# MỤC TIÊU

* Mục tiêu của buổi học hôm nay chúng ta sẽ đi tìm hiểu về vòng lặp WHILE và cách quản lý lỗi trong SQL Server.

# VÒNG LẶP

* Vòng lặp được sử dụng nếu muốn chạy lặp đi lặp lại một đoạn mã khi điều kiện cho trước trả về giá trị là TRUE.
* Từ những ngày đầu học lập trình chúng ta đã quá quen thuộc với Vòng lặp FOR thường được sử dụng để chạy lặp đi lặp lại một đoạn mã cho trường hợp biết trước số lần lặp lại. Tuy nhiên trong[SQL Server](https://quantrimang.com/sql-server) không tồn tại vòng lặp FOR. Mặc dù vậy, ta vẫn có thể sử dụng vòng lặp WHILE để mô phỏng FOR.
* Đầu tiên ta sẽ khởi tạo biến đếm.
* Tiếp theo chúng ta sẽ đi xét biểu thức điều kiện. Nếu dk đúng ta thực thi các câu lệnh, sau đó quay lại kiểm tra lại điều kiện vẫn còn đúng hay không, nếu đk vẫn đúng ta thực thi lại các câu lệnh, nếu đk đã sai, ta thoát khỏi vòng lặp while.

# SLIDE 21: CÚ PHÁP WHILE

* Cú pháp của WHILE như sau:

**WHILE bieuthucdk**

**LỆNH|[BEGIN Khối lệnh END]**

**[BREAK]**

**[CONTINUE]**

* Sử dụng câu lệnh vòng lặp WHILE khi bạn không chắc chắn về số lần bạn muốn thực thi.
* Vì điều kiện WHILE được đánh giá trước khi vào vòng lặp nên vòng lặp có thể không thực hiện được lần nào (khi dieukien là FALSE thì vòng lặp sẽ kết thúc ngay lập tức).
* Ta xét vd sau:
* Ta khai báo biến đếm là số nguyên và gán giá trị =0
* Ta sử dụng vòng lặp WHILE xét biểu thức đk: biến dem <5 hay không
* Ở vòng lặp 1: Nếu đk đúng ta thực thi khối lệnh trong BEGIN..END:
* In ra màn hình: Quan trọng là phương pháp học
* Thực hiện gán giá trị mới cho biến dem=dem+1
* Sau khi thực thi khối lệnh, ta quay lại xét tiếp đk trong WHILE ở vòng lặp tiếp theo
* Ta thực hiện cho đến khi biểu thức điểu kiện trong WHILE sai thì thoát khỏi vòng lặp WHILE.
* Thực hiện câu lệnh tiếp theo, in ra màn hình ‘Học lập trình thì ra cũng dễ’
* khi biến đếm bằng 5 (biểu thức đk 5<5=>sai) thì ta dừng vòng lặp WHILEmà biến dem bắt đầu =0, và sau mỗi vòng lặp sẽ tăng biến đếm lên 1dv, vậy tổng cộng ta thực thi 5 vòng lặp, in ra màn hình câu Quan trọng là phương pháp học 5 lần.

# SLIDE 22: LỆNH BREAK

* Khi ta có nhu cầu thoát khỏi vòng lặp WHILE sớm dù điều kiện trong WHILE vẫn còn thỏa, ta dùng lệnh Break (Ngắt điều khiển).
* Cú pháp BREAK;
* Lệnh BREAK Không có tham số và đối số nào nằm trong câu lệnh BREAK.
* Nếu trong đoạn code có WHILE lồng nhau, BREAK sẽ chấm dứt vòng lặp WHILE gần nhất.
* Ta xét vd:
* Trong ví dụ này, vòng lặp sẽ không thực hiện lần nào nếu ngay từ đầu @Number > 10, nó chỉ thực hiện và duy trì khi biến < = 10. Đến khi vượt quá điều kiện (> 10), vòng lặp sẽ kết thúc và tiếp tục thực thi các câu lệnh tiếp theo.
* Ta khai báo và gán trị cho biến number=1, biến total=0
* Ta xét dk trong WHILE: biến number=1 <=10 => ĐÚNG ta thực thi lệnh trong khối BEGIN.. END
* VÒNG LẶP 1:
* Ta kiểm tra dk trong IF number=1 =5=>SAI, ta thực thi lệnh ở mệnh đề ELSE:
* Gán giá trị total=total+number (TOTAL=0+1=1)
* Gán giá trị NUMBER= number+1(1+1=2)
* Ta quay lại xét dK trong WHILE: NUMBER=2<=10
* Ta thực thi VÒNG LẶP 2:
* Ta kiểm tra dk trong IF number=2 =5=>dk SAI, ta thực thi lệnh ở mệnh đề ELSE:
* Gán giá trị total=total+number (TOTAL=1+2=3)
* Gán giá trị NUMBER= number+1(2+1=3)
* Ta quay lại xét dK trong WHILE: NUMBER=3<=10
* Ta thực thi VÒNG LẶP 3:
* Ta kiểm tra dk trong IF number=3 =5 =>dk SAI, ta thực thi lệnh ở mệnh đề ELSE:
* Gán giá trị total=total+number (TOTAL=3+3=6)
* Gán giá trị NUMBER= number+1(3+1=4)
* Ta quay lại xét dK trong WHILE: NUMBER=4<=10
* Ta thực thi VÒNG LẶP 4:
* Ta kiểm tra dk trong IF number=4=5 =>dk SAI, ta thực thi lệnh ở mệnh đề ELSE:
* Gán giá trị total=total+number (TOTAL=6+4=10)
* Gán giá trị NUMBER= number+1(4+1=5)
* Ta quay lại xét dK trong WHILE: NUMBER=5<=10
* Ta thực thi VÒNG LẶP 5:
* Ta kiểm tra dk trong IF number=5=5 =>dk ĐÚNG, ta thực thi lệnh ở mệnh đề IF, BREAK thoát khỏi vòng lặp
* Sau đó in ra thông báo giá trị của biến total.=10

# SLIDE 23: LỆNH CONTINUE

* Khi gặp lệnh CONTINUE ta sẽ bỏ qua không thực hiện các lệnh đi sau CONTINUE trong lần lặp hiện tại, mà sẽ đi đến lần lặp kế tiếp.
* Giống nhứ BREAK, CONTINUE không có đối số

# SLIDE 24 DEMO

--Viết chương trình tính tổng các số chẵn từ 1 tới 10.

DECLARE @Number INT = 1 ;

DECLARE @Total INT = 0 ;

WHILE @Number < = 10

BEGIN

IF @Number%2 =0 --%: CHIA LẤY DƯ

BEGIN

SET @Total = @Total + @Number;

SET @Number = @Number + 1 ;

END

ELSE

SET @Number = @Number + 1 ;

END

PRINT N'Tổng các số chẵn từ 1 đến 10 là: '+ CONVERT(NVARCHAR, @Total)

--Viết chương trình tính tổng các số chẵn từ 1 tới 10 nhưng bỏ số 4.

DECLARE @Number INT = 1 ;

DECLARE @Total INT = 0 ;

WHILE @Number < = 10

BEGIN

IF @Number=4

BEGIN

SET @Number = @Number + 1 ;

CONTINUE

END

ELSE IF @Number%2 =0 --%: CHIA LẤY DƯ

BEGIN

SET @Total = @Total + @Number;

SET @Number = @Number + 1 ;

END

ELSE

SET @Number = @Number + 1 ;

END

PRINT N'Tổng các số chẵn từ 1 đến 10 trừ số 4 là: '+ CONVERT(NVARCHAR, @Total)

# SLIDE 25: TRY CATCH

Trong một vài trường hợp, logic của câu truy vấn sẽ bị sai lệch, gây ảnh hưởng tính logic của tổng thể hệ thống. T-SQL cung cấp cho chúng ta một số câu lệnh để xử lý những ngoại lệ đó như TRY … CATCH hoặc tục RAISEERROR

Cú pháp *TRY – CATCH* cho phép chúng ta xử lý ngoại lệ một cách chủ động. Ý tưởng của phương pháp này rất đơn giản. Yêu cầu hệ thống thực hiện một công việc nào đó. Nếu một lỗi phát sinh, xử lý lỗi đó theo một phương pháp đã định trước.

* Cú pháp như sau:

**BEGIN TRY**

**{<câu lệnh SQL>|<Khối câu lệnh>}**

**END TRY**

**BEGIN CATCH**

**{<Câu lệnh SQL>|<Khối câu lệnh>}**

**END CATCH**

**https://hoadm.net/xu-ly-ngoai-le-trong-t-sql/**

* Thực hiện các lệnh trong khối TRY, nếu gặp lỗi sẽ chuyển qua xử lý bằng các lệnh trong khối CATCH
* Các điểm cần lưu ý
* TRY và CATCH phải cùng lô xử lý
* Sau khối TRY phải là khối CATCH
* Có thể lồng nhiều cấp

# SLIDE 26 MỘT SỐ HÀM THÔNG DỤNG BẮT LỖI

* Một số hàm ERROR thường dùng như:
* ERROR\_NUMBER(): trả vể mã số lỗi
* ERROR\_MESSAGE(): trả về thông báo lỗi với hình thức văn bản
* ERROR\_SEVERITY(): Trả về mức độ nghiêm trọng của lỗi
* ERROR\_STATE(): trả về trạng thái của lỗi (dạng số)
* ERROR\_LINE(): =trả vể vị trị dòng lệnh gây ra lỗi
* ERROR\_PROCEDURE(): trả về tên thủ tục hay tên trigger gây ra lỗi
* Đầu tiên, ta xét một ví dụ phát sinh lỗi đơn giản sau: **select 1+ 'SQL'**
* Ở đây 1 là kiểu số không thể ép kiểu tự động về kiểu chuỗi để cộng chuỗi được nên hệ thống báo lỗi.
* Ta sẽ đưa câu lệnh cần bắt lỗi vào khối TRY
* Để đọc thông tin vể lỗi, trong khối CATCH ta dùng lệnh SELECT lấy ra một số hàm ERROR như ERROR\_NUMBER(): hiển thị mã số lỗi, ERROR\_MESSAGE(): hiển thị thông báo lỗi dạng text.

**BEGIN TRY**

**select 1+ 'SQL'**

**END TRY**

**BEGIN CATCH**

**SELECT ERROR\_NUMBER(),ERROR\_MESSAGE()**

**END CATCH**

* Sau khi thực thi câu lệnh, ta thu được kết quả: …

# SLIDE 27 VD TRY CATCH

Ở csdl QLDA, khi ta INSERT dữ liệu vào bảng PHONGBAN mà giá trị nhập cho cột MAPHG kiểu chuỗi thì nó sẽ báo lỗi vì cột MAPHG ta khai báo kiểu int

Để bắt lỗi cho câu lệnh này ta đặt lệnh trong khối TRY, ta thêm lệnh in ra màn hình: Thêm dữ liệu thành công để kiểm tra

Để xử lý lỗi ta đưa vào khối CATCH

In ra màn hình: Thêm dữ liệu vào bảng thất bại. Đồng thời hiển thị mã số lỗi bằng hàm ERROR\_NUMBER() và thông báo lỗi dạng text bằng hàm ERROR\_MESSAGE()

BEGIN TRY

INSERT INTO PHONGBAN VALUES ('ABC','20BP','001','05-23-2022')

PRINT N'Thêm dữ liệu thành công'

END TRY

BEGIN CATCH

PRINT N'Thêm dữ liệu thất bại'

PRINT N'Error ' +CONVERT(VARCHAR,ERROR\_NUMBER())+ ': ' +ERROR\_MESSAGE()

END CATCH

# SLIDE 28: RAISERROR

<https://www.vndba.net/tim-hieu-ve-cau-lenh-raiserror-cua-sql-server/>

# LAB 4 BÀI 2

------Cho biết thông tin nhân viên (HONV, TENLOT, TENNV) có MaNV là số chẵn.

DECLARE @a int =1,@msln int

SELECT @msln= MAX(CONVERT(INT,MANV)) FROM NHANVIEN

WHILE @a <= @msln

BEGIN

IF @a%2=0

BEGIN

IF EXISTS (SELECT \* FROM NHANVIEN WHERE MANV=@a)

BEGIN

SELECT HONV,TENLOT,TENNV,MANV

FROM NHANVIEN

WHERE MANV=@a

SET @a=@a+1

END

ELSE SET @a=@a+1

END

ELSE SET @a=@a+1

END

C2

DECLARE @A INT=2, @MASOMAX INT

SELECT @MASOMAX= MAX(CONVERT(INT,MANV)) FROM NHANVIEN

WHILE @A <= @SONV

BEGIN

IF @A=4

BEGIN

SET @A=@A+2

CONTINUE;

END

ELSE

BEGIN

IF EXISTS(SELECT \* FROM NHANVIEN WHERE MANV=@A)

BEGIN

select HONV, TENLOT, TENNV, MANV

FROM NHANVIEN

WHERE MANV=@A

SET @A=@A+2

END

ELSE SET @A=@A+2

END

END

--Cho biết thông tin nhân viên (HONV, TENLOT, TENNV) có MaNV là số chẵn nhưng

--không tính nhân viên có MaNV là 4.

DECLARE @a int =1,@msln int

SELECT @msln= MAX(CONVERT(INT,MANV)) FROM NHANVIEN

WHILE @a <= @msln

BEGIN

IF @a%2=0

BEGIN

IF @a=4 BEGIN SET @a=@a+1 CONTINUE END

ELSE IF EXISTS (SELECT \* FROM NHANVIEN WHERE MANV=@a)

BEGIN

SELECT HONV,TENLOT,TENNV,MANV

FROM NHANVIEN

WHERE MANV=@a

SET @a=@a+1

END

ELSE SET @a=@a+1

END

ELSE SET @a=@a+1

END

--C2

DECLARE @i int=2, @malonnhat int

SELECT @malonnhat=MAX(CAST(MANV AS INT)) FROM NHANVIEN

WHILE @i<=@malonnhat

BEGIN

IF EXISTS(SELECT \* FROM NHANVIEN WHERE MANV=@i)

IF @i!=4

SELECT HONV,TENLOT, TENNV, MANV

FROM NHANVIEN

WHERE MANV=@i

SET @i=@i+2

END

# LAB4 BÀI 3

--Thực hiện chèn thêm một dòng dữ liệu vào bảng PhongBan theo 2 bước

--o Nhận thông báo “ thêm dư lieu thành cong” từ khối Try

--o Chèn sai kiểu dữ liệu cột MaPHG để nhận thông báo lỗi “Them dư lieu that bai”

--từ khối Catch

BEGIN TRY

INSERT INTO PHONGBAN VALUES (N'Kỹ thuật','ABD','002','05-23-2022')

PRINT N'Thêm dữ liệu thành công'

END TRY

BEGIN CATCH

PRINT N'Thêm dữ liệu thất bại'

PRINT 'Error '+CONVERT(INT, ERROR\_NUMBER())+': '+ CONVERT(INT, ERROR\_MESSAGE())

END CATCH

--Viết chương trình khai báo biến @chia, thực hiện phép chia @chia cho số 0 và dùng

--RAISERROR để thông báo lỗi.

BEGIN TRY

DECLARE @a int=5

SET @a=@a/0

END TRY

BEGIN CATCH

DECLARE @thongbaoloi nvarchar(2048),@mucdo int,@matrangthai int

SELECT @thongbaoloi=ERROR\_MESSAGE(),

@mucdo= ERROR\_SEVERITY(),

@matrangthai=ERROR\_STATE()

RAISERROR(@thongbaoloi,@mucdo,@matrangthai)

END CATCH

# LAB4 BÀI 4

--Hiển thị tên phòng ban có mã phòng là số lẻ, không tính số 1,

--sử dụng vòng lặp WHILE

DECLARE @dem int =1,@mamax int

SELECT @mamax=MAX(MAPHG) FROM PHONGBAN

WHILE @dem <=@mamax

BEGIN

IF @dem%2=1

BEGIN

IF @dem=1

BEGIN

SET @dem=@dem+1

CONTINUE

END

ELSE IF EXISTS(SELECT \* FROM PHONGBAN WHERE MAPHG=@dem)

BEGIN

SELECT TENPHG,MAPHG

FROM PHONGBAN

WHERE MAPHG=@dem

SET @dem=@dem+1

END

ELSE SET @dem=@dem+1

END

ELSE SET @dem=@dem+1

END

--Hiển thị tên các công việc của đề án Tin học hóa, sử dụng WHILE

DECLARE @a int=1

WHILE @a <=(SELECT MAX(STT)

FROM DEAN a, CONGVIEC b

WHERE a.MADA=b.MADA

AND TENDEAN=N'Tin học hóa')

BEGIN

SELECT STT, TEN\_CONG\_VIEC

FROM DEAN a, CONGVIEC b

WHERE a.MADA=b.MADA

AND TENDEAN=N'Tin học hóa'

AND STT=@a

SET @a=@a+1

END

DECLARE @i int=1, @sl int

SELECT @sl= COUNT(\*) FROM CONGVIEC A, DEAN B

WHERE A.MADA=B.MADA

AND TENDEAN LIKE N'Tin học hóa'

WHILE @i<=@sl

BEGIN

SELECT TEN\_CONG\_VIEC

FROM CONGVIEC A, DEAN B

WHERE A.MADA=B.MADA

AND TENDEAN LIKE N'Tin học hóa'

AND STT=@i

SET @i=@i+1

END