Database Design

Lecture 11

Learning Outcomes

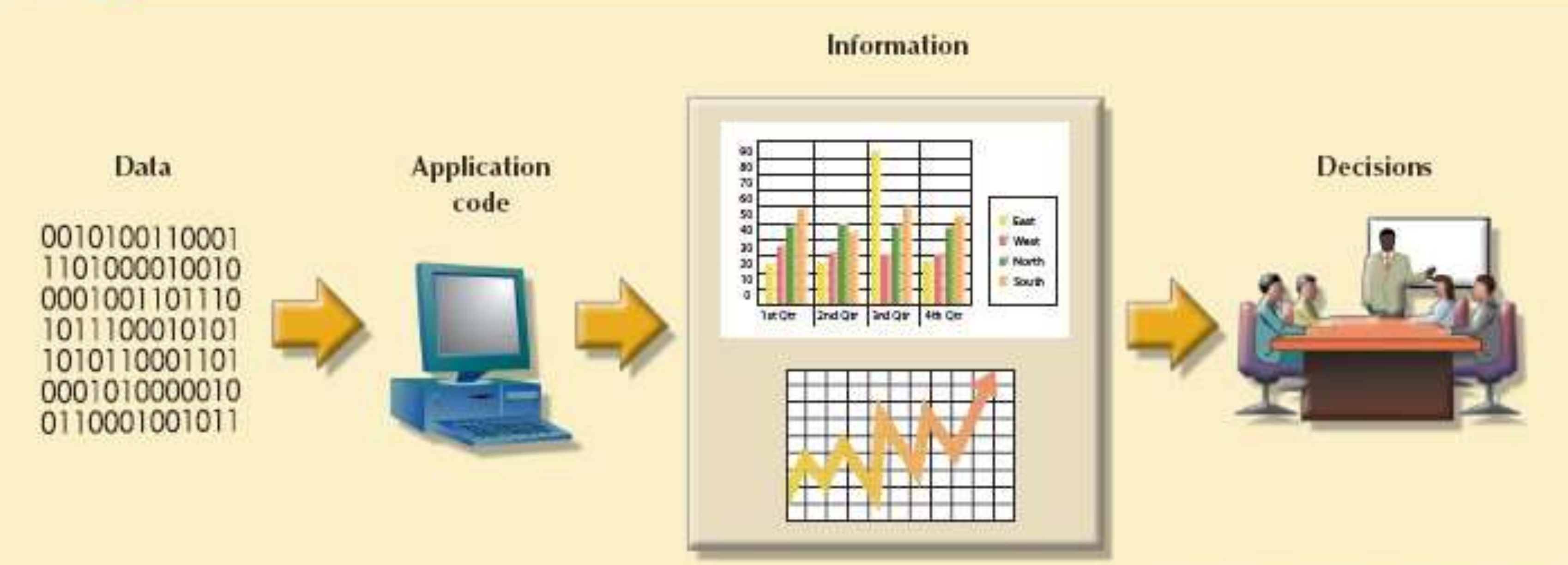
- To instill the concept that successful database design must reflect the information system of which the database is a part
- To discuss how successful IS are developed within a framework called SDLC
- To discuss the concept that within IS, the most successful databases are subject to frequent evaluation & revision within a framework called DBLC
- To be able to conduct evaluation & revision within the SDLC & DBLC frameworks

The Information System

- Provides for data collection, storage, and retrieval
- Information system helps to transform data into information
- Application program is usually composed of two parts:
  - *Data*
  - *Code by which the data are transformed into information*

Generating Information for Decision Making

FIGURE 2.1 Generating information for decision making



SOURCE: Course Technology/Cengage Learning

Information System

- Performance depends on triad of factors:
  - Database design and implementation
  - Application design and implementation
  - Administrative procedures
- Database development
  - DB design - to create complete, normalized, non-redundant and fully integrated conceptual, logical, and physical database models
  - DB implementation - to create storage structure, load data into database, provide data for management

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)

- Divided into five phases
  - *Planning*
  - *Analysis*
  - *Detailed systems design*
  - *Implementation*
  - *Maintenance*
- Iterative rather than sequential process

The Systems Development Life Cycle (SDLC)

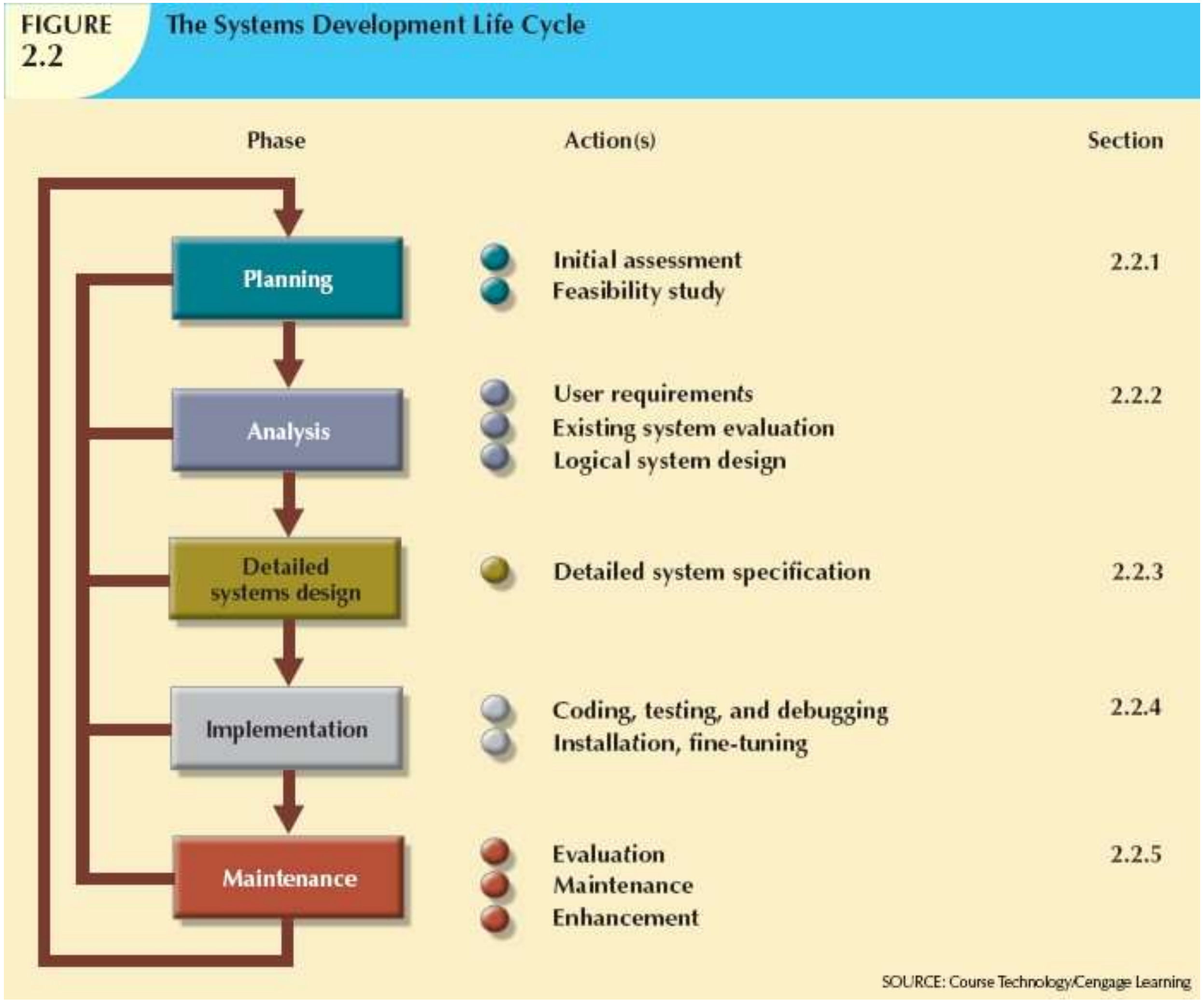


FIGURE 2.2 The Systems Development Life Cycle

Phase	Action(s)	Section
Planning	Initial assessment Feasibility study	2.2.1
Analysis	User requirements Existing system evaluation Logical system design	2.2.2
Detailed systems design	Detailed system specification	2.2.3
Implementation	Coding, testing, and debugging Installation, fine-tuning	2.2.4
Maintenance	Evaluation Maintenance Enhancement	2.2.5

SOURCE: Course Technology/Cengage Learning

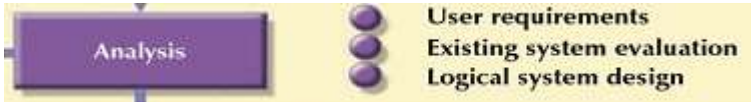
Planning

1
Planning
Initial assessment
Feasibility study

- Yields a ***general overview*** of the company and ***its objectives.***
- An ***initial assessment*** of the information-flow-and-extent requirements must be made:
  - Should the existing system be continued, modified or replaced?
- A ***feasibility study*** must address the following issues if a new system is necessary:
  - Technical aspects of hardware and software requirements.
  - System cost.

Analysis

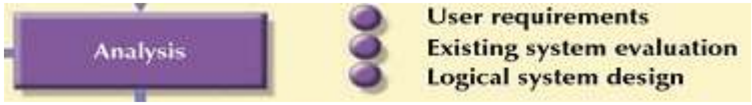
2



- Problems defined during the planning phase are examined in greater detail during analysis
- Goal : better understanding of system’s functional areas, actual and potential problems, and opportunities
- A thorough audit of user requirements conducted
- Existing systems hardware and software are studied

Analysis

2



- Includes creation of logical systems design
  - Must specify appropriate conceptual data model, inputs, processes, and expected output requirements
  - Might use tools such as data flow diagrams (DFD), Unified Modeling Language (UML) diagrams, or entity relationship (ER) diagrams
  - Yields functional descriptions of system's components (modules) for each process within database environment

Detailed System Design

- The designer completes the design of the system’s processes, including all technical specifications for:
  - Screen
  - Menus
  - Reports
  - Other devices
- Steps are laid out for conversion from old to new system
- Training principles and methodologies are planned.

Implementation

4

Implementation
Coding, testing, and debugging
Installation, fine-tuning

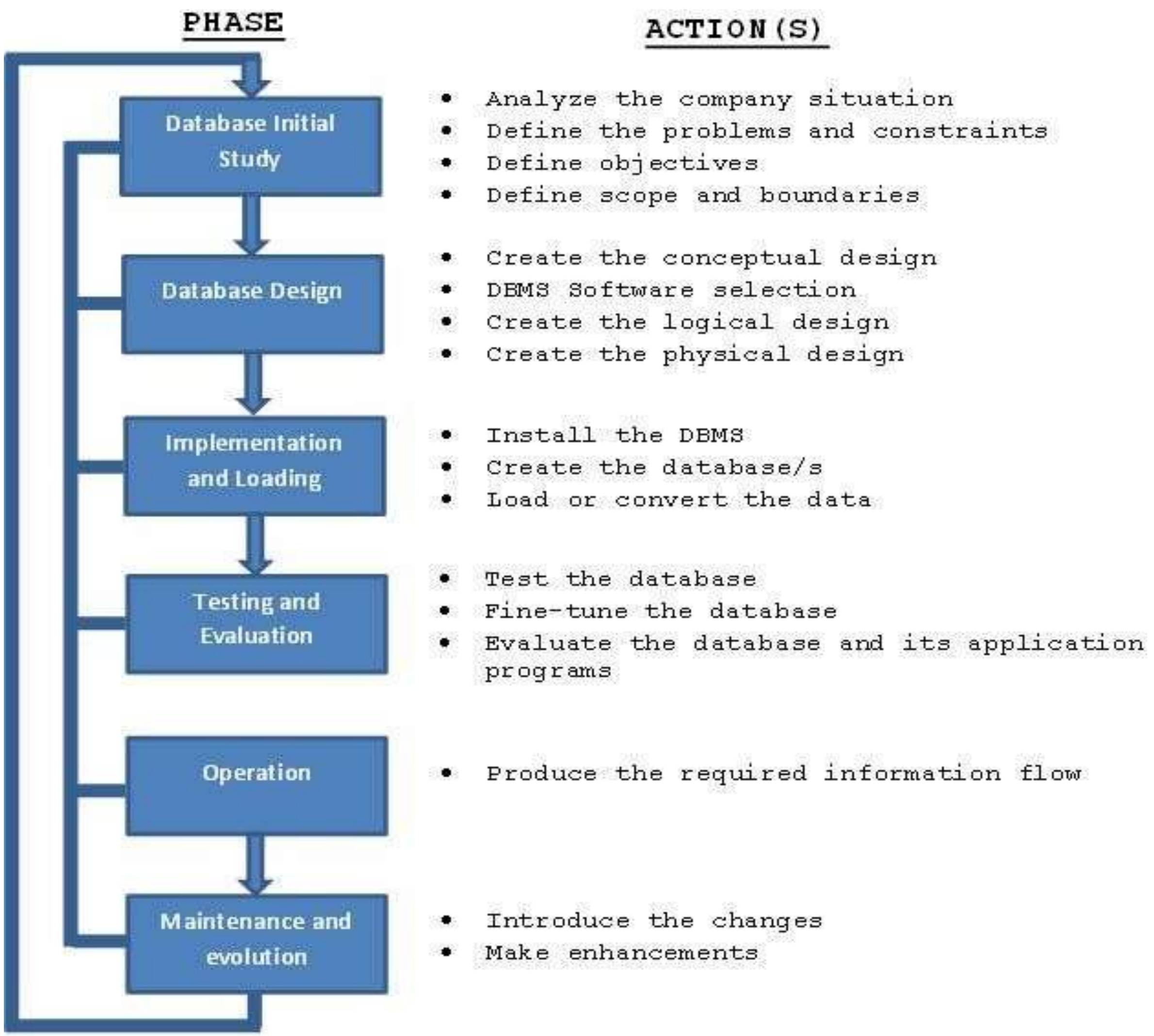
- Hardware, DBMS software, and application programs are installed
- Cycle of coding, testing, and debugging continues until database is ready to be delivered
- Database is created and system is customized by creation of tables and views, and user authorizations

Maintenance

5
Maintenance
Evaluation
Maintenance
Enhancement

- Maintenance activities:
  1. Corrective maintenance in response to systems errors
  2. Adaptive maintenance due to changes in the business environment
  3. Perfective maintenance to enhance the system

Database Life Cycle (DBLC)



PHASE

ACTION(S)

Database Initial Study

- Analyze the company situation
- Define the problems and constraints
- Define objectives
- Define scope and boundaries

Database Design

- Create the conceptual design
- DBMS Software selection
- Create the logical design
- Create the physical design

Implementation and Loading

- Install the DBMS
- Create the database/s
- Load or convert the data

Testing and Evaluation

- Test the database
- Fine-tune the database
- Evaluate the database and its application programs

Operation

- Produce the required information flow

Maintenance and evolution

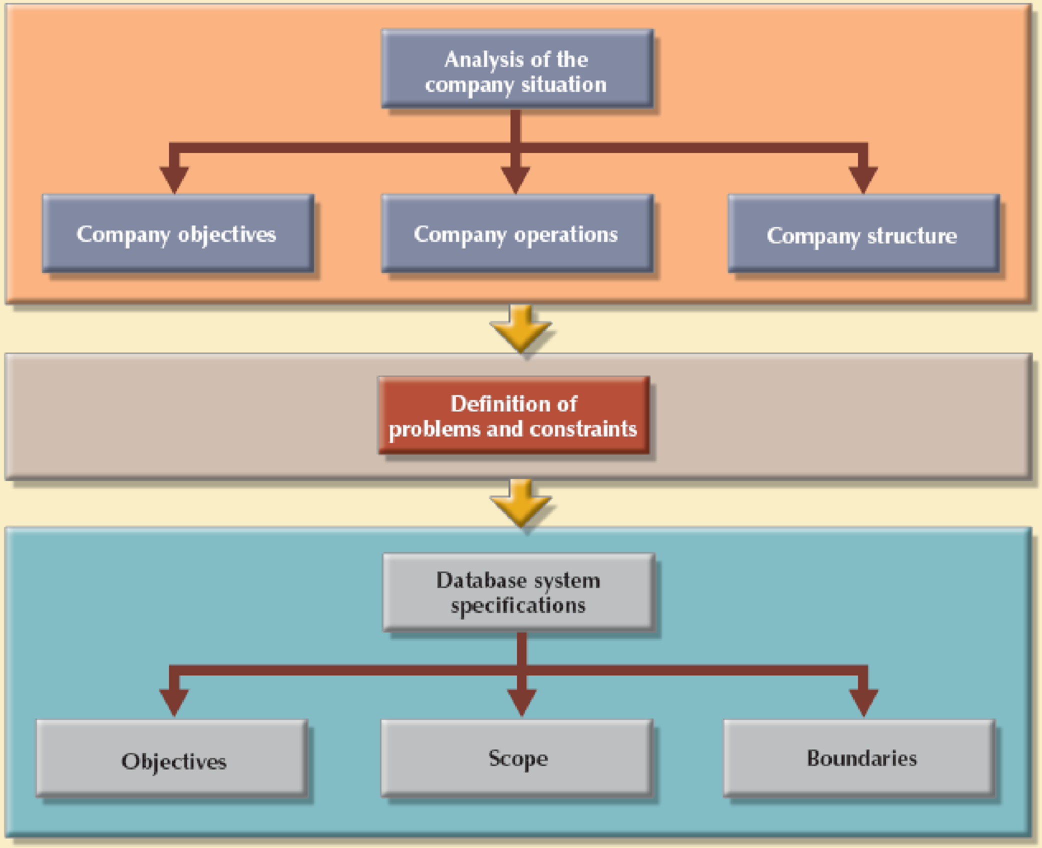
- Introduce the changes
- Make enhancements

The Database Initial Study

Overall purpose:

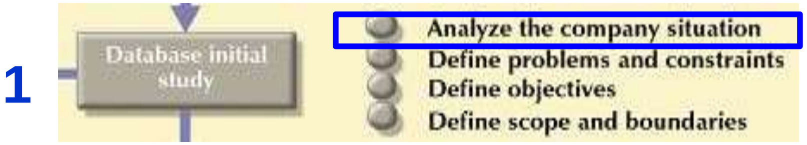
- Analyze the company situation
- Define problems and constraints
- Define objectives, scope and boundaries

FIGURE 2.4 A summary of activities in the database initial study



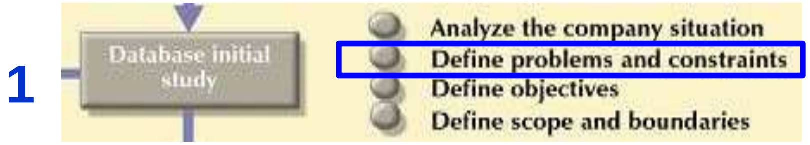
Analysis of the company situation
Company objectives
Company operations
Company structure
Definition of problems and constraints
Database system specifications
Objectives
Scope
Boundaries

SOURCE: Course Technology/Cengage Learning


1
Database initial study
Analyze the company situation
Define problems and constraints
Define objectives
Define scope and boundaries

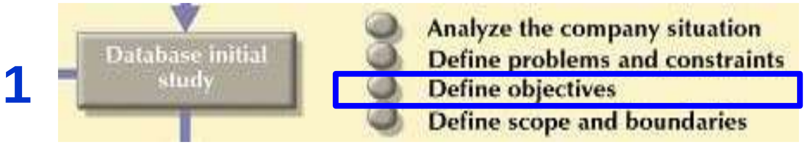
Analyze the Company Situation

- Analyze the company situation
  - Discover what the company's operational components are, how they function, and how they interact
  - Also analyze the company organization's structure


1
Database initial study
Analyze the company situation
Define problems and constraints
Define objectives
Define scope and boundaries

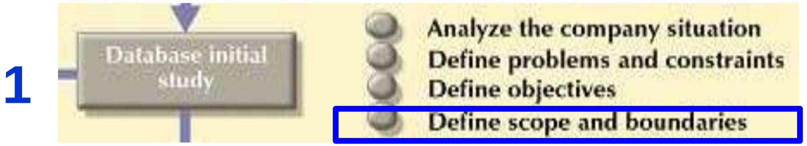
Define Problems and Constraints

- 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.
- Must address questions such as
  - *What are the limits and constraints imposed on the system?*
  - *What is the input and required output?*


1
Database initial study
Analyze the company situation
Define problems and constraints
Define objectives
Define scope and boundaries

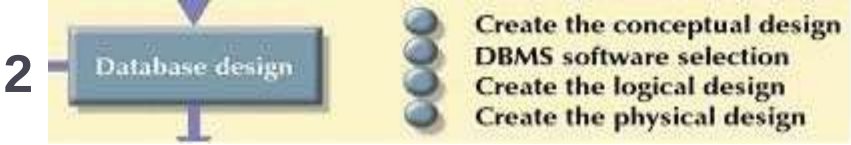
Define Objectives

- 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?*


1
Database initial study
Analyze the company situation
Define problems and constraints
Define objectives
Define scope and boundaries

Define Scope and Boundaries

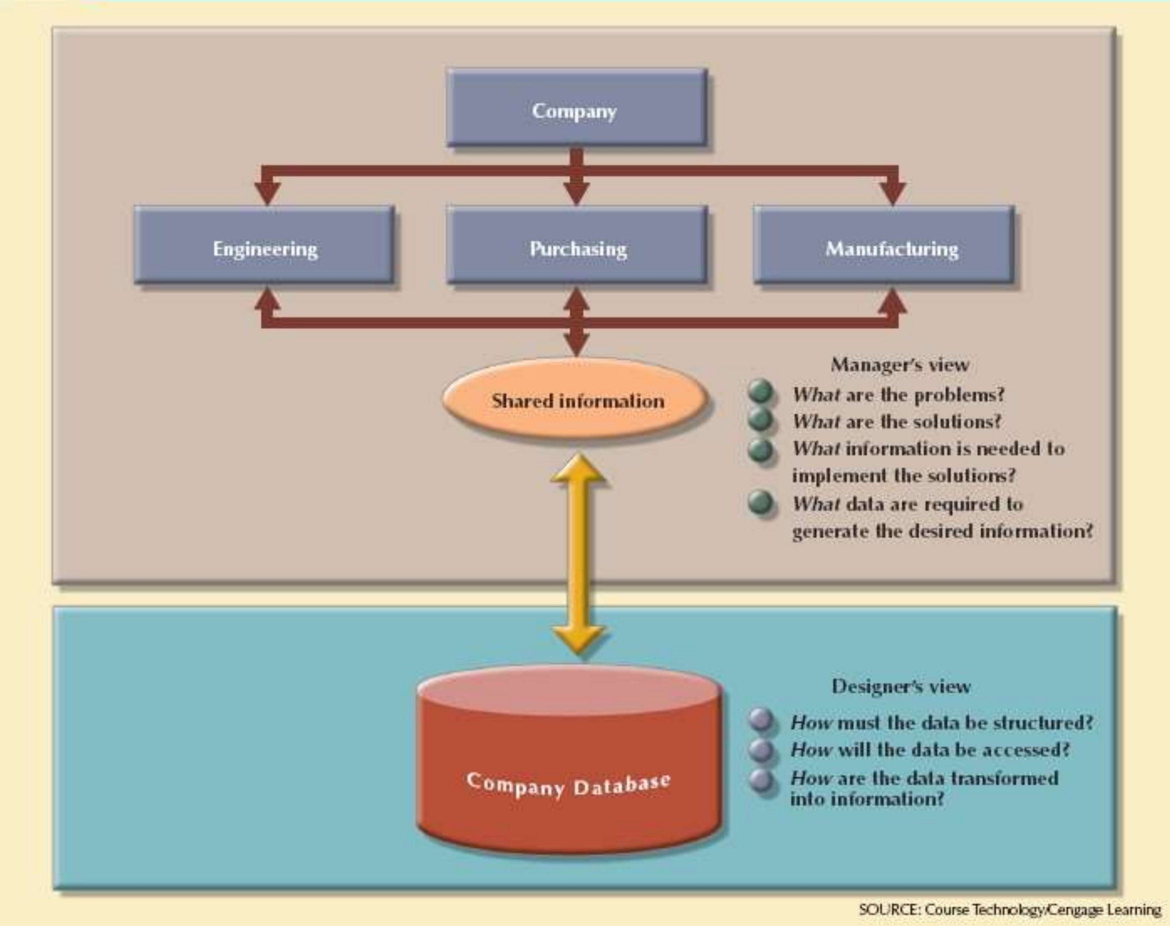
- Scope
  - *Defines extent of design according to operational requirements (one department vs. entire org.)*
  - *Helps define required data structures, type and number of entities, and physical size of the database*
- Boundaries
  - *Limits (external to the system)*
  - *Often imposed by existing hardware and software and budget allocated*


2
Database design
Create the conceptual design
DBMS software selection
Create the logical design
Create the physical design

Database Design

- Two views of data within system:
  - Business view
  - Designer's view

FIGURE 2.5 Two views of data: business manager and database designer


Company
Engineering
Purchasing
Manufacturing
Shared information
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?
Company Database
Designer's view
How must the data be structured?
How will the data be accessed?
How are the data transformed into information?

SOURCE: Course Technology/Cengage Learning

Database Design Process

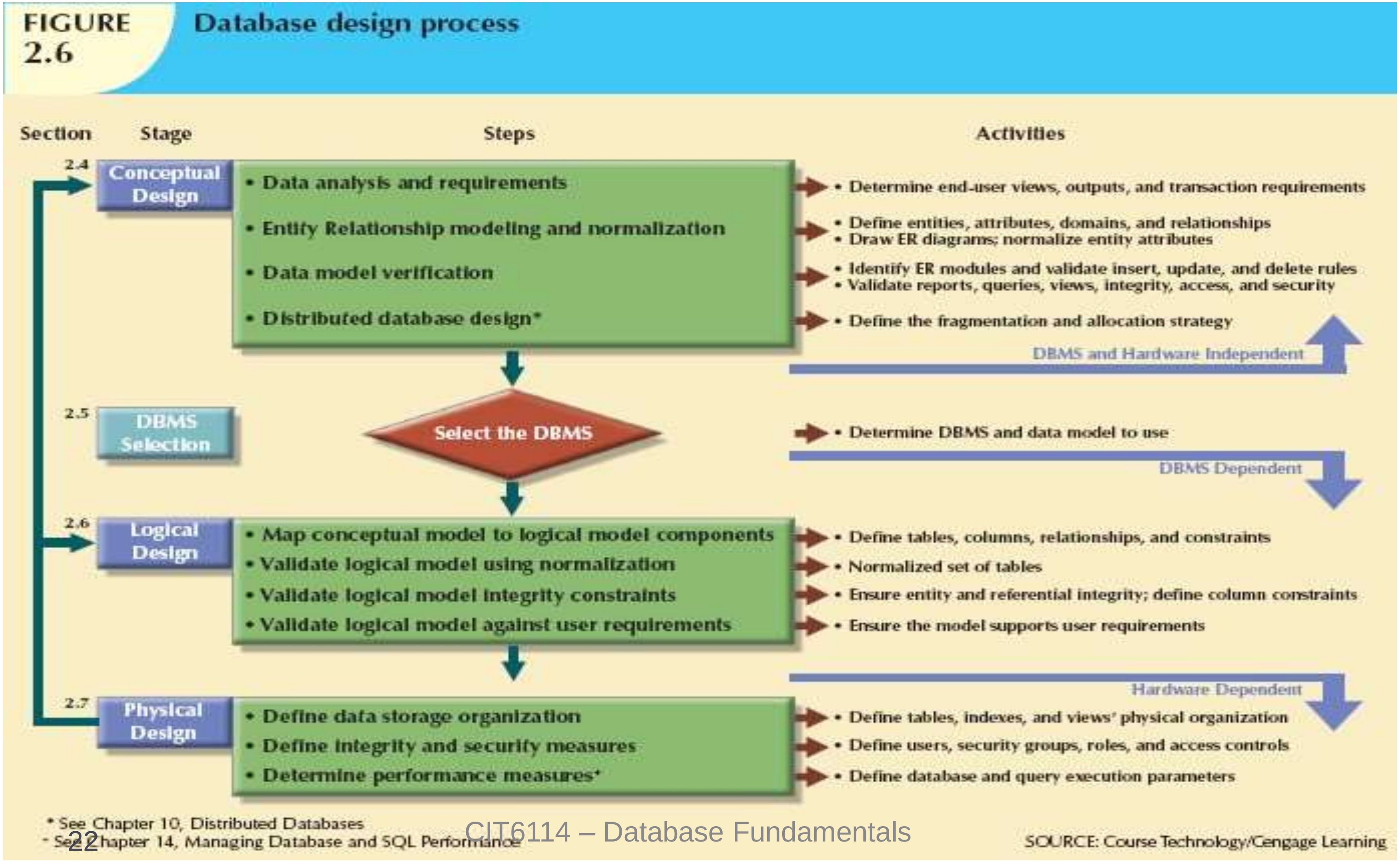


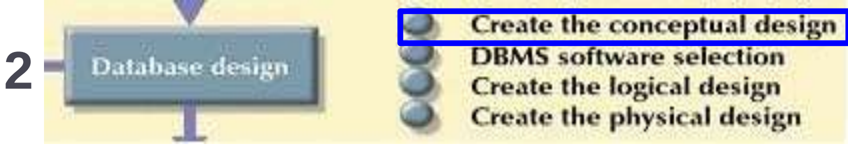
FIGURE 2.6 Database design process

Section	Stage	Steps	Activities
2.4	Conceptual Design	• Data analysis and requirements	• Determine end-user views, outputs, and transaction requirements
		• Entity Relationship modeling and normalization	• Define entities, attributes, domains, and relationships • Draw ER diagrams; normalize entity attributes
		• Data model verification	• Identify ER modules and validate insert, update, and delete rules • Validate reports, queries, views, integrity, access, and security
		• Distributed database design*	• Define the fragmentation and allocation strategy
			DBMS and Hardware Independent
2.5	DBMS Selection	Select the DBMS	• Determine DBMS and data model to use
			DBMS Dependent
2.6	Logical Design	• Map conceptual model to logical model components	• Define tables, columns, relationships, and constraints
		• Validate logical model using normalization	• Normalized set of tables
		• Validate logical model integrity constraints	• Ensure entity and referential integrity; define column constraints
		• Validate logical model against user requirements	• Ensure the model supports user requirements
			Hardware Dependent
2.7	Physical Design	• Define data storage organization	• Define tables, indexes, and views' physical organization
		• Define integrity and security measures	• Define users, security groups, roles, and access controls
		• Determine performance measures*	• Define database and query execution parameters

* See Chapter 10, Distributed Databases

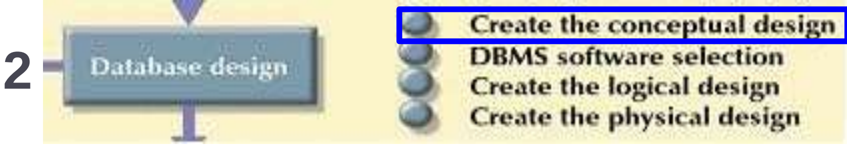
* See Chapter 14, Managing Database and SQL Performance

SOURCE: Course Technology/Cengage Learning


2
Database design
Create the conceptual design
DBMS software selection
Create the logical design
Create the physical design

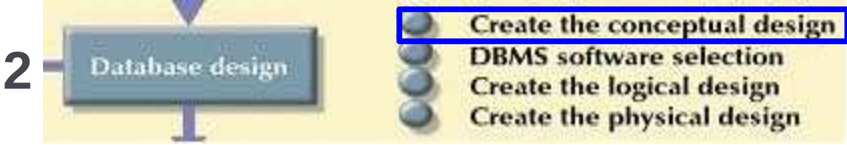
Conceptual Design

- Ensure that all data needed are in the **ER model**, and that all data in the model are needed
- Must take into account business rules (derived from description of operations)


2
Database design
Create the conceptual design
DBMS software selection
Create the logical design
Create the physical design

Conceptual Design

- 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*


2
Database design
Create the conceptual design
DBMS software selection
Create the logical design
Create the physical design

Conceptual Design

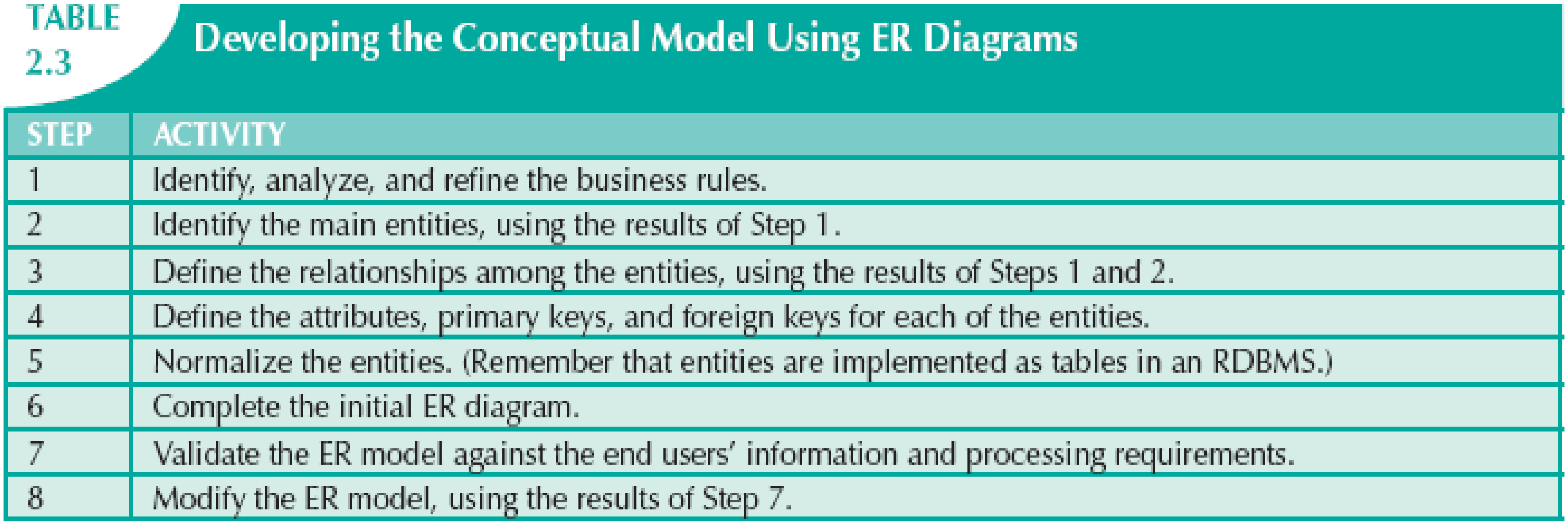
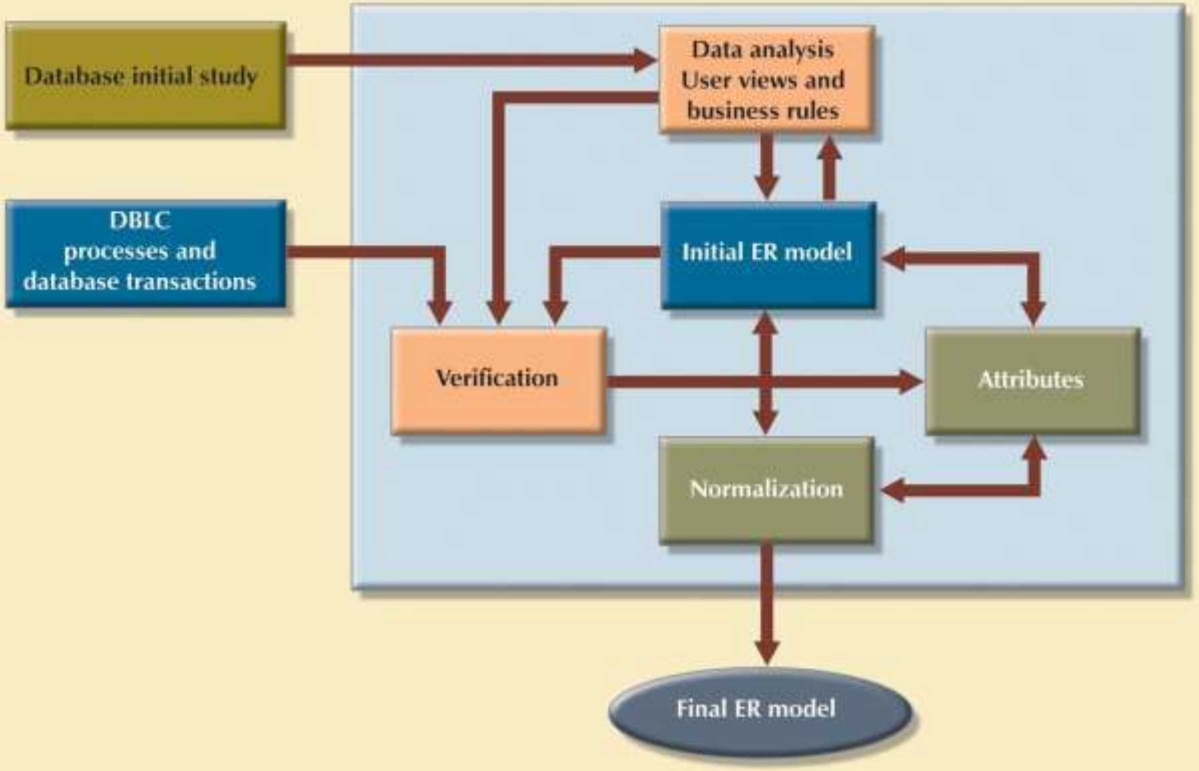


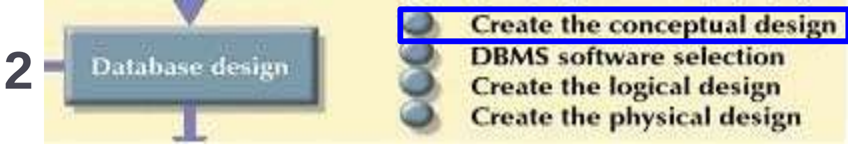
TABLE 2.3 Developing the Conceptual Model Using ER Diagrams

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. (Remember that entities are implemented as tables in an RDBMS.)
6	Complete the initial ER diagram.
7	Validate the ER model against the end users' information and processing requirements.
8	Modify the ER model, using the results of Step 7.

ER Modeling Is an Iterative Process Based on Many Activities

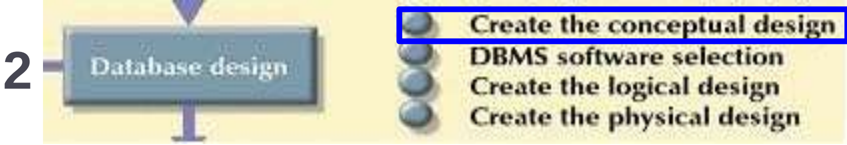
FIGURE 9.8 ER modeling is an iterative process based on many activities




2
Database design
Create the conceptual design
DBMS software selection
Create the logical design
Create the physical design

Conceptual Design

- 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


2
Database design
Create the conceptual design
DBMS software selection
Create the logical design
Create the physical design

The ER Model Verification Process

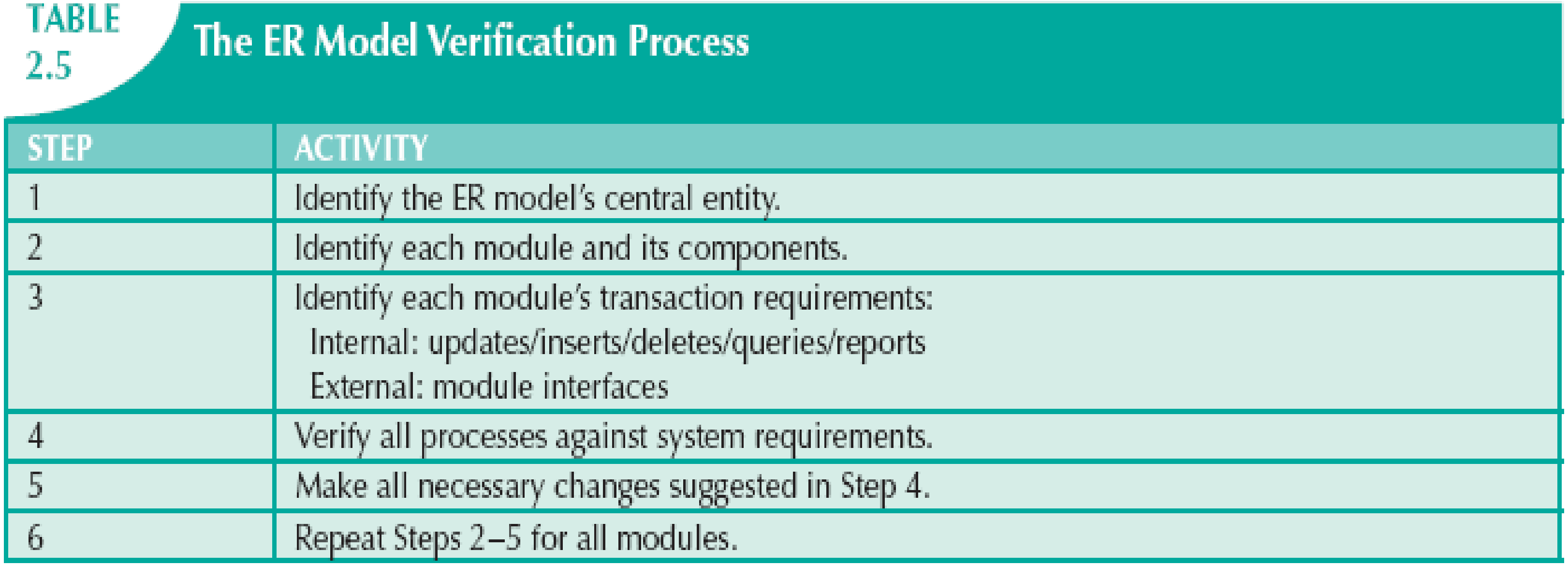
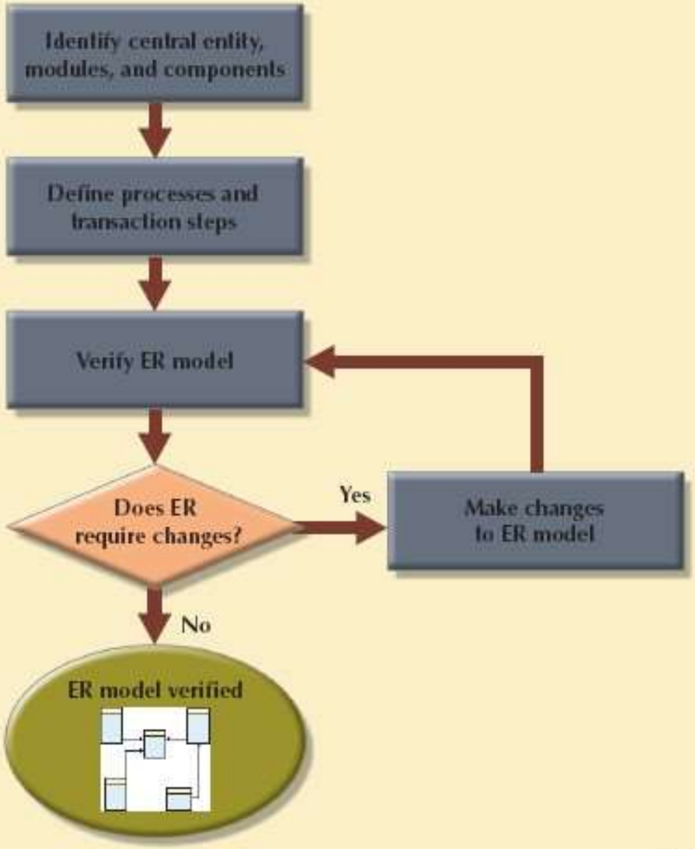


TABLE 2.5 The ER Model Verification Process

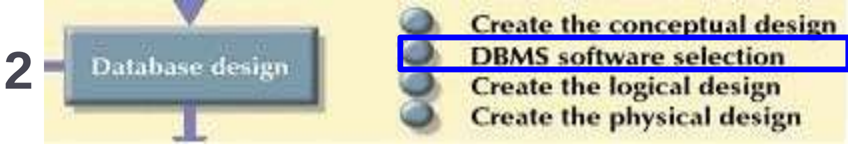
STEP	ACTIVITY
1	Identify the ER model's central entity.
2	Identify each module and its components.
3	Identify each module's transaction requirements: Internal: updates/inserts/deletes/queries/reports External: module interfaces
4	Verify all processes against system requirements.
5	Make all necessary changes suggested in Step 4.
6	Repeat Steps 2–5 for all modules.

The ER Model Verification Process

FIGURE 2.12 Iterative ER model verification process

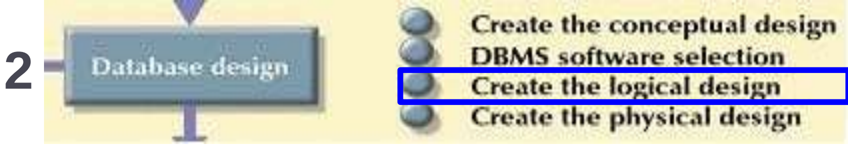

Identify central entity, modules, and components
Define processes and transaction steps
Verify ER model
Does ER require changes?
Yes
Make changes to ER model
No
ER model verified

SOURCE: Course Technology/Cengage Learning


2
Database design
Create the conceptual design
DBMS software selection
Create the logical design
Create the physical design

DBMS Software Selection

- Critical to the information system’s smooth operation
- Advantages and disadvantages should be carefully studied
  - *Cost*
  - *Features and tools*
  - *Underlying model*
  - *Portability*
  - *Hardware requirements*


2
Database design
Create the conceptual design
DBMS software selection
Create the logical design
Create the physical design

Logical Design

- 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

Logical Design

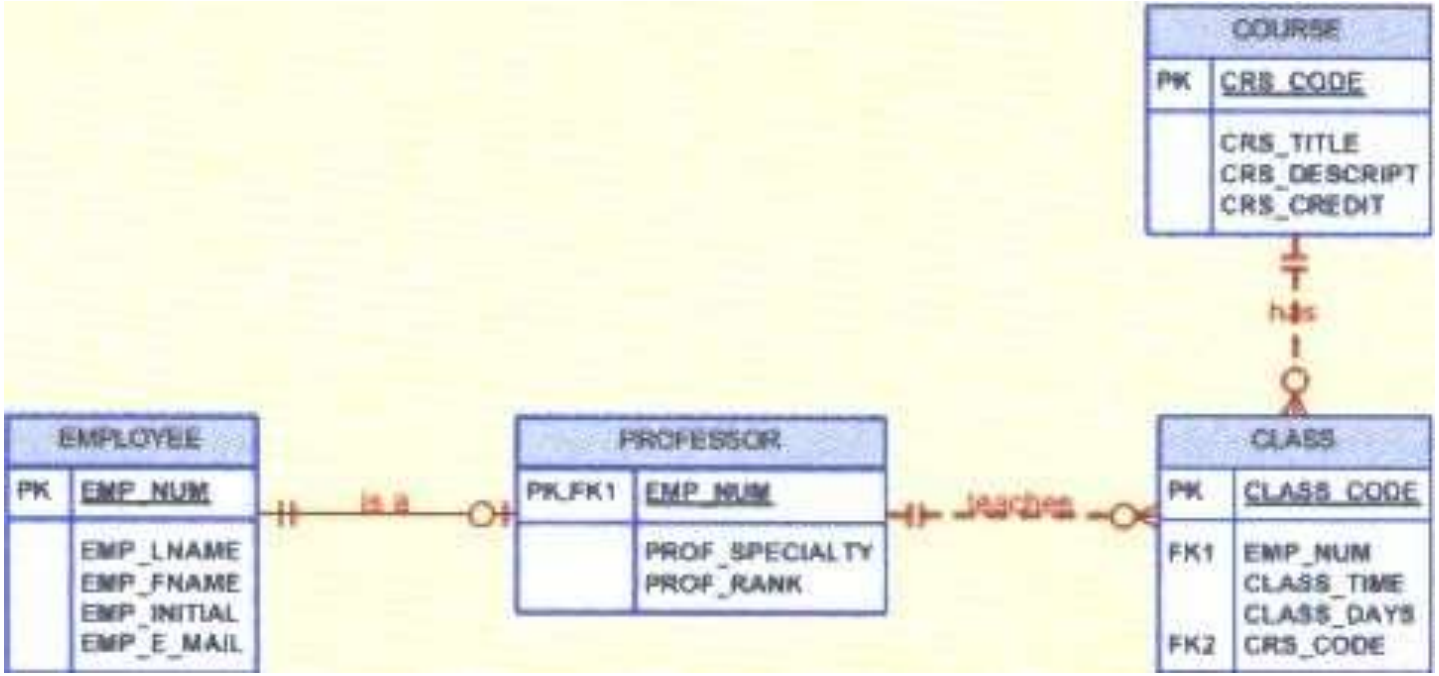
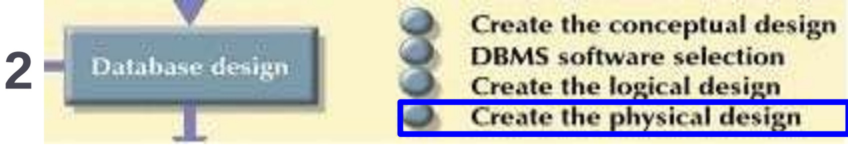
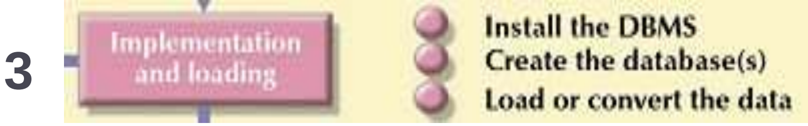


Figure 2.13: A simple conceptual model which had been converted into Logical Model


2
Database design
Create the conceptual design
DBMS software selection
Create the logical design
Create the physical design

Physical Design

- Process of selecting data storage and data access characteristics of the database
- Particularly important in the older hierarchical and network models
- Becomes more complex when data are distributed at different locations



Implementation and Loading

- Install database
- Create database to house the end-user tables

Implementation and Loading Issues

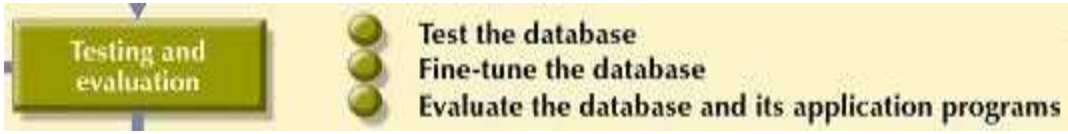
- Performance
  - *Varies according to the hardware and software environment used.*
- Security
  - *Data must be protected from access by unauthorized users*
    - Eg: password security, access right, audit trails
- Backup and Recovery
  - *Database can be subject to data loss through unintended data deletion and power outages*
    - Full backup – copied whole database
    - Differential backup – copied the last modifications done on database
    - Backup transaction log – copied the last transaction log operations made

Implementation and Loading Issues

- 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*

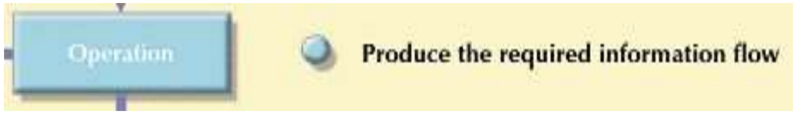
4



Testing and Evaluation

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

5



Operation

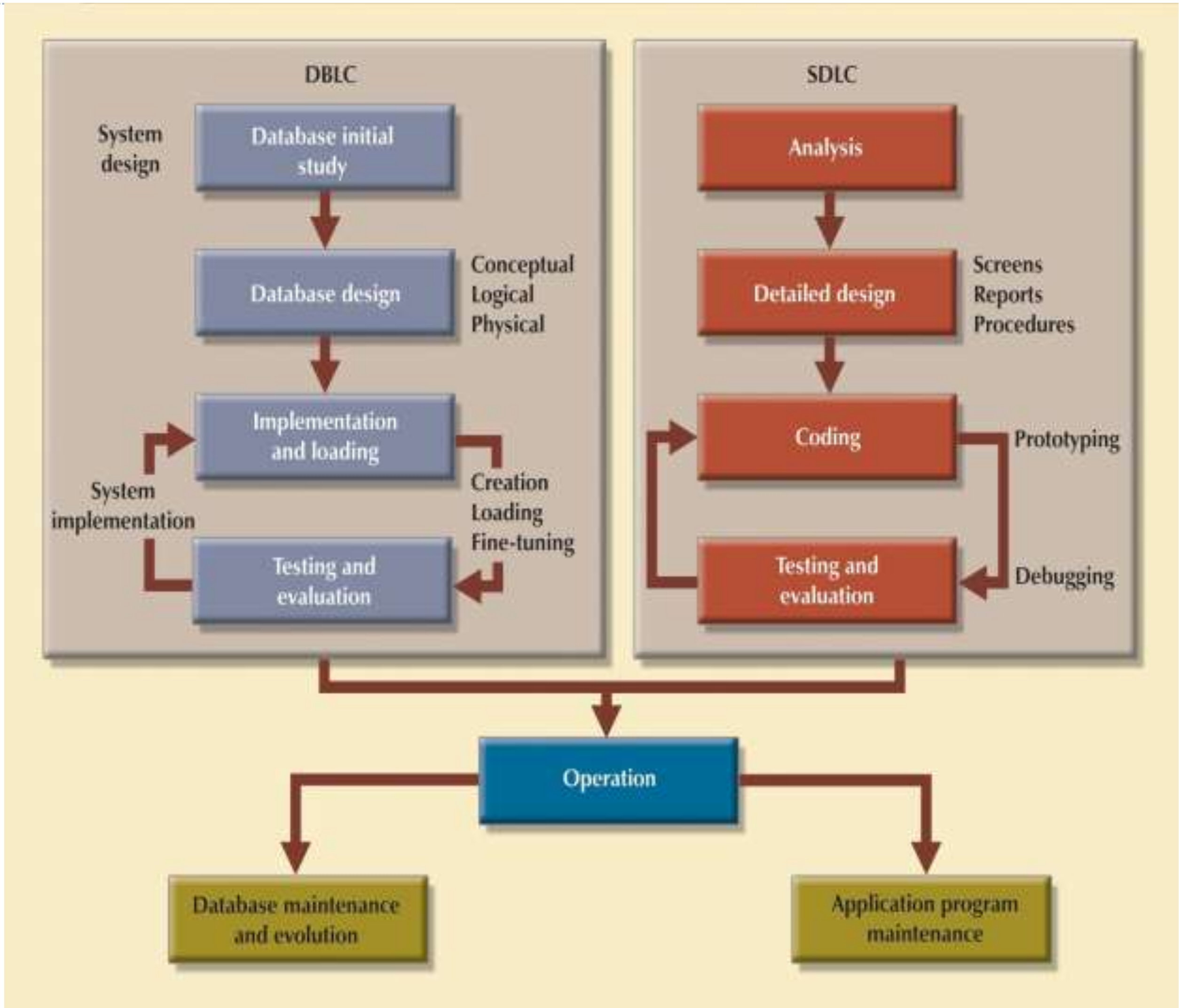
- Once the database has passed the evaluation stage, it is considered operational
- Beginning of the operational phase starts the process of system evolution

6
Maintenance and evaluation
Introduce changes
Make enhancements

Maintenance and Evolution

- Required periodic maintenance:
  - *Perfective maintenance (enhancement)*
  - *Corrective maintenance (recovery)*
  - *Adaptive maintenance (changes in env.)*
- Assignment of access permissions and their maintenance for new and old users
- Periodic security audits (DB access statistics)
- Periodic system-usage summaries

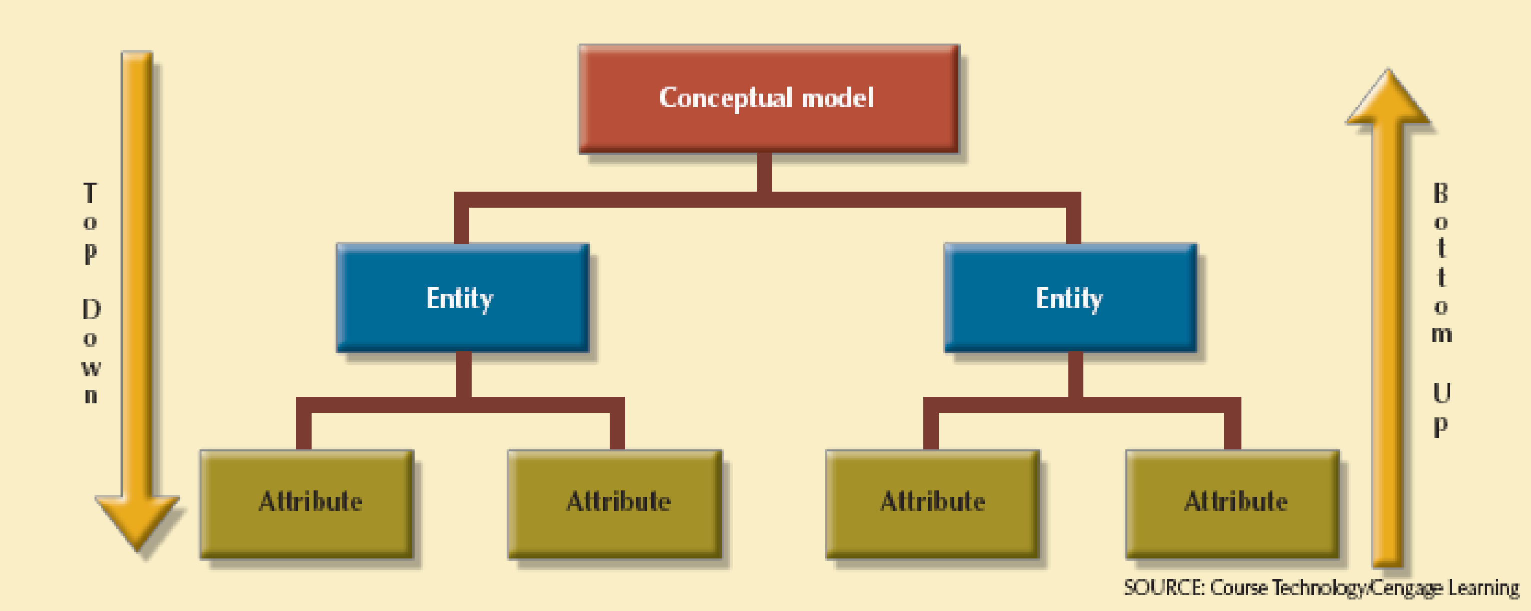
Parallel Activities in the DBLC and the SDLC



Database Design Strategies

- Two classical approaches to database design:
  - *Top-down design*
  - *Bottom-up design*

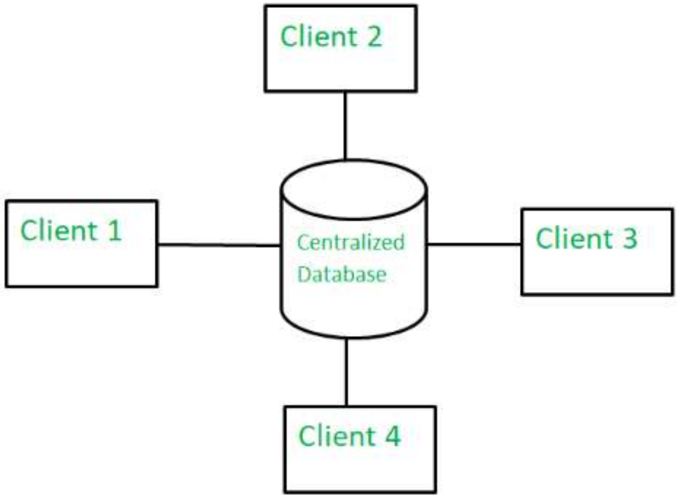
FIGURE 2.14 Top-down vs. bottom-up design sequencing

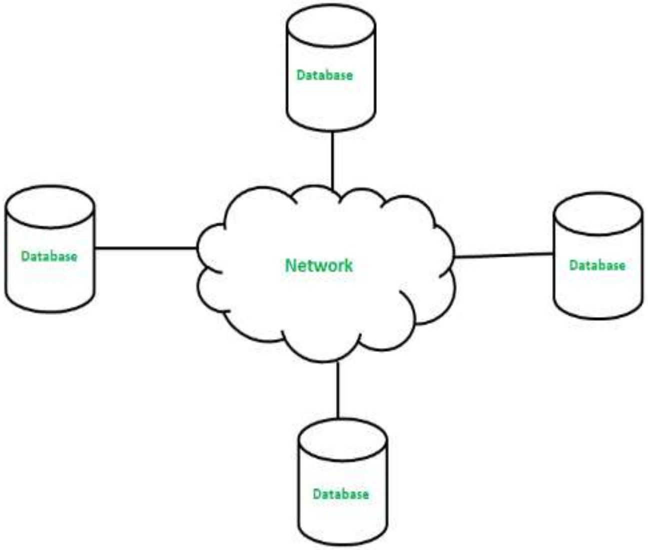


SOURCE: Course Technology/Cengage Learning

Centralized vs. Decentralized Design

- Database design may be based on two very different design philosophies:
  1. *Centralized design*
  2. *Decentralized design*





Summary

- Information system facilitates transformation of data into information
  - Manages both data and information
- SDLC traces history (life cycle) of an application within the information system
- DBLC describes history of database within the information system
- Database design and implementation process moves through series of well-defined stages