DataBase System Concepts

Deng Fang

Dengfang@bupt.edu.cn

Introduction

- Book:

  Database System Concepts

- Teaching hours: total 68 hours, including

  lecture time: 48 hours

  practice time: 16 hours

Reference Book:

数据库系统概念　　　　　　机械工业出版社

数据库系统概论，王珊著，高等教育出版社

数据库系统原理，李建中，王珊编著，电子工业出版社

Contents(1)

Part 1 Introduction

Database Structure, Language, Users, Data Model…

Part 2 Relational Databases

Relational Model

SQL

Integrity and Security

Part 3 Database Design

E_R Model

Relational-database Design

Contents(2)

Part 4 Data Storage and Querying

Storage and File Structure

Indexing and Hashing

Querying Processing

Query Optimization

Part 5 Transaction Management

Transactions

Concurrency control

Recovery System

Contents(3)

Part 6 Database System Architecture

Part 7 Object-Based Databases and XML

Part 8 Case Studies

Part 9 Other Topics

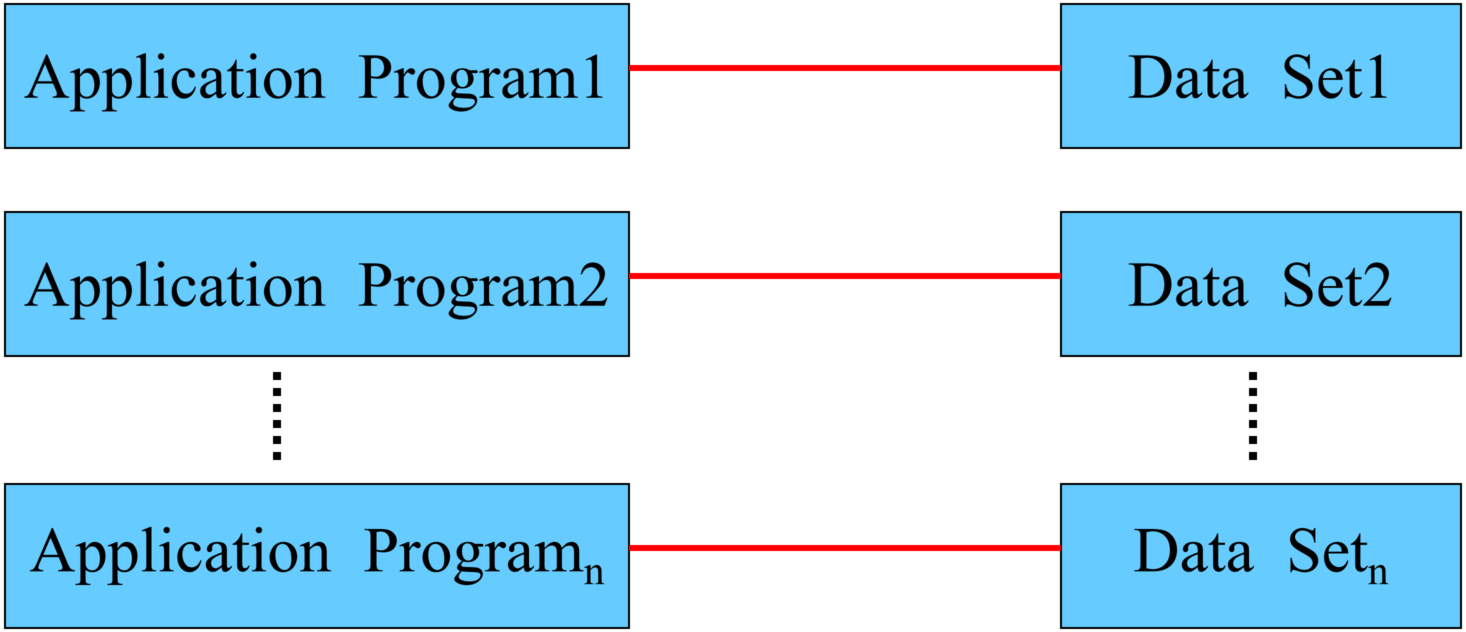
Chapter One Introduction

- History of Database systems
- What is Database systems
- Three levels and Data Abstraction
- Data Models
- Database languages
- Database users
- Database system structure

History of Database systems

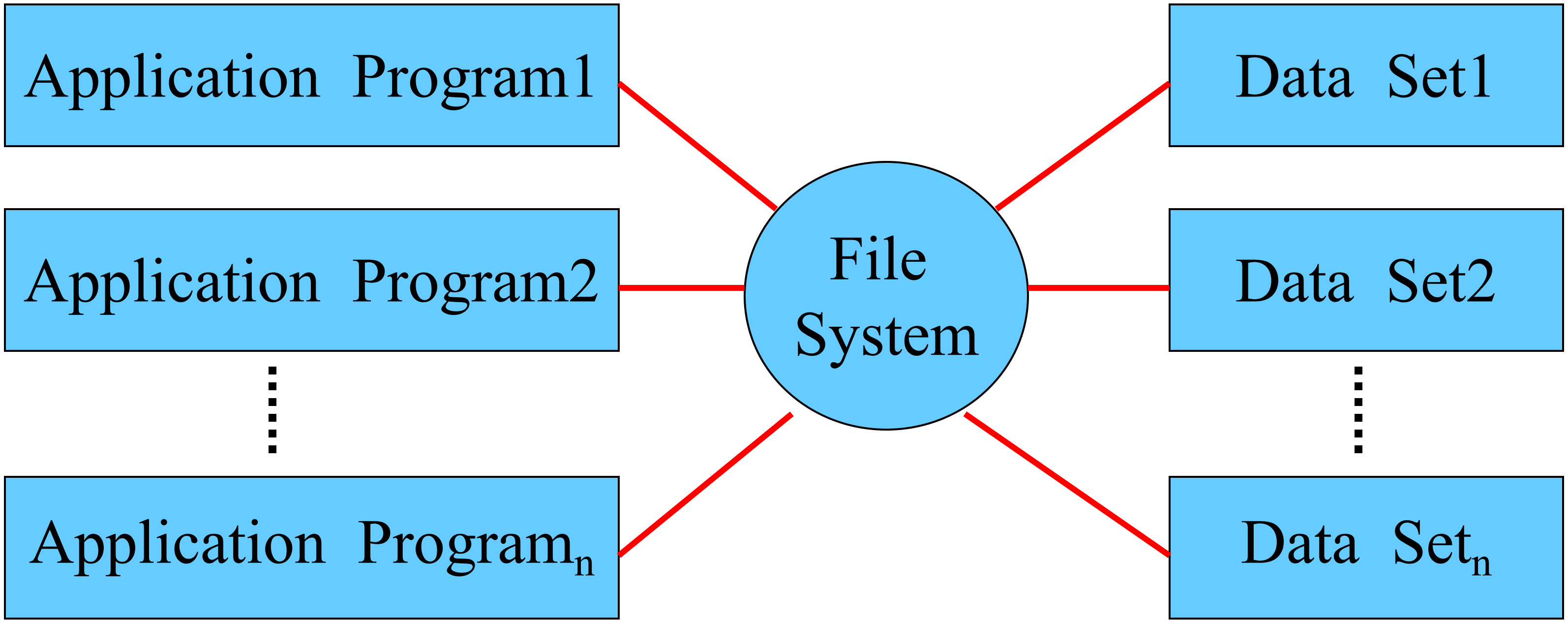
Development of Data Management

Manual Model



History of Database systems

File System



Application Program1

Application Program2

Application Program$_n$

File System

Data Set1

Data Set2

Data Set$_n$

History of Database systems

Drawbacks of using file systems to store Data:

- Data redundancy and inconsistency

  Multiple file formats, duplication of information in different files

- Difficulty in accessing data

  Need to write a new program to carry out each new task

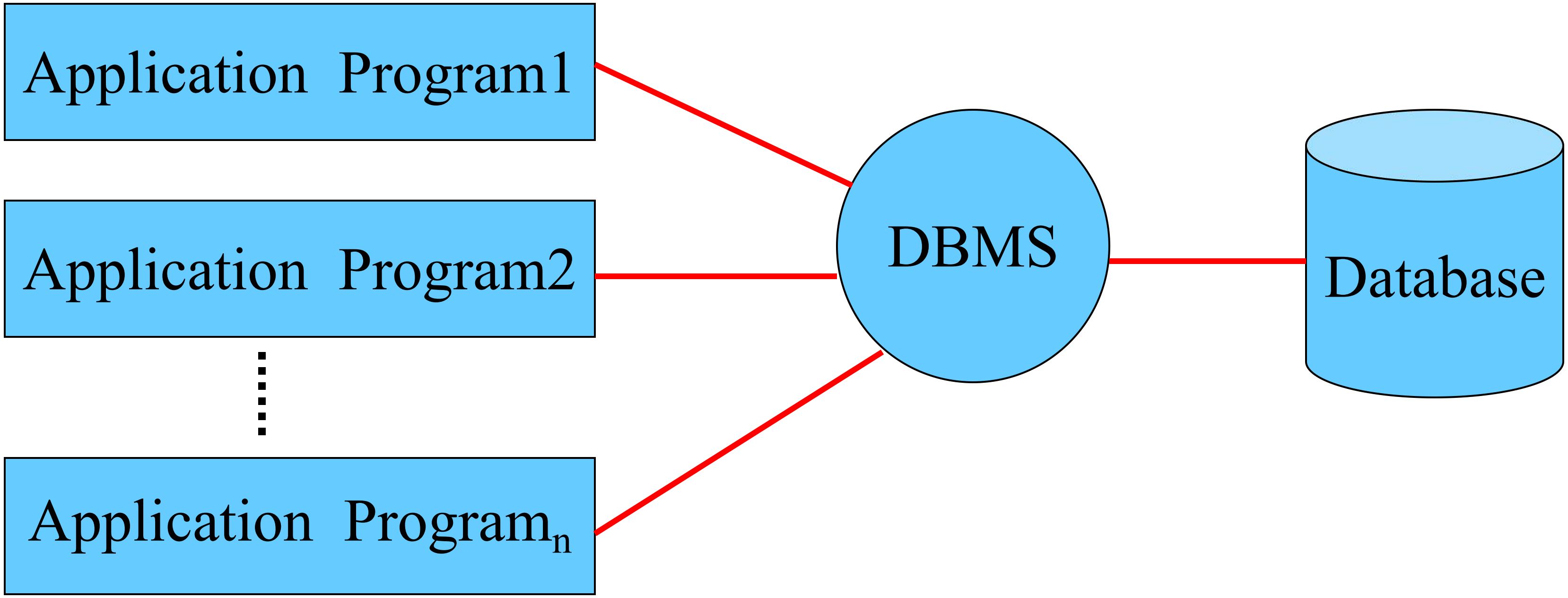
- Data isolation — multiple files and formats

History of Database systems

- Integrity problems
  - Integrity constraints (e.g. account balance > 0) become part of program code
  - Hard to add new constraints or change existing ones
- Atomicity of updates
- Concurrent access by multiple users
- Security problems

History of Database systems

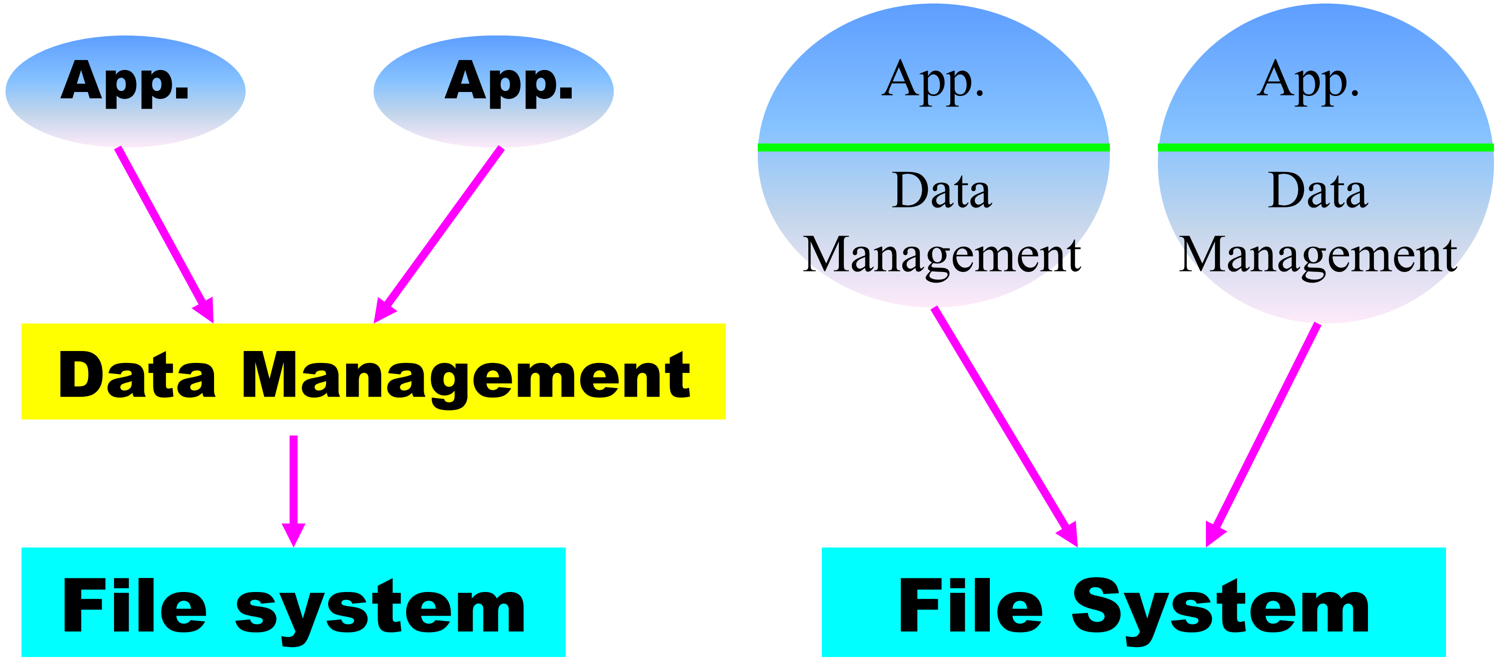
Database System



Benefits of Database Approach

- Data can be shared;
- Redundancy can be reduced;
- Inconsistency can be avoided(to some extent);
- Transaction support can be provided;
- Integrity can be maintained;
- Security can be enforced;
- …

Database System Vs File system



History of Database systems

- 1950s and early 1960s:
  - Data processing using magnetic tapes for storage
  - Punched cards for input
- Late 1960s and 1970s:
  - Hard disks allow direct access to data
  - Network and hierarchical data models in widespread use
  - Ted Codd defines the relational data model
    - Would win the ACM Turing Award for this work
    - IBM Research begins System R prototype
    - UC Berkeley begins Ingres prototype
  - High-performance (for the era) transaction processing

History of Database systems

- 1980s:
  - Research relational prototypes evolve into commercial systems
    - SQL becomes industrial standard
  - Parallel and distributed database systems
  - Object-oriented database systems
- 1990s:
  - Large decision support and data-mining applications
  - Large multi-terabyte data warehouses
  - Emergence of Web commerce
- Early 2000s:
  - XML and XQuery standards
  - Automated database administration

History of Database systems

- Later 2000s:

  -Giant data storage systems

  Google BigTable, Yahoo PNuts, Amazon, ..

History of Database systems

- 1968, IBM's IMS (Information Management system)

  hierarchical data model;
- 1970s, Cullinet Software's IDMS
- 1971, CODASYL(Conference on Data System Language) report network data model;
- 1970, E.F.Codd defined the relational model
- 1990s, object-relational database, WEB database, parallel database
- 2000s, NoSQL

What is Database systems

- **DB**: database

  a collection of interrelated data

- **DBMS**:

  DataBase Management System

- **Applications**

  Set of programs to access the data

What is Database systems

DBMS(教材上） contains information about a particular enterprise

- Collection of interrelated data
- Set of programs to access the data
- An environment that is both *convenient* and *efficient* to use

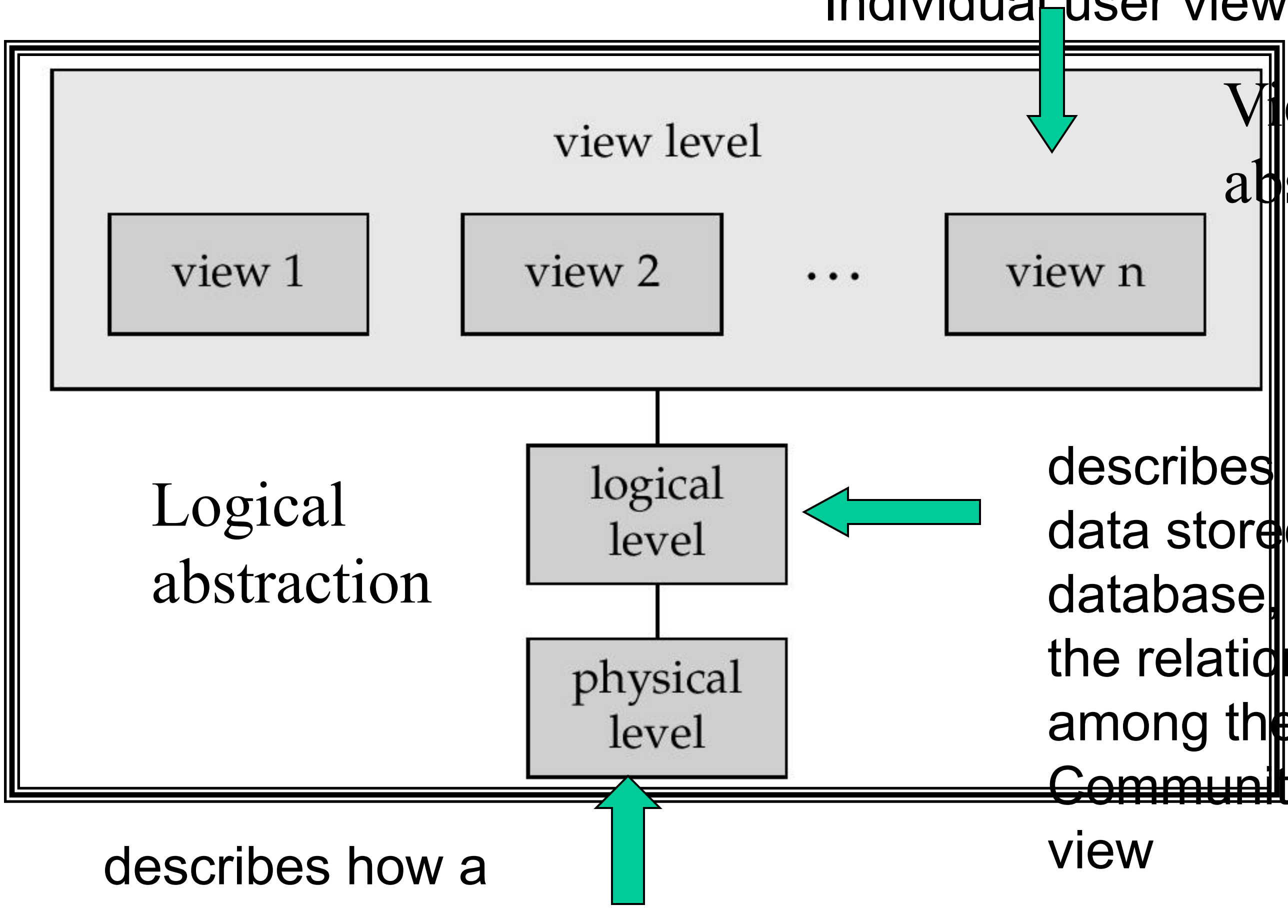
Database Applications

- Database Applications:
  - Banking: all transactions
  - Airlines: reservations, schedules
  - Universities: registration, grades
  - Sales: customers, products, purchases
  - Manufacturing: production, inventory, orders, supply chain
  - Human resources: employee records, salaries, tax deductions
- Databases touch all aspects of our lives

Three –level Architecture

Part of entire database

Individual user views



View abstraction

Logical abstraction

describes logical data stored in database, and the relationships among the data. Community user view

describes how a record is physically stored.

Physical abstraction

Levels of Abstraction

如何存储

- **Physical level:** describes how a record (e.g., customer) is stored.
- **Logical level:** describes data stored in database, and the relationships among the data. 数据之间的关系

```
type customer = record
            customer_id : string;
            customer_name : string;
            customer_street : string;
            customer_city : integer;
        end;
```

只描述数据库的部分

- **View level:** Part of the entire database. Application programs hide details of data types. Views can also hide information (such as an employee's salary) for security purposes.

Data Abstraction

- **View abstraction**
- **Logical abstraction**
- **Physical abstraction**

Mappings（映象）

- **The Physical(internal)/Logical(conceptual) mapping**

  defines the correspondence between the logical view and the stored database;

  specifies how logical records and fields are represented at the physical level;

- **The View(external)/Logical(conceptual) mapping**

  defines the correspondence between a particular view and the logical view;

Data Independence

- **Physical Data Independence** （物理数据独立性）–

  the ability to modify the physical schema without changing the logical schema or the views

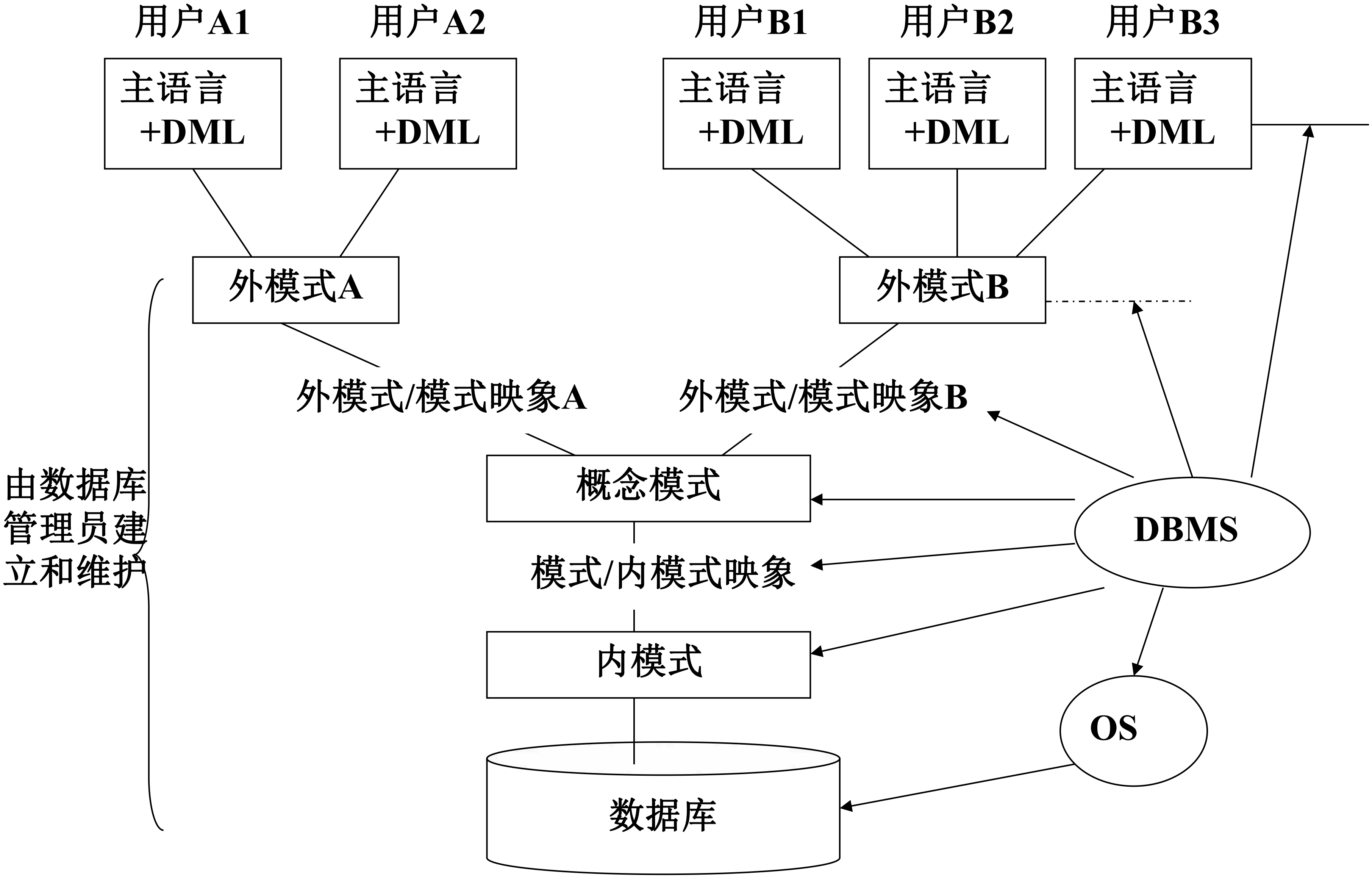
  (The Physical(internal)/Logical(conceptual) mapping)

  Application programs need not be rewritten if the physical schema changes.

- **Logical Data Independence** （逻辑数据独立性）–

  the ability to modify the logical schema without changing the logical views

  (The View(external)/Logical(conceptual) mapping)


用户A1
用户A2
用户B1
用户B2
用户B3
主语言 +DML
主语言 +DML
主语言 +DML
主语言 +DML
主语言 +DML
外模式A
外模式B
外模式/模式映象A
外模式/模式映象B
由数据库管理员建立和维护
概念模式
DBMS
模式/内模式映象
内模式
OS
数据库

数据库系统的体系结构

Instances and Schemas

- **Schema** （模式）– the description of the structure of the data in a database
  - e.g., the database consists of information about a set of customers and accounts and the relationship between them);
  - Analogous to type information of a variable in a program;
  - **Physical schema**: database design at the physical level;
  - **Logical schema**: database design at the logical level;
- **Instance**（实例） – ~~the actual content of~~ the database at a ~~particular point in~~ time
  - Analogous to the value of a variable

Data Models

A collection of conceptual tools for describing **data, data relationships,data semantics ,and consistency constrains.**

Data Structure;

Data Operations;

Constraint Rules.

Data Models

Object_based data model:

- Entity-Relational Model (E_R Model);
- Object-oriented Model;

Record_based data model:

- hierarchical data model;
- network data model;
- relational data model

Relational Data Model

Customer-id	*customer-name*	*customer-street*	*customer-city*	*account-number*
192-83-7465	Johnson	Alma	Palo Alto	A-101
019-28-3746	Smith	North	Rye	A-215
192-83-7465	Johnson	Alma	Palo Alto	A-201
321-12-3123	Jones	Main	Harrison	A-217
019-28-3746	Smith	North	Rye	A-201

Database Languages

- **Data Definition Language (DDL):**
- Specification notation for defining the database schema; Eg:

```
Create table account (
            account-number    char(10),
            balance          integer)
```

- Including:
- Database schema
- Data *storage and definition* language

Database Languages

Data Manipulation Language (DML):

- Language for accessing and manipulating the data organized by the appropriate data model;

(query, insert , delete, update)

DML also known as query language

- Two classes of languages

  Procedural – user specifies what data is required and how to get those data ;

  Nonprocedural – user specifies what data is required without specifying how to get those data;

- SQL is the most widely used query language

Database Users

Users are differentiated by the way they expect to interact with the system

- Application programmers – interact with system through DML calls
- Sophisticated users – form requests in a database query language
- Specialized users – write specialized database applications that do not fit into the traditional data processing framework
- Naive users – invoke one of the permanent application programs that have been written previously
  - –E.g. people accessing database over the web, bank tellers, clerical staff

Database Users

- End Users:

  naive Users

  casual users -----Interactive SQL
- Application Programmers-----Procedural SQL,Transaction
- Database analyzer and designer -----Data modeling,

  Normalization theory,
- Database Administrators, DBA -----Database maintenance ,

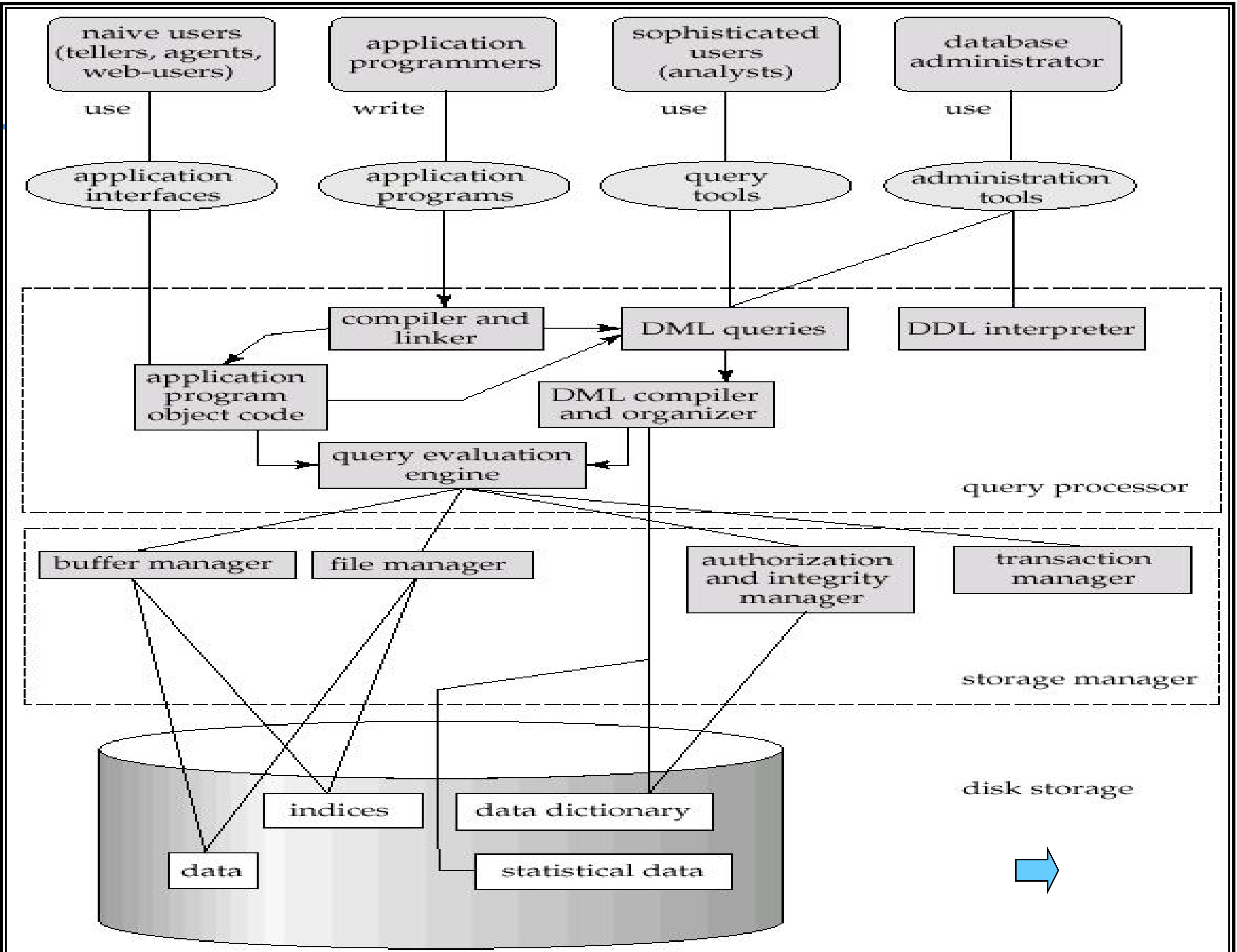
  Security , Integrity,Recovery
- Database Management System designer and implementer -----

  Implementation technique of above techniques for Special and New Database Management System

Database Administrator

DBA

in requirements


naive users (tellers, agents, web-users)
application programmers
sophisticated users (analysts)
database administrator
use
write
use
use
application interfaces
application programs
query tools
administration tools
compiler and linker
DML queries
DDL interpreter
application program object code
DML compiler and organizer
query evaluation engine
query processor
buffer manager
file manager
authorization and integrity manager
transaction manager
storage manager
indices
data dictionary
data
statistical data
disk storage

Database System Structure

Storage Management:

- Storage manager is a program module that provides the interface between the low-level data stored in the database and the application programs and queries submitted to the system.
- is responsible for efficient storing, retrieving and updating of data;
  - Authorization and integrity manager;
  - Transaction manager;
  - File manager;
  - Buffer manager;
- Data Structure
  - Data files;
  - Data dictionary;
  - Indices;

Database System Structure

Query Processor:

DDL interpreter;

DML complier;

Query optimization;

Query evaluation engine;

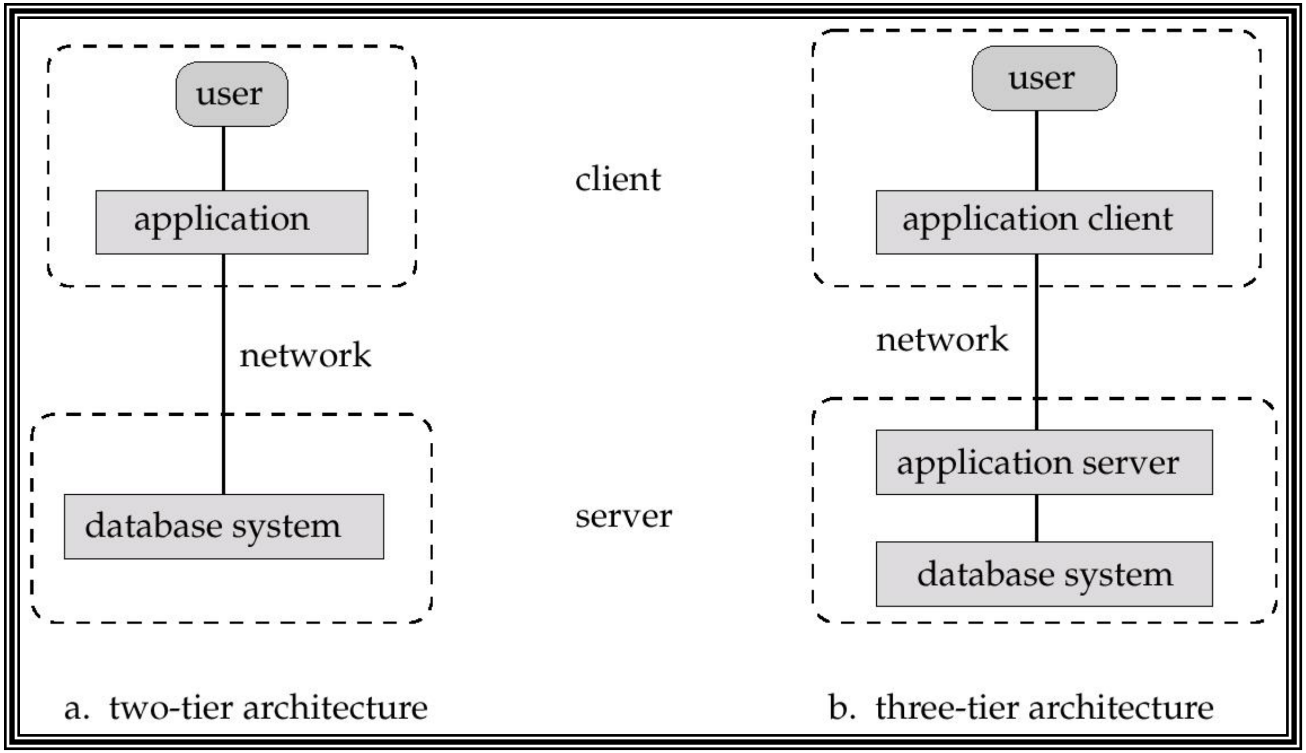
Transaction Management

- A *transaction* is a collection of operations that performs a single logical function in a database application
- Transaction-management component ensures that the database remains in a consistent (correct) state.
- Concurrency-control manager controls the interaction among the concurrent transactions, to ensure the consistency of the database.

Functions of DBMS

- **Data Definition and storage management;**
- **Data Manipulation, Data Access;**
- **Data Security and integrity;**
- **Transaction management, Data recovery and concurrency;**
- **Data dictionary;**

Application Architectures



- **Two-tier architecture**: E.g. client programs using ODBC/JDBC to communicate with a database
- **Three-tier architecture**: E.g. web-based applications, and applications built using "middleware"

Summary

- Database System structure
  - DB
  - DBMS, DBMS Functions
  - Three-level structure
  - Data independence
- Data Model
- Database Languages
- Database Users