

HÀM ĐIỀU KIỆN – VÒNG LẶP – QUẢN LÝ LỖI

I. CẤU TRÚC IF - ELSE

1. Cú pháp:

a. Dùng hàm của hệ thống (gắn trong mệnh đề SELECT):

IF(điều kiện, thực hiện khi đk đúng, thực hiện khi đk sai)

b. Theo cấu trúc IF-ELSE (dùng để kiểm tra sự tồn tại của đối tượng)

IF biểu thức điều kiện

{câu lệnh}

ELSE

{câu lệnh}

Lưu ý: nếu thực thi hai hay nhiều câu lệnh trong IF hoặc ELSE thì phải có BEGIN – END cho khối lệnh đó.

I. CẤU TRÚC IF - ELSE

b. Theo cấu trúc IF-ELSE (dùng để kiểm tra sự tồn tại của đối tượng)

if exists (câu truy vấn chứa điều kiện)

{câu lệnh}

ELSE

{câu lệnh}

Lưu ý: nếu thực thi hai hay nhiều câu lệnh trong IF hoặc ELSE thì phải có BEGIN – END cho khối lệnh đó.

I. CẤU TRÚC IF - ELSE

Ví dụ:

Tạo query cho xem kết quả thi của sinh viên, thông tin gồm các field: MaSV, HoTen, TenHP, DiemLan1, KetQua, trong đó: field KetQua được xét như sau: **nếu DiemLan1 >= 5 thì đạt, ngược lại thì không đạt.**

```
SELECT SV.MASV, HODEM + ' ' + TEN AS HOTENSV, TENHP,  
DIEMLAN1,  
IIF(DIEMLAN1 >= 5, N'ĐẠT', N'KHÔNG ĐẠT') AS KETQUA  
FROM SINHVIEN SV, HOCPHAN HP, KETQUA KQ  
WHERE SV.MASV = KQ.MASV AND HP.MAHP = KQ.MAHP
```

I. CẤU TRÚC IF - ELSE

Ví dụ: sử dụng csdl QLDA:

1. In ra lương trung bình của công ty.
2. Kiểm tra lương nhân viên, nếu có nhân viên có lương nhỏ hơn lương trung bình của công ty thì ghi là “có nhân viên lương nhỏ hơn lương trung bình công ty” và cho biết mã phòng ban và mã nhân viên đó; ngược lại ghi là “không có nhân viên nào có mức lương nhỏ hơn lương trung bình của công ty”.

I. CẤU TRÚC IF - ELSE

Ví dụ: sử dụng csdl QLDA:

1. In ra lương trung bình của công ty.

```
select avg(luong)  
from NHANVIEN
```

2.

```
declare @ltb float
select @ltb = avg(luong)
from NHANVIEN
print @ltb
if exists (select * from NHANVIEN where luong < @ltb)
begin
    print ('có nhân viên lương nhỏ hơn lương trung bình của công ty')
    select phg, manv, tennv, luong
    from NHANVIEN
    where luong < @ltb
end
Else print 'không có nhân viên lương nhỏ hơn lương trung bình của công ty'
```

I. CẤU TRÚC IF - ELSE

Bài tập sử dụng csdl QLDA:

1. In ra họ tên nhân viên, lương, tình trạng. Trong đó, nếu phái là Nam thì ghi thêm “Mr.” trước tên nhân viên, ngược lại thêm “Ms.” trước tên nhân viên. Nếu lương nhỏ hơn lương Trung bình công ty thì tình trạng ghi là “Tăng lương”, ngược lại ghi là “Không tăng lương”.
2. Cho biết mã phòng, số lượng nhân viên của từng phòng, tình trạng. Trong đó, nếu số lượng nhân viên nhỏ hơn 4 thì ghi là “thiếu nhân viên”, bằng 4 thì ghi là “đủ nhân viên”, còn lại ghi là “thừa nhân viên”.

II. CẤU TRÚC CASE

- Case kiểm tra theo danh sách điều kiện được đưa ra, và trả về kết quả tương ứng với từng điều kiện đó.
- Case được sử dụng khi có nhiều trường hợp xảy ra tương ứng với điều kiện nhất định.

II. CẤU TRÚC CASE

- Cú pháp:

Case <biểu thức>

When <biểu thức 1> then < câu lệnh 1>

When <biểu thức 2> then < câu lệnh 2>

...

When <biểu thức n - 1> then < câu lệnh n-1>

Else <thức thức n>

End AS [biểu thức nhận kết quả trả về]

II. CẤU TRÚC CASE

- Ví dụ: sử dụng CSDL QL_sinhvien_BTH1

Tạo query cho xem điểm trung bình của từng sinh viên, thông tin gồm các Field: MãSV, HoTen, DiemTB, XepLoai. Trong đó:

DiemTB=
Round(Avg(IIF([diemlan1]>ISNULL([diemlan2],0),[diemlan1],[diemlan2])),
1);

XepLoai: Nếu DiemTB >=8, xếp loại Giỏi,
8> DiemTB >=6.5, xếp loại Khá,
6.5>DiemTB>=5, xếp loại Trung bình,
Còn lại là loại Yếu.

II. CẤU TRÚC CASE

```
SELECT SV.MASV, HODEM + ' ' + TEN AS HOTENSV,  
ROUND(AVG(IIF(DIEMLAN1 > ISNULL(DIEMLAN2,0),DIEMLAN1,DIEMLAN2)),1)  
AS DIEMTB,  
CASE  
    WHEN ROUND(AVG(IIF(DIEMLAN1 >  
        ISNULL(DIEMLAN2,0),DIEMLAN1,DIEMLAN2)),1) >= 8 THEN N'GIỎI'  
    WHEN ROUND(AVG(IIF(DIEMLAN1 >  
        ISNULL(DIEMLAN2,0),DIEMLAN1,DIEMLAN2)),1) >= 6.5 THEN N'KHÁ'  
    WHEN ROUND(AVG(IIF(DIEMLAN1 >  
        ISNULL(DIEMLAN2,0),DIEMLAN1,DIEMLAN2)),1) >= 5 THEN N'TRUNG BÌNH'  
    ELSE N'YẾU'  
END AS XEPLAI  
FROM SINHVIEN SV, KETQUA KQ  
WHERE SV.MASV = KQ.MASV  
GROUP BY SV.MASV, HODEM + ' ' + TEN
```

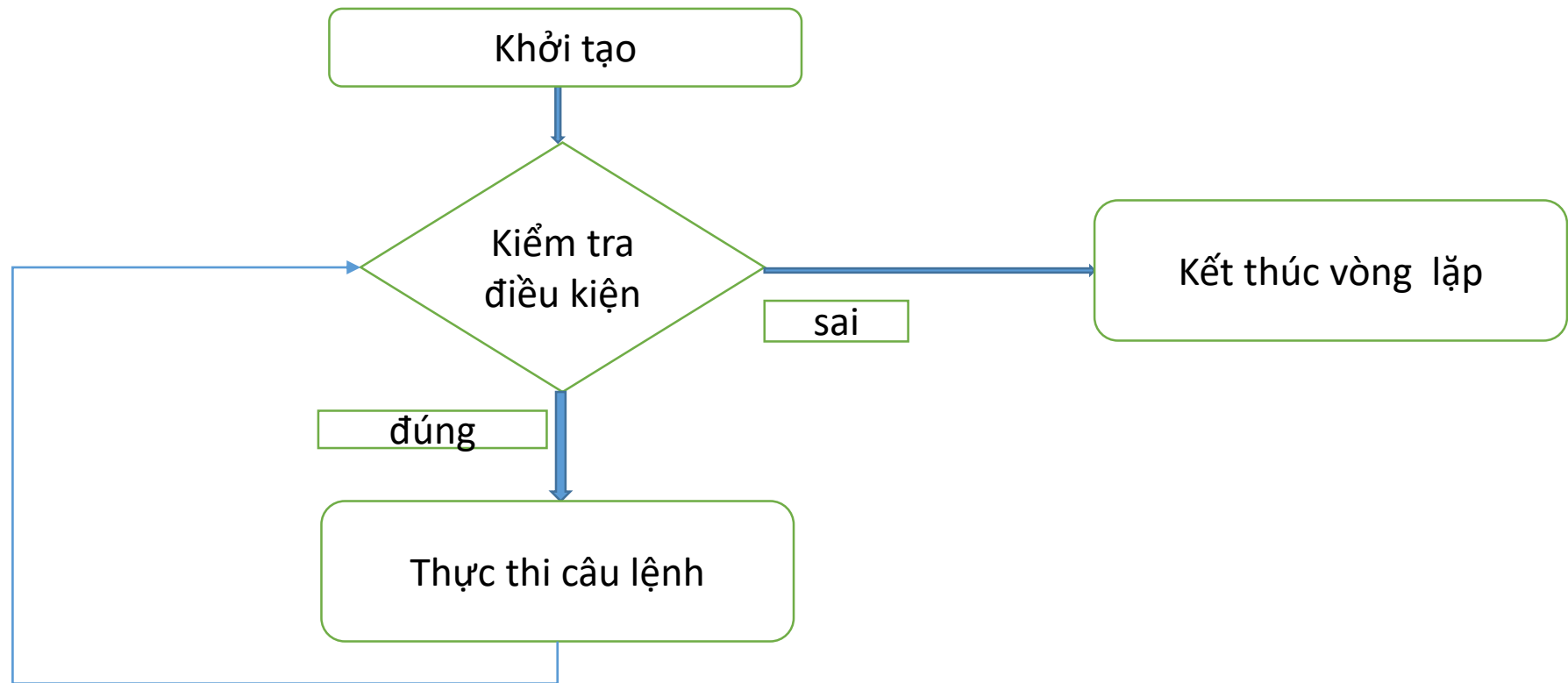
II. CẤU TRÚC CASE

- Bài tập: sử dụng CSDL QLDA

1. Cho biết họ tên, manv, ngaysinh, độ tuổi. Trong đó, độ tuổi được xác định như sau Tuổi từ 18 đến < 35 : Thanh niên; Tuổi từ 35 đến < 50 : Trung niên; Tuổi ≥ 50 : Cao niên.
2. Cho biết họ tên (thêm Mr hoặc Ms trước tên dựa vào phái: phái là Nam thì thêm Mr, ngược lại là Ms), lương, thuế. Trong đó nếu lương < 20000 thì thuế là $5\% * \text{lương}$, nếu lương từ 20000 tới dưới 35000 thì thuế là $10\% * \text{lương}$, nếu lương từ 35000 tới dưới 45000 thì thuế là $15\% * \text{lương}$, còn lại thuế là $20\% * \text{lương}$.

III. VÒNG LẶP WHILE

1. Vòng lặp được sử dụng khi muốn thực thi một khối lệnh/lệnh khi điều kiện vẫn còn đúng.



III. VÒNG LẶP WHILE

1. Cú pháp:

WHILE <biểu thức điều kiện >

{câu lệnh} hoặc

BEGIN

khối lệnh

END

[BREAK]

[CONTINUE]

III. VÒNG LẶP WHILE

Ví dụ: đoạn code sau dùng để làm gì?

```
DECLARE @dem int
DECLARE @sum int
SET @dem = 5
SET @sum = 0
WHILE @dem > 0
BEGIN
    SET @sum = @sum + @dem
    SET @dem = @dem - 1
END
PRINT @SUM
```


III. VÒNG LẶP WHILE

Bài tập:

1. Viết chương trình tính tổng các số từ 0 tới 10.
2. Viết chương trình tính tổng các số chẵn từ 0 tới 10.
3. Viết chương trình tính tổng các số chẵn (trừ số 4) từ 0 tới 10.

III. VÒNG LẶP WHILE

```
DECLARE @dem int
DECLARE @sum int
SET @dem = 10
SET @sum = 0
WHILE @dem > 0
    BEGIN
        SET @sum = @sum + @dem
        SET @dem = @dem - 1
    END
PRINT @SUM
```

III. VÒNG LẶP WHILE

```
DECLARE @dem int = 1
DECLARE @sum int = 0
WHILE @dem <= 10
BEGIN
    IF (@dem % 2 = 0)
    BEGIN
        SET @sum = @sum + @dem
    END
    --else continue
    SET @dem = @dem + 1
END
PRINT @SUM
```

III. VÒNG LẶP WHILE

```
declare @num int = 0
declare @sum int = 0
while @num <= 10
begin
    if @num = 4
        begin
            set @num = @num + 1
            continue
        end
    if @num % 2 = 0
        begin
            set @sum = @sum + @num
        end
    set @num = @num + 1
end
print ('tổng các số chẵn từ 0 - 10 (trừ số 4) là:' + cast(@sum as varchar))
```

III. VÒNG LẶP WHILE

Bài tập: sử dụng CSDL QLDA

1. Viết chương trình in ra thông tin nhân viên có mã số là chẵn.
2. Viết chương trình in ra thông tin nhân viên có năm sinh là năm lẻ.

IV. XỬ LÝ LỖI TRY ... CATCH

Thực hiện các lệnh trong TRY, nếu có lỗi thì sẽ chuyển qua xử lý các câu lệnh trong CATCH.

BEGIN TRY

{câu lệnh SQL | khối câu lệnh }

END TRY

BEGIN CATCH

{câu lệnh SQL | khối câu lệnh }

END CATCH

Lưu ý:

Try, Catch cùng logic xử lý

Sau Try là Catch

Có thể lồng nhiều cấp với nhau

IV. XỬ LÝ LỖI TRY ... CATCH

Một số hàm ERROR thường dùng trong khối CATCH

ERROR_NUMBER(): trả về mã số của lỗi

ERROR_MESSAGE(): trả về chuỗi lỗi

ERROR_SEVERITY(): cho biết mức độ nghiêm trọng của lỗi

ERROR_LINE(): trả về dòng gây lỗi

ERROR_PROCEDURE(): trả về thủ tục hay trigger gây ra lỗi.

IV. XỬ LÝ LỖI TRY ... CATCH

Ví dụ:

```
begin try
```

```
    select 'today is:' + getdate() as ngay
```

```
end try
```

```
begin catch
```

```
    select convert(varchar,error_number(),1) as maloi,
```

```
    error_message() as message,
```

```
    error_severity() as mucdonghiemtrong
```

```
end catch
```


IV. XỬ LÝ LỖI TRY ... CATCH

Bài tập: Sử dụng CSDL BTH1

1. Chèn thêm dòng dữ liệu vào bảng NHANVIEN. Nếu dữ liệu chèn thành công thì xuất hiện thông báo đã chèn thành công, ngược lại xuất hiện thông báo dữ liệu chèn không thành công.
2. Viết câu lệnh thêm nhân viên "Hứa Quảng Hà" vào dự án "Hệ thống xác thực vân tay" vào ngày "5/5/2006".