

SQL 語法—Select 節狀子查詢

使用 Microsoft Access

銘傳大學 電腦與通訊工程學系
資料庫程式設計 王穎聰

集合成員(練習用資料表 1/4)

- depositor 資料表

customer_name	account_number
Hayes	A_102
Johnson	A_101
Johnson	A_201
Jones	A_217
Lindsay	A_222
Smith	A_215
Turner	A_305

集合咸員(練習用資料表2/4)

- account 資料表

account_number	branch_name	balance
A_101	Brighton	500
A_102	Perryridge	400
A_201	Brighton	900
A_215	Mianus	700
A_217	Brighton	750
A_222	Redwood	700
A_305	Round Hill	350

集合成員(練習用資料表 3/4)

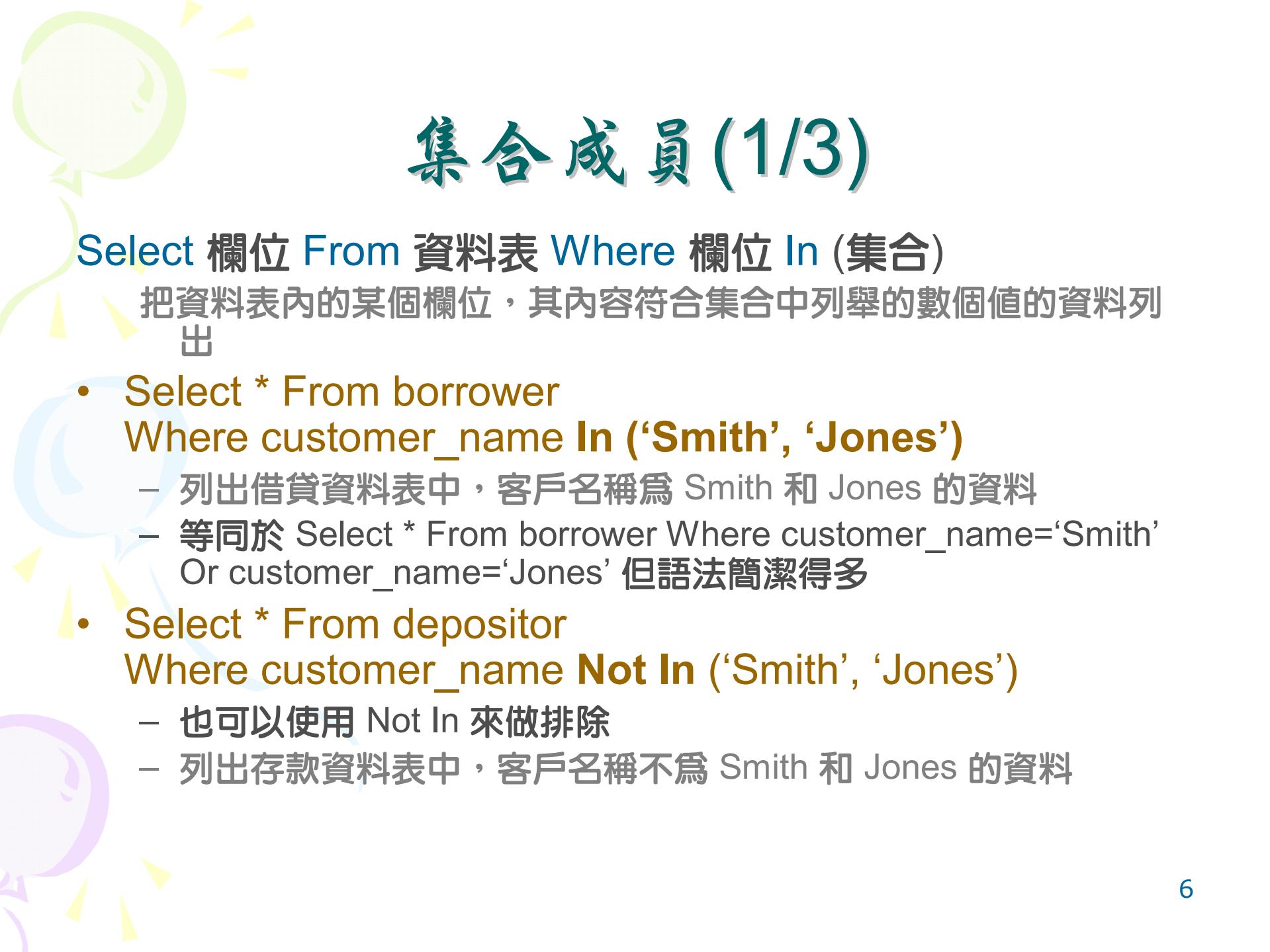
- borrower 資料表

customer_name	loan_number
Adams	L_16
Curry	L_93
Hayes	L_15
Jackson	L_14
Jones	L_17
Smith	L_11
Smith	L_23
Williams	L_17

集合咸員(練習用資料表 4/4)

- loan 資料表

loan_number	branch_name	amount
L_11	Round Hill	900
L_14	Downtown	1500
L_15	Perryridge	1500
L_16	Perryridge	1300
L_17	Downtown	1000
L_23	Redwood	2000
L_93	Mianus	500



集合成員 (1/3)

Select 欄位 From 資料表 Where 欄位 In (集合)

把資料表內的某個欄位，其內容符合集合中列舉的數個值的資料列出

- **Select * From borrower
Where customer_name In ('Smith', 'Jones')**
 - 列出借貸資料表中，客戶名稱為 Smith 和 Jones 的資料
 - 等同於 Select * From borrower Where customer_name='Smith'
Or customer_name='Jones' 但語法簡潔得多
- **Select * From depositor
Where customer_name Not In ('Smith', 'Jones')**
 - 也可以使用 Not In 來做排除
 - 列出存款資料表中，客戶名稱不為 Smith 和 Jones 的資料



集合成員 (2/3)

Select 欄位 From 資料表 Where 欄位 In (Select 語法)

- 如果我們想要查詢哪些客戶既有借貸、也有存款，就可以使用巢狀子查詢
 - Select distinct customer_name From borrower
Where customer_name In
(Select distinct customer_name From depositor)
- 上述例子，即使不用巢狀子查詢，亦可做到近似相同結果：
 - Select distinct borrower.customer_name From borrower, depositor
Where borrower.customer_name=depositor.customer_name
- 雖然兩種寫法都可達成相同結果，但是運算過程所耗費的記憶體與時間卻差很多：
 - 第一例：(borrower 筆數 + depositor 筆數) 個單位空間與時間
 - 第二例：(borrower 筆數 * depositor 筆數) 個單位空間與時間

集合成員 (3/3)

Select 欄位 From 資料表 Where (欄位1, 欄位2) In
(Select 欄位A, 欄位B From 資料表 Where 判斷式)

相當於

Select 欄位 From 資料表 Where
欄位1 In (Select 欄位A From 資料表 Where 判斷式) And
欄位2 In (Select 欄位B From 資料表 Where 判斷式)

- 第一種寫法語句精簡得多，但這種回傳多個欄位的方法，Access 不支援
- 請找出有在 Perryridge 分部，同時具有存款和借貸的所有客戶名稱

集合成員 (3/3)

- 上例可以有很多不同的寫法：
 - `Select distinct borrower.customer_name From borrower, loan
Where borrower.loan_number=loan.loan_number And
branch_name='Perryridge' And
(branch_name, customer_name) In
(Select branch_name, customer_name From depositor, account
Where depositor.account_number=account.account_number)`
- 或
 - `Select distinct borrower.customer_name From borrower, loan
Where borrower.loan_number=loan.loan_number And
(branch_name, customer_name) In
(Select branch_name, customer_name From depositor, account
Where depositor.account_number=account.account_number And
branch_name='Perryridge')`
- 以上兩種回傳多個欄位的寫法 Access 不支援，請試著改寫成 Access 可接受的語法

集合比較(練習用資料表)

- branch 資料表

branch_name	branch_city	assets
Brighton	Brooklyn	7100000
Downtown	Brooklyn	9000000
Mianus	Horseneck	400000
North Town	Rye	3700000
Perryridge	Horseneck	1700000
Pownal	Bennington	300000
Redwood	Palo Alto	2100000
Round Hill	Horseneck	8000000

集合比較

Select 欄位 From 資料表 Where 欄位 關係運算 Some (Select 語法)

Select 欄位 From 資料表 Where 欄位 關係運算 All (Select 語法)

- 還記得在 “Select基本語法” 投影片中提到的一個例子嗎：

Select distinct T.branch_name From branch As T, branch As S
Where T.assets > S.assets

這個例子是”列出資產非最少的所有分部名稱“

- 上述的語法不是很容易看得出要做的事，因此若改成下述寫法就明瞭得多：

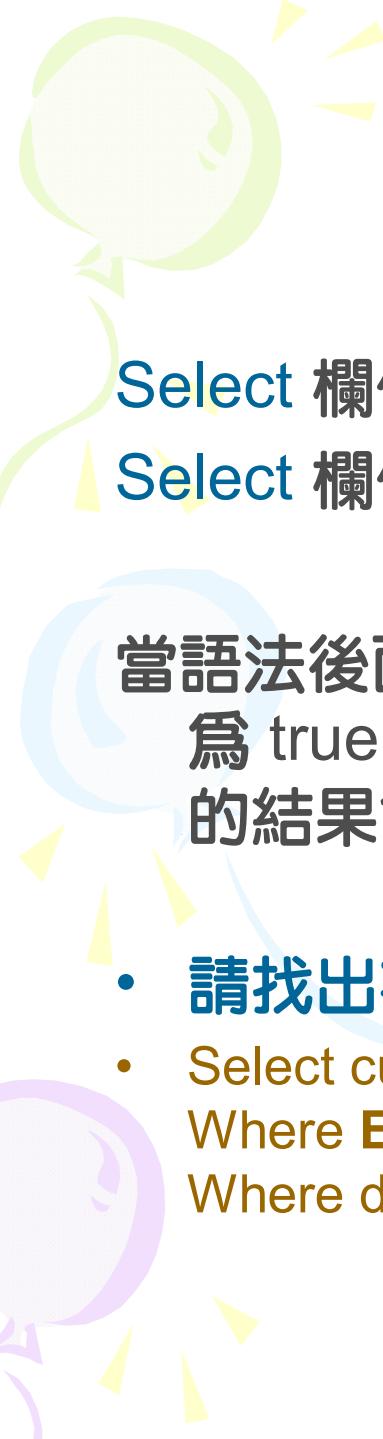
Select distinct branch_name From branch
Where assets > **some (Select assets From branch)**

- 另外若利用聚合函數，也可以寫成：

Select distinct branch_name From branch
Where assets > **(Select Min(assets) From branch)**

集合比較

- 上頁中最底下加入聚合函數的寫法，需要特別注意的是：
聚合函數不可組合使用，例如 $\max(\text{avg}(...))$
- 若我們要找出存款帳戶 (account) 資料表中，哪個分部的
平均結存是最多的
- ```
Select branch_name From account
Group By branch_name
Having avg(balance) >= all
(Select avg(balance) From account
Group By branch_name)
```



# 測試室的關聯(1/2)

Select 欄位 From 資料表 Where Exists ( Select 語法 )

Select 欄位 From 資料表 Where Not Exists ( Select 語法 )

當語法後面的子查詢 Select 回傳有資料時，則 Exists 回傳為 true，才會繼續執行前段的 Select 語法，並且子查詢的結果會帶至主查詢

- 請找出在銀行有存款帳戶且有借貸帳戶的客戶
- Select customer\_name From borrower  
Where Exists ( Select \* From depositor  
Where depositor.customer\_name=borrower.customer\_name)



## 測試室的關聯(2/2)

- 在上例中，子查詢內 Select 的資料表中沒有 borrower，為什麼 Where 子句中有用到 borrower 資料表？
- 其實上例中，不用 Exists 子句，採用過去我們熟悉的 Select 語法就做得到：
- ```
Select borrower.customer_name  
From borrower, depositor Where  
depositor.customer_name=borrower.customer_name
```
- 請找出所有在 Brooklyn 城市的銀行分部有存款帳戶的客戶

測試有否重複的資料(1/2)

Select 欄位 From 資料表 Where Unique (Select 語法)

Select 欄位 From 資料表 Where Not Unique (Select 語法)

當語法後面的子查詢 Select 回傳的資料沒有重複時，則 Unique 回傳為 true，才會繼續執行前段的 Select 語法，並且子查詢的結果會帶至主查詢

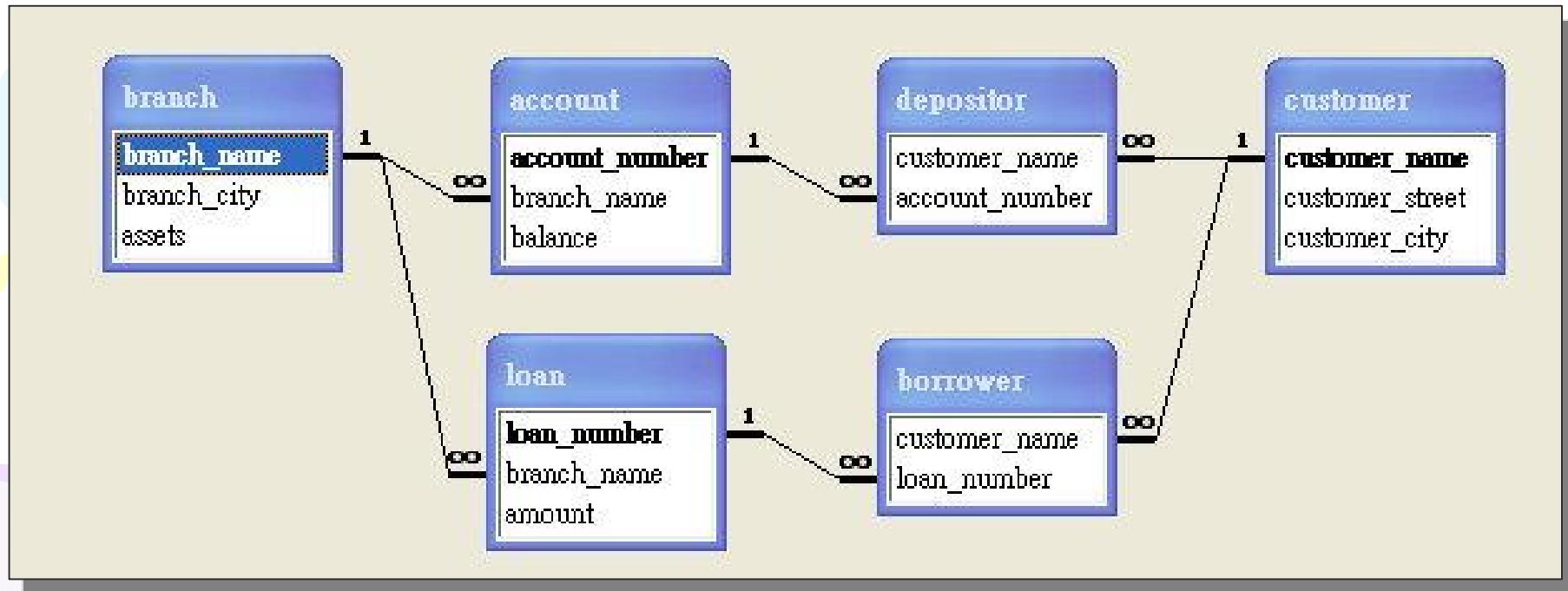
- 請找出在 Perryridge 分部最多只有一個存款帳戶的所有客戶
- Select T.customer_name From depositor as T Where Unique
(Select R.customer_name From account, depositor as R
Where T.customer_name=R.customer_name And
R.account_number=account.account_number And
account.branch_name='Perryridge')

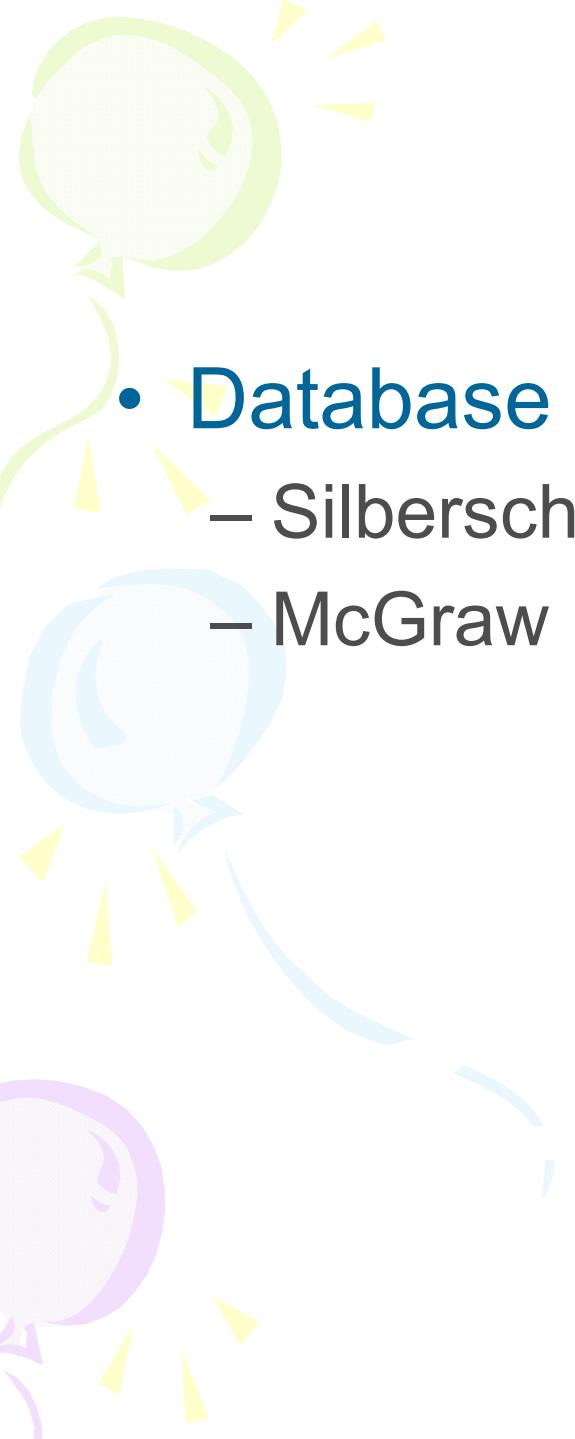
測試有否重複的資料(2/2)

- 反過來說，若請找出在 Perryridge 分部至少有兩個存款帳戶的所有客戶
- Select **distinct** T.customer_name
From depositor as T Where **Not Unique**
(Select R.customer_name From account, depositor as R
Where T.customer_name=R.customer_name And
R.account_number=account.account_number And
account.branch_name='Perryridge')

資料庫綱要圖(練習用) Schema Diagram

- 此資料庫為銀行業務相關，其六個資料表的綱要圖如下所示，本簡報用到其中兩個來練習





參考資料

- Database System Concepts
 - Silberschatz • Korth • Sudarshan
 - McGraw Hill