Chapter 7

Database Design

Database Life Cycle (DBLC)
In this chapter, you will learn:

• That successful database design must reflect the information system of which the database is a part

• That successful information systems are developed within a framework known as the Systems Development Life Cycle (SDLC)
In this chapter, you will learn (continued):

- That within the information system, the most successful databases are subject to frequent evaluation and revision within a framework known as the Database Life Cycle (DBLC)
- How to conduct evaluation and revision within the SDLC and DBLC frameworks
- About database design strategies: top-down vs. bottom-up design and centralized vs. decentralized design
The Information System

• Provides for data collection, storage, and retrieval

• Composed of people, hardware, software, database(s), application programs, and procedures

• Systems analysis
  – Process that establishes need for and extent of information system

• Systems development
  – Process of creating information system
The Information System (continued)

• Applications
  – Transform data into information that forms basis for decision making
  – Usually produce the following:
    • Formal report
    • Tabulations
    • Graphic displays
  – Composed of following two parts:
    • Data
    • Code by which data are transformed into information
The Information System (continued)
The Information System (continued)

• Information system performance depends on triad of factors:
  – Database design and implementation
  – Application design and implementation
  – Administrative procedures

• Database development
  – Process of database design and implementation
  – Primary objective is to create complete, normalized, nonredundant (to the extent possible), and fully integrated conceptual, logical, and physical database models
The Systems Development Life Cycle (SDLC)

- Traces history (life cycle) of information system
- Provides “big picture” within which database design and application development can be mapped out and evaluated
The Systems Development Life Cycle (SDLC) (continued)

• Divided into following five phases:
  – Planning
  – Analysis
  – Detailed systems design
  – Implementation
  – Maintenance

• Iterative rather than sequential process
The Systems Development Life Cycle (SDLC) (continued)

**FIGURE 9.2**

**The Systems Development Life Cycle (SDLC)**

<table>
<thead>
<tr>
<th>Phase</th>
<th>Action(s)</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planning</td>
<td>Initial assessment, Feasibility study</td>
<td>9.2.1</td>
</tr>
<tr>
<td></td>
<td>User requirements, Existing system evaluation, Logical system design</td>
<td>9.2.2</td>
</tr>
<tr>
<td>Detailed systems design</td>
<td>Detailed system specification</td>
<td>9.2.3</td>
</tr>
<tr>
<td>Implementation</td>
<td>Coding, testing, and debugging, Installation, fine-tuning</td>
<td>9.2.4</td>
</tr>
<tr>
<td>Maintenance</td>
<td>Evaluation, Maintenance, Enhancement</td>
<td>9.2.5</td>
</tr>
</tbody>
</table>
Planning

• Yields general overview of company and its objectives

• Initial assessment made of information-flow-and-extent requirements

• Must begin to study and evaluate alternate solutions
  – Technical aspects of hardware and software requirements
  – System cost
Analysis

- Problems defined during planning phase are examined in greater detail during analysis
- Thorough audit of user requirements
- Existing hardware and software systems are studied
- Goal is better understanding of system’s functional areas, actual and potential problems, and opportunities
Analysis (continued)

• Includes creation of logical system design
  – Must specify appropriate conceptual data model, inputs, processes, and expected output requirements
  – Might use tools such as data flow diagrams (DFDs), hierarchical input process output (HIPO) diagrams, and entity relationship (ER) diagrams
  – Yields functional descriptions of system’s components (modules) for each process within database environment
Detailed Systems Design

- Designer completes design of system’s processes
- Includes all necessary technical specifications
- Steps are laid out for conversion from old to new system
- Training principles and methodologies are also planned
Implementation

- Hardware, DBMS software, and application programs are installed, and database design is implemented
- Cycle of coding, testing, and debugging continues until database is ready to be delivered
- Database is created and system is customized by creation of tables and views, and user authorizations
Maintenance

- Maintenance activities group into three types:
  - Corrective maintenance in response to systems errors
  - Adaptive maintenance due to changes in business environment
  - Perfective maintenance to enhance system

- Computer-assisted systems engineering
  - Make it possible to produce better systems within reasonable amount of time and at reasonable cost
The Database Life Cycle (DBLC)

**FIGURE 9.3** The Database Life Cycle (DBLC)

<table>
<thead>
<tr>
<th>Phase</th>
<th>Action(s)</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database initial study</td>
<td>Analyze the company situation, Define problems and constraints, Define objectives, Define scope and boundaries</td>
<td>9.3.1</td>
</tr>
<tr>
<td>Database design</td>
<td>Create the conceptual design, DBMS software selection, Create the logical design, Create the physical design</td>
<td>9.3.2</td>
</tr>
<tr>
<td>Implementation and loading</td>
<td>Install the DBMS, Create the database(s), Load or convert the data</td>
<td>9.3.3</td>
</tr>
<tr>
<td>Testing and evaluation</td>
<td>Test the database, Fine-tune the database, Evaluate the database and its application programs</td>
<td>9.3.4</td>
</tr>
<tr>
<td>Operation</td>
<td>Produce the required information flow</td>
<td>9.3.5</td>
</tr>
<tr>
<td>Maintenance and evolution</td>
<td>Introduce changes, Make enhancements</td>
<td>9.3.6</td>
</tr>
</tbody>
</table>
The Database Initial Study

• Overall purpose:
  – Analyze company situation
  – Define problems and constraints
  – Define objectives
  – Define scope and boundaries

• Interactive and iterative processes required to complete first phase of DBLC successfully
The Database Initial Study (continued)
Analyze the Company Situation

• Analysis—To break up any whole into its parts so as to find out their nature, function, and so on

• Company situation
  – General conditions in which company operates, its organizational structure, and its mission

• Analyze company situation
  – Discover what company’s operational components are, how they function, and how they interact
Define Problems and Constraints

- Managerial view of company’s operation is often different from that of end users
- Designer must continue to carefully probe to generate additional information that will help define problems within larger framework of company operations
- Finding precise answers is important
- Defining problems does not always lead to perfect solution
Define Objectives

- Designer must ensure that database system objectives correspond to those envisioned by end user(s)
- Designer must begin to address following questions:
  - What is proposed system’s initial objective?
  - Will system interface with other existing or future systems in the company?
  - Will system share data with other systems or users?
Define Scope and Boundaries

• Scope
  – Defines extent of design according to operational requirements
  – Helps define required data structures, type and number of entities, and physical size of database

• Boundaries
  – Limits external to system
  – Often imposed by existing hardware and software
Database Design

• Necessary to concentrate on data

• Characteristics required to build database model

• Two views of data within system:
  – Business view of data as information source
  – Designer’s view of data structure, its access, and activities required to transform data into information
Database Design (continued)

FIGURE 9.5 Two views of data: business manager and designer

- Company
  - Engineering
  - Purchasing
  - Manufacturing

Manager’s view
- What are the problems?
- What are the solutions?
- What information is needed to implement the solutions?
- What data are required to generate the desired information?

Designer’s view
- How must the data be structured?
- How will the data be accessed?
- How are the data transformed into information?

Company Database
Database Design (continued)

- Loosely related to analysis and design of larger system
- Systems analysts or systems programmers are in charge of designing other system components
  - Their activities create procedures that will help transform data within database into useful information
- Does not constitute sequential process
  - Iterative process that provides continuous feedback designed to trace previous steps
Database Design (continued)

FIGURE 9.6 Procedure flow in the database design

I. Conceptual Design
- Database analysis and requirements
  - Determine end-user views, outputs, and transaction-processing requirements.
- Entity relationship modeling and normalization
- Data model verification
  - Identify main processes. Insert, update, and delete rules. Validate reports, queries, views, integrity, sharing, and security.
- Distributed database design*
  - Define the location of tables, access requirements, and fragmentation strategy.

II. DBMS software selection

III. Logical design
- Translate the conceptual model into definitions for tables, views, and so on.

IV. Physical design
- Define storage structures and access paths for optimum performance.

* See Chapter 12, “Distributed Database Management Systems.”
I. Conceptual Design

- Data modeling used to create an abstract database structure that represents real-world objects in most realistic way possible
- Must embody clear understanding of business and its functional areas
- Ensure that all data needed are in model, and that all data in model are needed
- Requires four steps
I. Conceptual Design (continued)

• Data Analysis and Requirements
  – First step is to discover data element characteristics
    • Obtains characteristics from different sources
  – Must take into account business rules
    • Derived from description of operations
      – Document that provides precise, detailed, up-to-date, and thoroughly reviewed description of activities that define organization’s operating environment
I. Conceptual Design (continued)

- Entity Relationship (ER) Modeling and Normalization
  - Designer must communicate and enforce appropriate standards to be used in documentation of design
    - Use of diagrams and symbols
    - Documentation writing style
    - Layout
    - Other conventions to be followed during documentation
I. Conceptual Design (continued)

<table>
<thead>
<tr>
<th>STEP</th>
<th>ACTIVITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Identify, analyze, and refine the business rules.</td>
</tr>
<tr>
<td>2</td>
<td>Identify the main entities, using the results of Step 1.</td>
</tr>
<tr>
<td>3</td>
<td>Define the relationships among the entities, using the results of Steps 1 and 2.</td>
</tr>
<tr>
<td>4</td>
<td>Define the attributes, primary keys, and foreign keys for each of the entities.</td>
</tr>
<tr>
<td>5</td>
<td>Normalize the entities. (Remember that entities are implemented as tables in an RDBMS.)</td>
</tr>
</tbody>
</table>
### Developing the Conceptual Model, Using ER Diagrams (continued)

<table>
<thead>
<tr>
<th>STEP</th>
<th>ACTIVITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>Complete the initial ER diagram.</td>
</tr>
<tr>
<td>7</td>
<td>Have the main end users verify the model in Step 6 against the data, information, and processing requirements.</td>
</tr>
<tr>
<td>8</td>
<td>Modify the ER diagram, using the results of Step 7.</td>
</tr>
</tbody>
</table>
I. Conceptual Design (continued)
### I. Conceptual Design (continued)

#### TABLE 9.2
**Data Redundancies in the VIDEO Table**

<table>
<thead>
<tr>
<th>VIDEO_ID</th>
<th>VIDEO_TITLE</th>
<th>VIDEO_COPY</th>
<th>VIDEO_CHG</th>
<th>VIDEO_DAYS</th>
</tr>
</thead>
<tbody>
<tr>
<td>SF-12345FT-1</td>
<td>Adventures on Planet III</td>
<td>1</td>
<td>$4.50</td>
<td>1</td>
</tr>
<tr>
<td>SF-12345FT-2</td>
<td>Adventures on Planet III</td>
<td>2</td>
<td>$4.50</td>
<td>1</td>
</tr>
<tr>
<td>SF-12345FT-3</td>
<td>Adventures on Planet III</td>
<td>3</td>
<td>$4.50</td>
<td>1</td>
</tr>
<tr>
<td>WE-5432GR-1</td>
<td>TipToe Canu and Tyler 2: A Journey</td>
<td>1</td>
<td>$2.99</td>
<td>2</td>
</tr>
<tr>
<td>WE-5432GR-2</td>
<td>TipToe Canu and Tyler 2: A Journey</td>
<td>2</td>
<td>$2.99</td>
<td>2</td>
</tr>
</tbody>
</table>
I. Conceptual Design (continued)
I. Conceptual Design (continued)
I. Conceptual Design (continued)

- Entity Relationship (ER) Modeling and Normalization (continued)
  - Data dictionary
    - Defines all objects (entities, attributes, relations, views, and so on)
    - Used in tandem with the normalization process to help eliminate data anomalies and redundancy problems
I. Conceptual Design (continued)

• Data Model Verification
  – Model must be verified against proposed system processes to corroborate that intended processes can be supported by database model
  – Revision of original design starts with careful reevaluation of entities, followed by detailed examination of attributes that describe these entities
  – Define design’s major components as modules:
    • An information system component that handles specific function
## I. Conceptual Design (continued)

<table>
<thead>
<tr>
<th>STEP</th>
<th>ACTIVITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Identify the ER model’s central entity.</td>
</tr>
<tr>
<td>2</td>
<td>Identify each module and its components.</td>
</tr>
</tbody>
</table>
| 3    | Identify each module’s transaction requirements:  
      | Internal: Updates/Inserts/Deletes/Queries/Reports  
      | External: Module interfaces |
| 4    | Verify all processes against the ER model. |
| 5    | Make all necessary changes suggested in Step 4. |
| 6    | Repeat Steps 2–5 for all modules. |
I. Conceptual Design (continued)
I. Conceptual Design (continued)

• Data Model Verification (continued)
  – Verification process
    • Select central (most important) entity
      – Defined in terms of its participation in most of model’s relationships
    • Identify module or subsystem to which central entity belongs and define boundaries and scope
    • Place central entity within module’s framework
I. Conceptual Design (continued)

• Distributed Database Design
  – Portions of database may reside in different physical locations
  • Designer must also develop data distribution and allocation strategies
II. DBMS Software Selection

• Critical to information system’s smooth operation

• Advantages and disadvantages should be carefully studied
III. Logical Design

- Used to translate conceptual design into internal model for selected database management system

- Logical design is software-dependent

- Requires that all objects in model be mapped to specific constructs used by selected database software
III. Logical Design (continued)

FIGURE 9.11  A simple conceptual model
### III. Logical Design (continued)

#### TABLE 9.4
Sample Layout for the COURSE Table

<table>
<thead>
<tr>
<th>CRS_CODE</th>
<th>CRS_TITLE</th>
<th>CRS_DESCRIPNT</th>
<th>CRS_CREDIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIS-4567</td>
<td>Database Systems Design</td>
<td>Design and implementation of database systems; includes conceptual design,</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>logical design, implementation, and management; prerequisites: CIS 2040,</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CIS 2345, and CIS 3680 and upper-division standing</td>
<td></td>
</tr>
<tr>
<td>QM-3456</td>
<td>Statistics II</td>
<td>Statistical applications; course requires use of statistical software (MINITAB</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>and SAS) to interpret data; prerequisites: MATH 2345 and QM 2233</td>
<td></td>
</tr>
</tbody>
</table>
IV. Physical Design

• Process of selecting data storage and data access characteristics of database

• Storage characteristics are function of device types supported by hardware, type of data access methods supported by system, and DBMS

• Particularly important in older hierarchical and network models

• Becomes more complex when data are distributed at different locations
Implementation and Loading

• New database implementation requires creation of special storage-related constructs to house end-user tables
Implementation and Loading (continued)
Performance

• One of most important factors in certain database implementations

• Not all DBMSs have performance-monitoring and fine-tuning tools embedded in their software

• There is no standard measurement for database performance

• Not only (nor even main) factor
Security

• Data must be protected from access by unauthorized users

• Must provide for following:
  – Physical security
  – Password security
  – Access rights
  – Audit trails
  – Data encryption
  – Diskless workstations
Backup and Recovery

• Database can be subject to data loss through unintended data deletion and power outages

• Data backup and recovery procedures
  – Create safety valve
    • Allow database administrator to ensure availability of consistent data
Integrity

- Enforced through proper use of primary and foreign key rules
Company Standards

- May partially define database standards
- Database administrator must implement and enforce such standards
Testing and Evaluation

• Occurs in parallel with applications programming
• Database tools used to prototype applications
• If implementation fails to meet some of system’s evaluation criteria:
  – Fine-tune specific system and DBMS configuration parameters
  – Modify physical design
  – Modify logical design
  – Upgrade or change DBMS software and/or hardware platform
Operation

• Once database has passed evaluation stage, it is considered operational

• Beginning of operational phase starts process of system evolution
Maintenance and Evolution

• Required periodic maintenance:
  – Preventive maintenance (backup)
  – Corrective maintenance (recovery)
  – Adaptive maintenance
  – Assignment of access permissions and their maintenance for new and old users
  – Generation of database access statistics
  – Periodic security audits
  – Periodic system-usage summaries
Maintenance and Evolution (continued)
Database Design Strategies

- Two classical approaches to database design:
  - Top-down design
    - Identifies data sets
    - Defines data elements for each of those sets
  - Bottom-up design
    - Identifies data elements (items)
    - Groups them together in data sets
Database Design Strategies (continued)

FIGURE 9.14 Top-down vs. bottom-up design sequencing

Conceptual model

Entity

Entity

Attribute

Attribute

Attribute

Attribute
Centralized vs. Decentralized Design

- Database design may be based on two very different design philosophies:
  - Centralized design
    - Productive when data component is composed of relatively small number of objects and procedures
  - Decentralized design
    - Used when data component of system has considerable number of entities and complex relations on which very complex operations are performed
Centralized vs. Decentralized Design (continued)
Centralized vs. Decentralized Design (continued)
Centralized vs. Decentralized Design (continued)

• Aggregation process
  – Requires designer to create single model in which various aggregation problems must be addressed:
    • Synonyms and homonyms
    • Entity and entity subtypes
    • Conflicting object definitions
Centralized vs. Decentralized Design (continued)
Summary

• Information system is designed to facilitate transformation of data into information and to manage both data and information

• SDLC traces history (life cycle) of an application within the information system
Summary (continued)

- DBLC describes history of database within the information system
- Database design and implementation process moves through series of well-defined stages
- Conceptual portion of design may be subject to several variations, based on two design philosophies
Summary

- Data management is critical activity for any organization
- Data should be treated as corporate asset
- DBMS is most commonly used electronic tool for corporate data management
- Impact of DBMS on organization’s managerial and cultural framework must be carefully examined