

SQL



匯入範例資料庫

1. 請先建立一個資料庫如下

```
CREATE DATABASE orders;
```

2. 選用 orders 資料庫

```
use orders;
```

phpMyAdmin

Server: 127.0.0.1

Databases SQL Status User accounts Export Import Settings

Databases

Create

orders utf8_unicode_ci Create

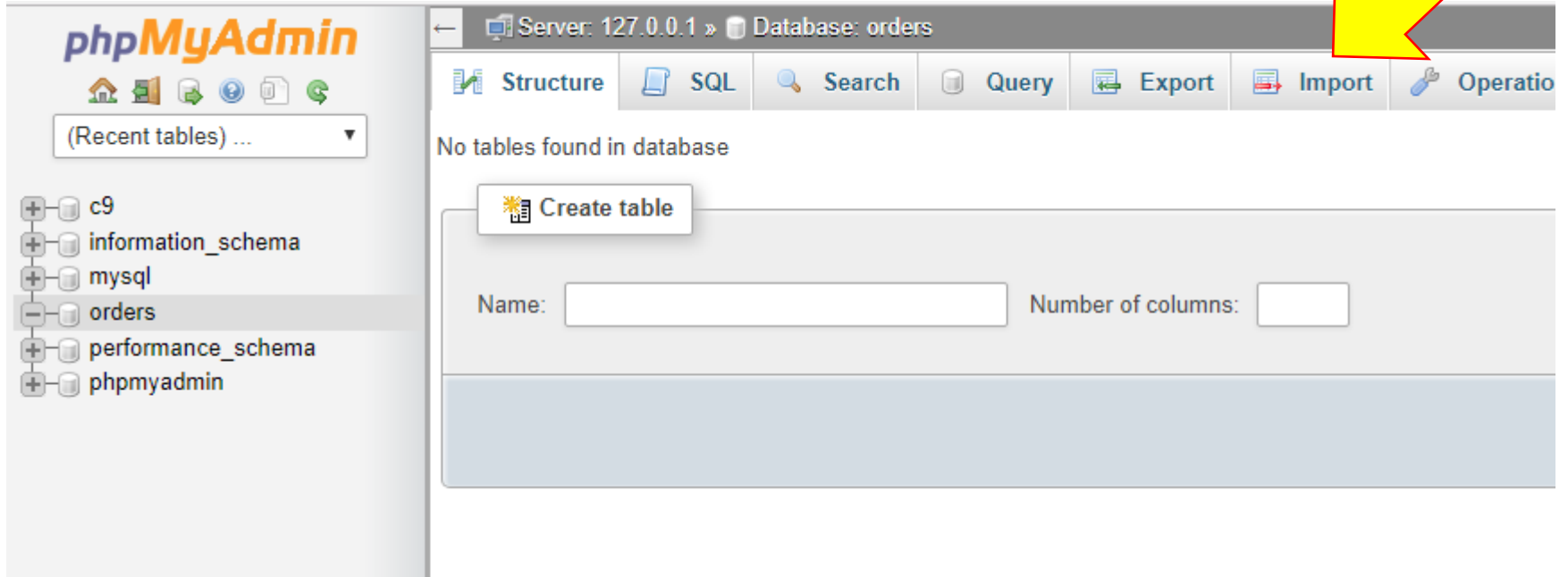
Database	Collation	Action
<input type="checkbox"/> information_schema	utf8_general_ci	<input type="checkbox"/> Check privileges
<input type="checkbox"/> mysql	latin1_swedish_ci	<input type="checkbox"/> Check privileges
<input type="checkbox"/> performance_schema	utf8_general_ci	<input type="checkbox"/> Check privileges
<input type="checkbox"/> phpmyadmin	utf8_bin	<input type="checkbox"/> Check privileges
<input type="checkbox"/> test	latin1_swedish_ci	<input type="checkbox"/> Check privileges
Total: 5	latin1_swedish_ci	

Check all With selected: Drop

Note: Enabling the database statistics here might cause heavy traffic between the web server and the MySQL

- **Enable statistics**

選擇 orders 資料庫，再選擇 Import



The screenshot displays the phpMyAdmin web interface. On the left sidebar, the 'orders' database is selected. The main content area shows the 'Database: orders' header and a toolbar with buttons for Structure, SQL, Search, Query, Export, Import, and Operations. A yellow arrow points to the 'Import' button. Below the toolbar, the text 'No tables found in database' is visible, followed by a 'Create table' button and input fields for 'Name:' and 'Number of columns:'.

Importing into the database "orders"

File to Import:

File may be compressed (gzip, bzip2, zip) or uncompressed.

A compressed file's name must end in `.[format].[compression]`. Example: `.sql.zip`

Browse your computer: orders_innoDB.sql (Max: 2,048KiB)

Character set of the file:



Partial Import:

Allow the interruption of an import in case the script detects it is close to the PHP timeout limit. (7 it can break transactions.)

Number of rows to skip, starting from the first row:

Format:

Format-Specific Options:

SQL compatibility mode:

Do not use `AUTO_INCREMENT` for zero values

- + c9
- + information_schema
- + mysql
- orders
 - Tables
 - New
 - + autonumber
 - + categories
 - + customers
 - + departments
 - + employees
 - + orderdetails
 - + orders
 - + postal
 - + products
 - + shippers
 - + suppliers
 - + Views
- + performance_schema
- + phpmyadmin

```

SELECT *
FROM `employees`
LIMIT 0 , 30

```

Profiling [[Inline](#)] [[Edit](#)] [[Explain S](#)]

Show : Start row: Number of rows: Headers every rows

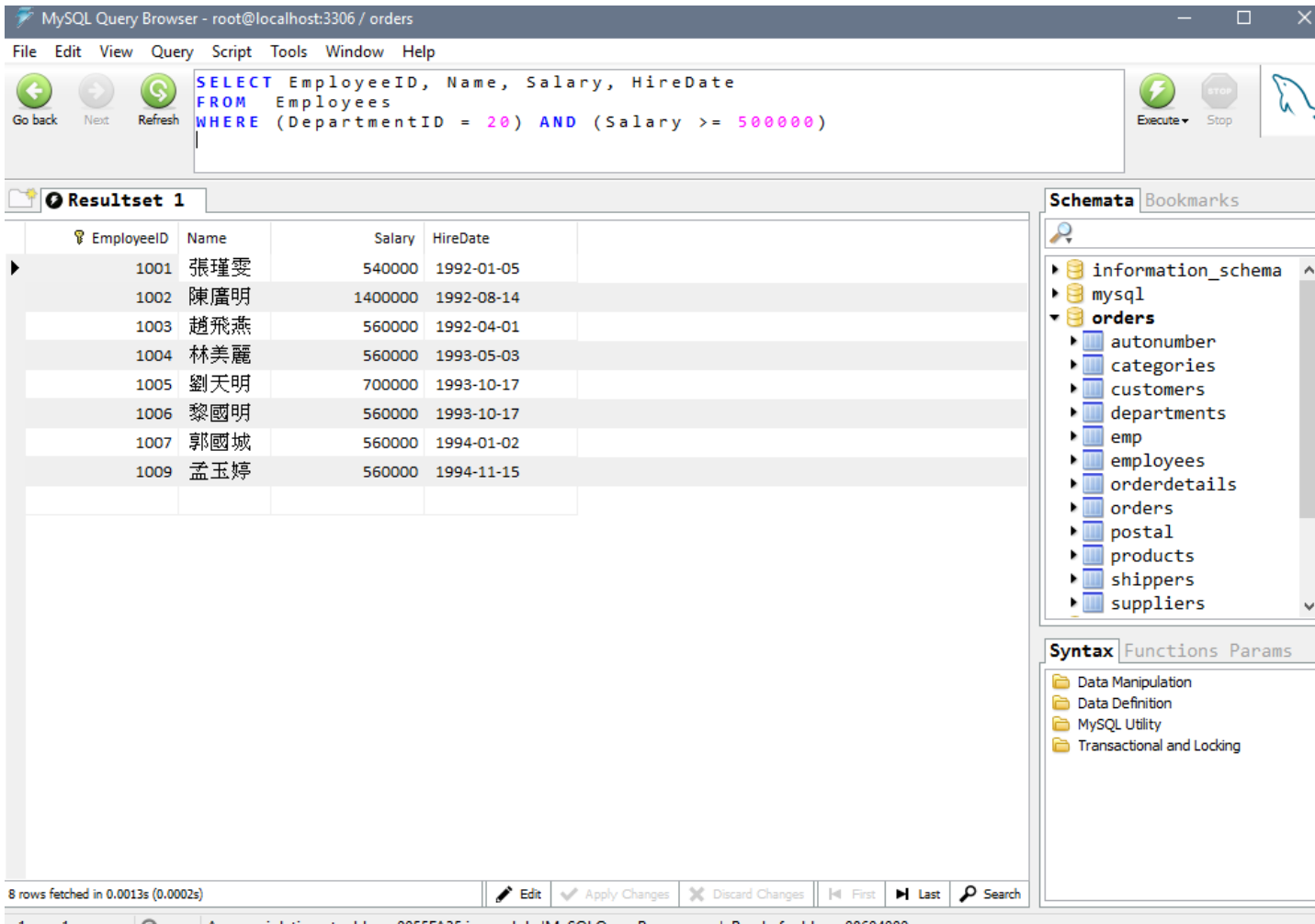
Sort by key:

+ Options

	EmployeeID	DepartmentID	SupervisorID	ReportsTo	Name	EngName	SSN	
<input type="checkbox"/> Edit Copy Delete	1001	20	1005	1002	張瑾 靈	Mary	A221112	
<input type="checkbox"/> Edit Copy Delete	1002	20	NULL	NULL	陳廣 明	Bradley	B121112	
<input type="checkbox"/> Edit Copy Delete	1003	20	1005	1002	趙飛 燕	Maria	B221132	

本章作業寫法範例 (SQL指令 & 結果截圖，用powerpoint 截圖，一頁一題):
請查詢員工之薪資年薪超過50萬且為業務部門20的員工之
工號、姓名、年薪及雇用日期。

```
SELECT EmployeeID, Name, Salary, HireDate  
FROM employees  
WHERE (DepartmentID = 20) AND (Salary >= 500000)
```



The screenshot shows the MySQL Query Browser interface. The query window contains the following SQL statement:

```
SELECT EmployeeID, Name, Salary, HireDate  
FROM employees  
WHERE (DepartmentID = 20) AND (Salary >= 500000)
```

The results are displayed in a table with the following columns: EmployeeID, Name, Salary, and HireDate. The results are as follows:

EmployeeID	Name	Salary	HireDate
1001	張瑾雯	540000	1992-01-05
1002	陳廣明	1400000	1992-08-14
1003	趙飛燕	560000	1992-04-01
1004	林美麗	560000	1993-05-03
1005	劉天明	700000	1993-10-17
1006	黎國明	560000	1993-10-17
1007	郭國城	560000	1994-01-02
1009	孟玉婷	560000	1994-11-15

The interface also shows a Schemata panel on the right with a tree view of the database structure, including schemas like information_schema, mysql, and orders. The Syntax panel at the bottom right shows categories like Data Manipulation, Data Definition, MySQL Utility, and Transactional and Locking.

撰寫SQL指令

- SQL語言是一種非程序性語言(Non-procedure Language)，是一種對資料庫逐次以一個個命令來操作資料庫的簡單語言。在SQL語法上和一般程式語言一樣，也有所謂的保留字或關鍵字(Keyword)。一般而言，保留字不可以用來當作資料表或欄位屬性的名稱。
- 程式設計者只須將自己打算做什麼表示出來，不須去理解電腦的執行過程。

SQL保留字(以MySQL為例)

- **alter**
- **asc**
- **between**
- **cross**
- **distinct**
- **from**
- **having**
- **avg**
- **case**
- **else**
- **for**
- **and**
- **char**
- **constraint**
- **desc**
- **as**
- **by**
- **creat**
- **describe**
- **drop**
- **exists**
- **group**
- **identified**

SQL保留字(以MySQL為例)(續)

- **if**
- **in**
- **order**
- **like**
- **jey**
- **insert**
- **update**
- **not**
- **lock**
- **is**
- **outer**
- **natural**
- **keys**
- **table**
- **null**
- **left**
- **to**
- **select**
- **on**
- **unique**
- **or**
- **set**
- **use**

撰寫SQL指令(續)

使用SQL時通常是以一個敘述或指令(Statement)來描述，並以分號(;)作為結束符號。

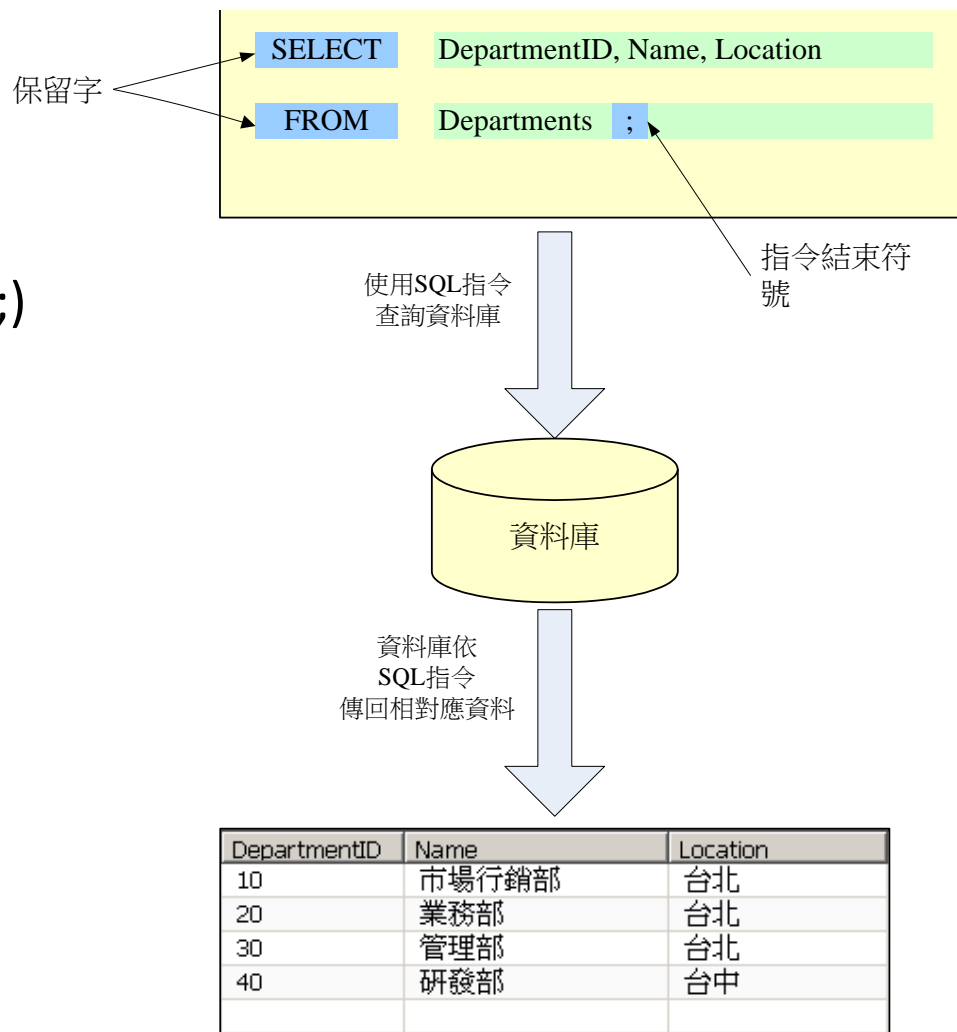


圖 3.1 SQL 敘述/指令

3.2 如何編輯及執行 SQL 指令

- 操作方式列舉如下：
 - A. ODBC Viewer (可連接任何ODBC連線資料庫)
 - B. MS Access 查詢精靈 (使用 QBE)
 - C. SQL Server
 - D. MySQL Query Browser
 - E. DB Visualizer
 - F. Oracle SQL Developer
 - G. phpmyadmin

例如：我們想要選取部門資料表(Departments)中的部門編號(DepartmentID)及部門名稱(Name)，則下達如下的

SQL指令即可：

```
SELECT DepartmentID, Name FROM Departments;
```

3.3 使用SELECT選取資料

- 最基本的SELECT指令是由SELECT子句(SELECT Clause)和FROM子句(FROM Clause)所組成的，其指令格式如下：

SELECT 欄位列表

FROM 資料表來源列表

使用SELECT選取資料(續)

- **SELECT**的部分設定/列舉要選取的資料表欄位名稱，而**FROM**的部分則指定資料表來源名稱。
- 例如：我們想要選取部門資料表(**Departments**)中的部門編號(**DepartmentID**)及部門名稱(**Name**)，則下達如下的SQL指令即可：
 - `SELECT DepartmentID, Name
FROM departments;`
- 執行結果如下圖：

Server: 127.0.0.1 Database: orders

Structure SQL Search Query Export Import Operations Privilege

Show query box

✓ Showing rows 0 - 3 (4 total, Query took 0.0005 sec)

```
SELECT DepartmentID, Name
FROM departments
LIMIT 0, 30
```

Profiling [Inline] [Edit] [Explain SC

Show : Start row: Number of rows: Headers every rows

Sort by key:

+ Options

				DepartmentID	Name
<input type="checkbox"/>	Edit	Copy	Delete	10	市場行銷部
<input type="checkbox"/>	Edit	Copy	Delete	20	業務部
<input type="checkbox"/>	Edit	Copy	Delete	30	管理部
<input type="checkbox"/>	Edit	Copy	Delete	40	研發部

Check All With selected: Change Delete Export

Show : Start row: Number of rows: Headers every rows

使用SELECT選取資料(續)

- 由於我們時常會想要選取資料表中的所有欄位的資料，此時我們可以使用 **SELECT *** 來選取所有的欄位。其指令格式如下：

- **【SELECT指令格式二】**

```
SELECT *  
FROM 資料表來源名稱
```


使用SELECT選取資料(續)

- 我們想要選取部門資料表(Departments)中的**所有欄位**，則下達如下的SQL指令即可：
 - `SELECT *`
`FROM departments;`

使用SELECT選取資料(續)

- 設定在 **SELECT** 子句中的欄位名稱可以在空白之後加入欄位的別名(Alias)或利用 **AS** 來設定別名(建議使用 **AS** 而不要使用空白)。其SQL指令格式如下：
 - **【SELECT指令格式三】**
SELECT 欄位名稱一 **AS** 別名一, 欄位名稱二 **AS** 別名二, ...
FROM 資料表來源名稱

使用SELECT選取資料(續)

- 如將 DepartmentID 欄位命名其別名為 Department，則下達指令如下：
 - SELECT DepartmentID AS '部門代號', '部門名稱'
 - FROM departments

使用SELECT選取資料(續)

- 有時候執行一個SQL指令可能會產生多筆相同的資料，如下達查詢部門位址列表的SQL指令：
 - `SELECT Location
FROM departments;`

使用SELECT選取資料(續)

- 此時我們會想要指定相同的資料只列出一次，則可於欄位列表之前加上 **DISTINCT** 這個識別字。即使用如下格式的SQL指令

```
SELECT DISTINCT 欄位名稱一 AS 別名一,  
        欄位名稱二 AS 別名二, .....  
FROM 資料表來源名稱
```

作業:



1. 請使用SQL指令查詢範例employees資料表中的
EmployeeID, DepartmentID, Name, Title 並將欄位名稱
命名為 員工證號，部門代號，姓名，職稱

使用WHERE設定條件選取資料

- 設定簡單的條件
 - 在SQL指令中，我們可以使用WHERE子句來過濾資料，只取出合乎特定條件的資料，即於WHERE子句中指定條件運算式。
 - 在WHERE子句中設定條件時需用到比較運算子(和一般的程式語言相似)。

比較運算子列表

比較運算子	意義	範例	範例說明
=	等於	性別 = '男'	性別等於 '男'
>	大於	數量 > 100	數量大於 100
>=	大於等於	數量 >= 150	數量大於等於 150
<	小於	數量 < 50	數量小於 50
<=	小於等於	數量 <= 25	數量小於等於 25
<>	不等於	數量 <> 1	數量不等於 1
!=	不等於(非SQL-92標準)	數量 != 1	數量不等於 0
IS NULL	為空值(NULL)	佣金 IS NULL	佣金為未知
IS NOT NULL	不為空值(NULL)	佣金 IS NOT NULL	佣金不為未知

註：文字用單引號括起來。

設定簡單的條件

- 我們可以使用的SQL指令格式如下：

```
SELECT 欄位名稱一 AS 別名一,  
        欄位名稱二 AS 別名二,.....
```

```
FROM 資料表來源名稱
```

```
WHERE 條件表示式
```

- 列出部門代號為10的資料：

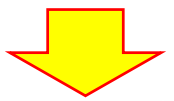
設定簡單的條件(續)

- 我們想列出所有部門代號為20(業務部門)的所有員工的資料，則可下達如下指令：

設定簡單的條件(IS NULL)

- 我們可以發現某些業務員的佣金未知，如上圖中的員工**1002**和**1008**。
- 假如我們想要選取佣金未知(即為**NULL**)的所有員工時應下達如下的指令：

```
SELECT *  
FROM employees  
WHERE Commission IS NULL;
```



	EmployeeID	DepartmentID	SupervisorID	ReportsTo	Name	EngName	SSN	Salary	Commission	Title	Sex 性別 (F/M)
<input type="checkbox"/> Edit Copy Delete	1002	20	NULL	NULL	陳廣明	Bradley	B121112233	1400000	NULL	副總經理	M
<input type="checkbox"/> Edit Copy Delete	1008	20	1005	1002	蘇韻涵	Maggie	E221432357	420000	NULL	業務助理	F
<input type="checkbox"/> Edit Copy Delete	1010	30	1002	1002	張文德	Tommy	P121322345	680000	NULL	總務經理	M
<input type="checkbox"/> Edit Copy Delete	1011	30	1010	1003	文師宣	Jesica	F221432312	380000	NULL	秘書	F
<input type="checkbox"/> Edit Copy Delete	1012	30	1010	1003	賴俊良	Eddie	F120056332	450000	NULL	資深工程師	M
<input type="checkbox"/> Edit Copy Delete	1013	10	1002	1002	何力宇	David	G121232312	670000	NULL	行銷經理	M
<input type="checkbox"/> Edit Copy Delete	1014	10	1013	1003	王明亮	John	H121002230	460000	NULL	行銷工程師	M
<input type="checkbox"/> Edit Copy Delete	1015	10	1013	NULL	顏子福	Luke	K121652456	450000	NULL	廣告設計師	M

設定簡單的條件(無效的指令)

- SELECT *
FROM employees
WHERE Commission IS NULL;

- 我們不可以使用如下的指令：(無效的指令)

```
SELECT *  
FROM employees  
WHERE Commission = NULL;
```

- 或

```
" 為兩個 ' 符號  
SELECT *  
FROM employees  
WHERE Commission = " ;
```

設定簡單的條件(續)

- 由於 NULL 值代表是未知、未定的資料，故任何一般比較的符號 (=、>、>=、<、<= ...)和NULL運算的結果皆為假(FALSE)。所以 WHERE Commission = NULL 是不正確的使用法。
- 我們若想要選出所有佣金不為NULL的員工資料，應下達如下的指令：

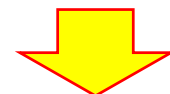
```
SELECT *  
FROM employees  
WHERE Commission IS NOT NULL;
```

```
SELECT *
FROM employees
WHERE Commission IS NOT NULL
LIMIT 0, 30
```

Profiling [[Inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP Code](#)]

Show: Start row: Number of rows: Headers every rows

Sort by key:



+ Options



	EmployeeID	DepartmentID	SupervisorID	ReportsTo	Name	EngName	SSN	Salary	Commission
<input type="checkbox"/> Edit Copy Delete	1001	20	1005	1002	張瑾 雲	Mary	A221112333	540000	120000
<input type="checkbox"/> Edit Copy Delete	1003	20	1005	1002	趙飛 燕	Maria	B221132237	560000	140000
<input type="checkbox"/> Edit Copy Delete	1004	20	1005	1002	林美 麗	Chris	C221112389	560000	130000
<input type="checkbox"/> Edit Copy Delete	1005	20	1002	1002	劉天 明	Mike	D121342354	700000	180000
<input type="checkbox"/> Edit Copy Delete	1006	20	1005	1005	黎國 明	Bill	D121443301	560000	120000
<input type="checkbox"/> Edit Copy Delete	1007	20	1005	1005	郭國 城	Steven	E121142302	560000	180000
<input type="checkbox"/> Edit Copy Delete	1009	20	1005	1005	孟玉 婷	Linda	P221114387	560000	130000

設定複雜的條件

- 在WHERE子句中可以設定由多個條件組合而成的複雜條件式。此時，我們可使用邏輯運算子(Logical Operator)來組合比較條件運算式，以設定出含多個條件的複雜條件式。
- SQL中可運用的邏輯運算子:

邏輯運算子	意義	範例	範例說明
AND	且；而且；和	性別 = '男' AND 年薪 <= 800000	性別等於 '男' 而且年 薪小於等於 80 萬
OR	或	職務='副理' OR 職務=' 經理'	職務為 '副理' 或者 職 務為 '經理'
NOT	非；不是	NOT(職務='副理')	職務不是 '副理'




設定複雜的條件(續)

- **AND**是用來設定「(條件一) **且** (條件二) 須同時成立」這樣子的條件的運算子。
- **OR** 是用來設定「(條件一) **或** (條件二) 須成立」這樣子的條件的運算子；換言之，用來設定「(條件一) 或者 (條件二) 任一條件成立」這樣子的條件的運算子。
- **NOT** 是用來設定否定的條件，即「**不是**(條件)」這樣子的條件的運算子。

設定複雜的條件(續)

- 我們想選取出職務為業務且稱呼為先生的男性經理之員工編號、姓名和年薪，則可下達如下的指令：

```
SELECT EmployeeID, Name, Salary, Title  
FROM employees  
WHERE Title = '業務' AND TitleOfCourtesy = '先生';
```

				EmployeeID	Name	Salary	Title
<input type="checkbox"/>	 Edit	 Copy	 Delete	1006	黎國明	560000	業務
<input type="checkbox"/>	 Edit	 Copy	 Delete	1007	郭國城	560000	業務

設定複雜的條件(續)

- 例如我們想找出職稱為業務或者佣金不為NULL的員工的員工編號、姓名、年薪、職稱和佣金資料。則可下達如下指令：
 - `SELECT EmployeeID, Name, Salary, Title, Commission
FROM employees
WHERE Title = '業務' OR Commission IS NOT NULL;`

設定複雜的條件(續)

- 例如我們想找出非職稱為業務的員工資料，則可下達如下的指令：

```
SELECT *  
FROM employees  
WHERE NOT (Title = '業務');
```

←T→	EmployeeID	DepartmentID	SupervisorID	ReportsTo	Name	EngName	SSN	Salary	Commission	Title
<input type="checkbox"/> Edit Copy Delete	1002	20	NULL	NULL	陳廣明	Bradley	B121112233	1400000	NULL	副總經理
<input type="checkbox"/> Edit Copy Delete	1005	20	1002	1002	劉天明	Mike	D121342354	700000	180000	業務經理
<input type="checkbox"/> Edit Copy Delete	1008	20	1005	1002	蘇韻涵	Maggie	E221432357	420000	NULL	業務助理
<input type="checkbox"/> Edit Copy Delete	1010	30	1002	1002	張文德	Tommy	P121322345	680000	NULL	總務經理
<input type="checkbox"/> Edit Copy Delete	1011	30	1010	1003	文師宣	Jesica	F221432312	380000	NULL	秘書
<input type="checkbox"/> Edit Copy Delete	1012	30	1010	1003	賴俊良	Eddie	F120056332	450000	NULL	資深工程師
<input type="checkbox"/> Edit Copy Delete	1013	10	1002	1002	何力宇	David	G121232312	670000	NULL	行銷經理
<input type="checkbox"/> Edit Copy Delete	1014	10	1013	1003	王明亮	John	H121002230	460000	NULL	行銷工程師
<input type="checkbox"/> Edit Copy Delete	1015	10	1013	NULL	顏子福	Luke	K121652456	450000	NULL	廣告設計師

設定複雜的條件(續)

- 當然我們想找出非業務的員工資料，也可以下達如下的指令：

```
SELECT *  
FROM employees  
WHERE Title <> '業務';
```

設定模糊的條件與限制範圍

- 在**WHERE**子句中設定條件時，我們可以指定模糊條件和限制資料範圍。此時即需要使用到模糊條件。如：

LIKE

BETWEEN ... AND ...

IN

表 3.5 邏輯運算子列表

邏輯運算子	意義	範例	範例說明
LIKE	如；像	職務 LIKE '%經理'	職務如樣式xx經理
BETWEEN... AND	介於	年薪 BETWEEN 500000 AND 800000	年薪介於500000 ~800000 之間
IN	在於...之中	選項 IN ('A','B','C')	選項在 ('A','B','C')之中

*LIKE 中可用的萬用字元：%表示任意個字元，_表示單一個任何字元。

*BETWEEN...AND 中的啓始值必須小於等於結束值。

設定模糊的條件與限制範圍(續)

- 例如我們想找出所有員工的職稱中含有‘經理’兩個字的員工(職稱以經理字眼結尾者)，則可下達如下的模糊指令：

```
SELECT *  
FROM employees  
WHERE Title LIKE '%經理';
```

設定模糊的條件與限制範圍(續)

- 練習: 想找出姓氏為 '張' 的所有員工(以張字開頭的員工姓名) ,

```
SELECT *  
FROM Employees
```

```
WHERE Name _____
```

設定模糊的條件與限制範圍(續)

- 我們想找出員工的英文字為 'M___' (三個底線)，即M開頭共計有四個字元的英文姓名，則可下達如下的指令：

```
SELECT *  
FROM employees  
WHERE EngName LIKE 'M___';
```

設定模糊的條件與限制範圍(續)

- 例如我們想找出員工的英文字含有 'ar' 這兩字元的員工，則可下達如下的指令：
 - `SELECT *`
`FROM employees`
`WHERE EngName LIKE '%ar%';`

設定模糊的條件與限制範圍(續)

- 如我們想找出所有員工的年薪介於420000～540000之間的所有員工，則此時可以利用**BETWEEN...AND**運算，即可下達如下的指令：

- `SELECT *`

- `FROM employees`

- `WHERE Salary BETWEEN 420000 AND 540000;`

設定模糊的條件與限制範圍(續)

- 也可以：

```
SELECT *
```

```
FROM employees
```

```
WHERE Salary >= 420000 AND Salary <= 540000;
```

設定模糊的條件與限制範圍(續)

- 在某些場合裡，我們會想用列舉的方式選取資料，例如：部門代號為 **10,20,30** 的資料，或如某一代碼為 'A', 'B', 'C' 其中之一。此時，我們即可使用 **IN** 運算子於條件之中。
- 如我們想列出員工代號為 **1002** 及 **1005** 的部屬列表(員工編號、姓名、上司員工編號)，即可下達如下的指令：

```
SELECT EmployeeID AS '員工編號', Name AS '姓名',
```

```
SupervisorID AS '上司的員工編號 '
```

```
FROM employees
```

```
WHERE SupervisorID IN (1002, 1005);
```

設定模糊的條件與限制範圍(續)

- 我們想列出英文姓名為 'Mary'、'Marua'、'Mike' 和 'Linda' 等人的資料，則可下達如下的指令：
 - ```
SELECT *
FROM employees
WHERE EngName IN ('Mary', 'Marua', 'Mike', 'Linda');
```



## 作業

2. 請使用SQL指令查詢範例 employees資料表中員工的  
姓氏為‘張’的員工之 員工ID、姓名、職務(Title)、薪水(Salary)  
且 薪水要大於六十萬
3. 請使用SQL指令查詢範例 employees資料表中員工的  
英文名為‘M開頭’的員工之 員工ID、姓名、職務(Title)、  
薪水(Salary)且 薪水要介於十~二百萬之間的經理

# 進行資料排序

- 關聯式資料庫中的資料是由多列資料所組成的，而這些資料列的儲存順序並不具有特殊的意義。
- 一般而言，資料列的順序就是依資料存入(新增)至資料庫時的先後次序。

# 進行資料排序(續)

- 若要對選取出來的資料列進行排序，那麼我們可以用**ORDER BY**子句來設定排序的方式。此時我們可以使用如下格式的**SQL**指令：

```
SELECT 欄位名稱一 AS 別名一,
 欄位名稱二 AS 別名二, ...
```

```
FROM 資料表來源名稱
```

```
WHERE 條件表示式
```

```
ORDER BY 排序欄位名稱一 [ASC|DESC],
 排序欄位名稱二 [ASC|DESC], ...
```

## 進行資料排序(續)

- 例如我們想列出所有的部門資料，而且依部門代號由大到小排序，則可下達如下的指令：
  - `SELECT *`  
`FROM departments`  
`ORDER BY DepartmentID DESC;`

## 進行資料排序(續)

- 我們想列出員工的部分資料(部門代號、員工編號、姓名、英文名字)，而列出時依照部門代號由小到大及員工編號由小到大的方式排序。則可下達如下的指令：

```
SELECT DepartmentID, EmployeeID, Name, EngName
```

```
FROM employees
```

```
ORDER BY DepartmentID, EmployeeID
```

File Edit View Bookmarks Tools Help



```
SELECT DepartmentID, EmployeeID, Name, EngName
FROM Employees
ORDER BY DepartmentID, EmployeeID
```

Transaction



TabSheet 1

| Departm... | Empl... | Name | EngName |
|------------|---------|------|---------|
| 10         | 1013    | 何力宇  | David   |
| 10         | 1014    | 王明亮  | John    |
| 10         | 1015    | 顏子福  | Luke    |
| 20         | 1001    | 張瑾雯  | Mary    |
| 20         | 1002    | 陳廣明  | Bradley |
| 20         | 1003    | 趙飛燕  | Maria   |
| 20         | 1004    | 林美麗  | Chris   |
| 20         | 1005    | 劉天明  | Mike    |
| 20         | 1006    | 黎國明  | Bill    |
| 20         | 1007    | 郭國城  | Steven  |
| 20         | 1008    | 蘇韻涵  | Maggie  |
| 20         | 1009    | 孟玉婷  | Linda   |
| 30         | 1010    | 張文德  | Tommy   |
| 30         | 1011    | 文師宣  | Jesica  |
| 30         | 1012    | 賴俊良  | Eddie   |

## 進行資料排序(續)

- 我們想要依部門代號由大到小(主排序)及員工編號由小到大(次要排序)，則可將上述指令更改為如下即可：

```
SELECT DepartmentID, EmployeeID, Name, EngName
```

```
FROM employees
```

```
ORDER BY DepartmentID DESC, EmployeeID ASC
```

| DepartmentID | Employee... | Name | EngName |
|--------------|-------------|------|---------|
| 30           | 1010        | 張文德  | Tommy   |
| 30           | 1011        | 文師宣  | Jesica  |
| 30           | 1012        | 賴俊良  | Eddie   |
| 20           | 1001        | 張瑾雯  | Mary    |
| 20           | 1002        | 陳廣明  | Bradley |
| 20           | 1003        | 趙飛燕  | Maria   |
| 20           | 1004        | 林美麗  | Chris   |
| 20           | 1005        | 劉天明  | Mike    |
| 20           | 1006        | 黎國明  | Bill    |
| 20           | 1007        | 郭國城  | Steven  |
| 20           | 1008        | 蘇韻涵  | Maggie  |
| 20           | 1009        | 孟玉婷  | Linda   |
| 10           | 1013        | 何力宇  | David   |
| 10           | 1014        | 王明亮  | John    |
| 10           | 1015        | 顏子福  | Luke    |



## 進行資料排序(續)

- 對於欲排序的欄位名稱，我們可以直接標明其欄位編號(從1開始)，而無須列出欄位名稱。
- 例如上述的SQL指令可以簡寫成：

```
SELECT DepartmentID, EmployeeID, Name, EngName
```

```
FROM employees
```


```
ORDER BY 1 DESC, 2
```

# Order by 針對不同字元集的排序原則

Server: 127.0.0.1 » Database: orders

Structure SQL Search Query Export Import Operations Privileges Routines Events Triggers

| Table ▲                               | Action     |           |        |        |       |      |        | Rows ⓘ       | Type            | Collation                | Size         |
|---------------------------------------|------------|-----------|--------|--------|-------|------|--------|--------------|-----------------|--------------------------|--------------|
| <input type="checkbox"/> autonumber   | Browse     | Structure | Search | Insert | Empty | Drop | ~1     | InnoDB       | big5_chinese_ci | 16 K                     |              |
| <input type="checkbox"/> categories   | Browse     | Structure | Search | Insert | Empty | Drop | ~7     | InnoDB       | big5_chinese_ci | 272 K                    |              |
| <input type="checkbox"/> customers    | Browse     | Structure | Search | Insert | Empty | Drop | ~92    | InnoDB       | big5_chinese_ci | 16 K                     |              |
| <input type="checkbox"/> departments  | Browse     | Structure | Search | Insert | Empty | Drop | ~4     | InnoDB       | big5_chinese_ci | 16 K                     |              |
| <input type="checkbox"/> dept         | Browse     | Structure | Search | Insert |       | Drop | ~0 ⓘ   | View         | ---             |                          |              |
| <input type="checkbox"/> emp          | Browse     | Structure | Search | Insert |       | Drop | ~0 ⓘ   | View         | ---             |                          |              |
| <input type="checkbox"/> employees    | Browse     | Structure | Search | Insert | Empty | Drop | ~14    | InnoDB       | big5_chinese_ci | 1.5 M                    |              |
| <input type="checkbox"/> orderdetails | Browse     | Structure | Search | Insert | Empty | Drop | ~1,886 | InnoDB       | big5_chinese_ci | 240 K                    |              |
| <input type="checkbox"/> orders       | Browse     | Structure | Search | Insert | Empty | Drop | ~915   | InnoDB       | big5_chinese_ci | 176 K                    |              |
| <input type="checkbox"/> postal       | Browse     | Structure | Search | Insert | Empty | Drop | ~369   | InnoDB       | big5_chinese_ci | 16 K                     |              |
| <input type="checkbox"/> products     | Browse     | Structure | Search | Insert | Empty | Drop | ~77    | InnoDB       | big5_chinese_ci | 48 K                     |              |
| <input type="checkbox"/> shippers     | Browse     | Structure | Search | Insert | Empty | Drop | ~3     | InnoDB       | big5_chinese_ci | 16 K                     |              |
| <input type="checkbox"/> suppliers    | Browse     | Structure | Search | Insert | Empty | Drop | ~29    | InnoDB       | big5_chinese_ci | 16 K                     |              |
| <b>13 tables</b>                      | <b>Sum</b> |           |        |        |       |      |        | <b>3,397</b> | <b>InnoDB</b>   | <b>latin1_swedish_ci</b> | <b>2.3 M</b> |



# 查看 mysql 字元集

SHOW CHARACTER SET

\_ci : 不分大小寫  
case in-senesitive  
\_cs : 區分大小寫  
case senesitive

字元的最大byte

| Charset | Description                 | Default collation | Maxlen |
|---------|-----------------------------|-------------------|--------|
| big5    | Big5 Traditional Chinese    | big5_chinese_ci   | 2      |
| dec8    | DEC West European           | dec8_swedish_ci   | 1      |
| cp850   | DOS West European           | cp850_general_ci  | 1      |
| hp8     | HP West European            | hp8_english_ci    | 1      |
| koi8r   | KOI8-R Relcom Russian       | koi8r_general_ci  | 1      |
| latin1  | cp1252 West European        | latin1_swedish_ci | 1      |
| latin2  | ISO 8859-2 Central European | latin2_general_ci | 1      |
| swe7    | 7bit Swedish                | swe7_swedish_ci   | 1      |
| ascii   | US ASCII                    | ascii_general_ci  | 1      |
| ujis    | EUC-JP Japanese             | ujis_japanese_ci  | 3      |
| sjis    | Shift-JIS Japanese          | sjis_japanese_ci  | 2      |
| hebrew  | ISO 8859-8 Hebrew           | hebrew_general_ci | 1      |
| tis620  | TIS620 Thai                 | tis620_thai_ci    | 1      |
| euckr   | EUC-KR Korean               | euckr_korean_ci   | 2      |
| koi8u   | KOI8-U Ukrainian            | koi8u_general_ci  | 1      |
| gb2312  | GB2312 Simplified Chinese   | gb2312_chinese_ci | 2      |

- 英文字母: 每一個字母都有編碼，A=65、B=66、...
- 根據字元編碼排序稱為: **binary collation**
- 但其他語言，例如中文，如果依照編碼排序，違反人類直覺
- 中文排序在 `big5_chinese_ci` 是依照 筆畫順序

**SELECT Title FROM employees ORDER BY Title;**

| Title ▲ |      |
|---------|------|
| 行銷工程師   | 6 劃  |
| 行銷經理    |      |
| 秘書      | 10 劃 |
| 副總經理    | 11 劃 |
| 業務      |      |
| 業務      |      |
| 業務      |      |
| 業務      | 13 劃 |
| 業務      |      |
| 業務      |      |
| 資深工程師   | 13 劃 |
| 業務助理    |      |
| 業務經理    |      |
| 廣告設計師   |      |
| 總務經理    |      |

# 改變排序原則 (改成 拼音順序)

```
SELECT Title
FROM employees
ORDER BY convert(`employees`.`Title` using gbk) DESC
```

```
SELECT Title
FROM employees
ORDER BY CONVERT(`employees`.`Title` USING utf8)
```

+ Options

Title

資深工程師

行銷經理

行銷工程師

總務經理

秘書

業務經理

業務助理

業務

業務

業務

業務

業務

業務

廣告設計師

副總經理

SELECT TITLE FROM EMPLOYEES ORDER BY **CAST(CONVERT(TITLE USING BIG5) AS BINARY)**;

| Title ▲ |      |
|---------|------|
| 行銷工程師   | 6 劃  |
| 行銷經理    |      |
| 秘書      | 10 劃 |
| 副總經理    | 11 劃 |
| 業務      | 13 劃 |
| 業務      |      |
| 業務      |      |
| 業務      |      |
| 業務      |      |
| 資深工程師   | 13 劃 |
| 業務助理    |      |
| 業務經理    |      |
| 廣告設計師   |      |
| 總務經理    |      |

| Title |
|-------|
| 行銷工程師 |
| 行銷經理  |
| 秘書    |
| 副總經理  |
| 業務    |
| 業務    |
| 業務    |
| 業務    |
| 業務    |
| 業務    |
| 業務助理  |
| 業務經理  |
| 資深工程師 |
| 廣告設計師 |
| 總務經理  |



## 進行資料排序(續)

- 對於欲排序的欄位名稱，我們可以直接標明其欄位編號(從1開始)，而無須列出欄位名稱。
- 例如上述的SQL指令可以簡寫成：
  - ```
SELECT DepartmentID, EmployeeID, Name, EngName  
FROM Employees  
ORDER BY 1 DESC, 2
```

作業

4.請使用SQL指令查詢範例資料庫中部門編號為20之員工資料，並以員工編號依序由小到大列表。