“Modern organizations are said to be drowning in data but starving for information”

p. 509
Learning Objectives

1. Describe why databases have become so important to modern organizations.
2. Describe what databases and database management systems are and how they work.
Learning Objectives

1. Describe why databases have become so important to modern organizations.
2. Describe what databases and database management systems are and how they work.
Database Management for Strategic Advantage

- Database technology is vital to an organization’s success
- Variety of information collected and stored
  - Stock prices
  - Potential customers
  - Credit ratings of wholesalers
  - Etc.
Database Management for Strategic Advantage (II)

- Database technology used for:
  - Gathering and storing customer information
  - Custom-tailoring catalogs and mailings
  - Fueling electronic commerce
  - Etc.
Learning Objectives

1. Describe why databases have become so important to modern organizations.
2. Describe what databases and database management systems are and how they work.
Database Foundations

• Then:
  o Card catalogs
  o File cabinets

• Now:
  o DBMS
    • Create
    • Store
    • Organize
    • Retrieve data
  o E.g., Microsoft Access
Database

- Database
  - Collection of related data organized in a way to facilitate searches
- Entities
  - Something you collect data about
  - E.g.: people, books
Record

- Contains data about a single entity
  - Similar to catalog card
Attributes

- Specific characteristic describing the entities
  - E.g.: name and social security number are attributes of a person
Example: Entity Student

• Entity represented as a table, with rows as records and columns as attributes

<table>
<thead>
<tr>
<th>ID Number</th>
<th>Last Name</th>
<th>First Name</th>
<th>Street Address</th>
<th>City</th>
<th>State</th>
<th>Zip code</th>
<th>Major</th>
</tr>
</thead>
<tbody>
<tr>
<td>209345</td>
<td>Vance</td>
<td>James</td>
<td>1242 N. Maple</td>
<td>Bloomington</td>
<td>Indiana</td>
<td>47401</td>
<td>Recreation</td>
</tr>
<tr>
<td>213009</td>
<td>Haggarty</td>
<td>Joe</td>
<td>3400 E. Longvi</td>
<td>Bloomington</td>
<td>Indiana</td>
<td>47405</td>
<td>Business Management</td>
</tr>
<tr>
<td>345987</td>
<td>Borden</td>
<td>Chris</td>
<td>367 Ridge Rd</td>
<td>Bloomington</td>
<td>Indiana</td>
<td>47405</td>
<td>Aeronautical Engineering</td>
</tr>
<tr>
<td>457838</td>
<td>Jessup</td>
<td>Mike</td>
<td>12 Long Lake</td>
<td>Bloomington</td>
<td>Indiana</td>
<td>47401</td>
<td>Computer Science</td>
</tr>
<tr>
<td>459987</td>
<td>Chan</td>
<td>Virginia</td>
<td>8009 Walnut</td>
<td>Bloomington</td>
<td>Indiana</td>
<td>47405</td>
<td>Sociology</td>
</tr>
<tr>
<td>466711</td>
<td>Monroe</td>
<td>Lisa</td>
<td>234 Jamie Ln</td>
<td>Bloomington</td>
<td>Indiana</td>
<td>47401</td>
<td>Pre-Medicine</td>
</tr>
<tr>
<td>512678</td>
<td>Austin</td>
<td>John</td>
<td>3837 Wood's E</td>
<td>Bloomington</td>
<td>Indiana</td>
<td>47401</td>
<td>Law</td>
</tr>
<tr>
<td>691112</td>
<td>Sherwin</td>
<td>Jordan</td>
<td>988 Woodbridg</td>
<td>Bloomington</td>
<td>Indiana</td>
<td>47404</td>
<td>Political Science</td>
</tr>
<tr>
<td>910234</td>
<td>Moore</td>
<td>Larry</td>
<td>1234 S. Grant</td>
<td>Bloomington</td>
<td>Indiana</td>
<td>47403</td>
<td>Civil Engineering</td>
</tr>
<tr>
<td>979776</td>
<td>Dunn</td>
<td>Pat</td>
<td>109 Hoosier Av</td>
<td>Bloomington</td>
<td>Indiana</td>
<td>47404</td>
<td>Psychology</td>
</tr>
<tr>
<td>983445</td>
<td>Pickett</td>
<td>Steve</td>
<td>999 College</td>
<td>Bloomington</td>
<td>Indiana</td>
<td>47401</td>
<td>Sports Science</td>
</tr>
</tbody>
</table>
Advantages of the Database Approach

1. Program-data independence
2. Minimal data redundancy
3. Improved data consistency
4. Improved data sharing
5. Increased productivity of application development
Advantages of the Database Approach (II)

6. Enforcement of standards
7. Improved data quality
8. Improved data accessibility
9. Reduced program maintenance
Cost and Risks of the Database Approach

1. New, specialized personnel
2. Installation and management cost and complexity
3. Conversion costs
4. Need for explicit backup and recovery
5. Organizational conflict
Effective Management of Databases

• Database Administrator (DBA)
  o Responsible for development and management of databases
    • Works with system analysts and programmers
    • Works with users and managers
    • Implements security features
    • Grants access rights
  o One of the key actors in creating a successful database
Entering Data

• Forms
  ○ Enter data about a record
  ○ Field in a form corresponds to attribute in a record
  ○ Used to add, modify, or delete data
Example: Computer-Based Form
Querying Data

- Query: used to retrieve information
- Structured Query Language (SQL)
  - Example: Display students who earned an “A”
    
    ```sql
    SELECT DISTINCTROW STUDENT_ID, GRADE
    FROM GRADES
    WHERE GRADE="A"
    ORDER BY STUDENT_ID;
    ```
  
  - Writing SQL queries can become very complex
Query by Example

- Simpler than SQL
- Drag-and-drop features
- Construct a sample of the data we would like to see
Creating Database Reports

• Report
  o Compilation of data from the database

• Report generators
  o Retrieve, manipulate, and display data

• Example
  o Quarterly sales for a restaurant
    • Adding
    • Grouping
Database Design

• Data need to be organized for retrieval and analysis
  o Key elements of a database
    • Data
    • Structure

• Data model
  o A map or a diagram representing entities and their relationships
Example: Entities and Their Attributes

<table>
<thead>
<tr>
<th>Students</th>
<th>Grades</th>
</tr>
</thead>
<tbody>
<tr>
<td>Student ID</td>
<td>Student ID</td>
</tr>
<tr>
<td>Name</td>
<td>Course ID</td>
</tr>
<tr>
<td>Campus Address</td>
<td>Section No.</td>
</tr>
<tr>
<td>Major</td>
<td>Term</td>
</tr>
<tr>
<td>Phone</td>
<td>Grade</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Students</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Student ID</td>
<td>Name</td>
</tr>
<tr>
<td>555-39-3232</td>
<td>Joe Jones</td>
</tr>
<tr>
<td>289-42-8776</td>
<td>Sally Carter</td>
</tr>
<tr>
<td>Campus Address</td>
<td>Campus Address</td>
</tr>
<tr>
<td>123 Any Avenue</td>
<td>1200 Wolf Street #12</td>
</tr>
<tr>
<td>Major</td>
<td>Major</td>
</tr>
<tr>
<td>Finance</td>
<td>Marketing</td>
</tr>
<tr>
<td>Phone</td>
<td>Phone</td>
</tr>
<tr>
<td>335-2211</td>
<td>335-8702</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Grades</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Student ID</td>
<td>Course ID</td>
</tr>
<tr>
<td>555-39-3232</td>
<td>MIS 250</td>
</tr>
<tr>
<td>555-39-3232</td>
<td>MIS 250</td>
</tr>
<tr>
<td>289-42-8776</td>
<td>MIS 250</td>
</tr>
<tr>
<td>Section No.</td>
<td>Section No.</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Term</td>
<td>Term</td>
</tr>
<tr>
<td>F’05</td>
<td>F’06</td>
</tr>
<tr>
<td>S’07</td>
<td></td>
</tr>
<tr>
<td>Grade</td>
<td>Grade</td>
</tr>
<tr>
<td>D+</td>
<td>A−</td>
</tr>
<tr>
<td></td>
<td>B+</td>
</tr>
</tbody>
</table>
Identifying Records

• Uniquely identifying records:
  o Primary Key
    • Unique identifier
    • E.g.: Student ID number, social security number
  o Combination primary key
    • Combination of two or more attributes
    • E.g.: identifying a student’s grade for a particular class for a particular term

• Identifying records that share a common value
  o Secondary Key
    • Attribute not used as a primary key
    • E.g., Major
## Associations

- Used to relate information between tables
- Needed to retrieve information
- Example: Basketball league database

<table>
<thead>
<tr>
<th>Relationship</th>
<th>Examples</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>One-to-one</td>
<td>Each team has only one home stadium, and each home stadium has only one team.</td>
<td>Place the primary key from one table into the other as a foreign key.</td>
</tr>
<tr>
<td>One-to-many</td>
<td>Each player is on only one team, but each team has many players.</td>
<td>Place the primary key from the table on the one side of the relationship as a foreign key in the table on the many side of the relationship.</td>
</tr>
<tr>
<td>Many-to-many</td>
<td>Each player participates in many games and each game has many players.</td>
<td>Create a third table and place the primary keys from each of the original tables together in the third as a combination primary key.</td>
</tr>
</tbody>
</table>
Example: Basketball League

- Each table contains important data
- No way to learn which team plays in a specific stadium
- Need to make associations

<table>
<thead>
<tr>
<th>Home Stadium</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stadium ID</td>
</tr>
<tr>
<td>Team</td>
</tr>
<tr>
<td>Team ID</td>
</tr>
<tr>
<td>Player</td>
</tr>
<tr>
<td>Player ID</td>
</tr>
<tr>
<td>Games</td>
</tr>
<tr>
<td>Team ID (1)</td>
</tr>
</tbody>
</table>
Example: Basketball League

- **Foreign keys**
  - Attributes used to link tables
  - Primary key in one table, foreign key in another
  - Need to create additional entity for many-to-many relationships

A. One-to-one relationship: Each team has only one home stadium, and each home stadium has only one team.

```
<table>
<thead>
<tr>
<th>Team ID</th>
<th>Team Name</th>
<th>Stadium ID</th>
</tr>
</thead>
</table>
```

B. One-to-many relationship: Each player is on only one team, but each team has many players.

```
<table>
<thead>
<tr>
<th>Player ID</th>
<th>Player Name</th>
<th>Position</th>
<th>Team ID</th>
</tr>
</thead>
</table>
```

C. Many-to-many relationship: Each player participates in many games, and each game has many players.

```
<table>
<thead>
<tr>
<th>Team 1</th>
<th>Team 2</th>
<th>Date</th>
<th>Player ID</th>
<th>Points</th>
<th>Minutes</th>
<th>Fouls</th>
</tr>
</thead>
</table>
```
Entity-Relationship Diagram

• Used to show associations between entities
• Important when designing complex databases
  - Entities: represented by boxes
  - Relationships: represented by lines

Associations
- Each Home Stadium has a Team.
- Each Team has Players.
- Each Team participates in Games.
- For each Player and Game, there are Game Statistics.
The Relational Model

- Primary DBMS approach (RDBMS)
- 3 dimensions
  - Entities represented as 2-dimensional tables
    - Rows = records
    - Columns = attributes
  - Tables joined based on common columns (3rd dimension)
- Good design eliminates redundancy
Data Redundancy

• Problematic if an attribute has to be changed
  - Need to change in multiple locations
    - E.g., instructor’s phone number
Normalization

- Eliminate unnecessary redundancy
  - Create separate tables
  - Data only needs to be changed in a single location
Data Dictionary

- Document that specifies what data needs to be entered
  - Attribute name
  - Key or not
  - Data type
  - Valid values

- Can be used to enforce business rules
Linking Website Applications to Organizational Databases

• Users can access a variety of data via a company’s web site
  - Web services help in integration of databases, regardless of physical location
• Need for adequate systems performance
• Get understanding of customer behavior