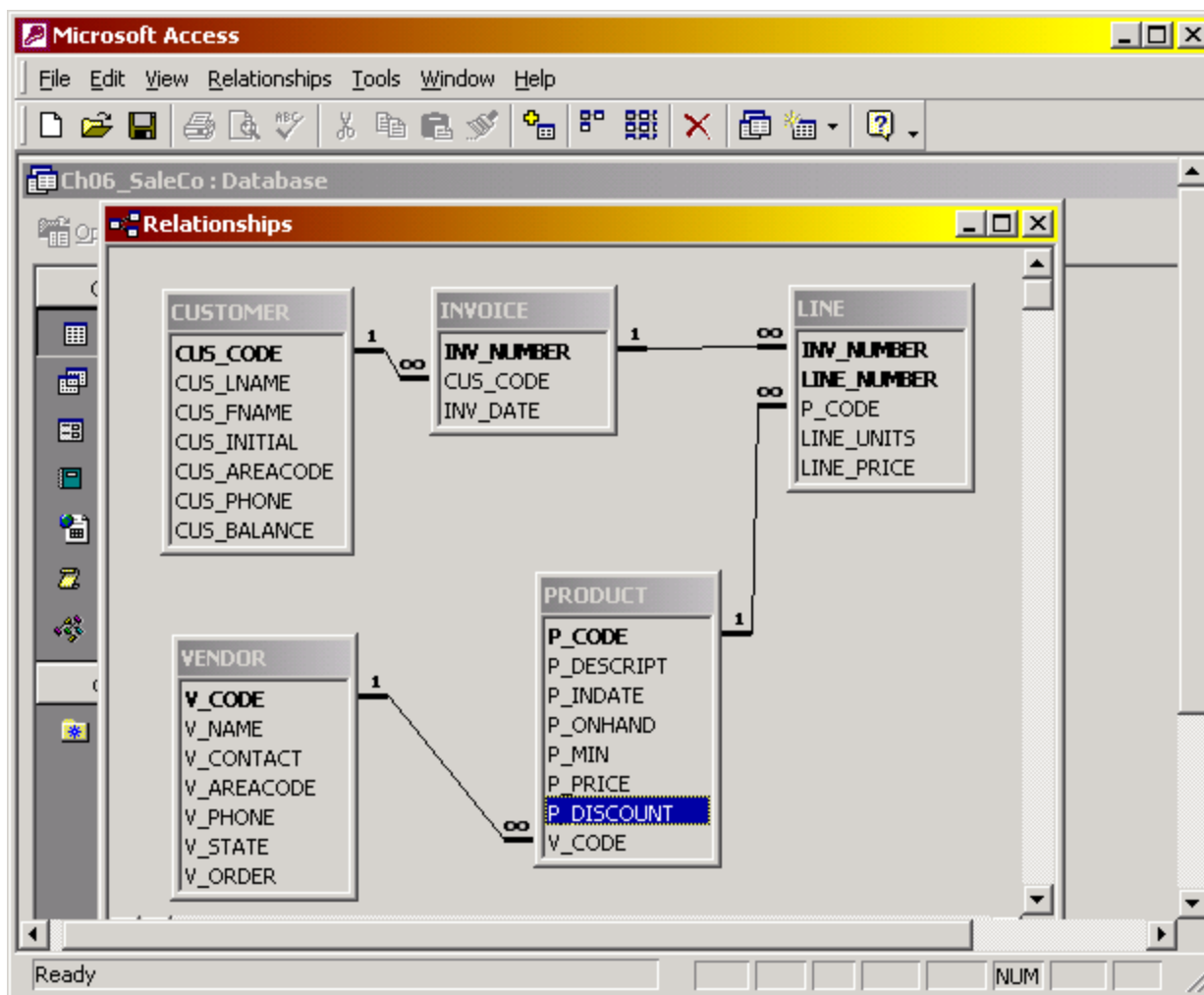


Giới thiệu ngôn ngữ SQL – phần 3

Posts and Telecommunications Institute of Technology-PTIT



Ví dụ về CSDL



Truy vấn nâng cao SELECT

- Một ưu điểm quan trọng của SQL là khả năng tạo các câu truy vấn phức tạp ở dạng tự do.
- Các phép toán logic đề cập trong bài trước có thể được áp dụng trong các câu truy vấn này.
- Ngoài ra, SQL cung cấp các hàm quan trọng như đếm, tìm lớn nhất, nhỏ nhất, tính trung bình, v.v.
- Hơn nữa, SQL cho phép giới hạn câu truy vấn vào các bản ghi không trùng nhau, hoặc nhóm các bản ghi giống nhau vào một.
- Các đặc tính này sẽ được tìm hiểu sâu hơn trong các phần tiếp theo.

Sắp xếp danh sách

- Vế câu ORDER BY rất hữu dụng trong việc sắp xếp danh sách theo yêu cầu nào đó

- Cú pháp:

```
SELECT columnlist  
FROM tablelist  
[ WHERE conditionlist ]  
[ORDER BY columnlist [ASC | DESC] ] ;
```

- Nếu cột được sắp xếp bao gồm giá trị rỗng thì các giá trị này được cho lên đầu hoặc xuống cuối tùy thuộc vào các hệ CSDL quan hệ cụ thể.
- Vế câu ORDER BY luôn nằm cuối cùng trong tập lệnh SELECT.
- Ta có thể chọn cụ thể kiểu sắp xếp (tăng dần hoặc giảm dần). Nếu không chọn, mặc định của hệ thống sẽ là tăng dần.

Sắp xếp danh sách (cont.)

- Câu truy vấn sau liệt kê các bản ghi của bảng PRODUCT được sắp xếp theo thứ tự tăng dần của P_PRICE:

```
SELECT  
    P_CODE, P_DESCRIPT,  
    P_INDATE, P_PRICE  
FROM PRODUCT  
ORDER BY P_PRICE;
```

P_CODE	P_DESCRIPT	P_INDATE	P_PRICE
54778-2T	Rat-tail file, 1/8-in. fine	15-Dec-03	\$4.99
PVC23DRT	PVC pipe, 3.5-in., 8-ft	20-Feb-04	\$5.87
SM-18277	1.25-in. metal screw, 25	01-Mar-04	\$6.99
SW-23116	2.5-in. wd. screw, 50	24-Feb-04	\$8.45
23109-HB	Claw hammer	20-Jan-04	\$9.95
23114-AA	Sledge hammer, 12 lb.	02-Jan-04	\$14.40
13-Q2/P2	7.25-in. pwr. saw blade	13-Dec-03	\$14.99
14-Q1/L3	9.00-in. pwr. saw blade	13-Nov-03	\$17.49
2238/QPD	B&D cordless drill, 1/2-in.	20-Jan-04	\$38.95
1546-QQ2	Hrd. cloth, 1/4-in., 2x50	15-Jan-04	\$39.95
1558-QW1	Hrd. cloth, 1/2-in., 3x50	15-Jan-04	\$43.99
2232/QWE	B&D jigsaw, 8-in. blade	24-Dec-03	\$99.87
2232/QTY	B&D jigsaw, 12-in. blade	30-Dec-03	\$109.92
11QER/31	Power painter, 15 psi., 3-nozzle	03-Nov-03	\$109.99
WR3/TT3	Steel matting, 4'x8'x1/6", .5" mesh	17-Jan-04	\$119.95
89-WRE-Q	Hicut chain saw, 16 in.	07-Feb-04	\$256.99
*			\$0.00

Sắp xếp danh sách (cont.)

- Câu truy vấn sau liệt kê các bản ghi của bảng PRODUCT được sắp xếp theo thứ tự giảm dần của P_PRICE :

```
SELECT  
    P_CODE, P_DESCRIPT,  
    P_INDATE, P_PRICE  
FROM PRODUCT  
ORDER BY P_PRICE DESC;
```

Microsoft Access

File Edit View Insert Format Records Tools Window Help

qryFig6-20 : Select Query

P_CODE	P_DESCRIPT	P_INDATE	P_PRICE
89-WRE-Q	Hicut chain saw, 16 in.	07-Feb-04	\$256.99
WR3/TT3	Steel matting, 4'x8'x1/6", .5" mesh	17-Jan-04	\$119.95
11QER/31	Power painter, 15 psi., 3-nozzle	03-Nov-03	\$109.99
2232/QTY	B&D jigsaw, 12-in. blade	30-Dec-03	\$109.92
2232/QWE	B&D jigsaw, 8-in. blade	24-Dec-03	\$99.87
1558-QW1	Hrd. cloth, 1/2-in., 3x50	15-Jan-04	\$43.99
1546-QQ2	Hrd. cloth, 1/4-in., 2x50	15-Jan-04	\$39.95
2238/QPD	B&D cordless drill, 1/2-in.	20-Jan-04	\$38.95
14-Q1/L3	9.00-in. pwr. saw blade	13-Nov-03	\$17.49
13-Q2/P2	7.25-in. pwr. saw blade	13-Dec-03	\$14.99
23114-AA	Sledge hammer, 12 lb.	02-Jan-04	\$14.40
23109-HB	Claw hammer	20-Jan-04	\$9.95
SWV-23116	2.5-in. wvd. screw, 50	24-Feb-04	\$8.45
SM-18277	1.25-in. metal screw, 25	01-Mar-04	\$6.99
PVC23DRT	PVC pipe, 3.5-in., 8-ft	20-Feb-04	\$5.87
54778-2T	Rat-tail file, 1/8-in. fine	15-Dec-03	\$4.99
*			\$0.00

Record: 1 of 16

Product code: Primary key

NUM

Sắp xếp theo các cấp

- Sắp xếp danh sách là yêu cầu thường gặp. Ví dụ, ta cần tạo một thư mục số điện thoại của nhân viên. Yêu cầu có thể là sắp xếp danh sách theo từng cấp như sau:
 1. Sắp xếp theo họ.
 2. Trong thứ tự sắp xếp đó, sắp xếp theo tên riêng.
 3. Trong thứ tự sắp xếp ở bước 2, sắp xếp theo tên đệm.
- Chuỗi sắp xếp nhiều cấp có thể được tạo ra bởi danh mục các thuộc tính khác nhau, phân tách bởi dấu phẩy sử dụng vế câu ORDER BY.
- Cách thức cụ thể sẽ được minh họa trong các phần tiếp theo.

Sắp xếp theo các cấp (cont.)

Microsoft Access

File Edit View Insert Format Records Tools Window Help

<

Sắp xếp theo các cấp (cont.)

- Để tạo danh mục có sắp xếp từ bảng EMPLOYEE, ta dùng câu truy vấn SQL sau:

```
SELECT EMP_LNAME, EMP_FNAME, EMP_INITIAL, EMP_AREACODE, EMP_PHONE  
FROM EMPLOYEE  
ORDER BY EMP_LNAME, EMP_FNAME, EMP_INITIAL;
```

- Kết quả câu truy vấn này được minh họa trong slide kế tiếp.

Sắp xếp theo các cấp (cont.)

Microsoft Access

File Edit View Insert Format Records Tools Window Help

qryFig6-22 : Select Query

	EMP_LNAME	EMP_FNAME	EMP_INITIAL	EMP_AREACODE	EMP_PHONE
▶	Brandon	Marie	G	901	882-0845
	Diante	Jorge	D	615	890-4567
	Genkazi	Leighla	W	901	569-0093
	Johnson	Edward	E	615	898-4387
	Jones	Anne	M	615	898-3456
	Kolmycz	George	D	615	324-5456
	Lange	John	P	901	504-4430
	Lewis	Rhonda	G	615	324-4472
	Saranda	Hermine	R	615	324-5505
	Smith	George	A	615	890-2984
	Smith	George	K	901	504-3339
	Smith	Jeanine	K	615	324-7883
	Smythe	Melanie	P	615	324-9006
	Vandam	Rhett		901	675-8993
	Washington	Rupert	E	615	890-4925
	Wiesenbach	Paul	R	615	897-4358
	Williams	Robert	D	615	890-3220
*					

Record: 1 of 17

Employee's last name

NUM

Bảng Employee – Sắp xếp theo LastName, FirstName, MiddleInitial

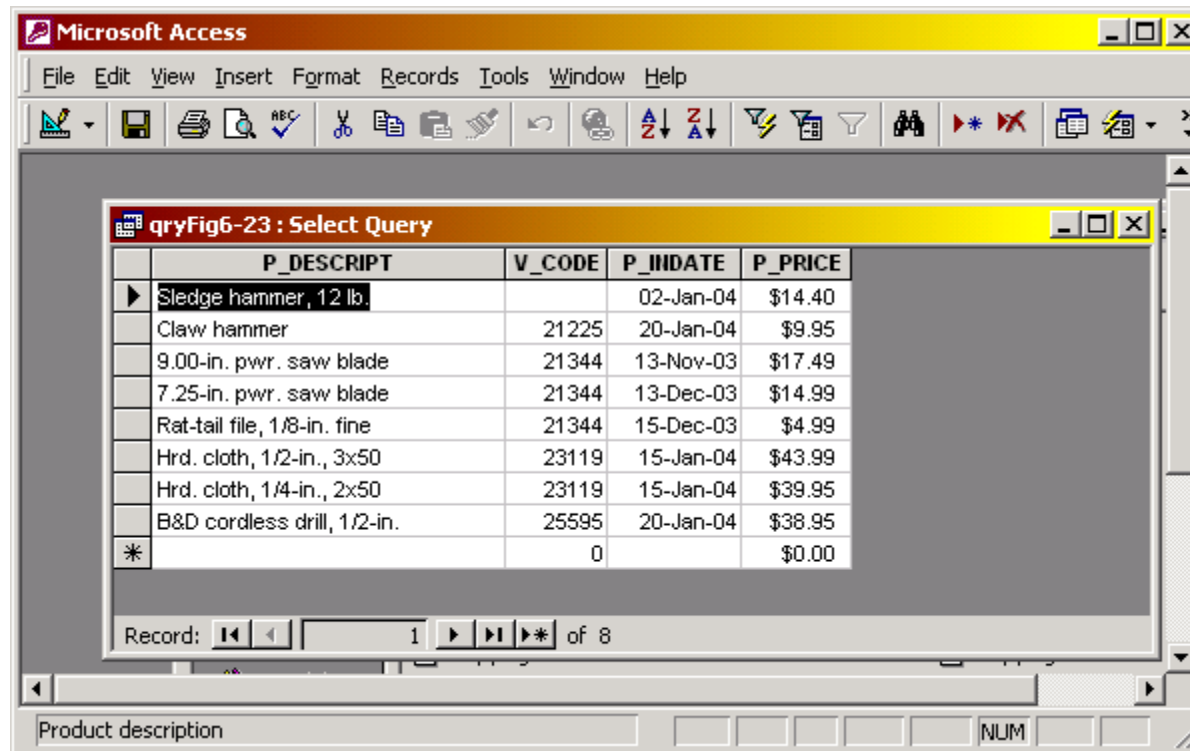
Một số ứng dụng phụ của ORDER BY

- Vế câu ORDER BY có thể được sử dụng kết hợp với các lệnh SQL khác.
- Ví dụ, chú ý việc sử dụng các điều kiện ràng buộc trên ngày và giá trong chuỗi truy vấn sau:

```
SELECT P_DESCRIPT, V_CODE, P_INDATE, P_PRICE  
FROM PRODUCT  
WHERE P_INDATE < '21-Jan-2004' AND P_PRICE <= 50.00  
ORDER BY V_CODE, P_PRICE DESC;
```

- Kết quả nằm ở slide kế tiếp:

Một số ứng dụng phụ của ORDER BY (cont.)



The screenshot shows the Microsoft Access application window. The title bar reads "Microsoft Access". The menu bar includes "File", "Edit", "View", "Insert", "Format", "Records", "Tools", "Window", and "Help". The toolbar contains various icons for file operations, editing, and navigation. The main window displays a query titled "qryFig6-23 : Select Query". The query results are shown in a table with the following columns: P_DESCRIPT, V_CODE, P_IIDATE, and P_PRICE. The table contains 8 records. The first record is "Sledge hammer, 12 lb." with V_CODE 21225, P_IIDATE 02-Jan-04, and P_PRICE \$14.40. The last record is marked with an asterisk (*) and has V_CODE 0 and P_PRICE \$0.00. The status bar at the bottom shows "Record: 1 of 8" and a "Product description" field.

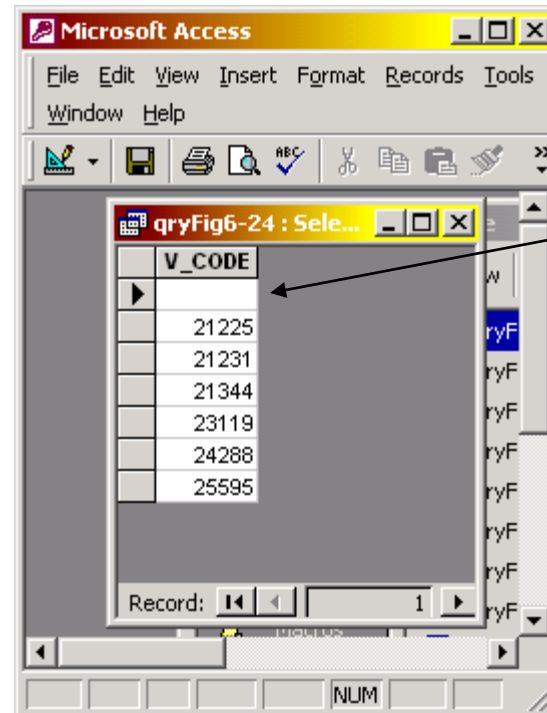
P_DESCRIPT	V_CODE	P_IIDATE	P_PRICE
Sledge hammer, 12 lb.		02-Jan-04	\$14.40
Claw hammer	21225	20-Jan-04	\$9.95
9.00-in. pwr. saw blade	21344	13-Nov-03	\$17.49
7.25-in. pwr. saw blade	21344	13-Dec-03	\$14.99
Rat-tail file, 1/8-in. fine	21344	15-Dec-03	\$4.99
Hrd. cloth, 1/2-in., 3x50	23119	15-Jan-04	\$43.99
Hrd. cloth, 1/4-in., 2x50	23119	15-Jan-04	\$39.95
B&D cordless drill, 1/2-in.	25595	20-Jan-04	\$38.95
*	0		\$0.00

Liệt kê các giá trị duy nhất

- Có bao nhiêu nhà cung cấp có tên trong bảng PRODUCT? Câu lệnh liệt kê đơn giản dùng SELECT không hữu hiệu cho câu truy vấn này, đặc biệt nếu trong bảng có hàng nghìn bộ dữ liệu.
- Về câu lệnh DISTINCT của SQL cho phép tạo danh mục các giá trị duy nhất từ một danh sách đã cho.
- Ví dụ, câu lệnh sau:

```
SELECT DISTINCT V_CODE  
FROM PRODUCT;
```

sẽ liệt kê các mã V_CODE khác nhau
có trong bảng PRODUCT.



Trong Oracle, V_CODE có giá trị rỗng sẽ nằm ở cuối, còn trong Access thì sẽ nằm ở đầu. Ta cũng có thể dùng về ORDER BY để sắp xếp.

Các hàm trong SQL

Trong quá trình truy vấn dữ liệu có thể sử dụng một số hàm trên các cột dữ liệu với cú pháp sau:

function_name(cột, [tham số 1, tham số 2, ...])

- Các hàm thao tác trên từng bản ghi:
 - LOWER(A) – chuyển đổi ký tự thành kiểu chữ thường
 - UPPER(a) – chuyển đổi ký tự thành kiểu chữ in hoa

Ví dụ: **SELECT UPPER(LastName), LOWER(FirstName) FROM student ;**

- ROUND(a) – làm tròn số
- PI() – lấy giá trị pi
- SQRT(a) – căn bậc hai
- POWER(a,b) – hàm mũ



Các hàm trong SQL

- Các hàm thao tác trên nhiều bản ghi:
 - MAX() – tìm giá trị lớn nhất cho các thuộc tính kiểu số
 - MIN() – tìm giá trị nhỏ nhất cho các thuộc tính kiểu số
 - AVG() – tìm giá trị trung bình cho các thuộc tính kiểu số
 - COUNT() – đếm số bản ghi
 - SUM() – tìm tổng các giá trị cho các thuộc tính kiểu số

Ví dụ:

```
SELECT AVG(Age) , SUM(Salary)
```

```
FROM nhanvien;
```



Gom nhóm các kết quả truy vấn

- Tần suất phân bố các giá trị trả về sẽ được tự động tạo ra bởi vế câu lệnh GROUP BY bên trong câu lệnh SELECT.

- Cú pháp là:

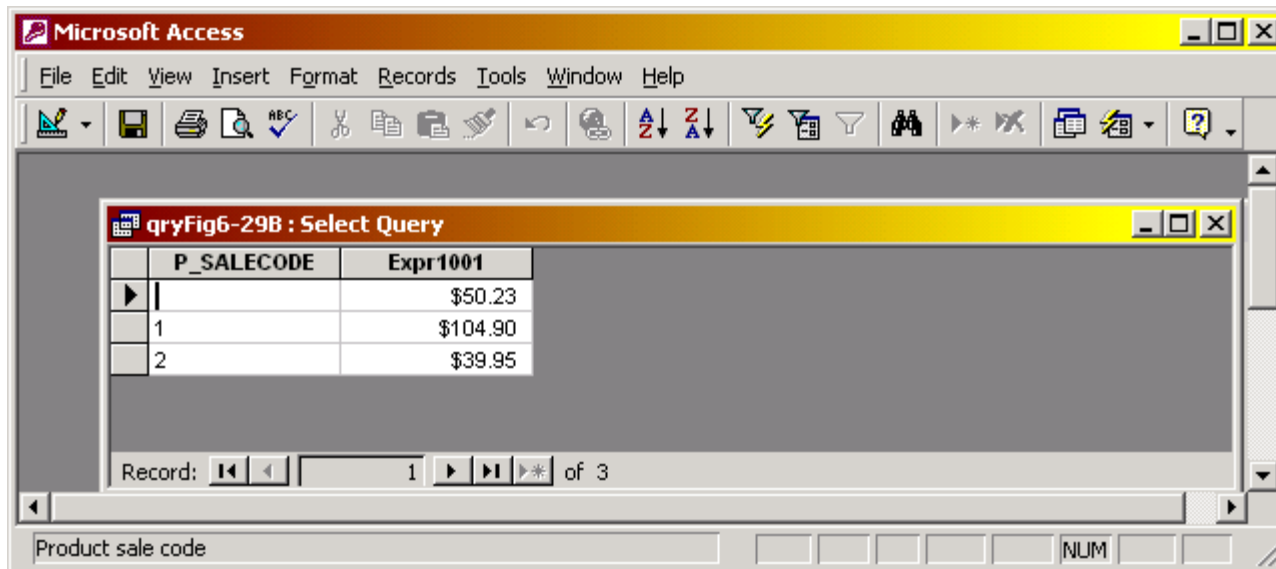
```
SELECT columnlist
FROM tablelist
[WHERE conditionlist ]
[GROUP BY columnlist ]
[HAVING condtionlist ]
[ORDER BY columnlist [ASC | DESC] ] ;
```

- Vế câu lệnh GROUP BY thường được sử dụng trong các cột thuộc tính có các hàm thống kê (MAX, MIN, AVG, SUM, COUNT) trong câu lệnh SELECT.
- Ví dụ, tìm giá bán nhỏ nhất cho mỗi mã hàng. Câu lệnh được thực hiện như ở slide kế tiếp

Gom nhóm các kết quả truy vấn (cont.)

- Câu truy vấn:

```
SELECT P_SALECODE, MIN(P_PRICE)
FROM PRODUCT
GROUP BY P_SALECODE;
```



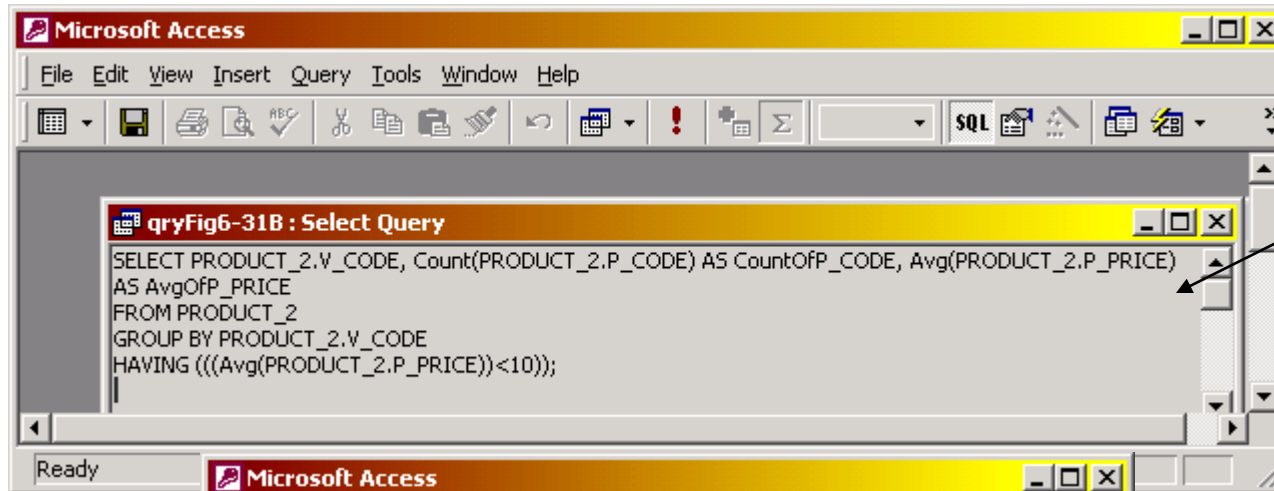
Gom nhóm các kết quả truy vấn (cont.)

- Một số quy tắc cần nhớ khi sử dụng vế câu GROUP BY trong câu lệnh SELECT:
 - Trong danh mục *columnlist* của SELECT **phải có** sự kết hợp giữa tên cột và các **hàm thống kê**.
 - Trong danh mục *columnlist* của vế GROUP BY bao gồm các cột trong *columnlist* của phần SELECT mà **không** chứa hàm thống kê. Tùy từng trường hợp, ta có thể gom nhóm theo bất kỳ cột chứa hàm thống kê nào có trong *columnlist* của phần SELECT.
 - Danh mục *columnlist* của vế GROUP BY có thể bao gồm bất kỳ cột nào trong bảng xác định bởi vế FROM của câu lệnh SELECT, kể cả khi chúng không xuất hiện trong danh mục *columnlist* của SELECT.

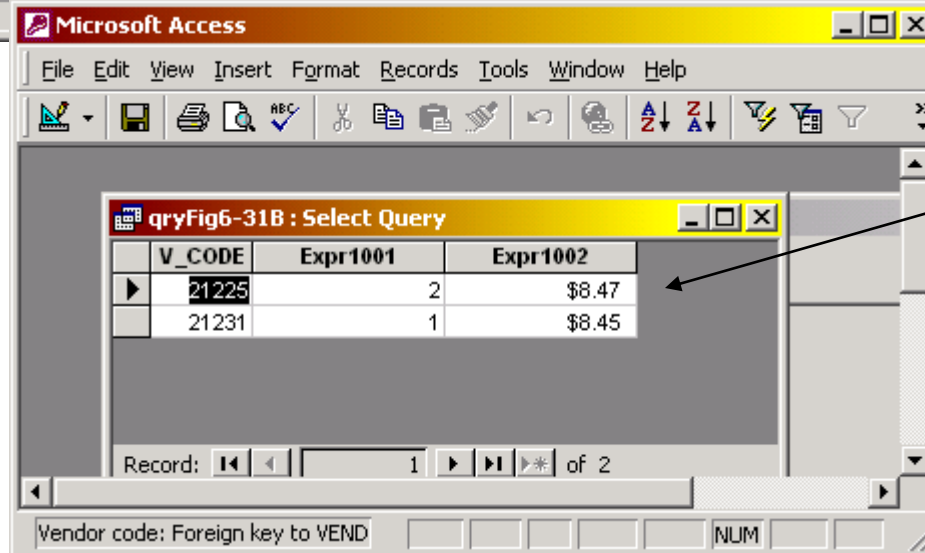
Về câu HAVING của GROUP BY

- Về cơ bản, HAVING hoạt động giống như vế câu WHERE trong câu lệnh SELECT. Tuy nhiên, vế câu WHERE chỉ đến các giá trị trong từng cột và từng hàng cụ thể của các bảng trong vế câu FROM, trong khi HAVING chỉ đến kết quả đầu ra của câu lệnh GROUP BY.
- Ví dụ, ta cần liệt kê số sản phẩm có trong kho của từng nhà cung cấp, tuy nhiên ta chỉ cần các mặt hàng có giá trung bình dưới \$10.00. Vế thứ nhất được thực hiện bởi lệnh GROUP BY, vế thứ hai được thực hiện bởi lệnh HAVING.

Về câu HAVING của GROUP BY (cont.)



The query



The results

Các bảng hiển thị: Tạo hiển thị

- Đầu ra của một câu lệnh quan hệ (chẳng hạn, SELECT trong SQL) là các quan hệ (hay là bảng).
- Ví dụ như trong CSDL đã có, giả sử ta cần tạo danh mục các sản phẩm cần nhập thêm vào cuối ngày, các sản phẩm này có số lượng tồn kho nhỏ hơn một ngưỡng cho trước.
- Thay vì phải nhập cùng một câu truy vấn từ ngày này sang ngày khác, ta cần phải lưu vĩnh viễn câu lệnh đó vào CSDL.
- Chức năng hiển thị quan hệ cho phép việc này. Trong SQL, phần hiển thị (**view**) là một bảng tạo ra từ câu truy vấn SELECT. Câu truy vấn này có thể gồm cột, các cột tính toán, cột tên riêng, hoặc các hàm thống kê từ một hoặc nhiều bảng.
- Các bảng cấp dữ liệu cho phần hiển thị gọi là bảng gốc (**base tables**).
- Giao diện hiển thị được tạo ra trong SQL bằng câu lệnh CREATE VIEW.

Các bảng hiển thị: Tạo hiển thị (cont.)

- Cú pháp của câu lệnh CREATE VIEW là:

```
CREATE VIEW viewname AS SELECT query
```

- Câu lệnh CREATE VIEW thuộc về DDL, nó bao gồm các câu lệnh con bên trong, chẳng hạn như lệnh SELECT để tạo ra các bảng hiển thị của CSDL.

- Ví dụ:

```
CREATE VIEW PRODUCT_3 AS  
    SELECT P_DESCRIPT, P_ONHAND, P_PRICE  
    FROM PRODUCT  
    WHERE P_PRICE > 50.00;
```

- Lưu ý: Câu lệnh CREATE VIEW không dùng được trực tiếp trong Access. Để tạo giao diện hiển thị trong Access, ta cần phải tạo câu truy vấn SQL rồi lưu nó lại.

Các bảng hiển thị: Tạo hiển thị (cont.)

- Một giao diện hiển thị có một vài đặc tính như sau:
 - Ta có thể sử dụng tên của mục hiển thị thay cho tên của bảng trong câu truy vấn SQL.
 - Giao diện hiển thị được cập nhật liên tục. Nói cách khác, giao diện được tự động tạo ra khi câu lệnh được kích hoạt.
 - Giao diện hiển thị cho phép giới hạn người xem theo từng cột và hàng dữ liệu.
 - Giao diện hiển thị có thể được sử dụng làm nội dung báo cáo. Câu lệnh sau tạo ra bảng tổng kết về tổng số giá trị hàng hóa cũng như số lượng tồn kho thống kê theo từng nhà cung cấp:

```
CREATE VIEW SUMPRDXVEN AS
  SELECT V_CODE, SUM(P_ONHAND*P_PRICE) AS TOTCOST,
         MAX(P_ONHAND) AS MAXQTY, MIN(P_ONHAND) AS MINQTY,
         AVG(P_ONHAND) AS AVGQTY
  FROM PRODUCT
  GROUP BY V_CODE;
```



```
select filds  
from table1  
inner join table2 on table1.fildi = table2.fildi
```

- **inner join** : trả về các bản ghi có giá trị phù hợp giữa hai bảng (*nhớ lại phép giao hai tập hợp*).
- **left join** : mọi bản ghi bảng bên trái được trả về, bản ghi nào phù hợp với bản ghi bên phải thì nó được bổ sung thêm dữ liệu từ bản ghi bảng bên phải (nếu không có thì nhận NULL)
- **right join** : mọi bản ghi bảng bên phải được trả về, sau bổ sung dữ liệu phù hợp từ bảng bên trái.
- **outer join** : (full join) mọi bản ghi ở bảng trái và bảng phải kết hợp lại