



CÁC HỆ QUẢN TRỊ CSDL

ĐẠI HỌC SƯ PHẠM TP.HCM
KHOA CÔNG NGHỆ THÔNG TIN
Phiên bản 2019



Ôn tập: CÀI ĐẶT RÀNG BUỘC

Chương 1. TỔNG QUAN

Chương 2. TỔ CHỨC LƯU TRỮ

Chương 3. TỐI ƯU TRUY VẤN

Chương 4. LẬP TRÌNH VỚI CURSORS

Chương 5. XỬ LÝ TRUY XUẤT ĐỒNG THỜI

Bài giảng

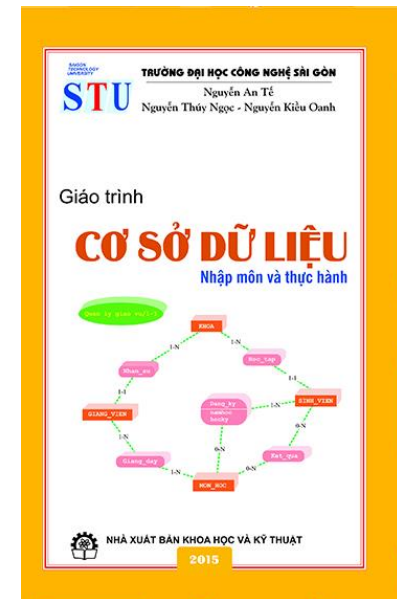


[1] Khoa CNTT- ĐH Sư Phạm, Slide bài giảng
Các hệ Cơ sở dữ liệu, 2014

[2] Nguyễn An Tế, Cơ sở dữ liệu nhập môn và
thực hành, ĐH Công Nghệ Sài Gòn 2016.

[3] Ramez Elmasri, Shamkant B Navathe,
Fundamentals of Database Systems, 6th
Edition, 2011.

[4] Jeffrey D. Ullman, Jennifer Widom, Hector
Garcia-Monlina, Database Systems: The
complete Book, 2001.





- ❑ Số buổi: 7 buổi + 1 buổi thi giữa kỳ.
- ❑ Phòng: i202, bắt đầu lúc 7h30.
- ❑ Bảng điểm: 50 – 50
 - Giữa kỳ: thi thực hành script + bài tập (+điểm)
 - Cuối kỳ: tự luận, thi tập trung
- ❑ Phân nhóm: 3 – 4 thành viên luyện tập thực hành.
- ❑ Nhóm trao đổi:
fb.com/groups/dbms.hcmue/



CÁC HỆ QUẢN TRỊ CSDL

ÔN TẬP CÀI ĐẶT RÀNG BUỘC



❑ Khai báo biến:

DECLARE @Tên_Biến Kiểu_Dữ_Liệu

❑ Ví dụ:

DECLARE @Tuoi **int**

DECLARE @MSSV **varchar**(5)

DECLARE @numCount **int**

❑ Tên biến: Bắt đầu bởi @

❑ Kiểu dữ liệu của biến:

Lấy kiểu dữ liệu hệ thống
trừ kiểu **text**, **ntext**, **image**



❑ Cách 1:

SET @Tên_Biến = **Giá_Trị**

❑ Ví dụ:

DECLARE @HoTen nvarchar(20)

SET @HoTen = N'**Nguyễn Hằng Nga**'



❑ Cách 2:

```
SELECT @Tên_Biến = Giá_Trị
```

❑ Ví dụ:

```
DECLARE @HoTen nvarchar(20)
```

```
SELECT @HoTen = N'Nguyễn Hằng Nga'
```




❑ Cách 3:

```
SELECT @Tên_Biến = Tên_cột  
FROM   Tên_Bảng
```

❑ Ví dụ:

Tìm lương lớn nhất của tất cả nhân viên:

```
DECLARE @MaxSalary decimal(18,2)  
SELECT  @MaxSalary = MAX(Luong)  
FROM    NhanVien
```



- ❑ Các biến được sử dụng trong câu truy vấn như là các tham số.
- ❑ Cho lược đồ CSDL và dữ liệu tương ứng như sau:

HocSinh			
MaHS	TenHS	NgaySinh	DiaChi
01	Bảo	10/10/1977	123
02	Hải	11/11/1981	456

DiemThi			
MaHS	HocKy	NamHoc	Diem
01	01	2001	10
03	01	2002	8



Ví dụ:

Liệt kê danh sách học sinh có ngày sinh vào ngày '10/10/1977'

```
DECLARE @NgàySinh datetime
SET      @NgàySinh = '10/10/1977'
SELECT * FROM HocSinh
WHERE NgàySinh = @NgàySinh
```



Ví dụ:

Liệt kê danh sách các học sinh có địa chỉ là '123' và điểm thi lớn hơn 7

```
DECLARE @DiaChi nvarchar(50), @Diem Decimal
SELECT @DiaChi='123', @Diem = 7
SELECT * FROM HocSinh JOIN DiemThi ON
        HocSinh.MaHS = DiemThi.MaHS
WHERE DiaChi = @DiaChi AND DiemThi > @Diem
```



❑ Cấu trúc điều kiện:

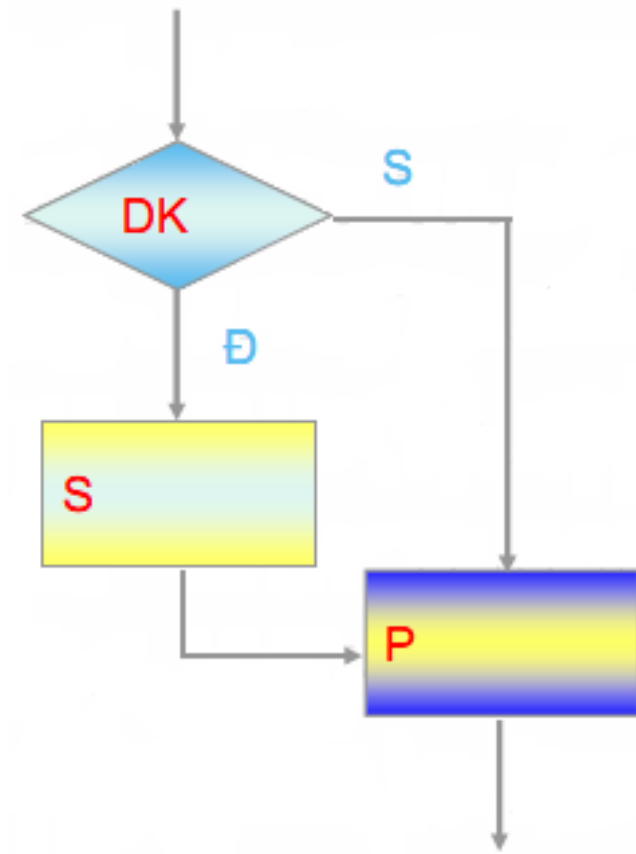
Nếu (biểu thức điều kiện) **thì**
Lệnh/Khối lệnh

IF (biểu thức điều kiện)
BEGIN

Lệnh/Khối lệnh **S**
hoặc SQL Statement

END

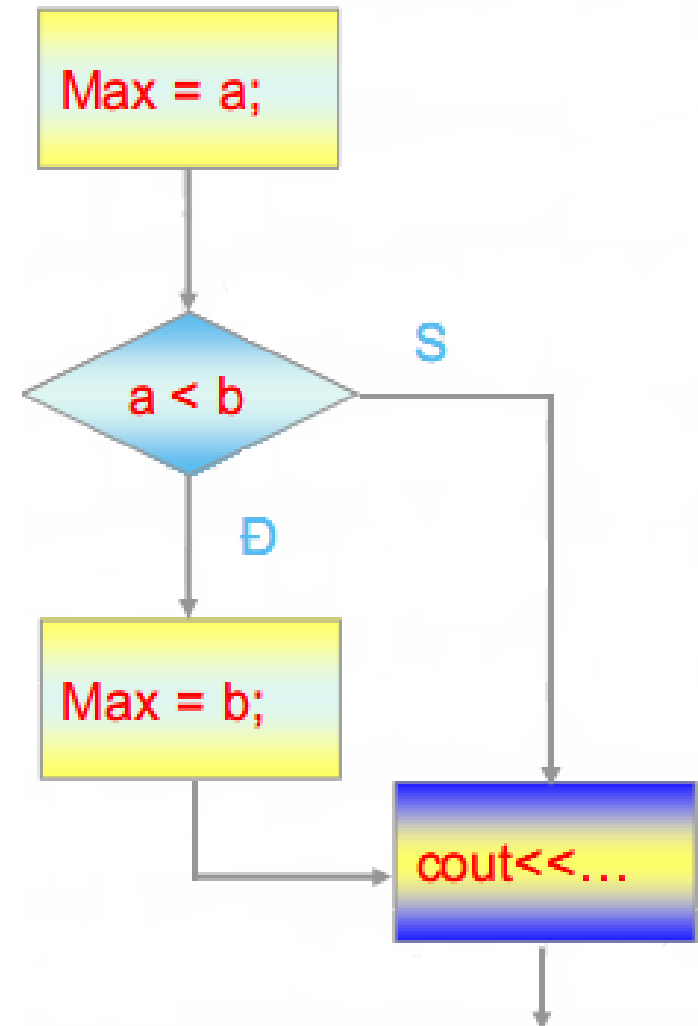
Lệnh/Khối lệnh **P**
hoặc SQL Statement





❑ Tìm Max 2 số

```
DECLARE @a, @b, @Max int
SET @Max = @a
IF (@a < @b)
BEGIN
    SET @Max = @b
END
Print @Max
```



Cấu trúc điều kiện



❑ Nếu (biểu thức điều kiện) thì

Lệnh/Khối lệnh S1

❑ Ngược lại

Lệnh/Khối lệnh S2

IF (biểu thức điều kiện)

BEGIN

Lệnh/Khối lệnh S1

END

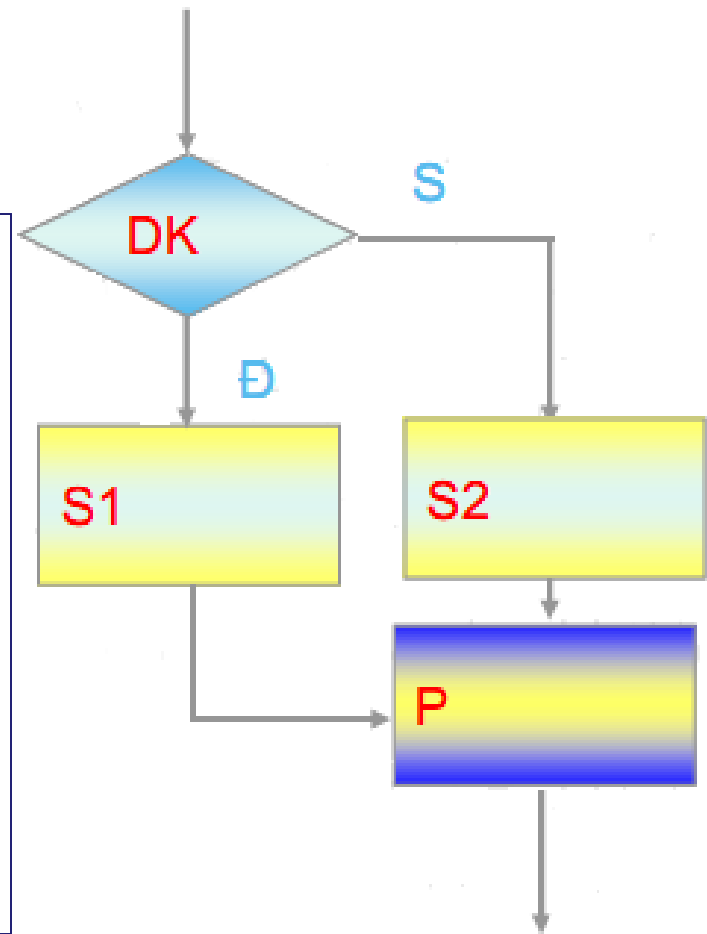
ELSE

BEGIN

Lệnh/Khối lệnh S

END

Lệnh/Khối lệnh P





□ Tìm Max 2 số

```
DECLARE @a, @b, @Max int
IF (@a < @b)
BEGIN
    SELECT @Max = @b
END
ELSE
BEGIN
    SELECT @Max = @a
END
Print @Max
```




- ❑ Cho phép kiểm tra điều kiện và xuất thông tin theo từng trường hợp
- ❑ Cú pháp 1

```
CASE <tên cột>/<biểu thức>
```

```
    WHEN <giá trị> THEN <biểu thức>
```

```
    WHEN <giá trị> THEN <biểu thức>
```

```
    ...
```

```
    [ELSE <biểu thức>]
```

```
END
```



□Cú pháp 2

```
CASE WHEN <giá trị> THEN <biểu thức>  
    WHEN <giá trị> THEN <biểu thức>  
    ...  
    [ELSE <biểu thức>]  
END
```



Ví dụ 1:

```
SELECT TENCLB1, 'Kết quả' =
```

```
CASE
```

```
    WHEN (SOBANTHANG – SOBANTHUA > 0) THEN 'Thắng'
```

```
    WHEN (SOBANTHANG – SOBANTHUA = 0) THEN 'Hòa'
```

```
    WHEN (SOBANTHANG – SOBANTHUA < 0) THEN 'Thua'
```

```
END,
```

```
TENCLB2
```

```
FROM vKETQUA
```



CHITIETBAN(sohd, masach, slban, dgban)

Ví dụ 2:

```
SELECT masach,sum(slban)as tongslban,  
      (CASE WHEN sum(slban)>10 THEN  
            N'Bán chạy'  
      ELSE  
            N'Bán chậm'  
      END) as thongtin  
FROM CHITIETBAN  
GROUP BY masach
```

Ví dụ Case (tt)



SACH (masach, tuasach, sotrang, matheloai, slton)

Ví dụ 3:

Tăng số trang của những sách thuộc thể loại Tin học (1) +10,
Giảm số trang của những sách thuộc thể loại Toán học(4) -10

UPDATE sach SET sotrang=sotrang+

CASE WHEN matheloai=1 THEN

10

WHEN matheloai=4 THEN

-10

ELSE

0

END



```
WHILE (biểu thức logic)
BEGIN
    //Lệnh/Khoi lệnh S
END
//Lệnh/khoi lệnh sau while
```

□Viết chương trình tính tổng $s = 1 + 2 + \dots + n$

```
DECLARE @i,@S INT
SELECT @i=1,@s=0
while (@i<=@n)
BEGIN
    SELECT @s = @s+@i;
    SELECT @i = @i+1;
END
PRINT @S
```



- ❑ **BREAK**: Thoát khỏi vòng lặp WHILE
- ❑ **CONTINUE**: Thực hiện lần lặp mới



❑ Đổi một số thành chuỗi:

STR(số_thực, số_ký_tự [, Số_lẻ])

Ví dụ :

❑ `SELECT STR(123);` **=> Result: '123'**

❑ `SELECT STR(123.5);` **=> Result: '124'**

(result is rounded because decimal places defaults to 0)

❑ `SELECT STR(123.5, 5);` **=> Result: '124'**

(result is rounded because decimal places defaults to 0)

❑ `SELECT STR(123.5, 5, 1);` **=> Result: '123.5'**

❑ `SELECT STR(123.456, 7, 3);` **=> Result: '123.456'**

❑ `SELECT STR(123.456, 7, 0);` **=> Result: '123'**

(result is rounded because decimal places is set to 0)

❑ `SELECT STR(123.456, 7);` **=> Result: '123'**

(result is rounded because decimal places defaults to 0)



❑ Đổi kiểu dữ liệu:

CAST(Biểu thức AS Kiểu dữ liệu)

Ví dụ : Cast (@tong as varchar(10))

❑ Đổi kiểu dữ liệu và định dạng:

**CONVERT(Kiểu_dữ_liệu, Biểu_thức
[, Định dạng])**

Ví dụ : Convert(char(10), Getdate(), 105)

Các hàm chuyển đổi kiểu dữ liệu



[http://www.sql-server-helper.com/
tips/date-formats.aspx](http://www.sql-server-helper.com/tips/date-formats.aspx)

Định dạng	Hiển thị dữ liệu
101	Mm/dd/yyyy
103	Dd/mm/yyyy
105	Dd-mm-yyyy
112	Yyyymmdd

transaction

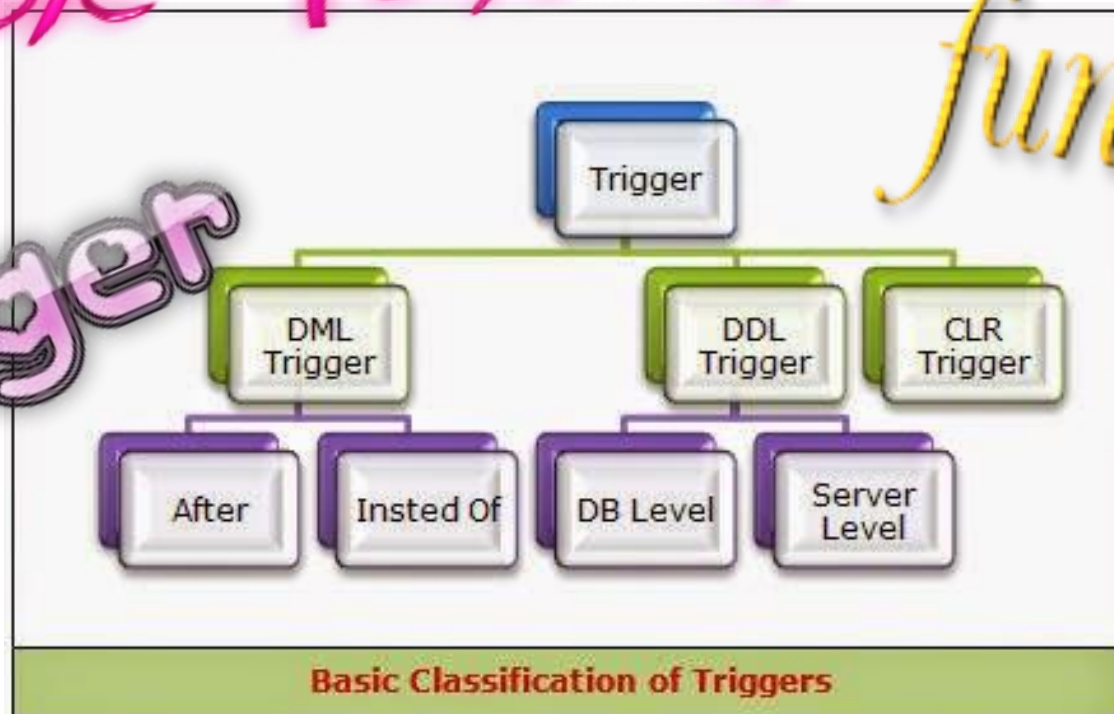


store procedure

rule (check)

function

trigger





- ❑ Trigger sẽ có hiệu lực khi chúng ta thay đổi dữ liệu trên một bảng dữ liệu cụ thể, hoặc các xử lý làm thay đổi dữ liệu của các lệnh: **insert, update, delete**.
- ❑ Trigger có thể chứa các lệnh truy vấn từ các bảng khác hoặc bao gồm những lệnh SQL phức tạp.



- ❑ Một số thuận lợi khi sử dụng trigger:
 - Trigger chạy một cách tự động: khi có sự thay đổi dữ liệu trên bảng dữ liệu.
 - Trigger có thể thực hiện cascade khi có ảnh hưởng đến những bảng liên quan.
 - Trigger có những hiệu lực ít bị hạn chế hơn so với ràng buộc giá trị (có thể ràng buộc tham chiếu đến những cột của những bảng dữ liệu khác).



❑ Lệnh tạo Trigger:

Create Trigger *trigger_name*

on *table_name*

for [insert,update,delete]

As

Begin

{Khai báo các biến xử lý}

{Các lệnh Transact-SQL}

End

❑ Lệnh xóa Trigger:

Drop Trigger *trigger_Name*

Xử lý lỗi trong Trigger



❑ Để thông báo lỗi trong trigger ta dùng hàm :

**Raiserror('Chuỗi thông báo lỗi',
Mức độ nghiêm trọng,
Tham số chuỗi báo lỗi)**

- Mức từ 17 đến 25 chỉ ra các lỗi về phần cứng và phần mềm và thường chỉ có nhà quản trị hệ thống mới giải quyết và theo dõi các lỗi này
- Mức từ 11 đến 16 được tạo ra bởi người dùng và người dùng có thể tự sửa chữa được.
- Mức 10 thường lỗi mang tính chất thông báo và thường do nhầm lẫn khi gõ lệnh.

❑ Không cho thay đổi dữ liệu : **Rollback Tran**



Khi thêm cầu thủ mới, kiểm tra vị trí trên sân của cầu thủ chỉ thuộc một trong các vị trí sau:

Thủ môn, Tiền đạo,
Tiền vệ, Trung vệ, Hậu vệ.

Ví dụ Trigger



```
Create trigger tg_Cauthu_vitri on Cauthu
FOR INSERT
AS
BEGIN
    declare @vitri nvarchar(20)
    select  @vitri=vitri from inserted
    if @vitri not in  (N'Thủ môn',N'Tiền đạo',N'Tiền vệ',
                      N'Trung vệ',N'Hậu vệ')
    begin
        raiserror ('Vi tri khong hop le',15,1)
        rollback tran
        return
    end
END
```

Ví dụ Trigger



```
Create trigger tg_Cauthu_vitri on Cauthu
FOR INSERT
AS
BEGIN
    declare @vitri nvarchar(20)
    select  @vitri=vitri from inserted
    if @vitri not in (N'Thủ môn',N'Tiền đạo',N'Tiền vệ', N'Trung vệ',N'Hậu vệ')
    begin
        raiserror ('Vi tri khong hop le',15,1)
        rollback tran
        return
    end
END
```

Select * from Cauthu

Insert into Cauthu

**Values (N'Nguyễn văn Tèo em', N'Hậu vệ', '2/14/2000',
Null, 'BBD', 'VN', 21)**

Ví dụ Trigger



```
Create trigger tg_Cauthu_soao on Cauthu
FOR INSERT
AS
BEGIN
    declare @so int, @macb varchar(5)
    select @so=so, @macb=macb from Inserted
    if ((select count(*) from cauthu
        where macb=@macb and so=@so) >1)
    begin
        raiserror (N'Số áo bị trùng',15,1)
        rollback tran
        return
    end
END
```



Khi thêm cầu thủ mới,
kiểm tra số áo của cầu thủ
thuộc cùng một câu lạc bộ
phải khác nhau.

Ví dụ Trigger



```
Create trigger tg_Cauthu_soao on Cauthu
FOR INSERT
AS
BEGIN
    declare @so int, @maclb varchar(5)
    select @so=@so, @maclb=@maclb from Inserted
    if ((select count(*) from cauthu where maclb=@maclb and so=@so) >1)
    begin
        raiserror (N'Số áo bị trùng',15,1)
        rollBack tran
        return
    end
END
```

```
select * from cauthu
insert into Cauthu
values(N'Nguyễn văn Tèo em', N'Hậu vệ',
'2/14/2000', Null, 'BBD', 'VN', 21)
```

Inserted và Deleted Table



- ❑ Khi thêm dữ liệu vào bảng, dữ liệu được thêm vào sẽ đưa vào bảng tạm Inserted
- ❑ Khi xóa dữ liệu ra bảng, dữ liệu xóa sẽ đưa vào bảng tạm Deleted
- ❑ Thao tác cập nhật được xử lý :
 - Xóa dòng dữ liệu cũ (*dữ liệu cũ sẽ đưa vào bảng Deleted*)
 - Thêm dòng dữ liệu mới (*dữ liệu mới sẽ đưa vào bảng Inserted*)
- ❑ Để lấy dữ liệu vừa mới cập nhật vào bảng ta truy cập vào bảng tạm Deleted hoặc Inserted



CÁC HỆ QUẢN TRỊ CSDL

BÀI TẬP CÀI ĐẶT

Phân công công việc



Lớp tự phân nhóm:

1. Tối đa 4 thành viên (Nhóm trưởng phân chia công việc theo theo yêu cầu).
2. Bài tập tuần 1:
 - Tạo Database và nhập liệu cho Database Quản lý đội bóng (Lab1) bằng lệnh script SQL.
 - Mỗi phần truy vấn tự chọn 3 câu để viết SQL.
 - Khi nộp bài sẽ chạy thử trên máy của nhóm.