

SQL 結構化查詢語言

第二堂 SQL 的基礎與基本操作

聯成電腦

張明泰 mtchang.tw@gmail.com

複習及補充

- 上次上課內容review



MS-SQL 與 MySQL

- MS-SQL 與 MySQL 為目前國內較常見被使用的關連式資料庫管理系統(RDBMS)。
 - MS-SQL 是微軟公司出品的。
目前最新版為 sql 2005
分成4種版本 dev, ent, std, express
 - MySQL 是 MySQL AB 公司出的。
目前最新版為 mysql 5.16
分成兩種主要版本 MAXDB 及 MySQL server
-
-

MSSQL express

- <http://www.microsoft.com/downloads/details.aspx?FamilyID=220549b5-0b07-4448-8848-dcc397514b41&DisplayLang=en>
- 請參考...



MySQL

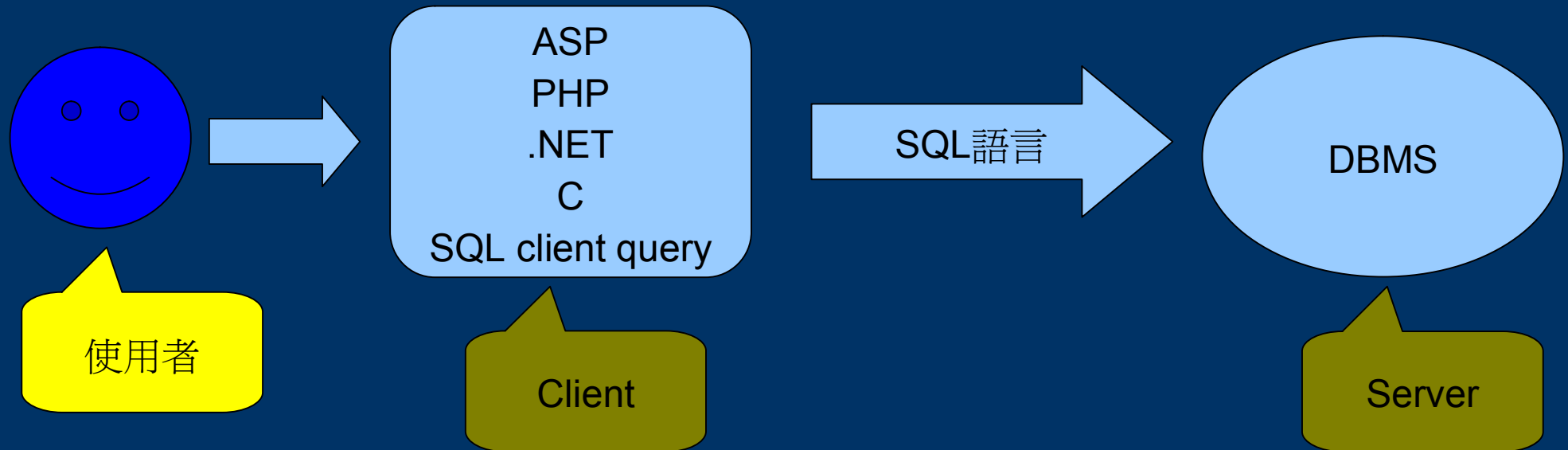
- MySQL Administrator → 管理server
- MySQL query → 傳送sql query
- MySQL server → DBMS
- MySQL migration → 轉移工具



MSSQL

- SQL Server Management studio
- 是管裡Server的工具
- 也是Client工具
- 也有轉移工具
- **SQL Server 2005 功能比較**
<http://www.microsoft.com/taiwan/sql/prodinfo/features/compare-features.msp>

SQL 架構



SQL Server 2005 中包括下列系統資料庫

- **master** 資料庫記錄 SQL Server 執行個體的所有系統層級資訊。
 - **msdb** 資料庫由 SQL Server Agent 用於排程警示和作業。
 - **model** 資料庫用來當作 SQL Server 執行個體上建立之所有資料庫的範本。對 **model** 資料庫進行的修改 (例如，資料庫大小、定序、復原模式和其他資料庫選項) 會套用到之後建立的任何資料庫。
 - 資源資料庫是一個唯讀的資料庫，其中包含 SQL Server 2005 擁有的系統物件。系統物件實際上會保存在 **Resource** 資料庫中，但邏輯上會出現在每個資料庫的 **sys** 結構描述中。
 - **tempdb** 資料庫是保存暫存物件或中繼結果集的工作空間。
-
-

設計資料庫

- 需先瞭解您要建立模型的企業功能。對於資料庫概念和要用來代表企業功能的各種功能，也需要加以瞭解。
- 請確定您已針對企業模型，精確設計好資料庫，因為在實作之後，若要大幅變更資料庫的設計，是很耗費時間的。一個設計精良的資料庫執行起來也較順暢。

開發資料庫計劃

- 在規劃資料庫時，不管其大小與複雜度為何，都請使用下列基本步驟：
 - 收集資訊
 - 識別物件
 - 建立物件模型
 - 識別每個物件的資訊類型
 - 識別物件之間的關聯
-
-

線上交易處理與決策支援的比較

- 資料庫應用程式可歸類成兩個主要的類別：
 - 線上交易處理 (OLTP) 及決策支援
 - 線上交易處理 (OLTP)
 - 良好的資料放置
 - 短的交易可減少長期的鎖定，並改善並行
 - 線上備份
 - 資料庫的高度正規化
 - 最少或沒有歷程記錄或彙總資料。
 - 小心地使用索引
 - 較佳的硬體組態可處理許多同時使用者以及 OLTP 系統所需的快速反應時間
-
-

線上交易處理與決策支援的比較

- 決策支援
 - 決策支援資料庫應用程式適用於不會變更資料的資料查詢。例如，公司可根據日期、銷售區域或產品，定期地建立摘要的銷售資料，並將此資訊儲存於個別的資料庫內，以供進階的管理分析使用。
 - 決策支援設計考量
 - 製作較多索引
 - 將資料庫解除正規化
 - 使用星形或雪花架構來整理資料庫內的資料。
-
-

正規化

- 資料庫邏輯設計，包括資料表和資料表之間的關係，是將關聯式資料庫最佳化的核心。良好的邏輯資料庫設計可替最佳的資料庫和應用程式效能奠定根基。不良的邏輯資料庫設計則會妨礙整個系統的效能。
 - 資料表必須擁有識別碼。
 - 資料表應該只儲存單一實體類型的資料。
 - 資料表應避免可為 Null 值的資料行。
 - 資料表不可有重複的數值或資料行。
-
-

統一上課時的連線方式

- 各位的帳號及密碼
 - 使用XSQL tools連到我的MsSQL server 或使用SQL Management Studio 連入
 - 主機：192.168.3.xx
 - 帳號：axx
 - 密碼：axx
 - 資料庫：axx
-
-

使用SQL Client工具匯入sampledb

- <http://mtchang.blogdns.org/wiki/index.php/Sql-sampledb>
- 請參考此文匯入範例資料庫





休息一下(10 min)

提醒：請記得上網到討論區註冊

第二堂

SQL的基礎與基本操作



一般識別碼的規則

- 1. 第一個字元必須是以下任一項：
 - Unicode Standard 3.2 中所定義的字母。Unicode 的字母定義包括從 a 到 z 以及從 A 到 Z 的拉丁字元，還有其他語系中的字母字元。
 - 底線 (_)、@ 符號或數字符號 (#)。
 - 識別碼開頭的某些符號在 SQL Server 中有特殊意義。以 @ 符號開頭的一般識別碼代表本機變數或參數，而且不能做為任何其他類型之物件的名稱。開頭為 # 符號的識別碼代表暫存資料表或程序。開頭為兩個 ## 符號的識別碼代表全域暫存物件。雖然 # 符號或兩個 ## 符號字元可做為其他類型之物件的名稱開頭，但是不建議此用法。
 - 部分 Transact-SQL 功能的名稱會以兩個 @@ 符號為開頭。為了避免與這些功能產生混淆，不應該使用以 @@ 為開頭的名稱。
- 2. 可包含的後續字元如下：
 - Unicode Standard 3.2 中所定義的字母。
 - 其他 Basic Latin 或其他國家（地區）字母中的十進位數字。
 - @ 符號、錢幣符號 (\$)、數字符號或底線。
- 3. 識別碼必須是 Transact-SQL 保留字。SQL Server 保留大、小寫版本的保留字。
- 4. 不允許內嵌的空格或特殊字元。
- 5. 不允許補充字元。

分隔識別碼

- 括在雙引號 (") 或方括號 ([]) 中的識別碼。符合識別碼格式規則的識別碼不一定要以符號分隔。例如：

```
SELECT *  
FROM [TableX]      --Delimiter is  
optional.  
WHERE [KeyCol] = 124 --  
Delimiter is optional.
```

分隔識別碼

- 不符合識別碼規則的識別碼在 Transact-SQL 陳述式中一定要以符號分隔。例如：

```
SELECT *  
FROM [My Table]    --Identifier contains a space and uses a  
reserved keyword.  
WHERE [order] = 10  --Identifier is a reserved keyword.
```

試著使用SQL

- 製作資料庫或表格時，使用CREATE的命令
- 表格需指定「資料型態」，如文字、數字等...
- 為了讓在表格內的資料正確我們可以用「限制(constraints)」的方式作限制。
- CREATE-->DCL

如何操作資料庫

- 登錄資料使用INSERT敘述-DML
 - 取出資料可以使用SELECT敘述-DML
 - SELECT有附加選項(option)，可以讓使用者變更從表格中取出的資料，將資料群組化或刪除資料。
 - 此種資料庫語言更嚴謹的區分成為三種：
資料定義語言(DDL)、資料操作語言(DML)、
資料控制語言(DCL)
-
-

SQL的類別(一)

- SQL 分為三大類別:DDL / DCL / DML
- DDL (Data Definition Language) : 定義資料庫物件使用的語法，常看到的關鍵字有：
 - Create : 建立資料庫的物件。
 - Alter : 變更資料庫的物件。
 - Drop : 刪除資料庫的物件。

SQL的類別(二)

- DCL (Data Control Language)：控制資料庫物件使用狀況的語法，常看到的關鍵字有：
 - Grant：賦予使用者使用物件的權限。
 - Revoke：取消使用者使用物件的權限。
 - Commit：Transaction 正常作業完成。
 - Rollback：Transaction 作業異常，異動的資料回復到 Transaction 開始的狀態。
-
-

SQL的類別(三)

- DML (Data Manipulation Language)：維護資料庫資料內容的語法，常看到的關鍵字有：
 - Insert：新增資料到 Table 中。
 - Update：更改 Table 中的資料。
 - Delete：刪除 Table 中的資料。
 - Select：選取資料庫中的資料。
-
-

製作資料庫

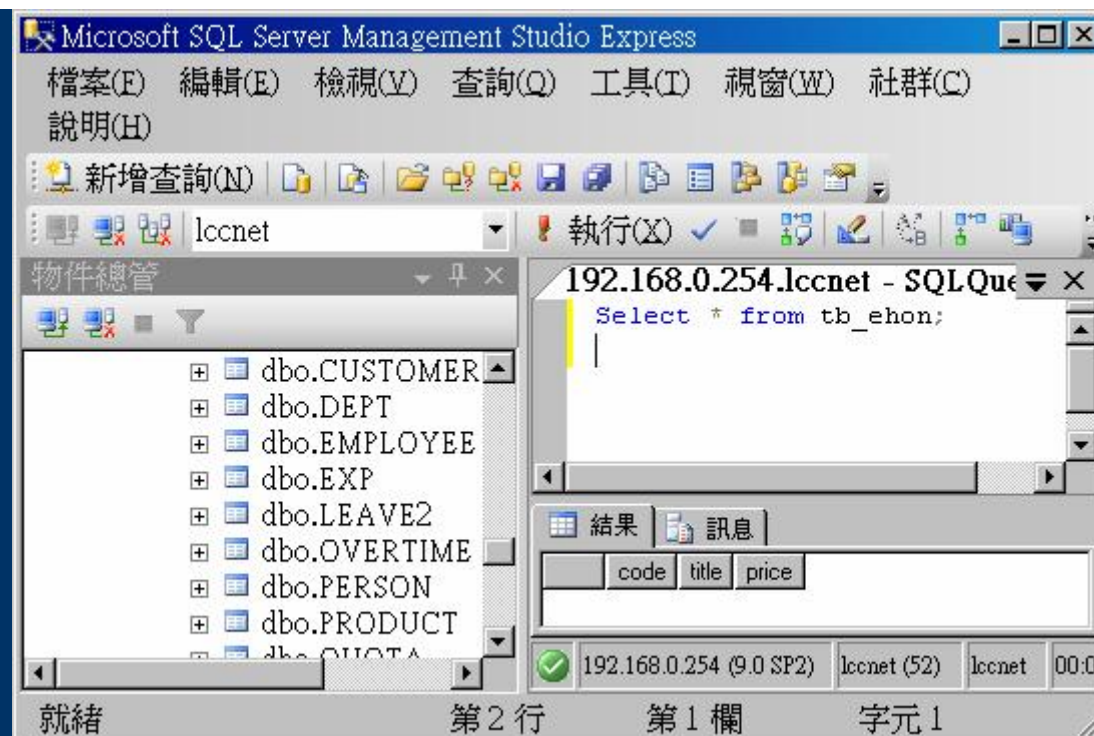
- 資料存在表格，表格存在資料庫。
- 語法：CREATE DATABASE *database_name*
[DATABASEPASSWORD 'database_password'
[ENCRYPTION {ON|OFF}]] [COLLATE
collation_name comparison_style
database password ::= identifier
- 範例：create database db_ehon;

製作表格

- 如果要製作一個有code、title、price這三列的表格，需定義：
 - 語法：create table 表格名(欄位1 資料型別,欄位2 資料型別,欄位3 資料型別.....);
 - 範例：`create table tb_ehon(code int,title varchar(30),price int);`
-
-

指定資料庫

- 語法：use 資料庫名;
- 範例：use lccnet;
- 說明：在使用資料前，需要先指定資料庫
- 顯示表格
- `Select * from tb_ehon;`
- 由於目前無資料，所以結果只顯示上方的欄位列。



限制(*constraint*)

- 為了讓登錄的資料落在正常的範圍而加上去的一些條件。
 - 範例：create table tb_ehon(code INT **UNIQUE**,title VARCHAR(30),price INT,**PRIMARY KEY**(title,price));
 - 局部範例：code **INT UNIQUE NOT NULL**
 - 說明：也可以對一欄位設定一個以上的限制。
-
-

限制(*constraint*)範例

The screenshot displays the Microsoft SQL Server Management Studio Express interface. The '物件總管' (Object Explorer) on the left shows the database structure for 'lccnet', including tables 'dbo.STUDENTS' and 'dbo.tb_ehon'. The 'tb_ehon' table is expanded, showing columns 'code (int, Null)', 'title (PK, varchar(30), 非 Null)', and 'price (PK, int, 非 Null)'. The 'SQL 查詢' (SQL Query) window on the right shows the following SQL code:

```
create table
tb_ehon(code INT UNIQUE,
title VARCHAR(30),
price INT, PRIMARY KEY(title, price));
```

The '訊息' (Messages) window below the query shows the message: '命令已順利完成。' (Command completed successfully.). The status bar at the bottom indicates the current position: '就緒' (Ready), '第 4 行' (Line 4), '第 14 欄' (Column 14), '字元 14' (Character 14), and 'INS' (Insert mode).

主要的限制種類

- 限制：意義
 - **PRIMARY KEY(主鍵)**：禁止資料重複、禁止NULL值的出現。
 - **UNIQUE**：資料不得重複。
 - **CHECK(條件式)**：事先準備條件式，禁止使用不符合條件的資料。
 - **NOT NULL**：資料不得為NULL。
 - **DEFAULT=值**：將指定的值設定為預設值。
-
-

休息一下(10 min)



提醒：請記得上網到討論區註冊



限制的範例：

- 範例：`use lccnet;`
`create table tb_height(id INT PRIMARY`
`KEY,name VARCHAR(20),height FLOAT);`
 - 說明：限制的結果並不會顯示出來，但當妳使用新增或是修改指令時，就會知道她的功用。
 - 觀看：`select * from tb_height;`
-
-

登錄資料

- 語法：insert into 表格名 (欄位1,欄位2,欄位3)
values (資料1,資料2,資料3);
 - 範例：insert into tb_ehon(code,title,price)
values (111,'SQL繪本',300);
 - 說明：欄位1對應資料1，如果資料有字串應用'(單引號)刮起來。
-
-

取出資料

- 語法：select 欄位 from 表格名稱(資料庫.表格名稱)
 - 範例：
`select * from tb_ehon;`
`select * from lccnet.dbo.tb_ehon;`
 - 說明：請確定前一個例子有執行。
-
-

取出資料-只取出一欄位

- 語法：select 欄位1,欄位2... from 表格名;
- 範例：
`select title from tb_ehon;`
`select title,price from tb_ehon;`
- 說明：只取出title此欄位的資料。
在欄位的選項裡，顯示會照這順序來顯示。
如果以*代表欄位，所表示的是全部欄位。

ORDER BY-遞增排序

- 指定欄為基準來排序資料並顯示。
 - 語法：select 欄位 from 表格 order by 指定排序的基準欄名 ascend;
 - 範例：`select * from tb_ehon order by price asc;`
 - 說明：以price欄位為基準做ASCEND遞增排序，可簡寫為ASC。
-
-

ORDER BY-遞減排序

- 指定欄為基準來排序資料並顯示。
 - 語法：select 欄位 from 表格 order by 指定排序的基準欄名 descend;
 - 範例：select * from tb_ehon order by price desc;
 - 說明：以price欄位為基準做DESCEND遞增排序，可簡寫為DESC。
-
-

比較複雜的排序

- 語法：select 欄位 from 表格 order by 基準欄位1,基準欄位2;
 - 範例：select * from tb_ehon order by price,code;
 - 說明：在此範例中，首先以price為基準作遞增排序，若值相同的話，則以id欄位為第二基準。
-
-

將資料群組化

- 使用GROUP BY的語法，如果欄位的資料相同時，把這些行的內容整理在一塊。通常會和合成函數一起使用。
 - 範例：`SELECT 班級座號, count(公假) as 公假累計 FROM RECORDS GROUP BY 班級座號;`
 - 說明：count為合成函數功能，是會傳會所指令欄位中存在的行數。
-
-

刪除重複的資料

- 這不是指真正的刪除，是在select時將指定欄位重複項去除。
 - 範例：**SELECT DISTINCT 班級座號as stu_no FROM RECORDS ;**
 - 說明：distinct 可以將指令欄位中重複的資料只顯示第一筆找出的資料。指令欄可以顯示一個上，但必須使用,(逗點)來區隔欄名。
-
-

變更欄位名來顯示

- 用不同的名稱顯示原來的欄位名稱。
- 範例：**SELECT 班級座號AS class_no,年月日,公假 FROM RECORDS;**
- 說明：班級座號這個欄位名稱被class_no取代了。



休息一下(10 min)



提醒：請記得上網到討論區註冊



整數型

- 資料型別(sql 99) 可使用資料範圍
使用方式(sql server)
- Integer 整數值 **INT**
- Smallint 比integer還要小的整數值
SMALLINT
- 詳細請見

[http://yes.nctu.edu.tw/SQL/DataType/Data
Type.htm](http://yes.nctu.edu.tw/SQL/DataType/Data
Type.htm)

實數型

- DECIMAL[(m[,n])] *1 使用者自訂精確度的小數值。以m表示整數位數，n表示小數點以下為數。 DECIMAL[(m[,n])]
- NUMERIC[(m[,n])] 同上 NUMERIC[(m[,n])]
- REAL 單精度浮點數 REAL
- FLOAT[(n)] 倍精度浮點數 FLOAT[(n)] *2
- DOUBLE PRECISION 倍精度浮點數
FLOAT(53)

• *1:[]的部分可以省略。 *2:n≤53。若n≤24和REAL同義。

字串型別(1)

- character[(n)]^{*3} 長度固定(n位元組以內)的字串
char[(n)]^{*3}
- character varying(n)^{*3} 長度不定(n位元組以內)的字串
varchar(n)^{*3}
- character large object 文章等大量的字串
TEXT
- ^{*3}:n=<8000。[]的部分可省略。此時n=1。

字串型別(2)

- National character[(n)]*4 長度固定(n字以內)的unicode字串

nchar[(n)]*4

- National character varying(n)*4 長度不定(n字以內)的unicode字串

nvarchar[(n)]*4

- National character large object 文章等大量的unicode字串 NTEXT

- *4:n=<4000。[]的部分可省略。此時n=1。

如何取出特定的資料

- 在select敘述中可以加入where條件來取得符合條件的的資料。
- 範例：**SELECT * FROM students WHERE 學號=911013;**

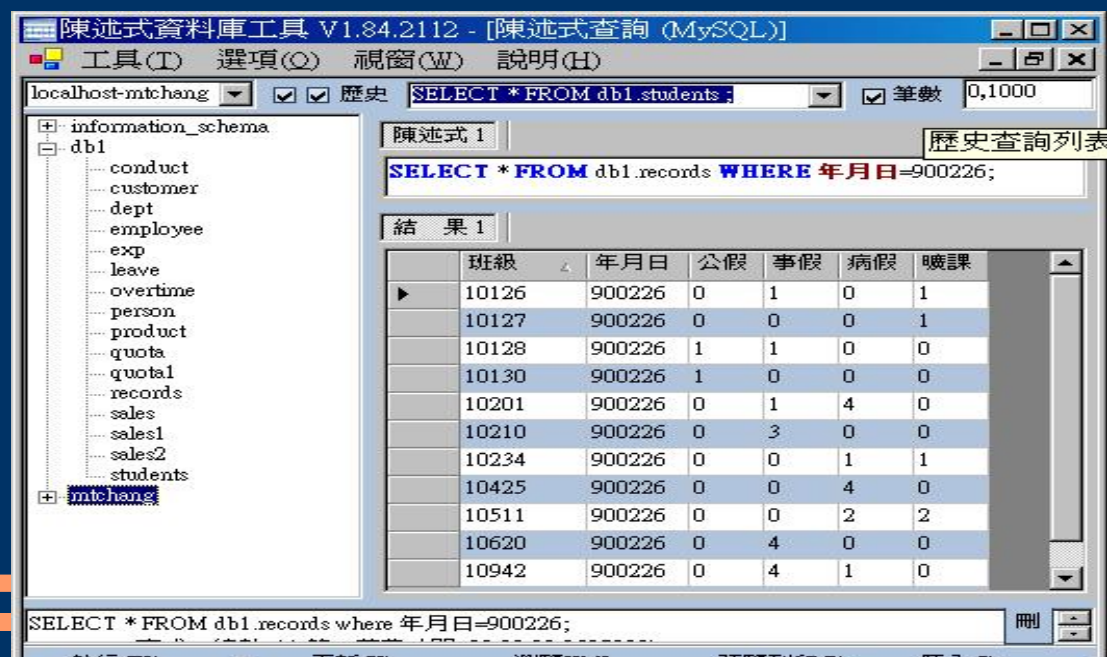
SELECT  FROM  WHERE

各種運算子

- 算術運算子(+ - x /)
- 比較運算子(> < = !=)
- 邏輯運算子(AND OR NOT)

取出符合條件的資料

- 範例：**SELECT * FROM RECORDS WHERE 年月日=900226;**
- 語法：select 欄位名 from 表格名 where 條件式;



把符合條件的資料排序後再取出

- 範例：**SELECT * FROM RECORDS
WHERE 年月日=900226 ORDER BY 班級座
號DESC;**
 - 練習1：取出沒有請過病假的學生，並依請假
次數由少到多排序。
-
-

在做數學運算時的運算子

- 「+」 $a+b$
 - 「-」 $a-b$
 - 「*」 $a*b$
 - 「/」 a/b
 - 「%」 $a\%b$ a除以b取餘數
 - 範例：SELECT 班級座號,(公假+事假+病假+曠課) AS sum_day FROM db1.records;
 - 統計每個學生的請假總數
-
-

運算子指定優先順序

- 「(運算式)/2」在這範例中，刮號中的運算式具有優先計算的權力

比較運算子

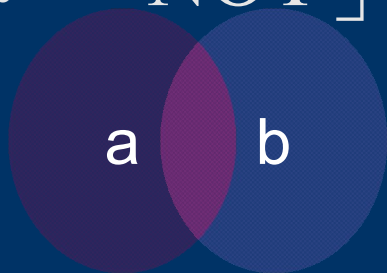
- 比較運算的結果會傳回ture或是false
 - 「=」 $a=b$ a等於b
 - 「<」 $a<b$ a小於b
 - 「<=」 $a<=b$ a小於等於b
 - 「>」 $a>b$ a大於b
 - 「>=」 $a>=b$ a大於等於b
 - 「<>」 $a<>b$ a不等於b
 - 「!=」 $a!=b$ a不等於b
-
-

條件式結果

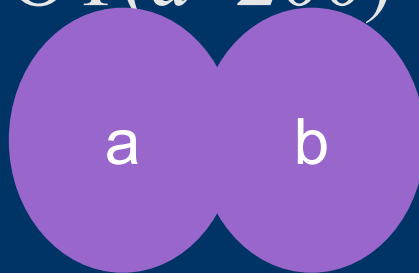
- 在經過運算後，產生的結果稱為條件式結果，一定只有ture或false兩種結果。
 - 範例：SELECT * FROM db1.records WHERE 病假 > 0;
 - 說明：列出(病假>0)的資料。ture->列出，false->不列出。
-
-

邏輯運算子

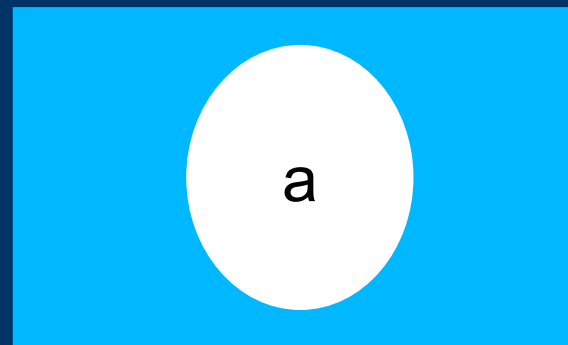
- 運用邏輯運算子可以組合多個條件式，表達更複雜的邏輯條件。
- 「AND」 且 $(a \geq 10) \text{ AND } (a \leq 20)$
- 「OR」 或 $(a = 5) \text{ OR } (a = 10)$
- 「NOT」 非 $\text{NOT}(a = 200)$



A and B



A OR B



NOT a

較複雜的運算式

- 範例：**SELECT * FROM RECORDS
WHERE 病假= 0 AND 公假=0 AND 事假=0 ;**
 - 說明：找出沒有請過病、公、事假的學生。
 - 練習：找出沒有請過事假、病假及曠課的學生，或是在900201以前請假的人。
-
-

操作文字的運算子(mysql不支援)

- 此功能會因為sql版本不同，而語法不同。
- 若要連結文字的話可以使用「||」運算子。
- 範例：`SELECT 姓名 || 家長 AS name FROM db1.students;`
- SQL server範例：`select 姓名+' '+家長 from dbo.STUDENTS;`

文字運算的執行結果(SQL server)

The screenshot displays the Microsoft SQL Server Management Studio Express interface. The '物件總管' (Object Explorer) on the left shows the database structure, including tables like dbo.CONDUCT, dbo.CUSTOMER, and a folder named '資料行' (Columns) with sub-items like '班級座號' and '年月日'. The central query window, titled '192.168.0.254.lccnet - SQLQuery2.sql*', contains the SQL query: `select 姓名+' '+家長 from dbo.STUDENTS;`. Below the query window, the '結果' (Results) tab shows the execution output as a table with 4 rows and 1 column. The status bar at the bottom indicates the query was successful ('已'), executed on 192.168.0.254 (9.0 SP2) using lccnet (52) in 00:00:00, returning 480 rows.

	(沒有資料行名稱)
1	王子穎 王世傑
2	王慧如 王文淵
3	王琇瑜 王進豐
4	朱勝真 朱水順

字串的比較

- 「like」運算子可以作字串比較，同常與底下符號合併使用作模糊的條件判斷。
- 「%」相當於任意字串 %明% 找出字串中有「明」的列。
- 「_」相當於任意一個字
- 範例：**SELECT * FROM STUDENTS
WHERE 姓名LIKE '吳%';**
- 練習：請找出家長姓名為「陳榮」開頭的資料。

*BETWEEN*運算子

- 使用 between~and 可以指定值的範圍
 - 範例：**SELECT * FROM STUDENTS
WHERE 學號 BETWEEN 913024 AND
913035;**
 - 練習：請使用 AND 寫成用 between 的描述。
-
-

IS NULL 運算子

- 用來檢查欄位的值是否為NULL
 - 欄位 IS NULL
 - 欄位 IS NOT NULL
 - 範例：**SELECT * FROM STUDENTS
WHERE 學號 IS NOT NULL;**
 - 範例：**SELECT * FROM STUDENTS
WHERE 學號 IS NULL;**
-
-

IN運算子

- IN用來取出()內指定的值的資料。
- 範例：**SELECT * FROM STUDENTS
WHERE 學號 IN (911002,911005);**
- 練習：請用OR或AND運算子達到一樣的功能。

指定件數取出

- 當我們的查詢一次列出很多，但是我們只要取其中列1到列10的話，可以用下列方式。
- 每個RDBMS都有其不同的作法
- MySQL範例：**SELECT * FROM STUDENTS
LIMIT 0,10;**
- SQL server、Access範例：
- **SELECT TOP 10 * FROM STUDENTS ;**
- **SELECT TOP 20 PERCENT * FROM
STUDENTS ;**
- PostgreSQL範例：

課後練習

- 請複習本日所教的指令用法。
- 請熟練每個範例。
- 參閱參考資料，加深印象。



參考資料

- <http://yes.nctu.edu.tw/SQL/index.htm>
 - <http://chensh.loxa.edu.tw/php/>
 - <http://ftp.tn.edu.tw/Study/SQL%ac%e3%b2%df/>
 - SQL99 手冊
-
-