

第六章 基本的SQL

- 目的
- SQL的關聯模式
- SQL的資料定義語言
 - ☞ CREATE TABLE
 - ☞ DROP TABLE
 - ☞ ALTER TABLE
- 基本的SQL查詢
 - ☞ SELECT...FROM...WHERE
- SQL的更新
 - ☞ INSERT
 - ☞ UPDATE
 - ☞ DELETE

©黃三益2007
資料庫的核心理論與實務第三版

6-1

目的

- 關聯模式定義嚴謹簡潔，但好用性和執行效率必須被考量
- IBM SYSTEM R (DB2)根據關聯模式、關聯代數、關聯邏輯計算法提出 SEQUEL 語言
- ANSI和ISO共同制訂資料庫電腦語言的標準，依 SEQUEL 為藍本，命名為 SQL
 - ☞ SQL/86
 - ☞ SQL/89 (完整限制語法)
 - ☞ SQL/92或SQL2 (包括四層：entry level、transitional level、intermediate level、和full level)
 - ☞ SQL99/SQL3 (物件導向與資料倉儲)
- 本章介紹SQL99的基本語法，有些DBMS所支援的SQL語法與本章所介紹的有一些差異，但基本觀念雷同

©黃三益2007
資料庫的核心理論與實務第三版

6-2

SQL的關聯模式

- 為了好用性和執行效率，SQL的資料模式與關聯模式有以下差別：
 - ☞ SQL 利用較通俗的名詞來稱呼關聯模式所定義的嚴謹數學名詞
 - 不稱關聯 (Relation)，改稱資料表 (Table)
 - 不稱序列值 (Tuple)，改稱記錄 (Record) 或列 (Row)
 - 不稱屬性 (Attribute)，改稱欄位 (Field) 或行 (Column)
 - ☞ SQL的資料表可以不需定義主鍵
 - ☞ SQL的資料表中可以有兩筆記錄是一模一樣的
 - ☞ 資料表中的記錄是有次序的
- SQL提供了三種語言：
 - ☞ 資料定義語言 (DDL)
 - ☞ 資料處理語言 (DML)
 - ☞ 資料控制語言 (DCL)

©黃三益2007
資料庫的核心理論與實務第三版

6-3

SQL的資料定義語言 (CREATE TABLE)

- **CREATE TABLE** 讓使用者定義一個資料表，包括
 - ☞ 資料表名稱
 - ☞ 欄位
 - ☞ 完整限制

```
CREATE TABLE Member
(mld CHAR(8) NOT NULL,
pld CHAR(10) NOT NULL,
name VARCHAR(8) NOT NULL,
birthday DATE,
phone VARCHAR(10),
address VARCHAR(40),
email VARCHAR(20),
introducer CHAR(8),
PRIMARY KEY (mld),
UNIQUE (pld),
FOREIGN KEY(introducer) REFERENCES
Member(mld)
ON DELETE SET NULL ON UPDATE
CASCADE);
```

©黃三益2007
資料庫的核心理論與實務第三版

6-4

SQL的資料定義語言 (CREATE TABLE)

- SQL所提供的定義域型態

- ☞ INT

- ☞ DECIMAL(i, j)

- DECIMAL(3, 1)

- DECIMAL(2)

- DECIMAL

- ☞ CHAR(n)

- ☞ VARCHAR(n)

- ☞ BIT(n)

- ☞ VARBIT(n)

©黃三益2007
資料庫的核心理論與實務第三版

6-5

SQL的資料定義語言 (CREATE TABLE) (Cont.)

- ☞ DATE

- 標準的日期欄位是yyyy-mm-dd

- ☞ TIME

- 標準的時間欄位是hh:mm:ss

- TIME(2) (13:25:50:30)

- TIME WITH TIME ZONE (13:20:50+08:00)

- ☞ TIMESTAMP (或稱DATETIME)

- 2003-07-10 13:27:50

- ☞ BLOB：表示是儲存大型的二元型態物件

- ☞ CLOB：表示是儲存大型的文字型態物件

©黃三益2007
資料庫的核心理論與實務第三版

6-6

SQL的資料定義語言 (CREATE TABLE)

- 自訂定義域 (**CREATE DOMAIN**)
 - ❧ **CREATE DOMAIN PID_TYPE CHAR(10);**
 - ❧ **CREAE DOMAIN SALES_TYPE INT
CHECK (SALES_TYPE > 100);**
- 作用在單一欄位的完整限制：定義該欄位時一併設定
 - ❧ **NOT NULL**：不得為空值。
 - ❧ **DEFAULT**：設定預設值

©黃三益2007
資料庫的核心理論與實務第三版

6-7

SQL的資料定義語言 (CREATE TABLE)

- 作用在數個欄位的完整限制：
 - ❧ **PRIMARY KEY**：用來設定一資料表的主鍵
 - ❧ **UNIQUE**：該欄位值為唯一，通常是用來設定次要鍵
 - ❧ **FOREIGN KEY**：用來設定外部鍵
 - 一筆記錄被刪除或其主鍵值被修改時，其相對應的外部鍵值會受影響。**FOREIGN KY**裡的語法可以設定這些外部鍵值的處理方式
 - **E.g.**
FOREIGN KEY pNo REFERENCES Product(pNo)
FOREIGN KEY pNo REFERENCES Product

©黃三益2007
資料庫的核心理論與實務第三版

6-8

SQL的資料定義語言 (CREATE TABLE) (Cont.)

■ 設定處理方式如下：

■ ON DELETE RESTRICT (為預設處理方式)

☞ 一筆記錄只有沒被參考時才可被刪除

■ ON DELETE SET NULL

☞ 一筆記錄被刪除時，所有參考它的外部鍵值全部變成空值

■ ON DELETE SET DEFULT

☞ 一筆記錄被刪除時，所有參考它的外部鍵值全部變成預設值

■ ON DELETE CASCADE

☞ 一筆記錄被刪除時，所有參考它的記錄全部跟著被刪除

■ ON UPDATE RESTRICT (為預設處理方式)

☞ 一筆記錄的主鍵值只有沒被參考時才可被修改

■ ON UPDATE CASCADE

☞ 一筆記錄的主鍵值被修改時，所有參考它的外部鍵值全部跟著修改

©黃三益2007
資料庫的核心理論與實務第三版

6-9

```
CREATE TABLE Member
(mId CHAR(8) NOT NULL,
pId VARCHAR(10) NOT NULL,
name VARCHAR(8) NOT NULL,
birthday DATE,
phone VARCHAR(10),
address VARCHAR(40),
email VARCHAR(20),
introducer CHAR(8),
PRIMARY KEY (mId),
UNIQUE (pId),
FOREIGN KEY (introducer) REFERENCES
Member(mId)
ON DELETE SET NULL ON UPDATE
CASCADE);
```

```
CREATE TABLE Cart
(mId CHAR(8) NOT NULL,
cartTime TIMESTAMP, NOT NULL,
tNo CHAR(5),
PRIMARY KEY (mId, cartTime),
FOREIGN KEY (tNo) REFERENCES
Transaction(tNo)
ON UPDATE CASCADE,
FOREIGN KEY (mId) REFERENCES
Member(mId)
ON DELETE CASCADE ON UPDATE
CASCADE);
```

```
CREATE TABLE Transaction
(tNo CHAR(5) NOT NULL,
transMid CHAR(8) NOT NULL,
transTime TIMESTAMP NOT NULL,
method VARCHAR(5) NOT NULL,
bankId VARCHAR(14) NOT NULL,
bankName VARCHAR(20),
cardId VARCHAR(10),
cardType VARCHAR(10),
dueDate DATE,
PRIMARY KEY (tNo),
FOREIGN KEY (transMid) REFERENCES
Member(mId));
```

```
CREATE TABLE Author
(pNo CHAR(6) NOT NULL,
name VARCHAR(8) NOT NULL,
PRIMARY KEY (pNo, name),
FOREIGN KEY (pNo) REFERENCES Product
(pNo));
```

線上購物系統資料庫的SQL定義

©黃三益2007
資料庫的核心理論與實務第三版

6-10

```

CREATE TABLE Browse
(mld CHAR(8) NOT NULL DEFAULT 'a0910001',
pNo CHAR(6) NOT NULL,
browseTime TIMESTAMP NOT NULL,
PRIMARY KEY (mld, pNo, browseTime ),
FOREIGN KEY (mld) REFERENCES Member(mld)
ON DELETE SET DEFAULT ON UPDATE CASCADE,
FOREIGN KEY (pNo) REFERENCES Product (pNo));

CREATE TABLE Order
(pNo CHAR(6) NOT NULL,
mld CHAR(8) NOT NULL,
cartTime TIMESTAMP NOT NULL,
amount INT DEFAULT 1,
PRIMARY KEY (pNo ,mld, cartTime ),
FOREIGN KEY (pNo) REFERENCES Product (pNo),
FOREIGN KEY (mld, cartTime) REFERENCES Cart (mld, cartTime));

CREATE TABLE Product
(pNo CHAR(6) NOT NULL,
pName VARCHAR(10),
unitPrice DECIMAL(10,2),
catalog VARCHAR(20),
PRIMARY KEY (pNo ),
CONSTRAINT UnitPrice_Check
CHECK (unitPrice > 100));

CREATE TABLE Record
(tNo CHAR(5) NOT NULL,
pNo CHAR(6) NOT NULL,
salePrice DECIMAL(10,2) NOT NULL,
amount INT NOT NULL,
PRIMARY KEY (tNo,pNo),
FOREIGN KEY (tNo) REFERENCES Transaction (tNo),
FOREIGN KEY (pNo) REFERENCES Product (pNo));

```

©黃三益2007
資料庫的核心理論與實務第三版

6-11

練習6-1

- 參考線上購物系統 [資料庫綱目](#)，請問在資料表 Member 裡，
 - ☞ 有哪些欄位可以為空值？
 - ☞ 主鍵和次要鍵各為何？
 - ☞ 考慮第四章圖4-4的資料庫，如果名為'Jennifer'的會員記錄被刪除，請問哪些會員的 introducer 欄位值會受到影響？怎樣的影響？
- Ans:
 - 可為空值的欄位有: birthday, phone, address, email, introducer
 - 主鍵: mld
 - 次要鍵: pld
 - 受影響的會員有2筆記錄:
 - ☞ mld為a0910001, name為Jenny
 - ☞ mld為c0927777, name為Su
 - ☞ 兩者的'介紹人'欄位會被設定為空值 (NULL)。

練習6-2

- 參考線上購物系統資料庫綱目的Transaction綱目，假設當一會員記錄被刪除時，我們希望其交易記錄也一併刪除，而當會員編號改變時，其交易記錄的transMid欄位也跟著修改。請問Transaction綱目定義該做怎樣的修改？

- Ans:

```
CREATE TABLE Transaction
```

```
( ...
```

```
FOREIGN KEY (transMid) REFERENCES Member(mId)  
ON DELETE CASCADE ON UPDATE CASCADE);
```

©黃三益2007
資料庫的核心理論與實務第三版

6-13

SQL的資料定義語言 (DROP TABLE)

- **DROP TABLE** 是用來刪除資料表定義
 - ☞ 一資料表被刪除時，另一資料表中參考到該資料表的部分就變的沒有意義了。所以SQL語法裡容許你在刪除資料表定義時設定該如何處理這種狀況
 - ☞ **DROP TABLE Product RESTRICT**
 - 資料表Product的定義只有在其未被參考到時才能被刪除
 - ☞ **DROP TABLE Product CASCADE**
 - 資料表Product的定義被刪除時，所有參考到這些定義的部分（如外部鍵定義）也一併被刪除

©黃三益2007
資料庫的核心理論與實務第三版

6-14

練習6-3

- 參考線上購物系統[資料庫綱目](#)，請問以下SQL指令的執行結果會對資料庫綱目造成何種影響？
 - ☞ DROP TABLE Transaction CASCADE;
 - ☞ DROP TABLE Cart RESTRICT;
- Ans:
 - ☞ 資料表Transaction定義被刪除時,其他資料表定義裡參考到資料表Transaction的部分(包括Cart的tNo外部鍵定義和Record的tNo外部鍵定義)也一併被刪除。
 - ☞ 當其他資料表定義裡都沒有參考到Cart時,才可刪除Cart資料表定義。以圖6-1的資料庫綱目來說,這個刪除不會成功,因為資料表Order有一個外部鍵(mld, cartTime)是參考到Cart。

©黃三益2007
資料庫的核心理論與實務第三版

6-15

SQL的資料定義語言 (ALTER TABLE)

- **ALTER TABLE**是用來修改資料表定義
 - ☞ 欄位定義的新增、刪除和修改的語法如下
 - **ALTER TABLE** Member **ADD** degree **VARCHAR**(10);
 - **ALTER TABLE** Member **DROP** address **CASCADE**;
 - **ALTER TABLE** Order **ALTER** amount **DROP** **DEFAULT**;
 - **ALTER TABLE** Order **ALTER** amount **SET** **DEFAULT** 100;

©黃三益2007
資料庫的核心理論與實務第三版

6-16

SQL的資料定義語言 (ALTER TABLE)

- 完整限制的修改包括完整限制的刪除和新增
 - ☞ 為了能正確刪除完整限制，首先要給完整限制一個名稱
 - ALTER TABLE Product DROP CONSTRAINT UnitPrice_Check;
 - CREATE TABLE Browse
(...
CONSTRAINT mldFK
FOREIGN KEY (mld) REFERENCES Member(mld)
ON DELETE SET DEFAULT ON UPDATE CASCADE,
...)
 - ALTER TABLE Browse DROP CONSTRAINT mldFK;
 - ALTER TABLE Browse ADD CONSTRAINT NewMldFk
FOREIGN KEY (mld) REFERENCES Member(mld)
ON DELETE CASCADE ON UPDATE CASCADE;
 - ☞ 若是忘了先給完整限制一個名稱，則需找出系統所給定的該完整限制編號 (e.g., 透過SHOW CREATE TABLE指令)，再據以修改

©黃三益2007
資料庫的核心理論與實務第三版

6-17

練習6-5

- 參考線上購物系統[資料庫綱目](#)，請將資料表Order的完整限制加上名稱
- Ans

```
CREATE TABLE Order
(pNo CHAR(10) NOT NULL,
mld CHAR(10) NOT NULL,
cartTime TIMESTAMP NOT NULL,
amount INT DEFAULT 0,
PRIMARY KEY (pNo, mld, cartTime),
CONSTRAINT pNoFk
FOREIGN KEY (pNo) REFERENCES Product (pNo),
CONSTRAINT mldFk
FOREIGN KEY (mld, cartTime) REFERENCES Cart (mld, cartTime));
```

©黃三益2007
資料庫的核心理論與實務第三版

6-18

商用DBMS的SQL資料定義語言

- 外部鍵設定時，有些DBMS不支援ON UPDATE或部分ON DELETE 語法
- 有些DBMS對於關鍵字（比如ORDER）拿來當資料表名稱或欄位名稱時須加上雙引號”（如Oracle和SQL Server），有些則需加上特殊引號`（比如MySQL）
- 詳細差異請參考書本第四節第四小節



©黃三益2007
資料庫的核心理論與實務第三版

6-19

基本的SQL查詢語法

- SQL的查詢句基本格式如下：
SELECT<屬性串列>
FROM <資料表串列>
WHERE <條件>
- Q1: 找出所有定價超過500的商品之編號、名稱和定價
 - ☞ 關聯代數
 $\pi_{pNo, pName, unitPrice} (\sigma_{unitPrice > 500} Product)$
 - ☞ 關聯邏輯計算式
{p.pNo, p.pName, p.unitPrice | Product(p), p.unitPrice>500}
 - ☞ SQL
SELECT pNo, pName, unitPrice
FROM Product
WHERE unitPrice > 500;



©黃三益2007
資料庫的核心理論與實務第三版

6-20

基本的SQL查詢語法 (Cont.)

- 列出「系統分析理論與實務」的作者姓名

系統分析理論與實務

Product (商品編號pNo, 商品名稱pName, 定價unitPrice, 種類catalog)

Author (商品編號pNo, 創作者名稱name)

```
SELECT name
FROM Product, Author
WHERE pName = '系統分析理論與實務' AND
      Product.pNo = Author.pNo;
```

©黃三益2007
資料庫的核心理論與實務第三版

6-21

基本的SQL查詢語法 (Cont.)

- 列出所有購買過”系統分析理論與實務”的會員之會員編號和會員姓名

系統分析理論與實務

Product (商品編號pNo, 商品名稱pName, 定價unitPrice, 種類catalog)

Record (交易編號tNo, 商品編號pNo, 交易數量amount, 售價salePrice)

Transaction (交易編號tNo, 會員編號transMid, 交易方式method, ……)

Member (會員編號mid, 姓名name, 會員ID pld, ……)

```
SELECT mid, name
FROM Product, Record, Transaction, Member
WHERE pName='系統分析理論與實務' AND
      Product.pNo = Record.pNo AND
      Record.tNo = Transaction.tNo AND transMid = mid;
```

©黃三益2007
資料庫的核心理論與實務第三版

6-22

練習6-6

- 請找出“陳水扁”所寫過的所有書之商品編號和書名

- Ans:

```
SELECT Product.pNo, pName
FROM Product , Author
WHERE name='陳水扁' AND Author.pNo=Product.pNo
AND catalog = 'Book' ;
```

©黃三益2007
資料庫的核心理論與實務第三版

6-23

基本的SQL查詢語法 (別名)

- 使用資料表名稱的縮寫

☞較簡潔

```
SELECT mld, name
FROM Product AS P, Record AS R, Transaction AS T,
Member
WHERE pName = '系統分析理論與實務' AND P.pNo =
R.pNo AND
R.tNo = T.tNo AND transMid = mld;
```

☞同一資料表出現兩次，且角色不同

- Q4：對於會員編號為'b0905555'的會員，列出其姓名以及所（直接）介紹的會員之會員編號和姓名

©黃三益2007
資料庫的核心理論與實務第三版

6-24

基本的SQL查詢語法 (別名)(Cont.)

I: b0905555
Member (會員編號 mld, 會員姓名 name, 會員ID pld, ……., 介紹人 introducer)
M:
Member (會員編號 mld, 會員姓名 name, 會員ID pld, ……., 介紹人 introducer)

- 其相對應的SQL查詢句如下：

```
SELECT I.name, M.mld, M.name
FROM MEMBER AS I, MEMBER AS M
WHERE I.mld = 'b0905555' AND I.mld = M.introducer;
```



©黃三益2007
資料庫的核心理論與實務第三版

6-25

基本的SQL查詢語法 (別名)

- AS除了可用在資料表的別名外，也可用來更改屬性名稱

☞ 將Q4所傳回的屬性名稱分別改成
introducer_name, member_id和member_name

```
SELECT I.name AS introducer_name, M.mld AS member_id,
       M.name AS member_name
FROM MEMBER AS I, MEMBER AS M
WHERE I.mld = 'b0905555' AND I.mld = M.introducer;
```



©黃三益2007
資料庫的核心理論與實務第三版

6-26

練習6-7

- 請找出會員編號為'b0905555'的會員之介紹者，並列出該會員之姓名和其介紹者的會員編號和姓名

- Ans:

```
SELECT M.name, I.mld, I.name  
FROM MEMBER AS M , MEMBER AS I  
WHERE M.mld = 'b0905555' AND M.introducer = I.mld ;
```

©黃三益2007
資料庫的核心理論與實務第三版

6-27

基本的SQL查詢語法

- 星號 (*)：取出資料表的所有屬性
SELECT *
FROM Product;
- 沒有WHERE子句：沒有條件設限
SELECT mld, name
FROM Member;
- DISTINCT：重複的記錄只保留一筆
SELECT DISTINCT name
FROM Member;
- 欄位值的比較：<、>、<=、>=、!=或<>、
↳ IS NULL：是否空值
↳ BETWEEN：區間

```
SELECT *  
FROM Product  
WHERE unitPrice BETWEEN 100 AND 999;
```

©黃三益2007
資料庫的核心理論與實務第三版

6-28

基本的SQL查詢語法 (Cont.)

- **LIKE**比較子：比較文字欄位的部分字串值

☞ %表示任意字串

☞ _代表任意字元

```
SELECT *  
FROM Member  
WHERE address LIKE '%高雄市%';
```

```
SELECT *  
FROM Member  
WHERE pld LIKE '_2%';
```

©黃三益2007
資料庫的核心理論與實務第三版

6-29

基本的SQL查詢語法 (Cont.)

- **Date**和**Time**的型態值不易直接表達，所以一般是將之轉成字串，或將字串轉成日期時間型態，再做比較

☞ 範例 (Oracle)

```
SELECT mld, name, birthday  
FROM Member  
WHERE birthday >= to_date('19700101', 'yyyymmdd');
```

或

```
SELECT mld, name, birthday  
FROM Member  
WHERE to_char(birthday, 'yyyymmdd') >= '19700101';
```

©黃三益2007
資料庫的核心理論與實務第三版

6-30

基本的SQL查詢語法 (Cont.)

- **ORDER BY**：設定查詢結果資料表裡記錄的排列次序

- ☞ **DESC**代表由大到小

- ☞ **ASC**代表由小到大 (預設)

```
SELECT mld, pld, name
FROM Member
ORDER BY name;
```

```
SELECT mld, pld, name
FROM Member
ORDER BY name DESC;
```

- 在傳回欄位值前，也可用先做一些簡單的運算

```
SELECT pNo, name, 0.9 * unitPrice
FROM Product;
```



©黃三益2007
資料庫的核心理論與實務第三版

6-31

商用DBMS的SQL基本查詢語法

- 有些DBMS沿襲SQL89舊制，別名之前不需加AS，如下：

```
SELECT I.name, M.mld, M.name
FROM MEMBER I, MEMBER M
WHERE I.mld = 'b0905555' AND I.mld = M.introducer;
```

- 有些DBMS對於日期時間欄位值的比較不需任何轉換函式。比如可用以下查詢句找出出生於1970年以後的會員資訊

```
SELECT mld, name, birthday
FROM Member
WHERE birthday >= '19700101';
```



©黃三益2007
資料庫的核心理論與實務第三版

6-32

SQL的更新（新增）

- 標準語法如下：

```
INSERT INTO <資料表名稱>[<屬性串列>]  
VALUES (attribute-value-list);
```

- ☞U1: 新增一筆完整的記錄到「商品」資料表。

```
INSERT INTO Product  
VALUES ('b00001', '資料庫管理與系統', 700, 'Book');
```

- ☞U2: 新增一筆只含商品編號和商品名稱的記錄到「商品」資料表。

```
INSERT INTO Product (pName, pNo)  
VALUES ('資料庫管理與系統', 'b00001');
```



©黃三益2007
資料庫的核心理論與實務第三版

6-33

SQL的更新（新增）(cont.)

- 將查詢的結果整批新增入一個資料表

- ☞U3

```
CREATE TABLE Book  
    (bNo      CHAR(10)  NOT NULL  
     bName    VARCHAR(10),  
     unitPrice DECIMAL(10,2));  
INSERT INTO Book  
SELECT pNo, pName, unitPrice  
FROM Product  
WHERE catalog = 'Book';
```



©黃三益2007
資料庫的核心理論與實務第三版

6-34

SQL的更新（修改）

- 標準語法如下：

UPDATE <資料表名稱>

SET <屬性名稱> = <屬性值>, <屬性名稱> = <屬性值>, ……

WHERE <條件>;

- ☞ U4: 將「資料庫理論與實務」這本書的定價改成700。

UPDATE Product

SET unitPrice = 700

WHERE pName = '資料庫理論與實務';



©黃三益2007
資料庫的核心理論與實務第三版

6-35

SQL的更新（刪除）

- 標準語法如下：

DELETE <資料表名稱>

WHERE <條件>;

- ☞ U5: 將名為「資料庫理論與實務」的商品刪除。

DELETE Product

WHERE name = '資料庫理論與實務';



©黃三益2007
資料庫的核心理論與實務第三版

6-36

練習6-9

- 請新增一本新書，書名為「總統大選分析」，單價為200，作者為「張大砲」。

- Ans: 假設其商品編號為'b11111'

```
INSERT INTO Product
```

```
VALUES ('b11111', '總統大選分析', 200, 'Book');
```

```
INSERT INTO Author
```

```
VALUES ('b11111', '張大砲');
```



©黃三益2007
資料庫的核心理論與實務第三版

6-37

商用DBMS 的SQL 更新語法

- MySQL 刪除語法DELETE 後需加上FROM，例如，

```
DELETE FROM Product
```

```
WHERE name = '資料庫理論與實務';
```

- 可參考本書網站上為各DBMS所準備的「線上購物系統資料庫」匯入檔，以瞭解各DBMS的細微語法差別
- 可上本書網站，點選「SQL語法練習」，按指示輸入帳號密碼後使用MySQL來練習SQL指令



©黃三益2007
資料庫的核心理論與實務第三版

6-38