

**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 NHÀ HÀNG**

STT	Mã sinh viên	Họ và tên	Ngày sinh	Lớp
1	1871040026	Đoàn Đức Phúc	09/08/2005	KHMT 18-01
2	1871040032	Nguyễn Văn Thành	22/04/2006	KHMT 18-01

**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 NHÀ HÀNG**

STT	Mã sinh viên	Họ và tên	Ngày sinh	Điểm	
				Bảng số	Bảng chữ
1	1871040026	Đoàn Đức Phúc	09/08/2005		
2	1871040032	Nguyễn Văn Thành	22/04/2006		

**CÁN BỘ CHẤM THI 1**

**CÁN BỘ CHẤM THI 2**

**Hà Nội, năm 2025**

## MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU.....	8
1.1.    Tóm tắt về cơ sở dữ liệu .....	8
1.1.1.    Giới thiệu .....	8
1.1.2.    Vai trò trong việc giải quyết vấn đề của đề tài .....	8
1.2.    Công cụ và công nghệ sử dụng.....	8
1.2.1.    SQL Server .....	8
1.2.2. SQL Server Management Studio .....	9
1.2.3. SQL Server Configuration Manager.....	9
1.3. Bảng phân công.....	9
CHƯƠNG 2: PHÂN TÍCH YÊU CẦU HỆ THỐNG.....	11
2.1. Phân tích và yêu cầu cơ sở dữ liệu.....	11
2.2. Các chức năng chính của hệ thống .....	11
CHƯƠNG 3: THIẾT KẾ CƠ SỞ DỮ LIỆU .....	13
3.1. Thiết kế mô hình thực thể quan hệ (ER).....	13
3.1.1. Thực thể và thuộc tính .....	13
3.1.2. Mối quan hệ .....	14
3.2. Chuyển đổi mô hình ER thành lược đồ quan hệ.....	15
3.3. Cấu trúc các bảng dữ liệu.....	15
CHƯƠNG 4: CÀI ĐẶT CƠ SỞ DỮ LIỆU .....	18
4.1. Tạo bảng.....	18
4.2. Sơ đồ luồng dữ liệu.....	22
CHƯƠNG 5: CÁC ĐỐI TƯỢNG CƠ SỞ DỮ LIỆU NÂNG CAO.....	23
5.1. Cài đặt các chỉ mục (Indexes).....	23

5.2. Views .....	23
5.3 Thủ tục (Stored Procedures) .....	24
5.3.1. Thủ tục 1 .....	24
5.3.2. Thủ tục 2 .....	26
5.3.3. Thủ tục 3 .....	28
5.4. Hàm người dùng định nghĩa (User Defined Functions) .....	29
5.5. Trigger.....	30
CHƯƠNG 6: BẢO MẬT VÀ QUẢN TRỊ CƠ SỞ DỮ LIỆU .....	32
6.1. Quản lý người dùng và quyền truy cập.....	32
6.1.2. Tạo Roles (vai trò) .....	32
6.1.2. Tạo người dùng (Logins và Users) .....	32
6.1.3. Phân quyền (GRANT) và Thu hồi quyền (REVOKE) .....	33
6.2. Bảo mật cơ sở dữ liệu .....	33
6.3. Quản lý sao lưu và khôi phục .....	34
6.4. Quản lý hiệu suất cơ sở dữ liệu.....	35
6.4.1. Tối ưu hóa bằng chỉ mục .....	35
6.4.2. Tối ưu hóa truy vấn.....	35
CHƯƠNG 7: KẾT LUẬN VÀ KHUYẾN NGHỊ .....	37
7.1. Tóm tắt kết quả .....	37
7.2. Khuyến nghị.....	38

## MỤC LỤC HÌNH ẢNH

4.1. Sơ đồ luồng dữ liệu .....	22
--------------------------------	----

## MỤC LỤC BẢNG

3.2. Bảng BanAn.....	15
3.3. Bảng ThucDon.....	15
3.4. Bảng DatBan.....	16
3.5. Bảng HoaDon.....	16
3.6. Bảng ChiTietHoaDon .....	17

## **BẢNG CÁC TỪ VIẾT TẮT**

<b>STT</b>	<b>TỪ VIẾT TẮT</b>	<b>VIẾT ĐẦY ĐỦ</b>
<b>1</b>	<b>CSDL</b>	<b>Cơ sở dữ liệu</b>

# CHƯƠNG 1. GIỚI THIỆU

## 1.1. Tóm tắt về cơ sở dữ liệu

### 1.1.1. Giới thiệu

Cơ sở dữ liệu là một tập hợp các dữ liệu có tổ chức, liên quan mật thiết với nhau, thường được lưu trữ điện tử trên hệ thống máy tính. CSDL giúp lưu trữ, tổ chức và quản lý lượng lớn dữ liệu (chữ, số, hình ảnh, v.v.) một cách có hệ thống, đảm bảo dễ dàng truy cập, tìm kiếm, cập nhật và chia sẻ.

Hệ quản trị cơ sở dữ liệu là một hệ thống phần mềm được sử dụng để quản lý, điều khiển và tương tác với CSDL, cho phép người dùng tạo lập, truy xuất (đọc), cập nhật (sửa đổi) và xóa dữ liệu trong CSDL một cách hiệu quả và an toàn.

### 1.1.2. Vai trò trong việc giải quyết vấn đề của đề tài

Với đề tài là quản lý nhà hàng, cơ sở dữ liệu và hệ quản trị cơ sở dữ liệu sẽ có các vai trò sau:

- Nơi lưu trữ có tổ chức toàn bộ thông tin cơ bản và giao dịch.
- Quản lý và thực thi các quy trình.
- Tự động hóa các quy trình quan trọng như tính tiền tự động, ghi nhận gọi món và hoàn tất thanh toán,...
- Công cụ quản trị.

## 1.2. Công cụ và công nghệ sử dụng

### 1.2.1. SQL Server

SQL Server là Hệ quản trị cơ sở dữ liệu quan hệ được phát triển bởi Microsoft, là phần mềm chịu trách nhiệm lưu trữ, quản lý, và xử lý toàn bộ dữ liệu. Cụ thể SQL Server thực hiện các thao tác tạo, đọc, cập nhật, và xóa dữ liệu (CRUD), đảm bảo tính bảo mật, toàn vẹn và hiệu suất cao cho cơ sở dữ liệu.



### 1.2.2. SQL Server Management Studio

SQL Server Management Studio là một môi trường phát triển tích hợp chính thức của Microsoft, dùng để quản lý, kết nối và tương tác với SQL Server Engine, hay có thể gọi là giao diện người dùng.

Chức năng:

- Viết, chỉnh sửa, và thực thi các câu lệnh SQL (T-SQL).
- Quản lý các đối tượng cơ sở dữ liệu (tạo bảng, thủ tục, hàm).
- Theo dõi, cấu hình, và tối ưu hóa hiệu suất của SQL Server.

### 1.2.3. SQL Server Configuration Manager

SQL Server Configuration Manager là một công cụ riêng được cài đặt cùng với SQL Server, dùng để quản lý các dịch vụ và cấu hình kết nối.

Chức năng:

- Quản lý dịch vụ: khởi động, dừng, hoặc tạm dừng các dịch vụ liên quan đến SQL Server.
- Cấu hình giao thức mạng: bật/tắt hoặc cấu hình các giao thức kết nối mạng cho phép các ứng dụng khách kết nối từ xa.
- Quản lý bí danh: tạo các tên gọi tắt cho các kết nối máy chủ phức tạp.

## 1.3. Bảng phân công

STT	Tên công việc	Người thực hiện	Ghi chú
1	Phân tích yêu cầu hệ thống	Nguyễn Văn Thành	50%
2	Phân tích yêu cầu hệ thống	Đoàn Đức Phúc	50%
3	Thiết kế cơ sở dữ liệu	Đoàn Đức Phúc	100%
4	Cài đặt cơ sở dữ liệu	Nguyễn Văn Thành	100%
5	Làm báo cáo	Nguyễn Văn Thành	25%

<b>6</b>	Làm báo cáo	Đoàn Đức Phúc	75%
----------	-------------	---------------	-----

## CHƯƠNG 2: PHÂN TÍCH YÊU CẦU HỆ THỐNG

### 2.1. Phân tích và yêu cầu cơ sở dữ liệu

Dựa trên khảo sát thực tế hoạt động của nhà hàng và yêu cầu quản lý, hệ thống cơ sở dữ liệu cần đáp ứng các yêu cầu lưu trữ và xử lý nghiệp vụ sau:

#### **Yêu cầu về lưu trữ thông tin:**

- Quản lý khách hàng: hệ thống cần lưu trữ thông tin cá nhân của khách hàng thân thiết để phục vụ việc đặt bàn và chăm sóc khách hàng. Thông tin bao gồm: Họ tên, số điện thoại (duy nhất), và địa chỉ.
- Quản lý bàn ăn: cần quản lý danh sách các bàn trong nhà hàng, bao gồm số bàn và trạng thái hiện tại (Trống, Có khách, Đã đặt) để nhân viên dễ dàng sắp xếp chỗ ngồi.
- Quản lý thực đơn: lưu trữ danh sách các món ăn, đồ uống với các thông tin: Tên món (không trùng lặp) và đơn giá niêm yết. Giá tiền phải đảm bảo không âm.
- Quản lý đặt bàn: ghi nhận thông tin khi khách hàng đặt trước bàn, bao gồm: Khách hàng nào đặt, bàn số mấy, thời gian đến và số lượng người.
- Quản lý hóa đơn và gọi món:
  - + Lưu trữ thông tin hóa đơn tổng quát: khách hàng, bàn ăn, thời gian thanh toán và tổng tiền.
  - + Lưu trữ chi tiết gọi món: món ăn nào được gọi, số lượng bao nhiêu và đơn giá tại thời điểm gọi.

#### **Yêu cầu về ràng buộc dữ liệu:**

- Tính toàn vẹn dữ liệu: mỗi hóa đơn phải gắn liền với một khách hàng và một bàn ăn cụ thể.
- Quy tắc tính toán: tổng tiền của hóa đơn phải được tính toán tự động dựa trên tổng thành tiền của các món ăn trong chi tiết hóa đơn (Số lượng x Đơn giá).
- Quy tắc trạng thái: khi bàn đang có khách hoặc đã đặt, trạng thái bàn phải được cập nhật tương ứng. Khi thanh toán xong, bàn phải được trả về trạng thái "Trống".
- Ràng buộc giá trị: số lượng món ăn phải lớn hơn 0, đơn giá không được âm.

### 2.2. Các chức năng chính của hệ thống

Dựa trên thiết kế cơ sở dữ liệu, hệ thống hỗ trợ các nhóm chức năng chính sau đây:

#### **Nhóm chức năng quản lý danh mục:**

- Quản lý khách hàng: thêm mới, cập nhật thông tin khách hàng. Tra cứu khách hàng qua số điện thoại.
- Quản lý thực đơn: cập nhật danh sách món ăn và điều chỉnh giá cả.
- Quản lý bàn ăn: theo dõi trạng thái các bàn (Trống/Có khách) để điều phối khách.

#### **Nhóm chức năng xử lý nghiệp vụ:**

- Đặt bàn: ghi nhận lịch đặt bàn của khách hàng.
- Gọi món:
  - + Thêm món ăn vào hóa đơn cho một bàn cụ thể.
  - + Tự động cập nhật số lượng nếu món ăn đã tồn tại trong hóa đơn.
- Tính tiền tự động:
  - + Hệ thống tự động tính tổng tiền hóa đơn mỗi khi có thay đổi về món ăn (thêm/sửa/xóa món).
- Thanh toán:
  - + Xử lý kết thúc đơn hàng, cập nhật thời gian thanh toán thực tế.
  - + Tự động chuyển trạng thái bàn về "Trống" sau khi thanh toán thành công để sẵn sàng đón khách mới.

#### **Nhóm chức năng báo cáo, thống kê:**

- Xem chi tiết hóa đơn: hiển thị đầy đủ thông tin tên món, số lượng, đơn giá và thành tiền cho khách hàng kiểm tra trước khi thanh toán.
- Thống kê doanh thu theo ngày: tính tổng số tiền thu được từ các hóa đơn đã thanh toán trong một ngày cụ thể.
- Báo cáo món ăn bán chạy: thống kê Top các món ăn được gọi nhiều nhất và doanh thu từ món đó để hỗ trợ quyết định nhập nguyên liệu hoặc điều chỉnh thực đơn.

## CHƯƠNG 3: THIẾT KẾ CƠ SỞ DỮ LIỆU

### 3.1. Thiết kế mô hình thực thể quan hệ (ER)

#### 3.1.1. Thực thể và thuộc tính

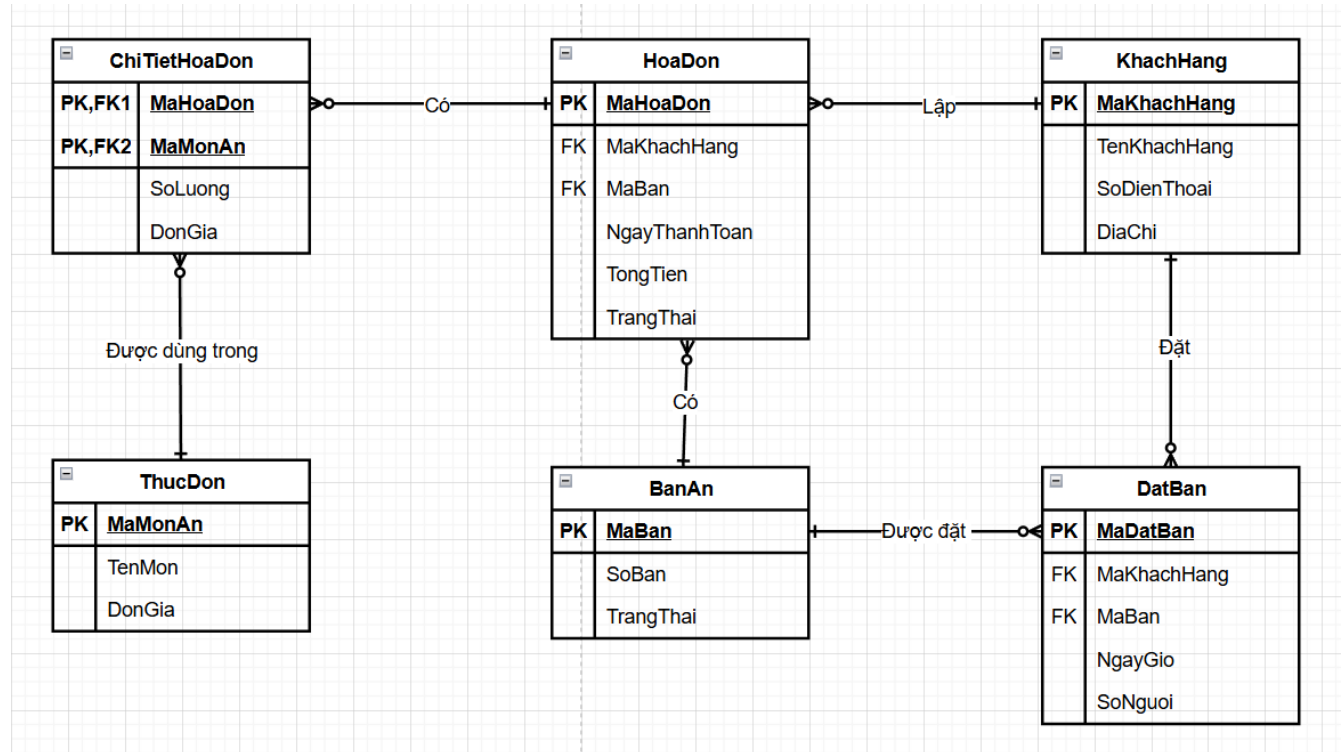
Thực Thể	Thuộc Tính
Khách hàng	Mã khách hàng
	Tên khách hàng
	Số điện thoại
	Địa chỉ
Bàn ăn	Mã bàn
	Số bàn
	Trạng thái
Thực đơn	Mã món ăn
	Tên món
	Đơn giá
Đặt bàn	Mã đặt bàn
	Mã khách hàng
	Mã bàn
	Ngày giờ
	Số người
Hóa đơn	Mã hóa đơn
	Mã khách hàng
	Mã bàn
	Ngày thanh toán
	Tổng tiền

	Trạng thái
<b>Chi tiết hóa đơn</b>	Mã hóa đơn
	Mã món ăn
	Số lượng
	Đơn giá

### 3.1.2. Mỗi quan hệ

Mối Quan Hệ	Thực Thể 1	Liên kết	Thực Thể 2	Kiểu Quan Hệ	Khóa Ngoại
Khách hàng đặt bàn	<b>KhachHang</b>	Đặt	<b>DatBan</b>	1-nhiều (1:N)	DatBan.MaKhachHang
Bàn được đặt	<b>BanAn</b>	Được đặt	<b>DatBan</b>	1-nhiều (1:N)	DatBan.MaBan
Khách hàng lập hóa đơn	<b>KhachHang</b>	Lập	<b>HoaDon</b>	1-nhiều (1:N)	HoaDon.MaKhachHang
Hóa đơn thuộc về bàn	<b>BanAn</b>	Có	<b>HoaDon</b>	1-nhiều (1:N)	HoaDon.MaBan
Chi tiết hóa đơn	<b>HoaDon</b>	Có	<b>ChiTietHoaDon</b>	1-nhiều (1:N)	ChiTietHoaDon.MaHoaDon
Chi tiết món ăn	<b>ThucDon</b>	Được dùng trong	<b>ChiTietHoaDon</b>	1-nhiều (1:N)	ChiTietHoaDon.MaMonAn

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



### 3.3. Cấu trúc các bảng dữ liệu

3.2. Bảng BanAn

STT	Tên trường	Kiểu dữ liệu	Độ rộng	Ràng buộc	Giải thích
1	MaBan	INT		PK, IDENTITY(1,1)	Mã bàn, tự động tăng, khóa chính.
2	SoBan	INT		NOT NULL	Số thứ tự của bàn.
3	TrangThai	NVARCHAR	50	DEFAULT N'Trống'	Trạng thái của bàn (ví dụ: Trống, Có khách, Đã đặt)

3.3. Bảng ThucDon

STT	Tên trường	Kiểu dữ liệu	Độ rộng	Ràng buộc	Giải thích
1	MaMonAn	INT		PK, IDENTITY(1,1)	Mã món ăn, tự động tăng, khóa chính.
2	TenMon	NVARCHAR	100	NOT NULL, UNIQUE	Tên của món ăn, không được trùng.

3	DonGia	DECIMAL	10, 2	NOT NULL, CHECK (DonGia >= 0)	Đơn giá của món ăn, phải lớn hơn hoặc bằng 0.
---	--------	---------	-------	-------------------------------------	---

3.4. Bảng DatBan

STT	Tên trường	Kiểu dữ liệu	Độ rộng	Ràng buộc	Giải thích
1	MaDatBan	INT		PK, IDENTITY(1,1)	Mã đặt bàn, tự động tăng, khóa chính.
2	MaKhachHang	INT		FK (KhachHang)	Khóa ngoại liên kết tới MaKhachHang của bảng KhachHang.
3	MaBan	INT		FK (BanAn)	Khóa ngoại liên kết tới MaBan của bảng BanAn.
4	NgayGio	DATETIME		NOT NULL	Ngày và giờ khách hàng đặt bàn.
5	SoNguoi	INT		DEFAULT 1	Số lượng người, mặc định là 1.

3.5. Bảng HoaDon

STT	Tên trường	Kiểu dữ liệu	Độ rộng	Ràng buộc	Giải thích
1	MaHoaDon	INT		PK, IDENTITY(1,1)	Mã hóa đơn, tự động tăng, khóa chính.
2	MaKhachHang	INT		FK (KhachHang)	Khóa ngoại liên kết tới MaKhachHang của bảng KhachHang.
3	MaBan	INT		FK (BanAn)	Khóa ngoại liên kết tới MaBan của bảng BanAn.
4	NgayThanhToan	DATETIME		DEFAULT GETDATE()	Ngày giờ thanh toán, mặc định là thời gian hiện tại.



5	TongTien	DECIMAL	18, 2	DEFAULT 0	Tổng tiền của hóa đơn, mặc định là 0.
6	TrangThai	NVARCHAR	50	DEFAULT N'Chưa thanh toán'	Trạng thái hóa đơn (ví dụ: Chưa thanh toán, Đã thanh toán)

*3.6. Bảng ChiTietHoaDon*

STT	Tên trường	Kiểu dữ liệu	Độ rộng	Ràng buộc	Giải thích
1	MaHoaDon	INT		PK, FK (HoaDon)	Mã hóa đơn, là khóa ngoại và một phần của khóa chính.
2	MaMonAn	INT		PK, FK (ThucDon)	Mã món ăn, là khóa ngoại và một phần của khóa chính.
3	SoLuong	INT		NOT NULL, CHECK (SoLuong > 0)	Số lượng món ăn, phải lớn hơn 0.
4	DonGia	DECIMAL	10, 2	NOT NULL	Đơn giá của món ăn tại thời điểm lập hóa đơn.

## CHƯƠNG 4: CÀI ĐẶT CƠ SỞ DỮ LIỆU

### 4.1. Tạo bảng

```
USE master;
GO

IF DB_ID(N'QL_NhaHang') IS NOT NULL
BEGIN
    ALTER DATABASE QL_NhaHang SET SINGLE_USER WITH ROLLBACK
    IMMEDIATE;
    DROP DATABASE QL_NhaHang;
END
GO
CREATE DATABASE QL_NhaHang;
GO

USE QL_NhaHang;
GO

CREATE TABLE KháchHang (
    MaKhachHang INT PRIMARY KEY IDENTITY(1,1),
    TenKhachHang NVARCHAR(100) NOT NULL,
    SoDienThoai VARCHAR(15) UNIQUE,
    DiaChi NVARCHAR(255)
);

CREATE TABLE BanAn (
    MaBan INT PRIMARY KEY IDENTITY(1,1),
    SoBan INT NOT NULL,
    TrangThai NVARCHAR(50) DEFAULT N'Trống'
);

CREATE TABLE ThucDon (
    MaMonAn INT PRIMARY KEY IDENTITY(1,1),
    TenMon NVARCHAR(100) NOT NULL UNIQUE,
    DonGia DECIMAL(10, 2) NOT NULL CHECK (DonGia >= 0)
);

CREATE TABLE DatBan (
    MaDatBan INT PRIMARY KEY IDENTITY(1,1),
    MaKhachHang INT FOREIGN KEY REFERENCES KháchHang(MaKhachHang),
    MaBan INT FOREIGN KEY REFERENCES BanAn(MaBan),
    NgayGio DATETIME NOT NULL,
    SoNguoi INT DEFAULT 1
);
```

```

CREATE TABLE HoaDon (
    MaHoaDon INT PRIMARY KEY IDENTITY(1,1),
    MaKhachHang INT FOREIGN KEY REFERENCES KhachHang(MaKhachHang),
    MaBan INT FOREIGN KEY REFERENCES BanAn(MaBan),
    NgayThanhToan DATETIME DEFAULT GETDATE(),
    TongTien DECIMAL(18, 2) DEFAULT 0,
    TrangThai NVARCHAR(50) DEFAULT N'Chưa thanh toán'
);
CREATE TABLE ChiTietHoaDon (
    MaHoaDon INT FOREIGN KEY REFERENCES HoaDon(MaHoaDon),
    MaMonAn INT FOREIGN KEY REFERENCES ThucDon(MaMonAn),
    SoLuong INT NOT NULL CHECK (SoLuong > 0),
    DonGia DECIMAL(10, 2) NOT NULL,
    PRIMARY KEY (MaHoaDon, MaMonAn)
);
GO

```

-- 1. Chèn KhachHang

```

INSERT INTO KhachHang (TenKhachHang, SoDienThoai, DiaChi) VALUES
(N'Nguyễn Văn An', '0901234567', N'123 Lê Lợi, Q1, TPHCM'),
(N'Trần Thị Bích', '0912345678', N'456 Hai Bà Trưng, Hà Nội'),
(N'Phạm Văn Cường', '0987654321', N'789 Nguyễn Trãi, Đà Nẵng'),
(N'Lê Thị Dung', '0909090909', N'101 Võ Văn Tần, Q3, TPHCM'),
(N'Hoàng Văn E', '0888888888', N'202 Hùng Vương, Huế');

```

-- 2. Chèn BanAn

```

INSERT INTO BanAn (SoBan, TrangThai) VALUES
(1, N'Trống'),
(2, N'Trống'),
(3, N'Có khách'),
(4, N'Đã đặt'),
(5, N'Trống');

```

-- 3. Chèn ThucDon

```

INSERT INTO ThucDon (TenMon, DonGia) VALUES
(N'Phở Bò', 50000.00),
(N'Cơm Gà Xối Mỡ', 45000.00),
(N'Bún Chả Hà Nội', 60000.00),
(N'Gỏi Cuốn', 30000.00),
(N'Nước Cam Vắt', 25000.00),
(N'Cà Phê Sữa Đá', 20000.00);

```

-- 4. Chèn DatBan

```

INSERT INTO DatBan (MaKhachHang, MaBan, NgayGio, SoNguoi) VALUES
(1, 4, '2025-11-06 19:00:00', 4),
(2, 4, '2025-11-06 5:40:43', 4),
(3, 4, '2025-11-06 18:00:06', 4),
(4, 4, '2025-11-06 12:30:00', 4),
(5, 4, '2025-11-06 11:00:06', 4);
-- Khai báo các biến để lưu MaHoaDon cho từng lần chèn
DECLARE @MaHD_1 INT, @MaHD_2 INT, @MaHD_3 INT, @MaHD_4 INT, @MaHD_5
INT;

-- 5. CHÈN BẢNG HOA DON

-- Hóa đơn 1 (Ban đầu): Khách Trần Thị Bích (MaKH=2), Bàn 3, Chưa
thanh toán
INSERT INTO HoaDon (MaKhachHang, MaBan, NgayThanhToan, TrangThai)
VALUES
(2, 3, GETDATE(), N'Chưa thanh toán');
SET @MaHD_1 = SCOPE_IDENTITY();

-- Hóa đơn 2: Khách Nguyễn Văn An (MaKH=1), Bàn 1, Đã thanh toán
INSERT INTO HoaDon (MaKhachHang, MaBan, NgayThanhToan, TrangThai)
VALUES
(1, 1, '2025-11-05 14:30:00', N'Đã thanh toán');
SET @MaHD_2 = SCOPE_IDENTITY();

-- Hóa đơn 3: Khách Phạm Văn Cường (MaKH=3), Bàn 2, Đã thanh toán
INSERT INTO HoaDon (MaKhachHang, MaBan, NgayThanhToan, TrangThai)
VALUES
(3, 2, '2025-11-05 19:45:00', N'Đã thanh toán');
SET @MaHD_3 = SCOPE_IDENTITY();

-- Hóa đơn 4: Khách Lê Thị Dung (MaKH=4), Bàn 5, Đã thanh toán
INSERT INTO HoaDon (MaKhachHang, MaBan, NgayThanhToan, TrangThai)
VALUES
(4, 5, '2025-11-06 13:15:00', N'Đã thanh toán');
SET @MaHD_4 = SCOPE_IDENTITY();

-- Hóa đơn 5: Khách Hoàng Văn E (MaKH=5), Bàn 1, Đã thanh toán
INSERT INTO HoaDon (MaKhachHang, MaBan, NgayThanhToan, TrangThai)
VALUES
(5, 1, '2025-11-06 20:00:00', N'Đã thanh toán');
SET @MaHD_5 = SCOPE_IDENTITY();

---
```

-- 6. CHÈN BẢNG CHI TIẾT HOA DON (5 khối INSERT, mỗi khối cho 1 Hóa đơn)

-- Chi tiết Hóa đơn 1 (@MaHD\_1): Phở Bò, Nước Cam, Cà Phê Sữa Đá  
INSERT INTO ChiTietHoaDon (MaHoaDon, MaMonAn, SoLuong, DonGia) VALUES  
(@MaHD\_1, 1, 2, 50000.00), -- Phở Bò  
(@MaHD\_1, 5, 1, 25000.00), -- Nước Cam Vắt  
(@MaHD\_1, 6, 1, 20000.00); -- Cà Phê Sữa Đá

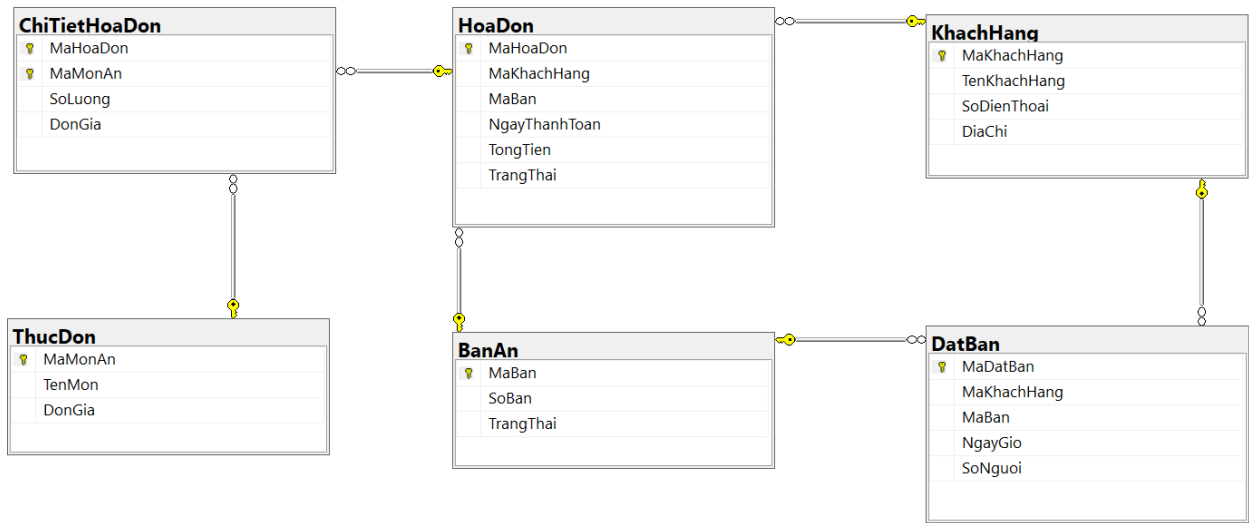
-- Chi tiết Hóa đơn 2 (@MaHD\_2): Phở Bò, Cà Phê Sữa Đá  
INSERT INTO ChiTietHoaDon (MaHoaDon, MaMonAn, SoLuong, DonGia) VALUES  
(@MaHD\_2, 1, 1, 50000.00), -- Phở Bò  
(@MaHD\_2, 6, 1, 20000.00); -- Cà Phê Sữa Đá

-- Chi tiết Hóa đơn 3 (@MaHD\_3): Bún Chả, Gỏi Cuốn  
INSERT INTO ChiTietHoaDon (MaHoaDon, MaMonAn, SoLuong, DonGia) VALUES  
(@MaHD\_3, 3, 2, 60000.00), -- Bún Chả Hà Nội  
(@MaHD\_3, 4, 3, 30000.00); -- Gỏi Cuốn

-- Chi tiết Hóa đơn 4 (@MaHD\_4): Cơm Gà Xối Mỡ, Nước Cam  
INSERT INTO ChiTietHoaDon (MaHoaDon, MaMonAn, SoLuong, DonGia) VALUES  
(@MaHD\_4, 2, 1, 45000.00), -- Cơm Gà Xối Mỡ  
(@MaHD\_4, 5, 2, 25000.00); -- Nước Cam Vắt

-- Chi tiết Hóa đơn 5 (@MaHD\_5): Phở Bò, Bún Chả, Nước Cam  
INSERT INTO ChiTietHoaDon (MaHoaDon, MaMonAn, SoLuong, DonGia) VALUES  
(@MaHD\_5, 1, 1, 50000.00), -- Phở Bò  
(@MaHD\_5, 3, 1, 60000.00), -- Bún Chả Hà Nội  
(@MaHD\_5, 5, 1, 25000.00); -- Nước Cam Vắt

## 4.2. Sơ đồ luồng dữ liệu



### 4.1. Sơ đồ luồng dữ liệu

## CHƯƠNG 5: CÁC ĐỐI TƯỢNG CƠ SỞ DỮ LIỆU NÂNG CAO

### 5.1. Cài đặt các chỉ mục (Indexes)

**Tên chỉ mục:** idx\_KhachHang\_SDT

**Mục đích:** tăng tốc độ truy vấn tìm kiếm khách hàng dựa trên số điện thoại (SoDienThoai), đây là thao tác thường xuyên trong nghiệp vụ bán hàng/đặt bàn.

**Câu lệnh SQL tạo chỉ mục:**

```
IF EXISTS (SELECT * FROM sys.indexes WHERE name =  
'idx_KhachHang_SDT' AND object_id = OBJECT_ID('KhachHang'))  
  
BEGIN  
  
    DROP INDEX idx_KhachHang_SDT ON KhachHang;  
  
END  
  
GO  
  
CREATE INDEX idx_KhachHang_SDT  
  
ON KhachHang (SoDienThoai);  
  
GO
```

### 5.2. Views

**Tên view:** vw\_ChiTietHoaDonDayDu

**Mục đích:** cung cấp khung nhìn tổng hợp và chi tiết của một hóa đơn bằng cách ghép (JOIN) dữ liệu từ 5 bảng (HoaDon, KhachHang, BanAn, ChiTietHoaDon, ThucDon). View giúp truy vấn báo cáo đơn giản và rõ ràng.

**Câu lệnh tạo view:**

```
IF OBJECT_ID('vw_ChiTietHoaDonDayDu', 'V') IS NOT NULL  
BEGIN  
    DROP VIEW vw_ChiTietHoaDonDayDu;  
END  
GO
```

```

CREATE VIEW vw_ChiTietHoaDonDayDu AS
SELECT
    hd.MaHoaDon,
    kh.TenKhachHang,
    ba.SoBan,
    td.TenMon,
    ct.SoLuong,
    ct.DonGia,
    (ct.SoLuong * ct.DonGia) AS ThanhTien
FROM HoaDon hd
JOIN KhachHang kh ON hd.MaKhachHang = kh.MaKhachHang
JOIN BanAn ba ON hd.MaBan = ba.MaBan
JOIN ChiTietHoaDon ct ON hd.MaHoaDon = ct.MaHoaDon
JOIN ThucDon td ON ct.MaMonAn = td.MaMonAn;
GO
SELECT * FROM vw_ChiTietHoaDonDayDu WHERE MaHoaDon = 1;
GO

```

**Kết quả của view:**

	MaHoaDon	TenKhachHang	SoBan	TenMon	SoLuong	DonGia	ThanhTien
1	1	Trần Thị Bích	3	Phở Bò	2	50000.00	100000.00
2	1	Trần Thị Bích	3	Nước Cam Vắt	1	25000.00	25000.00
3	1	Trần Thị Bích	3	Cà Phê Sữa Đá	1	20000.00	20000.00

## 5.3 Thủ tục (Stored Procedures)

### 5.3.1. Thủ tục 1

**Tên thủ tục:** sp\_ThemMonVaoHoaDon

**Mục đích của thủ tục:** xử lý logic nghiệp vụ thêm món ăn vào hóa đơn. Tự động kiểm tra nếu món đã có thì cập nhật số lượng, nếu chưa có thì thêm mới, đồng thời cập nhật tổng tiền trong bảng HoaDon.

**Câu lệnh tạo thủ tục:**

```

IF OBJECT_ID('sp_ThemMonVaoHoaDon', 'P') IS NOT NULL
BEGIN
    DROP PROCEDURE sp_ThemMonVaoHoaDon;
END
GO

```

-- Bước 2: Tạo Procedure mới



```

CREATE PROCEDURE sp_ThemMonVaoHoaDon
    @MaHoaDon INT,
    @MaMonAn INT,
    @SoLuong INT
AS
BEGIN
    -- 1. Lấy đơn giá tại thời điểm đặt
    DECLARE @DonGia DECIMAL(10, 2);
    SELECT @DonGia = DonGia FROM ThucDon WHERE MaMonAn = @MaMonAn;

    -- Kiểm tra xem món ăn có tồn tại trong thực đơn không
    IF @DonGia IS NULL
    BEGIN
        -- Báo lỗi nếu MaMonAn không hợp lệ
        RAISERROR(N'Mã món ăn không tồn tại trong Thực Đơn.', 16, 1);
        RETURN;
    END

    -- 2. Xử lý ChiTietHoaDon
    IF EXISTS (SELECT 1 FROM ChiTietHoaDon WHERE MaHoaDon = @MaHoaDon
    AND MaMonAn = @MaMonAn)
    BEGIN
        -- Món đã có, cập nhật số lượng
        UPDATE ChiTietHoaDon
        SET SoLuong = SoLuong + @SoLuong
        WHERE MaHoaDon = @MaHoaDon AND MaMonAn = @MaMonAn;
    END
    ELSE
    BEGIN
        -- Món chưa có, thêm mới
        INSERT INTO ChiTietHoaDon (MaHoaDon, MaMonAn, SoLuong,
    DonGia)
        VALUES (@MaHoaDon, @MaMonAn, @SoLuong, @DonGia);
    END

    UPDATE HoaDon
    SET TongTien = (
        SELECT ISNULL(SUM(SoLuong * DonGia), 0)
        FROM ChiTietHoaDon
        WHERE MaHoaDon = @MaHoaDon
    )
    WHERE MaHoaDon = @MaHoaDon;
END;
GO

```

### Câu lệnh gọi thủ tục:

```
EXEC sp_ThemMonVaoHoaDon @MaHoaDon = 1, @MaMonAn = 3, @SoLuong = 1;
```

### Kết quả của thủ tục:

	MaHoaDon	TenKhachHang	SoBan	TenMon	SoLuong	DonGia	ThanhTien
1	1	Trần Thị Bích	3	Phở Bò	2	50000.00	100000.00
2	1	Trần Thị Bích	3	Bún Chả Hà Nội	1	60000.00	60000.00
3	1	Trần Thị Bích	3	Nước Cam Vắt	1	25000.00	25000.00
4	1	Trần Thị Bích	3	Cà Phê Sữa Đá	1	20000.00	20000.00

### 5.3.2. Thủ tục 2

**Tên thủ tục:** sp\_ThanhToanHoaDon

**Mục đích của thủ tục:** xử lý toàn bộ quy trình thanh toán. Tính tổng tiền cuối cùng, cập nhật trạng thái hóa đơn thành 'Đã thanh toán', và quan trọng nhất là cập nhật trạng thái bàn ăn về 'Trống' (được bảo vệ bằng Transaction).

### Câu lệnh tạo thủ tục:

```
IF OBJECT_ID('sp_ThanhToanHoaDon', 'P') IS NOT NULL
BEGIN
    DROP PROCEDURE sp_ThanhToanHoaDon;
END
GO
```

```
CREATE PROCEDURE sp_ThanhToanHoaDon
    @MaHoaDon INT
AS
BEGIN

    BEGIN TRANSACTION

    DECLARE @TongTien DECIMAL(18, 2);
    DECLARE @MaBan INT;
    DECLARE @TrangThaiHienTai NVARCHAR(50);

    -- 0. Kiểm tra Hóa đơn có tồn tại không
    SELECT @MaBan = MaBan, @TrangThaiHienTai = TrangThai
    FROM HoaDon
    WHERE MaHoaDon = @MaHoaDon;
```

```

IF @MaBan IS NULL
BEGIN
    RAISERROR(N'Mã hóa đơn không hợp lệ.', 16, 1);
    IF @@TRANCOUNT > 0 ROLLBACK TRANSACTION;
    RETURN;
END

IF @TrangThaiHienTai = N'Đã thanh toán'
BEGIN
    RAISERROR(N'Hóa đơn này đã được thanh toán trước đó.', 16,
1);
    IF @@TRANCOUNT > 0 ROLLBACK TRANSACTION;
    RETURN;
END

-- 1. Tính tổng tiền từ ChiTietHoaDon
SELECT @TongTien = ISNULL(SUM(SoLuong * DonGia), 0)
FROM ChiTietHoaDon
WHERE MaHoaDon = @MaHoaDon;

-- 2. Cập nhật tổng tiền và trạng thái cho HoaDon
UPDATE HoaDon
SET
    TongTien = @TongTien,
    TrangThai = N'Đã thanh toán',
    NgayThanhToan = GETDATE()
WHERE MaHoaDon = @MaHoaDon;

IF @@ERROR <> 0 GOTO ERROR_HANDLER;

-- 3. Cập nhật trạng thái bàn về 'Trống'
UPDATE BanAn
SET TrangThai = N'Trống'
WHERE MaBan = @MaBan;

IF @@ERROR <> 0 GOTO ERROR_HANDLER;

-- 4. Hoàn tất giao dịch (COMMIT)
COMMIT TRANSACTION;

SELECT N'Thanh toán thành công' AS KetQua, * FROM HoaDon WHERE
MaHoaDon = @MaHoaDon;
RETURN;

```

```

ERROR_HANDLER:
    -- Xử lý lỗi (ROLLBACK)
    IF @@TRANCOUNT > 0 ROLLBACK TRANSACTION;
    RAISERROR(N'Đã xảy ra lỗi trong quá trình thanh toán. Giao dịch
đã được hoàn tác.', 16, 1);
    RETURN;
END;
GO

```

#### Câu lệnh gọi thủ tục:

```
EXEC sp_ThanhToanHoaDon @MaHoaDon = 5;
```

#### Kết quả của thủ tục:

```

Msg 50000, Level 16, State 1, Procedure sp_ThanhToanHoaDon, Line 28 [Batch Start Line 236]
Hóa đơn này đã được thanh toán trước đó.

Completion time: 2025-11-08T23:00:15.7456198+07:00

```

### 5.3.3. Thủ tục 3

**Tên thủ tục:** sp\_BaoCaoMonAnBanChay

**Mục đích của thủ tục:** tạo báo cáo động, thống kê và trả về Top N món ăn bán chạy nhất dựa trên số lượng bán.

#### Câu lệnh tạo thủ tục:

```

IF OBJECT_ID('sp_BaoCaoMonAnBanChay', 'P') IS NOT NULL
BEGIN
    DROP PROCEDURE sp_BaoCaoMonAnBanChay;
END
GO

CREATE PROCEDURE sp_BaoCaoMonAnBanChay
    @TopN INT = 5 -- Tham số để chọn Top N món ăn (mặc định Top 5)
AS
BEGIN
    SET NOCOUNT ON;

    -- Truy vấn tổng hợp dữ liệu từ ChiTietHoaDon và ThucDon
    SELECT TOP (@TopN)
        TD.TenMon,

```

```

SUM(CTHD.SoLuong) AS TongSoLuongBan,
SUM(CTHD.SoLuong * CTHD.DonGia) AS TongDoanhThu
FROM
    ChiTietHoaDon CTHD
INNER JOIN
    ThucDon TD ON CTHD.MaMonAn = TD.MaMonAn
GROUP BY
    TD.TenMon
ORDER BY
    TongSoLuongBan DESC, -- Sắp xếp ưu tiên theo số lượng bán
    TongDoanhThu DESC;   -- Sắp xếp thứ hai theo doanh thu
END;
GO

```

**Câu lệnh gọi thủ tục:**

```
EXEC sp_BaoCaoMonAnBanChay @TopN = 3;
```

**Kết quả của thủ tục:**

	TenMon	TongSoLuongBan	TongDoanhThu
1	Bún Chả Hà Nội	7	420000.00
2	Phở Bò	4	200000.00
3	Nước Cam Vắt	4	100000.00

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

**Tên hàm:** fn\_TinhTongDoanhThuTheoNgay

**Mục đích của hàm:** hàm vô hướng (Scalar Function) dùng để tính tổng doanh thu của tất cả các hóa đơn đã thanh toán trong một ngày cụ thể.

**Câu lệnh tạo hàm:**

```

IF OBJECT_ID('fn_TinhTongDoanhThuTheoNgay', 'FN') IS NOT NULL
BEGIN
    DROP FUNCTION fn_TinhTongDoanhThuTheoNgay;
END
GO

```

```

CREATE FUNCTION fn_TinhTongDoanhThuTheoNgay
(
    @Ngay DATE
)
RETURNS DECIMAL(18, 2)
AS
BEGIN
    DECLARE @TongDoanhThu DECIMAL(18, 2);

    SELECT @TongDoanhThu = SUM(TongTien)
    FROM HoaDon
    WHERE TrangThai = N'Đã thanh toán'
    AND CONVERT(DATE, NgayThanhToan) = @Ngay;

    -- Nếu không có doanh thu thì trả về 0
    RETURN ISNULL(@TongDoanhThu, 0);
END;
GO

```

**Câu lệnh gọi hàm:**

```

SELECT dbo.fn_TinhTongDoanhThuTheoNgay(CONVERT(DATE, GETDATE())) AS
DoanhThuHomNay;

```

**Kết quả của hàm:**

	DoanhThuHomNay
1	0.00

## 5.5. Trigger

**Tên trigger:** trg\_CapNhatTongTien

**Mục đích của trigger:** đảm bảo tính nhất quán dữ liệu (Data Consistency). Tự động tính lại và cập nhật trường TongTien trong bảng HoaDon bất cứ khi nào có thay đổi (thêm, sửa, xóa) trong bảng ChiTietHoaDon.

**Câu lệnh tạo trigger:**

```

IF OBJECT_ID('trg_CapNhatTongTien', 'TR') IS NOT NULL
    DROP TRIGGER trg_CapNhatTongTien;
GO

```

```

CREATE TRIGGER trg_CapNhatTongTien
ON ChiTietHoaDon
AFTER INSERT, UPDATE, DELETE
AS
BEGIN
    -- 1. Khai báo bảng tạm để chứa các MaHoaDon bị ảnh hưởng
    DECLARE @AffectedHoaDon TABLE (MaHoaDon INT PRIMARY KEY);

    -- 2. Đưa các MaHoaDon từ các bảng ảo 'inserted' và 'deleted' vào
    --    bảng tạm
    -- (Bao gồm các hóa đơn bị ảnh hưởng bởi INSERT, DELETE, và
    UPDATE)
    INSERT INTO @AffectedHoaDon (MaHoaDon)
    SELECT MaHoaDon FROM inserted
    UNION
    SELECT MaHoaDon FROM deleted;

    UPDATE HD
    SET
        HD.TongTien = ISNULL(
            -- Tính tổng tiền mới từ ChiTietHoaDon hiện tại
            (SELECT SUM(CT.SoLuong * CT.DonGia)
             FROM ChiTietHoaDon CT
             WHERE CT.MaHoaDon = HD.MaHoaDon), 0)
    FROM HoaDon HD
    INNER JOIN @AffectedHoaDon AHD ON HD.MaHoaDon = AHD.MaHoaDon;
END;
GO

```

**Câu lệnh gọi trigger:** không gọi trực tiếp. Trigger được kích hoạt tự động khi INSERT/UPDATE/DELETE trên bảng ChiTietHoaDon (Ví dụ: khi gọi EXEC sp\_ThemMonVaoHoaDon).

## CHƯƠNG 6: BẢO MẬT VÀ QUẢN TRỊ CƠ SỞ DỮ LIỆU

### 6.1. Quản lý người dùng và quyền truy cập

Script SQL của chúng em chủ yếu tập trung vào việc tạo cấu trúc dữ liệu (tables), đối tượng logic (views, procedures) và sao lưu. Script không bao gồm các lệnh tạo người dùng hoặc phân quyền cụ thể. Tuy nhiên, trong một dự án thực tế, việc quản lý truy cập cho CSDL QL\_NhaHang là tối quan trọng. Sau đây là cách thức triển khai (minh họa) cho CSDL này.

#### 6.1.2. Tạo Roles (vai trò)

Chúng ta sẽ tạo các vai trò (roles) để nhóm các quyền hạn theo chức năng công việc, giúp việc quản lý dễ dàng hơn:

- QuanLy: Có quyền hạn cao nhất trong CSDL (tạo, sửa, xóa bảng, xem báo cáo doanh thu, quản lý người dùng).
- NhanVienThuNgan: Chỉ có quyền thực thi các thủ tục (procedures) như sp\_ThemMonVaoHoaDon, sp\_ThanhToanHoaDon và xem các bảng liên quan (ThucDon, BanAn, HoaDon).
- NhanVienKho: Chỉ có quyền xem và cập nhật số lượng/đơn giá trong bảng ThucDon.

Ví dụ:

```
USE QL_NhaHang;
GO
-- Tạo các vai trò
CREATE ROLE QuanLy;
CREATE ROLE NhanVienThuNgan;
GO
```

#### 6.1.2. Tạo người dùng (Logins và Users)

Đầu tiên, tạo tài khoản đăng nhập (Login) ở cấp Server, sau đó tạo người dùng (User) trong CSDL QL\_NhaHang và liên kết với Login đó.

Ví dụ:

```
-- Tạo login ở cấp Server
USE master;
GO
CREATE LOGIN nv_thungan_01 WITH PASSWORD = 'SecurePassword123';
GO
-- Tạo user trong CSDL và liên kết với login
USE QL_NhaHang;
```



```
GO
CREATE USER user_thungan_01 FOR LOGIN nv_thungan_01;
GO
```

### 6.1.3. Phân quyền (GRANT) và Thu hồi quyền (REVOKE)

Sau khi có Role và User, chúng ta gán người dùng vào vai trò và cấp phát quyền cụ thể cho vai trò đó.

**Ví dụ:**

```
-- Thêm user vào role
ALTER ROLE NhanVienThuNgan ADD MEMBER user_thungan_01;

-- Cấp quyền cụ thể cho role NhanVienThuNgan
-- Chỉ cho phép thực thi các thủ tục đã định nghĩa
GRANT EXECUTE ON sp_ThanhToanHoaDon TO NhanVienThuNgan;
GRANT EXECUTE ON sp_ThemMonVaoHoaDon TO NhanVienThuNgan;
GRANT EXECUTE ON sp_BaoCaoMonAnBanChay TO NhanVienThuNgan;

-- Cấp quyền xem view chi tiết hóa đơn
GRANT SELECT ON vw_ChiTietHoaDonDayDu TO NhanVienThuNgan;

-- Ngăn chặn nhân viên xem trực tiếp bảng khách hàng
DENY SELECT ON KhachHang TO NhanVienThuNgan;
```

## 6.2. Bảo mật cơ sở dữ liệu

Trong script, một biện pháp bảo mật quan trọng đã được triển khai là sử dụng View để hạn chế quyền truy cập trực tiếp vào dữ liệu gốc của người dùng. Cụ thể hơn, view `vw_ChiTietHoaDonDayDu` được tạo ra để cung cấp một cái nhìn tổng hợp, chi tiết về các món ăn trong một hóa đơn, bao gồm thông tin khách hàng, số bàn, và thành tiền của từng món.

**Code đã sử dụng:**

```
CREATE VIEW vw_ChiTietHoaDonDayDu AS
SELECT
    hd.MaHoaDon,
    kh.TenKhachHang,
    ba.SoBan,
    td.TenMon,
    ct.SoLuong,
    ct.DonGia,
```

```

(ct.SoLuong * ct.DonGia) AS ThanhTien
FROM HoaDon hd
JOIN KhachHang kh ON hd.MaKhachHang = kh.MaKhachHang
JOIN BanAn ba ON hd.MaBan = ba.MaBan
JOIN ChiTietHoaDon ct ON hd.MaHoaDon = ct.MaHoaDon
JOIN ThucDon td ON ct.MaMonAn = td.MaMonAn;
GO

```

### Ý nghĩa bảo mật:

- Trừu tượng hóa (Abstraction): người dùng cuối (ví dụ: NhanVienThuNgan) không cần biết về cấu trúc phức tạp của 5 bảng (HoaDon, KhachHang, BanAn, ChiTietHoaDon, ThucDon) đang liên kết với nhau.
- Hạn chế truy cập (Access Control): thay vì cấp quyền SELECT trên cả 5 bảng gốc, quản trị viên chỉ cần cấp quyền SELECT trên View vw\_ChiTietHoaDonDayDu. Điều này ngăn người dùng xem hoặc sửa đổi các thông tin nhạy cảm khác không có trong View (ví dụ: DiaChi của khách hàng không được đưa vào View này).

### 6.3. Quản lý sao lưu và hồi phục

Script đã thực hiện một tác vụ quản trị cơ bản và quan trọng nhất là sao lưu toàn bộ (Full Backup) cơ sở dữ liệu để phòng ngừa rủi ro mất mát dữ liệu.

**Hướng dẫn sao lưu:** lệnh BACKUP DATABASE được sử dụng để tạo một bản sao đầy đủ của CSDL QL\_NhaHang và lưu trữ nó vào một tệp tin vật lý trên đĩa.

```

BACKUP DATABASE QL_NhaHang
TO DISK = 'D:\backupfilesql\QL_NhaHang_Full.bak'
WITH NAME = 'QL_NhaHang Full Backup',
    INIT, -- Ghi đè nếu file đã tồn tại
    FORMAT;
GO

```

- TO DISK: chỉ định đường dẫn và tên tệp tin backup.
- WITH INIT: tùy chọn này sẽ ghi đè lên tệp backup cũ nếu nó đã tồn tại tại đường dẫn trên.
- WITH FORMAT: tùy chọn này tạo một bộ phương tiện (media set) mới, thường dùng khi bắt đầu một chuỗi backup mới hoàn toàn.

Mặc dù script chỉ thực hiện Full Backup, một chiến lược sao lưu toàn diện và chuyên nghiệp cho CSDL QL\_NhaHang (vốn có giao dịch thường xuyên) nên bao gồm:

- Full Backup (sao lưu toàn bộ): thực hiện hàng tuần (ví dụ: 12:00 AM mỗi Chủ Nhật). Đây là nền tảng cho tất cả các loại sao lưu khác.
- Differential Backup (sao lưu vi sai): thực hiện hàng ngày (ví dụ: 10:00 PM mỗi ngày). Chỉ sao lưu những gì đã thay đổi kể từ lần Full Backup cuối cùng. Giúp giảm thời gian phục hồi so với chỉ dùng Full Backup.
- Transaction Log Backup (sao lưu nhật ký giao dịch): thực hiện thường xuyên (ví dụ: mỗi 15-30 phút trong giờ hoạt động). Điều này rất quan trọng để có thể phục hồi dữ liệu đến một thời điểm cụ thể (Point-in-Time Recovery) và giảm thiểu mất mát dữ liệu khi có sự cố bất ngờ.

#### 6.4. Quản lý hiệu suất cơ sở dữ liệu

Kịch bản đã áp dụng hai kỹ thuật quan trọng để tối ưu hóa hiệu suất cho CSDL QL\_NhaHang: tạo chỉ mục (Indexing) và sử dụng Stored Procedures để tối ưu hóa truy vấn.

##### 6.4.1. Tối ưu hóa bằng chỉ mục

Một chỉ mục đã được tạo trên cột SoDienThoai của bảng KhachHang:

```
CREATE INDEX idx_KhachHang_SDT  
ON KhachHang (SoDienThoai);  
GO
```

##### Ý nghĩa hiệu suất:

- Cột SoDienThoai (số điện thoại) là một thông tin có tính duy nhất (UNIQUE) và thường xuyên được sử dụng trong mệnh đề WHERE để tìm kiếm khách hàng.
- Không có chỉ mục, khi tìm một số điện thoại, SQL Server sẽ phải quét toàn bộ bảng KhachHang (gọi là Table Scan), rất tốn thời gian và tài nguyên khi bảng có hàng triệu khách hàng.
- Với chỉ mục idx\_KhachHang\_SDT, SQL Server sử dụng cấu trúc B-Tree để tìm kiếm số điện thoại gần như ngay lập tức (gọi là Index Seek), giúp tăng tốc độ truy vấn tìm kiếm khách hàng lên gấp nhiều lần.

##### 6.4.2. Tối ưu hóa truy vấn

Script sử dụng nhiều Stored Procedure như sp\_ThemMonVaoHoaDon, sp\_ThanhToanHoaDon và sp\_BaoCaoMonAnBanChay thay vì viết các câu lệnh SQL rời rạc từ phía ứng dụng.

##### Ý nghĩa hiệu suất:

- Kế hoạch thực thi được lưu trữ (Cached Execution Plan): lần đầu tiên một Stored Procedure được chạy, SQL Server sẽ phân tích, tối ưu hóa và tạo ra một "kế hoạch thực thi" (Execution Plan). Các lần gọi sau sẽ tái sử dụng kế hoạch này, giúp giảm đáng kể thời gian biên dịch và xử lý của máy chủ.
- Giảm lưu lượng mạng: thay vì gửi một loạt các câu lệnh SQL (INSERT, UPDATE, SELECT) phức tạp qua mạng cho nghiệp vụ "Thanh toán", ứng dụng chỉ cần gửi một lệnh EXEC sp\_ThanhToanHoaDon @MaHoaDon = 1 duy nhất.
- Tập trung logic: gói gọn các nghiệp vụ phức tạp (như sp\_ThanhToanHoaDon phải tính tổng tiền, cập nhật HoaDon, cập nhật BanAn trong một TRANSACTION) vào một thủ tục giúp mã nguồn dễ bảo trì và đảm bảo logic được thực thi nhất quán.

## CHƯƠNG 7: KẾT LUẬN VÀ KHUYẾN NGHỊ

### 7.1. Tóm tắt kết quả

Chúng em đã hoàn thành mục tiêu xây dựng và triển khai một cơ sở dữ liệu quan hệ hoàn chỉnh, QL\_NhaHang, trên hệ quản trị SQL Server, nhằm tin học hóa và tối ưu hóa các nghiệp vụ cốt lõi của một nhà hàng. Sau đây là các kết quả chính đạt được.

**Phân tích và Thiết kế CSDL:** đã phân tích các yêu cầu nghiệp vụ và thiết kế thành công mô hình thực thể quan hệ (ERD) và lược đồ quan hệ. Hệ thống xác định rõ các thực thể chính (**KhachHang**, **BanAn**, **ThucDon**, **DatBan**, **HoaDon**) và bảng liên kết (**ChiTietHoaDon**), đảm bảo tính toàn vẹn và chuẩn hóa dữ liệu.

**Triển khai CSDL vật lý:** xây dựng thành công cơ sở dữ liệu vật lý trên SQL Server bằng các câu lệnh T-SQL. Các bảng được tạo với đầy đủ các thuộc tính, kiểu dữ liệu, và các ràng buộc quan trọng (khóa chính, khóa ngoại, UNIQUE, CHECK, DEFAULT) để đảm bảo tính đúng đắn của dữ liệu.

**Xây dựng logic nghiệp vụ và tự động hóa:** triển khai các đối tượng CSDL nâng cao để xử lý các nghiệp vụ phức tạp và tự động hóa quy trình:

- **Stored Procedures:** xây dựng thành công các thủ tục quan trọng như sp\_ThemMonVaoHoaDon (tự động xử lý thêm/cập nhật món ăn) và sp\_ThanhToanHoaDon (tự động tính tổng tiền, cập nhật trạng thái hóa đơn và giải phóng bàn, được bảo vệ bằng TRANSACTION).
- **Trigger:** cài đặt trigger trg\_CapNhatTongTien trên bảng ChiTietHoaDon để tự động tính toán và cập nhật TongTien trong bảng HoaDon mỗi khi có thay đổi, đảm bảo tính nhất quán dữ liệu tuyệt đối.
- **Views và Functions:** tạo vw\_ChiTietHoaDonDayDu để cung cấp cái nhìn tổng hợp, đơn giản hóa truy vấn báo cáo và fn\_TinhTongDoanhThuTheoNgay để hỗ trợ thống kê doanh thu nhanh chóng.

**Quản trị và Tối ưu hóa:** đề xuất và thực thi các giải pháp quản trị cơ bản:

- **Hiệu suất:** tạo chỉ mục (idx\_KhachHang\_SDT) trên cột SoDienThoai để tăng tốc đáng kể tốc độ tìm kiếm khách hàng.
- **Bảo mật:** đề xuất mô hình phân quyền (Roles) và sử dụng View để che giấu độ phức tạp và hạn chế quyền truy cập trực tiếp vào dữ liệu.
- **Sao lưu:** thực hiện sao lưu (Full Backup) CSDL để đảm bảo khả năng phục hồi khi có sự cố.

## 7.2. Khuyến nghị

Để hoàn thiện và nâng cao hệ thống QL\_NhaHang, chúng em đề xuất các hướng phát triển và cải tiến trong tương lai:

**Phát triển ứng dụng Front-end:** xây dựng một ứng dụng (Web, Desktop hoặc Mobile) kết nối với CSDL này. Ứng dụng sẽ cung cấp giao diện đồ họa thân thiện cho các đối tượng người dùng khác nhau (Quản lý, Thu ngân, Phục vụ), cho phép họ tương tác với CSDL thông qua việc thực thi các Stored Procedure đã được tạo (thay vì thao tác trực tiếp trên SQL Server).

**Hoàn thiện hệ thống bảo mật (RBAC):** triển khai đầy đủ hệ thống Phân quyền dựa trên vai trò (Role-Based Access Control) đã đề xuất trong Chương 6. Cần tạo các Login và User cụ thể cho từng vai trò (NhanVienThuNgan, QuanLy) và cấp phát quyền EXECUTE trên các Stored Procedure và SELECT trên các View tương ứng, đồng thời DENY mọi quyền truy cập trực tiếp vào bảng.

**Tự động hóa tác vụ Quản trị:** sử dụng **SQL Server Agent** để tự động hóa các tác vụ quan trọng:

- **Sao lưu:** lên lịch chạy chiến lược sao lưu toàn diện (ví dụ: Full Backup hàng tuần, Differential Backup hàng ngày, và Transaction Log Backup mỗi 30 phút) để giảm thiểu rủi ro mất mát dữ liệu.
- **Bảo trì:** lên lịch chạy các tác vụ bảo trì CSDL như Rebuild Index (xây dựng lại chỉ mục) và Update Statistics (cập nhật thống kê) hàng tuần để duy trì hiệu suất cao.

**Mở rộng các module nghiệp vụ:**

- **Quản lý kho (Inventory):** bổ sung các bảng (ví dụ: NguyenVatLieu, PhieuNhapKho, DinhLuongMonAn) để quản lý số lượng nguyên vật liệu tồn kho, tự động trừ kho khi bán hàng và cảnh báo khi sắp hết hàng.
- **Quản lý nhân viên:** thêm module quản lý thông tin nhân viên, chấm công, và phân ca làm việc. Có thể liên kết HoaDon với MaNhanVien để ghi nhận nhân viên nào đã tạo hóa đơn.
- **Chương trình khuyến mãi:** thêm bảng KhuyenMai và logic nghiệp vụ (trong Stored Procedure hoặc Trigger) để tự động áp dụng giảm giá, chiết khấu cho hóa đơn dựa trên các điều kiện (ví dụ: giờ vàng, tổng tiền, món ăn đặc biệt).