

Information and
Communication Technology
資訊與通訊科技

Structured Query Language
SQL (1)
結構化查詢語言

姓名:_____ 班別:_____() CS____

1

結構化查詢語言

SQL是什麼？

- 當用戶發出一項**查詢Query**，便可從數據庫檔(_____)內獲得若干資料。
- 這項查詢是根據用戶所提供的條件(**condition**)所作出的一項檢索。
- 用戶只須列明查詢的_____, 而不須要實際知道有關的檢索_____。

1

引言

SQL的概念

- 用戶先列出**數據庫檔**及查詢的**條件**。
- SQL 程序便會在這數據庫檔內檢查每筆記錄是否符合這項條件，並把有關的資料顯示出來。這個過程稱為檢索。
- 可查問**統計**數項。
- 查詢所得的結果會以表格的形式顯示。

2 基本結構

一般語法	SELECT, DISTINCT, *, AS, FROM, WHERE
比較	IN, BETWEEN, LIKE "% _"
群組	GROUP BY, HAVING
函數	COUNT(), SUM(), AVG(), MAX(), MIN()
顯示次序	ORDER BY, ASC / DESC
邏輯運算符	AND, OR, NOT
輸出	CREATE TABLE AS SELECT ...
聯合	UNION

建立表格	CREATE TABLE ... (...)
插入紀錄	INSERT INTO ... VALUES (...)
刪除紀錄	DELETE ... FROM ... WHERE ...
刪除表格	DROP TABLE ...

SQL

3

2 實例：學生個人資料

STUDENT

<u>欄名</u>	<u>類型</u>	<u>欄寬</u>	<u>內容</u>
id	字符	4	學生編號
name	字符	10	學生名字
dob	日期	8	出生日期
gender	字符	1	性別：M / F
class	字符	2	班別
house	字符	1	社名：R, Y, B, G
district	字符	3	地區碼
remission	binary	1	學費減免
mtest	數字	2	數學測驗分數

SQL

4

```
CREATE TABLE student (
  id      CHAR(4) NOT NULL,
  name    VARCHAR(20),
  dob     DATE,
  gender  CHAR(1),
  class   CHAR(2),
  house   CHAR(1),
  district CHAR(3),
  remission BINARY(1),
  PRIMARY KEY (id)
);
```

```
CREATE TABLE MathTest(
  id _____,
  mtest _____,
  PRIMARY KEY (_____)
);
```

SQL

5 5

2 實例：學生個人資料

STUDENT

id	name	dob	gender	class	house	district	remission	mtest
9801	Peter	1996-06-04	M	1A	R	SSP	0	68
9802	Mary	1996-01-10	F	1A	Y	HHM	0	76
9803	Johnny	1996-03-16	M	1A	G	SSP	1	38
9804	Wendy	1996-07-09	F	1B	B	YMT	0	92
9805	Tobe	1996-10-17	M	1B	R	YMT	0	60
9806	Kitty	1997-01-25	F	1B	R	YMT	1	58
9807	Eric	1997-05-05	M	1C	Y	MKK	0	28
9808	Patty	1997-08-13	F	1C	Y	HHM	1	92

橫列
row

欄
field
column

資料表格table

SQL

6

一般語法

```
SELECT [DISTINCT] expr1 [AS col1], expr2 [AS col2];  
FROM tablename WHERE condition
```

- SQL 程序會從數據庫檔 *table* 選取符合條件的橫列 (row) 並以表格的格式顯示。
- 表達式 *expr1*, *expr2* 可以是
 - (1) 字段 *field*
 - (2) 以函數和字段組成的表達式
- 而 *col1*, *col2* 是表達式 *expr1*, *expr2* 在輸出結果的表格內的欄名。

一般語法

```
SELECT [DISTINCT] expr1 [AS col1], expr2 [AS col2];  
FROM tablename WHERE condition
```

- 選項 ***DIS***_____ 會把重覆出現的橫列刪去 (即只顯示一次)。
- 條件 ***condition*** 可以是
 - (1) 等式或不等式
 - (2) 字符串的比較
 - 使用邏輯運算符 ***AND***, ***OR***, _____

一般語法

例 1 列出所有學生的資料。

```
SELECT ____ FROM student
```

id	name	dob	gender	class	house	district	remission	mtest
9801	Peter	1996-06-04	M	1A	R	SSP	0	68
9802	Mary	1996-01-10	F	1A	Y	HHM	0	76
9803	Johnny	1996-03-16	M	1A	G	SSP	1	38
9804	Wendy	1996-07-09	F	1B	B	YMT	0	92
9805	Tobe	1996-10-17	M	1B	R	YMT	0	60
9806	Kitty	1997-01-25	F	1B	R	YMT	1	58
9807	Eric	1997-05-05	M	1C	Y	MKK	0	28
9808	Patty	1997-08-13	F	1C	Y	HHM	1	92

SQL

9

一般語法

例 2 列出 1A 班學生的名字和社名。

```
SELECT name, house, class  
FROM student  
WHERE class='1A';
```

name	house	class
Peter	R	1A
Mary	Y	1A
Johnny	G	1A
Luke	G	1A
Bobby	B	1A
Aaron	R	1A
:	:	:

SQL

10

一般語法

例 3 列出紅社社員的居住地區。

```
SELECT _____ district
FROM student
WHERE house="R"
```

district
HHM
KWC
MKK
SSP
TST
YMT
:

例 4 列出1B班女生的名字和年齡。

```
SELECT name, dob,
       year(_____) - year(dob) as age
FROM student
WHERE class="1B"
```

SQL

11

一般語法

例 5 列出1A班沒有學費減免的學生的名字和編號。

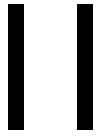
```
SELECT name, id, class
FROM student
WHERE class="1A"
      AND _____ remission;
```

比較

```
expr IN ( value1, value2, value3)
expr BETWEEN value1 AND value2
expr LIKE "%_"
```

SQL

12



比較

星期日 dayofweek(dob)=1
:
星期六 dayofweek(dob)=7

例 6 列出所有出生於星期三或星期六的學生。

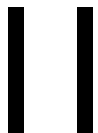
```
SELECT name, class, dayofweek(dob)
FROM student
WHERE dayofweek(dob)=___
      OR dayofweek(dob)=___
```

例 7 列出所有非在四月、六月、九月或十一月出生的學生。

```
SELECT name, class, month(dob)
FROM student
WHERE month(dob) _____(4,6,9,11)
```

SQL

13



比較

例 8 列出1A班的學生名字，其數學測驗分數界乎於80至90分之間。

```
SELECT name, mtest
FROM student
WHERE class="1A"
      AND mtest          AND mtest
```

AND mtest _____ 80 AND 90

SQL

14

比較

例 9 列出所有學生其名字是以"T"為起首。

```
SELECT name, class
FROM student
WHERE name LIKE "_____"
```

例10 列出所有紅社社員其名字的第二個字母是"a"

```
SELECT name, class, house
FROM student
WHERE name LIKE "_____a"
      AND house="R"
```

群組

```
SELECT ..... FROM ..... WHERE condition ;
GROUP BY groupexpr [HAVING requirement]
```

群組函數：

COUNT(), SUM(), AVG(), MAX(), MIN()

- **GROUP BY *groupexpr*** 列出群組組成所依照的表達式。一般都是數據庫檔的一欄。
- **WHERE *condition*** 列出個別橫列所須符合的條件，而 **HAVING *requirement*** 則列出個別群組須符合的條件。



群組 GROUP BY

例11 列出**每一班**的人數。

```
SELECT class, count(____)
FROM student
GROUP BY class
```

class	count
1A	10
1B	9
1C	9
2A	8
2B	8
2C	6

例12 列出**每一班**的數學測驗**平均分**。

```
SELECT class, _____(mtest)
FROM student
GROUP BY class
```

class	avg(mtest)
1A	85.9
1B	70.3
1C	37.9
2A	89.4
2B	53.1
2C	32.7



群組

例13 列出每一居住地區的女生數目。

```
SELECT district, _____
FROM student
WHERE gender="F"
GROUP BY _____
```

district	count
HHM	6
KWC	1
MKK	1
SSP	5
TST	4
YMT	6

例14 列出**每一區**中一學生數學測驗的最高分及最低分。

```
SELECT district, _____(mtest), _____(mtest)
FROM student
WHERE class LIKE "  "
GROUP BY district
```

District	min	max
HHM	36	92
MKK	19	91
SSP	31	91
TST	36	92
TSW	75	75
YMT	38	88

III

群組

例15 列出每一班男生數學測驗的**平均分**，
但若**該班男生人數少於3人**，則不計算在內。

```
SELECT class, ____ (mtest)
FROM student
WHERE gender="M"
GROUP BY class
HAVING _____
```

class	avg(mtest)
1A	86.0
1B	77.7
1C	35.6
2A	86.5
2B	56.5

```
SELECT gender, if( gender = "F", 1, 0 )
FROM school;
```

SQL

```
SELECT class,
sum( if( gender = "M", __, __ ) ) AS boys,
sum( if( gender = "F", __, __ ) ) AS girls
FROM school
GROUP by class;
```

19

IV

顯示次序

```
SELECT ..... FROM ..... WHERE .....
GROUP BY ..... ;
ORDER BY colname ASC / DESC
```

例16 列出1A班男生的名字，並按名字序顯示。

```
SELECT name, id
FROM student
WHERE gender="M"
      AND class="1A"
ORDER BY _____
```

name	id
Aaron	9812
Bobby	9811
John	9803
Luke	9810
Peter	9801
Ron	9813

SQL

20

IV

顯示次序

例17 列出2A班的學生資料，
並按居住地區序顯示。

```
SELECT *  
FROM student  
WHERE class="2A"  
ORDER BY district
```

name	id	class	dcode
Jimmy	9712	2A	HHM
Tim	9713	2A	HHM
Samual	9714	2A	SHT
Rosa	9703	2A	SSP
Helen	9702	2A	TST
Joseph	9715	2A	TSW
Paula	9701	2A	YMT
Susan	9704	2A	YMT

例18 列出每區居住學生的人數，
並按降冪顯示。

```
SELECT district, count(*) as cnt  
FROM student  
GROUP BY district  
ORDER BY cnt _____
```

cnt	docode
11	YMT
10	HHM
10	SSP
9	MKK
5	TST
2	TSW
1	KWC
1	MMK
1	SHT

SQL

21

IV

顯示次序

例19 列出每社的男社員名字，並按班別序顯示。
(即先排社，再排班次序)

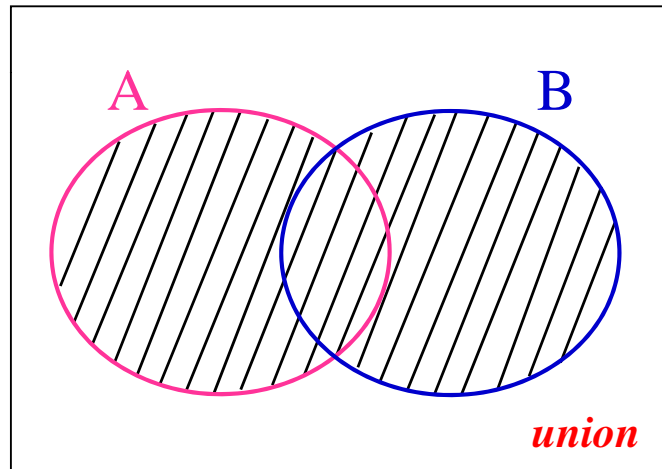
```
SELECT name, class, house  
FROM student  
WHERE gender="M"  
ORDER BY _____
```

SQL

22

3 數據庫聯合、相交及差分

A 和 **B** 的聯合 ($A \cup B$)



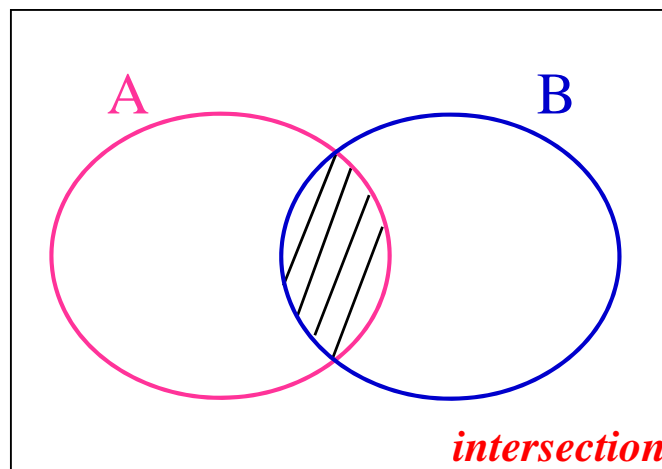
檢取屬於 A 或 B 的所有橫列。

SQL

23

3 數據庫聯合、相交及差分

A 和 **B** 的相交 ($A \cap B$)



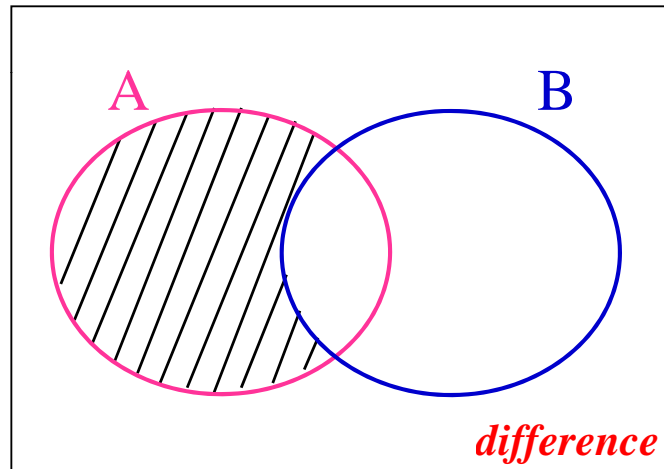
檢取 A 和 B 所共通的橫列。

SQL

24

3 數據庫聯合、相交及差分

A 和 B 的相交 (A-B)



檢取只屬於 A 而不屬於 B 的橫列。
(即從 A 把 B 的部分排出)

SQL

25

3 實例：橋牌會和棋藝會

考慮學校的橋牌會和棋藝會的會員，他們的資料分別貯存於同一結構的數據庫檔內：

BRIDGE / CHESS

欄名	類型	欄寬	內容
id	數字	4	學生編號
name	字符	10	學生名字
gender	字符	1	性別：M/F
class	字符	2	班別

SQL

26

3 數據庫聯合、相交及差分

橋牌會(A)和棋藝會(B)

Bridge [A]					Chess [B]				
	id	name	sex	class		id	name	sex	class
1	9812	Aaron	M	1A	1	9802	Mary	F	1A
2	9801	Peter	M	1A	2	9801	Peter	M	1A
3	9814	Kenny	M	1B	3	9815	Eddy	M	1B
4	9806	Kitty	F	1B	4	9814	Kenny	M	1B
5	9818	Edmond	M	1C	5	9817	George	M	1C
	:	:	:	:		:	:	:	:

SQL

27

3 數據庫聯合、相交及差分

```
SELECT ..... FROM ..... WHERE ..... ;  
UNION ;  
SELECT ..... FROM ..... WHERE .....
```

例22 本校計劃舉行一次棋橋活動。
列出兩會所有會員名單。

```
CREATE TABLE party AS  
SELECT * FROM bridge  
UNION  
SELECT * FROM chess  
ORDER BY class, name
```

SQL

28

3 數據庫聯合、相交及差分

```
SELECT ..... FROM table1 ;  
WHERE col IN ( SELECT col FROM table2 )
```

例23 列出兩會的共同會員。(即兩會的相交)

```
SELECT * FROM bridge  
WHERE id IN  
      ( )
```

```
SELECT *  
FROM chess, bridge  
WHERE _____ id = _____ id
```

SQL

29

3 數據庫聯合、相交及差分

```
SELECT ..... FROM table1 ;  
WHERE col NOT IN  
      ( SELECT col FROM table2 )
```

例24 列出只參加了橋牌會的名單。(即兩會之差)

```
CREATE TABLE diff AS
```

```
SELECT * FROM bridge  
WHERE id NOT IN  
      ( )
```

SQL

30

4 多個數據庫

- 當所查詢的資料貯放於兩個數據庫檔時
- 必須使用**接合 (join)**。

- 接合的作用是
- 把一個數據庫檔內的一個橫列
- 與另一個數據庫檔內的橫列**連合**起來
- 從而把所有不同的組合列出來。

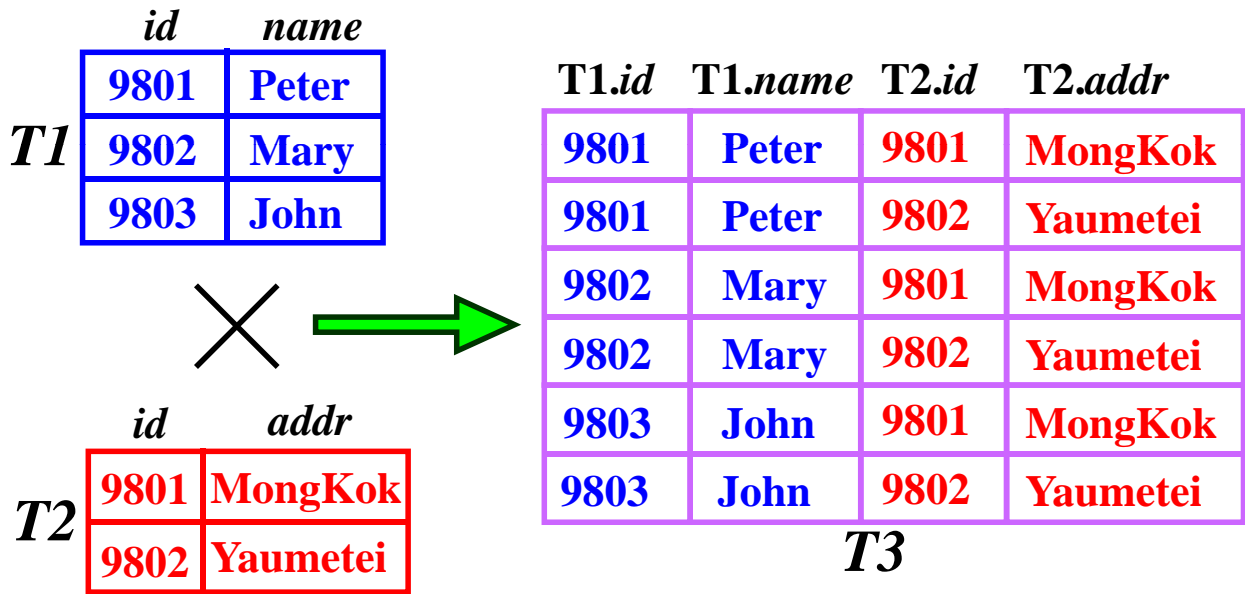
4 自然接合 Natural join

在接合中加上一項接合條件，要求兩檔的**共通欄 (common column)**的值是相同，這稱為**自然接合**。

這目的是要把這兩檔的**相關資料**連合起來，變成一個**合一的**大表格，再從這表格中執行查詢工作。

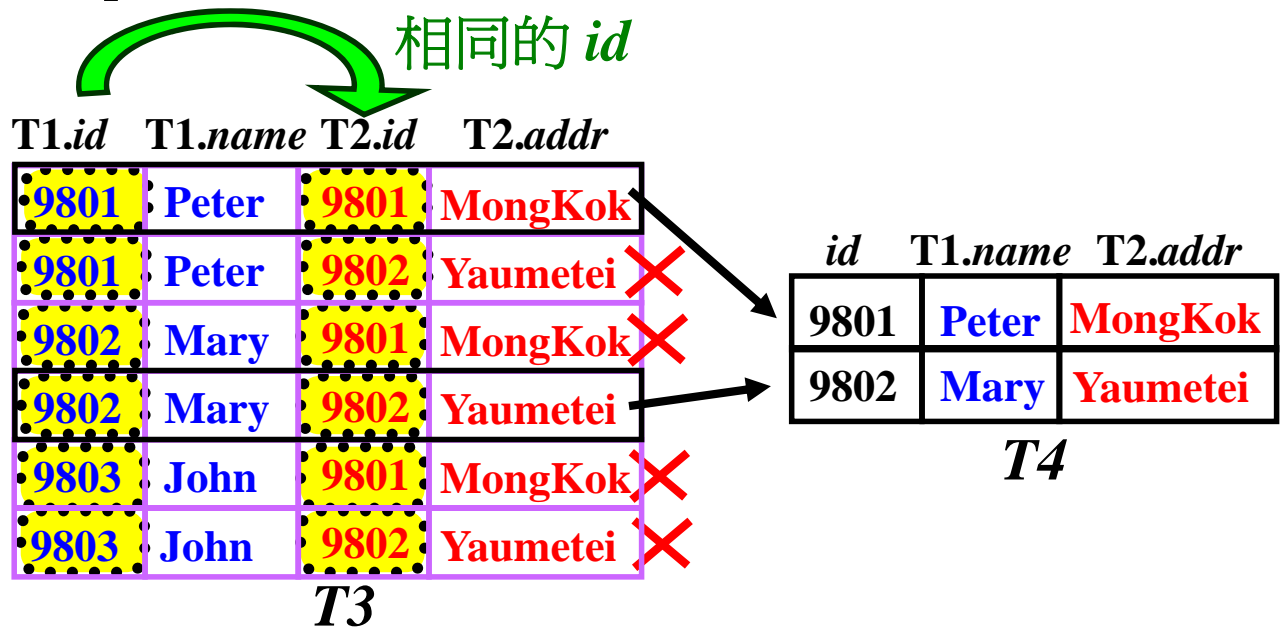
e.g. chess.id=bridge.id

4 多個數據庫



接合 cross join

4 多個數據庫



自然接合 natural join

4

實例：樂器班

學校規定每個學生都須要學習一件樂器。

兩檔：*STUDENT* & *MUSIC*

共同欄：學生編號 *id*

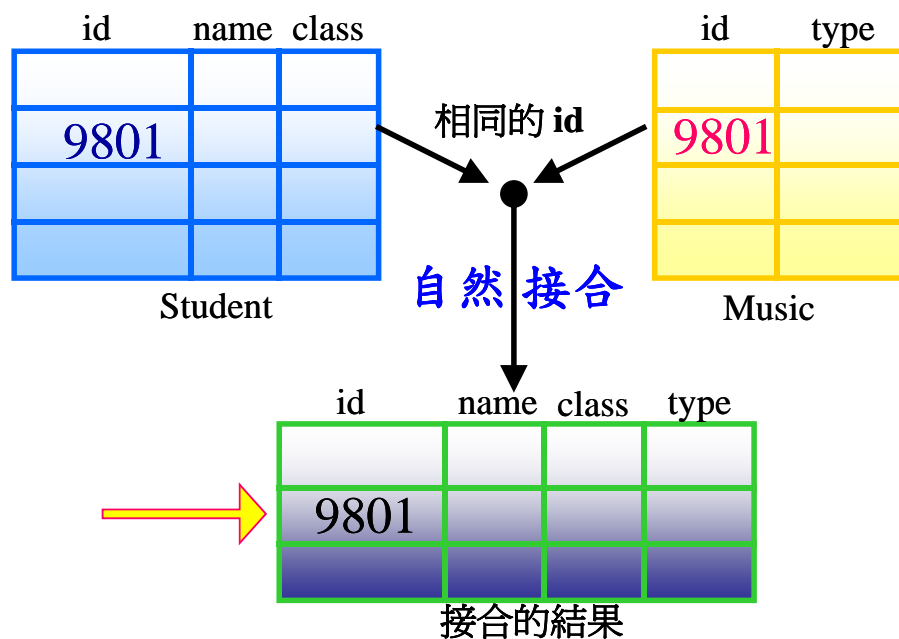
MUSIC

欄名	類型	欄寬	內容
<i>id</i>	數字	4	學生編號
<i>type</i>	字符	10	樂器名稱

4

自然接合 Natural join

例25 列出所有學生的名字及所學習的樂器名稱。



4 自然接合

例25 列出所有學生的名字及所學習的樂器名稱。

```
SELECT s.class, s.name, s.id, m.type ;  
FROM student s, music m ;  
WHERE _____  
ORDER BY class, name
```



class	name	id	type
1A	Aaron	9812	Piano
1A	Bobby	9811	Flute
1A	Gigi	9824	Recorder
1A	Jill	9820	Piano
1A	Johnny	9803	Violin
1A	Luke	9810	Piano
1A	Mary	9802	Flute
:	:	:	:

SQL

37

4 自然接合

例26 列出每班學習鋼琴的學生數目。

三部分：

- (1) 自然接合：以 *id* 來接合
- (2) 條件：*m.type="Piano"*
- (3) 群組：*GROUP BY class*

SQL

38

4 自然接合

例26 列出每班學習鋼琴的學生數目。

```
SELECT s.class, COUNT(*) ;  
FROM student s, music m ;  
WHERE s.id=m.id AND m.type="Piano" ;  
GROUP BY class  
ORDER BY class
```



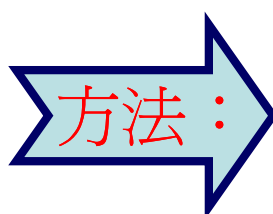
class	cnt
1A	4
1B	2
1C	1

SQL

39

4 外接合 Outer join

外接合是自然接合再加上沒有配合的部分。



- (1) 先列出自然接合
- (2) 再列出沒有配合
- (3) 把兩部分聯合起來

SQL

40

4 外接合

例27 列出尚未選擇樂器的學生名字。(即沒有相配)

```
SELECT class, name, id FROM student ;  
WHERE id NOT IN  
  ( SELECT id FROM music ) ;  
ORDER BY class, name
```



class	name	id
1A	Mandy	9821
1B	Kenny	9814
1B	Tobe	9805
1C	Edmond	9818
1C	George	9817
:	:	:

SQL

41

4 外接合

例28 列出一份名單，去查核所有
學生學習的樂器。

名單須包括尚未參加樂器班
的學生名字。
(即外接合)

SQL

42

4 外接合

例28 **SELECT s.class, s.name, s.id, m.type**
FROM student s, music m
WHERE s.id=m.id

自然接合

UNION

SELECT class, name, id, ''
FROM student
WHERE id NOT IN (SELECT id FROM music)

ORDER BY 1, 2 ;

SQL

43

4 外接合

自然接合

class	name	id	type
1A	Aaron	9812	Piano
1A	Bobby	9811	Flute
1A	Gigi	9824	Recorder
1A	Jill	9820	Piano
1A	Johnny	9803	Violin
1A	Luke	9810	Piano
1A	Mary	9802	Flute
:	:	:	:

class	name	id
1A	Mandy	9821
1B	Kenny	9814
1B	Tobe	9805
1C	Edmond	9818
1C	George	9817
:	:	:

沒有相配

外接合

class	name	id	type
1A	Aaron	9812	Piano
1A	Bobby	9811	Flute
1A	Gigi	9824	Recorder
1A	Jill	9820	Piano
1A	Johnny	9803	Violin
1A	Luke	9810	Piano
1A	Mandy	9821	
1A	Mary	9802	Flute
1A	Peter	9801	Piano
1A	Ron	9813	Guitar
1B	Eddy	9815	Piano
1B	Janet	9822	Guitar
1B	Kenny	9814	
1B	Kitty	9806	Recorder
:	:	:	:

空白

SQL

44