Database Review
Design and Implementation Process

Lecture 1

Outline

- What is a Database?
- Database Management Systems
- DBMS in Context: People, Architecture
- Database Design and Implementation Process

What is a Database?

A collection of related data, most often…

- reflects some aspect of the real world
- logically coherent with inherent meaning
- designed, built, and populated with data for a specific purpose
  - intended group of users
  - some preconceived applications with which these users are interested
  - application requirements in terms of performance, security, redundancy, concurrency, etc.

Database Management System

DBMS

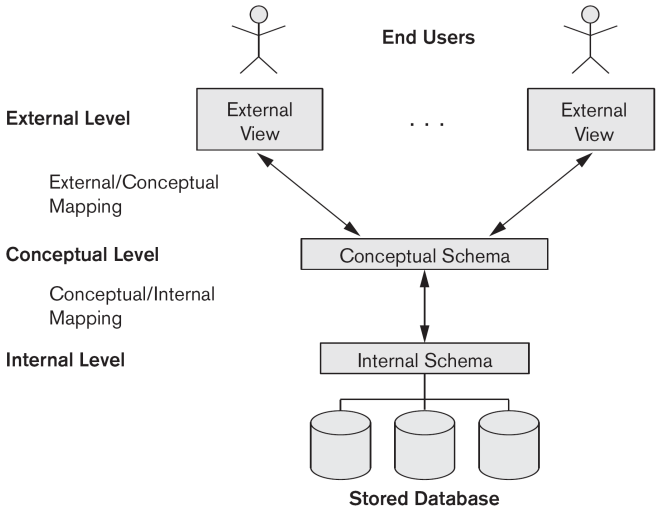
A collection of programs that enables users to create and maintain a database

- Supports specifying the data types, structures, and constraints of the data
- Stores the data on some medium under control of the DBMS
- Supports querying and updating the database
- Protects data against malfunction and unauthorized access

Typical DBMS Features (1)

- Data abstraction via **data models**
  - Conceptual representation independent of how underlying storage or operation implementation

Figure 2.2
The three-schema architecture.



Our Focus

- Conceptual Modeling: [E]ER
- Data Modeling: Relational

Typical DBMS Features (2)

- Operation abstraction via languages
  - Data… definition, manipulation, query
    - We will focus on SQL
  - Programmatic APIs
    - We will focus on function libraries
      - vs. embedded, stored procedures, etc.

Typical DBMS Features (3)

- Reliable transaction processing (not a focus)
  - (**A**)tomicity: "all or nothing"
  - (**C**)onsistency: valid -> valid'
  - (**I**)solation: parallel execution, serial result
  - (**D**)urability: once it is written, it is so
- High performance
  - Buffering, caching (not a focus)
  - Query optimization, redundant data structures (e.g. indexes, materialized views)

Typical DBMS Features (4)

- Authentication and authorization subsystems
  - Discussed lightly, in context of other security concerns/techniques
- Backup and recovery
  - Not discussed

DBMS in Context: People

1. **Database designers**
2. **System analysts & application programmers**
3. Database administrators
4. End users
5. Back-end
   a. DBMS designer/implementer
   b. Tool developers
   c. SysAdmins

DBMS in Context: Architecture

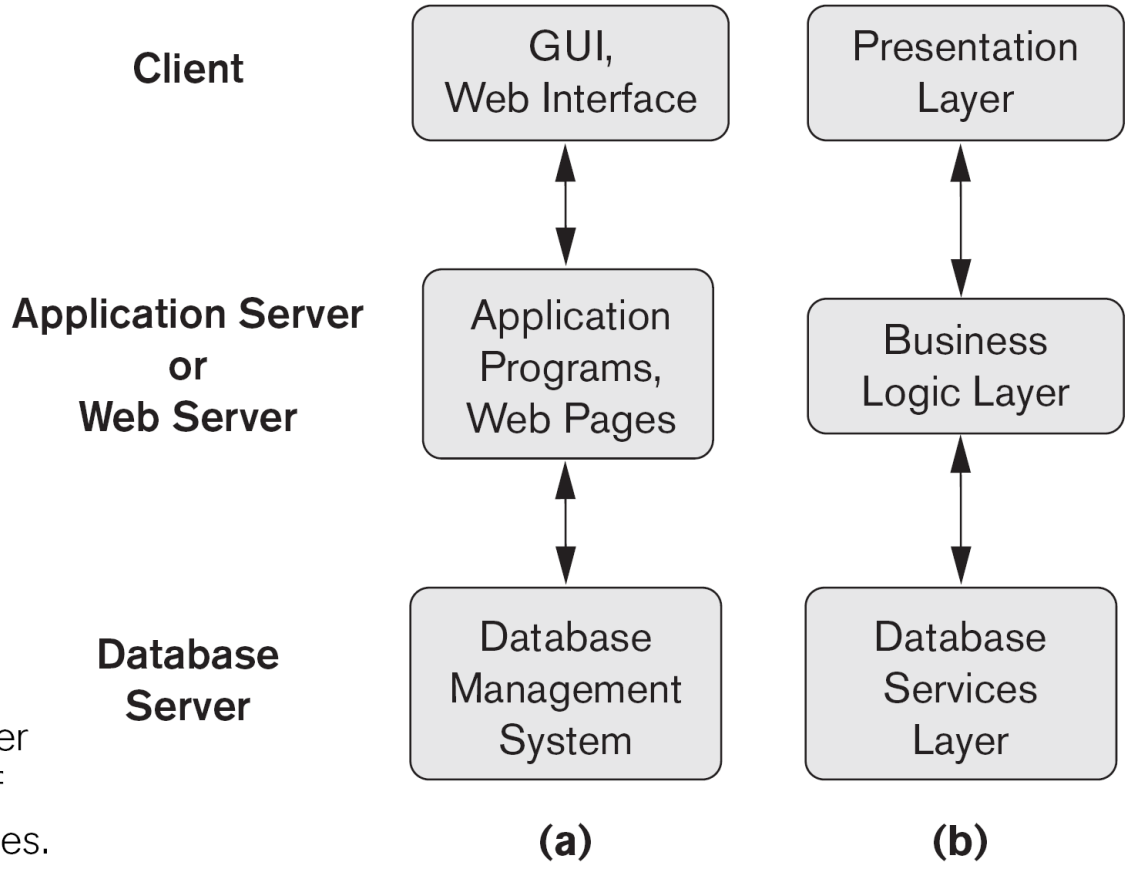


Figure 2.7
Logical three-tier client/server architecture, with a couple of commonly used nomenclatures.

When NOT to use a DBMS

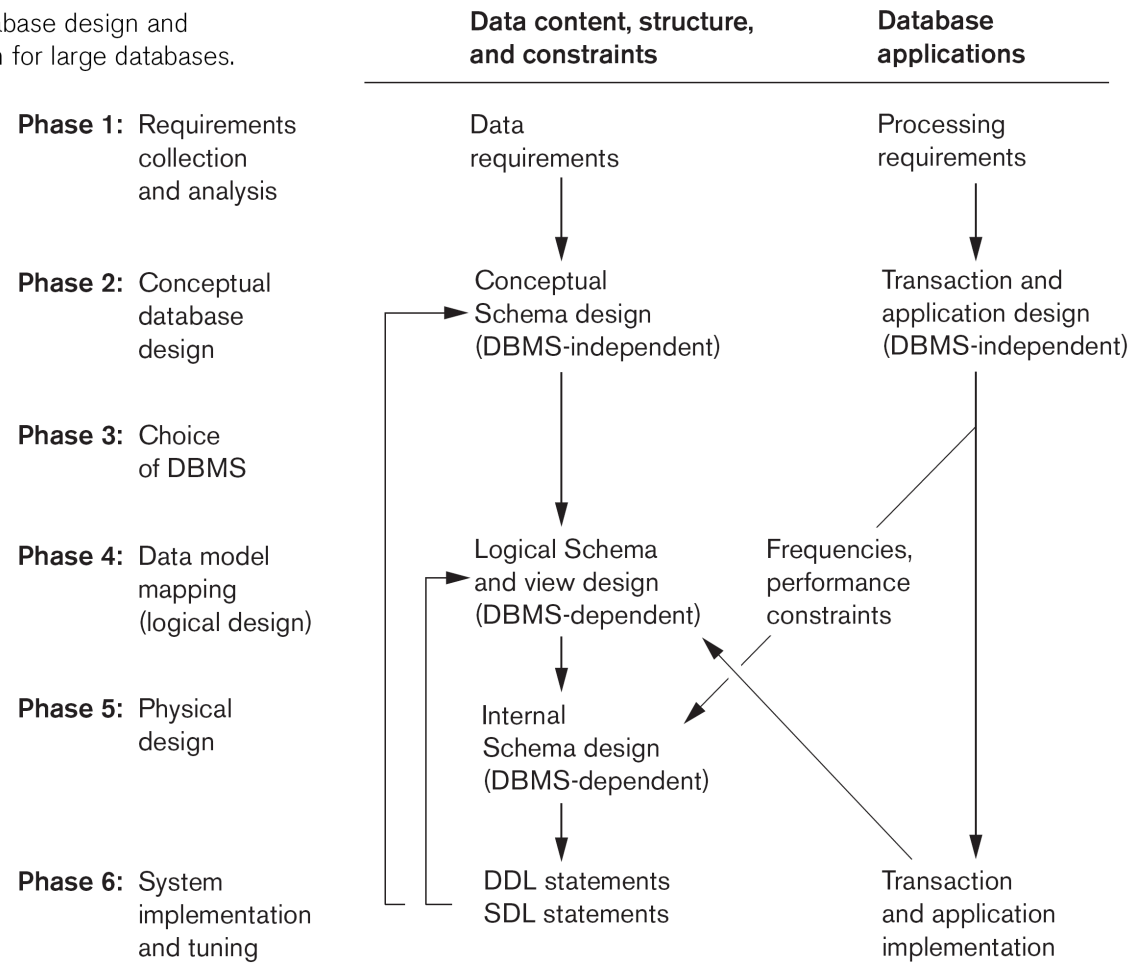
- Single-user data access
- Embedded/real-time systems
- Simple/well-defined (i.e. general DBMS may be overkill)

However, DBMS techniques may be useful

  - We will discuss useful and scalable indexing structures and processes

Database Design and Implementation Process

Figure 10.1
Phases of database design and implementation for large databases.



Requirements Collection & Analysis

- Data/Constraints

  "The company is organized into departments. Each department has a unique name, number, and a particular employee who manages the department. We keep track…"

- Functional Needs
  - Operations/queries/reports
    - Frequency
  - Performance, security, etc.

Conceptual Design

Data

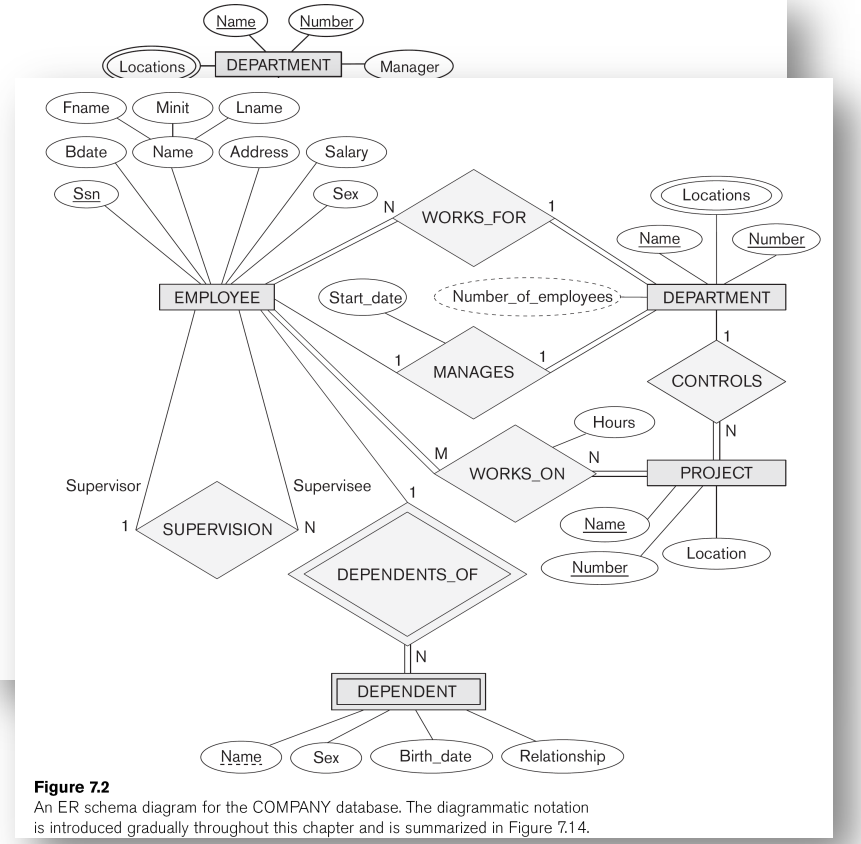


Figure 7.2
An ER schema diagram for the COMPANY database. The diagrammatic notation is introduced gradually throughout this chapter and is summarized in Figure 7.14.

Application

- Software
  - UML
  - Form design
- Database
  - Transaction design
  - Report design

Logical Design

Data

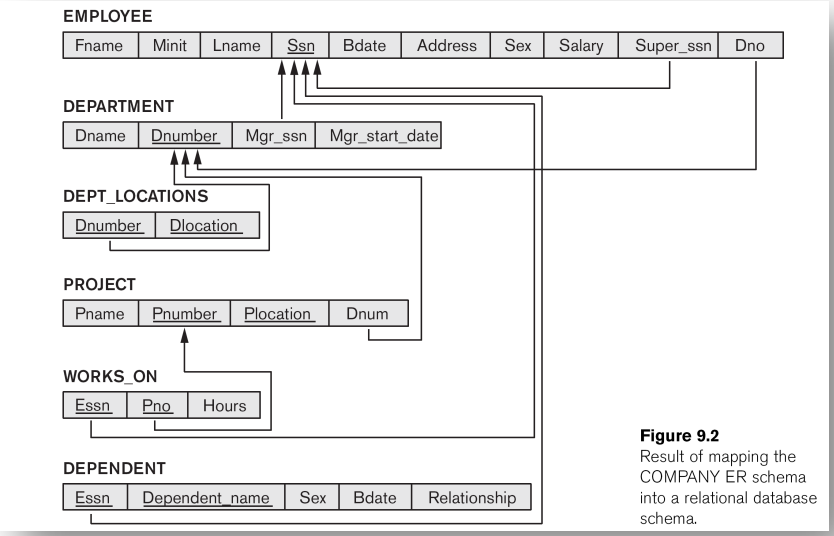


Figure 9.2
Result of mapping the COMPANY ER schema into a relational database schema.

- Normalization

Application

- Supporting code (that does not depend upon database)
  - Possibly using techniques from databases (e.g. indexing)

Physical Design

Data

- Index, materialized view selection and analysis

Application

- Implementing operations as queries
- Implementing constraints as keys, triggers, views
- Implementing multi-user security as grants

Implementation and Tuning

Data

- DDL statements
- De-normalization, updating indexes/ materialized views

Application

- Query integration
- Profiling queries/ operations
- Security, concurrency, performance, etc. analysis