

THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

Nguyễn Thị Mỹ Trân
Khoa Hệ thống Thông Tin
Đại học Công Nghệ Thông Tin
Thành phố Hồ Chí Minh, Việt Nam
20520322@gm.uit.edu.vn

Tôn Nữ Tú Quyền
Khoa Hệ thống Thông Tin
Đại học Công Nghệ Thông Tin
Thành phố Hồ Chí Minh, Việt Nam
20520296@gm.uit.edu.vn

Thái Tăng Đức
Khoa Hệ thống Thông Tin
Đại học Công Nghệ Thông Tin
Thành phố Hồ Chí Minh, Việt Nam
20521203@gm.uit.edu.vn

Trần Anh Huy
Khoa Hệ thống Thông Tin
Đại học Công Nghệ Thông Tin
Thành phố Hồ Chí Minh, Việt Nam
20520296@gm.uit.edu.vn

Tóm tắt: Đối với một hệ CSDL nằm phân tán trên mạng máy tính thì hệ quản trị CSDL có ý nghĩa rất quan trọng vì phải đảm bảo tính nhất quán và toàn vẹn của dữ liệu, đảm bảo cho các chương trình người dùng truy xuất đến CSDL phân tán như là một khối CSDL duy nhất. Ngoài ra hệ quản trị CSDL còn phải đảm bảo chức năng phân quyền truy nhập và bảo mật trên đường truyền. Trong các hệ quản trị CSDL phân tán hiện nay thì hệ quản trị CSDL Oracle được đánh giá là ưu việt nhất với các chức năng tiên tiến: phân tán, tạo bản sao (replication), bảo mật cao, tính nhất quán dữ liệu, thủ tục (storedprocedure), kích hoạt (triggers), khóa dữ liệu đến từng bản ghi, chạy thông suốt trên 120 loại phần cứng từ máy lớn đến máy nhỏ và 19 hệ điều hành, hỗ trợ hầu hết các giao thức mạng, là CSDL đầu tiên trên thế giới tích hợp Web.

Từ khóa: Phân tán oracle, phân mảnh, query phân tán..

I. 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 = $\delta_{((BRANCH_NAME = '@Mini mart chi nhanh Quan 1'))}$ 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

```
CREATE TABLE CN1.BRANCH(  
    BRANCH_ID VARCHAR2(5) CONSTRAINT  
    BRANCH_PK PRIMARY KEY,  
    BRANCH_NAME VARCHAR2(200),  
    ADDRESS VARCHAR2(200)
```

```
);  
CREATE TABLE CN2.BRANCH(  
    BRANCH_ID VARCHAR2(5) CONSTRAINT  
    BRANCH_PK PRIMARY KEY,  
    BRANCH_NAME VARCHAR2(200),  
    ADDRESS VARCHAR2(200)
```

```
);  
CREATE TABLE CN1.EMPLOYEE(  
    EMP_ID NUMBER CONSTRAINT EMPLOYEE_PK  
    PRIMARY KEY,
```

```
FIRST_NAME VARCHAR2(200),  
    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)  
);  
CREATE TABLE CN2.EMPLOYEE(  
    EMP_ID NUMBER CONSTRAINT EMPLOYEE_PK  
    PRIMARY KEY,  
    FIRST_NAME VARCHAR2(200),  
    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)  
);  
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,  
    SPENT_MONEY NUMBER,  
    POINT INT  
);  
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  
);  
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,
    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)
);
CREATE TABLE
CN2.WAREHOUSE_MANAGEMENT
(
    BRANCH_ID VARCHAR2(5),
    PRO_ID NUMBER,
    IMPORTED_DATE DATE,
    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)
);
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)
);
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)
);
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)
);
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)
);
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
KEY(BILL_ID) REFERENCES BILL(BILL_ID),
    CONSTRAINT FK_B_DETAILS_PRO FOREIGN
KEY(PRO_ID) REFERENCES PRODUCT(PRO_ID)
);
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
KEY(BILL_ID) REFERENCES BILL(BILL_ID),
    CONSTRAINT FK_B_DETAILS_PRO FOREIGN
KEY(PRO_ID) REFERENCES PRODUCT(PRO_ID)
);

```

1.3.2. Dml
CN1 CN2

```

INSERT INTO CN1.BRANCH(BRANCH_ID,
BRANCH_NAME, ADDRESS)
VALUES ('CN1', 'Mini mart Quan 9', '5, Le Van Viet, Quan
9, TPHCM'); INSERT INTO
CN2.BRANCH(BRANCH_ID, BRANCH_NAME,
ADDRESS)
VALUES ('CN2', 'Mini mart chi nhanh Quan 1', '55, Ly
Thanh Tong, Quan 1, TPHCM');
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');
INSERT INTO CN1.EMPLOYEE
VALUES (CN1.EMP_ID_SEQUENCE.nextval,'Ton Nu
Tu', 'Quyen','Female', TO_DATE('05/01/2002',
'dd/mm/yyyy'), '0977164279', 'KTX khu A',
TO_DATE('21/02/2022', 'dd/mm/yyyy'),5000000, 'Cashier',
'CN1');
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');
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');
INSERT INTO CN1.EMPLOYEE
VALUES (CN1.EMP_ID_SEQUENCE.nextval,'Nguyen
Thi Minh', 'Triet','Female', TO_DATE('05/09/2002',
'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 khu A', TO_DATE('21/02/2022',
'dd/mm/yyyy'), 5000000, 'Quality Checker', 'CN1');
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');
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'); 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');
INSERT INTO CN2.EMPLOYEE
VALUES (CN2.EMP_ID_SEQUENCE.nextval,'Ton Nu
Tu', 'Tran','Female', TO_DATE('05/01/2002',
'dd/mm/yyyy'), '0977164279', 'KTX khu A',
TO_DATE('21/02/2022', 'dd/mm/yyyy'),5000000, 'Cashier',
'CN2');
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');
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');
INSERT INTO CN2.EMPLOYEE
VALUES (CN2.EMP_ID_SEQUENCE.nextval,'Nguyen
Thi Minh', 'Huong','Female', TO_DATE('05/09/2002',
'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 khu A', TO_DATE('21/02/2022',
'dd/mm/yyyy'), 5000000, 'Quality Checker', 'CN2');
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');
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');

```

```

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'),
TO_DATE('01/09/2022', 'dd/mm/yyyy'), 279072, 0);
INSERT INTO CN1.CUSTOMER
VALUES(CN1.CUS_ID_SEQUENCE.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);
INSERT INTO CN1.CUSTOMER
VALUES(CN1.CUS_ID_SEQUENCE.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);
INSERT INTO CN1.CUSTOMER
VALUES(CN1.CUS_ID_SEQUENCE.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);
INSERT INTO CN1.CUSTOMER
VALUES(CN1.CUS_ID_SEQUENCE.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);
INSERT INTO CN1.CUSTOMER
VALUES(CN1.CUS_ID_SEQUENCE.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);
INSERT INTO CN1.CUSTOMER
VALUES(CN1.CUS_ID_SEQUENCE.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);
INSERT INTO CN1.CUSTOMER
VALUES(CN1.CUS_ID_SEQUENCE.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);
INSERT INTO CN1.CUSTOMER
VALUES(CN1.CUS_ID_SEQUENCE.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);
INSERT INTO CN1.CUSTOMER
VALUES(CN1.CUS_ID_SEQUENCE.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);
INSERT INTO CN1.CUSTOMER
VALUES(CN1.CUS_ID_SEQUENCE.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);
INSERT INTO CN1.CUSTOMER
VALUES(CN1.CUS_ID_SEQUENCE.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);
INSERT INTO CN1.CUSTOMER

```

```

VALUES(CN1.CUS_ID_SEQUENCE.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);
INSERT INTO CN1.CUSTOMER
VALUES(CN1.CUS_ID_SEQUENCE.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);
INSERT INTO CN1.CUSTOMER
VALUES(CN1.CUS_ID_SEQUENCE.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);
INSERT INTO CN1.CUSTOMER
VALUES(CN1.CUS_ID_SEQUENCE.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);
INSERT INTO CN1.CUSTOMER
VALUES(CN1.CUS_ID_SEQUENCE.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);
INSERT INTO CN1.CUSTOMER
VALUES(CN1.CUS_ID_SEQUENCE.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);
INSERT INTO CN1.CUSTOMER
VALUES(CN1.CUS_ID_SEQUENCE.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);
INSERT INTO CN1.CUSTOMER
VALUES(CN1.CUS_ID_SEQUENCE.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);
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);
INSERT INTO CN2.CUSTOMER
VALUES(CN2.CUS_ID_SEQUENCE.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);
INSERT INTO CN2.CUSTOMER
VALUES(CN2.CUS_ID_SEQUENCE.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);
INSERT INTO CN2.CUSTOMER
VALUES(CN2.CUS_ID_SEQUENCE.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);
INSERT INTO CN2.CUSTOMER

```

```

VALUES(CN2.CUS_ID_SEQUENCE.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);
INSERT INTO CN2.CUSTOMER
VALUES(CN2.CUS_ID_SEQUENCE.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);
INSERT INTO CN2.CUSTOMER
VALUES(CN2.CUS_ID_SEQUENCE.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);
INSERT INTO CN2.CUSTOMER
VALUES(CN2.CUS_ID_SEQUENCE.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);
INSERT INTO CN2.CUSTOMER
VALUES(CN2.CUS_ID_SEQUENCE.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);
INSERT INTO CN2.CUSTOMER
VALUES(CN2.CUS_ID_SEQUENCE.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);
INSERT INTO CN2.CUSTOMER
VALUES(CN2.CUS_ID_SEQUENCE.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);
INSERT INTO CN2.CUSTOMER
VALUES(CN2.CUS_ID_SEQUENCE.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);
INSERT INTO CN2.CUSTOMER
VALUES(CN2.CUS_ID_SEQUENCE.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);
INSERT INTO CN2.CUSTOMER
VALUES(CN2.CUS_ID_SEQUENCE.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);
INSERT INTO CN2.CUSTOMER
VALUES(CN2.CUS_ID_SEQUENCE.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);
INSERT INTO CN2.CUSTOMER
VALUES(CN2.CUS_ID_SEQUENCE.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);
INSERT INTO CN2.CUSTOMER

```

```

VALUES(CN2.CUS_ID_SEQUENCE.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);
INSERT INTO CN2.CUSTOMER
VALUES(CN2.CUS_ID_SEQUENCE.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);
INSERT INTO CN2.CUSTOMER
VALUES(CN2.CUS_ID_SEQUENCE.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);
INSERT INTO CN2.CUSTOMER
VALUES(CN2.CUS_ID_SEQUENCE.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);
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);
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);
INSERT INTO CN1.PRODUCT
VALUES (CN1.PRO_ID_SEQUENCE.nextval, 'Soap',
'France', 10000, 15000, TO_DATE('01/01/2022',
'dd/mm/yyyy'), null, 'Toiletries', 2, 29);
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);
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);
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);
INSERT INTO CN1.PRODUCT
VALUES (CN1.PRO_ID_SEQUENCE.nextval, 'Apple',
'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, '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,'NoteBook',
'China', 5000, 8000, TO_DATE('01/01/2022',
'dd/mm/yyyy'), null, 'Requisite', 2, 25);
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);
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);
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);
INSERT INTO CN1.PRODUCT
VALUES (CN1.PRO_ID_SEQUENCE.nextval, 'Strong
bow', 'USA', 5000, 9000, 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, '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, '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);
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);
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);
INSERT INTO CN2.PRODUCT
VALUES (CN2.PRO_ID_SEQUENCE.nextval, 'Soap',
'France', 10000, 15000, TO_DATE('01/01/2022',
'dd/mm/yyyy'), null, 'Toiletries', 2, 29);
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);
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);
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);
INSERT INTO CN2.PRODUCT
VALUES (CN2.PRO_ID_SEQUENCE.nextval, 'Apple',
'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, '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, 'NoteBook',
'China', 5000, 8000, TO_DATE('01/01/2022',
'dd/mm/yyyy'), null, 'Requisite', 2, 25);
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);
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);
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);
INSERT INTO CN2.PRODUCT
VALUES (CN2.PRO_ID_SEQUENCE.nextval, 'Strong
bow', 'USA', 5000, 9000, 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, '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

```



```

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', 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 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,
TO_DATE('01/01/2022', 'dd/mm/yyyy'), 53295);
INSERT INTO CN1.BILL
VALUES(BILL_ID_SEQUENCE.nextval, 200003, 300011,
TO_DATE('10/02/2022', 'dd/mm/yyyy'), 75582);
INSERT INTO CN1.BILL
VALUES(BILL_ID_SEQUENCE.nextval, 200002, 300003,
TO_DATE('20/02/2022', 'dd/mm/yyyy'), 77520);
INSERT INTO CN1.BILL
VALUES(BILL_ID_SEQUENCE.nextval, 200003, 300003,
TO_DATE('01/03/2022', 'dd/mm/yyyy'), 26163);
INSERT INTO CN1.BILL
VALUES(BILL_ID_SEQUENCE.nextval, 200002, 300015,
TO_DATE('09/03/2022', 'dd/mm/yyyy'), 72675);
INSERT INTO CN1.BILL
VALUES(BILL_ID_SEQUENCE.nextval, 200003, 300015,
TO_DATE('18/03/2022', 'dd/mm/yyyy'), 96900);
INSERT INTO CN1.BILL
VALUES(BILL_ID_SEQUENCE.nextval, 200002, 300006,
TO_DATE('25/03/2022', 'dd/mm/yyyy'), 87210);
INSERT INTO CN1.BILL
VALUES(BILL_ID_SEQUENCE.nextval, 200003, 300007,
TO_DATE('05/04/2022', 'dd/mm/yyyy'), 11628);
INSERT INTO CN1.BILL
VALUES(BILL_ID_SEQUENCE.nextval, 200002, 300009,
TO_DATE('17/04/2022', 'dd/mm/yyyy'), 145350);
INSERT INTO CN1.BILL

```

```

VALUES(BILL_ID_SEQUENCE.nextval, 200003, 300019,
TO_DATE('23/04/2022', 'dd/mm/yyyy'), 77520);
INSERT INTO CN2.BILL
VALUES(BILL_ID_SEQUENCE.nextval, 600002, 300011,
TO_DATE('01/01/2022', 'dd/mm/yyyy'), 53295);
INSERT INTO CN2.BILL
VALUES(BILL_ID_SEQUENCE.nextval, 600003, 300011,
TO_DATE('10/02/2022', 'dd/mm/yyyy'), 75582);
INSERT INTO CN2.BILL
VALUES(BILL_ID_SEQUENCE.nextval, 600002, 300006,
TO_DATE('20/02/2022', 'dd/mm/yyyy'), 77520);
INSERT INTO CN2.BILL
VALUES(BILL_ID_SEQUENCE.nextval, 600003, 300013,
TO_DATE('01/03/2022', 'dd/mm/yyyy'), 26163);
INSERT INTO CN2.BILL
VALUES(BILL_ID_SEQUENCE.nextval, 600002, 300016,
TO_DATE('09/03/2022', 'dd/mm/yyyy'), 72675);
INSERT INTO CN2.BILL
VALUES(BILL_ID_SEQUENCE.nextval, 600003, 300016,
TO_DATE('18/03/2022', 'dd/mm/yyyy'), 96900);
INSERT INTO CN2.BILL
VALUES(BILL_ID_SEQUENCE.nextval, 600002, 300009,
TO_DATE('25/03/2022', 'dd/mm/yyyy'), 87210);
INSERT INTO CN2.BILL
VALUES(BILL_ID_SEQUENCE.nextval, 600003, 300017,
TO_DATE('05/04/2022', 'dd/mm/yyyy'), 11628);
INSERT INTO CN2.BILL
VALUES(BILL_ID_SEQUENCE.nextval, 600002, 300019,
TO_DATE('17/04/2022', 'dd/mm/yyyy'), 145350);
INSERT INTO CN2.BILL
VALUES(BILL_ID_SEQUENCE.nextval, 600003, 300003,
TO_DATE('23/04/2022', 'dd/mm/yyyy'), 77520);
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);
INSERT INTO CN1.BILL_DETAILS(BILL_ID, PRO_ID,
AMOUNT)
VALUES (500002, 400005, 2);
INSERT INTO CN1.BILL_DETAILS(BILL_ID, PRO_ID,
AMOUNT)
VALUES (500002, 400003, 1);
INSERT INTO CN1.BILL_DETAILS(BILL_ID, PRO_ID,
AMOUNT)
VALUES (500003, 400015, 10);
INSERT INTO CN1.BILL_DETAILS(BILL_ID, PRO_ID,
AMOUNT)
VALUES (500004, 400005, 2);
INSERT INTO CN1.BILL_DETAILS(BILL_ID, PRO_ID,
AMOUNT)
VALUES (500004, 400007, 1);
INSERT INTO CN1.BILL_DETAILS(BILL_ID, PRO_ID,
AMOUNT)
VALUES (500005, 400003, 3);
INSERT INTO CN1.BILL_DETAILS(BILL_ID, PRO_ID,
AMOUNT)
VALUES (500006, 400014, 1);

```

```

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 (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);
INSERT INTO CN2.BILL_DETAILS(BILL_ID, PRO_ID,
AMOUNT)
VALUES (700002, 400015, 2);
INSERT INTO CN2.BILL_DETAILS(BILL_ID, PRO_ID,
AMOUNT)
VALUES (700002, 400013, 1);
INSERT INTO CN2.BILL_DETAILS(BILL_ID, PRO_ID,
AMOUNT)
VALUES (700003, 400017, 10);
INSERT INTO CN2.BILL_DETAILS(BILL_ID, PRO_ID,
AMOUNT)
VALUES (700004, 400015, 2);
INSERT INTO CN2.BILL_DETAILS(BILL_ID, PRO_ID,
AMOUNT)
VALUES (700004, 400007, 1);
INSERT INTO CN2.BILL_DETAILS(BILL_ID, PRO_ID,
AMOUNT)
VALUES (700005, 400013, 3);
INSERT INTO CN2.BILL_DETAILS(BILL_ID, PRO_ID,
AMOUNT)
VALUES (700006, 400005, 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, PRODUCT, 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;
```

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;
```

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_quanlyk
ho 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_quanlyk
ho WM1
WHERE P1.PRO_ID = WM1.PRO_ID;
```

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.CUSTOMER@cn1_link_nhanvien C1,
CN1.BILL@cn1_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;
```

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';
```

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)))));
```

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, TENSP, SUM(SOLUONGBANDUOC)
AS TONGBANDUOC
FROM (SELECT P1.PRO_ID AS MASP,
P1.PRODUCT_NAME AS TENSP, 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 TENSP, 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, TENSP
ORDER BY TONGBANDUOC DESC
```

FETCH NEXT 5 ROWS ONLY;

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);
```

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;
```

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 TENSP, 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_giamdo
c 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);

```

*** Nguồn gốc:**

- Neo4j là hệ quản trị cơ sở dữ liệu đồ thị, được phát triển bởi Neo4j, Inc. Hiện nay, neo4j là một trong các cơ sở dữ liệu đồ thị mã nguồn mở phổ biến nhất.

- Neo4j có sẵn trong một GPL3-licensed mã nguồn mở “phiên bản cộng đồng”, với sao lưu trực tuyến và sẵn sàng cao mở rộng cấp phép theo giấy cấp phép thương mại mã nguồn đóng. Neo cũng cung cấp cho Neo4j với các tiện ích mở rộng này theo các điều khoản thương mại nguồn đóng

- Neo4j được triển khai bằng Java và có thể truy cập được từ phần mềm được viết bằng các ngôn ngữ khác bằng ngôn ngữ truy vấn Cypher thông qua điểm cuối HTTP giao dịch hoặc thông qua giao thức “bu lông” nhị phân.

- Neo4j là một cơ sở dữ liệu đồ thị, lưu trữ dữ liệu trong các node và các quan hệ của một đồ thị. Các cấu trúc dữ liệu chung nhất là đồ thị, biểu diễn bất kỳ kiểu dữ liệu nào, đảm bảo cấu trúc tự nhiên của một cơ sở dữ liệu. Neo4j không giống như cơ sở dữ liệu truyền thống, nó là NOSQL(Not Only SQL), nghĩa là không chỉ truy vấn dữ liệu bằng các câu sql thông thường, chúng ta truy vấn trong một cơ sở dữ liệu đồ thị dựa trên phép duyệt đồ thị.

- Neo4j một dự án mã nguồn mở dùng trong cộng đồng GPLv3, được hỗ trợ bởi công ty Neo Technology

*** Lịch sử**

- Năm 2000: Những người sáng lập Neo4j là Emil, Johan và Peter đã gặp phải các vấn đề về hiệu suất với RDBMS và bắt đầu xây dựng Neo4j prototype đầu tiên

- Năm 2002: Đã phát triển phiên bản Neo4j đầu tiên

- Năm 2003: Triển khai Neo4j sản xuất 24 x 7 đầu tiên

- Năm 2007: Thành lập một công ty có trụ sở tại Thụy Điển đứng sau Neo4j. Cũng mở nguồn cơ sở dữ liệu đồ thị đầu tiên, Neo4j, theo GPL

- Năm 2009: Tăng vốn tài trợ 2,5 triệu đô la, từ Sunstone và Conor và tiếp tục phát triển. 2000 khách hàng toàn cầu đầu tiên.

- Năm 2010: Phát hành phiên bản Neo4j 1.0 vào tháng 2

- Năm 2011: Tăng một vòng và chuyển trụ sở đến Thung lũng Silicon

- Năm 2012: Đã huy động được 11 triệu đô la Series B từ Fidelity, Sunstone và Conor GraphConnect, hội nghị đầu tiên về cơ sở dữ liệu đồ thị.

- Năm 2013: Neo4j phiên bản 2.0 được phát hành vào tháng 12.

- Năm 2015: Đã huy động được 20 triệu đô la Series C từ Creandum với Dawn và các nhà đầu tư hiện tại. Hơn 2 triệu lượt tải xuống Neo4j

- Năm 2016: Neo4j 3.0 đã phát hành. \$ 36 triệu Series D từ Greenbridge Investment

- Năm 2017: Máy tính để bàn Neo4j đã phát hành. Neo4j công bố Nền tảng đồ thị đầu tiên của ngành.

- Năm 2018: Neo4j Bloom đã phát hành. 80 triệu đô la Series E do Morgan Stanley Mở rộng Capital & One Peak Partners dẫn đầu.

- Năm 2019: Neo4j AuraDB ra mắt

- Năm 2020:

- Neo4j đã được công nhận là người dẫn đầu trong The Forrester Wave™ cho nền tảng dữ liệu đồ thị, Q4 2020.
- Bản phát hành GA của AuraDB Enterprise.
- Ra mắt Thư viện Khoa học Dữ liệu Đồ thị Neo4j
- Đồ thị hình thành 4 COVID-19
- Hội chợ triển lãm và hội nghị dành cho nhà phát triển trực tuyến Neo4j (NODES) đã đánh dấu sự kiện đồ thị kỹ thuật số lớn nhất của chúng tôi
- Ra mắt Graphs 4 COVID-19 “GraphHack”, một sáng kiến mới của Chương trình Graphs4Good để giúp chống lại sự lây lan của vi rút.

- Năm 2021:

- Đã huy động được 390 triệu đô la Series F do Eurazeo dẫn đầu với GV, Inovia Capital và các nhà đầu tư hiện tại; đây là khoản đầu tư lớn nhất trong lịch sử cơ sở dữ liệu
- Ra mắt Neo4j AuraDB Enterprise
- Neo4j phá vỡ rào cản quy mô với biểu đồ quan hệ nghìn tỷ
- Ra mắt Neo4j AuraDB miễn phí
- Năm 2022: Ra mắt Neo4j AuraDS (Khoa học dữ liệu đồ thị như một dịch vụ)

A. Cấp phép và phiên bản:

*** Cấp phép và phiên bản:**

- Neo4j có hai phiên bản: Cộng đồng và Doanh nghiệp. Nó được cấp phép kép: GPL v3 và giấy phép thương mại. Phiên bản Cộng đồng miễn phí nhưng chỉ giới hạn chạy trên một nút do thiếu phân cụm và không có bản sao lưu nóng.

- Phiên bản doanh nghiệp mở ra những hạn chế này, cho phép phân cụm, sao lưu nóng và giám sát. Phiên bản doanh nghiệp có sẵn theo giấy phép Thương mại nguồn đóng.

*** Đặc điểm:**

- Neo4j cung cấp công nghệ đồ thị đã được kiểm tra hiệu suất và mở rộng quy mô.

Kiến trúc cụm phân tán hiệu suất cao của Neo4j cho phép khách hàng chạy khối lượng công việc khoa học dữ liệu và OLTP thách thức nhất, đồng thời duy trì tính tuân thủ ACID và tính toàn vẹn của dữ liệu.

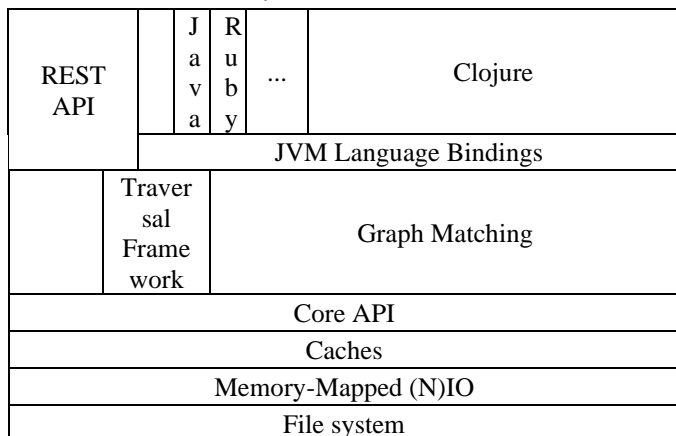
- Neo4j có thể xử lý các đồ thị với hàng tỉ các node/ các mối quan hệ/ các thuộc tính trên 1 máy tính và có thể mở rộng ra trên nhiều máy khác.

- Với Neo4j, khách hàng có thể tự do lựa chọn triển khai trên nền tảng tự lưu trữ, kết hợp hoặc đa đám mây

Một framework duyệt mạnh mẽ với tốc độ duyệt trên các node cực nhanh trong không gian node. Độ sâu của quá trình duyệt có thể lên đến 1000 mức và dưới tốc độ 1 giây

- Full transactional như một cơ sở dữ liệu thực sự, với đầy đủ các đặc tính ACID (Atomicity, Consistency, Isolation, Durability).

B. Kiến trúc của Neo4j



Hình A.5. Kiến trúc logic của Neo4j

- File system: là các file trên ổ cứng, được lưu trữ cẩn thận để tính toán các offset và tìm đến bất kỳ một record nào trong các file một cách nhanh nhất. Ta lưu trữ tách biệt các node, các quan hệ và các thuộc tính, và tối ưu trong những trường hợp chung nhập để đảm bảo dữ liệu được tìm thấy trên 1 file.

- Memory-mapped (N) IO: ta sử dụng java IO cho mục đích nhanh chóng.

- Caches: cho phép làm việc nhanh chóng trên các đĩa quay, cho phép chúng ta duyệt hàng triệu phép duyệt mỗi giây trên một phần cứng của máy laptop.

- Core API: là phần nhân của Neo4j, lưu trữ các cấu trúc mức trừu tượng của đồ thị, mang tính hiệu quả cao.

- Traversal Framework: tầng truy vấn dữ liệu.

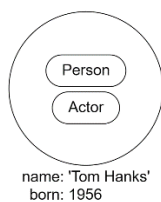
- JVM Language Bindings: các thành phần của Java API như Jruby, Jython, Scala,...

C. Node

- Node mô tả những thực thể (các đối tượng rời rạc) của một miền.

- Node có thể có 0 hoặc nhiều label để định nghĩa (phân loại) đó là loại node nào.

- Đồ thị đơn giản nhất là đồ thị chỉ chứa một node và không có relationship. Ta xem xét đồ thị chỉ chứa duy nhất một node dưới đây:



Hình A.6. Ví dụ một node

- Các node label bao gồm:

- Person
- Actor

- Các thuộc tính bao gồm:

- name: Tom Hanks
- born: 1956

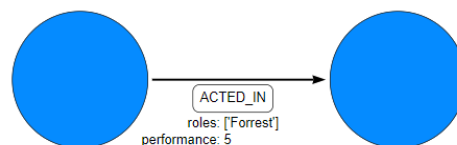
D. Relationship

- Một relationship mô tả cách kết nối giữa node nguồn và node đích với nhau. Ngoài ra, một node cũng có thể có một relationship tới chính nó.

- Đặc điểm của relationship:

- Kết nối node nguồn và node đích.
- Có hướng (một chiều)
- Phải có type (một type) để định nghĩa (phân loại) đó là loại relationship nào.
- Có thể có các thuộc tính (cặp key-value) mô tả thêm về relationship.

- Relationship sắp xếp các node theo những cấu trúc, cho phép một đồ thị trở nên giống như một danh sách, một cây, một map, hoặc một thực thể phức hợp – bất cứ thứ gì trong số đó có thể được kết hợp thành các cấu trúc phức tạp hơn, được kết nối phong phú hơn.



Hình A.7.1. Ví dụ relationship giữa 2 node

- Relationship type: ACTED_IN
- Thuộc tính:
 - roles: ['Forrest']
 - performance: 5

- Thuộc tính roles có một mảng giá trị với item ('Forrest') duy nhất.

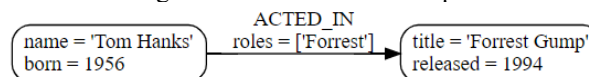
- Relationship luôn luôn có hướng. Tuy nhiên, hướng có thể được bỏ qua nếu nó không cần thiết. Điều này có nghĩa là ta không nhất thiết phải thêm các relationship trùng lặp theo hướng ngược lại trừ khi cần phải mô tả data model một cách chính xác.

- Một node có thể có relationship tới chính nó. Giả sử ta muốn thể hiện rằng Tom Hanks KNOWS chính bản thân mình, ta có thể vẽ relationship như sau:



Hình A.7.2. Ví dụ relationship của 1 node tới chính nó

- Một relationship cần phải có đúng một relationship type. Dưới đây là ví dụ về relationship ACTED_IN, có node Tom Hanks là node nguồn và node Forrest Gump là node đích:



Hình A.7.3. Ví dụ relationship type

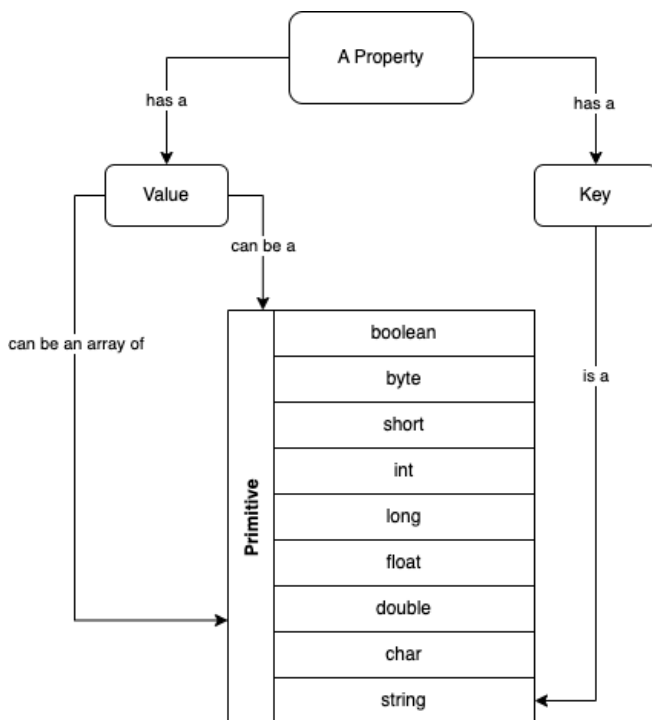
- Quan sát và biết được rằng node Tom Hanks có một outgoing relationship, trong khi đó node Forrest Gump có một incoming relationship.

E. Properties

- Cả node và relationships đều có các thuộc tính (properties).
 - Properties là cặp key – value (khóa – giá trị) trong đó key có kiểu String (kiểu mô tả này tương tự như kiểu dữ liệu Map trong Java). Các giá trị của property có thể là 1 kiểu nguyên thủy, hoặc một mảng của nhiều kiểu nguyên thủy.

Ví dụ: kiểu String, kiểu int, hoặc mảng int[].

- Properties không chứa giá trị null, nếu một thuộc tính có giá trị = null, nghĩa là không tồn tại key đó trong các properties của node hoặc relationships.



II. VIẾT HÀM, THỦ TỤC VÀ RÀNG BUỘC TOÀN VỆN

A. Function

Cypher là ngôn ngữ truy vấn trong Neo4j. Các câu query là tập hợp các mệnh đề được liên kết với nhau.

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;
```

B. 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;

```

C. 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:

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

$C.CUS_ID = B.CUS_ID \wedge B.BILL_DATE \geq REG_DATE$
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: NGÀY MUA HÀNG CỦA KHÁCH HÀNG
KHÔNG HỢP LÉ');

            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: NGAY DANG KY KHONG HOP LE');
END IF;
END LOOP;
CLOSE CURSOR_BILL_DATE;
DBMS_OUTPUT.PUT_LINE('THANH CONG');
END;

```

III. DEMO CÁC MỨC CÔ LẬP (ISOLATION LEVEL) TRONG MÔI TRƯỜNG PHẦN TẤN

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

Ti me	Transacti on T1(Employee A) - CN2	T1 Outp ut	Transaction T1(Employee B) - CN1	T2 Outp ut
t0	SELECT ADDRES S FROM CN2.CUS TOMER WHERE CUS_ID = 300001;	117/2 Nguy en Trai, Q5, TpH CM	SELECT ADDRESS FROM CN2.CUSTOME R@cn2_link WHERE CUS_ID = 300001;	117/2 Nguy en Trai, Q5, TpH CM
t1	UPDATE CN2.CUS TOMER SET Address	1 row updat ed.		

	= '731 Tran Hung Dao,Q5, TpHCM' WHERE CUS_ID = 300001;			
t2			UPDATE CN2.CUSTOME R@cn2_link SET Address = '200/11 Nguyen Van Cu, Q5, TpHCM' WHERE CUS_ID = 300001;	
t3	COMMIT ;	Com mit comp lete.		1 row updat ed.
t4			COMMIT;	Com mit comp lete.
t5	SELECT ADDRES S FROM CN2.CUS TOMER WHERE CUS_ID = 300001;	200/1 1 Nguy en Van Cu, Q5, TpH CM	SELECT ADDRESS FROM CN2.CUSTOME R@cn2_link WHERE CUS_ID = 300001;	200/1 1 Nguy en Van Cu, Q5, TpH CM

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

đư 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
t0	SET SERVEROUTPUT ON; BEGIN CN2.PRODUCT_INFO (400001); END; /	Thông tin sản phẩm có mã: 400001 MASP = 400001, NAME =Book, SALE		

		PRICE = 8000 PL/SQL procedure successfully completed.		
t1			BEGIN UPDATE_PRICE@ (10000, 400001); END; /	PL/SQL UPDATE_PRICE successfully completed.
t2			COMMIT;	Commit completed.
t3	BEGIN CN2.PRODUCT_INFO (400021); END; /	Thông tin sản phẩm có mã: 400001 MASP = 400001, NAME =Book, SALE PRICE		

		E = 18000 PL/S QL proce dure succes sfully compl eted.		
--	--	---	--	--

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

T i m e	Transact ion T1(Empl oyeee A)	T1 Output	Transaction T1(Director B)	T2 Out put
t0	SET SERVER OUTPUT ON; BEGIN CN2.PR ODUCT_ INFO (400009);	Thông tin sản phẩm có mã: 400009 MASP = 400009, NAME =terius,		

	END; / SALE PRICE = 5000 PL/SQL procedure successfully completed.			
t1			BEGIN CN2.DEL_PR O@cn2_link(4 00009); END; / PL/ SQ L pro ced ure suc cess fu lly co mpl eted .	
t2			COMMIT;	Co mm it co mpl ete.
t3	BEGIN CN2.PR ODUCT_ INFO (400009- END; / Error report - ORA-20008: Ma san pham khong lop le ORA-06512: at "C##USSER1 .PRODUCT_I NFO", 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

Ti m e	Transaction T1(Employee e A)	T1 Out put	Transaction T1(Employee B)	T2 Ou tpu t
t0	UPDATE CN2.WARE HOUSE_SAL ES SET STATUS = 'Hết hàng' WHERE PRO_ID = 400001;	1 row upd ated .		
t1			UPDATE CN2.WAREHOUS E_SALES@cn2_li nk SET STATUS = 'Hết hàng' WHERE PRO_ID	1 row upd ate d.

			= 400002;	
t2	UPDATE CN2.WARE HOUSE_SAL ES SET STATUS = 'Cho nhập hang' WHERE PRO_ID = 400002;			
t3			UPDATE CN2.WAREHOUS E_SALES@cn2_li nk SET STATUS = 'Cho nhập hang' WHERE PRO_ID = 400001;	
t4		ER RO R at line 3: OR A- 000 60: dea dloc k dete cted whi le wait ing for reso urc		

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**.

IV. THỰC HIỆN TỐI ƯU HÓA TRUY VẤN TRÊN MÔI TRƯỜNG PHÂN TÁN 1 CÂU TRUY VẤN ĐƠN GIẢN

A. 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 nhánh 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;
```

B. EXPLAIN QUERY câu truy vấn đơn giản

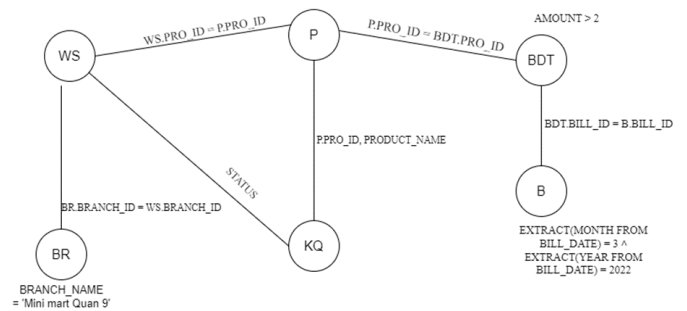
```
SELECT /*+ GATHER_PLAN_STATISTICS */
DISTINCT P.PRO_ID,      PRODUCT_NAME,
STATUS
FROM      PRODUCT P, WAREHOUSE_SALES
WS, BILL B,
BILL_DETAILS BDT, BRANCH BR
WHERE      BR.BRANCH_ID = WS.BRANCH_ID
AND
WS.PRO_ID = P.PRO_ID AND
P.PRO_ID = BDT.PRO_ID AND
BDT.BILL_ID = B.BILL_ID AND
BR.BRANCH_NAME = '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=>'ALLS
TATS LAST'));
```

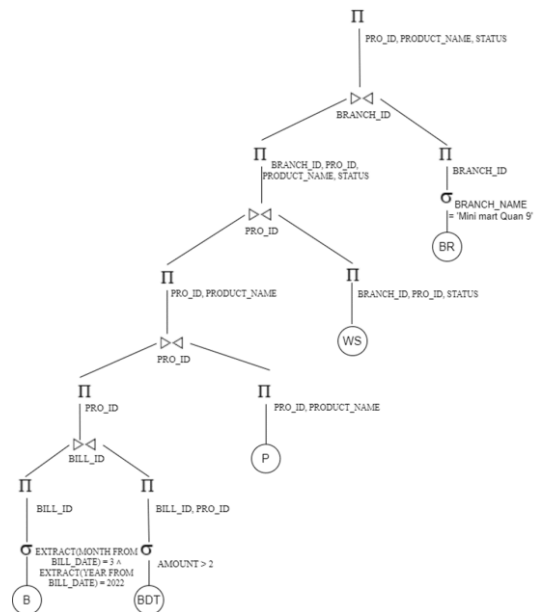
Kết quả

Id	Operation	Name	Starts	E-Rows	A-Rows	A-Time	Buffers	Optim	Plan	Mem	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				

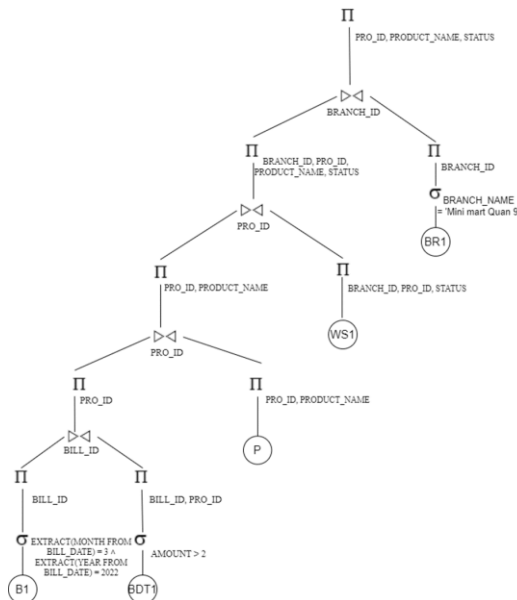
C. Tối ưu hóa câu truy vấn cục bộ, phân tán



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

D. 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=>'ALLS
TATS LAST'));
```

Kết quả

	Plan	Operation	Name	Starts	Ends	A-Rows	A-Time	Buffers	IO	Temp	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			

LỜI CẢM ƠN

Lời đầu tiên, nhóm xin cảm ơn thầy Nguyễn Minh Nhựt đã cung cấp kiến thức để chúng tôi có thể thực hiện bài tập này cũng như những lời khuyên nhiệt tình, chân thật và luôn hữu ích của thầy. Kịp thời trả lời các câu hỏi của chúng tôi. Nếu không có sự hướng dẫn của anh Nhựt, chúng tôi nghĩ phần báo cáo rất khó hoàn thành. Đây cũng là cơ hội để mỗi thành viên trong nhóm làm việc với những người bạn mới, học hỏi thêm kỹ năng làm việc nhóm, học hỏi lẫn nhau và quan trọng là có cơ hội thực hiện sản phẩm thông qua khóa học.

Trong quá trình thực hiện dự án, nhóm áp dụng những điều đã học được đồng thời áp dụng những điều mới với mong muốn hoàn thành công việc một cách hoàn hảo nhất. Nhưng thời gian, kiến thức và kinh nghiệm còn hạn chế, không tránh khỏi những thiếu sót, chúng tôi rất mong nhận được sự góp ý quý báu của các thầy cô, các anh chị đi trước để nhóm bổ sung và hoàn thiện kiến thức, phục vụ tốt hơn cho đồ án và thực tiễn sau này công việc.

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 (ĐHQT - TPHCM), Oct. 30, 2021. [Online]. Available: https://courses.uit.edu.vn/pluginfile.php/313931/mod_resource/content/1/10_KiemTra3.pdf