

# **BÀI TẬP LỚN 1**

## **THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN**

Môn: Cơ sở dữ liệu phân tán

Lớp: **IS211.M11.2** - Nhóm 10

**Phạm Quốc Hùng**    19521579    [19521579@gm.uit.edu.vn](mailto:19521579@gm.uit.edu.vn)

**Chu Hà Thảo Ngân**    19521882    [19521882@gm.uit.edu.vn](mailto:19521882@gm.uit.edu.vn)

**Thái Minh Triết**    19522397    [19522397@gm.uit.edu.vn](mailto:19522397@gm.uit.edu.vn)

**Võ Tuấn Anh**    19521226    [19521226@gm.uit.edu.vn](mailto:19521226@gm.uit.edu.vn)

**THÁNG 12/2021**

# Mục lục

Yêu cầu 1: Thiết kế cơ sở dữ liệu trên môi trường phân tán	3
1. Thiết kế cơ sở dữ liệu	3
2. Tạo User quản lý, sử dụng database	8
2.1. Các Role và ý nghĩa	8
2.2. Các User và phân quyền cho User	9
3. Tạo database link trên Oracle	10
4. Thực hiện truy vấn trên Oracle	10
4.1. Tại máy ZEHN01	10
4.2. Tại máy ZEHN02	18
Yêu cầu 2: Trigger, Procedure, Function trên môi trường phân tán	24
1. Trigger	24
2. Procedure	25
3. Function	26
Yêu cầu 3: Isolation level trong môi trường phân tán	28
1. Lost update	28
2. Non-repeatable	29
3. Deadlock	30
4. Phantom read	32
Yêu cầu 4: Thực hiện tối ưu hóa truy vấn trên môi trường phân tán	34
1. Câu truy vấn ban đầu	34
2. Tối ưu hoá tập trung	36
3. Explain Plan	39
3.1. Câu truy vấn ban đầu	39
3.2. Câu truy vấn tối ưu hoá	39

4. Câu truy vấn tối ưu hoá trên môi trường phân tán	40
Phụ lục: Các đường dẫn liên quan	42

*Yêu cầu 1:*

## Thiết kế cơ sở dữ liệu trên môi trường phân tán

### 1. Thiết kế cơ sở dữ liệu

Công ty TNHH dược phẩm Zehn :) đang quản lý các hệ thống nhà thuốc ở 3 quận: Gò Vấp, Thủ Đức và Quận 9

Lược đồ quan hệ toàn cục “Quản lý nhà thuốc” của Công ty TNHH dược phẩm Zehn như sau, với mỗi quan hệ ta có bảng tên từ như bên dưới:

1. **ZEHNSTORE**(StoreId, StoreName, Address)
2. **PHARMACIST**(PharmacistId, FullName, Gender, DoB, PhoneNumber, Address, WorkYear, WorkShift, *StoreId*)
3. **CUSTOMER**(PhoneNumber, FullName, Gender, DoB, ZehnPoint, CustomerType)
4. **PRODUCT**(ProductId, ProductName, ProductType, ExpiredDate, CountUnit, Price)
5. **RECEIPT**(ReceiptId, *CustomerId*, *PharmacistId*, *StoreId*, PaymentTime, Total, PaymentMethod)
6. **RECEIPTDETAIL**(ReceiptId, ProductId, Quantity, Price, Amount)

Quan hệ ZEHNSTORE		
Tên thuộc tính	Thuộc tính Khoá	Ý nghĩa thuộc tính
<u>StoreId</u>	PRIMARY	Mã nhà thuốc
StoreName		Tên nhà thuốc
Address		Địa chỉ nhà thuốc
Quan hệ PHARMACIST		

Tên thuộc tính	Thuộc tính Khoá	Ý nghĩa thuộc tính
<u>PharmacistId</u>	<b>PRIMARY</b>	Mã nhân viên
FullName		Họ và tên
Gender		Giới tính
DoB		Ngày tháng năm sinh
PhoneNumber		Số điện thoại
Address		Địa chỉ thường trú
WorkYear		Năm vào làm
WorkShift		Ca làm việc (1, 2, 3, 4 tương ứng với ca sáng, trưa, chiều-tối, đêm)
StoreID	<b>ZEHNSTORE</b> .StoreId	Mã nhà thuốc nhân viên làm việc
<b>Quan hệ CUSTOMER</b>		
Tên thuộc tính	Thuộc tính Khoá	Ý nghĩa thuộc tính
<u>PhoneNumber</u>	<b>PRIMARY</b>	Số điện thoại duy nhất để phân biệt với khách hàng khác
FullName		Họ tên khách hàng
Gender		Giới tính
DoB		Ngày tháng năm sinh
ZehnPoint		Điểm tích lũy khi mua hàng
CustomerType		Khách là 1 trong 3 loại khách hàng (“BasicCare”, “StandardCare”, “ExtraCare”)

Quan hệ <b>PRODUCT</b>		
Tên thuộc tính	Thuộc tính Khoá	Ý nghĩa thuộc tính
<u>ProductId</u>	<b>PRIMARY</b>	Mã sản phẩm ( <i>sử dụng mã vạch duy nhất</i> )
ProductName		Tên sản phẩm
ProductType		Loại sản phẩm (“TPCN”, “KD”, “KKD”, “Others” <i>tương ứng với Thực phẩm Chức năng, Thuốc kê đơn, thuốc không kê đơn, khác</i> )
ExpiredDate		Ngày hết hạn
CountUnit		Đơn vị tính
Price		Giá sản phẩm / đơn vị tính
Quan hệ <b>RECEIPT</b>		
Tên thuộc tính	Thuộc tính Khoá	Ý nghĩa thuộc tính
<u>ReceiptId</u>	<b>PRIMARY</b>	Mã hoá đơn
CustomerId	CUSTOMER .PhoneNumber	Mã khách hàng ( <i>mỗi khách hàng có một hoặc nhiều hoá đơn</i> )
PharmacistId	PHARMACIST .PharmacistId	Mã nhân viên lập hoá đơn
StoreId	STORE .StoreId	Mã nhà thuốc thực hiện lập hoá đơn
PaymentTime		Thời gian thanh toán hoá đơn
Total		Tổng tiền các chi tiết hoá đơn
Payment		Hình thức thanh toán (“Cash”, “Credit”,

Method		<i>“Zehn Point” tương ứng với thanh toán sử dụng tiền mặt, thẻ tín dụng hoặc ghi nợ, và điểm tích lũy Zehn Point)</i>
<b>Quan hệ RECEIPTDETAIL</b>		
<b>Tên thuộc tính</b>	<b>Thuộc tính Khoá</b>	<b>Ý nghĩa thuộc tính</b>
<u>ReceiptId</u>	RECEIPT .ReceiptId	Mã hoá đơn liên kết ( <i>Mỗi hoá đơn có một hoặc nhiều chi tiết</i> )
<u>ProductId</u>	PRODUCT .ProductId	Mã sản phẩm
Quantity		Số lượng sản phẩm
Price	PRODUCT .Price	Đơn giá sản phẩm
Amount		Tổng tiền chi tiết (Quantity * Price)

Quan hệ **ZEHNSTORE** phân mảnh ngang chính theo địa chỉ

$$ZEHNSTORE_1 = \sigma_{Address='GoVap'} ZEHNSTORE$$

$$ZEHNSTORE_2 = \sigma_{Address='ThuDuc'} ZEHNSTORE$$

$$ZEHNSTORE_3 = \sigma_{Address='Quan9'} ZEHNSTORE$$

Quan hệ **RECEIPT** phân mảnh ngang dẫn xuất như sau:

$$RECEIPT_1 = RECEIPT_{\times StoreId} ZEHNSTORE_1$$

$$RECEIPT_2 = RECEIPT_{\times StoreId} ZEHNSTORE_2$$

$$RECEIPT_3 = RECEIPT_{\times StoreId} ZEHNSTORE_3$$

Quan hệ **RECEIPTDETAIL** phân mảnh ngang dẫn xuất như sau:

$$RECEIPTDETAIL_1 = RECEIPTDETAIL_{\times StoreId} RECEIPT_1$$

$$RECEIPTDETAIL_2 = RECEIPTDETAIL_{\times StoreId} RECEIPT_2$$

$$RECEIPTDETAIL_3 = RECEIPTDETAIL_{\times StoreId} RECEIPT_3$$

Quan hệ **PHARMACIST** phân mảnh ngang dẫn xuất như sau:

$$PHARMACIST_1 = PHARMACIST_{\times StoreId} ZEHNSTORE_1$$

$$PHARMACIST_2 = PHARMACIST_{\times StoreId} ZEHNSTORE_2$$

$$PHARMACIST_3 = PHARMACIST_{\times StoreId} ZEHNSTORE_3$$

Quan hệ **CUSTOMER, PRODUCT** được nhân bản tại tất cả các nhà thuốc



## 2. Tạo User quản lý, sử dụng database

### 2.1. Các Role và ý nghĩa

Tên quyền	Phân quyền					
	ZEHN STORE	PHARMA CIST	CUSTOM ER	PRODUC T	RECEIPT	RECEIPT DETAIL
Giám đốc	INSERT UPDATE DELETE SELECT	INSERT UPDATE DELETE SELECT	SELECT	INSERT UPDATE DELETE SELECT	SELECT	SELECT
Cửa hàng trưởng	SELECT	INSERT UPDATE DELETE SELECT	INSERT UPDATE DELETE SELECT	INSERT UPDATE DELETE SELECT	INSERT UPDATE DELETE SELECT	INSERT UPDATE DELETE SELECT
Thu ngân	SELECT	SELECT	INSERT UPDATE DELETE SELECT	SELECT	INSERT UPDATE DELETE SELECT	INSERT UPDATE DELETE SELECT

#### Lưu ý:

- Quyền giám đốc **truy cập, chỉnh sửa** được mọi quan hệ tại mọi tiệm thuốc. Ngoài ra, DIRECTOR còn có quyền CREATE, ALTER, DROP, SELECT DATABASE LINK để tạo database link và quyền CREATE, ALTER, DROP, SELECT USER để quản lý user và GRANT ROLE để quản lý phân quyền
- Quyền cửa hàng trưởng chỉ **chỉnh sửa** được quan hệ tại tiệm thuốc mà cửa hàng trưởng quản lý. Nhưng có thể **truy cập** được quan hệ tại các tiệm thuốc khác
- Quyền Thu ngân chỉ được **truy cập, chỉnh sửa** quan hệ tại tiệm thuốc mà nhân viên được phân công. Và quyền thu ngân được cấp quyền thủ công nên sẽ không có Role cụ thể trong Oracle Database.
- Ứng với mỗi role sẽ có chung quyền CONNECT để đăng nhập.

## 2.2. Các User và phân quyền cho User

- Trên mỗi máy trạm, đều có tài khoản Oracle database tại máy đó, được phân quyền DBA. Ví dụ tại tiệm thuốc 1 sẽ có tài khoản zehn\_01. Tài khoản này được tạo bởi quyền SYSDBA
- Tài khoản director sẽ được phân bổ tại mọi tiệm thuốc để Giám đốc có thể truy cập tại bất kỳ đâu.
- Tài khoản manager\_01, manager\_02, manager\_03 là tài khoản của Cửa hàng trưởng dùng để đăng nhập vào tiệm thuốc của mình quản lý.
- Tài khoản zehn\_bridge được phân bổ đều tại mọi tiệm thuốc, đây là tài khoản trung gian giúp cho các Cửa hàng trưởng có thể truy xuất thông tin của tiệm thuốc khác.
- Để dễ quản lý trong quá trình thao tác, mọi tài khoản sẽ được đặt cùng một mật khẩu là: 123456

Phân quyền	Tên User tại các tiệm thuốc		
	zehn_01	zehn_02	zehn_03
Giám đốc	director	director	director
Cửa hàng trưởng	manager_01 zehn_bridge	manager_02 zehn_bridge	manager_03 zehn_bridge
Thu ngân	cashier_01	cashier_02	cashier_03

### 3. Tạo database link trên Oracle

#### Tại máy ZEHN01

```
CREATE DATABASE LINK director_01_02 CONNECT TO director
IDENTIFIED BY "123456" USING 'ZEHN_01';
CREATE DATABASE LINK manager_01_02 CONNECT TO zehn_bridge
IDENTIFIED BY "123456" USING 'ZEHN_01';
```

#### Tại máy ZEHN02

```
CREATE DATABASE LINK director_02_01 CONNECT TO director
IDENTIFIED BY "123456" USING 'ZEHN_02';
CREATE DATABASE LINK manager_02_01 CONNECT TO zehn_bridge
IDENTIFIED BY "123456" USING 'ZEHN_02';
```

### 4. Thực hiện truy vấn trên Oracle

#### 4.1. Tại máy ZEHN01

**Câu 1:** *Tìm tất cả khách hàng có giới tính là nữ mua sản phẩm thực phẩm chức năng vào ngày 20/10 và có giá trị hóa đơn từ 500.000 trở lên.*

**Ý nghĩa:** Dùng trong xem xét tỉ lệ khách nữ mua hàng vào ngày 20/10 so với tổng lượng mua của ngày đó.

*Truy vấn tại máy ZEHN01*

**Tài khoản của hàng trưởng:** manager\_01/123456

```
SELECT
    C.PhoneNumber, C.FullName
FROM
    ZEHN_01.CUSTOMER C
WHERE Gender = 'Nu'
    AND NOT EXISTS (
        SELECT * FROM ZEHN_01.RECEIPT R
        WHERE EXTRACT(DAY FROM R.PaymentTime)= 20
            AND EXTRACT(MONTH FROM R.PaymentTime)= 10
            AND R.Total >= 500000
```

```

AND NOT EXISTS (
  SELECT * FROM
    ZEHN_01.RECEIPTDETAIL D,
    ZEHN_01.PRODUCT Pr
  WHERE C.PhoneNumber = R.CustomerId
        AND R.ReceiptId = D.ReceiptId
        AND D.ProductId = Pr.ProductId
        AND Pr.ProductType = 'TPCN'
)
);

```

PHONENUMBER	FULLNAME
0700382483	Le Diem Thu
0976716565	Pham Ha Nhi
0349093356	Nguyen Chi Lan
0845126570	Ngo Nguyen Ngoc Quynh
0359055883	Nguyen Lam Ha
0836564633	Nguyen Thi My Dung
0355972520	Nguyen Hoang Phi Phi
0827397398	Nguyen Nguyet Minh
0966657718	Tran Nguyen Bao Truc
0852896366	Huynh Nguyen My Hoan
0836289221	Pham Phuong Nhung
PHONENUMBER	FULLNAME
0828228999	Le Nhat Thuong
0854263834	Nguyen Mai Thy
0785368107	Ngo Thanh Vy
0847900236	Nguyen Le Minh Khue
0861347402	Huynh Thuy Du
0361230015	Nguyen Pham An Binh
0373677397	Le Thanh Hao

18 rows selected.

**Câu 2:** *Tìm những sản phẩm thực phẩm chức năng (mã sản phẩm, tên sản phẩm) bán được tại nhà thuốc ZEHN01 nhưng không bán được tại nhà thuốc ZEHN02.*

**Ý nghĩa:** Nhằm nắm được loại sản phẩm không có phân khúc khách hàng phù hợp với mỗi địa phương khác nhau.

*Truy vấn tại máy ZEHN01 tới máy ZEHN02*

**Tài khoản giám đốc:** director/123456

```
SELECT
    Pr.ProductId, Pr.ProductName
FROM
    ZEHN_01.PRODUCT Pr INNER JOIN ZEHN_01.RECEIPTDETAIL D1
        ON Pr.ProductId = D1.ProductId
WHERE Pr.ProductType = 'TPCN'
MINUS
SELECT
    Pr.ProductId, Pr.ProductName
FROM
    ZEHN_01.PRODUCT Pr INNER JOIN
ZEHN_02.RECEIPTDETAIL@director_01_02 D2
        ON Pr.ProductId = D2.ProductId
WHERE Pr.ProductType = 'TPCN';
```

```
SQL> SELECT
2     Pr.ProductId, Pr.ProductName
3 FROM
4     ZEHN_01.PRODUCT Pr INNER JOIN ZEHN_01.RECEIPTDETAIL D1
5         ON Pr.ProductId = D1.ProductId
6 WHERE Pr.ProductType = 'TPCN'
7 MINUS
8 SELECT
9     Pr.ProductId, Pr.ProductName
10 FROM
11     ZEHN_01.PRODUCT Pr INNER JOIN ZEHN_02.RECEIPTDETAIL@director_01_02 D2
12         ON Pr.ProductId = D2.ProductId
13 WHERE Pr.ProductType = 'TPCN';
```

PRODUCTID	PRODUCTNAME
PR14	Baigout
PR31	Livers Gold Plus
PR33	Tam That Cali USA Nano Gold

SQL> |

**Câu 3:** Khách hàng có số điện thoại “0985367353” và “0399988381” được phát hiện dương tính với Covid-19. Xuất thông tin nhân viên (mã nhân viên, tên nhân viên, số điện thoại, ngày bán) ở tất cả chi nhánh từng tiếp xúc với hai khách hàng trên trong khoảng thời gian từ ngày 15/11/2021 đến 30/11/2021.

Truy vấn tại máy ZEHN01 tới máy ZEHN02

**Tài khoản cửa hàng trưởng:** manager\_01/123456

```
SELECT
    Ph1.PharmacistId, Ph1.FullName,
    Ph1.PhoneNumber, R1.PaymentTime
FROM
    ZEHN_01.PHARMACIST Ph1,
    ZEHN_01.RECEIPT R1,
    ZEHN_01.CUSTOMER C1
WHERE
    Ph1.PharmacistId = R1.PharmacistId
    AND R1.CustomerId = C1.PhoneNumber
    AND (C1.PhoneNumber = '0985367353'
        OR C1.PhoneNumber = '0399988381')
    AND R1.PaymentTime BETWEEN
        TO_DATE('2021-11-15', 'YYYY-MM-DD')
        AND TO_DATE('2021-11-30', 'YYYY-MM-DD')
UNION
SELECT
    Ph2.PharmacistId, Ph2.FullName,
    Ph2.PhoneNumber, R2.PaymentTime
FROM
    ZEHN_02.PHARMACIST@manager_01_02 Ph2,
    ZEHN_02.RECEIPT@manager_01_02 R2,
    ZEHN_01.CUSTOMER C2
WHERE
    Ph2.PharmacistId = R2.PharmacistId
    AND R2.CustomerId = C2.PhoneNumber
    AND R2.PaymentTime BETWEEN
        TO_DATE('2021-11-15', 'YYYY-MM-DD')
        AND TO_DATE('2021-11-30', 'YYYY-MM-DD')
    AND (C2.PhoneNumber = '0985367353'
        OR C2.PhoneNumber = '0399988381');
```

```

SQL> SELECT
2     Ph1.PharmacistId, Ph1.FullName, Ph1.PhoneNumber, R1.PaymentTime
3 FROM
4     ZEHN_01.PHARMACIST Ph1,
5     ZEHN_01.RECEIPT R1,
6     ZEHN_01.CUSTOMER C1
7 WHERE
8     Ph1.PharmacistId = R1.PharmacistId
9     AND R1.CustomerId = C1.PhoneNumber
10    AND (C1.PhoneNumber = '0985367353' OR C1.PhoneNumber = '0399988381')
11    AND R1.PaymentTime BETWEEN
12        TO_DATE('2021-11-15', 'YYYY-MM-DD')
13        AND TO_DATE('2021-11-30', 'YYYY-MM-DD')
14 UNION
15 SELECT
16     Ph2.PharmacistId, Ph2.FullName, Ph2.PhoneNumber, R2.PaymentTime
17 FROM
18     ZEHN_02.PHARMACIST@manager_01_02 Ph2,
19     ZEHN_02.RECEIPT@manager_01_02 R2,
20     ZEHN_01.CUSTOMER C2
21 WHERE
22     Ph2.PharmacistId = R2.PharmacistId
23     AND R2.CustomerId = C2.PhoneNumber
24     AND R2.PaymentTime BETWEEN
25        TO_DATE('2021-11-15', 'YYYY-MM-DD')
26        AND TO_DATE('2021-11-30', 'YYYY-MM-DD')
27    AND (C2.PhoneNumber = '0985367353' OR C2.PhoneNumber = '0399988381');

```

PHARMACIST	FULLNAME	PHONENUMBER	PAYMENTTI
PH05	Tran Mai Anh	0868278597	25-NOV-21
PH34	Nguyen Bao Duy	0361521452	15-NOV-21

```
SQL> |
```

**Câu 4:** *Tìm sản phẩm còn hạn sử dụng 30 ngày và đều chưa bán được lần nào ở tất cả chi nhánh*

**Ý nghĩa:** Nhằm tìm ra sản phẩm không bán được để có thể hủy bỏ hàng tồn và tránh nhập thêm đơn hàng mới

*Truy vấn tại máy ZEHN01 tới máy ZEHN02*

**Tài khoản cửa hàng trưởng:** manager\_01/123456

```
SELECT * FROM ZEHN_01.PRODUCT Pr
WHERE
    Pr.ExpiredDate <= (SYSDATE + 30)
    AND Pr.ExpiredDate > SYSDATE
    AND Pr.ProductId NOT IN (
        SELECT Pr.ProductId FROM
            ZEHN_01.RECEIPT R1, ZEHN_01.RECEIPTDETAIL D1
        WHERE
            R1.ReceiptId = D1.ReceiptId
            AND D1.ProductId = Pr.ProductId
    )
INTERSECT
SELECT * FROM ZEHN_01.PRODUCT Pr
WHERE
    Pr.ExpiredDate <= (SYSDATE + 30)
    AND Pr.ExpiredDate > SYSDATE
    AND Pr.ProductId NOT IN (
        SELECT Pr.ProductId FROM
            ZEHN_02.RECEIPT@manager_01_02 R2,
            ZEHN_02.RECEIPTDETAIL@manager_01_02 D2
        WHERE R2.ReceiptId = D2.ReceiptId
            AND D2.ProductId = Pr.ProductId
    );
```



```

SQL> SELECT * FROM ZEHN_01.PRODUCT Pr
2  WHERE
3      Pr.ExpiredDate <= (SYSDATE + 30)
4      AND Pr.ExpiredDate > SYSDATE
5      AND Pr.ProductId NOT IN (
6          SELECT Pr.ProductId FROM
7              ZEHN_01.RECEIPT R1, ZEHN_01.RECEIPTDETAIL D1
8          WHERE
9              R1.ReceiptId = D1.ReceiptId
10             AND D1.ProductId = Pr.ProductId
11     )
12 INTERSECT
13 SELECT * FROM ZEHN_01.PRODUCT Pr
14 WHERE
15     Pr.ExpiredDate <= (SYSDATE + 30)
16     AND Pr.ExpiredDate > SYSDATE
17     AND Pr.ProductId NOT IN (
18         SELECT Pr.ProductId FROM
19             ZEHN_02.RECEIPT@manager_01_02 R2,
20             ZEHN_02.RECEIPTDETAIL@manager_01_02 D2
21         WHERE R2.ReceiptId = D2.ReceiptId
22             AND D2.ProductId = Pr.ProductId
23     );
rows will be truncated

rows will be truncated

rows will be truncated

rows will be truncated

no rows selected

SQL> |

```

**Câu 5:** In thông tin tổng số lượng đơn vị sản phẩm thanh toán bằng “Credit Card” và tổng doanh thu theo thời gian trong ngày (ca).

**Ý nghĩa:** Thống kê được tỷ lệ người dùng thanh toán bằng thẻ ngân hàng và lập bảng chi tiết sao kê cho bên ngân hàng liên kết.

Truy xuất tại máy ZEHN01

**Tài khoản thu ngân:** cashier\_01/123456

```
SELECT
    Ph1.WorkShift, SUM(D1.Quantity), SUM(D1.AMOUNT)
FROM
    ZEHN_01.PHARMACIST Ph1,
    ZEHN_01.RECEIPT R1,
    ZEHN_01.RECEIPTDETAIL D1
WHERE
    Ph1.PharmacistId = R1.PharmacistId
    AND R1.ReceiptId = D1.ReceiptId
    AND R1.PaymentMethod = 'Credit'
GROUP BY Ph1.WorkShift;
```

```
SQL> SELECT
2     Ph1.WorkShift, SUM(D1.Quantity), SUM(D1.AMOUNT)
3 FROM
4     ZEHN_01.PHARMACIST Ph1,
5     ZEHN_01.RECEIPT R1,
6     ZEHN_01.RECEIPTDETAIL D1
7 WHERE
8     Ph1.PharmacistId = R1.PharmacistId
9     AND R1.ReceiptId = D1.ReceiptId
10    AND R1.PaymentMethod = 'Credit'
11 GROUP BY Ph1.WorkShift;

WORKSHIFT SUM(D1.QUANTITY) SUM(D1.AMOUNT)
-----
1          1          25000
2          1          112000
3          9          2530000

SQL> |
```

## 4.2. Tại máy ZEHN02

**Câu 6:** In ra thông tin khách hàng những khách hàng đang có ZehnPoint tích lũy tối thiểu là 100k trở lên.

**Ý nghĩa:** Lập danh sách những khách hàng có thể sử dụng điểm ZehnPoint để thanh toán mà không cần dùng tiền mặt cho lần thanh toán tiếp theo.

*Truy vấn tại máy ZEHN\_02*

**Tài khoản thu ngân:** cashier\_02/123456

```
SELECT
    C2.PhoneNumber, FullName, ZehnPoint
FROM
    zehn_02.CUSTOMER C2
WHERE
    ZehnPoint >= 100000;
```

```
SQL> SELECT
  2      C2.PhoneNumber, FullName, ZehnPoint
  3 FROM
  4      zehn_02.CUSTOMER C2
  5 WHERE
  6      ZehnPoint ≥ 100000;
```

PHONENUMBER	FULLNAME	ZEHNPOINT
0349093356	Nguyen Chi Lan	103000
0332051377	Pham Tran Hai Thuy	503000
0967973007	Dang Trung Nhan	103500
0362974048	Nguyen Tran Tien Vo	1050000
0828228999	Le Nhat Thuong	514000
0333967718	Nguyen Quang Danh	890500

6 rows selected.

**Câu 7:** *Số lượng tiêu thụ của từng product theo từng tháng từ tất cả các chi nhánh*

**Ý nghĩa:** Để biết được doanh số của các sản phẩm theo từng vị trí để đưa ra dự đoán kinh doanh và điều chỉnh số lượng nhập sản phẩm phù hợp.

*Truy vấn tại máy ZEHN02 tới máy ZEHN01*

**Tài khoản cửa hàng trưởng:** manager\_02/123456

```
SELECT
    EXTRACT(month FROM R2.PaymentTime) AS "Month" ,
    Sum(D2.Quantity) AS "Tong_san_luong",
    Pr.CountUnit,
    Pr.ProductName
FROM
    zehn_02.RECEIPTDETAIL D2,
    zehn_02.RECEIPT R2,
    zehn_02.PRODUCT Pr
WHERE
    R2.ReceiptId = D2.ReceiptId
    AND D2.ProductId = Pr.ProductId
GROUP BY R2.PaymentTime, D2.Quantity, Pr.CountUnit,
Pr.ProductName
UNION
SELECT
    EXTRACT(month FROM R1.PaymentTime) AS "Month" ,
    Sum(D1.Quantity) AS "Tong_san_luong",
    Pr.CountUnit,
    Pr.ProductName
FROM
    zehn_01.RECEIPTDETAIL@manager_02_01 D1,
    zehn_01.RECEIPT@manager_02_01 R1,
    zehn_02.PRODUCT Pr
WHERE
    R1.ReceiptId = D1.ReceiptId
    AND D1.ProductId = Pr.ProductId
GROUP BY R1.PaymentTime, D1.Quantity, Pr.CountUnit,
Pr.ProductName;
```

12	1 Hop	Bo cam bien may do duong huyet nhanh FreeSt	
12	1 Hop	Buscopan	
12	1 Hop	Dosaff	
12	1 Hop	Effer-Paralmax 500	
12	1 Hop	Livers Gold Plus	
12	1 Hop	Natures Way Beauty Collagen Gummies	
Month	Tong_san_luong	COUNTUNIT	PRODUCTNAME
12	1 Hop	Pepsane	
12	1 Hop	Viem ngam dieu tri viem hong Dorithricin	
12	1 Tuyp	Dr. Frei Magnesium + B complex	
12	2 Hop	Bao cao su Okamoto 0.03 Platinum	
12	2 Hop	Tam That Cali USA Nano Gold	
12	2 Hop	Thuc pham bao ve suc khoe Ferrovit C	
12	2 Hop	Thuoc dieu tri loet da day-ta trang Sucralf	
12	2 Hop	Viem ngam dieu tri viem hong Dorithricin	
12	3 Goi	Khau trang 3D Jomi Freesize	
12	3 Hop	Acyclovir	
12	3 Hop	Bo xet nghiem nhanh COVID-19 tai nha Humasi	
Month	Tong_san_luong	COUNTUNIT	PRODUCTNAME
12	3 Hop	Livers Gold Plus	
12	4 Hop	Pepsane	
12	5 Vien	Rupafin 10 mg	

80 rows selected.

**Câu 8:** Xuất ra những product chỉ còn hạn sử dụng trong 14 ngày (tính từ ngày hôm nay sysdate) tại máy ZEHN02

**Ý nghĩa:** Để nhân viên hiệu thuốc có thể thanh lý hoặc tiêu hủy trước khi đến hạn sử dụng

Truy vấn tại máy ZEHN02

**Tài khoản cửa hàng trưởng:** manager\_02/123456

```
SELECT
    ProductId, ProductName, ExpiredDate
FROM
    zehn_02.PRODUCT
WHERE
    ExpiredDate <= (SYSDATE + 14) AND ExpiredDate > SYSDATE;
```

```
SQL> SELECT
  2     ProductId, ProductName
  3 FROM
  4     zehn_02.PRODUCT
  5 WHERE
  6     ExpiredDate ≤ (SYSDATE + 14) AND ExpiredDate > SYSDATE;

PRODUCTID  PRODUCTNAME
-----
PR39       Bo cam bien may do duong huyet nhanh FreeStyle Libre
```

**Câu 9:** In thông tin của những dược sĩ có *WorkYear* >= "2015" và *WorkShift* = 4 tại tất cả chi nhánh

**Ý nghĩa:** Tìm xem những dược sĩ có gắn bó lâu với cửa hàng để xét tăng lương

Truy vấn tại máy ZEHN02 tới máy ZEHN01

**Tài khoản giám đốc:** director/123456

```
SELECT
    Ph2.*
FROM
    zehn_02.PHARMACIST Ph2
WHERE
    Ph2.WorkYear >= 2015 AND Ph2.WorkShift = 4
UNION
SELECT
    Ph1.*
FROM
    zehn_01.PHARMACIST@director_02_01 Ph1
WHERE
    Ph1.WorkYear >= 2015 AND Ph1.WorkShift = 4;
```

```
SQL> SELECT  Ph2.*
   2  FROM
   3      zehn_02.PHARMACIST Ph2
   4  WHERE
   5      Ph2.WorkYear ≥ 2015 AND Ph2.WorkShift = 4
   6  UNION
   7  SELECT Ph1.*
   8  FROM
   9      zehn_01.PHARMACIST@director_02_01 Ph1
  10  WHERE
  11      Ph1.WorkYear ≥ 2015 AND Ph1.WorkShift = 4;
rows will be truncated
```

rows will be truncated

rows will be truncated

PHARMACIST	FULLNAME	GENDE	DOB	PHONENUMBER	ADDRES
PH21	Hoang Van Huan Vo	Nam	21-JUL-84	0795685309	Xa Tru
PH27	Nguyen Vo Yen Anh	Nu	13-DEC-84	0842911997	Xa Pho
PH33	Do Nguyen Nhat Lan	Nu	05-JAN-99	0966603606	Phuong
PH37	Pham Quyet Thang	Nam	11-AUG-94	0336011148	Xa An
PH38	Nguyen Phuong Hien	Nu	19-APR-86	0354660091	Xa Ngh

**Câu 10:** *Liệt kê các product được tiêu thụ nhiều nhất tại từng chi nhánh.*

**Ý nghĩa:** *Tương tự câu 7*

*Truy vấn tại máy ZEHN02 tới máy ZEHN01*

**Tài khoản giám đốc:** director/123456

```
SELECT
    R2.StoreId, COUNT(D2.ProductID), ProductName
FROM
    zehn_02.RECEIPTDETAIL D2,
    zehn_02.PRODUCT Pr,
    zehn_02.RECEIPT R2
WHERE
    Pr.ProductID = D2.ProductID
    AND R2.ReceiptId = D2.ReceiptId
GROUP BY
    R2.StoreId, D2.ProductID, ProductName
HAVING COUNT(D2.ProductID) >= (
    SELECT MAX(COUNT(D2A.ProductID))
    FROM zehn_02.RECEIPTDETAIL D2A
    GROUP BY D2A.ProductID
)
)
UNION
SELECT
    R1.StoreId, COUNT(D1.ProductID), ProductName
FROM
    zehn_01.RECEIPTDETAIL@director_02_01 D1,
    zehn_01.PRODUCT@director_02_01 Pr,
    zehn_01.RECEIPT@director_02_01 R1
WHERE
    Pr.ProductID = D1.ProductID
    AND R1.ReceiptId = D1.ReceiptId
GROUP BY
    R1.StoreId, D1.ProductID, ProductName
HAVING COUNT(D1.ProductID) >= (
    SELECT MAX(COUNT(D1A.ProductID))
    FROM zehn_01.RECEIPTDETAIL@director_02_01 D1A
    GROUP BY D1A.ProductID
);
```

STOREID	COUNT(D2.PRODUCTID)	PRODUCTNAME
ZS01	6	Buscopan
ZS02	13	Khau trang 3D Jomi Freesize



## Yêu cầu 2:

# Trigger, Procedure, Function trên môi trường phân tán

## 1. Trigger

Dược sĩ phải đủ 18 tuổi khi vào làm việc

- Bối cảnh: PHARMACIST
- Nội dung:  $\forall p \in PHARMACIST(p.(WorkYear - YEAR(DoB)) \geq 18)$
- Bảng tầm ảnh hưởng:

	Insert	Delete	Update
PHARMACIST	+	-	+(WorkYear, DoB)

```
CREATE OR REPLACE TRIGGER trg_PHARMACIST_insert_update
AFTER INSERT OR UPDATE ON PHARMACIST FOR EACH ROW
BEGIN
    IF (:NEW.WorkYear - EXTRACT(year FROM :NEW.Dob) < 18)
    THEN
        RAISE_APPLICATION_ERROR(-20100,
            'Được sĩ phải tối thiểu 18 tuổi khi vào làm việc');
    END IF;
END;
```

**Kiểm tra trigger:** Khi dược sĩ vào làm việc mà chưa đủ 18 tuổi

```
SQL> INSERT INTO zehn_02.PHARMACIST VALUES('PH_test', 'Nguyen Tran Minh Long', 'Nam', '2006-12-23', '0366399947', 'Xã Hong Phong, Huyện Cao Lóc, Lạng Sơn',
2021, 1, 'ZS02');
INSERT INTO zehn_02.PHARMACIST VALUES('PH_test', 'Nguyen Tran Minh Long', 'Nam', '2006-12-23', '0366399947', 'Xã Hong Phong, Huyện Cao Lóc, Lạng Sơn', 2021,
1, 'ZS02')
*
ERROR at line 1:
ORA-20100: Được sĩ phải tối thiểu 18 tuổi khi vào làm việc
ORA-06512: at "ZEHN_02.TRG_PHARMACIST_INSERT", line 3
ORA-04088: error during execution of trigger 'ZEHN_02.TRG_PHARMACIST_INSERT'
```

## 2. Procedure

Procedure	Thay đổi ca làm WorkShift của pharmacist	
Procedure Name	ChangeWorkShift	
Arguments	v_PharmacistID v_WorkShift	Mã dược sĩ Ca làm việc
Side-effect	Tìm dược sĩ có v_PharmacistID trong bảng PHARMACIST tại từng chi nhánh và nếu tìm thấy thay đổi WorkShift thành v_WorkShift	
Query	<pre>CREATE OR REPLACE PROCEDURE ChangeWorkShift (v_PharmacistID in VARCHAR2, v_WorkShift in NUMBER) AS     dem int; BEGIN     SELECT COUNT(Ph2.PharmacistID) INTO dem     FROM zehn_02.PHARMACIST Ph2     WHERE Ph2.PharmacistID = v_PharmacistID;      IF (dem=1) THEN         UPDATE zehn_02.PHARMACIST         SET WorkShift = v_WorkShift         WHERE PharmacistID = v_PharmacistID;     ELSE         SELECT COUNT(Ph1.PharmacistID) INTO dem         FROM             zehn_01.PHARMACIST@director_02_01 Ph1         WHERE Ph1.PharmacistID = v_PharmacistID;         IF (dem =1) THEN             UPDATE zehn_01.PHARMACIST@director_02_01             SET WorkShift = v_WorkShift             WHERE PharmacistID = v_PharmacistID;         END IF;     END IF;     COMMIT; END;</pre>	
Trước khi chạy Procedure:		

	WORK...	PHARMACISTID
1	3	PH21

Thực hiện Procedure:

```
begin
    ChangeWorkShift ('PH21', 4);
end;
```

Sau khi chạy Procedure:

	WORK...	PHARMACISTID
1	4	PH21

### 3. Function

<b>Function</b>	Tính tổng tiền các hóa đơn của một khách hàng bất kỳ	
<b>Function Name</b>	SumTotalMoney	
<b>Arguments</b>	v_CustomerId	Mã khách hàng
<b>Output</b>	Tổng tiền các hóa đơn của khách hàng v_CustomerId	
<b>Side-effect</b>	Không ( <i>do dùng DDL, DML không tính là side-effect</i> )	
<b>Query</b>	<pre>CREATE OR REPLACE FUNCTION     SumToTalMoney(v_CustomerId IN VARCHAR2)     RETURN NUMBER IS     total_sum NUMBER :=0; BEGIN     SELECT SUM(R.Total) INTO total_sum     FROM ZEHN_01.Receipt R     WHERE R.CustomerId = v_CustomerId;     RETURN total_sum; END;</pre>	

```
SQL> select distinct SumTotalMoney('0985367353') from CUSTOMER;
```

```
SUMTOTALMONEY('0985367353')
```

```
-----
```

```
8076800
```

```
SQL> |
```

### *Yêu cầu 3:*

## **Isolation level trong môi trường phân tán**

### **1. Lost update**

<b>Time</b>	<b>ZEHN_01</b>	<b>ZEHN_02</b>
<i>T0</i>	ALTER SESSION SET ISOLATION_LEVEL=READ COMMITTED;	ALTER SESSION SET ISOLATION_LEVEL=READ COMMITTED;
<i>T1</i>	UPDATE zehn_01.PHARMACIST SET WorkYear = 2017 WHERE PharmacistId = 'PH01';	
		UPDATE zehn_01.PHARMACIST@director_02_01 SET WorkYear = 2010 WHERE PharmacistId = 'PH01';
<i>T2</i>	COMMIT;	
		COMMIT;
<i>T3</i>	SELECT * FROM zehn_01.Pharmacist Ph1 WHERE Ph1.PharmacistId = 'PH01';  (Kết quả UPDATE từ ZEHN_01 đã bị ghi đề bởi UPDATE ZEHN_02)	

### **Cách ngăn chặn**

<b>Time</b>	<b>ZEHN_01</b>	<b>ZEHN_02</b>
<i>T0</i>	ALTER SESSION SET ISOLATION_LEVEL=SERIALIZABLE;	
<i>T1</i>		ALTER SESSION SET ISOLATION_LEVEL=SERIALIZABLE;
<i>T2</i>	UPDATE zehn_01.PHARMACIST SET WorkYear = 2018 WHERE PharmacistId = 'PH02';	
<i>T3</i>	COMMIT;	

<b>T4</b>		UPDATE zehn_01.PHARMACIST@director_02_01 SET WorkYear = 2010 WHERE PharmacistId = 'PH02';  COMMIT;
<b>T5</b>	SELECT * FROM zehn_01.Pharmacist Ph1 WHERE Ph1.PharmacistId = 'PH02';  <i>(Kết quả UPDATE từ ZEHN_01 không bị ghi đè bởi UPDATE ZEHN_02)</i>	

## 2. Non-repeatable

<b>Time</b>	<b>ZEHN_01</b>	<b>ZEHN_02</b>
<b>T0</b>	ALTER SESSION SET ISOLATION_LEVEL=READ COMMITTED;	
<b>T1</b>		ALTER SESSION SET ISOLATION_LEVEL=READ COMMITTED;
<b>T2</b>	SELECT * FROM ZEHN_01.PHARMACIST WHERE PharmacistId = 'PH03';	
<b>T3</b>		UPDATE zehn_01.PHARMACIST@director_02_01 SET WorkYear = 2010 WHERE PharmacistId = 'PH03';
<b>T4</b>		COMMIT;
<b>T5</b>	SELECT * FROM ZEHN_01.PHARMACIST WHERE PharmacistId = 'PH03';  <i>(Kết quả truy xuất khác kết quả ban đầu)</i>	

### Cách ngăn chặn:

<b>Time</b>	<b>ZEHN_01</b>	<b>ZEHN_02</b>
<b>T0</b>	ALTER SESSION SET ISOLATION_LEVEL=SERIALIZABLE;	

<i>T1</i>		ALTER SESSION SET ISOLATION_LEVEL=SERIALIZABLE;
<i>T2</i>	SELECT * FROM ZEHN_01.PHARMACIST WHERE PharmacistId = 'PH04';	
<i>T3</i>		UPDATE zehn_01.PHARMACIST@director_02_01 SET WorkYear = 2010 WHERE PharmacistId = 'PH04'
<i>T4</i>		COMMIT;
<i>T5</i>	SELECT * FROM ZEHN_01.PHARMACIST WHERE PharmacistId = 'PH04'; (Kết quả như ban đầu)	
<i>T6</i>	COMMIT;	
<i>T7</i>	SELECT * FROM ZEHN_01.PHARMACIST WHERE PharmacistId = 'PH04'; (Nhận kết quả được update)	

### 3. Deadlock

#### Mức cô lập Read Committed

Time	ZEHN_01	ZEHN_02
<i>T0</i>	ALTER SESSION SET ISOLATION_LEVEL=READ COMMITTED;	
<i>T1</i>		ALTER SESSION SET ISOLATION_LEVEL=READ COMMITTED;
<i>T2</i>	UPDATE ZEHN_01.PRODUCT SET ProductType = 'TPCN' WHERE PRODUCTID = 'PR01';	
<i>T3</i>		UPDATE ZEHN_01.PRODUCT@director_02_01 SET ProductType= 'Others' WHERE ProductID= 'PR02';
<i>T4</i>	UPDATE ZEHN_01.PRODUCT SET ProductType = 'KD'	

	WHERE PRODUCTID = 'PR02';  (Chờ ZEHN_02)	
<b>T5</b>		UPDATE ZEHN_01.PRODUCT@director_02_01 SET ProductType = 'Others' WHERE PRODUCTID = 'PR01';
<b>T3</b>	<b>DEADLOCK</b>	
<b>T4</b>	COMMIT;	
<b>T5</b>		COMMIT;
<b>T6</b>	SELECT * FROM ZEHN_01.PRODUCT WHERE PRODUCTID = 'PR01' OR PRODUCTID = 'PR02';  (Kết quả cập nhật 2 khách hàng của ZEHN02)	

### Giải quyết bằng mức cô lập Serializable

<b>Time</b>	<b>ZEHN_01</b>	<b>ZEHN_02</b>
<b>T0</b>	ALTER SESSION SET ISOLATION_LEVEL=SERIALIZABLE;	
<b>T1</b>		ALTER SESSION SET ISOLATION_LEVEL=SERIALIZABLE;
<b>T2</b>	UPDATE ZEHN_01.PRODUCT SET ProductType = 'TPCN' WHERE PRODUCTID = 'PR03';	
<b>T3</b>		UPDATE ZEHN_01.PRODUCT@director_02_01 SET ProductType= 'Others' WHERE ProductID= 'PR04';
<b>T4</b>	UPDATE ZEHN_01.PRODUCT SET ProductType = 'KD' WHERE PRODUCTID = 'PR04';  Chờ ZEHN_02	



<i>T5</i>		UPDATE ZEHN_01.PRODUCT@director_02_01 SET ProductType = 'Others' WHERE PRODUCTID = 'PR03';
<i>T6</i>	<i>DEADLOCK</i>	
<i>T7</i>	COMMIT;	
<i>T8</i>		COMMIT;
<i>T9</i>	SELECT * FROM ZEHN_01.PRODUCT WHERE PRODUCTID = 'PR03' OR PRODUCTID = 'PR04';	

## 4. Phantom read

### Mức cô lập Read Committed

Time	ZEHN_01	ZEHN_02
<i>T0</i>	ALTER SESSION SET ISOLATION_LEVEL=READ COMMITTED;	
<i>T1</i>		ALTER SESSION SET ISOLATION_LEVEL=READ COMMITTED;
<i>T2</i>	SELECT COUNT(*) FROM ZEHN_01.PRODUCT;  <i>Nhận 40 dòng kết quả</i>	
<i>T3</i>		DELETE FROM ZEHN_01.PRODUCT@director_02_01 WHERE ProductID = 'PR01';
<i>T4</i>		COMMIT;
<i>T5</i>	SELECT COUNT(*) FROM ZEHN_01.PRODUCT;  <i>Nhận 39 dòng kết quả (Số dòng bị thay đổi so với kết quả truy vấn ở T1)</i>	

### Giải quyết bằng mức cô lập Serializable

Time	ZEHN_01	ZEHN_02
<i>T0</i>	ALTER SESSION SET ISOLATION_LEVEL=SERIALIZABLE;	
<i>T1</i>		ALTER SESSION SET ISOLATION_LEVEL=SERIALIZABLE;
<i>T2</i>	SELECT COUNT(*) FROM ZEHN_01.PRODUCT;  <i>Nhận 39 dòng kết quả</i>	
<i>T3</i>		DELETE FROM ZEHN_01.PRODUCT@director_02_01 WHERE ProductID = 'PR02';
<i>T4</i>		COMMIT;
<i>T5</i>	SELECT COUNT(*) FROM ZEHN_01.PRODUCT;  <i>Nhận 39 dòng kết quả (Số dòng không bị thay đổi so với kết quả truy vấn ở T1)</i>	
<i>T6</i>	COMMIT;	
<i>T7</i>	SELECT COUNT(*) FROM ZEHN_01.PRODUCT;  <i>Nhận 38 dòng kết quả (Số dòng thay đổi so với kết quả truy vấn ở T1)</i>	

*Yêu cầu 4:*

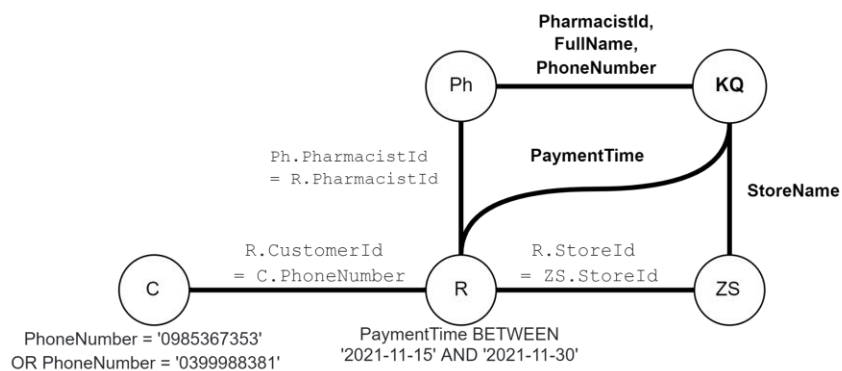
## **Thực hiện tối ưu hóa truy vấn trên môi trường phân tán**

### **1. Câu truy vấn ban đầu**

Khách hàng có số điện thoại “0985367353” và “0399988381” được phát hiện dương tính với Covid-19. Xuất thông tin nhân viên (mã nhân viên, tên nhân viên, số điện thoại, ngày bán) và tên chi nhánh nhân viên làm việc từng tiếp xúc với hai khách hàng trên trong khoảng thời gian từ ngày 15/11/2021 đến 30/11/2021.

```
SELECT
    Ph.PharmacistId, Ph.FullName, Ph.PhoneNumber,
    R.PaymentTime,
    ZS.StoreName
FROM
    PHARMACIST Ph,
    RECEIPT R,
    CUSTOMER C,
    ZEHNSTORE ZS
WHERE
    Ph.PharmacistId = R.PharmacistId
    AND R.CustomerId = C.PhoneNumber
    AND R.StoreId = ZS.StoreId
    AND (C.PhoneNumber = '0985367353'
        OR C.PhoneNumber = '0399988381')
    AND R.PaymentTime BETWEEN '2021-11-15'
        AND '2021-11-30';
```

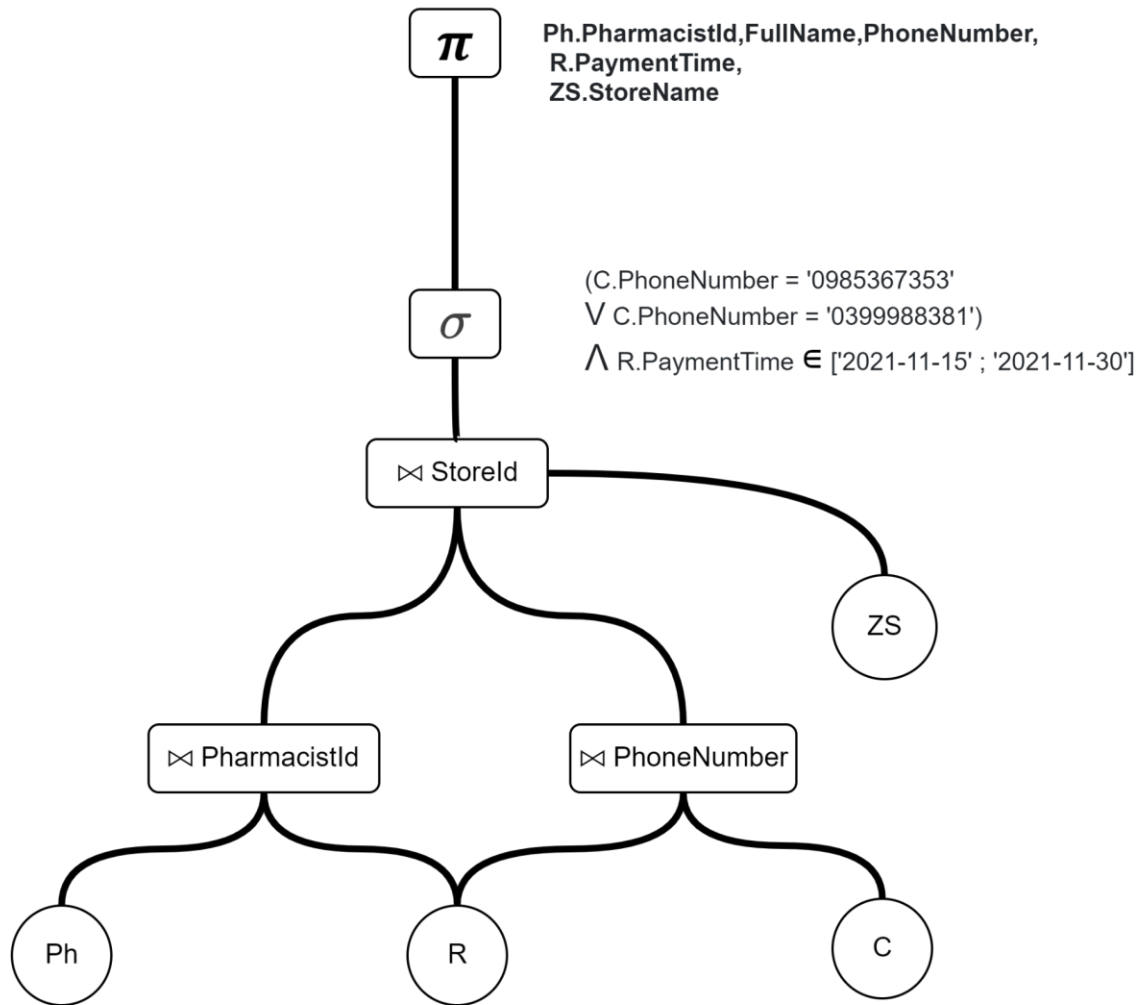
Kiểm tra câu truy vấn Q viết đúng ngữ nghĩa hay không?



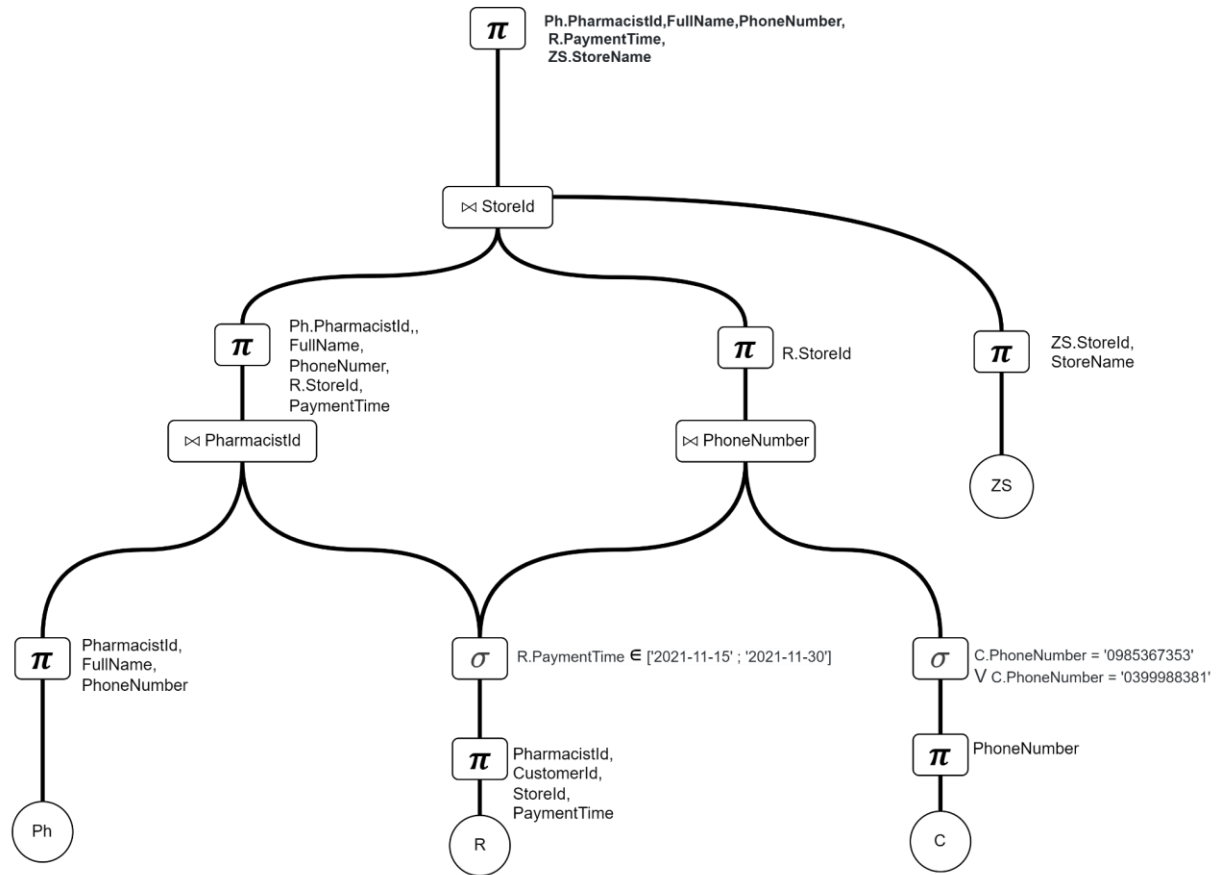
*Đồ thị truy vấn liên  
thông nên câu truy vấn  
Q đúng ngữ nghĩa*

## 2. Tối ưu hoá tập trung

*Cây truy vấn đại số quan hệ ban đầu của Q*



*Câu truy vấn đại số quan hệ sau khi tối ưu hoá toàn cục*



**Câu truy vấn đã tối ưu hoá**

```
SELECT
    RPh.PharmacistId, RPh.FullName, RPh.PhoneNumber,
    RPh.PaymentTime,
    ZS.StoreName
FROM
    (
        SELECT StoreId, StoreName FROM ZEHNSTORE
    ) ZS,
    (
        SELECT R.StoreId
        FROM
            (
                SELECT
                    PharmacistId, CustomerId, StoreId, PaymentTime
                FROM RECEIPT
                WHERE PaymentTime BETWEEN ('2021-11-15')
                    AND ('2021-11-30')
            ) R,
        (
            SELECT C.PhoneNumber
            FROM CUSTOMER C
            WHERE C.PhoneNumber = '0985367353'
            AND V C.PhoneNumber = '0399988381'
        ) C
    ) C
```

```

    ) R,
    (
        SELECT PhoneNumber FROM CUSTOMER
        WHERE PhoneNumber = '0985367353'
        OR PhoneNumber = '0399988381'
    ) C
    WHERE C.PhoneNumber = R.CustomerId
) RC,
(
    SELECT
        Ph.PharmacistId, FullName, PhoneNumber,
        R.StoreId, PaymentTime
    FROM
        (
            SELECT
                PharmacistId, CustomerId, StoreId, PaymentTime
            FROM RECEIPT
            WHERE PaymentTime BETWEEN ('2021-11-15')
                AND ('2021-11-30')
        ) R,
        (
            SELECT PharmacistId, FullName, PhoneNumber
            FROM PHARMACIST
        ) Ph
    WHERE Ph.PharmacistId = R.PharmacistId
) RPh
WHERE
    RPh.StoreId = RC.StoreId AND RPh.StoreId = ZS.StoreId;

```

## 3. Explain Plan

### 3.1. Câu truy vấn ban đầu

14	Id	Operation	Name	Starts	E-Rows	A-Rows	A-Time	Buffers	Reads	OMem	lMem	Used-Mem
15	-----											
16	0	SELECT STATEMENT		1		11	00:00:00.01	22	19			
17	* 1	FILTER		1		11	00:00:00.01	22	19			
18	* 2	HASH JOIN		1	1	11	00:00:00.01	22	19	995K	995K	842K (0)
19	* 3	HASH JOIN		1	1	11	00:00:00.01	16	13	1399K	1399K	475K (0)
20	4	NESTED LOOPS		1	1	11	00:00:00.01	10	7			
21	* 5	TABLE ACCESS FULL	RECEIPT	1	1	11	00:00:00.01	6	6			
22	* 6	INDEX UNIQUE SCAN	PK_CUSTOMER	11	1	11	00:00:00.01	4	1			
23	7	TABLE ACCESS FULL	ZEHNSTORE	1	1	1	00:00:00.01	6	6			
24	8	TABLE ACCESS FULL	PHARMACIST	1	1	22	00:00:00.01	6	6			
25	-----											
27	Predicate Information (identified by operation id):											
28	-----											
29												
30	1 -	filter(TO_DATE('2021-11-30')>=TO_DATE('2021-11-15'))										
31	2 -	access("PH"."PHARMACISTID"="R"."PHARMACISTID")										
32	3 -	access("R"."STOREID"="ZS"."STOREID")										
33	5 -	filter(((INTERNAL_FUNCTION("R"."CUSTOMERID") AND "R"."PAYMENTTIME">='2021-11-15' AND										
34		"R"."PAYMENTTIME"<='2021-11-30'))										
35	6 -	access("R"."CUSTOMERID"="C"."PHONENUMBER")										
36		filter(("C"."PHONENUMBER"='0399988381' OR "C"."PHONENUMBER"='0985367353'))										
37												
38	-----											

### 3.2. Câu truy vấn tối ưu hoá

21	-----												
22	Id	Operation	Name	Starts	E-Rows	A-Rows	A-Time	Buffers	Reads	OMem	lMem	Used-Mem	
23	-----												
24	0	SELECT STATEMENT		1		50	00:00:00.01	27	19				
25	* 1	FILTER		1		50	00:00:00.01	27	19				
26	* 2	HASH JOIN		1	1	50	00:00:00.01	27	19	995K	995K	833K (0)	
27	3	NESTED LOOPS		1	1	121	00:00:00.01	22	13				
28	* 4	HASH JOIN		1	1	121	00:00:00.01	18	12	1196K	1196K	449K (0)	
29	* 5	HASH JOIN		1	1	121	00:00:00.01	12	6	1538K	1538K	790K (0)	
30	* 6	TABLE ACCESS FULL	RECEIPT	1	1	11	00:00:00.01	6	6				
31	* 7	TABLE ACCESS FULL	RECEIPT	1	2	11	00:00:00.01	6	0				
32	8	TABLE ACCESS FULL	ZEHNSTORE	1	1	1	00:00:00.01	6	6				
33	* 9	INDEX UNIQUE SCAN	PK_CUSTOMER	121	1	121	00:00:00.01	4	1				
34	10	TABLE ACCESS FULL	PHARMACIST	1	1	2	00:00:00.01	5	6				
35	-----												
36													
40	1 -	filter(TO_DATE('2021-11-30')>=TO_DATE('2021-11-15'))											
41	2 -	access("PHARMACISTID"="PHARMACISTID")											
42	4 -	access("STOREID"="STOREID")											
43	5 -	access("STOREID"="STOREID")											
44	6 -	filter((INTERNAL_FUNCTION("CUSTOMERID") AND "PAYMENTTIME">='2021-11-15' AND "PAYMENTTIME"<='2021-11-30'))											
45	7 -	filter(("PAYMENTTIME">='2021-11-15' AND "PAYMENTTIME"<='2021-11-30'))											
46	9 -	access("PHONENUMBER"="CUSTOMERID")											
47		filter(("PHONENUMBER"='0399988381' OR "PHONENUMBER"='0985367353'))											
48													



## 4. Câu truy vấn tối ưu hoá trên môi trường phân tán

*Tại máy ZEHN02. Tương tự còn lại với máy ZEHN01*

```
SELECT DISTINCT
    RPh.PharmacistId, RPh.FullName, RPh.PhoneNumber,
    RPh.PaymentTime,
    ZS.StoreName
FROM
    (
        SELECT StoreId, StoreName
        FROM zehn_02.ZEHNSTORE
        UNION
        SELECT StoreId, StoreName
        FROM zehn_01.ZEHNSTORE@manager_02_01
    ) ZS,
    (
        SELECT R.StoreId
        FROM
            (
                SELECT PharmacistId, CustomerId, StoreId, PaymentTime
                FROM zehn_02.RECEIPT
                WHERE PaymentTime BETWEEN ('2021-11-15')
                    AND ('2021-11-30')
                UNION
                SELECT PharmacistId, CustomerId, StoreId, PaymentTime
                FROM zehn_01.RECEIPT@manager_02_01
                WHERE PaymentTime BETWEEN ('2021-11-15')
                    AND ('2021-11-30')
            ) R,
        (
            SELECT PhoneNumber
            FROM zehn_02.CUSTOMER
            WHERE PhoneNumber = '0985367353'
                OR PhoneNumber = '0399988381'
        ) C
        WHERE C.PhoneNumber = R.CustomerId
    ) RC,
    (
        SELECT
            Ph.PharmacistId, FullName, PhoneNumber,
            R.StoreId, PaymentTime
        FROM
            (
                SELECT PharmacistId, CustomerId, StoreId, PaymentTime
                FROM zehn_02.RECEIPT
```

```

        WHERE PaymentTime BETWEEN ('2021-11-15') AND ('2021-11-
30')
        UNION
        SELECT PharmacistId, CustomerId, StoreId, PaymentTime
        FROM zehn_01.RECEIPT@manager_02_01
        WHERE PaymentTime BETWEEN ('2021-11-15')
            AND ('2021-11-30')
    ) R,
    (
        SELECT PharmacistId, FullName, PhoneNumber
        FROM zehn_02.PHARMACIST
        UNION
        SELECT PharmacistId, FullName, PhoneNumber
        FROM zehn_01.PHARMACIST@manager_02_01
    ) Ph
    WHERE Ph.PharmacistId = R.PharmacistId
) RPh
WHERE
    RPh.StoreId = RC.StoreId AND RPh.StoreId = ZS.StoreId;

```

*Phụ lục:*

## **Các đường dẫn liên quan**

Github: <https://github.com/tori4582/phantan> IS221.M11 10

Video báo cáo: [https://uithcm.sharepoint.com/sites/IS211-Zehn/Shared%20Documents/General/Recordings/Meeting%20in%20General - 20211212 13 5116-Meeting%20Recording.mp4?web=1](https://uithcm.sharepoint.com/sites/IS211-Zehn/Shared%20Documents/General/Recordings/Meeting%20in%20General-20211212%2013-5116-Meeting%20Recording.mp4?web=1)