

結構化查詢語言-異動處理

SQL語言的概念

1.第一種為資料定義語言(Data Definition Language; DDL)

→用來「**定義**」**資料庫**的結構、欄位型態及長度。

2.第二種為資料操作語言(Data Manipulation Language; DML)

→用來「**操作**」**資料庫**的新增資料、修改資料、刪除資料、查詢資料等功能。

3.第三種為資料控制語言(Data Control Language; DCL)

→用來「**控制**」**使用者**對「**資料庫內容**」的存取權利。

因此，SQL語言透過**DDL**、**DML**及**DCL**來建立各種複雜的表格關聯，成為一個查詢資料庫的**標準語言**。

SQL 的 DDL 語言

【定義】

資料定義語言(Data Definition Language; **DDL**)，利用DDL，使用者可以**定義資料表**(**關聯網目**；**基底資料表**)和**設定完整性限制**。並且**DDL**允許資料庫使用者**建立**、**更改**或**刪除**(**資料表**(Table)、**索引**(Index)與**檢視**(View))資料庫物件。主要指令有三：**CREATE**、**ALTER** 與 **DROP**。如表所示。

DDL語言提供的三種指令表

Table	Index	View
(1)Create Table	(1)Create Index	(1)Create View
(2)Alter Table	(2)Alter Index	(2)Alter View
(3)Drop Table	(3)Drop Index	(3)Drop View

CREATE TABLE(建立資料表)

【定義】

Create Table命令是用來讓使用者**定義**一個**新的關聯(資料表)**，並設定關聯(表格)的**名稱、屬性及限制條件**。

【建立新資料表的步驟】

- (1)決定**資料表名稱與相關欄位**
- (2)決定欄位的**資料型態**
- (3)決定欄位的**限制 (指定值域)**
- (4)決定那些欄位可以**NULL(空值)**與**不可NULL**的欄位
- (5)找出必須**具有唯一值的欄位(主鍵)**
- (6)找出**主鍵-外來鍵**配對**(兩個表格)**
- (7)決定**預設值**(欄位值的初值設定)

【格式】

Create Table 資料表

(欄位{資料型態|定義域}[NULL|NOT NULL][預設值][定義整合限制]

⋮

Primary Key(欄位集合) ← 當主鍵

Unique(欄位集合) ← 當候選鍵

Foreign Key(欄位集合) References 基本表(屬性集合) ← 當外鍵

[ON Delete 選項] [ON Update 選項]

Check(檢查條件))

說明：

- (1) PRIMARY KEY 用來定義某一欄位為**主鍵**，不可為空值
- (2) UNIQUE 用來定義某一欄位**具有唯一的索引值**，可以有空值
- (3) NULL/NOT NULL 可以為空值/不可為空值
- (4) FOREIGN KEY 用來定義某一欄位為**外部鍵**
- (5) CHECK 用來**額外的檢查條件**

【舉例】

利用Create Table來建立「學生選課系統」的關聯式資料庫，其相關的資料表有三個，如下所示：

學生表(學號, 姓名, 電話, 地址)

選課表(學號, 課號, 成績)

課程表(課號, 課名, 學分數, 必選修)



建立「學生表、課程表」

建立「學生表」

```
CREATE TABLE 學生表  
(學號 CHAR(8) NOT NULL,  
姓名 CHAR(4),  
電話 CHAR(12),  
地址 CHAR(20),  
PRIMARY KEY(學號),  
UNIQUE(電話));
```

主鍵

建立「課程表」

```
CREATE TABLE 課程表  
(課號 CHAR(5) NOT NULL,  
課名 CHAR(20) NULL,  
學分數 INT NULL,  
必選修 CHAR(2) NULL,  
PRIMARY KEY(課號),  
);
```

主鍵

建立「選課表」

建立「選課表」

```
CREATE TABLE 選課表  
(學號 CHAR(8) NOT NULL,  
課號 CHAR(5) NOT NULL,  
成績 INT,  
PRIMARY KEY(學號, 課號));
```

複合主鍵

ALTER TABLE(修改資料表)

【定義】

ALTER TABLE命令是用來對已存在的資料表增加欄位、修改欄位、刪除欄位，並且增加定義、修改定義或刪除定義等。

【格式】

```
ALTER TABLE 基本表  
[ALTER] COLUMN [欄位] [資料型態] [NULL|NOTNULL]  
[RESTRICT|CASCADE]  
[ADD|DROP][限制 | 屬性]  
[ADD][欄位] {資料型態 | 定義域} [NULL|NOTNULL]  
[預設值] [定義整合限制]
```

【符號說明】

- { | } 代表在大括號內的項目是**必要項**，但可以擇一。
- [] 代表在中括號內的項目是**非必要項**，依實際情況來選擇。

【分析1】「新增」欄位定義

題目：原來的學生資料表，再增加一個 e-mail 欄位

舊： 學生資料表 (學號, 姓名, 電話, 地址)



ALTER TABLE 學生資料表 **ADD** e-mail CHAR(50)



新： 學生資料表(學號, 姓名, 電話, 地址, e-mail)

【分析2】「修改」欄位定義

原來的學生資料表中的「地址」之資料型態(大小20→50)。

```
ALTER TABLE 學生資料表 ALTER 地址 CHAR(50);
```

【分析3】「刪除」欄位定義

將學生資料表中的「地址」欄位刪除。

```
ALTER TABLE 學生資料表 DROP 地址;
```

DROP TABLE

【定義】

是指用來刪除資料表定義，當然，如果一個資料表內還有剩餘的紀錄，則這些紀錄會一併被刪除，因為如果資料表定義被刪除，則資料表的紀錄就沒有存在的意義了。

【格式】

```
DROP TABLE 資料表名稱
```

【分析】DROP TABLE 學生表;

SQL 的 DML 指令介紹

資料操作語言(Data Manipulation Language; DML)，利用DML，使用者可以從事對資料表記錄的新增、修改、刪除及查詢等功能。

DML有四種基本指令：

1. INSERT(新增)
2. UPDATE(修改)
3. DELETE(刪除)
4. SELECT(查詢)

INSERT指令

【定義】指新增一筆記錄到新的資料表內。

【格式】

```
INSERT INTO 資料表名稱 <欄位串列>  
VALUES(<欄位值串列> | <SELECT指令>)
```

【實例1】未指定欄位串列的新增(但是欲新增資料值必須能夠配合欄位型態及個數)

假設現在新增「產品代號」為A005,「品名」為桌球衣及「單價」為1200的記錄到產品資料表中。

SQL 指令			
INSERT INTO 產品資料表			
VALUES('A005','桌球衣','1200')			
產品資料表			
	產品代號	品名	單價
#1	C001	羽球拍	3000
#2	B004	桌球鞋	2300
→#3	A005	桌球衣	1200

[註]在VALUES子句內的屬性值串列的排列順序要跟當初定義時一致。

【實例2】指定欄位串列的新增

欲新增資料值個數可以自行指定，不一定要與定義的欄位個數相同

假設現在新增產品代號為D002，品名為桌球皮的記錄到產品資料表

SQL 指令			
INSERT INTO 產品資料表(產品代號 , 品名)			
VALUES('D002','桌球皮')			

產品資料表

	產品代號	品名	單價
#1	C001	羽球拍	3000
#2	B004	桌球鞋	2300
#3	A005	桌球衣	1200
#4	D002	桌球皮	NULL

註：未指定對映的屬性，會被設定為DEFAULT值或NULL值，

如「產生資料表」的單價屬性值為NULL。

【實例3】

將羽球相關的品名從「產品資料表」中整批新增到另一個資料表中。

SQL 指令

```
INSERT INTO 羽球產品資料表  
SELECT *  
FROM 產品資料表  
WHERE 產品資料表.品名 LIKE '羽*'
```

羽球產品資料表

	產品代號	品名	單價
#1	C021	羽球衣	1200
#2	C032	羽球鞋	3200
#3	C001	羽球拍	3000

UPDATE指令

【定義】指修改一個資料表中某些值組(記錄)之屬性值。

【格式】

```
UPDATE 資料表名稱
```

```
SET {<欄位名稱1>=<欄位值1>, ..., <欄位名稱n>=<欄位值n>}
```

```
[WHERE <條件子句>]
```

[實例]

在「產品資料表」中，有關桌球相關產品單價調升30%。

【解答】

SQL指令

```
UPDATE 產品資料表  
SET 單價 = 單價*1.3  
WHERE 品名 LIKE '桌球%'
```

產品資料表

	產品代號	品名	單價
#1	C001	羽球拍	3000
#2	B004	桌球鞋	2300 →2990
#3	A005	桌球衣	1200 →1560
#4	D002	桌球皮	550 →715

DELETE指令

【定義】把合乎條件的值組(記錄)，從資料表中刪除。

【格式】

```
DELETE FROM 資料表名稱  
[WHERE <條件式>]
```

[實例1]

將尚未決定單價的產品記錄刪除。

SQL 指令

```
DELETE FROM 產品資料表  
WHERE 單價 IS NULL
```

產品資料表

	產品代號	品名	單價
#1	C001	羽球拍	3000
#2	B004	桌球鞋	2300
#3	A005	桌球衣	1200
#4	D002	桌球皮	NULL



產品資料表

	產品代號	品名	單價
#1	C001	羽球拍	3000
#2	B004	桌球鞋	2300
#3	A005	桌球衣	1200

[實例2]

將在「產品資料表」中刪除單價未達2500的產品記錄。

SQL指令

```
DELETE FROM 產品資料表  
WHERE產品代號 IN  
(SELECT產品代號  
FROM 產品資料表  
WHERE 單價 <=2500)
```

產品資料表

	產品代號	品名	單價
#1	C001	羽球拍	3000
#2	B004	桌球鞋	2300
#3	A005	桌球衣	1200



產品資料表

	產品代號	品名	單價
#1	C001	羽球拍	3000

DCL的指令介紹

【定義】

資料控制語言(Data Control Language; **DCL**)，**DCL控制**使用者對資料庫內容的**存取權利**。

【指令】

1.**GRANT**(授權)

2.**REVOKE**(移除權限)

【DCL提供交易控制指令】

(1)**COMMIT**指令：確認(儲存)資料庫的交易。

(2)**ROLLBACK**指令：回復(復原Recovery)資料庫的交易。

GRANT指令

【定義】

GRANT指令用來取得現有資料庫使用者帳號的權限。

【格式】

```
GRANT 權限 ON 資料表名稱  
TO 使用者
```

其中，「權限」可分為四種：Insert, Update, Delete, Select

[實例1]

對USER1與USER2提供SELECT與INSERT對客戶資料表的使用者權限功能。

【解答】

SQL語法

```
GRANT SELECT, INSERT ON 客戶資料表  
TO USER1, USER2
```


[實例2]

對**所有的使用者**提供Select的功能權限。

【解答】

SQL語法

```
GRANT SELECT ON 客戶資料表  
TO PUBLIC
```

REVOKE指令

【定義】 REVOKE指令用來**取消**資料庫**使用者已取得的權限**。

【格式】

```
REVOKE 權限 資料表名稱  
FROM 使用者
```

【實例】 表示從USER2帳號**移除**對的INSERT權限。

SQL語法

```
REVOKE INSERT ON 客戶資料表  
FROM USER2
```

COMMIT指令

【定義】

是指用來**確認最後一次COMMIT指令對資料庫所做的交易**。

【格式】

```
COMMIT [WORK];
```

【實例】 確認對資料庫所做的交易

SQL語法

```
DELETE FROM 產品資料表  
WHERE 產品代號='D002'  
COMMIT; ←保證永久DELETE
```

ROLLBACK指令

【定義】

ROLLBACK指令用來回復(Recovery)尚未被COMMIT的資料庫交易。

```
ROLLBACK [TO SAVEPOINT]
```

[實例]

SELECT * FROM 產品資料表

產品資料表			
	產品代號	品名	單價
#1	C001	羽球拍	3000

**UPDATE 產品資料表
SET 單價=單價*1.5**



SELECT * FROM 產品資料表

產品資料表			
	產品代號	品名	單價
#1	C001	羽球拍	4500

ROLLBACK;



SELECT * FROM 產品資料表

產品資料表			
	產品代號	品名	單價
#1	C001	羽球拍	3000