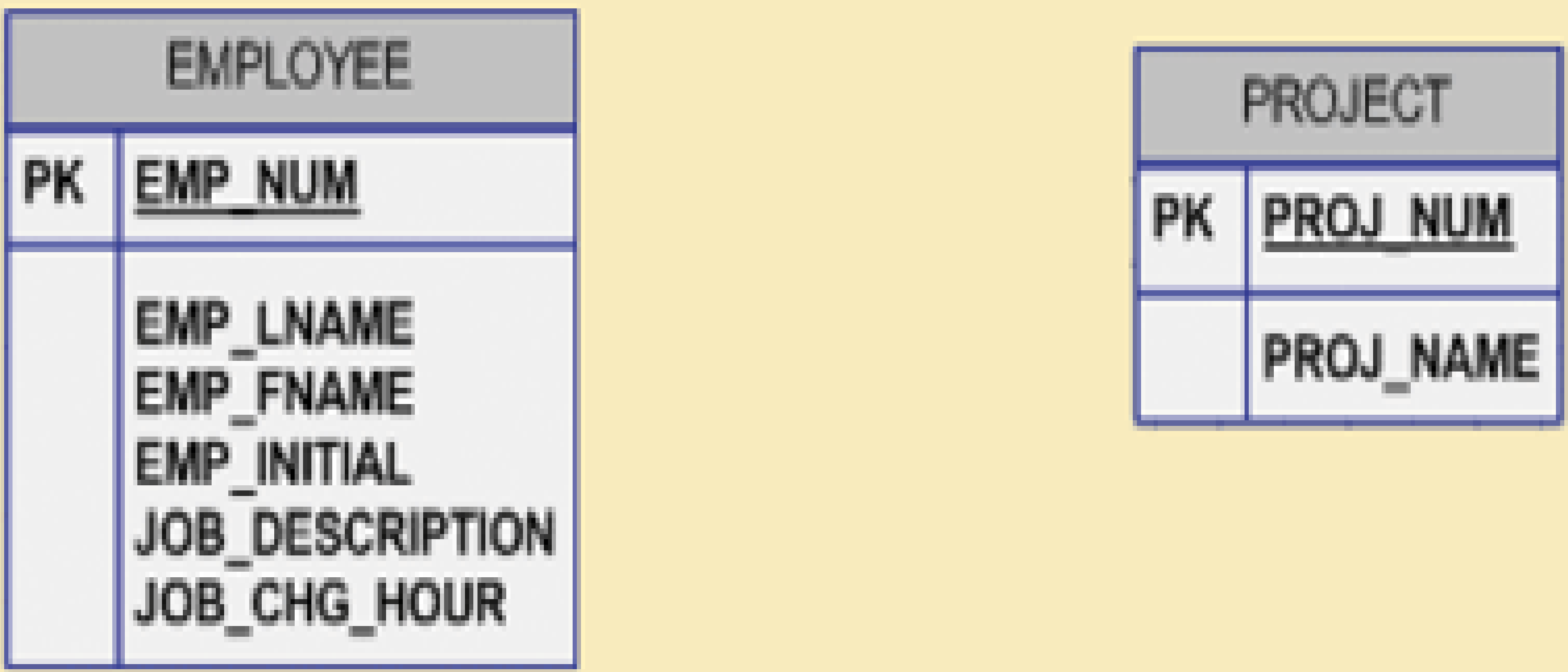
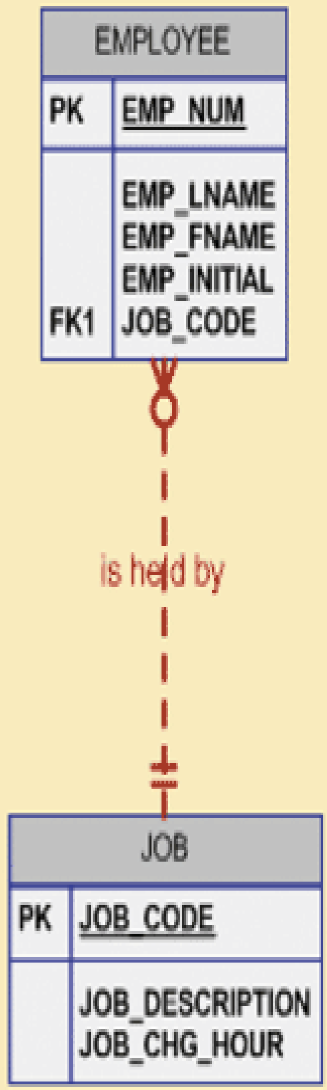
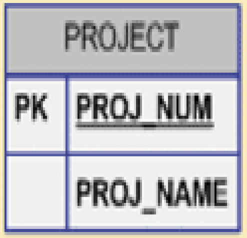


FIGURE 6.13

Modified contracting company ERD





Each EMPLOYEE has one (main) JOB classification.
Any JOB classification may be held by many EMPLOYEEs.

Some JOB classifications have not yet been staffed.
Therefore, EMPLOYEE is optional to JOB.

FIGURE 6.14

Incorrect M:N relationship representation

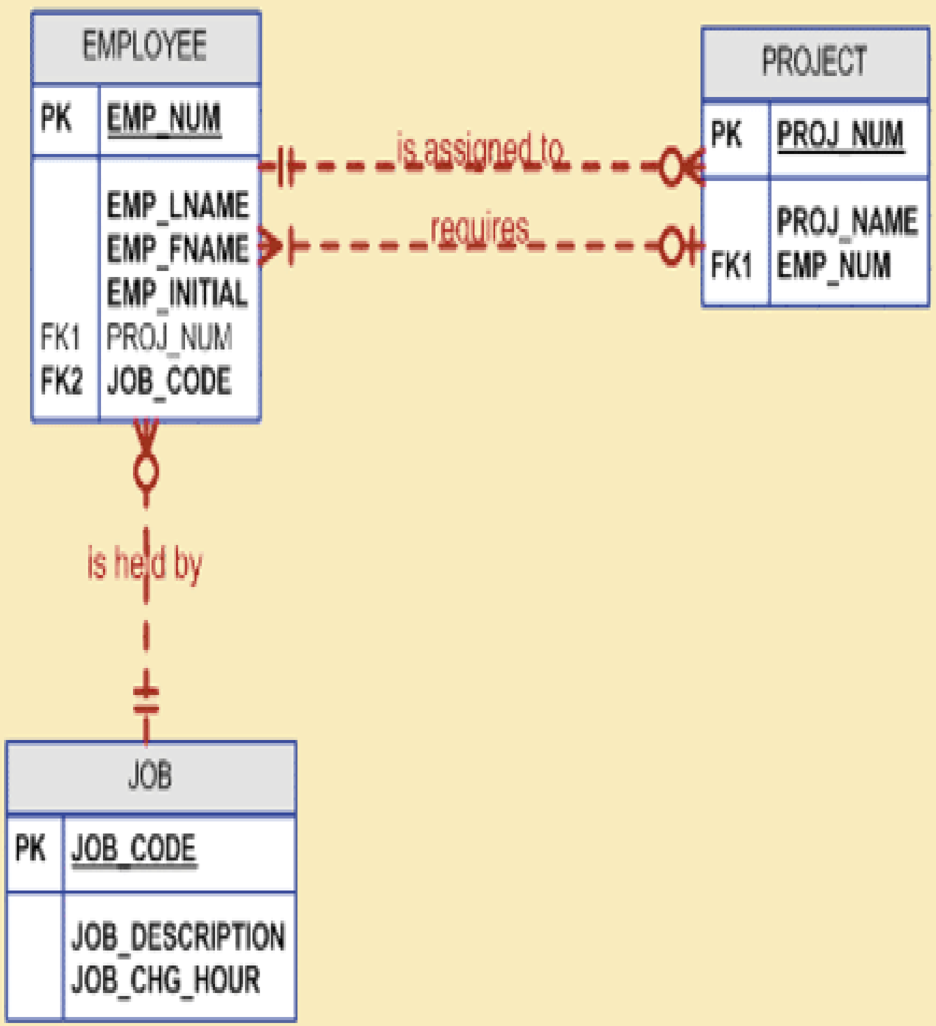


FIGURE 6.15 Final contracting company ERD

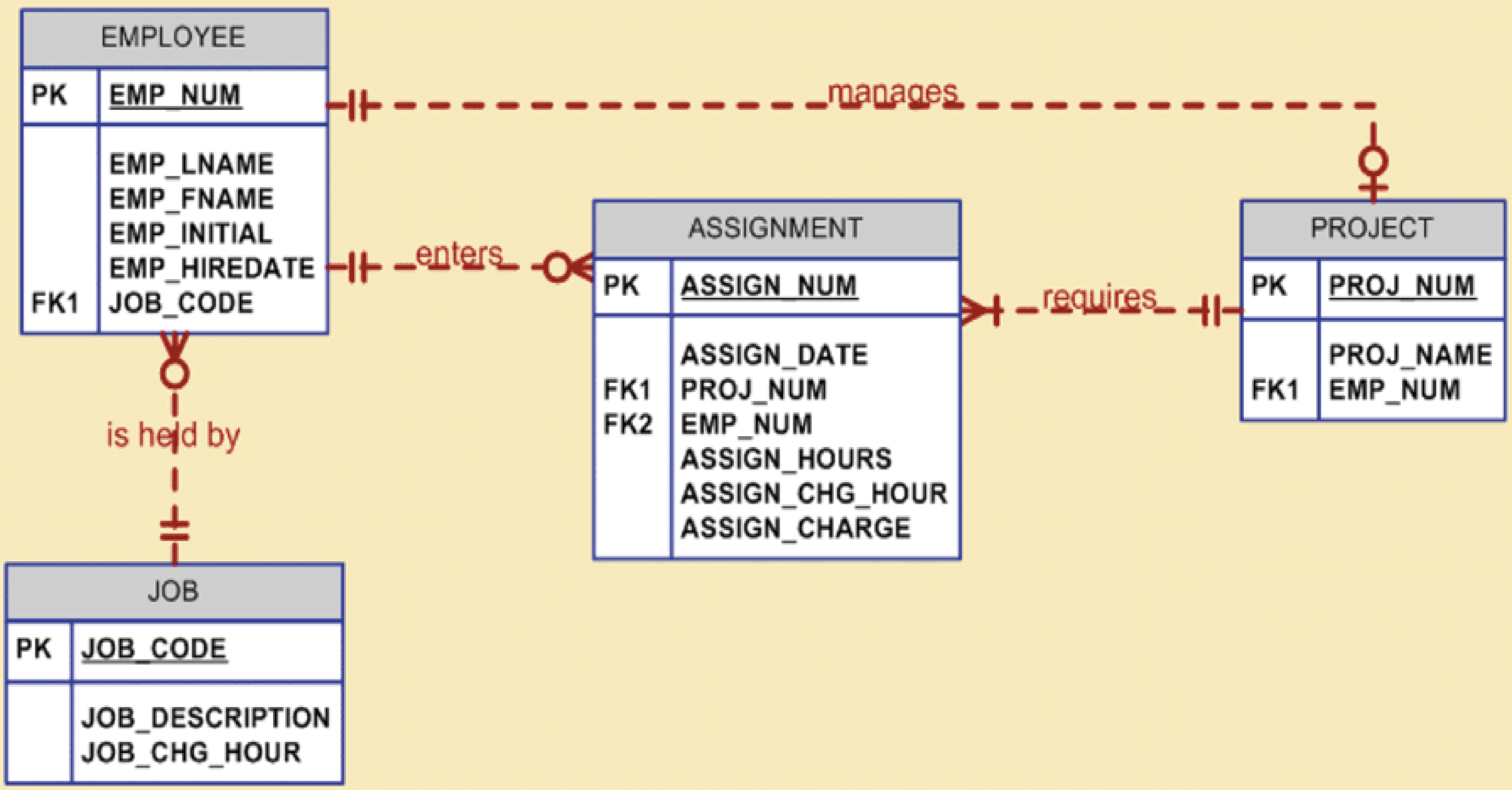


FIGURE 6.16 The implemented database

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

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-10	15	103	2.6	84.50	219.70
1002	04-Mar-10	18	118	1.4	18.36	25.70
1003	05-Mar-10	15	101	3.6	105.00	378.00
1004	05-Mar-10	22	113	2.5	48.10	120.25
1005	05-Mar-10	15	103	1.9	84.50	160.55
1006	05-Mar-10	25	115	4.2	96.75	406.35
1007	05-Mar-10	22	105	5.2	105.00	546.00
1008	05-Mar-10	25	101	1.7	105.00	178.50
1009	05-Mar-10	15	105	2.0	105.00	210.00
1010	06-Mar-10	15	102	3.8	96.75	367.65
1011	06-Mar-10	22	104	2.6	96.75	251.55
1012	06-Mar-10	15	101	2.3	105.00	241.50
1013	06-Mar-10	25	114	1.8	48.10	86.58
1014	06-Mar-10	22	111	4.0	26.87	107.48
1015	06-Mar-10	25	114	3.4	48.10	163.54
1016	06-Mar-10	18	112	1.2	45.95	55.14
1017	06-Mar-10	18	118	2.0	18.36	36.72
1018	06-Mar-10	18	104	2.6	96.75	251.55
1019	06-Mar-10	15	103	3.0	84.50	253.50
1020	07-Mar-10	22	105	2.7	105.00	283.50
1021	08-Mar-10	25	108	4.2	96.75	406.35
1022	07-Mar-10	25	114	5.8	48.10	278.98
1023	07-Mar-10	22	106	2.4	35.75	85.80

Denormalization

- Creation of normalized relations is important database design goal
- Processing requirements should also be a goal
- If tables are decomposed to conform to normalization requirements:
  - Number of database tables expands

Denormalization (cont'd.)

- Joining the larger number of tables reduces system speed
- Conflicts are often resolved through compromises that may include denormalization
- Defects of unnormalized tables:
  - Data updates are less efficient because tables are larger
  - Indexing is more cumbersome
  - No simple strategies for creating virtual tables known as views