**Crosstab query**

Khi viết query để lấy dữ liệu, tính toán và hiển thị ta có thể gặp trường hợp phải chuyển dữ liệu từ dạng dòng sang dạng cột. Kiểu query như vậy được gọi là Crosstab-query.

Trong một số Hệ quản trị CSDL hiện nay có hỗ trợ lệnh trực tiếp thực hiện việc này, ví dụ SQL Server phiên bản 2005 trở lên có hỗ trợ lệnh PIVOT (tham khảo tại <http://msdn.microsoft.com/en-us/library/ms140308.aspx>). Trong khuôn khổ bài viết này, tôi sẽ trình bày một cách đơn giản và dễ hiểu nhất về kỹ thuật này, sử dụng các lệnh trung gian như CASE, CURSOR… và vòng lặp. Bởi không phải hệ quản trị CSDL nào cũng hỗ trợ lệnh PIVOT.

Tôi sẽ sử dụng database của w3schools, bảng Orders, Employees và bảng Shippers (các bạn có thể tham khảo tại <http://www.w3schools.com/sql/trysql.asp?filename=trysql_select_all>).

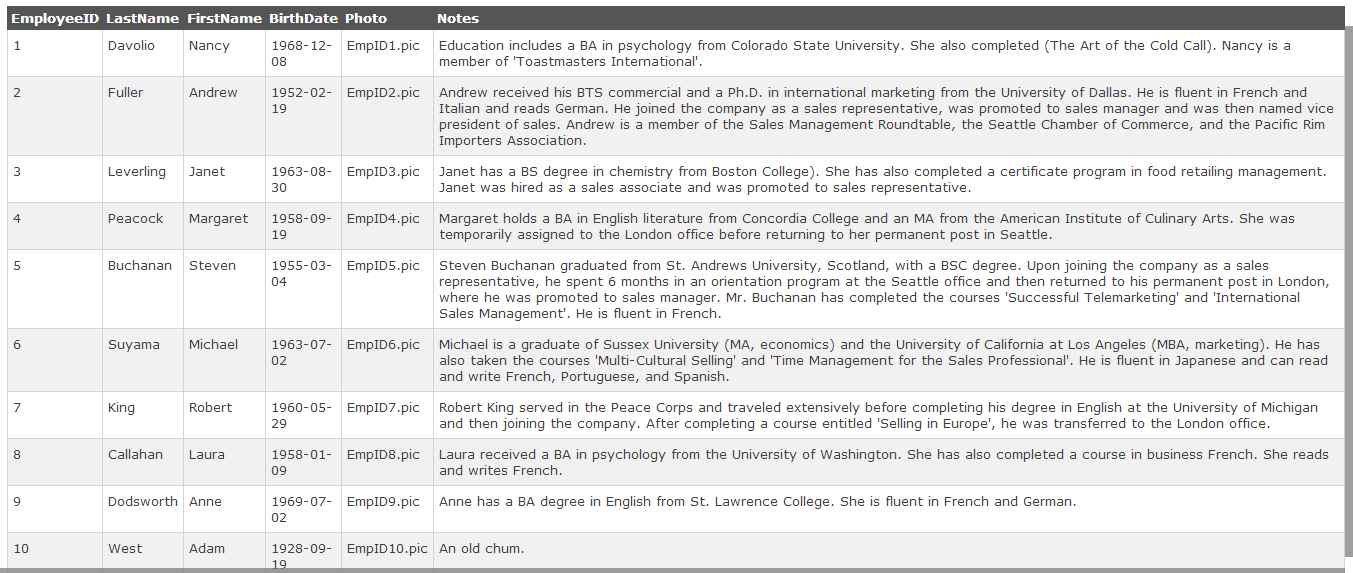
Bảng Orders:



* OrderID (PK): mã của đơn hàng (order), mỗi đơn hàng chỉ có 1 mã duy nhất.
* CustomerID: mã khách hàng
* EmployeeID: mã của nhân viên bán hàng, với mỗi đơn hàng thì có 1 và chỉ 1 nhân viên xuất ra nó.
* OrderDate: ngày xuất đơn hàng.
* ShipperID: mã của công ty vận chuyển hàng (shipper)

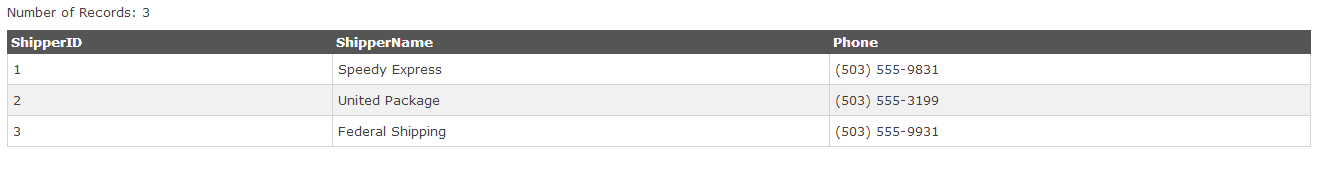
Trong ví dụ này chúng ta sẽ không quan tâm đến CustomerID và OrderDate.

Bảng Employees:



Chúng ta có 10 nhân viên tất cả với các thông tin như trên hình.

Bảng Shippers:



Có 3 hãng vận chuyển với tên và số điện thoại như trên hình.

Yêu cầu: Hãy thống kê số lượng đơn hàng theo từng nhân viên, và theo từng phương thức vận chuyển (shipper) khác nhau.

Với yêu cầu như trên, ta hoàn toàn có thể giải quyết dựa trên bảng Orders, với đầy đủ yêu cầu mà bài toán đặt ra. Câu truy vấn như sau:

SELECT EmployeeID,

ShipperID,

COUNT(\*) Total

FROM Orders

GROUP BY

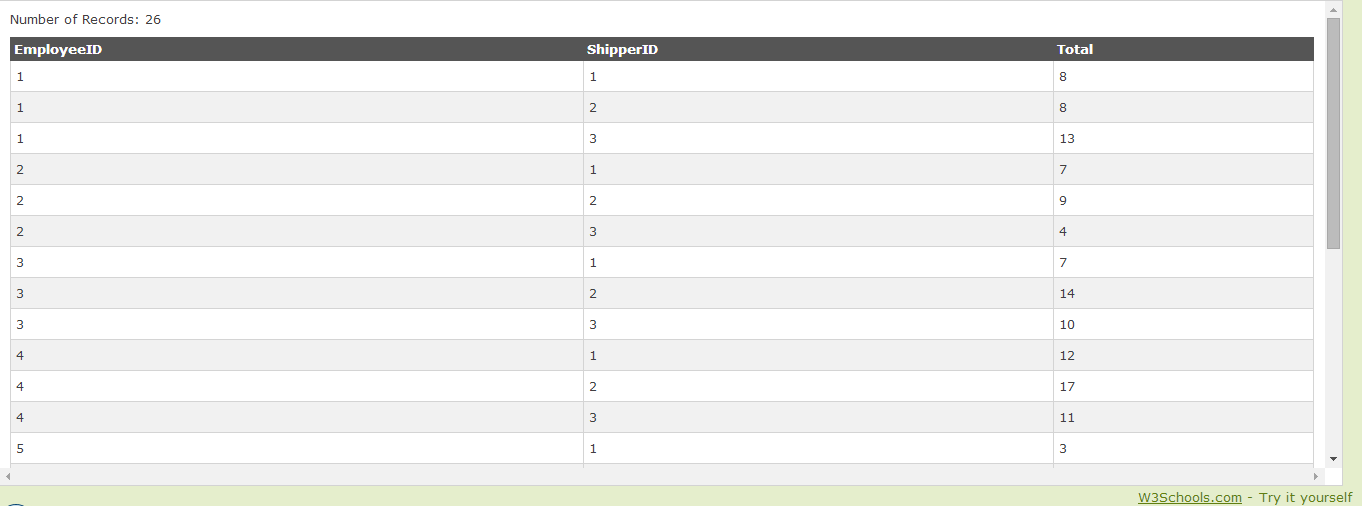
EmployeeID,

ShipperID

ORDER BY

EmployeeID

Kết quả:



Từ bảng trên, ta thấy được nhân viên có mã là 1 xuất tổng cộng 29 đơn hàng, trong đó có 8 đơn hàng được ship bằng hãng vận chuyển có mã là 1, 8 đơn hàng được ship bằng hãng vận chuyển có mã là 2, 13 đơn hàng được ship bằng hãng vận chuyển có mã là 3.

Kết quả như vậy đã đáp ứng được yêu cầu mà bài toán đặt ra. Tuy nhiên, có một vấn đề ở đây. Đó là ShipperID đang ở dạng dòng, tương đối khó nhìn (hãy tưởng tượng nếu có nhiều hơn 3 ShipperID ở đây).

Giờ chúng ta sẽ sử dụng Crosstab query để chuyển ShipperID sang dạng cột.

Câu truy vấn đơn giản như sau:

SELECT

EmployeeID,

COUNT(CASE ShipperID WHEN 1 THEN 1 END) 'Shipper 1',

COUNT(CASE ShipperID WHEN 2 THEN 1 END) 'Shipper 2',

COUNT(CASE ShipperID WHEN 3 THEN 1 END) 'Shipper 3',

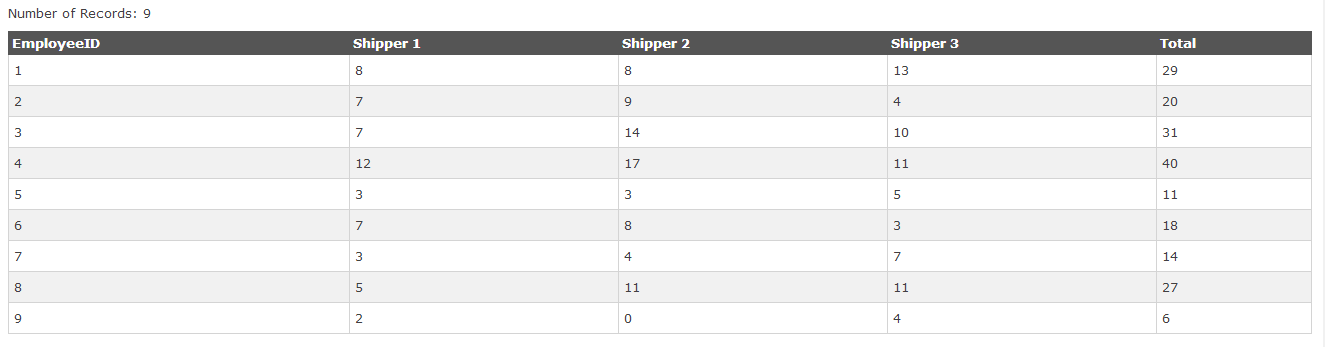
COUNT(\*) Total

FROM

Orders

GROUP BY EmployeeID

Kết quả:



Đã dễ nhìn hơn rất nhiều.

Câu truy vấn trên được thực hiện như sau:

* Ta thực hiện 1 GROUP BY theo từng nhân viên
* Tại mỗi group, ta đếm số lượng các đơn hàng theo từng loại vận chuyển
* COUNT(\*) Total dùng để tính tổng số đơn hàng mà mỗi nhân viên bán ra.

Trên thực tế, khi hiển thị dữ liệu cho người dùng, rất ít khi dữ liệu của chúng ta được hiển thị ở dạng mã như trên. Các alias Shipper 1, Shipper 2, Shipper 3 ở trên cũng không mang nhiều ý nghĩa. Thêm vào đó, không phải lúc nào cũng phải hiển thị tất cả các Shipper như trên (ví dụ ShipperID có trong điều kiện tìm kiếm chẳng hạn).

Chúng ta sẽ tiếp tục hoàn thiện câu truy vấn trên. Ở đây ta cần JOIN với các bảng Employees để có được của từng tên nhân viên. Thêm nữa, khi đưa vào điều kiện tìm kiếm (ShipperID), số lượng các cột sinh ra sẽ là động. Tôi sẽ sử dụng Stored procedure cho ví dụ này.

Trước hết, chúng ta tạo 1 procedure với input là shipperID:

CREATE PROCEDURE order\_list(IN input\_shipper\_id VARCHAR(1))

Tiếp tục khai báo 2 biến để lưu ShipperID và ShipperName (cur\_shipper\_id, cur\_shipper\_name). Chúng ta cũng cần 1 biến dạng CURSOR để lưu các thông tin lấy được trong bảng Shippers (thông tin về CURSOR các bạn có thể tham khảo tại <http://msdn.microsoft.com/en-us/library/ms181441.aspx>). Và 1 biến lưu trữ câu lệnh SQL là command\_text:

DECLARE cur\_shipper\_id VARCHAR(1);

DECLARE cur\_shipper\_name VARCHAR(20);

DECLARE command\_text VARCHAR(10000);

DECLARE shipperCur CURSOR FOR (SELECT ShipperID, ShipperName FROM Shippers WHERE CASE WHEN input\_shipper\_id IS NOT NULL THEN ShipperID = input\_shipper\_id ELSE 1 END);

Lưu ý ở đây input\_shipper\_id chính là điều kiện tìm kiếm. Có thể được chỉ định (not null) hoặc không (null).

Sau khi đã định nghĩa xong các biến cần dùng, chúng ta sẽ lấy ra trường đầu tiên - tên của nhân viên - theo cú pháp: employee\_name = FirstName + LastName (trong bảng Employees).

SET command\_text = select concat(concat(e.FirstName, '' ''), e.LastName) as employee\_name';

Với “e” là alias của bảng Employees (sẽ được định nghĩa ở dưới).

Tiếp theo, chúng ta tiến hành mở con trỏ đã định nghĩa ở trên và đưa vào vòng lặp, để lấy ra các cột (là ShipperName) và dữ liệu tương ứng của từng cột:

OPEN shipperCur;

SELECT COUNT(ShipperID) INTO @COUNTER FROM Shippers WHERE CASE WHEN input\_shipper\_id IS NOT NULL THEN ShipperID = input\_shipper\_id ELSE 1 END;

SET @INDEX = 0;

CUR\_LOOP: WHILE @COUNTER > @INDEX DO

FETCH shipperCur INTO cur\_shipper\_id, cur\_shipper\_name;

IF cur\_shipper\_id IS NOT NULL THEN

SET command\_text = CONCAT(command\_text, CONCAT(CONCAT(CONCAT(',count(case when s.ShipperID = ''', cur\_shipper\_id), ''' then 1 end) as '''), cur\_shipper\_name, ''''));

END IF;

SET @INDEX = @INDEX + 1;

END WHILE;

CLOSE shipperCur;

Cuối cùng chúng ta join Orders với Employees để phục vụ cho việc lấy employee\_name ở trên:

IF input\_shipper\_id IS NOT NULL THEN

SET command\_text = CONCAT(CONCAT(command\_text, ', count(case ShipperID when ', input\_shipper\_id, ' then 1 end) as Total from orders o inner join employees e on o.EmployeeID = e.EmployeeID group by o.EmployeeID'));

ELSE

SET command\_text = CONCAT(command\_text, ', count(\*) as Total from orders o inner join employees e on o.EmployeeID = e.EmployeeID group by o.EmployeeID');

END IF;

Như vậy là đã xong. Tuy nhiên, khi sử dụng con trỏ, chúng ta nên khai báo 1 điều khiển (HANDLER) NOT FOUND để tránh gặp lỗi “No data – zero rows fetched”. Lưu ý rằng điều khiển này phải được khai báo sau khai báo con trỏ, nếu không sẽ gây lỗi:

DECLARE done INT DEFAULT 0;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

Khi không có dữ liệu thỏa mãn, biến done sẽ có giá trị là 1, ta chỉ cần kiểm tra khi done = 1 thì sẽ nhảy khỏi vòng lặp:

IF done = 1 THEN

LEAVE CUR\_LOOP;

END IF;

Với CUR\_LOOP là tên vòng lặp đã định nghĩa ở trên.

Đoạn SQL động đầy đủ của chúng ta sẽ như sau (trong hệ QTCSDL MySQL):

delimiter $$

DROP PROCEDURE IF EXISTS order\_list$$

CREATE PROCEDURE order\_list(IN input\_shipper\_id VARCHAR(1))

BEGIN

DECLARE done INT DEFAULT 0;

DECLARE cur\_shipper\_id VARCHAR(1);

DECLARE cur\_shipper\_name VARCHAR(40);

DECLARE command\_text VARCHAR(10000);

DECLARE shipperCur CURSOR FOR (SELECT ShipperID, ShipperName FROM Shippers WHERE CASE WHEN input\_shipper\_id IS NOT NULL THEN ShipperID = input\_shipper\_id ELSE 1 END);

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

SET command\_text = 'select concat(e.FirstName, '' '', e.LastName) as employee\_name';

OPEN shipperCur;

SELECT COUNT(ShipperID) INTO @COUNTER FROM Shippers WHERE CASE WHEN input\_shipper\_id IS NOT NULL THEN ShipperID = input\_shipper\_id ELSE 1 END;

SET @INDEX = 0;

CUR\_LOOP: WHILE @COUNTER > @INDEX DO

IF done = 1 THEN

LEAVE cur\_loop;

END IF;

FETCH shipperCur INTO cur\_shipper\_id, cur\_shipper\_name;

IF cur\_shipper\_id IS NOT NULL THEN

SET command\_text = CONCAT(command\_text, CONCAT(CONCAT(CONCAT(',count(case o.ShipperID when ', cur\_shipper\_id), ' then 1 end) as '''), cur\_shipper\_name, ''''));

END IF;

SET @INDEX = @INDEX + 1;

END WHILE;

CLOSE shipperCur;

IF input\_shipper\_id IS NOT NULL THEN

SET command\_text = CONCAT(CONCAT(command\_text, ', count(case ShipperID when ', input\_shipper\_id, ' then 1 end) as Total from orders o inner join employees e on o.EmployeeID = e.EmployeeID group by o.EmployeeID'));

ELSE

SET command\_text = CONCAT(command\_text, ', count(\*) as Total from orders o inner join employees e on o.EmployeeID = e.EmployeeID group by o.EmployeeID');

END IF;

SET @SQL = command\_text;

PREPARE stmt FROM @SQL;

EXECUTE stmt;

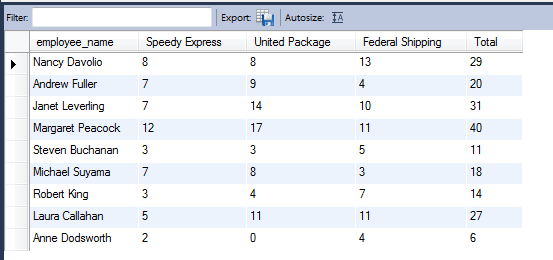
END

$$

delimiter ;

Chạy thử với trường hợp không đưa vào điều kiện tìm kiếm:

CALL order\_list(null);



Chạy thử với điều kiện tìm kiếm:

CALL order\_list(1);

