CSE 204 - INTRO TO DATABASE SYSTEMS DATABASE DEVELOPMENT

Joseph LEDET
Department of Computer Engineering
Akdeniz University
josephledet@akdeniz.edu.tr

OUTLINE

- Main components of an information system.
- Main stages of database system development lifecycle.
- Main phases of database design: conceptual, logical, and physical design.
- Benefits of CASE tools.
- How to evaluate and select a DBMS.
- Distinction between data administration and database administration.
- Purpose and tasks associated with data administration and database administration.

SOFTWARE DEPRESSION

- Last few decades have seen proliferation of software applications, many requiring constant maintenance involving:
  - correcting faults,
  - implementing new user requirements,
  - modifying software to run on new or upgraded platforms.
- Effort spent on maintenance began to absorb resources at an alarming rate.
- As a result, many major software projects were
  - late,
  - over budget,
  - unreliable,
  - difficult to maintain,
  - performed poorly.
- In late 1960s, led to ‘software crisis’, now refer to as the ‘software depression’.

SOFTWARE DEPRESSION

- Major reasons for failure of software projects includes:
  - lack of a complete requirements specification;
  - lack of appropriate development methodology;
  - poor decomposition of design into manageable components.
- Structured approach to development was proposed called Information Systems Lifecycle (ISLC).

INFORMATION SYSTEM

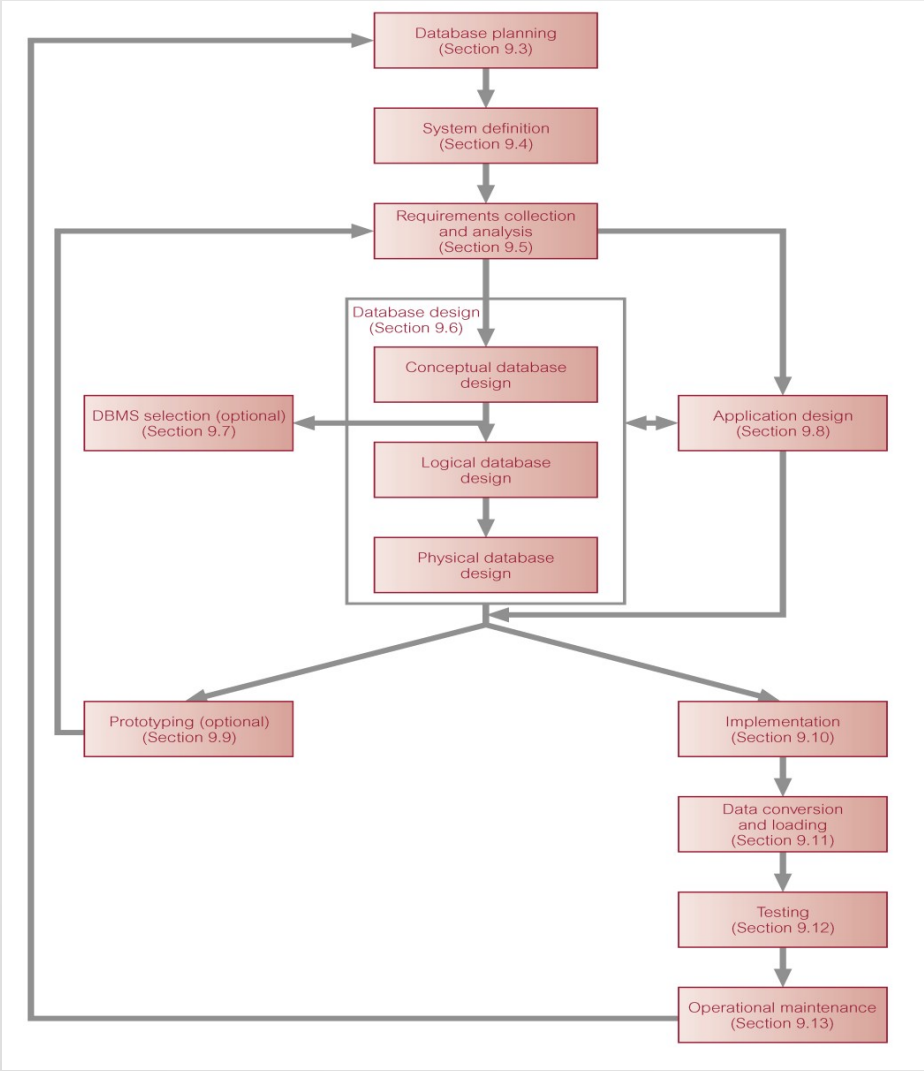
Resources that enable collection, management, control, and dissemination of information throughout an organization.

- Database is fundamental component of IS, and its development/usage should be viewed from perspective of the wider requirements of the organization.

DATABASE SYSTEM DEVELOPMENT LIFECYCLE

- Database planning
- System definition
- Requirements collection and analysis
- Database design
- DBMS selection (optional)
- Application design
- Prototyping (optional)
- Implementation
- Data conversion and loading
- Testing
- Operational maintenance

STAGES OF THE DATABASE SYSTEM DEVELOPMENT LIFECYCLE



DATABASE PLANNING

- Management activities that allow stages of database system development lifecycle to be realized as efficiently and effectively as possible.
- Must be integrated with overall IS strategy of the organization.

DATABASE PLANNING – MISSION STATEMENT

- *Mission statement* for the database project defines major aims of database application.
- Those driving database project normally define the mission statement.
- Mission statement helps clarify purpose of the database project and provides clearer path towards the efficient and effective creation of required database system.

DATABASE PLANNING – MISSION OBJECTIVES

- Once mission statement is defined, *mission objectives* are defined.
- Each objective should identify a particular task that the database must support.
- May be accompanied by some additional information that specifies the work to be done, the resources with which to do it, and the money to pay for it all.

DATABASE PLANNING

- Database planning should also include development of standards that govern:
  - how data will be collected,
  - how the format should be specified,
  - what necessary documentation will be needed,
  - how design and implementation should proceed.

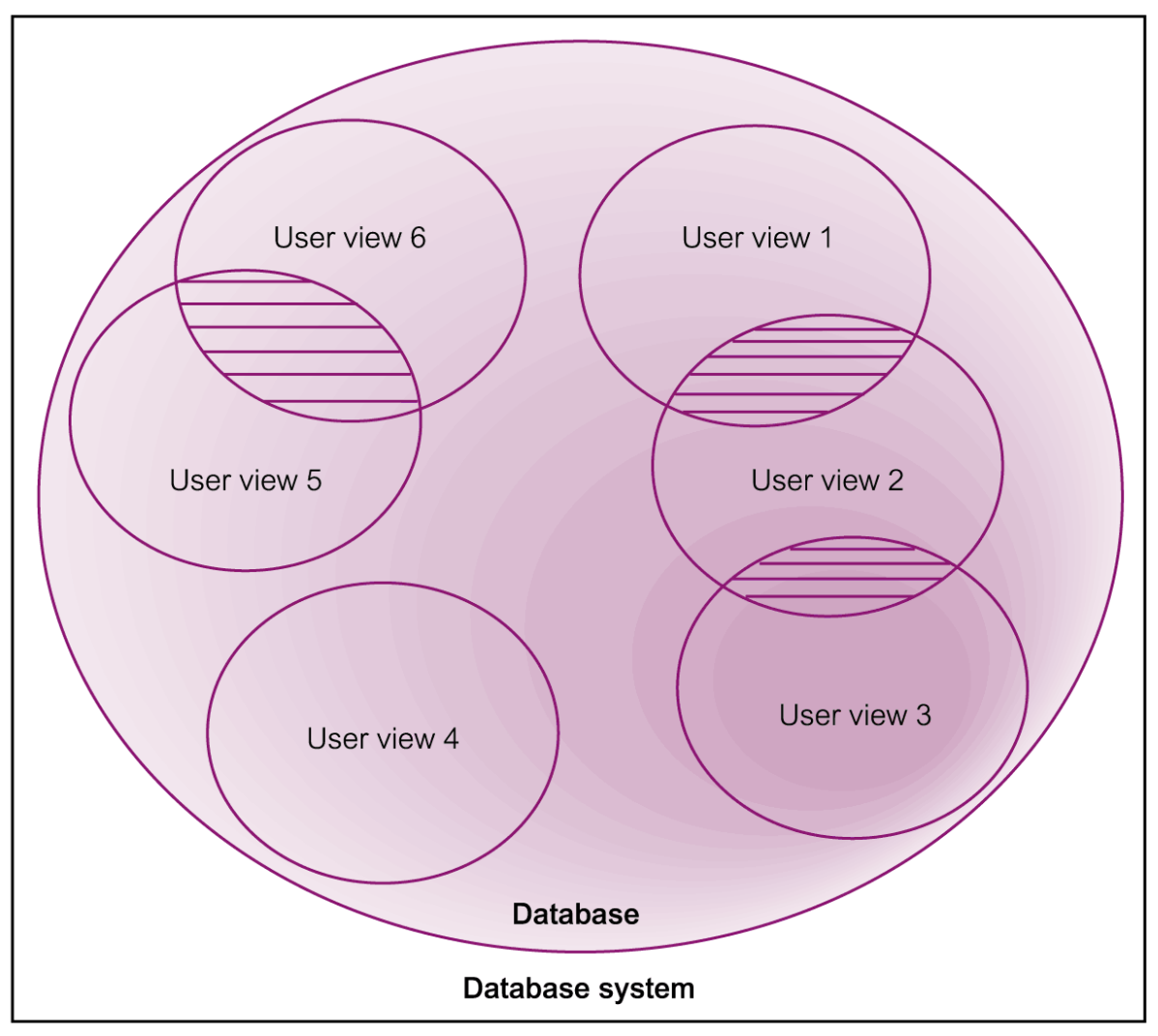
SYSTEM DEFINITION

- Describes scope and boundaries of database system and the major user views.
- User view defines what is required of a database system from perspective of:
  - a particular job role (such as Manager or Supervisor) or
  - enterprise application area (such as marketing, personnel, or stock control).

SYSTEM DEFINITION

- Database application may have one or more user views.
- Identifying user views helps ensure that no major users of the database are forgotten when developing requirements for new system.
- User views also help in development of complex database system allowing requirements to be broken down into manageable pieces.

REPRESENTATION OF A DATABASE SYSTEM WITH MULTIPLE USER VIEWS



REQUIREMENTS COLLECTION AND ANALYSIS

- Process of collecting and analyzing information about the part of organization to be supported by the database system, and using this information to identify users' requirements of new system.
- Information is gathered for each major user view including:
  - a description of data used or generated;
  - details of how data is to be used/generated;
  - any additional requirements for new database system.
- Information is analyzed to identify requirements to be included in new database system. Described in the requirements specification.

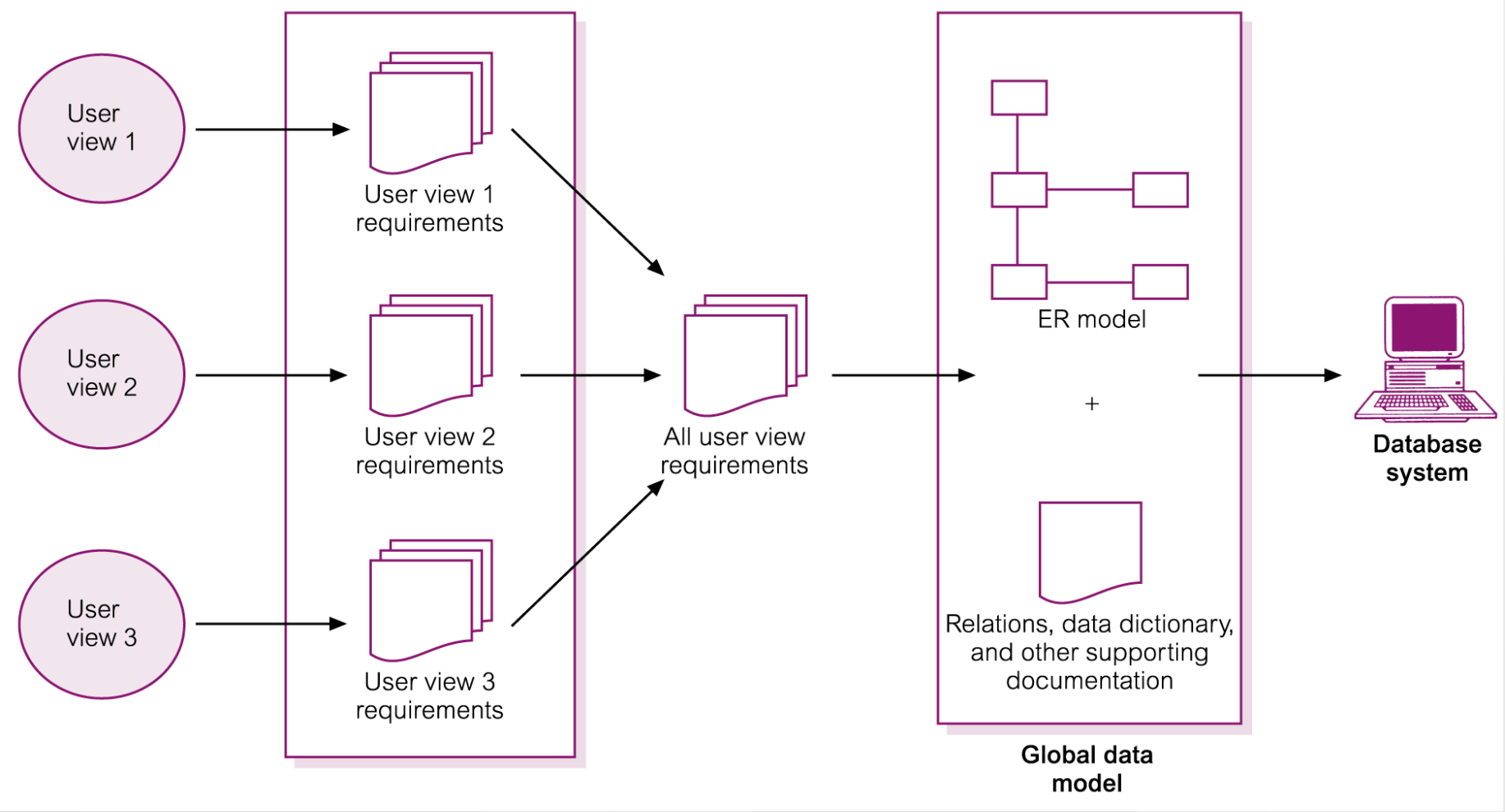
REQUIREMENTS COLLECTION AND ANALYSIS

- Another important activity is deciding how to manage the requirements for a database system with multiple user views.
- Three main approaches:
  - centralized approach;
  - view integration approach;
  - combination of both approaches.

REQUIREMENTS COLLECTION AND ANALYSIS

- Centralized approach
  - Requirements for each user view are merged into a single set of requirements.
  - A data model is created representing all user views during the database design stage.

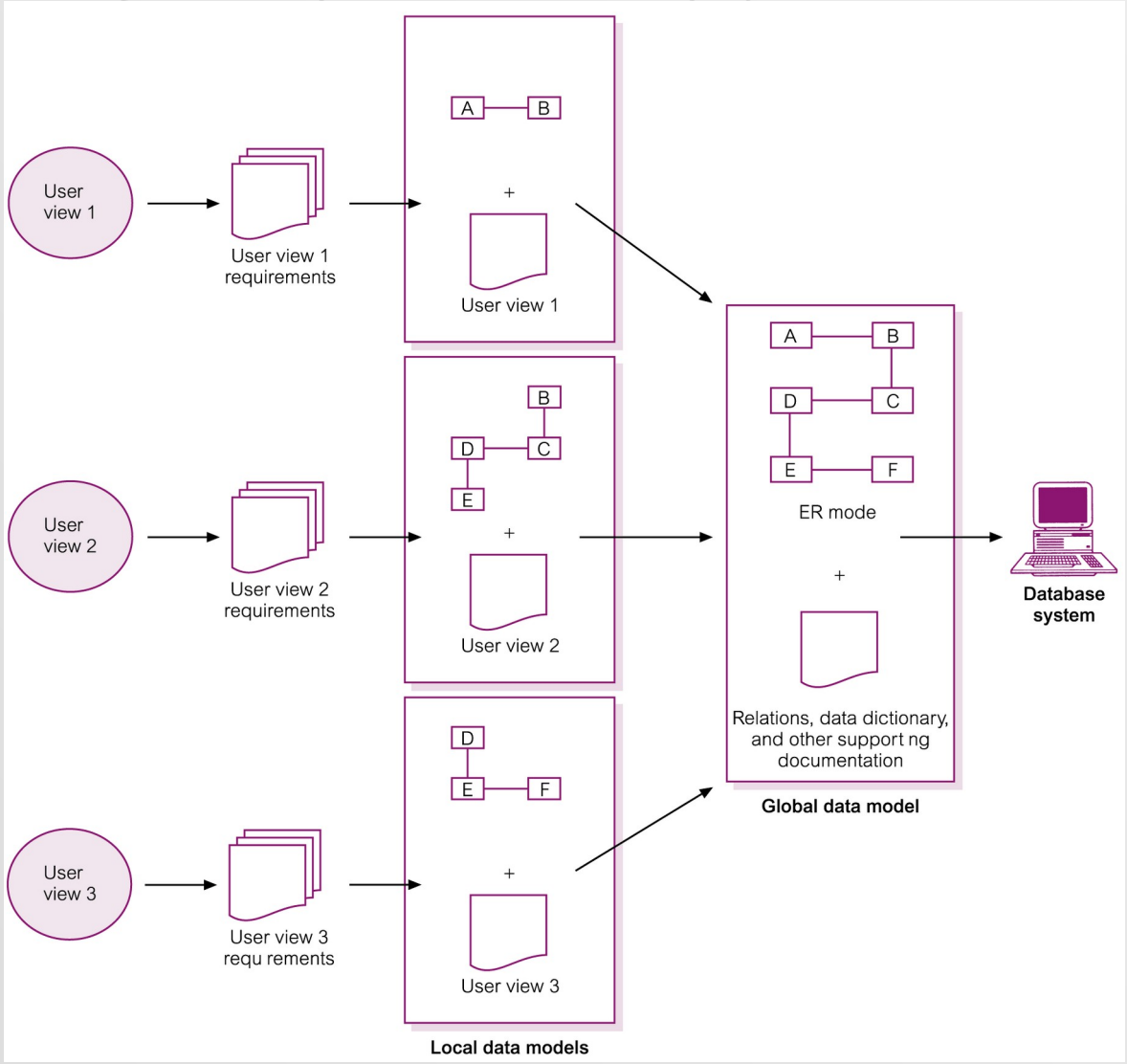
CENTRALIZED APPROACH TO MANAGING MULTIPLE USER VIEWS



REQUIREMENTS COLLECTION AND ANALYSIS

- View integration approach
  - Requirements for each user view remain as separate lists.
  - Data models representing each user view are created and then merged later during the database design stage.
- Data model representing single user view (or a subset of all user views) is called a local data model.
- Each model includes diagrams and documentation describing requirements for one or more but not all user views of database.
- Local data models are then merged at a later stage during database design to produce a global data model, which represents all user views for the database.

VIEW INTEGRATION APPROACH TO MANAGING MULTIPLE USER VIEWS



DATABASE DESIGN

- Process of creating a design for a database that will support the enterprise's mission statement and mission objectives for the required database system.
- Main approaches include:
  - Top-down
  - Bottom-up
  - Inside-out
  - Mixed

DATABASE DESIGN

- Main purposes of data modeling include:
  - to assist in understanding the meaning (semantics) of the data;
  - to facilitate communication about the information requirements.
- Building data model requires answering questions about entities, relationships, and attributes.
- A data model ensures we understand:
  - each user's perspective of the data;
  - nature of the data itself, independent of its physical representations;
  - use of data across user views.

CRITERIA TO PRODUCE AN OPTIMAL DATA MODEL

Structural validity	Consistency with the way the enterprise defines and organizes information.
Simplicity	Ease of understanding by IS professionals and non-technical users.
Expressibility	Ability to distinguish between different data, relationships between data, and constraints.
Nonredundancy	Exclusion of extraneous information; in particular, the representation of any one piece of information exactly once.
Shareability	Not specific to any particular application or technology and thereby usable by many.
Extensibility	Ability to evolve to support new requirements with minimal effect on existing users.
Integrity	Consistency with the way the enterprise uses and manages information.
Diagrammatic representation	Ability to represent a model using an easily understood diagrammatic notation.

DATABASE DESIGN

- Three phases of database design:
  - Conceptual database design
  - Logical database design
  - Physical database design.

CONCEPTUAL DATABASE DESIGN

- Process of constructing a model of the data used in an enterprise, independent of all physical considerations.
- Data model is built using the information in users' requirements specification.
- Conceptual data model is source of information for logical design phase.

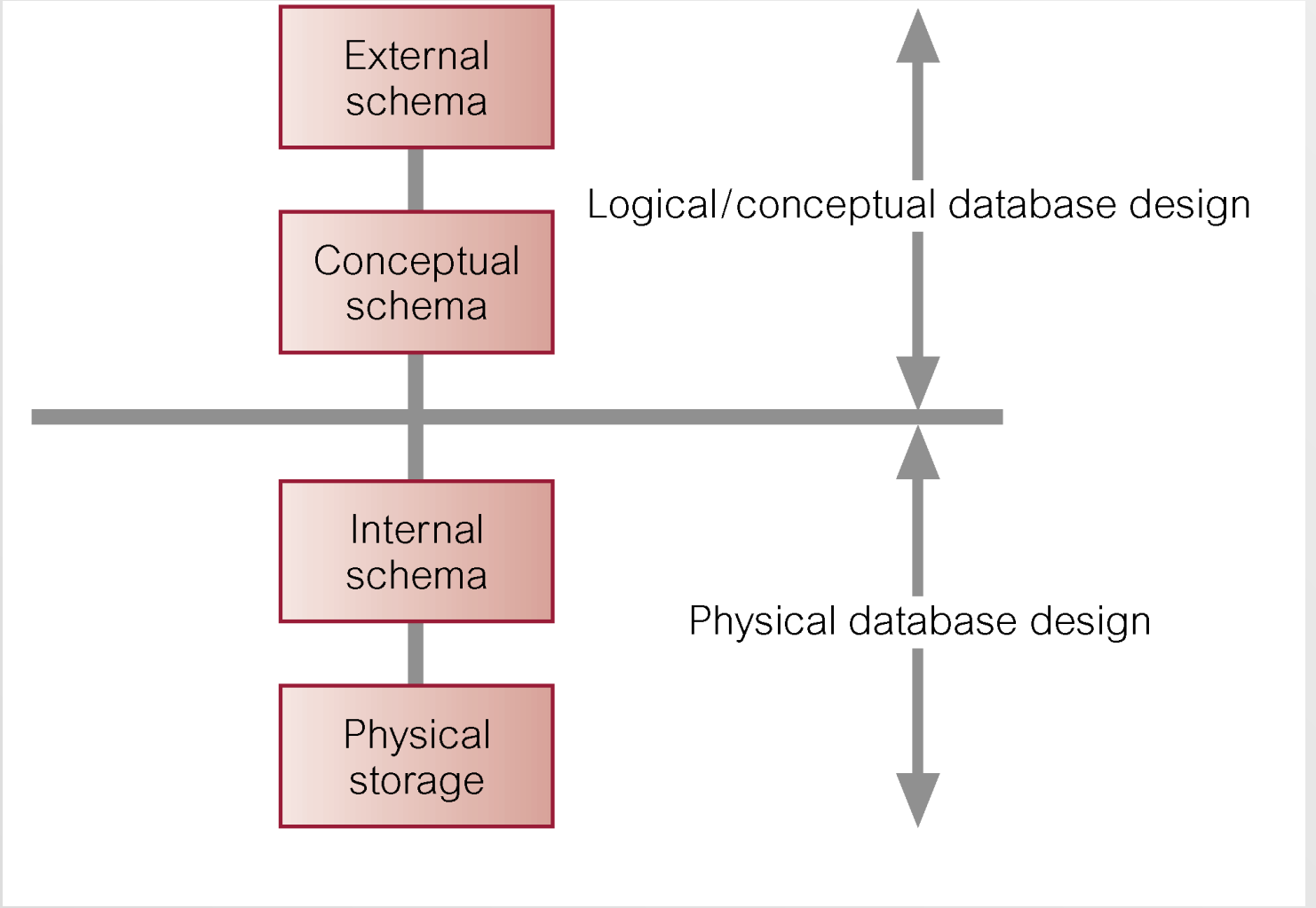
LOGICAL DATABASE DESIGN

- Process of constructing a model of the data used in an enterprise based on a specific data model (e.g. relational), but independent of a particular DBMS and other physical considerations.
- Conceptual data model is refined and mapped on to a logical data model.

PHYSICAL DATABASE DESIGN

- Process of producing a description of the database implementation on secondary storage.
- Describes base relations, file organizations, and indexes used to achieve efficient access to data. Also describes any associated integrity constraints and security measures.
- Tailored to a specific DBMS system.

THREE-LEVEL ANSI-SPARC ARCHITECTURE AND PHASES OF DATABASE DESIGN



DBMS SELECTION

- Selection of an appropriate DBMS to support the database system.
- Undertaken at any time prior to logical design provided sufficient information is available regarding system requirements.
- Main steps to selecting a DBMS:
  - define Terms of Reference of study;
  - shortlist two or three products;
  - evaluate products;
  - recommend selection and produce report.

DBMS EVALUATION FEATURES

DATA DEFINITION	PHYSICAL DEFINITION
Primary key enforcement	File structures available
Foreign key specification	File structure maintenance
Data types available	Ease of reorganization
Data type extensibility	Indexing
Domain specification	Variable length fields/records
Ease of restructuring	Data compression
Integrity controls	Encryption routines
View mechanism	Memory requirements
Data dictionary	Storage requirements
Data independence	
Underlying data model	
Schema evolution	
ACCESSIBILITY	**TRANSACTION HANDLING**
Query language: SQL2/SQL:2011/ODMG compliant	Backup and recovery routines Checkpointing facility
Interfacing to 3GLs	Logging facility
Multi-user	Granularity of concurrency
Security	Deadlock resolution strategy
• Access controls	Advanced transaction models
• Authorization mechanism	Parallel query processing

DBMS EVALUATION FEATURES

UTILITIES	DEVELOPMENT
Performance measuring	4GL/5GL tools
Tuning	CASE tools
Load/unload facilities	Windows capabilities
User usage monitoring	Stored procedures, triggers, and rules
Database administration support	Web development tools
OTHER FEATURES	
Upgradability	Interoperability with other DBMSs and other systems
Vendor stability	Web integration
User base	Replication utilities
Training and user support	Distributed capabilities
Documentation	Portability
Operating system required	Hardware required
Cost	Network support
Online help	Object-oriented capabilities
Standards used	Architecture (2- or 3-tier client/server)
Version management	Performance
Extensibile query optimization	Transaction throughput
Scalability	Maximum number of concurrent users
Support for reporting and analytical tools	XML and Web services support

EXAMPLE - EVALUATION OF DBMS PRODUCT

DBMS: Sample product
Vendor: Sample vendor

Physical Definition Group

Features	Comments	Rating	Weighting	Score
File structures available	Choice of 4	8	0.15	1.2
File structure maintenance	NOT self-regulating	6	0.2	1.2
Ease of reorganization		4	0.25	1.0
Indexing		6	0.15	0.9
Variable length fields/records		6	0.15	0.9
Data compression	Specify with file structure	7	0.05	0.35
Encryption routines	Choice of 2	4	0.05	0.2
Memory requirements		0	0.00	0
Storage requirements		0	0.00	0
Totals		41	1.0	**5.75**
Physical definition group		5.75	0.25	**1.44**

APPLICATION DESIGN

- Design of user interface and application programs that use and process the database.
- Database design and application design are parallel activities.
- Includes two important activities:
  - transaction design;
  - user interface design.

APPLICATION DESIGN - TRANSACTIONS

- An action, or series of actions, carried out by a single user or application program, which accesses or changes content of the database.
- Should define and document the high-level characteristics of the transactions required.

APPLICATION DESIGN - TRANSACTIONS

- Important characteristics of transactions:
  - data to be used by the transaction;
  - functional characteristics of the transaction;
  - output of the transaction;
  - importance to the users;
  - expected rate of usage.
- Three main types of transactions: retrieval, update, and mixed.

PROTOTYPING

- Building working model of a database system.
- Purpose
  - to identify features of a system that work well, or are inadequate;
  - to suggest improvements or even new features;
  - to clarify the users' requirements;
  - to evaluate feasibility of a particular system design.

IMPLEMENTATION

- Physical realization of the database and application designs.
  - Use DDL to create database schemas and empty database files.
  - Use DDL to create any specified user views.
  - Use 3GL or 4GL to create the application programs. This will include the database transactions implemented using the DML, possibly embedded in a host programming language.

DATA CONVERSION AND LOADING

- Transferring any existing data into new database and converting any existing applications to run on new database.
- Only required when new database system is replacing an old system.
  - DBMS normally has utility that loads existing files into new database.
- May be possible to convert and use application programs from old system for use by new system.

TESTING

- Process of running the database system with intent of finding errors.
- Use carefully planned test strategies and realistic data.
- Testing cannot show absence of faults; it can show only that software faults are present.
- Demonstrates that database and application programs appear to be working according to requirements.

TESTING

- Should also test usability of system.
- Evaluation conducted against a usability specification.
- Examples of criteria include:
  - Learnability;
  - Performance;
  - Robustness;
  - Recoverability;
  - Adaptability.

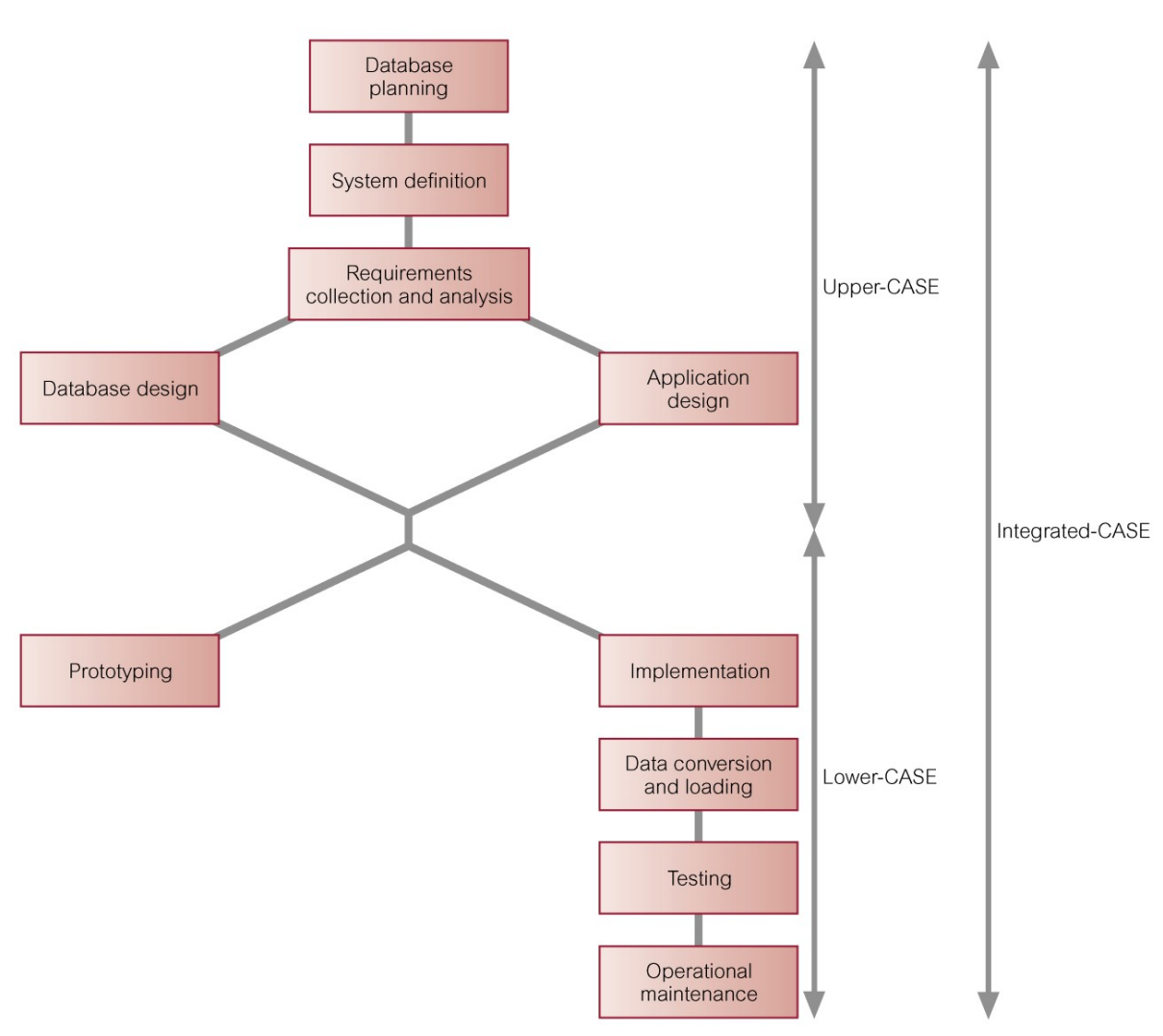
OPERATIONAL MAINTENANCE

- Process of monitoring and maintaining database system following installation.
- Monitoring performance of system.
- if performance falls, may require tuning or reorganization of the database.
- Maintaining and upgrading database application (when required).
- Incorporating new requirements into database application.

CASE TOOLS

- Support provided by CASE tools include:
  - data dictionary to store information about database system's data;
  - design tools to support data analysis;
  - tools to permit development of corporate data model, and conceptual and logical data models;
  - tools to enable prototyping of applications.
- Provide following benefits:
  - Standards;
  - Integration;
  - Support for standard methods;
  - Consistency;
  - Automation .

CASE TOOLS AND DATABASE SYSTEM DEVELOPMENT LIFECYCLE



DATA ADMINISTRATION AND DATABASE ADMINISTRATION

- The Data Administrator (DA) and Database Administrator (DBA) are responsible for managing and controlling the corporate data and corporate database, respectively.
- DA is more concerned with early stages of database system development lifecycle and DBA is more concerned with later stages.

DATA ADMINISTRATION

- Management of data resource including:
  - database planning,
  - development and maintenance of standards, policies and procedures, and conceptual and logical database design.

DATABASE ADMINISTRATION

- Management of physical realization of a database system including:
  - physical database design and implementation,
  - setting security and integrity controls,
  - monitoring system performance, and reorganizing the database.