

# PIVOT và UNPIVOT (Phần 1)

13:09 POSTED BY NTP ANALYSIS , DATABASE NO COMMENTS

Trong quá trình lấy số liệu, thiết lập bảng biểu và báo cáo, một trường hợp thường gặp là bạn phải hiển thị dữ liệu từ dạng dòng chuyển thành cột - kiểu truy vấn như vậy gọi là Pivot Query và ngược lại nghĩa là chuyển từ cột thành dòng - kiểu truy vấn như vậy gọi là Unpivot Query.

Để hiểu rõ hơn về 2 khái niệm trên ta thực hành ví dụ sau.

## A. Khởi tạo dữ liệu

```
1 create table cust_sales
2 (
3     customer_id varchar2(10),
4     month varchar2(3),
5     product_id varchar2(6),
6     sales number
7 );
8
9 insert into cust_sales values('Adam', 'Jan', 'Prod A', 34);
10 insert into cust_sales values('Adam', 'Feb', 'Prod B', 49);
11 insert into cust_sales values('Adam', 'Mar', 'Prod C', 134);
12 insert into cust_sales values('Jones', 'Feb', 'Prod A', 92);
13 insert into cust_sales values('Jones', 'Mar', 'Prod A', 34);
14 insert into cust_sales values('Kanes', 'Jan', 'Prod B', 24);
15 insert into cust_sales values('Kanes', 'Feb', 'Prod C', 83);
16 insert into cust_sales values('Kanes', 'Mar', 'Prod A', 39);
17
18 create table cust_sales_month
19 (
20     customer_id varchar2(10),
21     product_id varchar2(6),
22     january number,
23     february number,
```

```
24      march number
25 );
26
27 insert into cust_sales_month values( 'Kanes' , 'Prod A' ,0,0,1);
28 insert into cust_sales_month values( 'Jones' , 'Prod A' ,0,1,1);
29 insert into cust_sales_month values( 'Kanes' , 'Prod C' ,0,1,0);
30 insert into cust_sales_month values( 'Kanes' , 'Prod B' ,1,0,0);
31 insert into cust_sales_month values( 'Adam' , 'Prod B' ,0,1,0);
32 insert into cust_sales_month values( 'Adam' , 'Prod A' ,1,0,0);
33 insert into cust_sales_month values( 'Adam' , 'Prod C' ,0,0,1);
34
35 commit;
```

(Nhóm lệnh 1)

Chúng ta chú ý như sau:

- Bảng cust\_sales: dùng cho việc thực hành các vấn đề liên quan đến PIVOT
- Bảng cust\_sales\_month: dùng cho việc thực hành các vấn đề liên quan đến UNPIVOT

### **B. Các vấn đề liên quan đến PIVOT:**

Ta chạy câu lệnh sau:

```
1 SELECT * FROM cust_sales;
```

(Câu lệnh 1)

Kết quả:

	A	B	C	D
1	CUSTOMER_ID	MONTH	PRODUCT_ID	SALES
2	Adam	Jan	Prod A	34
3	Adam	Feb	Prod B	49
4	Adam	Mar	Prod C	134
5	Jones	Feb	Prod A	92
6	Jones	Mar	Prod A	34
7	Kanes	Jan	Prod B	24
8	Kanes	Feb	Prod C	83
9	Kanes	Mar	Prod A	39

(Hình 1)

Với kết quả trên thì việc thống kê rất khó để theo dõi cho các yêu cầu sau:

- Yêu cầu 1: Thống kê kết quả bán hàng theo sản phẩm

	A	B	C	D	E
1	CUSTOMER_ID	MONTH	PRODUCT_A	PRODUCT_B	PRODUCT_C
2	Jones	Feb	92		
3	Adam	Jan	34		
4	Jones	Mar	34		
5	Adam	Mar			134
6	Adam	Feb		49	
7	Kanes	Mar	39		
8	Kanes	Jan		24	
9	Kanes	Feb			83

(Hình 2)

- Yêu cầu 2: Thống kê kết quả bán hàng theo tháng

	A	B	C	D	E
1	<b>CUSTOMER_ID</b>	<b>PRODUCT_ID</b>	<b>JANUARY</b>	<b>FEBRUARY</b>	<b>MARCH</b>
2	Kanes	Prod A			39
3	Jones	Prod A		92	34
4	Kanes	Prod C		83	
5	Kanes	Prod B	24		
6	Adam	Prod B		49	
7	Adam	Prod A	34		
8	Adam	Prod C			134

(Hình 3)

- Yêu cầu 3: Thống kê kết quả bán hàng theo khách hàng

	A	B	C	D	E
1	<b>MONTH</b>	<b>PRODUCT_ID</b>	<b>'Adam'</b>	<b>'Jones'</b>	<b>'Kanes'</b>
2	Jan	Prod A	34		
3	Jan	Prod B			24
4	Feb	Prod A		92	
5	Feb	Prod B	49		
6	Feb	Prod C			83
7	Mar	Prod A		34	39
8	Mar	Prod C	134		

(Hình 4)

Để thực hiện được các yêu cầu thống kê như trên ta có thể dùng CASE để làm (hàm này không được giới thiệu trong phạm vi bài này) hoặc dùng hàm PIVOT để thực hiện (lưu ý: hàm PIVOT chỉ có từ phiên bản Oracle 11g trở lên).

Cú pháp hàm như sau:

**SELECT...**

**FROM ...**

**PIVOT[XML]**

( pivot\_clause

  pivot\_for\_clause

  pivot\_in\_clause )

## WHERE ...

Trong đó:

- pivot\_clause: khai báo cột được tính toán bởi các hàm thống kê như SUM, COUNT, MAX, MIN, AVG, ...
- pivot\_for\_clause: khai báo cột sẽ được nhóm lại khi thực hiện thống kê bởi hàm được dùng trong pivot\_clause
- pivot\_in\_clause: khai báo các giá trị thỏa mãn điều kiện để thực hiện pivot

Để hiểu rõ hơn về các mệnh đề được dùng trong PIVOT ta map các nội dung của yêu cầu 1 vào các mệnh đề PIVOT như sau:

Yêu cầu 1: Thống kê kết quả bán hàng theo sản phẩm

- pivot\_clause: thống kê kết quả bán hàng --> sum(sales)
- pivot\_for\_clause: theo sản phẩm --> product\_id
- pivot\_in\_clause: kết quả hiển thị gồm các sản phẩm Prod A, Prod B, Prod C ở 3 cột --> ('Prod A', 'Prod B', 'Prod C')

Ta phân tích tương tự cho các yêu cầu 2 và 3 còn lại. Và đây là Pivot Query cho 3 yêu cầu trên:

- Yêu cầu 1: Thống kê kết quả bán hàng theo sản phẩm

```
1 SELECT *
2 FROM cust_sales PIVOT (SUM(sales)
3     FOR product_id
4     IN ('Prod A' AS product_a,
5        'Prod B' AS product_b,
6        'Prod C' AS product_c));
```

(Câu lệnh 2)

- Yêu cầu 2: Thống kê kết quả bán hàng theo tháng

```
1 SELECT *
2 FROM cust_sales PIVOT (SUM (sales)
3     FOR month
4     IN ('Jan' AS january, 'Feb' AS february, 'Mar' AS march));
```

(Câu lệnh 3)

- Yêu cầu 3: Thống kê kết quả bán hàng theo khách hàng

```
1 SELECT *
```

```

2 FROM cust_sales PIVOT (SUM(sales)
3     FOR customer_id
4     IN('Adam', 'Jones', 'Kanes'));

```

(Câu lệnh 4)

### C. Các vấn đề liên quan đến UNPIVOT:

Unpivot là trường hợp ngược lại của PIVOT nghĩa là khi ta có 1 tập kết quả đang được hiển thị dưới dạng cột ta muốn biến chuyển kết quả này hiển thị dưới dạng dòng. Xem ví dụ sau:

```
1 SELECT * FROM cust_sales_month;
```

(Câu lệnh 5)

	A	B	C	D	E
1	CUSTOMER_ID	PRODUCT_ID	JANUARY	FEBRUARY	MARCH
2	Kanes	Prod A	0	0	1
3	Jones	Prod A	0	1	1
4	Kanes	Prod C	0	1	0
5	Kanes	Prod B	1	0	0
6	Adam	Prod B	0	1	0
7	Adam	Prod A	1	0	0
8	Adam	Prod C	0	0	1

(Hình 5)

Bảng dữ liệu trên cho biết các khách hàng có mua sản phẩm vào các tháng nào trong năm. Nếu như tháng nào mua thì đánh dấu là 1 ngược lại là 0. Với bảng dữ liệu trên ta muốn tùy biến hiển thị dữ liệu theo dạng dòng như sau:

CUSTOMER_ID	PRODUCT_ID	MONTH	YES
Adam	Prod A	JANUARY	1
Adam	Prod A	FEBRUARY	0
Adam	Prod A	MARCH	0
Adam	Prod B	FEBRUARY	1
Adam	Prod B	JANUARY	0
Adam	Prod B	MARCH	0
Adam	Prod C	FEBRUARY	0
Adam	Prod C	JANUARY	0

Adam	Prod C	MARCH	1
Jones	Prod A	FEBRUARY	1
Jones	Prod A	JANUARY	0
Jones	Prod A	MARCH	1
Kanes	Prod A	MARCH	1
Kanes	Prod A	JANUARY	0
Kanes	Prod A	FEBRUARY	0
Kanes	Prod B	FEBRUARY	0
Kanes	Prod B	JANUARY	1
Kanes	Prod B	MARCH	0
Kanes	Prod C	MARCH	0
Kanes	Prod C	FEBRUARY	1
Kanes	Prod C	JANUARY	0

(Bảng kết quả 1)

Trước khi đi đến cách thực hiện, ta tìm hiểu sơ qua cú pháp của hàm UNPIVOT.

```

1 UNPIVOT
2 ( unpivot_clause
3  unpivot_for_clause
4  unpivot_in_clause )

```

Trong đó:

- **unpivot\_clause**: khác với PIVOT, mệnh đề được khai báo ở đây không thực hiện tính toán; nó chỉ đơn thuần là tên cột dùng để hiển thị dữ liệu sau khi unpivot
- **unpivot\_for\_clause**: khai báo cột cần unpivot để lấy giá trị tương ứng với tên cột được khai báo ở unpivot\_clause
- **unpivot\_in\_clause**: khai báo danh sách các giá trị khi thực hiện UNPIVOT

Để hiểu rõ hơn cách sử dụng hàm UNPIVOT, ta xét yêu cầu của bảng kết quả 4: Thống kê danh sách khách hàng có sản phẩm nào được mua / không mua theo từng tháng. Ta thực hiện map các nội dung trên vào các mệnh đề unpivot:

- **unpivot\_clause**: mua / không mua --> tên cột chứa giá trị

- **unpivot\_for\_clause**: theo từng tháng --> giá trị cột month
- **unpivot\_in\_clause**: theo từng tháng --> ở đây cần thống kê 3 tháng: JANUARY, FEBRUARY và MARCH

Và đây là kết quả:

```
1 SELECT *
2   FROM cust_sales_month UNPIVOT (yes
3                                   FOR month
4                                   IN ("JANUARY", "FEBRUARY", "MARCH"))
5 ORDER BY customer_id, product_id;
```

(Câu lệnh 6)