

1. Tạo CSDL QLBanHang với kích thước ban đầu lúc khởi tạo của tập tin dữ liệu chính là 50MB, tự động tăng kích thước lên 10% khi dữ liệu đầy, kích thước tăng trưởng tập tin dữ liệu tối đa không quá 200MB. Tập tin lưu vết với kích thước ban đầu là 10MB, tự động tăng trưởng lên 5MB khi dữ liệu đầy, kích thước tăng trưởng tập tin không giới hạn.

➤ Tạo các Table sau:

VT

- tblVATTU: **MaVTu** char(4), TenVtu nvarchar(30), DVTTinh nvarchar(10), Dongia int.

DH

- tblDONDH: **SoDH** Int, NgayDH datetime, Manhacc char(4)

CTDH

- tblCTDONDH: **SoDH** Int, **MaVTu** char(4), SLDat int

NC

- tblNHACC: **Manhacc** char(4), Tennhacc nvarchar(30), Diachi nvarchar(30), Dienthoai char(15)

PN

- tblPNHAP: **SoPN** char(4), SoDH Int, Ngaynhap datetime,

CTPN

- tblCTPNHAP: **SoPN** char(4), **MaVTu** char(4), Slnhap int.

PX

- tblPXUAT: **SoPX** char(4), Ngayxuat datetime.

CTPX

- tblCTPXUAT: **SoPX** char(4), **MaVTu** char(4), SLXuat int.

- tblTONKHO: NAMTHANG char(7), MaVTu char(4), TONGNHAP int, TONGXUAT int, SLTonkho int

Các bước làm câu lệnh SELECT:

Bước 1:

- Bôi đen các trường cần (hạn chế tối thiểu các bảng)
- Các bảng tương ứng với các trường trên, và cho các kết nối

Bước 2: Gõ câu lệnh SELECT:

- Cú pháp:

Select Danh_sách_các_cột

```
From Tên_bảng [Bí_danh]
[ Inner | Left | Right | Full Join Tên_bảng_qh [Bí_danh] On điều_kiện_quan_hệ... ]
[ Where Điều_kiện ]
[ Group By Danh_sách_cột_nhóm_dl ]
[ Having Điều_kiện_lọc_nhóm ]
[ Order By Tên_cột_sx Desc,...]
```

Bước 3: [Where Điều_kiện]

Bước 4: [Group By Danh_sách_cột_nhóm_dl]: khi đề ra thấy có tính TỔNG, ĐÉM, TRUNG BÌNH, MAX, MIN,...

- Viết các trường cần vào * sau SELECT (**CẨN THẬN**)
- Viết các hàm vào trường cần
- Gõ Group By, và COPY các trường sau SELECT nhưng không gõ hàm vào đây

Bước 5: [Having Điều_kiện_lọc_nhóm]

Lưu ý: nếu câu SELECT phải dùng GROUP BY mà có điều kiện:

- Nếu đk liên quan đến **trường không có hàm** thì ta chỉ cần dùng WHERE
- Nếu đk liên quan đến **trường có hàm** thì ta phải dùng HAVING
- Nếu viết đk bằng WHERE thì cũng có thể viết bằng HAVING, nhưng ưu tiên dùng WHERE

Khoá ngoại:

- + Sodh trong Table CTDONDH tham chiếu sodh trong DONDH
- + Sopn trong Table CTPNHAP tham chiếu sopn trong PNHAP
- + MaVTu trong CTDONDH tham chiếu MaVTu trong VATTU
- + MaVTu trong TONKHO tham chiếu MaVTu trong VATTU
- + MaVTu trong CTPNHAP tham chiếu MaVTu trong VATTU
- + Manhacc trong DONDH tham chiếu Manhacc trong NHACC

2. Bảng ảo – VIEW

- 2.1. Hiện thị thông tin SoDH, MaVTu, SLDat, NgayDH, Manhacc của 2 bảng tblCTDonDH và tblDonDH.
- 2.2. Hiện thị thông tin trong bảng DONDH và 2 cột địa chỉ, Tên nhà cung cấp trong bảng NHACC
- 2.3. Hiện thông tin các đơn đặt hàng trong bảng DONDH và Họ tên nhà cung cấp trong bảng NHACC với yêu cầu sắp xếp theo mã nhà cung cấp tăng dần.
- 2.4. Hiện thông tin Nhà cung cấp, số đặt hàng, tên vật tư, số lượng đặt hàng.
- 2.5. Hiện thông tin Nhà cung cấp, số đặt hàng, tên vật tư, số lượng đặt hàng có số lượng đặt hàng $\geq A$.
- 2.6. Hiện các phiếu nhập, tên vật tư, số lượng nhập theo ngày nhập hàng tăng dần.
- 2.7. Hiện số lượng đặt hàng của các vật tư có đơn đặt hàng là N.
- 2.8. Hiện số lượng đặt hàng của các vật tư có số lượng đặt hàng $\geq N$
- 2.9. Hiện số lượng xuất hàng của các vật tư có phiếu xuất là N.
- 2.10. Hiện số lượng xuất hàng của các vật tư có số lượng xuất hàng $\geq N$
- 2.11. Hiện số lượng nhập hàng của các vật tư có đơn đặt hàng là N.
- 2.12. Hiện số lượng nhập hàng của các vật tư có số lượng đặt hàng $\geq N$
- 2.13. Hiện tổng số lượng đặt hàng của các vật tư.
- 2.14. Hiện các vật tư có tổng lượng đặt hàng $\geq N$.
- 2.15. Hiện tổng số lượng bán hàng của các vật tư.
- 2.16. Hiện tổng số lượng bán hàng của các vật tư có tên là Gạch và Ngói.
- 2.17. Thông kê tổng số lượng nhập hàng của các vật tư

- 2.18. Thống kê tổng số lượng nhập hàng của các vật tư có tên vật tư bắt đầu là G
- 2.19. Thống kê mỗi vật tư có bao nhiêu số đơn hàng (SoDH)
- 2.20. Hiện tổng số đơn hàng (SoDH) của các vật tư có tên vật tư là Gạch và Ngói
- 2.21. Thống kê mỗi vật tư có bao nhiêu số nhập hàng (SoPN)
- 2.22. Hiện tổng số phiếu nhập hàng (SoPN) của các tên vật tư có tổng số phiếu nhập hàng $\geq N$
- 2.23. Thống kê mỗi vật tư có bao nhiêu số phiếu xuất (SoPX)
- 2.24. Hiện tổng số phiếu xuất (SoPX) của các tên vật tư bắt đầu là G
- 2.25. Thống kê tổng số đơn đặt hàng mà công ty đã đặt hàng theo từng nhà cung cấp và sắp xếp dữ liệu theo tổng số đơn đặt hàng tăng dần.
- 2.26. Bảng câu trên nhưng lọc những bản ghi có mã nhà cung cấp bắt đầu = 'C' và tổng số đơn đặt hàng > 1
- 2.27. Tính tổng số lượng đặt hàng của các vật tư theo từng nhà cung cấp
- 2.28. Tính tổng số lượng nhập hàng của các vật tư theo từng nhà cung cấp
- 2.29. Tính tổng số lượng đặt, tổng số lượng nhập của các vật tư theo các nhà cung cấp

3. Truy vấn con

- 3.1. Hiện các đơn đặt hàng trong ngày gần đây nhất
- 3.2. Hiện SoDH, MaVTu có SLDat lớn nhất.
- 3.3. Hiện SoDH, Tên vật tư có SLĐặt lớn nhất.
- 3.4. Hiện SoDH, Tên nhà cung cấp, Tên Vật tư có số lượng đặt lớn nhất.
- 3.5. Hiện Tên vật tư có tổng sldat lớn nhất.
- 3.6. Hiện Tên nhà cung cấp có tổng số lượng đặt Tên Vật tư A lớn nhất.
- 3.7. Hiện Tên vật tư có tổng số lượng nhập hàng lớn nhất.
- 3.8. Hiện Tên vật tư có tổng số lượng bán hàng lớn nhất.
- 3.9. Hiện họ tên nhà cung cấp, Điện thoại của bảng NHACC mà công ty đã có đơn đặt hàng trong tháng 01-2002
- 3.10. Hiện danh sách các nhà cung cấp mà công ty chưa bao giờ đặt hàng.
- 3.11. Hiện danh sách vật tư mà công ty chưa bao giờ đặt hàng.
- 3.12. Hiện danh sách các nhà cung cấp chưa nhập hàng mà ta đã đặt hàng
- 3.13. Hiện danh sách các nhà cung cấp và số đơn đặt hàng chưa nhập hàng mà ta đã đặt hàng
- 3.14. Hiện danh sách các nhà cung cấp, số đơn đặt hàng, tên vật tư chưa nhập hàng đủ so với đơn đặt hàng
- 3.15. Hiện danh sách các nhà cung cấp, số đơn đặt hàng, tên vật tư chưa nhập hàng đủ so với đơn đặt hàng và số lượng còn thiếu**

4. Ví dụ về INSERT:

+ Tạo bảng ảo tính tổng số lượng nhập của các vật tư trong tháng 10/2002:

```
CREATE VIEW vw_tongnhap  
AS
```

```
SELECT ct.mavtu, SUM(ct.slnhap) AS Tongnhap  
FROM tblpnhap pn  
INNER JOIN tblctpnhap ct ON pn.sopn = ct.sopn  
WHERE (CONVERT(char(7), pn.ngaynhap, 21)) = '2010-02'  
GROUP BY ct.mavtu
```

+ Tạo bảng ảo dùng để tính tổng số lượng xuất của các vật tư trong tháng 01/2002:

```
CREATE VIEW vw_tongxuat  
AS
```

```
SELECT ct.mavtu, SUM(ct.slxuat) AS Tongxuat  
FROM tblpxuat px  
INNER JOIN tblctpxuat ct ON px.sopx = ct.sopx  
WHERE (CONVERT(char(7), px.ngayxuat, 21)) = '2010-02'  
GROUP BY ct.mavtu
```

+ Sử dụng INSERT...INTO để thêm dữ liệu vào bảng TONKHO:

```
INSERT tbltonkho
```

```
Select '02-2010', TN.Mavtu, TONGNHAP, TONGXUAT,  
TONGNHAP-TONGXUAT
```

```
From vw_tongnhap tn left join vw_tongxuat tx
```

```
On tn.mavtu=tx.mavtu
```

5. Ví dụ DELETE...

Huỷ bỏ các nhà cung cấp mà công ty chưa bao giờ đặt hàng:

```
delete nhacc from nhacc ncc
```

```
left join dondh dh on dh.manhacc=ncc.manhacc
```

```
where dh.sodh is null
```

Huỷ bỏ các vật tư mà công ty chưa bao giờ đặt hàng:

+

6. Ví dụ UPDATE...SET

BT: Tạo nội dung trường khen thưởng của bảng tblDSHS nếu tổng điểm của học sinh trong bảng tblDiem $\geq A$.

7. Ví dụ IF...ELSE...

BT: Nếu tổng số lượng nhập hàng của các vật tư theo Nhà cung cấp nào đó lớn hơn A thì hiện tên nhà cung cấp, tên vật tư; nếu không có hiện dòng thông báo.

+ Tìm xem có vật tư nào bán ra với số lượng >5 hay không?

```
if (select count(*) from tblctpxuat where slxuat>5) >0
begin
print 'danh sach các hàng hoá ban v?i so >5'
select distinct vt.mavtu,vt.tenvtu
from tblvattu vt
inner join tblctpxuat ct on vt.mavtu=ct.mavtu
where ct.slxuat>5
end
else
PRINT 'Chua ban hang nao voi so luong >5'
```

+ Lưu ý: if kết hợp với EXISTS với bài trên gõ:

```
if exists (select * from tblctpxuat where slxuat>5)
begin
print 'danh sach các hàng hoá ban v?i so >5'
select distinct vt.mavtu,vt.tenvtu
from tblvattu vt
inner join tblctpxuat ct on vt.mavtu=ct.mavtu
where ct.slxuat>5
end
else
print 'Chua ban hang nao voi so luong >5'
```

7. Ví dụ biểu thức CASE

Ví dụ 1:

```
select      CASE left(mavtu,2)
When 'DD' Then 'Dau DVD'
When 'GG' Then 'Biểu_thức_2'
when 'TV' then 'Tivi'
ELSE 'Biểu_thức_N '
END, tblVATTU.*
```

```
from tblVATTU
```

Ví dụ 2:

```
select CASE
When dongia>100 Then 'gia cao'
When dongia>50 Then 'gia vua'
ELSE 'gia thap '
```

```
END, tblVATTU.*
```

```
from tblVATTU
```

8. Biến kiểu dữ liệu CURSOR:

+ Ví dụ 1: Đọc dữ liệu cursor của bảng Vattu chỉ lọc các bản ghi có mã vật tư là B:

```
declare @s_mavtu char(4), @s_tenvtu nvarchar(20)
declare @s_dvtinh nvarchar(10),@s_dongia int
declare cur_vt cursor
keyset
for
    select * from tblvattu
    where tenvtu like 'B%'
        order by mavtu
open cur_vt

while 0=0
begin
    fetch next from cur_vt into @s_mavtu, @s_tenvtu,@s_dvtinh,@s_dongia
    if @@fetch_status<>0 break
    print @s_mavtu+' '+@s_tenvtu+@s_dvtinh+convert(char(10),@s_dongia)
end
close cur_vt
deallocate cur_vt
```

Bài tập: Sử dụng biến dữ liệu kiểu CURSOR làm các bài tập sau:

- 2.1. Hiện thông tin Nhà cung cấp, số đặt hàng, tên vật tư, số lượng đặt hàng.
- 2.2. Hiện thông tin Nhà cung cấp, số đặt hàng, tên vật tư, số lượng đặt hàng có số lượng đặt hàng $\geq A$.
- 2.3. Hiện các phiếu nhập, tên vật tư, số lượng nhập theo ngày nhập hàng tăng dần.
- 2.4. Hiện số lượng đặt hàng của các vật tư có đơn đặt hàng là N.
- 2.5. Hiện số lượng đặt hàng của các vật tư có số lượng đặt hàng $\geq N$
- 2.6. Hiện số lượng xuất hàng của các vật tư có phiếu xuất là N.
- 2.7. Hiện số lượng xuất hàng của các vật tư có số lượng xuất hàng $\geq N$
- 2.8. Hiện tổng số lượng đặt hàng của các vật tư.
- 2.9. Hiện các vật tư có tổng lượng đặt hàng $\geq N$.
- 2.10. Hiện tổng số lượng bán hàng của các vật tư.
- 2.11. Thống kê số lượng đặt hàng của các vật tư theo Nhà cung cấp.
- 2.12. Thống kê số lượng nhập hàng của các vật tư theo Nhà cung cấp.

9. Thủ tục

- Ví dụ 1: Xây dựng thủ tục tính tổng trị giá của một sodh

Input: số sodh

Output: Tổng tiền

```
ALTER proc tt_sodh @sdh char(4), @tt int output
```

```
as
```

```
select @tt = sum(vt.dongia*ct.sldat)
from TBLctdondh ct inner join TBLvattu vt on ct.mavtu=vt.mavtu
where ct.sodh=@sdh
group by ct.sodh
```


--Và gọi thủ tục trên:

```
declare @ttien int
exec tt_sodh '2',@ttien output
print 'Tong tien:' + str(@ttien,7)
```

1.2. Tạo thủ tục tính tổng tiền của 1 phiếu nhập hàng.

1.3. Tạo thủ tục tính tổng tiền của 1 phiếu xuất hàng.

- Ví dụ 2: Xây dựng thủ tục tính tổng số lượng nhập, Tổng số lượng xuất của 1 vật tư trong tháng năm nào đó:

Input: Tên vật tư, Tháng năm

Output: Sum(SLNhap) trong bảng Nhaphang, Sum(SLDat) trong CTDondh.

```
alter proc tt2 @tenvattu nvarchar(30),@ngay char(7),@Tx int output
```

```
as
```

```
Select vt.tenvtu, sum(Sldat) as Tongxuat into #bt
```

```
From tblCTDONDH ctdh
```

```
Inner join tbldondh dh on ctdh.sodh=dh.sodh
```

```
inner join tblvattu vt on vt.mavtu=ctdh.mavtu
```

```
where convert(char(7), ngaydh,21)=@ngay and vt.tenvtu=@tenvattu
```

```
Group by vt.tenvtu
```

```
select @tx=Tongxuat from #bt
```

```
Drop table #bt
```


```
declare @tong int
```

```
exec tt2 N'Bánh m?', N'2010-02',@tong output
```

```
print @tong
```

- Ví dụ 3: Xây dựng thủ tục trả về danh sách các tên vật tư có số lượng đặt hàng nhiều nhất trong năm tháng nào đó.

```
Alter proc Max_vt @ngay char(7)
```

```
AS
```

```
-- Tạo bảng tạm lưu Tên vật tư với Tổng số lượng đặt hàng giảm dần của  
-- năm tháng yêu cầu
```

```
SELECT VT.TenVtu, sum(CTDH.SLDat) as Tongsld into #bt
```

```
FROM tblCTDONDH ctdh
```

```
INNER JOIN tblVATTU vt ON CTDH.MaVTu = VT.MaVTu
```

```
INNER JOIN tblDONDH dh ON CTDH.SoDH = DH.SoDH
```

```
where convert(char(7),ngaydh,21)=@ngay
```

```
group by VT.TenVtu
```

```
order by sum(CTDH.SLDat) Desc
```

```
-- Hiện các Tên vật tư có số lượng đặt = Tổng số đặt hàng lớn nhất
```

```
Select tenvtu,tongsld from #bt
```

```
where tongsld=( select max(tongsld) from #bt)
```

```
--Xoá bảng tạm
```

```
Drop table #bt
```

```
--Gọi thực hiện thủ tục
```

```
Exec Max_vt N'2010-02'
```

```
-- Lệnh bỏ dòng đếm dữ liệu
```

```
Set nocount on
```

+ Ví dụ 5: Hiện vật tư nào có số lượng đặt hàng lớn nhất.

```
alter proc tt @ten nvarchar(20) output
```

```
as
```

```
SELECT tblvattu.tenvtu, SUM(tblctdondh.sldat) as tsld into #bt
```

```
FROM tblvattu INNER JOIN tblctdondh
```

```
ON tblvattu.mavtu = tblctdondh.mavtu
```

```
GROUP BY tblvattu.tenvtu
```

```
Select @ten = tenvtu
```

```
from #bt
```

```
where tsld =(select max(tsld) from #bt)
```

```
Drop table #bt
```

```
-----  
--Gọi thực hiện thủ tục  
Declare @tdv nvarchar(20)  
Exec tt @tdv output  
print @tdv
```

THỦ TỤC RETURN:

Ví dụ 4: Tạo thủ tục cho biết tiền phải trả cho nhà cung cấp A.
Input: tên nhà cung cấp
Output: Tổng tiền

```
alter proc tttt @ten nvarchar(30), @tt int output  
as
```

```
If not exists ( SELECT * FROM tblnhacc WHERE  
    Tennhacc =@ten )  
    return 1  
else  
If not exists (SELECT * FROM tblctdondh ct  
    INNER JOIN tbldondh dh ON ct.sodh = dh.sodh  
    INNER JOIN tblvattu vt ON ct.mavtu = vt.mavtu  
    INNER JOIN tblnhacc nc ON dh.manhacc = nc.manhacc  
    GROUP BY nc.tennhacc  
    HAVING (nc.tennhacc = @ten) )  
  
return 2  
else  
SELECT @tt=SUM(ct.sldat * vt.Dongia)  
FROM tblctdondh ct  
    INNER JOIN tbldondh dh ON ct.sodh = dh.sodh  
    INNER JOIN tblvattu vt ON ct.mavtu = vt.mavtu  
    INNER JOIN tblnhacc nc ON dh.manhacc = nc.manhacc  
    GROUP BY nc.Tennhacc  
    HAVING (nc.Tennhacc = @ten)  
return
```

```
-----  
--Lời gọi:
```

```

declare @tien int, @rt int
exec @rt =tttt N'Cơ Phát', @tien output
If @rt =1
print 'khong co nhacc nay'
else
If @rt = 2
print 'nhacc nay chua co gi'
else
print 'Tổng tiền :' + cast(@tien as char(10))

```

Bài tập:

- Viết thủ tục với đầu vào là số phiếu nhập, ra là tổng tiền của số phiếu nhập này, yêu cầu: nếu không tồn tại số phiếu nhập thì hiện dòng thông báo

- Tạo thủ tục cursor: Hiện số lượng nhập hàng của các vật tư có đơn đặt hàng là N

```

alter proc tt_cur1
    @s_sodh int, @s_cur cursor varying output
as
set @s_cur=cursor
keyset
for
    select TenVTu, sum(SLNhap)
        from tblVATTU vt
        inner join tblCTPNhap ct on ct.MaVTu=vt.MaVTu
        inner join tblPNhap pn on pn.SoPN=ct.SoPN
        where SoDH=@s_sodh
        group by vt.MaVTu, vt.TenVTu
open @s_cur
-----loi goi
declare @s_tenvtu nvarchar(20),@s_tsln int
declare @ss_cur cursor
exec tt_cur1 2,@ss_cur output
while 0=0
begin
    fetch next from @ss_cur into
@s_tenvtu,@s_tsln
        if @@FETCH_STATUS <>0 break
        print @s_tenvtu+str(@s_tsln,10)
end
close @ss_cur
deallocate @ss_cur

```

- Ví dụ 1: Viết thủ tục tính tổng số lượng đặt hàng của 1 vật tư trong tháng năm nào đó

```

alter proc tt2 @tenvattu nvarchar(30), @ngay char(7), @Tx int
output
as
if not exists(select * from tblVATTU where TenVTu=@tenvattu)
    return 1
else
    if not exists(Select *
        From tblCTDONDH ctdh
            Inner join tbldondh dh on ctdh.sodh=dh.sodh
            inner join tblvattu vt on vt.mavtu=ctdh.mavtu
    where convert(char(7), ngaydh, 21)=@ngay and vt.tenvtu=@tenvattu
        )
        return 2
    else

Select @Tx= sum(Sldat)
    From tblCTDONDH ctdh
        Inner join tbldondh dh on ctdh.sodh=dh.sodh
        inner join tblvattu vt on vt.mavtu=ctdh.mavtu
    where convert(char(7), ngaydh, 21)=@ngay and vt.tenvtu=@tenvattu

declare @tong int, @sb int
exec @sb=tt2 N'Gạch', N'2012-07', @tong output
if @sb=1
    print 'khong ton tai vt nay'
else
    if @sb=2
        print 'VT nay khong nhap trong tn nay'
    else
        print @tong

alter proc tt3 @snt char(7), @cur_vt CURSOR VARYING output
as

if not exists (select *
from tblVATTU vt inner join tblCTPXuat ct on ct.MaVTu =vt.MaVTu
            inner join tblPXuat px on px.SoPX=ct.SoPX
            where convert(char(7), Ngayxuat, 21)=@snt
        )
        return 1
    else

set @cur_vt =CURSOR
KEYSET
FOR
select TenVTu, SUM(SLXuat) as ts1
from tblVATTU vt inner join tblCTPXuat ct on ct.MaVTu =vt.MaVTu
            inner join tblPXuat px on px.SoPX=ct.SoPX
            where convert(char(7), Ngayxuat, 21)=@snt
            group by TenVTu

```

```

OPEN @CUR_VT
-----LOI GOI
declare @cur_dsvt CURSOR
declare @stvt nvarchar(20), @sts1 int,@sb int
EXec @sb=tt3 '2013-04', @cur_dsvt output
if @sb=1
    print 'khong co'
else
begin
while 0=0
begin
    fetch next from @cur_dsvt into @stvt, @sts1
    if @@FETCH_STATUS <>0 break
    print @stvt+cast(@sts1 as char(10))
end
close @cur_dsvt
deallocate @cur_dsvt

```

end

Bài tập thủ tục CURSOR:

- Viết thủ tục CURSOR cho ra tên vật tư, đơn giá, số lượng đặt của 1 số đặt hàng, với đầu vào là SoDH đó
- Viết thủ tục CURSOR cho ra các tên vật tư, tổng tiền của 1 số đặt hàng, với đầu vào là SoDH đó
- Viết thủ tục CURSOR cho ra các tên vật tư, , tổng tiền của 1 nhà cung cấp với đầu vào là họ tên nhà cung cấp đó

+ Ví dụ 6: giả sử có table sau:

Tblcanbo (MãCB, MãDV, TênCB, Giới tính, Sónămcôngtác, Địa chỉ, Hesólương, HesóPC, Lương).

Yêu cầu:

- MãCB : không trùng
- MãDV: phải có trong TblDonvi
- Giới tính: chỉ nhận 0 hoặc 1
- Hệ số lương : between 0 and 8
- Hệ số PC: between 0 and HSLuong
- Lương = (HSPC + HSL) * 730

```

alter proc tt_cn
    @smacb  char(4),  @sMadv  char(4),  @sTencb  nvarchar(20),
@sGioitinh int,
    @sSnct int, @sDiachi nvarchar(20), @sHesoluong float, @sHesoPC
float
as
--Kiểm tra mãcb

```

```
if exists(select macb from tblcanbo where macb = @smacb)
begin
print 'da co ma nay'
return
end
```

```
--Kiểm tra Mã DV
if not exists (select madv from tbldonvi where
    madv =@smadv)
begin
print 'Chua có madv nay'
return
end
```

```
--Kiểm tra gioitinh
if (@sgioitinh < 0) or (@sgioitinh>1)
begin
print 'gioi tinh chi nhan 0 hoac 1'
Return
end
```

```
--Kiểm tra Hệ số lương
If @sHesoluong not between 0 and 8
begin
Print 'He so luong nam ngoai'
Return
end
```

```
--Kiểm tra Hệ số PC
if not ((@sHesoPC >0) and (@sHesoPC <@sHesoluong))
begin
print ' Heso PC nam ngoai'
return
end
```

```
--Tính luong
declare @sLuong int
Set @sluong =(@shesoPC +@sHesoluong)*730
```

```
--Thêm vào bảng
Insert into tblcanbo
Values
```

(@smacb, @stencb,@sDiachi,@smadv,@sHesoluong, @sgioitinh,
@sSnct, @sHesopc, @sluong)

--Gọi thủ tục

Exec tt_cn 'CB11','DV01','Le Thi C',0,13,'Hung phúc',3.1,5

Bài tập thủ tục cập nhật:

1. Viết thủ tục cập nhật dữ liệu cho bảng tblCTDonDH, với yêu cầu kiểm tra khóa chính, khóa ngoại, miền giá trị: $10 \leq SLDat \leq 500$
2. Viết thủ tục cập nhật dữ liệu cho bảng tblPNhap, với yêu cầu kiểm tra khóa chính, khóa ngoại, miền giá trị: NgayNhap phải sau NgayDH ở bảng tblDonDH của SoDH này.
3. Viết thủ tục cập nhật dữ liệu cho bảng tblCTPNhap, với yêu cầu kiểm tra khóa chính, khóa ngoại, miền giá trị: SLNhap phải $\leq SLDat -$ tổng số lượng đã nhập của các phiếu nhập có cùng số đặt hàng này.

+ Ví dụ 7: Xây dựng thủ tục hiện thị dữ liệu cho báo cáo đơn đặt hàng. Tham số vào soDH, nếu gọi thủ tục mà không truyền tham số thì xem như hiện thị toàn bộ các Sodh có trong bảng tbldondh.

```
create proc tt_bc @sSohd char(4) = NULL
as
If @sSohd is Null
    SELECT dh.sodh, tenvtu, sldat, Tennhacc
    FROM tbldondh dh
        INNER JOIN tbldondh ct ON dh.sodh = ct.sodh
        INNER JOIN tblnhacc nc ON dh.manhacc=nc.manhacc
        INNER JOIN tbhvattu vt ON ct.mavtu = vt.mavtu
    ORDER BY dh.sodh, vt.tenvtu
else
    SELECT dh.sodh, tenvtu, sldat, Tennhacc
    FROM tbldondh dh
        INNER JOIN tbldondh ct ON dh.sodh = ct.sodh
        INNER JOIN tblnhacc nc ON dh.manhacc=nc.manhacc
        INNER JOIN tbhvattu vt ON ct.mavtu = vt.mavtu
        Where dh.sodh = @sSohd
    ORDER BY vt.tenvtu
```

Bài tập thủ tục hiện thị:

1. Xây dựng thủ tục hiện thị dữ liệu cho báo cáo xuất hàng: tên vật tư, số lượng xuất, đơn giá, tổng tiền. Tham số vào soPX, nếu gọi thủ tục mà không truyền tham số thì xem như hiện thị toàn bộ các SoPX có trong bảng tblpxuat.
2. Xây dựng thủ tục hiện thị dữ liệu cho báo cáo gồm Họ tên nhàn cung cấp, tên vật tư, số lượng đặt, tổng tiền. Tham số vào tên nhà cung cấp, nếu gọi thủ tục mà không truyền tham số thì xem như hiện thị toàn bộ các nhà cc.

10. Hàm:

+ Ví dụ 1: Xây dựng hàm chuỗi chuẩn:

```
Create function Chuanchuoi(@Chuoi varchar(50))
Returns varchar(50)
AS
Begin
declare @i int
while Charindex(' ',@Chuoi)>0
    set @Chuoi=replace(@chuoi,' ',' ')
set @Chuoi=Ltrim(Rtrim(@Chuoi))
set @Chuoi=Lower(@Chuoi)
set @Chuoi=' '+@Chuoi
set @i=charIndex(' ',@Chuoi)
While @i<>0
Begin
    set @Chuoi=STUFF(@Chuoi,@i+1,1,
                      Upper(substring(@Chuoi,@i+1,1)))
    set @i=charIndex(' ',@Chuoi,@i+1)
end
set @Chuoi=Stuff(@Chuoi,1,1,null)

Return(@Chuoi)
End
```

+ Ví dụ 2: viết hàm hiển thị các nhà cung cấp đã cc hàng hoá A

```
create function hh_4 (@sTenvtu nvarchar(30) )
returns Table
as
return
(select distinct TenNhacc, DiaChi, DienThoai
```

```

from tblNhaCC nc inner join tblDonDH dh on
dh.Manhacc=nc.Manhacc
    inner join tblPNhap pn on pn.SoDH =dh.SoDH
    inner join tblCTPNhap ct on ct.SoPN=pn.SoPN
    inner join tblVATTU vt on vt.MaVTu =ct.MaVTu
where TenVTu =@sTenvtu)
-----
select * from dbo.hh_4(N'Xi măng')
select * from dbo.hh_4(N'Gạch')

```

■ Bài tập:

1. Viết hàm tham số vào là mã hàng hoá trả về đơn giá của hàng hoá này.
2. Viết hàm tham số vào là tên hàng hoá trả về đơn giá của hàng hoá này.
3. Viết hàm tham số vào là nhà cung cấp trả về tổng tiền phải trả cho nhà cung cấp này.
4. Viết hàm tham số vào là tên nhà cung cấp và tên hàng hoá trả về tổng tiền phải trả cho nhà cung cấp này với tên hàng hoá trên.
5. Viết hàm tham số vào số đặt hàng trong tbldondh trả về tổng tiền của số đặt hàng này.
6. Viết hàm tham số vào là tên vật tư trả về tổng số lượng nhập hàng của vật tư này.
7. Viết hàm tham số vào là tên vật tư trả về tổng số lượng xuất hàng của vật tư này.
8. Viết hàm tham số vào là số phiếu xuất trả về tổng tiền của nó.

■ Bài tập:

QLBH:

1. Tạo hàm tham số vào là nhacc, ra là tên các hàng hoá đã cung cấp
2. Tạo hàm hiển thị các tên hàng hoá, số lượng đặt mua, đơn giá của số đặt hàng s.
3. Tạo hàm tham số vào là sopn, ra là mavtu, tenvtu, slnhap, dongia, tổng tiền của Sopn trên.
4. Tạo hàm tham số vào là sopx, ra là mavtu, tenvtu, slnhap, dongia, tổng tiền của Sopx trên.
5. Tạo hàm tham số vào là nhacc, ra là mã vật tư, tên vật tư, tổng số lượng nhập, đơn giá, tổng tiền do nhà cung cấp trên đã cung cấp.
6. Tạo hàm tham số vào là tên vật tư, ra là mã nhà cung cấp, tên nhà cung cấp, tổng số lượng nhập, đơn giá, tổng tiền của vật tư trên.

QLCB:

1. Tạo hàm hiển thị tên cán bộ, địa chỉ, số điện thoại của cán bộ có đơn vị công tác A

2. Tạo hàm hiển thị đơn vị có tổng lương nhận >S
3. Tạo hàm hiển thị tổng lương của cán bộ nữ của từng đơn vị

+ Ví dụ 3: Tạo hàm đầu vào là sodh, ra là table chứa tên hàng hoá, số lượng đặt, đơn giá, tổng tiền của sodh này

Alter function h_sodh(@ssodh char(4))

Returns @hh table

```
(tenvtu nvarchar(30),
dongia int,
sldat int,
tongtien int)
```

As

begin

Insert into @hh (tenvtu, dongia, sldat)

Select tenvtu, dongia, sldat

From tbldondh dh

Inner join tblctdondh ct on dh.sodh=ct.sodh

inner join tblvattu vt on ct.mavtu=vt.mavtu

Where (dh.sodh=@ssodh)

Update @hh

Set tongtien=dongia*sldat

Return

end

--L?i g?i

Select * from h_sodh(4)

■ Bài tập:

1. QLCB: Tạo hàm tạo table gồm tên đơn vị, tổng lương đơn vị, BHXH=5%, BHYT=1%, tổng lương của đơn vị đó.
2. QLHS: Tạo hàm tạo table gồm tên hs, điểm các môn, tổng điểm, xếp loại cho hs dựa trên tổng điểm.
3. QLBH: Tạo hàm tạo table gồm: tên nhà cc, tổng tiền, giảm giá 10% nếu tổng tiền>A, nếu từ A đến B giảm 5%, ít hơn B giảm =0, Cột tiền phải trả=tổng tiền-giảm.

11. TRIGGER

- Ví dụ: Xây dựng trigger trong tblpnhap để kiểm tra khi người dùng thêm mới mẫu tin.

- Khóa ngoại: sodh phải tồn tại trong tbldondh, manhacc phải tồn tại trong tblnhacc.
- Miền giá trị: ngày nhập phải sau ngaydh trong bảng tbldondh

+ Ví dụ 2: Xóa 1 Sodh trong tbldondh. Nếu sodh này đã có trong bảng tblpnhap thì không xóa được. Nếu chưa có thì xóa nó cùng bản ghi có sodh đó trong tblctdondh.

--L?i g?i:

Alter table tên_table DISABLE TRIGGER tên_trigger

alter table tblCTDONDH nocheck constraint all

+Ví dụ 3: Sửa thông tin bảng tbldondh thỏa mãn:

- Không cho phép sửa dữ liệu cột sodh
- Sửa manhacc phải tồn tại trong tblnhacc
- Cột ngaydh phải trước ngày nhập hàng trong tblpnhap.

Bài tập 1: Xây dựng trigger trong tblctdondh để kiểm tra khi người dùng thêm mới mẫu tin.

- Khóa ngoại: sodh phải tồn tại trong tbldondh, mavtu phải tồn tại trong tblvattu.
- Miền giá trị: sldat >=A và <=B.

Bài tập 2: Xây dựng trigger trong tblpnhap để kiểm tra khi người dùng thêm mới mẫu tin.

- Khóa ngoại: sodh phải tồn tại trong tbldondh.
- Miền giá trị: ngày nhập hàng phải sau ngày đặt trong bảng tbldondh.

Bài tập 3: Xây dựng trigger trong tblctpnhap để kiểm tra khi người dùng thêm mới mẫu tin.

- Khóa ngoại: sopnhap phải tồn tại trong bảng tblpnhap, mavtu phải tồn tại trong tblvattu.
- Miền giá trị: slnhap phải thỏa mãn: tổng số lượng nhập của các vật tư này của các số phiếu nhập có cùng số đặt hàng phải \leq số lượng đặt hàng của vật tư trong số đặt hàng đó.

Bài tập 4: Xây dựng trigger trong tblctpxuat để kiểm tra khi người dùng thêm mới mẫu tin.

- Khóa ngoại: sopx phải tồn tại trong tblpxuat, mavtu phải tồn tại trong tblvattu.
- Miền giá trị: slxuat >=A và <=B.

Đề 1. Tạo CSDL QLBH với các bảng sau:

tblDMSANPHAM (**MaDM** char(4), TenDanhMuc nvarchar(30), MoTa nvarchar(50))

tblSANPHAM (**MaSP** char(4), MaDM char(4), TenSP nvarchar(30), GiaTien money)

tblDONHANG (**MaDH** char(4), **MaSP** char(4), SoLuong int)

1. Tạo kết nối, nhập dữ liệu vào bảng (bảng 3 có ít nhất 5 bản ghi)
2. Dùng biến con trả CURSOR hiện thị cho xem Tên sản phẩm, Tên danh mục, số lượng, Thành tiền của Mã đơn hàng A.
3. Tạo thủ tục cho xem Tổng tiền của MaDH là A, với tham số đầu vào là A đó. Nếu A không tồn tại, cho dòng thông báo
4. Tạo thủ tục hiện TenSP có tổng tiền thu về lớn nhất
5. Viết thủ tục tạo MADH tự động....
6. Tạo thủ tục đầu vào là Tên Danh mục, ra là Tên sản phẩm, Giá tiền, tổng số lượng, Tổng tiền của các mặt hàng thuộc danh mục này.
7. Viết thủ tục xóa TenSP có trong bảng tblSANPHAM và các bảng liên quan với **TenSP là tham số đầu vào của thủ tục**
8. Viết thủ tục CURSOR tìm TenSP có đơn giá >A (với A là tham số đầu vào), để xóa tất cả các bản ghi có tên TenSP đó trong bảng tblSANPHAM và các bảng liên quan
9. Tạo thủ tục CURSOR với đầu vào là Mã đơn hàng, ra Tên sản phẩm, Tên danh mục,số lượng, Thành tiền của Mã đơn hàng đó.
10. Tạo thủ tục cập nhật dữ liệu cho bảng tblDONHANG với yêu cầu kiểm tra tính hợp lệ: không trùng khóa chính và kiểm tra dữ liệu khóa ngoại.
11. Viết thủ tục hiện thị dữ liệu với đầu vào là Mã đơn hàng, ra là Tên sản phẩm, Tên danh mục,số lượng, Thành tiền của Mã đơn hàng này, nếu gọi mà không truyền tham số thì hiện thị tất cả nội dung các đơn hàng.
12. Viết hàm cho xem Tổng tiền của MaDH là A, với tham số đầu vào là A đó
13. Viết hàm tạo bảng Inline Table tạo bảng gồm Tên sản phẩm, Tên danh mục,số lượng, Thành tiền của Mã đơn hàng A
14. Viết hàm tạo MADH tự động....
15. Viết hàm tạo bảng Multi Statement tạo bảng gồm Tên sản phẩm, Tên danh mục,số lượng, Thành tiền của Mã đơn hàng A
16. Xây dựng trigger trong tbldondh để kiểm tra khi người dùng thêm mới mẫu tin.
 - Khóa ngoại: mas phải tồn tại trong tblsanpham.
 - Miền giá trị: SoLuong phải $\geq A$ và $\leq B$.

Câu 16: Viết trigger để khi xóa một bản ghi trong bảng tblDMSANPHAM thì phải xóa luôn MaDM đó có trong bảng tblSANPHAM.

Câu 17: Viết trigger để khi thêm một bản ghi mới vào bảng tblSANPHAM thì thỏa các yêu cầu sau:

- MaSP phải bắt đầu bằng 2 ký tự ‘SP’.
- $0 < \text{GiaTien} < 100$

Câu 18: Viết trigger để khi sửa một bản ghi trong bảng tblDONHANG thì thỏa các yêu cầu sau:

- Không cho phép sửa cột MaDH.
- $20 \leq \text{SoLuong} \leq 50$

Câu 19: Viết trigger để khi xóa một bản ghi trong bảng tblSANPHAM thỏa mãn: nếu MaSP đó đã có trong bảng tblDONHANG thì không được xóa.

Câu 20:- Hãy thêm cột tổng tiền kiểu money cho bảng tblDONHANG.

Sau đó viết trigger để sửa một bản ghi trong bảng tblDONHANG thì thỏa các yêu cầu sau:

- Không được sửa cột **MaSP**.
 - Nếu sửa cột SoLuong thì phải sửa thêm cột TongTien = SoLuong * GiaTien. (GiaTien trong bảng tblSANPHAM)

```
alter trigger them_dondh
on tbldondh
INSTEAD OF insert
as
-----kiem tra khoa chinh (neu loi thi RETURN)
if exists (select * from tb1DonDH,inserted
           where
tbldondh.SoDH=inserted.SoDH)
begin
raiserror ('TBL: trung khoa chinh',16,1)
rollback tran
RETURN
end
-----kiem tra khoa ngoai
if NOT exists (select * from tblnhac,inserted
               where
tblnhac.Manhacc=inserted.Manhacc)
begin
```

```
raiserror ('TBL: khong co khoa ngoai',16,1)
rollback tran
RETURN
end
-----kiem tra mien gia tri
if (select NgayDH from inserted)< getDATE()
begin
print 'TBL: MIEN GIA TRI'
rollback tran
end
-----loi goi
alter table tblDONDH nocheck constraint all
insert into tblDonDH
values (100,'2018-09-09','cc03')
```

Cho cơ sở dữ liệu **THUCTAP** gồm các quan hệ sau:

tblSV (**MaSV**, TenSV, QueQuan, NamSinh, DiemRL)

tblDETAI (**MaDT**, TenDT, KinhPhi)

tblDIEM (**MaSV**, **MaDT**, Diem)

Trong đó: DiemRL: Điểm rèn luyện

1. Viết hàm tạo bảng gồm các thông tin MaSV, TenSV, QueQuan, DiemRL thực hiện câu lệnh cập nhật: DiemRL = DiemRL+10
2. Sử dụng biến kiểu dữ liệu CURSOR in danh sách gồm MaSV, TenSV, QueQuan, TenDT của những sinh viên sinh trước 1980
3. Viết thủ tục nhận vào là D, kết quả in ra là danh sách các thông tin MaSV, TenSV, QueQuan, NamSinh, TenDT, Diem của những sinh viên có Diem>= D
4. Tạo thủ tục cho xem Diem của Tên sinh viên là A, với tham số đầu vào là A đó. Nếu A không tồn tại, cho dòng thông báo
5. Viết thủ tục tạo MASV tự động....
6. Viết thủ tục xóa TenSV có trong bảng tblSV và các bảng liên quan với TenSV là tham số đầu vào của thủ tục
7. Viết thủ tục CURSOR tìm TenDT có KinhPhi >A (với A là tham số đầu vào), để xóa tất cả các bản ghi có tên TenDT đó trong bảng tblDETAI và các bảng liên quan
8. Tạo thủ tục cập nhật dữ liệu cho bảng tblDiem với yêu cầu kiểm tra tính hợp lệ: không trùng khóa chính và kiểm tra dữ liệu khóa ngoại.
9. Viết thủ tục hiện thị dữ liệu với đầu vào là Mã sinh viên, ra là Tên sinh viên, MaSV, TenSV, QueQuan, TenDT, Diem của Mã sinh viên này, nếu gọi mà không truyền tham số thì hiện thị tất cả nội dung các đơn hàng.
10. Tạo thủ tục CURSOR với đầu vào là Mã sinh viên, ra Tên sinh viên, MaSV, TenSV, QueQuan, TenDT, Diem của Mã sinh viên này.
11. Viết hàm cho xem KinhPhi của MaDT là A, với tham số đầu vào là A đó
12. Viết hàm tạo bảng Inline Table tạo bảng gồm Tên sinh viên, MaSV, TenSV, QueQuan, TenDT, Diem của Mã sinh viên A với A tham số đầu vào
13. Viết hàm tạo MADT tự động....

Câu 16: Viết trigger để khi xóa một bản ghi trong bảng tblSV thì phải xóa luôn MaSV đó có trong bảng tblDIEM

Câu 17: Viết trigger để khi thêm một bản ghi mới vào bảng tblSV thì thỏa các yêu cầu sau:

- MaSV phải bắt đầu bằng 2 ký tự 'SV'.
- $0 \leq \text{DiemRL} \leq 10$

Câu 18: Viết trigger để khi sửa một bản ghi trong bảng tblDETAI thì thỏa các yêu cầu sau:

- Không cho phép sửa cột MaDT.
- $20 \leq \text{KinhPhi} \leq 50$

Câu 19: Viết trigger để khi xóa một bản ghi trong bảng tblSV thỏa mãn: nếu MaSV đó đã có trong bảng tblDIEM thì không được xóa.