SQL語法的使用

CREATE:建立資料庫及資料表

1. 建立資料庫

CREATE DATABASE [IF NOT EXISTS] 資料庫名稱

[DEFAULT] CHARACTER SET 字元集

[DEFAULT] COLLATE 連線校對

[ ]裡的內容可以不填，但是[IF NOT EXISTS]表示如果沒有該資料庫時才執行建立資料庫的動作

例：

CREATE DATABASE ‘class’

DEFAULT CHARACTER SET utf8

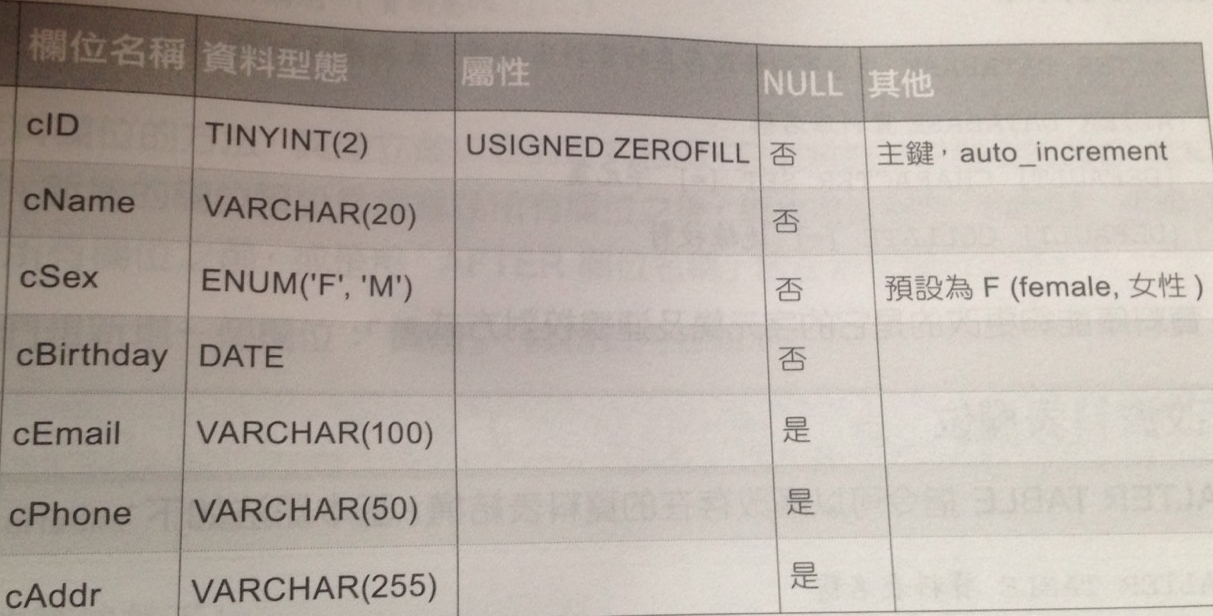
COLLATE utf8\_unicide\_ci;

1. 建立資料表

CREATE TABLE [IF NOT EXISTS] 資料表名稱

(欄位名稱 資料類別 [資料屬性] , 欄位名稱 資料類別 [資料屬性]……)

例：



CREATE TABLE `students` (

`cID` TINYINT(2) UNSIGNED ZEROFILL NOTNULL AUTO\_INCREMENT PRIMARY KEY,

`Cname` VARCHAR(20) CHARACTER SET utf8 collate utf8\_unicode\_ci NOT NULL,

……

)

ALTER:變更資料庫及資料表內容

修改資料庫-資料庫能變更的是它的字元集跟連線校對方式

ALTER DATABASE 資料庫名稱

[DEFAULT] CHARACTER SET 字元集

[DEFAULT] COLLATE 連線校對

修改資料表欄位

ALTER TABLE 資料表名稱

CHANGE 原欄位名稱 新欄位名稱 資料類別 [資料屬性];

例：

ALTER TABLE `students` CHANGE `cID` `cID` INT(2)

新增資料表欄位

ALTER TABLE 資料表名稱 [資料屬性]

ADD 欄位名稱 資料類別 [資料屬性],…)

[FIRST | AFTER 欄位名稱];

可以利用FIRST將欄位新增在所有欄位之前，或是用AFTER放在指定欄位之後。

例：

ALTER TABLE `students` ADD `cHabits` VARCHAR(255) CHARACTER SET utf8 COLLAR utf8\_unicode\_ci NULL;

刪除資料表欄位

ALTER TABLE 資料表名稱 DROP 欄位名稱;

例：

ALTER TABLE `students` DROP ` cHabits `;

查詢資料庫資料的內容

SELECT查詢資料

SELECT欄位名稱 FROM資料表名稱

例：

SELECT \* FROM `students`; (查詢students資料表裡的所有欄位)

AS 設定選取欄位別名

SELECT 欄位名稱 as 欄位別名 FROM 資料表名稱

例：

SELECT `cID` as座號,`cName`AS 姓名 FROM `students`;

原本 設定別名後 

SELECT DISTINCT:去除重複資料顯示一筆

SELECT DISTINCT 欄位名稱 FROM 資料表名稱;

例： 例如我們想知道全班學生共有幾種性別

SELECT DISTINCT `cSex` FROM `students`;

結果：



WHERE :設定篩選條件

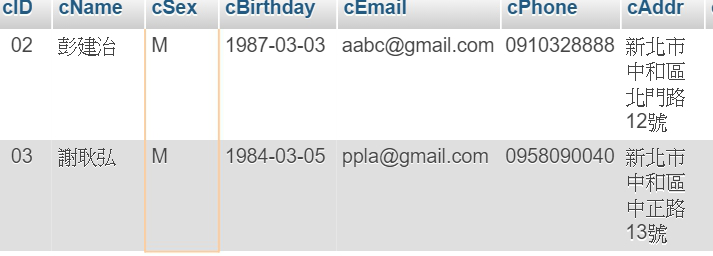
SELECT 欄位名稱 FROM 資料表 WHERE 條件敘述句

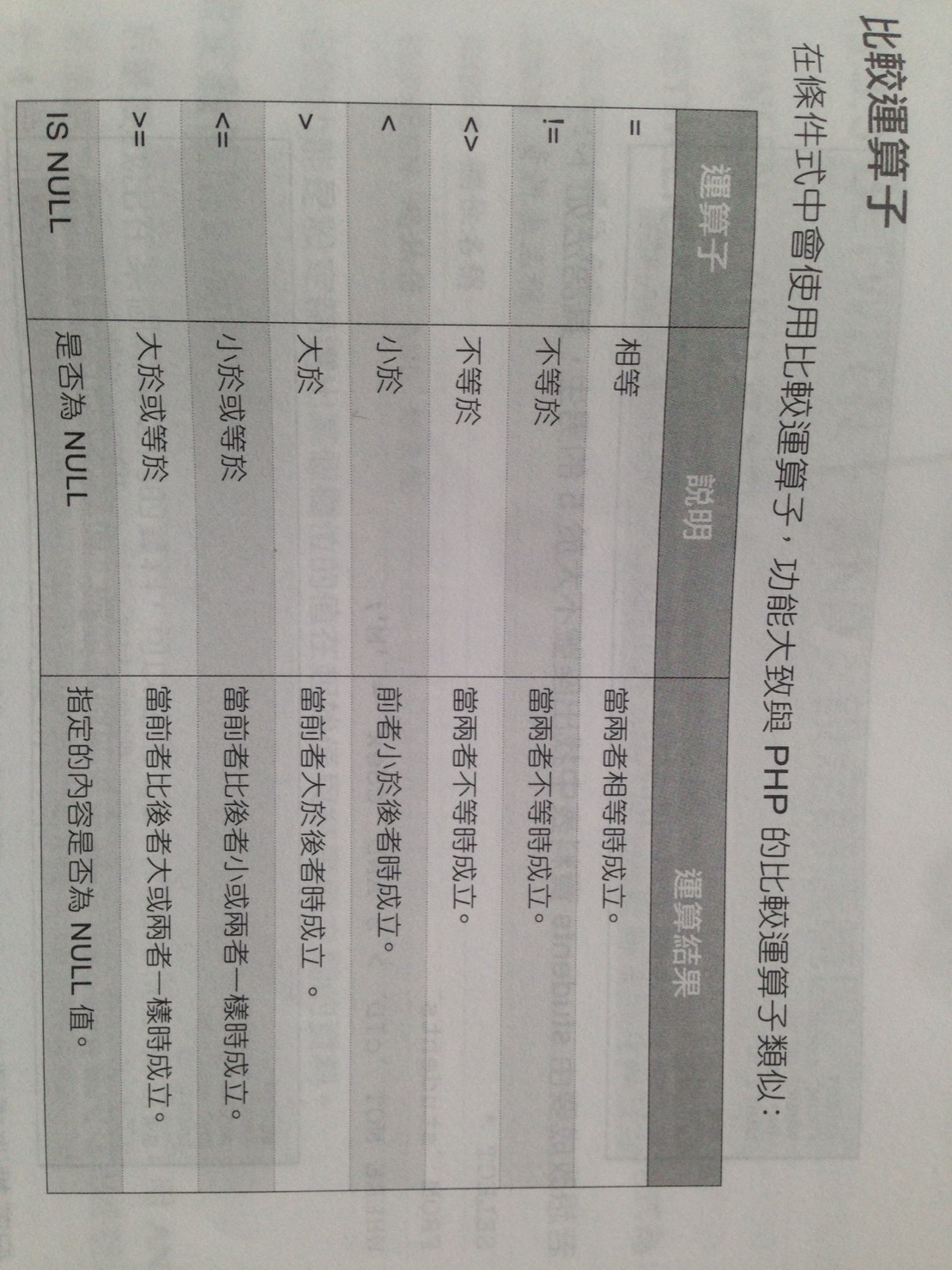
例： 從資料表挑選出所有男性的資料

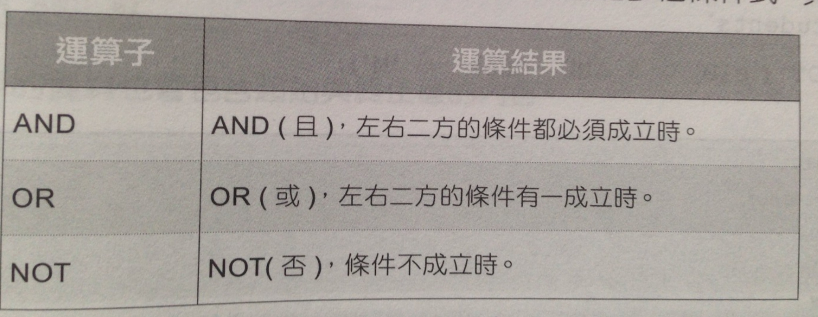
SELECT \* FROM students WHERE cSEX =”M”

(因為是屬於文字性質的資料型別，所以要用””將值框起來才能正確比對)

結果：







例：

SELECT \* FROM `students` WHERE `cID` > 5 AND `cSex`=”M”;

BETWEEN…AND:設定篩選範圍

SELECT 欄位名稱 FROM 資料表名稱

WHERE 欄位名稱 BETWEEN 起始值 AND 結束值

例：

SELECT \* FROM `students` WHERE `cID` BETWEEN 2 AND 4;

也可以拿來設定時間日期

SELECT \* FROM `students` WHERE `cBirthday` BETWEEN “1985-03-24” AND “1989-12-23”;

IN:指定多個篩選值

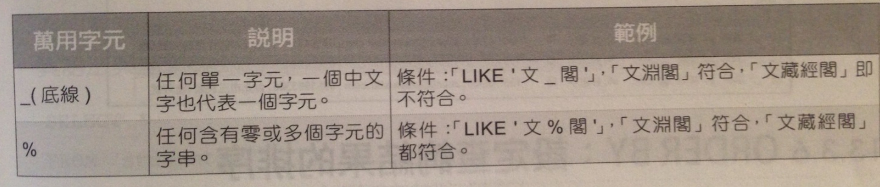
SELECT 欄位名稱 FROM 資料表名稱

WHERE 欄位名稱 IN(欄位值1,欄位值2…);

例：

SELECT \* FROM `students` WHERE `cID` IN (1,3,4);

LIKE:設定字串比對的篩選值



例：找尋電話號碼0918開頭的學生

SELECT \* FROM `students` WHERE `cPhone` LIKE `0918%`;

ORDER BY:設定查詢結果的排序

SELECT \* FROM 資料表名稱

ORDER BY 指定排序的欄位 排序方式

排序方式有兩種:

ASC:遞增排序-由小排到大，也是預設的排序方法

DESC:遞減排序-由大排到小

例：由年紀最輕的排到最大

SELECT \* FROM `students`

ORDER BY `cBirthday` DESC;

多欄位排序

ORDER BY 指定排序的欄位1 排序方式,指定排序的欄位2 排序方式…

以逗號區隔開

例：

SELECT \* FROM `students`

ORDER BY `cSex` ASC,`cBirthday` DESC,`;

顯示結果就會先以性別的值做排序，由於是文字型態 會以字母的排序作依據，若是性別相同會再依生日進行遞減排序

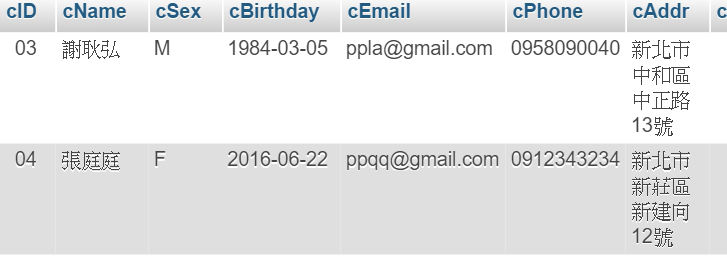
LIMTIT:設定查詢顯示的筆數

SELECT 欄位名稱 FROM 資料表名稱 LIMIT 開始顯示的筆數,顯示多少筆數

例：

[SELECT](http://localhost/phpMyAdmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) \* FROM `students` LIMIT 2,4

結果：



MYSQL常用函式

算術運算子-

+加法

-減法

\*乘法

/除法

例：要計算每個學生的標準體重，其公式為(身高-70)\*0.6

並顯示一欄計算目前體重與標準體重的差距

SELECT `cID` AS `座號`, cName` AS `姓名`, `cHeight` AS `身高`,`cWeight` AS `體重`,(`cHeight`-70)\*0.6 AS `標準體重`,

(`cWeight`-(`cHeight`-70)\*0.6) AS `差距` FROM students;

結果：



數學函式

ABS(數值):取絕對值

POW(基數,次方數):取次方值

SQRT(數值):取平方根

PI():取得圓周率，PI()=3.1415926535898

ROUND(數值[,小數位數]):取得四捨五入的值，可設定要取到小數第幾位，預設

是0

例：計算BMI值

SELECT cID AS '座號',cName AS '姓名',cHeight AS '身高',cWieght AS '體重', ROUND((cWieght/POW(cHeight/100,2)),2) AS 'BMI值' FROM students

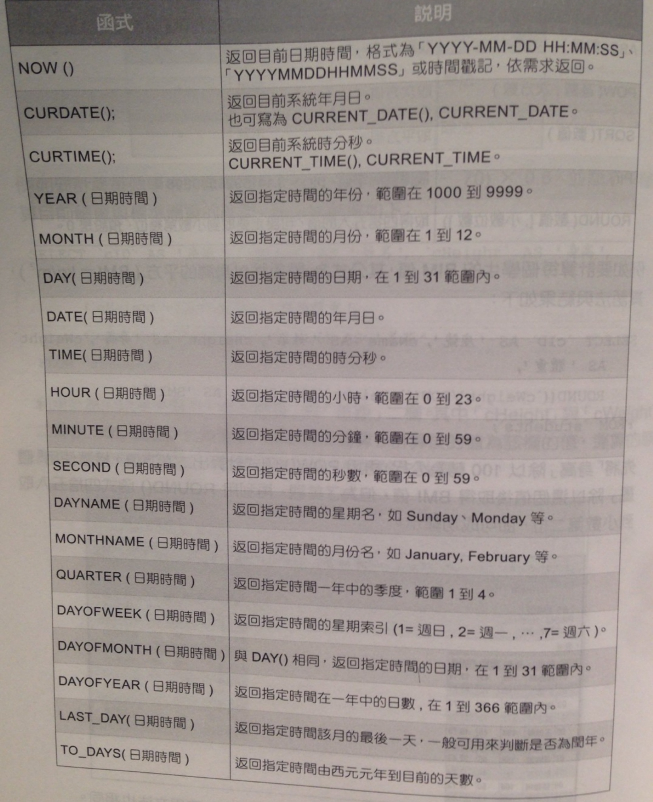
結果：



BMI公式：體重除以身高的平方(BMI=kg/m^2)

將身高除以100轉換成公尺

日期時間函式



例：顯示出生日期是星期幾

SELECT \*,dayname(cBirthday) AS ‘出生星期’FROM students

結果：



例：篩選出今年生日32歲的同學

SELECT \* FROM students WHERE YEAR(NOW())-YEAR(cBirthday)=32

統計函式

SUM():可以統計出選取欄位的總和

AVG():可以統計出選取欄位的平均值

COUNT():可以統計出選取欄位的出現次數

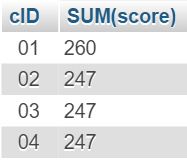
MAX() MIN():可以統計出選取欄位的最大或最小值

GROUP BY:(分組排列)選取一個欄位來做統計顯示

HAVING:GROUP BY的條件式

例：顯示每個學生的總分

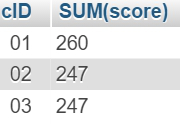
SELECT cID , SUM(score) FROM scorelist GROUP by cID



顯示每個學生的總分 以cID分類排列後 將相同的cID的score分數加總顯示

例：顯示每個學生的總分，座號1-3

SELECT cID , SUM(score) FROM scorelist GROUP by cID HAVING cID<=3



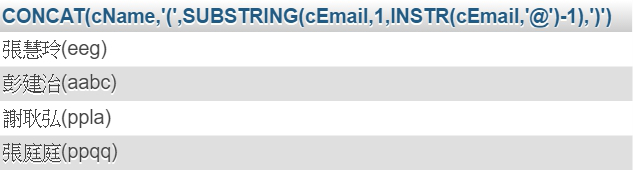
字串函式



例：電子郵件@之前的帳號當作英文名 同時顯示出來

SELECT CONCAT(cName,'(',SUBSTRING(cEmail,1,INSTR(cEmail,'@')-1),')') FROM students

結果：



新增 更新 刪除資料

INSERT:新增資料

INSERT [INTO] 資料表名稱[(欄位名稱1,欄位名稱2,…]

VALUES (值1,值2,值3…);

例：

INSERT INTO students (cName,cSex,cBirthday,cEmail,cPhone,cAddr,cHeight,cWieght)

VALUES('李伯恩','M','1981-03-03','okla@gmail.com','0921837953','台中市美村南路12號','165','87')

UPDATE:更新資料

UPDATE 資料表名稱 SET 欄位名稱1=值1 , 欄位名稱2=值2 ,… WHERE 條件式;

例：

UPDATE students SET cHeight=175,cWieght=67 WHERE cID=5

更新資料可以一次更動多筆資料的內容，所以WHERE後加上的條件式非常重要，為了精確的指出更新目標，最好用該資料的主鍵作為條件式的篩選值

DELETE:刪除資料

DELETE FROM 資料表名稱 WHERE 條件式

例：

DELETE FROM students WHERE cID>11

如此就會將cID大於11的刪除

多資料表的關聯查詢

結合資料表的查詢

SELECT 顯示欄位… FROM 資料表A ,資料表B

WHERE 資料表A.相關欄位=資料表B.相關欄位

例：

SELECT students.cID,students.cName,scorelist.score FROM students,scorelist WHERE students.cID=scorelist.cID AND scorelist.course='國文'

結果：



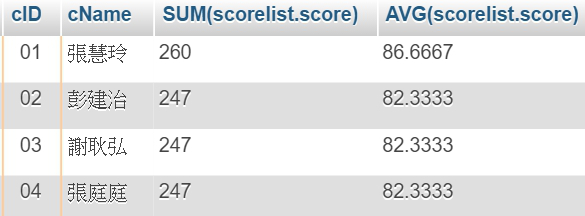
加入統計欄位

非統計欄位的欄位都必須加入GROUP BY的欄位並以『,』區隔

例：

[SELECT](http://localhost/phpMyAdmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) students.cID,students.cName,[SUM](http://localhost/phpMyAdmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/group-by-functions.html#function_sum)(scorelist.score),[AVG](http://localhost/phpMyAdmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/group-by-functions.html#function_avg)(scorelist.score)FROM students,scorelist WHEREstudents.cID=scorelist.cID GROUP BY students.cID, students.cName

結果：



使用JOIN結合資料表

SELECT 顯示欄位…FROM 資料表A [INNER] JOIN 資料表B

ON A.相關欄位=B.相關欄位

另一個方式為

SELECT 顯示欄位…FROM 資料表A [INNER] JOIN 資料表B

USING (相關欄位)

例：

SELECT students.cID,students.cName,scorelist.score FROM students JOIN scorelist USING (cID) WHERE scorelist.course='國文'

LEFT JOIN、RIGHT JOIN 語法

若是用上述JOIN的方式，必須在兩邊的資料表都有資料才會顯示，若是有一個學生沒有登錄成績，結果中就不會顯示這個學生的資訊，如此就會遺漏資訊

此時可以使用LEFT JOIN或 RIGHT JOIN的方法來執行資料表連結 語法為

SELECT 顯示欄位…FROM 資料表A LEFT|RIGHT JOIN 資料表B

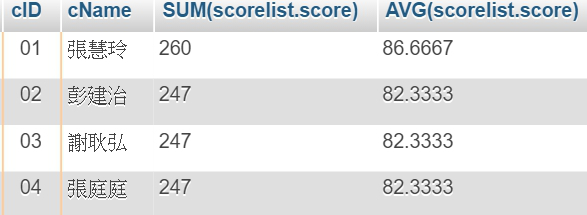
ON A.相關欄位=B.相關欄位

例：有個同學沒有登入成績，所以不會顯示出來

SELECT students.cID,students.cName,SUM(scorelist.score),AVG(scorelist.score) FROM students,scorelist WHERE students.cID = scorelist.cID

GROUP BY students.cID,students.cName

結果：



例：改為LEFT JOIN方式

SELECT students.cID,students.cName,SUM(scorelist.score),AVG(scorelist.score) FROM students LEFT JOIN scorelist ON students.cID = scorelist.cID GROUP BY students.cID,students.cName

結果：



沒有登錄成績的同學也能顯示出來了