**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**CƠ SỞ TẠI THÀNH PHỐ HỒ CHÍ MÍNH**

A logo with a star and a book

Description automatically generated

**BÁO CÁO ĐỒ ÁN**

**ĐỀ TÀI: XÂY DỰNG CƠ SỞ DỮ LIỆU**

**QUẢN LÝ CỬA HÀNG BÁN GIÀY DÉP**

Môn học: Thực tập cơ sở

**Giảng viên: Nguyễn Thị Bích Nguyên**

**Thành viên:**

**Nguyễn Thị Huyền My N21DCCN147**

**Lê Ngọc Luân N21DCCN144**

**Nguyễn Đình Luật N21DCCN145**

**Đặng Xuân Nam N21DCCN148**

**Hồ Chí Minh, 2024**

**MỤC LỤC**

[A. Khảo sát về cửa hàng 4](#_Toc167971361)

[I. Về nhân sự (quản lý và nhân viên của TEAM-9) 4](#_Toc167971362)

[II. Về tài nguyên cửa hàng 4](#_Toc167971363)

[III. Về những sự kiện xoay quanh sản phẩm 4](#_Toc167971364)

[IV. Về trải nghiệm của khách hàng khi đến với TEAM-9 5](#_Toc167971365)

[V. Về giao hàng, thanh toán và hậu phát sinh 5](#_Toc167971366)

[B. Yêu cầu thiết kế và tính năng 6](#_Toc167971367)

[I. Quản lý 6](#_Toc167971368)

[II. Nhân viên 7](#_Toc167971369)

[III. Khách hàng 7](#_Toc167971370)

[C. Xoay quanh về cấu trúc, nguyên lý hoạt động của cơ sở dữ liệu 8](#_Toc167971371)

[I. XÁC ĐỊNH THỰC THỂ CƠ SỞ DỮ LIỆU 8](#_Toc167971372)

[II. MÔ HÌNH QUAN HỆ ĐẠT DẠNG CHUẨN 3 9](#_Toc167971373)

[III. DIAGRAM 10](#_Toc167971374)

[1. Về nhân sự và tài nguyên của cửa hàng 10](#_Toc167971375)

[2. Về khách hàng 10](#_Toc167971376)

[3. Về hoạt động của cửa hàng và các vấn đề phát sinh 11](#_Toc167971377)

[IV. TỪ ĐIỂN DỮ LIỆU QUAN HỆ 12](#_Toc167971378)

[1. Bảng Nhân viên 12](#_Toc167971379)

[2. Bảng Quá trình Quản lý 12](#_Toc167971380)

[3. Bảng Tài khoản Nhân viên 12](#_Toc167971381)

[4. Bảng Hãng 13](#_Toc167971382)

[5. Bảng Loại 13](#_Toc167971383)

[6. Bảng Sản phẩm 13](#_Toc167971384)

[7. Bảng Thay đổi giá 14](#_Toc167971385)

[8. Bảng Khuyến mãi 14](#_Toc167971386)

[9. Bảng Chi tiết Khuyến mãi 14](#_Toc167971387)

[10. Bảng Khách hàng 15](#_Toc167971388)

[11. Bảng Tài khoản Khách hàng 15](#_Toc167971389)

[12. Bảng Giỏ hàng 15](#_Toc167971390)

[13. Bảng Đơn đặt hàng 16](#_Toc167971391)

[14. Bảng Chi tiết Đơn đặt hàng 16](#_Toc167971392)

[15. Bảng Giao hàng 16](#_Toc167971393)

[16. Bảng Hóa đơn 17](#_Toc167971394)

[17. Bảng Đánh giá 17](#_Toc167971395)

[18. Bảng Đổi trả 17](#_Toc167971396)

[19. Bảng Phiếu bảo hành 18](#_Toc167971397)

[20. Bảng Nhận bảo hành 18](#_Toc167971398)

[21. Bảng Trả bảo hành 18](#_Toc167971399)

[V. MỘT SỐ RÀNG BUỘC TOÀN VẸN 19](#_Toc167971400)

[1. Ràng buộc toàn vẹn về miền giá trị 19](#_Toc167971401)

[2. Ràng buộc toàn vẹn liên thuộc tính 20](#_Toc167971402)

[3. Ràng buộc toàn vẹn liên thuộc tính liên quan hệ 20](#_Toc167971403)

[VI. TRIGGER VÀ STORED PROCEDURE 21](#_Toc167971404)

[1. Trigger 21](#_Toc167971405)

[2. Stored Procedure 25](#_Toc167971406)

[3. View 32](#_Toc167971407)

[VII. PHÂN QUYỀN 33](#_Toc167971408)

[1. Quản lý: 33](#_Toc167971409)

[2. Nhân viên: 33](#_Toc167971410)

[3. Khách hàng: 34](#_Toc167971411)

[D. Backup 35](#_Toc167971412)

[I. Backup 35](#_Toc167971413)

[II. Restore 38](#_Toc167971414)

[III. Lập lịch Backup 39](#_Toc167971415)

**CỬA HÀNG GIÀY DÉP TEAM-9**

**Mục đích của cơ sở dữ liệu quản lý cửa hàng bán giày dép:** *quản lý các hoạt động mua bán của cửa hàng. Sau đó thống kê doanh thu và các đơn hàng được bán ra của cửa hàng để đưa ra những chiến lược kinh doanh hợp lý.*

A. Khảo sát về cửa hàng

I. Về nhân sự (quản lý và nhân viên của TEAM-9)

* Cửa hàng sẽ có một đội ngũ nhân viên, bao gồm cả quản lý và các nhân viên khác, để đảm bảo hoạt động kinh doanh diễn ra một cách suôn sẻ và hiệu quả nhất. Trong cửa hàng giày dép, luôn chỉ có một quản lý duy nhất trong một thời điểm, người này cũng chính là một nhân viên của cửa hàng. Quản lý có thể được thay đổi theo thời gian tùy theo nhu cầu của cửa hàng trong suốt quá trình hoạt động.
* Thông tin của nhân viên sẽ bao gồm họ tên, giới tính, ngày sinh, số căn cước công dân (CCCD), số điện thoại và địa chỉ email. Quản lý sẽ có thông tin về thời gian bắt đầu làm việc và nhiệm kỳ quản lý (tính theo tháng). Trong trường hợp quản lý phải nghỉ việc đột ngột hoặc vì bất kỳ lý do gì không thể tiếp tục vị trí quản lý, một nhân viên khác sẽ được chọn để thay thế. Quản lý cũng có trách nhiệm tạo tài khoản cho mỗi nhân viên mới tham gia làm việc tại cửa hàng. Mỗi nhân viên chỉ được phép có một tài khoản duy nhất, bao gồm tên đăng nhập (mã nhân viên sẽ được dùng làm tên đăng nhập) và mật khẩu. Thông tin cá nhân của mỗi nhân viên chỉ có thể được thay đổi bởi quản lý khi cần thiết.

II. Về tài nguyên cửa hàng

* Trong cửa hàng, có một loạt sản phẩm đa dạng. Mỗi sản phẩm sẽ được phân loại theo hãng sản xuất như Adidas, Converse,… và các loại như Sneaker, Sandal,.... Thông tin về sản phẩm bao gồm tên, màu sắc, kích thước, chất liệu (như da, vải polyester,…), xuất xứ, giá bán, thời gian bảo hành và số lượng tồn kho. Các thông tin sản phẩm luôn được mô tả cụ thể để khách hàng yên tâm về chất lượng sản phẩm của **TEAM-9.**

III. Về những sự kiện xoay quanh sản phẩm

* Giá của các sản phẩm có thể biến động tùy thuộc vào nhu cầu của thị trường hoặc chiến lược kinh doanh của cửa hàng. Có một nhân viên được ủy quyền chịu trách nhiệm cập nhật giá cho các sản phẩm và ghi chính xác thời gian bắt đầu áp dụng giá mới, bao gồm cả ngày, tháng và năm, kèm theo giờ và phút.
* Trong quá trình hoạt động, cửa hàng sẽ tổ chức nhiều đợt khuyến mãi khác nhau, mỗi đợt có thời gian diễn ra riêng. Trong mỗi đợt khuyến mãi, nhiều sản phẩm sẽ được giảm giá. Việc tổ chức mỗi đợt khuyến mãi sẽ do một nhân viên chịu trách nhiệm, bao gồm việc nhập phần trăm giảm giá cho từng sản phẩm. Mỗi đợt khuyến mãi sẽ có thời gian bắt đầu và kết thúc cụ thể, cùng với lí do tổ chức (như kỷ niệm, lễ Tết, và các dịp đặc biệt khác).

IV. Về trải nghiệm của khách hàng khi đến với TEAM-9

* *Khách hàng muốn mua hàng trên nền tảng thương mại điện tử* phải bắt đầu bằng việc tạo tài khoản, trừ khi họ đã có sẵn. Quá trình đăng ký tài khoản mới bao gồm việc cung cấp các thông tin như họ tên, ngày sinh, số điện thoại, và email (nếu có). Thông tin đăng nhập sẽ bao gồm tên đăng nhập và mật khẩu. Mỗi khách hàng có thể tạo nhiều tài khoản khác nhau, nhưng một tài khoản chỉ có thể thuộc về duy nhất một khách hàng. Khách hàng cũng có thể thay đổi thông tin cá nhân như họ tên, email, và mật khẩu đăng nhập khi cần thiết.
* *Khách hàng đến cửa hàng mua trực tiếp* sẽ được nhân viên hỗ trợ trong việc tạo tài khoản, tương tự như khi mua hàng trực tuyến. Nhân viên sẽ thu thập thông tin như họ tên và số điện thoại trực tiếp từ khách hàng để tạo tài khoản.
* Sau khi đăng nhập vào tài khoản, khách hàng có thể xem các sản phẩm hiện đang được kinh doanh tại cửa hàng. Mỗi tài khoản được gắn với một giỏ hàng riêng, trong đó khách hàng có thể thêm các sản phẩm yêu thích và xem thông tin ngắn gọn về mỗi sản phẩm cùng với số lượng đã chọn. Từ đó, khách hàng có thể lựa chọn sản phẩm và tiến hành đặt hàng.
* Mỗi đơn đặt hàng có thể chứa một hoặc nhiều sản phẩm khác nhau. Trong mỗi đơn đặt hàng, sẽ được liệt kê tên các sản phẩm được đặt cùng với số lượng, giá tiền, và thông tin chi tiết về người đặt hàng. Trạng thái của đơn hàng sẽ được ghi rõ, có thể là "**đang duyệt**", "**đang chuẩn bị hàng**", "**đang giao hàng**", "**đã hoàn thành**", hoặc "**đã hủy**". Hình thức thanh toán cũng được xác định bởi người mua hàng, có thể là chuyển khoản hoặc thanh toán bằng tiền mặt. Hình thức mua hàng có 3 dạng: online, offline (giao hàng) và offline (không giao hàng).
* Mỗi đơn hàng chỉ thuộc về một tài khoản khách hàng duy nhất. Một tài khoản khách hàng có thể đặt nhiều đơn hàng khác nhau.

Lưu ý: *Khách hàng chỉ có thể hủy đơn hàng khi đơn hàng đó chưa bắt đầu trong quá trình vận chuyển.*

V. Về giao hàng, thanh toán và hậu phát sinh

* Sau khi một đơn hàng được đặt, nó sẽ được chuyển đến một nhân viên để tiến hành duyệt. Nhân viên này sẽ xem xét các chi tiết của đơn hàng, bao gồm sản phẩm được đặt, số lượng, và thông tin của người mua. Sau khi đơn hàng được duyệt, nhân viên sẽ bắt đầu chuẩn bị các sản phẩm trong đơn hàng để tiến hành đóng gói và lập hóa đơn. Trong hóa đơn sẽ bao gồm thông tin chi tiết về người mua và tổng số tiền của các sản phẩm đã mua. Nếu có khuyến mãi, số tiền đã được trừ sẽ được hiển thị rõ ràng trong hóa đơn. Thời gian xuất hóa đơn cũng sẽ được ghi lại để tiện theo dõi. Mỗi sản phẩm sẽ được tính thời gian bảo hành bắt đầu từ thời điểm hóa đơn mua hàng được lập.
* Sau khi xuất hóa đơn thì sẽ có 1 nhân viên khác phụ trách việc giao hàng cho khách. Thông tin giao hàng sẽ bao gồm tên nhân viên giao, thông tin đơn hàng và thông tin giao hàng đã được khách hàng cung cấp từ trước. Trong trường hợp khách hàng đặt hàng cho người khác, họ sẽ cung cấp thông tin về người nhận hàng như tên, số điện thoại và địa chỉ giao hàng của người đó. Nếu không, thông tin giao hàng sẽ mặc định là thông tin của tài khoản khách hàng hiện tại. Thời gian bắt đầu và thời gian hoàn thành việc giao hàng sẽ được ghi nhận lại. (đề phòng có sự cố xảy ra hay cần truy xuất lại thông tin,…)
* Chỉ tài khoản nào đã mua sản phẩm đó thì mới được đánh giá sản phẩm. Mỗi sản phẩm chỉ được một tài khoản đánh giá một lần duy nhất. Phần đánh giá bao gồm việc đánh số sao từ 1 đến 5 và bổ sung một nội dung chi tiết. Điều này giúp người dùng diễn đạt ý kiến của họ về sản phẩm một cách cụ thể và đầy đủ.
* Khách hàng chỉ được quyền đổi trả sản phẩm trong trường hợp sản phẩm đã được giao không vừa size hoặc không đúng mẫu mã đã đặt. Điều kiện tiên quyết để đổi trả là khách hàng cần quay video khi mở gói hàng và sản phẩm được giao không quá 7 ngày kể từ lúc nhận hàng. Khách hàng có thể chọn sử dụng dịch vụ giao hàng để đổi trả hoặc đưa sản phẩm trực tiếp đến cửa hàng. Một nhân viên sẽ được ủy quyền lập phiếu đổi trả, bao gồm thông tin chi tiết về khách hàng, lý do đổi trả, và thời gian thực hiện việc đổi trả. Điều này giúp quy trình đổi trả diễn ra một cách trơn tru và hiệu quả.
* Việc bảo hành sản phẩm chỉ áp dụng trong trường hợp sản phẩm bị lỗi do nhà sản xuất, như việc đế giày bung ra hoặc dây giày bị đứt,…. Không chấp nhận bảo hành đối với các tình huống hư hỏng phát sinh do việc sử dụng không đúng cách hoặc do các yếu tố tự nhiên trong quá trình sử dụng.
* Khi nhận sản phẩm để bảo hành, một nhân viên sẽ đảm nhận trách nhiệm lập phiếu nhận, ghi lại các thông tin về khách hàng, lý do bảo hành, và thời gian nhận sản phẩm. Khi trả sản phẩm sau khi bảo hành, cũng tương tự, sẽ có một nhân viên lập phiếu trả, dựa trên thông tin từ phiếu nhận, mô tả tình trạng sản phẩm sau khi đã được bảo hành, và thời gian trả sản phẩm. Điều này giúp quy trình bảo hành diễn ra một cách dễ dàng và minh bạch.

B. Đối tượng sử dụng và tính năng

I. Quản lý

- Xem thông tin của các sản phẩm trong cửa hàng.

- Có quyền thêm tài khoản khi có nhân viên mới, xóa tài khoản nhân viên khi người đó nghỉ việc.

- Chỉnh sửa thông tin cá nhân của nhân viên.

- Xem các đơn hàng đã hoặc đang được xử lý.

- Xem doanh thu (lợi nhuận) theo từng giai đoạn. (ngày, tháng, quý,…)

II. Nhân viên

- Đăng nhập vào phần mềm.

- Thay đổi mật khẩu của tài khoản.

- Thêm, xóa sản phẩm trong cửa hàng. Chỉnh sửa thông tin của sản phẩm.

- Cập nhật các đợt khuyến mãi của cửa hàng cho từng sản phẩm cụ thể.

- Lập phiếu đặt hàng cho các đơn hàng online. Duyệt các đơn hàng và xuất hóa đơn.

- Lập đơn giao hàng.

- Xem lại lịch sử mua hàng.

- Lập phiếu đổi trả hàng.

- Lập phiếu nhận, trả bảo hành.

- Thống kê lợi nhuận theo từng mốc thời gian cụ thể. (ngày, tháng, năm, quý,…)

III. Khách hàng

- Đăng ký tài khoản.

- Quản lý tài khoản của mình như chỉnh sửa thông tin cá nhân, đổi mật khẩu,…

- Xem các sản phẩm và thông tin sản phẩm trong cửa hàng.

- Đặt mua hàng.

- Quản lý giỏ hàng như thêm, xóa, sửa sản phẩm.

- Theo dõi đơn hàng (chờ duyệt, đang chuẩn bị hàng, đang giao hàng, đã hoàn tất đơn hàng hay đã hủy)

- Đổi trả hàng và bảo hành (có một vài điều kiện nhất định khi đổi trả hàng và bảo hành).

C. Xoay quanh về cấu trúc, nguyên lý hoạt động của cơ sở dữ liệu

I. XÁC ĐỊNH THỰC THỂ CƠ SỞ DỮ LIỆU

1. **QTQuanLy**(MaQL, Ngay\_BD, Ngay\_KT)
2. **NhanVien**(MaNV, Ho, Ten, GioiTinh, NgaySinh, CCCD, SDT, Email, HinhAnh)
3. **Loai**(MaLoai, TenLoai)
4. **Hang**(MaHang, TenHang)
5. **SanPham**(MaSP, TenSP, MauSac, Size, ChatLieu, XuatXu, Gia, HinhAnh, TG\_BaoHanh, SLTon)
6. **PhieuBH**(MaBH, Ngay\_BD, Ngay\_KT)
7. **KhachHang**(MaKH, Ho, Ten, SDT, Email)
8. **DatHang**(MaDDH, TrangThai, TG\_Dat)
9. **GiaoHang**(MaGH, ThoiGian\_BD, ThoiGian\_KT, DiaChiGH)
10. **KhuyenMai**(MaDotKM, LyDo, Ngay\_BD, Ngay\_KT)
11. **DoiTra**(MaDT, LyDo)
12. **HoaDon**(MaHD, TongTien)

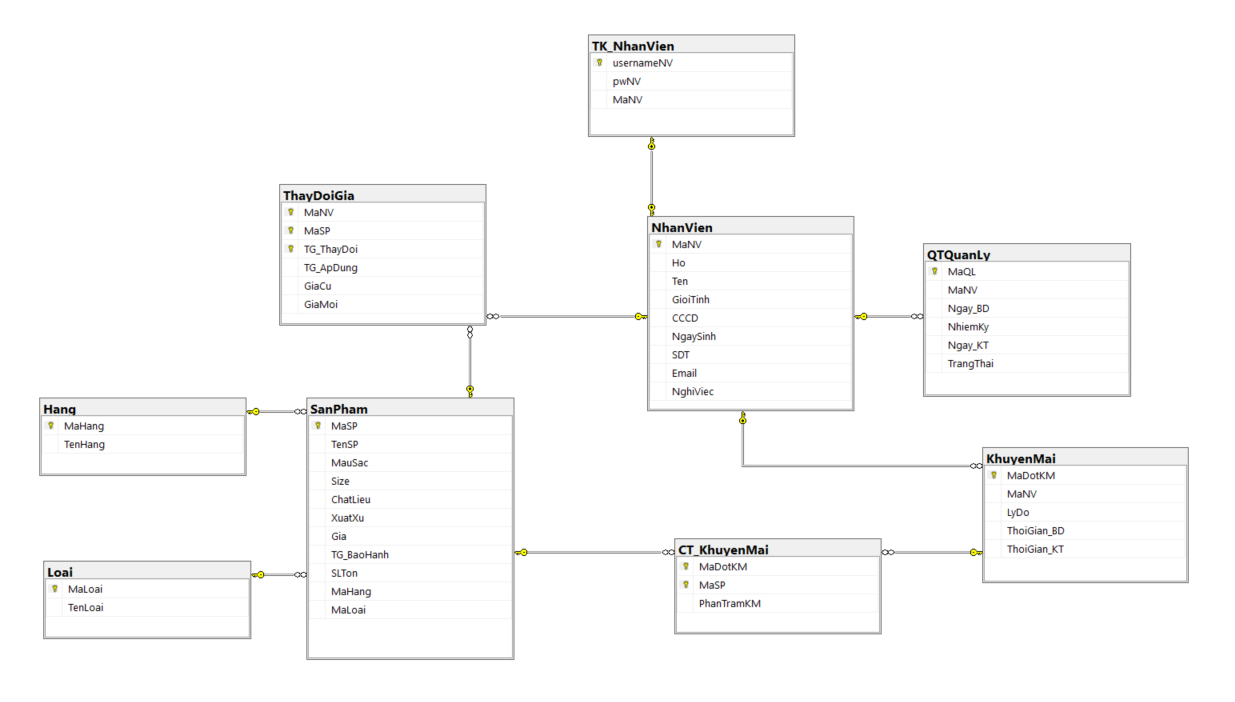
II. MÔ HÌNH QUAN HỆ ĐẠT DẠNG CHUẨN 3

(***Chú thích*** : **ABC** : Tên bảng ; Abc : Khóa chính; Abc : Khóa ngoại )

* **NhanVien**(MaNV, Ho, Ten, GioiTinh, *CCCD*, NgaySinh, *SDT*, *Email*, NghiViec)
* **QTQuanLy**(MaQL, MaNV, Ngay\_BD, NhiemKy, Ngay\_KT, TrangThai)
* **TK\_NhanVien**(usernameNV, pwNV, MaNV)
* **Hang**(MaHang, TenHang)
* **Loai**(MaLoai, TenLoai)
* **SanPham**(MaSP, TenSP, MauSac, Size, ChatLieu, XuatXu, Gia, TG\_BaoHanh, SLTon, MaHang, MaLoai)
* **ThayDoiGia**(MaNV, MaSP, TG\_ThayDoi, TG\_ApDung, GiaCu, GiaMoi)
* **KhuyenMai**(MaDotKM, MaNV, LyDo, ThoiGian\_BD, ThoiGian\_KT)
* **CT\_KhuyenMai**(MaDotKM, MaSP, PhanTramKM)
* **KhachHang**(MaKH, Ho, Ten, *SDT*, *Email*)
* **TK\_KhachHang**(usernameKH, pwKH, MaKH)
* **GioHang**(MaGio, usernameKH, MaSP, SoLuong, TrangThai)
* **DonDH**(MaDDH, TG\_Dat, TrangThai, HinhThuc, ThanhToan, usernameKH , MaNV)
* **CT\_DonDH**(MaCTDDH, MaDDH, MaGio, SoLuong, MaHD)
* **GiaoHang**(MaGH, MaNV, MaDDH, ThoiGian\_BD, ThoiGian\_KT, TenNguoiNhan, SDT, DiaChiGH)
* **HoaDon**(MaHD, TongTien, ThoiGian, MaNV)
* **DanhGia**(MaCTDDH, Sao, NoiDung)
* **DoiTra**(MaDT, MaNV, MaCTDDH, LyDo, TG\_DoiTra)
* **PhieuBH**(MaBH, MaCTDDH, MaNV, Ngay\_BD, Ngay\_KT)
* **NhanBH**(MaNhanBH, MaBH, MaNV, LyDoBH, TG\_BaoHanh)
* **TraBH**(MaTraBH, MaNhanBH, MaNV, TinhTrangSauBH, TG\_Tra)

III. DIAGRAM

1. Về nhân sự và tài nguyên của cửa hàng

****

2. Về khách hàng

**A screenshot of a computer

Description automatically generated**

3. Về hoạt động của cửa hàng và các vấn đề phát sinh

A computer screen shot of a computer flow chart

Description automatically generated

IV. TỪ ĐIỂN DỮ LIỆU QUAN HỆ

1. Bảng Nhân viên

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Type** | **Length** | **Constraint** |
| MaNV | int |  | PK |
| Ho | nvarchar | 100 | Not Null |
| Ten | nvarchar | 50 | Not Null |
| GioiTinh | nvarchar | 3 | Not Null |
| CCCD | varchar | 12 | UK, Not Null |
| NgaySinh | date |  | Not Null |
| SDT | varchar | 10 | UK, Not Null |
| Email | varchar | 50 | UK, Not Null |
| NghiViec | bit |  | Not Null |

Khóa duy nhất: CCCD, SDT và Email

1. Bảng Quá trình Quản lý

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Type** | **Length** | **Constraint** |
| MaQL | int |  | PK |
| MaNV | varchar | 10 | FK. Not Null |
| Ngay\_BD | date |  | Not Null |
| NhiemKy | tinyint |  | Not Null |
| Ngay\_KT | date |  |  |
| TrangThai | bit |  | Not Null |

Khóa duy nhất: MaNV + Ngay\_BD

1. Bảng Tài khoản Nhân viên

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Type** | **Length** | **Constraint** |
| usernameNV | varchar | 10 | PK |
| pwNV | varchar | 50 | Not Null |
| MaNV | varchar | 10 | FK, Not Null |

Khóa duy nhất: MaNV

1. Bảng Hãng

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Type** | **Length** | **Constraint** |
| MaHang | varchar | 10 | PK |
| TenHang | nvarchar | 50 | UK, Not Null |

Khóa duy nhất: TenHang

1. Bảng Loại

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Type** | **Length** | **Constraint** |
| MaLoai | varchar | 10 | PK |
| TenLoai | nvarchar | 50 | UK, Not Null |

Khóa duy nhất: TenLoai

1. Bảng Sản phẩm

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Type** | **Length** | **Constraint** |
| MaSP | Int |  | PK |
| TenSP | nvarchar | 50 | Not Null |
| MauSac | nvarchar | 10 | Not Null |
| Size | tinyint |  | Not Null |
| ChatLieu | nvarchar | 30 | Not Null |
| XuatXu | nvarchar | 30 | Not Null |
| Gia | money |  | Not Null |
| TG\_BaoHanh | tinyint |  | Not Null |
| SLTon | smallint |  | Not Null |
| MaHang | varchar | 10 | FK, Not Null |
| MaLoai | varchar | 10 | FK, Not Null |

Khóa duy nhất: TenSP + MauSac + Size

1. Bảng Thay đổi giá

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Type** | **Length** | **Constraint** |
| MaNV | int |  | PK, FK |
| MaSP | int |  | PK, FK |
| TG\_ThayDoi | datetime |  | PK |
| TG\_ApDung | datetime |  | Not Null |
| GiaCu | money |  | Not Null |
| GiaMoi | money |  | Not Null |

1. Bảng Khuyến mãi

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Type** | **Length** | **Constraint** |
| MaDotKM | int |  | PK |
| MaNV | int |  | FK, Not Null |
| LyDo | nvarchar | 100 | Not Null |
| ThoiGian\_BD | datetime |  | Not Null |
| ThoiGian\_KT | datetime |  | Not Null |

1. Bảng Chi tiết Khuyến mãi

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Type** | **Length** | **Constraint** |
| MaDotKM | int |  | PK, FK |
| MaSP | int |  | PK, FK |
| PhanTramKM | float |  | Not Null |

1. Bảng Khách hàng

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Type** | **Length** | **Constraint** |
| MaKH | int |  | PK |
| Ho | nvarchar | 100 | Not Null |
| Ten | nvarchar | 50 | Not Null |
| SDT | varchar | 10 | UK, Not Null |
| Email | varchar | 50 | UK, Not Null |

Khóa duy nhất: SDT, Email

1. Bảng Tài khoản Khách hàng

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Type** | **Length** | **Constraint** |
| usernameKH | varchar | 50 | PK |
| pwKH | varchar | 100 | Not Null |
| MaKH | int |  | FK, Not Null |

1. Bảng Giỏ hàng

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Type** | **Length** | **Constraint** |
| MaGio | int |  | PK |
| usernameKH | varchar | 50 | FK |
| MaSP | Int |  | FK, Not Null |
| SoLuong | tinyint |  | Not Null |
| TrangThai | bit |  | Not Null |

1. Bảng Đơn đặt hàng

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Type** | **Length** | **Constraint** |
| MaDDH | int |  | PK |
| TG\_Dat | datetime |  | Not Null |
| TrangThai | nvarchar | 20 | Not Null |
| HinhThuc | nvarchar | 20 | Not Null |
| ThanhToan | nvarchar | 20 | Not Null |
| MaNV | int |  | FK |

1. Bảng Chi tiết Đơn đặt hàng

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Type** | **Length** | **Constraint** |
| MaCTDDH | int |  | PK |
| MaGio | int |  | FK, Not Null |
| MaDDH | int |  | FK. Not Null |
| SoLuong | tinyint |  | Not Null |
| MaHD | varchar | 10 | FK |

1. Bảng Giao hàng

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Type** | **Length** | **Constraint** |
| MaGH | int |  | PK |
| MaNV | Int |  | FK |
| MaDDH | int |  | FK |
| ThoiGian\_BD | datetime |  | Not Null |
| ThoiGian\_KT | datetime |  |  |
| TenNguoiNhan | nvarchar | 50 |  |
| SDT | varchar | 10 |  |
| DiaChiGH | nvarchar | 100 | Not Null |

1. Bảng Hóa đơn

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Type** | **Length** | **Constraint** |
| MaHD | varchar | 10 | PK |
| TongTien | money |  |  |
| ThoiGian | datetime |  |  |
| MaNV | int |  | FK |

1. Bảng Đánh giá

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Type** | **Length** | **Constraint** |
| MaCTDDH | int |  | PK, FK |
| Sao | tinyint |  | PK |
| NoiDung | nvarchar | 100 | Not Null |

1. Bảng Đổi trả

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Type** | **Length** | **Constraint** |
| MaDT | int |  | PK |
| MaNV | int |  | FK, Not Null |
| MaCTDDH | int |  | FK, Not Null |
| LyDo | nvarchar | 100 | Not Null |
| TG\_DoiTra | datetime |  | Not Null |

1. Bảng Phiếu bảo hành

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Type** | **Length** | **Constraint** |
| MaBH | Int |  | PK |
| MaCTDDH | int |  | FK, Not Null |
| MaNV | int |  | FK, Not Null |
| Ngay\_BD | date |  | Not Null |
| Ngay\_KT | date |  | Null |

Khóa duy nhất: MaCTDDH

1. Bảng Nhận bảo hành

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Type** | **Length** | **Constraint** |
| MaNhanBH | int |  | PK |
| MaBH | int |  | FK, Not Null |
| MaNV | int |  | FK, Not Null |
| LyDoBH | nvarchar | 200 | Not Null |
| TG\_Nhan | datetime |  | Not Null |

1. Bảng Trả bảo hành

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Type** | **Length** | **Constraint** |
| MaTraBH | int |  | PK |
| MaNhanBH | int |  | FK, Not Null |
| MaNV | int |  | FK, Not Null |
| TinhTrangSauBH | nvarchar | 100 | Not Null |
| TG\_Tra | datetime |  | Not Null |

Khóa duy nhất: MaNhanBH

V. MỘT SỐ RÀNG BUỘC TOÀN VẸN

1. Ràng buộc toàn vẹn về miền giá trị

* Nhân viên chỉ có 2 giới tính Nam và Nữ.

∀ nv ∈ NhanVien: nv.GioiTinh = N’Nam’ OR nv.GioiTinh = N’Nữ’

* Ngày sinh của nhân viên phải nhỏ hơn thời gian hiện tại.

∀ nv ∈ NhanVien: nv.NgaySinh < CONVERT(DATE, GETDATE())

* Size sản phẩm nằm trong khoảng từ 15 đến 45.

∀ sp ∈ SanPham: sp.size BETWEEN 15 AND 45

* Giá sản phẩm phải lớn hơn 0.

∀ sp ∈ SanPham: sp.Gia > 0

* Thời gian bảo hành của mỗi sản phẩm phải lớn hơn hoặc bằng 0.

∀ sp ∈ SanPham: sp.TG\_BaoHanh >= 0

* Số lượng tồn của sản phẩm phải lớn hơn hoặc bằng 0.

∀ sp ∈ SanPham: sp.SLTon >= 0

* Thời gian bắt đầu của đợt khuyến mãi sẽ lớn hơn hoặc bằng thời gian hiện tại.

∀ km ∈ KhuyenMai: km.ThoiGian\_BD >= GETDATE()

* Phần trăm khuyến mãi phải lớn hơn 0 và nhỏ hơn 100.

∀ km ∈ KhuyenMai: km.PhanTramKM > 0 AND km.PhanTramKM < 100

* Số lượng sản phẩm trong giỏ hàng phải lớn hơn hoặc bằng 0.

∀ gh ∈ GioHang: gh.SoLuong >= 0

* Thời gian đặt hàng phải lớn hơn hoặc bằng thời gian hiện tại.

∀ ddh ∈ DonDH: ddh.TG\_Dat >= GETDATE()

* Đơn hàng có 5 trạng thái: đang duyệt, đang chuẩn bị, đang giao hàng, đã giao hàng, đã duyệt.

∀ ddh ∈ DonDH: ddh.TrangThai = N’Đang duyệt’ OR ddh.TrangThai = N’Đang chuẩn bị’ OR ddh.TrangThai = N’Đang giao hàng’ OR ddh.TrangThai = N’Đã hoàn thành’ OR ddh.TrangThai = N’Đã hủy’

* Đơn hàng có 2 hình thức thanh toán: chuyển khoản và tiền mặt.

∀ ddh ∈ DonDH: ddh.ThanhToan = N’Chuyển khoản’ OR N’Tiền mặt’

* Số lượng sản phẩm trong Chi tiết đơn đặt hàng phải lớn hơn 0.

∀ ct ∈ CT\_DonDH: ct.SoLuong > 0

* Thời gian bắt đầu giao hàng phải lớn hơn hoặc bằng thời gian hiện tại.

∀ gh ∈ GiaoHang: gh.ThoiGian\_BD >= GETDATE()

* Thời gian xuất hóa đơn phải lớn hơn hoặc bằng thời gian hiện tại.

∀ hd ∈ HoaDon: hd.ThoiGian >= GETDATE()

* Số sao đánh giá nằm trong khoảng từ 1 đến 5 sao.

∀ dg ∈ DanhGia: dg.Sao BETWEEN 1 AND 5

* Thời gian đổi trả phải lớn hơn hoặc bằng thời gian hiện tại.

∀ dt ∈ DoiTra: dt.TG\_DoiTra >= GETDATE()

* Ngày bắt đầu của phiếu bảo hành phải lớn hơn hoặc bằng ngày hiện tại.

∀ pbh ∈ PhieuBH: pbh.Ngay\_BD >= CONVERT(DATE, GETDATE())

* Thời gian nhận bảo hành phải lớn hơn hoặc bằng thời gian hiện tại.

∀ bh ∈ NhanBH: bh.TG\_BaoHanh >= GETDATE()

2. Ràng buộc toàn vẹn liên thuộc tính

* Ngày kết thúc quá trình quản lý phải lớn hơn ngày bắt đầu.

∀ ql ∈ QTQuanLy: ql.Ngay\_KT > ql.Ngay\_BD

* Khi thay đổi giá thì giá mới phải khác giá cũ.

∀ td ∈ ThayDoiGia: td.GiaCu <> td.GiaMoi

* Thời gian kết thúc khuyến mãi phải lớn hơn thời gian bắt đầu.

∀ km ∈ KhuyenMai: km.ThoiGian\_KT > km.ThoiGian\_BD

* Thời gian kết thúc quá trình giao hàng phải lớn hơn thời gian bắt đầu.

∀ gh ∈ GiaoHang: gh.ThoiGian\_KT > gh.ThoiGian\_BD

* Ngày kết thúc bảo hành phải lớn hơn ngày bắt đầu.

∀ pbh ∈ PhieuBH: pbh.Ngay\_KT > pbh.Ngay\_BD

3. Ràng buộc toàn vẹn liên thuộc tính liên quan hệ

* Thời gian trả bảo hành phải lớn hơn thời gian nhận bảo hành.

VI. TRIGGER VÀ STORED PROCEDURE

1. Trigger

a. Khi thay đổi giá thì giá cũ phải giống giá hiện tại và khác với giá mới.

CREATE TRIGGER TG\_TDG\_ThayDoiGia ON ThayDoiGia

FOR INSERT, UPDATE

AS

BEGIN

IF EXISTS (

SELECT 1

FROM inserted i

INNER JOIN SanPham sp ON i.MaSP = sp.MaSP

WHERE i.GiaCu <> sp.Gia

)

BEGIN

RAISERROR ('Giá cũ phải giống với giá hiện tại của Sản phẩm.', 16, 1);

END;

IF EXISTS (

SELECT 1

FROM inserted

WHERE GiaCu = GiaMoi

)

BEGIN

RAISERROR ('Giá cũ phải khác với Giá mới.', 16, 1);

END;

UPDATE SanPham

SET Gia = (SELECT i.GiaMoi FROM inserted i)

WHERE MaSP = (SELECT i.MASP FROM inserted i)

END;

b. Nếu nhân viên đã nghỉ việc thì không được tạo đợt khuyến mãi.

CREATE TRIGGER TG\_KM\_CheckNhanVienNghiViec

ON KhuyenMai

FOR INSERT, UPDATE

AS

BEGIN

DECLARE @MaNV VARCHAR(10);

SELECT @MaNV = inserted.MaNV

FROM inserted;

IF NOT EXISTS (SELECT 1 FROM NhanVien WHERE MaNV = @MaNV AND NghiViec = 0)

BEGIN

RAISERROR('Nhân viên này hiện đang nghỉ việc.', 16, 1);

ROLLBACK TRANSACTION;

RETURN;

END;

END;

c. Nếu nhân viên đã nghỉ việc thì không được thay đổi giá sản phẩm.

CREATE TRIGGER TG\_TDG\_CheckNhanVienNghiViec

ON ThayDoiGia

FOR INSERT, UPDATE

AS

BEGIN

DECLARE @MaNV VARCHAR(10);

SELECT @MaNV = inserted.MaNV

FROM inserted;

IF NOT EXISTS (SELECT 1 FROM NhanVien WHERE MaNV = @MaNV AND NghiViec = 0)

BEGIN

RAISERROR('Nhân viên này hiện đang nghỉ việc.', 16, 1);

ROLLBACK TRANSACTION;

RETURN;

END;

END;

d. Nếu nhân viên đã nghỉ việc thì không được duyệt đơn hàng.

CREATE TRIGGER TG\_DDH\_CheckNhanVienNghiViec

ON DonDH

FOR INSERT, UPDATE

AS

BEGIN

DECLARE @MaNV VARCHAR(10);

SELECT @MaNV = inserted.MaNV

FROM inserted;

IF NOT EXISTS (SELECT 1 FROM NhanVien WHERE MaNV = @MaNV AND NghiViec = 0)

BEGIN

RAISERROR('Nhân viên này hiện đang nghỉ việc.', 16, 1);

ROLLBACK TRANSACTION;

RETURN;

END;

END;

e. Nếu nhân viên đã nghỉ việc thì không được giao hàng.

CREATE TRIGGER TG\_GH\_CheckNhanVienNghiViec

ON GiaoHang

FOR INSERT, UPDATE

AS

BEGIN

DECLARE @MaNV VARCHAR(10);

SELECT @MaNV = inserted.MaNV

FROM inserted;

IF NOT EXISTS (SELECT 1 FROM NhanVien WHERE MaNV = @MaNV AND NghiViec = 0)

BEGIN

RAISERROR('Nhân viên này hiện đang nghỉ việc.', 16, 1);

ROLLBACK TRANSACTION;

RETURN;

END;

END;

f. Nếu nhân viên đã nghỉ việc thì không được lập phiếu đổi trả.

CREATE TRIGGER TG\_DT\_CheckNhanVienNghiViec

ON DoiTra

FOR INSERT, UPDATE

AS

BEGIN

DECLARE @MaNV VARCHAR(10);

SELECT @MaNV = inserted.MaNV

FROM inserted;

IF NOT EXISTS (SELECT 1 FROM NhanVien WHERE MaNV = @MaNV AND NghiViec = 0)

BEGIN

RAISERROR('Nhân viên này hiện đang nghỉ việc.', 16, 1);

ROLLBACK TRANSACTION;

RETURN;

END;

END;

g. Nếu nhân viên đã nghỉ việc thì không được lập phiếu bảo hành.

CREATE TRIGGER TG\_PBH\_CheckNhanVienNghiViec

ON PhieuBH

FOR INSERT, UPDATE

AS

BEGIN

DECLARE @MaNV VARCHAR(10);

SELECT @MaNV = inserted.MaNV

FROM inserted;

IF NOT EXISTS (SELECT 1 FROM NhanVien WHERE MaNV = @MaNV AND NghiViec = 0)

BEGIN

RAISERROR('Nhân viên này hiện đang nghỉ việc.', 16, 1);

ROLLBACK TRANSACTION;

RETURN;

END;

END;

h. Nếu nhân viên đã nghỉ việc thì không được lập phiếu nhận bảo hành.

CREATE TRIGGER TG\_NBH\_CheckNhanVienNghiViec

ON NhanBH

FOR INSERT, UPDATE

AS

BEGIN

DECLARE @MaNV VARCHAR(10);

SELECT @MaNV = inserted.MaNV

FROM inserted;

IF NOT EXISTS (SELECT 1 FROM NhanVien WHERE MaNV = @MaNV AND NghiViec = 0)

BEGIN

RAISERROR('Nhân viên này hiện đang nghỉ việc.', 16, 1);

ROLLBACK TRANSACTION;

RETURN;

END;

END;

i. Nếu nhân viên đã nghỉ việc thì không được lập phiếu trả bảo hành.

CREATE TRIGGER TG\_TBH\_CheckNhanVienNghiViec

ON TraBH

FOR INSERT, UPDATE

AS

BEGIN

DECLARE @MaNV VARCHAR(10);

SELECT @MaNV = inserted.MaNV

FROM inserted;

IF NOT EXISTS (SELECT 1 FROM NhanVien WHERE MaNV = @MaNV AND NghiViec = 0)

BEGIN

RAISERROR('Nhân viên này hiện đang nghỉ việc.', 16, 1);

ROLLBACK TRANSACTION;

RETURN;

END;

END;

j. Trong một thời điểm chỉ tồn tại một quản lý.

CREATE TRIGGER TG\_QTQL\_TrangThaiQuanLy

ON QTQuanLy

AFTER INSERT, UPDATE

AS

BEGIN

SET NOCOUNT ON;

IF EXISTS (SELECT 1 FROM inserted WHERE TrangThai = 1)

BEGIN

IF EXISTS (SELECT 1 FROM QTQuanLy WHERE TrangThai = 1 AND MaQL NOT IN (SELECT MaQL FROM inserted WHERE TrangThai = 1))

BEGIN

RAISERROR ('Đang có nhân viên hiện là Quản lý.', 16, 1)

ROLLBACK TRANSACTION;

RETURN;

END

END

END;

k. Số lượng của sản phẩm ở giỏ hàng không được lớn hơn số lượng tồn của sản phẩm đó.

CREATE TRIGGER TG\_GioH\_CheckSLTon

ON GioHang

FOR INSERT, UPDATE

AS

BEGIN

SET NOCOUNT ON;

IF EXISTS (

SELECT 1

FROM inserted i

INNER JOIN SanPham s ON i.MaSP = s.MaSP

WHERE i.SoLuong > s.SLTon

)

BEGIN

RAISERROR ('Số lượng sản phẩm trong giỏ hàng không được lớn hơn số lượng tồn.', 16, 1)

ROLLBACK TRANSACTION;

RETURN;

END

END;

l. Cập nhật số lượng tồn của sản phẩm và giỏ hàng khi được đặt hàng.

CREATE TRIGGER [dbo].[TG\_CTDDH\_CapNhatSLTon]

ON [dbo].[CT\_DonDH]

FOR INSERT, UPDATE

AS

BEGIN

SET NOCOUNT ON;

Update GioHang

set soluong = soluong - (select inserted.SoLuong from inserted,GioHang where inserted.MaGio = GioHang.MaGio)

if exists (select 1 from GioHang where soluong<0)

begin

RAISERROR ('Số lượng không thể âm.', 16, 1);

ROLLBACK TRANSACTION;

end

UPDATE SanPham

SET SLTon = SLTon - (SELECT inserted.SoLuong

FROM SanPham, inserted, GioHang

WHERE inserted.MaGio = GioHang.MaGio

AND GioHang.MaSP = SanPham.MaSP)

IF EXISTS (SELECT 1 FROM SanPham WHERE SLTon < 0)

BEGIN

RAISERROR ('Số lượng tồn không thể âm.', 16, 1);

ROLLBACK TRANSACTION;

END

END;

m. Tự động thêm mã hóa đơn khi có đơn đặt hàng mới.

ALTER TRIGGER [dbo].[TG\_DDH\_ThemHoaDon]

ON [dbo].[DonDH]

AFTER INSERT

AS

BEGIN

INSERT INTO HoaDon (TongTien)

VALUES (0)

END;

n. Tự động thêm ngày kết thúc trong bảng Phiếu bảo hành.

CREATE TRIGGER [dbo].[TG\_PBH\_NgayKT]

ON [dbo].[PhieuBH]

FOR INSERT, UPDATE

AS

BEGIN

DECLARE @thang int

SET @thang = (

SELECT TG\_BaoHanh

FROM SanPham SP

INNER JOIN GioHang GH

ON SP.MaSP = GH.MaSP

INNER JOIN CT\_DonDH

ON GH.MaGio = CT\_DonDH.MaGio

INNER JOIN inserted i

ON CT\_DonDH.MaCTDDH = i.MaCTDDH)

UPDATE PhieuBH

SET Ngay\_KT = DATEADD(MONTH, @thang, Ngay\_BD)

END;

o. Tự động update trạng thái khi SoLuong bằng 0 trong bảng Giỏ hàng.

CREATE TRIGGER TG\_GioH\_UpdateTrangThaiGioHang

ON GioHang

AFTER INSERT, UPDATE

AS

BEGIN

SET NOCOUNT ON;

UPDATE GioHang

SET TrangThai = CASE

WHEN SoLuong = 0 THEN 'False'

ELSE 'True'

END

WHERE MaGio IN (SELECT MaGio FROM inserted);

END;

2. Stored Procedure

a. SP\_TaoTKNhanVien

CREATE PROCEDURE SP\_TaoTKNhanVien

@Ho NVARCHAR(50),

@Ten NVARCHAR(50),

@GioiTinh NVARCHAR(3),

@NgaySinh date,

@SDT NVARCHAR(20),

@cccd VARCHAR(12),

@Email NVARCHAR(100),

@Username NVARCHAR(50),

@Password NVARCHAR(50)

AS

BEGIN TRANSACTION

begin try

SET NOCOUNT ON;

INSERT INTO NhanVien (Ho, Ten, SDT, Email, NghiViec, GioiTinh, CCCD, NgaySinh)

VALUES (@Ho, @Ten, @SDT, @Email, 0, @GioiTinh, @cccd, @NgaySinh);

declare @MANV varchar(10)

SELECT TOP 1 @MANV = MANV FROM NHANVIEN ORDER BY MANV DESC

INSERT INTO TK\_NhanVien (usernameNV, pwNV, MaNV)

VALUES (@Username, @Password, @MaNV);

commit

end try

BEGIN CATCH

ROLLBACK

DECLARE @ErrorMessage VARCHAR(2000)

SELECT @ErrorMessage = 'Lỗi: ' + ERROR\_MESSAGE()

RAISERROR(@ErrorMessage, 16, 1)

END CATCH

b. SP\_NhanVienDangNhap

CREATE PROC SP\_NhanVienDangNhap

@username NVARCHAR(50),

@pw NVARCHAR(50)

AS

BEGIN

SET NOCOUNT ON;

IF EXISTS (SELECT 1 FROM TK\_NhanVien WHERE usernameNV = @username)

BEGIN

IF EXISTS (SELECT 1 FROM TK\_NhanVien WHERE usernameNV = @username AND pwNV = @pw)

BEGIN

SELECT NV.Ho, NV.Ten, NV.GioiTinh, NV.CCCD, NV.NgaySinh, NV.SDT, NV.Email

FROM NhanVien NV

INNER JOIN TK\_NhanVien TK ON NV.MaNV = TK.MaNV

WHERE TK.usernameNV = @username;

END

ELSE

BEGIN

RAISERROR('Sai mật khẩu.', 16, 1);

END

END

ELSE

BEGIN

RAISERROR('Tài khoản này không tồn tại.', 16, 1);

END

END;

c. SP\_ChinhSuaThongTinNhanVien

CREATE PROC SP\_ChinhSuaThongTinNhanVien

@MANV INT,

@HO NVARCHAR(100), @TEN NVARCHAR(50), @GIOITINH NVARCHAR(3), @CCCD VARCHAR(12),

@NGAYSINH DATE, @SDT VARCHAR(10), @EMAIL VARCHAR(50),@NGHIVIEC BIT

AS

BEGIN

UPDATE NhanVien

SET Ho=@HO,Ten=@TEN,GioiTinh=@GIOITINH,CCCD=@CCCD,NgaySinh=@NGAYSINH,SDT=@SDT,Email=@EMAIL,NghiViec=@NGHIVIEC

WHERE MANV = @MANV

END

d. SP\_TaoTKKhachHang

CREATE PROCEDURE SP\_TaoTKKhachHang

@Ho NVARCHAR(50),

@Ten NVARCHAR(50),

@SDT NVARCHAR(20),

@Email NVARCHAR(100),

@Username NVARCHAR(50),

@Password NVARCHAR(50)

AS

BEGIN

SET NOCOUNT ON;

INSERT INTO KhachHang (Ho, Ten, SDT, Email)

VALUES (@Ho, @Ten, @SDT, @Email);

declare @MaKH NVARCHAR(10)

select top 1 @MaKH = MaKH from KhachHang order by MaKH desc

INSERT INTO TK\_KhachHang (usernameKH, pwKH, MaKH)

VALUES (@Username, @Password, @MaKH);

END;e. SP\_KhachHangDangNhap

CREATE PROC SP\_KhachHangDangNhap

@username NVARCHAR(50),

@pw NVARCHAR(50)

AS

BEGIN

SET NOCOUNT ON;

IF EXISTS (SELECT 1 FROM TK\_KhachHang WHERE usernameKH = @username)

BEGIN

IF EXISTS (SELECT 1 FROM TK\_KhachHang WHERE usernameKH = @username AND pwKH = @pw)

BEGIN

SELECT KH.MaKH, KH.Ho, KH.Ten, KH.SDT, KH.Email

FROM KhachHang KH

INNER JOIN TK\_KhachHang TK ON KH.MaKH = TK.MaKH

WHERE TK.usernameKH = @username;

END

ELSE

BEGIN

RAISERROR('Sai mật khẩu.', 16, 1);

END

END

ELSE

BEGIN

RAISERROR('Tài khoản không tồn tại.', 16, 1);

END

END;

f. SP\_DoiMatKhauTKKhachHang

CREATE PROC SP\_DoiMatKhauTKKhachHang

@username NVARCHAR(50),

@oldPW NVARCHAR(50),

@newPW NVARCHAR(50)

AS

BEGIN

SET NOCOUNT ON;

DECLARE @existingPW NVARCHAR(50);

SELECT @existingPW = pwKH

FROM TK\_KhachHang

WHERE usernameKH = @username;

IF @existingPW <> @oldPW

BEGIN

RAISERROR ('Mật khẩu cũ không chính xác.', 16, 1);

RETURN;

END;

UPDATE TK\_KhachHang

SET pwKH = @newPW

WHERE usernameKH = @username;

END;

g. SP\_ChinhSuaThongTinKhachHang

CREATE PROCEDURE SP\_ChinhSuaThongTinKhachHang

@usernameKH VARCHAR(50),

@Ho NVARCHAR(100),

@Ten NVARCHAR(50),

@Sdt NVARCHAR(20),

@Email NVARCHAR(100)

AS

BEGIN

SET NOCOUNT ON;

DECLARE @MaKh VARCHAR(10);

SELECT @MaKh = MaKh

FROM TK\_KhachHang

WHERE usernameKH = @usernameKH;

IF @MaKh IS NULL

BEGIN

PRINT 'không tìm thấy KH nào';

RETURN;

END

UPDATE KhachHang

SET

Ho = COALESCE(@Ho, Ho),

Ten = COALESCE(@Ten, Ten),

Email = COALESCE(@Email, Email),

SDT = COALESCE(@Sdt, Sdt)

WHERE maKh = @MaKh

END;

h. SP\_ThongTinGioHangTheoTKKhachHang

CREATE PROC SP\_ThongTinGioHangTheoTKKhachHang

@username NVARCHAR(50)

AS

BEGIN

SET NOCOUNT ON;

IF EXISTS (SELECT 1 FROM GioHang WHERE usernameKH = @username AND TrangThai = 1)

BEGIN

SELECT SP.TenSP, SP.MauSac, SP.Size, GH.SoLuong, SP.Gia

FROM GioHang GH

INNER JOIN SanPham SP ON GH.MaSP = SP.MaSP

WHERE GH.usernameKH = @username AND GH.TrangThai = 1;

END

END;

i. SP\_ChinhSuaSoLuongSanPhamTrongGioHang

CREATE PROCEDURE SP\_ChinhSuaSoLuongSanPhamTrongGioHang

@MaSP INT,

@SoLuong TINYINT

AS

BEGIN

IF EXISTS (SELECT 1 FROM SanPham WHERE MaSP = @MaSP)

BEGIN

IF EXISTS (SELECT 1 FROM GioHang WHERE MaSP = @MaSP)

BEGIN

UPDATE GioHang

SET SoLuong = SoLuong + @SoLuong

WHERE MaSP = @MaSP;

PRINT N'Số lượng sản phẩm trong giỏ hàng đã được cập nhật.';

END

END

ELSE

BEGIN

RAISERROR('Không tồn tại tài khoản này.', 16, 1);

END

END

j. SP\_TaoDonHang

CREATE PROC SP\_TaoDonHang

@HINHTHUC NVARCHAR(20), @THANHTOAN NVARCHAR(20),@MANV VARCHAR(10)

AS

BEGIN

INSERT INTO DonDH(TG\_Dat,TrangThai,HinhThuc,ThanhToan,MaNV)

VALUES(GETDATE(),'Đang duyệt',@HINHTHUC,@THANHTOAN,@MANV);

END

k. SP\_XemDonHangTheoTKKhachHang

CREATE PROC SP\_XemDonHangTheoTKKhachHang

@username NVARCHAR(50)

AS

BEGIN

SET NOCOUNT ON;

IF EXISTS (SELECT 1 FROM TK\_KhachHang WHERE usernameKH = @username)

BEGIN

SELECT DISTINCT DonDH.MaDDH, @username, Ho, Ten, TG\_Dat, TrangThai, HinhThuc, ThanhToan

FROM (SELECT MaKH, Ho, Ten FROM KhachHang) KH

JOIN (SELECT usernameKH, MaKH FROM TK\_KhachHang) TKKH

ON KH.MaKH = TKKH.MaKH

JOIN (SELECT MaGio, usernameKH, MaSP FROM GioHang) GH

ON TKKH.usernameKH = GH.usernameKH

JOIN (SELECT MaSP, TenSP, Gia FROM SanPham) SP

ON GH.MaSP = SP.MaSP

JOIN (SELECT MaDDH, MaCTDDH, MaGio FROM CT\_DonDH) CTDDH

ON GH.MaGio = CTDDH.MaGio

JOIN DonDH

ON CTDDH.MaDDH = DonDH.MaDDH

WHERE TKKH.usernameKH = @username

END

ELSE

BEGIN

RAISERROR('Không tồn tại tài khoản này.', 16, 1);

END

END

l. SP\_XemDHTheoTrangThai

CREATE PROC SP\_XemDonDHDangXuLy

@trangthai NVARCHAR(30)

AS

BEGIN

SET NOCOUNT ON;

SELECT \*

FROM DonDH

WHERE TrangThai = @trangthai;

END;

m. SP\_TaoDotKhuyenMai

CREATE PROC SP\_TaoDotKhuyenMai

@manv INT,

@lyDo NVARCHAR(100),

@tgianBD DATETIME,

@tgianKT DATETIME

AS

BEGIN

SET NOCOUNT ON;

IF EXISTS (SELECT 1 FROM KhuyenMai WHERE ThoiGian\_BD = @tgianBD AND ThoiGian\_KT = @tgianKT)

BEGIN

RAISERROR('Đã tồn tại mã đợt khuyến mãi khác trùng với khoảng thời gian này.', 16, 1);

END

ELSE

BEGIN

INSERT INTO KhuyenMai(MaNV, LyDo, ThoiGian\_BD, ThoiGian\_KT)

VALUES (@manv, @lyDo, @tgianBD, @tgianKT)

END

END

n. SP\_ThayDoiThongTinSanPham

CREATE PROCEDURE [dbo].[SP\_ThemSanPhamMoi]

@tensp NVARCHAR(50),

@mausac NVARCHAR(10),

@size TINYINT,

@chatlieu NVARCHAR(30),

@xuatxu NVARCHAR(30),

@gia money,

@tg\_bh tinyint,

@slt smallint,

@mahang varchar(10),

@maloai varchar(10)

AS

BEGIN

INSERT INTO SanPham(TenSP, MauSac, Size, ChatLieu, XuatXu,Gia,TG\_BaoHanh,SLTon,MaHang,MaLoai)

VALUES (@tensp, @mausac, @size, @chatlieu, @xuatxu,@gia,@tg\_bh,@slt,@mahang,@maloai);

END;

o. SP\_ThayDoiThongTinSanPham

CREATE PROCEDURE SP\_ThayDoiThongTinSanPham

@masp INT,

@tensp NVARCHAR(50),

@mausac NVARCHAR(10),

@size TINYINT,

@chatlieu NVARCHAR(30),

@xuatxu NVARCHAR(30),

@gia money,

@tg\_bh tinyint,

@slt smallint,

@mahang varchar(10),

@maloai varchar(10)

AS

BEGIN

SET NOCOUNT ON;

UPDATE SanPham

SET TenSP = @tensp,

MauSac = @mausac,

Size = @size,

ChatLieu=@chatlieu,

XuatXu=@xuatxu,

Gia=@gia,

TG\_BaoHanh=@tg\_bh,

SLTon=@slt,

MaHang=@mahang,

MaLoai=@maloai

WHERE MaSP = @masp;

END;

p. SP\_TinhTienHoaDonTheoMa

ALTER PROC [dbo].[SP\_TinhTienHoaDonTheoMa]

@MADDH int,

@MANV int

AS

BEGIN

DECLARE @TONGTIEN FLOAT, @MAHD VARCHAR(20)

SELECT DISTINCT @MAHD = MAHD FROM CT\_DonDH WHERE MADDH = @MADDH

SELECT SP.MASP,SP.GIA, CTDH.SOLUONG, PhanTramKM = (

CASE

WHEN (SP.MASP IN (SELECT MASP FROM CT\_KhuyenMai) AND GETDATE() BETWEEN KM.ThoiGian\_BD AND KM.ThoiGian\_KT) THEN ISNULL(PhanTramKM,0)

ELSE 0

END), THANHTIEN = (

CASE

WHEN (SP.MASP IN (SELECT MASP FROM CT\_KhuyenMai) AND GETDATE() BETWEEN KM.ThoiGian\_BD AND KM.ThoiGian\_KT) THEN CTDH.SOLUONG\*SP.GIA\*((100-ISNULL(PhanTramKM,0))/100.00)

ELSE CTDH.SOLUONG\*SP.GIA

END)

INTO #TEMP

FROM CT\_DonDH CTDH INNER JOIN GioHang GH ON CTDH.MaGio = GH.MaGio INNER JOIN SanPham SP ON GH.MaSP = SP.MASP

LEFT JOIN CT\_KhuyenMai CTKM ON SP.MaSP = CTKM.MASP LEFT JOIN KhuyenMai KM ON CTKM.MaDotKM = KM.MaDotKM

WHERE CTDH.MADDH = @MADDH

SELECT @TONGTIEN = SUM(THANHTIEN) FROM #TEMP

UPDATE HoaDon SET TongTien=@TONGTIEN,ThoiGian=GETDATE(), MaNV = @MANV WHERE MAHD = @MAHD

END

q. SP\_TongDoanhThuTheoNam

CREATE PROC SP\_DoanhThuTheoNam

@nam INT

AS

BEGIN

SELECT DOANHTHU = SUM(TongTien)

FROM HoaDon

WHERE YEAR([ThoiGian]) = @nam

END;

r. SP\_TongDoanhThuTheoKhoangTG

CREATE PROC SP\_TongDoanhThuTheoKhoangTG

@fromDate DATETIME,

@toDate DATETIME

AS

BEGIN

SET NOCOUNT ON;

SELECT SUM(TongTien) AS TongDoanhThu

FROM HoaDon

WHERE ThoiGian BETWEEN @fromDate AND @toDate;

END;

3. View

a. Xem các sản phẩm hiện có

CREATE VIEW V\_XemCacSanPhamHienCo AS

SELECT \* FROM SanPham

b. Xem doanh thu theo ngày hiện tại

CREATE VIEW V\_XemDoanhThuTheoNgayHienTai AS

SELECT SUM(TongTien) AS TongDoanhThu

FROM HoaDon

WHERE CONVERT(date, ThoiGian) = CONVERT(date, GETDATE());

VII. PHÂN QUYỀN

1. Quản lý:

Vai trò và Quyền hạn:

* Nhân viên: Quản lý các thông tin liên quan đến nhân viên trong hệ thống.
* Tạo tài khoản nhân viên: Quản lý có thể tạo tài khoản cho nhân viên mới.
* Tạo tài khoản Quản lý: Quản lý có thể tạo tài khoản khi có Quản lý mới lên thay.
* Chỉ xem tất cả các thông tin về cửa hàng (trừ thông tin của Khách hàng).
* Xem doanh thu và báo cáo.

Quyền trong SQL Server:

* db\_datareader: Đọc dữ liệu từ tất cả các bảng trong cơ sở dữ liệu.
* db\_securityadmin: Quản lý các quyền bảo mật, có thể thay đổi quyền của các tài khoản khác.
* db\_accessadmin: Quản lý quyền truy cập các tài khoản vào cơ sở dữ liệu.

A screenshot of a computer

Description automatically generated

1. Nhân viên:

Vai trò và Quyền hạn:

* Xem và sửa dữ liệu: Có thể xem và chỉnh sửa dữ liệu các bảng Sản phẩm, Loại, Hãng, Thay đổi giá, Khuyến mãi, Chi tiết khuyến mãi, Giao hàng, Nhận bảo hành, Trả bảo hành.
* Tạo tài khoản Khách hàng: Nhân viên có thể tạo tài khoản mới cho khách hàng.

Quyền trong SQL Server:

* db\_datareader: Đọc dữ liệu từ một số bảng trong cơ sở dữ liệu.
* db\_datawriter: Ghi dữ liệu một số bảng trong cơ sở dữ liệu.
* db\_accessadmin: Quản lý quyền truy cập các tài khoản vào cơ sở dữ liệu.

A screenshot of a computer

Description automatically generated

1. Khách hàng:

Vai trò và quyền hạn:

* Tạo tài khoản: Khách hàng có thể tự tạo tài khoản mới.
* Thao tác:
  + Có thể xem danh sách sản phẩm.
  + Có thể thêm sản phẩm vào giỏ hàng.
  + Có thể tạo đơn đặt hàng mới.
* Sửa dữ liệu: Khách hàng có thể chỉnh sửa giỏ hàng và thông tin cá nhân của mình.

Quyền trong SQL Server:

* db\_datareader: Đọc dữ liệu từ một số bảng trong cơ sở dữ liệu.
* db\_datawriter: Ghi dữ liệu vào một số bảng trong cơ sở dữ liệu.

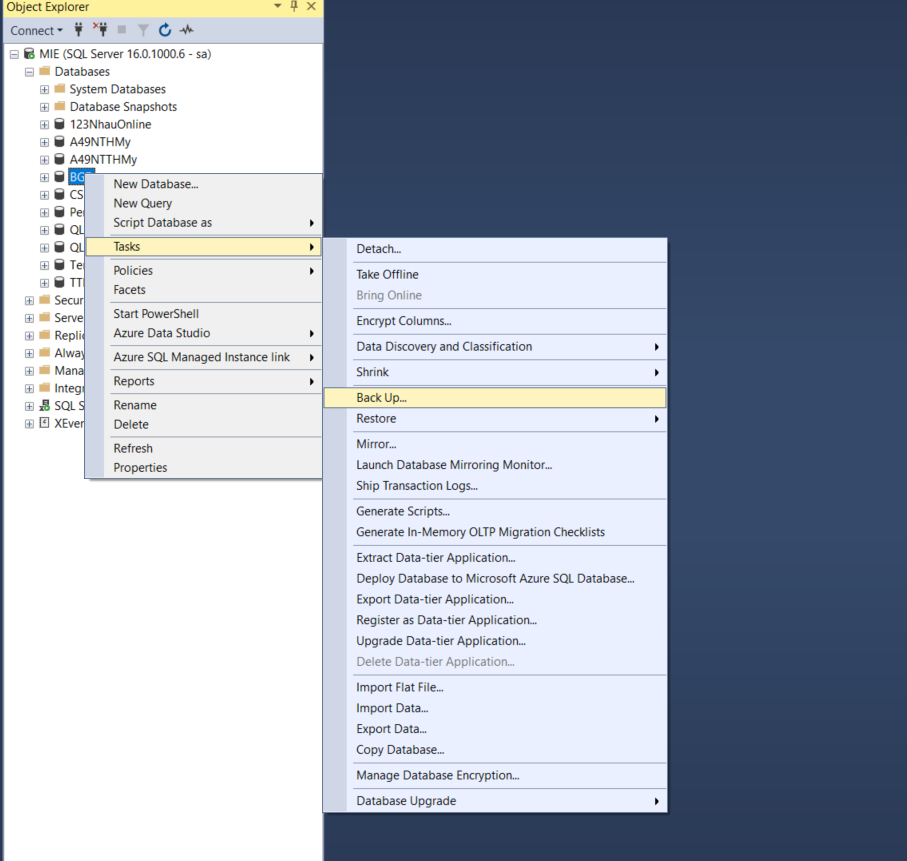
A screenshot of a computer

Description automatically generated

D. Backup

I. Backup

1. Chọn Cơ sở dữ liệu cần Backup rồi ấn Tasks 🡪 Backup.



2. Chọn Add để thêm Đường dẫn tới thư mục lưu bản Backup.

A screenshot of a computer

Description automatically generated

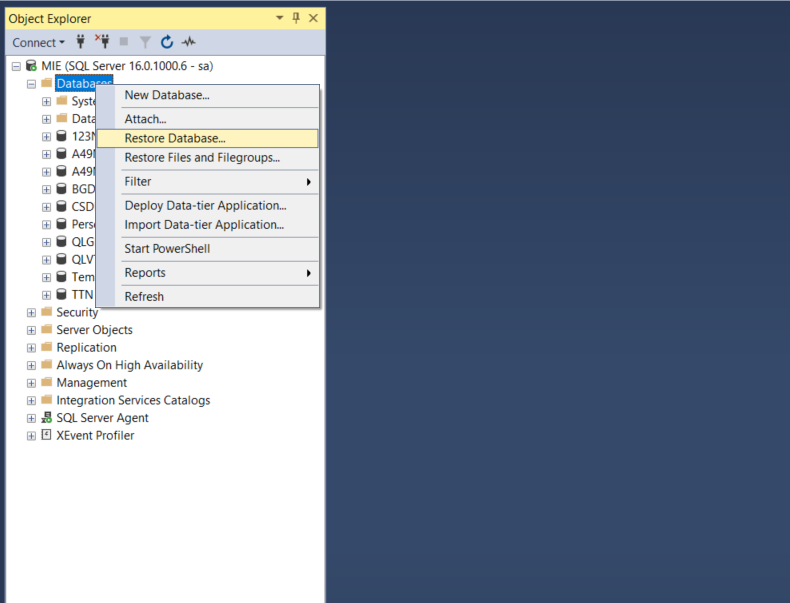
A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

II. Restore

1. Trong SQL Server, nhấp chuột phải vào Databases chọn Restore Database.



2. Chọn Device và ấn Add để chọn file Backup rồi nhấn OK.

A screenshot of a computer

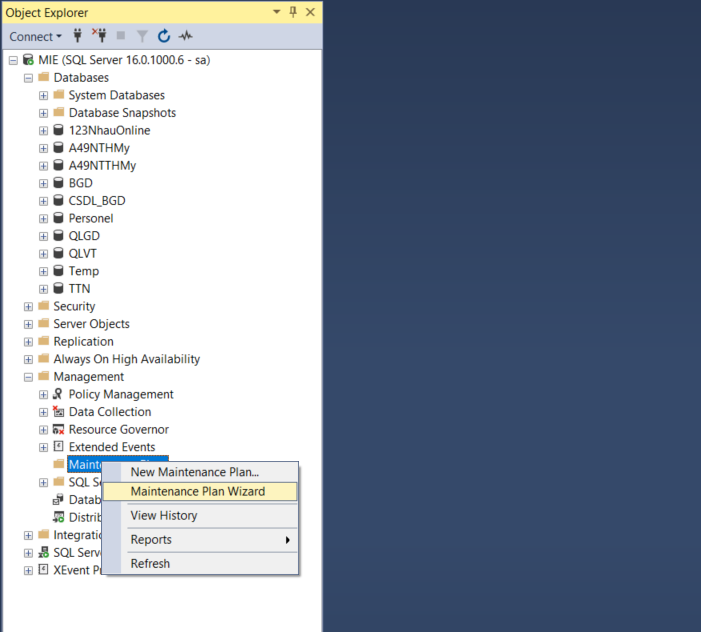
Description automatically generated

A screenshot of a computer

Description automatically generated

III. Lập lịch Backup

1. Vào management 🡪 Maintenance Plans 🡪 Maintenance Plan Wizard



2. Bấm Next

A screenshot of a computer

Description automatically generated

3. Đặt tên cho Job rồi chọn lịch Backup.

A screenshot of a computer

Description automatically generated

4. Chọn Back Up Database (Full)

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

5. Ở tab General, chọn những Database cần backup.

A screenshot of a computer

Description automatically generated

6. Qua tab Destination để chọn vị trí backup.

A screenshot of a computer

Description automatically generated

7. Chọn vị trí lưu file Report rồi bấm Finish để kết thúc.

A screenshot of a computer

Description automatically generated