

第七章進階的SQL

- 集合運算式
- 巢狀查詢句
 - ↻ IN
 - ↻ EXISTS
- JOIN的查詢句
- 分群彙總查詢句
- SQL的VIEW
- 其他進階語法
 - ↻ 權限控制
 - ↻ 語意限制
 - ↻ 索引
 - ↻ CURSOR和儲存程式
 - ↻ TRIGGER

©黃三益2007
資料庫的核心理論與實務第三版

7-1



集合運算式

- 相乘
SELECT mId, pNo
FROM Member, Product;
- 交集、聯集、差集等運算子則分別用**INTERSECT**、**UNION**，和**EXCEPT**
 - ↻ Q5: 找出所有瀏覽過或購買過「系統分析理論與實務」的會員之會員編號和會員姓名。

```
(SELECT M.mId, M.name  
FROM Product AS P, Browse AS B, Member AS M  
WHERE pName = '系統分析理論與實務' AND P.pNo = B.pNo AND B.mId = M.mId)  
UNION  
(SELECT M.mId, M.name  
FROM Product AS P, Record AS R, Transaction AS T, Member AS M  
WHERE pName = '系統分析理論與實務' AND P.pNo = R.pNo AND R.tNo = T.tNo  
AND T.transMid = M.mId);
```

©黃三益2007
資料庫的核心理論與實務第三版

7-2

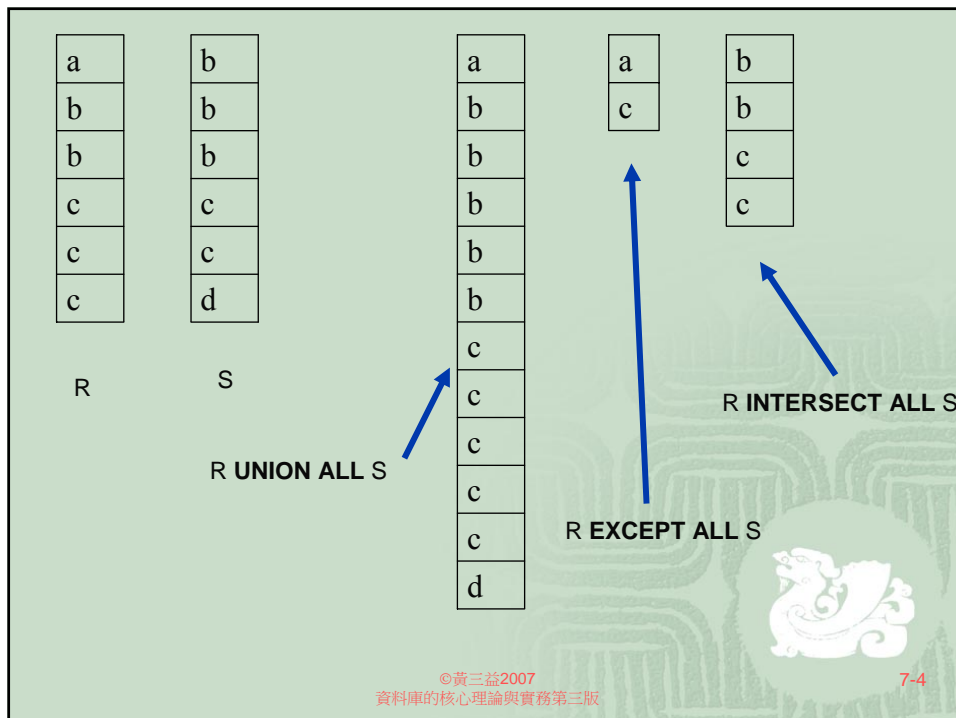


Bag 運算式

- 包括 **UNION ALL**、**EXCEPT ALL** 和 **INTERSECT ALL**
- **R1 UNION ALL R2**：結合 R1 跟 R2 的所有記錄 (不去除重複)。
- **R1 EXCEPT ALL R2**：若一個記錄在 R1 出現 k_1 次但在 R2 出現 K_2 次，則最後的結果裡該記錄出現 $K_1 - K_2$ 次。
- **R1 INTERSECT ALL R2**：若一個記錄在 R1 出現 k_1 次但在 R2 出現 K_2 次，則最後的結果裡該記錄出現 $\text{Min}(K_1, K_2)$ 次。

©黃三益2007
資料庫的核心理論與實務第三版

7-3



©黃三益2007
資料庫的核心理論與實務第三版

7-4

練習7-1

- 請找出所有既出CD也出書的創作者。
- Ans:

```
(SELECT name
FROM Product AS P, Author AS A
WHERE catalog= 'Book' AND P.pNo = A.pNo)
INTERSECT
(SELECT name
FROM Product AS P, Author AS A
WHERE catalog= 'CD' AND P.pNo = A.pNo);
```

©黃三益2007
資料庫的核心理論與實務第三版

7-5

巢狀查詢句

- 巢狀查詢句：在**FROM**子句或**WHERE**子句裡容納另一個查詢子句。

☞ 在**FROM**子句裡的查詢子句

- 列出「系統分析理論與實務」的作者姓名

```
SELECT name
FROM (SELECT * FROM Product WHERE pName = '系統分析理論與
實務' ) AS P, Author
WHERE P.pNo = Author.pNo;
```

☞ 在**WHERE**子句裡的查詢子句

- 很強的表達能力
- IN 和 EXISTS (用來表達邏輯計算式裡的限制子)

©黃三益2007
資料庫的核心理論與實務第三版

7-6

巢狀查詢句 (IN)

- 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

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

©黃三益2007
資料庫的核心理論與實務第三版

7-7

巢狀查詢句 (IN)

- Q5: 找出所有瀏覽過或購買過「系統分析理論與實務」的會員之會員編號和會員姓名

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

©黃三益2007
資料庫的核心理論與實務第三版

7-8

巢狀查詢句 (IN)

- IN左邊的記錄也可包含兩個或兩個以上的屬性

Q6: 找出購物車裡含有「系統分析理論與實務」的交易之編號

```
SELECT tNo
FROM Cart
WHERE (mld, cartTime) IN
      (SELECT mld, cartTime
       FROM Order, Product
       WHERE Order.pNo=Product.pNo AND pName= '系統分析理論與實務' );
```

或是

```
SELECT tNo
FROM Cart
WHERE (mld, cartTime) IN
      (SELECT mld, cartTime
       FROM Order
       WHERE pNo IN
            (SELECT pNo
             FROM Product
             WHERE pName = '系統分析理論與實務' ));
```

©黃三益2007
資料庫的核心理論與實務第三版



7-9

巢狀查詢句 (IN)

- IN前可以加上NOT來表示否定

找出沒有創作者的商品編號和商品名稱

```
SELECT pNo, pName
FROM Product
WHERE pNo NOT IN
      (SELECT pNo
       FROM Author);
```

©黃三益2007
資料庫的核心理論與實務第三版



7-10

練習7-2

- 用巢狀查詢句找出所有瀏覽但沒有購買過「系統分析理論與實務」的會員之會員編號和會員姓名。

- Ans:

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

©黃三益2007
資料庫的核心理論與實務第三版

7-11

巢狀查詢句 (IN)

- 子查詢句也可以參考到上層查詢句的資料表

Q7: 找出「黃三益」所瀏覽過的商品裡，哪些有真正被他購買。

```
SELECT pNo
FROM Browse AS B, Member AS M
WHERE name = '黃三益' AND B.mld = M.mld AND
      pNo IN (SELECT pNo
              FROM Record AS R, Transaction AS T
              WHERE transMid = B.mld AND T.tNo = R.tNo);
```

©黃三益2007
資料庫的核心理論與實務第三版

7-12

練習7-3

- 請將第五章的Q4用巢狀查詢句表達。(列出所有瀏覽過或購買過「系統分析理論與實務」的會員之會員編號和會員姓名。)

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

- 請將Q7用一般查詢句表達(找出「黃三益」所瀏覽過的商品裡，哪些有真正被他購買)。

```
SELECT DISTINCT pNo
FROM Browse AS B, Member AS M,
Record AS R, Transaction AS T
WHERE M.name = '黃三益' AND
M.mld = B.mld AND
B.pNo=R.pNo AND
R.tNo=T.tNo AND
transMid = M.mld ;
```

©黃三益2007
資料庫的核心理論與實務第三版

7-13



巢狀查詢句 (IN)

- 除了IN之外，還有其他相關的比較運算子，如

☞ $=(>, >=, <, <=)$ **SOME (SELECT...)**。

☞ $=(>, >=, <, <=)$ **ALL (SELECT ...)**

- Q8：找出定價比所有書籍都高的商品：

```
SELECT pNo, pName
FROM Product
WHERE unitPrice > ALL( SELECT unitPrice
FROM Product
WHERE catalog= 'Book' );
```

- IN右邊的資料表內容也可直接寫出

```
SELECT DISTINCT mld
FROM Browse
WHERE pNo IN ( 'b30999' , 'b10234' , 'd11222' );
```

©黃三益2007
資料庫的核心理論與實務第三版

7-14



巢狀查詢句 (EXISTS)

- 單元運算子，用來測試一個資料表是否有記錄

☞ 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

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

©黃三益2007
資料庫的核心理論與實務第三版

7-15



巢狀查詢句 (EXISTS)

- **EXISTS**前可以加上**NOT**來表示否定。

☞ Q10：找出所有非由購物車而來的交易的交易編號和會員編號。

```
SELECT tNo, transMid
FROM Transaction AS T
WHERE NOT EXISTS (SELECT *
                  FROM Cart
                  WHERE tNo = T.tNo);
```

©黃三益2007
資料庫的核心理論與實務第三版

7-16



巢狀查詢句 (EXISTS)

- 可表達邏輯關係較複雜的查詢
- Q11：找出購買所有「Jackey」所創作商品的會員之會員編號和會員姓名。

❏ 假設Jackey所創作的產品所成的集合為J，一位會員（比如張三）所購買的所有產品為C，張三若符合條件則 $J-C=\emptyset$

J：

```
SELECT pNo
FROM Author
WHERE name = 'Jackey' ;
```

C：

```
SELECT pNo
FROM Transaction AS T, Record AS R
WHERE transMid=M.mld AND T.tNo = R.tNo;
```

©黃三益2007
資料庫的核心理論與實務第三版

7-17



巢狀查詢句 (EXISTS)

```
SELECT mld, name
FROM Member AS M
WHERE NOT EXISTS (
    (SELECT pNo
     FROM Author
     WHERE name = 'Jackey' )
EXCEPT
    (SELECT pNo
     FROM Transaction AS T, Record AS R
     WHERE transMid=M.mld AND T.tNo = R.tNo));
```

©黃三益2007
資料庫的核心理論與實務第三版

7-18



JOIN的查詢句

- **JOIN**、**NATURAL JOIN**、和**OUTER JOIN**也可以設定在**FROM**子句裡

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

- 或

```
SELECT name
FROM Product NATURAL JOIN Author
WHERE pName = '系統分析理論與實務';
```

©黃三益2007
資料庫的核心理論與實務第三版

7-19



JOIN的查詢句

- Q12：列出每一位會員的會員編號、姓名、生日，以及其介紹者的會員編號和姓名（如果有的話）。
- 本題需用**LEFT OUTER JOIN**

```
SELECT M.mld AS member_id, M.name AS  
member_name, I.mld AS introducer_mld, I.name AS  
introducer_name  
FROM Member AS M LEFT OUTER JOIN Member AS I  
ON M.introducer = I.mld;
```

©黃三益2007
資料庫的核心理論與實務第三版

7-20



JOIN的查詢句

- Q13：列出每一位會員的會員編號、姓名，以及2005年所瀏覽的商品之商品編號（如果有的話）。

~~SELECT M.mld, name, pNo
FROM Member AS M LEFT OUTER JOIN Browse AS B
ON (M.mld=B.mld AND to_char(browseTime, 'yyyy') = '2005');
WHERE to_char(browseTime, 'yyyy') = '2005' ;~~



©黃三益2007
資料庫的核心理論與實務第三版

7-21

練習7-4

- 列出所有商品的產品編號、產品名稱，以及創作者姓名（如果有的話）。

- Ans:

SELECT pNo, pName, A.name
FROM Product AS P LEFT OUTER JOIN Author AS
A ON P.pNo=A.pNo;



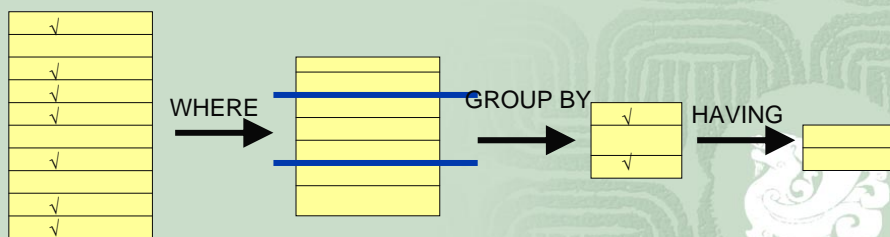
©黃三益2007
資料庫的核心理論與實務第三版

7-22

SQL查詢的彙總函數和分群

- 一般式如下：

```
SELECT <分群屬性>,<彙總函數>  
FROM <資料表>  
WHERE <記錄選取條件>  
GROUP BY <分群屬性>  
HAVING <記錄群選取條件>
```



©黃三益2007
資料庫的核心理論與實務第三版

7-23

彙總函數

- 彙總函數：**SUM**、**AVG**、**COUNT**、**MAX**、和**MIN**
- Q14：列出所有商品數，平均定價，最高定價，和最低定價

```
SELECT COUNT(*), AVG(unitPrice), MAX(unitPrice), MIN(unitPrice)  
FROM Product;
```

```
SELECT COUNT(DISTINCT unitPrice)  
FROM Product;
```

©黃三益2007
資料庫的核心理論與實務第三版

7-24

分群查詢句(cont.)

- Q15：列出每一筆交易的交易編號和交易總金額

```
SELECT tNo, SUM(salePrice)
FROM Record
GROUP BY tNo;
```

- Q16：列出每一筆上網達成的交易（即method = 'cart'）之交易編號，和購買商品總樣數：

```
SELECT tNo, COUNT(*)
FROM Transaction NATURAL JOIN Record
WHERE method = 'cart'
GROUP BY tNo;
```



©黃三益2007
資料庫的核心理論與實務第三版

7-25

分群查詢句(cont.)

- 彙總函數也可用在巢狀查詢句的WHERE子句中

Q17：找出有兩筆以上交易的會員之會員編號和姓名。

```
SELECT mld, name
FROM Member
WHERE (SELECT COUNT(*)
      FROM Transaction
      WHERE mld=transMid)>2;
```

Q18：列出每一筆交易的交易編號、會員編號和交易總金額

```
SELECT tNo, transMid, SUM(salePrice)
FROM Transaction NATURAL JOIN Record
GROUP BY tNo, transMid;
```



©黃三益2007
資料庫的核心理論與實務第三版

7-26

分群查詢句(cont.)

- 數群記錄可用**HAVING**子句來設定挑選條件

Q19：對於每一筆上網達成（即method = 'cart'），且購買商品種樣數超過2的交易，列出其交易編號和購買商品種樣數。

```
SELECT tNo, COUNT(pNo)
FROM Transaction NATURAL JOIN Record
WHERE method = 'cart'
GROUP BY tNo
HAVING COUNT(pNo) > 2;
```

■ :

©黃三益2007
資料庫的核心理論與實務第三版



7-27

練習7-5

- 對於每一台商品種樣數超過3的購物車，列出其購物時間、會員編號、會員姓名

```
SELECT M.mld, M.name, cartTime
FROM Member AS M , Order AS O
WHERE O.mld = M.mld
GROUP BY M.mld, M.name, cartTime
HAVING COUNT(pNo) > 3;
```

©黃三益2007
資料庫的核心理論與實務第三版



7-28

分群查詢句(cont.)

■ HAVING 子句上也可有子查詢句

- 列出所有使用購物車且購買一件以上「黃三益」創作的商品的交易之交易編號和購買總商品樣數

```
SELECT tNo, COUNT(pNo)
FROM (Transaction NATURAL JOIN Record) AS T
WHERE method = 'cart'
GROUP BY T.tNo
HAVING COUNT(pNo) > 2 AND
      1 < (SELECT COUNT(*)
          FROM Record NATURAL JOIN Author
          WHERE name = '黃三益' AND
                T.tNo = tNo);
```

©黃三益2007
資料庫的心理理論與實務第三版

7-29

分群查詢句(cont.)

■ SQL99 允許 EVERY 和 ANY

- 列出所有使用購物車、購買商品樣數超過2，且所有購買的商品價格都超過300的交易之交易編號和購買商品樣數

```
SELECT tNo, COUNT(pNo)
FROM Transaction NATURAL JOIN Record
WHERE method = 'cart'
GROUP BY tNo
HAVING COUNT(pNo) > 2 AND EVERY (salePrice > 300)
```

©黃三益2007
資料庫的心理理論與實務第三版

7-30

分群查詢句

■ 分群彙總結合巢狀查詢

Q20：對於每一筆上網達成（即method = 'cart'），且購買商品種樣數超過2的交易，列出其交易編號和所購買定價超過500的商品種類數。

```
SELECT tNo, COUNT(pNo)
FROM (Transaction NATURAL JOIN Record) NATURAL JOIN Product
WHERE method = 'cart' AND unitPrice > 500
GROUP BY tNo
HAVING COUNT(pNo) > 2;
```

WRONG!



©黃三益2007
資料庫的核心理論與實務第三版

7-31

分群查詢句

■ 正確為：

```
SELECT tNo, COUNT(pNo)
FROM (Transaction NATURAL JOIN Record) NATURAL JOIN Product
WHERE method = 'cart' AND unitPrice > 500
  AND tNo IN (SELECT tNo
              FROM Transaction NATURAL JOIN Record
              WHERE method = 'cart'
              GROUP BY tNo
              HAVING COUNT(pNo) > 2)
GROUP BY tNo;
```



©黃三益2007
資料庫的核心理論與實務第三版

7-32

分群彙總結合OUTER JOIN

- Q21：對於每一位會員，列出其會員編號、總交易數和總交易金額。

```
SELECT mld, COUNT(DISTINCT tNo), SUM(salePrice)  
FROM (Member LEFT OUTER JOIN Transaction  
      ON mld = transMid) NATURAL JOIN Record  
GROUP BY mld;
```



©黃三益2007
資料庫的核心理論與實務第三版

7-33

分群彙總結合OUTER JOIN

- 正確為

```
SELECT mld, COUNT(DISTINCT tNo), SUM(salePrice)  
FROM Member LEFT OUTER JOIN  
      (Transaction NATURAL JOIN Record)  
      ON mld = transMid  
GROUP BY mld;
```



©黃三益2007
資料庫的核心理論與實務第三版

7-34

分群彙總結合更新語法

- U6: 產生一個資料表Transaction_total(tNo, totalAmount)來儲存每一筆交易的總金額。

```
CREATE TABLE Transaction_total
(tNo    VARCHAR(10) NOT NULL,
totalAmount INT);
INSERT INTO Transaction_total
SELECT tNo, SUM(salePrice)
FROM Transaction NATURAL JOIN Record
GROUP BY tNo;
```

- U7: 將有兩位以上創作者的商品定價提高二成。

```
UPDATE Product
SET unitPrice = unitPrice * 1.2
WHERE pNo IN (SELECT pNo
FROM Author
GROUP BY tNo
HAVING COUNT(*) > 2);
```



©黃三益2007
資料庫的核心理論與實務第三版

7-35

練習7-6

- 將居住在台北市的會員之交易商品售價打九折

- Ans:

```
UPDATE Record
SET salePrice = salePrice*0.9
WHERE tNo IN
(SELECT tNo
FROM Transaction, Member
WHERE address LIKE '%台北市%'
AND mId=transMid);
```



©黃三益2007
資料庫的核心理論與實務第三版

7-36

SQL的VIEW

- **VIEW**就是虛擬資料表
 - ☞ 使用者可以像使用一般資料表般的使用**VIEW**，但其實**VIEW**的實體並不存在
- 主要用途：
 - ☞ 常用的查詢定義成**VIEW**，以方便將來使用。
 - ☞ **VIEW**可以形成外部綱目，並據以設定權限，讓某些使用者只能存取**VIEW**裡的欄位，而無法存取實體資料表裡的其他欄位。
- 產生**VIEW**的語法如下：
 - ☞ **CREATE VIEW** <VIEW名稱>
AS <SQL查詢句>
- V1：產生一個**VIEW** Transaction_total(tNo, totalAmount)來表示每一筆交易的總金額。

```
CREATE VIEW Transaction_total(tNo, totalAmount)
AS (SELECT tNo, SUM(salePrice)
    FROM Record
    GROUP BY tNo);
```



©黃三益2007
資料庫的核心理論與實務第三版

7-37

SQL的VIEW (Cont.)

- **VIEW**在查詢句裡的使用方式如同資料表

```
SELECT totalAmount
FROM Transaction_total
WHERE tNo = '91100' ;
```
- 刪除一個**VIEW**

```
DROP VIEW Transaction_total;
```



©黃三益2007
資料庫的核心理論與實務第三版

7-38

SQL的VIEW (Cont.)

- 修改**VIEW**的記錄意味著修改相對應資料表的記錄

V2 : **CREATE VIEW** Cheap_product
 AS (SELECT pNo, pName, unitPrice
 FROM Product
 WHERE unitPrice < 300);

此時，我們可以執行以下的SQL修改句：

UV1 : **UPDATE** Cheap_product
 SET unitPrice = unitPrice * 0.9;



©黃三益2007
資料庫的核心理論與實務第三版

7-39

SQL的VIEW (Cont.)

- 以下的**VIEW**不能被修改，因為沒有唯一的資料表修改方式

- 此**view**包含彙總函數在它的定義中。

UV3 : **UPDATE** Transaction_total
 SET totalAmount = totalAmount - 100;

- 此**view**不包含任何關聯鍵 (key)。

V3 : **CREATE VIEW** Catalog_price
 AS (SELECT catalog, unitPrice
 FROM Product
 WHERE unitPrice > 300);

- 此**view**由兩個或以上個資料表所JOIN而成。



©黃三益2007
資料庫的核心理論與實務第三版

7-40

SQL的VIEW (Cont.)

■ 考慮以下的VIEW

V4 : **CREATE VIEW** Trans_product
AS (SELECT tNo, pName
FROM Record **NATURAL JOIN** Product);

UV4 :
UPDATE Trans_product
SET pName = 'OLAP進階'
WHERE tNo = '91100' **AND** pName = '資料庫理論與實務';

☞ 以下兩種修改方式都滿足UV4

UV5 : **UPDATE** Product
SET pName = 'OLAP進階',
WHERE pNo = 'b30999';

UV6 : **UPDATE** Record
SET pNo = 'b20666'
WHERE tNo = '91100' **AND** pNo = 'b30999';

©黃三益2007
資料庫的核心理論與實務第三版

7-41

SQL的VIEW (Cont.)

■ 練習7-7：

- ☞ 請產生一個VIEW列出每一類商品的名稱和總商品樣數。
- ☞ 該VIEW可以修改嗎？
- ☞ 請刪除該VIEW。

■ Ans:

☞ **CREATE VIEW** Product_amount (catalog, catalogAmount)
AS (SELECT catalog, COUNT(pNo)
FROM Product
GROUP BY catalog);

☞ 因為此view包含彙總函數(COUNT), 所以會造成修改的不明確, 故此view不能被修改

☞ **DROP VIEW** Product_amount

©黃三益2007
資料庫的核心理論與實務第三版

7-42

SQL的權限控制

- 產生table的權力
 - ❏ **GRANT CREATETAB TO Account1;**
- 新增和刪除記錄的權力
 - ❏ **GRANT INSERT, DELETE ON Product TO Account2;**
- 修改記錄屬性的權力
 - ❏ **GRANT UPDATE ON Product (unitPrice) TO Account3;**
- 查詢的權力
 - ❏ **GRANT SELECT ON Product TO Account4;**
- 權利轉移的權力
 - ❏ **GRANT SELECT ON Product TO Account4 WITH GRANT OPTION;**
- 取消權力
 - ❏ **REVOKE SELECT ON Product FROM Account4;**
- 存取VIEW的權力
 - ❏ **GRANT SELECT ON Trans_total TO Account5;**

©黃三益2007
資料庫的核心理論與實務第三版

7-43



語意的限制

- 可用以下語法：
 - ❏ **CREATE ASSERTION <ASSERTION名稱>
CHECK <條件句>;**
 - 交易總金額不得少於100才可用網路交易”。
- ```
CREATE ASSERTION TransactionAmount_Constraint
CHECK (NOT EXISTS
 (SELECT *
 FROM Transaction NATURAL JOIN Record
 WHERE method = 'cart'
 GROUP BY tNo
 HAVING SUM(salesPrice) < 100)) ;
```

©黃三益2007  
資料庫的核心理論與實務第三版

7-44





## SQL的索引

- 索引(Index) 的目的是為了加速查詢的處理

☞ 針對常用來做查詢條件的屬性，建立一個資料結構（稱為**INDEX**），以便加速對該屬性值的搜尋

```
CREATE INDEX Price_index
ON Product(unitPrice);
```

☞ 如下的查詢速度便會大幅加快：

```
SELECT *
FROM Product
WHERE unitPrice BETWEEN 100 AND 2000;
```

©黃三益2007  
資料庫的核心理論與實務第三版

7-45

## SQL的索引 (Cont.)

- 也可以設定在兩個或以上的屬性

```
CREATE INDEX CatPrice_index
ON Product(catalog ASC, unitPrice DESC);
```

☞ 如下的查詢速度便會大幅加快：

```
SELECT *
FROM Product
WHERE catalog = 'Book' AND
unitPrice BETWEEN 100 AND 2000;
```

- 若將索引設定成**CLUSTER**，表示在硬碟中該資料表的記錄實體是按照該索引所設定的次序排列

```
CREATE INDEX PNo_index
ON Product(pNo)
CLUSTER;
```

©黃三益2007  
資料庫的核心理論與實務第三版

7-46

## SQL的CURSOR和儲存程式

- CURSOR是一種將查詢的結果一次回傳一筆記錄的機制

- PL/SQL程式片段：

```
CURSOR price_cursor IS SELECT pNo, pName, unitPrice FROM
Product;
...
OPEN price_cursor;
LOOP
 FETCH unit_cursor INTO a1, a2, a3;
 EXIT WHEN unit_cursor%NOTFOUND;
 ...
ENDLOOP;
CLOSE unit_cursor;
```

©黃三益2007  
資料庫的核心理論與實務第三版



7-47

## SQL的TRIGGER

- DBMS主動偵測資料的內容並採取行動

☞ PL/SQL的一個簡單例子

```
CREATE TRIGGER Transaction_check
AFTER INSERT ON Record
FOR EACH ROW
WHEN (New.salePrice IS NOT NULL)
UPDATE Transaction_total
SET totalAmount=totalAmount + New.salePrice
WHERE tNo=New.tNo;
```

©黃三益2007  
資料庫的核心理論與實務第三版



7-48

## 商用DBMS 的SQL 語法差異

- 有些DBMS沿襲SQL89舊制，使用MINUS而非EXCEPT來表示集合的差集
- 大部分DBMS不支援CREATE ASSERTION，若有語意完整限制的需求，可改用CREATE TRIGGER來達到類似的效果
- 有些DBMS不提供NATURAL JOIN
- 有些DBMS裡，JOIN或不包含關聯鍵的VIEW也可以被修改

©黃三益2007  
資料庫的核心理論與實務第三版

7-49

## 商用DBMS 的SQL 語法差異(Cont.)

- 許多DBMS有特有的CREATE INDEX CLUSTER語法
- 大部分DBMS有其特有的CREATE TRIGGER語法
- 愈來愈多的DBMS有提供全文索引的功能（如SQL Server和MySQL），但語法不同
- 其他差異點請參考書本

©黃三益2007  
資料庫的核心理論與實務第三版

7-50