Normalization

The Need of Normalization

TABLE 5.1 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

FIGURE 5.1 Tabular representation of the report format

Table name: RPT_FORMAT

Database name: Ch05_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

Structural Deficiencies

- The structure of the data set in Figure 5.1 does not conform to the requirements to handle data very well. Consider the following deficiencies:

1. The project number (PROJ_NUM) is apparently intended to be a primary key or at least a part of a PK, but it contains nulls. (PROJ_NUM + EMP_NUM will define each row.)
2. The table entries invite data inconsistencies. For example, the JOB_CLASS value "Elect. Engineer" might be entered as "Elect.Eng." in some cases, "El. Eng." in others, and "EE" in still others.

Anomalies

3. The table displays data redundancies. Those data redundancies yield the following anomalies:

Update anomalies. Modifying the JOB_CLASS for employee number 105 requires (potentially) many 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. To prevent the loss of the project information, a fictitious employee must be created just to save the project information.

Other than Structural Deficiencies

- For "Database Designer" report will not include data for "DB Design" and "Database Design" data entries. Such reporting anomalies cause a multitude of problems for managers—and cannot be fixed through applications programming.
- Remember that the naming convention makes it easy to see what each attribute stands for and what its likely origin is. For example, PROJ_NAME uses the prefix PROJ to indicate that the attribute is associated with the PROJECT table, while the NAME component is self-documenting, too. However, keep in mind that name length is also an issue, especially in the prefix designation. For that reason, the prefix CHG was used rather than CHARGE. (Given the database's context, it is not likely that that prefix will be misunderstood.)

The Normalization Process

TABLE 5.2 Normal Forms

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

Normalization of Database Tables

- What normalization is and what role it plays in the database design process
- About the normal forms 1NF, 2NF, 3NF, BCNF, 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

- Normalization is a process for evaluating and correcting table structures to minimize data redundancies, thereby reducing the likelihood of data anomalies.
- Normalization works through a series of stages called normal forms. The first three stages are described as first normal form (1NF), second normal form (2NF), and third normal form (3NF). From a structural point of view, 2NF is better than 1NF, and 3NF is better than 2NF. For most purposes in business database design, 3NF is as high as you need to go in the normalization process.

Conversion to First Normal Form

- Step 1: Eliminate the Repeating Groups

FIGURE 5.2 A table in first normal form

Table name: DATA_ORG_1NF   Database name: Ch05_ConstructCo

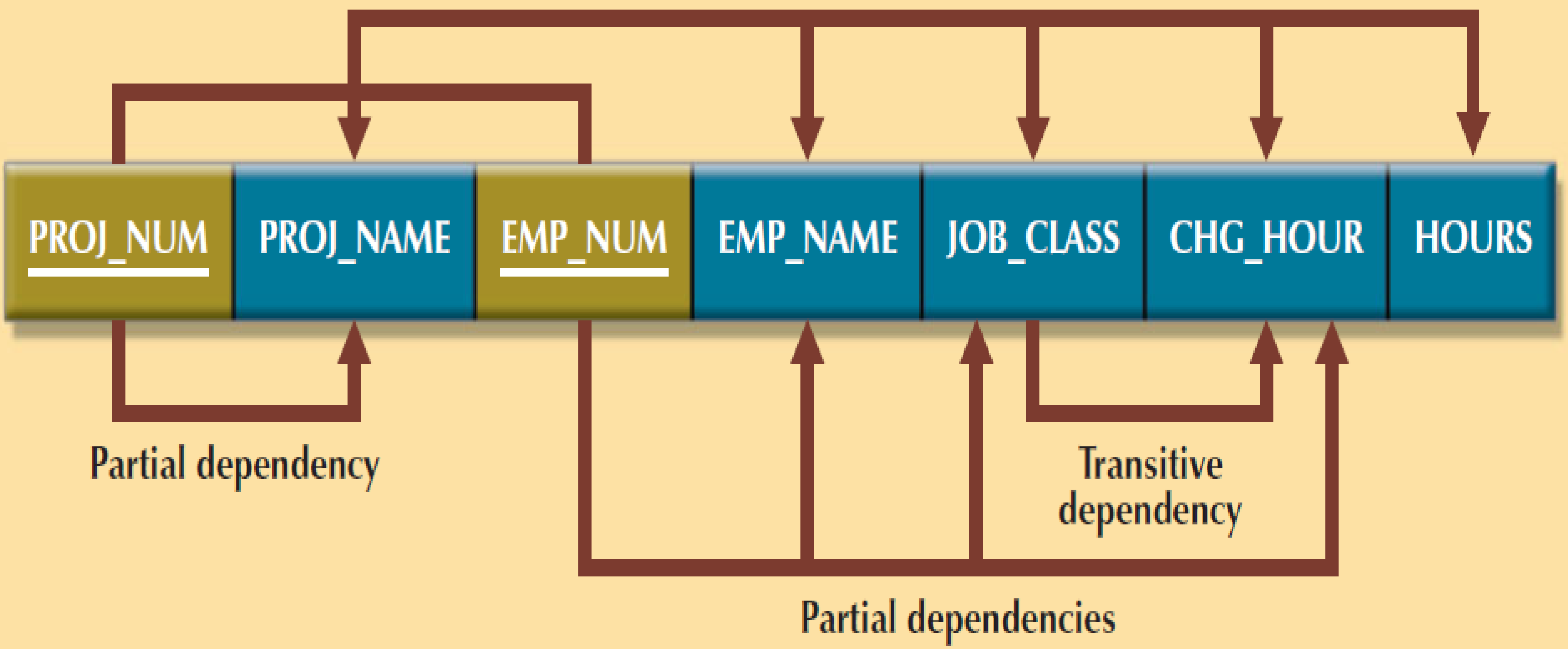
PROJ_NUM	PROJ_NAME	EMP_NUM	EMP_NAME	JOB_CLASS	CHG_HOUR	HOURS
15	Evergreen	103	June E. Arbough	Elect. Engineer	84.50	23.8
15	Evergreen	101	John G. News	Database Designer	105.00	19.4
15	Evergreen	105	Alice K. Johnson *	Database Designer	105.00	35.7
15	Evergreen	106	William Smithfield	Programmer	35.75	12.6
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

- Step 2: Identify the Primary Key
  - a combination of PROJ_NUM and EMP_NUM.
- Step 3: Identify All Dependencies

- PROJ_NUM, EMP_NUM → PROJ_NAME, EMP_NAME, JOB_CLASS, CHG_HOUR, HOURS
- PROJ_NUM → PROJ_NAME
- EMP_NUM → EMP_NAME, JOB_CLASS, CHG_HOUR
- JOB_CLASS → CHG_HOUR

FIGURE 5.3 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)

The term **first normal form** (1NF) describes the tabular format in which:

- All of the key attributes are defined.
- There are no repeating groups in the table. In other words, each row/column intersection contains one and only one value, not a set of values.
- All attributes are dependent on the primary key.

Conversion to Second Normal Form

- Step 1: Write Each Key Component on a Separate Line:
  - PROJ_NUM
  - EMP_NUM
  - PROJ_NUM & EMP_NUM
- Step 2: Assign Corresponding Dependent Attributes
  - PROJECT (**PROJ_NUM**, PROJ_NAME)
  - EMPLOYEE (**EMP_NUM**, EMP_NAME, JOB_CLASS, CHG_HOUR)
  - ASSIGNMENT (**PROJ_NUM, EMP_NUM,** ASSIGN_HOURS)

FIGURE 5.4 Second normal form (2NF) conversion results

Table name: PROJECT

PROJECT (PROJ_NUM, PROJ_NAME)

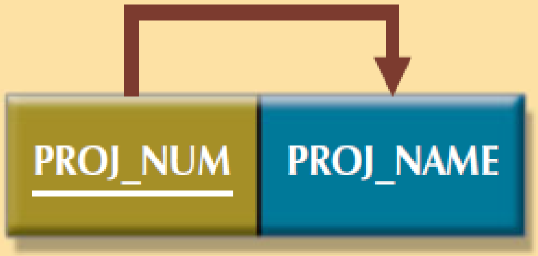


Table name: EMPLOYEE

EMPLOYEE (EMP_NUM, EMP_NAME, JOB_CLASS, CHG_HOUR)

TRANSITIVE DEPENDENCY
(JOB_CLASS → CHG_HOUR)

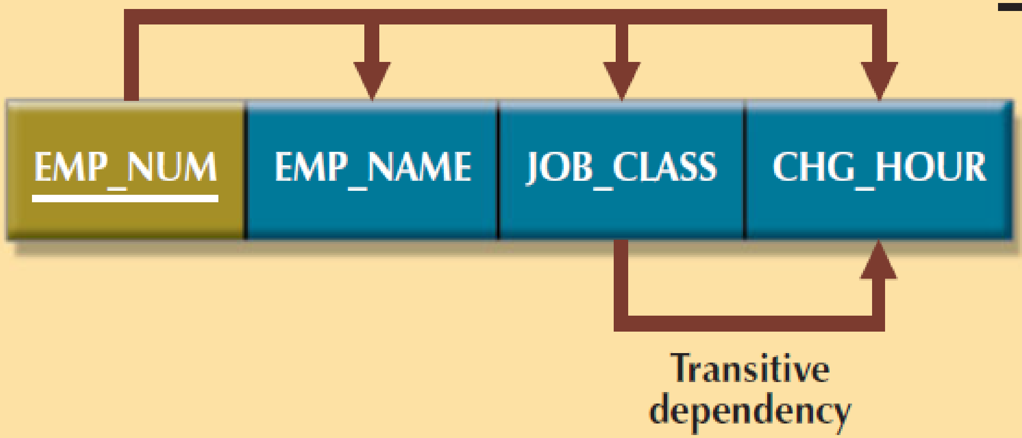
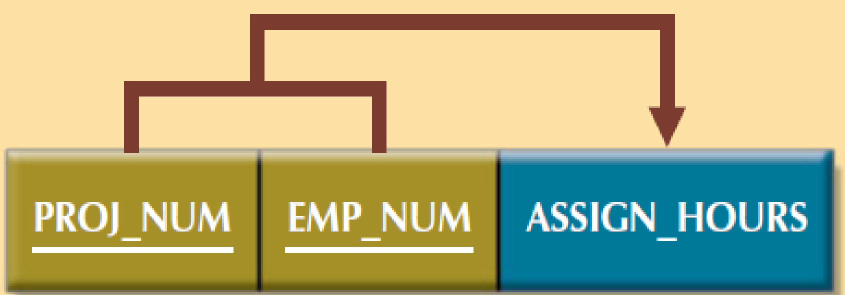


Table name: ASSIGNMENT

ASSIGNMENT (PROJ_NUM, EMP_NUM, ASSIGN_HOURS)



A table is in **second normal form** (2NF) when:

- It is in 1NF.

and

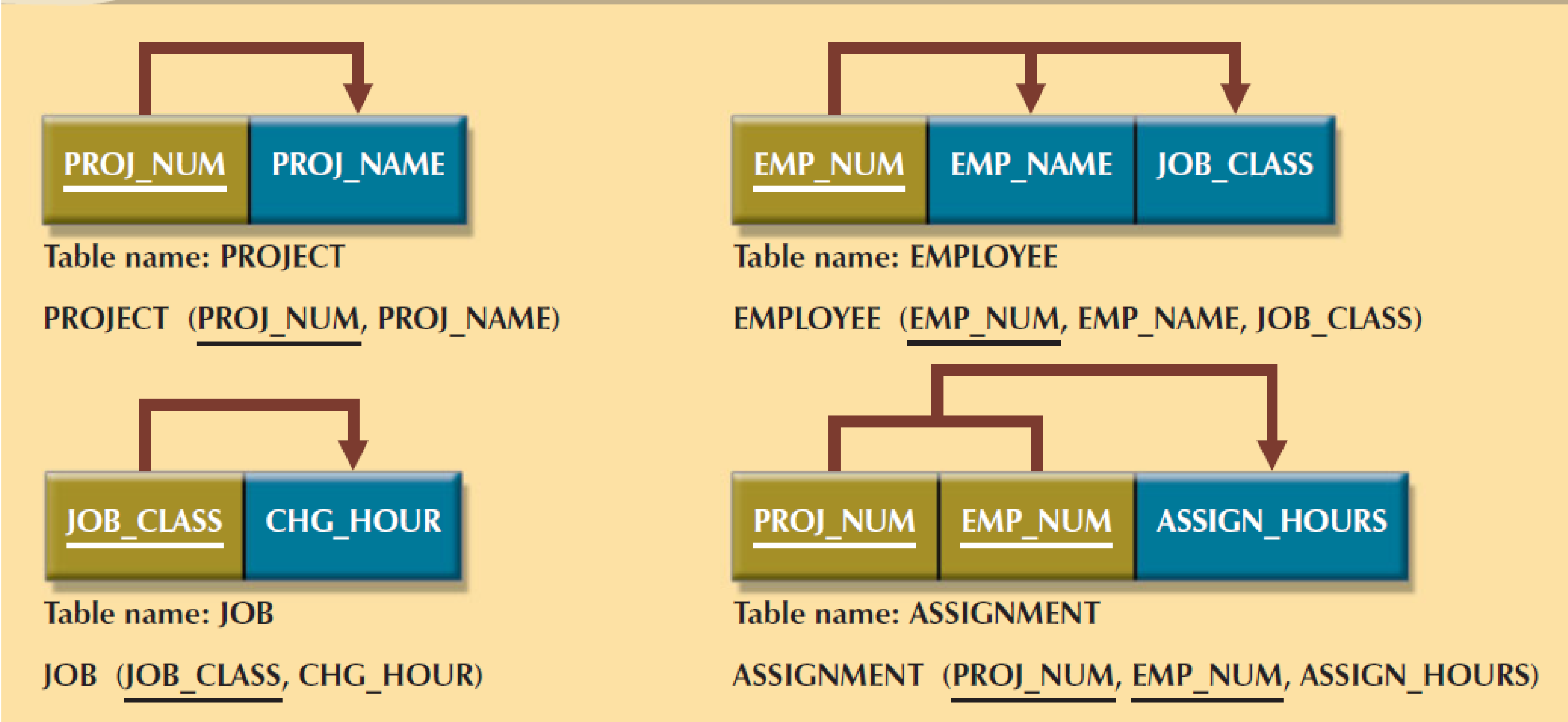
- It includes no partial dependencies; that is, no attribute is dependent on only a portion of the primary key.

Note that it is still possible for a table in 2NF to exhibit transitive dependency; that is, one or more attributes may be functionally dependent on nonkey attributes.

Conversion to Third Normal Form

- Step 1: Identify Each New Determinant
  - A determinant is any attribute whose value determines other values within a row.
  - JOB_CLASS
- Step 2: Identify the Dependent Attributes
  - JOB_CLASS → CHG_HOUR
- Step 3: Remove the Dependent Attributes from Transitive Dependencies
  - EMP_NUM → EMP_NAME, JOB_CLASS

FIGURE 5.5 Third normal form (3NF) conversion results



In other words, after the 3NF conversion has been completed, your database contains four tables:

PROJECT (**PROJ_NUM**, PROJ_NAME)

EMPLOYEE (**EMP_NUM**, EMP_NAME, JOB_CLASS)

JOB (**JOB_CLASS**, CHG_HOUR)

ASSIGNMENT (**PROJ_NUM**, **EMP_NUM**, ASSIGN_HOURS)

A table is in **third normal form** (3NF) when:

- It is in 2NF.

and

- It contains no transitive dependencies.

Improving The Design

- **Evaluate PK Assignments**
  - JOB_CODE → JOB_CLASS, CHG_HOUR
  - transitive dependency:
    - JOB_CLASS → CHG_HOUR
  - JOB_CODE is the chosen primary key as well as a surrogate key.
  - A surrogate key is an artificial PK introduced by the designer with the purpose of simplifying the assignment of primary keys to tables. Surrogate keys are usually numeric, they are often automatically generated by the DBMS, they are free of semantic content (they have no special meaning), and they are usually hidden from the end users.

Improving The Design

- **Evaluate Naming Conventions**
  - HOURS was changed to ASSIGN_HOURS
- **Refine Attribute Atomicity**
  - An atomic attribute is one that cannot be further subdivided. Such an attribute is said to display atomicity.
  - EMP_LNAME,
  - EMP_FNAME, and EMP_INITIAL
- **Identify New Attributes**
  - in a real-world environment, adding an employee hire date attribute (EMP_HIREDATE) could be used to track an employee's job longevity and serve as a basis for awarding bonuses to long-term employees and for other morale-enhancing measures.

Improving The Design

- **Identify New Relationships**
  - The system's ability to supply detailed information about each project's manager is ensured by using the EMP_NUM as a foreign key in PROJECT. That action ensures that you can access the details of each PROJECT's manager data without producing unnecessary and undesirable data duplication.
- **Refine Primary Keys as Required for Data Granularity**
  - Granularity refers to the level of detail represented by the values stored in a table's row. Data stored at their lowest level of granularity are said to be atomic data.

Improving The Design

- **Refine Primary Keys as Required for Data Granularity**
  - Do the ASSIGN_HOURS represent the hourly total, daily total, weekly total, monthly total, or yearly total?
  - Clearly, ASSIGN_HOURS requires more careful definition.
- **Maintain Historical Accuracy**
  - It would be appropriate to name this attribute ASSIGN_CHG_HOUR. Although this attribute would appear to have the same value as JOB_CHG_HOUR, that is true only if the JOB_CHG_HOUR value remains forever the same.

Improving The Design

- **Evaluate Using Derived Attributes**
  - derived attribute, to be named ASSIGN_CHARGE, is the result of multiplying the ASSIGN_HOURS by the ASSIGN_CHG_HOUR.
  - storing the derived attribute in the table makes it easy to write the application software to produce the desired results.
  - if many transactions must be reported and/or summarized, the availability of the derived attribute will save reporting time

FIGURE 5.6 **The completed database**

Table name: PROJECT

Table name: JOB

Database name: Ch05_ConstructCo

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: 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: ASSIGNMENT

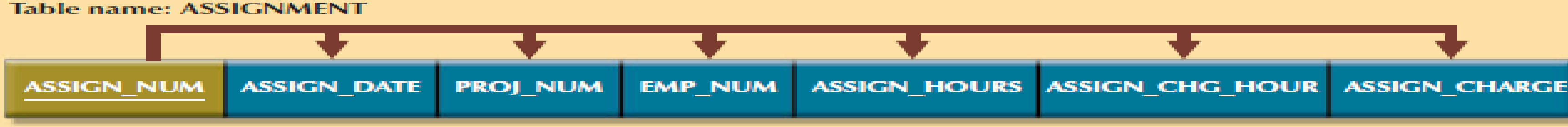


Table name: ASSIGNMENT

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

FIGURE 5.6 The completed database (continued)

Table name: EMPLOYEE

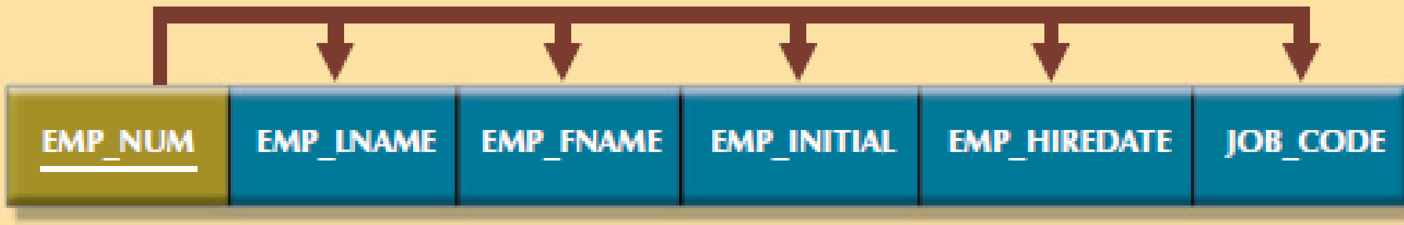


Table name: EMPLOYEE

EMP_NUM	EMP_LNAME	EMP_FNAME	EMP_INITIAL	EMP_HIREDATE	JOB_CODE
101	News	John	G	00-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

- In any case, if JOB_CODE is to be the surrogate PK, you still must ensure the existence of unique values in the JOB_DESCRIPTION through the use of a unique index.

Higher-level Normal Forms

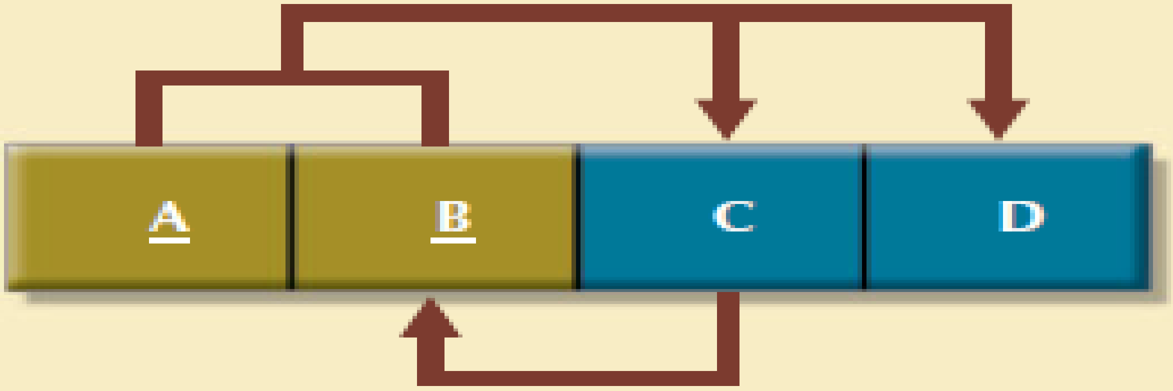
- Tables in 3NF will perform suitably in business transactional databases. However, there are occasions when higher normal forms are useful.
- A special case of 3NF, known as Boyce-Codd normal form (BCNF), and fourth normal form (4NF)

The Boyce-Codd Normal Form (BCNF)

- A table is in Boyce-Codd normal form (BCNF) when every determinant in the table is a candidate key.
- when a table contains only one candidate key, the 3NF and the BCNF are equivalent.
- Putting that proposition another way, BCNF can be violated only when the table contains more than one candidate key.

FIGURE 5.7

A table that is in 3NF but not in BCNF

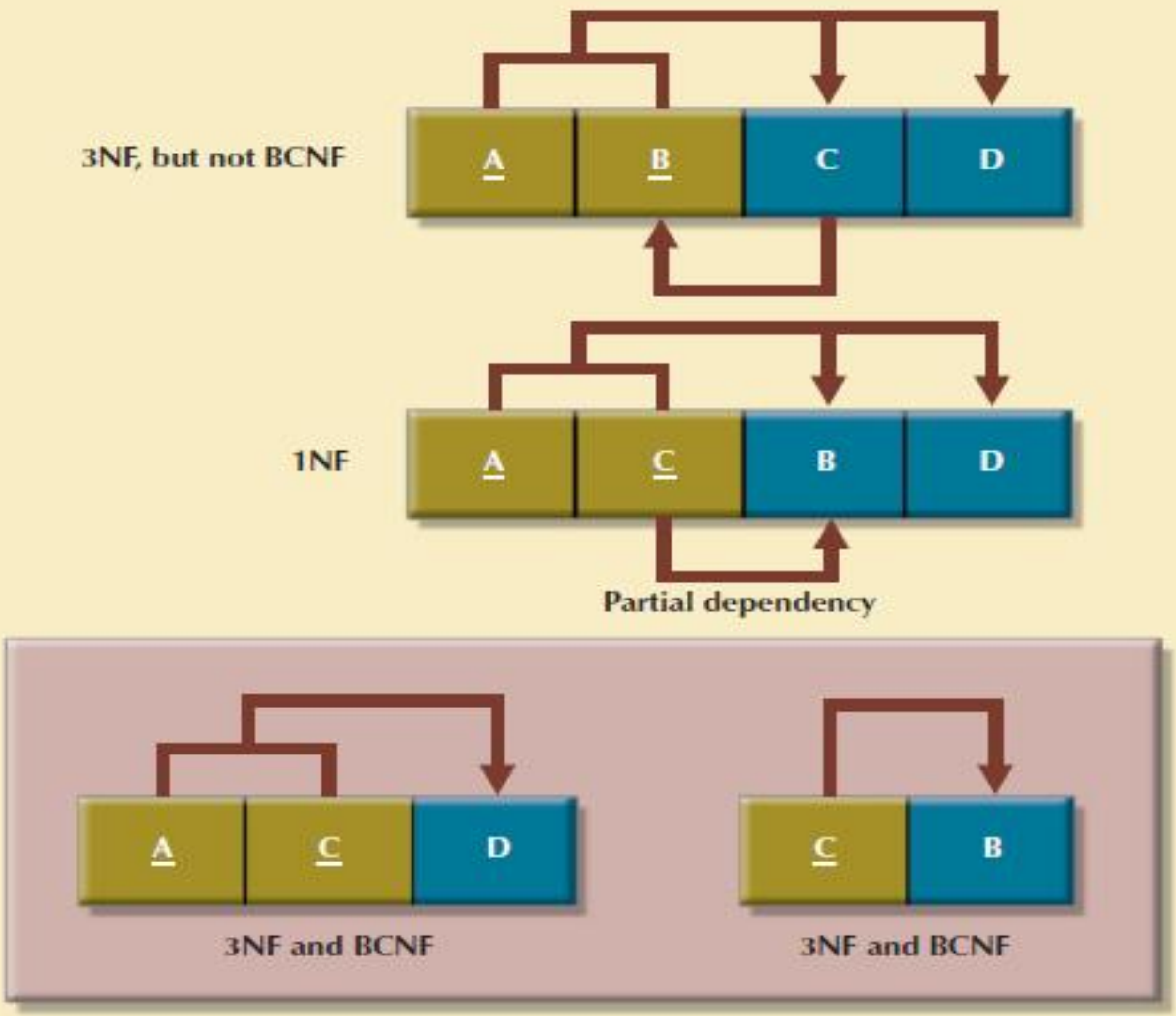


The Boyce-Codd Normal Form (BCNF)

- The table structure shown in Figure 5.7 has no partial dependencies, nor does it contain transitive dependencies. (The condition C → B indicates that a nonkey attribute determines part of the primary key—and that dependency is not transitive!) Thus, the table structure in Figure 5.7 meets the 3NF requirements. Yet the condition C → B causes the table to fail to meet the BCNF requirements.

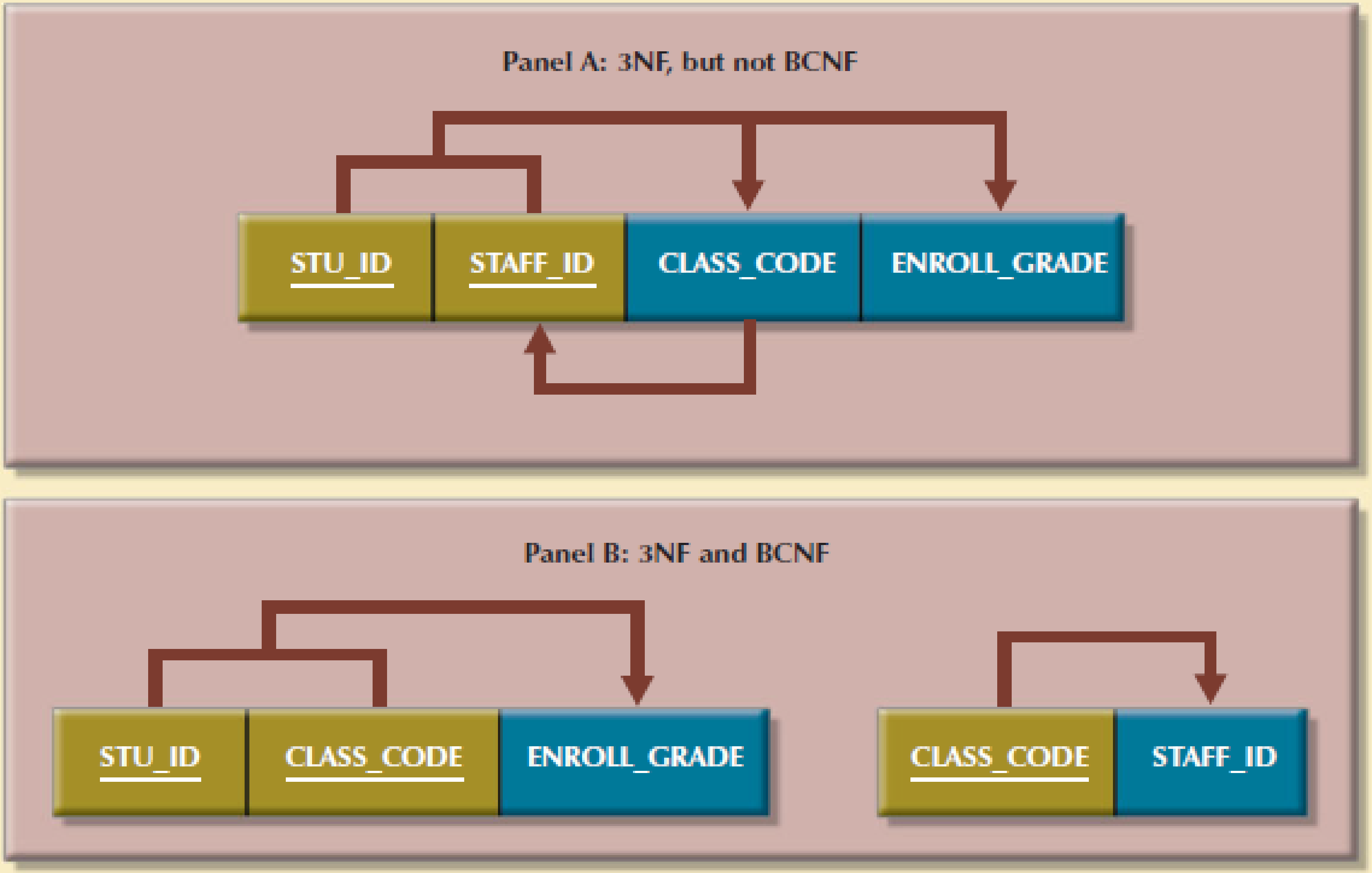
The Boyce-Codd Normal Form (BCNF)

FIGURE 5.8 Decomposition to BCNF



The Boyce-Codd Normal Form (BCNF)

FIGURE 5.9 Another BNCF decomposition



Fourth Normal Form (4NF)

- When encountered with poorly designed databases, or it can be asked to convert spreadsheets into a database format in which multiple multivalued attributes exist.

FIGURE 5.10 Tables with multivalued dependencies

Database name: Ch05_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
10223		4

Table name: VOLUNTEER_V3

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

Fourth Normal Form (4NF)

- Specifically, the discussion of 4NF is largely academic if you make sure that your tables conform to the following two rules:

1. All attributes must be dependent on the primary key, but they must be independent of each other.
2. No row may contain two or more multivalued facts about an entity.

FIGURE 5.11 A set of tables in 4NF

NOTE

A table is in **fourth normal form (4NF)** when it is in 3NF and has no multiple sets of multivalued dependencies.

Table name: EMPLOYEE

Database name: Ch05_Service

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

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: ORGANIZATION

ORG_CODE	ORG_NAME
RC	Red Cross
UW	United Way
WF	Wildlife Fund

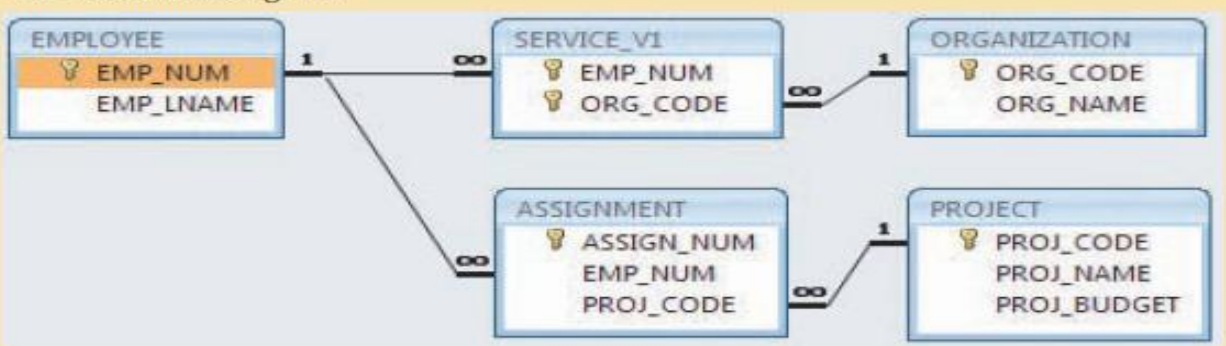
Table name: ASSIGNMENT

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

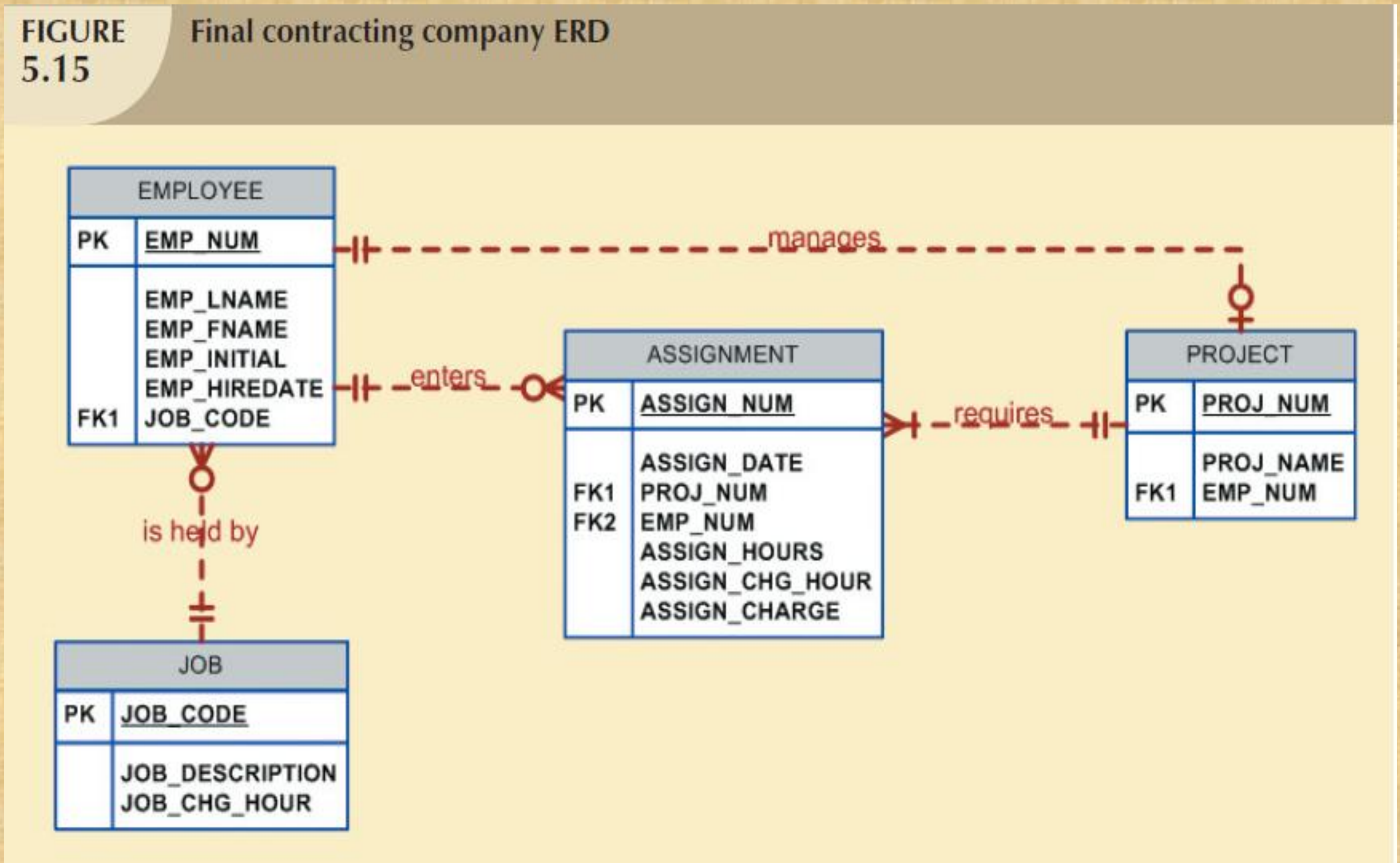
Table name: SERVICE_V1

EMP_NUM	ORG_CODE
10123	RC
10123	UW
10123	WF

The relational diagram



Normalization And Database Design



Normalization And Database Design

PROJECT (**PROJ_NUM**, PROJ_NAME, EMP_NUM)

EMPLOYEE (**EMP_NUM**, EMP_LNAME, EMP_FNAME, EMP_INITIAL, EMP_HIREDATE, JOB_CODE)

JOB (**JOB_CODE**, JOB_DESCRIPTION, JOB_CHG_HOUR)

ASSIGNMENT (**ASSIGN_NUM**, ASSIGN_DATE, PROJ_NUM, EMP_NUM, ASSIGN_HOURS, ASSIGN_CHG_HOUR, ASSIGN_CHARGE)

FIGURE 5.16 The implemented database

Database name: Ch05_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-08	15	103	2.6	84.50	219.70
1002	04-Mar-08	18	118	1.4	18.36	25.70
1003	05-Mar-08	15	101	3.6	105.00	378.00
1004	05-Mar-08	22	113	2.5	48.10	120.25
1005	05-Mar-08	15	103	1.9	84.50	160.55
1006	05-Mar-08	25	115	4.2	96.75	406.35
1007	05-Mar-08	22	105	5.2	105.00	546.00
1008	05-Mar-08	25	101	1.7	105.00	178.50
1009	05-Mar-08	15	105	2.0	105.00	210.00
1010	06-Mar-08	15	102	3.8	96.75	367.65
1011	06-Mar-08	22	104	2.6	96.75	251.55
1012	06-Mar-08	15	101	2.3	105.00	241.50
1013	06-Mar-08	25	114	1.8	48.10	86.58
1014	06-Mar-08	22	111	4.0	26.87	107.48
1015	06-Mar-08	25	114	3.4	48.10	163.54
1016	06-Mar-08	18	112	1.2	45.95	55.14
1017	06-Mar-08	18	118	2.0	18.36	36.72
1018	06-Mar-08	18	104	2.6	96.75	251.55
1019	06-Mar-08	15	103	3.0	84.50	253.50
1020	07-Mar-08	22	105	2.7	105.00	283.50
1021	08-Mar-08	25	108	4.2	96.75	406.35
1022	07-Mar-08	25	114	5.8	48.10	278.98
1023	07-Mar-08	22	106	2.4	35.75	85.80

Denormalization

- Occasionally it is expected to denormalize some portions of a database design in order to meet performance requirements.
- Denormalization produces a lower normal form; that is, a 3NF will be converted to a 2NF through denormalization. However, the price which is payed for increased performance through denormalization is greater data redundancy.

Denormalization

- Good database design also considers processing (or reporting) requirements and processing speed. The problem with normalization is that as tables are decomposed to conform to normalization requirements, the number of database tables expands. Therefore, in order to generate information, data must be put together from various tables.

Denormalization

- Joining a large number of tables takes additional input/output (I/O) operations and processing logic, thereby reducing system speed. Most relational database systems are able to handle joins very efficiently. However, rare and occasional circumstances may allow some degree of denormalization so processing speed can be increased.

Denormalization

- For example, should people in a real-world database environment worry that a ZIP_CODE determines CITY in a CUSTOMER table whose primary key is the customer number? Is it really practical to produce a separate table for
  - ZIP (ZIP_CODE, CITY)

TABLE 5.6 Common Denormalization Examples

CASE	EXAMPLE	RATIONALE AND CONTROLS
Redundant data	Storing ZIP and CITY attributes in the CUSTOMER table when ZIP determines CITY. (See Table 1.3.)	• 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 3.29.)	• Avoid extra join operations • Program can validate classification (lookup) based on the student hours
Pre-aggregated 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 3.29.)	• 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 is stored in the table as rows. (See Figure 5.17 and Figure 5.18.)	• 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.

End of Course