

Bài thực hành tuần 6

Truy vấn nhóm

❖ **Nội dung chính:** Trong bài này, chúng ta sẽ làm quen với các hàm nhóm và truy vấn nhóm:

- Các hàm nhóm: Sum, AVG, MAX và MIN, Count
- Mệnh đề GROUP BY
- Mệnh đề HAVING

1. Các hàm nhóm

Hàm SUM

Đôi khi các thông tin chúng ta cần không được lưu trữ thực sự trong các bảng cơ sở dữ liệu, nhưng chúng ta có thể lấy được chúng bằng cách tính toán từ dữ liệu được lưu trữ. Ví dụ, chúng ta có bảng *OrderDetails* để lưu trữ thông tin về các đơn đặt hàng. Khi chúng ta nhìn vào đó, chúng ta không biết tổng số tiền của tất cả các sản phẩm bán được là bao nhiêu. Tuy nhiên, hàm tính tổng SUM có thể giúp chúng ta trả lời câu hỏi này. Trước hết chúng ta xem hoạt động của hàm SUM, việc thực hiện nhóm dữ liệu sẽ trình bày trong phần 2

Ví dụ: Tính tổng số lượng hàng hóa hiện còn trong kho

```
SELECT sum(quantityInStock)
FROM products
```

Kết quả trả về như sau:

	sum(quantityInStock)
▶	555131

Hoặc để tính tổng số tiền chúng ta đã thu được từ đầu tới giờ, viết truy vấn như sau:

```
SELECT sum(priceEach * quantityOrdered) total
```

FROM orderdetails

Kết quả trả về như sau:

	total
▶	9857225.609999985

Hàm AVG

AVG được sử dụng để tính giá trị trung bình của một biểu thức, Nó không chấp nhận giá trị NULL. Chúng ta có thể sử dụng AVG để tính toán giá trung bình của tất cả các sản phẩm đã mua như sau:

```
SELECT AVG(buyPrice) average_buy_price  
FROM Products
```

Kết quả trả về như sau:

	average_buy_price
▶	54.395181818181825

Hàm MAX và MIN

Hàm MAX trả về giá trị lớn nhất và hàm MIN trả về giá trị nhỏ nhất của một tập các giá trị.

MAX(expression)

MIN(expression)

Ví dụ: Sử dụng MAX và MIN để lấy ra mức giá cao nhất và mức giá nhỏ nhất của sản phẩm.

```
SELECT MAX(buyPrice) highest_price,  
       MIN(buyPrice) lowest_price  
FROM Products
```

Kết quả trả về như sau:

	highest_price	lowest_price
▶	103.42	15.91

Hàm COUNT

Hàm COUNT là hàm đếm số lượng, chẳng hạn chúng ta có thể đếm số lượng sản phẩm đang được bán như sau:

```
SELECT COUNT(*) AS Total
FROM products
```

Kết quả trả về như sau:

	Total
▶	110

Lưu ý: một phiên bản khác của hàm COUNT sử dụng tham số là tên cột. Nếu cách này được sử dụng, sẽ chỉ đếm các dòng mà giá trị tại cột đó là khác NULL.

2. Mệnh đề nhóm GROUP BY

Mệnh đề **GROUP BY** được sử dụng để gộp các bản ghi có cùng giá trị tại một hay nhiều cột, thành một tập hợp. GROUP BY nếu có thì nó phải đứng sau mệnh đề WHERE hoặc FROM. Theo sau từ khoá GROUP BY là một danh sách các biểu thức, phân cách nhau bởi dấu phẩy.

```
SELECT col1_, col_2, ... col_n, các hàm nhóm(biểu thức)
FROM tên bảng
WHERE điều kiện
GROUP BY col_1, col_2, ... col_n
ORDER BY danh sách cột
```

Theo định nghĩa, hàm nhóm cho phép chúng ta thực hiện một phép tính trên một tập bản ghi và trả về một giá trị. Hàm nhóm bỏ qua các giá trị null khi thực hiện tính toán, ngoại trừ hàm COUNT. Hàm nhóm thường được sử dụng với mệnh đề GROUP BY của câu lệnh SELECT.

Ví dụ: Giả sử muốn phân chia các đơn đặt hàng theo các nhóm phụ thuộc vào tình trạng của các đơn hàng, có thể làm như sau:

```
SELECT status
FROM orders
GROUP BY status
```

Kết quả trả về như sau:

	status
▶	Cancelled
	Disputed
	In Process
	On Hold
	Resolved
	Shipped

Các hàm nhóm được sử dụng với GROUP BY để thực hiện tính toán trên mỗi nhóm các bản ghi và trả về một giá trị duy nhất cho mỗi hàng.

Ví dụ: muốn biết có bao nhiêu đơn đặt hàng trong từng nhóm trạng thái, có thể sử dụng hàm COUNT như sau:

```
SELECT status, count(*)
FROM orders
GROUP BY status
```

Kết quả trả về như sau:

	status	count(*)
▶	Cancelled	6
	Disputed	3
	In Process	6
	On Hold	4
	Resolved	4
	Shipped	303

Ví dụ: muốn biết có bao nhiêu loại sản phẩm trong mỗi loại dòng sản phẩm

```
SELECT productLine, count(*)  
FROM products  
GROUP BY productline
```

	productLine	count(*)
►	Classic Cars	38
	Motorcycles	13
	Planes	12
	Ships	9
	Trains	3
	Trucks and Buses	11
	Vintage Cars	24

Ví dụ: Để có được tổng số tiền cho mỗi sản phẩm đã bán, chúng ta chỉ cần sử dụng chức năng SUM và nhóm sản phẩm. Dưới đây là truy vấn:

```
SELECT productCode, sum(priceEach * quantityOrdered) total  
FROM orderdetails  
GROUP by productCode
```

Kết quả trả về như sau:

	productCode	total
►	S10_1678	90157.770000000002
	S10_1949	190017.959999999996
	S10_2016	109998.819999999998
	S10_4698	170685.999999999997
	S10_4757	127924.319999999999
	S10_4962	123123.009999999998
	S12_1099	161531.479999999992
	S12_1108	190755.86
	S12_1666	119085.249999999999
	S12_2823	135767.030000000003
	S12_3148	132363.789999999998
	S12_3380	98718.760000000001

Ví dụ: Giả sử chúng ta muốn xem các kết quả của truy vấn trên, hiển thị theo thứ tự tăng dần chúng ta làm như sau:

```
SELECT productCode, sum(priceEach * quantityOrdered) total
FROM orderdetails
GROUP BY productCode
ORDER BY total DESC
```

Kết quả trả về như sau:

	productCode	total
►	S18_3232	276839.98
	S12_1108	190755.86
	S10_1949	190017.95999999996
	S10_4698	170685.99999999997
	S12_1099	161531.47999999992
	S12_3891	152543.02
	S18_1662	144959.90999999997
	S18_2238	142530.62999999998
	S18_1749	140535.60000000003
	S12_2823	135767.03000000003
	S24_3856	134240.71
	S12_3148	132363.78999999998
	S18_2795	132275.97999999998
	S18_4721	130749.31000000001
	S10_4757	127924.31999999999
	S10_4962	123123.00999999998
	S18_4027	122254.75
	S18_3482	121890.6

Lưu ý: sự khác nhau giữa GROUP BY trong MySQL và ANSI SQL

MySQL tuân theo chuẩn ANSI SQL. Tuy nhiên, có 2 sự khác biệt khi sử dụng GROUP BY trong MySQL như sau:

- Trong ANSI SQL, phải thực hiện GROUP BY tất cả các cột xuất hiện trong mệnh đề SELECT. MySQL không đòi hỏi như vậy, có thể đưa thêm các cột vào trong mệnh đề SELECT và không bắt buộc chúng phải xuất hiện ở mệnh đề GROUP BY.

- MySQL cũng cho phép sắp xếp các nhóm theo thứ tự các kết quả tính toán, mặc định là giảm dần.

3. Mệnh đề điều kiện HAVING

HAVING cũng là một mệnh đề có thể xuất hiện hoặc không trong mệnh đề SELECT. Nó chỉ ra một điều kiện lọc trên dữ liệu là một nhóm các bản ghi hoặc là kết quả của việc thực hiện hàm nhóm. HAVING thường được sử dụng cùng với GROUP BY, khi đó điều kiện lọc chỉ được áp dụng trên các cột xuất hiện trong mệnh đề GROUP BY mà thôi. Nếu HAVING không đi kèm với GROUP BY, khi đó nó có ý nghĩa như WHERE mà thôi. Lưu ý rằng, HAVING áp dụng trên các nhóm bản ghi, còn WHERE áp dụng trên từng bản ghi riêng lẻ.

Ví dụ: Chúng ta sử dụng mệnh đề GROUP BY để có được tất cả các đơn đặt hàng, số lượng các mặt hàng bán ra và tổng giá trị trong mỗi đơn đặt hàng như sau:

```
SELECT ordernumber,
       sum(quantityOrdered) AS itemCount,
       sum(priceEach * quantityOrdered) AS total
FROM orderdetails
GROUP BY ordernumber
```

	ordernumber	itemsCount	total
▶	10100	151	10223.829999999998
	10101	142	10549.01
	10102	80	5494.78
	10103	541	50218.950000000004
	10104	443	40206.2
	10105	545	53959.21
	10106	675	52151.810000000005
	10107	229	22292.620000000003
	10108	561	51001.219999999994
	10109	212	25833.14
	10110	570	48425.69
	10111	217	16537.850000000002
	10112	52	7674.9400000000005
	10113	143	11044.300000000001
	10114	351	33383.140000000001
	10115	210	21665.980000000003
	10116	27	1627.56
	10117	402	44380.15

Bây giờ, có thể yêu cầu hiển thị chỉ những đơn hàng có tổng giá trị lớn hơn \$1000 bằng cách sử dụng HAVING như sau:

```
SELECT ordernumber,  
       sum(quantityOrdered) AS itemCount,  
       sum(priceEach * quantityOrdered) AS total  
FROM orderdetails  
GROUP BY ordernumber  
HAVING total > 1000
```

	ordernumber	itemCount	total
▶	10100	151	10223.829999999998
	10101	142	10549.01
	10102	80	5494.78
	10103	541	50218.950000000004
	10104	443	40206.2
	10105	545	53959.21
	10106	675	52151.810000000005
	10107	229	22292.620000000003
	10108	561	51001.219999999994
	10109	212	25833.14
	10110	570	48425.69
	10111	217	16537.850000000002
	10112	52	7674.9400000000005
	10113	143	11044.300000000001
	10114	351	33383.140000000001
	10115	210	21665.980000000003
	10116	27	1627.56
	10117	402	44380.15

Chúng ta sử dụng bí danh cho cột `sum (priceEach * quantityOrdered)` là *total*, như vậy trong mệnh đề HAVING, chúng ta chỉ cần dùng *bí danh* đó thay vì gõ `sum(priceeach)` một lần nữa.

Có thể sử dụng một điều kiện kết hợp trong mệnh đề HAVING với các toán tử OR, AND.

Ví dụ: nếu muốn biết những đơn hàng có tổng giá trị lớn hơn \$ 1000 và có hơn 600 mặt hàng trong đó, có thể sử dụng truy vấn sau đây:


```

SELECT ordernumber,
       sum(quantityOrdered) AS itemCount,
       sum(priceeach) AS total
FROM orderdetails
GROUP BY ordernumber
HAVING total > 1000 AND itemCount > 600

```

Kết quả trả về như sau:

	ordernumber	itemCount	total
▶	10106	675	1427.2800000000002
	10126	617	1623.71
	10135	607	1494.86
	10165	670	1794.9399999999996
	10168	642	1472.5
	10204	619	1619.73
	10207	615	1560.08
	10212	612	1541.8300000000002
	10222	717	1389.51
	10262	605	1217.38
	10275	601	1455.4099999999999
	10310	619	1656.2600000000002
	10312	601	1494.1900000000003
	10316	623	1375.59

❖ Bài tập thực hành

1. Đưa ra tên các thành phố và số lượng khách hàng tại từng thành phố.
2. Đưa ra số lượng các đơn đặt hàng trong tháng 3/2005. Đưa ra số đơn hàng của mỗi tháng trong năm 2005.
3. Đưa ra 10 mã đơn đặt hàng có giá trị lớn nhất.
4. Đưa ra mã nhóm hàng và tổng số lượng hàng hoá còn trong kho của nhóm hàng đó.
5. Đưa ra mã khách hàng và tổng số tiền khách đã đó trả.