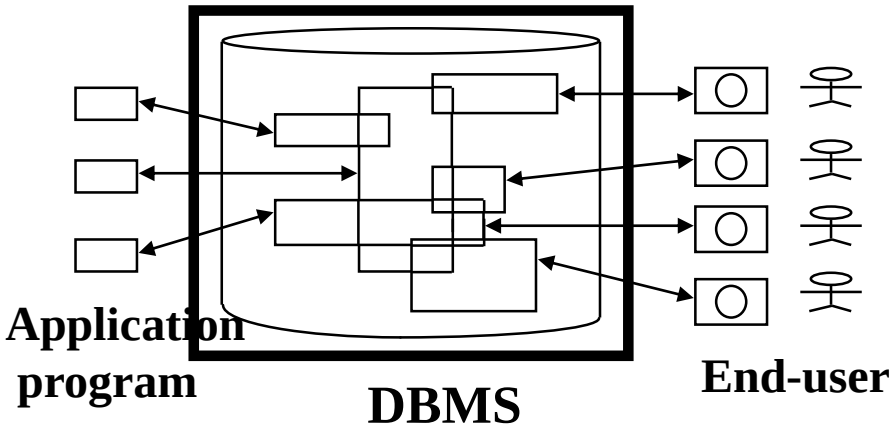
Unit 1

Introduction to DBMS

(Database Management Systems)



本課程講授內容

- **PART I: 入門與導論**
  - Overview
  - DB2系統及SQL語言
  - 闡述關連式資料模型(The Relational Model)
  - 階層式資料模型(The Hierarchical Model)簡介
  - 網狀式資料模型(The Network Model)簡介
- **PART II: 資料庫設計 (Database Design)**
  - 資料庫問題分析與 E-R Model
  - 資料庫的表格正規化
  - 設計介面增刪查改資料庫
- **PART III: 進階探討**
  - 快速存取方法(Access Methods)
  - 資料庫回復(Database Recovery)
  - 協同控制(Concurrency Control)
  - 資料安全與資料正確(Security and Integrity)
  - 查詢最佳化(Query Optimization)
  - 分散式資料庫系統(Distributed Database)

PART I: 入門與導論

- DB2系統及SQL語言:
  - 介紹最多人使用的查詢語言SQL
  - 配合實作習題，先試用一個免費的DBMS系統MySQL
  - 好比要學開車可先在大停車場讓你繞一圈
  - 這樣我們在介紹下單元關連式資料模型的設計原理時會比較容易想像
- 關連式資料模型(The Relational Model):
  - 闡述使用者視資料庫為許多表格(tables)組成的關連式資料庫之原始設計原理
  - 這是關連式資料庫的理論基礎
- 階層式資料模型(The Hierarchical Model)及網狀式資料模型(The Network Model):
  - 這是最早的二個資料模型
  - 介紹這二模型將有助於我們對資料庫模型更深入了解，並知其來龍去脈
  - 我們將簡要的說明其原始設計原理

Contents of PART I: 入門與導論

- **Unit 1 Introduction to DBMS**
- **Unit 2 DB2 and SQL**
- **Unit 3 The Relational Model**
- **Unit 4 The Hierarchical Model**
- **Unit 5 The Network Model**

- **References:**

1. C. J. Date, An Introduction to Database Systems, 8th edition, 2004.
2. J. D. Ullman, Principles of Database and Knowledge-Base, Vol.I, 1988.
3. Cited papers

Outline of Unit 1

1.1 Information Systems

1.2 An Overview of a Database System

1.3 Why Database Systems?

1.4 An Architecture for a Database System

1.5 Data Models

1.6 Establish/Design a Database System

1.7 Extending Database Technology

1.8 Discussion and Remarks

1.1 Information Systems

Stages of Information System

- Stage 0: Manual Information System
  - Records
  - Files
  - Index Cards
- Stage 1: Sequential Information Systems
  - Tapes
  - Files
  - slow, non-interactive, redundancy,... .
- Stage 2: File Based Information Systems
  - Disk (direct access)
  - application program has its own file ⇨ data dependence
  - data redundancy
- Stage 3: **DBMS based** Information Systems
  - Generalized data management software
  - Transaction processing

Stage 0: Manual Information System

- 圖書館index card

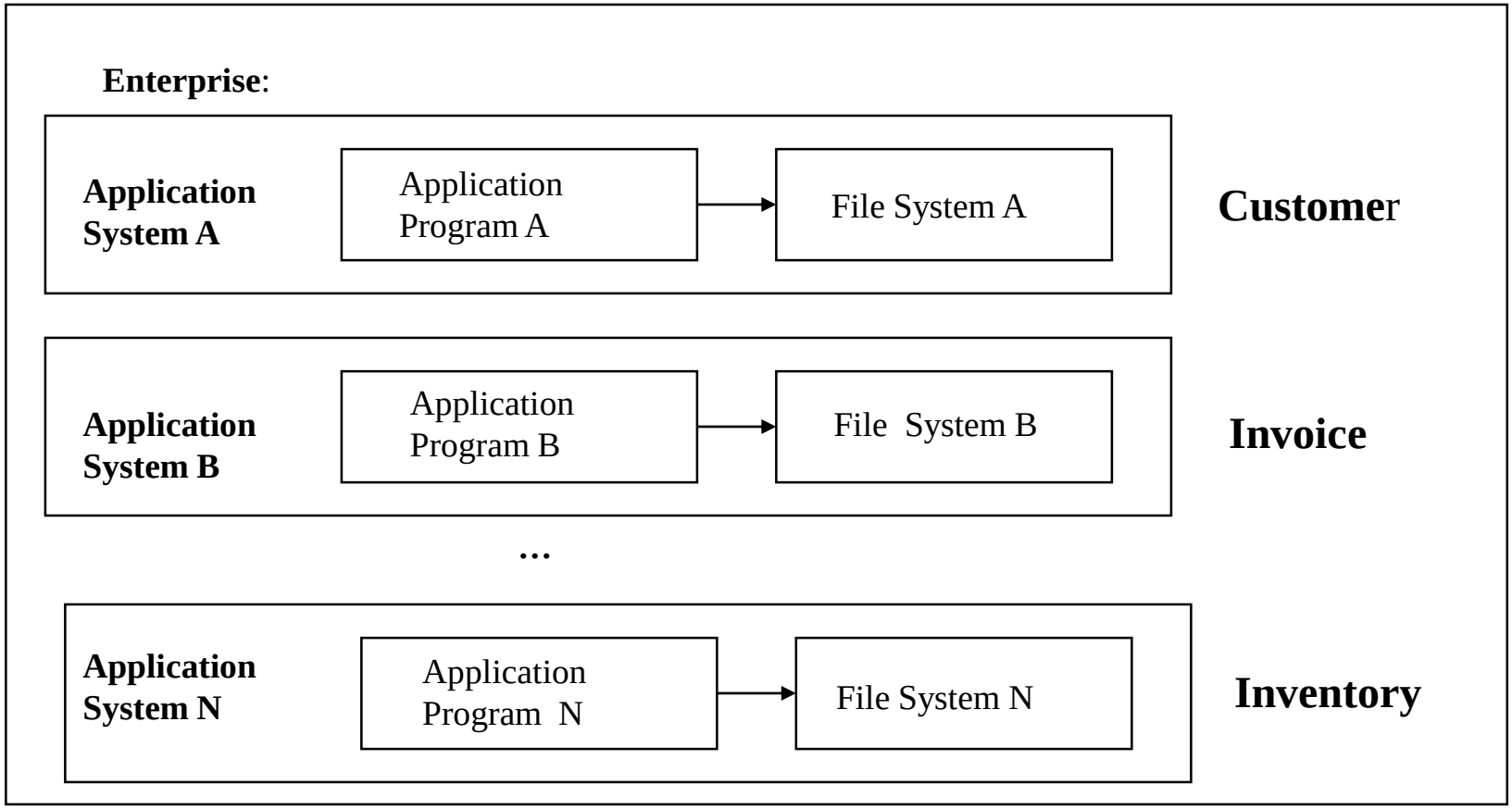
- 醫院診所病歷卡

Stage 1: Sequential Information Systems

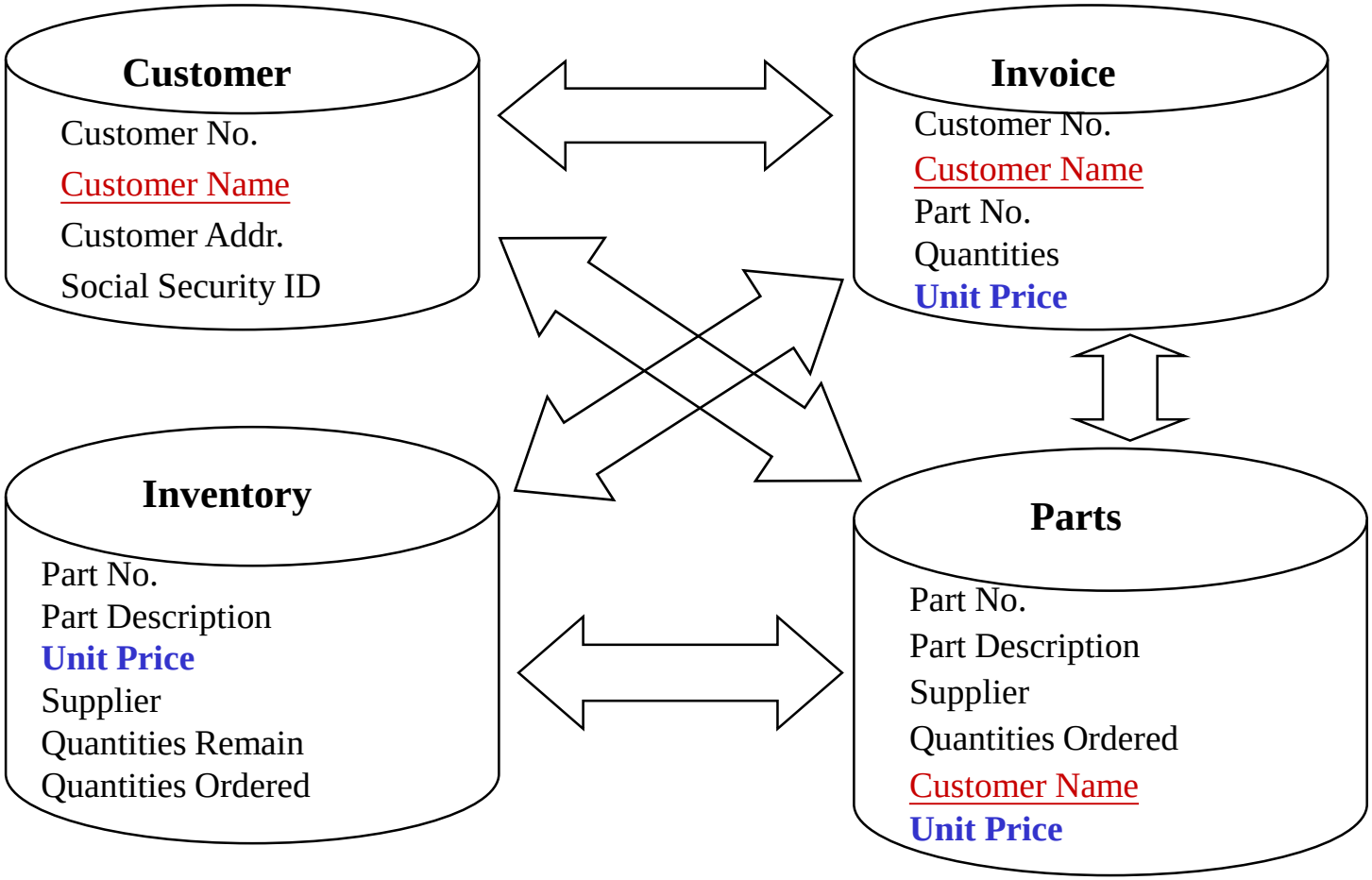
- *The old computer data center at NASA's Jet Propulsion Laboratory ...*

Stage 2: File Based Information Systems

- Conventional **Data Processing** techniques:



Stage 2: File Based Information Systems (cont.)



Stage 2: File Based Information Systems (cont.)

- Advantages: File Systems are simple in design
- Disadvantages:
  - Data Redundancy:
    - a waste of memory
    - high update cost
    - data inconsistency
  - Data Incompleteness
  - Data Insecure
  - Application Program Unstable
    - file system (application program)

      data changed → data structure changed

      → program changed

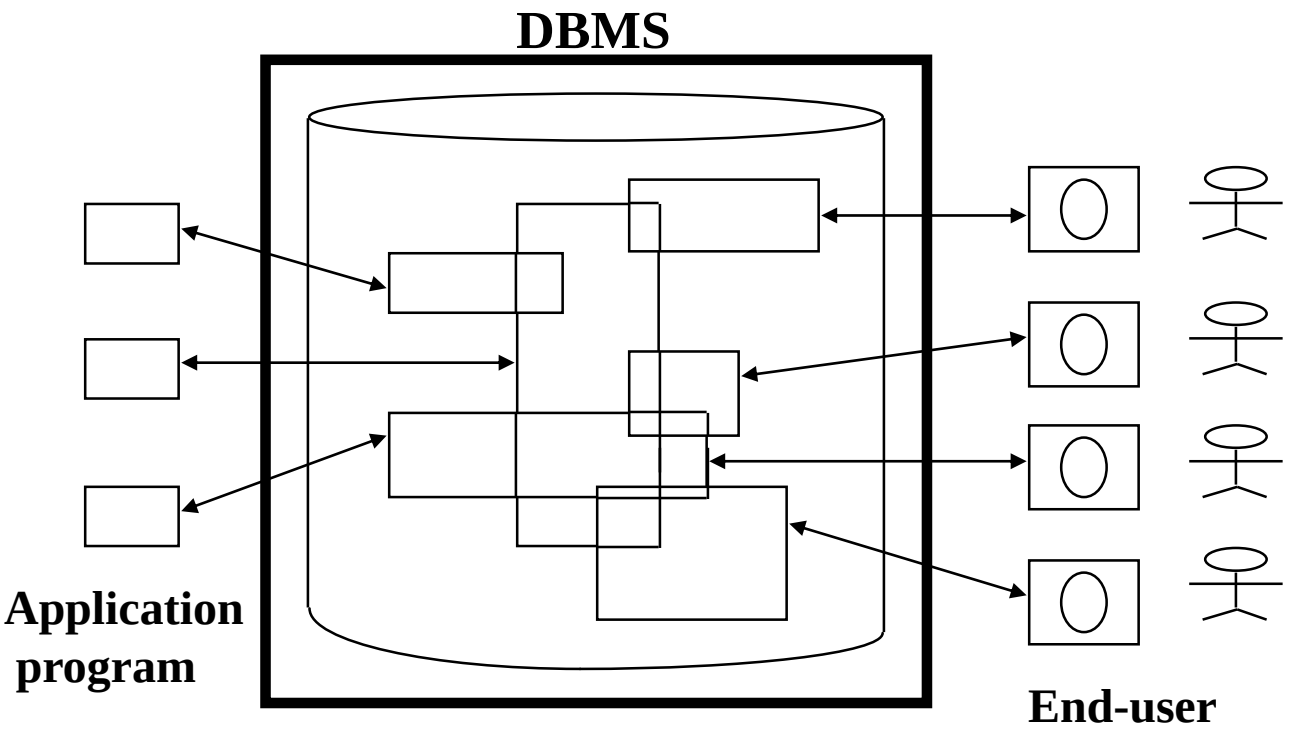
Solution: Database Systems!

- Objectives of Database Systems:
  - eliminate unnecessary data redundancy
  - maintain data integrity
  - control security
  - achieve data independence
  - add program stability

Stage 3: DBMS based Information Systems:

Basic Approach - Integration

- (1) Integration of Information
  - Description of the *integrated view* of data is the "*Conceptual Schema*" of the database



Stage 3: DBMS based Information Systems:
Basic Approach – Simple views and High level language

- (2) Provide simple views (External Schema) and high level language (e.g. SQL) for users to manipulate (handle) data
  - High level language: e.g. **SQL** (Structured Query Language)

```
<e.g.>:   SELECT SNAME
          FROM S
          WHERE S#= 'S4';
```

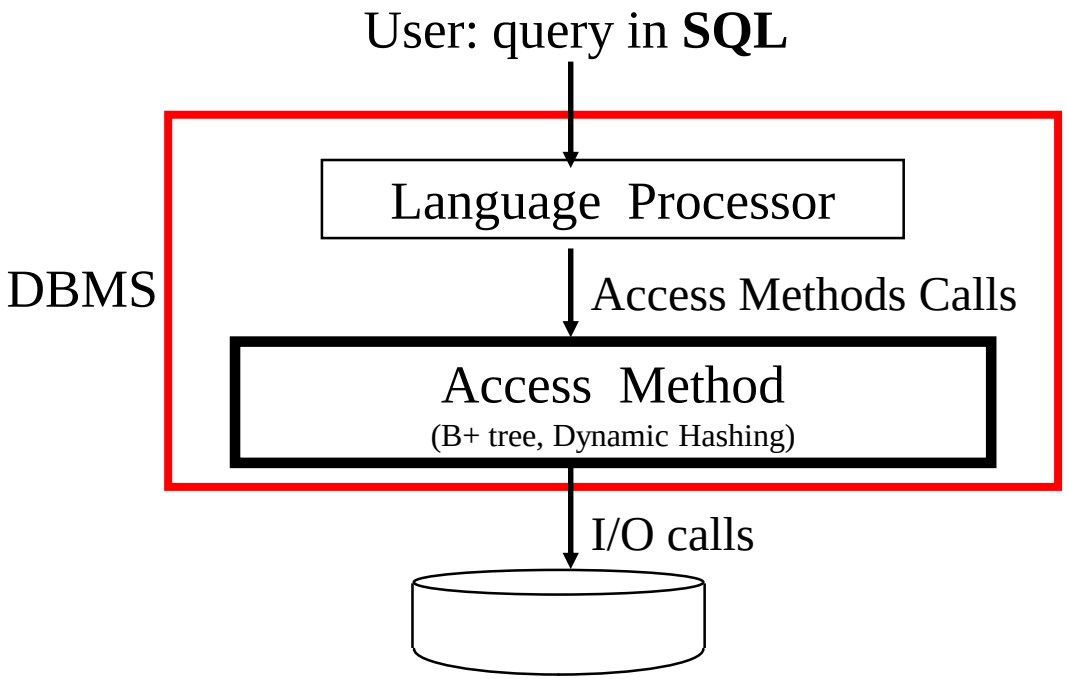
  - Description of user's view of data is the "external schema" or "subschema" or "view".
  - High-level languages (Query Language): **SQL**
    (1) Data Definition Language:
    define format
    (2) Data Manipulation Language:
    retrieve, insert, delete, update
  - Emphasize: *EASE OF USE !!*

S

S#	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

Stage 3: DBMS based Information Systems:
Basic Approach - Storage/Access Method

- (3) Efficient Storage/Access Techniques:
  - implemented once rather than duplicated in all application programs.



Stage 3: DBMS based Information Systems:
Basic Approach - Transaction Management

- (4) Provide Transaction Management:
  - Concurrency Control
  - Recovery
  - Security
  - .
    .
    .

Example: A Simple Query Processing

Query in SQL :

```
SELECT CUSTOMER. NAME
FROM CUSTOMER, INVOICE
WHERE REGION = 'N.Y.' AND
  AMOUNT > 10000 AND
  CUTOMER.C#=INVOICE.C
```

Internal Form :

$$\Pi(\sigma(S \bowtie SP)$$

Operator :

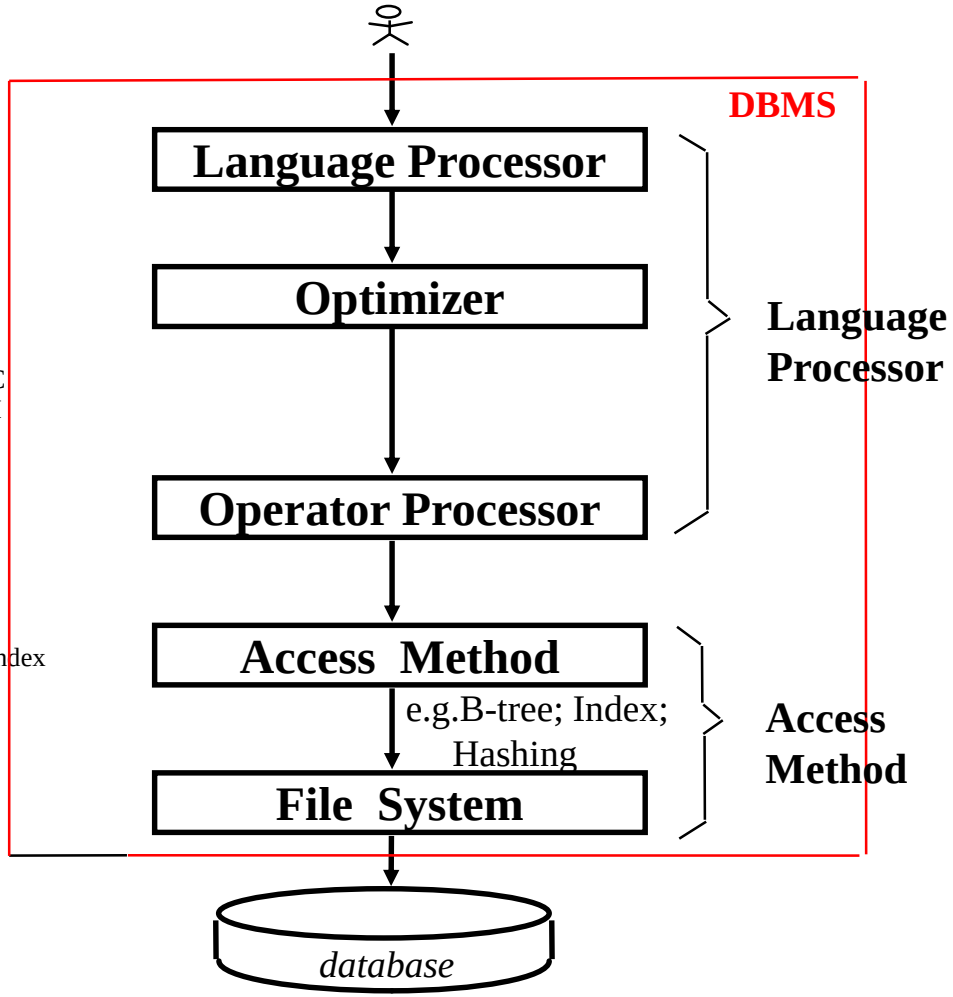
```
SCAN C using region index, create C
SCAN I using amount index, create I
SORT C?and I?on C#
JOIN C?and I?on C#
EXTRACT name field
```

Calls to Access Method :

```
OPEN SCAN on C with region index
GET next tuple
      .
      .
      .
```

Calls to file system :

```
GET10th to 25th bytes from
  block #6 of file #5
```



1.2 An Overview of a Database System

Database System: Introduction

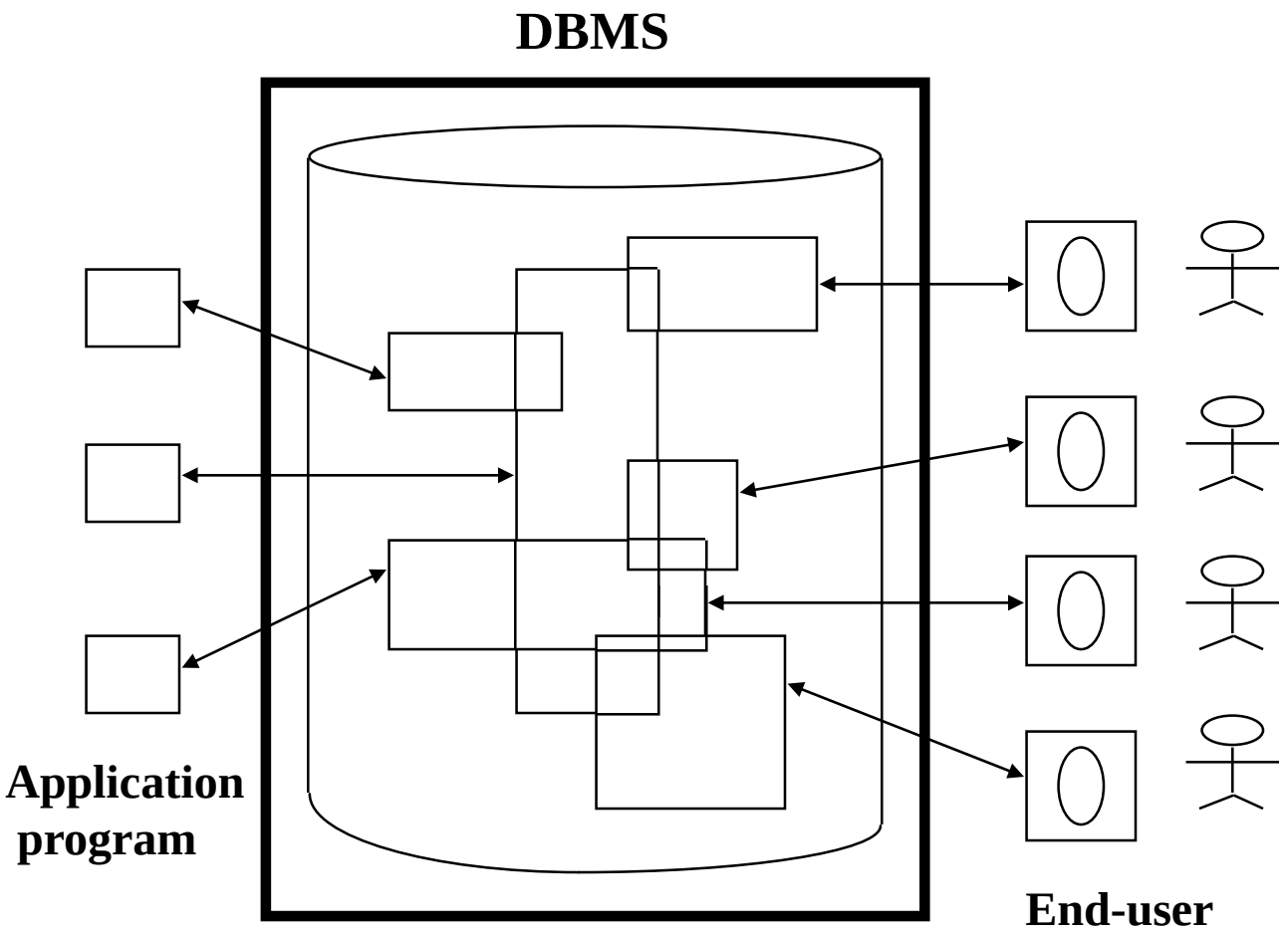
- Database Management System (DBMS)
  - Contains a large bodies of information
  - Collection of interrelated data (**database**)
  - **Set of programs** to access the data

- Goal of a DBMS:
  - provides a way to store and retrieve database information
    - ***convenient*** and
    - ***efficient***

Database System: Functions of DBMS

- Functions of DBMS: Management of Data (MOD)
  - **Defining structure** for storage data
  - Proving mechanisms for **manipulation of data**
  - **Ensure safety** of data (system crashes, unauthorized access, misused, …)
  - **Concurrent control** in multi-user environment
- Computer Scientists:
  - Developed a lot of **concepts** and **techniques** for MOD
  - **Concepts** and **techniques** form the focus of this **course**

Database System: Data Integrated and Shared



Database System: Major components

- Data: integrated and shared
- Hardware: disk, CPU, Main Memory, ...
- Software: DBMS
- Users:
    1. Application programmers
    2. End users
    3. ***Database administrator (DBA)***
        - Defining external schema
        - Defining conceptual schema
        - Defining internal schema
        - Liaison with users
        - Defining security and integrity checks
        - Defining backup and recovery procedures
        - Monitoring performance and changing requirements

An Example: Supplier-and-Parts Database

<e.g.> Supplier-and-Parts Database

S

S#	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

P

P#	PNAME	COLOR	WEIGHT	CITY
P1	Nut	Red	12	London
P2	Bolt	Green	17	Paris
P3	Screw	Blue	17	Rome
P4	Screw	Red	14	London
P5	Cam	Blue	12	Paris
P6	Cog	Red	19	London

SP

S#	P#	QTY
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S1	P6	100
S2	P1	300
S2	P2	400
S3	P2	200
S4	P2	200
S4	P4	300
S4	P5	400

- Query:
  - 列出住在London 的供應商名字?
  - Get the total number of suppliers.
  - Total QTY of SP?
  - Get supplier names for suppliers who supply part P2?
  - Get supplier names for suppliers who supply red color parts?
  - …

1.3 Why Database System?

Why Database System?

- Answer: **Easy to retrieve information!**
- Word, Excel vs. Access

<**e.g.**> Supplier-and-Parts Database

S

S#	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

SP

S#	P#	QTY
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S1	P6	100
S2	P1	300
S2	P2	400
S3	P2	200
S4	P2	200
S4	P4	300
S4	P5	400

P

P#	PNAME	COLOR	WEIGHT	CITY
P1	Nut	Red	12	London
P2	Bolt	Green	17	Paris
P3	Screw	Blue	17	Rome
P4	Screw	Red	14	London
P5	Cam	Blue	12	Paris
P6	Cog	Red	19	London

- Query:
  - 列出住在London 的供應商名字?
  - Get the total number of suppliers.
  - Total QTY of SP?
  - Get supplier names for suppliers who supply part P2?
  - Get supplier names for suppliers who supply red color parts?
  - …

Retrieval Operations

- **Easy to retrieve information!**

P

P#	PNAME	COLOR	WEIGHT	CITY
P1	Nut	Red	12	London
P2	Bolt	Green	17	Paris
P3	Screw	Blue	17	Rome
P4	Screw	Red	14	London
P5	Cam	Blue	12	Paris
P6	Cog	Red	19	London

- **Get color and city for "non-Paris" parts with weight greater than ten.**

```
SELECT  P.COLOR,  P.CITY
FROM    P
WHERE  P.CITY < >  'Paris'
AND      P.WEIGHT > 10;
```

COLOR	CITY
Red	London
Blue	Rome
Red	London
Red	London

- **DISTINCT**

```
SELECT  DISTINCT  P.COLOR,  P.CITY
FROM    P
WHERE  P.CITY < >  'Paris'
AND      P.WEIGHT  >  10;
```

COLOR	CITY
Red	London
Blue	Rome

Retrieval Operations (cont.)

- For all parts, get the part number and the weight of that part in grams.

```
SELECT  P.P#,  P.WEIGHT * 454  AS  GMWT
FROM    P ;
```

P

P#	PNAME	COLOR	WEIGHT	CITY
P1	Nut	Red	12	London
P2	Bolt	Green	17	Paris
P3	Screw	Blue	17	Rome
P4	Screw	Red	14	London
P5	Cam	Blue	12	Paris
P6	Cog	Red	19	London

- Get the maximum and minimum quantity for part P2.

```
SELECT MAX (SP.QTY) AS MAXQ,
       MIN (SP.QTY) AS MINQ
FROM   SP
WHERE  SP. P# = 'P2';
```

SP

S#	P#	QTY
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S1	P6	100
S2	P1	300
S2	P2	400
S3	P2	200
S4	P2	200
S4	P4	300
S4	P5	400

- For each part supplied, get the part number and the total shipment quantity.

```
SELECT SP.P#,  SUM (SP.QTY) AS TOTQTY
FROM   SP
GROUP  BY  SP.P#;
```

Retrieval Operations (cont.)

- Get part numbers for all parts supplied by more than one supplier.

```
SELECT  SP.P#
FROM    SP
GROUP   BY  SP.P#
HAVING  COUNT  ( SP. S# )  >  1;
```

- Get supplier names for suppliers who supply part P2.

```
SELECT   DISTINCT  S.SNAME
FROM     S
WHERE    S. S#  IN
         ( SELECT  SP. S#
           FROM    SP
           WHERE   SP.P#  =  'P2');
```

<e.g> Supplier-and-Parts Database

S

S#	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

SP

S#	P#	QTY
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S1	P6	100
S2	P1	300
S2	P2	400
S3	P2	200
S4	P2	200
S4	P4	300
S4	P5	400

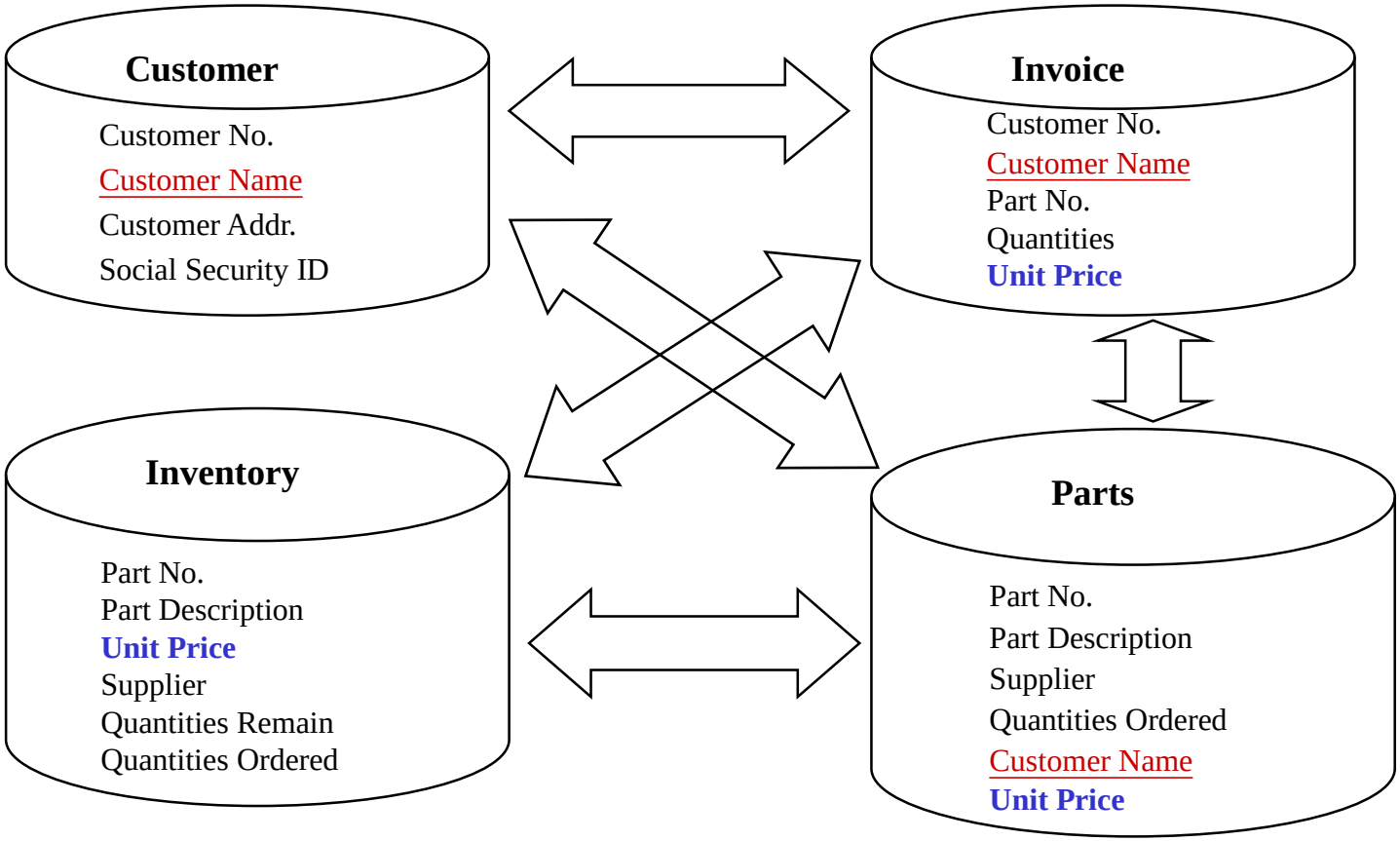
P

P#	PNAME	COLOR	WEIGHT	CITY
P1	Nut	Red	12	London
P2	Bolt	Green	17	Paris
P3	Screw	Blue	17	Rome
P4	Screw	Red	14	London
P5	Cam	Blue	12	Paris
P6	Cog	Red	19	London

Why Database ?

- **Easy to retrieve information!**
- Redundancy can be reduced
- Inconsistency can be avoided
- The data can be shared
- Standards can be enforced
- Security restrictions can be applied
- Integrity can be maintained
- Provision of *data independence* ⇦ objective !
- Database Growth Fast!

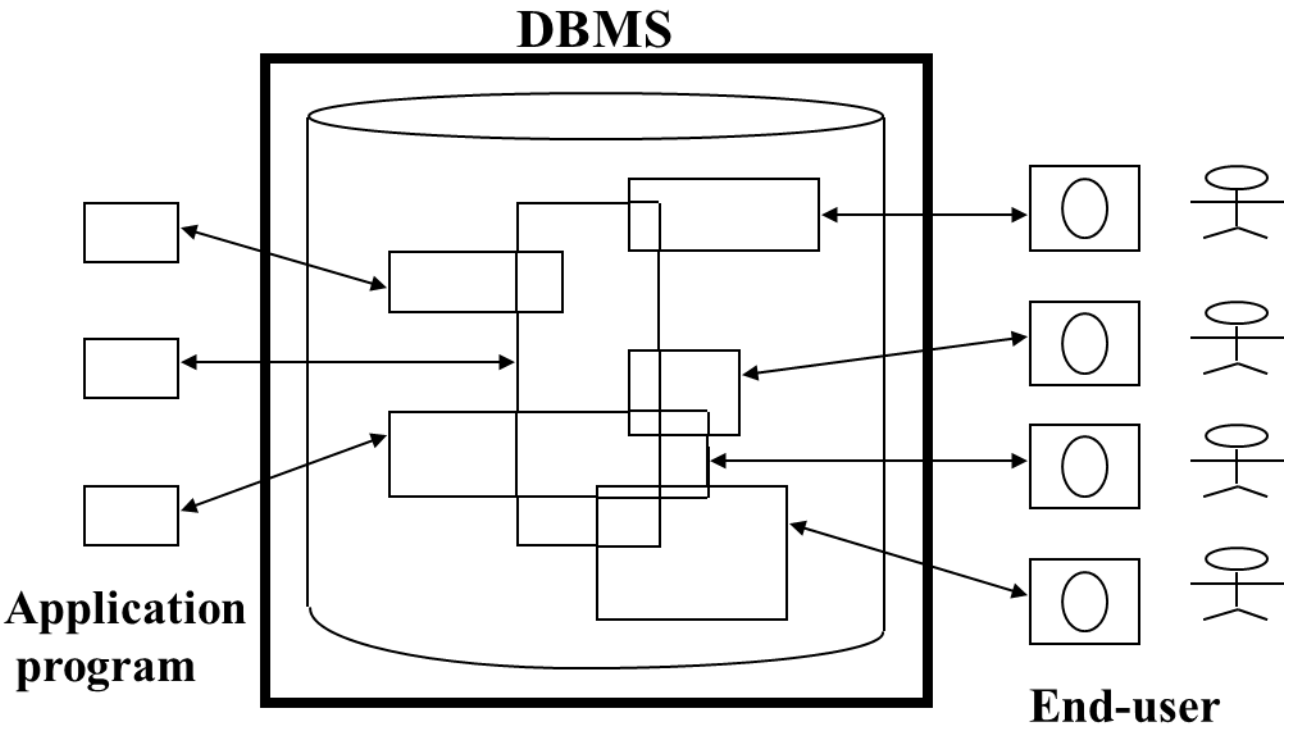
Why Database: Redundancy can be reduced



Why Database: Inconsistency can be avoided

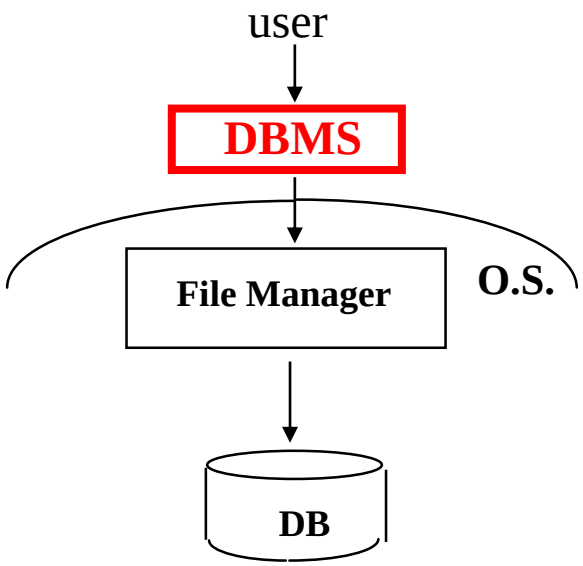
Why Database: The data can be shared

Database System: Data Integrated and Shared



Why Database: Standards can be enforced

Why Database: Security restrictions can be applied



S

S#	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

- **<e.g.1> [GRANT]**

```
GRANT SELECT ON TABLE S TO CHARLEY;
GRANT SELECT, UPDATE ( STATUS, CITY ) ON TABLE S TO JUDY, JACK, JOHN;
GRANT ALL ON TABLE S, P, SP TO FRED, MARY;
GRANT SELECT ON TABLE P TO PUBLIC;
GRANT INDEX ON TABLE S TO PHIL;
```

Why Database: Integrity can be maintained

- Consider Supplier-and-Parts Database,

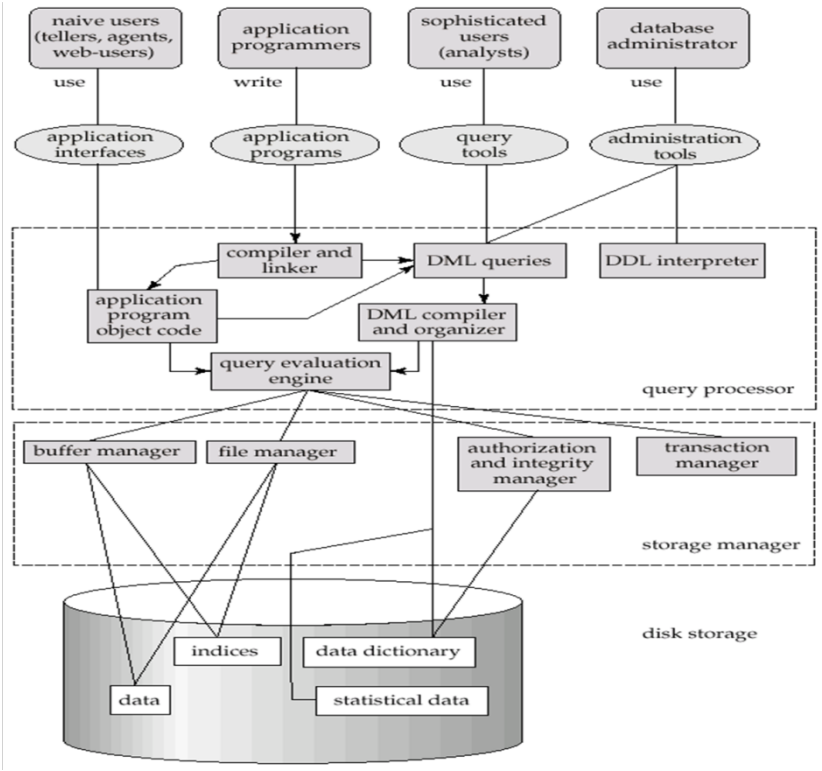
  Assume the STATUS should always be positive value.

S

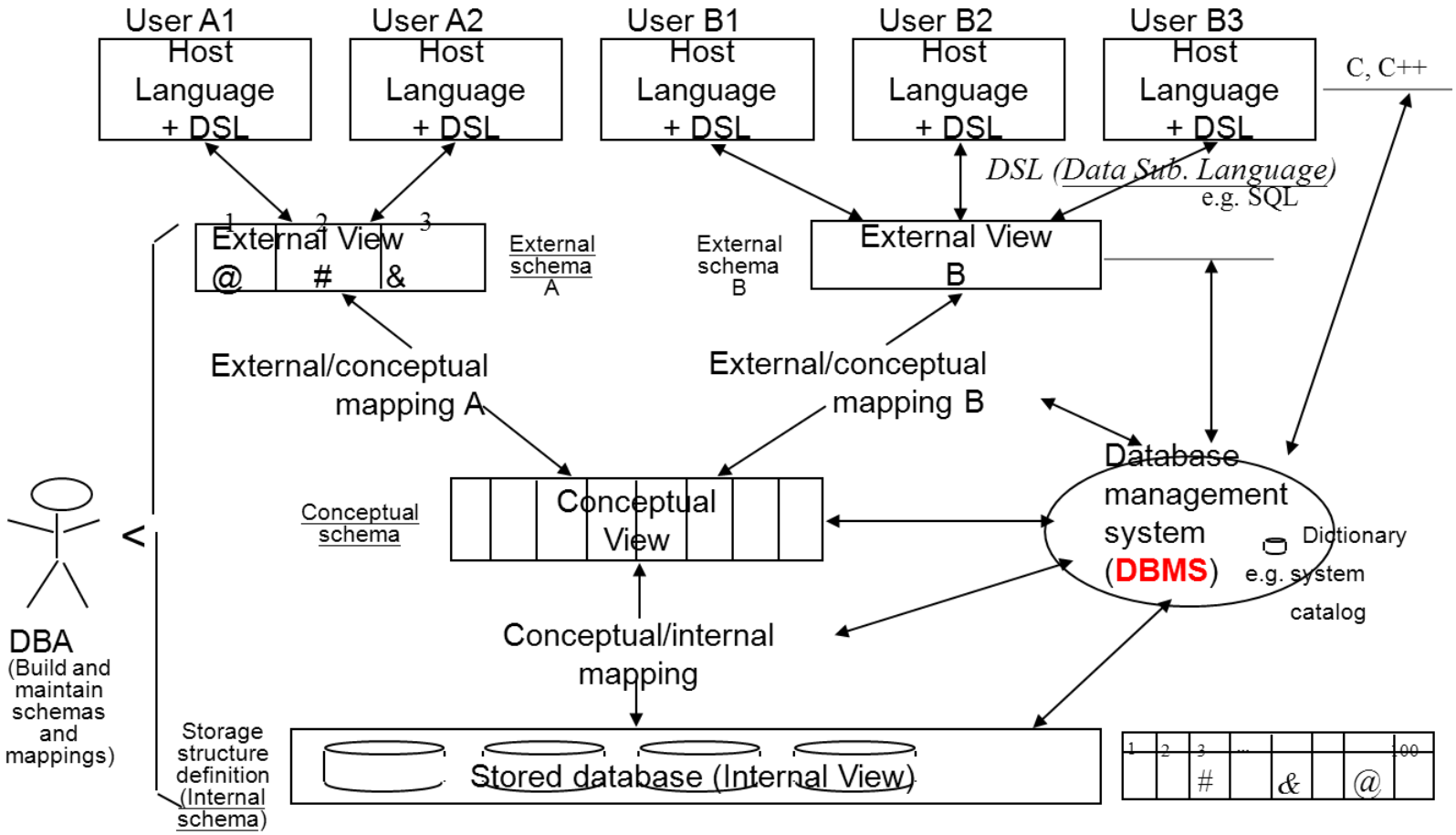
S#	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

- CREATE INTEGRITY RULE

```
CREATE INTEGRITY RULE R1
   ON INSERT   S.STATUS,
         UPDATE S.STATUS;
   CHECK FORALL S ( S.STATUS > 0 )
   ELSE REJECT;
```



Why Database: Provision of data independence



Data Independence

- Application Program
  ➔ Data Structure
- Immunity of application to change in storage structure and access strategy.

Data Dependence *vs.* Data Independence

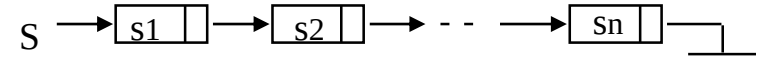
- Data Dependent

e.g.
```
SELECT CITY
FROM   S
WHERE  ITEM = 'X';
```

S

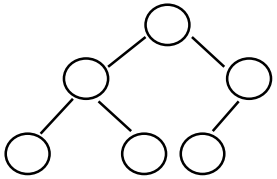
S#	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

- **Linked list:** TOP



Top if item = TOP . item then .....

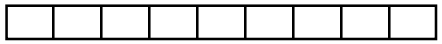
- **Tree:**



.

if item < root.data then root := root .left .....

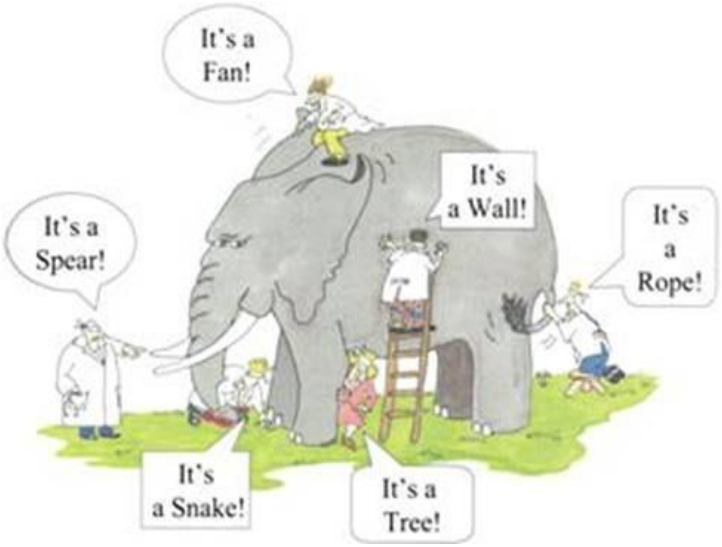
- **Array:**



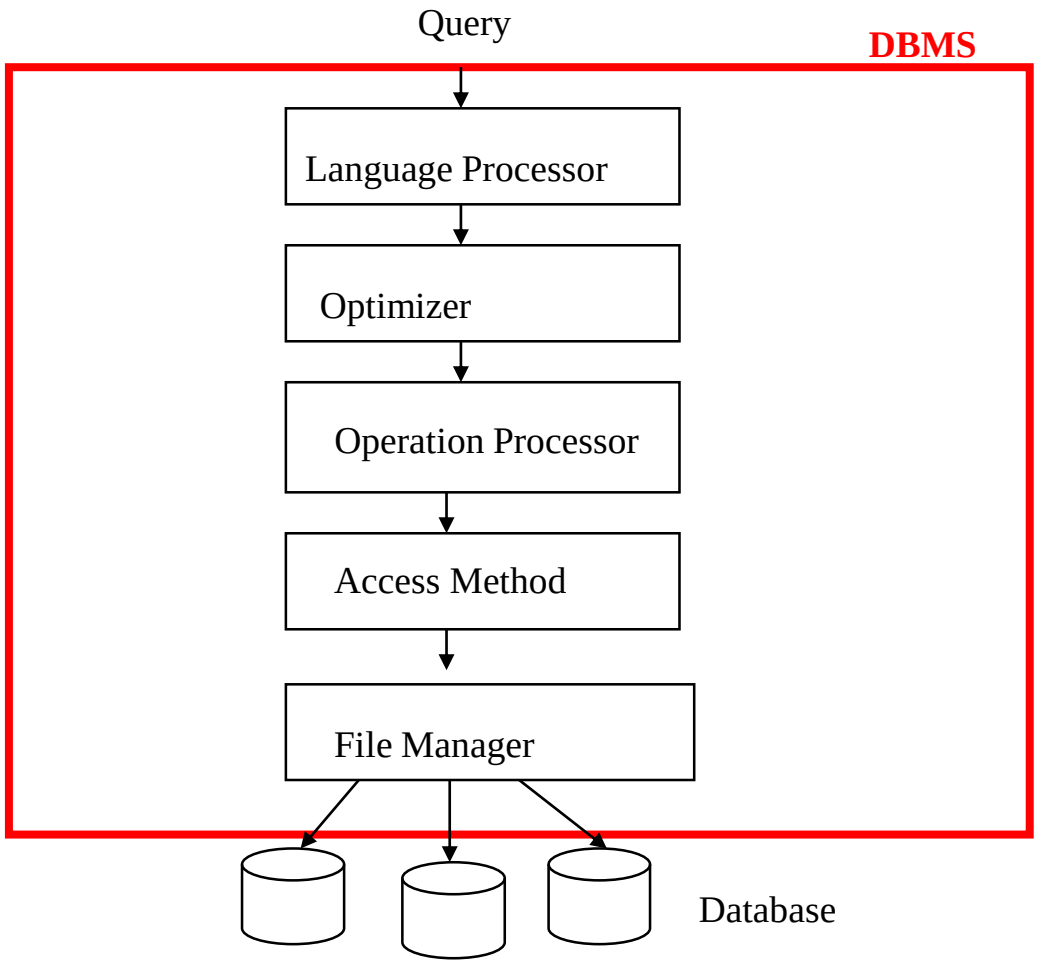
if A[I] = item then .....

- ***Storage structure changed ➔ program changed***

1.4 An Architecture for a Database System

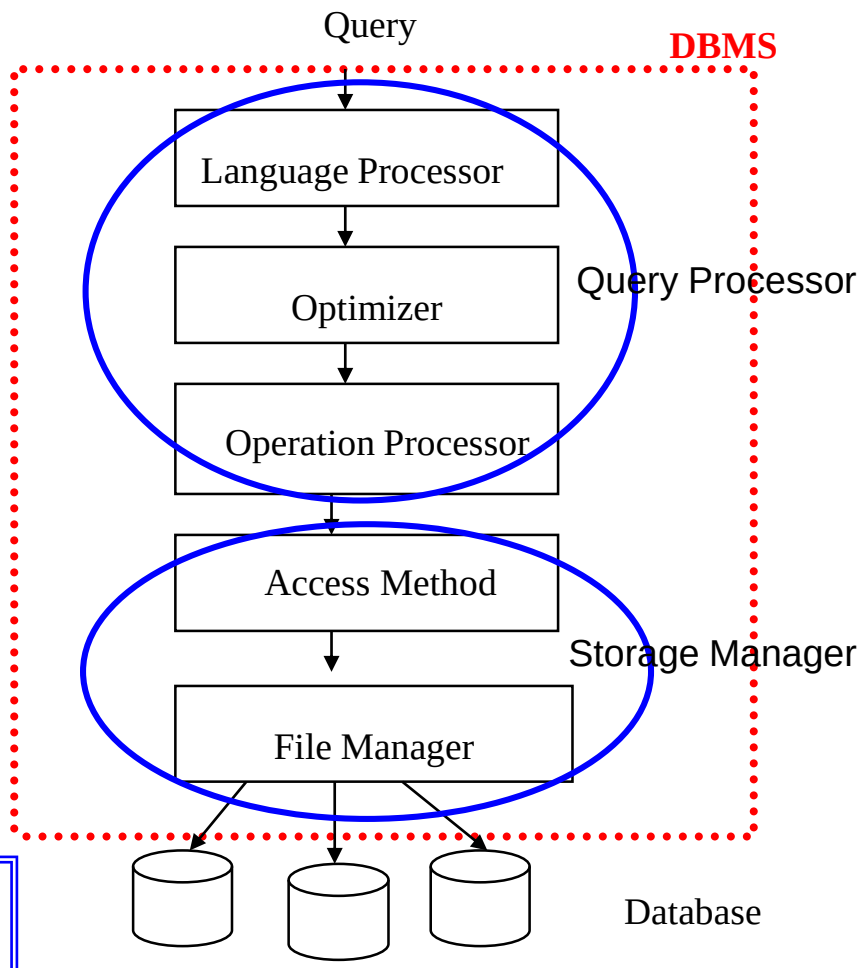


Architecture for a Database System: view 1



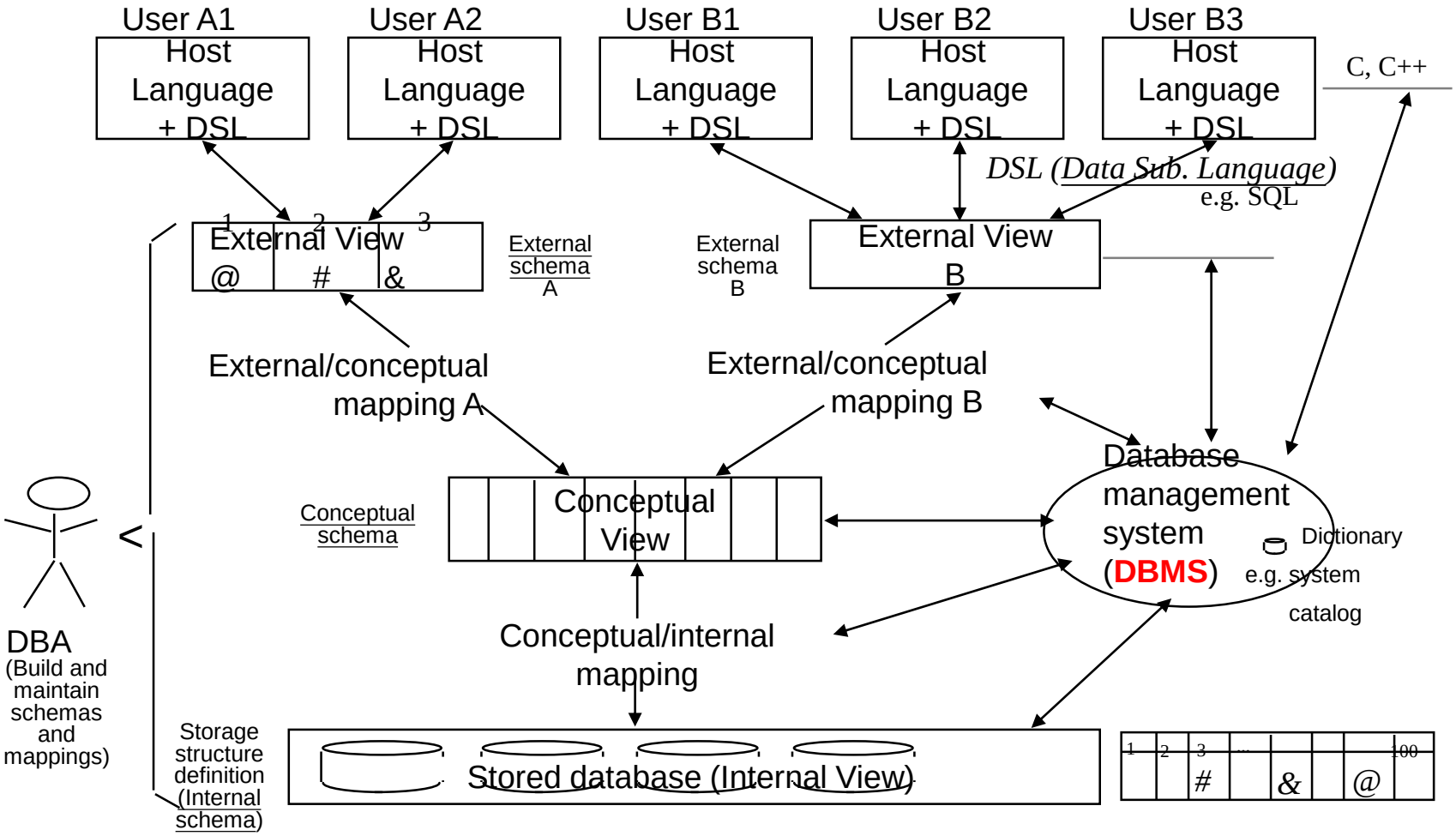
Querying and Data Storage

- Components of Database System
  - **Query Processor**
    - Helps to simplify to access data
    - High-level view
    - Users are not be burdened unnecessarily with the physical details
  - **Storage Manager**
    - Require a large amount of space
    - Can not store in main memory
    - Disk speed is slower
    - Minimize the need to move data between disk and main memory

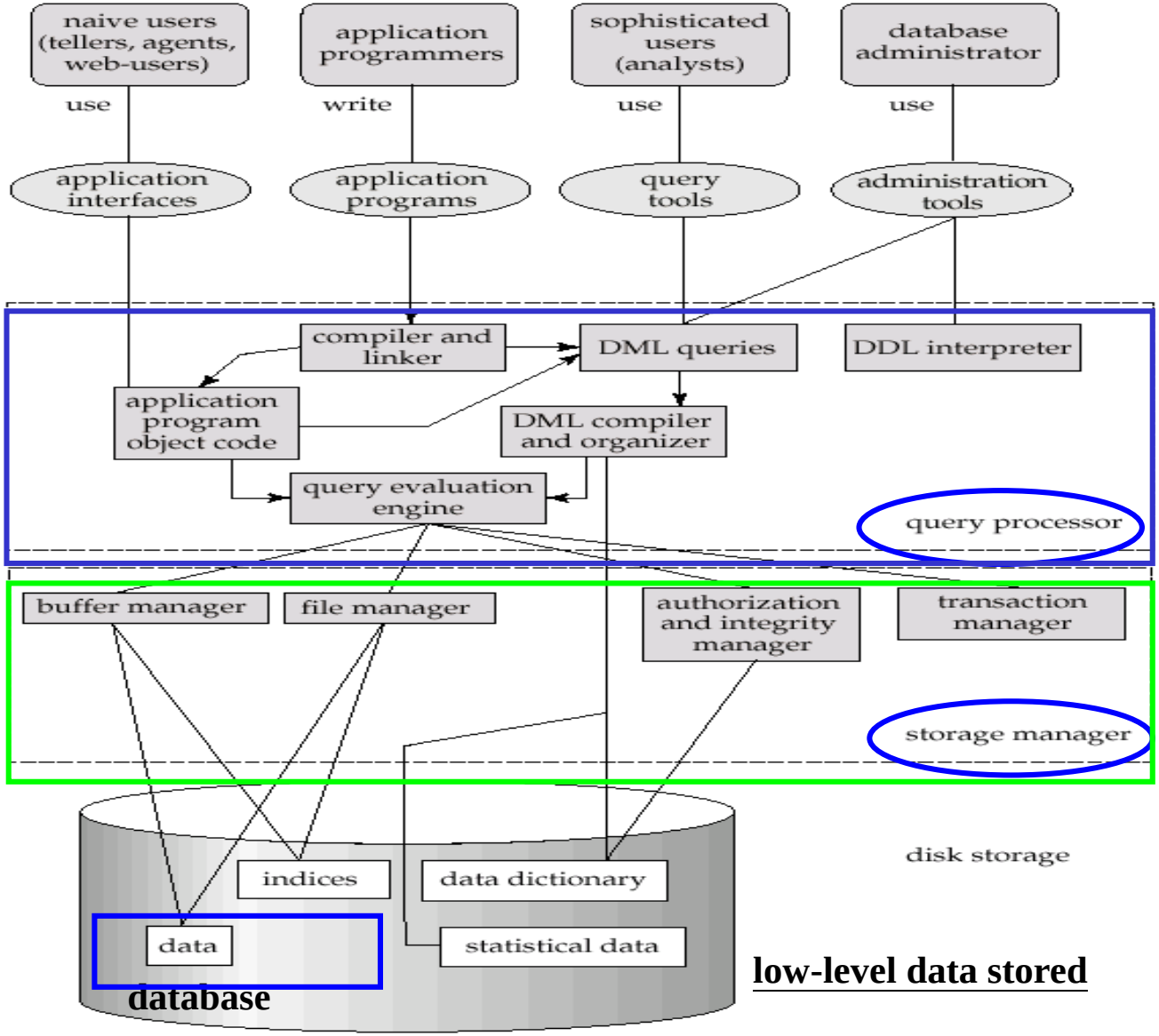


Goal of a DBMS: provides a way to **store** and **retrieve** data that is both ***convenient*** and ***efficient***.

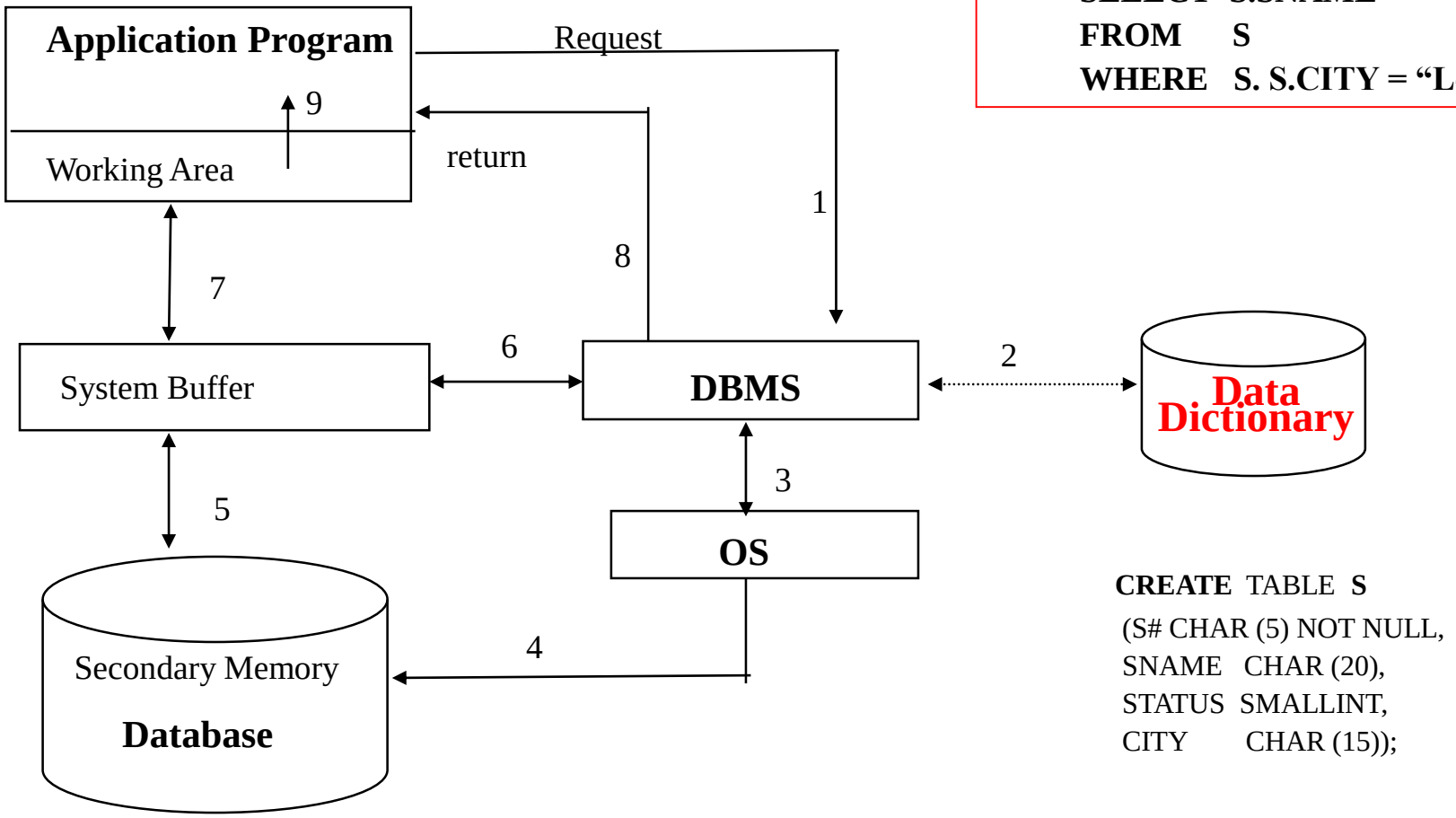
Architecture for a Database System: view 2



Overall System Structure



Data Dictionary in DBMS




```
SELECT  S.SNAME
FROM    S
WHERE   S. S.CITY = "London";
```

```
CREATE TABLE S
(S# CHAR (5) NOT NULL,
SNAME   CHAR (20),
STATUS  SMALLINT,
CITY    CHAR (15));
```

Architecture for a Database System: view 3

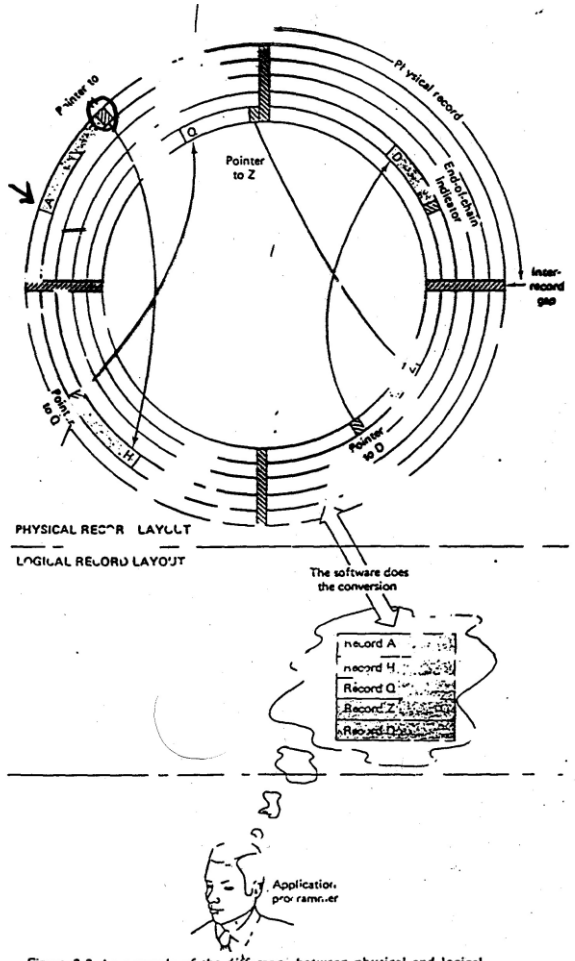


Figure 2.2 An example of the difference between physical and logical data organization.

Architecture for a Database System: view 4

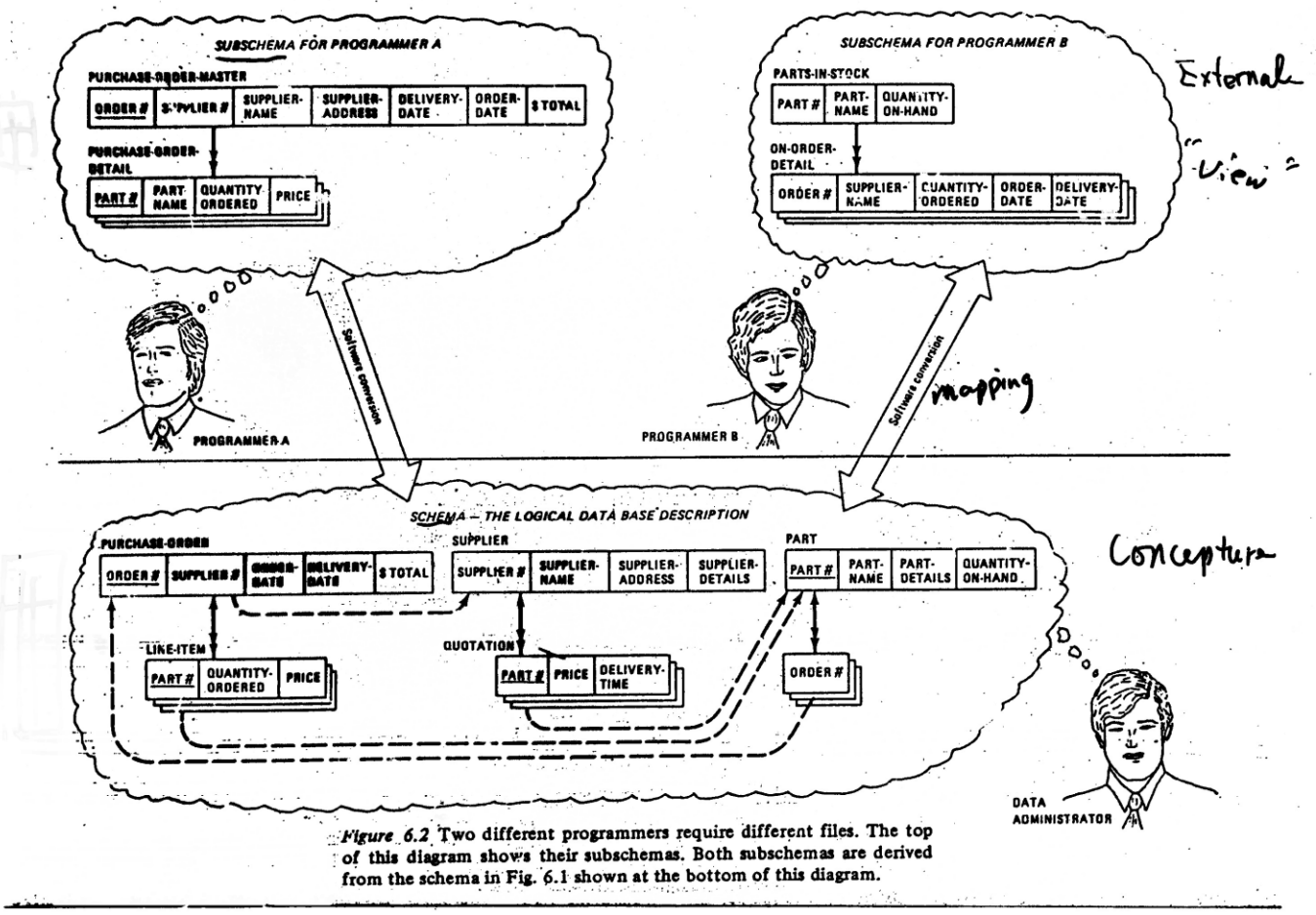


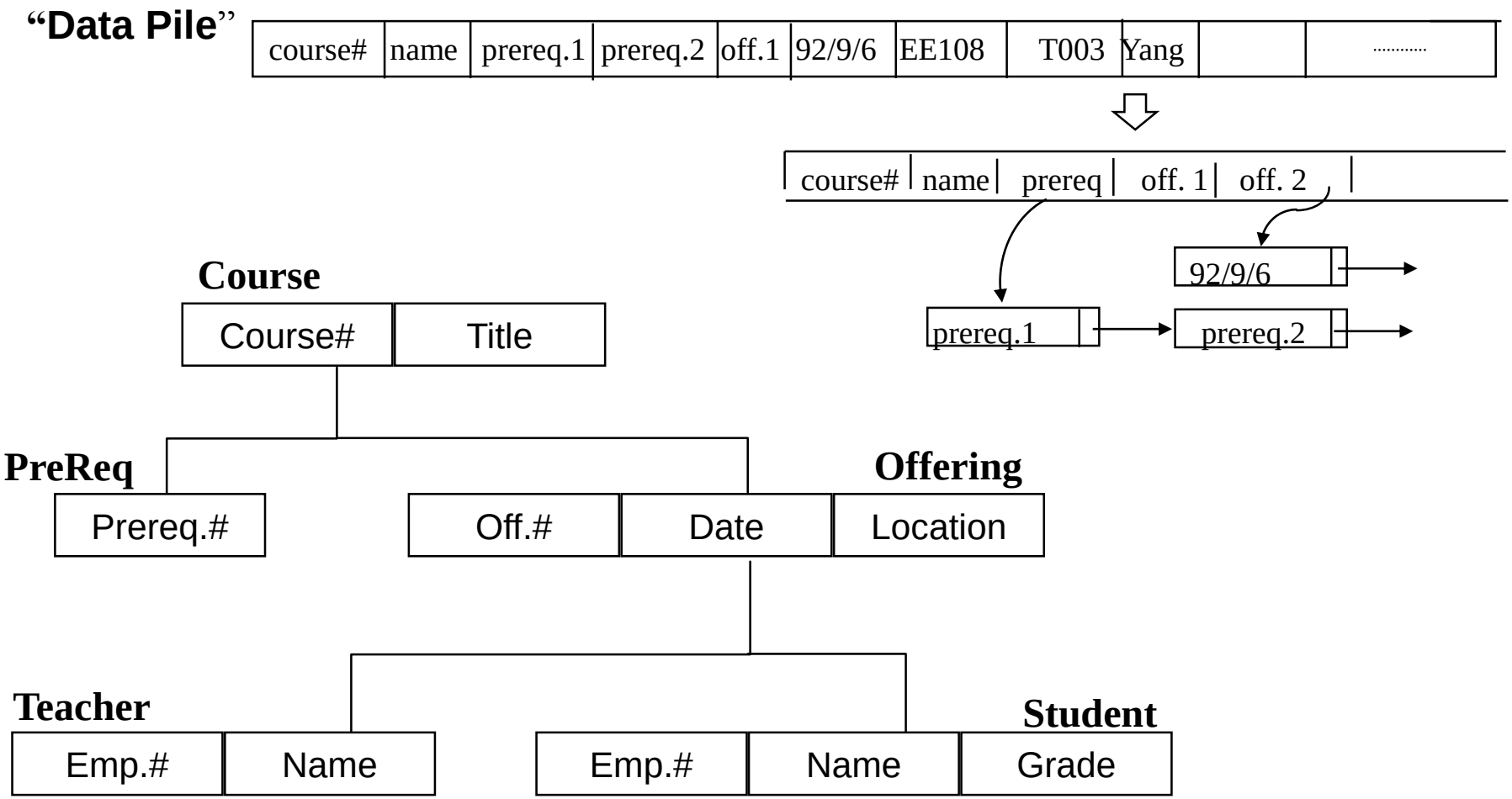
Figure 6.2 Two different programmers require different files. The top of this diagram shows their subschemas. Both subschemas are derived from the schema in Fig. 6.1 shown at the bottom of this diagram.

1.5 Data Models

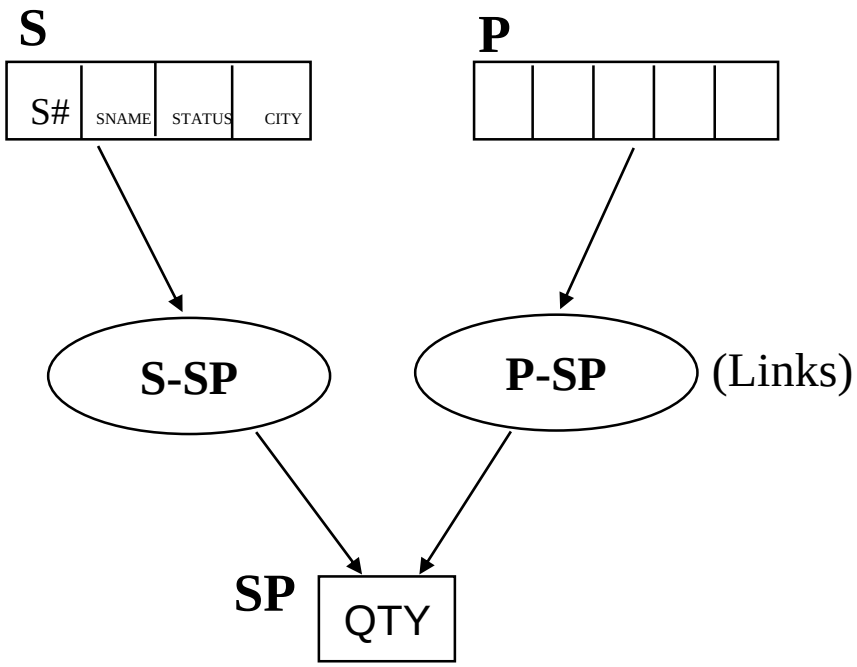
Data Models:

- Hierarchical Data Model
- Network Data Model
- Relational Data Model
- Object-Oriented Data Model
- …

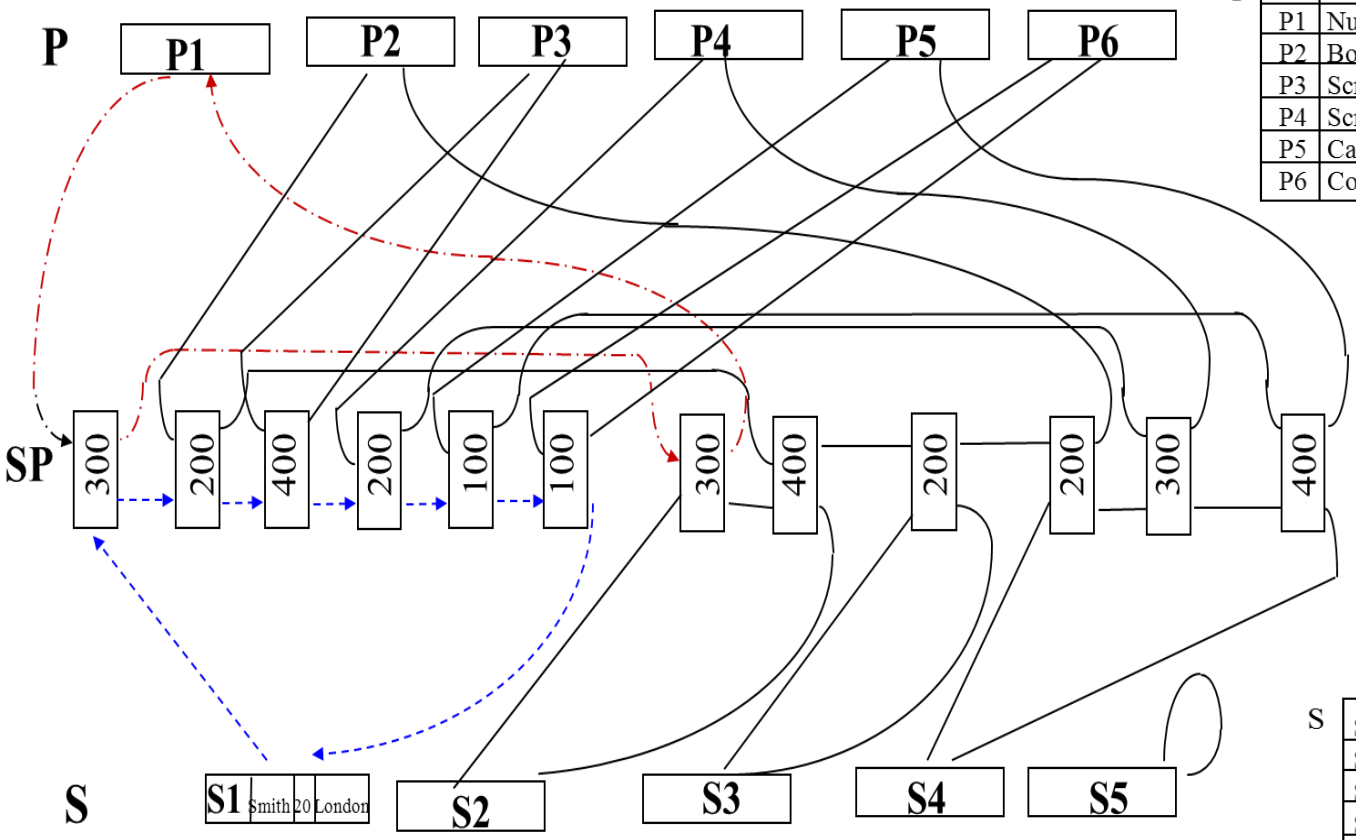
Hierarchical Data Model



Network Data Model



The Network Model: Sample Values



P

P#	PNAME	COLOR	WEIGHT	CITY
P1	Nut	Red	12	London
P2	Bolt	Green	17	Paris
P3	Screw	Blue	17	Rome
P4	Screw	Red	14	London
P5	Cam	Blue	12	Paris
P6	Cog	Red	19	London

SP

S#	P#	QTY
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S1	P6	100
S2	P1	300
S2	P2	400
S3	P2	200
S4	P2	200
S4	P4	300
S4	P5	400

S

S#	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

Relational Data Model: [Codd70]

- System R
- DB2
- INGRES
- Oracle
- Informix
- ACCESS
- mySQL
- …

<e.g.> Supplier-and-Parts Database

S

S#	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

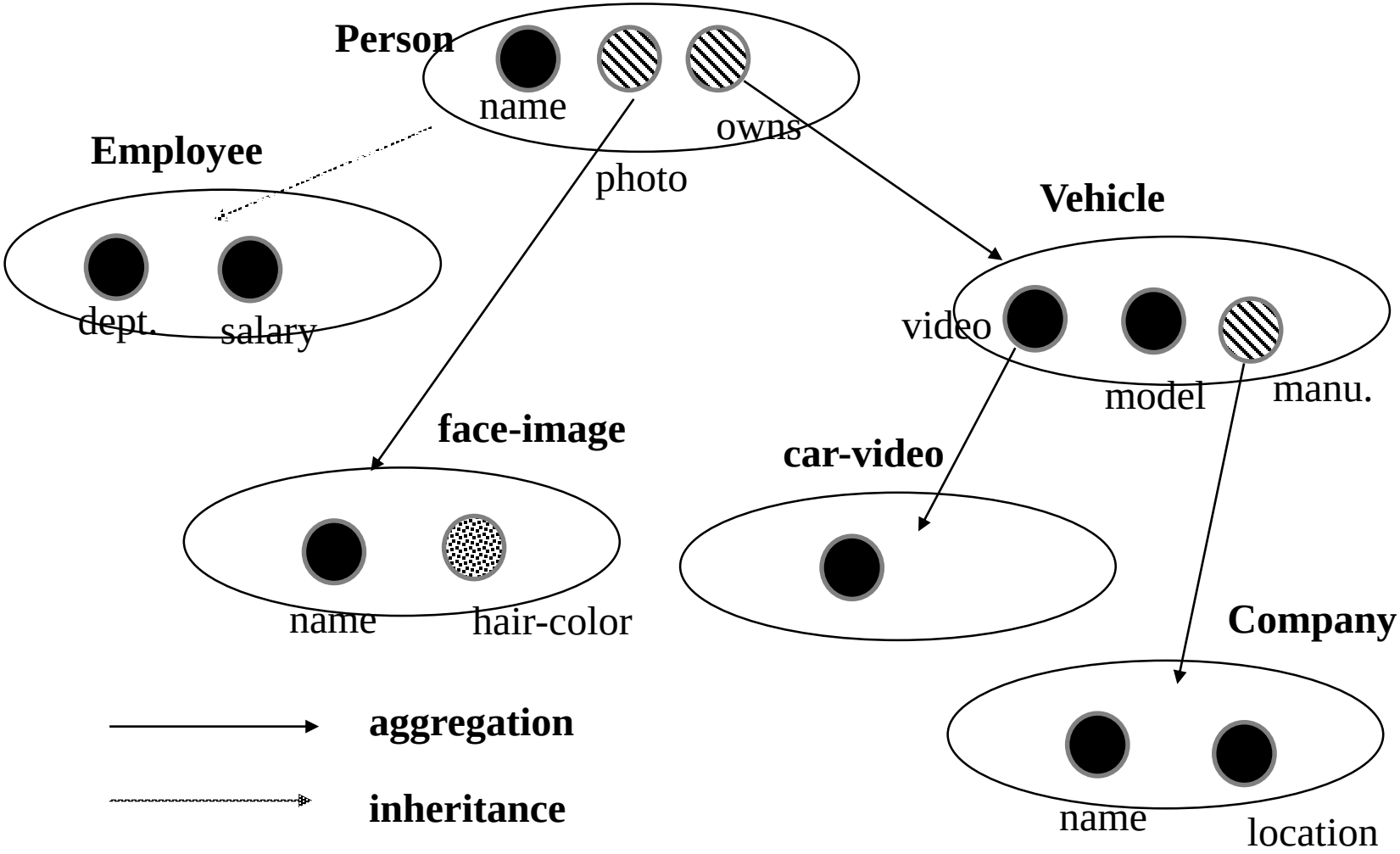
P

P#	PNAME	COLOR	WEIGHT	CITY
P1	Nut	Red	12	London
P2	Bolt	Green	17	Paris
P3	Screw	Blue	17	Rome
P4	Screw	Red	14	London
P5	Cam	Blue	12	Paris
P6	Cog	Red	19	London

SP

S#	P#	QTY
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S1	P6	100
S2	P1	300
S2	P2	400
S3	P2	200
S4	P2	200
S4	P4	300
S4	P5	400

Object-Oriented Data Model



Database Technology Trends

	1960s to Mid-1970s	1970s to Mid-1980s	Late 1980s	Future
Data Model	Network Hierarchical	Relational	Semantic Object-oriented Logic	Merging data models, knowledge representation, and programming languages
Database Hardware	Mainframes	Mainframes Minis PCs	Faster PCs Workstations Database machines	Parallel processing Optical memories
User Interface	None Forms	Query languages - SQL, QUEL	Graphics Menus Query-by-forms	Natural language Speech input
Program Interface	Procedural	Embedded query language	4GL Logic programming	Integrated database and programming language
Presentation and display processing	Reports Processing data	Report generators Information and transaction processing	Business graphics Image output Knowledge processing	Generalized display managers Distributed knowledge processing

1.6 Establish/Design a Database System

PART II: 資料庫設計 (Database Design)

- 資料庫問題分析與架構規劃:
  - 若有一大量資料想利用DBMS建資料庫來管理。第一步要分析問題，找到使用者需求
  - 實體-關係模型(Entity-Relationship Model,簡稱E-R Model)是一套資料庫的設計工具。我們可以利用E-R Model分析資料庫問題。它可以把真實世界中複雜的問題中的事物和關係轉化為資料庫中的資料架構
  - 由於利用實體-關係模型設計資料庫時, 並不會牽涉到資料庫的操作、儲存方式等複雜的電腦運作。所以,我們會把心力放在需求分析去規劃想要的資料庫，並以實體-關係圖(E-R Diagram)來呈現
- 資料庫的表格正規化:
  - 實體-關係圖很容易轉化為表格(Tables)，而資料庫就是由許多表格(tables)組成的
  - 這些表格要正規化(Normalization)才能避免將來操作時的異常現象發生
- 設計介面增刪查改資料庫:
  - 如何方便、又有效率的管理存取資料庫是使用者最關心的二個要素
  - 良好的介面設計，可以讓使用者方便的查詢、方便的新增、方便的刪除、方便的修改的處理資料庫

Database Design

- **Database Design -** The process of designing the general structure of the database:
  - Logical Design
  - Physical Design
- **Logical Design –** Deciding on the database schema.
  - To find a "good" collection of relation schemas.
  - Business decision – What attributes should we record in the database?
  - Computer Science decision – What relation schemas should we have and how should the attributes be distributed among the various relation schemas?
- **Physical Design –** Deciding on the physical layout of the database

Design Process

- **Phase I**
  - Specification of user requirement (with domain experts)
- **Phase II**
  - Conceptual design (unit 6)
  - Choose a data model
  - Design tables
  - Normalization (unit 7)
- **Phase III**
  - Specification of functional requirements
- **Phase IV**
  - User interface design (unit 8)
  - Implementation

Contents of PART II: 資料庫設計

- **Unit 6 Database Design and the E-R Model**
- **Unit 7 Normalization (表格正規化)**
- **Unit 8 User Interfaces (使用者介面)**
- **Unit 9 實作範例一:**
- **Unit 10 實作範例二:**

- **References:**

1. C. J. Date, An Introduction to Database Systems, 8th edition, 2004.
2. A. Silberschatz, etc., Database System Concepts, 5th edition, McGraw Hill, 2006
3. J. D. Ullman, Principles of Database and Knowledge-Base, vol. I, 1988.
4. Cited papers

How to Establish a Database System?

- STEP 1: Database Design
  - Logical database vs. physical database
  - Collect data of applications
  - Analyze data to eliminate redundancy (using normalization theory and E-R Model…)
  - Describe data in the specific Data Model the DBMS use.
  - Describe each schema in DDL
- STEP 2: Implementation
  - schema
  - data
- SETP3:Evaluation and Correction (by DBA)
  - tuning
  - statistical analysis

Components of a Database System

- DDL (Data Definition Language)
- DML (Data Manipulation Language)
- Data Dictionary
- Utility Routines

Components of a Database System: DDL

- **DDL (Data Definition Language)**

```
CREATE TABLE S
   (S#          CHAR(5)     NOT NULL,
    SNAME       CHAR(20)    NOT NULL,
    STATUS      SMALLINT    NOT NULL,
    CITY        CHAR(15)    NOT NULL,
    PRIMARY KEY (S#));

ALTER TABLE S ADD DISCOUNT SMALLINT;

DROP TABLE S

CREATE INDEX XSC ON S (CITY);
CREATE UNIQUE INDEX X ON S (S#);

DROP INDEX XSC;
```

S

S#	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

Components of a Database System: DML

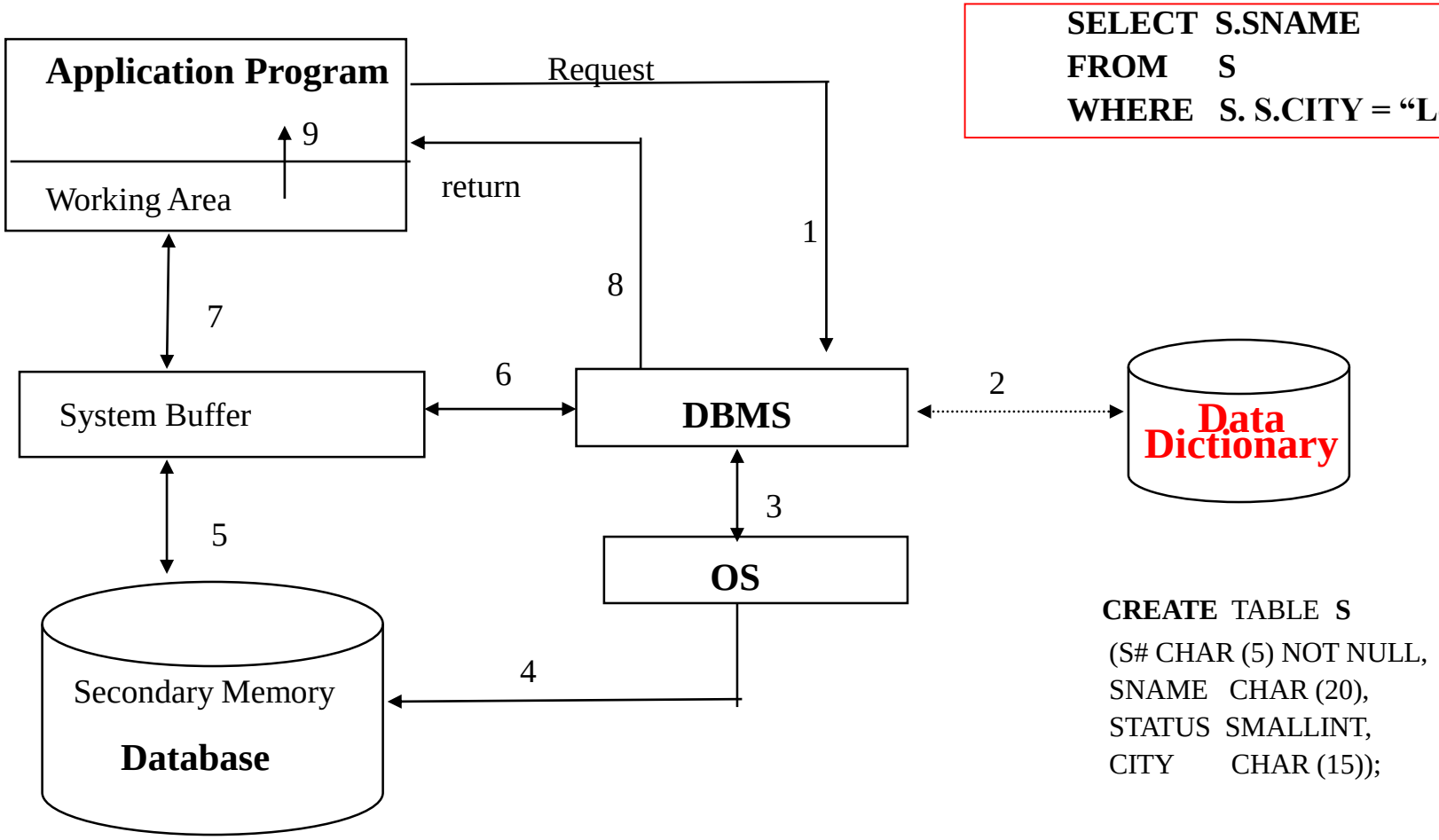
- **DML (Data Manipulation Language)**

```
SELECT S#, STATUS
FROM S
WHERE CITY='PARIS'

UPDATE S
SET STATUS= 2*STATUS
WHERE CITY='LONDON';

DELETE
FROM S
WHERE S#= 'S5'

INSERT
INTO S (S#, SNAME, STATUS, CITY)
VALUES('S6','TSENG',100,'HSINCHU')
```

S

S#	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

Components of a Database System: Data Dictionary



Components of a Database System: Utility Routines

- Loading Routines
- Reorganization Routines
- Journalizing routines (log)
- Database Dump Routines
- Recovery Routines
- Statistical Analysis Routines
- …

1.7 Extending Database Technology

Extending Database Technology

- Expert Database Systems
  Knowledge-base Management Systems
  AI + DB

- Image Database Systems
  Intelligent Pictorial Databases
  Image + DB

1-64

- Object-Oriented Database Systems
  OO Programming + DB

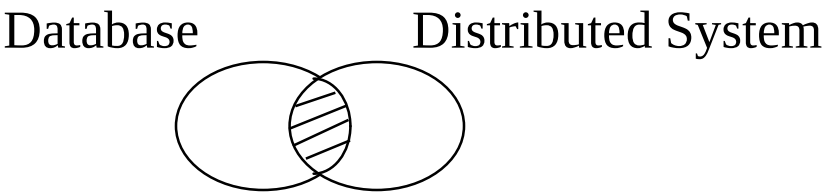
- Multimedia Database
  Text + Voice + Image + …..+ DB

- Multidatabases
  Integrate heterogeneous /homogeneous database systems

Extending Database Technology (cont.)

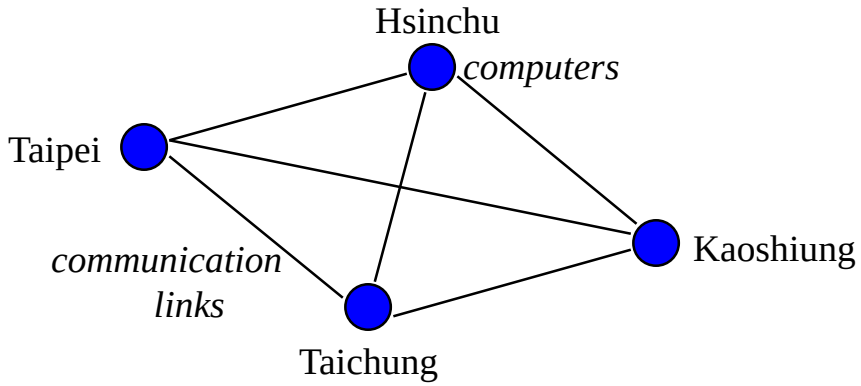
- Real-time Database Systems
  Real-time Tech. + DB
- Video Database Systems
  MPEG + DB
- Digital Library
  Library + DB
- Bioinformatics Database Systems
  Biological + DB
- **…**

Distributed Databases



- Distributed database is a database that is not stored in its entirety at a single physical location, but rather is spread across a network of computer.

< e.g.>



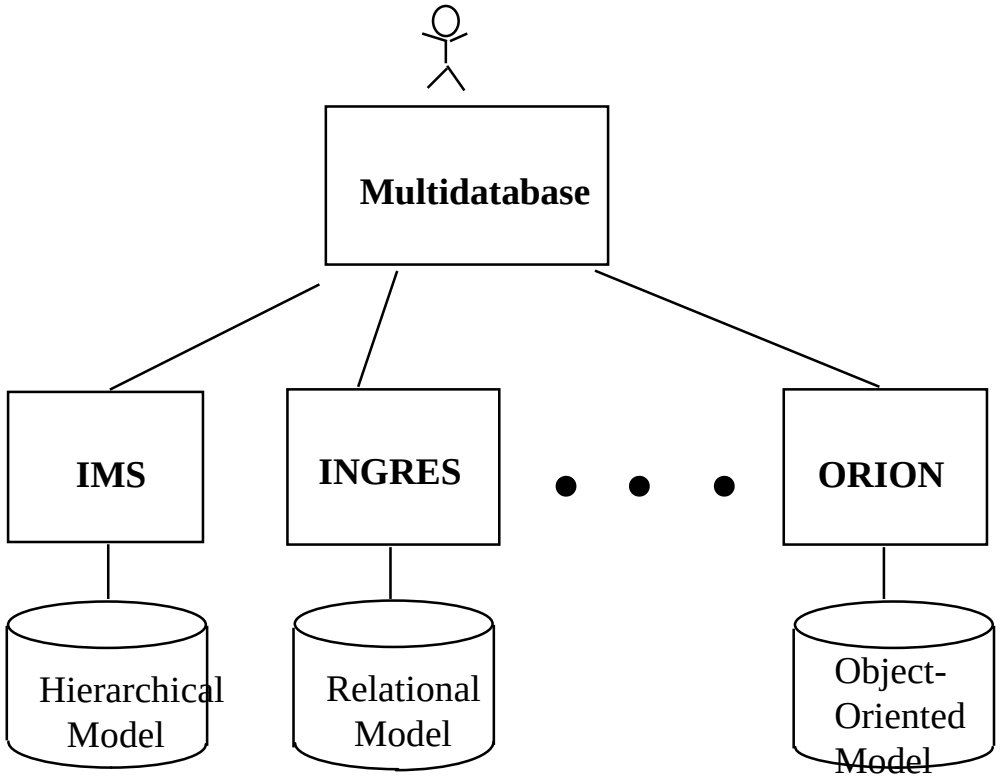
Distributed Databases (cont.)

- Advantages:
  - efficiency of local processing
  - data sharing
- Disadvantages:
  - communication overhead
  - implementation difficulties
- Reference:

  S. Ceri and G. Pelagatti

  "Distributed Databases: principles and systems"

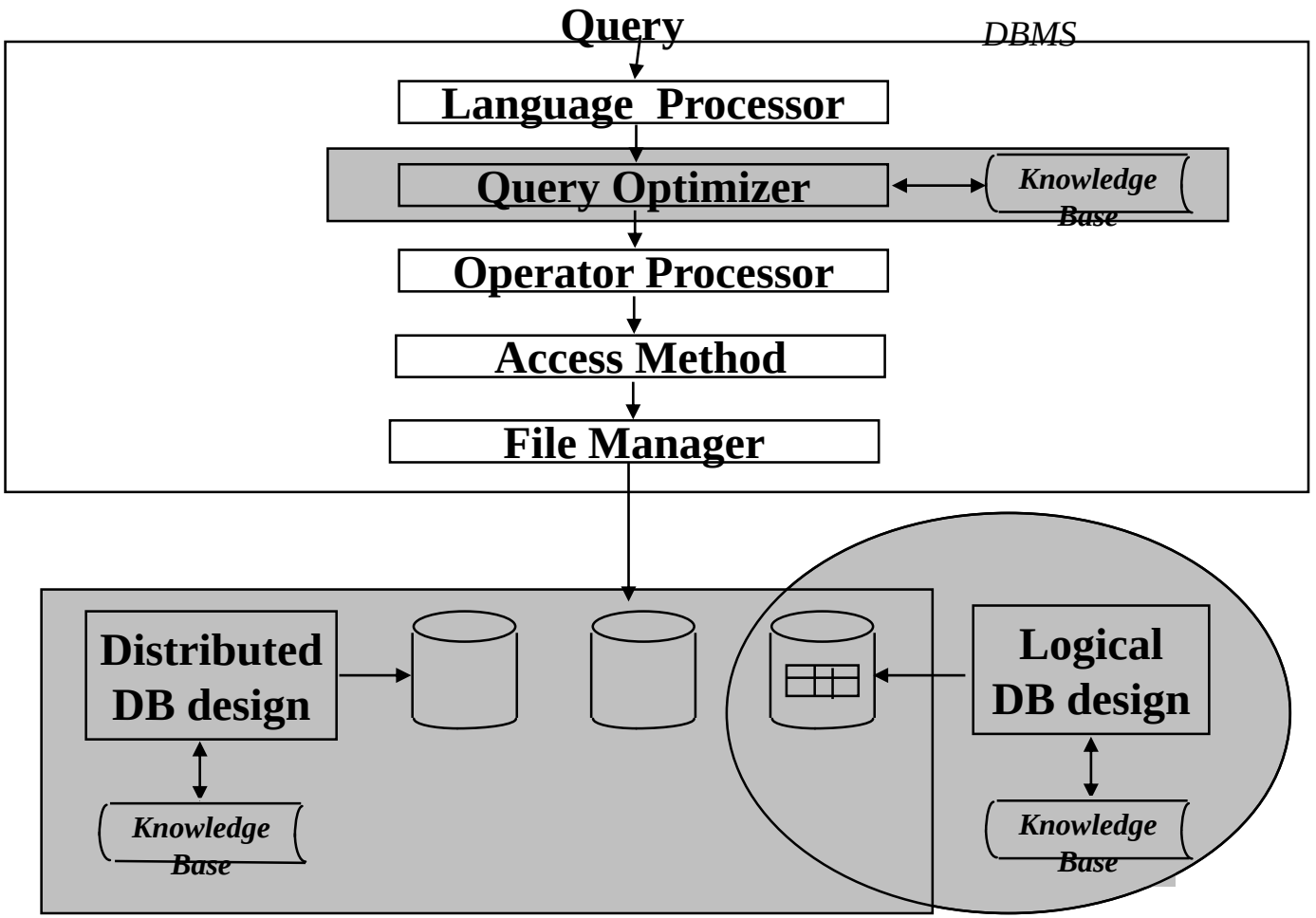
Multi-Database/Heterogeneous Database



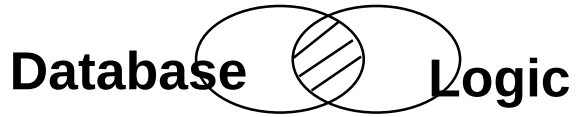
- **semantic inconsistency**
- **data incompleteness**
- **global schema**

DB + AI


Database
AI


Query
DBMS
Language Processor
Query Optimizer
Knowledge Base
Operator Processor
Access Method
File Manager
Distributed DB design
Knowledge Base
Logical DB design
Knowledge Base

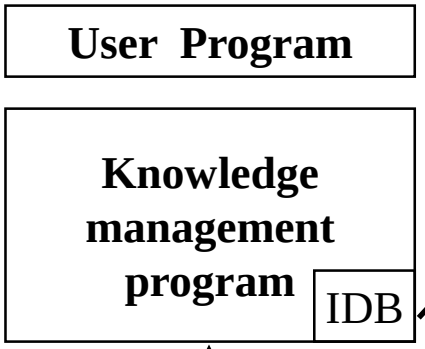
KBMS



- A Combined Model :

  Logic Programming + Relational DB

- Three layers :



Query :

? :- ancestor (taro, Y)

? :- grandfather (?, c)

IDB:

ancestor(X,Y):- parent(X,Y)

ancestor(X,Y) :- parent(X,Z), ancestor(Z,Y)

parent(X,Y):-edb(father(X,Y))

parent(X,Y):-edb(mother(X,Y))

grandfather(X,Z):- father(X,Y) ^ father(Y,Z)

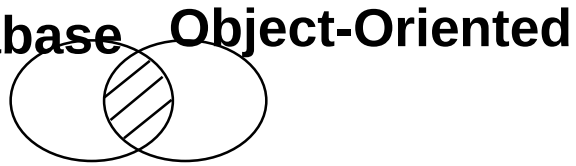
EDB:

father

father	son
A	B
X	Y
.	.
B	C

mother

OODB



- A typical Document : MEMO [Woelk86, SIGMOD]

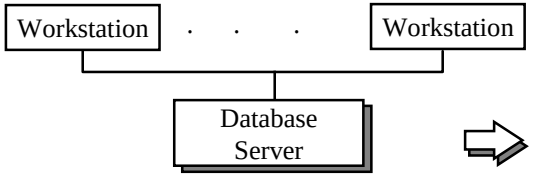
MCC

To: W. Kim
From: D. Woolk
Date: September 18, 1992
Subject: Workstations

speaker voice message associated

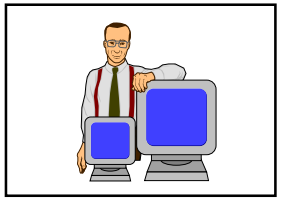
In the computer center of National Chiao-Tung University, there are a lot of workstations. There are HP RS serials, SUNs, Apollo, and so on. The students in NCTU learn to use workstation since they are fresh-men. The configuration of the workstations follows:

text



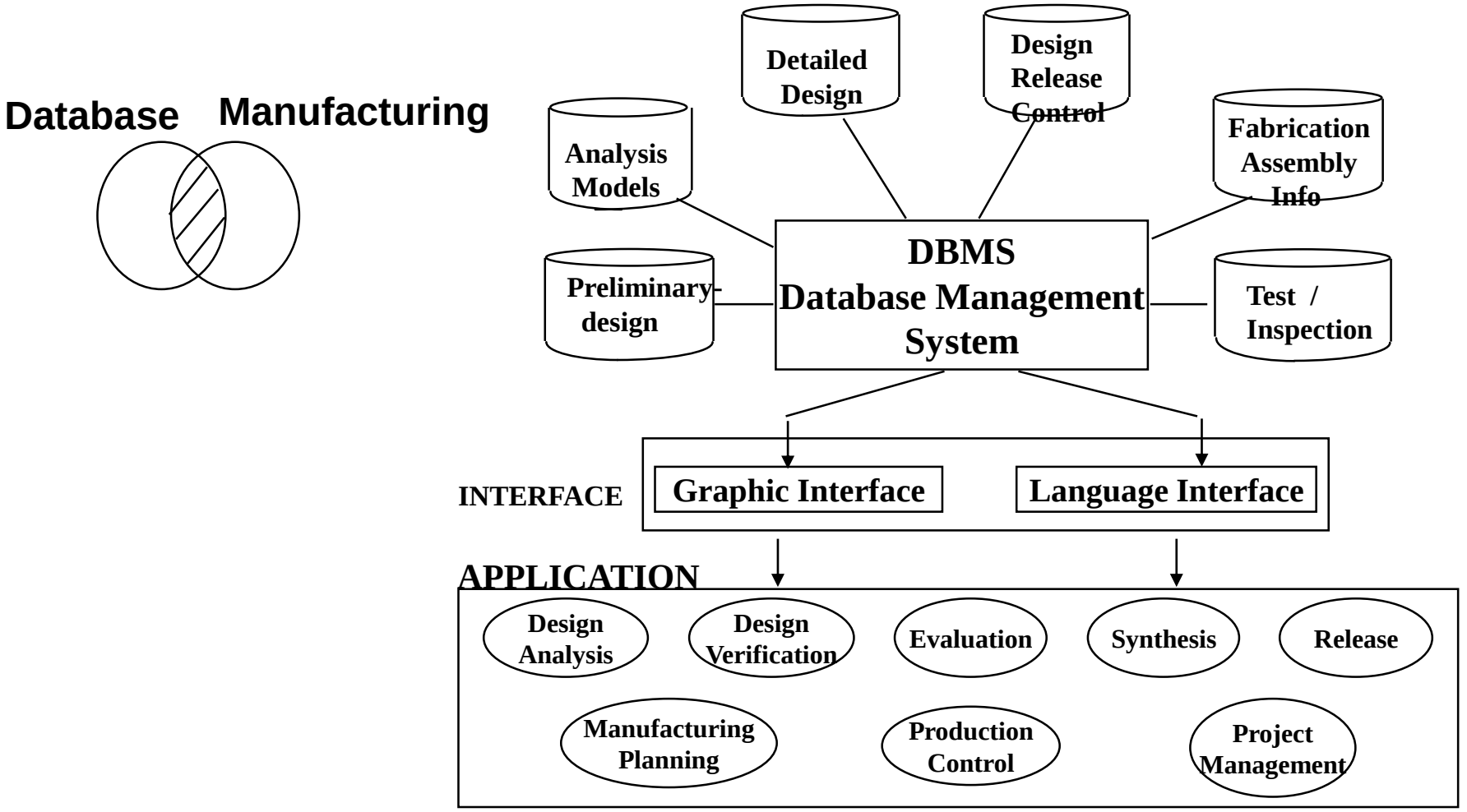
graphics

In the course introduction to Computer Science? students do their homework's on workstations.

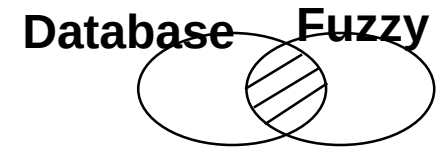


image

Use of a Database Management System in Design and Application



Fuzzy Database



- Fuzzy Query

```
<e.g.>      SELECT      STUDENT.NAME
            FROM        STUDENT
            WHERE       SEX = M
                        AND  HEIGH = TALLER
                        AND  WEIGH = SLIMMER
```

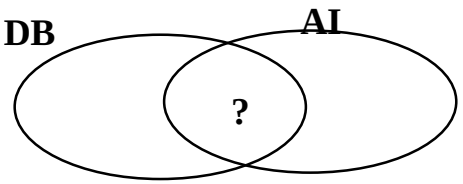
STUDENT:

NAME	SEX	HEIGHT	WEIGHT	IQ
Mary	F	158	55	High
Linda	F	165	55	Medium
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.

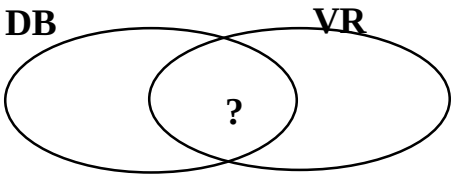
```
<e.g.>          SELECT    STUDENT.NAME
                FROM      STUDENT
                WHERE     IQ >= 130
```

More?

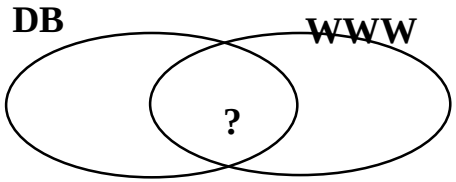
1993



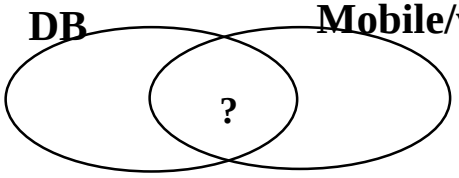
1995



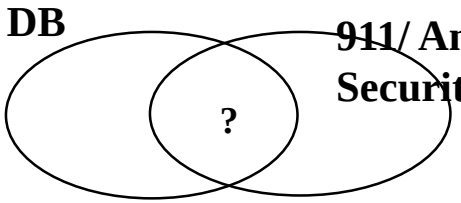
1997



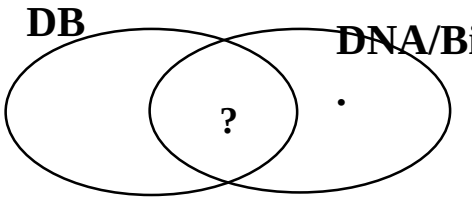
1999



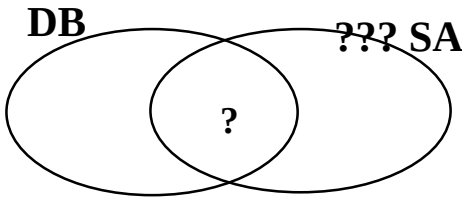
2001



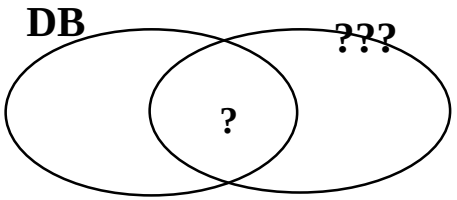
2003



2004



20xx



1.8 Discussion and Remarks

Discussion and Remarks

- Advantages of database systems
  - Easy to retrieve information!
  - Redundancy can be reduced
  - Inconsistency can be avoid
  - Data can be shared
  - Standards can be enforced
  - Security restrictions can be applied
  - Integrity can be maintained
  - Provision of data independence

- Disadvantages of database systems
  - Database design and control are a complicated matter.

Contents of Part I: 入門與導論

- **Unit 1 Introduction to DBMS**
- **Unit 2 DB2 and SQL**
- **Unit 3 The Relational Model**
- **Unit 4 The Hierarchical Model**
- **Unit 5 The Network Model**

- **References:**

1. C. J. Date, An Introduction to Database Systems, 8th edition, 2004.
2. J. D. Ullman, Principles of Database and Knowledge-Base, Vol.I, 1988.
3. Cited papers

Contents of PART II: 資料庫設計

- **Unit 6 Database Design and the E-R Model**
- **Unit 7 Normalization (表格正規化)**
- **Unit 8 User Interfaces (使用者介面)**
- **Unit 9 實作範例一:**
- **Unit 10 實作範例二:**

- **References:**

1. C. J. Date, An Introduction to Database Systems, 8th edition, 2004.
2. A. Silberschatz, etc., Database System Concepts, 5th edition, McGraw Hill, 2006
3. J. D. Ullman, Principles of Database and Knowledge-Base, vol. I, 1988.
4. Cited papers

Contents of PART III: 進階探討

- **Unit 11 Access Methods**
- **Unit 12 Database Recovery**
- **Unit 13 Concurrency Control**
- **Unit 14 Security and Integrity**
- **Unit 15 Query Optimization**
- **Unit 16 Distributed Database**
- **Unit 17 More on E-R Model**
- **Unit 18 More on Normalization**
- **Unit 19 More on User Interfaces**
- **Unit 20 More on X?**

- **References:**

1. C. J. Date, An Introduction to Database Systems, 8th edition, 2004.
2. J. D. Ullman, Principles of Database and Knowledge-Base, vol. I, 1988.
3. Cited papers

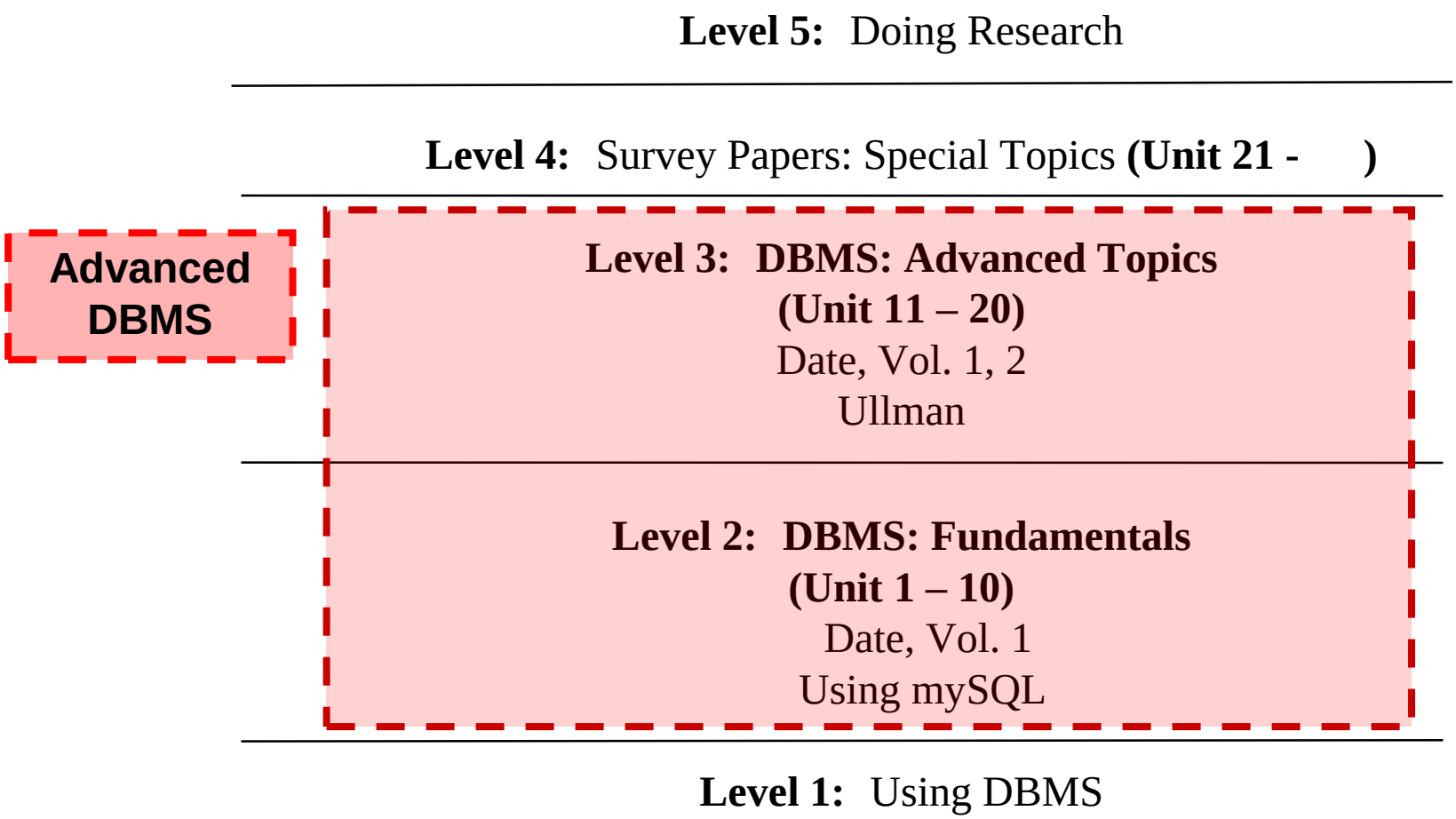
Contents of PART VI: 主題研究

- **Unit 21 Object-Oriented Database**
- **Unit 22 Logic-Based Database**
- **Unit 23 Image Database**
- **Unit 24 Multimedia Database**
- **Unit 25 Real-Time Database**
- **Unit 26 Parallel Database**
- **Unit 27 Temporal Database**
- **Unit 28 Active Database**
- **Unit 29 Bioinformatics Database**
- **Unit 30 ….**

- **References:**

  1. Cited papers

Study and Research on Databases



end of unit 1