

Chapter 6

- Objectives: to learn
 - 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

CS275 Fall 2010

1

Database Tables & Normalization

- **Normalization:**
 - A process for assigning attributes to entities
 - Reduces data redundancies
 - Helps eliminate data anomalies
 - Produces controlled redundancies to link tables
- **Normal Forms** are a series of stages done in Normalization
 - 1NF - First normal form,
 - 2NF - Second normal form,
 - 3NF - Third normal form,
 - 4NF - Fourth normal form

CS275 Fall 2010

2

Database Tables & Normalization

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

CS275 Fall 2010

3

The Need for Normalization

- Example: Company which manages building projects.
- The business rules are:
 - 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

CS275 Fall 2010

4

The Need for Normalization

- Desired Output - Classic control-break report. A common type of report from a database.

TABLE 6.1 A Sample Report Layout

PROJECT #	PROJECT NAME	EMPLOYEE NUMBER	EMPLOYEE NAME	JOB CLASS	STARTED DATE	PAID TO DATE	STARTED AMOUNT	PAID TO AMOUNT	STARTED HOURS	PAID TO HOURS
15	Evergreen	101	June E. Arbough	Elect. Engineer	\$ 85.50	21.8	\$ 2,254.50			
		101	John G. News	Database Designer	\$105.00	19.4	\$ 2,037.00			
		105	Alice K. Johnson *	Database Designer	\$105.00	15.7	\$ 1,508.50			
		106	William Smithfield	Programmer	\$ 33.75	12.6	\$ 430.43			
		102	David H. Semon	Systems Analyst	\$ 96.75	23.8	\$ 2,293.43			
		Subtotal				\$18,173.50				
18	Amber Wave	114	Annalise Jones	Applications Designer	\$ 48.10	25.6	\$ 1,181.20			
		118	James J. Frommer	General Support	\$ 18.36	43.3	\$ 813.73			
		104	Anne K. Ramonas *	Systems Analyst	\$ 96.75	32.4	\$ 3,114.75			
		112	Darlene M. Smithson	DSS Analyst	\$ 45.93	45.0	\$ 2,062.73			
				Subtotal				\$ 7,262.52		
22	Rolling Tide	105	Alice K. Johnson	Database Designer	\$105.00	65.7	\$ 6,796.50			
		104	Anne K. Ramonas	Systems Analyst	\$ 96.75	48.4	\$ 4,682.70			
		112	Darlene M. Smithson	Applications Designer	\$ 48.10	23.6	\$ 1,115.16			
		111	Geoff B. Wabash	Clinical Support	\$ 26.87	22.0	\$ 591.14			
		106	William Smithfield	Programmer	\$ 33.75	12.8	\$ 437.60			
		Subtotal				\$13,755.50				
25	Starflight	107	Maria D. Alonso	Programmer	\$ 33.75	25.6	\$ 915.20			
		115	Travis B. Stawangi	Systems Analyst	\$ 96.75	45.8	\$ 4,413.15			
		101	John G. News *	Database Designer	\$105.00	56.3	\$ 5,913.00			
		114	Annalise Jones	Applications Designer	\$ 48.10	33.1	\$ 1,592.11			
		108	Ralph B. Washington	Systems Analyst	\$ 96.75	23.6	\$ 2,283.30			
		Subtotal				\$15,916.76				
		Total				\$49,199.69				

Note: * indicates project broker

CS275 Fall 2010

5

The Need for Normalization

- Data often comes from tabular reports

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	101	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	15.7
		106	William Smithfield	Programmer	35.75	12.6
		102	David H. Semon	Systems Analyst	96.75	23.8
18	Amber Wave	114	Annalise Jones	Applications Designer	48.10	24.6
		118	James J. Frommer	General Support	18.36	45.3
		104	Anne K. Ramonas *	Systems Analyst	96.75	32.4
		112	Darlene M. Smithson	DSS Analyst	45.93	44.0
		106	William Smithfield	Programmer	35.75	12.8
22	Rolling Tide	105	Alice K. Johnson	Database Designer	105.00	64.7
		104	Anne K. Ramonas	Systems Analyst	96.75	48.4
		112	Darlene M. Smithson	Applications Designer	48.10	23.6
		111	Geoff B. Wabash	Clinical Support	26.87	22.0
		106	William Smithfield	Programmer	35.75	12.8
25	Starflight	107	Maria D. Alonso	Programmer	35.75	24.6
		115	Travis B. Stawangi	Systems Analyst	96.75	45.8
		101	John G. News *	Database Designer	105.00	56.3
		114	Annalise Jones	Applications Designer	48.10	33.1
		108	Ralph B. Washington	Systems Analyst	96.75	23.6

CS275 Fall 2010

6

Creating Entities from Tabular Data

- Structure of data set in Figure 6.1 does not handle data very well
 - Primary key - Project # contains nulls
 - Table displays data redundancies
- Report may yield different results depending on what data anomaly has occurred
 - Update - Modifying JOB_CLASS
 - Insertion - New employee must be assigned project
 - Deletion - If employee deleted, other vital data lost

CS275 Fall 2010

7

The Normalization Process

- Relational database environment is suited to help designer avoid data integrity problems
 - 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
- Normalizing table structure will reduce data redundancies

CS275 Fall 2010

8

The Normalization Process

- Objective of normalization is to ensure that all tables are in at least 3NF
- Normalization works one Entity at a time
- It progressively breaks table into new set of relations based on identified dependencies
- Normalization from 1NF to 2NF is three-step procedure.

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

CS275 Fall 2010

9

Conversion to First Normal Form

- 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 values
 - New key can be composed of multiple attributes
- Step 3: Identify All Dependencies
 - Dependencies are depicted with a diagram

CS275 Fall 2010

10

Step 1: Conversion to 1NF

- Step 1: Eliminate the Repeating Groups
 - A Repeating group is group of multiple entries of same type existing for any single key attribute occurrence
 - 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 Repeating groups must be eliminated

CS275 Fall 2010

11

Step 1 - Eliminate the Repeating Groups

FIGURE 6.2 A table in first normal form

Table name: DATA_ORG_1NF				Database name: CM95_ConstructCo	
PROJ_NUM1	PROJ_NAME	EMP_NUM	EMP_NAME	JOB_CLASS	CHG_HOUR HOURLY
15	Evergreen	103	Jane E. Adrough	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 20.7
15	Evergreen	106	William Smithfield	Programmer	35.75 12.6
15	Evergreen	102	David M. Senior	Systems Analyst	95.75 22.8
19	Amber Ware	114	Amanda Jones	Applications Designer	48.10 24.6
19	Amber Ware	119	James J. Frommer	General Support	18.36 45.3
18	Amber Ware	104	Anne K. Romano *	Systems Analyst	95.75 32.4
18	Amber Ware	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. Romano	Systems Analyst	95.75 48.4
22	Rolling Tide	113	Delbert K. Jambrook *	Applications Designer	48.10 23.6
22	Rolling Tide	111	Gord B. Walsath	Clinical Support	25.87 22.0
22	Rolling Tide	106	William Smithfield	Programmer	35.75 12.8
25	Starlight	107	Maria D. Absence	Programmer	35.75 24.8
25	Starlight	115	Tara B. Blawerg	Systems Analyst	95.75 45.6
25	Starlight	101	John G. News *	Database Designer	105.00 56.3
25	Starlight	114	Amanda Jones	Applications Designer	48.10 32.1
25	Starlight	108	Ralph B. Washington	Systems Analyst	95.75 23.6
25	Starlight	119	James J. Frommer	General Support	18.36 30.5
25	Starlight	112	Darlene M. Smithson	DSS Analyst	45.95 41.4

CS275 Fall 2010

12

Step 2 - Conversion to 1NF

- Step 2 - Identify the Primary Key
 - Review (from Chapter 3) Determination and attribute dependence.

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 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.

- All attribute values in the occurrence are 'determined' by the Primary Key. The Primary Key Must uniquely identify the attribute(s).
- Resulting Composite Key : PROJ_NUM and EMP_NUM

CS275 Fall 2010

13

Step 3- Conversion to 1NF

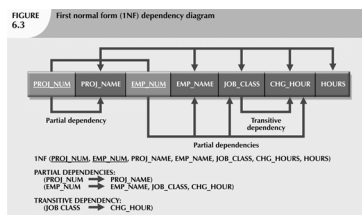
- Step 3 - Identify All Dependencies
 - Depicts all dependencies found within given table structure
 - Helpful in getting bird's-eye view of all relationships among table's attributes
 1. Draw desirable dependencies based on PKey
 2. Draw less desirable dependencies
 - Partial
 - » based on part of composite primary key
 - Transitive
 - » one nonprime attribute depends on another nonprime attribute

CS275 Fall 2010

14

Step 3 - Dependency Diagram (1NF)

- The connections above the entity show attributes dependent on the currently chosen Primary Key, the combination of PROJ_NUM and EMP_NUM.
- The arrows below the dependency diagram indicate less desirable partial and transitive dependencies



CS275 Fall 2010

15

Resulting First Normal Form

- 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 other dependencies and should be used with caution
 - Partial dependencies - an attribute dependent on only part of the primary key
 - Transitive dependencies - an attribute dependent on another attribute that is not part of the primary key.

CS275 Fall 2010

16

Conversion to Second Normal Form

- Step 1: Eliminate Partial Dependencies
 - Start with 1NF format and convert by:
 - Write each part of the composite key on it's own line.
 - Write the original (composite) key on last line
 - Each component will become key in new table
- Step 2: Assign Dependent Attributes
 - From the original 1NF determine which attributes are dependent on which key attributes
- Step 3: Name the tables to reflect its contents & function

```
PROJECT (PROJ_NUM, PROJ_NAME)
EMPLOYEE (EMP_NUM, EMP_NAME, JOB_CLASS, CHG_HOUR)
ASSIGN (PROJ_NUM, EMP_NUM, HOURS)
```

CS275 Fall 2010

17

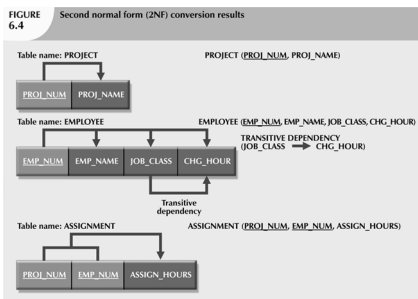
Completed Conversion to 2NF

- Each Key component establishes a new table
- 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
 - Note: it is still possible to exhibit transitive dependency
 - Attributes may be functionally dependent on nonkey attributes

CS275 Fall 2010

18

Completed Conversion to 2NF



CS275 Fall 2010

19

Conversion to Third Normal Form

- Step 1: Eliminate Transitive Dependencies
 - Write its determinant as PK for new table.
 - And Leave it in the Original Table
- Step 2: Reassign Corresponding Dependent Attributes
 - Identify attributes dependent on each determinant identified in Step 1, and list on new table.
- Step 3: Name the new table(s) to reflect its contents and function

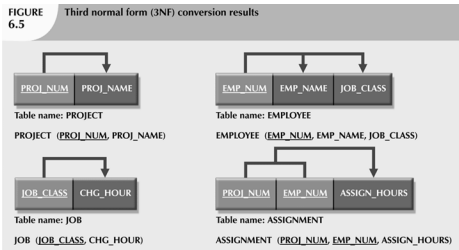
```
PROJECT (PROJ_NUM, PROJ_NAME)
EMPLOYEE (EMP_NUM, EMP_NAME, JOB_CLASS)
ASSIGN (PROJ_NUM, EMP_NUM, HOURS)
JOB (JOB_CLASS, CHG_HOUR)
```

CS275 Fall 2010

20

Resulting Third Normal Form

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



CS275 Fall 2010

21

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
- It reduces data redundancy and builds controlled redundancy.
- The higher the NF,
 - the more entities one has,
 - the more flexible the database will be,
 - the more joins (and less efficiency) you have.

CS275 Fall 2010

22

Improving the Design

- Additional issues to address and possibly change, in order to produce a good normalized set of tables:
 - Evaluate PK Assignments
 - Evaluate Naming Conventions
 - Refine Attribute Atomicity
 - Identify New Attributes
 - Identify New Relationships
 - Refine Primary Keys as Required for Data Granularity
 - Maintain Historical Accuracy
 - Evaluate Using Derived Attributes

CS275 Fall 2010

23

Surrogate Key Considerations

- When primary key is considered to be unsuitable, designers use surrogate keys
- System-assigned primary keys may not prevent confusing entries, but do prevent violation of entity integrity.
- Example: data entries in Table 6.4 are inappropriate because they duplicate existing records

TABLE 6.4 Duplicate Entries in the Job Table

JOB_CODE	JOB_DESCRIPTION	JOB_CHG_HOUR
511	Programmer	\$35.75
512	Programmer	\$35.75

CS275 Fall 2010

24

Improving the Design

- Identifying new attributes

FIGURE 6.6 The completed database (continued)

Table name: EMPLOYEE

EMP_NUM	EMP_INAME	EMP_FNAME	EMP_INITIAL	EMP_HIREDATE	JOB_CODE
101	Neve	John	D	09-Nov-07 502	
102	Senior	Dave	M	12-Jul-89 501	
103	Admough	Jane	E	01-Dec-87 503	
104	Flamoras	Anna	K	15-Nov-88 501	
105	Johanson	Alice	K	01-Feb-84 502	
106	Smithfield	William		23-Jun-95 500	
107	Alvico	Maria	D	10-Oct-84 500	
108	Washington	Plath	D	22-Aug-89 501	
109	Smith	Larry	W	18-Jul-89 501	
110	Olenka	Gerard	A	11-Dec-86 506	
111	Warrack	Geoff	D	04-Apr-89 506	
112	Smithson	Darlene	M	23-Oct-95 507	
113	Jordani	Darab	K	15-Nov-84 508	
114	Jones	Annellise	D	20-Aug-91 508	
115	Bewang	Tara	D	26-Nov-90 501	
116	Phyll	Gerard	L	06-Mar-95 510	
117	Hollmeyer	Angie	H	10-Jul-84 500	
118	Froemer	Jarvis	J	04-Jan-96 510	

CS275 Fall 2010

25

Higher-Level Normal Forms

- Tables in 3NF perform suitably in business transactional databases
- Higher-order normal forms are useful on occasion
- Two special cases of 3NF:
 - Boyce-Codd normal form (BCNF)
 - Fourth normal form (4NF)

CS275 Fall 2010

26

The Boyce-Codd Normal Form (BCNF)

- Every determinant in table is a candidate key
 - Has same characteristics as primary key, but for some reason, not chosen to be primary key
- When table contains only one candidate key, the 3NF and the BCNF are equivalent
- BCNF can be violated only when table contains more than one candidate key
 - example:

Section	(<u>coursename</u> , <u>sectionno</u> , time, days)
---------	--

CS275 Fall 2010

27

The Boyce-Codd Normal Form (BCNF)

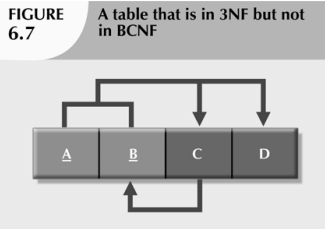
- Most designers consider the BCNF as a special case of 3NF
- Table is in 3NF when it is in 2NF and there are no transitive dependencies
- Table can be in 3NF and fail to meet BCNF
 - No partial dependencies, nor does it contain transitive dependencies
 - A nonkey attribute is the determinant of a key attribute

CS275 Fall 2010

28

The Boyce-Codd Normal Form (BCNF)

- When part of the key is dependent on another non-key attribute, ie. another candidate key.

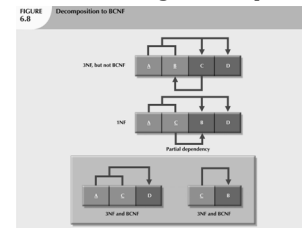


CS275 Fall 2010

29

The Boyce-Codd Normal Form (BCNF)

- Occurs most often when the wrong attribute was chosen as part of the composite Primary Key.
- Return to 2NF and correct by:
 - Create a new composite key with C, not B.
 - Create a new table eliminating the new partial dependency.



CS275 Fall 2010

30

The Boyce-Codd Normal Form (BCNF)

- Non-Boyce-Codd Normal Form
 - Can only exist with composite Primary Key –
 - Example Enroll entity: Enroll(Stu_ID, Staff_ID, Class_Code, Enroll_Grade)

TABLE 6.5 Sample Data for a BCNF Conversion

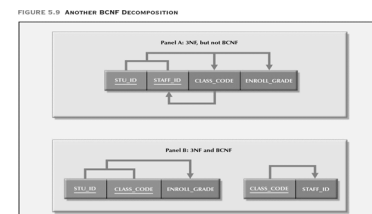
STU_ID	STAFF_ID	CLASS_CODE	ENROLL_GRADE
125	25	21334	A
125	20	32456	C
135	20	28458	B
144	25	27563	C
144	20	32456	B

CS275 Fall 2010

31

The Boyce-Codd Normal Form (BCNF)

- Resulting BCNF with two entities
 - Enroll, with composite PK Stu_ID & Class_code.
 - Class with Class_code as it's PK.



CS275 Fall 2010

32

Fourth Normal Form (4NF)

- Table is in fourth normal form (4NF) when both of the following are true:
 - It is in 3NF
 - No multiple sets of multivalued dependencies
- 4NF is largely academic if tables conform to following two rules:
 - All attributes dependent on primary key, independent of each other
 - No row contains two or more multivalued facts about an entity

CS275 Fall 2010

33

Fourth Normal Form (4NF)

- Two Examples of multi-valued dependencies
 - StudentID,StName,Phones(Home,Work,Cell,Fax)
 - StudentID,Addresses(permanent, mailing, current)
- Convert multi-valued phones using two additional tables in 3NF
 - Student(StudentID, StName,.....)
 - StuPhones(StudentID, PhoneType, Phone#)
 - Phones(PhoneType, Description)

CS275 Fall 2010

34

Fourth Normal Form (4NF)

- Example: Tracking employee's volunteer service

FIGURE 6.10 Tables with multivalued dependencies

Database name: CHM_Service

Table name: VOLUNTEER_X1			
EMP_NUM	ORG_CODE	PROJ_CODE	HOURS
1012	01	01	1
1012	01	02	1
1012	01	03	1
1012	01	04	1
1012	01	05	1

Table name: VOLUNTEER_X2			
EMP_NUM	ORG_CODE	PROJ_CODE	HOURS
1012	01	01	1
1012	01	02	1
1012	01	03	1
1012	01	04	1
1012	01	05	1

Table name: VOLUNTEER_X3			
EMP_NUM	ORG_CODE	PROJ_CODE	HOURS
1012	01	01	1
1012	01	02	1
1012	01	03	1
1012	01	04	1
1012	01	05	1

FIGURE 6.11 A set of tables in 4NF

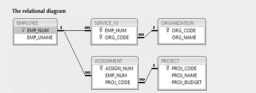
Database name: CHM_Service

Table name: PROJECT			
PROJ_CODE	PROJ_NAME	PROJ_BUDGET	PROJ_MANAGER
01	Research	100,000.00	1012
02	Software	200,000.00	1012
03	Hardware	200,000.00	1012
04	Software	50,000.00	1012
05	Software	50,000.00	1012

Table name: EMPLOYEE			
EMP_NUM	EMP_NAME	EMP_PHONE	EMP_ADDRESS
1012	John Doe	123-456-7890	123 Main St, Anytown, CA 94024

Table name: ORGANIZATION			
ORG_CODE	ORG_NAME	ORG_ADDRESS	ORG_PHONE
01	United Way	123 Main St, Anytown, CA 94024	123-456-7890

Table name: SERVICE_X1			
EMP_NUM	ORG_CODE	PROJ_CODE	HOURS
1012	01	01	1
1012	01	02	1
1012	01	03	1
1012	01	04	1
1012	01	05	1



CS275 Fall 2010

35

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
 - Causing additional processing
 - Loss of system speed

CS275 Fall 2010

36

Denormalization

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

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

Data-Modeling Checklist

- Data modeling translates specific real-world environment into a data model
- Data-modeling checklist helps ensure that data-modeling tasks are successfully performed

TABLE 6.7 Data-Modeling Checklist	
BUSINESS RULES	<ul style="list-style-type: none"> • Properly document and verify all business rules with the end users. • Ensure that all business rules are written precisely, clearly, and unambiguously. 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.
DATA MODELING	<ul style="list-style-type: none"> • Naming Conventions: All names should be limited in length (database-dependent size). • Entity Names: <ul style="list-style-type: none"> • 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, use hyphens as a combination of abbreviated names of the entities linked through the composite entity • Method Names: <ul style="list-style-type: none"> • Should be unique within the entity • Should use the entity abbreviation as a prefix • Should be unique for the entire model • Should use suffixes such as _ID, _NAME, or _CODE for the PK attribute • Should not be reserved words • Relationship Names: <ul style="list-style-type: none"> • Should be active or passive verbs that clearly indicate the nature of the relationship
Entities	<ul style="list-style-type: none"> • Each entity should represent a single object. • Each entity should represent a set of distinguishable entity instances. • All entities should be 1NF or higher; any entities below 1NF 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	<ul style="list-style-type: none"> • Should be single and single-valued (atomic data) • Should document default values, constraints, synonyms, and aliases • Default attributes should be clearly identified and include occurrence • Should not be redundant unless this is required for transaction accuracy, performance, or maintaining a history • The PK should be clearly defined and support the selected data granularity
Relationships	<ul style="list-style-type: none"> • Should clearly identify relationship participants • Should clearly define relationship participation, connectivity, and document cardinality
ER models	<ul style="list-style-type: none"> • Should be validated against expected processes, inserts, updates, and deletes • Should maintain referential integrity and have to maintain a history • Should not contain redundant relationships that are required for data verification • Should minimize data redundancy to ensure single point updates • Should continue to the normal data size. "If that is needed it there, and if that is there it is needed."

Normalization and Database Design

- 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

Summary

- Normalization is a technique used to minimize data redundancies
- Normalization is an important part of the design process
- Whereas ERD's provide a macro view, normalization provides micro view of entities
 - Focuses on characteristics of specific entities
 - May yield additional entities
- Difficult to separate normalization from E-R diagramming – do both techniques concurrently.

CS275 Fall 2010

41

Summary

- First three normal forms (1NF, 2NF, and 3NF) are most commonly encountered
- Table is in 1NF when:
 - All key attributes are defined
 - All remaining attributes are dependent on primary key
- 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

CS275 Fall 2010

42

Summary

- Table that is not in 3NF may be split into new tables until all of the tables meet 3NF requirements
- Table in 3NF may contain multivalued dependencies
 - Numerous null values or redundant data
- Convert 3NF table to 4NF by:
 - Splitting table to remove multivalued dependencies
- Tables are sometimes denormalized to yield less I/O, which increases processing speed

CS275 Fall 2010

43

CS275 Fall 2010

44

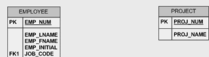
Improving the Design

- Contracting Company Example

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.

Improving the Design

- Contracting Company Example

FIGURE 6.14 Incorrect MN relationship representation

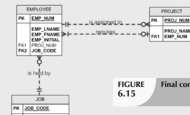
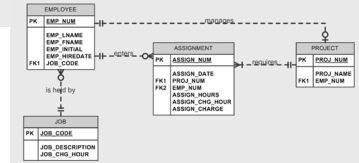


FIGURE 6.15 Final contracting company ERD



Improving the Design

- Contracting Company Example

FIGURE 6.16 The implemented database

Database name: CWB_ContractCo

Table name: EMPLOYEE						
EMP_NUM	EMP_LNAME	EMP_FNAME	EMP_INITIAL	PROJ_NUM	JOB_CODE	JOB_CHG_HOUR
101	Smith	John		10	10	10
102	Smith	John		10	10	10
103	Johnson	John		10	10	10
104	Johnson	John		10	10	10
105	Johnson	John		10	10	10
106	Johnson	John		10	10	10
107	Johnson	John		10	10	10
108	Johnson	John		10	10	10
109	Johnson	John		10	10	10
110	Johnson	John		10	10	10
111	Johnson	John		10	10	10
112	Johnson	John		10	10	10
113	Johnson	John		10	10	10
114	Johnson	John		10	10	10
115	Johnson	John		10	10	10
116	Johnson	John		10	10	10
117	Johnson	John		10	10	10
118	Johnson	John		10	10	10
119	Johnson	John		10	10	10
120	Johnson	John		10	10	10
121	Johnson	John		10	10	10
122	Johnson	John		10	10	10
123	Johnson	John		10	10	10
124	Johnson	John		10	10	10
125	Johnson	John		10	10	10
126	Johnson	John		10	10	10
127	Johnson	John		10	10	10
128	Johnson	John		10	10	10
129	Johnson	John		10	10	10
130	Johnson	John		10	10	10
131	Johnson	John		10	10	10
132	Johnson	John		10	10	10
133	Johnson	John		10	10	10
134	Johnson	John		10	10	10
135	Johnson	John		10	10	10
136	Johnson	John		10	10	10
137	Johnson	John		10	10	10
138	Johnson	John		10	10	10
139	Johnson	John		10	10	10
140	Johnson	John		10	10	10
141	Johnson	John		10	10	10
142	Johnson	John		10	10	10
143	Johnson	John		10	10	10
144	Johnson	John		10	10	10
145	Johnson	John		10	10	10
146	Johnson	John		10	10	10
147	Johnson	John		10	10	10
148	Johnson	John		10	10	10
149	Johnson	John		10	10	10
150	Johnson	John		10	10	10
151	Johnson	John		10	10	10
152	Johnson	John		10	10	10
153	Johnson	John		10	10	10
154	Johnson	John		10	10	10
155	Johnson	John		10	10	10
156	Johnson	John		10	10	10
157	Johnson	John		10	10	10
158	Johnson	John		10	10	10
159	Johnson	John		10	10	10
160	Johnson	John		10	10	10
161	Johnson	John		10	10	10
162	Johnson	John		10	10	10
163	Johnson	John		10	10	10
164	Johnson	John		10	10	10
165	Johnson	John		10	10	10
166	Johnson	John		10	10	10
167	Johnson	John		10	10	10
168	Johnson	John		10	10	10
169	Johnson	John		10	10	10
170	Johnson	John		10	10	10
171	Johnson	John		10	10	10
172	Johnson	John		10	10	10
173	Johnson	John		10	10	10
174	Johnson	John		10	10	10
175	Johnson	John		10	10	10
176	Johnson	John		10	10	10
177	Johnson	John		10	10	10
178	Johnson	John		10	10	10
179	Johnson	John		10	10	10
180	Johnson	John		10	10	10
181	Johnson	John		10	10	10
182	Johnson	John		10	10	10
183	Johnson	John		10	10	10
184	Johnson	John		10	10	10
185	Johnson	John		10	10	10
186	Johnson	John		10	10	10
187	Johnson	John		10	10	10
188	Johnson	John		10	10	10
189	Johnson	John		10	10	10
190	Johnson	John		10	10	10
191	Johnson	John		10	10	10
192	Johnson	John		10	10	10
193	Johnson	John		10	10	10
194	Johnson	John		10	10	10
195	Johnson	John		10	10	10
196	Johnson	John		10	10	10
197	Johnson	John		10	10	10
198	Johnson	John		10	10	10
199	Johnson	John		10	10	10
200	Johnson	John		10	10	10

Table name: JOB						
JOB_CODE	JOB_DESCRIPTION	JOB_CHG_HOUR				
10	Engineer	20				
20	Systems Analyst	20				
30	Systems Programmer	20				
40	Systems Operator	20				
50	Database Operator	20				
60	Database Administrator	20				
70	Systems Engineer	20				
80	Systems Specialist	20				
90	Systems Analyst	20				
100	Systems Programmer	20				
110	Systems Operator	20				
120	Database Operator	20				
130	Database Administrator	20				
140	Systems Engineer	20				
150	Systems Specialist	20				
160	Systems Analyst	20				
170	Systems Programmer	20				
180	Systems Operator	20				
190	Database Operator	20				
200	Database Administrator	20				

Table name: PROJECT						
PROJ_NUM	PROJ_NAME					
10	Project A					
20	Project B					
30	Project C					
40	Project D					
50	Project E					
60	Project F					
70	Project G					
80	Project H					
90	Project I					
100	Project J					
110	Project K					
120	Project L					
130	Project M					
140	Project N					
150	Project O					
160	Project P					
170	Project Q					
180	Project R					
190	Project S					
200	Project T					

Table name: ASSIGNMENT						
ASSIGN_NUM	ASSIGN_DATE	PROJ_NUM	EMP_NUM	ASSIGN_HOURS	ASSIGN_CHANGE	
1001	2010-01-01	10	101	10	10	
1002	2010-01-01	10	102	10	10	
1003	2010-01-01	10	103	10	10	
1004	2010-01-01	10	104	10	10	
1005	2010-01-01	10	105	10	10	
1006	2010-01-01	10	106	10	10	
1007	2010-01-01	10	107	10	10	
1008	2010-01-01	10	108	10	10	
1009	2010-01-01	10	109	10	10	
1010	2010-01-01	10	110	10	10	
1011	2010-01-01	10	111	10	10	
1012	2010-01-01	10	112	10	10	
1013	2010-01-01	10	113	10	10	
1014	2010-01-01	10	114	10	10	
1015	2010-01-01	10	115	10	10	
1016	2010-01-01	10	116	10	10	
1017	2010-01-01	10	117	10	10	
1018	2010-01-01	10	118	10	10	
1019	2010-01-01	10	119	10	10	
1020	2010-01-01	10	120	10	10	
1021	2010-01-01	10	121	10	10	
1022	2010-01-01	10	122	10	10	
1023	2010-01-01	10	123	10	10	
1024	2010-01-01	10	124	10	10	
1025	2010-01-01	10	125	10	10	
1026	2010-01-01	10	126	10	10	
1027	2010-01-01	10	127	10	10	
1028	2010-01-01	10	128	10	10	
1029	2010-01-01	10	129	10	10	
1030	2010-01-01	10	130	10	10	
1031	2010-01-01	10	131	10	10	
1032	2010-01-01	10	132	10	10	
1033	2010-01-01	10	133	10	10	
1034	2010-01-01	10	134	10	10	
1035	2010-01-01	10	135	10	10	
1036	2010-01-01	10	136	10	10	
1037	2010-01-01	10	137	10	10	
1038	2010-01-01	10	138	10	10	
1039	2010-01-01	10	139	10	10	
1040	2010-01-01	10	140	10	10	
1041	2010-01-01	10	141	10	10	
1042	2010-01-01	10	142	10	10	
1043	2010-01-01	10	143	10	10	
1044	2010-01-01	10	144	10	10	
1045	2010-01-01	10	145	10	10	
1046	2010-01-01	10	146	10	10	
1047	2010-01-01	10	147	10	10	
1048	2010-01-01	10	148	10	10	
1049	2010-01-01	10	149	10	10	
1050	2010-01-01	10	150	10	10	
1051	2010-01-01	10	151	10	10	
1052	2010-01-01	10	152	10	10	
1053	2010-01-01	10	153	10	10	
1054	2010-01-01	10	154	10	10	
1055	2010-01-01	10	155	10	10	
1056	2010-01-01	10	156	10	10	
1057	2010-01-01	10	157	10	10	
1058	2010-01-01	10	158	10	10	
1059	2010-01-01	10	159	10	10	
1060	2010-01-01	10	160	10	10	
1061	2010-01-01	10	161	10	10	
1062	2010-01-01	10	162	10	10	
1063	2010-01-01	10	163	10	10	
1064	2010-01-01	10	164	10	10	
1065	2010-01-01	10	165	10	10	
1066	2010-01-01	10	166	10	10	
1067	2010-01-01	10	167	10	10	
1068	2010-01-01	10	168	10	10	
1069	2010-01-01	10	169	10	10	
1070	2010-01-01	10	170	10	10	
1071	2010-01-01	10	171	10	10	
1072	2010-01-01	10	172	10	10	
1073	2010-01-01	10	173	10	10	
1074	2010-01-01	10	174	10	10	
1075	2010-01-01	10	175	10	10	
1076	2010-01-01	10	176	10	10	
1077	2010-01-01	10	177	10	10	
1078	2010-01-01	10	178	10	10	
1079	2010-01-01	10	179	10	10	
1080	2010-01-01	10	180	10	10	
1081	2010-01-01	10	181	10	10	
1082	2010-01-01	10	182	10	10	
1083	2010-01-01	10	183	10	10	