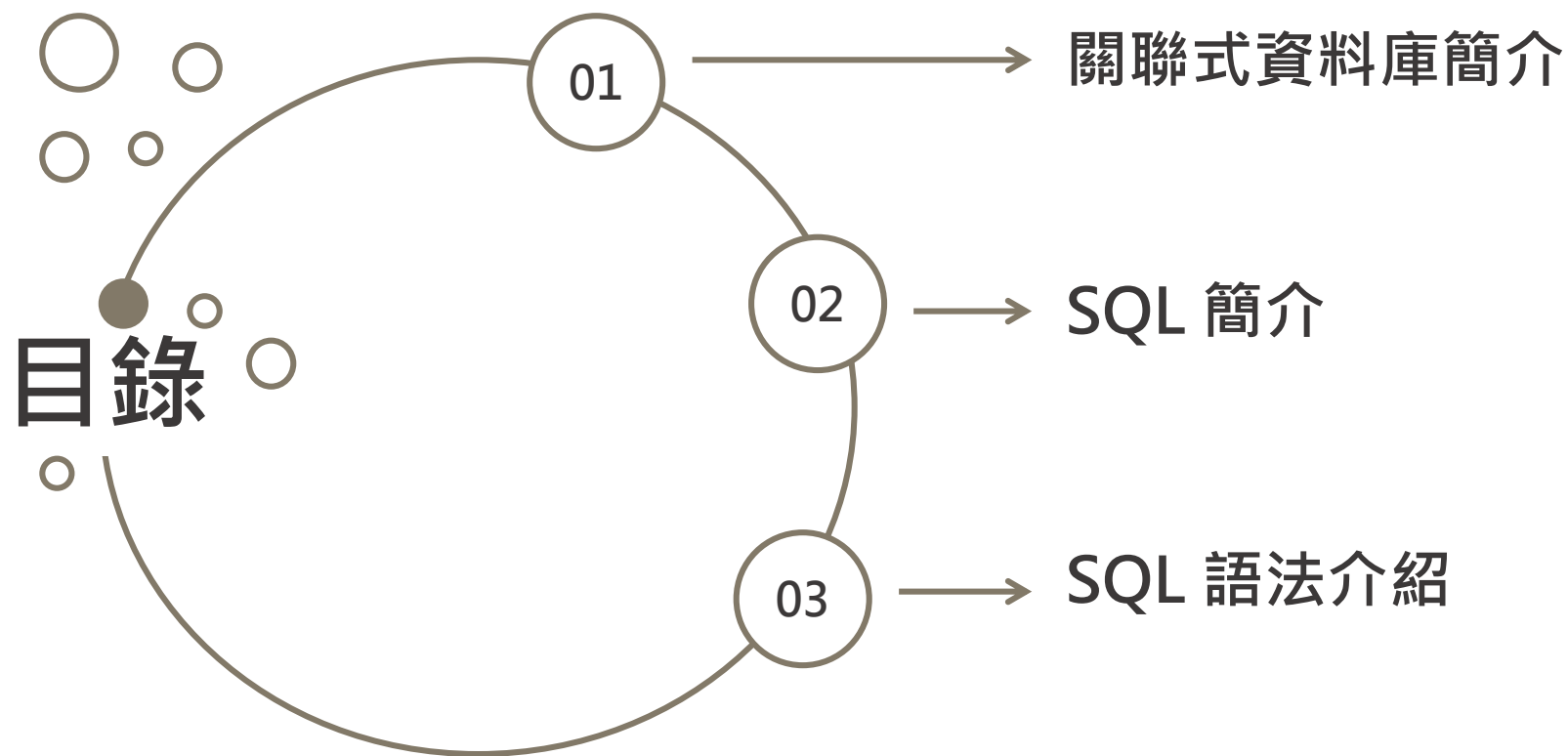


SQL 簡介

系統開發部 科技研發科



關聯式資料庫簡介

數據型態

- 結構化數據

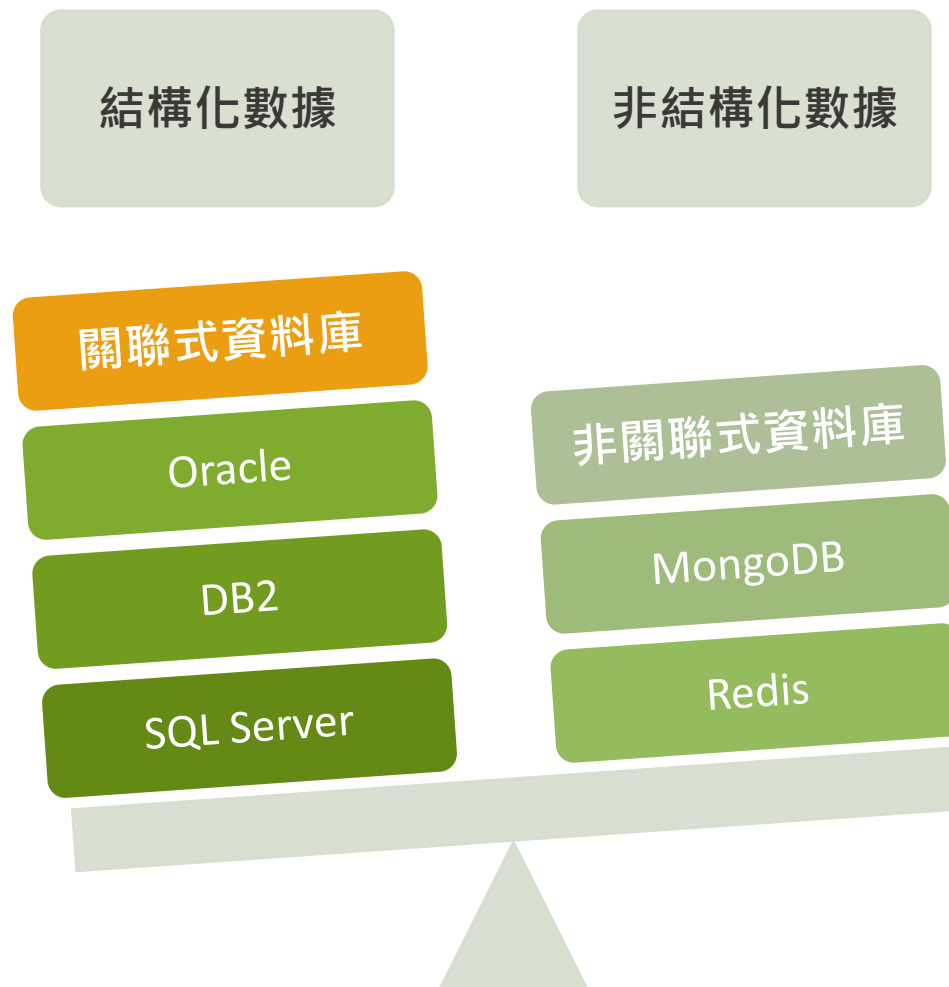
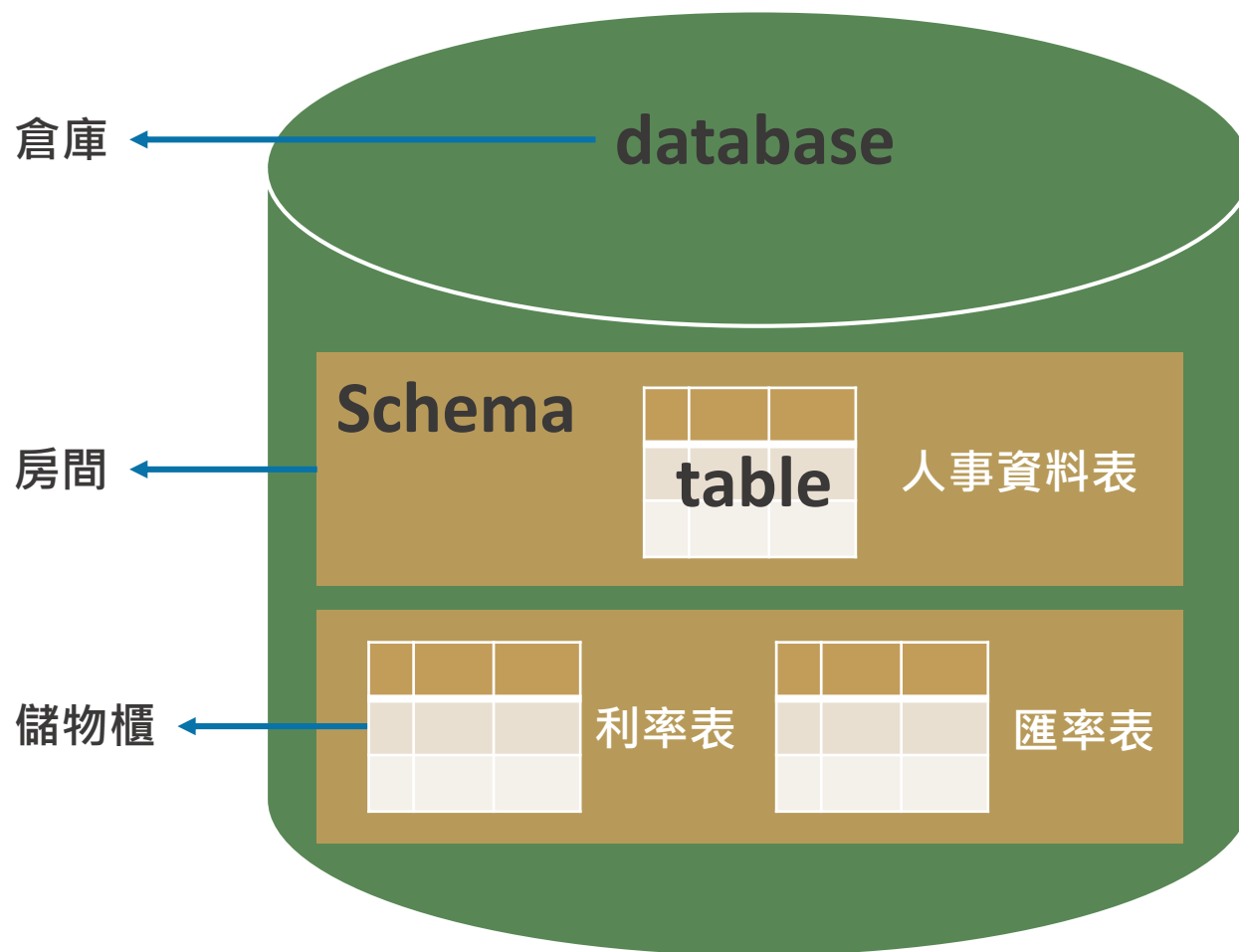
NAME	CLASS	MARK
John Deo	Four	75
Max Ruin	Three	85
Arnold	Three	55
Krish Star	Four	60
John Mike	Four	60
Alex John	Four	55
My John Rob	Fifth	78
Asruid	Five	85
Tes Qry	Six	78
Big John	Four	55

- 非結構化數據

```
1 {
2   "address": {
3     "building": "1007",
4     "coord": [ -73.856077, 40.848447 ],
5     "street": "Morris Park Ave",
6     "zipcode": "10462"
7   },
8   "borough": "Bronx",
9   "cuisine": "Bakery",
10  "grades": [
11    { "date": { "$date": 1393804800000 }, "grade": "A", "score": 2 },
12    { "date": { "$date": 1378857600000 }, "grade": "A", "score": 6 },
13    { "date": { "$date": 1358985600000 }, "grade": "A", "score": 10 },
14    { "date": { "$date": 1322006400000 }, "grade": "A", "score": 9 },
15    { "date": { "$date": 1299715200000 }, "grade": "B", "score": 14 }
16  ],
17  "name": "Morris Park Bake Shop",
18  "restaurant_id": "30075445"
19 }
```

資料庫

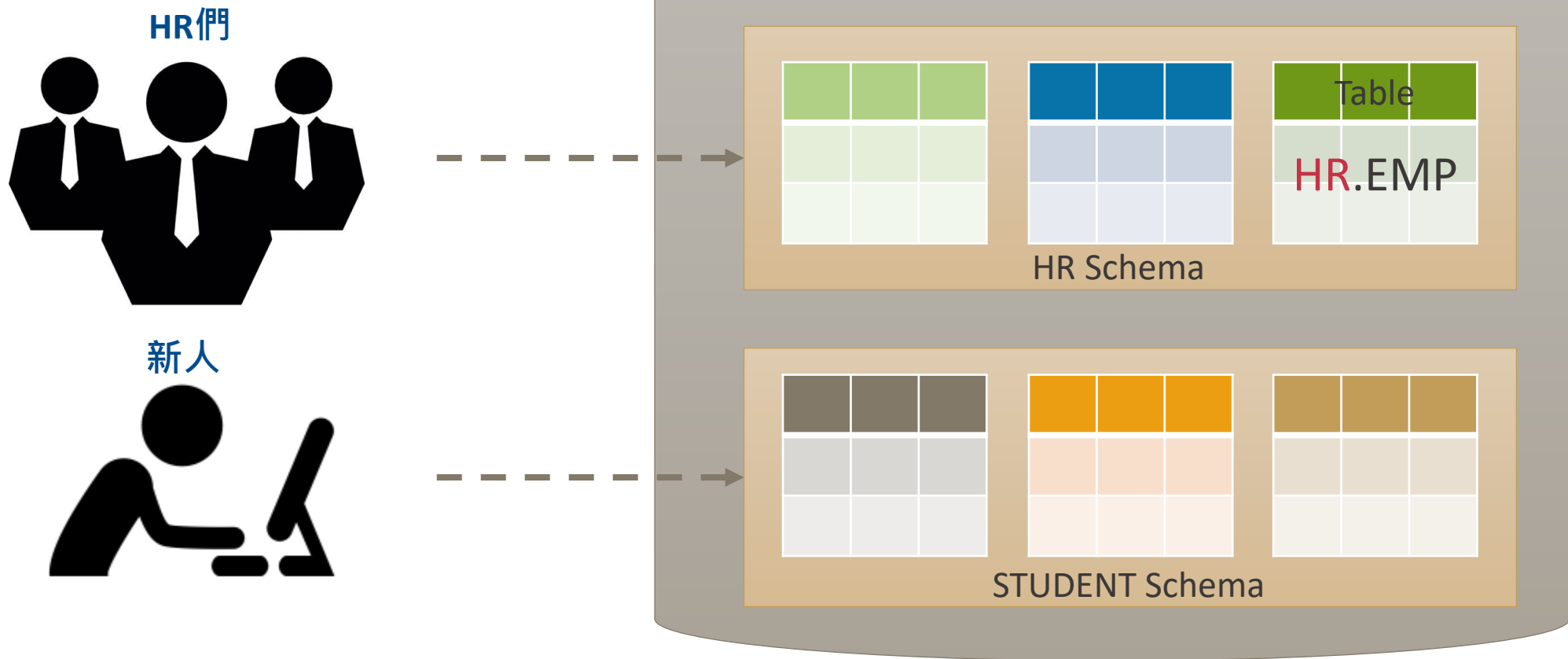
資料庫：數位化資料的集合體，包含許多資料表。



Schema

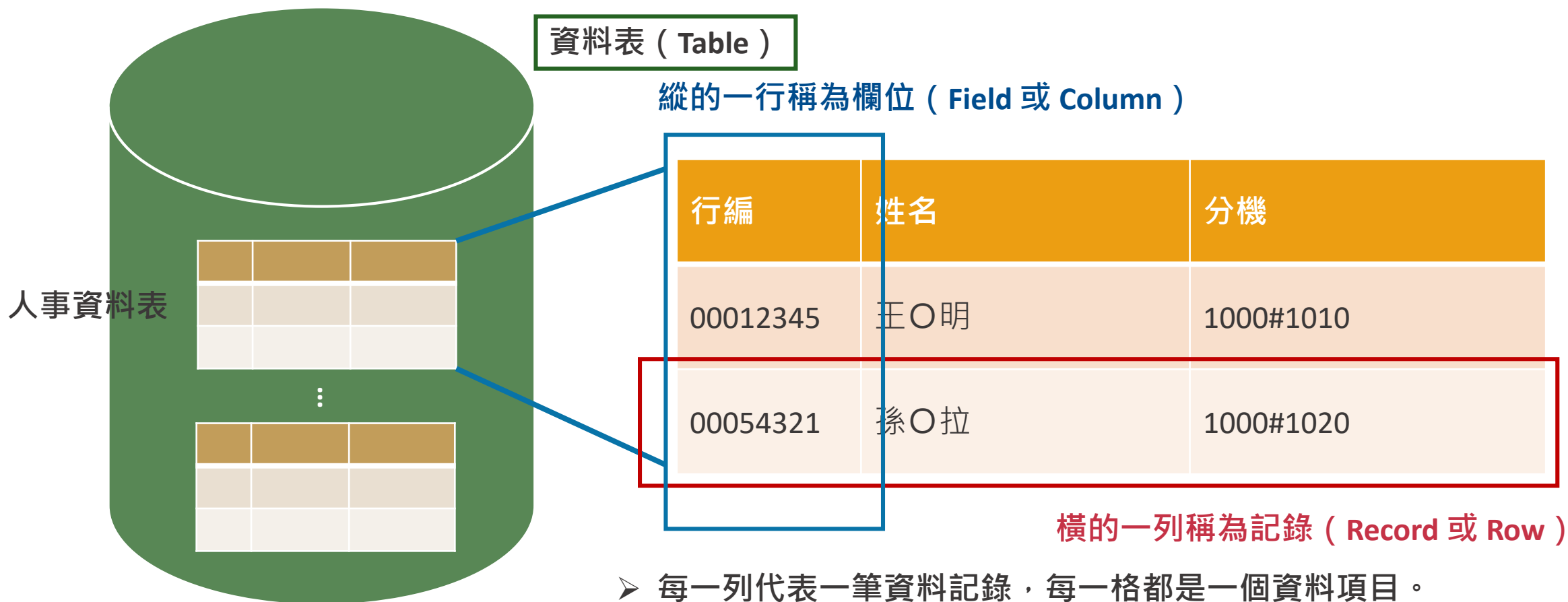
以群組為單位做權限控制。

※ oracle 以連線「帳號」作為 Schema



關聯式資料庫

在關聯式資料庫中的所有數據都儲存在彼此具有明確定義關係的資料表中。
每一筆資料可以視為一個實體，每一欄可以是為每一個實體的屬性。



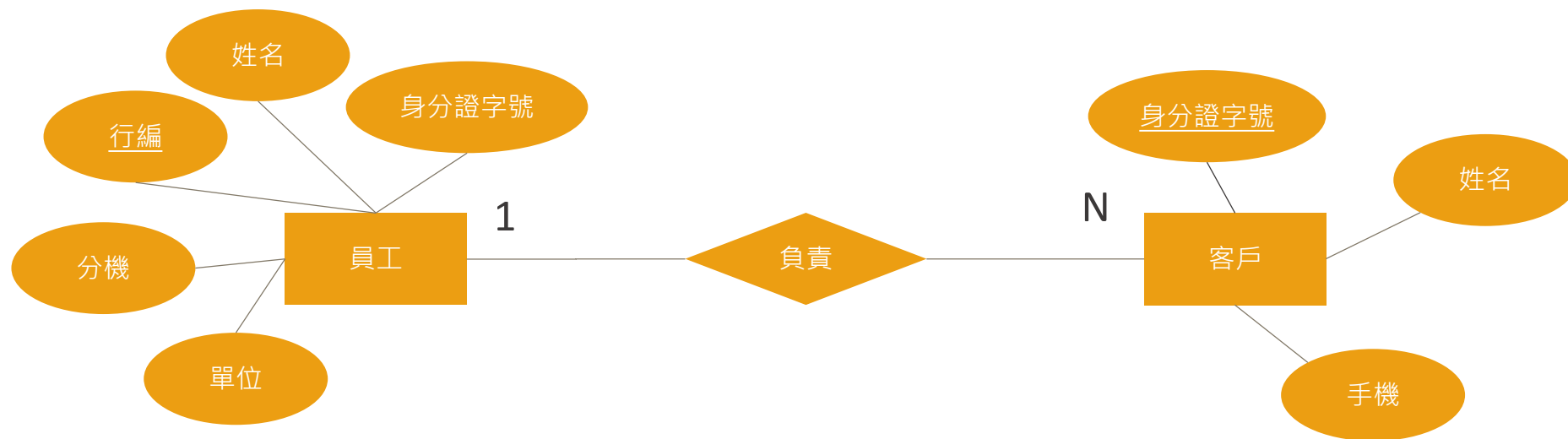
關聯式資料庫

現實世界中的各種「實體」以及「實體」之間的各種聯絡均用關聯模型來表示。

例如：

「每位」員工有行編、姓名、身分證字號、分機、單位等資料，代表一個實體。

「每位」員工又可以負責「多位」客戶，「每位」客戶會有身分證字號、手機資料



關聯式資料庫



良好的數據結構：可靠性和穩定性

資料分割為最小的關係表：
無冗餘和一致的數據儲存

標準化的 SQL 查詢語言



有限的可擴展性

非結構化數據很難映射到
關聯式資料庫的模式中。

SQL 簡介

Structured Query Language

結構化查詢語言：

- ✓ 讓使用者處理資料庫中的資料
- ✓ 是一種 ANSI 標準的計算機語言

ANSI

1. 是一個準國家式的標準機構
2. 批准標準成為美國國家標準，但本身不製定標準。

- ✓ 除了標準之外，各資料庫大都擁有屬於自己的擴充功能或函式。



用途：

- ✓ 建立資料庫
- ✓ 在資料庫建立資料表
- ✓ 設定權限
- ✓ 從資料庫取得資料
- ✓ 新增、修改、刪除資料庫的資料

注意事項：

- 大小寫視為相同
- 分號為該句結束符號

SQL 語言類別



資料定義

- create
- alter
- drop
- NotNull
- PrimaryKey
- ForeignKey
- Index



資料查詢

- select
- as
- where
- join
- distinct
- 子查詢
- 函數



資料操縱

- insert
- update
- delete

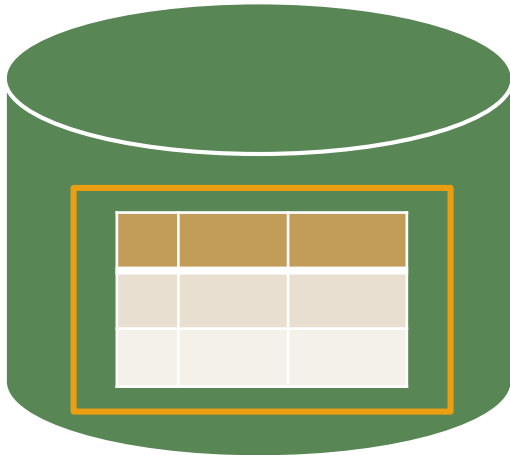
SQL 語法介紹

Data Definition Language , DDL

資料定義

資料表 - 建立

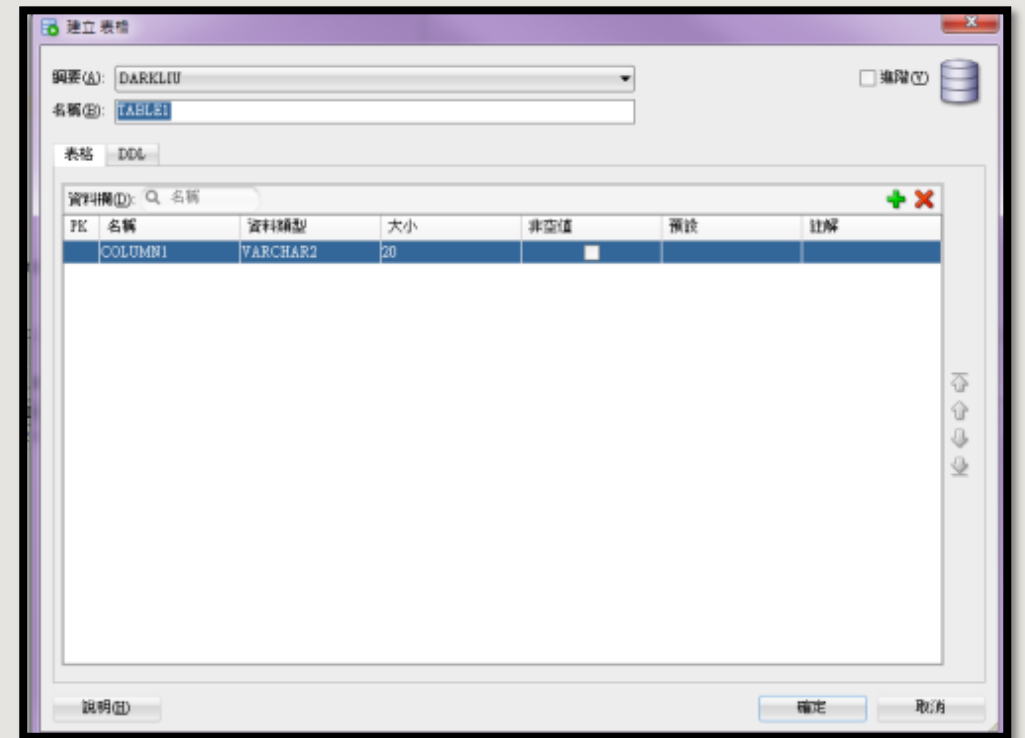
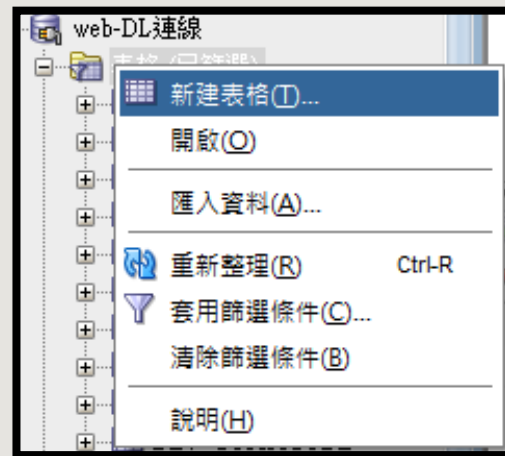
create



【語法】

```
create table TABLE_NM (  
  COLIMN_NM1 data_type,  
  COLIMN_NM2 data_type,  
  COLIMN_NM3 data_type, ... );
```

【畫面】



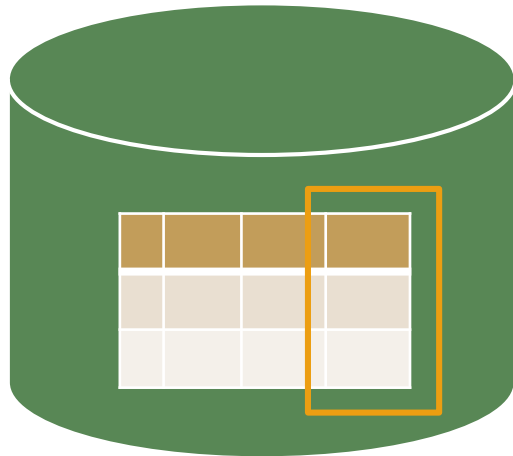
常用 oracle data type

型別	含義	儲存描述	備註
CHAR	固定長度字串	最大長度 2000 bytes	不足自動補空格
VARCHAR2	可變長度的字串	最大長度 4000 bytes	所有字元都佔兩位元組、 把空串等同於 null 處理
NVARCHAR2	根據字符集而定的可變長度字串	最大長度 4000 bytes	For 中文 / 全形字
DATE	日期 (日-月-年)	DD-MM-YY(HH-MI-SS)	
TIMESTAMP	日期 (日-月-年)	DD-MM-YY(HH-MI-SS:FF3)	與 DATE 相比較， TIMESTAMP 有小數位秒資訊
NUMBER(P,S)	數字型別	P 為有效數位，S 為小數位	精度可以高達 38 位
DECIMAL(P,S)	數字型別	P 為有效數位，S 為小數位	
INTEGER	整數型別	小的整數	

◆ Ex. NUMBER(5,2) 的合理值：99,999、543.21

資料表 - 調整

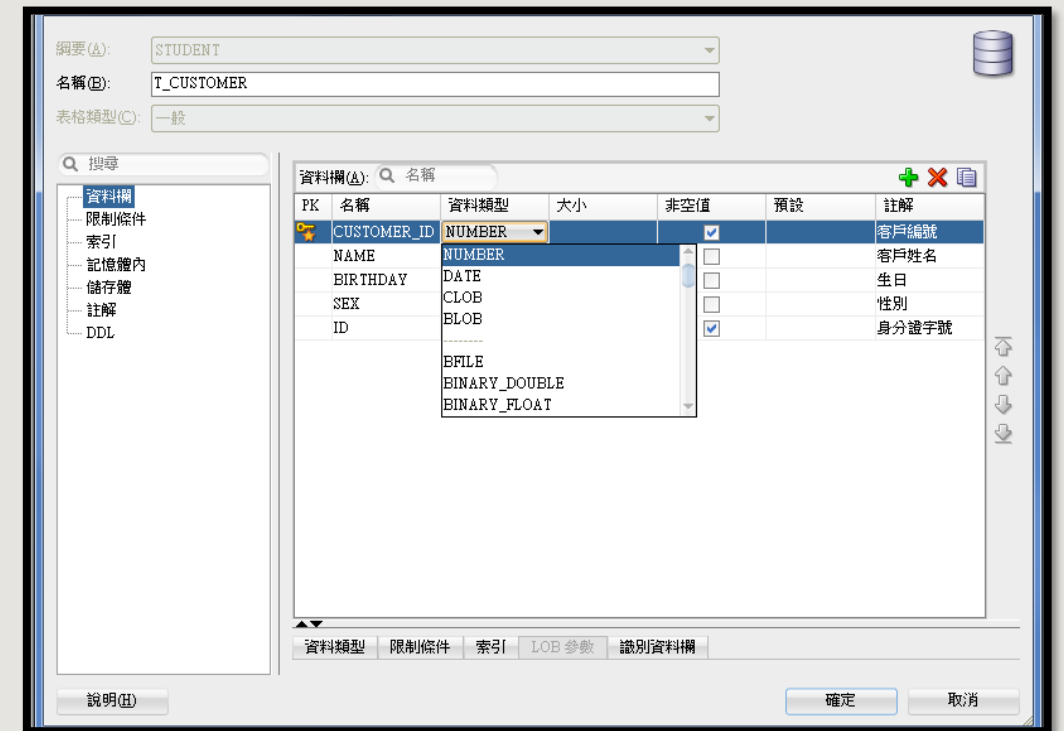
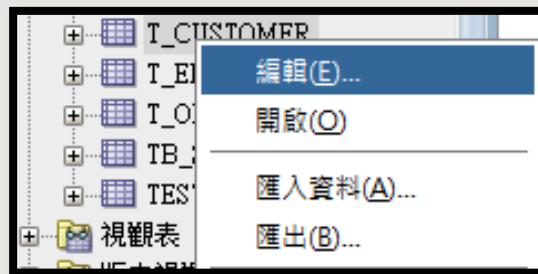
alter



【語法】

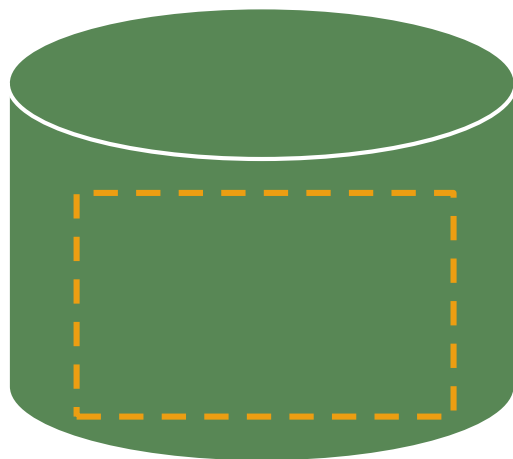
- ◆ 增加欄位：**alter table** TABLE_NM **add** COLIMN_NM datatype;
- ◆ 更改欄位型別：**alter table** TABLE_NM **modify** COLIMN_NM datatype;
- ◆ 更改欄位名稱：**alter table** TABLE_NM **rename column** OLD_NM **to** NEW_NM;
- ◆ 刪除欄位：**alter table** TABLE_NM **drop column** COLIMN_NM;

【畫面】



資料表 - 刪除

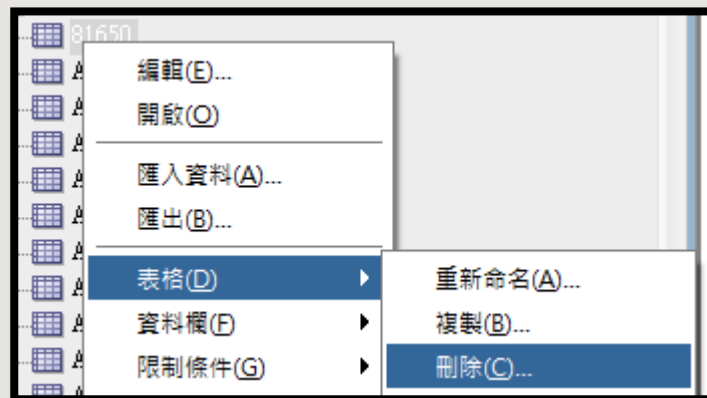
drop



【語法】

drop table TABLE_NM;


【畫面】



課堂練習①

- 將人事資料表在資料庫內新增資料表，table 名稱 EMP_學號。

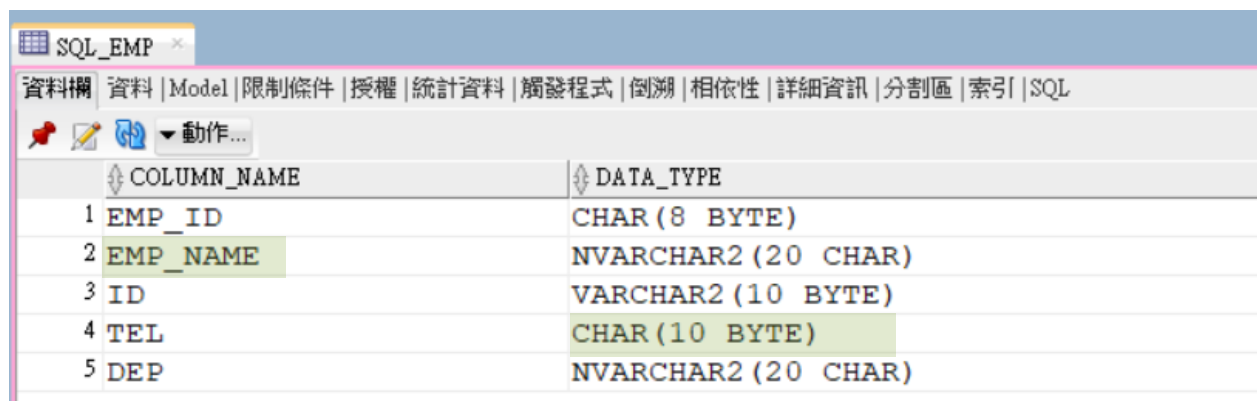
行編	姓名	身分證字號	分機	單位
00012345	王○明	A100100100	1000#1010	系統開發部
00054321	孫○拉	B200200200	1000#1020	開發系統部



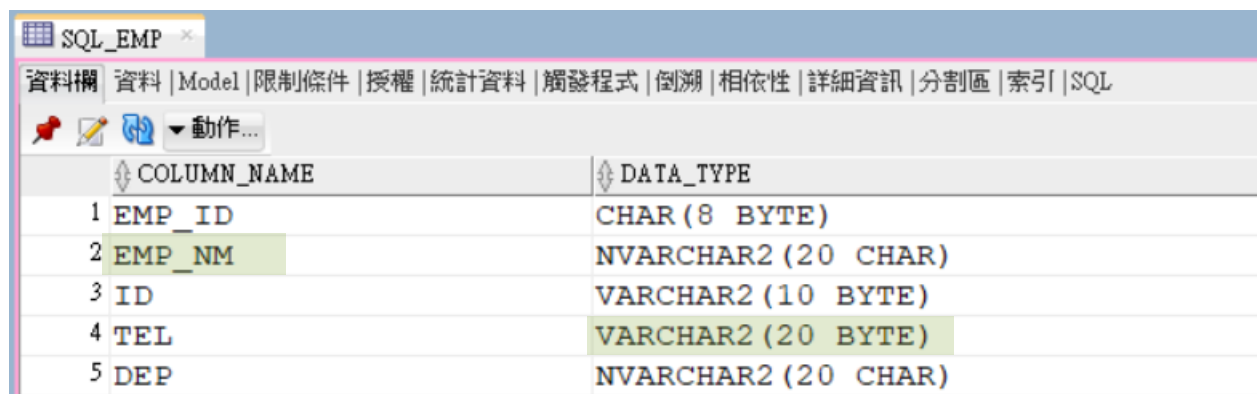
SQL_EMP		
資料欄 資料 Model 限制條件 授權 統計資料 觸發程式 倒溯 相依性 詳細資訊 分割區 索引 SQL		
📌 ✎ 🔗 ▼ 動作...		
	↕ COLUMN_NAME	↕ DATA_TYPE
1	EMP_ID	CHAR(8 BYTE)
2	EMP_NAME	NVARCHAR2(20 CHAR)
3	ID	VARCHAR2(10 BYTE)
4	TEL	CHAR(10 BYTE)
5	DEP	NVARCHAR2(20 CHAR)

課堂練習①

- 將人事資料表更改欄位名稱及型別。



	COLUMN_NAME	DATA_TYPE
1	EMP_ID	CHAR(8 BYTE)
2	EMP_NAME	NVARCHAR2(20 CHAR)
3	ID	VARCHAR2(10 BYTE)
4	TEL	CHAR(10 BYTE)
5	DEP	NVARCHAR2(20 CHAR)



	COLUMN_NAME	DATA_TYPE
1	EMP_ID	CHAR(8 BYTE)
2	EMP_NM	NVARCHAR2(20 CHAR)
3	ID	VARCHAR2(10 BYTE)
4	TEL	VARCHAR2(20 BYTE)
5	DEP	NVARCHAR2(20 CHAR)

資料限制與關聯 - P.K.

- Primary Key (**P.K.**)

- 用來識別資料表中記錄的唯一值的鍵值 (欄位值) ，稱為「**主鍵**」。

主鍵 P.K.

<u>行編</u>	姓名	身分證字號	分機	單位
-----------	----	-------	----	----

挑選原則

① 不可以為空值或重覆：

依照「關聯式資料完整性規則」，主鍵的鍵值**不可以重覆**，也**不可以為空值 (NotNull)**。

例如：「姓名」欄位就不適合當作主鍵欄位，因為可能會重複。

② 固定不會再變更的值：

否則會增加爾後的管理和維護資料的困難度與複雜性。

例如：「分機」欄位就不適合當作主鍵欄位，因為可能變更。

③ 單一的屬性：

在一個資料表中，**最好**只選取「單一屬性」作為主鍵，因為可以節省記憶體空間及提高執行效率。

例如：「姓名 + 員工編號」與「員工編號」，雖然二者都具有唯一性，但是後者「員工編號」是單一屬性。

若需要多組欄位才能辨識一組資料稱為「複合主鍵」。



資料限制與關聯 - P.K.

- 主鍵建置時機：
 - 新建表格 (create table)
 - 改變現有的表格 (alter table)

【畫面】

PK	名稱	資料類型	大小	非空值	預設	註解
<input checked="" type="checkbox"/>	EMP_ID	CHAR	8	<input checked="" type="checkbox"/>		行編
<input type="checkbox"/>	EMP_NM	NVARCHAR2	20	<input checked="" type="checkbox"/>		姓名
<input type="checkbox"/>	ID	VARCHAR2	10	<input type="checkbox"/>		身分證字號
<input type="checkbox"/>	TEL	VARCHAR2	10	<input type="checkbox"/>		分機
<input type="checkbox"/>	DEP	NVARCHAR2	20	<input type="checkbox"/>		單位

【語法】

◆ 新增表格時

```
create table SQL_EMP (  
  EMP_ID CHAR(8 BYTE) primary key,  
  EMP_NM NVARCHAR2(20),  
  ID VARCHAR2(10 BYTE),  
  TEL VARCHAR2(10 BYTE),  
  DEP NVARCHAR2(20));
```


◆ 改變既有表格時

```
alter table SQL_EMP add primary key (EMP_ID);
```

課堂練習②

- 替練習 1 的人事資料表新增 EMP_ID 為 PK

行編	姓名	身分證字號	分機	單位
00012345	王○明	A100100100	1000#1010	系統開發部
00054321	孫○拉	B200200200	1000#1020	開發系統部






SQL_EMP	
資料欄	資料 Model 限制條件 授權 統計資料 觸發
	動作...
COLUMN_NAME	DATA_TYPE
1 EMP_ID	CHAR(8 BYTE)
2 EMP_NM	NVARCHAR2(20 CHAR)
3 ID	VARCHAR2(10 BYTE)
4 TEL	VARCHAR2(10 BYTE)
5 DEP	NVARCHAR2(20 CHAR)

STUDENT.SQL_EMP	
P	EMP_ID CHAR(8 BYTE)
	EMP_NM NVARCHAR2(20)
	ID VARCHAR2(10 BYTE)
	TEL VARCHAR2(10 BYTE)
	DEP NVARCHAR2(20)
SQL_EMP_PK (EMP_ID)	

課堂練習②

- 建立客戶資料表並將 ID 設為 PK，table 名稱 CUSTOMER_學號。

身分證字號	姓名	手機	理專行編
C300300300	陳○圓	0930300300	00012345
E500500500	李○人	0950500500	

SQL_CUSTOMER	
資料欄	資料 Model 限制條件 授權 統計資料 觸發程式 倒溯 相依性 詳細資訊 分割區 索引 SQL
  	▼ 動作...

STUDENT.SQL_CUSTOMER		
P	ID	CHAR(10 BYTE)
	NAME	NVARCHAR2(20)
F	EMP	CHAR(8 BYTE)
👉 SQL_CUSTOMER_PK (ID)		

資料限制與關聯 - F.K.

- Foreign Key (F.K.)

- 任兩個資料表要進行關聯時，須透過「外來鍵」參考「主鍵」才能建立。



主要目的是用來確定資料的參考完整性。

「主鍵」值的所在資料表稱為「父關聯」。

「外來鍵」值的所在資料表稱為「子關聯」。

➤ 「子關聯」的外鍵必須對應「父關聯」的主鍵。

【注意】

- 欄位必須要具有相同的資料型態和欄位長度。
- 欄位名稱則可以不相同。
- 外來鍵的欄位值可以是重複值或空值。

✓ 先新增「父關聯」再新增「子關聯」

✓ 先移除「子關聯」才能刪除「父關聯」

資料限制與關聯 - F.K.

- 外來鍵建置時機：
 - 新建表格 (create table)
 - 改變現有的表格 (alter table)

【畫面】

編輯表格

綱要(A): STUDENT
名稱(B): SQL_CUSTOMER
表格類型(C): 一般

搜尋

資料欄
限制條件
索引
記憶體內
儲存體
註解
DDL

限制條件(D): 名稱

類型	名稱	啟用	可延續狀態
主索引鍵	SQL_CUSTOMER_PK	<input checked="" type="checkbox"/>	不可延續
外來索引鍵	SQL_CUSTOMER_FK1	<input checked="" type="checkbox"/>	不可延續

— 參照的限制條件

綱要(E): STUDENT
表格(F): SQL_EMP
限制條件(G): SQL_EMP_PK
刪除時(H): 不執行任何動作

關聯(I)(J):

本檔資料欄	參照的資料欄
EMP	EMP_ID

說明(I): 確定 取消

【語法】

◆ 新增表格時

```
create table SQL_CUSTOMER (  
  ID VARCHAR2(10 BYTE) primary key,  
  NAME NVARCHAR2(20),  
  EMP CHAR(8 BYTE) references SQL_EMP (EMP_ID))
```

◆ 改變既有表格時

```
alter table SQL_CUSTOMER  
add foreign key (EMP) references SQL_EMP (EMP_ID);
```


◆ 替外鍵命名與多欄位的外鍵：

```
alter table SQL_CUSTOMER add constraint SQL_CUSTOMER_FK1  
foreign key (EMP) references SQL_EMP (EMP_ID);
```

課堂練習③

- 將練習2的客戶資料表設定 FK，FK 來源為練習1 人事資料表的 EMP_ID

身分證字號	姓名	手機	理專行編
C300300300	陳○圓	0930300300	00012345
E500500500	李○人	0950500500	



SQL_CUSTMER	
資料欄	資料 Model 限制條件 授權 統計資料 觸發程式 倒溯 相依性 詳細資訊 分割區 索引 SQL
	動作...
COLUMN_NAME	DATA_TYPE
1 ID	CHAR(10 BYTE)
2 NAME	NVARCHAR2(20 CHAR)
3 EMP	CHAR(8 BYTE)

STUDENT.SQL_CUSTMER		
P	ID	CHAR(10 BYTE)
	NAME	NVARCHAR2(20)
F	EMP	CHAR(8 BYTE)
SQL_CUSTMER_PK (ID)		

課堂練習 參考解答

- 練習 1

新建 table

```
create table EMP_XX ( EMP_ID CHAR(8),EMP_NAME NVARCHAR2(20),ID VARCHAR2(10), TEL CHAR(10), DEP NVARCHAR2(20) );
```

修改 table

```
alter table EMP_XX modify TEL VARCHAR2(20);
```

```
alter table EMP_XX rename column EMP_NAME to EMP_NM;
```

- 練習 2

新建 table 時，設定 PK

```
create table CUSTOMER_XX (EMP_ID CHAR(8 BYTE) primary key, MP_NM NVARCHAR2(20), ID VARCHAR2(10 BYTE), TEL VARCHAR2(10 BYTE), DEP NVARCHAR2(20));
```

舊有 Table 加上 PK

```
alter table EMP_XX add primary key (EMP_ID);
```

- 練習 3

修改舊有 table 的 FK

```
alter table CUSTOMER_XX add foreign key (EMP_ID) references EMP_XX (EMP_ID);
```

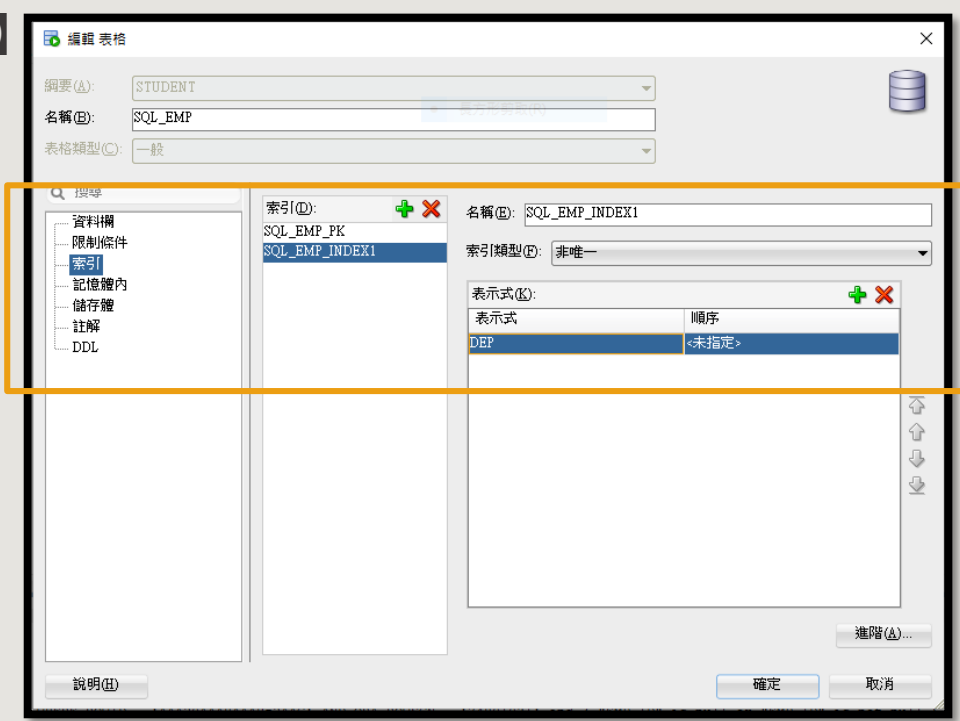
commit ;



資料索引 Index

- 可加快搜尋速度
 - 如同書的目錄，可以快速地翻到內容所在的位置
 - 但注意它是以空間換取時間
- 可多欄位設定為 Index

【畫面】



【語法】

◆ 新增索引

create index SQL_EMP_INDEX1 **on** SQL_EMP (DEP)

【觀念補充】

- 索引不是越多越好
 - 建立過多 index 會占用儲存空間
 - 資料增刪修時也需要更新索引
- 索引欄位長度選擇
 - 越短越好、固定比變動好
- 單列索引與多列索引：
 - 「A + B + C」設為 index 時，僅搜尋 A 是不會用到的
 - 「A + B + C」和「A」index 效果不同，如搜尋 A + B + C
 - 前者直接回覆 A+B+C 索引值
 - 後者為 A 用索引縮小範圍後，再搜尋 B + C

Data Query Language , DQL

資料查詢

查詢語法結構

select 欄位名
from 表格名
where 條件 ;

【範例】

取得人事檔裡單位是系統開發部的姓名資料。

select 姓名
from 人事檔
where 單位 = '系統開發部'

字串使用單引號

行編	姓名	身分證字號	分機	單位
00012345	王○明	A100100100	1000#1010	系統開發部
00054321	孫○拉	B200200200	1000#1020	開發系統部

欄位資料

- 從表中取得**所需的欄位**數據：

① 可條列欄位名稱（以，串接）

```
SELECT column1, column2, ...  
FROM table_name;
```

② 「*」可以查詢表中全部的欄位

```
SELECT * FROM table_name;
```

【範例】

```
select EMP_ID, EMP_NM  
from SQL_EMP;
```

EMP_ID	EMP_NM
00012345	王○明
00054321	孫○拉

```
select *  
from SQL_EMP;
```

EMP_ID	EMP_NM	ID	TEL	DEP
00012345	王○明	A100100100	1000#1010	系統開發部
00054321	孫○拉	B200200200	1000#1020	開發系統部

- 運用 **as** 來指定欄位別名。
- 表格別名直接於表格名後用“ ”空格銜接

select 表格別名.欄位名 **as** 欄位別名
from 表格名 表格別名;

【範例】

select EMP.EMP_NM as NAME
from SQL_EMP EMP;

NAME
王○明
孫○拉

select EMP_ID as ‘行編’, EMP_NM as ‘姓名’
from SQL_EMP;

行編	姓名
00012345	王○明
00054321	孫○拉

課堂練習④

- 請查詢 TAOYUAN_WATER 表格，並將搜尋結果呈現如下：
- 問題 1：欄位名稱可以使用其他語言嗎？
- 問題 2：請問搜尋的結果是有序的嗎？
- 問題 3：請問在**使用表格別名**跟**不使用表格別名**有甚麼差異？

	SERIAL_NUMBER	HYDROGEN	TURBIDITY	CHLORINE
1	532	7.7	0.7	0.58
2	533	7.7	0.6	0.67
3	534	7.6	0.3	0.59
4	535	7.7	0.4	0.71
5	536	7.7	0.6	0.78
6	537	7.7	0.6	0.77
7	538	7.6	0.6	0.67
8	539	7.5	0.4	0.65
9	540	7.4	0.2	0.78
10	541	7.9	0.4	0.5
11	542	7.9	0.3	0.4
12	543	7.7	0.4	0.62
13	544	7.8	0.7	0.74
14	545	7.7	0.4	0.44
15	546	7.4	0.4	0.73

	序號	氫離子濃度指數	濁度	自由有效餘氯
1	521	7.2	0.4	0.79
2	522	7.8	0.7	0.7
3	523	7.8	0.2	0.7
4	524	7.8	0.4	0.66
5	525	7.7	0.6	0.67
6	526	7.7	0.6	0.67
7	527	8	0.4	0.48
8	528	7.7	0.6	0.66
9	529	7.7	1	0.62
10	537	7.7	0.6	0.77
11	644	7.8	0.6	0.77
12	694	7.8	0.6	0.77

where 條件

- 指定查詢所傳回之資料列的搜尋條件。

Operator	Description
=	等於
<>	不等於 (或是 !=)
between	在某範圍內
like	相似，用於模糊查詢
in	一次給定多個條件值 (擇一符合即可)

Operator	Description	Operator	Description
>	大於	<	小於
>=	大於等於	<=	小於等於
and	多個條件皆符合	or	多個條件擇一符合

等於

where 欄位名 = '條件值'

不等於 <> 用法相似。

【範例】

```
select * from SQL_EMP  
where DEP = '系統開發部';
```

```
select * from SQL_EMP  
where DEP <> '開發系統部';
```

EMP_ID	EMP_NM	ID	TEL	DEP
00012345	王○明	A100100100	1000#1010	系統開發部

比較運算子

- 大於 >、小於 <、大於等於 >=、小於等於 <=

where 欄位名 比較運算子 '條件值'

【範例】

行編大於 00051111 的人事資料

```
select * from SQL_EMP
```

```
where EMP_ID > '00051111';
```

EMP_ID	EMP_NM	ID	TEL	DEP
00054321	孫○拉	B200200200	1000#1020	開發系統部

BETWEEN

- 尋找位在兩條件值之間資料列

where 欄位名 **between** '起始條件值' **and** '結束條件值'

【範例】

行編介於 00011111 ~ 00055555 的人事資料

```
select * from SQL_EMP
```

```
where EMP_ID between '00011111' and '00055555';
```

EMP_ID	EMP_NM	ID	TEL	DEP
00012345	王○明	A100100100	1000#1010	系統開發部
00054321	孫○拉	B200200200	1000#1020	開發系統部

NULL

- 判斷欄位是否為空值

where 欄位名 **is null**

where 欄位名 **is not null**

【範例】

沒有設定分機的人事資料

```
select * from SQL_EMP  
where TEL is null;
```

LIKE

• 模糊查詢，符合指定格式，搭配關鍵符號：

- `_`：1 個字元
- `%`：0 ~ n 個字元

where 欄位名 **like** '指定格式'

索引失效：若將萬用字元（%）置於關鍵詞前方就起不了作用（但後方可以）。

【範例】

條件	指定格式	符合資料	不合資料
姓林的人名	like '林%'	林、林依、林依俄參	李寺
叫林O的	like '林_'	林依、林俄	林依俄
叫OO偉的	like '__偉'	林依偉、王大偉	林偉、陳偉人
名字裡有林的	like '%林%'	林依、王林、陳林琳	李寺、陳偉人

邏輯運算子

- 且 and 、或 or
- 可搭配 () 組成條件群

EMP_ID	EMP_NM	ID	TEL	DEP
00012345	王○明	A100100100	1000#1010	系統開發部
00054321	孫○拉	B200200200	1000#1020	開發系統部

where 條件1 邏輯運算子 條件2 邏輯運算子 條件3...

【範例】

系統開發部姓王的資料

```
select * from SQL_EMP  
where DEP = '系統開發部' and EMP_NM like '王%';
```

系統開發部、開發系統部的資料

```
select * from SQL_EMP  
where DEP = '系統開發部' or DEP = '開發系統部';
```

系統開發部姓王、開發系統部姓孫的資料

```
select * from SQL_EMP  
where  
    (DEP = '系統開發部' and EMP_NM like '王%')  
or  
    (DEP = '開發系統部' and EMP_NM like '孫%')
```



- 一個欄位想要設定多個條件值
- 優化同一欄位 or 條件
 - where DEP = '系統開發部' or DEP = '開發系統部' or DEP = '開發開發部'

where 欄位 in ('條件值1', '條件值2', ...)

【範例】

```
select * from SQL_EMP  
where DEP in ('系統開發部', '開發系統部');
```

EMP_ID	EMP_NM	ID	TEL	DEP
00012345	王○明	A100100100	1000#1010	系統開發部
00054321	孫○拉	B200200200	1000#1020	開發系統部

課堂練習⑤

- 請查詢 TAOYUAN_WATER 表格

問題 1：

- 氫離子濃度指數符合 7.5 或 8
- 自由有效餘氯大於 0.6
- 地點為 T22

	序號	氫離子濃度指數	濁度	自由有效餘氯
1	603	8	0.7	0.82
2	653	8	0.7	0.82
3	188	7.5	0.7	0.62

問題 2：

- 序號是 521 到 529 之間
- 序號不在此區間，但符合以下敘述
自由有效餘氯是 0.76、0.77、0.66
且氫離子濃度指數不是 7.8
且濁度是 0.6

	序號	氫離子濃度指數	濁度	自由有效餘氯
1	521	7.2	0.4	0.79
2	522	7.8	0.7	0.7
3	523	7.8	0.2	0.7
4	524	7.8	0.4	0.66
5	525	7.7	0.6	0.67
6	526	7.7	0.6	0.67
7	527	8	0.4	0.48
8	528	7.7	0.6	0.66
9	529	7.7	1	0.62
10	537	7.7	0.6	0.77
11	644	7.8	0.6	0.77
12	694	7.8	0.6	0.77

提示：這種情況下，要如何分離不同邏輯？

課堂練習⑤ 參考解答

問題 1 :

```
select  
    SERIAL_NUMBER as 序號 ,HYDROGEN as 氫離子濃度指數,TURBIDITY as 濁度 ,CHLORINE as 自由有效餘氯  
from TAOYUAN_WATER  
where HYDROGEN in(7.5,8) and CHLORINE > 0.6 and PLACE = 'T22';
```

問題 2 :

```
select  
    SERIAL_NUMBER as 序號 ,HYDROGEN as 氫離子濃度指數,TURBIDITY as 濁度 ,CHLORINE as 自由有效餘氯  
from TAOYUAN_WATER  
where (SERIAL_NUMBER > 520 and SERIAL_NUMBER < 530) or (CHLORINE in (0.77 , 0.76) and TURBIDITY = 0.6);
```

DISTINCT

- 移除重複資料組，每組只留一筆
 - 出現的次數並不重要

```
SELECT DISTINCT column1, column2, ...  
FROM table_name;
```

EMP_ID	EMP_NM	ID	TEL	DEP
00012345	王○明	A100100100	1000#1010	系統開發部
00012468	王○明	A200200200	1000#1012	系統開發部
00054321	孫○拉	B200200200	1000#1020	開發系統部
00055333	王○明	C300300300	1000#1030	開發系統部

【範例】

公司有哪些部門

```
select distinct DEP  
from SQL_EMP;
```

DEP
系統開發部
開發系統部

```
select distinct DEP, EMP_NM  
from SQL_EMP;
```

可視為 distinct (DEP, EMP_NM)

DEP	EMP_NM
系統開發部	王○明
開發系統部	孫○拉
開發系統部	王○明

ORDER BY

資料取出後可指定欄位值排序

- asc (ascending) : 正向，由小到大，由舊到新 (預設)
- desc (descending) : 反向，由大到小，由新到舊

```
SELECT column1, column2, ...  
FROM table_name  
ORDER BY column1, column2, ... ASC|DESC;
```

EMP_ID	EMP_NM	ID	TEL	DEP
00012345	王○明	A100100100	1000#1010	系統開發部
00055333	王○茗	C300300300	1000#1030	開發系統部
00012468	王○名	A200200200	1000#1012	系統開發部
00054321	孫○拉	B200200200	1000#1020	開發系統部

【範例】

行編由大到小的名單

```
select EMP_NM  
from SQL_EMP  
order by EMP_ID desc;
```

EMP_NM
王○茗
孫○拉
王○名
王○明

依部門正向排序，部內再依 ID 反向

```
select DEP, ID, EMP_NM  
from SQL_EMP  
order by DEP asc, ID desc;
```

或寫成 order by DEP, ID desc

DEP	ID	EMP_NM
系統開發部	A200200200	王○名
系統開發部	A100100100	王○明
開發系統部	C300300300	王○茗
開發系統部	B200200200	孫○拉

課堂練習⑥

- 呈練習 5 第二題

- 條件：

1. 查詢資料欄位加上地點

2. 排除相同地點

提示：排除相同的關鍵字是.....？該放在哪個位置？

3. 資料排序優先濁度大到小，再排序號小到大

提示：複合排序該怎麼去分隔？

	地點	序號	氫離子濃度指數	濁度	自由有效餘氯
1	T26	523	7.8	0.2	0.7
2	T33	527	8	0.4	0.48
3	T26	524	7.8	0.4	0.66
4	T31	521	7.2	0.4	0.79
5	T40	537	7.7	0.6	0.77
6	T36	528	7.7	0.6	0.66
7	T21	526	7.7	0.6	0.67
8	T19	525	7.7	0.6	0.67
9	T23	515	7.7	0.6	0.66
10	T26	522	7.8	0.7	0.7
11	T51	529	7.7	1	0.62

課堂練習⑥

```
select (distinct PLACE as 地點), SERIAL_NUMBER as 序號 ,HYDROGEN as 氫離子濃度指數,TURBIDITY as 濁  
度,CHLORINE as 自由有效餘氯  
from TAOYUAN_WATER  
where (SERIAL_NUMBER > 520 and SERIAL_NUMBER < 530) or CHLORINE in (0.77 , 0.76,0.66)and HYDROGEN!=7.8  
and TURBIDITY = 0.6  
order by TURBIDITY asc , SERIAL_NUMBER desc ;
```


多張表格的關聯查詢

人事檔
SQL_EMP

EMP_ID	EMP_NM	ID	TEL	DEP
00012345	王○明	A100100100	1000#1010	系統開發部
00054321	孫○拉	B200200200	1000#1020	開發系統部



客戶檔
SQL_CUSTOMER

ID	NAME	EMP
C300300300	陳○圓	00012345
D400400400	郭○多	00012345
A100100100	王○明	



子查詢

- 子查詢是指巢狀於 SQL 陳述式中的查詢
 - 在一個 SQL 語句中放入另一個 select 語句
 - 通常是位在 select 的 where 子句，透過子查詢取得查詢條件

where 欄位 比較運算子 (select ...)

【範例】

有當客戶理專的員工資料

select *

from SQL_EMP

where EMP_ID in (select distinct EMP from SQL_CUSTOMER);

EMP_ID	EMP_NM	ID	TEL	DEP
00012345	王○明	A100100100	1000#1010	系統開發部
00054321	孫○拉	B200200200	1000#1020	開發系統部

ID	NAME	EMP
C300300300	陳○圓	00012345
A100100100	王○明	00012345

EMP_ID	EMP_NM	ID	TEL	DEP
00012345	王○明	A100100100	1000#1010	系統開發部

UNION ALL & UNION

- 將兩個 SQL 結果合併在一起。

- 查詢欄位數、順序、資料型態需一致

① 列出所有符合條件的資料，不剔除重複。

```
SELECT column_name(s) FROM table1  
UNION ALL  
SELECT column_name(s) FROM table2;
```

② 重複只出現 1 次 (distinct)。

```
SELECT column_name(s) FROM table1  
UNION  
SELECT column_name(s) FROM table2;
```

【範例】

所有姓名和 ID (不分員工 or 客戶)

```
select EMP_NM, ID from SQL_EMP
```

```
union all
```

```
select NAME, ID from SQL_CUSTOMER;
```

union all

EMP_NM	ID
王○明	A100100100
孫○拉	B200200200
陳○圓	C300300300
王○明	A100100100

union

EMP_NM	ID
王○明	A100100100
孫○拉	B200200200
陳○圓	C300300300

課堂練習⑦

- 呈練習 6，利用練習 5 的SQL語法進行改寫
- 問題 1：利用子查詢、 UNION ALL 及 UNION 語法改寫

提示：UNION ALL 及 UNION 的情況下 ORDER BY 可以直接使用嗎？

- 問題 2：請問哪個結果與練習 5 的結果相同？
- 問題 3：若有不相同，是哪裡有不相同處？

提示：資料增加或者是資料減少

	地點	序號	氫離子濃度指數	濁度	自由有效餘氯
1	T26	523	7.8	0.2	0.7
2	T33	527	8	0.4	0.48
3	T26	524	7.8	0.4	0.66
4	T31	521	7.2	0.4	0.79
5	T40	537	7.7	0.6	0.77
6	T36	528	7.7	0.6	0.66
7	T21	526	7.7	0.6	0.67
8	T19	525	7.7	0.6	0.67
9	T26	522	7.8	0.7	0.7
10	T51	529	7.7	1	0.62

課堂練習⑦

select 地點,序號,氫離子濃度指數,濁度,自由有效餘氯 from (

select PLACE as 地點,SERIAL_NUMBER as 序號,HYDROGEN as 氫離子濃度指數,TURBIDITY as 濁度,CHLORINE as 自由有效餘氯
from TAOYUAN_WATER

where (SERIAL_NUMBER> 520 and SERIAL_NUMBER < 530)

union all

select PLACE as 地點,SERIAL_NUMBER as 序號,HYDROGEN as 氫離子濃度指數,TURBIDITY as 濁度,CHLORINE as 自由有效餘氯
from TAOYUAN_WATER

where CHLORINE in (0.77, 0.76,0.66)and HYDROGEN!=7.8 and TURBIDITY = 0.6)

order by 濁度 asc, 序號 desc

Union

	地點	序號	氫離子濃度指數	濁度	自由有效餘氯
1	T19	525	7.7	0.6	0.67
2	T21	526	7.7	0.6	0.67
3	T23	515	7.7	0.6	0.66
4	T26	522	7.8	0.7	0.7
5	T26	523	7.8	0.2	0.7
6	T26	524	7.8	0.4	0.66
7	T31	521	7.2	0.4	0.79
8	T33	527	8	0.4	0.48
9	T36	528	7.7	0.6	0.66
10	T40	537	7.7	0.6	0.77
11	T51	529	7.7	1	0.62

Union all

	地點	序號	氫離子濃度指數	濁度	自由有效餘氯
1	T31	521	7.2	0.4	0.79
2	T26	522	7.8	0.7	0.7
3	T26	523	7.8	0.2	0.7
4	T26	524	7.8	0.4	0.66
5	T19	525	7.7	0.6	0.67
6	T21	526	7.7	0.6	0.67
7	T33	527	8	0.4	0.48
8	T36	528	7.7	0.6	0.66
9	T51	529	7.7	1	0.62
10	T40	537	7.7	0.6	0.77
11	T23	515	7.7	0.6	0.66
12	T36	528	7.7	0.6	0.66

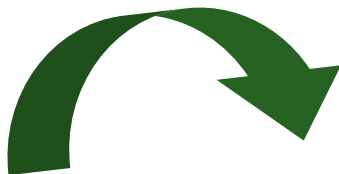
JOIN

- 利用不同資料表之間欄位的**關聯性**，來**結合多個資料表**而組成一抽象的**暫時性**資料表以供資料查詢。

透過關聯欄位
SQL_CUSTOMER.EMP = SQL_EMP.EMP_ID
可以知道該客戶的理專人事資訊

客戶檔
SQL_CUSTOMER

ID	NAME	EMP
C300300300	陳○圓	00012345
D400400400	郭○多	00012345
A100100100	王○明	



人事檔
SQL_EMP

EMP_ID	EMP_NM	ID	TEL	DEP
00012345	王○明	A100100100	1000#1010	系統開發部
00054321	孫○拉	B200200200	1000#1020	開發系統部

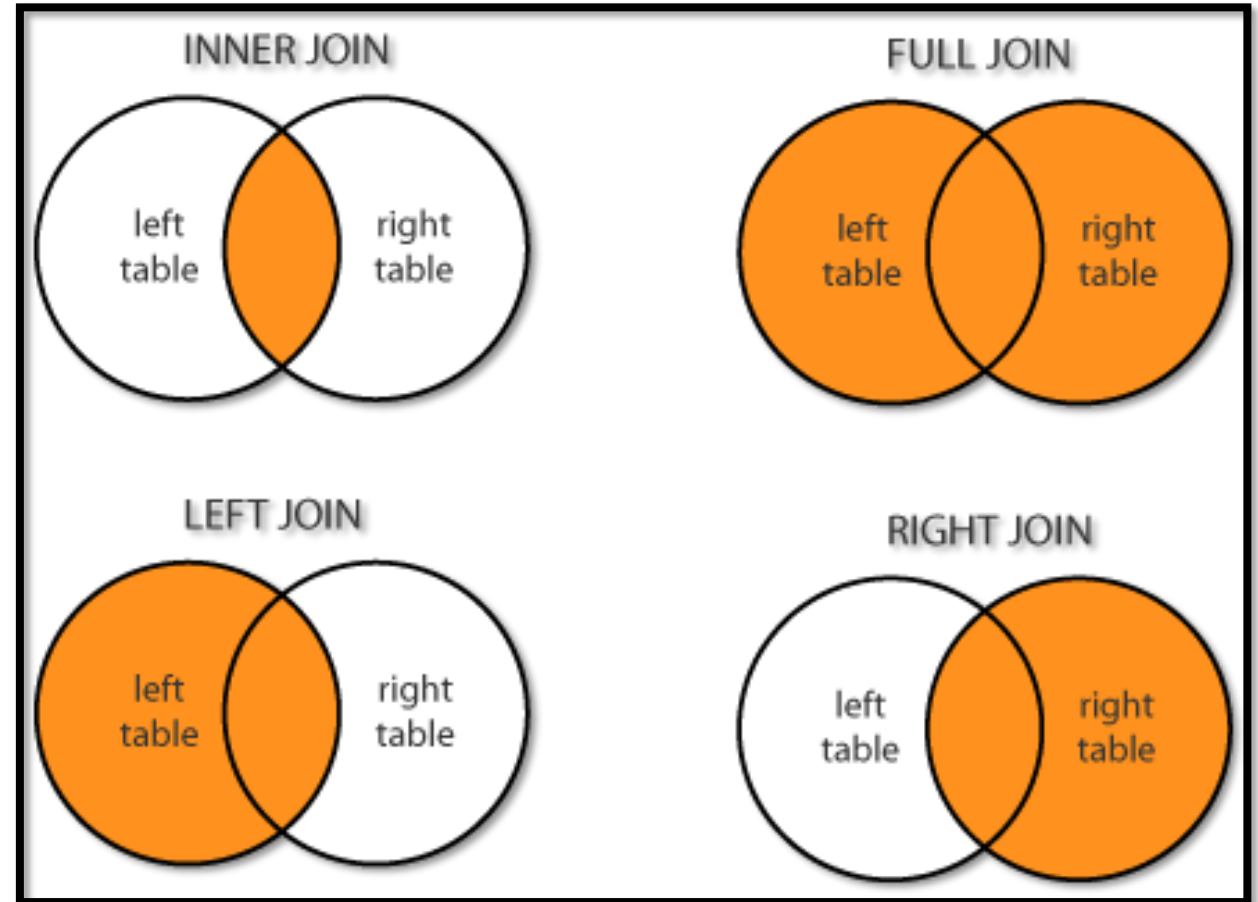
➤ 依關聯條件可能為 1:n 或 n:n 排列組合結果

JOIN

select 欄位
from 左表格名
join 類別 右表格名 on 關聯性;

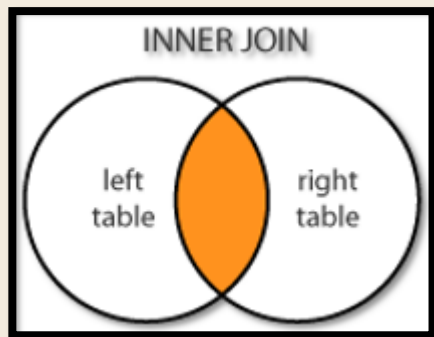
左表格名 join 右表格名

select 欄位
from 左表格名
join 類別 右1表格名 on 關聯性1
join 類別 右2表格名 on 關聯性2
...;



INNER JOIN

- 取得關聯欄位兩邊都有的資料。



ID	NAME	EMP
C300300300	陳○圓	00012345
D400400400	郭○多	00012345
A100100100	王○明	

EMP_ID	EMP_NM	ID	TEL	DEP
00012345	王○明	A100100100	1000#1010	系統開發部
00054321	孫○拉	B200200200	1000#1020	開發系統部

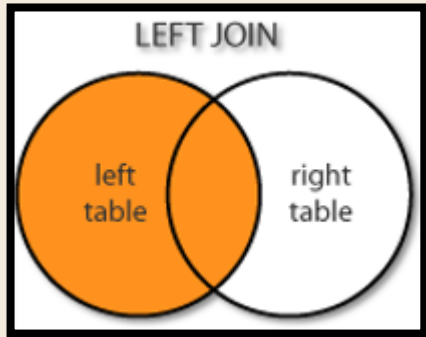
【範例】

```
select CUT.ID, CUT.NAME, EMP.EMP_NM  
from SQL_CUSTOMER CUT  
inner join SQL_EMP EMP on CUT.EMP = EMP.EMP_ID
```

ID	NAME	EMP_NM
C300300300	陳○圓	王○明
D400400400	郭○多	王○明

LEFT JOIN

- 以左邊表格資料為主，有關聯到右邊就一併帶出。



ID	NAME	EMP
C300300300	陳○圓	00012345
D400400400	郭○多	00012345
A100100100	王○明	

EMP_ID	EMP_NM	ID	TEL	DEP
00012345	王○明	A100100100	1000#1010	系統開發部
00054321	孫○拉	B200200200	1000#1020	開發系統部

【範例】

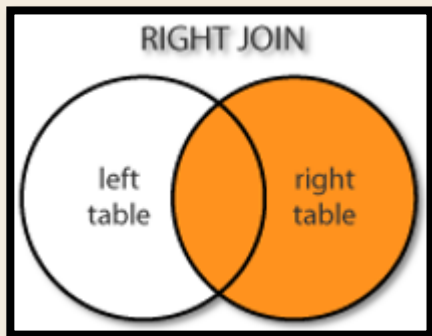
```
select CUT.ID, CUT.NAME, EMP.EMP_NM  
from SQL_CUSTOMER CUT  
left join SQL_EMP EMP on CUT.EMP = EMP.EMP_ID
```

ID	NAME	EMP_NM
C300300300	陳○圓	王○明
D400400400	郭○多	王○明
A100100100	王○明	null

沒有關聯到的欄位就是 null

RIGHT JOIN

- 以右邊表格資料為主，有關聯到左邊就一併帶出。



ID	NAME	EMP
C300300300	陳○圓	00012345
D400400400	郭○多	00012345
A100100100	王○明	

EMP_ID	EMP_NM	ID	TEL	DEP
00012345	王○明	A100100100	1000#1010	系統開發部
00054321	孫○拉	B200200200	1000#1020	開發系統部

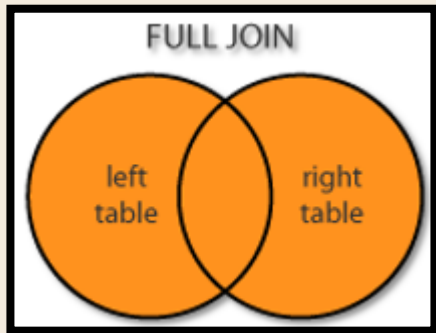
【範例】

```
select CUT.ID, CUT.NAME, EMP.EMP_NM  
from SQL_CUSTOMER CUT  
right join SQL_EMP EMP on CUT.EMP = EMP.EMP_ID
```

ID	NAME	EMP_NM
C300300300	陳○圓	王○明
D400400400	郭○多	王○明
null	null	孫○拉

FULL JOIN

- 所有兩邊表格資料都會出現並帶出關聯的另一邊，為 LEFT JOIN 與 RIGHT JOIN 的聯集。



ID	NAME	EMP
C300300300	陳○圓	00012345
D400400400	郭○多	00012345
A100100100	王○明	

EMP_ID	EMP_NM	ID	TEL	DEP
00012345	王○明	A100100100	1000#1010	系統開發部
00054321	孫○拉	B200200200	1000#1020	開發系統部

【範例】

```
select CUT.ID, CUT.NAME, EMP.EMP_NM  
from SQL_CUSTOMER CUT  
full join SQL_EMP EMP on CUT.EMP = EMP.EMP_ID
```

ID	NAME	EMP_NM
C300300300	陳○圓	王○明
D400400400	郭○多	王○明
null	null	孫○拉
A100100100	王○明	null

JOIN 補充

- 特定條件關聯。

ID	NAME	EMP	EMP_ID	EMP_NM	ID	TEL	DEP
C300300300	陳○圓	00012345	00012345	王○明	A100100100	1000#1010	系統開發部
D400400400	郭○多	00012345	00054321	孫○拉	B200200200	1000#1020	開發系統部
E500500500	李○人	00054321					
A100100100	王○明						

【範例】

取得所有客戶姓名，若理專是系統開發部就連帶理專姓名

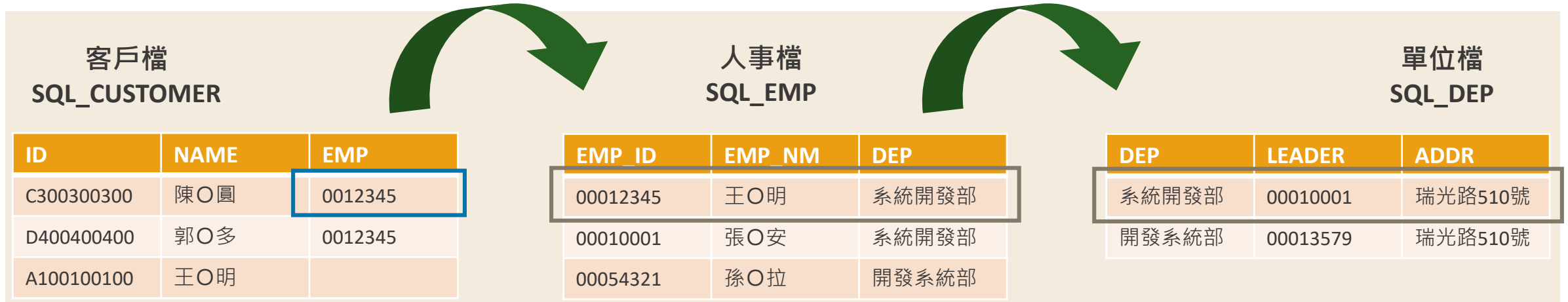
```
select CUT.NAME, EMP.EMP_NM
```

```
from SQL_CUSTOMER CUT
```

```
left join SQL_EMP EMP on CUT.EMP = EMP.EMP_ID and EMP.DEP = '系統開發部'
```

NAME	EMP_NM
陳○圓	王○明
郭○多	王○明
李○人	null
王○明	null

JOIN 補充



【範例】

取得所有客戶姓名並連帶理專姓名和辦公地點

```
select CUT.NAME, EMP.EMP_NM, DEP.ADDR
from SQL_CUSTOMER CUT
left join SQL_EMP EMP on CUT.EMP = EMP.EMP_ID
left join SQL_DEP DEP on EMP.DEP = DEP.DEP
```

NAME	EMP_NM	ADDR
陳○圓	王○明	瑞光路510號
郭○多	王○明	瑞光路510號
王○明	null	null

JOIN 補充

客戶檔
SQL_CUSTOMER

ID	NAME	EMP
C300300300	陳○圓	0012345
D400400400	郭○多	0012345
A100100100	王○明	

人事檔
SQL_EMP

EMP_ID	EMP_NM	ID	TEL	DEP
00012345	王○明	A100100100	1000#1010	系統開發部
00054321	孫○拉	B200200200	1000#1020	開發系統部

【範例】

是客戶也是員工的姓名

```
select CUT.NAME  
from SQL_CUSTOMER CUT  
inner join SQL_EMP EMP on CUT.ID = EMP.ID
```

NAME

王○明

課堂練習⑧

- 呈練習 7，利用練習 7 的SQL語法進行改寫

- 條件：

- 地點編號和Taoyuan_regions 表格進行JOIN
- 供水系統編號和Taoyuan_water_system 表格進行JOIN

提示：原本的語法足夠進行join嗎？有需要補上欄位嗎？

- 欄位刪除地點，並加上淨水廠、供水系統、行政區、路段

	序號	氫離子濃度指數	濁度	自由有效餘氯	淨水廠	供水系統	行政區	路段
1	523	7.8	0.2	0.7	平鎮淨水場	石門區	平鎮區	南豐路
2	527	8	0.4	0.48	復興淨水場	復興區	復興區	中正路
3	524	7.8	0.4	0.66	平鎮淨水場	石門區	平鎮區	南豐路
4	521	7.2	0.4	0.79	大湳淨水場	石門區	桃園區	國際路
5	537	7.7	0.6	0.77	龍潭淨水場	石門區	龍潭區	十一分路
6	528	7.7	0.6	0.66	平鎮淨水場	石門區	楊梅區	中山路
7	528	7.7	0.6	0.66	平鎮淨水場	石門區	楊梅區	中山路
8	526	7.7	0.6	0.67	平鎮淨水場	石門區	中壢區	龍昌路
9	525	7.7	0.6	0.67	平鎮淨水場	石門區	中壢區	中正路
10	515	7.7	0.6	0.66	石門淨水場	石門區	平鎮區	大昌路
11	522	7.8	0.7	0.7	平鎮淨水場	石門區	平鎮區	南豐路
12	529	7.7	1	0.62	平鎮淨水場	石門區	觀音區	大觀路



課堂練習⑧ 參考解答

```
select 序號,氫離子濃度指數,濁度,自由有效餘氯, WATER_PURIFICATION_PLANT as 淨水廠, WATER_SUPPLY as 供水系統, TOWNSHIP
as 行政區, PLACE as 路段
from (
    select PLACE as 地點, SERIAL_NUMBER as 序號, HYDROGEN as 氫離子濃度指數, TURBIDITY as 濁度, CHLORINE as 自由有效餘氯,
    WATER_SYSTEM as 供水系統
    from TAOYUAN_WATER
    where (SERIAL_NUMBER > 520 and SERIAL_NUMBER < 530)
    union all
    select PLACE as 地點, SERIAL_NUMBER as 序號, HYDROGEN as 氫離子濃度指數, TURBIDITY as 濁度, CHLORINE as 自由有效餘氯,
    WATER_SYSTEM as 供水系統
    from TAOYUAN_WATER
    where CHLORINE in (0.77, 0.76, 0.66) and HYDROGEN != 7.8 and TURBIDITY = 0.6
) data
left join TAOYUAN_REGIONS
on TAOYUAN_REGIONS.SERIAL = data.地點
left join TAOYUAN_WATER_SYSTEM
on TAOYUAN_WATER_SYSTEM.SERIAL = data.供水系統
order by 濁度 asc, 序號 desc;
```



函數

- MAX & MIN
- COUNT
- SUM & AVG
- GROUP BY
- HAVING

MAX() & MIN()

- 來取得一個欄位的**最大值** or **最小值**。

```
SELECT MAX(column_name)
FROM table_name
WHERE condition;
```

```
SELECT MIN(column_name)
FROM table_name
WHERE condition;
```

【範例】

取得系統開發部最大行編

```
select max(EMP_ID) from SQL_EMP
where DEP = '系統開發部';
```

max(EMP_ID)

00054321

若無資料，則為 null

COUNT()

- 計算資料數量（不在意資料內容）。

```
SELECT COUNT(column_name)
FROM table_name
WHERE condition;
```

【範例】

取得系統開發部人數

```
select count(1) as CNT
from SQL_EMP
where DEP = '系統開發部';
```

EMP_ID	DEP
00012345	系統開發部
00055333	開發系統部
00012468	系統開發部
00054321	開發系統部

若無符合資料，則為 0

CNT
2

有理專服務的客戶數

```
select count(EMP) as CNT
from SQL_CUSTOMER;
或
select count(1) as CNT
from SQL_CUSTOMER
where EMP is not null;
```

count 欄位值為 null，不計入
(不同資料庫，結果可能不同)

ID	EMP
C300300300	0012345
D400400400	0012345
A100100100	

CNT
2

SUM() & AVG()

- 計算一個欄位的**總合** or **平均**。
 - 僅能使用在數值欄位

```
SELECT SUM(column_name)
FROM table_name
WHERE condition;
```

```
SELECT AVG(column_name)
FROM table_name
WHERE condition;
```

【範例】

取得 2021-01-01 那天的平均訂單金額

```
select avg(AMOUNT) as AVG
from SQL_ORDER
where ORDER_DT = '2021-01-01';
```

ORDER_NO	AMOUNT	ORDER_DT
000001	399	2021-01-01
000002	501	2021-01-01
000003	183	2021-01-02
000004	50	2021-01-02

AVG

450

GROUP BY

- 用來將查詢結果中特定欄位值相同的資料分為若干個群組，而每一個群組都會傳回一個資料列。

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
ORDER BY column_name(s);
```

【範例】

各部門人數

```
select DEP, count(EMP_ID) as CNT
from SQL_EMP
group by DEP;
```

select 欄位只能是：

1. group by 欄位
2. 其他欄位搭配函數

EMP_ID	EMP_NM	ID	TEL	DEP
00012345	王○明	A100100100	1000#1010	系統開發部
00055333	王○茗	C300300300	1000#1030	開發系統部
00012468	王○名	A200200200	1000#1012	系統開發部
00054321	孫○拉	B200200200	1000#1020	開發系統部

DEP	CNT
系統開發部	2
開發系統部	2

HAVING

- 對函數產生的值來設定條件

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
HAVING condition
ORDER BY column_name(s);
```

【範例】

服務超過 1 位客戶的理專排名

```
select EMP, count(ID) as CNT
from SQL_CUSTOMER
where EMP is not null
group by EMP
having count(ID) > 1
order by CNT desc, EMP;
```

ID	NAME	EMP
C300300300	陳○圓	0054321
D400400400	郭○多	0012345
A100100100	王○明	
C300300333	李○圓	
D400400444	張○多	0012345
A100100111	蔡○明	0054321

EMP	CNT
0012345	2
0054321	2

100

- 請利用函數寫出以下需求SQL語法
- 問題 1：在Taoyuan_water中，地區代號為T41、T44、T16的資訊
地區總筆數、最大自由有效餘氯、最小自由有效餘氯、氯離子濃度指數總和、濁度平均

[illegible]

- **問題 2：**在Taoyuan_water中，所有地區中，濁度平均大於 0.7 的地區，並正確顯示行政區

提示：應該要 GROUP BY 哪一個欄位？HAVING 後面接的是什麼？是欄位嗎？

	行政地區	地區總筆數	濁度平均
1	新屋區	32	0.734375
2	觀音區	32	0.79375
3	中壢區	63	0.719047619047619047619047619047619047619047619

課堂練習⑨

- 問題 1

select PLACE, count(CHLORINE) as 地區總筆數, max(CHLORINE) as 最大自由有效餘氯, min(CHLORINE) as 最小自由有效餘氯, sum(HYDROGEN) as 氫離子濃度指數總和, avg(TURBIDITY) as 濁度平均

from TAOYUAN_WATER

where PLACE in('T40','T44','T16')

group by PLACE

- 問題 2

select TR.TOWNSHIP as 行政地區, COUNT(CHLORINE) as 地區總筆數, avg(TURBIDITY) as 濁度平均

from TAOYUAN_WATER TW

left join TAOYUAN_REGIONS TR on TW.PLACE = TR.SERIAL

group by TR.TOWNSHIP

having avg(TURBIDITY) > 0.7 ;

補充：想要格式化輸出的字串？

to_char(avg(TURBIDITY),'0.9') as 濁度平均

其他查詢建議

- 盡量避免大量的排序操作，如ORDER BY、GROUP BY、HAVING等等，會讓資料庫做額外的計算，增加處理時間。
- 如果沒有需要剔除重複資料的需求，UNION ALL效能會比UNION好，因為UNION會加入類似DISTINCT的演算法。
- 盡可能在資料來源層就先過濾資料。

Data Manipulation Language , DML

資料操控

INSERT

- 用於**新增**數據：

① 自定義欄位順序寫入

```
INSERT INTO table_name (column1, column2, column3, ...)
VALUES (value1, value2, value3, ...);
```

② 依表格欄位順序寫入

```
INSERT INTO table_name
VALUES (value1, value2, value3, ...);
```

※ 未設定欄位則為 null

【範例】

insert into SQL_EMP (EMP_NM, DEP, EMP_ID)
values ('王〇明', '系統開發部', '00012345');

EMP_ID	EMP_NM	ID	TEL	DEP
00012345	王〇明	null	null	系統開發部

insert into SQL_EMP
values ('王〇明', '系統開發部', '00012345');

EMP_ID	EMP_NM	ID	TEL	DEP
王〇明	系統開發部	00012345	null	null

INSERT

- 從其他表格新增數據，搭配子查詢：

```
INSERT INTO table2  
SELECT * FROM table1  
WHERE condition;
```

```
INSERT INTO table2 (column1, column2, column3, ...)  
SELECT column1, column2, column3, ...  
FROM table1  
WHERE condition;
```

※ 注意欄位型態與數量是否對應

【範例】

insert into SQL_EMP_HIS

select EMP_ID, EMP_NM, ID, null, '資訊板塊' from SQL_EMP;

EMP_ID	EMP_NM	ID	TEL	DEP
00012345	王○明	A100100100	null	資訊板塊
00054321	孫○拉	B200200200	null	資訊板塊

UPDATE

- 用於更新數據：

```
UPDATE table_name  
SET column1 = value1, column2 = value2, ...  
WHERE condition;
```

※ 未設定欄位則不異動，保持原資料

也可搭配子查詢，從其他表格更新數據。
※ 搜尋關鍵字：update set from

【範例】

更新所有資料的單位為資訊板塊

update SQL_EMP
set DEP = '資訊板塊'

EMP_ID	EMP_NM	ID	TEL	DEP
00012345	王○明	A100100100	1000#1010	資訊版塊
00054321	孫○拉	B200200200	1000#1020	資訊版塊

DELETE

- 用於刪除數據：

```
DELETE FROM table_name WHERE condition;
```

※ 未設定 where 條件就整張表格清空

【範例】

delete from SQL_EMP
where DEP = '開發系統部';

EMP_ID	EMP_NM	ID	TEL	DEP
00012345	王○明	A100100100	1000#1010	系統開發部
00054321	孫○拉	B200200200	1000#1020	開發系統部



課堂練習⑩

- 呈練習1建立的資料表 EMP_學號，進行以下練習：

問題 1：新增一筆資料

EMP_ID	EMP_NM	ID	TEL	DEP
E + 日期 + 自己 學號	自己姓名	Z999999999	1000#1010	XXXX部

問題 2：更新資料

EMP_ID	EMP_NM	ID	TEL	DEP
E + 日期 + 自己 學號	自己姓名	P000000000	9999#9999	XXXX部

問題 3：刪除資料問題 1 資料

Data Normalization

資料庫正規化

ER模型 (Entity -Relation Model) 實體關係模式

■ 用來描述「實體」與「實體」之間關係的工具。。

■ 專有名詞：

- 實體：是指用以描述真實世界的物件

實體

- 屬性：用來描述實體的性質

屬性

- 鍵值：用來辨認某一實體集中的每一個實體的唯一值

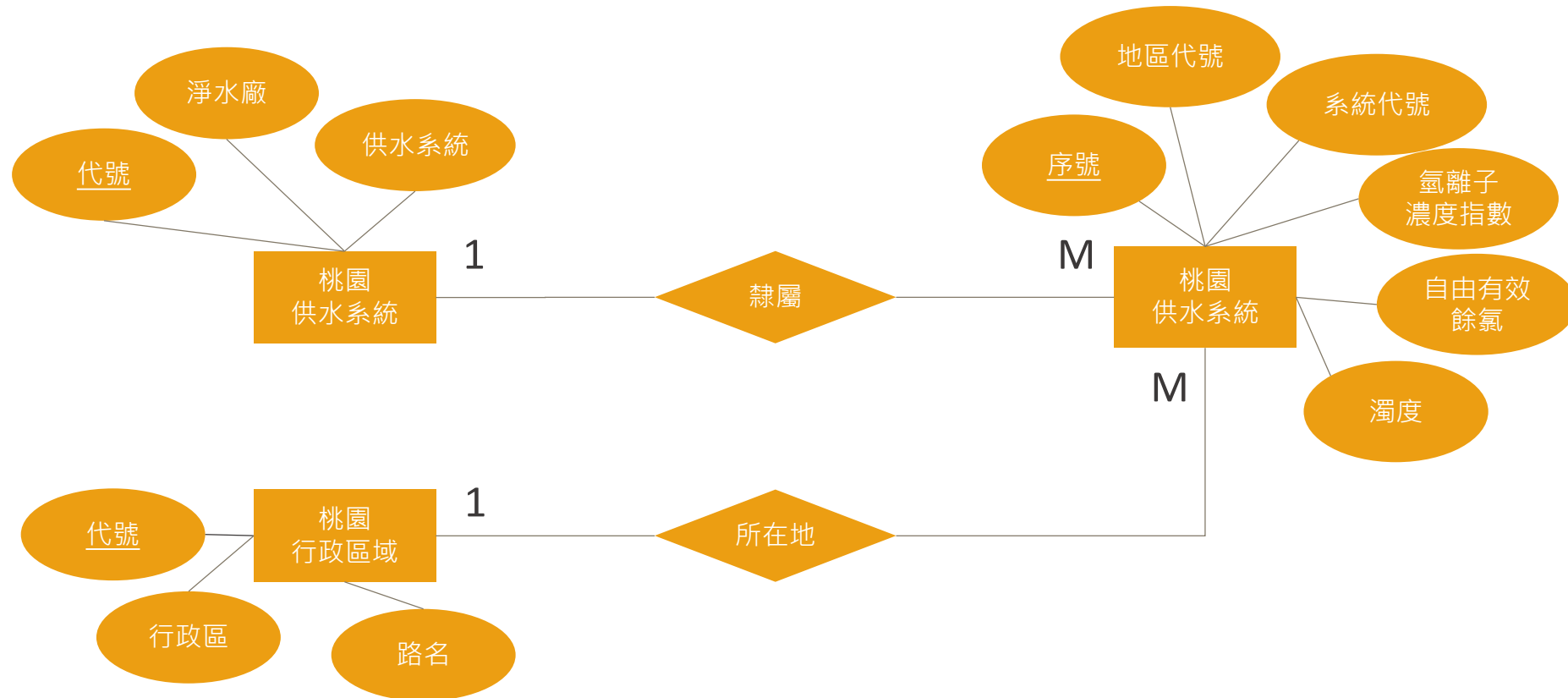
鍵值

- 關係：用來表示「一個實體」與「另一個實體」關聯的方式 (1:1 、 1:M 、 M:N)

關係

ER模型 (Entity -Relation Model) 實體關係模式

■ 利用之前SQL練習題的架構畫出 ER model 。

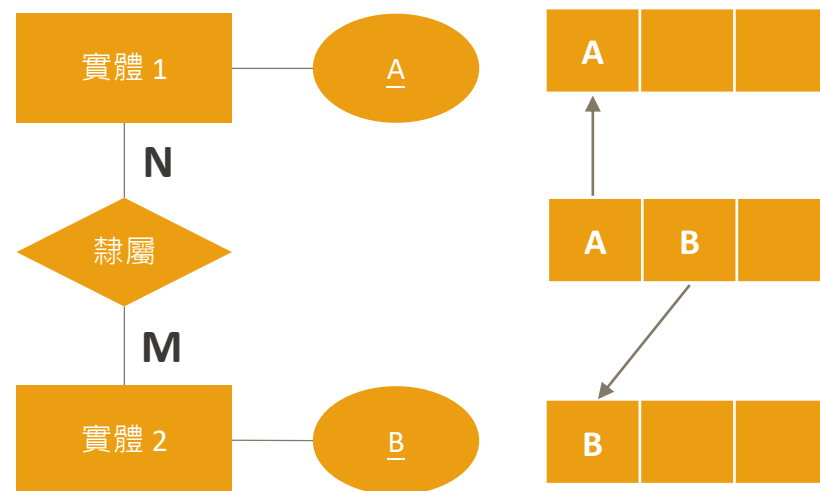


ER模型 (Entity -Relation Model) 實體關係模式

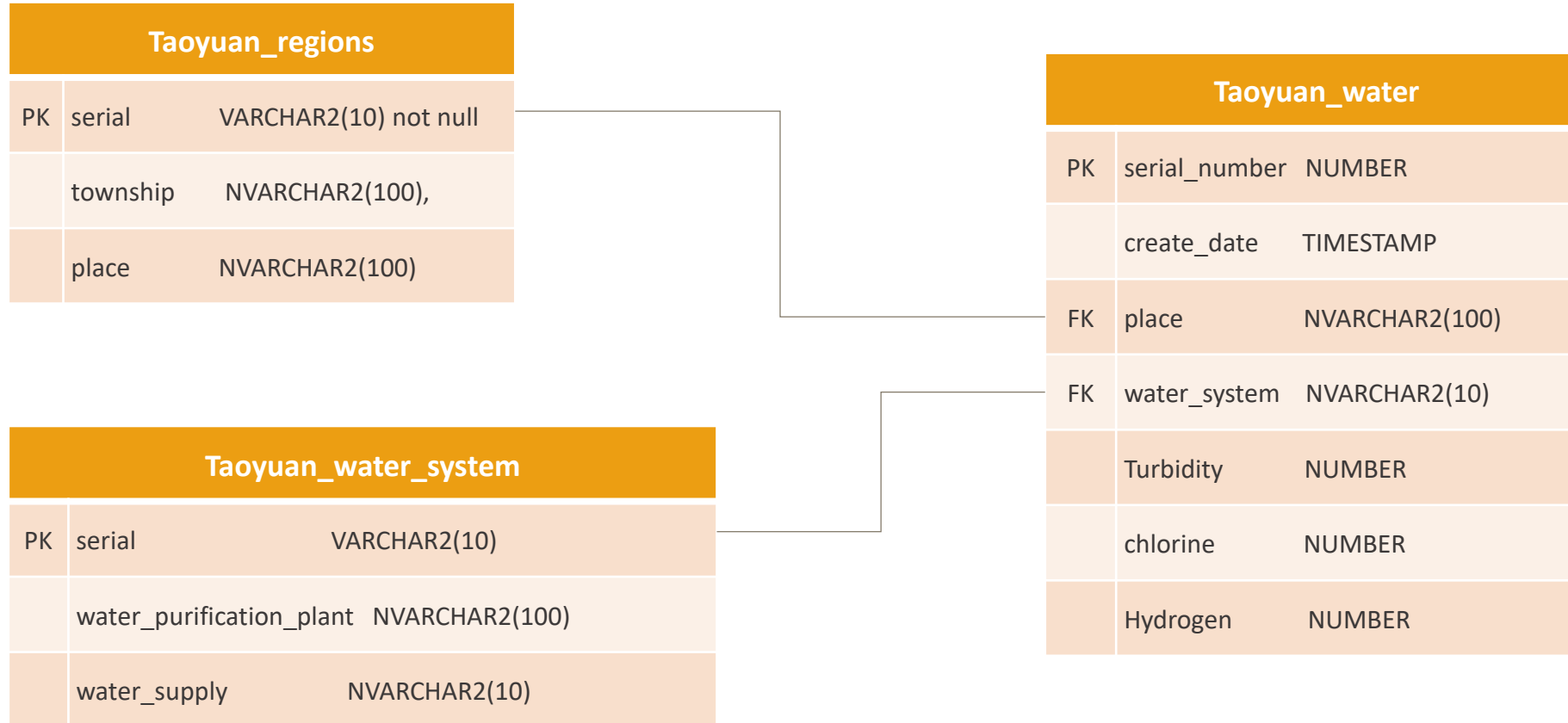
問題 1：為什麼上一頁的 ER model 只有 1:M，那 1:1 和 M:N 呢？

- 在設計表格的時候，會希望能有效的使用表格，若為 1:1 的情況，便可以整合在同一個表格內
- 若為M:N的表格，我們可以直接把拆成兩個 1:M 的關係

問題 2：所以我們在設計資料庫的時候只要畫出 ER model 就行了嗎？



ER模型 (Entity -Relation Model) 實體關係模式



什麼是資料庫正規化?

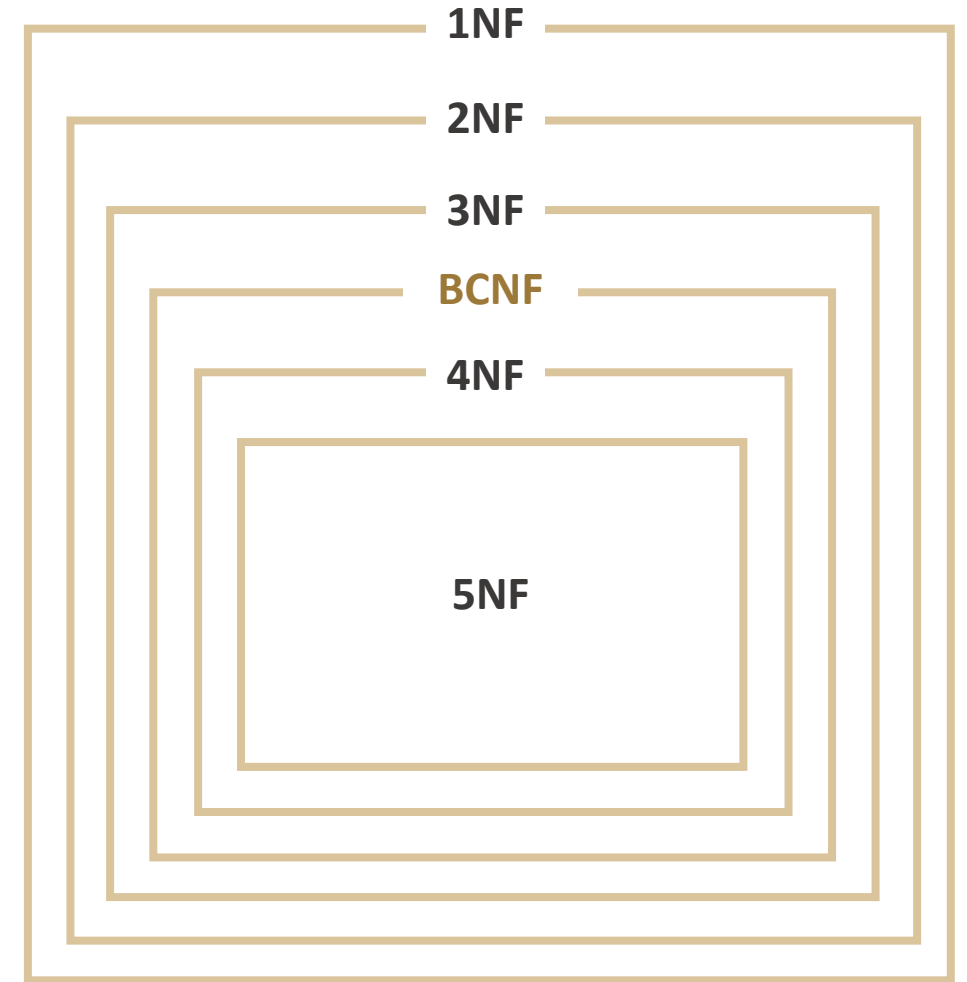
- 藉由刪除重複資料與單一化每張表格紀錄的資訊，減少資料儲存空間與資料內容不一致的過程。
- 經過正規化後的資料庫，應具備以下特性：
 - 欄位唯一性：每個欄位只儲存一項資料
 - 主關鍵欄位：每筆資料都擁有一個主鍵，來區別這些資料
 - 功能關聯性：欄位之間的關聯應該要明確
 - 欄位獨立性：欄位之間不應存在遞移相依

正規化的規則

- 資料庫在正規化時會有一些規則，並且每條規則都稱為「正規形式」。如果符合第一條規則，則資料庫就稱為「第一正規化形式 (1NF)」；如果符合前二條規則，則資料庫就被視為屬於「第二正規化形式 (2NF)」。
- 雖然資料庫的正規化最多可以進行到**第五正規化**形式，但是在實務上，**BCNF**被視為大部分應用程式所需的最高階正規形式。

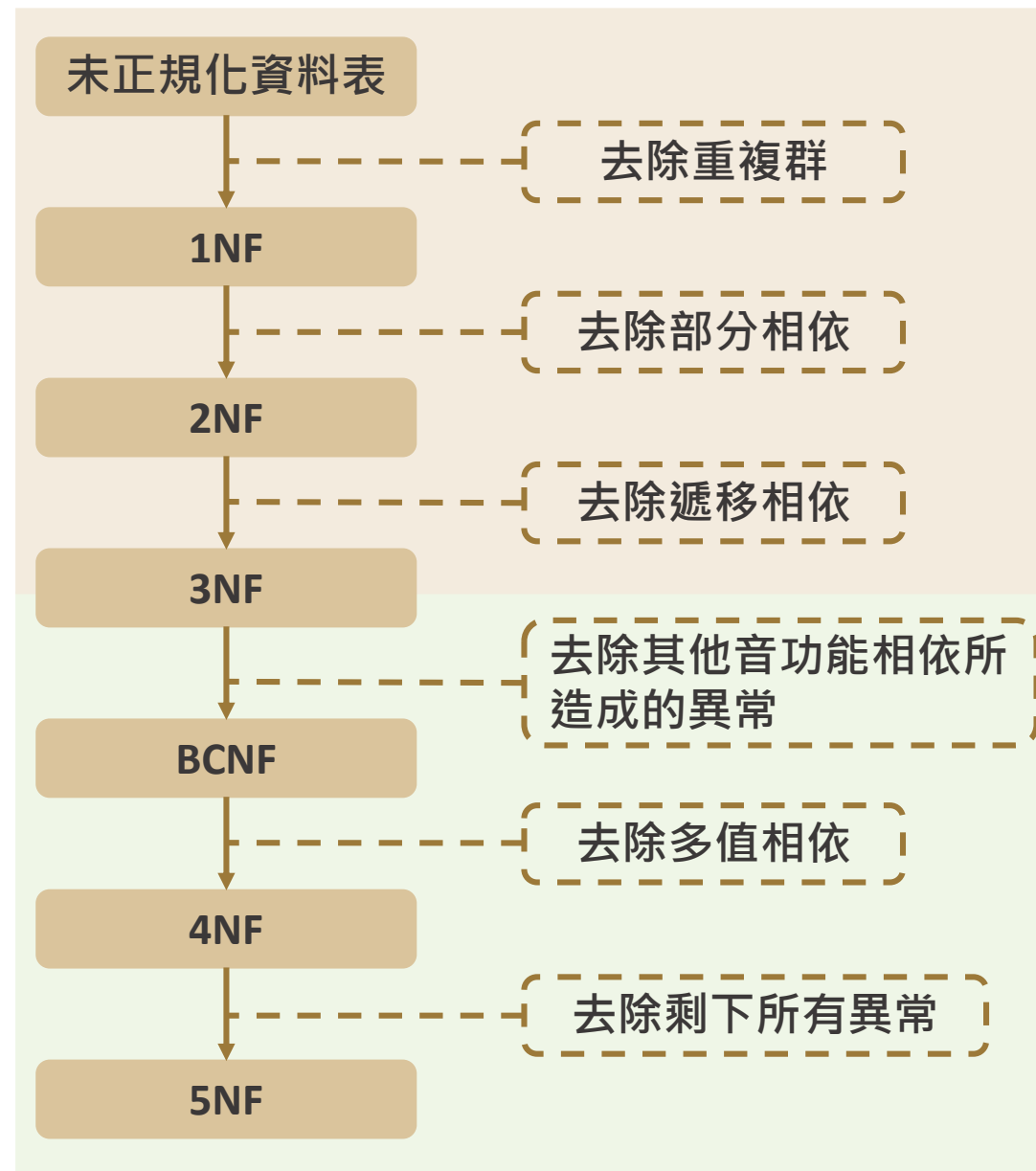
What is BCNF?

- Boyce-Codd正規形式 (Boyce-Codd normal form, BCNF) ，是資料庫規格化的一種正規形式。是在第三正規化的基礎上加上稍微更嚴格的約束，一般資料庫正規化到達此階段即可。



正規化的步驟

- 在資料表正規化的過程 (1NF 到 BCNF) 中, 每一個階段都是以欄位的「相依性」, 做為分割資料表的依據之一。



第一正規化

■ 1NF要完成的工作：

- 一個欄位只能有單值
- 消除意義上重複的欄位
- 決定主鍵

姓名	性別	項目	價格	數量	日期
小明	男	培根三明治, 奶茶	35, 15	1, 1	11/03
阿花	女	玉米蛋餅, 奶茶	25, 15	1, 1	11/05
大助	男	培根三明治, 鐵板麵, 紅茶	35, 50, 15	1, 1, 2	11/03



訂單編號	姓名	性別	項目	價格	數量	總金額	日期
O0001	小明	男	培根三明治	35	1	35	11/03
O0001	小明	男	奶茶	15	1	15	11/03
O0002	阿花	女	玉米蛋餅	25	1	25	11/05
O0002	阿花	女	奶茶	15	1	15	11/05
O0003	大助	男	培根三明治	35	1	35	11/03
O0003	大助	男	鐵板麵	50	1	50	11/03
O0003	大助	男	紅茶	15	2	30	11/03

仍有重複資訊



第二正規化

■ 2NF要完成的工作：

- 消除部分相依：跟主鍵只有一部份有關係，另一部份沒有關係的欄位，把這些欄位獨立出另一張表。

編號	姓名	性別
U1	小明	男
U2	阿花	女
U3	大助	男

編號	項目	價格
F1	培根三明治	35
F2	奶茶	15
F3	玉米蛋餅	25
F4	鐵板麵	50
F5	紅茶	15

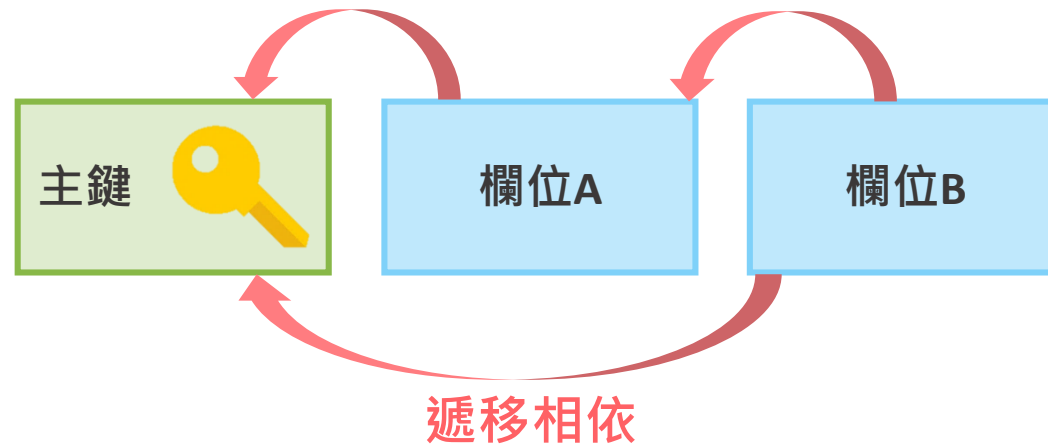
訂單編號	客戶編號	項目編號	數量	總金額	日期
O0001	U1	F1	1	35	11/03
O0001	U1	F2	1	15	11/03
O0002	U2	F3	1	25	11/05
O0002	U2	F2	1	15	11/05
O0003	U3	F1	1	35	11/03
O0003	U3	F4	1	50	11/03
O0003	U3	F5	2	30	11/03

第三正規化

■ 3NF要完成的工作：

- 消除資料表中與主欄位的遞移相依。
- 遞移相依：非主鍵屬性的欄位都只能和候選鍵相關，非主鍵屬性的欄位彼此間應該要是獨立無關的。

例：欄位A跟主鍵相關，欄位B只跟欄位A相關，那麼欄位B跟主鍵就是遞移相依。



第三正規化

■ 消除資料表中遞移相依：移除總金額欄位

編號	姓名	性別
U1	小明	男
U2	阿花	女
U3	大助	男

編號	項目	價格
F1	培根三明治	35
F2	奶茶	15
F3	玉米蛋餅	25
F4	鐵板麵	50
F5	紅茶	15

訂單編號	客戶編號	項目編號	數量	日期
O0001	U1	F1	1	11/03
O0001	U1	F2	1	11/03
O0002	U2	F3	1	11/05
O0002	U2	F2	1	11/05
O0003	U3	F1	1	11/03
O0003	U3	F4	1	11/03
O0003	U3	F5	2	11/03

- 然而並非每次都一定要依照正規化設計表格才是最好的，3NF後要查客戶買了什麼產品反而需要關聯較多張表格

課堂練習 正規化

- 請利用下面的表格進行正規化

單號	小類	大類	標題	表單狀態	流程	進行流程	流程處理人	建立人員
269011	金融	金融及保險業	辦理全行安控○○○	待結案	1,2,3	3	令狐冲,任盈盈,東方不敗	儀琳
269055	保險	金融及保險業	請協助辦理○○○○	待承辦	1,2,3	2	左冷禪,任我行,岳不羣	藍鳳凰
269801	高雄	不動產業	申請○○○	待承辦	1,2,3,4	2	岳不羣,左冷禪,任盈盈,東方不敗	令狐冲
569011	技術	科學及技術服務業	為處理○○○○請協助辦理	待會辦	1,2,3,4	3	藍鳳凰,東方不敗,任盈盈,岳不羣	左冷禪

課堂練習 第一正規化

單號	小類	大類	標題	表單狀態	流程	進行流程	流程處理人	建立人員
269011	金融	金融及保險業	辦理全行安控○○○	待結案	1	3	令狐冲	儀琳
269011	金融	金融及保險業	辦理全行安控○○○	待結案	2	3	任盈盈	儀琳
269011	金融	金融及保險業	辦理全行安控○○○	待結案	3	3	東方不敗	儀琳
269055	保險	金融及保險業	請協助辦理○○○○	待承辦	1	2	左冷禪	藍鳳凰
269055	保險	金融及保險業	請協助辦理○○○○	待承辦	2	2	任我行	藍鳳凰
269055	保險	金融及保險業	請協助辦理○○○○	待承辦	3	2	岳不羣	藍鳳凰
269801	高雄	不動產業	申請○○○	待承辦	1	2	岳不羣	令狐冲
269801	高雄	不動產業	申請○○○	待承辦	2	2	左冷禪	令狐冲
269801	高雄	不動產業	申請○○○	待承辦	3	2	任盈盈	令狐冲
269801	高雄	不動產業	申請○○○	待承辦	4	2	東方不敗	令狐冲
569011	技術	科學及技術服務業	為處理○○○○請協助辦理	待會辦	1	3	藍鳳凰	左冷禪
569011	技術	科學及技術服務業	為處理○○○○請協助辦理	待會辦	2	3	東方不敗	左冷禪
569011	技術	科學及技術服務業	為處理○○○○請協助辦理	待會辦	3	3	任盈盈	左冷禪
569011	技術	科學及技術服務業	為處理○○○○請協助辦理	待會辦	4	3	岳不羣	左冷禪



課堂練習 第二正規化-1

問題 1：表格可以歸類出哪些大項？

單號	標題	表單狀態	建立人員	流程	進行流程	流程處理人	小類	大類
269011	辦理全行安控○○○	待結案	儀琳	1	3	令狐冲	金融	金融及保險業
269011	辦理全行安控○○○	待結案	儀琳	2	3	任盈盈	金融	金融及保險業
269011	辦理全行安控○○○	待結案	儀琳	3	3	東方不敗	金融	金融及保險業
269055	請協助辦理○○○○	待承辦	藍鳳凰	1	2	左冷禪	保險	金融及保險業
269055	請協助辦理○○○○	待承辦	藍鳳凰	2	2	任我行	保險	金融及保險業
269055	請協助辦理○○○○	待承辦	藍鳳凰	3	2	岳不羣	保險	金融及保險業
269801	申請○○○	待承辦	令狐冲	1	2	岳不羣	高雄	不動產業
269801	申請○○○	待承辦	令狐冲	2	2	左冷禪	高雄	不動產業
269801	申請○○○	待承辦	令狐冲	3	2	任盈盈	高雄	不動產業
269801	申請○○○	待承辦	令狐冲	4	2	東方不敗	高雄	不動產業
569011	為處理○○○○請協助辦理	待會辦	左冷禪	1	3	藍鳳凰	技術	科學及技術服務業

課堂練習 第二正規化-1

問題 2：哪些項目的層級有可能是比較高的

單號	標題	表單狀態	建立人員	流程	進行流程	流程處理人	小類	大類
269011	辦理全行安控○○○	待結案	儀琳	1	3	令狐冲	金融	金融及保險業
269011	辦理全行安控○○○	待結案	儀琳	2	3	任盈盈	金融	金融及保險業
269011	辦理全行安控○○○	待結案	儀琳	3	3	東方不敗	金融	金融及保險業
269055	請協助辦理○○○○	待承辦	藍鳳凰	1	2	左冷禪	保險	金融及保險業
269055	請協助辦理○○○○	待承辦	藍鳳凰	2	2	任我行	保險	金融及保險業
269055	請協助辦理○○○○	待承辦	藍鳳凰	3	2	岳不羣	保險	金融及保險業
269801	申請○○○	待承辦	令狐冲	1	2	岳不羣	高雄	不動產業
269801	申請○○○	待承辦	令狐冲	2	2	左冷禪	高雄	不動產業
269801	申請○○○	待承辦	令狐冲	3	2	任盈盈	高雄	不動產業
269801	申請○○○	待承辦	令狐冲	4	2	東方不敗	高雄	不動產業
569011	為處理○○○○請協助辦理	待會辦	左冷禪	1	3	藍鳳凰	技術	科學及技術服務業

課堂練習 第二正規化-1

我們先來建立層級較高的表格

員工編號(PK)	員工姓名
E001	儀琳
E002	藍鳳凰
E003	莫大
E004	丹青生
E005	令狐冲
E006	任盈盈
E007	東方不敗
E008	左冷禪
E009	任我行
E010	岳不羣

小類	大類
金融	金融及保險業
保險	金融及保險業
高雄	不動產業
技術	科學及技術服務業

PK	狀態
ST01	待結案
ST02	待承辦
ST03	待會辦

PK	大類
M01	金融及保險業
M02	不動產業
M03	科學及技術服務業

PK	大類	小類
S01	M01	金融
S02	M01	保險
S03	M02	高雄
S04	M03	技術

課堂練習 第二正規化-1

問題 3：下一個層級的表格是.....？

單號	標題	表單狀態	建立人員	流程	進行流程	流程處理人	小類
269011	辦理全行安控○○○	ST01	E001	1	3	E005	S01
269011	辦理全行安控○○○	ST01	E001	2	3	E006	S01
269011	辦理全行安控○○○	ST01	E001	3	3	E007	S01
269055	請協助辦理○○○○	ST02	E002	1	2	E008	S02
269055	請協助辦理○○○○	ST02	E002	2	2	E009	S02
269055	請協助辦理○○○○	ST02	E002	3	2	E010	S02
269801	申請○○○	ST02	E005	1	2	E010	S03
269801	申請○○○	ST02	E005	2	2	E008	S03
269801	申請○○○	ST02	E005	3	2	E006	S03
269801	申請○○○	ST02	E005	4	2	E007	S03
569011	為處理○○○○請協助辦理	ST03	E008	1	3	E002	S04



課堂練習 第二正規化-1

流程必須憑藉表單而存在，所以先處理表單資訊，再處理流程

單號	標題	表單狀態	建立人員	流程	進行流程	流程處理人	小類
269011	辦理全行安控○○○	ST01	E001	1	3	E005	S01
269011	辦理全行安控○○○	ST01	E001	2	3	E006	S01
269011	辦理全行安控○○○	ST01	E001	3	3	E007	S01
269055	請協助辦理○○○○	ST02	E002	1	2	E008	S02
269055	請協助辦理○○○○	ST02	E002	2	2	E009	S02
269055	請協助辦理○○○○	ST02	E002	3	2	E010	S02
269801	申請○○○	ST02	E005	1	2	E010	S03
269801	申請○○○	ST02	E005	2	2	E008	S03
269801	申請○○○	ST02	E005	3	2	E006	S03
269801	申請○○○	ST02	E005	4	2	E007	S03
569011	為處理○○○○請協助辦理	ST03	E008	1	3	E002	S04



課堂練習 第二正規化-1

剔除表單表格內重複資訊，可以得到已經整理完的表單表格

單號	標題	表單狀態	建立人員	小類
269011	辦理全行安控○○○	ST01	E001	S01
269055	請協助辦理○○○○	ST02	E002	S02
269801	申請○○○	ST02	E005	S03
569011	為處理○○○○請協助辦理	ST03	E008	S04

單號	流程	進行流程	流程處理人
269011	1	3	E005
269011	2	3	E006
269011	3	3	E007
269055	1	2	E008
269055	2	2	E009
269055	3	2	E010
269801	1	2	E010
269801	2	2	E008
269801	3	2	E006
269801	4	2	E007
569011	1	3	E002



課堂練習 第二正規化-1

最後要來處理流程表格

注意：

1. 流程必須憑藉表單而存在，所以必須保留表單單號
2. 進行流程代表目前流程的狀態為：進行中，其他流程的狀態可能是已完成或未完成，故可將每一個流程設定不同狀態
3. 流程狀態也可建立一個表格管理
4. PK有兩種做法：
 - 統一流水號
 - 單號 + 流程 組成複合主鍵

PK	流程狀態
F01	進行中
F02	已完成
F03	未完成

單號	流程	進行流程	流程處理人
269011	1	已完成	E005
269011	2	已完成	E006
269011	3	進行中	E007
269055	1	已完成	E008
269055	2	進行中	E009
269055	3	未完成	E010
269801	1	已完成	E010
269801	2	進行中	E008
269801	3	未完成	E006
269801	4	未完成	E007
569011	1	已完成	E002



課堂練習 第二正規化-1

PK	員工姓名
E001	儀琳
E002	藍鳳凰
E003	莫大
E004	丹青生
E005	令狐冲
E006	任盈盈
E007	東方不敗
E008	左冷禪
E009	任我行
E010	岳不羣

PK	大類
M01	金融及保險業
M02	不動產業
M03	科學及技術服務業

PK	大類	小類
S01	M01	金融
S02	M01	保險
S03	M02	高雄
S04	M03	技術

單號	標題	表單狀態	建立人員	小類
269011	辦理全行安控○○○	ST01	E001	S01
269055	請協助辦理○○○○	ST02	E002	S02
269801	申請○○○	ST02	E005	S03
569011	為處理○○○○請協助辦理	ST03	E008	S04

PK	狀態
ST01	待結案
ST02	待承辦
ST03	待會辦

PK	流程狀態
F01	進行中
F02	已完成
F03	未完成

單號	流程	進行流程	流程處理人
269011	1	F02	E005
269011	2	F02	E006
269011	3	F01	E007
269055	1	F02	E008
269055	2	F01	E009
269055	3	F03	E010
269801	1	F02	E010
269801	2	F01	E008
269801	3	F03	E006
269801	4	F03	E007
569011	1	F02	E002

課堂練習 第三正規化-1

問題 1：在完成第二正規化後，還需要第三正規化嗎？

問題 2：如果需要的話，有哪邊是可以第三正規化的？

問題 3：如果不需要的話，我們在哪個環節避免第三正規化呢？

SQL規範

SQL 相關規範

- 目的：撰寫易讀且好維護的程式，亦可以避免一些寫法導致的問題，大多數規範會包含註解、命名、方法

1. SCHEMA、Table、欄位、別名大寫
2. SQL原生字小寫
3. Table 要加上 SCHEMA
4. SQL第二行以下須由空格開始
5. SQL 檔名 程式代碼_ + 方法名稱 + _流水號3碼

檔案命名

檔名：AGANTINFOQ029_getAgentList_001.sql

原生字小寫

SQL內容：

```
select PLACE as 地點 ,SERIAL_NUMBER  
from STUDENT.TAOYUAN_WATER  
left join STUDENT.TAOYUAN_REGIONS  
on TAOYUAN_REGIONS.SERIAL = TAOYUAN_WATER.PLACE  
where CHLORINE in : CHLORINE  
order by PLACE asc ;
```

Table SCHEMA

第二行開始
空格開頭

參考資源

- https://www.cc.ntu.edu.tw/chinese/epaper/0031/20141220_3109.html

The screenshot shows the header of the National Taiwan University (NTU) Computer and Information Network Center (C&INC) e-paper. The header includes the NTU logo, the center's name in Chinese, and the title '電子報' (E-paper). It also displays the issue number '第0031期', the date '2014.12.20發行', and the ISSN 'ISSN 2077-8813'. A navigation bar contains links for '歷史回顧', '訂閱/取消', '校務服務', '專題報導', '技術論壇', and '推薦刊物'. The main content area shows the breadcrumb '首頁 > 技術論壇' and the article title 'SQL指令優化SQL Tuning'. The author is listed as '作者：陳淑萍 / 臺灣大學計算機及資訊網路中心程式設計組程式設計師'. The article text begins with 'SQL指令敘述調校是最佳化SQL指令敘述的過程，以建立作佳化的SQL指令來更有效率地存取資料庫，包含SELECT, INSERT, DELETE和UPDATE操作指令。SQL指令敘述調校的目的則是在建立有效率的SQL查詢指令，其調校過程是從查詢指令的元素開始，特別針對WHERE條件子句。'.

國立臺灣大學
計算機及資訊網路中心 電子報

第0031期 · 2014.12.20發行
ISSN 2077-8813

歷史回顧 訂閱/取消 校務服務 專題報導 技術論壇 推薦刊物

首頁 > 技術論壇

技術論壇

SQL指令優化SQL Tuning

作者：陳淑萍 / 臺灣大學計算機及資訊網路中心程式設計組程式設計師

SQL指令敘述調校是最佳化SQL指令敘述的過程，以建立作佳化的SQL指令來更有效率地存取資料庫，包含SELECT, INSERT, DELETE和UPDATE操作指令。SQL指令敘述調校的目的則是在建立有效率的SQL查詢指令，其調校過程是從查詢指令的元素開始，特別針對WHERE條件子句。

SQL Tutorial

- SQL HOME
- SQL Intro
- SQL Syntax
- SQL Select**
- SQL Select Distinct
- SQL Where
- SQL And, Or, Not
- SQL Order By
- SQL Insert Into
- SQL Null Values
- SQL Update
- SQL Delete
- SQL Select Top
- SQL Min and Max
- SQL Count, Avg, Sum
- SQL Like
- SQL Wildcards
- SQL In
- SQL Between
- SQL Aliases
- SQL Joins
- SQL Inner Join
- SQL Left Join
- SQL Right Join
- SQL Full Join
- SQL Self Join
- SQL Union

SQL SELECT Statement

[< Previous](#) [Next >](#)

The SQL SELECT Statement

The **SELECT** statement is used to select data from a database.

The data returned is stored in a result table, called the result-set.

SELECT Syntax

```
SELECT column1, column2, ...  
FROM table_name;
```

Here, column1, column2, ... are the field names of the table you want to select data from. If you want to select all the fields available in the table, use the following syntax:

```
SELECT * FROM table_name;
```

COLOR PICKER

Thanks
