

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA HỆ THỐNG THÔNG TIN

□□□

Báo cáo thực hành:

BÀI TẬP LỚN 1

THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN



Môn học:

CƠ SỞ DỮ LIỆU PHÂN TÁN

Giảng viên giảng dạy: **Nguyễn Minh Nhựt**

Lớp: IS211.N11.HTCL

Sinh viên thực hiện:

Nguyễn Thị Mỹ Trân	-	20520322
Tôn Nữ Tú Quyên	-	20520296
Thái Tăng Đức	-	20521203
Trần Anh Huy	-	20520551

Tp.HCM, tháng 12 năm 2022

MỤC LỤC

MỤC LỤC.....	1
1. THIẾT KẾ CSDL PHÂN TÁN	3
1.1. Thiết kế và mô tả csdl phân tán.....	3
1.2. Kiến trúc phân mảnh	5
1.2.1. Mô tả phân mảnh.....	5
1.2.2. Minh họa trên 2 chi nhánh	5
1.3. Xây dựng csdl phân tán.....	6
1.3.1. Dll.....	6
1.3.2. Dml.....	11
1.4. Kiến trúc phân quyền	34
1.4.1. Mô tả kiến trúc phân quyền.....	34
1.4.2. Minh họa	35
1.5. Thực hiện 10 câu truy vấn.....	40
2. VIẾT HÀM, THỦ TỤC VÀ RÀNG BUỘC TOÀN VỆN	52
2.1. Function.....	52
2.1. Procedure.....	54
2.3. Ràng buộc toàn vẹn.....	55
3. DEMO CÁC MỨC CÔ LẬP (ISOLATION LEVEL) TRONG MÔI TRƯỜNG PHÂN TÁN.....	58
3.1. Trường hợp lost update	58
3.1.1. Mô tả tình huống	58
3.1.2. Minh họa	58
3.1.3. Giải pháp	59

3.2. Trường hợp dirty read	59
3.3. Trường hợp unrepeatable read	60
3.4. Trường hợp phantom read.....	61
3.5. Trường hợp deadlock	63
4. THỰC HIỆN TỐI ƯU HÓA TRUY VẤN TRÊN MÔI TRƯỜNG PHÂN TÁN.....	65
4.1. Câu truy vấn đơn giản chưa tối ưu	65
4.2. EXPLAIN QUERY câu truy vấn đơn giản	66
4.3. Tối ưu hóa câu truy vấn cục bộ, phân tán	67
4.4. Viết lại câu query trên môi trường phân tán	70
5. CƠ CHẾ NHÂN BẢN TRONG MICROSOFT SQL SERVER	72
5.1. Tìm hiểu cơ chế nhân bản, cách thức hoạt động.....	72
5.2. Demo cơ chế nhân bản trên nhiều cụm máy	76
E. PHÂN CÔNG CÔNG VIỆC	76
TÀI LIỆU THAM KHẢO.....	77

1. THIẾT KẾ CSDL PHÂN TÁN

1.1. Thiết kế và mô tả csdl phân tán

CSDL Quản lý các chi nhánh Mini Mart

Mini Mart có 2 chi nhánh tại Thành Phố Hồ Chí Minh. Trụ sở chính của Mini Mart nằm tại chi nhánh ‘Lê Văn Việt, Quận 9’

Lược đồ cơ sở dữ liệu của mỗi chi nhánh của Mini Mart như sau:

BRANCH (BRANCH_ID, BRANCH_NAME, ADDRESS)

Tên từ: Mỗi chi nhánh có mã chi nhánh (BRANCH_ID) dùng để phân biệt các chi

nhánh với nhau ngoài ra còn lưu tên chi nhánh (BRANCH_NAME) và địa chỉ của chi nhánh đó (ADDRESS). Trường tên chi nhánh là duy nhất

EMPLOYEE (EMP_ID, FIRST_NAME, LAST_NAME, GENDER, BIRTHDAY, PHONE, ADDRESS, START_DATE, SALARY, ROLE, BRANCH_ID)

Tên từ: Mỗi nhân viên có mã nhân viên (EMPLOYEE_ID) là duy nhất, tên nhân viên

(FIRST_NAME, LAST_NAME), giới tính (GENDER), ngày sinh (BIRTHDAY), điện thoại (PHONE), địa chỉ (ADDRESS), chức vụ của nhân viên (JOB), ngày vào làm (START_DATE), mức lương của nhân viên (SALARY), mã chi nhánh mà nhân viên đó làm việc (BRANCH_ID).

CUSTOMER (CUS_ID, FIRST_NAME, LAST_NAME, GENDER, ADDRESS, PHONE, BIRTHDAY, REG_DATE, SPENT_MONEY, POINT)

Tên từ: Mỗi khách hàng có mã khách hàng (CUSTOMER_ID) là duy nhất, họ và tên

khách hàng (FIRST_NAME, LAST_NAME), giới tính (GENDER), ngày sinh

(BIRTHDAY), địa chỉ khách hàng (ADDRESS), điện thoại (), ngày đăng ký (PHONE), tổng tiền chi mua (SPENT_MONEY), điểm tích lũy đơn hàng (POINT)

PRODUCT (PRO_ID, PRODUCT_NAME, COUNTRY, ORIGINAL_PRICE, SALE_PRICE, MFG, EXP, PRODUCT_TYPE, VAT, REMAINING_QUANTITY)

Tên từ: mỗi sản phẩm có mã sản phẩm (PRO_ID) là duy nhất, tên sản phẩm (PRODUCT_NAME), nước sản xuất (COUNTRY), giá gốc (ORIGINAL_PRICE), giá bán (SALE_PRICE), ngày sản xuất (MFG), ngày hết hạn (EXP), loại sản phẩm (PRODUCT_TYPE), vat (VAT)

WAREHOUSE_MANAGEMENT (BRANCH_ID, PRO_ID, IMPORTED_DATE, IMPORTED_QUANTITY)

Tên từ: Kho quản lý sản phẩm có mã chi nhánh (BRANCH_ID), mã sản phẩm (PRO_ID), ngày nhập (IMPORTED_DATE), số lượng nhập (IMPORTED_QUANTITY)

WAREHOUSE_SALES (BRANCH_ID, PRO_ID, STATUS)

Tên từ: quản lý tình trạng bán hàng: mã chi nhánh (BRANCH_ID), mã sản phẩm (PRO_ID), tình trạng (STATUS)

BILL (BILL_ID, EMP_ID, CUS_ID, BILL_DATE, TOTAL_MONEY)

Tên từ: hóa đơn bán hàng có mã hóa đơn (BILL_ID), mã nhân viên thanh toán (EMP_ID), mã khách hàng (CUS_ID), ngày hóa đơn (BILL_ID), tổng tiền hóa đơn (TOTAL_MONEY)

BILL_DETAILS (BILL_ID, PRO_ID, AMOUNT)

Tên từ: chi tiết hóa đơn có mã hóa đơn (BILL_ID), mã sản phẩm (PRO_ID), và số lượng sản phẩm (AMOUNT)

1.2. Kiến trúc phân mảnh

1.2.1. Mô tả phân mảnh

- Quan hệ BRANCH là phân mảnh ngang chính
- Quan hệ EMPLOYEE, BILL, BILL_DETAILS là phân mảnh ngang dẫn xuất
- Quan hệ WAREHOUSE được phân mảnh hỗn hợp thành WAREHOUSE_MANAGEMENT và WAREHOUSE_SALES. Trong đó:
 - Quan hệ WAREHOUSE_MANAGEMENT chứa các thông tin dùng để quản lý thông tin nhập sản phẩm;
 - Quan hệ WAREHOUSE_SALES quản lý thông tin trạng thái bán hàng của sản phẩm
- Quan hệ CUSTOMER, PRODUCT được nhân bản tại tất cả chi nhánh

1.2.2. Minh họa trên 2 chi nhánh

* Quan hệ BRANCH là phân mảnh ngang chính

CN1.BRANCH = $\delta_{(BRANCH_NAME = 'Mini mart Quan 9')}$ BRANCH

CN2.BRANCH = $\delta_{(BRANCH_NAME = 'Mini mart chi nhánh Quan 1')}$ BRANCH

* Quan hệ EMPLOYEE, BILL, BILL_DETAILS là phân mảnh ngang dẫn xuất

CN1.EMPLOYEE = EMPLOYEE \bowtie_{BRANCH_ID} CN1.BRANCH

CN2.EMPLOYEE = EMPLOYEE \bowtie_{BRANCH_ID} CN2.BRANCH

CN1.BILL_ID = BILL_ID \bowtie_{EMP_ID} CN1.EMPLOYEE

CN2.BILL_ID = BILL_ID \bowtie_{EMP_ID} CN2.EMPLOYEE

CN1.BILL_DETAILS = BILL_DETAILS \bowtie_{BILL_ID} CN1.BILL_ID

CN2.TRANSACTION = BILL_DETAILS \bowtie_{BILL_ID} CN2.BILL_ID

*Quan hệ WAREHOUSE được phân mảnh hỗn hợp thành

WAREHOUSE_MANAGEMENT và WAREHOUSE_SALES. Trong đó:

CN1. WAREHOUSE_MANAGEMENT =

$\pi_{(BRANCH_ID, PRO_ID, IMPORTED_DATE, IMPORTED_QUANTITY)}(WAREHOUSE$
 $\bowtie_{BRANCH_ID} CN1.BRANCH)$

CN1. WAREHOUSE_SALES = $\pi_{(BRANCH_ID, PRO_ID, STATUS)}$
 $(WAREHOUSE \bowtie_{BRANCH_ID} CN1.BRANCH)$

CN2. WAREHOUSE_MANAGEMENT =

$\pi_{(BRANCH_ID, PRO_ID, IMPORTED_DATE, IMPORTED_QUANTITY)}(WAREHOUSE$
 $\bowtie_{BRANCH_ID} CN1.BRANCH)$

CN2. WAREHOUSE_SALES = $\pi_{(BRANCH_ID, PRO_ID, STATUS)}$
 $(WAREHOUSE \bowtie_{BRANCH_ID} CN1.BRANCH)$

*Quan hệ CUSTOMER, PRODUCT được nhân bản tại tất cả chi nhánh

1.3. Xây dựng csdl phân tán

1.3.1. Dll

CN1	CN2
<pre>CREATE TABLE CN1.BRANCH(BRANCH_ID VARCHAR2(5) CONSTRAINT BRANCH_PK PRIMARY KEY, BRANCH_NAME VARCHAR2(200), ADDRESS VARCHAR2(200));</pre>	<pre>CREATE TABLE CN2.BRANCH(BRANCH_ID VARCHAR2(5) CONSTRAINT BRANCH_PK PRIMARY KEY, BRANCH_NAME VARCHAR2(200), ADDRESS VARCHAR2(200));</pre>
<pre>CREATE TABLE CN1.EMPLOYEE(EMP_ID NUMBER CONSTRAINT EMPLOYEE_PK PRIMARY KEY, FIRST_NAME VARCHAR2(200),</pre>	<pre>CREATE TABLE CN2.EMPLOYEE(EMP_ID NUMBER CONSTRAINT EMPLOYEE_PK PRIMARY KEY, FIRST_NAME VARCHAR2(200),</pre>

<p> LAST_NAME VARCHAR2(200), GENDER VARCHAR(10), BIRTHDAY DATE, PHONE VARCHAR2(200), ADDRESS VARCHAR2(200), START_DATE DATE, SALARY NUMBER, ROLE VARCHAR2(200), BRANCH_ID VARCHAR2(5), CONSTRAINT FK_EMP_BRANCH FOREIGN KEY(BRANCH_ID) REFERENCES BRANCH(BRANCH_ID)); </p>	<p> LAST_NAME VARCHAR2(200), GENDER VARCHAR(10), BIRTHDAY DATE, PHONE VARCHAR2(200), ADDRESS VARCHAR2(200), START_DATE DATE, SALARY NUMBER, ROLE VARCHAR2(200), BRANCH_ID VARCHAR2(5), CONSTRAINT FK_EMP_BRANCH FOREIGN KEY(BRANCH_ID) REFERENCES BRANCH(BRANCH_ID)); </p>
<p> CREATE TABLE CN1.CUSTOMER (CUS_ID NUMBER CONSTRAINT PK_STUDENT PRIMARY KEY, FIRST_NAME VARCHAR(20), LAST_NAME VARCHAR(20), GENDER VARCHAR(10), ADDRESS VARCHAR(200), PHONE VARCHAR(20), BIRTHDAY DATE, REG_DATE DATE, </p>	<p> CREATE TABLE CN2.CUSTOMER (CUS_ID NUMBER CONSTRAINT PK_STUDENT PRIMARY KEY, FIRST_NAME VARCHAR(20), LAST_NAME VARCHAR(20), GENDER VARCHAR(10), ADDRESS VARCHAR(200), PHONE VARCHAR(20), BIRTHDAY DATE, REG_DATE DATE, SPENT_MONEY NUMBER, POINT INT); </p>

SPENT_MONEY NUMBER, POINT INT);	
CREATE TABLE CN1.PRODUCT (PRO_ID NUMBER CONSTRAINT PK_PRO PRIMARY KEY, PRODUCT_NAME VARCHAR(200), COUNTRY VARCHAR(50), ORIGINAL_PRICE NUMBER, SALE_PRICE NUMBER, MFG DATE, EXP DATE, PRODUCT_TYPE VARCHAR(200), VAT NUMBER, REMAINING_QUANTITY NUMBER);	CREATE TABLE CN2.PRODUCT (PRO_ID NUMBER CONSTRAINT PK_PRO PRIMARY KEY, PRODUCT_NAME VARCHAR(200), COUNTRY VARCHAR(50), ORIGINAL_PRICE NUMBER, SALE_PRICE NUMBER, MFG DATE, EXP DATE, PRODUCT_TYPE VARCHAR(200), VAT NUMBER, REMAINING_QUANTITY NUMBER);
CREATE TABLE CN1.WAREHOUSE_MANAGEMENT (BRANCH_ID VARCHAR2(5), PRO_ID NUMBER, IMPORTED_DATE DATE,	CREATE TABLE CN2.WAREHOUSE_MANAGEMENT (BRANCH_ID VARCHAR2(5), PRO_ID NUMBER, IMPORTED_DATE DATE,

<pre> IMPORTED_QUANTITY NUMBER, CONSTRAINT PK_WM PRIMARY KEY(BRANCH_ID, PRO_ID), CONSTRAINT FK_WM_BRANCH FOREIGN KEY (BRANCH_ID) REFERENCES BRANCH(BRANCH_ID), CONSTRAINT FK_WM_PRO FOREIGN KEY (PRO_ID) REFERENCES PRODUCT(PRO_ID)); </pre>	<pre> IMPORTED_QUANTITY NUMBER, CONSTRAINT PK_WM PRIMARY KEY(BRANCH_ID, PRO_ID), CONSTRAINT FK_WM_BRANCH FOREIGN KEY (BRANCH_ID) REFERENCES BRANCH(BRANCH_ID), CONSTRAINT FK_WM_PRO FOREIGN KEY (PRO_ID) REFERENCES PRODUCT(PRO_ID)); </pre>
<pre> CREATE TABLE CN1.WAREHOUSE_SALES (BRANCH_ID VARCHAR2(5), PRO_ID NUMBER, STATUS VARCHAR2(200), CONSTRAINT WS_PK PRIMARY KEY(BRANCH_ID, PRO_ID), CONSTRAINT FK_WS_BRANCH FOREIGN KEY (BRANCH_ID) REFERENCES BRANCH(BRANCH_ID), CONSTRAINT FK_WS_PRO FOREIGN KEY (PRO_ID) REFERENCES PRODUCT(PRO_ID)); </pre>	<pre> CREATE TABLE CN2.WAREHOUSE_SALES (BRANCH_ID VARCHAR2(5), PRO_ID NUMBER, STATUS VARCHAR2(200), CONSTRAINT WS_PK PRIMARY KEY(BRANCH_ID, PRO_ID), CONSTRAINT FK_WS_BRANCH FOREIGN KEY (BRANCH_ID) REFERENCES BRANCH(BRANCH_ID), CONSTRAINT FK_WS_PRO FOREIGN KEY (PRO_ID) REFERENCES PRODUCT(PRO_ID)); </pre>

<pre> CREATE TABLE CN1.BILL (BILL_ID NUMBER CONSTRAINT PK_BILL PRIMARY KEY, EMP_ID NUMBER, CUS_ID NUMBER, BILL_DATE DATE, TOTAL_MONEY NUMBER, CONSTRAINT FK_BILL_EMP FOREIGN KEY(EMP_ID) REFERENCES EMPLOYEE(EMP_ID), CONSTRAINT FK_BILL_CUS FOREIGN KEY(CUS_ID) REFERENCES CUSTOMER(CUS_ID)); </pre>	<pre> CREATE TABLE CN2.BILL (BILL_ID NUMBER CONSTRAINT PK_BILL PRIMARY KEY, EMP_ID NUMBER, CUS_ID NUMBER, BILL_DATE DATE, TOTAL_MONEY NUMBER, CONSTRAINT FK_BILL_EMP FOREIGN KEY(EMP_ID) REFERENCES EMPLOYEE(EMP_ID), CONSTRAINT FK_BILL_CUS FOREIGN KEY(CUS_ID) REFERENCES CUSTOMER(CUS_ID)); </pre>
<pre> CREATE TABLE CN1.BILL_DETAILS (BILL_ID NUMBER, PRO_ID NUMBER, AMOUNT INT, CONSTRAINT PK_B_DETAILS PRIMARY KEY(BILL_ID, PRO_ID), CONSTRAINT FK_B_DETAILS_BILL FOREIGN </pre>	<pre> CREATE TABLE CN1.BILL_DETAILS (BILL_ID NUMBER, PRO_ID NUMBER, AMOUNT INT, CONSTRAINT PK_B_DETAILS PRIMARY KEY(BILL_ID, PRO_ID), CONSTRAINT FK_B_DETAILS_BILL FOREIGN </pre>

<p>KEY(BILL_ID) REFERENCES BILL(BILL_ID), CONSTRAINT FK_B_DETAILS_PRO FOREIGN KEY(PRO_ID) REFERENCES PRODUCT(PRO_ID));</p>	<p>KEY(BILL_ID) REFERENCES BILL(BILL_ID), CONSTRAINT FK_B_DETAILS_PRO FOREIGN KEY(PRO_ID) REFERENCES PRODUCT(PRO_ID));</p>
---	---

1.3.2. Dml

CN1	CN2
<p>INSERT INTO CN1.BRANCH(BRANCH_ID, BRANCH_NAME, ADDRESS) VALUES ('CN1', 'Mini mart Quan 9', '5, Le Van Viet, Quan 9, TPHCM');</p>	<p>INSERT INTO CN2.BRANCH(BRANCH_ID, BRANCH_NAME, ADDRESS) VALUES ('CN2', 'Mini mart chi nhanh Quan 1', '55, Ly Thanh Tong, Quan 1, TPHCM');</p>
<p>INSERT INTO CN1.EMPLOYEE VALUES (CN1.EMP_ID_SEQUENCE.nextval, 'Nguyen Thi My', 'Tran','Female', TO_DATE('05/05/2002', 'dd/mm/yyyy'), '0921231741', 'KTX khu B', TO_DATE('21/02/2022', 'dd/mm/yyyy'), 10000000,'Manager', 'CN1');</p> <p>INSERT INTO CN1.EMPLOYEE VALUES (CN1.EMP_ID_SEQUENCE.nextval, 'Ton Nu Tu', 'Quyen','Female',</p>	<p>INSERT INTO CN2.EMPLOYEE VALUES (CN2.EMP_ID_SEQUENCE.nextval, 'Nguyen Thi My', 'Tam','Female', TO_DATE('05/05/2002', 'dd/mm/yyyy'), '0921231741', 'KTX khu B', TO_DATE('21/02/2022', 'dd/mm/yyyy'), 10000000,'Manager', 'CN2');</p> <p>INSERT INTO CN2.EMPLOYEE VALUES (CN2.EMP_ID_SEQUENCE.nextval, 'Ton Nu Tu', 'Tran','Female',</p>

<p>TO_DATE('05/01/2002', 'dd/mm/yyyy'), '0977164279', 'KTX khu A', TO_DATE('21/02/2022', 'dd/mm/yyyy'), 5000000, 'Cashier', 'CN1');</p> <p>INSERT INTO CN1.EMPLOYEE VALUES (CN1.EMP_ID_SEQUENCE.nextval, 'Ngo Thi', 'Anh', 'Female', TO_DATE('25/08/2002', 'dd/mm/yyyy'), '0356315873', '731 Tran Hung Dao, Q5, TpHCM', TO_DATE('21/02/2022', 'dd/mm/yyyy'), 5000000, 'Cashier', 'CN1');</p> <p>INSERT INTO CN1.EMPLOYEE VALUES (CN1.EMP_ID_SEQUENCE.nextval, 'Nguyen Anh', 'Viet', 'Male', TO_DATE('05/07/2002', 'dd/mm/yyyy'), '0585784600', '45 Nguyen Canh Chan, Q1, TPHCM', TO_DATE('21/02/2022', 'dd/mm/yyyy'), 5000000, 'Storekeeper', 'CN1');</p> <p>INSERT INTO CN1.EMPLOYEE VALUES (CN1.EMP_ID_SEQUENCE.nextval, 'Nguyen Thi Minh', 'Triet', 'Female', TO_DATE('05/09/2002',</p>	<p>TO_DATE('05/01/2002', 'dd/mm/yyyy'), '0977164279', 'KTX khu A', TO_DATE('21/02/2022', 'dd/mm/yyyy'), 5000000, 'Cashier', 'CN2');</p> <p>INSERT INTO CN2.EMPLOYEE VALUES (CN2.EMP_ID_SEQUENCE.nextval, 'Ngo Thi', 'Bap', 'Female', TO_DATE('25/08/2002', 'dd/mm/yyyy'), '0356315873', '731 Tran Hung Dao, Q5, TpHCM', TO_DATE('21/02/2022', 'dd/mm/yyyy'), 5000000, 'Cashier', 'CN2');</p> <p>INSERT INTO CN2.EMPLOYEE VALUES (CN2.EMP_ID_SEQUENCE.nextval, 'Nguyen Anh', 'Kiet', 'Male', TO_DATE('05/07/2002', 'dd/mm/yyyy'), '0585784600', '45 Nguyen Canh Chan, Q1, TPHCM', TO_DATE('21/02/2022', 'dd/mm/yyyy'), 5000000, 'Storekeeper', 'CN2');</p> <p>INSERT INTO CN2.EMPLOYEE VALUES (CN2.EMP_ID_SEQUENCE.nextval, 'Nguyen Thi Minh', 'Huong', 'Female', TO_DATE('05/09/2002',</p>
--	--

<p>'dd/mm/yyyy'), '0932342312', '50/34 Le Dai Hanh, Q10, TPHCM', TO_DATE('21/02/2022', 'dd/mm/yyyy'), 5000000, 'Salesperson', 'CN1'); INSERT INTO CN1.EMPLOYEE VALUES (CN1.EMP_ID_SEQUENCE.nextval, 'Tran Anh', 'Huy','Male', TO_DATE('26/09/2002', 'dd/mm/yyyy'), '0916430073', 'KTX khu B', TO_DATE('21/02/2022', 'dd/mm/yyyy'), 7000000, 'Salesperson', 'CN1'); INSERT INTO CN1.EMPLOYEE VALUES (CN1.EMP_ID_SEQUENCE.nextval, 'Phan Hong', 'Anh','Female', TO_DATE('20/08/2001', 'dd/mm/yyyy'), '0944562311', '16 Le Van Viet, Q9, TPHCM', TO_DATE('21/02/2022', 'dd/mm/yyyy'), 7000000, 'Customer Service Assistant', 'CN1'); INSERT INTO CN1.EMPLOYEE VALUES (CN1.EMP_ID_SEQUENCE.nextval, 'Thai Tang', 'Duc','Male', TO_DATE('16/10/2002', 'dd/mm/yyyy'), '0931231756', 'KTX</p>	<p>'dd/mm/yyyy'), '0932342312', '50/34 Le Dai Hanh, Q10, TPHCM', TO_DATE('21/02/2022', 'dd/mm/yyyy'), 5000000, 'Salesperson', 'CN2'); INSERT INTO CN2.EMPLOYEE VALUES (CN2.EMP_ID_SEQUENCE.nextval, 'Tran Anh', 'Tuan','Male', TO_DATE('26/09/2002', 'dd/mm/yyyy'), '0916430073', 'KTX khu B', TO_DATE('21/02/2022', 'dd/mm/yyyy'), 7000000, 'Salesperson', 'CN2'); INSERT INTO CN2.EMPLOYEE VALUES (CN2.EMP_ID_SEQUENCE.nextval, 'Phan Hong', 'Duc','Female', TO_DATE('20/08/2001', 'dd/mm/yyyy'), '0944562311', '16 Le Van Viet, Q9, TPHCM', TO_DATE('21/02/2022', 'dd/mm/yyyy'), 7000000, 'Customer Service Assistant', 'CN2'); INSERT INTO CN2.EMPLOYEE VALUES (CN2.EMP_ID_SEQUENCE.nextval, 'Thai Tang', 'Minh','Male', TO_DATE('16/10/2002', 'dd/mm/yyyy'), '0931231756', 'KTX</p>
--	--

<p>khu A', TO_DATE('21/02/2022', 'dd/mm/yyyy'), 5000000, 'Quality Checker', 'CN1');</p> <p>INSERT INTO CN1.EMPLOYEE VALUES (CN1.EMP_ID_SEQUENCE.nextval, 'Le Nguyen Thuy', 'Vi', 'Female', TO_DATE('06/06/2001', 'dd/mm/yyyy'), '0948965678', 'KTX khu B', TO_DATE('21/02/2022', 'dd/mm/yyyy'), 5000000, 'Sanitation Worker', 'CN1');</p> <p>INSERT INTO CN1.EMPLOYEE VALUES (CN1.EMP_ID_SEQUENCE.nextval, 'Nguyen Thi Kim', 'Hieu', 'Female', TO_DATE('28/08/2002', 'dd/mm/yyyy'), '0931231756', 'KTX khu B', TO_DATE('21/02/2022', 'dd/mm/yyyy'), 5000000, 'Security Guard', 'CN1');</p>	<p>khu A', TO_DATE('21/02/2022', 'dd/mm/yyyy'), 5000000, 'Quality Checker', 'CN2');</p> <p>INSERT INTO CN2.EMPLOYEE VALUES (CN2.EMP_ID_SEQUENCE.nextval, 'Le Nguyen Thuy', 'Lieu', 'Female', TO_DATE('06/06/2001', 'dd/mm/yyyy'), '0948965678', 'KTX khu B', TO_DATE('21/02/2022', 'dd/mm/yyyy'), 5000000, 'Sanitation Worker', 'CN2');</p> <p>INSERT INTO CN2.EMPLOYEE VALUES (CN2.EMP_ID_SEQUENCE.nextval, 'Nguyen Thi Kim', 'Ngan', 'Female', TO_DATE('28/08/2002', 'dd/mm/yyyy'), '0931231756', 'KTX khu B', TO_DATE('21/02/2022', 'dd/mm/yyyy'), 5000000, 'Security Guard', 'CN2');</p>
<p>INSERT INTO CN1.CUSTOMER VALUES(CN1.CUS_ID_SEQUENCE.nextval, 'Tran Minh', 'Tien', 'Male', '117/2 Nguyen Trai, Q5, TpHCM', '0883644231', TO_DATE('15/04/2002', 'dd/mm/yyyy'),</p>	<p>INSERT INTO CN2.CUSTOMER VALUES(CN2.CUS_ID_SEQUENCE.nextval, 'Tran Minh', 'Tien', 'Male', '117/2 Nguyen Trai, Q5, TpHCM', '0883644231', TO_DATE('15/04/2002', 'dd/mm/yyyy'), TO_DATE('01/09/2022', 'dd/mm/yyyy'), 279072, 0);</p>

<p>TO_DATE('01/09/2022', 'dd/mm/yyyy'), 279072, 0);</p> <p>INSERT INTO CN1.CUSTOMER</p> <p>VALUES(CN1.CUS_ID_SEQ UENCE.nextval, 'Nguyen Bang', 'Huu', 'Male', '731 Tran Hung Dao, Q5, TpHCM', '0975244479', TO_DATE('15/09/2000', 'dd/mm/yyyy'), TO_DATE('01/09/2022', 'dd/mm/yyyy'), 26163, 0);</p> <p>INSERT INTO CN1.CUSTOMER</p> <p>VALUES(CN1.CUS_ID_SEQ UENCE.nextval, 'Nguyen Huu', 'Tho', 'Male', '23/5 Nguyen Trai, Q5, TpHCM', '0361234578', TO_DATE('18/06/2001', 'dd/mm/yyyy'), TO_DATE('01/09/2022', 'dd/mm/yyyy'), 98838, 0);</p> <p>INSERT INTO CN1.CUSTOMER</p> <p>VALUES(CN1.CUS_ID_SEQ UENCE.nextval, 'Nguyen Phuong', 'Thanh', 'Female', '27/2 Nguyen Trai, Q5, TpHCM', '0365238774', TO_DATE('24/06/1999', 'dd/mm/yyyy'),</p>	<p>INSERT INTO CN2.CUSTOMER</p> <p>VALUES(CN2.CUS_ID_SEQUENC E.nextval, 'Nguyen Bang', 'Huu', 'Male', '731 Tran Hung Dao, Q5, TpHCM', '0975244479', TO_DATE('15/09/2000', 'dd/mm/yyyy'), TO_DATE('01/09/2022', 'dd/mm/yyyy'), 26163, 0);</p> <p>INSERT INTO CN2.CUSTOMER</p> <p>VALUES(CN2.CUS_ID_SEQUENC E.nextval, 'Nguyen Huu', 'Tho', 'Male', '23/5 Nguyen Trai, Q5, TpHCM', '0361234578', TO_DATE('18/06/2001', 'dd/mm/yyyy'), TO_DATE('01/09/2022', 'dd/mm/yyyy'), 98838, 0);</p> <p>INSERT INTO CN2.CUSTOMER</p> <p>VALUES(CN2.CUS_ID_SEQUENC E.nextval, 'Nguyen Phuong', 'Thanh', 'Female', '27/2 Nguyen Trai, Q5, TpHCM', '0365238774', TO_DATE('24/06/1999', 'dd/mm/yyyy'), TO_DATE('02/09/2022', 'dd/mm/yyyy'), 96900, 0);</p> <p>INSERT INTO CN2.CUSTOMER</p> <p>VALUES(CN2.CUS_ID_SEQUENC E.nextval, 'Le Thi Tuong', 'Vi',</p>
---	---

<p>TO_DATE('02/09/2022', 'dd/mm/yyyy'), 96900, 0);</p> <p>INSERT INTO CN1.CUSTOMER</p> <p>VALUES(CN1.CUS_ID_SEQ UENCE.nextval, 'Le Thi Tuong', 'Vi', 'Female', '45 Nguyen Canh Chan, Q1, TPHCM', '0938776266', TO_DATE('22/02/2002', 'dd/mm/yyyy'), TO_DATE('02/09/2022', 'dd/mm/yyyy'), 222870, 1);</p> <p>INSERT INTO CN1.CUSTOMER</p> <p>VALUES(CN1.CUS_ID_SEQ UENCE.nextval, 'Nguyen Thanh', 'Duy', 'Male', '50/34 Le Dai Hanh, Q10, TPHCM', '0938826866', TO_DATE('18/01/2001', 'dd/mm/yyyy'), TO_DATE('02/09/2022', 'dd/mm/yyyy'), 38760, 0);</p> <p>INSERT INTO CN1.CUSTOMER</p> <p>VALUES(CN1.CUS_ID_SEQ UENCE.nextval, 'Vo Minh', 'Trong', 'Male', '837 Le Hong Phong,Q5,TPHCM', '0937825255', TO_DATE('14/10/2001', 'dd/mm/yyyy'),</p>	<p>'Female', '45 Nguyen Canh Chan, Q1, TPHCM', '0938776266', TO_DATE('22/02/2002', 'dd/mm/yyyy'), TO_DATE('02/09/2022', 'dd/mm/yyyy'), 222870, 1);</p> <p>INSERT INTO CN2.CUSTOMER</p> <p>VALUES(CN2.CUS_ID_SEQUENC E.nextval, 'Nguyen Thanh', 'Duy', 'Male', '50/34 Le Dai Hanh, Q10, TPHCM', '0938826866', TO_DATE('18/01/2001', 'dd/mm/yyyy'), TO_DATE('02/09/2022', 'dd/mm/yyyy'), 38760, 0);</p> <p>INSERT INTO CN2.CUSTOMER</p> <p>VALUES(CN2.CUS_ID_SEQUENC E.nextval, 'Vo Minh', 'Trong', 'Male', '837 Le Hong Phong,Q5,TPHCM', '0937825255', TO_DATE('14/10/2001', 'dd/mm/yyyy'), TO_DATE('03/09/2022', 'dd/mm/yyyy'), 14535, 0);</p> <p>INSERT INTO CN2.CUSTOMER</p> <p>VALUES(CN2.CUS_ID_SEQUENC E.nextval, 'Dinh Van', 'Quyet', 'Male', '50/34 Le Dai Hanh, Q10, TPHCM', '0837822285', TO_DATE('16/04/2002',</p>
--	--

<p>TO_DATE('03/09/2022', 'dd/mm/yyyy'), 14535, 0);</p> <p>INSERT INTO CN1.CUSTOMER</p> <p>VALUES(CN1.CUS_ID_SEQ UENCE.nextval, 'Dinh Van', 'Quyet', 'Male', '50/34 Le Dai Hanh, Q10, TPHCM', '0837822285', TO_DATE('16/04/2002', 'dd/mm/yyyy'), TO_DATE('03/09/2022', 'dd/mm/yyyy'), 7752, 0);</p> <p>INSERT INTO CN1.CUSTOMER</p> <p>VALUES(CN1.CUS_ID_SEQ UENCE.nextval, 'Ho Nguyen Thu', 'An', 'Female', '34/34B Nguyen Trai, Q5, TPHCM', '0237825224', TO_DATE('9/11/2000', 'dd/mm/yyyy'), TO_DATE('03/09/2022', 'dd/mm/yyyy'), 96900, 0);</p> <p>INSERT INTO CN1.CUSTOMER</p> <p>VALUES(CN1.CUS_ID_SEQ UENCE.nextval, 'Nguyen Anh', 'Phi', 'Male', '227 Nguyen Van Cu, Q5, TPHCM', '0937885255', TO_DATE('06/12/2002', 'dd/mm/yyyy'),</p>	<p>'dd/mm/yyyy'), TO_DATE('03/09/2022', 'dd/mm/yyyy'), 7752, 0);</p> <p>INSERT INTO CN2.CUSTOMER</p> <p>VALUES(CN2.CUS_ID_SEQUENC E.nextval, 'Ho Nguyen Thu', 'An', 'Female', '34/34B Nguyen Trai, Q5, TPHCM', '0237825224', TO_DATE('9/11/2000', 'dd/mm/yyyy'), TO_DATE('03/09/2022', 'dd/mm/yyyy'), 96900, 0);</p> <p>INSERT INTO CN2.CUSTOMER</p> <p>VALUES(CN2.CUS_ID_SEQUENC E.nextval, 'Nguyen Anh', 'Phi', 'Male', '227 Nguyen Van Cu, Q5, TPHCM', '0937885255', TO_DATE('06/12/2002', 'dd/mm/yyyy'), TO_DATE('04/09/2022', 'dd/mm/yyyy'), 145350, 1);</p> <p>INSERT INTO CN2.CUSTOMER</p> <p>VALUES(CN2.CUS_ID_SEQUENC E.nextval, 'Tran Minh', 'Minh', 'Male', '117/2 Nguyen Trai, Q5, TpHCM', '0883644231', TO_DATE('15/04/2002', 'dd/mm/yyyy'), TO_DATE('01/09/2022', 'dd/mm/yyyy'), 279072, 0);</p>
--	--

<p>TO_DATE('04/09/2022', 'dd/mm/yyyy'), 145350, 1);</p> <p>INSERT INTO CN1.CUSTOMER</p> <p>VALUES(CN1.CUS_ID_SEQ UENCE.nextval, 'Tran Minh', 'Minh', 'Male', '117/2 Nguyen Trai, Q5, TpHCM', '0883644231', TO_DATE('15/04/2002', 'dd/mm/yyyy'), TO_DATE('01/09/2022', 'dd/mm/yyyy'), 279072, 0);</p> <p>INSERT INTO CN1.CUSTOMER</p> <p>VALUES(CN1.CUS_ID_SEQ UENCE.nextval, 'Nguyen Bang', 'Ha', 'Male', '731 Tran Hung Dao, Q5, TpHCM', '0975244479', TO_DATE('15/09/2000', 'dd/mm/yyyy'), TO_DATE('01/09/2022', 'dd/mm/yyyy'), 26163, 0);</p> <p>INSERT INTO CN1.CUSTOMER</p> <p>VALUES(CN1.CUS_ID_SEQ UENCE.nextval, 'Nguyen Bang', 'Ha', 'Male', '731 Tran Hung Dao, Q5, TpHCM', '0975244479', TO_DATE('15/09/2000', 'dd/mm/yyyy'), TO_DATE('01/09/2022', 'dd/mm/yyyy'), 26163, 0);</p> <p>INSERT INTO CN1.CUSTOMER</p> <p>VALUES(CN1.CUS_ID_SEQ UENCE.nextval, 'Nguyen Huu', 'Tham', 'Male', '23/5 Nguyen Trai, Q5, TpHCM', '0361234578', TO_DATE('18/06/2001', 'dd/mm/yyyy'),</p>	<p>INSERT INTO CN2.CUSTOMER</p> <p>VALUES(CN2.CUS_ID_SEQUENC E.nextval, 'Nguyen Bang', 'Ha', 'Male', '731 Tran Hung Dao, Q5, TpHCM', '0975244479', TO_DATE('15/09/2000', 'dd/mm/yyyy'), TO_DATE('01/09/2022', 'dd/mm/yyyy'), 26163, 0);</p> <p>INSERT INTO CN2.CUSTOMER</p> <p>VALUES(CN2.CUS_ID_SEQUENC E.nextval, 'Nguyen Huu', 'Tham', 'Male', '23/5 Nguyen Trai, Q5, TpHCM', '0361234578', TO_DATE('18/06/2001', 'dd/mm/yyyy'), TO_DATE('01/09/2022', 'dd/mm/yyyy'), 98838, 0);</p> <p>INSERT INTO CN2.CUSTOMER</p> <p>VALUES(CN2.CUS_ID_SEQUENC E.nextval, 'Nguyen Phuong', 'Thao', 'Female', '27/2 Nguyen Trai, Q5, TpHCM', '0365238774', TO_DATE('24/06/1999', 'dd/mm/yyyy'), TO_DATE('02/09/2022', 'dd/mm/yyyy'), 96900, 0);</p> <p>INSERT INTO CN2.CUSTOMER</p> <p>VALUES(CN2.CUS_ID_SEQUENC E.nextval, 'Le Thi Tuong', 'An',</p>
--	--

<p>TO_DATE('01/09/2022', 'dd/mm/yyyy'), 98838, 0);</p> <p>INSERT INTO CN1.CUSTOMER</p> <p>VALUES(CN1.CUS_ID_SEQ UENCE.nextval, 'Nguyen Phuong', 'Thao', 'Female', '27/2 Nguyen Trai, Q5, TpHCM', '0365238774', TO_DATE('24/06/1999', 'dd/mm/yyyy'), TO_DATE('02/09/2022', 'dd/mm/yyyy'), 96900, 0);</p> <p>INSERT INTO CN1.CUSTOMER</p> <p>VALUES(CN1.CUS_ID_SEQ UENCE.nextval, 'Le Thi Tuong', 'An', 'Female', '45 Nguyen Canh Chan, Q1, TPHCM', '0938776266', TO_DATE('22/02/2002', 'dd/mm/yyyy'), TO_DATE('02/09/2022', 'dd/mm/yyyy'), 222870, 1);</p> <p>INSERT INTO CN1.CUSTOMER</p> <p>VALUES(CN1.CUS_ID_SEQ UENCE.nextval, 'Nguyen Thanh', 'Khiet', 'Male', '50/34 Le Dai Hanh, Q10, TPHCM', '0938826866', TO_DATE('18/01/2001', 'dd/mm/yyyy'),</p>	<p>'Female', '45 Nguyen Canh Chan, Q1, TPHCM', '0938776266', TO_DATE('22/02/2002', 'dd/mm/yyyy'), TO_DATE('02/09/2022', 'dd/mm/yyyy'), 222870, 1);</p> <p>INSERT INTO CN2.CUSTOMER</p> <p>VALUES(CN2.CUS_ID_SEQUENC E.nextval, 'Nguyen Thanh', 'Khiet', 'Male', '50/34 Le Dai Hanh, Q10, TPHCM', '0938826866', TO_DATE('18/01/2001', 'dd/mm/yyyy'), TO_DATE('02/09/2022', 'dd/mm/yyyy'), 38760, 0);</p> <p>INSERT INTO CN2.CUSTOMER</p> <p>VALUES(CN2.CUS_ID_SEQUENC E.nextval, 'Vo Minh', 'Tam', 'Male', '837 Le Hong Phong,Q5,TPHCM', '0937825255', TO_DATE('14/10/2001', 'dd/mm/yyyy'), TO_DATE('03/09/2022', 'dd/mm/yyyy'), 14535, 0);</p> <p>INSERT INTO CN2.CUSTOMER</p> <p>VALUES(CN2.CUS_ID_SEQUENC E.nextval, 'Dinh Van', 'Toan', 'Male', '50/34 Le Dai Hanh, Q10, TPHCM', '0837822285', TO_DATE('16/04/2002',</p>
--	---

<p>TO_DATE('02/09/2022', 'dd/mm/yyyy'), 38760, 0);</p> <p>INSERT INTO CN1.CUSTOMER</p> <p>VALUES(CN1.CUS_ID_SEQ UENCE.nextval, 'Vo Minh', 'Tam', 'Male', '837 Le Hong Phong,Q5,TPHCM', '0937825255', TO_DATE('14/10/2001', 'dd/mm/yyyy'), TO_DATE('03/09/2022', 'dd/mm/yyyy'), 14535, 0);</p> <p>INSERT INTO CN1.CUSTOMER</p> <p>VALUES(CN1.CUS_ID_SEQ UENCE.nextval, 'Dinh Van', 'Toan', 'Male', '50/34 Le Dai Hanh, Q10, TPHCM', '0837822285', TO_DATE('16/04/2002', 'dd/mm/yyyy'), TO_DATE('03/09/2022', 'dd/mm/yyyy'), 7752, 0);</p> <p>INSERT INTO CN1.CUSTOMER</p> <p>VALUES(CN1.CUS_ID_SEQ UENCE.nextval, 'Ho Nguyen Thu', 'Minh', 'Female', '34/34B Nguyen Trai, Q5, TPHCM', '0237825224', TO_DATE('9/11/2000', 'dd/mm/yyyy'),</p>	<p>'dd/mm/yyyy'), TO_DATE('03/09/2022', 'dd/mm/yyyy'), 7752, 0);</p> <p>INSERT INTO CN2.CUSTOMER</p> <p>VALUES(CN2.CUS_ID_SEQUENC E.nextval, 'Ho Nguyen Thu', 'Minh', 'Female', '34/34B Nguyen Trai, Q5, TPHCM', '0237825224', TO_DATE('9/11/2000', 'dd/mm/yyyy'), TO_DATE('03/09/2022', 'dd/mm/yyyy'), 96900, 0);</p> <p>INSERT INTO CN2.CUSTOMER</p> <p>VALUES(CN2.CUS_ID_SEQUENC E.nextval, 'Nguyen Anh', 'Tu', 'Male', '227 Nguyen Van Cu, Q5, TPHCM', '0937885255', TO_DATE('06/12/2002', 'dd/mm/yyyy'), TO_DATE('04/09/2022', 'dd/mm/yyyy'), 145350, 1);</p>
--	--

<p>TO_DATE('03/09/2022', 'dd/mm/yyyy'), 96900, 0);</p> <p>INSERT INTO CN1.CUSTOMER</p> <p>VALUES(CN1.CUS_ID_SEQ UENCE.nextval, 'Nguyen Anh', 'Tu', 'Male', '227 Nguyen Van Cu, Q5, TPHCM', '0937885255', TO_DATE('06/12/2002', 'dd/mm/yyyy'), TO_DATE('04/09/2022', 'dd/mm/yyyy'), 145350, 1);</p>	
<p>INSERT INTO CN1.PRODUCT VALUES (CN1.PRO_ID_SEQUENCE.nextval, ' Book', 'China', 5000, 8000, TO_DATE('01/01/2022', 'dd/mm/yyyy'), null, 'Requisite', 2, 25);</p> <p>INSERT INTO CN1.PRODUCT VALUES (CN1.PRO_ID_SEQUENCE.nextval, 'Pen', 'USA', 3000, 5000, TO_DATE('01/01/2022', 'dd/mm/yyyy'), null, 'Requisite', 2, 27);</p> <p>INSERT INTO CN1.PRODUCT VALUES (CN1.PRO_ID_SEQUENCE.nextval, 'Soap', 'France', 10000, 15000,</p>	<p>INSERT INTO CN2.PRODUCT VALUES (CN2.PRO_ID_SEQUENCE.nextval, ' Book', 'China', 5000, 8000, TO_DATE('01/01/2022', 'dd/mm/yyyy'), null, 'Requisite', 2, 25);</p> <p>INSERT INTO CN2.PRODUCT VALUES (CN2.PRO_ID_SEQUENCE.nextval, 'Pen', 'USA', 3000, 5000, TO_DATE('01/01/2022', 'dd/mm/yyyy'), null, 'Requisite', 2, 27);</p> <p>INSERT INTO CN2.PRODUCT VALUES (CN2.PRO_ID_SEQUENCE.nextval, 'Soap', 'France', 10000, 15000,</p>

<p>TO_DATE('01/01/2022', 'dd/mm/yyyy'), null, 'Toiletries', 2, 29);</p> <p>INSERT INTO CN1.PRODUCT VALUES (CN1.PRO_ID_SEQUENCE.nextval, 'Cleanser', 'ThaiLand', 50000, 100000, TO_DATE('01/01/2022', 'dd/mm/yyyy'), null, 'Toiletries', 2, 28);</p> <p>INSERT INTO CN1.PRODUCT VALUES (CN1.PRO_ID_SEQUENCE.nextval, 'Coca', 'USA', 5000, 9000, TO_DATE('01/01/2022', 'dd/mm/yyyy'), TO_DATE('01/01/2023', 'dd/mm/yyyy'), 'Drinks', 2, 28);</p> <p>INSERT INTO CN1.PRODUCT VALUES (CN1.PRO_ID_SEQUENCE.nextval, 'Pepsi', 'USA', 5000, 9000, TO_DATE('01/01/2022', 'dd/mm/yyyy'), TO_DATE('01/01/2023', 'dd/mm/yyyy'), 'Drinks', 2, 29);</p> <p>INSERT INTO CN1.PRODUCT VALUES (CN1.PRO_ID_SEQUENCE.nextval, 'Apple', 'USA', 5000, 10000,</p>	<p>TO_DATE('01/01/2022', 'dd/mm/yyyy'), null, 'Toiletries', 2, 29);</p> <p>INSERT INTO CN2.PRODUCT VALUES (CN2.PRO_ID_SEQUENCE.nextval, 'Cleanser', 'ThaiLand', 50000, 100000, TO_DATE('01/01/2022', 'dd/mm/yyyy'), null, 'Toiletries', 2, 28);</p> <p>INSERT INTO CN2.PRODUCT VALUES (CN2.PRO_ID_SEQUENCE.nextval, 'Coca', 'USA', 5000, 9000, TO_DATE('01/01/2022', 'dd/mm/yyyy'), TO_DATE('01/01/2023', 'dd/mm/yyyy'), 'Drinks', 2, 28);</p> <p>INSERT INTO CN2.PRODUCT VALUES (CN2.PRO_ID_SEQUENCE.nextval, 'Pepsi', 'USA', 5000, 9000, TO_DATE('01/01/2022', 'dd/mm/yyyy'), TO_DATE('01/01/2023', 'dd/mm/yyyy'), 'Drinks', 2, 29);</p> <p>INSERT INTO CN2.PRODUCT VALUES (CN2.PRO_ID_SEQUENCE.nextval, 'Apple', 'USA', 5000, 10000,</p>
---	---

<p>TO_DATE('01/01/2022', 'dd/mm/yyyy'), TO_DATE('01/01/2023', 'dd/mm/yyyy'), 'Organic food', 2, 26); INSERT INTO CN1.PRODUCT VALUES (CN1.PRO_ID_SEQUENCE.nextval, 'Banana', 'VietNam', 5000, 15000, TO_DATE('01/01/2022', 'dd/mm/yyyy'), TO_DATE('01/01/2023', 'dd/mm/yyyy'), 'Organic food', 2, 22); INSERT INTO CN1.PRODUCT VALUES (CN1.PRO_ID_SEQUENCE.nextval, 'Tissue', 'China', 2000, 5000, TO_DATE('01/01/2022', 'dd/mm/yyyy'), null, 'Toiletries', 2, 30); INSERT INTO CN1.PRODUCT VALUES (CN1.PRO_ID_SEQUENCE.nextval, 'Milk', 'Netherlands', 5000, 8000, TO_DATE('01/01/2022', 'dd/mm/yyyy'), TO_DATE('01/01/2023', 'dd/mm/yyyy'), 'Drinks', 2, 28); INSERT INTO CN1.PRODUCT VALUES (CN1.PRO_ID_SEQUENCE.nextval,'</p>	<p>TO_DATE('01/01/2022', 'dd/mm/yyyy'), TO_DATE('01/01/2023', 'dd/mm/yyyy'), 'Organic food', 2, 26); INSERT INTO CN2.PRODUCT VALUES (CN2.PRO_ID_SEQUENCE.nextval, 'Banana', 'VietNam', 5000, 15000, TO_DATE('01/01/2022', 'dd/mm/yyyy'), TO_DATE('01/01/2023', 'dd/mm/yyyy'), 'Organic food', 2, 22); INSERT INTO CN2.PRODUCT VALUES (CN2.PRO_ID_SEQUENCE.nextval, 'Tissue', 'China', 2000, 5000, TO_DATE('01/01/2022', 'dd/mm/yyyy'), null, 'Toiletries', 2, 30); INSERT INTO CN2.PRODUCT VALUES (CN2.PRO_ID_SEQUENCE.nextval, 'Milk', 'Netherlands', 5000, 8000, TO_DATE('01/01/2022', 'dd/mm/yyyy'), TO_DATE('01/01/2023', 'dd/mm/yyyy'), 'Drinks', 2, 28); INSERT INTO CN2.PRODUCT VALUES (CN2.PRO_ID_SEQUENCE.nextval,'</p>
--	--

<p>NoteBook', 'China', 5000, 8000, TO_DATE('01/01/2022', 'dd/mm/yyyy'), null, 'Requisite', 2, 25);</p> <p>INSERT INTO CN1.PRODUCT VALUES (CN1.PRO_ID_SEQUENCE.nextval, 'Pencil', 'USA', 3000, 5000, TO_DATE('01/01/2022', 'dd/mm/yyyy'), null, 'Requisite', 2, 27);</p> <p>INSERT INTO CN1.PRODUCT VALUES (CN1.PRO_ID_SEQUENCE.nextval, 'Vim', 'France', 10000, 15000, TO_DATE('01/01/2022', 'dd/mm/yyyy'), null, 'Toiletries', 2, 29);</p> <p>INSERT INTO CN1.PRODUCT VALUES (CN1.PRO_ID_SEQUENCE.nextval, 'Scouring pad', 'ThaiLand', 50000, 100000, TO_DATE('01/01/2022', 'dd/mm/yyyy'), null, 'Toiletries', 2, 28);</p> <p>INSERT INTO CN1.PRODUCT VALUES (CN1.PRO_ID_SEQUENCE.nextval, 'Strong bow', 'USA', 5000, 9000, TO_DATE('01/01/2022',</p>	<p>NoteBook', 'China', 5000, 8000, TO_DATE('01/01/2022', 'dd/mm/yyyy'), null, 'Requisite', 2, 25);</p> <p>INSERT INTO CN2.PRODUCT VALUES (CN2.PRO_ID_SEQUENCE.nextval, 'Pencil', 'USA', 3000, 5000, TO_DATE('01/01/2022', 'dd/mm/yyyy'), null, 'Requisite', 2, 27);</p> <p>INSERT INTO CN2.PRODUCT VALUES (CN2.PRO_ID_SEQUENCE.nextval, 'Vim', 'France', 10000, 15000, TO_DATE('01/01/2022', 'dd/mm/yyyy'), null, 'Toiletries', 2, 29);</p> <p>INSERT INTO CN2.PRODUCT VALUES (CN2.PRO_ID_SEQUENCE.nextval, 'Scouring pad', 'ThaiLand', 50000, 100000, TO_DATE('01/01/2022', 'dd/mm/yyyy'), null, 'Toiletries', 2, 28);</p> <p>INSERT INTO CN2.PRODUCT VALUES (CN2.PRO_ID_SEQUENCE.nextval, 'Strong bow', 'USA', 5000, 9000, TO_DATE('01/01/2022',</p>
--	--

<pre>'dd/mm/yyyy'), TO_DATE('01/01/2023', 'dd/mm/yyyy'), 'Drinks', 2, 28); INSERT INTO CN1.PRODUCT VALUES (CN1.PRO_ID_SEQUENCE.nextval, 'C2', 'USA', 5000, 9000, TO_DATE('01/01/2022', 'dd/mm/yyyy'), TO_DATE('01/01/2023', 'dd/mm/yyyy'), 'Drinks', 2, 29); INSERT INTO CN1.PRODUCT VALUES (CN1.PRO_ID_SEQUENCE.nextval, 'Orange', 'USA', 5000, 10000, TO_DATE('01/01/2022', 'dd/mm/yyyy'), TO_DATE('01/01/2023', 'dd/mm/yyyy'), 'Organic food', 2, 26); INSERT INTO CN1.PRODUCT VALUES (CN1.PRO_ID_SEQUENCE.nextval, 'Mango', 'VietNam', 5000, 15000, TO_DATE('01/01/2022', 'dd/mm/yyyy'), TO_DATE('01/01/2023', 'dd/mm/yyyy'), 'Organic food', 2, 22); INSERT INTO CN1.PRODUCT VALUES (CN1.PRO_ID_SEQUENCE.nextval,</pre>	<pre>'dd/mm/yyyy'), TO_DATE('01/01/2023', 'dd/mm/yyyy'), 'Drinks', 2, 28); INSERT INTO CN2.PRODUCT VALUES (CN2.PRO_ID_SEQUENCE.nextval, 'C2', 'USA', 5000, 9000, TO_DATE('01/01/2022', 'dd/mm/yyyy'), TO_DATE('01/01/2023', 'dd/mm/yyyy'), 'Drinks', 2, 29); INSERT INTO CN2.PRODUCT VALUES (CN2.PRO_ID_SEQUENCE.nextval, 'Orange', 'USA', 5000, 10000, TO_DATE('01/01/2022', 'dd/mm/yyyy'), TO_DATE('01/01/2023', 'dd/mm/yyyy'), 'Organic food', 2, 26); INSERT INTO CN2.PRODUCT VALUES (CN2.PRO_ID_SEQUENCE.nextval, 'Mango', 'VietNam', 5000, 15000, TO_DATE('01/01/2022', 'dd/mm/yyyy'), TO_DATE('01/01/2023', 'dd/mm/yyyy'), 'Organic food', 2, 22); INSERT INTO CN2.PRODUCT VALUES (CN2.PRO_ID_SEQUENCE.nextval,</pre>
--	--

<p>'Mop', 'China', 2000, 5000, TO_DATE('01/01/2022', 'dd/mm/yyyy'), null, 'Toiletries', 2, 30); INSERT INTO CN1.PRODUCT VALUES (CN1.PRO_ID_SEQUENCE.nextval, 'Milk tea', 'Netherlands', 5000, 8000, TO_DATE('01/01/2022', 'dd/mm/yyyy'), TO_DATE('01/01/2023', 'dd/mm/yyyy'), 'Drinks', 2, 28);</p>	<p>'Mop', 'China', 2000, 5000, TO_DATE('01/01/2022', 'dd/mm/yyyy'), null, 'Toiletries', 2, 30); INSERT INTO CN2.PRODUCT VALUES (CN2.PRO_ID_SEQUENCE.nextval, 'Milk tea', 'Netherlands', 5000, 8000, TO_DATE('01/01/2022', 'dd/mm/yyyy'), TO_DATE('01/01/2023', 'dd/mm/yyyy'), 'Drinks', 2, 28);</p>
<p>INSERT INTO CN1.WAREHOUSE_MANAGEMENT VALUES('CN1', 400001, TO_DATE('01/02/2022','dd/mm/yyyy'), 30); INSERT INTO CN1.WAREHOUSE_MANAGEMENT VALUES('CN1', 400002, TO_DATE('01/02/2022','dd/mm/yyyy'), 30); INSERT INTO CN1.WAREHOUSE_MANAGEMENT VALUES('CN1', 400003, TO_DATE('01/02/2022','dd/mm/yyyy'), 30);</p>	<p>INSERT INTO CN2.WAREHOUSE_MANAGEMENT VALUES('CN2', 400001, TO_DATE('01/02/2022','dd/mm/yyyy'), 30); INSERT INTO CN2.WAREHOUSE_MANAGEMENT VALUES('CN2', 400002, TO_DATE('01/02/2022','dd/mm/yyyy'), 30); INSERT INTO CN2.WAREHOUSE_MANAGEMENT VALUES('CN2', 400003, TO_DATE('01/02/2022','dd/mm/yyyy'), 30);</p>

<p>INSERT INTO CN1.WAREHOUSE_MANAGEMENT VALUES('CN1', 400004, TO_DATE('01/02/2022','dd/mm/yyyy'), 30);</p> <p>INSERT INTO CN1.WAREHOUSE_MANAGEMENT VALUES('CN1', 400005, TO_DATE('01/02/2022','dd/mm/yyyy'), 30);</p> <p>INSERT INTO CN1.WAREHOUSE_MANAGEMENT VALUES('CN1', 400006, TO_DATE('01/02/2022','dd/mm/yyyy'), 30);</p> <p>INSERT INTO CN1.WAREHOUSE_MANAGEMENT VALUES('CN1', 400007, TO_DATE('01/02/2022','dd/mm/yyyy'), 30);</p> <p>INSERT INTO CN1.WAREHOUSE_MANAGEMENT VALUES('CN1', 400008, TO_DATE('01/02/2022','dd/mm/yyyy'), 30);</p>	<p>INSERT INTO CN2.WAREHOUSE_MANAGEMENT VALUES('CN2', 400004, TO_DATE('01/02/2022','dd/mm/yyyy'), 30);</p> <p>INSERT INTO CN2.WAREHOUSE_MANAGEMENT VALUES('CN2', 400005, TO_DATE('01/02/2022','dd/mm/yyyy'), 30);</p> <p>INSERT INTO CN2.WAREHOUSE_MANAGEMENT VALUES('CN2', 400011, TO_DATE('01/02/2022','dd/mm/yyyy'), 30);</p> <p>INSERT INTO CN2.WAREHOUSE_MANAGEMENT VALUES('CN2', 400012, TO_DATE('01/02/2022','dd/mm/yyyy'), 30);</p> <p>INSERT INTO CN2.WAREHOUSE_MANAGEMENT VALUES('CN2', 400013, TO_DATE('01/02/2022','dd/mm/yyyy'), 30);</p>
--	--

INSERT INTO CN1.WAREHOUSE_MANAGEMENT VALUES('CN1', 400009, TO_DATE('01/02/2022','dd/mm/yyyy'), 30);	INSERT INTO CN2.WAREHOUSE_MANAGEMENT VALUES('CN2', 400014, TO_DATE('01/02/2022','dd/mm/yyyy'), 30);
INSERT INTO CN1.WAREHOUSE_MANAGEMENT VALUES('CN1', 400010, TO_DATE('01/02/2022','dd/mm/yyyy'), 30);	INSERT INTO CN2.WAREHOUSE_MANAGEMENT VALUES('CN2', 400015, TO_DATE('01/02/2022','dd/mm/yyyy'), 30);
INSERT INTO CN1.WAREHOUSE_MANAGEMENT VALUES('CN1', 400011, TO_DATE('01/02/2022','dd/mm/yyyy'), 30);	INSERT INTO CN2.WAREHOUSE_MANAGEMENT VALUES('CN2', 400016, TO_DATE('01/02/2022','dd/mm/yyyy'), 30);
INSERT INTO CN1.WAREHOUSE_MANAGEMENT VALUES('CN1', 400012, TO_DATE('01/02/2022','dd/mm/yyyy'), 30);	INSERT INTO CN2.WAREHOUSE_MANAGEMENT VALUES('CN2', 400017, TO_DATE('01/02/2022','dd/mm/yyyy'), 30);
INSERT INTO CN1.WAREHOUSE_MANAGEMENT VALUES('CN1', 400013, TO_DATE('01/02/2022','dd/mm/yyyy'), 30);	INSERT INTO CN2.WAREHOUSE_MANAGEMENT VALUES('CN2', 400018, TO_DATE('01/02/2022','dd/mm/yyyy'), 30);

INSERT INTO CN1.WAREHOUSE_MANAGEMENT VALUES('CN1', 400014, TO_DATE('01/02/2022','dd/mm/yyyy'), 30); INSERT INTO CN1.WAREHOUSE_MANAGEMENT VALUES('CN1', 400015, TO_DATE('01/02/2022','dd/mm/yyyy'), 30);	INSERT INTO CN2.WAREHOUSE_MANAGEMENT VALUES('CN2', 400019, TO_DATE('01/02/2022','dd/mm/yyyy'), 30); INSERT INTO CN2.WAREHOUSE_MANAGEMENT VALUES('CN2', 400020, TO_DATE('01/02/2022','dd/mm/yyyy'), 30);
INSERT INTO CN1.WAREHOUSE_SALES VALUES('CN1', 400001, 'Còn hàng'); INSERT INTO CN1.WAREHOUSE_SALES VALUES('CN1', 400002, 'Còn hàng'); INSERT INTO CN1.WAREHOUSE_SALES VALUES('CN1', 400003, 'Còn hàng'); INSERT INTO CN1.WAREHOUSE_SALES VALUES('CN1', 400004, 'Còn hàng'); INSERT INTO CN1.WAREHOUSE_SALES VALUES('CN1', 400005, 'Còn hàng'); INSERT INTO CN1.WAREHOUSE_SALES VALUES('CN1', 400006, 'Còn hàng');	INSERT INTO CN2.WAREHOUSE_SALES VALUES('CN2', 400001, 'Còn hàng'); INSERT INTO CN2.WAREHOUSE_SALES VALUES('CN2', 400002, 'Còn hàng'); INSERT INTO CN2.WAREHOUSE_SALES VALUES('CN2', 400003, 'Còn hàng'); INSERT INTO CN2.WAREHOUSE_SALES VALUES('CN2', 400004, 'Còn hàng'); INSERT INTO CN2.WAREHOUSE_SALES VALUES('CN2', 400005, 'Còn hàng'); INSERT INTO CN2.WAREHOUSE_SALES VALUES('CN2', 400011, 'Còn hàng');

INSERT INTO CN1.WAREHOUSE_SALES VALUES('CN1', 400007, 'Còn hàng'); INSERT INTO CN1.WAREHOUSE_SALES VALUES('CN1', 400008, 'Còn hàng'); INSERT INTO CN1.WAREHOUSE_SALES VALUES('CN1', 400009, 'Còn hàng'); INSERT INTO CN1.WAREHOUSE_SALES VALUES('CN1', 400010, 'Còn hàng'); INSERT INTO CN1.WAREHOUSE_SALES VALUES('CN1', 400011, 'Còn hàng'); INSERT INTO CN1.WAREHOUSE_SALES VALUES('CN1', 400012, 'Còn hàng'); INSERT INTO CN1.WAREHOUSE_SALES VALUES('CN1', 400013, 'Còn hàng'); INSERT INTO CN1.WAREHOUSE_SALES VALUES('CN1', 400014, 'Còn hàng'); INSERT INTO CN1.WAREHOUSE_SALES VALUES('CN1', 400015, 'Còn hàng');	INSERT INTO CN2.WAREHOUSE_SALES VALUES('CN2', 400012, 'Còn hàng'); INSERT INTO CN2.WAREHOUSE_SALES VALUES('CN2', 400013, 'Còn hàng'); INSERT INTO CN2.WAREHOUSE_SALES VALUES('CN2', 400014, 'Còn hàng'); INSERT INTO CN2.WAREHOUSE_SALES VALUES('CN2', 400015, 'Còn hàng'); INSERT INTO CN2.WAREHOUSE_SALES VALUES('CN2', 400016, 'Còn hàng'); INSERT INTO CN2.WAREHOUSE_SALES VALUES('CN2', 400017, 'Còn hàng'); INSERT INTO CN2.WAREHOUSE_SALES VALUES('CN2', 400018, 'Còn hàng'); INSERT INTO CN2.WAREHOUSE_SALES VALUES('CN2', 400019, 'Còn hàng'); INSERT INTO CN2.WAREHOUSE_SALES VALUES('CN2', 400020, 'Còn hàng');
INSERT INTO CN1.BILL VALUES(BILL_ID_SEQUENCE.nextval, 200002, 300001,	INSERT INTO CN2.BILL VALUES(BILL_ID_SEQUENCE.nextval, 600002, 300011,

<p>TO_DATE('01/01/2022', 'dd/mm/yyyy'), 53295); INSERT INTO CN1.BILL VALUES(BILL_ID_SEQUENCE.ne xtval, 200003, 300011, TO_DATE('10/02/2022', 'dd/mm/yyyy'), 75582); INSERT INTO CN1.BILL VALUES(BILL_ID_SEQUENCE.ne xtval, 200002, 300003, TO_DATE('20/02/2022', 'dd/mm/yyyy'), 77520); INSERT INTO CN1.BILL VALUES(BILL_ID_SEQUENCE.ne xtval, 200003, 300003, TO_DATE('01/03/2022', 'dd/mm/yyyy'), 26163); INSERT INTO CN1.BILL VALUES(BILL_ID_SEQUENCE.ne xtval, 200002, 300015, TO_DATE('09/03/2022', 'dd/mm/yyyy'), 72675); INSERT INTO CN1.BILL VALUES(BILL_ID_SEQUENCE.ne xtval, 200003, 300015, TO_DATE('18/03/2022', 'dd/mm/yyyy'), 96900); INSERT INTO CN1.BILL VALUES(BILL_ID_SEQUENCE.ne xtval, 200002, 300006,</p>	<p>TO_DATE('01/01/2022', 'dd/mm/yyyy'), 53295); INSERT INTO CN2.BILL VALUES(BILL_ID_SEQUENCE.ne xtval, 600003, 300011, TO_DATE('10/02/2022', 'dd/mm/yyyy'), 75582); INSERT INTO CN2.BILL VALUES(BILL_ID_SEQUENCE.ne xtval, 600002, 300006, TO_DATE('20/02/2022', 'dd/mm/yyyy'), 77520); INSERT INTO CN2.BILL VALUES(BILL_ID_SEQUENCE.ne xtval, 600003, 300013, TO_DATE('01/03/2022', 'dd/mm/yyyy'), 26163); INSERT INTO CN2.BILL VALUES(BILL_ID_SEQUENCE.ne xtval, 600002, 300016, TO_DATE('09/03/2022', 'dd/mm/yyyy'), 72675); INSERT INTO CN2.BILL VALUES(BILL_ID_SEQUENCE.ne xtval, 600003, 300016, TO_DATE('18/03/2022', 'dd/mm/yyyy'), 96900); INSERT INTO CN2.BILL VALUES(BILL_ID_SEQUENCE.ne xtval, 600002, 300009,</p>
--	--

<p>TO_DATE('25/03/2022', 'dd/mm/yyyy'), 87210); INSERT INTO CN1.BILL VALUES(BILL_ID_SEQUENCE.ne xtval, 200003, 300007, TO_DATE('05/04/2022', 'dd/mm/yyyy'), 11628); INSERT INTO CN1.BILL VALUES(BILL_ID_SEQUENCE.ne xtval, 200002, 300009, TO_DATE('17/04/2022', 'dd/mm/yyyy'), 145350); INSERT INTO CN1.BILL VALUES(BILL_ID_SEQUENCE.ne xtval, 200003, 300019, TO_DATE('23/04/2022', 'dd/mm/yyyy'), 77520);</p>	<p>TO_DATE('25/03/2022', 'dd/mm/yyyy'), 87210); INSERT INTO CN2.BILL VALUES(BILL_ID_SEQUENCE.ne xtval, 600003, 300017, TO_DATE('05/04/2022', 'dd/mm/yyyy'), 11628); INSERT INTO CN2.BILL VALUES(BILL_ID_SEQUENCE.ne xtval, 600002, 300019, TO_DATE('17/04/2022', 'dd/mm/yyyy'), 145350); INSERT INTO CN2.BILL VALUES(BILL_ID_SEQUEN CE.nextval, 600003, 300003, TO_DATE('23/04/2022', 'dd/mm/yyyy'), 77520);</p>
<p>INSERT INTO CN1.BILL_DETAILS(BILL_ID, PRO_ID, AMOUNT) VALUES (500001, 400001, 5); INSERT INTO CN1.BILL_DETAILS(BILL_ID, PRO_ID, AMOUNT) VALUES (500001, 400012, 3); INSERT INTO CN1.BILL_DETAILS(BILL_ID, PRO_ID, AMOUNT) VALUES (500002, 400014, 4);</p>	<p>INSERT INTO CN2.BILL_DETAILS(BILL_ID, PRO_ID, AMOUNT) VALUES (700001, 400001, 5); INSERT INTO CN2.BILL_DETAILS(BILL_ID, PRO_ID, AMOUNT) VALUES (700001, 400012, 3); INSERT INTO CN2.BILL_DETAILS(BILL_ID, PRO_ID, AMOUNT) VALUES (700002, 400012, 4);</p>

<p>INSERT INTO CN1.BILL_DETAILS(BILL_ID, PRO_ID, AMOUNT) VALUES (500002, 400005, 2);</p> <p>INSERT INTO CN1.BILL_DETAILS(BILL_ID, PRO_ID, AMOUNT) VALUES (500002, 400003, 1);</p> <p>INSERT INTO CN1.BILL_DETAILS(BILL_ID, PRO_ID, AMOUNT) VALUES (500003, 400015, 10);</p> <p>INSERT INTO CN1.BILL_DETAILS(BILL_ID, PRO_ID, AMOUNT) VALUES (500004, 400005, 2);</p> <p>INSERT INTO CN1.BILL_DETAILS(BILL_ID, PRO_ID, AMOUNT) VALUES (500004, 400007, 1);</p> <p>INSERT INTO CN1.BILL_DETAILS(BILL_ID, PRO_ID, AMOUNT) VALUES (500005, 400003, 3);</p> <p>INSERT INTO CN1.BILL_DETAILS(BILL_ID, PRO_ID, AMOUNT) VALUES (500006, 400014, 1);</p>	<p>INSERT INTO CN2.BILL_DETAILS(BILL_ID, PRO_ID, AMOUNT) VALUES (700002, 400015, 2);</p> <p>INSERT INTO CN2.BILL_DETAILS(BILL_ID, PRO_ID, AMOUNT) VALUES (700002, 400013, 1);</p> <p>INSERT INTO CN2.BILL_DETAILS(BILL_ID, PRO_ID, AMOUNT) VALUES (700003, 400017, 10);</p> <p>INSERT INTO CN2.BILL_DETAILS(BILL_ID, PRO_ID, AMOUNT) VALUES (700004, 400015, 2);</p> <p>INSERT INTO CN2.BILL_DETAILS(BILL_ID, PRO_ID, AMOUNT) VALUES (700004, 400007, 1);</p> <p>INSERT INTO CN2.BILL_DETAILS(BILL_ID, PRO_ID, AMOUNT) VALUES (700005, 400013, 3);</p> <p>INSERT INTO CN2.BILL_DETAILS(BILL_ID, PRO_ID, AMOUNT) VALUES (700006, 400005, 1);</p>
---	---

INSERT INTO CN1.BILL_DETAILS(BILL_ID, PRO_ID, AMOUNT) VALUES (500007, 400007, 6); INSERT INTO CN1.BILL_DETAILS(BILL_ID, PRO_ID, AMOUNT) VALUES (500008, 400013, 3); INSERT INTO CN1.BILL_DETAILS(BILL_ID, PRO_ID, AMOUNT) VALUES (500009, 400009, 1); INSERT INTO CN1.BILL_DETAILS(BILL_ID, PRO_ID, AMOUNT) VALUES (500010, 400010, 1);	INSERT INTO CN2.BILL_DETAILS(BILL_ID, PRO_ID, AMOUNT) VALUES (700007, 400017, 6); INSERT INTO CN2.BILL_DETAILS(BILL_ID, PRO_ID, AMOUNT) VALUES (700008, 400004, 3); INSERT INTO CN2.BILL_DETAILS(BILL_ID, PRO_ID, AMOUNT) VALUES (700009, 400019, 1); INSERT INTO CN2.BILL_DETAILS(BILL_ID, PRO_ID, AMOUNT) VALUES (700010, 400020, 1);
--	--

1.4. Kiến trúc phân quyền

1.4.1. Mô tả kiến trúc phân quyền

* Chi nhánh 1: Có 3 quyền

- GiamDoc:
 - Xem, thêm, sửa, xóa được thông tin của các quan hệ EMPLOYEE của cả hai chi nhánh
 - Xem được thông tin của các quan hệ, BILL, BILL_DETAILS, WAREHOUSE_MANAGEMENT, WAREHOUSE_SALES,
 - Xem được thông tin CUSTOMER, PRODUC
- QuanLyKho:
 - Xem, thêm, sửa, xóa được thông tin của các quan hệ WAREHOUSE_MANAGEMENT, WAREHOUSE_SALES của chi nhánh

- Xem được thông tin WAREHOUSE_MANAGEMENT, WAREHOUSE_SALES ở chi nhánh 2 (đã xong)
- xem, thêm, sửa xóa được thông tin PRODUCT
- NhanVien:
 - Xem, thêm được thông tin của các quan hệ BILL, BILL_DETAILS ở chi nhánh 1
 - Xem, thêm, sửa, xóa được thông tin CUSTOME

*** Chi nhánh 2: Có 2 quyền:**

- QuanLyKho:
 - Xem, thêm, sửa, xóa được thông tin PRODUCT
 - Xem, thêm, sửa, xóa được thông tin của các quan hệ WAREHOUSE_MANAGEMENT, WAREHOUSE_SALES của chi nhánh 2
 - Xem được thông tin WAREHOUSE_MANAGEMENT, WAREHOUSE_SALES ở chi nhánh 1
- NhanVien:
 - Xem, thêm được thông tin của các quan hệ BILL, BILL_DETAILS ở chi nhánh 2 (đã xong)
 - Xem, thêm, sửa, xóa được thông tin CUSTOMER

1.4.2. Minh họa

CHI NHÁNH 1

```
alter session set "_ORACLE_SCRIPT" = true;
```

```
--TÀI KHOẢN CN1;
```

```
create user cn1 identified by cn1;
```

```
grant connect, dba to cn1;
```

```
--TÀI KHOẢN GiamDoc
```

```
create user GiamDoc identified by giamdoc;
```

grant connect to GiamDoc;

--Xem được thông tin của các quan hệ, BILL, BILL_DETAILS,
WAREHOUSE_MANAGEMENT, WAREHOUSE_SALES, CUSTOMER,
PRODUCT

grant select on CN1.BILL to GiamDoc;

grant select on CN1.BILL_DETAILS to GiamDoc;

grant select on CN1.WAREHOUSE_MANAGEMENT to GiamDoc;

grant select on CN1.WAREHOUSE_SALES to GiamDoc;

grant select on CN1.CUSTOMER to GiamDoc;

grant select on CN1.PRODUCT to GiamDoc;

--Xem, thêm, sửa, xóa được thông tin của các quan hệ EMPLOYEE của CN1

grant select, insert, update, delete on CN1.EMPLOYEE to GiamDoc;

--TÀI KHOẢN QuanLyKho

create user QuanLyKho identified by quanlykho;

grant connect to QuanLyKho;

--Xem, thêm, sửa, xóa được thông tin của các quan hệ

WAREHOUSE_MANAGEMENT, WAREHOUSE_SALES của chi nhánh 1

grant select, insert, update, delete on CN1.WAREHOUSE_MANAGEMENT to
QuanLyKho;

grant select, insert, update, delete on CN1.WAREHOUSE_SALES to
QuanLyKho;

--xem, thêm sửa xóa được thông tin PRODUCT ở CN1

grant select on CN1.PRODUCT to QuanLyKho;

--TÀI KHOẢN NhanVien

```
create user NhanVien identified by nhanvien;  
grant connect to NhanVien;
```

--Xem, thêm được thông tin của các quan hệ BILL, BILL_DETAILS ở chi nhánh 1

```
grant select, insert on CN1.BILL to NhanVien;  
grant select, insert on CN1.BILL_DETAILS to NhanVien;
```

--Xem, thêm, sửa xóa được thông tin CUSTOMER ở CN1

```
grant select, insert, update, delete on CN1.CUSTOMER to NhanVien;
```

--TÀI KHOẢN CN2

```
create user cn2 identified by cn2;  
grant connect to cn2;
```

```
grant select on CN1.BILL_DETAILS to cn2;  
grant select on CN1.BILL to cn2;  
grant select on CN1.WAREHOUSE_SALES to cn2;  
grant select on CN1.WAREHOUSE_MANAGEMENT to cn2;  
grant select on CN1.PRODUCT to cn2;  
grant select on CN1.CUSTOMER to cn2;  
grant select on CN1.EMPLOYEE to cn2;  
grant select on CN1.BRANCH to cn2
```

--QuanLyKho Xem được thông tin WAREHOUSE_MANAGEMENT, WAREHOUSE_SALES ở chi nhánh 2

```
grant select on CN2.WAREHOUSE_MANAGEMENT to QuanLyKho;  
grant select on CN2.WAREHOUSE_SALES to QuanLyKho;
```

--CREATE DATABASE LINK


```
create public database link cn2_link connect to cn1 identified by cn1 using
'CN2';
create public database link cn2_link_giamdoc connect to GiamDoc identified by
giamdoc using 'CN2';
create public database link cn2_link_nhanvien connect to NhanVien identified
by nhanvien using 'CN2';
create public database link cn2_link_quanlykho connect to QuanLyKho
identified by quanlykho using 'CN2';
```

CHI NHÁNH 2

```
alter session set "_ORACLE_SCRIPT"=true;
```

--Tạo user và phân quyền

--USER CN2

```
create user cn2 identified by cn2;
```

```
grant connect, dba to cn2;
```

--Tài khoản GiamDoc

```
create user GiamDoc identified by giamdoc;
```

```
grant connect to GiamDoc;
```

--Xem được thông tin của các quan hệ, BILL, BILL_DETAILS,
WAREHOUSE_MANAGEMENT, WAREHOUSE_SALES, CUSTOMER,
PRODUCT

```
grant select on CN2.BILL to GiamDoc;
```

```
grant select on CN2.BILL_DETAILS to GiamDoc;
```

```
grant select on CN2.WAREHOUSE_MANAGEMENT to GiamDoc;
```

```
grant select on CN2.WAREHOUSE_SALES, to GiamDoc;
```

```
grant select on CN2.CUSTOMER to GiamDoc;
```

grant select on CN2.PRODUCT to GiamDoc;

--Xem, thêm, sửa, xóa được thông tin của các quan hệ EMPLOYEE của CN2

grant select, insert, update, delete on CN2.EMPLOYEE to GiamDoc;

--Tài khoản QuanLyKho

create user QuanLyKho identified by quanlykho;

grant connect to QuanLyKho;

--Xem, thêm, sửa, xóa được thông tin của các quan hệ

WAREHOUSE_MANAGEMENT, WAREHOUSE_SALES của chi nhánh 2

grant select, insert, update, delete on CN2.WAREHOUSE_MANAGEMENT to QuanLyKho;

grant select, insert, update, delete on CN2.WAREHOUSE_SALES to QuanLyKho;

--xem, thêm sửa xóa được thông tin PRODUCT ở CN2

grant select on CN2.PRODUCT to QuanLyKho;

--Tài khoản NhanVien

create user NhanVien identified by nhanvien;

grant connect to NhanVien;

--Xem, thêm được thông tin của các quan hệ BILL, BILL_DETAILS ở chi nhánh 2

grant select, insert on CN2.BILL to NhanVien;

grant select, insert on CN2.BILL_DETAILS to NhanVien;

--Xem, thêm, sửa xóa được thông tin CUSTOMER ở CN2

grant select, insert, update, delete on CN2.CUSTOMER to NhanVien;

--TÀI KHOẢN CN1

create user cn1 identified by cn1;

grant connect to cn1;

grant select on CN2.BILL_DETAILS to cn1;

grant select on CN2.BILL to cn1;

grant select on CN2.WAREHOUSE_SALES to cn1;

grant select on CN2.WAREHOUSE_MANAGEMENT to cn1;

grant select on CN2.PRODUCT to cn1;

grant select on CN2.CUSTOMER to cn1;

grant select on CN2.EMPLOYEE to cn1;

grant select on CN2.BRANCH to cn1

--QuanLyKho Xem được thông tin WAREHOUSE_MANAGEMENT,
WAREHOUSE_SALES ở chi nhánh 1

grant select on CN1.WAREHOUSE_MANAGEMENT to QuanLyKho;

grant select on CN1.WAREHOUSE_SALES to QuanLyKho;

--Tạo public database link

create public database link cn1_link connect to cn2 identified by cn2 using
'CN1';

create public database link cn1_link_nhanvien connect to NhanVien identified
by nhanvien using 'CN1';

create public database link cn1_link_quanlykho connect to QuanLyKho
identified by quanlykho using 'CN1';

1.5. Thực hiện 10 câu truy vấn

CÂU 1.(CN1 THỰC HIỆN)

"Giám đốc", thực hiện truy vấn cho biết ở CN1 có sản phẩm (PRO_ID, PRODUCT_NAME, COUNTRY, SALE_PRICE, SUM_AMOUNT) nào có nước sản xuất là 'USA' và bán được hơn 5 sản phẩm

```
SELECT P2.PRO_ID, P2.PRODUCT_NAME, P2.COUNTRY,
P2.SALE_PRICE, SUM(AMOUNT) AS SUM_AMOUNT
FROM CN2.PRODUCT@cn2_link_giamdoc P2,
CN2.BILL_DETAILS@cn2_link_giamdoc BD2
WHERE P2.COUNTRY = 'USA' AND P2.PRO_ID = BD2.PRO_ID
HAVING SUM(BD2.AMOUNT)>=5
GROUP BY P2.PRO_ID, P2.PRODUCT_NAME, P2.COUNTRY,
P2.SALE_PRICE;
```

```
SQL> connect giamdoc/giamdoc;
Connected.
SQL> SELECT P2.PRO_ID, P2.PRODUCT_NAME, P2.COUNTRY, P2.SALE_PRICE, SUM(AMOUNT) AS SUM_AMOUNT
2 FROM CN2.PRODUCT@cn2_link_giamdoc P2, CN2.BILL_DETAILS@cn2_link_giamdoc BD2
3 WHERE P2.COUNTRY = 'USA' AND P2.PRO_ID = BD2.PRO_ID
4 HAVING SUM(BD2.AMOUNT)>=5
5 GROUP BY P2.PRO_ID, P2.PRODUCT_NAME, P2.COUNTRY, P2.SALE_PRICE;

PRO_ID
-----
PRODUCT_NAME
-----
COU SALE_PRICE SUM_AMOUNT
---
400012
Pencil
USA 5000 7

400017
Orange
USA 10000 16

PRO_ID
-----
PRODUCT_NAME
-----
COU SALE_PRICE SUM_AMOUNT
---
SQL> |
```

CÂU 2. (CN1 THỰC HIỆN)

"Giám đốc", Thống kê tình trạng bán hàng các sản phẩm (PRO_ID, PRO_ID, PRODUCT_NAME, SUM(AMOUNT)) trên toàn bộ hệ thống minimart

```
SELECT PROID, PRONAME, COALESCE(SUM(SL_BANDUOC), 0) AS  
SL_BANDUOC  
FROM (  
    SELECT P2.PRO_ID AS PROID, P2.PRODUCT_NAME AS PRONAME ,  
    COALESCE(SUM(BD2.AMOUNT), 0) AS SL_BANDUOC  
    FROM CN2.PRODUCT@cn2_link_giamdoc P2  
    LEFT JOIN CN2.BILL_DETAILS BD2  
    ON P2.PRO_ID = BD2.PRO_ID  
    GROUP BY P2.PRO_ID, P2.PRODUCT_NAME  
    UNION ALL  
    SELECT P1.PRO_ID AS PROID, P1.PRODUCT_NAME AS PRONAME ,  
    COALESCE(SUM(BD1.AMOUNT), 0) AS SL_BANDUOC  
    FROM CN1.PRODUCT P1  
    LEFT JOIN CN1.BILL_DETAILS BD1  
    ON P1.PRO_ID = BD1.PRO_ID  
    GROUP BY P1.PRO_ID, P1.PRODUCT_NAME)  
GROUP BY PROID, PRONAME;
```



```

SQL> SELECT PROID, PRONAME, COALESCE(SUM(SL_BANDUOC), 0) AS SL_BANDUOC
2 FROM (
3     SELECT P2.PRO_ID AS PROID, P2.PRODUCT_NAME AS PRONAME , COALESCE(SUM(BD2.AMOUNT), 0) AS SL_BANDUOC
4     FROM CN2.PRODUCT@cn2_link_giamdoc P2
5     LEFT JOIN CN2.BILL_DETAILS@cn2_link_giamdoc BD2
6     ON P2.PRO_ID = BD2.PRO_ID
7     GROUP BY P2.PRO_ID, P2.PRODUCT_NAME
8     UNION ALL
9     SELECT P1.PRO_ID AS PROID, P1.PRODUCT_NAME AS PRONAME , COALESCE(SUM(BD1.AMOUNT), 0) AS SL_BANDUOC
10    FROM CN1.PRODUCT P1
11    LEFT JOIN CN1.BILL_DETAILS BD1
12    ON P1.PRO_ID = BD1.PRO_ID
13    GROUP BY P1.PRO_ID, P1.PRODUCT_NAME)
14 GROUP BY PROID, PRONAME;

```

```

      PROID
-----
PRONAME
-----
SL_BANDUOC
-----
400013
Vim          7
      400007
Apple        8
      PROID
-----
PRONAME
-----
SL_BANDUOC
-----
400003
Soap         4
      400008
Banana
      PROID
-----
PRONAME
-----
SL_BANDUOC
-----

```

CÂU 3 (THỰC HIỆN Ở CHI NHÁNH 2) PHÉP TRỪ

"Quản lý kho" ở chi nhánh 2: Liệt kê ra những sản phẩm (PRO_ID, PRODUCT_NAME) đã nhập kho ở CN2 nhưng không được nhập kho ở CN1

```

SELECT P2.PRO_ID, P2.PRODUCT_NAME
FROM CN2.PRODUCT P2,
CN2.WAREHOUSE_MANAGEMENT@cn2_link_quanlykho WM2
WHERE P2.PRO_ID = WM2.PRO_ID
MINUS
SELECT P1.PRO_ID, P1.PRODUCT_NAME

```

```
FROM CN1.PRODUCT@cn1_link_quanlykho P1,
CN1.WAREHOUSE_MANAGEMENT@cn1_link_quanlykho WM1
WHERE P1.PRO_ID = WM1.PRO_ID;
```

```
SQL> connect QuanLyKho/quanlykho;
Connected.
SQL> SELECT P2.PRO_ID, P2.PRODUCT_NAME
2 FROM CN2.PRODUCT P2, CN2.WAREHOUSE_MANAGEMENT WM2
3 WHERE P2.PRO_ID = WM2.PRO_ID
4 MINUS
5 SELECT P1.PRO_ID, P1.PRODUCT_NAME
6 FROM CN1.PRODUCT@cn1_link_quanlykho P1, CN1.WAREHOUSE_MANAGEMENT@cn1_link_quanlykho WM1
7 WHERE P1.PRO_ID = WM1.PRO_ID;

  PRO_ID
-----
PRODUCT_NAME
-----
  400016
C2
  400017
Orange
  400018
Mango

  PRO_ID
-----
PRODUCT_NAME
-----
  400019
Mop
  400020
Milk tea
SQL>
```

CÂU 4: (THỰC HIỆN ở chi nhánh 2) --PHÉP HỢP

Nhan viên bán hàng ở chi nhánh 2" Liệt kê ra những khách hàng đã mua hàng ở chi nhánh mình làm việc và cũng mua ở những chi nhánh còn lại

```
SELECT DISTINCT C1.CUS_ID, C1.FIRST_NAME || ' ' || C1.LAST_NAME
AS HO_TEN
FROM CN1.CUSTOMERcn1_link_nhanvien C1, CN1.BILLcn1_link_nhanvien
B1
WHERE C1.CUS_ID = B1.CUS_ID
```

UNION

SELECT DISTINCT C2.CUS_ID, C2.FIRST_NAME || ' ' || C2.LAST_NAME
AS HO_TEN

FROM CN2.CUSTOMER C2, CN2.BILL B2

WHERE C2.CUS_ID = B2.CUS_ID;

```
SQL> connect NhanVien/nhanvien;
Connected.
SQL> SELECT DISTINCT C1.CUS_ID, C1.FIRST_NAME || ' ' || C1.LAST_NAME AS HO_TEN
  2  FROM CN1.CUSTOMER@cn1_link_nhanvien C1, CN1.BILL@cn1_link_nhanvien B1
  3  WHERE C1.CUS_ID = B1.CUS_ID
  4  UNION
  5  SELECT DISTINCT C2.CUS_ID, C2.FIRST_NAME || ' ' || C2.LAST_NAME AS HO_TEN
  6  FROM CN2.CUSTOMER C2, CN2.BILL B2
  7  WHERE C2.CUS_ID = B2.CUS_ID;

  CUS_ID HO_TEN
-----
300001 Tran Minh Tien
300003 Nguyen Huu Tho
300006 Nguyen Thanh Duy
300007 Vo Minh Trong
300009 Ho Nguyen Thu An
300011 Tran Minh Minh
300013 Nguyen Huu Tham
300015 Le Thi Tuong An
300016 Nguyen Thanh Khiết
300017 Vo Minh Tam
300019 Ho Nguyen Thu Minh

11 rows selected.

SQL>
```

CÂU 5: (THỰC HIỆN Ở CHI NHÁNH 1) --PHÉP GIAO

"Thủ kho ở chi nhánh 1 "Liệt kê ra danh sách những sản phẩm (PRO_ID, PRODUCT_NAME) 'Còn hàng' ở chi nhánh mình làm cũng hết hàng ở chi nhánh còn lại

SELECT P1.PRO_ID, P1.PRODUCT_NAME

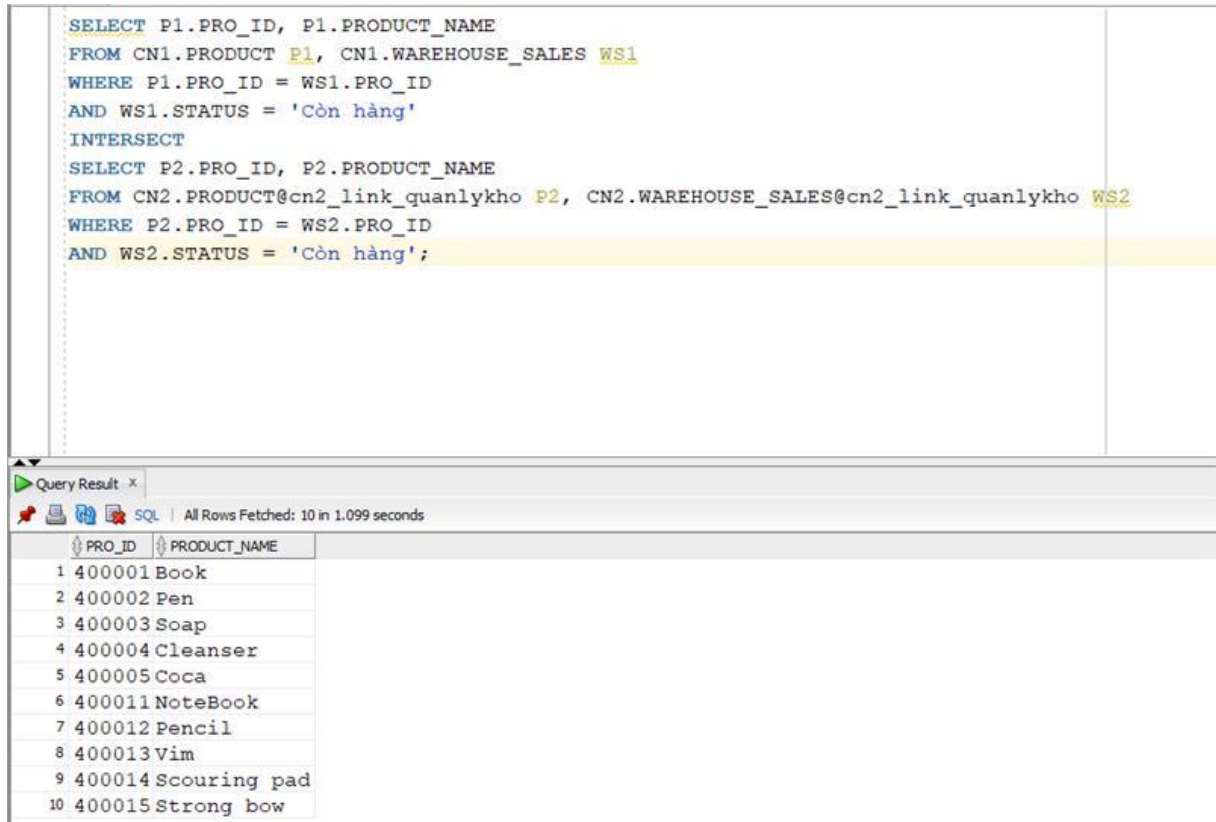
FROM CN1.PRODUCT P1, CN1.WAREHOUSE_SALES WS1

WHERE P1.PRO_ID = WS1.PRO_ID

AND WS1.STATUS = 'Còn hàng'

INTERSECT

```
SELECT P2.PRO_ID, P2.PRODUCT_NAME
FROM CN2.PRODUCT@cn2_link_quanlykho P2,
CN2.WAREHOUSE_SALES@cn2_link_quanlykho WS2
WHERE P2.PRO_ID = WS2.PRO_ID
AND WS2.STATUS = 'Còn hàng';
```



The screenshot shows a SQL query window with the following text:

```
SELECT P1.PRO_ID, P1.PRODUCT_NAME
FROM CN1.PRODUCT P1, CN1.WAREHOUSE_SALES WS1
WHERE P1.PRO_ID = WS1.PRO_ID
AND WS1.STATUS = 'Còn hàng'
INTERSECT
SELECT P2.PRO_ID, P2.PRODUCT_NAME
FROM CN2.PRODUCT@cn2_link_quanlykho P2, CN2.WAREHOUSE_SALES@cn2_link_quanlykho WS2
WHERE P2.PRO_ID = WS2.PRO_ID
AND WS2.STATUS = 'Còn hàng';
```

Below the query, the 'Query Result' window shows the following data:

PRO_ID	PRODUCT_NAME
1 400001	Book
2 400002	Pen
3 400003	Soap
4 400004	Cleanser
5 400005	Coca
6 400011	NoteBook
7 400012	Pencil
8 400013	Vim
9 400014	Scouring pad
10 400015	Strong bow

CÂU 6. (THỰC HIỆN Ở CHI NHÁNH 1) PHÉP CHIA

"Giám đốc" tìm khách hàng đã mua tất cả những sản phẩm có nước sản xuất là 'VietNam'

```
SELECT C1.CUS_ID, C1.FIRST_NAME || ' ' || C1.LAST_NAME AS
HO_TEN
FROM CN1.CUSTOMER C1
WHERE NOT EXISTS (SELECT *
FROM CN1.PRODUCT P1
WHERE P1.COUNTRY = 'VietNam')
```

```

AND NOT EXISTS (SELECT * FROM (
    (SELECT * FROM CN1.BILL B1,
CN1.BILL_DETAILS BD1
    WHERE B1.CUS_ID = C1.CUS_ID
    AND BD1.BILL_ID = B1.BILL_ID
    AND BD1.PRO_ID = P1.PRO_ID)
    UNION
    (SELECT * FROM
CN2.BILL@cn2_link_giamdoc B2, CN2.BILL_DETAILS@cn2_link_giamdoc
BD2
    WHERE B2.CUS_ID = C1.CUS_ID
    AND BD2.BILL_ID = B2.BILL_ID
    AND BD2.PRO_ID = P1.PRO_ID)))));

```

```

SQL> SELECT P1.PRO_ID, P1.PRODUCT_NAME
  2 FROM CN1.PRODUCT P1, CN1.WAREHOUSE_SALES WS1
  3 WHERE P1.PRO_ID = WS1.PRO_ID
  4 AND WS1.STATUS = 'Còn hàng';

no rows selected

SQL> connect giamdoc/giamdoc;
Connected.
SQL> SELECT C1.CUS_ID, C1.FIRST_NAME || ' ' || C1.LAST_NAME AS HO_TEN
  2 FROM CN1.CUSTOMER C1
  3 WHERE NOT EXISTS (SELECT *
  4 FROM CN1.PRODUCT P1
  5 WHERE P1.COUNTRY = 'VietNam'
  6 AND NOT EXISTS (SELECT * FROM (
  7 (SELECT * FROM CN1.BILL B1, CN1.BILL_DETAILS BD1
  8 WHERE B1.CUS_ID = C1.CUS_ID
  9 AND BD1.BILL_ID = B1.BILL_ID
 10 AND BD1.PRO_ID = P1.PRO_ID)
 11 UNION
 12 (SELECT * FROM CN2.BILL@cn2_link_giamdoc B2,
 13 CN2.BILL_DETAILS@cn2_link_giamdoc BD2
 14 WHERE B2.CUS_ID = C1.CUS_ID
 15 AND BD2.BILL_ID = B2.BILL_ID
  AND BD2.PRO_ID = P1.PRO_ID)))));

CUS_ID HO_TEN
-----
300001 Tran Minh Tien

SQL> |

```


CÂU 7: (THỰC HIỆN Ở CHI NHÁNH 1) HÀM GOM NHÓM VÀ TÍNH TOÁN

"Giám đốc" Liệt kê top 5 sản phẩm bán chạy (PRO_ID, PRODUCT_NAME, SUM(AMOUNT)) trong toàn bộ hệ thống MiniMart

```
SELECT MASP, TENSF, SUM(SOLUONGBANDUOC) AS
TONGBANDUOC
FROM (SELECT P1.PRO_ID AS MASP, P1.PRODUCT_NAME AS TENSF,
SUM(BD1.AMOUNT) AS SOLUONGBANDUOC
FROM CN1.PRODUCT P1, CN1.BILL_DETAILS BD1
WHERE P1.PRO_ID = BD1.PRO_ID
GROUP BY P1.PRO_ID, P1.PRODUCT_NAME
UNION ALL
SELECT P2.PRO_ID AS MASP, P2.PRODUCT_NAME AS TENSF,
SUM(BD2.AMOUNT) AS SOLUONGBANDUOC
FROM CN2.PRODUCT@cn2_link_giamdoc P2,
CN2.BILL_DETAILS@cn2_link_giamdoc BD2
WHERE P2.PRO_ID = BD2.PRO_ID
GROUP BY P2.PRO_ID, P2.PRODUCT_NAME)
GROUP BY MASP, TENSF
ORDER BY TONGBANDUOC DESC
FETCH NEXT 5 ROWS ONLY;
```

```

SQL> SELECT  MASP, TENS, SUM(SOLUONGBANDUOC) AS TONGBANDUOC
2 FROM      (SELECT P1.PRO_ID AS MASP, P1.PRODUCT_NAME AS TENS, SUM(BD1.AMOUNT) AS SOLUONGBANDUOC
3            FROM    CN1.PRODUCT P1, CN1.BILL_DETAILS BD1
4            WHERE   P1.PRO_ID = BD1.PRO_ID
5            GROUP BY P1.PRO_ID, P1.PRODUCT_NAME
6            UNION ALL
7            SELECT P2.PRO_ID AS MASP, P2.PRODUCT_NAME AS TENS, SUM(BD2.AMOUNT) AS SOLUONGBANDUOC
8            FROM    CN2.PRODUCT@cn2_link_giamdoc P2, CN2.BILL_DETAILS@cn2_link_giamdoc BD2
9            WHERE   P2.PRO_ID = BD2.PRO_ID
10           GROUP BY P2.PRO_ID, P2.PRODUCT_NAME)
11 GROUP BY MASP, TENS
12 ORDER BY TONGBANDUOC DESC
13 FETCH NEXT 5 ROWS ONLY;

```

```

      MASP
-----
TENS
-----
TONGBANDUOC
-----
      400017
Orange          16

      400015
Strong bow     14

      MASP
-----
TENS
-----
TONGBANDUOC
-----

      400012
Pencil         10

      400007
Apple

      MASP
-----
TENS
-----
TONGBANDUOC
-----
      8

```

CÂU 8. (THỰC HIỆN Ở CHI NHÁNH 1)

"Giám đốc" Liệt kê số lượng các nhân viên làm việc tại từng chi nhánh

```

SELECT MACN, SOLUONGNHANVIEN
FROM (SELECT E1.BRANCH_ID AS MACN, COUNT(E1.EMP_ID) AS
SOLUONGNHANVIEN
      FROM  CN1.EMPLOYEE E1
      WHERE E1.BRANCH_ID = 'CN1'
      GROUP BY E1.BRANCH_ID

```

UNION ALL

SELECT E2.BRANCH_ID AS MACN, COUNT(E2.EMP_ID) AS

SOLUONGNHANVIEN

FROM CN2.EMPLOYEE@cn2_link_giamdoc E2

WHERE E2.BRANCH_ID = 'CN2'

GROUP BY E2.BRANCH_ID);

```
SQL> SELECT  MACN, SOLUONGNHANVIEN
  2 FROM      (SELECT E1.BRANCH_ID AS MACN, COUNT(E1.EMP_ID) AS SOLUONGNHANVIEN
  3            FROM      CN1.EMPLOYEE E1
  4            WHERE     E1.BRANCH_ID = 'CN1'
  5            GROUP BY  E1.BRANCH_ID
  6            UNION ALL
  7            SELECT E2.BRANCH_ID AS MACN, COUNT(E2.EMP_ID) AS SOLUONGNHANVIEN
  8            FROM      CN2.EMPLOYEE@cn2_link_giamdoc E2
  9            WHERE     E2.BRANCH_ID = 'CN2'
 10            GROUP BY  E2.BRANCH_ID);
```

MAC	SOLUONGNHANVIEN
CN1	10
CN2	10

CÂU 9. (THỰC HIỆN Ở CHI NHÁNH 1)

"Giám đốc" Tìm khách hàng đã mua ít nhất 3 lần và được ít nhất 2 nhân viên thanh toán trên toàn bộ hệ thống MiniMart

SELECT MAKH, HO_TEN, SUM(SO_LAN_MUA) AS

TONG_SO_LAN_MUA, SUM(SO_NV_THANHTOAN) AS

TONG_SO_NV_THANHTOAN

FROM

(SELECT C1.CUS_ID AS MAKH, C1.FIRST_NAME || ' ' ||

C1.LAST_NAME AS HO_TEN, COUNT(B1.BILL_ID) AS SO_LAN_MUA,

COUNT(B1.EMP_ID) AS SO_NV_THANHTOAN

FROM CN1.CUSTOMER C1, CN1.BILL B1

WHERE C1.CUS_ID = B1.CUS_ID

GROUP BY C1.CUS_ID, C1.FIRST_NAME || ' ' || C1.LAST_NAME

```

UNION ALL
SELECT C2.CUS_ID, C2.FIRST_NAME || ' ' || C2.LAST_NAME AS
HO_TEN, COUNT(B2.BILL_ID) AS SO_LAN_MUA, COUNT(B2.EMP_ID)
AS SO_NV_THANHTOAN
FROM CN2.CUSTOMER@cn2_link_giamdoc C2,
CN2.BILL@cn2_link_giamdoc B2
WHERE C2.CUS_ID = B2.CUS_ID
GROUP BY C2.CUS_ID, C2.FIRST_NAME || ' ' || C2.LAST_NAME)
GROUP BY MAKH, HO_TEN
HAVING SUM(SO_LAN_MUA) >= 3 AND SUM(SO_NV_THANHTOAN)
>= 2;

```

```

SQL> SELECT MAKH, HO_TEN, SUM(SO_LAN_MUA) AS TONG_SO_LAN_MUA, SUM(SO_NV_THANHTOAN) AS TONG_SO_NV_THAN
HTOAN
2 FROM
3 (SELECT C1.CUS_ID AS MAKH, C1.FIRST_NAME || ' ' || C1.LAST_NAME AS HO_TEN, COUNT(B1.BILL_ID)
AS SO_LAN_MUA, COUNT(B1.EMP_ID) AS SO_NV_THANHTOAN
4 FROM CN1.CUSTOMER C1, CN1.BILL B1
5 WHERE C1.CUS_ID = B1.CUS_ID
6 GROUP BY C1.CUS_ID, C1.FIRST_NAME || ' ' || C1.LAST_NAME
7 UNION ALL
8 SELECT C2.CUS_ID, C2.FIRST_NAME || ' ' || C2.LAST_NAME AS HO_TEN, COUNT(B2.BILL_ID) AS SO_LA
N_MUA, COUNT(B2.EMP_ID) AS SO_NV_THANHTOAN
9 FROM CN2.CUSTOMER@cn2_link_giamdoc C2, CN2.BILL@cn2_link_giamdoc B2
10 WHERE C2.CUS_ID = B2.CUS_ID
11 GROUP BY C2.CUS_ID, C2.FIRST_NAME || ' ' || C2.LAST_NAME)
12 GROUP BY MAKH, HO_TEN
13 HAVING SUM(SO_LAN_MUA) >= 3 AND SUM(SO_NV_THANHTOAN) >= 2;

MAKH HO_TEN TONG_SO_LAN_MUA
-----
TONG_SO_NV_THANHTOAN
-----
300011 Tran Minh Minh 3

```

CÂU 10. (THỰC HIỆN TẠI CHI NHÁNH 1)

"Giám đốc" Tìm sản phẩm được mua nhiều nhất tại từng chi nhánh

```

(SELECT P1.PRO_ID AS MASP, P1.PRODUCT_NAME AS TENSP,
WM1.BRANCH_ID AS MACN, SUM(BD1.AMOUNT) AS SL_BANDUOC
FROM CN1.PRODUCT P1, CN1.BILL_DETAILS BD1,
CN1.WAREHOUSE_MANAGEMENT WM1
WHERE P1.PRO_ID = BD1.PRO_ID AND P1.PRO_ID = WM1.PRO_ID
GROUP BY P1.PRO_ID, P1.PRODUCT_NAME, WM1.BRANCH_ID

```

```

ORDER BY SUM(BD1.AMOUNT) DESC
FETCH FIRST 1 ROW WITH TIES)
UNION
(SELECT P2.PRO_ID AS MASP, P2.PRODUCT_NAME AS TENS,
WM2.BRANCH_ID AS MACN, SUM(BD2.AMOUNT) AS SL_BANDUOC
FROM CN2.PRODUCT@cn2_link_giamdoc P2,
CN2.BILL_DETAILS@cn2_link_giamdoc BD2,
CN2.WAREHOUSE_MANAGEMENT@cn2_link_giamdoc WM2
WHERE P2.PRO_ID = BD2.PRO_ID AND P2.PRO_ID = WM2.PRO_ID
GROUP BY P2.PRO_ID, P2.PRODUCT_NAME, WM2.BRANCH_ID
ORDER BY SUM(BD2.AMOUNT) DESC
FETCH FIRST 1 ROW WITH TIES);

```

```

SQL> SELECT MAKH, HO_TEN, SUM(SO_LAN_MUA) AS TONG_SO_LAN_MUA, SUM(SO_NV_THANHTOAN) AS TONG_SO_NV_THANH
HTOAN
2 FROM
3 (SELECT C1.CUS_ID AS MAKH, C1.FIRST_NAME || ' ' || C1.LAST_NAME AS HO_TEN, COUNT(B1.BILL_ID)
AS SO_LAN_MUA, COUNT(B1.EMP_ID) AS SO_NV_THANHTOAN
4 FROM CN1.CUSTOMER C1, CN1.BILL B1
5 WHERE C1.CUS_ID = B1.CUS_ID
6 GROUP BY C1.CUS_ID, C1.FIRST_NAME || ' ' || C1.LAST_NAME
7 UNION ALL
8 SELECT C2.CUS_ID, C2.FIRST_NAME || ' ' || C2.LAST_NAME AS HO_TEN, COUNT(B2.BILL_ID) AS SO_LA
N_MUA, COUNT(B2.EMP_ID) AS SO_NV_THANHTOAN
9 FROM CN2.CUSTOMER@cn2_link_giamdoc C2, CN2.BILL@cn2_link_giamdoc B2
10 WHERE C2.CUS_ID = B2.CUS_ID
11 GROUP BY C2.CUS_ID, C2.FIRST_NAME || ' ' || C2.LAST_NAME)
12 GROUP BY MAKH, HO_TEN
13 HAVING SUM(SO_LAN_MUA) >= 3 AND SUM(SO_NV_THANHTOAN) >= 2;

      MAKH HO_TEN                                TONG_SO_LAN_MUA
-----
TONG_SO_NV_THANHTOAN
-----
300011 Tran Minh Minh                                3

```

2. VIẾT HÀM, THỦ TỤC VÀ RÀNG BUỘC TOÀN VỆ

2.1. Function

Nhập vào mã khách hàng, tính tổng số tiền mà khách hàng này đã chi tiêu, mua sắm trên toàn hệ thống minimart

* Cài đặt function

```

CREATE OR REPLACE FUNCTION FUNCTION1(CUSID
CUSTOMER.CUS_ID%TYPE)
RETURN NUMBER

```



```
AS
V_TONGTIEN NUMBER;
BEGIN
  SELECT SUM(TOTAL) INTO V_TONGTIEN
  FROM(
    SELECT SUM(B1.TOTAL_MONEY) AS TOTAL
    FROM CN1.BILL B1
    WHERE B1.CUS_ID = 300001
    UNION ALL
    SELECT SUM(B2.TOTAL_MONEY) AS TOTAL
    FROM CN2.BILL@cn2_link B2
    WHERE B2.CUS_ID=300001
  );
  RETURN V_TONGTIEN;
EXCEPTION
  WHEN NO_DATA_FOUND THEN
    RETURN NULL;
END;
```

* Thực thi function

```
SET SERVEROUTPUT ON
DECLARE
  CUSID CUSTOMER.CUS_ID%TYPE := 300001;
BEGIN
  DBMS_OUTPUT.PUT_LINE( 'Tong tien chi mua:
'||FUNCTION1(CUSID));
END;
```

2.1. Procedure

Nhập vào mã khách hàng, cho biết thông tin sản phẩm được khách hàng này mua nhiều nhất ở từng chi nhánh của Minimart

* Cài đặt procedure

```
CREATE OR REPLACE PROCEDURE PROCEDURE1(CUSID
CUSTOMER.CUS_ID%TYPE)
AS
BEGIN
    FOR item IN (
        (SELECT BR1.BRANCH_NAME as BRANCHNAME,
P1.PRO_ID AS PROID, P1.PRODUCT_NAME AS PRONAME,
SUM(AMOUNT) AS SOLD_AMOUNT
        FROM CN1.PRODUCT P1
        JOIN CN1.BILL_DETAILS BD1 ON P1.PRO_ID = BD1.PRO_ID
        JOIN CN1.BILL B1 ON B1.BILL_ID = BD1.BILL_ID
        JOIN CN1.EMPLOYEE E1 ON B1.EMP_ID = E1.EMP_ID
        JOIN CN1.BRANCH BR1 ON E1.BRANCH_ID =
BR1.BRANCH_ID
        WHERE B1.CUS_ID = CUSID
        GROUP BY BR1.BRANCH_NAME, P1.PRO_ID,
P1.PRODUCT_NAME
        ORDER BY SOLD_AMOUNT DESC
        FETCH FIRST 1 ROW WITH TIES)
        UNION
        (SELECT BR2.BRANCH_NAME as BRANCHNAME,
P2.PRO_ID AS PROID, P2.PRODUCT_NAME AS PRONAME,
SUM(AMOUNT) AS SOLD_AMOUNT
        FROM CN2.PRODUCT@cn2_link P2
        JOIN CN2.BILL_DETAILS@cn2_link BD2 ON P2.PRO_ID =
BD2.PRO_ID
```

```

        JOIN CN2.BILL@cn2_link B2 ON B2.BILL_ID = BD2.BILL_ID
        JOIN CN2.EMPLOYEE@cn2_link E2 ON B2.EMP_ID =
E2.EMP_ID
        JOIN CN2.BRANCH@cn2_link BR2 ON E2.BRANCH_ID =
BR2.BRANCH_ID
        WHERE B2.CUS_ID = CUSID
        GROUP BY BR2.BRANCH_NAME, P2.PRO_ID,
P2.PRODUCT_NAME
        ORDER BY SOLD_AMOUNT DESC
        FETCH FIRST 1 ROW WITH TIES)
    )
LOOP
    DBMS_OUTPUT.PUT_LINE(
        'Ten chi nhánh = ' || item.BRANCHNAME
        || ', Ma san pham = ' || item.PROID
        || ', Ten san pham ='||item.PRONAME
        || ',So luong ='||item.SOLD_AMOUNT);
END LOOP;
END;

```

* Thực thi procedure

```

SET SERVEROUTPUT ON
BEGIN
    PROCEDURE1(200001);
END;

```

2.3. Ràng buộc toàn vẹn

Ngày mua hàng (BILL_DATE) của khách hàng thành viên phải lớn hơn ngày khách hàng đó đăng ký thành viên (REG_DATE)

BỐI CẢNH: CUSTOMER, BILL

NỘI DUNG:

$\forall B \in \text{BILL}, \exists C \in \text{CUSTOMER}:$

$C.\text{CUS_ID} = B.\text{CUS_ID} \wedge B.\text{BILL_DATE} \geq \text{REG_DATE}$

BẢNG TẦM ẢNH HƯỞNG

	THÊM	XÓA	SỬA
CUSTOMER	-	-	+(REG_DATE)
BILL	+	-	+(BILL_DATE, CUS_ID)

* Trigger trên bảng BILL

```

CREATE OR REPLACE TRIGGER TRIGGER_INSERT_UPDATE_BILL
AFTER INSERT OR UPDATE OF BILL_DATE ON BILL
FOR EACH ROW
DECLARE
    B_BILL_DATE BILL.BILL_DATE%TYPE;
    C_REG_DATE CUSTOMER.REG_DATE%TYPE;
    C_CUSID CUSTOMER.CUS_ID%TYPE;
BEGIN
    SELECT REG_DATE INTO C_REG_DATE
    FROM CUSTOMER
    WHERE CUS_ID = :NEW.CUS_ID;

    IF(:NEW.BILL_DATE < C_REG_DATE) THEN
        BEGIN
            RAISE_APPLICATION_ERROR(-20100, 'ERROR:
NGAY MUA HANG CUA KHACH HANG KHONG HOP LE');
        END;
    END IF;
END;
```

* Trigger trên bảng CUSTOMER

```
CREATE OR REPLACE TRIGGER TRIGGER_UPDATE_CUSTOMER
AFTER INSERT OR UPDATE OF REG_DATE ON CUSTOMER
FOR EACH ROW
DECLARE
    B_BILL_ID BILL.BILL_ID%TYPE;
    B_BILL_DATE BILL.BILL_DATE%TYPE;
    CURSOR CURSOR_BILL_DATE IS
        SELECT BILL_ID
        FROM BILL
        WHERE CUS_ID = :NEW.CUS_ID;
BEGIN
    OPEN CURSOR_BILL_DATE;
    LOOP
        FETCH CURSOR_BILL_DATE INTO B_BILL_ID;
        EXIT WHEN CURSOR_BILL_DATE%NOTFOUND;

        SELECT BILL_DATE INTO B_BILL_DATE
        FROM BILL
        WHERE BILL_ID = B_BILL_ID;

        IF(B_BILL_DATE < :NEW.REG_DATE) THEN
            RAISE_APPLICATION_ERROR(-20100, 'ERROR: NGÀY
DANG KY KHONG HOP LE');
        END IF;
    END LOOP;
    CLOSE CURSOR_BILL_DATE;
    DBMS_OUTPUT.PUT_LINE('THANH CONG');
END;
```


3. DEMO CÁC MỨC CÔ LẬP (ISOLATION LEVEL) TRONG MÔI TRƯỜNG PHÂN TÁN

3.1. Trường hợp lost update

3.1.1. Mô tả tình huống

Tại t₀, địa chỉ của khách hàng C là “117/2 Nguyễn Trãi, Q5, TpHCM”

Tại t₁, nhân viên A cập nhật địa chỉ cho khách hàng C là “731 Trần Hưng Đạo , Q5, TpHCM”

Tại t₂, nhân viên B cũng cập nhật địa chỉ cho khách hàng là “200/11 Nguyễn Văn Cừ, Q5, TpHCM”

Tại t₃, nhân viên A thực hiện COMMIT

Tại t₄, nhân viên B thực hiện COMMIT. Thông tin cập nhật của nhân viên B sẽ ghi đè lên thông tin cập nhật của nhân viên A. Như vậy, kết quả là dữ liệu cập nhật của nhân viên A sẽ bị mất.

Vậy tại t₅, địa chỉ của khách hàng C là “200/11 Nguyễn Văn Cừ, Q5, TpHCM”

3.1.2. Minh họa

Time	Transaction T1(Employee A) - CN2	T1 Output	Transaction T2(Employee B) - CN1	T2 Output
t ₀	SELECT ADDRESS FROM CN2.CUSTOMER WHERE CUS_ID = 300001;	117/2 Nguyen Trai, Q5, TpHCM	SELECT ADDRESS FROM CN2.CUSTOMER@cn 2_link WHERE CUS_ID = 300001;	117/2 Nguyen Trai, Q5, TpHCM
t ₁	UPDATE CN2.CUSTOMER SET Address ='731 Tran Hung Dao,Q5, TpHCM'	1 row updated.		

	WHERE CUS_ID = 300001;			
t2			UPDATE CN2.CUSTOMER@cn 2_link SET Address = '200/11 Nguyen Van Cu, Q5, TpHCM' WHERE CUS_ID = 300001;	
t3	COMMIT;	Commit complete .		1 row updated.
t4			COMMIT;	Commit complete .
t5	SELECT ADDRESS FROM CN2.CUSTOMER WHERE CUS_ID = 300001;	200/11 Nguyen Van Cu, Q5, TpHCM	SELECT ADDRESS FROM CN2.CUSTOMER@cn 2_link WHERE CUS_ID = 300001;	200/11 Nguyen Van Cu, Q5, TpHCM

3.1.3. Giải pháp

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

SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;

3.2. Trường hợp dirty read

Mô tả tình huống: Khi khách hàng A đang chuyển tiền nhưng chưa commit.

Cùng lúc đó, nhân viên B đang xem số dư tài khoản của khách hàng A. Sau đó, giao dịch này bị rollback do số tài khoản của khách hàng A chuyển đến không hợp lệ.

Kết quả là nhân viên B đã đọc nhầm thông tin từ tài khoản của khách hàng A. Tuy nhiên, trong hệ quản trị cơ sở dữ liệu Oracle, mức cô lập mặc định là read committed vì vậy không bao giờ có trường hợp Dirty Read.

3.3. Trường hợp unrepeatable read

3.3.1. Mô tả tình huống

Tại t₀, nhân viên A đang xem thông tin của sản phẩm C. Tại thời điểm này, giá của sản phẩm C là 9.000.000 VND.

Tại t₁, giám đốc B cập nhật giá sản phẩm thành 10.000.000 VND

Tại t₂, giám đốc B thực hiện thay đổi.

Tại t₃, nhân viên A xem xét lại thông tin của sản phẩm C và nhận thấy rằng giá tiền của sản phẩm C đã tăng thêm 1.000.000 VND. Như vậy, hai lần xem thông tin khách hàng 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.

3.3.2. Minh họa

Time	Transaction T1(Employee A)	T1 Output	Transaction T1(Director B)	T2 Output
t ₀	SET SERVEROUTPUT ON; BEGIN CN2.PRODUCT_IN FO (400001); END; /	MASP = 400021, NAME = ..., SALE PRICE = 9.000.000 PL/SQL procedure successfull y completed.		

t1			BEGIN UPDATE_PRICE (10000, 400001); END; /	PL/SQL procedure successfully completed.
t2			COMMIT;	Commit complete.
t3	BEGIN CN2.PRODUCT_IN FO (400001); END; /	MASP = 400001, NAME = ..., SALE PRICE = 10.000.000 PL/SQL procedure successfull y completed.		

3.3.3. 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;

3.4. Trường hợp phantom read

3.4.1. Mô tả tình huống

Tại t0, nhân viên A xem thông tin của sản phẩm C.

Tại t2, giám đốc B xóa thông tin của sản phẩm C.

Tại t3, nhân viên A thử đọc thông tin của sản phẩm C 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 đó.

3.4.2. Minh họa

Time	Transaction T1(Employee A)	T1 Output	Transaction T1(Director B)	T2 Output
t0	SET SERVEROUTPUT ON; BEGIN CN2.PRODUCT_I NFO (400009); END; /	MASP = 400021, NAME = ..., SALE PRICE = 9.000.000 PL/SQL procedure successfully completed.		
t1			BEGIN CN2.DEL_PRO@cn2_1 ink(400009); END; /	PL/SQ L procedu re success fully complet ed .
t2			COMMIT;	Comm it

				complete.
t3	BEGIN CN2.PRODUCT_INFORMATION (400009); END; /	Error report - ORA- 20008: Missing product information ORA- 06512: at "C##USSE R1.PRODUCT_INFORMATION", line 40 ORA- 06512: at line 2		

3.4.3. Giải pháp

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

SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;

3.5. Trường hợp deadlock

3.5.1. Mô tả tình huống

Tại t0, nhân viên A cập nhật trạng thái của đơn hàng C.

Tại t1, nhân viên B cập nhật trạng thái của đơn hàng D.

Tại t2, nhân viên A cập nhật trạng thái của đơn hàng D.

Tại t3, nhân viên B cập nhật trạng thái của đơn hàng C. 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.

3.5.2. Minh họa

Time	Transaction T1(Employee A)	T1 Output	Transaction T2(Employee B)	T2 Output
t0	UPDATE CN2.WAREHOUSE_SALES SET STATUS = 'Hết hàng' WHERE PRO_ID = 400001;	1 row updated .		
t1			UPDATE CN2.WAREHOUSE_SALES@cn2_link SET STATUS = 'Hết hàng' WHERE PRO_ID = 400002;	1 row updated .
t2	UPDATE CN2.WAREHOUSE_SALES SET STATUS = 'Cho nhập hàng' WHERE PRO_ID = 400002;			
t3			UPDATE	

			CN2.WAREHOUSE_ SALES@cn2_link SET STATUS = 'Cho nhập hàng' WHERE PRO_ID = 400001;	
t4		ERROR at line 3: ORA- 00060: deadlock detected while waiting for resource		

3.5.3. 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**.

4. THỰC HIỆN TỐI ƯU HÓA TRUY VẤN TRÊN MÔI TRƯỜNG PHÂN TÁN

4.1. Câu truy vấn đơn giản chưa tối ưu

Cho biết thông tin những sản phẩm(PRO_ID, PRODUCT_NAME, STATUS) ở chi nhánh 'Mini mart chi nhanh Quan 9' có ngày bán ra là 3/2022 và có số lượng bán lớn hơn 2.

```
SELECT  DISTINCT P.PRO_ID, PRODUCT_NAME, STATUS
FROM    PRODUCT P, WAREHOUSE_SALES WS, BILL B,
          BILL_DETAILS BDT, BRANCH BR
WHERE   BR.BRANCH_ID = WS.BRANCH_ID AND
          WS.PRO_ID = P.PRO_ID AND
          P.PRO_ID = BDT.PRO_ID AND
          BDT.BILL_ID = B.BILL_ID AND
          BR.BRANCH_NAME = 'Mini mart Quan 9' AND
          EXTRACT(YEAR FROM BILL_DATE) = 2022
AND EXTRACT(MONTH FROM BILL_DATE) = 3
AND AMOUNT > 2;
```

4.2. EXPLAIN QUERY câu truy vấn đơn giản

```
SELECT  /*+ GATHER_PLAN_STATISTICS */ DISTINCT P.PRO_ID,
          PRODUCT_NAME, STATUS
FROM    PRODUCT P, WAREHOUSE_SALES WS, BILL B,
          BILL_DETAILS BDT, BRANCH BR
WHERE   BR.BRANCH_ID = WS.BRANCH_ID AND
          WS.PRO_ID = P.PRO_ID AND
          P.PRO_ID = BDT.PRO_ID AND
          BDT.BILL_ID = B.BILL_ID AND
          BR.BRANCH_NAME = 'Mini mart Quan 9' AND
          EXTRACT(YEAR FROM BILL_DATE) = 2022
AND EXTRACT(MONTH FROM BILL_DATE) = 3
AND AMOUNT > 2;
```

```

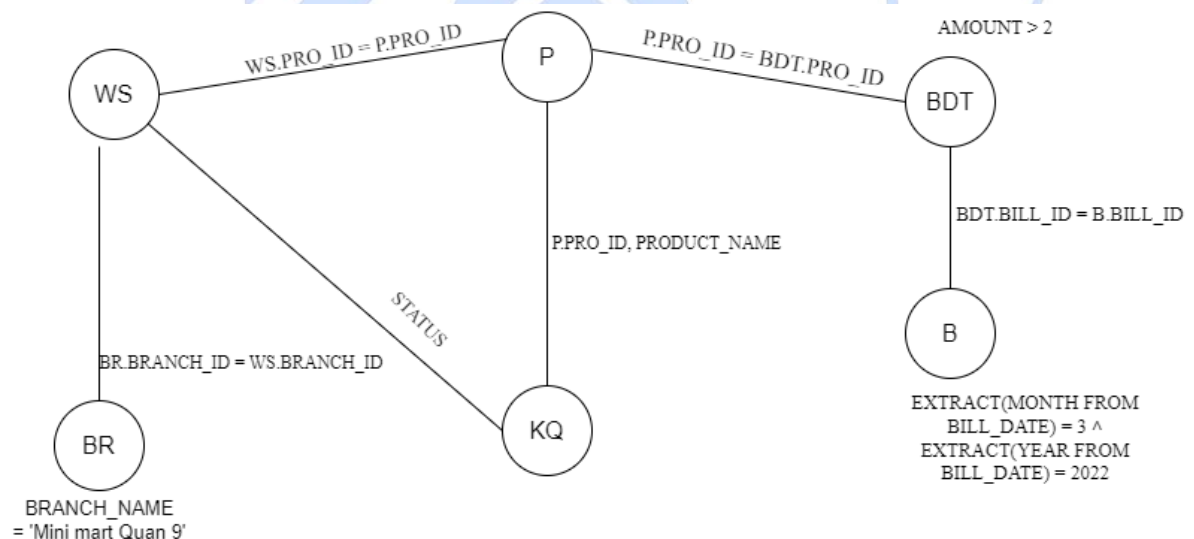
SELECT * FROM
TABLE(DBMS_XPLAN.display_cursor(format=>'ALLSTATS LAST'));

```

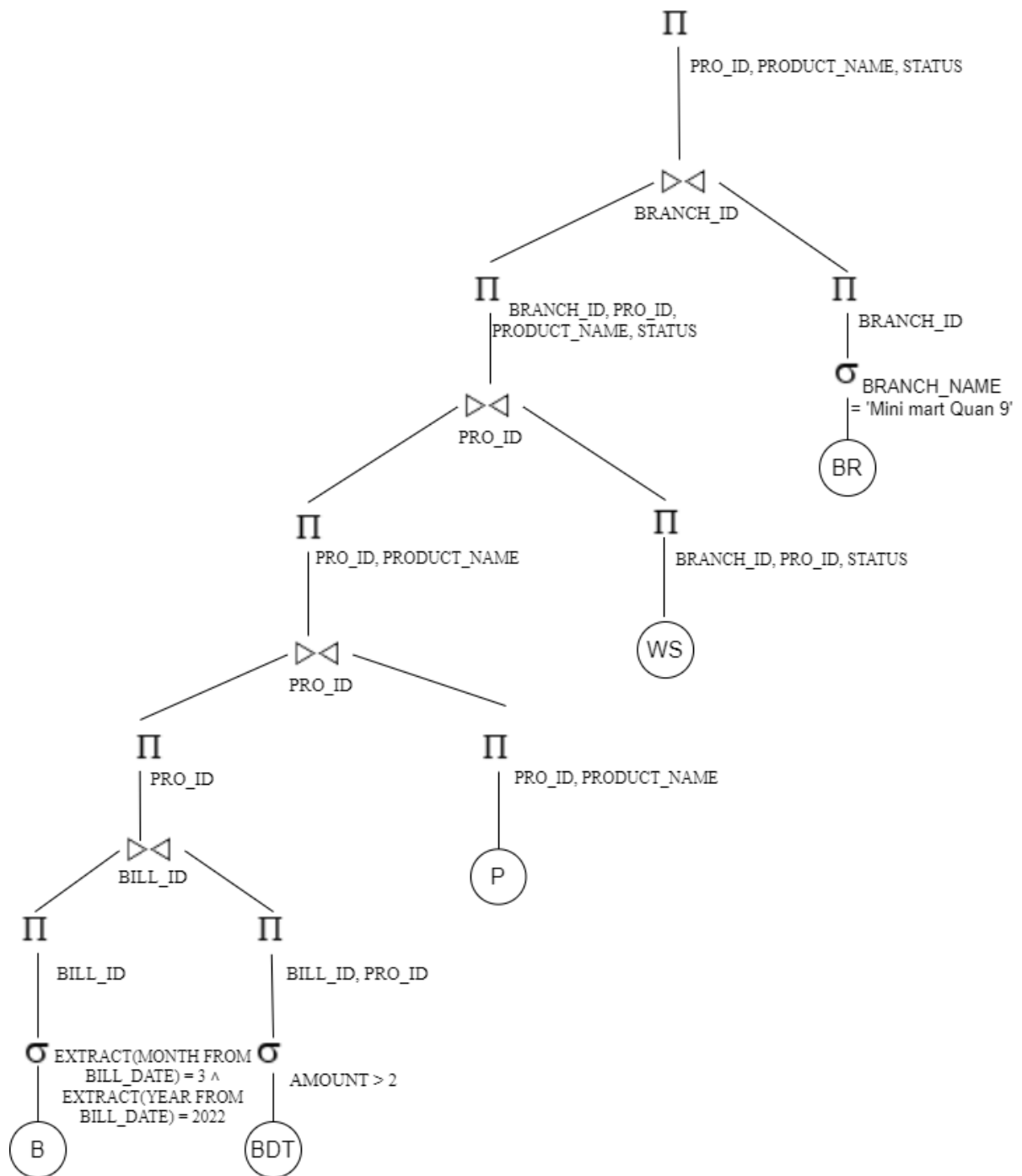
Kết quả

Id	Operation	Name	Starts	E-Rows	A-Rows	A-Time	Buffers	OMem	1Mem	Used-Mem
0	SELECT STATEMENT		1		3	00:00:00.02	30			
1	HASH UNIQUE		1	1	3	00:00:00.02	30	1230K	1230K	760K (0)
* 2	HASH JOIN SEMI		1	1	3	00:00:00.02	30	1123K	1123K	1286K (0)
* 3	HASH JOIN SEMI		1	1	120	00:00:00.02	24	1133K	1133K	605K (0)
* 4	HASH JOIN		1	1	240	00:00:00.01	18	1209K	1209K	1488K (0)
5	MERGE JOIN CARTESIAN		1	1	240	00:00:00.01	12			
* 6	TABLE ACCESS FULL	BILL	1	1	8	00:00:00.01	6			
7	BUFFER SORT		8	30	240	00:00:00.01	6	2048	2048	2048 (0)
8	TABLE ACCESS FULL	WAREHOUSE_SALES	1	30	30	00:00:00.01	6			
9	TABLE ACCESS FULL	PRODUCT	1	1	40	00:00:00.01	6			
* 10	TABLE ACCESS FULL	BRANCH	1	1	1	00:00:00.01	6			
* 11	TABLE ACCESS FULL	BILL_DETAILS	1	6	12	00:00:00.01	6			

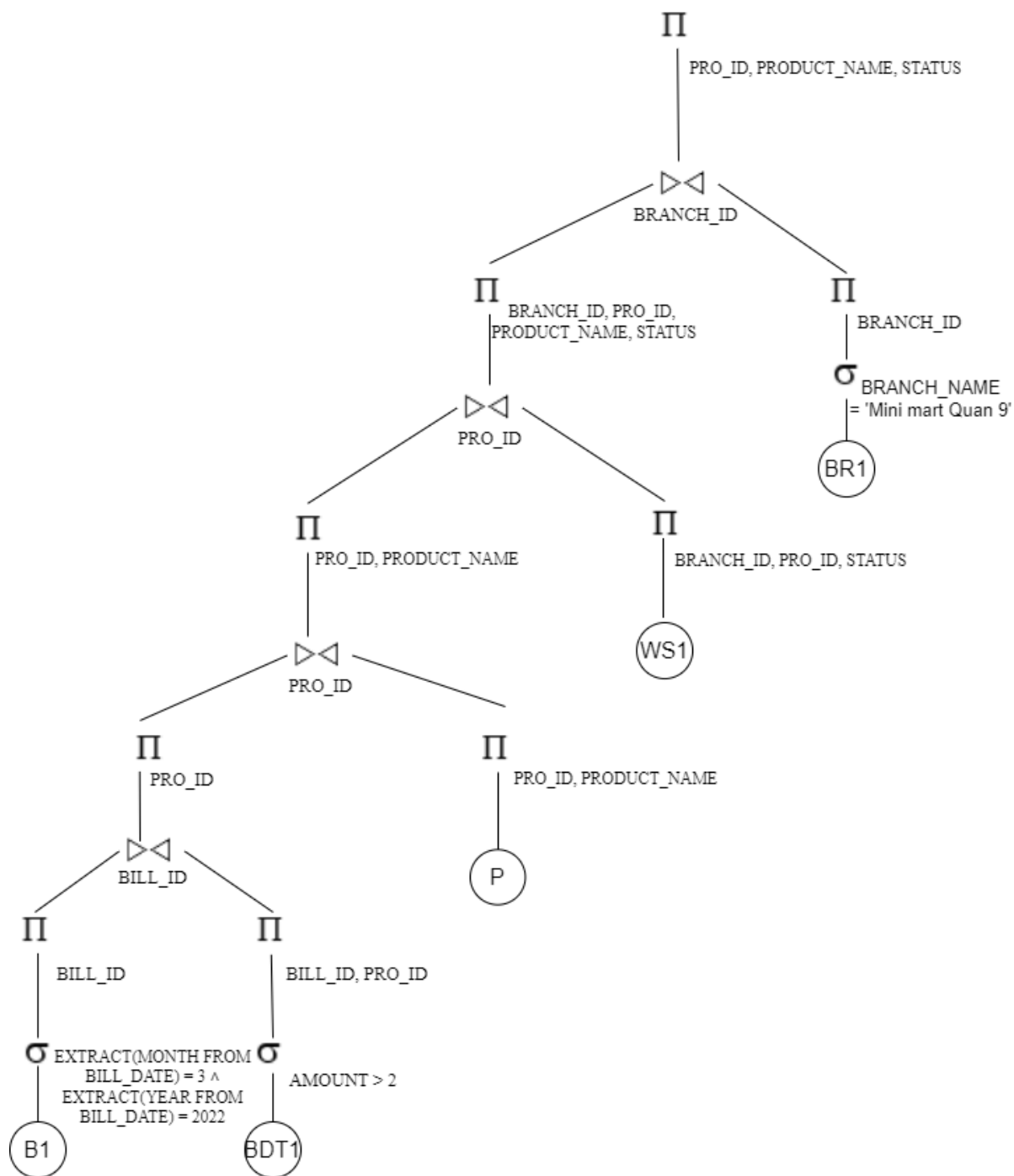
4.3. Tối ưu hóa câu truy vấn cục bộ, phân tán



Hình 1. Đồ thị truy vấn



Hình 2. Cây truy vấn tối ưu trên môi trường tập trung



Hình 3. Cây truy vấn tối ưu trên môi trường phân tán

4.4. Viết lại câu query trên môi trường phân tán

```
SELECT DISTINCT PRO_ID, PRODUCT_NAME, STATUS
FROM ((SELECT BRANCH_ID, E.PRO_ID, PRODUCT_NAME, STATUS
      FROM ((SELECT C.PRO_ID, PRODUCT_NAME
            FROM ((SELECT PRO_ID
                  FROM ((SELECT BILL_ID FROM CN1.BILL
                        WHERE EXTRACT(YEAR FROM BILL_DATE) = 2022
AND
                        EXTRACT(MONTH FROM
BILL_DATE) = 3) A
                  INNER JOIN (SELECT BILL_ID, PRO_ID
                        FROM CN1.BILL_DETAILS WHERE AMOUNT
> 2) B
                        ON A.BILL_ID = B.BILL_ID)) C
                  INNER JOIN (SELECT PRO_ID, PRODUCT_NAME
                        FROM PRODUCT) D ON C.PRO_ID = D.PRO_ID)) E
            INNER JOIN (SELECT BRANCH_ID, PRO_ID, STATUS
                  FROM CN1.WAREHOUSE_SALES) F ON E.PRO_ID =
F.PRO_ID)) G
      INNER JOIN (SELECT BRANCH_ID
            FROM CN1.BRANCH WHERE BRANCH_NAME = 'Mini mart
Quan 9') H
      ON G.BRANCH_ID = H.BRANCH_ID);
```

EXPLAIN query trên môi trường phân tán

```

SELECT /*+ GATHER_PLAN_STATISTICS */ DISTINCT PRO_ID,
PRODUCT_NAME, STATUS
FROM ((SELECT BRANCH_ID, E.PRO_ID, PRODUCT_NAME, STATUS
      FROM ((SELECT C.PRO_ID, PRODUCT_NAME
            FROM ((SELECT PRO_ID
                  FROM ((SELECT BILL_ID FROM CN1.BILL
                        WHERE EXTRACT(YEAR FROM BILL_DATE) = 2022
AND
                        EXTRACT(MONTH FROM
BILL_DATE) = 3) A
                  INNER JOIN (SELECT BILL_ID, PRO_ID
                        FROM CN1.BILL_DETAILS WHERE AMOUNT
> 2) B
                        ON A.BILL_ID = B.BILL_ID)) C
                  INNER JOIN (SELECT PRO_ID, PRODUCT_NAME
                        FROM PRODUCT) D ON C.PRO_ID = D.PRO_ID)) E
                  INNER JOIN (SELECT BRANCH_ID, PRO_ID, STATUS
                        FROM CN1.WAREHOUSE_SALES) F ON E.PRO_ID =
F.PRO_ID)) G
                  INNER JOIN (SELECT BRANCH_ID
                        FROM CN1.BRANCH WHERE BRANCH_NAME = 'Mini mart
Quan 9') H
                        ON G.BRANCH_ID = H.BRANCH_ID);

SELECT * FROM
TABLE(DBMS_XPLAN.display_cursor(format=>'ALLSTATS LAST'));

```

Kết quả

Id	Operation	Name	Starts	E-Rows	A-Rows	A-Time	Buffers	OMem	lMem	Used-Mem
0	SELECT STATEMENT		1		2	00:00:00.01	24			
1	HASH UNIQUE		1	1	2	00:00:00.01	24	1230K	1230K	645K (0)
2	NESTED LOOPS SEMI		1	1	2	00:00:00.01	24			
3	NESTED LOOPS		1	1	2	00:00:00.01	22			
* 4	HASH JOIN		1	1	2	00:00:00.01	18	1236K	1236K	1643K (0)
5	MERGE JOIN CARTESIAN		1	1	60	00:00:00.01	12			
* 6	TABLE ACCESS FULL	BILL	1	1	4	00:00:00.01	6			
7	BUFFER SORT		4	15	60	00:00:00.01	6	2048	2048	2048 (0)
8	TABLE ACCESS FULL	WAREHOUSE_SALES	1	15	15	00:00:00.01	6			
* 9	TABLE ACCESS FULL	BILL_DETAILS	1	1	7	00:00:00.01	6			
10	TABLE ACCESS BY INDEX ROWID	PRODUCT	2	1	2	00:00:00.01	4			
* 11	INDEX UNIQUE SCAN	PK_PRO	2	1	2	00:00:00.01	2			
* 12	TABLE ACCESS BY INDEX ROWID	BRANCH	1	1	1	00:00:00.01	2			
* 13	INDEX UNIQUE SCAN	BRANCH_PK	1	1	1	00:00:00.01	1			

Nhận xét câu truy vấn tập trung chưa được tối ưu và đã được tối ưu:

Thời gian thực hiện câu truy vấn đã được tối ưu nhanh hơn thời gian thực hiện câu truy vấn chưa được tối ưu

5. CƠ CHẾ NHÂN BẢN TRONG MICROSOFT SQL SERVER**5.1. Tìm hiểu cơ chế nhân bản, cách thức hoạt động****5.1.1. Tổng quan về nhân bản:**

Nhân bản là một kỹ thuật quan trọng và hữu hiệu trong việc phân bố cơ sở dữ liệu (CSDL) và thực thi các Stored procedure.

Nhân bản là một kỹ thuật quan trọng và hữu hiệu trong việc phân bố cơ sở dữ liệu (CSDL) và thực thi các Stored procedure. Kỹ thuật nhân bản trong SQL Server cho phép bạn tạo ra những bản sao dữ liệu giống hệt nhau, di chuyển các bản sao này đến những vùng khác nhau và đồng bộ hoá dữ liệu một cách tự động để tất cả các bản sao có cùng giá trị dữ liệu. Nhân bản có thể thực thi giữa những CSDL trên cùng một server hay những server khác nhau được kết nối bởi mạng LANs, WANs hay Internet.

SQL Server đã đưa ra nhiều cơ chế nhân bản để đáp ứng các yêu cầu khác nhau của ứng dụng. Mỗi loại cung cấp các khả năng và thuộc tính khác nhau nhằm đạt đến mục tiêu của tính độc lập “Site” và sự nhất quán các giao dịch

Mục tiêu chính của nhân bản

SQL Server đã đưa ra nhiều cơ chế nhân bản để đáp ứng các yêu cầu khác nhau của ứng dụng. Mỗi loại cung cấp các khả năng và thuộc tính khác nhau nhằm đạt đến mục tiêu của tính độc lập “Site” và sự nhất quán dữ liệu.

Nhất quán dữ liệu (Data consistency)

Có 2 cách để đạt được tính nhất quán dữ liệu:

- Nhất quán giao dịch (Transactional Consistency)
- Hội tụ dữ liệu (Data Convergence)

Nhất quán giao dịch

- Bảo đảm tất cả dữ liệu giống nhau tại mọi site ở bất kì thời điểm.

- Tất cả giao dịch thực hiện tại một site duy nhất.

Có 2 loại :

Nhất quán lập tức (Immediate Transactional Consistency hay Tight Consistency):

Ở kiểu này, tất cả các site được bảo đảm là luôn thấy cùng giá trị dữ liệu tại cùng một thời điểm. Cách duy nhất để đạt được nhất quán giao dịch (transactional consistency) trong môi trường cập nhật phân tán (distributed update environment) là sử dụng 2-phase commit protocol giữa tất cả site tham gia (participating site). Mỗi site phải commit đồng thời mọi thay đổi hoặc không site nào commit những thay đổi. Giải pháp này rõ ràng không khả thi khi số lượng site quá lớn.

Nhất quán ngầm (Latent Transactional Consistency hay Loose Consistency) :

Có một sự nhất quán ngầm giữa các site tham gia do có một sự trì hoãn trong việc phản ánh các giá trị dữ liệu đến các site tham gia và vào lúc này các site không bảo đảm có cùng giá trị dữ liệu. Việc sửa đổi các giá trị dữ liệu có thể bị trì hoãn đủ lâu để tất cả các site cùng cập nhật, sau đó tất cả các site sẽ có cùng giá trị dữ liệu. Ngoài ra các giá trị dữ liệu này cũng phải giống với những giá trị đạt được khi thực hiện các công việc tại một site. Sự khác nhau duy nhất giữa nhất quán giao dịch lập tức và nhất quán giao dịch ngầm là dữ liệu có nhất quán tại cùng một lúc hay không.

Hội tụ dữ liệu

Với sự hội tụ dữ liệu, tất cả các site có thể quy về cùng một giá trị dữ liệu nhưng không nhất thiết là giá trị dữ liệu này bị gây ra bởi những tác vụ được làm trên một site duy nhất. User có thể tự do thao tác trên các site theo các cách khác nhau. Khi các nút (node) đồng bộ, tất cả các site sẽ hội tụ về cùng một giá trị. Nếu xung đột gây ra bởi sự sửa đổi cùng một dữ liệu tại những site khác nhau thì những sửa đổi này sẽ được giải quyết một cách tự động (chọn site có độ ưu tiên cao hơn hay site đưa sửa đổi đến trước...).

Độc lập site (site autonomy)

Độc lập site xét đến ảnh hưởng của những thao tác trên một site đến các site khác. Thường độc lập site càng tăng thì tính nhất quán dữ liệu giảm. Nhân bản kết hợp (Merge replication) có mức độc lập site cao nhất, tạo ra sự hội tụ nhưng lại không đảm bảo tính nhất quán dữ liệu. 2PC (two phase commit) có tính nhất quán dữ liệu cao nhưng lại không có tính độc lập site. Những giải pháp khác thì thường ở giữa hai tính này.

Kiến trúc nhân bản:

Các thành phần chính của nhân bản:

- Publisher: Là một server tạo dữ liệu để nhân bản đến các server khác. Nó xác định dữ liệu nào được nhân bản, dữ liệu nào thay đổi và duy trì những thông tin về các công bố tại site đó.
- Subscriber: Là một server lưu giữ nhân bản và nhận các tác vụ cập nhật. SQL Server 2000 cho phép Subscriber cập nhật dữ liệu nhưng quá trình cập nhập ở Subscriber không giống như ở Publisher. Một Subscriber có thể là một Publisher của các Subscriber khác.
- Distributor: Là một server mà chứa CSDL phân tán (distribution database) và lưu trữ metadata, history data và transaction. SQL Server sử dụng CSDL phân tán để lưu và chuyển (store_and_forward) dữ liệu nhân bản từ Publisher đến các Subscriber.

Có 2 loại Distributor : Local Distributor và remote Distributor.

- Publication: Đơn giản là một tập hợp các mẫu dữ liệu (article). Một mẫu là một nhóm dữ liệu được nhân bản. Một mẫu có thể bao gồm một table hay chỉ là một vài hàng (horizontal fragment) hay cột (vertical fragment). Một Publication thường gồm nhiều mẫu.

Chiều di chuyển dữ liệu

Có 2 kiểu di chuyển dữ liệu:

Push subscription

- Publisher đẩy (push) những thay đổi đến Subscriber mà không quan tâm Subscriber có cập nhật hay không.

- Push subscription được sử dụng trong những ứng dụng mà yêu cầu gửi những thay đổi đến Subscriber ngay khi những thay đổi này xảy ra ở Publisher.
- Push Subscription giúp việc quản lý các Subscriber đơn giản và tập trung hơn, đồng thời giúp bảo mật tốt hơn vì quá trình khởi động (initialization process) sẽ được quản lý tại một chỗ . Nhưng vì thế, Distributor có thể phải đảm nhận nhiều quá trình phân bổ subscription đến các Subscriber cùng một lúc . Điều này dễ dẫn đến hiện tượng thắt cổ chai (bottleneck) .
- Mô hình này không thích hợp khi số lượng các Subscriber trở nên quá lớn.
- Push subscription gây ra 1 phí xử lý cao hơn tại Publisher. Để tránh hiện tượng này, những thay đổi có thể được đẩy đến Subscriber theo một lịch định kì.

Pull subscription

- Subscriber kéo (pull) những thay đổi tại Publisher về theo một khoảng thời gian định kì.
- Tốt cho những user độc lập thay đổi bởi vì chúng cho phép user xác định khi nào thì những thay đổi dữ liệu được đồng bộ
- Ngược với push subscription ,pull subscription bảo mật thấp nhưng cho phép số lượng Subscriber cao hơn .
- Một publication có thể sử dụng cả hai push và pull subscription.

Tác nhân (Agent)

Việc thiết kế các nhân bản có thể tạo ra 1 hay nhiều agent.

Snapshot agent:

- Chuẩn bị lược đồ, data file, stored procedure
- Lưu snapshot lên Distributor và ghi lại những thông tin về trạng thái đồng bộ vào CSDL phân bố (distribution database) .
- Mỗi publication có 1 snapshot agent riêng chạy trên Distributor và liên kết với Publisher.

Log Reader agent:

- Di chuyển những transaction cần nhân bản từ transaction log trên Publisher đến CSDL phân bố .

- Mỗi publication dùng nhân bản transaction có một log reader agent, chạy trên Distributor và liên kết (connect) đến Publisher.

Distribution Agent:

- Di chuyển transaction và những tác vụ sao chép giữ trong CSDL phân bố đến Subscriber.

- TH: Nhân bản transaction hay snapshot mà đồng bộ lập tức (immediate synchronization): khi 1 push subscription được tạo, mỗi publication có 1 distribution agent riêng, chạy trên Distributor và liên kết với Subscriber.

- TH: Nhân bản transaction và snapshot không đồng bộ lập tức : Publisher và Subscriber sẽ dùng chung distribution agent , chạy trên Distributor và liên kết với Subscriber.

- TH: pull subscription đến snapshot publication hay transactional publication: có distribution agent, chạy trên Subscriber

- Nhân bản kết hợp (merge replication) không có distribution agent.

Merge agent:

Di chuyển và điều hòa những thay đổi dữ liệu xảy ra sau khi 1 snapshot khởi động (initial snapshot) được tạo. Mỗi merge publication có một merge agent, liên kết và cập nhật được với cả hai Publisher và Subscriber.

5.2. Demo cơ chế nhân bản trên nhiều cụm máy

Xem demo tại video phần cuối

E. PHÂN CÔNG CÔNG VIỆC

Công việc	Trân	Quyên	Đức	Huy
Yêu cầu 1		X		
Yêu cầu 2	X			
Yêu cầu 3			X	
Yêu cầu 4				X
Yêu cầu 5	X	X		

Làm file báo cáo	X	X	X	X
Edit video	X			

TÀI LIỆU THAM KHẢO

[1] Khoa Hệ thống Thông tin - Trường Đại học Công nghệ Thông tin (ĐHQG - TPHCM), “Đề thi cuối kì I năm học 2017 - 2018 môn Cơ sở dữ liệu phân tán.”

Dec. 19, 2017.

[2] Thông tin - Trường Đại học Công nghệ Thông tin (ĐHQG - TPHCM), Oct. 30, 2021. [Online]. Available:

https://courses.uit.edu.vn/pluginfile.php/313931/mod_resource/content/1/10_KiemTra3.pdf

