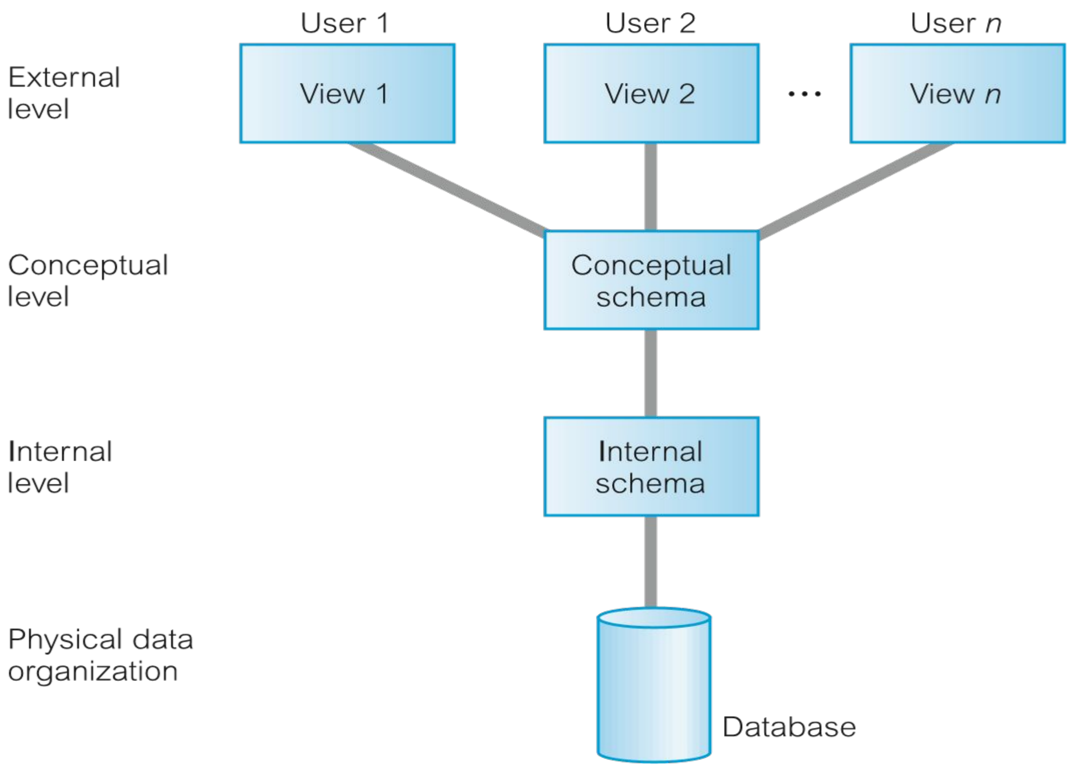
Database Planning, Design, and Administration

Naranbat Bayanmunkh

National University of Mongolia

The Three-Level ANSI-SPARC Architecture

External level	**The users' view of the database. This level describes that part of the database that is relevant to each user.**
Conceptual level	**The community view of the database. This level describes what data is stored in the database and the relationships among the data.**
Internal level	**The physical representation of the database on the computer. This level describes how the data is stored in the database.**
Physical level	**that may be managed by the operating system under the direction of the DBMS.**



Schemas, Mappings, and Instances

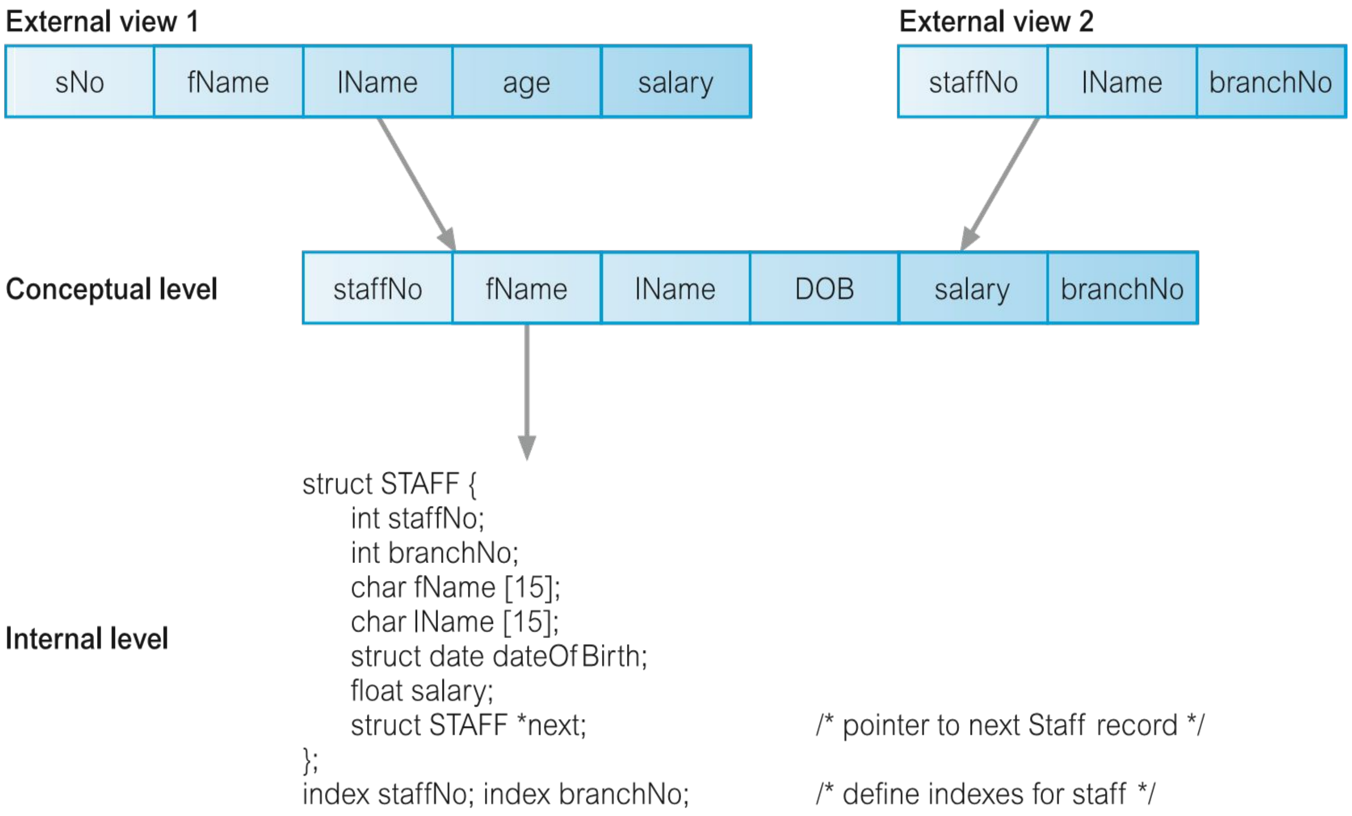
Database schema.

- The overall description of the database is called the database schema.

There are three different types of schema

- **Conceptual schema**, which describes all the entities, attributes, and relationships together with integrity constraints.
- **Internal schema**, which is a complete description of the internal model, containing the definitions of stored records, the methods of representation, the data fields, and the indexes and storage structures used.
- There is only one conceptual schema and one internal schema per database.

Differences between the three levels.



External view 1

sNo	fName	lName	age	salary

External view 2

staffNo	lName	branchNo

Conceptual level

staffNo	fName	lName	DOB	salary	branchNo

Internal level

```
struct STAFF {
      int staffNo;
      int branchNo;
      char fName [15];
      char lName [15];
      struct date dateOfBirth;
      float salary;
      struct STAFF *next;             /* pointer to next Staff record */
};
index staffNo; index branchNo;        /* define indexes for staff */
```

Major components of DBMS

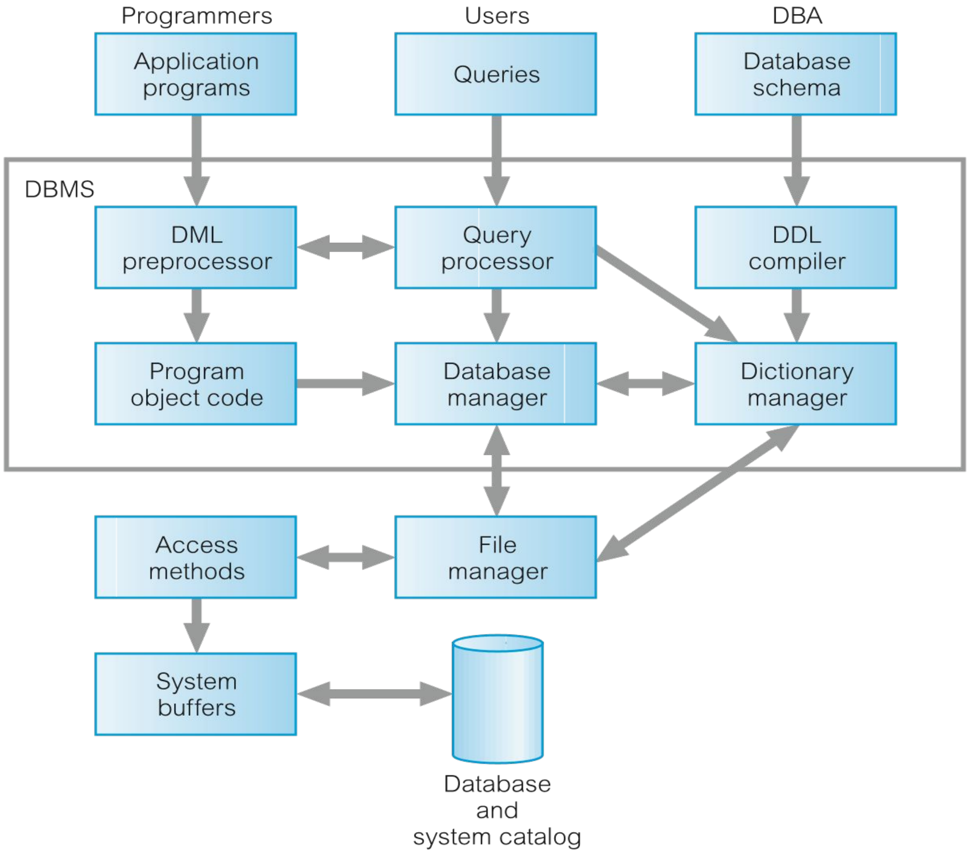


Figure 2.8
Major components of a DBMS.

Components of a Database manager

Figure 2.9
Components of a database manager.

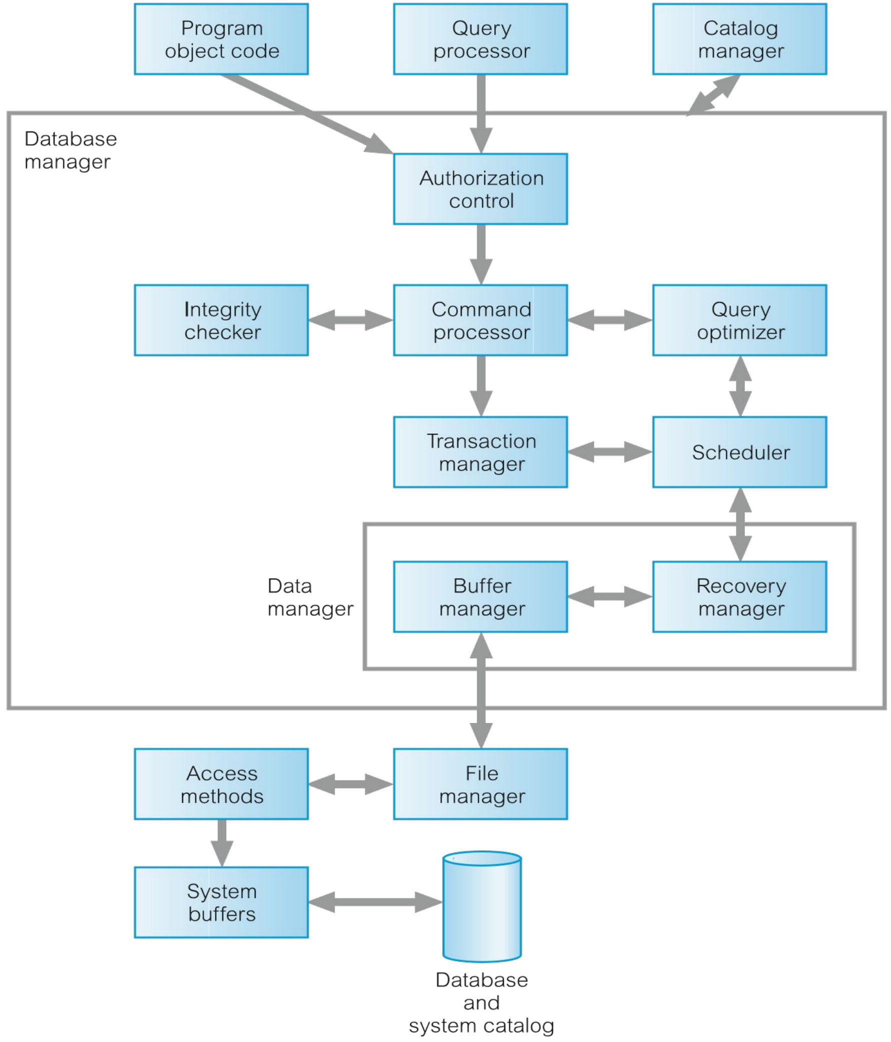


Figure 2.8 shows the following components:

Components of DBMS

Query processor

This is a major DBMS component that transforms queries into a series of low-level instructions directed to the database manager.

Database manager (DM)

The DM interfaces with user-submitted application programs and queries. The DM accepts queries and examines the external and conceptual schemas to determine what conceptual records are required to satisfy the request. The DM then places a call to the file manager to perform the request.

File manager

The file manager manipulates the underlying storage files and manages the allocation of storage space on disk. It establishes and maintains the list of structures

Components of a DBMS and indexes defined in the internal schema.

If hashed files are used it calls on the hashing functions to generate record addresses. However, the file manager does not directly manage the physical input and output of data. Rather it passes the requests on to the appropriate access methods, which either read data from or write data into the system buffer (or cache).

Components of DBMS

DML preprocessor

This module converts DML statements embedded in an application program into standard function calls in the host language. The DML preprocessor must interact with the query processor to generate the appropriate code.

DDL compiler

The DDL compiler converts DDL statements into a set of tables containing metadata. These tables are then stored in the system catalog while control information is stored in data file headers.

Catalog manager

The catalog manager manages access to and maintains the system catalog. The system catalog is accessed by most DBMS components. The major software components for the database manager are as follows:

Authorization control

This module checks that the user has the necessary authorization to carry out the required operation.

Components of DBMS

Command processor

Once the system has checked that the user has authority to carry out the operation, control is passed to the command processor.

Integrity checker For an operation that changes the database, the integrity checker checks that the requested operation satisfies all necessary integrity constraints (such as key constraints).

Query optimizer

This module determines an optimal strategy for the query execution. We discuss query optimization in Chapter 21.

Transaction manager

This module performs the required processing of operations it receives from transactions.

Scheduler

This module is responsible for ensuring that concurrent operations on the database proceed without conflicting with one another. It controls the relative order in which transaction operations are executed.

Components of DBMS

Recovery manager

This module ensures that the database remains in a consistent state in the presence of failures. It is responsible for transaction commit and abort.

Buffer manager

This module is responsible for the transfer of data between main memory and secondary storage, such as disk and tape. The recovery manager and the buffer manager are sometimes referred to collectively as the data manager. The buffer manager is sometimes known as the cache manager.

Information system

Information system	**The resources that enable the collection, management, control, and dissemination of information throughout an organization.**

A computer-based **information system** includes a

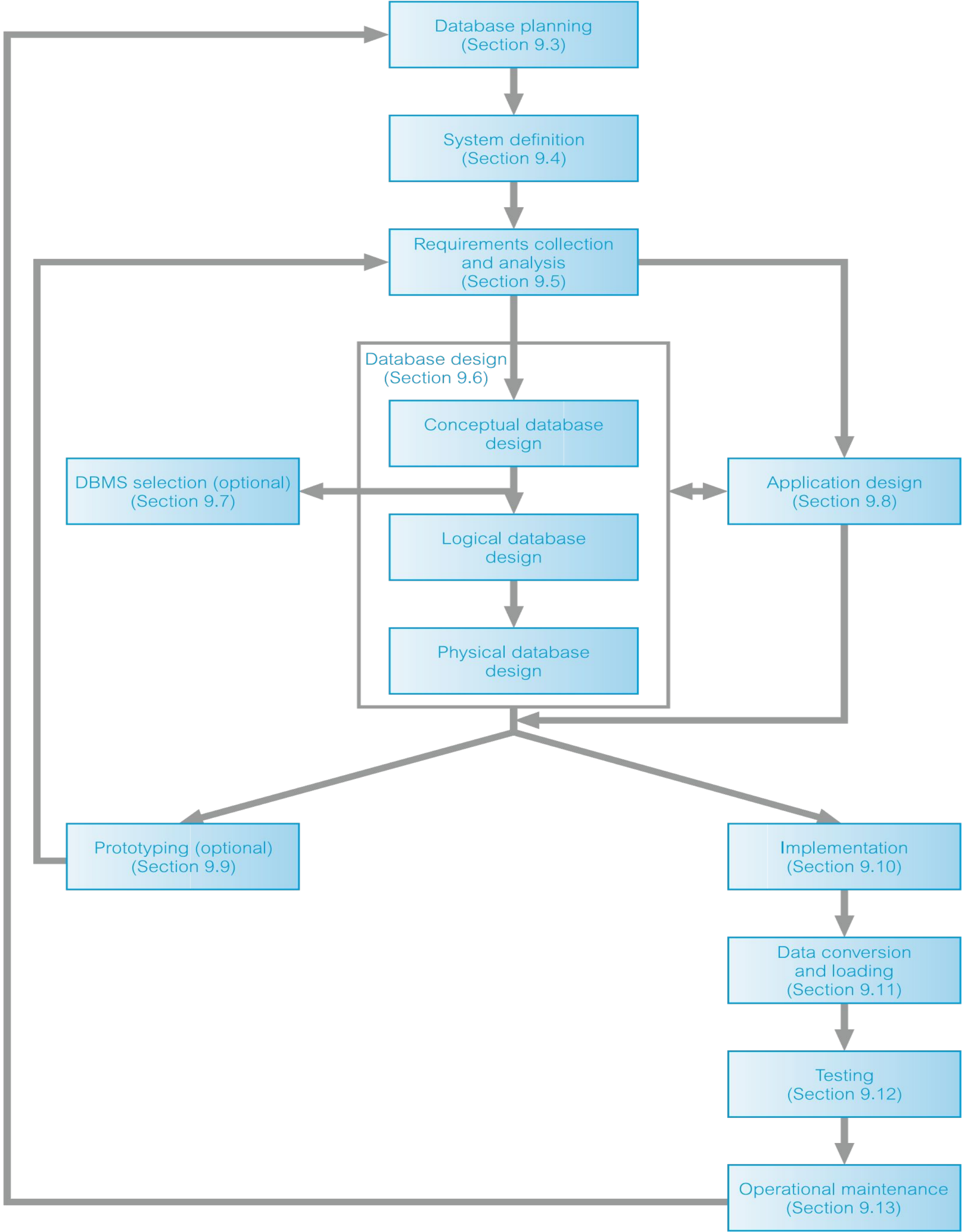
- database,
- database software,
- application software,
- computer hardware, and
- personnel using and developing the system.

The **database is a fundamental component** of an information system, and its development and usage should be viewed from the perspective of the wider requirements of the organization.

Therefore, the lifecycle of an organization's information system is inherently linked to the lifecycle of the database system that supports it.

Typically, the stages in the lifecycle of an **information system** include: planning, requirements collection and analysis, design, prototyping, implementation, testing, conversion, and operational maintenance.

DB Life cycle



Database life cycle

Stage	Main activities
Database planning	Planning how the stages of the lifecycle can be realized most efficiently and effectively.
System definition	Specifying the scope and boundaries of the database system, including the major user views, its users, and application areas.
Requirements collection and analysis Database design	Collection and analysis of the requirements for the new database system.
	Conceptual, logical, and physical design of the database.
DBMS selection (optional)	Selecting a suitable DBMS for the database system.
Application design	Designing the user interface and the application programs that use and process the database.
Prototyping (optional)	Building a working model of the database system, which allows the designers or users to visualize and evaluate how the final system will look and function.
Implementation	Creating the physical database definitions and the application programs.
Data conversion and loading	Loading data from the old system to the new system and, where possible, converting any existing applications to run on the new database.
Testing	Database system is tested for errors and validated against the requirements specified by the users.
Operational maintenance	Database system is fully implemented. The system is continuously monitored and maintained. When necessary, new requirements are incorporated into the database system through the preceding stages of the lifecycle.

User View

User view	**Defines what is required of a database system from the perspective of a particular job role (such as Manager or Supervisor) or enterprise application area (such as marketing, personnel, or stock control).**

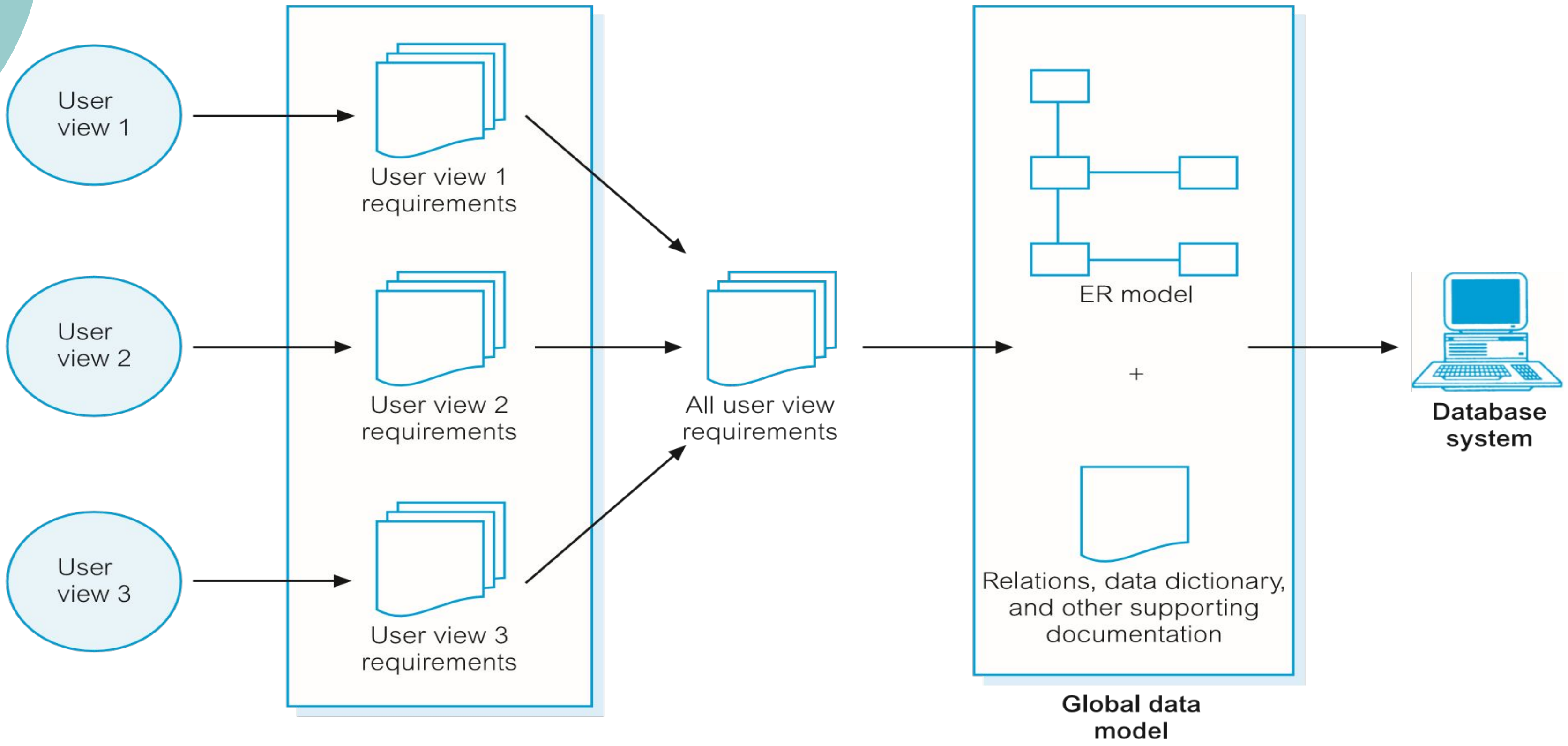


Figure 9.3 The centralized approach to managing multiple user views 1 to 3

Database life cycle

System definition involves identifying the scope and boundaries of the database system and user views. A user view defines what is required of a database system from the perspective of a particular job role (such as Manager or Supervisor) or enterprise application (such as marketing, personnel, or stock control).

Requirements collection and analysis is the process of collecting and analyzing information about the part of the organization that is to be supported by the database system, and using this information to identify the requirements for the new system

Database design is the process of creating a design that will support the enterprise's mission statement and mission objectives for the required database system. There are three phases of database design, namely conceptual, logical, and physical database design.

Conceptual database design is the process of constructing a model of the data used in an enterprise, independent of all physical considerations.

Logical database design is the process of constructing a model of the data used in an enterprise based on a specific data model, but independent of a particular DBMS and other physical considerations.

Database life cycle

Physical database design is the process of producing a description of the implementation of the database on secondary storage; it describes the base relations, file organizations, and indexes used to achieve efficient access to the data, and any associated integrity constraints and security measures. n DBMS selection involves selecting a suitable DBMS for the database system.

Application design involves user interface design and transaction design, which describes the application programs that use and process the database. A database transaction is an action, or series of actions, carried out by a single user or application program, which accesses or changes the content of the database.

Prototyping involves building a working model of the database system, which allows the designers or users to visualize and evaluate the system.

Implementation is the physical realization of the database and application designs.

Data conversion and loading involves transferring any existing data into the new database and converting any existing applications to run on the new database.

Database life cycle

Testing is the process of running the database system with the intent of finding errors.

Operational maintenance is the process of monitoring and maintaining the system following installation.

Data administration is the management of the data resource, including database planning, development and maintenance of standards, policies and procedures, and conceptual and logical database design.

Database administration is the management of the physical realization of a database system, including physical database design and implementation, setting security and integrity controls, monitoring system performance, and reorganizing the database as necessary.

Overview of the Database Design Methodology

In this section, we present an overview of the database design methodology. The steps in the methodology are as follows.

Conceptual database design

Build conceptual data model

- Identify entity types
- Identify relationship types
- Identify and associate attributes with entity or relationship types
- Determine attribute domains
- Determine candidate, primary, and alternate key attributes
- Consider use of enhanced modeling concepts (optional step)
- Check model for redundancy
- Validate conceptual model against user transactions
- Review conceptual data model with user

Overview of the Database Design Methodology

Logical database design for the relational model

Build and validate logical data model

- Derive relations for logical data model
- Validate relations using normalization
- Validate relations against user transactions
- Check integrity constraints
- Review logical data model with user
- Merge logical data models into global model (optional step)
- Check for future growth

Physical database design for relational databases

Translate logical data model for target DBMS

- Design base relations
- Design representation of derived data
- Design general constraints

Overview of the Database Design Methodology

Design file organizations and indexes

- Analyze transactions
- Choose file organizations
- Choose indexes
- Estimate disk space requirements

Design user views

Design security mechanisms

Consider the introduction of controlled redundancy

Monitor and tune the operational system