Chapter 9

Database Design

Learning Objectives (1 of 2)

- 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)

Learning Objectives (2 of 2)

- In this chapter, you will learn:
  - 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 versus 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 - 2

- 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

NEU CPS Related Courses

- ALY 6060 – Decision Support and Business Intelligence
  - Introduces current and emerging business analytical concepts and information technologies to support decision making and business intelligence.
  - Commercial decision support systems in various application areas are introduced and discussed using case studies (CRM, SFDC)
  - Introduces business intelligence technology and applications
  - Offers students an opportunity to gain hands-on experience using business intelligence tools, including Tableau or QlikView.

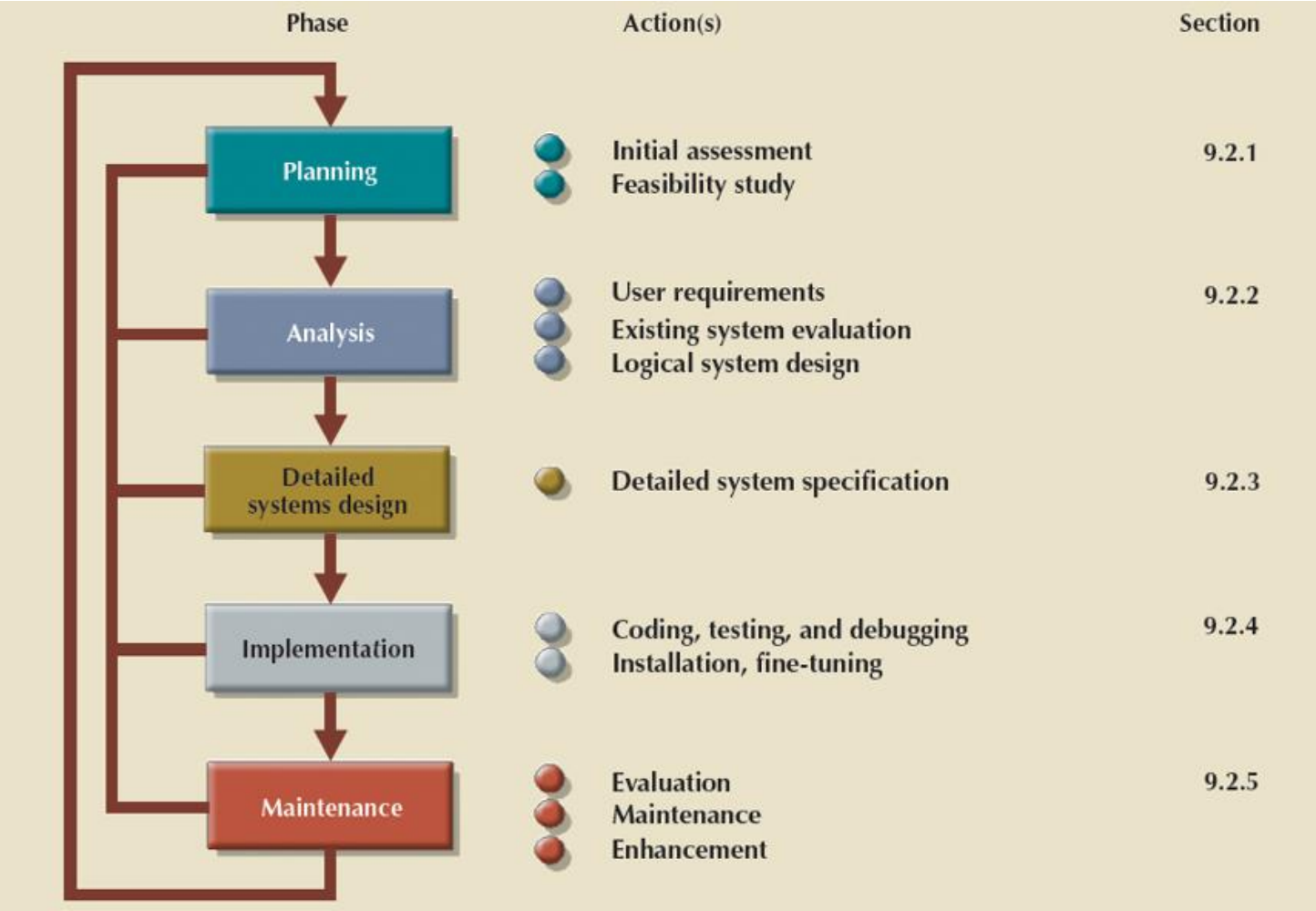
Performance Factors of an Information System

- Define “Performance”
- Depends on three factors:
  - Database design and implementation
  - Application design and implementation
  - Administrative procedures
- **Database development**: Process of database design and its implementation
- Implementation phase includes:
  - Create database storage structure
  - Loading data into the data
  - Providing for data management

Systems Development Life Cycle (SDLC)

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

Figure 9.2 - The Systems Development Life Cycle (SDLC)



Planning

- General overview of company and objectives
- Assessment to answer important questions:
  - Should the existing system be continued?
  - Should the existing system be modified
  - Should the existing system be replaced?
- Study and evaluate alternative solutions
  - Technical aspects of hardware and software requirements (CPU, RAM, storage space, redundancy, availability, geo-location, RTO and RPO)
  - System Cost
  - Operational Cost

Analysis

- Problems defined during planning phase are examined in great detail during analysis
- Thorough audit of user requirements
- Existing hardware and software systems are studied
- Goal: Better understanding of:
  - System's functional areas and strengths
  - Actual problems – user's "pain points"
  - Potential problems
  - Opportunities – e.g. outsource?

Detailed System Design

- Designer completes design of system's processes
- Includes all necessary technical specifications
- Steps laid out for conversion from old to new system
  - Typically called a "Statement of Work" (SOW) if being completed by a 3rd party or consultant
- Training principles and methodologies are also planned
- Costs, timeline, milestones, deliverables, etc all must be approved by management and stakeholders

Implementation

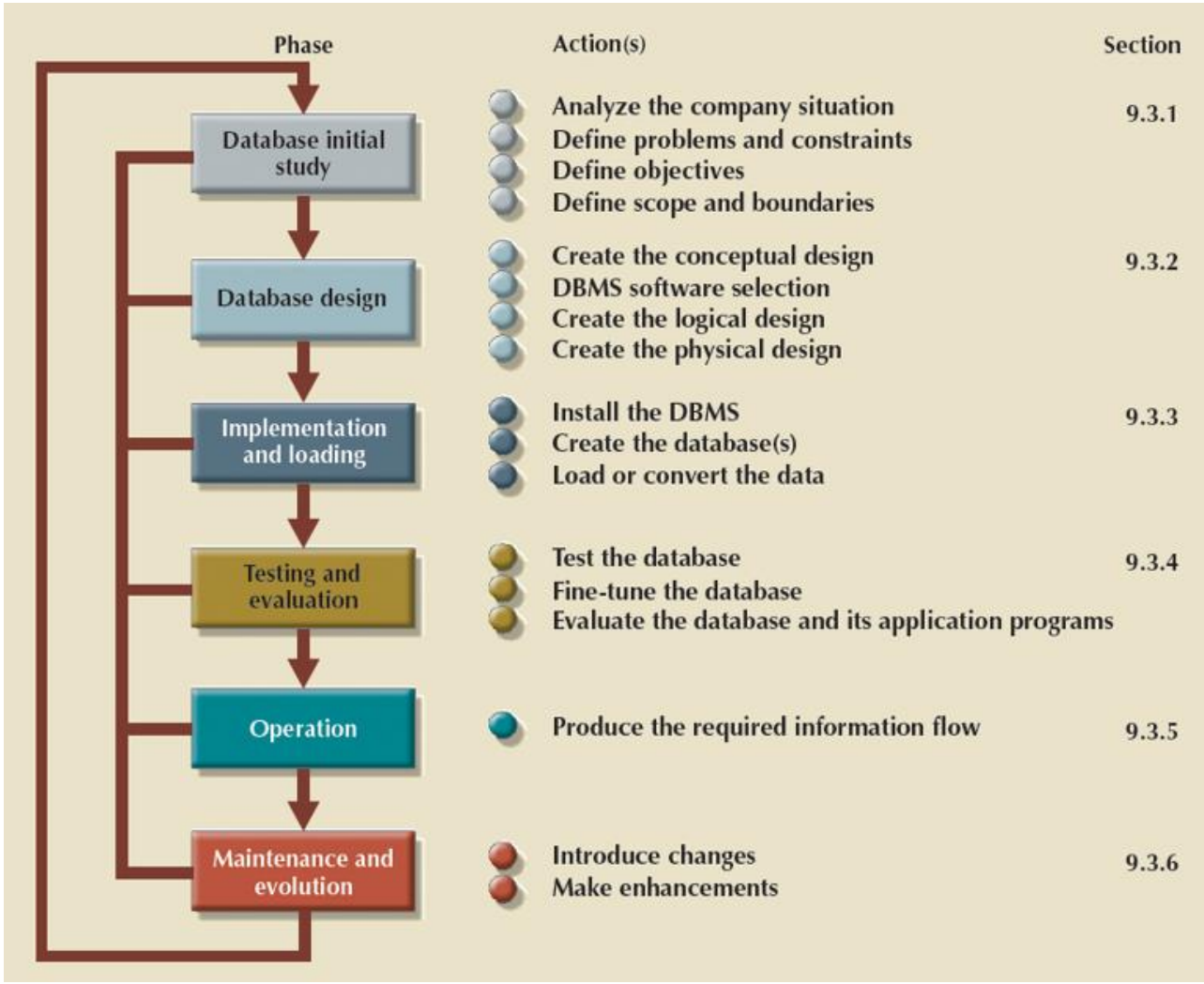
- Hardware is installed (on-prem) or provisioned (in cloud)
- DBMS software is installed – typically in partnership with DMBS vendor's Professional Services team
- Cycle of coding, testing, and debugging continues until system is ready for delivery
- Database is created and system is customized
  - Created of tables and view
  - User permissions and "Role Based Access Control" (RBAC)

Maintenance

- Three types of maintenance activity:
  - Corrective Maintenance
    - In response to system errors or issues
    - Typically “service packs” or vulnerability patches performed during “maintenance windows” after going through testing and “Change Control” processess
  - Adaptive Maintenance
    - In response to changes in business needs
    - E.g. Increased storage, RAM, or CPU
  - Perfective Maintenance
    - Upgrades, add-ons, or enhancements

The Database Life Cycle (DBLC)

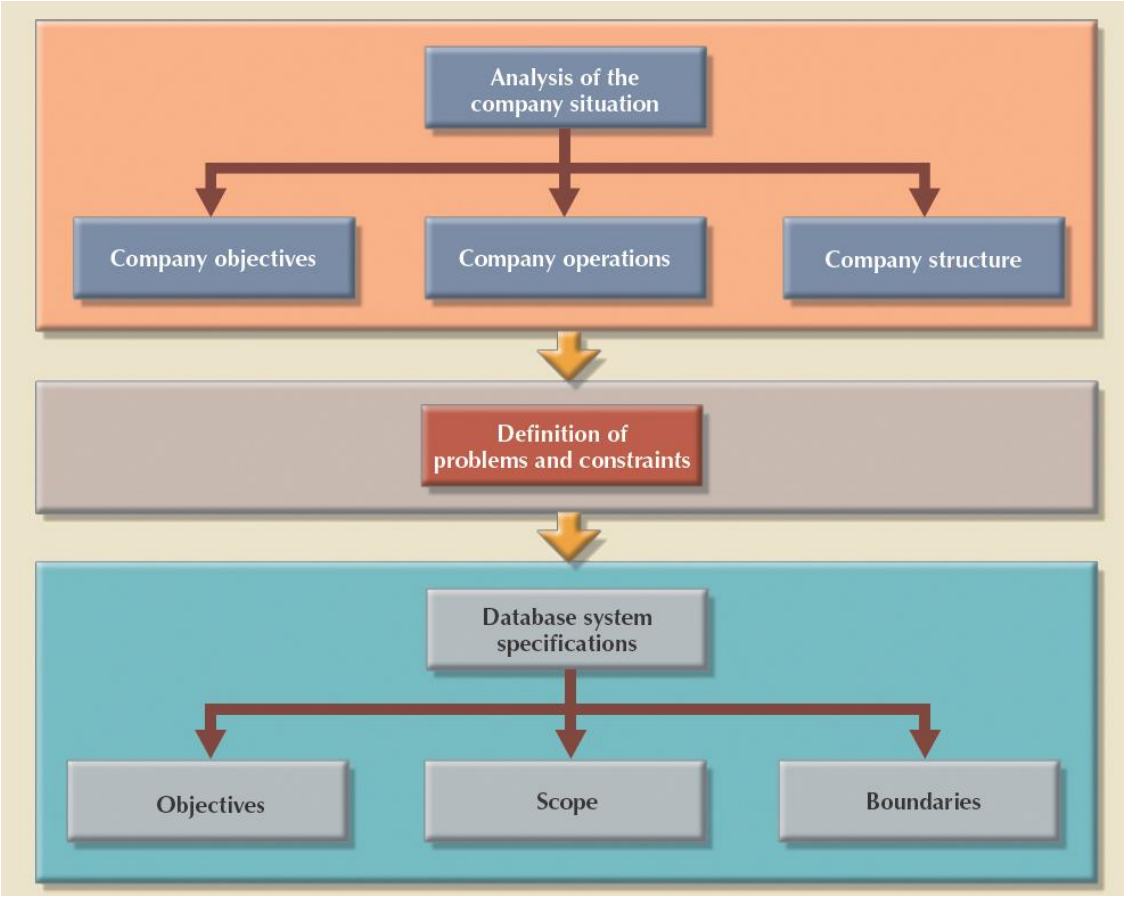
- Six Phases:
  - Database initial study
  - Database design
  - Implementation and Loading
  - Testing and Evaluation
  - Operation
  - Maintenance and Evolution



Purpose of Database Initial Study

- Analyze company situation
- Define problems and constraints
  - Formal and informal information sources
  - Finding precise answers is important
  - Accurate problem definition does not always yield a solution
- Define objectives
- Define scope – extent of design according to operational req's
- Define boundaries – limits that are external to the system (e.g. cost)
- Define Availability – How many 9's? (99.9%, 99.99%, 99.999%)

Figure 9.4 - A Summary of Activities in the Database Initial Study



Database Design

- Supports company's operations and objectives
- Most critical phase
  - Ensures final product meets user and system requirements
- Points for examining completion procedures
  - Data component is an element of whole system
  - System analysts/programmers design procedures to convert data into information
  - Database design is an iterative process

Figure 9.5 - Two Views of Data: Business Manager and Database Designer

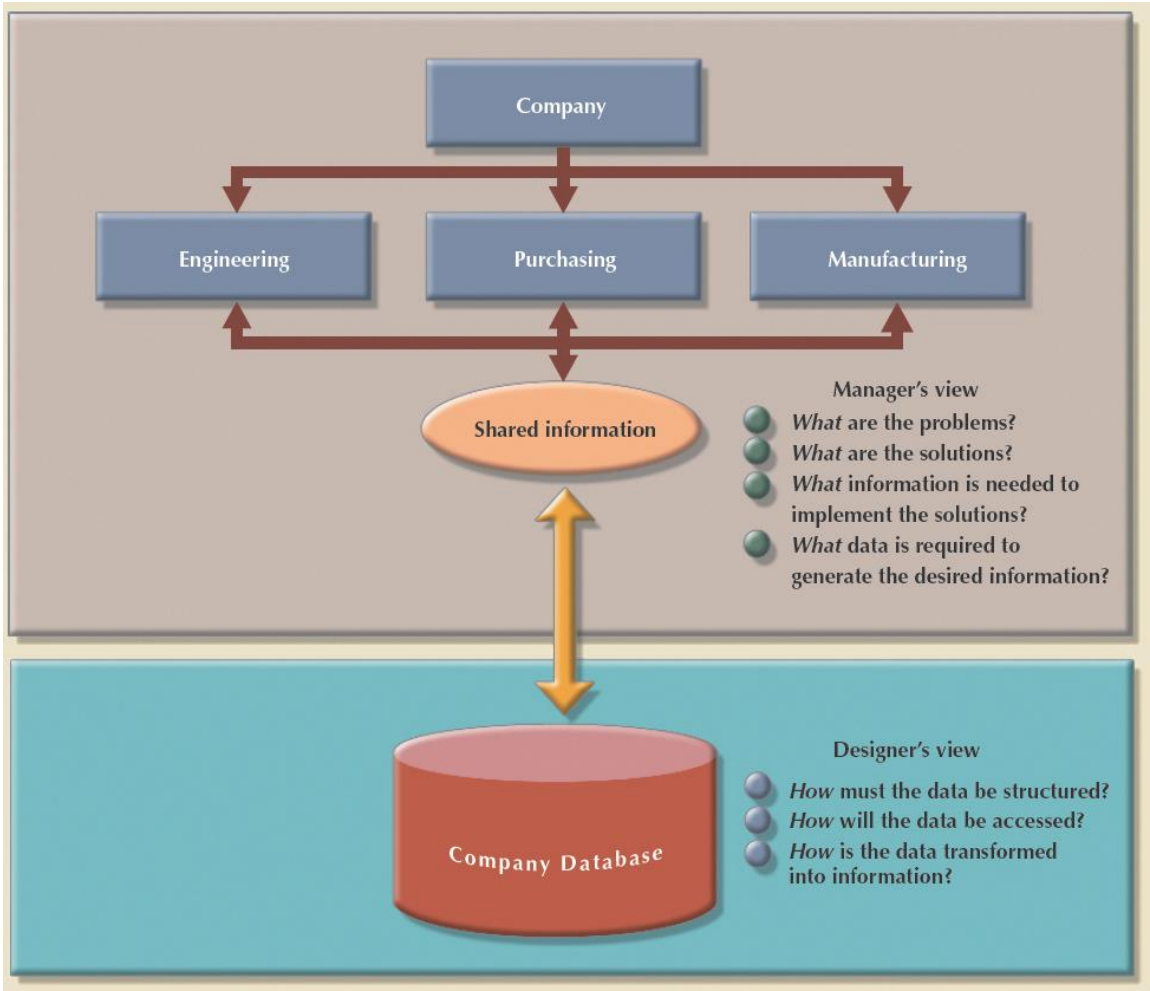
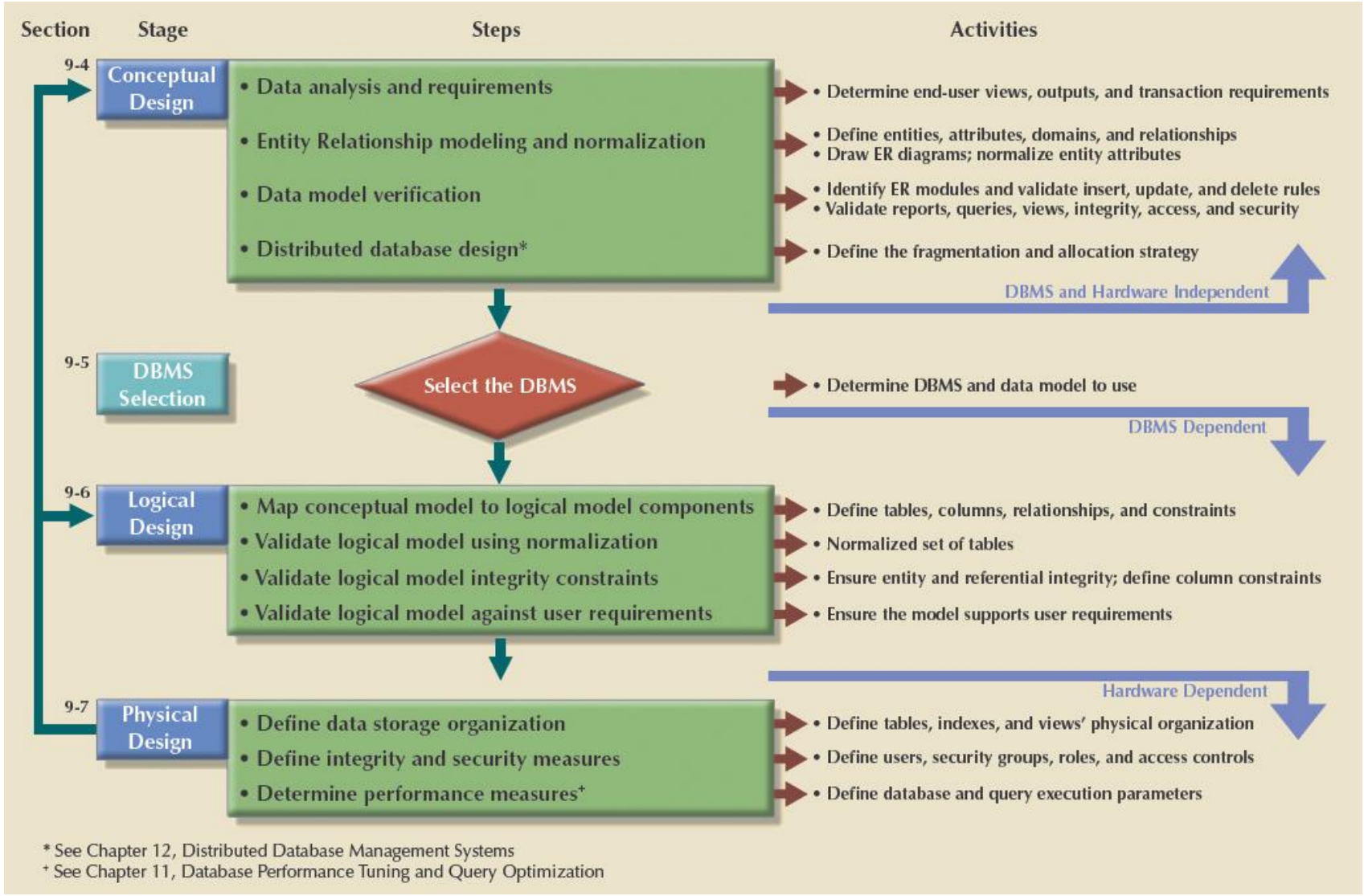


Figure 9.6 - Database Design Process



Implementation and Loading

- Install the DBMS
  - **Virtualization**: Creates logical representations of computing resources independent of underlying physical computing resources
- Create the databases
  - Requires the creation of special storage-related constructs to house the end-user tables
- Load or convert the data
  - Requires aggregating data from multiple sources

Testing and Evaluation

- Occurs in parallel with applications programming
- Physical security
- Password security
- Access rights
- Audit trails
- Data encryption
- Optimization

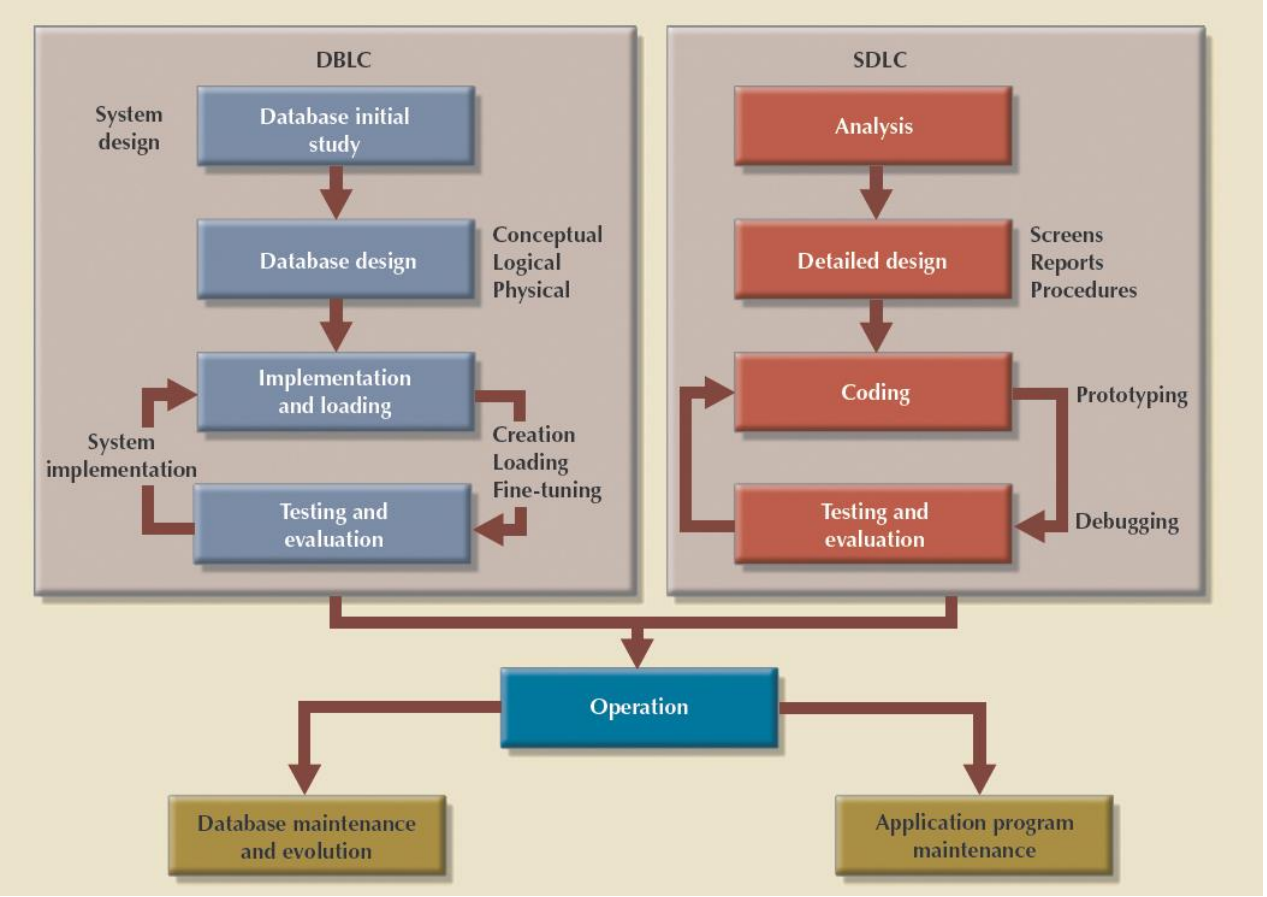
Levels of Database Backups

- **Full backup**/dump: All database objects are backed up in their entirety
- **Differential backup**: Only modified/updated objects since last full backup are backed up
- **Transaction log backup**: Only the transaction log operations that are not reflected in a previous backup are backed up
- Backups are provided with high security
- Recovery Point Objective (RPO) – How far back do we need to go?
- Recovery Time Objective (RTO) – How quickly do you need it?

Maintenance and Evolution

- 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

Figure 9.8 - Parallel Activities in the DBLC and the SDLC



Conceptual Design

- Designs a database independent of database software and physical details
- Conceptual data model - Describes main data entities, attributes, relationships, and constrains
- Designed as software and hardware independent
- **Minimum data rule**: All that is needed is there, and all that is there is needed

Table 9.2 - Conceptual Design Steps

STEP	ACTIVITY
1	Data analysis and requirements
2	Entity relationship modeling and normalization
3	Data model verification
4	Distributed database design

Data Analysis and Requirements

- Designers efforts are focused on
  - Information needs, users, sources and constitution
- Answers obtained from a variety of sources
  - Developing and gathering end-user data views
  - Directly observing current system: existing and desired output
  - Interfacing with the systems design group

Description of Operations

Provides precise, up-to-date, and reviewed description of activities defining an organization's operating environment

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.

Figure 9.10 – ER Modeling is an Iterative Process Based on Many Activities

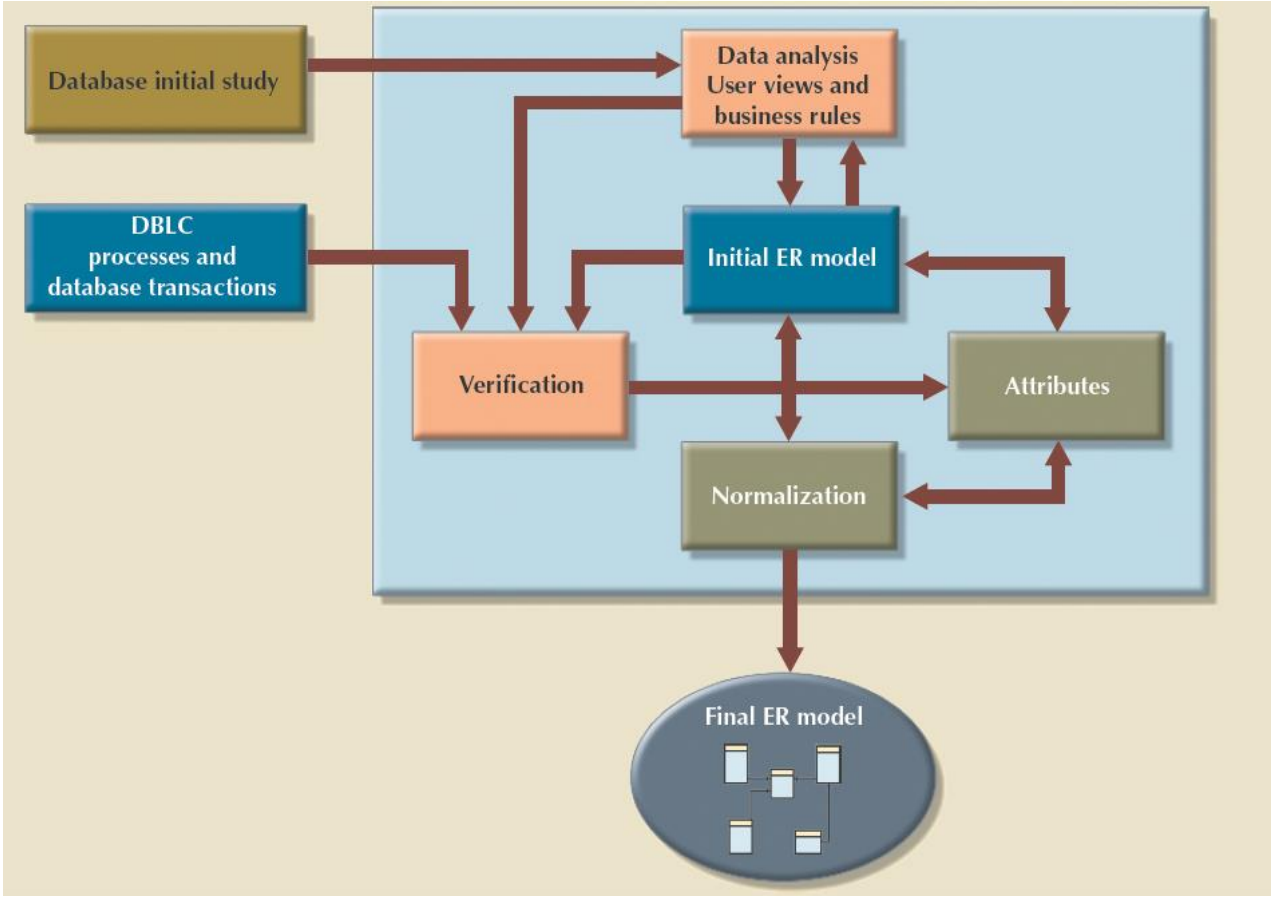
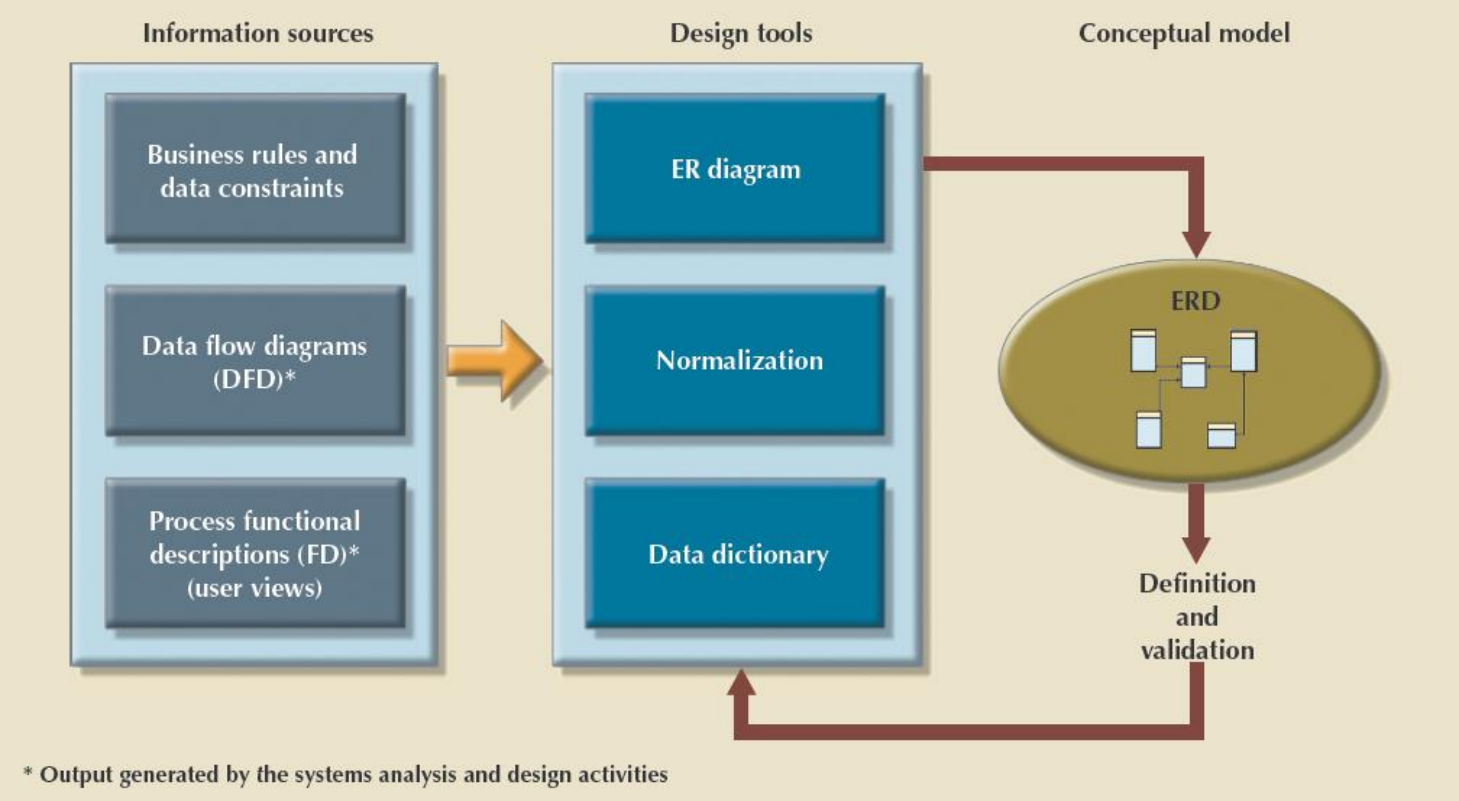


Figure 9.11 - Conceptual Design Tools and Information Sources



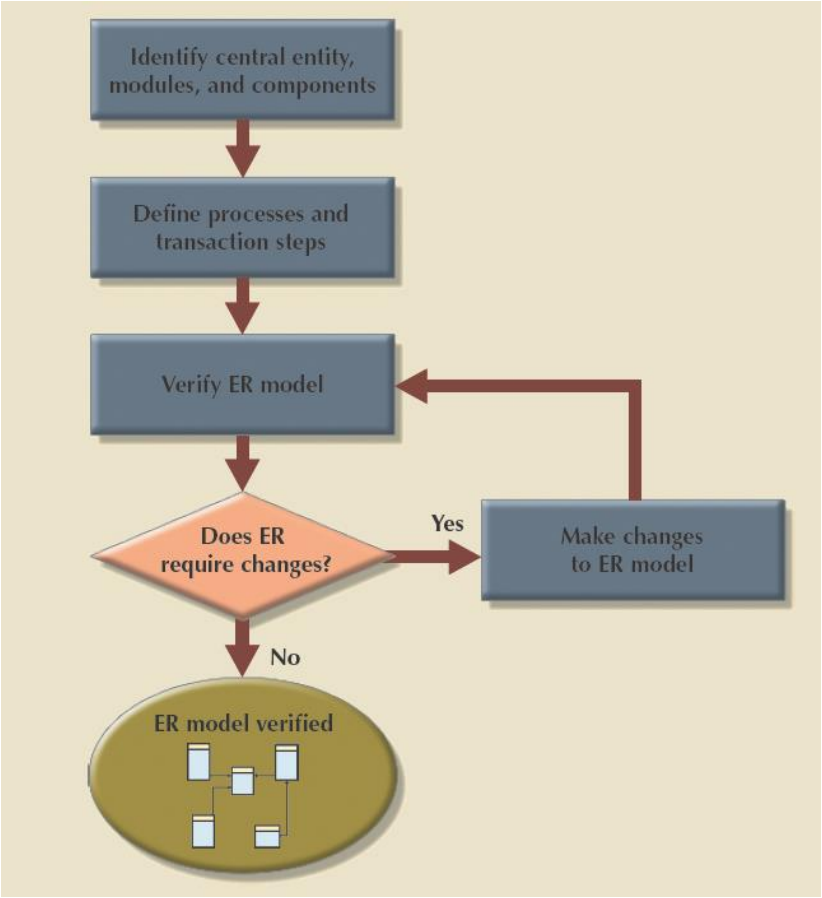
Data Model Verification

- Verified against proposed system processes
- **Module**: Information system component that handles specific business function
- Revision of original design
  - Careful reevaluation of entities
  - Detailed examination of attributes describing entities
- Resulting model verified against each of the module's processes

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.

Figure 9.12 - Iterative ER Model Verification Process



Cohesivity and Module Coupling

- **Cohesivity**: Strength of the relationships among the module's entities
  - Must display HIGH cohesivity – entities are strongly related, module complete and self-sufficient
- **Module coupling**: Extent to which modules are independent to one another
  - Low coupling decreases unnecessary intermodule dependencies

Distributed Database Design

- Portions of database may reside in different physical locations
- **Database fragment**: Subset of a database stored at a given location
- Ensures database integrity, security, and performance

DBMS Software Selection

- Cost
- DBMS features and tools
- Underlying model
- Portability
- DBMS hardware requirements
- Existing staff skillset
- Availability of resources
- Ease of analytics

Logical and Physical Design

- **Logical design**: Designs an enterprise-wide database that is based on a specific data model but independent of physical-level details
- Validates logical model:
  - Using normalization
  - Integrity constraints
  - Against user requirements
- **Physical design**: Process of data storage organization and data access characteristics of the database

Table 9.6 - Logical Design Steps

STEP	ACTIVITY
1	Map the conceptual model to logical model components.
2	Validate the logical model using normalization.
3	Validate the logical model integrity constraints.
4	Validate the logical model against user requirements.

Table 9.7 - Mapping the Conceptual Model to the Relational Model

STEP	ACTIVITY
1	Map strong entities
2	Map supertype/subtype relationships.
3	Map weak entities.
4	Map binary relationships.
5	Map higher-degree relationships.

Table 9.8 - Physical Design Steps

STEP	ACTIVITY
1	Define data storage organization.
2	Define integrity and security measures.
3	Determine performance measurements.

Clustered Tables and Database Role

Clustered Tables: Technique that stores related rows from two related tables in adjacent data blocks on disk

Database Role: Set of database privileges that could be assigned as a unit to a user or group

- SysAdmin Role
- Developer Role
- End-User role
- Auditor Role

Figure 9.14 - Top-down versus Bottom-up Design Sequencing

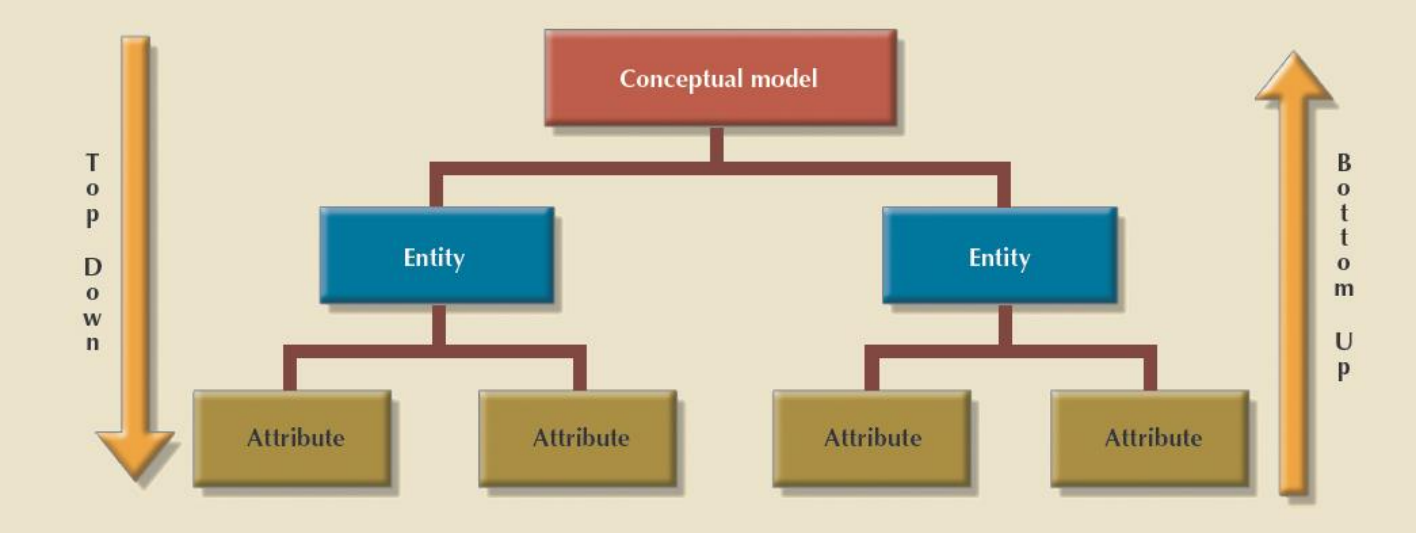


Figure 9.15 - Centralized Design

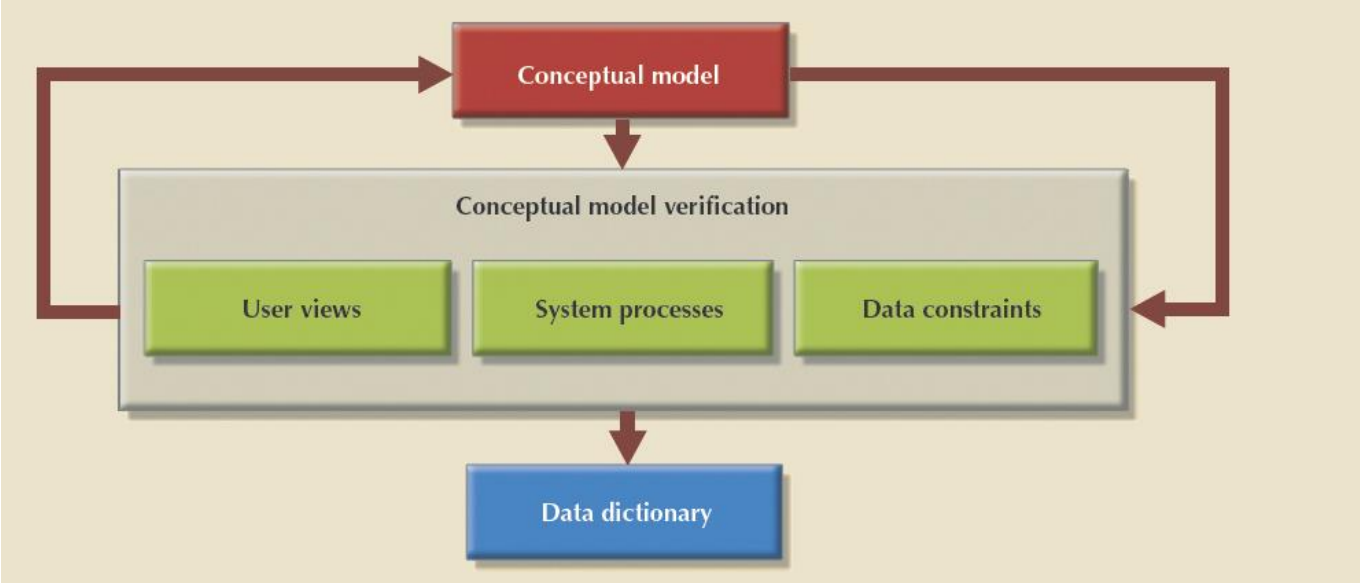


Figure 9.16 - Decentralized Design

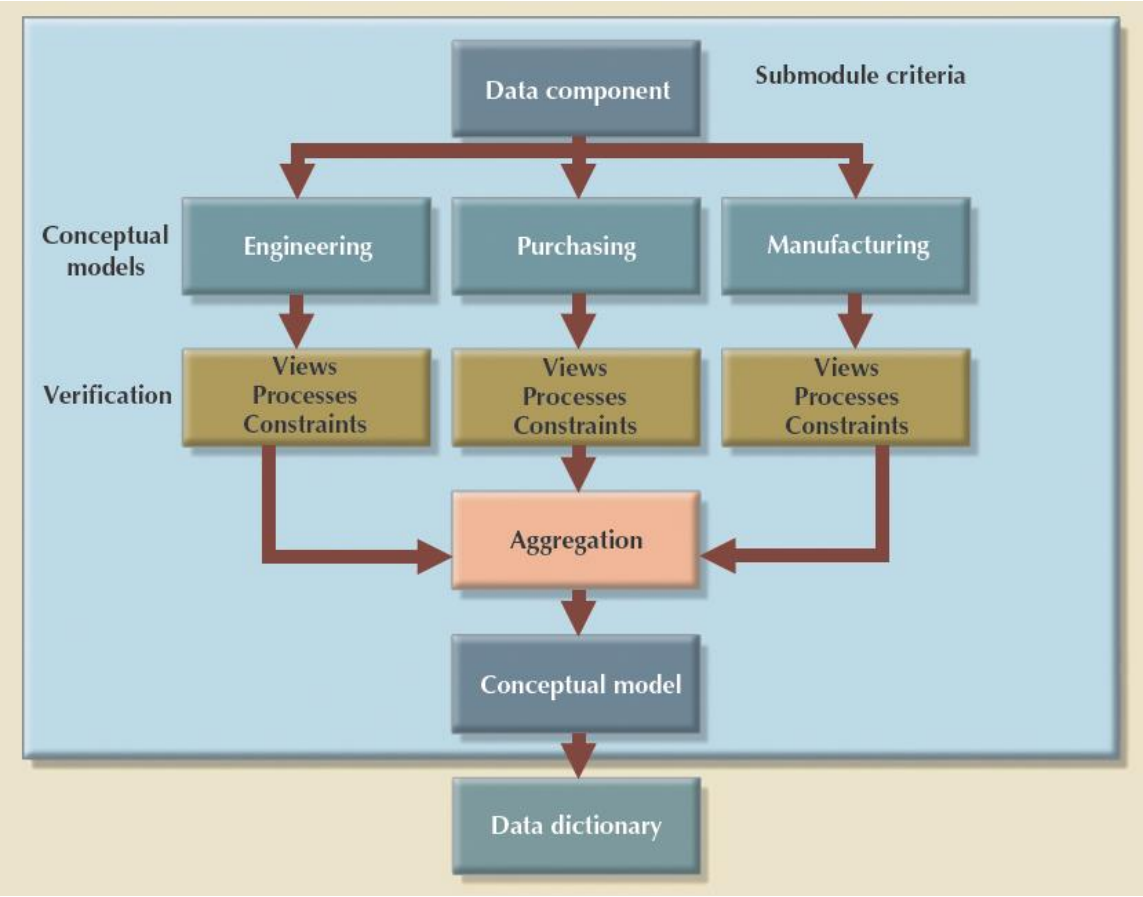


Figure 9.17 - Summary of Aggregation Problems