**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

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

**KHOA CÔNG NGHỆ THÔNG TIN**

🙤🟑🟊🟑🙦

****

**MÔN HỌC: HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

**BÁO CÁO CUỐI KỲ**

**ĐỀ TÀI**

**PHẦN MỀM ĐẶT HÀNG TRÀ SỮA**

**GVHD: TS.NGUYỄN THÀNH SƠN**

**SVTH: MSSV**

**Nguyễn Ngọc Hoàng Vy 16110530**

**Lê Thiện Đức 15110192**

**TP. Hồ Chí Minh, tháng 05 năm 2018**

**MỤC LỤC**

[**1.** **Tổng Quan Về Phần Mềm** 4](#_Toc514018271)

[**1.1.** **Thông Tin Cơ Bản** 4](#_Toc514018272)

[**1.2.** **Mô Tả** 4](#_Toc514018273)

[**1.3.** **Các chức năng chính** 5](#_Toc514018274)

[**2.** **Lược đồ quan hệ** 5](#_Toc514018275)

[**3.** **Danh Sách Các Table, Stored procedure, Function, Trigger.** 6](#_Toc514018276)

[**3.1.** **Danh sách các Table** 6](#_Toc514018277)

[**3.2.** **Danh sách các View** 7](#_Toc514018278)

[**3.3.** **Danh sách các Stored Procedure** 7](#_Toc514018279)

[**3.4.** **Danh sách các Function** 8](#_Toc514018280)

[**3.5.** **Danh sách các Trigger** 8](#_Toc514018281)

[**4.** **Câu lệnh SQL** 10](#_Toc514018282)

[**4.1.** **Table** 10](#_Toc514018283)

[**4.1.1.** **LoaiThucUong** 10](#_Toc514018284)

[**4.1.2.** **ThucUong** 10](#_Toc514018285)

[**4.1.3.** **Them** 10](#_Toc514018286)

[**4.1.4.** **KhachHang** 11](#_Toc514018287)

[**4.1.5.** **HoaDon** 12](#_Toc514018288)

[**4.1.6.** **Orders** 12](#_Toc514018289)

[**4.1.7.** **ChiTietThem** 12](#_Toc514018290)

[**4.1.8.** **GioHang** 13](#_Toc514018291)

[**4.1.9.** **ChiTietHoaDon** 13](#_Toc514018292)

[**4.2.** **View** 14](#_Toc514018293)

[**4.2.1.** **V\_LOAITHUCUONG** 14](#_Toc514018294)

[**4.2.2.** **V\_THUCUONG** 14](#_Toc514018295)

[**4.2.3.** **V\_THEM** 14](#_Toc514018296)

[**4.2.4.** **V\_KHACHHANG** 14](#_Toc514018297)

[**4.2.5.** **V\_HOADON** 14](#_Toc514018298)

[**4.2.6.** **V\_ORDERS** 14](#_Toc514018299)

[**4.2.7.** **V\_CHITIETHOADON** 14](#_Toc514018300)

[**4.2.8.** **V\_CHITIETTHEM** 14](#_Toc514018301)

[**4.2.9.** **V\_GIOHANG** 14](#_Toc514018302)

[**4.3.** **Stored Procedure** 14](#_Toc514018303)

[**4.3.1.** **sp\_insertLoaiThucUong** 14](#_Toc514018304)

[**4.3.2.** **sp\_deleteLoaiThucuong** 15](#_Toc514018305)

[**4.3.3.** **sp\_updateLoaiThucUong** 15](#_Toc514018306)

[**4.3.4.** **sp\_insertThucUong** 16](#_Toc514018307)

[**4.3.5.** **sp\_deleteThucUong** 16](#_Toc514018308)

[**4.3.6.** **sp\_updateThucUong** 16](#_Toc514018309)

[**4.3.7.** **sp\_insertThem** 17](#_Toc514018310)

[**4.3.8.** **sp\_deleteThem** 17](#_Toc514018311)

[**4.3.9.** **sp\_updateThem** 17](#_Toc514018312)

[**4.3.10.** **sp\_insertKhachHang** 17](#_Toc514018313)

[**4.3.11.** **sp\_deleteKhachHang** 18](#_Toc514018314)

[**4.3.12.** **sp\_updateKhachHang** 18](#_Toc514018315)

[**4.3.13.** **sp\_resetPassword** 18](#_Toc514018316)

[**4.3.14.** **sp\_insertBill** 19](#_Toc514018317)

[**4.3.15.** **sp\_cancelBill** 19](#_Toc514018318)

[**4.3.16.** **sp\_payedBill** 20](#_Toc514018319)

[**4.3.17.** **sp\_checkAllBill** 20](#_Toc514018320)

[**4.4.** **Trigger** 21](#_Toc514018321)

[**4.4.1.** **tg\_addLoaiThucUong** 21](#_Toc514018322)

[**4.4.2.** **tg\_updateLoaiThucUong** 21](#_Toc514018323)

[**4.4.3.** **tg\_addThucUong** 22](#_Toc514018324)

[**4.4.4.** **tg\_updateThucUong** 23](#_Toc514018325)

[**4.4.5.** **tg\_addTHEM** 23](#_Toc514018326)

[**4.4.6.** **tg\_updateTHEM** 24](#_Toc514018327)

[**4.4.7.** **tg\_addKHACHHANG** 25](#_Toc514018328)

[**4.4.8.** **tg\_updateKHACHHANG** 25](#_Toc514018329)

[**4.5.** **Function** 26](#_Toc514018330)

[**4.5.1.** **fn\_Login** 26](#_Toc514018331)

[**4.5.2.** **fn\_ThongKeHoaDon** 27](#_Toc514018332)

[**4.5.3.** **fn\_getOrderByUserId** 28](#_Toc514018333)

[**4.5.4.** **fn\_getListDrink** 29](#_Toc514018334)

[**4.5.5.** **fn\_getListBillOrdering** 31](#_Toc514018335)

[**4.5.6.** **fn\_getListBillCanceled** 32](#_Toc514018336)

[**5.** **Phân quyền** 33](#_Toc514018337)

[**5.1.** **Nhóm quyền Adminitrator** 33](#_Toc514018338)

[**5.2.** **Nhóm quyền Customer** 34](#_Toc514018339)

1. **Tổng Quan Về Phần Mềm**
   1. **Thông Tin Cơ Bản**

Tên phần mềm: Phần mềm đặt hàng trà sữa

Ngôn ngữ: C#

Phần mềm hỗ trợ: Devexpress 17.1

Tài Khoản Đăng Nhập Database:

- Quyền Admin:

- Username: tshoangvy, password: 12345

- Quyền Customer:

- Username: tsannguyen, password: 12345

- Username: tsmainguyen, password: 123

- Username: tsnamle, password: 123123

- Database: orcl

* 1. **Mô Tả**

Để sử dụng phần mềm, người dùng cần có tài khoản để đăng nhập vào ứng dụng, Mỗi tài khoảng sẽ có username , password để đăng nhập vào ứng dụng Ngoài ra User được chia thành 2 loại quyền đó là Admin và Customer. Admin được phép truy cập đầy đủ toàn bộ các chức năng của phần mềm.

Về mặt thức uống, quán sẽ có nhiều chủng loại thức uống và mỗi chủng loại ấy sẽ có nhiều thức uống để khách hàng có thể dễ dàng lựa chọn. Ngoài ra khách có thể chọn thêm nhiều loại topping để thêm vào ly nước. Mỗi loại thức uống sẽ được phân biệt với nhau bằng mã, mỗi mã là mỗi loại khác nhau, và mỗi mã sẽ có một tên gọi để gọi tên cho các loại thức uống. Mỗi thức uống sẽ được phân biệt với nhau bằng mã, mỗi mã là mỗi thức uống khác nhau, và mỗi mã sẽ có một tên gọi để gọi tên cho các thức uống. Mỗi topping thêm sẽ được phân biệt với nhau bằng mã, mỗi mã là mỗi topping khác nhau, và mỗi mã sẽ có một tên gọi để gọi tên cho các topping.

Khách hàng khi mua sẽ order thức uống và nhiều loại topping khác nhau vào trong một ly nước. Các order của khách sẽ được bỏ vào trong giỏ hàng của mình , khách cũng có thể xóa các order ra khỏi giỏ hàng. Mỗi order có mã phân biệt để truy xuất đến thức uống và topping mà khách hàng đã order tại quán. Khi thanh toán khách hàng sẽ nhận được hóa đơn thanh toán với, thức uống đã order tại quán và số tiền cần phải trả. Mỗi hóa đơn thanh toán sẽ có mã để phân biệt, ngày giờ thanh toán và trạng thái của hóa đơn là đã thanh toán hay đã hủy hay đã đặt, mỗi chi tiết hóa đơn giúp ta biết được order và số lượng của từng hóa đơn riêng biệt. Sau khi thanh toán, khách hàng có thể xem lại những hóa đơn cũ của mình.

* 1. **Các chức năng chính**

**-** Chọn loại thức uống, thức uống, topping thêm, số lượng.

- Đặt hàng, tạo hóa đơn bán hàng.

- Thanh toán hóa đơn bán hàng.

- Tra cứu doanh thu theo tháng.

- Thêm, xóa sửa loại thức uống, thức uống, topping.

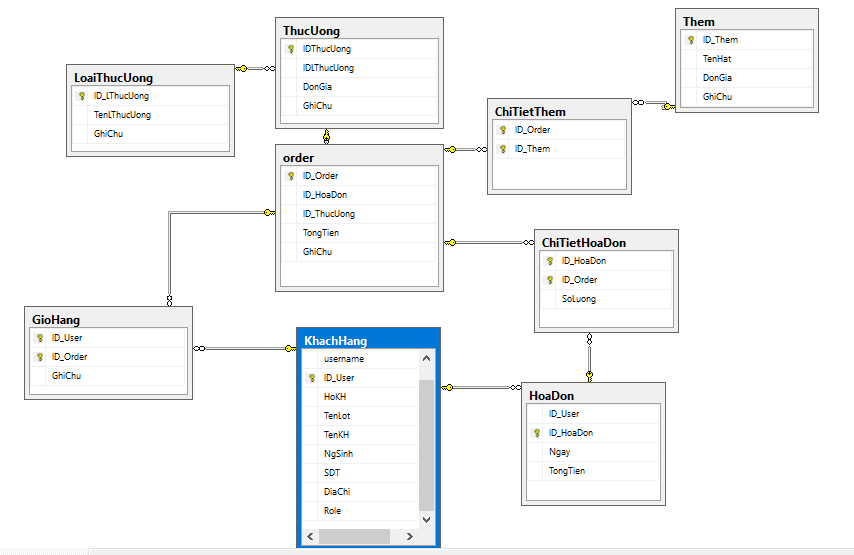
- Quản lý khách hàng:

+ Xem danh sách khách hàng.

+ Xóa, Sửa thông tin khách hàng.

1. **Lược đồ quan hệ**

* **LoaiThucUong**(id, tenLThucUong, ghiChu)
* **ThucUong**(id, tenThucUong, idLThucUong, donGia,ghiChu)
* **Them**(id, tenHat, donGia,ghiChu)
* **KhachHang**(id, userName, password, hoTen, ngSinh, sdt, diaChi, role)
* **HoaDon**(id, idUser, ngay, tongTien, tinhTrang)
* **Orders**(id, idThucUong, tongTien, ghiChu)
* **ChiTietThem**(idOrder, idThem)
* **GioHang** (idUser, idOrder, ghiChu)
* **ChiTietHoaDon** (idHoaDon, idOrder, soLuong)

****

1. **Danh Sách Các Table, Stored procedure, Function, Trigger.**
   1. **Danh sách các Table**

|  |  |  |
| --- | --- | --- |
| **STT** | **TÊN TABLE** | **MÔ TẢ** |
| 1 | LoaiThucUong | Danh sách loại thức uống |
| 2 | ThucUong | Danh sách thức uống |
| 3 | Them | Danh các món toping |
| 4 | KhachHang | Danh sách thông tin khách hàng |
| 5 | HoaDon | Danh sách các hóa đơn |
| 6 | Orders | Các món khách hàng orders trong 1 hóa đơn |
| 7 | ChiTietThem | Chi tiết các món toping |
| 8 | GioHang | Danh sách các món đã lựa chọn nhưng chưa dặt hàng. |
| 9 | ChiTietHoaDon | Danh sách, số lượng các món đã orders ứng với mỗi hóa đơn. |

* 1. **Danh sách các View**

|  |  |  |
| --- | --- | --- |
| **STT** | **TÊN VIEW** | **MÔ TẢ** |
| 1 | v\_loaithucuong | Khung nhìn loại thức uống. |
| 2 | v\_thucuong | Khung nhìn thức uống. |
| 3 | v\_them | Khung nhìn của toping |
| 4 | v\_khachhang | Khung nhìn khách hàng |
| 5 | v\_hoadon | Khung nhìn hóa đơn |
| 6 | v\_orders | Khung nhìn những món được orders |
| 7 | v\_chitiethoadon | Khung nhìn chi tiết hóa đơn của từng orders ứng với từng hóa đơn. |
| 8 | v\_chitietthem | Khung nhìn chi tiết thêm |
| 9 | v\_giohang | Khung nhìn các món trong giỏ hàng. |

* 1. **Danh sách các Stored Procedure**

|  |  |  |
| --- | --- | --- |
| **STT** | **TÊN STORED PROCEDURE** | **MÔ TẢ** |
| 1 | sp\_insertLoaiThucUong | Thêm loại thức uống |
| 2 | sp\_deleteLoaiThucuong | Xóa loại thức uống |
| 3 | sp\_updateLoaiThucUong | Cập nhật thông tin loại thức uống |
| 4 | sp\_insertThucUong | Thêm thức uống |
| 5 | sp\_deleteThucUong | Xóa thức uống |
| 6 | sp\_updateThucUong | Cập nhật thông tin thức uống |
| 7 | sp\_insertThem | Thêm Topping |
| 8 | sp\_deleteThem | Xóa topping |
| 9 | sp\_updateThem | Cập nhật thông tin topping |
| 10 | sp\_insertKhachHang | Thêm tài khoản |
| 11 | sp\_deleteKhachHang | Xóa tài khoảng |
| 12 | sp\_updateKhachHang | Cập nhật thông tin tài khoảng |
| 13 | sp\_resetPassword | Đặt lại mật khẩu |
| 14 | sp\_insertBill | Thêm hóa đơn |
| 15 | sp\_cancelBill | Hủy hóa đơn |
| 16 | sp\_payedBill | Thanh toán cho hóa đơn |
| 17 | sp\_checkAllBill | Kiểm tra hóa đơn |

* 1. **Danh sách các Function**

|  |  |  |
| --- | --- | --- |
| **STT** | **TÊN FUNCTION** | **MÔ TẢ** |
| 1 | fn\_Login | Đăng nhập tài khoảng |
| 2 | fn\_ThongKeHoaDon | Thống kê hóa đơn |
| 3 | fn\_getOrderByUserId | Xem order của mỗi khách hàng |
| 4 | fn\_getListDrink | Xem danh sách thức uống |
| 5 | fn\_getListBillOrdering | Xem danh sách hóa đơn đã đặt |
| 6 | fn\_getListBillCanceled | Xem danh sách hóa đơn đã hủy |

* 1. **Danh sách các Trigger**

|  |  |  |
| --- | --- | --- |
| **STT** | **TÊN TRIGGER** | **MÔ TẢ** |
| 1 | tg\_addLoaiThucUong | Thêm một loại thức uống |
| 2 | tg\_updateLoaiThucUong | Cập nhật một loại thức uống |
| 3 | tg\_addThucUong | Thêm thức uống |
| 4 | tg\_updateThucUong | Cập nhật thức uống |
| 5 | tg\_addTHEM | Thêm toppping |
| 6 | tg\_updateTHEM | Cập nhật một loại topping |
| 7 | tg\_addKHACHHANG | Đăng ký tài khoảng |
| 8 | tg\_updateKHACHHANG | Sửa đổi thông tin tài khoảng |

1. **Câu lệnh SQL**
   1. **Table**
      1. **LoaiThucUong**

CREATE TABLE LOAITHUCUONG(

id int GENERATED ALWAYS as IDENTITY(START with 1 INCREMENT by 1) PRIMARY KEY NOT NULL,

tenLThucUong NVARCHAR2(50),

ghichu NVARCHAR2(50)

);

* + 1. **ThucUong**

CREATE TABLE THUCUONG(

id int GENERATED ALWAYS as IDENTITY(START with 1 INCREMENT by 1) PRIMARY KEY NOT NULL,

tenThucUong NVARCHAR2(50),

idLThucUong int,

donGia int,

ghiChu NVARCHAR2(50),

FOREIGN KEY (idLThucUong) REFERENCES LOAITHUCUONG(id)

);

* + 1. **Them**

CREATE TABLE THEM(

id INT GENERATED ALWAYS as IDENTITY(START with 1 INCREMENT by 1) PRIMARY KEY NOT NULL,

TENHAT NVARCHAR2(50),

DONGIA INT,

GHICHU NVARCHAR2(50)

);

* + 1. **KhachHang**

CREATE TABLE KHACHHANG(

ID INT GENERATED ALWAYS as IDENTITY(START with 1 INCREMENT by 1)

PRIMARY KEY NOT NULL,

USERNAME NVARCHAR2(50) NOT NULL,

PASSWORD VARCHAR2(50) NOT NULL,

HOTEN NVARCHAR2(50),

NGSINH DATE ,

SDT VARCHAR2(11),

DIACHI NVARCHAR2(50),

ROLE INT

);

* + 1. **HoaDon**

CREATE TABLE HOADON(

ID INT GENERATED ALWAYS as IDENTITY(START with 1 INCREMENT by 1) PRIMARY KEY NOT NULL,

IDUSER INT ,

NGAY DATE,

TONGTIEN INT,

TINHTRANG NVARCHAR2(50)

);

* + 1. **Orders**

CREATE TABLE ORDERS(

ID INT GENERATED ALWAYS as IDENTITY(START with 1 INCREMENT by 1)

PRIMARY KEY NOT NULL,

IDTHUCUONG INT NOT NULL,

TONGTIEN INT NOT NULL,

GHICHU NVARCHAR2(50),

FOREIGN KEY (IDTHUCUONG) REFERENCES THUCUONG(id)

);

* + 1. **ChiTietThem**

CREATE TABLE CHITIETTHEM(

IDORDER INT NOT NULL,

IDTHEM INT NOT NULL,

SOLUONG INT,

PRIMARY KEY (IDORDER,IDTHEM),

FOREIGN KEY (IDTHEM) REFERENCES THEM(ID),

FOREIGN KEY (IDORDER) REFERENCES ORDERS(ID)

);

* + 1. **GioHang**

CREATE TABLE GIOHANG(

IDUSER INT NOT NULL,

IDORDER INT NOT NULL,

SOLUONG INT NOT NULL,

GHICHU NVARCHAR2(50),

PRIMARY KEY (IDUSER,IDORDER),

FOREIGN KEY (IDUSER) REFERENCES KHACHHANG(ID),

FOREIGN KEY (IDORDER) REFERENCES ORDERS(ID)

);

* + 1. **ChiTietHoaDon**

CREATE TABLE CHITIETHOADON(

IDHOADON INT NOT NULL,

IDORDER INT NOT NULL,

SOLUONG INT NOT NULL,

PRIMARY KEY (IDHOADON,IDORDER),

FOREIGN KEY (IDHOADON) REFERENCES HOADON(ID),

FOREIGN KEY (IDORDER) REFERENCES ORDERS(ID)

);

* 1. **View**
     1. **V\_LOAITHUCUONG**

|  |
| --- |
| CREATE OR REPLACE VIEW V\_LOAITHUCUONG AS SELECT \* FROM LOAITHUCUONG**;** |

* + 1. **V\_THUCUONG**

|  |
| --- |
| CREATE OR REPLACE VIEW V\_THUCUONG AS SELECT \* FROM THUCUONG; |

* + 1. **V\_THEM**

|  |
| --- |
| CREATE OR REPLACE VIEW V\_THEM AS SELECT \* FROM THEM; |

* + 1. **V\_KHACHHANG**

|  |
| --- |
| CREATE OR REPLACE VIEW V\_KHACHHANG AS SELECT \* FROM KHACHHANG; |

* + 1. **V\_HOADON**

|  |
| --- |
| CREATE OR REPLACE VIEW V\_HOADON AS SELECT \* FROM HOADON; |

* + 1. **V\_ORDERS**

|  |
| --- |
| CREATE OR REPLACE VIEW V\_ORDERS AS SELECT \* FROM ORDERS; |

* + 1. **V\_CHITIETHOADON**

|  |
| --- |
| CREATE OR REPLACE VIEW V\_CHITIETHOADON AS SELECT \* FROM CHITIETHOADON; |

* + 1. **V\_CHITIETTHEM**

|  |
| --- |
| CREATE OR REPLACE VIEW V\_CHITIETTHEM AS SELECT \* FROM CHITIETTHEM; |

* + 1. **V\_GIOHANG**

|  |
| --- |
| CREATE OR REPLACE VIEW V\_GIOHANG AS SELECT \* FROM GIOHANG; |

* 1. **Stored Procedure**
     1. **sp\_insertLoaiThucUong**

|  |
| --- |
| CREATE OR REPLACE PROCEDURE sp\_insertLoaiThucUong( TenLThucUong$ NVARCHAR2, GHICHU$ NVARCHAR2)  AS  BEGIN  INSERT INTO v\_LOAITHUCUONG(tenLThucUong,ghichu) VALUES (TenLThucUong$,GHICHU$);  END; |

* + 1. **sp\_deleteLoaiThucuong**

|  |
| --- |
| CREATE OR REPLACE PROCEDURE sp\_deleteLoaiThucuong( ID$ int)  AS  IDTHUCUONG$ INT;  CURSOR cus IS SELECT ID FROM THUCUONG WHERE idLThucUong=ID$;  BEGIN  OPEN cus;  LOOP  FETCH cus INTO IDTHUCUONG$;  EXIT WHEN cus%NOTFOUND;  DELETE FROM GIOHANG WHERE IDORDER IN (SELECT ID FROM ORDERS WHERE IDTHUCUONG=IDTHUCUONG$);  DELETE FROM CHITIETTHEM WHERE IDORDER IN (SELECT ID FROM ORDERS WHERE IDTHUCUONG=IDTHUCUONG$);  DELETE FROM CHITIETHOADON WHERE IDORDER IN (SELECT ID FROM ORDERS WHERE IDTHUCUONG=IDTHUCUONG$);  DELETE FROM ORDERS WHERE IDTHUCUONG=IDTHUCUONG$;  DELETE FROM V\_THUCUONG WHERE ID = IDTHUCUONG$;  END LOOP;  CLOSE cus;  DELETE FROM V\_LoaiThucuong WHERE ID = ID$;  END; |

* + 1. **sp\_updateLoaiThucUong**

|  |
| --- |
| CREATE OR REPLACE PROCEDURE sp\_updateLoaiThucUong(TenLThucUong$ NVARCHAR2,GHICHU$ NVARCHAR2, ID$ int )  AS  BEGIN  update v\_LOAITHUCUONG set tenLThucUong=TenLThucUong$,GhiChu=GHICHU$ WHERE id=ID$;  END; |

* + 1. **sp\_insertThucUong**

|  |
| --- |
| CREATE OR REPLACE PROCEDURE sp\_insertThucUong(tenThucUong$ NVARCHAR2,IDLTHUCUONG$ INT, DONGIA$ INT, GHICHU$ NVARCHAR2 )  AS  BEGIN  INSERT INTO V\_THUCUONG (tenThucUong,IDLTHUCUONG,DONGIA,GHICHU) VALUES (tenThucUong$,IDLTHUCUONG$,DONGIA$,GHICHU$);  END; |

* + 1. **sp\_deleteThucUong**

|  |
| --- |
| CREATE OR REPLACE PROCEDURE sp\_deleteThucUong(ID$ INT )  AS  BEGIN  DELETE FROM GIOHANG WHERE IDORDER IN (SELECT ID FROM ORDERS WHERE IDTHUCUONG=ID$);  DELETE FROM CHITIETTHEM WHERE IDORDER IN (SELECT ID FROM ORDERS WHERE IDTHUCUONG=ID$);  DELETE FROM CHITIETHOADON WHERE IDORDER IN (SELECT ID FROM ORDERS WHERE IDTHUCUONG=ID$);  DELETE FROM ORDERS WHERE IDTHUCUONG=ID$;  DELETE FROM V\_THUCUONG WHERE ID = ID$;  END; |

* + 1. **sp\_updateThucUong**

|  |
| --- |
| CREATE OR REPLACE PROCEDURE sp\_updateThucUong( tenThucUong$ NVARCHAR2, IDLTHUCUONG$ INT, DONGIA$ INT, GhiChu$ NVARCHAR2, ID$ INT)  AS  BEGIN  UPDATE V\_THUCUONG set tenThucUong=tenThucUong$,IDLTHUCUONG=IDLTHUCUONG$,DONGIA=DONGIA$,GhiChu=GhiChu$ WHERE id=ID$;  END; |

* + 1. **sp\_insertThem**

|  |
| --- |
| CREATE OR REPLACE PROCEDURE sp\_insertThem( TENHAT$ NVARCHAR2, DONGIA$ INT, GHICHU$ NVARCHAR2)  AS  BEGIN  INSERT INTO V\_THEM (TENHAT,DONGIA,GHICHU) VALUES (TENHAT$,DONGIA$,GHICHU$);  END; |

* + 1. **sp\_deleteThem**

|  |
| --- |
| CREATE OR REPLACE PROCEDURE sp\_deleteThem( ID$ int)  AS  BEGIN  DELETE FROM CHITIETTHEM WHERE IDTHEM = ID$;  DELETE FROM V\_THEM WHERE ID = ID$;  END; |

* + 1. **sp\_updateThem**

|  |
| --- |
| CREATE OR REPLACE PROCEDURE sp\_updateThem (TENHAT$ NVARCHAR2, DONGIA$ INT, GHICHU$ NVARCHAR2, ID$ INT)  AS  BEGIN  UPDATE v\_THEM SET TENHAT=TENHAT$,DONGIA=DONGIA$,GHICHU=GHICHU$ WHERE ID=ID$;  END; |

* + 1. **sp\_insertKhachHang**

|  |
| --- |
| CREATE OR REPLACE PROCEDURE sp\_insertKhachHang (USERNAME$ NVARCHAR2, PASSWORD$ NVARCHAR2, HOTEN$ NVARCHAR2,  NGSINH$ DATE, SDT$ VARCHAR2, DIACHI$ NVARCHAR2, ROLE$ INT)  AS  BEGIN  INSERT INTO V\_KHACHHANG (USERNAME,PASSWORD,HOTEN,NGSINH,SDT,DIACHI,ROLE) VALUES (USERNAME$,PASSWORD$,HOTEN$,NGSINH$,SDT$,DIACHI$,ROLE$);  END; |

* + 1. **sp\_deleteKhachHang**

|  |
| --- |
| CREATE OR REPLACE PROCEDURE sp\_deleteKhachHang (ID$ INT )  AS  USERNAME$ NVARCHAR2(50);  BEGIN    SELECT USERNAME INTO USERNAME$ FROM KHACHHANG WHERE ID=ID$;  DELETE FROM GIOHANG WHERE IDUSER = ID$;  DELETE FROM HOADON WHERE IDUSER = ID$;  DELETE FROM V\_KHACHHANG WHERE ID = ID$;    END; |

* + 1. **sp\_updateKhachHang**

|  |
| --- |
| CREATE OR REPLACE PROCEDURE sp\_updateKhachHang (NGSINH$ DATE,SDT$ VARCHAR2, DIACHI$ NVARCHAR2,ROLE$ INT,ID$ INT)  AS  BEGIN  UPDATE V\_KHACHHANG SET NGSINH=NGSINH$,SDT=SDT$,DIACHI=DIACHI$,ROLE=ROLE$ WHERE ID=ID$;  END; |

* + 1. **sp\_resetPassword**

|  |
| --- |
| CREATE OR REPLACE PROCEDURE sp\_resetPassword(USERNAME$ NVARCHAR2,PASS$ NVARCHAR2,HOTEN$ NVARCHAR2,NGAYSINH$ DATE,  SDT$ VARCHAR2, DIACHI$ NVARCHAR2, ID$ INT)  AS  BEGIN  UPDATE V\_KHACHHANG SET USERNAME=USERNAME$,PASSWORD=PASS$,HOTEN=HOTEN$,NGSINH=NGAYSINH$,SDT=SDT$,DIACHI=DIACHI$ WHERE ID=ID$;  COMMIT;  END; |

* + 1. **sp\_insertBill**

|  |
| --- |
| CREATE OR REPLACE PROCEDURE sp\_insertBill(IDUSER$ INT,NGAY$ DATE,TONGTIEN$ INT)  AS  IDBILL$ INT;  IDORDER$ INT;  COUNT$ INT;  CURSOR cus IS SELECT IDORDER,SOLUONG FROM GIOHANG WHERE IDUSER = IDUSER$;  BEGIN  INSERT INTO HOADON (IDUSER,NGAY,TONGTIEN,TINHTRANG) VALUES (IDUSER$,NGAY$,TONGTIEN$,'Đã đặt');  SELECT ID INTO IDBILL$ FROM (SELECT \* FROM HOADON ORDER BY ID DESC) WHERE IDUSER = 1 AND ROWNUM <= 1;  OPEN cus;  LOOP  FETCH cus INTO IDORDER$,COUNT$;  EXIT WHEN cus%NOTFOUND;  INSERT INTO CHITIETHOADON (IDHOADON,IDORDER,SOLUONG) VALUES (IDBILL$,IDORDER$,COUNT$);  END LOOP;  CLOSE cus;  DELETE GIOHANG WHERE IDUSER = IDUSER$;  END; |

* + 1. **sp\_cancelBill**

|  |
| --- |
| CREATE OR REPLACE PROCEDURE sp\_cancelBill(IDBILL$ IN HOADON.ID%TYPE)  AS  TINHTRANG$ NVARCHAR2(50);  BEGIN  SELECT TINHTRANG INTO TINHTRANG$ FROM HOADON WHERE ID = IDBILL$;  IF(TINHTRANG$ = 'Đã đặt') THEN  UPDATE HOADON SET TINHTRANG = 'Đã hủy' WHERE ID = IDBILL$;  ELSIF(TINHTRANG$ = 'Đã hủy' ) THEN  raise\_application\_error(-20022, 'Đơn hàng đã hủy rồii !');  ELSE  raise\_application\_error(-20022,''Đơn hàng đã hủy rồii !');  END IF;  END; |

* + 1. **sp\_payedBill**

|  |
| --- |
| CREATE OR REPLACE PROCEDURE sp\_payedBill(IDBILL$ IN HOADON.ID%TYPE)  AS  TINHTRANG$ NVARCHAR2(50);  BEGIN  SELECT TINHTRANG INTO TINHTRANG$ FROM HOADON WHERE ID = IDBILL$;  IF(TINHTRANG$ = 'Đã đặt') THEN  UPDATE HOADON SET TINHTRANG = 'Đã thanh toán' WHERE ID = IDBILL$;  ELSIF(TINHTRANG$ = 'Đã thanh toán' ) THEN  raise\_application\_error(-20022,'Đơn hàng đã thanh toán rồi!');  ELSE  raise\_application\_error(-20022,'ĐƠN HÀNG ĐÃ BỊ HỦY !');  END IF;  END; |

* + 1. **sp\_checkAllBill**

|  |
| --- |
| CREATE OR REPLACE PROCEDURE sp\_checkAllBill  AS  H1$ INT;M1$ INT;  H2$ INT;M2$ INT;  ID$ INT;  NGAY$ DATE;  CURSOR cus IS SELECT ID,NGAY FROM HOADON WHERE TINHTRANG='Đã đặt';  BEGIN  select to\_CHAR(to\_date(SYSDATE,'dd-MON-yy HH24:MI:SS'),'HH24') hour INTO H1$ from dual;  select to\_CHAR(to\_date(SYSDATE,'dd-MON-yy HH24:MI:SS'),'MI') MI INTO M1$ from dual;  OPEN cus;  LOOP  FETCH cus INTO ID$,NGAY$;  EXIT WHEN cus%NOTFOUND;  select to\_CHAR(to\_date(NGAY$,'dd-MON-yy HH24:MI:SS'),'HH24') hour INTO H2$ from dual;  select to\_CHAR(to\_date(NGAY$,'dd-MON-yy HH24:MI:SS'),'MI') MI INTO M2$ from dual;    IF (((H1$-H2$)\*60 + M1$+60-M2$) >= 60 OR NGAY$+1<SYSDATE) THEN  UPDATE HOADON SET TINHTRANG='Đã hủy' WHERE ID=ID$;  END IF;  END LOOP;  CLOSE cus;  END; |

* 1. **Trigger**
     1. **tg\_addLoaiThucUong**

|  |
| --- |
| CREATE OR REPLACE TRIGGER tg\_addLoaiThucUong  INSTEAD OF INSERT ON v\_LOAITHUCUONG  FOR EACH ROW  DECLARE  count$ int;  BEGIN  select COUNT(\*) into count$ from LOAITHUCUONG L where L.tenLThucUong=:new.TENLTHUCUONG;  IF(count$>0) THEN  BEGIN  raise\_application\_error(-20022,'TÊN LOẠI THỨC UỐNG ĐÃ TỒN TẠI');  ROLLBACK;  RETURN;  END;  END IF;  INSERT INTO LOAITHUCUONG(tenLThucUong,ghichu) VALUES (:NEW.tenLThucUong,:new.ghichu);  END; |

* + 1. **tg\_updateLoaiThucUong**

|  |
| --- |
| CREATE OR REPLACE TRIGGER tg\_updateLoaiThucUong  INSTEAD OF UPDATE ON v\_LOAITHUCUONG  FOR EACH ROW  DECLARE  count$ int;  BEGIN  select COUNT(\*) into count$ from LOAITHUCUONG L where L.tenLThucUong=:new.TENLTHUCUONG;  IF(count$>0 AND :new.TENLTHUCUONG<>:old.TENLTHUCUONG) THEN  BEGIN  raise\_application\_error(-20022,'TÊN LOẠI THỨC UỐNG ĐÃ TỒN TẠI');  ROLLBACK;  RETURN;  END;  END IF;  UPDATE LOAITHUCUONG SET tenLThucUong=:new.TENLTHUCUONG,GHICHU=:NEW.GHICHU WHERE ID=:new.ID;  END; |

* + 1. **tg\_addThucUong**

|  |
| --- |
| CREATE OR REPLACE TRIGGER tg\_addThucUong  INSTEAD OF INSERT ON v\_THUCUONG  FOR EACH ROW  DECLARE  count$ int;  BEGIN  select COUNT(\*) into count$ from THUCUONG T where T.tenThucUong=:new.TENTHUCUONG;  IF(count$>0) THEN  BEGIN  raise\_application\_error(-20022,'TÊN THỨC UỐNG ĐÃ TỒN TẠI');  ROLLBACK;  RETURN;  END;  END IF;  INSERT INTO THUCUONG (tenThucUong,IDLTHUCUONG,DONGIA,GHICHU)  VALUES (:new.TENTHUCUONG,:new.IDLTHUCUONG,:new.DONGIA,:new.GHICHU);  END; |

* + 1. **tg\_updateThucUong**

|  |
| --- |
| CREATE OR REPLACE TRIGGER tg\_updateThucUong  INSTEAD OF UPDATE ON v\_THUCUONG  FOR EACH ROW  DECLARE  count$ int;  BEGIN  IF(:NEW.DONGIA < :OLD.DONGIA) THEN  BEGIN  raise\_application\_error(-20022,'ĐƠN GIÁ PHẢI LỚN HƠN HAY BẰNG ĐƠN GIÁ CŨ');  ROLLBACK;RETURN;  END;  END IF;  select COUNT(\*) into count$ from THUCUONG T where T.tenThucUong=:new.TENTHUCUONG;  IF(count$>0 AND :new.TENTHUCUONG<>:old.TENTHUCUONG) THEN  BEGIN  raise\_application\_error(-20022,'TÊN THỨC UỐNG ĐÃ TỒN TẠI');  ROLLBACK;RETURN;  END;  END IF;  UPDATE THUCUONG SET tenThucUong=:new.TENTHUCUONG,  IDLTHUCUONG=:new.IDLTHUCUONG,  dongia=:new.dongia,  ghichu=:new.ghichu  WHERE ID=:new.ID;  END; |

* + 1. **tg\_addTHEM**

|  |
| --- |
| CREATE OR REPLACE TRIGGER tg\_addTHEM  INSTEAD OF INSERT ON v\_THEM  FOR EACH ROW  DECLARE  count$ int;  BEGIN  select COUNT(\*) into count$ from THEM T where T.tenHAT=:new.TENHAT;  IF(count$>0) THEN  BEGIN  raise\_application\_error(-20022,'TÊN TOPPING ĐÃ TỒN TẠI');  ROLLBACK;  RETURN;  END;  END IF;  INSERT INTO THEM (TENHAT,DONGIA,GHICHU)  VALUES (:new.TENHAT,:new.DONGIA,:new.GHICHU);  END; |

* + 1. **tg\_updateTHEM**

|  |
| --- |
| CREATE OR REPLACE TRIGGER tg\_updateTHEM  INSTEAD OF UPDATE ON v\_THEM  FOR EACH ROW  DECLARE  count$ int;  BEGIN  IF(:NEW.DONGIA < :OLD.DONGIA) THEN  BEGIN  raise\_application\_error(-20022,'ĐƠN GIÁ PHẢI LỚN HƠN HAY BẰNG ĐƠN GIÁ CŨ');  ROLLBACK;RETURN;  END;  END IF;  select COUNT(\*) into count$ from THEM T where T.tenHAT=:new.TENHAT;  IF(count$>0 AND :new.TENHAT<>:old.TENHAT) THEN  BEGIN  raise\_application\_error(-20022,'TÊN TOPPING ĐÃ TỒN TẠI');  ROLLBACK;RETURN;  END;  END IF;  UPDATE THEM SET tenHAT=:new.TENHAT,  dongia=:new.dongia,  ghichu=:new.ghichu  WHERE ID=:new.ID;  END; |

* + 1. **tg\_addKHACHHANG**

|  |
| --- |
| CREATE OR REPLACE TRIGGER tg\_addKHACHHANG  INSTEAD OF INSERT ON v\_KHACHHANG  FOR EACH ROW  DECLARE  count$ int;  BEGIN  select COUNT(\*) into count$ from KHACHHANG K where K.USERNAME=:new.USERNAME;  IF(count$>0) THEN  BEGIN  raise\_application\_error(-20022,'USERNAME ĐÃ TỒN TẠI');  ROLLBACK;  RETURN;  END;  END IF;  INSERT INTO KHACHHANG (USERNAME,PASSWORD,HOTEN,NGSINH,SDT,DIACHI,ROLE) VALUES  (:NEW.USERNAME,:NEW.PASSWORD,:NEW.HOTEN,:NEW.NGSINH,:NEW.SDT,:NEW.DIACHI,:NEW.ROLE);  END; |

* + 1. **tg\_updateKHACHHANG**

|  |
| --- |
| CREATE OR REPLACE TRIGGER tg\_updateKHACHHANG  INSTEAD OF UPDATE ON v\_KHACHHANG  FOR EACH ROW  DECLARE  count$ int;  BEGIN  select COUNT(\*) into count$ from KHACHHANG T where T.USERNAME=:new.USERNAME;  IF(count$>0 AND :new.USERNAME<>:old.USERNAME) THEN  BEGIN  raise\_application\_error(-20022,'USERNAME ĐÃ TỒN TẠI');  ROLLBACK;RETURN;  END;  END IF;  UPDATE KHACHHANG SET USERNAME=:new.USERNAME,  PASSWORD=:new.PASSWORD,  HOTEN=:new.HOTEN,  NGSINH=:new.NGSINH,SDT=:new.SDT,  DIACHI=:new.DIACHI,ROLE=:new.ROLE  WHERE ID=:new.ID;  END; |

* 1. **Function**
     1. **fn\_Login**

|  |
| --- |
| CREATE TYPE table\_row AS OBJECT (  ID INT,  USERNAME NVARCHAR2(50),  PASSWORD VARCHAR2(50),  HOTEN NVARCHAR2(50),  NGSINH DATE ,  SDT VARCHAR2(11),  DIACHI NVARCHAR2(50),  ROLE INT  );  CREATE TYPE table\_tab IS TABLE OF table\_row;  -- Build the table function  CREATE OR REPLACE FUNCTION fn\_Login(name\_ in KHACHHANG.USERNAME%TYPE,  pass\_ in KHACHHANG.PASSWORD%TYPE )  RETURN table\_tab AS  l\_tab table\_tab := table\_tab();  ID\_ INT;  USERNAME\_ NVARCHAR2(50);  PASSWORD\_ VARCHAR2(50);  HOTEN\_ NVARCHAR2(50);  NGSINH\_ DATE;  SDT\_ VARCHAR2(11);  DIACHI\_ NVARCHAR2(50);  ROLE\_ INT;  CURSOR cus IS SELECT ID,USERNAME,PASSWORD,HOTEN,NGSINH,SDT,DIACHI,ROLE  FROM KHACHHANG KH where KH.USERNAME=name\_ and KH.PASSWORD=pass\_;  BEGIN  OPEN cus;  LOOP  FETCH cus INTO ID\_,USERNAME\_,PASSWORD\_,HOTEN\_,NGSINH\_,SDT\_,DIACHI\_,ROLE\_;  EXIT WHEN cus%NOTFOUND;  l\_tab.extend;  l\_tab(l\_tab.last) := table\_row(ID\_,USERNAME\_,PASSWORD\_,HOTEN\_,NGSINH\_,SDT\_,DIACHI\_,ROLE\_);  END LOOP;  CLOSE cus;  RETURN l\_tab;  END; |

* + 1. **fn\_ThongKeHoaDon**

|  |
| --- |
| CREATE TYPE table\_row1 AS OBJECT (  ID INT,  NGAY DATE,  TONGTIEN INT,  TINHTRANG NVARCHAR2(50),  KHACHHANG NVARCHAR2(50)  );  CREATE TYPE table\_tab1 IS TABLE OF table\_row1;  -- Build the table function  CREATE OR REPLACE FUNCTION fn\_ThongKeHoaDon(DATE1$ DATE, DATE2$ DATE)  RETURN table\_tab1 AS  l\_tab table\_tab1 := table\_tab1();  ID$ INT;  NGAY$ DATE;  TONGTIEN$ INT;  TINHTRANG$ NVARCHAR2(50);  KHACHHANG$ NVARCHAR2(50);  CURSOR cus IS SELECT H.ID,H.NGAY,H.TONGTIEN,H.TINHTRANG,K.HOTEN  FROM HOADON H,KHACHHANG K  WHERE H.IDUSER=K.ID  AND NGAY >= DATE1$  AND NGAY <= DATE2$+1;  BEGIN  OPEN cus;  LOOP  FETCH cus INTO ID$,NGAY$,TONGTIEN$,TINHTRANG$,KHACHHANG$;  EXIT WHEN cus%NOTFOUND;  l\_tab.extend;  l\_tab(l\_tab.last) := table\_row1(ID$,NGAY$,TONGTIEN$,TINHTRANG$,KHACHHANG$);  END LOOP;  CLOSE cus;  RETURN l\_tab;  END; |

* + 1. **fn\_getOrderByUserId**

|  |
| --- |
| CREATE TYPE table\_row4 AS OBJECT (  ID INT,  IDTHUCUONG NVARCHAR2(50),  TONGTIEN INT,  GHICHU NVARCHAR2(50)  );  CREATE TYPE table\_tab4 IS TABLE OF table\_row4;  -- Build the table function  CREATE OR REPLACE FUNCTION fn\_getOrderByUserId(USERID$ INT)  RETURN table\_tab4 AS  l\_tab table\_tab4 := table\_tab4();  ID$ INT;  IDTHUCUONG$ NVARCHAR2(50);  TONGTIEN$ INT;  GHICHU$ NVARCHAR2(50);  CURSOR cus IS SELECT O.ID,O.IDTHUCUONG,O.TONGTIEN,O.GHICHU  FROM NGOCHOANG2.ORDERS O, NGOCHOANG2.GIOHANG G  WHERE O.ID=G.IDORDER AND G.IDUSER =USERID$  ORDER BY ID ASC;  BEGIN  OPEN cus;  LOOP  FETCH cus INTO ID$,IDTHUCUONG$,TONGTIEN$,GHICHU$;  EXIT WHEN cus%NOTFOUND;  l\_tab.extend;  l\_tab(l\_tab.last) := table\_row4(ID$,IDTHUCUONG$,TONGTIEN$,GHICHU$);  END LOOP;  CLOSE cus;  RETURN l\_tab;  END; |

* + 1. **fn\_getListDrink**

|  |
| --- |
| CREATE TYPE table\_row2 AS OBJECT (  ID INT,  TENTHUCUONG NVARCHAR2(50),  TENLTHUCUONG NVARCHAR2(50),  DONGIA INT,  GHICHU NVARCHAR2(50)  );  CREATE TYPE table\_tab2 IS TABLE OF table\_row2;  -- Build the table function  CREATE OR REPLACE FUNCTION fn\_getListDrink  RETURN table\_tab2 AS  l\_tab table\_tab2 := table\_tab2();  ID$ INT;  TENTHUCUONG$ NVARCHAR2(50);  TENLTHUCUONG$ NVARCHAR2(50);  DONGIA$ INT;  GHICHU$ NVARCHAR2(50);  CURSOR cus IS SELECT T.ID, T.TENTHUCUONG,L.TENLTHUCUONG,T.DONGIA,T.GHICHU  FROM ThucUong T,LoaiThucUong L  WHERE T.idLThucUong=L.ID  ORDER BY L.ID ASC;  BEGIN  OPEN cus;  LOOP  FETCH cus INTO ID$,TENTHUCUONG$,TENLTHUCUONG$,DONGIA$,GHICHU$;  EXIT WHEN cus%NOTFOUND;  l\_tab.extend;  l\_tab(l\_tab.last) := table\_row2(ID$,TENTHUCUONG$,TENLTHUCUONG$,DONGIA$,GHICHU$);  END LOOP;  CLOSE cus;  RETURN l\_tab;  END; |

* + 1. **fn\_getListBillOrdering**

|  |
| --- |
| CREATE TYPE table\_row3 AS OBJECT (  ID INT,  NGAY DATE,  TONGTIEN INT,  TINHTRANG NVARCHAR2(50),  KHACHHANG NVARCHAR2(50)  );  CREATE TYPE table\_tab3 IS TABLE OF table\_row3;  -- Build the table function  CREATE OR REPLACE FUNCTION fn\_getListBillOrdering(DATE1$ DATE, DATE2$ DATE)  RETURN table\_tab3 AS  l\_tab table\_tab3 := table\_tab3();  ID$ INT;  NGAY$ DATE;  TONGTIEN$ INT;  TINHTRANG$ NVARCHAR2(50);  KHACHHANG$ NVARCHAR2(50);  CURSOR cus IS SELECT H.ID,H.NGAY,H.TONGTIEN,H.TINHTRANG,K.HOTEN  FROM HOADON H,KHACHHANG K  WHERE H.IDUSER=K.ID  AND TINHTRANG='?ã ??t'  AND NGAY >= DATE1$  AND NGAY <= DATE2$+1;  BEGIN  OPEN cus;  LOOP  FETCH cus INTO ID$,NGAY$,TONGTIEN$,TINHTRANG$,KHACHHANG$;  EXIT WHEN cus%NOTFOUND;  l\_tab.extend;  l\_tab(l\_tab.last) := table\_row3(ID$,NGAY$,TONGTIEN$,TINHTRANG$,KHACHHANG$);  END LOOP;  CLOSE cus;  RETURN l\_tab;  END; |

* + 1. **fn\_getListBillCanceled**

|  |
| --- |
| CREATE OR REPLACE FUNCTION fn\_getListBillCanceled(DATE1$ DATE, DATE2$ DATE)  RETURN table\_tab3 AS  l\_tab table\_tab3 := table\_tab3();  ID$ INT;  IDUSER$ INT;  NGAY$ DATE;  TONGTIEN$ INT;  TINHTRANG$ NVARCHAR2(50);  KHACHHANG$ NVARCHAR2(50);  CURSOR cus IS SELECT H.ID,H.NGAY,H.TONGTIEN,H.TINHTRANG,K.HOTEN  FROM HOADON H,KHACHHANG K  WHERE H.IDUSER=K.ID  AND TINHTRANG='?ã h?y'  AND NGAY >= DATE1$  AND NGAY <= DATE2$+1;  BEGIN  OPEN cus;  LOOP  FETCH cus INTO ID$,NGAY$,TONGTIEN$,TINHTRANG$,KHACHHANG$;  EXIT WHEN cus%NOTFOUND;  l\_tab.extend;  l\_tab(l\_tab.last) := table\_row3(ID$,NGAY$,TONGTIEN$,TINHTRANG$,KHACHHANG$);  END LOOP;  CLOSE cus;  RETURN l\_tab;  END; |

1. **Phân quyền**
   1. **Nhóm quyền Adminitrator**

|  |
| --- |
| -- tạo nhóm quyền  CREATE ROLE Adminitrator;  -- cấp quyền  GRANT CONNECT, RESOURCE TO Adminitrator;  GRANT CREATE USER TO Adminitrator;  GRANT DROP USER TO Adminitrator;  GRANT ALTER USER TO Adminitrator;  GRANT CREATE ROLE TO Adminitrator;  -----------------------------  GRANT SELECT,INSERT,DELETE,UPDATE ON NGOCHOANG2.LOAITHUCUONG TO Adminitrator;  GRANT SELECT,INSERT,DELETE,UPDATE ON NGOCHOANG2.THUCUONG TO Adminitrator;  GRANT SELECT,INSERT,DELETE,UPDATE ON NGOCHOANG2.THEM TO Adminitrator;  GRANT SELECT,INSERT,DELETE,UPDATE ON NGOCHOANG2.KHACHHANG TO Adminitrator;  GRANT SELECT,INSERT,DELETE,UPDATE ON NGOCHOANG2.ORDERS TO Adminitrator;  GRANT SELECT,INSERT,DELETE,UPDATE ON NGOCHOANG2.HOADON TO Adminitrator;  GRANT SELECT,INSERT,DELETE,UPDATE ON NGOCHOANG2.GIOHANG TO Adminitrator;  GRANT SELECT,INSERT,DELETE,UPDATE ON NGOCHOANG2.CHITIETTHEM TO Adminitrator;  GRANT SELECT,INSERT,DELETE,UPDATE ON NGOCHOANG2.CHITIETHOADON TO Adminitrator;  GRANT SELECT,INSERT,DELETE,UPDATE ON NGOCHOANG2.V\_LOAITHUCUONG TO Adminitrator;  GRANT SELECT,INSERT,DELETE,UPDATE ON NGOCHOANG2.V\_THUCUONG TO Adminitrator;  GRANT SELECT,INSERT,DELETE,UPDATE ON NGOCHOANG2.V\_THEM TO Adminitrator;  GRANT SELECT,INSERT,DELETE,UPDATE ON NGOCHOANG2.V\_KHACHHANG TO Adminitrator;  -- thực thi các stored  GRANT EXECUTE ON NGOCHOANG2.sp\_insertLoaiThucUong TO Adminitrator;  GRANT EXECUTE ON NGOCHOANG2.sp\_DeleteLoaiThucuong TO Adminitrator;  GRANT EXECUTE ON NGOCHOANG2.sp\_updateLoaiThucUong TO Adminitrator;  GRANT EXECUTE ON NGOCHOANG2.sp\_insertThucUong TO Adminitrator;  GRANT EXECUTE ON NGOCHOANG2.sp\_deleteThucUong TO Adminitrator;  GRANT EXECUTE ON NGOCHOANG2.sp\_updateThucUong TO Adminitrator;  GRANT EXECUTE ON NGOCHOANG2.sp\_insertThem TO Adminitrator;  GRANT EXECUTE ON NGOCHOANG2.sp\_deleteThem TO Adminitrator;  GRANT EXECUTE ON NGOCHOANG2.sp\_updateThem TO Adminitrator;  GRANT EXECUTE ON NGOCHOANG2.sp\_insertKhachHang TO Adminitrator;  GRANT EXECUTE ON NGOCHOANG2.sp\_deleteKhachHang TO Adminitrator;  GRANT EXECUTE ON NGOCHOANG2.sp\_updateKhachHang TO Adminitrator;  GRANT EXECUTE ON NGOCHOANG2.sp\_resetPassword TO Adminitrator;  GRANT EXECUTE ON NGOCHOANG2.sp\_insertBill TO Adminitrator;  GRANT EXECUTE ON NGOCHOANG2.sp\_cancelBill TO Adminitrator;  GRANT EXECUTE ON NGOCHOANG2.sp\_payedBill TO Adminitrator;  GRANT EXECUTE ON NGOCHOANG2.sp\_checkAllBill TO Adminitrator;  GRANT EXECUTE ON NGOCHOANG2.sp\_resetPassword TO Adminitrator;  GRANT EXECUTE ON NGOCHOANG2.sp\_isAdmin TO Adminitrator;  -- thực thi function  GRANT EXECUTE ON NGOCHOANG2.fn\_Login TO Adminitrator;  GRANT EXECUTE ON NGOCHOANG2.fn\_ThongKeHoaDon TO Adminitrator;  GRANT EXECUTE ON NGOCHOANG2.fn\_getOrderByUserId TO Adminitrator;  GRANT EXECUTE ON NGOCHOANG2.fn\_getListDrink TO Adminitrator;  GRANT EXECUTE ON NGOCHOANG2.fn\_getListBillOrdering TO Adminitrator;  GRANT EXECUTE ON NGOCHOANG2.fn\_getListBillCanceled TO Adminitrator;  -- tạo user  CREATE USER tshoangvy IDENTIFIED BY 12345;  -- gán quyền cho user  GRANT Adminitrator TO tshoangvy; |

* 1. **Nhóm quyền Customer**

|  |
| --- |
| -- tạo nhóm quyền  CREATE ROLE Customer;  -- cấp quyền cho nhóm  GRANT CONNECT, RESOURCE TO Customer;  GRANT ALTER USER TO Customer;  -----------------------------  GRANT SELECT ON NGOCHOANG2.LOAITHUCUONG TO Customer;  GRANT SELECT ON NGOCHOANG2.THUCUONG TO Customer;  GRANT SELECT ON NGOCHOANG2.THEM TO Customer;  GRANT SELECT ON NGOCHOANG2.KHACHHANG TO Customer;  GRANT SELECT,INSERT ON NGOCHOANG2.ORDERS TO Customer;  GRANT SELECT,INSERT,UPDATE ON NGOCHOANG2.HOADON TO Customer;  GRANT SELECT,INSERT,DEsLETE,UPDATE ON NGOCHOANG2.GIOHANG TO Customer;  GRANT SELECT,INSERT ON NGOCHOANG2.CHITIETTHEM TO Customer;  GRANT SELECT,INSERT ON NGOCHOANG2.CHITIETHOADON TO Customer;  -- thực thi các stored  GRANT EXECUTE ON NGOCHOANG2.sp\_resetPassword TO Customer;  GRANT EXECUTE ON NGOCHOANG2.sp\_insertBill TO Customer;  GRANT EXECUTE ON NGOCHOANG2.sp\_cancelBill TO Customer;  GRANT EXECUTE ON NGOCHOANG2.sp\_payedBill TO Customer;  GRANT EXECUTE ON NGOCHOANG2.sp\_checkAllBill TO Customer;  GRANT EXECUTE ON NGOCHOANG2.sp\_resetPassword TO Customer;  -- thực thi function  GRANT EXECUTE ON NGOCHOANG2.fn\_Login TO Customer;  GRANT EXECUTE ON NGOCHOANG2.fn\_getListDrink TO Customer;  GRANT EXECUTE ON NGOCHOANG2.fn\_getOrderByUserId TO Customer;  -- tạo user  CREATE USER tsannguyen IDENTIFIED BY 12345;  CREATE USER tsmainguyen IDENTIFIED BY 123;  CREATE USER tsnamle IDENTIFIED BY 123123;  CREATE USER tsdaobui IDENTIFIED BY 1234;  -- phân quyền cho user  GRANT Customer TO tsannguyen;  GRANT Customer TO tsmainguyen;  GRANT Customer TO tsnamle;  GRANT Customer TO tsdaobui; |