

-- 1. PROCEDURE, FUNCTION

-- 1.1 Thêm dữ liệu vào bảng NganHang, ChiNhanh. Nếu thành công trả về 1, ngược lại trả về 0.

delimiter //

USE qlnh//

create procedure them(Manh int, Macn varchar(4), ThanhPhocn varchar(45), Taisan float)

begin

insert into qlnh.chinhanh(MaNH, MaCN, ThanhPhoCN, TaiSan) values (Manh, Macn, ThanhPhocn, Taisan);

end //

delimiter ;

call qlnh.them\_chinhanh('10','CN05','Hai Phong','20000000');

delimiter //

USE qlnh//

create procedure them\_nganhang(Manh int, Tennh varchar(45))

begin

insert into qlnh.chinhanh(MaNH, TenNH) values (Manh, TenNH);

end //

delimiter ;

call qlnh.them\_nganhang('10','Ngan Hang agribank');

-- 1.2 Cập nhật dữ liệu cho bảng NganHang, KháchHang. Nếu thành công trả về 1. Ngược lại trả về 0.

delimiter //

USE qlnh//

create procedure sua\_nganhang(Manh int, Tennh varchar(45))

begin

SET SQL\_SAFE\_UPDATES = 0;

update qlnh.nganhang set MaNH = Manh, TenNH = Tennh where MaNH = '5';

end //

delimiter ;

call qlnh.sua\_nganhang('6','Ngan Hang Vietcombank');

delimiter //

USE qlnh //

create procedure sua\_khachhang(Makh int, Tenkh varchar(45), Diachi varchar(45))

begin

SET SQL\_SAFE\_UPDATES = 0;

update qlnh.khachhang set MaKH = Makh, TenKH = Tenkh, DiaChi = Diachi where MaKH = '111222333';

end //

delimiter ;

call qlnh.sua\_khachhang('1234','Ho Van Tham','95 Ly Nam De');

-- 1.3 Xóa, dữ liệu trong các bảng KháchHang, TaiKhoanGoi. Nếu thành công trả về 1, ngược lại trả về 0.

```
delimiter //
USE qlnh//
create procedure xoa_khachhang()
begin
    declare Makh int;
    delete from qlnh.khachhang where MaKH = Makh;
end //
delimiter ;
set Makh = '111222333';
call xoa_khachhang();
```

```
delimiter //
USE qlnh//
create procedure xoa_taikhoangoi()
begin
    declare TKG int;
    delete from qlnh.taikhoangoi where SoTKG = TKG;
end //
delimiter ;
set TKG = '00001A';
call xoa_taikhoangoi();
```

-- 1.4 Đưa vào số tài khoản, nếu tài khoản gửi đó không tồn tại, trả về 0, ngược lại trả về số tiền gửi và mã khách hàng của người đó.

```
delimiter //
USE qlnh//
create function Sotk(Sotkg varchar(45))
returns nvarchar(255)
begin
    declare kq nvarchar(255);
    if Sotkg is null then
        set kq = (select concat('Kết quả: 0') as KiemTra);
    else
        set kq = (select MaKH, SoTienGoi from qlnh.taikhoangoi where SoTKG = Sotkg)
    ;
    end if;
    return kq;
end//
delimiter ;
select MaKH, SoTienGoi, Sotk('CN01') from qlnh.taikhoangoi;
```

-- 1.5 Đưa vào mã khách hàng, nếu mã khách hàng đó không tồn tại, trả về 0, ngược lại trả về tên và địa chỉ của khách hàng đó.

```
delimiter //
USE qlnh//
create function MaKhachHang(Makh int)
returns nvarchar(255)
begin
declare kq nvarchar(255);
    if Makh is null then
        set kq = (select concat('Kết quả: 0') as KiemTra);
    else
        set kq = (select TenKH, DiaChi from qlnh.khachhang where MaKH = Makh) ;
    end if;
    return kq;
end//
delimiter ;
select TenKH, DiaChi, MaKhachHang('111222333') from qlnh.khachhang;
```

-- 2. VIEW

-- 2.1 Xem tất cả những thành phố mà có chi nhánh của ngân hàng công thương.

```
CREATE VIEW ThanhPho_VIEW AS
SELECT ThanhPhoCN
FROM chinhanh a, nganhang b
WHERE a.MaNH = b.MaNH and b.TenNH like 'Ngan Hang Cong Thuong'
WITH CHECK OPTION;
select * from thanhpho_VIEW;
```

-- 2.2 Xem thông tin về tên ngân hàng và tổng tài sản của các ngân hàng.

```
CREATE VIEW NganHang_VIEW AS
SELECT TenNH, TaiSan
FROM chinhanh a, nganhang b
WHERE a.MaNH = b.MaNH
WITH CHECK OPTION;
select * from NganHang_VIEW;
```

-- 2.3 Xem MaCN và TaiSan của chi nhánh có tài sản lớn nhất.

```
CREATE VIEW CNTSLonNhat_VIEW AS
SELECT MaCN, MAX(TaiSan)
FROM chinhanh ;
select * from CNTSLonNhat_VIEW;
```

-- 2.4 Xem tổng số tiền mà mỗi chi nhánh ngân hàng đang được khách hàng gửi.

```
CREATE VIEW TongSoTien_VIEW AS
```

```
SELECT a.MaCN, SUM(a.SoTienGoi) as TongSoTien
FROM qlnh.taikhoangoi a
GROUP BY a.MaCN;
select * from TongSoTien_VIEW;
```

-- 3 Trigger

-- 3.1 Tạo trigger khi xóa dữ liệu khách hàng thì tài khoản gửi và vay của khách hàng đó cũng xóa.

```
DROP TRIGGER IF EXISTS HandleTriggerDeleteCustomer;
DELIMITER $$
CREATE TRIGGER HandleTriggerDeleteCustomer
BEFORE DELETE ON `KhachHang`
FOR EACH ROW
BEGIN
    Set @maKH = OLD.MaKH;
    DELETE FROM `TaiKhoanGoi` WHERE MaKH = @maKH;
    DELETE FROM `TaiKhoanVay` WHERE MaKH = @maKH;
END $$
```

-- 3.2. Tạo trigger khi xóa dữ liệu chi nhánh ngân hàng thì các tài khoản gửi và vay thuộc chi nhánh đó cũng xóa.

```
DROP TRIGGER IF EXISTS HandleTriggerDeleteBranch;
DELIMITER $$
CREATE TRIGGER HandleTriggerDeleteBranch
BEFORE DELETE ON `ChiNhanh`
FOR EACH ROW
BEGIN
    Set @maCN = OLD.MaCN;
    DELETE FROM `TaiKhoanGoi` WHERE MaCN = @maCN;
    DELETE FROM `TaiKhoanVay` WHERE MaCN = @maCN;
END $$
```

-- test

```
DELETE FROM `ChiNhanh` WHERE MaCN = 'CN01';
```

-- 3.3 tạo trigger để không cho phép xóa 2 mẫu tin trong bảng NganHang đồng thời.

```
DROP TRIGGER IF EXISTS NotAloDeleteTwoRecords;
DELIMITER $$
CREATE TRIGGER NotAloDeleteTwoRecords
BEFORE DELETE ON `NganHang`
FOR EACH ROW
BEGIN
    IF(@row_being_deleted IS NULL) THEN
        SET @row_being_deleted = 1;
```

```
ELSE
    SET @row_being_deleted = NULL;
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Không thể xóa từ 2 ngân hàng cùng lúc trở lên!';
END IF;
END $$
```