

第四堂 複雜的資料操作

聯成電腦

張明泰 mtchang.tw@gmail.com



複習

- 練習：從records表格中取出以班級座號10109學生所請的每一個假別的時數的累積總和。
- 結果如下：

	n1	n2	s1	s2	s3	4
▶	10109	900228	33	2	4	4
*						



View

- 視界的優點
- 當從一個表格中取出資料的時候，常常會用到某一select敘述，但每次都寫一樣的東西實在很麻煩….
- 我們可以使用select製作出一個view，這樣就可以用較為簡潔的select敘述。



製作view

- 語法：`create view 視界名稱 as (select敘述);`
- 原先範例：`SELECT n1,n2,s1 FROM (SELECT 班級座號 AS n1, 年月日 AS n2, SUM(公假) AS s1, SUM(事假) AS s2, SUM(病假) AS s3, SUM(曠課) AS s4 FROM records GROUP BY 班級座號) AS rec WHERE n1=10109;`
- 範例：`CREATE VIEW view_records AS SELECT 班級座號 AS n1, 年月日 AS n2, SUM(公假) AS s1, SUM(事假) AS s2, SUM(病假) AS s3, SUM(曠課) AS s4 FROM records GROUP BY 班級座號;`
- 用view的範例：`SELECT n1,n2,s1 FROM view_records WHERE n1=10109;`



使用view來進行資料登錄、更新、刪除

- 在視界中的使用，可以視同表格使用。
- 使用限制：
只有在視界是由單一表格製作出來的時候，
才可以使用資料變更。
使用group by, having, distinct製作出來的
view則不能進行資料變更，因為被群組化
後，無法得知原本是哪一列。



刪除view

- 範例：`drop view view_records;`
- 說明：刪除視界表格



操作多個表格

- 把表格與表格連結在一起
- 結合 join
- 交差結合：不一致列出的交差結合
- 內部結合：指令列一致的結合
- 外部結合
 - 左外部結合
 - 右外部結合
 - 全外部結合

何謂結合？

交差結合 cross join

1		1	
2		1	
3		1	
1		2	
2		2	
3		2	
1		3	
2		3	
3		3	

內部結合 inner join

1	1
2	2

表格a 表格b

全外部結合 full join

1	1
2	2
Null	3
3	null

表格a 表格b

左外部結合 left join

1	1
2	2
3	null

表格a 表格b

右外部結合 right join

1	1
2	2
null	4

表格a 表格b

表格a		+		表格b	
1				1	
2				2	
3				4	



Cross join

- 定義：將多個表格中的所有行單純的連結起來的方法
- 範例：**SELECT * FROM students CROSS JOIN view_records;**
- 範例：**SELECT * FROM students, records;**
- 說明：以上兩種寫法出現的結果是一樣的。



Cross join-欄位的指定

- 範例：**SELECT students. 學號, view_records.s1 FROM students , view_records;**
- 說明：要取出某一方的所有欄位可以寫成「students.*」要取出指令欄位可以寫成「student. 學號」其中用句點來作表格名稱區分。



Inner join

- 定義：在交差結合中把指令的欄位中的值一致的資料取出來的方法。
- 範例：**SELECT * FROM students INNER JOIN view_records ON 班級座號=n1;**
- 說明：條件欄位名稱如果名稱一樣，需要以表格名稱作區分。



Inner join ~ where

- 範例：**SELECT * FROM students INNER JOIN view_records WHERE 班級座號=n1;**
- 說明：也可以使用where的寫法達到inner join的效果



Left join

- 定義：在左外部結合中，會以左邊的表格為準來進行結合，即使右邊的表格沒有一致的資料左側的也會被取出。
- 範例：**SELECT * FROM students LEFT JOIN view_records ON 班級座號=n1;**
- 說明：以students為準，結合view_records



Right join

- 定義：在右外部結合中，會以右邊的表格為準來進行結合，即使左邊的表格沒有一致的資料右側的也會被取出。
- 範例：**SELECT * FROM students RIGHT JOIN view_records ON 班級座號=n1;**
- 說明：以view_records為基準，結合students



Full join

- 全外部結合把左外部與右外部的功能結合，不管有沒有值，所有的資料都會被取出。
- 範例：**SELECT * FROM students FULL JOIN view_records ON 班級座號=n1;**
- MySQL, ACCESS不支援

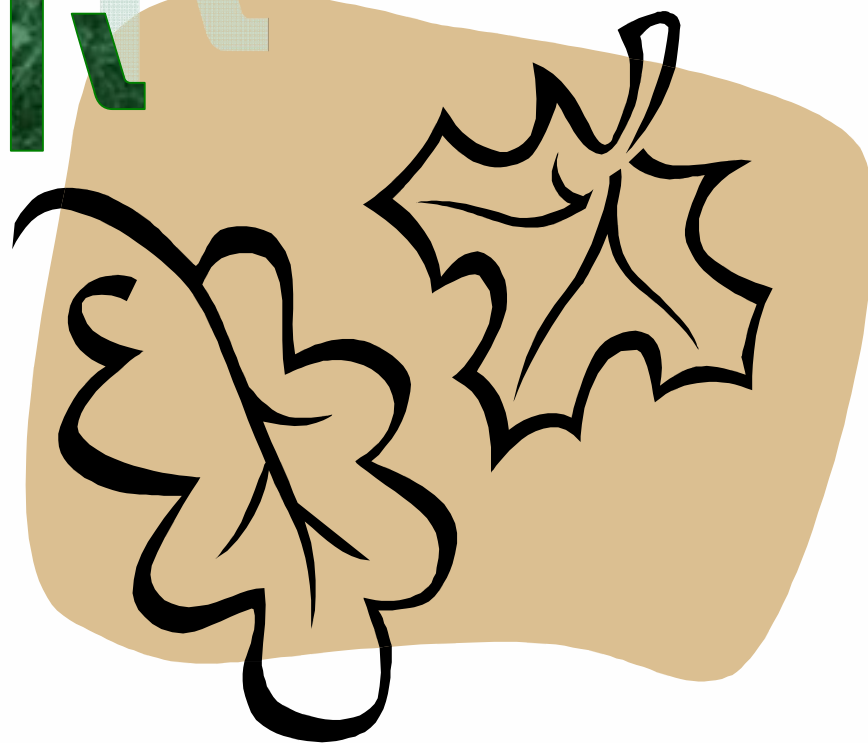
合併練習

- 題目：請合併students及records兩個表格，並計算出每個學生的公假及病假總計請了多少小時？並將此結果製作成view_rec2 視界表格。
查詢結果如下圖所示。（後面課程需用到此表格）

結果1						
	學號	姓名	出生年月日	身分證號碼	公假之總計	病假之總計
▶	911001	王子穎	1984/3/5	C100000012	1	0
	911009	吳泓宜	1984/7/9	F100000097	33	4
	911012	李佩君	1984/9/17	G200000123	7	1
	911015	林秋桂	1983/11/16	G200000150	1	0
	911016	林進財	1984/2/26	F100000168	3	0
	911019	林時雨	1984/9/16	K100000199	1	7
	911027	許益僊	1984/9/29	C100000272	0	0
	911028	郭玟玲	1983/9/19	C200000283	2	0
	911030	陳金瑜	1984/2/17	C200000309	2	0
	911032	曾惠粵	1984/2/11	C100000325	14	5
	911034	楊陵宸	1983/10/23	F100000346	3	15
	911037	廖千慧	1984/10/6	F200000375	0	0



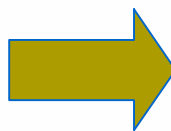
休息一下



何謂集合運算子？

- 所謂集合運算，指的是將兩個select的結果進行加減乘除的運算。所欲結合的select敘述欄位及資料型態必須一樣才可以結合。

學號	姓名	出生年月日	身分證號碼	公假之總計	病假之總計
911034	楊陵羸	1983/10/23	F100000346	3	15
913041	蔡佳真	1982/6/11	A100001286	1	11
915042	李欣怡	1984/2/10	F100004380	0	17



學號	姓名	出生年月日	身分證號碼	公假之總計	病假之總計
911009	吳泓宜	1984/7/9	F100000097	33	4
914012	林君美	1983/2/14	V100002709	25	7

學號	姓名	出生年月日	身分證號碼	公假之總計	病假之總計
911009	吳泓宜	1984/7/9	F100000097	33	4
914012	林君美	1983/2/14	V100002709	25	7
911034	楊陵羸	1983/10/23	F100000346	3	15
913041	蔡佳真	1982/6/11	A100001286	1	11
915042	李欣怡	1984/2/10	F100004380	0	17



UNION

- UNION聯集
- 把兩個select的結果相加，並把重複的整理在一起。如不想把重複整理在一起，可以使用union all
- 範例：**SELECT * FROM view_rec2 WHERE 公假之總計 >10
UNION SELECT * FROM view_rec2 WHERE 病假之總計 >4;**
- 範例：**SELECT * FROM view_rec2 WHERE 公假之總計 >10
UNION ALL SELECT * FROM view_rec2 WHERE 病假之總計 >4;**
- 說明：請先分別操作兩個select敘述，再進行整個union的操作。



INNERSECT

- Innersect差集，只有在兩個集合中一樣的才會被取出。
- 範例：
- 說明：mysql不支援

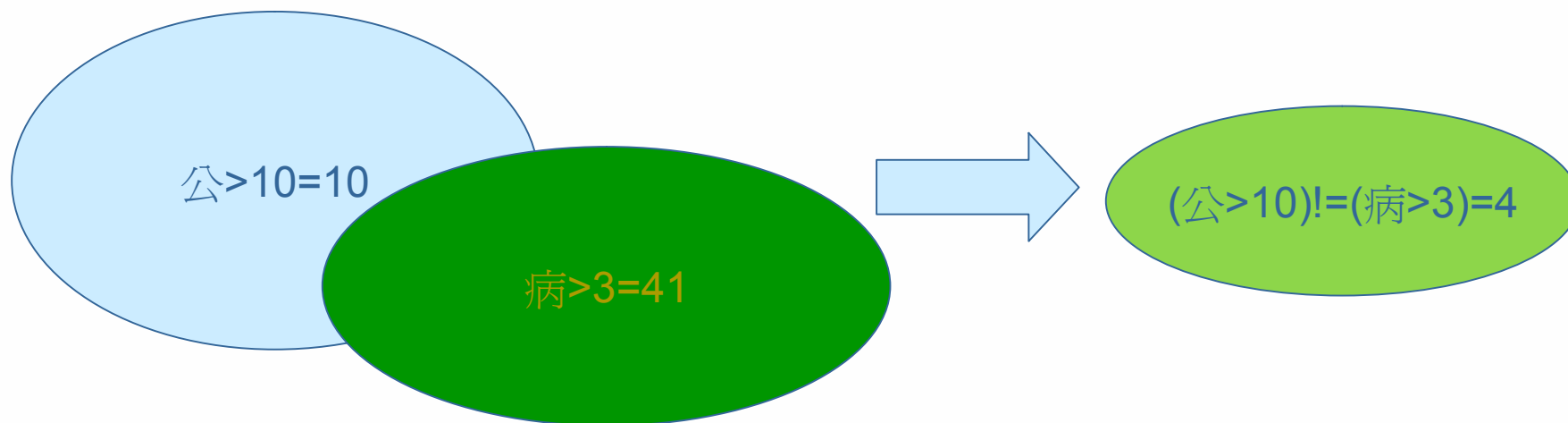


EXCEPT、MINUS

- 差集合，「把~除外」只會取出左邊有右邊沒有的資料。
- 範例：
- 說明：`mysql`不支援

ALL運算子

- 會去和所有的值比較。使用之後會和所有用子查詢取到的資料進行比較。
- 範例：`SELECT 學號 FROM (SELECT * FROM view_rec2 WHERE 公假之總計 >10) AS sel_a WHERE 學號 != ALL(SELECT 學號 FROM (SELECT * FROM view_rec2 WHERE 病假之總計 >3) AS sel_b);`
- 說明：找出請公假大於10的學號，但這些人不在病假大於3的人裡面的學號。





EXISTS運算子

- 在子查詢中取出的資料存在時會傳回ture，若不存在則會傳回false。若是在ture會執行主查詢，若是false的話不會被執行。
- 範例：
- 說明： mysql不支援



ANY運算子

- 這意義是「和任意的值一致」當我們想和子查詢取得的任一進行比對時可以使用此運算子。
- 範例：

```
SELECT 學號 FROM (SELECT * FROM view_rec2 WHERE 公假之總計 >10 ) AS sel_a WHERE 學號 = ANY( SELECT 學號 FROM (SELECT * FROM view_rec2 WHERE 公假之總計 >20 ) AS sel_b);
```
- 說明：

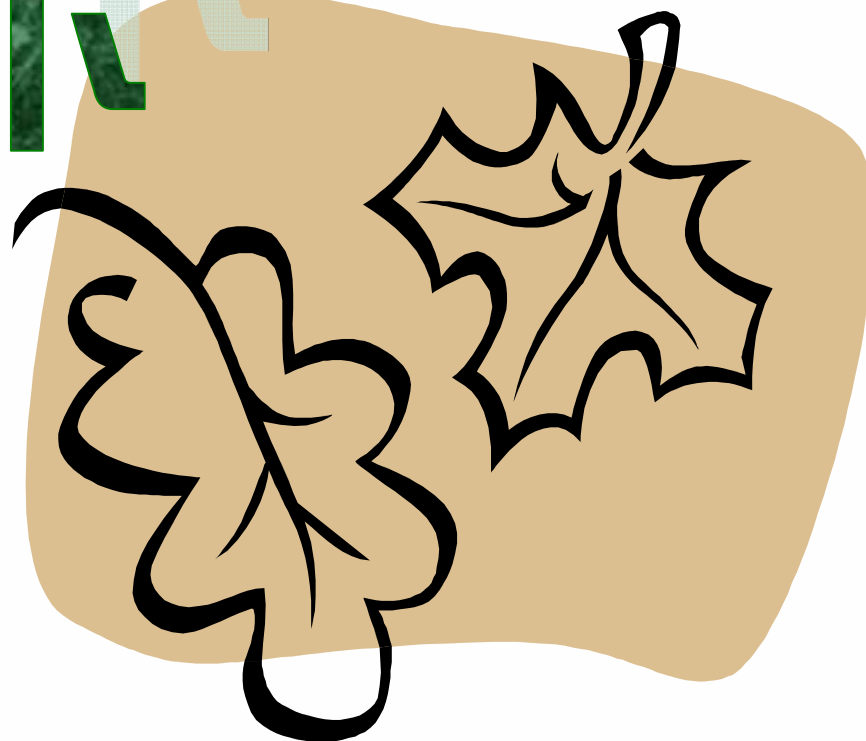


索引編號

- 說明：建立索引提高搜尋效率
- 範例：**CREATE INDEX idx ON students(學號);**
- 說明：刪除索引idx
- 範例：**drop index idx;**
- 使用時請比較前後搜尋所花費的時間。以**SELECT * FROM students WHERE 學號=911044;**為例



休息一下





保護資料的機制

- 交易功能(transaction)
- Lock上鎖-讓交易功能在運作時別人不能存取這份資料



自己決定是否要反映出結果

- Commit送出-若要確定使用交易功能並結束的話可以使用
- Rollback回滾-可以用於程式中，當錯誤發生時就rollback



交易功能transaction

- 我們以轉帳這個動作舉例說明：
- (1)小明確認銀行帳戶的餘額(select敘述)
- (2)從帳戶取出100000元(update敘述)
- (3)將100000匯到小明老婆的帳戶(update敘述)
- 如果在(3)步驟時，剛好系統電源中斷，那會發生什麼事？



交易功能的流程

- (1)將查詢指令交給RDBMS
- (2)接收查詢指令
- (3)進行處理，但是結果先保留起來，等候使用者的指示
- (4)對RDBMS下指示，看是否確定執行處理
- (5)等候使用者的指示，確定執行或是取消



交易功能的宣告

- 範例：（mysql 以start transaction宣告）
begin transaction;
select ...
insert ...
update ...
- ORACLE不需宣告交易



結束交易功能

- 要結束交易功能有兩種方式：
- (1)確定執行處理 → **commit**
- (2)取消執行兩種方法 → **rollback**



交易範例：

- Rollback範例
- `use mtchang;`
- `start TRANSACTION ;`
- `delete FROM test WHERE 住址 LIKE '基隆%' ;`
- (使用`select * from test;` 觀看有無刪除)
- `Rollback`
- (使用`select * from test;` 觀看有無刪除)
- commit範例
- `use mtchang;begin tracsaction;`
- `delete FROM test WHERE 住址 LIKE '基隆%' ;`
- (使用`select * from test;` 觀看有無刪除)
- `commit;`
- (使用`select * from test;` 觀看有無刪除)



將資料上鎖lock

- 我們以轉帳這個動作舉例說明：
- (1)小明確認銀行帳戶的餘額(select敘述)
- (2)從帳戶取出100000元(update敘述)
- (3)將100000匯到小明老婆的帳戶(update敘述)
- 但在(2)的時候，同時間另一個提款操作領取150000元，這時會發生什麼事？



鎖定的種類

- 互斥鎖定 (WRITE locks) – 無法參照或操作
If there are no locks on the table, put a write lock on it.
Otherwise, put the lock request in the write lock queue.
- 共享鎖定 (READ locks) – 可以參照但不能變更
If there are no write locks on the table, put a read lock on it.
Otherwise, put the lock request in the read lock queue.



死結deadlock

- 範例：
- 有兩個交易功能A和交易功能B，假設同時被執行。
- 交易功能A把表格a1鎖住下，要去存取表格b1。
- 交易功能B把表格b1鎖住下，要去存取表格a1。
- 於是雙方便進入永久待機的狀態，被稱為死結。
- 大多數的RDBMS，會有一段時間當作time



參考

- Mysql reference manual
13.4. MySQL Transactional and Locking Statements