**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THỰC PHẨM TP. HỒ CHÍ MINH**



**TIỂU LUẬN : HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU ORACLE**

**ĐỀ TÀI: Quản lí cửa hàng bán đồng hồ**

Giảng viên hướng dẫn: **Nguyễn Đình Thành**

Nhóm sinh viên thực hiện: Nhóm 5

**THÀNH PHỐ HỒ CHÍ MINH, THÁNG 9 NĂM 2023**

# Lời cảm ơn

Để hoàn thành được bài tiểu luận này, em xin chân thành cảm ơn Ban Giám hiệu, các khoa, phòng và quý thầy, cô của trường Đại Học Công Thương Thành Phố Hồ Chí Minh những người đã tận tình giúp đỡ và tạo điều kiện cho em trong quá trình học tập. Đặc biệt, em xin gửi lời cảm ơn sâu sắc đến thầy Nguyễn Đình Thành - người đã trực tiếp giảng dạy và hướng dẫn em thực hiện bài tiểu luận này bằng tất cả lòng nhiệt tình và sự quan tâm sâu sắc.

Trong quá trình thực hiện bài tiểu luận này, do hiểu biết còn nhiều hạn chế nên bài làm khó tránh khỏi những thiếu sót. Em rất mong nhận được những lời góp ý của quý thầy cô để bài tiểu luận ngày càng hoàn thiện hơn.

Em xin chân thành cảm ơn!

**BẢNG PHÂN CÔNG VIỆC**

|  |  |  |
| --- | --- | --- |
| **Họ và tên** | **Công việc** | **Đánh giá** |
| **Phạm Minh Nhựt - 2001216012** | **Phân tích nghiệp vụ, code ứng dụng, viết csdl,demo** | **40%** |
| **Huỳnh Khánh Nam**  **2001215968** | **Code ứng dụng, viết csdl** | **30%** |
| **Nguyễn Minh Trí**  **2001216238** | **Code ứng dụng, viết csdl** | **30%** |

**MỤC LỤC**

[Lời cảm ơn 2](#_Toc1559)

[CHƯƠNG 1. GIỚI THIỆU 1](#_Toc2854)

[1.1. Lý do chọn đề tài 1](#_Toc15267)

[1.2. Mô tả bài toán 1](#_Toc16033)

[1.3. Mô tả chức năng hệ thống 1](#_Toc15218)

[1.4. Mô hình quan hệ ERD 2](#_Toc10766)

[1.5. Mô hình Daiaram 2](#_Toc3005)

[1.6. Mô tả các bảng 2](#_Toc2075)

[1.7. Các thực thể 5](#_Toc28912)

[1.8. Mối quan hệ 5](#_Toc5650)

[CHƯƠNG 2. THIẾT KẾ VÀ CÀI ĐẶT CƠ SỞ DỮ LIỆU 6](#_Toc23418)

[2.1. Cài đặt các bảng 6](#_Toc12748)

[2.2. Câu lệnh 8](#_Toc11366)

[2.3. Truy vấn 9](#_Toc12375)

[2.4. Hàm 12](#_Toc17401)

[2.5. Thủ tục 13](#_Toc31401)

[2.6. Trigger 17](#_Toc20997)

[CHƯƠNG 3. CÀI ĐẶT ỨNG DỤNG MINH HỌA 18](#_Toc10522)

[3.1. Chức năng danh mục 18](#_Toc24081)

[3.2. Chức năng thống kê 21](#_Toc23416)

[3.3. Chức năng tìm kiếm 21](#_Toc28042)

[3.4. Chức năng nghiệp vụ 21](#_Toc15091)

[CHƯƠNG 4. TỔNG KẾT 23](#_Toc8331)

[4.1. Ưu điểm 23](#_Toc15413)

[4.2. Khuyết điểm 23](#_Toc8722)

[4.3. Hướng phát triển 23](#_Toc11099)

**PHỤ LỤC TRA CỨU**

**ẢNH**

[Ảnh 1.1 .Mô hình ERD. 2](#_Toc21093)

[Ảnh 1.2 .Mô hình Daiaram. 2](#_Toc3413)

[Ảnh 3.1 .Giao diện sản phẩm. 19](#_Toc4438)

[Ảnh 3.2 .Giao diện hóa đơn. 19](#_Toc7204)

[Ảnh 3.3 .Giao diện chi tiết hóa đơn. 20](#_Toc18081)

[Ảnh 3.4 .Giao diện nhân viên. 21](#_Toc26803)

[Ảnh 3.5 .Giao diện thống kê. 21](#_Toc28940)

[Ảnh 3.6 .Giao diện login. 22](#_Toc30752)

[Ảnh 3.7 .Thêm sản phẩm. 22](#_Toc20437)

**BẢNG**

[Bảng 1.1 .Bảng Product 3](#_Toc3699)

[Bảng 1.2 .Bảng Category 3](#_Toc27851)

[Bảng 1.3 .Bảng Vendor 3](#_Toc20060)

[Bảng 1.4 .Bảng Account 3](#_Toc28978)

[Bảng 1.5 .Bảng UserDetails 3](#_Toc10701)

[Bảng 1.6 .Bảng Costumer 4](#_Toc26458)

[Bảng 1.7 .Bảng Gender 4](#_Toc24266)

[Bảng 1.8 .Bảng IsAdmin 4](#_Toc15237)

[Bảng 1.9 .Bảng Order 4](#_Toc27337)

[1.10 .Bảng OrderDetail 5](#_Toc32505)

# 

# GIỚI THIỆU

## Lý do chọn đề tài

## Mô tả bài toán

Quản lí cửa hàng bán đồng hồ là một phần mềm yêu cầu các tác vụ về :

Bán hàng , quản lý sản phẩm , quản lí nhân viên ,...Để các tác vụ này diễn ra thuận lợi , chúng ta cần một cơ sở dữ liệu hoàn chỉnh theo quy trình : bán hàng , báo cáo , doanh thu.

Thêm sản phẩm mới vào cửa hàng :

## Mô tả chức năng hệ thống

Quản lí cửa hàng bán đồng hồ gồm các chức năng :

**Quản lí sản phẩm :**

- Thêm sản phẩm mới vào cho cửa hàng :

- Cập nhật sản phẩm : thêm xóa sửa , thông tin , giá cả sản phẩm khi thông tin bị sai , hoặc giá cả sản phẩm phầm khi sản phẩm tồn quá lâu.

- Thêm : khi nhu cầu mua tăng và cửa hàng không còn hàng.

- Xóa :Khi cửa hàng không còn kinh doanh mặt hàng đó nữa.

**Quản lí hóa đơn**

- Tạo hóa đơn nhập

- Tạo chi tiết hóa đơn cho hóa đơn nhập

**Quản lí nhân viên**

-Thêm : Khi có nhân viên mới vào làm , tạo tài khoản để nhân viên có thể đăng nhập vào phần mềm của cửa hàng.

- Xóa : Khi nhân viên đó nghỉ việc .

- Cập nhật : Giúp nhân viên cập nhật lại thông tin của mình , hoặc admin cấp lại mật khẩu khi nhân viên đó quên mật khẩu.

**Báo cáo :**

- Thống kê : Tổng tiền của hàng bán được theo hóa đơn , được thống kê theo , ngày , tháng , hoặc năm.

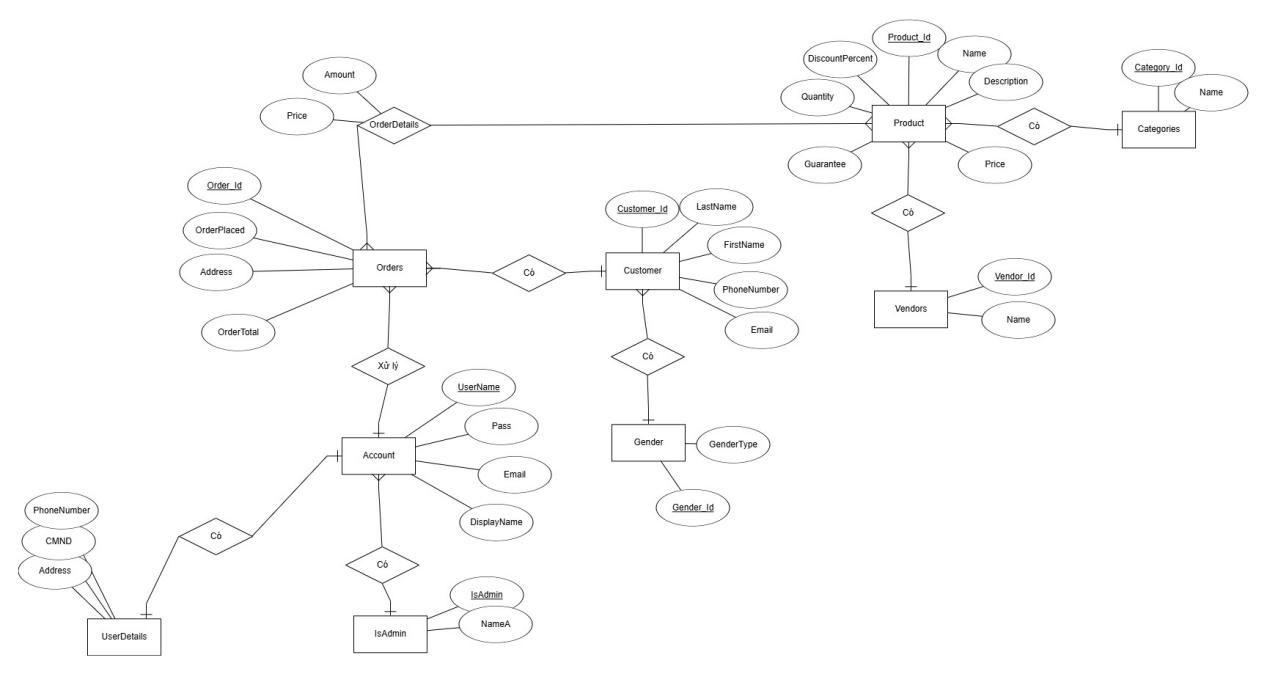
**Phân quyền :**

Quản lí cửa hàng gồm có 2 nhóm quyền : Sale và Admin

Sale : được phép thêm , xóa sửa sản phẩm , taọ nhập hóa đơn , thống kê doanh thu .

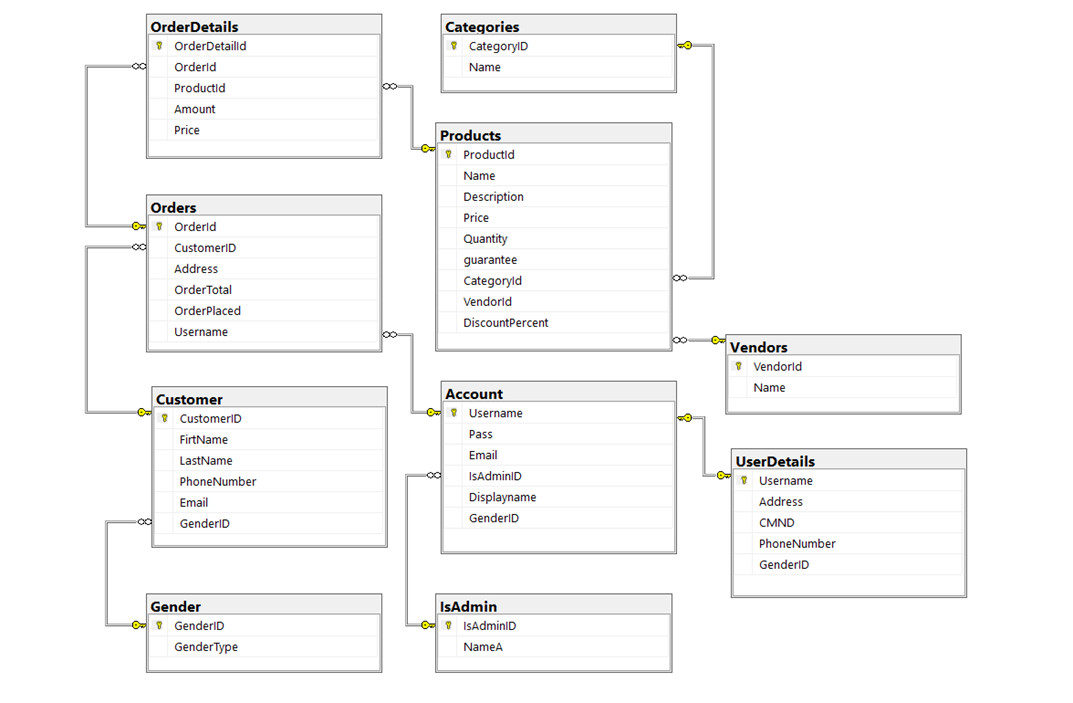
Admin : được phép toàn quyền hệ thống , thêm xóa sửa thông tin nhân viên ra khỏi cửa hàng.

## Mô hình quan hệ ERD



Ảnh 1.1.Mô hình ERD.

## Mô hình Daiaram



Ảnh 1.2.Mô hình Daiaram.

## Mô tả các bảng

Bảng liên quan tới sản phẩm

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Mô tả | Kiểu dữ liệu | Độ dài |
| ProductId | Mã sản phẩm | Varchar2 | 50 |
| Name | Tên sản phẩm | Nvarchar2 | 100 |
| Description | Mô tả | Nvarchar2 | Max |
| Price | Giá | Price |  |
| Quantity | Số lượng | Decimal |  |
| Guarantee | Bảo hành | Int |  |
| CategoryId | Mã nhà cung cấp | Varchar2 | 50 |
| VendorId | Mã loại sản phẩm | Varchar2 | 50 |
| DiscountPercent | Giảm giá | Float |  |

Bảng 1.1.Bảng Product

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Mô tả | Kiểu dữ liệu | Độ dài |
| CategoryID | Mã loại | Varchar2 | 50 |
| Name | Tên loại | Nvarchar2 | 100 |

Bảng 1.2.Bảng Category

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Mô tả | Kiểu dữ liệu | Độ dài |
| VendorID | Mã nhà cung cấp | Varchar2 | 50 |
| Name | Tên nhà cung cấp | Nvarchar2 | 50 |

Bảng 1.3.Bảng Vendor

Bản liên quan đến người dùng

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Mô tả | Kiểu dữ liệu | Độ dài |
| Username | Tên đăng nhập | Varchar2 | 50 |
| Pass | Mật khẩu | Int | 100 |
| Email | Email | Varchar2 | 100 |
| IsAdmin | Quyền | Varchar2 | 8 |
| Displayname | Tên hiển thị | Nvarchar2 | 50 |
| GenderID | Mã giới tính | Varchar2 | 2 |

Bảng 1.4.Bảng Account

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Mô tả | Kiểu dữ liệu | Độ dài |
| Username | Tên đăng nhập | Varchar2 | 50 |
| Address | Địa chỉ | Int | 100 |
| CMND | CMND | Varchar2 | 12 |
| Phone | Số điện thoại | Char | 11 |

Bảng 1.5.Bảng UserDetails

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Mô tả | Kiểu dữ liệu | Độ dài |
| CustomerID | Tên đăng nhập | Varchar2 | 50 |
| FirtName | Địa chỉ | Int | 100 |
| LastName | CMND | Varchar2 | 12 |
| PhoneNumber | Số điện thoại | Varchar2 | 11 |
| Email | Email | Varchar2 | 50 |
| GenderID | Mã giới tính | Int |  |

Bảng 1.6.Bảng Costumer

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Mô tả | Kiểu dữ liệu | Độ dài |
| GenderID | Mã giới tính | Int |  |
| GenderType | Giới tính | Nvarchar2 | 5 |

Bảng 1.7.Bảng Gender

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Mô tả | Kiểu dữ liệu | Độ dài |
| IsAdmin | Mã quyền | Varchar2 | 8 |
| NameA | Tên nhóm quyền | NVarchar2 | 100 |

Bảng 1.8.Bảng IsAdmin

Bảng liên quan đến hóa đơn

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Mô tả | Kiểu dữ liệu | Độ dài |
| OrderID | Mã hóa đơn | Varchar2 | 50 |
| Custom**Account:**  erID | Mã khách hàng | Varchar2 | 20 |
| Address | Địa chỉ lập hóa đơn | Nvarchar2 | 100 |
| OrderTotal | Tổng hóa đơn | Float |  |
| OrderPlace | Thời gian lập | Date |  |
| Usename | Tên người lập | Varchar2 | 50 |

Bảng 1.9.Bảng Order

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Mô tả | Kiểu dữ liệu | Độ dài |
| OrderDetaild | Mã chi tiết hóa đơn | Varchar2 | 50 |
| OrderId | Mã hóa đơn | Varchar2 | 50 |
| ProductId | Mã sản phẩm | Varchar2 | 50 |
| Amount | Số lượng | Float |  |
| Price | Giá của sản phẩm | Date |  |

1.10.Bảng OrderDetail

## Các thực thể

**Account:** Primary key: Username.

***Foreign key***: IsAdminID references IsAdmin.

**UserDetails**: Primary key: Username.

***Foreign key***: Username references Account.

**IsAdmin**: Primary key: IsAdminID.

**Categories**: Primary key: CategoryID.

**Vendors**: Primary key: VendorId.

**Orders**: Primary key: OrderId.

***Foreign keys***: Username references Account, CustomerID references Customer.

**Customer**:Primary key: CustomerID.

***Foreign key***: GenderID references Gender.

**Gender**: Primary key: GenderID.

**OrderDetails**: Primary key: OrderDetailId.

***Foreign keys***: OrderId references Orders, ProductId references Products.

**Products**: Primary key: ProductId.

***Foreign keys***: CategoryId references Categories, VendorId references Vendors.

## Mối quan hệ

Account và IsAdmin (1)

Orders và Account (N:1)

Orders và Customer (N:1)

Customer và Gender (1:1)

OrderDetails và Orders (N:1)

OrderDetails và Products (N:1)

Products và Categories (N:1)

Products và Vendors (N:1)

Account và UserDetails (1:1)

# THIẾT KẾ VÀ CÀI ĐẶT CƠ SỞ DỮ LIỆU

## Cài đặt các bảng

A)Bảng Account

|  |
| --- |
| CREATE TABLE Account  (  Username VARCHAR2(50) NOT NULL,  Pass VARCHAR2(50),  Email varchar2(100),  IsAdminID varchar2(8),  Displayname NVarchar2(50),  CONSTRAINT pk\_tk PRIMARY KEY (Username)  ); |

1. Bảng UserDetails

|  |
| --- |
| CREATE TABLE UserDetails  (  Username VARCHAR(50) NOT NULL PRIMARY KEY,  Address NVARCHAR(100),  CMND VARCHAR(12),  PhoneNumber VARCHAR(11),  ); |

1. Bảng IsAdmin

|  |
| --- |
| CREATE TABLE IsAdmin  (  IsAdminID varchar(8) not null,  NameA NVarChar(100),  Constraint pk\_Quyen primary key (IsAdminID) |

1. Bảng Categories

|  |
| --- |
| CREATE TABLE Categories  (  CategoryID varchar(50) not null,  Name nvarchar(100),  Constraint pk\_cat primary key (CategoryID)  ) |

1. Bảng Vendors

|  |
| --- |
| CREATE TABLE Vendors  (  VendorId varchar(50) not null,  Name nvarchar(50),  Constraint pk\_Vendors primary key (VendorId)) |

1. Bảng Orders

|  |
| --- |
| CREATE TABLE Orders  (  OrderId varchar(50) not null,  CustomerID nvarchar(20),  Address nvarchar(100),  OrderTotal float,  OrderPlaced nvarchar(50),  Username VARCHAR(50) ,  Constraint pk\_Orders primary key (OrderId)  ) |

alter table Orders

alter column OrderPlaced date

1. Bảng Customer

|  |
| --- |
| CREATE TABLE Customer  (  CustomerID nvarchar(20)primary key not null,  FirtName nvarchar(20),  LastName nvarchar(50),  PhoneNumber varchar(50),  Email varchar(50),  GenderID INT,  ) |

1. Bảng Gender

|  |
| --- |
| CREATE TABLE Gender  (  GenderID INT primary key not null ,  GenderType NVARCHAR(5),  ) |

1. Bảng OrderDetails

|  |
| --- |
| CREATE TABLE OrderDetails  (  OrderDetailId varchar(50) not null,  OrderId varchar(50),  ProductId varchar(50),  Amount int ,  Price float ,  Constraint pk\_OrderDetails primary key (OrderDetailId)  ) |

1. Bảng Products

|  |
| --- |
| CREATE TABLE Products  (  ProductId varchar(50) not null,  Name nvarchar(100),  Description nvarchar(100),  Price float,  Quantity int ,  guarantee varchar(50),  CategoryId varchar(50),  VendorId varchar(50),  DiscountPercent float,  Constraint pk\_Products primary key (ProductId)  ) |

## Câu lệnh

**Insert**

|  |  |
| --- | --- |
| **1.Thêm quyền vào bảng IsAdmin** | |
| **Code** | Insert into IsAdmin VALUES  ('ad' , 'Admin'), |
| **Kết quả** |  |

|  |  |
| --- | --- |
| **2.Thêm nhân viên vào bảng Account** | |
| **Code** | Insert into Account VALUES  ('anhminh' ,'nguyenhai1231', 'anhmin123@gmail.com' , 'sl' , N'Anh Minh' , 1), |
| **Kết quả** |  |

|  |  |
| --- | --- |
| **3.Thêm thông tin bảng Gender** | |
| **Code** | INSERT INTO Gender VALUES  (1,N'NAM'),  (2,N'NỮ') |
| **Kết quả** |  |

|  |  |
| --- | --- |
| **4.Thêm nhân viên cho bảng Customer** | |
| **Code** | Insert into Customer VALUES  ('KH01',N'Anh',N'Nguyễn Hải','0903926730',N'haianh34@gmail.com',1), |
| **Kết quả** |  |

|  |  |
| --- | --- |
| **5.Thêm quyền vào bảng Categories** | |
| **Code** | INSERT INTO Categories  VALUES ('CAT01',N'Đồng Hồ Cơ'), |
| **Kết quả** |  |

|  |  |
| --- | --- |
| **6.Thêm thông tin cho bảng Vendor** | |
| **Code** | INSERT INTO Vendors  VALUES ('VEN01',N'Thế Giới Di Động'), |
| **Kết quả** |  |

|  |  |
| --- | --- |
| **7.Thêm thông tin cho bảng Products** | |
| **Code** | INSERT INTO Products  VALUES ('PRO01',N'Đồng hồ thông minh Microwear GT4 Max',N'Sở hữu thiết kế thời thượng, sang trọng. Cùng với các thuật toán thông minh giúp việc hoạt động hiệu quả, cung cấp sức mạnh nâng cao khả năng tính toán…',950000,10,N'2 Năm','CAT03','VEN04',20), |
| **Kết quả** |  |

## Truy vấn

|  |
| --- |
| 1)Truy vấn tính tổng doanh số bán hàng cho từng danh mục sản phẩm |
| SELECT  P.CategoryId,  C.Name AS CategoryName,  SUM(OD.Amount \* OD.Price) AS TotalSales  FROM  OrderDetails OD  JOIN  Products P ON OD.ProductId = P.ProductId  JOIN  Categories C ON P.CategoryId = C.CategoryId  WHERE  OD.OrderId IS NOT NULL  GROUP BY  P.CategoryId, C.Name  HAVING  SUM(OD.Amount \* OD.Price) > 1000  ORDER BY  TotalSales DESC; |

|  |
| --- |
| 2)Tính sổ lượng đơn bán của nhân viên từ ngày “bắt đầu” -> “ngày kết thúc” |
| SELECT  A.Username,  A.Displayname,  COUNT(O.OrderId) AS NumberOfOrders  FROM  Orders O  JOIN  Account A ON O.Username = A.Username  WHERE  O.OrderPlaced BETWEEN TO\_DATE('2023-01-01', 'DD-MM-YYYY') AND TO\_DATE('2023-12-31', 'DD-MM-YYYY')  GROUP BY  A.Username, A.Displayname  HAVING  COUNT(O.OrderId) > 5  ORDER BY  NumberOfOrders DESC; |

|  |
| --- |
| 3)Truy vấn tính tổng doanh số bán hàng cho từng danh mục sản phẩm |
| SELECT a.Username, o.OrderId, o.OrderTotal  FROM Account a  JOIN Orders o ON a.Username = o.Username  WHERE a.IsAdminID = 'ad'  ORDER BY o.OrderTotal DESC; |

|  |
| --- |
| 1. Truy vấn danh sách các Hóa đơn đã được xử lý bởi các quản trị viên, sắp xếp theo tổng giá trị đơn hàng từ cao đến thấp |
| SELECT a.Username, o.OrderId, o.OrderTotal  FROM Account a  JOIN Orders o ON a.Username = o.Username  WHERE a.IsAdminID = 'ad'  ORDER BY o.OrderTotal DESC; |
| 1. Truy vấn danh danh sách các nhân viên đã xử lý nhiều hơn 5 đơn hàng và trung bình Tổng tiền được sắp xếp giảm dần |
| SELECT A.Username, AVG(O.OrderTotal) AS AverageOrderTotal  FROM Orders O, Account A  WHERE O.Username = A.Username  GROUP BY A.Username  HAVING COUNT(O.OrderId) > 5  ORDER BY AverageOrderTotal DESC; |

|  |
| --- |
| 1. Truy vấn danh sách những nhân viên là 'Admin' đã xử lý hơn 10 đơn hàng và tổng số tiền số đơn hàng thanh toán lớn hơn 1000 được sắp xếp giảm dần |
| SELECT A.Username, SUM(O.OrderTotal) AS TotalOrderValue, COUNT(O.OrderId) AS NumberOfOrders  FROM Orders O, Account A  WHERE O.Username = A.Username AND A.IsAdminID = 'ADMIN'  GROUP BY A.Username  HAVING COUNT(O.OrderId) > 10 AND SUM(O.OrderTotal) > 1000  ORDER BY TotalOrderValue DESC; |

|  |
| --- |
| 1. Truy vấn lấy ra danh sách các đơn hàng có tổng giá trị lớn hơn 50,000 và sắp xếp theo tổng giá trị giảm dần. |
| SELECT o.OrderId, c.FirtName, c.LastName, o.OrderTotal  FROM Orders o  JOIN Customer c ON o.CustomerID = c.CustomerID  WHERE o.OrderTotal > 50000  ORDER BY o.OrderTotal DESC; |

|  |
| --- |
| 1. Truy vấn đếm số lượng sản phẩm và tính giá trung bình của các sản phẩm theo từng danh mục, chỉ lấy ra các danh mục có số lượng sản phẩm tồn kho lớn hơn 10 và có ít nhất 5 sản phẩm đã được bán. |
| SELECT p.CategoryId, COUNT(\*) AS TotalProducts, AVG(p.Price) AS AveragePrice  FROM Products p  JOIN OrderDetails od ON p.ProductId = od.ProductId  WHERE p.Quantity > 10  GROUP BY p.CategoryId  HAVING COUNT(\*) > 5  ORDER BY TotalProducts DESC; |

|  |
| --- |
| 1. Truy vấn tính tổng số lượng sản phẩm mà mỗi khách hàng đã mua, chỉ lấy ra những khách hàng có tổng giá trị mua hàng lớn hơn 100,000 |
| SELECT c.CustomerID, c.FirtName, c.LastName, SUM(od.Amount) AS TotalAmount  FROM Customer c  JOIN OrderDetails od ON c.CustomerID = od.OrderId  GROUP BY c.CustomerID, c.FirtName, c.LastName  HAVING SUM(od.Price \* od.Amount) > 100000  ORDER BY TotalAmount DESC; |

## Hàm

|  |
| --- |
| 1)Sản phẩm bán chạy nhất |
| CREATE OR REPLACE FUNCTION f\_mathang\_banchaynhat  RETURN NVARCHAR2  IS  TENMH NVARCHAR2(50);  BEGIN  BEGIN  SELECT pro.Name INTO TENMH  FROM OrderDetails ord  INNER JOIN Products pro ON pro.ProductId = ord.ProductId  GROUP BY pro.Name  ORDER BY SUM(ord.Amount) DESC  FETCH FIRST 1 ROW ONLY;  EXCEPTION  WHEN NO\_DATA\_FOUND THEN  TENMH := 'No Data';  END;    RETURN TENMH;  END;  /  --Khởi chạy câu lệnh  DECLARE  v\_result NVARCHAR2(50);  BEGIN  v\_result := f\_mathang\_banchaynhat();  DBMS\_OUTPUT.PUT\_LINE('Sản phẩm bán chạy nhất là: ' || v\_result);  END;  / |

|  |
| --- |
| 1)Sản phẩm bán chạy nhất |
| CREATE OR REPLACE FUNCTION f\_mathang\_banchaynhat  RETURN NVARCHAR2  IS  TENMH NVARCHAR2(50);  BEGIN  BEGIN  SELECT pro.Name INTO TENMH  FROM OrderDetails ord  INNER JOIN Products pro ON pro.ProductId = ord.ProductId  GROUP BY pro.Name  ORDER BY SUM(ord.Amount) DESC  FETCH FIRST 1 ROW ONLY;  EXCEPTION  WHEN NO\_DATA\_FOUND THEN  TENMH := 'No Data';  END;    RETURN TENMH;  END;  /  --Khởi chạy câu lệnh  DECLARE  v\_result NVARCHAR2(50);  BEGIN  v\_result := f\_mathang\_banchaynhat();  DBMS\_OUTPUT.PUT\_LINE('Sản phẩm bán chạy nhất là: ' || v\_result);  END;  / |

## Thủ tục

|  |
| --- |
| 1)Thêm nhân viên bằng thủ tục |
| CREATE OR REPLACE PROCEDURE InsertAccount(  p\_Username IN VARCHAR2,  p\_Pass IN VARCHAR2,  p\_Email IN VARCHAR2,  p\_IsAdminID IN VARCHAR2,  p\_Displayname IN NVARCHAR2  ) AS  v\_count INTEGER;  BEGIN  -- Validate Username  IF p\_Username IS NULL THEN  RAISE\_APPLICATION\_ERROR(-20001, 'Username cannot be null.');  END IF;  -- Validate Password  IF p\_Pass IS NULL THEN  RAISE\_APPLICATION\_ERROR(-20002, 'Password cannot be null.');  END IF;  -- Validate Email  IF p\_Email IS NULL THEN  RAISE\_APPLICATION\_ERROR(-20003, 'Email cannot be null.');  ELSIF NOT REGEXP\_LIKE(p\_Email, '^[A-Za-z0-9.\_%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}$') THEN  RAISE\_APPLICATION\_ERROR(-20004, 'Invalid email format.');  END IF;  -- Validate IsAdminID  IF p\_IsAdminID IS NOT NULL THEN  SELECT COUNT(\*) INTO v\_count FROM IsAdmin WHERE IsAdminID = p\_IsAdminID;  IF v\_count = 0 THEN  RAISE\_APPLICATION\_ERROR(-20005, 'IsAdminID does not exist.');  END IF;  END IF;  -- Insert Data into Account Table  INSERT INTO Account (Username, Pass, Email, IsAdminID, Displayname)  VALUES (p\_Username, p\_Pass, p\_Email, p\_IsAdminID, p\_Displayname);  COMMIT;  EXCEPTION  WHEN OTHERS THEN  ROLLBACK;  RAISE\_APPLICATION\_ERROR(-20006, 'An error occurred during insert: ' || SQLERRM);  END InsertAccount; |

**Kết quả test**

|  |  |
| --- | --- |
| Code | --Thêm thông tin  BEGIN  InsertAccount('john\_doe', 'securepassword', 'john.doe@gmail.com', 'ad', 'John Doe');  END; |
| Kết quả |  |

|  |
| --- |
| 2)Thủ tục hiện thông tin vừa thêm vào |
| CREATE OR REPLACE PROCEDURE DisplayAccountData AS  CURSOR account\_cursor IS  SELECT Username, Pass, Email, IsAdminID, Displayname FROM Account;  BEGIN  FOR account\_record IN account\_cursor LOOP  DBMS\_OUTPUT.PUT\_LINE('Username: ' || account\_record.Username);  DBMS\_OUTPUT.PUT\_LINE('Password: ' || account\_record.Pass);  DBMS\_OUTPUT.PUT\_LINE('Email: ' || account\_record.Email);  DBMS\_OUTPUT.PUT\_LINE('IsAdminID: ' || account\_record.IsAdminID);  DBMS\_OUTPUT.PUT\_LINE('Displayname: ' || account\_record.Displayname);  DBMS\_OUTPUT.PUT\_LINE('--------------------------');  END LOOP;  END DisplayAccountData;  / |

|  |
| --- |
| 3)Thủ tục thêm một khách hàng vào csdl |
| CREATE OR REPLACE PROCEDURE themKhachHang(  p\_CustomerID IN Customer.CustomerID%TYPE,  p\_FirtName IN Customer.FirtName%TYPE,  p\_LastName IN Customer.LastName%TYPE,  p\_PhoneNumber IN Customer.PhoneNumber%TYPE,  p\_Email IN Customer.Email%TYPE,  p\_GenderID IN Customer.GenderID%TYPE  ) AS  v\_count INT;  BEGIN  -- Ki?m tra xem các tham s? có h?p l? hay không  IF p\_CustomerID IS NULL THEN  RAISE\_APPLICATION\_ERROR(-20002, 'CustomerID không ???c ?? tr?ng');  END IF;  IF p\_FirtName IS NULL THEN  RAISE\_APPLICATION\_ERROR(-20003, 'FirtName không ???c ?? tr?ng');  END IF;  IF p\_LastName IS NULL THEN  RAISE\_APPLICATION\_ERROR(-20004, 'LastName không ???c ?? tr?ng');  END IF;  IF p\_PhoneNumber IS NULL THEN  RAISE\_APPLICATION\_ERROR(-20005, 'PhoneNumber không ???c ?? tr?ng');  ELSE  IF LENGTH(p\_PhoneNumber) != 10 THEN  RAISE\_APPLICATION\_ERROR(-20008, 'PhoneNumber ph?i có ?úng 10 s?');  END IF;  END IF;  IF p\_Email IS NULL THEN  RAISE\_APPLICATION\_ERROR(-20006, 'Email không ???c ?? tr?ng');  ELSE  IF NOT REGEXP\_LIKE(p\_Email, '^[a-zA-Z0-9.\_%-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,4}$') THEN  RAISE\_APPLICATION\_ERROR(-20009, 'Email không h?p l?');  END IF;  END IF;  IF p\_GenderID IS NULL THEN  RAISE\_APPLICATION\_ERROR(-20007, 'GenderID không ???c ?? tr?ng');  END IF;  -- Ki?m tra xem GenderID có t?n t?i trong b?ng Gender hay không  SELECT COUNT(\*) INTO v\_count FROM Gender WHERE GenderID = p\_GenderID;  IF v\_count = 0 THEN  RAISE\_APPLICATION\_ERROR(-20001, 'GenderID không t?n t?i trong b?ng Gender');  END IF;  -- Thêm d? li?u vào b?ng Customer  INSERT INTO Customer(CustomerID, FirtName, LastName, PhoneNumber, Email, GenderID)  VALUES (p\_CustomerID, p\_FirtName, p\_LastName, p\_PhoneNumber, p\_Email, p\_GenderID);  COMMIT;  EXCEPTION  WHEN OTHERS THEN  ROLLBACK;  RAISE;  END themKhachHang; |

|  |  |
| --- | --- |
| Code | BEGIN  themKhachHang(  p\_CustomerID => 'CUST456',  p\_FirtName => 'Nguyen',  p\_LastName => 'Van A',  p\_PhoneNumber => '0123456789',  p\_Email => 'nguyenvana@example.com',  p\_GenderID => 1  );  END; |

|  |
| --- |
| 4)Thủ tục hiện thông tin vừa thêm vào |
| CREATE OR REPLACE PROCEDURE dsKhachHang AS  CURSOR c\_customers IS  SELECT \* FROM Customer;  v\_customer c\_customers%ROWTYPE;  BEGIN  OPEN c\_customers;  LOOP  FETCH c\_customers INTO v\_customer;  EXIT WHEN c\_customers%NOTFOUND;  DBMS\_OUTPUT.PUT\_LINE('CustomerID: ' || v\_customer.CustomerID);  DBMS\_OUTPUT.PUT\_LINE('FirtName: ' || v\_customer.FirtName);  DBMS\_OUTPUT.PUT\_LINE('LastName: ' || v\_customer.LastName);  DBMS\_OUTPUT.PUT\_LINE('PhoneNumber: ' || v\_customer.PhoneNumber);  DBMS\_OUTPUT.PUT\_LINE('Email: ' || v\_customer.Email);  DBMS\_OUTPUT.PUT\_LINE('GenderID: ' || v\_customer.GenderID);  DBMS\_OUTPUT.PUT\_LINE('-------------------------');  END LOOP;  CLOSE c\_customers;  EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('Đã xảy ra lỗi: ' || SQLERRM);  END dsKhachHang; |

## Trigger

|  |
| --- |
| 1)Trigger Cập nhật tổng tiền của hóa đơn theo chi tiết hóa đơn. |
| CREATE OR REPLACE TRIGGER updateOrderTotal  AFTER INSERT  ON orderDetails  DECLARE  BEGIN  -- Cập nhật OrderTotal cho tất cả các OrderId tương ứng  FOR order\_rec IN (  SELECT OrderId, SUM(amount \* price) AS total  FROM orderDetails  GROUP BY OrderId  )  LOOP  UPDATE Orders  SET OrderTotal = order\_rec.total  WHERE OrderId = order\_rec.OrderId;  END LOOP;  EXCEPTION  WHEN OTHERS THEN  NULL; -- Bỏ qua lỗi để tránh việc trigger bị vô hiệu hóa  END;  / |

Kết quả test

|  |  |
| --- | --- |
|  |  |

|  |
| --- |
| 2)Trigger giảm số lượng hàng trong kho khi thêm số lượng vào OrderDetails |
| CREATE OR REPLACE TRIGGER updateOrderTotalBeforeDelete  BEFORE DELETE ON orderDetails  FOR EACH ROW  DECLARE  v\_total\_price NUMBER;  BEGIN  -- Tính toán tổng số tiền của mặt hàng bị xóa  v\_total\_price := :OLD.amount \* :OLD.price;  -- Cập nhật tổng tiền của đơn hàng tương ứng  UPDATE Orders  SET OrderTotal = OrderTotal - v\_total\_price  WHERE OrderId = :OLD.OrderId;  END; |

|  |
| --- |
| 3)Trigger cập nhật mô tả “Hết sản phẩm” khi số lượng về 0 |
| CREATE TRIGGER trg\_UpdateProductDescription  ON Products  AFTER UPDATE  AS  BEGIN  IF UPDATE(Quantity)  BEGIN  UPDATE Products  SET Description = N'Hết sản phẩm'  FROM inserted i  WHERE Products.ProductId = i.ProductId  AND i.Quantity = 0  END  END |

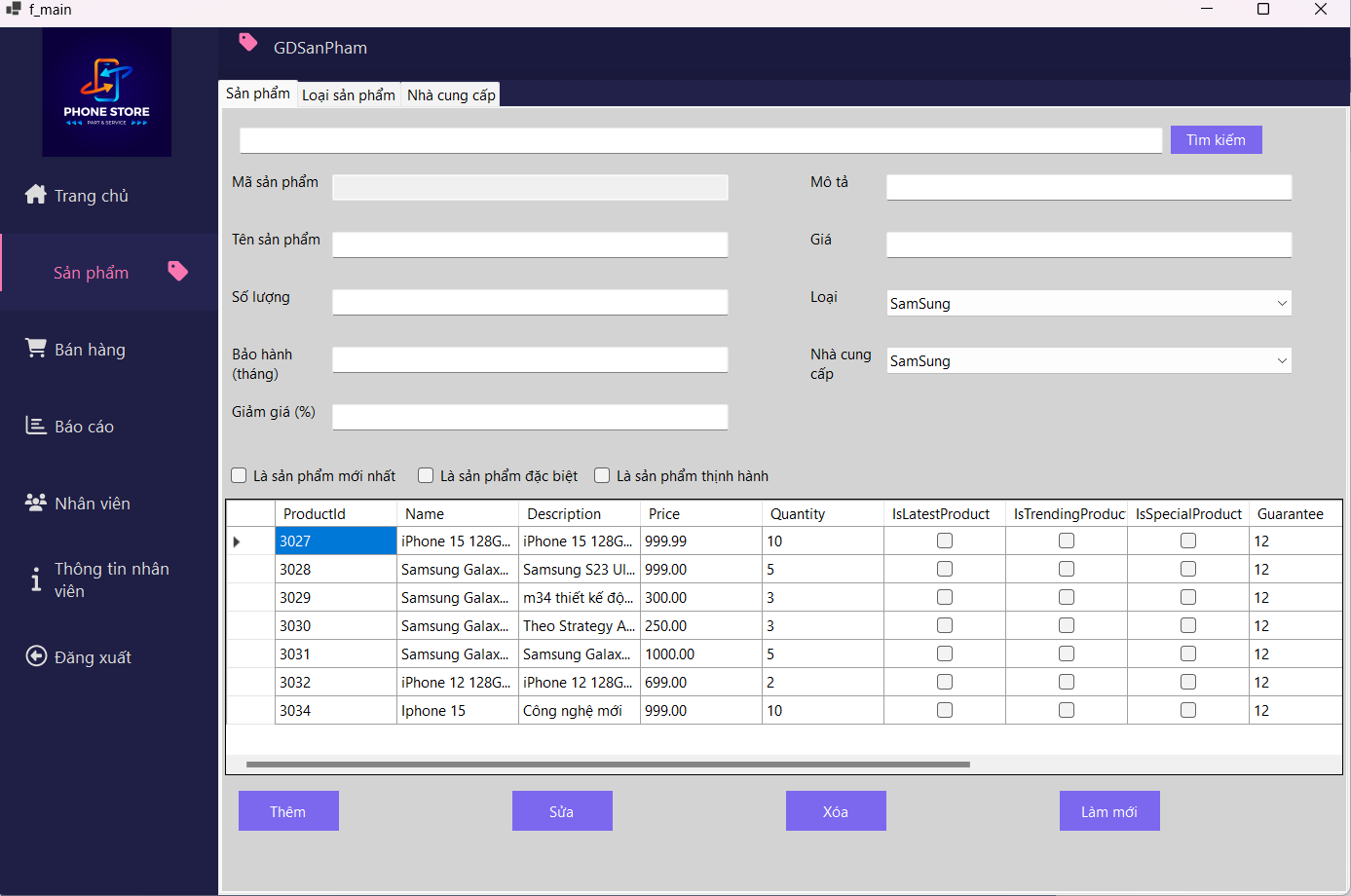
|  |
| --- |
| 4)Khi tạo account , tạo thông tin cho userDetails |
| CREATE TRIGGER set\_undefined\_address  ON Account  AFTER INSERT  AS  BEGIN  SET NOCOUNT ON;  -- Insert vào UserDetails cho mỗi dòng được chèn vào bảng Account  INSERT INTO UserDetails (Username, Address)  SELECT ins.Username, 'Không xác định'  FROM inserted ins;  END; |

# CÀI ĐẶT ỨNG DỤNG MINH HỌA

## Chức năng danh mục

**Danh mục sản phẩm**

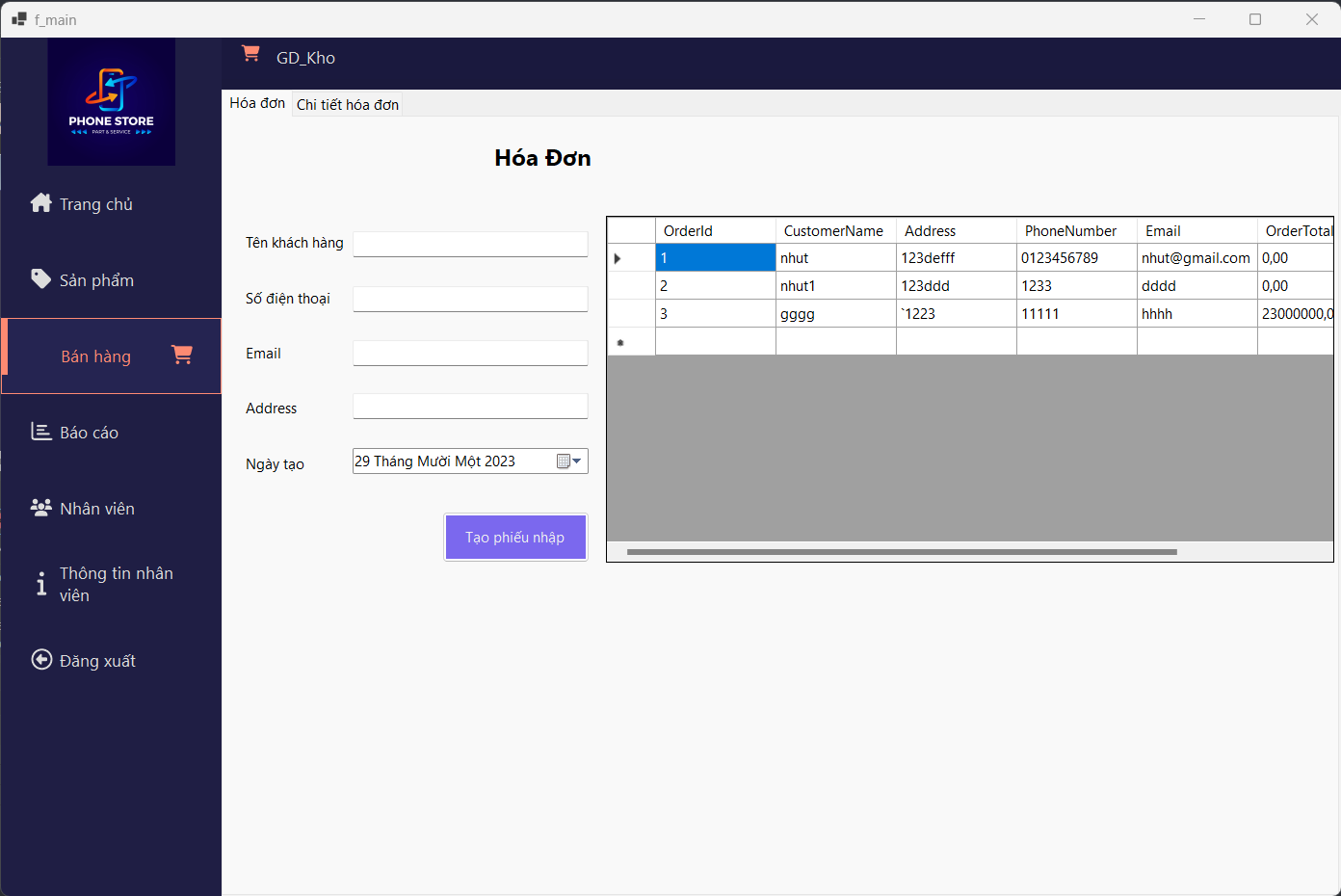
Hiển thị đầy đủ các thông tin về mã sản phẩm , số lượng , giá ,...của mỗi sản phẩm



Ảnh 3.1.Giao diện sản phẩm.

**Danh mục hóa đơn**

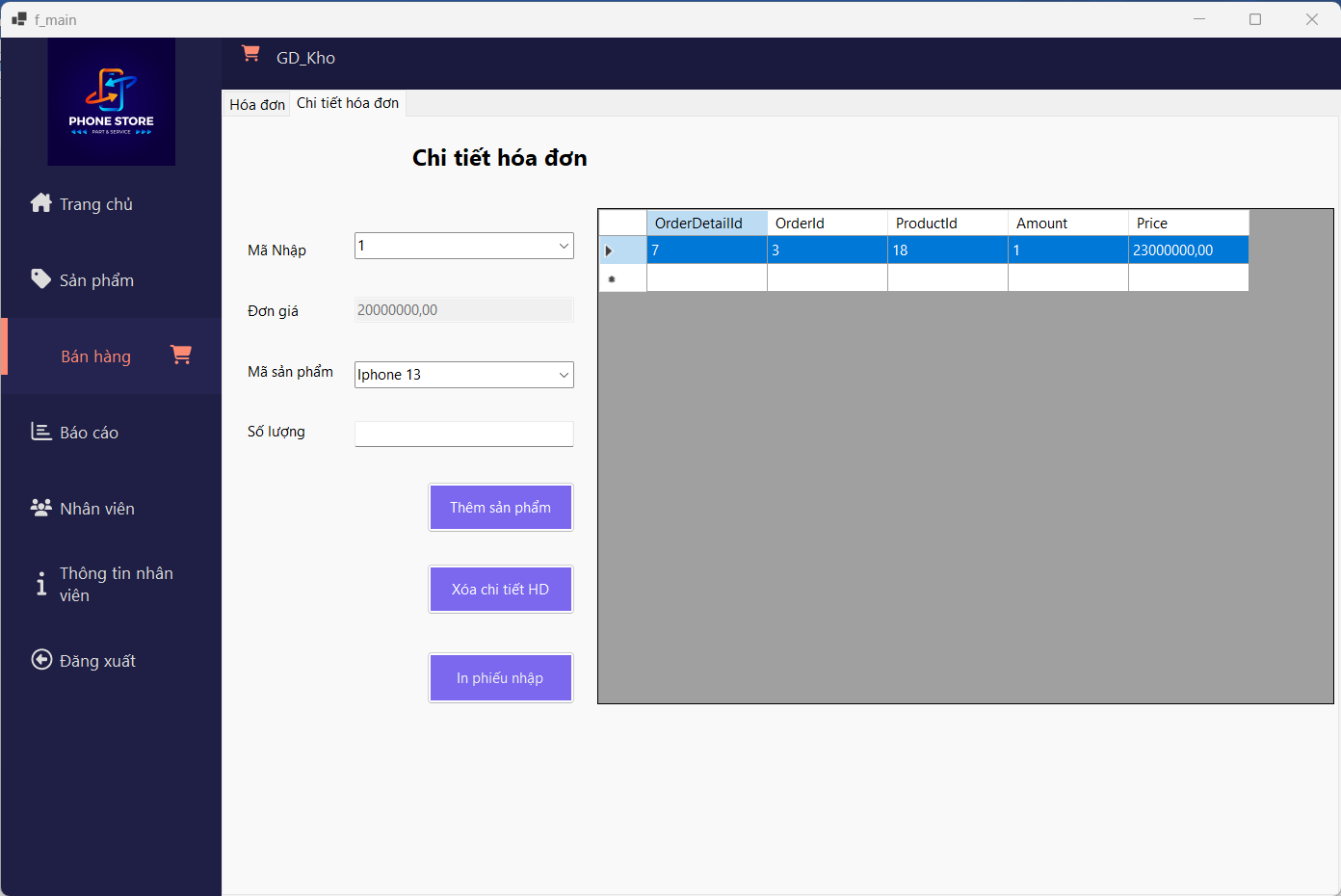
- Hiển thị các thông tin cần thiết trong hóa đơn , mã hóa đơn , địa chỉ , tên nhân viên lập, tên khách hàng , ngày lập và địa chỉ lập



Ảnh 3.2.Giao diện hóa đơn.

**Danh mục chi tiết hóa đơn**

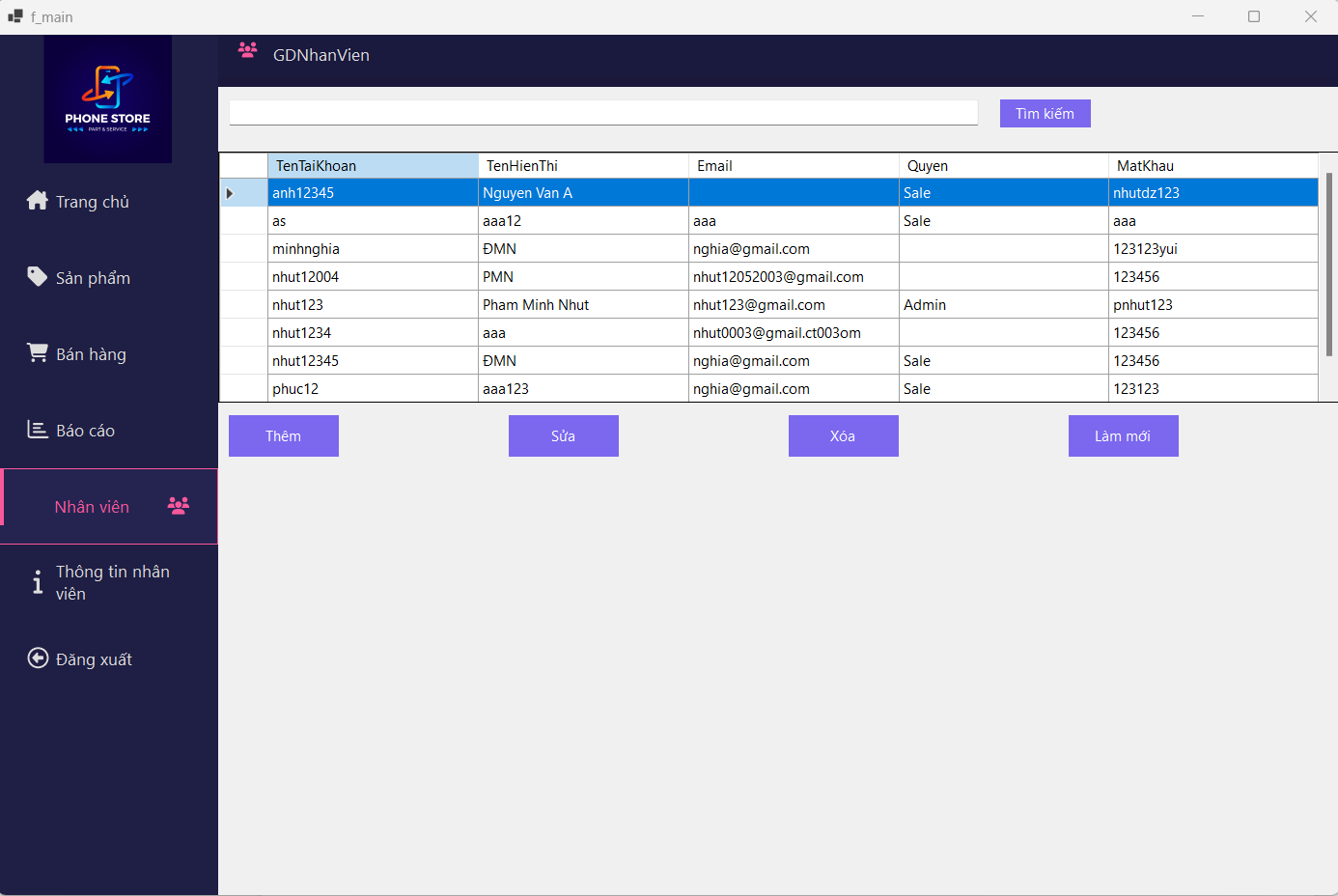
- Hiển thị thông tin mã chi tiết hóa đơn , mã hóa đơn tương ứng với hóa đơn đó , mã sản phẩm bán , số lượng , giá tiền của sản phẩm.



Ảnh 3.3.Giao diện chi tiết hóa đơn.

**Danh mục nhân viên**

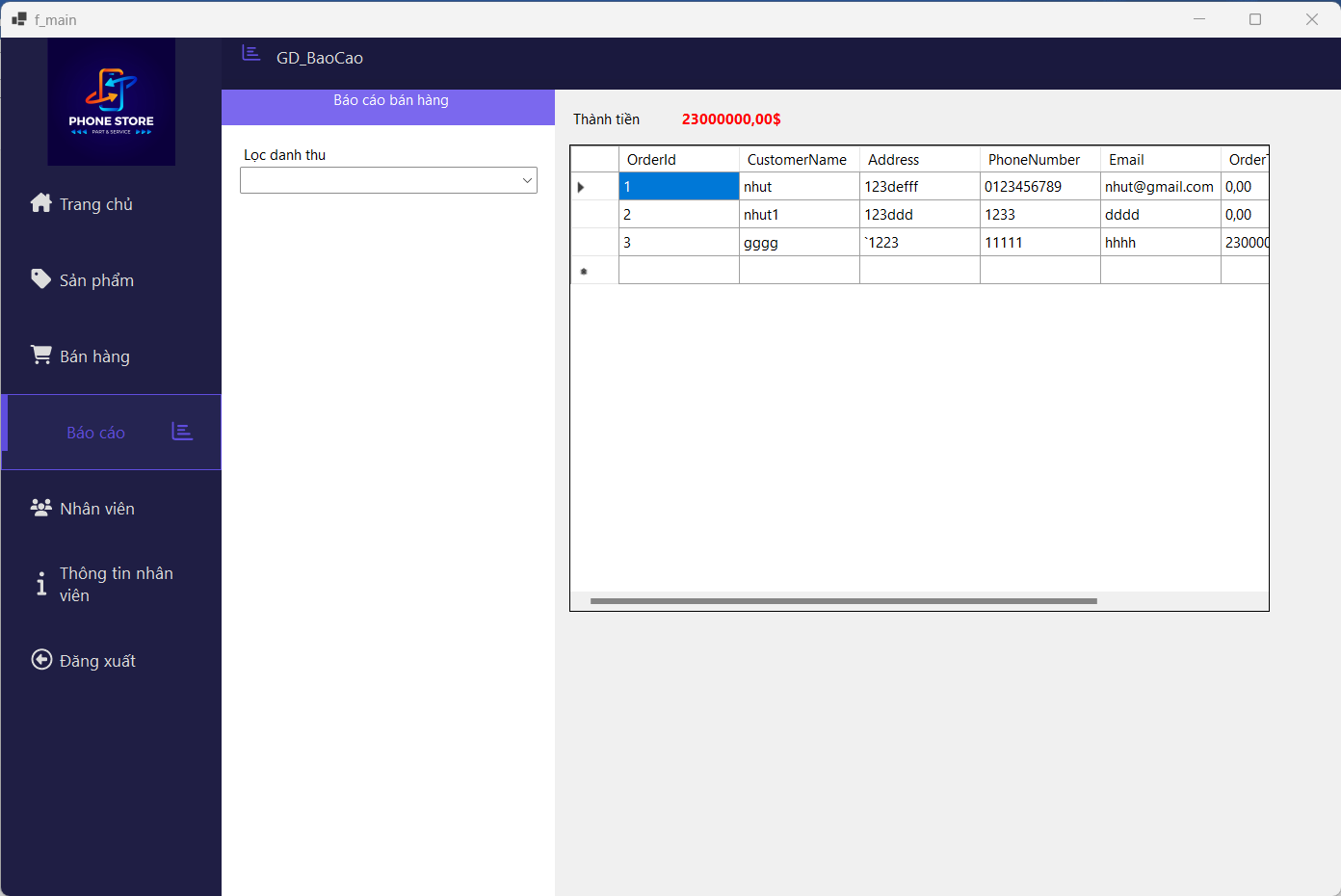
- Hiển thị các thông tin của nhân viên trong cửa hàng , tên hiển thị , số điện thoại , email , nhóm quyền của nhân viên , và mật khẩu , cho việc cấp lại tài khoản



Ảnh 3.4.Giao diện nhân viên.

## Chức năng thống kê

Ở giao diện này người dùng có thể lựa chọn thống kế tiền bán được của cửa hàng theo ngày , tháng , nằm bằng việc lựa chọn thông tin tương ứng trên combobox , sau khi lựa chọn , danh sách hóa đơn sẽ hiện thị ra và tổng tiền bán được theo thời gian lựa chọn.



Ảnh 3.5.Giao diện thống kê.

## Chức năng tìm kiếm

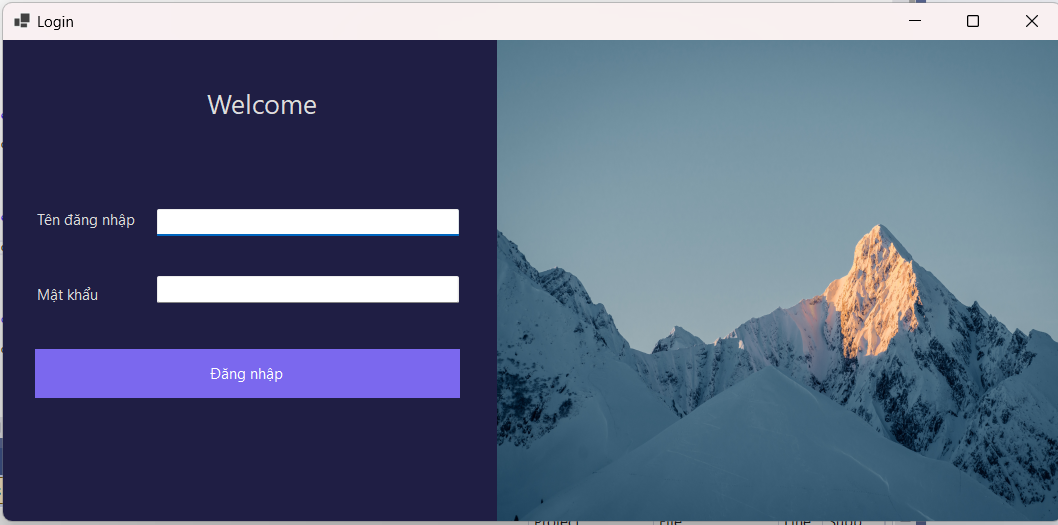
Ở chúc năng này , phần mềm hỗ trợ người dùng tìm kiếm sản phẩm theo tên .

- Tìm kiếm nhân viên , khách hàng . Hỗ trợ các tác vụ luên quan đến tìm kiếm.

## Chức năng nghiệp vụ

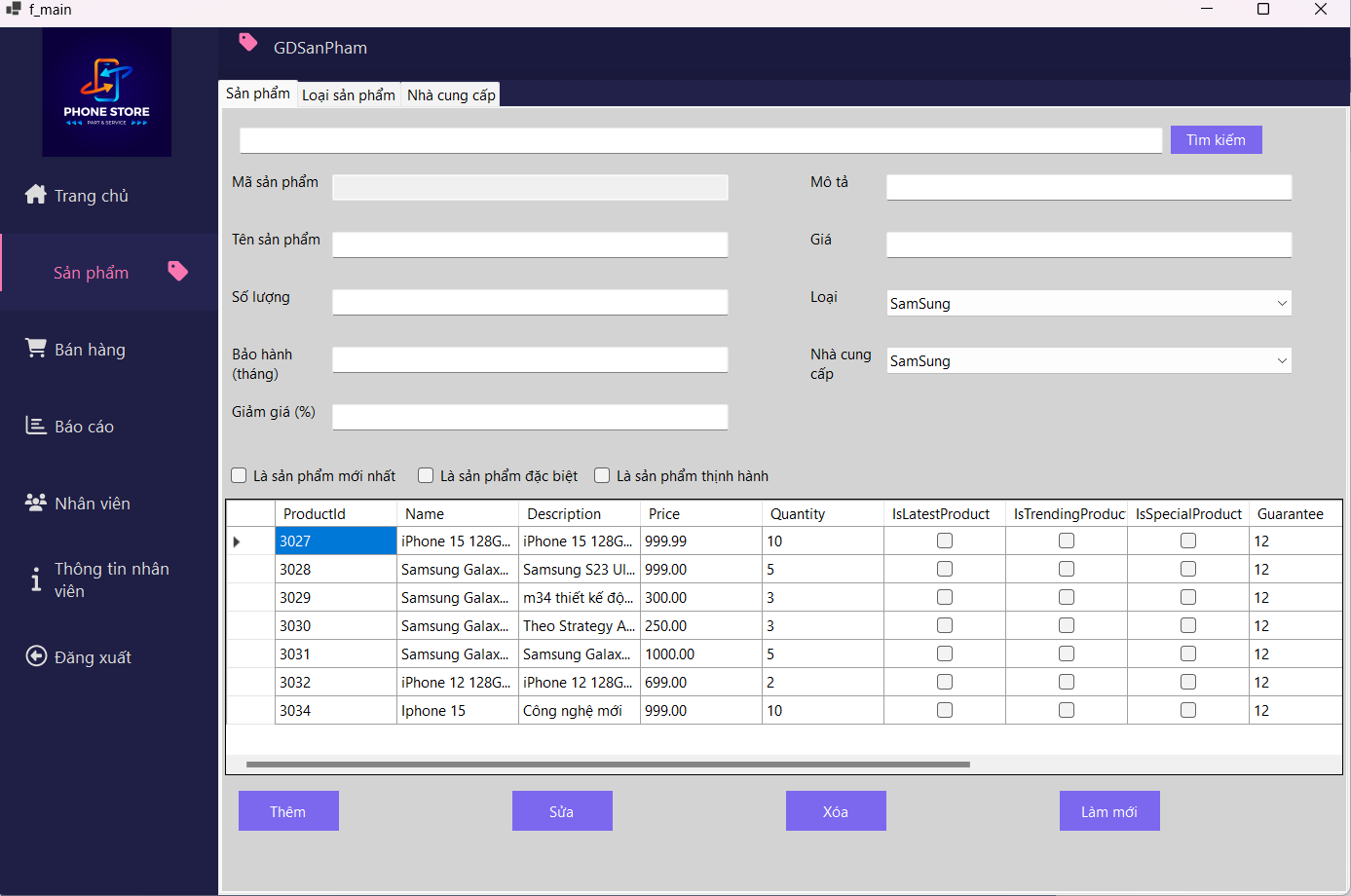
**Login**

Nhân viên nhập mật khẩu do chủ cửa hàng cung cấp.



Ảnh 3.6.Giao diện login.

**Thêm , xóa , sửa sản phẩm**



Ảnh 3.7.Thêm sản phẩm.

**Quản lí nhân viên**

Ở tác vụ này chỉ có người có quyền admin mới được phép truy cập .

Người dùng có thể thực hiện , thêm xóa sửa , cấp lại mật khẩu cho nhân viên

.

# TỔNG KẾT

## Ưu điểm

- Phần mềm trực quan , dễ thao tác , quản lí.

- Dữ liệu được tự động cập nhật khi thêm xóa , sửa.

## Khuyết điểm

- Hệ thống cơ sở dữ liệu có vài chỗ còn sai sót .

- Tính bảo mật chưa cao : chưa mã hóa mật khẩu khi đẩy từ trên phần mềm xuống.

## Hướng phát triển

- Tiếp tục nghiên cứu , phát triển để hoàn thiện phần mềm , cơ sở dữ liệu hơn trong tương lai .

- Xây dựng hệ thống có tính bảo mật hơn.

- Đề ra các giải pháp về sao lưu phục hồi khi mất mát dữ liệu.