

前言

三國演義

三國演義是一部在華人世界非常普及的歷史小說，由羅貫中根據元朝的三國志平話改編，他以東漢末年魏、蜀、吳三國鬥爭為主題，收集歷史資料和說書人的故事，成為這一部大家都非常熟悉的故事。或許我們現在覺得這些歷史已經跟我們沒什麼關係了，不過大家都知道關公過五關斬六將、劉備三顧茅蘆、諸葛孔明的空城記。這些老掉牙的故事，總是不斷的出現在電影、電視劇和各種平台的遊戲，一代又一代的傳承下去。這應該是因為三國演義的確是一個好故事，很多很精采的好故事，就像美國暢銷作家史帝芬金所說的，一個好故事是不會寂寞的。

三國演義的普及，讓大家以為裡面講的故事其實就是真的歷史。羅貫中在編這本書的時候，大概是為了讓它可以比較戲劇化一些，採用了很多當時說書人的內容，這些內容是在民間流傳或由說書人編造的，跟歷史並不一樣。例如大家熟悉的關公斬華雄，在三國演義中是一段非常精采的故事，作者使用很短的內容讓關雲長的豪勇，簡單、清楚而且非常震撼的呈現給讀者。不過根據史料的考證，其實華雄的頭是被孫堅砍掉的。這也是為什麼清朝的時候，就有人評論三國演義是「七實三虛，惑亂觀者」。

提倡白話文的胡適，對三國演義的批評更是激烈，他認為三國演義把諸葛亮的足智多謀寫成一個呼風喚雨的妖道。張飛在歷史上其實是一個很有君子風度的武將，可是卻被寫成粗魯的莽夫。¹雖然有很多精采的故事，可是沒有經過更好的整理，所以三國演義在華人古典文學上的地位，一直不如紅樓夢，甚至連水滸傳都比不上。²

¹ 比胡適更早的有魯迅在「中國小說史略」說三國演義是「顯劉備的長厚而似偽，狀諸葛之多智而近妖」。

² 紅樓夢在文學上的重視讓它演變成一門「紅學」，可是紅樓夢的故事與人物對一般人來說，卻不如三國演義來得熟悉。

MySQL 與 SQL

MySQL 在資訊應用的角色，好像跟三國演義這本著作有點類似。MySQL 是目前最普及的資料庫伺服器，可是大家也最不在意它，可能因為它是一套免費的軟體，如果不要對它太過份，它會默默的在電腦中為你服務，在一般情況下都不太會出問題。MySQL 跟其它一般的資料庫管理系統一樣，同樣支援 ANSI SQL92，也加入少許 MySQL 自己特別的指令。不論是網頁或應用程式的開發人員，當你第一次接觸資料庫管理系統，學習 SQL 這種古老的指令，應該不會覺得太難。如果你正要進入開發應用程式的領域，在學習的路上，會分配給 SQL 的時間應該也不會太多，因為它跟程式語言比較起來是比較單純一些的。

因為 MySQL 和 SQL 幾乎是最常見的應用，而且大家也覺得它們是簡單的，當然就不會在它們身上花太多時間。所以慢慢的我們會發現一些情況，有一些應用程式發生的問題，其實是來自 MySQL 資料庫伺服器和應用程式中的 SQL 敘述，這些問題相對是比較單純的，只是大家忽略了。

例如 MySQL 提供方便好用的「LIMIT」子句，在應用程式中讓開發人員可以很容易完成一些特定的功能，例如網頁應用程式中的分頁查詢。不過 LIMIT 子句是 MySQL 才有的，如果應用程式更換資料庫伺服器，例如 Oracle，應用程式就會產生一堆錯誤了。還有資料庫的交易（transaction）管理，MySQL 提供的 MYISAM 儲存引擎並沒有支援交易管理，因為比較簡單一些，所以運作的效率也會比較好。如果應用程式需要執行交易管理，就要在建立資料庫的時候指定儲存引擎為 InnoDB。

各種關於 MySQL 資料庫管理和 SQL 的問題，開發人員通常在遇到錯誤的時候，才會開始尋求解決問題的方法。這似乎也是 MySQL 的宿命，因為我們雖然一直在使用它，可是卻不太重視它，也認為這本來就是合理的，開發人員不應該分配太多時間給它。有一個很明顯的情況，在逛書局的時候，你應該已經看不到只有討論關於 MySQL 和 SQL 的書籍了。

OCP MySQL Developer

跟開發人員相關的認證考試，這應該算是最冷門的 OCP 認證科目之一。這個認證考試的主要內容是 MySQL 的 SQL，通過這個考試的人，表示它具備在應用程式中使用 SQL 的技能。你應該會覺的這是一個有點詭異的認證考試，它

好像沒有存在的必要。對一個有經驗的開發人員來說，使用 SQL 的技能就像是本來就應該存在的，你甚至已經忘記當初是怎麼學會 SQL。對一個新手來說，不會有人建議你去買一本關於 SQL 的書籍來學習這方面的技能，因為可能也買不到了，不過有各種網站提供 SQL 的學習，認識一些基礎的敘述後，遇到問題再說吧！

SQL 在目前的環境下，越來越不受到開發人員的關愛，尤其是現在各種關於資料庫應用的框架，例如 Hibernate 和 MyBatis，它們的任務就是要殺死 SQL 這隻遠古巨獸，讓開發人員不用受到 SQL 的煎熬。我也認為開發應用程式一直是一件很困難的事情，各種越來越進步的科技讓生活更方便，可是應用程式開發技術卻越來越複雜，開發人員必須具備的技能也更多，如果真的能有一種技術可以完全消滅 SQL，那絕對是一件非常美好的事情。不過目前的情況應該還是有很多困難，就以大約十年前開發的應用程式來說，SQL 還是一個必要的成員，除非放棄原來已經運作正常的程式，否則你還是要面對這些冗長的 SQL 敘述。

MySQL 新手入門超級手冊

這就是《MySQL 新手入門超級手冊》這本書的目的，內容的範圍涵蓋 OCP MySQL 5 Developer 認證考試，它的範圍也是一個開發人員必須具備的 SQL 技能。從安裝 MySQL 資料庫與相關的工具程式開始，到學習所有 MySQL 提供的 SQL，雖然是針對 MySQL 資料庫撰寫的，不過絕大部份都符合 ANSI SQL92 的標準，也就是在其它資料庫產品也可以正確的運作。

MySQL 新手入門超級手冊總共有二十二章與一個附錄：

- 第一章說明資料庫概念與安裝需要的應用程式。
- 第二章到第五章說明基本的新增、修改、刪除與查詢。
- 第六章到第九章說明資料庫、儲存引擎、表格與索引，這個部份的內容會有比較多 MySQL 獨有的特色。
- 第十章是子查詢，熟悉子查詢的應用，可以完成很多複雜的工作。
- 第十一章是 Views 元件，它可以把經常執行的工作保存起來並重複使用。

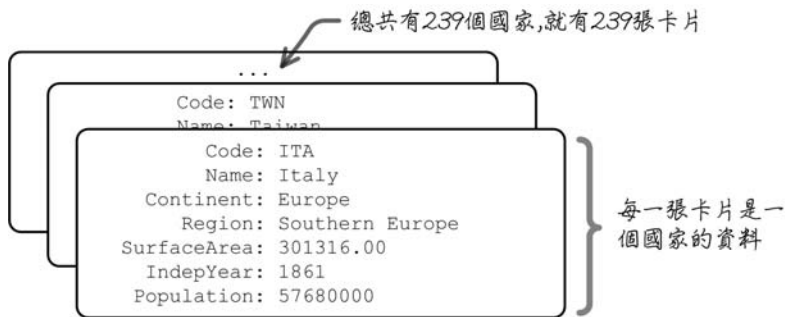
- 第十二章到第十六章是 MySQL 資料庫進階應用，其它資料庫產品也有類似的技術，例如 Oracle 的 PL/SQL。
- 第十七章到第二十章討論的內容，比較偏向於資料庫管理和效率的進階應用，這些也是一個開發人員需要瞭解的。
- 第二十一章與二十二章分別說明如何使用 Python 與 Java 程式設計語言，連線到資料庫與執行資料的新增、修改、刪除與查詢。
- MariaDB 是 MySQL 的分支，雖然目前 MySQL 仍然是最多人使用的資料庫管理系統，不過有越來越多人使用 MariaDB 取代 MySQL，所以附錄 A 說明在 Windows 作業系統安裝與設定 MariaDB 的作法，這本書說明的內容也可以在 MariaDB 正確的運作。

為方便教學，本書另提供教學投影片與各章課後習題，採用本書授課教師可向碁峰業務索取。

資料庫概論與 MySQL

1.1 儲存與管理資料

儲存與管理資料一直是資訊應用上最基本、也最常見的技術。在還沒有使用電腦管理資料的時候，你可能會使用這樣的方式來保存世界上所有的國家資料：



這樣的作法在生活中是很常見的，例如親友的通訊錄，你可能也會使用一張卡片來記錄一個親友的通訊資料，上面有名字、電話、住址，與所有你想要保存的資料。這種保存資料的方式很直接，也很省錢。不過你應該會遇到這樣的問題：

```

Code: AFG
Name: Afghanistan
Continent: Asia
Region: Southern and Central Asia
SurfaceArea: 652090.00
IndepYear: 1919
Population: 22720000
LifeExpectancy: 45.9
  GNP: 5976.00
  GNPOld: NULL
  LocalName: Afganistan/Afqanestan
GovernmentForm: Islamic Emirate
HeadOfState: Mohammad Omar
  Capital: 1
  Code2: AF
    
```

就算你依照字母的順序
把它們排好,想要在這
疊找到某一個國家的資
料,應該是很困難的

如果你買了一台個人電腦，電腦中也安裝了工作表的應用程式，像這類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的應用程式
來儲存所有國家的資料...

1	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

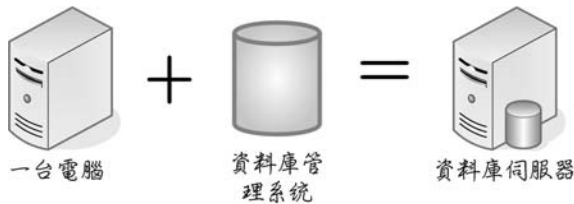
每一列都是一個國家的
資料,這樣應該好多了

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：

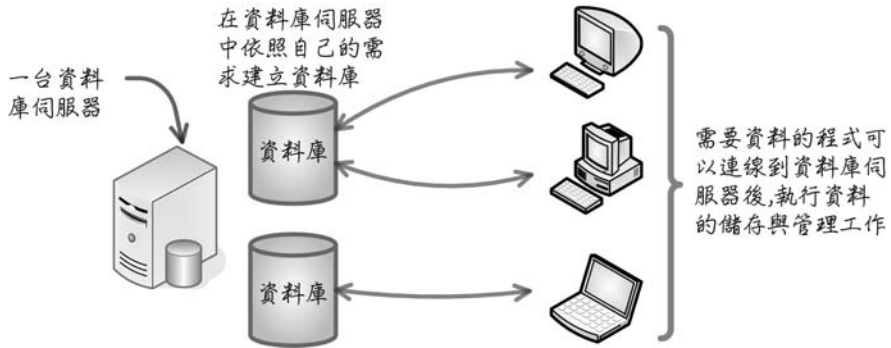
A	B	C	D	E	F	G	H
Code	Name	Continent	Region	SurfaceArea	IndepYear	Population	LifeExpect
1	DZA	Algeria	Africa	Northern Africa	2381741	1962	31471000
2	AGO	Angola	Africa	Central Africa	1246700	1975	12878000
3	ARE	United Arab Emirates	Asia	Middle East	83600	1971	2441000
4	ARM	Armenia	Asia	Western Asia	242900	1991	2997000
5	AZE	Azerbaijan	Asia	Western Asia	86600	1991	622000
6	BHR	Bahrain	Asia	Middle East	41526	1581	15864000
7	BGD	Bangladesh	Asia	Southern Asia	28748	1912	3401200
8	BTN	Bhutan	Asia	Southern Asia	468	1278	78000
9	BRN	Brunei	Asia	Southern Asia	30518	1830	10239000
10	PHL	Philippines	Asia	Southern Asia	51197	1992	3972000
11	GEO	Georgia	Asia	Western Asia	242900	1066	59623400
12	HKG	Hong Kong	Asia	Eastern Asia	110994	1908	8190900
13	IDN	Indonesia	Asia	Southern Asia	505992	1492	39441700
14	IND	India	Asia	Southern Asia	1399		43000
15	IRQ	Iraq	Asia	Middle East	6		25000
16	SJM	Svalbard and Jan Mayen	Europe	Nordic Countries	62422		3200
17	IRL	Ireland	Europe	British Isles	70273	1921	3775100
18	ISL	Iceland	Europe	Nordic Countries	103000	1944	279000
19	ITA	Italy	Europe	Southern Europe	301316	1861	57680000

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，你也會發現這是一件很困難的工作。

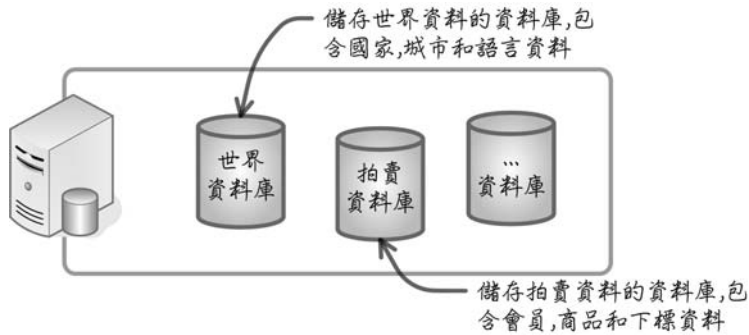
在資訊的管理與應用，「資料庫管理系統」是一種用來儲存與管理資料的應用程式，它使用安全、穩定與有效率的方式把資料儲存起來，也可以方便與快速的維護資料。尤其是資料的數量很龐大的時候，使用資料庫管理系統儲存與管理資料，會是一種令人安心而且比較有效率的方式。資料庫管理系統是一種應用程式，它主要的工作就是儲存與管理資料，如果你把這個應用程式安裝在一台電腦中，這台電腦就會稱為「資料庫伺服器」：



在你有了一台資料庫伺服器以後，就可以依照自己的需求，使用資料庫管理系統建立一些資料庫：



在使用資料庫之前,要先在資料庫伺服器中建立需要的「資料庫、database」。你會依照自己的需求,建立一個或多個資料庫:



以儲存世界國家資料的資料庫來說,你想要把世界上所有的國家、城市和語言資料,都放在這個資料庫中儲存與管理。所以你會針對國家資料的部份,在世界資料庫中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	Indep-Year	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	DEA	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

儲存在世界資料庫中的國家資料，隨時可以依照不同的需求，查詢需要的國家資料：

可以隨時從表格中查詢需要的資料

A	B	C	D	E	F	G		
1	Code	Name	Continent	Region	SurfaceArea	IndependYear	Population	LifeExp
2	ALG	Algeria	Africa	Northern Africa	2381741	1962	31471000	
3	AGO	Angola	Africa	Central Africa	1246700	1975	12878000	
4	BEN	Benin	Africa	Western Africa	112620	1960	6097000	
5	BWA	Botswana	Africa	Southern Africa	581730	1966	1620000	
6	BFA	Burkina Faso	Africa	Western Africa	274000	1960	11937000	
7	BUR	Burundi	Africa	Eastern Africa	27834	1962	6669000	
8	DJI	Djibouti	Africa	Eastern Africa	23200	1977	638000	
9	EGY	Egypt	Africa	Northern Africa	6847000		35200000	
10	AFQ	Alqanistan	Asia	Southern and Central Asia	652090	1919	22720000	
11	SGO	Socatar	Asia	Southern and Central Asia	41528	1981	15964000	
12	HKG	Hong K	Asia	Middle East	29749	1912	3401200	
13	IDN	Indonesia	Asia	Southern Asia	1904569		183639000	
14	IND	India	Asia	Southern Asia	3287263		688641400	
15	JPN	Japan	Asia	Eastern Asia	377837		123128400	
16	KOR	South Korea	Asia	Eastern Asia	100200		41989900	
17	CHN	China	Asia	Eastern Asia	9596960		1193256500	
18	PRC	China	Asia	Eastern Asia	9596960		1193256500	
19	THA	Thailand	Asia	Southeastern Asia	513120		58263500	
20	VNM	Vietnam	Asia	Southeastern Asia	331210		72994000	
21	PHL	Philippines	Asia	Southeastern Asia	340000		75927000	
22	MDG	Madagascar	Africa	Eastern Africa	458840		12696000	
23	MLD	Moldova	Europe	Eastern Europe	33840		4502000	
24	ROU	Romania	Europe	Eastern Europe	238391		21851000	
25	UKR	Ukraine	Europe	Eastern Europe	603700		47652000	
26	BLG	Bulgaria	Europe	Eastern Europe	110994		8100000	
27	CZE	Czechia	Europe	Central Europe	78866		10500000	
28	HUN	Hungary	Europe	Central Europe	93028		10301000	
29	POL	Poland	Europe	Central Europe	312685		38438000	
30	DEU	Germany	Europe	Central Europe	357022		82164700	
31	FRA	France	Europe	Western Europe	640390		64380000	
32	GBR	Great Britain	Europe	Western Europe	244810		56858000	
33	IRL	Ireland	Europe	British Isles	70273		3775000	
34	ISL	Iceland	Europe	Northern Europe	103000		2944000	
35	ITA	Italy	Europe	Southern Europe	301316		57688000	
36	ESP	Spain	Europe	Southern Europe	505992		41754000	
37	GRE	Greece	Europe	Southern Europe	131250		11351000	
38	CYP	Cyprus	Europe	Southern Europe	9241		770000	
39	ISR	Israel	Asia	Middle East	20760		4510000	
40	JOR	Jordan	Asia	Middle East	92800		5209000	
41	LIB	Lebanon	Asia	Middle East	10400		3204000	
42	SYR	Syria	Asia	Middle East	198500		16907000	
43	IRQ	Iraq	Asia	Middle East	438300		23324000	
44	YEM	Yemen	Asia	Middle East	527970		18505000	
45	OMN	Oman	Asia	Middle East	121300		2505000	
46	ARE	U.A.E.	Asia	Middle East	83760		2200000	
47	SAU	Saudi Arabia	Asia	Middle East	2149690		21500000	
48	IRQ	Iraq	Asia	Middle East	438300		23324000	
49	YEM	Yemen	Asia	Middle East	527970		18505000	
50	OMN	Oman	Asia	Middle East	121300		2505000	
51	ARE	U.A.E.	Asia	Middle East	83760		2200000	
52	SAU	Saudi Arabia	Asia	Middle East	2149690		21500000	

除了國家表格外，你還會在世界資料庫中建立儲存城市和語言資料的表格：

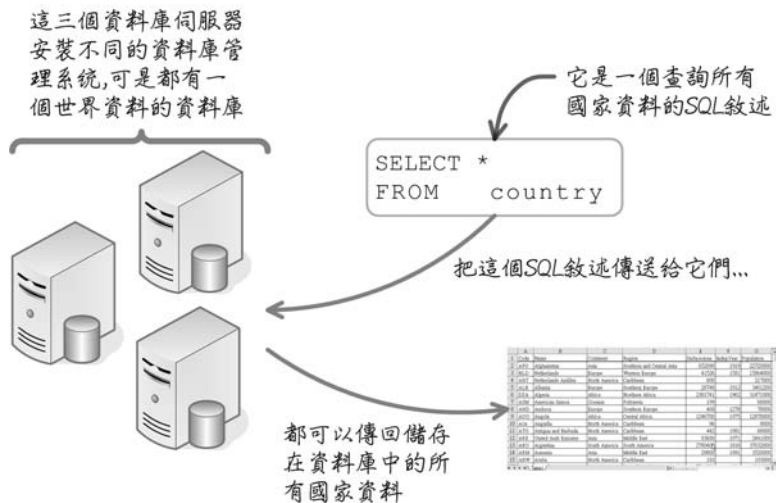
在國家表格裡儲存國家的資訊

A	B	C	D	E	F	G		
1	Code	Name	Continent	Region	SurfaceArea	IndependYear	Population	LifeExp
2	AFQ	Alqanistan	Asia	Southern and Central Asia	652090	1919	22720000	
3	NLD	Netherlands	Europe	Western Europe	41528	1581	15964000	
4	ANT	Netherlands Antilles	North America	Caribbean	800		217000	
5	ALB	Albania	Europe	Southern Europe	28748	1912	3401200	
6	ALG	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	ATG	Antigua and Barbuda	North America	Caribbean	96		8000	
11	ARG	Argentina	South America	South America	2780406	1816	37032000	
12	AUS	Australia	Oceania	Oceania	29600	1991	3520000	
13	AUT	Austria	Europe	Central Europe	193		103000	
14	BAN	Bangladesh	Asia	Southern Asia	147580		1193256500	
15	BLR	Belarus	Europe	Eastern Europe	207600		10301000	
16	BEL	Belgium	Europe	Western Europe	30518	1830	10239000	
17	BEN	Benin	Africa	Western Africa	112620	1960	6097000	
18	BFA	Burkina Faso	Africa	Western Africa	274000	1960	11937000	
19	BUR	Burundi	Africa	Eastern Africa	27834	1962	6669000	
20	BWA	Botswana	Africa	Southern Africa	581730	1966	1620000	
21	BFA	Burkina Faso	Africa	Western Africa	274000	1960	11937000	
22	BUR	Burundi	Africa	Eastern Africa	27834	1962	6669000	
23	BWA	Botswana	Africa	Southern Africa	581730	1966	1620000	
24	BFA	Burkina Faso	Africa	Western Africa	274000	1960	11937000	
25	BUR	Burundi	Africa	Eastern Africa	27834	1962	6669000	
26	BWA	Botswana	Africa	Southern Africa	581730	1966	1620000	
27	BFA	Burkina Faso	Africa	Western Africa	274000	1960	11937000	
28	BUR	Burundi	Africa	Eastern Africa	27834	1962	6669000	
29	BWA	Botswana	Africa	Southern Africa	581730	1966	1620000	
30	BFA	Burkina Faso	Africa	Western Africa	274000	1960	11937000	
31	BUR	Burundi	Africa	Eastern Africa	27834	1962	6669000	
32	BWA	Botswana	Africa	Southern Africa	581730	1966	1620000	
33	BFA	Burkina Faso	Africa	Western Africa	274000	1960	11937000	
34	BUR	Burundi	Africa	Eastern Africa	27834	1962	6669000	
35	BWA	Botswana	Africa	Southern Africa	581730	1966	1620000	
36	BFA	Burkina Faso	Africa	Western Africa	274000	1960	11937000	
37	BUR	Burundi	Africa	Eastern Africa	27834	1962	6669000	
38	BWA	Botswana	Africa	Southern Africa	581730	1966	1620000	
39	BFA	Burkina Faso	Africa	Western Africa	274000	1960	11937000	
40	BUR	Burundi	Africa	Eastern Africa	27834	1962	6669000	
41	BWA	Botswana	Africa	Southern Africa	581730	1966	1620000	
42	BFA	Burkina Faso	Africa	Western Africa	274000	1960	11937000	
43	BUR	Burundi	Africa	Eastern Africa	27834	1962	6669000	
44	BWA	Botswana	Africa	Southern Africa	581730	1966	1620000	
45	BFA	Burkina Faso	Africa	Western Africa	274000	1960	11937000	
46	BUR	Burundi	Africa	Eastern Africa	27834	1962	6669000	
47	BWA	Botswana	Africa	Southern Africa	581730	1966	1620000	
48	BFA	Burkina Faso	Africa	Western Africa	274000	1960	11937000	
49	BUR	Burundi	Africa	Eastern Africa	27834	1962	6669000	
50	BWA	Botswana	Africa	Southern Africa	581730	1966	1620000	
51	BFA	Burkina Faso	Africa	Western Africa	274000	1960	11937000	
52	BUR	Burundi	Africa	Eastern Africa	27834	1962	6669000	
53	BWA	Botswana	Africa	Southern Africa	581730	1966	1620000	
54	BFA	Burkina Faso	Africa	Western Africa	274000	1960	11937000	
55	BUR	Burundi	Africa	Eastern Africa	27834	1962	6669000	
56	BWA	Botswana	Africa	Southern Africa	581730	1966	1620000	
57	BFA	Burkina Faso	Africa	Western Africa	274000	1960	11937000	
58	BUR	Burundi	Africa	Eastern Africa	27834	1962	6669000	
59	BWA	Botswana	Africa	Southern Africa	581730	1966	1620000	
60	BFA	Burkina Faso	Africa	Western Africa	274000	1960	11937000	
61	BUR	Burundi	Africa	Eastern Africa	27834	1962	6669000	
62	BWA	Botswana	Africa	Southern Africa	581730	1966	1620000	
63	BFA	Burkina Faso	Africa	Western Africa	274000	1960	11937000	
64	BUR	Burundi	Africa	Eastern Africa	27834	1962	6669000	
65	BWA	Botswana	Africa	Southern Africa	581730	1966	1620000	
66	BFA	Burkina Faso	Africa	Western Africa	274000	1960	11937000	
67	BUR	Burundi	Africa	Eastern Africa	27834	1962	6669000	
68	BWA	Botswana	Africa	Southern Africa	581730	1966	1620000	
69	BFA	Burkina Faso	Africa	Western Africa	274000	1960	11937000	
70	BUR	Burundi	Africa	Eastern Africa	27834	1962	6669000	
71	BWA	Botswana	Africa	Southern Africa	581730	1966	1620000	
72	BFA	Burkina Faso	Africa	Western Africa	274000	1960	11937000	
73	BUR	Burundi	Africa	Eastern Africa	27834	1962	6669000	
74	BWA	Botswana	Africa	Southern Africa	581730	1966	1620000	
75	BFA	Burkina Faso	Africa	Western Africa	274000	1960	11937000	
76	BUR	Burundi	Africa	Eastern Africa	27834	1962	6669000	
77	BWA	Botswana	Africa	Southern Africa	581730	1966	1620000	
78	BFA	Burkina Faso	Africa	Western Africa	274000	1960	11937000	
79	BUR	Burundi	Africa	Eastern Africa	27834	1962	6669000	
80	BWA	Botswana	Africa	Southern Africa	581730	1966	1620000	
81	BFA	Burkina Faso	Africa	Western Africa	274000	1960	11937000	
82	BUR	Burundi	Africa	Eastern Africa	27834	1962	6669000	
83	BWA	Botswana	Africa	Southern Africa	581730	1966	1620000	
84	BFA	Burkina Faso	Africa	Western Africa	274000	1960	11937000	
85	BUR	Burundi	Africa	Eastern Africa	27834	1962	6669000	
86	BWA	Botswana	Africa	Southern Africa	581730	1966	1620000	
87	BFA	Burkina Faso	Africa	Western Africa	274000	1960	11937000	
88	BUR	Burundi	Africa	Eastern Africa	27834	1962	6669000	
89	BWA	Botswana	Africa	Southern Africa	581730	1966	1620000	
90								

1.2 認識 Structured Query Language

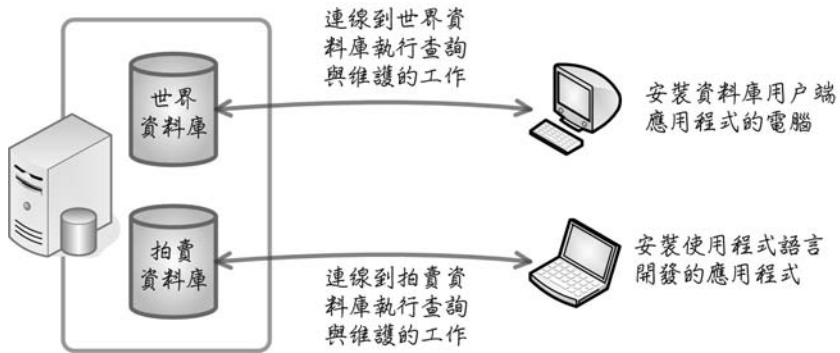
有許多廠商開發各種不同的資料庫管理系統產品，它們都可以執行儲存與管理資料的工作，而且使用的方式都是差不多的。執行資料儲存與管理的工作，主要有建立資料庫與表格，還有執行資料的新增、修改、刪除與查詢。想要請資料庫管理系統執行這些工作，你會使用一種叫作「Structured Query Language、SQL」的敘述，一般會把「SQL」唸為「sequel」。

SQL 在很久以前就已經是一種標準的技術，不同的資料庫管理系統產品，在執行資料庫的工作時，使用的 SQL 敘述幾乎是一樣的：

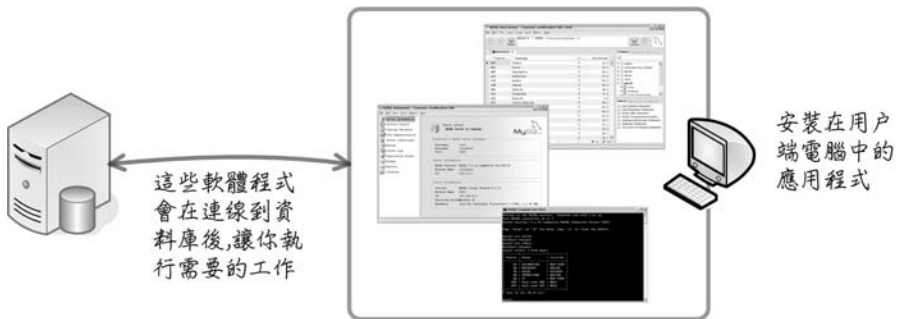


SQL 有一套國際通用的標準，裡面規定所有執行資料庫工作的 SQL 敘述要怎麼寫，不同的資料庫管理系統產品都會以這套標準為基礎。不過不同的產品通常會增加或修改一些 SQL 敘述，其它的資料庫管理系統就不認識這些 SQL 敘述了。

與資料庫伺服器相對的是「用戶端、client」：



資料庫伺服器通常會提供一些用戶端應用程式，讓使用者可以輸入與執行 SQL 敘述，或是執行管理與設定資料庫伺服器的工作：



使用像是 Java 程式設計技術開發的各種應用程式，例如進銷存系統或會計系統，對資料庫伺服器來說，也算是一種用戶端軟體：



Chapter 5

資料維護

在瞭解資料庫查詢的各種應用以後，這一章說明資料庫的新增、修改與刪除。包含新增資料的「INSERT」與「REPLACE」敘述，修改資料的「UPDATE」敘述，還有刪除資料的「DELETE」與「TRUNCATE」敘述。

5.1 取得表格資訊

在執行資料維護的時候，例如新增一筆資料到指定的表格，應該要清楚知道表格的詳細資訊，包含表格欄位的名稱、型態、順序和預設值，還有表格的主索引鍵。這些資訊都會影響資料維護敘述的寫法，如果違反表格的設計架構，就可能產生錯誤或儲存錯誤的資料。

「DESCRIBE」是 MySQL 資料庫提供的指令，它只能在 MySQL 資料庫中使用，這個指令可以取得指定表格的結構資訊，下面是它的語法：

「DESCRIBE」是MySQL才有的指令,並不是SQL指令 → DESCRIBE 表格名稱 ←

也可以使用縮寫「DESC」,不過不要跟排序使用的「DESC」搞混了 → DESC 表格名稱 ←

取得後面指定的表格結構資訊

執行「DESC cmdev.dept」指令以後，MySQL 會傳回「cmdev.dept」表格的結構資訊：

Field	Type	Null	Key	Default	Extra
deptno	int (2)	NO	PRI	NULL	
dname	varchar (16)	NO		NULL	
location	varchar (16)	YES		NULL	

根據上列查詢後得到的結果，cmdev.dept 表格總共有三個欄位，分別是「deptno」、「dname」與「location」，deptno 是儲存部門編號的數字欄位，dname 與 location 是儲存部門名稱與地點的文字欄位。部門編號與名稱欄位不可以儲存 NULL 值，地點欄位可以儲存 NULL 值。這個表格把 deptno 欄位設定為主索引鍵，所以部門編號的值不可以重複。

每一個表格在設計的時候，都會決定它有哪一些欄位和各自的詳細設定。另外也會決定表格中的欄位順序，在新增資料的時候需要明確知道表格的欄位順序：

DESC cmdev.emp

Field	Type	Null	Key	Default	Extra
empno	int (4)	NO	PRI	NULL	
ename	varchar (16)	NO		NULL	
job	varchar (16)	YES		NULL	
manager	int (4)	YES		NULL	
hiredate	date	YES		NULL	
salary	float (7,2)	YES		NULL	
comm	float (7,2)	YES		NULL	
deptno	int (2)	YES		NULL	

5.2 新增資料

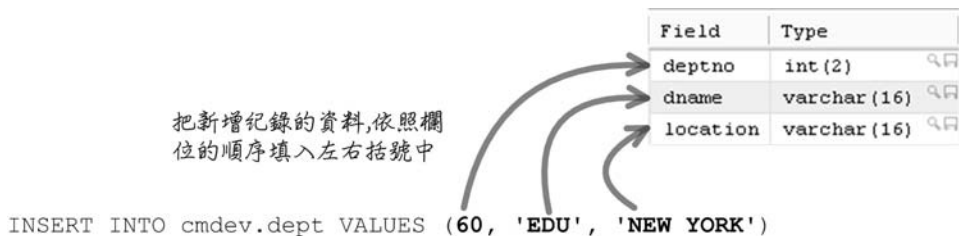
MySQL 提供「INSERT」與「REPLACE」兩種新增資料到指定表格的敘述，一般新增資料的需求可以使用 INSERT 敘述。INSERT 搭配「IGNORE」與「ON DUPLICATE KEY UPDATE」子句，還可以完成一些特殊的新增資料需求。REPLACE 敘述的語法跟 INSERT 非常類似，不過它同時具有新增或修改資料的功能。接下來的內容會詳細說明新增敘述的語法和應用。

5.2.1 基礎新增敘述

需要新增資料到資料庫的表格中使用「INSERT」敘述，下面它的基本語法：



使用這個語法新增紀錄的時候，要特別注意表格的欄位個數與順序，下列的新增敘述會新增一筆部門資料到「cmdev.dept」表格中：



除了明確指定新增紀錄的每一個欄位資料外，你也可以使用「DEFAULT」關鍵字，讓 MySQL 使用在設計表格的時候為欄位指定的預設值。下列的新增敘述同樣會新增一筆部門資料到「cmdev.dept」表格中，不過部門的所在位置（location）欄位值指定為使用預設值：


```
INSERT INTO cmdev.dept
VALUES (70, 'MARKETING', DEFAULT)
```

Field	Type	Null	Key	Default
deptno	int (2)	NO	PRI	NULL
dname	varchar (16)	NO		NULL
location	varchar (16)	YES		NULL

使用「DEFAULT」
關鍵字寫入預設值

這個欄位的預設值是
「NULL」,所以這筆新增
紀錄的「location」欄位
值會寫入「NULL」


使用這種語法新增紀錄的時候，如果在 VALUES 後面提供的資料個數與欄位個數不一樣的話，就會發生錯誤：

 INSERT INTO cmdev.dept
VALUES (80, 'PURCHASING')

這個表格總共有三個欄位

左右括號中只有兩個資料

資料個數雖然沒有錯，順序卻不對了，也有可能造成錯誤：

 INSERT INTO cmdev.dept
VALUES ('PURCHASING', 80, 'NEW YORK')

Field	Type
deptno	int (2)
dname	varchar (16)
location	varchar (16)

錯誤的順序

新增敘述的另外一種語法提供比較靈活的作法，你可以自己指定新增紀錄的欄位個數和順序：

```
INSERT [INTO] 表格名稱 [(欄位名稱,...)]
VALUES (運算式|DEFAULT,...)
```

根據自己的需求,指
定要儲存資料的欄位

根據指定的欄位,填
入要儲存的資料

在額外為這個新增敘述指定欄位以後，指定儲存資料的時候，就要依照自己指定的欄位個數與順序：

```
INSERT INTO cmdev.dept (dname, deptno, location)
VALUES ('PURCHASING', 80, 'NEW YORK')
```

使用自己指定的欄位順序填入要儲存的資料

如果沒有依照自己指定的欄位個數與順序，就會發生錯誤：

指定三個欄位...

```
✘ INSERT INTO cmdev.dept (deptno, dname, location)
VALUES (90, 'SHIPPING')
```

卻只有給兩個資料

因為這種新增敘述的語法可以自己指定欄位的個數與順序，所以只要指定寫入欄位的資料就可以了。不過要特別注意下列兩種語法的差異：

Field	Type	Null	Key	Default
deptno	int (2)	NO	PRI	NULL
dname	varchar (16)	NO		NULL
location	varchar (16)	YES		NULL

```
INSERT INTO cmdev.dept
VALUES (90, 'SHIPPING', DEFAULT)
```

使用「DEFAULT」是寫入欄位設定的預設值

```
INSERT INTO cmdev.dept (deptno, dname)
VALUES (90, 'SHIPPING')
```

沒有指定的欄位一律寫入「NULL」

因為上面說明的規定，下列這個新增敘述在語法上雖然沒有錯誤，可是違反表格設計上的規定，所以執行敘述以後會發生錯誤：

 `INSERT INTO cmdev.dept (deptno, location)`
`VALUES (80, 'NEW YORK')`

沒有指定的「dname」欄位會寫入「NULL」...

Field	Type	Null	Key	Default
deptno	int(2)	NO	PRI	NULL
dname	varchar(16)	NO		NULL
location	varchar(16)	YES		NULL

可是「dname」欄位卻不可以儲存「NULL」

這種新增敘述的語法還有一種比較特別的用法，如果你要新增的紀錄，所有欄位的值都要使用預設值，就可以使用下面說明的寫法。不過要特別注意，下列的新增敘述執行以後會造成錯誤，因為「deptno」與「dname」欄位的預設值是「NULL」，可是它們又不能儲存「NULL」：

`INSERT INTO cmdev.dept () VALUES ()`

沒有指定任何欄位

沒有指定任何資料

Field	Type	Null	Key	Default
deptno	int(2)	NO	PRI	NULL
dname	varchar(16)	NO		NULL
location	varchar(16)	YES		NULL

全部使用欄位設定的預設值

下列是新增敘述的第三種語法：

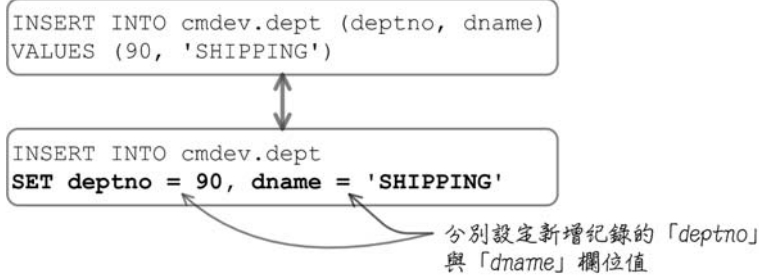
`INSERT [INTO] 表格名稱`
`SET 欄位名稱 = 運算式|DEFAULT [, ...]`

使用「SET」子句

設定新增紀錄的欄位值

可以設定多個欄位值

這種語法只是提供你另外一種新增紀錄的寫法，下列兩個新增敘述的效果是一樣的：



5.2.2 同時新增多筆紀錄

上列說明的新增敘述在執行以後，如果沒有發生任何錯誤，都只會新增一筆紀錄到指定的表格。如果需要的話，你也可以使用一個 INSERT 敘述新增多筆紀錄，差異只有在「VALUES」子句後面新增資料的指定：

```
INSERT [INTO] 表格名稱 [(欄位名稱,...)]
VALUES (運算式|DEFAULT,...) [,...]
```

在「VALUE」子句後面一個左右括號表示新增一筆紀錄

可以在一個敘述新增多筆紀錄

如果需要新增下列三個員工資料到「cmdev.emp」表格中：

empno	ename	job	manager	hiredate	salary	comm	deptno
8001	SIMON	MANAGER	7369	2001 02 03	3300	NULL	50
8002	JOHN	PROGRAMMER	8001	2002 01 01	2300	NULL	50
8003	GREEN	ENGINEER	8001	2003 05 01	2000	NULL	50

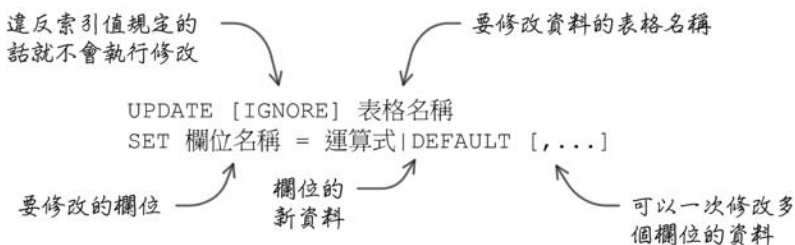
你可以分別執行三個新增敘述，將三個員工資料新增到「cmdev.emp」表格中。也可以使用下列說明的作法，這個敘述執行以後會一次新增三筆紀錄：

```
INSERT INTO cmdev.emp VALUES
(8001, 'SIMON', 'MANAGER', 7369, '2001-02-03', 3300, NULL, 50),
(8002, 'JOHN', 'PROGRAMMER', 8001, '2002-01-01', 2300, NULL, 50),
(8003, 'GREEN', 'ENGINEER', 8001, '2003-05-01', 2000, NULL, 50)
```

在「VALUE」子句後面一個左右括號表示新增一筆紀錄，這個敘述總共新增三筆紀錄

5.3 修改資料

修改已經儲存在表格中的紀錄使用「UPDATE」敘述，下列是它的基本語法：



使用「UPDATE」敘述的時候，通常會搭配使用「WHERE」子句，用來指定要修改的紀錄：

```
UPDATE [IGNORE] 表格名稱
SET 欄位名稱 = 運算式|DEFAULT [, ...]
```

修改這個表格裡的所有紀錄

```
UPDATE [IGNORE] 表格名稱
SET 欄位名稱 = 運算式|DEFAULT [, ...]
WHERE 條件
```

只有符合條件的紀錄才會修改

所以在執行「UPDATE」敘述的時候，通常會依照實際的需求，正確設定修改的條件。以下列兩個修改敘述來說，它們執行後的差異是很大的：

```
UPDATE cmdev.emp
SET salary = salary + 100
```

所有的員工都加薪100

```
UPDATE cmdev.emp
SET salary = salary + 100
WHERE salary < 1500
```

只有薪水小於1500的員工才會加薪100

5.3.1 搭配「IGNORE」

在使用「UPDATE」敘述的時候，可以依照需求加入「IGNORE」關鍵字，它可以防止錯誤的修改敘述出現錯誤訊息：

把部門編號30改為50,因為違反「主索引鍵」的規則,所以會發生錯誤

✘

```
UPDATE cmdev.dept
SET deptno = 50
WHERE deptno = 30
```

deptno	dname	location
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON
50	IT	NEW YORK

```
UPDATE IGNORE cmdev.dept
SET deptno = 50
WHERE deptno = 30
```

使用「IGNORE」關鍵字,所以不會發生錯誤,也不會修改資料

除了因為主索引鍵造成的問題，另外也要注意修改多個欄位值的情況。首先是沒有使用「IGNORE」關鍵字的時候，錯誤的資料會在執行修改敘述的時候產生錯誤訊息，所以也不會執行任何修改的動作：

執行修改前

empno	salary	comm
7369	800.00	NULL

```
UPDATE cmdev.emp
SET salary = 'HELLO', comm = 1000
WHERE empno = 7369
```

✘

執行修改後

empno	salary	comm
7369	800.00	NULL

同樣的修改敘述加入「IGNORE」關鍵字，執行後的結果就不一樣了：

執行修改前

empno	salary	comm
7369	800.00	NULL

```
UPDATE IGNORE cmdev.emp
SET salary = 'HELLO', comm = 1000
WHERE empno = 7369
```

只有警告訊息

Description	ErrorNr.
Data truncated for column 'salary' at row 1	1265

執行修改後

empno	salary	comm
7369	0.00	1000.00

因為「HELLO」不能被轉換為數字,所以填入0

這個欄位的值沒有問題,所以還是會修改

5.3.2 搭配「ORDER BY」與「LIMIT」

執行修改的時候使用「WHERE」子句是一般最常見的作法，在處理一些比較特殊的修改需求時，也會搭配「ORDER BY」與「LIMIT」子句：

```
UPDATE [IGNORE] 表格名稱
SET 欄位名稱 = 運算式|DEFAULT [, ...]
[WHERE 條件]
[ORDER BY 排序] ← 先執行排序
[LIMIT 限制] ← 設定修改資料的紀錄數量
```

「LIMIT」子句也可以在查詢敘述中使用，不過在「UPDATE」敘述中使用「LIMIT」子句只能夠指定一個數字：

```
SELECT ...
LIMIT 數字
SELECT ...
LIMIT 數字1, 數字2
UPDATE ...
LIMIT 數字
```

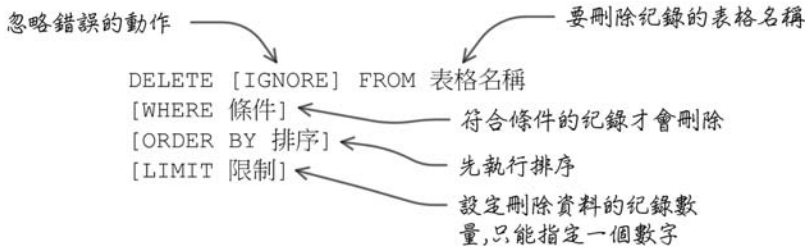
} 搭配「SELECT」使用
 的時候有一個數字和
 兩個數字兩種用法
 } 搭配「UPDATE」使用
 的時候只有一個數字
 的用法

以同樣為員工加薪一百的需求來說，搭配「ORDER BY」與「LIMIT」子句，可以完成許多不同的需求：

<pre>UPDATE cmdev.emp SET salary = salary + 100 ORDER BY salary</pre>	所有員工加薪100
<pre>UPDATE cmdev.emp SET salary = salary + 100 LIMIT 3</pre>	前三個員工加薪100
<pre>UPDATE cmdev.emp SET salary = salary + 100 ORDER BY salary LIMIT 3</pre>	薪水最低的三個員工加薪100
<pre>UPDATE cmdev.emp SET salary = salary + 100 ORDER BY salary DESC LIMIT 3</pre>	薪水最高的三個員工加薪100

5.4 刪除資料

刪除表格中已經不需要的紀錄，可以使用「DELETE」與「TRUNCATE」敘述。單獨使用 DELETE 敘述的時候，可以刪除表格中的所有記錄，搭配其它子句可以刪除特定的記錄。下列是 DELETE 敘述的語法：



使用「DELETE」敘述的時候，通常也會使用「WHERE」子句設定要刪除哪些紀錄：

DELETE FROM cmdev.emp	刪除所有員工
DELETE FROM cmdev.emp WHERE salary < 1500	刪除薪水小於1500的員工

執行刪除的時候也可以搭配「ORDER BY」與「LIMIT」子句：

DELETE FROM cmdev.emp ORDER BY salary	刪除所有員工
DELETE FROM cmdev.emp LIMIT 3	刪除前三個員工
DELETE FROM cmdev.emp ORDER BY salary LIMIT 3	刪除薪水最低的三個員工
DELETE FROM cmdev.emp ORDER BY salary DESC LIMIT 3	刪除薪水最高的三個員工

如果要刪除一個表格中所有的紀錄，你可以選擇使用「TRUNCATE」敘述，下列是它的語法：

TRUNCATE [TABLE] 表格名稱



要執行刪除表格中所有的紀錄，下列兩個敘述的效果是一樣的：

<pre>DELETE FROM cmdev.emp</pre>	}	這兩個敘述都可以刪除所有紀錄
<pre>TRUNCATE TABLE cmdev.emp</pre>		

「TRUNCATE」敘述在執行刪除紀錄的時候，會比使用「DELETE」敘述的效率好一些，尤其是表格中的紀錄非常多的時候會更明顯。