Chapter 8

Database Design

Database Systems:
Design, Implementation, and Management,
Sixth Edition, Rob and Coronel

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)
- 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)

In this chapter, you will learn (continued):

- How to conduct evaluation and revision within the SDLC and DBLC frameworks
- What database design strategies exist: top-down vs. bottom-up design, and centralized vs. decentralized design

Changing Data into Information

- Information
  - Data processed and presented in a meaningful form
  - Can be as simple as tabulating the data, thereby making certain data patterns more obvious
- Transformation
  - Any process that changes data into information

A Simple Cross-Classification Table: Transforming Data into Information

TABLE 8.1 A Simple Cross-Classification Table: Transforming Data into Information

	UNDER 25	25 – 45	46 – 60	61 AND OVER	TOTAL
Male	119	1,892	2,641	876	5,528
Female	48	1,117	1,805	542	3,512
Total	167	3,009	4,446	1,418	9,040

The Information System

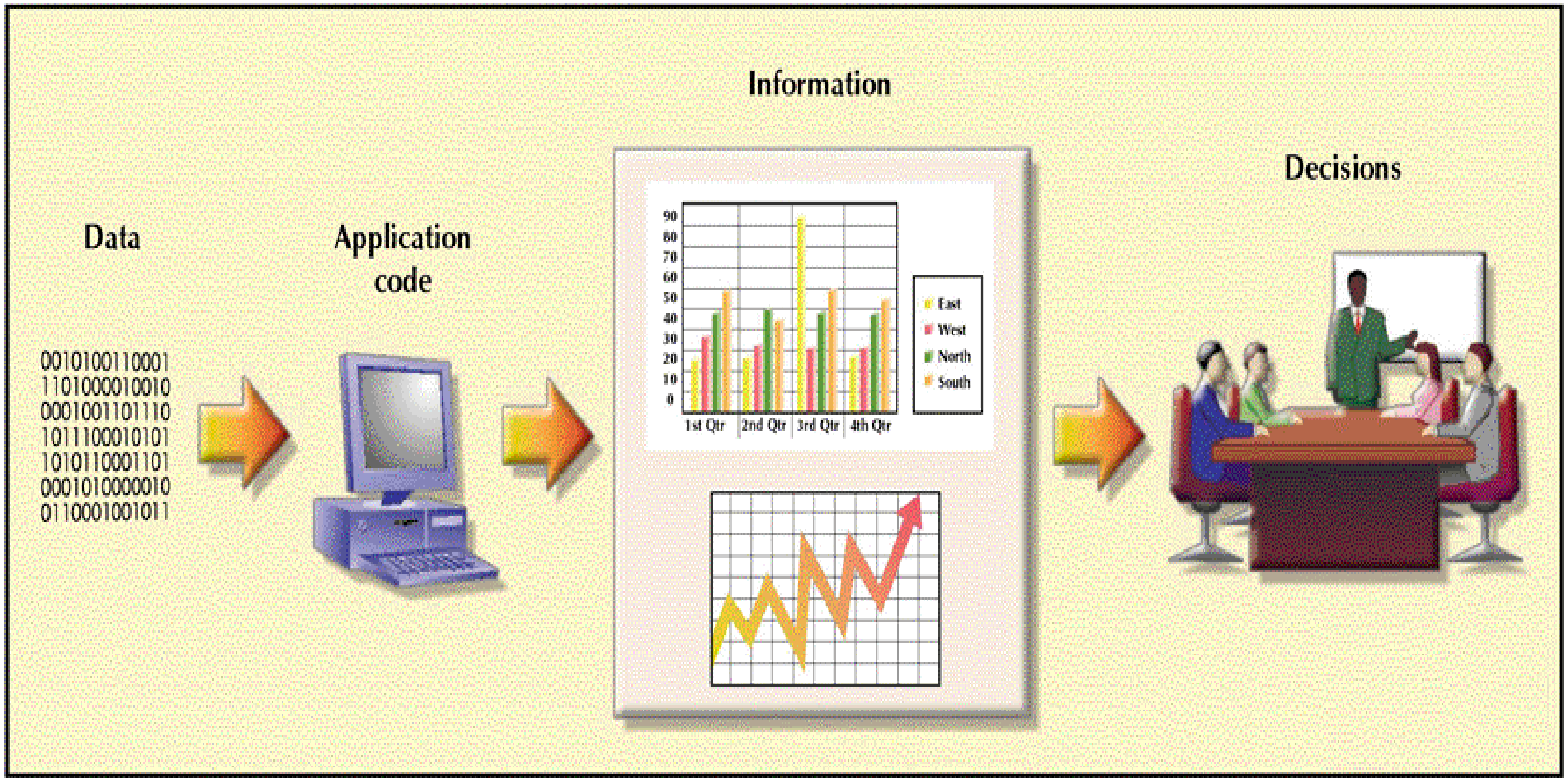
- Provides for data collection, storage, and retrieval
- Composed of people, hardware, software, database(s), application programs, and procedures
- Systems analysis
  - Process that establishes need for and extent of an information system
- Systems development
  - Process of creating an information system

Applications

- Transform data into information that forms the basis for decision making
- Usually produce
  - Formal report
  - Tabulations
  - Graphic displays
- Composed of two parts
  - Data
  - Code by which the data are transformed into information

Generating Information for Decision Making

FIGURE 8.1 GENERATING INFORMATION FOR DECISION MAKING



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 an 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)

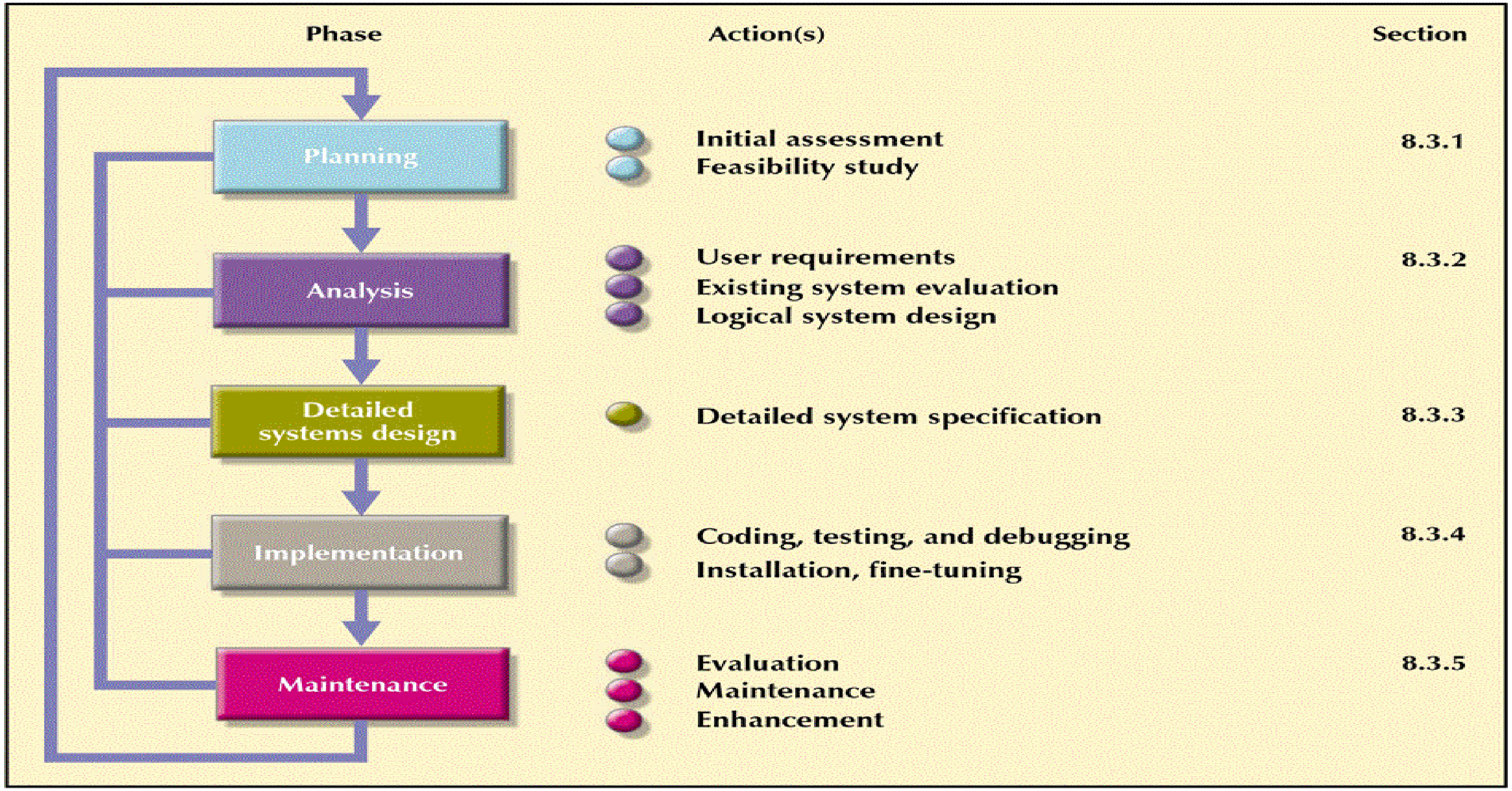
- Divided into five phases
  1. Planning
  2. Analysis
  3. Detailed systems design
  4. Implementation
  5. Maintenance
- Iterative rather than sequential process

Planning

- Yields general overview of the 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

The Systems Development Life Cycle (SDLC)

FIGURE 8.2 THE SYSTEMS DEVELOPMENT LIFE CYCLE (SDLC)



Analysis

- Problems defined during the planning phase are examined in greater detail during analysis
- A 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

Logical Systems Design

- Must specify appropriate conceptual data model, inputs, processes, and expected output requirements
- Might use tools such as data flow diagrams (DFD), hierarchical input process output (HIPO) diagrams, or 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

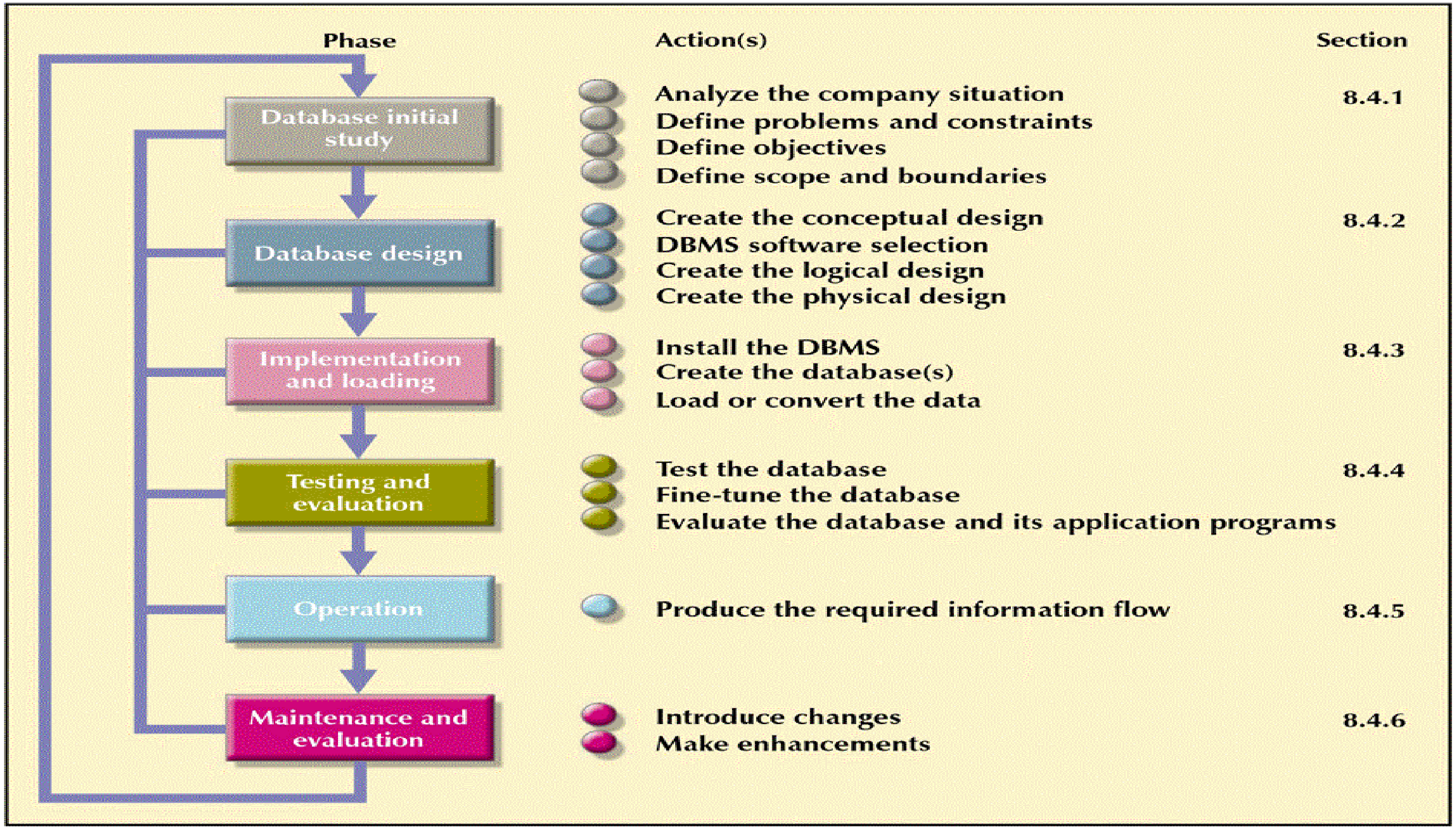
- Hardware, DBMS software, and application programs are installed, and database design is implemented
- Cycle of coding, testing, and debugging continues until database is ready to be delivered
- Database is created and system is customized by creation of tables and views, and user authorizations

Maintenance

- Three types:
  - *Corrective maintenance* in response to systems errors
  - *Adaptive maintenance* due to changes in the business environment
  - *Perfective maintenance* to enhance the system
- Computer-assisted systems engineering
  - Make it possible to produce better systems within reasonable amount of time and at a reasonable cost

The Database Life Cycle (DBLC)

FIGURE 8.3 THE DATABASE LIFE CYCLE (DBLC)

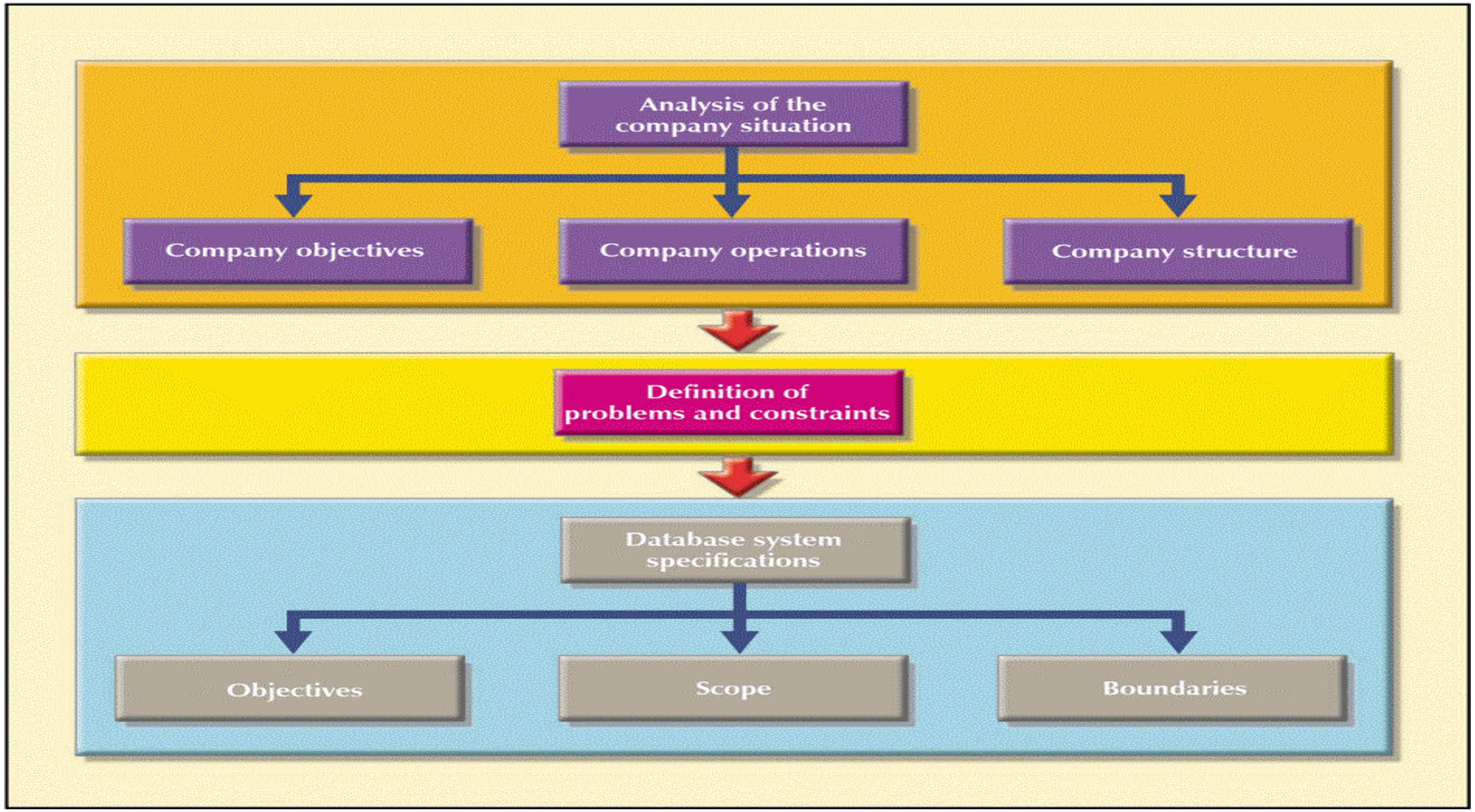


The Database Initial Study

- Overall purpose:
  - Analyze the company situation
  - Define problems and constraints
  - Define objectives
  - Define scope and boundaries
- Interactive and iterative processes required to complete the first phase of the DBLC successfully

Summary of Activities in the Database Initial Study

FIGURE 8.4 SUMMARY OF ACTIVITIES IN THE DATABASE INITIAL STUDY



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 a company operates, its organizational structure, and its mission
- Analyze the company situation
  - Discover what the 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 the perfect solution

Define Objectives

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

Define Scope and Boundaries

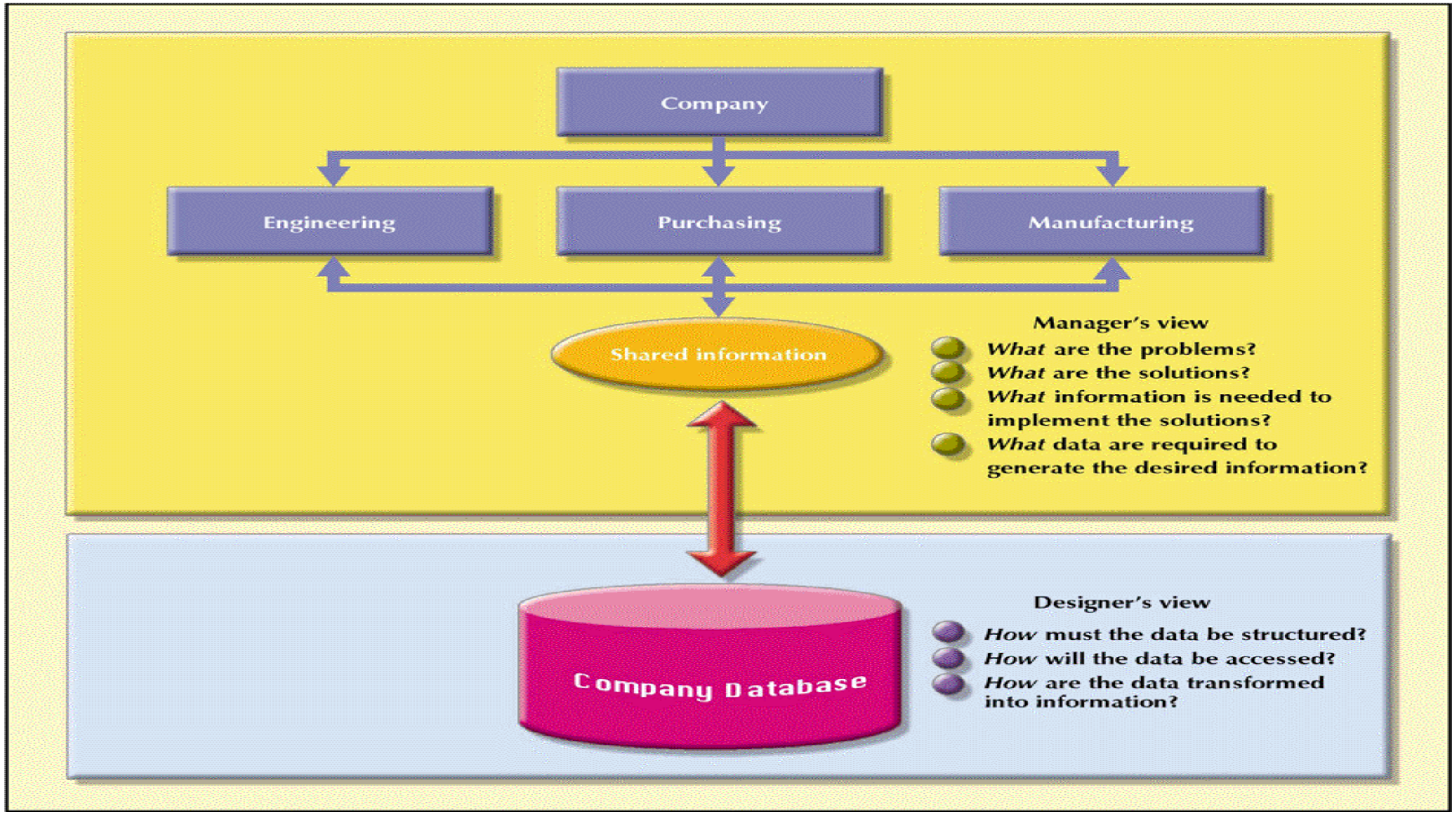
- Scope
  - Defines extent of design according to operational requirements
  - Helps define required data structures, type and number of entities, and physical size of the database
- Boundaries
  - Limits external to the system
  - Often imposed by existing hardware and software

Database Design

- Necessary to concentrate on the 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 the activities required to transform the data into information

Two Views of Data: Business Manager and Designer

FIGURE 8.5 Two Views of Data: Business Manager and Designer

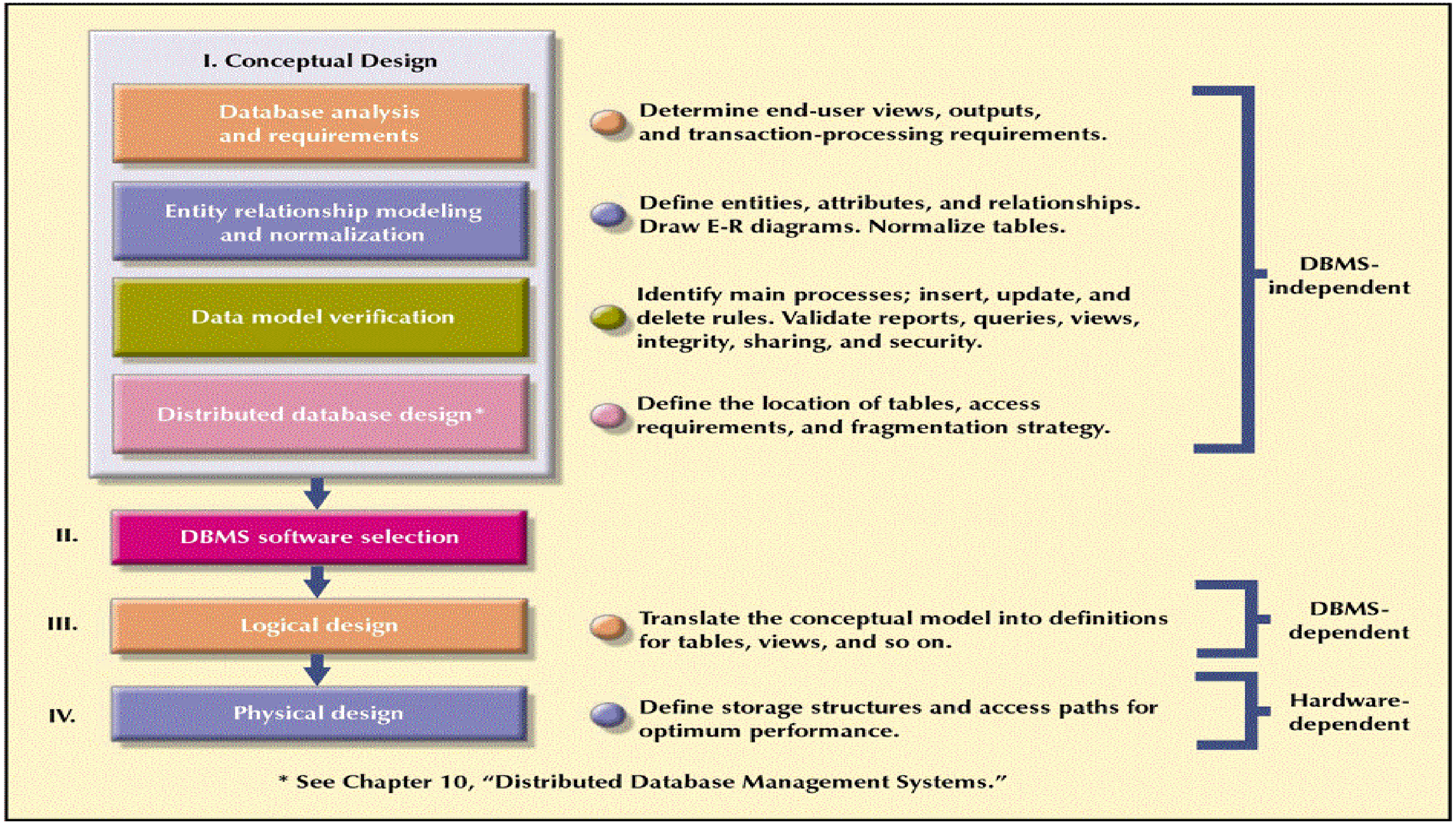


Database Design (continued)

- 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 the database into useful information
- Does not constitute a sequential process
  - Iterative process that provides continuous feedback designed to trace previous steps

Procedure Flow in the Database Design

FIGURE 8.6 PROCEDURE FLOW IN THE DATABASE DESIGN



Conceptual Design

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

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 an organization’s operating environment

Entity Relationship (ER) Modeling and Normalization

- Designer must communicate and enforce appropriate standards to be used in the documentation of design
  - Use of diagrams and symbols
  - Documentation writing style
  - Layout
  - Other conventions to be followed during documentation

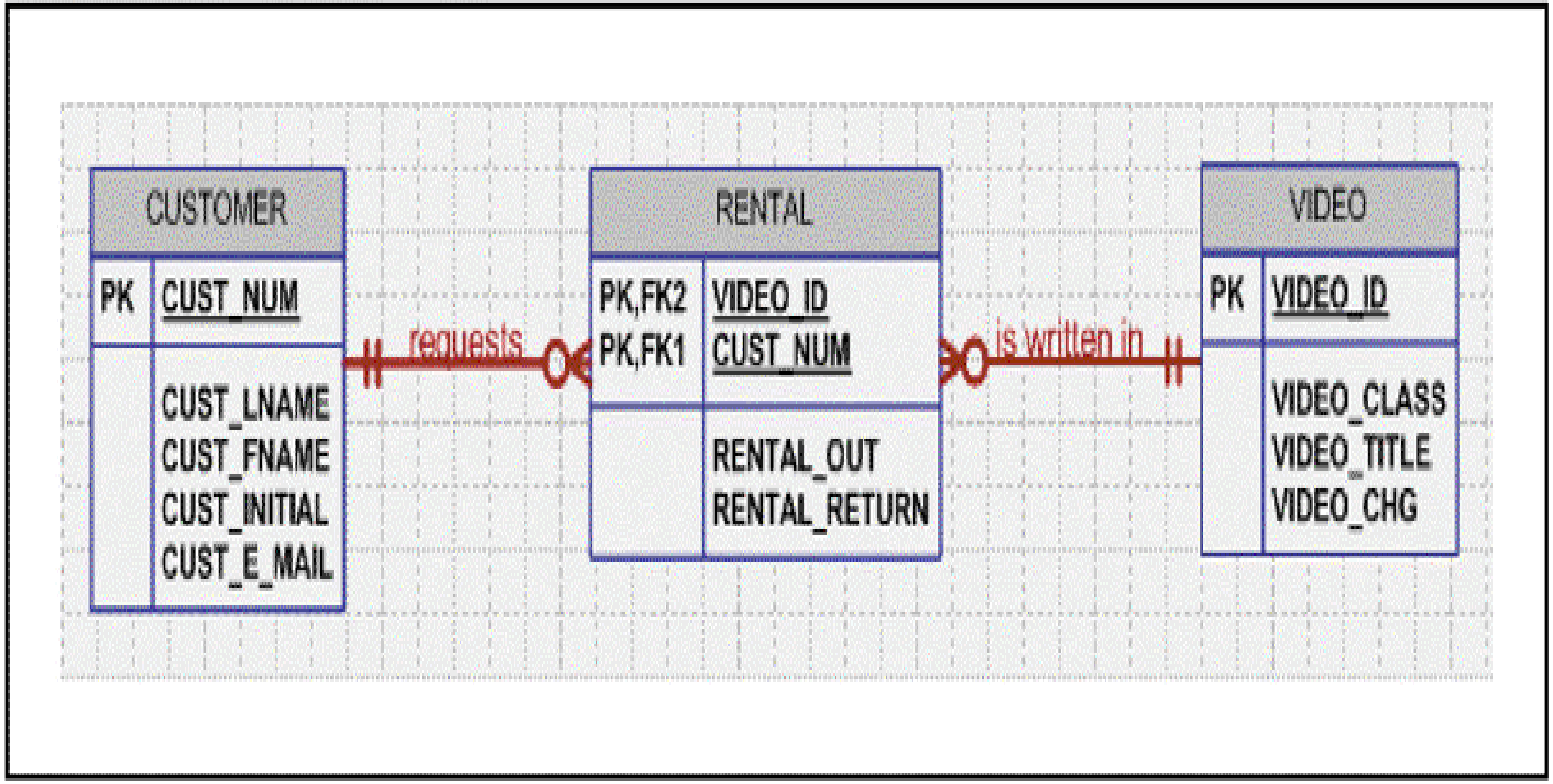
Developing the Conceptual Model Using ER Diagrams

TABLE 8.2 DEVELOPING THE CONCEPTUAL MODEL USING ER DIAGRAMS

STEP	ACTIVITY
1	Identify, analyze, and refine the business rules.
2	Identify the main entities, using the results of Step 1.
3	Define the relationships among the entities, using the results of Steps 1 and 2.
4	Define the attributes, primary keys, and foreign keys for each of the entities.
5	Normalize the entities.
6	Complete the initial ER diagram.
7	Have the main end users verify the model in Step 6 against the data, information, and processing requirements.
8	Modify the ER diagram, using the results of Step 7.

A Composite Entity

FIGURE 8.7 A COMPOSITE ENTITY



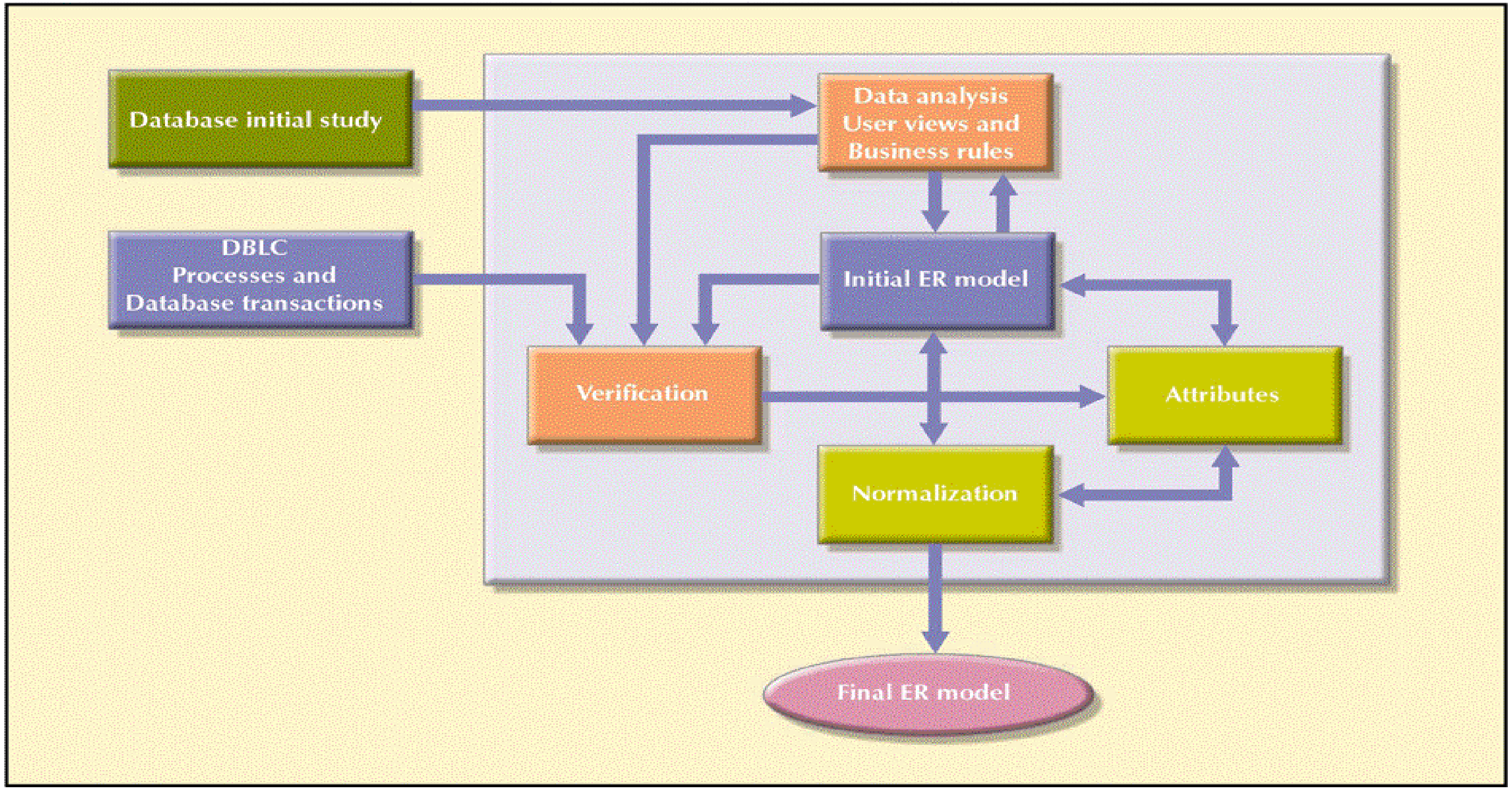
Data Redundancies in the VIDEO Table

TABLE 8.3 DATA REDUNDANCIES IN THE VIDEO TABLE

VIDEO_ID	VIDEO_TITLE	VIDEO_COPY	VIDEO_CHG	VIDEO_DAYS
SF-12345FT-1	Adventures on Planet III	1	$2.45	1
SF-12345FT-2	Adventures on Planet III	2	$2.45	1
SF-12345FT-3	Adventures on Planet III	3	$2.45	1
WE-5432GR-1	TipToe Canu and Tyler 2: A Journey	1	$1.99	2
WE-5432GR-2	TipToe Canu and Tyler 2: A Journey	2	$1.99	2

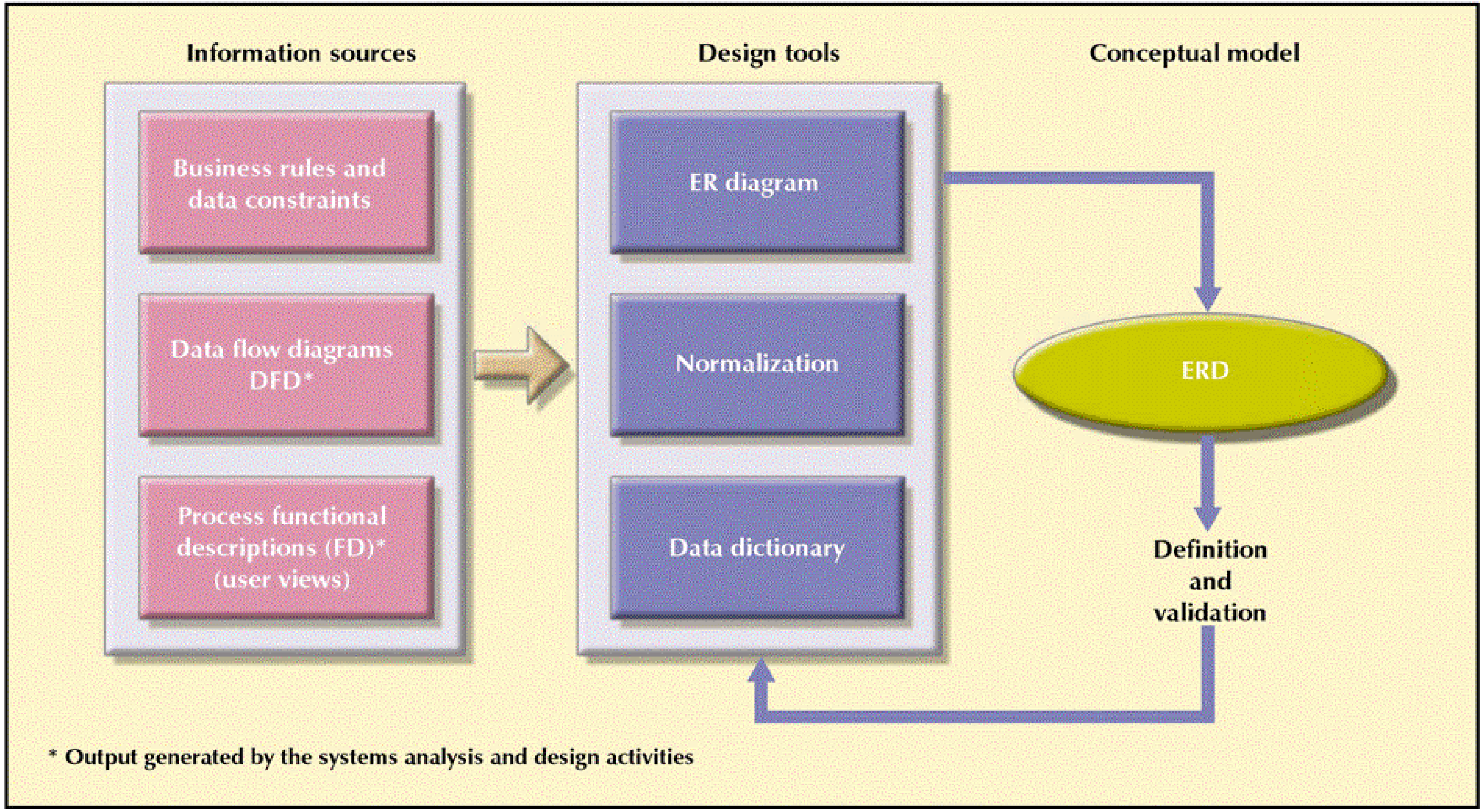
ER Modeling Is an Iterative Process Based on Many Activities

FIGURE 8.8 ER MODELING IS AN ITERATIVE PROCESS BASED ON MANY ACTIVITIES



Conceptual Design Tools and Information Sources

FIGURE 8.9 CONCEPTUAL DESIGN TOOLS AND INFORMATION SOURCES



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

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 a careful reevaluation of entities, followed by a detailed examination of attributes that describe these entities
- Define design's major components as *modules:*
  - An information system component that handles a specific function

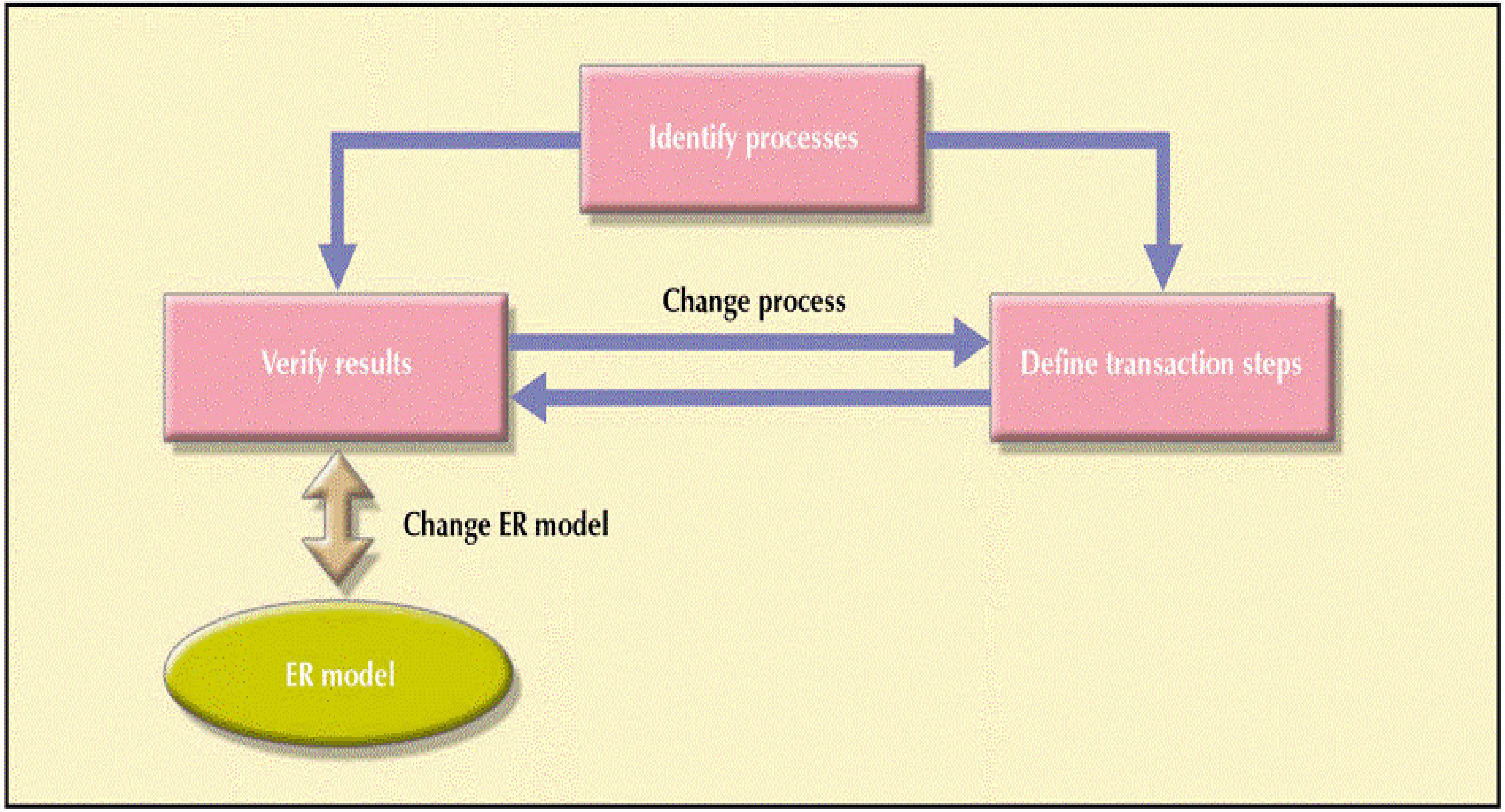
The ER Model Verification Process

TABLE 8.4 THE ER MODEL VERIFICATION PROCESS

STEP	ACTIVITY
1	Identify the ER model's central entity.
2	Identify each module and its components.
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 through 5 for all modules.

Iterative ER Model Verification Process

FIGURE 8.10 ITERATIVE ER MODEL VERIFICATION PROCESS



Verification Process

- Select the central (most important) entity
  - Defined in terms of its participation in most of the model's relationships
- Identify the module or subsystem to which the central entity belongs and define boundaries and scope
- Place central entity within the module's framework

DBMS Software Selection

- Critical to the information system's smooth operation
- Advantages and disadvantages should be carefully studied

Logical Design

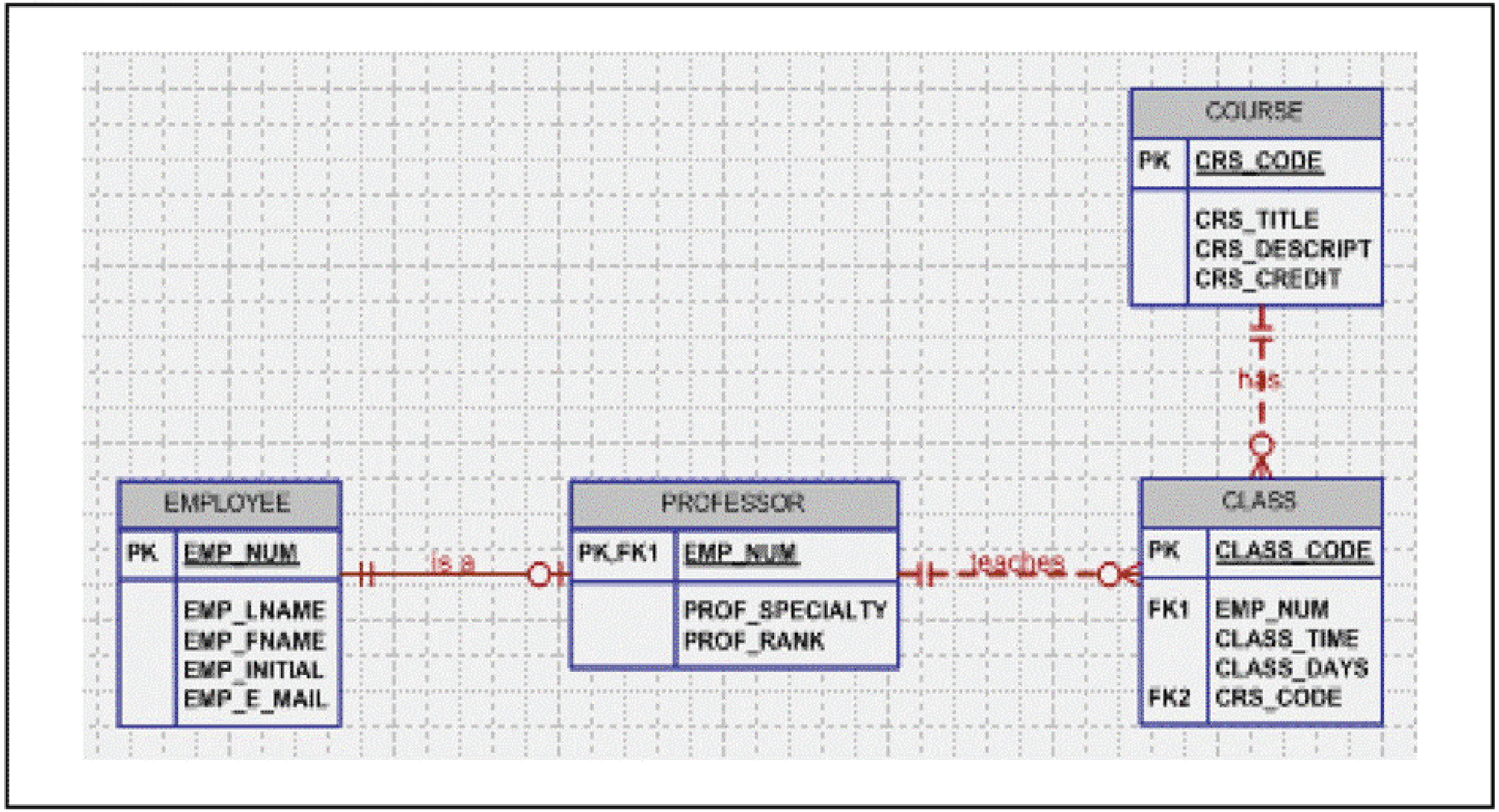
- Used to translate conceptual design into internal model for a selected database management system

- Logical design is software-dependent

- Requires that all objects in the model be mapped to specific constructs used by selected database software

A Simple Conceptual Model

FIGURE 8.11 A SIMPLE CONCEPTUAL MODEL



Sample Layout for the COURSE Table

TABLE 8.5 SAMPLE LAYOUT FOR THE COURSE TABLE

CRS_CODE	CRS_TITLE	CRS_DESCRIPT	CRS_CREDIT
CIS-4567	Database Systems Design	Design and implementation of database systems. Includes conceptual design, logical design, implementation, and management. Prerequisites: CIS 2040, CIS-2345, and CIS 3680 and upper division standing.	4
QM-3456	Statistics II	Statistical applications. Course requires use of statistical software (MINITAB and SAS) to interpret data. Prerequisites: MATH-2345 and QM-2233.	3

Physical Design

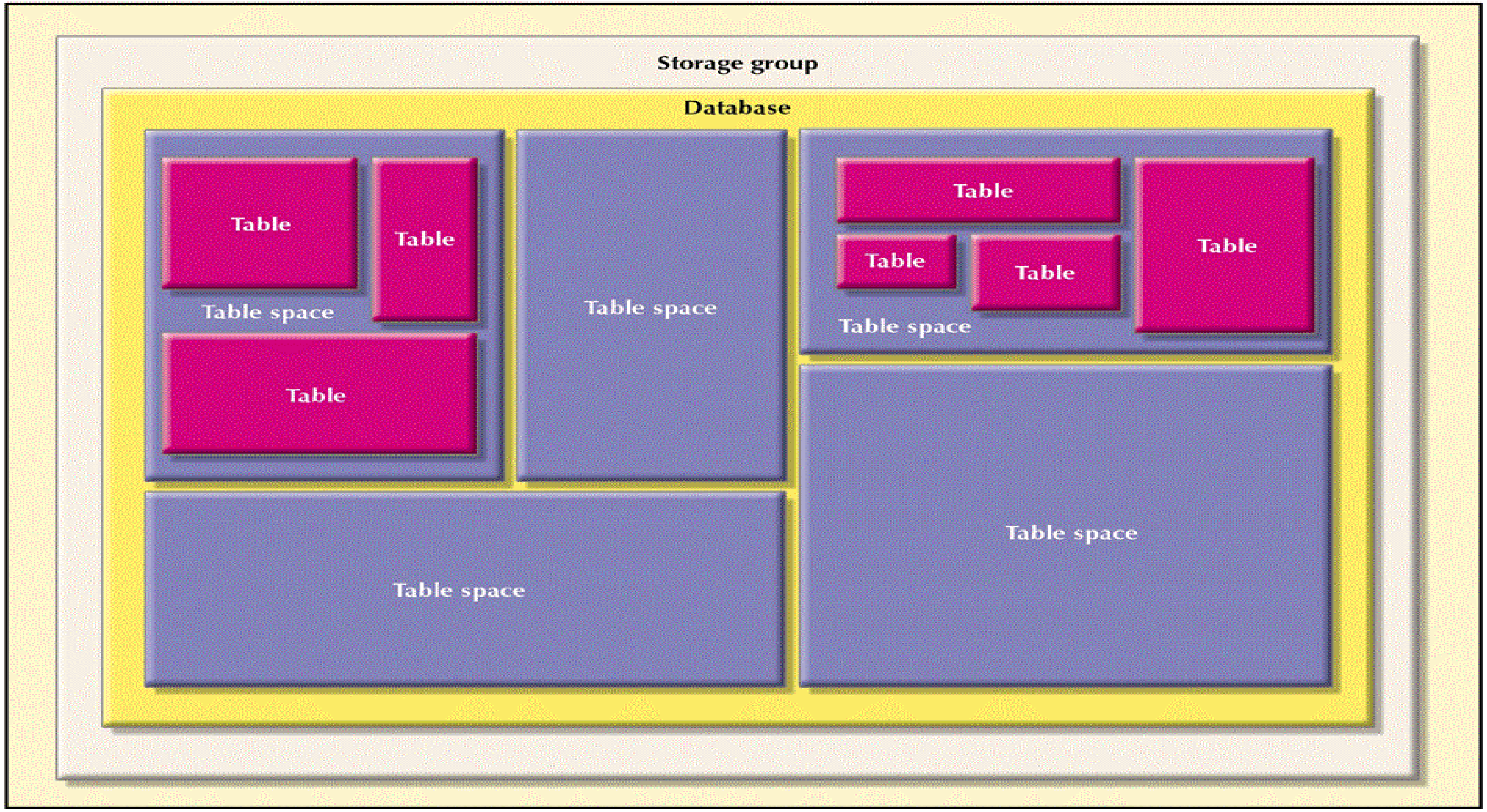
- Process of selecting data storage and data access characteristics of the database
- Storage characteristics are a function of device types supported by the hardware, type of data access methods supported by system, and DBMS
- Particularly important in the older hierarchical and network models
- Becomes more complex when data are distributed at different locations

Implementation and Loading

- New database implementation requires the creation of special storage-related constructs to house the end-user tables

Physical Organization of a DB2 Database Environment

FIGURE 8.12 PHYSICAL ORGANIZATION OF A DB2 DATABASE ENVIRONMENT



Performance

- One of the 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 the 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 a 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

Concurrency Control

- Feature that allows simultaneous access to a database while preserving data integrity
- Failure to maintain can quickly destroy a database's effectiveness

The Need for Concurrency Control

TABLE 8.6 THE NEED FOR CONCURRENCY CONTROL

PROCESS A (MEMORY)	PROCESS B (MEMORY)	DATABASE (STORED STOCK)	TIME LINE
Read 500 ←	←	500	T1
	Read 500 ←	500	T2
Stock = 500 – 150	Write →	350	T3
	Stock = 500 – 300—write →	200	T4
	Value stored in database:	200	

Testing and Evaluation

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

Operation

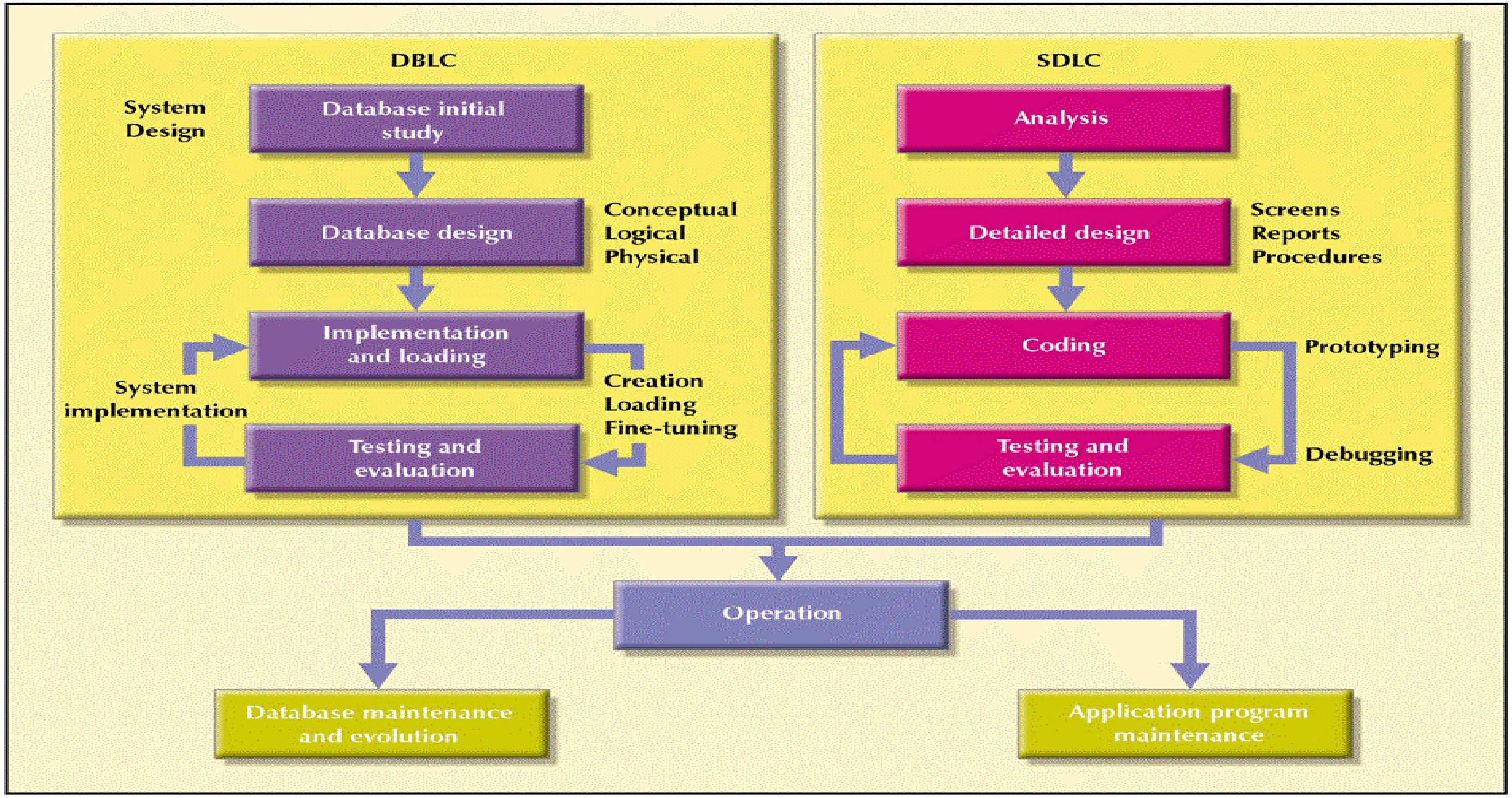
- Once the database has passed the evaluation stage, it is considered operational
- Beginning of the operational phase starts the process of system evolution

Maintenance and Evolution

- Required periodic maintenance:
  - Preventive maintenance
  - Corrective maintenance
  - 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

Parallel Activities in the DBLC and the SDLC

FIGURE 8.13 PARALLEL ACTIVITIES IN THE DBLC AND THE SDLC

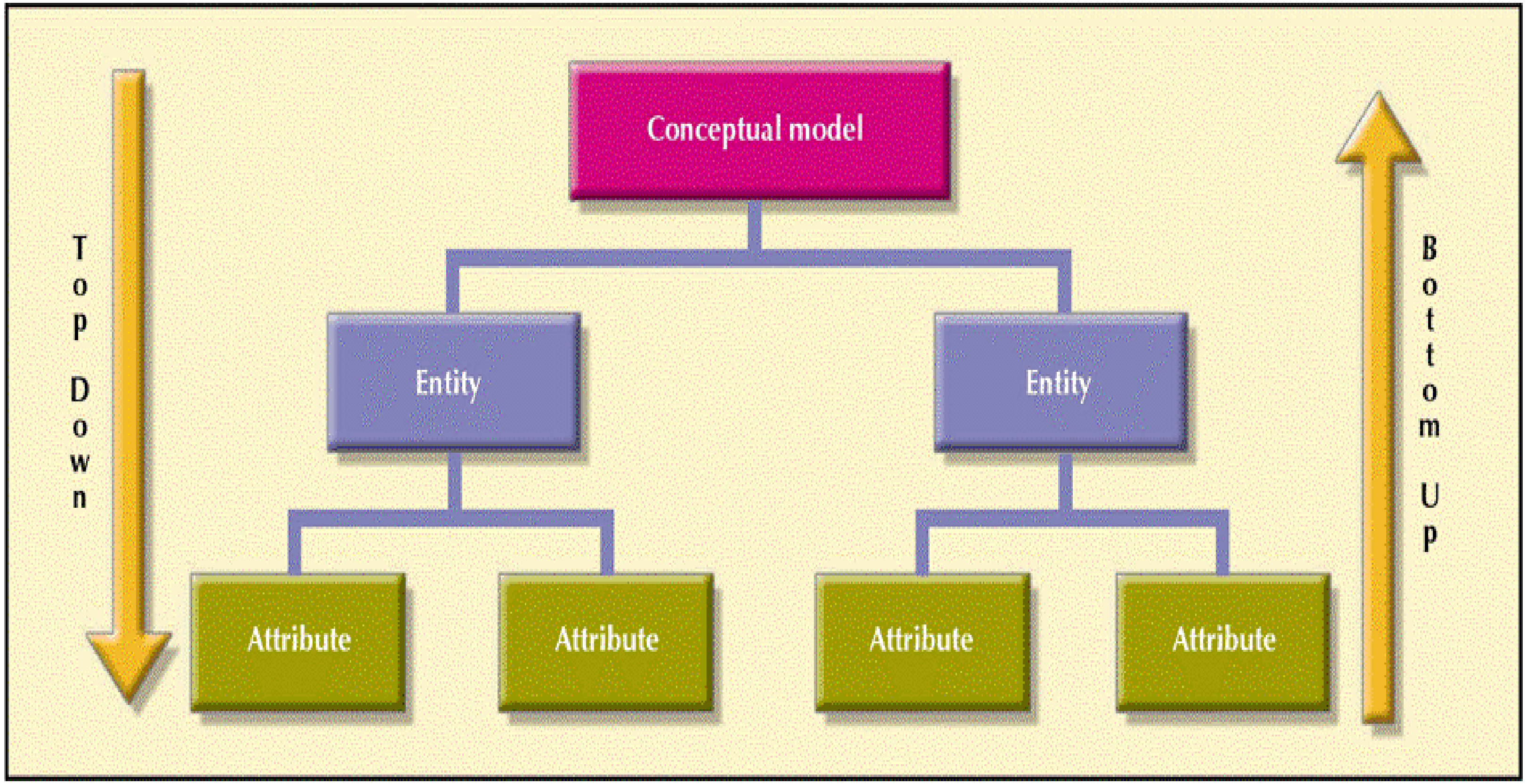


A Special Note about Database Design Strategies

- 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

Top-Down vs. Bottom-Up Design Sequencing

FIGURE 8.14 TOP-DOWN VS. BOTTOM-UP DESIGN SEQUENCING

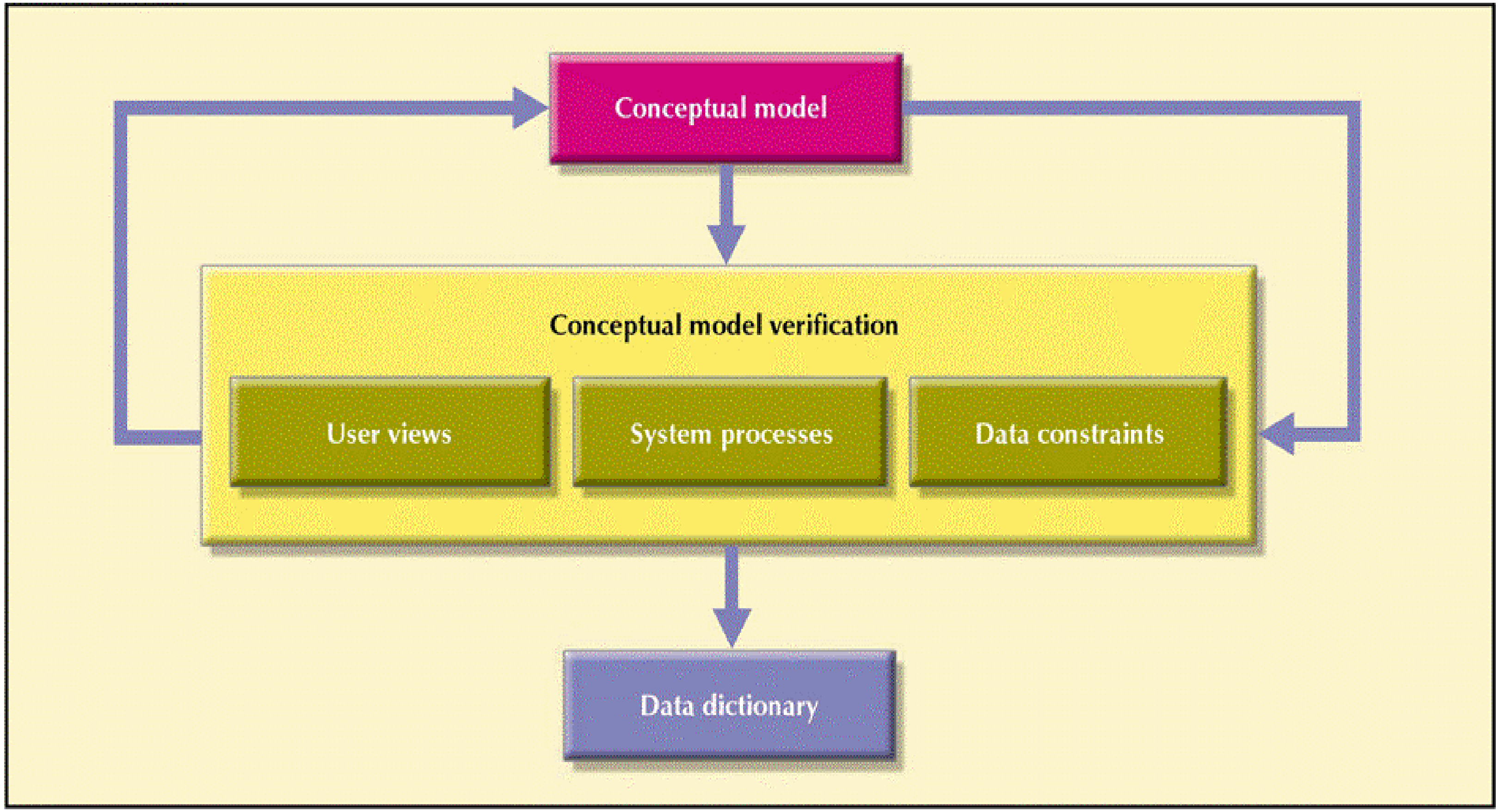


Centralized vs. Decentralized Design

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

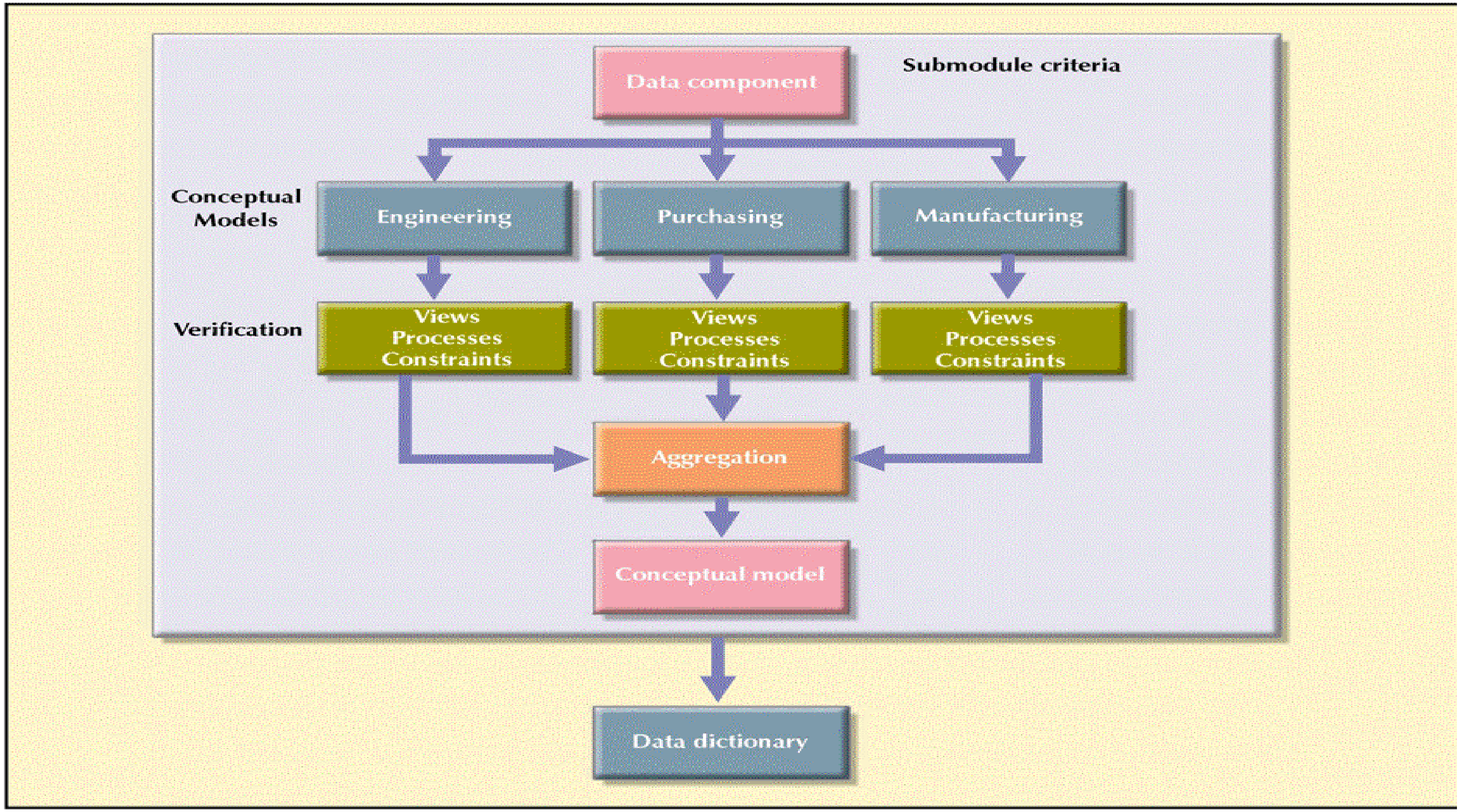
Centralized Design

FIGURE 8.15 CENTRALIZED DESIGN



Decentralized Design

FIGURE 8.16 DECENTRALIZED DESIGN

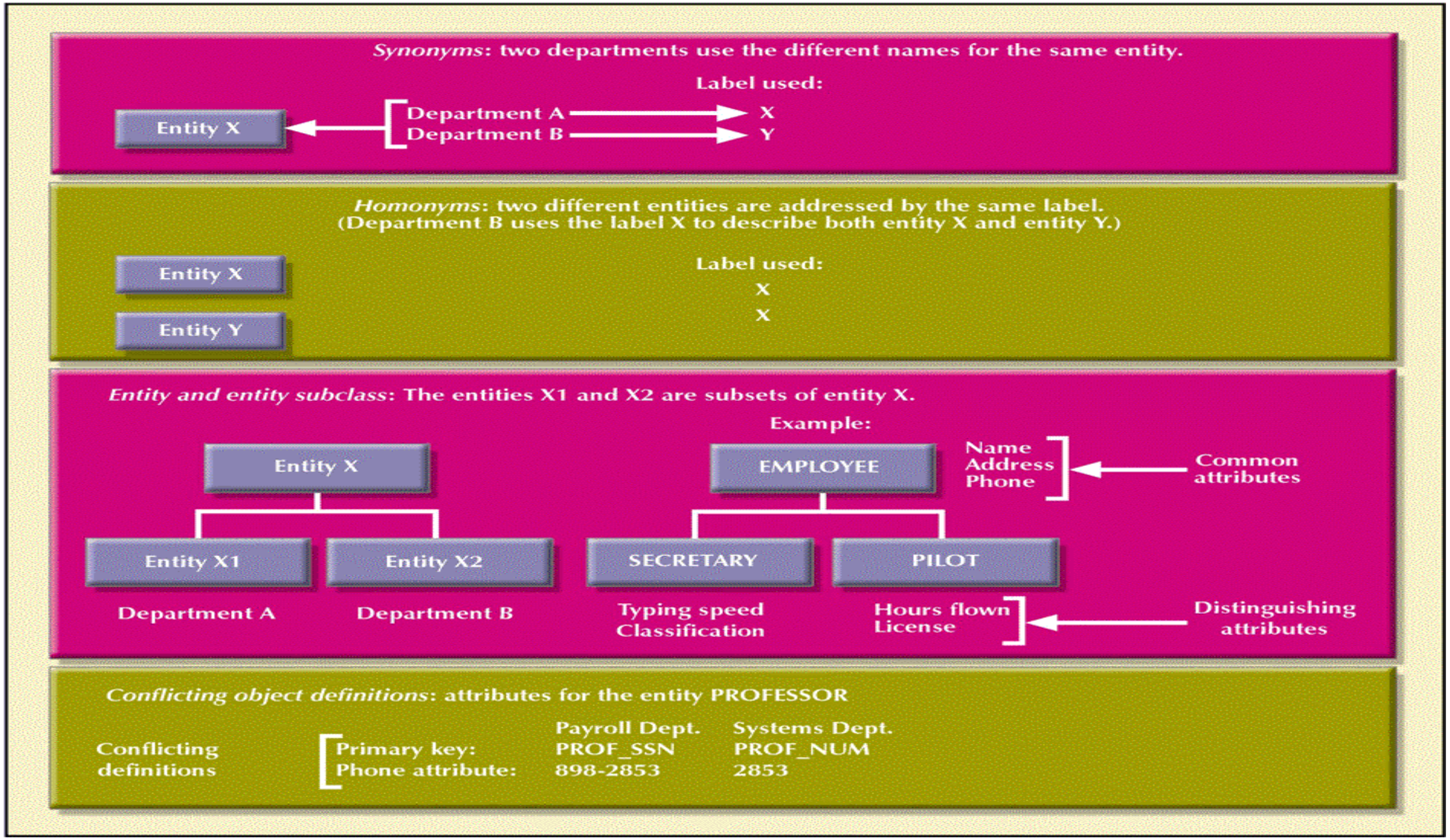


Aggregation Process

- Requires designer to create a single model in which various aggregation problems must be addressed:
  - Synonyms and homonyms
  - Entity and entity subtypes
  - Conflicting object definitions

Summary of Aggregation Problems

FIGURE 8.17 SUMMARY OF AGGREGATION PROBLEMS



Summary

- Transformation from data to information is produced when programming code operates on the data, thus producing applications
- Information system is designed to facilitate transformation of data into information and to manage both data and information
- SDLC traces the history (life cycle) of an application within the information system

Summary (continued)

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