

Bài thực hành tuần 5

Các hàm xử lý của MySQL

❖ **Nội dung chính:** Trong bài này, chúng ta sẽ làm quen với một số hàm (functions) cơ bản:

- Hàm xử lý chuỗi ký tự: Substring, Concat, Replace
- Hàm điều kiện If
- Hàm LAST_INSERT_ID
- Hàm xử lý thời gian: DATEDIFF, ADDDATE, EXTRACT

1. Hàm xử lý chuỗi SUBSTRING

Hàm Substring cho phép trích xuất một chuỗi con từ một chuỗi khác, bắt đầu tại vị trí cụ thể và với một độ dài nhất định. Sau đây minh họa các hình thức sử dụng khác nhau của hàm này.

```
SUBSTRING(str,pos);  
SUBSTRING(str FROM pos);
```

Kết quả của câu lệnh ở trên trả về một chuỗi con từ một chuỗi **str** bắt đầu từ vị trí **pos**

```
SUBSTRING(str,pos,len);  
SUBSTRING(str FROM pos FOR len);
```

Hai câu lệnh ở trên trả về một chuỗi con từ một chuỗi **str**, bắt đầu tại vị trí **pos** và chuỗi con trả về chỉ có **len** ký tự. Lưu ý rằng FROM là từ khoá cú pháp SQL chuẩn. Chúng ta hãy xem xét một số ví dụ sau”

```
SELECT substring('MySQL Substring',7);
```

Trả về: Substring

```
SELECT substring('MySQL Substring' FROM 7);
```

Trả về: Substring

```
SELECT substring('MySQL Substring',7,3);
```

Trả về: Sub

```
SELECT substring('MySQL Substring' FROM 7 FOR 3);
```

Trả về: Sub

cũng có thể sử dụng giá trị âm cho tham số pos. Nếu sử dụng giá trị âm cho tham số pos, sự bắt đầu của chuỗi con được tính từ cuối của chuỗi, ví dụ

```
SELECT substring('MySQL Substring',-9);
```

Trả về: Substring

Đôi khi thấy đoạn mã sử dụng *substr ()* thay vì hàm *substring ()*. *Substr* là từ đồng nghĩa với *substring*, vì vậy nó có tác dụng tương tự.

2. Hàm CONCAT

Hàm Concat được sử dụng để nối hai hoặc nhiều chuỗi. Nếu các đối số là số, chúng sẽ được chuyển đổi thành chuỗi trước khi nối. Nếu bất kỳ đối số trong danh sách đối số là NULL, hàm concat sẽ trả về NULL.

```
CONCAT(str1,str2,...)
```

Ví dụ: Để hiển thị tên đầy đủ đầu tiên của địa chỉ liên lạc của khách hàng chúng tôi sử dụng hàm concat để nối các tên đầu tiên và tên cuối cùng và dấu phân cách giữa chúng. Dưới đây là truy vấn:

```
SELECT CONCAT(contactLastname,', ',contactFirstname)
fullname
FROM customers
```

	fullname
►	Schmitt, Carine
	King, Jean
	Ferguson, Peter
	Labrune, Janine
	Bergulfsen, Jonas
	Nelson, Susan
	Piestrzeniewicz, Zbyszek
	Keitel, Roland
	Murphy, Julie
	Lee, Kwai
	Freyre, Diego
	Berglund, Christina
	Petersen, Jytte
	Saveley, Mary
	Natividad, Eric
	Young, Jeff

MySQL cũng hỗ trợ hàm `concat_ws` cho phép chúng ta nối hai hay nhiều hơn hai chuỗi với một dấu phân cách được xác định trước. Cú pháp của hàm `concat_ws` là:

```
CONCAT_WS(seperator, str1, str2, ...)
```

Tham số đầu tiên là dấu phân cách do định nghĩa và sau đó là những chuỗi muốn nối. Kết quả trả về là một chuỗi đã được ghép nối, với dấu phân cách giữa mỗi thành phần ghép nối. Ví dụ, có thể đạt được kết quả tương tự trong ví dụ trên bằng cách sử dụng `concat_ws` thay vì hàm `concat`.

```
SELECT CONCAT_WS('; ', contactLastname, contactFirstname)
       fullname
FROM customers
```

	fullname
►	Schmitt; Carine
	King; Jean
	Ferguson; Peter
	Labrune; Janine
	Bergulfsen; Jonas
	Nelson; Susan
	Piestrzeniewicz; Zbyszek
	Keitel; Roland
	Murphy; Julie
	Lee; Kwai
	Freyre; Diego
	Berglund; Christina
	Petersen; Jytte
	Saveley; Mary
	Natividad; Eric
	Young; Jeff

Dưới đây là một ví dụ khác của việc sử dụng concat_ws để có được định dạng địa chỉ của khách hàng.

```
SELECT CONCAT_WS(char(10),
                CONCAT_WS(' ',contactLastname,contactFirstname),
addressLine1, addressLine2,
                CONCAT_WS(' ',postalCode,city), country,
                CONCAT_WS(char(10),'')) AS Customer_Address
FROM customers
```

	Customer_Address
►	Schmitt Carine 54, rue Royale44000 NantesFrance
	King Jean8489 Strong St.83030 Las VegasUSA
	Ferguson Peter636 St Kilda RoadLevel 33004 MelbourneAustralia
	Labrune Janine 67, rue des Cinquante Otages44000 NantesFrance
	Bergulfsen Jonas Erling Skakkes gate 784110 StavemNorway
	Nelson Susan5677 Strong St.97562 San RafaelUSA
	Piestrzeniewicz Zbyszek ul. Filtrowa 6801-012 WarszawaPoland
	Keitel RolandLyonerstr. 3460528 FrankfurtGermany
	Murphy Julie5557 North Pendale Street94217 San FranciscoUSA
	Lee Kwai897 Long Airport Avenue10022 NYCUSA
	Freyre Diego C/ Moralarzal, 8628034 MadridSpain
	Berglund Christina Berguvsvägen 8S-958 22 LuleåSweden
	Petersen Jytte Vinbæltet 341734 KobenhavnDenmark
	Saveley Mary 2, rue du Commerce69004 LyonFrance
	Natividad EricBronz Sok.Bronz Apt. 3/6 Tesvikiye079903 SingaporeSingapore
	Young Jeff4092 Furth CircleSuite 40010022 NYCUSA

3. Hàm REPLACE

MySQL cung cấp cho một hàm xử lý chuỗi hữu ích là Replace, cho phép thay thế một chuỗi trong một cột của một bảng bằng một chuỗi mới.

Cú pháp của hàm như sau:

```
UPDATE <tên bảng>
SET tên cột = REPLACE(tên cột,xâu cần tìm,xâu thay thế)
WHERE <các điều kiện>
```

Lưu ý: rằng khi tìm kiếm các văn bản để thay thế, MySQL có phân biệt chữ hoa và chữ thường.

Ví dụ: nếu muốn sửa lỗi chính tả trong bảng Product trong cơ sở dữ liệu mẫu, sử dụng hàm Replace như sau:

```
UPDATE products
SET productDescription =
REPLACE(productDescription,'abuot','about')
```

Truy vấn sẽ xem xét cột productDescription và tìm thấy tất cả các lần xuất hiện của lỗi chính tả 'abuoat' và thay thế nó bằng từ chính xác 'about'.

Điều rất quan trọng cần lưu ý rằng trong hàm Replace, tham số đầu tiên là tên trường không đặt trong dấu ‘’. Nếu đặt dấu để tên trường như 'field_name', truy vấn sẽ cập nhật nội dung của cột 'field_name', gây mất dữ liệu.

Hiện nay hàm Replace không hỗ trợ biểu thức chính quy vì vậy nếu cần phải thay thế một chuỗi văn bản bằng một mẫu, cần phải sử dụng hàm do người dùng định nghĩa (UDF) từ thư viện bên ngoài.

4. Hàm IF

IF là một hàm điều khiển, trả về kết quả là một chuỗi hoặc số dựa trên một điều kiện cho trước. Cú pháp của hàm IF như sau:

```
IF(expr, if_true_expr, if_false_expr)
```

- Tham số đầu tiên là expr sẽ được kiểm tra là đúng hay sai. Giá trị thực có nghĩa là expr không bằng 0 và expr không bằng NULL. Lưu ý rằng NULL là một giá trị đặc biệt, không bằng bất cứ điều gì khác, ngay cả bản thân nó.
- Nếu expr được đánh giá là đúng, hàm IF sẽ trả lại if_true_expr, nếu không nó sẽ trả lại if_false_expr.

Ví dụ:

```
SELECT IF(1 = 2, 'true', 'false');
```

Trả về: false

```
SELECT IF(1 = 1, ' true', 'false');
```

Trả về: true

Ví dụ: Trong bảng khách hàng, không phải tất cả các khách hàng đều có thông tin về state. Vì vậy, khi chúng ta lựa chọn khách hàng, thông tin state sẽ hiển thị giá trị NULL, không có ý nghĩa cho mục đích báo cáo.

```
SELECT customerNumber,  
       customerName,
```

```

        state,
        country
FROM customers;

```

	customerNumber	customerName	state	country
▶	103	Atelier graphique	NULL	France
	112	Signal Gift Stores	NV	USA
	114	Australian Collectors, Co.	Victoria	Australia
	119	La Rochelle Gifts	NULL	France
	121	Baane Mini Imports	NULL	Norway
	124	Mini Gifts Distributors Ltd.	CA	USA
	125	Havel & Zbyszek Co	NULL	Poland
	128	Blauer See Auto, Co.	NULL	Germany
	129	Mini Wheels Co.	CA	USA
	131	Land of Toys Inc.	NY	USA
	141	Euro+ Shopping Channel	NULL	Spain
	144	Volvo Model Replicas, Co	NULL	Sweden
	145	Danish Wholesale Imports	NULL	Denmark
	146	Saveley & Henriot, Co.	NULL	France
	148	Dragon Souvenirs, Ltd.	NULL	Singapore
	151	Muscle Machine Inc.	NY	USA

Chúng ta có thể sử dụng IF để hiển thị trạng thái của khách hàng là N / A nếu nó là NULL như sau:

```

SELECT customerNumber,
        customerName,
        IF(state IS NULL, 'N/A', state) state,
        country
FROM customers;

```

	customerNumber	customerName	state	country
▶	103	Atelier graphique	N/A	France
	112	Signal Gift Stores	NV	USA
	114	Australian Collectors, Co.	Victoria	Australia
	119	La Rochelle Gifts	N/A	France
	121	Baane Mini Imports	N/A	Norway
	124	Mini Gifts Distributors Ltd.	CA	USA
	125	Havel & Zbyszek Co	N/A	Poland
	128	Blauer See Auto, Co.	N/A	Germany
	129	Mini Wheels Co.	CA	USA
	131	Land of Toys Inc.	NY	USA
	141	Euro+ Shopping Channel	N/A	Spain
	144	Volvo Model Replicas, Co	N/A	Sweden
	145	Danish Wholesale Imports	N/A	Denmark
	146	Saveley & Henriot, Co.	N/A	France
	148	Dragon Souvenirs, Ltd.	N/A	Singapore

Ví dụ: Hàm IF cũng rất hữu ích với chức năng tổng hợp. Giả sử nếu muốn biết có bao nhiêu đơn đặt hàng đã vận chuyển và huỷ bỏ cùng một lúc, chúng ta có thể sử dụng IF để đếm như sau:

```
SELECT SUM(IF(status = 'Shipped',1,0))    AS Shipped,
       SUM(IF(status = 'Cancelled',1,0)) AS Cancelled
FROM orders;
```

	Shipped	Cancelled
▶	303	6

Trong truy vấn trên, nếu tình trạng của đơn đặt hàng là SHIPPED hoặc CANCELLED, IF sẽ trả lại giá trị 1, nếu không nó trả về 0. Và sau đó hàm SUM sẽ tính toán tổng số để vận chuyển và bị hủy bỏ dựa trên giá trị trả về của hàm IF.

5. Hàm LAST_INSERT_ID

Hàm LAST_INSERT_ID trả về ID của bản ghi cuối cùng được chèn vào bảng, với điều kiện đó là ID của cột có thuộc tính AUTO_INCREMENT. Trong thiết kế cơ sở dữ liệu, thường sử dụng một cột tự động tăng AUTO_INCREMENT. Khi chèn một bản ghi mới

vào bảng có cột AUTO_INCREMENT, MySQL tạo ra ID cho tự động dựa trên các thiết lập của cột đó. có thể có được ID này bằng cách sử dụng hàm LAST_INSERT_ID.

Ví dụ: tạo ra một bảng mới để thử nghiệm được gọi là TBL. Trong bảng TBL, chúng ta sử dụng ID là cột AUTO_INCREMENT.

```
CREATE TABLE tbl(  
    id INT AUTO_INCREMENT NOT NULL PRIMARY KEY,  
    description varchar(250) NOT NULL  
);
```

Sau đó, chúng ta sử dụng hàm LAST_INSERT_ID () để có được ID mới chèn.

```
INSERT INTO tbl(description)  
VALUES('MySQL last_insert_id');
```

Thực hiện truy vấn:

```
SELECT LAST_INSERT_ID();
```

LAST_INSERT_ID()
1

Điều quan trọng cần lưu ý rằng nếu chèn nhiều bản ghi vào bảng bằng cách sử dụng câu lệnh INSERT duy nhất, hàm LAST_INSERT_ID sẽ trả lại giá trị tạo ra cho các bản ghi chèn vào đầu tiên. Hãy thử các bước sau:

```
INSERT INTO tbl(description)  
VALUES('record 1'),  
    ('record 2'),  
    ('record 3');
```

Thực hiện truy vấn:

```
SELECT LAST_INSERT_ID();
```

	LAST_INSERT_ID()
▶	2

Chúng ta đã chèn 3 bản ghi bằng cách sử dụng câu lệnh INSERT và hàm LAST_INSERT_ID trả lại ID của bản ghi đầu tiên như mong muốn. MySQL LAST_INSERT_ID hoạt động dựa trên nguyên tắc độc lập với client. Nó có nghĩa là giá trị được trả về bởi hàm LAST_INSERT_ID cho một client cụ thể là giá trị mà client đó tạo ra. Điều này đảm bảo rằng mỗi client có thể nhận được ID riêng của mình mà không cần phải quan tâm đến các hoạt động của các client khác và không cần sử dụng cơ chế lock hay transaction (sẽ học sau).

6. Hàm DATEDIFF

Trong một số trường hợp, cần phải tính toán số ngày giữa hai mốc thời gian, ví dụ số ngày từ ngày vận chuyển và ngày yêu cầu trong một đơn đặt hàng. Trong những trường hợp này, cần phải sử dụng hàm DATEDIFF.

Cú pháp DATEDIFF như sau:

DATEDIFF(expr1, expr2)

expr1 và *expr2* là hai mốc thời gian.

Ví dụ:

```
SELECT DATEDIFF('2011-08-17', '2011-08-17');
```

Trả về: 0 day

```
SELECT DATEDIFF('2011-08-17', '2011-08-08');
```

Trả về: 9 days

```
SELECT DATEDIFF('2011-08-08', '2011-08-17');
```

Trả về: -9 days

Ví dụ: Để tính toán số ngày còn lại giữa ngày vận chuyển và ngày yêu cầu để trong đơn đặt hàng, chúng ta sử dụng DATEDIFF như sau:

```
SELECT orderNumber,  
DATEDIFF(requiredDate, shippedDate) AS daysLeft
```

```
FROM orders
ORDER BY daysLeft DESC;
```

	orderNumber	daysLeft
▶	10409	11
	10410	10
	10419	9
	10398	9
	10299	9
	10377	9
	10302	9
	10314	9
	10315	9
	10371	9

7. Hàm ADDDATE, EXTRACT

MySQL cũng hỗ trợ một số hàm xử lý ngày tháng khác như: ADDDATE, EXTRACT

Hàm ADDDATE: trả về một giá trị thời gian là kết quả của thao tác trên một giá trị thời gian khác.

Ví dụ: đưa ra ngày tháng sau ngày giờ hiện tại 30 ngày:

```
SELECT ADDDATE(NOW(), INTERVAL 30 DAY);
```

	ADDDATE(NOW(), INTERVAL 30 DAY)
▶	2012-06-24 08:14:35

Sử dụng từ khóa DAY để chỉ giá trị sẽ cộng vào là ngày. Tương tự ta có thể sử dụng một số từ khóa sau:

Giá trị	Định dạng đầu vào
MICROSECOND	MICROSECONDS
SECOND	SECONDS
MINUTE	MINUTES
HOURL	HOURS
DAY	DAYS
WEEK	WEEKS

Giá trị	Định dạng đầu vào
MONTH	MONTHS
QUARTER	QUARTERS
YEAR	YEARS
SECOND_MICROSECOND	'SECONDS.MICROSECONDS '
MINUTE_MICROSECOND	'MINUTES:SECONDS.MICROSECONDS '
MINUTE_SECOND	'MINUTES:SECONDS '
HOURL_MICROSECOND	'HOURS:MINUTES:SECONDS.MICROSECONDS '
HOURL_SECOND	'HOURS:MINUTES:SECONDS '
HOURL_MINUTE	'HOURS:MINUTES '
DAY_MICROSECOND	'DAYS HOURS:MINUTES:SECONDS.MICROSECONDS '
DAY_SECOND	'DAYS HOURS:MINUTES:SECONDS '
DAY_MINUTE	'DAYS HOURS:MINUTES '
DAY_HOURL	'DAYS HOURS '
YEAR_MONTH	'YEARS-MONTHS '

Ví dụ: đưa ra các đơn đặt hàng trong khoảng 30 ngày tính từ ngày 1/5/2005

```
SELECT *
FROM orders
WHERE orderDate >= '2005-5-1' AND orderDate < ADDDATE('2005-
5-1', INTERVAL 30 DAY);
```

Kết quả truy vấn:

	orderNumber	orderDate	requiredDate	shippedDate	status	comments
▶	10411	2005-05-01 00:00:00	2005-05-08 00:00:00	2005-05-06 00:00:00	Shipped	NULL
	10412	2005-05-03 00:00:00	2005-05-13 00:00:00	2005-05-05 00:00:00	Shipped	NULL
	10413	2005-05-05 00:00:00	2005-05-14 00:00:00	2005-05-09 00:00:00	Shipped	Customer requested that DHL is used for this shipping
	10414	2005-05-06 00:00:00	2005-05-13 00:00:00	NULL	On Hold	Customer credit limit exceeded. Will ship when a payment is received.
	10415	2005-05-09 00:00:00	2005-05-20 00:00:00	2005-05-12 00:00:00	Disputed	Customer claims the scales of the models don't match what was discuss
	10416	2005-05-10 00:00:00	2005-05-16 00:00:00	2005-05-14 00:00:00	Shipped	NULL
	10417	2005-05-13 00:00:00	2005-05-19 00:00:00	2005-05-19 00:00:00	Disputed	Customer doesn't like the colors and precision of the models.
	10418	2005-05-16 00:00:00	2005-05-24 00:00:00	2005-05-20 00:00:00	Shipped	NULL
	10419	2005-05-17 00:00:00	2005-05-28 00:00:00	2005-05-19 00:00:00	Shipped	NULL
	10420	2005-05-29 00:00:00	2005-06-07 00:00:00	NULL	In Process	NULL
	10421	2005-05-29 00:00:00	2005-06-06 00:00:00	NULL	In Process	Custom shipping instructions were sent to warehouse

Ví dụ: đưa ra các đơn đặt hàng tính từ trước ngày 1/5/2005, 30 ngày đến ngày 1/5/2005

```
SELECT *
FROM orders
```

```
WHERE orderDate<= '2005-5-1' AND orderDate > ADDDATE('2005-5-1', INTERVAL -30 DAY);
```

	orderNumber	orderDate	requiredDate	shippedDate	status	comments
▶	10401	2005-04-03 00:00:00	2005-04-14 00:00:00	NULL	On Hold	Customer credit limit exceeded. Will ship when a payment is received.
	10402	2005-04-07 00:00:00	2005-04-14 00:00:00	2005-04-12 00:00:00	Shipped	NULL
	10403	2005-04-08 00:00:00	2005-04-18 00:00:00	2005-04-11 00:00:00	Shipped	NULL
	10404	2005-04-08 00:00:00	2005-04-14 00:00:00	2005-04-11 00:00:00	Shipped	NULL
	10405	2005-04-14 00:00:00	2005-04-24 00:00:00	2005-04-20 00:00:00	Shipped	NULL
	10406	2005-04-15 00:00:00	2005-04-25 00:00:00	2005-04-21 00:00:00	Disputed	Customer claims container with shipment was damaged during shipment.
	10407	2005-04-22 00:00:00	2005-05-04 00:00:00	NULL	On Hold	Customer credit limit exceeded. Will ship when a payment is received.
	10408	2005-04-22 00:00:00	2005-04-29 00:00:00	2005-04-27 00:00:00	Shipped	NULL
	10409	2005-04-23 00:00:00	2005-05-05 00:00:00	2005-04-24 00:00:00	Shipped	NULL
	10410	2005-04-29 00:00:00	2005-05-10 00:00:00	2005-04-30 00:00:00	Shipped	NULL
	10411	2005-05-01 00:00:00	2005-05-08 00:00:00	2005-05-06 00:00:00	Shipped	NULL

Nếu thời gian cộng vào là tháng, năm thì từ khóa tương ứng được sử dụng là MONTH, YEAR.

Ví dụ trên có thể viết lại như sau

```
SELECT *
FROM orders
WHERE orderDate<= '2005-5-1' AND orderDate > ADDDATE('2005-5-1', INTERVAL -1 MONTH);
```

	orderNumber	orderDate	requiredDate	shippedDate	status	comments
▶	10401	2005-04-03 00:00:00	2005-04-14 00:00:00	NULL	On Hold	Customer credit limit exceeded. Will ship when a payment is received.
	10402	2005-04-07 00:00:00	2005-04-14 00:00:00	2005-04-12 00:00:00	Shipped	NULL
	10403	2005-04-08 00:00:00	2005-04-18 00:00:00	2005-04-11 00:00:00	Shipped	NULL
	10404	2005-04-08 00:00:00	2005-04-14 00:00:00	2005-04-11 00:00:00	Shipped	NULL
	10405	2005-04-14 00:00:00	2005-04-24 00:00:00	2005-04-20 00:00:00	Shipped	NULL
	10406	2005-04-15 00:00:00	2005-04-25 00:00:00	2005-04-21 00:00:00	Disputed	Customer claims container with shipment was damaged during shipment.
	10407	2005-04-22 00:00:00	2005-05-04 00:00:00	NULL	On Hold	Customer credit limit exceeded. Will ship when a payment is received.
	10408	2005-04-22 00:00:00	2005-04-29 00:00:00	2005-04-27 00:00:00	Shipped	NULL
	10409	2005-04-23 00:00:00	2005-05-05 00:00:00	2005-04-24 00:00:00	Shipped	NULL
	10410	2005-04-29 00:00:00	2005-05-10 00:00:00	2005-04-30 00:00:00	Shipped	NULL
	10411	2005-05-01 00:00:00	2005-05-08 00:00:00	2005-05-06 00:00:00	Shipped	NULL

Hàm EXTRACT: tách ra các giá trị như ngày, tháng, năm từ một giá trị có kiểu thời gian. (chú ý có thể dùng hàm MONTH hoặc YEAR thay thế)

Ví dụ: đưa ra tháng của một giá trị thời gian:

```
SELECT EXTRACT(MONTH FROM '2004-12-31 23:59:59');
```

	EXTRACT(MONTH FROM '2004-12-31 23:59:59')
▶	12

Ví dụ: đưa ra tháng của một giá trị thời gian:

```
SELECT EXTRACT(YEAR FROM '2004-12-31 23:59:59');
```

	EXTRACT(YEAR FROM '2004-12-31 23:59:59')
▶	2004

Ví dụ: đưa ra các đơn hàng đặt năm 2005

```
SELECT *
```

```
FROM orders
```

```
WHERE EXTRACT(YEAR FROM orderDate) = 2005
```

	orderNumber	orderDate	requiredDate	shippedDate	status	comments
▶	10362	2005-01-05 00:00:00	2005-01-16 00:00:00	2005-01-10 00:00:00	Shipped	NULL
	10363	2005-01-06 00:00:00	2005-01-12 00:00:00	2005-01-10 00:00:00	Shipped	NULL
	10364	2005-01-06 00:00:00	2005-01-17 00:00:00	2005-01-09 00:00:00	Shipped	NULL
	10365	2005-01-07 00:00:00	2005-01-18 00:00:00	2005-01-11 00:00:00	Shipped	NULL
	10366	2005-01-10 00:00:00	2005-01-19 00:00:00	2005-01-12 00:00:00	Shipped	NULL
	10367	2005-01-12 00:00:00	2005-01-21 00:00:00	2005-01-16 00:00:00	Resolved	This order was disputed and resolved on 2/1/2005. Customer
	10368	2005-01-19 00:00:00	2005-01-27 00:00:00	2005-01-24 00:00:00	Shipped	Can we renegotiate this one?
	10369	2005-01-20 00:00:00	2005-01-28 00:00:00	2005-01-24 00:00:00	Shipped	NULL
	10370	2005-01-20 00:00:00	2005-02-01 00:00:00	2005-01-25 00:00:00	Shipped	NULL
	10371	2005-01-23 00:00:00	2005-02-03 00:00:00	2005-01-25 00:00:00	Shipped	NULL
	10372	2005-01-26 00:00:00	2005-02-05 00:00:00	2005-01-28 00:00:00	Shipped	NULL
	10373	2005-01-31 00:00:00	2005-02-08 00:00:00	2005-02-06 00:00:00	Shipped	NULL
	10374	2005-02-02 00:00:00	2005-02-09 00:00:00	2005-02-03 00:00:00	Shipped	NULL
	10375	2005-02-03 00:00:00	2005-02-10 00:00:00	2005-02-06 00:00:00	Shipped	NULL
	10376	2005-02-08 00:00:00	2005-02-18 00:00:00	2005-02-13 00:00:00	Shipped	NULL
	10377	2005-02-09 00:00:00	2005-02-21 00:00:00	2005-02-12 00:00:00	Shipped	Cautious optimism. We have happy customers here, if we can
	10378	2005-02-10 00:00:00	2005-02-18 00:00:00	2005-02-11 00:00:00	Shipped	NULL
	10379	2005-02-10 00:00:00	2005-02-18 00:00:00	2005-02-11 00:00:00	Shipped	NULL
	10380	2005-02-16 00:00:00	2005-02-24 00:00:00	2005-02-18 00:00:00	Shipped	NULL
	10381	2005-02-17 00:00:00	2005-02-25 00:00:00	2005-02-18 00:00:00	Shipped	NULL
	10382	2005-02-17 00:00:00	2005-02-25 00:00:00	2005-02-18 00:00:00	Shipped	Custom shipping instructions sent to warehouse

Ví dụ: đưa ra các đơn hàng đặt trong tháng 5 năm 2005

```
SELECT *
FROM orders
WHERE EXTRACT(YEAR FROM orderDate) = 2005 and EXTRACT(MONTH
FROM orderDate) = 5;
```

	orderNumber	orderDate	requiredDate	shippedDate	status	comments
▶	10411	2005-05-01 00:00:00	2005-05-08 00:00:00	2005-05-06 00:00:00	Shipped	NULL
	10412	2005-05-03 00:00:00	2005-05-13 00:00:00	2005-05-05 00:00:00	Shipped	NULL
	10413	2005-05-05 00:00:00	2005-05-14 00:00:00	2005-05-09 00:00:00	Shipped	Customer requested that DHL is used for this shipping
	10414	2005-05-06 00:00:00	2005-05-13 00:00:00	NULL	On Hold	Customer credit limit exceeded. Will ship when a payment is received.
	10415	2005-05-09 00:00:00	2005-05-20 00:00:00	2005-05-12 00:00:00	Disputed	Customer claims the scales of the models don't match what was discuss
	10416	2005-05-10 00:00:00	2005-05-16 00:00:00	2005-05-14 00:00:00	Shipped	NULL
	10417	2005-05-13 00:00:00	2005-05-19 00:00:00	2005-05-19 00:00:00	Disputed	Customer doesn't like the colors and precision of the models.
	10418	2005-05-16 00:00:00	2005-05-24 00:00:00	2005-05-20 00:00:00	Shipped	NULL
	10419	2005-05-17 00:00:00	2005-05-28 00:00:00	2005-05-19 00:00:00	Shipped	NULL
	10420	2005-05-29 00:00:00	2005-06-07 00:00:00	NULL	In Process	NULL
	10421	2005-05-29 00:00:00	2005-06-06 00:00:00	NULL	In Process	Custom shipping instructions were sent to warehouse
	10422	2005-05-30 00:00:00	2005-06-11 00:00:00	NULL	In Process	NULL
	10423	2005-05-30 00:00:00	2005-06-05 00:00:00	NULL	In Process	NULL

❖ **Bài tập thực hành:**

1. Lấy ra 50 ký tự đầu tiên của phần mô tả sản phẩm, đặt tên là ‘Title of products’
2. Đưa ra mô tả về các nhân viên theo định dạng ‘Fullname, jobTitle.’
3. Thay thế toàn bộ tên nhóm hàng ‘Cars’ thành ‘Automobiles’.
4. Tìm 5 đơn hàng được vận chuyển sớm nhất so với ngày hẹn.
5. Đưa ra các đơn đặt hàng trong tháng 5 năm 2005 và có ngày chuyển hàng đến chưa xác định.