

# SQL SERVER



# Chương 1 :GIỚI THIỆU TỔNG QUAN VỀ SQL SERVER

## Tổng quan :

**SQL Server là một hệ quản trị CSDL quan hệ**

### Cơ Sở Dữ Liệu Quan Hệ :

- **Dữ liệu được tổ chức lưu trữ thành từng bảng**
- **Một bảng dùng để lưu trữ thông tin của một đối tượng ngoài thế giới thực**
- **Một bảng được tạo từ dòng và cột**

## I. Tổng Quan : (tt)

➤ **Cột :Dùng để lưu trữ thuộc tính của đối tượng**

➤ **Một bảng phải có tối thiểu một cột,Thứ tự cột không quan trọng**

➤ **Một cột có ba thuộc tính cơ bản**

✓ **Column Name :Tên cột,trong một bảng không thể có hai cột trùng tên**

✓ **Data Type :Kiểu dữ liệu**

✓ **Length :chiều dài của dữ liệu**

# I:Tổng Quan(tt)

- **Dòng :tập hợp dữ liệu của nhiều cột**
- **Trong một bảng không thể có hai dòng trùng thông tin**
- **Một dòng cho biết thông tin cụ thể của một đối tượng lưu trữ trong bảng**
- **Một bảng có thể không có dòng nào**

# I.Tổng Quan(tt)

## ➤ Khoá Chính (Primary Key) :

- ✓ Tập hợp của một hay nhiều cột trong bảng
- ✓ Dữ liệu tại cột này khác rỗng
- ✓ Dữ liệu tại cột này là duy nhất trong bảng

# I.Tổng Quan (tt)

## ➤ Khoá ngoại (Foreign Key) :

- ✓ Tập hợp của một hay nhiều cột trong bảng, mà các cột này là khoá chính của bảng khác
- ✓ Dữ liệu tại các cột này hoặc rỗng, hoặc phải lấy từ bảng mà cột làm khoá chính

## I.Tổng Quan(tt)

### B.Hệ Quản Trị CSDL Quan Hệ :

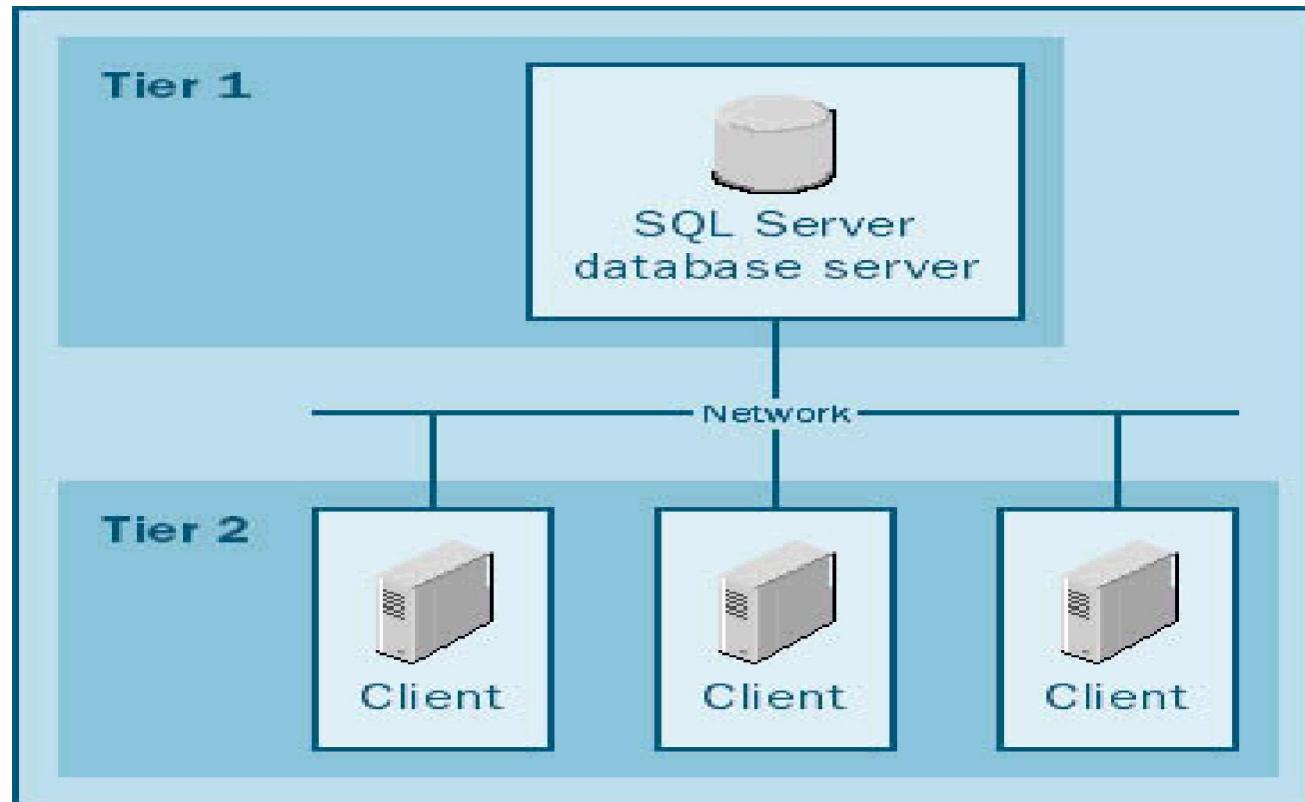
- Là một hệ thống các chương trình có các chức năng
  - ✓ Tạo và quản lý một CSDL Quan hệ
  - ✓ khai thác Một CSDL Quan hệ
  - ✓ Bảo mật Một CSDL Quan hệ
  - ✓ Tạo bảng sao (Backup) Một CSDL Quan hệ
  - ✓ Phục hồi(Restore) Một CSDL Quan hệ
- SQL Server là một hệ quản trị CSDL Quan hệ

# Các hệ thống SQL Server

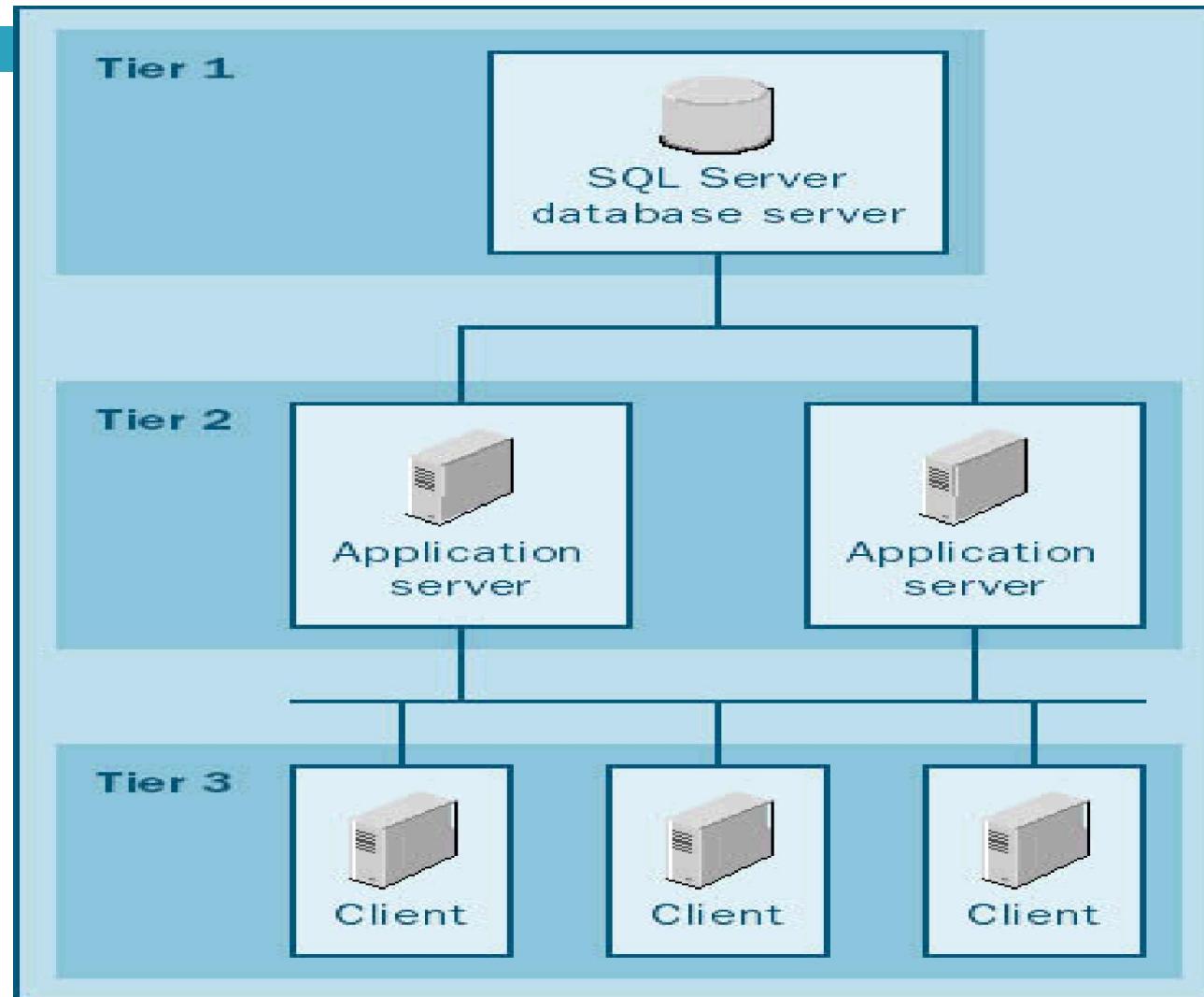
- Hệ Thống SQL Server có thể triển khai cho một hệ thống kiểu Client-Server hoặc triển khai cho một hệ thống máy đơn
- Hệ thống được triển khai phụ thuộc vào số người truy cập đồng thời đến CSDL để thực hiện một công việc

# Các hệ thống SQL Server

Trong mô hình Client-Server ta có thể triển khai theo mô hình 2 lớp hoặc 3 lớp



# Các hệ thống SQL Server



## II.Cài Đặt SQL Server

### A.Yêu cầu phần cứng :

Phần cứng	Yêu cầu
CPU	Từ Pentium 166 MHz trở lên
RAM	Từ 64MB trở lên
Đĩa cứng	Còn trống 100MB trở lên
Màn hình	VGA có độ phân giải 800X600 trở lên

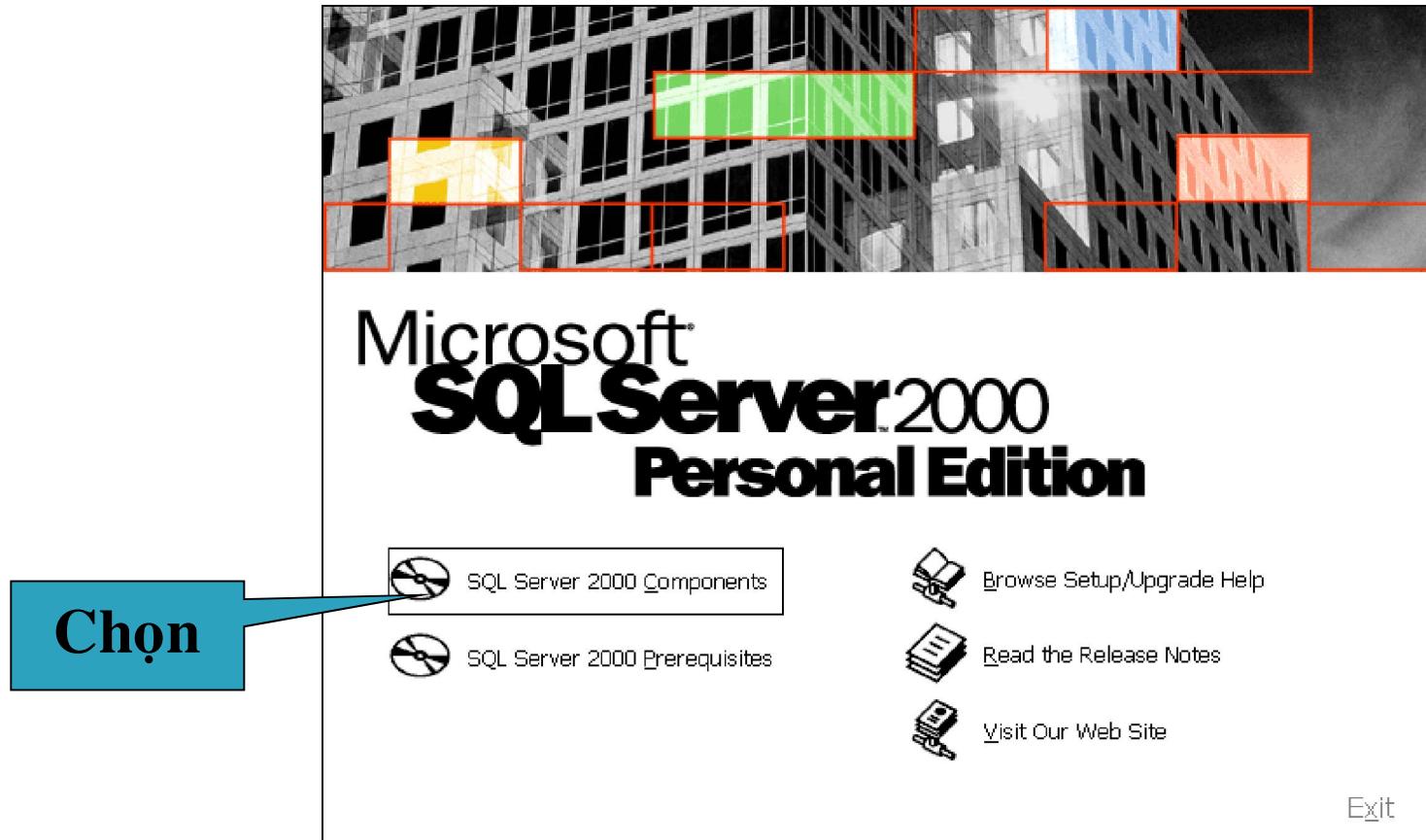
## II.Cài Đặt SQL Server

### A.Yêu cầu phần Mềm:

Phiên bản SQL Server	Hệ Điều Hành
➤ Enterprise Edition ➤ Standard Edition	Microsoft Windows Server(nt,2000,Xp Server)
➤ Personal Edition	Có thể chạy trên tất cả các hệ điều hành

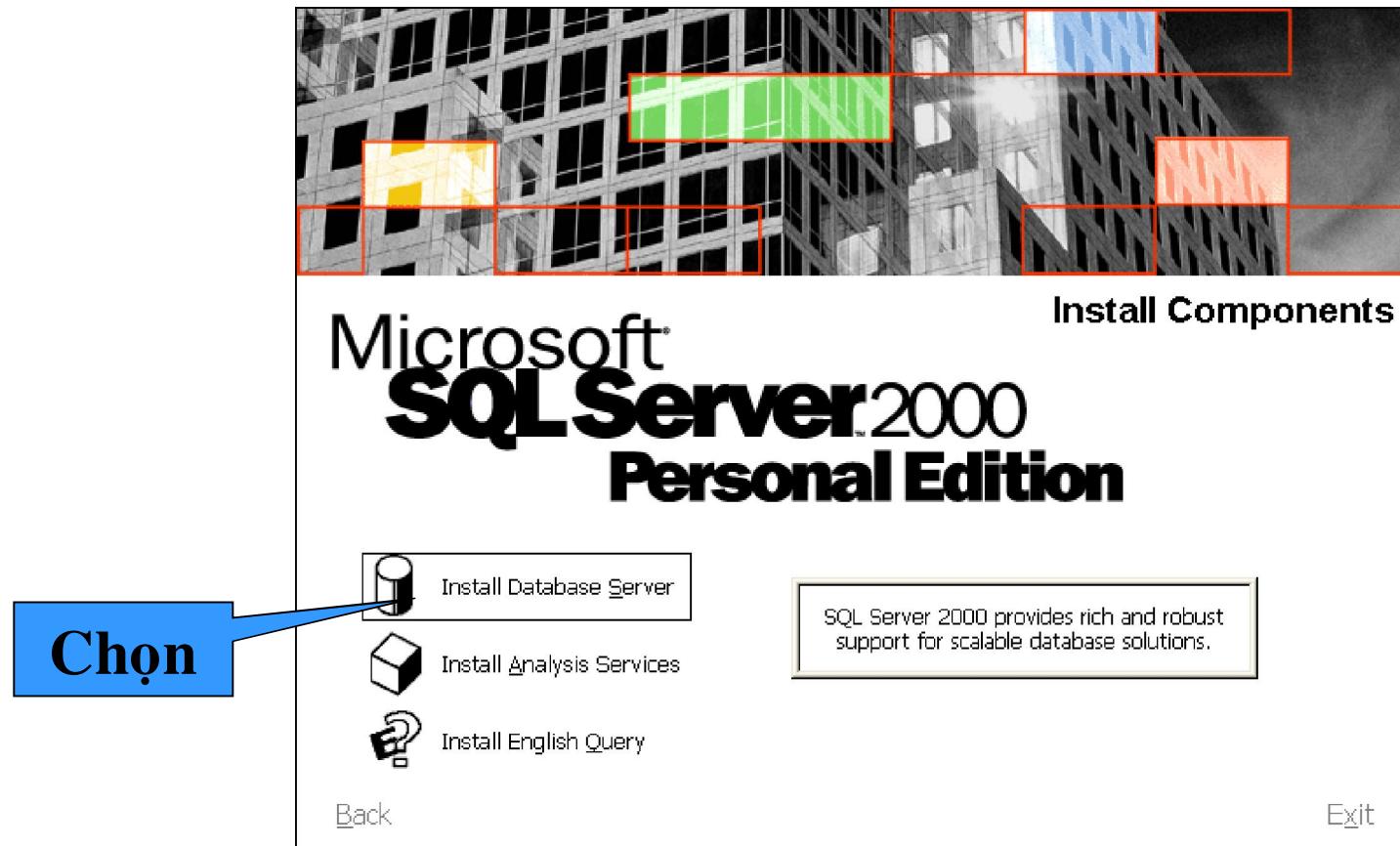
## II. Cài đặt Personal Edition SQL Server

### 1. Đưa đĩa CD chương trình vào ,màn hình sau xuất hiện



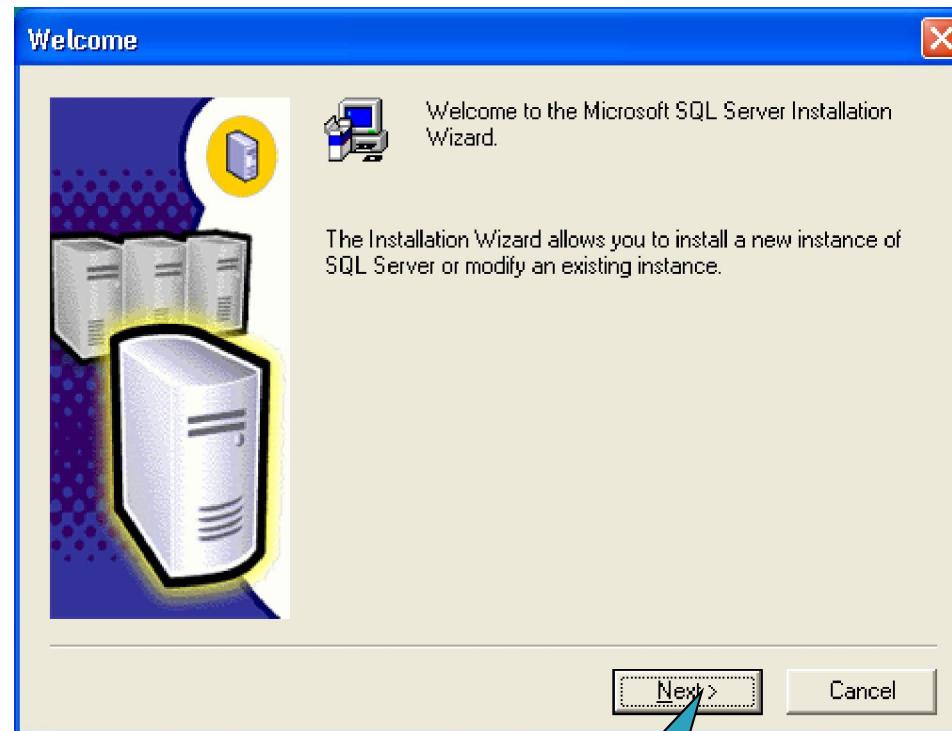
## II.Cài đặt Personal Edition SQL Server(tt)

### 2.Màn hình sau xuất hiện



## II.Cài đặt Personal Edition SQL Server(tt)

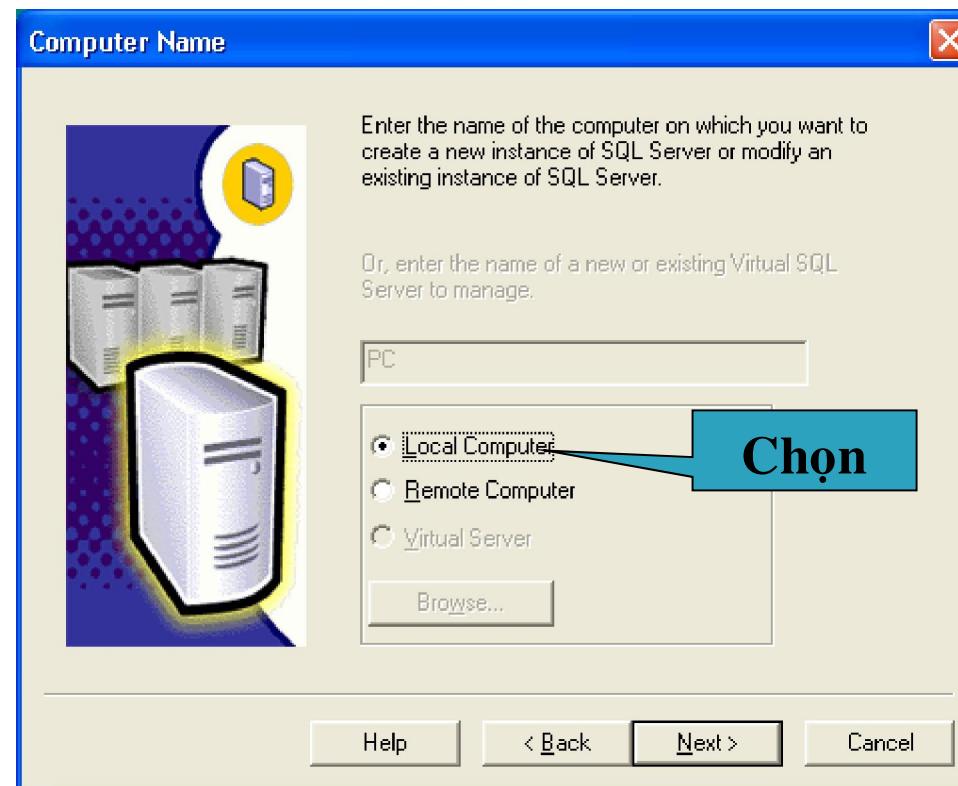
### 3.Màn hình sau xuất hiện



Chọn

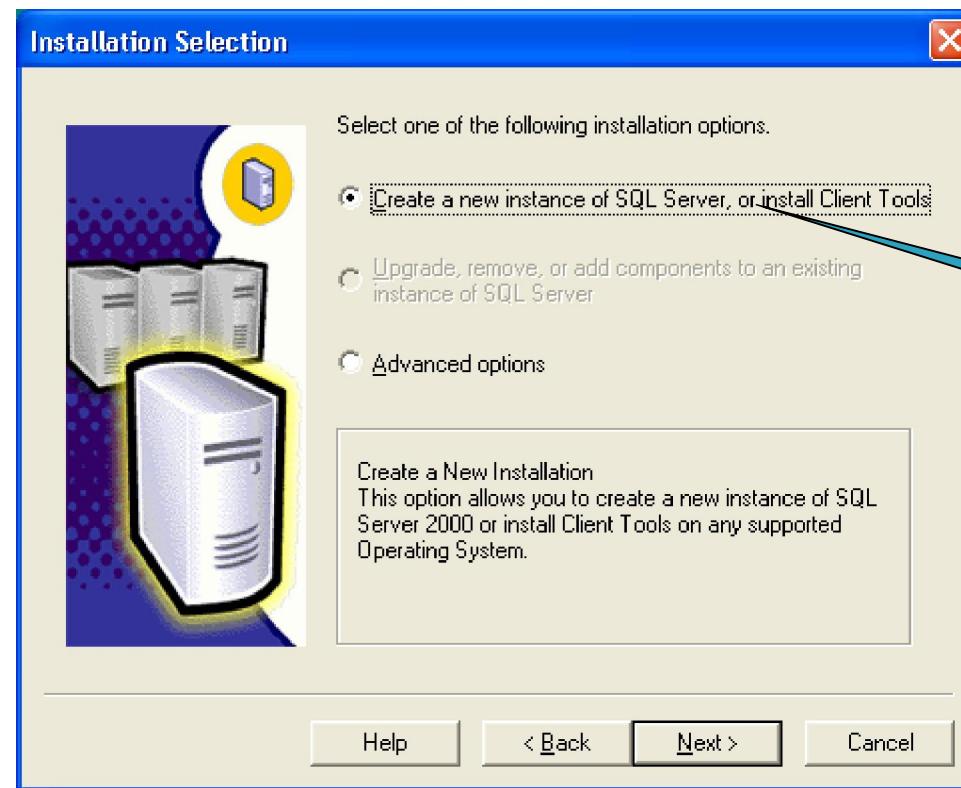
## II.Cài đặt Personal Edition SQL Server(tt)

### 4.Màn hình sau xuất hiện



## II.Cài đặt Personal Edition SQL Server(tt)

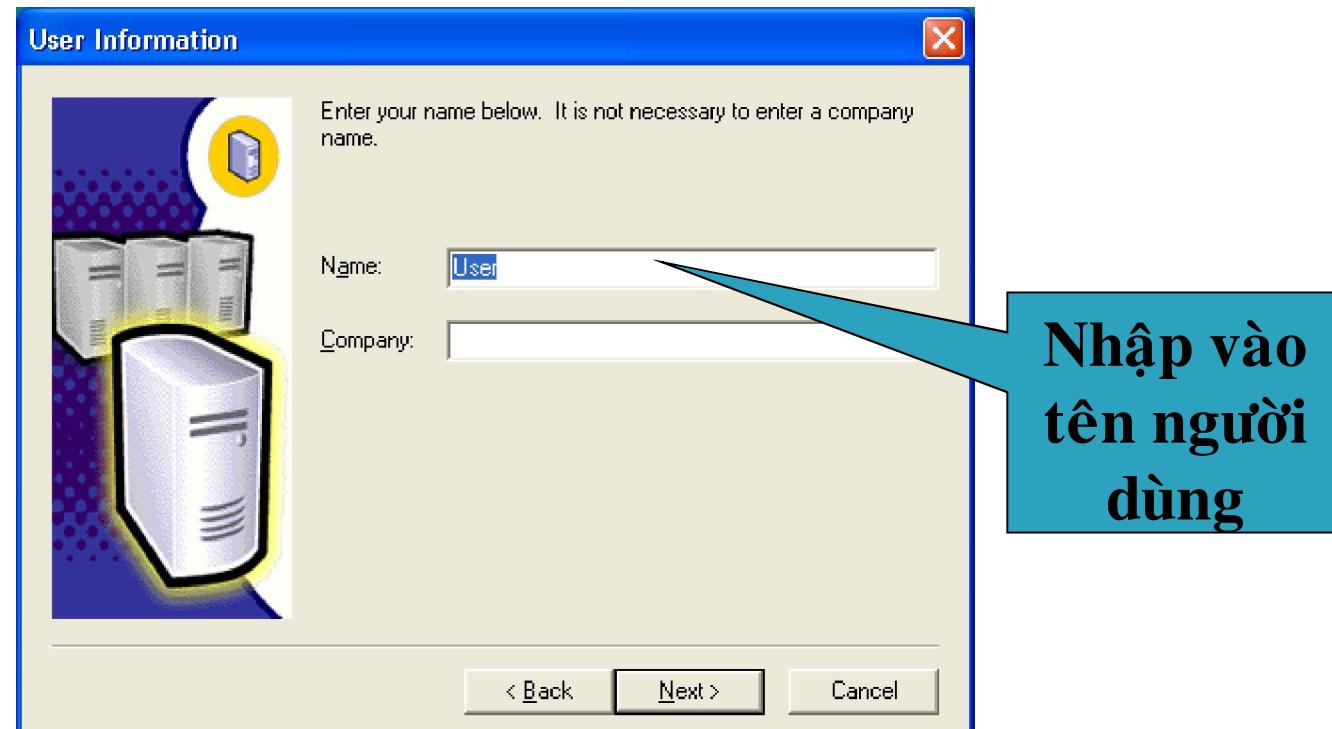
### 5.Màn hình sau xuất hiện



Chọn

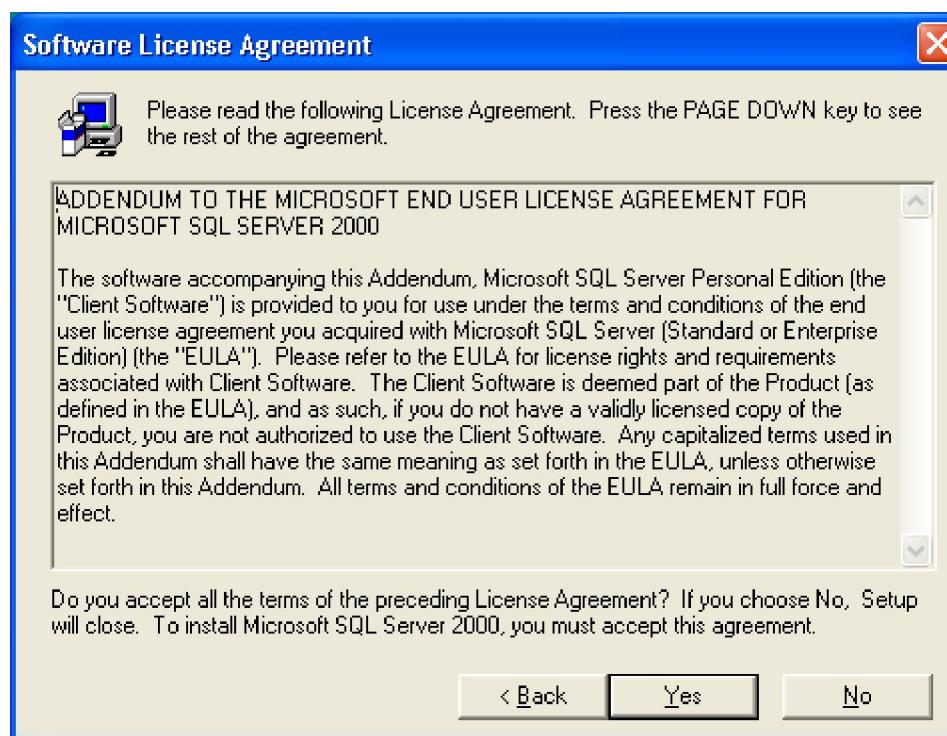
## II.Cài đặt Personal Edition SQL Server(tt)

### 6.Màn hình sau xuất hiện



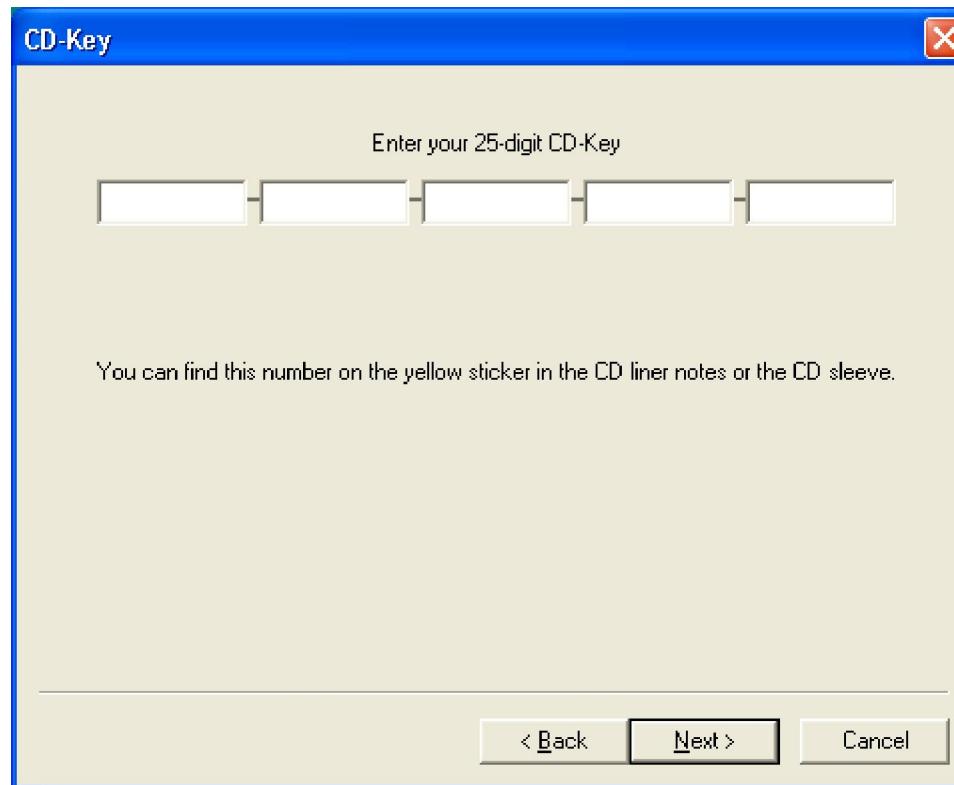
## II.Cài đặt Personal Edition SQL Server(tt)

### 7.Màn hình cảnh báo xuất hiện, chọn yes



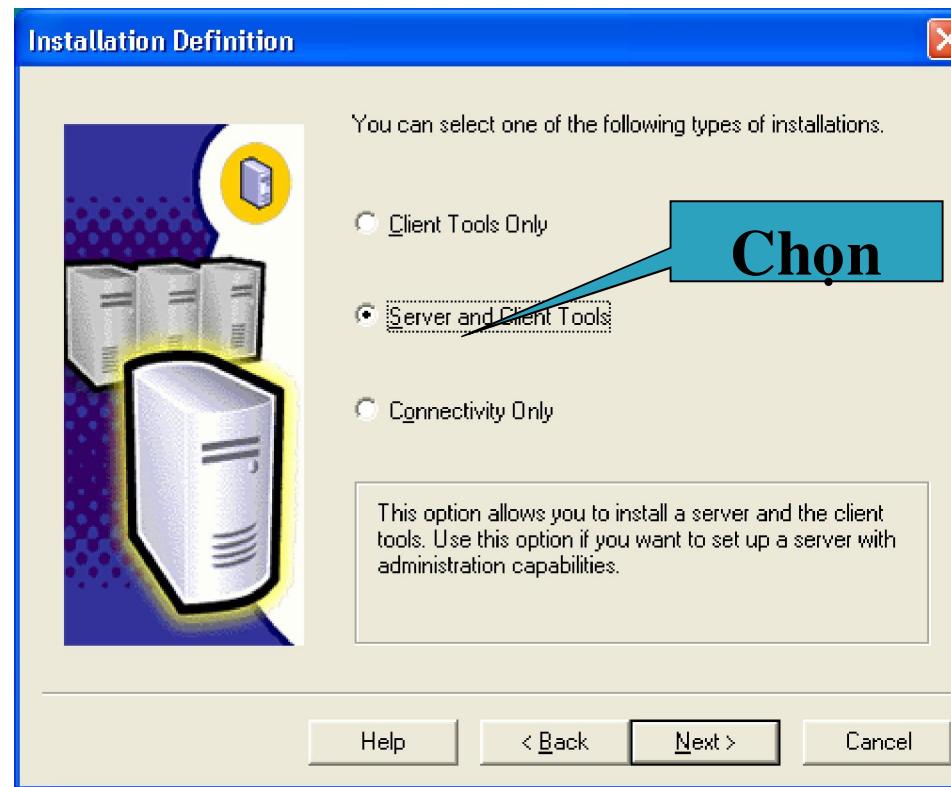
## II.Cài đặt Personal Edition SQL Server(tt)

### 8.Màn hình đăng nhập xuất hiện ,nhập vào CD Key



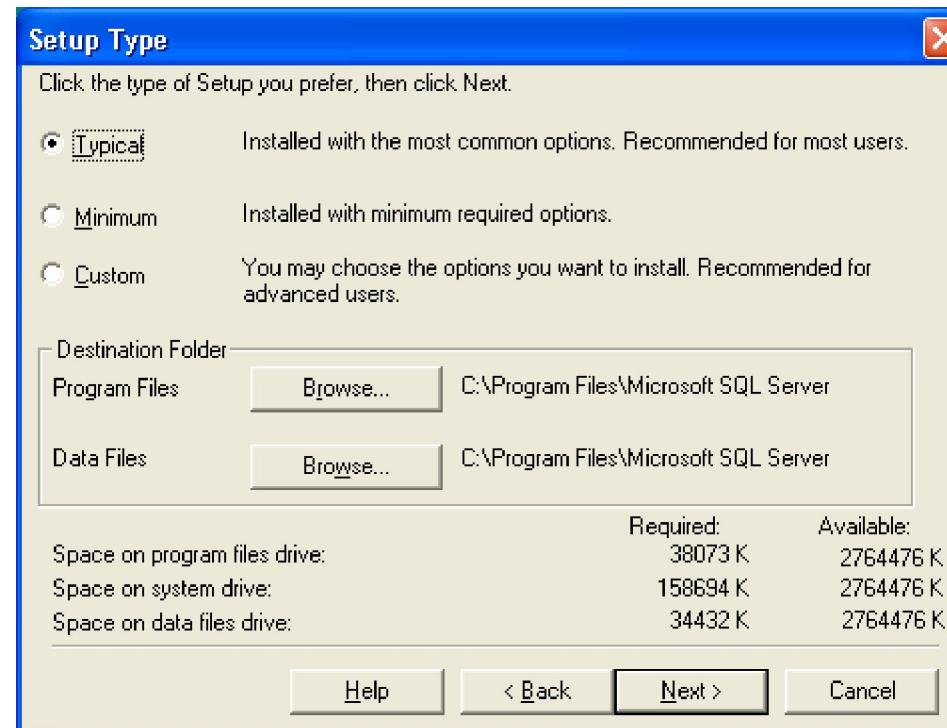
## II.Cài đặt Personal Edition SQL Server(tt)

### 9.Chọn kiểu cài đặt



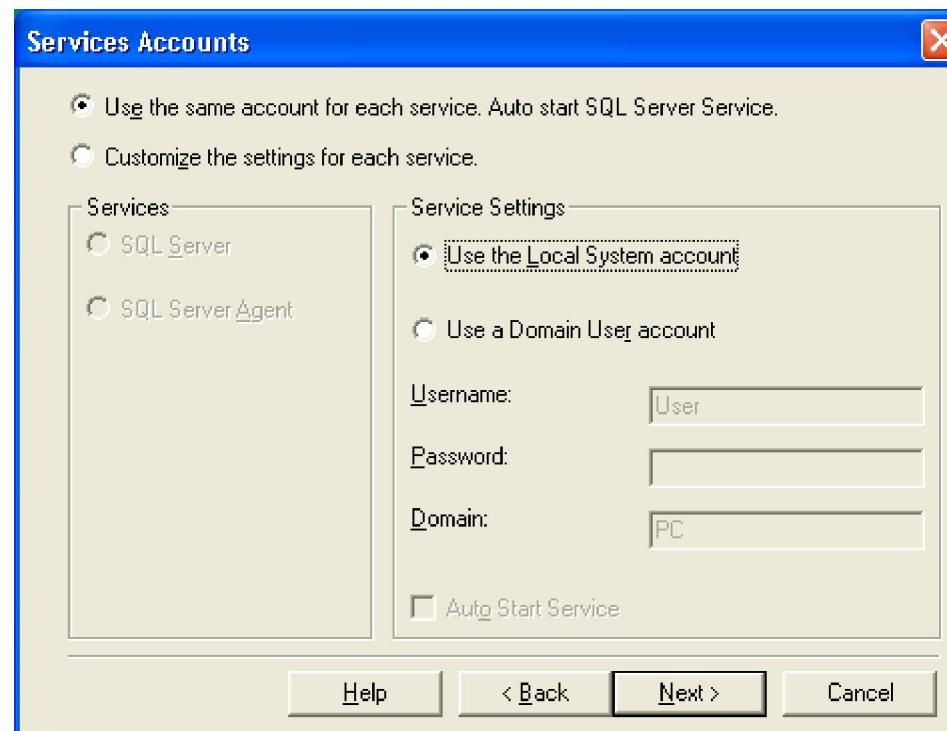
## II.Cài đặt Personal Edition SQL Server(tt)

### 10.Chọn Dạng cài đặt



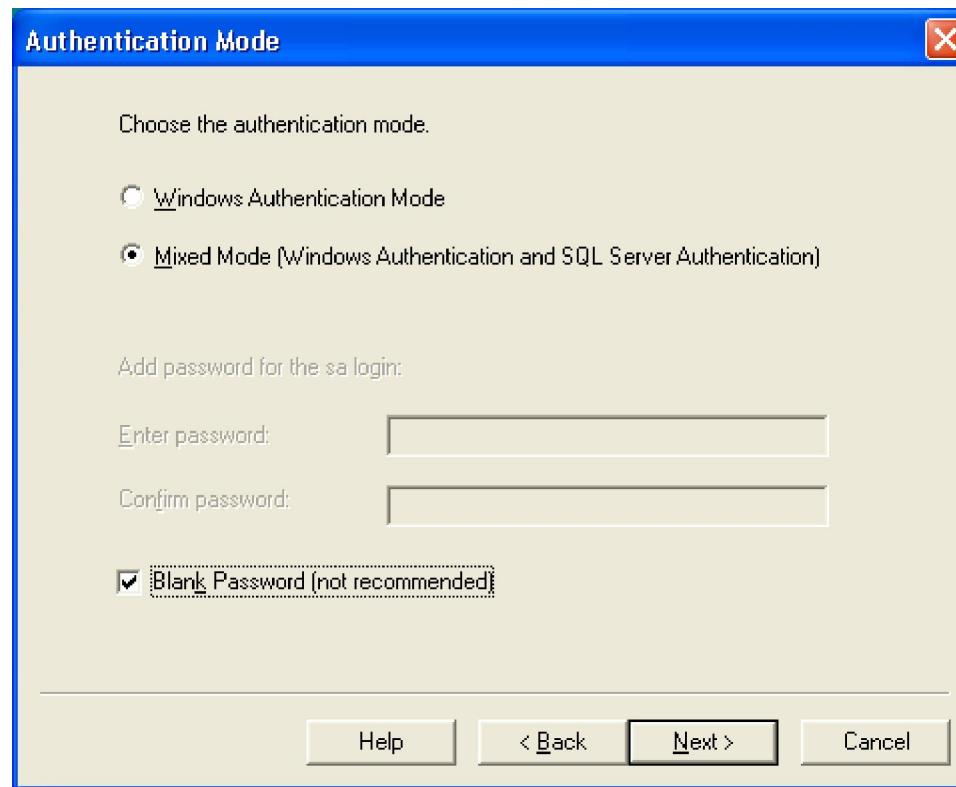
## II.Cài đặt Personal Edition SQL Server(tt)

### 11.Chọn kiểu đăng nhập vào SQL Server



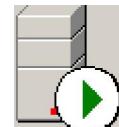
## II.Cài đặt Personal Edition SQL Server(tt)

### 12.Chọn kiểu đăng nhập vào SQL Server



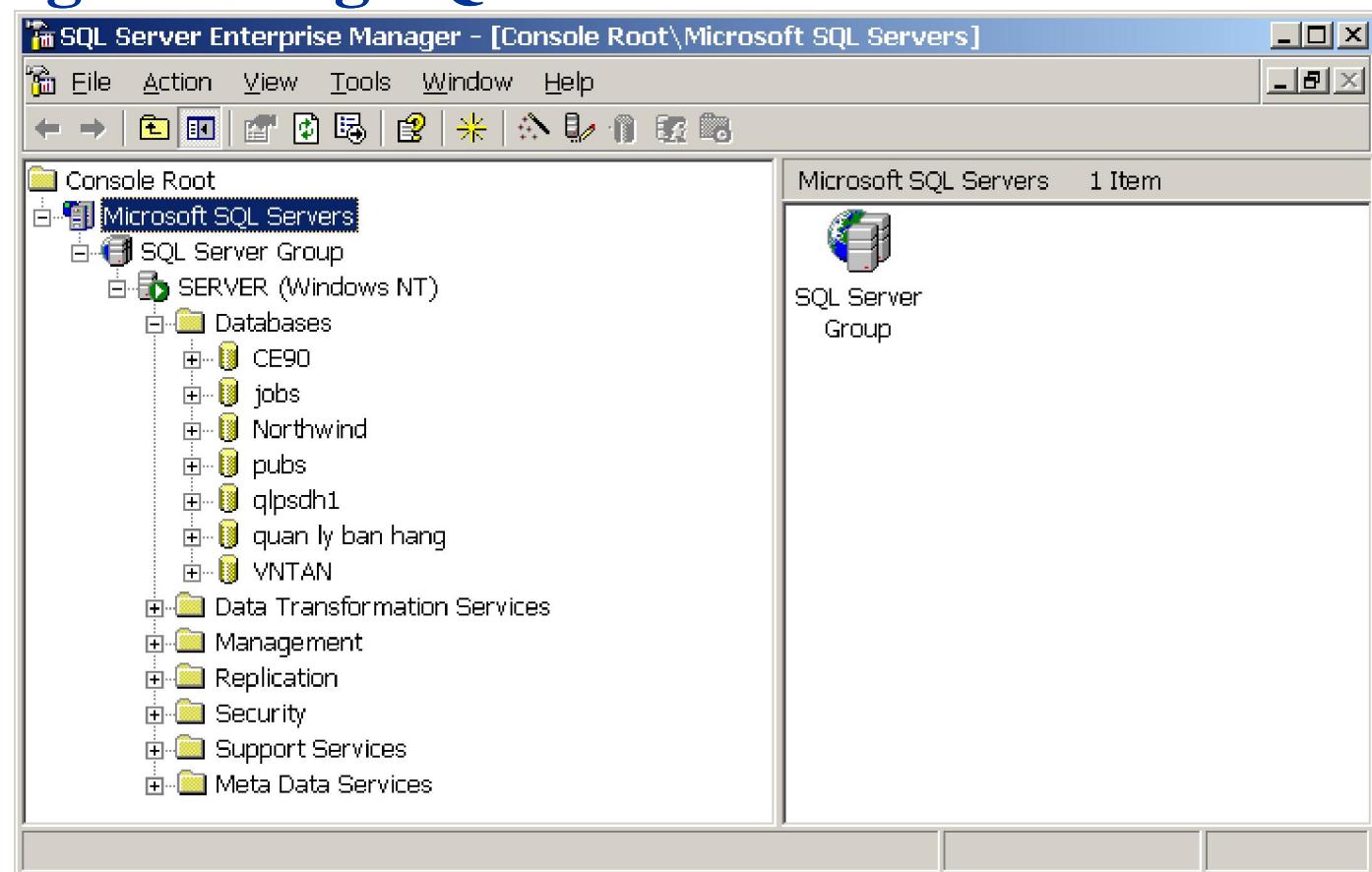
## II.Cài đặt Personal Edition SQL Server(tt)

- Nhấp Next để chương trình cài đặt vào máy
- Sau khi cài đặt xong, khởi động lại máy, sau khi khởi động xong, thấy xuất hiện biểu tượng Trên thanh Status của Windows



### III.Các tiện ích của SQL Server

#### 1.Tiện ích EnterPriseManager:Màn hình quản lý các đối tượng có trong SQL Server



### III.Các tiện ích của SQL Server

#### A> Microsoft SQL Server:Chứa các nhóm Server

- Tạo mới một nhóm Server ta làm như sau :
- Click chuột phải vào Microsoft SQL Server, chọn New SQL Server Group



### III.Các tiện ích của SQL Server

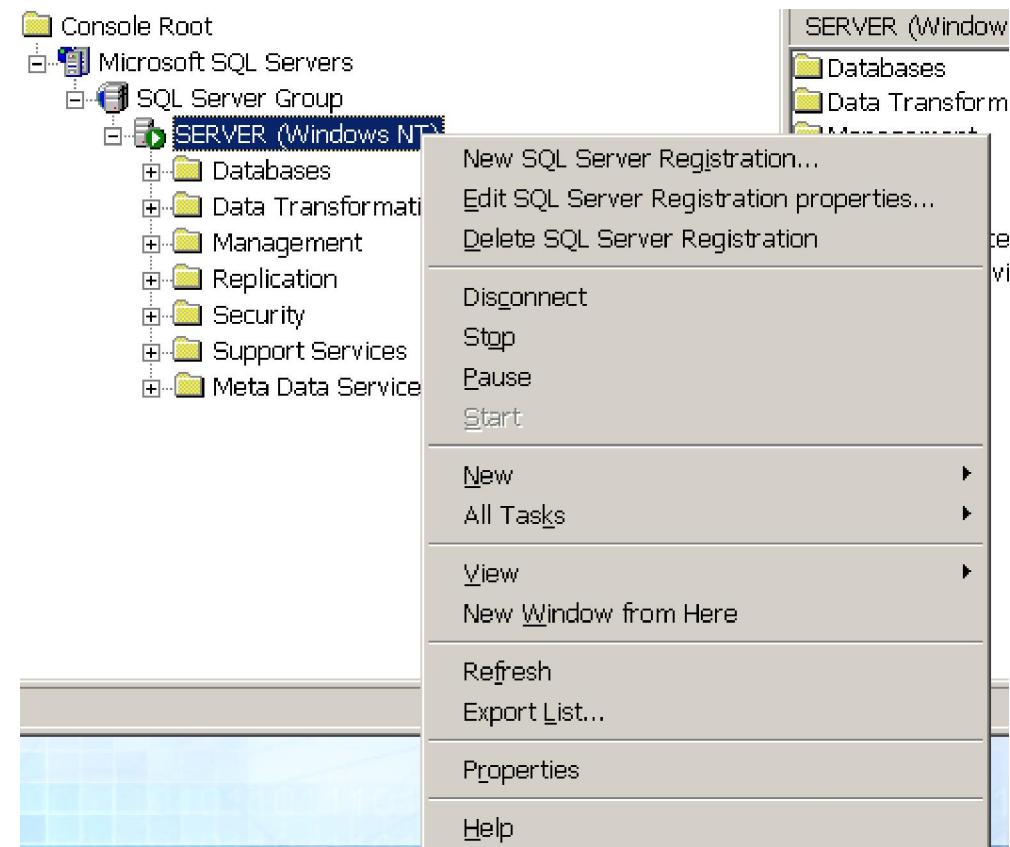
#### Màn hình sau xuất hiện

- Chọn Top Level Group : Tạo một nhóm gốc
- Chọn Sub-group of: tạo một nhóm là con của một nhóm khác
- Name:tên nhóm



### III.Các tiện ích của SQL Server

B>Đăng ký một Server vào hệ thống :Click chuột phải vào SQL Server Group, chọn New SQL Server Registration



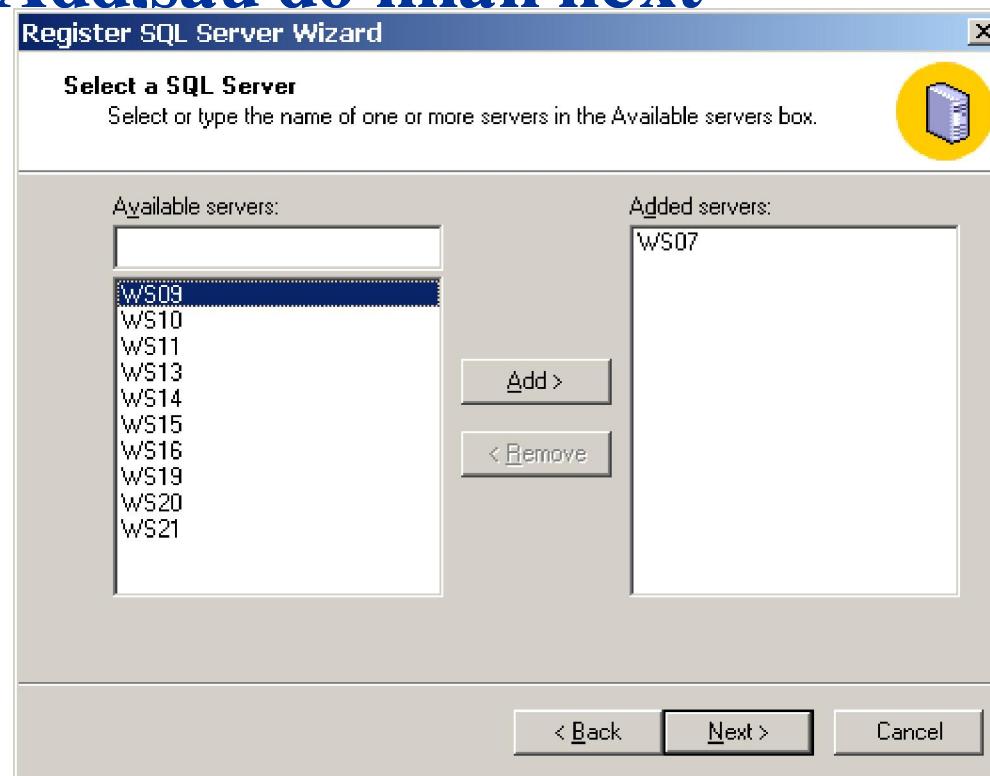
### III.Các tiện ích của SQL Server

#### Màn hình sau xuất hiện : Nhấn Next



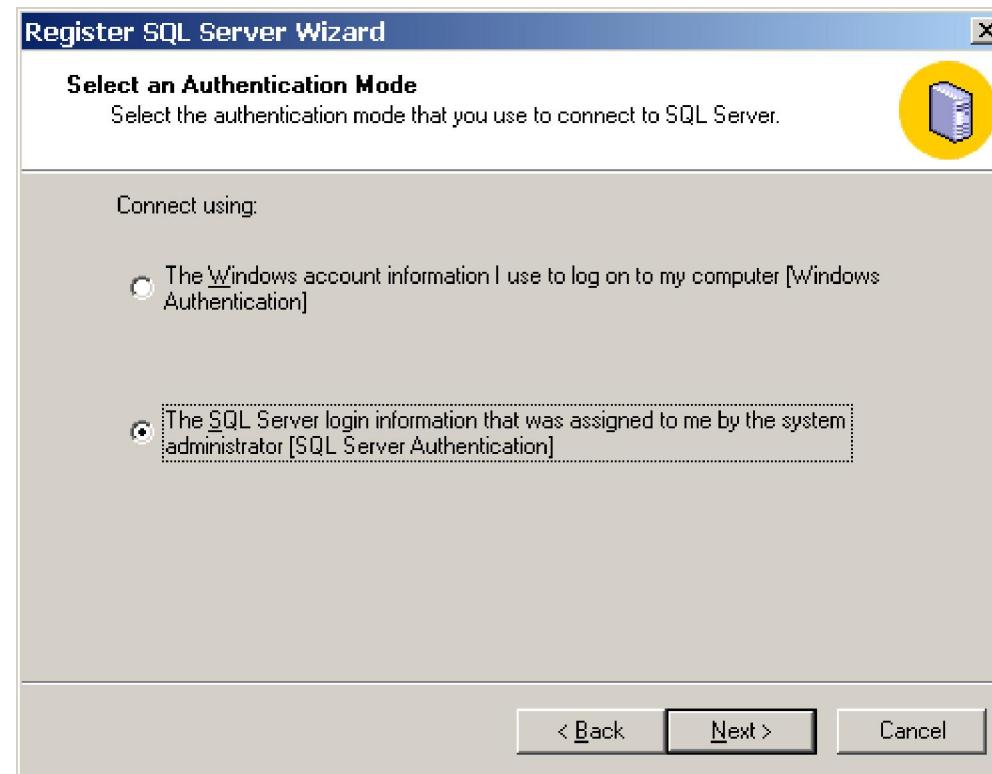
### III.Các tiện ích của SQL Server

**Màn hình sau xuất hiện : chọn Server cần đăng ký,nhấn Add.sau đó nhấn next**



### III.Các tiện ích của SQL Server

#### Màn hình sau xuất hiện :Màn hình sau xuất hiện chọn dạng kết nối ,chọn lựa chọn 1:



### III.Các tiện ích của SQL Server

#### Màn hình sau xuất hiện :Nhập Login name,PassWord để đăng nhập vào SQL Server

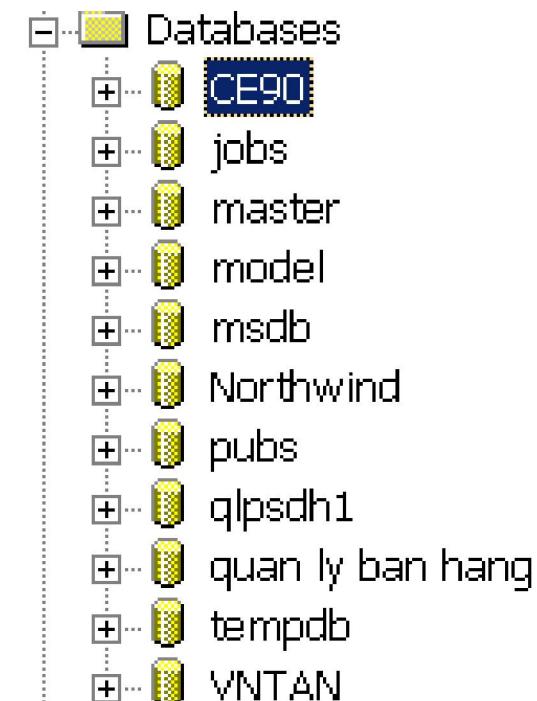
- **Chú ý : Nếu quá trình cài đặt ở bước 11 và 12 chọn như trên thì**
- **Login name :Sa**
- **Password :**



### III.Các tiện ích của SQL Server

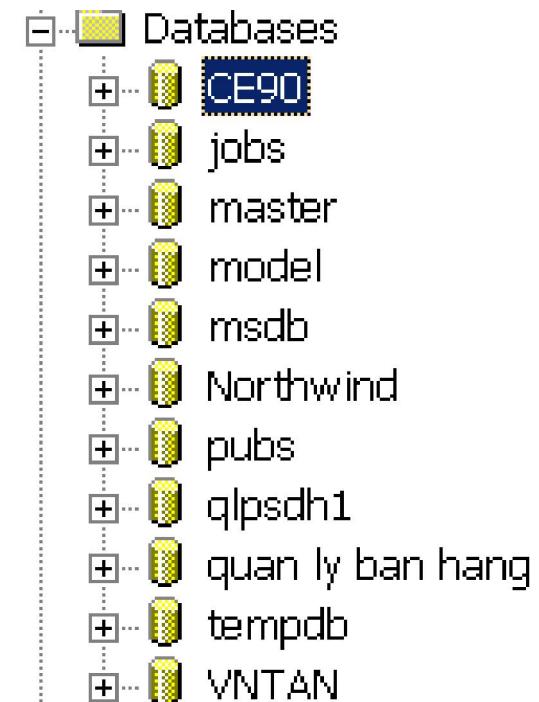
C>DataBases:Chứa các CSDL Trong SQL Server,  
có 7 CSDL có sẵn ,chia làm 2 loại

CSDL Hệ thống :không được xoá,sửa	
Tên CSDL	Ý nghĩa
Master	Là CSDL quản lý thông tin của các CSDL còn lại
Mode	CSDL mẫu
Msdb	CSDL phục vụ cho dịch vụ SQL Server Agent



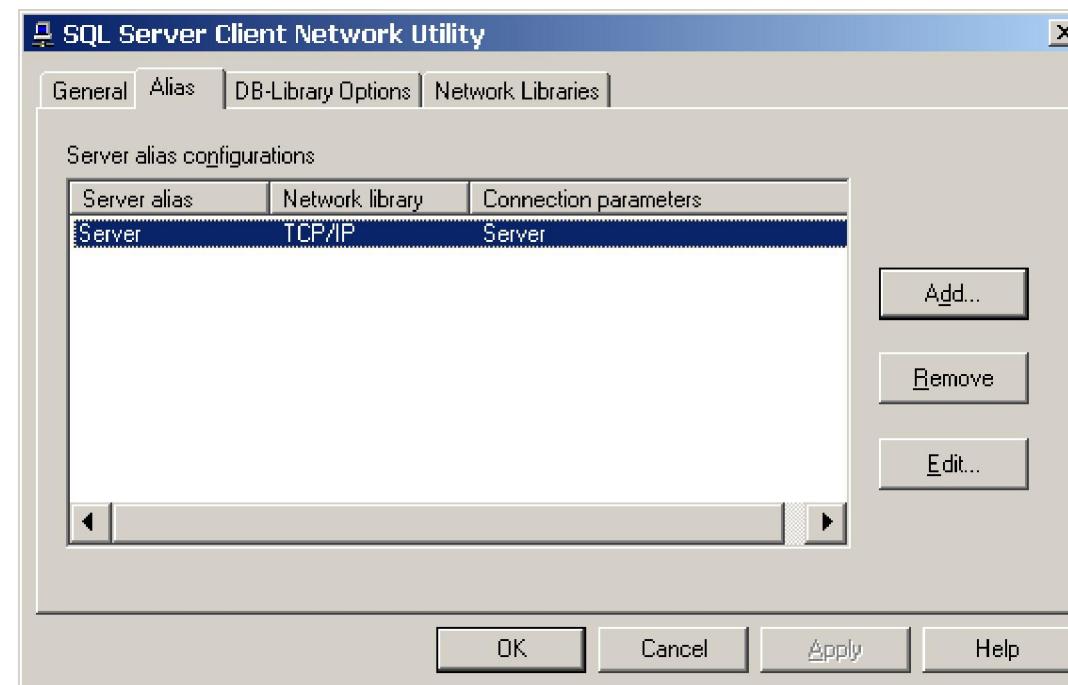
### III.Các tiện ích của SQL Server

<b>Tempdb</b>	Là CSDL chứa các bảng tạm được phát sinh trong quá trình xử lý
<b>CSDL mẫu</b>	
<b>Tên CSDL</b>	<b>Ý nghĩa</b>
<b>Jobs</b>	
<b>Northwind</b>	
<b>pubs</b>	



### III.Các tiện ích của SQL Server

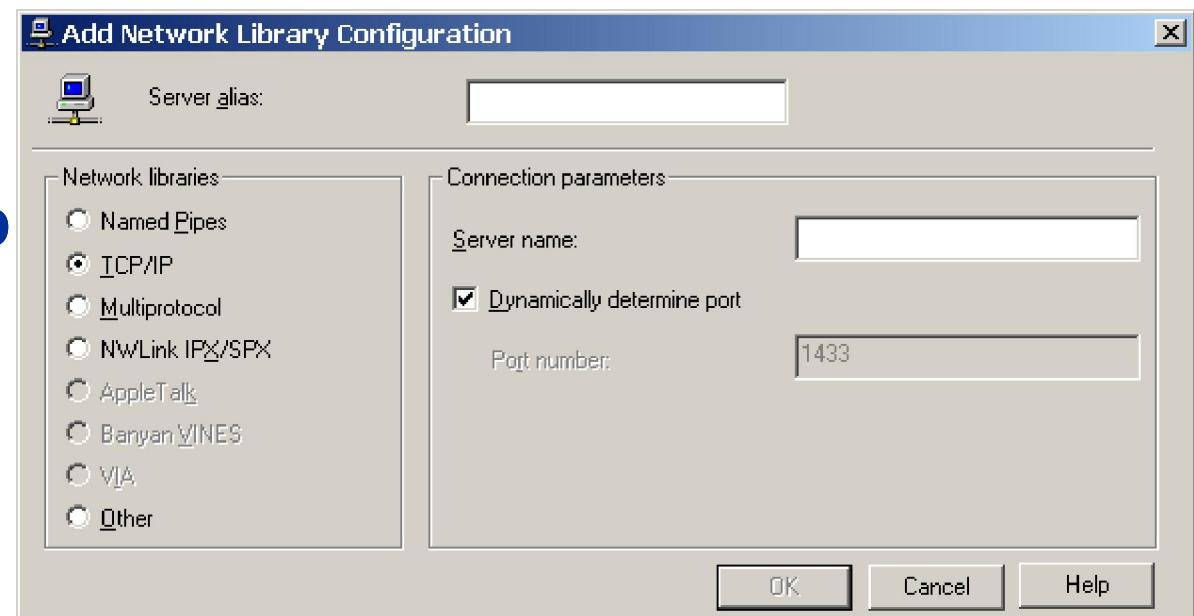
C>**Client Network Utility : dùng để đăng ký một Server cho hệ thống**



### III.Các tiện ích của SQL Server

#### Cách đăng ký : chọn Tab Alias nhấn nút Add,màn hình sau xuất hiện

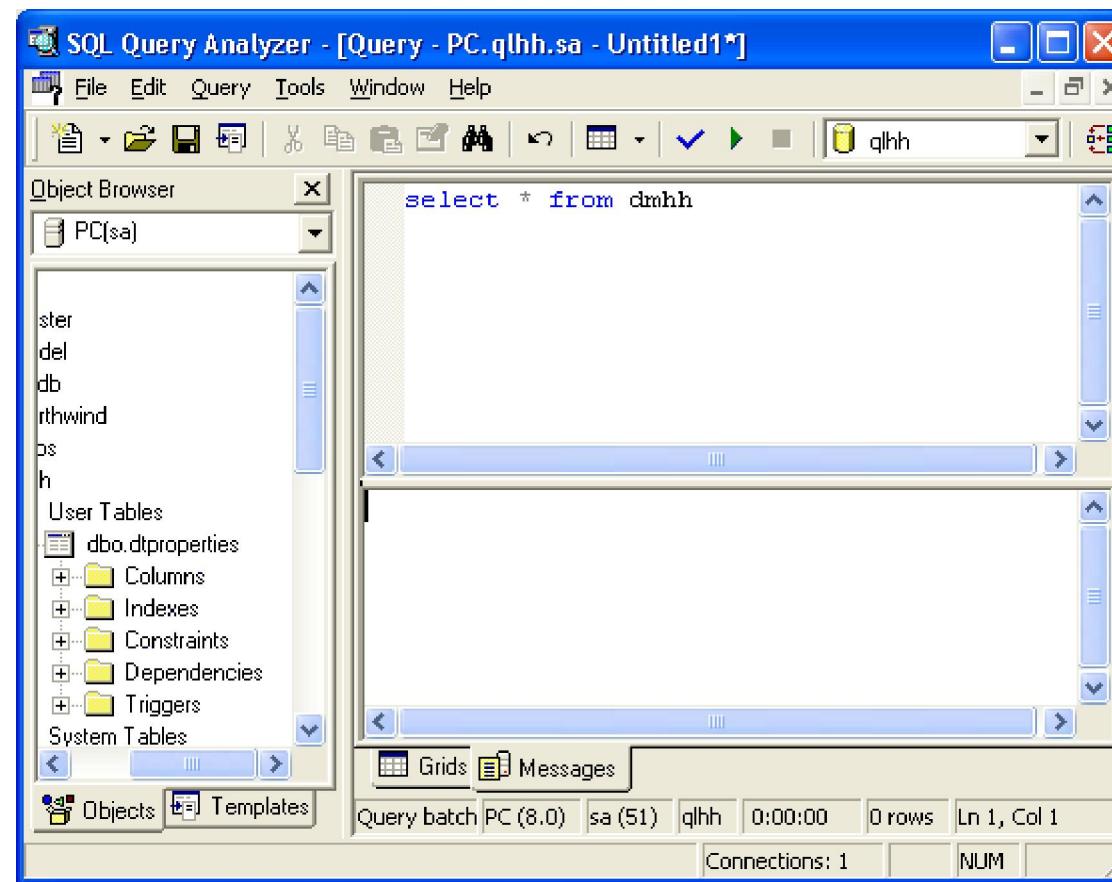
- Chọn TCP/IP
- Server Alias :nhập vào nhän
- Server name : nhập vào địa chỉ IP của Server



Chú ý: Tiện ích này chỉ sử dụng trong hệ thống mạng

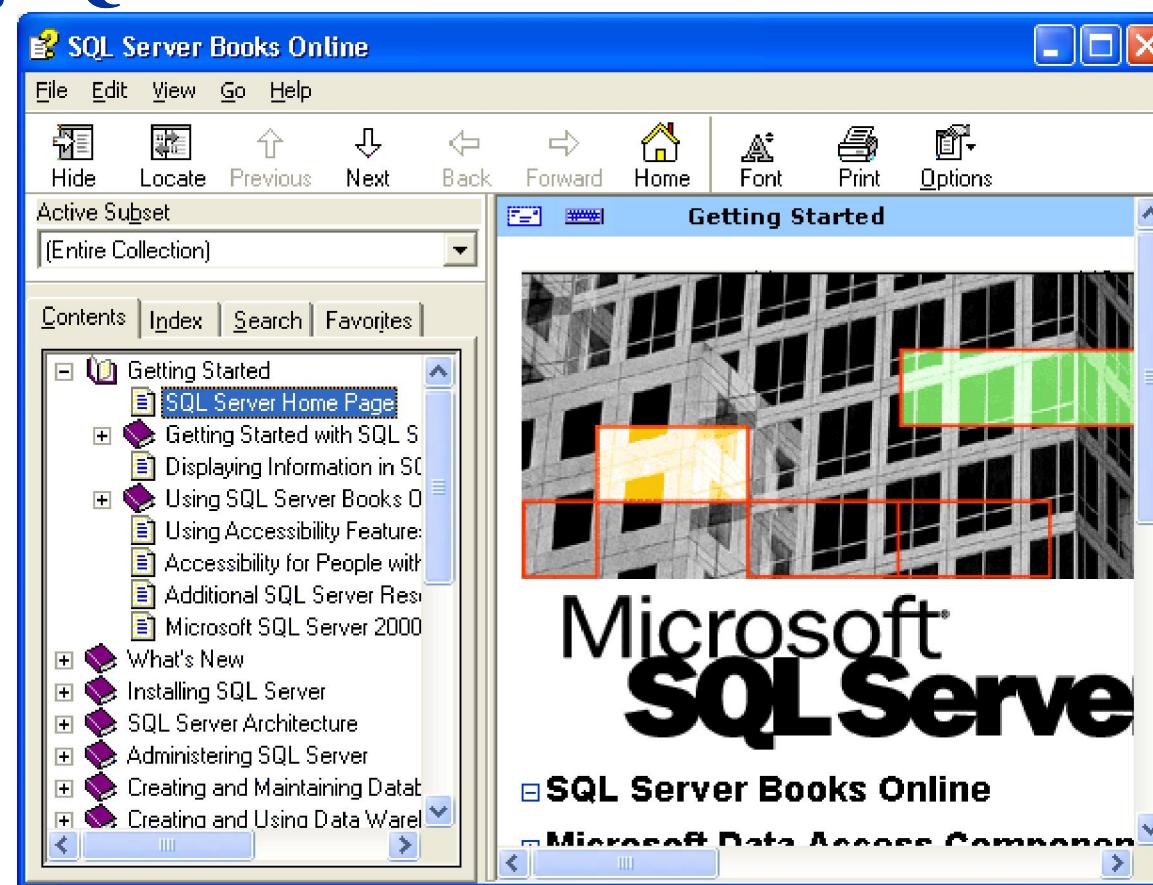
### III.Các tiện ích của SQL Server

#### Tiện ích Query Enalyzer :là màn hình viết lệnh của SQL Server



### III.Các tiện ích của SQL Server

#### Tiện ích Book Online :Tài liệu hướng dẫn cách sử dụng SQL Server

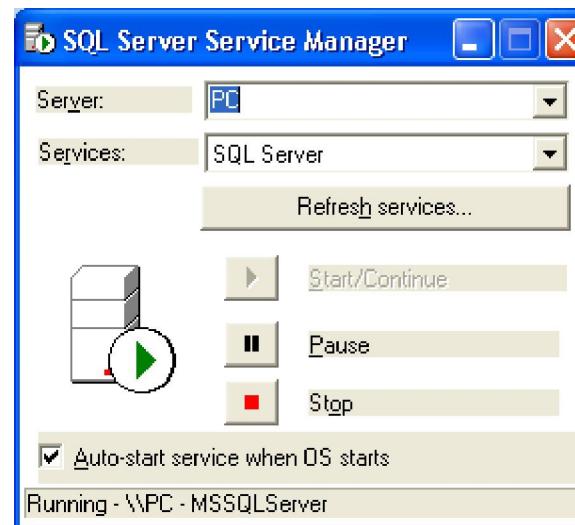


### III.Các tiện ích của SQL Server

#### Tiện ích Service Manager :quản lý các dịch vụ liên quan đến SQL Server

Muốn sử dụng SQL Server phải khởi động dịch vụ này

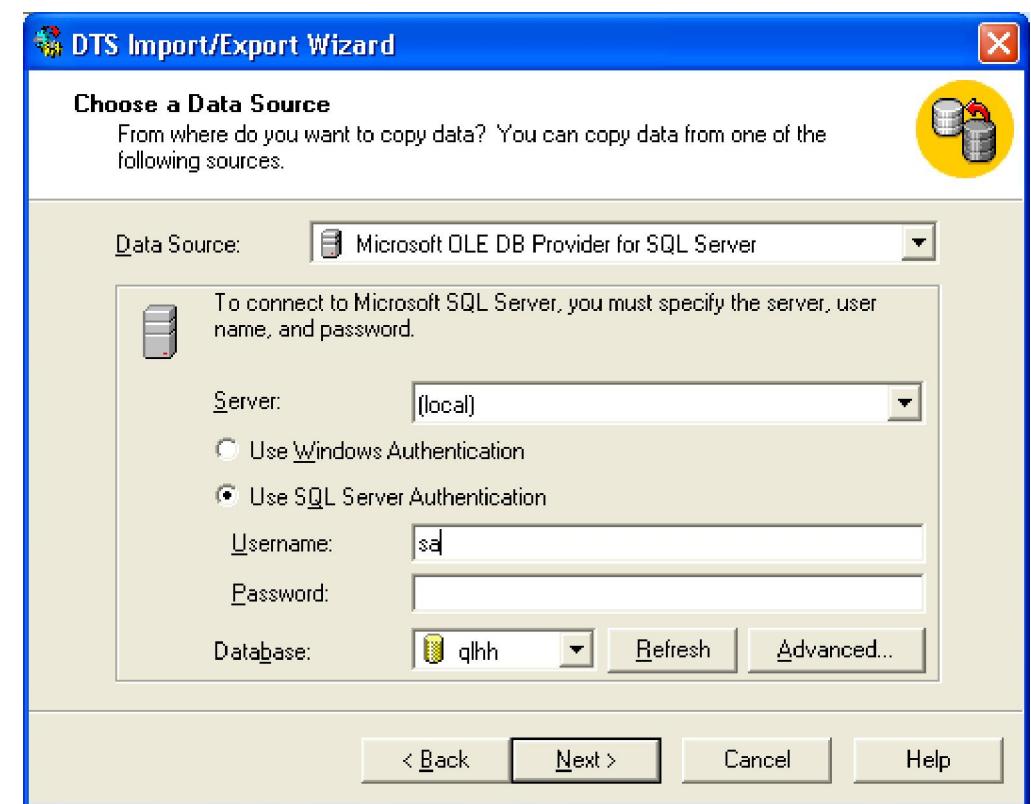
Các dịch vụ này khởi động bằng cách : Start -> Programs->Microsoft SQL Server->Dịch vụ



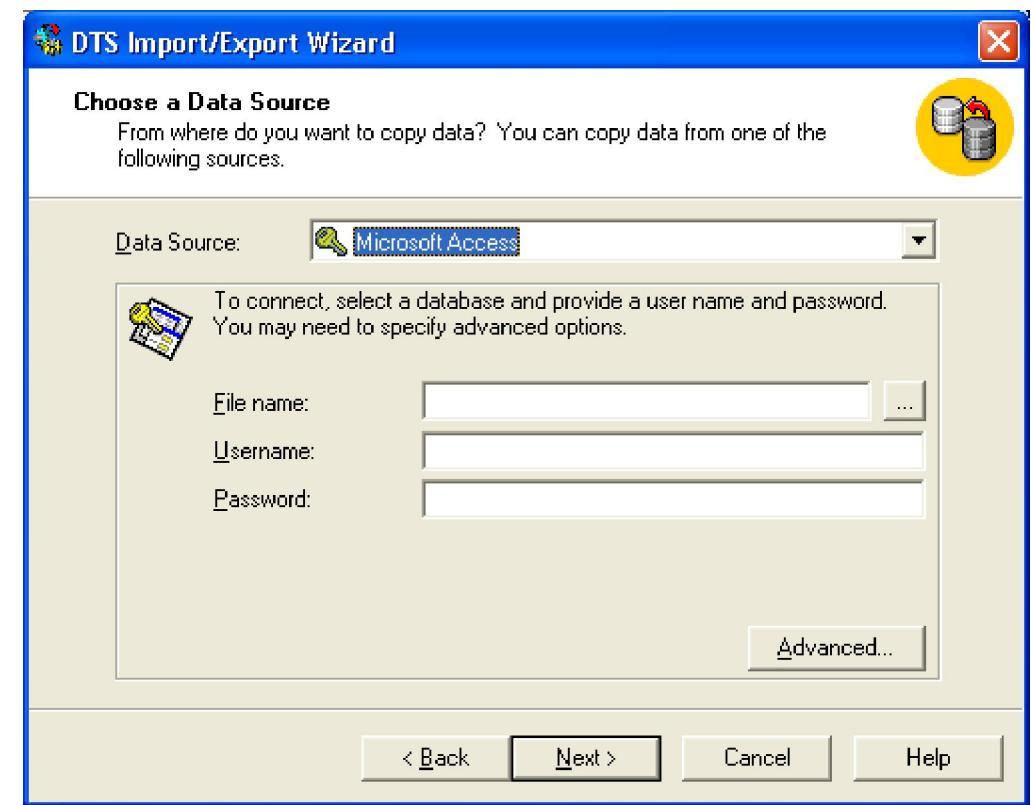
### III.Các tiện ích của SQL Server



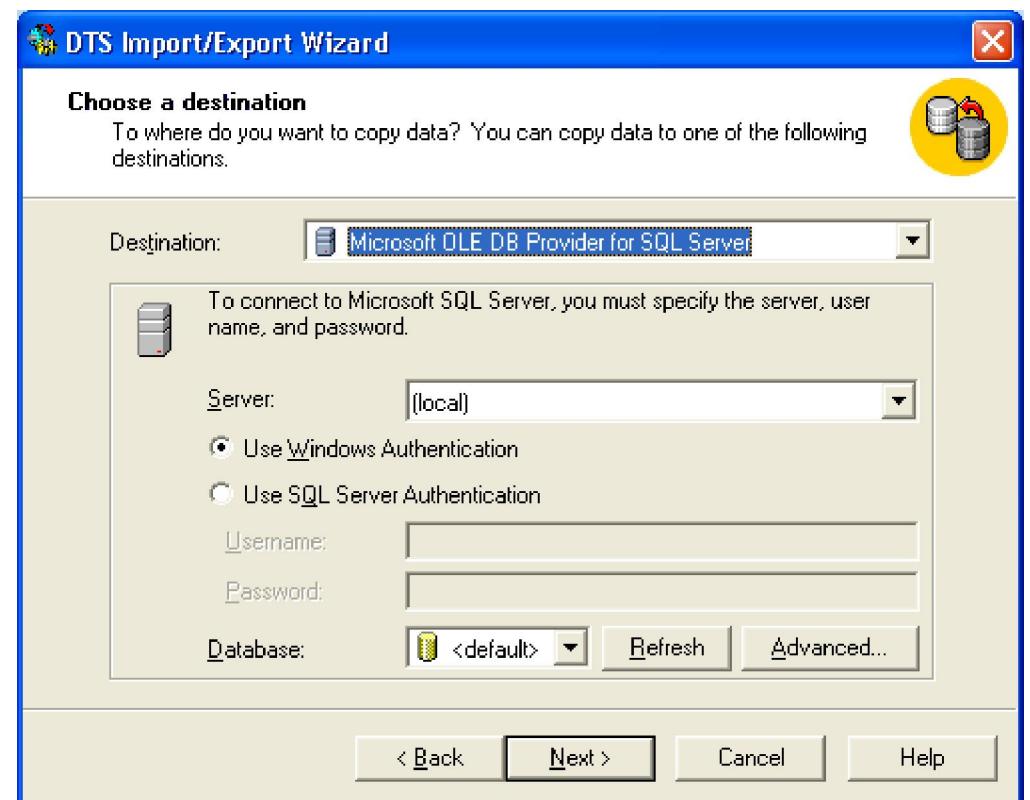
### III.Các tiện ích của SQL Server



### III.Các tiện ích của SQL Server



### III.Các tiện ích của SQL Server



## Chương II: Xây dựng một CSDL trong SQL Server

- Một (CSDL) của SQL Server là tập hợp các đối tượng
  - ✓ bảng (Table):Chứa dữ liệu
  - ✓ bảng ảo (View):chứa dữ liệu lấy từ Bảng
  - ✓ thủ tục nội tại (Stored Procedure): Chứa các lệnh thao tác với Dữ liệu

## Chương II: Xây dựng một CSDL trong SQL Server

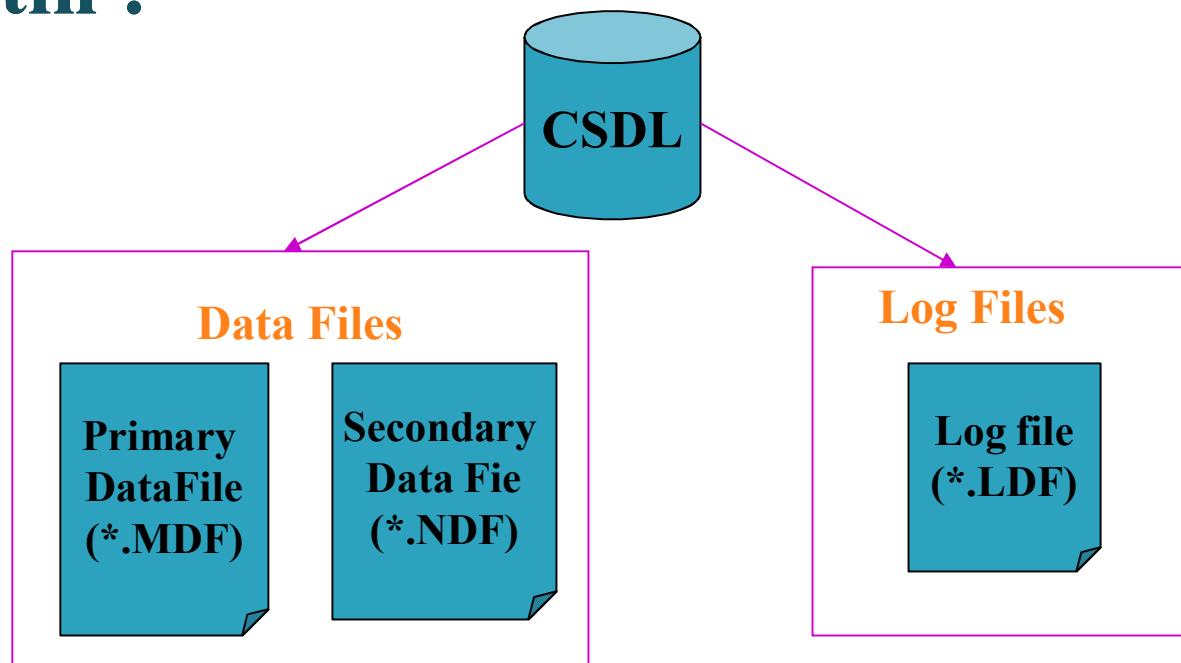
- ✓ Triger :Chứa các lệnh thao tác với Dữ liệu,kiểm tra ràng buộc tự động
- ✓ Constraint :Chứa các ràng buộc dữ liệu
- Một CSDL do một hoặc nhiều người tạo nhưng có thể được nhiều người truy cập
- Trong SQL Server có thể tạo được 32767 CSDL khác nhau

## Chương II: Xây dựng một CSDL trong SQL Server

- **Cấu trúc lưu trữ của một CSDL :**  
**Một CSDL trong SQL Server tối thiểu sẽ dùng hai tập tin vật lý để lưu trữ dữ liệu:**
  - ✓ **Một dùng lưu trữ dữ liệu (Data file)**
  - ✓ **Một dùng để lưu trữ các vết giao tác (Transaction log file)**

## Chương II: Xây dựng một CSDL trong SQL Server

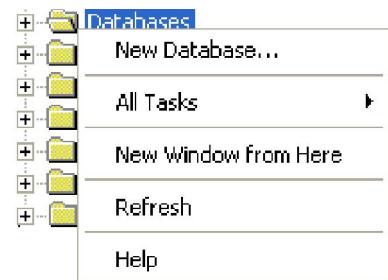
✓ Thường được tổ chức lưu trữ trong 3 tập tin :



## Chương II: Xây dựng một CSDL trong SQL Server

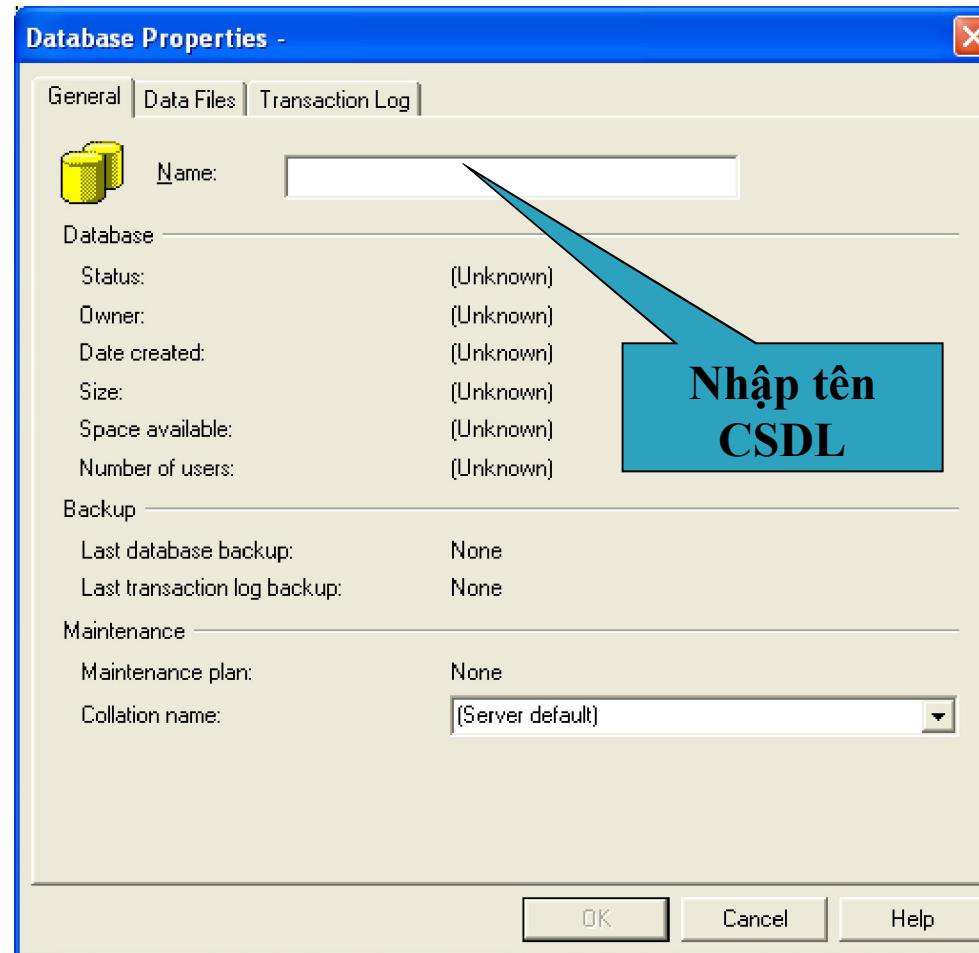
- ✓ Tạo mới một CSDL trong SQL Server
- ✓ Cách 1: Sử dụng tiện ích

EnterPriseManager : Click chuột phải vào đối tượng Databases chọn New Database



## Chương II: Xây dựng một CSDL trong SQL Server

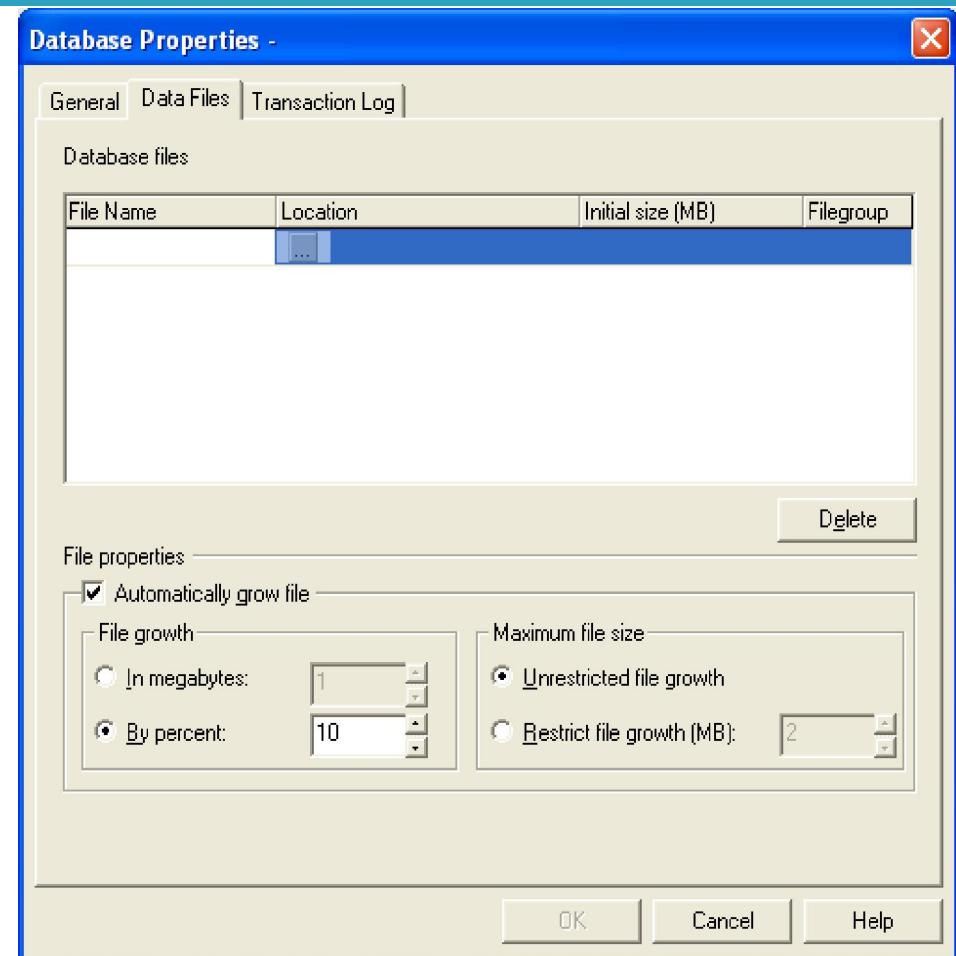
Xuất hiện hộp thoại :



## Chương II: Xây dựng một CSDL trong SQL Server

### Chọn Tab Data Files:

- ❑ **Filename :**Tên tập tin Data Files
- ❑ **Location:** Vị trí lưu trữ
- ❑ **Initial size(Mb) :**Kích thước ban đầu
- ❑ **Automatically grow file:** Tự động tăng kích thước
  - In megabytes : tăng bao nhiêu mB
  - By percent :Tăng theo phần trăm
- ❑ **Maximun file size :** kích thước tối đa của CSDL
  - Unrestricted file growth :Không giới hạn kích thước
  - Restrict file growth(MB) :giới hạn
- ❑ **Tương tự cho Transaction Log**



## Chương II: Xây dựng một CSDL trong SQL Server

Tạo CSDL bằng cách viết lệnh trong Query Enalyzer

# Chương II: Xây dựng một CSDL trong SQL Server

## Các kiểu dữ liệu trong SQL Server


## Chương II: Xây dựng một CSDL trong SQL Server

## Các kiểu dữ liệu trong SQL Server

# Chương II: Xây dựng một CSDL trong SQL Server

## Các kiểu dữ liệu trong SQL Server

### > Các kiểu dữ liệu dạng tiền tệ

<b>Money</b>	<b>8 bytes</b>	từ <b>-922,337,203,685,477.5808</b> đến <b>+922,337,203,685,477.5807</b>
--------------	----------------	--

<b>SmallMoney</b>	<b>4 bytes</b>	từ <b>-214,748.3648</b> đến <b>+ 214,748.3647</b>
-------------------	----------------	---

### > Các kiểu dữ liệu dạng ngày và giờ

<b>DateTime</b>	<b>8 bytes</b>	từ <b>01/01/1753</b> đến <b>31/12/9999</b>
-----------------	----------------	--

<b>SmallDateTime</b>	<b>4 bytes</b>	từ <b>01/01/1900</b> đến <b>06/06/2079</b>
----------------------	----------------	--

### > Các kiểu dữ liệu dạng chuỗi nhị phân (Binary String)

<b>Binary</b>	<b>N bytes</b>	<b>từ 1 đến 8000 bytes</b>
---------------	----------------	----------------------------

<b>VarBinary</b>	<b>N bytes</b>	<b>từ 1 đến 8000 bytes</b>
------------------	----------------	----------------------------

<b>Image</b>	<b>N bytes</b>	<b>từ 1 đến 2,147,483,647 bytes</b>
--------------	----------------	-------------------------------------

# Chương II: Xây dựng một CSDL trong SQL Server

## Các thuộc tính cơ bản của Một Bảng

- **Tên bảng** (Table Name) Độ dài $\leq$  128 ký tự.  
Tên bảng phải duy nhất bên **trong phạm vi** của người đã tạo ra nó trong một CSDL
- **Tên cột** (Column Name): là tên của các cột bên trong bảng, tên của các cột bên trong bảng phải duy nhất.
- **Kiểu dữ liệu** (Data Type): qui định kiểu dữ liệu mà cột sẽ lưu trữ bên trong bảng.

## Chương II: Xây dựng một CSDL trong SQL Server

- **Độ dài dữ liệu** (Data Length): dùng để qui định độ dài dữ liệu mà cột sẽ lưu trữ đối với các kiểu dữ liệu dạng chuỗi, số.
- **Số ký số lưu trữ** (Precision): là số ký số tối đa mà các kiểu dữ liệu dạng số có thể lưu trữ được.
- **Số lẻ lưu trữ** (Scale): là số lẻ tối đa mà các kiểu dữ liệu dạng số thập phân dùng để lưu trữ.

## Chương II: Xây dựng một CSDL trong SQL Server

- **Cột định danh (Identity):** Giá trị của cột định danh là số nguyên không trùng lặp do hệ thống Microsoft SQL Server tự động cấp phát.
  - Các kiểu dữ liệu của cột định danh : Int, SmallInt, TinyInt, Decimal , Numeric
  - Thuộc tính **Identity Seed :Số đầu tiên**
  - Thuộc tính **Identity Increment :giá trị tăng**

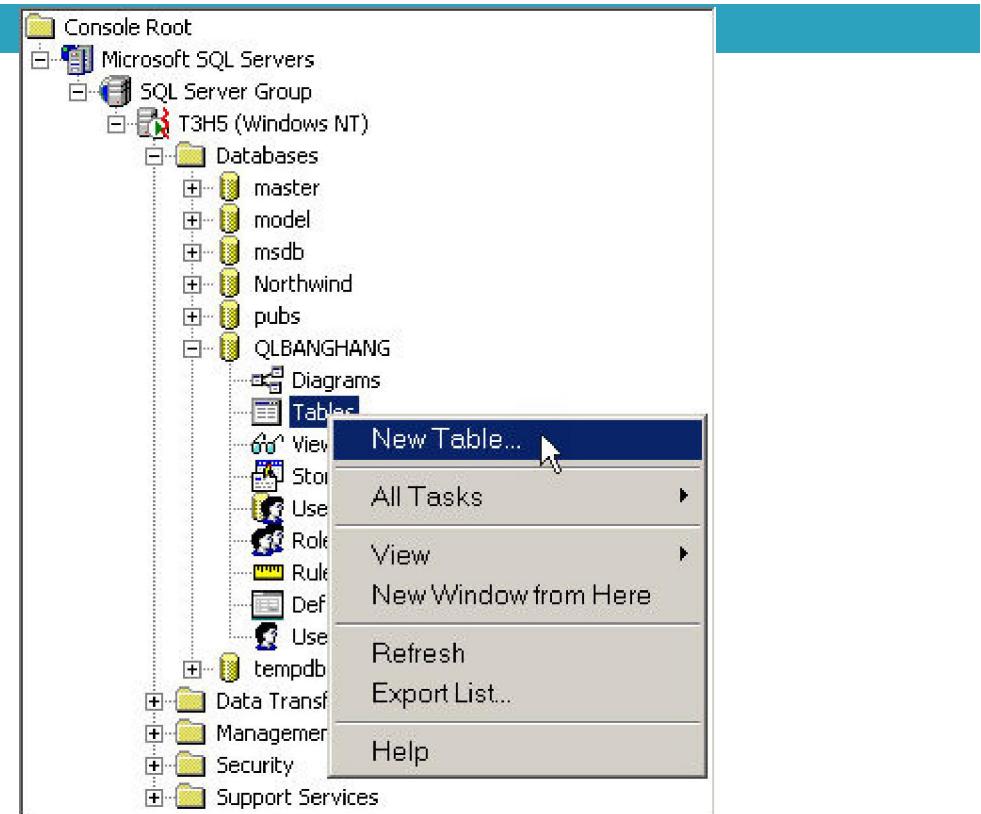
## Chương II: Xây dựng một CSDL trong SQL Server

- **Cho phép để trống dữ liệu** (Allow Null): qui định dữ liệu có thể được phép để trống hay là phải nhập
- **Giá trị mặc định** (Default Value): là giá trị mặc nhiên sẽ được gán vào cột dữ liệu khi người sử dụng thêm mới một mẫu tin nhưng lại để trống giá trị tại cột dữ liệu đó.

# Chương II: Xây dựng một CSDL trong SQL Server

## Tạo Bảng trong Sql Server

- Cách 1: Sử dụng tiện ích Enterprise Manager
  - Mở CSDL Click chuột phải trên đối tượng **Tables**, chọn chức năng **New Table...** trong thực đơn tắt.

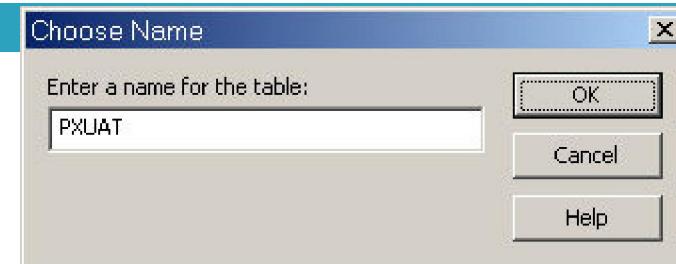


# Chương II: Xây dựng một CSDL trong SQL Server

Màn hình sau xuất hiện :

Nhập tên bảng,nhấp OK

- Màn hình thiết kế bảng xuất hiện :nhập thông tin và chọn cột làm khoá chính



	Column Name	Datatype	Length	Precision	Scale	Allow Nulls	Default Value	Identity	Identity Seed	Identity Increment
▶	SOPX	int	4	10	0	<input type="checkbox"/>		<input checked="" type="checkbox"/>	1	1
	NGAYXUAT	datetime	8	0	0	<input type="checkbox"/>	(GETDATE())	<input type="checkbox"/>		
	HOTENKH	varchar	100	0	0	<input checked="" type="checkbox"/>		<input type="checkbox"/>		
						<input type="checkbox"/>		<input type="checkbox"/>		
						<input type="checkbox"/>		<input type="checkbox"/>		
						<input type="checkbox"/>		<input type="checkbox"/>		

## Chương II: Xây dựng một CSDL trong SQL Server

Tạo bảng bằng lệnh trong màn hình Query Elyzer

## Chương II: Xây dựng một CSDL trong SQL Server

### Tạo bảng bằng lệnh trong màn hình Query Elyzer

**Tạo cấu trúc bảng có giá trị mặc định**

**Cú pháp:**

**CREATE TABLE Tên\_bảng**

**(**

**Tên\_cột\_1 Kiểu\_dữ\_liệu DEFAULT Giá\_trị|Hàm ,**

**Tên\_cột\_2 Kiểu\_dữ\_liệu [NOT NULL] [ ,...]**

**)**

**Trong đó:**

+ **Giá trị:** là giá trị cụ thể sẽ được gán vào cột.

+ **Hàm:** là tên hàm cùng với các tham số (nếu có), tuy nhiên kiểu dữ liệu mà hàm trả về phải là cùng với kiểu dữ liệu mà cột sẽ lưu trữ.

## Chương II: Xây dựng một CSDL trong SQL Server

### Tạo bảng bằng lệnh trong màn hình Query Elyzer

#### Ví dụ 1

```
CREATE TABLE VATTU
```

```
(  
    MAVTU CHAR(4) NOT NULL,  
    TENVTU VARCHAR(100) NOT NULL,  
    DVTINH VARCHAR(10) NOT NULL,  
    PHANTRAM REAL DEFAULT 20  
)
```

#### Ví dụ 2

```
CREATE TABLE DONDH
```

```
(  
    SODH CHAR(4) NOT NULL,  
    NGAYDH DATETIME DEFAULT GETDATE(),  
    MANHACC CHAR(3) NOT NULL  
)
```

## Chương II: Xây dựng một CSDL trong SQL Server

### Tạo cấu trúc bảng có cột định danh

Cú pháp:

**CREATE TABLE Tên\_bảng**

(

**Tên\_cột\_1 Kiểu\_dữ\_liệu\_số IDENTITY [(Số\_bắt\_đầu, Chỉ\_số\_tăng)] ,  
Tên\_cột\_2 Kiểu\_dữ\_liệu [NOT NULL] [ ,...]**

)

- Trong một bảng chỉ có tối đa một cột được chỉ định làm cột định danh.
- Số bắt đầu: là số mà SQL Server sử dụng để cấp phát cho mẫu tin đầu tiên. Mặc định là 1.
- Chỉ số tăng: là chỉ số mà SQL Server cộng lên để cấp phát cho từng mẫu tin kết tiếp. Mặc định là 1.

## Chương II: Xây dựng một CSDL trong SQL Server

Tính toàn vẹn dữ liệu trong cơ sở dữ liệu : các ràng buộc trong SQL Server được quản lý bởi hai đối tượng :CONSTRAINT và TRIGGER

- Constraint : quản lý các ràng buộc như :
  - ▣ Ràng buộc dữ liệu duy nhất
  - ▣ Ràng buộc dữ liệu tồn tại
  - ▣ Ràng buộc miền giá trị
- Trigger :Quản lý các ràng buộc phức tạp, liên quan đến dữ liệu trên nhiều bảng

# Chương II: Xây dựng một CSDL trong SQL Server

## Xây dựng các Constraint

- A>Kiểm tra duy nhất dữ liệu :có hai loại
  - Primary Key (khoá chính ) :dữ liệu khác rỗng và duy nhất
  - Unique : dữ liệu có thể rỗng và dữ liệu tồn tại duy nhất

## Chương II: Xây dựng một CSDL trong SQL Server

Định nghĩa ràng buộc khoá chính :

```
CREATE TABLE Tên_bảng
(
    Tên_cột_1 Kiểu_dữ_liệu DEFAULT Giá_trị|Hàm ,
    Tên_cột_2 Kiểu_dữ_liệu [NOT NULL] [,...]
    [CONSTRAINT PK_Tên_bảng_TenCot ] PRIMARY KEY
        Danh_sách_cột_khóa_chính)
)
```

## Chương II: Xây dựng một CSDL trong SQL Server

Định nghĩa ràng buộc dữ liệu duy nhất :

```
CREATE TABLE Tên_bảng  
(  
    Tên_cột_1 Kiểu_dữ_liệu DEFAULT Giá_trị|Hàm ,  
    Tên_cột_2 Kiểu_dữ_liệu [NOT NULL] [ ,...]  
    [CONSTRAINT UQ_Tên_bảng_Tên_cột]  
    UNIQUE (Danh_sách_các_cột)  
)
```

## Chương II: Xây dựng một CSDL trong SQL Server

Ví dụ :

```
CREATE TABLE VATTU
(
MAVTU varchar(4) not null,
TenVtu varchar(100),
Constraint PK_Vattu_Mavtu_tenvtu primary key(Mavtu,tenvtu),
constraint UQ_Vattu_tenvtu unique(tenvtu)
)
```

## Chương II: Xây dựng một CSDL trong SQL Server

B>Kiểm tra tồn tại Dữ liệu :Định nghĩa khoá ngoại

**CREATE TABLE Tên\_bảng**

(

Tên\_cột\_1 Kiểu\_dữ\_liệu DEFAULT Giá\_trị|Hàm ,

Tên\_cột\_2 Kiểu\_dữ\_liệu [NOT NULL] [ ,...]

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

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

**REFERENCES                      Tên\_bảng\_tham\_chiếu**

**(Danh\_sách\_cột\_tham\_chiếu)**

)

## Chương II: Xây dựng một CSDL trong SQL Server

B>Ví dụ :

```
CREATE TABLE CTHOADON
(
    SOHD Varchar(3) not null ,
    MAVTU Varchar(3) not null,
    Soluong int,
    Dongia float,
    CONSTRAINT FK_CTHOADON_SOHD
    FOREIGN KEY (SOHD)
    REFERENCES HOADON(SOHD),
    CONSTRAINT FK_CTHOADON_MAVTU
    FOREIGN KEY (MAVTU)
    REFERENCES VATTU(MAVTU)
)
```

## Chương II: Xây dựng một CSDL trong SQL Server

C>Kiểm tra ràng buộc miền giá trị :

Cú pháp :

```
CREATE TABLE Tên_bảng  
(  
    Tên_cột_1 Kiểu_dữ_liệu DEFAULT Giá_trị|Hàm ,  
    Tên_cột_2 Kiểu_dữ_liệu [NOT NULL] [,...]  
    [CONSTRAINT CK_Tên_bảng_Tên_cột]  
    CHECK (Biểu_thức_luận_lý)  
)
```

## Chương II: Xây dựng một CSDL trong SQL Server

B>Ví dụ :

```
CREATE TABLE CTHOADON
(
SOHD Varchar(3) not null ,
MAVTU Varchar(3) not null,
Soluong int,
Dongia float,
CONSTRAINT CK_CTHOADON_SOLUONG
CHECK (Soluong>0),
CONSTRAINT CK_CTHOADON_DonGia
CHECK (Dongia>0)
```

)

## Chương II: Xây dựng một CSDL trong SQL Server

B>Thêm Constraint vào bảng đã có

Cú pháp chung:

ALTER TABLE Tên\_bảng

ADD [ CONSTRAINT Tên\_Constraint ]

LOẠI Các\_tham\_số [ , ...]

## Chương II: Xây dựng một CSDL trong SQL Server

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)

## Chương II: Xây dựng một CSDL trong SQL Server

### UNIQUE

```
ALTER TABLE Tên_bảng ADD [ CONSTRAINT  
    UQ_Tên_bảng_Tên_cột ]  
UNIQUE (Danh_sách_các_cột)
```

## Chương II: Xây dựng một CSDL trong SQL Server

### 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)

## Chương II: Xây dựng một CSDL trong SQL Server

### Kiểm tra miền giá trị :

ALTER TABLE Tên\_bảng

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

CHECK (Biểu\_thức\_luận\_lý)

## Chương II: Xây dựng một CSDL trong SQL Server

Thêm giá trị mặc định :

```
ALTER TABLE Tên_bảng  
ADD [ CONSTRAINT DF_Tên_bảng_Tên_cột ]  
DEFAULT Giá_trị_mặc_định FOR Tên_cột
```

## Chương II: Xây dựng một CSDL trong SQL Server

Hủy một Constraint :

ALTER TABLE Tên\_bảng

DROP CONSTRAINT Tên\_constraint [ , ...]

Tắt các Constraint :

ALTER TABLE Tên\_bảng

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

Bật các Constraint :

ALTER TABLE Tên\_bảng

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

## Chương II: Xây dựng một CSDL trong SQL Server

### Xóa cấu trúc bảng

DROP TABLE Danh\_sách\_tên\_các\_bảng

### Thay đổi cấu trúc bảng

#### a. Thêm một cột mới trong bảng

**ALTER TABLE Tên\_bảng**

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

#### b. Hủy bỏ cột hiện có bên trong bảng

**ALTER TABLE Tên\_bảng**

**DROP COLUMN Tên\_cột [ , ... ]**

## Chương II: Xây dựng một CSDL trong SQL Server

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”**

Đổi tên bảng

**EXEC sp\_rename “Tên\_bảng”, “Tên\_mới”**

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

# BÀI 3: BÀNG ẢO(view)

## □ 1>Định Nghĩa :

□ Bảng ào là một đối tượng chứa câu lệnh Select lấy dữ liệu  
Từ các bảng trong csdl  
Khi truy cập vào CSLD ,người sử dụng không phân biệt được  
Bảng ảo và Bảng Thật

## 2> Tạo View

Create view Ten\_view

As

Câu lệnh Select

# Cấu trúc câu lệnh Select

## □ 2>Cấu trúc câu lệnh Select

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]

# Cấu trúc câu lệnh Select

- Mệnh đề Select :chứa các cột,các hàm muốn hiển thị
- Nếu cột tồn tại trong hai bảng trong mệnh đề From ta phải ghi rõ :Tên\_Bảng.Tên\_Cột  
Trong mệnh đề Select ta có thể dùng cấu trúc lệnh

**Case**

**When điều kiện 1 Then Giá trị 1**

**When Điều kiện 2 then Giá trị 2**

...

**Else**

**Giá trị n**

**End**

Để hiển thị dữ liệu

- Mệnh đề from : cho biết dữ liệu trên câu truy vấn được lấy từ các bảng nào
- nếu dữ liệu lấy từ nhiều bảng thì các bảng này phải có quan hệ với nhau:
  - Cú pháp :

# Cấu trúc câu lệnh Select

## Inner

**Left**

**Join** **Bảng 2 on** **Bảng 1.tên cột= bảng 2.tên cột**

**Right**

**full**

- Mệnh đề Where : mệnh đề lọc dữ liệu trả về hoặc để gom nhóm
- Trong mệnh đề Where có thể sử dụng các toán tử :
  - Like tìm chuỗi gần đúng
    - % Thay thế 1 chuỗi
    - \_ (Gạch dưới) thay thế 1 ký tự
  - Toán tử BETWEEN Gt1 AND GT2

# Cấu trúc câu lệnh Select

- ❑ Mệnh đề Group by : mệnh đề gom nhóm dữ liệu
  - ❖ Mệnh đề này phải có khi trong câu truy vấn có sử dụng hàm thống kê và trên mệnh đề Select có lấy các giá trị không sử dụng hàm thống kê
  - ❖ Cú pháp : Group By Danh sách cột,Biểu thức
  - ❖ Với Danh sách cột,Biểu thức :được lấy trên mệnh đề Select không sử dụng hàm thống kê
- ❑ Các Hàm thống kê:
  - ❖ Sum(Ten\_Cot) :tính tổng với cột có kiểu số
  - ❖ Count(Ten\_Cot/\*) :tính tổng số dòng
  - ❖ AVG(Ten\_Cot) :tính giá trị trung bình
  - ❖ Max(Ten\_cot) : giá trị lớn nhất
  - ❖ Min(Ten\_Cot) :giá trị nhỏ nhất)

# Cấu trúc câu lệnh Select

- ❑ Mệnh đề HaVing : mệnh đề Lọc nhóm dữ liệu
  - ❖ Mệnh đề này chỉ tồn tại khi trong câu truy vấn có mệnh đề Group By và điều kiện lọc dữ liệu có hàm thống kê
  - ❖ Chú ý : trong câu truy vấn có hai mệnh đề lọc dữ liệu là Where và havinh
    - ❖ Where :lọc dữ liệu để tạo nhóm ( có điều kiện không sử dụng hàm thống kê)
    - ❖ Having :lọc nhóm( các điều kiện có thống kê)
- ❑ Mệnh đề OrDer By : Mệnh đề sắp xếp dữ liệu
  - ❖ Cú pháp Order by [Ten\_bang.]Ten\_Cot [asc/Desc] [,...]
- ❑ Mệnh đề Compute /Compute .. By : là mệnh đề tạo giá trị thống kê cuối kết quả hoặc thống kê theo cột
  - ❖ Cú pháp
    - ❖ Compute Hàm thống kê : tạo thống kê cuối kết quả
    - ❖ Compute Hàm thống kê By [Ten\_Bang.]Ten\_cot :thống kê theo cột,dữ liệu phải được sắp xếp theo cột này
  - ❖ **Chú ý:** Trong View không có mệnh đề này

# Cấu trúc câu lệnh Select

- ❑ Các toán tử dùng trên mệnh đề Select
  - ❖ Cú pháp : Select Toán tử ...
  - ❖ Với các toán tử :
    - ❖ Distict : loại bỏ các dòng dữ liệu trùng
    - ❖ Top n : lấy n dòng đầu tiên
    - ❖ Top n with ties : lấy những dòng thuộc n cấp đầu tiên
    - ❖ Top n PERCENT : lấy N % dòng đầu tiên

# Cấu trúc câu lệnh InSert into

- ❑ **Insert into** : dùng thêm dữ liệu vào bảng,có 2 dạng
  - ❑ **Thêm một dòng**
    - ❖ Cú pháp : Inser into Ten\_bang(Ten\_cot1,Ten\_cot2[,...])  
Values(Gia\_Tri1,...)  
với giá trị :
      - ✓ kiểu chuỗi : nằm trong hai dấu '' ví dụ ‘Trần thanh nhât’
      - ✓ Kiểu Ngày : ‘yyyy-mm-dd’      ví dụ :’1982-02-29’
  - ❑ **Thêm nhiều dòng**
    - ❖ Cú pháp : Inser into Ten\_bang\_Dich(Ten\_cot1,Ten\_cot2[,...])  
Select Ten\_bang\_nguon.ten\_cot ,...  
from ...

# Cấu trúc câu lệnh UpDate

□ 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 :** *dữ liệu cập nhật có thể là một giá trị thống kê trả về từ câu truy vấn con*

*Có thể dùng cấu trúc Case trong lệnh cập nhật*

*Trong câu truy vấn update không sử dụng hàm thống kê*

# Cấu trúc câu lệnh Delete

❑ 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*

# Cấu trúc câu lệnh Select .. into

- ❑ Select .. into : dùng tạo một bảng có cấu trúc và dữ liệu lấy từ các bảng có trong CSDL

❖ Cú pháp :

    Select [Ten\_Bang.]tenCot [,...]

    From ..

# Câu truy vấn Con (Sub Query)

- ❑ Sub Query :là câu truy vấn mà kết quả trả về là điều kiện để so sánh,gán giá trị cho câu truy vấn khác
- ❑ Đặt điểm :
  - ❖ Chỉ trả về 1 cột dữ liệu
  - ❖ Nếu trả về 1 dòng : điều kiện so sánh giữa câu truy vấn chính và câu truy vấn con là các toán tử  $:=$ ,  $>$ ,  $<$ ,  $\neq$ ,  $>=$ ,  $<=$
  - ❖ Nếu trả về nhiều dòng : điều kiện so sách : in (nằm trong) hoặc Not in(không nằm trong)

# Bài 4: Lập trình với SQL Server

- I> Biến và các cấu trúc lệnh trong SQL Server
- A> Biến :
  - 1> 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
    - Gán giá trị cho biến:  
Gán bằng 1 giá trị :  
**Set @tên\_biến=giátrị/hàm**  
hoặc **Select @tên\_biến=Giátrị/hàm**
    - Gán bằng kết quả trả về của câu truy vấn  
**Select @tên\_Biến=TênCOLUMN/Hàm ..from ..**

# Bài 4: Lập trình với SQL Server

- 1> Biến hệ thống (t155) : là biến do SQL Server cung cấp, Cho biết trạng thái của hệ thống, Biến hệ thống luôn bắt đầu bằng hai ký tự @@, Chỉ đọc
- Một số các biến thường dùng

Tên Biến	Ý nghĩa
<b>@@Error</b>	<b>Cho biết các lệnh trước đó có lỗi hay không, nếu có lỗi <math>@@Error &lt;&gt; 0</math></b>
<b>@@Rowcount</b>	<b>Cho biết số dòng bị tác động bởi câu truy vấn gần nhất</b>
<b>@@Fetch_Status</b>	<b>Trả về <math>&lt;&gt; 0</math> nếu lệnh duyệt Cursor gây lỗi</b>

# Bài 4: Lập trình với SQL Server

## □ B> Các cấu trúc lệnh

### □ 1>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

# Bài 4: Lập trình với SQL Server

- B>Các cấu trúc lệnh
  - 2>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

# Bài 4: Lập trình với SQL Server

- 2> Các hàm cơ bản trong SQL Server
- A> Các hàm chuyển đổi kiểu dữ liệu : thường dùng để chuyển dữ liệu từ số, ngày sang chuỗi
  - Hàm Cast : chuyển đổi 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 đổi từ ngày sang chuỗi
    - ▣ Cú pháp : **Convert(Kiểu\_dữ\_Liệu,Biểu\_Thức[,Định\_Dạng])**  
Ví dụ : Convert(char(10),Getdate(),105)

# Bài 4: Lập trình với SQL Server

Một số định dạng thường dùng

Định dạng năm(YY)	Định dạng năm(YYYY)	Hiển thị dữ liệu
1	101	Mm/dd/yyyy
3	103	Dd/mm/yyyy
5	105	Dd-mm-yyyy
12	112	Yyyymmdd

•Hàm Str :chuyển số thành chuỗi

Cú pháp :STR(số thực,Số ký tự[,số lẻ])

Ví dụ Str(12.345,5,2)->12.35

Chú ý: khi nối chuỗi với số, ngày ta phải chuyển các giá trị này sang chuỗi,sau đó sử dụng dấu + để nối

# Bài 4: Lập trình với SQL Server

## B> Các hàm ngày giờ

### Bảng mô tả các định dạng trong các hàm thời gian

Giá trị	Định dạng
Năm	yy, yyyy
Quý	qq, q
Tháng	mm, m
Ngày trong năm	dy, y
Ngày trong tuần	dw
Ngày trong tháng	dd, d
Tuần	wk, ww
Giờ	hh
phút	mi, n
giây	ss, s

# Bài 4: Lập trình với SQL Server

**Hàm DateAdd** : dùng cộng một số vào giá trị ngày và trả về một giá trị ngày

Cú pháp : *DateAdd(Định dạng,Số,Ngày)->Ngày mới*

Ví dụ

```
declare @ngaymoi datetime  
set @ngaymoi=dateadd(mm,5,getdate())  
print convert(char(10),@ngaymoi,105)
```

=>11-01-2007

**Hàm DateDiff**: trả về khoảng cách của hai ngày

Cú pháp : *Datediff(định dạng,ngày\_1,ngày\_2)->số nguyên*

Ví dụ :*print Datediff(mm,getdate(),@ngaymoi) ->5*

# Bài 4: Lập trình với SQL Server

## Hàm DateName: trả về chuỗi thời gian

Cú pháp : *Datename(Định dạng, Ngày)->chuỗi*

Ví dụ

*print datename(dw,getdate())->Friday*

## Hàm DatePart: trả về một giá trị trong của ngày

Cú pháp : *Datepart(định dạng, ngày)->số nguyên*

Ví dụ : *print datepart(qq,getdate())->3*

Hàm Getdate() ->*Trả về ngày hiện hành*

Hàm Day(), Month(), Year() -> *trả về ngày, tháng, năm*

# Bài 4: Lập trình với SQL Server

## Hàm Round :hàm làm tròn số

Cú pháp :Round(số,số chữ số thập phân)

ví dụ : round(12.3456,2)->12.35

Left(chuỗi nguồn,số ký tự) ->cắt chuỗi từ trái sang lấy số ký tự

Right(chuỗi nguồn,số ký tự)-> cắt chuỗi từ phải sang lấy số ký tự

Substring(chuỗi,vị trí,số ký tự) -> cắt chuỗi từ vị trí sang lấy số ký tự

Hàm Exists(Câu truy vấn Select)=True nếu câu truy vấn có giá trị trả về

# Bài 4: Lập trình với SQL Server

3>CurSor :là đối tượng dùng để chứa dữ liệu lấy từ CSDL ,giống  
đối tượng recordset trong VB

❖Cú pháp khai báo biến Cursor :

*Declare Tên\_Biến CURSOR*

*[phạm vi] [di chuyển][trạng thái][xử lý]*

*For câu lệnh Select*

*[For update [OF danh sách cột]]*

Trong đó :

▪Phạm vi :

▪**LoCal** :chỉ sử dụng trong phạm vi khai báo(mặc định)

▪**Global** :sử dụng chung cho cả kết nối

# Bài 4: Lập trình với SQL Server

## ▪ Di chuyển :

- **ForWard\_Only** : chỉ di chuyển một hướng từ trước ra sau(mặt định)
- **Scroll** : di chuyển tùy ý

## ▪ Trạng thái

- **Static** : dữ liệu trên Cursor không thay đổi mặt dù dữ liệu trong bảng nguồn thay đổi(mặt định)
- **Dynamic** : dữ liệu trên Cursor sẽ thay đổi mặt dù dữ liệu trong bảng nguồn thay đổi
- **KeySet** : giống Dynamic nhưng chỉ thay đổi những dòng bị cập nhật

# Bài 4: Lập trình với SQL Server

## ▪ Xử lý :

- **Read\_Only** : chỉ đọc(mặt định)
- **Scroll\_Lock** : đọc/ghi
- **Câu lệnh select** : không chứa các mệnh đề Into, Compute, Compute by
- **Danh sách cột cập nhật** : là danh sách các cột sẽ thay đổi được
  - ❖ **Mở CurSor:** *Open tên\_Biến\_Cursor*
  - ❖ **Đọc và xử lý dữ liệu trong cursor**
- **Cú pháp :**

*FETCH Hướng di chuyển From Tên\_biến\_Cursor Into Danh sách biến*

# Bài 4: Lập trình với SQL Server

## Trong đó:

- **Hướng di chuyển :**
  - **NEXT**: Di chuyển về sau
  - **PRIOR** : Di chuyển về trước
  - **FIRST** : Di chuyển về đầu
  - **LAST** : Di chuyển về cuối
  - **ABSOLUTE n** : di chuyển đến mẫu tin thứ n tính từ mẫu tin đầu tiên ,nếu  $n < 0$  : tính từ mẫu tin cuối
  - **RELATIVE n** :di chuyển đến mẫu tin thứ n tính từ mẫu tin hiện hành

# Bài 4: Lập trình với SQL Server

- Trong quá trình di chuyển để kiểm tra việc di chuyển có thành công hay không ta kiểm tra biến hệ thống `@@FETCH_STATUS` nếu  $<>0$  thất bại
- ❖ Đóng Cursor :*Close @Tên\_Biến*
- ❖ Giải phóng CurSor khỏi bộ nhớ :*DEALLOCATE @Tên\_Biến*
- ❖ Chú ý :thứ tự các thao tác khi xử lý dữ liệu trên CurSor
  1. Định nghĩa biến Cursor
  2. Mở Cursor
  3. Duyệt và xử lý dữ liệu trên Cursor
  4. Đóng và giải phóng Cursur

# Bài 4: Lập trình với SQL Server

## Ví dụ

```
declare Cur_Vattu Cursor  
for select Mavtu,tenvtu from Vattu  
open Cur_Vattu  
declare @mavtu char(4),@tenvtu varchar(100)  
while 0=0  
begin  
    fetch next from Cur_vattu into @mavtu,@tenvtu  
    if @@fetch_status<>0 break  
    print 'Mã vật tư :'+@mavtu +' Tên Vật tư :'+@tenvtu  
end  
close Cur_Vattu  
deallocate Cur_Vattu
```

# Bài 5: Thủ tục nội tại(Stored procedure)

## 1>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 VB,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 điều tập trung tại server, trong sql server các xử lý này nằm trong ttnt

# Bài 5: Thủ tục nội bộ(Stored procedure)

2>cú pháp xây dựng ttnt

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

# Bài 5: Thủ tục nội bộ(Stored procedure)

Gọi thực hiện ttnt

Exec Ten\_Thu\_Tuc [ @ten\_Tham\_so\_vao= gia\_tri[...]  
@ten\_Tham\_so\_ra=@ten\_Bien output]

LÊNH Return trong TTNT:

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  
Hoặc

Return : Thoát khỏi thủ tục

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]

# Bài 5: Thủ tục nội tại(Stored procedure)

## 3>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

# Bài 5: Thủ tục nội tại(Stored procedure)

## C> 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’*

*set @bien\_tra\_ve= giá trị lỗi Hoặc Return giá trị lỗi*

*Return*

*End*

# Bài 5: Thủ tục nội bộ(Stored procedure)

- Kiểm tra ràng buộc khóa ngoại (Foreign Key)

- Cú pháp

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

*begin*

*print ‘thông báo lỗiI’*

*set @bien\_tra\_ve= giá trị lỗiI Hoặc Return giá trị lỗi  
Return*

*End*

- Kiểm tra ràng buộc Miền giá trị (Check)

*if @ten\_cot không nằm trong miền giá trị*

*begin*

*print ‘thông báo lỗiI’*

*set @bien\_tra\_ve= giá trị lỗiI Hoặc Return giá trị lỗi*

*Return*

*End*

# Bài 5: Thủ tục nội bộ(Stored procedure)

- **Ghi dữ liệu vào bảng**

- Cú pháp

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

# Bài 5: Thủ tục nội tại(Stored procedure)

## d>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\_cot\_khoa\_chinh)  
begin*

*print ‘thông báo lỗi’*

*set @bien\_tra\_ve= giá trị lỗi*

*Return*

Hoặc

*Return giá trị lỗi*

*End*

# Bài 5: Thủ tục nội bộ(Stored procedure)

- 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

# Bài 5: Thủ tục nội tại(Stored procedure)

## d>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\_nhieu where  
Ten\_Cot\_khoa ngoai=@ten\_cot\_khoa\_chinh)*

*begin*

*print ‘thông báo lỗi’*

*return*

*End*

# Bài 5: Thủ tục nội bộ(Stored procedure)

- xóa dữ liệu vào bảng

- Cú pháp

Delete from Ten\_Bang

Where ten cot khoa chinh=@ten cot khoa chinh

Chú ý : ngoài các lệnh trên có thể còn các lệnh cập nhật dữ liệu  
của các bảng liên quan

# Bài 5: Thủ tục nội tại(Stored procedure)

## 4>Giao tác :(Transaction):

✓ Là tập hợp các lệnh sẽ được thực hiện nếu tất cả đều thành công, nếu có một lệnh thất bại ,thì sẽ không có lệnh nào được thực hiện

✓ Ví dụ : giao tác chuyển tiền của ngân hàng : chuyển số lượng N từ tài khoản A sang tài khoản B, các công việc được thực hiện :

$$\cdot \text{TaiKhoanA} = \text{TaiKhoanA} - N$$

$$\cdot \text{TaiKhoanB} = \text{TaiKhoanB} + N$$

✓ Hai công việc này sẽ được thực hiện nếu không có lệnh nào gây lỗi

# Bài 5: Thủ tục nội tại(Stored procedure)

## Cú pháp xây dựng 1 giao tác trong SQL Server

- Lệnh Bắt đầu 1 Giao tác : *Begin Tran*
  - Lệnh kết thúc thành công 1 giao tác : *Commit Tran*
  - Lệnh kết thúc thất bại 1 giao tác : *Rollback Tran*
- Để kiểm tra các lệnh thực hiện có thành công hay không : truy cập đến giá trị của biến *@@Error*, nếu *@@Error<>0* : thất bại

# Bài 5: Thủ tục nội bộ(Stored procedure)

Cú pháp :

*Begin Tran*

*< tập các lệnh>*

*If @a@error <> 0*

*begin*

*print ‘giao tac that bai’*

*Rollback tran* 

*end*

*Else*

*Commit tran* 

# Bài 5: Thủ tục nội tại(Stored procedure)

**Thủ tục nội tại trả về Cursor :** Thủ tục trả về một Cursor chứa các dòng dữ liệu thỏa điều kiện

❖ Cú pháp truyền tham số trả về kiểu Cursor :

*@Ten\_bien Cursor Varying Output*

❖ Các xử lý :

➤ Bên trong thủ tục :

✓ Gán giá trị cho Cursor, cú pháp :

*Set @Ten\_Bien=CurSor [phạm vi] [di chuyển][trạng thái][xử lý]*

*For câu lệnh Select*

✓ Mở Cursor

# Bài 5: Thủ tục nội bộ(Stored procedure)

➤ *Ngoài thủ tục :*

- ✓ *Duyệt và xử lý dữ liệu trong Cursor*
- ✓ *Đóng và giải phóng Cursor*

# Bài 6:Hàm Do Người Dùng Định Nghĩa

## 1>Khái Niệm:

- Là đối tượng mới được bổ sung vào SQL Server 2000
- Mang đầy đủ tính chất của một hàm : có thể có tham số vào, xử lý và trả về kết quả
- Các loại hàm :
  - Hàm xác định (deter-ministic) : luôn trả về 1 giá trị khi nhận các giá trị truyền vào như nhau
  - Hàm không xác định (non-deterministic) : cho giá trị khác nhau tùy thời gọi như hàm Getdate()..

# Bài 6:Hàm Do Người Dùng Định Nghĩa

## 2>Xây dựng Hàm:

A> 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**

# Bài 6:Hàm Do Người Dùng Định Nghĩa

Ta có thể gọi các hàm loại này trong câu truy vấn, trong lệnh tạo bảng,trong thủ tục nội tại ...

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)

Gọi thủ tục loại này giống như view

# Bài 6:Hàm Do Người Dùng Định Nghĩa

C> **hàm tạo bảng** : tạo và trả về một bảng, trong hàm có thể chứa tất cả các lệnh T-SQL

Cú pháp :

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

Returns **@Ten\_Bang\_trả\_về Table**

(**Tên\_cột Kiểu\_dữ\_liệu[,...]**) as

**Begin**

**<các lệnh T-SQL>**

**Insert into @ten\_bang\_tra\_ve ..**

**Return**

**end**

# Bài 7:Trigger

## 1>Tổng quan:

Trigger là một dạng đặt biệt của thủ tục nội tại nhưng :

- Trigger không có tham số
- Không gọi thực hiện bằng lệnh Exec, mà tự động kích hoạt khi dữ liệu trên bảng có liên quan đến Trigger được cập nhật
- Một Trigger được tạo cho một bảng và dùng để kiểm tra các ràng buộc toàn vẹn phức tạp hoặc cập nhật dữ liệu của các bảng liên quan

# Bài 7:Trigger

## 2>xây dựng Trigger:

Cú pháp

*Create Trigger Ten\_trigger on Ten\_Bang/Tên View*

*For/Instead of Insert[,Update,Delete]*

*AS*

*<Tập lệnh T-SQL>*

# Bài 7:Trigger

## Trong đó :

- *Tên bảng/view : là bảng/view mà trigger được tạo*
- *Nếu Trigger được khai báo với từ khóa For :trigger này sẽ được kích hoạt sau khi dữ liệu đã cập nhật vào bảng*
- *Nếu Trigger được khai báo với từ khóa Instead of :trigger này sẽ được kích hoạt trước khi dữ liệu đã cập nhật vào bảng, thường được dùng để kiểm tra dữ liệu cập nhật trên View*
- *Insert,Update,Delete : Trigger được kích hoạt ứng với hành động thêm,sửa ,xóa dữ liệu trong bảng*
- *Thứ tự kiểm tra các ràng buộc toàn vẹn dữ liệu trong bảng : Trigger Instead of->constraint-> trigger For*

# Bài 7:Trigger

## 3> Hai bảng tạm Inserted và Deleted:

- 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 , chỉ truy cập được hai bảng này trong Trigger

# Bài 7:Trigger

## 4> Các loại Trigger:

- Thông báo lỗi trong Trigger : để thông báo lỗi trong trigger ta dùng hàm :

*Raiserror('Chuỗi thông báo lỗi',16,1)*

- Không cho thay đổi dữ liệu :

*Rollback Tran*

## A> Các xử lý trong Trigger kiểm tra khi thêm dữ liệu vào bảng

- Khai báo biến : trong bảng có bao nhiêu cột tương ứng khai báo bấy nhiêu biến

- Dùng câu lệnh Select lấy dữ liệu từ bảng Inserted gán cho các biến này

- Kiểm tra ràng buộc toàn vẹn giống thủ tục nội tai thêm dl

# Bài 7:Trigger

**B>Các xử lý trong Trigger kiểm tra khi Sửa dữ liệu vào bảng**

- Kiểm tra không cho phép cập nhật các cột
- If Update(Ten\_Cot)

*Begin*

*Raiserror('không thay đổi dữ liệu cột Ten\_Cot',16,1)*

*Rollback Tran*

*end*

- Khai báo các biến lấy dữ liệu để kiểm tra
- Dùng câu lệnh Select lấy dữ liệu mới được cập nhật từ bảng Inserted , lấy dữ liệu trước khi cập nhật từ bảng Deleted

# Bài 7:Trigger

C>*Các xử lý trong Trigger kiểm tra khi xóa dữ liệu vào bảng*

- Khai báo các biến lấy dữ liệu để kiểm tra
- Dùng câu lệnh Select lấy dữ liệu mới xóa từ bảng Deleted
- Kiểm tra ràng buộc toàn vẹn giống thủ tục nội tại xóa dl

# Bài 8 : Kỹ thuật lập trình Vb với SQL Server

## I. Liên kết dữ liệu VB và SQL Server

### A. Cú pháp

Giống như việc liên kết dữ liệu giữa VB và Access, việc liên kết giữa Vb và SQL Server cũng được thực hiện thông qua đối tượng *ADODB.Connection*

Định giá trị của chuỗi kết nối - ConnectionString

```
ConnectionString = "Provider=SQLOLEDB.1;Initial  
Catalog=Tên_CSDL; " &  
        "Data Source=Tên_Máy_Tính;User  
Id=Tên_Tài_Khoản;Pwd=Mật_Khẩu"
```

# Bài 8 :Kỹ thuật lập trình Vb với SQL Server

```
Public cnSQL As ADODB.Connection
Public Sub LinkSQL_DB()
    On Error Goto LoiLienKet:
    If cnSQL Is Nothing Then
        Set cnSQL = New ADODB.Connection
    End If
    cnSQL.ConnectionString = "Provider=SQLOLEDB.1;Initial
Catalog=QuanLyBanHang;" & _
        "Data Source=172.29.7.5;User Id=TuyetNhi;Pwd=abc"
    cnSQL.CursorLocation = adUseClient
    cnSQL.Open
    Exit Sub
LoiLienKet:
    MsgBox "Có lỗi trong quá trình liên kết dữ liệu"
End Sub
```

# Bài 8 : Kỹ thuật lập trình Vb với SQL Server

**2. Đối tượng Command :** là đối tượng dùng để gọi thực thi một thủ tục nội tại trong SQL Server

Các thuộc tính :

**ActiveConnection :** đối tượng connection

**CommandText :** tên thủ tục

**CommandType :** kiểu là adCmdStoredProc

**Gọi thực hiện :** Execute

# Bài 8 : Kỹ thuật lập trình Vb với SQL Server

**3.Đối tượng Parameter : dùng để truyền tham số vào ,tham số ra, nhận giá trị trả về trong ttnt**

Khai báo và khởi tạo :

Dim p As ADODB.Parameter

p= cmd.CreateParameter("tents",kieudl,kieuthamso,kichthuoc,giatri)

Trong đó :

**tên tham số : là tên của tham số trong ttnt**

**Kiểu dữ liệu :là kiểu của tham số**

- adDate** :kiểu ngày

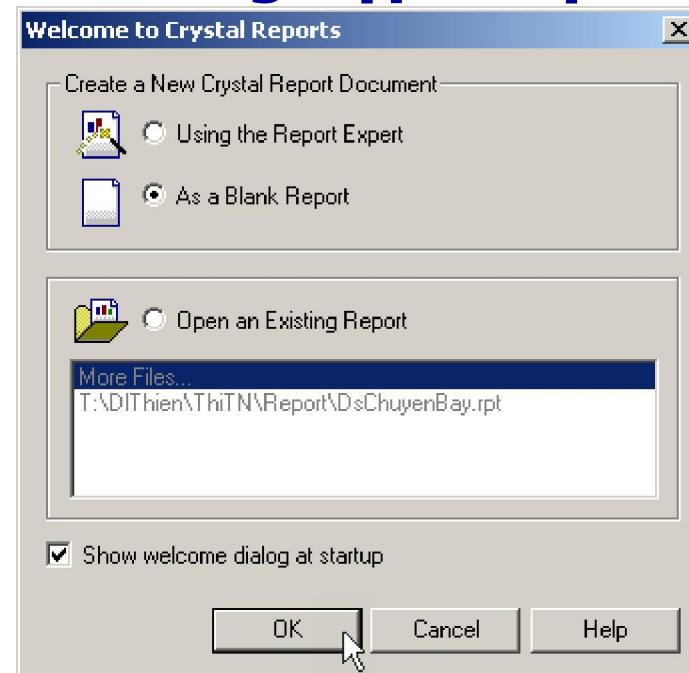
- adBSTR** :kiểu chuỗi

# Bài 8 : Kỹ thuật lập trình Vb với SQL Server

## Tạo báo cáo với Crystal Report

### 1. Tạo báo cáo:

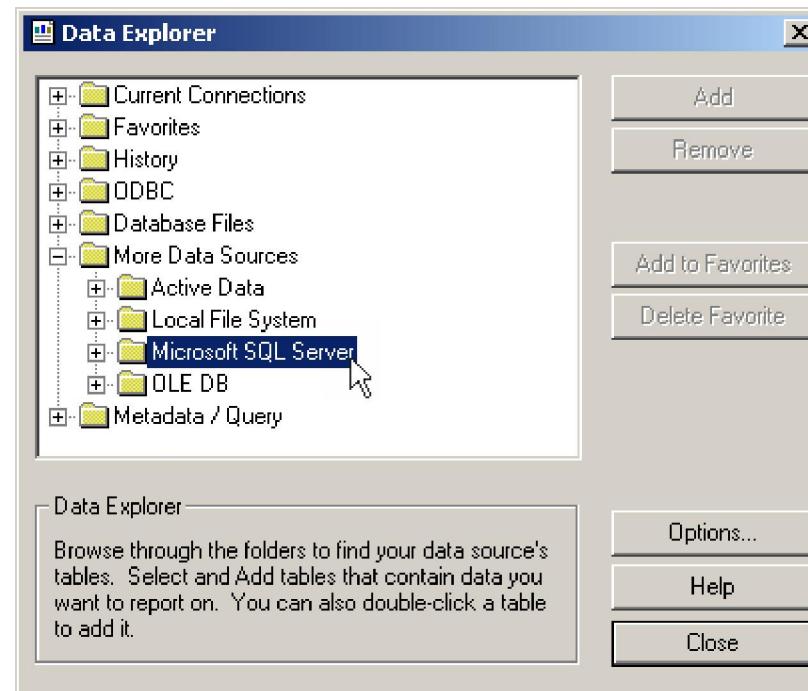
**Bước 1: Chọn As a Blank Report trong hộp thoại Welcome to Crytal Reports**



# Bài 8 : Kỹ thuật lập trình Vb với SQL Server

## Tạo báo cáo với Crystal Report

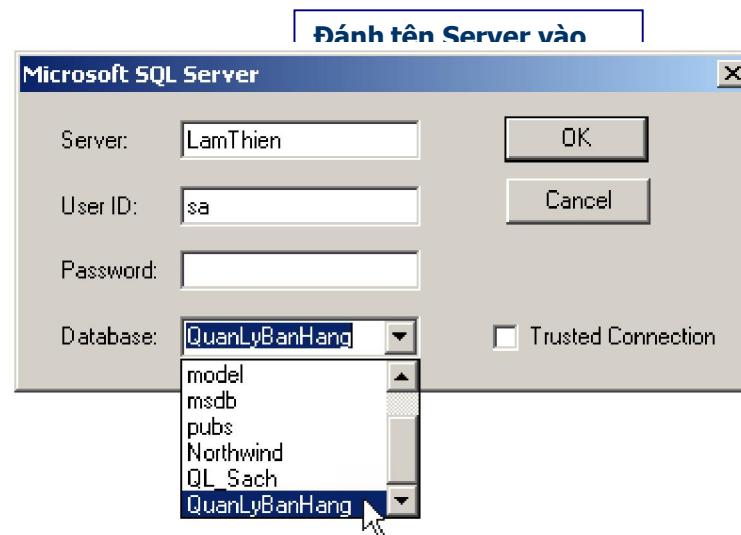
±



# Bài 8 : Kỹ thuật lập trình Vb với SQL Server

## Tạo báo cáo với Crystal Report

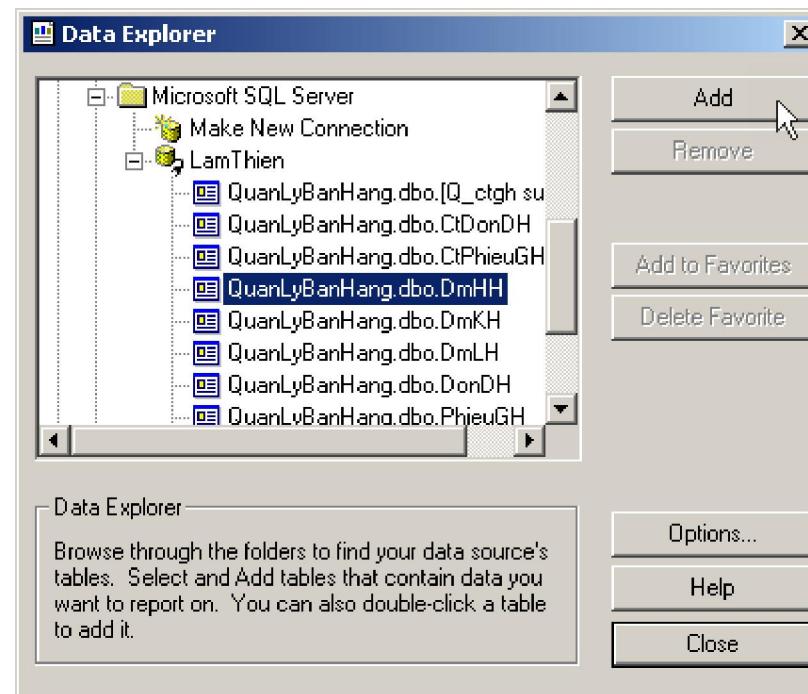
### Bước 3: Điền đầy đủ thông tin cho màn hình đăng nhập



# Bài 8 : Kỹ thuật lập trình Vb với SQL Server

## Tạo báo cáo với Crystal Report

**Bước 4: Chọn các bảng cần thiết cho báo cáo, nhấn nút Add, sau đó nhấn Close**



# Bài 8 : Kỹ thuật lập trình Vb với SQL Server

## Kiểu tham số :

- adParamInput** : tham số vào
- adParamOutput** : tham số ra
- adParamReturnValue** : giá trị trả về

Chiều dài : đối với kiểu dữ liệu chuỗi( char, varchar..) ,ta phải truyền vào chiều dài của dữ liệu

Giá trị : đối với các tham số vào ta phải có giá trị truyền vào

Đưa tham số vào command

**cmd.Parameters.Append tên\_tham\_so**

Thứ tự truyền tham số : tham số trả về,các tham số theo thứ tự được khai báo trong ttnt

## Bài 8 : Kỹ thuật lập trình Vb với SQL Server

```
DK_In.ReportFileName = App.Path & "\baocao.rpt"  
DK_In.Connect =  
"DSN=admin;Uid=sa;pwd=;dsq=Ql_ban_hang"  
DK_In.WindowState = crptMaximized  
DK_In.SelectionFormula = "{Bao_cao_don_dh.sodh}=''" &  
So_Dat_Hang & """  
DK_In.Action = 1
```