Objectives

- In this chapter, you will learn:
  - What normalization is and what role it plays in the database design process
  - About the normal forms 1NF, 2NF, 3NF, BCNF
  - 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

5.1 Database Tables and Normalization

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

- 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
  - Price paid for increased performance is greater data redundancy
5.2 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 5.1

Table 5.1 A Sample Report Layout

<table>
<thead>
<tr>
<th>EMPLOYEE NAME</th>
<th>PROJECT NAME</th>
<th>BILLING RATE</th>
<th>CHARGED HOURS</th>
<th>TOTAL CHARGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>John G. News</td>
<td>Evergreen</td>
<td>$15.00</td>
<td>12.4</td>
<td>$184.80</td>
</tr>
<tr>
<td>Alice K. Johnson</td>
<td>Amber Wave</td>
<td>$15.00</td>
<td>12.4</td>
<td>$184.80</td>
</tr>
<tr>
<td>William Smithfield</td>
<td></td>
<td>$30.00</td>
<td>12.4</td>
<td>$374.40</td>
</tr>
<tr>
<td>David H. Senior</td>
<td></td>
<td>$45.00</td>
<td>12.4</td>
<td>$531.60</td>
</tr>
<tr>
<td>Tom E. Smith</td>
<td>Rolling Tide</td>
<td>$15.00</td>
<td>12.4</td>
<td>$184.80</td>
</tr>
<tr>
<td>Alice K. Johnson</td>
<td></td>
<td>$15.00</td>
<td>12.4</td>
<td>$184.80</td>
</tr>
<tr>
<td>William Smithfield</td>
<td></td>
<td>$30.00</td>
<td>12.4</td>
<td>$374.40</td>
</tr>
<tr>
<td>Darlene M. Smithson</td>
<td></td>
<td>$45.00</td>
<td>12.4</td>
<td>$531.60</td>
</tr>
</tbody>
</table>

Subtotal: $971.40

Total: $49,195.69

Table 5.1 A Sample Report Layout

The Need for Normalization (continued)

- Structure of data set in Figure 5.1 does not handle data very well
- Table structure appears to work; report generated with ease
- Report may yield (產生) different results depending on what data anomaly has occurred
- Relational database environment suited to help designer avoid data integrity (完整性) problems
- Check p. 155
5.3 The Normalization Process

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

The Normalization Process (continued)

- Objective of normalization is to ensure all tables in at least 3NF
- Higher forms 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 5.2 Normal Forms

<table>
<thead>
<tr>
<th>NORMAL FORM</th>
<th>CHARACTERISTIC</th>
<th>SECTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>First normal form (1NF)</td>
<td>Table format, no repeating groups, and PK identified</td>
<td>5.3.1</td>
</tr>
<tr>
<td>Second normal form (2NF)</td>
<td>In 1NF and no partial dependencies</td>
<td>5.3.2</td>
</tr>
<tr>
<td>Third normal form (3NF)</td>
<td>In 2NF and no transitive dependencies</td>
<td>5.3.3</td>
</tr>
<tr>
<td>Boyce-Codd normal form (BCNF)</td>
<td>Every determinant is a candidate key (special case of 3NF)</td>
<td>5.6.1</td>
</tr>
<tr>
<td>Fourth normal form (4NF)</td>
<td>In 3NF and no independent multivalued dependencies</td>
<td>5.6.2</td>
</tr>
</tbody>
</table>

### Table 5.3 Functional Dependency Concepts

<table>
<thead>
<tr>
<th>CONCEPT</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functional dependency</td>
<td>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.</td>
</tr>
<tr>
<td>Functional dependency (generalized definition)</td>
<td>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.</td>
</tr>
<tr>
<td>Fully functional dependency (composite key)</td>
<td>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.</td>
</tr>
</tbody>
</table>
Conversion to First Normal Form

• Repeating group (重覆群組)
  – Group of multiple entries of same type 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 (continued)

• 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 (相依性，功能相依性)
  – Dependencies depicted with a diagram

Figure 5.2 A table in First Normal Form
Conversion to First Normal Form (continued)

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

Conversion to Second Normal Form

- Step 1: Write Each Key Component on a Separate Line
  - Write each key component on separate line, then write original (composite) key on last line
  - Each component will become key in new table
- Step 2: Assign Corresponding Dependent Attributes
  - Determine those attributes that are dependent on other attributes
  - At this point, most anomalies have been eliminated
Conversion to Second Normal Form (continued)

- 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: Identify Each New Determinant
  - For every transitive dependency, write its determinant as PK for new table
  - Determinant (決定因素，決定屬性): any attribute whose value determines other values within a row

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

- Step 3: Remove the Dependent Attributes from Transitive Dependencies (遞移相依性)
  - Eliminate all dependent attributes in transitive relationship(s) from each of the tables
  - Draw new dependency diagram to show all tables defined in Steps 1–3
  - Check new tables as well as tables modified in Step 3
    - Each table has determinant
    - No table contains inappropriate dependencies

FIGURE 5.5  Third normal form (3NF) conversion results
Conversion to Third Normal Form (continued)

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

5.4 Improving the Design

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

Improving the Design (continued)

• Issues to address in order to produce a good normalized set of tables:
  – Evaluate PK Assignments
  – Evaluate Naming Conventions
  – Refine Attribute Atomicity (單元性)
  – Identify New Attributes
  – Identify New Relationships
  – Refine Primary Keys as Required for Data Granularity (粒度，內容的詳細程度)
  – Maintain Historical Accuracy
  – Evaluate Using Derived Attributes
5.5 Surrogate Key Considerations

• When primary key is considered to be unsuitable, designers use surrogate keys (代理鍵)

• Data entries in Table 5.4 are inappropriate because they duplicate existing records
  – No violation of entity or referential integrity

5.6 Higher-Level Normal Forms

• Tables in 3NF perform suitably in business transactional databases

• Higher order normal forms useful on occasion

• Two special cases of 3NF:
  – Boyce-Codd normal form (BCNF)
  – Fourth normal form (4NF) [跳過 5.6.2]
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

**The Boyce-Codd Normal Form (BCNF) (continued)**

- Most designers consider the BCNF as 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 non-key attribute is the determinant of a key attribute
5.7 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

• 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

Example about Project Management

• Business rules
  – The company manages many projects
  – Each project requires the services of many employees
  – An employee may be assigned to several different projects
  – Some employees are not assigned to a project and perform duties not specifically related to a project. Some employees are part of a labor pool, to be shared by all project teams
  – Each employee has a single primary job classification, which determines the hourly billing rate
  – Many employees can have the same job classification.
FIGURE 5.13  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 5.14  Incorrect M:N relationship representation

FIGURE 5.15  Final contracting company ERD

FIGURE 5.16  The implemented database
5.7 Denormalization

- Creation of normalized relations is important database design goal
- Processing requirements should also be a goal
- If tables decomposed to conform to normalization requirements:
  - Number of database tables expands
  - Example: ZIP(ZIP_CODE, CITY)

Denormalization (continued)

- Common Denormalization Examples
  - Redundant data
  - Derived data
  - Pre-aggregated data
  - Information requirement
    - Temporary denormalized table for report data
      - Example: faculty evaluation report
      - Example: data warehouse

Denormalization (continued)

- Joining the larger number of tables reduces system speed
- Conflicts 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.
Summary

• Normalization is used to minimize data redundancies
• 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

Summary (continued)

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

Summary (continued)

• Table in 3NF may contain multivalued dependencies
  – Numerous null values or redundant data
• Tables are sometimes denormalized to yield less I/O, which increases processing speed