**※ TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI ※**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ**

**TRUYỀN THÔNG**

**-------------------------------------------------------------------------**



**-------------------------------------------------------------------------**

**Báo Cáo Thực Hành Cơ Sở Dữ Liệu**

**Đề Tài: Xây dựng hệ thống quản trị ngân hàng**

**Giảng Viên Hướng Dẫn: TS.**Vũ Tuyết Trinh

**Sinh viên thực hiện:**   
Nguyễn Đắc Kiên - 20225871

Đặng Minh Trí - 20225939

1. **MÔ TẢ BÀI TOÁN**
   1. **Lời mở đầu**
   2. **Mục tiêu của dự án**
2. **THIẾT KẾ CƠ SỞ DỮ LIỆU**
   1. **Khảo sát dữ liệu**
   2. **Phân tích và thiết kế**
      1. **Xác định các thực thể**
      2. **Xây dựng mô hình thực thể liên kết**
      3. **Xây dụng mô hình quan hệ**
   3. **Thông tin các bảng và nhập dữ liệu**
      1. **Cách insert dữ liệu**
      2. **Mô tả bảng**
         1. **Bảng CaNhan**
         2. **Bảng NhanVien**
         3. **Bảng KieuNhanVien**
         4. **Bảng KhachHang**
         5. **Bảng TaiKhoan**
         6. **Bảng LoaiTaiKhoan**
         7. **Bảng ChiNhanh**
         8. **Bảng Giaodich**
         9. **Bảng KieuGD**
3. **CÁC CHỨC NĂNG THỰC HIỆN**
   1. Các câu truy vấn cơ bản
   2. Phân quyền
   3. Cập nhật lãi suất
   4. Cập nhật phí duy trì
   5. Gửi tiền
   6. Rút tiền
   7. Chuyển tiền
   8. Xóa tài khoản và giao dịch
   9. Xóa tài khoản
   10. Chuyển nhân viên
   11. Chuyển bộ phận
4. **ĐÁNH GIÁ**
5. **KẾT LUẬN, THU HOẠCH TỪ DỰ ÁN**

**LỜI MỞ ĐẦU**

Trong bối cảnh phát triển không ngừng của công nghệ thông tin, quản lý dữ liệu trở thành yếu tố then chốt giúp các tổ chức và doanh nghiệp tối ưu hóa hiệu quả hoạt động. Đặc biệt, trong lĩnh vực ngân hàng, việc quản lý và xử lý dữ liệu một cách hiệu quả không chỉ giúp cải thiện dịch vụ khách hàng mà còn đảm bảo tính an toàn và bảo mật của thông tin tài chính.với chi phí rẻ nhất chính là mục tiêu mà chúng em muốn hướng đến. Từ đó khách hàng có thể tin yêu và sử dụng ứng dụng của chúng em.

**Mục Tiêu Của Dự Án**

Dự án "Quản Lý Ngân Hàng Bằng SQL" được triển khai nhằm mục đích xây dựng một hệ thống quản lý dữ liệu ngân hàng sử dụng cơ sở dữ liệu SQL. Hệ thống này sẽ hỗ trợ các chức năng cơ bản như quản lý tài khoản khách hàng, giao dịch tài chính, quản lý nhân viên, thêm sửa xóa khách hàng, thay đổi chức vụ và chi nhánh làm việc của nhân viên, tự động cập nhật lãi suất cho khách hàng, phân quyền cho quản trị viên, nhân viên và khách hàng, cùng nhiều chức năng khác.

**2:THIẾT KẾ CƠ SỞ DỮ LIỆU**

**A.Khảo Sát Cơ Sở Dữ Liệu SQL về Ngân Hàng (Banking)**

**1. Khảo Sát Cơ Sở Dữ Liệu**

Khảo sát các dữ liệu cần thiết để lưu trữ trong cơ sở dữ liệu của một hệ thống ngân hàng, bao gồm thông tin khách hàng, tài khoản, giao dịch, nhân viên, chi nhánh.

**2. Các Dữ Liệu Cần Lưu Trữ**

**Thông Tin Khách Hàng:**

Mã khách hàng

Họ và tên

Ngày sinh

Chứng minh nhân dân

Giới tính

Số điện thoại

Địa chỉ

**Thông Tin Tài Khoản:**

Số tài khoản

Mã khách hàng

Số dư

Lãi suất(%)

Loại tài khoản(Tài khoản ngắn hạn, Tài khoản tiết kiệm)

Ngày mở tài khoản

**Thông Tin Giao Dịch:**

Mã giao dịch

Số tài khoản thực hiện

Số tài khoản nhận

Loại giao dịch (Ví dụ: Gửi tiền, Rút tiền, Chuyển khoản)

Số tiền

Ngày giao dịch

Mô tả giao dịch

**Thông Tin Nhân Viên:**

Mã nhân viên

Họ tên

Ngày sinh

Giới tính

Chứng minh nhân dân

Số điện thoại

Địa chỉ

Loại nhân viên(Nhân viên ngân hàng, Quản trị viên hệ thống)

Mã chi nhánh

**Thông Tin Chi Nhánh:**

Mã chi nhánh

Tên chi nhánh

Địa chỉ

**B. PHÂN TÍCH VÀ THIẾT KẾ**

**i. Xác định các thực thể**

CaNhan(uid, cmnd, dc, ho, ten, phai, sdt)

KhachHang(maKH, uid)

ChiNhanh(maCN, diachi, tenCN)

VaiTro(VaiTroid, TenVaiTro)

NhanVien(maNV, maCN, uid, VaiTroid)

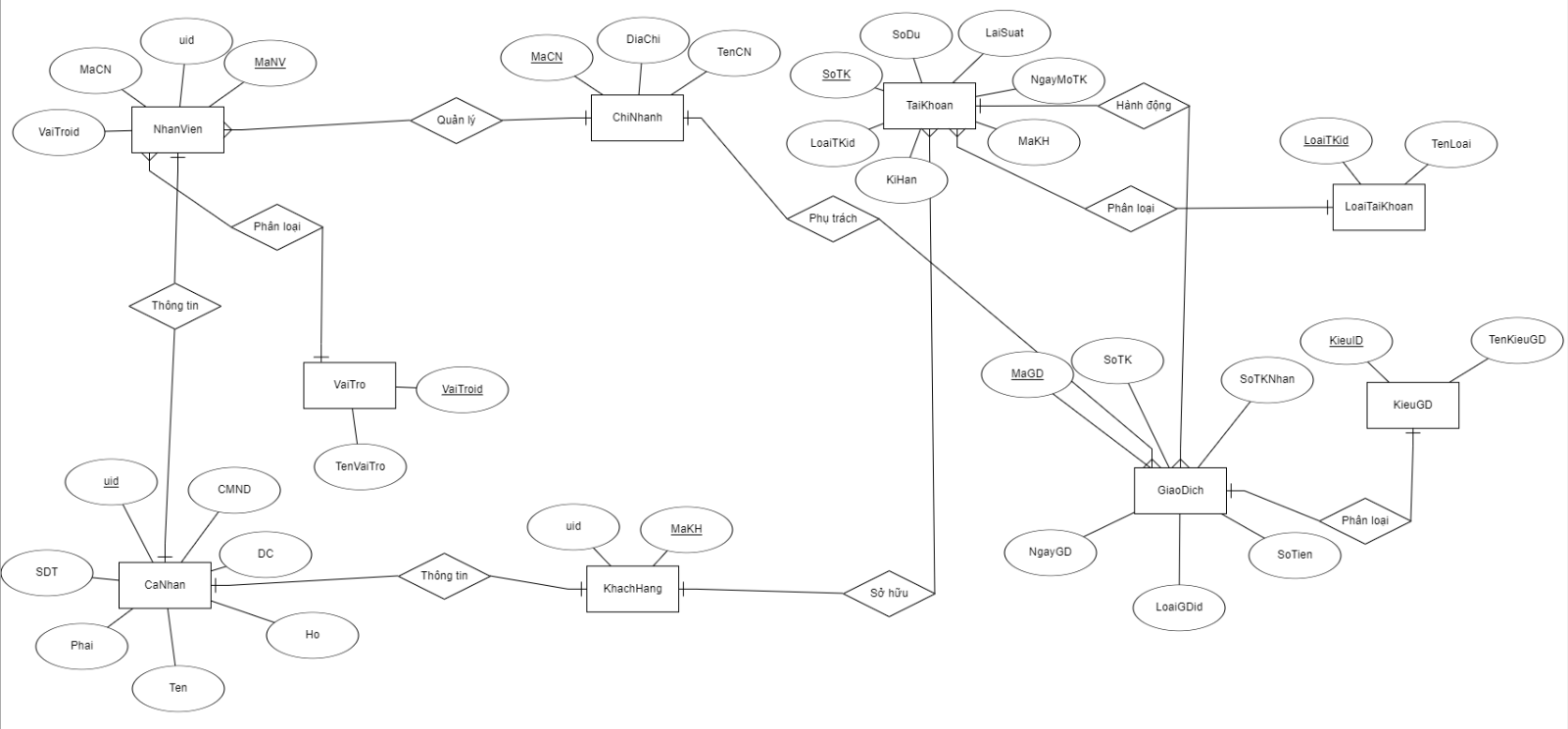
LoaiTaiKhoan(LoaiTKid, TenLoai)

TaiKhoan(SoTK, soDu, LaiSuat, ngaymoTK, MaKH, LoaiTKid)

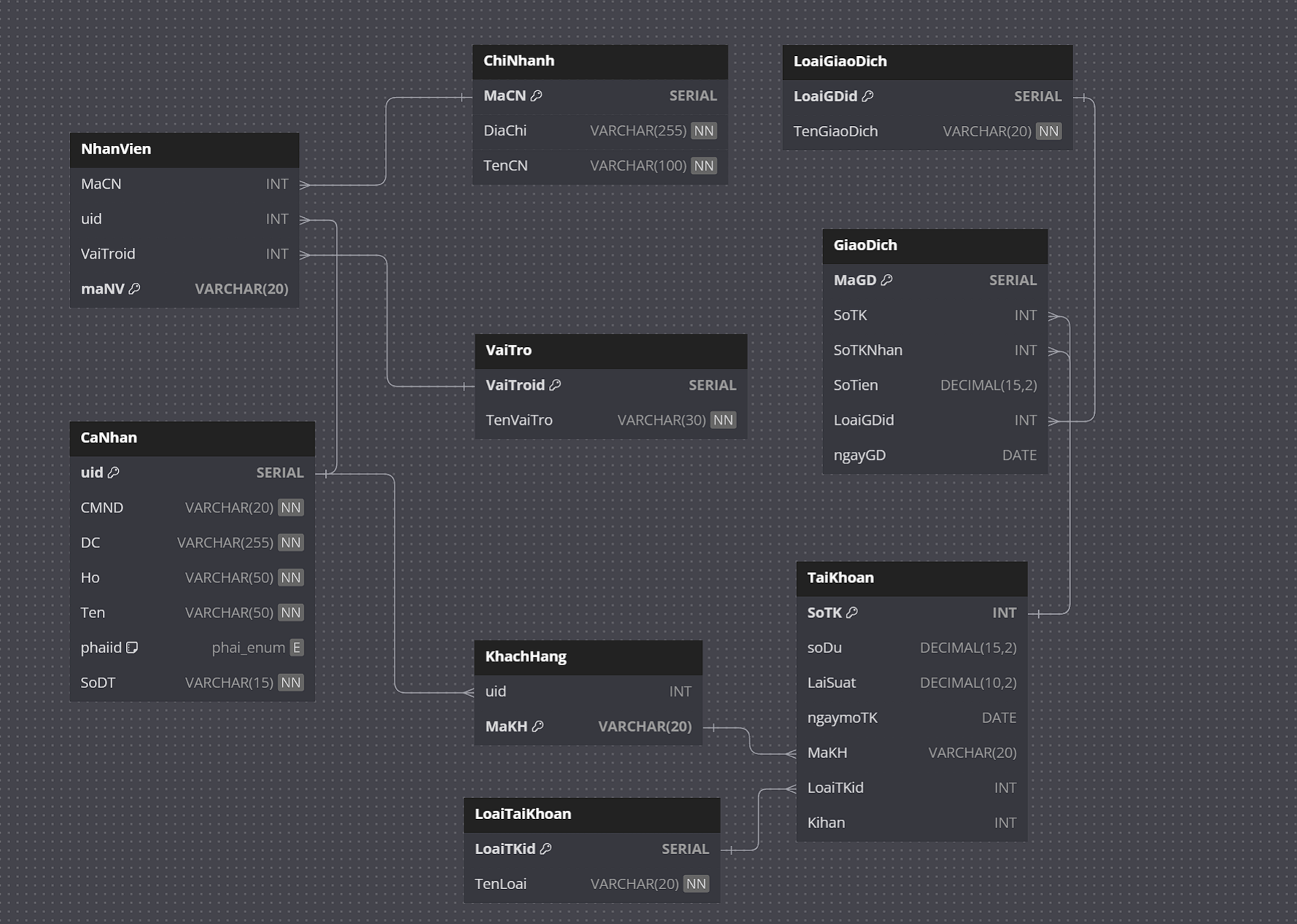
LoaiGiaoDich(LoaiGDid, TenGiaoDich)

GiaoDich(MaGD, SoTK, SoTKNhan, SoTien, LoaiGDid, ngayGD)

**ii. Xây dựng mô hình thực thể liên kết**



**iii. XÂY DỰNG MÔ HÌNH DỮ LIỆU QUAN HỆ**



**C. THÔNG TIN CÁC BẢNG VÀ NHẬP DỮ LIỆU**

**i. Cách insert dữ liệu**

Mỗi thành viên trong nhóm phụ trách việc tìm dữ liệu phù hợp với bảng được phân công.

Sau khi tìm được dữ liệu phù hợp sẽ tổng hợp và lưu trữ vào các file Exel.

Dữ liệu của thuộc tính DiaChi trong các bảng CaNhan,NhanVien,KhachHang được tạo ra bằng câu lệnh tạo dữ liệu ngẫu nhiên, các thuộc tính còn lại được từng thành viên tự khởi tạo

Tiếp theo đó sẽ thưc hiện việc rà soát và sửa lại dữ liệu để đảm bảo các thuộc tính khóa chính, unique không bị trùng lặp cũng như dữ liệu các khóa ngoài phải đảm bảo tính chính xác. Việc này sẽ thực hiện cho đến khi dữ liệu không còn sai sót.

Cuối cùng, từ dữ liệu của các bảng đã khởi tạo sẽ tạo ra các câu lệnh nhập dữ liệu và copy các câu lệnh này vào PostgreSQL để khởi tạo cơ sở dữ liệu.

**ii. Mô tả bảng**

**Bảng CaNhan**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên thuộc tính | Khóa chính | Khóa ngoài | Kiểu dữ liệu | Mô tả |
| Uid | Yes |  | Serial | Khóa chính |
| CMND |  |  | VarChar(20) | Xác định chứng minh nhân dân |
| DiaChi |  |  | VarChar(25) | Xác định địa chỉ |
| Ho |  |  | VarChar(50) | Xác định họ |
| Ten |  |  | VarChar(50) | Xác định tên |
| Phai |  |  | Varchar(50) | Xác định giới tính |
| SoDT |  |  | VarChar(50) | Xác định số điện thoại |

**Bảng NhanVien**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên thuộc tính | Khóa chính | Khóa ngoài | Kiểu dữ liệu | Mô tả |
| MaNV | Yes |  | VarChar(20) | Xác định mã nhân viên(khóa chính) |
| MaCN |  | Yes | Integer | Xác định nhân viên thuộc chi nhánh nào |
| Uid |  | Yes | Integer | Xác định id |
| VaiTroid |  | Yes | Integer | Xác định vai trò nhân viên |

**Bảng VaiTro**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên thuộc tính | Khóa chính | Khóa ngoài | Kiểu dữ liệu | Mô tả |
| VaiTroID | Yes |  | Serial | Xác định mã vai trò |
| TenVaiTro |  | Yes | VarChar(30) | Xác định tên vai trò |

**Bảng KhacHang**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên thuộc tính | Khóa chính | Khóa ngoài | Kiểu dữ liệu | Mô tả |
| MaKH | Yes |  | Serial | Xác định mã khách hàng |
| Uid |  | Yes | Serial | Khóa xác định id |

**Bảng Chi Nhanh**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên thuộc tính | Khóa chính | Khóa ngoài | Kiểu dữ liệu | Mô tả |
| MaCN | Yes |  | Serial | Khóa chính |
| DiaChi |  |  | Varchar(25) | Xác định địa chỉ của chi nhánh |
| TenCN |  |  | Varchar(100) | Xác định tên chi nhánh |

**Bảng LoaiTaiKhoan**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên thuộc tính | Khóa chính | Khóa ngoài | Kiểu dữ liệu | Mô tả |
| LoaiTKID | Yes |  | Serial | Xác định mã Loại tài khoản |
| TenVaiTro |  | Yes | VarChar(30) | Xác định tên loại tài khoản |

**Bảng TaiKhoan**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên thuộc tính | Khóa chính | Khóa ngoài | Kiểu dữ liệu | Mô tả |
| SoTK | Yes |  | Serial | Khóa chính |
| Sodu |  |  | Decimal(15,2) | Xác định số dư |
| LaiSuat |  |  | Decimal(10,2) | Xác định lãi suất |
| NgayMoTK |  |  | Date | Xác định ngày mở tài khoản |
| MaKH |  | Yes | Serial | Xác định chủ tài khoản |
| LoaiTKid |  | Yes | Integer | Xác định loại tài khoản |
| KiHan |  |  | Integer | Xác định kì hạn gửi tiền khách hàng đăng kí |

**Bảng LoaiGiaoDich**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên thuộc tính | Khóa chính | Khóa ngoài | Kiểu dữ liệu | Mô tả |
| LoaiGDid | Yes |  | Serial | Khóa chính |
| TenGiaoDich |  |  | Varchar(50) | Xác định tên kiểu giao dịch |

**Bảng GiaoDich**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên thuộc tính | Khóa chính | Khóa ngoài | Kiểu dữ liệu | Mô tả |
| MaGD | Yes |  | Serial | Xác định mã giao dịch |
| SoTK |  | Yes | Integer | Xác định số TK thực hiện giao dịch |
| SoTKNhan |  |  | Integer | Xác định số tài khoản nhận(Trong chuyển tiền) |
| SoTien |  |  | numenic(15,2) | Xác định số tiền giao dịch |
| Kieuid |  | Yes | integer | Xác định kiểu giao dịch |
| NgayGD |  |  | Date | Xác định ngày giao dịch |

**3. Các chức năng thực hiện**

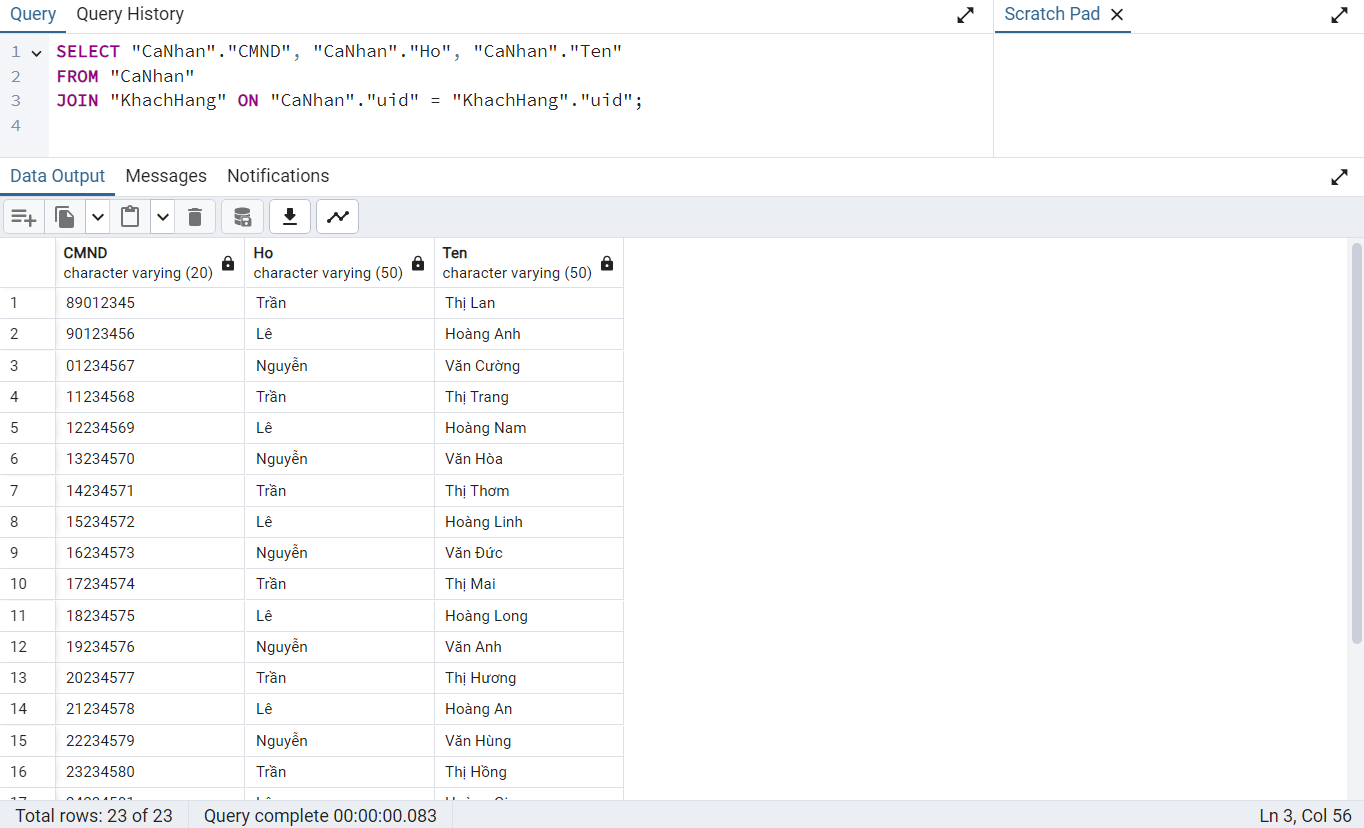
**a. Các câu truy vấn cơ bản**

Câu 1: **Lấy thông tin khách hàng bao gồm CMND và tên**:

SELECT "CaNhan"."CMND", "CaNhan"."Ho", "CaNhan"."Ten"

FROM "CaNhan"

JOIN "KhachHang" ON "CaNhan"."uid" = "KhachHang"."uid";

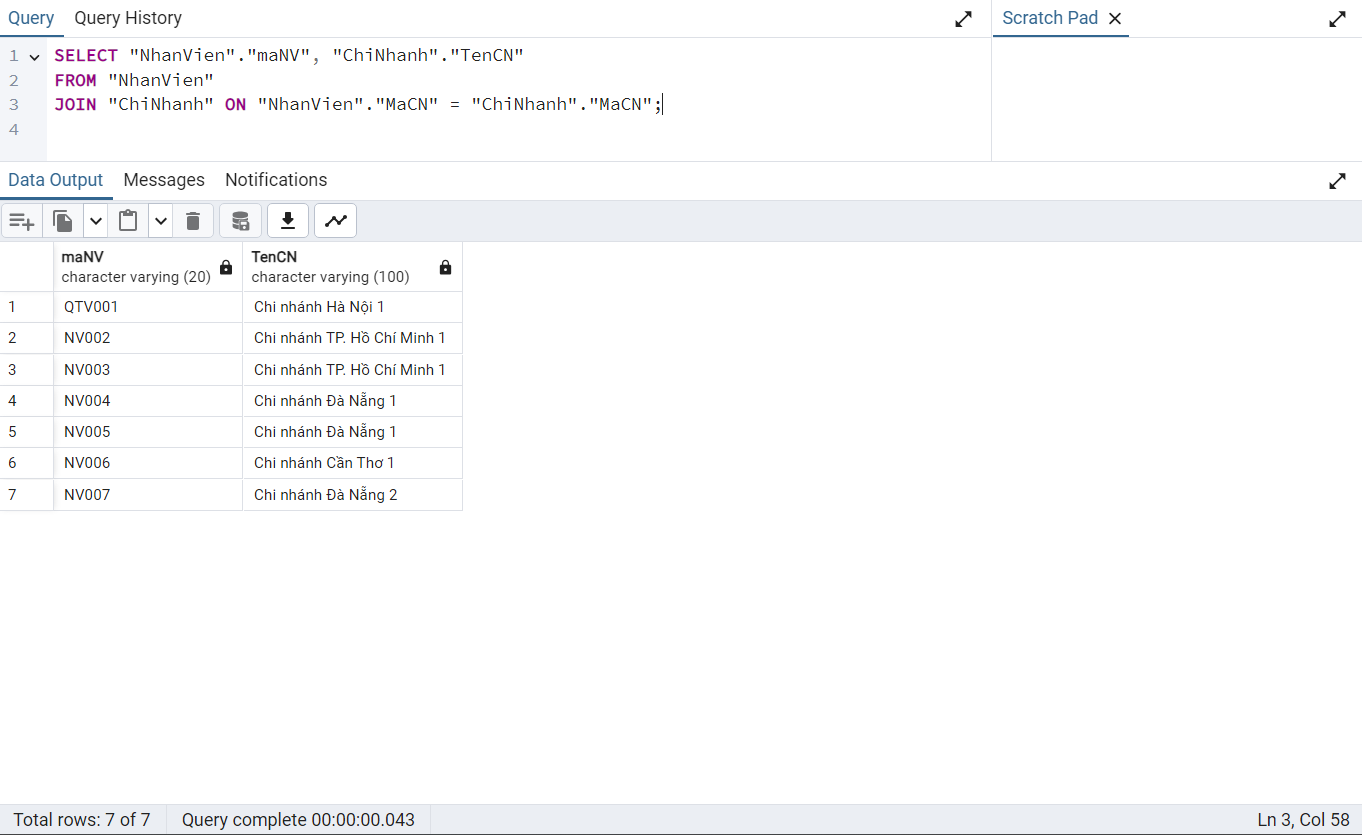


Câu 2: **Lấy danh sách tất cả các nhân viên cùng với tên chi nhánh họ làm việc**:

SELECT "NhanVien"."maNV", "ChiNhanh"."TenCN"

FROM "NhanVien"

JOIN "ChiNhanh" ON "NhanVien"."MaCN" = "ChiNhanh"."MaCN";



Câu 3: **Lấy tên loại giao dịch và số tiền giao dịch, ngày giao dịch , số tài khoản thực hiện giao dịch và số tài khoản nhận của tất cả các giao dịch**

SELECT

"LoaiGiaoDich"."TenGiaoDich" AS "LoaiGiaoDich",

"GiaoDich"."SoTien" AS "SoTien",

"GiaoDich"."ngayGD" AS "NgayGiaoDich",

"GiaoDich"."SoTK" AS "SoTaiKhoanThucHien",

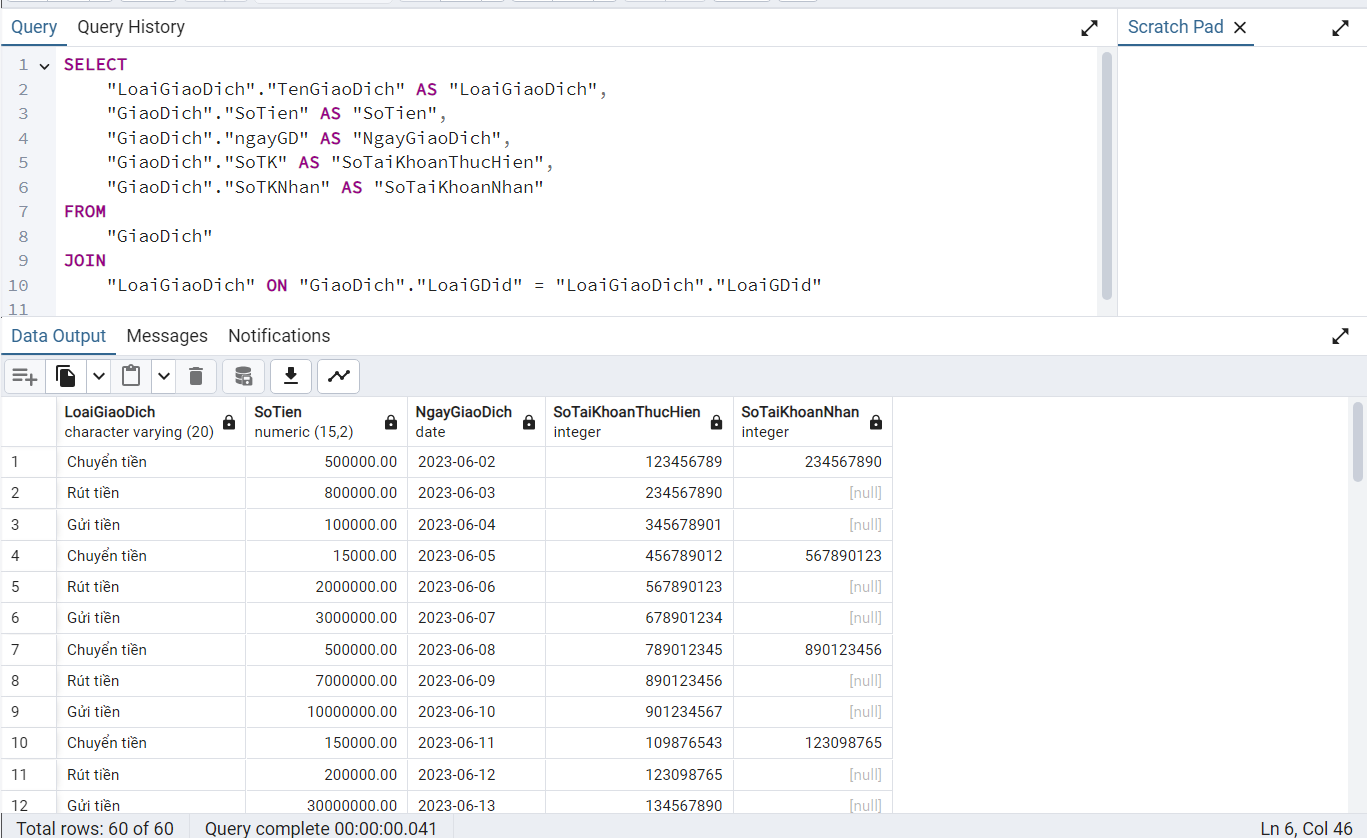
"GiaoDich"."SoTKNhan" AS "SoTaiKhoanNhan"

FROM

"GiaoDich"

JOIN

"LoaiGiaoDich" ON "GiaoDich"."LoaiGDid" = "LoaiGiaoDich"."LoaiGDid"



Câu 4: **Lấy danh sách khách hàng và số tài khoản họ có**:

SELECT

"CaNhan"."Ho",

"CaNhan"."Ten",

COUNT(CASE WHEN "TaiKhoan"."KiHan" IS NULL THEN 1 END) AS "SoTaiKhoanNganHan",

COUNT(CASE WHEN "TaiKhoan"."KiHan" IS NOT NULL THEN 1 END) AS "SoTaiKhoanDaiHan"

FROM

"CaNhan"

JOIN

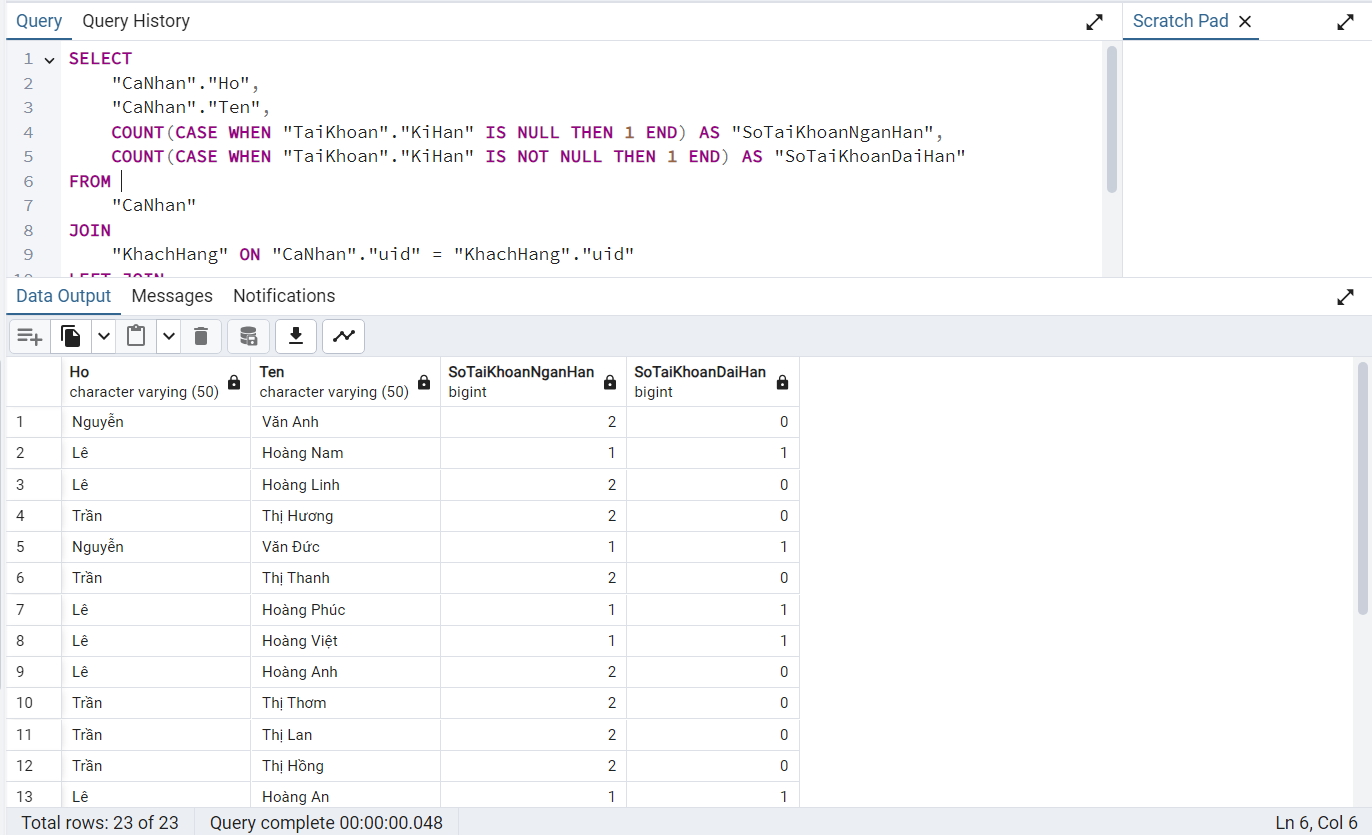
"KhachHang" ON "CaNhan"."uid" = "KhachHang"."uid"

LEFT JOIN

"TaiKhoan" ON "KhachHang"."MaKH" = "TaiKhoan"."MaKH"

GROUP BY

"CaNhan"."Ho", "CaNhan"."Ten";



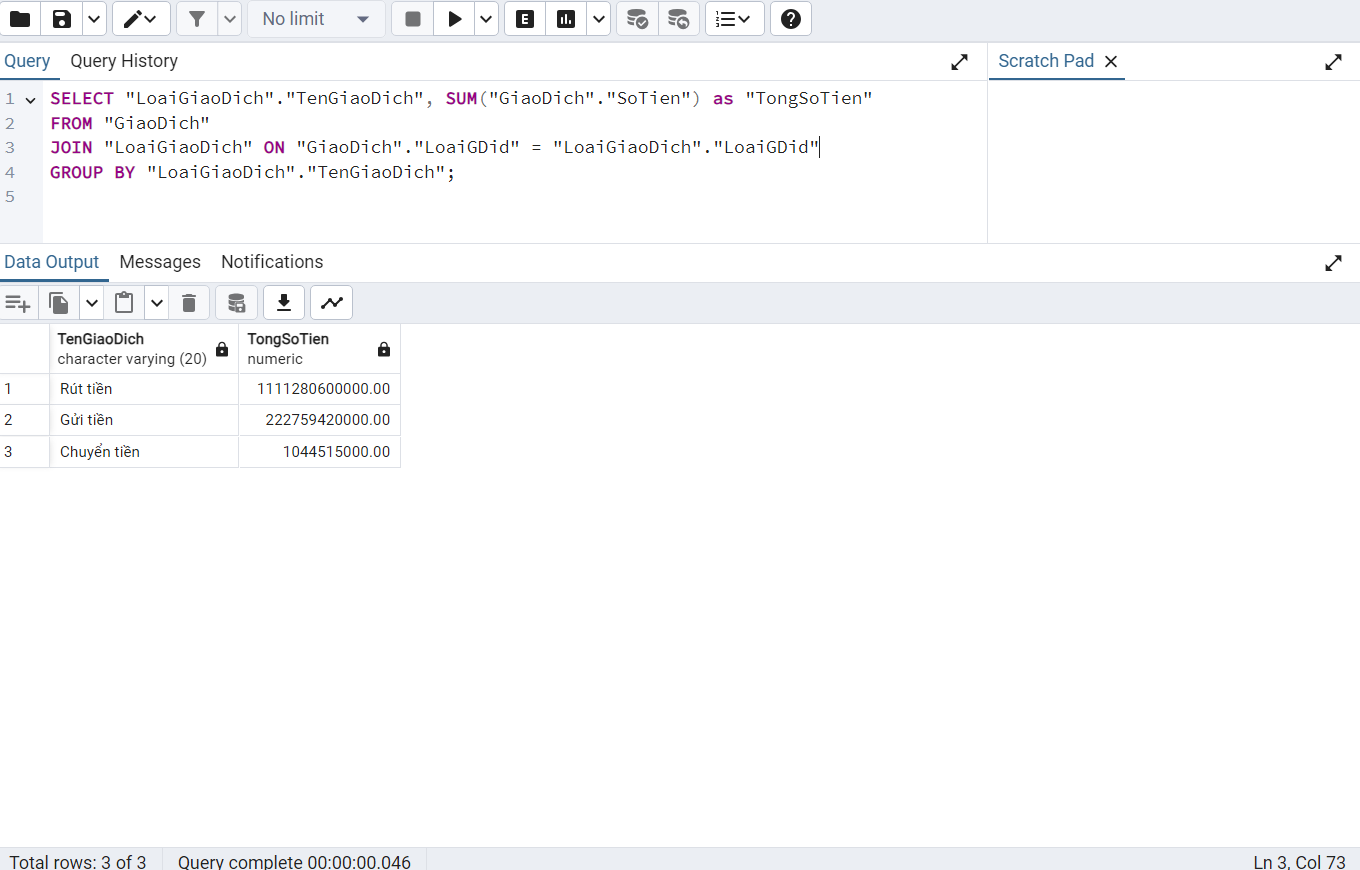
Câu 5: **Tính tổng số tiền giao dịch cho mỗi loại giao dịch**:

SELECT "LoaiGiaoDich"."TenGiaoDich", SUM("GiaoDich"."SoTien") as "TongSoTien"

FROM "GiaoDich"

JOIN "LoaiGiaoDich" ON "GiaoDich"."LoaiGDid" = "LoaiGiaoDich"."LoaiGDid"

GROUP BY "LoaiGiaoDich"."TenGiaoDich";

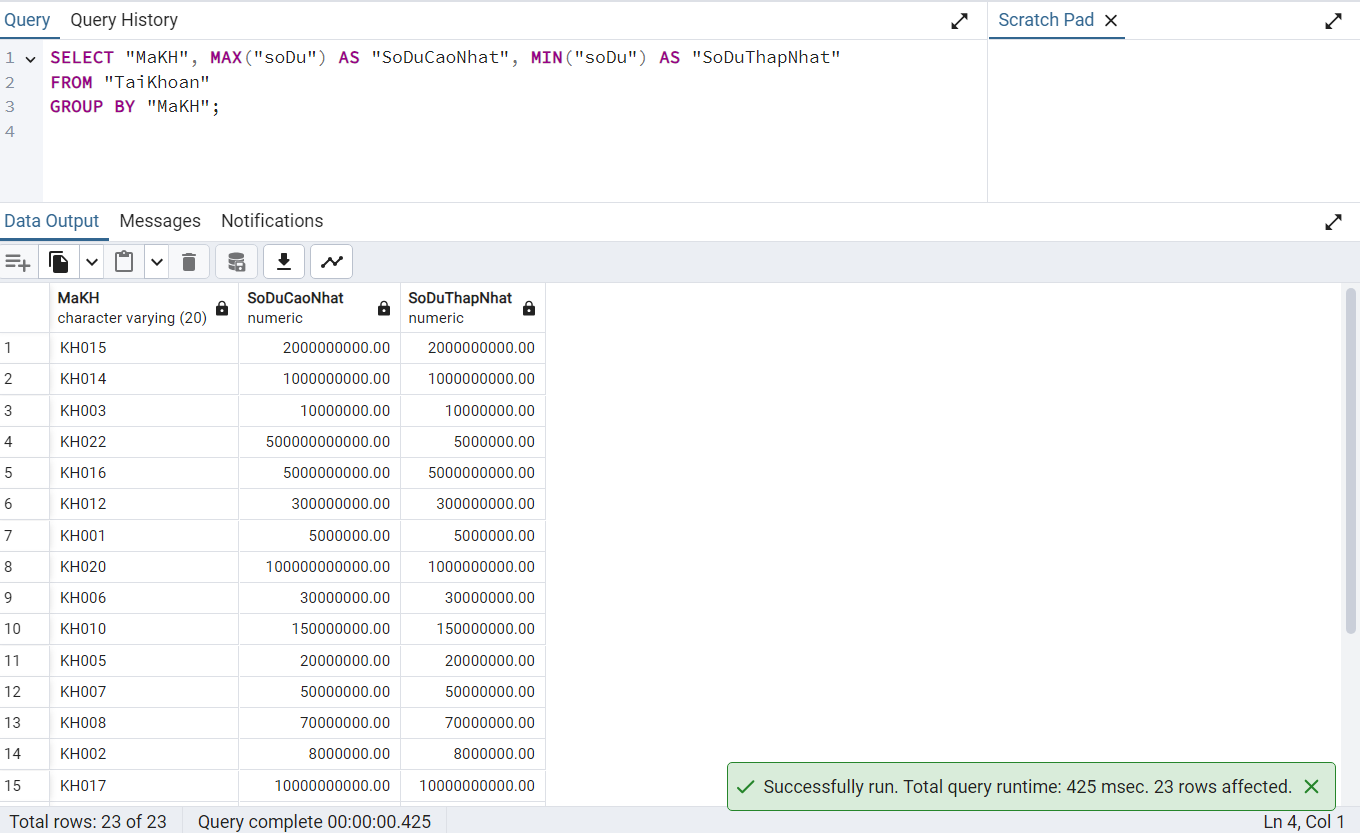


Câu 6: Lấy danh sách tài khoản có số dư cao nhất và thấp nhất của mỗi khách hàng:

SELECT "MaKH", MAX("soDu") AS "SoDuCaoNhat", MIN("soDu") AS "SoDuThapNhat"

FROM "TaiKhoan"

GROUP BY "MaKH";

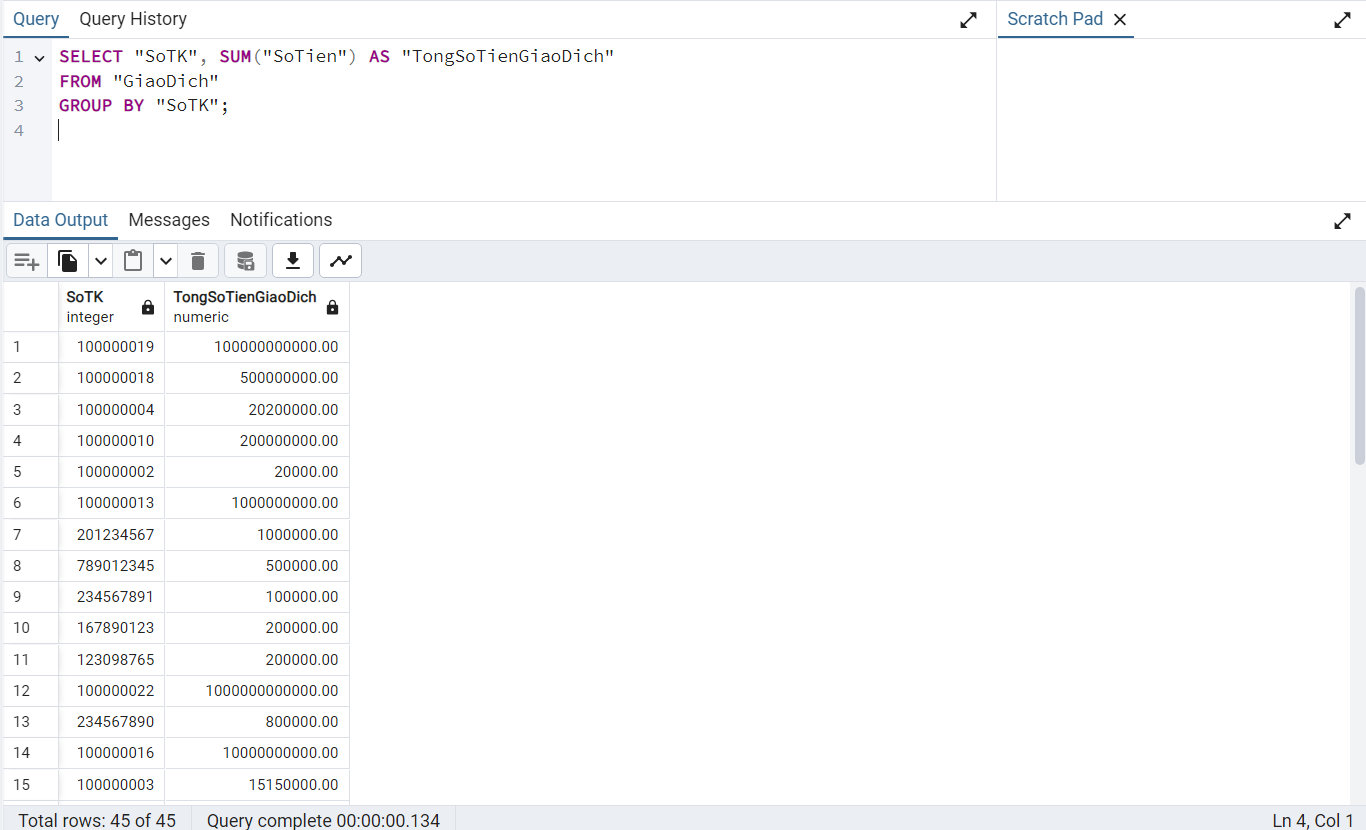


Câu 7: Lấy tổng số tiền giao dịch ứng với mỗi tài khoản

SELECT "SoTK", SUM("SoTien") AS "TongSoTienGiaoDich"

FROM "GiaoDich"

GROUP BY "SoTK";



Câu 8: Tổng số giao dịch theo mỗi tháng và tổng số tiền mỗi loại giao dịch trong tháng đó

SELECT

EXTRACT(YEAR FROM "ngayGD") AS "Nam",

EXTRACT(MONTH FROM "ngayGD") AS "Thang",

COUNT(\*) AS "TongSoGiaoDich",

SUM("SoTien") AS "TongSoTien"

FROM

"GiaoDich"

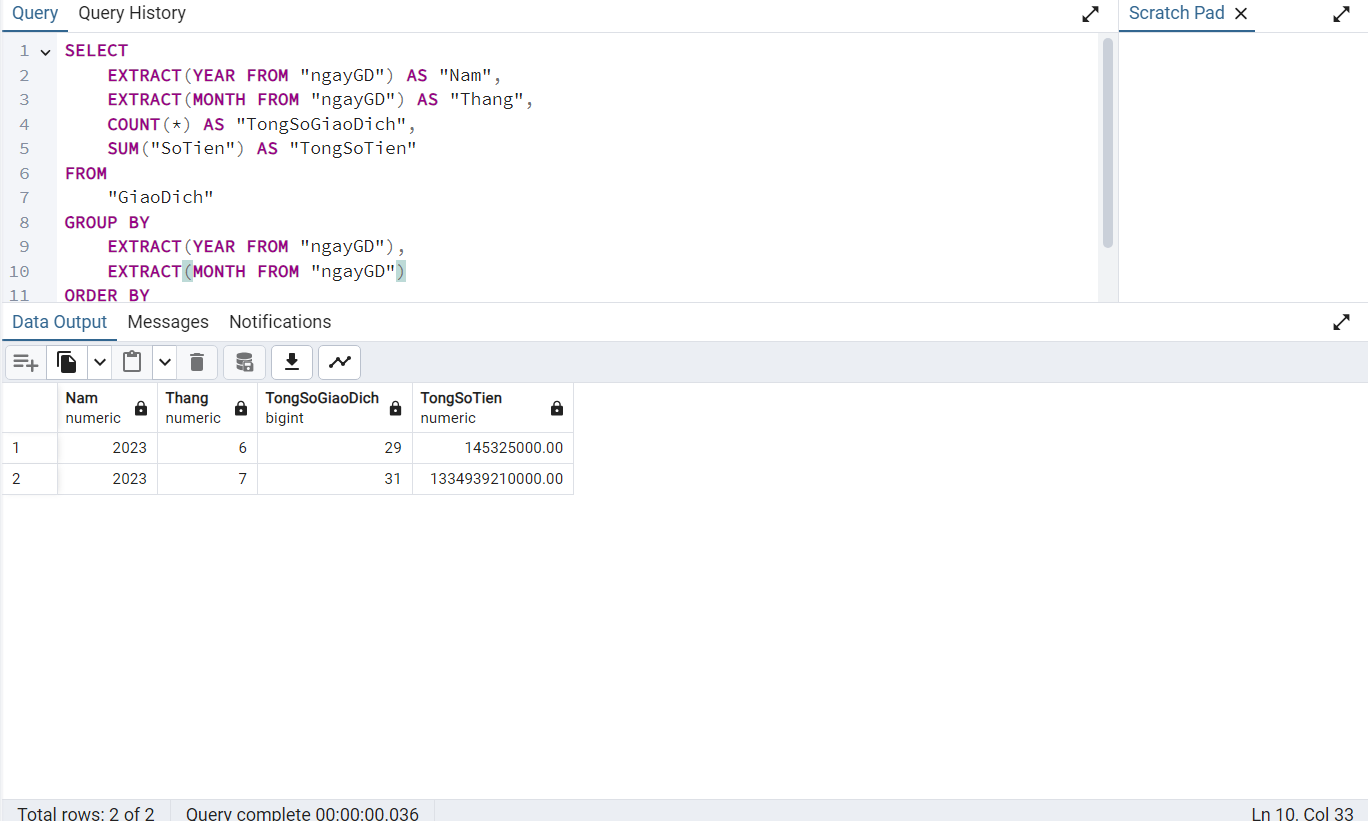
GROUP BY

EXTRACT(YEAR FROM "ngayGD"),

EXTRACT(MONTH FROM "ngayGD")

ORDER BY

"Nam", "Thang";



Câu 9: Lấy danh sách khách hàng có tổng số tiền giao dịch lớn nhất:

SELECT

"KhachHang"."MaKH",

SUM("GiaoDich"."SoTien") AS "TongSoTienGiaoDich"

FROM

"GiaoDich"

JOIN

"TaiKhoan" ON "GiaoDich"."SoTK" = "TaiKhoan"."SoTK"

JOIN

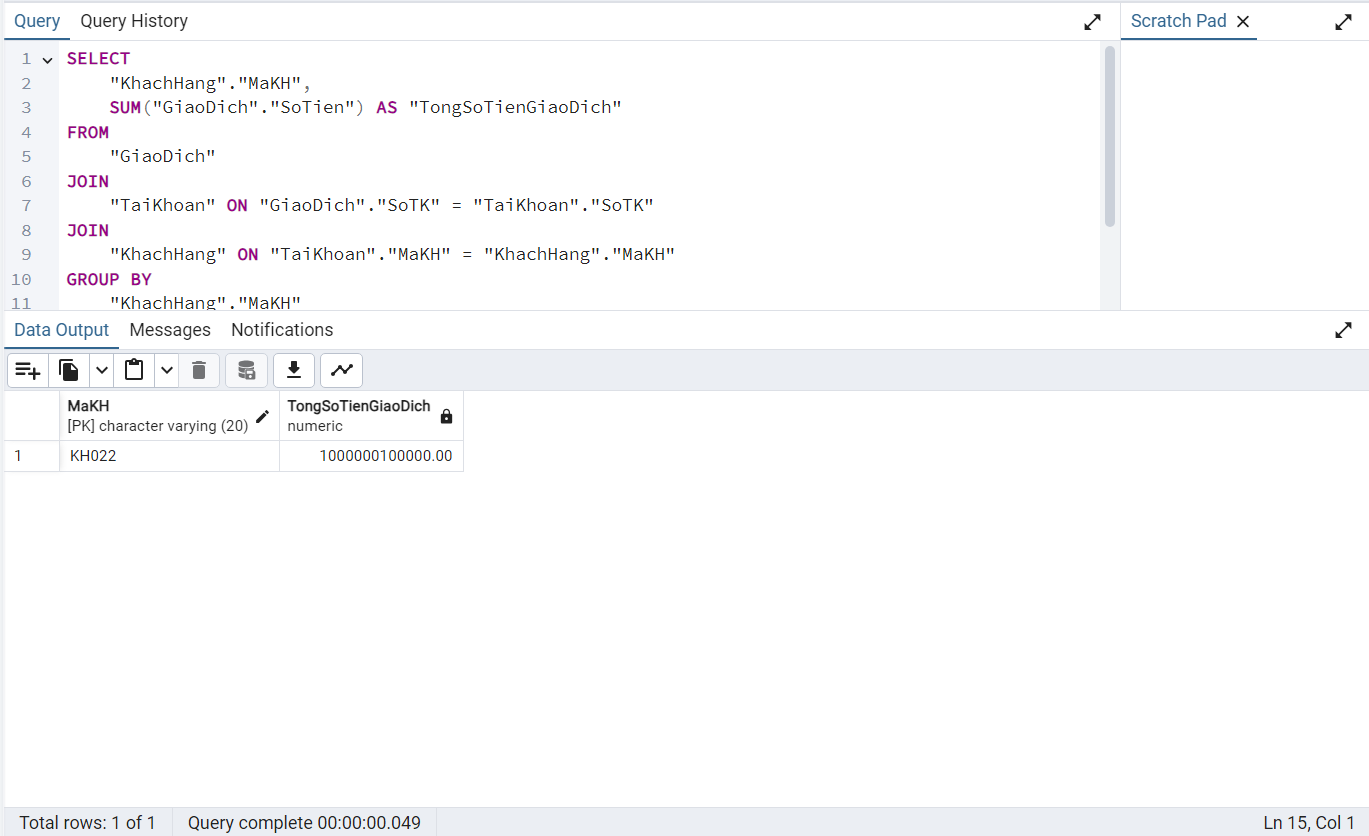
"KhachHang" ON "TaiKhoan"."MaKH" = "KhachHang"."MaKH"

GROUP BY

"KhachHang"."MaKH"

ORDER BY

"TongSoTienGiaoDich" DESC

LIMIT 1;  


Câu 10: số lượng khách hàng ứng với mỗi thành phố

SELECT

"CaNhan"."DC" AS "ThanhPho",

COUNT("KhachHang"."uid") AS "SoLuongKhachHang"

FROM

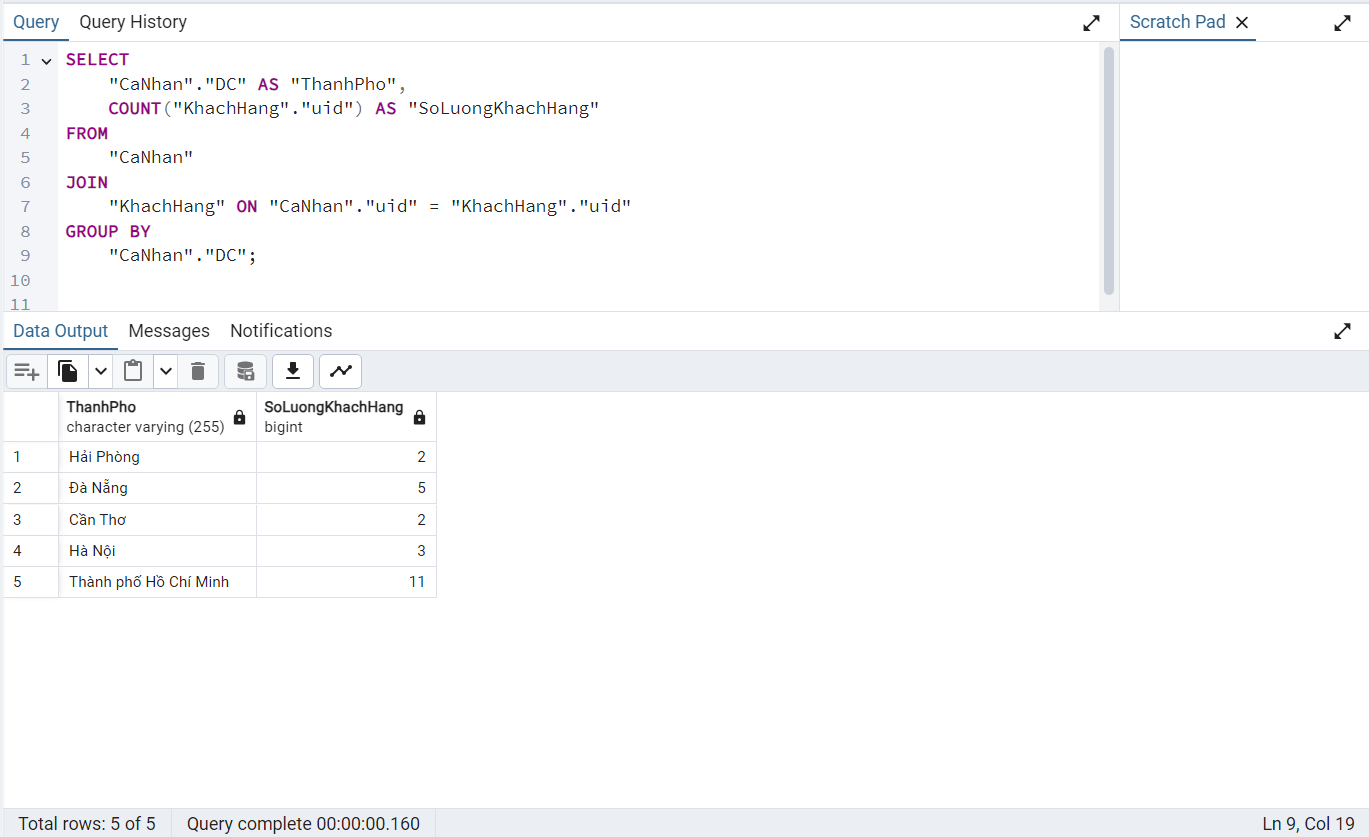
"CaNhan"

JOIN

"KhachHang" ON "CaNhan"."uid" = "KhachHang"."uid"

GROUP BY

"CaNhan"."DC";



**b : Phân quyền**

-- Tạo vai trò cho quản trị viên

CREATE ROLE admin\_role;

-- Tạo vai trò cho nhân viên

CREATE ROLE employee\_role;

-- Tạo vai trò cho khách hàng

CREATE ROLE customer\_role;

--Quản trị viên sẽ có toàn quyền truy cập vào tất cả các bảng và sequence.

GRANT ALL ON ALL TABLES IN SCHEMA public TO admin\_role;

GRANT ALL ON ALL SEQUENCES IN SCHEMA public TO admin\_role;

--Nhân viên chỉ được phép xem và thực hiện các thao tác cơ bản trên dữ liệu, nhưng không có quyền thực hiện các thao tác quản trị.

GRANT SELECT, INSERT, UPDATE, DELETE ON TABLE "GiaoDich" TO employee\_role;

--Khách hàng chỉ có quyền xem thông tin tài khoản và các giao dịch của riêng họ.

GRANT SELECT ON TABLE "TaiKhoan" TO customer\_role;

GRANT SELECT ON TABLE "GiaoDich" TO customer\_role;

-- Tạo và gán vai trò cho quản trị viên

CREATE USER admin\_user WITH PASSWORD 'password';

GRANT admin\_role TO admin\_user;

-- Tạo và gán vai trò cho nhân viên

CREATE USER employee\_user WITH PASSWORD 'password';

GRANT employee\_role TO employee\_user;

-- Tạo và gán vai trò cho khách hàng

CREATE USER customer\_user WITH PASSWORD 'password';

GRANT customer\_role TO customer\_user;

**Trigger update lại lãi xuất mỗi khi khách hang thực hiện giao dịch (Khi tài khoản tiết kiệm thực hiện giao dịch rút tiền trước ngày đáo hạn thì lãi suất sẽ về 0)**

CREATE OR REPLACE FUNCTION update\_interest()

RETURNS TRIGGER

LANGUAGE plpgsql

AS $$

DECLARE

v\_ngay\_dao\_han DATE;

BEGIN

-- Lấy ngày đáo hạn

SELECT "ngaymoTK" + INTERVAL '1 month' \* "Kihan" INTO v\_ngay\_dao\_han

FROM "TaiKhoan"

WHERE "SoTK" = NEW."SoTK";

-- Nếu giao dịch là rút tiền trước kỳ hạn và loại tài khoản là 2

IF NEW."LoaiGDid" = 2 AND NEW."ngayGD" < v\_ngay\_dao\_han THEN

-- Cập nhật lãi suất về 0

UPDATE "TaiKhoan"

SET "LaiSuat" = 0

WHERE "TaiKhoan"."SoTK" = NEW."SoTK";

END IF;

RETURN NEW;

END;

$$;

**c.Procedure cập nhật số dư cho tài khoản tiết kiệm khi đến kì hạn trả lãi suất**

BEGIN

-- Cập nhật số dư cho các tài khoản có lãi suất > 0 và đến kỳ hạn

UPDATE "TaiKhoan"

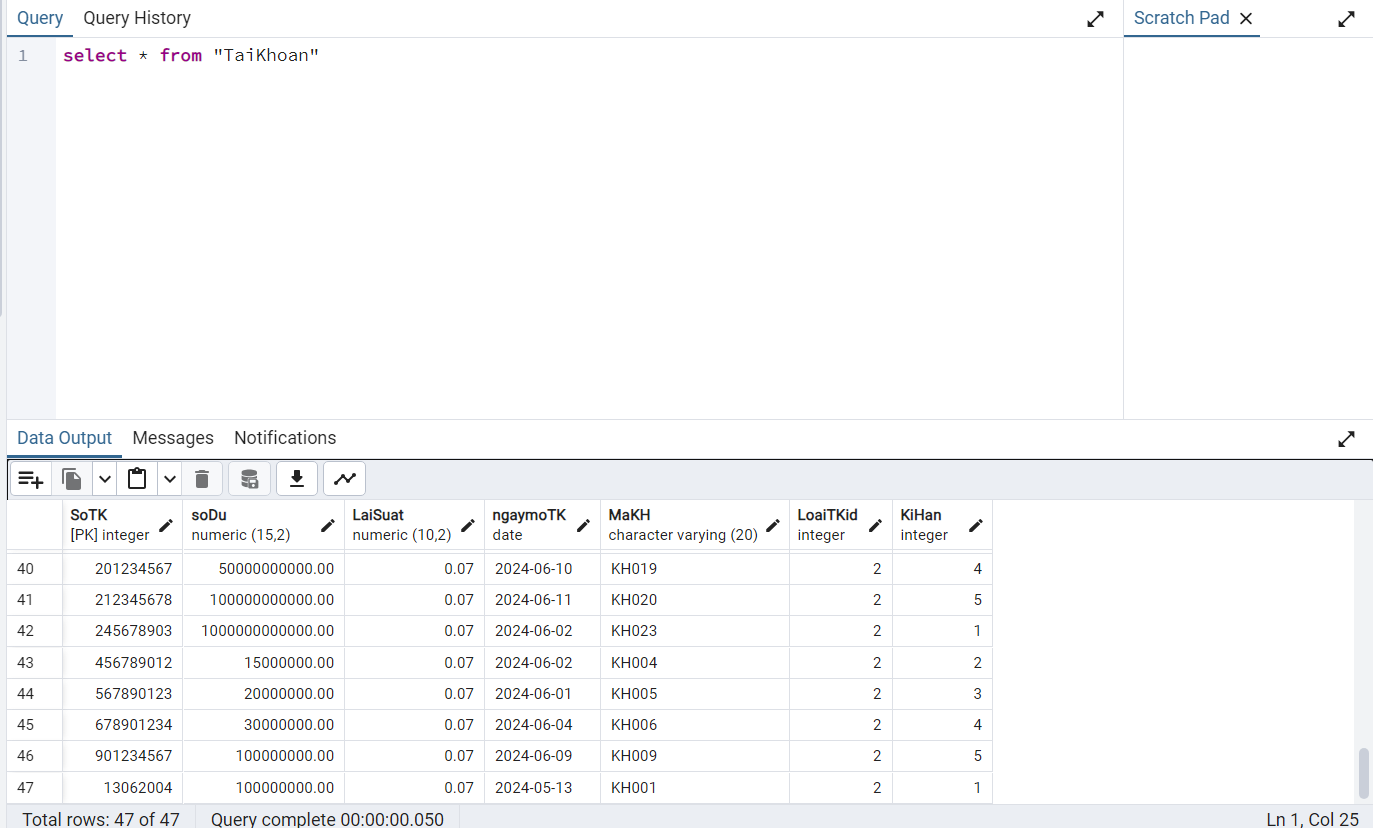
SET "soDu" = "soDu" \* POWER((1 + "LaiSuat" / 100), "KiHan"),

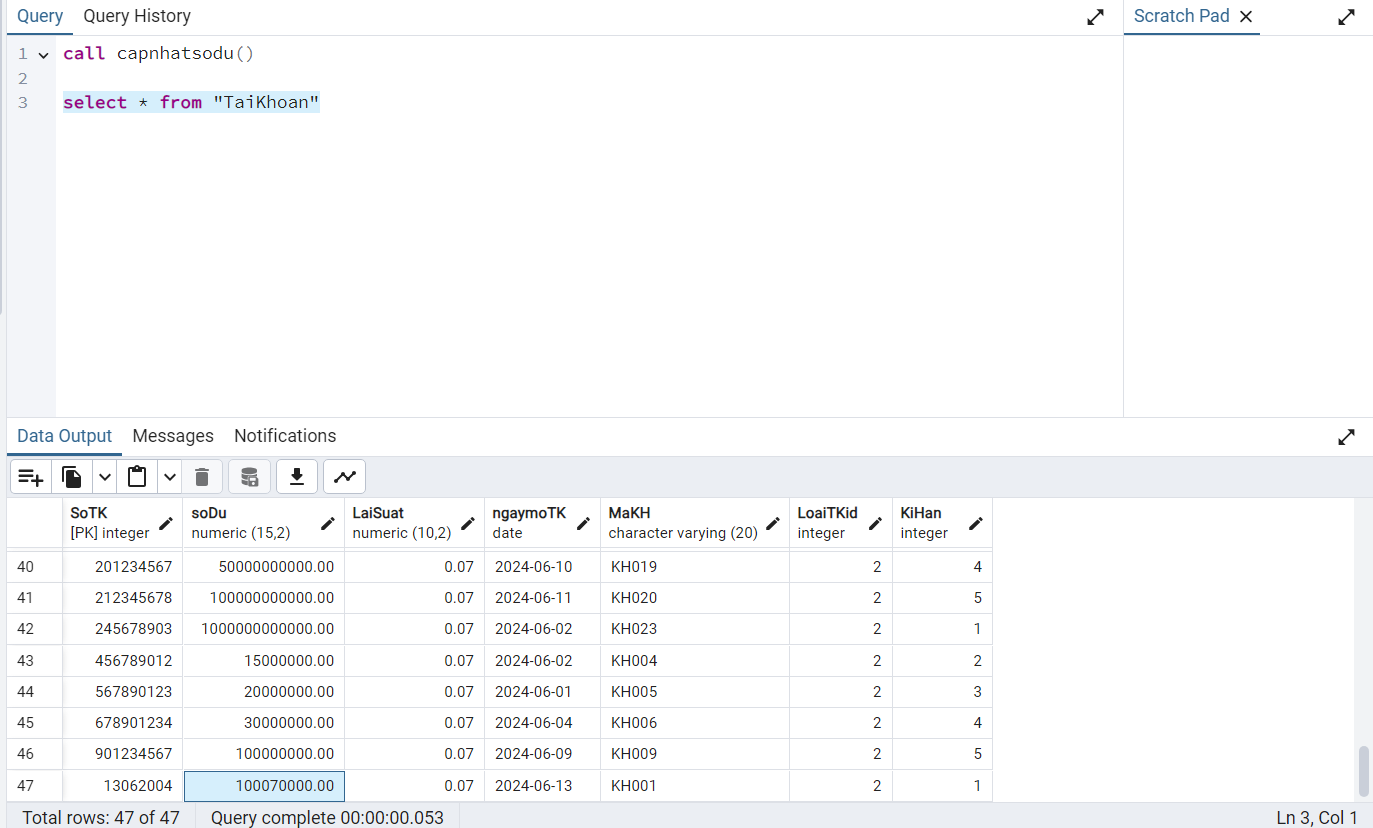
"ngaymoTK" = "ngaymoTK" + INTERVAL '1 month' \* "KiHan"

WHERE "LaiSuat" > 0

AND CURRENT\_DATE = "ngaymoTK" + INTERVAL '1 month' \* "KiHan";

END;

Trước: Tài khoản thứ 47 có số dư là 1 triệu  
****

Sau:Tài khoản số 47 đã cập nhật số dư cộng thêm lãi suất  
****

**d.Procedure cập nhật phí duy trì tài khoản tiết kiệm(Khi tài khoản có số dư nhỏ hơn 1 triệu)**

CREATE OR REPLACE PROCEDURE CapNhatSoDuTK2()

LANGUAGE plpgsql

AS $$

BEGIN

-- Áp dụng phí duy trì cho tài khoản loại 2

UPDATE "TaiKhoan"

SET "soDu" = CASE

WHEN "soDu" < 50000 THEN 0

WHEN "soDu" < 1000000 THEN "soDu" - 50000

ELSE "soDu"

END

WHERE "LoaiTKid" = 2

AND date\_part('day', CURRENT\_DATE) = date\_part('day', "ngaymoTK")

AND date\_part('month', age(CURRENT\_DATE, "ngaymoTK")) > 0;

END;

$$;

**e. Procedure gửi tiền**

DECLARE

v\_LoaiTK INT;

v\_TongTienTrongNgay DECIMAL;

BEGIN

-- Lấy thông tin loại tài khoản

SELECT "LoaiTKid"

INTO v\_LoaiTK

FROM "TaiKhoan"

WHERE "SoTK" = p\_SoTK;

-- Kiểm tra loại tài khoản

IF v\_LoaiTK = 1 THEN

-- Kiểm tra tổng số tiền trong ngày

SELECT SUM("SoTien")

INTO v\_TongTienTrongNgay

FROM "GiaoDich"

WHERE "SoTK" = p\_SoTK

AND DATE("ngayGD") = CURRENT\_DATE;

IF p\_SoTien >= 1000 AND p\_SoTien <= 200000000 AND v\_TongTienTrongNgay + p\_SoTien <= 500000000 THEN

-- Thực hiện giao dịch gửi tiền

INSERT INTO "GiaoDich" ("SoTK", "SoTien", "LoaiGDid", "ngayGD")

VALUES (p\_SoTK, p\_SoTien, 3, CURRENT\_TIMESTAMP);

-- Cập nhật số dư của tài khoản

UPDATE "TaiKhoan"

SET "soDu" = "soDu" + p\_SoTien

WHERE "SoTK" = p\_SoTK;

ELSE

RAISE EXCEPTION 'Số tiền gửi không hợp lệ hoặc vượt quá giới hạn ngày';

END IF;

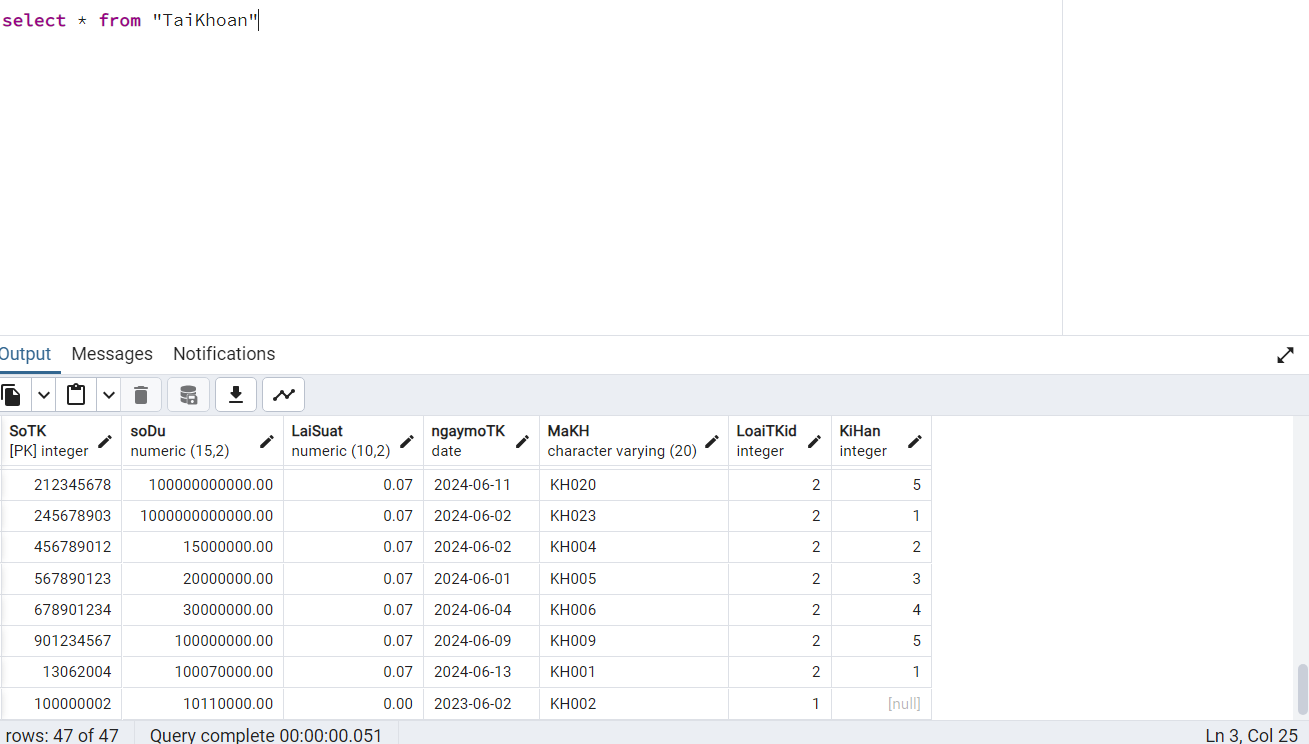
ELSE

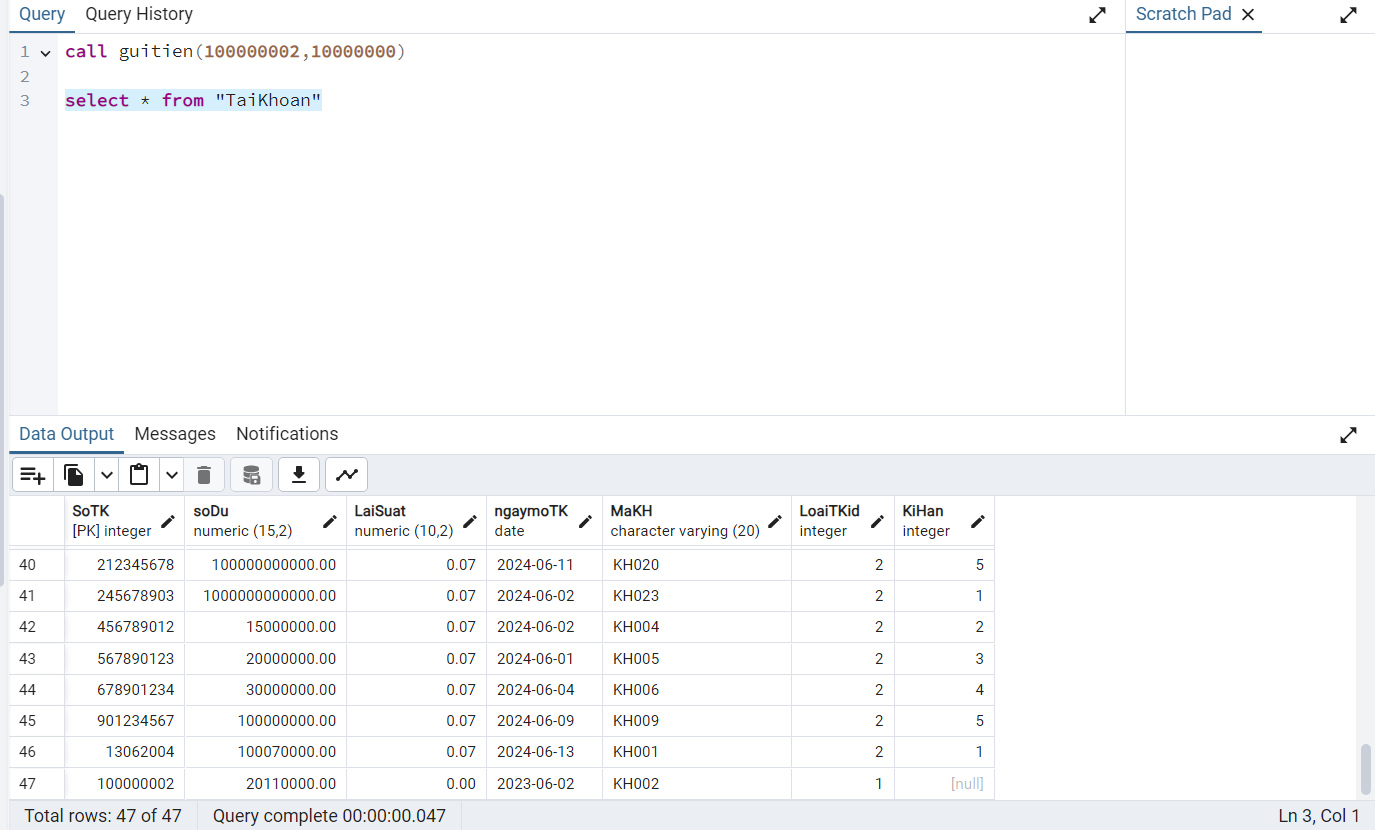
RAISE EXCEPTION 'Không hỗ trợ loại tài khoản này';

END IF;

END;

Trước



Sau: 

**f.Procedure Rút Tiền**

CREATE OR REPLACE PROCEDURE RutTien(

p\_SoTK INT,

p\_SoTien DECIMAL

)

LANGUAGE plpgsql

AS $$

DECLARE

v\_SoDu DECIMAL;

BEGIN

-- Lấy số dư hiện tại của tài khoản

SELECT "soDu"

INTO v\_SoDu

FROM "TaiKhoan"

WHERE "SoTK" = p\_SoTK;

-- Kiểm tra số tiền rút hợp lệ

IF p\_SoTien > 0 AND p\_SoTien <= v\_SoDu THEN

-- Thực hiện giao dịch rút tiền

INSERT INTO "GiaoDich" ("SoTK", "SoTien", "LoaiGDid", "ngayGD")

VALUES (p\_SoTK, -p\_SoTien, 1, CURRENT\_TIMESTAMP);

-- Cập nhật số dư của tài khoản

UPDATE "TaiKhoan"

SET "soDu" = "soDu" - p\_SoTien

WHERE "SoTK" = p\_SoTK;

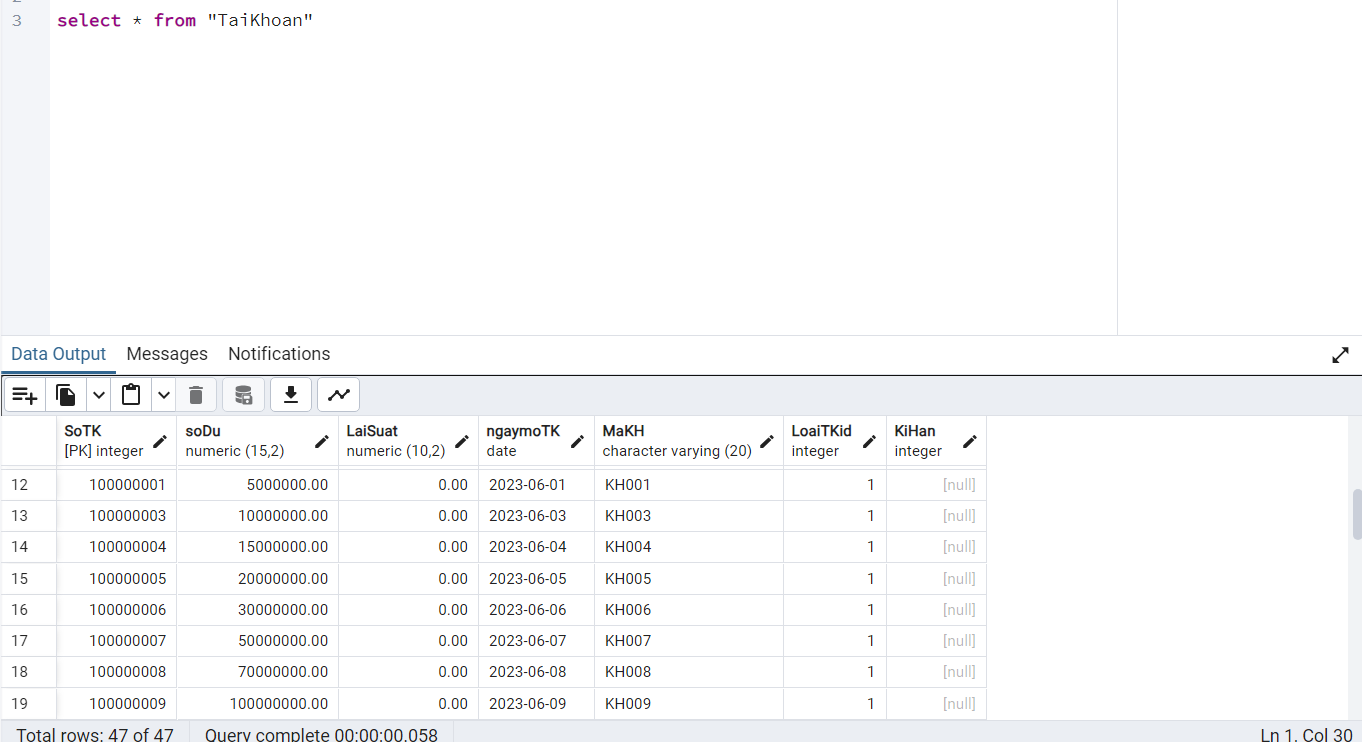
ELSE

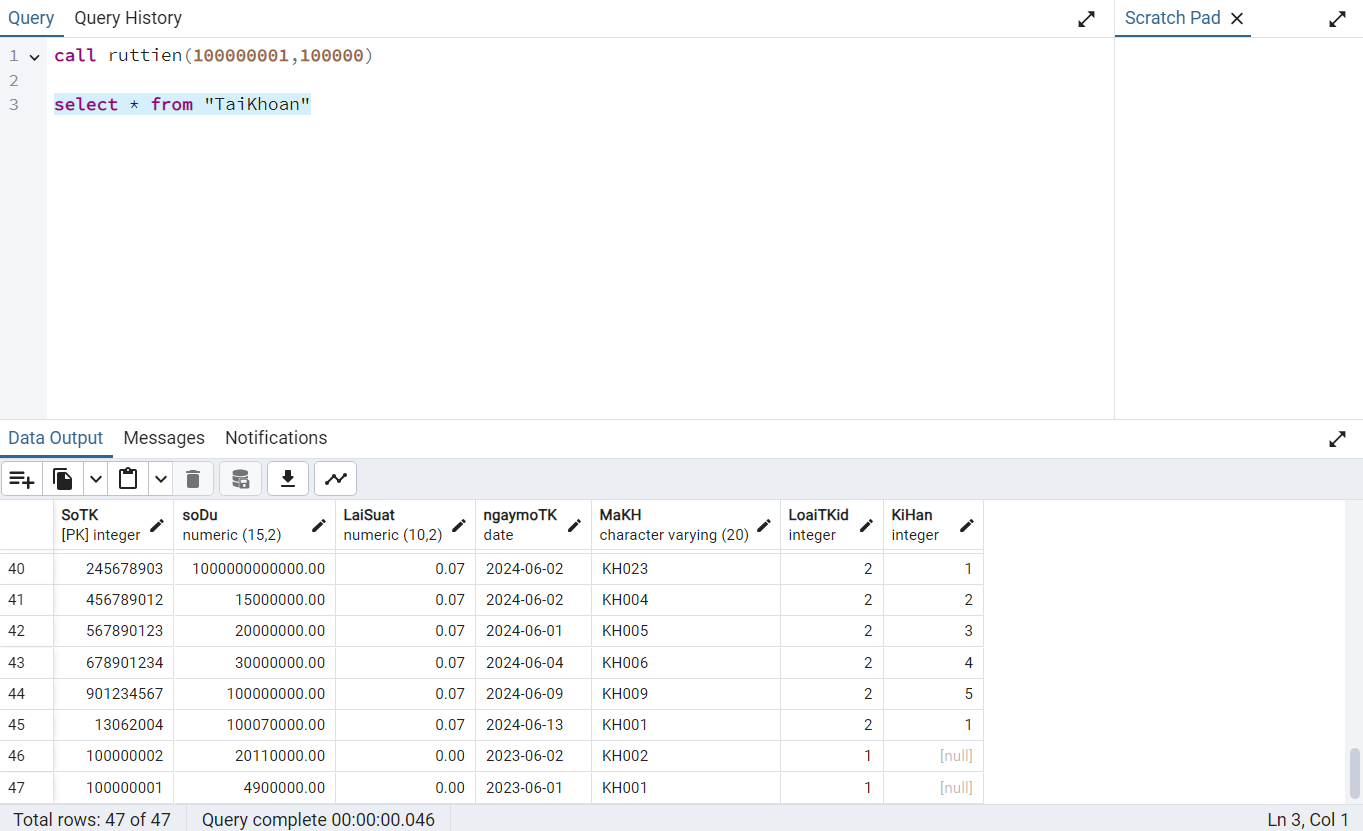
RAISE EXCEPTION 'Số tiền rút không hợp lệ hoặc vượt quá số dư';

END IF;

END;

$$;

Trước: 

Sau: 

**g. Procedure chuyển khoản**

DECLARE

v\_SoDuGui DECIMAL;

BEGIN

-- Lấy số dư hiện tại của tài khoản gửi

SELECT "soDu" INTO v\_SoDuGui

FROM "TaiKhoan"

WHERE "SoTK" = p\_SoTKGui;

-- Kiểm tra số tiền chuyển hợp lệ

IF p\_SoTien > 0 AND p\_SoTien <= v\_SoDuGui THEN

-- Thực hiện giao dịch chuyển tiền

INSERT INTO "GiaoDich" ("SoTK", "SoTKNhan", "SoTien", "LoaiGDid", "ngayGD")

VALUES (p\_SoTKGui, p\_SoTKNhan, p\_SoTien, 1, CURRENT\_TIMESTAMP);

-- Cập nhật số dư của tài khoản gửi và tài khoản nhận

UPDATE "TaiKhoan"

SET "soDu" = "soDu" - p\_SoTien

WHERE "SoTK" = p\_SoTKGui;

UPDATE "TaiKhoan"

SET "soDu" = "soDu" + p\_SoTien

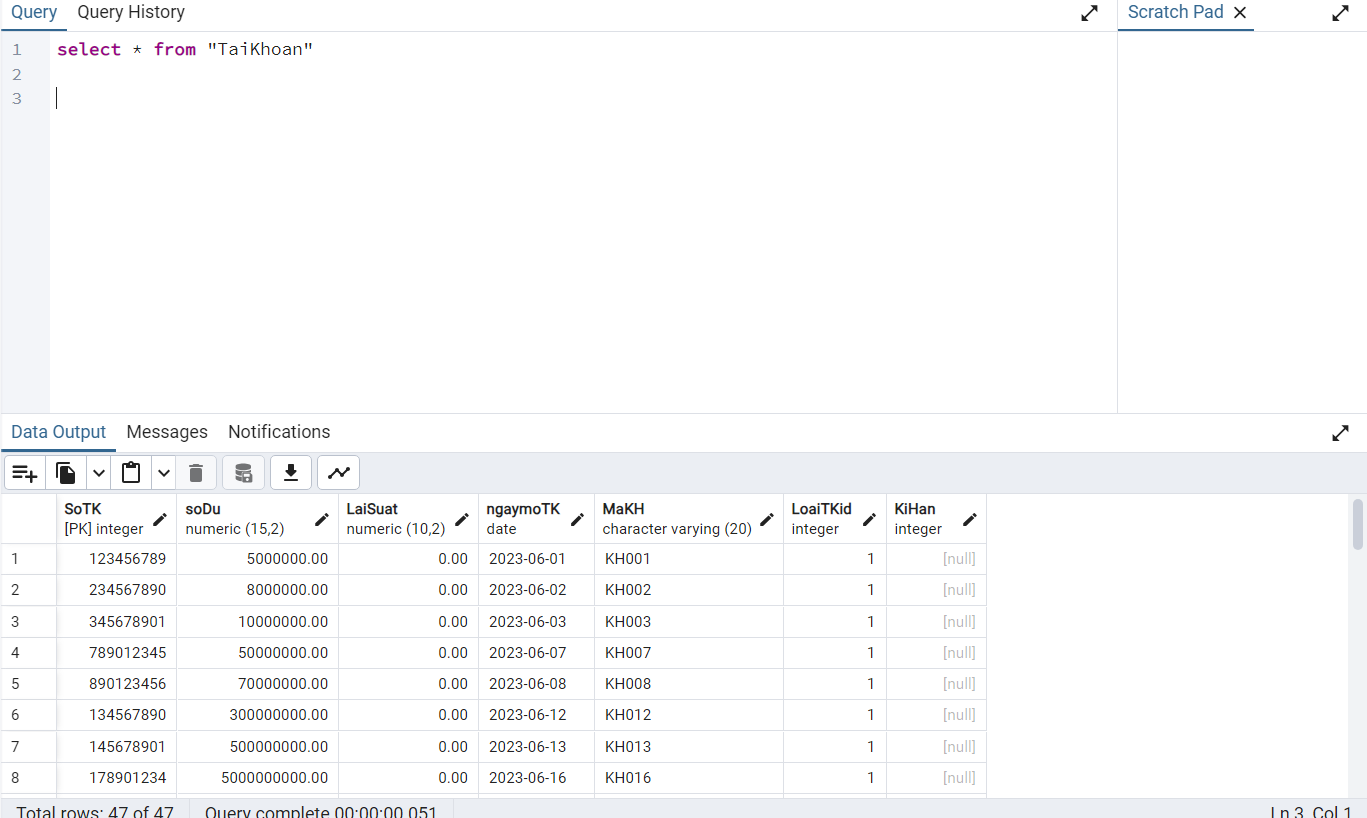
WHERE "SoTK" = p\_SoTKNhan;

ELSE

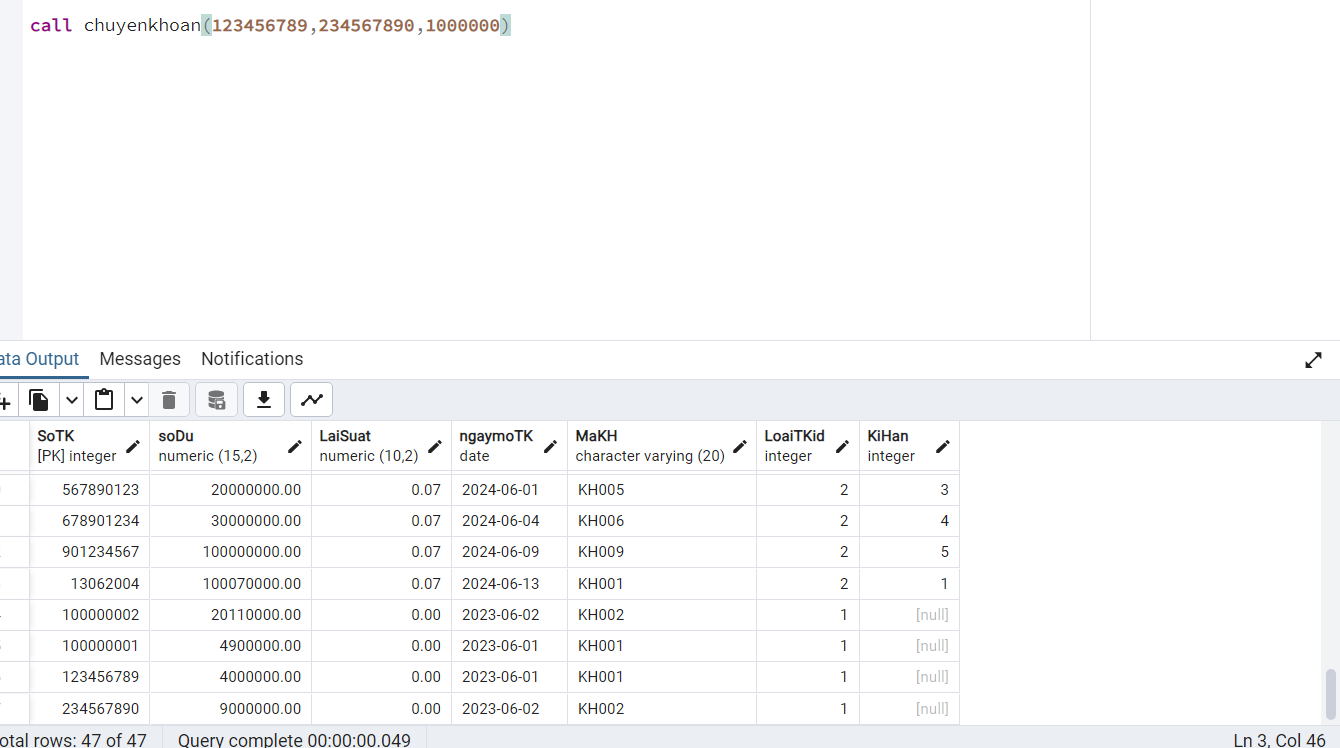
RAISE EXCEPTION 'Số tiền chuyển không hợp lệ hoặc vượt quá số dư';

END IF;

END;

Trước : 

Sau :



**h. Xóa tài khoản và giao dịch**

CREATE OR REPLACE PROCEDURE XoaTaiKhoanVaGiaoDich(SoTKToDelete INT)

LANGUAGE plpgsql

AS $$

DECLARE

v\_so\_du DECIMAL(15,2);

BEGIN

-- Kiểm tra số dư của tài khoản

SELECT "soDu" INTO v\_so\_du

FROM "TaiKhoan"

WHERE "SoTK" = SoTKToDelete;

-- Nếu số dư bằng 0, thì mới thực hiện xóa

IF v\_so\_du = 0 THEN

-- Xóa các giao dịch liên quan đến tài khoản

DELETE FROM "GiaoDich"

WHERE "SoTK" = SoTKToDelete;

-- Xóa tài khoản

DELETE FROM "TaiKhoan"

WHERE "SoTK" = SoTKToDelete;

RAISE NOTICE 'Đã xóa tài khoản và các giao dịch liên quan thành công.';

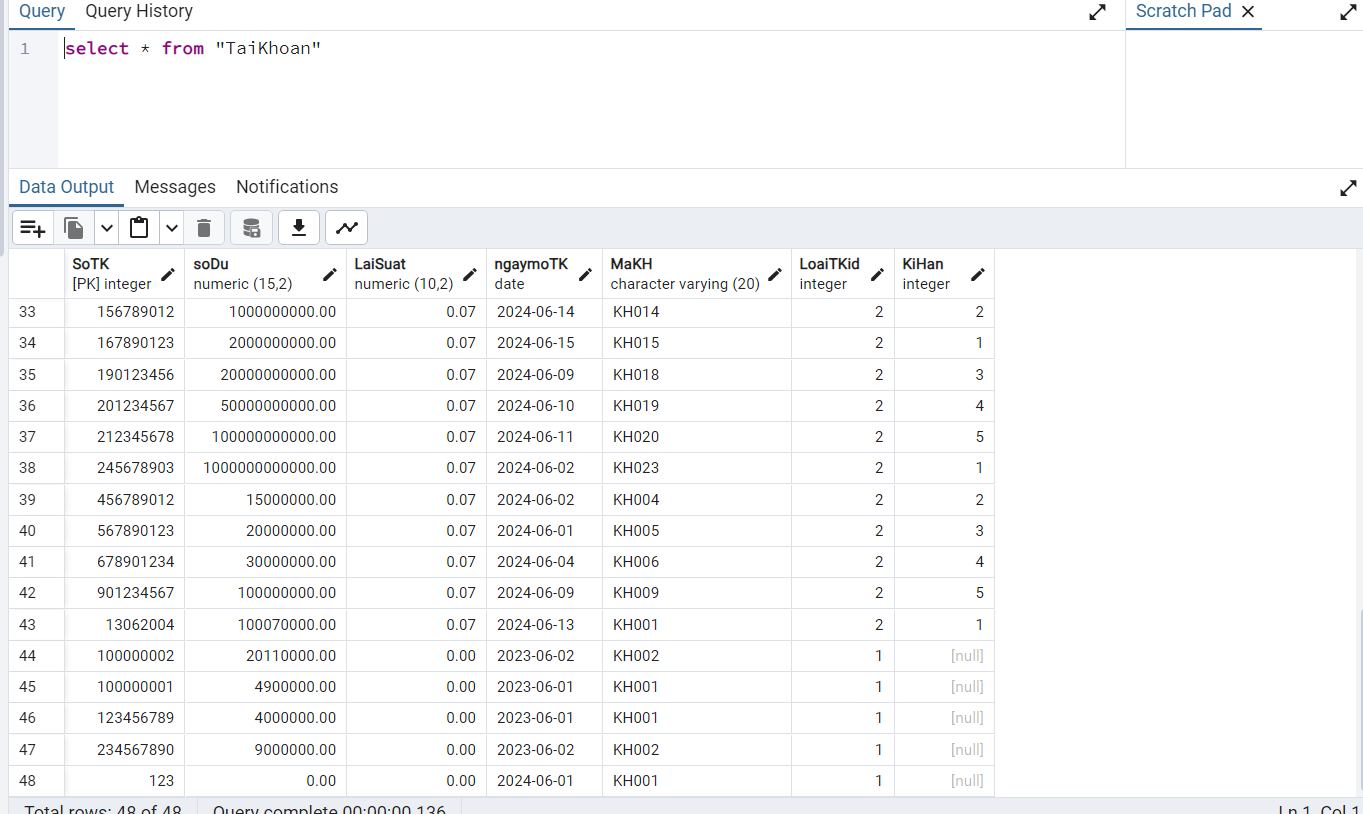
ELSE

RAISE EXCEPTION 'Không thể xóa tài khoản với số dư khác 0.';

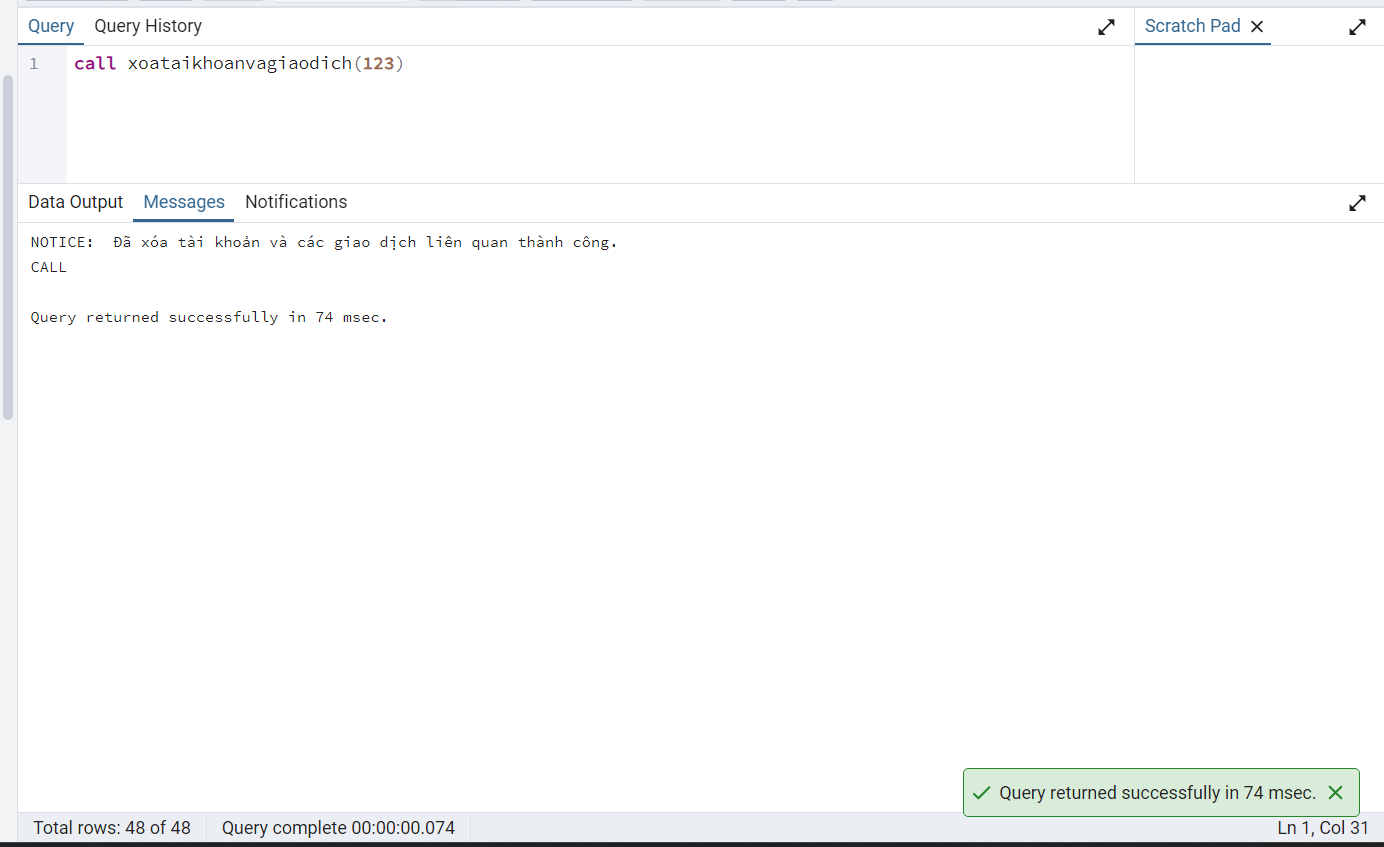
END IF;

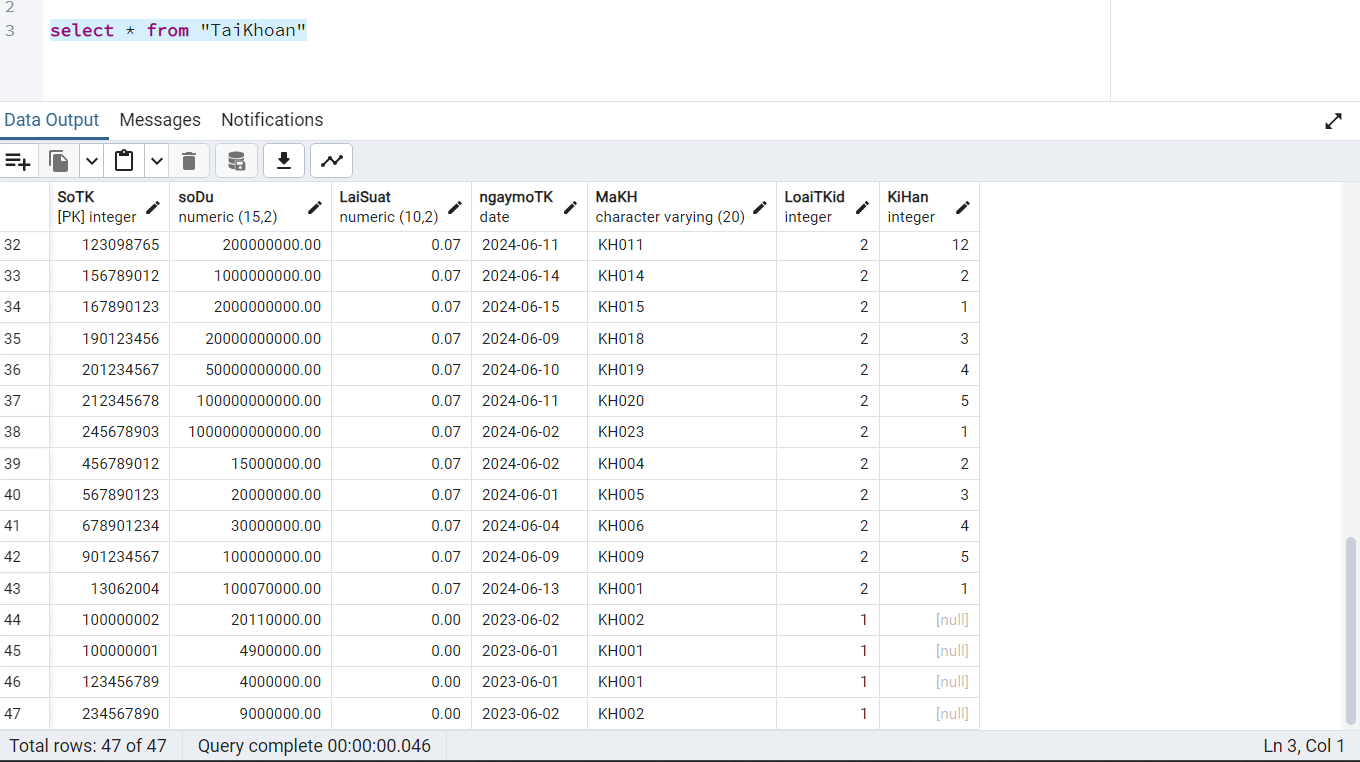
END $$;

**i : Xóa tài khoản**

Trước: 

Sau:





**j. Chuyển nhân viên**

CREATE OR REPLACE PROCEDURE ChuyenNhanVien(

p\_maNV VARCHAR(20), -- Mã nhân viên của nhân viên cần chuyển

p\_maCN\_cu INT, -- Mã chi nhánh cũ của nhân viên

p\_maCN\_moi INT -- Mã chi nhánh mới của nhân viên

)

LANGUAGE plpgsql

AS $$

BEGIN

UPDATE "NhanVien"

SET "MaCN" = p\_maCN\_moi

WHERE "maNV" = p\_maNV AND "MaCN" = p\_maCN\_cu;

IF NOT FOUND THEN

RAISE EXCEPTION 'Không tìm thấy nhân viên với maNV: %, maCN: %', p\_maNV, p\_maCN\_cu;

END IF;

EXCEPTION

WHEN OTHERS THEN

RAISE EXCEPTION 'Có lỗi xảy ra khi chuyển nhân viên: %', SQLERRM;

END;

$$;

**k.Chuyển bộ phận**

CREATE OR REPLACE PROCEDURE ChuyenBoPhan(

p\_maNV VARCHAR(20), -- Mã nhân viên của nhân viên cần chuyển

p\_maBoPhan\_cu VARCHAR(20), -- Mã bộ phận cũ của nhân viên

p\_maBoPhan\_moi VARCHAR(20) -- Mã bộ phận mới của nhân viên

)

LANGUAGE plpgsql

AS $$

BEGIN

UPDATE "NhanVien"

SET "maNV" = p\_maBoPhan\_moi

WHERE "maNV" = p\_maNV AND "maNV" = p\_maBoPhan\_cu;

IF NOT FOUND THEN

RAISE EXCEPTION 'Không tìm thấy nhân viên với maNV: %, mã bộ phận: %', p\_maNV, p\_maBoPhan\_cu;

END IF;

EXCEPTION

WHEN OTHERS THEN

RAISE EXCEPTION 'Có lỗi xảy ra khi chuyển bộ phận: %', SQLERRM;

END;

$$;

**4. Đánh giá**

Hệ thống quản lý dữ liệu ngân hàng (KTBank) có một số điểm cần cải thiện và một số hạn chế như sau:

Áp dụng lý thuyết thiết kế cơ sở dữ liệu:

-Hệ thống chưa áp dụng được nhiều lý thuyết thiết kế cơ sở dữ liệu tiên tiến, dẫn đến cấu trúc cơ sở dữ liệu còn đơn giản và chưa tối ưu.

-Cần nghiên cứu và áp dụng thêm các nguyên lý thiết kế CSDL để nâng cao hiệu quả và khả năng mở rộng của hệ thống.

Các câu truy vấn SQL:

-Một số câu truy vấn trong hệ thống còn quá cơ bản, chưa tận dụng được sức mạnh của SQL để thực hiện các truy vấn phức tạp và tối ưu.

-Cần phát triển thêm các câu truy vấn nâng cao để hỗ trợ tốt hơn cho các báo cáo và phân tích dữ liệu.

Chức năng hệ thống:

-Hệ thống hiện tại có ít chức năng, chỉ bao gồm các chức năng cơ bản như quản lý khách hàng, tài khoản, và giao dịch.

-Chưa có các chức năng nâng cao như phân tích tài chính, quản lý rủi ro, hay hỗ trợ ra quyết định chiến lược.

-Cần bổ sung thêm các chức năng để hệ thống trở nên toàn diện và hữu ích hơn.

Dữ liệu trong hệ thống:

-Hệ thống hiện tại có ít dữ liệu, chưa đủ để thử nghiệm và kiểm tra tính hiệu quả của các chức năng quản lý và báo cáo.

-Cần bổ sung thêm dữ liệu mẫu đa dạng và phong phú hơn để kiểm tra và đánh giá hệ thống một cách toàn diện.

Phân loại tài khoản:

-Hệ thống chưa điều khiển được việc phân loại tài khoản tiết kiệm và tài khoản thông thường, gây khó khăn trong việc quản lý và theo dõi.

-Cần thêm chức năng để phân biệt rõ ràng giữa các loại tài khoản và áp dụng các quy định cụ thể cho từng loại.

Xử lý lãi suất tài khoản tiết kiệm:

-Hiện tại, lãi suất của tài khoản tiết kiệm bị đặt về 0 khi rút tiền, điều này không đúng với thực tế và quy định của các ngân hàng.

-Cần điều chỉnh lại cách tính lãi suất cho tài khoản tiết kiệm để phù hợp với thực tế và đảm bảo quyền lợi cho khách hang

1. **Kết luận/thu hoạch từ thực hiện đồ án**

Sau khi thực hiện và đánh giá dự án hệ thống quản lý dữ liệu ngân hàng (Banking Management System), nhóm đã rút ra được nhiều kinh nghiệm quý báu và nhận định về những điểm mạnh cũng như những hạn chế cần khắc phục:

**Hạn chế:**

**Áp dụng lý thuyết thiết kế cơ sở dữ liệu chưa đầy đủ**: Cần nghiên cứu và áp dụng thêm các nguyên lý thiết kế cơ sở dữ liệu để nâng cao hiệu quả và khả năng mở rộng của hệ thống.

**Câu truy vấn SQL còn đơn giản**: Cần phát triển thêm các câu truy vấn nâng cao để hỗ trợ tốt hơn cho các báo cáo và phân tích dữ liệu.

**Chức năng hệ thống còn ít**: Chưa có các chức năng nâng cao như phân tích tài chính, quản lý rủi ro, hay hỗ trợ ra quyết định chiến lược.

**Phân loại và quản lý tài khoản tiết kiệm chưa chính xác**: Cần điều chỉnh lại cách tính lãi suất và quản lý tài khoản tiết kiệm để phù hợp với thực tế và đảm bảo quyền lợi cho khách hàng.

**Thu hoạch từ thực hiện đồ án**

Kiến thức và kỹ năng học được:

Thiết kế và quản lý cơ sở dữ liệu: Nhóm đã nắm vững các kiến thức cơ bản về thiết kế và quản lý cơ sở dữ liệu, từ việc tạo bảng, quan hệ đến truy vấn dữ liệu bằng SQL.

Phát triển phần mềm: Nhóm đã học được quy trình phát triển một hệ thống phần mềm từ khâu thiết kế, lập trình đến kiểm thử và triển khai.

**Kinh nghiệm thực tiễn:**

Làm việc nhóm: Kinh nghiệm làm việc nhóm giúp nhóm học cách phân chia công việc, hợp tác và hỗ trợ lẫn nhau để hoàn thành dự án.

Giải quyết vấn đề: Quá trình phát triển hệ thống gặp nhiều khó khăn và thử thách, từ đó nhóm đã học được cách giải quyết vấn đề, tìm kiếm giải pháp và cải thiện hệ thống.

Thực hiện dự án thực tế: Dự án cung cấp cơ hội áp dụng các kiến thức đã học vào thực tế, giúp nhóm hiểu rõ hơn về quá trình phát triển và triển khai một hệ thống quản lý dữ liệu ngân hàng.

**Hướng phát triển trong tương lai**

Nâng cấp và mở rộng hệ thống: Dựa trên những hạn chế đã phát hiện, nhóm có thể tiếp tục nâng cấp và mở rộng hệ thống, bổ sung thêm các chức năng nâng cao và tối ưu hóa hiệu quả hoạt động.