HCMUTE FIT HCMUTE

BẢO MẬT WEB

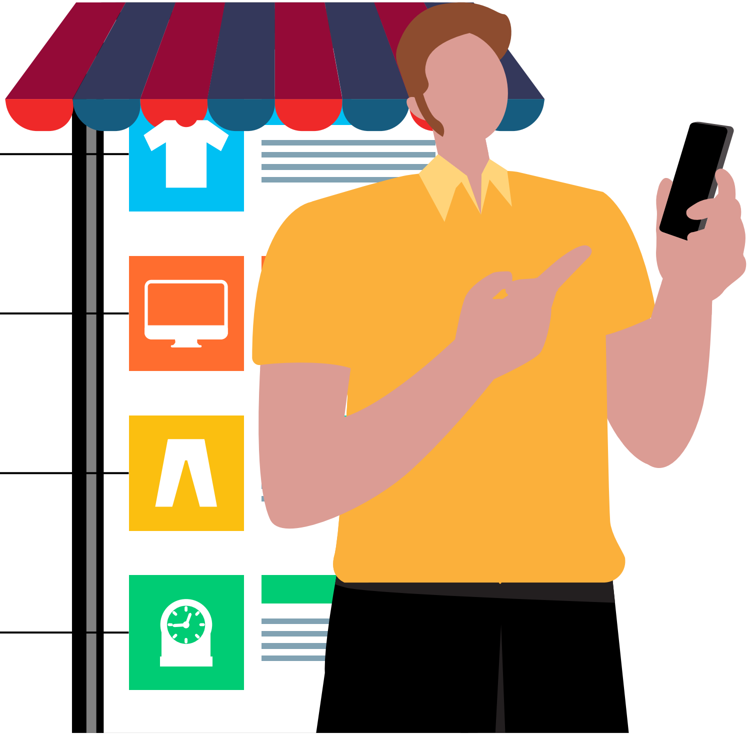
XÂY DỰNG WEBSITE KINH DOANH GIÀY VÀ PHỤ KIỆN

NHÓM 7

GVHD: THS. LÊ THỊ MINH CHÂU

DANH SÁCH THÀNH VIÊN

DIỆP TẤN LUÂN	19110055
LÊ THỊ MINH NGUYỆT	19110413
TRẦN KIỆN KHANG	19110375
BÙI ANH ĐỨC	19110348



Xây dựng Website kinh doanh giày và phụ kiện

1

Tổng quan

Lý do, phạm vi, mục tiêu chọn đề tài

2

Lược đồ quan hệ

ERD, Database diagram

3

Cài đặt CSDL

Views, Stored procedures, Functions, Triggers

4

Phân quyền

Roles, user, phân quyền

5

Code cài đặt ứng dụng

Coding...

6

Demo sản phẩm

Hướng dẫn sử dụng, kết luận

Tổng quan đề tài

Mục tiêu

Phù hợp với nhu cầu xã hội hiện đại.
TMĐT đang ngày càng phát triển. Đặc biệt,
trong đại dịch COVID-19.

Phạm vi

Quản lý một trang TMĐT
(hoạt động mua – bán).

Công cụ

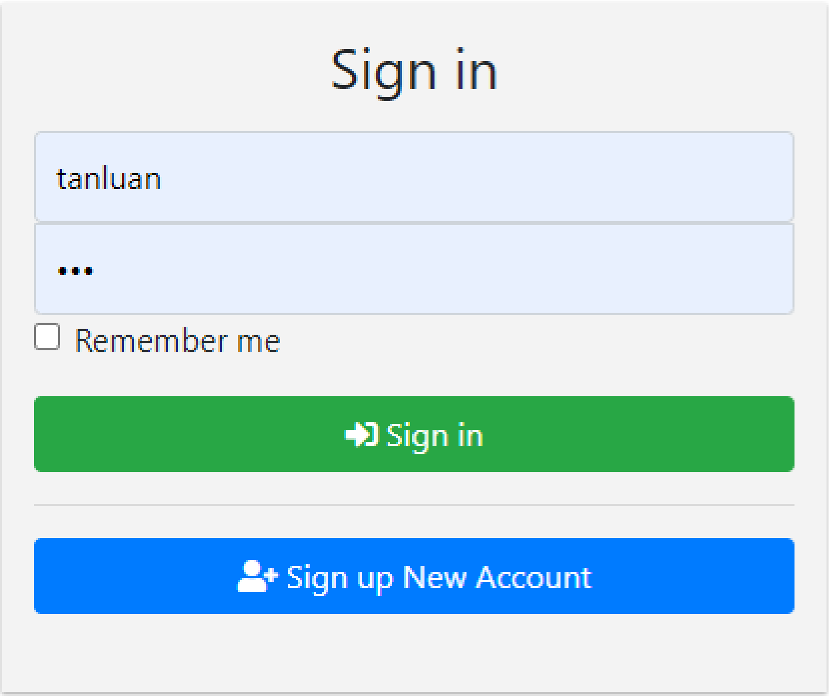
- Ngôn ngữ lập trình: Java
- Công nghệ: Serlet và jsp.
- Front-end: HTML, JS, CSS, Boostrap framework.
- CSDL: SQL Server.

Mô tả

Đáp ứng đầy đủ các
yêu cầu của một
website TMĐT.

Chức năng

CHỨC NĂNG
WEBSITE KINH DOANH GIÀY VÀ PHỤ KIỆN

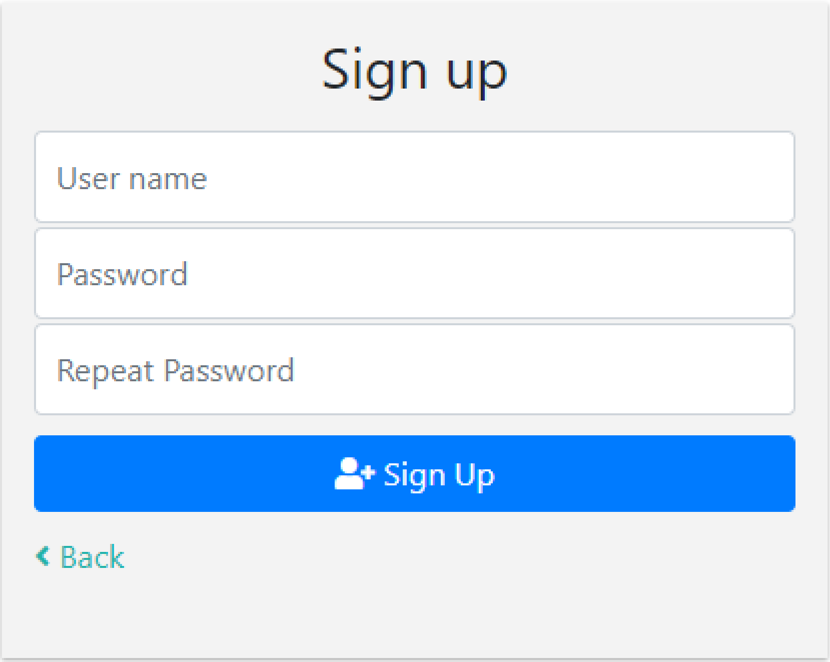


Sign in

tanluan

•••

☐ Remember me



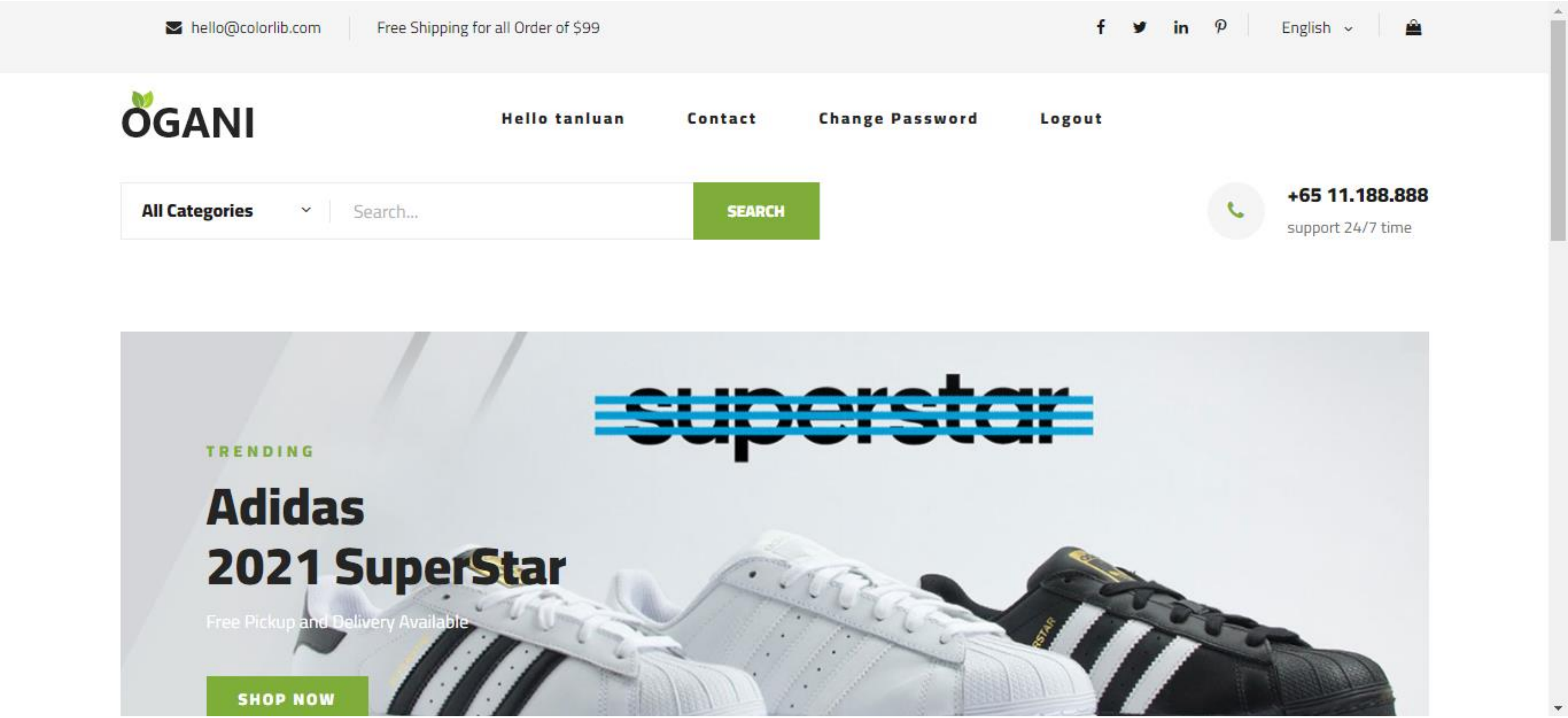
Sign up

User name

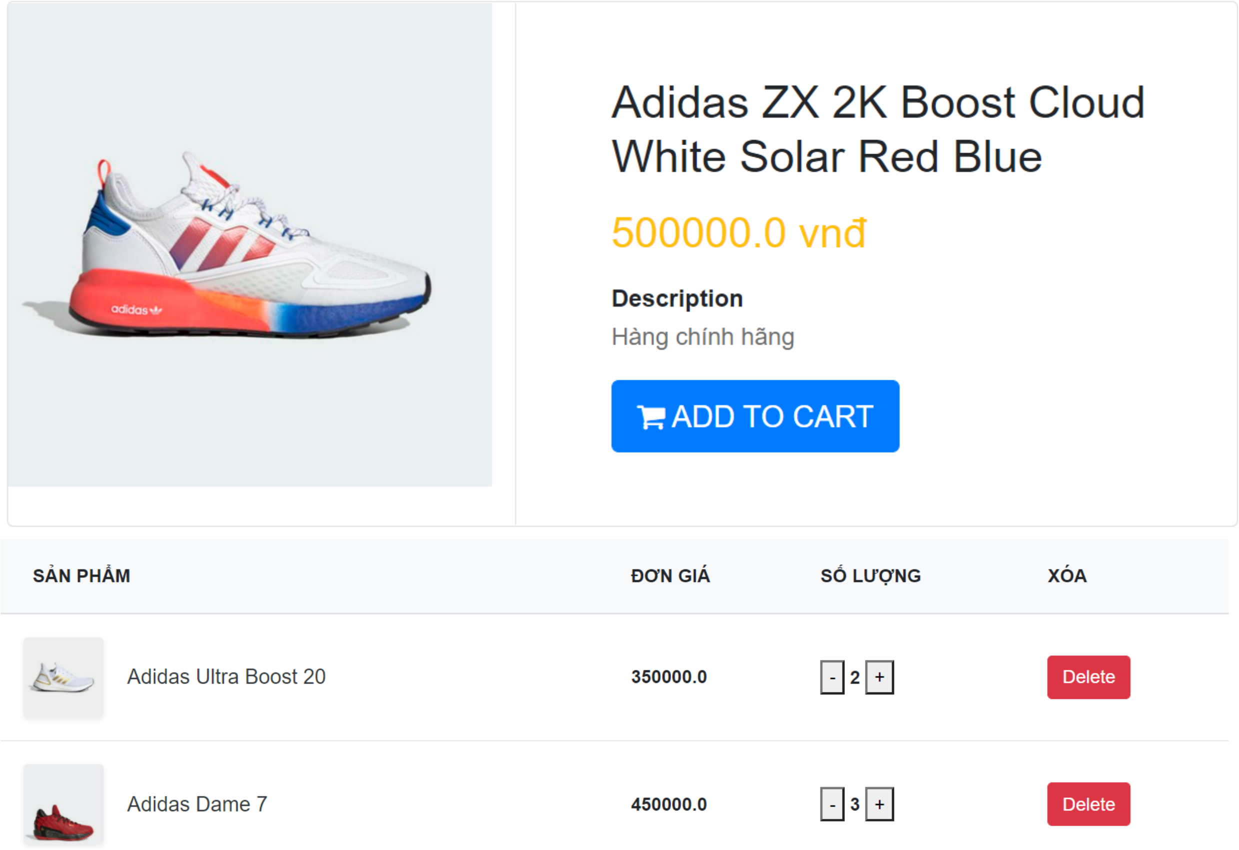
Password

Repeat Password

Đăng nhập – Tạo tài khoản mới



Giao diện chính



Adidas ZX 2K Boost Cloud White Solar Red Blue

500000.0 vnđ

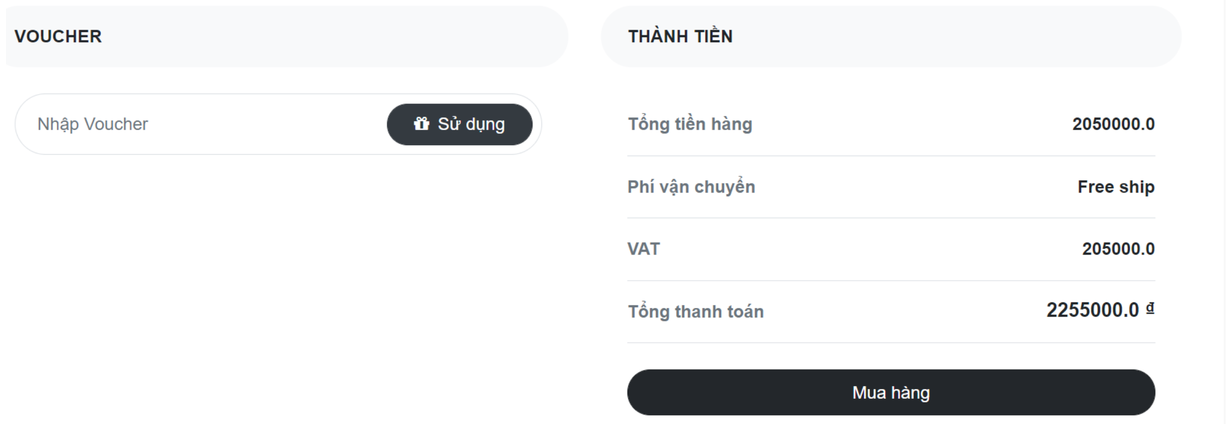
Description

Hàng chính hãng

SẢN PHẨM	ĐƠN GIÁ	SỐ LƯỢNG	XÓA
Adidas Ultra Boost 20	350000.0	- 2 +	Delete
Adidas Dame 7	450000.0	- 3 +	Delete

Mua hàng

Thanh toán

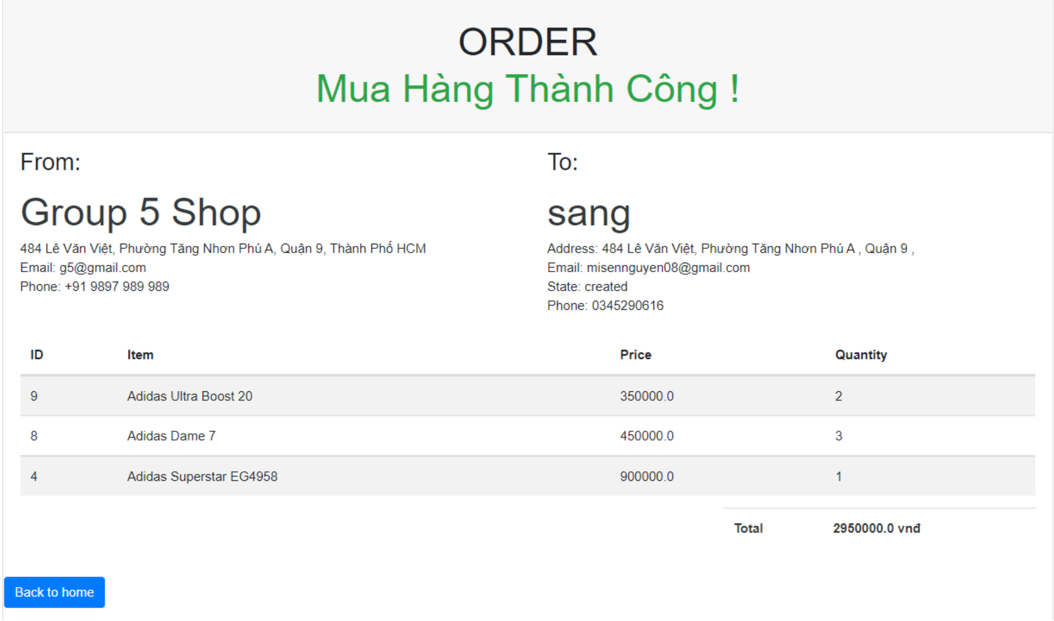


VOUCHER

Nhập Voucher

THÀNH TIỀN

Tổng tiền hàng	2050000.0
Phí vận chuyển	Free ship
VAT	205000.0
Tổng thanh toán	2255000.0 ₫



ORDER

Mua Hàng Thành Công !

From:

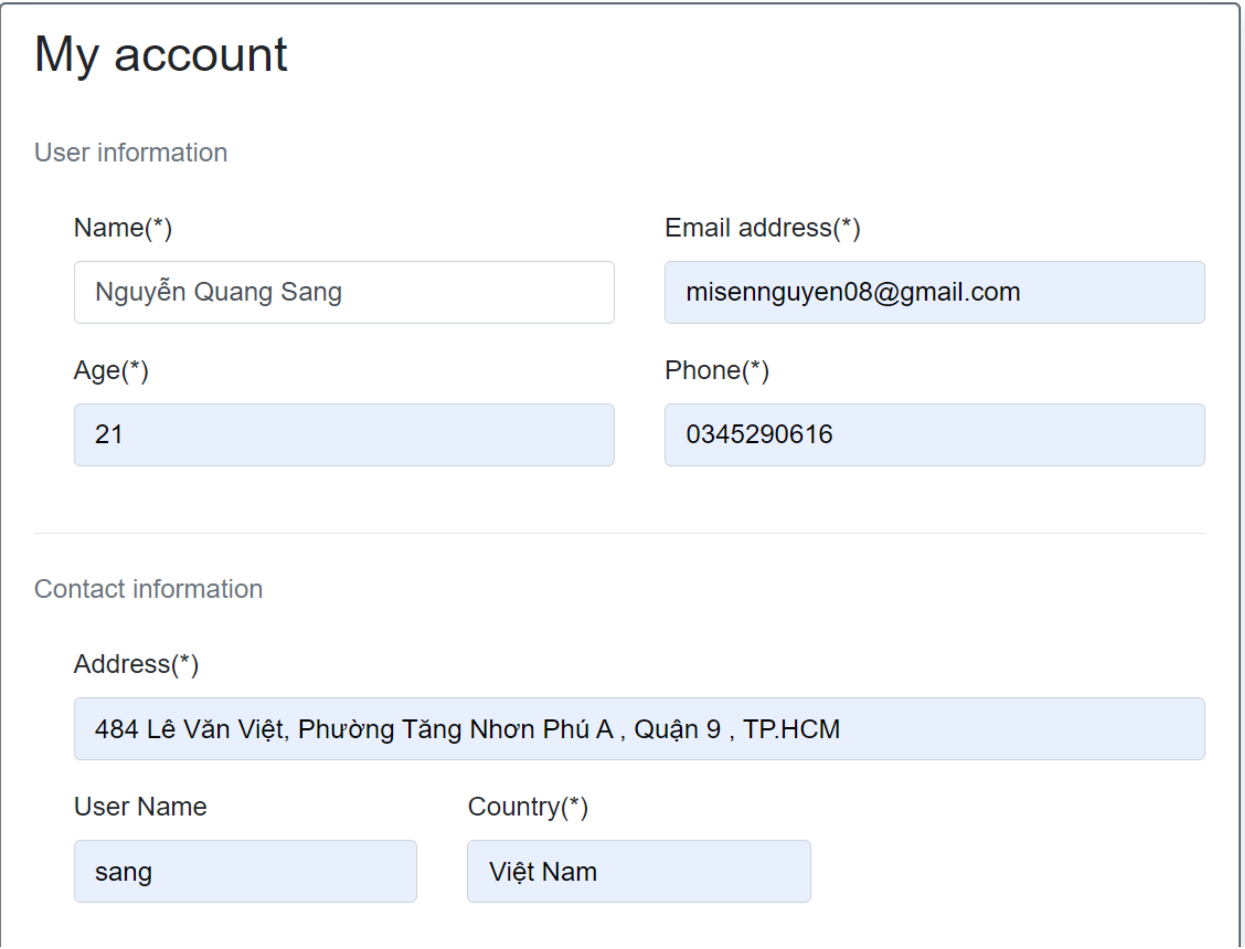
Group 5 Shop
484 Lê Văn Việt, Phường Tăng Nhơn Phú A, Quận 9, Thành Phố HCM
Email: g5@gmail.com
Phone: +91 9897 989 989

To:

sang
Address: 484 Lê Văn Việt, Phường Tăng Nhơn Phú A , Quận 9 ,
Email: misennguyen08@gmail.com
State: created
Phone: 0345290616

ID	Item	Price	Quantity
9	Adidas Ultra Boost 20	350000.0	2
8	Adidas Dame 7	450000.0	3
4	Adidas Superstar EG4958	900000.0	1

Total **2950000.0 vnđ**



My account

User information

Name(*)

Nguyễn Quang Sang

Email address(*)

misennguyen08@gmail.com

Age(*)

21

Phone(*)

0345290616

Contact information

Address(*)

484 Lê Văn Việt, Phường Tăng Nhơn Phú A , Quận 9 , TP.HCM

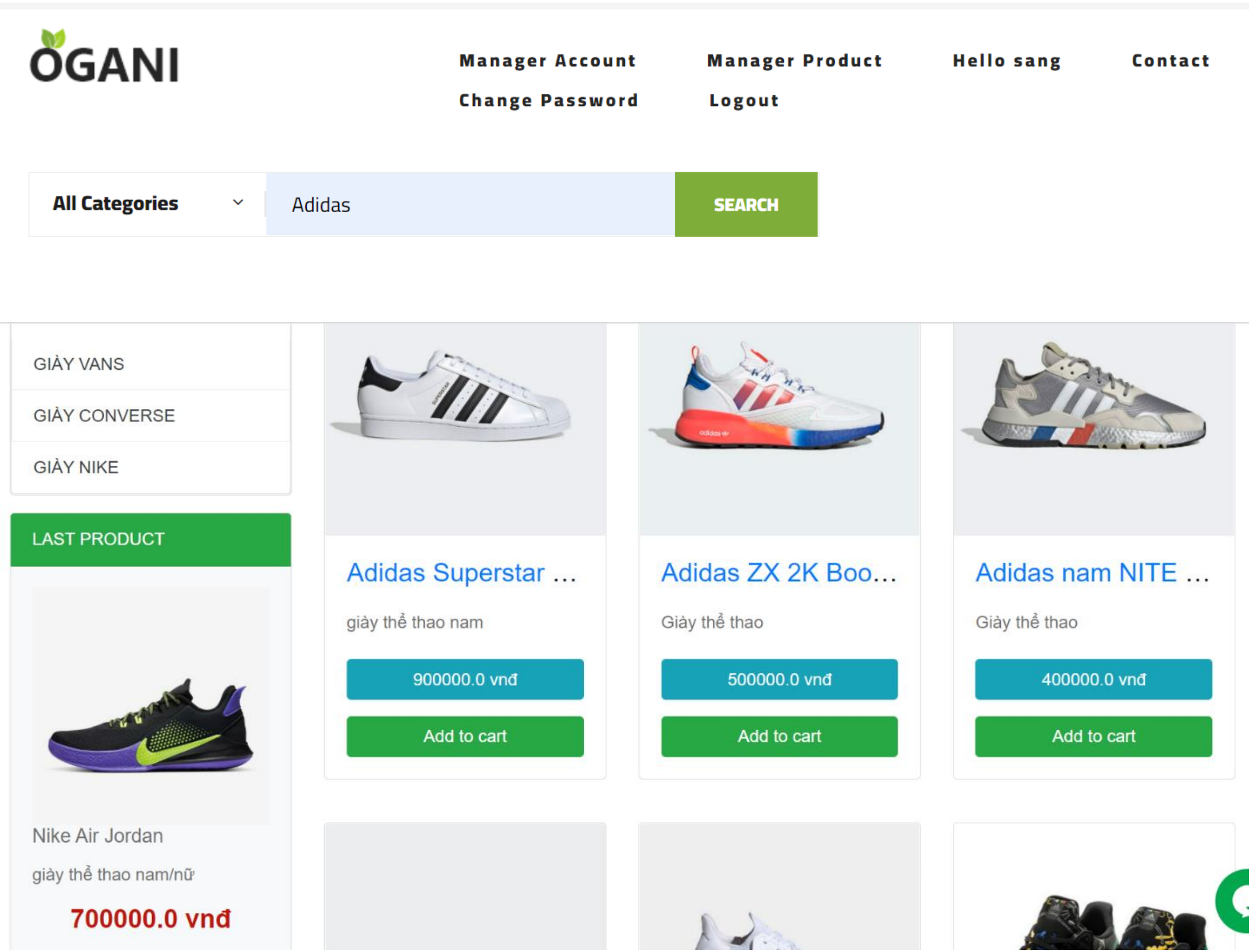
User Name

sang

Country(*)

Việt Nam

Thêm thông tin cá nhân



Adidas

Nike Air Jordan

giày thể thao nam/nữ

700000.0 vnđ

Adidas Superstar …

giày thể thao nam

900000.0 vnđ

Adidas ZX 2K Boo…

Giày thể thao

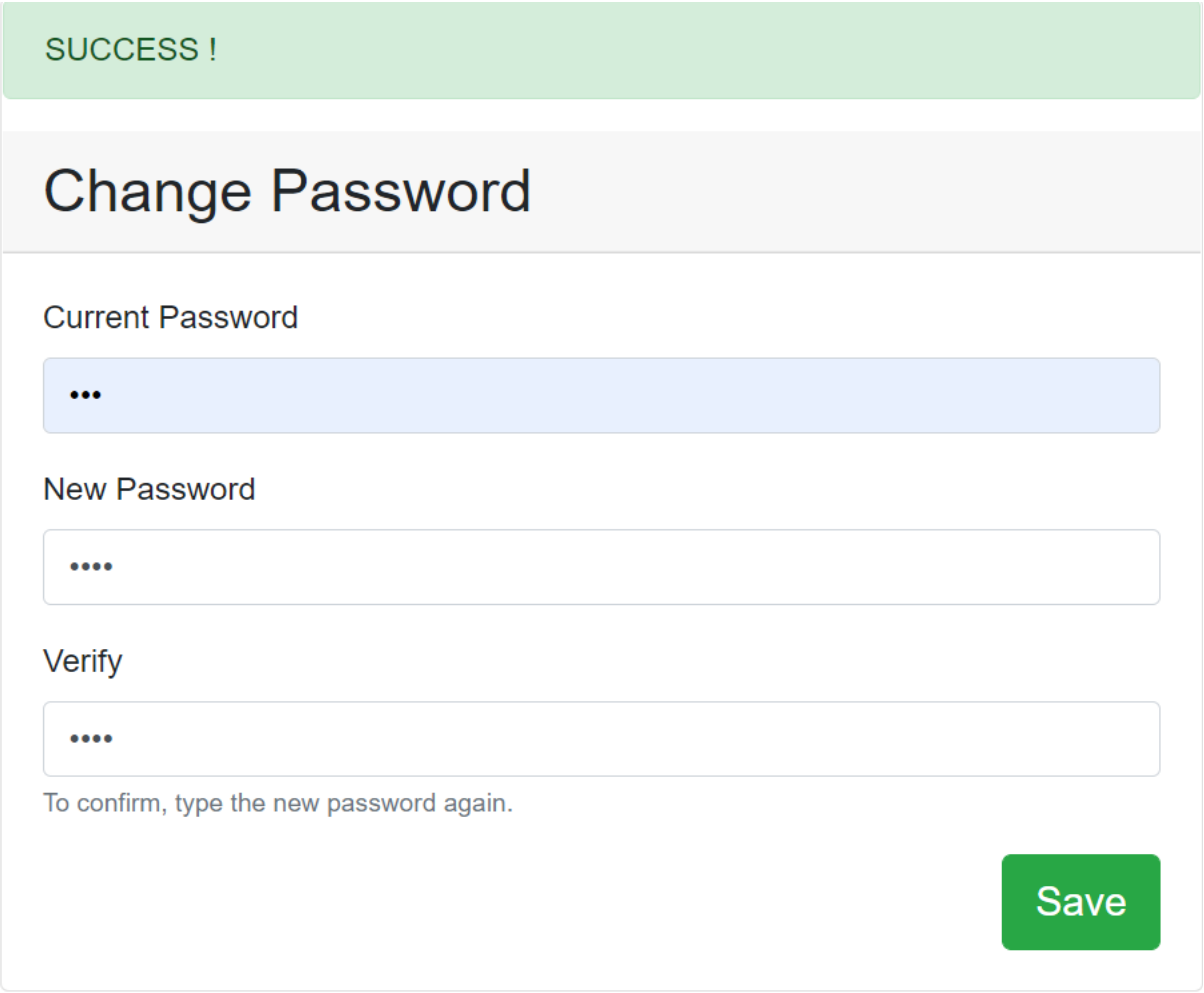
500000.0 vnđ

Adidas nam NITE …

Giày thể thao

400000.0 vnđ

Tìm kiếm sản phẩm



SUCCESS !

Change Password

Current Password

•••

New Password

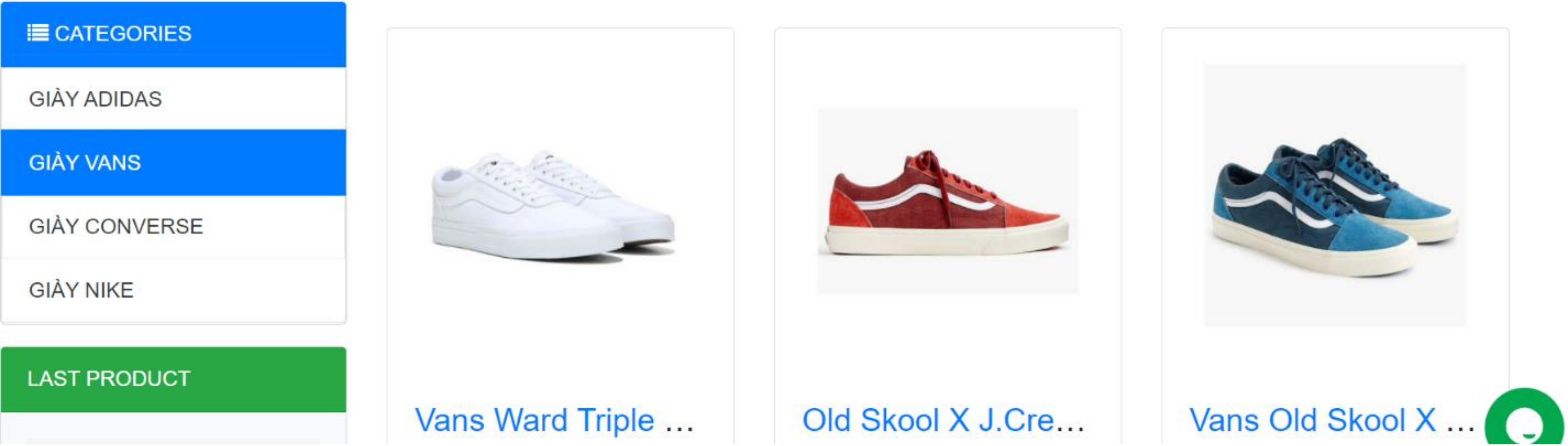
••••

Verify

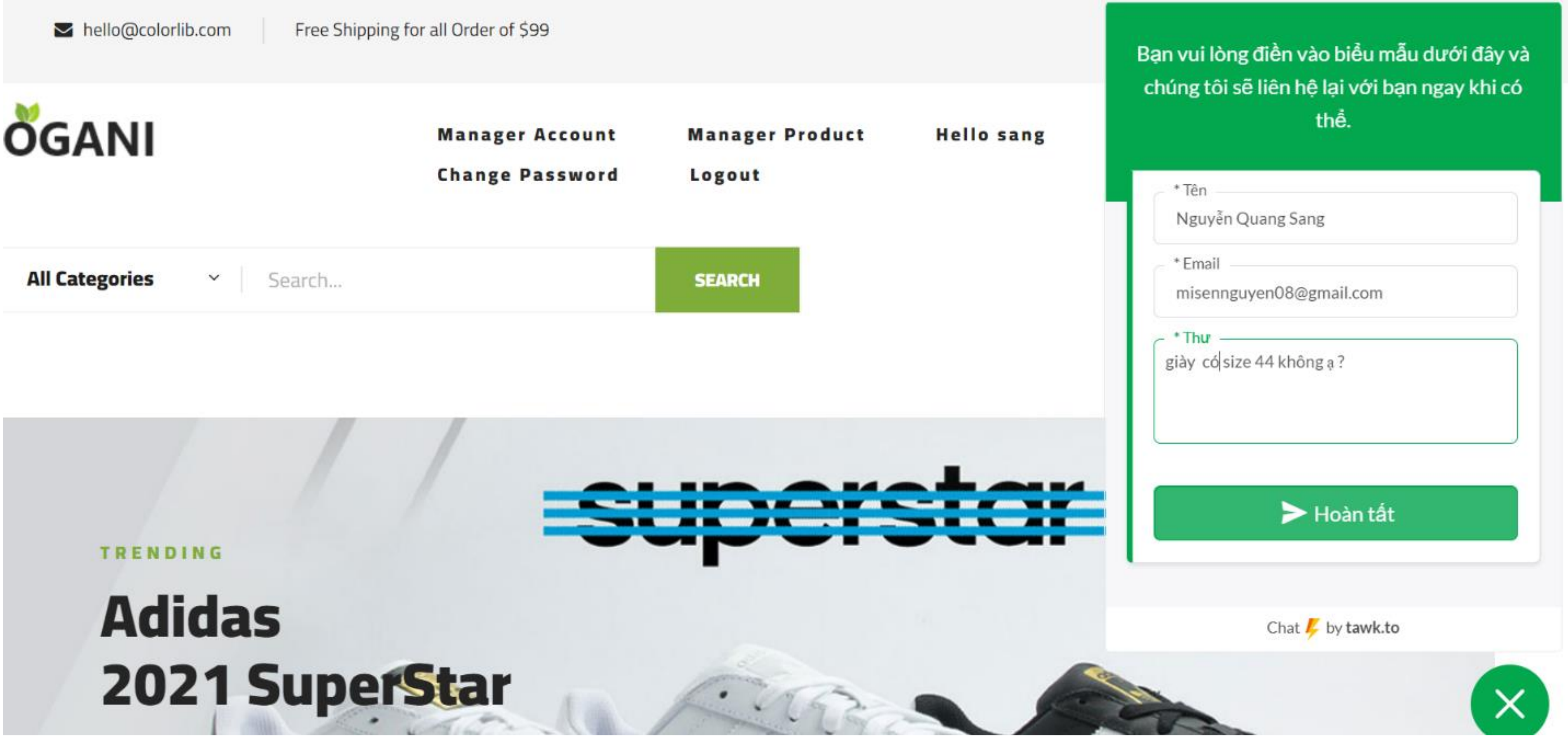
••••

To confirm, type the new password again.

Đổi mật khẩu



Tìm kiếm sản phẩm theo danh mục



Bạn vui lòng điền vào biểu mẫu dưới đây và chúng tôi sẽ liên hệ lại với bạn ngay khi có thể.

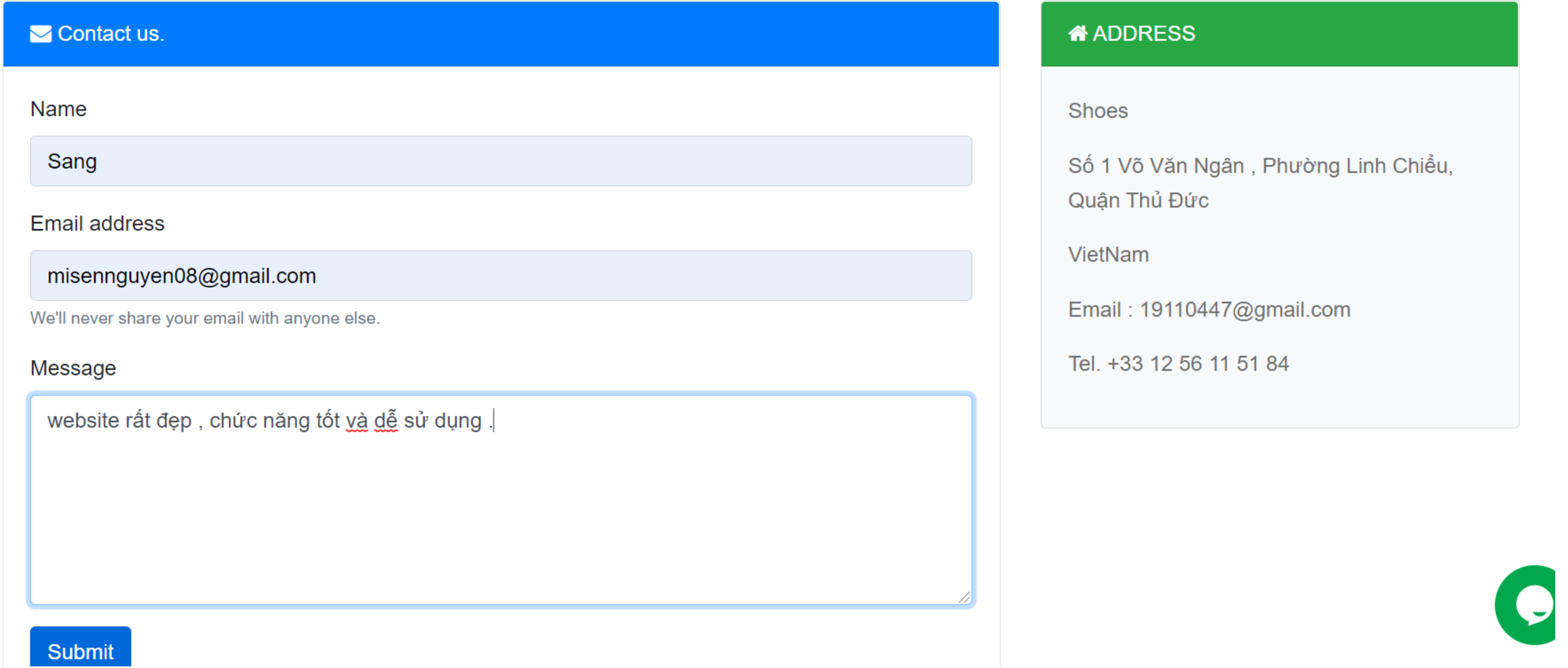
* Tên: Nguyễn Quang Sang

* Email: misennguyen08@gmail.com

* Thư: giày có size 44 không ạ ?

Hoàn tất

Chat với người bán hàng



Contact us.

Name

Sang

Email address

misennguyen08@gmail.com

We'll never share your email with anyone else.

Message

website rất đẹp , chức năng tốt và dễ sử dụng .

ADDRESS

Shoes

Số 1 Võ Văn Ngân , Phường Linh Chiểu,
Quận Thủ Đức

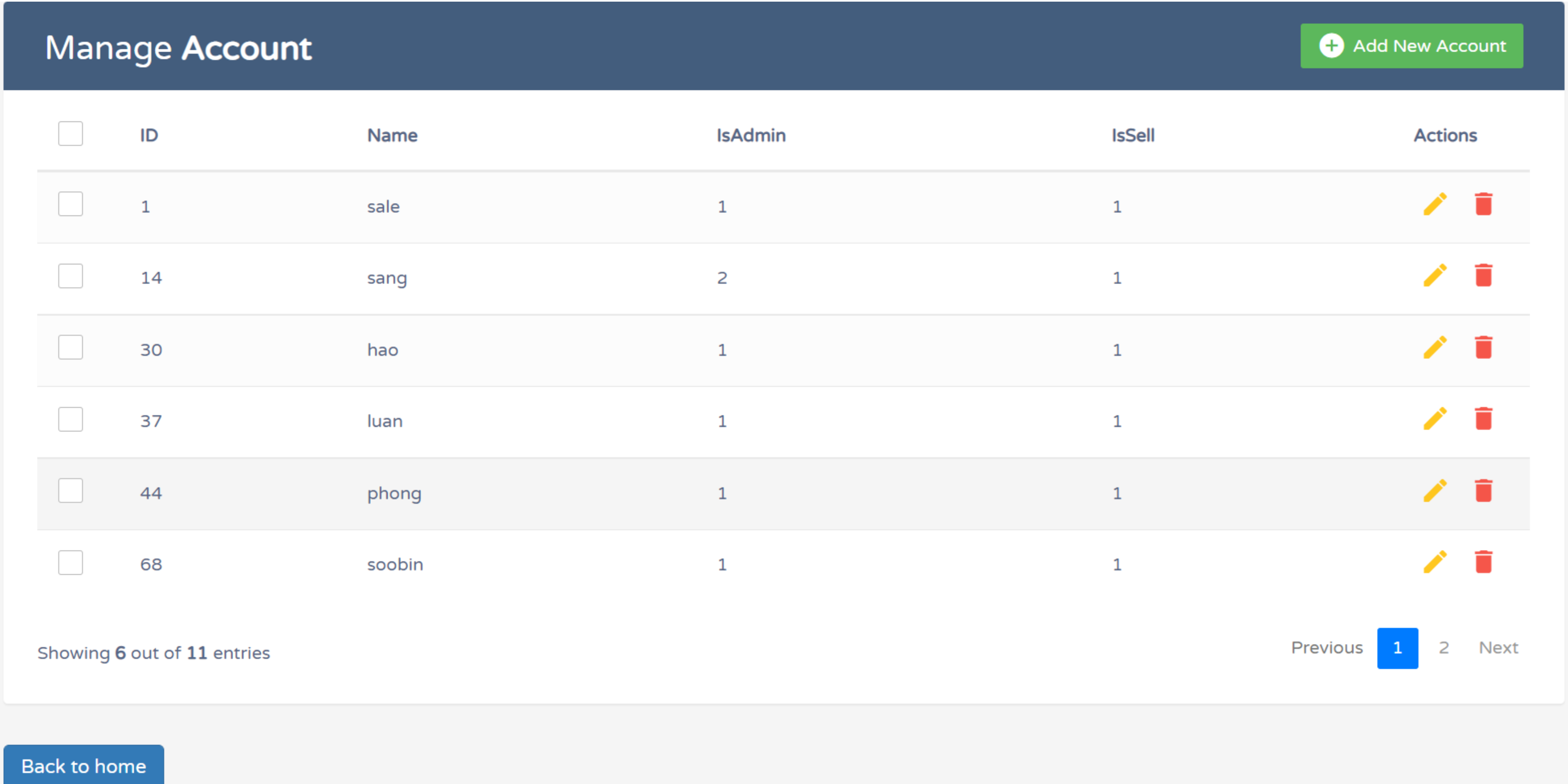
VietNam

Email : 19110447@gmail.com

Tel. +33 12 56 11 51 84

Liên hệ để nhận xét hoặc phản hồi về website

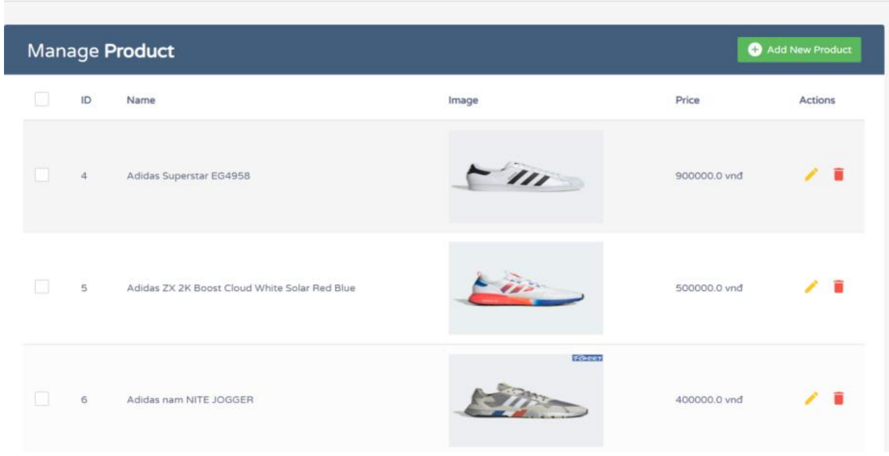
Quản lý tài khoản của admin (Thêm – Xóa – Chỉnh sửa)



□	ID	Name	IsAdmin	IsSell	Actions
□	1	sale	1	1	✏ 🗑
□	14	sang	2	1	✏ 🗑
□	30	hao	1	1	✏ 🗑
□	37	luan	1	1	✏ 🗑
□	44	phong	1	1	✏ 🗑
□	68	soobin	1	1	✏ 🗑

Showing **6** out of **11** entries

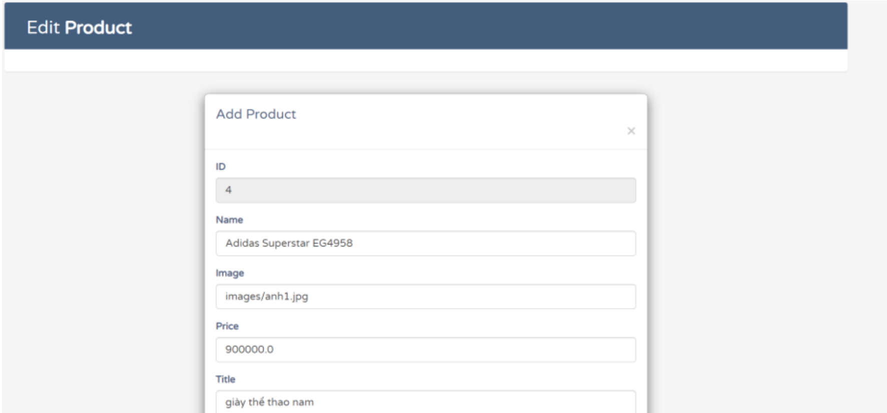
Quản lý sản phẩm (Thêm – Xóa – Chỉnh sửa)



Manage Product

⊕ Add New Product

□	ID	Name	Image	Price	Actions
□	4	Adidas Superstar EG4958		900000.0 vnđ	✎ 🗑
□	5	Adidas ZX 2K Boost Cloud White Solar Red Blue		500000.0 vnđ	✎ 🗑
□	6	Adidas nam NITE JOGGER		400000.0 vnđ	✎ 🗑



Edit Product

Add Product ×

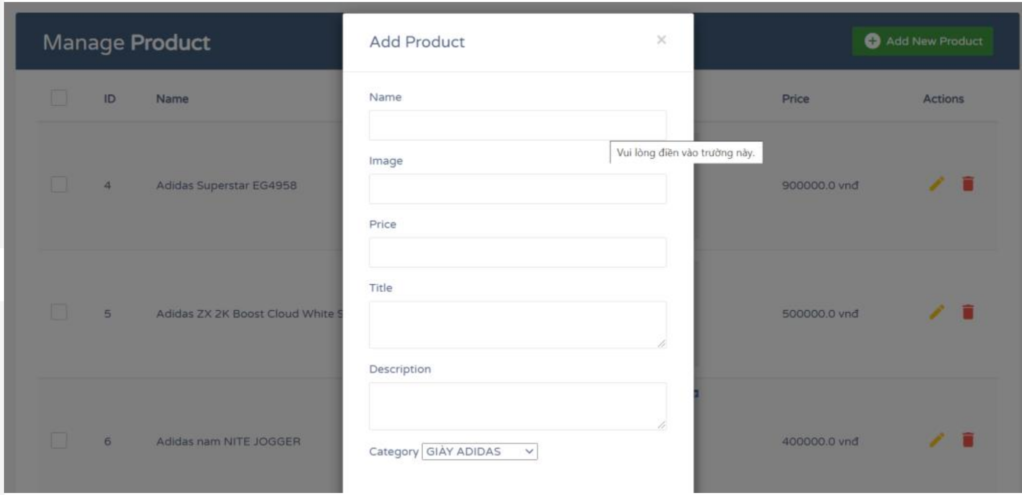
ID: 4

Name: Adidas Superstar EG4958

Image: images/anh1.jpg

Price: 900000.0

Title: giày thể thao nam



Manage Product

⊕ Add New Product

□	ID	Name	Price	Actions
□	4	Adidas Superstar EG4958	900000.0 vnđ	✎ 🗑
□	5	Adidas ZX 2K Boost Cloud White S	500000.0 vnđ	✎ 🗑
□	6	Adidas nam NITE JOGGER	400000.0 vnđ	✎ 🗑

Add Product ×

Name

Vui lòng điền vào trường này.

Image

Price

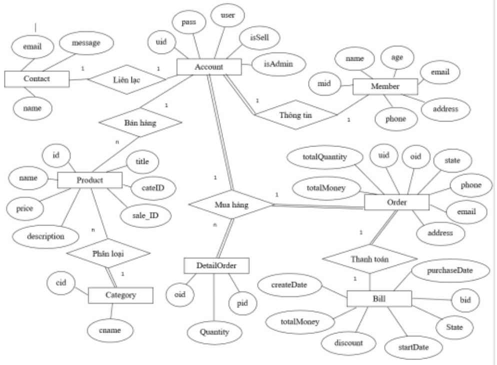
Title

Description

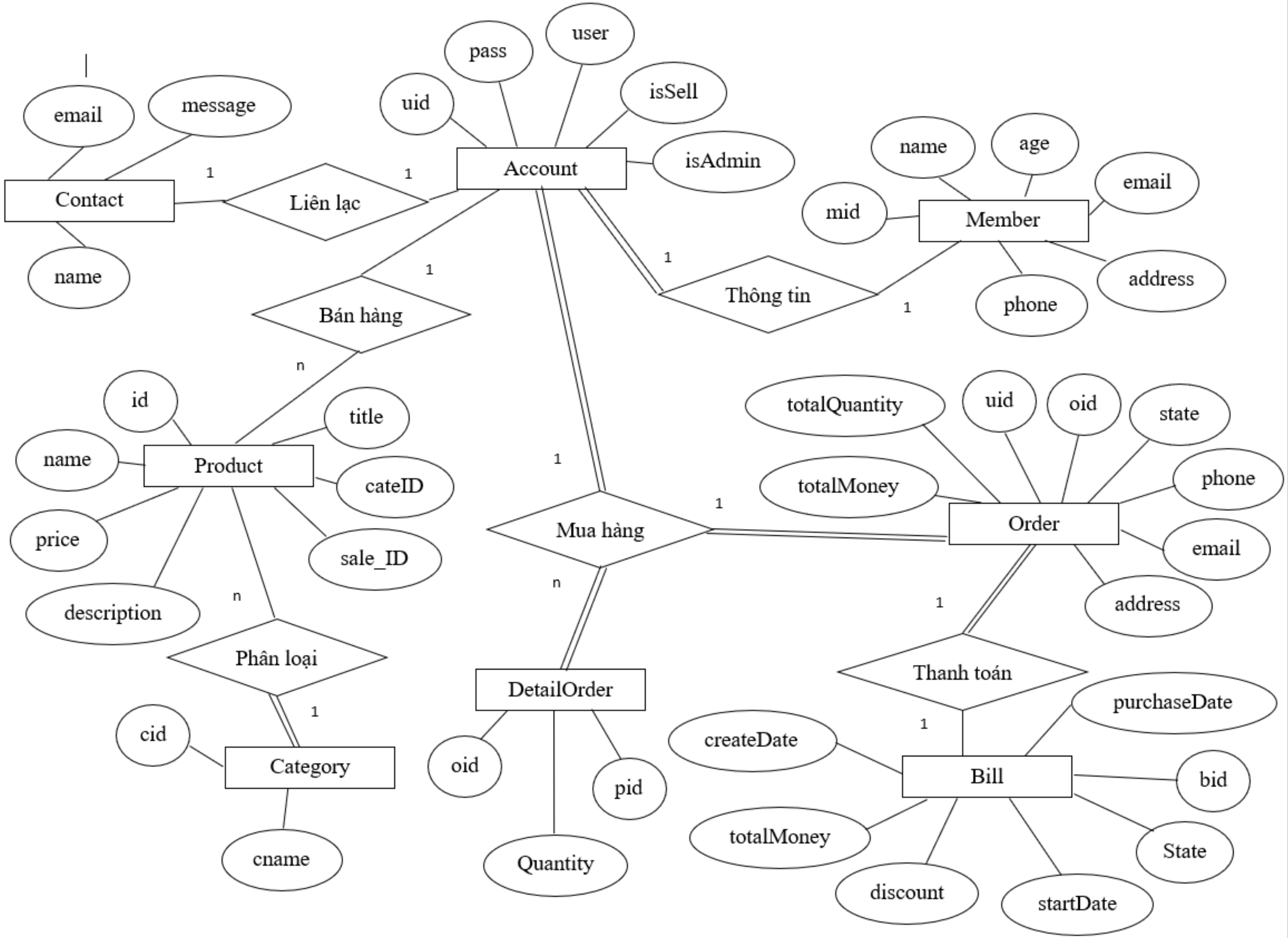
Category: GIÀY ADIDAS

Xây dựng Website kinh doanh giày và phụ kiện

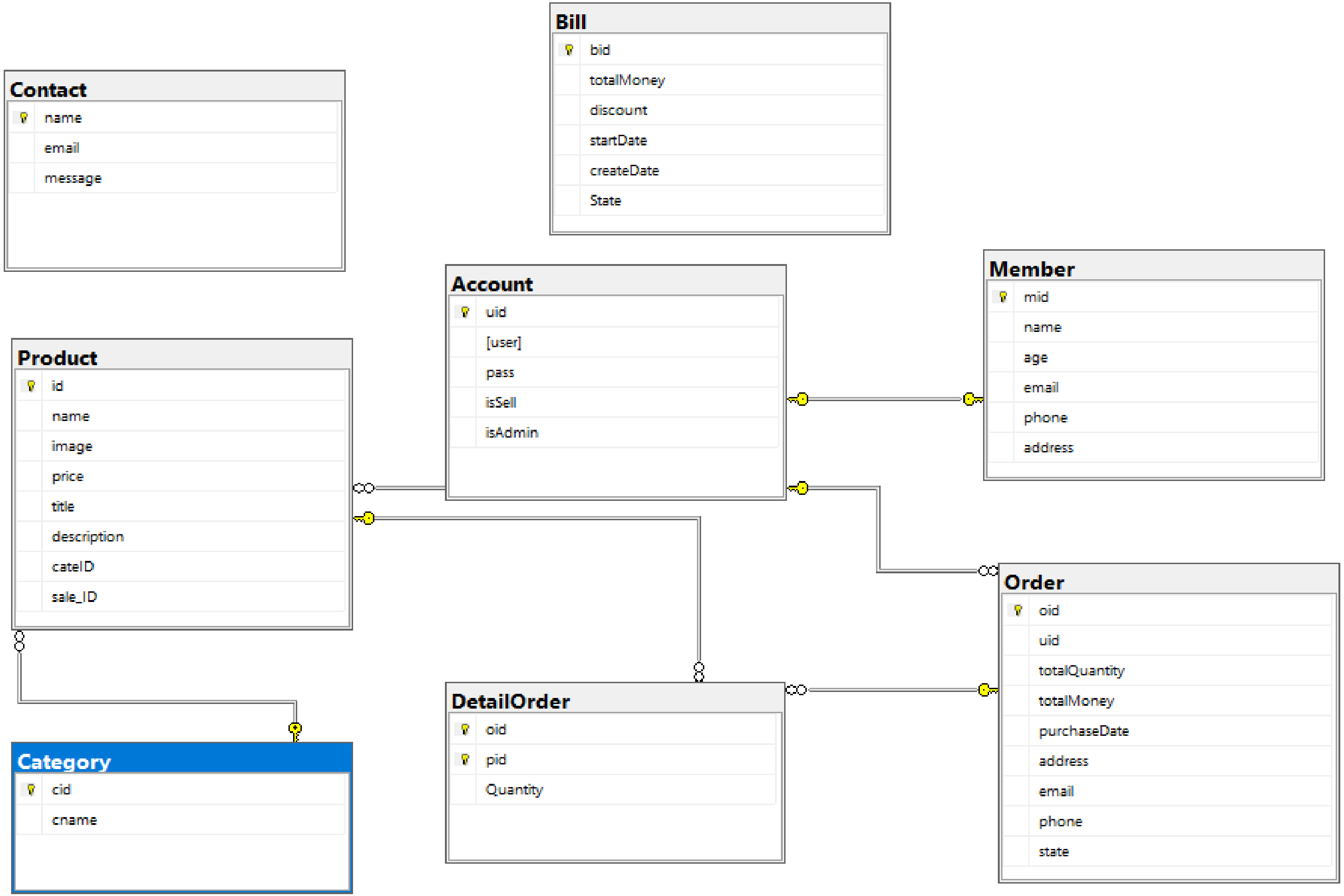
LƯỢC ĐỒ QUAN HỆ



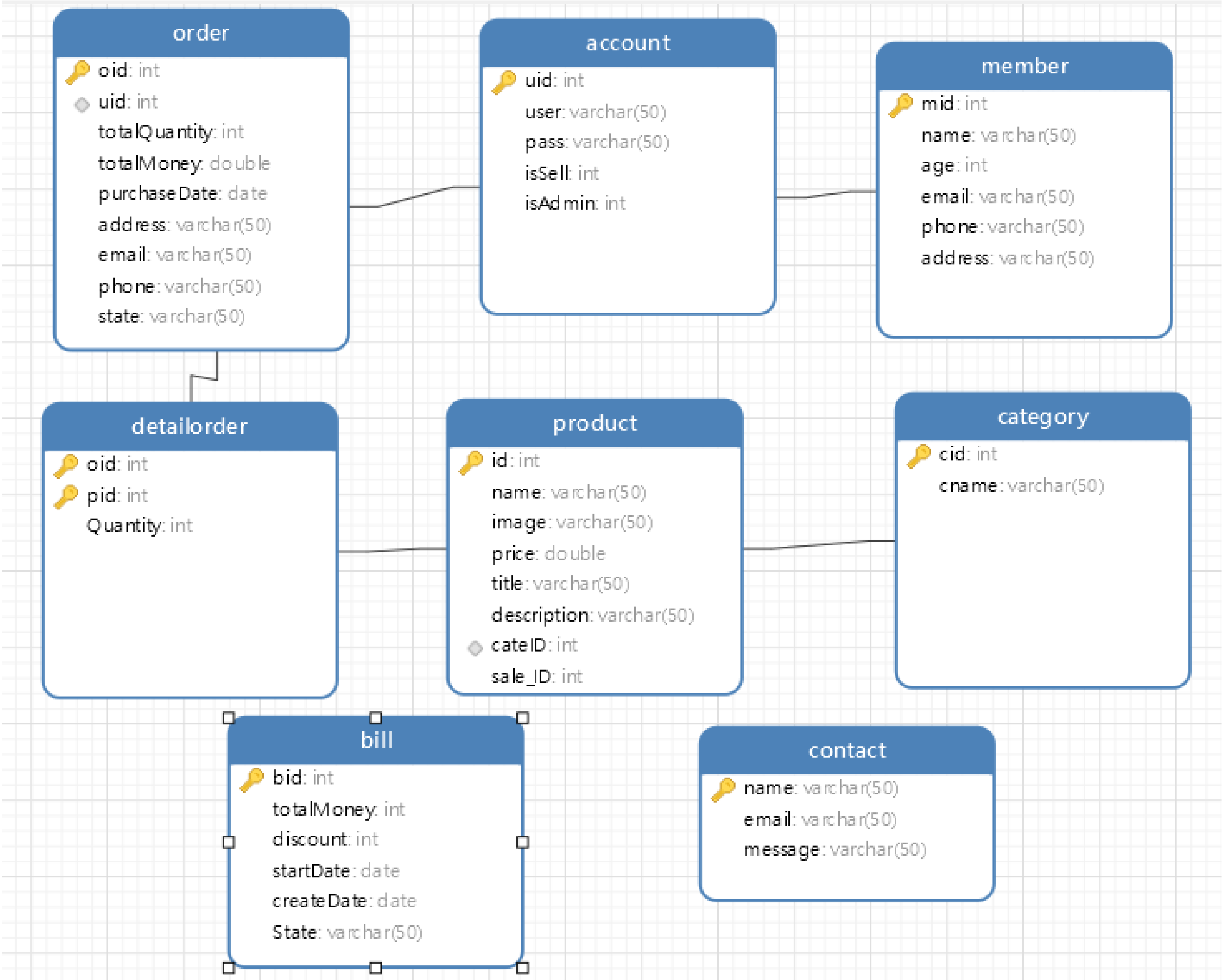
ERD


Contact
email
message
name
Liên lạc
1
1
Account
uid
pass
user
isSell
isAdmin
Bán hàng
1
n
Thông tin
1
1
Member
name
age
email
mid
address
phone
Product
id
title
name
cateID
price
sale_ID
description
n
Phân loại
1
Category
cid
cname
1
Mua hàng
1
n
Order
totalQuantity
uid
oid
state
totalMoney
phone
email
address
DetailOrder
oid
pid
Quantity
1
Thanh toán
1
Bill
purchaseDate
createDate
bid
totalMoney
State
discount
startDate

DATABASE DIAGRAM



DATABASE DIAGRAM



order
oid: int
uid: int
totalQuantity: int
totalMoney: double
purchaseDate: date
address: varchar(50)
email: varchar(50)
phone: varchar(50)
state: varchar(50)

account
uid: int
user: varchar(50)
pass: varchar(50)
isSell: int
isAdmin: int

member
mid: int
name: varchar(50)
age: int
email: varchar(50)
phone: varchar(50)
address: varchar(50)

detailorder
oid: int
pid: int
Quantity: int

product
id: int
name: varchar(50)
image: varchar(50)
price: double
title: varchar(50)
description: varchar(50)
cateID: int
sale_ID: int

category
cid: int
cname: varchar(50)

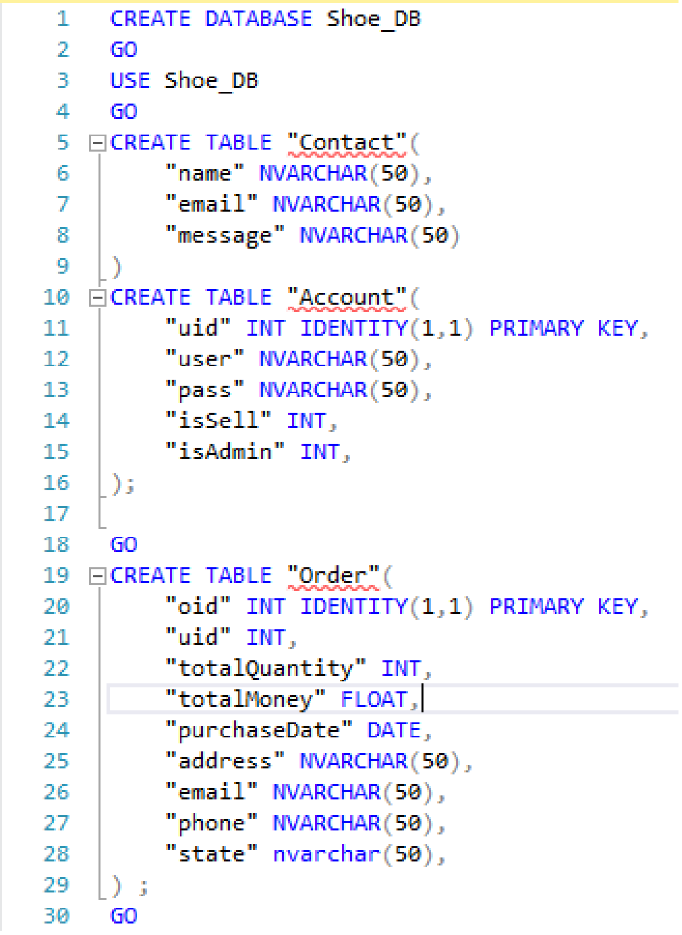
bill
bid: int
totalMoney: int
discount: int
startDate: date
createDate: date
State: varchar(50)

contact
name: varchar(50)
email: varchar(50)
message: varchar(50)

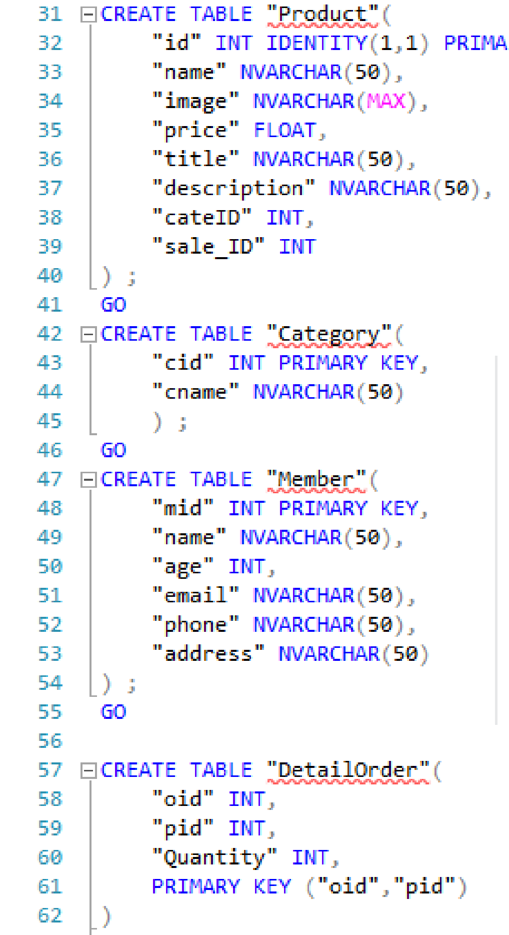
Xây dựng Website kinh doanh giày và phụ kiện

CÀI ĐẶT CSDL

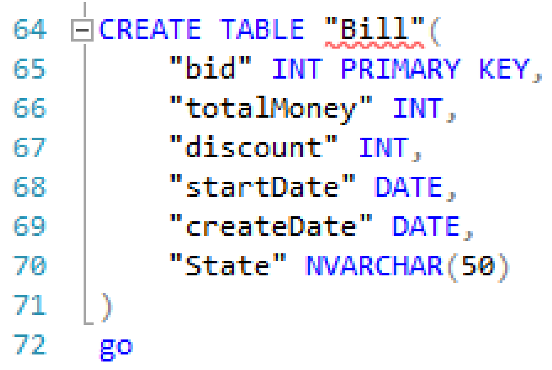
CODE TẠO BẢNG



```sql
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
```

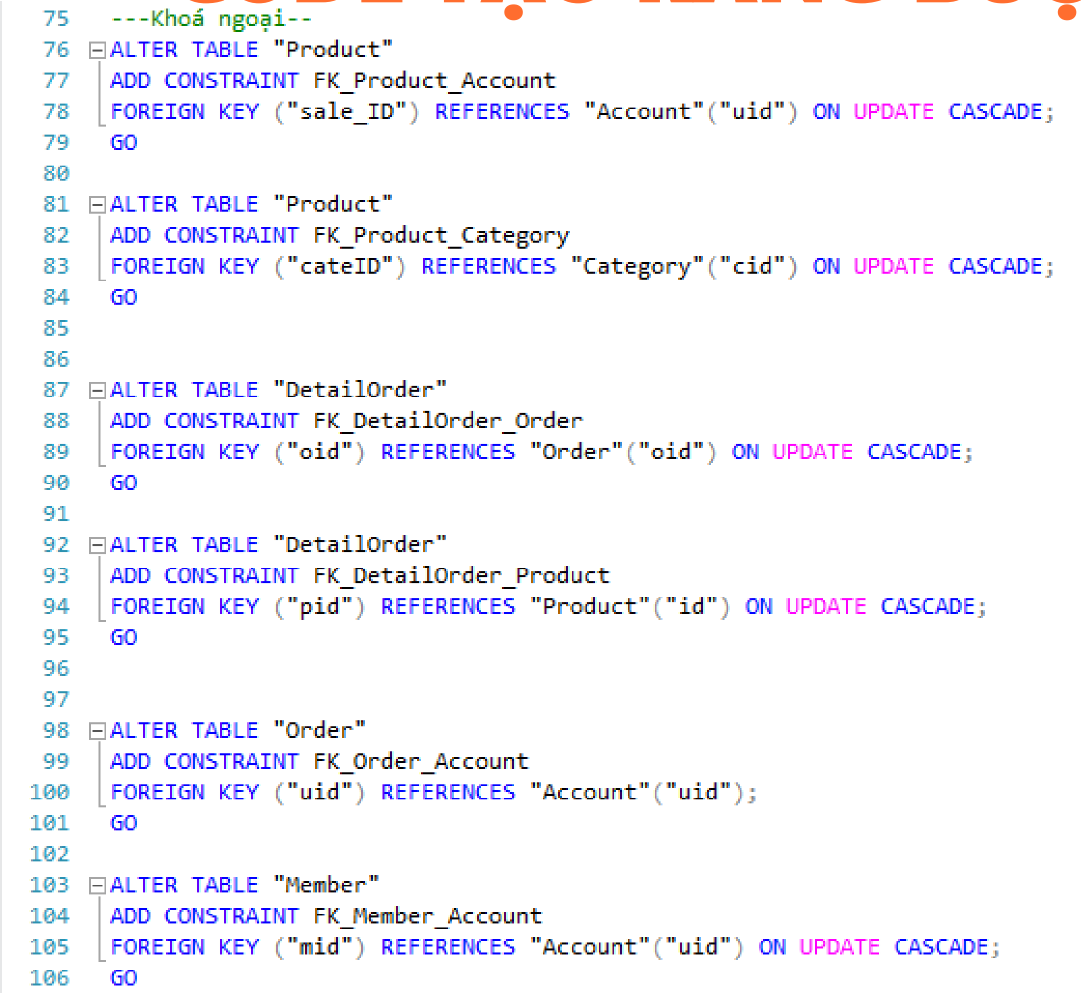


```sql
CREATE TABLE "Product"(
    "id" INT IDENTITY(1,1) PRIMARY KEY,
    "name" NVARCHAR(50),
    "image" NVARCHAR(MAX),
    "price" FLOAT,
    "title" NVARCHAR(50),
    "description" NVARCHAR(50),
    "cateID" INT,
    "sale_ID" INT
) ;
GO
CREATE TABLE "Category"(
    "cid" INT PRIMARY KEY,
    "cname" 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")
)
```



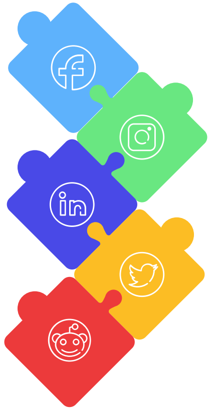
```sql
CREATE TABLE "Bill"(
    "bid" INT PRIMARY KEY,
    "totalMoney" INT,
    "discount" INT,
    "startDate" DATE,
    "createDate" DATE,
    "State" NVARCHAR(50)
)
go
```

CODE TẠO RÀNG BUỘC KHÓA NGOẠI



```sql
---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
```

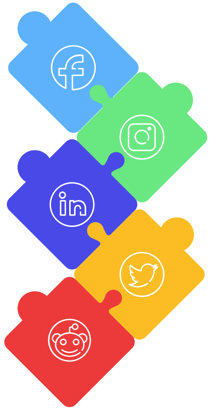
VIEWS

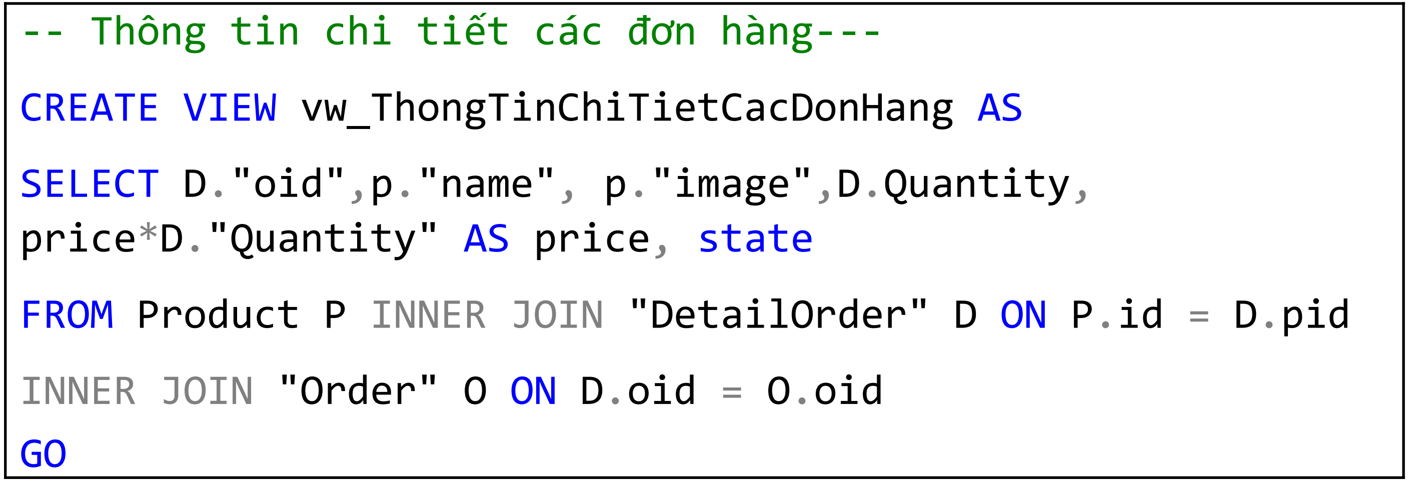


```sql
-- Hiển thị thông tin sản phẩm
CREATE VIEW vw_ThongTinSanPham AS
SELECT "name", "image", price, title, "description"
FROM Product;
GO
```

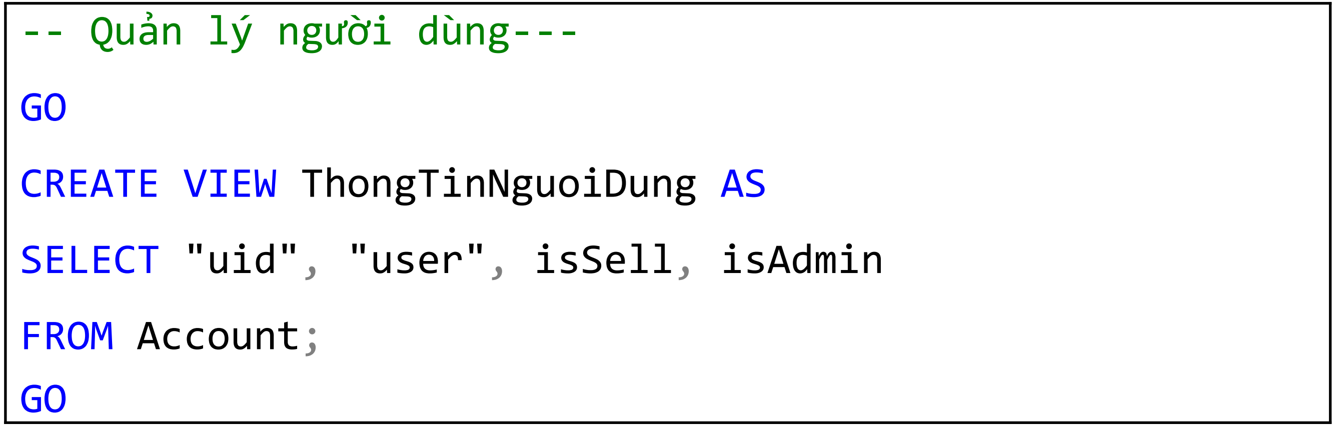
```sql
-- Quản lý thông tin cá nhân
CREATE VIEW vw_ThongTinCaNhan AS
SELECT name, age, email, phone, address
FROM Member;
GO
```

VIEWS



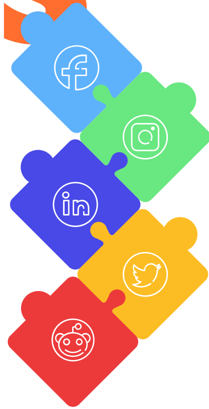


```
-- Thông tin chi tiết các đơn hàng---

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
```



```
-- Quản lý người dùng---

GO

CREATE VIEW ThongTinNguoiDung AS

SELECT "uid", "user", isSell, isAdmin

FROM Account;

GO
```

STORED PROCEDURES



Account

Product

Bill

DetailOrder

Categoty

Contact

Member

Order

STORED PROCEDURES

ACCOUNT

[SP_GETALLACCOUNT]

[SP_GETONEACCOUNT]

[SP_DELETEACCOUNT]

[SP_INSERTACCOUNT]

[SP_UPDATEACCOUNT]

ACCOUNT

STORED PROCEDURES

```
--lấy tất cả Account---
CREATE PROC sp_getAllAccount
AS
BEGIN
    SELECT
            "uid",
            "user",
            "pass",
            "isSell",
            "isAdmin"
    FROM
        "Account"
    ORDER BY
        "uid";
END;
go
```

[SP_GETALLACCOUNT]

STORED PROCEDURES

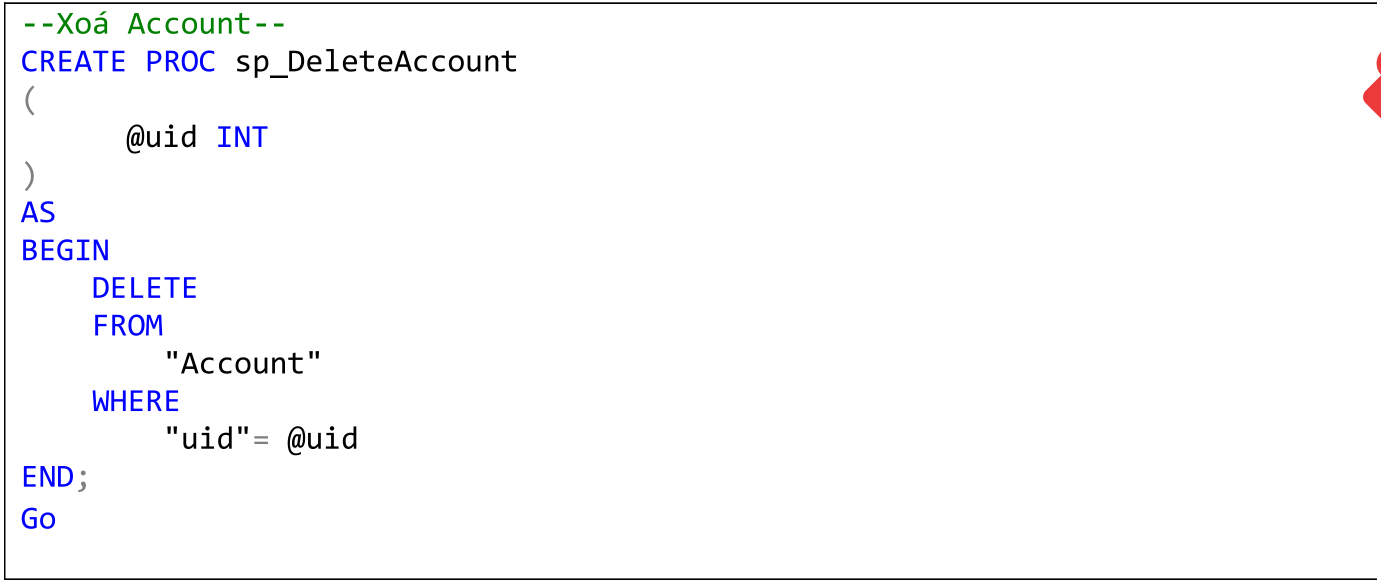
ACCOUNT

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

[SP_GETONEACCOUNT]

ACCOUNT

STORED PROCEDURES

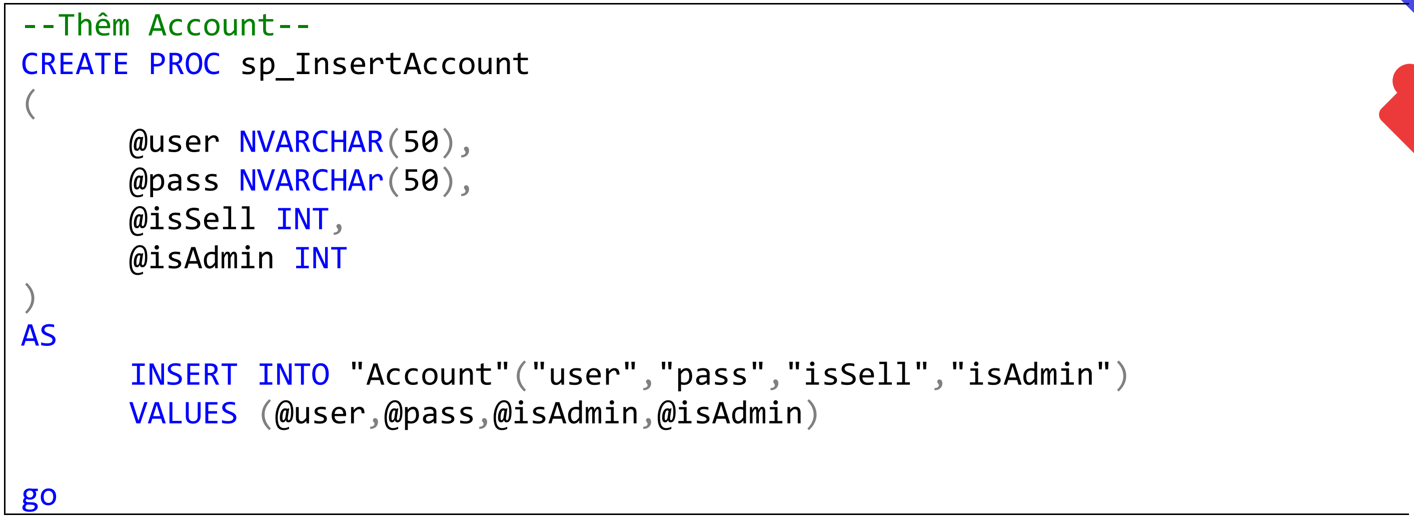


```
--Xoá Account--
CREATE PROC sp_DeleteAccount
(
      @uid INT
)
AS
BEGIN
    DELETE
    FROM
        "Account"
    WHERE
        "uid"= @uid
END;
Go
```

[SP_DELETEACCOUNT]

STORED PROCEDURES

ACCOUNT

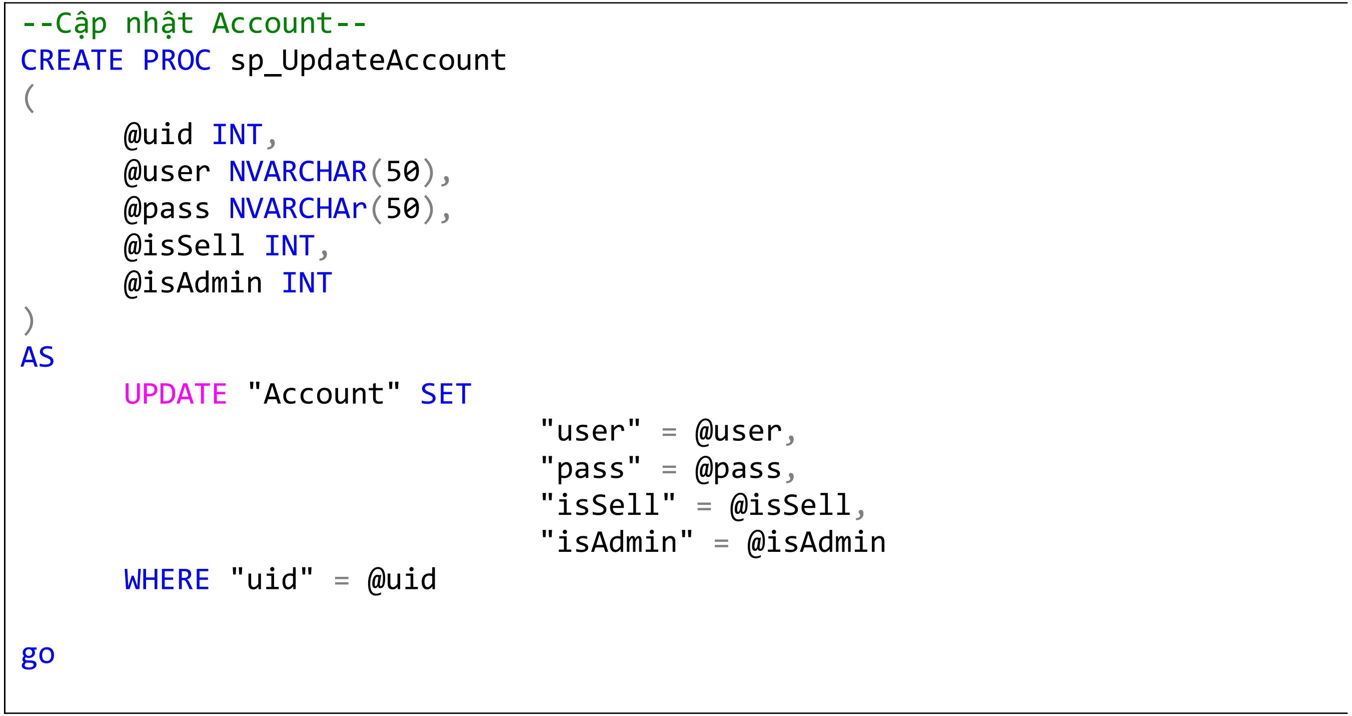


```
--Thêm Account--
CREATE PROC sp_InsertAccount
(
        @user NVARCHAR(50),
        @pass NVARCHAr(50),
        @isSell INT,
        @isAdmin INT
)
AS
        INSERT INTO "Account"("user","pass","isSell","isAdmin")
        VALUES (@user,@pass,@isAdmin,@isAdmin)

go
```

[SP_INSERTACCOUNT]

ACCOUNT

STORED PROCEDURES



```
--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
```

[SP_UPDATEACCOUNT]

STORED PROCEDURES

BILL

[SP_GETALLBILL]

[SP_GETONEBILL]

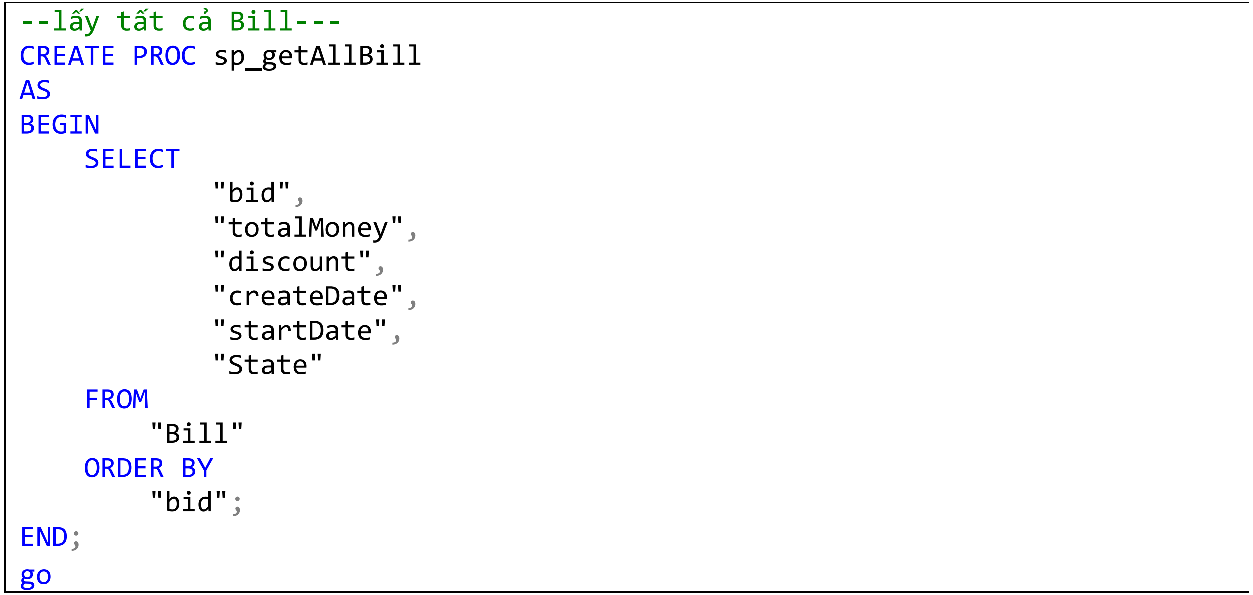
[SP_DELETEBILL]

[SP_INSERTBILL]

[SP_UPDATEBILL]

STORED PROCEDURES

BILL

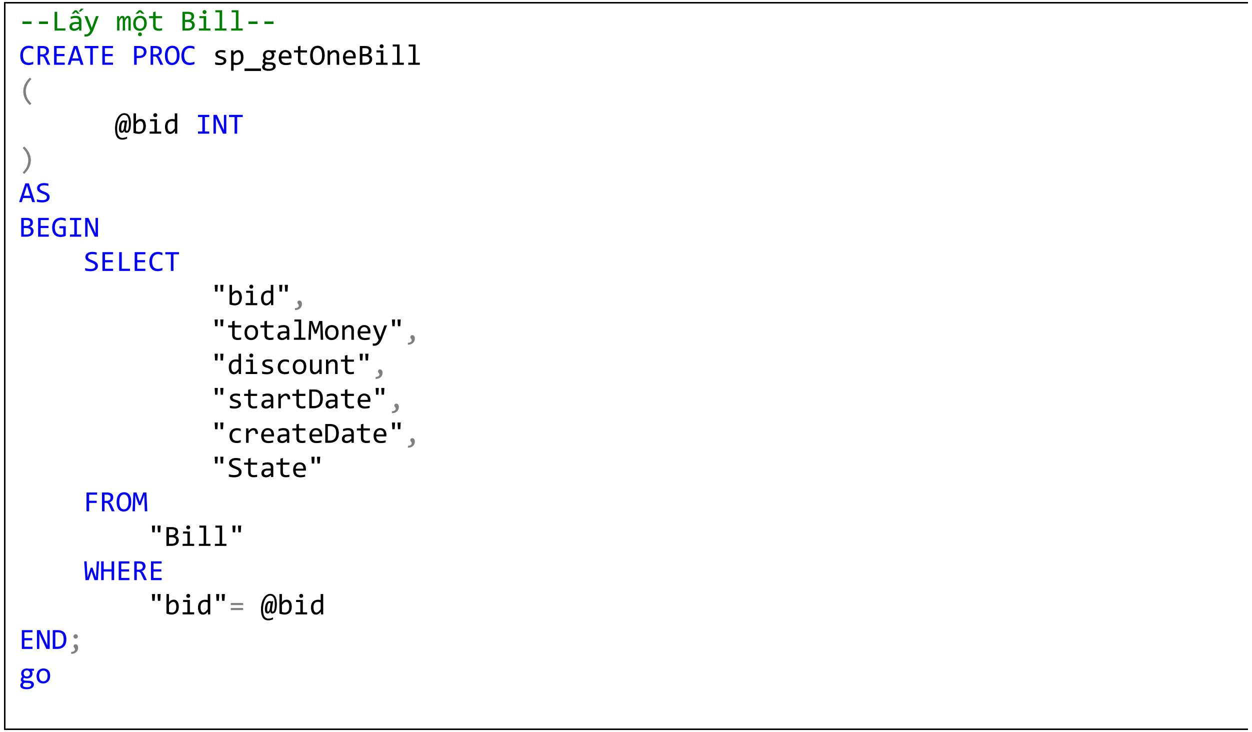


```
--lấy tất cả Bill---
CREATE PROC sp_getAllBill
AS
BEGIN
    SELECT
            "bid",
            "totalMoney",
            "discount",
            "createDate",
            "startDate",
            "State"
    FROM
        "Bill"
    ORDER BY
        "bid";
END;
go
```

[SP_GETALLBILL]

BILL

STORED
PROCEDURES

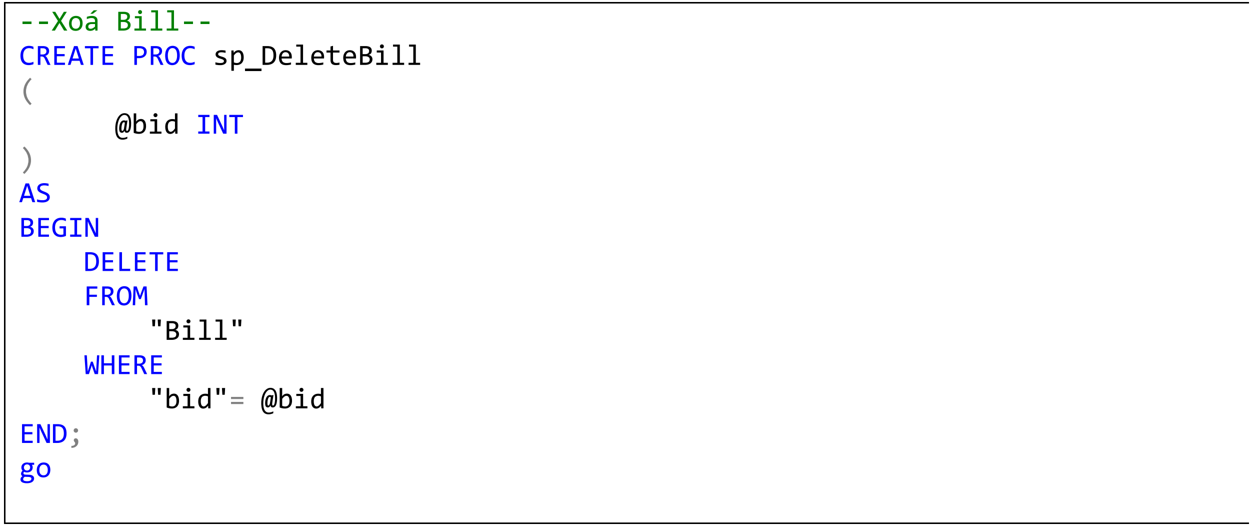


```
--Lấy một Bill--
CREATE PROC sp_getOneBill
(
	@bid INT
)
AS
BEGIN
    SELECT
            "bid",
            "totalMoney",
            "discount",
            "startDate",
            "createDate",
            "State"
    FROM
        "Bill"
    WHERE
        "bid"= @bid
END;
go
```

[SP_GETONEBILL]

STORED PROCEDURES

BILL

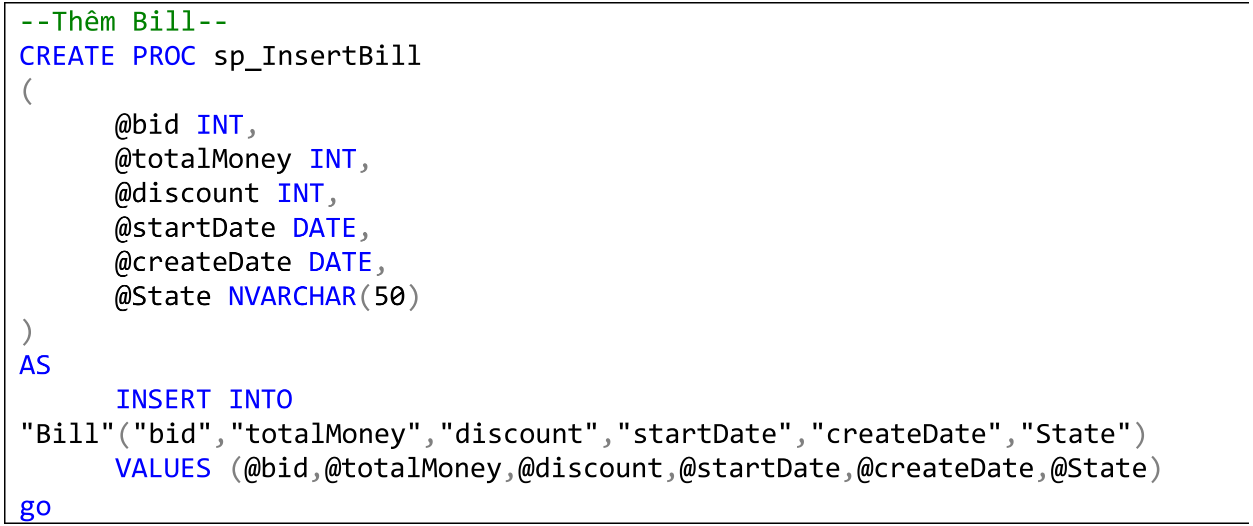


```
--Xoá Bill--
CREATE PROC sp_DeleteBill
(
	@bid INT
)
AS
BEGIN
    DELETE
    FROM
        "Bill"
    WHERE
        "bid"= @bid
END;
go
```

[SP_DELETEBILL]

BILL

STORED PROCEDURES

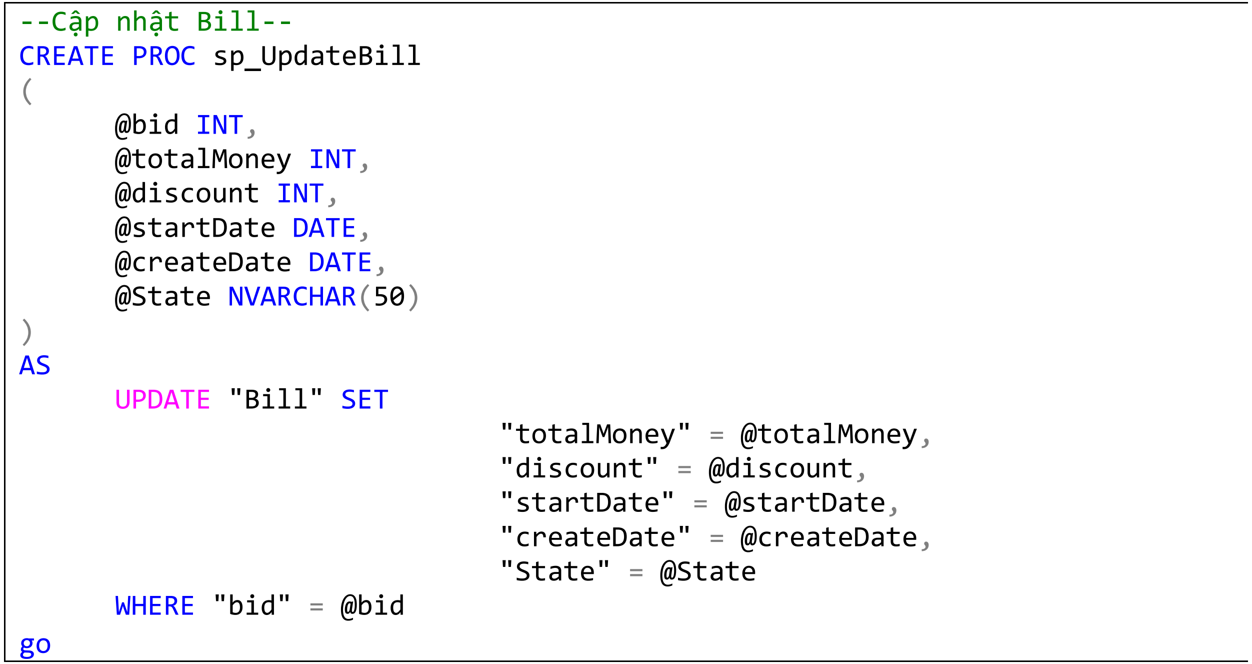


```sql
--Thêm Bill--
CREATE PROC sp_InsertBill
(
      @bid INT,
      @totalMoney INT,
      @discount INT,
      @startDate DATE,
      @createDate DATE,
      @State NVARCHAR(50)
)
AS
      INSERT INTO
"Bill"("bid","totalMoney","discount","startDate","createDate","State")
      VALUES (@bid,@totalMoney,@discount,@startDate,@createDate,@State)
go
```

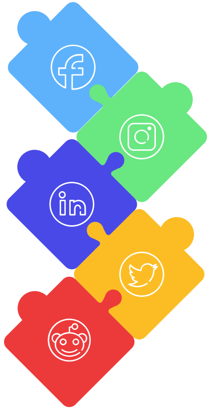
[SP_INSERTBILL]

STORED PROCEDURES

BILL



```
--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
```



[SP_UPDATEBILL]

STORED PROCEDURES

CATEGORY

[SP_GETALLCATEGORY]

[SP_GETONECATEGORY]

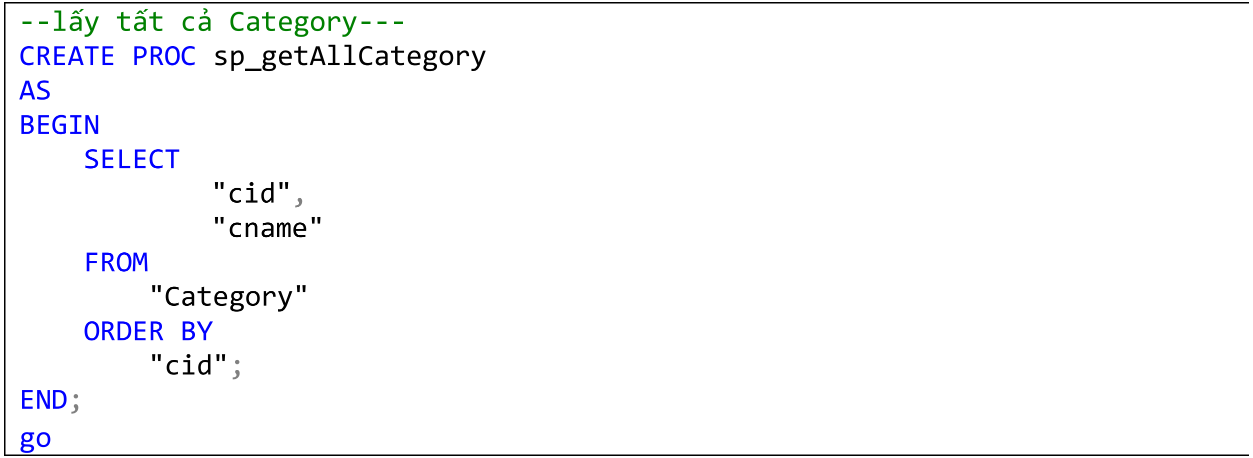
[SP_DELETECATEGORY]

[SP_INSERTCATEGORY]

[SP_UPDATECATEGORY]

CATEGORY

STORED PROCEDURES

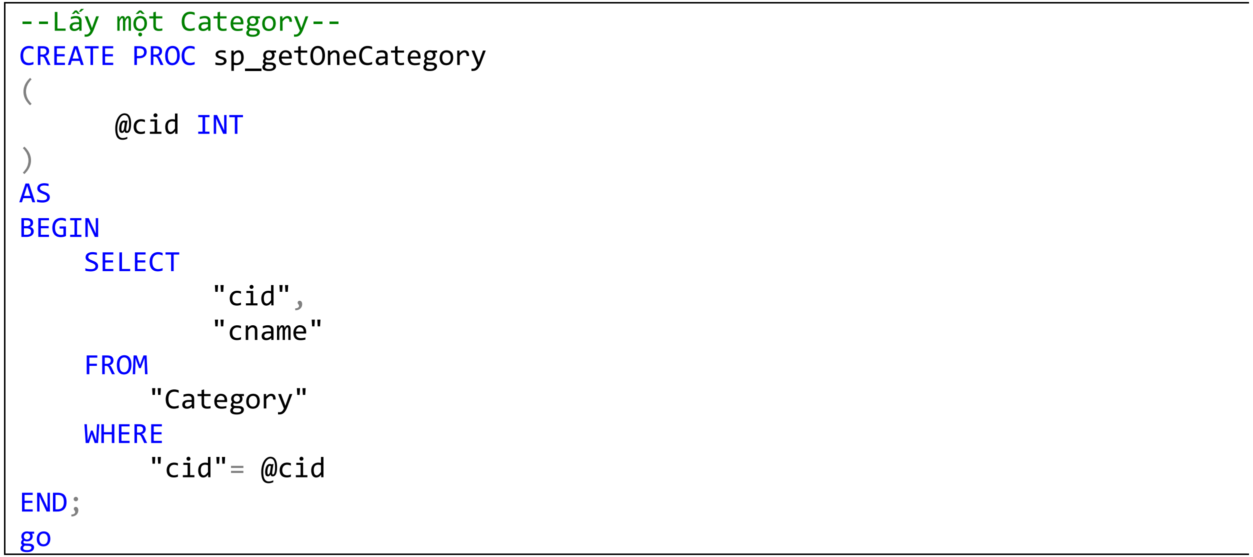


```
--lấy tất cả Category---
CREATE PROC sp_getAllCategory
AS
BEGIN
    SELECT
            "cid",
            "cname"
    FROM
        "Category"
    ORDER BY
        "cid";
END;
go
```

[SP_GETALLCATEGORY]

CATEGORY

STORED PROCEDURES

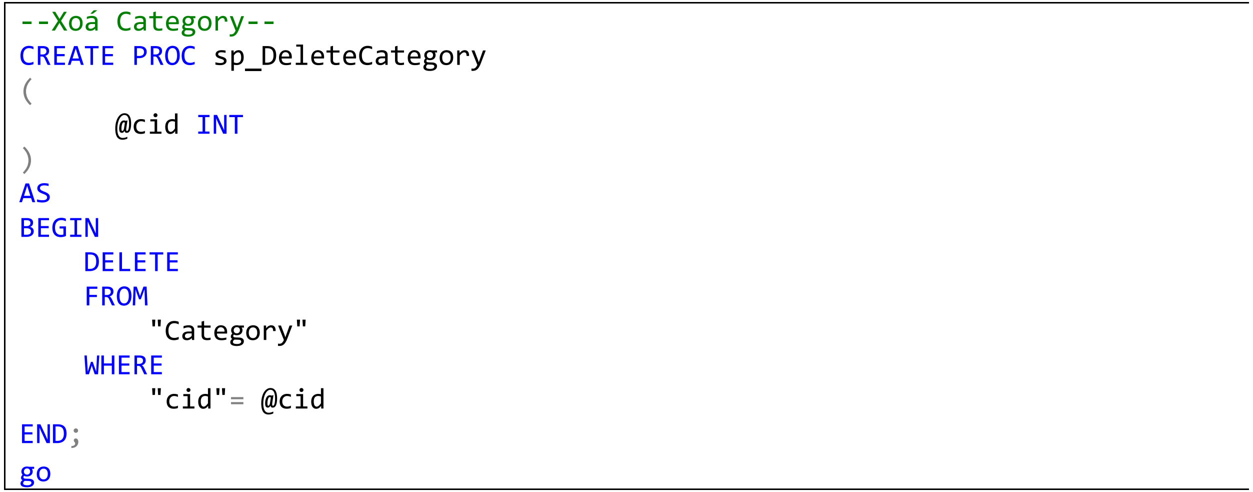


```sql
--Lấy một Category--
CREATE PROC sp_getOneCategory
(
        @cid INT
)
AS
BEGIN
     SELECT
             "cid",
             "cname"
     FROM
          "Category"
     WHERE
          "cid"= @cid
END;
go
```

[SP_GETONECATEGORY]

CATEGORY

STORED PROCEDURES



```
--Xoá Category--
CREATE PROC sp_DeleteCategory
(
	@cid INT
)
AS
BEGIN
    DELETE
    FROM
        "Category"
    WHERE
        "cid"= @cid
END;
go
```

[SP_DELETECATEGORY]

CATEGORY

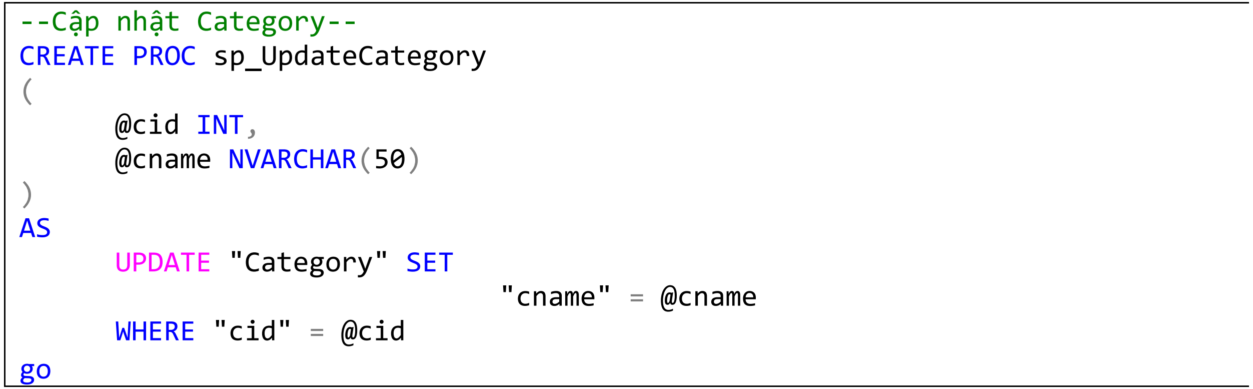
STORED PROCEDURES

```
--Thêm Category--
CREATE PROC sp_InsertCategory
(
	@cid INT,
	@cname NVARCHAR(50)
)
AS
	INSERT INTO "Category"("cid","cname")
	VALUES (@cid,@cname)
go
```

[SP_INSERTCATEGORY]

CATEGORY

STORED PROCEDURES



```
--Cập nhật Category--
CREATE PROC sp_UpdateCategory
(
	@cid INT,
	@cname NVARCHAR(50)
)
AS
	UPDATE "Category" SET
					"cname" = @cname
	WHERE "cid" = @cid
go
```

[SP_UPDATECATEGORY]

STORED PROCEDURES

CONTACT

[SP_GETALLCONTACT]

[SP_GETONECONTACT]

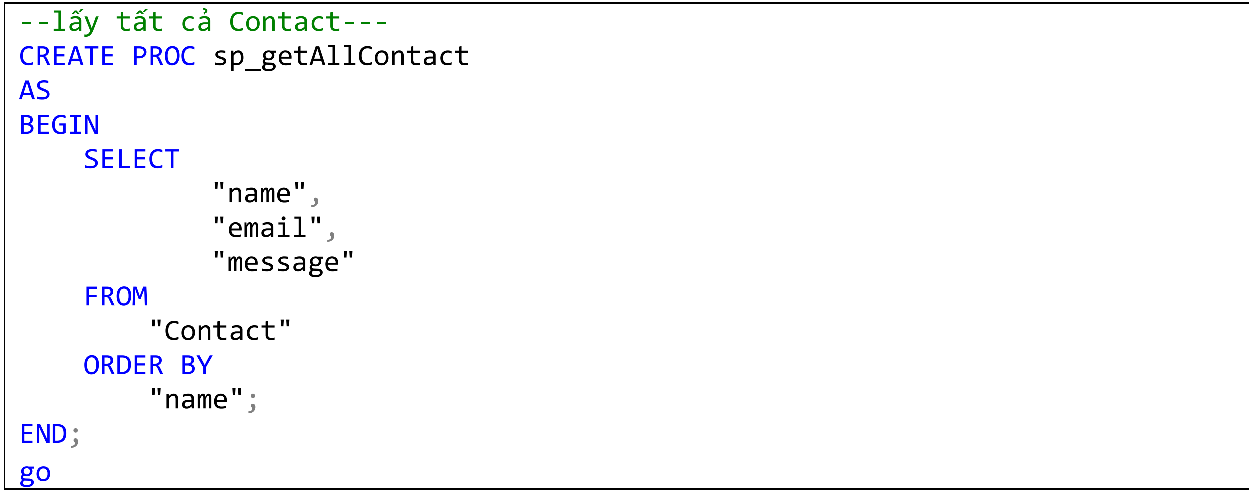
[SP_DELETECONTACT]

[SP_INSERTCONTACT]

[SP_UPDATECONTACT]

CONTACT

STORED PROCEDURES

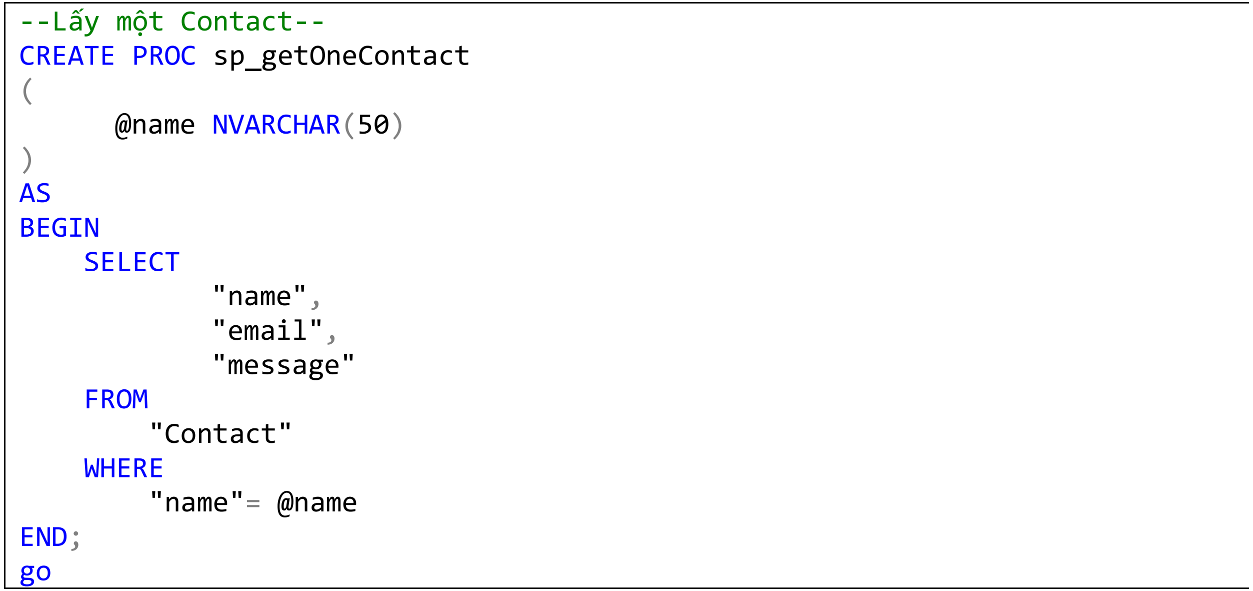


```
--lấy tất cả Contact---
CREATE PROC sp_getAllContact
AS
BEGIN
    SELECT
            "name",
            "email",
            "message"
    FROM
        "Contact"
    ORDER BY
        "name";
END;
go
```

[SP_GETALLCONTACT]

STORED PROCEDURES

CONTACT

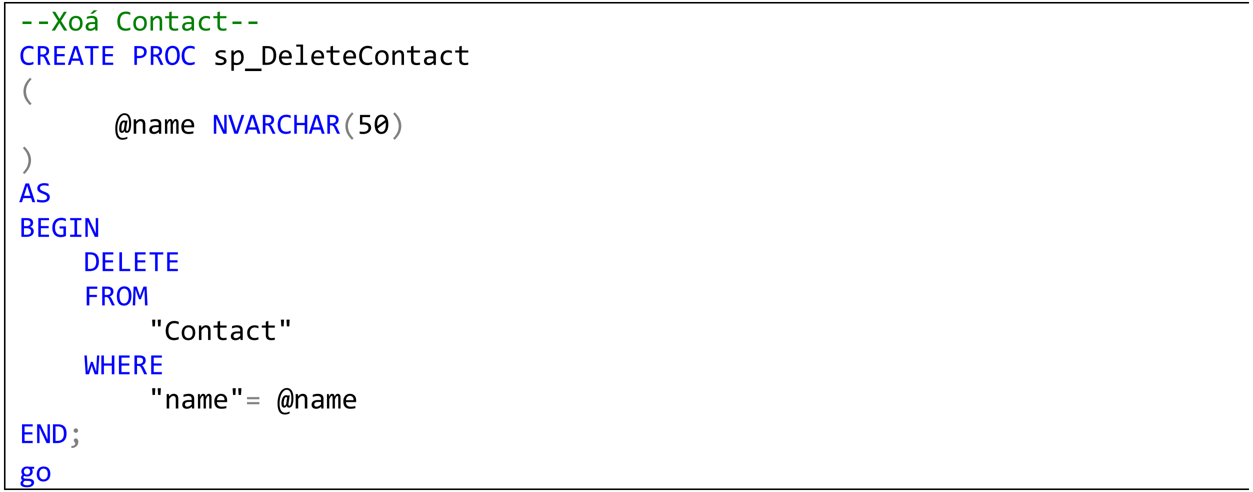


```
--Lấy một Contact--
CREATE PROC sp_getOneContact
(
      @name NVARCHAR(50)
)
AS
BEGIN
    SELECT
            "name",
            "email",
            "message"
    FROM
        "Contact"
    WHERE
        "name"= @name
END;
go
```

[SP_GETONECONTACT]

CONTACT

STORED PROCEDURES

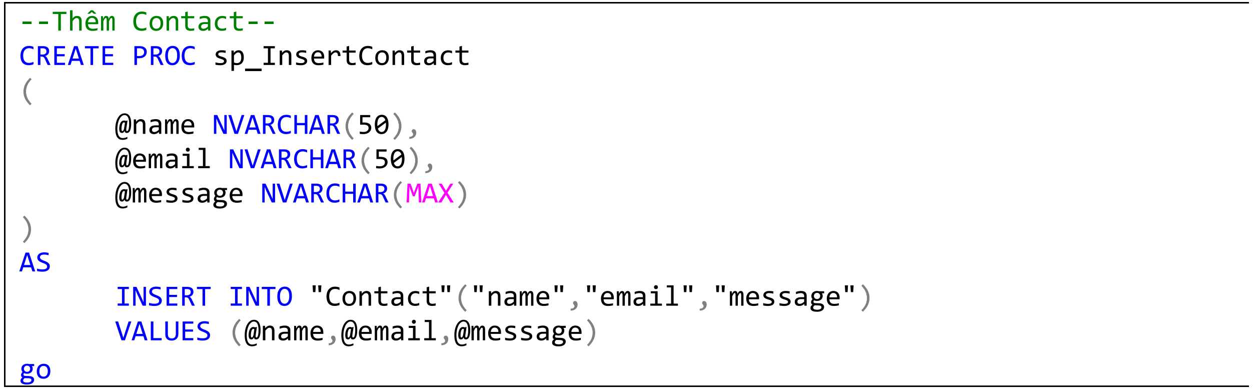


```
--Xoá Contact--
CREATE PROC sp_DeleteContact
(
	@name NVARCHAR(50)
)
AS
BEGIN
    DELETE
    FROM
        "Contact"
    WHERE
        "name"= @name
END;
go
```

[SP_DELETECONTACT]

CONTACT

STORED PROCEDURES

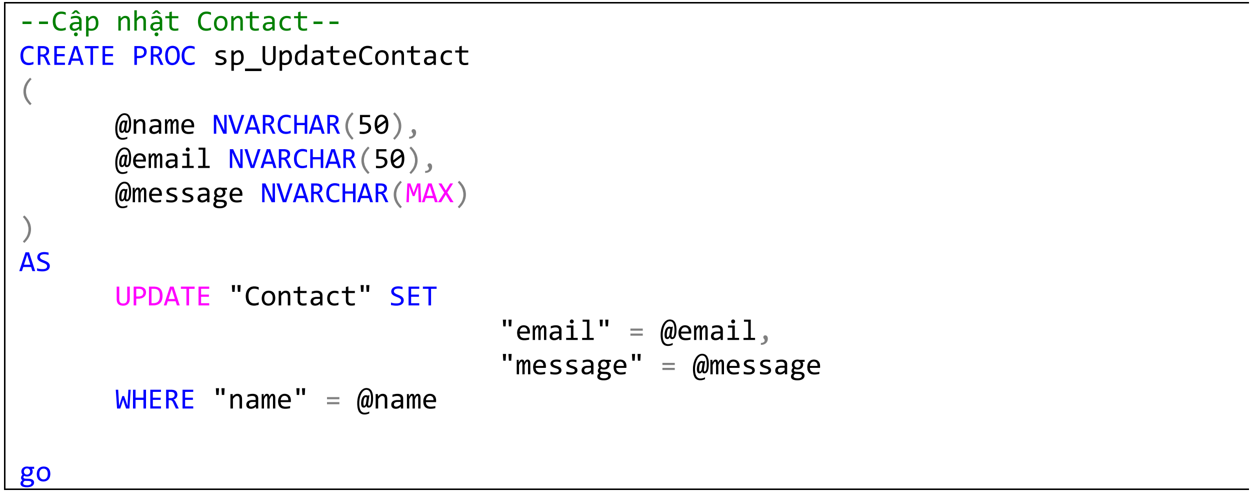


```
--Thêm Contact--
CREATE PROC sp_InsertContact
(
	@name NVARCHAR(50),
	@email NVARCHAR(50),
	@message NVARCHAR(MAX)
)
AS
	INSERT INTO "Contact"("name","email","message")
	VALUES (@name,@email,@message)
go
```

[SP_INSERTCONTACT]

STORED PROCEDURES

CONTACT



```
--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
```

[SP_UPDATECONTACT]

DETAILORDER

STORED PROCEDURES

[SP_GETALLDETAILORDER]

[SP_GETONEDETAILORDER]

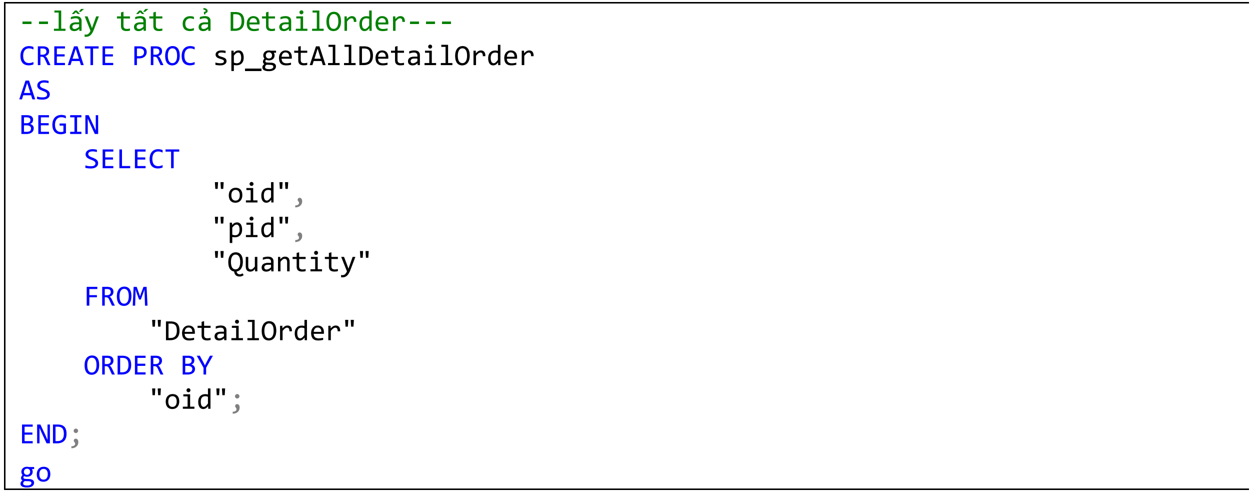
[SP_DELETEDETAILORDER]

[SP_INSERTDETAILORDER]

[SP_UPDATEDETAILORDER]

DETAILORDER

STORED PROCEDURES

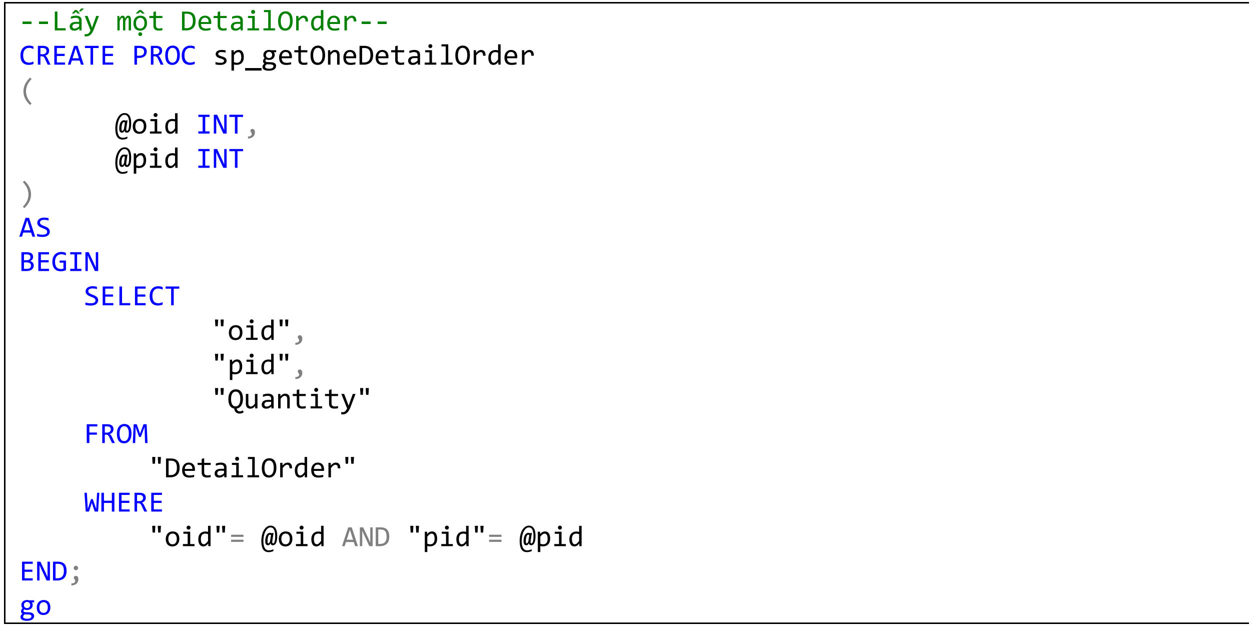


```
--lấy tất cả DetailOrder---
CREATE PROC sp_getAllDetailOrder
AS
BEGIN
    SELECT
            "oid",
            "pid",
            "Quantity"
    FROM
        "DetailOrder"
    ORDER BY
        "oid";
END;
go
```

[SP_GETALLDETAILORDER]

DETAILORDER

STORED PROCEDURES

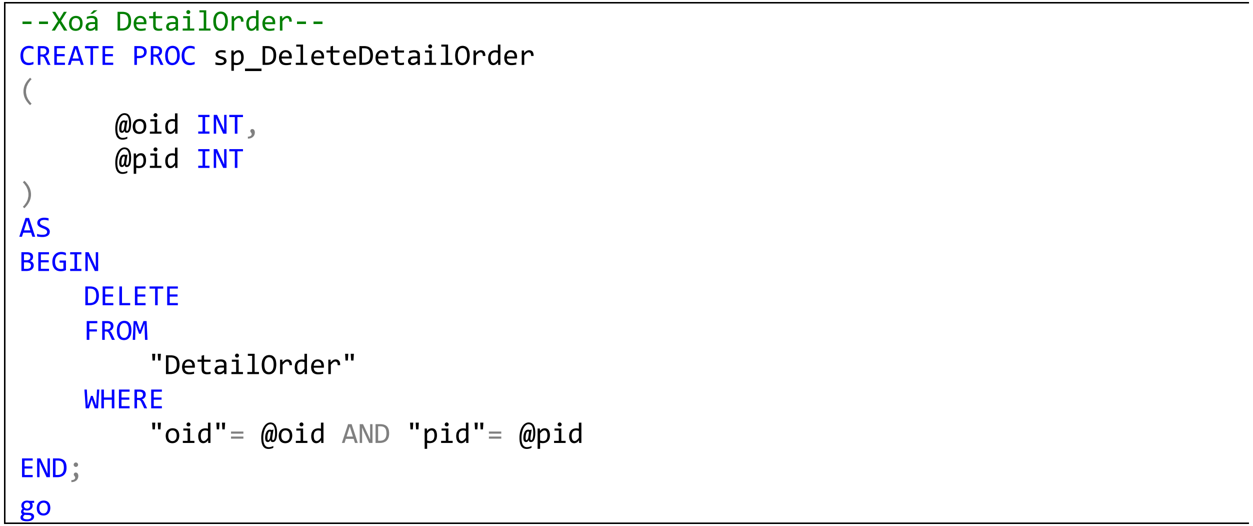


```
--Lấy một DetailOrder--
CREATE PROC sp_getOneDetailOrder
(
      @oid INT,
      @pid INT
)
AS
BEGIN
    SELECT
            "oid",
            "pid",
            "Quantity"
    FROM
        "DetailOrder"
    WHERE
        "oid"= @oid AND "pid"= @pid
END;
go
```

[SP_GETONEDETAILORDER]

DETAILORDER

STORED PROCEDURES

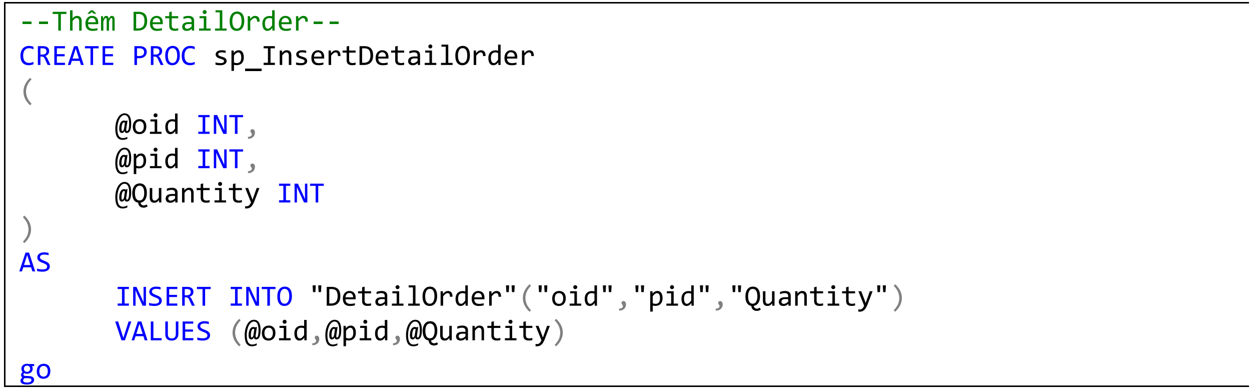


```
--Xoá DetailOrder--
CREATE PROC sp_DeleteDetailOrder
(
      @oid INT,
      @pid INT
)
AS
BEGIN
    DELETE
    FROM
        "DetailOrder"
    WHERE
        "oid"= @oid AND "pid"= @pid
END;
go
```

[SP_DELETEDETAILORDER]

DETAILORDER

STORED PROCEDURES

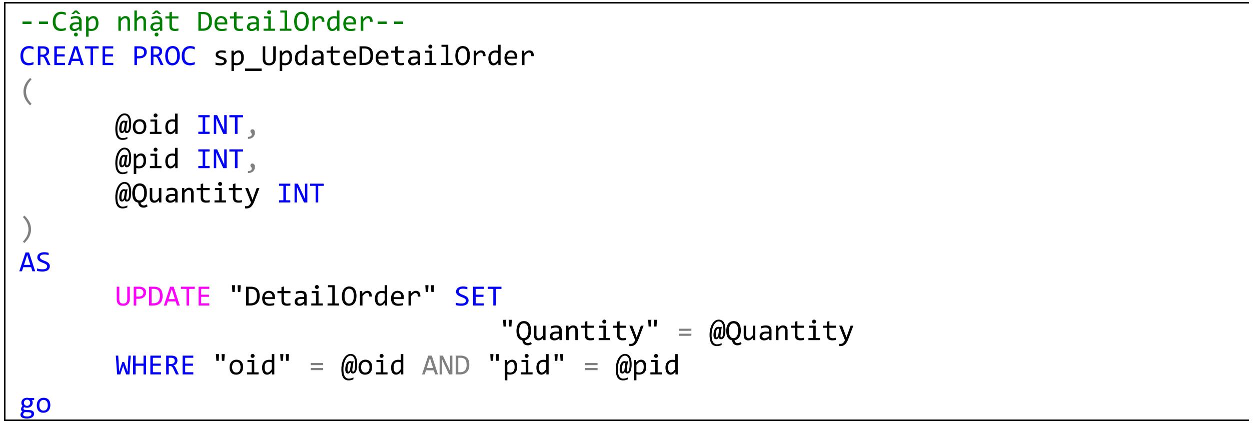


```
--Thêm DetailOrder--
CREATE PROC sp_InsertDetailOrder
(
	@oid INT,
	@pid INT,
	@Quantity INT
)
AS
	INSERT INTO "DetailOrder"("oid","pid","Quantity")
	VALUES (@oid,@pid,@Quantity)
go
```

[SP_INSERTDETAILORDER]

DETAILORDER

STORED PROCEDURES



```
--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
```

[SP_UPDATEDETAILORDER]

STORED PROCEDURES

MEMBER

[SP_GETALLMEMBER]

[SP_GETONEMEMBER]

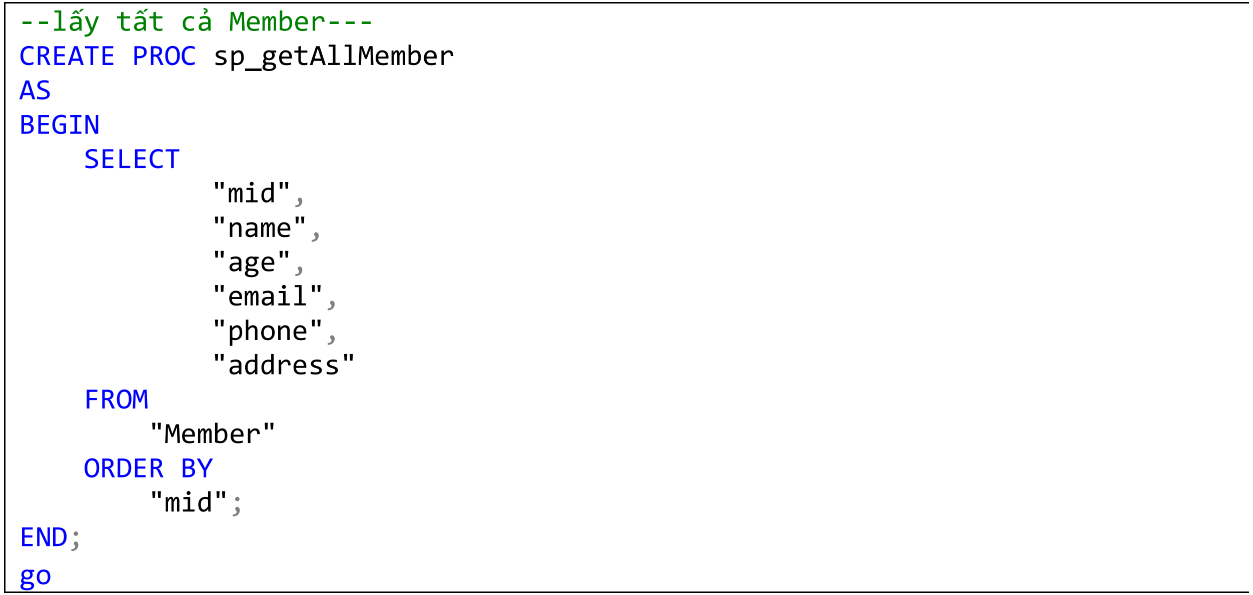
[SP_DELETEMEMBER]

[SP_INSERTMEMBER]

[SP_UPDATEMEMBER]

STORED PROCEDURES

MEMBER



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

[SP_GETALLMEMBER]

MEMBER

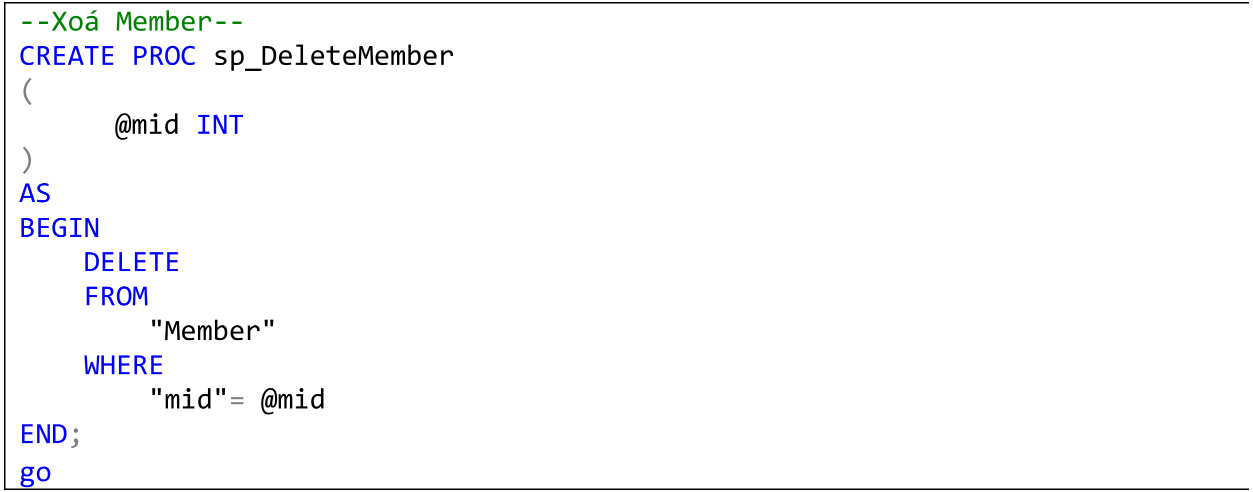
STORED PROCEDURES

```
--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
```

[SP_GETONEMEMBER]

STORED PROCEDURES

MEMBER

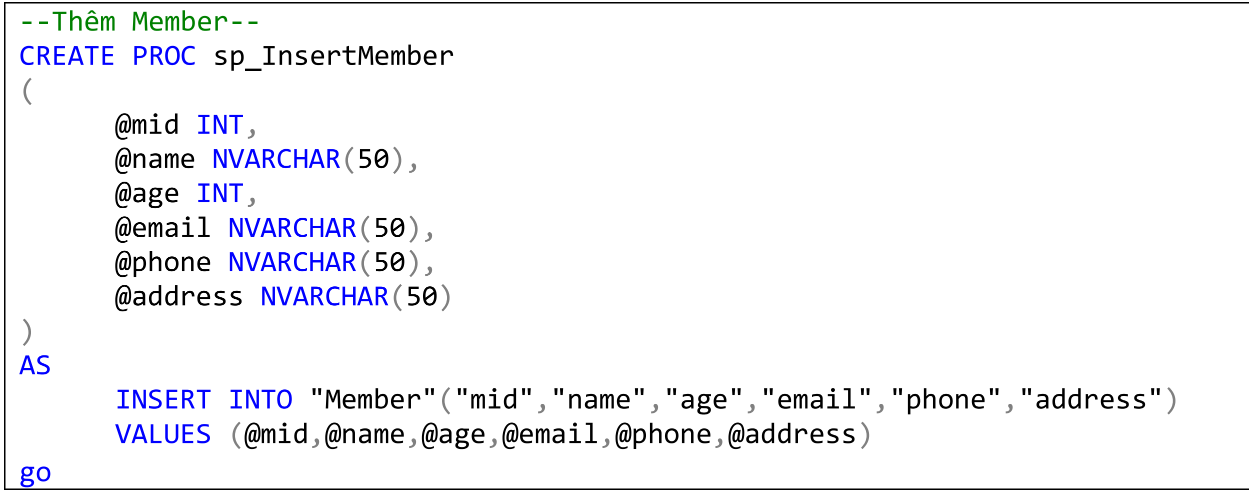


```
--Xoá Member--
CREATE PROC sp_DeleteMember
(
	@mid INT
)
AS
BEGIN
    DELETE
    FROM
        "Member"
    WHERE
        "mid"= @mid
END;
go
```

[SP_DELETEMEMBER]

STORED PROCEDURES

MEMBER

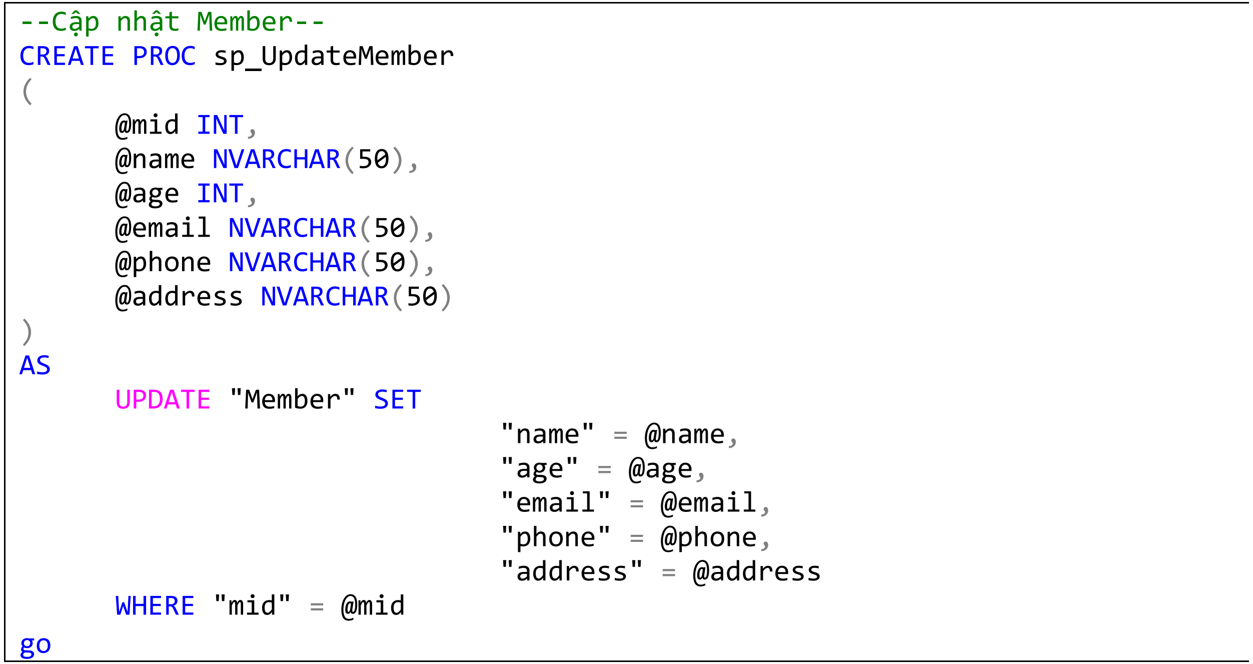


```
--Thêm Member--
CREATE PROC sp_InsertMember
(
	@mid INT,
	@name NVARCHAR(50),
	@age INT,
	@email NVARCHAR(50),
	@phone NVARCHAR(50),
	@address NVARCHAR(50)
)
AS
	INSERT INTO "Member"("mid","name","age","email","phone","address")
	VALUES (@mid,@name,@age,@email,@phone,@address)
go
```

[SP_INSERTMEMBER]

STORED PROCEDURES

MEMBER



```sql
--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
```

[SP_UPDATEMEMBER]

STORED PROCEDURES

ORDER

[SP_GETALLORDER]

[SP_GETONEORDER]

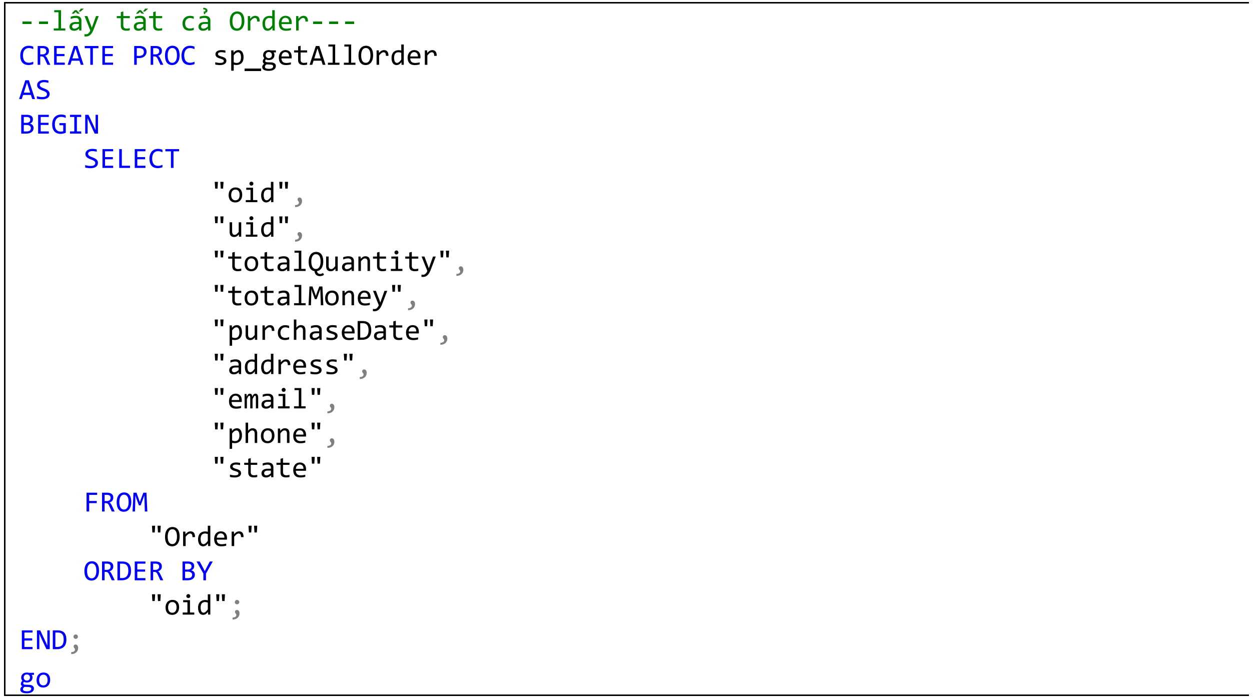
[SP_DELETEORDER]

[SP_INSERTORDER]

[SP_UPDATEORDER]

STORED PROCEDURES

ORDER

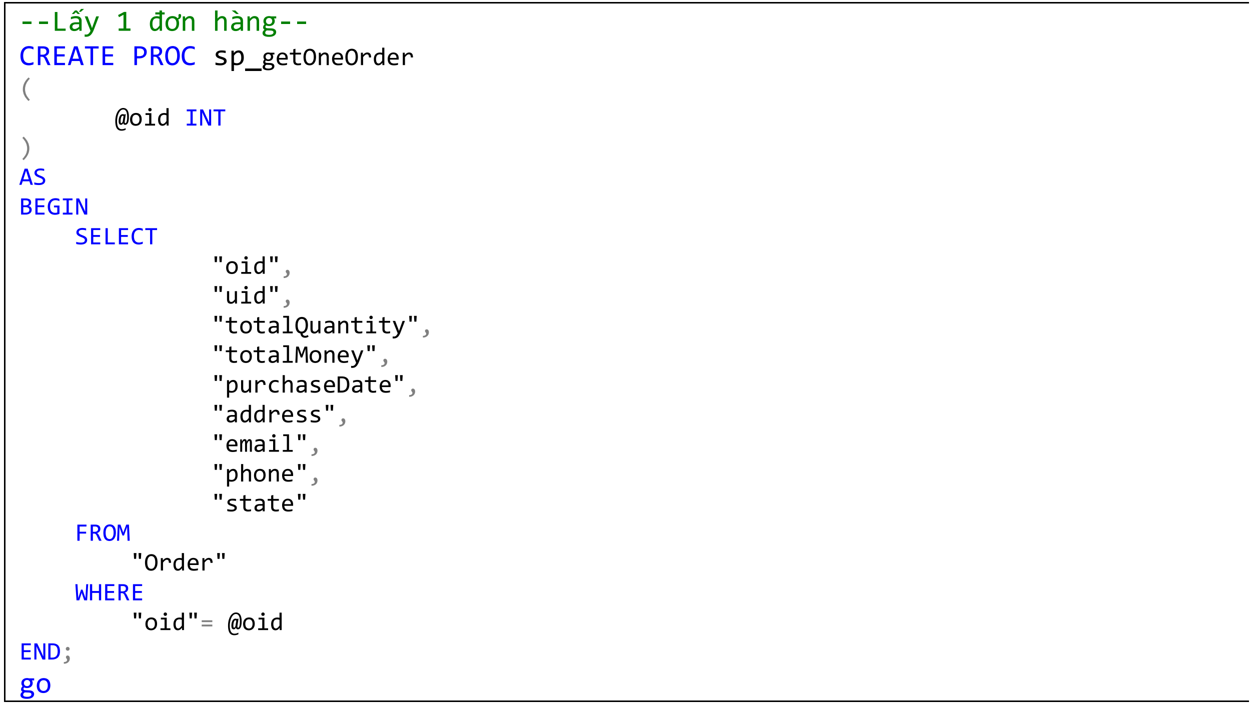


```
--lấy tất cả Order---
CREATE PROC sp_getAllOrder
AS
BEGIN
    SELECT
            "oid",
            "uid",
            "totalQuantity",
            "totalMoney",
            "purchaseDate",
            "address",
            "email",
            "phone",
            "state"
    FROM
        "Order"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

[SP_GETALLORDER]

STORED PROCEDURES

ORDER

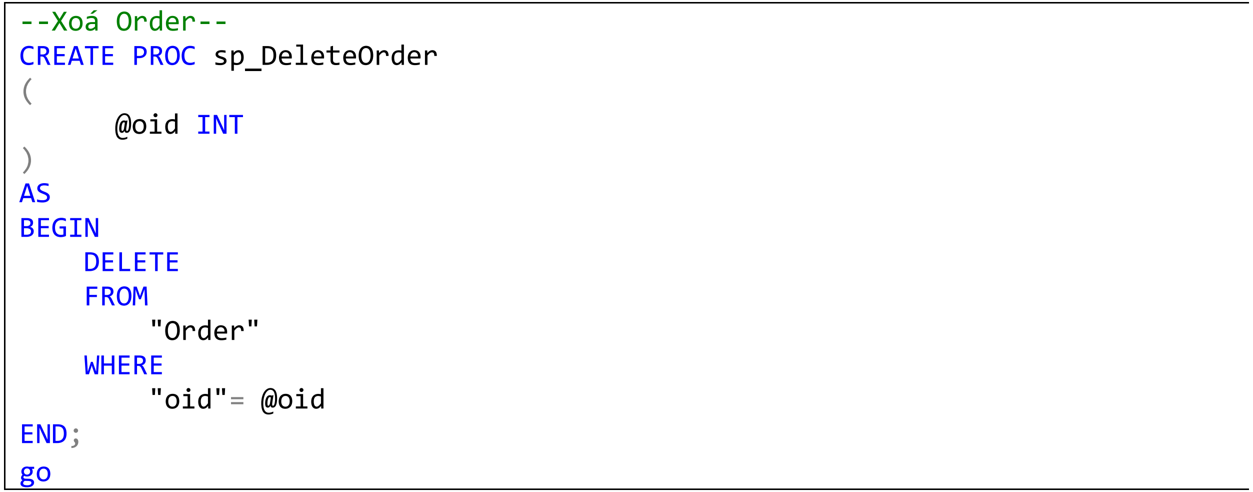


```
--Lấy 1 đơn hàng--
CREATE PROC sp_getOneOrder
(
	@oid INT
)
AS
BEGIN
    SELECT
              "oid",
              "uid",
              "totalQuantity",
              "totalMoney",
              "purchaseDate",
              "address",
              "email",
              "phone",
              "state"
    FROM
        "Order"
    WHERE
        "oid"= @oid
END;
go
```

[SP_GETONEORDER]

ORDER

STORED PROCEDURES

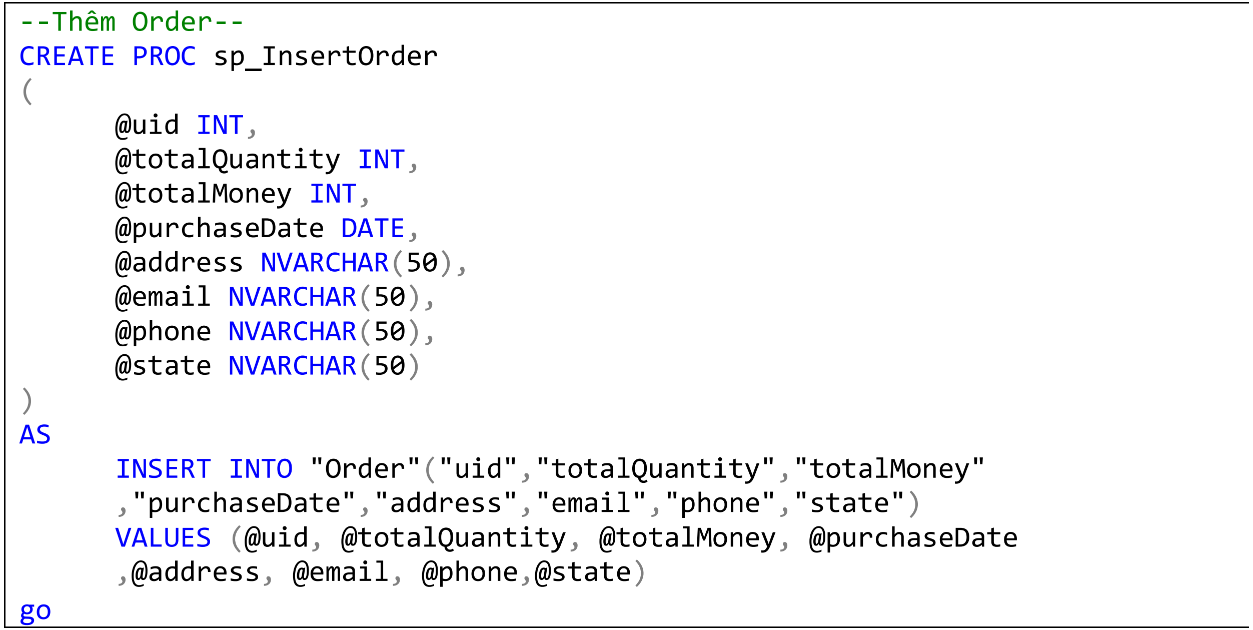


```
--Xoá Order--
CREATE PROC sp_DeleteOrder
(
      @oid INT
)
AS
BEGIN
    DELETE
    FROM
        "Order"
    WHERE
        "oid"= @oid
END;
go
```

[SP_DELETEORDER]

ORDER

STORED PROCEDURES

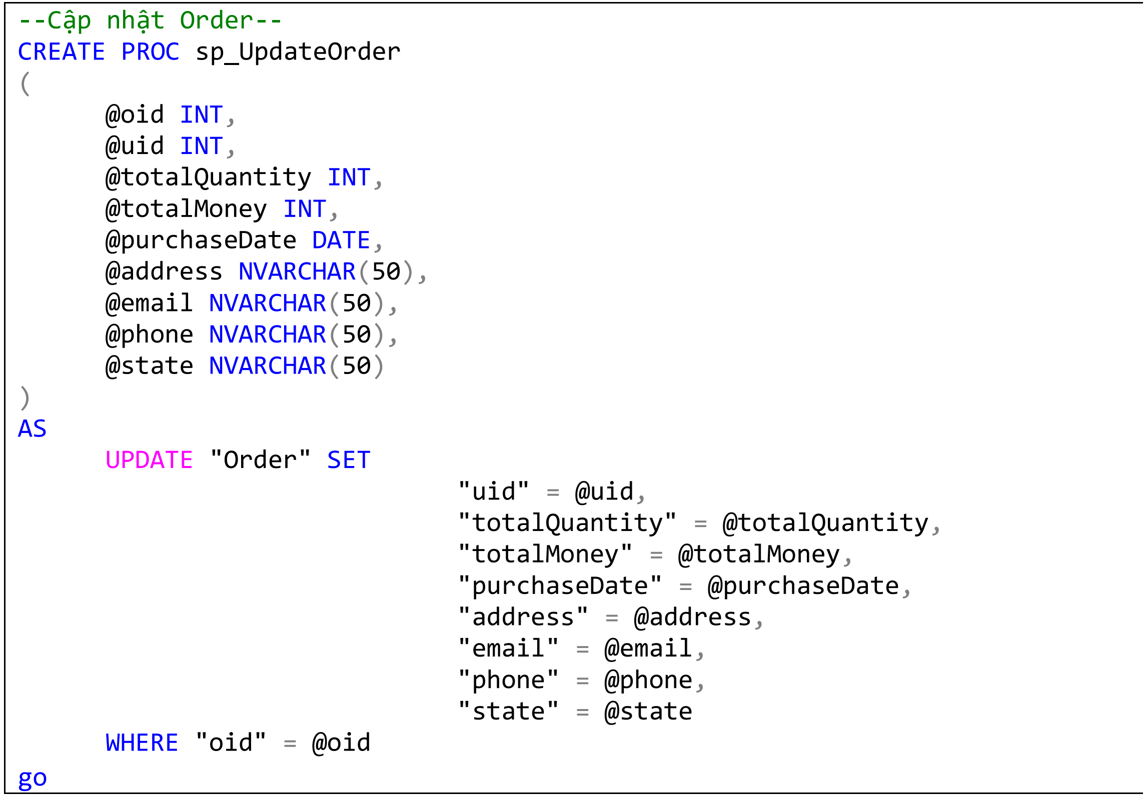


```sql
--Thêm Order--
CREATE PROC sp_InsertOrder
(
	@uid INT,
	@totalQuantity INT,
	@totalMoney INT,
	@purchaseDate DATE,
	@address NVARCHAR(50),
	@email NVARCHAR(50),
	@phone NVARCHAR(50),
	@state NVARCHAR(50)
)
AS
	INSERT INTO "Order"("uid","totalQuantity","totalMoney"
	,"purchaseDate","address","email","phone","state")
	VALUES (@uid, @totalQuantity, @totalMoney, @purchaseDate
	,@address, @email, @phone,@state)
go
```

[SP_INSERTORDER]

ORDER

STORED PROCEDURES



```
--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
```

[SP_UPDATEORDER]

STORED PROCEDURES

PRODUCT

[SP_GETALLPRODUCT]

[SP_GETONEPRODUCT]

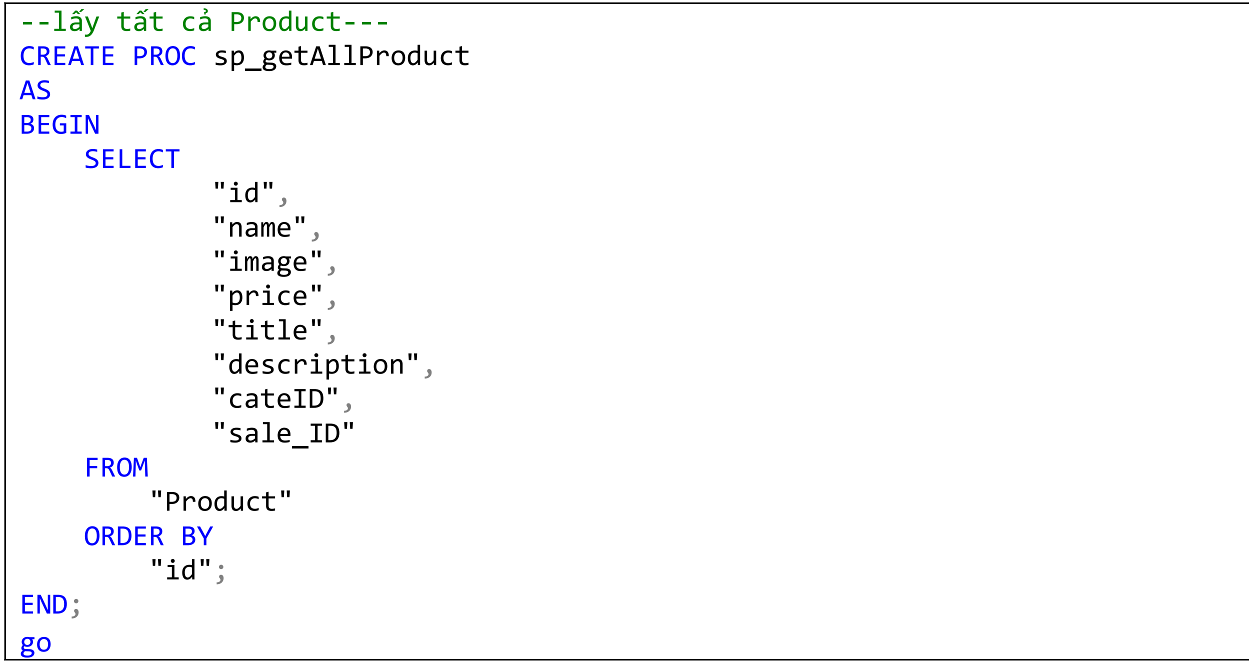
[SP_DELETEPRODUCT]

[SP_INSERTPRODUCT]

[SP_UPDATEPRODUCT]

PRODUCT

STORED
PROCEDURES

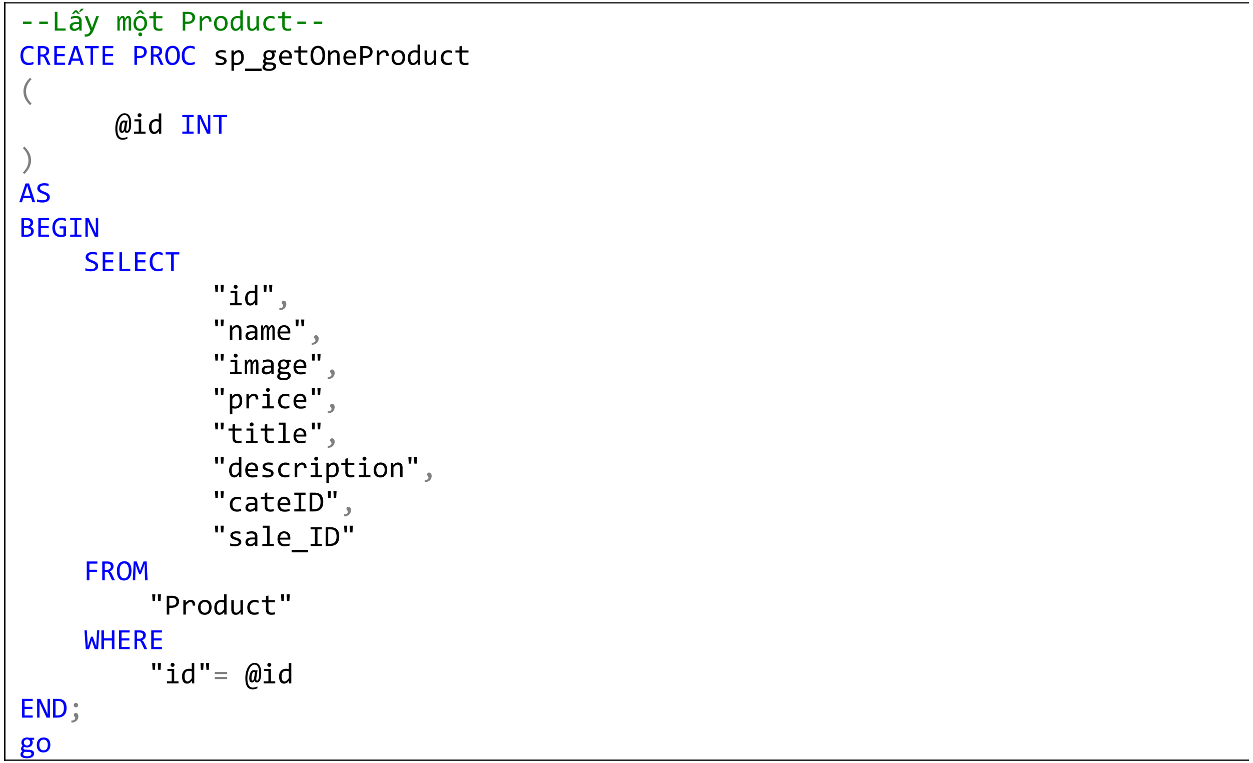


```
--lấy tất cả Product---
CREATE PROC sp_getAllProduct
AS
BEGIN
    SELECT
            "id",
            "name",
            "image",
            "price",
            "title",
            "description",
            "cateID",
            "sale_ID"
    FROM
        "Product"
    ORDER BY
        "id";
END;
go
```

[SP_GETALLPRODUCT]

STORED PROCEDURES

PRODUCT

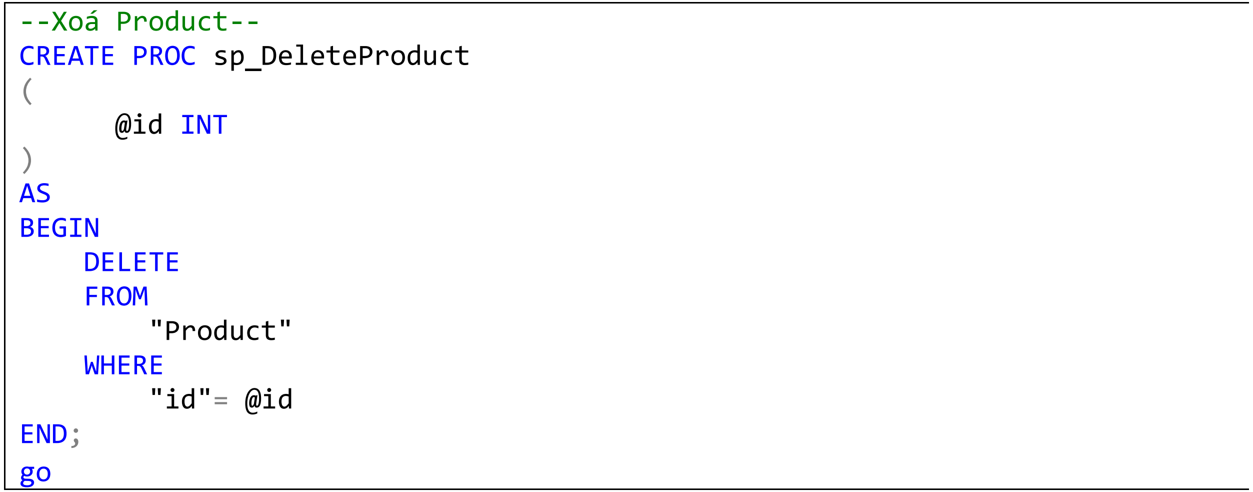


```
--Lấy một Product--
CREATE PROC sp_getOneProduct
(
	@id INT
)
AS
BEGIN
    SELECT
            "id",
            "name",
            "image",
            "price",
            "title",
            "description",
            "cateID",
            "sale_ID"
    FROM
        "Product"
    WHERE
        "id"= @id
END;
go
```

[SP_GETONEPRODUCT]

PRODUCT

STORED PROCEDURES

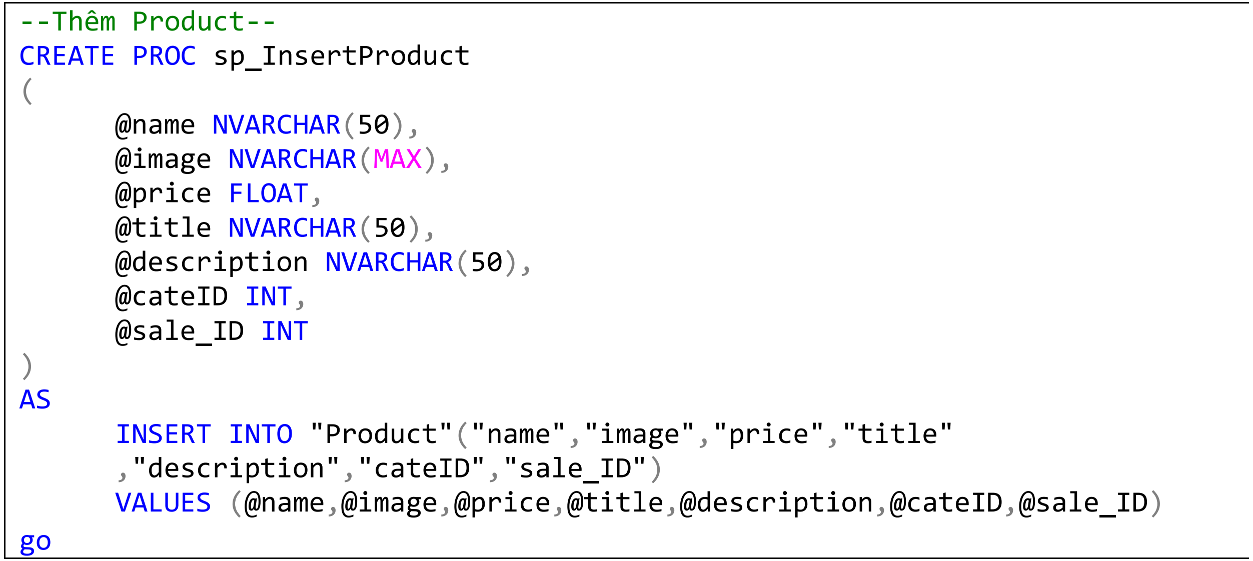


```
--Xoá Product--
CREATE PROC sp_DeleteProduct
(
        @id INT
)
AS
BEGIN
    DELETE
    FROM
        "Product"
    WHERE
        "id"= @id
END;
go
```

[SP_DELETEPRODUCT]

STORED PROCEDURES

PRODUCT

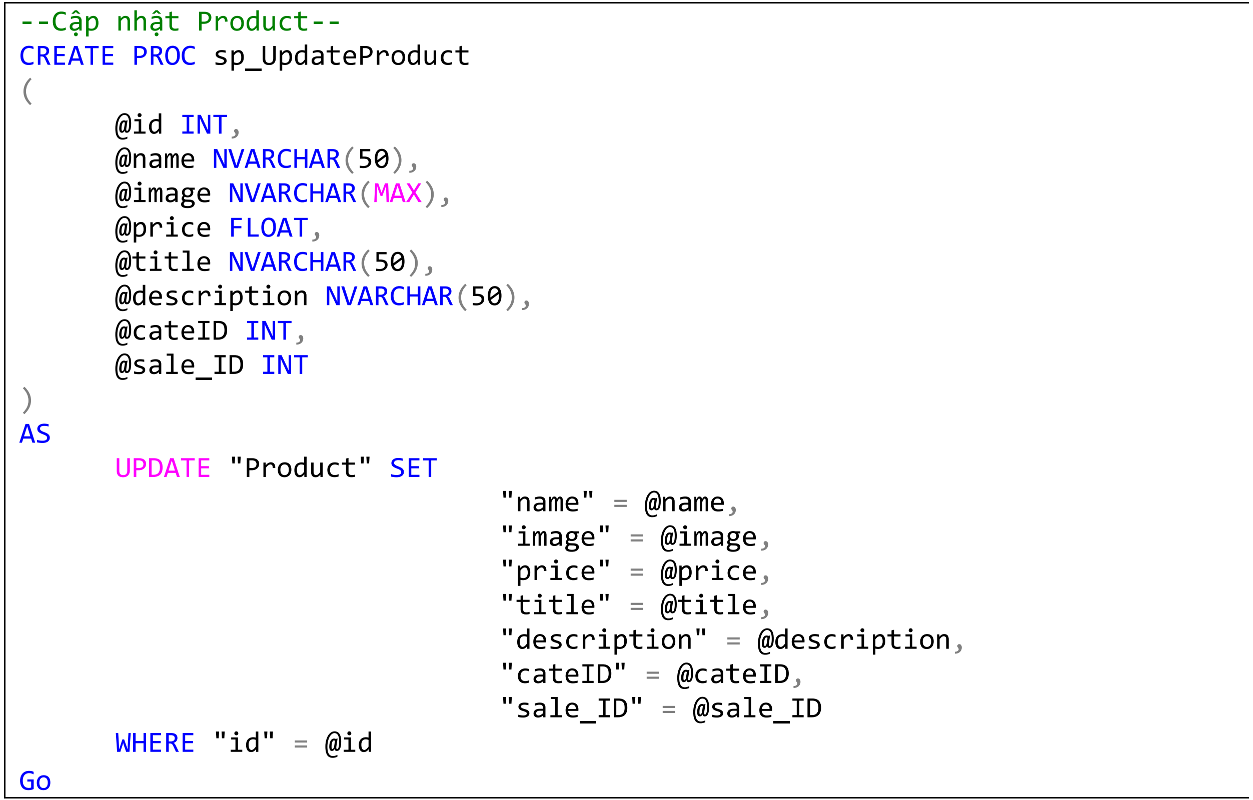


```sql
--Thêm Product--
CREATE PROC sp_InsertProduct
(
        @name NVARCHAR(50),
        @image NVARCHAR(MAX),
        @price FLOAT,
        @title NVARCHAR(50),
        @description NVARCHAR(50),
        @cateID INT,
        @sale_ID INT
)
AS
        INSERT INTO "Product"("name","image","price","title"
        ,"description","cateID","sale_ID")
        VALUES (@name,@image,@price,@title,@description,@cateID,@sale_ID)
go
```

[SP_INSERTPRODUCT]

STORED PROCEDURES

PRODUCT



```
--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
```

[SP_UPDATEPRODUCT]

STORED PROCEDURES

KHÁC

[SP_GETALLPRODUCTBYSALEID]

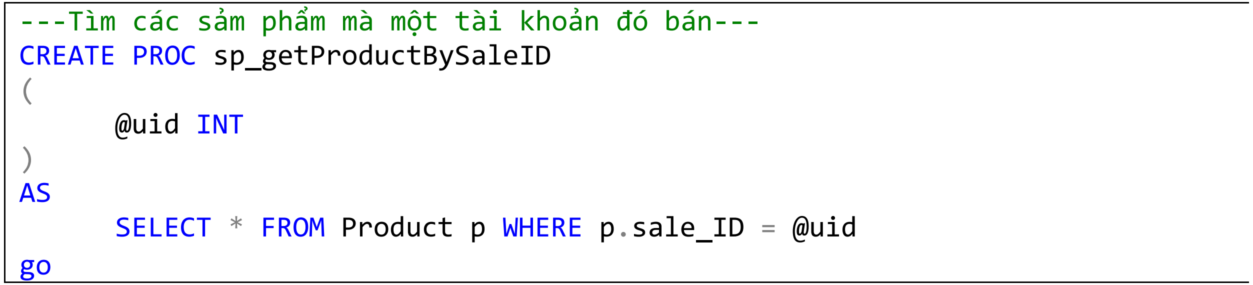
[SP_INSERTORDERWITHDETAIL]

[SP_GETLASTPRODUCT]

[SP_CREATE_LOGIN_USER]

KHÁC

STORED PROCEDURES

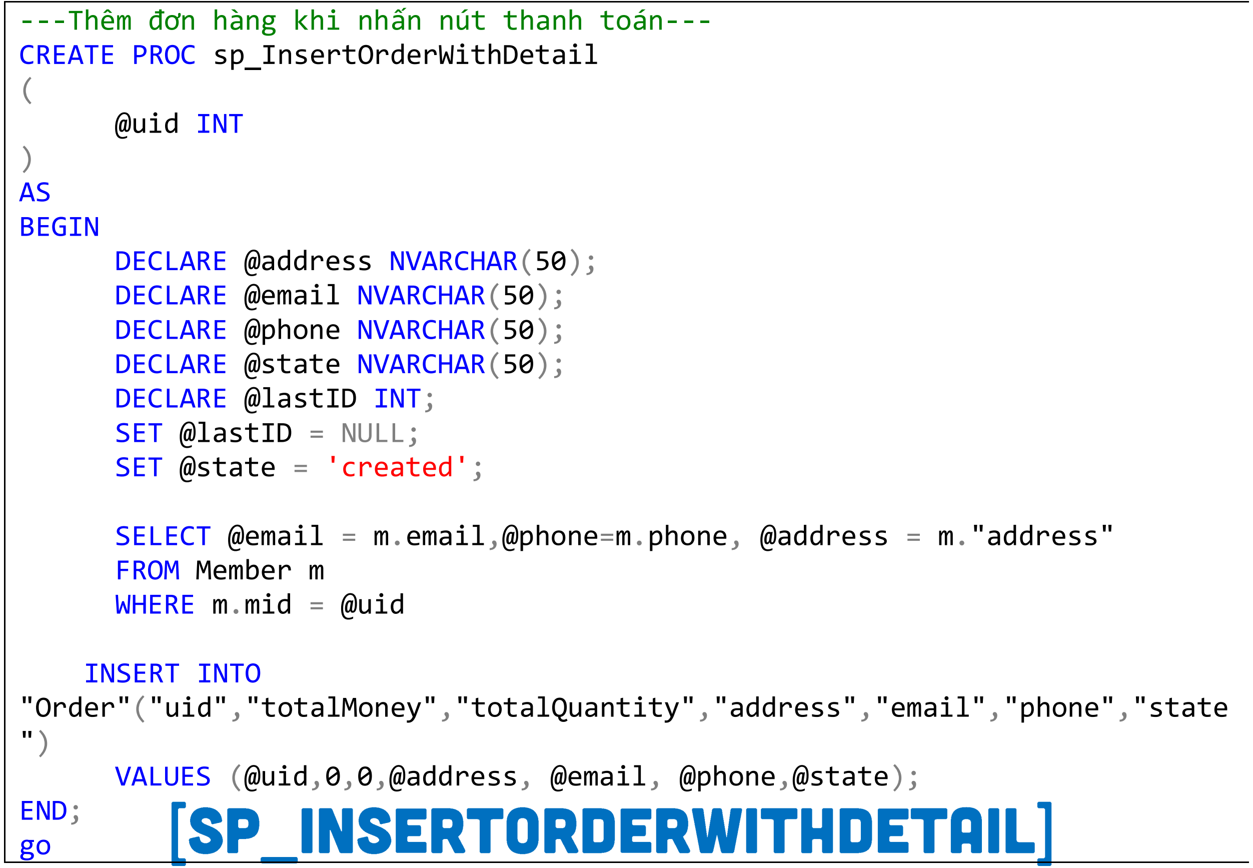


```
---Tìm các sảm phẩm mà một tài khoản đó bán---
CREATE PROC sp_getProductBySaleID
(
      @uid INT
)
AS
      SELECT * FROM Product p WHERE p.sale_ID = @uid
go
```

[SP_GETALLPRODUCTBYSALEID]

STORED PROCEDURES

KHÁC

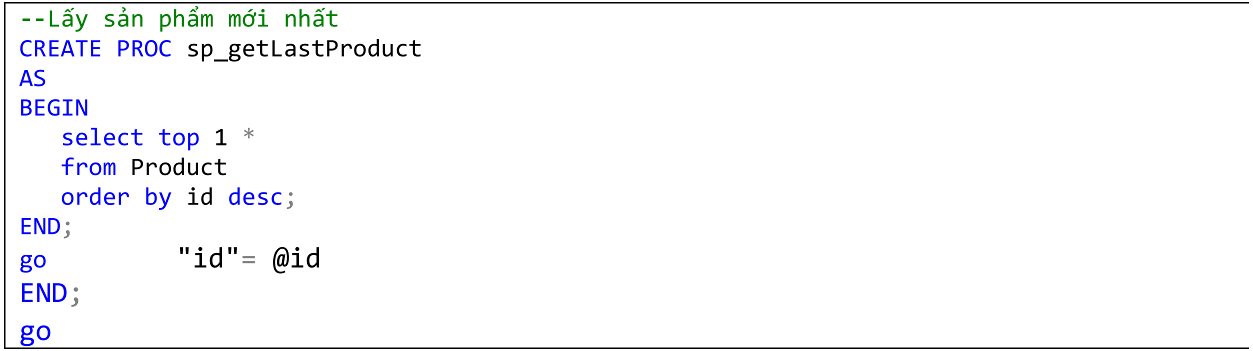


```
---Thêm đơn hàng khi nhấn nút thanh toán---
CREATE PROC sp_InsertOrderWithDetail
(
      @uid INT
)
AS
BEGIN
      DECLARE @address NVARCHAR(50);
      DECLARE @email NVARCHAR(50);
      DECLARE @phone NVARCHAR(50);
      DECLARE @state NVARCHAR(50);
      DECLARE @lastID INT;
      SET @lastID = 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
```

[SP_INSERTORDERWITHDETAIL]

KHÁC

STORED PROCEDURES

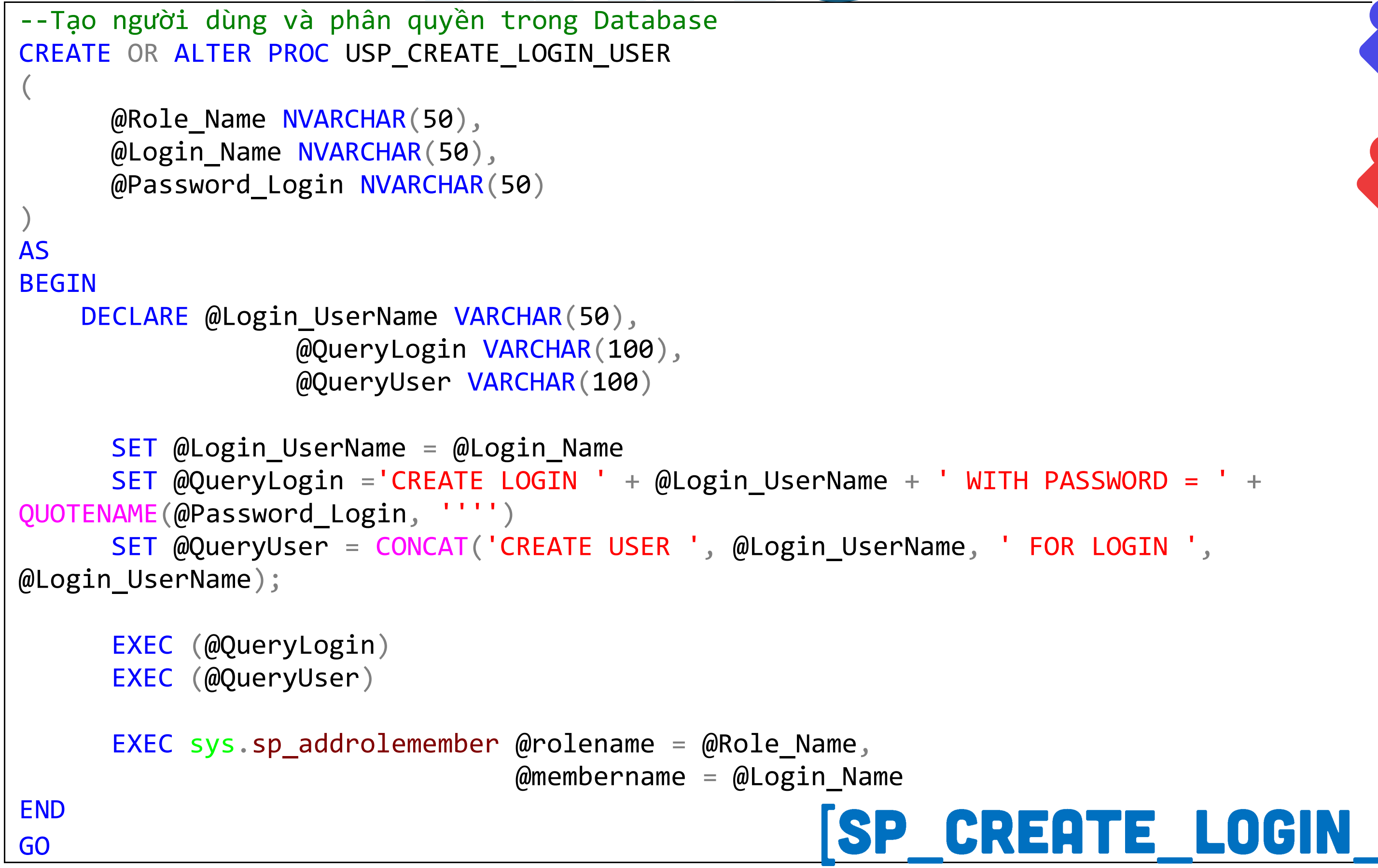


```
--Lấy sản phẩm mới nhất
CREATE PROC sp_getLastProduct
AS
BEGIN
   select top 1 *
   from Product
   order by id desc;
END;
go         "id"= @id
END;
go
```

[SP_GETLASTPRODUCT]

STORED PROCEDURES

KHÁC



```
--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
```

[SP_CREATE_LOGIN_USER]

FUNCTIONS

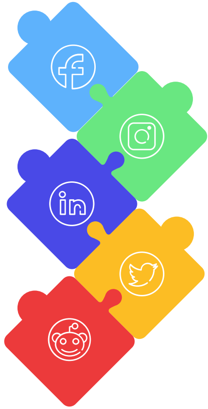


Table-valued functions

[fn_SearchProductName]

[fn_SearchMaxPriceByCategory

[fn_ProductInRangePrice]

[fn_ProductByCateName]

[fn_ProductBySalerName]

[fn_ProductAndAmount]

[fn_HoaDon]

Scalar-valued functions

[fn_LastOrderID]

FUNCTIONS

[fn_SearchProductName]

```
---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
```

[fn_SearchMaxPriceByCategory

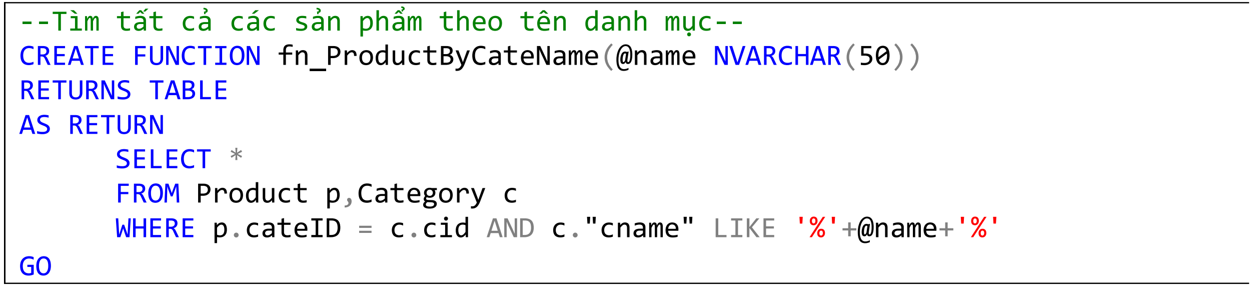
```
--Tìm giá của sản phẩm cao nhất trong từng danh mục--
CREATE FUNCTION fn_SearchMaxPriceByCategory()
RETURNS TABLE
AS RETURN
      SELECT c.cname, MAX(p.price) AS maxPrice
      FROM Product p, Category c
      WHERE p.cateID= c.cid
      GROUP BY c.cname
GO
```

[fn_ProductInRangePrice]

```
--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
```

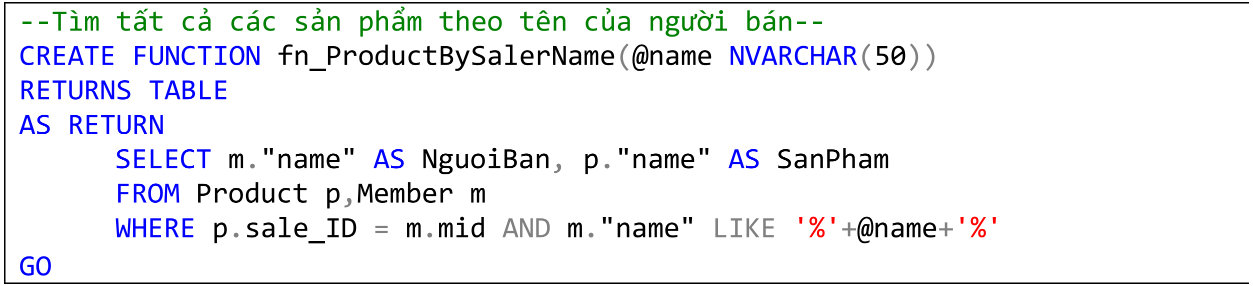
FUNCTIONS

[fn_ProductByCateName]



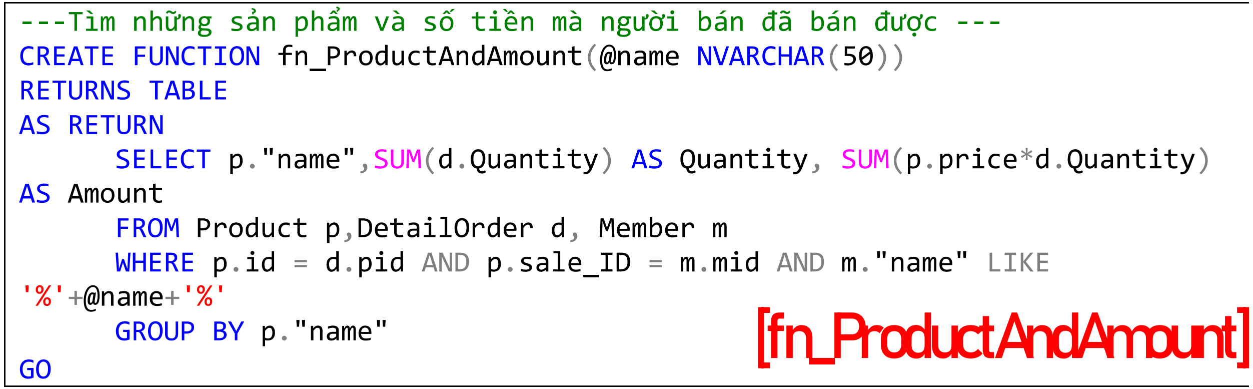
```sql
--Tìm tất cả các sản phẩm theo tên danh mục--
CREATE FUNCTION fn_ProductByCateName(@name NVARCHAR(50))
RETURNS TABLE
AS RETURN
      SELECT *
      FROM Product p,Category c
      WHERE p.cateID = c.cid AND c."cname" LIKE '%'+@name+'%'
GO
```

[fn_ProductBySalerName]



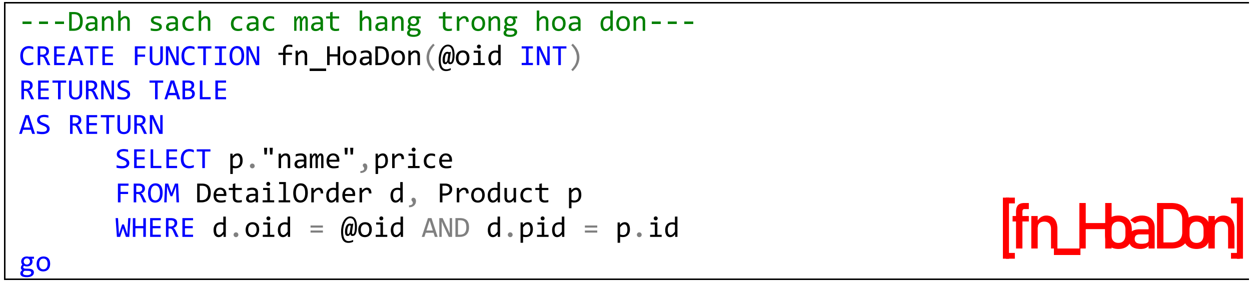
```sql
--Tìm tất cả các sản phẩm theo tên của người bán--
CREATE FUNCTION fn_ProductBySalerName(@name NVARCHAR(50))
RETURNS TABLE
AS RETURN
      SELECT m."name" AS NguoiBan, p."name" AS SanPham
      FROM Product p,Member m
      WHERE p.sale_ID = m.mid AND m."name" LIKE '%'+@name+'%'
GO
```

FUNCTIONS



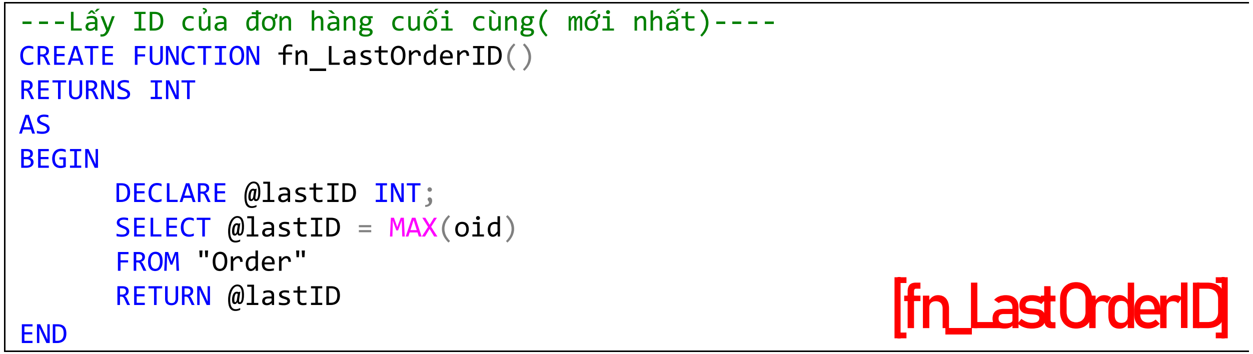
```
---Tìm những sản phẩm và số tiền mà người bán đã bán được ---
CREATE FUNCTION fn_ProductAndAmount(@name NVARCHAR(50))
RETURNS TABLE
AS RETURN
	SELECT p."name",SUM(d.Quantity) AS Quantity, SUM(p.price*d.Quantity)
AS Amount
	FROM Product p,DetailOrder d, Member m
	WHERE p.id = d.pid AND p.sale_ID =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

[fn_ProductAndAmount]



```
---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
```

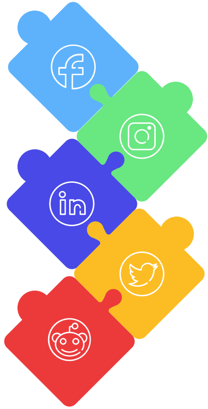
[fn_HoaDon]



```
---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
```

[fn_LastOrderID]

TRIGGERS



[AutoCreateUserDB]

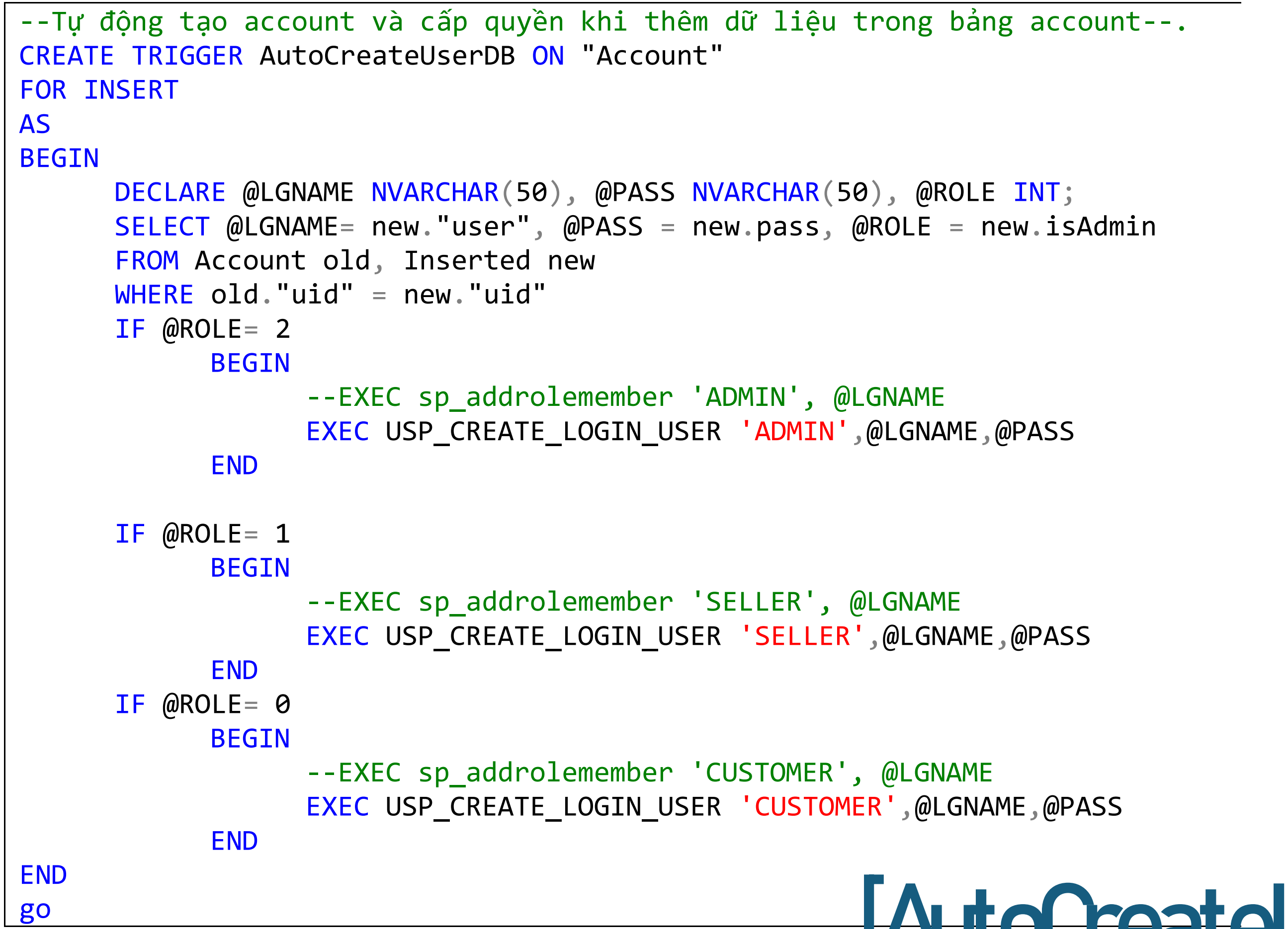
[Prevent_Age_lr_10]

[Discount_for_Cat]

[tg_AutoDate]

[tg_TotalQuantityMoneyOfOrder_inOder]

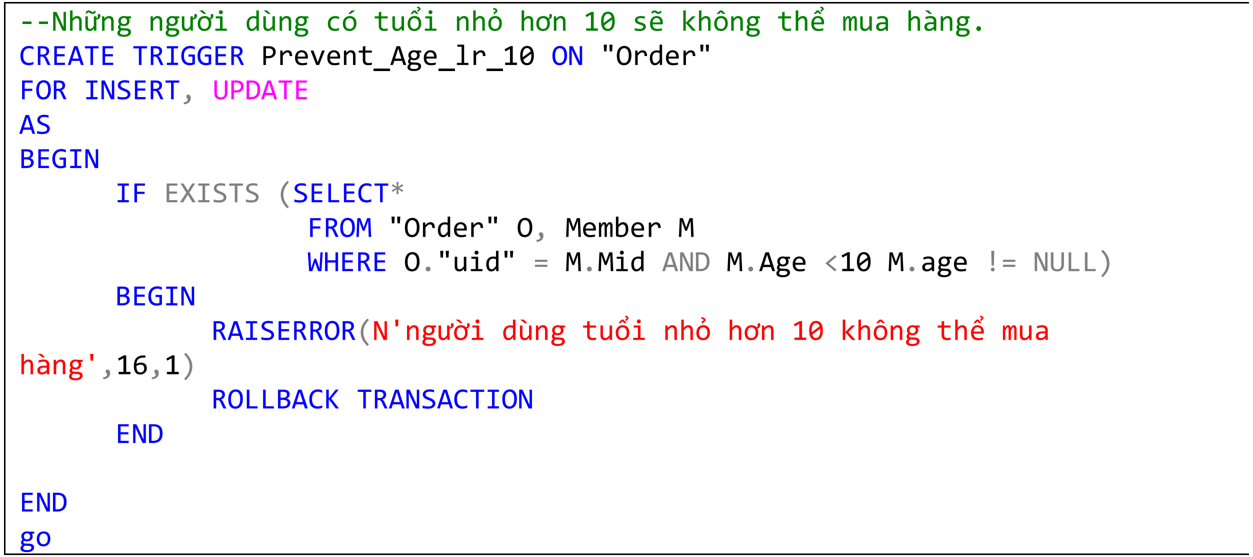
TRIGGERS



```
--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
```

[AutoCreateUserDB]

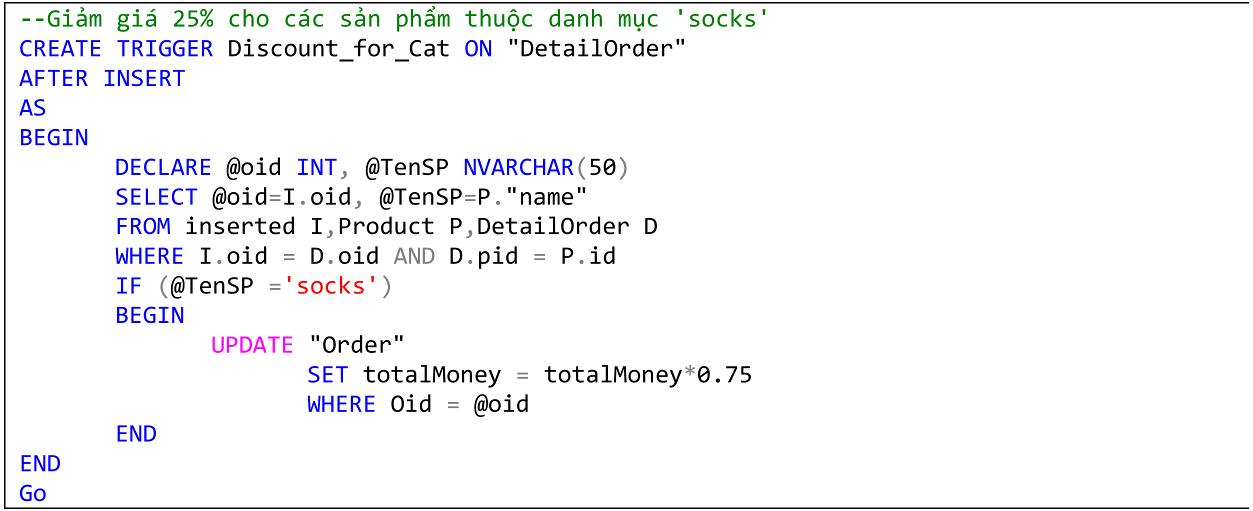
TRIGGERS



```
--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 M.age != NULL)
      BEGIN
            RAISERROR(N'người dùng tuổi nhỏ hơn 10 không thể mua
hàng',16,1)
            ROLLBACK TRANSACTION
      END

END
go
```

[Prevent_Age_lr_10]

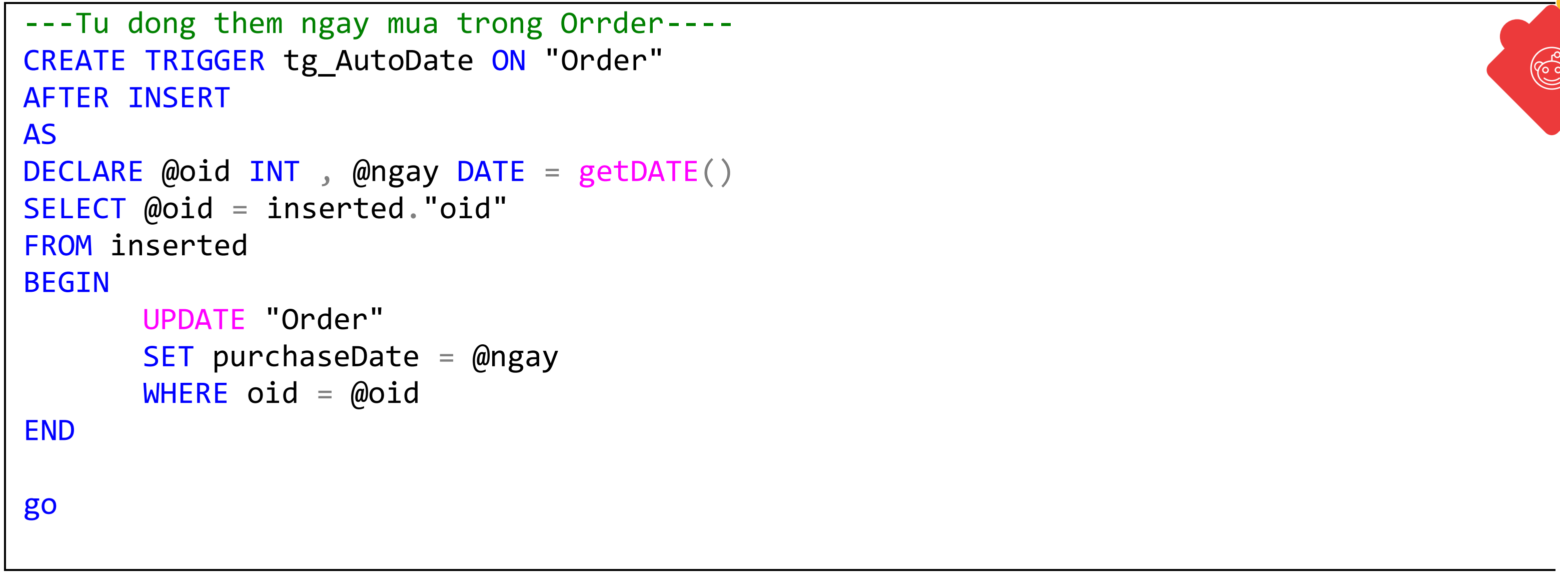
TRIGGERS



```
--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
```

[Discount_for_Cat]

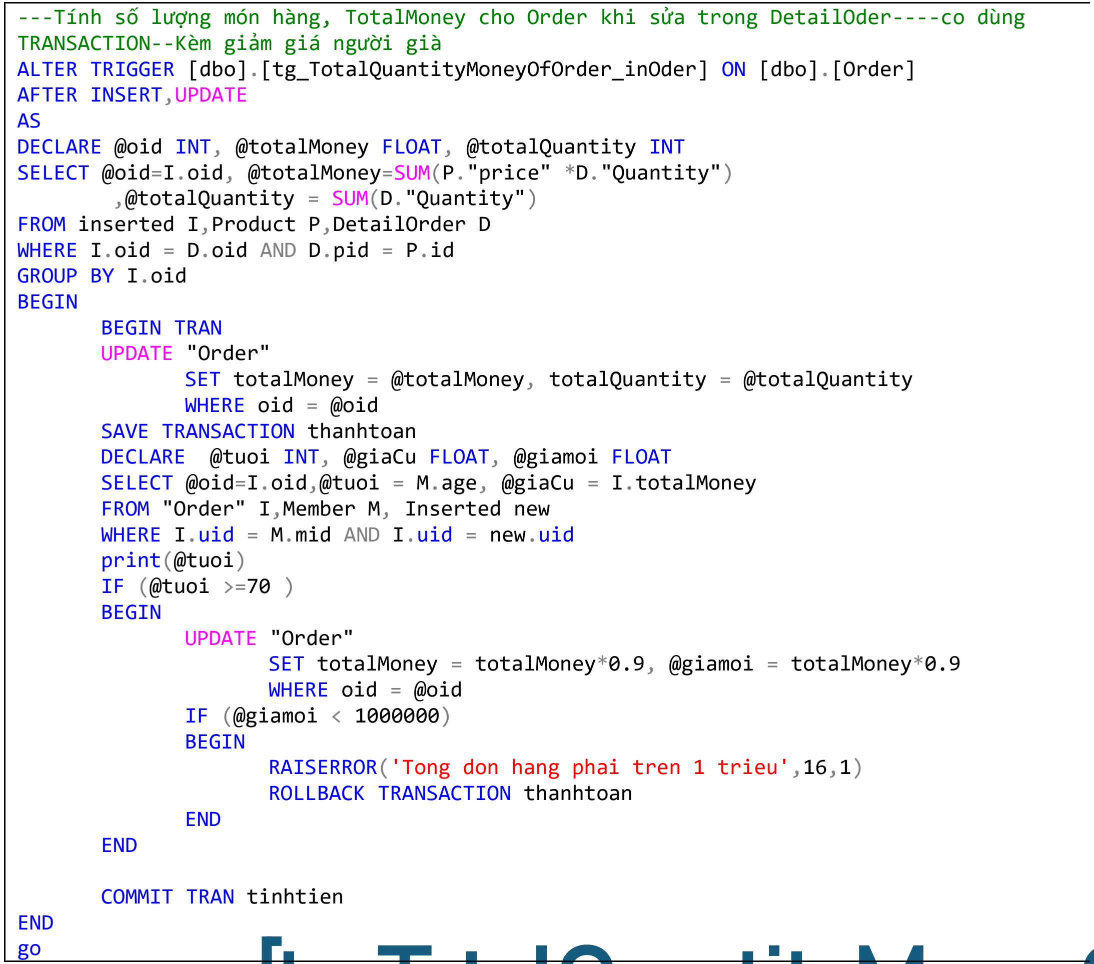
TRIGGERS



```sql
---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
```

[tg_AutoDate]

TRIGGERS



```
---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
```

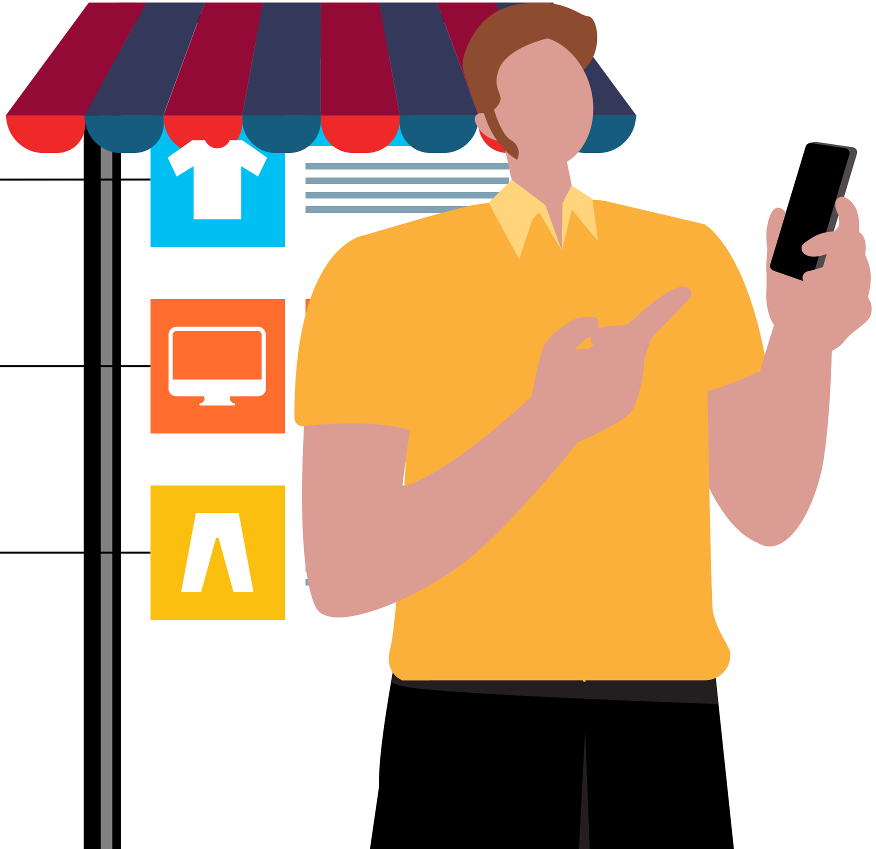
[tg_TotalQuantityMoneyOfOrder_inOder]

PHÂN QUYỀN

ADMIN

SELLER

CUSTOMER



TẠO ROLES

```
CREATE ROLE ADMIN
CREATE ROLE CUSTOMER
CREATE ROLE SELLER
```

ADMIN

SELLER

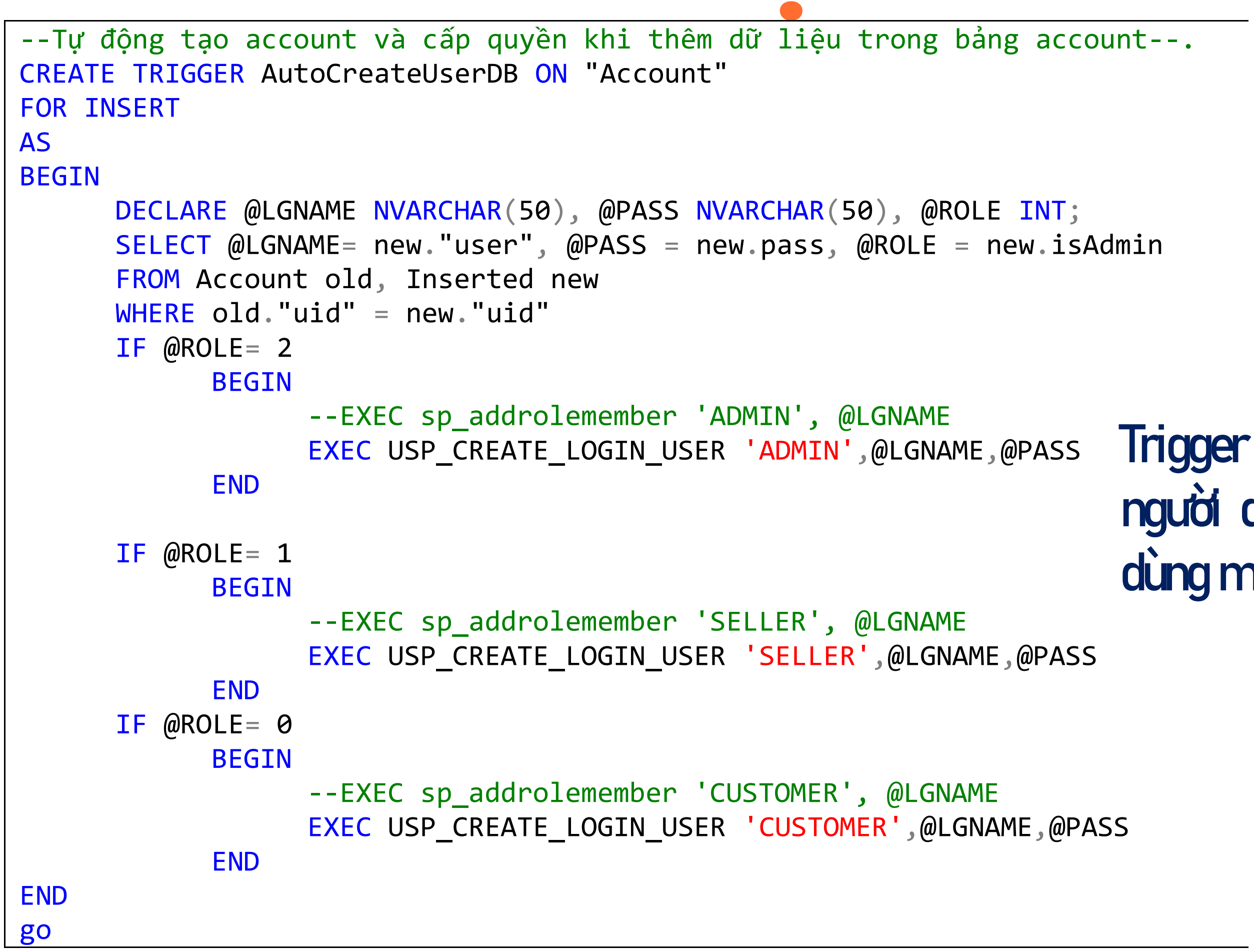
CUSTOMER

TẠO USER

```
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
```

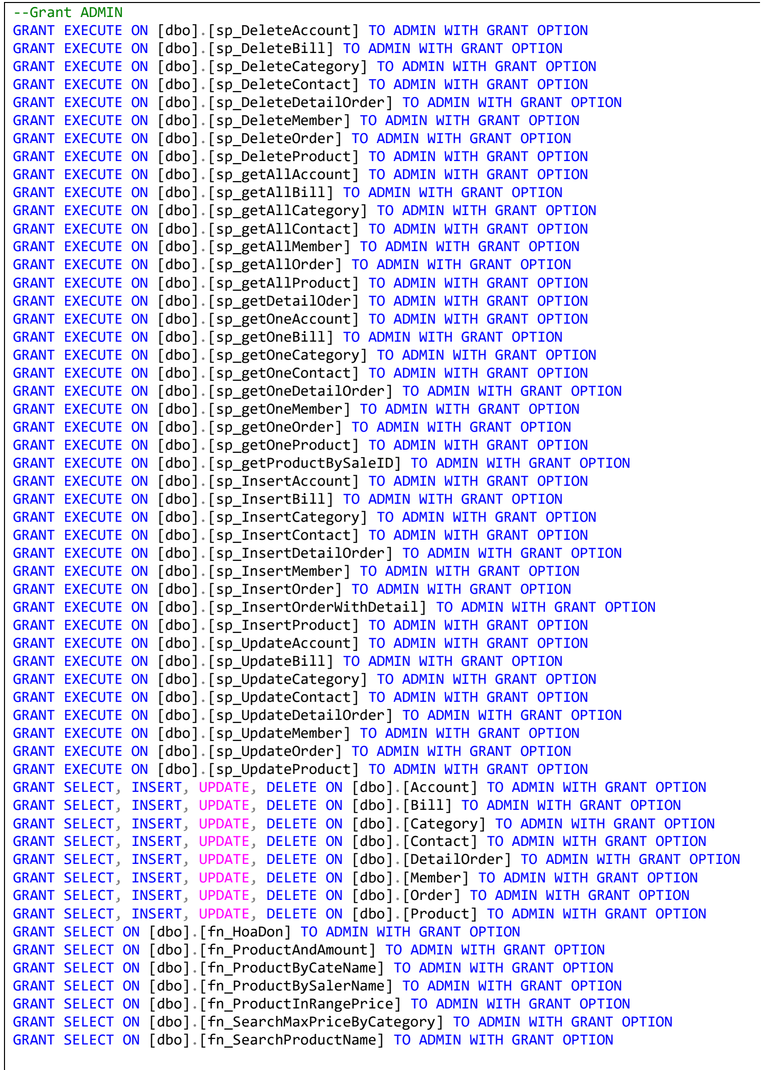
Stored Procedure tạo user và phân quyền

TẠO USER



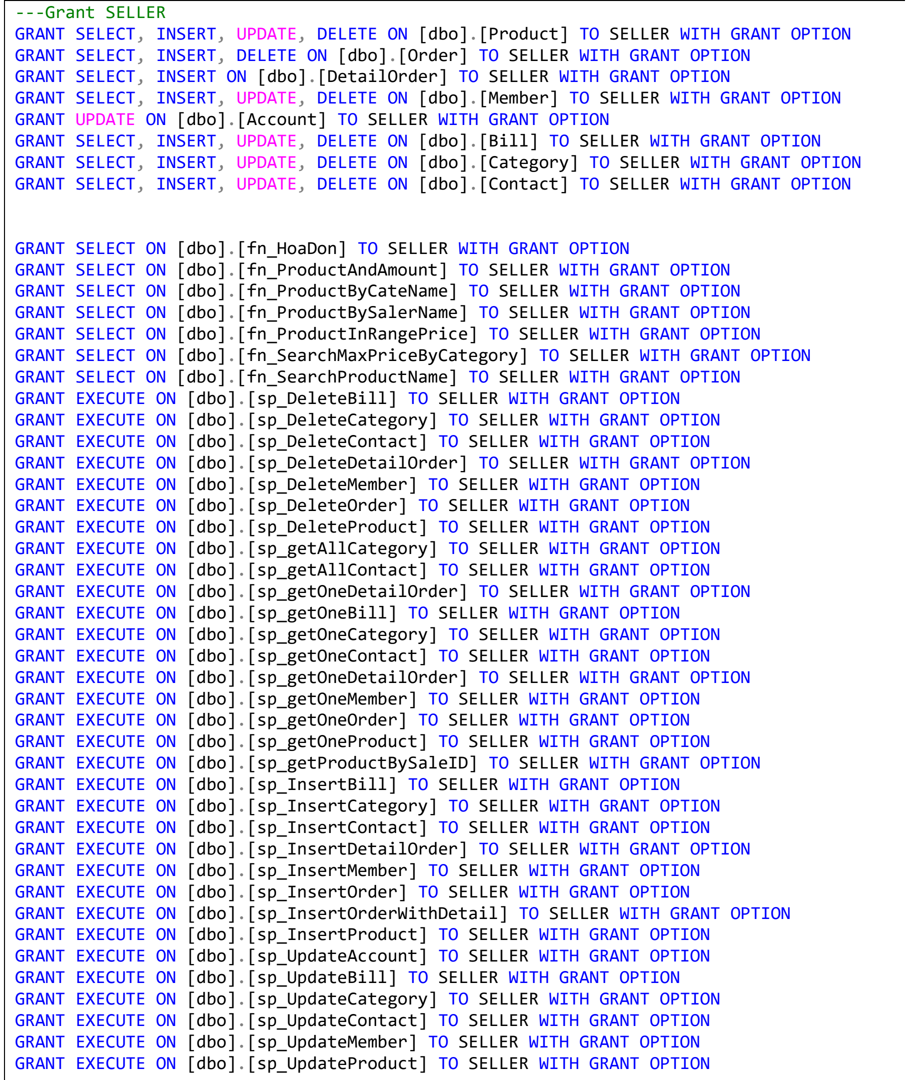
```sql
--Tự động tạo account và cấp quyền khi thêm dữ liệu trong bảng account--.
CREATE TRIGGER AutoCreateUserDB ON "Account"
FOR INSERT
AS
BEGIN
    DECLARE @LGNAME NVARCHAR(50), @PASS NVARCHAR(50), @ROLE INT;
    SELECT @LGNAME= new."user", @PASS = new.pass, @ROLE = new.isAdmin
    FROM Account old, 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
```

Trigger tự động gọi stored procedure tạo người dùng và phân quyền khi thêm một dùng mới trong bảng Account.

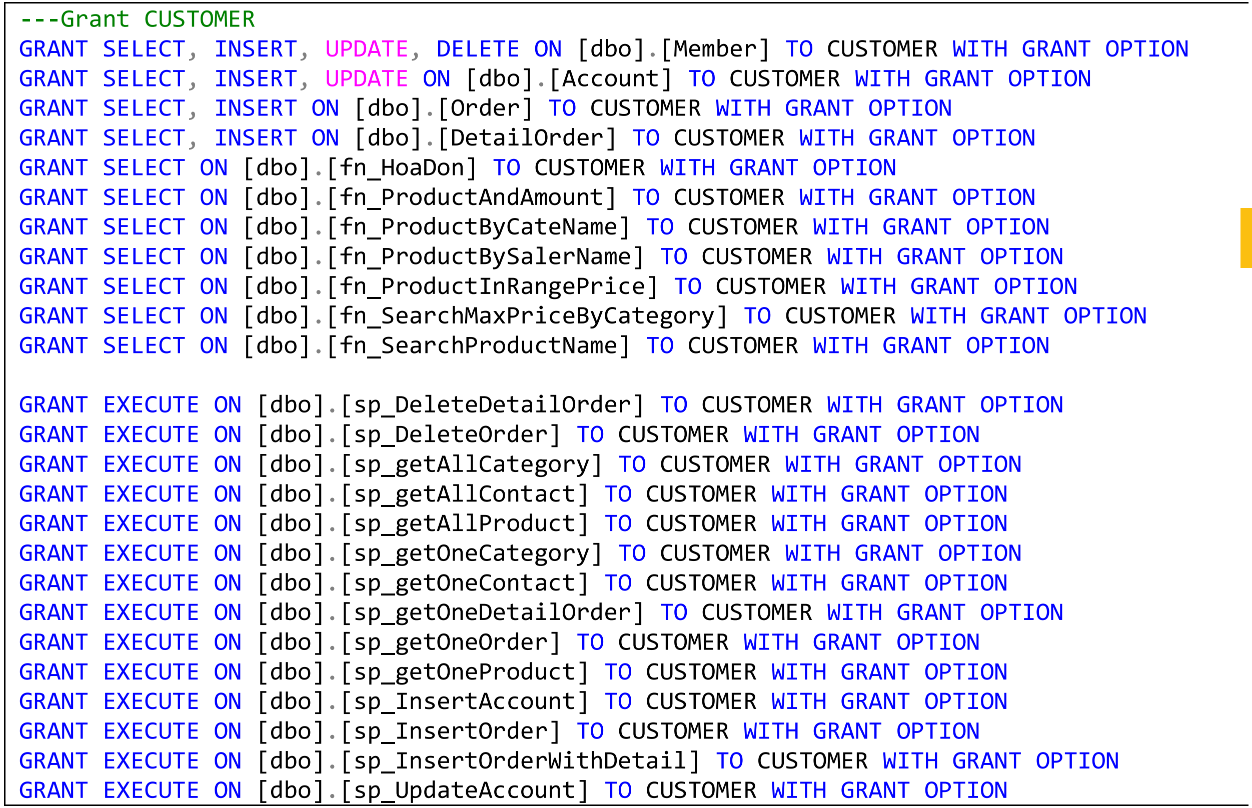


```
--Grant ADMIN
GRANT EXECUTE ON [dbo].[sp_DeleteAccount] TO ADMIN WITH GRANT OPTION
GRANT EXECUTE ON [dbo].[sp_DeleteBill] TO ADMIN WITH GRANT OPTION
GRANT EXECUTE ON [dbo].[sp_DeleteCategory] TO ADMIN WITH GRANT OPTION
GRANT EXECUTE ON [dbo].[sp_DeleteContact] TO ADMIN WITH GRANT OPTION
GRANT EXECUTE 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_getDetailOder] 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
```

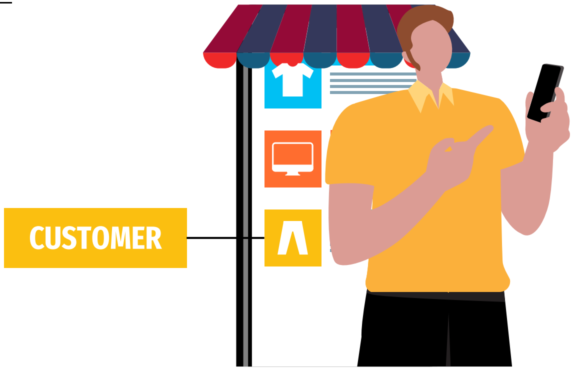
TOÀN QUYỀN



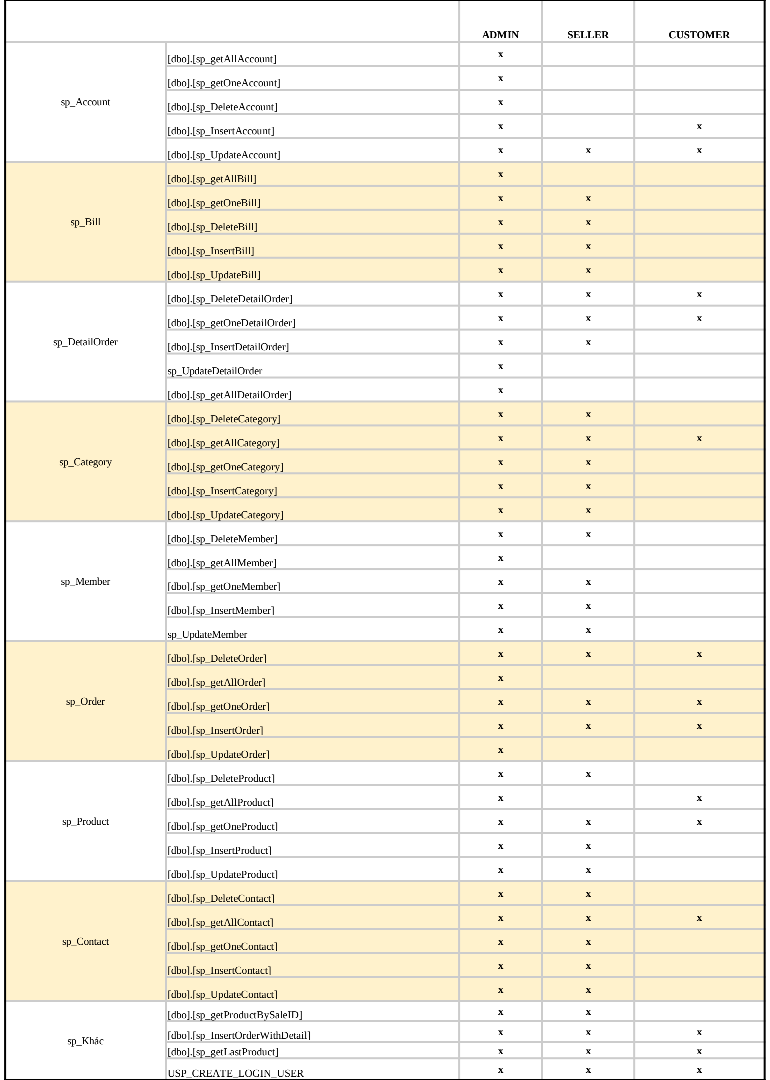
```
---Grant SELLER
GRANT SELECT, INSERT, UPDATE, DELETE ON [dbo].[Product] TO SELLER WITH GRANT OPTION
GRANT SELECT, INSERT, DELETE ON [dbo].[Order] TO SELLER WITH GRANT OPTION
GRANT SELECT, INSERT ON [dbo].[DetailOrder] TO SELLER WITH GRANT OPTION
GRANT SELECT, INSERT, UPDATE, DELETE ON [dbo].[Member] 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_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
```



```sql
---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
```

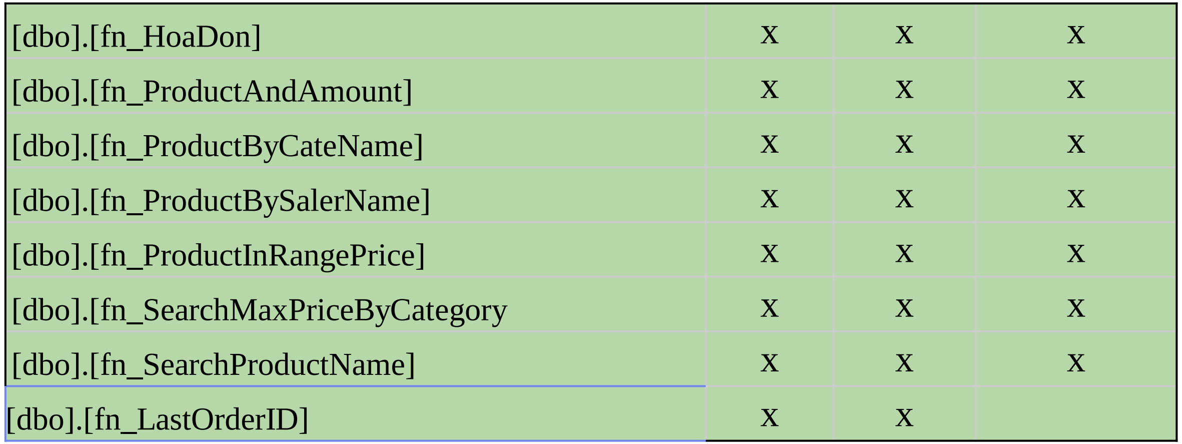


PHÂN QUYỀN STORED PROCEDURE



		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
	[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

PHÂN QUYỀN FUNCTIONS



[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
[dbo].[fn_LastOrderID]	x	x	

ADMIN
SELLER
CUSTOMER

PHÂN QUYỀN BẢNG

		ADMIN	SELLER	CUSTOMER
[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	

CODE CÀI ĐẶT

WEBSITE KINH DOANH GIÀY VÀ PHỤ KIỆN

CODE CÀI ĐẶT

TÌM KIẾM SẢN PHẨM

```java
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;
    }
```

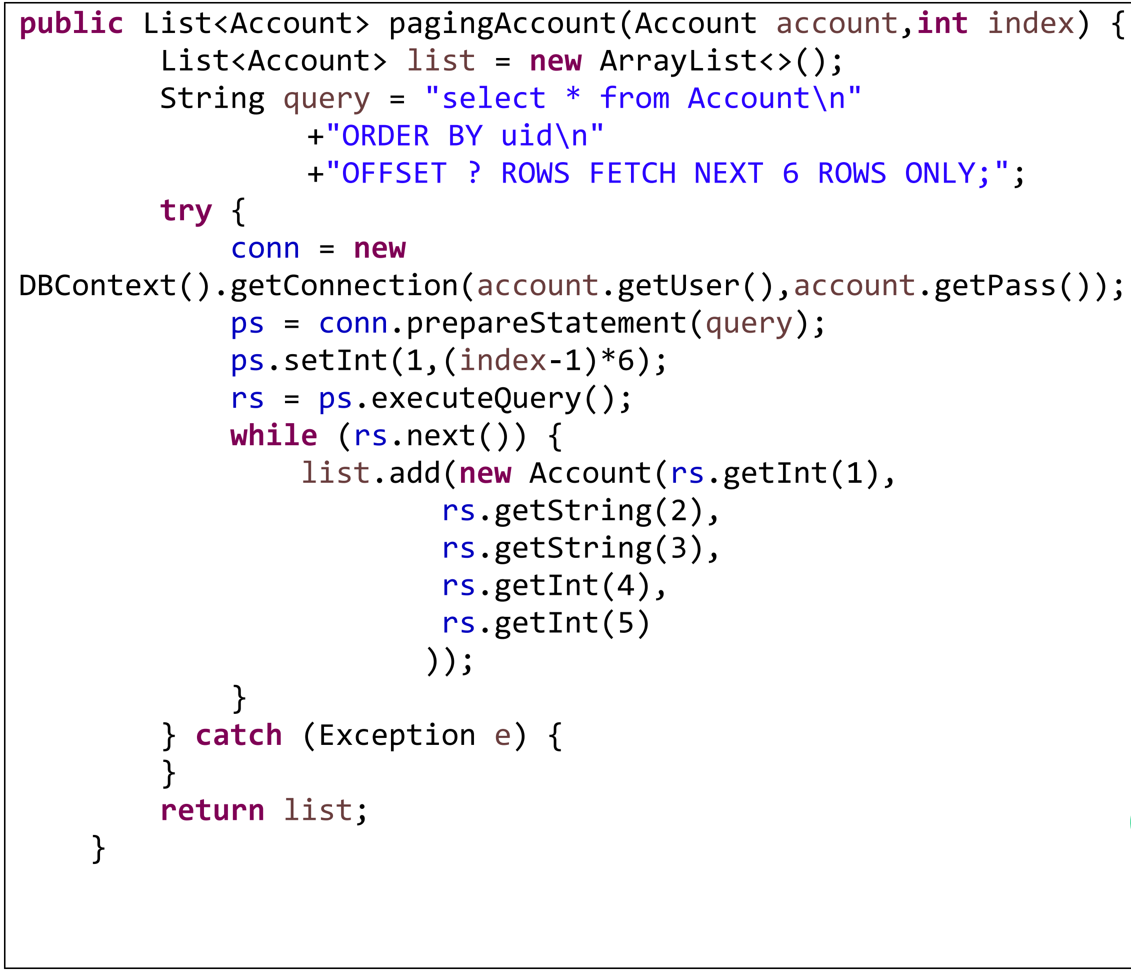
CODE CÀI ĐẶT

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;
    }
```

CODE CÀI ĐẶT

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;
    }
```

CODE CÀI ĐẶT

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) {
        }
    }
```

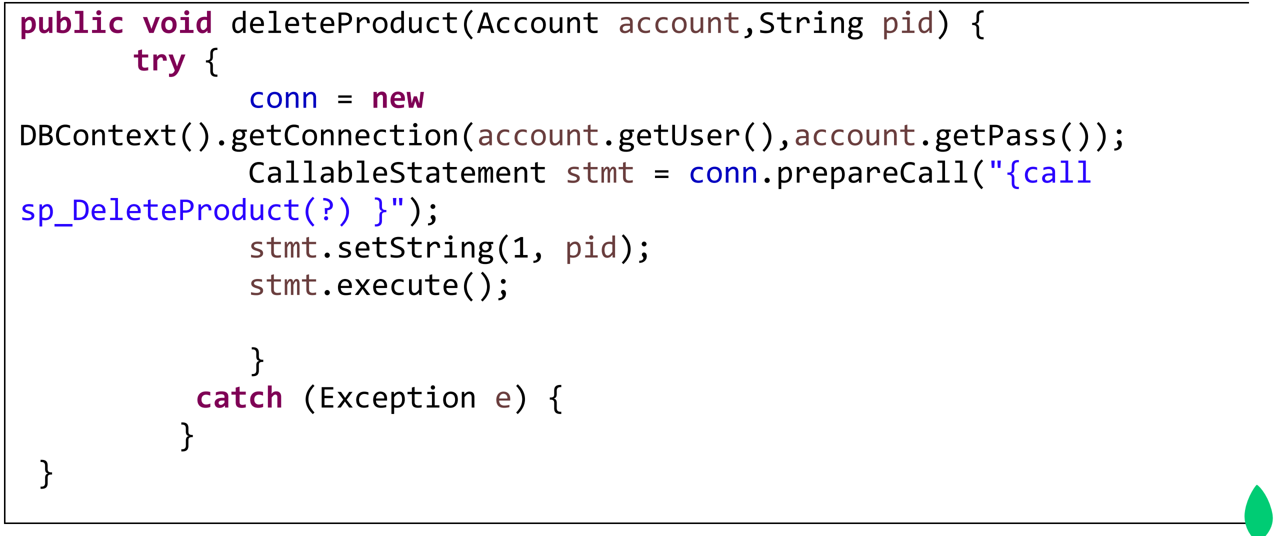
CODE CÀI ĐẶT

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) {
        }
    }
```

CODE CÀI ĐẶT

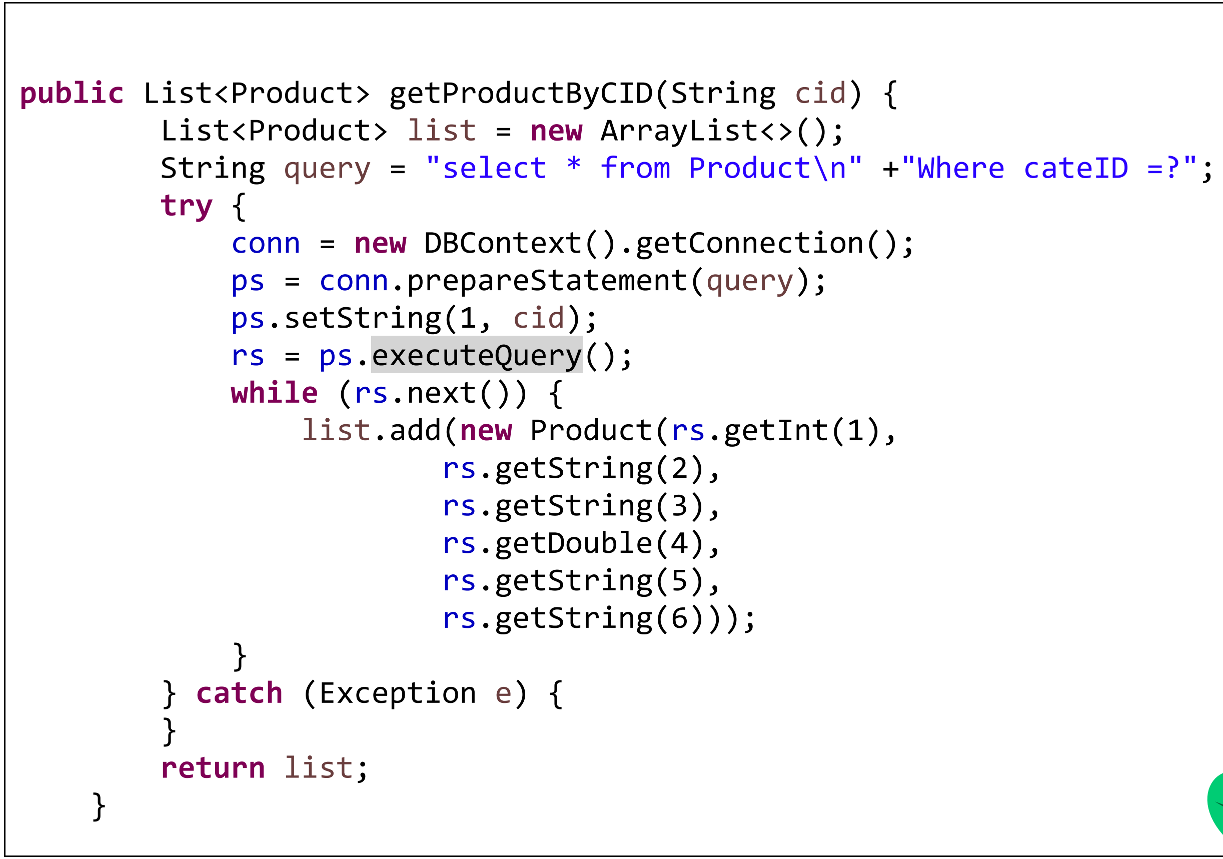
XÓA SẢN PHẨM



```
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) {
          }
 }
```

CODE CÀI ĐẶT

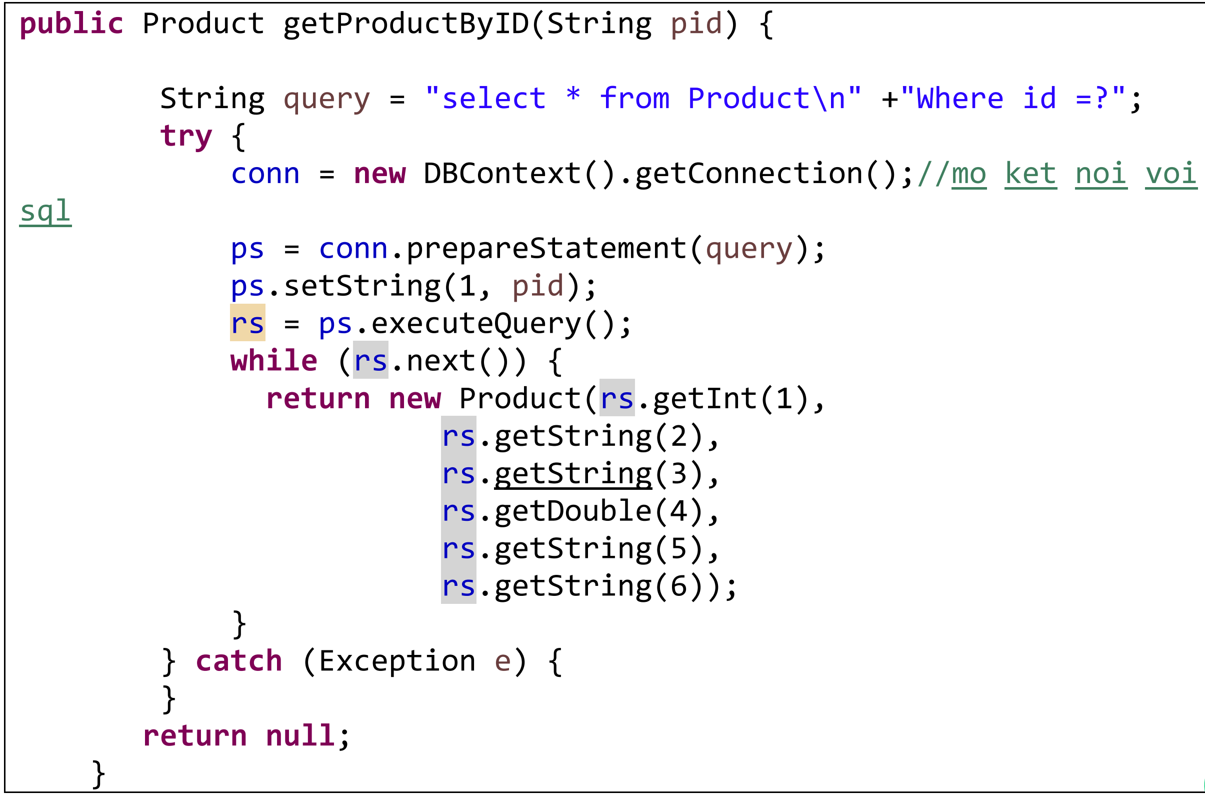
LOAD SẢN PHẨM THEO CATEGORY



```
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;
    }
```

CODE CÀI ĐẶT

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;
    }
```

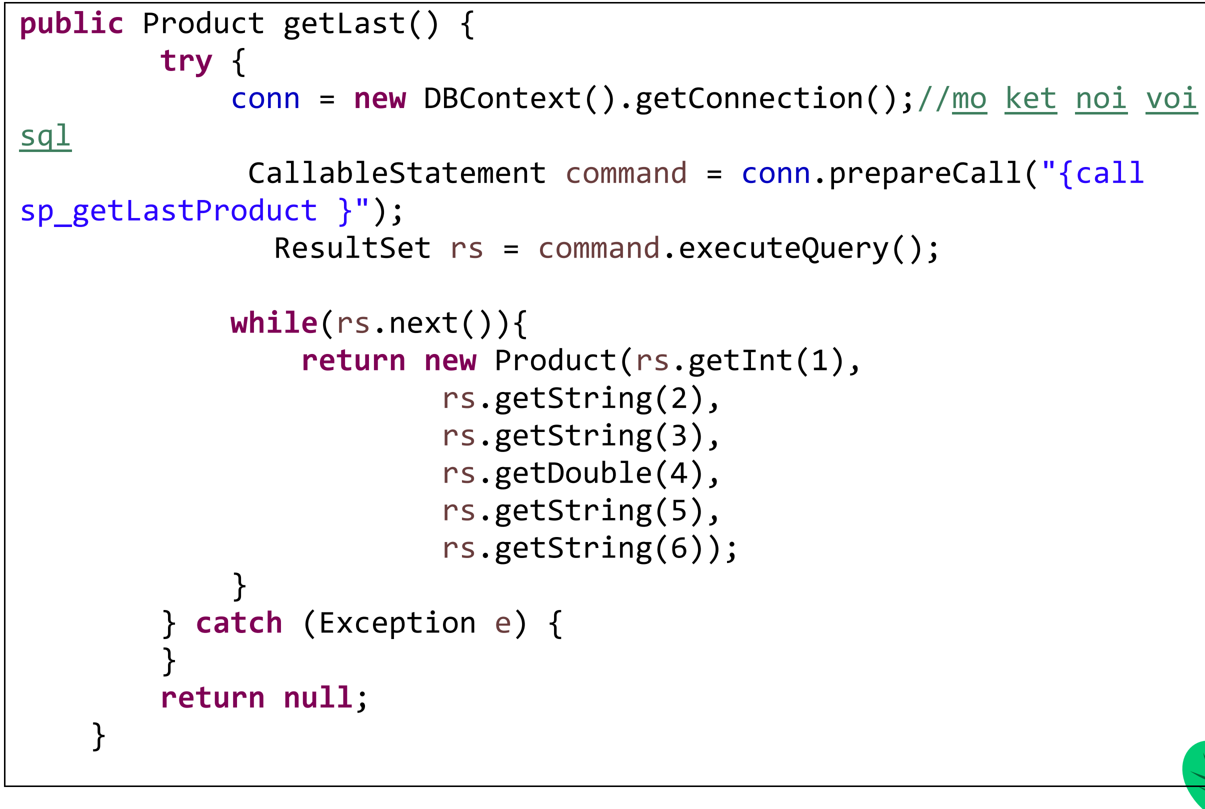
CODE CÀI ĐẶT

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;
	}
```

CODE CÀI ĐẶT

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;
    }
```

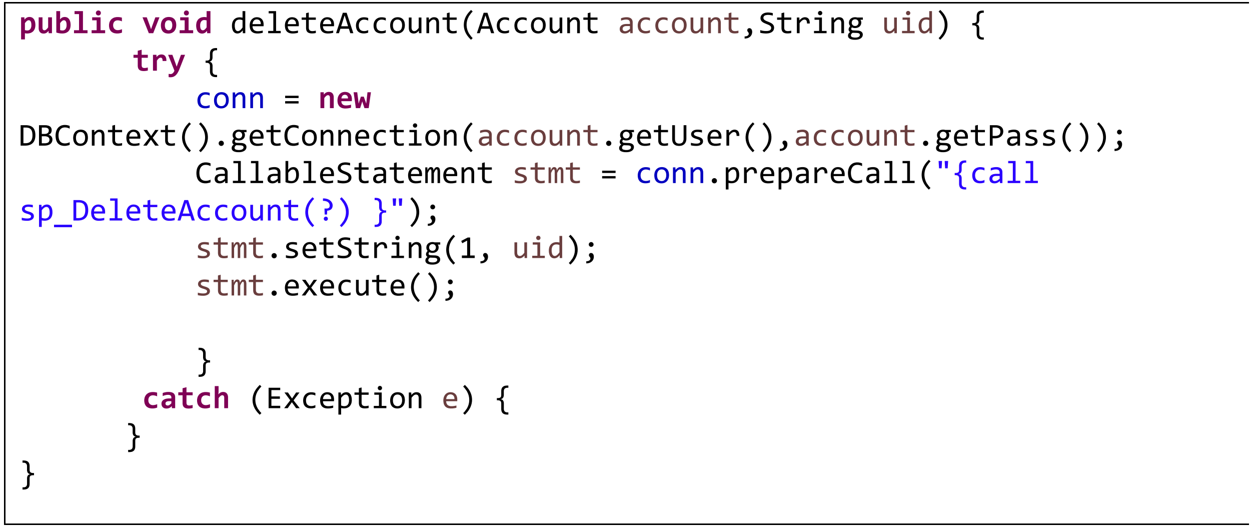
CODE CÀI ĐẶT

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) {
        }
    }
```

CODE CÀI ĐẶT

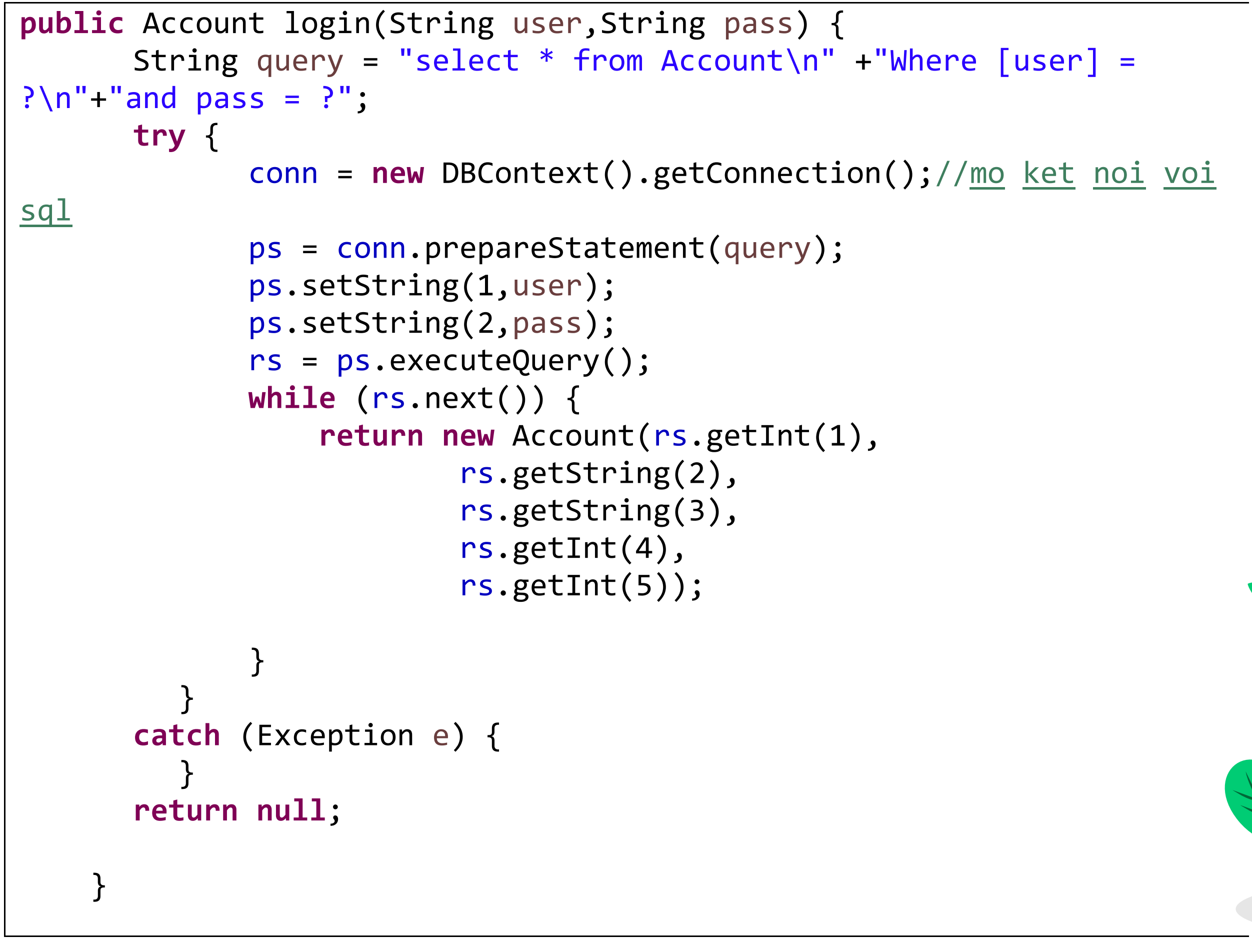
XÓA ACCOUNT



```
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) {
		}
}
```

CODE CÀI ĐẶT

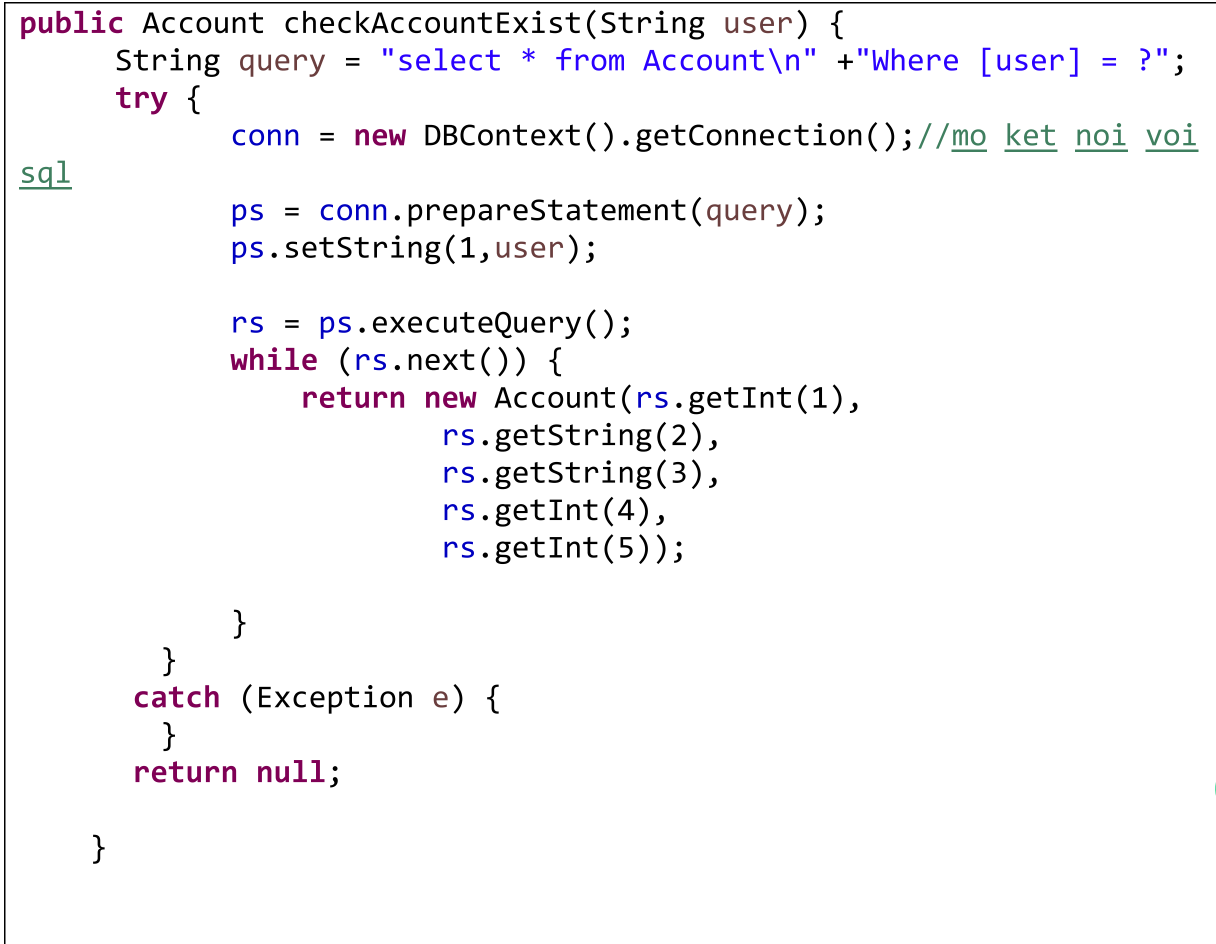
ĐĂNG NHẬP



```
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;

    }
```

CODE CÀI ĐẶT

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;

    }
```

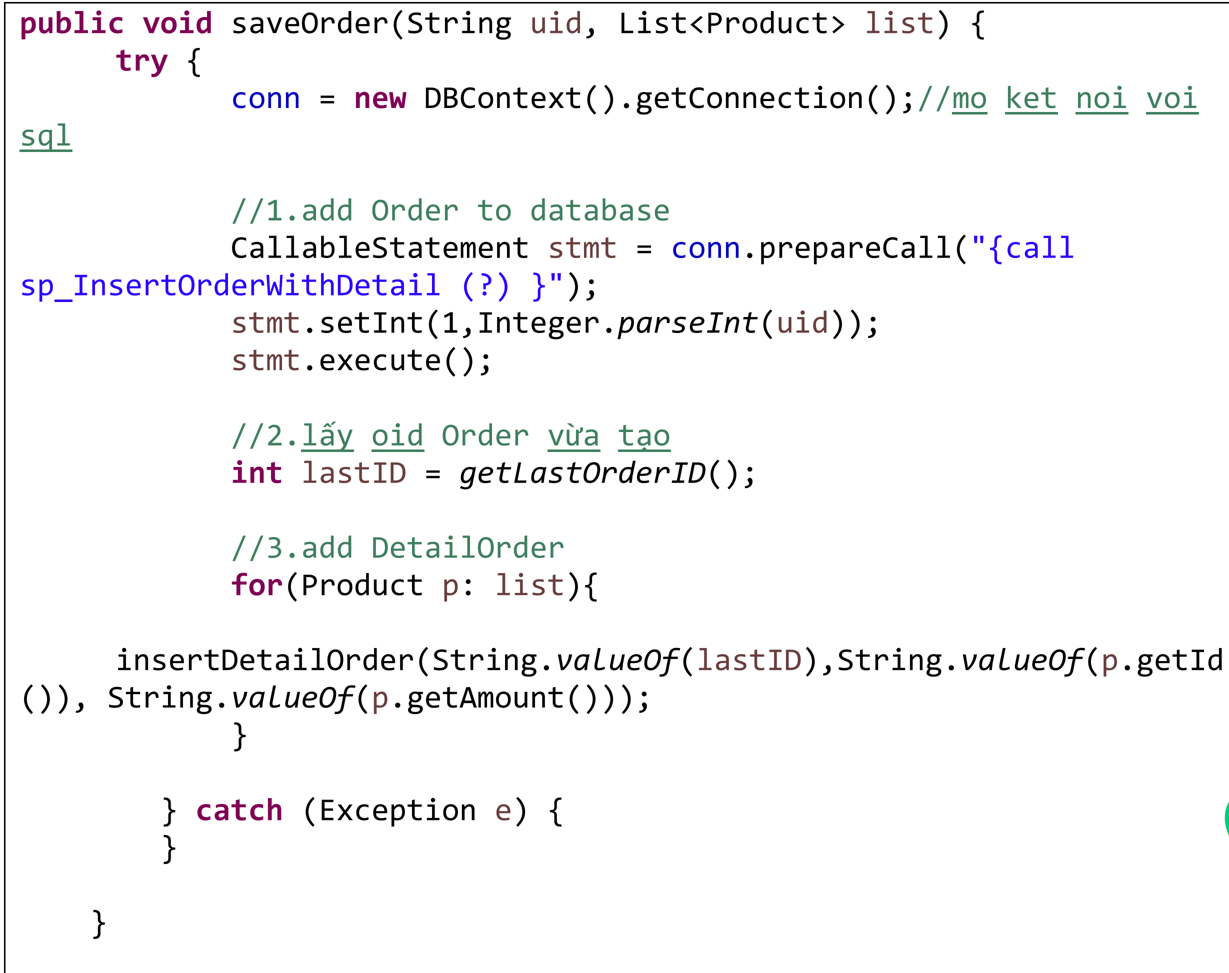
CODE CÀI ĐẶT

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) {
         }

    }
```

CODE CÀI ĐẶT

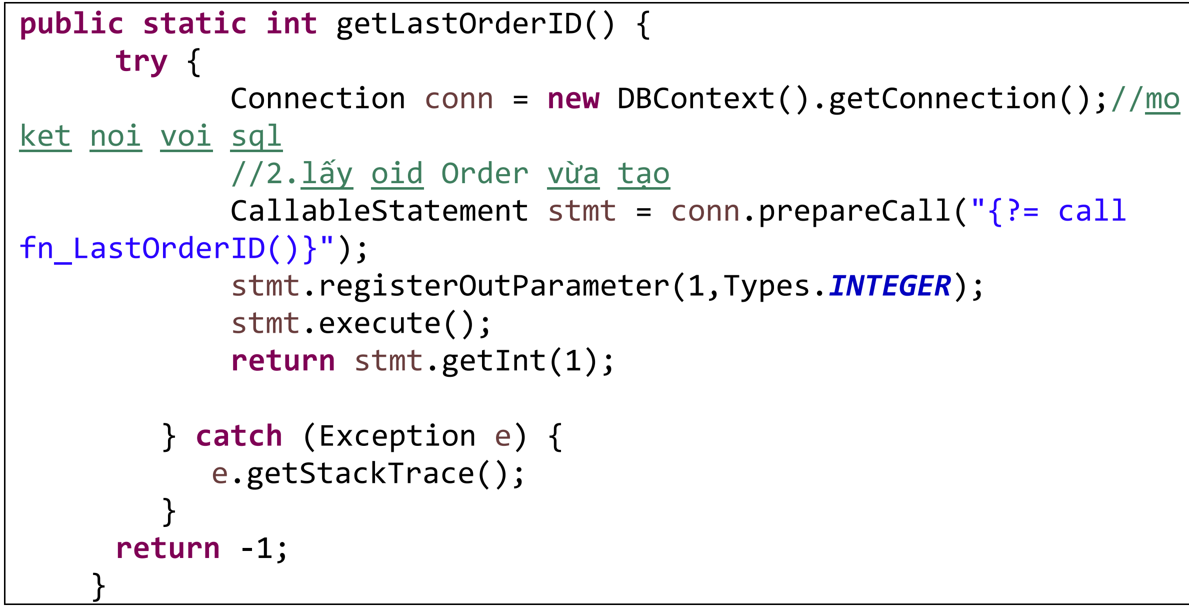
THÊM THÔNG TIN SẢN PHẨM VÀO BẢNG ORDER VÀ DETAIL ORDER



```
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) {
        }

    }
```

CODE CÀI ĐẶT

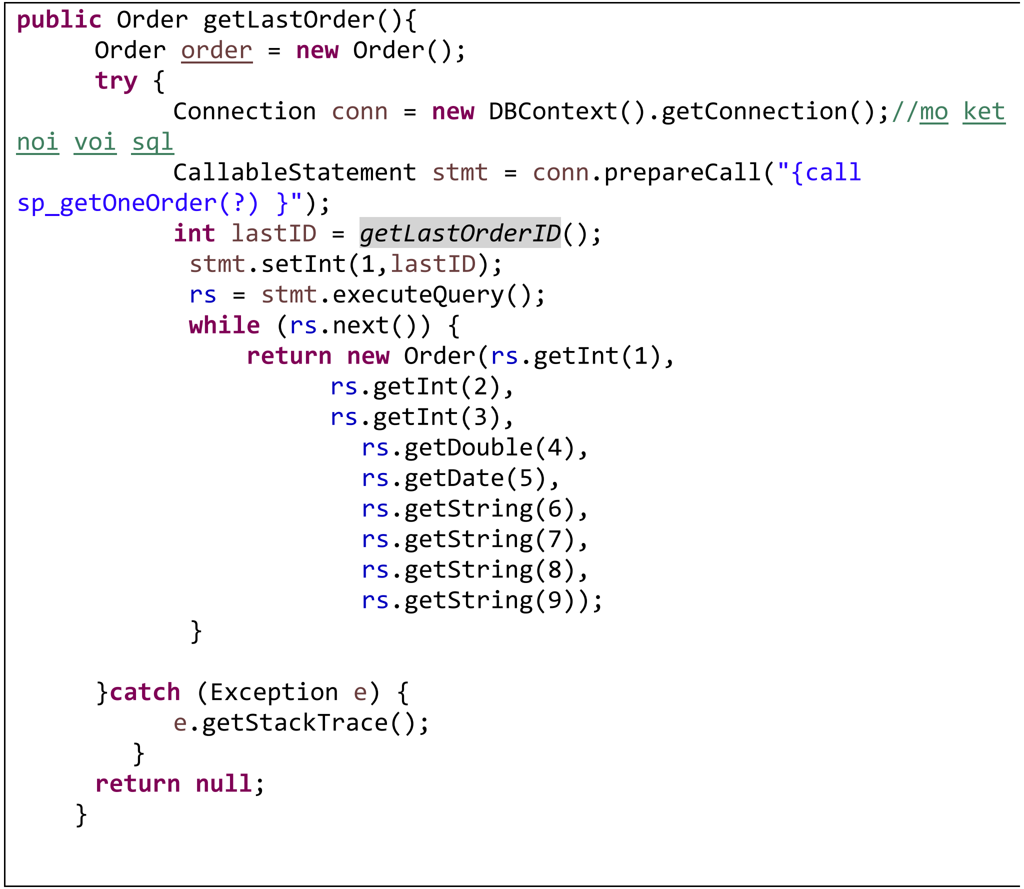
LẤY OID ORDER VỪA TẠO



```
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;
    }
```

CODE CÀI ĐẶT

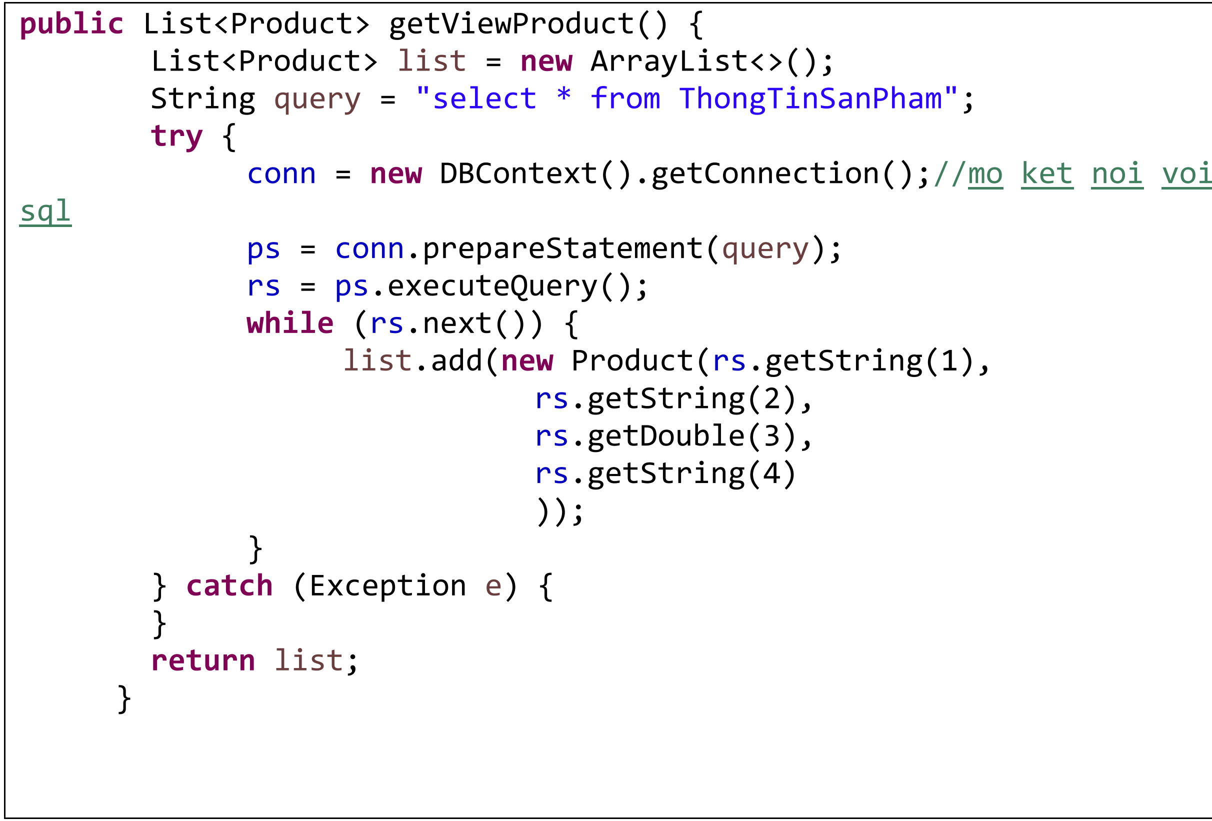
LẤY RA ORDER VỪA TẠO



```
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;
    }
```

CODE CÀI ĐẶT

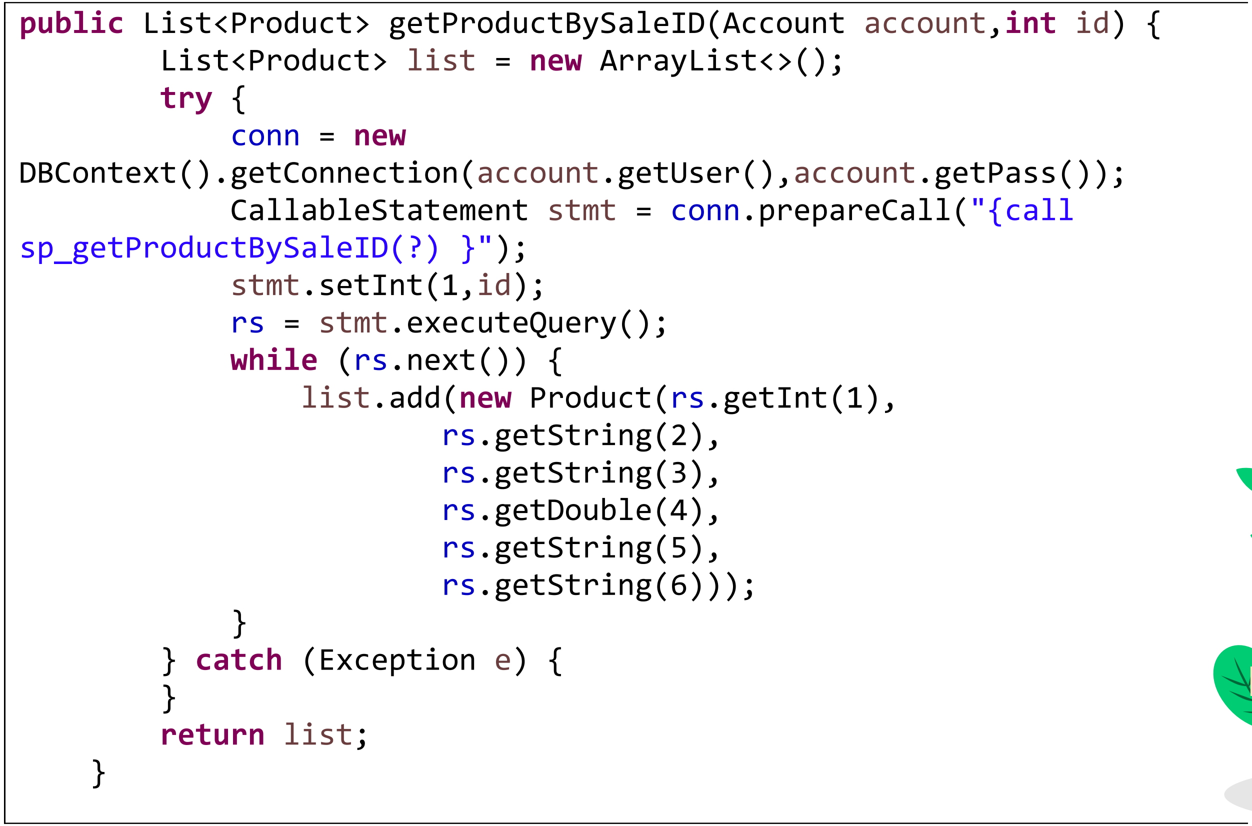
XEM THÔNG TIN TRONG VIEW THONGTINSANPHAM



```
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;
     }
```

CODE CÀI ĐẶT

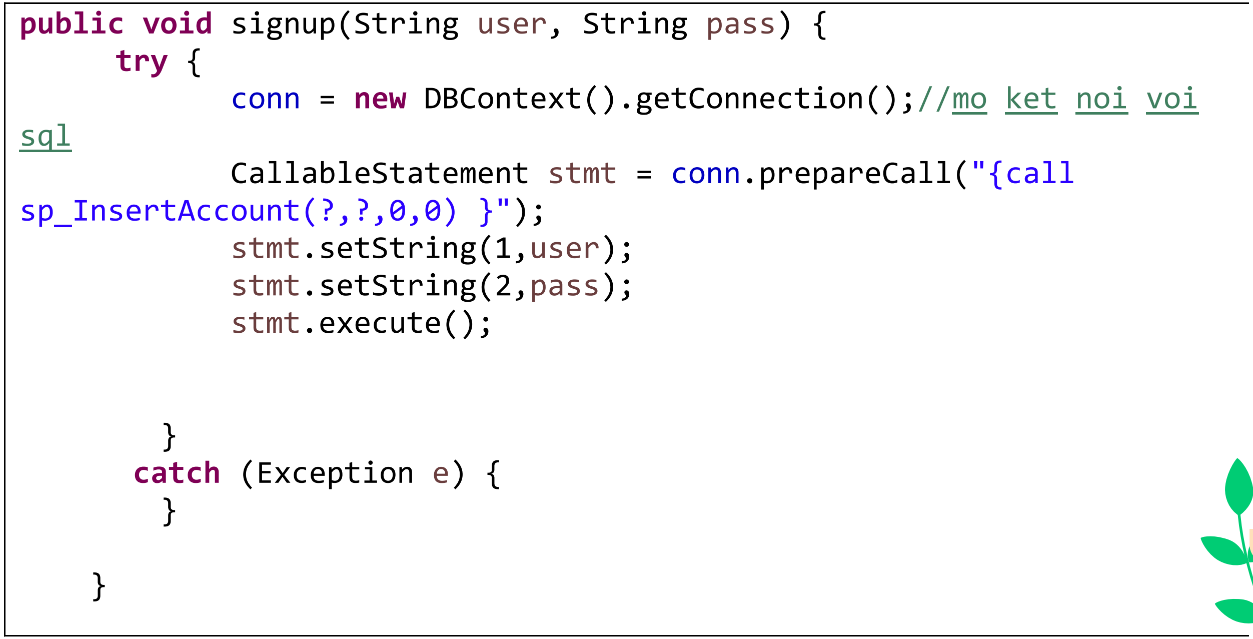
LẤY NHỮNG PRODUCT MÀ DO NGƯỜI ĐÓ BÁN ĐỂ LOAD LÊN TRONG MANAGE PRODUCT



```
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;
    }
```

CODE CÀI ĐẶT

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) {
         }

     }
```

DEMO

WEBSITE KINH DOANH GIÀY VÀ PHỤ KIỆN

THANK YOU

WEBSITE KINH DOANH GIÀY VÀ PHỤ KIỆN