

ĐẠI HỌC QUỐC GIA TP HCM
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA HỆ THỐNG THÔNG TIN



BÀI TẬP LỚN 01

Đề tài:

THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN QUẢN LÝ QUÁN CÀ PHÊ

Môn học: Cơ sở dữ liệu phân tán

Lớp: IS211.O11

Giảng viên hướng dẫn: Lê Đức Tín

Nhóm sinh viên thực hiện: Nhóm 09

STT	Họ và tên	Mã số sinh viên
1	Lê Minh Nguyệt	21521211
2	Trần Hoài Bảo	21520628
3	Nguyễn Hải Linh	19521753
4	Nguyễn Thừa An Thái	19522192

MỤC LỤC

I. THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN	4
1. Lược đồ cơ sở dữ liệu toàn cục.....	4
2. Kiến trúc phân mảnh	5
2.1. Mô tả phân mảnh.....	5
2.2. Phân mảnh trên hai chi nhánh	5
3. Xây dựng cơ sở dữ liệu phân tán.....	6
3.1. DDL.....	6
3.2. DML	10
4. Kiến trúc phân quyền.....	17
4.1. Mô tả kiến trúc phân quyền.....	17
4.2. Phân quyền trên cơ sở dữ liệu phân tán	18
5. Thực hiện 10 câu truy vấn	20
5.1. Câu truy vấn 1	20
5.2. Câu truy vấn 2	20
5.3. Câu truy vấn 3	21
5.4. Câu truy vấn 4	22
5.5. Câu truy vấn 5	23
5.6. Câu truy vấn 6	23
5.7. Câu truy vấn 7	24
5.8. Câu truy vấn 8	25
5.9. Câu truy vấn 9	25
5.10. Câu truy vấn 10	26
II. VIẾT HÀM (THỦ TỤC) VÀ RÀNG BUỘC TOÀN VỆN	27
1. Hàm (Function).....	27
2. Ràng buộc toàn vẹn (Trigger).....	28
III. CÁC MỨC CÔ LẬP TRONG MÔI TRƯỜNG PHÂN TÁN	30
1. Trường hợp Lost Update.....	30
1.1. Mô tả tình huống	30
1.2. Minh họa	31

1.3. Giải pháp	32
2. Trường hợp Unrepeatable Read.....	32
2.1. Mô tả tình huống	32
2.2. Minh họa	32
2.3. Giải pháp	33
3. Trường hợp Phantom Read	33
3.1. Mô tả tình huống	33
3.2. Minh họa	33
3.3. Giải pháp	34
4. Trường hợp Deadlock.....	34
4.1. Mô tả tình huống	34
4.2. Minh họa	34
4.3. Giải pháp	35
IV. TỐI ƯU HÓA TRUY VẤN TRÊN MÔI TRƯỜNG PHÂN TÁN	35
1. Câu truy vấn đơn giản chưa tối ưu	35
2. Explain câu truy vấn đơn giản	35
3. Tối ưu hóa câu truy vấn cục bộ, phân tán.....	37
4. Viết lại câu truy vấn trên môi trường phân tán và nhận xét.....	39

DANH MỤC HÌNH ẢNH

Hình 1. Kết quả câu truy vấn 1	20
Hình 2. Kết quả câu truy vấn 2	21
Hình 3. Kết quả câu truy vấn 3	22
Hình 4. Kết quả câu truy vấn 4	22
Hình 5. Kết quả câu truy vấn 5	23
Hình 6. Kết quả câu truy vấn 6	24
Hình 7. Kết quả câu truy vấn 7	25
Hình 8. Kết quả câu truy vấn 8	25
Hình 9. Kết quả câu truy vấn 9	26
Hình 10. Kết quả câu truy vấn 10	27
Hình 11. Kết quả câu truy vấn kiểm thử Function	28
Hình 12. Dữ liệu khách hàng ‘CU01’ trước khi thêm hóa đơn	30
Hình 13. Dữ liệu khách hàng ‘CU01’ sau khi thêm hóa đơn và áp dụng Trigger	30
Hình 14. Kết quả explain câu truy vấn chưa tối ưu hóa cục bộ	37
Hình 15. Đồ thị truy vấn	37
Hình 16. Cây truy vấn tối ưu trên môi trường tập trung	38
Hình 17. Cây truy vấn tối ưu trên môi trường phân tán	39
Hình 18. Kết quả explain câu truy vấn rút gọn trên môi trường phân tán	40

I. THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

1. Lược đồ cơ sở dữ liệu toàn cục

Chuỗi quán Cà phê có 2 chi nhánh tại đường Phan Văn Trị (quận Bình Thạnh) và đường Nguyễn Du (quận 1), TP.HCM. Trong đó, cơ sở chính của quán nằm tại chi nhánh ‘Phan Văn Trị’. Lược đồ cơ sở dữ liệu toàn cục của Chuỗi quán Cà phê như sau.

BRANCH (BranchID, BranchName, BranchAddress, BranchPhone)

Tên từ: Mỗi chi nhánh có mã chi nhánh (BranchID) dùng để phân biệt các chi nhánh, ngoài ra còn lưu tên chi nhánh (BranchName), địa chỉ (BranchAddress) và số điện thoại của chi nhánh (BranchPhone). Trường tên chi nhánh là duy nhất.

EMPLOYEE (EmployeeID, EmployeeName, Birthday, PhoneNumber, EmployeeAddress, BranchID)

Tên từ: Mỗi nhân viên có mã nhân viên (EmployeeID) là duy nhất, tên nhân viên (EmployeeName), ngày sinh (Birthday), số điện thoại (PhoneNumber), địa chỉ (EmployeeAddress), và chi nhánh mà nhân viên đó làm việc (BranchID).

CUSTOMER (CustomerID, CustomerName, PhoneNumber, CustomerAddress, Birthday, CustomerType, CumulativeTotal)

Tên từ: Mỗi khách hàng có mã khách hàng (CustomerID) là duy nhất, tên khách hàng (CustomerName), số điện thoại (PhoneNumber), địa chỉ (CustomerAddress), ngày sinh (Birthday), loại khách hàng (CustomerType), và tổng số tiền khách hàng đã thanh toán tại các chi nhánh của cửa hàng (CumulativeTotal). Khách hàng có số tiền tích lũy dưới 300.000 là loại ‘Standard’, từ 300.000 đến 500.000 là loại ‘Member’, và trên 500.000 là loại ‘VIP’.

ITEM (ItemID, ItemName, ItemType, SalePrice)

Tên từ: Mỗi món ăn/thức uống trong menu có mã món (ItemID) là duy nhất, tên món (ItemName), loại món (ItemType), và giá bán (SalePrice).

ITEMMANAGE (BranchID, ItemID, SaleStatus, Discount, ServingTime, Availability)

Tên từ: Mỗi món trong menu tại mỗi chi nhánh có trạng thái bán (SaleStatus), phần trăm giảm giá (Discount), thời gian phục vụ món (ServingTime), và số lượng có thể bán trong ngày (Availability).

BILL (BillID, BillDate, CustomerID, EmployeeID, Total)

Tên từ: Mỗi hóa đơn bán hàng có mã hóa đơn (BillID) là duy nhất, ngày lập hóa đơn (BillDate), mã khách hàng thanh toán hóa đơn (CustomerID), mã nhân viên chịu trách nhiệm xử lý hóa đơn (EmployeeID), và tổng số tiền thanh toán (Total).

BILLDETAIL (BillID, ItemID, Quantity, SubTotal)

Tên từ: Chi tiết hóa đơn cho mỗi mục trong hóa đơn, gồm có mã hóa đơn (BillID), mã món (ItemID), số lượng của món (Quantity), và thành tiền cho món đó (SubTotal).

2. Kiến trúc phân mảnh

2.1. Mô tả phân mảnh

- Quan hệ BRANCH phân mảnh ngang chính
- Quan hệ EMPLOYEE phân mảnh ngang dẫn xuất theo quan hệ BRANCH
- Quan hệ BILL phân mảnh ngang dẫn xuất theo quan hệ EMPLOYEE
- Quan hệ BILLDETAIL phân mảnh ngang dẫn xuất theo quan hệ BILL
- Quan hệ CUSTOMER phân mảnh dọc và được nhân bản ở tất cả các chi nhánh:
 - **CUSTOMER_STAFF** (CustomerID, CustomerName, PhoneNumber, CustomerAddress, Birthday, CustomerType)
 - **CUSTOMER_MANAGER** (CustomerID, CumulativeTotal)
- Quan hệ ITEMMANAGE phân mảnh hỗn hợp. Trong đó, mỗi chi nhánh có hai quan hệ:
 - **ITEMMANAGE_MANAGER** (BranchID, ItemID, SaleStatus, Discount)
 - **ITEMMANAGE_STAFF** (BranchID, ItemID, ServingTime, Availability)
- Quan hệ CUSTOMER_MANAGER, CUSTOMER_STAFF, ITEM nhân bản tại tất cả các chi nhánh

2.2. Phân mảnh trên hai chi nhánh

CN1.BRANCH = $\sigma_{BranchName="Phan Van Tri"}(BRANCH)$

CN2.BRANCH = $\sigma_{BranchName="Nguyen Du"}(BRANCH)$

CN1.EMPLOYEE = EMPLOYEE $\bowtie_{BranchID}$ (CN1.BRANCH)

CN2.EMPLOYEE = EMPLOYEE $\bowtie_{BranchID}$ (CN2.BRANCH)

CN1.BILL = BILL $\bowtie_{EmployeeID}$ (CN1.EMPLOYEE)

CN2.BILL = BILL $\bowtie_{EmployeeID}$ (CN2.EMPLOYEE)

$CN1.BILLDETAIL = BILLDETAIL \bowtie_{BillID} (CN1. BILL)$

$CN2.BILLDETAIL = BILLDETAIL \bowtie_{BillID} (CN2. BILL)$

$CUSTOMER_MANAGER = \pi_{CustomerID, CumulativeTotal} (CUSTOMER)$

$CUSTOMER_STAFF =$

$\pi_{CustomerID, CustomerName, PhoneNumber, CustomerAddress, Birthday, CustomerType} (CUSTOMER)$

$CN1.ITEMMANAGE_MANAGER = \pi_{BranchID, ItemID, SaleStatus, Discount} (ITEMMANAGE \bowtie_{BranchID} CN1.BRANCH)$

$CN2.ITEMMANAGE_MANAGER = \pi_{BranchID, ItemID, SaleStatus, Discount} (ITEMMANAGE \bowtie_{BranchID} CN2.BRANCH)$

$CN1.ITEMMANAGE_STAFF = \pi_{BranchID, ItemID, ServingTime, Availability} (ITEMMANAGE \bowtie_{BranchID} CN1.BRANCH)$

$CN2.ITEMMANAGE_STAFF = \pi_{BranchID, ItemID, SaleServingTime, Availability} (ITEMMANAGE \bowtie_{BranchID} CN2.BRANCH)$

Quan hệ CUSTOMER_MANAGER, CUSTOMER_STAFF, ITEM nhân bản tại tất cả các chi nhánh.

3. Xây dựng cơ sở dữ liệu phân tán

3.1. DDL

CN1	CN2
CREATE TABLE CN1.BRANCH (BranchID VARCHAR2(5) NOT NULL, BranchName VARCHAR2(50) NOT NULL, BranchAddress VARCHAR2(100) NULL,	CREATE TABLE CN2.BRANCH (BranchID VARCHAR2(5) NOT NULL, BranchName VARCHAR2(50) NOT NULL, BranchAddress VARCHAR2(100) NULL,

BranchPhone VARCHAR2(10) NULL, CONSTRAINT PK_BRANCH PRIMARY KEY (BranchID));	BranchPhone VARCHAR2(10) NULL, CONSTRAINT PK_BRANCH PRIMARY KEY (BranchID));
CREATE TABLE CN1.EMPLOYEE (EmployeeID VARCHAR2(5) NOT NULL, EmployeeName VARCHAR2(50) NOT NULL, Birthday DATE NULL, PhoneNumber VARCHAR2(10) NULL, EmployeeAddress VARCHAR2(50) NULL, BranchID VARCHAR2(5) NOT NULL, CONSTRAINT PK_EMPLOYEE PRIMARY KEY (EmployeeID)); ALTER TABLE CN1.EMPLOYEE ADD CONSTRAINT FK_EMPLOYEE_BRANCH FOREIGN KEY (BranchID) REFERENCES CN1.BRANCH(BranchID);	CREATE TABLE CN2.EMPLOYEE (EmployeeID VARCHAR2(5) NOT NULL, EmployeeName VARCHAR2(50) NOT NULL, Birthday DATE NULL, PhoneNumber VARCHAR2(10) NULL, EmployeeAddress VARCHAR2(50) NULL, BranchID VARCHAR2(5) NOT NULL, CONSTRAINT PK_EMPLOYEE PRIMARY KEY (EmployeeID)); ALTER TABLE CN2.EMPLOYEE ADD CONSTRAINT FK_EMPLOYEE_BRANCH FOREIGN KEY (BranchID) REFERENCES CN2.BRANCH(BranchID);
CREATE TABLE CN1.CUSTOMER_STAFF (CustomerID VARCHAR2(5) NOT NULL, CustomerName VARCHAR2(50) NOT NULL, PhoneNumber VARCHAR2(10) NULL, CustomerAddress VARCHAR2(50) NULL, Birthday DATE NULL, CustomerType VARCHAR2(25) NULL, CONSTRAINT PK_CUSTOMERSTAFF PRIMARY KEY (CustomerID));	CREATE TABLE CN2.CUSTOMER_STAFF (CustomerID VARCHAR2(5) NOT NULL, CustomerName VARCHAR2(50) NOT NULL, PhoneNumber VARCHAR2(10) NULL, CustomerAddress VARCHAR2(50) NULL, Birthday DATE NULL, CustomerType VARCHAR2(25) NULL, CONSTRAINT PK_CUSTOMERSTAFF PRIMARY KEY (CustomerID));
CREATE TABLE CN1.CUSTOMER_MANAGER (CustomerID VARCHAR2(5) NOT NULL, CumulativeTotal NUMBER,	CREATE TABLE CN2.CUSTOMER_MANAGER (CustomerID VARCHAR2(5) NOT NULL, CumulativeTotal NUMBER,

CONSTRAINT PK_CUSTOMERMANAGER PRIMARY KEY (CustomerID));	CONSTRAINT PK_CUSTOMERMANAGER PRIMARY KEY (CustomerID));
CREATE TABLE CN1.ITEM (ItemID VARCHAR2(5) NOT NULL, ItemName VARCHAR2(50) NOT NULL, ItemType VARCHAR2(50) NULL, SalePrice NUMBER NULL, CONSTRAINT PK_ITEM PRIMARY KEY (ItemID));	CREATE TABLE CN2.ITEM (ItemID VARCHAR2(5) NOT NULL, ItemName VARCHAR2(50) NOT NULL, ItemType VARCHAR2(50) NULL, SalePrice NUMBER NULL, CONSTRAINT PK_ITEM PRIMARY KEY (ItemID));
CREATE TABLE CN1.ITEMMANAGE_MANAGER (BranchID VARCHAR2(5) NOT NULL, ItemID VARCHAR2(5) NOT NULL, SaleStatus VARCHAR2(25) NULL, Discount NUMBER NULL, CONSTRAINT PK_ITEMMANAGER PRIMARY KEY (BranchID, ItemID)); ALTER TABLE CN1.ITEMMANAGE_MANAGER ADD CONSTRAINT FK_ITEMMANAGER_BRANCH FOREIGN KEY (BranchID) REFERENCES CN1.BRANCH(BranchID); ALTER TABLE CN1.ITEMMANAGE_MANAGER ADD CONSTRAINT FK_ITEMMANAGER_ITEM FOREIGN KEY (ItemID) REFERENCES CN1.ITEM(ItemID);	CREATE TABLE CN2.ITEMMANAGE_MANAGER (BranchID VARCHAR2(5) NOT NULL, ItemID VARCHAR2(5) NOT NULL, SaleStatus VARCHAR2(25) NULL, Discount NUMBER NULL, CONSTRAINT PK_ITEMMANAGER PRIMARY KEY (BranchID, ItemID)); ALTER TABLE CN2.ITEMMANAGE_MANAGER ADD CONSTRAINT FK_ITEMMANAGER_BRANCH FOREIGN KEY (BranchID) REFERENCES CN2.BRANCH(BranchID); ALTER TABLE CN2.ITEMMANAGE_MANAGER ADD CONSTRAINT FK_ITEMMANAGER_ITEM FOREIGN KEY (ItemID) REFERENCES CN2.ITEM(ItemID);
CREATE TABLE CN1.ITEMMANAGE_STAFF (BranchID VARCHAR2(5) NOT NULL, ItemID VARCHAR2(5) NOT NULL, ServingTime VARCHAR2(25) NULL, Availability NUMBER NULL,	CREATE TABLE CN2.ITEMMANAGE_STAFF (BranchID VARCHAR2(5) NOT NULL, ItemID VARCHAR2(5) NOT NULL, ServingTime VARCHAR2(25) NULL, Availability NUMBER NULL,

<p>CONSTRAINT PK_ITEMSTAFF PRIMARY KEY (BranchID, ItemID)); ALTER TABLE CN1.ITEMMANAGE_STAFF ADD CONSTRAINT FK_ITEMSTAFF_BRANCH FOREIGN KEY (BranchID) REFERENCES CN1.BRANCH(BranchID); ALTER TABLE CN1.ITEMMANAGE_STAFF ADD CONSTRAINT FK_ITEMSTAFF_ITEM FOREIGN KEY (ItemID) REFERENCES CN1.ITEM(ItemID);</p>	<p>CONSTRAINT PK_ITEMSTAFF PRIMARY KEY (BranchID, ItemID)); ALTER TABLE CN2.ITEMMANAGE_STAFF ADD CONSTRAINT FK_ITEMSTAFF_BRANCH FOREIGN KEY (BranchID) REFERENCES CN2.BRANCH(BranchID); ALTER TABLE CN2.ITEMMANAGE_STAFF ADD CONSTRAINT FK_ITEMSTAFF_ITEM FOREIGN KEY (ItemID) REFERENCES CN2.ITEM(ItemID);</p>
<p>CREATE TABLE CN1.BILL (BillID VARCHAR2(5) NOT NULL, BillDate DATE NOT NULL, CustomerID VARCHAR2(5) NOT NULL, EmployeeID VARCHAR2(5) NOT NULL, Total NUMBER NULL, CONSTRAINT PK_INVOICE PRIMARY KEY (BillID)); ALTER TABLE CN1.BILL ADD CONSTRAINT FK_BILL_CUSTOMER FOREIGN KEY (CustomerID) REFERENCES CN1.CUSTOMER_STAFF(CustomerID); ALTER TABLE CN1.BILL ADD CONSTRAINT FK_BILL_EMPLOYEE FOREIGN KEY (EmployeeID) REFERENCES CN1.EMPLOYEE(EmployeeID);</p>	<p>CREATE TABLE CN2.BILL (BillID VARCHAR2(5) NOT NULL, BillDate DATE NOT NULL, CustomerID VARCHAR2(5) NOT NULL, EmployeeID VARCHAR2(5) NOT NULL, Total NUMBER NULL, CONSTRAINT PK_INVOICE PRIMARY KEY (BillID)); ALTER TABLE CN2.BILL ADD CONSTRAINT FK_BILL_CUSTOMER FOREIGN KEY (CustomerID) REFERENCES CN2.CUSTOMER_STAFF(CustomerID); ALTER TABLE CN2.BILL ADD CONSTRAINT FK_BILL_EMPLOYEE FOREIGN KEY (EmployeeID) REFERENCES CN2.EMPLOYEE(EmployeeID);</p>
<p>CREATE TABLE CN1.BILLDETAIL (BillID VARCHAR2(5) NOT NULL, ItemID VARCHAR2(5) NOT NULL, Quantity NUMBER NULL, SubTotal NUMBER NULL,</p>	<p>CREATE TABLE CN2.BILLDETAIL (BillID VARCHAR2(5) NOT NULL, ItemID VARCHAR2(5) NOT NULL, Quantity NUMBER NULL, SubTotal NUMBER NULL,</p>

CONSTRAINT PK_BILLDETAIL PRIMARY KEY (BillID, ItemID)); ALTER TABLE CN1.BILLDETAIL ADD CONSTRAINT FK_BILLDETAIL_BILL FOREIGN KEY (BillID) REFERENCES CN1.BILL(BillID); ALTER TABLE CN1.BILLDETAIL ADD CONSTRAINT FK_BILLDETAIL_ITEM FOREIGN KEY (ItemID) REFERENCES CN1.ITEM(ItemID);	CONSTRAINT PK_BILLDETAIL PRIMARY KEY (BillID, ItemID)); ALTER TABLE CN2.BILLDETAIL ADD CONSTRAINT FK_BILLDETAIL_BILL FOREIGN KEY (BillID) REFERENCES CN2.BILL(BillID); ALTER TABLE CN2.BILLDETAIL ADD CONSTRAINT FK_BILLDETAIL_ITEM FOREIGN KEY (ItemID) REFERENCES CN2.ITEM(ItemID);
---	---

3.2. DML

Dưới đây là một phần DML xây dựng cơ sở dữ liệu:

CN1	CN2
INSERT INTO CN1.BRANCH VALUES ('BR01','Phan Van Tri','190C Phan Van Tri, Phuong 12, Quan Binh Thanh','0938223929');	INSERT INTO CN2.BRANCH VALUES ('BR02','Nguyen Du','53G Nguyen Du, Phuong Ben Nghe, Quan 1','0979159599');
INSERT INTO CN1.ITEM VALUES ('IT01','Sinh To Dau','Sinh To',20000); INSERT INTO CN1.ITEM VALUES ('IT02','Sinh To Bo','Sinh To',15000); INSERT INTO CN1.ITEM VALUES ('IT03','Sinh To Mang Cau','Sinh To',18000); INSERT INTO CN1.ITEM VALUES ('IT04','Sinh To Sapoché','Sinh To',15000); INSERT INTO CN1.ITEM VALUES ('IT05','Ca Phe Den Nong/Da','Ca Phe Pha Phin',24000); INSERT INTO CN1.ITEM VALUES ('IT06','Ca Phe Nau Nong/Da','Ca Phe Pha Phin',28000); INSERT INTO CN1.ITEM VALUES ('IT07','Ca Phe Sua Tuoi Nong/Da','Ca Phe Pha Phin',32000); INSERT INTO CN1.ITEM VALUES ('IT08','Ca Phe Sua Bac Ha','Ca Phe Pha Phin',34000);	INSERT INTO CN2.ITEM VALUES ('IT01','Sinh To Dau','Sinh To',20000); INSERT INTO CN2.ITEM VALUES ('IT02','Sinh To Bo','Sinh To',15000); INSERT INTO CN2.ITEM VALUES ('IT03','Sinh To Mang Cau','Sinh To',18000); INSERT INTO CN2.ITEM VALUES ('IT04','Sinh To Sapoché','Sinh To',15000); INSERT INTO CN2.ITEM VALUES ('IT05','Ca Phe Den Nong/Da','Ca Phe Pha Phin',24000); INSERT INTO CN2.ITEM VALUES ('IT06','Ca Phe Nau Nong/Da','Ca Phe Pha Phin',28000); INSERT INTO CN2.ITEM VALUES ('IT07','Ca Phe Sua Tuoi Nong/Da','Ca Phe Pha Phin',32000); INSERT INTO CN2.ITEM VALUES ('IT08','Ca Phe Sua Bac Ha','Ca Phe Pha Phin',34000);

INSERT INTO CN1.ITEM VALUES ('IT09','Ca Phe Trung','Ca Phe Pha Phin',70000); INSERT INTO CN1.ITEM VALUES ('IT10','Latte Macchiato','Ca Phe Italy',45000);	INSERT INTO CN2.ITEM VALUES ('IT09','Ca Phe Trung','Ca Phe Pha Phin',70000); INSERT INTO CN2.ITEM VALUES ('IT10','Latte Macchiato','Ca Phe Italy',45000);
INSERT INTO CN1.CUSTOMER_STAFF VALUES ('CU01','Bui Ngoc Bao Thy','0375130491','Ho Chi Minh',to_date('28/06/2000','dd/mm/yyyy'),'St ardard'); INSERT INTO CN1.CUSTOMER_STAFF VALUES ('CU02','Vo Ngoc Hoai Thuong','0901006848','Binh Dinh',to_date('07/06/2001','dd/mm/yyyy'),'St ardard'); INSERT INTO CN1.CUSTOMER_STAFF VALUES ('CU03','Nguyen Ngoc Hai Yen','0931127827','Ho Chi Minh',to_date('19/08/1998','dd/mm/yyyy'),'St ardard'); INSERT INTO CN1.CUSTOMER_STAFF VALUES ('CU04','Nguyen Huu Khac Phuc','0931199244','Quang Nam',to_date('28/06/2001','dd/mm/yyyy'),'Sta rdard'); INSERT INTO CN1.CUSTOMER_STAFF VALUES ('CU05','Nguyen Ba Hoc','0931199844','Da Lat',to_date('08/07/2000','dd/mm/yyyy'),'Star dard'); INSERT INTO CN1.CUSTOMER_STAFF VALUES ('CU06','Cao Huy','0901006025','Sa Pa',to_date('18/12/2004','dd/mm/yyyy'),'Stard ard'); INSERT INTO CN1.CUSTOMER_STAFF VALUES ('CU07','Ngo Le Thao Dung','0931900611','Nha	INSERT INTO CN2.CUSTOMER_STAFF VALUES ('CU01','Bui Ngoc Bao Thy','0375130491','Ho Chi Minh',to_date('28/06/2000','dd/mm/yyyy'),'St ardard'); INSERT INTO CN2.CUSTOMER_STAFF VALUES ('CU02','Vo Ngoc Hoai Thuong','0901006848','Binh Dinh',to_date('07/06/2001','dd/mm/yyyy'),'St ardard'); INSERT INTO CN2.CUSTOMER_STAFF VALUES ('CU03','Nguyen Ngoc Hai Yen','0931127827','Ho Chi Minh',to_date('19/08/1998','dd/mm/yyyy'),'St ardard'); INSERT INTO CN2.CUSTOMER_STAFF VALUES ('CU04','Nguyen Huu Khac Phuc','0931199244','Quang Nam',to_date('28/06/2001','dd/mm/yyyy'),'Sta rdard'); INSERT INTO CN2.CUSTOMER_STAFF VALUES ('CU05','Nguyen Ba Hoc','0931199844','Da Lat',to_date('08/07/2000','dd/mm/yyyy'),'Star dard'); INSERT INTO CN2.CUSTOMER_STAFF VALUES ('CU06','Cao Huy','0901006025','Sa Pa',to_date('18/12/2004','dd/mm/yyyy'),'Stard ard'); INSERT INTO CN2.CUSTOMER_STAFF VALUES ('CU07','Ngo Le Thao Dung','0931900611','Nha

Trang',to_date('12/12/2005','dd/mm/yyyy'),'Standard'); INSERT INTO CN1.CUSTOMER_STAFF VALUES ('CU08','Nguyen Minh Toan','0904391156','Ha Noi',to_date('12/12/2005','dd/mm/yyyy'),'Standard'); INSERT INTO CN1.CUSTOMER_STAFF VALUES ('CU09','Le Huu Nhat Khoa','0375130491','Quang Nam',to_date('02/08/1990','dd/mm/yyyy'),'Member'); INSERT INTO CN1.CUSTOMER_STAFF VALUES ('CU10','Nguyen Trong Kien','0901006848','Gia Lai',to_date('23/09/1997','dd/mm/yyyy'),'VIP');	Trang',to_date('12/12/2005','dd/mm/yyyy'),'Standard'); INSERT INTO CN2.CUSTOMER_STAFF VALUES ('CU08','Nguyen Minh Toan','0904391156','Ha Noi',to_date('12/12/2005','dd/mm/yyyy'),'Standard'); INSERT INTO CN2.CUSTOMER_STAFF VALUES ('CU09','Le Huu Nhat Khoa','0375130491','Quang Nam',to_date('02/08/1990','dd/mm/yyyy'),'Member'); INSERT INTO CN2.CUSTOMER_STAFF VALUES ('CU10','Nguyen Trong Kien','0901006848','Gia Lai',to_date('23/09/1997','dd/mm/yyyy'),'VIP');
INSERT INTO CN1.CUSTOMER_MANAGER VALUES ('CU01',0); INSERT INTO CN1.CUSTOMER_MANAGER VALUES ('CU02',0); INSERT INTO CN1.CUSTOMER_MANAGER VALUES ('CU03',20000); INSERT INTO CN1.CUSTOMER_MANAGER VALUES ('CU04',0); INSERT INTO CN1.CUSTOMER_MANAGER VALUES ('CU05',0); INSERT INTO CN1.CUSTOMER_MANAGER VALUES ('CU06',0); INSERT INTO CN1.CUSTOMER_MANAGER VALUES ('CU07',0);	INSERT INTO CN2.CUSTOMER_MANAGER VALUES ('CU01',0); INSERT INTO CN2.CUSTOMER_MANAGER VALUES ('CU02',0); INSERT INTO CN2.CUSTOMER_MANAGER VALUES ('CU03',20000); INSERT INTO CN2.CUSTOMER_MANAGER VALUES ('CU04',0); INSERT INTO CN2.CUSTOMER_MANAGER VALUES ('CU05',0); INSERT INTO CN2.CUSTOMER_MANAGER VALUES ('CU06',0); INSERT INTO CN2.CUSTOMER_MANAGER VALUES ('CU07',0);

INSERT INTO CN1.CUSTOMER_MANAGER VALUES ('CU08',0); INSERT INTO CN1.CUSTOMER_MANAGER VALUES ('CU09',310000); INSERT INTO CN1.CUSTOMER_MANAGER VALUES ('CU10',0);	INSERT INTO CN2.CUSTOMER_MANAGER VALUES ('CU08',0); INSERT INTO CN2.CUSTOMER_MANAGER VALUES ('CU09',310000); INSERT INTO CN2.CUSTOMER_MANAGER VALUES ('CU10',0);
INSERT INTO CN1.EMPLOYEE VALUES ('EMP01','Vu Thuy Duong',to_date('18/09/2001','dd/mm/yyyy'),'0366866701','Vinh Phuc','BR01'); INSERT INTO CN1.EMPLOYEE VALUES ('EMP02','Phan Thai Tam',to_date('28/06/2001','dd/mm/yyyy'),'0764853497','Quang Nam','BR01'); INSERT INTO CN1.EMPLOYEE VALUES ('EMP03','Nguyen Thanh Sang',to_date('01/01/2001','dd/mm/yyyy'),'0366866701','Ho Chi Minh','BR01'); INSERT INTO CN1.EMPLOYEE VALUES ('EMP04','Hoang Van Tien Nhat',to_date('01/01/2001','dd/mm/yyyy'),'0366866701','Ho Chi Minh','BR01'); INSERT INTO CN1.EMPLOYEE VALUES ('EMP05','Tran Thi Ngoc An',to_date('06/06/2001','dd/mm/yyyy'),'0366866701','Binh Dinh','BR01'); INSERT INTO CN1.EMPLOYEE VALUES ('EMP06','Nguyen Xuan Minh Thu',to_date('01/01/2001','dd/mm/yyyy'),'0366866701','Kien Giang','BR01'); INSERT INTO CN1.EMPLOYEE VALUES ('EMP07','Le Thi Hong Cuc',to_date('27/01/2001','dd/mm/yyyy'),'0366866701','Tien Giang','BR01'); INSERT INTO CN1.EMPLOYEE VALUES ('EMP08','Ta Quang	INSERT INTO CN2.EMPLOYEE VALUES ('EMP11','Nguyen Chung Thuy Dan',to_date('21/07/2001','dd/mm/yyyy'),'0366866701','Ho Chi Minh','BR02'); INSERT INTO CN2.EMPLOYEE VALUES ('EMP12','Nguyen Ngoc Linh Dan',to_date('07/07/2001','dd/mm/yyyy'),'0366866701','Nam Dinh','BR02'); INSERT INTO CN2.EMPLOYEE VALUES ('EMP13','Ho Nam Kha',to_date('01/01/2001','dd/mm/yyyy'),'0366866701','Nghe An','BR02'); INSERT INTO CN2.EMPLOYEE VALUES ('EMP14','Vu Lan Anh',to_date('01/01/2001','dd/mm/yyyy'),'0366866701','Da Lat','BR02'); INSERT INTO CN2.EMPLOYEE VALUES ('EMP15','Bui Nhat Linh',to_date('01/01/2001','dd/mm/yyyy'),'0366866701','Dong Nai','BR02'); INSERT INTO CN2.EMPLOYEE VALUES ('EMP16','Nguyen Do Thuy Tran',to_date('01/01/2001','dd/mm/yyyy'),'0366866701','Da Nang','BR02'); INSERT INTO CN2.EMPLOYEE VALUES ('EMP17','Ngo Ngoc Quynh Hoa',to_date('18/09/2000','dd/mm/yyyy'),'0366866701','Ha Noi','BR02'); INSERT INTO CN2.EMPLOYEE VALUES ('EMP18','Phan Quynh

Huy',to_date('01/01/2001','dd/mm/yyyy'),'036 6866701','Bien Hoa','BR01'); INSERT INTO CN1.EMPLOYEE VALUES ('EMP09','Tran Thanh Trung',to_date('01/01/2001','dd/mm/yyyy'),'0 366866701','Ben Tre','BR01'); INSERT INTO CN1.EMPLOYEE VALUES ('EMP10','Luu Tran Anh Khoa',to_date('01/01/2001','dd/mm/yyyy'),'03 66866701','Long An','BR01');	Hoa',to_date('01/01/1999','dd/mm/yyyy'),'036 6866701','Vinh Phuc','BR02'); INSERT INTO CN2.EMPLOYEE VALUES ('EMP19','Dao Gia Bao',to_date('01/01/2000','dd/mm/yyyy'),'036 6866701','Phu Tho','BR02'); INSERT INTO CN2.EMPLOYEE VALUES ('EMP20','Nguyen Viet Anh',to_date('01/01/2000','dd/mm/yyyy'),'036 6866701','Ho Chi Minh','BR02');
INSERT INTO CN1.ITEMMANAGE_MANAGER VALUES ('BR01','IT01','Duoc Phep Ban',0); INSERT INTO CN1.ITEMMANAGE_MANAGER VALUES ('BR01','IT02','Duoc Phep Ban',0); INSERT INTO CN1.ITEMMANAGE_MANAGER VALUES ('BR01','IT03','Duoc Phep Ban',0); INSERT INTO CN1.ITEMMANAGE_MANAGER VALUES ('BR01','IT04','Duoc Phep Ban',0); INSERT INTO CN1.ITEMMANAGE_MANAGER VALUES ('BR01','IT05','Duoc Phep Ban',10); INSERT INTO CN1.ITEMMANAGE_MANAGER VALUES ('BR01','IT06','Duoc Phep Ban',0); INSERT INTO CN1.ITEMMANAGE_MANAGER VALUES ('BR01','IT07','Duoc Phep Ban',0); INSERT INTO CN1.ITEMMANAGE_MANAGER VALUES ('BR01','IT08','Duoc Phep Ban',0); INSERT INTO CN1.ITEMMANAGE_MANAGER VALUES ('BR01','IT09','Duoc Phep Ban',0);	INSERT INTO CN2.ITEMMANAGE_MANAGER VALUES ('BR02','IT01','Duoc Phep Ban',0); INSERT INTO CN2.ITEMMANAGE_MANAGER VALUES ('BR02','IT02','Duoc Phep Ban',0); INSERT INTO CN2.ITEMMANAGE_MANAGER VALUES ('BR02','IT03','Duoc Phep Ban',5); INSERT INTO CN2.ITEMMANAGE_MANAGER VALUES ('BR02','IT04','Duoc Phep Ban',0); INSERT INTO CN2.ITEMMANAGE_MANAGER VALUES ('BR02','IT05','Duoc Phep Ban',0); INSERT INTO CN2.ITEMMANAGE_MANAGER VALUES ('BR02','IT06','Duoc Phep Ban',0); INSERT INTO CN2.ITEMMANAGE_MANAGER VALUES ('BR02','IT07','Duoc Phep Ban',0); INSERT INTO CN2.ITEMMANAGE_MANAGER VALUES ('BR02','IT08','Duoc Phep Ban',10); INSERT INTO CN2.ITEMMANAGE_MANAGER VALUES ('BR02','IT09','Duoc Phep Ban',0);

INSERT INTO CN1.ITEMMANAGE_MANAGER VALUES ('BR01','IT10','Duoc Phep Ban',0);	INSERT INTO CN2.ITEMMANAGE_MANAGER VALUES ('BR02','IT10','Duoc Phep Ban',0);
INSERT INTO CN1.ITEMMANAGE_STAFF VALUES ('BR01','IT01','Ngay',29); INSERT INTO CN1.ITEMMANAGE_STAFF VALUES ('BR01','IT02','Sang',7); INSERT INTO CN1.ITEMMANAGE_STAFF VALUES ('BR01','IT03','Toi',39); INSERT INTO CN1.ITEMMANAGE_STAFF VALUES ('BR01','IT04','Sang',39); INSERT INTO CN1.ITEMMANAGE_STAFF VALUES ('BR01','IT05','Sang',24); INSERT INTO CN1.ITEMMANAGE_STAFF VALUES ('BR01','IT06','Ngay',37); INSERT INTO CN1.ITEMMANAGE_STAFF VALUES ('BR01','IT07','Chieu',14); INSERT INTO CN1.ITEMMANAGE_STAFF VALUES ('BR01','IT08','Sang',29); INSERT INTO CN1.ITEMMANAGE_STAFF VALUES ('BR01','IT09','Sang',46); INSERT INTO CN1.ITEMMANAGE_STAFF VALUES ('BR01','IT10','Ngay',3);	INSERT INTO CN2.ITEMMANAGE_STAFF VALUES ('BR02','IT01','Ngay',30); INSERT INTO CN2.ITEMMANAGE_STAFF VALUES ('BR02','IT02','Sang',20); INSERT INTO CN2.ITEMMANAGE_STAFF VALUES ('BR02','IT03','Ngay',6); INSERT INTO CN2.ITEMMANAGE_STAFF VALUES ('BR02','IT04','Ngay',44); INSERT INTO CN2.ITEMMANAGE_STAFF VALUES ('BR02','IT05','Chieu',39); INSERT INTO CN2.ITEMMANAGE_STAFF VALUES ('BR02','IT06','Toi',7); INSERT INTO CN2.ITEMMANAGE_STAFF VALUES ('BR02','IT07','Ngay',44); INSERT INTO CN2.ITEMMANAGE_STAFF VALUES ('BR02','IT08','Ngay',37); INSERT INTO CN2.ITEMMANAGE_STAFF VALUES ('BR02','IT09','Ngay',2); INSERT INTO CN2.ITEMMANAGE_STAFF VALUES ('BR02','IT10','Sang',24);
INSERT INTO CN1.BILL VALUES ('BI01',to_date('01/11/2021','dd/mm/yyyy'),'C U05','EMP01',725000); INSERT INTO CN1.BILL VALUES ('BI02',to_date('01/11/2021','dd/mm/yyyy'),'C U06','EMP01',720000);	INSERT INTO CN2.BILL VALUES ('BI13',to_date('11/11/2021','dd/mm/yyyy'),'C U06','EMP11',144000); INSERT INTO CN2.BILL VALUES ('BI14',to_date('11/11/2021','dd/mm/yyyy'),'C U07','EMP11',114000);

INSERT INTO CN1.BILL VALUES ('BI03',to_date('01/11/2021','dd/mm/yyyy'),'C U07','EMP01',735000); INSERT INTO CN1.BILL VALUES ('BI04',to_date('01/11/2021','dd/mm/yyyy'),'C U08','EMP01',908000); INSERT INTO CN1.BILL VALUES ('BI05',to_date('01/11/2021','dd/mm/yyyy'),'C U09','EMP01',660000); INSERT INTO CN1.BILL VALUES ('BI06',to_date('01/11/2021','dd/mm/yyyy'),'C U10','EMP01',763000); INSERT INTO CN1.BILL VALUES ('BI07',to_date('01/11/2021','dd/mm/yyyy'),'C U11','EMP02',1225000); INSERT INTO CN1.BILL VALUES ('BI08',to_date('05/11/2021','dd/mm/yyyy'),'C U12','EMP02',623000); INSERT INTO CN1.BILL VALUES ('BI09',to_date('05/11/2021','dd/mm/yyyy'),'C U13','EMP02',469000); INSERT INTO CN1.BILL VALUES ('BI10',to_date('05/11/2021','dd/mm/yyyy'),'C U14','EMP02',3441000);	INSERT INTO CN2.BILL VALUES ('BI15',to_date('11/11/2021','dd/mm/yyyy'),'C U08','EMP11',150000); INSERT INTO CN2.BILL VALUES ('BI16',to_date('11/11/2021','dd/mm/yyyy'),'C U09','EMP11',384000); INSERT INTO CN2.BILL VALUES ('BI17',to_date('11/11/2021','dd/mm/yyyy'),'C U10','EMP17',90000); INSERT INTO CN2.BILL VALUES ('BI18',to_date('12/11/2021','dd/mm/yyyy'),'C U11','EMP17',30000); INSERT INTO CN2.BILL VALUES ('BI19',to_date('12/11/2021','dd/mm/yyyy'),'C U12','EMP17',330000); INSERT INTO CN2.BILL VALUES ('BI20',to_date('12/11/2021','dd/mm/yyyy'),'C U13','EMP17',192000); INSERT INTO CN2.BILL VALUES ('BI33',to_date('15/11/2021','dd/mm/yyyy'),'C U32','EMP12',264000); INSERT INTO CN2.BILL VALUES ('BI34',to_date('15/11/2021','dd/mm/yyyy'),'C U20','EMP12',300000);
INSERT INTO CN1.BILLDETAIL VALUES ('BI01','IT01',13,260000); INSERT INTO CN1.BILLDETAIL VALUES ('BI01','IT02',1,15000); INSERT INTO CN1.BILLDETAIL VALUES ('BI01','IT10',10,450000); INSERT INTO CN1.BILLDETAIL VALUES ('BI02','IT11',14,630000); INSERT INTO CN1.BILLDETAIL VALUES ('BI02','IT12',2,90000); INSERT INTO CN1.BILLDETAIL VALUES ('BI03','IT29',11,330000); INSERT INTO CN1.BILLDETAIL VALUES ('BI03','IT30',6,270000); INSERT INTO CN1.BILLDETAIL VALUES ('BI03','IT31',3,135000);	INSERT INTO CN2.BILLDETAIL VALUES ('BI13','IT49',6,144000); INSERT INTO CN2.BILLDETAIL VALUES ('BI14','IT32',3,114000); INSERT INTO CN2.BILLDETAIL VALUES ('BI15','IT18',5,150000); INSERT INTO CN2.BILLDETAIL VALUES ('BI16','IT28',12,384000); INSERT INTO CN2.BILLDETAIL VALUES ('BI17','IT30',2,90000); INSERT INTO CN2.BILLDETAIL VALUES ('BI18','IT18',1,30000); INSERT INTO CN2.BILLDETAIL VALUES ('BI19','IT19',11,330000); INSERT INTO CN2.BILLDETAIL VALUES ('BI20','IT07',6,192000);

INSERT INTO CN1.BILLDETAIL VALUES ('BI04','IT34',6,228000);	INSERT INTO CN2.BILLDETAIL VALUES ('BI33','IT47',11,264000);
INSERT INTO CN1.BILLDETAIL VALUES ('BI04','IT35',8,320000);	INSERT INTO CN2.BILLDETAIL VALUES ('BI34','IT20',10,300000);
INSERT INTO CN1.BILLDETAIL VALUES ('BI04','IT36',9,360000);	INSERT INTO CN2.BILLDETAIL VALUES ('BI35','IT38',7,280000);

4. Kiến trúc phân quyền

4.1. Mô tả kiến trúc phân quyền

Chi nhánh 1: Có 3 quyền

- DIRECTOR:
 - Thêm, xóa, sửa được thông tin của quan hệ BRANCH, EMPLOYEE ở cả 2 chi nhánh.
 - Xem thông tin của các quan hệ ở cả 2 chi nhánh.
 - Create Procedure.
- MANAGER_BR1:
 - Xem, thêm, xóa, sửa được thông tin của các quan hệ ITEMMANAGE_MANAGER tại chi nhánh 1
 - Xem được các quan hệ ITEMMANAGE_STAFF, EMPLOYEE, BILL, BILLDETAIL, BRANCH tại chi nhánh 1
 - Xem được các quan hệ CUSTOMER_MANAGER, CUSTOMER_STAFF, ITEM
- STAFF_BR1:
 - Xem, thêm, xóa, sửa được thông tin của các quan hệ ITEMMANAGE_STAFF, BILL, BILLDETAIL tại chi nhánh 1
 - Xem được các quan hệ CUSTOMER_STAFF, ITEM, BRANCH tại chi nhánh 1

Chi nhánh 2: Có 2 quyền

- MANAGER_BR2:
 - Xem, thêm, xóa, sửa được thông tin của các quan hệ ITEMMANAGE_MANAGER tại chi nhánh 2
 - Xem được các quan hệ ITEMMANAGE_STAFF, EMPLOYEE, BILL, BILLDETAIL, BRANCH tại chi nhánh 2
 - Xem được các quan hệ CUSTOMER_MANAGER, CUSTOMER_STAFF, ITEM

- STAFF_BR2:
 - Xem, thêm, xóa, sửa được thông tin của các quan hệ ITEMMANAGE_STAFF, BILL, BILDETAIL tại chi nhánh 2
 - Xem được các quan hệ CUSTOMER_STAFF, ITEM, BRANCH tại chi nhánh 2

4.2. Phân quyền trên cơ sở dữ liệu phân tán

CN1	CN2
ALTER SESSION SET "_ORACLE_SCRIPT" = TRUE;	ALTER SESSION SET "_ORACLE_SCRIPT" = TRUE;
CREATE USER CN1 IDENTIFIED BY CN1;	CREATE USER CN2 IDENTIFIED BY CN2;
GRANT CONNECT, RESOURCE, UNLIMITED TABLESPACE TO CN1;	GRANT CONNECT, RESOURCE, UNLIMITED TABLESPACE TO CN2;
CREATE USER DIRECTOR IDENTIFIED BY DR;	CREATE USER DIRECTOR_BR2 IDENTIFIED BY DR;
GRANT CONNECT, CREATE DATABASE LINK TO DIRECTOR;	GRANT CONNECT, CREATE DATABASE LINK TO DIRECTOR_BR2;
BEGIN FOR t IN (SELECT table_name FROM all_tables WHERE owner = 'CN1') LOOP EXECUTE IMMEDIATE 'GRANT SELECT ON CN1.' t.table_name ' TO DIRECTOR'; END LOOP; END; /	BEGIN FOR t IN (SELECT table_name FROM all_tables WHERE owner = 'CN2') LOOP EXECUTE IMMEDIATE 'GRANT SELECT ON CN2.' t.table_name ' TO DIRECTOR_BR2'; END LOOP; END; /
GRANT INSERT, DELETE, UPDATE ON CN1.EMPLOYEE TO DIRECTOR;	GRANT INSERT, DELETE, UPDATE ON CN2.EMPLOYEE TO DIRECTOR_BR2;
GRANT CREATE PROCEDURE TO DIRECTOR;	
CREATE USER MANAGER_BR1 IDENTIFIED BY MN;	CREATE USER MANAGER_BR2 IDENTIFIED BY MN;

<p>GRANT CONNECT, CREATE DATABASE LINK TO MANAGER_BR1;</p> <p>GRANT INSERT, DELETE, UPDATE ON CN1.ITEMMANAGE_MANAGER TO MANAGER_BR1;</p> <p>BEGIN FOR t IN (SELECT table_name FROM all_tables WHERE owner = 'CN1') LOOP EXECUTE IMMEDIATE 'GRANT SELECT ON CN1.' t.table_name ' TO MANAGER_BR1'; END LOOP; END; /</p>	<p>GRANT CONNECT, CREATE DATABASE LINK TO MANAGER_BR2;</p> <p>GRANT INSERT, DELETE, UPDATE ON CN2.ITEMMANAGE_MANAGER TO MANAGER_BR2;</p> <p>BEGIN FOR t IN (SELECT table_name FROM all_tables WHERE owner = 'CN2') LOOP EXECUTE IMMEDIATE 'GRANT SELECT ON CN2.' t.table_name ' TO MANAGER_BR2'; END LOOP; END; /</p>
<p>CREATE USER STAFF_BR1 IDENTIFIED BY ST;</p> <p>GRANT CONNECT, CREATE DATABASE LINK TO STAFF_BR1;</p> <p>GRANT SELECT, INSERT, DELETE, UPDATE ON CN1.ITEMMANAGE_STAFF TO STAFF_BR1; GRANT SELECT, INSERT, DELETE, UPDATE ON CN1.BILL TO STAFF_BR1; GRANT SELECT, INSERT, DELETE, UPDATE ON CN1.BILLDETAIL TO STAFF_BR1;</p> <p>GRANT SELECT ON CN1.CUSTOMER_STAFF TO STAFF_BR1; GRANT SELECT ON CN1.ITEM TO STAFF_BR1;</p>	<p>CREATE USER STAFF_BR2 IDENTIFIED BY ST;</p> <p>GRANT CONNECT, CREATE DATABASE LINK TO STAFF_BR2;</p> <p>GRANT SELECT, INSERT, DELETE, UPDATE ON CN2.ITEMMANAGE_STAFF TO STAFF_BR2; GRANT SELECT, INSERT, DELETE, UPDATE ON CN2.BILL TO STAFF_BR2; GRANT SELECT, INSERT, DELETE, UPDATE ON CN2.BILLDETAIL TO STAFF_BR2;</p> <p>GRANT SELECT ON CN2.CUSTOMER_STAFF TO STAFF_BR2; GRANT SELECT ON CN2.ITEM TO STAFF_BR2;</p>

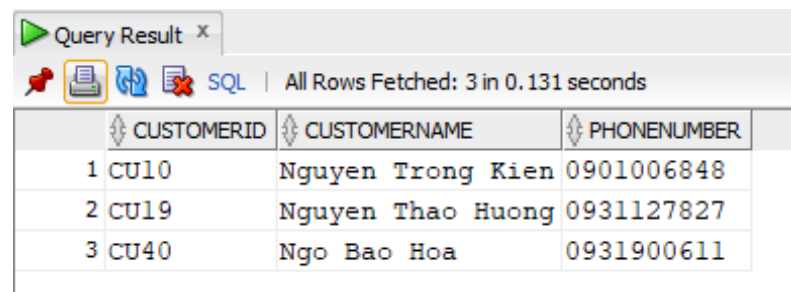
5. Thực hiện 10 câu truy vấn

5.1. Câu truy vấn 1

Mô tả: Tại chi nhánh 1, DIRECTOR tìm khách hàng VIP đã mua hàng tại cả 2 chi nhánh.

```
CONNECT DIRECTOR/DR;  
CREATE DATABASE LINK DR_DR2_LINK CONNECT TO DIRECTOR_BR2  
IDENTIFIED BY DR USING 'CN2';  
SELECT DISTINCT CU1.CustomerID, CustomerName, PhoneNumber  
FROM CN1.CUSTOMER_STAFF CU1  
JOIN CN1.BILL BI1  
ON CU1.CustomerID = BI1.CustomerID  
WHERE CustomerType = 'VIP'  
INTERSECT  
SELECT DISTINCT CU2.CustomerID, CustomerName, PhoneNumber  
FROM CN2.CUSTOMER_STAFF@DR_DR2_LINK CU2  
JOIN CN2.BILL@DR_DR2_LINK BI2  
ON CU2.CustomerID = BI2.CustomerID  
WHERE CustomerType = 'VIP';
```

Kết quả câu truy vấn:



The screenshot shows a 'Query Result' window with a toolbar and a table of results. The table has three columns: CUSTOMERID, CUSTOMERNAME, and PHONENUMBER. There are three rows of data.

	CUSTOMERID	CUSTOMERNAME	PHONENUMBER
1	CU10	Nguyen Trong Kien	0901006848
2	CU19	Nguyen Thao Huong	0931127827
3	CU40	Ngo Bao Hoa	0931900611

Hình 1. Kết quả câu truy vấn 1

5.2. Câu truy vấn 2

Mô tả: Tại chi nhánh 1, DIRECTOR thống kê số lượng đã bán tại tất cả các chi nhánh của từng món (ITEM).

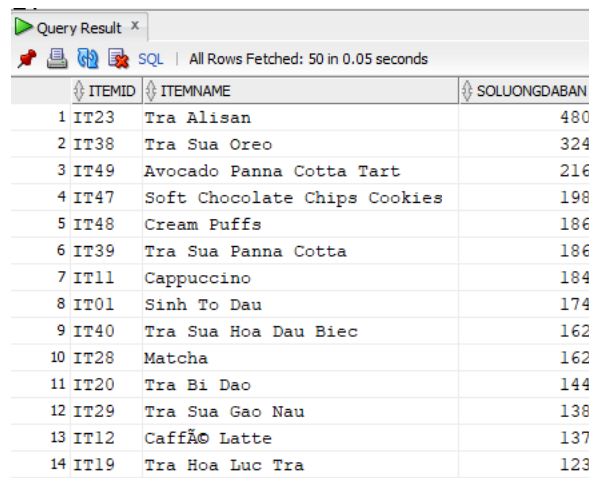
```
CONNECT DIRECTOR/DR;  
SELECT ItemID, ItemName, COALESCE(SUM(SoLuongDaBan), 0) AS SoLuongDaBan  
FROM (  
    SELECT IT1.ItemID, ItemName, COALESCE(SUM(Quantity), 0) AS SoLuongDaBan  
    FROM CN1.ITEM IT1  
    LEFT JOIN CN1.BILDETAIL BD1
```

```

ON IT1.ItemID = BD1.ItemID
GROUP BY IT1.ItemID, ItemName
UNION
SELECT IT2.ItemID, ItemName, COALESCE(SUM(Quantity), 0) AS SoLuongDaBan
FROM CN2.ITEM@DR_DR2_LINK IT2
LEFT JOIN CN2.BILDETAIL@DR_DR2_LINK BD2
ON IT2.ItemID = BD2.ItemID
GROUP BY IT2.ItemID, ItemName
)
GROUP BY ItemID, ItemName
ORDER BY SoLuongDaBan DESC;

```

Kết quả câu truy vấn:



ITEMID	ITEMNAME	SOLUONGDABAN
1 IT23	Tra Alisan	480
2 IT38	Tra Sua Oreo	324
3 IT49	Avocado Panna Cotta Tart	216
4 IT47	Soft Chocolate Chips Cookies	198
5 IT48	Cream Puffs	186
6 IT39	Tra Sua Panna Cotta	186
7 IT11	Cappuccino	184
8 IT01	Sinh To Dau	174
9 IT40	Tra Sua Hoa Dau Biec	162
10 IT28	Matcha	162
11 IT20	Tra Bi Dao	144
12 IT29	Tra Sua Gao Nau	138
13 IT12	Caffè Latte	137
14 IT19	Tra Hoa Luc Tra	123

Hình 2. Kết quả câu truy vấn 2

5.3. Câu truy vấn 3

Mô tả: Tại chi nhánh 1, DIRECTOR tìm những món đang được giảm giá tại chi nhánh 2 nhưng không được giảm giá tại chi nhánh 1.

```

CONNECT DIRECTOR/DR;
SELECT IT2.ItemID, ItemName, ItemType, Discount
FROM CN2.ITEM@DR_DR2_LINK IT2
JOIN CN2.ITEMMANAGE_MANAGER@DR_DR2_LINK IT_MA2
ON IT2.ItemID = IT_MA2.ItemID
WHERE DISCOUNT > 0
MINUS
SELECT IT1.ItemID, ItemName, ItemType, Discount
FROM CN1.ITEM IT1


```

```

JOIN CN1.ITEMMANAGE_MANAGER IT_MA1
ON IT1.ItemID = IT_MA1.ItemID
WHERE DISCOUNT > 0;

```

Kết quả câu truy vấn:



	ITEMID	ITEMNAME	ITEMTYPE	DISCOUNT
1	IT03	Sinh To Mang Cau	Sinh To	5
2	IT08	Ca Phe Sua Bac Ha	Ca Phe Pha Phin	10
3	IT16	Lungo Espresso	Ca Phe Italy	10
4	IT27	Choco Caramel	Da Xay	10
5	IT36	Tra Sua Earl Grey	Tra Sua	15
6	IT46	Mini Cookies	Banh Ngot	5

Hình 3. Kết quả câu truy vấn 3

5.4. Câu truy vấn 4

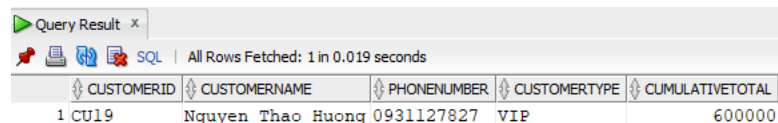
Mô tả: Tại chi nhánh 2, MANAGER tìm khách hàng có số tiền tích lũy lớn nhất đã mua hàng tại chi nhánh 2.

```

CONNECT MANAGER_BR2/MN;
SELECT CU_ST.CustomerID, CustomerName, PhoneNumber, CustomerType,
CumulativeTotal
FROM CN2.CUSTOMER_STAFF CU_ST
JOIN CN2.CUSTOMER_MANAGER CU_MA
ON CU_ST.CustomerID = CU_MA.CustomerID
JOIN CN2.BILL BI
ON CU_MA.CustomerID = BI.CustomerID
ORDER BY CumulativeTotal DESC
FETCH FIRST 1 ROW ONLY;

```

Kết quả câu truy vấn:



	CUSTOMERID	CUSTOMERNAME	PHONENUMBER	CUSTOMERTYPE	CUMULATIVETOTAL
1	CU19	Nguyen Thao Huong	0931127827	VIP	600000

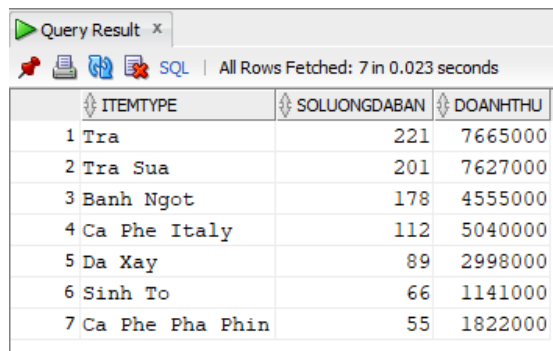
Hình 4. Kết quả câu truy vấn 4

5.5. Câu truy vấn 5

Mô tả: Tại chi nhánh 2, STAFF thống kê số lượng món ăn từng loại đã bán được trong tháng 11 tại chi nhánh 2.

```
CONNECT STAFF_BR2/ST;  
  
SELECT IT.ItemType, SUM(BD.QUANTITY) AS SoLuongDaBan, SUM(BD.SubTotal) AS  
DoanhThu  
FROM CN2.ITEM IT  
JOIN CN2.BILLDETAIL BD  
ON IT.ItemID = BD.ItemID  
JOIN CN2.BILL BI  
ON BI.BillID = BD.BillID  
WHERE EXTRACT(MONTH FROM BI.BillDate) = 11  
GROUP BY IT.ItemType  
ORDER BY SoLuongDaBan DESC, DoanhThu DESC;
```

Kết quả câu truy vấn:



ITEMTYPE	SOLUONGDABAN	DOANHTHU
1 Tra	221	7665000
2 Tra Sua	201	7627000
3 Banh Ngot	178	4555000
4 Ca Phe Italy	112	5040000
5 Da Xay	89	2998000
6 Sinh To	66	1141000
7 Ca Phe Pha Phin	55	1822000

Hình 5. Kết quả câu truy vấn 5

5.6. Câu truy vấn 6

Mô tả: Tại chi nhánh 1, DIRECTOR tìm thông tin các món đã bán hơn 50 sản phẩm tại cả 2 chi nhánh.

```
CONNECT DIRECTOR/DR;  
  
SELECT I.ItemID, I.ItemName,  
      (SELECT SUM(BD.Quantity)  
       FROM CN1.BILLDETAIL BD  
       WHERE BD.ItemID = I.ItemID) AS TotalQuantityCN1,  
      (SELECT SUM(BD.Quantity)  
       FROM CN2.BILLDETAIL@DR_DR2_LINK BD  
       WHERE BD.ItemID = I.ItemID) AS TotalQuantityCN2
```




```

FROM CN1.ITEM I
WHERE EXISTS (SELECT 1
              FROM CN2.ITEM@DR_DR2_LINK I2
              WHERE I.ItemID = I2.ItemID)
GROUP BY I.ItemID, I.ItemName
HAVING (SELECT SUM(BD.Quantity)
       FROM CN1.BILLDETAIL BD
       WHERE BD.ItemID = I.ItemID) > 50
AND
(SELECT SUM(BD.Quantity)
 FROM CN2.BILLDETAIL@DR_DR2_LINK BD
 WHERE BD.ItemID = I.ItemID) > 50;

```

Kết quả câu truy vấn:



	ITEMID	ITEMNAME	TOTALQUANTITYCN1	TOTALQUANTITYCN2
1	IT49	Avocado Panna Cotta Tart	126	90
2	IT28	Matcha	96	66
3	IT19	Tra Hoa Luc Tra	70	53
4	IT47	Soft Chocolate Chips Cookies	121	77
5	IT20	Tra Bi Dao	86	58
6	IT38	Tra Sua Oreo	203	121
7	IT11	Cappuccino	107	77
8	IT01	Sinh To Dau	116	58
9	IT39	Tra Sua Panna Cotta	124	62
10	IT40	Tra Sua Hoa Dau Biec	108	54
11	IT48	Cream Puffs	124	62
12	IT23	Tra Alisan	320	160

Hình 6. Kết quả câu truy vấn 6

5.7. Câu truy vấn 7

Mô tả: Tại chi nhánh 1, DIRECTOR tính tổng doanh thu của mỗi chi nhánh.

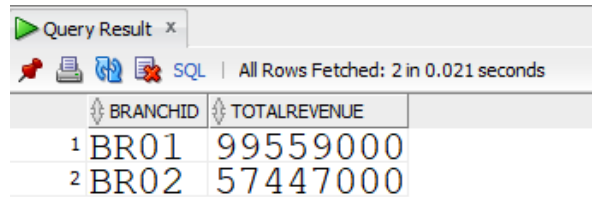
```

CONNECT DIRECTOR/DR;
SELECT BranchID, TotalRevenue
FROM (
  SELECT BranchID, SUM(Total) AS TotalRevenue
  FROM CN1.BILL, CN1.BRANCH
  GROUP BY BranchID
  UNION ALL
  SELECT BranchID, SUM(Total) AS TotalRevenue
  FROM CN2.BILL@DR_DR2_LINK, CN2.BRANCH@DR_DR2_LINK
  GROUP BY BranchID

```

);

Kết quả câu truy vấn:



Query Result x

All Rows Fetched: 2 in 0.021 seconds

	BRANCHID	TOTALREVENUE
1	BR01	99559000
2	BR02	57447000

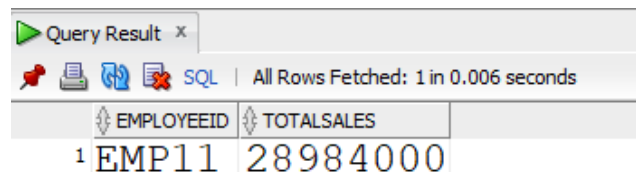
Hình 7. Kết quả câu truy vấn 7

5.8. Câu truy vấn 8

Mô tả: Tại chi nhánh 2, MANAGER tìm nhân viên có doanh số bán hàng cao nhất tại chi nhánh 2.

```
CONNECT MANAGER_BR2/MN;  
SELECT EmployeeID, SUM(Total) AS TotalSales  
FROM CN2.BILL B  
GROUP BY EmployeeID  
ORDER BY TotalSales DESC  
FETCH FIRST 1 ROW ONLY;
```

Kết quả câu truy vấn:



Query Result x

All Rows Fetched: 1 in 0.006 seconds

	EMPLOYEEID	TOTALSALES
1	EMP11	28984000

Hình 8. Kết quả câu truy vấn 8

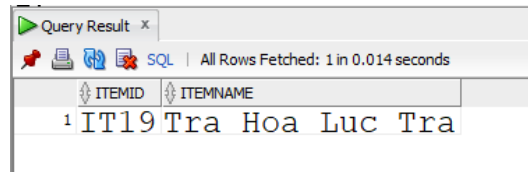
5.9. Câu truy vấn 9

Mô tả: Tại chi nhánh 2, STAFF tìm món không có sẵn để bán.

```
CONNECT MANAGER_BR2/MN;  
SELECT I.ItemID, I.ItemName  
FROM CN2.ITEM I  
WHERE NOT EXISTS (  
    SELECT 1  
    FROM CN2.ITEMMANAGE_STAFF IMS
```

```
WHERE I.ItemID = IMS.ItemID AND IMS.Availability > 0);
```

Kết quả câu truy vấn:



ITEMID	ITEMNAME
1 IT19	Tra Hoa Luc Tra

Hình 9. Kết quả câu truy vấn 9

5.10. Câu truy vấn 10

Mô tả: Tại chi nhánh 1, DIRECTOR xác định số lượng món mà mỗi nhân viên tại cả 2 chi nhánh đã lập hóa đơn.

```
CONNECT DIRECTOR/DR;
```

```
SELECT
```

```
    BR.BranchId,  
    E.EmployeeID,  
    E.EmployeeName,  
    COUNT(BD.ItemID) AS TotalItemsSold
```

```
FROM
```

```
    CN1.BILL B  
    JOIN CN1.BILLDETAIL BD ON B.BillID = BD.BillID  
    JOIN CN1.EMPLOYEE E ON B.EmployeeID = E.EmployeeID,  
    CN1.BRANCH BR
```

```
GROUP BY
```

```
    E.EmployeeID,  
    E.EmployeeName,  
    BR.BranchId
```

```
UNION ALL
```

```
SELECT
```

```
    BR.BranchId,  
    E.EmployeeID,  
    E.EmployeeName,  
    COUNT(BD.ItemID) AS TotalItemsSold
```

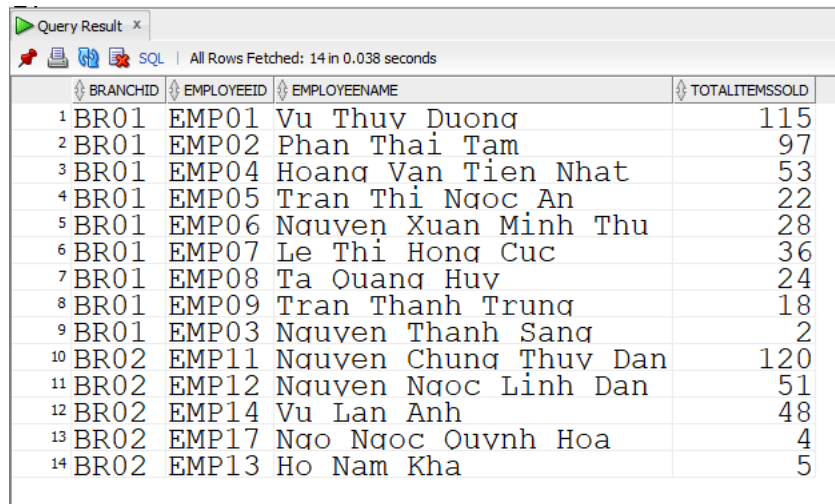
```
FROM
```

```
    CN2.BILL@DR_DR2_LINK B  
    JOIN CN2.BILLDETAIL@DR_DR2_LINK BD ON B.BillID = BD.BillID  
    JOIN CN2.EMPLOYEE@DR_DR2_LINK E ON B.EmployeeID = E.EmployeeID,  
    CN2.BRANCH@DR_DR2_LINK BR
```

GROUP BY

E.EmployeeID,
E.EmployeeName,
BR.BranchId;

Kết quả câu truy vấn:



	BRANCHID	EMPLOYEEID	EMPLOYEENAME	TOTALITEMSSOLD
1	BR01	EMP01	Vu Thuy Duong	115
2	BR01	EMP02	Phan Thai Tam	97
3	BR01	EMP04	Hoang Van Tien Nhat	53
4	BR01	EMP05	Tran Thi Ngoc An	22
5	BR01	EMP06	Nguyen Xuan Minh Thu	28
6	BR01	EMP07	Le Thi Hong Cuc	36
7	BR01	EMP08	Ta Quang Huy	24
8	BR01	EMP09	Tran Thanh Trung	18
9	BR01	EMP03	Nguyen Thanh Sang	2
10	BR02	EMP11	Nguyen Chung Thuy Dan	120
11	BR02	EMP12	Nguyen Ngoc Linh Dan	51
12	BR02	EMP14	Vu Lan Anh	48
13	BR02	EMP17	Ngoc Ngoc Ouyvnh Hoa	4
14	BR02	EMP13	Ho Nam Kha	5

Hình 10. Kết quả câu truy vấn 10

II. VIẾT HÀM (THỦ TỤC) VÀ RÀNG BUỘC TOÀN Vẹn

1. Hàm (Function)

Mô tả: Hàm thực hiện tính doanh thu trung bình tại cả 2 chi nhánh của các món thuộc 01 LOẠI món bất kỳ.

```
CONNECT DIRECTOR/DR;  
  
CREATE OR REPLACE FUNCTION  
calculate_avg_revenue_by_item_type(item_type_input VARCHAR2)  
RETURN NUMBER IS  
    avg_revenue_cn1 NUMBER := 0;  
    avg_revenue_cn2 NUMBER := 0;  
    avg_revenue_combined NUMBER := 0;  
    total_items_cn1 NUMBER := 0;  
    total_items_cn2 NUMBER := 0;  
BEGIN  
    -- Calculate CN1's average revenue
```



```

FOR EACH ROW
DECLARE
    v_cumulative_total NUMBER;
    v_customer_type VARCHAR2(25);
    v_new_discount NUMBER;
BEGIN
    -- Check if customer exists in the CUSTOMER_MANAGER table
    SELECT CumulativeTotal INTO v_cumulative_total
    FROM CUSTOMER_MANAGER
    WHERE CustomerID = :NEW.CustomerID;

    IF SQL%NOTFOUND THEN
        -- If not found, insert a new record with the Total from the BILL
        INSERT INTO CUSTOMER_MANAGER (CustomerID, CumulativeTotal)
        VALUES (:NEW.CustomerID, :NEW.Total);
        v_cumulative_total := :NEW.Total;
    ELSE
        -- If found, update the cumulative total
        v_cumulative_total := v_cumulative_total + :NEW.Total - NVL(:OLD.Total, 0);
        UPDATE CUSTOMER_MANAGER
        SET CumulativeTotal = v_cumulative_total
        WHERE CustomerID = :NEW.CustomerID;
    END IF;

    -- Determine customer type based on the updated cumulative total
    IF v_cumulative_total >= 500000 THEN
        v_customer_type := 'VIP';
    ELSIF v_cumulative_total >= 300000 THEN
        v_customer_type := 'Member';
    ELSE
        v_customer_type := 'Standard';
    END IF;

    -- Update the customer type
    UPDATE CUSTOMER_STAFF
    SET CustomerType = v_customer_type
    WHERE CustomerID = :NEW.CustomerID;

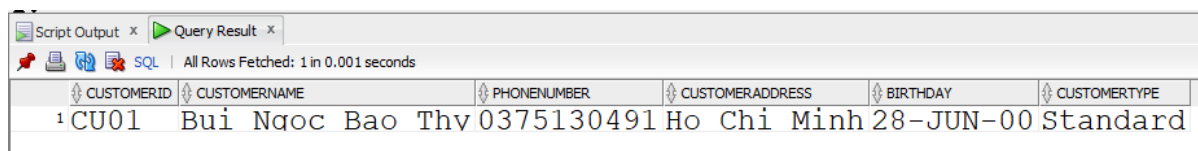
END Bill_After_Insert_Or_Update;

```

Câu truy vấn kiểm thử: Thêm 1 hóa đơn mới có tổng tiền lớn hơn 300000 cho khách hàng có mã ‘CU01’.

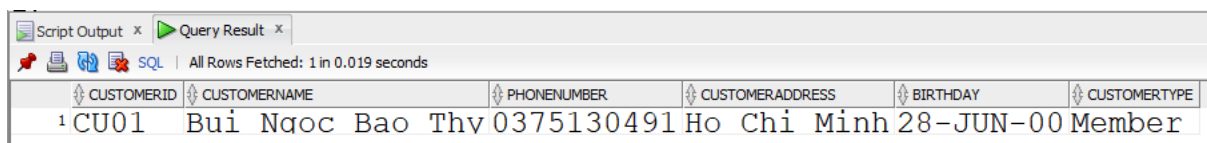
CONNECT STAFF_BR1/ST;
SELECT * FROM CN1.CUSTOMER_STAFF WHERE CustomerID = 'CU01';
INSERT INTO BILL (BillID, BillDate, CustomerID, EmployeeID, Total) VALUES ('12346', SYSDATE, 'CU01', 'EMP01', 301000.00);
COMMIT;
SELECT * FROM CN1.CUSTOMER_STAFF WHERE CustomerID = 'CU01';

Kết quả câu truy vấn:



CUSTOMERID	CUSTOMERNAME	PHONENUMBER	CUSTOMERADDRESS	BIRTHDAY	CUSTOMERTYPE
1 CU01	Bùi Ngọc Bảo Thy	0375130491	Hồ Chí Minh	28-JUN-00	Standard

Hình 12. Dữ liệu khách hàng ‘CU01’ trước khi thêm hóa đơn



CUSTOMERID	CUSTOMERNAME	PHONENUMBER	CUSTOMERADDRESS	BIRTHDAY	CUSTOMERTYPE
1 CU01	Bùi Ngọc Bảo Thy	0375130491	Hồ Chí Minh	28-JUN-00	Member

Hình 13. Dữ liệu khách hàng ‘CU01’ sau khi thêm hóa đơn và áp dụng Trigger

III. CÁC MỨC CÔ LẬP TRONG MÔI TRƯỜNG PHÂN TÁN

1. Trường hợp Lost Update

1.1. Mô tả tình huống

Tại T0, địa chỉ của nhân viên EMP01 là “Vinh Phúc”.

Tại T1, giám đốc 1 cập nhật địa chỉ cho nhân viên EMP01 là “TP.HCM”..

Tại T2, giám đốc 2 cũng cập nhật địa chỉ cho nhân viên EMP01 là “Hai Phong”.

Tại T3, giám đốc 1 thực hiện COMMIT.

Tại T4, nhân viên 2 thực hiện COMMIT. Thông tin cập nhật của giám đốc 2 sẽ ghi đè lên thông tin cập nhật của giám đốc 1. Như vậy, kết quả là dữ liệu cập nhật của giám đốc 1 sẽ bị mất.

Vậy tại T5, địa chỉ của nhân viên EMP01 là “Hai Phong”.

1.2. Minh họa

Time	Transaction T1 (Director 1) - CN1	T1 Output	Transaction T1 (Director 2) - CN2	T2 Output
			CREATE DATABASE LINK DR2_DR_LINK CONNECT TO DIRECTOR IDENTIFIED BY DR USING 'CN1'	
T0	SELECT EmployeeAddress FROM CN1.EMPLOYEE WHERE EmployeeID = 'EMP01';	Vinh Phuc	SELECT EmployeeAddress FROM CN1.EMPLOYEE@ DR2_DR_LINK WHERE EmployeeID = 'EMP01';	Vinh Phuc
T1	UPDATE CN1.EMPLOYEE SET EmployeeAddress = 'TP.HCM' WHERE EmployeeID='EMP01';	1 row updated.		
T2			UPDATE CN1.EMPLOYEE@ DR2_DR_LINK SET EmployeeAddress = 'Hai Phong' WHERE EmployeeID = 'EMP01';	
T3	Commit;	Commit complete.		1 row updated
T4			Commit;	Commit complete.

T5	SELECT EmployeeAddress FROM CN1.EMPLOYEE WHERE EmployeeID = 'EMP01';	Hai Phong	SELECT EmployeeAdress FROM CN1.EMPLOYEE@ DR2_DR_LINK WHERE EmployeeID = 'EMP01';	Hai Phong
----	---	-----------	--	-----------

1.3. Giải pháp

Thay đổi mức cô lập mặc định (Default isolation level – Read committed) thành Serializable bằng cách thực hiện câu lệnh:

SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;

2. Trường hợp Unrepeatable Read

2.1. Mô tả tình huống

Tại T0 giám đốc 1 đang xem thông tin của nhân viên EMP01. Tại thời điểm này, số điện thoại của nhân viên EMP01 là '01234678'.

Tại T1, giám đốc 1 cập nhật số điện thoại cho nhân viên EMP01 là "0999999"

Tại T2, giám đốc 2 commit.

Tại T3, giám đốc 1 xem xét lại thông tin của nhân viên EMP01 và nhận thấy rằng số điện thoại đã khác. Như vậy, hai lần xem thông tin nhân viên trả về hai kết quả khác nhau.

Nguyên nhân: Khi giao dịch T1 đọc dữ liệu hai lần, giao dịch T2 cập nhật dữ liệu giữa hai lần đọc. Như vậy, hai lần đọc dữ liệu trả về hai kết quả khác nhau.

2.2. Minh họa

Time	Transaction T1(Director 1)	T1 Output	Transaction T1(Director 2)	T2 Output
T0	SELECT PhoneNumber FROM CN1.EMPLOYEE WHERE EmployeeID = 'EMP01';	'012345678 ,		
T1			UPDATE CN1.EMPLOYEE@D R2_DR_LINK SET PhoneNumber = '0999999' WHERE EmployeeID = 'EMP01';	1 row update.

T2			Commit;	Commit complete.
T3	SELECT PhoneNumber FROM CN1.EMPLOYEE WHERE EmployeeID = 'EMP01';	'0999999'		

2.3. Giải pháp

Thay đổi mức cô lập mặc định (Default isolation level – Read committed) thành Serializable bằng cách thực hiện câu lệnh:

SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;

3. Trường hợp Phantom Read

3.1. Mô tả tình huống

Tại T0, giám đốc 1 xem thông tin của nhân viên EMP10.

Tại T2, giám đốc 2 xóa thông tin của nhân viên EMP10.

Tại T3, giám đốc 1 thử đọc thông tin của nhân viên EMP10 nhưng không tìm thấy.

Nguyên nhân: Đây là vấn đề Phantom Read khi một Transaction T2 đọc dữ liệu hai lần, Transaction T1 xóa dữ liệu giữa hai lần đọc. Lần thứ hai xảy ra lỗi do Transaction T1 đã xóa dữ liệu.

3.2. Minh họa

Time	Transaction T1(Director 1)	T1 Output	Transaction T2(Director 2)	T2 Output
T0	SELECT * FROM CN1.EMPLOYEE WHERE EmployeeID = 'EMP10';	1 row fetched.		
T1			DELETE FROM CN1.EMPLOYEE@DR2 _DR_LINK WHERE EmployeeID='EMP10';	1 row deleted.
T2			Commit;	Commit complete.
T3	SELECT * FROM CN1.EMPLOYEE WHERE EmployeeID = 'EMP10';	0 row fetched.		

3.3. Giải pháp

Thay đổi mức cô lập mặc định (Default isolation level – Read committed) thành Serializable bằng cách thực hiện câu lệnh:

```
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
```

4. Trường hợp Deadlock

4.1. Mô tả tình huống

Tại T0, nhân viên 1 cập nhật trạng thái của sản phẩm IT01.

Tại T1, nhân viên 2 cập nhật trạng thái của sản phẩm IT02.

Tại T2, nhân viên 1 cập nhật trạng thái của sản phẩm IT02.

Tại T3, nhân viên 2 cập nhật trạng thái của sản phẩm IT01. Và bế tắc xảy ra.

Nguyên nhân: Transaction T1 giữ khóa đơn vị dữ liệu của A, chờ khóa đơn vị dữ liệu của B và Transaction T2 giữ khóa đơn vị dữ liệu của B và chờ khóa đơn vị dữ liệu của B từ nhân viên A. Hai giao dịch chờ khóa vô hạn gây ra trạng thái deadlock

4.2. Minh họa

Time	Transaction T1(Director 1)	T1 Output	Transaction T1(Director 2)	T2 Output
T0	UPDATE CN1.ITEMMANAGE_M ANAGER SET SaleStatus = 'Khong duoc ban' WHERE ItemID='IT01';	1 row updated.		
T1			UPDATE CN1.ITEMMANAGE_ MANAGER@MN2_M N_LINK SET SaleStatus ='Khong duoc ban' WHERE ItemID='IT02';	1 row updated.
T2	UPDATE CN1.ITEMMANAGE_M ANAGER SET SaleStatus ='Duoc phep ban' WHERE ItemID='IT02';			
T3			UPDATE CN1.ITEMMANAGE_ MANAGER@MN2 M	

			N_LINK SET SaleStatus = 'Duoc pheap ban' WHERE ItemID='IT01';	
T4		ORA-00060: deadlock detected while waiting for resource		

4.3. Giải pháp

Hệ quản trị cơ sở dữ liệu Oracle sẽ tự động ROLLBACK giao dịch không thành công. Trong tình huống này, Transaction T1 được hệ quản trị cơ sở dữ liệu Oracle ROLLBACK.

IV. TỐI ƯU HÓA TRUY VẤN TRÊN MÔI TRƯỜNG PHÂN TÁN

1. Câu truy vấn đơn giản chưa tối ưu

Mô tả: Cho biết thông tin khách hàng (mã số, tên, loại khách hàng) đã mua trên 1 triệu vào ngày 5/12/2021 ở chi nhánh 'Nguyễn Du'.

```
SELECT DISTINCT C.CustomerID, CustomerName, CustomerType
FROM EMPLOYEE E, BILL B, BRANCH BR, CUSTOMER C
WHERE B.EmployeeID = E. EmployeeID
      AND B. CustomerID = C. CustomerID
      AND BR.BranchID = E. BranchID
      AND Total >= 1000000
      AND BillDate = '05-DEC-21'
      AND BranchName = 'Nguyễn Du';
```

2. Explain câu truy vấn đơn giản

Câu truy vấn đơn giản trên các phân mảnh:

```
SELECT DISTINCT C.CustomerID, CustomerName, CustomerType
FROM
  (SELECT * FROM CN1.EMPLOYEE
   UNION
   SELECT * FROM CN2.EMPLOYEE@DR_DR2_LINK) E,
  (SELECT * FROM CN1.BILL
```

```

UNION
SELECT * FROM CN2.BILL@DR_DR2_LINK) B,
(SELECT * FROM CN1.BRANCH
UNION
SELECT * FROM CN2.BRANCH@DR_DR2_LINK) BR,
(SELECT * FROM CN1.CUSTOMER_STAFF
UNION
SELECT * FROM CN2.CUSTOMER_STAFF@DR_DR2_LINK) C
WHERE B.EmployeeID = E.EmployeeID
      AND B.CustomerID = C.CustomerID
      AND BR.BranchID = E.BranchID
      AND Total >= 1000000
      AND BillDate = '05-DEC-21'
      AND BranchName = 'Nguyen Du';

```

Explain câu truy vấn trên:

```

EXPLAIN PLAN FOR
SELECT /*+ gather_plan_statistics */ DISTINCT C.CustomerID, CustomerName,
CustomerType
FROM
  (SELECT * FROM CN1.EMPLOYEE
  UNION
  SELECT * FROM CN2.EMPLOYEE@DR_DR2_LINK) E,
  (SELECT * FROM CN1.BILL
  UNION
  SELECT * FROM CN2.BILL@DR_DR2_LINK) B,
  (SELECT * FROM CN1.BRANCH
  UNION
  SELECT * FROM CN2.BRANCH@DR_DR2_LINK) BR,
  (SELECT * FROM CN1.CUSTOMER_STAFF
  UNION
  SELECT * FROM CN2.CUSTOMER_STAFF@DR_DR2_LINK) C
WHERE B.EmployeeID = E.EmployeeID
      AND B.CustomerID = C.CustomerID
      AND BR.BranchID = E.BranchID
      AND Total >= 1000000
      AND BillDate = '05-DEC-21'
      AND BranchName = 'Nguyen Du';

SELECT * FROM table(dbms_xplan.display );

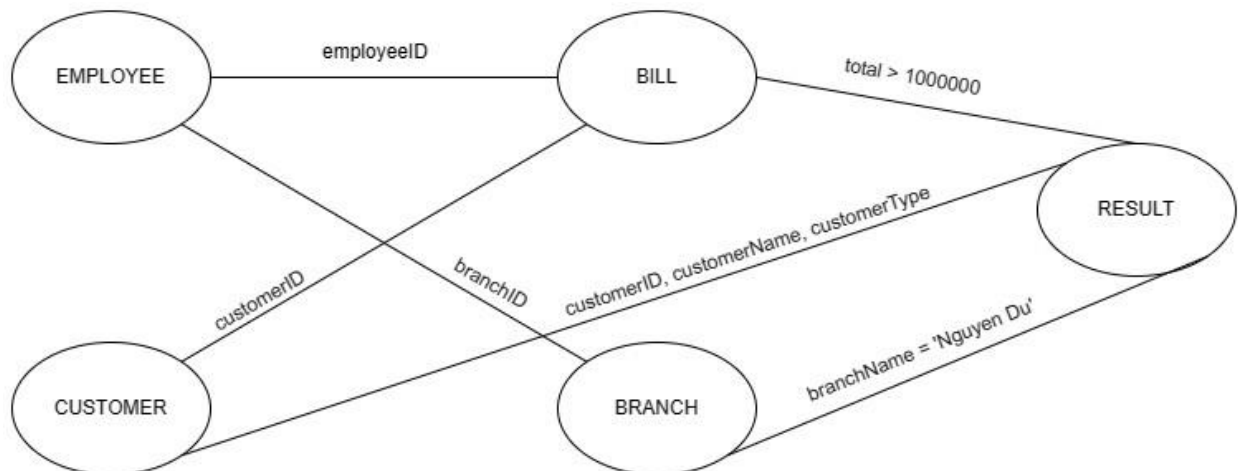
```

Kết quả:

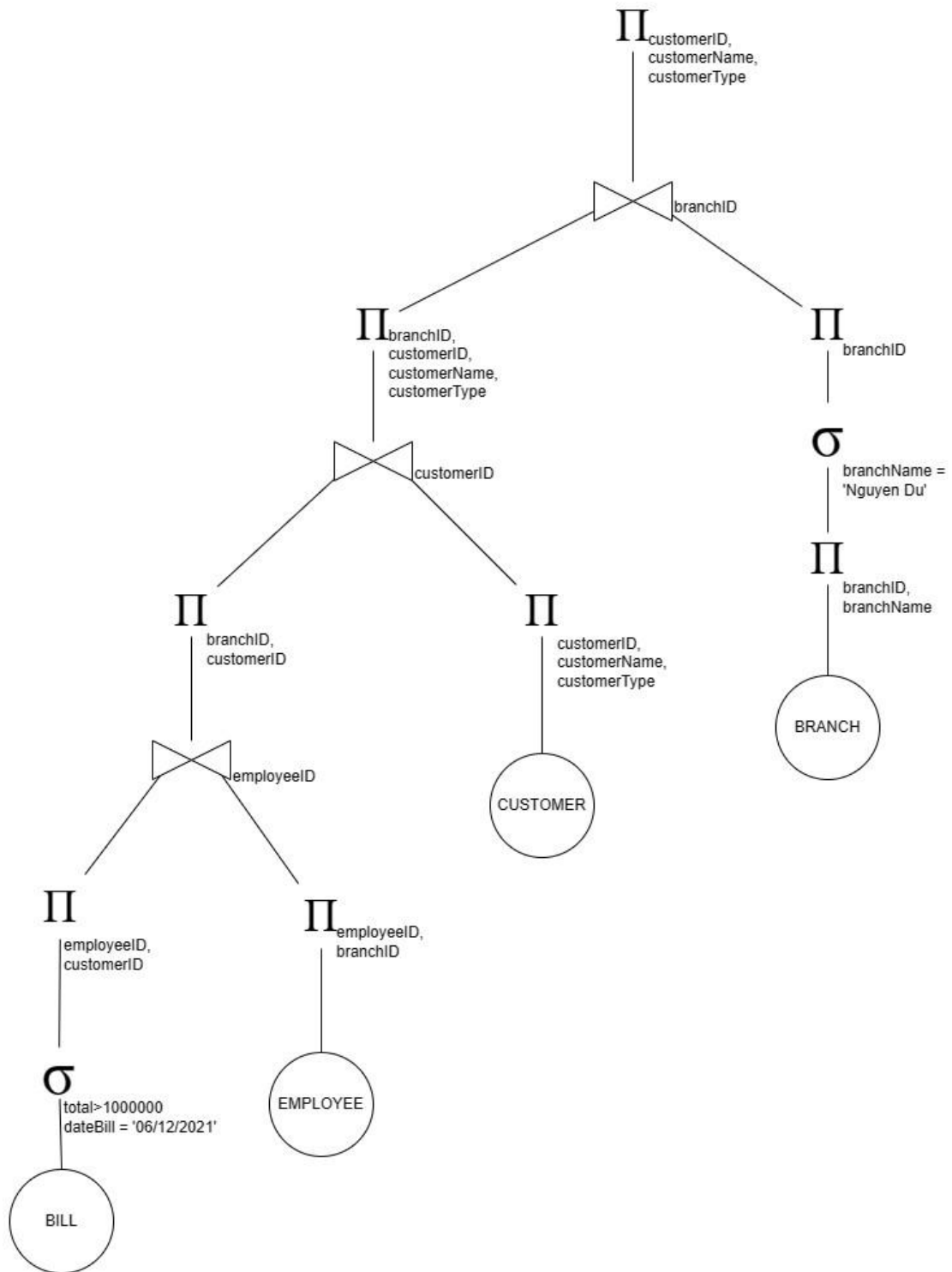
Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time	Inst
0	SELECT STATEMENT		1	114	18 (23)	00:00:01	
1	HASH UNIQUE		1	114	18 (23)	00:00:01	
2	NESTED LOOPS		1	114	17 (18)	00:00:01	
3	NESTED LOOPS SEMI		1	69	16 (19)	00:00:01	
4	HASH JOIN		3	114	13 (24)	00:00:01	
5	VIEW		16	480	9 (34)	00:00:01	
6	HASH UNIQUE		16	464	9 (34)	00:00:01	
7	UNION-ALL						
8	TABLE ACCESS FULL	BILL	8	232	3 (0)	00:00:01	DR DR~
9	REMOTE	BILL	8	232	3 (0)	00:00:01	
10	VIEW		20	1120	7 (15)	00:00:01	
11	HASH UNIQUE		20	1120	7 (15)	00:00:01	
12	UNION-ALL						
13	TABLE ACCESS FULL	EMPLOYEE	10	570	3 (0)	00:00:01	DR DR~
14	REMOTE	EMPLOYEE	10	550	3 (0)	00:00:01	
15	VIEW		1	31	1 (0)	00:00:01	
16	SORT UNIQUE						
17	UNION-ALL PARTITION						
18	TABLE ACCESS BY INDEX ROWID	BRANCH	1	75	1 (0)	00:00:01	
19	INDEX UNIQUE SCAN	PK BRANCH	1		0 (0)	00:00:01	
20	REMOTE	BRANCH	1	65	1 (0)	00:00:01	DR DR~
21	VIEW		2	90	1 (0)	00:00:01	
22	SORT UNIQUE						
23	UNION-ALL PARTITION						
24	TABLE ACCESS BY INDEX ROWID	CUSTOMER STAFF	1	57	1 (0)	00:00:01	
25	INDEX UNIQUE SCAN	PK CUSTOMERSTAFF	1		0 (0)	00:00:01	
26	REMOTE	CUSTOMER STAFF	1	57	1 (0)	00:00:01	DR DR~

Hình 14. Kết quả explain câu truy vấn chưa tối ưu hóa cục bộ

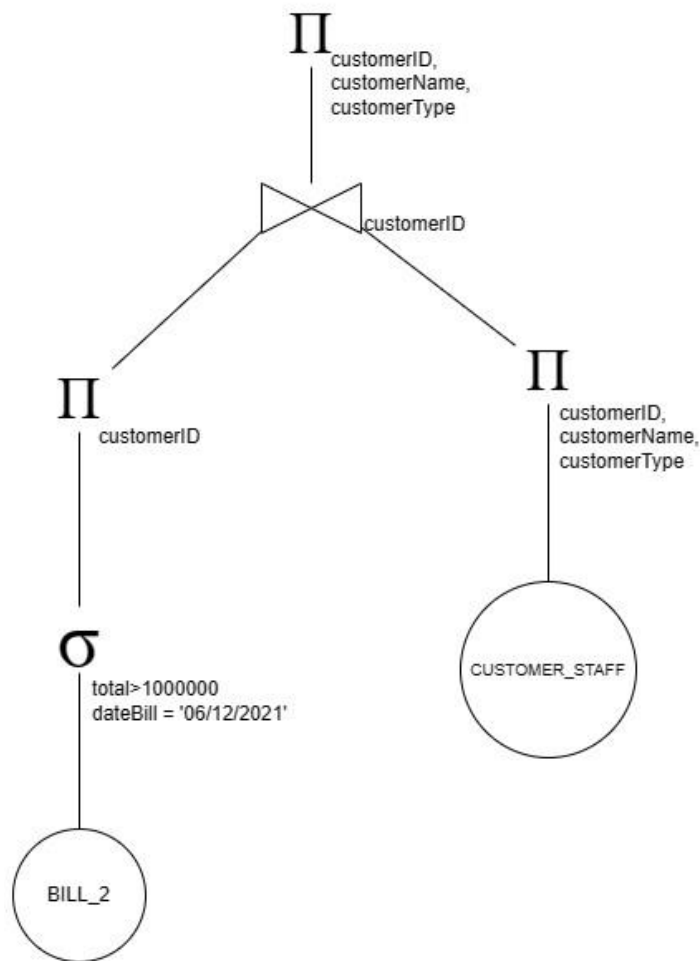
3. Tối ưu hóa câu truy vấn cục bộ, phân tán



Hình 15. Đồ thị truy vấn



Hình 16. Cây truy vấn tối ưu trên môi trường tập trung



Hình 17. Cây truy vấn tối ưu trên môi trường phân tán

4. Viết lại câu truy vấn trên môi trường phân tán và nhận xét

```
SELECT DISTINCT C.CustomerID, CustomerName, CustomerType
FROM CN2.BILL B, CUSTOMER_STAFF C
WHERE B. CustomerID = C. CustomerID
      AND Total >= 1000000
      AND BillDate = '05-DEC-21';
```

Explain câu truy vấn trên môi trường phân tán:

```
EXPLAIN PLAN FOR
SELECT /*+ gather_plan_statistics */ DISTINCT C.CustomerID, CustomerName,
CustomerType
FROM CN2.BILL B, CUSTOMER_STAFF C
WHERE B. CustomerID = C. CustomerID
      AND Total >= 1000000
```


AND BillDate = '05-DEC-21';
SELECT * FROM table(dbms_xplan.display);

Kết quả:

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time	Inst	IN-OUT
0	SELECT STATEMENT		8	464	7 (29)	00:00:01		
1	HASH UNIQUE		8	464	7 (29)	00:00:01		
2	MERGE JOIN		8	464	6 (17)	00:00:01		
3	TABLE ACCESS BY INDEX ROWID	CUSTOMER STAFF	100	3200	2 (0)	00:00:01		
4	INDEX FULL SCAN	PK CUSTOMERSTAFF	100		1 (0)	00:00:01		
5	SORT JOIN		8	208	4 (25)	00:00:01		
6	REMOTE	BILL	8	208	3 (0)	00:00:01	DR DR~	R->S

Hình 18. Kết quả explain câu truy vấn rút gọn trên môi trường phân tán

Nhận xét: Thời gian thực hiện câu truy vấn đã được tối ưu nhanh hơn thời gian thực hiện câu truy vấn chưa được tối ưu.