

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**  
**KHOA HỆ THỐNG THÔNG TIN**



**BÀI TẬP LỚN 1**  
**MÔN HỌC CƠ SỞ DỮ LIỆU PHÂN TÁN**  
**ĐỀ TÀI:**  
**THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN**

Giảng viên hướng dẫn: **Nguyễn Minh Nhựt**

Lớp: **IS211.M11.HTCL**

Nhóm sinh viên thực hiện: **Peekaboo Team**

STT	Họ và Tên	MSSV
1	Tạ Quang Huy	19520113
2	Lê Thị Hồng Cúc	19521295
3	Chu Xuân Sơn	19522130
4	Nguyễn Ảnh Trường Thắng	19522204

TP. Hồ Chí Minh, tháng 9 năm 2021

## MỤC LỤC

YÊU CẦU 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.....	7
1.3.1. DLL.....	7
1.3.2. DML.....	12
1.4. KIẾN TRÚC PHÂN QUYỀN .....	25
1.4.1. Mô tả kiến trúc phân quyền .....	25
1.4.2. Minh họa.....	26
1.5. CÁC CÂU TRUY VẤN .....	28
YÊU CẦU 2: VIẾT HÀM, THỦ TỤC VÀ RÀNG BUỘC TOÀN VỆ.....	35
2.1. FUNCTION .....	35
2.1. PROCEDURE .....	36
Procedure thực hiện giao dịch chuyển tiền trong ngân hàng .....	36
Procedure in ra thông tin khách hàng và số lần thực hiện giao dịch trong tháng. ....	39
Procedure xóa toàn bộ thông tin của khách hàng.....	42
2.3. RÀNG BUỘC TOÀN VỆ.....	44

## THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

YÊU CẦU 3: DEMO CÁC MỨC CÔ LẬP (ISOLATION LEVEL) TRONG MÔI TRƯỜNG PHÂN TÁN.....	46
3.1. TRƯỜNG HỢP LOST UPDATE.....	46
3.2. TRƯỜNG HỢP DIRTY READ.....	47
3.3. TRƯỜNG HỢP UNREPEATABLE READ .....	47
3.4. TRƯỜNG HỢP PHANTOM READ .....	50
3.5. TRƯỜNG HỢP DEADLOCK .....	52
YÊU CẦU 4: THỰC HIỆN TỐI ƯU HÓA TRUY VẤN TRÊN MÔI TRƯỜNG PHÂN TÁN .....	54
4.1. Câu truy vấn đơn giản chưa tối ưu .....	54
4.2. EXPLAIN QUERY câu truy vấn đơn giản.....	55
4.3. Tối ưu hóa câu truy vấn cục bộ, phân tán .....	56
4.4. Viết lại câu query trên môi trường phân tán .....	61
TÀI LIỆU THAM KHẢO .....	62

## YÊU CẦU 1: THIẾT KẾ CSDL PHÂN TÁN

### 1.1. THIẾT KẾ VÀ MÔ TẢ CSDL PHÂN TÁN

#### CSDL Quản lý các chi nhánh ngân hàng ACB

Ngân hàng ACB có 36 chi nhánh lớn nhỏ tại Thành Phố Hồ Chí Minh. Trụ sở chính của Ngân hàng ACB nằm tại chi nhánh 'Nguyễn Thị Minh Khai – Quận 3'

Lược đồ cơ sở dữ liệu của mỗi chi nhánh của ngân hàng ACB như sau:

**BRANCH**(BRANCH\_ID, BRANCH\_NAME, PHONE\_NUMBER)

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 số điện thoại của chi nhánh đó (PHONE\_NUMBER). Trường tên chi nhánh, số điện thoại là duy nhất.

**CUSTOMER\_BD** (CUSTOMER\_ID, FIRST\_NAME, LAST\_NAME, ID\_CARD, BRANCH\_ID)

Tên từ: Mỗi khách hàng có mã khách hàng (CUSTOMER\_ID) là duy nhất, họ và tên khách hàng (FIRST\_NAME, LAST\_NAME), mã chi nhánh mà khách hàng đó đăng kí tài khoản (BRANCH\_ID), số CMND, CCCD của khách hàng (ID\_CARD).

**CUSTOMER\_CSD**(CUSTOMER\_ID, FIRST\_NAME, LAST\_NAME, GENDER, ADDRESS, DATE\_OF\_BIRTH, PHONE\_NUMBER)

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 (DATE\_OF\_BIRTH), địa chỉ khách hàng (ADDRESS).

**EMPLOYEE**(EMPLOYEE\_ID, EMPLOYEE\_NAME, JOB, START\_DATE, SALARY, BRANCH\_ID)

Tên từ: Mỗi nhân viên có mã nhân viên (EMPLOYEE\_ID) là duy nhất, tên nhân viên (EMPLOYEE\_NAME), 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).

## THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

### **ACCOUNT\_TYPE**(ACCOUNT\_TYPE\_ID, ACCOUNT\_TYPE\_NAME, INTEREST\_RATE)

Tên từ: Mỗi loại tài khoản có mã loại tài khoản (ACCOUNT\_TYPE\_ID) là duy nhất, tên loại tài khoản (ACCOUNT\_TYPE), lãi suất của loại tài khoản đó (INTEREST\_RATE). Có 2 loại tài khoản chính: *tài khoản thanh toán* và *tài khoản tiết kiệm*. Tài khoản thanh toán có mã loại tài khoản bắt đầu bằng ký tự 'PA', tài khoản tiết kiệm chứa ký tự 'SA'. Tài khoản tiết kiệm bao gồm tiết kiệm có kỳ hạn ('TSA') và không kỳ hạn ('NSA').

### **ACCOUNT**(ACCOUNT\_ID, ACCOUNT\_TYPE\_ID, CURRENT\_BALANCE, OPEN\_DAY, MATURITY\_DATE, ANTICIPATED\_INTEREST, STATUS, CUSTOMER\_ID)

Tên từ: Mỗi tài khoản có số tài khoản (ACCOUNT\_ID) là duy nhất, mã loại tài khoản (ACCOUNT\_TYPE\_ID), số dư hiện tại (CURRENT\_BALANCE), ngày mở tài khoản (OPEN\_DAY), trạng thái (STATUS), chủ tài khoản (CUSTOMER\_ID). Tài khoản có 2 loại trạng thái 'Đang hoạt động' và 'Khoa'. Nếu loại tài khoản là tài khoản tiết kiệm thì sẽ có thêm ngày đáo hạn (MATURITY\_DATE) và tiền lãi sự kiến (ANTICIPATED\_INTEREST).

### **TRANSACTION\_TYPE**(TRANSACTION\_TYPE\_ID, TRANSACTION\_TYPE\_NAME, FEE)

Tên từ: Mỗi loại giao dịch có mã loại giao dịch (TRANSACTION\_TYPE\_ID) là duy nhất, tên loại giao dịch (TRANSACTION\_TYPE\_NAME), phí thực hiện giao dịch đó (FEE).

### **TRANSACTION**(TRANSACTION\_ID, TRANSACTION\_TYPE\_ID, TRANSACTION\_DATE, TOTAL\_TRANSACTION\_AMOUNT, ACCOUNT\_ID)

Tên từ: Mỗi giao dịch có mã giao dịch (TRANSACTION\_ID) là duy nhất, mã loại giao dịch (TRANSACTION\_TYPE\_ID), ngày thực hiện giao dịch (TRANSACTION\_DATE), tổng số tiền thực hiện giao dịch (TOTAL\_TRANSACTION\_AMOUNT), tài khoản thực hiện giao dịch đó (ACCOUNT\_ID).

### **BANK**(BANK\_ID, BANK\_NAME)

Tên từ: Mỗi ngân hàng có mã ngân hàng (BANK\_ID) là duy nhất, tên ngân hàng, (BANK\_NAME).

## THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

**TRANSFER\_DETAIL**(TRANSFER\_DETAIL\_ID, *SENDER\_ACCOUNT*,  
*RECEIVER\_ACCOUNT*, *BANK\_ID*, AMOUNT, CONTENT, *TRANSACTION\_ID*)

Tên từ: Mỗi chi tiết chuyển tiền có mã chi tiết chuyển tiền (TRANSFER\_DETAIL\_ID) là duy nhất, mã tài khoản người gửi (SENDER\_ACCOUNT), mã tài khoản người nhận (RECEIVER\_ACCOUNT), ngân hàng người nhận (BANK\_ID), số tiền đã chuyển (AMOUNT), nội dung chuyển tiền (CONTENT) và mã giao dịch của chi tiết chuyển tiền đó (TRANSACTION\_ID).

### 1.2. KIẾN TRÚC PHÂN MẢNH

#### 1.2.1. Mô tả phân mảnh

- Quan hệ BRANCH là phân mảnh ngang chính.
- Quan hệ EMPLOYEE, ACCOUNT, TRANSACTION, TRANSFER\_DETAIL là phân mảnh ngang dẫn xuất
- Quan hệ CUSTOMER được phân mảnh hỗn hợp thành CUSTOMER\_BD và CUSTOMER\_CSD. Trong đó, quan hệ CUSTOMER\_BD chứa các thông tin dùng để thực hiện giao dịch, quan hệ này do phòng ban Kinh doanh của mỗi chi nhánh quản lý; quan hệ CUSTOMER\_CSD chứa các thông tin cá nhân khác của khách hàng, quan hệ này do phòng ban Chăm sóc khách hàng của mỗi chi nhánh quản lý.
- Quan hệ ACCOUNT\_TYPE, TRANSACTION\_TYPE, BANK được nhân bản tại tất cả chi nhánh.

#### 1.2.2. Minh họa trên 2 chi nhánh

❖ Quan hệ **BRANCH** phân mảnh ngang chính theo tên chi nhánh

CN1.BRANCH =  $\delta_{(TenChiNhanh='CN Tien Giang')}$  BRANCH

CN2.BRANCH =  $\delta_{(TenChiNhanh='CN TP.Bien Hoa')}$  BRANCH

## THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

- ❖ Quan hệ **EMPLOYEE, ACCOUNT, TRANSACTION, TRANSFER\_DETAIL** phân mảnh ngang dẫn xuất như sau

$CN1.EMPLOYEE = EMPLOYEE \bowtie_{BRANCH\_ID} CN1.BRANCH$

$CN2.EMPLOYEE = EMPLOYEE \bowtie_{BRANCH\_ID} CN2.BRANCH$

$CN1.ACCOUNT = ACCOUNT \bowtie_{CUSTOMER\_ID} CN1.CUSTOMER\_BD$

$CN2.ACCOUNT = ACCOUNT \bowtie_{CUSTOMER\_ID} CN2.CUSTOMER\_BD$

$CN1.TRANSACTION = TRANSACTION \bowtie_{ACCOUNT\_ID} CN1.ACCOUNT$

$CN2.TRANSACTION = TRANSACTION \bowtie_{ACCOUNT\_ID} CN2.ACCOUNT$

$CN1.TRANSFER\_DETAIL = TRANSFER\_DETAIL \bowtie_{ACCOUNT\_ID} CN1.TRANSACTION$

$CN2.TRANSFER\_DETAIL = TRANSFER\_DETAIL \bowtie_{ACCOUNT\_ID} CN2.TRANSACTION$

- ❖ Quan hệ **CUSTOMER** được phân mảnh hỗn hợp như sau

$CN1.CUSTOMER\_BD = \pi_{(CUSTOMER\_ID, FIRST\_NAME, LAST\_NAME, ID\_CARD, BRANCH\_ID)}(CUSTOMER \bowtie_{BRANCH\_ID} CN1.BRANCH)$

$CN1.CUSTOMER\_CSD = \pi_{(CUSTOMER\_ID, FIRST\_NAME, LAST\_NAME, GENDER, ADDRESS, DATE\_OF\_BIRTH, PHONE\_NUMBER)}(CUSTOMER \bowtie_{BRANCH\_ID} CN1.BRANCH)$

$CN2.CUSTOMER\_BD = \pi_{(CUSTOMER\_ID, FIRST\_NAME, LAST\_NAME, ID\_CARD, BRANCH\_ID)}(CUSTOMER \bowtie_{BRANCH\_ID} CN2.BRANCH)$

$CN2.CUSTOMER\_CSD = \pi_{(CUSTOMER\_ID, FIRST\_NAME, LAST\_NAME, GENDER, ADDRESS, DATE\_OF\_BIRTH, PHONE\_NUMBER)}(CUSTOMER \bowtie_{BRANCH\_ID} CN2.BRANCH)$

- ❖ Quan hệ **ACCOUNT\_TYPE, TRANSACTION\_TYPE, BANK** được nhân bản tại tất cả chi nhánh.

# THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

## 1.3. XÂY DỰNG CSDL PHÂN TÁN

### 1.3.1. DDL

Thực hiện tạo bảng, tạo khóa chính, khóa ngoại và tạo sequences cho 2 User CN1 và CN2:

<b>ALTER SESSION SET NLS_DATE_FORMAT = 'DD/MM/YY HH24:MI:SS';</b>	
<b>Chi nhánh 1</b>	<b>Chi nhánh 2</b>
<b>CREATE TABLE</b> CN1.BRANCH ( BRANCH_ID VARCHAR2(10) <b>CONSTRAINT</b> BRANCH_ID_PK <b>PRIMARY KEY</b> , BRANCH_NAME VARCHAR2(50) NOT NULL, PHONE_NUMBER VARCHAR2(11) NOT NULL );	<b>CREATE TABLE</b> CN2.BRANCH ( BRANCH_ID VARCHAR2(10) <b>CONSTRAINT</b> BRANCH_ID_PK <b>PRIMARY KEY</b> , BRANCH_NAME VARCHAR2(50) NOT NULL, PHONE_NUMBER VARCHAR2(11) NOT NULL );
<b>CREATE TABLE</b> CN1.CUSTOMER_BD ( CUSTOMER_ID NUMBER(10) <b>CONSTRAINT</b> CUSTOMER_BD_ID_PK <b>PRIMARY KEY</b> , FIRST_NAME VARCHAR2(50) NOT NULL, LAST_NAME VARCHAR2(50) NOT NULL, ID_CARD NUMBER(20) NOT NULL, BRANCH_ID VARCHAR2(10), <b>CONSTRAINT</b> CUSTOMER_BRANCH_ID_FK FOREIGN KEY (BRANCH_ID) <b>REFERENCES</b> CN1.BRANCH (BRANCH_ID) );	<b>CREATE TABLE</b> CN2.CUSTOMER_BD ( CUSTOMER_ID NUMBER(10) <b>CONSTRAINT</b> CUSTOMER_BD_ID_PK <b>PRIMARY KEY</b> , FIRST_NAME VARCHAR2(50) NOT NULL, LAST_NAME VARCHAR2(50) NOT NULL, ID_CARD NUMBER(20) NOT NULL, BRANCH_ID VARCHAR2(10), <b>CONSTRAINT</b> CUSTOMER_BRANCH_ID_FK FOREIGN KEY (BRANCH_ID) <b>REFERENCES</b> CN2.BRANCH (BRANCH_ID) );
<b>CREATE TABLE</b> CN1.CUSTOMER_CSD ( CUSTOMER_ID NUMBER(10) <b>CONSTRAINT</b> CUSTOMER_CSD_PK <b>PRIMARY KEY</b> , FIRST_NAME VARCHAR2(50) NOT NULL, LAST_NAME VARCHAR2(50) NOT NULL, GENDER VARCHAR2(6) CHECK (GENDER IN('Nam', 'Nu')), DATE_OF_BIRTH DATE NOT NULL, );	<b>CREATE TABLE</b> CN2.CUSTOMER_CSD ( CUSTOMER_ID NUMBER(10) <b>CONSTRAINT</b> CUSTOMER_CSD_PK <b>PRIMARY KEY</b> , FIRST_NAME VARCHAR2(50) NOT NULL, LAST_NAME VARCHAR2(50) NOT NULL, GENDER VARCHAR2(6) CHECK (GENDER IN('Nam', 'Nu')), DATE_OF_BIRTH DATE NOT NULL, );



## THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

<pre> ADDRESS VARCHAR2(400) NOT NULL, PHONE_NUMBER VARCHAR2(11) NOT NULL, <b>CONSTRAINT</b> CUSTOMERCSD_CUSTOMER_ID_FK <b>FOREIGN</b> <b>KEY</b>(CUSTOMER_ID) <b>REFERENCES</b> CN1.CUSTOMER_BD(CUSTOMER_ID) ); </pre>	<pre> ADDRESS VARCHAR2(400) NOT NULL, PHONE_NUMBER VARCHAR2(11) NOT NULL, <b>CONSTRAINT</b> CUSTOMERCSD_CUSTOMER_ID_FK <b>FOREIGN</b> <b>KEY</b>(CUSTOMER_ID) <b>REFERENCES</b> CN2.CUSTOMER_BD(CUSTOMER_ID) ); </pre>
<pre> <b>CREATE TABLE</b> CN1.EMPLOYEE (     EMPLOYEE_ID NUMBER(7, 0) <b>CONSTRAINT</b> EMPLOYEE_ID_PK <b>PRIMARY</b> <b>KEY</b>,     EMPLOYEE_NAME VARCHAR2(50) NOT NULL,     START_DAY DATE NOT NULL,     JOB VARCHAR2(50) NOT NULL,     SALARY NUMBER(12),     BRANCH_ID VARCHAR2(10),     <b>CONSTRAINT</b> EMPLOYEE_BRANCH_ID_FK <b>FOREIGN KEY</b>(BRANCH_ID) <b>REFERENCES</b> CN1.BRANCH(BRANCH_ID) ); </pre>	<pre> <b>CREATE TABLE</b> CN2.EMPLOYEE (     EMPLOYEE_ID NUMBER(7, 0) <b>CONSTRAINT</b> EMPLOYEE_ID_PK <b>PRIMARY</b> <b>KEY</b>,     EMPLOYEE_NAME VARCHAR2(50) NOT NULL,     START_DAY DATE NOT NULL,     JOB VARCHAR2(50) NOT NULL,     SALARY NUMBER(12),     BRANCH_ID VARCHAR2(10),     <b>CONSTRAINT</b> EMPLOYEE_BRANCH_ID_FK <b>FOREIGN KEY</b>(BRANCH_ID) <b>REFERENCES</b> CN2.BRANCH(BRANCH_ID) ); </pre>
<pre> <b>CREATE TABLE</b> CN1.ACCOUNT_TYPE (     ACCOUNT_TYPE_ID CHAR(5) <b>CONSTRAINT</b> ACCOUNTTYPE_ID_PK <b>PRIMARY</b> <b>KEY</b>,     ACCOUNT_TYPE_NAME VARCHAR2(200) NOT NULL,     INTEREST_RATE NUMBER(4, 4) ); </pre>	<pre> <b>CREATE TABLE</b> CN2.ACCOUNT_TYPE (     ACCOUNT_TYPE_ID CHAR(5) <b>CONSTRAINT</b> ACCOUNTTYPE_ID_PK <b>PRIMARY</b> <b>KEY</b>,     ACCOUNT_TYPE_NAME VARCHAR2(200) NOT NULL,     INTEREST_RATE NUMBER(4, 4) ); </pre>
<pre> <b>CREATE TABLE</b> CN1.ACCOUNT (     ACCOUNT_ID NUMBER(7) <b>CONSTRAINT</b> ACCOUNT_ID_PK <b>PRIMARY KEY</b>,     ACCOUNT_TYPE_ID CHAR(5) NOT NULL, </pre>	<pre> <b>CREATE TABLE</b> CN2.ACCOUNT (     ACCOUNT_ID NUMBER(7) <b>CONSTRAINT</b> ACCOUNT_ID_PK <b>PRIMARY KEY</b>,     ACCOUNT_TYPE_ID CHAR(5) NOT NULL, </pre>

## THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

<pre> CURRENT_BALANCE NUMBER(12) NOT NULL, OPEN_DAY DATE NOT NULL, MATURITY_DATE DATE, ANTICIPATED_INTEREST NUMBER(12), STATUS VARCHAR2(20) CHECK (STATUS IN ('Hoat dong', 'Khoa')), CUSTOMER_ID NUMBER(10) NOT NULL, <b>CONSTRAINT</b> ACCOUNT_CUSTOMERID_FK <b>FOREIGN KEY</b> (CUSTOMER_ID) <b>REFERENCES</b> CN1.CUSTOMER_BD (CUSTOMER_ID), <b>CONSTRAINT</b> ACCOUNT_ACCOUNTTYPEID_FK <b>FOREIGN</b> <b>KEY</b> (ACCOUNT_TYPE_ID) <b>REFERENCES</b> CN1.ACCOUNT_TYPE (ACCOUNT_TYPE_ID) ); </pre>	<pre> CURRENT_BALANCE NUMBER(12) NOT NULL, OPEN_DAY DATE NOT NULL, MATURITY_DATE DATE, ANTICIPATED_INTEREST NUMBER(12), STATUS VARCHAR2(20) CHECK (STATUS IN ('Hoat dong', 'Khoa')), CUSTOMER_ID NUMBER(10) NOT NULL, <b>CONSTRAINT</b> ACCOUNT_CUSTOMERID_FK <b>FOREIGN KEY</b> (CUSTOMER_ID) <b>REFERENCES</b> CN2.CUSTOMER_BD (CUSTOMER_ID), <b>CONSTRAINT</b> ACCOUNT_ACCOUNTTYPEID_FK <b>FOREIGN</b> <b>KEY</b> (ACCOUNT_TYPE_ID) <b>REFERENCES</b> CN2.ACCOUNT_TYPE (ACCOUNT_TYPE_ID) ); </pre>
<pre> <b>CREATE TABLE</b> CN1.TRANSACTION_TYPE ( TRANSACTION_TYPE_ID CHAR(4) <b>CONSTRAINT</b> TRANSACTIONTYPE_ID_PK <b>PRIMARY KEY</b>, NAME VARCHAR2(50) NOT NULL, FEE NUMBER(12) ); </pre>	<pre> <b>CREATE TABLE</b> CN2.TRANSACTION_TYPE ( TRANSACTION_TYPE_ID CHAR(4) <b>CONSTRAINT</b> TRANSACTIONTYPE_ID_PK <b>PRIMARY KEY</b>, NAME VARCHAR2(50) NOT NULL, FEE NUMBER(12) ); </pre>
<pre> <b>CREATE TABLE</b> CN1.TRANSACTION ( TRANSACTION_ID NUMBER(7) <b>CONSTRAINT</b> TRANSACTION_ID_PK <b>PRIMARY</b> <b>KEY</b>, TRANSACTION_TYPE_ID CHAR(4) NOT NULL, TRANSACTION_DATE DATE NOT NULL, TOTAL_TRANSACTION_AMOUNT NUMBER(12) NOT NULL, ACCOUNT_ID NUMBER(7) NOT NULL, <b>CONSTRAINT</b> TRANSACTION_TRANSACTIONTYPEID_FK </pre>	<pre> <b>CREATE TABLE</b> CN2.TRANSACTION ( TRANSACTION_ID NUMBER(7) <b>CONSTRAINT</b> TRANSACTION_ID_PK <b>PRIMARY</b> <b>KEY</b>, TRANSACTION_TYPE_ID CHAR(4) NOT NULL, TRANSACTION_DATE DATE NOT NULL, TOTAL_TRANSACTION_AMOUNT NUMBER(12) NOT NULL, ACCOUNT_ID NUMBER(7) NOT NULL, <b>CONSTRAINT</b> TRANSACTION_TRANSACTIONTYPEID_FK </pre>

## THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

<b>FOREIGN KEY</b> (TRANSACTION_TYPE_ID) <b>REFERENCES</b> CN1.TRANSACTION_TYPE (TRANSACTION_TYPE_ID), <b>CONSTRAINT</b> TRANSACTION_ACCOUNTID_FK <b>FOREIGN KEY</b> (ACCOUNT_ID) <b>REFERENCES</b> CN1.ACCOUNT (ACCOUNT_ID) );	<b>FOREIGN KEY</b> (TRANSACTION_TYPE_ID) <b>REFERENCES</b> CN2.TRANSACTION_TYPE (TRANSACTION_TYPE_ID), <b>CONSTRAINT</b> TRANSACTION_ACCOUNTID_FK <b>FOREIGN KEY</b> (ACCOUNT_ID) <b>REFERENCES</b> CN2.ACCOUNT (ACCOUNT_ID) );
<b>CREATE TABLE</b> CN1.BANK ( BANK_ID CHAR(4) <b>CONSTRAINT</b> BANK_ID_PK <b>PRIMARY KEY</b> , BANK_NAME VARCHAR2(50) <b>NOT NULL</b> );	<b>CREATE TABLE</b> CN2.BANK ( BANK_ID CHAR(4) <b>CONSTRAINT</b> BANK_ID_PK <b>PRIMARY KEY</b> , BANK_NAME VARCHAR2(50) <b>NOT NULL</b> );
<b>CREATE TABLE</b> CN1.TRANSFER_DETAIL ( TRANSFER_DETAIL_ID <b>NUMBER</b> (7) <b>CONSTRAINT</b> TRANSFERDETAILS_ID_PK <b>PRIMARY KEY</b> , SENDER_ACCOUNT <b>NUMBER</b> (7) <b>NOT NULL</b> , RECEIVER_ACCOUNT <b>NUMBER</b> (7) <b>NOT NULL</b> , BANK_ID <b>CHAR</b> (4), AMOUNT <b>NUMBER</b> (12) <b>NOT NULL</b> , CONTENT <b>VARCHAR</b> (200), TRANSACTION_ID <b>NUMBER</b> (7), <b>CONSTRAINT</b> TRANSFERDETAILS_SENDER_ACCOUNT_FK <b>FOREIGN KEY</b> (SENDER_ACCOUNT) <b>REFERENCES</b> CN1.ACCOUNT (ACCOUNT_ID), <b>CONSTRAINT</b> TRANSFERDETAILS_BANK_ID_FK <b>FOREIGN KEY</b> (BANK_ID) <b>REFERENCES</b> CN1.BANK (BANK_ID),	<b>CREATE TABLE</b> CN2.TRANSFER_DETAIL ( TRANSFER_DETAIL_ID <b>NUMBER</b> (7) <b>CONSTRAINT</b> TRANSFERDETAILS_ID_PK <b>PRIMARY KEY</b> , SENDER_ACCOUNT <b>NUMBER</b> (7) <b>NOT NULL</b> , RECEIVER_ACCOUNT <b>NUMBER</b> (7) <b>NOT NULL</b> , BANK_ID <b>CHAR</b> (4), AMOUNT <b>NUMBER</b> (12) <b>NOT NULL</b> , CONTENT <b>VARCHAR</b> (200), TRANSACTION_ID <b>NUMBER</b> (7), <b>CONSTRAINT</b> TRANSFERDETAILS_SENDER_ACCOUNT_FK <b>FOREIGN KEY</b> (SENDER_ACCOUNT) <b>REFERENCES</b> CN2.ACCOUNT (ACCOUNT_ID), <b>CONSTRAINT</b> TRANSFERDETAILS_BANK_ID_FK <b>FOREIGN KEY</b> (BANK_ID) <b>REFERENCES</b> CN2.BANK (BANK_ID),

## THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

<b>CONSTRAINT</b> TRANSFERDETAILS_TRANSACTION_ID_FK <b>FOREIGN KEY</b> (TRANSACTION_ID) <b>REFERENCES</b> CN1.TRANSACTION (TRANSACTION_ID) );	<b>CONSTRAINT</b> TRANSFERDETAILS_TRANSACTION_ID_FK <b>FOREIGN KEY</b> (TRANSACTION_ID) <b>REFERENCES</b> CN2.TRANSACTION (TRANSACTION_ID) );
<b>CREATE SEQUENCE</b> CN1.CUSTOMER_ID_SEQUENCE <b>INCREMENT BY</b> 1 <b>START WITH</b> 10952050 <b>MAXVALUE</b> 10952999 <b>NOCYCLE</b> ;	<b>CREATE SEQUENCE</b> CN2.CUSTOMER_ID_SEQUENCE <b>INCREMENT BY</b> 1 <b>START WITH</b> 10952050 <b>MAXVALUE</b> 10952999 <b>NOCYCLE</b> ;
<b>CREATE SEQUENCE</b> CN1.ACCOUNT_ID_SEQUENCE <b>INCREMENT BY</b> 1 <b>START WITH</b> 2012050 <b>MAXVALUE</b> 2012999 <b>NOCYCLE</b> ;	<b>CREATE SEQUENCE</b> CN2.ACCOUNT_ID_SEQUENCE <b>INCREMENT BY</b> 1 <b>START WITH</b> 2012050 <b>MAXVALUE</b> 2012999 <b>NOCYCLE</b> ;
<b>CREATE SEQUENCE</b> CN1.TRANSACTION_ID_SEQUENCE <b>INCREMENT BY</b> 1 <b>START WITH</b> 1052050 <b>MAXVALUE</b> 1052999 <b>NOCYCLE</b> ;	<b>CREATE SEQUENCE</b> CN2.TRANSACTION_ID_SEQUENCE <b>INCREMENT BY</b> 1 <b>START WITH</b> 1052050 <b>MAXVALUE</b> 1052999 <b>NOCYCLE</b> ;
<b>CREATE SEQUENCE</b> CN1.TRANSFER_DETAILS_ID_SEQUENCE <b>INCREMENT BY</b> 1 <b>START WITH</b> 1050 <b>MAXVALUE</b> 99999 <b>NOCYCLE</b> ;	<b>CREATE SEQUENCE</b> CN2.TRANSFER_DETAILS_ID_SEQUENCE <b>INCREMENT BY</b> 1 <b>START WITH</b> 1050 <b>MAXVALUE</b> 99999 <b>NOCYCLE</b> ;

## THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

### 1.3.2. DML

Thực hiện thêm dữ liệu cho 2 User CN1 và CN2:

Chi nhánh 1	Chi nhánh 2
<pre>-- Insert bang BRANCH  INSERT INTO CN1.BRANCH VALUES('CN01', 'CN Tien Giang', '02838334085');</pre>	<pre>-- Insert bang BRANCH  INSERT INTO CN2.BRANCH VALUES('CN01', 'CN TP.Ho Chi Minh', '02838334085');</pre>
<pre>-- Insert bang BANK (Nhan ban)  INSERT INTO CN1.BANK VALUES('ACB', 'Ngan hang thuong mai co phan A Chau');</pre> <pre>INSERT INTO CN1.BANK VALUES('VCB', 'Ngan hang thuong mai co phan Ngoai thuong Viet Nam');</pre> <pre>INSERT INTO CN1.BANK VALUES('BIDV', 'Ngan hang thuong mai co phan Dau tu Viet Nam');</pre> <pre>INSERT INTO CN1.BANK VALUES('MB', 'Ngan hang thuong mai co phan Quan doi');</pre>	<pre>-- Insert bang BANK  INSERT INTO CN2.BANK VALUES('ACB', 'Ngan hang thuong mai co phan A Chau');</pre> <pre>INSERT INTO CN2.BANK VALUES('VCB', 'Ngan hang thuong mai co phan Ngoai thuong Viet Nam');</pre> <pre>INSERT INTO CN2.BANK VALUES('BIDV', 'Ngan hang thuong mai co phan Dau tu Viet Nam');</pre> <pre>INSERT INTO CN2.BANK VALUES('MB', 'Ngan hang thuong mai co phan Quan doi');</pre>
<pre>-- Insert bang CUSTOMER_BD  INSERT INTO CN1.CUSTOMER_BD VALUES(10952011, 'Binh', 'Dinh Huynh Thai', '272934200', 'CN01');</pre> <pre>INSERT INTO CN1.CUSTOMER_BD VALUES(10952012, 'Thu', 'Le Ngoc Minh', '272930201', 'CN01');</pre>	<pre>-- Insert bang CUSTOMER_BD  INSERT INTO CN2.CUSTOMER_BD VALUES(10952001, 'Cuc', 'Nguyen Thi Kim', '272934278', 'CN02');</pre> <pre>INSERT INTO CN2.CUSTOMER_BD VALUES(10952002, 'Tuan', 'Le Ngoc', '272930292', 'CN02');</pre>

## THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

<pre> INSERT INTO CN1.CUSTOMER_BD VALUES (10952013, 'Tuan', 'Vo Minh','272957666', 'CN01');  INSERT INTO CN1.CUSTOMER_BD VALUES (10952014, 'Truong', 'Le Nhat', '272993111', 'CN01');  INSERT INTO CN1.CUSTOMER_BD VALUES (10952015, 'Phuc', 'Tran Hoang','272993222', 'CN01');  INSERT INTO CN1.CUSTOMER_BD VALUES (10952016, 'Long', 'Pham Nhut','272443888', 'CN01');  INSERT INTO CN1.CUSTOMER_BD VALUES (10952017, 'Huong', 'Pham Thi Huynh','272443271', 'CN01');  INSERT INTO CN1.CUSTOMER_BD VALUES (10952018, 'Quynh', 'Nguyen Khanh','272443309', 'CN01');  INSERT INTO CN1.CUSTOMER_BD VALUES (10952019, 'Ha', 'Trinh Thai','272441989', 'CN01');  INSERT INTO CN1.CUSTOMER_BD VALUES (10952020, 'Tan', 'Huynh Van','272447878', 'CN01'); </pre>	<pre> INSERT INTO CN2.CUSTOMER_BD VALUES (10952003, 'Tu', 'Vo Ngoc Cam', '272957432', 'CN02');  INSERT INTO CN2.CUSTOMER_BD VALUES (10952004, 'Thuong', 'Le Hoai', '272993244', 'CN02');  INSERT INTO CN2.CUSTOMER_BD VALUES (10952005, 'Linh', 'Tran Ngoc', '272993323', 'CN02');  INSERT INTO CN2.CUSTOMER_BD VALUES (10952006, 'Long', 'Tran Minh', '272443521', 'CN02');  INSERT INTO CN2.CUSTOMER_BD VALUES (10952007, 'Trang', 'Pham Nguyen Minh', '272443521', 'CN02');  INSERT INTO CN2.CUSTOMER_BD VALUES (10952008, 'Huong', 'Nguyen Thi', '272443521', 'CN02');  INSERT INTO CN2.CUSTOMER_BD VALUES (10952009, 'Vinh', 'Le Ha', '272443521', 'CN02');  INSERT INTO CN2.CUSTOMER_BD VALUES (10952010, 'Tam', 'Nguyen Van', '272443521', 'CN02'); </pre>
<pre> -- Insert bang CUSTOMER_CSD  INSERT INTO CN1.CUSTOMER_CSD VALUES (10952014, 'Binh', 'Dinh Huynh Thai', 'Nam', TO_DATE('23/7/2001', </pre>	<pre> -- Insert bang CUSTOMER_CSD  INSERT INTO CN2.CUSTOMER_CSD VALUES (10952001, 'Cuc', 'Nguyen Thi Kim', 'Nu', TO_DATE('11/8/2001', </pre>

## THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

<p>'dd/mm/yyyy'), '117/2 Nguyen Trai, Q5, TpHCM', '0975244123');</p> <p><b>INSERT INTO</b> CN1.CUSTOMER_CSD <b>VALUES</b>(10952012, 'Thu', 'Le Ngoc Minh', 'Nam', TO_DATE('3/2/2001', 'dd/mm/yyyy'), '731 Tran Hung Dao, Q5, TpHCM', '0361234432');</p> <p><b>INSERT INTO</b> CN1.CUSTOMER_CSD <b>VALUES</b>(10952013, 'Tuan', 'Vo Minh', 'Nu', TO_DATE('12/2/1997', 'dd/mm/yyyy'), '23/5 Nguyen Trai, Q5, TpHCM', '0365238771');</p> <p><b>INSERT INTO</b> CN1.CUSTOMER_CSD <b>VALUES</b>(10952014, 'Truong', 'Le Nhat', 'Nam', TO_DATE('21/5/1885', 'dd/mm/yyyy'), '27/2 Nguyen Trai, Q5, TpHCM', '0938776255');</p> <p><b>INSERT INTO</b> CN1.CUSTOMER_CSD <b>VALUES</b>(10952015, 'Phuc', 'Tran Hoang', 'Nam', TO_DATE('21/7/1958', 'dd/mm/yyyy'), '45 Nguyen Canh Chan, Q1, TP HCM', '0938827890');</p> <p><b>INSERT INTO</b> CN1.CUSTOMER_CSD <b>VALUES</b>(10952016, 'Long', 'Pham Nhat', 'Nam', TO_DATE('5/11/1980', 'dd/mm/yyyy'), '50/34 Le Dai Hanh, Q10, TP HCM', '0937825525');</p> <p><b>INSERT INTO</b> CN1.CUSTOMER_CSD <b>VALUES</b>(10952017, 'Huong', 'Pham Thi Huynh', 'Nu', TO_DATE('1/12/1982', 'dd/mm/yyyy'), '837 Le Hong Phong, Q5, TP HCM', '0965822285');</p>	<p>'dd/mm/yyyy'), '117/2 Nguyen Trai, Q5, TpHCM', '0975244479');</p> <p><b>INSERT INTO</b> CN2.CUSTOMER_CSD <b>VALUES</b>(10952002, 'Tuan', 'Le Ngoc', 'Nam', TO_DATE('1/2/2001', 'dd/mm/yyyy'), '731 Tran Hung Dao, Q5, TpHCM', '0361234578');</p> <p><b>INSERT INTO</b> CN2.CUSTOMER_CSD <b>VALUES</b>(10952003, 'Tu', 'Vo Ngoc Cam', 'Nu', TO_DATE('12/5/2001', 'dd/mm/yyyy'), '23/5 Nguyen Trai, Q5, TpHCM', '0365238774');</p> <p><b>INSERT INTO</b> CN2.CUSTOMER_CSD <b>VALUES</b>(10952004, 'Thuong', 'Le Hoai', 'Nam', TO_DATE('21/5/2000', 'dd/mm/yyyy'), '27/2 Nguyen Trai, Q5, TpHCM', '0938776266');</p> <p><b>INSERT INTO</b> CN2.CUSTOMER_CSD <b>VALUES</b>(10952005, 'Linh', 'Tran Ngoc', 'Nam', TO_DATE('21/7/1998', 'dd/mm/yyyy'), '45 Nguyen Canh Chan, Q1, TP HCM', '0938826866');</p> <p><b>INSERT INTO</b> CN2.CUSTOMER_CSD <b>VALUES</b>(10952006, 'Long', 'Tran Minh', 'Nam', TO_DATE('10/11/1990', 'dd/mm/yyyy'), '50/34 Le Dai Hanh, Q10, TP HCM', '0937825255');</p> <p><b>INSERT INTO</b> CN2.CUSTOMER_CSD <b>VALUES</b>(10952007, 'Trang', 'Pham Nguyen Minh', 'Nu', TO_DATE('15/12/1992', 'dd/mm/yyyy'), '837 Le Hong Phong, Q5, TP HCM', '0837822285');</p>
--	---

## THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

<pre> INSERT INTO CN1.CUSTOMER_CSD VALUES (10952018, 'Quynh', 'Nguyen Khanh', 'Nu', TO_DATE('24/10/1981', 'dd/mm/yyyy'), '50/34 Le Dai Hanh, Q10, TPHCM', '0879825224');  INSERT INTO CN1.CUSTOMER_CSD VALUES (10952019, 'Ha', 'Trinh Thai', 'Nam', TO_DATE('25/8/1980', 'dd/mm/yyyy'), '34/34B Nguyen Trai, Q5, TPHCM', '0937885000');  INSERT INTO CN1.CUSTOMER_CSD VALUES (10952020, 'Tan', 'Huynh Van', 'Nam', TO_DATE('18/4/1999', 'dd/mm/yyyy'), '227 Nguyen Van Cu, Q5, TPHCM', '0837824433'); </pre>	<pre> INSERT INTO CN2.CUSTOMER_CSD VALUES (10952008, 'Huong', 'Nguyen Thi', 'Nu', TO_DATE('10/11/1991', 'dd/mm/yyyy'), '50/34 Le Dai Hanh, Q10, TPHCM', '0237825224');  INSERT INTO CN2.CUSTOMER_CSD VALUES (10952009, 'Vinh', 'Le Ha', 'Nam', TO_DATE('25/8/1990', 'dd/mm/yyyy'), '34/34B Nguyen Trai, Q5, TPHCM', '0937885255');  INSERT INTO CN2.CUSTOMER_CSD VALUES (10952010, 'Tam', 'Nguyen Van', 'Nam', TO_DATE('8/1/1989', 'dd/mm/yyyy'), '227 Nguyen Van Cu, Q5, TPHCM', '0837825251'); </pre>
<pre> -- Insert bang ACCOUNT_TYPE  INSERT INTO CN1.ACCOUNT_TYPE VALUES ('PA', 'Tai khoan thanh toan', 0);  INSERT INTO CN1.ACCOUNT_TYPE VALUES ('TSA01', 'Tai khoan tiet kiem co ky han (1 thang)', 0.034);  INSERT INTO CN1.ACCOUNT_TYPE VALUES ('TSA03', 'Tai khoan tiet kiem co ky han (3 thang)', 0.036);  INSERT INTO CN1.ACCOUNT_TYPE VALUES ('TSA06', 'Tai khoan tiet kiem co ky han (6 thang)', 0.049);  INSERT INTO CN1.ACCOUNT_TYPE VALUES ('NSA01', 'Tai khoan tiet kiem khong ky han (1 thang)', 0.004); </pre>	<pre> -- Insert bang ACCOUNT_TYPE  INSERT INTO CN2.ACCOUNT_TYPE VALUES ('PA', 'Tai khoan thanh toan', 0);  INSERT INTO CN2.ACCOUNT_TYPE VALUES ('TSA01', 'Tai khoan tiet kiem co ky han (1 thang)', 0.034);  INSERT INTO CN2.ACCOUNT_TYPE VALUES ('TSA03', 'Tai khoan tiet kiem co ky han (3 thang)', 0.036);  INSERT INTO CN2.ACCOUNT_TYPE VALUES ('TSA06', 'Tai khoan tiet kiem co ky han (6 thang)', 0.049);  INSERT INTO CN2.ACCOUNT_TYPE VALUES ('NSA01', 'Tai khoan tiet kiem khong ky han (1 thang)', 0.004); </pre>



## THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

<pre> INSERT INTO CN1.ACCOUNT_TYPE VALUES ('NSA03', 'Tai khoan tiet kiem khong ky han (3 thang)', 0.008);  INSERT INTO CN1.ACCOUNT_TYPE VALUES ('NSA06', 'Tai khoan tiet kiem khong ky han (6 thang)', 0.012); </pre>	<pre> INSERT INTO CN2.ACCOUNT_TYPE VALUES ('NSA03', 'Tai khoan tiet kiem khong ky han (3 thang)', 0.008);  INSERT INTO CN2.ACCOUNT_TYPE VALUES ('NSA06', 'Tai khoan tiet kiem khong ky han (6 thang)', 0.012); </pre>
<pre> -- Insert bang TRANSACTION_TYPE  INSERT INTO CN1.TRANSACTION_TYPE VALUES('CT01', 'Chuyen tien noi bo duoi 20 trieu', 1000);  INSERT INTO CN1.TRANSACTION_TYPE VALUES('CT02', 'Chuyen tien noi bo duoi 50 trieu', 5000);  INSERT INTO CN1.TRANSACTION_TYPE VALUES('CT03', 'Chuyen tien noi bo tren 50 trieu', 10000);  INSERT INTO CN1.TRANSACTION_TYPE VALUES('CT04', 'Chuyen tien lien ngan hang duoi 20 trieu', 7000);  INSERT INTO CN1.TRANSACTION_TYPE VALUES('CT05', 'Chuyen tien lien ngan hang duoi 50 trieu', 10000);  INSERT INTO CN1.TRANSACTION_TYPE VALUES('CT06', 'Chuyen tien lien ngan hang tren 50 trieu', 1500);  INSERT INTO CN1.TRANSACTION_TYPE VALUES('TT01', 'Thanh toan hoa don', 0); </pre>	<pre> -- Insert bang TRANSACTION_TYPE  INSERT INTO CN2.TRANSACTION_TYPE VALUES('CT01', 'Chuyen tien noi bo duoi 20 trieu', 1000);  INSERT INTO CN2.TRANSACTION_TYPE VALUES('CT02', 'Chuyen tien noi bo duoi 50 trieu', 5000);  INSERT INTO CN2.TRANSACTION_TYPE VALUES('CT03', 'Chuyen tien noi bo tren 50 trieu', 10000);  INSERT INTO CN2.TRANSACTION_TYPE VALUES('CT04', 'Chuyen tien lien ngan hang duoi 20 trieu', 7000);  INSERT INTO CN2.TRANSACTION_TYPE VALUES('CT05', 'Chuyen tien lien ngan hang duoi 50 trieu', 10000);  INSERT INTO CN2.TRANSACTION_TYPE VALUES('CT06', 'Chuyen tien lien ngan hang tren 50 trieu', 15000);  INSERT INTO CN2.TRANSACTION_TYPE VALUES('TT01', 'Thanh toan hoa don', 0); </pre>

## THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

<pre> INSERT INTO CN1.TRANSACTION_TYPE VALUES('TK00', 'Nop tien',0);  INSERT INTO CN1.TRANSACTION_TYPE VALUES('TK01', 'Gui vao tai khoan tiet kiem',0);  INSERT INTO CN1.TRANSACTION_TYPE VALUES('TK02', 'Rut tien tu tai khoan tiet kiem',0);  INSERT INTO CN1.TRANSACTION_TYPE VALUES('NT01', 'Nhan tien',0); </pre>	<pre> INSERT INTO CN2.TRANSACTION_TYPE VALUES('TK00', 'Nop tien',0);  INSERT INTO CN2.TRANSACTION_TYPE VALUES('TK01', 'Gui vao tai khoan tiet kiem',0);  INSERT INTO CN2.TRANSACTION_TYPE VALUES('TK02', 'Rut tien tu tai khoan tiet kiem',0);  INSERT INTO CN2.TRANSACTION_TYPE VALUES('NT01', 'Nhan tien',0); </pre>
<pre> -- Insert bang ACCOUNT  INSERT INTO CN1.ACCOUNT VALUES(2012011, 'PA', 140000000, TO_DATE('20/05/2010','dd/MM/yyyy'), NULL, NULL, 'Hoat dong', 10952011);  INSERT INTO CN1.ACCOUNT VALUES(2012012, 'PA', 830000000, TO_DATE('01/09/2012','dd/MM/yyyy'), NULL, NULL, 'Khoa', 10952012);  INSERT INTO CN1.ACCOUNT VALUES(2012013, 'PA', 510000000, TO_DATE('21/02/2014','dd/MM/yyyy'), NULL, NULL, 'Hoat dong', 10952013);  INSERT INTO CN1.ACCOUNT VALUES(2012014, 'PA', 222000000, TO_DATE('19/03/2018','dd/MM/yyyy'), NULL, NULL, 'Hoat dong', 10952014);  INSERT INTO CN1.ACCOUNT VALUES(2012015, 'PA', 665000000, </pre>	<pre> -- Insert bang ACCOUNT  INSERT INTO CN2.ACCOUNT VALUES(2012001, 'PA', 500000000, TO_DATE('20/05/2010','dd/MM/yyyy'), NULL, NULL, 'Hoat dong', 10952001);  INSERT INTO CN2.ACCOUNT VALUES(2012002, 'PA', 883000000, TO_DATE('01/09/2012','dd/MM/yyyy'), NULL, NULL, 'Khoa', 10952002);  INSERT INTO CN2.ACCOUNT VALUES(2012003, 'PA', 901000000, TO_DATE('21/02/2014','dd/MM/yyyy'), NULL, NULL, 'Khoa', 10952003);  INSERT INTO CN2.ACCOUNT VALUES(2012004, 'PA', 103000000, TO_DATE('09/04/2016','dd/MM/yyyy'), NULL, NULL, 'Hoat dong', 10952004);  INSERT INTO CN2.ACCOUNT VALUES(2012005, 'PA', 60000000, </pre>

## THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

<p>TO_DATE('10/05/2019','dd/MM/yyyy'), NULL, NULL, 'Khoa', 10952015);</p> <p><b>INSERT INTO</b> CN1.ACCOUNT <b>VALUES</b>(2012016, 'PA', 13000000, TO_DATE('14/03/2020','dd/MM/yyyy'), NULL, NULL, 'Hoat dong', 10952016);</p> <p><b>INSERT INTO</b> CN1.ACCOUNT <b>VALUES</b>(2012017, 'PA', 55500000, TO_DATE('11/04/2018','dd/MM/yyyy'), NULL, NULL, 'Hoat dong', 10952017);</p> <p><b>INSERT INTO</b> CN1.ACCOUNT <b>VALUES</b>(2012018, 'PA', 150000000, TO_DATE('25/07/2021','dd/MM/yyyy'), NULL, NULL, 'Hoat dong', 10952018);</p> <p><b>INSERT INTO</b> CN1.ACCOUNT <b>VALUES</b>(2012019, 'PA', 250000000, TO_DATE('08/01/2021','dd/MM/yyyy'), NULL, NULL, 'Khoa', 10952019);</p> <p><b>INSERT INTO</b> CN1.ACCOUNT <b>VALUES</b>(2012020, 'PA', 111000000, TO_DATE('17/10/2021','dd/MM/yyyy'), NULL, NULL, 'Hoat dong', 10952020);</p> <p><b>INSERT INTO</b> CN1.ACCOUNT <b>VALUES</b>(2012040, 'TSA01', 83000000, TO_DATE('25/12/2021','dd/MM/yyyy'), NULL, NULL, 'Hoat dong', 10952014);</p> <p><b>INSERT INTO</b> CN1.ACCOUNT <b>VALUES</b>(2012041, 'TSA06', 70000000, TO_DATE('24/06/2021','dd/MM/yyyy'), NULL, NULL, 'Hoat dong', 10952012);</p>	<p>TO_DATE('11/02/2017','dd/MM/yyyy'), NULL, NULL, 'Khoa', 10952005);</p> <p><b>INSERT INTO</b> CN2.ACCOUNT <b>VALUES</b>(2012006, 'PA', 3000000, TO_DATE('21/02/2020','dd/MM/yyyy'), NULL, NULL, 'Hoat dong', 10952006);</p> <p><b>INSERT INTO</b> CN2.ACCOUNT <b>VALUES</b>(2012007, 'PA', 35000000, TO_DATE('12/05/2020','dd/MM/yyyy'), NULL, NULL, 'Hoat dong', 10952007);</p> <p><b>INSERT INTO</b> CN2.ACCOUNT <b>VALUES</b>(2012008, 'PA', 50000000, TO_DATE('25/07/2020','dd/MM/yyyy'), NULL, NULL, 'Hoat dong', 10952008);</p> <p><b>INSERT INTO</b> CN2.ACCOUNT <b>VALUES</b>(2012009, 'PA', 15200000, TO_DATE('08/12/2019','dd/MM/yyyy'), NULL, NULL, 'Khoa', 10952009);</p> <p><b>INSERT INTO</b> CN2.ACCOUNT <b>VALUES</b>(2012010, 'PA', 13000000, TO_DATE('25/12/2020','dd/MM/yyyy'), NULL, NULL, 'Hoat dong', 10952010);</p> <p><b>INSERT INTO</b> CN2.ACCOUNT <b>VALUES</b>(2012040, 'TSA01', 13000000, TO_DATE('25/12/2020','dd/MM/yyyy'), NULL, NULL, 'Hoat dong', 10952010);</p> <p><b>INSERT INTO</b> CN2.ACCOUNT <b>VALUES</b>(2012041, 'TSA03', 1400000, TO_DATE('25/12/2020','dd/MM/yyyy'), NULL, NULL, 'Hoat dong', 10952007);</p> <p><b>INSERT INTO</b> CN2.ACCOUNT <b>VALUES</b>(2012042, 'NSA01', 53000000,</p>
--	---

## THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

	TO_DATE('25/12/2020','dd/MM/yyyy'), NULL, NULL, 'Khoa', 10952003);
<pre>-- Insert bang TRANSACTION  INSERT INTO CN1.TRANSACTION VALUES(1052021, 'CT04', TO_DATE('13/10/2021','dd/MM/yyyy'), 307000, 2012020);  INSERT INTO CN1.TRANSACTION VALUES(1052022, 'TK00', TO_DATE('04/08/2021','dd/MM/yyyy'), 80000, 2012017);  INSERT INTO CN1.TRANSACTION VALUES(1052023, 'NT01', TO_DATE('20/11/2021','dd/MM/yyyy'), 80000, 2012015);  INSERT INTO CN1.TRANSACTION VALUES(1052024, 'CT01', TO_DATE('13/11/2021','dd/MM/yyyy'), 81000, 2012013);  INSERT INTO CN1.TRANSACTION VALUES(1052025, 'NT01', TO_DATE('05/11/2021','dd/MM/yyyy'), 50000, 2012020);  INSERT INTO CN1.TRANSACTION VALUES(1052026, 'CT03', TO_DATE('08/11/2021','dd/MM/yyyy'), 200010000, 2012012);  INSERT INTO CN1.TRANSACTION VALUES(1052027, 'TK00', TO_DATE('14/11/2021','dd/MM/yyyy'), 88000000, 2012020);</pre>	<pre>-- Insert bang TRANSACTION  INSERT INTO CN2.TRANSACTION VALUES(1052001, 'CT04', TO_DATE('13/10/2021','dd/MM/yyyy'), 307000, 2012010);  INSERT INTO CN2.TRANSACTION VALUES(1052002, 'TK00', TO_DATE('04/08/2021','dd/MM/yyyy'), 80000, 2012007);  INSERT INTO CN2.TRANSACTION VALUES(1052003, 'NT01', TO_DATE('20/11/2021','dd/MM/yyyy'), 80000, 2012005);  INSERT INTO CN2.TRANSACTION VALUES(1052004, 'CT01', TO_DATE('13/11/2021','dd/MM/yyyy'), 81000, 2012003);  INSERT INTO CN2.TRANSACTION VALUES(1052005, 'NT01', TO_DATE('05/11/2021','dd/MM/yyyy'), 50000, 2012010);  INSERT INTO CN2.TRANSACTION VALUES(1052006, 'CT03', TO_DATE('08/11/2021','dd/MM/yyyy'), 200010000, 2012002);  INSERT INTO CN2.TRANSACTION VALUES(1052007, 'TK00', TO_DATE('14/11/2021','dd/MM/yyyy'), 88000000, 2012010);</pre>

## THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

<pre> INSERT INTO CN1.TRANSACTION VALUES (1052028, 'NT01', TO_DATE('22/11/2021','dd/MM/yyyy'), 12400000, 2012020);  INSERT INTO CN1.TRANSACTION VALUES (1052029, 'CT01', TO_DATE('28/11/2021','dd/MM/yyyy'), 1901000, 2012019);  INSERT INTO CN1.TRANSACTION VALUES (1052030, 'NT01', TO_DATE('14/11/2021','dd/MM/yyyy'), 8800000, 2012020);  INSERT INTO CN1.TRANSACTION VALUES (1052031, 'NT01', TO_DATE('22/11/2021','dd/MM/yyyy'), 12400000, 2012020);  INSERT INTO CN1.TRANSACTION VALUES (1052032, 'CT06', TO_DATE('28/11/2021','dd/MM/yyyy'), 90015000, 2012012);  INSERT INTO CN1.TRANSACTION VALUES (1052033, 'CT02', TO_DATE('21/05/2021','dd/MM/yyyy'), 18900000, 2012018);  INSERT INTO CN1.TRANSACTION VALUES (1052034, 'NT01', TO_DATE('11/11/2021','dd/MM/yyyy'), 14900000, 2012018);  INSERT INTO CN1.TRANSACTION VALUES (1052035, 'CT03', </pre>	<pre> INSERT INTO CN2.TRANSACTION VALUES (1052008, 'NT01', TO_DATE('22/11/2021','dd/MM/yyyy'), 12400000, 2012010);  INSERT INTO CN2.TRANSACTION VALUES (1052009, 'CT01', TO_DATE('28/11/2021','dd/MM/yyyy'), 1901000, 2012009);  INSERT INTO CN2.TRANSACTION VALUES (1052010, 'NT01', TO_DATE('14/11/2021','dd/MM/yyyy'), 8800000, 2012010);  INSERT INTO CN2.TRANSACTION VALUES (1052011, 'NT01', TO_DATE('22/11/2021','dd/MM/yyyy'), 12400000, 2012010);  INSERT INTO CN2.TRANSACTION VALUES (1052012, 'CT06', TO_DATE('28/11/2021','dd/MM/yyyy'), 90015000, 2012002);  INSERT INTO CN2.TRANSACTION VALUES (1052013, 'CT02', TO_DATE('21/05/2021','dd/MM/yyyy'), 18900000, 2012008);  INSERT INTO CN2.TRANSACTION VALUES (1052014, 'NT01', TO_DATE('11/11/2021','dd/MM/yyyy'), 14900000, 2012008);  INSERT INTO CN2.TRANSACTION VALUES (1052015, 'CT03', </pre>
---	---

## THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

<pre> TO_DATE('13/11/2021','dd/MM/yyyy'), 230000000, 2012018);  INSERT INTO CN1.TRANSACTION VALUES(1052036, 'NT01', TO_DATE('08/10/2021','dd/MM/yyyy'), 230000000, 2012017);  INSERT INTO CN1.TRANSACTION VALUES(1052037, 'TT01', TO_DATE('18/09/2021','dd/MM/yyyy'), 80000000, 2012012);  INSERT INTO CN1.TRANSACTION VALUES(1052038, 'CT06', TO_DATE('30/10/2021','dd/MM/yyyy'), 65000000, 2012012);  INSERT INTO CN1.TRANSACTION VALUES(1052039, 'CT05', TO_DATE('08/10/2021','dd/MM/yyyy'), 23010000, 2012020);  INSERT INTO CN1.TRANSACTION VALUES(1052040, 'NT01', TO_DATE('27/06/2021','dd/MM/yyyy'), 230000000, 2012016); </pre>	<pre> TO_DATE('13/11/2021','dd/MM/yyyy'), 230000000, 2012008);  INSERT INTO CN2.TRANSACTION VALUES(1052016, 'NT01', TO_DATE('08/10/2021','dd/MM/yyyy'), 230000000, 2012007);  INSERT INTO CN2.TRANSACTION VALUES(1052017, 'TT01', TO_DATE('18/09/2021','dd/MM/yyyy'), 80000000, 2012002);  INSERT INTO CN2.TRANSACTION VALUES(1052018, 'CT06', TO_DATE('30/10/2021','dd/MM/yyyy'), 65000000, 2012002);  INSERT INTO CN2.TRANSACTION VALUES(1052019, 'CT05', TO_DATE('08/10/2021','dd/MM/yyyy'), 23010000, 2012010);  INSERT INTO CN2.TRANSACTION VALUES(1052020, 'NT01', TO_DATE('27/06/2021','dd/MM/yyyy'), 230000000, 2012006);  INSERT INTO CN2.TRANSACTION VALUES(1052040, 'CT01', TO_DATE('14/11/2021','dd/MM/yyyy'), 298000000, 2012010);  INSERT INTO CN2.TRANSACTION VALUES(1052041, 'CT02', TO_DATE('15/11/2021','dd/MM/yyyy'), 8800000, 2012010); </pre>
--	--

## THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

```
INSERT INTO CN2.TRANSACTION
VALUES (1052042, 'CT03',
TO_DATE('17/11/2021','dd/MM/yyyy'),
38800000, 2012010);

INSERT INTO CN2.TRANSACTION
VALUES (1052043, 'CT06',
TO_DATE('11/12/2021','dd/MM/yyyy'),
118800000, 2012010);

INSERT INTO CN2.TRANSACTION
VALUES (1052044, 'TT01',
TO_DATE('02/12/2021','dd/MM/yyyy'),
81800000, 2012010);

INSERT INTO CN2.TRANSACTION
VALUES (1052045, 'TK01',
TO_DATE('05/12/2021','dd/MM/yyyy'),
50800000, 2012010);

INSERT INTO CN2.TRANSACTION
VALUES (1052046, 'TK02',
TO_DATE('20/12/2021','dd/MM/yyyy'),
45800000, 2012010);

INSERT INTO CN2.TRANSACTION
VALUES (1052047, 'TK02',
TO_DATE('20/12/2021','dd/MM/yyyy'),
45800000, 2012010);

INSERT INTO CN2.TRANSACTION
VALUES (1052060, 'CT01', SYSDATE,
45000000, 2012004);

INSERT INTO CN2.TRANSACTION
VALUES (1052061, 'CT02', SYSDATE,
15800000, 2012006);
```

## THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

<pre>-- Insert bang TRANSFER_DETAIL  INSERT INTO CN1.TRANSFER_DETAIL VALUES(1020, 2012020, 2012017, 'VCB', 300000, 'Van Tan chuyen tien', 1052021);  INSERT INTO CN1.TRANSFER_DETAIL VALUES(1021, 2012013, 2012012, 'ACB', 80000, 'Tra tien cafe', 1052024);  INSERT INTO CN1.TRANSFER_DETAIL VALUES(1022, 2012012, 2012020, 'ACB', 200000000, 'LE NGOC MINH THU chuyen tien thue nha 1 nam', 1052026);  INSERT INTO CN1.TRANSFER_DETAIL VALUES(1023, 2012019, 2012018, 'ACB', 1900000, 'TRINH THAI HA chuyen tien', 1052019);  INSERT INTO CN1.TRANSFER_DETAIL VALUES(1024, 2012017, 2012018, 'BIDV', 90000000, 'Tra tien luong cho nhan vien NGUYEN KHANH QUYNH', 1052022);  INSERT INTO CN1.TRANSFER_DETAIL VALUES(1025, 2012019, 2012014, 'ACB', 23000000, 'Tra tien mua labtop', 1052029);</pre>	<pre>-- Insert bang TRANSFER_DETAIL  INSERT INTO CN2.TRANSFER_DETAIL VALUES(1001, 2012010, 2012007, 'VCB', 300000, 'NGUYEN VAN TAM chuyen tien', 1052001);  INSERT INTO CN2.TRANSFER_DETAIL VALUES(1002, 2012003, 2012002, 'ACB', 80000, 'Tra tien café', 1052004);  INSERT INTO CN2.TRANSFER_DETAIL VALUES(1003, 2012002, 2012010, 'ACB', 200000000, 'LE NGOC TUAN chuyen tien thue nha 1 nam', 1052006);  INSERT INTO CN2.TRANSFER_DETAIL VALUES(1004, 2012009, 2012008, 'ACB', 1900000, 'LE HA VINH chuyen tien', 1052009);  INSERT INTO CN2.TRANSFER_DETAIL VALUES(1005, 2012002, 2012008, 'BIDV', 90000000, 'Tra tien luong cho nhan vien NGUYEN THI HUONG', 1052012);  INSERT INTO CN2.TRANSFER_DETAIL VALUES(1006, 2012010, 2012004, 'ACB', 23000000, 'Tra tien mua labtop', 1052019);</pre>
<pre>-- Insert bang EMPLOYEE  INSERT INTO CN1.EMPLOYEE VALUES(70060, 'Nguyen Van Tien', TO_DATE('06/11/2012', 'dd/mm/yyyy'), 'Truong phong Ke toan', 30000000, 'CN01');</pre>	<pre>-- Insert bang EMPLOYEE  INSERT INTO CN2.EMPLOYEE VALUES(70040, 'Duong Minh Anh', TO_DATE('12/11/2012', 'dd/mm/yyyy'), 'Truong phong Ke toan', 30000000, 'CN02');</pre>



## THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

**INSERT INTO** CN1.EMPLOYEE

**VALUES** (70051, 'Pham Thi Thanh Hang',  
TO\_DATE('12/11/2012', 'dd/mm/yyyy'),  
'Giao dich vien', 15000000, 'CN01');

**INSERT INTO** CN1.EMPLOYEE

**VALUES** (70052, 'Ta Minh Nhat',  
TO\_DATE('04/11/2012', 'dd/mm/yyyy'),  
'Bao ve', 7000000, 'CN02');

**INSERT INTO** CN1.EMPLOYEE

**VALUES** (70053, 'Truong Thi Kim Thanh',  
TO\_DATE('31/5/2017', 'dd/mm/yyyy'),  
'Giao dich vien', 15000000, 'CN01');

**INSERT INTO** CN1.EMPLOYEE

**VALUES** (70054, 'Nguyen Hoang Nam',  
TO\_DATE('13/10/2013', 'dd/mm/yyyy'),  
'Truong chi nhanh', 30000000,  
'CN01');

**INSERT INTO** CN1.EMPLOYEE

**VALUES** (70055, 'Do Van Sang',  
TO\_DATE('2/12/2014', 'dd/mm/yyyy'),  
'Giao dich vien', 15000000, 'CN01');

**INSERT INTO** CN1.EMPLOYEE

**VALUES** (70056, 'Pham Nhat Cuong',  
TO\_DATE('14/8/2019', 'dd/mm/yyyy'),  
'Giao dich vien', 15000000, 'CN01');

**INSERT INTO** CN1.EMPLOYEE

**VALUES** (70057, 'Chau Thi Thuy Tien',  
TO\_DATE('12/5/2020', 'dd/mm/yyyy'),  
'Bao ve', 7000000, 'CN02');

**INSERT INTO** CN1.EMPLOYEE

**VALUES** (70058, 'Vu Kim Hong',

**INSERT INTO** CN2.EMPLOYEE

**VALUES** (70041, 'Nguyen Minh Sang',  
TO\_DATE('12/11/2012', 'dd/mm/yyyy'),  
'Giao dich vien', 15000000, 'CN02');

**INSERT INTO** CN2.EMPLOYEE

**VALUES** (70042, 'Ta Vu Ngoc Tuyen',  
TO\_DATE('12/11/2012', 'dd/mm/yyyy'),  
'Bao ve', 7000000, 'CN02');

**INSERT INTO** CN2.EMPLOYEE

**VALUES** (70043, 'Nguyen Xuan Minh Thu',  
TO\_DATE('31/1/2017', 'dd/mm/yyyy'),  
'Giao dich vien', 15000000, 'CN02');

**INSERT INTO** CN2.EMPLOYEE

**VALUES** (70044, 'Doan Van Tien Nhat',  
TO\_DATE('3/12/2013', 'dd/mm/yyyy'),  
'Truong chi nhanh', 30000000,  
'CN02');

**INSERT INTO** CN2.EMPLOYEE

**VALUES** (70045, 'Do Thi Cam Tu',  
TO\_DATE('28/4/2014', 'dd/mm/yyyy'),  
'Giao dich vien', 15000000, 'CN02');

**INSERT INTO** CN2.EMPLOYEE

**VALUES** (70046, 'Dang Van Quang',  
TO\_DATE('6/1/2018', 'dd/mm/yyyy'),  
'Giao dich vien', 15000000, 'CN02');

**INSERT INTO** CN2.EMPLOYEE

**VALUES** (70047, 'Tu Thi Ngoc Chau',  
TO\_DATE('18/5/2015', 'dd/mm/yyyy'),  
'Bao ve', 7000000, 'CN02');

**INSERT INTO** CN2.EMPLOYEE

**VALUES** (70048, 'Vu Kim Tuyen',

## THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

<pre>TO_DATE('10/6/2019', 'dd/mm/yyyy'), 'Bao ve', 7000000, 'CN02');  INSERT INTO CN1.EMPLOYEE VALUES(70059, 'Nguyen Tran Minh Anh', TO_DATE('20/6/2021', 'dd/mm/yyyy'), 'Giao dich vien', 15000000, 'CN01');  INSERT INTO CN1.EMPLOYEE VALUES(70050, 'Doan Van Dat', TO_DATE('9/11/2015', 'dd/mm/yyyy'), 'Truong phong kinh doanh', 30000000, 'CN01');</pre>	<pre>TO_DATE('18/7/2010', 'dd/mm/yyyy'), 'Bao ve', 7000000, 'CN02');  INSERT INTO CN2.EMPLOYEE VALUES(70049, 'Tran Doan Anh Vu', TO_DATE('26/10/2011', 'dd/mm/yyyy'), 'Giao dich vien', 15000000, 'CN02');  INSERT INTO CN2.EMPLOYEE VALUES(70050, 'Vu Thi Thu Hong', TO_DATE('12/11/2012', 'dd/mm/yyyy'), 'Truong phong kinh doanh', 30000000, 'CN02');</pre>
---	--

### 1.4. KIẾN TRÚC PHÂN QUYỀN

#### 1.4.1. Mô tả kiến trúc phân quyền

Chi nhánh 1: Có 3 quyền

- + Director: Xem được thông tin tất cả các quan hệ trong trên tất cả chi nhánh
- + Manager: Xem, thêm, sửa được thông tin của các quan hệ ACCOUNT\_TYPE, TRANSACTION\_TYPE, CUSTOMER\_BD, EMPLOYEE và chỉ xem được thông tin của các quan hệ ACCOUNT, TRANSACTION, BANK, TRANSFER\_DETAIL của chi nhánh 1.
- + Teller: Chỉ xem được thông tin của các quan hệ ACCOUNT, TRANSACTION, TRANSFER\_DETAIL, TRANSACTION\_TYPE, CUSTOMER\_BD của chi nhánh 1. Ngoài ra Teller được phép thêm thông tin vào quan hệ TRANSACTION và TRANSFER\_DETAIL.

## THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

Chi nhánh 2: Có 2 quyền

- + Manager: Xem, thêm, sửa được thông tin của các quan hệ ACCOUNT\_TYPE, TRANSACTION\_TYPE, CUSTOMER\_BD, EMPLOYEE và chỉ xem được thông tin của các quan hệ ACCOUNT, TRANSACTION, BANK, TRANSFER\_DETAIL của chi nhánh 2.
- + Teller: Chỉ xem được thông tin của các quan hệ ACCOUNT, TRANSACTION, TRANSFER\_DETAIL, TRANSACTION\_TYPE, CUSTOMER\_BD của chi nhánh 2. Ngoài ra Teller được phép thêm thông tin vào quan hệ TRANSACTION và TRANSFER\_DETAIL..

### 1.4.2. Minh họa

#### 1.4.2.1. Tạo USER ban đầu: Chi nhánh

```
ALTER SESSION SET "_ORACLE_SCRIPT"=TRUE;

CREATE USER CN1 IDENTIFIED BY CN1;
GRANT CONNECT, DBA, CREATE PUBLIC DATABASE LINK TO CN1;

CREATE USER CN2 IDENTIFIED BY CN2;
GRANT CONNECT, DBA, CREATE PUBLIC DATABASE LINK TO CN2;
```

#### 1.4.2.2. Tạo ROLE

```
CREATE ROLE DIRECTOR_ROLE;
CREATE ROLE MANAGER_ROLE;
CREATE ROLE TELLER_ROLE;
```

#### 1.4.2.3. Phân quyền: Giám đốc ở chi nhánh 1

```
GRANT CREATE SESSION, DBA TO DIRECTOR_ROLE;
```

## THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

### 1.4.2.4. Phân quyền: Trưởng chi nhánh ở hai chi nhánh

```
GRANT CONNECT, CREATE SESSION TO MANAGER_ROLE;  
GRANT SELECT ON CN1.ACCOUNT TO MANAGER_ROLE;  
GRANT SELECT ON CN2.ACCOUNT TO MANAGER_ROLE;  
GRANT SELECT, INSERT, UPDATE ON CN1.ACCOUNT_TYPE TO MANAGER_ROLE;  
GRANT SELECT, INSERT, UPDATE ON CN2.ACCOUNT_TYPE TO MANAGER_ROLE;  
GRANT SELECT ON CN1.TRANSACTION TO MANAGER_ROLE;  
GRANT SELECT ON CN2.TRANSACTION TO MANAGER_ROLE;  
GRANT SELECT, INSERT, UPDATE ON CN1.TRANSACTION_TYPE TO MANAGER_ROLE;  
GRANT SELECT, INSERT, UPDATE ON CN2.TRANSACTION_TYPE TO MANAGER_ROLE;  
GRANT SELECT ON CN1.BANK TO MANAGER_ROLE;  
GRANT SELECT ON CN2.BANK TO MANAGER_ROLE;  
GRANT SELECT ON CN1.TRANSFER_DETAIL TO MANAGER_ROLE;  
GRANT SELECT ON CN2.TRANSFER_DETAIL TO MANAGER_ROLE;;  
GRANT SELECT, INSERT, UPDATE ON CN1.CUSTOMER_BD TO MANAGER_ROLE;  
GRANT SELECT, INSERT, UPDATE ON CN2.CUSTOMER_BD TO MANAGER_ROLE;  
  
GRANT SELECT, INSERT, UPDATE ON CN1.EMPLOYEE TO MANAGER_ROLE;  
GRANT SELECT, INSERT, UPDATE ON CN2.EMPLOYEE TO MANAGER_ROLE;
```

### 1.4.2.5. Phân quyền: Giao dịch viên ở hai chi nhánh

```
GRANT CONNECT, CREATE SESSION TO TELLER_ROLE;  
GRANT SELECT ON CN2.ACCOUNT TO TELLER_ROLE;  
GRANT SELECT, INSERT ON CN2.TRANSACTION TO TELLER_ROLE;  
GRANT SELECT ON CN2.TRANSACTION_TYPE TO TELLER_ROLE;  
GRANT SELECT ON CN2.BANK TO TELLER_ROLE;  
GRANT SELECT, INSERT ON CN2.TRANSFER_DETAIL TO TELLER_ROLE;  
GRANT SELECT ON CN2.CUSTOMER_BD TO TELLER_ROLE;
```

### 1.4.2.6. Tạo User: Giám đốc ở chi nhánh 1

```
CREATE USER Director IDENTIFIED BY GD;  
GRANT DIRECTOR_ROLE TO Director;
```

## THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

### 1.4.2.7. Tạo User: Trưởng chi nhánh ở hai chi nhánh

```
CREATE USER ManagerCN1 IDENTIFIED BY TCN1;  
GRANT MANAGER_ROLE TO ManagerCN1;  
  
CREATE USER ManagerCN2 IDENTIFIED BY TCN2;  
GRANT MANAGER_ROLE TO ManagerCN2;
```

### 1.4.2.8. Tạo User: Giao dịch viên ở hai chi nhánh

```
CREATE USER TellerCN1 IDENTIFIED BY GDVCN1;  
GRANT TELLER_ROLE TO TellerCN1;  
  
CREATE USER TellerCN2 IDENTIFIED BY GDVCN2;  
GRANT TELLER_ROLE TO TellerCN2;
```

## 1.5. CÁC CÂU TRUY VẤN

### 1.5.1. Câu truy vấn 1

Sử dụng tài khoản Trưởng chi nhánh 1, xem sao kê của tài khoản có số tài khoản là 2012010 (ACCOUNT\_ID) tại chi nhánh 2 từ ngày 01/11/2021 đến ngày 26/11/2021. Thông tin hiển thị bao gồm mã giao dịch (TRANSACTION\_ID), thời gian giao dịch (TRANSACTION\_DATE), tổng số tiền giao dịch (TOTAL\_TRANSACTION\_AMOUNT), tên loại giao dịch (TRANSACTION\_NAME), phí thực hiện giao dịch (FEE).

```
CONNECT ManagerCN1/TCN1;  
  
SELECT T.TRANSACTION_ID, T.TRANSACTION_DATE, T.TOTAL_TRANSACTION_AMOUNT,  
TP.NAME TRANSACTION_NAME ,TP.FEE, A.ACCOUNT_ID  
FROM ((CN2.TRANSACTION@cn2_link T JOIN CN2.TRANSACTION_TYPE@cn2_link TP ON  
T.TRANSACTION_TYPE_ID = TP.TRANSACTION_TYPE_ID)  
      JOIN CN2.ACCOUNT@cn2_link A ON A.ACCOUNT_ID = T.ACCOUNT_ID)  
      JOIN CN2.CUSTOMER_BD@cn2_link C ON C.CUSTOMER_ID =  
A.CUSTOMER_ID  
WHERE A.ACCOUNT_ID = 2012010 AND ACCOUNT_TYPE_ID = 'PA'  
AND (TRANSACTION_DATE BETWEEN TO_DATE('01/11/2021', 'dd/MM/yyyy')  
AND TO_DATE('26/11/2021 23:59:59', 'dd/MM/yyyy HH24:MI:SS'));
```

### 1.5.2. Câu truy vấn 2

Sử dụng tài khoản giám đốc tại chi nhánh 1, in ra thông tin những khách hàng trên cả 2 chi nhánh vừa có mở tài khoản thành toán, vừa có mở tài khoản tiết kiệm. Thông tin hiển thị bao gồm họ tên khách hàng (FIRST\_NAME, LAST\_NAME), mã chi nhánh (BRANCH\_ID).

```
CONNECT Director/GD;  
  
(SELECT C.FIRST_NAME || ' ' || C.LAST_NAME Ten_KhachHang, C.BRANCH_ID  
FROM CN1.ACCOUNT ACC JOIN CN1.CUSTOMER_BD C ON ACC.CUSTOMER_ID =  
C.CUSTOMER_ID  
WHERE ACC.ACCOUNT_TYPE_ID = 'PA'  
INTERSECT  
SELECT C.FIRST_NAME || ' ' || C.LAST_NAME Ten_KhachHang ,C.BRANCH_ID  
FROM CN1.ACCOUNT ACC JOIN CN1.CUSTOMER_BD C ON ACC.CUSTOMER_ID =  
C.CUSTOMER_ID  
WHERE ACC.ACCOUNT_TYPE_ID LIKE '_SA%')  
UNION  
(SELECT C.FIRST_NAME || ' ' || C.LAST_NAME Ten_KhachHang ,C.BRANCH_ID  
FROM CN2.ACCOUNT@cn2_link ACC JOIN CN2.CUSTOMER_BD@cn2_link C ON  
ACC.CUSTOMER_ID = C.CUSTOMER_ID  
WHERE ACC.ACCOUNT_TYPE_ID = 'PA'  
INTERSECT  
SELECT C.FIRST_NAME || ' ' || C.LAST_NAME Ten_KhachHang ,C.BRANCH_ID  
FROM CN2.ACCOUNT@cn2_link ACC JOIN CN2.CUSTOMER_BD@cn2_link C ON  
ACC.CUSTOMER_ID = C.CUSTOMER_ID  
WHERE ACC.ACCOUNT TYPE ID LIKE ' SA%');
```

### 1.5.3. Câu truy vấn 3

Sử dụng tài khoản giao dịch viên chi nhánh 2, tính tổng doanh thu (tổng phí giao dịch) của khách hàng có doanh thu lớn hơn 10,000 VND trong năm 2021 ở chi nhánh 1. Sắp xếp tăng dần theo doanh thu.

```
CONNECT TellerCN2/GDVCN2;  
  
SELECT C.CUSTOMER_ID, SUM(FEE)  
FROM ((CN1.CUSTOMER_BD@cn1_link C JOIN CN1.ACCOUNT@cn1_link A ON  
A.CUSTOMER_ID = C.CUSTOMER_ID)  
      JOIN CN1.TRANSACTION@cn1_link T ON T.ACCOUNT_ID = A.ACCOUNT_ID)  
      JOIN CN1.TRANSACTION_TYPE@cn1_link TT ON TT.TRANSACTION_TYPE_ID  
= T.TRANSACTION_TYPE_ID
```

## THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

```
WHERE EXTRACT(YEAR FROM TRANSACTION_DATE) = 2021
GROUP BY C.CUSTOMER_ID
HAVING SUM(FEE) > 10000
ORDER BY SUM(FEE) DESC;
```

### 1.5.4. Câu truy vấn 4

Sử dụng tài khoản Trưởng chi nhánh 2, in ra những khách hàng chưa mở bất cứ tài khoản tiết kiệm nào ở tất cả chi nhánh.

```
CONNECT ManagerCN2/TCN2;

(SELECT CUSTOMER_ID, LAST_NAME || ' ' || FIRST_NAME Ten_KhachHang
FROM CN2.CUSTOMER_BD
MINUS
SELECT C.CUSTOMER_ID, C.LAST_NAME || ' ' || C.FIRST_NAME Ten_KhachHang
FROM CN2.ACCOUNT ACC JOIN CN2.CUSTOMER_BD C ON ACC.CUSTOMER_ID =
C.CUSTOMER_ID
WHERE ACC.ACCOUNT_TYPE_ID LIKE '_SA%')
UNION
(SELECT CUSTOMER_ID, LAST_NAME || ' ' || FIRST_NAME Ten_KhachHang
FROM CN1.CUSTOMER_BD@cn1_link
MINUS
SELECT C.CUSTOMER_ID, C.LAST_NAME || ' ' || C.FIRST_NAME Ten_KhachHang
FROM CN1.ACCOUNT@cn1_link ACC JOIN CN1.CUSTOMER_BD@cn1_link C ON
ACC.CUSTOMER_ID = C.CUSTOMER_ID
WHERE ACC.ACCOUNT_TYPE_ID LIKE '_SA%');
```

### 1.5.5 Câu truy vấn 5

Sử dụng tài khoản giám đốc chi nhánh 1, tìm tổng số tiền phải trả cho tất cả Giao dịch viên (JOB = 'Giao dịch viên') theo từng chi nhánh.

```
CONNECT Director/GD;

SELECT BR.BRANCH_ID, BR.BRANCH_NAME, SUM(E.SALARY) TONG_LUONG
FROM CN1.EMPLOYEE@cn1_link E JOIN CN1.BRANCH@cn1_link BR ON E.BRANCH_ID =
BR.BRANCH_ID
WHERE JOB = 'Giao dịch viên'
GROUP BY BR.BRANCH_ID, BR.BRANCH_NAME
UNION
SELECT BR.BRANCH_ID, BR.BRANCH_NAME, SUM(E.SALARY) TONG_LUONG
FROM CN2.EMPLOYEE E JOIN CN2.BRANCH BR ON E.BRANCH_ID = BR.BRANCH_ID
WHERE JOB = 'Giao dịch viên'
```

## THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

```
GROUP BY BR.BRANCH_ID, BR.BRANCH_NAME;
```

### 1.5.6. Câu truy vấn 6

Sử dụng tài khoản giao dịch viên chi nhánh 1, tìm tất cả chi tiết chuyển tiền đến Ngân hàng thương mại cổ phần Ngoại thương Việt Nam (BANK\_NAME = 'Ngan hang thuong mai co phan Ngoai thuong Viet Nam'). Thông tin hiển thị bao gồm mã giao dịch (TRANSACTION\_ID), số tài khoản người gửi (SENDER\_ACCOUNT), số tài khoản người nhận (RECEIVER\_ACCOUNT), số tiền chuyển (AMOUNT) và nội dung chuyển tiền (CONTENT).

```
CONNECT TellerCN1/GDVCN1;

SELECT TD.TRANSACTION_ID, SENDER_ACCOUNT, RECEIVER_ACCOUNT, AMOUNT, CONTENT
FROM (CN2.TRANSFER_DETAIL@cn2_link TD JOIN CN2.BANK@cn2_link B ON B.BANK_ID
= TD.BANK_ID)
JOIN CN2.TRANSACTION@cn2_link T ON T.TRANSACTION_ID = TD.TRANSACTION_ID
WHERE EXTRACT(MONTH FROM TRANSACTION_DATE) = EXTRACT(MONTH FROM SYSDATE)
AND EXTRACT(DAY FROM TRANSACTION_DATE) = EXTRACT(DAY FROM SYSDATE)
AND EXTRACT(YEAR FROM TRANSACTION_DATE) = EXTRACT(YEAR FROM SYSDATE)
AND BANK_NAME = 'Ngan hang thuong mai co phan Ngoai thuong Viet Nam';
```

### 1.5.7. Câu truy vấn 7

Sử dụng tài khoản giao dịch viên chi nhánh 1, tìm thông tin những khách hàng ở tất cả chi nhánh đã thực hiện tất cả các loại giao dịch.

```
CONNECT TellerCN1/GDVCN1;

SELECT C.CUSTOMER_ID, C.FIRST_NAME || ' ' || C.LAST_NAME TEN_KHACHHANG,
C.BRANCH_ID
FROM CN1.CUSTOMER_BD C JOIN CN1.ACCOUNT A ON A.CUSTOMER_ID = C.CUSTOMER_ID
WHERE NOT EXISTS ( SELECT *
FROM CN1.TRANSACTION_TYPE TT
WHERE NOT EXISTS (SELECT *
FROM CN1.TRANSACTION T
WHERE A.ACCOUNT_ID = T.ACCOUNT_ID
AND TT.TRANSACTION_TYPE_ID =
T.TRANSACTION_TYPE_ID))
UNION
```



## THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

```
SELECT C.CUSTOMER_ID, C.FIRST_NAME || ' ' || C.LAST_NAME TEN_KHACHHANG,  
C.BRANCH_ID  
FROM CN2.CUSTOMER_BD@cn2_link C JOIN CN1.ACCOUNT@cn2_link A ON A.CUSTOMER_ID  
= C.CUSTOMER_ID  
WHERE NOT EXISTS ( SELECT *  
                     FROM CN2.TRANSACTION_TYPE@cn2_link TT  
                     WHERE NOT EXISTS (SELECT *  
                                       FROM CN2.TRANSACTION@cn2_link T  
                                       WHERE A.ACCOUNT_ID = T.ACCOUNT_ID  
                                       AND TT.TRANSACTION_TYPE_ID =  
T.TRANSACTION_TYPE_ID));
```

### 1.5.8. Câu truy vấn 8

Sử dụng tài khoản Trưởng chi nhánh 2, tìm thông tin tài khoản thanh toán với trạng thái đang hoạt động (STATUS = 'Hoạt động') và có số lần giao dịch trong tháng 11/2021 lớn nhất tại tất cả chi nhánh.

```
CONNECT ManagerCN2/TCN2;  
  
SELECT *  
FROM (  
    SELECT A.ACCOUNT_ID, CUSTOMER_ID, COUNT(*) SOLANGD  
    FROM CN2.ACCOUNT A JOIN CN2.TRANSACTION T ON A.ACCOUNT_ID = T.ACCOUNT_ID  
    WHERE EXTRACT(MONTH FROM TRANSACTION_DATE) = 11 AND EXTRACT(YEAR FROM  
TRANSACTION_DATE) = 2021  
    AND ACCOUNT_TYPE_ID = 'PA' AND STATUS = 'Hoạt động'  
    GROUP BY A.ACCOUNT_ID, CUSTOMER_ID  
    HAVING COUNT(*) >= ALL( SELECT COUNT(*)  
                             FROM CN2.ACCOUNT A JOIN CN2.TRANSACTION T ON  
A.ACCOUNT_ID = T.ACCOUNT_ID  
                             WHERE EXTRACT(MONTH FROM TRANSACTION_DATE) = 11  
AND EXTRACT(YEAR FROM TRANSACTION_DATE) = 2021  
                             AND ACCOUNT_TYPE_ID = 'PA' AND STATUS = 'Hoạt  
động'  
                             GROUP BY A.ACCOUNT_ID, CUSTOMER_ID)  
    UNION  
    SELECT A.ACCOUNT_ID, CUSTOMER_ID, COUNT(*) SOLANGD  
    FROM CN1.ACCOUNT@cn1_link A JOIN CN1.TRANSACTION@cn1_link T ON  
A.ACCOUNT_ID = T.ACCOUNT_ID  
    WHERE EXTRACT(MONTH FROM TRANSACTION_DATE) = 11 AND EXTRACT(YEAR FROM  
TRANSACTION_DATE) = 2021  
    AND ACCOUNT_TYPE_ID = 'PA' AND STATUS = 'Hoạt động'  
    GROUP BY A.ACCOUNT_ID, CUSTOMER_ID  
    HAVING COUNT(*) >= ALL( SELECT COUNT(*)  
                             FROM CN1.ACCOUNT@cn1_link A JOIN  
CN1.TRANSACTION@cn1_link T ON A.ACCOUNT_ID = T.ACCOUNT_ID
```

## THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

```
WHERE EXTRACT (MONTH FROM TRANSACTION_DATE) = 11
AND EXTRACT (YEAR FROM TRANSACTION_DATE) = 2021
AND ACCOUNT_TYPE_ID = 'PA' AND STATUS = 'Hoat
dong'
GROUP BY A.ACCOUNT_ID, CUSTOMER_ID)
WHERE rownum = 1
ORDER BY SOLANGD DESC;
```

### 1.5.9. Câu truy vấn 9

Sử dụng tài khoản Trưởng chi nhánh 1, tìm loại giao dịch được thực hiện ở cả hai chi nhánh. Thông tin hiển thị bao gồm mã loại giao dịch (TRANSACTION\_TYPE\_ID) và tên loại giao dịch (NAME).

```
CONNECT ManagerCN1/TCN1;

SELECT DISTINCT (TT.TRANSACTION_TYPE_ID), NAME TRANSACTION_TYPE_NAME
FROM (CN1.TRANSACTION T JOIN CN1.TRANSACTION_TYPE TT ON
T.TRANSACTION_TYPE_ID = TT.TRANSACTION_TYPE_ID)
JOIN CN1.ACCOUNT A ON A.ACCOUNT_ID = T.ACCOUNT_ID
WHERE ACCOUNT_TYPE_ID = 'PA'
INTERSECT
SELECT DISTINCT (TT.TRANSACTION_TYPE_ID), NAME TRANSACTION_TYPE_NAME
FROM (CN2.TRANSACTION@cn2_link T JOIN CN2.TRANSACTION_TYPE@cn2_link TT ON
T.TRANSACTION_TYPE_ID = TT.TRANSACTION_TYPE_ID)
JOIN CN2.ACCOUNT@cn2_link A ON A.ACCOUNT_ID = T.ACCOUNT_ID
WHERE ACCOUNT_TYPE_ID = 'PA';
```

### 1.5.10. Câu truy vấn 10

Sử dụng tài khoản giám đốc chi nhánh 1, in ra danh sách các loại tài khoản tiết kiệm (TRANSACTION\_TYPE\_ID, TRANSACTION\_TYPE\_NAME) không được mở trong tháng 12 năm 2021 ở tất cả chi nhánh.

```
CONNECT Director/GD;

(SELECT ACCOUNT_TYPE_ID, ACCOUNT_TYPE_NAME
FROM CN1.ACCOUNT_TYPE
WHERE ACCOUNT_TYPE_ID LIKE '%SA%'
MINUS
SELECT A.ACCOUNT_TYPE_ID, ACCOUNT_TYPE_NAME
FROM CN1.ACCOUNT A JOIN CN1.ACCOUNT_TYPE AT ON A.ACCOUNT_TYPE_ID =
AT.ACCOUNT_TYPE_ID
WHERE A.ACCOUNT_TYPE_ID LIKE '%SA%'
AND EXTRACT (MONTH FROM OPEN_DAY) = 12 AND EXTRACT (YEAR FROM OPEN_DAY) =
2021)
```

## THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

**INTERSECT**

```
(SELECT ACCOUNT_TYPE_ID, ACCOUNT_TYPE_NAME  
FROM CN2.ACCOUNT_TYPE@cn2_link  
WHERE ACCOUNT_TYPE_ID LIKE '%SA%')
```

**MINUS**

```
SELECT A.ACCOUNT_TYPE_ID, ACCOUNT_TYPE_NAME  
FROM CN2.ACCOUNT@cn2_link A JOIN CN2.ACCOUNT_TYPE@cn2_link AT ON  
A.ACCOUNT_TYPE_ID = AT.ACCOUNT_TYPE_ID  
WHERE A.ACCOUNT_TYPE_ID LIKE '%SA%'  
AND EXTRACT(MONTH FROM OPEN_DAY) = 12 AND EXTRACT(YEAR FROM OPEN_DAY) =  
2021);
```

## YÊU CẦU 2: VIẾT HÀM, THỦ TỤC VÀ RÀNG BUỘC TOÀN VỆN

### 2.1. FUNCTION

*Function tính phí chuyển tiền dựa vào **BANK\_ID** và **AMOUNT***

```
CREATE OR REPLACE FUNCTION IDENTIFIEDTRANSFERTYPE(V_AMOUNT
TRANSFER_DETAIL.AMOUNT%TYPE,
V_BANK_ID TRANSFER_DETAIL.BANK_ID%TYPE ) RETURN VARCHAR2
AS
    V_TRANSFER_FEE NUMBER;
    V_TRANSFER_TYPE TRANSACTION_TYPE.TRANSACTION_TYPE_ID%TYPE;
BEGIN
    -- Chuyển tiền nội bộ
    IF(V_BANK_ID = 'ACB') THEN
        IF(V_AMOUNT < 20000000) THEN
            V_TRANSFER_TYPE := 'CT01';
        ELSIF(V_AMOUNT < 5000000) THEN
            V_TRANSFER_TYPE := 'CT02';
        ELSE
            V_TRANSFER_TYPE := 'CT03';
        END IF;
    -- Chuyển tiền liên ngân hàng
    ELSE
        IF(V_AMOUNT < 20000000) THEN
            V_TRANSFER_TYPE := 'CT04';
        ELSIF(V_AMOUNT < 5000000) THEN
            V_TRANSFER_TYPE := 'CT05';
        ELSE
            V_TRANSFER_TYPE := 'CT06';
        END IF;
    END IF;

    RETURN V_TRANSFER_TYPE;
END;
```

## 2.1. PROCEDURE

### *Procedure thực hiện giao dịch chuyển tiền trong ngân hàng*

```
CREATE OR REPLACE PROCEDURE CN2.TRANSFER(V_SENDER_ACCOUNT
ACCOUNT.ACCOUNT_ID%TYPE,

V_RECEIVER_ACCOUNT ACCOUNT.ACCOUNT_ID%TYPE,

V_BANK_ID TRANSFER_DETAIL.BANK_ID%TYPE,

V_AMOUNT TRANSFER_DETAIL.AMOUNT%TYPE,

V_CONTENT TRANSFER_DETAIL.CONTENT%TYPE,

V_TRANSACTION_ID OUT NUMBER)

AS

    R_SENDER_ACCOUNT ACCOUNT%ROWTYPE;

    R_RECEIVER_ACCOUNT ACCOUNT%ROWTYPE;

    V_TRANSACTION_TYPE TRANSACTION.TRANSACTION_TYPE_ID%TYPE;

    V_TOTAL_FEE NUMBER;

    V_TRANSACTION_FEE NUMBER;

    V_NEXT_TRANSACTIONID TRANSACTION.TRANSACTION_ID%TYPE;

    V_TRANSACTIONID_FOR_RECEIVER TRANSACTION.TRANSACTION_ID%TYPE;

    DEM int;

BEGIN

    -- Lay thông tin người gửi

    SELECT * INTO R_SENDER_ACCOUNT

    FROM CN2.ACCOUNT

    WHERE ACCOUNT_ID = V_SENDER_ACCOUNT;

    -- Lay thông tin người nhận
```

## THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

```
select COUNT (ACCOUNT_ID) INTO DEM

FROM CN2.ACCOUNT

WHERE ACCOUNT_ID = V_RECEIVER_ACCOUNT;

IF (DEM != 0) THEN

    SELECT * INTO R_RECEIVER_ACCOUNT

    FROM CN2.ACCOUNT

    WHERE ACCOUNT_ID = V_RECEIVER_ACCOUNT;

ELSE

    SELECT * INTO R_RECEIVER_ACCOUNT

    FROM CN1.ACCOUNT@cn1_link

    WHERE ACCOUNT_ID = V_RECEIVER_ACCOUNT;

END IF;

IF (R_SENDER_ACCOUNT.CURRENT_BALANCE >= V_AMOUNT) THEN

    --Xac dinh loai giao dich (chuyen tien) de tinh phi chuyen tien

    V_TRANSFER_TYPE := IDENTIFIEDTRANSFERTYPE (V_AMOUNT, V_BANK_ID);

    --Tinh tong phi chuyen tien

    SELECT FEE INTO V_TRANSFER_FEE

    FROM CN2.TRANSACTION_TYPE

    WHERE TRANSACTION_TYPE_ID = V_TRANSFER_TYPE;

    V_TOTAL_FEE := V_AMOUNT + V_TRANSFER_FEE;

    --Tru tien nguoi gui

    UPDATE CN2.ACCOUNT

    SET CURRENT_BALANCE = CURRENT_BALANCE - V_TOTAL_FEE
```

## THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

```
WHERE ACCOUNT_ID = V_SENDER_ACCOUNT;

--Cong tien nguoi nhan

IF(DEM != 0) THEN

    UPDATE CN2.ACCOUNT

    SET CURRENT_BALANCE = CURRENT_BALANCE + V_AMOUNT

    WHERE ACCOUNT_ID = V_RECEIVER_ACCOUNT;

ELSE

    UPDATE CN1.ACCOUNT@cn1_link

    SET CURRENT_BALANCE = CURRENT_BALANCE+ V_AMOUNT

    WHERE ACCOUNT_ID = V_RECEIVER_ACCOUNT;

END IF;

--Them giao dich vua thuc hien vao bang lich su giao dich

V_NEXT_TRANSACTIONID := TRANSACTION_ID_SEQUENCE.NEXTVAL;

V_TRANSACTIONID_FOR_RECEIVER := TRANSACTION_ID_SEQUENCE.NEXTVAL;

V_TRANSACTION_ID := V_NEXT_TRANSACTIONID; -- gan transactionID cho
bien OUT de lap bien lai chuyen tien

--Thong tin giao dich nguoi gui

INSERT INTO CN2.TRANSACTION VALUES (V_NEXT_TRANSACTIONID,
V_TRANSFER_TYPE, SYSDATE, V_TOTAL_FEE, V_SENDER_ACCOUNT);

--Thong tin giao dich nguoi nhan

IF(DEM != 0) THEN

    INSERT INTO CN2.TRANSACTION VALUES (V_TRANSACTIONID_FOR_RECEIVER,
'NT01', SYSDATE, V_AMOUNT, V_RECEIVER_ACCOUNT);

ELSE
```

## THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

```
        INSERT INTO CN1.TRANSACTION@cn1_link
VALUES (V_TRANSACTIONID_FOR_RECEIVER, 'NT01', SYSDATE, V_AMOUNT,
V_RECEIVER_ACCOUNT);

        END IF;

        --Them cac thong tin khac vao bang transfer_detail

        INSERT INTO CN2.TRANSFER_DETAIL
VALUES (TRANSFER_DETAILS_ID_SEQUENCE.NEXTVAL, V_SENDER_ACCOUNT,
V_RECEIVER_ACCOUNT, V_BANK_ID, V_AMOUNT, V_CONTENT, V_NEXT_TRANSACTIONID);

        ELSE

            RAISE_APPLICATION_ERROR(-20000, 'So du hien tai khong du de thuc
hien chuyen tien');

        END IF;

        DBMS_OUTPUT.PUT_LINE('Chuyen tien thanh cong!');

        EXCEPTION

            WHEN NO_DATA_FOUND THEN

                RAISE_APPLICATION_ERROR(-20001, 'Thong tin chuyen tien khong hop
le!');
END;
```

***Procedure in ra thông tin khách hàng và số lần thực hiện giao dịch trong tháng.***

```
CREATE OR REPLACE PROCEDURE VIEW_CUSTOMER_INFORMATION(V_CUSTOMER_ID
CUSTOMER_BD.CUSTOMER_ID%TYPE,

V_MONTH NUMBER, V_YEAR NUMBER)

AS

    TYPE r_customer_info

    IS
```



## THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

```
RECORD

(

    full_name VARCHAR(50),

    gender CUSTOMER_CSD.GENDER%TYPE,

    date_of_birth CUSTOMER_CSD.DATE_OF_BIRTH%TYPE,

    id_card CUSTOMER_BD.ID_CARD%TYPE,

    payment_account_id ACCOUNT.ACCOUNT_ID%TYPE,

    current_balance ACCOUNT.CURRENT_BALANCE%TYPE,

    number_of_transactions NUMBER);

-- declare record

r_customer r_customer_info;

DEM INT;

BEGIN

    select COUNT(*) INTO DEM

    FROM CN2.ACCOUNT

    WHERE CUSTOMER_ID = V_CUSTOMER_ID;

    IF (DEM != 0) THEN

        SELECT CBD.FIRST_NAME || ' ' || CBD.LAST_NAME, CCSD.GENDER,
        CCSD.DATE_OF_BIRTH, CBD.ID_CARD, A.ACCOUNT_ID, CURRENT_BALANCE,
        COUNT(T.TRANSACTION_ID)

        INTO r_customer

        FROM ((CN2.CUSTOMER_BD CBD JOIN CN2.CUSTOMER_CSD CCSD ON
        CCSD.CUSTOMER_ID = CBD.CUSTOMER_ID)

        JOIN CN2.ACCOUNT A ON CBD.CUSTOMER_ID = A.CUSTOMER_ID)

        JOIN CN2.TRANSACTION T ON T.ACCOUNT_ID = A.ACCOUNT_ID

        WHERE CBD.CUSTOMER_ID = V_CUSTOMER_ID
```

## THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

```
AND EXTRACT (MONTH FROM TRANSACTION_DATE) = V_MONTH

AND EXTRACT (YEAR FROM TRANSACTION_DATE) = V_YEAR

GROUP BY CBD.FIRST_NAME || ' ' || CBD.LAST_NAME, GENDER,
DATE_OF_BIRTH, ID_CARD, A.ACCOUNT_ID, CURRENT_BALANCE;

ELSE

SELECT CBD.FIRST_NAME || ' ' || CBD.LAST_NAME, GENDER,
DATE_OF_BIRTH, ID_CARD, A.ACCOUNT_ID, CURRENT_BALANCE,
COUNT (T.TRANSACTION_ID)

INTO r_customer

FROM ((CN1.CUSTOMER_BD@cn1_link CBD JOIN CN1.CUSTOMER_CSD@cn1_link
CCSD ON CCSD.CUSTOMER_ID = CBD.CUSTOMER_ID)

JOIN CN1.ACCOUNT@cn1_link A ON CBD.CUSTOMER_ID = A.CUSTOMER_ID)

JOIN CN1.TRANSACTION@cn1_link T ON T.ACCOUNT_ID =
A.ACCOUNT_ID

WHERE CBD.CUSTOMER_ID = V_CUSTOMER_ID

AND EXTRACT (MONTH FROM TRANSACTION_DATE) = V_MONTH

AND EXTRACT (YEAR FROM TRANSACTION_DATE) = V_YEAR

GROUP BY CBD.FIRST_NAME || ' ' || CBD.LAST_NAME, GENDER,
DATE_OF_BIRTH, ID_CARD, A.ACCOUNT_ID, CURRENT_BALANCE;

END IF;

DBMS_OUTPUT.PUT_LINE('Thong tin khách hàng có mã KH ' || V_CUSTOMER_ID
|| ' là:');

DBMS_OUTPUT.PUT_LINE('**Ho và tên: ' || r_customer.full_name);

DBMS_OUTPUT.PUT_LINE('**Giới tính: ' || r_customer.gender);

DBMS_OUTPUT.PUT_LINE('**Ngày sinh: ' ||
to_char(r_customer.date_of_birth, 'DD-MM-YYYY'));

DBMS_OUTPUT.PUT_LINE('**Số CMND: ' || r_customer.id_card);
```

## THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

```
DBMS_OUTPUT.PUT_LINE('**So tai khoan: ' ||  
r_customer.payment_account_id);  
  
DBMS_OUTPUT.PUT_LINE('**So du hien tai: ' ||  
TRIM(to_char(r_customer.current_balance, '9,999,999,999')) || ' VND');  
  
DBMS_OUTPUT.PUT_LINE('**So lan thuc hien giao dich trong thang ' ||  
V_MONTH || '/' || V_YEAR || ': ' || r_customer.number_of_transactions);  
  
EXCEPTION  
  
    WHEN NO_DATA_FOUND THEN  
  
        RAISE_APPLICATION_ERROR(-20008, 'Khong tim thay thong tin tu du  
lieu dau vao');  
END;
```

### *Procedure xóa toàn bộ thông tin của khách hàng*

```
CREATE OR REPLACE PROCEDURE CN2.DELETE_CUSTOMER(V_CUSTOMER_ID  
CUSTOMER_BD.CUSTOMER_ID%TYPE)  
  
AS  
  
    V_ACCOUNT_ID ACCOUNT.ACCOUNT_ID%TYPE;  
  
    DEM INT;  
  
BEGIN  
  
    SELECT COUNT(ACCOUNT_ID) INTO DEM  
  
    FROM CN2.ACCOUNT  
  
    WHERE CUSTOMER_ID = V_CUSTOMER_ID;  
  
    IF (DEM != 0) THEN  
  
        SELECT ACCOUNT_ID INTO V_ACCOUNT_ID
```

## THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

```
FROM CN2.ACCOUNT

WHERE ACCOUNT_TYPE_ID = 'PA' AND CUSTOMER_ID = V_CUSTOMER_ID;

DELETE FROM CN2.TRANSFER_DETAIL WHERE SENDER_ACCOUNT = V_ACCOUNT_ID
OR RECEIVER_ACCOUNT = V_ACCOUNT_ID;

DELETE FROM CN2.TRANSACTION WHERE ACCOUNT_ID = V_ACCOUNT_ID;

DELETE FROM CN2.ACCOUNT WHERE CUSTOMER_ID = V_CUSTOMER_ID;

DELETE FROM CN2.CUSTOMER_CSD WHERE CUSTOMER_ID = V_CUSTOMER_ID;

DELETE FROM CN2.CUSTOMER_BD WHERE CUSTOMER_ID = V_CUSTOMER_ID;

ELSE

SELECT ACCOUNT_ID INTO V_ACCOUNT_ID

FROM CN1.ACCOUNT@cn1_link

WHERE ACCOUNT_TYPE_ID = 'PA'

AND CUSTOMER_ID = V_CUSTOMER_ID;

DELETE FROM CN1.TRANSFER_DETAIL@cn1_link WHERE SENDER_ACCOUNT =
V_ACCOUNT_ID OR RECEIVER_ACCOUNT = V_ACCOUNT_ID;

DELETE FROM CN1.TRANSACTION@cn1_link WHERE ACCOUNT_ID =
V_ACCOUNT_ID;

DELETE FROM CN1.ACCOUNT@cn1_link WHERE CUSTOMER_ID = V_CUSTOMER_ID;

DELETE FROM CN1.CUSTOMER_CSD@cn1_link WHERE CUSTOMER_ID =
V_CUSTOMER_ID;

DELETE FROM CN1.CUSTOMER_BD@cn1_link WHERE CUSTOMER_ID =
V_CUSTOMER_ID;
```

## THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

```
END IF;

COMMIT;

END;
```

### 2.3. RÀNG BUỘC TOÀN VỆ

**Phát biểu ràng buộc:** Ngày sinh của khách hàng phải nhỏ hơn ngày mở tài khoản của khách hàng đó.

- Bối cảnh: CUSTOMER, ACCOUNT

- Nội dung:

$\forall a \in \text{ACCOUNT},$

$\exists c \in \text{CUSTOMER}: c.\text{CUSTOMER\_ID} = a.\text{CUSTOMER\_ID}$

$\wedge c.\text{DATE\_OF\_BIRTH} < a.\text{OPEN\_DAY}$

- Bảng tầm ảnh hưởng

	Thêm	Xóa	Sửa
<b>ACCOUNT</b>	+	-	+(OPEN_DAY, CUSTOMER_ID)
<b>CUSTOMER</b>	-	-	+(DATE_OF_BIRTH)

Trigger tại bảng CUSTOMER:

```
CREATE OR REPLACE TRIGGER TRG_DOB_OPENACCOUNT_CUSTOMER
AFTER UPDATE OF DATE_OF_BIRTH ON CUSTOMER
FOR EACH ROW
DECLARE
    V_OPEN_DATE ACCOUNT.OPEN_DAY%TYPE;
    V_ACCOUNT_ID ACCOUNT.ACCOUNT_ID%TYPE;

    CURSOR C_ACCOUNT IS
        SELECT ACCOUNT_ID
        FROM ACCOUNT
        WHERE CUSTOMER_ID = :NEW.CUSTOMER_ID;
BEGIN
```

## THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

```
OPEN C_ACCOUNT;
LOOP
    FETCH C_ACCOUNT INTO V_ACCOUNT_ID;
    EXIT WHEN C_ACCOUNT%NOTFOUND;

    SELECT OPEN_DAY INTO V_OPEN_DATE
    FROM ACCOUNT
    WHERE ACCOUNT_ID = V_ACCOUNT_ID;

    IF (:NEW.DATE_OF_BIRTH > V_OPEN_DATE) THEN
        RAISE_APPLICATION_ERROR(-20000, 'Ngày sinh KH phải be hơn ngày
mo TK');
    END IF;
END LOOP;
CLOSE C_ACCOUNT;
DBMS_OUTPUT.PUT_LINE('Update thành công');
END;
```

Trigger tại bảng ACCOUNT:

```
CREATE OR REPLACE TRIGGER TRG_DOB_OPENACCOUNT_ACCOUNT
AFTER INSERT OR UPDATE OF OPEN_DAY ON ACCOUNT
FOR EACH ROW
DECLARE
    V_DATE_OF_BIRTH CUSTOMER.DATE_OF_BIRTH%TYPE;
BEGIN
    SELECT DATE_OF_BIRTH INTO V_DATE_OF_BIRTH
    FROM CUSTOMER
    WHERE CUSTOMER_ID = :NEW.CUSTOMER_ID;

    IF (V_DATE_OF_BIRTH > :NEW.OPEN_DAY)
    THEN
        RAISE_APPLICATION_ERROR(-20090, 'Ngày sinh KH phải be hơn ngày mo
TK. ');
    END IF;
END;
```

## **YÊU CẦU 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 "227 Nguyễn Văn Cừ, Q5, TpHCM".

Tại t1, nhân viên A cập nhật địa chỉ cho khách hàng C là "14/5 Trần Quốc Thảo, Q3".

Tại t2, nhân viên B cũng cập nhật địa chỉ cho khách hàng C là "200/11 Nguyễn Trãi, Q5".

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 nhân viên A cập nhật sẽ bị mất.

Vậy tại t5, địa chỉ của khách hàng C là "200/11 Nguyễn Trãi, Q5".

#### **3.1.2. Minh họa**

Time	Transaction T1 (Employee A) - CN02	T1 output	Transaction T2 (Employee B) - CN01	T2 output
t0	SELECT ADDRESS FROM CN2.CUSTOMER WHERE CUSTOMER_ID = 10952010;	227 Nguyen Van Cu, Q5, TPHCM	SELECT ADDRESS FROM CN2.CUSTOMER@cn2_link WHERE CUSTOMER_ID = 10952010;	227 Nguyen Van Cu, Q5, TPHCM
t1	UPDATE CN2.CUSTOMER SET Address ='14/5 Tran Quoc Thao, Q3' WHERE CUSTOMER_ID = 10952010;	1 row updated.		
t2			UPDATE CN2.CUSTOMER@cn2_link SET Address ='200/11	

## THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

			Nguyen Trai, Q5, TpHCM' WHERE CUSTOMER_ID = 10952010;	
t3	COMMIT;	Commit complete.		1 row updated.
t4			COMMIT;	Commit complete.
t5	SELECT ADDRESS FROM CN2.CUSTOMER WHERE CUSTOMER_ID = 10952010;	200/11 Nguyen Trai, Q5, TpHCM	SELECT ADDRESS FROM CN2.CUSTOMER@cn2_link WHERE CUSTOMER_ID = 10952010;	200/11 Nguyen Trai, 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 sổ 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 khách hàng B. Tại thời điểm này, số dư trong tài khoản của khách hàng B là 22.982.000 đồng và số lần giao dịch vào tháng 6 năm 2021 là 8.

Tại t1, khách hàng B mở tài khoản tiết kiệm có kỳ hạn với số tiền gửi là 10.000.000 VND.



## THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

Tại t2, khách hàng B thực hiện giao dịch.

Tại t3, nhân viên A xem xét lại thông tin của khách hàng B và nhận thấy rằng số dư trong tài khoản của khách hàng B giảm 10.000.000 đồng. 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 T2 (Customer B)	T2 output
t0	SET SERVEROUTPUT ON;  BEGIN CN2.VIEW_CUSTO MER_INFORMATIO N(10952010, 11, 2021); END;  /	Thông tin khách hàng có mã KH 10952010 là: **Họ và tên: Tam Nguyen Van **Giới tính: Nam **Ngày sinh: 08- 01-1989 **Số CMND: 272443521 **Số tài khoản: 2012010 **Số dư hiện tại: <b>13,000,000</b> <b>VND</b> **Số lần thực hiện giao dịch		

## THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

		trong tháng 11/2021: 6 PL/SQL procedure successfully completed.		
t1			DECLARE V_TRANSACTION_ID NUMBER; BEGIN CN2.TRANSFER@cn2_link (2012010, 2012007, 'VCB', 500000, 'Thuc hien chuyen tien cho tai khoan 2012006', V_TRANSACTION_ID); END;  /	PL/SQL procedure successfully completed.
t2			COMMIT;	Commit complete.
t3	BEGIN CN2.VIEW_CUSTO MER_INFORMATIO N(10952010, 11, 2021); END;  /	Thông tin khách hàng có mã KH 10952010 là: **Họ và tên: Tam Nguyen Van **Giới tính: Nam **Ngày sinh: 08- 01-1989 **Số CMND: 272443521		

## THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

		<p><b>**Số tài khoản:</b> 2012010</p> <p><b>**Số dư hiện tại:</b> <b>12,493,000 VND</b></p> <p><b>**Số lần thực hiện giao dịch trong tháng 11/2021:</b> 7</p> <p>PL/SQL procedure successfully completed.</p>		
--	--	---	--	--

### 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 khách hàng C.

Tại t2, nhân viên B xóa thông tin của khách hàng đó.

Tại t3, nhân viên A thử đọc thông tin của khách hàng 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.

## THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

### 3.4.2. Minh họa

Time	Transaction T1 (Employee A)	T1 output	Transaction T2 (Employee B)	T2 output
t0	SET SERVEROUTPUT ON;  BEGIN CN2.VIEW_CUST OMER_INFORMAT ION(10952008, 11, 2021); END;  /	Thông tin khách hàng có ma KH 10952008 là:  **Ho và ten: Huong Nguyen Thi  **Gioi tinh: Nu  **Ngày sinh: 10-11-1991  **So CMND: 272443521  **So tai khoan: 2012008  **So du hien tai: 50,000,000 VND  **So lan thuc hien giao dich trong thang 11/2021: 4  PL/SQL procedure successfully completed.		
t1			BEGIN CN2.DELETE_CUSTOM ER@cn2_link(10952 008); END;  /	PL/SQL procedure successfu lly completed .
t2			COMMIT;	Commit complete.
t3	BEGIN CN2.VIEW_CUST OMER_INFORMAT ION(10952008, 11, 2021); END;  /	Error report - ORA-20008: Ma khách hàng không hợp lệ ORA-06512: at "C##USSER1.VIEW_CUSTOMER _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 E1 cập nhật trạng thái của tài khoản A1.

Tại t1, nhân viên E2 cập nhật trạng thái của tài khoản A2.

Tại t2, nhân viên E1 cập nhật trạng thái của tài khoản A2.

Tại t3, nhân viên E2 cập nhật trạng thái của tài khoản A1. Và bế tắc xảy ra.

Nguyên nhân: Transaction T1 giữ khóa đơn vị dữ liệu của E1, chờ khóa đơn vị dữ liệu của E2 và Transaction T2 giữ khóa đơn vị dữ liệu của E2 và chờ khóa đơn vị dữ liệu của E2 của nhà cung cấp E1. Hai giao dịch chờ khóa vô hạn gây ra trạng thái deadlock.

### 3.5.2. Minh họa

Time	Transaction T1 (Employee E1) – CN2	T1 output	Transaction T2 (Employee E2) – CN1	T2 output
t0	UPDATE CN2.ACCOUNT SET STATUS = 'Khoa' WHERE ACCOUNT_ID = 2012010;	1 row updated.		
t1			UPDATE CN2.ACCOUNT@cn2_1 link SET STATUS = 'Khoa' WHERE ACCOUNT_ID = 2012008;	1 row updated.
t2	UPDATE CN2.ACCOUNT SET STATUS = 'Khoa'			

## THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

	WHERE ACCOUNT_ID = 2012008;			
t3			UPDATE CN2.ACCOUNT@cn2_1 ink SET STATUS = 'Khoa' WHERE ACCOUNT_ID = 2012010;	
t4		Error report - 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.

### YÊU CẦU 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 khách hàng (CUSTOMER\_ID, FIRST\_NAME, LAST\_NAME, GENDER, PHONE\_NUMBER, ACC.ACCOUNT\_ID, ACC.CURRENT\_BALANCE) ở chi nhánh 'CN Tien Giang' có loại tài khoản là 'Tai khoan thanh toan' đã từng thực hiện giao dịch có tổng giá trị lớn hơn 1.000.0000đ.

```
SELECT DISTINCT CUS.CUSTOMER_ID, CUS.FIRST_NAME, CUS.LAST_NAME, CUS.GENDER,  
CUS.PHONE_NUMBER, ACC.ACCOUNT_ID, ACC.CURRENT_BALANCE  
FROM CUSTOMER CUS, BRANCH BR, ACCOUNT ACC,  
ACCOUNT_TYPE ACCT, TRANSACTION TRA  
WHERE BR.BRANCH_ID = CUS.BRANCH_ID AND  
CUS.CUSTOMER_ID = ACC.CUSTOMER_ID AND  
ACC.ACCOUNT_TYPE_ID = ACCT.ACCOUNT_TYPE_ID AND  
ACC.ACCOUNT_ID = TRA.ACCOUNT_ID AND  
BR.BRANCH_NAME = 'CN Tien Giang' AND  
ACCT.ACCOUNT_TYPE_NAME = 'Tai khoan thanh toan'  
AND TRA.TOTAL_TRANSACTION_AMOUNT > 1000000;
```

# THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

## 4.2. EXPLAIN QUERY câu truy vấn đơn giản

```

SELECT /*+ GATHER_PLAN_STATISTICS */ DISTINCT CUS.CUSTOMER_ID,
CUS.FIRST_NAME, CUS.LAST_NAME, CUS.GENDER, CUS.PHONE_NUMBER, ACC.ACCOUNT_ID,
ACC.CURRENT_BALANCE

FROM CUSTOMER CUS, BRANCH BR, ACCOUNT ACC,

ACCOUNT_TYPE ACCT, TRANSACTION TRA

WHERE BR.BRANCH_ID = CUS.BRANCH_ID AND

CUS.CUSTOMER_ID = ACC.CUSTOMER_ID AND

ACC.ACCOUNT_TYPE_ID = ACCT.ACCOUNT_TYPE_ID AND

ACC.ACCOUNT_ID = TRA.ACCOUNT_ID AND

BR.BRANCH_NAME = 'CN Tien Giang' AND

ACCT.ACCOUNT_TYPE_NAME = 'Tai khoan thanh toan'

AND TRA.TOTAL_TRANSACTION_AMOUNT > 1000000;

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

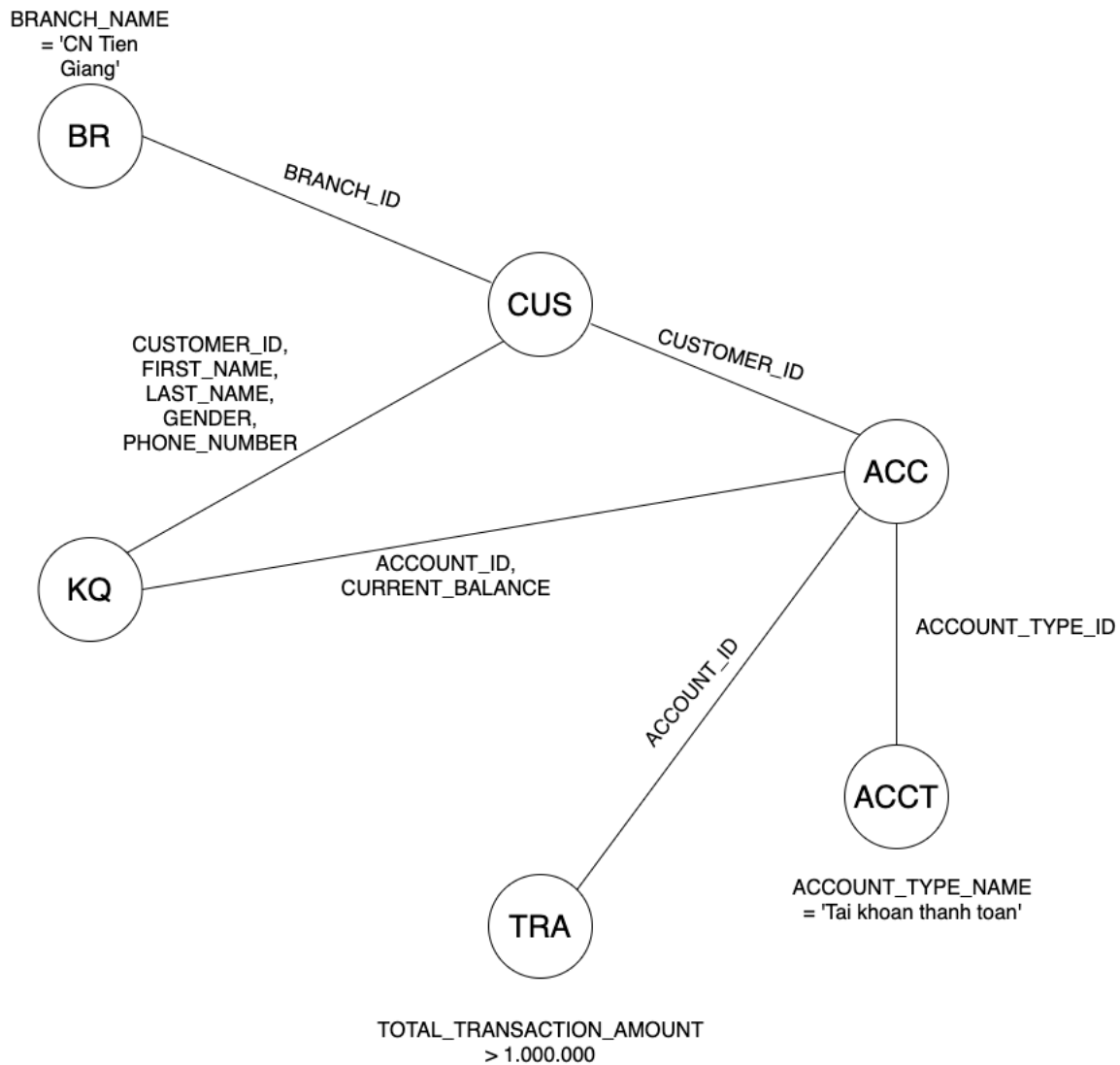
```

## Kết quả

Id	Operation	Name	Starts	E-Rows	A-Rows	A-Time	Buffers	Reads	OMem	lMem	Used-Mem
0	SELECT STATEMENT		1		6	00:00:00.29	23	26			
1	HASH UNIQUE		1	2	6	00:00:00.29	23	26	969K	969K	996K (0)
2	HASH JOIN SEMI		1	2	6	00:00:00.29	23	26	962K	962K	446K (0)
3	HASH JOIN SEMI		1	2	6	00:00:00.20	20	20	955K	955K	916K (0)
4	HASH JOIN		1	4	9	00:00:00.16	14	14	1265K	1265K	1150K (0)
5	MERGE JOIN		1	4	10	00:00:00.14	8	8			
6	TABLE ACCESS BY INDEX ROWID	ACCOUNT_TYPE	1	1	1	00:00:00.10	2	2			
7	INDEX FULL SCAN	ACCOUNTTYPE_ID_FK	1	7	7	00:00:00.08	1	1			
8	SORT JOIN		1	12	10	00:00:00.05	6	6	2048	2048	2048 (0)
9	TABLE ACCESS FULL	ACCOUNT	1	12	12	00:00:00.05	6	6			
10	TABLE ACCESS FULL	CUSTOMER	1	9	9	00:00:00.02	6	6			
11	TABLE ACCESS FULL	TRANSACTION	1	15	15	00:00:00.03	6	6			
12	TABLE ACCESS FULL	BRANCH	1	1	1	00:00:00.09	3	6			

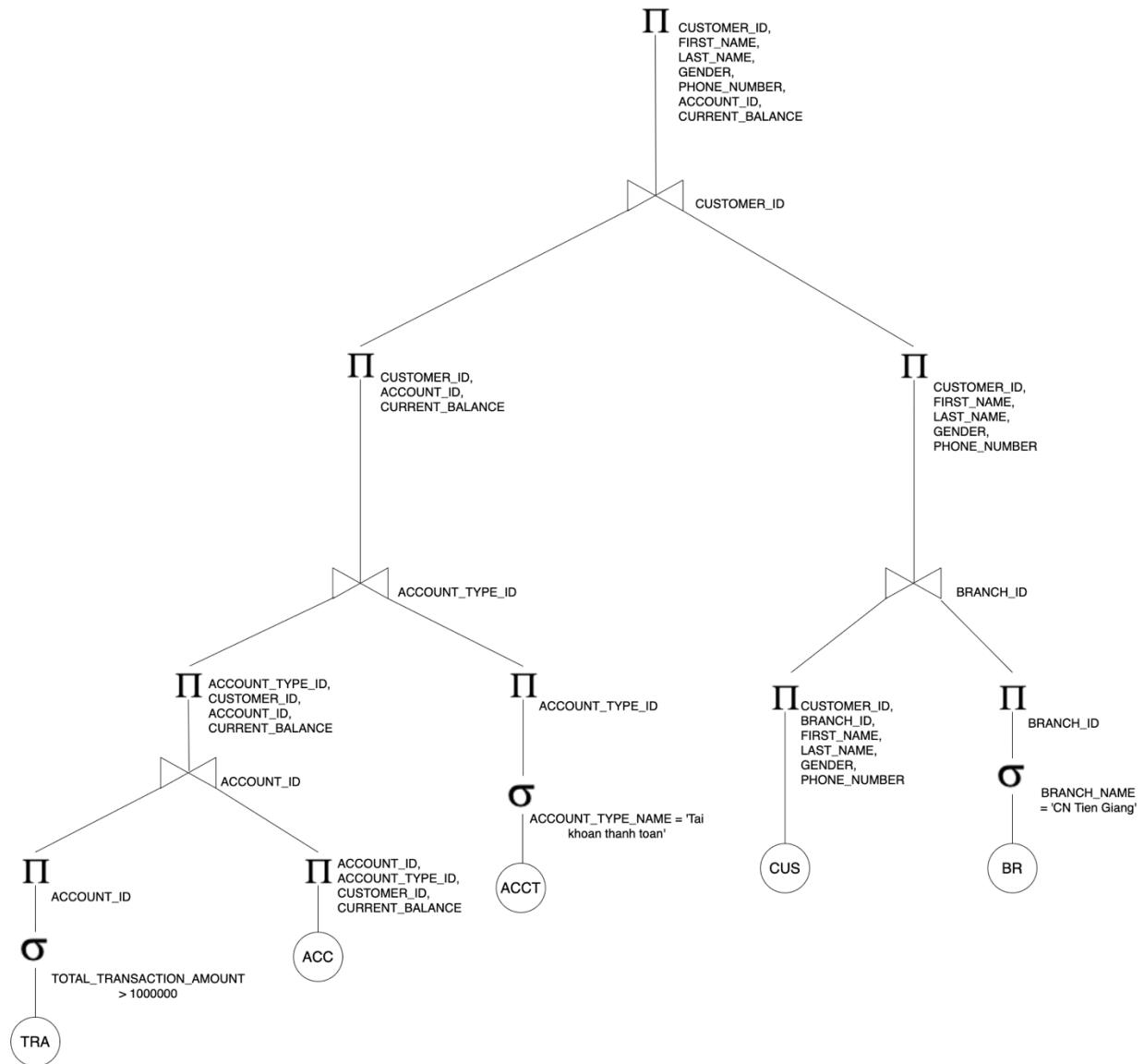


### 4.3. Tối ưu hóa câu truy vấn cục bộ, phân tán



Hình 1. Đồ thị truy vấn

# THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN



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

## THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

### Câu truy vấn tối ưu trên môi trường tập trung:

```
SELECT DISTINCT E.CUSTOMER_ID, FIRST_NAME, LAST_NAME, GENDER, PHONE_NUMBER,
ACCOUNT_ID, CURRENT_BALANCE
FROM ((SELECT CUSTOMER_ID, ACCOUNT_ID, CURRENT_BALANCE
      FROM ((SELECT CUSTOMER_ID, A.ACCOUNT_ID, CURRENT_BALANCE, ACCOUNT_TYPE_ID
            FROM ((SELECT ACCOUNT_ID FROM TRANSACTION WHERE
TOTAL_TRANSACTION_AMOUNT > 1000000) A
            INNER JOIN (SELECT ACCOUNT_ID, ACCOUNT_TYPE_ID, CUSTOMER_ID,
CURRENT_BALANCE
                      FROM ACCOUNT) B ON A.ACCOUNT_ID = B.ACCOUNT_ID)) C
      INNER JOIN
      (SELECT ACCOUNT_TYPE_ID
       FROM ACCOUNT_TYPE WHERE ACCOUNT_TYPE_NAME = 'Tai khoan thanh
toan') D ON C.ACCOUNT_TYPE_ID = D.ACCOUNT_TYPE_ID)) E
      INNER JOIN (SELECT CUSTOMER_ID, FIRST_NAME, LAST_NAME,
GENDER, PHONE_NUMBER
                FROM ((SELECT CUSTOMER_ID, FIRST_NAME, LAST_NAME, GENDER,
PHONE_NUMBER, BRANCH_ID
                      FROM CUSTOMER) F
                INNER JOIN (SELECT BRANCH_ID
                          FROM BRANCH WHERE BRANCH_NAME = 'CN Tien
Giang') G ON F.BRANCH_ID = G.BRANCH_ID)) H ON E.CUSTOMER_ID =
H.CUSTOMER_ID);
```

### EXPLAIN QUERY câu truy vấn đã được tối ưu trên môi trường tập trung:

```
SELECT /*+ GATHER_PLAN_STATISTICS */ DISTINCT E.CUSTOMER_ID, FIRST_NAME,
LAST_NAME, GENDER, PHONE_NUMBER, ACCOUNT_ID, CURRENT_BALANCE
FROM ((SELECT CUSTOMER_ID, ACCOUNT_ID, CURRENT_BALANCE
      FROM ((SELECT CUSTOMER_ID, A.ACCOUNT_ID, CURRENT_BALANCE, ACCOUNT_TYPE_ID
            FROM ((SELECT ACCOUNT_ID FROM TRANSACTION WHERE
TOTAL_TRANSACTION_AMOUNT > 1000000) A
            INNER JOIN (SELECT ACCOUNT_ID, ACCOUNT_TYPE_ID, CUSTOMER_ID,
CURRENT_BALANCE
                      FROM ACCOUNT) B ON A.ACCOUNT_ID = B.ACCOUNT_ID)) C
      INNER JOIN
      (SELECT ACCOUNT_TYPE_ID
       FROM ACCOUNT_TYPE WHERE ACCOUNT_TYPE_NAME = 'Tai khoan thanh
toan') D ON C.ACCOUNT_TYPE_ID = D.ACCOUNT_TYPE_ID)) E
      INNER JOIN (SELECT CUSTOMER_ID, FIRST_NAME, LAST_NAME,
GENDER, PHONE_NUMBER
                FROM ((SELECT CUSTOMER_ID, FIRST_NAME, LAST_NAME, GENDER,
PHONE_NUMBER, BRANCH_ID
                      FROM CUSTOMER) F
                INNER JOIN (SELECT BRANCH_ID
                          FROM BRANCH WHERE BRANCH_NAME = 'CN Tien
Giang') G ON F.BRANCH_ID = G.BRANCH_ID)) H ON E.CUSTOMER_ID =
H.CUSTOMER_ID);
```

## THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

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

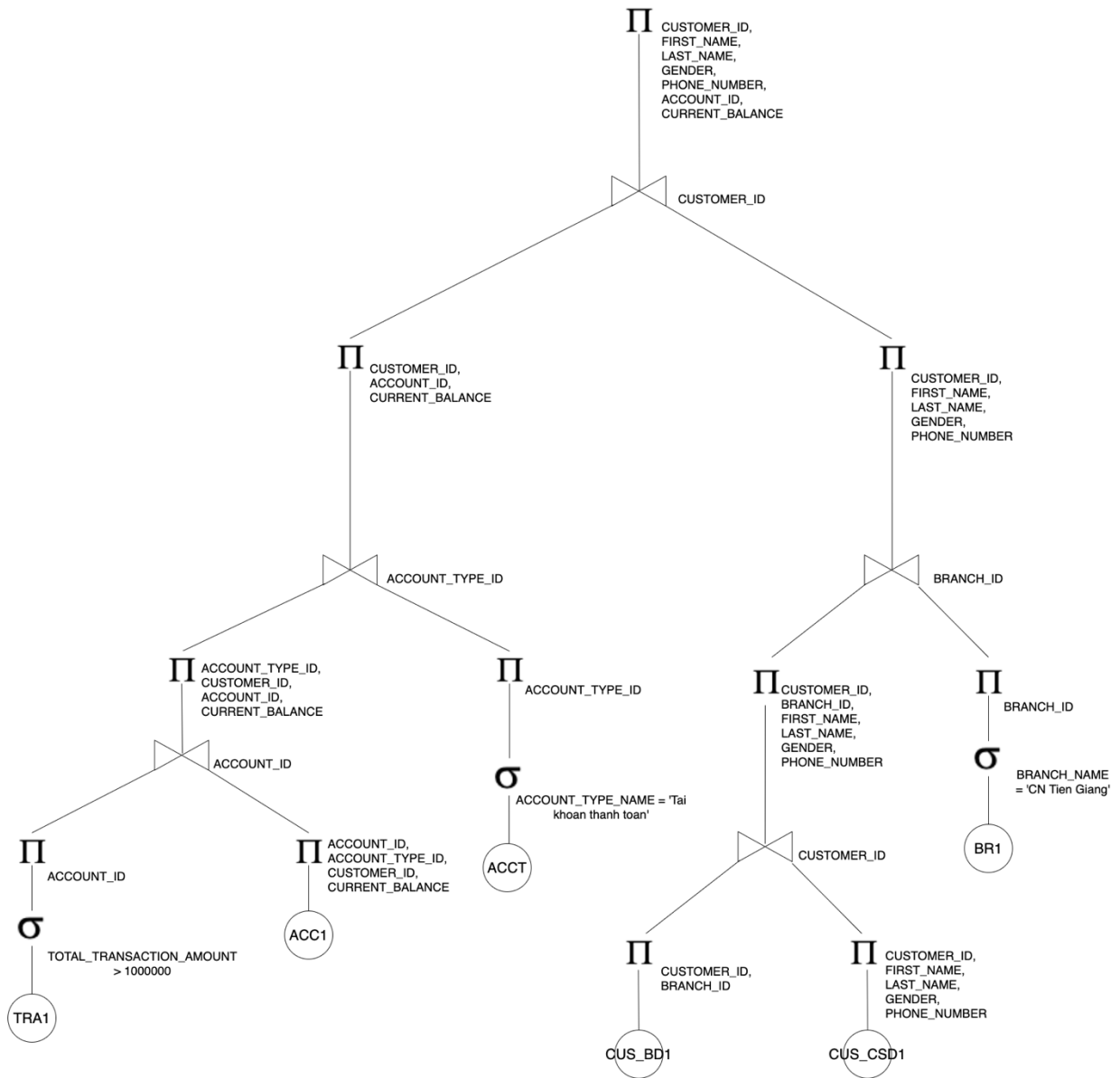
### Kết quả

15	-----											
16	Id	Operation	Name	Starts	E-Rows	A-Rows	A-Time	Buffers	Reads	OMem	lMem	Used-Mem
17	-----											
18	0	SELECT STATEMENT		1		15	00:00:00.04	30	30			
19	* 1	HASH JOIN		1	5	15	00:00:00.04	30	30	1209K	1209K	971K (0)
20	* 2	HASH JOIN		1	5	15	00:00:00.03	24	24	1476K	1476K	961K (0)
21	3	MERGE JOIN CARTESIAN		1	15	15	00:00:00.03	18	18			
22	4	MERGE JOIN CARTESIAN		1	1	1	00:00:00.03	12	12			
23	* 5	TABLE ACCESS FULL	BRANCH	1	1	1	00:00:00.02	6	6			
24	6	BUFFER SORT		1	1	1	00:00:00.01	6	6	2048	2048	2048 (0)
25	* 7	TABLE ACCESS FULL	ACCOUNT_TYPE	1	1	1	00:00:00.01	6	6			
26	8	BUFFER SORT		1	15	15	00:00:00.01	6	6	2048	2048	2048 (0)
27	* 9	TABLE ACCESS FULL	TRANSACTION	1	15	15	00:00:00.01	6	6			
28	10	TABLE ACCESS FULL	ACCOUNT	1	12	12	00:00:00.01	6	6			
29	11	TABLE ACCESS FULL	CUSTOMER	1	9	9	00:00:00.01	6	6			
30	-----											

**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

# THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN



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 E.CUSTOMER_ID, FIRST_NAME, LAST_NAME, GENDER, PHONE_NUMBER,
ACCOUNT_ID, CURRENT_BALANCE
FROM ((SELECT CUSTOMER_ID, ACCOUNT_ID, CURRENT_BALANCE
      FROM ((SELECT CUSTOMER_ID, A.ACCOUNT_ID, CURRENT_BALANCE, ACCOUNT_TYPE_ID
            FROM ((SELECT ACCOUNT_ID FROM CN1.TRANSACTION WHERE
TOTAL_TRANSACTION_AMOUNT > 1000000) A
            INNER JOIN (SELECT ACCOUNT_ID, ACCOUNT_TYPE_ID, CUSTOMER_ID,
CURRENT_BALANCE
                      FROM CN1.ACCOUNT) B ON A.ACCOUNT_ID = B.ACCOUNT_ID)) C
      INNER JOIN
      (SELECT ACCOUNT_TYPE_ID
      FROM ACCOUNT_TYPE WHERE ACCOUNT_TYPE_NAME = 'Tai khoan thanh
toan') D ON C.ACCOUNT_TYPE_ID = D.ACCOUNT_TYPE_ID)) E
      INNER JOIN (SELECT CUSTOMER_ID, FIRST_NAME, LAST_NAME,
GENDER, PHONE_NUMBER
                FROM ((SELECT X.CUSTOMER_ID, FIRST_NAME, LAST_NAME, GENDER,
PHONE_NUMBER, BRANCH_ID
                      FROM ((SELECT CUSTOMER_ID, BRANCH_ID FROM
CN1.CUSTOMER_BD) X JOIN (SELECT
CUSTOMER_ID, FIRST_NAME, LAST_NAME, GENDER, PHONE_NUMBER FROM CN1.CUSTOMER_CSD)
Y ON X.CUSTOMER_ID = Y.CUSTOMER_ID)) F
                INNER JOIN (SELECT BRANCH_ID
                          FROM CN1.BRANCH WHERE BRANCH_NAME = 'CN Tien
Giang') G ON F.BRANCH_ID = G.BRANCH_ID)) H ON E.CUSTOMER_ID =
H.CUSTOMER_ID);
```

### TÀI LIỆU THAM KHẢO

- [1] Phạm Quốc Hùng, Chu Hà Thảo Ngân, Thái Minh Triết, and Võ Tuấn Anh, "Bài thực hành số 3: Kiểm tra database link và truy vấn phân tán." Khoa Hệ thống 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](https://courses.uit.edu.vn/pluginfile.php/313931/mod_resource/content/1/10_KiemTra3.pdf)
- [2] 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.