ASSIGNMENT

**Manage a computer store**

**I. Statement of the problem:**

- The goal of this project is to build a database system to manage computer store operations, including managing products, orders, customers, warehouses, etc. effectively

- Identify the entity

* **Khách hàng** (mã khách hàng, tên khách hàng, số điện thoại, email, địa chỉ,giới tính )
* **Sản phẩm** (mã sản phẩm, tên sản phẩm, giá sản phẩm, mã nhà cung cấp)
* **Nhà cung cấp** (mã nhà cung cấp, tên nhà cung cấp, số điện thoại, email)
* **Hóa đơn** (mã hóa đơn, mã khách hàng, mã nhân viên, ngày đặt hàng)
* **Hóa đơn chi tiết** ( mã hóa đơn chi tiết, mã hóa đơn, mã sản phẩm, số lượng )
* **Nhân viên** (mã nhân viên, tên nhân viên, số điện thoại, email, địa chỉ, chức vụ, giới tính)
* **Phiếu nhập** ( mã phiếu nhập, ngày nhập, mã sản phẩm, số lượng, mã nhân viên )
* **Phiếu xuất** ( mã phiếu xuất, ngày xuất,mã hóa đơn, mã sản phẩm, số lượng )
* **Bảo hành** ( mã bảo hành, mã sản phẩm, mô tả )
* **Mã giảm giá** ( mã giảm giá, giá trị giảm, ngày bắt đầu, ngày kết thúc, mô tả )
* **Thông tin vận chuyển** ( mã vận chuyển, mã hóa đơn, mã đơn hàng, ngày vận chuyển, địa chỉ giao hàng, phí giao hàng, đơn vị vận chuyển )
* **Kho** ( mã kho, tên kho, địa chỉ, mã phiếu nhập, mã phiếu xuất)

- Determine the relationship

Khách hàng – [1] —--------- [1]|[N]-- Hóa đơn

Hóa đơn – [1] —--------- [1] – Khách hàng

Hóa đơn – [1] —--------- [1] – Nhân viên

Hóa đơn – [1] —--------- [1] – Hóa đơn chi tiết

Nhân viên – [1] —------- [N] – Hóa đơn

Phiếu nhập – [1] —-----[1]| [N] – Sản phẩm

Phiếu nhập – [N]|[1] —----- [1] – Kho

Phiếu xuất – [1] —------[ N] – Sản phẩm

Phiếu xuất – [N]|[1] —------[1] – Kho

Phiếu xuất – [1] —------[1] – Hóa đơn

Bảo hành — [1] —------ [1] – Sản phẩm

Mã giảm giá – [1]|[0] —----[ 1] – Đơn hàng

Thông tin vận chuyển – [1]|[0] —----- [1]-- Hóa đơn

Kho – [1] —-- [N] – Sản phẩm

Nhà cung cấp –[1] — [N] – Sản phẩm

* Each supplier will provide multiple products. Each product is provided by only one manufacturer.
* Products will be received through a goods receipt note. Each goods receipt note will receive one or multiple products.
* Each stock receipt may have one or multiple goods receipt notes.
* Each stock issue will use a stock issue note, which may use one or multiple stock issue notes.
* Each invoice will be generated corresponding to one stock issue note.
* Each invoice may or may not use a discount code.
* Each product has warranty information.
* Each invoice corresponds to information about one customer.
* Each customer may make multiple purchases corresponding to multiple invoices.
* Each employee will print multiple invoices (payment for products to customers), and each invoice is printed by one employee.
* Each invoice has a detailed invoice.
* Each invoice may or may not have a shipping service.

**II.Description of system operations:**

**1. CUSTOMER MANAGEMENT**

* Customer information is saved after purchasing the product. Information includes customer code, customer name, phone number, and email.
* If customers use a shipping service, additional shipping fees will be charged

**2. BILL MANAGEMENT**

* When customers need to buy products, staff will issue an invoice to the customer.
* If the customer uses a shipping service, the invoice will have a shipping code added
* If the customer uses a discount code, the invoice will have an additional discount code

**3. PRODUCT MANAGEMENT**

* Information includes product code, product name, price, and supplier code.
* Along with the product, information about the supplier will be saved. Information includes supplier code, supplier name, email, and phone number.

**4. STAFF MANAGEMENT**

* Staff will make an invoice when customers buy products
* Staff will carry out the warehousing process
* Staff will make deliveries to the shipping unit to deliver goods to customers if customers need to use shipping services.

**5. WAREHOUSE MANAGEMENT**

* The warehouse will store import slips, export slips, warehouse codes, and locations

**6. WARRANTY MANAGEMENT**

* Each product has a unique warranty, and can only be warranted before the warranty expires

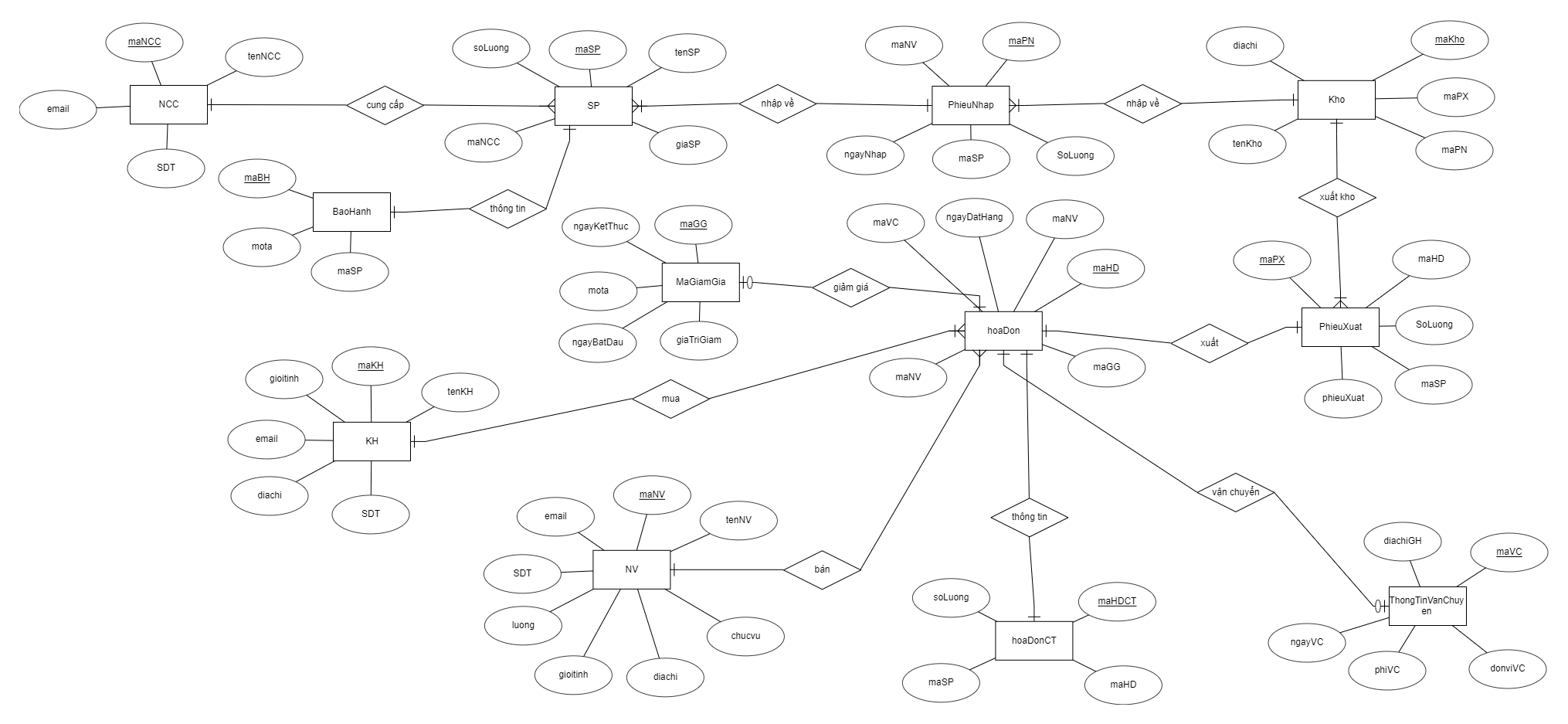
**7. MANAGE DISCOUNT CODES**

* The customer's invoice will be reduced when the discount code is applied. Discount codes can only be applied if the conditions are met

**8. MANAGE SHIPPING INFORMATION**

* When customers need to use shipping services, the invoice will have a shipping code added

**III. Describe the ER image of the system:**



**IV. Data requirements specification (data dictionary) and list of data constraints**

1. **Bảng khách hàng**

| Tên trường | Kiểu dl | Check | Key/Index/Ràng buộc |
| --- | --- | --- | --- |
| maKH | Char(5) |  | PK |
| tenKH | Nvarchar(200) |  | Not null |
| SDT | Char(10) | Từ 0 đến 9 | Not null |
| email | Nvarchar(200) |  | Not null |
| gioitinh | Nvarchar(10) | Nam,nữ, khác |  |
| diachi | Nvarchar(200) |  |  |

CREATE TABLE KH (

maKH Char(5) PRIMARY KEY,

tenKH Nvarchar(200) NOT NULL,

SDT NVARCHAR(10) CHECK (SDT LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]') NOT NULL,

email Nvarchar (200) NOT NULL,

gioitinh Nvarchar(10) CHECK (GioiTinh IN ('Nam', 'Nữ', 'Khác')),

diachi Nvarchar(200) NOT NULL

);

1. **Bảng sản phẩm**

| Tên trường | Kiểu dl | Check | Key/Index/Ràng buộc |
| --- | --- | --- | --- |
| maSP | Char(5) |  | PK |
| tenSP | Nvarchar(5) |  | Not null |
| giaSP | Decimal(20) | >0 | Not null |
| maNCC | Char(5) |  | FK |

CREATE TABLE SP (

maSP Char(5) PRIMARY KEY,

tenSP NVARCHAR(200) NOT NULL,

giaSP Decimal(20) CHECK ( giaSP > 0 ),

maNCC Char(5) NOT NULL,

FOREIGN KEY (maNCC) REFERENCES NCC(maNCC)

);

1. **Bảng nhà cung cấp**

| Tên trường | Kiểu dl | Check | Key/Index/Ràng buộc |
| --- | --- | --- | --- |
| maNCC | Char(5) |  | PK |
| tenNCC | Nvarchar(200) |  | Not null |
| SDT | Char(10) | Từ 0 đến 9 | Unique |
| email | Nvarchar(200) |  | Unique |

CREATE TABLE NCC (

maNCC Char(5) PRIMARY KEY,

tenNCC NVARCHAR(200) NOT NULL,

SDT NVARCHAR(10) CHECK (SDT LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]') NOT NULL,

email NVARCHAR(200) NOT NULL

);

1. **Bảng hóa đơn**

| Tên trường | Kiểu dl | Key/Index/Ràng buộc |
| --- | --- | --- |
| maHoaDon | Char(7) | PK |
| maKH | Char(5) | FK |
| maNV | Char(5) | FK |
| ngayDatHang | Date | Not null |

CREATE TABLE hoaDon (

maHD Char(7) PRIMARY KEY,

maKH Char(5) NOT NULL,

maNV Char(5) NOT NULL,

maGG Char(5),

maVC Char(5),

ngayDatHang DATE NOT NULL,

FOREIGN KEY (maKH) REFERENCES KH(maKH),

FOREIGN KEY (maNV) REFERENCES NV(maNV),

FOREIGN KEY (maGG) REFERENCES maGiamGia(maGG),

FOREIGN KEY (maVC) REFERENCES ThongTinVanChuyen(maVC),

);

1. **Bảng hóa đơn chi tiết**

| Tên trường | Kiểu dl | Check | Key/Index/Ràng buộc |
| --- | --- | --- | --- |
| maHoaDonCT | Char(7) |  | PK |
| maHoaDon | Char(7) |  | FK |
| maSP | Char(5) |  | FK |
| soLuong | Int | >0 | Not null |

CREATE TABLE hoaDonCT (

maHDCT Char(7) PRIMARY KEY,

maHD Char(7) NOT NULL,

maSP Char(5) NOT NULL,

SoLuong Int CHECK ( SoLuong > 0 ),

FOREIGN KEY (maSP) REFERENCES SP(maSP),

FOREIGN KEY (maHD) REFERENCES hoaDon(maHD)

);

1. **Bảng kho**

| Tên trường | Kiểu dl | Check | Key/Index/Ràng buộc |
| --- | --- | --- | --- |
| maKho | Char(5) |  | PK |
| diachi | Nvarchar(200) |  | Not null |
| tenKho | Nvarchar(200) | >0 | Not null |
| maPX | Char(5) |  | FK |
| maPN | Char(5) |  | FK |

CREATE TABLE Kho (

maKho Char(5) PRIMARY KEY,

tenKho Nvarchar(200) NOT NULL,

diachi Nvarchar(200) NOT NULL,

maPX Char(5) NOT NULL,

maPN Char(5) NOT NULL,

FOREIGN KEY (maPX) REFERENCES PhieuXuat(maPX),

FOREIGN KEY (maPN) REFERENCES PhieuNhap(maPN)

);

1. **Bảng phiếu nhập**

| Tên trường | Kiểu dl | Check | Key/Index/Ràng buộc |
| --- | --- | --- | --- |
| maPN | Char(5) |  | PK |
| maNV | Char(5) |  | FK |
| ngayNhap | DATE |  | Not null |
| maSP | Char(5) |  | FK |
| SoLuong | Int | >0 | Not null |

CREATE TABLE PhieuNhap (

maPN Char(5) PRIMARY KEY,

maNV Char(5) NOT NULL,

maSP Char(5) NOT NULL,

ngayNhap DATE NOT NULL,

SoLuong Int CHECK ( SoLuong > 0 ),

FOREIGN KEY (maSP) REFERENCES SP(maSP),

FOREIGN KEY (maNV) REFERENCES NV(maNV)

);

1. **Bảng phiếu xuất**

| Tên trường | Kiểu dl | Check | Key/Index/Ràng buộc |
| --- | --- | --- | --- |
| maPX | Char(5) |  | PK |
| maHD | Char(5) |  | FK |
| ngayXuat | DATE |  | Not null |
| maSP | Char(5) |  | FK |
| SoLuong | Int | >0 | Not null |

CREATE TABLE PhieuXuat (

maPX Char(5) PRIMARY KEY,

maSP Char(5) NOT NULL,

maHD Char(7) NOT NULL,

ngayXuat DATE NOT NULL,

SoLuong Int CHECK ( SoLuong > 0 ),

FOREIGN KEY (maSP) REFERENCES SP(maSP),

FOREIGN KEY (maHD) REFERENCES hoaDon(maHD)

);

1. **Bảng bảo hành**

| Tên trường | Kiểu dl | Check | Key/Index/Ràng buộc |
| --- | --- | --- | --- |
| maBH | Char(5) |  | PK |
| maSP | Char(5) |  | FK |
| mota | Nvarchar(200) |  | Not null |

CREATE TABLE BaoHanh (

maBH Char(5) PRIMARY KEY,

maSP Char(5) NOT NULL,

mota Nvarchar(200) NOT NULL,

FOREIGN KEY (maSP) REFERENCES SP(maSP)

);

1. **Bảng mã gỉam giá**

| Tên trường | Kiểu dl | Check | Key/Index/Ràng buộc |
| --- | --- | --- | --- |
| maGG | Char(5) |  | PK |
| maHD | Char(5) |  | FK |
| ngayBatDau | DATE |  | Not null |
| ngayKetThuc | DATE |  | Not null |
| mota | Nvarchar(200) |  | Not null |
| giatriGiam | Decimal(20) | >0 | Not null |

CREATE TABLE MaGiamGia (

maGG Char(5) PRIMARY KEY,

ngayBatDau DATE NOT NULL,

ngayKetThuc DATE NOT NULL,

mota Nvarchar(200) NOT NULL,

giatriGiam Decimal(20) CHECK ( giatriGiam >0 ),

);

1. **Bảng thông tin vân chuyển**

| Tên trường | Kiểu dl | Check | Key/Index/Ràng buộc |
| --- | --- | --- | --- |
| maVC | Char(5) |  | PK |
| maHD | Char(5) |  | FK |
| ngayVC | DATE |  | Not null |
| diachiGH | Nvarchar(200) |  | Not null |
| donviVC | Nvarchar(200) |  | Not null |
| phiVC | Decimal(20) |  | Not null |

CREATE TABLE ThongTinVanChuyen(

maVC Char(5) PRIMARY KEY,

ngayVC DATE NOT NULL,

diachiGH Nvarchar(200) NOT NULL,

donviVC Nvarchar (200) NOT NULL,

phiVC Decimal (20) NOT NULL,

);

1. **Bảng nhân viên**

| Tên trường | Kiểu dl | Check | Key/Index/Ràng buộc |
| --- | --- | --- | --- |
| maNV | Char(5) |  | PK |
| tenNV | Nvarchar(200) |  | Not null |
| SDT | Char(10) | Từ 0 đến 9 | Not null |
| email | Nvarchar(200) |  | Not null |
| gioitinh | Nvarchar(10) | Nam,nữ, khác |  |
| diachi | Nvarchar(200) |  |  |
| chucvu | Nvarchar(200) |  | Not null |

CREATE TABLE NV (

maNV Char(5) PRIMARY KEY,

tenNV Nvarchar(200) NOT NULL,

SDT NVARCHAR(10) CHECK (SDT LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]') NOT NULL,

email Nvarchar (200) NOT NULL,

gioitinh Nvarchar(10) CHECK (GioiTinh IN ('Nam', 'Nữ', 'Khác')),

diachi Nvarchar(200) NOT NULL,

chucvu Nvarchar(200) NOT NULL

);

V. Physical installation

1. Các câu truy vấn cùng câu trả lời SQL tương ứng
2. Liệt kê mã hóa đơn, ngày đặt hàng, tên khách hàng, và tổng số lượng sản phẩm đã mua trong mỗi hóa đơn.

SELECT hoaDon.maHD, hoaDon.ngayDatHang, KH.tenKH, SUM(hoaDonCT.SoLuong) AS TongSoLuongMua

FROM hoaDon

JOIN KH ON hoaDon.maKH = KH.maKH

JOIN hoaDonCT ON hoaDon.maHD = hoaDonCT.maHD

GROUP BY hoaDon.maHD, hoaDon.ngayDatHang, KH.tenKH;

1. Hiển thị thông tin sản phẩm và tổng số lượng sản phẩm đã xuất cho mỗi sản phẩm.

SELECT SP.maSP, SP.tenSP, SUM(PX.SoLuong) AS TongSoLuongXuat

FROM SP

JOIN PhieuXuat PX ON SP.maSP = PX.maSP

GROUP BY SP.maSP, SP.tenSP;

1. Tính tổng doanh thu của mỗi hóa đơn:

SELECT hoaDon.maHD, SUM(SP.giaSP \* hoaDonCT.SoLuong) AS TongDoanhThu

FROM hoaDon

JOIN hoaDonCT ON hoaDon.maHD = hoaDonCT.maHD

JOIN SP ON hoaDonCT.maSP = SP.maSP

GROUP BY hoaDon.maHD;

1. Liệt kê các sản phẩm có doanh thu cao nhất trong mỗi nhóm (nhà cung cấp):

SELECT NCC.tenNCC, SP.tenSP, SUM(SP.giaSP \* hoaDonCT.SoLuong) AS TongDoanhThu

FROM NCC

JOIN SP ON NCC.maNCC = SP.maNCC

JOIN hoaDonCT ON SP.maSP = hoaDonCT.maSP

GROUP BY NCC.tenNCC, SP.tenSP

ORDER BY SUM(SP.giaSP \* hoaDonCT.SoLuong) DESC;

1. Liệt kê tất cả các sản phẩm được bán với thông tin chi tiết của sản phẩm và tổng số lượng đã bán:

SELECT SP.maSP, SP.tenSP, SP.giaSP, SUM(hoaDonCT.SoLuong) AS TongSoLuongBan

FROM SP

JOIN hoaDonCT ON SP.maSP = hoaDonCT.maSP

GROUP BY SP.maSP, SP.tenSP, SP.giaSP;

1. Lấy danh sách các hóa đơn và tổng giá trị của mỗi hóa đơn, sắp xếp theo giá trị giảm dần

SELECT hoaDon.maHD, KH.tenKH, SUM(SP.giaSP \* hoaDonCT.soLuong) AS TongGiaTri

FROM hoaDon

JOIN hoaDonCT ON hoaDon.maHD = hoaDonCT.maHD

JOIN KH ON hoaDon.maKH = KH.maKH

JOIN SP ON hoaDonCT.maSP = SP.maSP

GROUP BY hoaDon.maHD, KH.tenKH

ORDER BY TongGiaTri DESC;

1. Lấy tổng giá trị của tất cả các hóa đơn trong cửa hàng.

SELECT SUM(SP.giaSP \* hoaDonCT.soLuong) AS TongGiaTriHoaDon

FROM hoaDonCT

JOIN SP ON hoaDonCT.maSP = SP.maSP;

1. Lấy danh sách các hóa đơn, bao gồm mã hóa đơn, tên khách hàng, tên sản phẩm, giá sản phẩm, số lượng và tổng giá trị của mỗi hóa đơn.

SELECT hoaDon.maHD, KH.tenKH, SP.tenSP, SP.giaSP, hoaDonCT.soLuong, SP.giaSP \* hoaDonCT.soLuong AS TongGiaTri

FROM hoaDon

JOIN hoaDonCT ON hoaDon.maHD = hoaDonCT.maHD

JOIN KH ON hoaDon.maKH = KH.maKH

JOIN SP ON hoaDonCT.maSP = SP.maSP;

2. Function

1. Tìm sản phẩm theo tên

CREATE FUNCTION TimSanPhamTheoTen

(

@tenSanPham nvarchar(200)

)

RETURNS TABLE

AS

RETURN (

SELECT maSP, tenSP, giaSP, maNCC

FROM SP

WHERE tenSP LIKE '%' + @tenSanPham + '%'

);

1. Tính số dư sản phẩm trong kho

CREATE FUNCTION TinhSoDuSanPhamTrongKho

(

@maSP Char(5)

)

RETURNS INT

AS

BEGIN

DECLARE @SoLuongNhap INT;

DECLARE @SoLuongXuat INT;

DECLARE @SoDu INT;

-- Tính tổng số lượng sản phẩm đã nhập vào kho

SELECT @SoLuongNhap = SUM(SoLuong)

FROM PhieuNhap

WHERE maSP = @maSP;

-- Tính tổng số lượng sản phẩm đã xuất từ kho

SELECT @SoLuongXuat = SUM(SoLuong)

FROM PhieuXuat

WHERE maSP = @maSP;

-- Tính số dư sản phẩm trong kho

SET @SoDu = ISNULL(@SoLuongNhap, 0) - ISNULL(@SoLuongXuat, 0);

--Tính số dư sản phẩm trong kho bằng cách lấy tổng số lượng nhập trừ đi tổng số lượng xuất. Hàm ISNULL được sử dụng để xử lý trường hợp không có dữ liệu.--

RETURN @SoDu;

END;

3. Trigger

1. Check valid email

CREATE TRIGGER CheckValidEmail\_Gmail

ON KH

AFTER INSERT, UPDATE

AS

BEGIN

IF EXISTS (SELECT 1 FROM inserted WHERE email NOT LIKE '%@gmail.com')

BEGIN

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

ROLLBACK TRANSACTION;

END

END;

4. View

1. View\_SanPhamConTrongKho

SELECT SP.maSP, SP.tenSP, dbo.TinhSoDuSanPhamTrongKho(SP.maSP) AS SoDu

FROM SP;

1. View\_ChiTietHoaDon

SELECT hoaDon.maHD, hoaDon.ngayDatHang, KH.tenKH, SP.tenSP, SP.giaSP, hoaDonCT.SoLuong, SP.giaSP \* hoaDonCT.SoLuong AS TongGiaTri

FROM hoaDon

JOIN hoaDonCT ON hoaDon.maHD = hoaDonCT.maHD

JOIN KH ON hoaDon.maKH = KH.maKH

JOIN SP ON hoaDonCT.maSP = SP.maSP;

1. View\_DanhSachSanPham

SELECT SP.maSP, SP.tenSP, SP.giaSP, NCC.tenNCC

FROM SP

JOIN NCC ON SP.maNCC = NCC.maNCC;

VI. Conclude

* We looked at the process of designing and creating an SQL database for a computer shop. This database includes important tables such as Products, Suppliers, Customers, Employees, and Invoices. We have ensured that this database follows the principles of Split Normalization (3NF) to ensure data integrity and performance.
* Some important points in this process include:
* Design tables with appropriate data structures, including constraints, checks, and foreign keys to ensure data integrity.
* Use SQL queries to query data from the database, including complex queries such as getting total invoice value, finding best-selling products, and finding low-quantity products.
* Consider using a database to track lists of products, invoices, customers, and suppliers.
* Ensure data security by applying check constraints and using foreign keys to verify data association and consistency.
* Finally, creating an SQL database is an important step in building and managing a computer shop. The database helps track product information, manage invoices, interact with customers and suppliers, and provide information needed for business decision-making. This helps create an efficient business environment and provides value to businesses and customers.