Bài thực hành số 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:

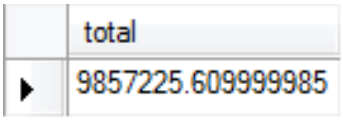
sum(quantityInStock)
555131

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:



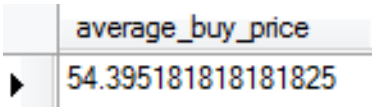
	total
▶	9857225.609999985

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

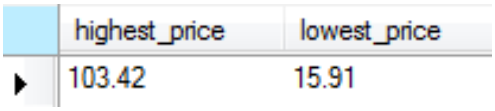
Hàm MAX trả về giá trị lớn nhất và hàm MIN trả về giá trị nhỏ nhất của một tập các giá trị.

```
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:



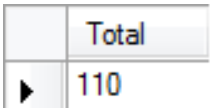
	highest_price	lowest_price
▶	103.42	15.91

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: 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	90157.77000000002
S10_1949	190017.95999999996
S10_2016	109998.81999999998
S10_4698	170685.99999999997
S10_4757	127924.31999999999
S10_4962	123123.00999999998
S12_1099	161531.47999999992
S12_1108	190755.86
S12_1666	119085.24999999999
S12_2823	135767.03000000003
S12_3148	132363.78999999998
S12_3380	98718.76000000001

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 status, count(*)
FROM orders
GROUP BY status DESC;
```

Kết quả trả về như sau:

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

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

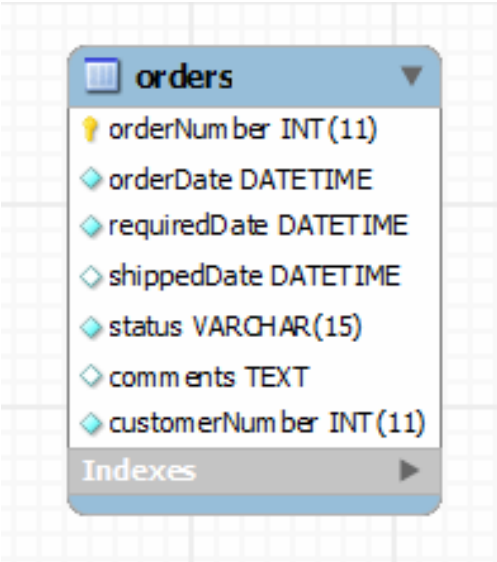
MySQL tuân theo chuẩn ANSI SQL. Tuy nhiên, có 2 sự khác biệt khi sử dụng GROUP BY trong MySQL như sau:

- 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 kiệ 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 di 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 itemsCount,
     sum(priceeach) AS total
FROM orderdetails
GROUP BY ordernumber
```

Kết quả trả về như sau:

ordernumber	itemsCount	total
10100	151	301.84000000000003
10101	142	352
10102	80	138.68
10103	541	1520.3699999999997
10104	443	1251.8899999999999
10105	545	1479.71
10106	675	1427.2800000000002
10107	229	793.2099999999999
10108	561	1432.86
10109	212	700.89
10110	570	1338.4699999999998
10111	217	460.15999999999997
10112	52	282.26
10113	143	325.22999999999996

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 itemsCount,
     sum(priceeach) AS total
FROM orderdetails
GROUP BY ordernumber
HAVING total > 1000
```

ordernumber	itemsCount	total
10103	541	1520.3699999999997
10104	443	1251.8899999999999
10105	545	1479.71
10106	675	1427.2800000000002
10108	561	1432.86
10110	570	1338.4699999999998
10117	402	1307.4700000000003
10119	442	1081.4400000000003
10120	525	1322.67
10122	545	1598.2699999999995
10126	617	1623.71
10127	540	1601.3899999999999
10135	607	1494.86
10140	385	1093.98

Chúng ta sử dụng bí danh cho cột sum *(priceeach)* 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 itemsCount,
```

```
    sum(priceeach) AS total
FROM orderdetails
GROUP BY ordernumber
HAVING total > 1000 AND itemsCount > 600
```

Kết quả trả về như sau:

ordernumber	itemsCount	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 của chúng ta tại từng thành phố.
2. Đưa ra số lượng các đơn đặt hàng trong tháng 3/2005.
3. Đưa ra 10 đơn đặt hàng có giá trị lớn nhất trong tháng 3/2005.
4. Đưa ra mã nhóm hàng và tổng số lượng hàng hoá còn trong kho của nhóm hàng đó.