

# Chương 3:

# Thao tác với cơ sở dữ liệu

---



# I MỤC TIÊU

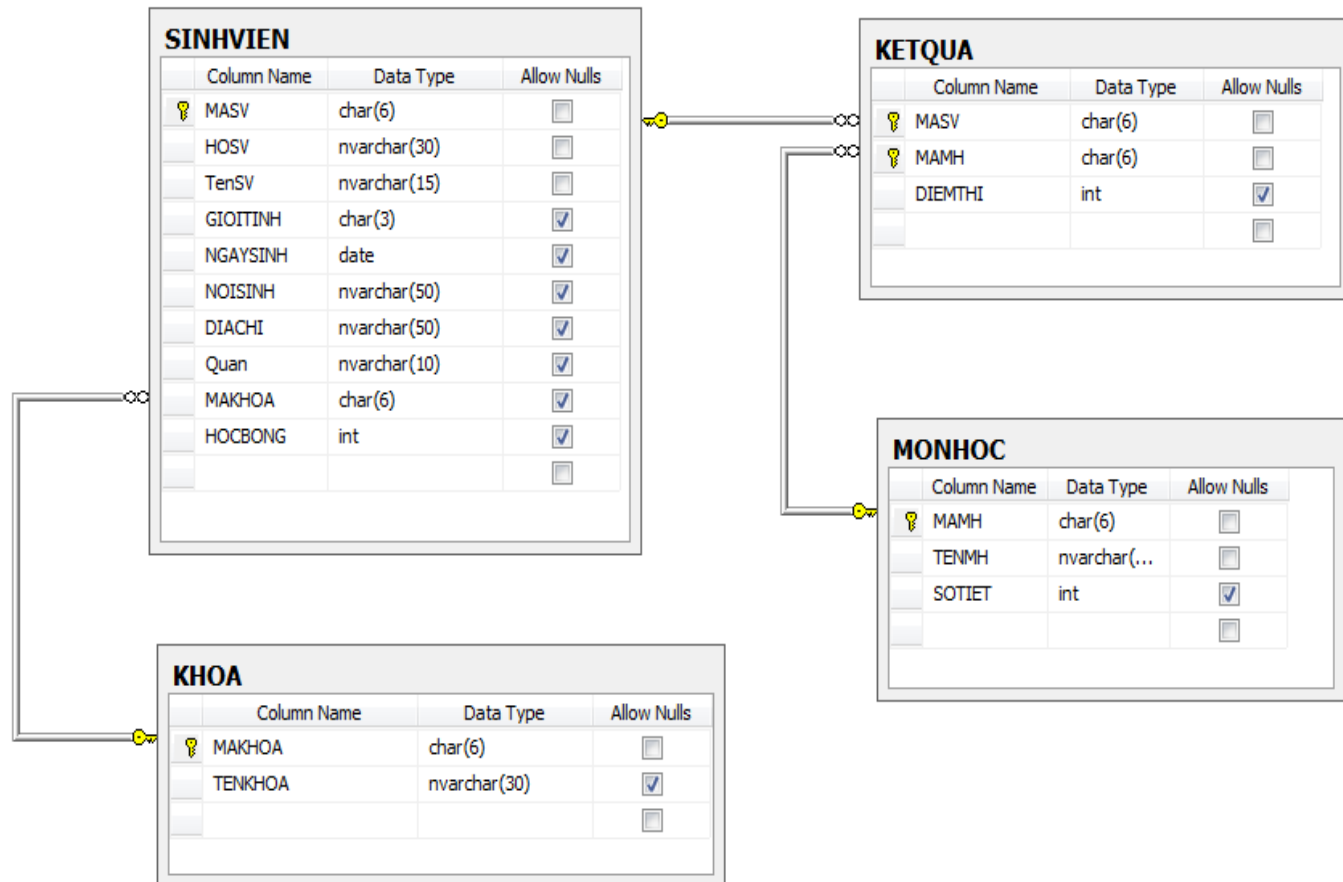
- thiết kế cơ sở dữ liệu và các ràng buộc cơ sở dữ liệu.
- Vận dụng các thao tác với cơ sở dữ liệu
- Viết được các storeProcedure, Function để truy xuất dữ liệu.

# I NỘI DUNG

- Thiết kế cơ sở dữ liệu theo yêu cầu
- Các ràng buộc cơ sở dữ liệu
- Các thao tác với cơ sở dữ liệu
- Storeprocedure
- Function
- Bài tập áp dụng



# Thiết kế cơ sở dữ liệu theo yêu cầu



# | Các ràng buộc cơ sở dữ liệu

Ràng buộc khoá chính :

*[CONSTRAINT PK\_Tên\_bảng\_TenCot ]  
PRIMARY KEY (Danh\_sách\_cột\_khóa\_chính)*

Ràng buộc khoá ngoại

*[CONSTRAINT FK\_Tên\_bảng\_Tên\_cột]  
FOREIGN KEY (Danh\_sách\_cột\_khóa\_ngoại)  
REFERENCES Tên\_bảng\_tham\_chiếu  
(Danh\_sách\_cột\_tham\_chiếu)*

# | Các ràng buộc cơ sở dữ liệu

❖ Thêm Costraint vào bảng đã có

Cú pháp chung:

*ALTER TABLE Tên\_bảng*

*ADD [ CONSTRAINT Tên\_Constraint ]*

*LOẠI Các\_tham\_số [ , ... ]*

# | Các ràng buộc cơ sở dữ liệu

Cú pháp chi tiết từng loại constraint

## PRIMARY KEY

*ALTER TABLE Tên\_bảng*

*ADD [ CONSTRAINT PK\_Tên\_bảng ]*

*PRIMARY KEY (Danh\_sách\_cột\_khóa\_chính)*

# | Các ràng buộc cơ sở dữ liệu

Khóa ngoại FOREIGN KEY

*ALTER TABLE Tên\_bảng*

*ADD [ CONSTRAINT FK\_Tên\_bảng\_Tên\_cột ]*

*FOREIGN KEY (Danh\_sách\_cột\_khóa\_ngoại)*

*REFERENCES Tên\_Bảng*

*(Danh\_sách\_cột\_tham\_chiếu)*



# | Các ràng buộc cơ sở dữ liệu

❖ Kiểm tra miền giá trị :

*ALTER TABLE Tên\_bảng*

*ADD [ CONSTRAINT CK\_Tên\_bảng\_Tên\_cột ]*

*CHECK (Biểu\_thức\_luận\_lý)*

# | Các ràng buộc cơ sở dữ liệu

## ❖ Thêm giá trị mặc định :

*ALTER TABLE Tên\_bảng*

*ADD [ CONSTRAINT DF\_Tên\_bảng\_Tên\_cột ]*

*DEFAULT Giá\_trị\_mặc\_định FOR Tên\_cột*

# Các ràng buộc cơ sở dữ liệu

- ❖ Hủy một Constraint:

*ALTER TABLE Tên\_bảng*

*DROP CONSTRAINT Tên\_constraint [ , ...]*

- ❖ Tắt các Constraint :Chỉ có tác dụng với Check và Foreign key

*ALTER TABLE Tên\_bảng*

*NOCHECK CONSTRAINT ALL| Tên\_constraint [ , ...]*

- ❖ Bật các Constraint:

*ALTER TABLE Tên\_bảng*

*CHECK CONSTRAINT ALL| Tên\_constraint [ , ...]*

# Các ràng buộc cơ sở dữ liệu

## ❖ Xóa cấu trúc bảng

DROP TABLE Danh\_sách\_tên\_các\_bảng

## ❖ Thay đổi cấu trúc bảng

### ➤ Thêm một cột mới trong bảng

*ALTER TABLE Tên\_bảng*

*ADD Tên\_cột Kiểu\_dữ\_liệu [ , ... ]*

### ➤ Hủy bỏ cột hiện có bên trong bảng

*ALTER TABLE Tên\_bảng*

*DROP COLUMN Tên\_cột [ , ...]*

# | Các ràng buộc cơ sở dữ liệu

- ❖ Sửa đổi kiểu dữ liệu của cột

*ALTER TABLE Tên\_bảng*

*ALTER COLUMN Tên\_cột Kiểu\_dữ\_liệu\_mới*

- ❖ Đổi tên cột

*EXEC SP\_Rename "Tên\_bảng.Tên\_cột"*  
*, "Tên\_mới", "COLUMN"*

# | Các ràng buộc cơ sở dữ liệu

## ❖ Đổi tên bảng

EXEC sp\_rename "Tên\_bảng","Tên\_mới"

*Chú ý: Muốn thay đổi cấu trúc của cột phải xóa tất cả các Constraint liên quan*

# | Các thao tác với cơ sở dữ liệu

Câu lệnh truy vấn dữ liệu là các câu lệnh cho phép lấy dữ liệu từ các bảng, thêm xóa sửa dữ liệu

# | Các thao tác với cơ sở dữ liệu

## Cấu trúc câu lệnh Select :dùng lấy dữ liệu

*Select [Ten\_Bang.]Ten\_Cot[,...]*

*From Ten\_Bang1 <Liên kết> Ten\_Bang2*

*Where <Điều kiện>*

*Group by <các cột gom nhóm>*

*HaVing <Điều kiện lọc nhóm>*

*Order by [Ten\_Bang.]Ten\_cot [asc/desc,...]*

*Compute Hàm thống kê [By Ten\_cot]*



# Các thao tác với cơ sở dữ liệu

**Insert into : dùng thêm dữ liệu vào bảng,có 2 dạng**

## ➤ Thêm một dòng

- ✓ Cú pháp : `Insert into Ten_bang(Ten_cot1,Ten_cot2[,...])  
Values(Gia_Tri1,...)`
- ✓ với giá trị :
  - Kiểu chuỗi : nằm trong hai dấu " ví dụ 'Trần thanh nhật',Nếu chuỗi kiểu Nvarchar :N'Trần Thanh Nhật'
  - Kiểu Ngày : 'yyyy-mm-dd' ví dụ : '1982-02-29'

## ➤ Thêm nhiều dòng

- ✓ Cú pháp : `Insert into Ten_bang_Dich(Ten_cot1,Ten_cot2[,...])  
Select Ten_bang_nguon.ten_cot,giatri,bieuthuc...  
from ...`

# Các thao tác với cơ sở dữ liệu

➤ **Update** : dùng cập nhật dữ liệu của một bảng

❖ **Dữ liệu Update chỉ trên một Bảng :**

Cú pháp :

*Update Ten\_Bang  
Set Ten\_Cot= Giá trị/Biểu thức  
[Where Điều kiện]*

❖ **Dữ liệu Update trên nhiều Bảng :**

Cú pháp :

*Update Ten\_Bang  
Set Ten\_Cot= Giá trị/Biểu thức  
From ...  
[Where Điều kiện]*

**Chú ý: giá trị cập nhật không được dùng hàm thống kê**

19-10-2021

Lập trình ứng dụng

18



# Các thao tác với cơ sở dữ liệu

□ **Delete** : dùng xóa dữ liệu của một bảng

❖ **Dữ liệu xóa chỉ trên một Bảng :**

Cú pháp :

*delete Ten\_Bang*  
*[Where Điều kiện]*

❖ **Dữ liệu delete trên nhiều Bảng :**

Cú pháp :

*delete Ten\_Bang*  
*From ...*  
*[Where Điều kiện]*

**Chú ý** : khi xóa dữ liệu trên bảng quan hệ 1 mà dữ liệu này đã có bên bảng quan hệ nhiều -> có lỗi

# Các thao tác với cơ sở dữ liệu

## Biến và các cấu trúc lệnh trong SQL Server

- **Biến :**

Biến cục bộ : là biến do người lập trình khai báo, Biến có thể được khai báo trong thủ tục nội tại, hàm, trong một lô lệnh

- Lô Lệnh : là tập hợp các lệnh được kết thúc bằng từ Go
- Cú pháp khai báo biến :

**Declare @Tên\_Biến Kiểu\_Dữ\_Liệu**

- ✓ Tên Biến : Luôn bắt đầu bằng ký tự @, không có khoảng trắng, ký tự đặt biệt, ký tự số đứng đầu

# Các thao tác với cơ sở dữ liệu

## Biến và các cấu trúc lệnh trong SQL Server

- Gán giá trị cho biến:

Gán bằng 1 giá trị :

*Set @tên\_biến=giá trị/hàm*

*hoặc Select @tên\_biến=Giá trị/hàm*

- Gán bằng kết quả trả về của câu truy vấn

*Select @tên\_Biến=Tên cột/Hàm ..from ..*

# | Các thao tác với cơ sở dữ liệu

## Các cấu trúc lệnh

Lệnh điều khiển IF .. Else

*Cú pháp :*

*If Điều kiện*

*Begin*

*<tập Lệnh 1>*

*end*

*else*

*Begin*

*<tập Lệnh 2>*

**End**

- Nếu điều kiện đúng: tập Lệnh 1 thực hiện, ngược lại tập lệnh 2 thực hiện
- Nếu tập hợp lệnh 1, lệnh 2 chỉ có 1 lệnh thì không cần Begin .. End

# Các thao tác với cơ sở dữ liệu

## Cấu trúc lặp: While

*While Biểu thức Điều kiện*

*Begin*

*<Tập lệnh>*

*end*

- ✓ Tập lệnh sẽ được thực hiện đến khi biểu thức điều kiện trả về False
- ✓ có thể dùng lệnh Break để thoát khỏi vòng lặp

# Các thao tác với cơ sở dữ liệu

## Các hàm cơ bản trong SQL Server

- ❑ Các hàm chuyển đổi kiểu dữ liệu :thường dùng để chuyển dữ liệu từ số,ngày sang chuỗi

Trong SQL Server ta thể nói các kiểu dữ liệu như chuỗi, ngày, số lại với nhau, muốn nói ta phải chuyển tất cả thành kiểu chuỗi

- ✓ Hàm **Cast** : chuyển một kiểu dữ liệu sang kiểu bất kỳ

- **Cú pháp** : *Cast(Biểu\_Thức as Kiểu\_Dữ\_Liệu)*

**Ví dụ** : *Cast (@tong as varchar(10))*

- ✓ Hàm **CONVERT** : chuyển từ ngày ⇔ chuỗi

**Cú pháp**:

*Convert(Kiểu\_dữ\_Liệu,Biểu\_Thức[,Định\_Dạng])*

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



# Storeprocedure (thủ tục nội tại)

- Storeprocedure (thủ tục nội tại) là gì ? là “Chương trình con” của Sql Server
- Trong thủ tục nội tại có thể chứa các câu lệnh truy vấn, các lệnh T-SQL...
- Thủ tục nội tại có thể có các tham số vào, tham số ra và có thể trả về kết quả
- Ta có thể gọi thủ tục nội tại ở trong SQL hay trong các ứng dụng được xây dựng bằng C#, VB.NET...
- Các xử lý xây dựng trong thủ tục nội tại sẽ chạy nhanh hơn khi xây dựng ngoài ứng dụng, và tập trung tại server nên dễ quản lý
- Theo mô hình lập trình client-server tất cả các xử lý đều tập trung tại server, trong sql server các xử lý này nằm trong thủ tục nội tại

# | Storeprocedure (thủ tục nội tại)

## Cú pháp xây dựng thủ tục nội tại

*Create Proc Ten\_Thu\_Tuc [(Các tham số)]*

*As*

*các câu lệnh*

## Trong đó :

Các tham số : chia làm hai loại

Tham số vào : nhận giá trị từ người dùng gửi vào cho ttnt xử lý

Cú pháp : @Ten\_bien Kiểu\_dữ\_liệu [=giá trị mặc định]

Tham số ra : nhận kết quả trả về từ Ttnt và hiển thị cho người dùng

Cú pháp : @Ten\_bien Kiểu\_dữ\_Liệu OutPut

# Storeprocedure (thủ tục nội tại)

## ❑ Gọi thực hiện thủ tục nội tại

*Exec Ten\_Thu\_Tuc [ @ten\_Tham\_so\_vao= gia\_tri[,..]*

*@ten\_Tham\_so\_ra=@ten\_Bien output]*

## ❑ Lệnh Return trong thủ tục nội tại :

Trong thủ tục nt ta có thể dùng lệnh Return để trả về kết quả hoặc để thoát khỏi thủ tục

Cú pháp :

➤ Return giá trị :thoát khỏi thủ tục và trả về kết quả là giá trị

Hoặc

➤ Return : Thoát khỏi thủ tục, trả về 0

Gọi thủ tục có giá trị trả về

*Exec @ten\_bien=Ten\_thu\_Tuc [ @ten\_Tham\_so\_vao= gia\_tri[,..]*

*@ten\_Tham\_so\_ra=@ten\_Bien output]*

# | Storeprocedure (thủ tục nội tại)

**Các dạng thủ tục nội tại :**

## **A. Thủ tục hiển thị dữ liệu:**

- ✓ Loại thủ tục này có thể có tham số vào
- ✓ Trong nó chứa câu lệnh Select

## **B. Thủ tục Tính Toán :**

- ✓ Loại thủ tục này có thể có tham số vào, tham số ra, giá trị trả về
- ✓ Trong nó chứa câu lệnh T-SQL để kiểm tra, tính toán giá trị trả về

# Storeprocedure (thủ tục nội tại)

## A. Thủ tục Thêm dữ liệu vào bảng:

- ✓ Loại thủ tục này có tham số vào là tên các cột có trong bảng, trừ các cột có kiểu dữ liệu tự tăng (identity)
- ✓ Tham số ra, giá trị trả về :có thể có, cho biết việc thêm dữ liệu có thành công hay không
- ✓ Trong nó chứa câu lệnh T-SQL :

Kiểm tra ràng buộc dữ liệu duy nhất(primary key,unique)

**Cú pháp :**

*if Exists (Select \* from Ten\_Bang where Ten\_Cot=@ten\_cot)*

*begin*

*print 'thông báo lỗi'*

*End*

# | Storeprocedure (thủ tục nội tại)

## ❖ Kiểm tra ràng buộc khóa ngoại (Foreign Key)

### Cú pháp

```
if not Exists (Select * from Ten_Bang1 where Ten_Cot=@ten_cot)  
begin
```

```
print 'thông báo lỗi'
```

## ❖ Kiểm tra ràng buộc Miền giá trị (Check)

```
if @ten_cot không nằm trong miền giá trị  
begin
```

```
print 'thông báo lỗi'
```

# | Storeprocedure (thủ tục nội tại)

## ➤ Ghi dữ liệu vào bảng

Cú pháp

*Insert into Ten\_Bang(Ten\_cot1[,...]) values( @Ten\_cot1[,...])*

## ➤ Cập nhật dữ liệu vào các bảng liên quan

Cú pháp

*Update Ten\_Bang Set Ten\_Cot= Giatri where <Điều kiện>*

# | Storeprocedure (thủ tục nội tại)

## ❖ Thủ tục Cập nhật dữ liệu của bảng:

- ✓ Loại thủ tục này có tham số vào là tên các cột có trong bảng, Tham số ra, giá trị trả về :có thể có, cho biết việc Cập nhật dữ liệu có thành công hay không
- ✓ Trong nó chứa câu lệnh T-SQL :  
Kiểm tra dữ liệu cập nhật có tồn tại hay không

**cú pháp:**

*if not Exists (Select \* from Ten\_Bang where  
Ten\_Cot\_khoa\_chinh=@ ten\_cotkhoa\_chinh)*

*begin*

*print 'thông báo lỗi'*

19-10-2021

*End*

Lập trình ứng dụng

32





# | Storeprocedure (thủ tục nội tại)

- ❖ Kiểm tra ràng buộc khóa ngoại
- ❖ Kiểm tra ràng buộc miền giá trị
- ❖ Cập nhật dữ liệu vào bảng

## Cú pháp

*Update Ten\_Bang*

*Set ten\_cot=@tencot [...]*

*Where ten\_cot khoa chinh=@ten\_cot khoa chinh*

**Chú ý:** không cập nhật cột làm khóa chính và cột có thuộc tính identity

Cập nhật dữ liệu của các bảng liên quan

# | Storeprocedure (thủ tục nội tại)

## ❖ Thủ tục xóa dữ liệu của bảng:

- Loại thủ tục này có tham số vào là các cột làm khóa chính trong bảng, Tham số ra, giá trị trả về :có thể có, cho biết việc xóa dữ liệu có thành công hay không
- Trong nó chứa câu lệnh T-SQL:
- Kiểm tra dữ liệu xóa có tồn tại trong bảng nhiều hay không

Cú pháp :

*if Exists (Select \* from Ten\_Bang\_nhiều where  
Ten\_Cot\_khoa\_ngoại=@ten\_cot\_khoa\_chính)*

*begin*

*print 'thông báo lỗi'*

*return*

*End*

# | Storeprocedure (thủ tục nội tại)

## ❖ Xóa dữ liệu khỏi bảng

Cú pháp

*Delete Ten\_Bang*

*Where ten\_cot\_khoa\_chinh=@ten\_cot\_khoa\_chinh*

Lệnh cập nhật dữ liệu của các bảng liên quan

# | Function

## ❖ Xây dựng Hàm:

- Hàm trả về một giá trị : giá trị trả về có kiểu dữ liệu là một trong các kiểu của SQL Server

Cú pháp :

*Create Function Ten\_Ham[(Các tham số)]*

*Returns Kiểu\_dữ\_liệu\_trả\_về as*

*Begin*

*<các xử lý>*

*Return ket\_qua*

*end*

# Ví dụ function

Xây dựng hàm truyền vào Sodh,Mavtu -> trả về tổng Slnhập

```
create function Tinh_Tong_Slnhap(@sodh char(4), @mavtu  
char(4))
```

```
returns int as
```

```
begin
```

```
declare @tongslnhap int
```

```
select @tongslnhap= isnull(sum(slnhap),0)
```

```
from ctpnhap inner join pnhap on
```

```
ctpnhap.sopn=pnhap.sopn
```

```
where sodh=@sodh and mavtu=@mavtu
```

```
return @tongslnhap
```

```
end
```

19-10-2021

Lập trình ứng dụng

37



# | Function

## Gọi hàm:

```
declare @kq int  
set @kq =dbo.Tinh_Tong_Slnhap('d001','dd01')  
print @kq
```

## Gọi trong câu truy vấn :

```
select Dondh.sodh,convert(char(10),ngaydh,103) as  
ngaydh,vattu.mavtu,tenvtu,sldat,  
dbo.Tinh_Tong_slnhap(dondh.sodh,Vattu.mavtu) as  
tongsldanhap  
from dondh inner join ctdondh on dondh.sodh=ctdondh.sodh  
inner join vattu on ctdondh.mavtu=vattu.mavtu
```

# | Function

## Sử dụng hàm trong lệnh tạo bảng

### Ví dụ : xây dựng hàm

```
create function lay_ngaydh(@sodh char(4))  
returns datetime as  
begin  
declare @ngay datetime  
select @ngay= ngaydh from dondh where sodh=@sodh  
return @ngay  
end
```

# | Function

**B.** Hàm trả về dữ liệu được lấy từ các bảng trong CSDL: giống như view nhưng có tham số vào

Cú pháp :

*Create Function Tên\_Hàm[(các tham số vào)]*

*Returns Table as*

*Return(Câu lệnh select)*



# | Ví dụ function

**ví dụ:** xây dựng hàm truyền vào sodh trả về ctdondh

```
create function lay_ctdondh(@sodh char(4))  
returns table as  
return (select sodh,tenvtu,sldat from vattu inner  
join ctdondh on vattu.mavtu=ctdondh.mavtu  
where sodh= @sodh)
```

**--goi hàm**

```
select * from dbo.lay_ctdondh('d002')
```

# | Bài tập áp dụng

19-10-2021

Lập trình ứng dụng

42



FACULTY OF INFORMATION  
TECHNOLOGY  
THU DUC COLLEGE OF  
TECHNOLOGY

