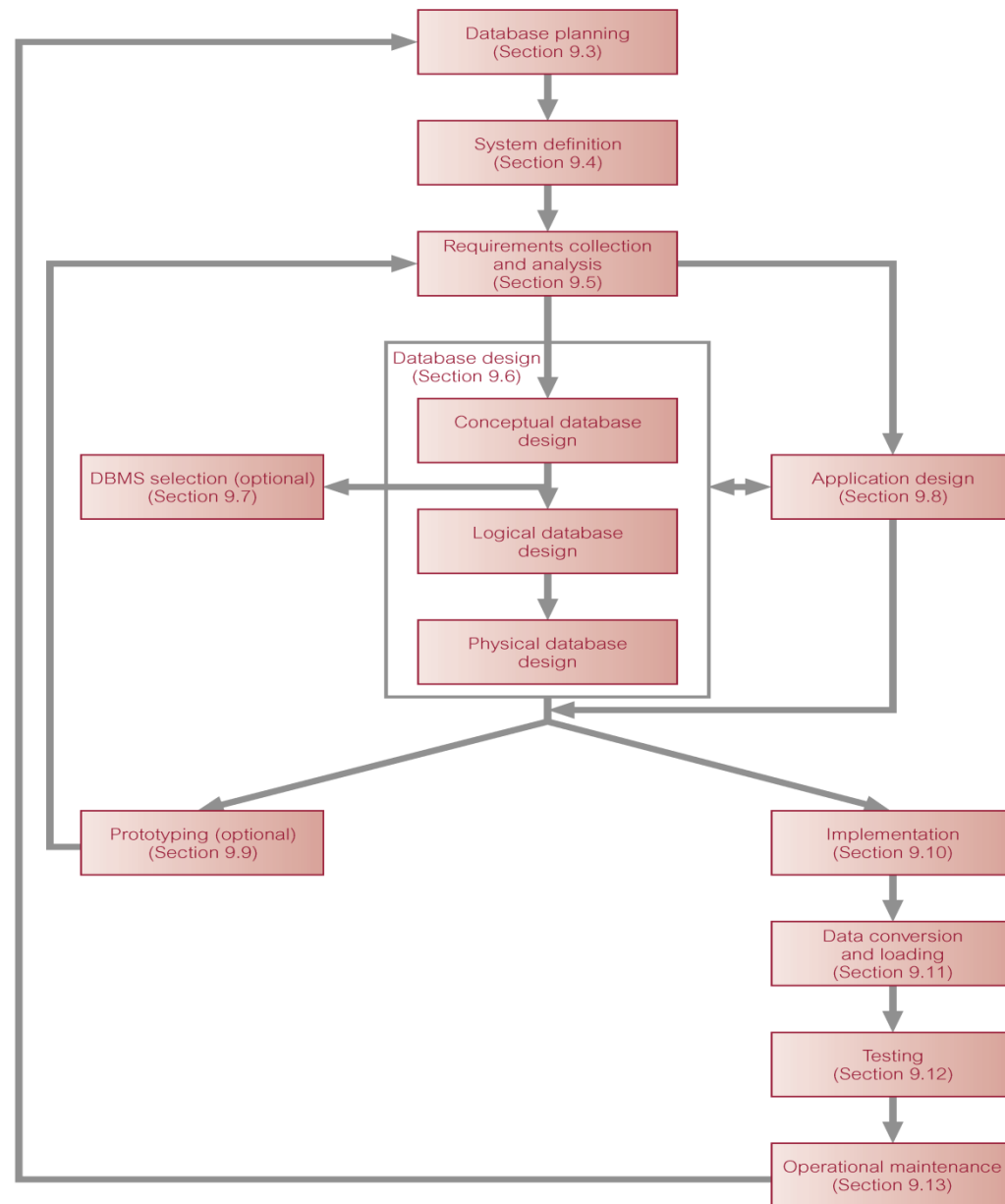# Database System
Development Lifecycle

Stages of the Database System Development Lifecycle

Page 264

Database System Development Lifecycle

- ➢ **Database planning**

- ➢ **System definition**

- ➢ **Requirements collection and analysis**

- ➢ **Database design (conceptual, logical, physical)**

- ➢ **DBMS selection (optional)**

Database System Development Lifecycle

- ➢ **Application design**

- ➢ **Prototyping (optional)**

- ➢ **Implementation**

- ➢ **Data conversion and loading**

- ➢ **Testing**

- ➢ **Operational maintenance**

Database Planning – Mission Statement

➤ *Mission statement* **defines major aims of the database application and helps clarify its purpose.**

Ex.

"The purpose of our HW database system is to maintain the data that is used to support hotel room rentals."

Database Planning – Mission Objectives

➢ **Once mission statement is defined,** *mission objectives* **are defined which should identify a particular task that the database must support.**

➢ **Ex.**

➢ **To maintain (insert, update, delete) data on the hotels, rooms, guests, and bookings.**

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 the perspective of:**

   – **a particular job role (such as Manager or Supervisor) or**

   – **enterprise application area (such as marketing, personnel, etc.).**

Representation of a Database System with Multiple User Views
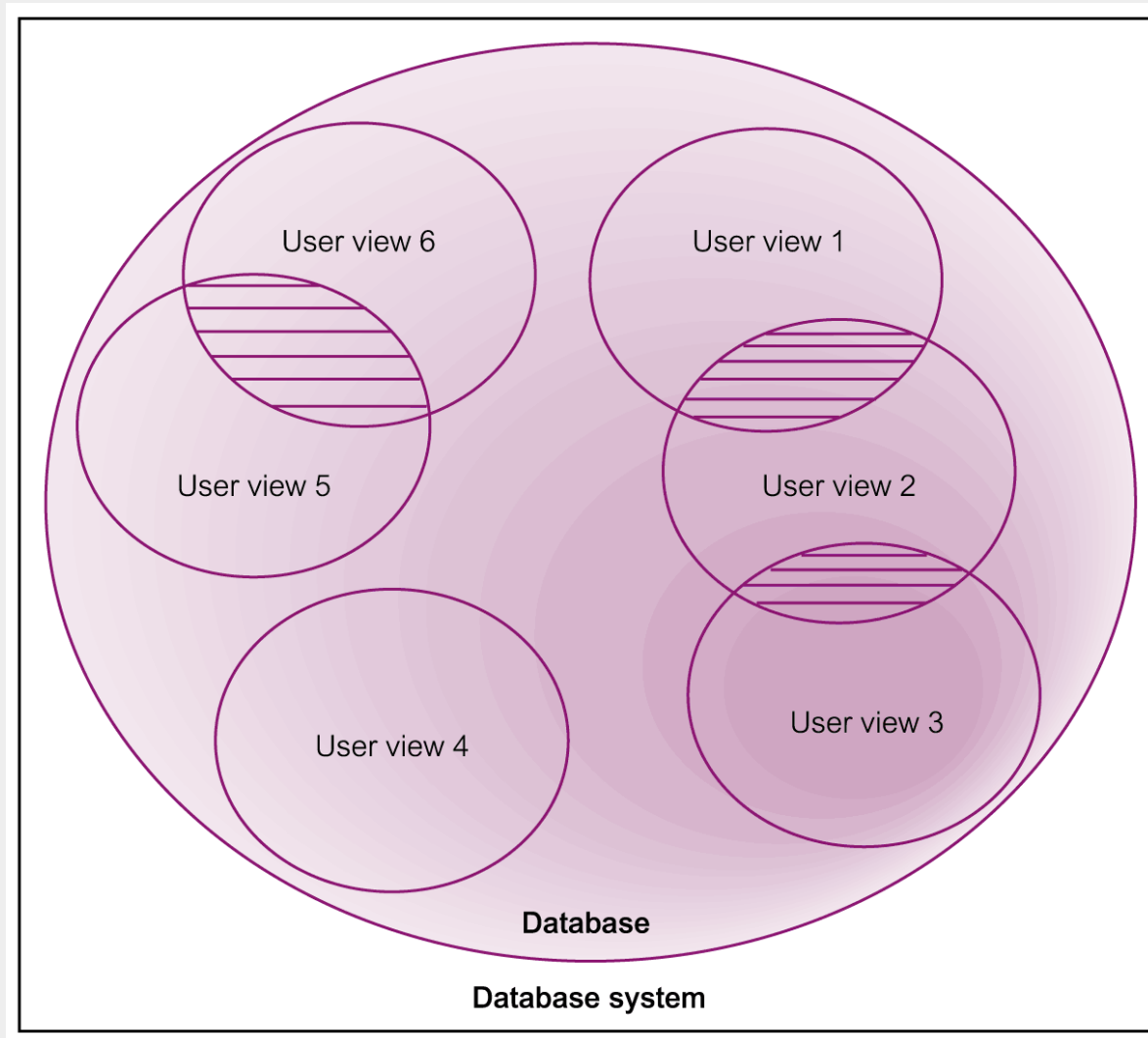
Requirements Collection and Analysis

➢ **Get user requirements - collect and analyze information about the part of organization to be supported by the database system.**

➢ **These requirements/features for the new database system are described in documents known as the requirements specifications.**

➢ **Many techniques for gathering this information (fact-finding techniques) discussed in the next chapter.**

Database Design

➢ **Database Design:** Creating a design for a database that will support the mission statement and mission objectives.

➢ **Data Modeling is in the Database Design Phase.**

➢ Building data model requires answering questions about entities, relationships, and attributes.

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, independent of *all* physical considerations.**

➢ **Conceptual data model is built using the information in users' requirements specification.**

➢ **Ex. ER Diagram**

Logical Database Design

➢ Conceptual data model is independent of all physical considerations, a **logical model** is derived knowing the underlying **data model** of the target **DBMS.**

➢ **Ex. relational data model, normalization**

Physical Database Design

➢ The physical design of the database specifies a description of the physical configuration of the database, such as the tables, file organizations, indexes, security, data types, and other parameters in the data dictionary.

➢ **To describe how we intend to physically implement the logical database design.**

DBMS Selection

➢ **Selection of an appropriate DBMS to support the database system (if none exist).**

➢ **Undertaken at any time prior to logical design provided sufficient information is available regarding system requirements.**

➢ **Check off DBMS features against requirements.**

➢ **Some DBMS examples include MySQL, Microsoft Access, SQL Server, Oracle.**

Example 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:2003/ODMG compliant	Backup and recovery routines
Interfacing to 3GLs	Checkpointing facility
Multi-user	Logging facility
Security	Granularity of concurrency
– Office Access controls	Deadlock resolution strategy
– Authorization mechanism	Advanced transaction models
	Parallel query processing

16

Application Design

➢ **Design of user interface and application programs that use and process the database.**

➢ **Database design and application design are parallel activities.**

Prototyping (Optional)

➢ **Building working model of a database system.**

➢ **Does not contain all the required features.**

➢ **Purpose**
   – **to identify features of a system that 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.**

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 the intent of finding errors.**

➢ **Use carefully planned test strategies and realistic data.**

➢ **Demonstrates that database and application programs *appear* to be working according to requirements.**

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 system (when required). Incorporating new requirements into database application.**