

SQL 語法教學

在 MySQL 使用 DML 資料處理語言

課程大綱

- 關聯式資料庫邏輯結構
- **SQL Server Management Studio 操作說明**
- **SELECT 基本句型**
- 資料彙總統計
- 以 **JOIN** 結合多個資料表
- 活用子查詢 (**Subquery**)
- 以 **Insert** 、**Update** 、**Delete** 修改資料
- 善用索引加快查詢效能

關聯式資料庫邏輯結構

- 資料以橫列直欄的方式組織於二維表格 (Table) 之中，各資料表 (Table) 存放現實世界中的實體或概念上認定存在的東西，例如：學生資料表、班級資料表、員工資料表。
- 每一直欄稱為欄位 (Field) 。
- 每一橫列稱為記錄 (Record) 。
- 每個資料表都各有其主鍵 (Primary Key, PK) 。
- 必要時，以某個欄位為外鍵 (Foreign Key, FK) 關聯到另一資料表的主鍵以獲得進一步的相關資料。

關聯式資料庫邏輯結構

- 每一直欄稱為欄位 (Field) 。

CityID	CityName
TP	台北
TC	台中
KS	高雄

EmplD	LastName	FirstName	CtryID	Extension	LastMod
integer	longstring	varchar(20)	char(2)	char(6)	longstring
101	Wang	Angle	TP	x19891	\HR\KarID
102	Chien	Wolfgang	TC	x19433	\HR\KarID
103	Martin	Jose	TP	x21467	\HR\AmyL

關聯式資料庫邏輯結構

- 每一橫列稱為記錄 (Record) 。

<i>CityID</i>	<i>CityName</i>
TP	台北
TC	台中
KS	高雄

<i>EmplD</i>	<i>LastName</i>	<i>FirstName</i>	<i>CtryID</i>	<i>Extension</i>	<i>LastMod</i>
integer	longstring	varchar(20)	char(2)	char(6)	longstring
101	Wang	Angle	TP	x19891	\HR\KarID
102	Chien	Wolfgang	TC	x19433	\HR\KarID
103	Martin	Jose	TP	x21467	\HR\AmyL

關聯式資料庫邏輯結構

- 每個資料表都各有其主鍵 (Primary Key , PK) 。
- 必要時，以某個欄位為外鍵 (Foreign Key , FK) 關聯到另一資料表的主鍵以獲得進一步的相關資料。

PK	CityID	CityName
	TP	台北
	TC	台中
	KS	高雄

PK

PK

FK

FK

EmpID	LastName	FirstName	CtryID	Extension	LastMod
integer	longstring	varchar(20)	char(2)	char(6)	longstring
101	Wang	Angle	TP	x19891	\HR\KarID
102	Chien	Wolfgang	TC	x19433	\HR\KarID
103	Martin	Jose	TP	x21467	\HR\AmyL

SELECT 基本句型

利用 SELECT 查詢資料

- **SELECT 敘述基本語法**
- **指定欄位清單**
- **資料排序**
- **篩選資料**
 - WHERE子句句型 – 比較型
 - WHERE子句句型 – 樣式比對型
 - WHERE子句句型 – 區間型
 - WHERE子句句型 – 列舉型
- **格式化結果集**

SELECT 敘述基本語法

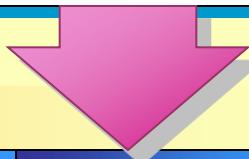
- **<select_list>**: 以逗號條列各個欄位
- 在 **FROM** 子句指定資料表名稱
- 以 **WHERE** 指定篩選欄位
- 以 **ORDER BY** 指定排序欄位

基本語法

```
SELECT [ALL | DISTINCT] <select_list>
  FROM {<table_source>} [,...n]
 WHERE <search_condition>
 ORDER BY <field name> [ASC | DESC] [,...n]
```

指定欄位清單

```
USE northwind;  
SELECT employeeid, lastname, firstname, title  
FROM employees;
```



<i>employeeid</i>	<i>lastname</i>	<i>firstname</i>	<i>title</i>
1	Davolio	Nancy	Sales Representative
2	Fuller	Andrew	Vice President, Sales
3	Leverling	Janet	Sales Representative
4	Peacock	Margaret	Sales Representative
5	Buchanan	Steven	Sales Manager
6	Suyama	Michael	Sales Representative
7	King	Robert	Sales Representative
8	Callahan	Laura	Inside Sales Coordinator
9	Dodsworth	Anne	Sales Representative

資料排序

```
USE northwind;  
SELECT productid, productname, categoryid, unitprice  
FROM products  
ORDER BY categoryid, unitprice DESC;
```



<i>productid</i>	<i>productname</i>	<i>categoryid</i>	<i>unitprice</i>
38	Cote de Blaye	1	263.5000
43	Ipoh Coffee	1	46.0000
2	Chang	1	19.0000
...
63	Vegie-spread	2	43.9000
8	Northwoods Cranberry Sauce	2	40.0000
61	Sirop d'éable	2	28.5000
...

練習: 基本SELECT句型



◆ 篩選資料

- **WHERE子句句型 – 比較型**
- **WHERE子句句型 – 樣式比對型**
- **WHERE子句句型 – 區間型**
- **WHERE子句句型 – 列舉型**

WHERE子句句型 – 比較型（一）

```
USE northwind;  
SELECT employeeid, lastname, firstname, title  
FROM employees  
WHERE employeeid = 5;
```



<i>employeeid</i>	<i>lastname</i>	<i>firstname</i>	<i>title</i>
5	Buchanan	Steven	Sales Manager

WHERE子句句型 – 比較型 (二)

Example 1

```
USE northwind;  
SELECT lastname, city  
FROM employees  
WHERE country = 'USA';
```

lastname

city

Davolio

Seattle

Fuller

Tacoma

Leverling

Kirkland

Peacock

Redmond

Callahan

Seattle

WHERE子句句型 – 樣式比對型

```
USE northwind;  
SELECT companyname  
FROM customers  
WHERE companyname LIKE '%Restaurant';
```



companyname

GROSELLA-Restaurante

Lonesome Pine Restaurant

Tortuga Restaurante

運用邏輯運算元

```
USE northwind;
SELECT productid, productname, supplierid, unitprice
FROM products
WHERE (productname LIKE 'T%' OR productid = 46)
AND (unitprice > 16.00) ;
```



<i>productid</i>	<i>productname</i>	<i>supplierid</i>	<i>unitprice</i>
14	Tofu	6	23.25
29	Thüringer Rostbratwurst	12	123.79
62	Tarte au sucre	29	49.3

WHERE子句句型 – 區間型

```
USE northwind;  
SELECT productname, unitprice  
FROM products  
WHERE unitprice BETWEEN 10 AND 20;
```



<i>productname</i>	<i>unitprice</i>
Chai	18
Chang	19
Aniseed Syrup	10
Genen Shouyu	15.5
Pavlova	17.45
Sir Rodney's Scones	10
...	...

WHERE子句句型 – 列舉型

```
USE northwind;  
SELECT companyname, country  
FROM suppliers  
WHERE country IN ('Japan', 'Italy');
```



<i>companyname</i>	<i>country</i>
Tokyo Traders	Japan
Mayumi's	Japan
Formaggi Fortini s.r.l.	Italy
Pasta Buttini s.r.l.	Italy

NULL = Unknown 「未知」

```
USE northwind;  
SELECT companynname, fax  
FROM suppliers  
WHERE fax IS NULL;
```



<i>companynname</i>	<i>fax</i>
Exotic Liquids	NULL
New Orleans Cajun Delights	NULL
Tokyo Traders	NULL
Cooperativa de Quesos 'Las Cabras'	NULL
...	...

◆ 格式化結果集

- 排序資料
- **DISTINCT**
- 變更欄位名稱
- 利用靜態文字修飾資料

排序資料

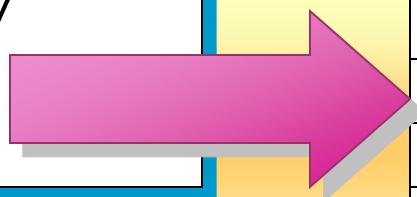
```
USE northwind;
SELECT productid, productname, categoryid, unitprice
FROM products
ORDER BY categoryid, unitprice DESC;
```



<i>productid</i>	<i>productname</i>	<i>categoryid</i>	<i>unitprice</i>
38	Cote de Blaye	1	263.5000
43	Ipoh Coffee	1	46.0000
2	Chang	1	19.0000
...
63	Vegie-spread	2	43.9000
8	Northwoods Cranberry Sauce	2	40.0000
61	Sirop d'éable	2	28.5000
...

DISTINCT，若記錄內容完全相同，只留一筆

```
USE northwind;  
SELECT DISTINCT country  
FROM suppliers  
ORDER BY country;
```



country
Australia
Brazil
Canada
Denmark
Finland
France
Germany
Italy
Japan
Netherlands
Norway
Singapore
Spain
Sweden
UK
USA

變更欄位名稱

```
USE northwind;
```

```
SELECT firstname AS First, lastname AS Last  
FROM employees;
```



<i>First</i>	<i>Last</i>	<i>Employee ID:</i>
Nancy	Davolio	1
Andrew	Fuller	2
Janet	Leverling	3
Margaret	Peacock	4
Steven	Buchanan	5
Michael	Suyama	6
Robert	King	7
Laura	Callahan	8
Anne	Dodsworth	9

利用靜態文字修飾資料

```
USE northwind;
SELECT Firstname, Lastname,
       Concat('Identification number: ', 
              Convert(employeeid, varchar(2))) as ID
FROM employees;
```



<i>Firstname</i>	<i>Lastname</i>	<i>ID</i>
Nancy	Davolio	Identification Number: 1
Andrew	Fuller	Identification Number: 2
Janet	Leverling	Identification Number: 3
Margaret	Peacock	Identification Number: 4
Steven	Buchanan	Identification Number: 5
Michael	Suyama	Identification Number: 6
Robert	King	Identification Number: 7
Laura	Callahan	Identification Number: 8
Anne	Dodsworth	Identification Number: 9

練習：各項資料篩選句型與資料格式化



資料彙總統計

資料彙總統計

- **LIMIT N**，列出排頭的資料
- 使用彙總函數
- 彙總函數與 Null 值
- 使用 **GROUP BY** 子句
 - 以 **HAVING**進行彙總之後的第二階段條件過濾
 - 以 **GROUP BY** 配合 **ROLLUP** 進行多層次統計

TOP N，列出排頭的資料

- 僅傳回結果集的前 N 筆記錄
- 排列次序由 ORDER BY 子句決定
- Update, Delete 亦有 LIMIT 子句

範例

```
USE northwind;
SELECT orderid, productid, quantity
FROM [order details]
ORDER BY quantity DESC
LIMIT 5;
```

◆ 使用彙總函數

函數名稱	功能描述
AVG	計算平均值
COUNT	有資料的共有幾筆
COUNT (*)	一共有多少筆 (有Null值的記錄也算進去)
MAX	傳回最大值
MIN	傳回最小值
SUM	計算總和
STDEV	計算標準差
VAR	計算變異數

彙總函數與 Null 值

- 絝多數的彙總函數均排除 Null，不列入計算
- COUNT(*) 例外，有 Null 值的資料仍然計入一筆

範例 1

```
USE northwind;
SELECT COUNT (*)
FROM employees;
```

範例 2

```
USE northwind;
SELECT COUNT(reportsto)
FROM employees;
```

使用 GROUP BY 子句

```
USE northwind;
SELECT productid, orderid
,quantity
FROM orderhist;
```

<i>productid</i>	<i>orderid</i>	<i>quantity</i>
1	1	5
1	1	10
2	1	10
2	2	25
3	1	15
3	2	30

```
USE northwind;
SELECT productid
, SUM(quantity) AS total_quantity
FROM orderhist
GROUP BY productid;
```

<i>productid</i>	<i>total_quantity</i>
1	15
2	35
3	45

<i>productid</i>	<i>total_quantity</i>
2	35

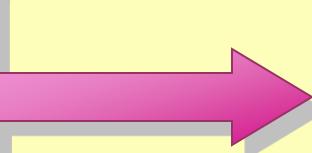
```
USE northwind;
SELECT productid
, SUM(quantity) AS total_quantity
FROM orderhist
WHERE productid = 2
GROUP BY productid;
```

以 HAVING 進行彙總之後的第二階段條件過濾

```
USE northwind;
SELECT productid, orderid
     ,quantity
  FROM orderhist;
```

```
USE northwind;
SELECT productid, SUM(quantity)
       AS total_quantity
  FROM orderhist
 GROUP BY productid
 HAVING SUM(quantity)>=30;
```

<i>productid</i>	<i>orderid</i>	<i>quantity</i>
1	1	5
1	1	10
2	1	10
2	2	25
3	1	15
3	2	30



<i>productid</i>	<i>total_quantity</i>
2	35
3	45

以 GROUP BY 配合 ROLLUP 進行多層次統計

```
USE northwind;
```

```
SELECT productid, orderid, SUM(quantity) AS total_quantity
FROM `Order Details`
GROUP BY productid, orderid
WITH ROLLUP
ORDER BY productid, orderid;
```

<i>productid</i>	<i>orderid</i>	<i>total_quantity</i>
NULL	NULL	95
1	NULL	15
1	1	5
1	2	10
2	NULL	35
2	1	10
2	2	25
3	NULL	45
3	1	15
3	2	30

練習: 資料彙總統計



JOIN
結合多個資料表

◆ JOIN 結合多個資料表

- 資料表別名
- Join 語法說明
 - Inner Join
 - Outer Join
 - Cross Join
 - Self Joining
- 以 UNION 上下整併資料表

資料表別名

■ 範例 1 (未使用資料表別名)

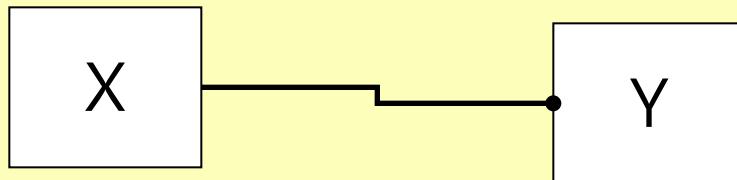
```
SELECT buyer_name, sales.buyer_id, qty
FROM buyers INNER JOIN sales
ON buyers.buyer_id = sales.buyer_id;
```

■ 範例 2 (使用資料表別名)

```
SELECT buyer_name, s.buyer_id, qty
FROM buyers AS b INNER JOIN sales AS s
ON b.buyer_id = s.buyer_id;
```

Join 語法說明

- 以 **JOIN** 註明另一個資料表
- 再以 **ON** 指定結合條件
- 通常利用主鍵與外鍵欄位指定 **ON** 條件
- 欄位名稱重複時，須加註資料表名稱（或別名）



```
SELECT X.欄位, Y.欄位, ...
FROM X JOIN Y ON Y.FK = X.PK
```

Inner Join

```
SELECT buyer_name, sales.buyer_id, qty
FROM buyers INNER JOIN sales
ON buyers.buyer_id = sales.buyer_id
```

buyers

buyer_name	buyer_id
Adam Barr	1
Sean Chai	2
Eva Corets	3
Erin O'Melia	4

sales

buyer_id	prod_id	qty
1	2	15
1	3	5
4	1	37
3	5	11
4	2	1003

Result

buyer_name	buyer_id	qty
Adam Barr	1	15
Adam Barr	1	5
Erin O'Melia	4	37
Eva Corets	3	11
Erin O'Melia	4	1003

Outer Join

```
SELECT buyer_name, sales.buyer_id, qty
  FROM buyers LEFT OUTER JOIN sales
    ON buyers.buyer_id = sales.buyer_id
```

buyers		sales		
buyer_name	buyer_id	buyer_id	prod_id	qty
Adam Barr	1	1	2	15
Sean Chai	2	1	3	5
Eva Corets	3	4	1	37
Erin O'Melia	4	3	5	11
Result				
buyer_name	buyer_id	qty		
Adam Barr	1	15		
Adam Barr	1	5		
Erin O'Melia	4	37		
Eva Corets	3	11		
Erin O'Melia	4	1003		
Sean Chai	NULL	NULL		

結合更多的資料表

```
SELECT buyer_name, prod_name, qty
  FROM buyers
 INNER JOIN sales
    ON buyers.buyer_id = sales.buyer_id
 INNER JOIN produce
    ON sales.prod_id = produce.prod_id
```

buyers		sales			produce	
buyer_id	buyer_name	buyer_id	prod_id	qty	prod_id	prod_name
1	Adam Barr	1	2	15	1	Apples
2	Sean Chai	1	3	5	2	Pears
3	Eva Corets	3	1	37	3	Oranges
4	Erin O'Melia	4	5	11	4	Bananas
Result		2	2	1003	5	Peaches
buyer_name		prod_name		qty		
Erin O'Melia		Apples		37		
Adam Barr		Pears		15		
Erin O'Melia		Pears		1003		
Adam Barr		Oranges		5		
Eva Corets		Peaches		11		

練習: Join 與 Union



活用子查詢

活用子查詢 (Subquery)

- 子查詢 (Subquery) 簡介
- 以子查詢的結果為基礎資料表
- 將「子查詢」視為運算式
- 子查詢以資料表別名參用外部查詢的內容
- 子查詢聯合 IN 子句的寫法

子查詢 (Subquery) 簡介

- 在一個查詢指令中，內嵌另一個查詢指令。
- 內嵌的子查詢必須前後以括號框住
- 如此，可將複雜的查詢分解成數個小查詢步驟
- 以某個查詢的結果為基礎，解決進一步的問題
- 例如，條列出最後一天的訂單：

```
SELECT *
  FROM Orders
 WHERE OrderDate =
    (SELECT MAX(OrderDate) FROM Orders)
```

以子查詢的結果為基礎資料表

- 將「子查詢的結果」視同是一個資料表。
- 必須加上資料表別名。

```
USE northwind;  
  
SELECT T.orderid, T.customerid  
FROM ( SELECT orderid, customerid  
      FROM orders ) AS T;
```

將「子查詢」視為運算式

- 凡是允許運算式的地方，都可以利用括號內嵌「子查詢」。

```
USE pubs;

SELECT title, price
      ,( SELECT AVG(price) FROM titles) AS average
      ,price-(SELECT AVG(price) FROM titles) AS difference
  FROM titles
 WHERE type='popular_comp' ;
```

子查詢以資料表別名參用外部查詢的內容

1

外部查詢將欄位值傳入內部查詢

```
USE northwind;
SELECT orderid, customerid
FROM orders AS or1
WHERE 20 < (SELECT quantity
              FROM [order details] AS od
              WHERE od.orderid = or1.orderid
              AND od.productid = 23);
```

2

內部查詢以傳入值進行資料查詢

3

內部查詢傳回結果到外部查詢

4

針對外部查詢的每一筆都重複1~3
的步驟

子查詢聯合 IN 子句的寫法

- 有時候，想查 A 資料表時，得靠 B 資料表幫忙。
- 先從訂單明細查出哪些訂單編號「有訂購 23 號商品而且訂超過 20 個」，再以此為基礎，列出這些編號的訂單資料。

```
select * from orders
where OrderID in
(
    select OrderID
    from [Order Details]
    where ProductID = 23 and Quantity > 20
)
```

練習: 子查詢



修改資料

Insert、Update、Delete

修改資料

■ **Insert** 語法說明

- Insert 新增單筆資料
- Insert ... Select
- Select ... Into

■ **Delete** 語法說明

■ **Update** 語法說明

新增單筆資料

- 將各個欄位以逗號分隔條列
- 再利用 VALUES —— 對應各欄位值
- 資料必須符合資料型態並且通過檢查規則（例如：主鍵鍵值不可重複），資料才可進入資料表。
- 自動編號欄位以及計算型欄位不能指定欄位值。

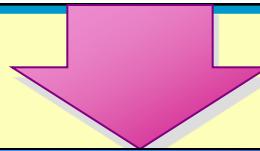
```
USE northwind;
INSERT customers
  (customerid, companyname, contactname, contacttitle
  , address, city, region, postalcode, country, phone
  , fax)
VALUES ('PECOF', 'Pecos Coffee Company', 'Michael Dunn'
        , 'Owner', '1900 Oak Street', 'Vancouver', 'BC'
        , 'V3F 2K1', 'Canada', '(604) 555-3392'
        , '(604) 555-7293');
```

若略過特定欄位的值，將代入預設值或Null

新增資料時，沒有指定「電話」的欄位內容...

```
USE northwind;
INSERT shippers (companyname)
VALUES ('Fitch & Mather');
```

```
USE northwind;
SELECT *
FROM shippers
WHERE companyname = 'Fitch & Mather';
```



shipperid	companyname	phone
37	Fitch & Mather	Null

INSERT...SELECT 敘述

- 先利用 **SELECT** 敘述，從「來源資料表」提出資料，然後將之匯入到「目的資料表」。
- 「目的資料表」必須是現有存在的資料表。
- 必須配合「目的資料表」，依序提供資料型態相容的欄位值。

```
USE northwind;
INSERT customers
SELECT Concat(substring(firstname, 1, 3),
              substring(lastname, 1, 2))
      , lastname, firstname, title, address, city
      , region, postalcode, country, homephone, NULL
  FROM employees;
```

◆ 刪除資料

- **DELETE** 語法說明
- **TRUNCATE TABLE**

DELETE 敘述

- 利用 **DELETE** 敘述可刪除符合 **WHERE** 條件的資料。
- 若是沒寫 **WHERE**，等同於刪除整份資料表的內容

```
USE northwind;  
DELETE orders  
WHERE shippeddate = '2010-04-01';
```

TRUNCATE TABLE 敘述

- **TRUNCATE TABLE** 敘述會刪除整份資料表的內容。
- 資料表的內容全部清空，資料表的資料結構仍在。
- 由於 **TRUNCATE TABLE** 敘述刪除資料時並不會進行 **LOG** 記錄，因此將無法以 **ROLLBACK** 回復資料。

```
USE northwind;  
TRUNCATE TABLE orders;
```

以 Update 敘述修改資料

- 以 WHERE 子句指定要修改的是哪些記錄
- 以 SET 設定新欄位值
- 新欄位值必須通過各項 Constraint 的資料檢驗
- 建議加上開放式並行檢查，將舊值也融入 WHERE，以確保沒有與其他使用者發生資料衝突。

```
USE northwind;
UPDATE products
SET unitprice = (unitprice * 1.1)
WHERE CategoryID = 1;
```

練習: 新增、修改、刪除資料



單元複習與討論

