

# TỔNG CỤC THỐNG KÊ

TRUNG TÂM TÍNH TOÁN THỐNG KÊ TRUNG ƯƠNG

GIÁO TRÌNH MICROSOFT SQL SERVER 7.0



## HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU

### SQL SERVER



# MỤC LỤC

- I. Giới thiệu chung - Tổng quan về SQL Server (4 - tiết).**
- II. Thiết kế CSDL (4 - tiết).**
- III. Cơ sở dữ liệu trong Microsoft SQL Server (8 - tiết)..**
- IV. Các thành phần của CSDL(12 - tiết)..**
- V. Quản trị và khai thác CSDL (8 - tiết).**
- VI. Tình hình phát triển CSDL thống kê trên SQL Server trong thời gian qua và Kết luận (4 - tiết).**

## **PHẦN I: TỔNG QUAN VỀ SQL SERVER.**

### **1.1. Khái niệm chung, phân biệt CSDL động (quản lý, xử lý các giao tác) và CSDL tĩnh (lưu trữ dữ liệu điều tra).**

Microsoft SQL Server 7.0 là một CSDL kiểu quan hệ được nâng cấp từ phiên bản Microsoft SQL Server 6.5 .

Hệ quản trị CSDL Microsoft SQL Server 7.0 là hệ CSDL đáp ứng tính tăng trưởng, hiệu suất cao cho môi trường tính toán khách chủ phân tán.

### **1.2. Các đặc điểm của hệ quản trị CSDL SQL Server.**

Các tính năng quan trọng chủ yếu như sau:

- + Tích hợp Windows NT gồm các dịch vụ đa luồng và lập lịch, Performance Monitor, Event Viewer. Tương thích mạng đơn, quản lý các khoảng mục thống nhất, giản tiện trong sử dụng.

- + Sử dụng phát tán dữ liệu để trao đổi dữ liệu với các máy chạy trên Windows NT, và phát hành cung cấp dữ liệu cho các máy khác qua ODBC. Nhằm cung cấp cho người sử dụng (Khách hàng) thông tin chính xác và có tính thời điểm.

- + Quản trị tập trung với khung phân tán toàn diện

- + Lập lịch cho các nhiệm vụ, tự động hoá thực hiện theo chu trình có cảnh báo

- + Sử dụng kiến trúc song song cho các chức năng nội tại

- + Sử dụng kiến trúc song song cho các CSDL cực lớn

- + Sử dụng OLE DMO (Distributed Management Object) nhằm cho phép các nhà phát triển phần mềm lập trình cho mọi công cụ cần thiết cũng như các tiện ích giúp đỡ cho các ứng dụng CSDL hoạt động tốt, hiệu suất cao khi sử dụng.

### **1.3. Các phiên bản.**

Microsoft SQL Server có các dạng sau:

- + Microsoft SQL Server 7.0 Standard Edition
- + Microsoft SQL Server 7.0 Enterprise Edition

Ngoài ra các phiên bản sau :

- + Microsoft SQL Server 7.0 office 2000 Edition
- + Microsoft SQL Server 7.0 Desktop Edition

1) Microsoft SQL Server enterprise Edition

Có giao diện đồ họa cho phép người quản trị dễ dàng sử dụng và thực hiện các thao tác cần thiết nhằm duy trì hoạt động.

2) Transact SQL :

Hoàn toàn phù hợp với chuẩn ANSI SQL 92

Tuy vậy, có thêm các mở rộng cho phép tăng các chức năng đáp ứng nhu cầu thực tế.

- + SQL Web Assistant

+ Hai toán tử Cube và Rollup: Tổng kết thông tin trên Server, giảm lưu lượng truyền tin trên mạng, xử lý tại máy trạm và ứng dụng “Kho dữ liệu lớn”.

3) SQL Server DMO (Distributed Management Object) :

**Cung cấp các đối tượng bổ ích như sau:**

- + Transfer
- + Bulk copy
- + Server Group
- + Registered Server.

4) Microsoft DTC:

Cung cấp cơ chế “ lưu trữ hai giai đoạn an toàn” để quản lý xử lý giao dịch giúp :

- + Ứng dụng cập nhật hai hay nhiều hơn SQL Server

+ Dùng lệnh Transact SQL viết các thủ tục lưu trữ nhằm gọi từ xa để cập nhật vào CSDL của SQL Server.

5) SNMP :

Hỗ trợ cho giao thức quản trị mạng, cho phép bất cứ ứng dụng nào cũng có thể cập nhật điều khiển SQL Server thông qua SNMP.

6) Database Maintenance Plan Wizard:

Cho phép tiến hành việc bảo trì CSDL theo kế hoạch.

7) OLAP Server:

Cung cấp cho người dùng các dịch vụ để thiết lập đầu ra (report) theo nhu cầu của người dùng tin. Rất tiện lợi cho người phân tích số liệu.

**Chú ý:**

+ Không hạn chế số lượng máy trạm truy nhập vào Microsoft SQL Server và mọi máy trạm đều phải có giấy phép truy nhập, lưu lượng giao dịch cao, ổn định.

+ Phù hợp cho các môi trường mà tại thời điểm chỉ có một số nhất định các máy trạm truy nhập vào Microsoft SQL Server. Máy cung cấp dịch vụ trực tuyến là một ví dụ: Nếu ta chỉ có 32 đường điện thoại cho khách hàng thì tại một thời điểm chỉ có nhiều nhất 32 khách sử dụng Microsoft SQL Server .

#### **1.4. Cài đặt.**

Vị trí cài đặt ngầm định Microsoft SQL Server 7.0 là c:\MSSQL7\

Trước khi cài đặt Microsoft SQL Server bạn phải biết **Phân bổ đĩa với các mức độ RAID**: RAID 1 và RAID 0+1 bảo vệ dữ liệu và có hiệu suất tốt nhất trong các mức độ RAID nhưng giá thành lại cao hơn giới hạn đĩa yêu cầu. RAID 5 có giá thành phù hợp nhất nhưng lại là sự kết hợp của RAID 1 và RAID 0+1 bởi việc tăng thêm phần vào/ra nên RAID 5 phải đọc và ghi thông tin vào đĩa với số lần tương đương nhau. Do đó, việc sử dụng RAID 5 không hiệu quả bằng RAID 1 và 0+1.

Hiệu suất vào/ra đĩa tốt nhất đạt được với RAID 0 (đĩa không được bảo vệ), chú ý sự khác nhau giữa RAID 0, RAID 1, RAID 5 và RAID 0+1 và để chứa được nội dung của 4 đĩa dữ liệu, RAID 1 (và RAID 0+1) cần 8 đĩa, trong khi đó RAID 5 cần 5 đĩa.

**Tempdb** là một cơ sở dữ liệu nội bộ được sử dụng như một vùng làm việc được chia sẻ cho nhiều hoạt động, bao gồm các bảng tạm thời, sắp xếp và các hàm tổng hợp. **Tempdb** là một vùng hỗn hợp được tạo lại mỗi lần khởi động SQL Server và mỗi lần cập nhật. RAID 1 hoặc 0+1 là lựa chọn tốt hơn RAID 5 cho tempdb. Vì **tempdb** được xây dựng lại mỗi lần khởi động lại máy chủ cơ sở dữ liệu. RAID 0 cung cấp hiệu suất RAID cao nhất cho **tempdb** với số lần ghi vật lý ít nhất. Lý do chính liên quan đến việc sử dụng RAID 0 cho **tempdb** trong môi trường hoạt động là SQL Server cần dừng lại và khởi động lại bất cứ khi nào xảy ra lỗi ghi vật lý trong phân bố RAID 0, điều này không đúng cho trường hợp **tempdb** trong phân bố RAID 1 hoặc 0+1.

**Master device: C:\MSSQL7\data\master.dat**

8) Book online:

Tập các trợ giúp cho các thành phần cấu thành Microsoft SQL Server. Hướng dẫn cho bạn sử dụng như thế nào Microsoft SQL Server 7.0.

**Các lựa chọn cài đặt có đặt trung:**

**Tập ký tự:** Không thể *đổi lại mà không xây dựng* CSDL gồm 256 chữ, số ký tự đặt biệt đặt trung cho mỗi quốc gia hay cho một ngôn ngữ nào đó.

**Chuẩn Latin1-ANSI:** Ngâm định cho:

+ UNIX

+ VMS với Sybase

+ Window, Windows Windows NT

Chuẩn CP 850: Gồm mọi ký tự cho hầu hết các quốc gia châu Âu, Bắc, Nam Mỹ.

Chuẩn CP 437: Chỉ dùng khi có ứng dụng dùng ký tự đồ họa mở rộng.

### **Thứ tự sắp xếp: (Indexes)**

Là tập các quy tắc xác định xem Microsoft SQL Server xử lý dữ liệu trả lời cho mẫu câu hỏi được lựa chọn phụ thuộc vào tập ký tự, nó (Indexes) rất quan trọng vì ảnh hưởng trực tiếp đến :

- + Tập kết quả - Ví dụ: chữ hoa và chữ thường N khác n
- + Hiệu suất: làm tăng tốc độ xử lý nếu ví dụ A=a không cần phân biệt (A khác a, B khác b.. ..)
- + Phát triển ứng dụng cho trạm : rất nhạy cảm cho việc phân biệt chữ hoa và chữ thường.

Người sử dụng phải biết phân biệt lúc nào chữ hoa, lúc nào chữ thường. Đổi lại trật tự sắp xếp dẫn đến “ **xây dựng lại CSDL**” người sử dụng và CSDL **master** vì thông tin hệ thống và CSDL, bảng hệ thống được sắp xếp theo trật tự được chỉ ra, (Hiệu suất quy nhập cao, tránh hiện tượng tắc nghẽn đường truyền)

### **\*) Chọn hỗ trợ mạng:**

Mọi máy đều cần giao thức mạng và thư viện mạng(Net-Library)

+Giao thức mạng: giao thức liên lạc giữa hai máy.

+Net-Library: cho phép kết nối giữa trạm và Microsoft SQL Server

Net-Library triển khai như DLL thực hiện thao tác mạng cần để giao tiếp sử dụng giao thức được chỉ ra. Microsoft SQL Server dùng Net-Library để chuyển các gói thông tin giữa trạm và Microsoft SQL Server.

Named pipes: (Ngâm định-không được bỏ) Nếu cần tích hợp bảo mật với Windows NT thì phải có Named Pipes hay MP Net-Library.

Named pipes cung cấp khả năng giao tiếp giữa trạm và Microsoft SQL Server và cho phép truy nhập đến tài nguyên mạng được chia sẻ. Tên ngâm định:

## \\Tên máy\Pipe\Microsoft SQL Server\query

Không được xoá **named pipes** khi cài đặt hay nâng cấp

+ **Multi-Protocol:** Cung cấp khả năng giao tiếp đồng thời qua một hay nhiều cơ chế IPC hỗ trợ bởi Windows NT Server.

+ **NW linc:** Cho phép Microsoft SQL Server giao tiếp với mạng NetWare Server.

+ **IPX/SPX/(TCP/IP)/Sockets:** Cho phép SQL giao tiếp sử dụng Windows socket như phương pháp IPC qua TCP/IP.

### **\*) Chọn tự động khởi động**

Cài đặt theo ngầm định Microsoft SQL Server và SQL Executive chạy như dịch vụ khởi động thủ công, nhưng có thể chọn khởi động tự động khi cài đặt hay sau đó chọn từ Service trong Control Panel. Các lợi ích khi chạy như dịch vụ tự động thì không phải vào mạng để khởi động lại.

Các cấu hình tối thiểu cho Microsoft SQL Server

	Standard Edition	Enterprise Edition	Office 2000 Edition	Desktop Edition
Phạm vi	Phạm vi nhỏ	Phạm vi lớn	cục bộ	đơn lẻ
Cấu hình	Windows NT Workstation Max 4CPUs 4 GB RAM	Windows NT Workstation Max 4CPUs 8 GB RAM	Windows 95/98 Windows NT No: SMP	Windows 95/98 Windows NT

### **Quy trình cài đặt Microsoft SQL Server 7.0**

Yêu cầu trên máy cài đặt Microsoft SQL Server phải có Internet Explorer 4.01 Nếu là Windows 98 thì bạn phải cài đặt nó bằng cách Install SQL Server 7.0 Prerequisites để cài Internet Explorer 4.01:



Sau đó lần lượt cài đặt theo hướng dẫn của chương trình Microsoft SQL Server .

Ngoài ra còn có Web site, Book online, OLAP service

Trong Microsoft SQL Server 7.0 Component gồm có hai lựa chọn:

+ Microsoft SQL Server 7.0 Standard Edition

+ Microsoft SQL Server 7.0 Desktop Edition

Các thành phần cấu thành Microsoft SQL Server :

1) Service manager:

Quản lý cung cấp các dịch vụ của Microsoft SQL Server

Nút Auto-start service when OS start: khi bạn check vào đây thì khi máy khởi động xong thì tất cả các dịch vụ khởi động theo sự cài đặt của bạn.

2) OLAP Service:

Cung cấp các dịch vụ ứng dụng nhằm cung cấp thông tin cho người sử dụng, khách hàng và các SQL Server khác.

**Chú ý:**

Nếu bạn xoá OLAP Service thì bạn không được xoá RESPOSITORY.

3) Client Network Utility:

Cung cấp thư việc mạng hỗ trợ cho Microsoft SQL Server quản lý CSDL và giao dịch trên mạng. Có hiệu suất cao khi trao đổi thông tin.

4) SQL Enterprise Manager:

Quản lý toàn bộ quyền kết xuất, thiết lập CSDL, tạo quyền bảo mật dữ liệu.. . của Microsoft SQL Server . Gồm có các đối tượng sau:

\*) Database : Trung tâm lưu trữ thông tin của CSDL của Microsoft SQL Server

\*) Data Transformation Service: Trung tâm quản lý các dịch vụ, cung cấp các dịch vụ nhằm chuyển đổi CSDL từ dạng này sang dạng khác.

\*) Management: Điều khiển quản lý các hoạt động của các đối tượng truy cập vào Microsoft SQL Server .

\*) Security: Quản lý các mối quan hệ, liên kết , và quan trọng nhất là chế thiết lập bảo mật CSDL (xa hay gần) của Microsoft SQL Server

\*) Support Service: Bạn có thể thêm các loại dịch vụ cần thiết mà bạn cần sử dụng vào đây.

+ DT Coordinator:

+SQL Mail.

\*) Import and Export data:

Cho phép người dùng các tiêu thức để truy xuất ra các dạng dữ liệu cần thiết. Trong đó cho phép bạn sử dụng các công cụ lập lịch, các hàm ứng dụng có thể.

\*) Microsoft DTC Administrator Console: Cung cấp cơ chế lưu trữ CSDL hai giai đoạn an toàn, quản lý việc xử lý giao dịch trên mạng

Các cơ chế này trợ giúp các ứng dụng cập nhật hai hay nhiều chiều giữa các đối tượng trên Microsoft SQL Server .

Trợ giúp các ứng dụng viết bằng lệnh Transact SQL trong các thủ tục SP (stored Procedure) và gọi các thủ tục SP xa khác để cập nhật vào CSDL của Microsoft SQL Server.

\*) ProFile:

+ Cho người sử dụng kết nối vào ra CSDL của Microsoft SQL Server.

+ Thực hiện từng khối công việc.

+ Thực hiện các lệnh trong các thủ tục SP được gọi đến.

+ Giải quyết các bế tắc của Microsoft SQL Server và thông báo lỗi.

**Tools của ProFile cung cấp cho chúng ta các chức năng sau:**

+ Registry SQL Server:

Đăng ký các nhóm (group) để truy nhập vào Microsoft SQL Server

+ Edittion SQL Server Registration: Cấu hình của Microsoft SQL Server

+ SQL Server Query Analyzer:

Kết nối vào Microsoft SQL Server database bằng các user\_login hay SA (ngầm định).

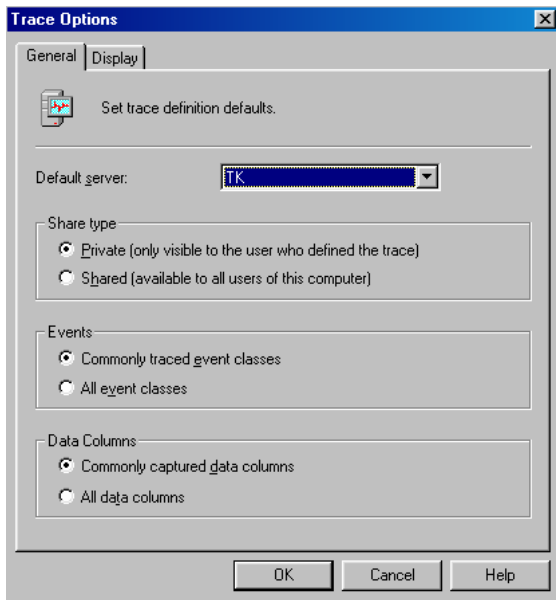
+ SQL Server Enterprise Manager:

+ SQL Server Client Configuration Utility.

Cấu hình của các mối liên kết trong Microsoft SQL Server

+ **Option :**

Các lựa chọn cấu hình cho profile



a) Share type : Kiểu chia sẻ tài nguyên

b) Events : Các sự kiện xảy ra trên Microsoft SQL Server

c) Data Columns: Số lượng các cột dữ liệu được theo dõi.

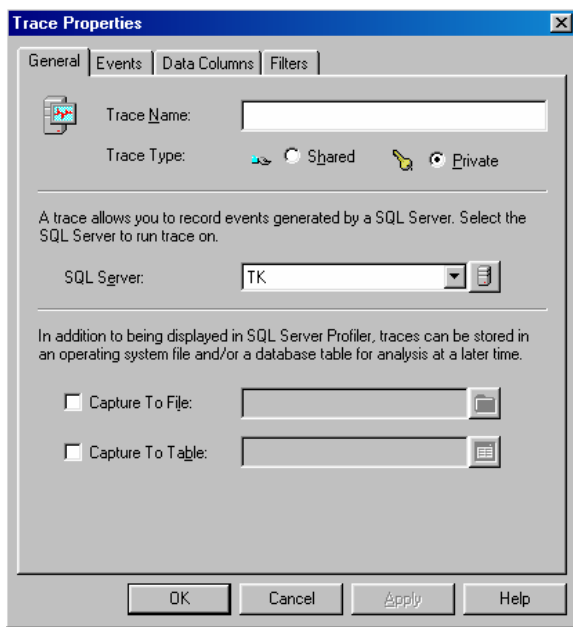
Profile cung cấp cho chúng ta một công cụ rất tiện ích đó là Trace.

Nếu bạn muốn theo dõi quá trình truy cập, các lời cảnh báo và nội dung của các lỗi xảy ra, thời gian truy cập như thế nào. thì bạn phải tạo ra TRACE để làm việc đó.

### **Làm thế nào để tạo ra TRACE?**

- 1) Vào menu File -> Chọn New -> Trace (Nhấn phím CTRL+N)
- 2) Nhập tên TRACE Chọn SERVER\_NAME
- 3) Chọn file lưu trữ nội dung trace (Capture to file):
- 4) Chọn table lưu trữ nội dung trace (Capture to table):

5) OK



\*) Microsoft SQL Server query Analyzer: Cho phép người dùng kết nối vào Microsoft SQL Server bằng user\_name hay logon\_ID có **password** (user do người dùng(**Admin**) định nghĩa)

\*) Server Network Utility: Cung cấp thông tin cấu hình của SQL Server, các giao thức mà nó đang sử dụng, tên Server... .. Cũng như cho phép người dùng thiết lập giao thức truyền tin

### 1.5. Khái niệm về kết nối và truy nhập vào CSDL

Mỗi khi bạn nối vào Microsoft SQL Server bạn phải cần các thông tin sau:

- + Tên Server cần nối.
- + Login\_ID,
- + Password:

Ngoài ra có phương pháp SQL Server DMO Connect cũng dùng 3 thông số trên.

### **Toán tử login. (Login SQL Server)**

Để quản lý (quản trị) SQL Server, ta **login** vào SQL Server. Nếu bạn dùng bảo mật tích hợp thì không cần cung cấp LOGIN\_ID mà SQL Server sẽ dùng khoảng mục Windows NT .

Nếu không dùng bảo mật tích hợp Windows NT thì phải dùng LOGIN\_ID (ngâm định là SA-**Admin**)

#### **Chú ý:**

Khi bạn đã cài xong toàn bộ SQL Server bạn nên kiểm tra lại các thư mục khoản mục , **master** và khoá trong registry:

Hkey\_local\_Machine\Software\Microsoft\MSSQLServer

-Quyền SA có thể đổi Password nếu dùng lệnh **SP\_password**

Thiết đặt các lựa chọn còn lại:

- + Root directory
- + Master database path
- + Error Log Path
- + Auto start server at boot Time

### **LOGIN bằng Enterprise manager**

Đầu tiên ta thêm (Add) các User (NSD) hay các nhóm User vào (Cần tên server, nhóm các User)

Khi đó bạn phải tạo LOGIN\_Name (có password)

### **LOGIN bằng lệnh Transact SQL .**

SP\_Addlogin “Login\_name”,”Password”,”[,defdatabasse]”

### **Thế nào là SQL login server?**

Nó gồm 3 vấn đề cần được quan tâm:

- + SA (**SYSTEM ADMINISTRATOR**)
- + DBO (Database Owner)

+ Guest (Guest user)

- i) SA: ở dạng ngầm định là quyền **SysAdmin** trong Fixed Server Roles , không được thay đổi. Nó được sử dụng cho người Administrator để tạo các user\_name hay quản lý các quyền của hệ thống, gọi là quyền quản trị **Admin**
- ii) DBO: dạng ngầm định , cũng là một thành phần của **SysAdmin** trong Fixed Server Roles. Nói chính xác DBO là quyền tối cao nhất chủ sở hữu CSDL (database).

VD: Nếu user **khanht** là bộ phận của SysAdmin thì khi thiết lập 1 bảng T1 nào đó, khi đó T1 chỉ hạn chế ở **DBO.T1** chứ không phải là **khanht.T1** . Nhưng nếu **khanht** là thành phần của DB\_Owner thì bảng T1 sẽ được **khanht** quản lý và có thể ký hiệu **khanht.T1**

Điều quan trọng nhất là : SysAdmin Fixed Server Roles phụ thuộc vào DBO cho nên khi thiết lập quyền cho user thì phải thiết lập cả quyền DB\_Owner.

- iii) **Guest user**: là một Account login không chỉ cho riêng user nào cả. Database nào cũng có thể dùng guest user để login vào. Account guest cũng cho phép bạn thêm và xoá user và nó được ngầm định trong MASTER và TEMPDB. Tuy nhiên nó không có được đầy đủ các quyền như (SA) hay (DBO).

## **PHẦN II: THIẾT KẾ CSDL**

### **2.1 Những yêu cầu cơ bản và yêu cầu phát triển của một thiết kế CSDL.**

Trước khi thiết lập CSDL, điều quan trọng nhất đối với người thiết kế là phải hiểu rõ các thành phần cấu thành CSDL mà bạn phải thiết kế. Các mối liên kết, quan hệ giữa các đối tượng trong CSDL phải đúng, chính xác, có ngữ nghĩa về mặt thông tin để cho một CSDL có ý nghĩa dữ liệu.

Có hai thành phần chính trong database được quản lý đó là thành phần điều khiển dữ liệu và các mối liên kết vật lý.

Thành phần điều khiển dữ liệu là quá trình thu thập thông tin dữ liệu, thay đổi và update dữ liệu. Vậy Database là một loại CSDL động.

Các thành phần liên kết vật lý của database : là các mối liên kết trong database cũng như các cấu trúc vật lý của nó dưới sự quản lý Microsoft SQL Server.

#### **Cơ sở dữ liệu (CSDL):**

Là tập hợp các table cùng các tập thuộc tính gọi là trường dữ liệu được kết nối với nhau bằng các quan hệ (liên kết) giữa các table với nhau tạo thành một CSDL.

Một CSDL được gọi là một CSDL có ngữ nghĩa nếu nó không tồn tại các thông tin thừa, không có nghĩa.

Mỗi khi CSDL đã được thiết lập thì tại mọi thời điểm nó có thể cung cấp cho người sử dụng hay khách hàng thông tin chính xác tại mọi thời điểm. Tất nhiên là nó phải có cấu trúc của một hệ CSDL kiểu quan hệ.

#### **Định hướng cho tương lai:**

Trong suốt quá trình khai thác CSDL, nếu chúng ta muốn mở rộng thêm CSDL này thì trước hết CSDL này phải được thiết kế dưới dạng mở, tiện lợi để thay đổi khi có nhu cầu phát triển.

#### **CSDL phải đảm bảo các yêu cầu sau:**

- + Có cấu trúc.
- + Dễ dàng sửa chữa, Update, modify dữ liệu
- + Dễ dàng khai thác thông tin.
- + Dễ dàng phát triển và xây dựng các ứng dụng (application)

### **Table và các thành phần của TABLE.**

**Dữ liệu:** Là thông tin được chứa trong các cột của mỗi TABLE và gọi là hàng dữ liệu(Hay record – bản ghi dữ liệu).

Ví dụ:

Khánh	65kg	1,69	kinh
-------	------	------	------

#### **Thông tin của dữ liệu :**

Là gồm các mô tả về cấu trúc các dữ liệu và thông tin ngữ nghĩa của dữ liệu đó. Ví dụ:

Tên	Cân nặng	Cao	Dân tộc
Khánh	65kg	1,69	kinh
..	..		..

#### **Bảng dữ liệu:**

Bảng (Table) bao gồm Tên, cấu trúc các trường dữ liệu, kiểu dữ liệu của trường dữ liệu... .. tạo thành các hàng (record) dữ liệu.

Thông tin ở trong mỗi bảng (TABLE) phải mô tả đầy đủ về một đối tượng nào đó hoặc được kết nối với một số các table khác có mối liên hệ với đối tượng này. Không được có hiện tượng dư thừa thông tin cần mô tả, ví dụ như các thông tin về người, khách hàng.. (một CSDL phải đảm bảo các yêu cầu cơ bản về thuyết CSDL kiểu quan hệ).



### **Field- Trường dữ liệu:**

Là các thuộc tính của một bảng (table) nhằm mô tả các đặc trưng riêng của từng đối tượng mà bạn đang mô tả. Trường dữ liệu có thể là khoá (có thể phụ thuộc hay không phụ thuộc vào các trường (field) của bảng khác) hoặc là mô tả thông tin của một thuộc tính.

### **Một trường dữ liệu đầy đủ cần:**

- + Tên trường
- + Kiểu dữ liệu của trường.
- + Chiều dài của trường (độ rộng)
- + Độ chính xác
- +.. ..

### **Record-Bản ghi**

Là một hàng dữ liệu của một bảng (TABLE) trong CSDL, mang thông tin mô tả về một đối tượng nào đó mà các thành phần mô tả là các cột dữ liệu (thuộc tính của đối tượng) của bảng.

### **Khoá:**

Khoá là tập hợp các thuộc tính độc lập với nhau có ngữ nghĩa nhất định, được mô tả trong các bảng (table) của CSDL có mục đích liên kết giữa các bảng(table) với nhau. Có hai mức khoá chính là

- + Primary Keys: Khoá trong (Xem phần khoá)
- + Foreign Keys: Khoá ngoài

### **Indexes:**

Điều quan trọng nhất là thứ tự sắp xếp dữ liệu trong các table của CSDL là bởi vì nó quyết định đến hiệu suất truy cập, tốc độ và cũng như độ chính xác của dữ liệu.(Xem thêm phần INDEXES)

## **2.2. Các công việc cần chuẩn bị trước khi thiết kế.**

- **Mục đích lập CSDL**
- **Thu thập thông tin**
- **Xác định đối tượng**
- **Xác định kiểu thông tin cho mỗi đối tượng**
- **Xác định các quan hệ và kiểu quan hệ**

### **Một CSDL phải đảm bảo các tiêu thức sau:**

- + Tính độc lập dữ liệu, kiểu dữ liệu, kiểu quan hệ dữ liệu.
- + Tính độc lập đối tượng.
- + Các thông tin thu thập thông tin về đối tượng.
- + Xác định chính xác được thông tin của đối tượng cần được mô tả trong CSDL.

### **Yêu cầu thông tin:**

Thông tin của một CSDL là rất quan trọng vì nếu bạn không nắm vững thông tin của các đối tượng thì khi xây dựng CSDL sẽ dẫn đến sai lệch thông tin về CSDL đó.

Thông tin chính xác cho phép bạn thiết kế một CSDL đúng đắn chính xác ..

### **Tính độc lập ở đây có nghĩa là :**

- \*) Đảm bảo không dư thừa thông tin.
- \*) Cung cấp “đủ” thông tin cho người khai thác cũng như khách hàng dùng tin- tức là có ý nghĩa số liệu (con số có ý nghĩa).
- \*) Khi thêm bớt thông tin thì nó có thể trở thành CSDL lớn hơn (có ý nghĩa mở rộng).

\*) Quan hệ của các khoá phải độc lập – tức là không được chồng chéo lên nhau.( Đảm bảo được tính độc lập dữ liệu – Tức là tính bất biến dữ liệu với thời gian)

Ví dụ:

Cust_ID	Name	Adress	Title_ID
123	Khanh	HN	1
124	Thu	ĐN	2

Title_ID	Title
1	Tranning
2	Work

### Quan hệ và các kiểu dữ liệu quan hệ trong CSDL.

Để truy nhập , khai thác dữ liệu có hiệu suất cao, chính xác.. ..thì chúng ta thiết kế một CSDL có các mối liên kết chặt chẽ giữa các table với nhau. Các mối liên kết này liên kết các thuộc tính của các đối tượng (table) lại với nhau.

**Quan hệ:** Là các mối liên kết giữa thuộc tính này với thuộc tính kia giữa bảng này với bảng kia.

#### Có 3 kiểu quan hệ:

a) Quan hệ 1-1: (one –to-one)

Là quan hệ giữa 1 record của bảng này liên kết tương ứng với 1 record của bảng kia.

Ví dụ:

STT	Lop	Ten
1	A	Khanh

2	B	Thu

Lop	Khoi
A	Khối A
B	Khối

b) Quan hệ 1 – nhiều: (One -to -many)

Là quan hệ giữa 1 record của bảng này liên kết kết nối với 1 hay nhiều record của bảng kia.

Ví dụ:

Matinh	Ten tinh
101	Hà Nội
103	Hải Phòng

Matinh	Mahuyen	Tenhuyen
101	01	Q. Ba Đình
101	03	Q. Hoà Kiếm
103	01	Q. Lê Chân

d) Quan hệ nhiều-nhiều: (many - to - many)

Là quan hệ giữa nhiều bản ghi của bảng này liên kết với nhiều bản ghi của bảng kia trong CSDL.

Ví dụ: Tổ hợp khoá (Thaygiao,hocvien) là quan hệ nhiều nhiều

Thaygiao	Hocvien	Lop
----------	---------	-----

Teo	A	A1
Thu	B	B1
Thanh	C	C1
Kha	A	A1

Thaygiao	Hocvien	Khoahoc	Truong
Teo	A	Toán	THọp
Teo	B	Lý	Thọp
Thanh	A	Toán	THọp
Thu	C	Toán	Thọp

**Chú ý:**

- Lời nhắc cho các Cơ quan, khách hàng cũng như người sử dụng nói chung: Các giá trị được mô tả trong các trường dữ liệu là rất quan trọng, mỗi khi các đặc trưng của các trường dữ liệu của các table không rõ ràng, không mô tả hết ý nghĩa số liệu làm cho người sử dụng hay người thiết kế CSDL hiểu sai lệch về thông tin thì như thế nó sẽ ảnh hưởng đến quan hệ trong CSDL (Roles, Default, Stored Procedure, Backend).
- Nếu bạn cài đặt Microsoft SQL Server trên máy chủ Windows NT thì cần phải tạo khoản mục Windows NT cho SQL Executive.

**Có hai dạng khoản mục:**

- + Khoản mục Windows NT với quyền quản trị SQL Executive trước khi cài đặt cần quyền đó để lập lịch cho nhiệm vụ và cảnh báo cũng như phát tán. Hoặc tạo khoản mục trong Account Domain để chia sẻ cho các SQL Server khác hoặc tạo khoản mục tại máy cài đặt.
- + Tạo khoản mục với user manager for Domain:

**Gồm:**

\*) User\_name: SQL EXEC

\*) Fullname : SQL EXECUTIVE

\*) Description: SQL EXECUTIVE

\*) Password: “123456”

User must change password at next logon: Xoá

User can not change password : Chọn

Password never expires: Chọn

Account disabled: Xoá

\*) Nháy đúp vào Group -> Chọn Administrator -> Add -> OK

\*) Close

Trong Policies chọn User Rights cho quyền truy cập CSDL

LOGON AS A Service: Chọn Add → Chọn SQL Exec → Add → OK hoàn thành.

### **Các thông tin có thể truy cập từ xa:**

-Remote Server Name: Tên của máy Server truy cập từ xa.

-SQL Installation drive: Tên ổ đĩa của máy xa để cài đặt.

-Windows NT installation Drive: Tên ổ đĩa mà Windows NT của máy xa được cài.

-Master database installation Drive: Tên ổ đĩa xa để cài CSDL **Master**

### **Các dịch vụ chính:**

+Microsoft SQL Server : Bộ máy CSDL

+SQL Executive: Đối tượng dịch vụ và cấu thành quản lý.

+Microsoft DTC : Quản trị quá trình giao dịch và phân tán.

Các thành phần quản lý Service:

+ SQL Service manager.

+ SQL Enterprise manager

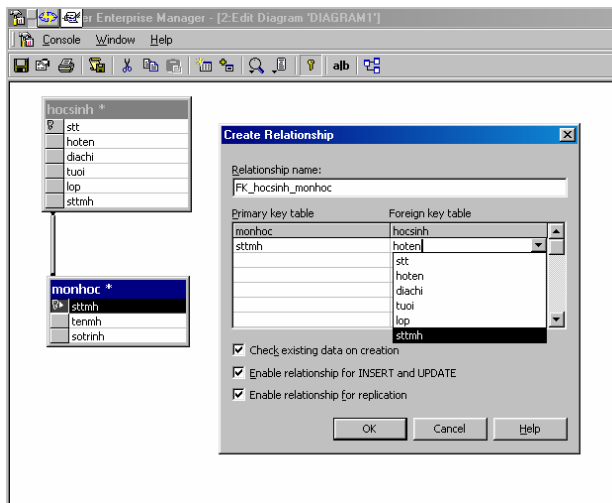
+ Service trong Control Panel.

Microsoft SQL Server có thể khởi động bằng file SQLSERVER.EXE

## 2.3. Giới thiệu về công cụ thiết kế và ví dụ thiết kế (lập sẵn mô hình logic và vật lý)

### Thế nào là Diagram?

**Diagram** là một công cụ do SQL Server cung cấp nhằm tạo điều kiện thuận lợi cho người dùng tạo bảng thiết kế CSDL (design database) rất tiện lợi (các version trước người ta phải dùng PowerPoint design để thiết kế sau đó mới biên dịch thành lệnh Transact SQL). Nếu muốn hiểu một CSDL mà bạn có quyền được xem thì việc đầu tiên là nên xem DIAGRAM.



### Làm thế nào để tạo 1 diagram?

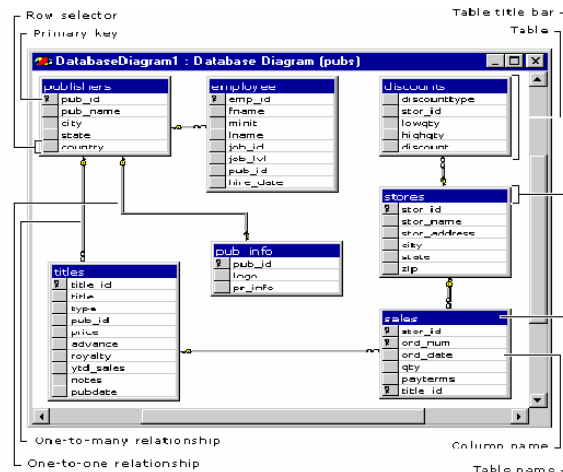
- 1) chọn Diagram-> Nhấn nút phải chuột – chọn New diagram ->Next
- 2) Chọn các loại bảng cần thiết kế CSDL -> Chọn Finish
- 3) Khi này các quá trình liên kết (quan hệ giữa các bảng - **relationship**) sẽ tự động kết nối theo các khoá trong các table mà bạn đã chọn Tuy nhiên nếu các cách thiết kế CSDL của các bạn là đúng đắn. sử dụng

các khoá không dư thừa thông tin thì Diagram sẽ cho ta một mô hình quan hệ của CSDL mà bạn đã thiết kế.

**Chú ý:**

Nếu các mối liên kết không tự động kết nối thì các bạn phải dùng chuột chỉ vào khoá cần liên kết (ứng dụng hiệu ứng Drap and drop) để kết nối mối liên kết này, khi bạn kéo đến khoá thứ hai cần kết nối SQL Server tự động hiện lên tên của hai khoá mà bạn cần kết nối. Khi này bạn chỉ việc chọn nó.

Ví dụ : Ta có các loại table sau: khi ta chọn xong các bảng được kết nối và nhấn nút Finish thì ngay lập tức Diagram sẽ cho ta mô hình quan hệ sau.



**+Quản lý Microsoft SQL Server với DMO (Distributed Management Object).**

Microsoft SQL Server cho phép sử dụng VB, VC++,VFP.. ..để tạo các công cụ quản lý theo sở thích tùy ý(cho Microsoft SQL Server ). Vì có thể điều khiển Microsoft SQL Server bằng VB, VC++, VFP.. .. nên ta có thể xây dựng các chức năng quản lý Microsoft SQL Server theo nhu cầu. Vậy ta cũng có thể viết ra các công cụ, ứng dụng nhằm sử dụng các chức năng mà Microsoft SQL Server cung cấp qua công cụ quản trị đồ họa.

Ví dụ: Xem các yêu cầu chứa trong mọi bảng của CSDL, có thể khai thác qua các màn hình khác nhau tương ứng với các công cụ khác nhau của SQL



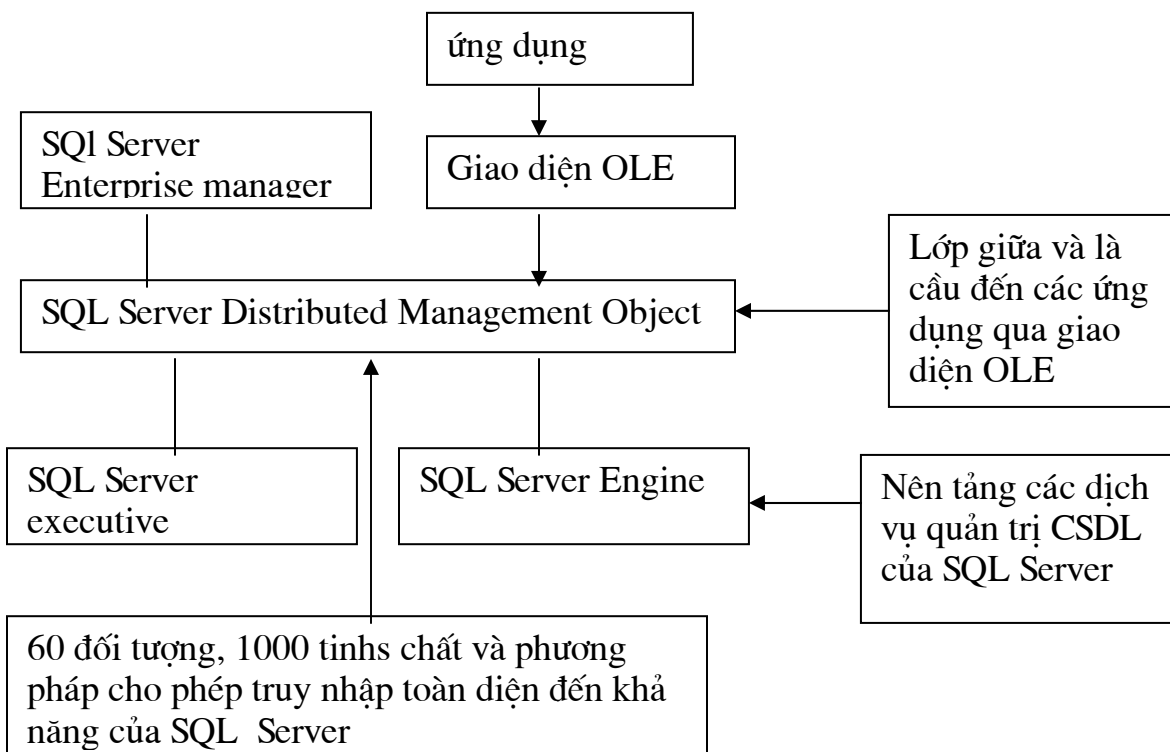
Server Enterprise manager hay viết các chương trình SQL (thủ tục lưu trữ - Stored Procedure), trong ngôn ngữ Transact SQL với một số có hạn các phần tử đồ họa.

Sử dụng DMO cho phép ta xây dựng các điều khiển trong VB, VC++, VFP.. để lấy số liệu từ Microsoft SQL Server và ứng dụng các điều khiển đồ họa (đã có) trong VB, VC++, VFP.

### Thế nào là DMO?

DMO là tập hợp các đối tượng OLE 32 bit cho phép các chương trình truy nhập đến các chức năng quản trị Microsoft SQL Server . DMO cho phép người sử dụng quản trị CSDL phân tán. Chính DMO mở thêm sức mạnh của chức năng quản lý của Microsoft SQL Server cho mọi trạm trong mạng. Vì Microsoft SQL Server triển khai DMO như đối tượng DLE nên ta chỉ có thể dùng nó từ các ứng dụng 32 bit (trong các công cụ VB, VC++, VFP.. ) điều có hỗ trợ 32 bit. (Cũng có thể tạo thành các Tools khi chúng ta dịch các ứng dụng này thành DLL hay OCX )

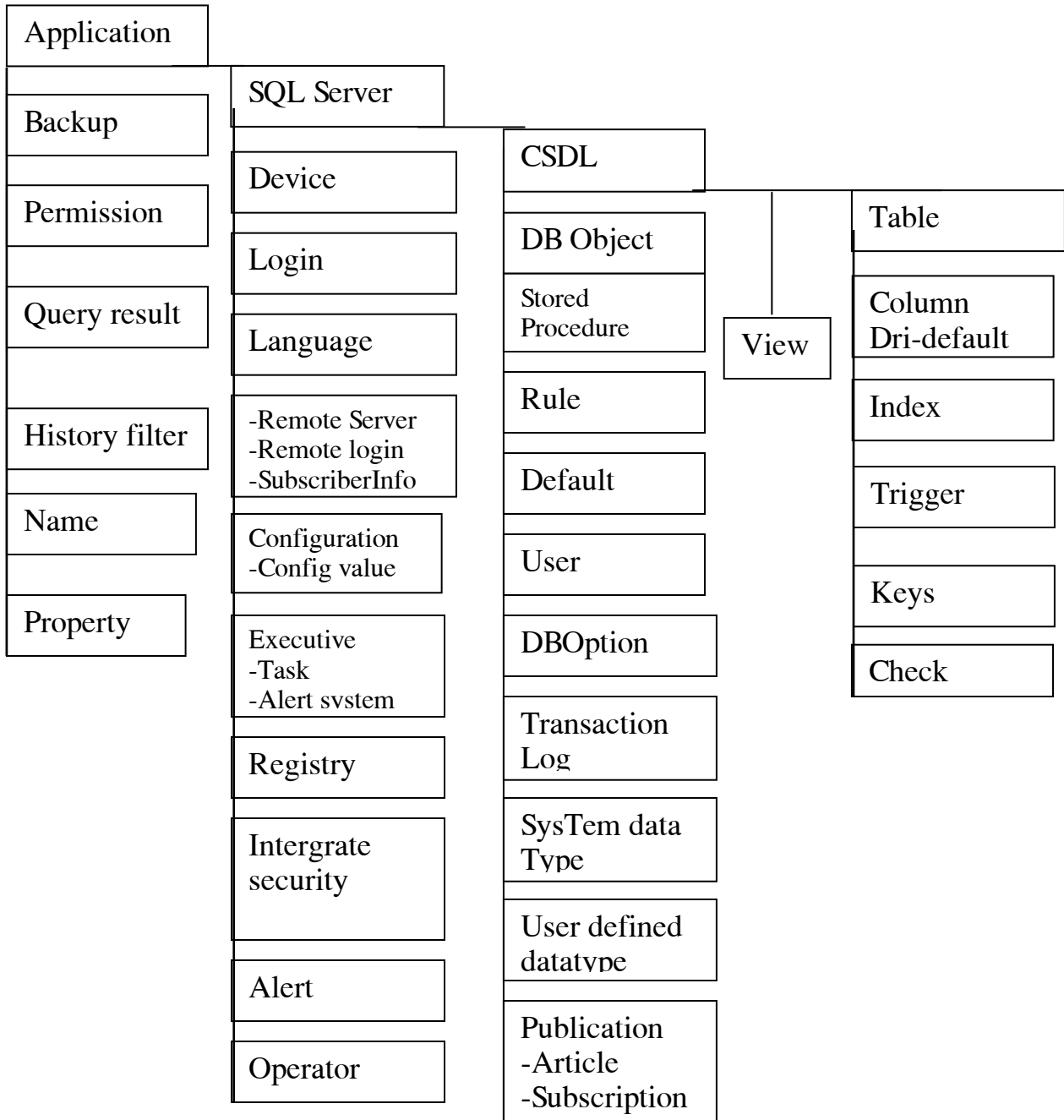
### Sau đây là mô hình kiến trúc 3 lớp của SQL Server DMO



DMO được tổ chức phân cấp như hình vẽ dưới đây, đối tượng ứng dụng (Application) nằm ở đỉnh của tổ chức DMO. SQL Server là đối tượng nằm dưới đối tượng ứng dụng. Một đối tượng SQL Server chứa một hay nhiều đối tượng CSDL. Một đối tượng CSDL chứa các đối tượng bảng DMO và các đối tượng khác của SQL Server như Trigger, Views, hay Stored procedure.. ) Mỗi bậc của phân cấp DMO chứa các đối tượng liên quan đến bậc đó.

Ví dụ: Bậc của SQL Server chứa các Device, login,.. Phân cấp này giống như tổ chức CSDL của SQL Server Enterprise.

*Sơ đồ phân cấp tổ chức CSDL của Microsoft SQL Server*



**Để thêm hỗ trợ DMO vào VB:**

Chọn reference từ Visual basic ->Tools

Tiếp tục chọn Microsoft SQL OLE Object Library. Việc này dẫn đến thêm SQLOLE 32 bit vào VBIDE (chức năng của Microsoft SQL Server DMO ở trong SQLOLE32.DLL) khi bạn đã thêm điều khiển OCXs hay ActiveX vào VB bạn sẽ thấy ở thanh VB Toolbox.

Để làm việc với đối tượng SQL OLE ta dùng Object Browser để xem và thêm các chức năng Microsoft SQL Server DMO.

## PHẦN III: CƠ SỞ DỮ LIỆU

### 3.1. Tạo lập CSDL (bằng Enterprise Manager và bằng Wizazd)

SQL Server enterprise Manager cho phép người sử dụng thực hiện các công việc sau:

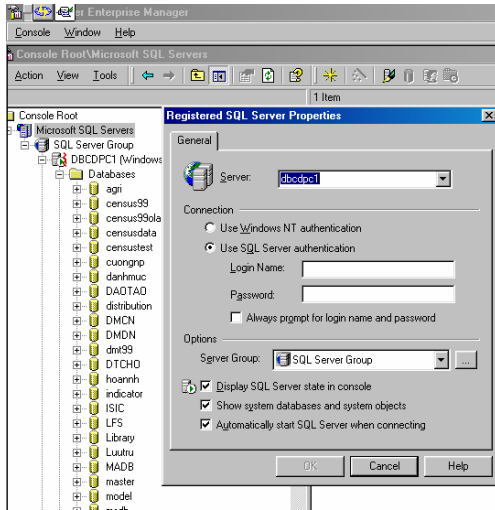
- + Lập lịch
- + Tạo lập CSDL
- + Quản trị khả năng giao dịch và cảnh báo khi có hiệu ứng xảy ra
- + Điều khiển hệ thống đa phương tiện
- + Xây dựng giao diện quản lý dữ liệu
- + Thiết lập font chữ
- + Backup dữ liệu
- + Chuyển đổi dữ liệu

SQL Server enterprise Manager cho phép bạn truy cập đến các thành phần mà nó quản lý trong Tools.

- 1) SQL Server configuration property chứa toàn bộ cấu hình của SQL Server. Nếu bạn muốn thay đổi bạn có thể chọn các mục cấu hình ở đây.
- 2) Đăng ký các Server truy cập từ xa.

->Menu **Action** -> **New SQL Server Registration**

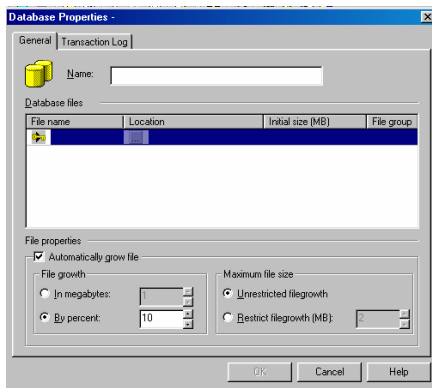
- ➔ Server: Tên Server cần đăng ký.
- ➔ Login Name: Login\_ID cần truy cập
- ➔ PassWord: Mật khẩu để login vào
- ➔ OK.



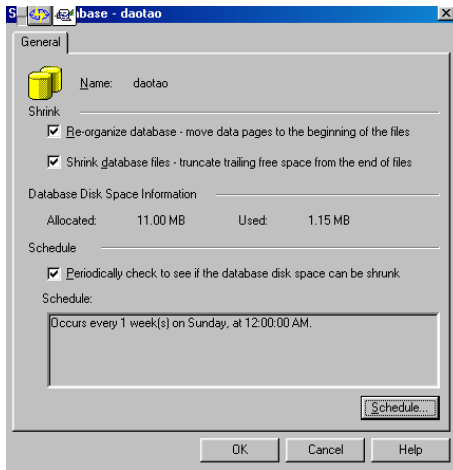
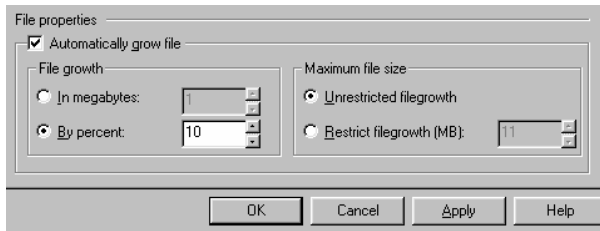
## 1) Thiết kế CSDL(database):

Làm thế nào để thiết kế database?

- Chọn Database->Nhấn nút phải chuột->New database
- Nhập tên CSDL (Database\_name) vào **Name**:
- Các thông số Filename, location, Initial size (MB) :cần phải phù hợp với CSDL mà bạn đã thiết kế.
- File Group: ngầm định là PRIMARY.
- Nếu bạn muốn thay đổi các giá trị ngầm định thì bạn phải thay đổi nó bằng cách nhấn chuột vào Transaction Log.



- Nút Automatically grow file : tự động tăng kích thước database
- Nút In megabytes: dùng trong khoảng bao nhiêu MB
- Nút Restrict Filegrowth: Giới hạn tăng trong khoảng



- Trong File Group: gồm 3 dạng chính

- 1) Primary
- 2) User\_defined
- 3) Default

Log file không bao giờ là thành phần của Filegroup

### 3.2. Các files và nhóm files trong CSDL.

Một CSDL là sự kết hợp của một hay nhiều files dữ liệu cũng như một hay nhiều nhóm(log) files. File dữ liệu có thể là các nhóm file do người dùng định nghĩa. Các bảng và chỉ số được thể hiện qua các mô tả khác nhau được điều khiển sắp xếp trên disk vật lý. Tuy nhiên nếu bạn sử dụng hệ thống RAID thì điều này không có lợi.

Có 3 kiểu của nhóm file:

+ **Primary** : nhóm các file dữ liệu chủ yếu (quan trọng) cùng một số các file khác mà nó không được đưa vào trong filegroup khác. Tất cả các table hệ thống được chỉ định là nhóm **Primary**.

+ **User-defined**: là nhóm file được chỉ định trạng thuộc tính trong SQL Server Enterprise manager.

+ **Default**: Chứa tất cả các table và indexes mà nó không thuộc hai nhóm file trên khi chúng ta thiết lập nó. Trong CSDL, tại mỗi thời điểm có duy nhất một nhóm file được ngầm định (**default**) Nếu không được chỉ định thì giá trị ngầm định của nó là PRIMARY.

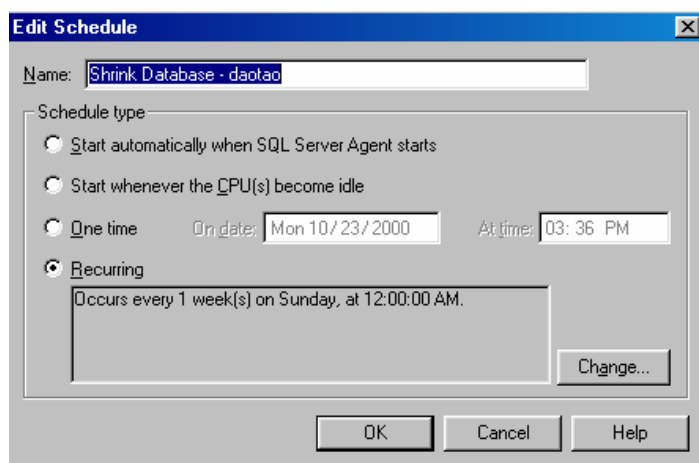
### 3.3. Sửa đổi CSDL, mở rộng CSDL và thu nhỏ CSDL

**Transaction log cung cấp thêm cho chúng ta các thành phần sau:**

- Bulkload/Operation
- Select into statement.
- Modify database
- Tăng hoặc giảm kích thước database :
  - Shrink database : Chọn chức năng này bạn sẽ có các tiêu thức để chọn cắt giảm kích cỡ database.

Cũng có thể dùng lệnh trong Transaction SQL:

```
DBCC Shrinkfile ( { filename | file_ID } { [,target size] } |  
[, { emptyfile | Nottruncate | Truncate_only } ] )
```



#### **Lựa chọn: Click Option**

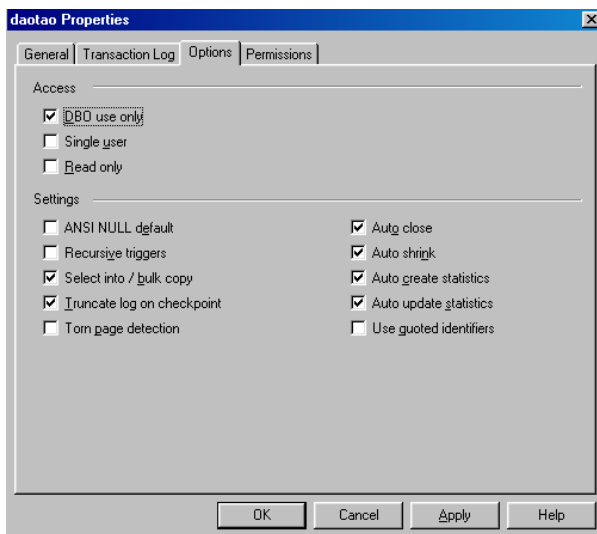
- 1) Database Access:
  - + DBO: Chỉ định quyền DBO
  - + Single user: Tại mỗi thời điểm có một người khai thác dữ liệu
  - + Readonly: Quyền chỉ được đọc
- 2) ANSI null: Ngầm định là Not null
- 3) Select into/Bulk copy: cho phép thực hiện các lệnh



- + Update text
- + Write text
- + Select into statement
- + BCP .. ..
- + Bulk insert statement
- 4) Truncate logon/Checkpoint

Cho phép thực hiện lệnh CHECKPOINT

- 5) Auto close
- 6) Auto Shrink: tự động giảm kích cỡ của file
- 7) Auto create statistic
- 8) Auto Update statistic
- 9) Use quated Identifiers



**Các thành phần khóa và các đặc trưng có liên quan:**

- 1) Primary Key
- 2) Foreign Key
- 3) Unique Constraint
- 4) Default Constraint
- 5) Check Constraint

**+ Primary key:**

“Khoá trong” là trường dữ liệu (khoá) chỉ tồn tại duy nhất trong hàng dữ liệu của một table. Khoá trong không chấp nhận giá trị NULL

**+ Foreign key :**

“Khoá ngoài” là 1 hay nhiều trường dữ liệu được thiết lập để kết nối quan hệ (liên kết) giữa các table với nhau

**+ Toán tử UNIQUE :**

Toán tử UNIQUE có thể giúp cho người sử dụng nó thay cho khoá trong vì :

- a) Trong một cột không bao giờ cho phép có hai giá trị khác hàng lại giống nhau
- b) Nếu bạn sử dụng nhiều toán tử UNIQUE thì có nghĩa là trong table đó có nhiều “khoá trong” được định nghĩa
- c) Cột có chứa giá trị null thì không thể là primary key nhưng ta có thể dùng toán tử này để sử dụng.

**+ Toán tử Check :**

CHECK bắt buộc hạn chế trên phạm vi toàn bộ giới hạn giá trị có thể chấp nhận của các cột. Nó chỉ định các giá trị hợp lệ thoả mãn biểu thức logic cơ bản của các cột dữ liệu khác.

**+Định nghĩa Default :**

Mỗi cột trong record chứa các giá trị, nếu nó ngầm định là NULL thì nó thay thế giá trị NULL này bằng giá trị ngầm định **Default**. Giá trị ngầm định của nó là “UNKNOWN” khi chúng ta không chỉ định được giá trị cho cột đó.

### **3.4. Chuyển các files dữ liệu từ môi trường ngoài vào CSDL**

- Ngoài ra ta cũng có thể sử dụng một số lệnh của Transact SQL để thực hiện chuyển đổi dữ liệu. Ví dụ như lệnh :

+ BCp database\_name..Tablename

{IN|OUT} dataFile [/F firstrow][/L lastrow][/b batchsize][/C ][/U login\_ID]  
[/P Password] [/S Servername].

+ DUMP Database Database\_name to Disk = “path\filename”

+ DUMP table table\_name to Disk = “path\filename”

+ DUMP Transaction Database\_name to Disk = “path\filename”

Nếu muốn xoá dữ liệu khi bị tràn ô nhớ:

+ DUMP Transaction Database\_name with [Truncate\_opnly | No\_log ]

Khi muốn LOAD vào hay phục hồi dữ liệu được BACKUP ra dùng lệnh LOAD .

+ LOAD Table Dbname..Tablename from DISK =”Path\FileName”

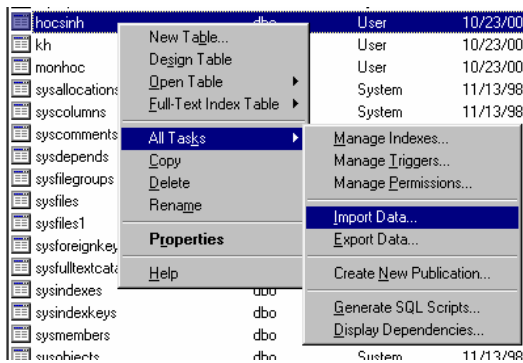
### 3.5. Di chuyển CSDL giữa các máy chủ:

SQL Server cho phép bạn xuất nhập dữ liệu (import and export data) .

Đây là một chức năng quan trọng của Microsoft SQL Server. Nó cung cấp DTS (Distributed Transaction Services) dịch vụ chuyển đổi dữ liệu.

**Gồm có các dạng dữ liệu sau:**

+ ASCII –Fixed –field length text file

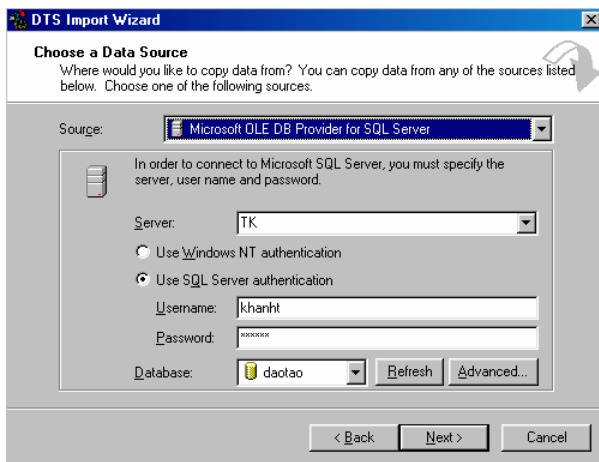


+ ODBC data source

+ SQL Server database

+ Microsoft excel Speadsheets

- + Microsoft Access database
- + Microsoft Foxpro database
- + Dbase or Paradox database
- Làm thế nào để nhập xuất CSDL(database)?
  - 1) Chọn All tasks
  - 2) Chọn import data (hay export data)
  - 3) Nhấn Next



Với hình trên bạn phải vào đủ thông tin dữ liệu nguồn:

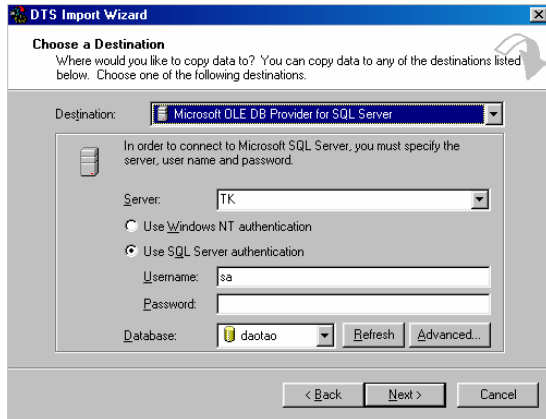
- + Chọn server.
- + Chọn số liệu nguồn

Sau khi chọn xong hai thành phần trên bạn cần phải xác định kiểu dữ liệu xuất/nhập ra.

Sử dụng: Windows NT hay SQL Server

Nếu bạn chọn SQL Server thì phải cung cấp thêm Logon\_ID, password  
+Chọn database : Thường giá trị ngầm định của nó là **default**.

Bạn nên làm tươi (Refresh) lại một lần -> Next ->OK

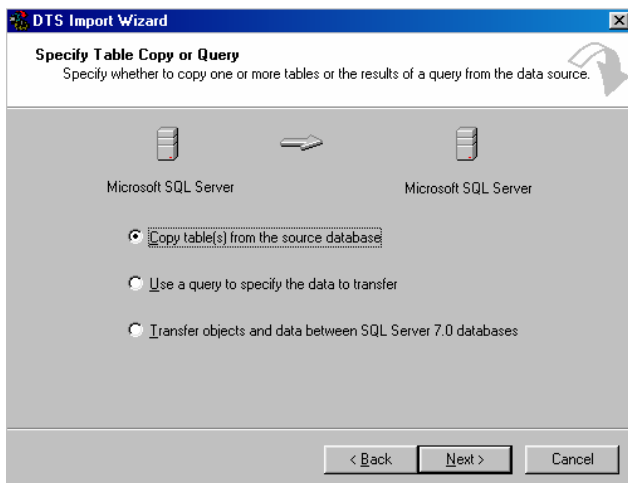


Với hình trên bạn phải vào đủ thông tin dữ liệu đích:

Tương tự như các bước trên bạn chọn từng bước một -> Sau đó chọn Next.

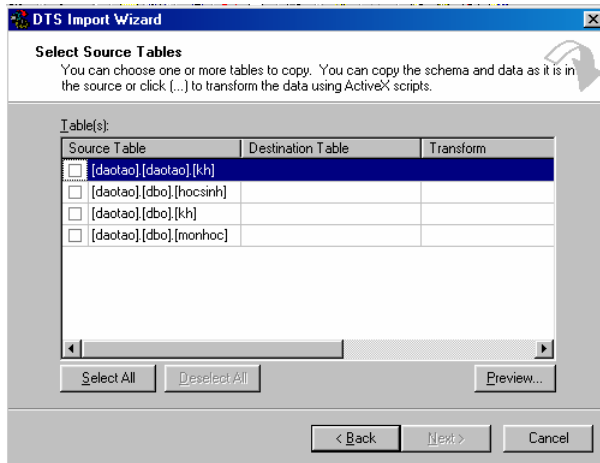
Khi đó bạn phải chọn các loại bảng nguồn được xuất/nhập, copy hay các query nguồn cần chuyển đổi -> Next -> OK

Nếu bạn chọn sai dữ liệu thì kết quả sẽ không đúng. Vì vậy phải cẩn thận khi chọn số



Với hình dưới bạn phải chọn các cột(trường) cần kết xuất sau đó

Nhấn vào nút Finish để hoàn thành công việc

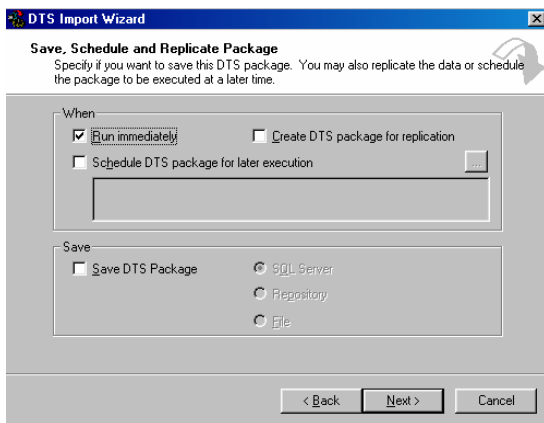


Với hình dưới bạn phải chọn loại dữ liệu cần được kết xuất ra theo các dạng :

- + Runtime immediately.
- + Create DTS package for replication
- + Shedule DTS package for later execution

### **SAVE DTS PACKAGE:**

- + SQL Server: Dữ liệu dạng table SQL server
- +Respository: Đóng gói vào DTS->Respository package
- +File: Dạng file



Tiếp tục -> Nhấn Next (hoàn thành).

## PHẦN IV: CÁC THÀNH PHẦN CỦA CSDL

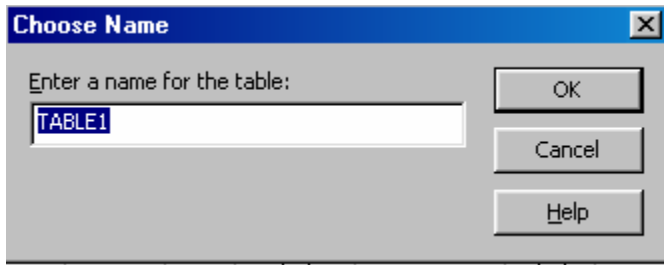
### 4.1. Bảng (bao gồm các thành phần đi cùng)

#### Làm thế nào để tạo 1 TABLE?

Bảng (table) là thành viên cấu thành CSDL chính vì vậy nó đóng một vai trò rất quan trọng trong việc tạo một hệ CSDL.

- Chọn Table-> Nhấn nút phải chuột-> chọn **New database**

Nhập tên của bảng(table\_name) bạn cần tạo vào dòng sau ->chọn OK



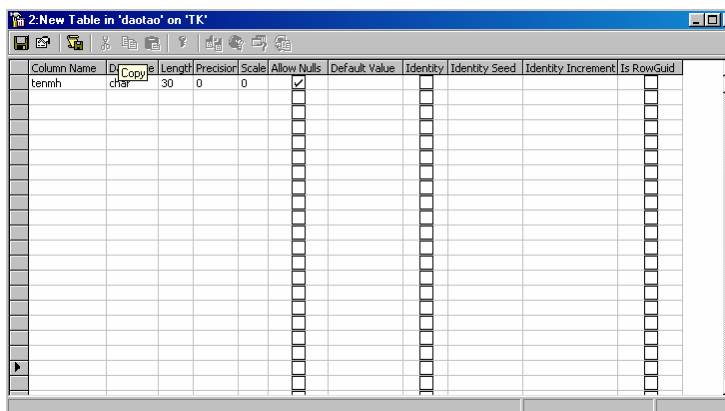
Sau đó lần lượt thực hiện các công việc sau:

- 1) Nhập tên cột (column name)
- 2) Chọn kiểu dữ liệu cho cột (field): (data type)

SQL server cung cấp cho người dùng các dạng dữ liệu sau:

DATA TYPE	DESCRIPTION
-Varbinary	Kiểu nhị phân Max 2 G bytes
-Image	
-Binary	
-Char	Kiểu ký tự (char –1 bytes)
-Varchar	Varchar 0..n bytes
-Text	Text Max 2 G Bytes

-nChar -nVarchar -nText	
-DateTime	Kiểu thời gian (8Byte)
- Smalldatetime	(4byte)
-Int	(-2.147.483.648,+ 2.147.483.647)
-Smallint	(-32.768,+ 32.767)
-Tinyint	0..255
-Decimal - Numeric	
-Float -Real	Kiểu số thực (8 bytes)
-Timestamp -Bit(On,OFF) -(Tru,false)	(0,1)



- 3) Length: Chiều dài của trường dữ liệu
- 4) Precision: Độ chính xác của trường dữ liệu
- 5) Allow null có giá trị null hay không

Nên nhớ rằng trường dữ liệu nào là khoá thì nó nhất định không tồn tại giá trị **null**



6) Default value: Giá trị ngầm định của trường dữ liệu.

Khi mà bạn đã hoàn thành quá trình trên trước khi đóng lại bạn nhớ là phải ghi lại.

- Một table là một thành viên của một CSDL(database); để có ngữ nghĩa về ngôn ngữ quản trị CSDL thì bất cứ một table nào cũng phải có nghĩa trong CSDL đó.
- Dữ liệu được tạo ra không phải user nào cũng có thể truy cập được chính vì vậy mà bạn phải cần thiết lập quyền truy cập cho từng loại table mà bạn đã tạo ra từng chức năng riêng biệt thì càng tốt.

## **4.2. Quan sát.**

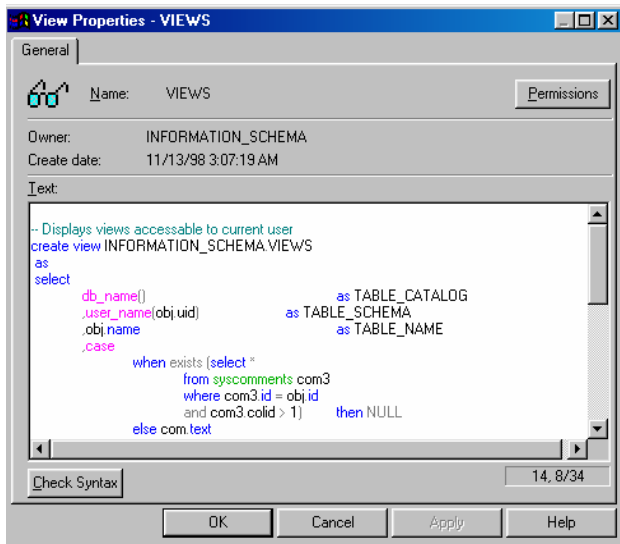
### **Thế nào gọi là Views?**

Là tập hợp các “thủ tục” chứa các đoạn mã Transact SQL lưu trữ toàn bộ nội dung được định nghĩa trong các VIEW đó dưới dạng các query động(real) thực sự. Một View chính là tập hợp các cột của các table trong CSDL. Tuy nhiên nếu database rỗng thì View không thực sự tồn tại

Tại sao ta phải dùng VIEWS?

+ Trong một CSDL : mỗi loại CSDL đều có những đặc trưng quan trọng của nó, chính vì vậy để giảm bớt các băng lưu trữ các thông tin đó trong CSDL người Sử Dụng có thể tạo ra các query trung gian được lưu giữ ngay trung tâm để khi cần dữ liệu chúng ta có thể khai thác ngay mà không tốn bộ nhớ để lưu trữ.

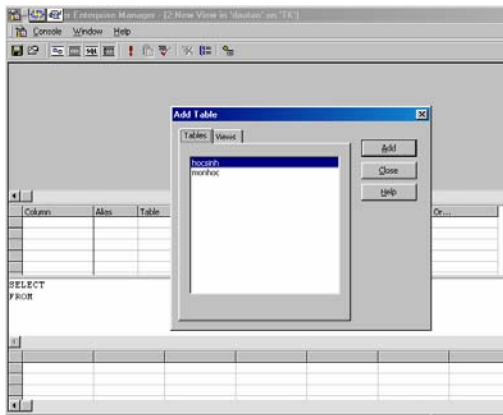
+ VIEWS cũng có những chức năng như một database có nghĩa là nó có thể chia sẻ nhiều mức khác nhau.



### -Làm thế nào thiết lập được VIEWS?

- 1) Chọn Views -> Nhấn nút phải chuột -> New Views
- 2) Chọn các table mà bạn định lấy làm CSDL của VIEWS.

(Hoàn toàn tương tự như lúc bạn chọn các table để thiết kế CSDL trong Diagram)



- 3) Dùng các lệnh trong bộ Transact SQL để soạn thảo trong ô TEXT:

Khi soạn thảo xong bạn nên thực hiện CHECK Syntax.

Bạn hãy dùng các công cụ hỗ trợ trên thanh Toolbar để làm việc với Views.

**Vì Views là dạng query nên ta cũng thiết lập quyền Permission cho nó.**

Thiết lập quyền cho Views hoàn toàn tương tự như thiết lập quyền cho database, nó gồm có các bước chính như vậy.

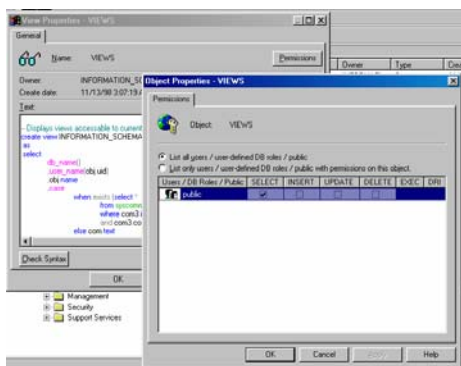
Trong nhóm 1: **user/DB roles/Public**

Gồm các login\_ID và các nhóm làm việc .

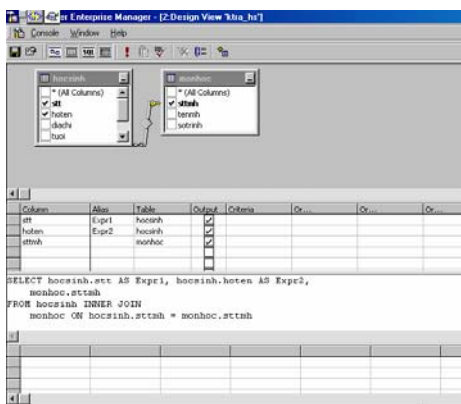
Nhóm 2: Tập các lệnh trong Transact SQL

Nếu bạn check vào ô nào của user/nhóm nào thì user/nhóm đó được quyền sử dụng lệnh đó.

Sau khi đã check các quyền hoàn thành -->Tiếp tục nhấn nút OK (có thể nhấn Apply)



Tiếp theo (lát cắt thứ hai trong Views) là chọn các tên cột dữ liệu (field\_name) trong các bảng đã được chọn để đưa ra trong Views .(hình sau)



Sau khi chọn xong các cột dữ liệu, SQL server tự động biên dịch các thao tác của bạn thành ngôn ngữ Transact SQL được lưu giữ trong (nguồn –Text) của Views.

Ngoài ra bạn cũng có thể check trên các table đã được thêm (Add) vào trên lát cắt thứ nhất. (trên sơ đồ quan hệ của CSDL)

### 4.3. Khoá.

Khoá là tập hợp các tuộc tính độc lập với nhau có ngữ nghĩa nhất định, được mô tả trong các bảng (table) của CSDL có mục đích liên kết giữa các bảng(table) với nhau. Có hai mức khoá chính là

+ Primary Keys: Khoá trong (Xem phần khoá)

+ Foreign Keys: Khoá ngoài

+ **Primary key:**

“Khoá trong” là trường dữ liệu (khoá) chỉ tồn tại duy nhất trong hàng dữ liệu của một table. Khoá trong không chấp nhận giá trị NULL

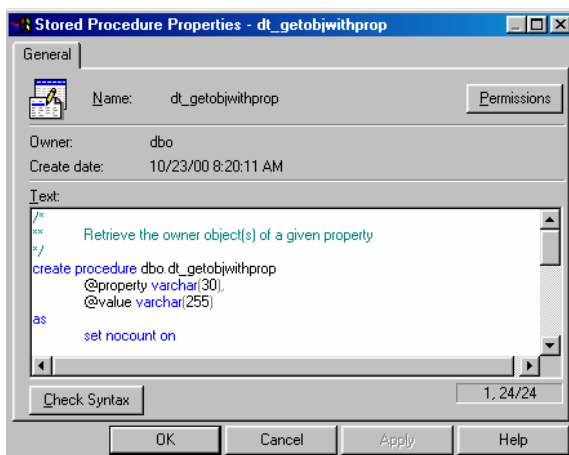
+ **Foreign key :**

“Khoá ngoài” là 1 hay nhiều trường dữ liệu được thiết lập để kết nối quan hệ (liên kết) giữa các table với nhau

### 4.4. Thủ tục ghi nhớ

-**Thế nào là Stored procedure?**

Là tập hợp các thủ tục được thiết lập để quản lý, xử lý và trao đổi thông tin với các ứng dụng với SQL server.



Mỗi khi bạn thiết lập các ứng dụng trên SQL Server, Transact SQL là giao diện quan trọng nhất giữa SQL Server database và ứng dụng, khi đang sử dụng chương trình Transact SQL có hai khả năng thuộc tính xảy ra như sau:

+ Mỗi chương trình sẽ được gán trong phạm vi cục bộ ở client . Ví dụ Visual Basic gửi một lệnh đến SQL Server thì nó đã được xử lý.

+ Hoặc là chương trình này sẽ tập trung tại SP trong SERVER.

- Sử dụng SP có lợi ích hơn là sử dụng lệnh Transact SQL ở máy client là bởi vì:

- 1) Thực hiện nhanh
- 2) Giảm lưu lượng giao dịch trên mạng
- 3) Bảo mật tốt ngay tại Server
- 4) Mã nguồn lưu trữ ngay tại trung tâm
- 5) Có thể sử dụng cho nhiều ứng dụng khác nhau cũng như các client khác nhau.

### **Làm thế nào để tạo SP?**

- 1) Chọn Stored procedure -> nhấn nút phải chuột -> chọn New **SP**
- 2) Dùng ngôn ngữ Transact SQL để soạn thảo SP.
- 3) Save

SP cũng tương tự như Views: người thiết lập ra SP có thể chia sẻ tài nguyên nếu cung cấp quyền sử dụng các SP này cho các client hay một số các ứng dụng khác trong các Server khác ( gần hay xa)

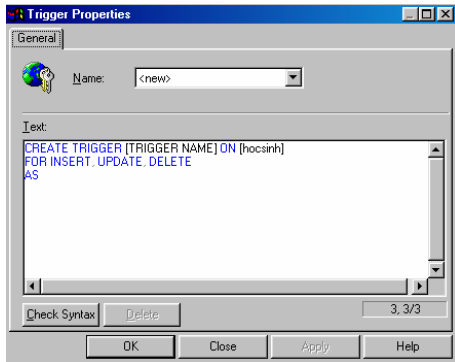
### **4.5. Thế nào là Trigger?**

Là dạng đặc biệt của **Stored procedure**, cũng là dạng thủ tục được chia sẻ như SP nhưng thực hiện (ngầm định là automatic ) khi đáp lại các lệnh (Insert, Update, Delete) mỗi khi được thực hiện. Một Trigger có thể là một query và cũng có thể là một đoạn mã lệnh của Transact SQL.

### **Tạo một Trigger.**

Chọn All tasks

- 1) Chọn manager trigger
- 2) Chọn New
- 3) Dùng các lệnh của Transact SQL để soạn thảo
- 4) Check Syntax
- 5) Save



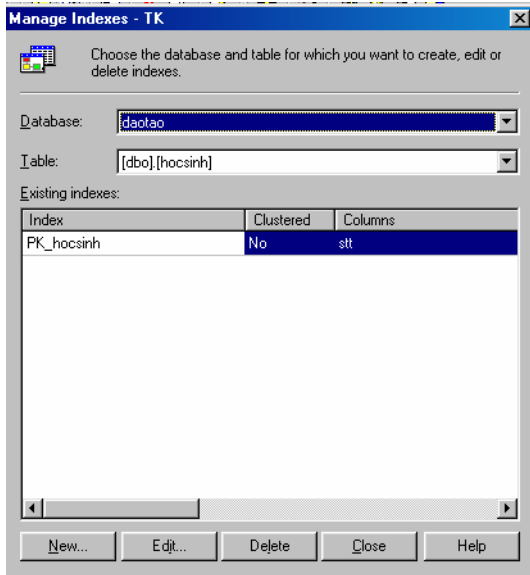
#### 4.6. Chỉ số - Indexes:

SQL Server tự động quản lý indexes cho các khoá trong cũng như các thông số: UNIQUE constraint. Tuy nhiên mỗi khi thực hiện INDEXES một bảng nào đó thì bảng này cần nhiều hơn bộ nhớ mà nó đang có để thực hiện index. Nếu mỗi lần thực hiện các dạng lệnh INSERT, UPDATE, DELETE có index thì cần một thời gian lâu hơn để xử lý.

Có hai dạng INDEXES sau:

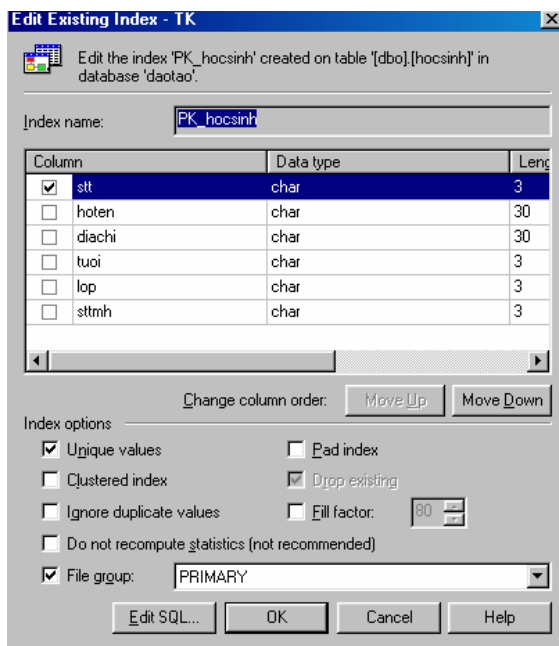
Clustered-Index: Là dạng index theo hàng các bảng ghi dữ liệu

NonClustered-Index: Là dạng index theo cột các bảng ghi dữ liệu



### -Edit Indexes:

Nếu bạn muốn thay đổi các cột giá trị cần indexes, bạn chọn cột dữ liệu - > click vào nút EDIT (index). Từ đây bạn có thể cập nhật (modify) cột cần sắp xếp.



Tại đây bạn có các lựa chọn về indexes:

- + Toán tử unique : Khử trùng
- + Clustered index : kiểu sắp xếp

- + Ignore duplicate value: Bỏ qua giá trị nhân bản
- + Filegroup: Chọn nhóm file là bao nhiêu?
- + Ngoài ra bạn có khả năng kết hợp với lệnh SQL bằng cách nhấn EDIT SQL.



## **PHẦN V. QUẢN TRỊ VÀ KHAI THÁC CSDL**

### **5.1. Các công cụ quản trị CSDL .**

#### **Giới thiệu về Microsoft SQL Server Roles:**

*Ý nghĩa:* Thiết lập các vai trò đặc biệt các quyền mặc nhiên mà không được gán cho các User Account. Nếu bạn có quyền đó thì phải thêm (Add) các User Account đó vào Roles .

#### **Roles có hai dạng:**

##### **1) Fixed Server Roles:**

- + SysAdmin
- + Server Admin
- + Setup Admin
- + Security Admin
- + ProcessAdmin
- + Dbcreator
- + DiskAdmin

##### **2) Fixed database roles**

- + Public
- + DB\_Owner
- + DB\_AccessAdmin
- + DB\_dataReader
- + DB\_dataWriter
- + DB\_ddlAdmin
- + DB\_SecurityAdmin
- + DB\_BackupAdmin
- + DB\_Denydatareader

## + DB\_DenydataWriter

Nếu bạn muốn huỷ bỏ ROLES thì bạn phải xoá toàn bộ các User trong database ROLES.

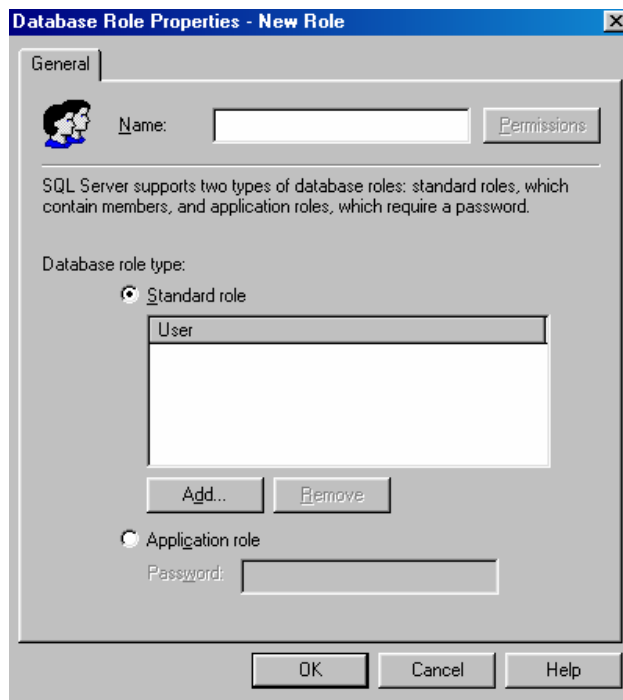
Tại sao phải dùng database Roles?

+ Là vì nó giúp bạn mở rộng database, tạo hiệu suất truy nhập.. .

+ Các user có cùng chung các đặc trưng nên người quản lý phải tạo ra các nhóm, hay các ROLES để quản lý dễ dàng hơn.

## Làm thế nào để tạo một User Roles?

1) Chọn ROLES → Nhấn nút phải chuột → chọn New Roles



Từ đây: nhập tên database roles

Add các user cần tạo ra → thực hiện xong nhấn OK

Nếu là các ứng dụng thì bạn chọn Application Roles

## -Tạo kiểu dữ liệu do người dùng định nghĩa - user defined data types.

Kiểu dữ liệu do người dùng định nghĩa nhằm mở rộng cho người sử dụng dễ dàng tạo ra riêng cho mình các loại dữ liệu tương ứng với nhu cầu đặt ra

## Làm thế nào để tạo user defined data types ?

### 2) Chọn user defined data types

3) Nhấn nút phải chuột

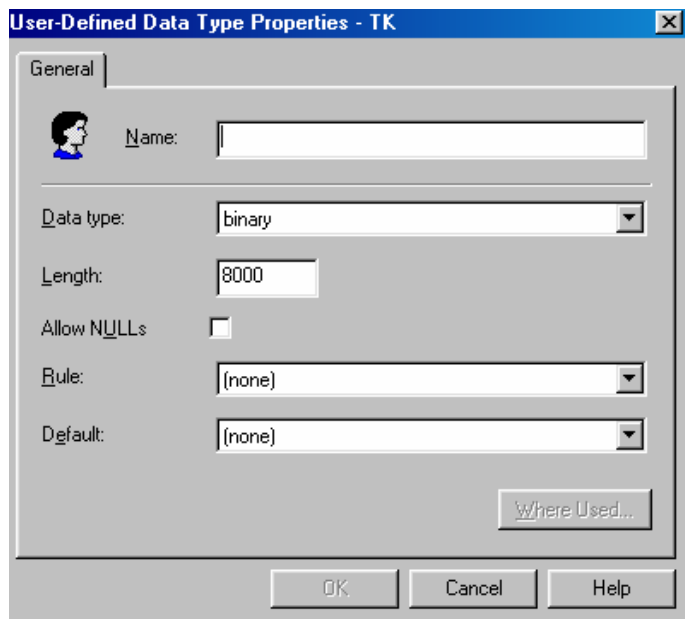
4) Chọn New user defined data types

Nhập tên loại dữ liệu cần tạo : (Name)

Kiểu dữ liệu: Data type

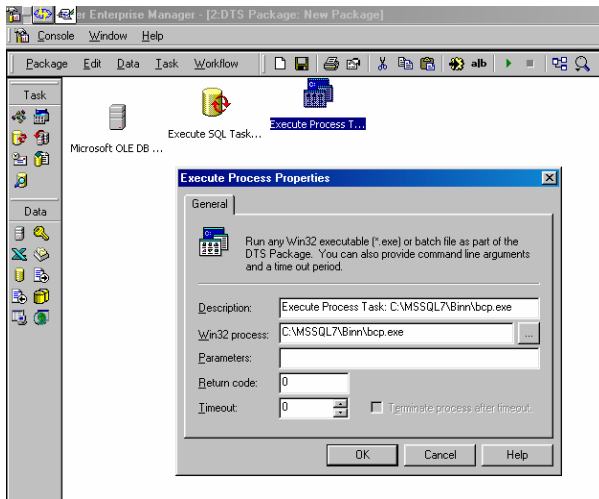
Chiều dài: Length

Giá trị NULL: Allow nulls



## DTS (data transformation services)

Là các dịch vụ của Microsoft SQL Server cung cấp cho người dùng chuyển đổi dữ liệu như trong phần Import and export data



Ngoài ra ta cũng có thể sử dụng một số lệnh của Transact SQL để thực hiện trên SQL Server Analyzer Ví dụ như lệnh :

BCP database\_name..Tablename

{IN|OUT} dataFile [/F firstrow][/L lastrow][/b batchsize][/C ][/U login\_ID]  
[/P Password] [/S Servername].

## 5.2. Các nhiệm vụ và quyền hạn trong việc quản trị và sở hữu CSDL (Management).

### 1) SQL server Agent- Các tác nhân quản lý server:

+ Cảnh báo: Alert

🔔 Demo: Full msdb log	Yes	9002	0	(Never ...	No	No
🔔 Demo: Full tempdb	Yes	9002	0	(Never ...	No	No
🔔 Demo: Sev. 19 Errors	Yes	0	19	(Never ...	No	No
🔔 Demo: Sev. 20 Errors	Yes	0	20	(Never ...	No	No
🔔 Demo: Sev. 21 Errors	Yes	0	21	(Never ...	No	No
🔔 Demo: Sev. 22 Errors	Yes	0	22	(Never ...	No	No
🔔 Demo: Sev. 23 Errors	Yes	0	23	(Never ...	No	No
🔔 Demo: Sev. 24 Errors	Yes	0	24	(Never ...	No	No
🔔 Demo: Sev. 25 Errors	Yes	0	25	(Never ...	No	No

+Operator:

Người điều hành công việc trong các ngày trong tuần

+Jobs:

Các công việc được định ra trong các ngày trong tuần. Ai làm việc ?

## 2) Current Activity 10/23/00

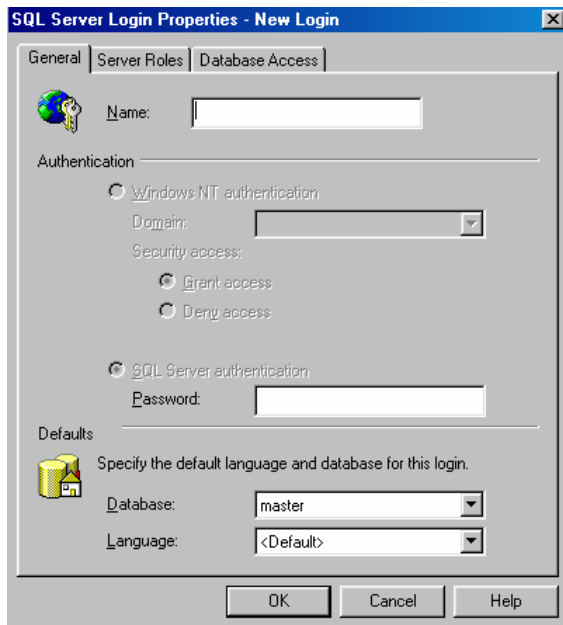
Nhật ký làm việc hàng ngày của SQL server. Nếu bạn là Admin thì bạn có thể nhìn thấy tất cả các thông tin mà SQL Server đã và đang làm việc.

## 3) SQL Server Log:

Ghi lại toàn bộ công việc mà SQL server đã thực hiện nhằm giúp người quản trị mạng hiểu biết thêm về thông tin được xử lý trên máy cũng như theo dõi các vấn đề có liên quan. Ví dụ như ai đã truy nhập vào database LUUTRU vào lúc 10:22:40 AM ngày 20 tháng 10 năm 2000 và đã xoá đi một số dữ liệu quan trọng.

## Security:

1) Login: Tạo login\_id để **login** vào mạng SQL server.



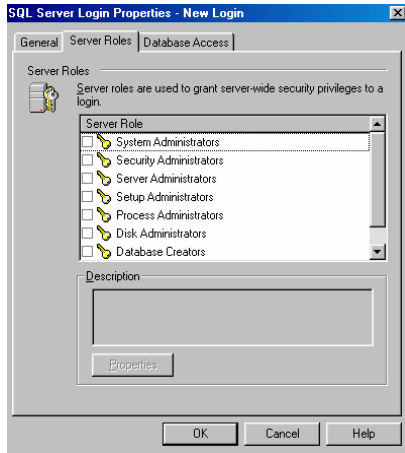
+ **Name:** Nhập login\_name

+ **Password:** nhập mật khẩu vào (hai lần nhập)

+ Chọn database cho user cần truy nhập.

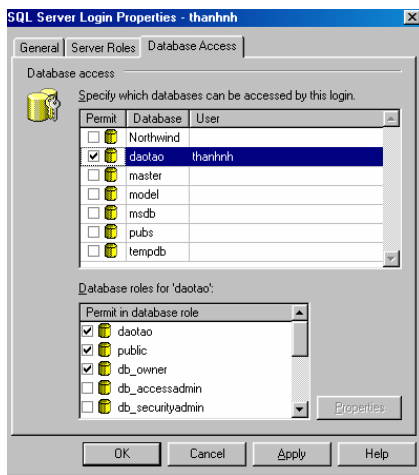
*(Ngôn ngữ ngầm định là English.)*

Tiếp theo bạn phải chọn các quyền truy xuất cho user đó khi truy xuất vào database.



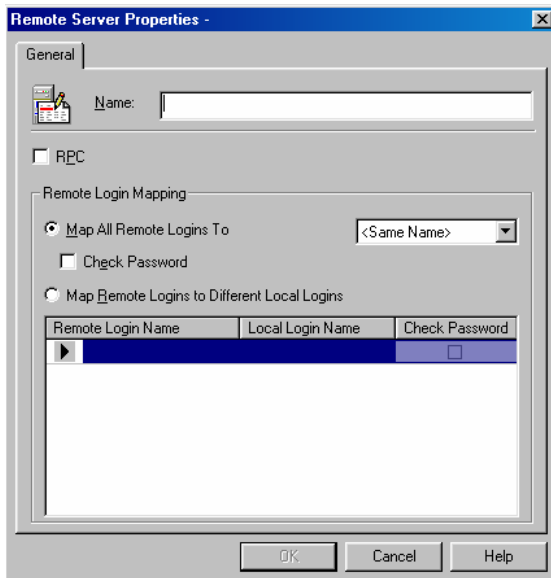
### Database access:

Chọn quyền truy xuất vào các database



+ Link Servser. Chỉ định các mối liên kết với các server khác.

Remote server: Tên máy Server truy nhập từ xa.



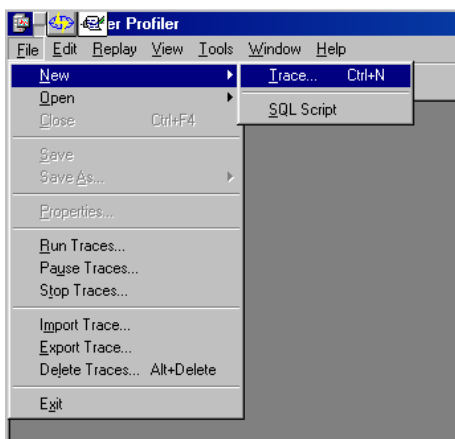
#### 4) Support Servic:

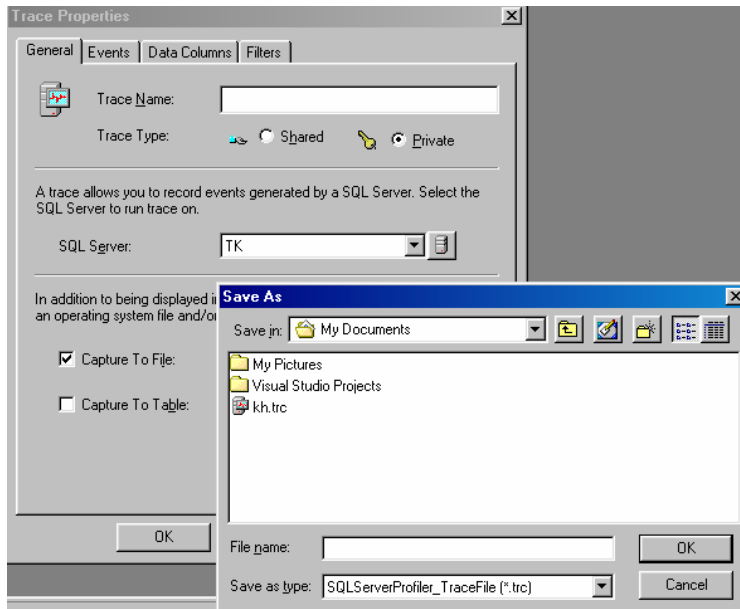
Cung cấp các dịch vụ, tạo hiệu suất truy nhập cao, đa dạng.

- Distributed Transaction Coordinator
- SQL Mail

## SQL Server Profile

Tạo các Trace nhằm theo dõi chính xác lúc có các tác nhân tác động vào CSDL.





### Làm thế nào để tạo ra TRACE?

- 1) Vào menu File Chọn New TRACE
- 2) Nhập tên TRACE cần tạo
- 3) Nhập tên Server
- 4) Capture to file: Tên file lưu trữ lại các thông tin mà trace nắm bắt được
- 5) Capture to table: Tên table lưu trữ lại các thông tin mà trace nắm bắt được .
- 6) Chọn kiểu của Trace : (Share type, Private type) trong phạm vi có chia sẻ hay chỉ tại máy đơn

### 5.3. Tính toàn vẹn và tính an toàn CSDL

#### - Toàn vẹn mức bảng, mức trường, mức quan hệ, mức ràng buộc logic

An toàn dữ liệu hay tính toàn vẹn dữ liệu là điểm mấu chốt trong công cuộc bảo vệ thông tin dữ liệu.

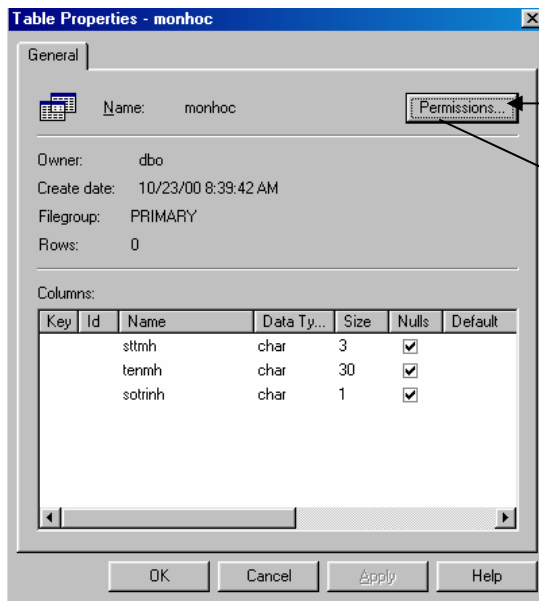
+ Ở dạng bảng: Các bảng thường được bảo mật bởi các quyền được Administrator gán cho kết hợp với việc được sử dụng những câu lệnh nào cho từng loại bảng trong CSDL.



+ Field: Trường dữ liệu là các cột dữ liệu hay các thuộc tính của bảng, tập hợp các cột của bảng tạo thành 1 bản ghi hay còn gọi là hàng dữ liệu được bảo vệ bởi quyền được gán cho bảng và cũng như các mối liên kết trong bảng ở mức quan hệ, kết nối giữa các khoá với nhau của các bảng khác nhau (khoá trong và khoá ngoài).

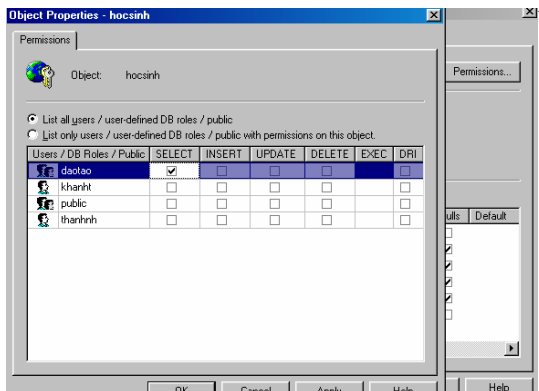
- **An toàn về sự kết nối truy nhập đến CSDL (authentication) và quyền thao tác (permission) trên các đối tượng của CSDL**

- Nếu muốn gán quyền bạn chọn **Permission**.



Click vào đây

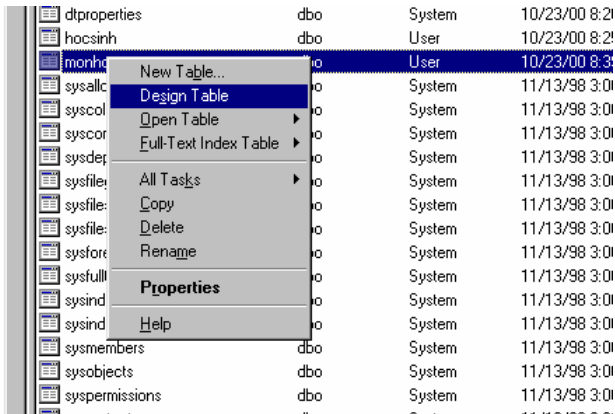
Trong nhóm 1: user/DB roles/Public  
Gồm các login\_ID và các nhóm làm việc .  
Nhóm 2: Tập các lệnh trong Transact SQL  
Nếu bạn check vào ô nào của user/nhóm nào thì user/nhóm đó được quyền sử dụng lệnh đó.  
Sau khi đã check xong “quyền”  
Tiếp tục nhấn nút OK (có thể Apply)



Nếu bạn muốn thay đổi bất cứ một điều gì trong table nào đó thì bạn hãy làm như sau:

- 1) Chọn table cần modify
- 2) Click nút phải chuột
- 3) Chọn Design database

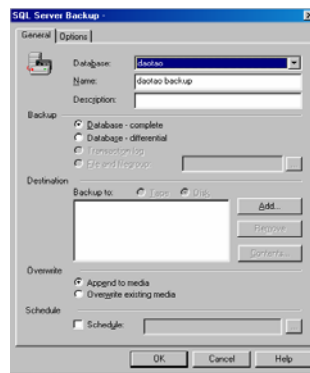
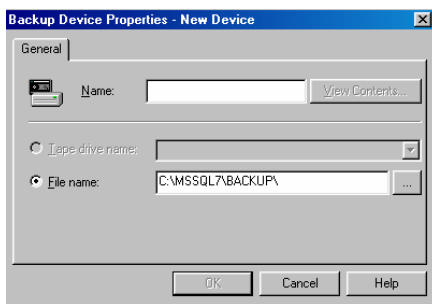
Ngoài ra bạn còn có rất nhiều chức năng ở đây khi nhấn nút phải chuột (Open table, Full text index table, All task , copy, delete , rename, property.. .)



### - Sao lưu CSDL - Backup dữ liệu:

Nếu bạn muốn tạo bản dữ liệu nhân bản thì sử dụng chức năng backup:

- Backup gồm hai bước: Tạo thiết bị backup và chọn CSDL cần backup.



#### **5.4. Các hình thức khai thác CSDL:**

##### **- Câu lệnh Microsoft SQL Server.**

Lệnh Microsoft SQL Server (Trong Transact SQL ) là tập hợp các câu hỏi (truy vấn) “**thông minh**” nhằm đáp ứng nhanh, chính xác và đáp ứng cho nhiều người dùng tin. Ngôn ngữ SQL người ta thường gọi là ngôn ngữ “Thông minh”.

**i) Các câu lệnh chính thường hay dùng trong Microsoft SQL Server cũng như trong DMO (các ứng dụng khai thác CSDL của SQL ).**

##### **1. Câu lệnh SELECT: Lấy dữ liệu từ bảng (table) trong cơ sở dữ liệu.**

###### **Cú pháp:**

```
SELECT [ALL | DISTINCT] select_list
    [INTO [new_table_name]]
[FROM {table_name | view_name}[(optimizer_hints)]
    [[, {table_name2 | view_name2}[(optimizer_hints)]
    [..., {table_name16 | view_name16}[(optimizer_hints)]]]
[WHERE clause]
[GROUP BY clause]
[HAVING clause]
[ORDER BY clause]
[COMPUTE clause]
[FOR BROWSE]
```

Trong đó:

**ALL:** Lấy tất cả các giá trị theo khoá. ALL là tham số ngầm định của truy vấn lựa chọn.

**DISTINCT:** Chỉ lấy các giá trị duy nhất. Giá trị NULL cũng được lấy vào kết quả nhưng chỉ có một giá trị NULL được lựa chọn mà thôi.

select\_list: Xác định các cột được lựa chọn lấy dữ liệu. Select\_list có thể là :

- Dấu sao (\*): Tất cả các cột theo thứ tự được tạo bằng câu lệnh CREATE TABLE của các bảng trong mệnh đề FROM của truy vấn lựa chọn.

- Danh sách tên của các cột theo thứ tự mong muốn, dùng dấu phẩy (,) để tách tên của các cột nếu select\_list có nhiều hơn một cột.

- Tên cột ngầm định là tiêu đề cột. Có thể thay đổi tiêu đề cột theo cú pháp:

Tiêu\_đề\_cột = Tên\_cột

hoặc

Tên\_cột Tiêu\_đề\_cột

Nếu có dấu cách (space) trong tên của tiêu đề cột thì phải dùng dấu ngoặc kép (' ' hoặc " ") đối với tiêu đề cột. Ví dụ:

```
SELECT 'Tên tác giả' = au_lname  
FROM authors
```

- Một biểu thức (tên cột, hằng số, hàm hoặc là sự kết hợp của tên các cột, các hằng số, các hàm bằng các toán tử hoặc các truy vấn con).

- Có thể dùng từ khoá IDENTITYCOL thay cho tên của cột có thuộc tính IDENTITY (giá trị của cột là NOT NULL).

- Biến cục bộ hoặc biến toàn cục.

- Gán biến cục bộ theo cú pháp:

@variable = Biểu thức

Chú ý: Nếu select\_list gồm việc gán biến thì không thể thực hiện lấy dữ liệu.

INTO Tên\_bảng\_mới: Tạo ra một bảng mới với các cột được chỉ ra trong select\_list và các giá trị thoả mãn mệnh đề WHERE. Để chọn dữ liệu

và đưa vào một bảng cố định, lựa chọn select into/bulkcopy phải được hoạt động (bằng cách thực hiện thủ tục lưu trữ của hệ thống sp\_dboption). Khi một cơ sở dữ liệu được tạo mới, lựa chọn select into/bulkcopy ngầm định là không hoạt động. Tên bảng mới phải tuân theo các qui tắc giống tên bảng gốc, gồm:

- Nếu select into/bulkcopy đang hoạt động trong cơ sở dữ liệu mà bảng sẽ được tạo ra, bảng cố định phải được tạo ra trước đó. Tên bảng phải duy nhất trong cơ sở dữ liệu và phải tuân theo các qui tắc nhận dạng.

- Nếu select into/bulkcopy không hoạt động trong cơ sở dữ liệu mà bảng sẽ được tạo ra thì không thể tạo ra được bảng cố định bằng lệnh SELECT INTO; chỉ có thể tạo ra các bảng tạm thời cục bộ hoặc toàn cục. Để tạo bảng tạm thời, tên của bảng tạm thời phải được bắt đầu bằng dấu # (#Tên\_bảng) đối với bảng tạm thời cục bộ và hai dấu ## (##Tên\_bảng) đối với bảng tạm thời toàn cục khi tạo bảng bằng CREATE TABLE.

- Hoạt động của SELECT INTO gồm hai bước. Bước đầu tiên là tạo bảng, người sử dụng phải có quyền tạo bảng trong cơ sở dữ liệu đích. Bước thứ hai là thêm các dòng giá trị vào bảng mới. Nếu bước thứ hai không thực hiện được vì bất kỳ lý do gì (lỗi phân cứng, vượt quá dung lượng đĩa của cơ sở dữ liệu, ...) thì bảng mới vẫn tồn tại nhưng không có giá trị.

- Có thể dùng SELECT INTO để tạo ra một bảng có cấu trúc giống nhau (khác tên bảng) bằng cách đưa điều kiện sai trong mệnh đề WHERE.

- Không thể dùng SELECT INTO cùng với mệnh đề COMPUTE hoặc trong một giao dịch do người dùng định nghĩa.

- Nếu chọn cột có giá trị NOT NULL vào bảng mới thì cột mới sẽ có thuộc tính IDENTITY (có giá trị NOT NULL) trừ khi một trong số các điều kiện sau được thoả mãn:

- . Câu lệnh SELECT bao gồm mệnh đề liên kết (join), mệnh đề GROUP BY hoặc các hàm tổng hợp.

. Nhiều câu lệnh SELECT được kết hợp với nhau bằng toán tử UNION.

. Cột có giá trị NOT NULL được liệt kê nhiều hơn trong select\_list.

. Cột có giá trị NOT NULL là một phần của biểu thức.

FROM: Chỉ ra các bảng được sử dụng trong lệnh SELECT. Nếu select\_list chỉ là hằng số, biến và các biểu thức số mà không có tên cột thì không cần mệnh đề FROM. Số lượng bảng tối đa trong mệnh đề FROM là 16 bảng, kể cả các truy vấn con.

- Tên\_bảng = [[Tên\_cơ\_sở\_dữ\_liệu.]Chủ\_cơ\_sở\_dữ\_liệu.]{Tên\_bảng}

Nếu có nhiều bảng trong mệnh đề FROM thì dùng dấu phẩy (,) để phân tách các bảng. Nếu các bảng tồn tại trong cơ sở dữ liệu khác thì phải đưa ra tên bảng đầy đủ (Tên\_cơ\_sở\_dữ\_liệu.Chủ\_cơ\_sở\_dữ\_liệu.Tên\_bảng).

Mỗi tên bảng có thể có bí danh để sử dụng cho thuận lợi hoặc để phân biệt các vai trò khác nhau của các bảng khi liên kết hoặc thực hiện các truy vấn con. Để dùng bí danh, trước hết phải chỉ ra tên bảng, sau đó là một khoảng trắng (space) và cuối cùng là tên bí danh, ví dụ:

```
SELECT au_lname, au_fname, title
      FROM titles t, authors a, titleauthor ta
      WHERE ta.title_id = t.title_id
      AND ta.au_id = a.au_id
      ORDER BY title, au_lname, au_fname
```

Thứ tự của các bảng sau từ khoá FROM không ảnh hưởng đến kết quả của câu lệnh.

WHERE: Xác định các điều kiện để lựa chọn dữ liệu trong câu lệnh. Số lượng các điều kiện là không hạn chế (trong câu lệnh của ngôn ngữ hỏi đáp có cấu trúc SQL).

```
WHERE search_conditions
```

GROUP BY: Xác định các nhóm trong bảng, nếu trong select\_list có các hàm tổng hợp thì xác định giá trị tổng quát cho mỗi nhóm. Các dữ liệu kiểu văn bản (text) và ảnh (image) không được dùng trong mệnh đề GROUP BY.

```
GROUP BY [ALL] aggregate_free_expression  
        [, aggregate_free_expression]...
```

HAVING: Xác định các kiểu hạn chế khác nhau đối với các hàm tổng hợp trong select\_list.

```
HAVING search_conditions
```

search\_conditions hạn chế các dòng giá trị kết quả của truy vấn mà không ảnh hưởng đến việc tính toán của các hàm tổng hợp. Khi dùng mệnh đề WHERE, search\_conditions hạn chế các dòng giá trị được tính toán trong các hàm tổng hợp nhưng không hạn chế các dòng giá trị kết quả của truy vấn. Các dữ liệu kiểu văn bản (text) và ảnh (image) không được dùng trong mệnh đề HAVING. Số lượng điều kiện không hạn chế trong search\_conditions. Mệnh đề HAVING có thể được dùng độc lập với mệnh đề GROUP BY. Nếu mệnh đề HAVING được dùng với mệnh đề GROUP BY ALL thì HAVING sẽ phủ nhận ý nghĩa của từ khoá ALL.

ORDER BY: Sắp xếp kết quả theo cột, có thể sắp xếp tối đa là 16 cột.

```
ORDER BY {{table_name. | view_name.}column_name  
        | select_list_number | expression} [ASC | DESC]  
        [...{{table_name16. | view_name16.}column_name  
        | select_list_number | expression} [ASC | DESC]]
```

Các giá trị Null được xếp trước các giá trị khác, các cột có giá trị văn bản (text) hoặc ảnh (image) không được dùng trong mệnh đề ORDER BY.

COMPUTE: Được dùng với các hàm tổng hợp theo dòng (SUM, AVG, MIN, MAX, và COUNT).

```
COMPUTE row_aggregate(column_name)
      [, row_aggregate(column_name)...]
      [BY column_name [, column_name]...]
```

FOR BROWSE: Cho phép cập nhật bằng cách sử dụng thư viện cơ sở dữ liệu (DB\_Library) khi xem dữ liệu trong các chương trình ứng dụng của máy khách.

**2. Câu lệnh INSERT:** Thêm một dòng giá trị mới vào bảng dữ liệu đang tồn tại trong CSDL.

**Cú pháp:**

```
INSERT [INTO]
      {table_name | view_name} [(column_list)]
      {DEFAULT VALUES | values_list | select_statement}
      where
```

Trong đó:

INTO: Là từ khoá lựa chọn.

table\_name | view\_name: Tên của bảng được dùng trong câu lệnh INSERT. Nếu bảng không tồn tại trong cơ sở dữ liệu hiện thời thì phải chỉ ra đầy đủ tên bảng (database\_name.owner.object\_name).

column\_list: Danh sách các cột được thêm dữ liệu. Có thể liệt kê các cột theo bất kỳ thứ tự nào nhưng dữ liệu được thêm vào bảng phải có cùng thứ tự với thứ tự của cột. Nếu column\_list không được chỉ ra thì tất cả các cột (theo thứ tự được tạo ra bằng lệnh CREATE TABLE) của bảng sẽ được thêm dữ liệu.

DEFAULT VALUES: Thêm giá trị ngầm định cho tất cả các cột. Đối với các cột có giá trị NOT NULL, giá trị phù hợp tiếp theo sẽ được thêm vào. Đối với các cột được nhận giá trị NULL thì giá trị ngầm định được thêm vào là NULL.



values\_list: Để liệt kê các giá trị cho mỗi cột theo thứ tự trong column\_list hoặc trong bảng.

```
VALUES (DEFAULT | constant_expression
        [, DEFAULT | constant_expression]...)
```

select\_statement: Là câu lệnh SELECT dùng để nhận dữ liệu mà sẽ được thêm vào bảng từ một bảng đang tồn tại.

Ví dụ:

A. Thêm giá trị cho tất cả các cột

```
INSERT titles
VALUES('BU2222', 'Faster!', 'business', '1389',
      NULL, NULL, NULL, NULL, 'ok', '06/17/87')
```

B. Thêm giá trị cho các cột được liệt kê trong column\_list:

```
INSERT titles(title_id, title, type, pub_id, notes, pubdate)
VALUES ('BU1237', 'Get Going!', 'business', '1389',
      'great', '06/18/86')
```

C. Thêm giá trị cho tất cả các cột từ một bảng khác:

```
INSERT INTO newauthors
SELECT *
FROM authors
WHERE city = 'San Francisco'
```

D. Thêm các giá trị ngầm định:

```
INSERT publishers DEFAULT VALUES
```

Vì giá trị của cột pub\_id trong bảng publishers là NOT NULL nên xuất hiện thông báo lỗi:

```
Msg 233, Level 16, State 2
```

```
The column pub_id in table publishers may not be null.
```

E. Thêm giá trị cho các cột sử dụng từ khoá DEFAULT

```
INSERT employee
```

```
VALUES ('KLT91469F', 'Katrina', 'L', 'Thompson',  
DEFAULT,  
DEFAULT,DEFAULT, '01/14/95')
```

**3. Câu lệnh UPDATE:** Thay đổi dữ liệu dòng trong bảng đang tồn tại bằng cách thêm dữ liệu mới hoặc sửa đổi dữ liệu đang có.

**Cú pháp:**

```
UPDATE {table_name | view_name}  
SET [{table_name | view_name}]  
    {column_list  
    | variable_list  
    | variable_and_column_list}  
    [, {column_list2  
        | variable_list2  
        | variable_and_column_list2}  
    ...    [, {column_listN  
            | variable_listN  
            | variable_and_column_listN}]]  
[WHERE clause]
```

**Trong đó:**

table\_name | view\_name: Tên bảng được thay đổi dữ liệu. Nếu bảng không thuộc cơ sở dữ liệu hiện tại thì phải chỉ ra đường dẫn đầy đủ của bảng đó (Tên\_cơ\_sở\_dữ\_liệu.Tên\_chủ\_cơ\_sở\_dữ\_liệu.tên\_bảng).

SET: Là từ khoá dùng để liệt kê danh sách các cột hoặc biến sẽ được thay đổi. Nếu có hơn một cột hoặc biến được liệt kê thì dùng dấu phẩy để tách chúng.

column\_list: bao gồm các biểu thức gán

Tên\_cột = { Biểu thức | DEFAULT | NULL }

variable\_list: bao gồm các biểu thức gán

Tên\_biến = { Biểu thức | NULL }

Trong đó:

Tên\_cột: Chỉ ra cột trong bảng.

Biểu thức: Là tên cột, hằng số, hàm (ngoại trừ hàm tổng hợp), hoặc là sự kết hợp của các tên cột, hằng số, và các hàm bằng các toán tử hoặc các truy vấn con.

DEFAULT: Thêm các giá trị ngầm định cho các cột đó.

variable\_and\_column\_list: Bao gồm các biểu thức gán:

Tên\_biến = Tên\_cột = { Biểu thức | NULL }

WHERE: Xác định điều kiện để thay đổi dữ liệu trong bảng. Nếu không có mệnh đề WHERE thì câu lệnh UPDATE sẽ thay đổi tất cả dữ liệu trong bảng.

WHERE {Điều\_kiện\_tìm\_kiểm}

Điều\_kiện\_tìm\_kiểm: Xác định các chỉ tiêu được thay đổi dữ liệu. Điều kiện tìm kiếm có thể là một biểu thức, một truy vấn con, một hằng số, ...

Ví dụ:

A. Câu lệnh UPDATE chỉ sử dụng mệnh đề SET:

```
UPDATE publishers
SET city = 'Atlanta', state = 'GA'
```

```
UPDATE publishers
SET pub_name = NULL
```

```
UPDATE titles
SET price = price * 2
```

B. Câu lệnh UPDATE có sử dụng mệnh đề WHERE:

```
UPDATE authors
      SET state = 'PC', city = 'Bay City'
      WHERE state = 'CA' AND city = 'Oakland'
```

C. Câu lệnh UPDATE sử dụng lệnh SELECT:

```
UPDATE titles
      SET ytd_sales = ytd_sales + qty
      FROM titles, sales
      WHERE titles.title_id = sales.title_id
            AND sales.date = (SELECT MAX(sales.date) FROM
sales)
```

**4. Câu lệnh DELETE:** Xoá các dòng giá trị trong bảng dữ liệu đang tồn tại.

**Cú pháp:**

```
DELETE [FROM] {table_name | view_name}
      [WHERE clause]
```

Trong đó:

table\_name | view\_name: Tên bảng được xoá dữ liệu. Nếu bảng không thuộc cơ sở dữ liệu hiện tại thì phải chỉ ra đường dẫn đầy đủ của bảng đó: (Tên\_cơ\_sở\_dữ\_liệu.Tên\_chủ\_cơ\_sở\_dữ\_liệu.tên\_bảng).

WHERE: Xác định điều kiện để xoá dữ liệu trong bảng. Nếu câu lệnh DELETE không có mệnh đề WHERE thì tất cả các dòng trong bảng sẽ bị xoá.

```
WHERE {search_conditions | CURRENT OF cursor_name}
search_conditions: Điều kiện để xoá dữ liệu.
```

Ví dụ:

A. Câu lệnh DELETE không có tham số:

Xoá tất cả các dòng dữ liệu trong bảng authors.

DELETE authors

B. Dùng lệnh DELETE để xoá một số dòng dữ liệu:

DELETE FROM authors

WHERE au\_lname = 'McBadden'

## 5. Các hàm tổng hợp:

Các hàm tổng hợp tính toán các giá trị như giá trị trung bình, tổng số theo giá trị của các cột được chỉ ra và trả về một giá trị.

### Cú pháp:

Tên\_hàm\_tổng\_hợp ([ALL | DISTINCT] Biểu thức)

Trong đó:

Tên\_hàm\_tổng\_hợp gồm:

AVG: Trả về giá trị trung bình của tất cả các giá trị hoặc chỉ đối với các giá trị không lặp trong biểu thức. AVG chỉ dùng cho các cột có giá trị số.

COUNT: Trả về số lượng các giá trị NOT NULL của biểu thức. Nếu dùng từ khoá DISTINCT thì hàm COUNT chỉ đếm số giá trị NOT NULL duy nhất. Hàm COUNT được dùng với cả cột có giá trị số và ký tự.

COUNT(\*): Trả về số dòng trong bảng. COUNT(\*) không có tham số và không được dùng với từ khoá DISTINCT. COUNT(\*) đếm tất cả các dòng, cả các dòng có giá trị NULL.

MAX: Trả về giá trị lớn nhất trong biểu thức. Hàm MAX được dùng với các cột có giá trị số, ký tự, và ngày giờ nhưng không dùng với cột có giá trị bit.

MIN: Trả về giá trị nhỏ nhất trong biểu thức. Hàm MIN được dùng với các cột có giá trị số, ký tự, và ngày giờ nhưng không dùng với cột có giá trị bit.

SUM: Trả về giá trị tổng của tất cả các giá trị hoặc chỉ đối với các giá trị không lặp trong biểu thức. SUM chỉ dùng cho các cột có giá trị số.

ALL: Hàm tổng hợp có tác dụng đối với tất cả các giá trị và ALL là giá trị ngầm định.

DISTINCT: Loại bỏ các giá trị đúp trước khi thực hiện hàm tổng hợp.

Biểu thức: Là tên cột, hằng số, hàm, hoặc là sự kết hợp của các tên cột, hằng số, và các hàm bằng các toán tử số học hoặc các toán tử bit.

Ví dụ:

A. Hàm SUM và hàm AVG

```
SELECT AVG(advance), SUM(ytd_sales)
FROM titles
WHERE type = 'business'
```

B. Hàm SUM và AVG dùng với mệnh đề GROUP BY

```
SELECT type, AVG(advance), SUM(ytd_sales)
FROM titles
GROUP BY type
```

C. Hàm COUNT sử dụng DISTINCT:

```
SELECT COUNT(DISTINCT city)
FROM authors
```

D. Hàm COUNT(\*) trong GROUP BY HAVING:

```
SELECT type
FROM titles
GROUP BY type
HAVING COUNT(*) > 1
```

E. Hàm SUM và AVG trong HAVING:

```
SELECT pub_id, SUM(advance), AVG(price)
FROM titles
```

```
GROUP BY pub_id  
HAVING SUM(advance) > $25000  
AND AVG(price) > $15
```

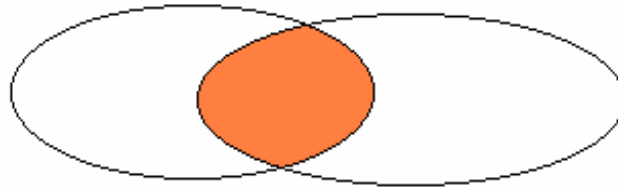
## 6. Các kết nối bảng:

Các mệnh đề kết nối bảng chuẩn của ANSI gồm;

- INNER JOIN
- LEFT JOIN
- LEFT [OUTER] JOIN
- RIGHT JOIN
- RIGHT [OUTER] JOIN
- FULL JOIN
- FULL [OUTER] JOIN

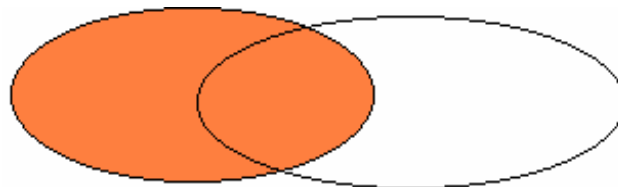
### Trong đó:

INNER JOIN: Xác định giá trị trả về gồm những dòng giá trị cùng



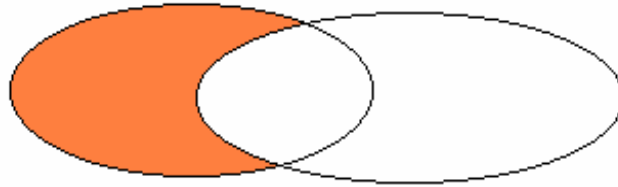
thuộc cả hai bảng.

LEFT JOIN: Xác định dòng giá trị trả về thuộc bảng bên trái của liên

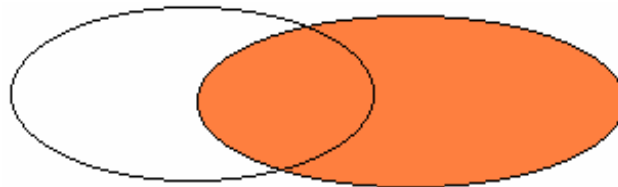


kết.

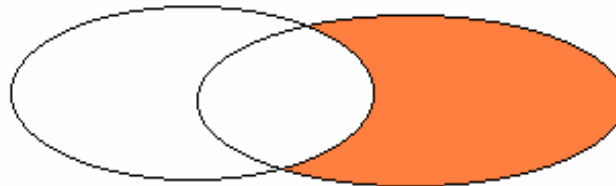
LEFT OUTER JOIN: Xác định dòng giá trị trả về thuộc bảng bên trái của liên kết mà không có trong bảng bên phải của liên kết.



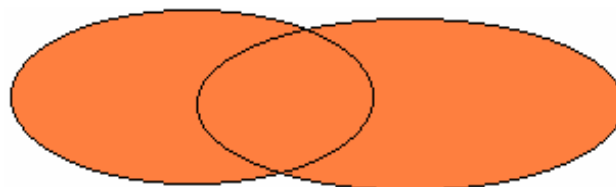
RIGHT JOIN: Xác định dòng giá trị trả về thuộc bảng bên phải của liên kết.



RIGHT OUTER JOIN: Xác định dòng giá trị trả về thuộc bảng bên phải của liên kết mà không có trong bảng bên trái của liên kết.

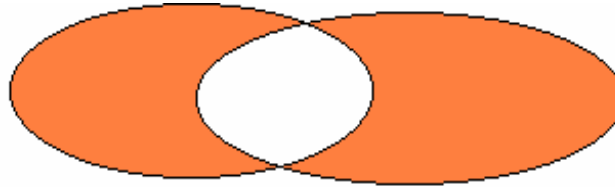


FULL JOIN: Xác định dòng giá trị trả về thuộc cả hai bảng của liên kết.





FULL OUTER JOIN: Xác định dòng giá trị trả về không thuộc cả hai bảng của liên kết.



ii) Sau đây là một số lệnh và hàm thường hay dùng trong Transact SQL .

**Chú ý :**

Các câu lệnh trong Microsoft SQL Server thường hay có các toán tử điều kiện, biểu thức logic, từ khoá, toán tử chỉ định và các tiêu thức lệnh sắp nhóm , sắp xếp indexes hay một số toán tử khác... Thường đi theo sau các từ khoá lệnh mà các từ khoá này đã diễn giải rõ ở trên các câu lệnh trên(Select, insert, update .. ..)

**1) Tạo table (Creates a new table.):** có hai loại table

- Các dạng table tạm thời
- Các dạng table cố định

**Cú pháp:**

CREATE TABLE

[ database\_name.[owner]. | owner.] table\_name

{ <column\_definition>

| column\_name AS computed\_column\_expression

| <table\_constraint> } [...n] )

[ON {filegroup | DEFAULT} ]

[TEXTIMAGE\_ON {filegroup | DEFAULT} ]

**Diễn giải:**

```
<column_definition> ::= { column_name data_type }  
[ [ DEFAULT constant_expression ]  
  | [ IDENTITY [(seed, increment) [NOT FOR REPLICATION]] ] ]  
]  
[ ROWGUIDCOL ]  
[ <column_constraint> ] [ ...n]
```

```
<column_constraint> ::= [CONSTRAINT constraint_name]  
{  
  [ NULL | NOT NULL ]  
  | [ { PRIMARY KEY | UNIQUE }  
    [ CLUSTERED | NONCLUSTERED ]  
    [ WITH FILLFACTOR = fillfactor ]  
    [ ON { filegroup | DEFAULT } ] ]  
  ]  
  | [ [ FOREIGN KEY ]  
    REFERENCES ref_table [(ref_column) ]  
    [ NOT FOR REPLICATION ]  
  ]  
  | CHECK [NOT FOR REPLICATION]  
    (logical_expression)  
}
```

```
<table_constraint> ::= [CONSTRAINT constraint_name]  
{  
  [ { PRIMARY KEY | UNIQUE }  
    [ CLUSTERED | NONCLUSTERED ]  
    { ( column[,...n] ) }  
}
```

```
[ WITH FILLFACTOR = fillfactor]
[ ON { filegroup | DEFAULT} ] ]
| FOREIGN KEY
    [(column[,...n])]
    REFERENCES ref_table [(ref_column[,...n])]
    [NOT FOR REPLICATION]
| CHECK [NOT FOR REPLICATION]
    (search_conditions)}
```

### **Các tham số :**

- a) **database\_name:** Tên database
- b) **owner :** quyền Owner
- d) **table\_name:** Tên Table
- e) **column\_name:** Số cột nằm trong Table

**Và sau đây là một số thông số quan trọng:**

+**computed\_column\_expression:** là các tên cột nằm trong các thành phần sau : PRIMARY KEY, UNIQUE, FOREIGN KEY, DEFAULT được định nghĩa.

+ ON {filegroup | DEFAULT} : Nhóm file –trong phần database mà bạn đã biết khi tạo CSDL.

+ TEXTIMAGE\_ON

+ **data\_type :** Kiểu dữ liệu

+ **NULL | NOT NULL :** Có giá trị NULL hay không có

+ **PRIMARY KEY :** Khoá trong

+ **UNIQUE :** Hạn chế trùng lặp

+ **CLUSTERED | NONCLUSTERED :** Sắp xếp

+ [WITH FILLFACTOR = fillfactor]

+ **FOREIGN KEY...REFERENCES** : Khoá ngoài

- **ref\_table** : Tên của bảng TABLE khác khi câu lệnh này cần liên kết với các khoá ngoài

- (**ref\_column**[,...n]) : từng cột của Table được chỉ định

+ **CHECK** : có dùng từ khoá CHECK (hạn chế trùng lặp) hay không ?

+ **logical\_expression**: Biểu thức logic kèm theo

Ngoài ra còn có các dạng thiết lập khác:

+ **Create View View\_name AS Select Statement**

+ **Create Procedure**

+ **Create default**

+ **Create Index**

+ **Create Rule**

+ **Create Trigger.**

+ **Create Database** :Thiết lập CSDL

**Dạng lệnh:**

**Tương tự như lệnh thiết lập bảng tuy nhiên có một số thông số khác.**

Cú pháp:

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

```
[, <filegroup> [,...n] ]]
```

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

```
[ FOR LOAD | FOR ATTACH ]
```

**Diễn giải:**

+**database\_name**: Tên database

+ **ON [PRIMARY] [ <filespec> [,...n]** : Chỉ định khoá trong

+ [ LOG ON { <filespec> [,...n] } ] : Chỉ định LOGON\_name

<filespec> ::=

( [ NAME = logical\_file\_name, ]

FILENAME = 'os\_file\_name'

[, SIZE = size]

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

[, FILEGROWTH = growth\_increment] ) [,...n]

<filegroup> ::=

FILEGROUP filegroup\_name <filespec> [,...n]

### **Các nhóm lệnh xoá :**

**Tương tự như trên nếu có tồn tại lệnh thiết lập thì sẽ tồn tại lệnh xoá .**

**+ DROP View View\_name AS Select Statemant**

**+ DROP Procedure**

**+ DROP default**

**+ DROP Index**

**+ DROP Rule**

**+ DROP Trigger.**

### **2) Dạng lệnh:**

Thêm hoặc xoá một số cột của bảng(table) hay hạn chế một số thuộc tính của CHECK, TRIGGER..

### **Cú pháp:**

ALTER TABLE table

{ [ALTER COLUMN column\_name

{ new\_data\_type [ (precision[, scale] ) ]

[ NULL | NOT NULL ]

```
        | {ADD | DROP} ROWGUIDCOL
    }
]
| ADD
    {    [ <column_definition> ]
        | column_name AS computed_column_expression
    }[,...n]
| [WITH CHECK | WITH NOCHECK] ADD
    { <table_constraint> }[,...n]
| DROP
    {    [CONSTRAINT] constraint_name
        | COLUMN column
    }[,...n]
| {CHECK | NOCHECK} CONSTRAINT
    { ALL | constraint_name[,...n] }
| {ENABLE | DISABLE} TRIGGER
    { ALL | trigger_name[,...n] }
}
```

Lệnh này cũng có các thống số như các câu lệnh Create table

3) Đổi tên Table hay đổi tên CSDL:

**Cú pháp:**

SP\_Rename “Tên cũ”,”Tên mới”

4) Xem thông tin về Database:

**Cú pháp:**

sp\_helpdb [[@dbname=] 'name']

+ [@dbname=] 'name': Cung cấp tên database

- 5) Cho biết thông số : số hàng trong table, dung lượng đang dùng trong ổ đĩa của database.

**Cú pháp:**

```
sp_spaceused [[@objname =] 'objname']  
              [,[@updateusage =] 'updateusage']  
+ [@objname =] 'objname': Tên database  
+ [@updateusage =] 'updateusage':
```

- 6) Xem thông tin về các đối tượng trong database

**Cú pháp:**

```
sp_help [[@objname =] name]
```

- 7) Cung cấp thông tin về user đang truy nhập vào Microsoft SQL Server

**Cú pháp:**

```
sp_who [[@login_name =] 'login']
```

- 8) **Executive:** Lời gọi một thủ tục trong Stored Procedure:

```
[[EXEC[UTE]]    [@return_status =]{procedure_name [;number] |  
@procedure_name_var}[[@parameter =] {value | @variable [OUTPUT] |  
[DEFAULT]] [,...n][WITH RECOMPILE]
```

**Thực hiện trong các xâu ký tự:**

```
EXEC[UTE] ({@string_variable | [N]'tsql_string'} [+...n])
```

- 9) Chuyển đổi số liệu: tạo bảng sao dữ liệu hay chuyển đổi dữ liệu vào thiết bị khác thì dùng lệnh DUMP

**Cú pháp:**

```
DUMP Database Database_name to Disk = “path\filename”
```

```
DUMP table table_name to Disk = “path\filename”
```

```
DUMP Transaction Database_name to Disk = “path\filename”
```

Nếu muốn xoá dữ liệu khi bị tràn ô nhớ:

DUMP Transaction Database\_name with [Truncate\_opnly | No\_log ]

Khi muốn LOAD vào hay phục hồi dữ liệu được BACKUP ra dùng lệnh LOAD .

LOAD Table Ddbname..Tablename from DISK ="Path\FileName"

Ngoài ra bạn có thể dùng lệnh:

+ BACKUP DATABASE { database\_name | @database\_name\_var }

TO <backup\_device> [...n] [WITH .. ..]

+ RESTORE DATABASE { database\_name | @database\_name\_var }

[FROM <backup\_device> [...n]]

[WITH [DBO\_ONLY].. ..]

**Chú ý:** Trước khi backup dữ liệu bạn nên lưu lại các thông tin về CSDL hay các TABLE mà bạn muốn BACKUP để sau này khi phục hồi cần thông số chính xác.

10) Lệnh SP\_DBOPTION:

Xem hoặc thay đổi quyền truy cập của các lệnh trong OPTION

**Cú pháp:**

```
sp_dboption [[@dbname =] 'database']  
            [, [@optname =] 'option_name']  
            [, [@optvalue =] 'value']
```

11) Xem thông tin đang hiện hành về các user:

**Cú pháp:**

```
sp_helpuser [[@name_in_db =] 'security_account']
```

12) Lệnh SP\_Password:

Thêm hay chuyển đổi một PASSWORD cho một LOGIN\_ID

**Cú pháp:**



```
sp_password [[@old =] 'old_password',] {[@new =] 'new_password'}  
[,[@loginame =] 'login']
```

13) Dạng lệnh SP\_AddType:

Thiết lập kiểu dữ liệu do người sử dụng định nghĩa

**Cú pháp:**

```
sp_addtype [@typename =] type,  
           [@phystype =] system_data_type  
           [, [@nulltype =] 'null_type']
```

14) Hàm RAISERROR():

Thông báo, cảnh báo lỗi đang xảy ra từ Microsoft SQL Server

**Cú pháp:**

```
RAISERROR ({msg_id | msg_str}{, severity, state}  
[, argument [...n]] ) [WITH option[,...n]]
```

15) Hàm USER\_ID(): Trả về thông số của USER\_ID đang làm việc

**Cú pháp:**

```
USER_ID(['user'])
```

**Tương tự như các hàm còn lại :**

+ Object\_ID(), User\_name(), DB.Name()

16) Câu lệnh SP\_ATTACH\_DB:

Gán dữ liệu vào SERVER.

**Cú pháp:**

```
sp_attach_db [@dbname =] 'dbname',  
             [@filename1 =] 'filename_n' [...16]
```

17) Câu lệnh SP\_AddGroup:

Thêm nhóm USER vào DATABASE

**Cú pháp:**

```
sp_addgroup [@grpname =] 'group'
```

18) Câu Lệnh SP\_AddUSER:

Thêm các USER vào ACCOUNT trong SERVER.

**Cú pháp:**

```
sp_adduser [@loginame =] 'login'
```

```
[,[@name_in_db =] 'user']
```

```
[,[@grpname =] 'group']
```

19) Câu lệnh GRANT:

Gán quyền thực hiện các câu lệnh cho các Table trong database.

**Cú pháp:**

```
GRANT {ALL | statement[,...n]} TO security_account[,...n]
```

**Ứng dụng cho permissions:**

```
GRANT {ALL [PRIVILEGES] | permission[,...n]} {
```

```
[(column[,...n])] ON {table | view} | ON {table | view} [(column[,...n])]
```

```
| ON {stored_procedure | extended_procedure} } TO
```

```
security_account[,...n]
```

```
[WITH GRANT OPTION] [AS {group | role}]
```

20) Huỷ bỏ các câu lệnh vừa được gán hay các quyền được thiết lập trong PERMISSION:

**Cú pháp:**

```
REVOKE {ALL | statement[,...n]}
```

```
FROM security_account[,...n]
```

**Ứng dụng cho permissions:**

```
REVOKE [GRANT OPTION FOR]{ALL [PRIVILEGES] |
permission[,...n]}{[(column[,...n])] ON {table | view}| ON {table |
view}{(column[,...n])| {stored_procedure | extended_procedure }}
{TO | FROM}security_account[,...n][CASCADE][AS {group | role }]
```

21) Câu lệnh Truncate:

Làm sạch (xoá dữ liệu) table mà bạn chỉ định

### Cú pháp:

```
TRUNCATE TABLE table_name
```

### - Kết nối với Excel, kết nối với SPSS

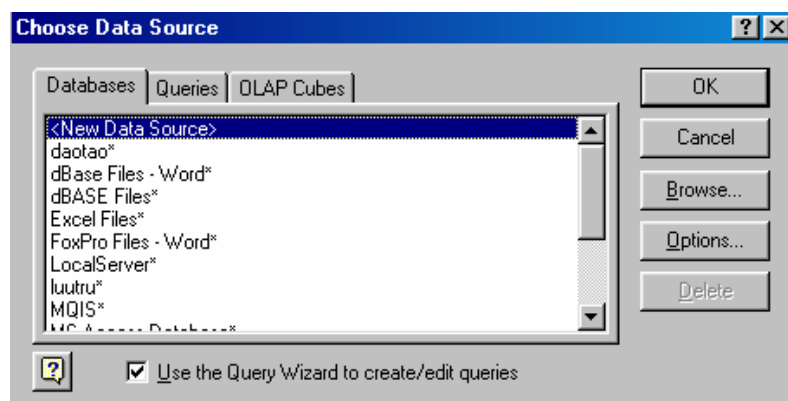
Các trình duyệt Microsoft office như WORD, EXCEL cũng như SPSS cũng có thể cập nhật CSDL của Microsoft SQL Server:

EXCEL trình duyệt lấy dữ liệu từ CSDL của Microsoft SQL Server và lập thành các biểu tính toán theo người lập biểu. Tương tự như phần mềm phân tích số liệu thống kê SPSS cũng vậy.

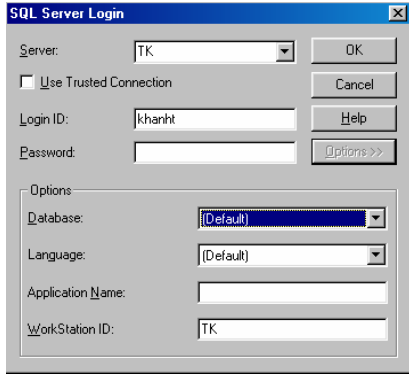
### Lấy dữ liệu từ database bằng EXCEL:

Trước khi thực hiện các bước sau bạn đã phải định nghĩa cấu hình database trong ODBC qua FSN.

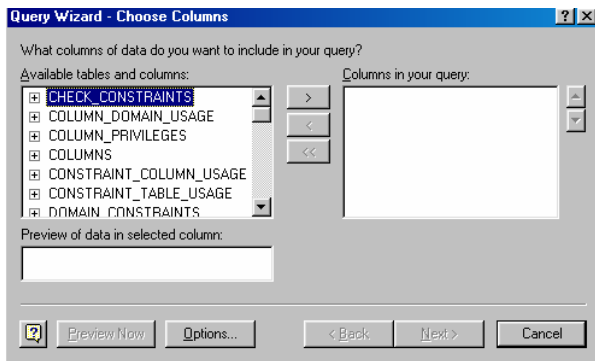
1) Mở EXCEL -> vào menu data -> Chọn Get External data -> Chọn tiếp Create New Query



Nhìn vào hình trên bạn phải chọn database cần lấy dữ liệu. Sau đó bạn phải kết nối với database này bằng cách sử dụng Logon\_ID và Password.



- 2) Tiếp theo bạn phải chọn các bảng (Table) và các cột (trường) trong mỗi bảng đó để tạo ra Query.

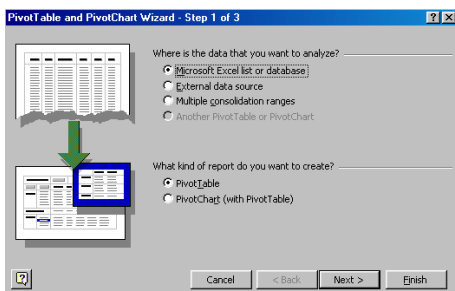


- 3) Sau khi chọn xong table và các cột tiếp theo bạn nhấn NEXT
- 4) Nếu bạn muốn sắp xếp (SORT ORDER) thì bạn chọn các cột cần sắp xếp ở đây.

Bước tiếp bạn chọn loại dữ liệu cần đưa ra (EXCEL, QUERY, OLAP) -> Nhấn Finish.

### Tạo các table dưới dạng cột bằng EXCEL: Gồm 3 bước chính

- 1) Mở EXCEL -> vào menu data -> Chọn Pivot table and Pivotchart Report



Chọn loại dữ liệu cần lấy để làm biểu

2) Get data:

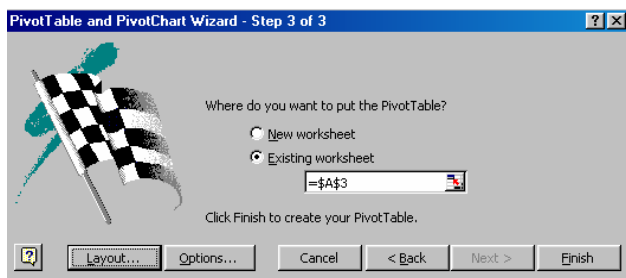


Nhấn vào get data bạn phải chọn database cần truy nhập (tương tự như trên).

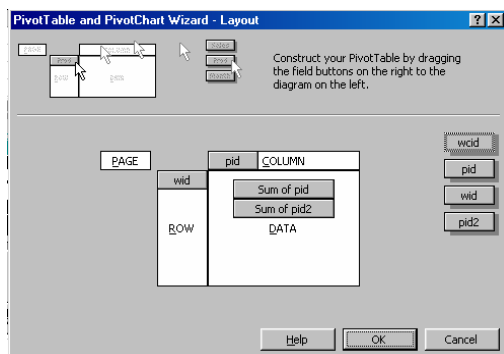
Tiếp theo là bạn phải kết nối vào database bằng cách dùng Logon\_ID và Password. (tương tự như ở trên)

Sau khi kết nối vào database bạn phải chọn các table và các cột cần truy xuất. -> tiếp theo nhấn NEXT

3) Dùng hiệu ứng LAYOUT để bố trí hàng cột của biểu đầu ra



Dùng chuột để kéo và thả các trường nằm trong cột của báo biểu làm sao cho thoả yêu cầu đặc ra -> tiếp theo nhấn OK .

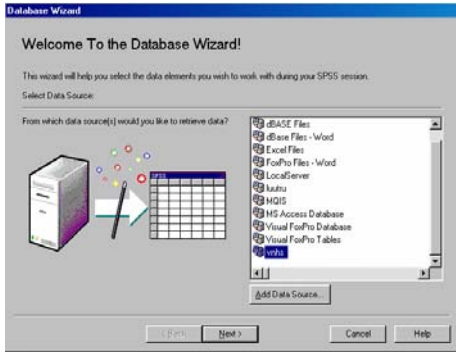


Tiếp nhấn FINISH để hoàn thành công việc.

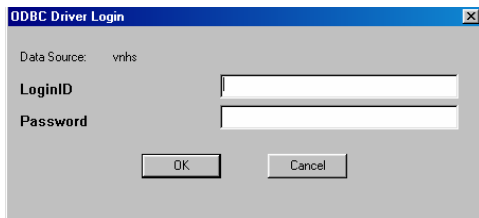
**Lấy dữ liệu bằng SPSS: Gồm 6 bước chính**

Tương tự như EXCEL trước khi lấy dữ liệu bạn phải thiết lập database trong ODBC.

1) Mở SPSS-> Chọn menu FILE-> Chọn Open data -> Chọn New Query

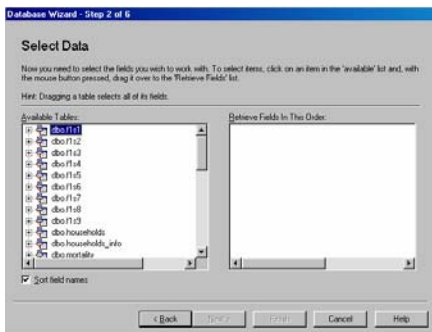


Từ đây bạn chọn CSDL cần truy xuất. Tuy nhiên ngay tại đây bạn cũng có thể thêm data source (database) vào ODBC. Khi chọn được database bạn nhấn NEXT (cho bạn hình sau).



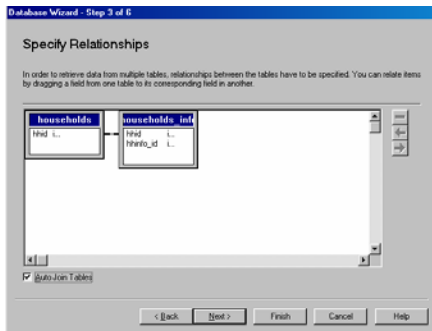
Hoàn toàn tương tự như EXCEL, bạn cũng phải kết nối vào Database qua ODBC bằng login\_ID và password. (nhấn OK và chờ một chút xíu)

2) Chọn dữ liệu (Select data):



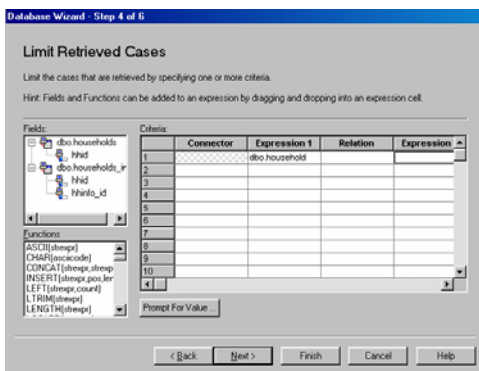
Tại đây bạn chọn các table và các trường nằm trong các table để đưa ra báo biểu. Ở đây, khi bạn cần sắp xếp một cột nào đó bạn check vào ô “sort field names”.

- 3) Chỉ định các mối quan hệ về dữ liệu, nếu bạn muốn điều khiển tự động bạn check vào ô Auto Join Tables.



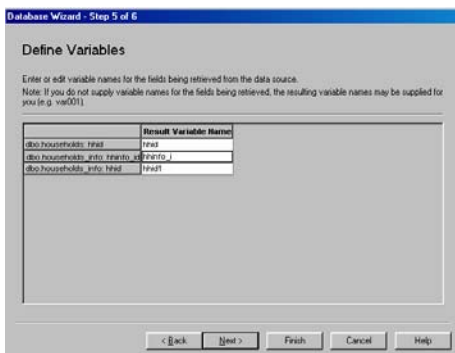
tiếp theo nhấn NEXT.

- 4) Sử dụng các hàm(function) và các toán tử logic để giới hạn dữ liệu theo nh cấu của báo biểu.



tiếp theo nhấn NEXT.

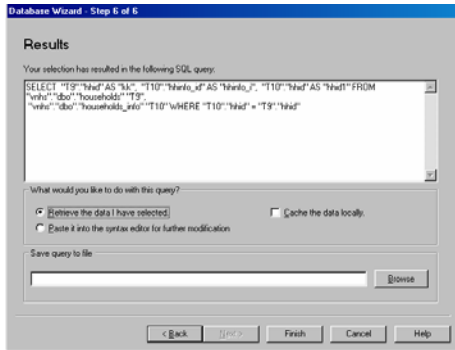
- 5) Định nghĩa các biến tương ứng với các cột dữ liệu cần đưa ra báo biểu.



Nhập tên biến vào ô **Result Variable name**:

Tiếp nhấn NEXT

- 6) Xem lại lần cuối bằng câu lệnh (ngôn ngữ SQL) có điều gì cần sửa.  
Sau đó, nếu bạn muốn ghi lại vào query thì bạn phải chọn tên query (cần ghi lại) vào ô **Save Query to File**.



nhấn FINISH để hoàn thành. (Trình áp dụng).

## 5.5. Một ví dụ minh họa (thể hiện theo thiết kế ở phần II).

**Ví dụ:** Tạo CSDL lưu trữ của ngành thống kê, hiện nay CSDL này đang được khai thác và cập nhật tại Tổng Cục Thống kê (Văn phòng Tổng Cục đang quản lý tài liệu lưu trữ).

### 1) Tạo database có tên là LUUTRU:

```
CREATE DATABASE LUUTRU ON
```

```
( NAME = luutru_dat, FILENAME = 'c:\mssql7\data\luutrudat.mdf',  
SIZE = 10, MAXSIZE = 50, FILEGROWTH = 5 )
```

```
LOG ON
```

```
( NAME = 'luutru_log', FILENAME = 'c:\mssql7\data\luutrulog.ldf',  
SIZE = 5MB, MAXSIZE = 25MB, FILEGROWTH = 5MB )
```

### 2) Tạo các table trong CSDL và thiết lập các chỉ số index:

```
+ create table dmluutru (ma1 char(1) null, ma2 char(1) null, ma3 char(1)  
null, ma4 char(1) null, ma5 char(1) null, tenPl varchar(120) not null)
```

```
+ create table NoiBH( mabh char(2) not null, tenbh varchar(50) not null)
```

```
+ insert NoiBH values("01", "Tổng Cục Thống Kê")
```



```
+ DELETE FROM DMLUUTRU WHERE TENPL="Tài liệu lưu trữ
TCTK"
+ create table dmlydo( lydo char(1) not null, tenlydo varchar(40) not null)
+ create table dmbiendong( mabd char(1) not null, tenbd varchar(20) not
null)
+ insert dmbiendong values("1","Thanh lý")
+ insert dmlydo values("1","Hết thời hạn bảo quản")
+ update dmlydo set tenlydo="Hết giá trị sử dụng" where lydo="2"
+ truncate table luu_file
+ create table HOSO (hoso_id char(10) not null, ngay char(2) null, thang
char(2) null, nam char(4) not null, ten varchar(255) not null, soto smallint
null, gia char(2) not null, ngan char(2) not null, cap char(3) null, kyhieu
varchar(15) null, noibh char(2) null, khi char(1) not null, loai char(1) not
null, nhom char(1) null, vande char(1) null,)
+ create table HOSOCU (hoso_id char(10) not null, ngay char(2) null,
thang char(2) null, nam char(4) not null, ten varchar(255) not null, soto
smallint null, kyhieu varchar(15) null, noibh char(2) not null, khi char(1)
not null, loai char(1) not null, nhom char(1) null, vande char(1) null,
ngaybo char(2) not null, thangbo char(2) not null, nambo char(4) not null,
loaibd char(1) not null, lydo char(1) not null, nduyet varchar(25) not null)
+ create table MUON (hoso_id char(10) not null, ngaym char(2) not null,
thangm char(2) not null, namm char(4) not null, ngoaim varchar(25) not
null, donvi char(2) not null, chatluong char(1), ngayh char(2) null, thangh
char(2) not null, namh char(4) not null, ngayt char(2) not null, thangt
char(2) not null, namt char(4) not null)
+ create table IN_HS( nam char(4) null, khi char(1) null, loai char(1) null,
nhom char(1) null, vande char(1) null, sodem smallint not null)
+ create unique clustered index hosoId_ind on hoso(hoso_id)
```

```
+ create index maphanloai_ind on hosokhloi,loai,nhom,vande)
+ create index nambh_ind on hosonam)
+ create unique clustered index manuibh_ind on noibhmabh)
+ create unique clustered index macl_ind on chatluongmacl)
+ create unique clustered index mabd_ind on dmbiendongmabd)
+ create unique clustered index mapl_ind on dmluutru(ma1,ma2,ma3,ma4)
+ create unique clustered index mald_ind on dmlydo(lydo)
+ create unique clustered index madv_ind on donvimadv)
+ create index hosoi_ind on muon(hosoi)
+ create index hosocu_ind on hosocuhosoi)
```

Sau khi thực hiện xong các lệnh trên bạn dùng lệnh: sp\_help để xem và kiểm tra lại có đủ tất cả table trong CSDL chưa?

**3) Bảo mật CSDL: tạo nhóm làm việc và thêm các user vào nhóm, gán quyền thao tác trên CSDL đó.**

```
+ sp_addgroup dbCSDL
+ sp_adduser khanht,khanht,dbCSDL
+ grant select on dmluutru to dbCSDL
+ grant select,update,delete,insert on NOIBH to dbCSDL
+ revoke update,delete,insert on HOSO to dbCSDL
..
```

## **PHẦN VI. TÌNH HÌNH PHÁT TRIỂN CSDL THỐNG KÊ TRÊN SQL SERVER TRONG THỜI GIAN QUA VÀ KẾT LUẬN**

**Chú ý :** nhắc lại một số tiêu thức quan trọng khi sử dụng Microsoft SQL Server .

### **Thiết kế CSDL cho Microsoft SQL Server :**

Nếu bạn có một CSDL tốt : Tức là từ lúc thiết kế - bạn phải đặt ra yêu cầu mà bạn cần khai thác ở CSDL. Từ đó thu thập thông tin về các đối tượng của CSDL; đầy đủ, chính xác, không dư thừa, đúng thời điểm. Từ đó thiết kế mô hình quan hệ của các thuộc tính thông tin có liên quan (Gọi là có phụ thuộc hàm giữa các thuộc tính). Thiết lập các khoá: Khoá trong và khoá ngoài. Tạo ra bao nhiêu bảng(table) trong CSDL? Khi đã chuẩn bị đầy đủ thì chúng ta bắt đầu thiết kế database.

### **Các ứng dụng được áp dụng vào để khai thác thông tin trên CSDL của Microsoft SQL Server :**

Microsoft SQL Server cung cấp cho bạn bộ lệnh trong TRANSACT SQL để khai thác dữ liệu trong CSDL.

Ngoài ra bạn cũng có thể dùng DMO (các ứng dụng OLE hỗ trợ 32 bit) để tạo ra công cụ nhằm khai thác dữ liệu. Ví dụ như VB, VC++.. .. Tuy nhiên nếu dùng VB thì phải qua ODBC(16-32bit) và phải viết các hàm truy cập này( gọi là DAO FUNCTION) và rất dễ sử dụng, nhưng nếu bạn dùng Delfi hay VC++ thì không nhất thiết phải qua ODBC (32bit). Và rất tiện lợi cho việc mở rộng các chức năng thiên về kỹ thuật chuyên sâu (tuy nhiên khó

sử dụng hơn Visual Basic), tốc độ truy nhập nhanh hơn nhiều so với cách sử dụng Visual Basic.

Hiện nay, Tổng Cục Thống Kê đã xây dựng một số CSDL cho ngành như:

- + CSDL Công nghiệp
- + CSDL doanh nghiệp
- + Dân số và nhà ở
- + Quản lý công văn đi đến
- + CSDL lưu trữ ..

các CSDL này luôn luôn được phát triển theo thời gian ngày một lớn hơn.

Sách điện tử, luận văn tốt nghiệp CNTT – <http://www.diachiweb.com>



Đưa lên mạng bởi Nghiêm Phú Cường, <http://www.diachiweb.com>