

*Trường Đại học Khoa học Tự nhiên*

*Khoa Công nghệ Thông tin*

*Bộ môn Hệ thống Thông tin*

# *Quản trị Cơ sở Dữ liệu*

## *Chương II :* **Xây dựng CSDL**

**GV : TUẤN NGUYỄN HOÀI ĐỨC**

**E-mail : [tnhduc@fit.hcmus.edu.vn](mailto:tnhduc@fit.hcmus.edu.vn)**

# Nội dung trình bày

- ◆ Kiểu dữ liệu
- ◆ Kiểu dữ liệu người dùng định nghĩa
- ◆ Tạo và thay đổi Cơ sở Dữ liệu
- ◆ Tạo và thay đổi bảng dữ liệu
- ◆ Các công cụ quan sát cấu trúc dữ liệu
- ◆ Cập nhật dữ liệu



# Kiểu dữ liệu

- ◆ Bit : Nhận giá trị 0 hoặc 1 (1 byte)
- ◆ Bigint :  $-2^{63}$  đến  $2^{63}-1$  (8 bytes)
- ◆ Int : -2,147,483,648 đến 2,147,483,647 (4 bytes)
- ◆ Smallint : -32,768 đến 32,767 (2 bytes)
- ◆ Tinyint : 0 đến 255 (1 byte)
- ◆ Numeric :  $-10^{38}-1$  đến  $10^{38}-1$  (số bytes thay đổi)



# Kiểu dữ liệu

- ❖ Money :  $-2^{63}$  đến  $2^{63}$ , cho phép 4 số thập phân (8 bytes)
- ❖ SmallMoney : -214,748,3648 đến 214,748,3647 (4 bytes)
- ❖ Float :  $-1.79E+308$  đến  $1.79E+308$  (số bytes thay đổi)
- ❖ DateTime : 1753-1-1 đến 9999-12-31 (8 bytes)
- ❖ Date (chỉ có từ phiên bản 2008)
- ❖ Time (chỉ có từ phiên bản 2008)



# Kiểu dữ liệu

- ❖ Small datetime : 1-1-1900 đến 31-12-2079 (4 bytes)
- ❖ Cursor : Sẽ tìm hiểu kỹ trong các chương sau
- ❖ Char : Tối đa 8000 ký tự, không hỗ trợ Unicode. Chiều dài cố định
- ❖ Varchar : Tối đa 8000 ký tự, không hỗ trợ Unicode. Chiều dài biến đổi



# Kiểu dữ liệu

- ❖ NChar : Tối đa 4000 ký tự, có hỗ trợ Unicode. Chiều dài cố định
- ❖ NVarchar : Tối đa 4000 ký tự, có hỗ trợ Unicode. Chiều dài biến đổi
- ❖ Text : Dài tối đa 2,147,483,647 ký tự. Không hỗ trợ Unicode.
- ❖ NText : Dài tối đa 1,073,741,823 ký tự. Có hỗ trợ Unicode.



# Kiểu dữ liệu

- ❖ Image : Dữ liệu nhị phân có độ dài biến đổi, tối đa 2,147,483,647 bytes.
- ❖ Table : Kiểu dữ liệu đặc biệt. Table và cursor là hai kiểu dữ liệu không phải vô hướng, hầu hết kiểu dữ liệu còn lại là dữ liệu vô hướng (scalar).



# Nội dung trình bày

- ◆ Kiểu dữ liệu
- ◆ Kiểu dữ liệu người dùng định nghĩa
- ◆ Tạo và thay đổi Cơ sở Dữ liệu
- ◆ Tạo và thay đổi bảng dữ liệu
- ◆ Các công cụ quan sát cấu trúc dữ liệu
- ◆ Cập nhật dữ liệu



# Kiểu dữ liệu người dùng định nghĩa

◆ Dùng thủ tục `sp_addtype`

`sp_addtype`

[ @typename = ] type,

[ @phystype = ] system\_data\_type

[ , [ @nulltype = ] 'null\_type' ]

[ , [ @owner = ] 'owner\_name' ]

Tên của kiểu dữ liệu mới  
(do người dùng đặt)

Kiểu dữ liệu built-in của SQL  
Server (không default)

Cho phép null hay  
không : 'NULL' (mặc  
định) hay 'NOT NULL'

Tên tác giả kiểu dữ liệu  
này, mặc định lấy tên  
user hiện hành



# Kiểu dữ liệu người dùng định nghĩa

## ❖ Ví dụ :

- ***EXEC sp\_addtype SoCMND, 'VARCHAR(11)', 'NOT NULL'***
- ***EXEC sp\_addtype NgaySinh, datetime, 'NULL'***
- ***EXEC sp\_addtype SoDT, 'varchar(24)', 'NOT NULL'***
- ***EXEC sp\_addtype SoFax, 'varchar(24)', 'NULL'***

## ❖ Xóa kiểu DL người dùng định nghĩa

- ***EXEC sp\_droptype NgaySinh***
- ***EXEC sp\_droptype SoCMND***



## Nội dung trình bày

- ◆ Kiểu dữ liệu
- ◆ Kiểu dữ liệu người dùng định nghĩa
- ◆ Tạo và thay đổi Cơ sở Dữ liệu
- ◆ Tạo và thay đổi bảng dữ liệu
- ◆ Các công cụ quan sát cấu trúc dữ liệu
- ◆ Cập nhật dữ liệu



# Tạo CSDL

```
CREATE DATABASE database_name  
[ ON [ [PRIMARY]< filespec > [ ,...n ] ] ]  
[ LOG ON { < filespec > [ ,...n ] } ]
```

< filespec > định nghĩa như sau

```
( [ NAME = logical_file_name , ]  
  FILENAME = 'os_file_name'  
  [ , SIZE = size ]  
  [ , MAXSIZE = { max_size | UNLIMITED } ]  
  [ , FILEGROWTH = growth_increment ] )  
[ ,...n ]
```



# Tạo CSDL

CREATE DATABASE database\_name

[ ON [ [PRIMARY]< filespec > [ ,...n ] ] ]

[ LOG ON { < filespec > [ ,...n ] } ]

Tên DB, không trùng,  
tối đa 123 char

Danh sách file  
DL nếu muốn  
chỉ định cụ thể

Danh sách file log nếu  
muốn chỉ định cụ thể

< filespec > định nghĩa như sau

( [ NAME = logical\_file\_name , ]

FILENAME = 'os\_file\_name'

[ , SIZE = size ]

[ , MAXSIZE = { max\_size | UNLIMITED } ]

[ , FILEGROWTH = { growth\_increment } ] )

[ ,...n ]

Tên file dùng trong môi  
trường SQL Server

Tên thực sự của file (bao gồm đường dẫn).  
Mặc định, các tập tin dữ liệu và log được lưu trong  
thư mục Microsoft SQL Server\ MSSQL\ DATA



# Tạo CSDL

```
CREATE DATABASE database_name
[ ON [ [PRIMARY]< filespec > [ ,...n
[ LOG ON { < filespec > [ ,...n ] } ]
```

Kích thước ban đầu cho các file dữ liệu. Mặc định 1 MB, tối thiểu 512 KB, đơn vị : KB, MB, TB (mặc định MB)

< filespec > định nghĩa như sau

```
( [ NAME = logical_file_name , ]
FILENAME = 'os_file_name'
[ , SIZE = size ]
[ , MAXSIZE = { max_size | UNLIMITED } ]
[ , FILEGROWTH = growth_increment ] )
[ ,...n ]
```

Kích thước tối đa cho các file dữ liệu. Mặc định là vô tận cho đến khi hết đĩa cứng

Bước tăng kích thước file. Tối thiểu 64 KB, mặc định 10%. Giá trị phần trăm tính trên kích thước hiện hành. Đơn vị mặc định là MB



# Tạo CSDL (Ví dụ 1)

```
CREATE DATABASE QuanLyHangHoa
ON
PRIMARY
( NAME = Primary_File,
  FILENAME = 'c:\program files\microsoft sql
server\mssql\data\archdat1.mdf',
  SIZE = 100MB,
  MAXSIZE = 200,
  FILEGROWTH = 20),
( NAME = Secondary_File_1,
  FILENAME = 'c:\program files\microsoft sql
server\mssql\data\archdat2.ndf',
  SIZE = 100MB,
  MAXSIZE = 200,
  FILEGROWTH = 20),
```

Primary đặt trước filespec nào thì đó là file chính. Nếu khuyết Primary thì mặc định file đầu tiên là file chính.



# Tạo CSDL (Ví dụ 1)

```
( NAME = Secondary_File_2,  
  FILENAME = 'c:\program files\microsoft sql  
server\mssql\data\archdat3.ndf',  
  SIZE = 100MB,  
  MAXSIZE = 200,  
  FILEGROWTH = 20)
```

LOG ON

```
( NAME = Log_File_1,  
  FILENAME = 'c:\program files\microsoft sql  
server\mssql\data\archlog1.ldf',  
  SIZE = 100MB,  
  MAXSIZE = 200,  
  FILEGROWTH = 20),
```



## Tạo CSDL (Ví dụ 1)

```
( NAME = Log_File_2,  
  FILENAME      = 'c:\program files\microsoft sql  
server\mssql\data\archlog2.ldf',  
  SIZE = 100MB,  
  MAXSIZE = 200,  
  FILEGROWTH = 20)  
GO
```



## Tạo CSDL (Ví dụ 2)

```
CREATE DATABASE QuanLySinhVien
ON
( NAME = Data_File1,
  FILENAME = 'c:\program files\microsoft sql
server\mssql\data\archdat1.mdf',
  SIZE = 100MB,
  MAXSIZE = 200,
  FILEGROWTH = 20),
( NAME = Data_File_2,
  FILENAME = 'c:\program files\microsoft sql
server\mssql\data\archdat2.ndf',
  SIZE = 100MB,
  MAXSIZE = 200,
  FILEGROWTH = 20)
```



# Thay đổi và xoá CSDL

- ❖ Xoá một CSDL đã tồn tại:  
***Drop Database database\_name***

- ❖ Thay đổi một CSDL

- ***Sửa tên :***

```
ALTER DATABASE AdventureWorks2012  
Modify Name = Northwind  
GO
```

- ***Thêm file mới :***

```
ALTER DATABASE [QLDA]  
ADD FILE ( NAME = N'QLDA_Data2',  
FILENAME = N'C:\Program Files\Microsoft SQL  
Server\MSSQL.1\MSSQL\DATA\QLDA_Data2.ndf',  
SIZE = 2048KB , FILEGROWTH = 1024KB )  
GO
```

Thêm file nhật ký thì dùng  
ADD LOG FILE



# Thay đổi và xoá CSDL

## ◆ Thay đổi một CSDL

### • *Sửa file đã có :*

Alter database QLDA

Modify File (Name= QLDA\_data,

NewName = QLDA1, FileName = 'D:\myfile.ndf',  
SIZE = 200MB)

GO

1 lệnh chỉ modify 1  
file

Phải lớn hơn kích  
thước hiện tại

Dùng để chuyển file  
sang nơi lưu trữ mới

### • *Xóa bớt file*

Alter database QLDA

Remove File QLDA1

GO



# Tháo gỡ và kết gắn CSDL

## ❖ Tháo gỡ CSDL

- **EXEC sp\_detach\_db 'AdventureWorks2012'**

## ❖ Kết gắn CSDL

- **Cách 1 :**

EXEC sp\_attach\_db

@dbname = N'AdventureWorks2012',

@filename1 = N'C:\Program Files\Microsoft SQL  
Server\MSSQL11.MSSQLSERVER\MSSQL\Data\AdventureWorks  
2012\_Data.mdf',

@filename2 = N'C:\Program Files\Microsoft SQL  
Server\MSSQL11.MSSQLSERVER\MSSQL\Data\AdventureWorks  
2012\_log.ldf'

- **Cách 2 :**

CREATE DATABASE MyAdventureWorks ON  
(FILENAME = 'C:\MySQLServer\AdventureWorks\_Data.mdf'),  
(FILENAME = 'C:\MySQLServer\AdventureWorks\_Log.ldf')  
FOR ATTACH



## Nội dung trình bày

- ◆ Kiểu dữ liệu
- ◆ Kiểu dữ liệu người dùng định nghĩa
- ◆ Tạo và thay đổi Cơ sở Dữ liệu
- ◆ Tạo và thay đổi bảng dữ liệu
- ◆ Các công cụ quan sát cấu trúc dữ liệu
- ◆ Cập nhật dữ liệu



# Schema

- ❖ Khái niệm : Schema của 1 user là đơn vị luận lý chứa đựng các đối tượng dữ liệu thuộc quyền sở hữu của user ấy.
- ❖ Tạo schema :
  - **CREATE SCHEMA PDT**
  - **AUTHORIZATION PhongDaoTao**
- ❖ Xóa schema
  - **DROP SCHEMA PDT**
- ❖ Chuyển schema
  - **ALTER SCHEMA HumanResource**
  - **TRANSFER Person.Address;**

Mặc định là dbo

Chuyển bảng  
Address từ schema  
Person sang  
schema  
HumanResources



# Tạo bảng dữ liệu

## ❖ Trước khi tạo bảng

- *Xác định các cột (các thuộc tính) của bảng*
- *Xác định khóa chính*
- *Xác định các thuộc tính null/ not null*
- *Xác định thuộc tính identity (nếu có) (phải là kiểu số nguyên)*



# Tạo bảng dữ liệu

## ❖ Lưu ý:

- *Luôn tạo khóa chính cho một bảng*
- *Ràng buộc khóa ngoại nên được tạo sau khi đã tạo xong tất cả các bảng liên quan và phải tạo trước khi insert dữ liệu*
- *Tên các thuộc tính trong cùng một bảng phải phân biệt*
- *Tên các bảng trong cùng một CSDL phải phân biệt*



## Tạo bảng dữ liệu

```
Create Table [ DBName.[ owner .] ]  
    Tabname  
( {  
    < col_definition >  
    | col_name AS computed_col_expression  
    } [ ,...n ]  
    | < table_constraint >  
)
```



# Tạo bảng dữ liệu

Create Table [ DBName.[ owner .] ]

Tabname

( {  
    < col\_definition >  
    | col\_name AS computed\_col\_expression  
} [ ,...n ]  
| < table\_constraint >  
)

Column datatype  
[Default DefValue |  
Identity(seed,inc)] |  
[Column constraint]

[Null | Not null] |  
[{Primary key | Unique} [Cluster | Noncluster] ] |  
[Foreign key references ref\_tab (Ref\_col)  
On delete {cascade | No action} On update {cascade | No action}] |  
Check (logical Expression)



# Tạo bảng dữ liệu

Create Table [ DBName.[ owner .] ]

Tabname

( {  
 < col\_definition >  
 | col\_name AS compute  
 } [ ,...n ]  
 | < table\_constraint >  
 )

```
[Constraint Cnstr_name]
{
  [
    {Primary key | unique}
    [clustered | nonclustered]
    (col [asc | desc] [...,n])
  ]
  |
  Foreign key
  (col [...,n]) references ref_tab
  (ref_col [...,n])
  On delete {cascade | No action}
  On update {cascade | No action}
  |
  Check (logical Exp)
}
```



# Tạo bảng dữ liệu

```
Create Table [ DBName.[ owner .] ]  
    Tabname  
( {  
    < col_definition >  
    | col_name AS computed_col_expression  
    } [ ,...n ]  
    | < table_constraint >  
)
```

Biểu thức giá trị cho cột dữ liệu ảo, không phải một cột vật lý. Giá trị của nó được tính toán từ giá trị của các cột còn lại



## Tạo bảng dữ liệu (Ví dụ 3)

Create table QuanLyHangHoa.MatHang

(

MaMH varchar(10) not null,

TenMN nvarchar(50) unique,

DonGia float default 0,

LoaiHang varchar(10),

constraint PKMH Primary key (MaMH),

constraint FKHH foreign key (LoaiHang)  
references LoaiHang(MaLH),

constraint ck\_DonGia check (DonGia > 0)

)

Column  
constraint

Col\_definition

Tab\_constraint



## Sửa bảng dữ liệu

- ◆ Thêm thuộc tính :  
*Alter table SinhVien*  
*Add Phai nvarchar(5) not null default N'Nữ'*
- ◆ Sửa thuộc tính : (Giả sử thuộc tính SinhVien.DiaChi đang là varchar(50), cần sửa thành nvarchar(50) để hỗ trợ Unicode)  
*Alter table SinhVien*  
*Alter column DiaChi nvarchar(50)*



## Nội dung trình bày

- ◆ Kiểu dữ liệu
- ◆ Kiểu dữ liệu người dùng định nghĩa
- ◆ Tạo và thay đổi Cơ sở Dữ liệu
- ◆ Tạo và thay đổi bảng dữ liệu
- ◆ Các công cụ quan sát cấu trúc dữ liệu
- ◆ Cập nhật dữ liệu



## Các công cụ quan sát cấu trúc DL

- ❖ Thông tin về các bảng, các ràng buộc được lưu trong bảng hệ thống sysobjects
- ❖ Ví dụ: Đọc thông tin về các bảng người dùng định nghĩa (không phải bảng hệ thống) trong database hiện hành:
- ❖ `Select * from sys.objects where type = 'U'`

User-defined



# Các công cụ quan sát cấu trúc DL

## ◆ Exec sp\_databases

- *Cho biết danh sách các CSDL hiện có cùng với kích thước của chúng (tính bằng KB)*

## ◆ EXEC sp\_tables

[@table\_name=]SinhVien,  
[@table\_owner=] ws201,  
[@table\_qualifier=] QuanLySinhVien,  
[@table\_type=] "TABLE"



# Các công cụ quan sát cấu trúc DL

## ◆ Exec sp\_databases

- **Cho biết danh sách các CSDL hiện có trong hệ thống, dung lượng của chúng (tính bằng KB)**

## ◆ EXEC sp\_tables

[@table\_name=] SinhVien,

[@table\_owner=] ws201,

[@table\_qualifier=] QuanLySinhVien

[@table\_type=] "TABLE"

Cho danh sách các table trong hệ thống, các tham số là không bắt buộc và dùng để lọc table. Càng chỉ định nhiều tham số, kết quả cho ra càng ít

Tên table cần xem

Tên user tạo table ấy

Tên DB chứa table ấy

Loại table (giữa cặp “ ”), là một trong các loại : TABLE, SYSTEM TABLE, VIEW



# Các công cụ quan sát cấu trúc DL

## ◆ Exec sp\_help [Object\_name]

- *Cung cấp thông tin cơ bản của các đối tượng bất kỳ lưu (trong bảng sysobjects) hay các kiểu dữ liệu người dùng định nghĩa (lưu trong bảng systypes)*

## ◆ Exec sp\_column

[@Table\_name=]Tab\_name

[@Column\_name=]Col\_name

- *Cung cấp thông tin về tất cả (hay chỉ một – khi có chỉ định Col\_name) cột trong một table có tên được chỉ định bởi Tab\_name*



# Các công cụ quan sát cấu trúc DL

## ◆ Exec sp\_helpconstraint Tab\_name

- *Cho biết thông tin về các constraints trên table chỉ định với Tab\_name*
- *Với constraint loại foreignkey, default và check, nó cung cấp cả biểu thức định nghĩa constraint.*



## Nội dung trình bày

- ◆ Kiểu dữ liệu
- ◆ Kiểu dữ liệu người dùng định nghĩa
- ◆ Tạo và thay đổi Cơ sở Dữ liệu
- ◆ Tạo và thay đổi bảng dữ liệu
- ◆ Các công cụ quan sát cấu trúc dữ liệu
- ◆ Cập nhật dữ liệu



# Cập nhật dữ liệu

## ❖ Chú ý khi cập nhật (ins / del / upd)

- *Dữ liệu nhập phải phù hợp với kiểu dữ liệu*
- *Đảm bảo tính nhất quán dữ liệu (tuân thủ các ràng buộc toàn vẹn trên CSDL hiện hành)*
- *Định dạng giá trị kiểu chuỗi unicode. Với kiểu datetime, lưu ý tình trạng hệ thống hiện đang chấp nhận 'd-m-y' hay 'm-d-y' hay...*
- *Chú ý giá trị rỗng (null)*



# Cập nhật dữ liệu

## ◆ Thêm dữ liệu :

- ***Thêm từng dòng dữ liệu vào bảng***

```
Insert [into] Table_name  
[ (column_name[,...n] )]  
values ( value [...n] )
```

- ***Thêm 0-n dòng dữ liệu từ bảng khác***

```
Insert [into] Table_name  
Select_statement
```



# Cập nhật dữ liệu

- *Lưu ý: Nếu thêm 0-n dòng DL bằng câu select, ta có thể đọc dữ liệu từ các bảng trong database khác. Khi đó, tên bảng được viết đầy đủ như sau:*
  - Database\_name.Owner.Table\_name
- *Ví dụ:*
  - Insert into SV select \* from QLSinhVien.dbo.SinhVien



# Cập nhật dữ liệu

## ❖ Sửa dữ liệu :

***Update*** Table\_name set

***[ (column\_name = new\_value[,...n] )]***

***[where (update condition)]***

### • Ví dụ :

Update HangHoa set

DonGia = DonGia\*110/100

Where LoaiHang = 'Milk'



# Cập nhật dữ liệu

## ❖ Xoá dữ liệu

***Delete [from] Table\_name  
[where (delete condition)]***

### • Ví dụ :

Delete from SinhVien

Where DTB < 3.0



# Cập nhật dữ liệu

## ❖ Chỉ thị OUTPUT

- *Update HangHoa set*  
*DonGia = DonGia\*110/100*  
*output deleted.DonGia, inserted.dongia*  
*where LoaiHang = 'Milk'*
- *Insert into LoaiDiaOc (maLoai,tenLoai)*  
*output inserted.tenLoai*  
*values(15, N'biệt thự')*
- *Delete from HangHoa*  
*output deleted.DonGia*  
*where DonGia < 100000*

Chỉ có với  
phiên  
Bản SQL  
Server 2005  
Trở lên



# Cập nhật dữ liệu

## ❖ Chỉ thị OUTPUT

- *Create table HH (  
MaHH varchar(10) primary key, DonGia float)*
- *Update HangHoa set  
DonGia = DonGia\*110/100  
output deleted.MaHg, deleted.DonGia into HH  
where LoaiHang = 'Milk'*
- *Delete from HangHoa  
output deleted.MaHg, deleted.DonGia into HH  
where DonGia < 100000*



# Cập nhật dữ liệu

## ◆ Chỉ thị TOP

- Update **Top(@n)** HangHoa set  $DonGia = DonGia * 110/100$  where  $LoaiHang = 'Milk'$
- Insert **Top(@n)** into LoaiDiaOc (maLoai,tenLoai)  
 $select\ maLoai, tenLoai\ from\ LoaiDiaOc2$
- Delete **Top(@n)** from HangHoa where  $DonGia < 100000$

Có thể là biến, hằng hay câu truy vấn

Liên hệ chỉ thị Top n trong câu select

Chỉ có với phiên  
Bản SQL  
Server 2005  
Trở lên



## Hết chương II

