**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN HỌC PHẦN: HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

**ĐỀ TÀI: THIẾT KẾ CƠ SỞ DỮ LIỆU QUẢN LÍ CÁC CHƯƠNG TRÌNH KHUYẾN MÃI**

**Giảng Viên Hướng Dẫn: Th.S Trần Thị Thanh Nhàn**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Mã Sinh Viên | Họ và Tên | Ngày Sinh | Lớp |
| 1 | 1771020325 | Lò Quanng Hưng | 27/04/2005 | CNTT17-04 |
| 2 | 1771020385 | Đỗ Quốc Khánh | 18/12/2004 | CNTT17-04 |
| 3 | 1771020491 | Trần Hải Nam | 12/10/2005 | CNTT17-04 |

### 

**Hà Nội, năm 2025**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN HỌC PHẦN: HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

**ĐỀ TÀI: THIẾT KẾ CƠ SỞ DỮ LIỆU QUẢN LÍ CÁC CHƯƠNG TRÌNH KHUYẾN MÃI**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Mã Sinh Viên | Họ và Tên | Ngày Sinh | Điểm | |
| Bằng Số | Bằng Chữ |
| 1 | 1771020325 | Lò Quang Hưng | 27/04/2005 |  |  |
| 2 | 1771020385 | Đỗ Quốc Khánh | 18/12/2004 |  |  |
| 3 | 1771020491 | Trần Hải Nam | 12/10/2005 |  |  |

### 

### CÁN BỘ CHẤM THI 1 CÁN BỘ CHẤM THI 2

**Hà Nội, năm 2025**

**MỤC LỤC**

**[LỜI NÓI ĐẦU](#_Toc193318056)** [6](#_Toc193318056)

**[CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI THIẾT KẾ CƠ SỞ DỮ LIỆU QUẢN LÝ CÁC CHƯƠNG TRÌNH KHUYẾN MÃI](#_Toc193318057)**

**[1.1. Giới thiệu đề tài](#_Toc193318058)** [7](#_Toc193318058)

**[1.2. Mục tiêu của đề tài](#_Toc193318059)** [7](#_Toc193318059)

**[1.3. Phạm vi nghiên cứu](#_Toc193318060)** [7](#_Toc193318060)

**[1.4. Ý nghĩa của đề tài](#_Toc193318061)** [8](#_Toc193318061)

**[1.5. Cấu trúc của tài liệu](#_Toc193318062)** [8](#_Toc193318062)

**[CHƯƠNG 2: MÔ HÌNH CƠ SỞ DỮ LIỆU QUAN HỆ](#_Toc193318063)** [9](#_Toc193318063)

**[2.1. Xác định các thực thể, thuộc tính và ràng buộc](#_Toc193318064)** [9](#_Toc193318064)

**[2.2. Xây dựng các bảng](#_Toc193318065)** [9](#_Toc193318065)

**[2.3. Mối quan hệ giữa các bảng](#_Toc193318066)** [10](#_Toc193318066)

**[2.4. Sơ đồ ERD (Entity-Relationship Diagram)](#_Toc193318067)** [11](#_Toc193318067)

**[CHƯƠNG 3: TẠO CƠ SỞ DỮ LIỆU](#_Toc193318068)** [12](#_Toc193318068)

**[3.1. Tạo Database](#_Toc193318069)** [12](#_Toc193318069)

**[3.2. Chèn dữ liệu](#_Toc193318070)** [14](#_Toc193318070)

**[3.3. In bảng dữ liệu](#_Toc193318071)** [15](#_Toc193318071)

**[CHƯƠNG 4: XÂY DỰNG CÁC VIEW](#_Toc193318072)** [17](#_Toc193318072)

**[CHƯƠNG 5: XÂY DỰNG CÁC PROCEDURE](#_Toc193318073)** [19](#_Toc193318073)

**[CHƯƠNG 6: XÂY DỰNG CÁC TRIGGER](#_Toc193318074)** [24](#_Toc193318074)

**[CHƯƠNG 7: PHÂN QUYỀN VÀ BẢO VỆ CƠ SỞ DỮ LIỆU](#_Toc193318075)** [29](#_Toc193318075)

**[KẾT LUẬN](#_Toc193318076)** [32](#_Toc193318076)

**[Tài liệu tham khảo](#_Toc193318077)** [34](#_Toc193318077)

**LỜI NÓI ĐẦU**

Trong bối cảnh cạnh tranh ngày càng gay gắt của thị trường, việc thu hút và giữ chân khách hàng trở thành yếu tố sống còn đối với các doanh nghiệp. Một trong những công cụ hiệu quả nhất để thực hiện điều này chính là các chương trình khuyến mãi. Tuy nhiên, để quản lý hiệu quả các chương trình khuyến mãi, đòi hỏi doanh nghiệp phải có một hệ thống cơ sở dữ liệu (CSDL) được thiết kế khoa học, đảm bảo tính chính xác, nhất quán và dễ dàng truy xuất thông tin.

Cuốn tài liệu này trình bày quy trình thiết kế cơ sở dữ liệu quản lý các chương trình khuyến mãi, từ việc phân tích yêu cầu, thiết kế mô hình dữ liệu, đến việc triển khai và quản lý hệ thống. Mục tiêu của tài liệu là cung cấp một cái nhìn tổng quan và chi tiết về cách xây dựng một hệ thống CSDL hiệu quả, giúp doanh nghiệp tối ưu hóa hoạt động khuyến mãi, nâng cao trải nghiệm khách hàng và tăng doanh thu.

Trong quá trình biên soạn, chúng tôi đã cố gắng đưa ra các ví dụ minh họa cụ thể, cùng với những lưu ý quan trọng để tránh các sai sót thường gặp. Hy vọng rằng tài liệu này sẽ là nguồn tham khảo hữu ích cho các nhà quản lý, kỹ thuật viên và những ai quan tâm đến lĩnh vực thiết kế cơ sở dữ liệu.

Mọi ý kiến đóng góp và phản hồi từ quý độc giả sẽ được chúng tôi trân trọng tiếp thu để hoàn thiện hơn nữa các nội dung trong tương lai.

Xin chân thành cảm ơn!

# **CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI THIẾT KẾ CƠ SỞ DỮ LIỆU QUẢN LÝ CÁC CHƯƠNG TRÌNH KHUYẾN MÃI**

## **1.1. Giới thiệu đề tài**

Trong thời đại công nghệ 4.0, việc quản lý hiệu quả các chương trình khuyến mãi trở thành yếu tố then chốt giúp doanh nghiệp duy trì lợi thế cạnh tranh. Các chương trình khuyến mãi không chỉ là công cụ thu hút khách hàng mà còn là phương tiện để tăng doanh thu, quảng bá thương hiệu và xây dựng lòng trung thành của khách hàng. Tuy nhiên, việc quản lý thủ công các chương trình này thường dẫn đến nhiều rủi ro như sai sót dữ liệu, khó khăn trong việc theo dõi hiệu quả và tốn kém thời gian, công sức.

Để giải quyết những thách thức trên, việc thiết kế một hệ thống cơ sở dữ liệu (CSDL) chuyên dụng để quản lý các chương trình khuyến mãi là vô cùng cần thiết. Hệ thống này sẽ giúp doanh nghiệp lưu trữ, truy xuất và phân tích dữ liệu một cách hiệu quả, từ đó đưa ra các quyết định kinh doanh chính xác và kịp thời.

## **1.2. Mục tiêu của đề tài**

Đề tài "Thiết kế cơ sở dữ liệu quản lý các chương trình khuyến mãi" hướng đến các mục tiêu cụ thể sau:

* Phân tích yêu cầu: Tìm hiểu và phân tích các yêu cầu cụ thể của doanh nghiệp trong việc quản lý các chương trình khuyến mãi.
* Thiết kế mô hình dữ liệu: Xây dựng mô hình CSDL phù hợp, đảm bảo tính nhất quán, toàn vẹn và dễ dàng mở rộng.
* Tối ưu hóa quy trình quản lý: Giúp doanh nghiệp quản lý hiệu quả các chương trình khuyến mãi, từ việc lên kế hoạch, triển khai đến đánh giá kết quả.
* Nâng cao trải nghiệm khách hàng: Cung cấp thông tin chính xác và kịp thời về các chương trình khuyến mãi, từ đó tăng sự hài lòng của khách hàng.

## **1.3. Phạm vi nghiên cứu**

Đề tài tập trung vào việc thiết kế CSDL để quản lý các chương trình khuyến mãi trong một doanh nghiệp cụ thể. Các nội dung chính bao gồm:

* Phân tích các yêu cầu chức năng và phi chức năng của hệ thống.
* Thiết kế mô hình ERD (Entity-Relationship Diagram) và chuyển đổi sang mô hình quan hệ.
* Xây dựng các bảng dữ liệu, thiết lập các ràng buộc và quan hệ giữa các thực thể.
* Đề xuất giải pháp triển khai và quản lý hệ thống CSDL.

## **1.4. Ý nghĩa của đề tài**

* Đối với doanh nghiệp: Giúp doanh nghiệp quản lý hiệu quả các chương trình khuyến mãi, tiết kiệm thời gian và chi phí, đồng thời nâng cao khả năng cạnh tranh trên thị trường.
* Đối với khách hàng: Đảm bảo khách hàng nhận được thông tin chính xác và kịp thời về các chương trình khuyến mãi, từ đó tăng trải nghiệm mua sắm.
* Đối với lĩnh vực công nghệ thông tin: Cung cấp một mô hình thiết kế CSDL tham khảo cho các bài toán quản lý khuyến mãi, góp phần phát triển các ứng dụng quản lý trong tương lai.

## **1.5. Cấu trúc của tài liệu**

Tài liệu được chia thành các chương chính như sau:

* Chương 1: Tổng quan về đề tài.
* Chương 2: Mô hình cơ sở dữ liệu quan hệ
* Chương 3: Tạo cơ sở dữ liệu
* Chương 4: Xây dựng các view
* Chương 5: Xây dựng các procedure
* Chương 6: Xây dựng các trigger
* Chương 7: Phân quyền và bảo vệ cơ sở dữ liệu

# **CHƯƠNG 2: MÔ HÌNH CƠ SỞ DỮ LIỆU QUAN HỆ**

## **2.1. Xác định các thực thể, thuộc tính và ràng buộc**

Để quản lý các chương trình khuyến mãi, chúng ta cần xác định các thực thể chính và mối quan hệ giữa chúng. Các thực thể chính bao gồm:

1. **Chương trình khuyến mãi (tblKhuyenMai):** Lưu trữ thông tin về các chương trình khuyến mãi.
2. **Sản phẩm (tblSanPham):** Lưu trữ thông tin về các sản phẩm được áp dụng khuyến mãi.
3. **Loại khuyến mãi (tblLoaiKhuyenMai):** Lưu trữ các loại khuyến mãi như giảm giá, tặng quà, tích điểm, v.v.
4. **Khách hàng (tblKhachHang):** Lưu trữ thông tin khách hàng tham gia chương trình khuyến mãi.
5. **Chi tiết khuyến mãi (tblChiTietKhuyenMai):** Lưu trữ thông tin chi tiết về việc áp dụng khuyến mãi cho từng sản phẩm.

## **2.2. Xây dựng các bảng**

**2.2.1. Bảng tblKhuyenMai:** Lưu trữ thông tin chương trình khuyến mãi

| TT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Ghi chú |
| --- | --- | --- | --- | --- |
| 1 | sMaKhuyenMai | Varchar(20) | PK | Mã chương trình khuyến mãi |
| 2 | sTenKhuyenMai | Nvarchar(100) |  | Tên chương trình khuyến mãi |
| 3 | dNgayBatDau | Date |  | Ngày bắt đầu khuyến mãi |
| 4 | dNgayKetThuc | Date |  | Ngày kết thúc khuyến mãi |
| 5 | fMucGiamGia | Float |  | Mức giảm giá (%) |
| 6 | sMaLoaiKhuyenMai | Varchar(20) | FK | Mã loại khuyến mãi |

**2.2.2. Bảng tblSanPham:** Lưu trữ thông tin sản phẩm

| TT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Ghi chú |
| --- | --- | --- | --- | --- |
| 1 | sMaSanPham | Varchar(20) | PK | Mã sản phẩm |
| 2 | sTenSanPham | Nvarchar(100) |  | Tên sản phẩm |
| 3 | fGiaBan | Float |  | Giá bán |
| 4 | sMoTa | Nvarchar(255) |  | Mô tả sản phẩm |

**2.2.3. Bảng tblLoaiKhuyenMai:** Lưu trữ thông tin loại khuyến mãi

| TT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Ghi chú |
| --- | --- | --- | --- | --- |
| 1 | sMaLoaiKhuyenMai | Varchar(20) | PK | Mã loại khuyến mãi |
| 2 | sTenLoaiKhuyenMai | Nvarchar(50) |  | Tên loại khuyến mãi |

**2.2.4. Bảng tblKhachHang:** Lưu trữ thông tin khách hàng

| TT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Ghi chú |
| --- | --- | --- | --- | --- |
| 1 | sMaKhachHang | Varchar(20) | PK | Mã khách hàng |
| 2 | sTenKhachHang | Nvarchar(100) |  | Tên khách hàng |
| 3 | sDienThoai | Varchar(15) |  | Số điện thoại |
| 4 | sEmail | Varchar(50) |  | Email |

**2.2.5. Bảng tblChiTietKhuyenMai:** Lưu trữ thông tin chi tiết khuyến mãi

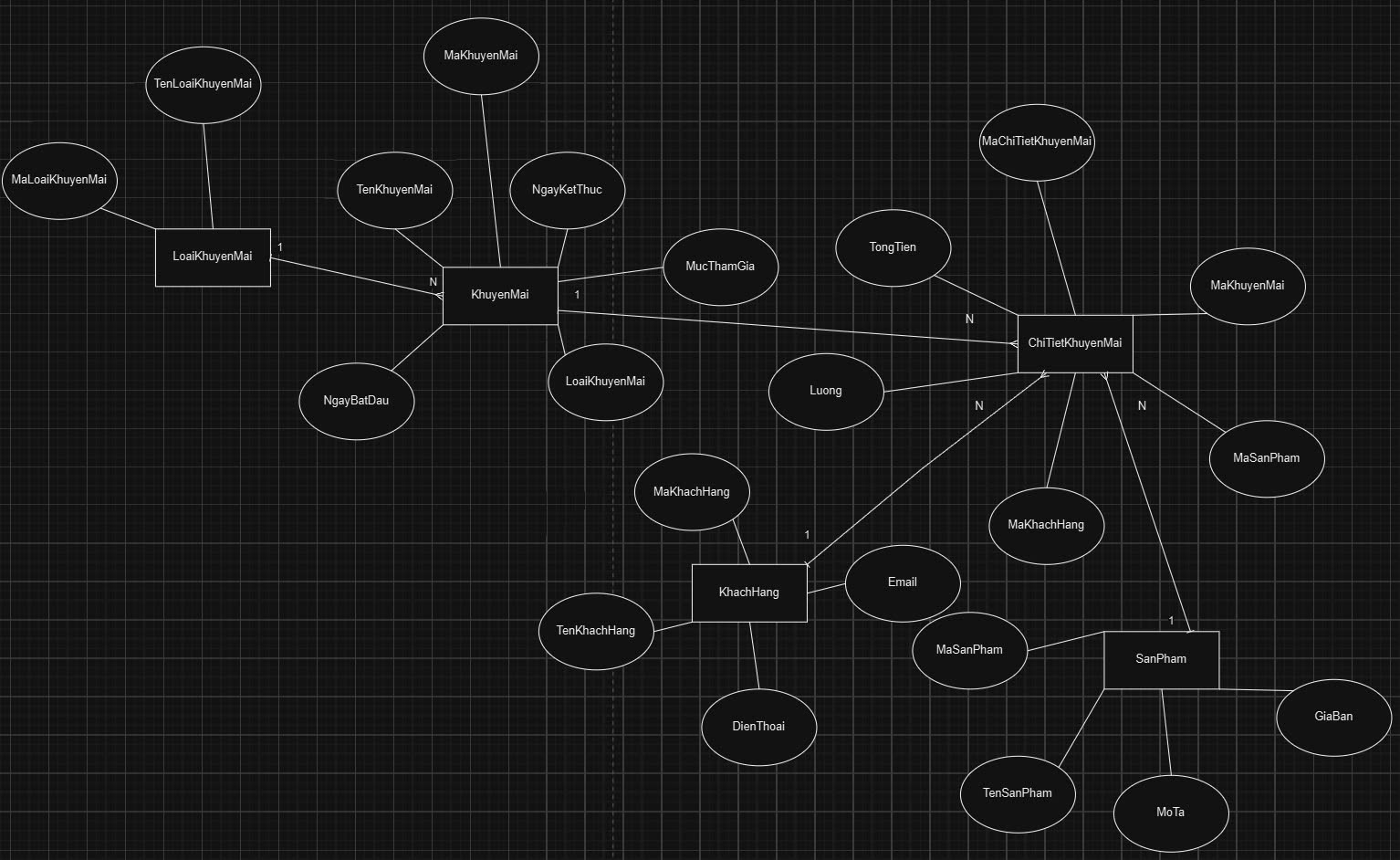
| TT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Ghi chú |
| --- | --- | --- | --- | --- |
| 1 | sMaChiTietKhuyenMai | Varchar(20) | PK | Mã chi tiết khuyến mãi |
| 2 | sMaKhuyenMai | Varchar(20) | FK | Mã chương trình khuyến mãi |
| 3 | sMaSanPham | Varchar(20) | FK | Mã sản phẩm |
| 4 | sMaKhachHang | Varchar(20) | FK | Mã khách hàng |
| 5 | iSoLuong | Int |  | Số lượng sản phẩm |
| 6 | fTongTien | Float |  | Tổng tiền sau khuyến mãi |

## **2.3. Mối quan hệ giữa các bảng**

* tblKhuyenMai và tblLoaiKhuyenMai: Một loại khuyến mãi có thể áp dụng cho nhiều chương trình khuyến mãi (1-N).
* tblKhuyenMai và tblChiTietKhuyenMai: Một chương trình khuyến mãi có thể áp dụng cho nhiều sản phẩm và khách hàng (1-N).
* tblSanPham và tblChiTietKhuyenMai: Một sản phẩm có thể tham gia nhiều chương trình khuyến mãi (1-N).
* tblKhachHang và tblChiTietKhuyenMai: Một khách hàng có thể tham gia nhiều chương trình khuyến mãi (1-N).

## **2.4. Sơ đồ ERD (Entity-Relationship Diagram)**

* Thực thể: tblKhuyenMai, tblSanPham, tblLoaiKhuyenMai, tblKhachHang, tblChiTietKhuyenMai.
* Mối quan hệ:
  + tblKhuyenMai (1) – (N) tblChiTietKhuyenMai
  + tblSanPham (1) – (N) tblChiTietKhuyenMai
  + tblKhachHang (1) – (N) tblChiTietKhuyenMai
  + tblLoaiKhuyenMai (1) – (N) tblKhuyenMai



# **CHƯƠNG 3: TẠO CƠ SỞ DỮ LIỆU**

## **3.1. Tạo Database**

**3.1.1. Database Diagram**

Dưới đây là sơ đồ cơ sở dữ liệu (Database Diagram) cho hệ thống quản lý các chương trình khuyến mãi:

tblLoaiKhuyenMai (1) ──── (N) tblKhuyenMai (1) ──── (N) tblChiTietKhuyenMai (N) ──── (1) tblSanPham

│

└─── (1) tblKhachHang

**3.1.2. Câu lệnh SQL tạo Database và các bảng**

-- Tạo Database

CREATE DATABASE QuanLyKhuyenMai;

GO

-- Sử dụng Database

USE QuanLyKhuyenMai;

GO

-- Tạo bảng tblLoaiKhuyenMai

CREATE TABLE tblLoaiKhuyenMai (

sMaLoaiKhuyenMai VARCHAR(20) PRIMARY KEY,

sTenLoaiKhuyenMai NVARCHAR(50) NOT NULL

);

-- Tạo bảng tblKhuyenMai

CREATE TABLE tblKhuyenMai (

sMaKhuyenMai VARCHAR(20) PRIMARY KEY,

sTenKhuyenMai NVARCHAR(100) NOT NULL,

dNgayBatDau DATE NOT NULL,

dNgayKetThuc DATE NOT NULL,

fMucGiamGia FLOAT NOT NULL,

sMaLoaiKhuyenMai VARCHAR(20) NOT NULL,

FOREIGN KEY (sMaLoaiKhuyenMai) REFERENCES tblLoaiKhuyenMai(sMaLoaiKhuyenMai)

);

-- Tạo bảng tblSanPham

CREATE TABLE tblSanPham (

sMaSanPham VARCHAR(20) PRIMARY KEY,

sTenSanPham NVARCHAR(100) NOT NULL,

fGiaBan FLOAT NOT NULL,

sMoTa NVARCHAR(255)

);

-- Tạo bảng tblKhachHang

CREATE TABLE tblKhachHang (

sMaKhachHang VARCHAR(20) PRIMARY KEY,

sTenKhachHang NVARCHAR(100) NOT NULL,

sDienThoai VARCHAR(15),

sEmail VARCHAR(50)

);

-- Tạo bảng tblChiTietKhuyenMai

CREATE TABLE tblChiTietKhuyenMai (

sMaChiTietKhuyenMai VARCHAR(20) PRIMARY KEY,

sMaKhuyenMai VARCHAR(20) NOT NULL,

sMaSanPham VARCHAR(20) NOT NULL,

sMaKhachHang VARCHAR(20) NOT NULL,

iSoLuong INT NOT NULL,

fTongTien FLOAT NOT NULL,

FOREIGN KEY (sMaKhuyenMai) REFERENCES tblKhuyenMai(sMaKhuyenMai),

FOREIGN KEY (sMaSanPham) REFERENCES tblSanPham(sMaSanPham),

FOREIGN KEY (sMaKhachHang) REFERENCES tblKhachHang(sMaKhachHang)

);

## **3.2. Chèn dữ liệu**

**3.2.1. Câu lệnh SQL chèn dữ liệu**

Dưới đây là các câu lệnh SQL để chèn dữ liệu vào các bảng:

-- Chèn dữ liệu vào bảng tblLoaiKhuyenMai

INSERT INTO tblLoaiKhuyenMai (sMaLoaiKhuyenMai, sTenLoaiKhuyenMai)

VALUES

('LKM001', N'Giảm giá trực tiếp'),

('LKM002', N'Tặng quà'),

('LKM003', N'Tích điểm'),

('LKM004', N'Giảm giá theo hóa đơn'),

('LKM005', N'Khuyến mãi combo');

-- Chèn dữ liệu vào bảng tblKhuyenMai

INSERT INTO tblKhuyenMai (sMaKhuyenMai, sTenKhuyenMai, dNgayBatDau, dNgayKetThuc, fMucGiamGia, sMaLoaiKhuyenMai)

VALUES

('KM001', N'Khuyến mãi Black Friday', '2023-11-20', '2023-11-25', 20, 'LKM001'),

('KM002', N'Tặng quà đặc biệt', '2023-12-01', '2023-12-31', 0, 'LKM002'),

('KM003', N'Tích điểm đổi quà', '2023-10-01', '2023-12-31', 0, 'LKM003'),

('KM004', N'Giảm giá cuối năm', '2023-12-15', '2023-12-31', 15, 'LKM004'),

('KM005', N'Combo mua 2 tặng 1', '2023-11-01', '2023-11-30', 0, 'LKM005');

-- Chèn dữ liệu vào bảng tblSanPham

INSERT INTO tblSanPham (sMaSanPham, sTenSanPham, fGiaBan, sMoTa)

VALUES

('SP001', N'Laptop Dell XPS 13', 25000000, N'Laptop cao cấp'),

('SP002', N'iPhone 15 Pro Max', 30000000, N'Điện thoại thông minh'),

('SP003', N'Samsung Galaxy S23', 20000000, N'Điện thoại Android'),

('SP004', N'Apple Watch Series 8', 10000000, N'Đồng hồ thông minh'),

('SP005', N'AirPods Pro 2', 5000000, N'Tai nghe không dây');

-- Chèn dữ liệu vào bảng tblKhachHang

INSERT INTO tblKhachHang (sMaKhachHang, sTenKhachHang, sDienThoai, sEmail)

VALUES

('KH001', N'Nguyễn Văn A', '0912345678', 'nguyenvana@gmail.com'),

('KH002', N'Trần Thị B', '0987654321', 'tranthib@gmail.com'),

('KH003', N'Lê Văn C', '0901234567', 'levanc@gmail.com'),

('KH004', N'Phạm Thị D', '0978123456', 'phamthid@gmail.com'),

('KH005', N'Hoàng Văn E', '0965432187', 'hoangvane@gmail.com');

-- Chèn dữ liệu vào bảng tblChiTietKhuyenMai

INSERT INTO tblChiTietKhuyenMai (sMaChiTietKhuyenMai, sMaKhuyenMai, sMaSanPham, sMaKhachHang, iSoLuong, fTongTien)

VALUES

('CTKM001', 'KM001', 'SP001', 'KH001', 1, 20000000),

('CTKM002', 'KM002', 'SP002', 'KH002', 2, 60000000),

('CTKM003', 'KM003', 'SP003', 'KH003', 1, 20000000),

('CTKM004', 'KM004', 'SP004', 'KH004', 3, 25500000),

('CTKM005', 'KM005', 'SP005', 'KH005', 2, 10000000);

## **3.3. In bảng dữ liệu**

-- In dữ liệu từ bảng tblLoaiKhuyenMai

SELECT \* FROM tblLoaiKhuyenMai;

-- In dữ liệu từ bảng tblKhuyenMai

SELECT \* FROM tblKhuyenMai;

-- In dữ liệu từ bảng tblSanPham

SELECT \* FROM tblSanPham;

-- In dữ liệu từ bảng tblKhachHang

SELECT \* FROM tblKhachHang;

-- In dữ liệu từ bảng tblChiTietKhuyenMai

SELECT \* FROM tblChiTietKhuyenMai;

| **sMaLoaiKhuyenMai** | **sTenLoaiKhuyenMai** |
| --- | --- |
| LKM001 | Giảm giá trực tiếp |
| LKM002 | Tặng quà |
| LKM003 | Tích điểm |
| LKM004 | Giảm giá theo hóa đơn |
| LKM005 | Khuyến mãi combo |

\* Kết quả

# 

# **CHƯƠNG 4: XÂY DỰNG CÁC VIEW**

**1. Lấy danh sách tất cả các chương trình khuyến mãi**

SELECT \* FROM tblKhuyenMai;

**2. Lấy danh sách sản phẩm có giá bán lớn hơn 15,000,000 VNĐ**

SELECT \* FROM tblSanPham

WHERE fGiaBan > 15000000;

**3. Lấy thông tin khách hàng có email là "[doquockhanh@gmail.com](mailto:doquockhanh@gmail.com)"**

SELECT \* FROM tblKhachHang

WHERE sEmail = 'doquockhanh@gmail.com';

**4. Đếm số lượng chương trình khuyến mãi đang diễn ra**

SELECT CT.sMaChiTietKhuyenMai, SP.sTenSanPham, KH.sTenKhachHang

FROM tblChiTietKhuyenMai CT

JOIN tblSanPham SP ON CT.sMaSanPham = SP.sMaSanPham

JOIN tblKhachHang KH ON CT.sMaKhachHang = KH.sMaKhachHang;

**5. Lấy danh sách chi tiết khuyến mãi kèm tên sản phẩm và tên khách hàng**

SELECT CT.sMaChiTietKhuyenMai, SP.sTenSanPham, KH.sTenKhachHang

FROM tblChiTietKhuyenMai CT

JOIN tblSanPham SP ON CT.sMaSanPham = SP.sMaSanPham

JOIN tblKhachHang KH ON CT.sMaKhachHang = KH.sMaKhachHang;

**6. Tính tổng số tiền khuyến mãi của từng chương trình**

SELECT KM.sMaKhuyenMai, KM.sTenKhuyenMai, SUM(CT.fTongTien) AS TongTienKhuyenMai

FROM tblKhuyenMai KM

JOIN tblChiTietKhuyenMai CT ON KM.sMaKhuyenMai = CT.sMaKhuyenMai

GROUP BY KM.sMaKhuyenMai, KM.sTenKhuyenMai;

**7. Lấy danh sách khách hàng tham gia chương trình khuyến mãi "Black Friday"**

SELECT KH.sTenKhachHang, KH.sDienThoai

FROM tblKhachHang KH

JOIN tblChiTietKhuyenMai CT ON KH.sMaKhachHang = CT.sMaKhachHang

JOIN tblKhuyenMai KM ON CT.sMaKhuyenMai = KM.sMaKhuyenMai

WHERE KM.sTenKhuyenMai = N'Khuyến mãi Black Friday';

**8. Cập nhật mức giảm giá của chương trình khuyến mãi "Black Friday" lên 25%**

UPDATE tblKhuyenMai

SET fMucGiamGia = 25

WHERE sTenKhuyenMai = N'Khuyến mãi Black Friday';

**9. Xóa các chi tiết khuyến mãi có tổng tiền nhỏ hơn 10,000,000 VNĐ**

DELETE FROM tblChiTietKhuyenMai

WHERE fTongTien < 10000000;

**10. Lấy danh sách sản phẩm chưa tham gia bất kỳ chương trình khuyến mãi nào**

SELECT SP.sMaSanPham, SP.sTenSanPham

FROM tblSanPham SP

LEFT JOIN tblChiTietKhuyenMai CT ON SP.sMaSanPham = CT.sMaSanPham

WHERE CT.sMaSanPham IS NULL;

# 

# **CHƯƠNG 5: XÂY DỰNG CÁC PROCEDURE**

**5.1. Procedure 1: Lấy danh sách chương trình khuyến mãi đang diễn ra**

CREATE PROCEDURE sp\_GetActivePromotions

AS

BEGIN

SELECT \* FROM tblKhuyenMai

WHERE dNgayBatDau <= GETDATE() AND dNgayKetThuc >= GETDATE();

END;

**Cách gọi:**

EXEC sp\_GetActivePromotions;

**5.2. Procedure 2: Lấy thông tin chi tiết khuyến mãi theo mã khuyến mãi**

CREATE PROCEDURE sp\_GetPromotionDetails

@MaKhuyenMai VARCHAR(20)

AS

BEGIN

SELECT CT.\*, SP.sTenSanPham, KH.sTenKhachHang

FROM tblChiTietKhuyenMai CT

JOIN tblSanPham SP ON CT.sMaSanPham = SP.sMaSanPham

JOIN tblKhachHang KH ON CT.sMaKhachHang = KH.sMaKhachHang

WHERE CT.sMaKhuyenMai = @MaKhuyenMai;

END;

**Cách gọi:**

EXEC sp\_GetPromotionDetails @MaKhuyenMai = 'KM001';

**5.3. Procedure 3: Thêm mới một chương trình khuyến mãi**

CREATE PROCEDURE sp\_AddPromotion

@MaKhuyenMai VARCHAR(20),

@TenKhuyenMai NVARCHAR(100),

@NgayBatDau DATE,

@NgayKetThuc DATE,

@MucGiamGia FLOAT,

@MaLoaiKhuyenMai VARCHAR(20)

AS

BEGIN

INSERT INTO tblKhuyenMai (sMaKhuyenMai, sTenKhuyenMai, dNgayBatDau, dNgayKetThuc, fMucGiamGia, sMaLoaiKhuyenMai)

VALUES (@MaKhuyenMai, @TenKhuyenMai, @NgayBatDau, @NgayKetThuc, @MucGiamGia, @MaLoaiKhuyenMai);

END;

**Cách gọi:**

EXEC sp\_AddPromotion

@MaKhuyenMai = 'KM006',

@TenKhuyenMai = N'Khuyến mãi Noel',

@NgayBatDau = '2023-12-20',

@NgayKetThuc = '2023-12-25',

@MucGiamGia = 10,

@MaLoaiKhuyenMai = 'LKM001';

**5.4. Procedure 4: Cập nhật mức giảm giá của chương trình khuyến mãi**

CREATE PROCEDURE sp\_UpdatePromotionDiscount

@MaKhuyenMai VARCHAR(20),

@MucGiamGia FLOAT

AS

BEGIN

UPDATE tblKhuyenMai

SET fMucGiamGia = @MucGiamGia

WHERE sMaKhuyenMai = @MaKhuyenMai;

END;

**Cách gọi:**

EXEC sp\_UpdatePromotionDiscount

@MaKhuyenMai = 'KM001',

@MucGiamGia = 25;

**5.5. Procedure 5: Xóa một chương trình khuyến mãi**

CREATE PROCEDURE sp\_DeletePromotion

@MaKhuyenMai VARCHAR(20)

AS

BEGIN

DELETE FROM tblKhuyenMai

WHERE sMaKhuyenMai = @MaKhuyenMai;

END;

**Cách gọi:**

EXEC sp\_DeletePromotion @MaKhuyenMai = 'KM006';

**5.6. Procedure 6: Tính tổng số tiền khuyến mãi của một khách hàng**

CREATE PROCEDURE sp\_GetTotalDiscountByCustomer

@MaKhachHang VARCHAR(20)

AS

BEGIN

SELECT KH.sTenKhachHang, SUM(CT.fTongTien) AS TongTienKhuyenMai

FROM tblChiTietKhuyenMai CT

JOIN tblKhachHang KH ON CT.sMaKhachHang = KH.sMaKhachHang

WHERE KH.sMaKhachHang = @MaKhachHang

GROUP BY KH.sTenKhachHang;

END;

**Cách gọi:**

EXEC sp\_GetTotalDiscountByCustomer @MaKhachHang = 'KH001';

**5.7. Procedure 7: Lấy danh sách sản phẩm tham gia khuyến mãi**

CREATE PROCEDURE sp\_GetProductsInPromotion

@MaKhuyenMai VARCHAR(20)

AS

BEGIN

SELECT SP.sMaSanPham, SP.sTenSanPham

FROM tblSanPham SP

JOIN tblChiTietKhuyenMai CT ON SP.sMaSanPham = CT.sMaSanPham

WHERE CT.sMaKhuyenMai = @MaKhuyenMai;

END;

**Cách gọi:**

EXEC sp\_GetProductsInPromotion @MaKhuyenMai = 'KM001';

**5.8. Procedure 8: Đếm số lượng khách hàng tham gia khuyến mãi**

CREATE PROCEDURE sp\_CountCustomersInPromotion

@MaKhuyenMai VARCHAR(20)

AS

BEGIN

SELECT COUNT(DISTINCT sMaKhachHang) AS SoLuongKhachHang

FROM tblChiTietKhuyenMai

WHERE sMaKhuyenMai = @MaKhuyenMai;

END;

**Cách gọi:**

EXEC sp\_CountCustomersInPromotion @MaKhuyenMai = 'KM001';

**5.9. Procedure 9: Lấy danh sách khuyến mãi theo loại**

CREATE PROCEDURE sp\_GetPromotionsByType

@MaLoaiKhuyenMai VARCHAR(20)

AS

BEGIN

SELECT \* FROM tblKhuyenMai

WHERE sMaLoaiKhuyenMai = @MaLoaiKhuyenMai;

END;

**Cách gọi:**

EXEC sp\_GetPromotionsByType @MaLoaiKhuyenMai = 'LKM001';

**5.10. Procedure 10: Tính tổng doanh thu từ khuyến mãi**

CREATE PROCEDURE sp\_GetTotalRevenueFromPromotions

AS

BEGIN

SELECT SUM(fTongTien) AS TongDoanhThu

FROM tblChiTietKhuyenMai;

END;

**Cách gọi:**

EXEC sp\_GetTotalRevenueFromPromotions;

# **CHƯƠNG 6: XÂY DỰNG CÁC TRIGGER**

Trigger (trình kích hoạt) là các thủ tục tự động được thực thi khi có sự kiện INSERT, UPDATE hoặc DELETE xảy ra trên một bảng cụ thể. Trong chương này, chúng ta sẽ xây dựng các trigger để đảm bảo tính toàn vẹn dữ liệu và tự động hóa các tác vụ trong cơ sở dữ liệu quản lý các chương trình khuyến mãi.

**6.1. Trigger 1: Kiểm tra ngày kết thúc khuyến mãi phải lớn hơn ngày bắt đầu**

CREATE TRIGGER trg\_CheckPromotionDate

ON tblKhuyenMai

FOR INSERT, UPDATE

AS

BEGIN

IF EXISTS (SELECT 1 FROM inserted WHERE dNgayKetThuc <= dNgayBatDau)

BEGIN

RAISERROR('Ngày kết thúc phải lớn hơn ngày bắt đầu.', 16, 1);

ROLLBACK TRANSACTION;

END

END;

**6.2. Trigger 2: Tự động cập nhật tổng tiền khi thêm chi tiết khuyến mãi**

CREATE TRIGGER trg\_UpdateTotalAmount

ON tblChiTietKhuyenMai

FOR INSERT

AS

BEGIN

UPDATE CT

SET fTongTien = SP.fGiaBan \* CT.iSoLuong \* (1 - KM.fMucGiamGia / 100)

FROM tblChiTietKhuyenMai CT

JOIN tblSanPham SP ON CT.sMaSanPham = SP.sMaSanPham

JOIN tblKhuyenMai KM ON CT.sMaKhuyenMai = KM.sMaKhuyenMai

WHERE CT.sMaChiTietKhuyenMai IN (SELECT sMaChiTietKhuyenMai FROM inserted);

END;

**6.3. Trigger 3: Ngăn chặn xóa chương trình khuyến mãi đang diễn ra**

CREATE TRIGGER trg\_PreventDeleteActivePromotion

ON tblKhuyenMai

FOR DELETE

AS

BEGIN

IF EXISTS (SELECT 1 FROM deleted WHERE dNgayBatDau <= GETDATE() AND dNgayKetThuc >= GETDATE())

BEGIN

RAISERROR('Không thể xóa chương trình khuyến mãi đang diễn ra.', 16, 1);

ROLLBACK TRANSACTION;

END

END;

**6.4. Trigger 4: Kiểm tra số lượng sản phẩm khi thêm chi tiết khuyến mãi**

CREATE TRIGGER trg\_CheckProductQuantity

ON tblChiTietKhuyenMai

FOR INSERT

AS

BEGIN

IF EXISTS (SELECT 1 FROM inserted WHERE iSoLuong <= 0)

BEGIN

RAISERROR('Số lượng sản phẩm phải lớn hơn 0.', 16, 1);

ROLLBACK TRANSACTION;

END

END;

**6.5. Trigger 5: Tự động cập nhật số lượng khách hàng tham gia khuyến mãi**

CREATE TRIGGER trg\_UpdateCustomerCount

ON tblChiTietKhuyenMai

FOR INSERT, DELETE

AS

BEGIN

UPDATE KM

SET iSoLuongKhachHang = (SELECT COUNT(DISTINCT sMaKhachHang) FROM tblChiTietKhuyenMai WHERE sMaKhuyenMai = KM.sMaKhuyenMai)

FROM tblKhuyenMai KM

WHERE KM.sMaKhuyenMai IN (SELECT sMaKhuyenMai FROM inserted UNION SELECT sMaKhuyenMai FROM deleted);

END;

**6.6. Trigger 6: Kiểm tra email khách hàng hợp lệ**

CREATE TRIGGER trg\_ValidateCustomerEmail

ON tblKhachHang

FOR INSERT, UPDATE

AS

BEGIN

IF EXISTS (SELECT 1 FROM inserted WHERE sEmail NOT LIKE '%\_@\_\_%.\_\_%')

BEGIN

RAISERROR('Email không hợp lệ.', 16, 1);

ROLLBACK TRANSACTION;

END

END;

**6.7. Trigger 7: Tự động ghi log khi thêm chương trình khuyến mãi**

CREATE TABLE tblLogKhuyenMai (

sMaLog VARCHAR(20) PRIMARY KEY,

sMaKhuyenMai VARCHAR(20),

sTenKhuyenMai NVARCHAR(100),

dNgayTao DATETIME DEFAULT GETDATE()

);

CREATE TRIGGER trg\_LogPromotionInsert

ON tblKhuyenMai

FOR INSERT

AS

BEGIN

INSERT INTO tblLogKhuyenMai (sMaLog, sMaKhuyenMai, sTenKhuyenMai)

SELECT NEWID(), sMaKhuyenMai, sTenKhuyenMai FROM inserted;

END;

**6.8. Trigger 8: Ngăn chặn cập nhật mã khuyến mãi**

CREATE TRIGGER trg\_PreventUpdatePromotionID

ON tblKhuyenMai

FOR UPDATE

AS

BEGIN

IF UPDATE(sMaKhuyenMai)

BEGIN

RAISERROR('Không thể cập nhật mã khuyến mãi.', 16, 1);

ROLLBACK TRANSACTION;

END

END;

**6.9. Trigger 9: Tự động cập nhật mô tả sản phẩm khi giá thay đổi**

CREATE TRIGGER trg\_UpdateProductDescription

ON tblSanPham

FOR UPDATE

AS

BEGIN

IF UPDATE(fGiaBan)

BEGIN

UPDATE SP

SET sMoTa = N'Sản phẩm có giá mới: ' + CAST(i.fGiaBan AS NVARCHAR)

FROM tblSanPham SP

JOIN inserted i ON SP.sMaSanPham = i.sMaSanPham;

END

END;

**6.10. Trigger 10: Kiểm tra loại khuyến mãi tồn tại khi thêm chương trình khuyến mãi**

CREATE TRIGGER trg\_CheckPromotionType

ON tblKhuyenMai

FOR INSERT, UPDATE

AS

BEGIN

IF EXISTS (SELECT 1 FROM inserted WHERE sMaLoaiKhuyenMai NOT IN (SELECT sMaLoaiKhuyenMai FROM tblLoaiKhuyenMai))

BEGIN

RAISERROR('Loại khuyến mãi không tồn tại.', 16, 1);

ROLLBACK TRANSACTION;

END

END;

# 

# **CHƯƠNG 7: PHÂN QUYỀN VÀ BẢO VỆ CƠ SỞ DỮ LIỆU**

**7.1. Tạo các tài khoản người dùng**

-- Tạo tài khoản Quản lý

CREATE LOGIN QuanLy WITH PASSWORD = 'QuanLy@123';

CREATE USER QuanLy FOR LOGIN QuanLy;

-- Tạo tài khoản Nhân viên

CREATE LOGIN NhanVien WITH PASSWORD = 'NhanVien@123';

CREATE USER NhanVien FOR LOGIN NhanVien;

-- Tạo tài khoản Khách hàng

CREATE LOGIN KhachHang WITH PASSWORD = 'KhachHang@123';

CREATE USER KhachHang FOR LOGIN KhachHang;

**7.2. Phân quyền cho Tài khoản Quản lý**

Tài khoản Quản lý có quyền **FULL CONTROL** trên toàn bộ CSDL, bao gồm:

* **SELECT**: Truy vấn dữ liệu.
* **INSERT**: Thêm dữ liệu.
* **UPDATE**: Cập nhật dữ liệu.
* **DELETE**: Xóa dữ liệu.
* **CREATE**: Tạo bảng, view, procedure, v.v.
* **ALTER**: Sửa đổi cấu trúc bảng.
* **DROP**: Xóa bảng, view, procedure, v.v.

-- Cấp quyền cho Quản lý

GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, ALTER, DROP ON DATABASE::QuanLyKhuyenMai TO QuanLy;

**7.3. Phân quyền cho Tài khoản Nhân viên**

Tài khoản Nhân viên có quyền hạn chế hơn, chỉ được phép:

* **SELECT**: Truy vấn dữ liệu.
* **INSERT**: Thêm dữ liệu vào một số bảng.
* **UPDATE**: Cập nhật dữ liệu trong một số bảng.

-- Cấp quyền SELECT trên tất cả các bảng

GRANT SELECT ON SCHEMA::dbo TO NhanVien;

-- Cấp quyền INSERT và UPDATE trên bảng tblChiTietKhuyenMai

GRANT INSERT, UPDATE ON tblChiTietKhuyenMai TO NhanVien;

-- Cấp quyền INSERT và UPDATE trên bảng tblKhachHang

GRANT INSERT, UPDATE ON tblKhachHang TO NhanVien;

**7.4. Phân quyền cho Tài khoản Khách hàng**

Tài khoản Khách hàng chỉ có quyền **SELECT** trên một số bảng để xem thông tin khuyến mãi và sản phẩm:

-- Cấp quyền SELECT trên bảng tblKhuyenMai

GRANT SELECT ON tblKhuyenMai TO KhachHang;

-- Cấp quyền SELECT trên bảng tblSanPham

GRANT SELECT ON tblSanPham TO KhachHang;

-- Cấp quyền SELECT trên bảng tblLoaiKhuyenMai

GRANT SELECT ON tblLoaiKhuyenMai TO KhachHang;

**7.5. Thu hồi quyền**

Nếu cần thu hồi quyền của một tài khoản, chúng ta có thể sử dụng lệnh REVOKE. Ví dụ:

-- Thu hồi quyền INSERT và UPDATE trên bảng tblChiTietKhuyenMai từ Nhân viên

REVOKE INSERT, UPDATE ON tblChiTietKhuyenMai FROM NhanVien;

-- Thu hồi quyền SELECT trên bảng tblKhuyenMai từ Khách hàng

REVOKE SELECT ON tblKhuyenMai FROM KhachHang;

**7.6. Kiểm tra quyền**

Để kiểm tra quyền của một tài khoản, chúng ta có thể sử dụng các câu lệnh sau:

-- Kiểm tra quyền của tài khoản Quản lý

SELECT \* FROM fn\_my\_permissions(NULL, 'DATABASE') WHERE grantee\_principal\_id = USER\_ID('QuanLy');

-- Kiểm tra quyền của tài khoản Nhân viên

SELECT \* FROM fn\_my\_permissions(NULL, 'DATABASE') WHERE grantee\_principal\_id = USER\_ID('NhanVien');

-- Kiểm tra quyền của tài khoản Khách hàng

SELECT \* FROM fn\_my\_permissions(NULL, 'DATABASE') WHERE grantee\_principal\_id = USER\_ID('KhachHang');

**7.7. Ví dụ về phân quyền chi tiết**

**7.7.1. Phân quyền cho Nhân viên chỉ được cập nhật số lượng sản phẩm trong chi tiết khuyến mãi**

-- Cấp quyền UPDATE trên cột iSoLuong của bảng tblChiTietKhuyenMai

GRANT UPDATE (iSoLuong) ON tblChiTietKhuyenMai TO NhanVien;

**7.7.2. Phân quyền cho Khách hàng chỉ được xem thông tin khuyến mãi đang diễn ra**

-- Tạo view chỉ hiển thị các chương trình khuyến mãi đang diễn ra

CREATE VIEW vw\_ActivePromotions AS

SELECT \* FROM tblKhuyenMai

WHERE dNgayBatDau <= GETDATE() AND dNgayKetThuc >= GETDATE();

-- Cấp quyền SELECT trên view cho Khách hàng

GRANT SELECT ON vw\_ActivePromotions TO KhachHang

# **KẾT LUẬN**

**1. Ưu điểm của hệ thống:**

* Tính phân tán: Hệ thống cho phép quản lý dữ liệu khuyến mãi trên nhiều chi nhánh, địa điểm khác nhau, giúp tăng tính linh hoạt và khả năng mở rộng.
* Hiệu suất cao: Việc phân tán dữ liệu giúp giảm tải cho một server duy nhất, cải thiện tốc độ truy vấn và xử lý dữ liệu.
* Tính nhất quán: Sử dụng các cơ chế đồng bộ hóa dữ liệu giữa các node, đảm bảo dữ liệu luôn nhất quán và chính xác.
* Bảo mật tốt: Hệ thống có thể triển khai các cơ chế bảo mật phân tán, giúp bảo vệ dữ liệu khỏi các truy cập trái phép.
* Dễ dàng mở rộng: Khi có thêm chi nhánh hoặc chương trình khuyến mãi mới, hệ thống có thể dễ dàng mở rộng mà không cần thay đổi nhiều cấu trúc hiện có.

**2. Nhược điểm của hệ thống:**

* Độ phức tạp cao: Việc thiết kế và quản lý một hệ thống phân tán đòi hỏi kiến thức chuyên sâu về cơ sở dữ liệu phân tán, cũng như các công nghệ liên quan.
* Chi phí đầu tư ban đầu lớn: Cần đầu tư vào phần cứng, phần mềm và nhân lực để triển khai và vận hành hệ thống.
* Khó khăn trong quản lý đồng bộ: Việc đồng bộ hóa dữ liệu giữa các node có thể gặp khó khăn, đặc biệt là khi có sự cố mạng hoặc lỗi phần cứng.
* Yêu cầu bảo trì thường xuyên: Hệ thống phân tán cần được theo dõi và bảo trì thường xuyên để đảm bảo hoạt động ổn định.

**3. Hướng phát triển trong tương lai:**

* Tích hợp AI và Machine Learning: Có thể tích hợp các công nghệ AI và Machine Learning để phân tích dữ liệu khuyến mãi, dự đoán xu hướng và đề xuất các chương trình khuyến mãi hiệu quả hơn.
* Nâng cao tính bảo mật: Tiếp tục cải thiện các cơ chế bảo mật, đặc biệt là trong việc bảo vệ dữ liệu nhạy cảm và ngăn chặn các cuộc tấn công mạng.
* Tối ưu hóa hiệu suất: Nghiên cứu và áp dụng các giải pháp tối ưu hóa hiệu suất, như caching, sharding, và load balancing để cải thiện tốc độ xử lý dữ liệu.
* Mở rộng hệ thống: Tiếp tục mở rộng hệ thống để hỗ trợ thêm nhiều chi nhánh và đối tác, đồng thời tích hợp với các hệ thống khác như ERP, CRM để tạo ra một hệ sinh thái quản lý doanh nghiệp toàn diện.
* Phát triển giao diện người dùng thân thiện: Cải thiện giao diện người dùng để dễ dàng sử dụng và quản lý, giúp người dùng không chuyên cũng có thể dễ dàng tương tác với hệ thống.

**4. Kết luận chung:**

Việc thiết kế và triển khai hệ thống cơ sở dữ liệu phân tán quản lý các chương trình khuyến mãi mang lại nhiều lợi ích về mặt hiệu suất, tính linh hoạt và khả năng mở rộng. Tuy nhiên, hệ thống cũng đòi hỏi sự đầu tư lớn về công nghệ và nhân lực. Trong tương lai, việc tích hợp các công nghệ tiên tiến và liên tục cải tiến hệ thống sẽ giúp nâng cao hiệu quả quản lý và tối ưu hóa các chương trình khuyến mãi, đem lại lợi ích tối đa cho doanh nghiệp.

# **Tài liệu tham khảo**

1. Sách và giáo trình

1. Özsu, M. T., & Valduriez, P. (2020).  
   *Principles of Distributed Database Systems (4th Edition).*  
   Springer.
   * Cuốn sách này cung cấp kiến thức nền tảng về hệ thống cơ sở dữ liệu phân tán, bao gồm các nguyên tắc thiết kế, quản lý và tối ưu hóa.
2. Elmasri, R., & Navathe, S. B. (2016).  
   *Fundamentals of Database Systems (7th Edition).*  
   Pearson.
   * Sách giáo trình về cơ sở dữ liệu, bao gồm các khái niệm cơ bản và nâng cao, phù hợp để thiết kế hệ thống quản lý khuyến mãi.
3. Garcia-Molina, H., Ullman, J. D., & Widom, J. (2008).  
   *Database Systems: The Complete Book (2nd Edition).*  
   Pearson.
   * Cung cấp kiến thức toàn diện về cơ sở dữ liệu, bao gồm cả các hệ thống phân tán.
4. Tanenbaum, A. S., & Van Steen, M. (2017).  
   *Distributed Systems: Principles and Paradigms (3rd Edition).*  
   Pearson.
   * Sách tập trung vào các nguyên tắc của hệ thống phân tán, phù hợp để hiểu rõ hơn về cách thức hoạt động của cơ sở dữ liệu phân tán.