**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

**\*\*\***



Báo cáo bộ môn

**HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

**ĐỀ TÀI:**

**XÂY DỰNG WEBSITE KINH DOANH GIÀY, PHỤ KIỆN**

**Giáo viên hướng dẫn:**

TS. NGUYỄN THÀNH SƠN

**Sinh viên thực hiện:**

**NHÓM 10**

|  |  |
| --- | --- |
| Diệp Tấn Luân (Nhóm trưởng) | **MSSV: 19110055** |
| Phạm Minh Mẫn | **MSSV: 19110397** |
| Nguyễn Quang Sang | **MSSV: 19110447** |
| Lê Vũ Hảo | **MSSV: 19110359** |

***TP.HCM, tháng 11 năm 2021***

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

**Ký tên**

**MỤC LỤC**

[1. Tổng quan về phần mềm 6](#_Toc88128673)

[**1.1. Mục tiêu** 6](#_Toc88128674)

[**1.2. Phạm vi** 6](#_Toc88128675)

[**1.3. Mô tả** 6](#_Toc88128676)

[**1.4. Thông tin phần mềm** 6](#_Toc88128677)

[**1.5. Chức năng của phần mềm** 7](#_Toc88128678)

[2. Lược đồ quan hệ 7](#_Toc88128679)

[**2.1. Mô hình ERD** 7](#_Toc88128680)

[2.1.1. Mô hình ERD 7](#_Toc88128681)

[2.1.2. Lược đồ quan hệ 8](#_Toc88128682)

[**2.2. Mô tả cơ sở dữ liệu** 8](#_Toc88128683)

[**2.3. Biểu đồ quan hệ giữa các bảng (Database diagram)** 10](#_Toc88128684)

[3. Code tạo bảng, khoá chính, ràng buộc khoá ngoại 12](#_Toc88128685)

[4. Danh sách các views, stored procedure, function, triggers 14](#_Toc88128686)

[**4.1. Views** 14](#_Toc88128687)

[4.1.1. Hiển thị thông tin sản phẩm 14](#_Toc88128688)

[4.1.2. Quản lý thông tin cá nhân 14](#_Toc88128689)

[4.1.3. Thông tin các đơn hàng 14](#_Toc88128690)

[4.1.4. Quản lý người dùng 14](#_Toc88128691)

[**4.2.** **Stored procedures** 14](#_Toc88128694)

[4.2.1. Account 14](#_Toc88128695)

[4.2.2. Bill 16](#_Toc88128696)

[4.2.3. Category 18](#_Toc88128697)

[4.2.4. Contact 20](#_Toc88128698)

[4.2.5. DetailOrder 21](#_Toc88128699)

[4.2.6. Member 23](#_Toc88128700)

[4.2.7. Order 25](#_Toc88128701)

[4.2.8. Product 27](#_Toc88128702)

[4.2.9. Một số stored procedure tiện ích khác. 29](#_Toc88128703)

[**4.3.** **Functions** 30](#_Toc88128704)

[4.3.1. Table-valued functions 30](#_Toc88128705)

[4.3.2. Scalar-valued functions 32](#_Toc88128706)

[**4.4.** **Triggers** 32](#_Toc88128707)

[4.4.1. Tự động tạo Account và cấp quyền khi thêm dữ liệu trong bảng Account 32](#_Toc88128708)

[4.4.2. Người dùng tuổi nhỏ hơn 10 không thể mua hàng 33](#_Toc88128709)

[4.4.3. Giảm giá 25% cho các sản phẩm thuộc danh mục ‘socks’ 33](#_Toc88128710)

[4.4.4. Tự động thêm ngày mua trong Order 33](#_Toc88128711)

[4.4.5. Tự động cập nhật số món hàng, tổng giá, giảm giá cho người già trong Order 34](#_Toc88128712)

[4.4.6. Tự động cập nhật số món hàng, tổng giá khi thêm hoặc sửa DetailOrder 35](#_Toc88128713)

[**4.5.** **Transaction** 35](#_Toc88128719)

[4.5.1. Tự động cập nhật số món hàng, tổng giá, giảm giá cho người già trong Order 35](#_Toc88128721)

[5. Phân quyền trong phần mềm 36](#_Toc88128722)

[**5.1.** **Tạo user và roles** 36](#_Toc88128725)

[5.1.1. Tạo role 36](#_Toc88128726)

[5.1.2. Tạo user 36](#_Toc88128727)

[**5.2.** **Phân quyền đăng nhập hệ thống** 37](#_Toc88128728)

[5.2.1. Phân quyền ADMIN 37](#_Toc88128729)

[5.2.2. Phân quyền SELLER 38](#_Toc88128730)

[5.2.3. Phân quyền CUSTOMER 39](#_Toc88128731)

[**5.3.** **Bảng mô tả phân quyền** 39](#_Toc88128732)

[6. Code cài đặt ứng dụng 43](#_Toc88128733)

[***6.1.*** *Kết nối cơ sở dữu liệu* 43](#_Toc88128736)

[***6.2.*** *Tìm kiếm sản phẩm* 43](#_Toc88128737)

[***6.3.*** *Load sản phẩm và phân trang* 44](#_Toc88128738)

[***6.4.*** *Load và phân trang account trong manage account* 44](#_Toc88128739)

[***6.5.*** *Thêm sản phẩm* 45](#_Toc88128740)

[***6.6.*** *Sửa sản phẩm* 45](#_Toc88128741)

[***6.7.*** *Xóa sản phẩm* 46](#_Toc88128742)

[***6.8.*** *Load sản phẩm theo category* 46](#_Toc88128743)

[***6.9.*** *Xem chi tiết sản phẩm* 47](#_Toc88128744)

[***6.10.*** *Load tất cả danh mục* 47](#_Toc88128745)

[***6.11.*** *Load sản phẩm cuối cùng (mới nhất)* 47](#_Toc88128746)

[***6.12.*** *Sửa Account* 48](#_Toc88128747)

[***6.13.*** *Thêm Account* 48](#_Toc88128748)

[***6.14.*** *Xoá Account* 49](#_Toc88128749)

[***6.15.*** *Đăng nhập* 49](#_Toc88128750)

[***6.16.*** *Kiểm tra tài khoản* 50](#_Toc88128751)

[***6.17.*** *Đổi mật khẩu* 50](#_Toc88128752)

[***6.18.*** *Thêm thông tin sản phẩm vào bảng Order và Detail Order* 51](#_Toc88128753)

[***6.19.*** *Lấy oid Order vừa tạo* 51](#_Toc88128754)

[***6.20.*** *Lấy ra Order vừa tạo* 52](#_Toc88128755)

[***6.21.*** *Xem thông tin trong view ThongTinSanPham* 52](#_Toc88128756)

[***6.22.*** *Lấy những product mà do người đó bán để load lên trong manage product* 53](#_Toc88128757)

[***6.23.*** *Đăng ký tài khoản* 53](#_Toc88128758)

[7. Hướng dẫn sử dụng 54](#_Toc88128759)

[***7.1.*** **Cài đặt chương trình** 54](#_Toc88128760)

[***7.2.*** **Chi tiết** 54](#_Toc88128761)

[8. Kết luận 66](#_Toc88128762)

[***8.1.*** **Các phần đã đạt được trong đồ án** 66](#_Toc88128763)

[***8.2.*** **Một số hạn chế** 67](#_Toc88128764)

[***8.3.*** **Cam kết** 67](#_Toc88128765)

1. **Tổng quan về phần mềm**
   1. **Mục tiêu**

Ngày nay, công nghệ thông tin đã có những bước phát triển mạnh mẽ theo cả chiều rộng và sâu. Máy tính điện tử không còn là một thứ phương tiện quý hiếm mà đang ngày càng trở thành một công cụ làm việc và giải trí thông dụng của con người, không chỉ ở nơi làm việc mà còn ngay cả trong gia đình. Đặc biệt là công nghệ thông tin được áp dụng trên mọi lĩnh vực kinh tế, chính trị, xã hội. Nó đóng vai trò hết sức quan trọng.

Trong hoạt động sản xuất, kinh doanh giờ đây thương mại điện tử và ứng dụng công nghệ thông tin đã khẳng định được vai trò xúc tiến và thúc đẩy sự phát triển của doanh nghiệp. Đối với một cửa hàng hay shop, việc quảng bá và giới thiệu đến khách hàng các sản phẩm mới đáp ứng được nhu cầu của khách hàng đồng thời cho khách hàng thấy được sự khác biệt giữa các sản phẩm của các nhà sản xuất (thương hiệu) khác nhau sẽ là cần thiết. Vậy phải quảng bá thế nào đó là xây dựng được một Website cho cửa hàng của mình giới thiệu tất cả các sản phẩm của mình bán. Vì vậy, nhóm em quyết định thực hiện đề tài ***“Xây dựng website kinh doanh giày và phụ kiện”*** cho một số thương hiệu giày nổi tiếng hiện nay.

* 1. **Phạm vi**

Đề tài được áp dụng dựa trên Quản lý một trang thương mại điện tử cơ bản. Các chức năng tập trung vào việc quản lý các thông tin của người dùng và mua bán, sản giữa người bán và khách hàng.

* 1. **Mô tả**

Những người tham gia sử dụng trang web gồm 3 thành phần: quản trị viên quản lý trang web, người bán và khách hàng (người mua). Để sử dụng trang web, mỗi người dùng cần thiết tạo một tài khoản (Account) phân biệt nhau qua 2 thuộc tính là user và pass. Bên cạnh đó để phân loại 3 nhóm trên, sẽ có 2 thuộc tính nữa là isSell và isAdmin đảm nhiệm. Tiếp đến trang web sẽ quản lý thêm một số thông tin cá nhân của người dùng như tên, tuổi, email, số điện thoại và địa chỉ. Có thể liên lạc với các người sử dụng khác qua tin nhắn trên trang web. Các sản phẩm được bán trên web sẽ được chia ra theo những danh mục nhất định. Thông tin các sản phẩm bao gồm tên sản phẩm, hình ảnh, giá sản phẩm, mô tả. Khi đặt hàng một (hoặc nhiều) sản phẩm nào đó người mua có thể nắm được số lượng sản phẩm, tổng số tiền phải trả, ngày giao dịch, địa chỉ nhận hàng, email và số điện thoại để liên hệ, cuối cùng là chọn cách thanh toán. Hóa đơn thanh toán sẽ cung cấp thông tin giảm giá, số tiền phải trả sau khi áp dụng giảm giá, ngày tạo hóa đơn, ngày bắt đầu trả góp (nếu trả góp) và trạng thái thanh toán.

* 1. **Thông tin phần mềm**

*Tên phần mềm:*Website kinh doanh giày, phụ kiện

*Ngôn ngữ lập trình:* Java

*Công nghệ sử dụng:* Servlet và JSP

*Phần mềm hỗ trợ:* SQL Server.

- Website được thiết kế theo hướng mở, có thể nâng cấp và cập nhật thêm các tính năng mới khi có nhu cầu.

- Hỗ trợ sử dụng tiếng việt theo đúng chuẩn Unicode. Tương thích với các loại trình duyệt Internet như Google Chrome, Microsoft edge,…

- Hiển thị trên mọi giao diện của các thiết bị như: Laptop, mobile, Ipad,…

- Công cụ phát triển Đề tài được xây dựng dựa trên các công cụ, ngôn ngữ: + Front-end : HTML, Javascript, CSS, Bootstrap framework.

+ Back-end : Java, Mô hình MVC, công nghệ servlet và jsp, hệ quản trị CSDL

* 1. **Chức năng của phần mềm**
* Tra cứu, tìm kiếm thông tin sản phẩm một cách nhanh chóng và chính xác.
* Thêm, sửa, xóa sản phẩm của người bán hàng.
* Hiển thị danh sách các sản phẩm theo thương hiệu và thông tin chi tiết của từng sản phẩm
* Chức năng quản lý thông tin cá nhân.
* Chức năng đặt hàng.
* Chức năng thêm, xóa, sửa sản phẩm trong giỏ hàng
* Chức năng phân quyền Admin , người mua, người bán.

1. **Lược đồ quan hệ**
   1. **Mô hình ERD**
      1. Mô hình ERD

Diagram

Description automatically generated

* + 1. Lược đồ quan hệ

**Category**(cid, cname)

**Product**(id, name, image, price, title, description, cateID, sale\_ID)

**Account**(uid, user, pass, isSell, isAdmin)

**Member**(mid, name, age, email, phone, address)

**Order**(oid, uid, totalQuantity, totalMoney, purchaseDate , address, email,

phone, state)

**DetailOrder**(oid , pid, Quantity)

**Bill**(bid, totalMoney, discount, startDate, createDate, State)

**Contact**(name, email, message)

* 1. **Mô tả cơ sở dữ liệu**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **STT** |
| Category | | | | |
| 1 | cid | int | Primary key | Mã danh mục |
| 2 | cname | varchar(50) |  | Tên danh mục |
| Product | | | | |
| 1 | id | int | Primary key | Mã sản phẩm |
| 2 | name | varchar(50) |  | Tên sản phẩm |
| 3 | image | varchar(50) |  | Hình ảnh sản phẩm |
| 4 | price | double |  | Giá sản phẩm |
| 5 | title | varchar(50) |  | Tiêu đề hiển thị cho người mua |
| 6 | description | varchar(50) |  | Mô tả |
| 7 | cateID | int | Foreign key | Mã danh mục |
| 8 | sale\_ID | int | Foreign key | Mã người bán |
| Account | | | | |
| 1 | uid | int | Primary key | Mã người dùng |
| 2 | user | varchar(50) |  | Tên đăng nhập của tài khoản |
| 3 | pass | varchar(50) |  | Mật khẩu cho tài khoản |
| 4 | isSell | int |  | Có phải người bán không |
| 5 | isAdmin | int |  | Có phải quản trị viên không |
| Member | | | | |
| 1 | mid | int | Primary key | Mã người tham gia |
| 2 | name | varchar(50) |  | Họ và tên |
| 3 | age | int |  | Tuổi |
| 4 | email | varchar(50) |  | Địa chỉ email |
| 5 | phone | varchar(50) |  | Số điện thoại |
| 6 | address | varchar(50) |  | Địa chỉ |
| Order | | | | |
| 1 | oid | int | Primary key | Mã đặt hàng |
| 2 | uid | int | Foreign key | Mã người dùng |
| 3 | totalQuantity | int |  | Tổng số lượng |
| 4 | totalMoney | int |  | Tổng số tiền |
| 5 | purchaseDate | date |  | Ngày đặt hàng |
| 6 | address | varchar(50) |  | Địa chỉ nhận |
| 7 | email | varchar(50) |  | Địa chỉ email |
| 8 | phone | varchar(50) |  | Số điện thoại liên hệ |
| 9 | state | varchar(50) |  | Chọn trạng thái thanh toán |
| DetailOrder | | | | |
| 1 | oid | int | Primary key | Mã đặt hàng |
| 2 | pid | int | Foreign key | Mã sản phẩm |
| 3 | Quantity | int |  | Số lượng |
| Bill | | | | |
| 1 | bid | int | Primary key | Mã hóa đơn |
| 2 | totalMoney | int |  | Tổng số tiền cuối cùng |
| 3 | discount | int |  | Thông tin giảm giá |
| 4 | startDate | date |  | Ngày bắt đầu trả góp (nếu có) |
| 5 | createDate | date |  | Ngày tạo hóa đơn |
| 6 | State | varchar(50) |  | Trạng thái thanh toán |
| Contact | | | | |
| 1 | name | varchar(50) | Primary key | Tên người dùng |
| 2 | email | varchar(50) |  | Địa chỉ email |
| 3 | message | longtext |  | Tin nhắn |

* 1. **Biểu đồ quan hệ giữa các bảng (Database diagram)**

**Chi tiết:**

A picture containing text, screenshot, indoor

Description automatically generated

Diagram

Description automatically generated

# **Code tạo bảng, khoá chính, ràng buộc khoá ngoại**

|  |
| --- |
| CREATE DATABASE Shoe\_DB  GO  USE Shoe\_DB  GO  CREATE TABLE "Contact"(  "name" NVARCHAR(50),  "email" NVARCHAR(50),  "message" NVARCHAR(50)  )  CREATE TABLE "Account"(  "uid" INT IDENTITY(1,1) PRIMARY KEY,  "user" NVARCHAR(50),  "pass" NVARCHAR(50),  "isSell" INT,  "isAdmin" INT,  );  GO  CREATE TABLE "Order"(  "oid" INT IDENTITY(1,1) PRIMARY KEY,  "uid" INT,  "totalQuantity" INT,  "totalMoney" FLOAT,  "purchaseDate" DATE,  "address" NVARCHAR(50),  "email" NVARCHAR(50),  "phone" NVARCHAR(50),  "state" nvarchar(50),  ) ;  GO  CREATE TABLE "Product"(  "id" INT IDENTITY(1,1) PRIMARY KEY,  "name" NVARCHAR(50),  "image" NVARCHAR(MAX),  "price" FLOAT,  "title" NVARCHAR(50),  "description" NVARCHAR(50),  "cateID" INT,  "sale\_ID" INT  ) ;  GO  CREATE TABLE "Category"(  "cid" INT PRIMARY KEY,  "cname" NVARCHAR(50)  ) ;  GO  CREATE TABLE "Member"(  "mid" INT PRIMARY KEY,  "name" NVARCHAR(50),  "age" INT,  "email" NVARCHAR(50),  "phone" NVARCHAR(50),  "address" NVARCHAR(50)  ) ;  GO  CREATE TABLE "DetailOrder"(  "oid" INT,  "pid" INT,  "Quantity" INT,  PRIMARY KEY ("oid","pid")  )  CREATE TABLE "Bill"(  "bid" INT PRIMARY KEY,  "totalMoney" INT,  "discount" INT,  "startDate" DATE,  "createDate" DATE,  "State" NVARCHAR(50)  )  go  ---Khoá ngoại--  ALTER TABLE "Product"  ADD CONSTRAINT FK\_Product\_Account  FOREIGN KEY ("sale\_ID") REFERENCES "Account"("uid") ON UPDATE CASCADE;  GO  ALTER TABLE "Product"  ADD CONSTRAINT FK\_Product\_Category  FOREIGN KEY ("cateID") REFERENCES "Category"("cid") ON UPDATE CASCADE;  GO  ALTER TABLE "DetailOrder"  ADD CONSTRAINT FK\_DetailOrder\_Order  FOREIGN KEY ("oid") REFERENCES "Order"("oid") ON UPDATE CASCADE;  GO  ALTER TABLE "DetailOrder"  ADD CONSTRAINT FK\_DetailOrder\_Product  FOREIGN KEY ("pid") REFERENCES "Product"("id") ON UPDATE CASCADE;  GO  ALTER TABLE "Order"  ADD CONSTRAINT FK\_Order\_Account  FOREIGN KEY ("uid") REFERENCES "Account"("uid");  GO  ALTER TABLE "Member"  ADD CONSTRAINT FK\_Member\_Account  FOREIGN KEY ("mid") REFERENCES "Account"("uid") ON UPDATE CASCADE;  GO |

1. Danh sách các views, stored procedure, function, triggers
   1. **Views**
      1. Hiển thị thông tin sản phẩm

|  |
| --- |
| -- Hiển thị thông tin sản phẩm  CREATE VIEW vw\_ThongTinSanPham AS  SELECT "name", "image", price, title, "description"  FROM Product;  GO |

* + 1. Quản lý thông tin cá nhân

|  |
| --- |
| -- Quản lý thông tin cá nhân  CREATE VIEW ThongTinCaNhan AS  SELECT name, age, email, phone, address  FROM Member;  GO |

* + 1. Thông tin các đơn hàng

|  |
| --- |
| -- Thông tin chi tiết các đơn hàng---  CREATE VIEW vw\_ThongTinChiTietCacDonHang AS  SELECT D."oid",p."name", p."image",D.Quantity, price\*D."Quantity" AS price, state  FROM Product P INNER JOIN "DetailOrder" D ON P.id = D.pid  INNER JOIN "Order" O ON D.oid = O.oid  GO |

* + 1. Quản lý người dùng

|  |
| --- |
| -- Quản lý người dùng---  GO  CREATE VIEW ThongTinNguoiDung AS  SELECT "uid", "user", isSell, isAdmin  FROM Account;  GO |

2. 1. **Stored procedures**
      1. Account
         1. Lấy tất cả nhân viên

|  |
| --- |
| --lấy tất cả Account---  CREATE PROC sp\_getAllAccount  AS  BEGIN  SELECT  "uid",  "user",  "pass",  "isSell",  "isAdmin"  FROM  "Account"  ORDER BY  "uid";  END;  go |

* + - 1. Lấy một tài khoản theo id

|  |
| --- |
| --Lấy một Account--  CREATE PROC sp\_getOneAccount  (  @uid INT  )  AS  BEGIN  SELECT  "uid",  "user",  "pass",  "isSell",  "isAdmin"  FROM  "Account"  WHERE  "uid"= @uid  END;  go |

* + - 1. *Xoá tài khoản*

|  |
| --- |
| --Xoá Account--  CREATE PROC sp\_DeleteAccount  (  @uid INT  )  AS  BEGIN  DELETE  FROM  "Account"  WHERE  "uid"= @uid  END;  go |

* + - 1. *Thêm tài khoản*

|  |
| --- |
| --Thêm Account--  CREATE PROC sp\_InsertAccount  (  @user NVARCHAR(50),  @pass NVARCHAr(50),  @isSell INT,  @isAdmin INT  )  AS  INSERT INTO "Account"("user","pass","isSell","isAdmin")  VALUES (@user,@pass,@isSell,@isAdmin)  go |

* + - 1. *Cập nhật Account*

|  |
| --- |
| --Cập nhật Account--  CREATE PROC sp\_UpdateAccount  (  @uid INT,  @user NVARCHAR(50),  @pass NVARCHAr(50),  @isSell INT,  @isAdmin INT  )  AS  UPDATE "Account" SET  "user" = @user,  "pass" = @pass,  "isSell" = @isSell,  "isAdmin" = @isAdmin  WHERE "uid" = @uid  go |

* + 1. *Bill*
       1. Lấy tất hoá đơn

|  |
| --- |
| --lấy tất cả Bill---  CREATE PROC sp\_getAllBill  AS  BEGIN  SELECT  "bid",  "totalMoney",  "discount",  "createDate",  "startDate",  "State"  FROM  "Bill"  ORDER BY  "bid";  END;  go |

* + - 1. Lấy một hoá đơn theo id

|  |
| --- |
| --Lấy một Bill--  CREATE PROC sp\_getOneBill  (  @bid INT  )  AS  BEGIN  SELECT  "bid",  "totalMoney",  "discount",  "startDate",  "createDate",  "State"  FROM  "Bill"  WHERE  "bid"= @bid  END;  go |

* + - 1. *Xoá hoá đơn*

|  |
| --- |
| --Xoá Bill--  CREATE PROC sp\_DeleteBill  (  @bid INT  )  AS  BEGIN  DELETE  FROM  "Bill"  WHERE  "bid"= @bid  END;  go |

* + - 1. *Thêm hoá đơn*

|  |
| --- |
| --Thêm Bill--  CREATE PROC sp\_InsertBill  (  @bid INT,  @totalMoney INT,  @discount INT,  @startDate DATE,  @createDate DATE,  @State NVARCHAR(50)  )  AS  INSERT INTO "Bill"("bid","totalMoney","discount","startDate","createDate","State")  VALUES (@bid,@totalMoney,@discount,@startDate,@createDate,@State)  go |

* + - 1. *Cập nhật hoá đơn*

|  |
| --- |
| --Cập nhật Bill--  CREATE PROC sp\_UpdateBill  (  @bid INT,  @totalMoney INT,  @discount INT,  @startDate DATE,  @createDate DATE,  @State NVARCHAR(50)  )  AS  UPDATE "Bill" SET  "totalMoney" = @totalMoney,  "discount" = @discount,  "startDate" = @startDate,  "createDate" = @createDate,  "State" = @State  WHERE "bid" = @bid  go |

* + 1. *Category*
       1. Lấy tất cả danh mục

|  |
| --- |
| --lấy tất cả Category---  CREATE PROC sp\_getAllCategory  AS  BEGIN  SELECT  "cid",  "cname"  FROM  "Category"  ORDER BY  "cid";  END;  go |

* + - 1. Lấy một danh mục theo id

|  |
| --- |
| --Lấy một Category--  CREATE PROC sp\_getOneCategory  (  @cid INT  )  AS  BEGIN  SELECT  "cid",  "cname"  FROM  "Category"  WHERE  "cid"= @cid  END;  go |

* + - 1. *Xoá danh mục*

|  |
| --- |
| --Xoá Category--  CREATE PROC sp\_DeleteCategory  (  @cid INT  )  AS  BEGIN  DELETE  FROM  "Category"  WHERE  "cid"= @cid  END;  go |

* + - 1. *Thêm danh mục*

|  |
| --- |
| --Thêm Category--  CREATE PROC sp\_InsertCategory  (  @cid INT,  @cname NVARCHAR(50)  )  AS  INSERT INTO "Category"("cid","cname")  VALUES (@cid,@cname)  go |

* + - 1. *Cập danh mục*

|  |
| --- |
| --Cập nhật Category--  CREATE PROC sp\_UpdateCategory  (  @cid INT,  @cname NVARCHAR(50)  )  AS  UPDATE "Category" SET  "cname" = @cname  WHERE "cid" = @cid  go |

* + 1. *Contact*
       1. Lấy tất cả liên lạc

|  |
| --- |
| --lấy tất cả Contact---  CREATE PROC sp\_getAllContact  AS  BEGIN  SELECT  "name",  "email",  "message"  FROM  "Contact"  ORDER BY  "name";  END;  go |

* + - 1. Lấy một liên lạc theo name

|  |
| --- |
| --Lấy một Contact--  CREATE PROC sp\_getOneContact  (  @name NVARCHAR(50)  )  AS  BEGIN  SELECT  "name",  "email",  "message"  FROM  "Contact"  WHERE  "name"= @name  END;  go |

* + - 1. *Xoá liên lạc*

|  |
| --- |
| --Xoá Contact--  CREATE PROC sp\_DeleteContact  (  @name NVARCHAR(50)  )  AS  BEGIN  DELETE  FROM  "Contact"  WHERE  "name"= @name  END;  go |

* + - 1. *Thêm liên lạc*

|  |
| --- |
| --Thêm Contact--  CREATE PROC sp\_InsertContact  (  @name NVARCHAR(50),  @email NVARCHAR(50),  @message NVARCHAR(MAX)  )  AS  INSERT INTO "Contact"("name","email","message")  VALUES (@name,@email,@message)  go |

* + - 1. *Cập liên lạc*

|  |
| --- |
| --Cập nhật Contact--  CREATE PROC sp\_UpdateContact  (  @name NVARCHAR(50),  @email NVARCHAR(50),  @message NVARCHAR(MAX)  )  AS  UPDATE "Contact" SET  "email" = @email,  "message" = @message  WHERE "name" = @name  go |

* + 1. *DetailOrder*
       1. Lấy tất cả chi tiết đơn hàng

|  |
| --- |
| --lấy tất cả DetailOrder---  CREATE PROC sp\_getAllDetailOrder  AS  BEGIN  SELECT  "oid",  "pid",  "Quantity"  FROM  "DetailOrder"  ORDER BY  "oid";  END;  go |

* + - 1. Lấy một chi tiết đơn hàng theo id

|  |
| --- |
| --Lấy một DetailOrder--  CREATE PROC sp\_getOneDetailOrder  (  @oid INT,  @pid INT  )  AS  BEGIN  SELECT  "oid",  "pid",  "Quantity"  FROM  "DetailOrder"  WHERE  "oid"= @oid AND "pid"= @pid  END;  go |

* + - 1. *Xoá chi tiết đơn hàng*

|  |
| --- |
| --Xoá DetailOrder--  CREATE PROC sp\_DeleteDetailOrder  (  @oid INT,  @pid INT  )  AS  BEGIN  DELETE  FROM  "DetailOrder"  WHERE  "oid"= @oid AND "pid"= @pid  END;  go |

* + - 1. *Thêm chi tiết đơn hàng*

|  |
| --- |
| --Thêm DetailOrder--  CREATE PROC sp\_InsertDetailOrder  (  @oid INT,  @pid INT,  @Quantity INT  )  AS  INSERT INTO "DetailOrder"("oid","pid","Quantity")  VALUES (@oid,@pid,@Quantity)  go |

* + - 1. *Cập chi tiết đơn hàng*

|  |
| --- |
| --Cập nhật DetailOrder--  CREATE PROC sp\_UpdateDetailOrder  (  @oid INT,  @pid INT,  @Quantity INT  )  AS  UPDATE "DetailOrder" SET  "Quantity" = @Quantity  WHERE "oid" = @oid AND "pid" = @pid  go |

* + 1. *Member*
       1. Lấy tất cả thông tin thành viên

|  |
| --- |
| --lấy tất cả Member---  CREATE PROC sp\_getAllMember  AS  BEGIN  SELECT  "mid",  "name",  "age",  "email",  "phone",  "address"  FROM  "Member"  ORDER BY  "mid";  END;  go |

* + - 1. Lấy một thông tin thành viên theo id

|  |
| --- |
| --Lấy một Member--  CREATE PROC sp\_getOneMember  (  @mid INT  )  AS  BEGIN  SELECT  "mid",  "name",  "age",  "email",  "phone",  "address"  FROM  "Member"  WHERE  "mid"= @mid  END;  go |

* + - 1. *Xoá thông tin thành viên*

|  |
| --- |
| --Xoá Member--  CREATE PROC sp\_DeleteMember  (  @mid INT  )  AS  BEGIN  DELETE  FROM  "Member"  WHERE  "mid"= @mid  END;  go |

* + - 1. *Thêm thông tin thành viên*

|  |
| --- |
| --Thêm Member--  CREATE PROC sp\_InsertMember  (  @mid INT,  @name NVARCHAR(50),  @age INT,  @email NVARCHAR(50),  @phone NVARCHAR(50),  @address NVARCHAR(50)  )  AS  INSERT INTO "Member"("mid","name","age","email","phone","address")  VALUES (@mid,@name,@age,@email,@phone,@address)  go |

* + - 1. *Cập nhật thông tin thành viên*

|  |
| --- |
| --Cập nhật Member--  CREATE PROC sp\_UpdateMember  (  @mid INT,  @name NVARCHAR(50),  @age INT,  @email NVARCHAR(50),  @phone NVARCHAR(50),  @address NVARCHAR(50)  )  AS  UPDATE "Member" SET  "name" = @name,  "age" = @age,  "email" = @email,  "phone" = @phone,  "address" = @address  WHERE "mid" = @mid  go |

* + 1. Order
       1. Lấy tất cả đơn hàng

|  |
| --- |
| --lấy tất cả Order---  CREATE PROC sp\_getAllOrder  AS  BEGIN  SELECT  "oid",  "uid",  "totalQuantity",  "totalMoney",  "purchaseDate",  "address",  "email",  "phone",  "state"  FROM  "Order"  ORDER BY  "oid";  END;  go |

* + - 1. Lấy một đơn hàng theo id

|  |
| --- |
| --Xoá Order--  CREATE PROC sp\_DeleteOrder  (  @oid INT  )  AS  BEGIN  DELETE  FROM  "Order"  WHERE  "oid"= @oid  END;  go |

* + - 1. *Xoá đơn hàng*

|  |
| --- |
| --Xoá Order--  CREATE PROC sp\_DeleteOrder  (  @oid INT  )  AS  BEGIN  DELETE  FROM  "Order"  WHERE  "oid"= @oid  END;  go |

* + - 1. *Thêm đơn hàng*

|  |
| --- |
| --Thêm Order--  CREATE PROC sp\_InsertOrder  (  @uid INT,  @totalQuantity INT,  @totalMoney INT,  @purchaseDate DATE,  @address NVARCHAR(50),  @email NVARCHAR(50),  @phone NVARCHAR(50),  @state NVARCHAR(50)  )  AS  INSERT INTO "Order"("uid","totalQuantity","totalMoney"  ,"purchaseDate","address","email","phone","state")  VALUES (@uid, @totalQuantity, @totalMoney, @purchaseDate  ,@address, @email, @phone,@state)  go |

* + - 1. *Cập nhật chi tiết đơn hàng*

|  |
| --- |
| --Cập nhật Order--  CREATE PROC sp\_UpdateOrder  (  @oid INT,  @uid INT,  @totalQuantity INT,  @totalMoney INT,  @purchaseDate DATE,  @address NVARCHAR(50),  @email NVARCHAR(50),  @phone NVARCHAR(50),  @state NVARCHAR(50)  )  AS  UPDATE "Order" SET  "uid" = @uid,  "totalQuantity" = @totalQuantity,  "totalMoney" = @totalMoney,  "purchaseDate" = @purchaseDate,  "address" = @address,  "email" = @email,  "phone" = @phone,  "state" = @state  WHERE "oid" = @oid  go |

* + 1. *Product* 
       1. Lấy tất cả sản phẩm

|  |
| --- |
| --lấy tất cả Product---  CREATE PROC sp\_getAllProduct  AS  BEGIN  SELECT  "id",  "name",  "image",  "price",  "title",  "description",  "cateID",  "sale\_ID"  FROM  "Product"  ORDER BY  "id";  END;  go |

* + - 1. Lấy một sản phẩm theo id

|  |
| --- |
| --Lấy một Product--  CREATE PROC sp\_getOneProduct  (  @id INT  )  AS  BEGIN  SELECT  "id",  "name",  "image",  "price",  "title",  "description",  "cateID",  "sale\_ID"  FROM  "Product"  WHERE  "id"= @id  END;  go |

* + - 1. *Xoá sản phẩm*

|  |
| --- |
| --Xoá Product--  CREATE PROC sp\_DeleteProduct  (  @id INT  )  AS  BEGIN  DELETE  FROM  "Product"  WHERE  "id"= @id  END;  go |

* + - 1. *Thêm sản phẩm*

|  |
| --- |
| --Thêm Product--  CREATE PROC sp\_InsertProduct  (  @name NVARCHAR(50),  @image NVARCHAR(MAX),  @price FLOAT,  @title NVARCHAR(50),  @description NVARCHAR(50),  @cateID INT,  @sale\_ID INT  )  AS  INSERT INTO "Product"("name","image","price","title"  ,"description","cateID","sale\_ID")  VALUES (@name,@image,@price,@title,@description,@cateID,@sale\_ID)  go |

* + - 1. *Cập nhật sản phẩm*

|  |
| --- |
| --Cập nhật Product--  CREATE PROC sp\_UpdateProduct  (  @id INT,  @name NVARCHAR(50),  @image NVARCHAR(MAX),  @price FLOAT,  @title NVARCHAR(50),  @description NVARCHAR(50),  @cateID INT,  @sale\_ID INT  )  AS  UPDATE "Product" SET  "name" = @name,  "image" = @image,  "price" = @price,  "title" = @title,  "description" = @description,  "cateID" = @cateID,  "sale\_ID" = @sale\_ID  WHERE "id" = @id  Go |

* + 1. Một số stored procedure tiện ích khác.
       1. Tìm các sản phẩm mà một tài khoản nào đó bán

|  |
| --- |
| ---Tìm các sảm phẩm mà một tài khoản đó bán---  CREATE PROC sp\_getProductBySaleID  (  @uid INT  )  AS  SELECT \* FROM Product p WHERE p.sale\_ID = @uid  go |

* + - 1. Thêm một đơn hàng khi nhấn nút thanh toán

|  |
| --- |
| ---Thêm đơn hàng khi nhấn nút thanh toán---  CREATE PROC sp\_InsertOrderWithDetail  (  @uid INT  )  AS  BEGIN  DECLARE @address NVARCHAR(50);  DECLARE @email NVARCHAR(50);  DECLARE @phone NVARCHAR(50);  DECLARE @state NVARCHAR(50);  DECLARE @lastID INT;  SET @lastID = NULL;  SET @state = 'created';  SELECT @email = m.email,@phone=m.phone, @address = m."address"  FROM Member m  WHERE m.mid = @uid  INSERT INTO "Order"("uid","totalMoney","totalQuantity","address","email","phone","state")  VALUES (@uid,0,0,@address, @email, @phone,@state);  END;  go |

* + - 1. *Lấy sản phẩm mới nhất*

|  |
| --- |
| --Lấy sản phẩm mới nhất  CREATE PROC sp\_getLastProduct  AS  BEGIN  select top 1 \*  from Product  order by id desc;  END;  go "id"= @id  END;  go |

* + - 1. *Tạo người dùng và phân quyền người dùng*

|  |
| --- |
| --Tạo người dùng và phân quyền trong Database  CREATE OR ALTER PROC USP\_CREATE\_LOGIN\_USER  (  @Role\_Name NVARCHAR(50),  @Login\_Name NVARCHAR(50),  @Password\_Login NVARCHAR(50)  )  AS  BEGIN  DECLARE @Login\_UserName VARCHAR(50),  @QueryLogin VARCHAR(100),  @QueryUser VARCHAR(100)  SET @Login\_UserName = @Login\_Name  SET @QueryLogin ='CREATE LOGIN ' + @Login\_UserName + ' WITH PASSWORD = ' + QUOTENAME(@Password\_Login, '''')  SET @QueryUser = CONCAT('CREATE USER ', @Login\_UserName, ' FOR LOGIN ', @Login\_UserName);  EXEC (@QueryLogin)  EXEC (@QueryUser)  EXEC sys.sp\_addrolemember @rolename = @Role\_Name,  @membername = @Login\_Name  END  GO |

* 1. **Functions**
     1. *Table-valued functions*
        1. *Tìm sản phẩm khi nhập tên gần đúng*

|  |
| --- |
| ---Tìm tên sản phẩm khi nhập gần đúng tên---  CREATE FUNCTION fn\_SearchProductName(@name NVARCHAR(100))  RETURNS TABLE  AS RETURN SELECT \* FROM Product WHERE "name" LIKE '%'+@name+'%'  GO |

* + - 1. *Tìm giá của sản phẩm cao nhất theo từng danh mục*

|  |
| --- |
| --Tìm giá của sản phẩm cao nhất trong từng danh mục--  CREATE FUNCTION fn\_SearchMaxPriceByCategory()  RETURNS TABLE  AS RETURN  SELECT c.cname, MAX(p.price) AS maxPrice  FROM Product p, Category c  WHERE p.cateID= c.cid  GROUP BY c.cname  GO |

* + - 1. *Tìm sản phẩm trong tầm giá*

|  |
| --- |
| --Tìm những sản phẩm trong tầm giá---  CREATE FUNCTION fn\_ProductInRangePrice(@min FLOAT, @max FLOAT)  RETURNS TABLE  AS RETURN  SELECT \*  FROM Product p  WHERE p.price >@min AND p.price<@max  GO |

* + - 1. *Tìm sản phẩm theo tên danh mục*

|  |
| --- |
| --Tìm tất cả các sản phẩm theo tên danh mục--  CREATE FUNCTION fn\_ProductByCateName(@name NVARCHAR(50))  RETURNS TABLE  AS RETURN  SELECT \*  FROM Product p,Category c  WHERE p.cateID = c.cid AND c."cname" LIKE '%'+@name+'%'  GO |

* + - 1. *Tìm sản phẩm theo tên của người bán*

|  |
| --- |
| --Tìm tất cả các sản phẩm theo tên của người bán--  CREATE FUNCTION fn\_ProductBySalerName(@name NVARCHAR(50))  RETURNS TABLE  AS RETURN  SELECT m."name" AS NguoiBan, p."name" AS SanPham  FROM Product p,Member m  WHERE p.sale\_ID = m.mid AND m."name" LIKE '%'+@name+'%'  GO |

* + - 1. *Tìm những sản phẩm và số tiền người bán đã bán được*

|  |
| --- |
| ---Tìm những sản phẩm và số tiền mà người bán đã bán được ---  CREATE FUNCTION fn\_ProductAndAmount(@name NVARCHAR(50))  RETURNS TABLE  AS RETURN  SELECT p."name",SUM(d.Quantity) AS Quantity, SUM(p.price\*d.Quantity) AS Amount  FROM Product p,DetailOrder d, Member m  WHERE p.id = d.pid AND p.sale\_ID = m.mid AND m."name" LIKE '%'+@name+'%'  GROUP BY p."name"  GO |

* + - 1. *Lấy danh sách các mặt hàng trong hoá đơn*

|  |
| --- |
| ---Danh sach cac mat hang trong hoa don---  CREATE FUNCTION fn\_HoaDon(@oid INT)  RETURNS TABLE  AS RETURN  SELECT p."name",price  FROM DetailOrder d, Product p  WHERE d.oid = @oid AND d.pid = p.id  go |

* + 1. *Scalar-valued functions*
       1. *Lấy ID của đơn hàng cuối cùng*

|  |
| --- |
| ---Lấy ID của đơn hàng cuối cùng( mới nhất)----  CREATE FUNCTION fn\_LastOrderID()  RETURNS INT  AS  BEGIN  DECLARE @lastID INT;  SELECT @lastID = MAX(oid)  FROM "Order"  RETURN @lastID  END |

* 1. **Triggers**
     1. *Tự động tạo Account và cấp quyền khi thêm dữ liệu trong bảng Account*

|  |
| --- |
| --Tự động tạo account và cấp quyền khi thêm dữ liệu trong bảng account--.  CREATE TRIGGER AutoCreateUserDB ON "Account"  FOR INSERT  AS  BEGIN  DECLARE @LGNAME NVARCHAR(50), @PASS NVARCHAR(50), @ROLE INT;  SELECT @LGNAME= new."user", @PASS = new.pass, @ROLE = new.isAdmin  FROM Account old, Inserted new  WHERE old."uid" = new."uid"  IF @ROLE= 2  BEGIN  --EXEC sp\_addrolemember 'ADMIN', @LGNAME  EXEC USP\_CREATE\_LOGIN\_USER 'ADMIN',@LGNAME,@PASS  END  IF @ROLE= 1  BEGIN  --EXEC sp\_addrolemember 'SELLER', @LGNAME  EXEC USP\_CREATE\_LOGIN\_USER 'SELLER',@LGNAME,@PASS  END  IF @ROLE= 0  BEGIN  --EXEC sp\_addrolemember 'CUSTOMER', @LGNAME  EXEC USP\_CREATE\_LOGIN\_USER 'CUSTOMER',@LGNAME,@PASS  END  END  go |

* + 1. *Người dùng tuổi nhỏ hơn 10 không thể mua hàng*

|  |
| --- |
| --Những người dùng có tuổi nhỏ hơn 10 sẽ không thể mua hàng.  CREATE TRIGGER Prevent\_Age\_lr\_10 ON "Order"  FOR INSERT, UPDATE  AS  BEGIN  IF EXISTS (SELECT\*  FROM "Order" O, Member M  WHERE O."uid" = M.Mid AND M.Age <10 AND M.age != NULL)  BEGIN  RAISERROR(N'người dùng tuổi nhỏ hơn 10 không thể mua hàng',16,1)  ROLLBACK TRANSACTION  END  END  go |

* + 1. *Giảm giá 25% cho các sản phẩm thuộc danh mục ‘socks’*

|  |
| --- |
| --Giảm giá 25% cho các sản phẩm thuộc danh mục 'socks'  CREATE TRIGGER Discount\_for\_Cat ON "DetailOrder"  AFTER INSERT  AS  BEGIN  DECLARE @oid INT, @TenSP NVARCHAR(50)  SELECT @oid=I.oid, @TenSP=P."name"  FROM inserted I,Product P,DetailOrder D  WHERE I.oid = D.oid AND D.pid = P.id  IF (@TenSP ='socks')  BEGIN  UPDATE "Order"  SET totalMoney = totalMoney\*0.75  WHERE Oid = @oid  END  END  Go |

* + 1. *Tự động thêm ngày mua trong Order*

|  |
| --- |
| ---Tu dong them ngay mua trong Orrder----  CREATE TRIGGER tg\_AutoDate ON "Order"  AFTER INSERT  AS  DECLARE @oid INT , @ngay DATE = getDATE()  SELECT @oid = inserted."oid"  FROM inserted  BEGIN  UPDATE "Order"  SET purchaseDate = @ngay  WHERE oid = @oid  END  go |

* + 1. *Tự động cập nhật số món hàng, tổng giá, giảm giá cho người già trong Order*

|  |
| --- |
| ---Tính số lượng món hàng, TotalMoney cho Order khi sửa trong DetailOder----co dùng TRANSACTION--Kèm giảm giá người già  ALTER TRIGGER [dbo].[tg\_TotalQuantityMoneyOfOrder\_inOder] ON [dbo].[Order]  AFTER INSERT,UPDATE  AS  DECLARE @oid INT, @totalMoney FLOAT, @totalQuantity INT  SELECT @oid=I.oid, @totalMoney=SUM(P."price" \*D."Quantity")  ,@totalQuantity = SUM(D."Quantity")  FROM inserted I,Product P,DetailOrder D  WHERE I.oid = D.oid AND D.pid = P.id  GROUP BY I.oid  BEGIN  BEGIN TRAN  UPDATE "Order"  SET totalMoney = @totalMoney, totalQuantity = @totalQuantity  WHERE oid = @oid  SAVE TRANSACTION thanhtoan  DECLARE @tuoi INT, @giaCu FLOAT, @giamoi FLOAT  SELECT @oid=I.oid,@tuoi = M.age, @giaCu = I.totalMoney  FROM "Order" I,Member M, Inserted new  WHERE I.uid = M.mid AND I.uid = new.uid  print(@tuoi)  IF (@tuoi >=70 )  BEGIN  UPDATE "Order"  SET totalMoney = totalMoney\*0.9, @giamoi = totalMoney\*0.9  WHERE oid = @oid  IF (@giamoi < 1000000)  BEGIN  RAISERROR('Tong don hang phai tren 1 trieu',16,1)  ROLLBACK TRANSACTION thanhtoan  END  END    COMMIT TRAN tinhtien  END  go |

* + 1. *Tự động cập nhật số món hàng, tổng giá khi thêm hoặc sửa DetailOrder*

|  |
| --- |
| ---Tính số lượng món hàng, TotalMoney cho Order khi sửa trong DetailOder----  CREATE TRIGGER tg\_TotalQuantityMoneyOfOrder ON "DetailOrder"  AFTER INSERT,UPDATE  AS  DECLARE @oid INT, @totalMoney FLOAT, @totalQuantity INT  SELECT @oid=I.oid, @totalMoney=SUM(P."price" \*D."Quantity")  ,@totalQuantity = SUM(D."Quantity")  FROM inserted I,Product P,DetailOrder D  WHERE I.oid = D.oid AND D.pid = P.id  GROUP BY I.oid  BEGIN  UPDATE "Order"  SET totalMoney = @totalMoney, totalQuantity = @totalQuantity  WHERE oid = @oid  END  GO |

2. 4. **Transaction**
   5. 1. *Tự động cập nhật số món hàng, tổng giá, giảm giá cho người già trong Order*

|  |
| --- |
| ---Tính số lượng món hàng, TotalMoney cho Order khi sửa trong DetailOder----co dùng TRANSACTION--Kèm giảm giá người già  ALTER TRIGGER [dbo].[tg\_TotalQuantityMoneyOfOrder\_inOder] ON [dbo].[Order]  AFTER INSERT,UPDATE  AS  DECLARE @oid INT, @totalMoney FLOAT, @totalQuantity INT  SELECT @oid=I.oid, @totalMoney=SUM(P."price" \*D."Quantity")  ,@totalQuantity = SUM(D."Quantity")  FROM inserted I,Product P,DetailOrder D  WHERE I.oid = D.oid AND D.pid = P.id  GROUP BY I.oid  BEGIN  BEGIN TRAN  UPDATE "Order"  SET totalMoney = @totalMoney, totalQuantity = @totalQuantity  WHERE oid = @oid  SAVE TRANSACTION thanhtoan  DECLARE @tuoi INT, @giaCu FLOAT, @giamoi FLOAT  SELECT @oid=I.oid,@tuoi = M.age, @giaCu = I.totalMoney  FROM "Order" I,Member M, Inserted new  WHERE I.uid = M.mid AND I.uid = new.uid  print(@tuoi)  IF (@tuoi >=70 )  BEGIN  UPDATE "Order"  SET totalMoney = totalMoney\*0.9, @giamoi = totalMoney\*0.9  WHERE oid = @oid  IF (@giamoi < 1000000)  BEGIN  RAISERROR('Tong don hang phai tren 1 trieu',16,1)  ROLLBACK TRANSACTION thanhtoan  END  END    COMMIT TRAN tinhtien  END  go |

1. **Phân quyền trong phần mềm**
3. 1. **Tạo user và roles**
      1. Tạo role

|  |
| --- |
| CREATE ROLE ADMIN  CREATE ROLE CUSTOMER  CREATE ROLE SELLER |

* + 1. Tạo user
       1. Stored Procedure tạo user và phân quyền.

|  |
| --- |
| CREATE OR ALTER PROC USP\_CREATE\_LOGIN\_USER  (  @Role\_Name VARCHAR(50),  @Login\_Name VARCHAR(50),  @Password\_Login VARCHAR(50)  )  AS  BEGIN  DECLARE @Login\_UserName VARCHAR(50),  @QueryLogin VARCHAR(100),  @QueryUser VARCHAR(100)  SET @Login\_UserName = @Login\_Name  SET @QueryLogin ='CREATE LOGIN ' + @Login\_UserName + ' WITH PASSWORD = ' + QUOTENAME(@Password\_Login, '''')  SET @QueryUser = CONCAT('CREATE USER ', @Login\_UserName, ' FOR LOGIN ', @Login\_UserName);  EXEC (@QueryLogin)  EXEC (@QueryUser)  EXEC sys.sp\_addrolemember @rolename = @Role\_Name,  @membername = @Login\_Name  END  GO |

* + - 1. Trigger tự động gọi stored procedure tạo người dùng và phân quyền khi thêm một dùng mời trong bảng Account.

|  |
| --- |
| --Tự động tạo account và cấp quyền khi thêm dữ liệu trong bảng account--.  CREATE TRIGGER AutoCreateUserDB ON "Account"  FOR INSERT  AS  BEGIN  DECLARE @LGNAME NVARCHAR(50), @PASS NVARCHAR(50), @ROLE INT;  SELECT @LGNAME= new."user", @PASS = new.pass, @ROLE = new.isAdmin  FROM Account old, Inserted new  WHERE old."uid" = new."uid"  IF @ROLE= 2  BEGIN  --EXEC sp\_addrolemember 'ADMIN', @LGNAME  EXEC USP\_CREATE\_LOGIN\_USER 'ADMIN',@LGNAME,@PASS  END  IF @ROLE= 1  BEGIN  --EXEC sp\_addrolemember 'SELLER', @LGNAME  EXEC USP\_CREATE\_LOGIN\_USER 'SELLER',@LGNAME,@PASS  END  IF @ROLE= 0  BEGIN  --EXEC sp\_addrolemember 'CUSTOMER', @LGNAME  EXEC USP\_CREATE\_LOGIN\_USER 'CUSTOMER',@LGNAME,@PASS  END  END  go |

* 1. **Phân quyền đăng nhập hệ thống**
     1. Phân quyền ADMIN

|  |
| --- |
| --Grant ADMIN  GRANT EXECUTE ON [dbo].[sp\_DeleteAccount] TO ADMIN WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_DeleteBill] TO ADMIN WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_DeleteCategory] TO ADMIN WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_DeleteContact] TO ADMIN WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_DeleteDetailOrder] TO ADMIN WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_DeleteMember] TO ADMIN WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_DeleteOrder] TO ADMIN WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_DeleteProduct] TO ADMIN WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_getAllAccount] TO ADMIN WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_getAllBill] TO ADMIN WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_getAllCategory] TO ADMIN WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_getAllContact] TO ADMIN WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_getAllMember] TO ADMIN WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_getAllOrder] TO ADMIN WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_getAllProduct] TO ADMIN WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_getAllDetailOrder] TO ADMIN WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_getOneAccount] TO ADMIN WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_getOneBill] TO ADMIN WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_getOneCategory] TO ADMIN WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_getOneContact] TO ADMIN WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_getOneDetailOrder] TO ADMIN WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_getOneMember] TO ADMIN WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_getOneOrder] TO ADMIN WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_getOneProduct] TO ADMIN WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_getProductBySaleID] TO ADMIN WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_InsertAccount] TO ADMIN WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_InsertBill] TO ADMIN WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_InsertCategory] TO ADMIN WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_InsertContact] TO ADMIN WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_InsertDetailOrder] TO ADMIN WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_InsertMember] TO ADMIN WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_InsertOrder] TO ADMIN WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_InsertOrderWithDetail] TO ADMIN WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_InsertProduct] TO ADMIN WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_UpdateAccount] TO ADMIN WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_UpdateBill] TO ADMIN WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_UpdateCategory] TO ADMIN WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_UpdateContact] TO ADMIN WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_UpdateDetailOrder] TO ADMIN WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_UpdateMember] TO ADMIN WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_UpdateOrder] TO ADMIN WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_UpdateProduct] TO ADMIN WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE, DELETE ON [dbo].[Account] TO ADMIN WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE, DELETE ON [dbo].[Bill] TO ADMIN WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE, DELETE ON [dbo].[Category] TO ADMIN WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE, DELETE ON [dbo].[Contact] TO ADMIN WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE, DELETE ON [dbo].[DetailOrder] TO ADMIN WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE, DELETE ON [dbo].[Member] TO ADMIN WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE, DELETE ON [dbo].[Order] TO ADMIN WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE, DELETE ON [dbo].[Product] TO ADMIN WITH GRANT OPTION  GRANT SELECT ON [dbo].[fn\_HoaDon] TO ADMIN WITH GRANT OPTION  GRANT SELECT ON [dbo].[fn\_ProductAndAmount] TO ADMIN WITH GRANT OPTION  GRANT SELECT ON [dbo].[fn\_ProductByCateName] TO ADMIN WITH GRANT OPTION  GRANT SELECT ON [dbo].[fn\_ProductBySalerName] TO ADMIN WITH GRANT OPTION  GRANT SELECT ON [dbo].[fn\_ProductInRangePrice] TO ADMIN WITH GRANT OPTION  GRANT SELECT ON [dbo].[fn\_SearchMaxPriceByCategory] TO ADMIN WITH GRANT OPTION  GRANT SELECT ON [dbo].[fn\_SearchProductName] TO ADMIN WITH GRANT OPTION |

* + 1. Phân quyền SELLER

|  |
| --- |
| ---Grant SELLER  GRANT SELECT, INSERT, UPDATE, DELETE ON [dbo].[Product] TO SELLER WITH GRANT OPTION  GRANT SELECT, INSERT, DELETE ON [dbo].[Order] TO SELLER WITH GRANT OPTION  GRANT SELECT, INSERT ON [dbo].[DetailOrder] TO SELLER WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE, DELETE ON [dbo].[Member] TO SELLER WITH GRANT OPTION  GRANT UPDATE ON [dbo].[Account] TO SELLER WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE, DELETE ON [dbo].[Bill] TO SELLER WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE, DELETE ON [dbo].[Category] TO SELLER WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE, DELETE ON [dbo].[Contact] TO SELLER WITH GRANT OPTION  GRANT SELECT ON [dbo].[fn\_HoaDon] TO SELLER WITH GRANT OPTION  GRANT SELECT ON [dbo].[fn\_ProductAndAmount] TO SELLER WITH GRANT OPTION  GRANT SELECT ON [dbo].[fn\_ProductByCateName] TO SELLER WITH GRANT OPTION  GRANT SELECT ON [dbo].[fn\_ProductBySalerName] TO SELLER WITH GRANT OPTION  GRANT SELECT ON [dbo].[fn\_ProductInRangePrice] TO SELLER WITH GRANT OPTION  GRANT SELECT ON [dbo].[fn\_SearchMaxPriceByCategory] TO SELLER WITH GRANT OPTION  GRANT SELECT ON [dbo].[fn\_SearchProductName] TO SELLER WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_DeleteBill] TO SELLER WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_DeleteCategory] TO SELLER WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_DeleteContact] TO SELLER WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_DeleteDetailOrder] TO SELLER WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_DeleteMember] TO SELLER WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_DeleteOrder] TO SELLER WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_DeleteProduct] TO SELLER WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_getAllCategory] TO SELLER WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_getAllContact] TO SELLER WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_getAllProduct] TO CUSTOMER WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_getOneDetailOrder] TO SELLER WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_getOneBill] TO SELLER WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_getOneCategory] TO SELLER WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_getOneContact] TO SELLER WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_getOneDetailOrder] TO SELLER WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_getOneMember] TO SELLER WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_getOneOrder] TO SELLER WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_getOneProduct] TO SELLER WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_getProductBySaleID] TO SELLER WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_InsertBill] TO SELLER WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_InsertCategory] TO SELLER WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_InsertContact] TO SELLER WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_InsertDetailOrder] TO SELLER WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_InsertMember] TO SELLER WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_InsertOrder] TO SELLER WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_InsertOrderWithDetail] TO SELLER WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_InsertProduct] TO SELLER WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_UpdateAccount] TO SELLER WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_UpdateBill] TO SELLER WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_UpdateCategory] TO SELLER WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_UpdateContact] TO SELLER WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_UpdateMember] TO SELLER WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_UpdateProduct] TO SELLER WITH GRANT OPTION |

* + 1. Phân quyền CUSTOMER

|  |
| --- |
| ---Grant CUSTOMER  GRANT SELECT, INSERT, UPDATE, DELETE ON [dbo].[Member] TO CUSTOMER WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE ON [dbo].[Account] TO CUSTOMER WITH GRANT OPTION  GRANT SELECT, INSERT ON [dbo].[Order] TO CUSTOMER WITH GRANT OPTION  GRANT SELECT, INSERT ON [dbo].[DetailOrder] TO CUSTOMER WITH GRANT OPTION  GRANT SELECT ON [dbo].[fn\_HoaDon] TO CUSTOMER WITH GRANT OPTION  GRANT SELECT ON [dbo].[fn\_ProductAndAmount] TO CUSTOMER WITH GRANT OPTION  GRANT SELECT ON [dbo].[fn\_ProductByCateName] TO CUSTOMER WITH GRANT OPTION  GRANT SELECT ON [dbo].[fn\_ProductBySalerName] TO CUSTOMER WITH GRANT OPTION  GRANT SELECT ON [dbo].[fn\_ProductInRangePrice] TO CUSTOMER WITH GRANT OPTION  GRANT SELECT ON [dbo].[fn\_SearchMaxPriceByCategory] TO CUSTOMER WITH GRANT OPTION  GRANT SELECT ON [dbo].[fn\_SearchProductName] TO CUSTOMER WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_DeleteDetailOrder] TO CUSTOMER WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_DeleteOrder] TO CUSTOMER WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_getAllCategory] TO CUSTOMER WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_getAllContact] TO CUSTOMER WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_getAllProduct] TO CUSTOMER WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_getOneCategory] TO CUSTOMER WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_getOneContact] TO CUSTOMER WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_getOneDetailOrder] TO CUSTOMER WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_getOneOrder] TO CUSTOMER WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_getOneProduct] TO CUSTOMER WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_InsertAccount] TO CUSTOMER WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_InsertOrder] TO CUSTOMER WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_InsertOrderWithDetail] TO CUSTOMER WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_UpdateAccount] TO CUSTOMER WITH GRANT OPTION |

* 1. **Bảng mô tả phân quyền**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | **ADMIN** | **SELLER** | **CUSTOMER** |
| sp\_Account | [dbo].[sp\_getAllAccount] | **x** |  |  |
| [dbo].[sp\_getOneAccount] | **x** |  |  |
| [dbo].[sp\_DeleteAccount] | **x** |  |  |
| [dbo].[sp\_InsertAccount] | **x** |  | **x** |
| [dbo].[sp\_UpdateAccount] | **x** | **x** | **x** |
| sp\_Bill | [dbo].[sp\_getAllBill] | **x** |  |  |
| [dbo].[sp\_getOneBill] | **x** | **x** |  |
| [dbo].[sp\_DeleteBill] | **x** | **x** |  |
| [dbo].[sp\_InsertBill] | **x** | **x** |  |
| [dbo].[sp\_UpdateBill] | **x** | **x** |  |
| sp\_DetailOrder | [dbo].[sp\_DeleteDetailOrder] | **x** | **x** | **x** |
| [dbo].[sp\_getOneDetailOrder] | **x** | **x** | **x** |
| [dbo].[sp\_InsertDetailOrder] | **x** | **x** |  |
| sp\_UpdateDetailOrder | **x** |  |  |
| [dbo].[sp\_getAllDetailOrder] | **x** |  |  |
| sp\_Category | [dbo].[sp\_DeleteCategory] | **x** | **x** |  |
| [dbo].[sp\_getAllCategory] | **x** | **x** | **x** |
| [dbo].[sp\_getOneCategory] | **x** | **x** |  |
| [dbo].[sp\_InsertCategory] | **x** | **x** |  |
| [dbo].[sp\_UpdateCategory] | **x** | **x** |  |
| sp\_Member | [dbo].[sp\_DeleteMember] | **x** | **x** |  |
| [dbo].[sp\_getAllMember] | **x** |  |  |
| [dbo].[sp\_getOneMember] | **x** | **x** |  |
| [dbo].[sp\_InsertMember] | **x** | **x** |  |
| sp\_UpdateMember | **x** | **x** |  |
| sp\_Order | [dbo].[sp\_DeleteOrder] | **x** | **x** | **x** |
| [dbo].[sp\_getAllOrder] | **x** |  |  |
| [dbo].[sp\_getOneOrder] | **x** | **x** | **x** |
| [dbo].[sp\_InsertOrder] | **x** | **x** | **x** |
| [dbo].[sp\_UpdateOrder] | **x** |  |  |
| sp\_Product | [dbo].[sp\_DeleteProduct] | **x** | **x** |  |
| [dbo].[sp\_getAllProduct] | **x** | **x** | **x** |
| [dbo].[sp\_getOneProduct] | **x** | **x** | **x** |
| [dbo].[sp\_InsertProduct] | **x** | **x** |  |
| [dbo].[sp\_UpdateProduct] | **x** | **x** |  |
| sp\_Contact | [dbo].[sp\_DeleteContact] | **x** | **x** |  |
| [dbo].[sp\_getAllContact] | **x** | **x** | **x** |
| [dbo].[sp\_getOneContact] | **x** | **x** |  |
| [dbo].[sp\_InsertContact] | **x** | **x** |  |
| [dbo].[sp\_UpdateContact] | **x** | **x** |  |
| sp\_Khác | [dbo].[sp\_getProductBySaleID] | **x** | **x** |  |
| [dbo].[sp\_InsertOrderWithDetail] | **x** | **x** | **x** |
| [dbo].[sp\_getLastProduct] | **x** | **x** | **x** |
| USP\_CREATE\_LOGIN\_USER | **x** | **x** | **x** |
| [dbo].[fn\_HoaDon] | | x | x | x |
| [dbo].[fn\_ProductAndAmount] | | x | x | x |
| [dbo].[fn\_ProductByCateName] | | x | x | x |
| [dbo].[fn\_ProductBySalerName] | | x | x | x |
| [dbo].[fn\_ProductInRangePrice] | | x | x | x |
| [dbo].[fn\_SearchMaxPriceByCategory | | x | x | x |
| [dbo].[fn\_SearchProductName] | | x | x | x |
| **[**fn\_LastOrderID**]** | | x | x |  |
| [dbo].[Account] | SELECT | **x** |  | **x** |
| INSERT | **x** |  | **x** |
| UPDATE | **x** | **x** | **x** |
| DELETE | **x** |  |  |
| [dbo].[Bill] | SELECT | **x** | **x** |  |
| INSERT | **x** | **x** |  |
| UPDATE | **x** | **x** |  |
| DELETE | **x** | **x** |  |
| [dbo].[Category] | SELECT | **x** | **x** |  |
| INSERT | **x** | **x** |  |
| UPDATE | **x** | **x** |  |
| DELETE | **x** | **x** |  |
| [dbo].[Contact] | SELECT | **x** | **x** |  |
| INSERT | **x** | **x** |  |
| UPDATE | **x** | **x** |  |
| DELETE | **x** | **x** |  |
| [dbo].[DetailOrder] | SELECT | **x** | **x** | **x** |
| INSERT | **x** | **x** | **x** |
| UPDATE | **x** |  |  |
| DELETE | **x** |  |  |
| [dbo].[Member] | SELECT | **x** | **x** | **x** |
| INSERT | **x** | **x** | **x** |
| UPDATE | **x** | **x** | **x** |
| DELETE | **x** | **x** | **x** |
| [dbo].[Order] | SELECT | **x** | **x** | **x** |
| INSERT | **x** | **x** | **x** |
| UPDATE | **x** |  |  |
| DELETE | **x** | **x** |  |
| [dbo].[Product] | SELECT | **x** | **x** |  |
| INSERT | **x** | **x** |  |
| UPDATE | **x** | **x** |  |
| DELETE | **x** | **x** |  |

1. **Code cài đặt ứng dụng**
3. 1. *Kết nối cơ sở dữ liệu*

|  |
| --- |
| **public** **class** DBContext {  **public** Connection getConnection() **throws** Exception {  String url = "jdbc:sqlserver://" + serverName + ":" + portNumber + "\\" + instance + ";databaseName=" + dbName;  **if** (instance == **null** || instance.trim().isEmpty())  url = "jdbc:sqlserver://" + serverName + ":" + portNumber + ";databaseName=" + dbName;  Class.*forName*("com.microsoft.sqlserver.jdbc.SQLServerDriver");  **return** DriverManager.*getConnection*(url, userID, password);  }  **private** String serverName = "localhost";  **private** String dbName = "Shoe\_DB";  **private** String portNumber = "1433";  **private** String instance = "";// LEAVE THIS ONE EMPTY IF YOUR SQL IS A SINGLE INSTANCE  **private** String userID = "sa";  **private** String password = "123456";  **public** Connection getConnection(String userID, String password) **throws** Exception {  **this**.userID = userID;  **this**.password = password;  **return** **this**.getConnection();  }  } |

* 1. *Tìm kiếm sản phẩm*

|  |
| --- |
| **public** List<Product> searchByName(String txtSearch) {  List<Product> list = **new** ArrayList<>();  String query = " SELECT \* from dbo.fn\_SearchProductName(?)";  **try** {  conn = **new** DBContext().getConnection();  ps = conn.prepareStatement(query);  ps.setString(1, txtSearch);  rs = ps.executeQuery();    **while** (rs.next()) {  list.add(**new** Product(rs.getInt(1),  rs.getString(2),  rs.getString(3),  rs.getDouble(4),  rs.getString(5),  rs.getString(6)));  }  } **catch** (Exception e) {  }  **return** list;  } |

* 1. *Load sản phẩm và phân trang*

|  |
| --- |
| **public** List<Product> pagingProduct(**int** index) {  List<Product> list = **new** ArrayList<>();  **try** {  conn = **new** DBContext().getConnection();//mo ket noi voi sql  CallableStatement command = conn.prepareCall("{call sp\_PagingProduct (?) }");  command.setInt(1, (index-1)\*6);  ResultSet rs = command.executeQuery();  ps = conn.prepareStatement(query);  ps.setInt(1,(index-1)\*6);  rs = ps.executeQuery();  **while** (rs.next()) {    list.add(**new** Product(rs.getInt(1),  rs.getString(2),  rs.getString(3),  rs.getDouble(4),  rs.getString(5),  rs.getString(6)));  }  } **catch** (Exception e) {  }  **return** list;  } |

* 1. *Load và phân trang account trong manage account*

|  |
| --- |
| **public** List<Account> pagingAccount(Account account,**int** index) {  List<Account> list = **new** ArrayList<>();  String query = "select \* from Account\n"  +"ORDER BY uid\n"  +"OFFSET ? ROWS FETCH NEXT 6 ROWS ONLY;";  **try** {  conn = **new** DBContext().getConnection(account.getUser(),account.getPass());  ps = conn.prepareStatement(query);  ps.setInt(1,(index-1)\*6);  rs = ps.executeQuery();  **while** (rs.next()) {  list.add(**new** Account(rs.getInt(1),  rs.getString(2),  rs.getString(3),  rs.getInt(4),  rs.getInt(5)  ));  }  } **catch** (Exception e) {  }  **return** list;  } |

* 1. *Thêm sản phẩm*

|  |
| --- |
| **public** **void** insertProduct(Account account,String name, String image, String price,String title, String description, String category, **int** sid) {  **try** {    conn = **new** DBContext().getConnection(account.getUser(),account.getPass());  CallableStatement stmt = conn.prepareCall("{call sp\_InsertProduct(?,?,?,?,?,?,?) }");  stmt.setString(1, name);  stmt.setString(2, image);  stmt.setString(3, price);  stmt.setString(4, title);  stmt.setString(5, description);  stmt.setString(6, category);  stmt.setInt(7, sid);  stmt.execute();      } **catch** (Exception e) {  }  } |

* 1. *Sửa sản phẩm*

|  |
| --- |
| **public** **void** editProduct(Account account,String name, String image, String price,  String title, String description, String category, String pid, **int** sid) {  **try** {  conn = **new** DBContext().getConnection(account.getUser(),account.getPass());  CallableStatement stmt = conn.prepareCall("{call sp\_UpdateProduct(?,?,?,?,?,?,?,?) }");  stmt.setString(1, pid);  stmt.setString(2, name);  stmt.setString(3, image);  stmt.setString(4, price);  stmt.setString(5, title);  stmt.setString(6, description);  stmt.setString(7, category);  stmt.setInt(8, sid);  stmt.execute();  } **catch** (Exception e) {  }  } |

* 1. *Xóa sản phẩm*

|  |
| --- |
| **public** **void** deleteProduct(Account account,String pid) {  **try** {  conn = **new** DBContext().getConnection(account.getUser(),account.getPass());  CallableStatement stmt = conn.prepareCall("{call sp\_DeleteProduct(?) }");  stmt.setString(1, pid);  stmt.execute();    }  **catch** (Exception e) {  }  } |

* 1. *Load sản phẩm theo category*

|  |
| --- |
| **public** List<Product> getProductByCID(String cid) {  List<Product> list = **new** ArrayList<>();  String query = "select \* from Product\n" +"Where cateID =?";  **try** {  conn = **new** DBContext().getConnection();  ps = conn.prepareStatement(query);  ps.setString(1, cid);  rs = ps.executeQuery();  **while** (rs.next()) {  list.add(**new** Product(rs.getInt(1),  rs.getString(2),  rs.getString(3),  rs.getDouble(4),  rs.getString(5),  rs.getString(6)));  }  } **catch** (Exception e) {  }  **return** list;  } |

* 1. *Xem chi tiết sản phẩm*

|  |
| --- |
| **public** Product getProductByID(String pid) {    String query = "select \* from Product\n" +"Where id =?";  **try** {  conn = **new** DBContext().getConnection();//mo ket noi voi sql  ps = conn.prepareStatement(query);  ps.setString(1, pid);  rs = ps.executeQuery();  **while** (rs.next()) {  **return** **new** Product(rs.getInt(1),  rs.getString(2),  rs.getString(3),  rs.getDouble(4),  rs.getString(5),  rs.getString(6));  }  } **catch** (Exception e) {  }  **return** **null**;  } |

* 1. *Load tất cả danh mục*

|  |
| --- |
| **public** List<Category> getAllCategory() {  List<Category> list = **new** ArrayList<>();  **try** {  conn = **new** DBContext().getConnection();//mo ket noi voi sql  CallableStatement command = conn.prepareCall("{call sp\_getAllCategory }");  ResultSet rs = command.executeQuery();  **while** (rs.next()) {  list.add(**new** Category(rs.getInt(1),  rs.getString(2)));  }  } **catch** (Exception e) {  }  **return** list;  } |

* 1. *Load sản phẩm cuối cùng (mới nhất)*

|  |
| --- |
| **public** Product getLast() {  **try** {  conn = **new** DBContext().getConnection();//mo ket noi voi sql  CallableStatement command = conn.prepareCall("{call sp\_getLastProduct }");  ResultSet rs = command.executeQuery();    **while**(rs.next()){  **return** **new** Product(rs.getInt(1),  rs.getString(2),  rs.getString(3),  rs.getDouble(4),  rs.getString(5),  rs.getString(6));  }  } **catch** (Exception e) {  }  **return** **null**;  } |

* 1. *Sửa Account*

|  |
| --- |
| **public** **void** editAccount(Account account, String user, String pass,  String isSell, String isAdmin,String uid) {    **try** {  conn = **new** DBContext().getConnection(account.getUser(),account.getPass());//mo ket noi voi sql  CallableStatement stmt = conn.prepareCall("{call sp\_UpdateAccount(?,?,?,?,?) }");  stmt.setString(1, uid);  stmt.setString(2, user);  stmt.setString(3, pass);  stmt.setString(4, isSell);  stmt.setString(5, isAdmin);  stmt.execute();  } **catch** (Exception e) {  }  } |

* 1. *Thêm Account*

|  |
| --- |
| **public** **void** insertAccount(Account account,String user, String pass, String isSell,  String isAdmin) {    **try** {  **if**(account.getIsAdmin() == 2) {  conn = **new** DBContext().getConnection();//account.getUser(),account.getPass());//mo ket noi voi sql  }  conn = **new** DBContext().getConnection(account.getUser(),account.getPass());  CallableStatement stmt = conn.prepareCall("{call sp\_InsertAccount(?,?,?,?) }");  stmt.setString(1, user);  stmt.setString(2, pass);  stmt.setString(3, isSell);  stmt.setString(4, isAdmin);    stmt.execute();  } **catch** (Exception e) {  e.printStackTrace();  }  } |

* 1. *Xoá Account*

|  |
| --- |
| **public** **void** deleteAccount(Account account,String uid) {  **try** {  conn = **new** DBContext().getConnection(account.getUser(),account.getPass());  CallableStatement stmt = conn.prepareCall("{call sp\_DeleteAccount(?) }");  stmt.setString(1, uid);  stmt.execute();    }  **catch** (Exception e) {  }  } |

* 1. *Đăng nhập*

|  |
| --- |
| **public** Account login(String user,String pass) {  String query = "select \* from Account\n" +"Where [user] = ?\n"+"and pass = ?";  **try** {  conn = **new** DBContext().getConnection();//mo ket noi voi sql  ps = conn.prepareStatement(query);  ps.setString(1,user);  ps.setString(2,pass);  rs = ps.executeQuery();  **while** (rs.next()) {  **return** **new** Account(rs.getInt(1),  rs.getString(2),  rs.getString(3),  rs.getInt(4),  rs.getInt(5));    }  }  **catch** (Exception e) {  }  **return** **null**;    } |

* 1. *Kiểm tra tài khoản*

|  |
| --- |
| **public** Account checkAccountExist(String user) {  String query = "select \* from Account\n" +"Where [user] = ?";  **try** {  conn = **new** DBContext().getConnection();//mo ket noi voi sql  ps = conn.prepareStatement(query);  ps.setString(1,user);  rs = ps.executeQuery();  **while** (rs.next()) {  **return** **new** Account(rs.getInt(1),  rs.getString(2),  rs.getString(3),  rs.getInt(4),  rs.getInt(5));    }  }  **catch** (Exception e) {  }  **return** **null**;    } |

* 1. *Đổi mật khẩu*

|  |
| --- |
| **public** **void** saveOrder(String uid, List<Product> list) {  **try** {  conn = **new** DBContext().getConnection();//mo ket noi voi sql    //1.add Order to database  CallableStatement stmt = conn.prepareCall("{call sp\_InsertOrderWithDetail (?) }");  stmt.setInt(1,Integer.*parseInt*(uid));  stmt.execute();    //2.lấy oid Order vừa tạo  **int** lastID = *getLastOrderID*();    //3.add DetailOrder  **for**(Product p: list){  insertDetailOrder(String.*valueOf*(lastID),String.*valueOf*(p.getId()), String.*valueOf*(p.getAmount()));  }  } **catch** (Exception e) {  }    } |

* 1. *Thêm thông tin sản phẩm vào bảng Order và Detail Order*

|  |
| --- |
| **public** **void** saveOrder(String uid, List<Product> list) {  **try** {  conn = **new** DBContext().getConnection();//mo ket noi voi sql    //1.add Order to database  CallableStatement stmt = conn.prepareCall("{call sp\_InsertOrderWithDetail (?) }");  stmt.setInt(1,Integer.*parseInt*(uid));  stmt.execute();    //2.lấy oid Order vừa tạo  **int** lastID = *getLastOrderID*();    //3.add DetailOrder  **for**(Product p: list){  insertDetailOrder(String.*valueOf*(lastID),String.*valueOf*(p.getId()), String.*valueOf*(p.getAmount()));  }  } **catch** (Exception e) {  }    } |

* 1. *Lấy oid Order vừa tạo*

|  |
| --- |
| **public** **static** **int** getLastOrderID() {  **try** {  Connection conn = **new** DBContext().getConnection();//mo ket noi voi sql  //2.lấy oid Order vừa tạo  CallableStatement stmt = conn.prepareCall("{?= call fn\_LastOrderID()}");  stmt.registerOutParameter(1,Types.***INTEGER***);  stmt.execute();  **return** stmt.getInt(1);    } **catch** (Exception e) {  e.getStackTrace();  }  **return** -1;  } |

* 1. *Lấy ra Order vừa tạo*

|  |
| --- |
| **public** Order getLastOrder(){  Order order = **new** Order();  **try** {  Connection conn = **new** DBContext().getConnection();//mo ket noi voi sql  CallableStatement stmt = conn.prepareCall("{call sp\_getOneOrder(?) }");  **int** lastID = *getLastOrderID*();  stmt.setInt(1,lastID);  rs = stmt.executeQuery();  **while** (rs.next()) {  **return** **new** Order(rs.getInt(1),  rs.getInt(2),  rs.getInt(3),  rs.getDouble(4),  rs.getDate(5),  rs.getString(6),  rs.getString(7),  rs.getString(8),  rs.getString(9));  }  }**catch** (Exception e) {  e.getStackTrace();  }  **return** **null**;  } |

* 1. *Xem thông tin trong view ThongTinSanPham*

|  |
| --- |
| **public** List<Product> getViewProduct() {  List<Product> list = **new** ArrayList<>();  String query = "select \* from ThongTinSanPham";  **try** {  conn = **new** DBContext().getConnection();//mo ket noi voi sql  ps = conn.prepareStatement(query);  rs = ps.executeQuery();  **while** (rs.next()) {  list.add(**new** Product(rs.getString(1),  rs.getString(2),  rs.getDouble(3),  rs.getString(4)  ));  }  } **catch** (Exception e) {  }  **return** list;  } |

* 1. *Lấy những product mà do người đó bán để load lên trong manage product*

|  |
| --- |
| **public** List<Product> getProductBySaleID(Account account,**int** id) {  List<Product> list = **new** ArrayList<>();  **try** {  conn = **new** DBContext().getConnection(account.getUser(),account.getPass());  CallableStatement stmt = conn.prepareCall("{call sp\_getProductBySaleID(?) }");  stmt.setInt(1,id);  rs = stmt.executeQuery();  **while** (rs.next()) {  list.add(**new** Product(rs.getInt(1),  rs.getString(2),  rs.getString(3),  rs.getDouble(4),  rs.getString(5),  rs.getString(6)));  }  } **catch** (Exception e) {  }  **return** list;  } |

* 1. *Đăng ký tài khoản*

|  |
| --- |
| **public** **void** signup(String user, String pass) {  **try** {  conn = **new** DBContext().getConnection();//mo ket noi voi sql  CallableStatement stmt = conn.prepareCall("{call sp\_InsertAccount(?,?,0,0) }");  stmt.setString(1,user);  stmt.setString(2,pass);  stmt.execute();    }  **catch** (Exception e) {  }    } |

1. **Hướng dẫn sử dụng**
   1. **Cài đặt chương trình**

* Bước 1: Restore Database
  + Chạy các file \*.sql theo thứ tự sau
    - 01\_TaoBang.sql
    - 02\_role.sql
    - 03\_sp\_fn\_tg\_vw.sql
    - 04\_data.sql
* Bước 2: Khởi chạy project
  + Click chuột phải vào project chọn debug để khởi động project bằng Eclipse 2021
  + Chạy phần mềm: các tài khoản đăng nhập:
    - Tên đăng nhập: admin1 ; Mật khẩu 123
  1. **Chi tiết**

Cài đặt

Yêu cầu về phần mềm:

Sử dụng Eclipse

SQL Server phiên bản từ 2012 trở về sau.

Máy tính của bạn phải được cài JDK và một server tương thích và thêm nó vào Eclipse (ở đây mình chọn Tomcat v8.5) bằng cách

Bạn mở Eclipse lên, chọn Window –> Preferences –> Server –> Runtime Environment –> Add –> Chọn loại server sau đó thêm đường dẫn và kết thúc ( Ở đây mình chọn là Tomcat v8.5 và mình để Tomcat tại ổ D, lưu ý là server mà bạn tải về chỉ là bản zip, không cần cài đặt, chỉ cần giải nén và đặt tại một thư mục bất kì và chọn đường dẫn của server tới nơi mà bạn để thư mục là được).

Graphical user interface, application, table

Description automatically generated

Sau khi restore database bằng file Script.sql hoặc file Backup.bak ta khởi động Eclipse , chọn file -> open Projects from File System để mở project

Graphical user interface, text, application, email

Description automatically generated

Click chuột phải vào thư mục chứa project chọn Debug As -> Debug on Server

Graphical user interface, text, application

Description automatically generated

Sau đó chọn server tomcat v8.5 -> next -> Finish để chạy chương trình

Graphical user interface, text, application, email

Description automatically generated

Ta tiến hành đăng kí một tài khoản khoản để đăng nhập hoặc có thể tạo một khoản từ database trong table Account

Graphical user interface, application

Description automatically generated

Tiến hành trải nghiệm website thôi nào !

1. **Chức năng**

**Mua Hàng**

Click vào sản phẩm muốn mua và chọn add to cart để thêm vào giỏ hàng

**Graphical user interface, application, Word

Description automatically generated**

Có thể chỉnh số lượng muốn mua hoặc xóa sản phẩm khỏi giỏ hàng

**Graphical user interface, application

Description automatically generated**

Nhấn mua hàng để thanh toán

**Graphical user interface, text, application

Description automatically generated**

**Graphical user interface, text, application, email

Description automatically generated**

**Thêm thông tin người dùng**

Nhấn vào Hello để thêm thông tin người dùng sau đó mới tiến hành mua hàng.

**Graphical user interface

Description automatically generated**

**Graphical user interface, application

Description automatically generated**

**Tìm kiếm sản phẩm**

Gõ từ khóa tìm kiếm vào ô input này, sau đó click vào nút search để tìm kiếm.

Timeline

Description automatically generated with medium confidence

Trang web sẽ hiện ra một list sản phẩm mà trong tên có chứa từ khóa mà bạn tìm kiếm

Website

Description automatically generated

**Đổi** **mật khẩu**

Click vào Change Password và tiến hành nhập mật khẩu hiện tại và mật khẩu mới để đổi mật khẩu

**Graphical user interface

Description automatically generated**

**Graphical user interface, text, application

Description automatically generated**

**Tìm sản phẩm theo danh mục**

Click vào tên danh mục muốn tìm để tìm kiếm

**Graphical user interface, application

Description automatically generated**

**Chat với người bán hàng**

Khách hàng có câu hỏi thắc mắc nào có thể click vào biểu tượng tawk.to để hỏi và bạn sẽ được phản hồi từ người bán trong thời gian sớm nhất

**Graphical user interface, application, website

Description automatically generated**

**Liên hệ**

Bạn có bất cứ phàn nàn hay những nhận xét , góp ý gì cho website chúng tôi thì click vào contact để thực hiện nhé .

Graphical user interface, application

Description automatically generated

**Quản lý tài khoản (chức năng của Admin)**

Admin có thể thêm, sửa xóa tài khoản bằng cách click vào manage account .

Chọn add new account để thêm tài khoản mới

Click vào biểu tượng thùng rác để xóa tài khoản

Click vào biểu tượng cây bút để chỉnh sửa tài khoản

**Application, table

Description automatically generated**

**Quản lý Sản Phẩm (chức năng của người bán)**

Người bán có thể thêm, sửa xóa sản phẩm của mình bán bằng cách click vào manage product .

Chọn add new product để thêm sản phẩm mới

Click vào biểu tượng thùng rác để xóa sản phẩm

Click vào biểu tượng cây bút để chỉnh sửa sản phẩm

Graphical user interface, application, website

Description automatically generated

**Graphical user interface, text, application, email

Description automatically generated**

**Graphical user interface, application, Teams

Description automatically generated**

1. **Kết luận**
   1. **Các phần đã đạt được trong đồ án**

Thông qua đồ án cuối kỳ “Xây dựng website kinh doanh giày và phụ kiện” nhóm chúng em đã đạt được những mục tiêu, chức năng ban đầu đề ra. Và điều đặt biệt là đã vận dụng tốt những kiến thức của môn Hệ quản trị cơ sở dữ liệu để thực hiện đề tài này:

* Tra cứu, tìm kiếm thông tin sản phẩm một cách nhanh chóng và chính xác.
* Thêm, sửa, xóa sản phẩm của người bán hàng.
* Hiển thị danh sách các sản phẩm theo thương hiệu và thông tin chi tiết của từng sản phẩm
* Chức năng quản lý thông tin cá nhân.
* Chức năng đặt hàng.
* Chức năng thêm, xóa, sửa sản phẩm trong giỏ hàng
* Chức năng phân quyền Admin , người mua, người bán.
* Đăng nhập , đăng ký tài khoản để truy cập vào website.

Bên cạnh đó, nhóm chúng em còn đạt được như:

Giao diện đẹp mắt, thân thiện với người dùng.

Code đơn giản, tinh gọn nhưng đầy đủ các tính năng.

Làm việc nhóm ăn ý giữa các thành viên.

Thực hiện đúng kế hoạch và tiến độ ban đầu đề ra.

* 1. **Một số hạn chế**

Một số kiến thức vẫn còn khá mới mẻ nên việc tìm hiểu để vận dụng vào đồ án cũng gặp nhiều khó khăn và tốn thời gian.

* 1. **Cam kết**

Chúng em xin cam đoan đồ án ***“Xây dựng website kinh doanh giày và phụ kiện”*** là thành quả của quá trình học tập và tìm hiểu dưới sự hướng dẫn của thầy Nguyễn Thành Sơn. Với lòng biết ơn sâu sắc và chân thành nhất, chúng em xin gửi đến thầy đã cùng với tri thức và tâm huyết của mình, truyền đạt vốn kiến thức quý báu cho chúng em trong suốt học kỳ này. Với điều kiện thời gian cũng như kinh nghiệm còn hạn chế, bài báo cáo này không thể tránh được những thiếu sót. Chúng em rất mong nhận được sự chỉ bảo, đóng góp ý kiến của thầy để chúng em có điều kiện bổ sung, nâng cao ý thức của mình, phục vụ tốt hơn công tác thực tế sau này. Mọi sự tham khảo đều có trích nguồn. Nếu có sự vi phạm nào chúng em xin hoàn toàn chịu trách nhiệm.