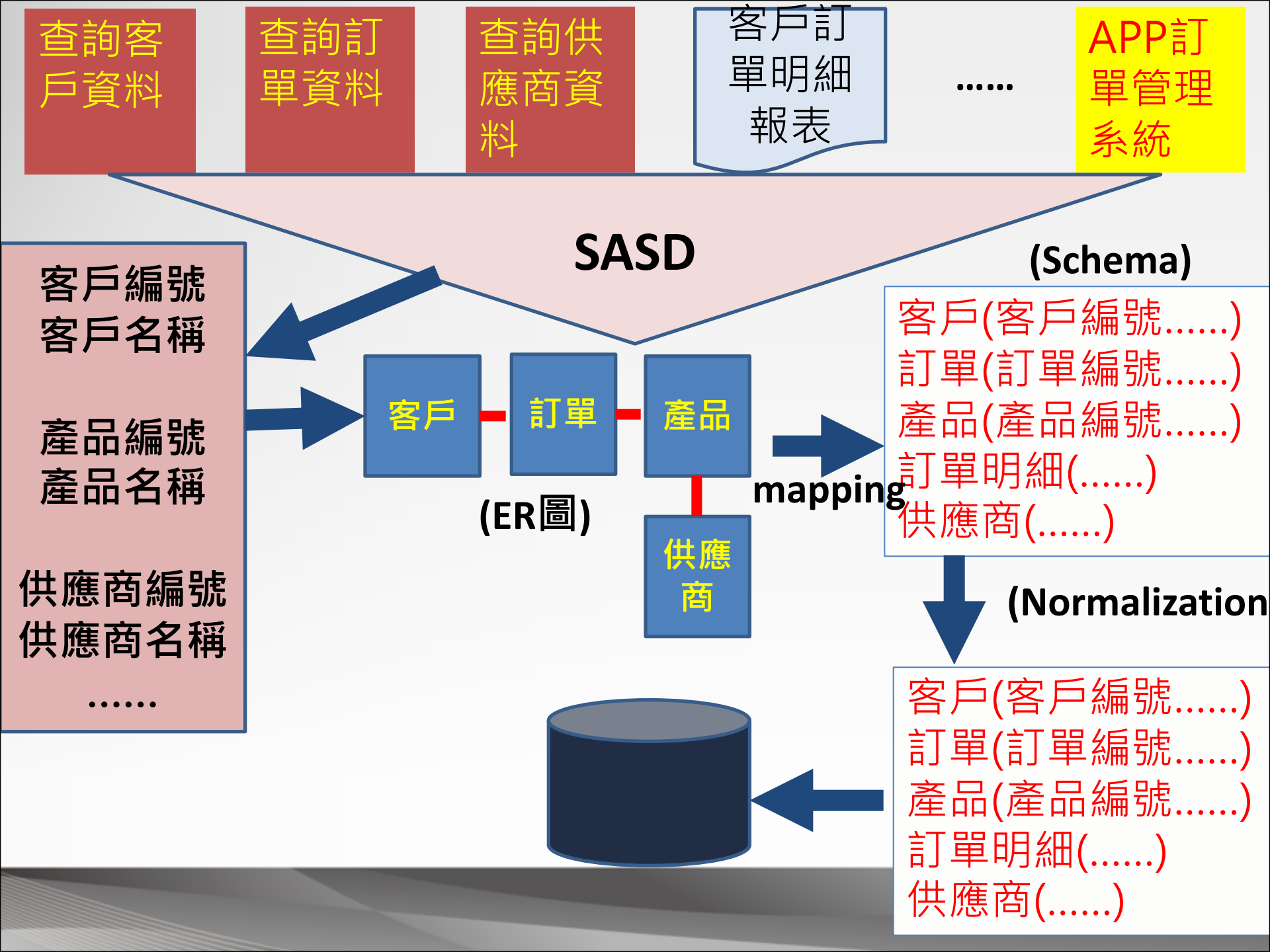

查詢客戶資料
查詢訂單資料
查詢供應商資料
客戶訂單明細報表
.....
APP訂單管理系統
SASD
客戶編號
客戶名稱
產品編號
產品名稱
供應商編號
供應商名稱
.....
客戶
訂單
產品
供應商
(ER圖)
mapping
(Schema)
客戶(客戶編號.....)
訂單(訂單編號.....)
產品(產品編號.....)
訂單明細(.....)
供應商(.....)
(Normalization
客戶(客戶編號.....)
訂單(訂單編號.....)
產品(產品編號.....)
訂單明細(.....)
供應商(.....)

資料庫學習地圖—1/2

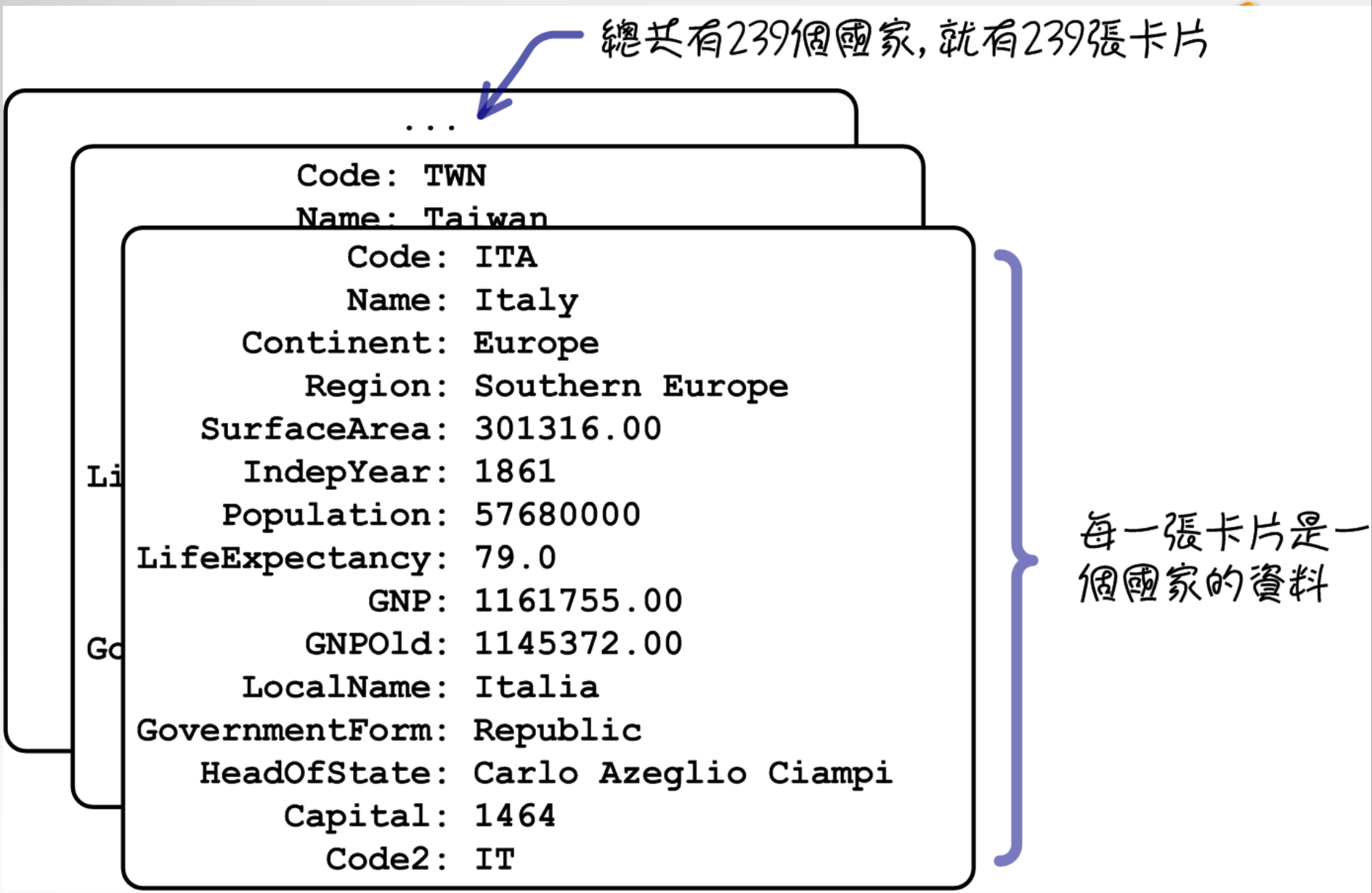
- 資料庫就像蓋房子時的原物料
- 依照藍圖設計每一原料格式
  - 資料需求分析後產生的規格書—ER圖
  - 名稱、資料型態及格式
  - ER mapping、正規化
- 建立原料庫
  - SQL指令：CREATE database, table, index

資料庫學習地圖—2/2

- 蓋房子(建立應用程式)、使用原物料(程式中使用SQL指令：A、I、D、U)
- 原物料庫亂了(資料庫備份、復原)
- 原物料庫自動管理(交易管理)
- 原物料庫安全管理(安全管理)

資料庫的資料儲存

- 儲存與管理資料一直是資訊應用上最基本、也是最常見的技術
- 在還沒有使用電腦來管理你的資料時，你可能會使用這樣的方式來保存世界上所有的國家資料：


總共有239個國家，就有239張卡片
...
Code: TWN
Name: Taiwan
Code: ITA
Name: Italy
Continent: Europe
Region: Southern Europe
SurfaceArea: 301316.00
IndepYear: 1861
Population: 57680000
LifeExpectancy: 79.0
GNP: 1161755.00
GNPOld: 1145372.00
LocalName: Italia
GovernmentForm: Republic
HeadOfState: Carlo Azeglio Ciampi
Capital: 1464
Code2: IT
每一張卡片是一個國家的資料

- 如果你買了一台電腦，電腦中也安裝了一種工作表的軟體
- 國家或是親友通訊錄的資料，可能就會用這樣的方式把它們儲存在電腦裡面：

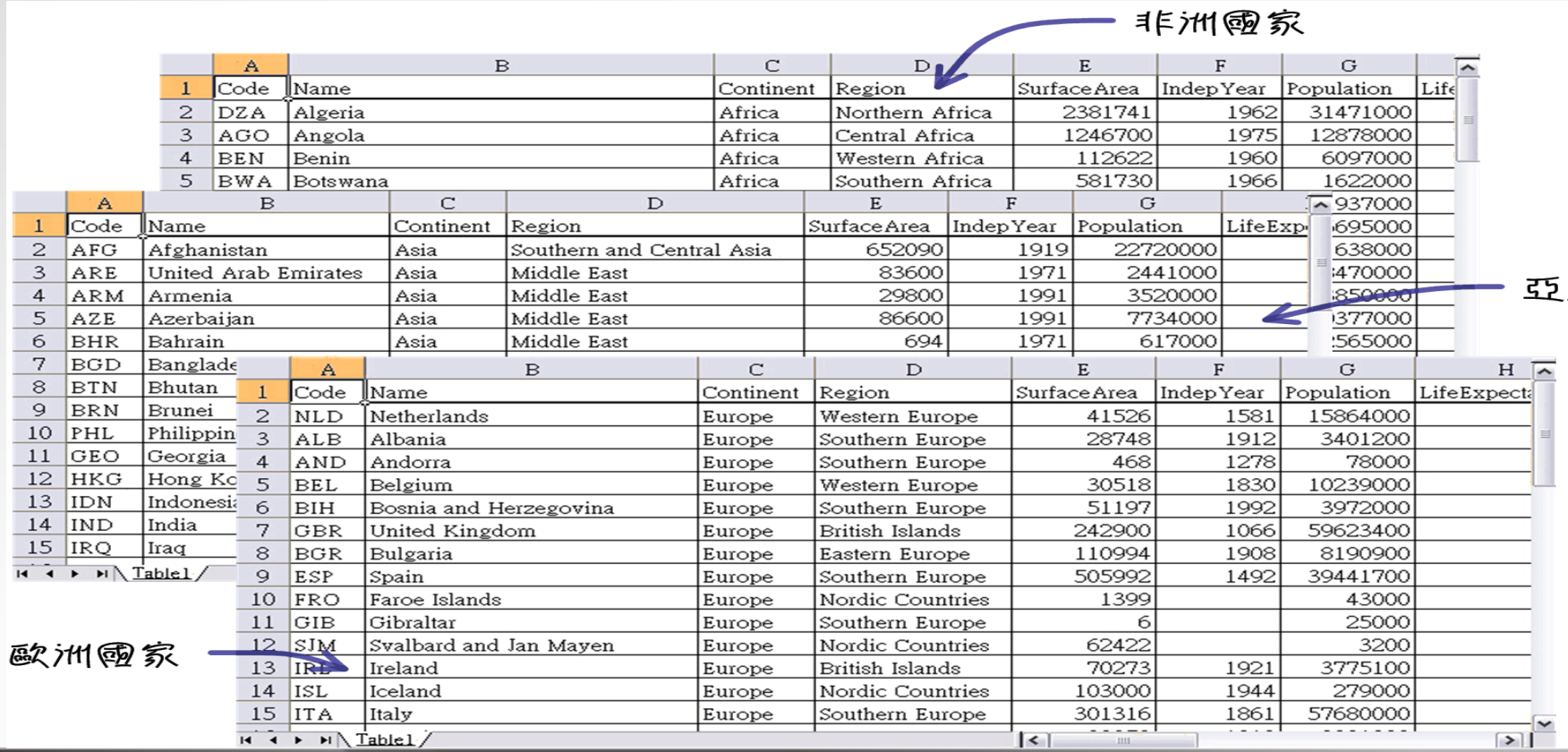
使用類似Excel的軟體來
儲存所有國家的資料...

	A	B	C	D	E	F	G
1	Code	Name	Continent	Region	SurfaceArea	IndepYear	Population
2	AFG	Afghanistan	Asia	Southern and Central Asia	652090	1919	22720000
3	NLD	Netherlands	Europe	Western Europe	41526	1581	15864000
4	ANT	Netherlands Antilles	North America	Caribbean	800		217000
5	ALB	Albania	Europe	Southern Europe	28748	1912	3401200
6	DZA	Algeria	Africa	Northern Africa	2381741	1962	31471000
7	ASM	American Samoa	Oceania	Polynesia	199		68000
8	AND	Andorra	Europe	Southern Europe	468	1278	78000
9	AGO	Angola	Africa	Central Africa	1246700	1975	12878000
10	AIA	Anguilla	North America	Caribbean	96		8000
11	ATG	Antigua and Barbuda	North America	Caribbean	442	1981	68000
12	ARE	United Arab Emirates	Asia	Middle East	83600	1971	2441000
13	ARG	Argentina	South America	South America	2780400	1816	37032000
14	ARM	Armenia	Asia	Middle East	29800	1991	3520000
15	ABW	Aruba	North America	Caribbean	193		103000

Table1

每一列都是一個國家的
資料，這樣應該好多了

- 使用這種工作表來儲存國家資料，當然比用卡片好多了，尤其是想要尋找某個國家的資料，然後修改它的人口數量
- 雖然方便多了，不過在你查詢國家資料時，可能會有這樣的問題：



	A	B	C	D	E	F	G
1	Code	Name	Continent	Region	SurfaceArea	IndepYear	Population
2	DZA	Algeria	Africa	Northern Africa	2381741	1962	31471000
3	AGO	Angola	Africa	Central Africa	1246700	1975	12878000
4	BEN	Benin	Africa	Western Africa	112622	1960	6097000
5	BWA	Botswana	Africa	Southern Africa	581730	1966	1622000

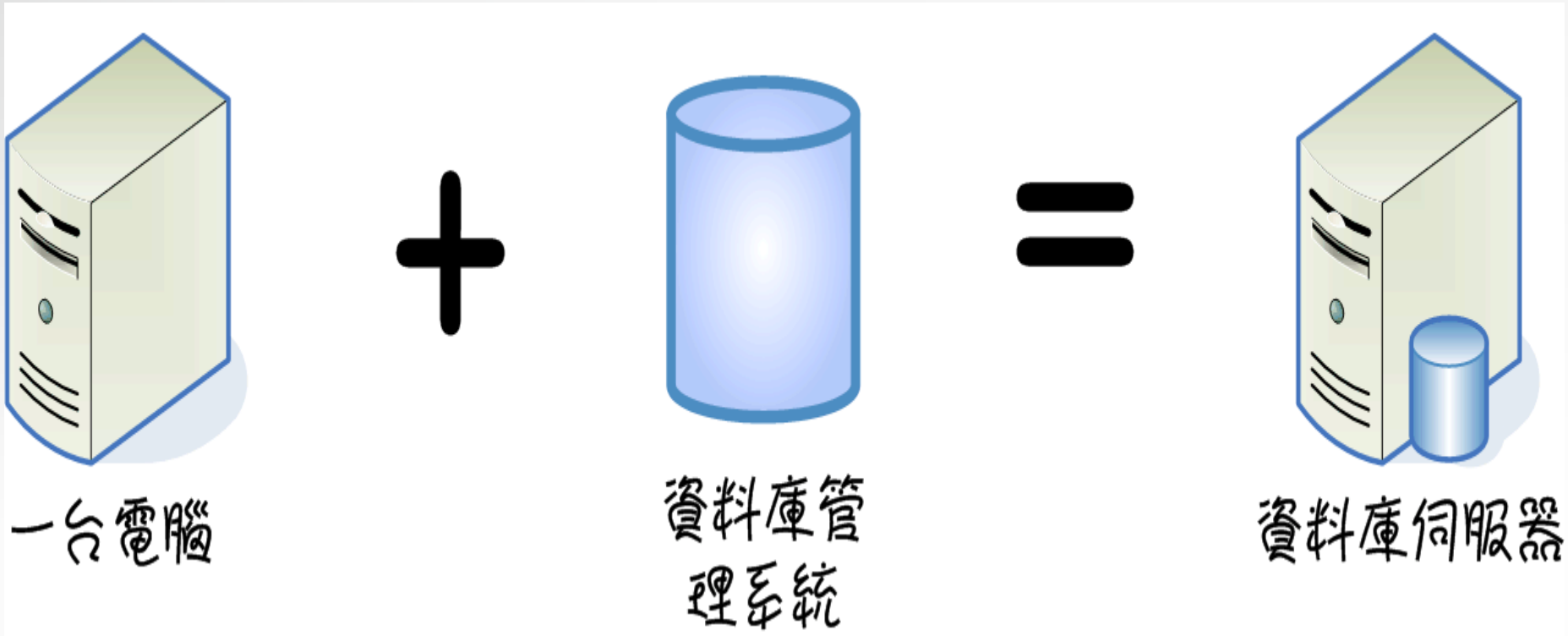
	A	B	C	D	E	F	G
1	Code	Name	Continent	Region	SurfaceArea	IndepYear	Population
2	AFG	Afghanistan	Asia	Southern and Central Asia	652090	1919	22720000
3	ARE	United Arab Emirates	Asia	Middle East	83600	1971	2441000
4	ARM	Armenia	Asia	Middle East	29800	1991	3520000
5	AZE	Azerbaijan	Asia	Middle East	86600	1991	7734000
6	BHR	Bahrain	Asia	Middle East	694	1971	617000

	A	B	C	D	E	F	G	H
1	Code	Name	Continent	Region	SurfaceArea	IndepYear	Population	LifeExpect
2	NLD	Netherlands	Europe	Western Europe	41526	1581	15864000	
3	ALB	Albania	Europe	Southern Europe	28748	1912	3401200	
4	AND	Andorra	Europe	Southern Europe	468	1278	78000	
5	BEL	Belgium	Europe	Western Europe	30518	1830	10239000	
6	BIH	Bosnia and Herzegovina	Europe	Southern Europe	51197	1992	3972000	
7	GBR	United Kingdom	Europe	British Islands	242900	1066	59623400	
8	BGR	Bulgaria	Europe	Eastern Europe	110994	1908	8190900	
9	ESP	Spain	Europe	Southern Europe	505992	1492	39441700	
10	FRO	Faroe Islands	Europe	Nordic Countries	1399		43000	
11	GIB	Gibraltar	Europe	Southern Europe	6		25000	
12	SJM	Svalbard and Jan Mayen	Europe	Nordic Countries	62422		3200	
13	IRL	Ireland	Europe	British Islands	70273	1921	3775100	
14	ISL	Iceland	Europe	Nordic Countries	103000	1944	279000	
15	ITA	Italy	Europe	Southern Europe	301316	1861	57680000	

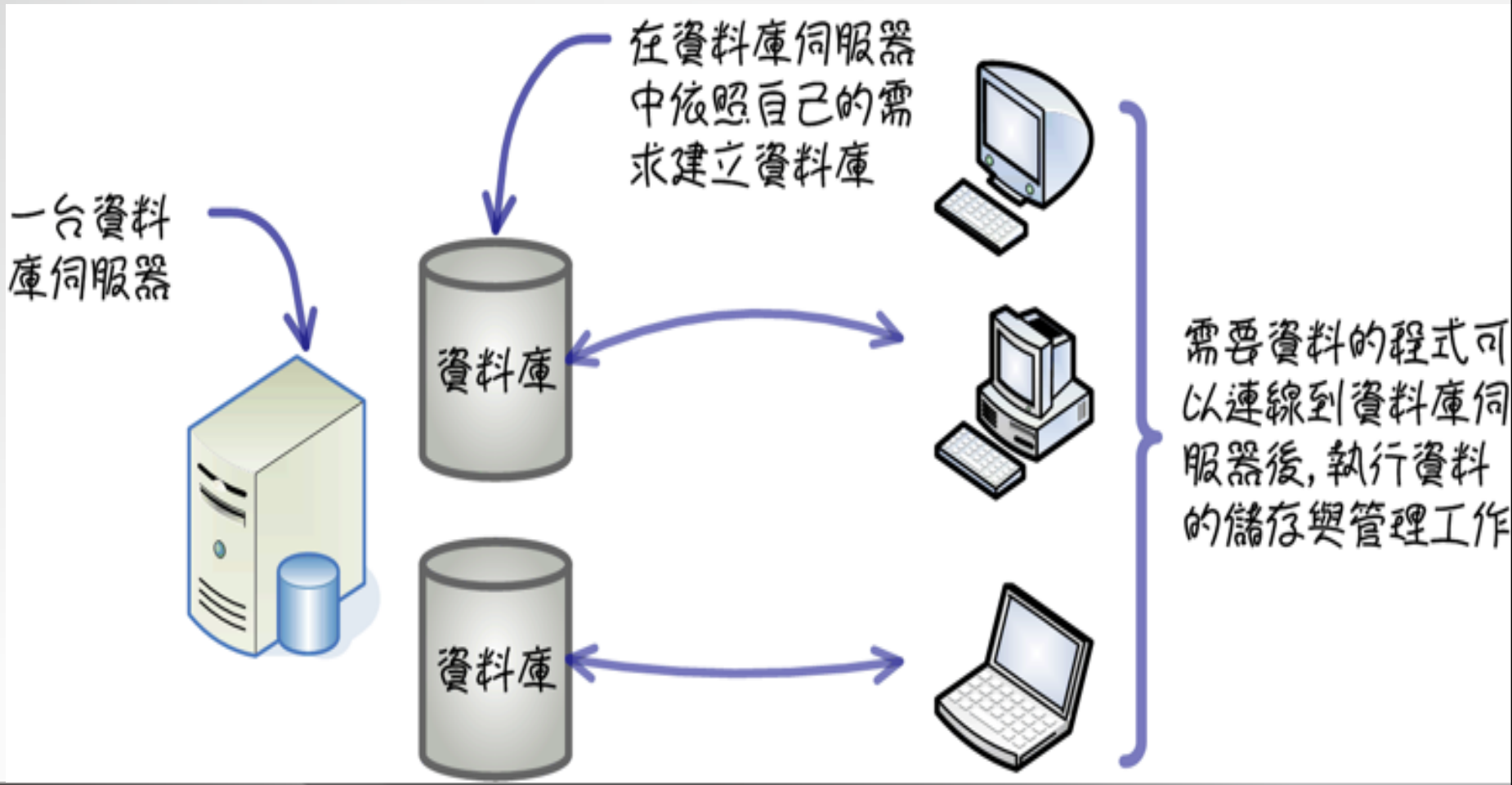
不方便彈性查詢

- 你不太可能把一個洲的國家資料，儲存為一個工作表檔案
- 就算你這麼作了，如果你想要查詢人口數小於十萬的國家時，你會發現這會是一件很困難的工作

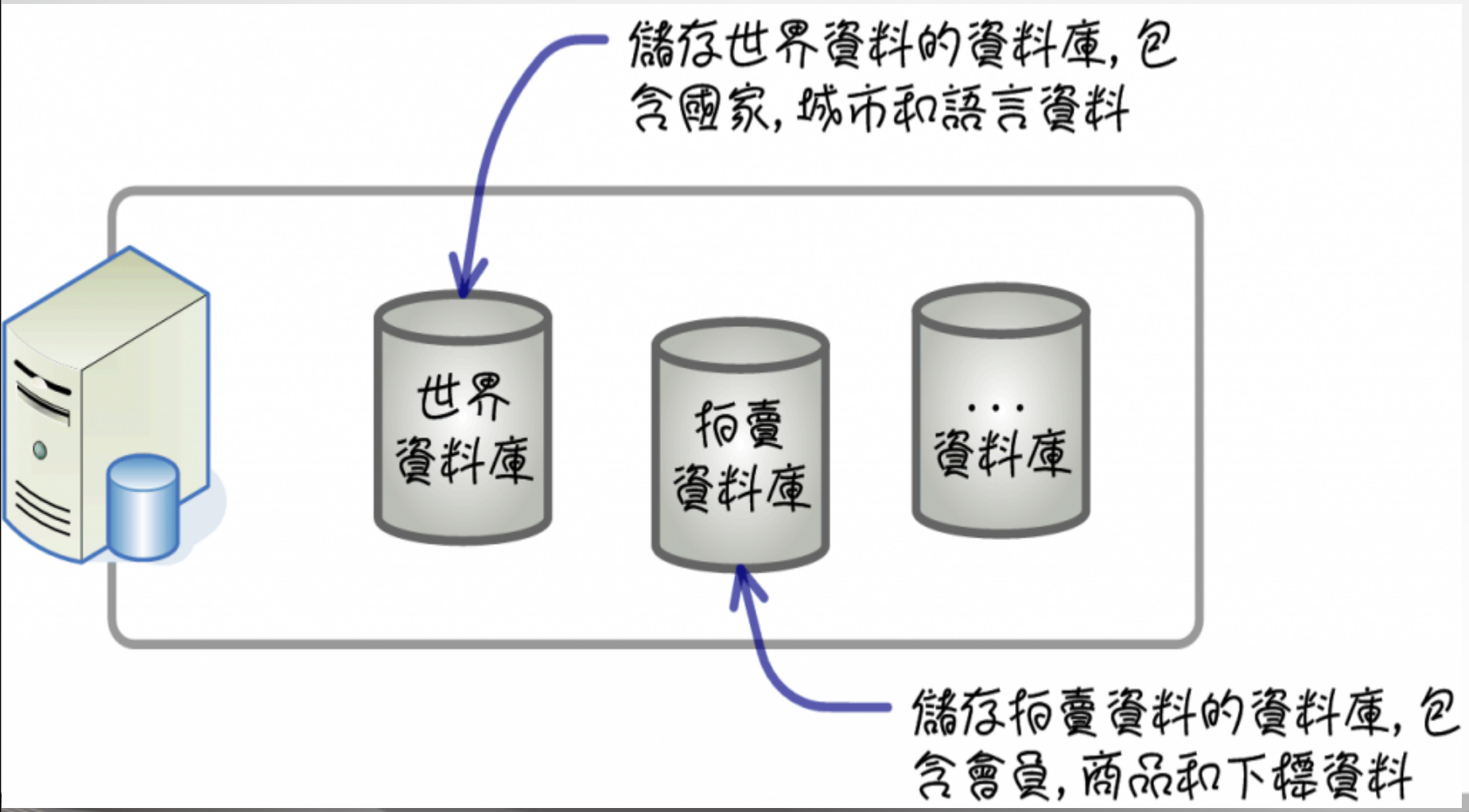
資料庫伺服器



資料庫伺服器

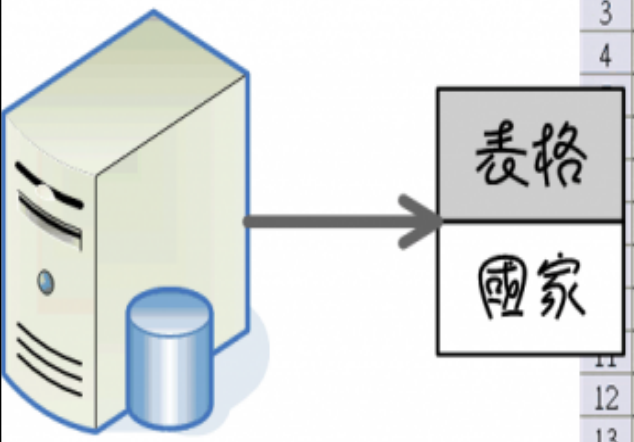

在資料庫伺服器中依照自己的需求建立資料庫
一台資料庫伺服器
資料庫
資料庫
需要資料的程式可以連線到資料庫伺服器後，執行資料的儲存與管理工作

資料庫伺服器


儲存世界資料的資料庫，包含國家，城市和語言資料
世界資料庫
拍賣資料庫
…資料庫
儲存拍賣資料的資料庫，包含會員，商品和下標資料

建立一個儲存國家資料的「表格、table」

為國家資料建立一個表格，使用表格來儲存與管理所有國家的資訊



	A	B	C	D	E	F	G
1	Code	Name	Continent	Region	SurfaceArea	IndepYear	Population
2	AFG	Afghanistan	Asia	Southern and Central Asia	652090	1919	22720000
3	NLD	Netherlands	Europe	Western Europe	41526	1581	15864000
4	ANT	Netherlands Antilles	North America	Caribbean	800		217000
	ALB	Albania	Europe	Southern Europe	28748	1912	3401200
	DZA	Algeria	Africa	Northern Africa	2381741	1962	31471000
	ASM	American Samoa	Oceania	Polynesia	199		68000
	AND	Andorra	Europe	Southern Europe	468	1278	78000
	AGO	Angola	Africa	Central Africa	1246700	1975	12878000
	AIA	Anguilla	North America	Caribbean	96		8000
11	ATG	Antigua and Barbuda	North America	Caribbean	442	1981	68000
12	ARE	United Arab Emirates	Asia	Middle East	83600	1971	2441000
13	ARG	Argentina	South America	South America	2780400	1816	37032000
14	ARM	Armenia	Asia	Middle East	29800	1991	3520000
15	ABW	Aruba	North America	Caribbean	193		103000

Table1

依照不同的需求，查詢需要的國家資料

可以隨時從表格中
查詢需要的資料

表格

國家

非洲國家

Code	Name	Continent	Region	SurfaceArea	IndepYear	Population
DZA	Algeria	Africa	Northern Africa	2381741	1962	31471000
AGO	Angola	Africa	Central Africa	1246700	1975	12878000
BEN	Benin		Western Africa	112622	1960	6097000
BWA	Botswana		Southern Africa	581730	1966	1622000
BFA	Burkina Faso		Western Africa	274000	1960	11937000
BDI	Burundi		Eastern Africa	27834	1962	6695000
DJI	Djibouti	Africa	Eastern Africa	23200	1977	638000

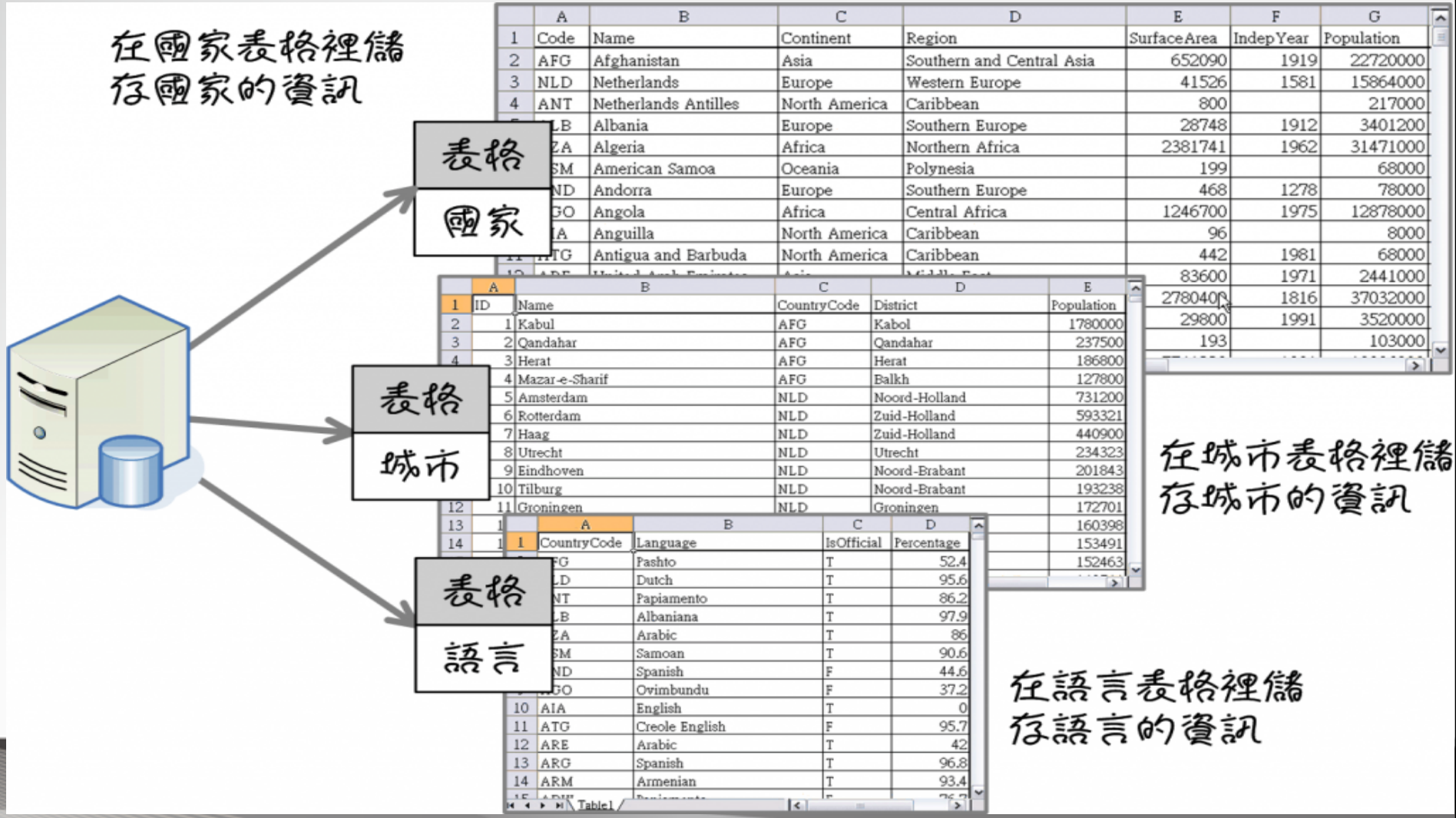
亞洲國家

Code	Name	Continent	Region	SurfaceArea	IndepYear	Population	LifeExp
AFG	Afghanistan	Asia	Southern and Central Asia	652090	1919	22720000	
ARE	United Arab Emirates	Asia	Middle East	83600	1971	2441000	
ARM	Armenia	Asia	Middle East	29800	1991	3520000	
AZE	Azerbaijan	Asia	Middle East				
BHR	Bahrain	Asia	Middle East				
BGD	Bangladesh	Asia	Southern and Central Asia				
BTN	Bhutan	Asia	Southern and Central Asia				
BRN	Brunei						
PHL	Philippi						
GEO	Georgia						
HKG	Hong K						
IDN	Indones						
IND	India						
IRQ	Iraq						

歐洲國家

Code	Name	Continent	Region	SurfaceArea	IndepYear	Population	LifeExpect
NLD	Netherlands	Europe	Western Europe	41526	1581	15864000	
ALB	Albania	Europe	Southern Europe	28748	1912	3401200	
AND	Andorra	Europe	Southern Europe	468	1278	78000	
BEL	Belgium	Europe	Western Europe	30518	1830	10239000	
BIH	Bosnia and Herzegovina	Europe	Southern Europe	51197	1992	3972000	
GBR	United Kingdom		lands	242900	1066	59623400	
BGR	Bulgaria		urope	110994	1908	8190900	
ESP	Spain		Europe	505992	1492	39441700	
FRO	Faroe Islands		Countries	1399		43000	
GIB	Gibraltar	Europe	Southern Europe	6		25000	
SJM	Svalbard and Jan Mayen	Europe	Nordic Countries	62422		3200	
IRL	Ireland	Europe	British Islands	70273	1921	3775100	
ISL	Iceland	Europe	Nordic Countries	103000	1944	279000	
ITA	Italy	Europe	Southern Europe	301316	1861	57680000	

建立儲存城市和語言資料的表格



	A	B	C	D	E	F	G
1	Code	Name	Continent	Region	SurfaceArea	IndepYear	Population
2	AFG	Afghanistan	Asia	Southern and Central Asia	652090	1919	22720000
3	NLD	Netherlands	Europe	Western Europe	41526	1581	15864000
4	ANT	Netherlands Antilles	North America	Caribbean	800		217000
	LB	Albania	Europe	Southern Europe	28748	1912	3401200
	ZA	Algeria	Africa	Northern Africa	2381741	1962	31471000
	SM	American Samoa	Oceania	Polynesia	199		68000
	ND	Andorra	Europe	Southern Europe	468	1278	78000
	GO	Angola	Africa	Central Africa	1246700	1975	12878000
	IA	Anguilla	North America	Caribbean	96		8000
	ATG	Antigua and Barbuda	North America	Caribbean	442	1981	68000
					83600	1971	2441000
					2780400	1816	37032000
					29800	1991	3520000
					193		103000

	A	B	C	D	E
1	ID	Name	CountryCode	District	Population
2	1	Kabul	AFG	Kabol	1780000
3	2	Qandahar	AFG	Qandahar	237500
4	3	Herat	AFG	Herat	186800
5	4	Mazar-e-Sharif	AFG	Balkh	127800
6	5	Amsterdam	NLD	Noord-Holland	731200
7	6	Rotterdam	NLD	Zuid-Holland	593321
8	7	Haag	NLD	Zuid-Holland	440900
9	8	Utrecht	NLD	Utrecht	234323
10	9	Eindhoven	NLD	Noord-Brabant	201843
11	10	Tilburg	NLD	Noord-Brabant	193238
12	11	Groningen	NLD	Groningen	172701
13	1				160398
14	1				153491
					152463

	A	B	C	D
1	CountryCode	Language	IsOfficial	Percentage
	FG	Pashto	T	52.4
	LD	Dutch	T	95.6
	NT	Papiamento	T	86.2
	LB	Albaniana	T	97.9
	ZA	Arabic	T	86
	SM	Samoan	T	90.6
	ND	Spanish	F	44.6
	GO	Ovimbundu	F	37.2
10	AIA	English	T	0
11	ATG	Creole English	F	95.7
12	ARE	Arabic	T	42
13	ARG	Spanish	T	96.8
14	ARM	Armenian	T	93.4

Table1

第1章 資料庫的基礎

- 1-1 資料與資料處理
- 1-2 資料庫
- 1-3 資料管理系統
- 1-4 資料庫系統發展的歷史演進
- 1-5 資料庫技術的發展趨勢

1-1 資料與資料處理

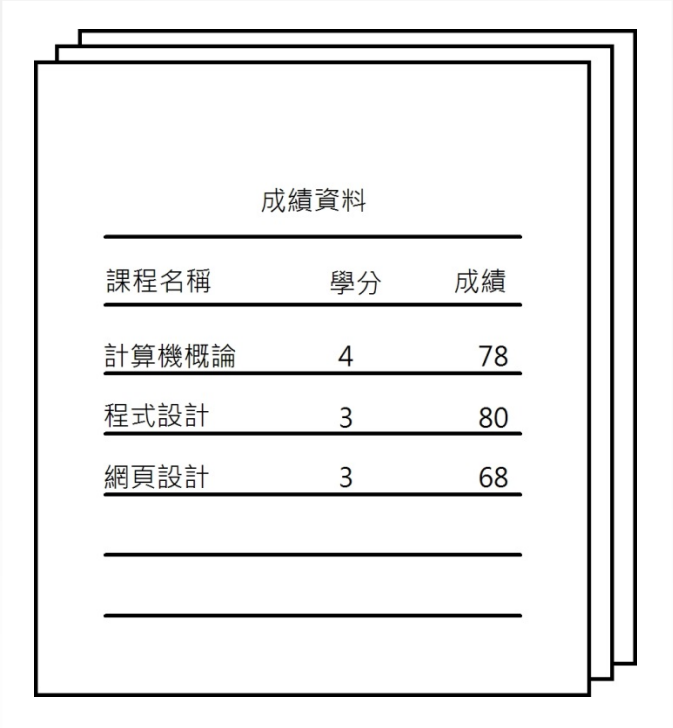
- 1-1-1 資料與資訊的基礎
- 1-1-2 資料處理
- 1-1-3 資料階層

1-1-1 資料與資訊的基礎 – 資料(定義)

- 「資料」（Data）是指收集但沒有經過整理和分析的原始數值、文字或符號，屬於資訊的原始型態
- 「美國國家標準局」（American National Standards Institute; ANSI）定義資料：
  - 資料是使用定義語法或規則所描述的事實、概念或指令，可以適用在人類或程式間進行通訊、解釋和處理

1-1-1 資料與資訊的基礎 – 資料(範例)

- 例如：學校整班學生必修課程的一疊成績資料

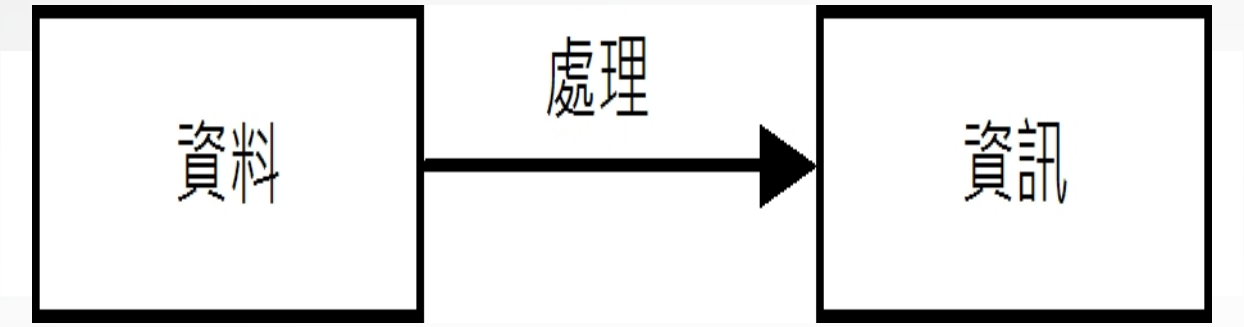


成績資料

課程名稱	學分	成績
計算機概論	4	78
程式設計	3	80
網頁設計	3	68

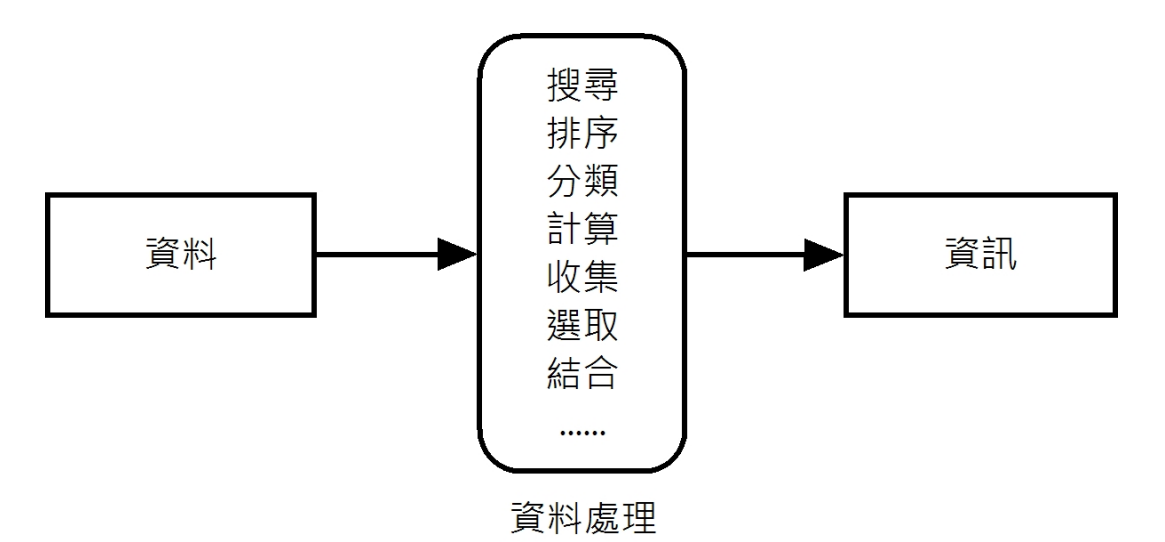
1-1-1 資料與資訊的基礎 – 資訊

- 「資訊」（Information）是經過處理的資料，資料在經過整理和分析後，可以成為有用或可供決策的資訊



1-1-2 資料處理

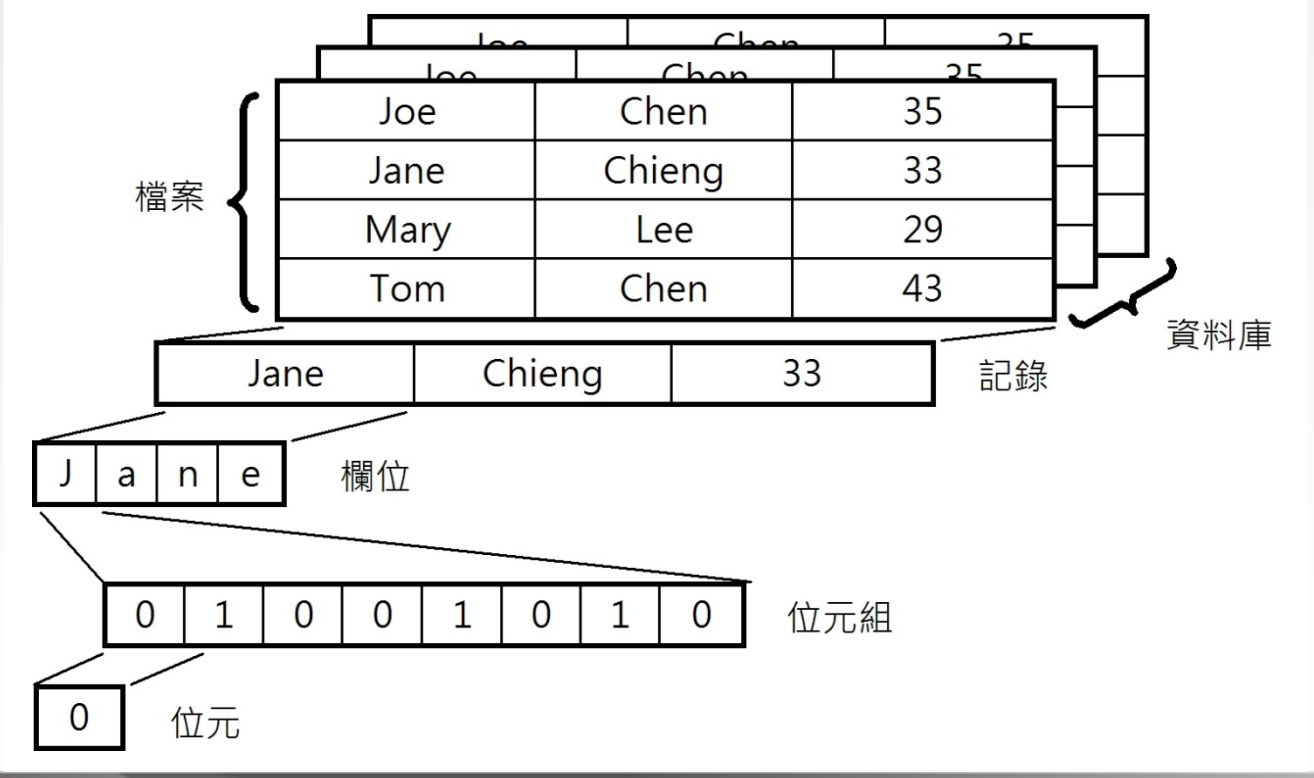
- 「資料處理」（Data Processing）是使用特定方法將資料轉換成資訊的過程
  - 資料可以進行搜尋、排序、分類、計算、收集、選取或結合等操作，以便產生所需的資訊



資料處理

1-1-3 資料階層 – 說明

- 資料階層共分成六個階層：位元、位元組、欄位、記錄、檔案和資料庫

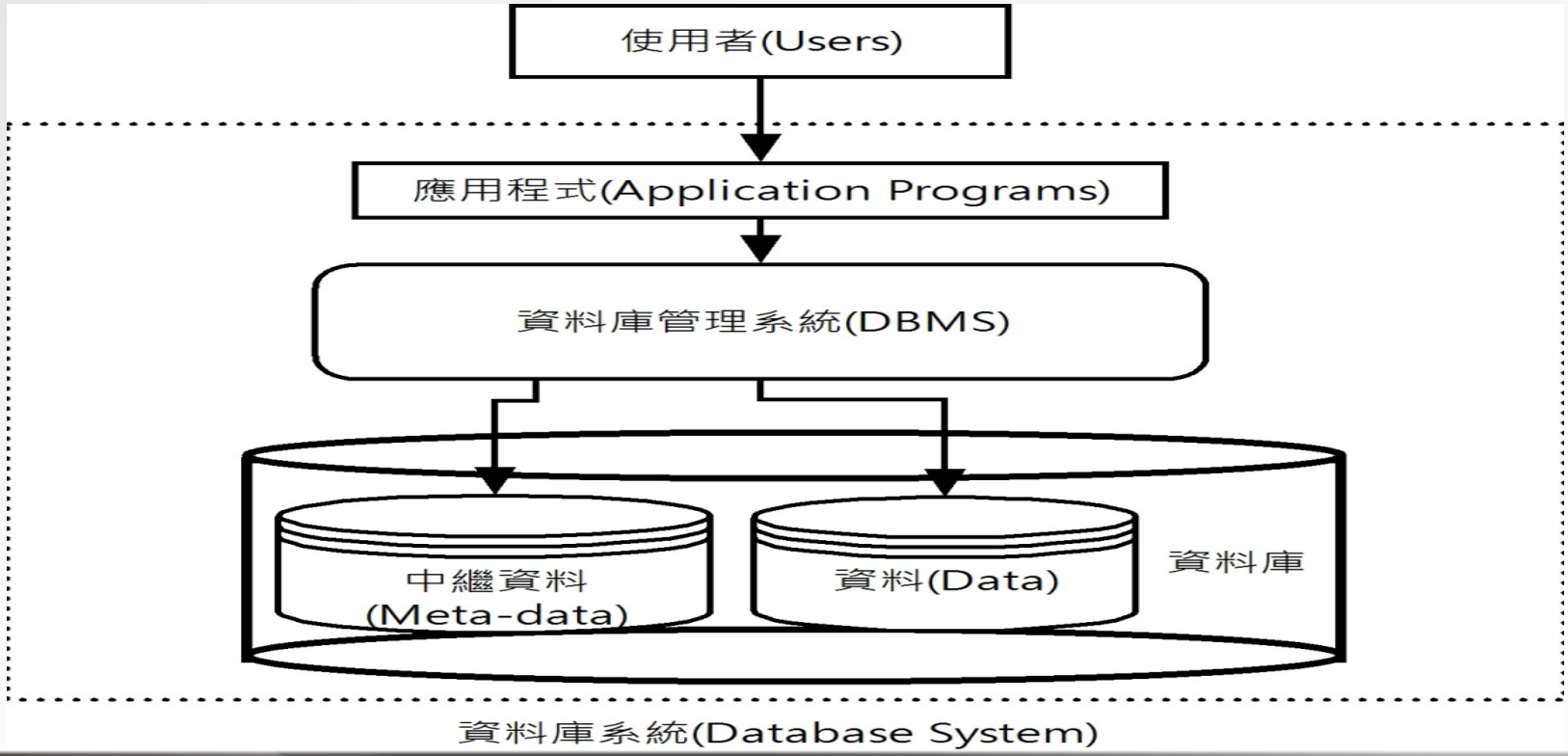


1-2 資料庫

- 1-2-1 資料庫的定義
- 1-2-2 資料塑模
- 1-2-3 資料庫資料的三個層次

1-2 資料庫

- 資料庫系統是由「資料庫」（Database）和「資料庫管理系統」（Database Management System, DBMS）所組成：



1-2-1 資料庫的定義 – 通用定義

- 資料庫比較通用的定義，如下所示：

定義1.1：資料庫（Database）是一個儲存資料的電子文件檔案櫃（An Electronic Filing Cabinet）

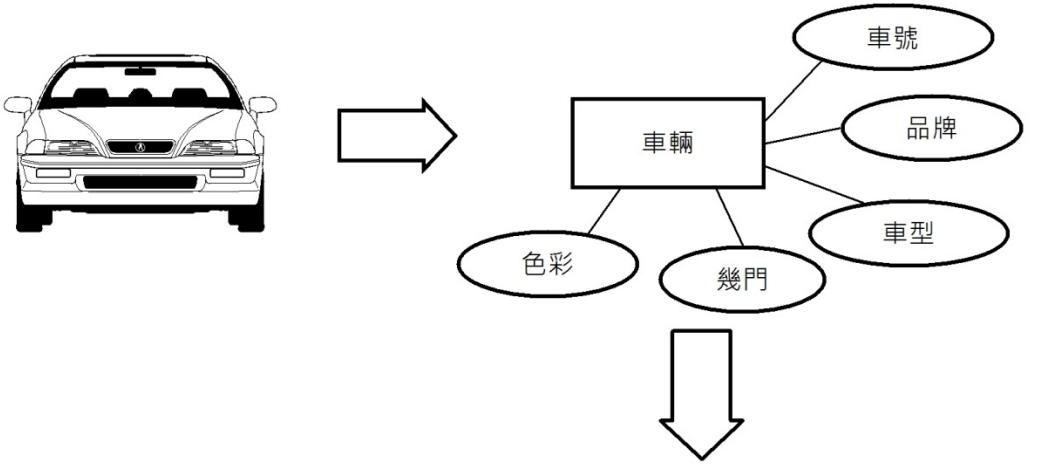
- 上述電子文件檔案櫃是儲存結構化（Structured）、整合的（Integrated）、相關聯（Interrelated）、共享（Shared）和可控制（Controlled）的資料

1-2-1 資料庫的定義 – 長存資料

- 在企業或組織的資料庫中儲存的大量資料，並非是一種短暫儲存的暫時資料，而是一種長時間存在的資料，稱為「長存資料」（Persistent Data）：
  - 在組織中的資料需要一些操作或運算來維護資料
  - 資料是相關聯的
  - 資料不包含輸出資料、暫存資料或任何延伸資訊

1-2-2 資料塑模 – 說明

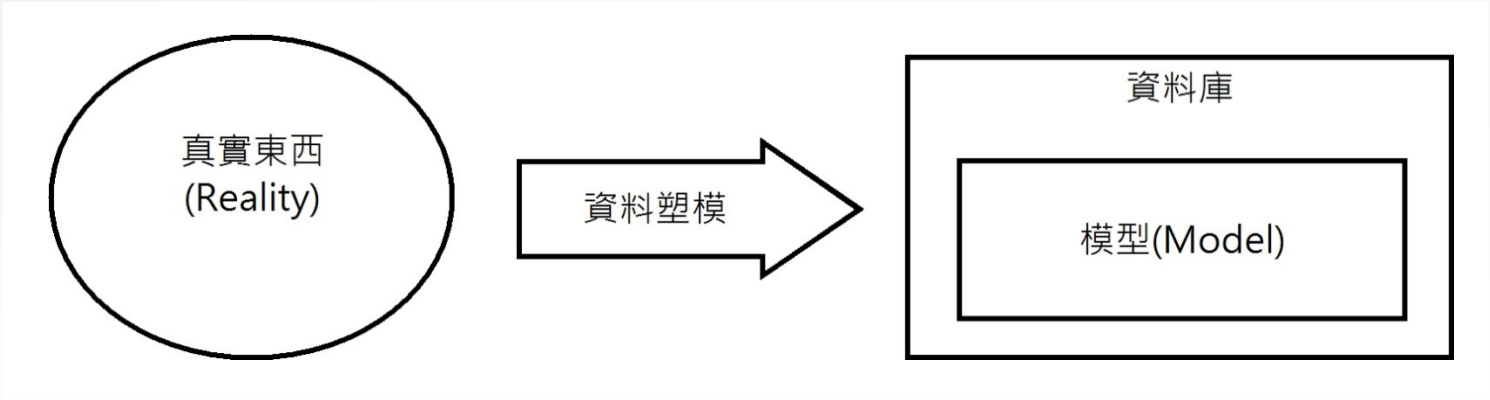
- 在資料庫儲存是結構化收集的「實體」（Entity）資料，實體是現實生活中存在的東西，我們可以將它塑模（Modeling）成資料庫儲存的結構化資料：



車號	品牌	車型	幾門	色彩
DA-1111	裕隆	Sentra	4	紅
AU-2345	福特	Metrostar	4	黑
TU-3456	日產	Civil	4	白

1-2-2 資料塑模 – 基礎

- 「資料塑模」（Data Modeling）是將真實東西轉換成模型，這是一種分析客戶需求的技術
- 其目的是建立客戶所需資訊和商業處理的正確模型，將需求使用圖形方式來表示

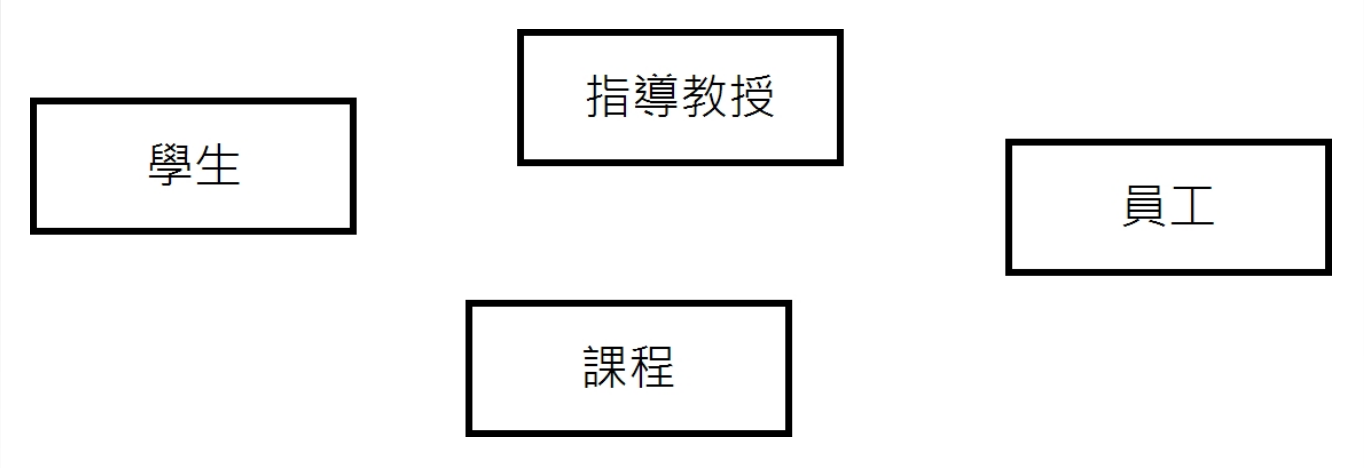


1-2-2 資料塑模 – 邏輯關聯資料

- 資料庫是將真實東西轉換成模型定義的資料結構
- 例如：塑模一間大學或技術學院，也就是從大學或技術學院儲存的資料中識別出：
  - 實體
  - 屬性
  - 關聯性

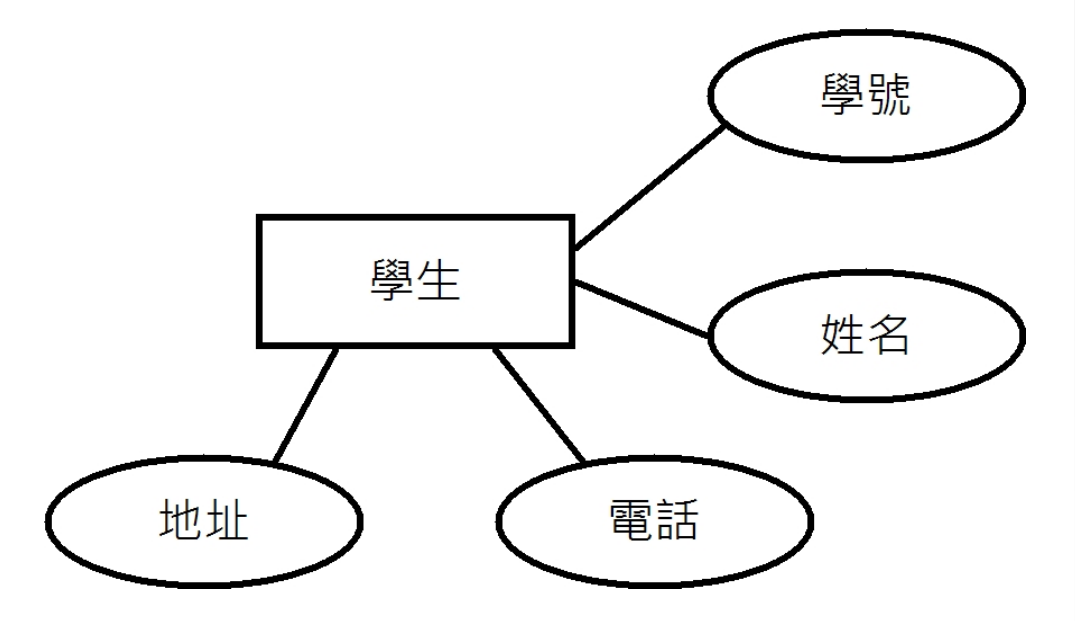
1-2-2 資料塑模 – 邏輯關聯資料(實體)

- 實體（Entities）是在真實世界識別出的東西
- 例如：從大學和技術學院可以識別出學生、指導教授、課程和員工等實體



1-2-2 資料塑模 – 邏輯關聯資料(屬性)

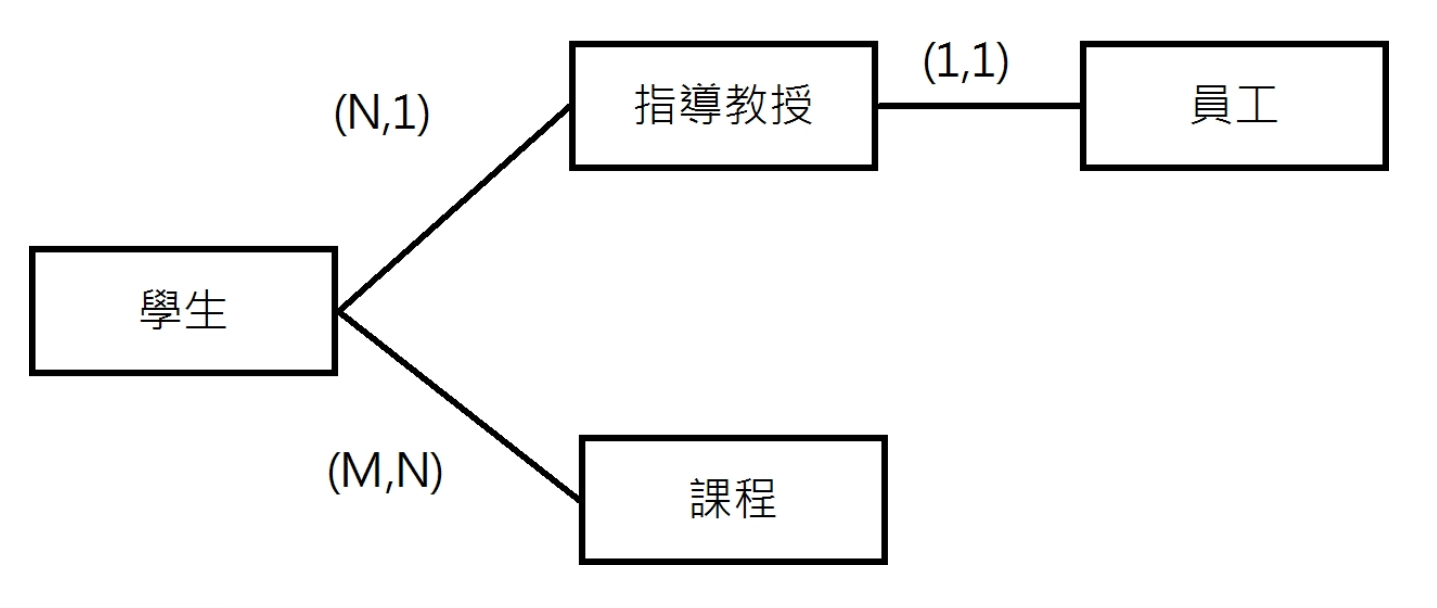
- 屬性（Attributes）是每一個實體擁有的特性
- 例如：學生擁有學號、姓名、地址和電話等屬性



1-2-2 資料塑模 – 邏輯關聯資料(關聯性)

- 關聯性（Relationships）：在二個或多個實體間擁有的關係，主要分為三種，如下所示：
  - 一對一（1:1）：指一個實體只關聯到另一個實體。例如：指導教授是一位學校員工，反過來，此員工就是指這位指導教授
  - 一對多（1:N）：指一個實體關聯到多個實體。例如：學生寫論文時可以找一位指導教授，而一位指導教授可以同時收多位學生
  - 多對多（M:N）：指多個實體關聯到多個其他實體。例如：一位學生可以選修多門課程，反過來，同一門課程可以讓多位不同學生來選修

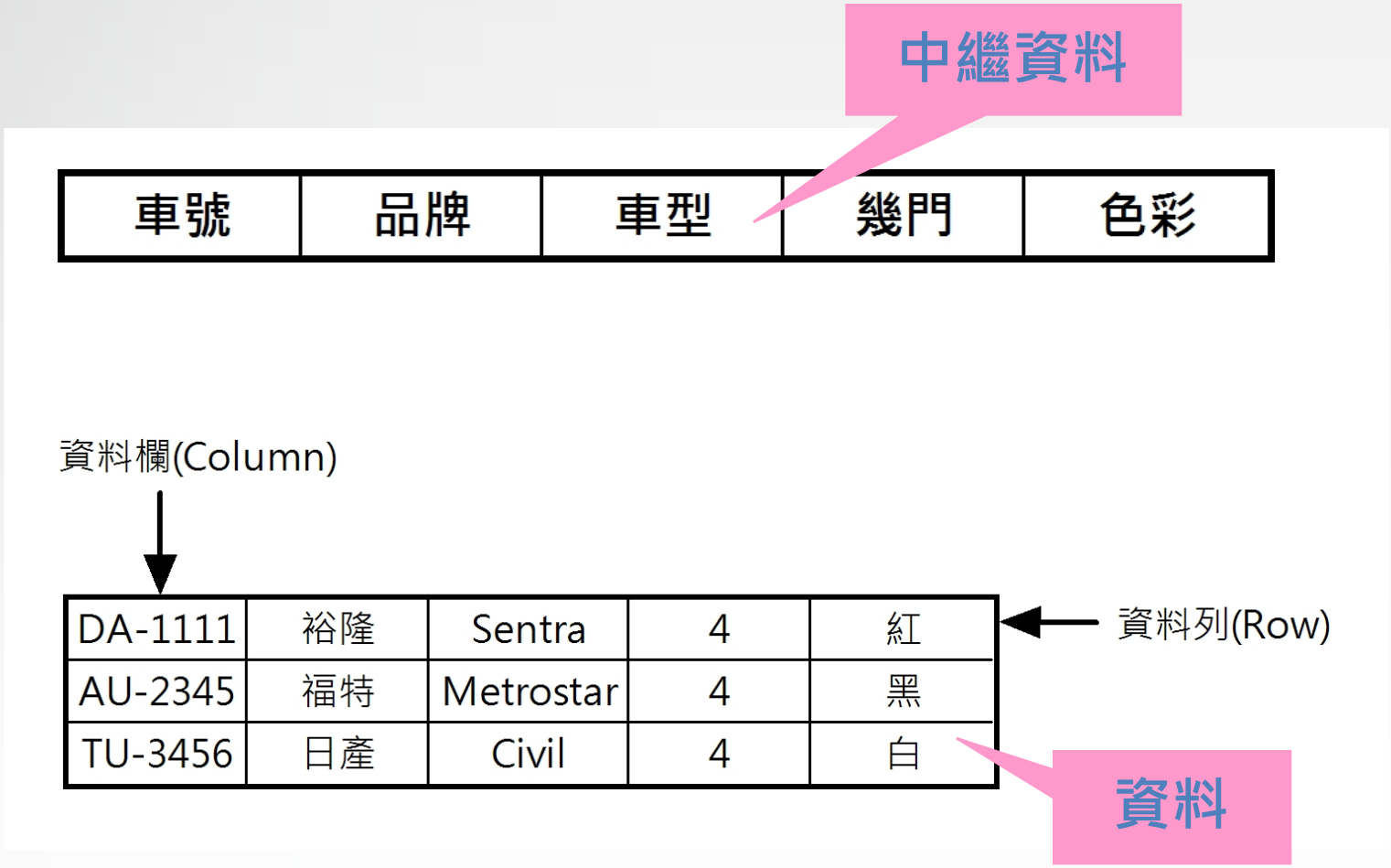
1-2-2 資料塑模 – 邏輯關聯資料(關聯性圖例)



1-2-3 資料庫資料的三個層次 – 說明

- 在資料庫儲存的資料是使用模型找出的實體和屬性所轉換成的資料，可以分成三個層次，如下：
  - 資料模型（Data Model）：將真實東西轉換成資料模型的實體、屬性和關聯性，使用圖形化的高階模型來描述這些資料，通常使用在資料庫設計階段來分析資料庫儲存的資料
  - 中繼資料（Meta-data）：這是用來描述資料庫儲存的是什麼樣的資料，定義資料列（Rows）或記錄（Record）型態，也就是定義各資料欄（Columns）或資料項目（Data Item）的型態
  - 資料（Data）：資料庫實際儲存的資料列（Rows），或稱為記錄（Records）

1-2-3 資料庫資料的三個層次 – 圖例



1-3 資料管理系統

- 1-3-1 資料管理系統的基礎
- 1-3-2 使用檔案處理方式
- 1-3-3 使用資料庫方式

1-3-1 資料管理系統的基礎

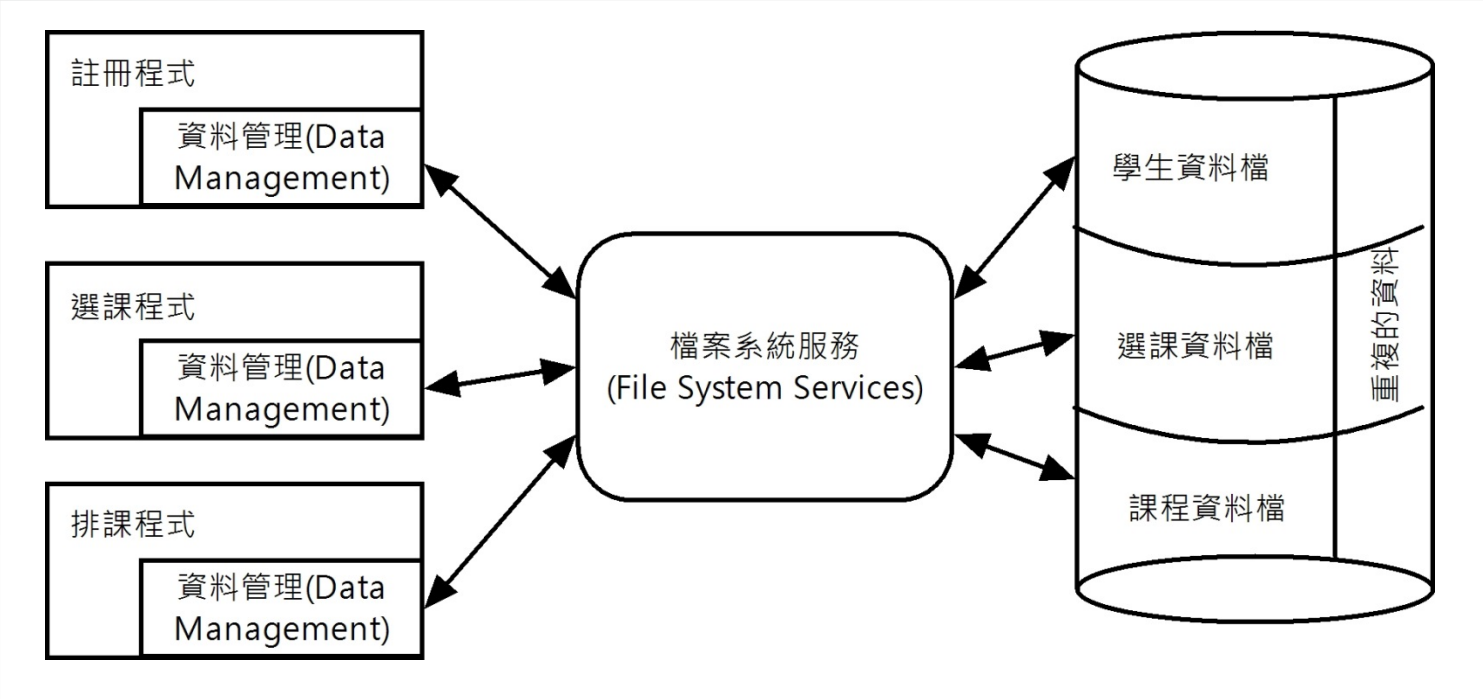
- 檔案處理和資料庫系統都屬於「資料管理系統」（Data Management System）的一環
- 資料管理是在探討組織、存取、更新和保存資料的方法
  - 結構（Structuring）：組織資料建立其資料模型（Data Model）或綱要（Schema），也就是儲存的資料結構
  - 儲存（Storing）：依據建立的資料模型來儲存資料
  - 取出（Retrieving）：取出資料以便進一步執行資料處理
  - 更新（Updating）：更新儲存的資料
  - 保存（Archiving）：長時間保存資料

1-3-2 使用檔案處理方式 – 說明

- 「檔案系統」（File System）是一種非常原始的資料庫。不論是UNIX/Linux作業系統的檔案系統，或Windows作業系統的NTFS檔案系統，資料都是使用相同觀念，以檔案方式來儲存
  - 程式設計師撰寫應用程式來處理檔案儲存的資料，稱為「檔案處理系統」（File Processing System）
- 檔案處理系統是使用檔案處理方式（File Processing Approach）建立的應用程式。早期在資料庫尚未出現的年代，大部分公司都是使用第三代程式語言建立應用程式，例如：COBOL語言，使用作業系統的檔案系統來儲存資料

1-3-2 使用檔案處理方式 – 架構

- 傳統檔案處理系統一樣提供資料庫功能，只是處理的對象是儲存在檔案的資料，如下圖所示：



1-3-2 使用檔案處理方式 – 資料檔案的內容

- 檔案管理系統主要是處理邏輯檔案的資料，在學生註冊、選課和排課系統中的資料檔案是分散儲存在不同部門的電腦檔案，檔案格式可能是文字檔或試算表檔案
- 實際儲存的文字檔案內容是以特殊分隔字元儲存欄位資料，如下所示：

S001:江小魚:中和景平路1000號:02-22222222:1978/2/2
S002:劉得華:桃園市三民路1000號:03-33333333:1982/3/3
S003:郭富成:台中市中港路三段500號:04-44444444:1978/5/5
S004:張學有:高雄市四維路1000號:05-55555555:1979/6/6

1-3-2 使用檔案處理方式 – 問題

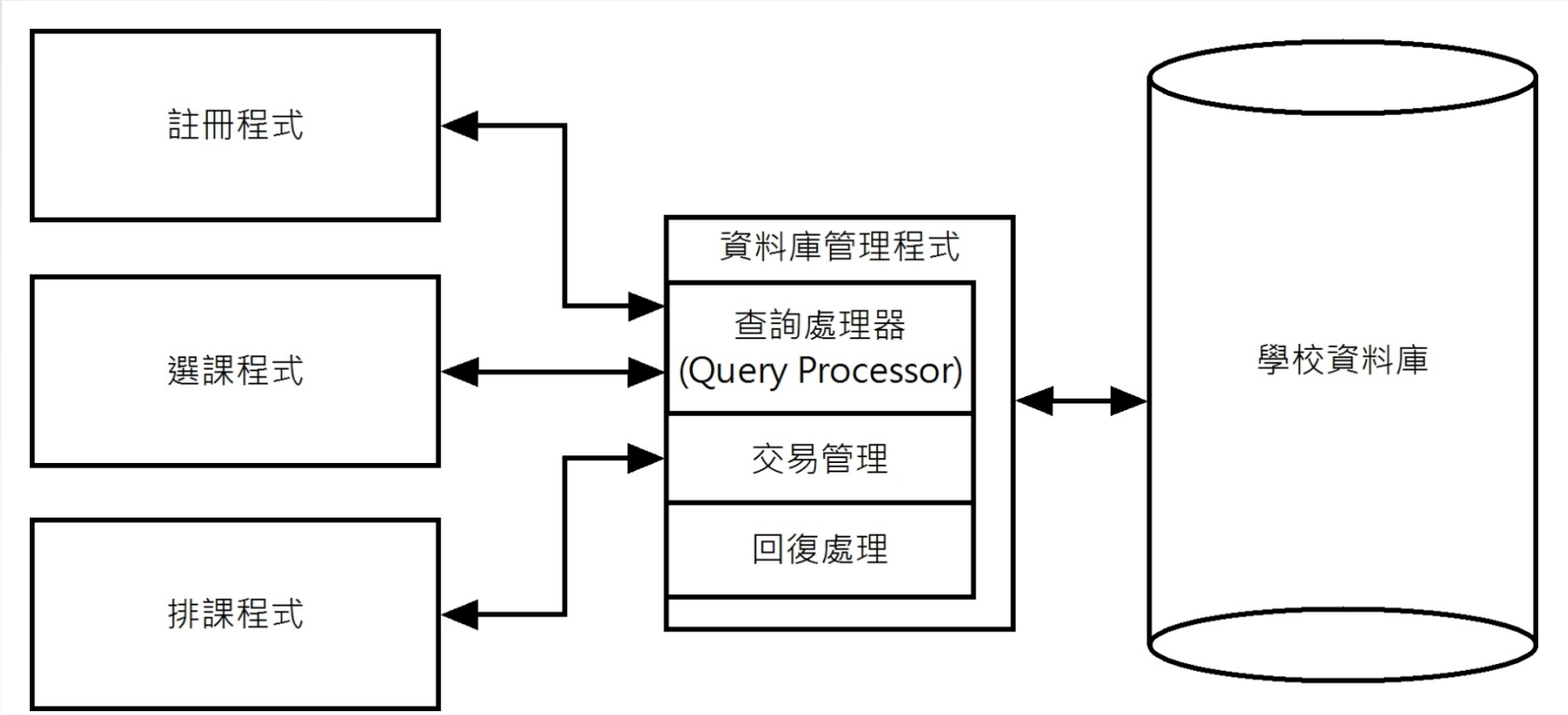
- 結構與資料相關（Structural and Data Dependence）
- 資料分隔與孤立（Data Separation and Isolation）
- 資料沒有集中管理
- 檔案格式不相容
- 更新系統困難
- 資料重複與不一致（Data Redundancy and Inconsistency）
- 多使用者問題（Multiple Users Problems）一個檔案只能單人使用，資料庫可多使用者
- 安全問題（Security Problems）權限
- 資料完整性問題（Data Integrity Problems）

1-3-3 使用資料庫方式 – 說明

- 資料庫方式（Database Approach）建立的資料庫系統可以更有效率的管理和處理資料，解決檔案處理系統產生的問題
- 檔案處理系統和資料庫系統架構的差異，詳細說明請參閱在第2章

1-3-3 使用資料庫方式 – 架構

- 資料庫方式的學生註冊、選課和排課系統



1-3-3 使用資料庫方式 – 目的

- 在資料庫管理系統擁有多種程式模組：查詢處理模組、交易管理和回復處理等，可以進行資料庫的資料管理，將實際資料庫結構和存取都隱藏在資料庫管理系統之後，如此可以達到：
  - 多人使用，資料共享 → 不是不会
  - 資料一致和最少的資料重複
  - 資料獨立（相反的是資料相關）
  - 改進資料完整性問題
  - 更佳的資料安全管理
  - 同步與交易管理
  - 資料備份與回復

1-4 資料庫系統發展的歷史演進

- 1-4-1 1960年代：
  網路與階層式資料庫模型
- 1-4-2 1970年代：
  關聯式資料庫模型與實體關聯模型
- 1-4-3 1980年代：
  關聯式資料庫與SQL結構化查詢語言
- 1-4-4 1990年代：
  物件導向式資料庫模型與主從架構
  - NoSQL, NewSQL

1-4-1 1960年代：網路與階層式資料庫模型

- 1960年代是資料庫系統開始萌芽的年代，隨著磁碟的出現，檔案也從循序索引（Indexed-sequential）的循序存取轉變成「集合導向記錄模型」（Set-oriented Record Model）的直接存取
  - **1961年**：Bachman替GE（General Electric）奇異設計第一套資料庫管理系統IDS（Integrated Data Store），1964年才廣泛的使用，這是一套使用「網路式資料庫模型」（Network Database Model）的資料庫
  - **1965年**：IBM公司開發「IMS」（Information Management System）是使用「階層式資料庫模型」（Hierarchical Database Model）的資料庫

都有索引的影子在

1-4-2 1970年代：關聯式資料庫模型與實體關聯模型 - 1

- 1970年代是資料庫技術快速起飛的年代，資料庫管理系統已經成為大學學科和研究領域，眾多使用網路式和階層式資料庫模型的商用資料庫大量出現在市面。
  - 1970年：IBM研究科學家E. F. Codd博士發表「關聯式資料庫模型」（Relational Database Model）的重要論文
  - 1976年：Peter Chen定義資料庫設計的「實體關聯模型」（Entity-Relationship Model），這是目前資料庫系統分析和設計的基礎 ER Model
  - 1978年：ANSI定義ANSI/SPARC三層資料庫系統架構

*階層、網路式Data base 轉換成關聯式方法：
用ER → 關聯式

1-4-2 1970年代：關聯式資料庫模型與實體關聯模型 - 2

- 1970年代的後期，有二個主要的關聯式資料庫研究計劃開始進行，如下所示：
  - **INGRES**：加州大學柏克萊分校的研究計劃，最後成立Ingres公司，這個研究計劃開發的資料庫系統使用QUEL查詢語言，它是Informix、Sybase和SQL Server資料庫系統的前身。
  - **System R**：IBM公司的研究計劃，最後成為IBM的DB2和Oracle資料庫的前身，使用的SEQUEL查詢語言就是第四篇SQL結構化查詢語言的前身。
- 資料庫查詢語言（Query Language）隨著上述研究計劃，在1970年代開始發展，例如：QUEL、SEQUEL、SQL和QBE查詢語言

1-4-3 1980年代：關聯式資料庫與SQL結構化查詢語言

- 1980年代是商用關聯式資料庫大放異彩的年代，80年代初期已經開發超過100個ANSI/SPARC關聯式資料庫系統，例如：DB2、Oracle、Sybase和Informix等
  - **1980年代中期**：「SQL」（Structure Query Language）成為ISO標準的資料庫查詢語言，IBM DB2也成為IBM公司最重要的資料庫產品
  - **在1980年代後期**：專家系統（Expert Database System）、物件導向資料庫管理系統（Object-Oriented Database Management System）和主從架構分散式系統逐漸成為資料庫系統的未來趨勢

1-4-4 1990年代：物件導向式資料模型與主從架構 - 1

- 1990年代關聯式資料庫的相關技術仍然持續的發展，隨著1990年代中期程式設計技術進入物件導向分析和設計，應用物件導向觀念的資料庫模型也逐漸成形，如下所示：
  - 物件導向式資料庫模型（Object-Orient Database Model）：這是使用物件（Object）觀念代替記錄儲存資料，以繼承減少資料重複，因為程式語言也支援物件導向，所以資料庫與程式語言可以使用一致的資料模型
  - 物件關聯式資料庫模型（Object-Relational Database Model）：這是由Won Kim和Michael Stonebraker博士研究的資料庫模型，將物件導向的觀念整合至關聯式資料庫模型，強調這不是革命（Revolution），而是進化（Evolution）

1-4-4 1990年代：物件導向式資料庫模型與主從架構 - 2

- 隨著Internet與WWW的興起和個人電腦的普及，集中處理的資料庫系統已經改為分散式主從架構（Client/Server）資料庫系統：
  - **客戶端（Client）**：從端的應用程式負責使用者的資料輸入和顯示輸出的結果
  - **伺服端（Server）**：主端的資料庫系統是負責回應從端的請求，將查詢結果傳回從端的應用程式
- 再加上平行資料庫處理（Parallel Database Processing）應用在關聯式資料庫上，可以將表格的資料水平或重直分割成多個資料庫且並行的進行資料處理，即「分散式資料庫系統」（Distributed Database System）

1-5 資料庫技術的發展趨勢

	1960年-1970年中	1970年-1980年中	1980年之後	未來趨勢
資料模型 (Data Model)	網路式(Network)/ 階層式(Hierarchical)	關聯式(Relational)	物件導向式 (Object-oriented)	合併資料模型-物件關聯式(Object-Relational)
資料庫硬體 (Hardware)	大型主機(Mainframes)	大型主機(Mainframes) /迷你主機(Minis)/ 個人電腦(PC)	工作站(Workstation)/ 快速個人電腦(Fast PC)	平行處理(Parallel Processing) /光學儲存媒體
系統架構 (System Architecture)	集中式(Centralized)	集中式(Centralized)	主從架構(Client/ Server)/分散式(Distributed)	異質分散/行動運算
使用介面 (User Interface)	沒有/ 表單(Forms)	查詢語言(Query Language)	圖形使用介面(GUI)	自然語言(Natural Language)/語音輸入
程式介面 (Program Interface)	程序式(Procedure)	內嵌查詢語言 (Embedded Query Langauge)	第四代程式語言(4GL)	整合資料庫和程式語言

補充新興資料庫模式- Big Data下的資料庫

巨量資料儲存和管理資料庫系統

- 平行資料庫
- NoSQL資料管理系統
- NewSQL資料管理系統
- 雲端資料管理

Next

平行資料庫

- 採用關聯式資料模型，支援SQL敘述查詢；採用兩個關鍵技術：資料表水平分割放在不同節點、SQL查詢的分區執行
  - 如欲取得表T中某一數值範圍內資料項，系統首先產生整體執行計畫P，然後將P拆分為n個子計畫$\{P_1,...,P_n\}$，子計畫P_i在節點n_i上獨立執行，最後每個節點將產出結果傳送到指定節點進行聚集產生結果

Return

NoSQL資料管理系統

- 主要指非關聯式、分散式，不提供ACID(單元性、一致性、隔離性、持久性)的資料庫設計模式；沒有固定資料模式並且可以水平擴充的系統統稱NoSQL，即所謂Not Only SQL
  - 採用弱一致性，避免不必要的複雜性
  - 高處理量
  - 高水平擴充和低階硬體叢集
  - 避免昂貴的物件-關係映射：NoSQL系統能儲存資料物件，避免資料庫關係模型和程式物件模型相互轉換的代價

Return

NewSQL資料管理系統

- 不認為傳統資料庫支援ACID和SQL等特性，限制了資料庫的擴充和處理巨量資料的效能；而是其他機制如鎖定機制、紀錄檔機制、緩衝區管理等限制了系統效能，只要最佳化這些技術，關聯式資料庫在處理巨量資料仍能獲得很好的效能
  - 取消耗費資源的緩衝區，在記憶體中執行整個資料庫
  - 放棄鎖定機制，透過容錯機器來實現複製和故障復原，取代原有昂貴的復原機制
  - 高擴充、高性能的SQL資料庫

Return

雲端資料管理

- 雲端資料管理指的是資料庫即服務；使用者不需安裝資料庫管理軟體，只要使用服務提供者提供的資料庫服務
- Amazon提供的關聯式資料庫服務RDS、非關聯式資料庫服務SimpleDB
  - 通透性
  - 可伸縮性
  - 高性價比

Return

巨量資料的處理和分析

- 目前對於巨量資料處理和分析的應用集中在資料倉儲技術、預測分析、即時分析、商業智慧、及資料統計
- SAS、SPSS等軟體受限於單機的運算能力，對巨量資料處理顯得力不從心，因此，Hadoop、MapReduce、R、Python等開放軟體的巨量資料分析工具越來越受青睞

最常用的大數據平台—Hadoop

Hadoop的定義

- Hadoop軟體庫是一個框架，允許在叢集中使用簡單的程式設計模型，對大規模資料集進行分散式運算
  - 它被設計為可以從**單一伺服器擴展到數以千計**的本地運算和儲存的節點
  - 並且Hadoop會在應用層面監測和處理錯誤，而不依靠硬體的高可用性
  - 所以，Hadoop能夠在一個每個節點都有可能出錯的叢集上，提供一個高可用服務

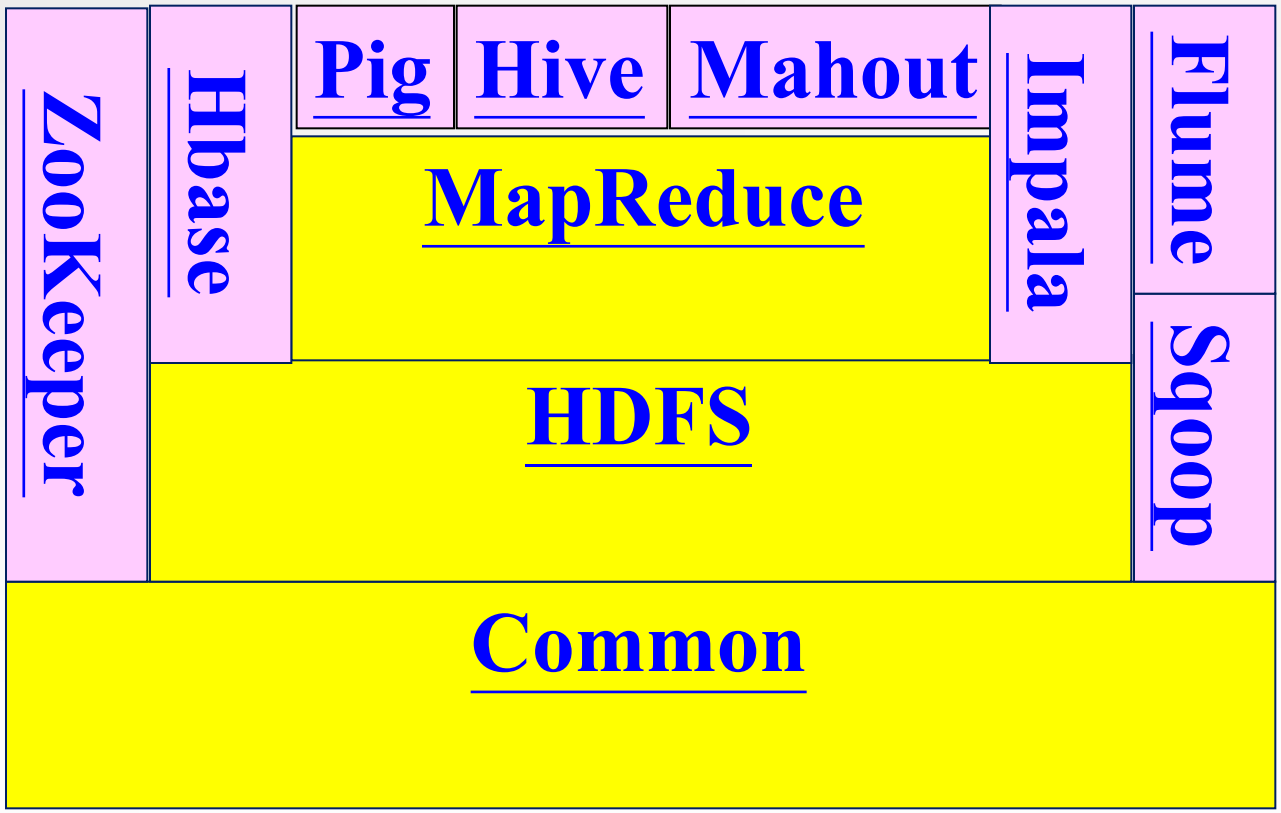
Hadoop發布者Apache軟體基金會

Hadoop特點

- Hadoop是由一系列軟體庫組成的框架
  - Common提供遠端程序呼叫
  - HDFS負責資料分散儲存
  - MapReduce負責資料分散運算
- 適合處理大規模資料：實作分散式儲存和分散式運算
- Hadoop被部署在一個**叢集**上
  - 承載Hadoop是一個實體叢集(一群透過網路互聯的電腦，每一電腦稱節點)

Hadoop生態圈

廣義Hadoop生態圈



Next

Common

- Hadoop體系最底層的模組，為Hadoop各子專案提供各種工具
  - 系統設定工具configuration
  - 遠端程序呼叫RPC
  - 序列化機制
  - 日誌操作

RETURN

HDFS

- HDFS (Hadoop Distributed File System)是Hadoop基石，是一具高容錯性的檔案系統，能將檔案以固定大小(預設64M)的block分散儲存在叢集(cluster)中，並透過備份多份block來確保檔案的可用性

RETURN

MapReduce

- 是一種程式設計模型，利用函數式程式設計的思想，將對資料集處理的過程分為Map及Reduce兩階段，適合進行分散式運算
- Hadoop提供MapReduce的運算框架，使用者可用Java, C++, R, Python, PHP等多種語言實作這種程式設計模型

RETURN

HBase

- HBase為一個分散式、列導向(Column oriented)的開源資料庫；採用BigTable的資料模型—鍵/值儲存
- Hbase擅長大規模資料的隨機、即時讀寫存取

RETURN

ZooKeeper

- ZooKeep作為一個分散式服務框架，解決分散式式系統中一致性問題
  - 設定維護
  - 名稱服務
  - 分散式同步
  - 群組服務

RETURN

Hive

- 由Facebook開發並使用，為基於Hadoop的一個資料倉儲工具，可將結構化的資料檔案對應為一張表，提供簡單的SQL like查詢功能，將SQL語句轉換為MapReduce運行
- 對常見的資料分析需求，不必開發專門的MapReduce作業，降低Hadoop的使用門檻

RETURN

Pig

- Pig和Hive類似，也是對大型資料集進行分析和評估的工具，不同於Hive提供SQL介面，它提供高層、領域導向的抽象語言：Pig Latin，Pig可以將Pig Latin腳本轉化為MapReduce作業
- Pig Latin更靈活但學習成本較高

RETURN

Impala

- 由Cloudera公司開發，可以對儲存在HDFS、Hbase的資料提供直接查詢互動的SQL；對於中等資料的查詢非常迅速，Impala並沒有基於MapReduce的運算框架，大幅領先Hive

RETURN

Mahout

- 是一個機器學習和資料挖掘庫，利用MapReduce程式設計模型實作了K-means, Native Bayes等經典機器學習演算法，具良好擴展性

RETURN

Flume

- 是Cloudera提供的高可用、高可靠、分散式的巨量日誌採集、聚合、傳輸系統

RETURN

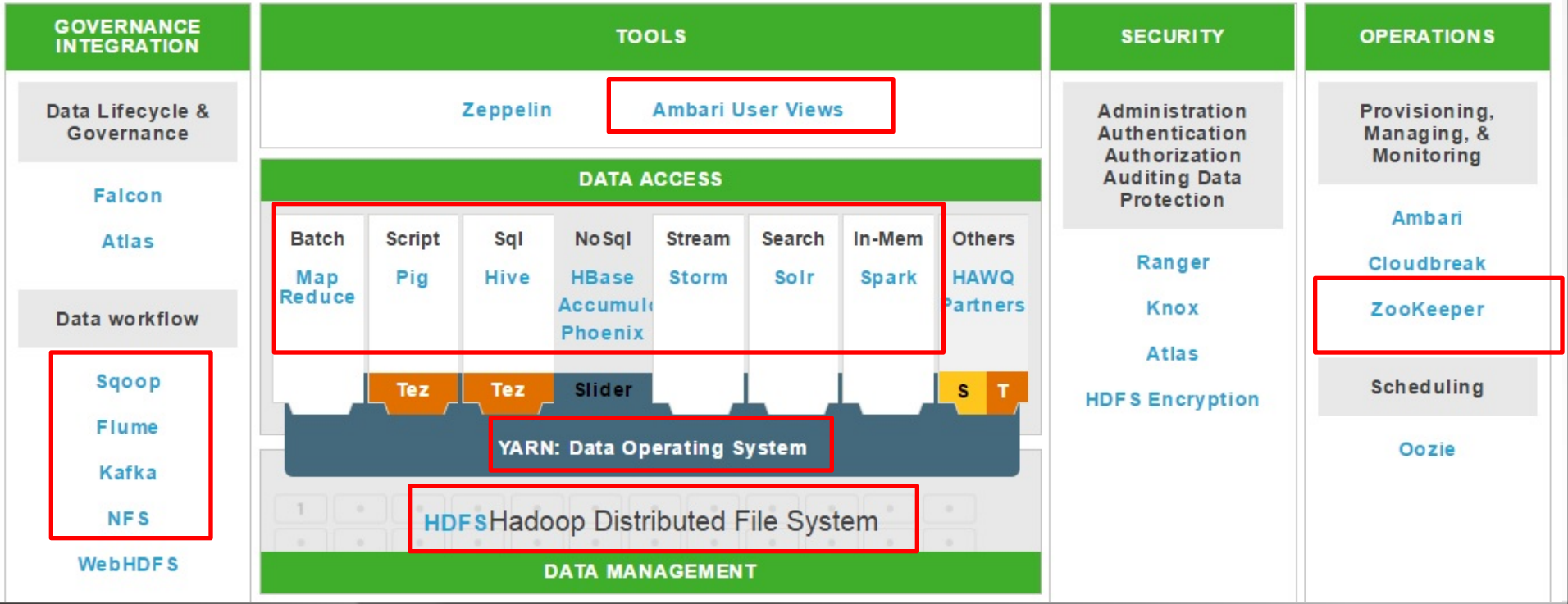
Sqoop

- Sqoop是SQL to Hadoop的縮寫，主要作用在於結構化的資料儲存(資料庫)與Hadoop之間進行雙向交換
  - 即Sqoop可以將關聯式資料庫(如MySql, Oracle)的資料匯入Hadoop的HDFS、Hive中，也可以將HDFS、Hive的資料匯出到關聯式資料庫中

RETURN

Hadoop生態系

- 最著名的兩大巨量資料整合廠商：
  - Cloudera, Hortonworks



BDAS生態系

- Berkeley Data Analytics Stack (BDAS)
- 多數由Apache基金會贊助，皆為開放原始碼
- 大量使用in-memory技術
- Spark元件
  - 結合in-memory技術、DAG網路技術之輕量級巨量資料運算框架

Hadoop和大數據

- Hadoop以一種可靠、高效、可擴展的方式儲存、管理大數據， Hadoop及其生態圈為管理、挖掘大數據提供了一整套相對成熟可靠的解決方案
  - 巨量資料的搖籃—**HDFS**：HDFS的設計理念是以串流式資料存取模式，儲存超大檔案，運行於廉價硬體叢集之上

    列式儲存(Column)—**Hbase**：基於HDFS的列導向分散式資料庫
  - 處理巨量資料的利器— **MapReduce**分散式運算

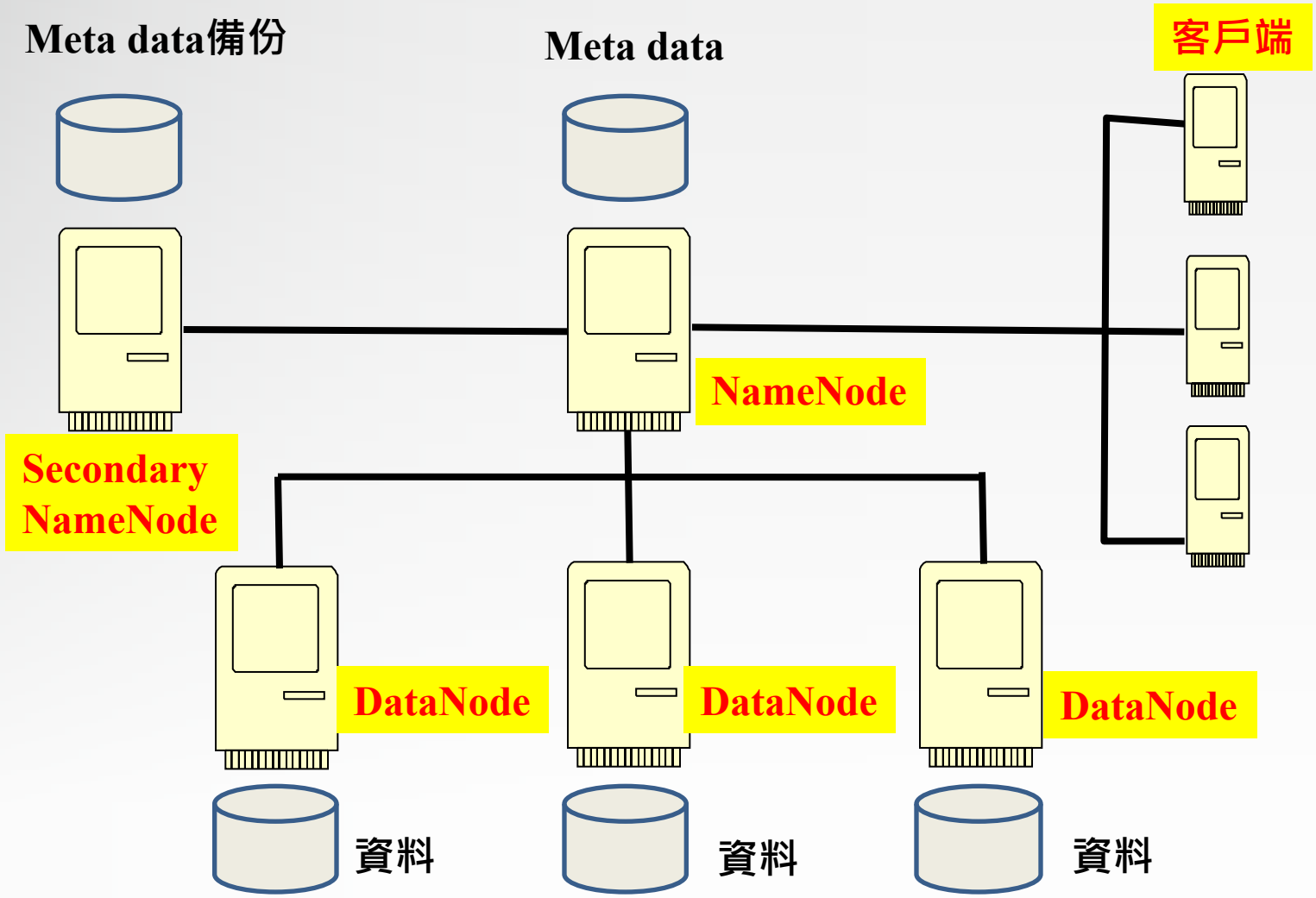
Hadoop架構

- Hadoop主要由兩部分構成，分散式檔案系統HDFS、分散式運算框架MapReduce
  - **HDFS**：當資料集合的大小超過單台電腦的儲存能力時，有必要將其分割(partition)儲存到若干台單獨的電腦上，而管理網路中跨多台電腦儲存的檔案系統，統稱為分散式檔案系統(distributed file system)
  - **MapReduce**：將資料處理過程拆分為主要的Map(對應)與Reduce(化簡)兩步，即使使用者不懂分散式框架內部運行機制，只要能用Map和Reduce的思想描述清楚要處理的問題，即編寫map()、reduce()函數，就能輕鬆實作分散式

Hadoop HDFS架構

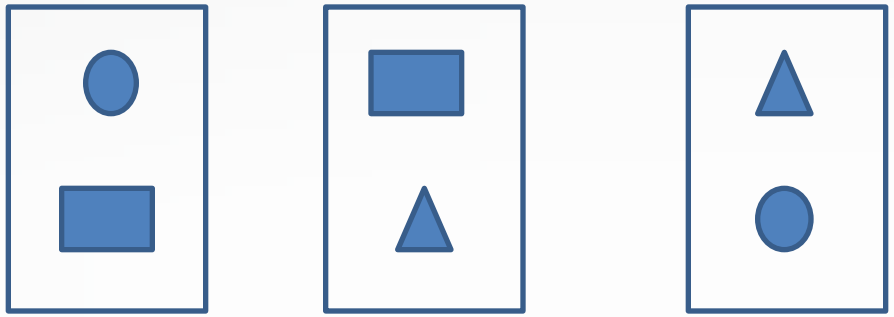
- 構成HDFS叢集的主要是兩類節點，並以主、從架構模式運行
- 一個NameNode(管理者)和多個DataNode(工作者)；另外，Secondary NameNode作為NameNode映像資料備份
  - NameNode：儲存檔案系統的meta data、儲存檔案與資料區塊對應，並提供檔案的全景圖
  - DataNode：儲存區塊資料
  - Secondary NameNode：備分NameNode資料，並負責映像與NameNode日誌資料的合併

Hadoop HDFS架構

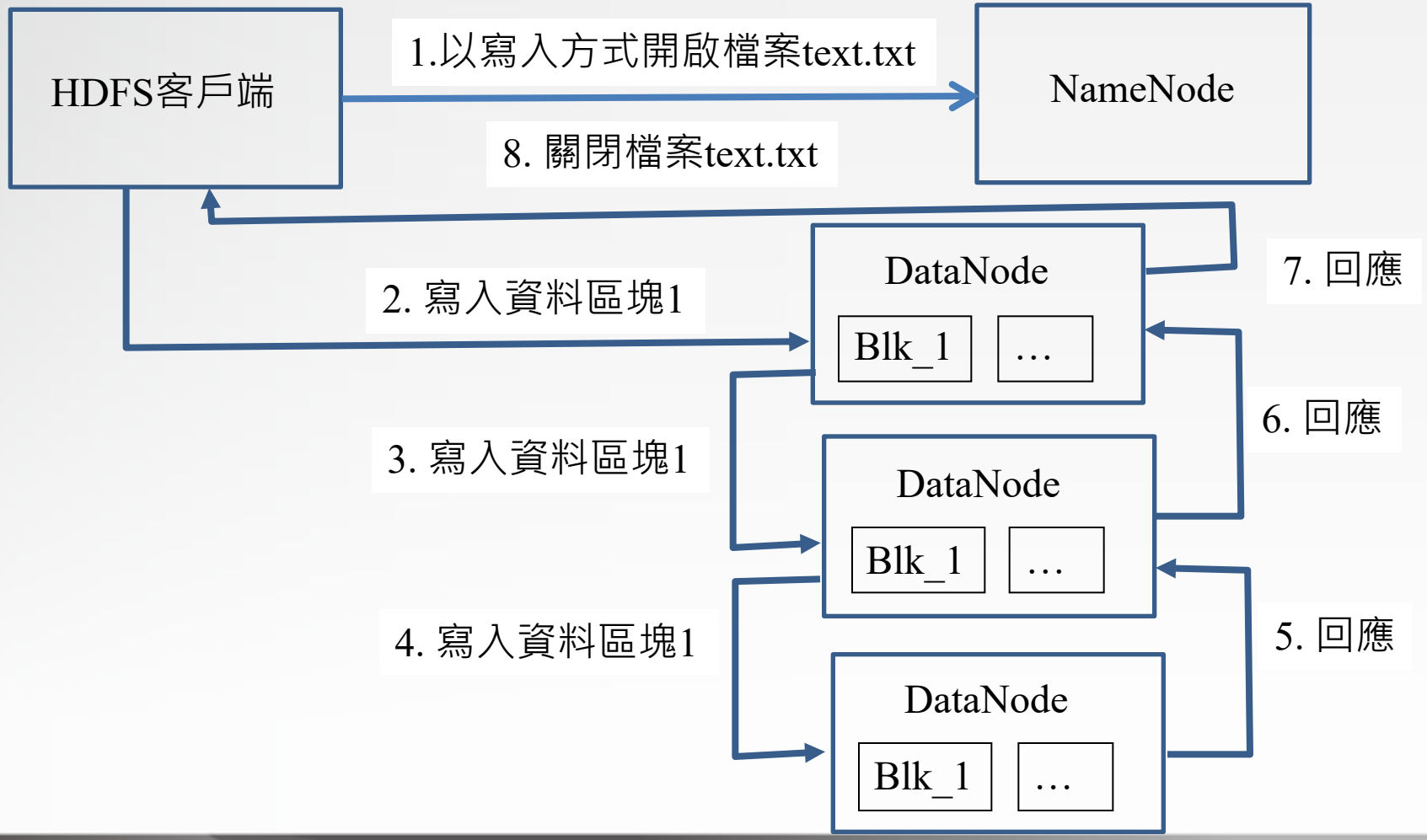


HDFS

- HDFS區塊大小預設為64MB，可以隨需要改變；另外設定有每區塊在Hadoop叢集中存放的份數
  - 假設料檔150MB，存放於三個dataNode：64, 64, 22
  - 若設每區塊存2份於叢集中



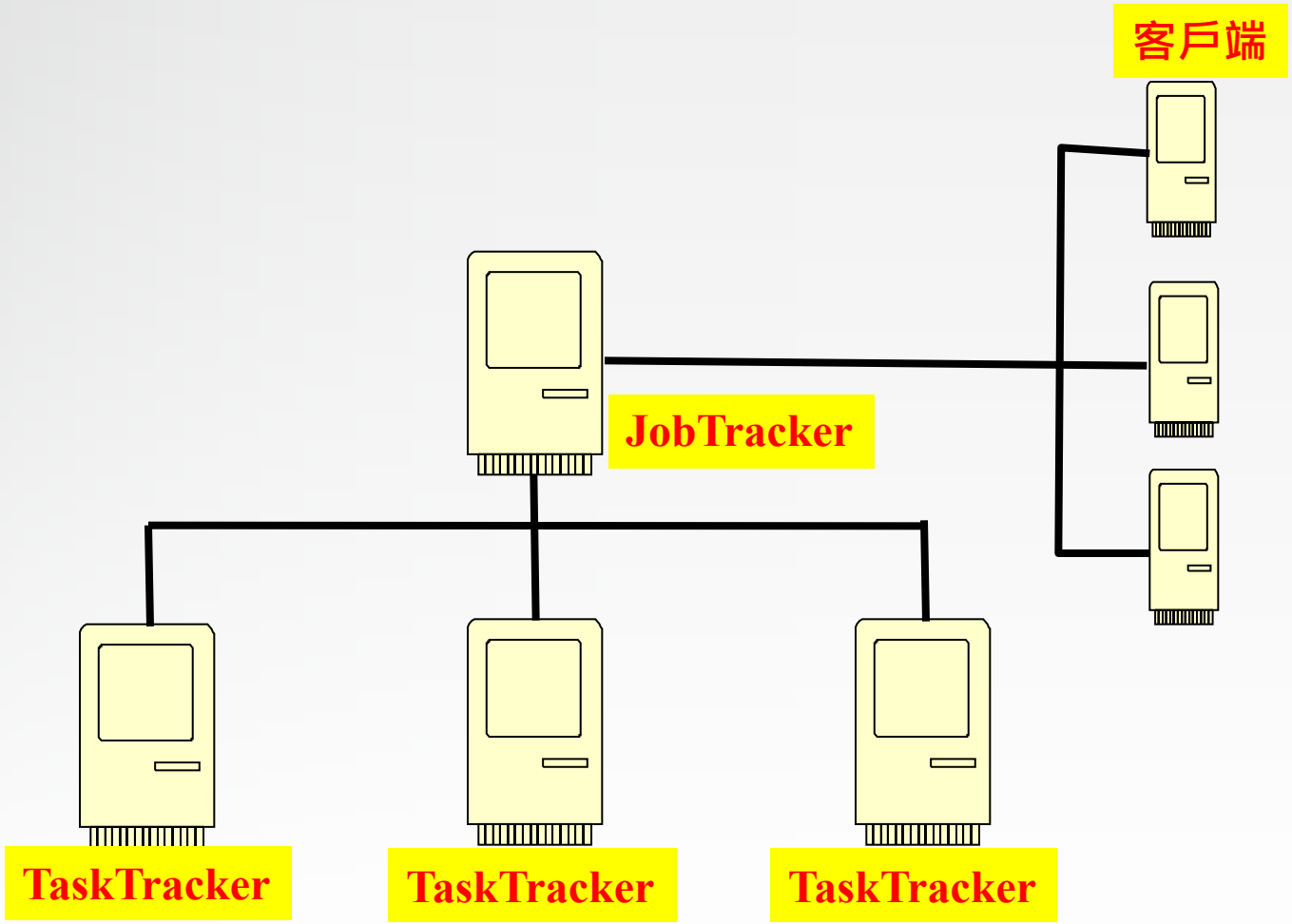
HDFS寫入資料


HDFS客戶端
1.以寫入方式開啟檔案text.txt
NameNode
8. 關閉檔案text.txt
7. 回應
2. 寫入資料區塊1
DataNode
Blk_1
…
6. 回應
3. 寫入資料區塊1
DataNode
Blk_1
…
5. 回應
4. 寫入資料區塊1
DataNode
Blk_1
…

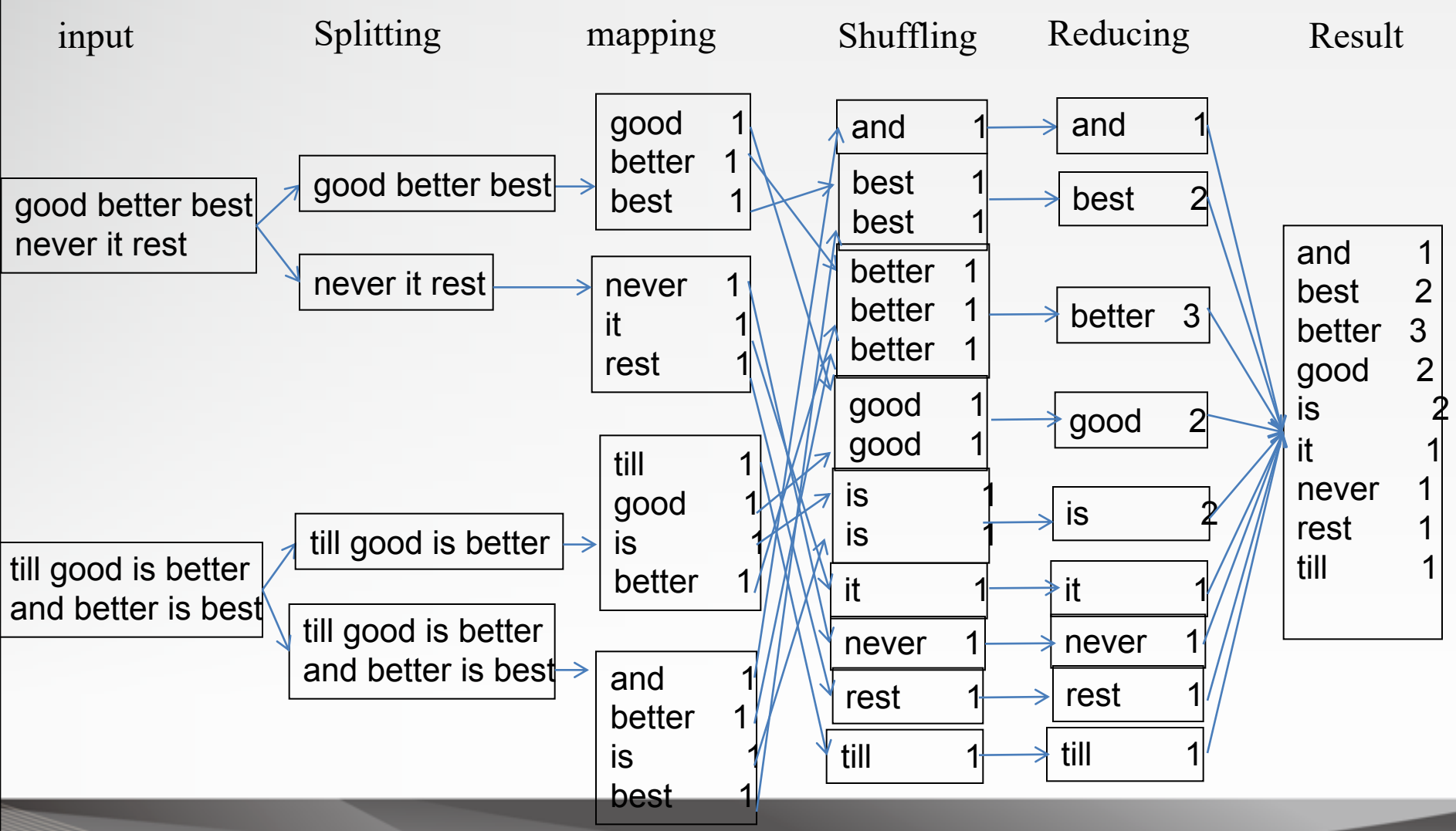
Hadoop MapReduce架構

- 構成MapReduce叢集兩類節點：JobTracker、TaskTracker， MapReduce也是採主、從式架構
  - JobTracker：負責接受客戶端作業提交，調度任務到TaskTracker上運行，並提供監控TaskTracker及任務進度等管理功能
  - TaskTracker：實例化使用者程式，在本地執行任務並周期性地向JobTracker彙報狀態

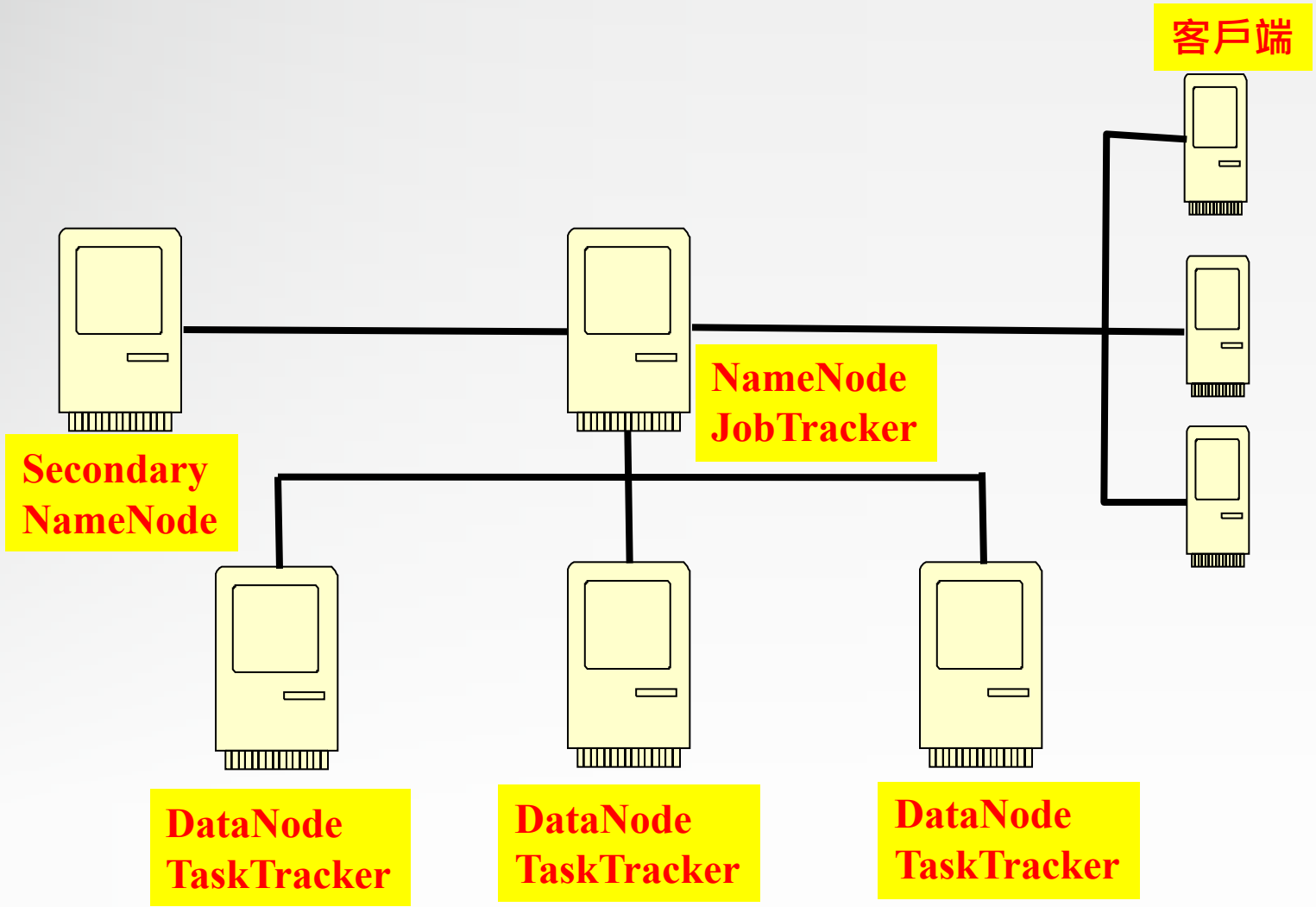
Hadoop MapReduce架構



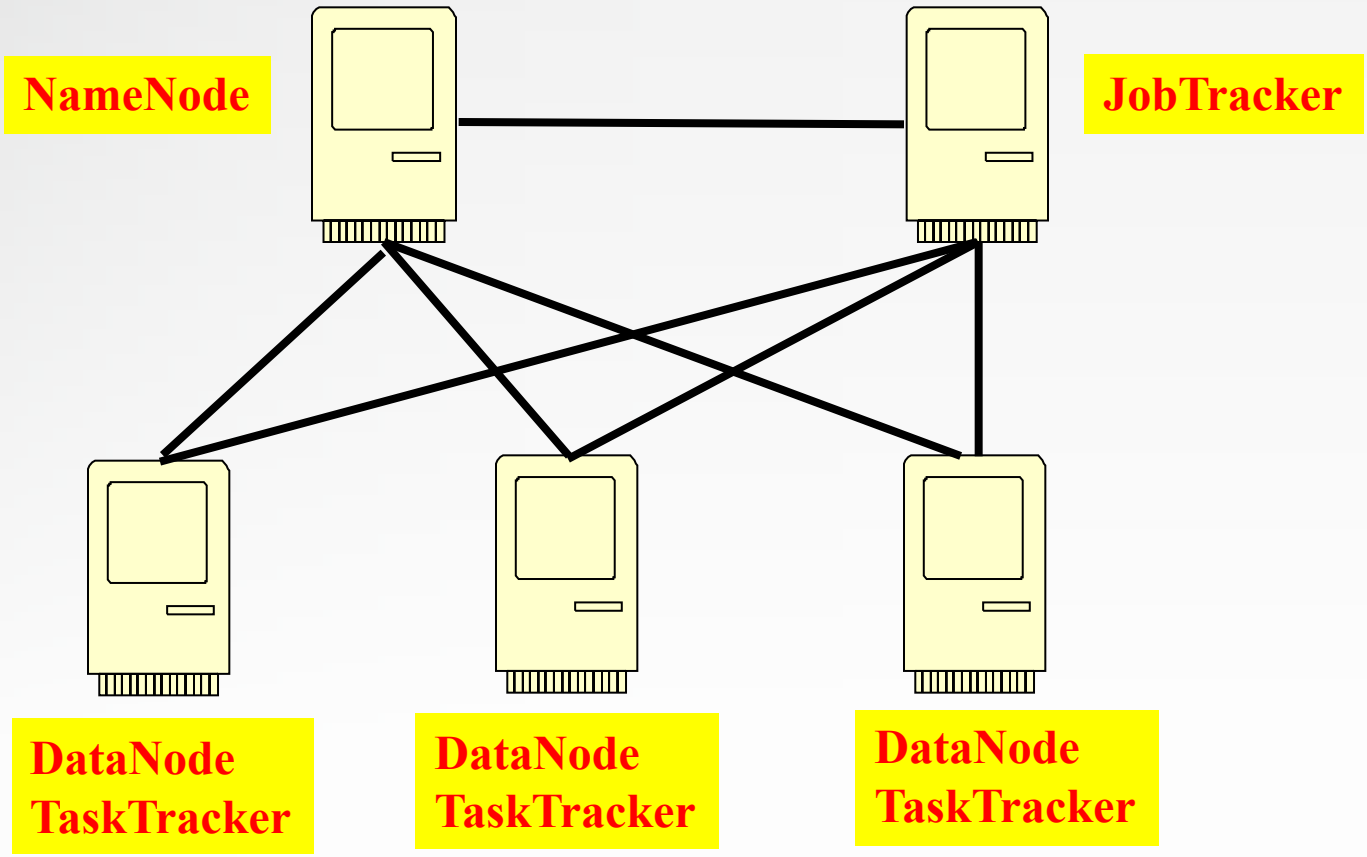
以MapReduce思想完成單詞計算



Hadoop 叢集架構


客戶端
NameNode
JobTracker
Secondary
NameNode
DataNode
TaskTracker
DataNode
TaskTracker
DataNode
TaskTracker

另一種Hadoop 叢集架構




End