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

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

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

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

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 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

- 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	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

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

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

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

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

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

Step 1 - Eliminate the Repeating Groups

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

Step 2 - Conversion to 1NF

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

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*.

  - 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

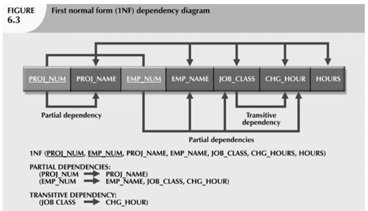
Step 3- Conversion to 1NF

- 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

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

FIGURE 6.3 First normal form (1NF) dependency diagram



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.

Conversion to Second Normal Form

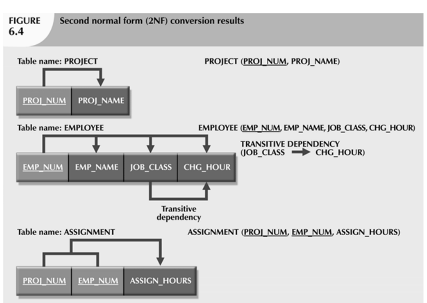
- Step 1: Eliminate Partial Dependencies
  - Start with 1NF format and convert by:
    - Write each part of the composite key on it's own line.
    - Write the original (composite) key on last line
  - Each component will become key in new table
- Step 2: Assign Dependent Attributes
  - From the original 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)

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

Completed Conversion to 2NF



Conversion to Third Normal Form

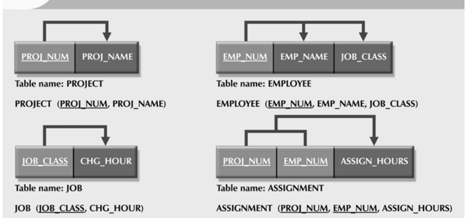
- Step 1: Eliminate Transitive Dependencies
  - Write its determinant as PK for new table.
  - And Leave it in the Original Table
- Step 2: Reassign Corresponding Dependent Attributes
  - Identify attributes dependent on each determinant identified in Step 1, and list on new table.
- Step 3: Name the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

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

FIGURE 6.5 Third normal form (3NF) conversion results



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.

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

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

Improving the Design

- Identifying new 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

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)

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, sectionno</u>, courseno, time, days

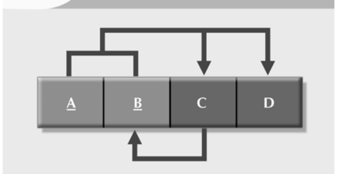
The Boyce-Codd Normal Form (BCNF)

- 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

The Boyce-Codd Normal Form (BCNF)

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

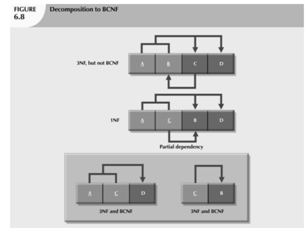
FIGURE 6.7 A table that is in 3NF but not in BCNF



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.

FIGURE 6.8 Decomposition to BCNF



The Boyce-Codd Normal Form (BCNF)

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

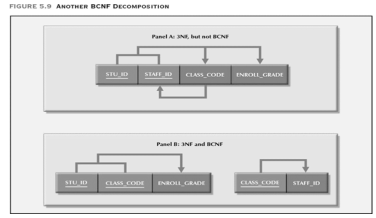
TABLE 6.5 Sample Data for a BCNF Conversion

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

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.

FIGURE 5.9 Another BCNF Decomposition



Fourth Normal Form (4NF)

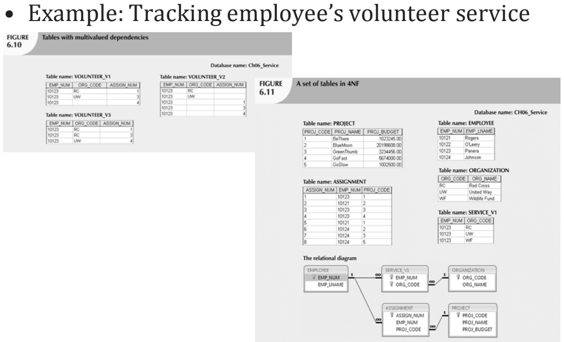
- Table is in fourth normal form (4NF) when both of the following are true:
  - It is in 3NF
  - No multiple sets of multivalued dependencies
- 4NF is largely academic if tables conform to following two rules:
  - All attributes dependent on primary key, independent of each other
  - No row contains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

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)

Fourth Normal Form (4NF)

- Example: Tracking employee's volunteer service



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

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

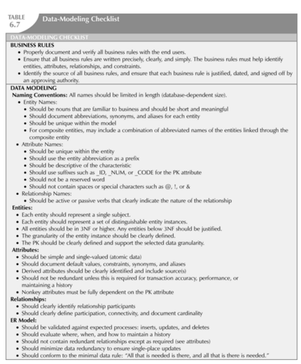


TABLE 6.7 Data-Modeling Checklist

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.

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

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

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 deletes
- 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."

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.

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

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

Improving the Design

- Contracting Company Example

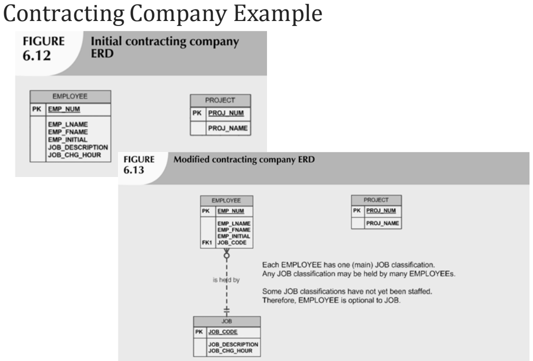


FIGURE 6.12 Initial contracting company ERD

FIGURE 6.13 Modified contracting company ERD

Improving the Design

- Contracting Company Example

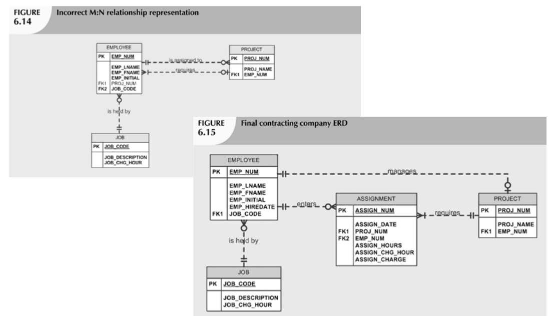


FIGURE 6.14 Incorrect M:N relationship representation

FIGURE 6.15 Final contracting company ERD

Improving the Design

- Contracting Company Example

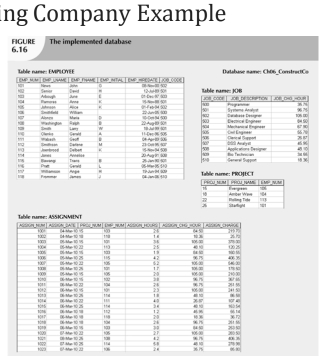


FIGURE 6.16 The implemented database