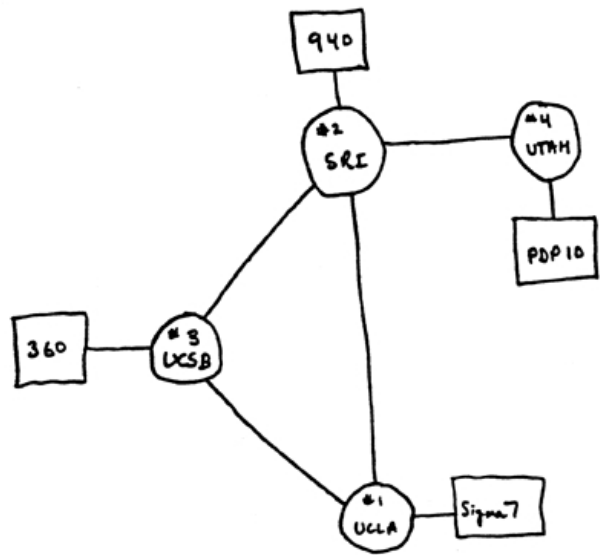


# Introduction to Computer Science Fall 2022 #10 Chi-Jen Wu

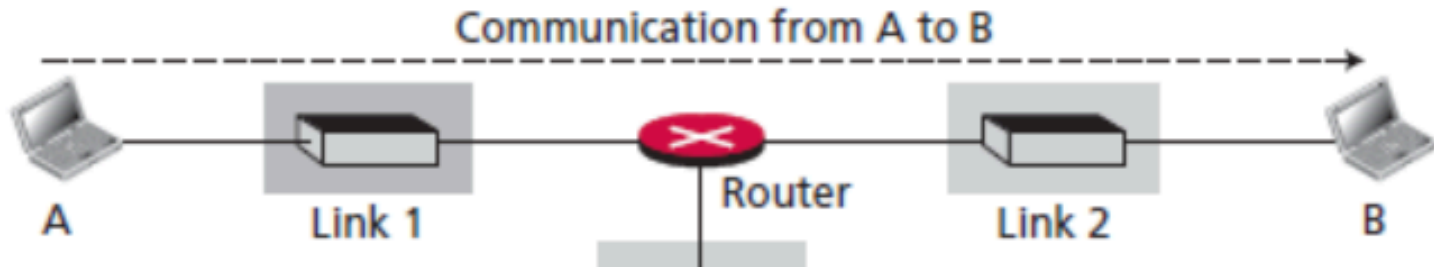
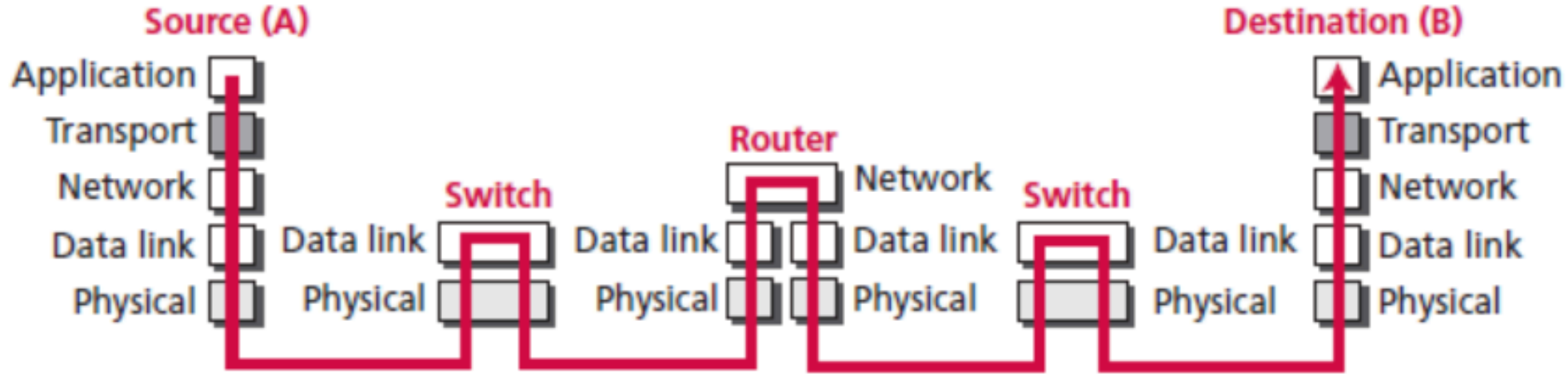
# 現代電腦網路主要概念

- **Autonomous** (自治)
  - 排長 (Router, Gateway)
- **Layer** (分層設計)
  - 學號 (Address)
  - 姓名 (DNS)
  - 傳送媒介 (Carrier)
- **Protocol** (協定)
  - TCP/IP (共同語言)

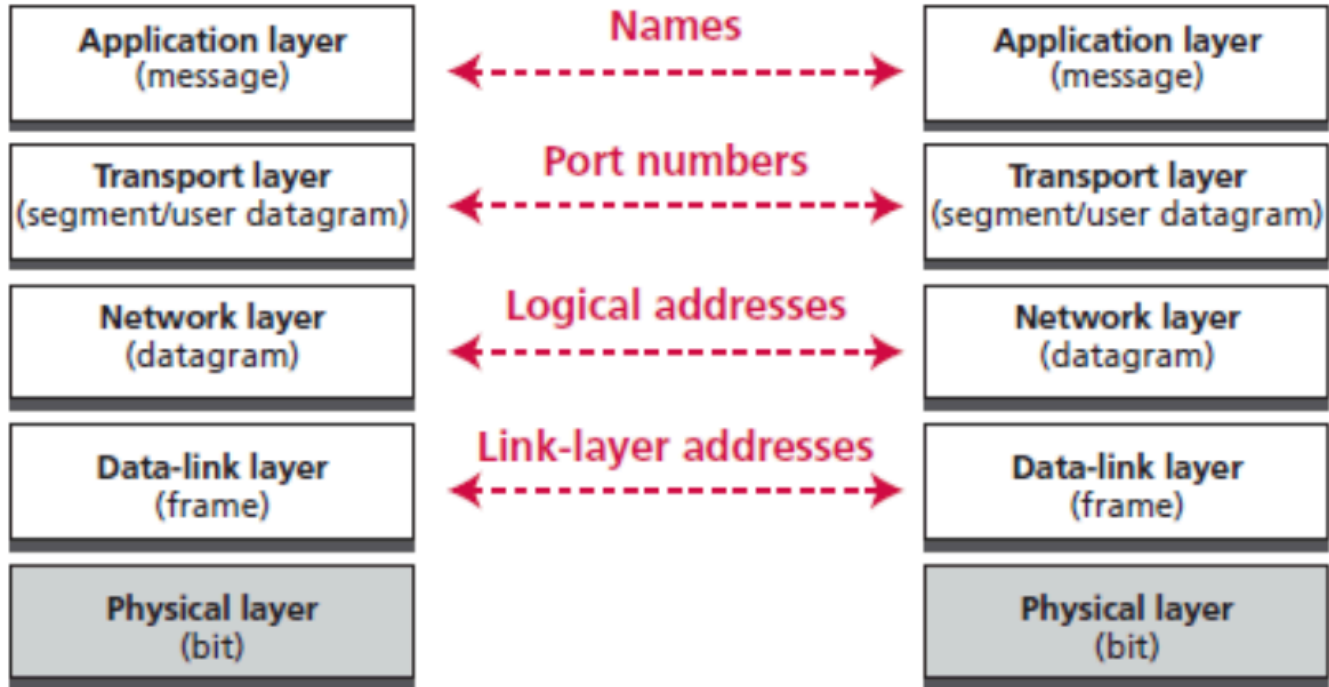


ARPANet

# Layered architecture



# Addressing and Session



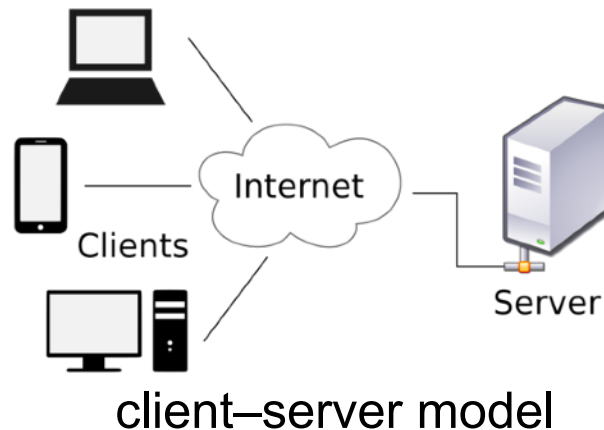


# Circuit Switching vs Packet Switching

Feature	Circuit Switching	Packet Switching
Dedicated Path	Yes	No
Path Formation	Path dedicated for one conversation	Route is established on a per packet basis of the conversation using datagram (or per conversation with virtual circuit)
Delay	Call setup delay	Packet transmission delay (call setup delay for virtual circuit)
Bandwidth Type	Fixed Bandwidth	Dynamic bandwidth
Overload Effects	Stops call establishment	Increases packet delay (can block call establishment and increase packet delay with virtual circuit)

# The services paradigms in the Internet

- Client–server paradigm
  - Simple, old-school
- Peer-to-peer paradigm
  - Complicated, distributed
- Clouding computing paradigm
  - Centralized outside, distributed inside
  - Complicated
  - Simple



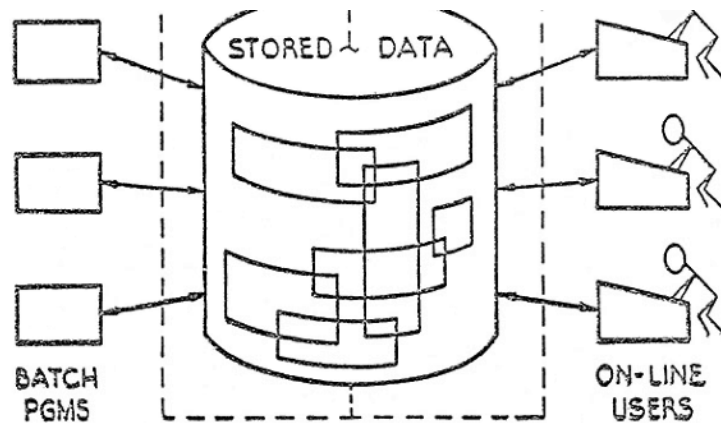


# The services over the Internet

- Ping / ICMP
- secure shell (SSH)
- Domain name service (DNS)
- World Wide Web (WWW)
  - HyperText Transfer Protocol (HTTP)
- Web Proxy
- File Transfer Protocol (FPT)
- Email/POP/SMTP

# Topics

- An Introduction to Computer Science
- The Shapes of Computers Today
- Computer Organization and Architecture
- Operating system
- Networking & The Internet
- **Database Systems**
- Software Engineering
- Cloud Platform/Cloud Shell Editor
- Cloud Platform/Cloud Source Repositories (git)
- C/C++ Programming







# Database Systems

其實又是一段跟分享有關的  
故事 ....

要共同維護一份資料



# Database Systems

- The history of database systems
- Database fundamentals
- Traditional File Structures
- Maintaining Database Integrity
- The Relational Model
- Object-Oriented Databases
- Data Mining

儲存和連結  
資料和  
資料之間的關係

A database is a collection of related, logically coherent, data used by the application programs in an organization.



# Introduction— why?

- Data storage used **individual, unrelated** files, called files
- **Each application program** used its own file
- In a university, each department might have its own **files**
  - A file about the student information and their grades
  - Name of the professors and the courses they were teaching
  - The whole staff

# 最早檔案儲存管理的想法

## 教授檔

教授A 課程1  
教授A 課程2  
教授B 課程3  
教授B 課程4  
教授C 課程5  
教授C 課程6

## 課程檔

課程1 課程資訊1  
課程2 課程資訊2  
課程3 課程資訊3  
課程4 課程資訊4  
課程5 課程資訊5  
課程6 課程資訊6

## 修課學生檔

課程1 學生1  
課程1 學生2  
課程1 學生3  
課程1 學生4  
課程1 學生5  
課程1 學生6  
課程2 學生1  
課程2 學生2  
課程2 學生3

● 問題：當我們要印出課程1的參與老師和學生時 怎麼辦？

- 問題：當我們要印出課程1的參與老師和學生時
  - 教授A  $\rightarrow$  課程1  $\leftarrow$  [學生1, 學生2, 學生3]
- 這時候我們要怎麼辦？怎麼辦？怎麼辦？
- All of these files can be combined in a single database,





● 當我們要印出 課程1的參與老師和學生時

● 教授A → 課程1 ← [學生1, 學生2, 學生3]

教授 + 課程 + 修課學生檔

教授A	課程1	課程資訊1	學生1
教授A	課程1	課程資訊1	學生2
教授A	課程1	課程資訊1	學生3
教授A	課程1	課程資訊1	學生4
教授A	課程1	課程資訊1	學生5
教授A	課程1	課程資訊1	學生6

聚合成一張很大的表

問題：教授 + 課程 + 修課學生 這樣會有什麼問題？



# We do need a database

- 問題：當我們要印出
  - 教授A → 課程1 ← [學生1, 學生2, 學生3]
- All of these files can be combined in a single database
- 處理資料的整合呈現
- 需要一個系統來管理資料 < — database system



# The advantages for a database system

- Advantages of databases

- Less redundancy**

- A lot of redundancy in files

- Inconsistency avoidance**

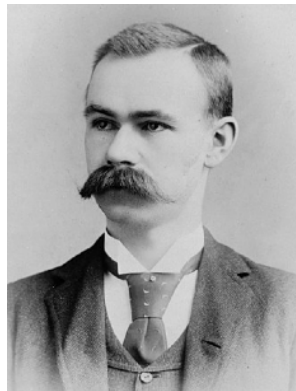
- Any changes in the data need to occur in all places

教授A	課程1
教授A	課程2
教授B	課程3
教授B	課程4
教授C	課程5
教授C	課程6



# The history of database systems

- 赫爾曼·何樂禮, 1882
  - 現代機械數據處理之父
- 美國人口普查
  - 8 年 —> 1 年
- 1924, IBM
  - System/360, 1969
  - DB2, 1980



赫爾曼·何樂禮



# 美國人口普查, 1888

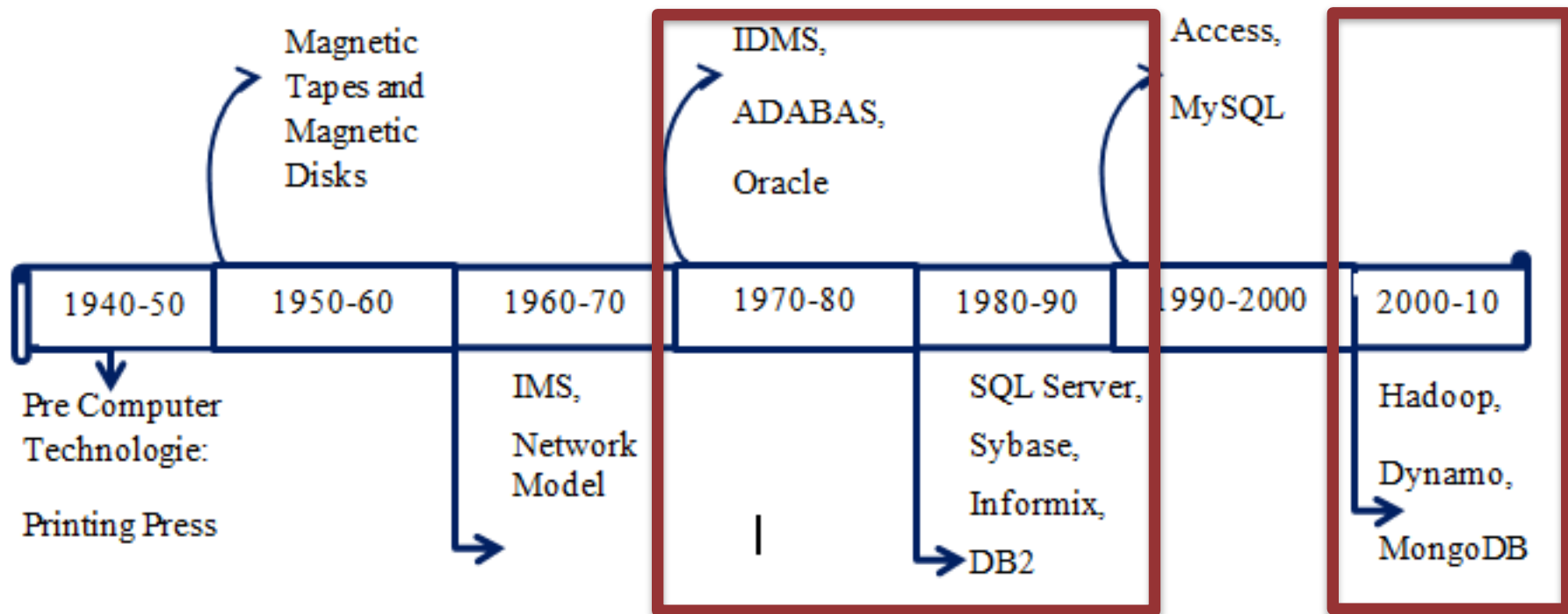
- 何樂禮設計的打孔卡片

1	1	3	0	2	4	10	On	S	A	C	E	a	c	e	g		EB	SB	Ch	Sy	U	Sh	Hk	Br	Rm
2	2	4	1	3	E	15	Off	IS	B	D	F	b	d	f	h		SY	X	Fp	Cn	R	X	Al	Cg	Kg
3	0	0	0	0	W	20			0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0
A	1	1	1	1	0	25	A	1	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1
B	2	2	2	2	5	30	B	2	2		2	2	2	2	2	2		2		2	2	2	2	2	2
C	3	3	3	3	0	3	C	3	3	3		3	3	3	3	3		3	3		3	3	3	3	3
D	4	4	4	4	1	4	D	4	4	4	4		4	4	4	4		4	4	4		4	4	4	4
E	5	5	5	5	2		E	5	5	5	5	5		5	5	5	5	5	5	5	5		5	5	5
F	6	6	6	6	A	D	F	6	6	6	6	6	6		6	6	6	6	6	6	6	6		6	6
Q	7	7	7	7	B	E	Q	7	7	7	7	7	7	7		7	7	7	7	7	7	7	7		7
H	8	8	8	8	a	F	H	8	8	8	8	8	8	8		8	8	8	8	8	8	8	8		8
I	9	9	9	9	b	c	I	9	9	9	9	9	9	9		9	9	9	9	9	9	9	9		9

# 人口普查資料量龐大

- 電腦開始發展, 1950
- 普查資料也開始利用電腦的計算能力
- 但是資料量龐大
  - 若單以檔案儲存
  - 無法管理並利用這龐大的資料
  - 人口數
    - 性別 年紀 血型 居住地 通訊地 聯絡方式
- 電腦工程師們開始構想 如何利用電腦管理大量的資料

# Database 發展



商用DB開始發展

雲端DB開始發展



# Oracle database systems

- University of California, Berkeley, 1973
  - Ingres project
- Be commercialized by Oracle Corp (甲骨文)
  - 1979
- Oracle database
  - 基本上應該是這個地球上最強的單一資料庫

# 資料庫市佔排名



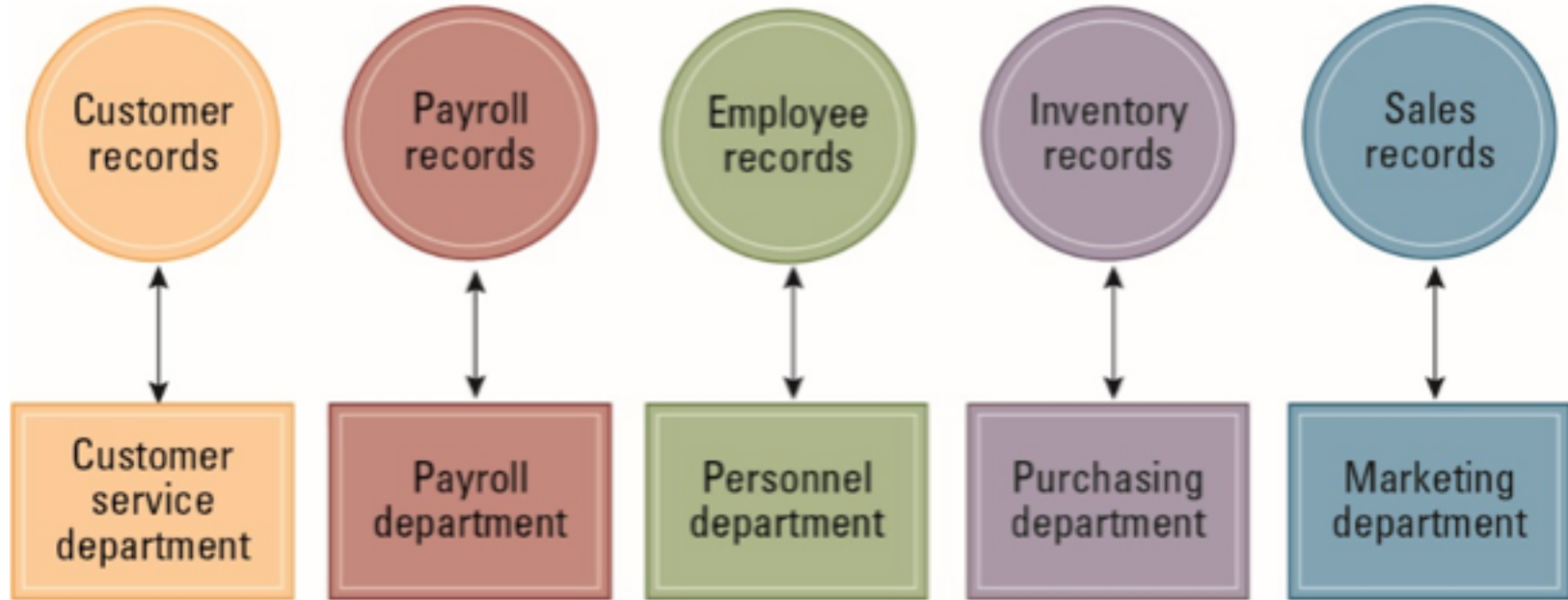
排名	2017.01	2016.01	名称	类型	主导公司	分数	去年 同比增减
➡	1	1	Oracle	Relational DBMS	Oracle	1416.72	-79.36
➡	2	2	MySQL	Relational DBMS	Oracle	1366.29	67.03
➡	3	3	Microsoft SQL Server	Relational DBMS	Microsoft	1220.95	76.89
➡	4	4	MongoDB	Document store	MongoDB	331.90	25.88
➡	5	5	PostgreSQL	Relational DBMS	PostgreSQL Global Development	330.37	47.97
	6	6	DB2	Relational DBMS	IBM	182.49	-13.88
	7 ↑	8	Cassandra	*Wide column store	DataStax , Apache	136.44	5.49
	8 ↓	7	Microsoft Access	Relational DBMS	Microsoft	127.45	-6.59
➡	9 ↑	10	Redis	Key-value store	redislabs	118.70	17.54
➡	10 ↓	9	SQLite	Relational DBMS	hwaci	112.38	8.64
➡	11 ↑	12	Elasticsearch	Search engine	Elastic	106.17	28.96
	12 ↑	14	Teradata	Relational DBMS	Teradata	74.17	-0.78
	13 ↓	11	SAP Adaptive Server	Relational DBMS	SAP	69.10	-14.08
	14 ↓	13	Solr	Search engine	Apache	68.08	-7.32
	15 ↑	16	HBase	Wide column store	cloudera , Apache	59.14	5.77
	16 ↑	18	Splunk	Search engine	Splunk	55.49	12.37
	17	17	FileMaker	Relational DBMS	Apple	53.49	4.66
	18 ↑	19	SAP HANA	Relational DBMS	SAP	51.93	13.32
	19 ↓	15	Hive	Relational DBMS	cloudera , Apache	51.14	-2.45
	20 ↑	23	MariaDB	Relational DBMS	MariaDB Foundation	45.04	17.28



# Database fundamentals

- A collection of data
  - **Multidimensional**
  - Internal links (**relationship**)
    - make the information accessible from a variety of **perspectives**
- 假設資料很多，無法全部放到記憶體裡！
  - 現在是“可以”全部放到記憶體裡的，但是大部分不會
- **File-oriented database system**
  - Traditional File Structures
- **Database system**

# File-oriented database system



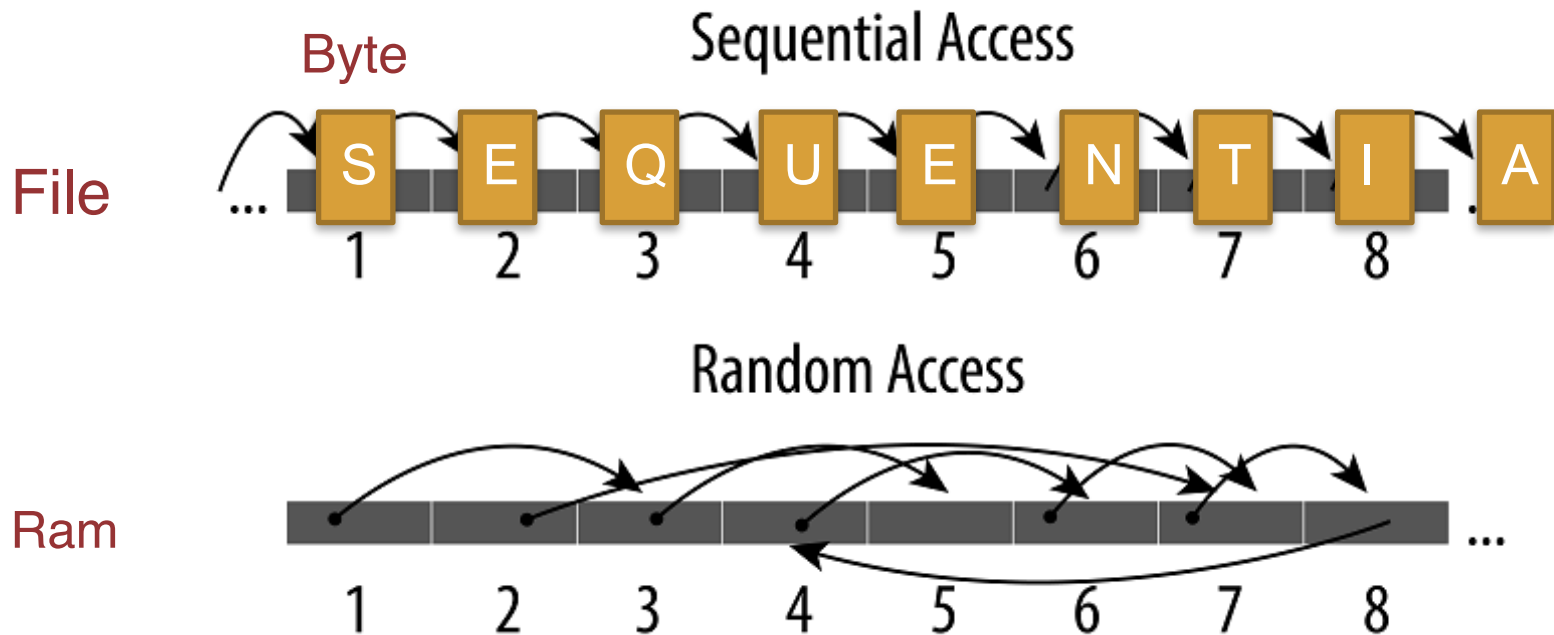




# Traditional File Structures

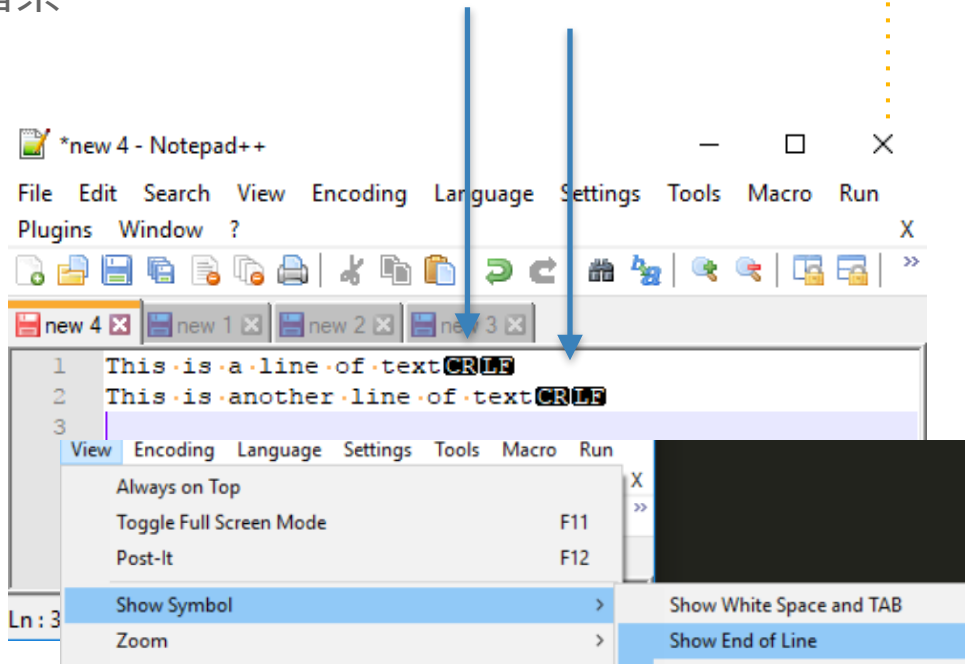
- Sequential file
  - A file whose contents can only be read in order
- Reader must be able to detect end-of-file (**EOF**)
- Data can be stored in logical records, sorted by a key field
  - Greatly increases the speed of batch updates

# Sequential files, text file



# 一個text file組成

- 字元原生編碼構成的二進位電腦檔案
  - **ASCII, UTF-8**
  - 通用的、跨平台的
- 一行text的結束
- 換行符號 CR LF
  - Carriage Return (**\n**)
    - **0x0A**
  - Line feed (**\r**)
    - **0x0D**



不要小看這個CRLF 他拿來攻擊網站

# 換行符號 CR LF

- 老式機械打字機
  - CR和LF分別具有不同的作用
    - LF會將列印紙張上移一行位置
    - CR則會將滾回到列印紙張的最左側
- Windows
  - CR (\n)
  - LF (\r)
- Others
  - \n or \r



# End-Of-File (EOF)

- EOF 定義為檔案結束
  - Standart I/O
- 視為檔案結束了 →
- `stdio.h`



```
/* The value returned by fgetc and similar  
   functions to indicate the end of the file. */  
#define EOF (-1)
```

# 回到最早檔案儲存管理的想法

## 教授檔

教授A 課程1  
教授A 課程2  
教授B 課程3  
教授B 課程4  
教授C 課程5  
教授C 課程6

## 課程檔

課程1 課程資訊1  
課程2 課程資訊2  
課程3 課程資訊3  
課程4 課程資訊4  
課程5 課程資訊5  
課程6 課程資訊6

## 修課學生檔

課程1 學生1  
課程1 學生2  
課程1 學生3  
課程1 學生4  
課程1 學生5  
課程1 學生6  
課程2 學生1  
課程2 學生2  
課程2 學生3

● 問題：當我們要印出課程1的參與老師和學生時 怎麼辦？

# key field

- 每一筆資料有一個獨一無二的**Key**
- 避免資料內容一樣
  - 同名同姓 課程名稱一樣
- 確定在檔案資料位址

- 1 教授A 課程1
- 2 教授A 課程2
- 3 教授B 課程3
- 4 教授B 課程4
- 5 教授C 課程5
- 6 教授C 課程6

學年/學期 Semester	科目代號 Course Code	開課序號 Serial No.	開課單位 Department	年級 Year	課程名稱 Course Title	授課教師 Instructor	學分 Credits	上課時間 Class Sched.	選課人數 Seats
110/ 1	EN1070	46521	電子系	1	計算機概論 Introduction to Computer Science <small>詳細資料 Detail</small>	游智仁	3 <small>必修</small>	Tue 09:10-12:00	52/60
110/ 1	EN1070	46531	電子系	1	計算機概論 Introduction to Computer Science <small>詳細資料 Detail</small>	吳志峰	3 <small>必修</small>	Fri 09:10-12:00	60/60
110/ 1	IT1015	46642	資工系	1	計算機概論(1) Introduction to Computer Science( I ) <small>詳細資料 Detail</small>	吳齊人	3 <small>必修</small>	Wed 15:10-17:00 Thu 15:10-17:00	71/60
110/ 1	IM1203	46846	資管系	1	計算機概論 Introduction to Computer Science <small>詳細資料 Detail</small>	林維昭	3 <small>必修</small>	Wed 09:10-12:00	61/60
110/ 1	IM1203	46883	資管系	1	計算機概論 Introduction to Computer Science <small>詳細資料 Detail</small>	陳昱仁	3 <small>必修</small>	Tue 09:10-12:00	60/60
110/ 1	AI1003	47395	人工智慧學士學程	1	計算機概論(1) Introduction to Computer Science( I ) <small>詳細資料 Detail</small>	張賢宗	3 <small>必修</small>	Tue 10:10-12:00 Thu 08:10-10:00	32/60

key field (id)

流水號

# 進化後檔案儲存管理

## 教授檔

1	教授A	課程1
2	教授A	課程2
3	教授B	課程3
4	教授B	課程4
5	教授C	課程5
6	教授C	課程6

## 課程檔

1	課程1	課程資訊1
2	課程2	課程資訊2
3	課程3	課程資訊3
4	課程4	課程資訊4
5	課程5	課程資訊5
6	課程6	課程資訊6

## 修課學生檔

1	課程1	學生1
2	課程1	學生2
3	課程1	學生3
4	課程1	學生4
5	課程1	學生5
6	課程1	學生6
7	課程2	學生1
8	課程2	學生2
9	課程2	學生3



key field (id) key field



# 怎麼一堆重複的資料！想辦法弄掉！

## 教授檔

1	教授A	課程1
2	教授A	課程2
3	教授B	課程3
4	教授B	課程4
5	教授C	課程5
6	教授C	課程6

## 課程檔

1	課程1	課程資訊1
2	課程2	課程資訊2
3	課程3	課程資訊3
4	課程4	課程資訊4
5	課程5	課程資訊5
6	課程6	課程資訊6

## 修課學生檔

1	課程1	學生1
2	課程1	學生2
3	課程1	學生3
4	課程1	學生4
5	課程1	學生5
6	課程1	學生6
7	課程2	學生1
8	課程2	學生2
9	課程2	學生3



key field

key field

# 進化檔案儲存內容

## 教授檔

- 1 教授A 皮卡丘
- 2 教授B 雷丘
- 3 教授C 妙蛙

## 課程檔

- 1 課程1 微積分
- 2 課程2 計算機概論
- 3 課程3 人工智慧概論
- 4 課程4 基礎英文
- 5 課程5 普通物理學
- 6 課程6 體育大一

## 學生檔

- 1 學生1 小智
- 2 學生2 大智
- 3 學生3 大木
- 4 學生4 大大
- 5 學生5 小弱
- 6 學生6 大弱



沒有重複的資料了！喔耶！但感覺怪怪的？

# 資料關係不見了！



## P檔

- 1 教授A 皮卡丘
- 2 教授B 雷丘
- 3 教授C 妙蛙

## C檔

- 1 課程1 微積分
- 2 課程2 計算機概論
- 3 課程3 人工智慧概論
- 4 課程4 基礎英文
- 5 課程5 普通物理學
- 6 課程6 體育大一

## S檔

- 1 學生1 小智
- 2 學生2 大智
- 3 學生3 大木
- 4 學生4 大大
- 5 學生5 小弱
- 6 學生6 大弱



教授A 教了什麼？那些學生上課程1？

# 補漏進化檔案儲存方式 — 修補關係



## P檔

- 1 教授A 皮卡丘
- 2 教授B 雷丘
- 3 教授C 妙蛙

## C檔

- 1 課程1 微積分
- 2 課程2 計算機概論
- 3 課程3 人工智慧概論
- 4 課程4 基礎英文
- 5 課程5 普通物理學
- 6 課程6 體育大一

## S檔

- 1 學生1 小智
- 2 學生2 大智
- 3 學生3 大木
- 4 學生4 大大
- 5 學生5 小弱
- 6 學生6 大弱



## 教授教學檔

- 1 P.1 C.1
- 2 P.1 C.2
- 3 P.1 C.3
- 4 P.2 C.4
- 5 P.2 C.5

## 學生修課檔

- 1 C.1 S.1
- 2 C.1 S.2
- 3 C.1 S.3
- 4 C.1 S.4
- 5 C.1 S.5
- 6 C.1 S.6

# P檔

- 1 教授A 皮卡丘
- 2 教授B 雷丘
- 3 教授C 妙蛙

# C檔

- 1 課程1 微積分
- 2 課程2 計算機概論
- 3 課程3 人工智慧概論
- 4 課程4 基礎英文
- 5 課程5 普通物理學
- 6 課程6 體育大一

# S檔

- 1 學生1 小智
- 2 學生2 大智
- 3 學生3 大木
- 4 學生4 大大
- 5 學生5 小弱
- 6 學生6 大弱



- 1 P.1 C.1
- 2 P.1 C.2
- 3 P.1 C.3
- 4 P.2 C.4
- 5 P.2 C.5

教授教學檔

- 1 C.1 S.1
- 2 C.1 S.2
- 3 C.1 S.3
- 4 C.1 S.4
- 5 C.1 S.5
- 6 C.1 S.6

學生修課檔

CRUD 操作:

Create

Read

Update

Delete

# P檔

- 1 教授A 皮卡丘
- 2 教授B 雷丘
- 3 教授C 妙蛙

4 教授D 種子

# C檔

- 1 課程1 微積分
- 2 課程2 計算機概論
- 3 課程3 人工智慧概論
- 4 課程4 基礎英文
- 5 課程5 普通物理學
- 6 課程6 體育大一

7 課程7 軍訓課程

# S檔

- 1 學生1 小智
- 2 學生2 大智
- 3 學生3 大木
- 4 學生4 大大
- 5 學生5 小弱
- 6 學生6 大弱

7 學生7 王大寶

1 P.1 C.1  
2 P.1 C.2  
3 P.1 C.3  
4 P.1 C.4

3 C.1 S.3  
4 C.1 S.4

新增一個老師 課程 學生

教授教學檔

學生修課檔



CRUD 操作:

Create

Read

Update

Delete

# P檔

- 1 教授A 皮卡丘
- 2 教授B 雷丘
- 3 教授C 妙蛙

# C檔

- 1 課程1 微積分
- 2 課程2 計算機概論
- 3 課程3 人工智慧概論
- 4 課程4 基礎英文
- 5 課程5 普通物理學
- 6 課程6 體育大一

# S檔

- 1 學生1 小智
- 2 學生2 大智
- 3 學生3 大木
- 4 學生4 大大
- 5 學生5 小弱
- 6 學生6 大弱



- 1 P.1 C.1
- 2 P.1 C.2
- 3 P.1 C.3
- 4 P.2 C.4

- 1 C.1 S.1
- 2 C.1 S.2
- 3 C.1 S.3
- 4 C.1 S.4
- 5 C.1 S.5

讀取一個老師 課程 學生

教授教學檔

學生修課檔

CRUD 操作:

Create

Read

Update

Delete

# P檔

- 1 教授A 皮卡丘
- 2 教授B 雷丘
- 3 教授C 妙蛙



3 教授C 妙蛙種子

1	P.1	C.1
2	P.1	C.2
3	P.1	C.3

教授教學檔

# C檔

- 1 課程1 微積分
- 2 課程2 計算機概論
- 3 課程3 人工智慧概論
- 4 課程4 基礎英文
- 5 課程5 普通物理學
- 6 課程6 體育大一

1	C.1	S.1
2	C.1	S.2
3	C.1	S.3
4	C.1	S.4

學生修課檔

# S檔

- 1 學生1 小智
- 2 學生2 大智
- 3 學生3 大木
- 4 學生4 大大
- 5 學生5 小弱
- 6 學生6 大弱



CRUD 操作:

Create

Read

Update

Delete

更新/刪除一個老師 課程 學生



# P檔

- 1 教授A 皮卡丘
- 2 教授B 雷丘
- 3 教授C 妙蛙

# C檔

- 1 課程1 微積分
- 2 課程2 計算機概論
- 3 課程3 人工智慧概論
- 4 課程4 基礎英文
- 5 課程5 普通物理學
- 6 課程6 體育大一

# S檔

- 1 學生1 小智
- 2 學生2 大智
- 3 學生3 大木
- 4 學生4 大大
- 5 學生5 小弱
- 6 學生6 大弱



6 P.3 C6

1	P.1	C.1
2	P.1	C.2
3	P.1	C.3
4	P.2	C.4
5	P.2	C.5

教授教學檔

1	C.1	S.1
2	C.1	S.2
3	C.1	S.3
4	C.1	S.4
5	C.1	S.5
6	C.1	S.6

學生修課檔

7 C.2 S.6

**CRUD** 操作:

Create

Read

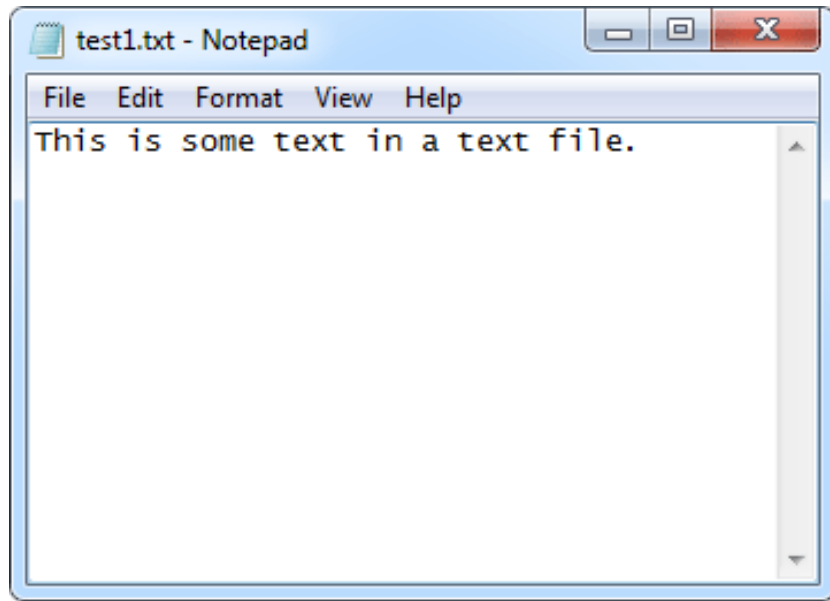
Update

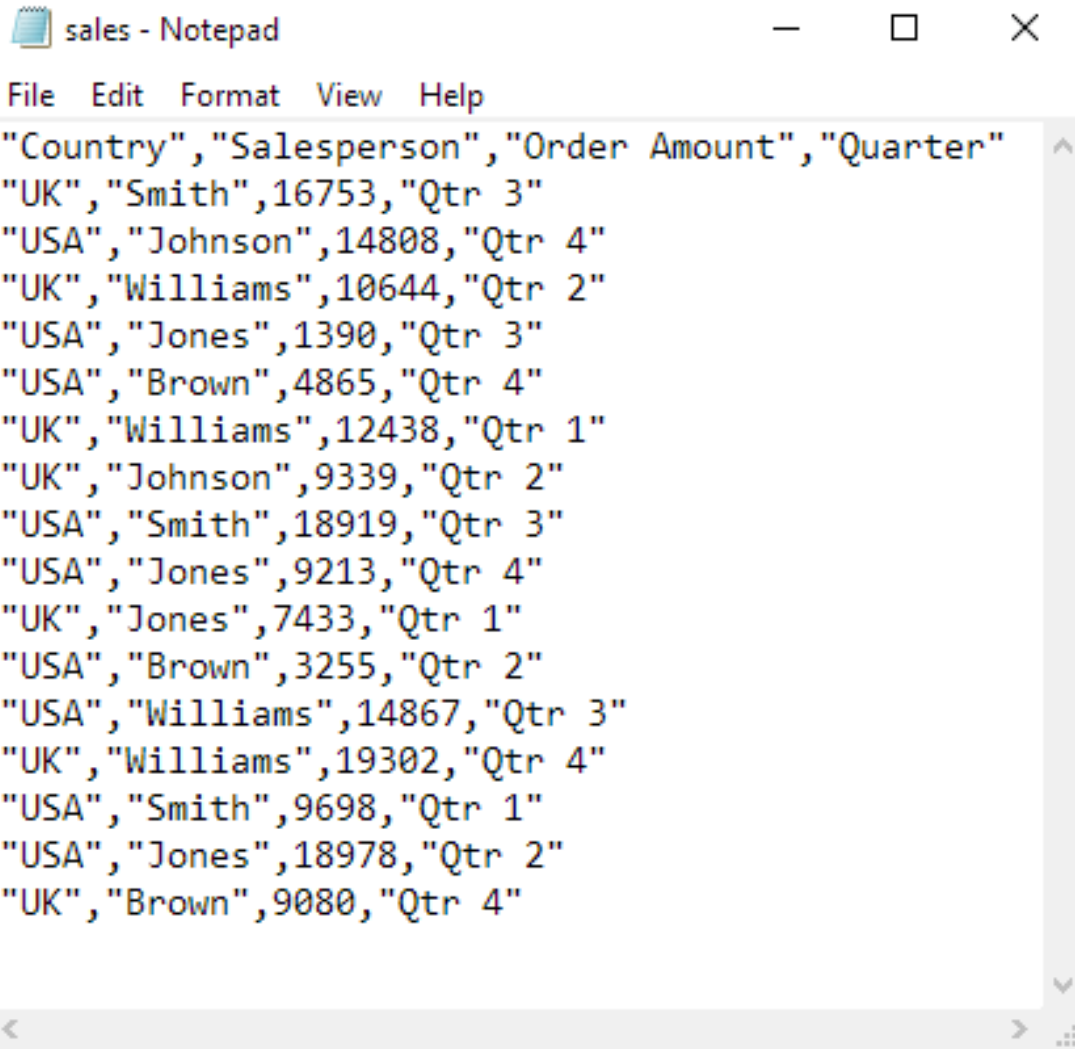
Delete

新增一個老師課程學生關係

# 單純text file 遇到的問題

- 目前我們是用text file
  - 可以空白或是，
  - 來區別資料
- 但是如果資料本身有
  - 空白或是，
  - 怎麼辦？





```
sales - Notepad
File Edit Format View Help
"Country","Salesperson","Order Amount","Quarter"
"UK","Smith",16753,"Qtr 3"
"USA","Johnson",14808,"Qtr 4"
"UK","Williams",10644,"Qtr 2"
"USA","Jones",1390,"Qtr 3"
"USA","Brown",4865,"Qtr 4"
"UK","Williams",12438,"Qtr 1"
"UK","Johnson",9339,"Qtr 2"
"USA","Smith",18919,"Qtr 3"
"USA","Jones",9213,"Qtr 4"
"UK","Jones",7433,"Qtr 1"
"USA","Brown",3255,"Qtr 2"
"USA","Williams",14867,"Qtr 3"
"UK","Williams",19302,"Qtr 4"
"USA","Smith",9698,"Qtr 1"
"USA","Jones",18978,"Qtr 2"
"UK","Brown",9080,"Qtr 4"
```



# Text based logical records

- 1. 雙引號 和 逗號，組成
- 2. 利用程式語言裡的 data structure
  - 定義 binary based logical record

# Text based logical records

P檔

1 教授A 皮卡丘  
2 教授B 雷丘  
3 教授C 妙蛙

New P檔

"1", "皮卡丘", "資工" \n  
"2", "雷丘", "電子" \n  
"3", "妙蛙", "資管" \n

Logical record

我們可以定義一個老師應該有什麼屬性  
在P檔裡 目前一個教師會有

屬性

1. id (獨一無二的編號)
2. 姓名
3. 系所

Logical record

# Text based logical records 問題

New P檔

“1”，“皮卡丘”，“資工”  
“2”，“雷丘”，“電子”  
“3”，“妙蛙”，“資管”



Sequential read

讀取的順序還是要依照text file的規定

一行一行照順序讀

要讀取第10個，你要從第一個開始數

無法跳到第十行，你不知道每一行多長！

問題：如果要讀取第一萬個教師怎麼辦？

# Logical records 再進化



- 電腦處理是binary的
- 所以用電腦的思考方式
- Logical records 可以用binary 來思考

```
00000000 7B 22 62 69 6E 64 53 74 61 63 6B 22 3A 6E 75 6C {"bindStack":nul
00000010 6C 2C 22 62 69 6E 64 54 69 6D 65 22 3A 6E 75 6C 1,"bindTime":nul
00000020 6C 2C 22 63 6F 6E 73 6F 6C 65 22 3A 5B 5D 2C 22 1,"console":[],"
00000030 63 75 73 74 6F 6D 65 72 22 3A 7B 22 61 70 70 6C customer":{"appl
00000040 69 63 61 74 69 6F 6E 22 3A 22 68 6F 6D 65 70 61 ication":"homepa
00000050 67 65 22 2C 22 63 6F 72 72 65 6C 61 74 69 6F 6E ge","correlation
00000060 49 64 22 3A 22 62 39 34 30 32 65 32 32 2D 33 39 Id":"b9402e22-39
00000070 65 31 2D 34 34 65 66 2D 38 36 31 34 2D 63 33 36 el-44ef-8614-c36
00000080 37 64 32 65 62 32 37 61 34 22 2C 22 73 65 73 73 7d2eb27a4","sess
00000090 69 6F 6E 49 64 22 3A 22 22 2C 22 74 6F 6B 65 6E ionId":"","token
000000A0 22 3A 22 32 38 35 35 63 37 32 63 31 39 66 32 34 ":"2855c72c19f24
000000B0 32 31 37 61 39 63 34 35 30 36 65 34 34 38 61 34 217a9c4506e448a4
000000C0 65 35 31 22 2C 22 75 73 65 72 49 64 22 3A 22 22 e51","userId":""
```

# Binary Logical records



- 再次定義 教師的 logical record
- P.id
  - Integer (4 bytes)
- P.name
  - 64 bytes
- P.dept
  - 128 bytes

Integer (4 bytes)

64 characters (64 bytes)

128 characters (128 bytes)

# 教師 Binary Logical records

	P.id	P.name	P.dept	
0	Integer (4 bytes)	64 characters (64 bytes)	128 characters (128 bytes)	195
196	Integer (4 bytes)	64 characters (64 bytes)	128 characters (128 bytes)	391
392	Integer (4 bytes)	64 characters (64 bytes)	128 characters (128 bytes)	587

$$3 * 196 = 588 \text{ bytes}$$



# Real Binary Logical records

P.id	P.name	P.dept
1	皮卡丘	資工
2	雷丘	電子
3	CJ	CS

000011

CJ\0

CS\0

編碼ASCII

經過編碼ASCII

434A00000....

435300000 ....

## Random Access in Binary Logical records

	P.id	P.name	P.dept
0	1	皮卡丘	資工
196	2	雷丘	電子
392	3	CJ	CS

要讀第三個老師，可以直接跳到  $392 (196 * (3-1))$

一次拿196個位元 就是第三個

要讀第十個 直接 要硬碟去讀 1764 位址

要讀第一萬個 就是 1,959,804 這個位址



# Binary Logical records DB

- 以上就是很簡單的檔案形式資料庫
- 但是其實還沒完
- 如果考慮到資料刪除的狀況
- 你以為的第三個，可能不是第三個
- 此話怎講？

CRUD 操作:

Create

Read

Update

Delete



# 跳號情況 in Binary Logical records

	P.id	P.name	P.dept
0	1	皮卡丘	資工
	2	雷丘	電子
196	3	CJ	CS



要讀第三個老師，可以直接跳到  $392 (196 * (3-1))$   
 一次拿196個位元 就是第三個  
這句話就不成立了！！！！

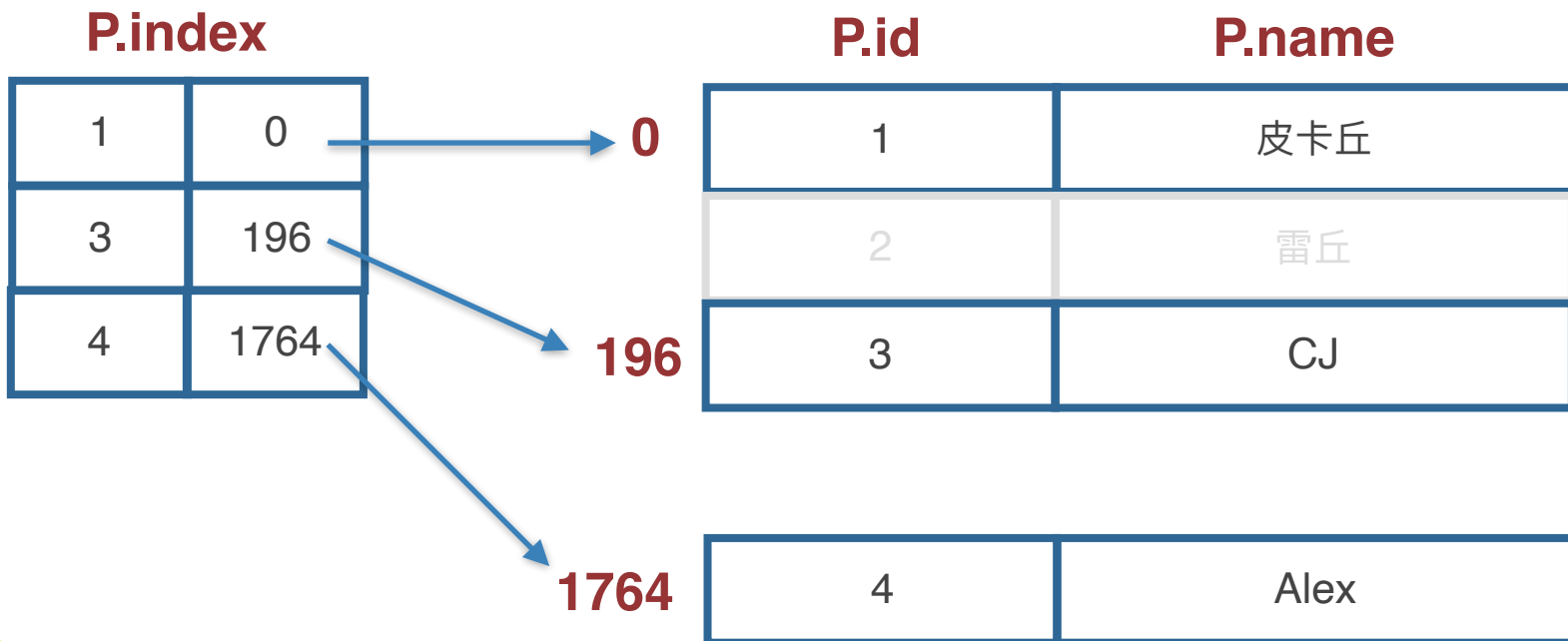
# 考慮到資料刪除的狀況

- 不要真的刪除資料
  - 留下資料
  - 多一個屬性，是否在職
- 加入index 技術
  - A separate file
  - A list of key values and the location of their **associated records**
  - **Efficient way** to quickly identify the **location of a desired record**



# Index 技術

- 類似書的目錄



# 電腦科學中的 index

- Index 技術在電腦科學中非常重要
- 應用範圍非常廣
  - 你每天用的google 也是靠特別的 index 技術
- 大概也是整個電腦科學都會使用的技術
- 先記住他是一種有排序的資料結構
  - 利用索引鍵值來指向一個資料區塊

4	1764
---	------

索引鍵值



# The Role of Schemas

- Schema

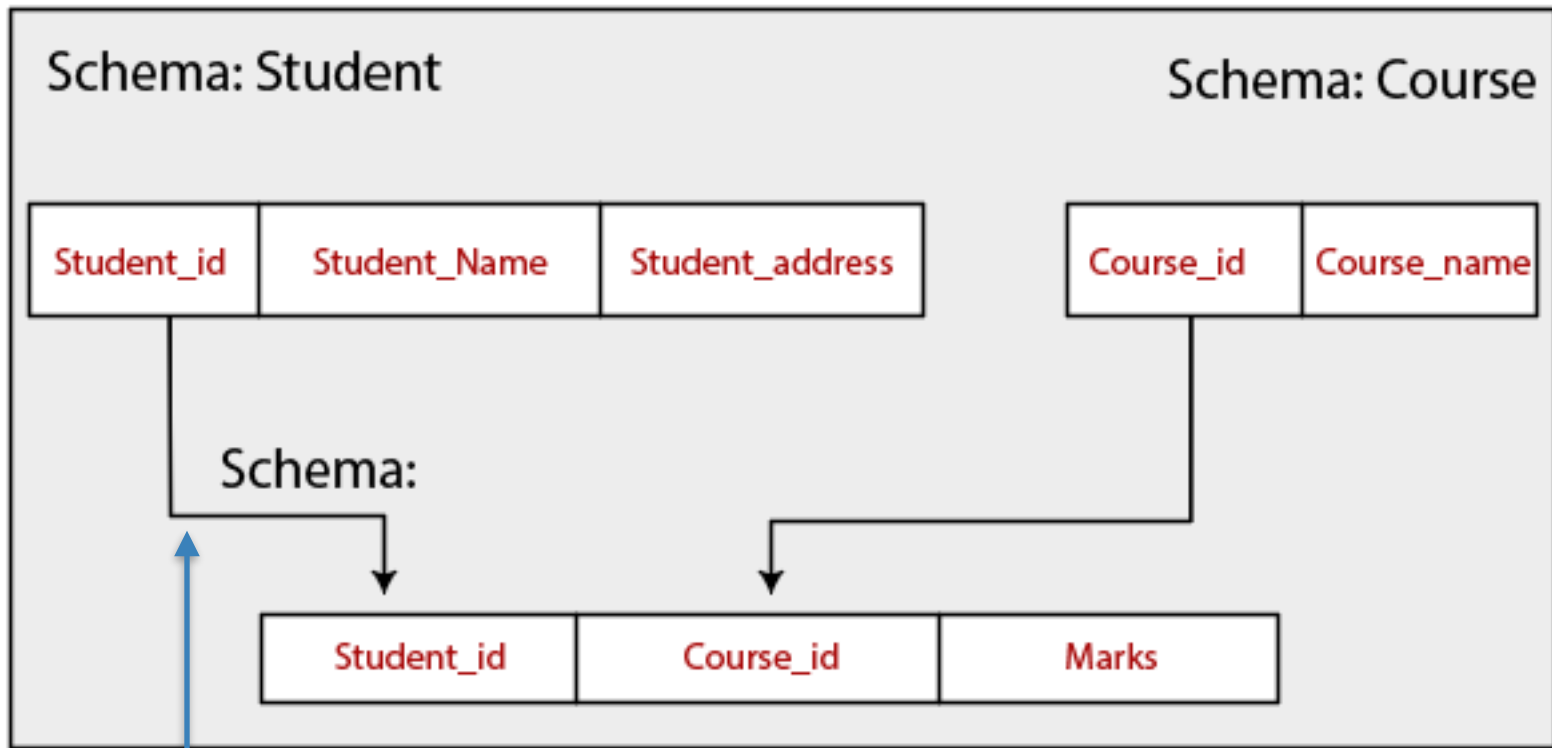
- A description of the structure of an database
  - 表格結構，欄位格式，資料關聯

- Subschema

- A description of the portion of the database to a particular user's needs
  - to prevent **sensitive** data from being accessed by unauthorized personnel



# Schema (Logical record)



資料的關係

# LAB#5 Deal with csv files

- 請畫出 Schema
  - Lab5-csvfiles.zip
    - departments.csv
    - employees.csv
- 兩個檔案的Schema
  - 類似 P57頁的圖
- 繳交一個 pdf 實驗報告上傳Google表單
  - 11/16號 15:00



# Conclusion

- The history of database systems
- Database fundamentals
- Traditional File Structures
- Maintaining Database Integrity
- The Relational Model
- Object-Oriented Databases
- Data Mining

儲存和連結  
資料和  
資料之間的關係



# Thanks!

## Open for any questions

**CJ Wu**

[cjwu@mail.cgu.edu.tw](mailto:cjwu@mail.cgu.edu.tw)