Chapter 6
Database Design

Objectives

In this chapter, you will learn:

- That successful database design must reflect the information system of which the database is a part
- Systems Development Life Cycle (SDLC)
- Database Life Cycle (DBLC)
- 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, procedures
- **Systems analysis**
  - Process that establishes need for and extent of information system
- **Systems development**
  - Process of creating information system

The Information System (cont'd.)

- Applications
  - Transform data into information that forms basis for decision making
  - Usually produce the following:
    - Formal report
    - Tabulations
    - Graphic displays
  - Composed of the following two parts:
    - Data
    - Code: program instructions

FIGURE 9.1 Generating information for decision making

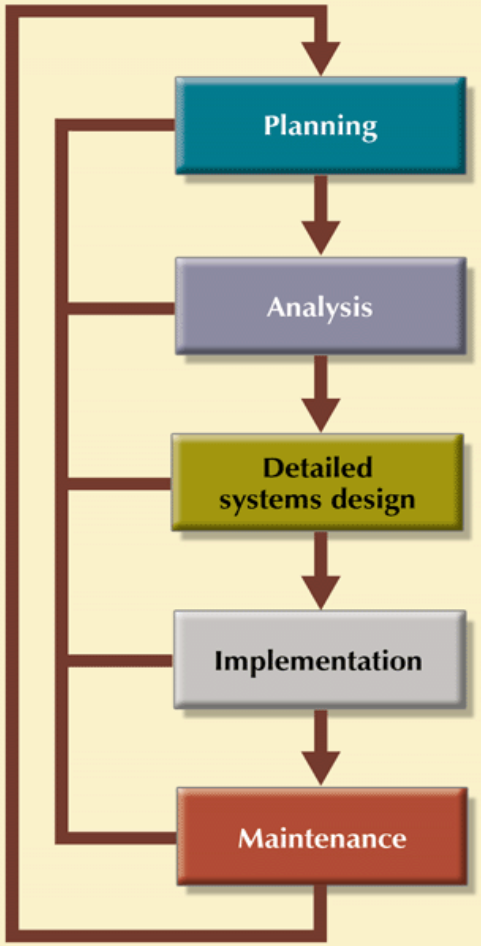
The Information System (cont'd.)

- Performance depends on three factors:
  - Database design and implementation
  - Application design and implementation
  - Administrative procedures
- **Database development**
  - Process of database design and implementation
  - Implementation phase includes:
    - Creating database storage structure
    - Loading data into the database
    - Providing for data management

The Systems Development Life Cycle (SDLC)

- Traces history (life cycle) of information system
- Database design and application development mapped out and evaluated
- Divided into following five phases:
  - Planning
  - Analysis
  - Detailed systems design
  - Implementation
  - Maintenance
- Iterative rather than sequential process

FIGURE 9.2 The Systems Development Life Cycle (SDLC)



Phase	Action(s)	Section
Planning	Initial assessment Feasibility study	9.2.1
Analysis	User requirements Existing system evaluation Logical system design	9.2.2
Detailed systems design	Detailed system specification	9.2.3
Implementation	Coding, testing, and debugging Installation, fine-tuning	9.2.4
Maintenance	Evaluation Maintenance Enhancement	9.2.5

Planning

- General overview of company and objectives
- Assessment of flow-and-extent requirements
  - Should the existing system be continued?
  - Should the existing system be modified?
  - Should the existing system be replaced?
- Study and evaluate alternate solutions
  - Technical aspects of hardware and software requirements
  - System cost
  - Operational 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:
  - Better understanding of:
    - System's functional areas
    - Actual and potential problems
    - Opportunities

Analysis-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 laid out for conversion from old to new system
- Training principles and methodologies are also planned
  - Submitted for management approval

Implementation

- Hardware, DBMS software, and application programs are installed
  - Database design is implemented
- Cycle of coding, testing, and debugging continues until database is ready for delivery
- Database is created and system is customized
  - Creation of tables and views
  - User authorizations

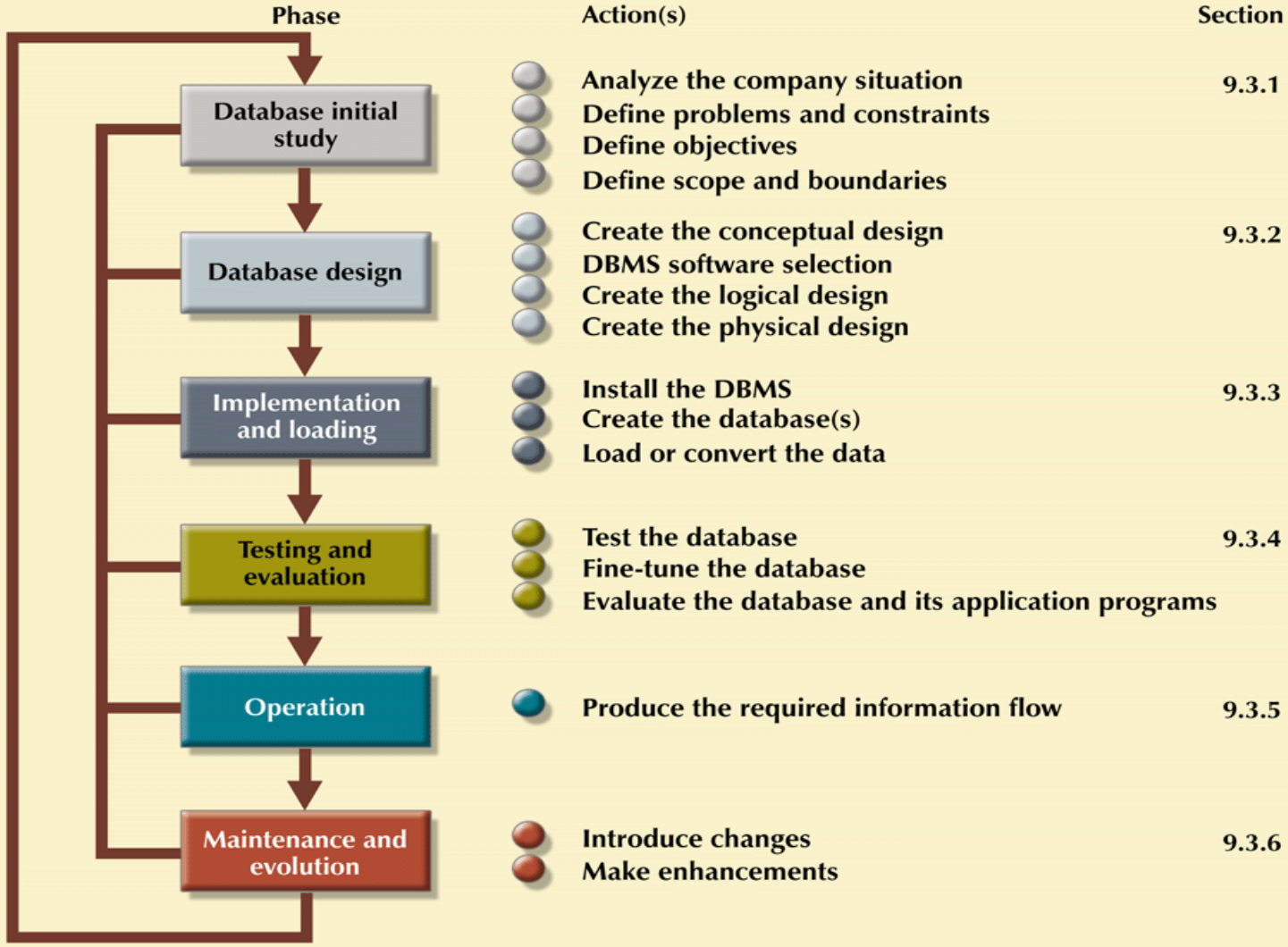
Maintenance

- Three types of maintenance activity:
  - Corrective maintenance (response to system error)
  - Adaptive maintenance(due to changes in business envi)
  - Perfective maintenance(to enhance the system)
- **Computer-aided systems engineering (CASE)**
  - Produce better systems within reasonable amount of time and at reasonable cost
  - CASE-produced applications are structured, documented, and standardized

The Database Life Cycle (DBLC)

- Six phases:
  - Database initial study
  - Database design
  - Implementation and loading
  - Testing and evaluation
  - Operation
  - Maintenance and evolution

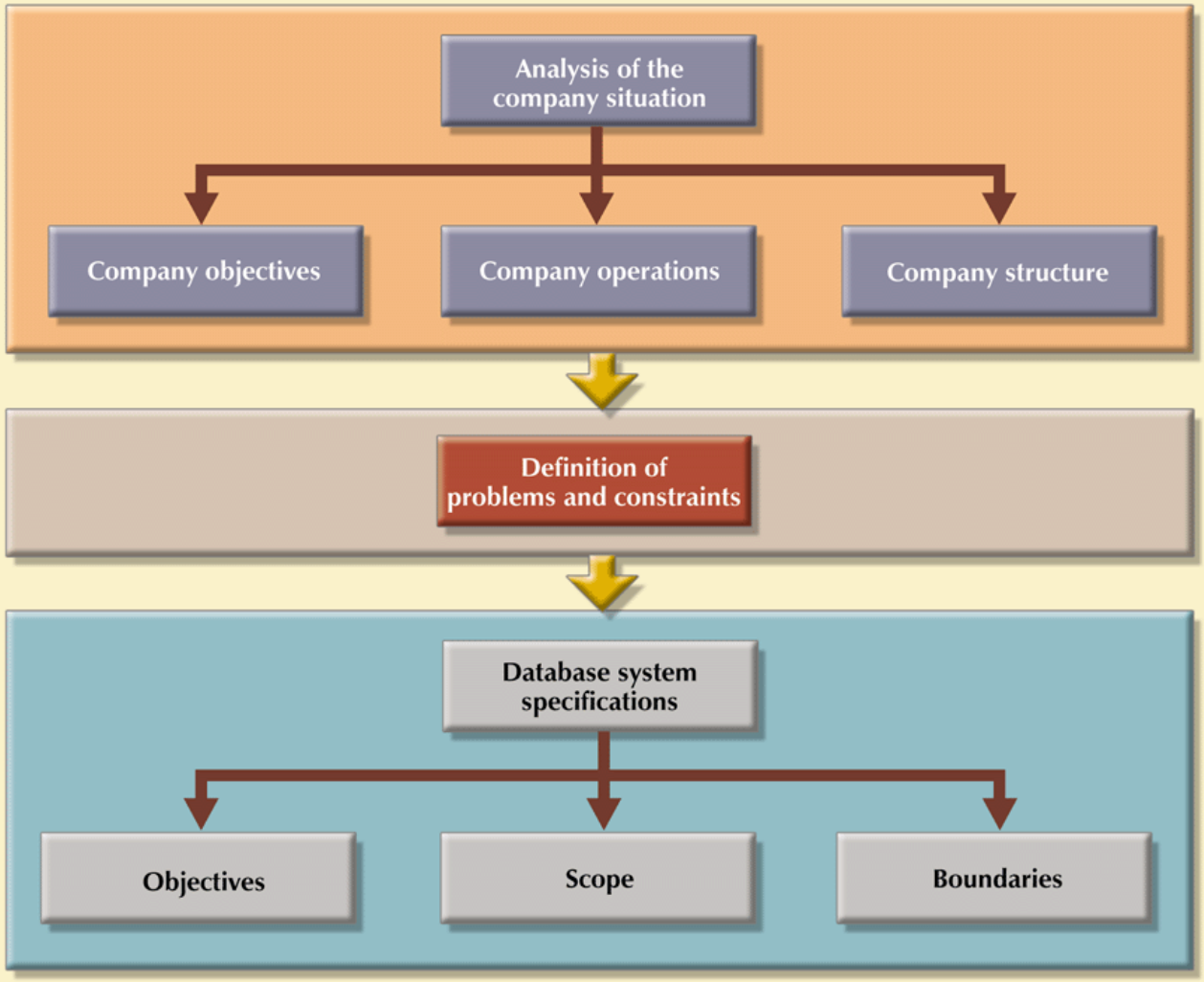
FIGURE 9.3 The Database Life Cycle (DBLC)



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

FIGURE 9.4 A summary of activities in the database initial study



The Database Initial Study (cont'd.)

- Analyze the company situation
  - General conditions in which company operates, its organizational structure, and its mission
  - Discover what company's operational components are, how they function, and how they interact

The Database Initial Study (cont'd.)

- Define problems and constraints
  - Formal and informal information sources
  - Finding precise answers is important
  - Accurate problem definition does not always yield a solution

The Database Initial Study (cont'd.)

- Database system objectives must correspond to those envisioned by end users
  - What is proposed system's initial objective?
  - Will system interface with other systems in the company?
  - Will system share data with other systems or users?

- **Scope**: extent of design according to operational requirements→define data structure,type and number of entities, physical size of database and so on.
- **Boundaries**: limits external to system (existing hardware+software,

Database Design

- Necessary to concentrate on data characteristics required to build database model
- Two views of data within system:
  - Business view
    - Data as information source
  - Designer's view
    - Data structure, access, and activities required to transform data into information

FIGURE 9.5 Two views of data: business manager and database designer

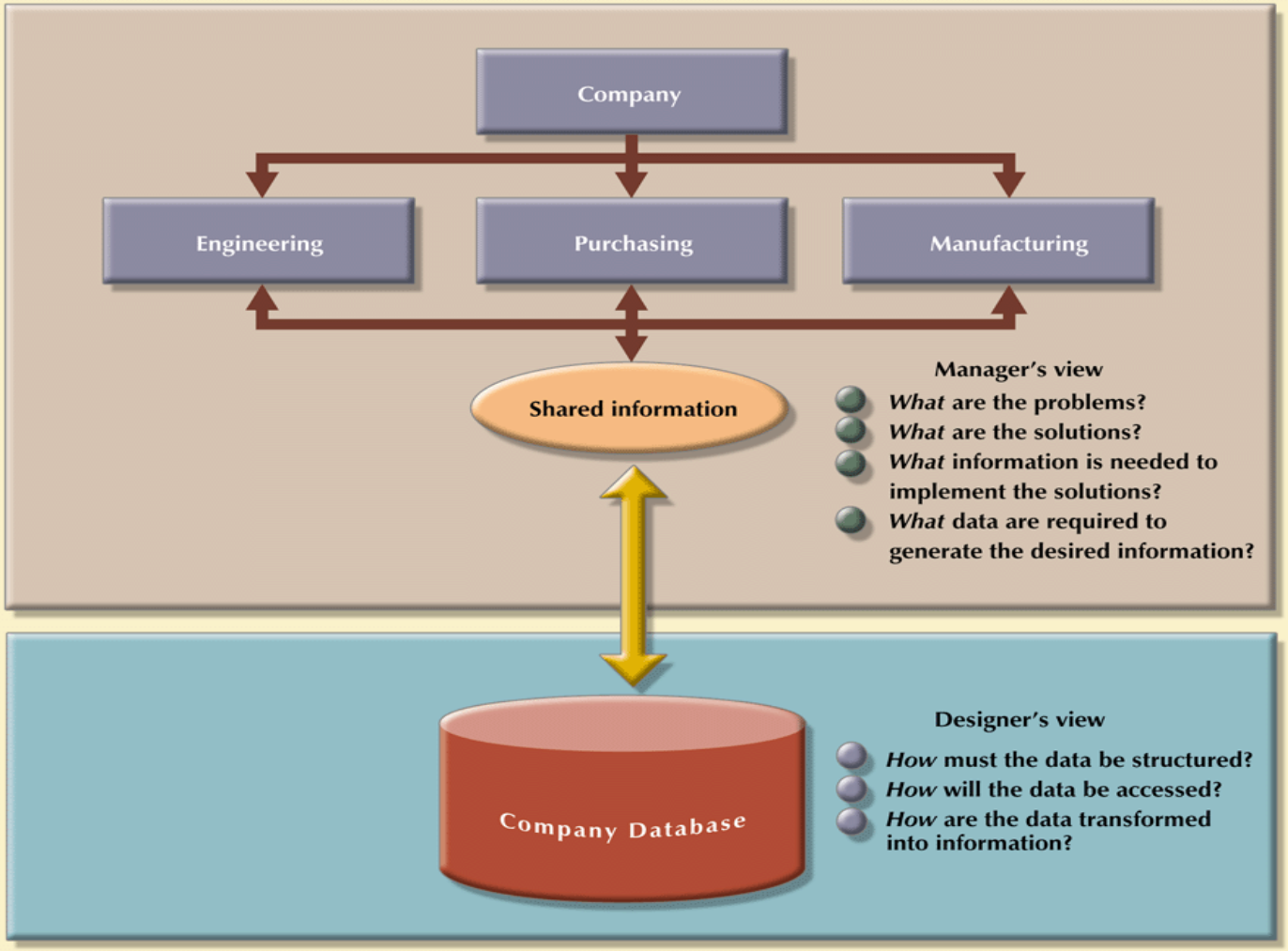
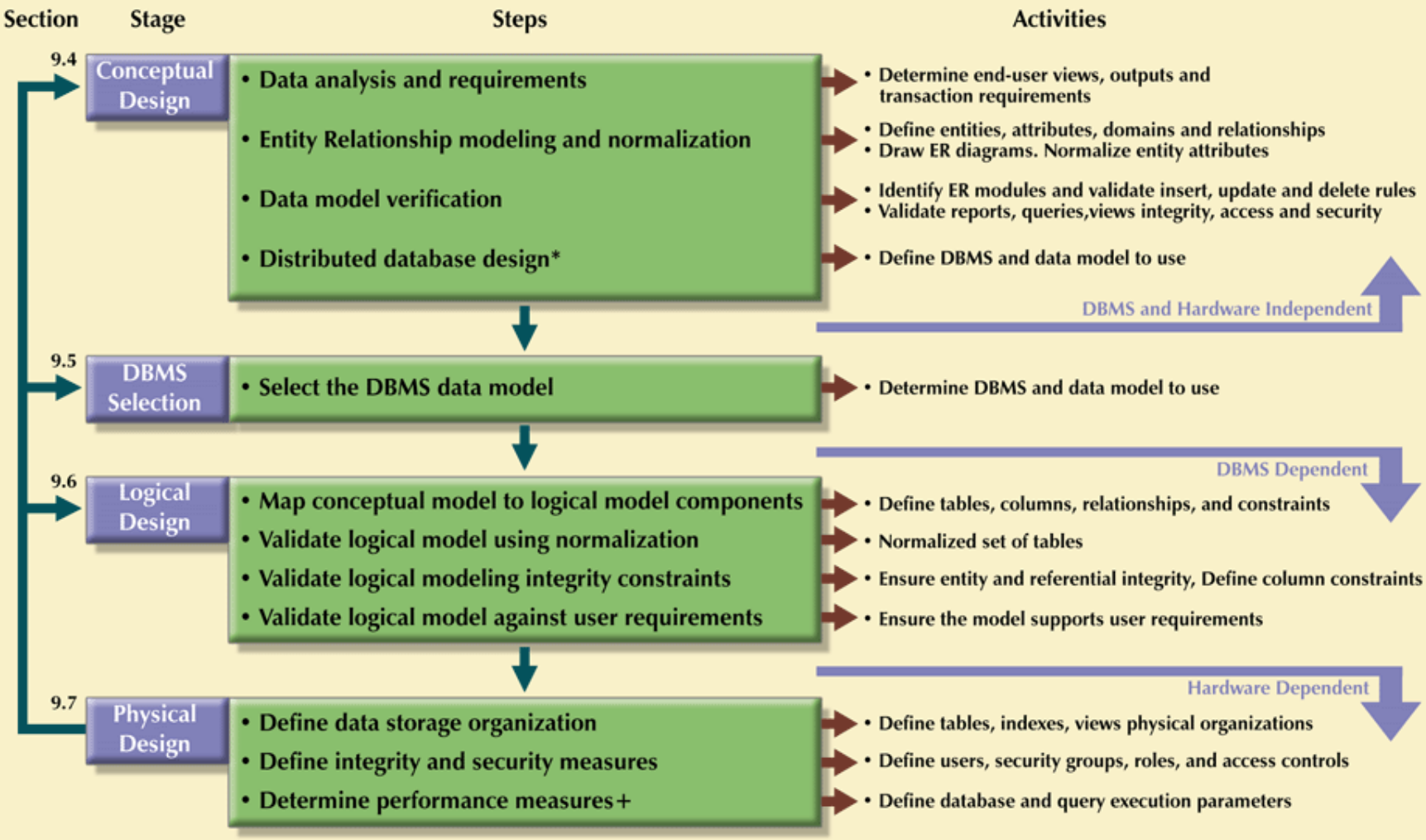


FIGURE 9.6 Database design process



* See Chapter 12, Distributed Database Management Systems
\+ See Chapter 11, Database Performance Tuning and Query Optimization

Implementation and Loading

- Actually implement all design specifications from previous phase:
  - Install the DBMS
    - Virtualization: creates logical representations of computing resources independent of physical resources
  - Create the Database
  - Load or Convert the Data

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 or logical design
  - Upgrade software and/or hardware platform

Testing and Evaluation (cont'd.)

- Integrity
  - Enforced via proper use of primary, foreign key rules
- Backup and Recovery
  - **Full backup**(entire database)
  - **Differential backup** (only object that have been updated or modifies since the last full backup are backed up).
  - **Transaction log backup** (back up only the transaction log operations that are not reflected in a previous backup copy of the database)

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(enhancing performance, adding entities and attributes and so on)
  - Assignment of access permissions and their maintenance for new and old users
  - Generation of database access statistics
  - Periodic security audits
  - Periodic system-usage summaries(monthly, quarterly or yearly) for internal billing or budgeting purposes.

Database design process : Stage 1: Conceptual Design

- Data modeling creates an abstract database structure
  - Represents real-world objects
- Embodies 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.Data Analysis and Requirements

- Discover data element characteristics
  - Obtains characteristics from different sources
- Requires thorough understanding of the company's data types and their extent and uses
- Take into account business rules
  - Derived from **description of operations**

ii. Entity Relationship Modeling and Normalization

- Designer enforces standards in design documentation
  - Use of diagrams and symbols, documentation writing style, layout, other conventions
- Business rules must be incorporated into conceptual model
- ER model is a communications tool as well as design blueprint

TABLE 9.3 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.

iii. Data Model Verification

- Verified against proposed system processes
- Revision of original design
  - Careful reevaluation of entities
  - Detailed examination of attributes describing entities
- Define design's major components as modules:
  - **Module:** information system component that handles specific function

Data Model Verification (cont'd.)

TABLE 9.5 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.

iv. Distributed Database Design

- Portions of database may reside in different physical locations
  - Database fragment: subset of a database stored at a given location
- Processes accessing the database vary from one location to another
- Designer must also develop data distribution and allocation strategies

Stage 2: DBMS Software Selection

- Critical to information system's smooth operation
- Common factors affecting purchasing decisions:
  - Cost
  - DBMS features and tools
  - Underlying model
  - Portability
  - DBMS hardware requirements

Stage 3: Logical Design

- Map conceptual design to specific data model
- Still independent of physical-level details
- Requires all objects be mapped to specific constructs used by selected database software
  - Definition of attribute domains, design of required tables, access restriction formats
  - Tables must correspond to entities in conceptual design
- Translates software-independent conceptual model into software-dependent model

i. Map the Conceptual Model to the Logical Model

- Map the conceptual model to the chosen database constructs
- Five mapping steps involved:
  - Strong entities
  - Supertype/subtype relationships
  - Weak entities
  - Binary relationships
  - Higher degree relationships

ii. Validate the Logical Model Using Normalization

- Translation requires the definition of the attribute domains and appropriate constraints
- All defined constraints must be supported by the logical data model

iii. Validate the Logical Model against User Requirements

- Final step in the logical design process
- Validate all logical model definitions against all end-user data, transaction, and security requirements

Stage 4: 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
  - DBMS
- More complex when data are distributed

i. Define Data Storage Organization

- Designer must determine several attributes:
  - Data volume
  - Data usage patterns
- Which in turn influence:
  - Location and physical storage organization for each table
  - What indexes and the type of indexes to be used for each table
  - What views and the type of views to be used on each table

ii. Define Integrity and Security Measures

- Define user and security groups and roles
  - Database role: set of database privileges that could be assigned as a unit to a user or group
- Assign security controls
  - Specific access rights on database objects to a user or group of users

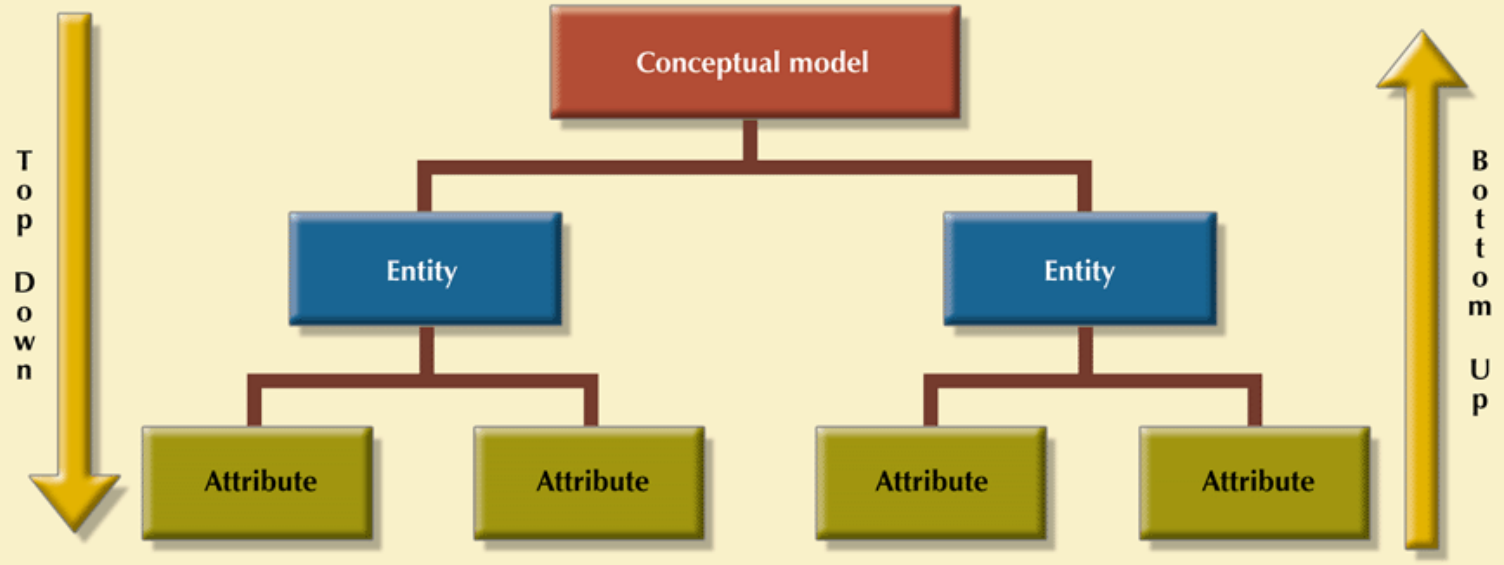
iii.Determine Performance Measures

- Performance can be affected by characteristics:
  - Storage media
  - Seek time
  - Sector and block (page) size
  - And more…
- Fine-tuning the DBMS and queries to ensure that they will meet end-user performance requirements

Database Design Strategies

- **Top-down design**
  - Identifies data sets
  - Defines data elements for each of those sets
    - Definition of different entity types
    - Definition of each entity's attributes
- **Bottom-up design**
  - Identifies data elements (items)
  - Groups them together in data sets

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



Centralized vs. Decentralized Design

- **Centralized design**
  - When data component is composed of small number of objects and procedures
  - Typical of small systems
- **Decentralized design**
  - Data component has large number of entities
  - Complex relations on which complex operations are performed
  - Problem is spread across several operational sites

FIGURE 9.15 Centralized design

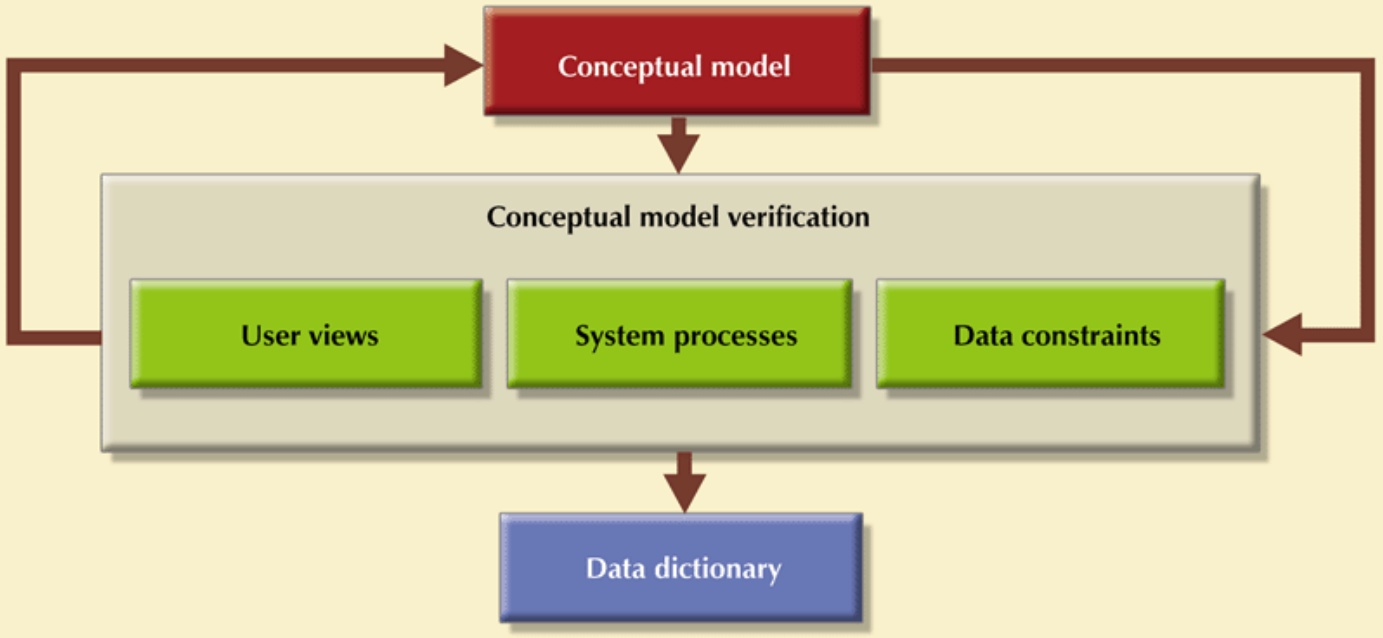
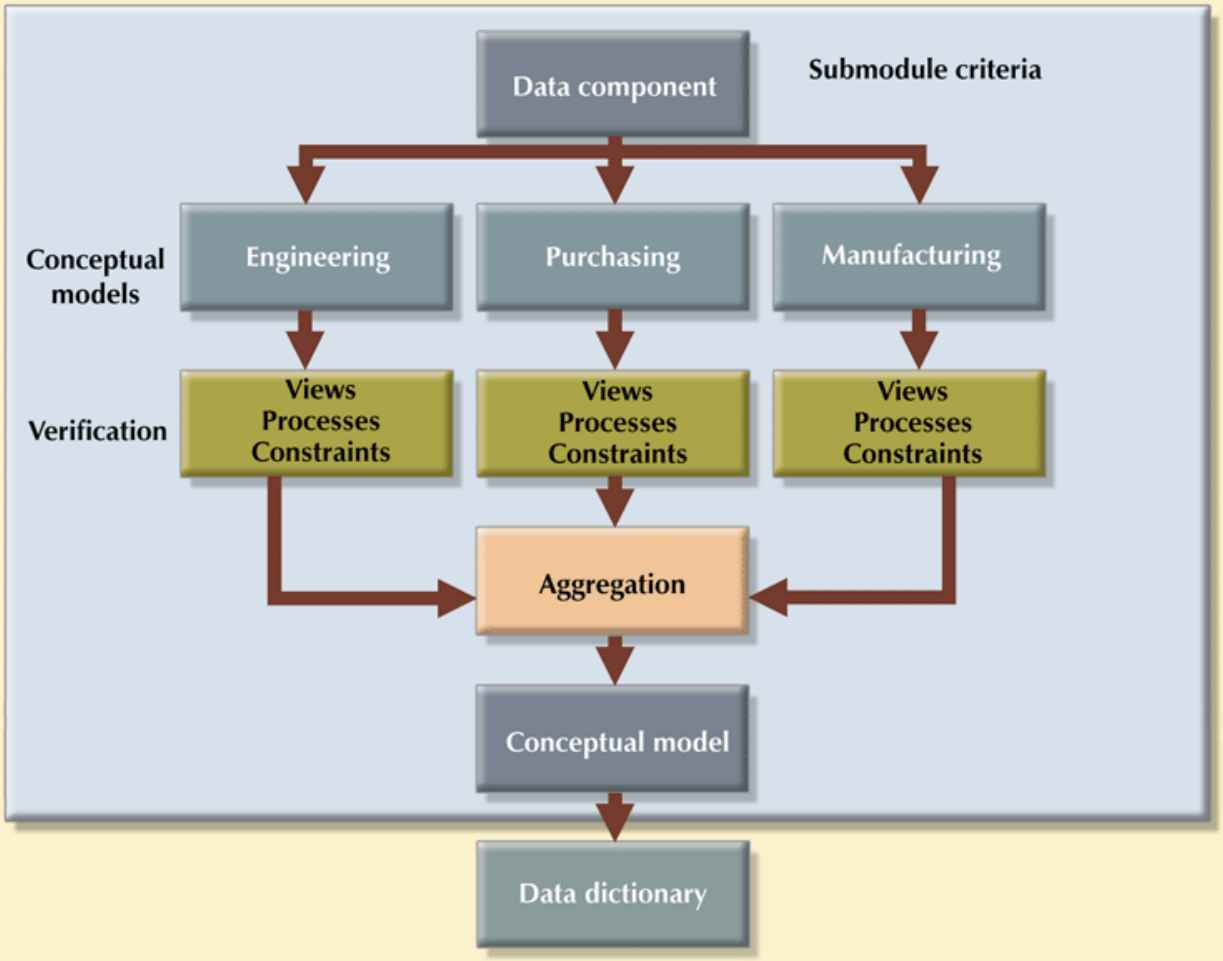


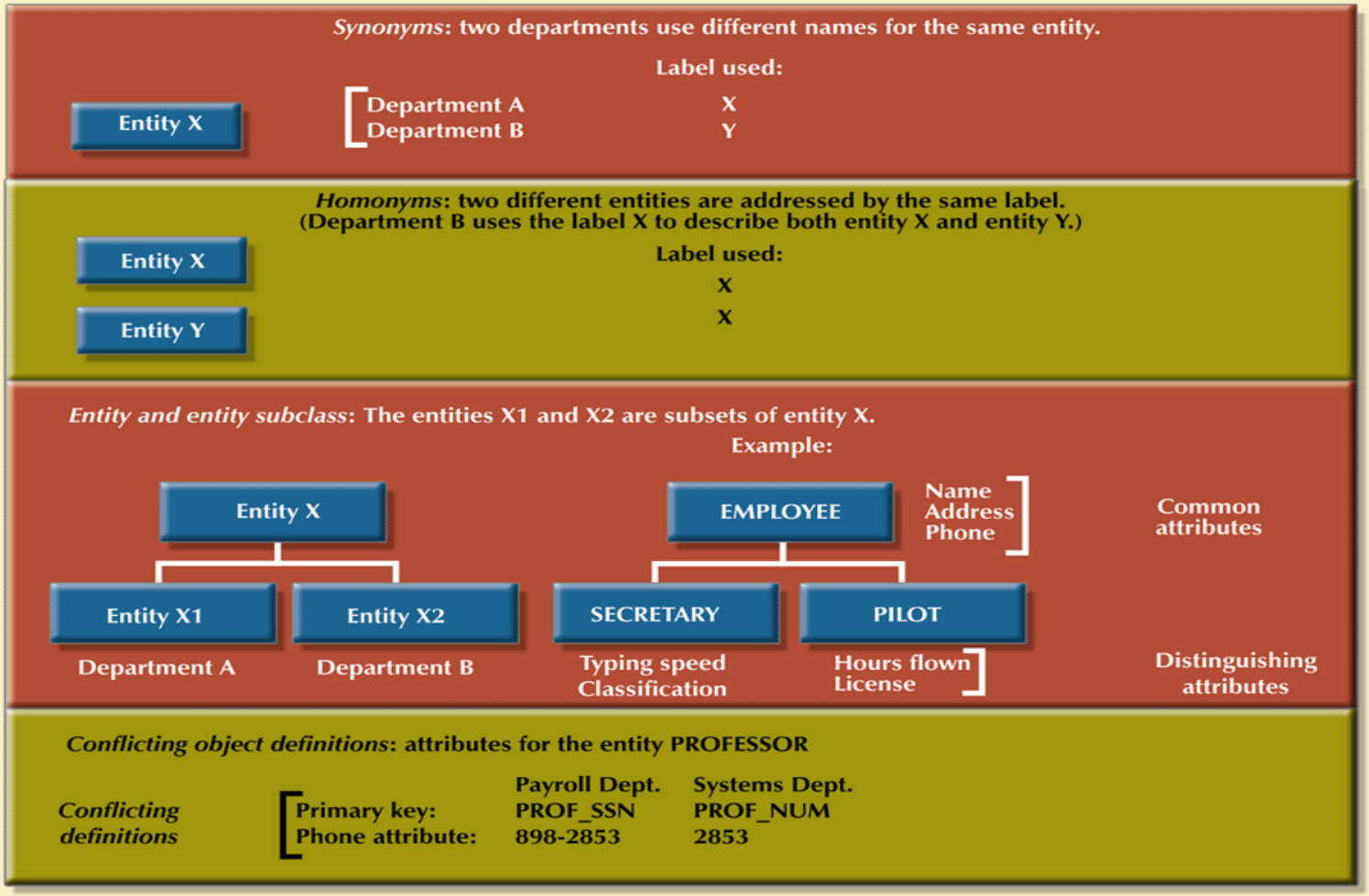
FIGURE 9.16 Decentralized design



Centralized vs. Decentralized Design (cont'd.)

- All modules are integrated into one model
- Aggregation problems to be addressed:
  - Synonyms and homonyms
  - Entity and entity subtypes
  - Conflicting object definitions

FIGURE 9.17 Summary of aggregation problems



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

Summary (cont'd.)

- Database design and implementation process moves through series of well-defined stages
- Conceptual design subject to several variations:
  - Top-down vs. bottom-up
  - Centralized vs. decentralized