

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC ĐẠI NAM



BÀI TẬP LỚN

TÊN HỌC PHẦN: HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU

ĐỀ TÀI: Quản lý tài chính cá nhân

STT	Mã sinh viên	Họ và tên	Ngày sinh	Lớp
1	1871040034	Đinh Đức Thọ	30/05/2005	KHMT 18-01
2	1871040031	Nguyễn Bá Anh Thái	18/08/2006	KHMT 18-01
3				

Hà Nội, năm 2025

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC ĐẠI NAM



BÀI TẬP LỚN

TÊN HỌC PHẦN: HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU

ĐỀ TÀI: Quản lý tài chính cá nhân

STT	Mã sinh viên	Họ và tên	Ngày sinh	Điểm	
				Bảng số	Bảng chữ
1	1871040034	Đinh Đức Thọ	30/05/2005		
2	1871040031	Nguyễn Bá Anh Thái	18/08/2006		
3					

CÁN BỘ CHẤM THI 1

CÁN BỘ CHẤM THI

Hà Nội, năm 2025

LỜI NÓI ĐẦU

Trong bối cảnh xã hội hiện đại, công nghệ thông tin đã và đang len lỏi vào mọi khía cạnh của đời sống. Cùng với sự phát triển đó, nhu cầu quản lý thông tin một cách khoa học, hiệu quả và chính xác ngày càng trở nên cấp thiết. Đối với mỗi cá nhân, việc quản lý tài chính hiệu quả là nền tảng cho một cuộc sống ổn định và thực hiện các mục tiêu tương lai. Tuy nhiên, việc theo dõi các khoản thu nhập, chi tiêu một cách thủ công thường tốn thời gian, dễ sai sót và khó đưa ra được cái nhìn tổng quan.

Nhận thức được tầm quan trọng của vấn đề này, việc ứng dụng hệ thống cơ sở dữ liệu để xây dựng một công cụ hỗ trợ quản lý tài chính cá nhân là một giải pháp vô cùng thiết thực. Hệ thống không chỉ giúp lưu trữ các giao dịch một cách có tổ chức mà còn cho phép người dùng dễ dàng truy vấn, phân tích và lập báo cáo về tình hình tài chính của mình.

Đó cũng chính là lý do nhóm chúng em, gồm **Đinh Đức Thọ** và **Nguyễn Bá Anh Thái**, đã chọn thực hiện đề tài "**Quản lý tài chính cá nhân**" cho bài tập lớn môn Hệ Quản trị Cơ sở dữ liệu.

Mục tiêu của đề tài là vận dụng các kiến thức đã học về phân tích, thiết kế và cài đặt cơ sở dữ liệu để xây dựng một hệ thống hoàn chỉnh, có khả năng quản lý thông tin người dùng, các tài khoản ngân hàng, các loại chi tiêu, đồng thời ghi nhận các giao dịch thu nhập và chi tiêu. Hệ thống cũng sẽ hỗ trợ các chức năng nâng cao như tạo báo cáo, thống kê và bảo mật thông tin.

Trong quá trình thực hiện đề tài, mặc dù đã rất cố gắng, nhưng do hạn chế về thời gian và kinh nghiệm, bài báo cáo không thể tránh khỏi những thiếu sót. Chúng em rất mong nhận được những ý kiến đóng góp quý báu từ thầy cô và các bạn để đề tài được hoàn thiện hơn.

Chúng em xin chân thành cảm ơn giảng viên bộ môn đã tận tình hướng dẫn, cung cấp những kiến thức nền tảng quý giá để chúng em có thể hoàn thành bài tập lớn này.

Xin chân thành cảm ơn!

MỤC LỤC

Nội dung

LỜI NÓI ĐẦU	3
CHƯƠNG 1. GIỚI THIỆU.....	6
1.1. Tóm tắt CSDL.....	6
1.2. Công cụ và công nghệ sử dụng.....	6
1.3. Bảng phân công.....	6
CHƯƠNG 2. PHÂN TÍCH YÊU CẦU HỆ THỐNG.....	8
2.1. Phân tích và yêu cầu cơ sở dữ liệu.....	8
2.1.1. Nghiệp vụ Người dùng (User Management).....	8
2.1.2. Nghiệp vụ Tài khoản Ngân hàng (Account Management).....	8
2.1.3. Nghiệp vụ Loại Chi tiêu (Category Management).....	8
2.1.4. Nghiệp vụ Giao dịch Thu nhập (Income Transaction).....	8
2.1.5. Nghiệp vụ Giao dịch Chi tiêu (Expense Transaction).....	8
2.2. Các chức năng chính của hệ thống.....	8
CHƯƠNG 3. THIẾT KẾ CƠ SỞ DỮ.....	10
3.2. Chuyển đổi mô hình ER thành lược đồ quan hệ.....	10
3.3. Cấu trúc các bảng dữ liệu.....	10
3.3.1. Bảng NGUOIDUNG.....	10
3.3.2. Bảng LOAICHITIEU.....	11
3.3.3. Bảng TAIKHOANNGANHANG.....	11
3.3.4. Bảng THUNHAP.....	12
CHƯƠNG 4. CÀI ĐẶT CƠ SỞ DỮ LIỆU.....	14
4.1. Tạo bảng.....	14

4.2. Chèn dữ liệu mẫu.....	15
4.3. Sơ đồ cơ sở dữ liệu.....	16
4.4. Sơ Đồ Quan Hệ.....	17
CHƯƠNG 5. CÁC ĐỐI TƯỢNG CƠ SỞ DỮ LIỆU NÂNG CAO.....	17
5.1. Cài đặt các chỉ mục (Indexes).....	17
5.1.1. Tên chỉ mục 1: idx_CHITIEU_Ngay_NguoiDung.....	17
5.1.2. Tên chỉ mục 2: idx_THUNHAP_Ngay_NguoiDung.....	17
5.2. Views.....	18
5.2.1. Tên view 1: vw_BaoCaoChiTieu_TheoLoai.....	18
5.3. Thủ tục (Stored Procedures).....	18
5.3.1. Tên thủ tục 1: sp_GhiNhanGiaoDichThuNhap.....	18
5.4. Hàm người dùng định nghĩa (User Defined Functions).....	19
5.4.1. Tên hàm 1: fn_TinhTongSoDuHienTai.....	19
5.5. Trigger.....	20
5.5.1. Tên trigger 1: trg_CapNhatSoDu_ThuNhap.....	20
5.5.2. Tên trigger 2: trg_CapNhatSoDu_ChiTieu.....	20
CHƯƠNG 6. BẢO MẬT VÀ QUẢN TRỊ CƠ SỞ DỮ LIỆU.....	22
6.1. Quản lý người dùng và quyền truy cập.....	22
6.2. Bảo mật cơ sở dữ liệu.....	23
6.3. Quản lý sao lưu và phục hồi.....	23
6.4. Quản lý hiệu suất cơ sở dữ liệu.....	24
KẾT LUẬN.....	25
DANH MỤC TÀI LIỆU THAM KHẢO.....	26

CHƯƠNG 1. GIỚI THIỆU

1.1. Tóm tắt CSDL

Cơ sở dữ liệu (CSDL) là thành phần cốt lõi của mọi hệ thống thông tin hiện đại. Hệ quản trị cơ sở dữ liệu (HQTCSDL) đóng vai trò trung gian, cung cấp các công cụ để lưu trữ, truy xuất và quản lý dữ liệu một cách an toàn và hiệu quả.

Đối với đề tài **Quản lý Tài chính Cá nhân**, CSDL được xây dựng để giải quyết vấn đề quản lý dữ liệu tài chính phân tán và khó kiểm soát. CSDL sẽ là nơi lưu trữ các khoản **thu nhập**, **chi tiêu**, thông tin **người dùng** và **tài khoản ngân hàng** một cách có hệ thống, giúp người dùng theo dõi dòng tiền, phân tích thói quen chi tiêu và đưa ra các quyết định tài chính sáng suốt.

1.2. Công cụ và công nghệ sử dụng

Các công cụ và công nghệ được lựa chọn để triển khai dự án, đảm bảo tính ổn định và khả năng mở rộng:

Hệ quản trị CSDL: Microsoft SQL Server 2019 (hoặc phiên bản mới hơn). Đây là HQTCSDL mạnh mẽ, phù hợp với các nghiệp vụ giao dịch (OLTP) và cung cấp các tính năng nâng cao như Views, Stored Procedures, và Triggers.

Công cụ quản lý và thiết kế: SQL Server Management Studio (SSMS). SSMS được sử dụng để thiết kế lược đồ, viết và kiểm thử các đối tượng CSDL, quản lý bảo mật và thực hiện các tác vụ quản trị.

Ngôn ngữ lập trình: Transact-SQL (T-SQL) để tạo, thao tác và quản lý dữ liệu.

1.3. Bảng phân công

Bảng phân công chi tiết công việc cho hai thành viên:

STT	Tên công việc	Người thực hiện
1	Chương 1 & 7: Tổng quan, Lời nói đầu, Kết luận	Nguyễn Bá Anh Thái
2	Chương 2: Phân tích yêu cầu hệ thống	Đinh Đức Thọ
3	Chương 3.1 & 6: Thiết kế ERD, Bảo mật & Quản trị	Đinh Đức Thọ
4	Chương 3.2, 3.3 & 4: Thiết kế chi tiết, Cài đặt CSDL	Nguyễn Bá Anh Thái
5	Chương 5.1, 5.2, 5.5: Cài đặt Indexes, Views, Triggers	Đinh Đức Thọ
6	Chương 5.3, 5.4: Cài đặt Stored Procedures, Functions	Nguyễn Bá Anh Thái

CHƯƠNG 2. PHÂN TÍCH YÊU CẦU HỆ THỐNG

2.1. Phân tích và yêu cầu cơ sở dữ liệu

Dựa trên gợi ý của đề tài, hệ thống cần xử lý các nghiệp vụ (yêu cầu dữ liệu) sau:

2.1.1. Nghiệp vụ Người dùng (User Management)

N Yêu cầu: Lưu trữ thông tin định danh của người dùng (tên, email, số điện thoại) để phân biệt các khoản tài chính cá nhân. Email phải là duy nhất.

2.1.2. Nghiệp vụ Tài khoản Ngân hàng (Account Management)

Yêu cầu: Mỗi người dùng có thể liên kết nhiều tài khoản ngân hàng. Cần lưu tên ngân hàng và số dư hiện tại của tài khoản.

2.1.3. Nghiệp vụ Loại Chi tiêu (Category Management)

Yêu cầu: Cần có danh mục các loại chi tiêu để phân loại giao dịch (ví dụ: Ăn uống, Thuê nhà, Di chuyển).

2.1.4. Nghiệp vụ Giao dịch Thu nhập (Income Transaction)

Yêu cầu: Ghi nhận số tiền và ngày nhận của các khoản thu nhập. Mỗi khoản thu nhập phải được liên kết với một người dùng và một tài khoản cụ thể để cập nhật số dư.

2.1.5. Nghiệp vụ Giao dịch Chi tiêu (Expense Transaction)

Yêu cầu: Ghi nhận số tiền, ngày chi tiêu, loại chi tiêu. Mỗi khoản chi tiêu phải được liên kết với một người dùng và một tài khoản để cập nhật số dư.

2.2. Các chức năng chính của hệ thống

CSDL phải hỗ trợ các chức năng chính:

Quản lý Master Data: Cho phép thêm, sửa, xóa thông tin trong các bảng NGUOIDUNG, TAIKHOANNGANHANG, LOAICHIITIEU.

Quản lý Giao dịch: Cho phép nhập mới các bản ghi THUNHAP và CHITIEU.

Cập nhật số dư tự động: Tự động điều chỉnh SoDu trong TAIKHOANNGANHANG khi một giao dịch được ghi nhận (sử dụng Trigger).

Báo cáo tổng hợp: Thống kê tổng Thu nhập và tổng Chi tiêu theo các khoảng thời gian (ngày, tháng, năm).

Phân tích chi tiêu: Hiện thị chi tiêu theo tỷ lệ của từng loại chi tiêu (LOAIIsCHIITIEU).

Truy vấn nhanh: Xem tổng số dư hiện tại của tất cả tài khoản của một người dùng.

CHƯƠNG 3. THIẾT KẾ CƠ SỞ DỮ

Mô hình ERD bao gồm các thực thể và mối quan hệ sau:

Thực thể NGUOIDUNG: Người tham gia hệ thống.

Thực thể TAIKHOANNGANHANG: Tài sản tiền mặt/ngân hàng của người dùng.

Quan hệ 1-N: NGUOIDUNG 1 - N TAIKHOANNGANHANG.

Thực thể LOAICHIITIEU: Danh mục phân loại giao dịch.

Thực thể THUNHAP: Các giao dịch thu tiền vào.

Quan hệ 1-N: NGUOIDUNG 1 - N THUNHAP.

Thực thể CHITIEU: Các giao dịch chi tiền ra.

Quan hệ 1-N: NGUOIDUNG 1 - N CHITIEU.

Quan hệ 1-N: LOAICHIITIEU 1 - N CHITIEU.

3.2. Chuyển đổi mô hình ER thành lược đồ quan hệ

Chuyển đổi các thực thể thành lược đồ quan hệ, bao gồm các ràng buộc chính (PK) và ngoại (FK).

NGUOIDUNG (<u>MaNguoiDung</u>, TenNguoiDung, Email, SoDienThoai)

LOAICHIITIEU (<u>MaLoaiChiTieu</u>, TenLoaiChiTieu)

TAIKHOANNGANHANG (<u>MaTaiKhoan</u>, MaNguoiDung, SoDu, NganHang)

THUNHAP (<u>MaThuNhap</u>, MaNguoiDung, MaTaiKhoan, SoTien, NgayThuNhap)

CHITIEU (<u>MaChiTieu</u>, MaNguoiDung, MaLoaiChiTieu, MaTaiKhoan, SoTien, NgayChiTieu)

3.3. Cấu trúc các bảng dữ liệu

3.3.1. Bảng NGUOIDUNG

Tên trường	Kiểu dữ liệu	Độ rộng	Ràng buộc	Giải thích
MaNguoiDung	INT		PK,IDENTITY	Mã người dùng
TenNguoiDung	NVARCHAR	100	NOT NULL	Tên người dùng
Email	VARCHAR	100	NOT NULL, UNIQUE	Email
SoDienThoai	VARCHAR	15		Số điện thoại

3.3.2. Bảng *LOAICHITIEU*

Tên trường	Kiểu dữ liệu	Độ rộng	Ràng buộc	Giải thích
<i>MaLoaiChiTieu</i>	<i>INT</i>		<i>PK, IDENTITY(1,1)</i>	<i>Mã loại chi tiêu (PK)</i>
<i>TenLoaiChiTieu</i>	<i>NVARCHAR</i>	<i>100</i>	<i>NOT NULL</i>	<i>Tên loại chi tiêu</i>

3.3.3. Bảng *TAIKHOANNGANHANG*

Tên trường	Kiểu dữ liệu	Độ rộng	Ràng buộc	Giải thích
<i>MaTaiKhoan</i>	<i>INT</i>		<i>PK, IDENTITY(1,1)</i>	<i>Mã tài khoản (PK)</i>
<i>MaNguoiDung</i>	<i>INT</i>		<i>FK (NGUOIDUNG), NOT NULL</i>	<i>Mã người dùng (FK)</i>
<i>NganHang</i>	<i>NVARCHAR</i>	<i>100</i>	<i>NOT NULL</i>	<i>Tên ngân hàng</i>
<i>SoDu</i>	<i>DECIMAL</i>	<i>18,2</i>	<i>NOT NULL, DEFAULT 0</i>	<i>Số dư hiện tại</i>

3.3.4. Bảng THUNHAP

Tên trường	Kiểu Dữ Liệu	Độ Rộng	Ràng buộc	Giải thích
MaThuNhap	INT		PK, IDENTITY(1,1)	Mã thu nhập (PK)
MaNguoiDung	INT		FK (NGUOIDUNG), NOT NULL	Mã người dùng (FK)
MaTaiKhoan	INT		FK (TAIKHOANNGANHANG), NOT NULL	Tài khoản nhận tiền (FK)
SoTien	DECIMAL	18,2	NOT NULL, CHECK (> 0)	Số tiền
NgayThuNhap	DATE		NOT NULL, DEFAULT GETDATE()	Ngày thu nhập

3.3.5

Tên trường	Kiểu dữ liệu	Độ rộng	Ràng buộc	Giải thích
MaChiTieu	INT		PK, IDENTITY(1,1)	Mã chi tiêu (PK)
MaNguoiDung	INT		FK (NGUOIDUNG), NOT NULL	Mã người dùng (FK)
MaLoaiChiTieu	INT		FK (LOAICHIITIEU), NOT NULL	Mã loại chi tiêu (FK)
MaTaiKhoan	INT		FK (TAIKHOANNGANHANG), NOT NULL	Tài khoản chi tiền (FK)
SoTien	DECIMAL	18, 2	NOT NULL, CHECK (> 0)	Số tiền
NgayChiTieu	DATE		NOT NULL, DEFAULT GETDATE()	Ngày chi tiêu

CHƯƠNG 4. CÀI ĐẶT CƠ SỞ DỮ LIỆU

4.1. Tạo bảng

Các câu lệnh SQL được sử dụng để tạo CSDL và 5 bảng dữ liệu với đầy đủ các ràng buộc:

-- 1. Tạo Database

```
CREATE DATABASE QLTaiChinhCaNhan;
GO
USE QLTaiChinhCaNhan;
GO
```

-- 2. Tạo bảng NGUOIDUNG

```
CREATE TABLE NGUOIDUNG (
    MaNguoiDung INT IDENTITY(1,1) PRIMARY KEY,
    TenNguoiDung NVARCHAR(100) NOT NULL,
    Email VARCHAR(100) NOT NULL UNIQUE,
    SoDienThoai VARCHAR(15)
);
```

-- 3. Tạo bảng LOAICHITIEU

```
CREATE TABLE LOAICHITIEU (
    MaLoaiChiTieu INT IDENTITY(1,1) PRIMARY KEY,
    TenLoaiChiTieu NVARCHAR(100) NOT NULL
);
```

-- 4. Tạo bảng TAIKHOANNGANHANG

```
CREATE TABLE TAIKHOANNGANHANG (
    MaTaiKhoan INT IDENTITY(1,1) PRIMARY KEY,
    MaNguoiDung INT NOT NULL,
    NganHang NVARCHAR(100) NOT NULL,
    SoDu DECIMAL(18, 2) NOT NULL DEFAULT 0,
    CONSTRAINT FK_TK_NGUOIDUNG FOREIGN KEY (MaNguoiDung) REFERENCES
    NGUOIDUNG(MaNguoiDung),
    CONSTRAINT CHK_SoDu CHECK (SoDu >= 0) -- Số dư không âm
);
```

-- 5. Tạo bảng THUNHAP

```
CREATE TABLE THUNHAP (
    MaThuNhap INT IDENTITY(1,1) PRIMARY KEY,
    MaNguoiDung INT NOT NULL,
    MaTaiKhoan INT NOT NULL,
    SoTien DECIMAL(18, 2) NOT NULL,
    NgayThuNhap DATE NOT NULL DEFAULT GETDATE(),
    CONSTRAINT FK_TN_NGUOIDUNG FOREIGN KEY (MaNguoiDung) REFERENCES
    NGUOIDUNG(MaNguoiDung),
    CONSTRAINT FK_TN_TAIKHOAN FOREIGN KEY (MaTaiKhoan) REFERENCES
    TAIKHOANNGANHANG(MaTaiKhoan),
    CONSTRAINT CHK_TN_SoTien CHECK (SoTien > 0)
);
```

-- 6. Tạo bảng CHITIEU

```
CREATE TABLE CHITIEU (
    MaChiTieu INT IDENTITY(1,1) PRIMARY KEY,
    MaNguoiDung INT NOT NULL,
    MaLoaiChiTieu INT NOT NULL,
```

```

MaTaiKhoan INT NOT NULL,
SoTien DECIMAL(18, 2) NOT NULL,
NgàyChiTieu DATE NOT NULL DEFAULT GETDATE(),
CONSTRAINT FK_CT_NGUOIDUNG FOREIGN KEY (MaNguoiDung) REFERENCES
NGUOIDUNG(MaNguoiDung),
CONSTRAINT FK_CT_LOAICHIITIEU FOREIGN KEY (MaLoaiChiTieu) REFERENCES
LOAICHIITIEU(MaLoaiChiTieu),
CONSTRAINT FK_CT_TAIKHOAN FOREIGN KEY (MaTaiKhoan) REFERENCES
TAIKHOANNGANHANG(MaTaiKhoan),
CONSTRAINT CHK_CT_SoTien CHECK (SoTien > 0)
);

```

4.2. Chèn dữ liệu mẫu

Các câu lệnh **INSERT** để tạo 5-10 bản ghi mẫu cho mỗi bảng.

```

-- 1. Chèn dữ liệu NGUOIDUNG (2 bản ghi)
INSERT INTO NGUOIDUNG (TenNguoiDung, Email, SoDienThoai) VALUES
(N'Đinh Đức Thọ', 'thodd@example.com', '0912345678'),
(N'Nguyễn Bá Anh Thái', 'thainba@example.com', '0987654321');

-- 2. Chèn dữ liệu LOAICHIITIEU (5 bản ghi)
INSERT INTO LOAICHIITIEU (TenLoaiChiTieu) VALUES
(N'Ăn uống'), (N'Di chuyển'), (N'Thuê nhà'), (N'Giải trí'), (N'Tiết kiệm');

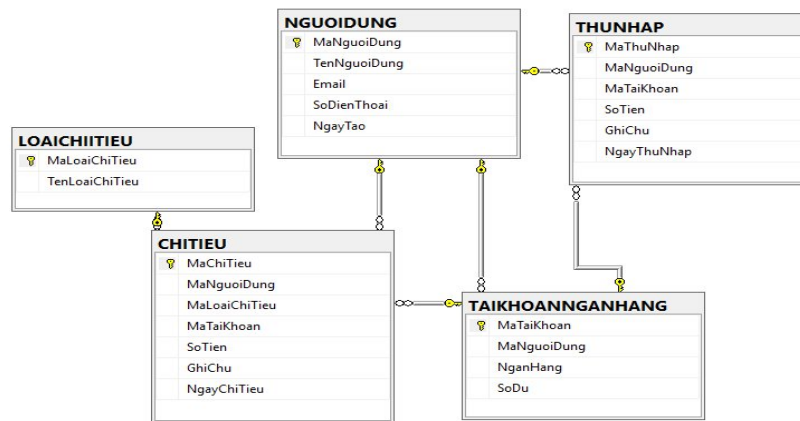
-- 3. Chèn dữ liệu TAIKHOANNGANHANG (3 bản ghi)
INSERT INTO TAIKHOANNGANHANG (MaNguoiDung, NganHang, SoDu) VALUES
(1, N'Vietcombank', 5000000.00), -- Thọ
(1, N'Techcombank', 2500000.00), -- Thọ
(2, N'ACB', 10000000.00); -- Thái

-- 4. Chèn dữ liệu THUNHAP (5 bản ghi)
INSERT INTO THUNHAP (MaNguoiDung, MaTaiKhoan, SoTien, NgàyThuNhap) VALUES
(1, 1, 15000000.00, '2025-11-01'), -- Thọ VCB
(2, 3, 20000000.00, '2025-11-05'), -- Thái ACB
(1, 2, 5000000.00, '2025-11-10'), -- Thọ TCB
(1, 1, 200000.00, '2025-11-15'), -- Thọ VCB
(2, 3, 300000.00, '2025-11-16'); -- Thái ACB

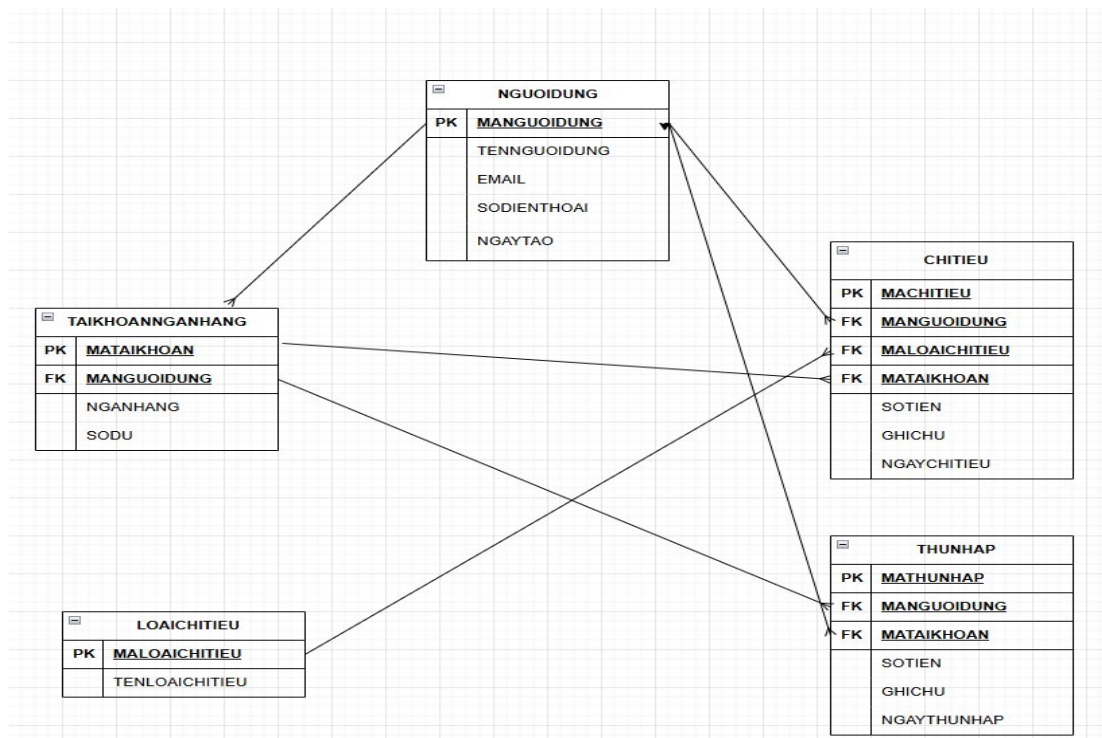
-- 5. Chèn dữ liệu CHITIEU (7 bản ghi)
INSERT INTO CHITIEU (MaNguoiDung, MaLoaiChiTieu, MaTaiKhoan, SoTien, NgàyChiTieu) VALUES
(1, 1, 1, 50000.00, '2025-11-16'), -- Thọ, Ăn uống, VCB
(1, 2, 1, 30000.00, '2025-11-16'), -- Thọ, Di chuyển, VCB
(2, 3, 3, 3000000.00, '2025-11-17'), -- Thái, Thuê nhà, ACB
(2, 1, 3, 120000.00, '2025-11-17'), -- Thái, Ăn uống, ACB
(1, 4, 2, 500000.00, '2025-11-18'), -- Thọ, Giải trí, TCB
(1, 5, 1, 1000000.00, '2025-11-18'), -- Thọ, Tiết kiệm, VCB
(2, 4, 3, 200000.00, '2025-11-18'); -- Thái, Giải trí, ACB

```

4.3. Sơ đồ cơ sở dữ liệu



4.4. Sơ Đồ Quan Hệ



CHƯƠNG 5. CÁC ĐỐI TƯỢNG CƠ SỞ DỮ LIỆU NÂNG CAO

5.1. Cài đặt các chỉ mục (Indexes)

Chỉ mục được cài đặt để tăng tốc độ truy vấn trên các cột thường xuyên được sử dụng trong mệnh đề WHERE và ORDER BY.

5.1.1. Tên chỉ mục 1: *idx_CHITIEU_Ngay_NguoiDung*

Mục đích: Tối ưu hóa truy vấn khi lập báo cáo chi tiêu theo ngày của một người dùng cụ thể.

Câu lệnh SQL:

```
CREATE NONCLUSTERED INDEX idx_CHITIEU_Ngay_NguoiDung  
ON CHITIEU(MaNguoiDung, NgayChiTieu DESC);
```

5.1.2. Tên chỉ mục 2: *idx_THUNHAP_Ngay_NguoiDung*

Mục đích: Tối ưu hóa truy vấn khi lập báo cáo thu nhập theo ngày của một người dùng cụ thể.

Câu lệnh SQL:

```
CREATE NONCLUSTERED INDEX idx_THUNHAP_Ngay_NguoiDung  
ON THUNHAP(MaNguoiDung, NgayThuNhap DESC);
```

5.2. Views

Views được sử dụng để đơn giản hóa các truy vấn phức tạp và kiểm soát quyền truy cập dữ liệu thô.

5.2.1. Tên view 1: *vw_BaoCaoChiTieu_TheoLoai*

Mục đích: Thống kê tổng chi tiêu của mỗi người dùng, phân loại theo từng nhóm chi tiêu.

Câu lệnh tạo view:

```

CREATE VIEW vw_BaoCaoChiTieu_TheoLoai
AS
SELECT
    n.TenNguoiDung,
    l.TenLoaiChiTieu,
    SUM(c.SoTien) AS TongChiTieu,
    COUNT(c.MaChiTieu) AS SoGiaoDich
FROM CHITIEU c
JOIN NGUOIDUNG n ON c.MaNguoiDung = n.MaNguoiDung
JOIN LOAICHITIEU l ON c.MaLoaiChiTieu = l.MaLoaiChiTieu
GROUP BY n.TenNguoiDung, l.TenLoaiChiTieu;

```

5.3. Thủ tục (Stored Procedures)

SPs được sử dụng để thực hiện các nghiệp vụ phức tạp và tự động hóa các bước kiểm tra dữ liệu.

5.3.1. Tên thủ tục 1: *sp_GhiNhanGiaoDichThuNhap*

Mục đích: Thực hiện chèn một giao dịch thu nhập mới.

Câu lệnh tạo thủ tục:

```

CREATE PROCEDURE sp_GhiNhanGiaoDichThuNhap
    @MaNguoiDung INT,
    @MaTaiKhoan INT,
    @SoTien DECIMAL(18, 2),
    @NgayThuNhap DATE = NULL
AS
BEGIN
    -- Kiểm tra người dùng và tài khoản có tồn tại không
    IF NOT EXISTS (SELECT 1 FROM NGUOIDUNG WHERE MaNguoiDung = @MaNguoiDung)
        THROW 51000, N'Người dùng không tồn tại.', 1;
    IF NOT EXISTS (SELECT 1 FROM TAIKHOANNGANHANG WHERE MaTaiKhoan = @MaTaiKhoan
    AND MaNguoiDung = @MaNguoiDung)
        THROW 51001, N'Tài khoản không tồn tại hoặc không thuộc người dùng này.', 1;

    -- Chèn giao dịch
    INSERT INTO THUNHAP (MaNguoiDung, MaTaiKhoan, SoTien, NgayThuNhap)
    VALUES (@MaNguoiDung, @MaTaiKhoan, @SoTien, ISNULL(@NgayThuNhap, GETDATE()));

    -- Ghi chú: Trigger (5.5) sẽ tự động cập nhật SoDu.
    END

```

Câu lệnh gọi thủ tục:

-- Ví dụ gọi thủ tục (Giả định MaNguoiDung 2, MaTaiKhoan 3)

EXEC sp_GhiNhanGiaoDichThuNhap @MaNguoiDung = 2, @MaTaiKhoan = 3, @SoTien = 1500000.00;

5.4. Hàm người dùng định nghĩa (User Defined Functions)

5.4.1. Tên hàm 1: fn_TinhTongSoDuHienTai

Mục đích: Tính tổng số dư hiện tại của tất cả tài khoản ngân hàng của một người dùng (Hàm Scalar-valued).

Câu lệnh tạo hàm:

```
CREATE FUNCTION fn_TinhTongSoDuHienTai
(
    @MaNguoiDung INT
)
RETURNS DECIMAL(18, 2)
AS
BEGIN
    DECLARE @TongSoDu DECIMAL(18, 2);

    SELECT @TongSoDu = ISNULL(SUM(SoDu), 0)
    FROM TAIKHOANNGANHANG
    WHERE MaNguoiDung = @MaNguoiDung;

    RETURN @TongSoDu;
END
```

Câu lệnh gọi hàm:

```
SELECT dbo.fn_TinhTongSoDuHienTai(1) AS TongSoDuCuaTho;
```

5.5. Trigger

Trigger được sử dụng để tự động cập nhật số dư tài khoản khi có giao dịch phát sinh, đảm bảo tính toàn vẹn và đồng bộ dữ liệu.

5.5.1. Tên trigger 1: trg_CapNhatSoDu_ThuNhap

Mục đích: Tự động cộng tiền vào TAIKHOANNGANHANG khi có một bản ghi THUNHAP mới được chèn.

Câu lệnh tạo trigger:

```
CREATE TRIGGER trg_CapNhatSoDu_ThuNhap
ON THUNHAP
AFTER INSERT
AS
BEGIN
    UPDATE t
    SET SoDu = SoDu + i.SoTien
    FROM TAIKHOANNGANHANG t
    JOIN inserted i ON t.MaTaiKhoan = i.MaTaiKhoan;
END
```

Câu lệnh gọi trigger: (Thực hiện lệnh INSERT, ví dụ đã thực hiện ở 4.2)

```
-- Thêm một khoản thu nhập mới (sẽ tự động kích hoạt trigger)
INSERT INTO THUNHAP (MaNguoiDung, MaTaiKhoan, SoTien) VALUES (2, 3, 500000.00);
```

5.5.2. Tên trigger 2: trg_CapNhatSoDu_ChiTieu

Mục đích: Tự động trừ tiền khỏi TAIKHOANNGANHANG khi có một bản ghi CHITIEU mới được chèn. Trigger này sẽ kiểm tra số dư trước khi trừ.

Câu lệnh tạo trigger:

```
CREATE TRIGGER trg_CapNhatSoDu_ChiTieu
ON CHITIEU
INSTEAD OF INSERT -- Thay vì AFTER INSERT, dùng INSTEAD OF để kiểm tra
AS
BEGIN
    -- 1. Kiểm tra số dư trước khi cho phép giao dịch
    IF EXISTS (
        SELECT 1
        FROM inserted i
        JOIN TAIKHOANNGANHANG t ON i.MaTaiKhoan = t.MaTaiKhoan
        WHERE t.SoDu < i.SoTien
    )
    BEGIN
        -- Báo lỗi nếu số dư không đủ
        THROW 52000, N'Số dư tài khoản không đủ để thực hiện giao dịch chi tiêu.', 1;
        RETURN;
    END

    -- 2. Nếu đủ, thực hiện chèn ChiTieu và cập nhật SoDu
    INSERT INTO CHITIEU (MaNguoiDung, MaLoaiChiTieu, MaTaiKhoan, SoTien, NgayChiTieu)
    SELECT MaNguoiDung, MaLoaiChiTieu, MaTaiKhoan, SoTien, NgayChiTieu FROM inserted;

    UPDATE t
    SET SoDu = SoDu - i.SoTien
    FROM TAIKHOANNGANHANG t
```

```
JOIN inserted i ON t.MaTaiKhoan = i.MaTaiKhoan;  
END
```

Câu lệnh gọi trigger:

-- Thêm một khoản chi tiêu mới (sẽ tự động kích hoạt trigger)

```
INSERT INTO CHITIEU (MaNguoiDung, MaLoaiChiTieu, MaTaiKhoan, SoTien) VALUES (1, 1, 1,  
10000.00);
```

CHƯƠNG 6. BẢO MẬT VÀ QUẢN TRỊ CƠ SỞ DỮ LIỆU

6.1. Quản lý người dùng và quyền truy cập

Tạo Login (Cấp Server) và User (Cấp Database): Tạo tài khoản đăng nhập SQL Server và liên kết với tài khoản người dùng cơ sở dữ liệu.

Tạo Roles (Vai trò): Định nghĩa hai vai trò chính:

role_UserApp: Dành cho các ứng dụng hoặc người dùng chỉ thực hiện ghi nhận giao dịch và xem báo cáo.

role_DBA: Dành cho quản trị viên CSDL.

Phân quyền (GRANT/REVOKE):

```
-- 1. Tạo Roles
CREATE ROLE role_UserApp;
CREATE ROLE role_DBA;

-- 2. Cấp quyền cho role_UserApp
-- Chỉ cho phép thực thi các Stored Procedure
GRANT EXECUTE ON sp_GhiNhanGiaoDichThuNhap TO role_UserApp;
GRANT EXECUTE ON sp_BaoCaoCanDoiTaiChinh TO role_UserApp;
-- Chỉ cho phép SELECT trên các Views
GRANT SELECT ON vw_BaoCaoChiTieu_TheoLoai TO role_UserApp;

-- 3. Ngăn chặn truy cập trực tiếp vào các bảng nhạy cảm (CHITIEU, TAIKHOANNGANHANG)
DENY SELECT ON CHITIEU TO role_UserApp;
DENY SELECT ON THUNHAP TO role_UserApp;
DENY SELECT ON TAIKHOANNGANHANG TO role_UserApp;
```

6.2. Bảo mật cơ sở dữ liệu

Bảo mật qua Stored Procedures: Bằng cách chỉ cấp quyền EXECUTE cho role_UserApp, người dùng không thể viết các lệnh SQL tùy ý, ngăn chặn hiệu quả tấn công SQL Injection và đảm bảo logic nghiệp vụ (ví dụ: kiểm tra số dư) luôn được tuân thủ.

Mã hóa dữ liệu (Encryption): Có thể sử dụng tính năng **Always Encrypted** của SQL Server để mã hóa các cột dữ liệu nhạy cảm như Email hoặc SoDienThoai trong bảng NGUOIDUNG, đảm bảo chỉ có ứng dụng hoặc người dùng có khóa truy cập mới có thể đọc được dữ liệu.

Sử dụng Views: Các Views (như vw_BaoCaoChiTieu_TheoLoai) được tạo để cung cấp dữ liệu tổng hợp thay vì dữ liệu thô, hạn chế người dùng xem thông tin chi tiết không cần thiết.

6.3. Quản lý sao lưu và phục hồi

Mô hình phục hồi (Recovery Model): Chọn Full Recovery Model để có thể phục hồi dữ liệu về bất kỳ thời điểm nào (Point-in-Time Recovery), cần thiết cho hệ thống giao dịch tài chính.

Chiến lược sao lưu (Backup Strategy):

Sao lưu Toàn bộ (Full Backup): Thực hiện 1 lần/tuần (Ví dụ: 00:00 Chủ Nhật).

Sao lưu Khác biệt (Differential Backup): Thực hiện 1 lần/ngày (Ví dụ: 01:00 các ngày trong tuần).

Sao lưu Log Giao dịch (Transaction Log Backup): Thực hiện 1 lần/giờ, do dữ liệu tài chính thay đổi liên tục, cần đảm bảo mất mát dữ liệu tối thiểu.

Hướng dẫn phục hồi: Phục hồi theo trình tự: FULL backup -> DIFFERENTIAL backup -> Chuỗi TRANSACTION LOG backup cuối cùng (sử dụng WITH RECOVERY cho bước cuối).

6.4. Quản lý hiệu suất cơ sở dữ liệu

Tối ưu hóa Index: Đảm bảo các chỉ mục đã tạo (mục 5.1) được sử dụng hiệu quả. Định kỳ chạy DBCC SHOW_STATISTICS và sys.dm_db_index_usage_stats để kiểm tra hiệu suất Index.

Tối ưu hóa Truy vấn:

Thường xuyên kiểm tra **Execution Plan** cho các truy vấn báo cáo phức tạp (như sp_BaoCaoCanDoiTaiChinh).

Sử dụng lệnh SET STATISTICS TIME ON và SET STATISTICS IO ON để đo lường chi phí truy vấn và chỉnh sửa code T-SQL để giảm thiểu Table Scan.

Định kỳ bảo trì: Thiết lập các công việc định kỳ (Maintenance Plans) để **Rebuild/Reorganize Indexes** và **Update Statistics** nhằm duy trì hiệu suất truy vấn tối ưu.

KẾT LUẬN

Báo cáo Bài tập lớn đã hoàn thành việc xây dựng và triển khai một CSDL hoàn chỉnh cho hệ thống Quản lý Tài chính Cá nhân:

- Đã phân tích, thiết kế mô hình ERD và lược đồ quan hệ với 5 thực thể chính.
- Đã cài đặt CSDL trên SQL Server, thiết lập đầy đủ các ràng buộc về tính toàn vẹn dữ liệu (PK, FK, CHECK, UNIQUE).
- Đã xây dựng thành công các đối tượng CSDL nâng cao như Views để đơn giản hóa truy vấn, Stored Procedures để tự động hóa nghiệp vụ, và Triggers để đảm bảo tính đồng bộ (tự động cập nhật số dư).
- Đã đề xuất các giải pháp chi tiết về Bảo mật (phân quyền) và Quản trị (sao lưu/tối ưu hóa) CSDL.

Để phát triển hệ thống Quản lý Tài chính Cá nhân trong tương lai, nhóm nghiên cứu đưa ra các khuyến nghị sau:

- **Mở rộng nghiệp vụ Ngân sách (Budgeting):** Thêm một bảng NGANSACH để người dùng có thể đặt giới hạn chi tiêu cho từng LOAICHITIEU hàng tháng. Xây dựng Trigger hoặc Stored Procedure để cảnh báo/ngăn chặn giao dịch khi vượt quá ngân sách.
- **Hỗ trợ Vay và Nợ:** Mở rộng CSDL để quản lý các khoản vay (tiền đi) và các khoản nợ (tiền đến), bao gồm lịch trả nợ và tính lãi suất.
- **Tối ưu hóa hiệu năng chuyên sâu:** Khi dữ liệu tăng lên, cần sử dụng công cụ SQL Server Profiler để theo dõi các truy vấn chạy chậm nhất và thực hiện tinh chỉnh (tuning) mức độ thấp (ví dụ: sử dụng Partitioning cho các bảng giao dịch lớn).

DANH MỤC TÀI LIỆU THAM KHẢO

- [1]. Nguyễn Hồng Sơn (2007), *Giáo trình hệ thống Mạng máy tính CCNA* (Semester 1), NXB Lao động xã hội.
- [2]. Phạm Quốc Hùng (2017), *Đề cương bài giảng Mạng máy tính*, Đại học SPKT Hưng Yên.
- [3]. James F. Kurose and Keith W. Ross (2013), *Computer Networking: A top-down approach sixth Edition*, Pearson Education.