

**TRƯỜNG ĐẠI HỌC HÀNG HẢI
KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI GIẢNG
HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

Biên soạn: K.S Nguyễn Văn Thành

Hải Phòng – 2008

B GIAO THÔNG VÀ NT I
TRƯỜNG ĐẠI HỌC CHÂNG H I
KHOA CÔNG NGHỆ THÔNG TIN
B MÔN HỌC THÔNG TIN

BÀI GIẢNG
HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU

TÊN HỌC PHẦN : CƠ SỞ DỮ LIỆU
MÃ HỌC PHẦN : 17402
TRÌNH ĐỘ TỐT NGHIỆP : ĐẠI HỌC CHÍNH QUY
DÙNG CHO SINH VIÊN NGÀNH : CÔNG NGHỆ THÔNG TIN

H I PHÒNG - 2008

Tên học phần: Học quản trị Cơ sở dữ liệu
Bộ môn phụ trách giảng dạy: Học trường Thông tin
Mã học phần: 17402

Loại học phần: 4
Khoa phụ trách: CNTT.
Tổng số TC: 4

TS tiết	Lý thuyết	Thực hành/ Xemina	T h c	Bài tập l n	án môn h c
90	45	45	0	x	0

Điều kiện tiên quyết:

Không yêu cầu.

Mục tiêu của học phần:

Cung cấp cho sinh viên những khái niệm cơ bản về học quản trị cơ sở dữ liệu, vai trò và chức năng của học quản trị cơ sở dữ liệu trong quá trình công tác giảng dạy học quản trị cơ sở dữ liệu và các phần mềm quản lý cơ sở dữ liệu. Giúp sinh viên tiếp cận với những học quản trị cơ sở dữ liệu Microsoft SQL Server.

Nội dung chủ yếu:

Khái niệm về CSDL và học quản trị CSDL; Các thành phần cơ bản trong MS SQL Server; Ngôn ngữ T-SQL và các loại học CSDL; Các tác vụ học quản trị học trường.

Nội dung chi tiết:

TÊN CHƯƠNG M C	PHÂN PH IS TI T				
	TS	LT	BT	TH	KT
Chương 1. Giới thiệu	6				
1.1. Mục đích khái niệm cơ bản					
1.2. Giới thiệu về SQL Server và mô hình Client/Server					
1.3. Cài đặt và cấu hình SQL Server					
1.4. Các thành phần cơ bản trong SQL Server					
Chương 2. Làm việc với CSDL và bảng	9				1
2.1. Làm việc với CSDL					
2.1.1. Tạo học tập tin trong CSDL SQL Server					
2.1.2. Tạo học CSDL					
2.1.3. Sửa học CSDL					
2.1.4. Xóa học CSDL					
2.2. Làm việc với bảng dữ liệu					
2.2.1. Các thuộc tính cơ bản					
2.2.2. Ràng buộc (Constraint) và thuộc tính					
2.2.3. Tạo học bảng dữ liệu					
2.2.3. Sửa học bảng dữ liệu					
2.2.4. Xóa học bảng dữ liệu					
2.3. Làm việc với các học ghi					
2.3.1. Thêm học ghi mới (Insert)					
2.3.2. Cập nhật học ghi (Update)					
2.3.3. Xóa học ghi (Delete)					
2.4. Khái niệm về học (Index)					
Chương 3. Truy vấn dữ liệu và học o (View)	9				1
3.1. Cấu trúc truy vấn cơ bản					
3.1.1. Câu học SELECT					
3.1.2. Học WHERE và học điều kiện					
3.1.3. Mục đích học dùng					
3.1.4. Truy vấn học kê học GROUP BY và HAVING					
3.1.5. Học xếp học ORDER BY					
3.2. Truy vấn dữ liệu học học					
3.2.1. Inner Join					
3.2.2. Left Outer Join					

TÊN CHƯƠNG MỤC	PHÂN PHỐI SỐ TIẾT				
	TS	LT	BT	TH	KT
3.2.3. Right Outer Join					
3.2.4. Full Outer Join					
3.2.5. Cross Join					
3.2.6. Self Join					
3.2.7. Non - Equal Join					
3.2.8. Union					
3.3. Truy vấn lồng nhau (Subquery)					
3.3.1. Nested Scalar Subquery					
3.3.2. Correlated Subquery					
3.4. Bảng nhìn (View)					
3.4.1. Bảng nhìn a View					
3.4.2. Bảng nhìn b View					
3.4.3. Xóa View					
Chương 4. Thủ tục thủ tục và hàm người dùng	9				1
4.1. Lập trình với T - SQL					
4.2. Thủ tục thủ tục (Stored Procedure)					
4.3. Hàm người dùng người dùng (User Defined Function)					
Chương 5. Trigger và Transaction	6				
5.1. Trigger và người dùng của Trigger					
5.1.1. After Trigger					
5.1.2. Instead Of Trigger					
5.2. Các loại Transaction và người dùng					
5.2.1. Transaction người dùng					
5.2.2. Transaction người dùng mình					
5.2.3. Transaction người dùng					
5.2.4. Transaction lồng nhau					
Chương 6: Các tác vụ quản trị hệ thống	6				
6.1. Quản lý đăng ký SQL Server (Server Registration)					
6.2. Backup và phân quyền người dùng					
6.3. Sao lưu (Backup) và phục hồi (Restore) dữ liệu					
6.4. Hồ sơ hệ thống quản trị					

Nhiệm vụ của sinh viên:

Tham dự các buổi học lý thuyết và thực hành, làm các bài tập cá nhân, làm các bài thi giữa học kỳ và bài thi kết thúc học kỳ theo đúng quy định.

Tài liệu học tập:

1. Đặng Quang Thiển, *SQL Server 2000: Lập trình T - SQL*, NXB Văn hóa Sài Gòn, 2007.
2. Paul Turley & Dan Wood, *Beginning Transact-SQL with SQL Server 2000 and 2005*, Wrox Press, 2006.
3. Brian Knight et al, *Professional SQL Server 2005 Administration*, Wrox Press, 2007.
4. Ray Rankins, Paul Bertucci, Chris Gallelli, Alex T. Silverstein, *Microsoft SQL Server 2005 Unleashed*, Sams Publishing, 2007.

Hình thức và tiêu chuẩn đánh giá sinh viên:

- Hình thức thi: thi viết học viên áp.
- Tiêu chuẩn đánh giá sinh viên: căn cứ vào sự tham gia học tập của sinh viên trong các buổi học lý thuyết và thực hành, kết quả làm các bài tập cá nhân, kết quả các bài thi giữa học kỳ và bài thi kết thúc học kỳ.

Thang điểm: Thang điểm A, B, C, D, F.

Điểm đánh giá học kỳ: $Z = 0,4X + 0,6Y$.

Bài giảng này là tài liệu **chính thức và chuyên t** của Bộ môn Hệ thống Thông tin, Khoa Công nghệ Thông tin và **được** giảng dạy cho sinh viên.

Trang Bộ môn

Ngày phê duyệt: / / .

M C L C

M C L C.....	1
M U.....	4
PH N I. QU NTR SQL SERVER.....	4
B T U V I SQL SERVER	4
TÌM HI U V H QU NTR CSDL SQL SERVER	4
MÔ HÌNH HO T NG C A SQL SERVER TRÊN M NG MÁY TÍNH.....	6
CÁC THÀNH PH N C A SQL SERVER.....	11
CÁC THÀNH PH N C A SQL SERVER.....	11
CÀI T SQL SERVER.....	13
QU NTR SERVER	22
INSTANCE.....	22
I U KHI N CÁC D CH V C A SQL SERVER.....	22
QU NTR SERVER.....	26
THI T L P K T N I N SERVER.....	27
C U HÌNH K T N I M NG C A SERVER.....	37
QU NTR CÁC CLIENT.....	38
QU NTR C S D LI U.....	45
C U TRÚC C S D LI U.....	45
QU N LÝ C S D LI U.....	50
B NG D LI U – TABLE.....	58
CÁC CHU NT C.....	58
THI T K B NG D LI U.....	60
T O B NG D LI U.....	69
KHÓA INDEX	77
THI T K KHÓA INDEX.....	77
T O KHÓA INDEX.....	79
XÓA INDEX.....	81
KHUNG NHÌN – VIEW	82
KHÁI NI M KHUNG NHÌN.....	82
T O KHUNG NHÌN.....	82
S D NG VIEW.....	84
TH T CL UTR	86
KHÁI NI M TH T CL UTR VÀ HÀM.....	86

PHÂN LOẠI THỰC DỤNG	87
THIẾT LẬP THỰC DỤNG	88
SẠ, XÓA THỰC DỤNG.....	95
TRIGGER.....	96
KHÁI NIỆM TRIGGER.....	96
NHÀNG TRIGGER.....	96
CẤM CHẤM TRIGGER.....	96
TẠO TRIGGER.....	97
SẠ, XÓA TRIGGER.....	101
XUẤT-NHẬP DỮ LIỆU.....	102
SERVER LIÊN KẾT – LINKED SERVER.....	102
SẠ DỮ LIỆU BCP VÀ BULK INSERT NHẬP DỮ LIỆU.....	110
DETACH VÀ ATTACH CẤM DỮ LIỆU.....	113
IMPORT VÀ EXPORT CẤM DỮ LIỆU.....	115
EXPORT – XUẤT DỮ LIỆU.....	119
SAO LƯU, KHÔI PHỤC DỮ LIỆU.....	120
NHÀNG LÝ DO PHỤC SAO LƯU VÀ KHÔI PHỤC DỮ LIỆU.....	120
CÁC LOẠI BACKUP.....	120
CÁC MÔ HÌNH PHỤC HỒI DỮ LIỆU.....	121
SAO LƯU CẤM DỮ LIỆU - BACKUP DATABASE.....	122
KHÔI PHỤC DỮ LIỆU – RESTORE DATABASE.....	123
CHẾ ĐỘ X. PHÂN QUYỀN, BẢO MẬT.....	125
CHẾ ĐỘ BẢO MẬT – SECURITY MODE.....	125
SERVER ROLE, DATABASE ROLE.....	127
QUẢN LÝ DỮ LIỆU.....	131
NHÂN BẢN DỮ LIỆU.....	133
GIỚI THIỆU VỀ NHÂN BẢN DỮ LIỆU.....	133
CẤM HÌNH PUBLISHER VÀ DISTRIBUTOR.....	139
TẠO PUBLICATION.....	141
TẠO PUSH SUBSCRIPTION.....	143
TẠO PULL SUBSCRIPTION.....	145
THỰC HIỆN NHÀNG DỮ LIỆU.....	147
PHẦN II. CÂU LỆNH T-SQL	148
NHÀNG A DỮ LIỆU (DATA DEFINITION LANGUAGE - DDL).....	148
THAO TÁC VỚI DỮ LIỆU (DATA MANIPULATION LANGUAGE - DML).....	153
TRUY VẤN DỮ LIỆU.....	165

<i>T O B N G B N G L N H SELECT INTO.</i>	173
<i>L N H COMPUTE BY.</i>	173
<i>TOÁN T UNION.</i>	174
<i>TRUY V N D L I U T N H I U B N G.</i>	175
<i>TRUY V N T N G H P.</i>	183
<i>TRUY V N L N G N H A U.</i>	186
<i>UPDATE, DELETE, INSERT V I L N H TRUY V N L N G N H A U.</i>	189
<i>L N H READTEXT – C TEXT, IMAGE.</i>	190
<i>THAO TÁC D L I U N G OÀI.</i>	190
<i>M T S HÀM C B N.</i>	193
<i>TRANSACTION – PHIÊN GIAO D CH.</i>	199
<i>LOCKING – KHÓA.</i>	203
<i>GRAND – GÁN QUY N.</i>	206
<i>REVOKE – T C QUY N.</i>	211
<i>DENY – T CH I QUY N.</i>	211
<i>TR GIÚP.</i>	212
PH N III. PHÁT TRI N N G D N G V I SQL SERVER.	213
<i>GI I THI U.</i>	213
<i>K T N I V I SQL SERVER B N G ADO.</i>	213
<i>K T N I V I SQL SERVER B N G SQL-DMO.</i>	233

Phần 1. QUẢN TRỊ SQL SERVER

BỘ TỤ VỚI SQL SERVER

TÌM HIỂU VỀ HỆ QUẢN TRỊ CSDL SQL SERVER

Giới thiệu SQL Server.

SQL Server là hệ thống quản trị cơ sở dữ liệu quan hệ (Relational DataBase Management System- RDBMS) sử dụng các ngôn ngữ chuyên Transaction-SQL trao đổi dữ liệu giữa Client Computer và Server Computer.

SQL Server có một số đặc tính sau:

- Cho phép quản trị một hệ CSDL lớn (lên đến vài tera byte), có tốc độ xử lý dữ liệu nhanh đáp ứng yêu cầu về thời gian.
- Cho phép nhiều người cùng khai thác trong một thời điểm với một CSDL và toàn bộ quản trị CSDL (lên đến vài chục ngàn user).
- Có hệ thống phân quyền bổ mốt tiên tiến thích với hệ thống bổ mốt của công nghệ NT (Network Technology), tích hợp với hệ thống bổ mốt của Windows NT hoặc sử dụng hệ thống bổ mốt của SQL Server.
- Hỗ trợ trong việc triển khai CSDL phân tán và phát triển ứng dụng trên Internet
- Cho phép lập trình kết nối với nhiều ngôn ngữ lập trình khác dùng xây dựng các ứng dụng đồ họa (Visual Basic, C, C++, ASP, ASP.NET, XML,...).
- Sử dụng các ngôn ngữ truy vấn dữ liệu Transaction-SQL (Access là SQL, Oracle là PL/SQL).

Các phiên bản của SQL Server.

SQL Server có các phiên bản chính sau:

- Enterprise Manager: Là phiên bản đầy đủ của SQL Server có thể chạy trên 32CPU và 64GB RAM. Có các dịch vụ phân tích dữ liệu Analysis Service.
- Standard: Giống như Enterprise nhưng bổ mốt tính năng cao cấp, có thể chạy trên 2CPU, 4GB RAM.

- Personal: Phiên bản này chủ yếu chạy trên PC, nên có thể chạy trên các hệ điều hành Windows 9x, Windows XP, Windows 2000, Windows 2003...
- Developer: Là phiên bản tổng thể Enterprise nhằm giúp ích cho nhà phát triển.
- Desktop Engine: Là phiên bản motor engine chạy trên desktop và không có giao diện đồ họa (GUI), kích thước CSDL giúp ích cho bộ nhớ 2GB.
- Win CE: Sử dụng cho các ứng dụng chạy trên Windows CE.
- Trial: Phiên bản dùng thử, giúp ích cho nhà phát triển.
- SQL Client: Là phiên bản dành cho máy khách, khi thiết lập kết nối khai thác server, phiên bản này cung cấp giao diện GUI khai thác cho người sử dụng.
- SQL Connectivity only: Là phiên bản sử dụng chỉ cho các ứng dụng kết nối với SQL Server, phiên bản này không cung cấp công cụ GUI cho người dùng khai thác SQL Server.

Các phiên bản này cần cài đặt phần mềm vào bộ cài đặt mà bạn chọn hoặc là cần khai cài đặt (ví dụ phiên bản Enterprise, Standard, Personal,... bản phù hợp theo bộ cài đặt, phiên bản SQL Client, Connectivity,... do bản chọn trong các hộp thoại trong quá trình cài đặt).

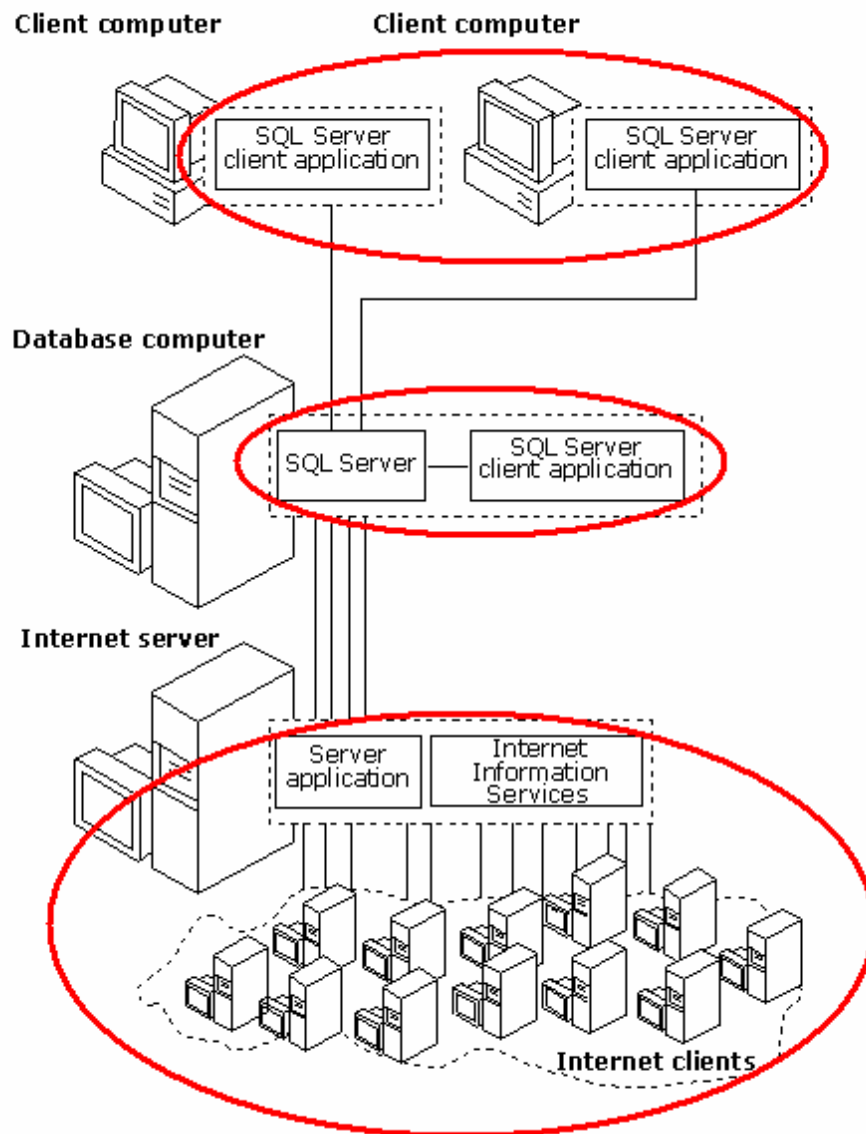
Một số tính năng của Enterprise manager.

- Dễ cài đặt
- Hỗ trợ mô hình Client/Server.
- Thích hợp trên các hệ điều hành Windows.
- Hỗ trợ giao diện người giao thức truy vấn thông.
- Hỗ trợ dịch vụ Data Warehousing.
- Thích hợp với chuẩn ANSI/ISO SQL-92.
- Hỗ trợ nhân bản dữ liệu.
- Cung cấp dịch vụ tìm kiếm Full-Text.
- Sách trợ giúp- Book Online.

MÔ HÌNH HO T NG C A SQL SERVER TRÊN M NG MÁY TÍNH.

Mô hình chung SQL Server trên m ng.

SQL Server là h qu n tr CSDL ho t ng trên m ng, có th th c hi n trao i d li u theo nhi u mô hình m ng khác nhau, nhi u giao th c và ph ng th c truy n tin khác nhau.



Trong s trên th hi n ba ki u k t n i ng d ng n SQL Server:

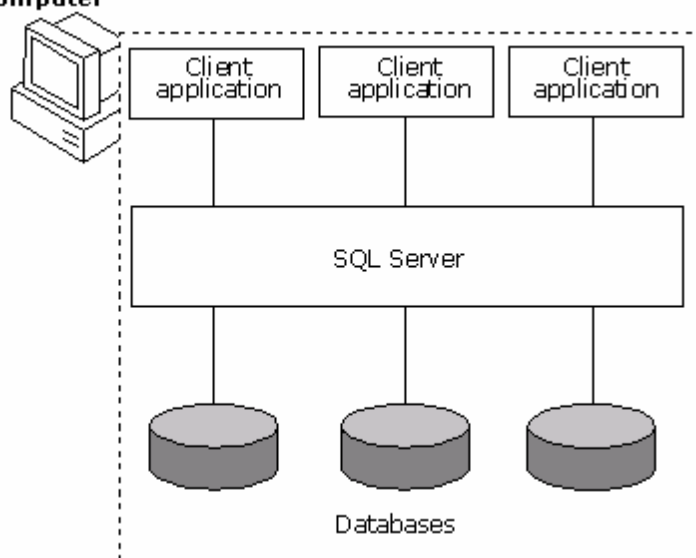
- K t n i trên Desktop: Có th trên cùng máy tính v i SQL Server ho c k t n i qua m ng n i b .

- Kết nối qua mạng địa phương: Thông qua mạng truy cập mạng xa kết nối vào SQL Server.
- Kết nối qua mạng Internet: Các ứng dụng kết nối thông qua máy chủ Internet, dịch vụ IIS thực hiện ứng dụng trên Internet (ASP, JSP, ASP.net,...)

Mô hình Desktop.

Nếu xét trên một máy Desktop sẽ kết nối trao đổi dữ liệu có thể hiện như sau:

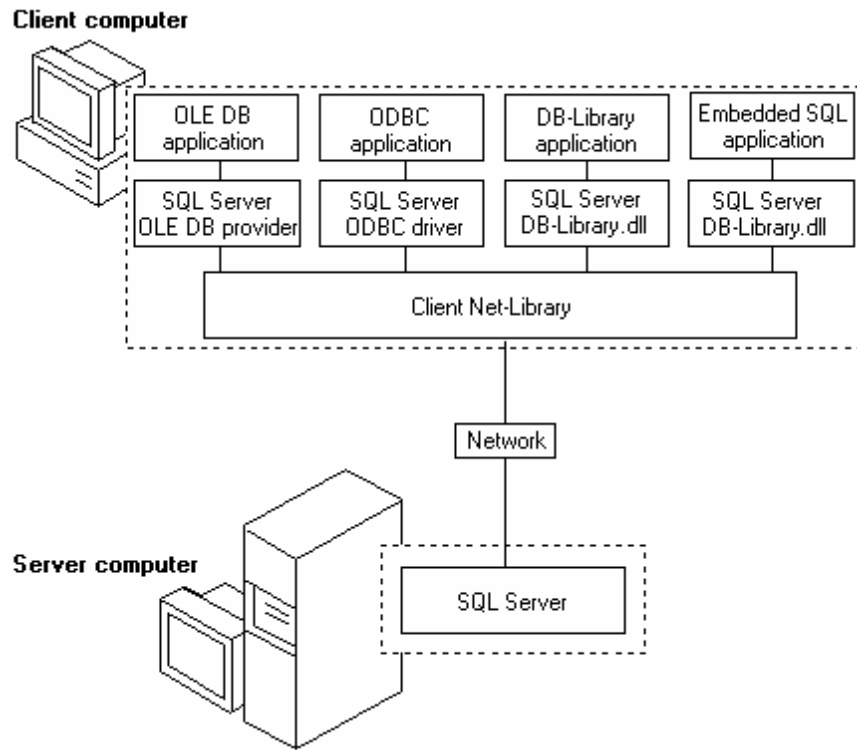
Desktop computer



Trên một Desktop có thể có nhiều ứng dụng, mỗi ứng dụng có thể thực hiện thao tác với nhiều CSDL.

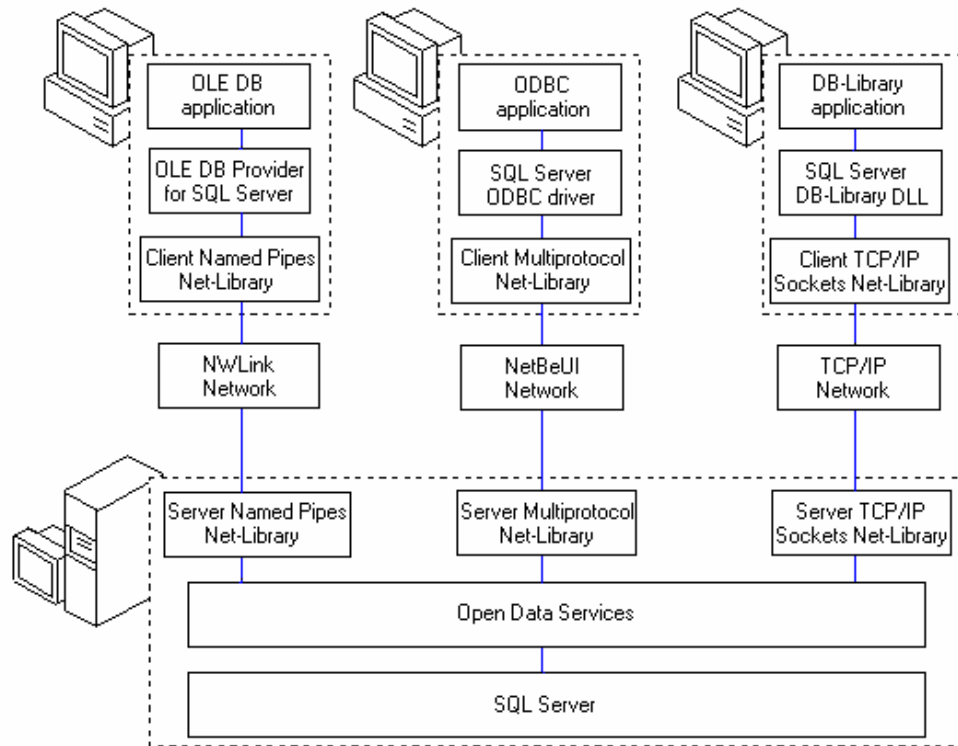
Mô hình Client/Server.

Nếu xét theo mô hình client/server, ứng dụng trao đổi với SQL Server theo sau:



Nhờ sự trợ giúp của các công cụ kết nối trên nền tảng SQL Server cho phép các ứng dụng kết nối theo các phương thức sau: OLE DB, ODBC, DB-Library, Embedded SQL, đây là các phương thức kết nối thích hợp cho nhà phát triển ứng dụng.

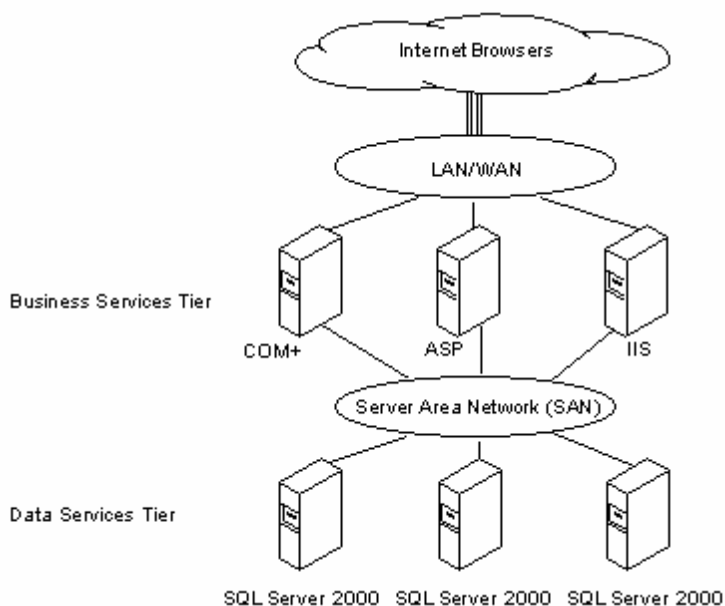
Nếu xem xét cách thức này ta có thể xem sau:



Trong sơ đồ trên cho thấy, SQL Server có thể thực hiện trao đổi dữ liệu với các ứng dụng theo nhiều giao thức truy vấn tin khác nhau (TCP/IP, NetBeUI, Names Pipes,...), các ứng dụng có thể sử dụng nhiều phương thức kết nối khác nhau (OLE DB, ODBC, DB-Library).

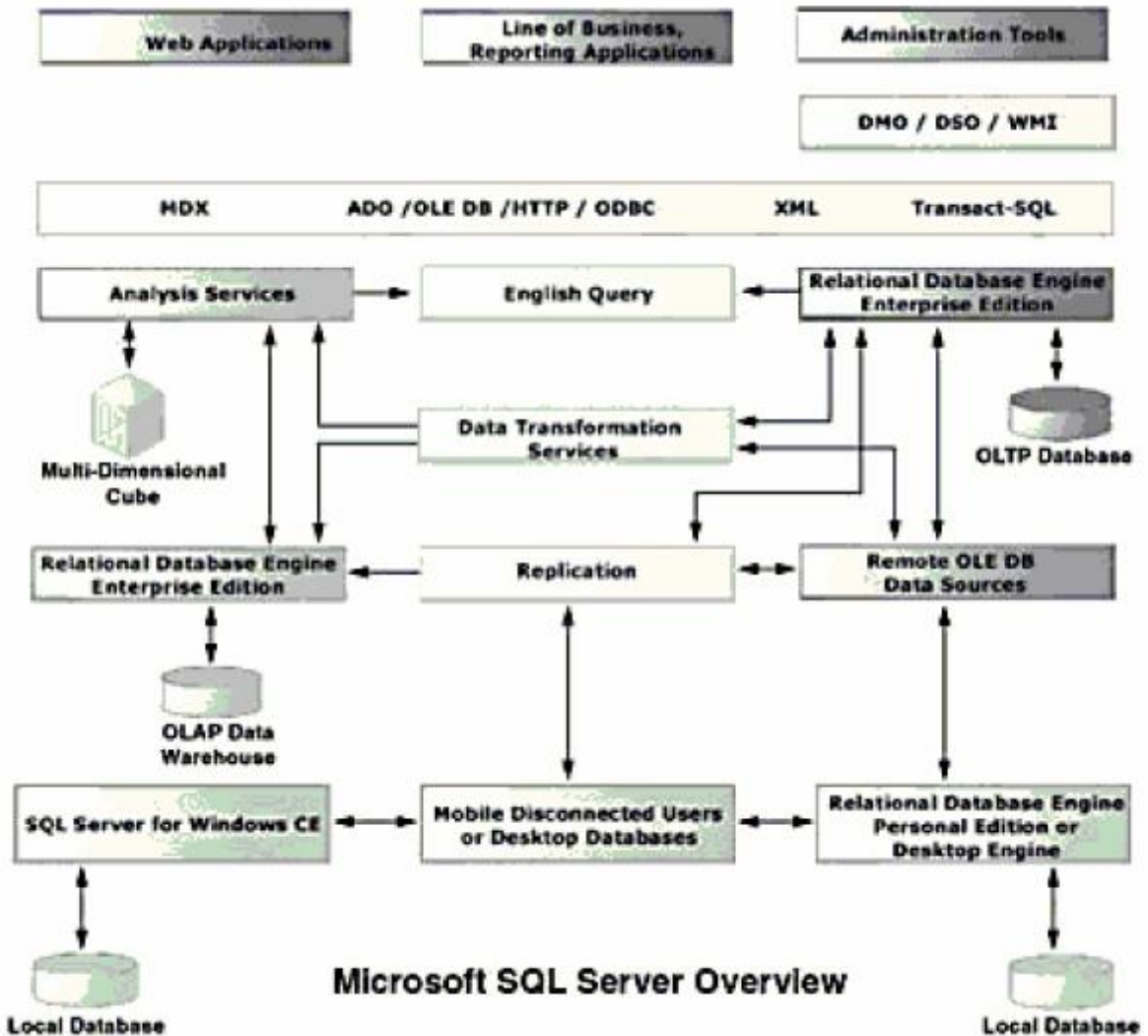
Mô hình kiến trúc ứng dụng trên mạng Internet.

Nếu xét riêng các ứng dụng kiến trúc dựa trên SQL Server trên mạng Internet, các máy chủ SQL Server sẽ được quản lý thông qua các hệ thống máy chủ mạng, hệ thống điều hành mạng, các ứng dụng (COM+, ASP, IIS) sẽ thông qua máy chủ kiến trúc dựa trên SQL Server, mô hình này có thể áp dụng cho các mạng nội bộ, di động, ứng dụng được khai thác trên trình duyệt Internet Browser. Xem xét mô hình dưới đây:



CÁC THÀNH PHẦN CỦA SQL SERVER.

SQL Server có các thành phần và nhiều thành phần khác nhau, các thành phần có mối quan hệ trong một hệ thống, phụ thuộc vào nhau và tạo thành một giải pháp hoàn chỉnh, nâng cao hiệu quả quản trị, phân tích, lưu trữ dữ liệu.



Relational DataBase Engine.

Đây là một engine có khả năng chấp nhận dữ liệu từ nhiều quy mô khác nhau, theo định dạng, hỗ trợ nhiều phương thức kết nối ADO, OLE DB, ODBC.

Replication.

Là công cụ dùng nhân bản dữ liệu, bạn có thể tạo một Server khác với bản gốc để lưu trữ dữ liệu trên Server chính. Công cụ tổ chức việc sao chép dữ liệu giữa Server chính và Server nhân bản. Mục đích của việc tạo Server nhân bản là giảm tải

cho Server chính, nâng cao hiệu quả xử lý dữ liệu, phiên giao dịch online.

Data Transformation Service – DTS.

Là công cụ giúp bạn chuyển dữ liệu giữa các Server quản trị CSDL khác nhau, DTS có thể chuyển dữ liệu từ SQL Server sang Oracle, Access, DB,... trực tiếp khi chuyển dữ liệu DTS sẽ định nghĩa dữ liệu chuyển sang hệ quản trị CSDL khác.

Analysis service.

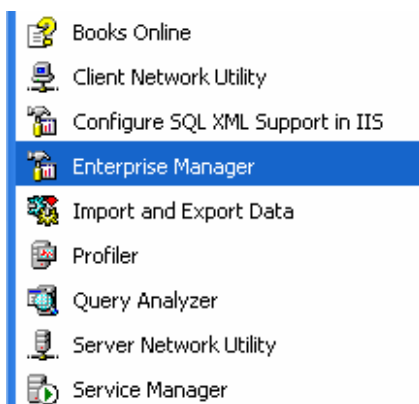
Là công cụ giúp khai thác phân tích dữ liệu, hay khai phá dữ liệu theo phương thức nào đó. Tất cả dữ liệu sẵn có bạn có thể khai phá rồi đưa ra những nhận xét, phân tích, đánh giá và dự đoán theo lĩnh vực nào đó, mọi chi tiết trong ngành này đều có thể coi là một tiêu chí xem xét các dữ liệu.

English query.

Đây là công cụ tra cứu dữ liệu bằng tiếng Anh, cú pháp có thể sử dụng theo văn pháp tiếng Anh thông thường.

SQL Server tools.

Là bộ công cụ cung cấp giao diện cho người quản trị như Enterprise manager, Query Analyzer, ...SQL Server sau khi cài đặt SQL Server group gồm những thành phần cần thiết trong group như sau:



Một số công cụ quan trọng: Enterprise manager, Query Analyzer, Profiler..., các công cụ sẽ giúp ích cho việc khai thác sau.

CÀI ĐẶT SQL SERVER.

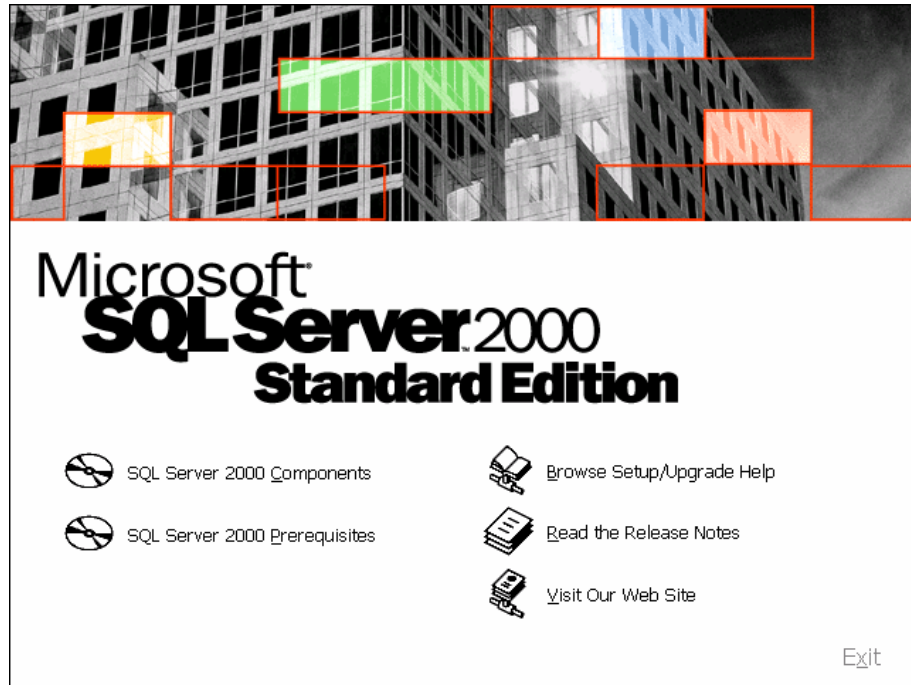
Chuẩn cài đặt.

Tùy theo môi trường của máy tính của bạn mà thực hành cài đặt phiên bản nào, bảng sau là tham số của SQL Server 2000 phiên bản Standard.

Computer	Intel® hoặc tương đương Pentium 166 MHz hoặc cao hơn
Memory (RAM)	Enterprise Edition: Tối thiểu 64 MB, 128 MB hoặc nhiều hơn. Standard Edition: Tối thiểu 64 MB. Personal Edition: Tối thiểu 64 MB trên Windows 2000, tối thiểu 32 MB trên các hệ điều hành khác. Developer Edition: Tối thiểu 64 MB. Desktop Engine: Tối thiểu 64 MB trên Windows 2000, tối thiểu 32 MB trên hệ điều hành khác.
Hard disk	SQL Server database components: Tối thiểu 270 MB, thông thường 250 MB. Analysis Services: Tối thiểu 50 MB, thông thường 130 MB. English Query: 80 MB Desktop Engine: 44 MB
Monitor	VGA hoặc phân độ cao hơn. 800x600 hoặc phân độ cao hơn.

Thực hành cài đặt.

- Sử dụng đĩa CD ROM có bộ cài đặt SQL Server 2000 (tùy theo yêu cầu của bạn là Standard, Personal hay Enterprise,...)
- Chạy trình Autorun.exe (thông thường tự chạy khi đĩa vào máy tính)

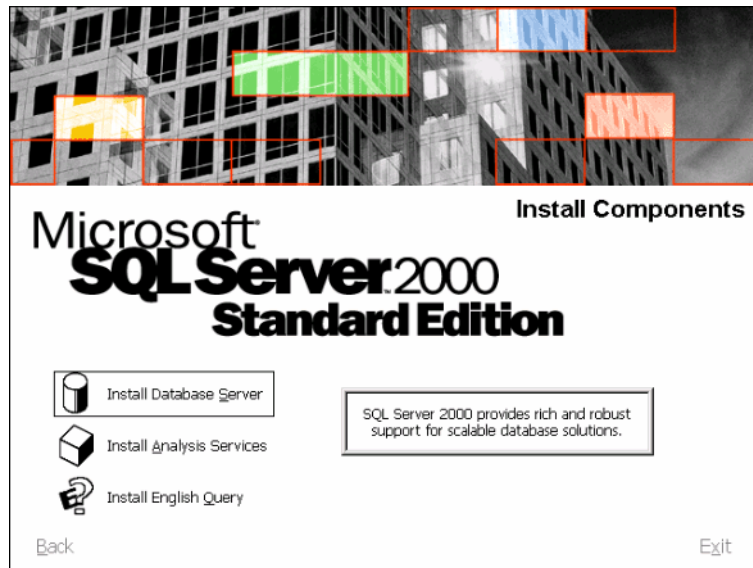


Trong màn hình trên ta có một số lựa chọn:

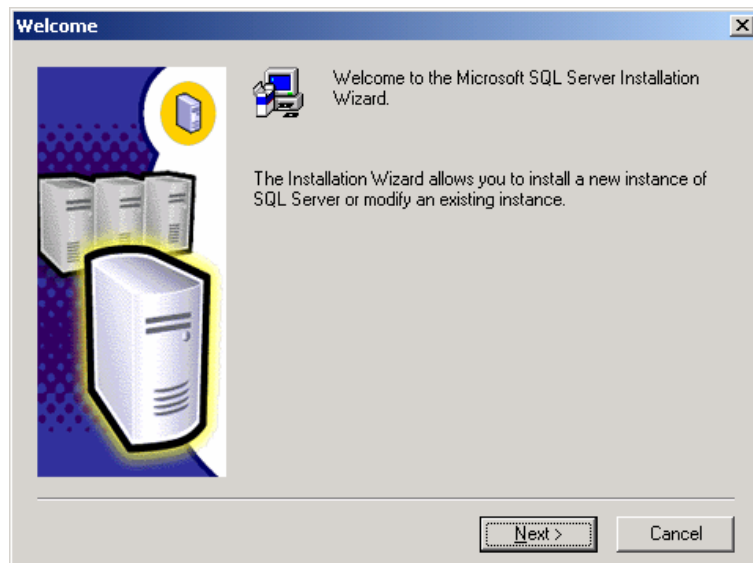
SQL Server Components: Thực hiện trong bước tiếp.

SQL Server 2000 Prerequisites: Dùng cài đặt những yêu cầu cần cung cấp sẵn cho việc cài đặt những thành phần trong máy cài đặt khác.

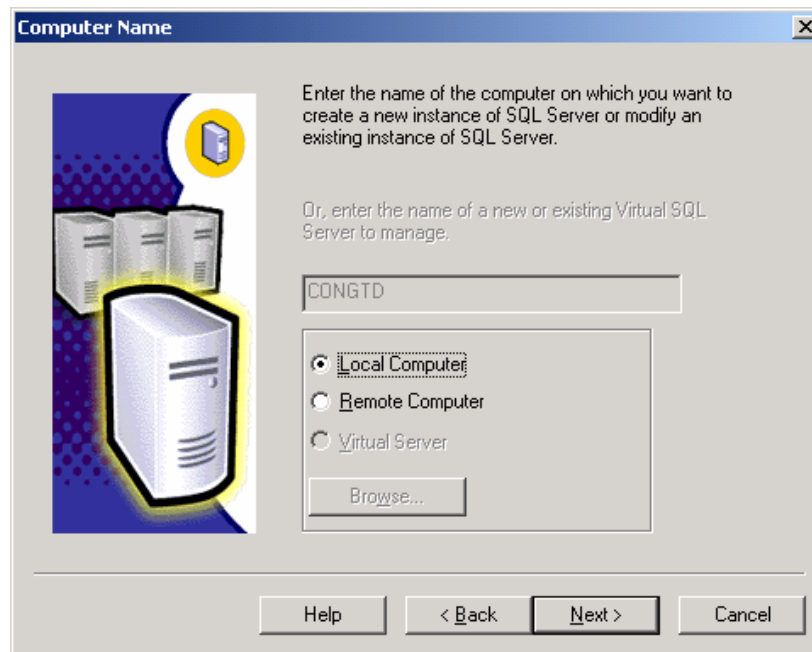
- Ch n SQL Server Components.



- Ch n Install Database Server.



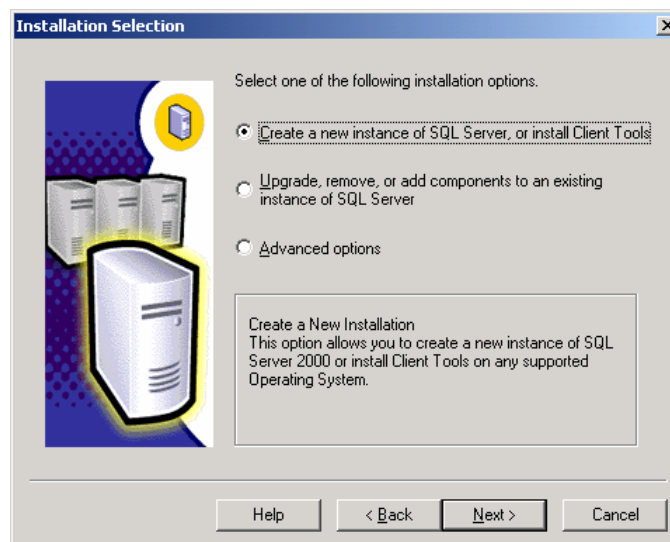
- Chọn Next.



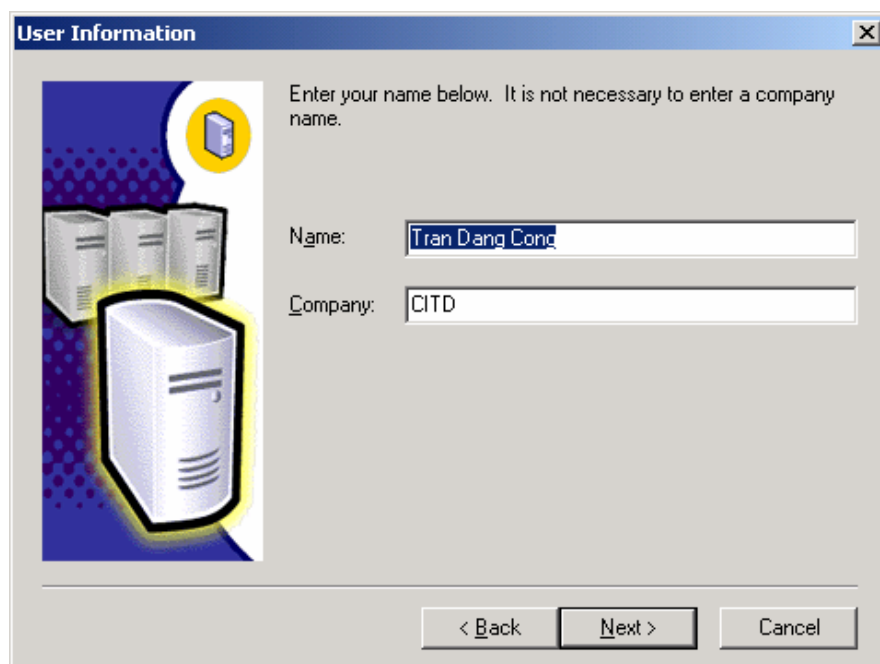
Nếu cài đặt SQL Server trên chính máy bản đang ngồi thì sẽ chọn Local Computer

Nếu cài đặt dùng kết nối với máy khác thì sẽ chọn Remote Computer sau đó nhập tên máy hoặc chọn vị trí máy bằng cách sẽ chọn Browse

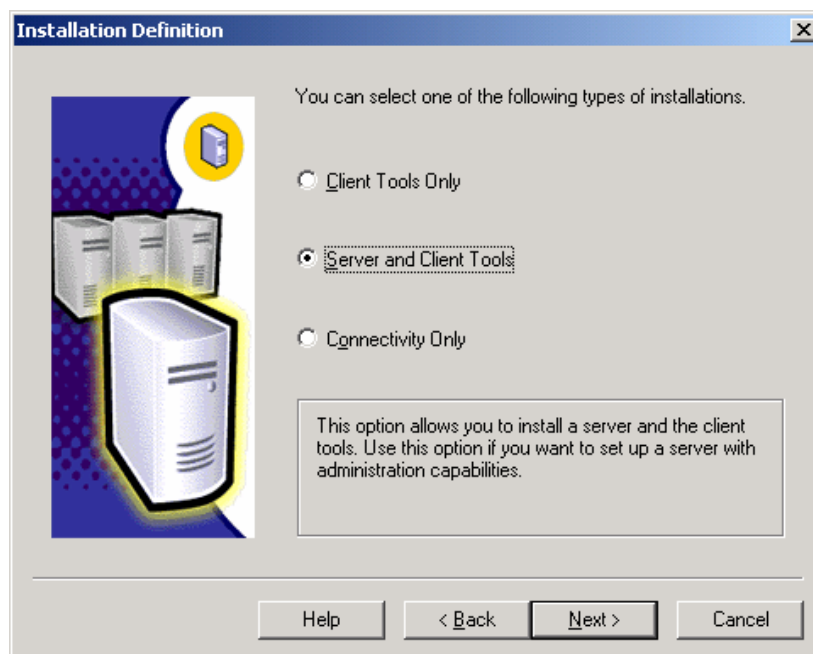
- Chọn next.



- Chọn tùy chọn theo hướng dẫn (tùy chọn, thay đổi cái đã có, thêm các chức năng khác,...).
- Trong trường hợp chọn tùy chọn (lựa chọn thứ nhất) sau đó chọn Next.



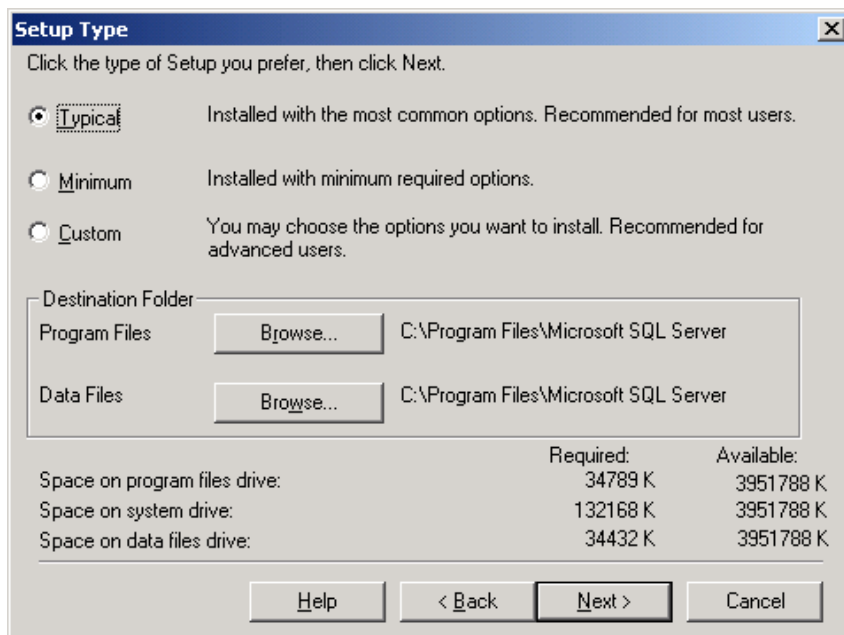
- Nhập tên cá nhân, tên công ty, sau đó chọn Next, Yes.



Trong các hình trên 3 lựa chọn:

1. Cài đặt các công cụ truy vấn: Sử dụng cho các máy khách không lưu trữ dữ liệu nhưng có chức năng truy vấn dữ liệu trên SQL Server có CSDL
2. Cài đặt Server và các công cụ truy vấn: Cài đặt SQL Server có dữ liệu và các công cụ của máy khác truy vấn dữ liệu
3. Cài đặt kết nối: Dùng cho các máy chủ sử dụng kết nối trên Server, thường dùng cài đặt cho các máy sử dụng các ứng dụng kết nối trên server

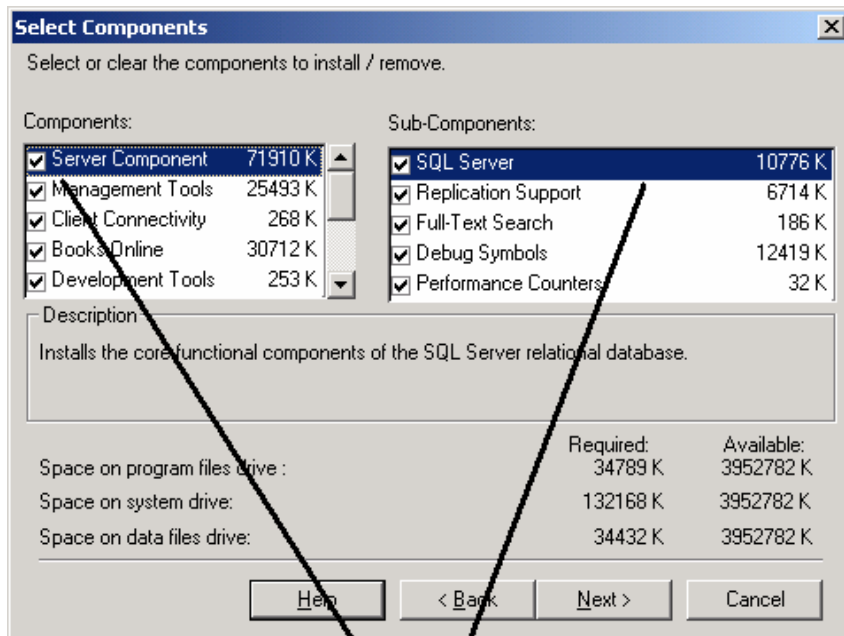
- Chọn lựa chọn 2, sau đó Next



Dùng các lựa chọn như sau để:

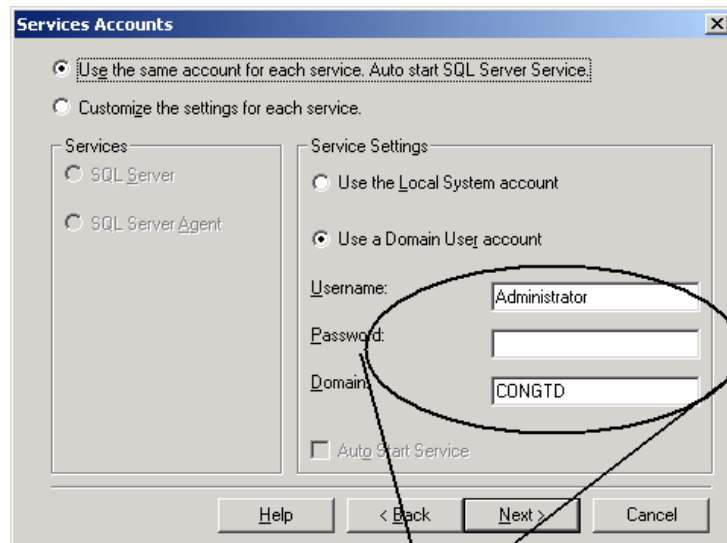
- + *Typical*: Cài đặt những chức năng cơ bản để hệ thống vận hành (chức năng thông thường).
- + *Minimum*: Cài đặt những chức năng tối thiểu của hệ thống.
- + *Custom*: Lựa chọn những chức năng cần cài đặt theo yêu cầu của người dùng.

Trong cách 1 a ch n Custom ta c n thêm b c ch n các ch c n ng nh sau:



Lựa chọn các chức năng

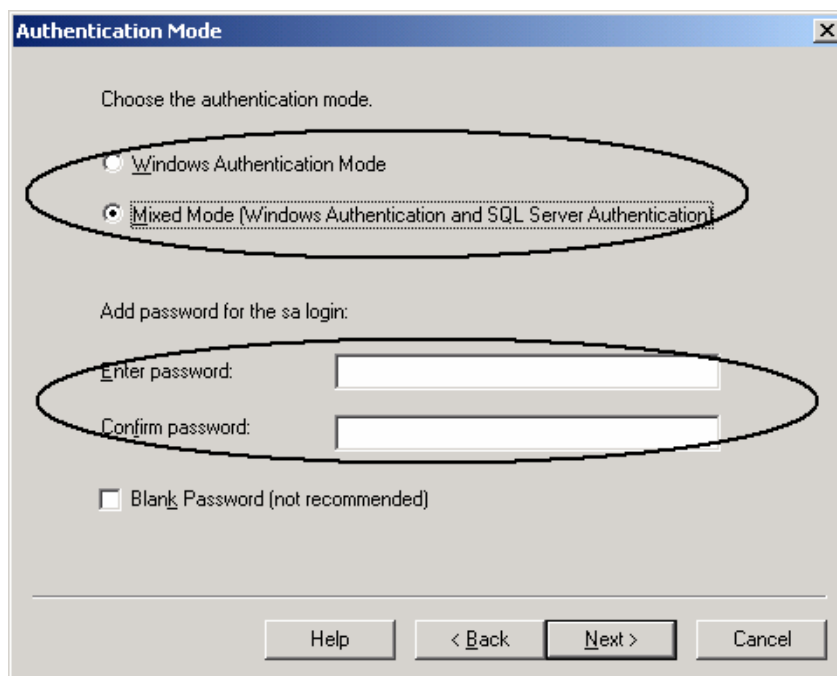
- n nút Next ti p t c.



Nhập tên và mật khẩu vùng

Trong các bước trên ta cần nhập tên, mật khẩu của user cũng ký truy nhập vùng, thông tin SQL Server cài đặt sẽ thể hiện theo quyền Administrator của máy tính chủ, khi đó bạn lựa chọn nút chọn Use a Domain User account.

- Chọn nút next tiếp tục.



Trong các bước trên cho phép ta sử dụng 2 lựa chọn:

+ *Lựa chọn thứ nhất*: Người dùng sử dụng hình thức bảo mật của Windows (hệ điều hành của máy chủ cài đặt – thông tin khi cài đặt dùng lựa chọn này).

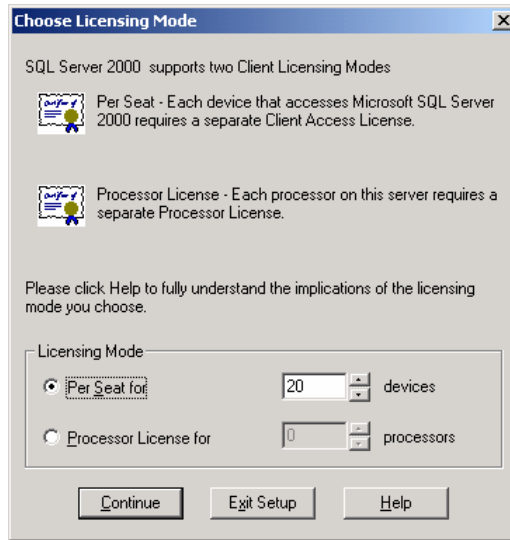
+ *Lựa chọn thứ hai*: Người dùng sử dụng hình thức bảo mật của Windows và các quyền truy cập CSDL SQL Server.

Trong các trường hợp trên đều có thể sử dụng tên và mật khẩu cũng cung cấp theo vùng (domain) của hệ điều hành. Nếu sử dụng lựa chọn thứ 2 ta sử dụng tên và mật khẩu của người quản trị vùng (Administrator).

Đối với SQL Server ta có thể thay tên Administrator bằng tên sa (viết tắt của System Administrator).

Vấn đề thực hiện chọn bảo mật nào sẽ được bàn trong những bài sau.

- in next tip t c.

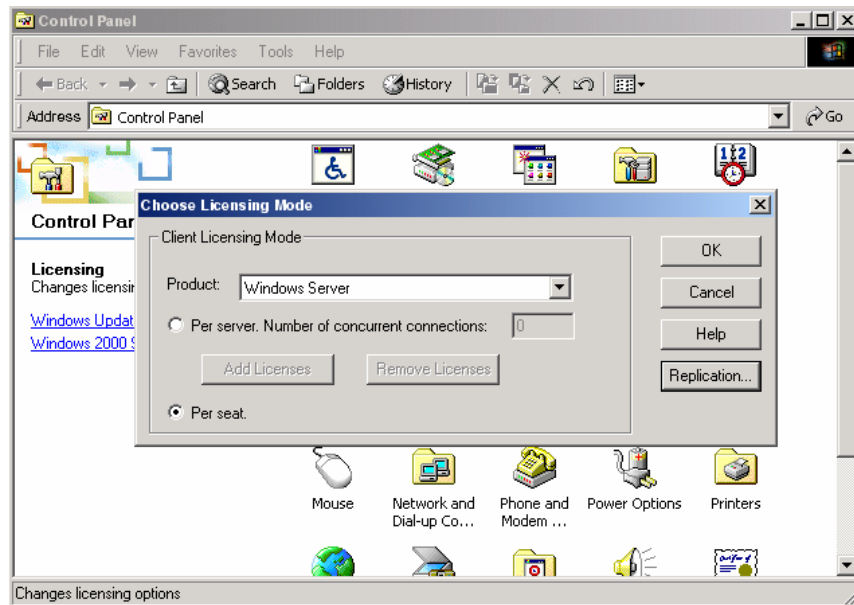


Trong c a s trên ta có 2 l a ch n:

+ *Per Seat for*: L a ch n cho phép xác nh s thi t b (khái ni m s c gi i thi u sau) trên m i v trí khai thác h th ng theo b n quy n c phép c a Microsoft.

+ *Processor License for*: Xác nh s Processor cho phép s d ng theo b n quy n c cung c p b i Microsoft.

Ngoài vì c ng k b n quy n t i th i i m này, ta có th ng ký b n quy n trong công c i u khi n c a Control Panel.



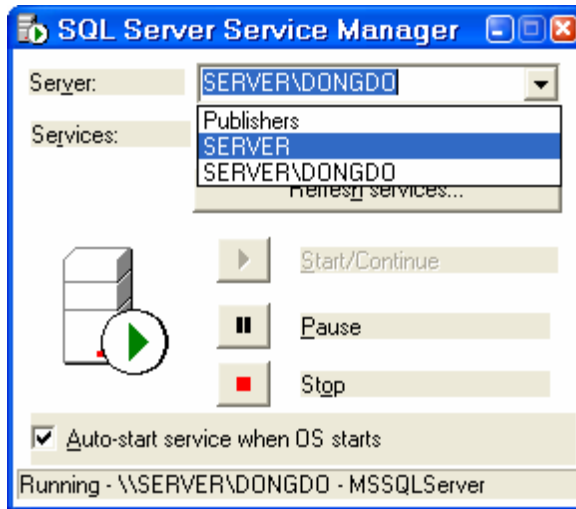
QUẢN TRỊ SERVER

INSTANCE

SQL Server hỗ trợ nhiều hoạt động trên mạng, như các mô hình đã xem xét trước ta có thể thiết lập nhiều máy tính cài đặt SQL Server, các máy tính có thể liên kết với nhau, trao đổi dữ liệu với nhau.

Tuy nhiên một máy tính cũng có thể thiết lập nhiều hệ thống SQL Server khác nhau, mỗi hệ thống có một tên quy định, mỗi hệ thống này ví dụ gọi là một Instance.

Một Instance trên một máy tính được coi như một hệ thống SQL Server độc lập, từng một hệ thống SQL Server cài đặt trên các máy tính khác nhau.



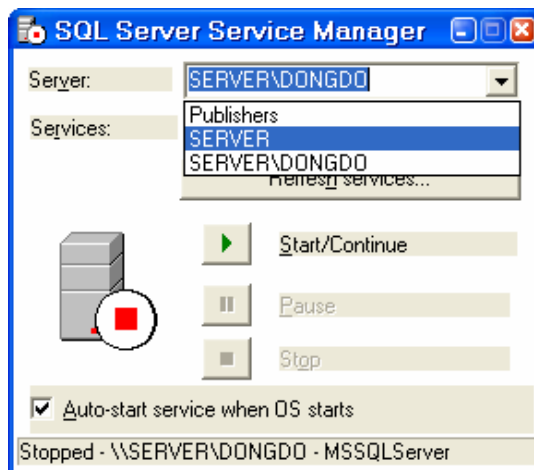
LIÊN KẾT CÁC DỊCH VỤ CỦA SQL SERVER.

SQL Server sau khi cài đặt xong, khi khởi động máy thông thường sẽ có biểu tượng ở góc dưới, trái màn hình như sau:



Biểu tượng SQL Server

Bi u t ng máy ch có v i máy tính cài t phiên b n SQL Server và là bi u t ng c a trình qu n lý d ch v Service Manager.



G m các d ch v c b n sau:

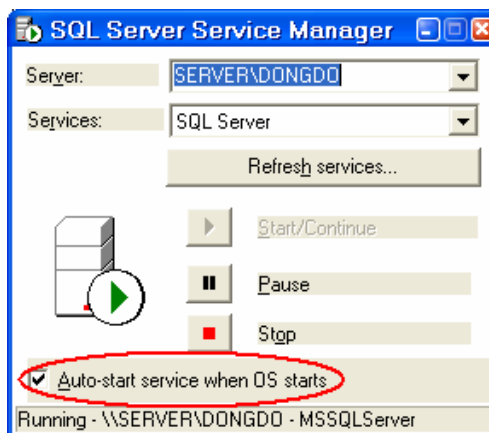
- + Distributed Transaction Coordinator - DTC.
- + Microsoft Search.
- + SQL Server.
- + SQL Server Agent.

Các d ch v này ta có th b t u, t m d ng ho c k t thúc, m i d ch v u i u khi n các ng d ng, công c qu n tr c a SQL Server.

th c hi n i u khi n d ch v u tiên ta làm nh sau:

Services -> Start/Continue (Pause, Stop)

d ch v kh i ng t ng khi kh i ng h i u hành hã ch n vào nút ch n ***Auto-start service when OS starts.***

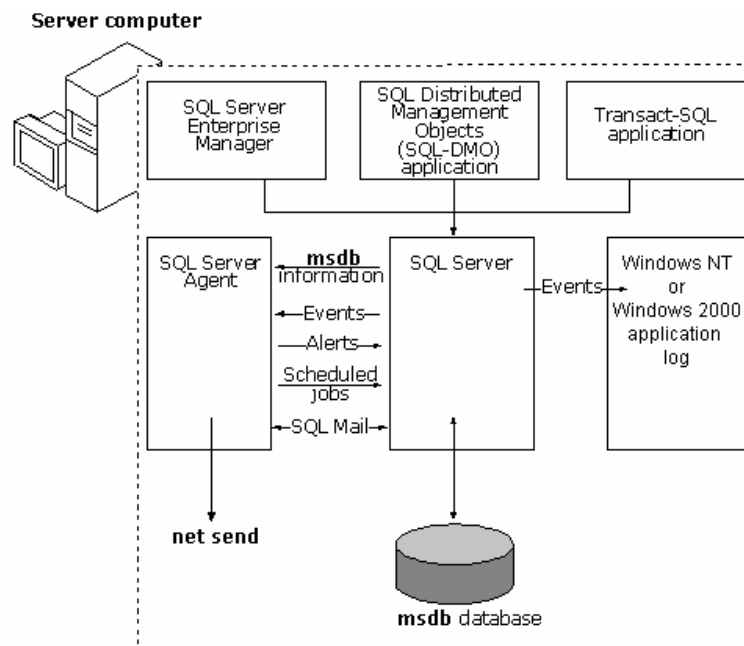


Dịch vụ MS SQLServer.

Dùng quản lý tất cả các file gồm các CSDL mà SQL Server quản lý, là thành phần xử lý tất cả các lệnh của Transact-SQL cũng như các trình độ client, phân phối các nguồn tài nguyên khi có nhiều user cùng truy cập một lúc. Đây là dịch vụ quản trị cơ bản, khi ngừng dịch vụ này hệ thống sẽ ngừng tất cả các công việc khai thác dữ liệu.

Dịch vụ SQLServerAgent.

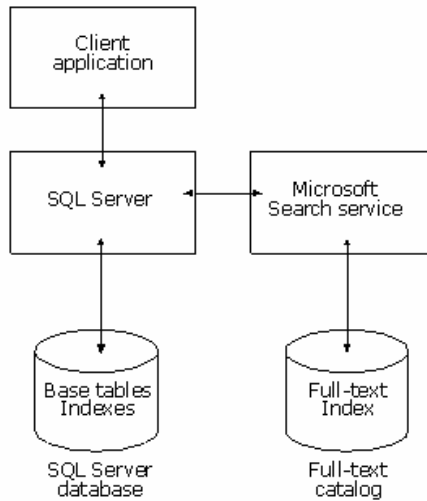
Hỗ trợ các tính năng cho phép lập thời gian biểu cho các hoạt động theo từng giai đoạn trên SQL Server, hoặc thông báo cho người quản lý hệ thống về những sự cố của hệ thống, bao gồm các thành phần Jobs, Alerts, Operator.



Dịch vụ Microsoft Search.

Cung cấp dịch vụ tìm kiếm và tìm kiếm văn bản với các phép toán cơ bản sau:

- + Ký tự (chủ yếu): =, >, >=, <, <= để so sánh với một chuỗi hình.
- + So sánh chuỗi trong văn bản hoặc chuỗi có kích thước lớn, văn bản.

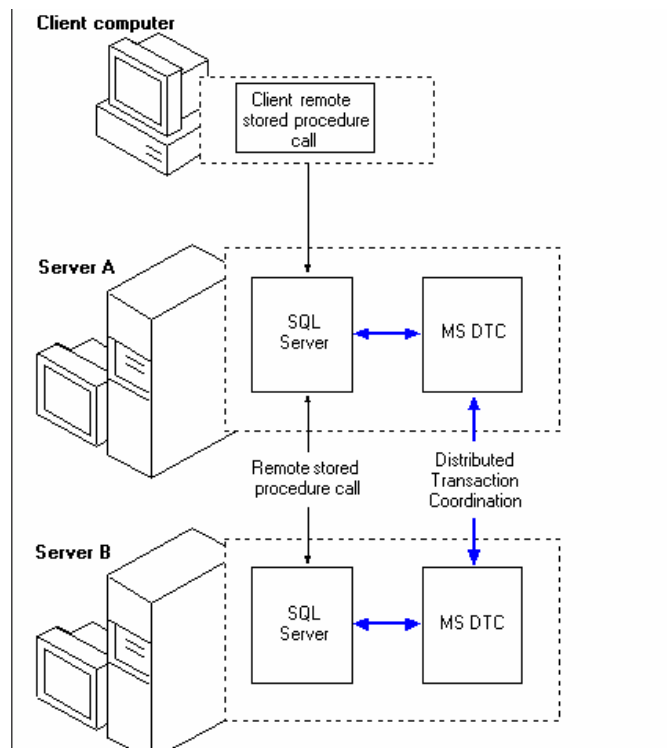


Dịch vụ MS DTC.

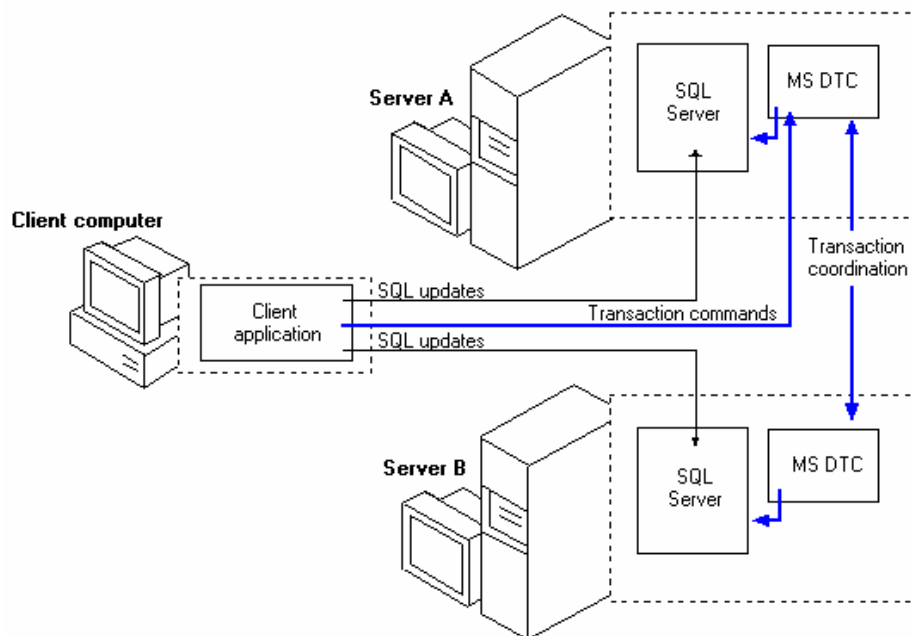
Là dịch vụ cho phép trong môi trường phiên giao dịch có thể sử dụng dữ liệu được phân phối trên nhiều server khác nhau, thể hiện theo các bước cơ bản sau:

- + Gửi các thủ tục lưu trữ trên các server xa sử dụng SQL Server
- + Tạo ứng dụng hoặc tạo các phiên giao dịch cục bộ và các giao dịch với các máy chủ xa
- + Tạo bộ dữ liệu cục bộ nhúng hoặc phân phối ở các server xa.

Xem xét sơ đồ như sau:



Như vậy trên khi client truy cập thì tất cả sẽ nằm trên server cục bộ, khi có yêu cầu dữ liệu trên server khác, thông qua dịch vụ MS DTC server cục bộ sẽ truy cập các thành phần server khác, kết quả có thể tạo ra các dữ liệu tập trung tại nhiều server khác nhau.



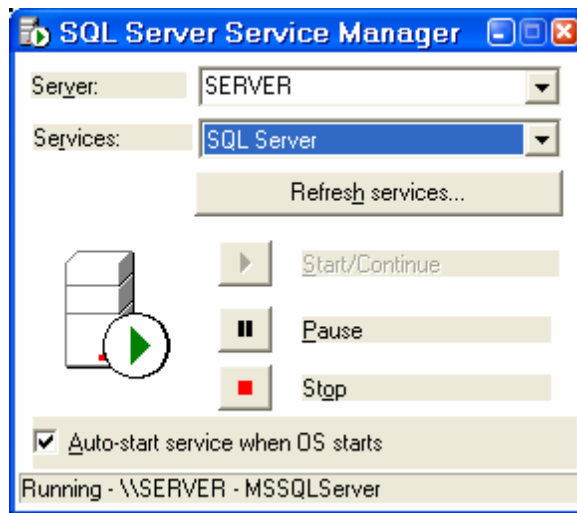
QUẢN TRỊ SERVER.

Như đã nêu trên mỗi Instance sẽ coi là một hệ thống quản trị CSDL SQL Server và có thể gọi tắt là Server. Server có chức năng quản trị toàn bộ hệ thống của SQL Server (dữ liệu, bảo mật, ngừng/dừng, tác vụ, các dịch vụ khác,...).

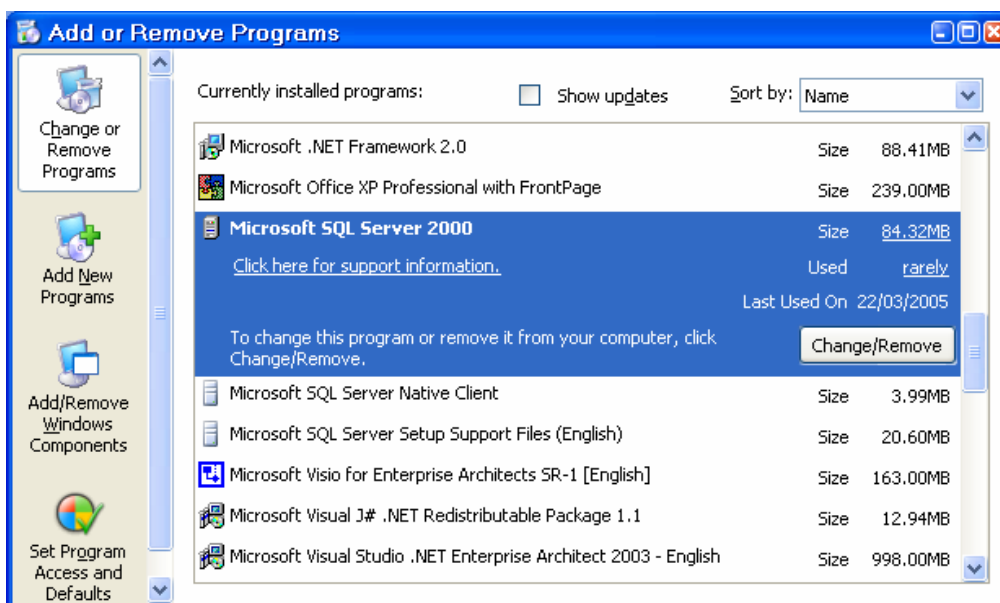
Các ứng dụng hoặc các công cụ khai thác dữ liệu (GUI) sẽ thể hiện khai thác dữ liệu do Server quản lý khi có kết nối đến Server. Tuy nhiên nhiệm vụ quản trị là Server có thể sử dụng phần mềm.

Thích hiển thị quản trị Server là vì thích hiển thị các công việc sau:

+ Bắt đầu/ tạm dừng dịch vụ của SQL Server.



+ Giảm bộ cài đặt Server (Instance).



+ Thay đổi, nâng cấp phiên bản.

THIẾT LẬP KẾT NỐI SERVER.

khởi tạo các dữ liệu của hệ thống SQL Server ta phải thiết lập kết nối (connect) đến Server, vì kết nối có thể thiết lập các vị trí: ứng dụng, công cụ khởi tạo của SQL Server là SQL Client. Trong phần này ta sẽ xem xét về kết nối từ SQL Client đến Server.

M i Server khi cài t ã có m t tên là tên c a Instance c t. Trên m t m ng máy tính n u có quy n h n ta hoàn toàn có th th c hi n k t n i n Server nói trên.

T m t máy SQL Client có th th c hi n ng th i k t n i n nhi u Server khác nhau, ây c ng chính là u i m c a SQL Server.

N u b n cài t phiên b n SQL Server trên máy tính b n c ng ph i làm ti đ n b các b c k t n i nh SQL Client, phiên b n SQL Server c coi nh g m 2 ph n: H th ng qu n tr , công c khai thác SQL Client.

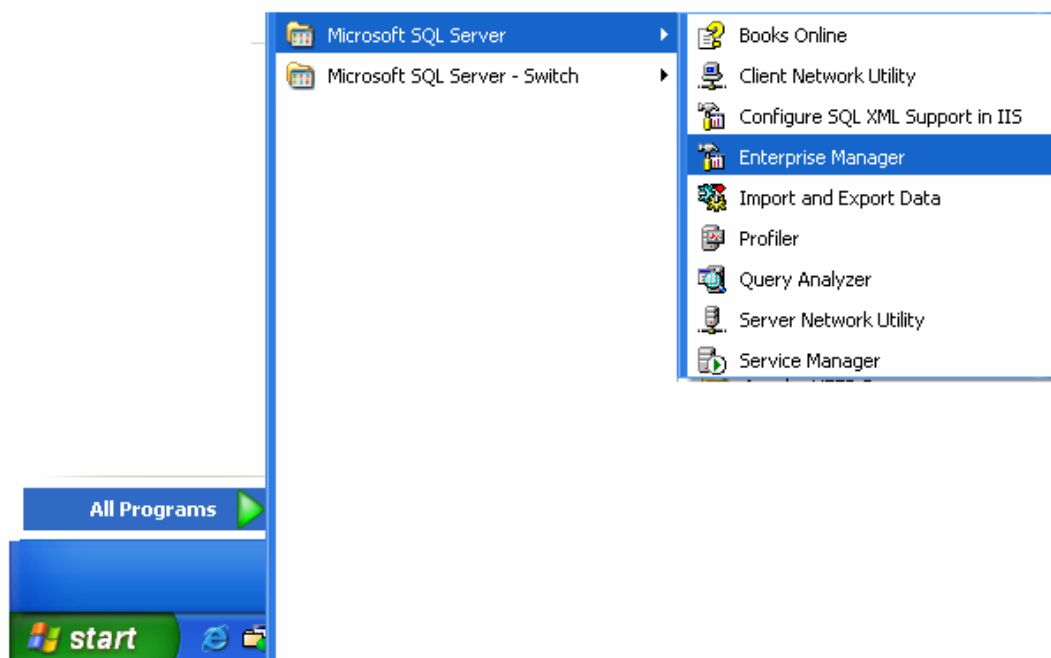
Dù s d ng công c nào khai thác ã c cài t trên máy tính c a b n, quy n h n khai thác, qu n tr ph thu c vào user th c hi n k t n i.

Qu n tr Server Group.

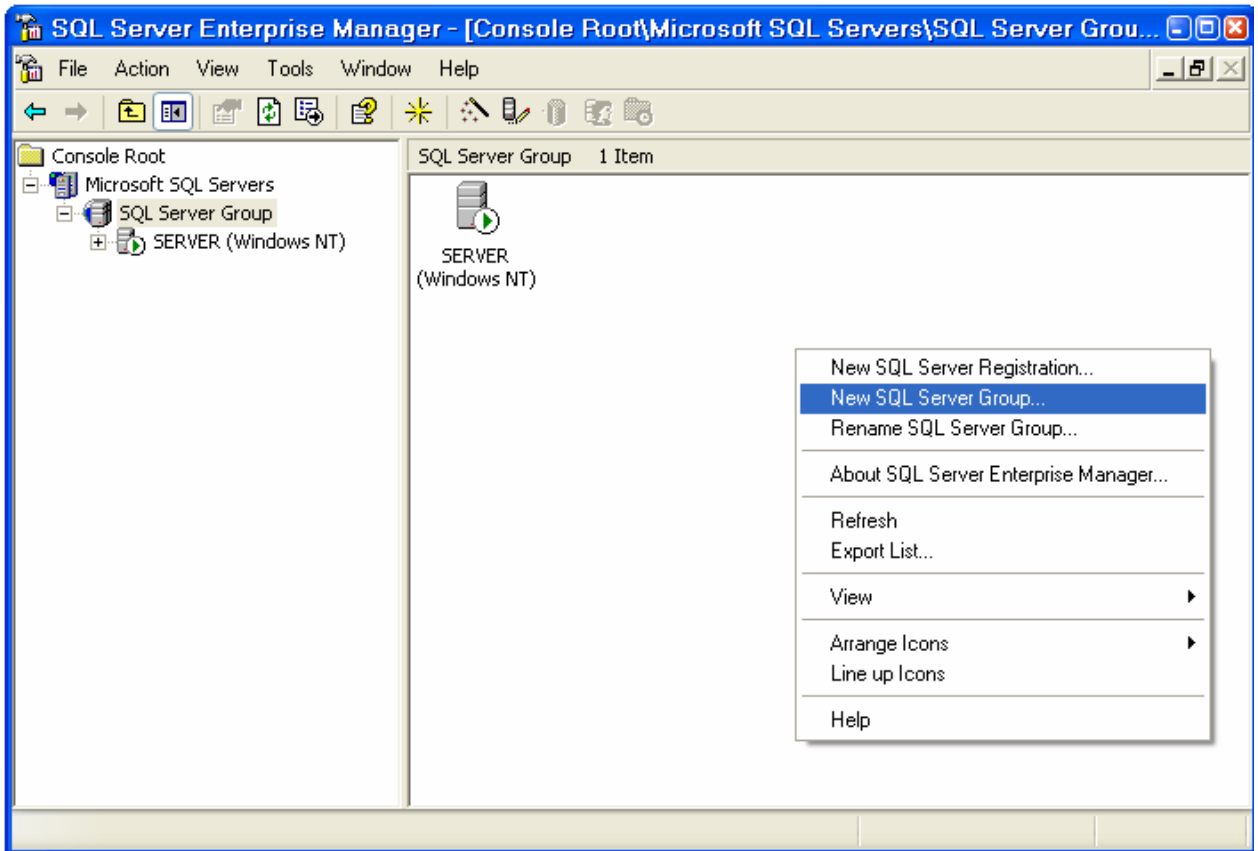
Server Group là công c dùng qu n lý các k t n i (s th c hành sau) t ng t nh khái ni m th m c trong h i u hành, trong các Server Group ch a các Server Group con ho c các k t n i n Server.

Các b c th c hi n nh sau:

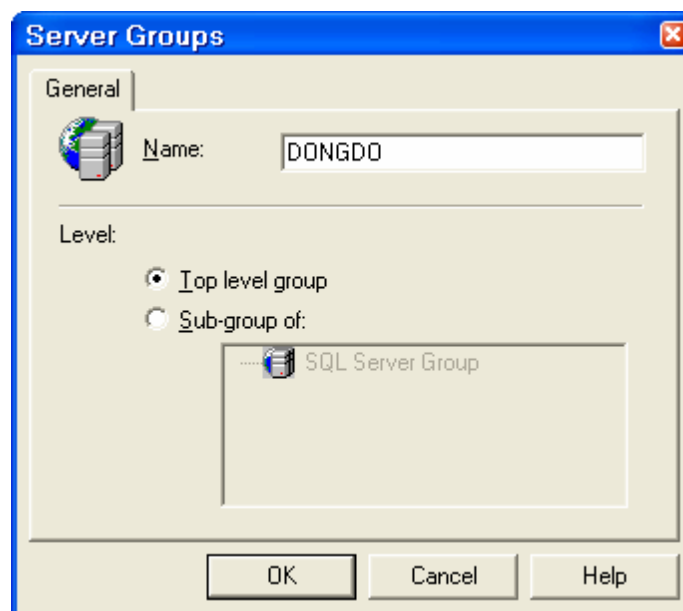
- Vào ch c n ng Enterprise manager nh hình d i



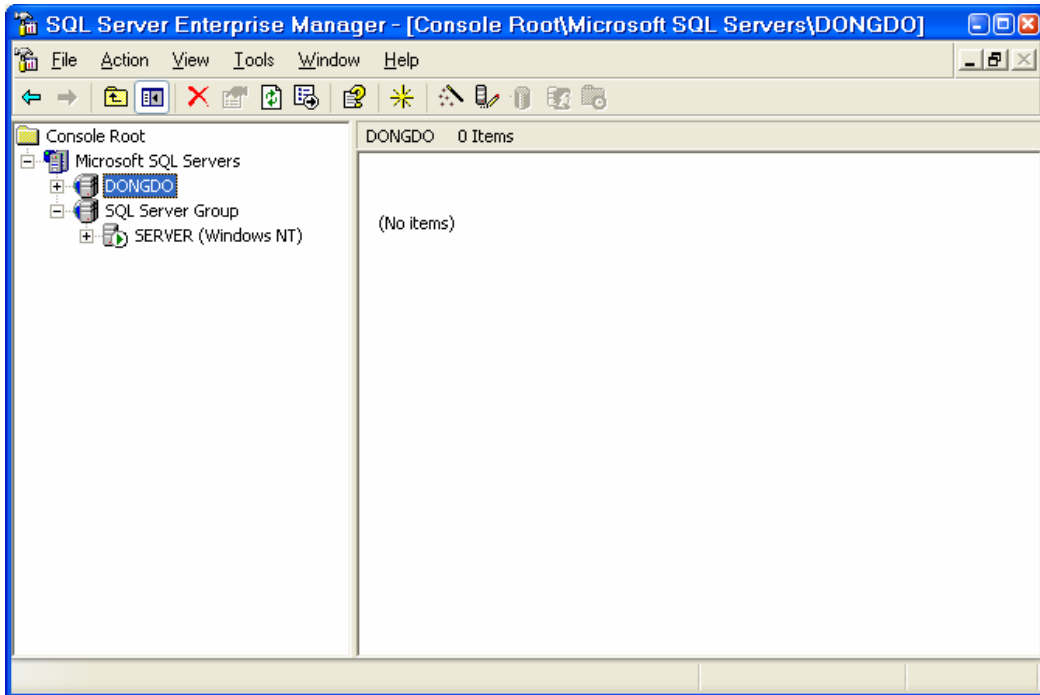
- Di chuyển vào menu trong bảng cách nhúng vào đầu + các cây các nút.



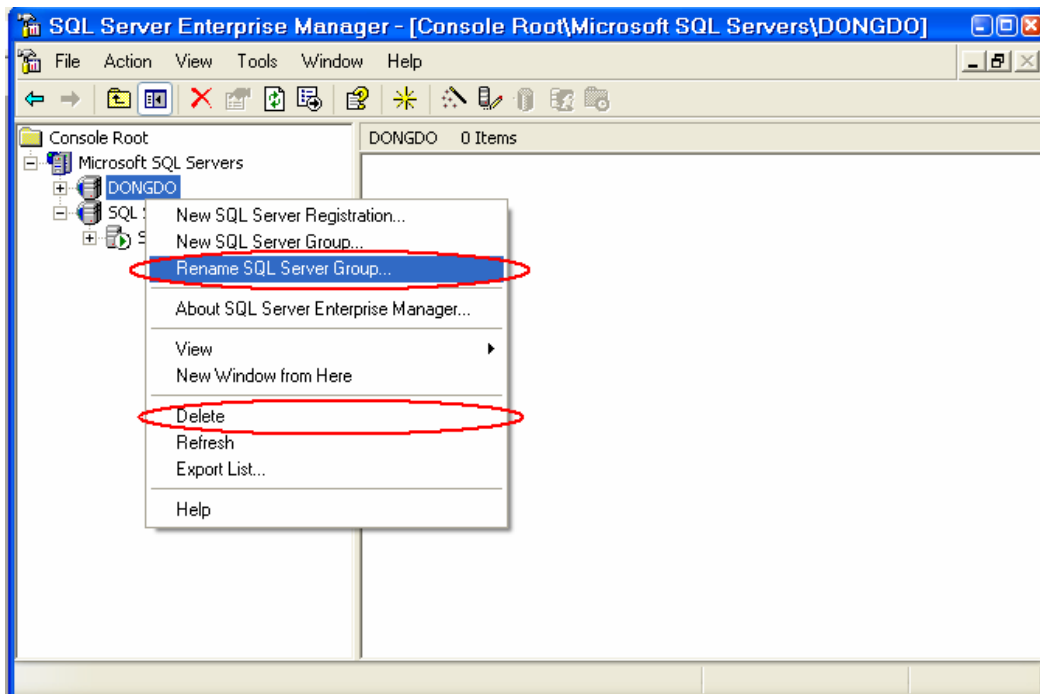
- Chọn New SQL Server group từ menu



- Nhập tên group -> Ok



Các thao tác đổi tên, xóa có thể thực hiện bằng cách nhấp chuột vào group cần thao tác.



Thi t l p k t n i n Server (thi t l p Server).

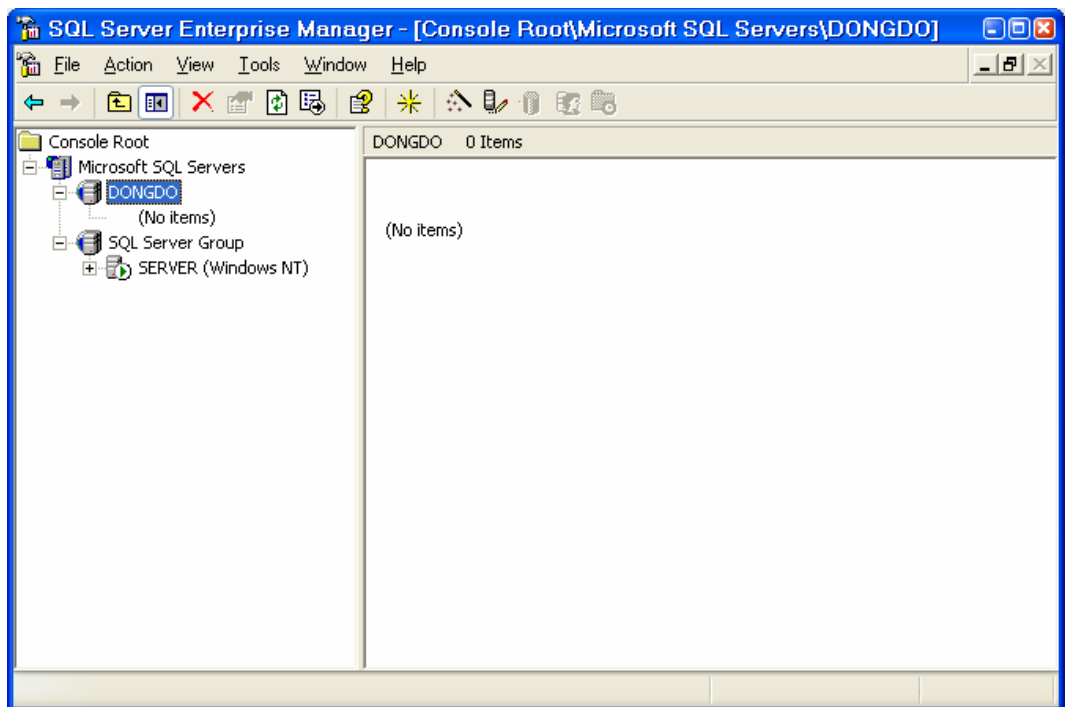
Là b c thi t l p k t n i n Server t Client, Server khác. Các k t n i c th hi n b ng tên c a Server k t n i n (hay còn là tên c a Instance), chính vì v y nên tên các k t n i trên m t Client là duy nh t, không trùng nhau trong toàn b client.

Tr c khi th c hi n t o k t n i ta ph i chu n b các tham s sau:

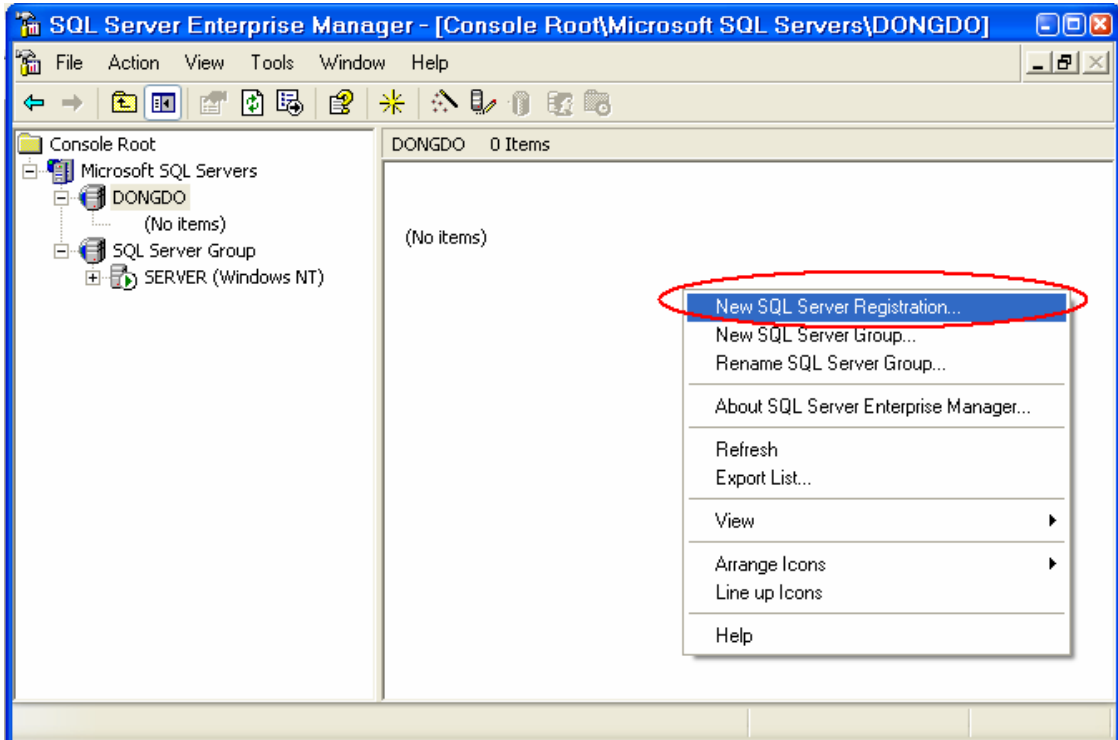
- + Tên Server (Instance) mu n k t n i n
- + User name và Password c a Server ta c n k t n i n (tham s này do ng i qu n tr Server c p).

Cách làm nh sau:

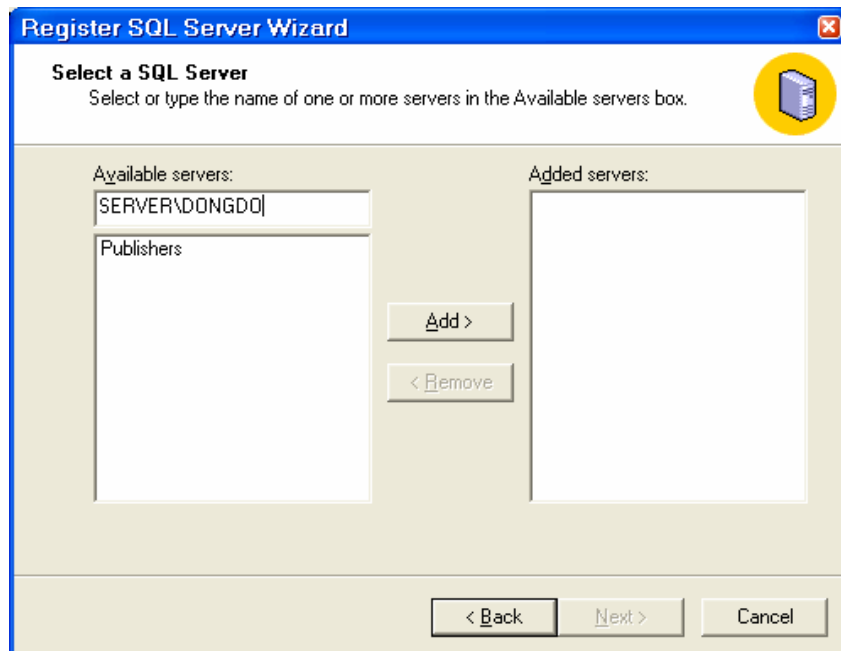
- Vào Enterprise và ch n Server group



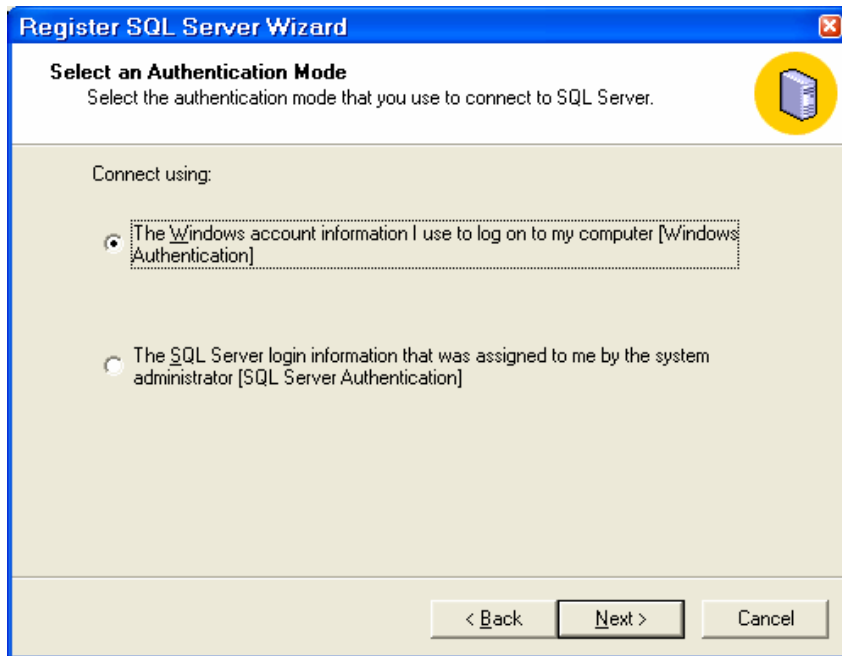
- Nh n nút ph i chu t vào c a s bên ph i, ch n New SQL Server Registration.



- Nh p tên Server.



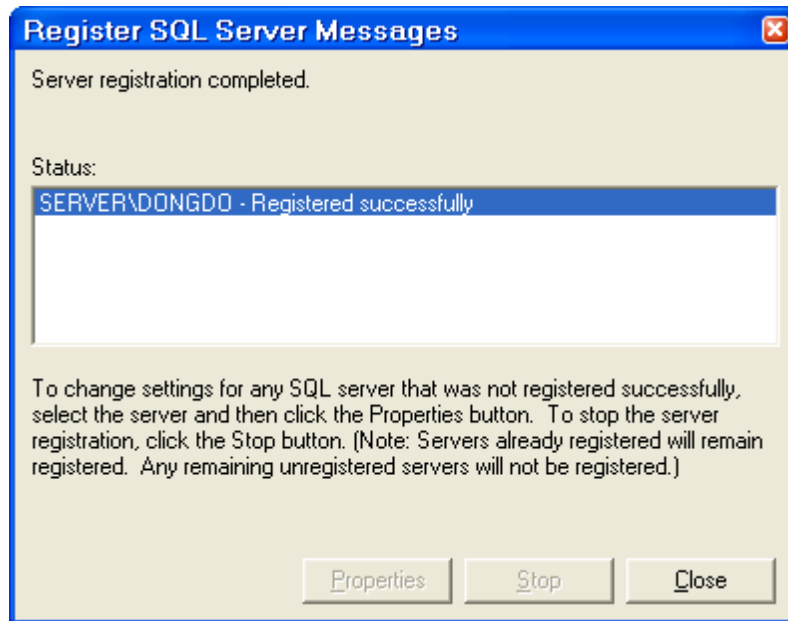
- Nh n Add -> Next



- Chọn chế độ bảo mật (thông tin đăng nhập The Windows account information chọn bảo mật của Windows, phần này sẽ xem xét kỹ trong bài sau) -> Next -> Chọn Server Group.



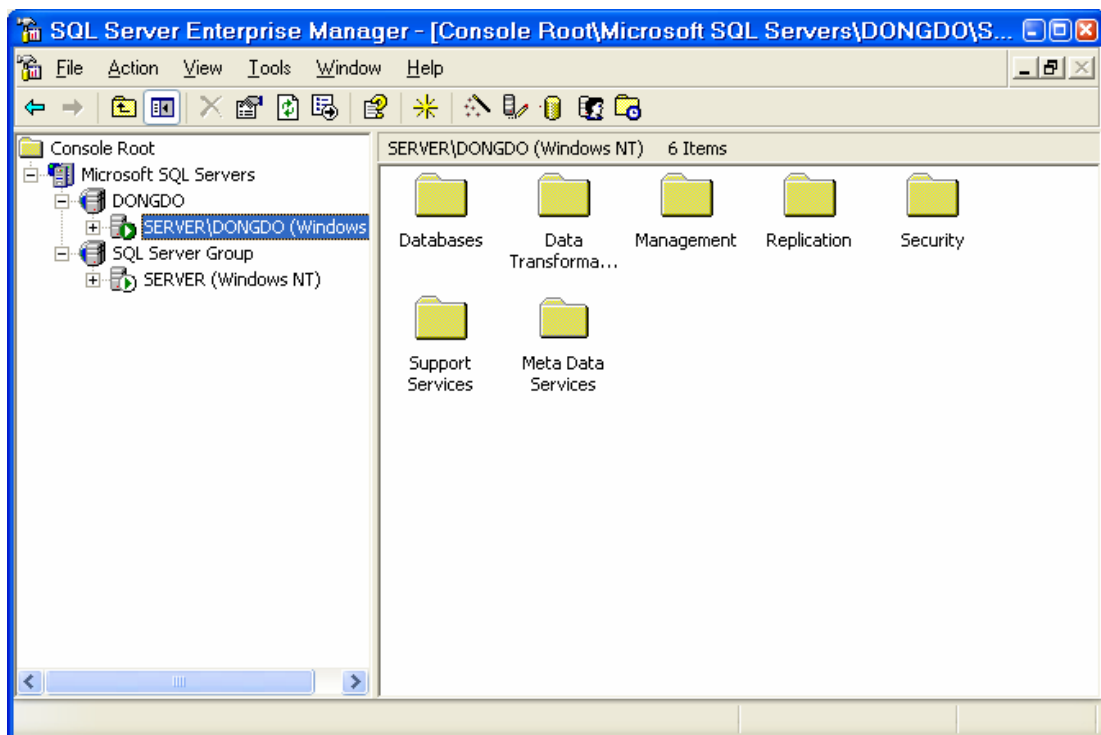
- Nhấn Finish.



Khi màn hình xuất hiện thông báo Registered successfully là vì cài đặt đã thành công.

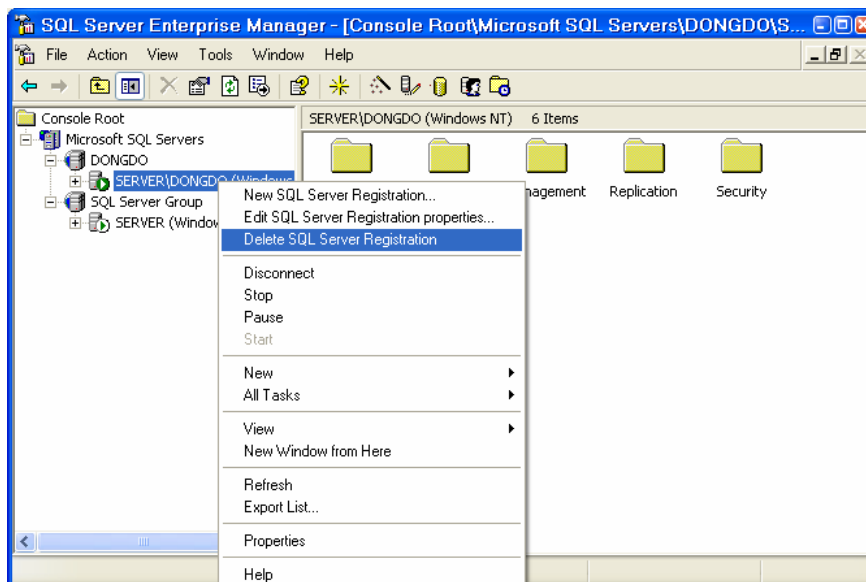
- Nhấn Close.

Sau khi cài đặt xong kết nối xuất hiện trên danh sách các kết nối.



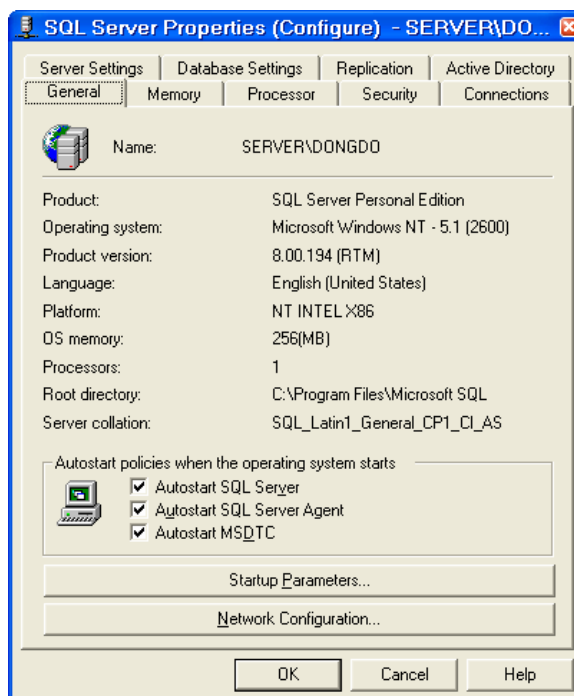
K t n i nh ã thi t l p có quy n h n khai thác ph thu c vào user k t n i, trong ví d trên quy n h n ph thu c vào user ã truy nh p vào Windows, tuy nhiên trong nh ng bài sau s gi i thi u cách th ac t o user, s d ng user c a SQL Server th c hi n t o k t n i và khai thác.

xóa ho c s a thông tin cho k t n i ch n Delete ho c Edit khi nh n nút ph i chu t vào tên k t n i c n thao tác.



Xem và thay i tham s cho Server.

thay i tham s cho Server, hãy ch n tên k t n i -> nh n nút ph i chu t -> ch n Properties.

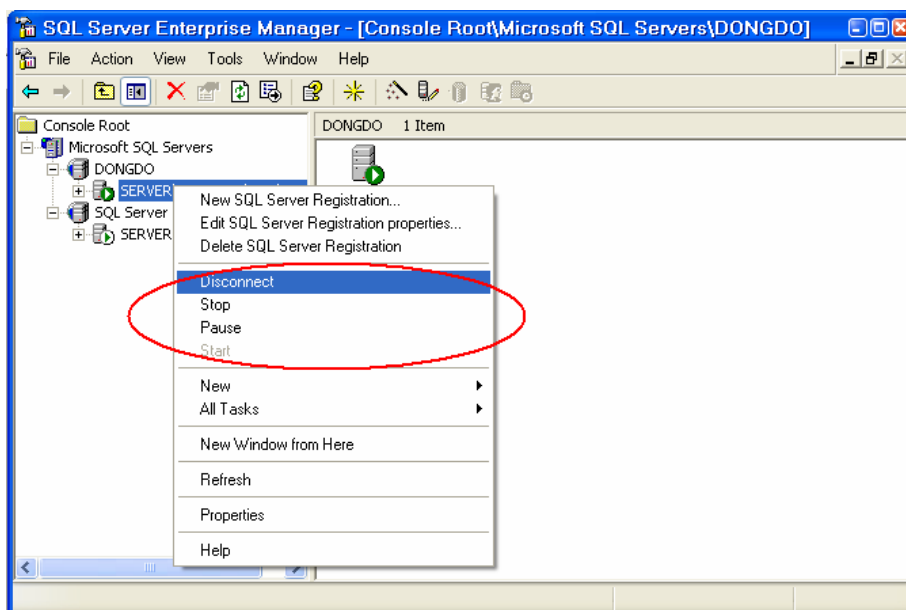


Tuy nhiên các tham số trên có thể bạn chưa xem xét, nên trong bài này chỉ giới thiệu mã số xem xét một số tham số cơ bản trong những bài liên quan.

Bắt đầu cài đặt và khởi động SQL Server.

Ta có thể thực hiện một số việc khi cài đặt MS SQL Server như sau.

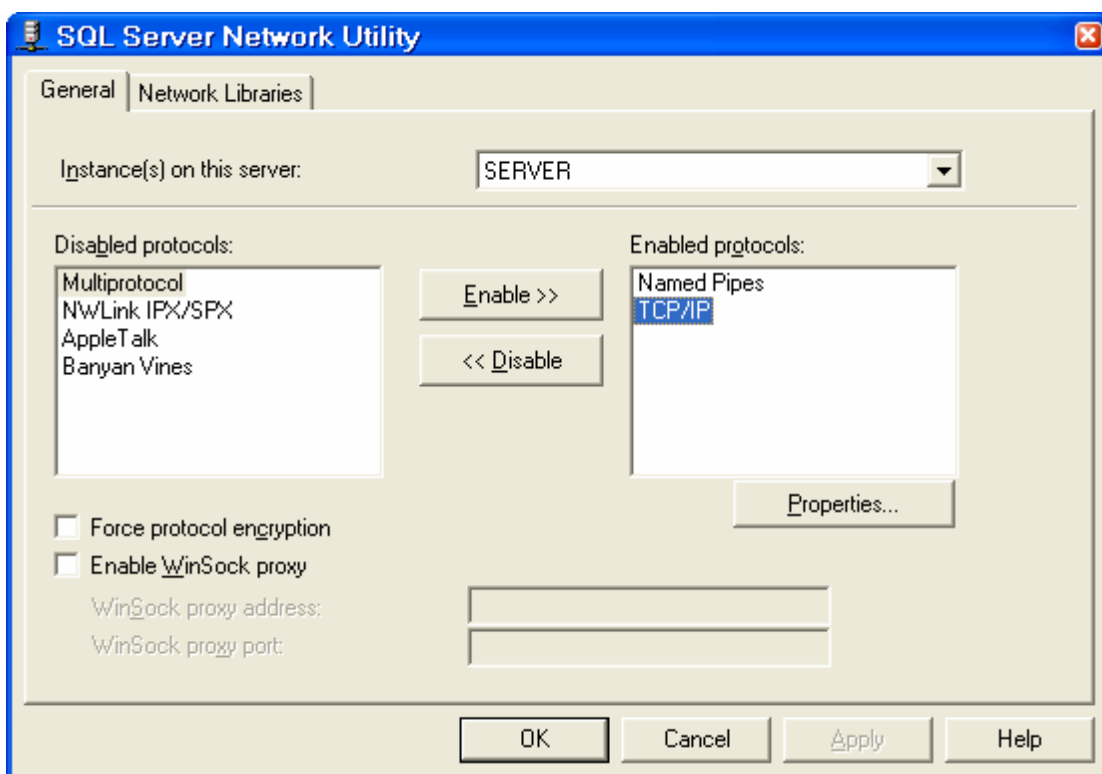
- Chọn tên kết nối
- Nhấn nút phù hợp



C U HÌNH K T N I M NG C A SERVER.

các Client hoặc các ứng dụng kết nối đến Server, ta phải cấu hình các phương thức kết nối phù hợp với kết nối mạng. Kết nối mạng có thể sử dụng kết nối thông qua Proxy, thông qua mạng Internet.

Khi sử dụng kết nối nào thì trình cấu hình giao thức phù hợp với giao thức mạng sử dụng. Trình cấu hình cách chọn Server network utility



- Chọn giao thức vào danh sách enabled sử dụng và bỏ danh sách Disabled không sử dụng.
- Chọn Properties để cấu hình, tham số của giao thức.
- Chọn Enable WinSock proxy để kết nối qua Proxy.
- Chọn Force protocol encryption sử dụng kết nối qua Internet không dùng Fire Wall với SQL Server.

QUẢN TRỊ CÁC CLIENT.

Khi Server đã sẵn sàng cho kết nối, việc tiếp theo là xem xét đến các client kết nối đến server. Trong phần này ta sẽ xem xét cấu hình client kết nối đến server.

Các client kết nối đến server được thể hiện trên hệ thống truy cập mạng máy tính, tuy nhiên các ứng dụng client kết nối đến server được thể hiện khai thác dữ liệu trên server thông qua một số phương thức kết nối sau:

- OLE DB: Có 2 kiểu Microsoft OLE DB Provider for SQL Server và Microsoft OLE DB Provider for ODBC.

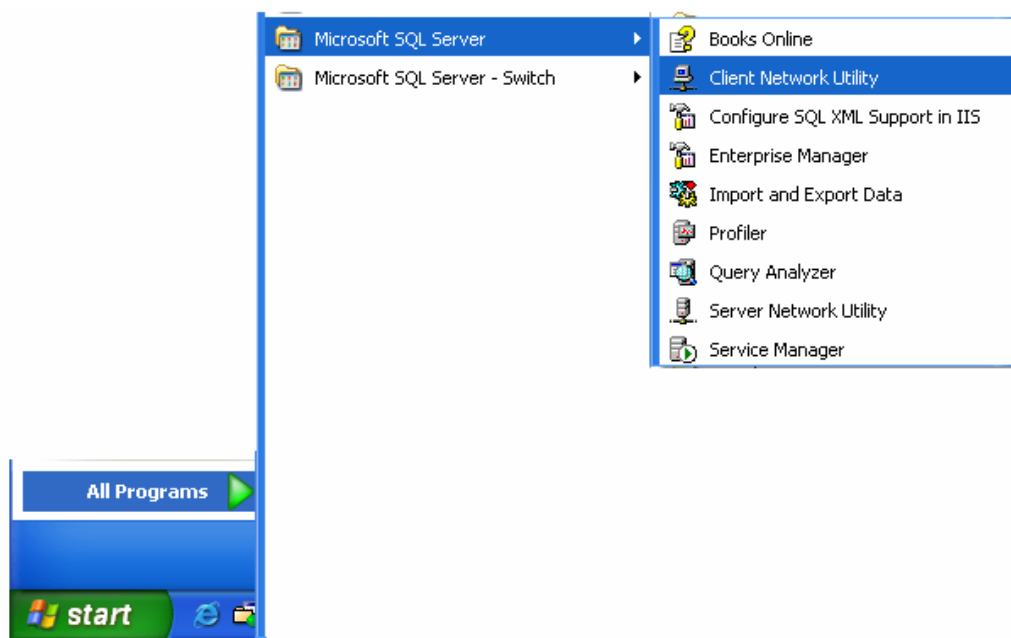
- ODBC: Kết nối thông qua SQL Server Enterprise Manager và SQL Query Analyzer sử dụng SQL Server ODBC.

- DB-Library: Sử dụng lệnh SQL Server **isql**.

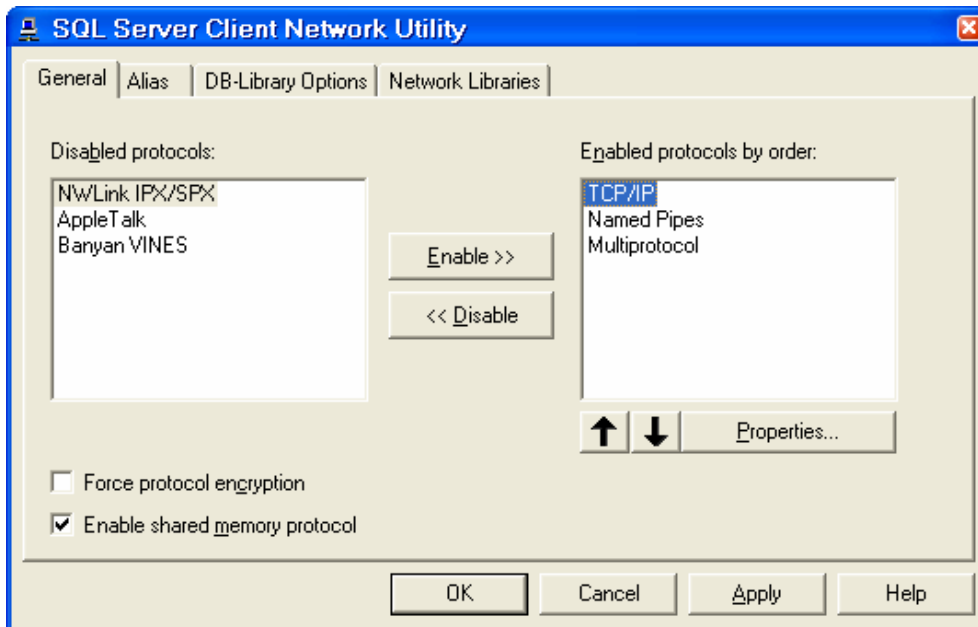
Cấu hình Net-Library.

Như đã xem xét trước khi Instance khi cấu hình xác định máy chủ và sử dụng riêng, nên việc kết nối thông qua Net-Library là kết nối thông qua máy chủ và máy khách đã xác định các Instance.

Trên Server thông thường cấu hình theo TCP/IP Sockets và Named Pipes Net-Libraries, trên client thông thường cấu hình theo Thước kẻ cấu hình ta sử dụng Client network utility.



- Chọn Client network utility.



- Chọn giao thức và các tham số liên quan tới ứng dụng của server, có thể thay đổi hiển thị tên Server sang tên mới trên bảng Alias.

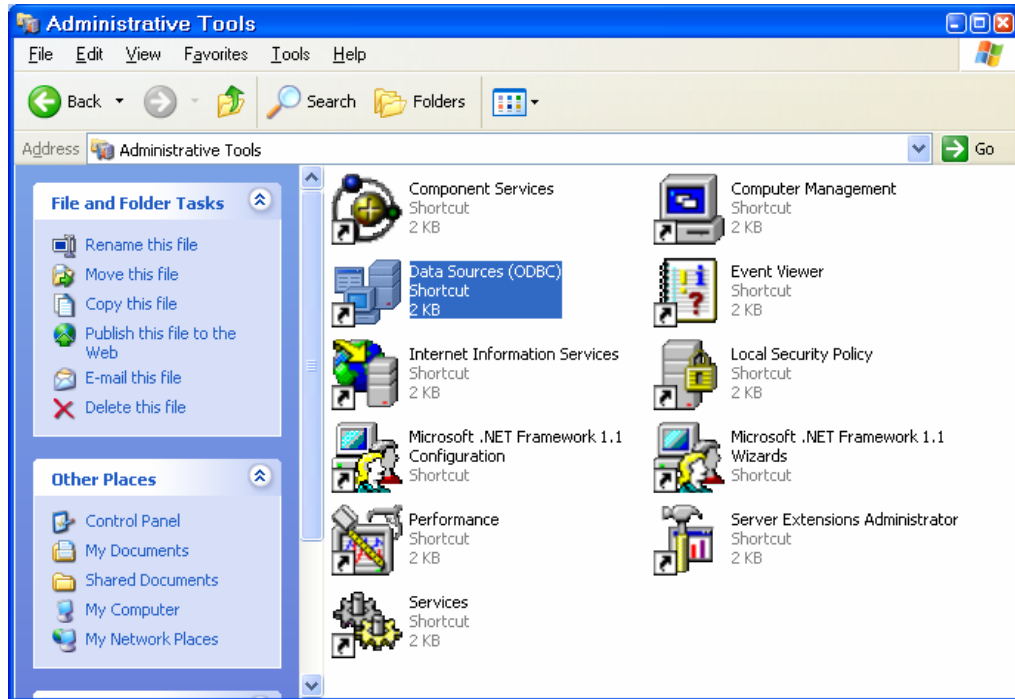
Cấu hình ODBC.

ODBC viết tắt là Open DataBase Connectivity, là công cụ kết nối. ODBC được cài đặt trên Windows ngay khi cài đặt, được sử dụng làm kết nối trung gian giữa ứng dụng và các hệ quản trị CSDL (Dbase, Access, SQL Server, Oracle,...).

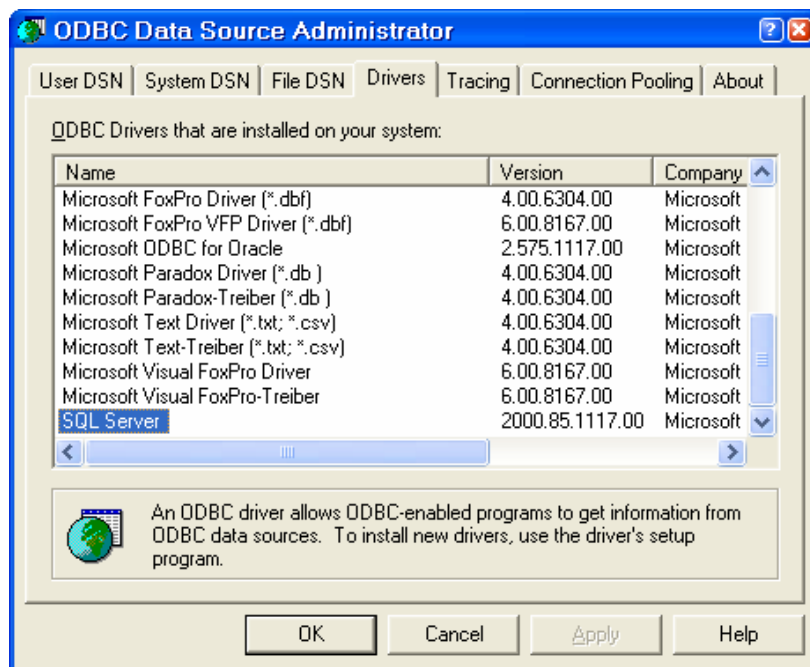
Thông qua ODBC ứng dụng chỉ cần xác định tên cá nhân trong ODBC (gọi là Data Source) và tài khoản khi truy nhập thì hiển thị quy định mà không cần quan tâm đến cấu trúc dữ liệu ứng dụng.

Thông thường khi cài đặt hệ quản trị CSDL mới thì Windows sẽ tự động thêm vào danh sách các Driver của hệ quản trị CSDL đó. Để hiển thị cấu hình ODBC cho SQL Server như sau:

- Chọn ODBC trong Administrative tools (Control panel).

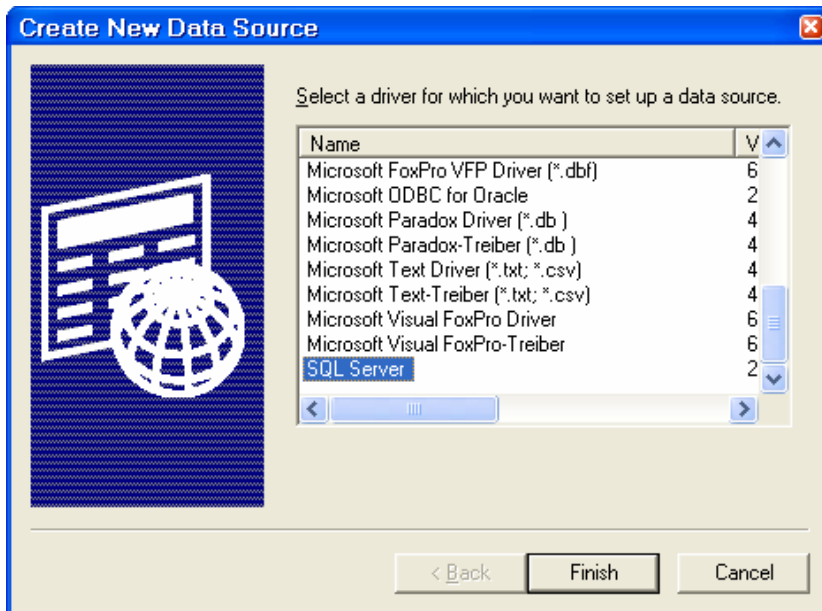


- Chọn bảng Drivers, trong danh sách kiểm tra xem đã có SQL Server chưa

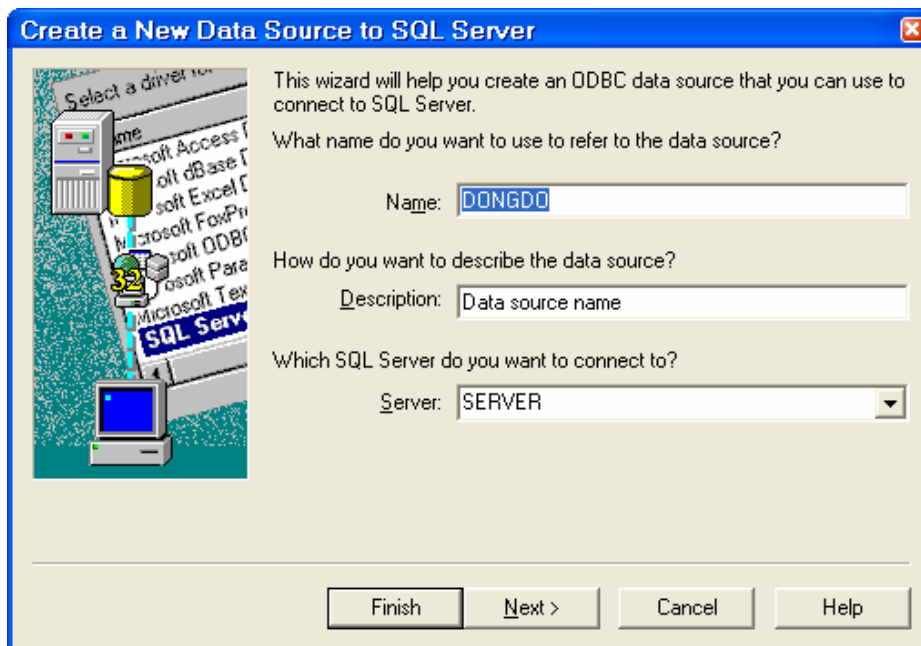


Nếu chưa có kiểm tra cách cài đặt SQL Server (thông thường Windows tự cập nhật).

- Chọn bảng User DSN (Data Source Name) -> Add.

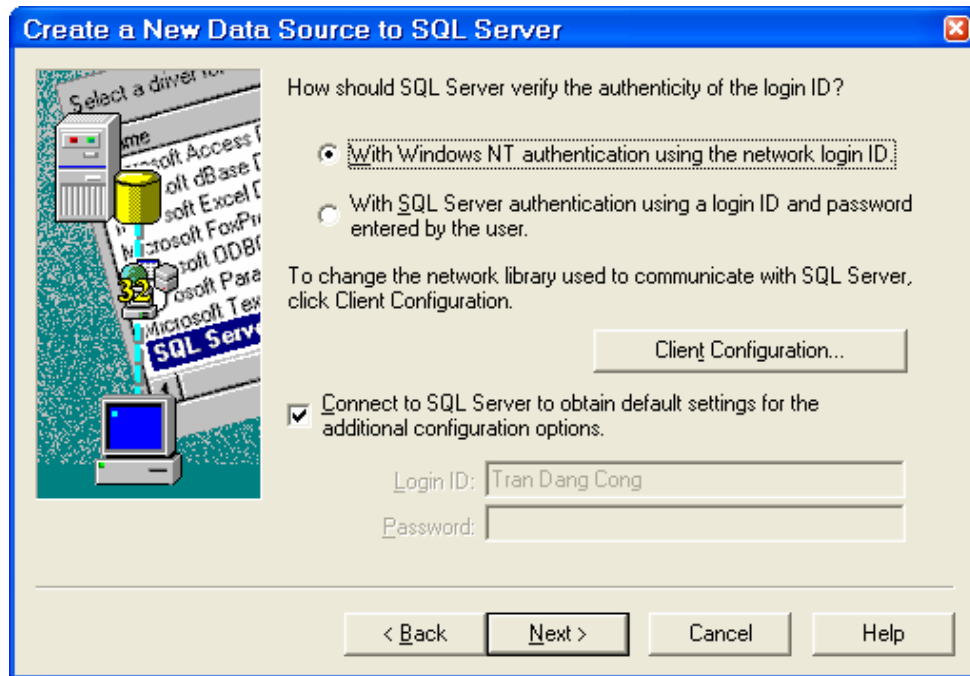


- Nhấn Finish.



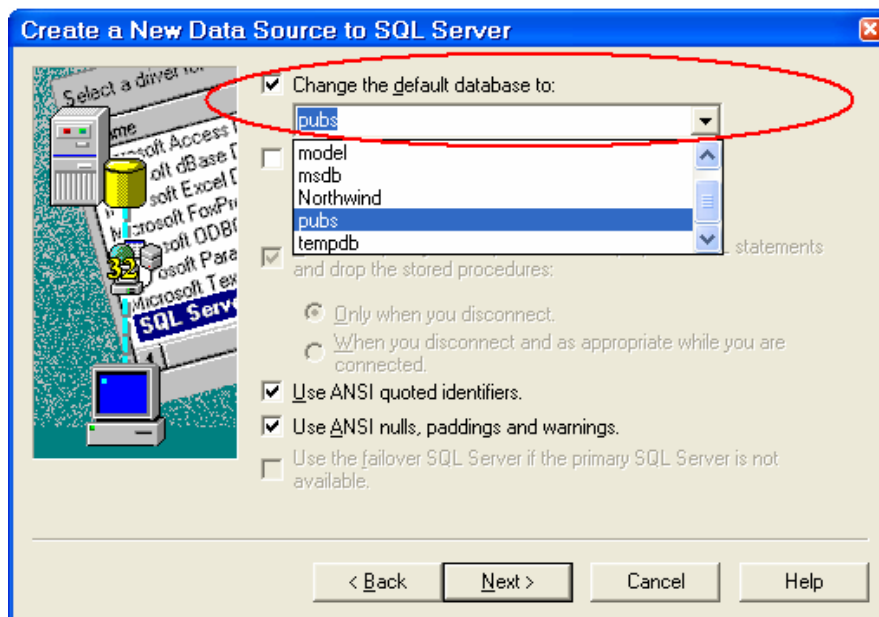
- Nhập tên DSN (đây là tên sẽ sử dụng cho ứng dụng), thông tin mô tả, tên Server (Instance).

- Nhấn Next.

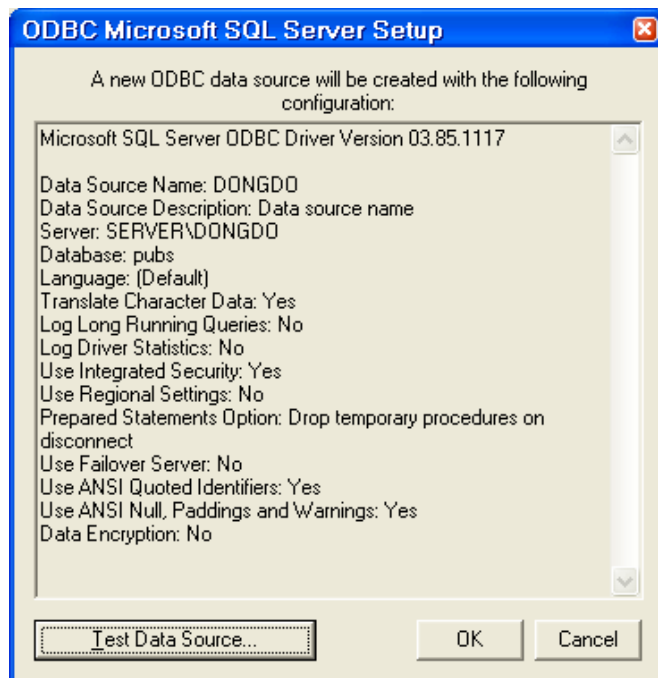


Trong case có 2 lựa chọn: Sử dụng chế độ bảo mật kết hợp của Windows NT hoặc của SQL Server (sẽ xem xét sau), trình tiếp theo này chọn lựa kết hợp của Windows NT (lúc này quy định khai thác là quy định của người truy cập vào Windows).

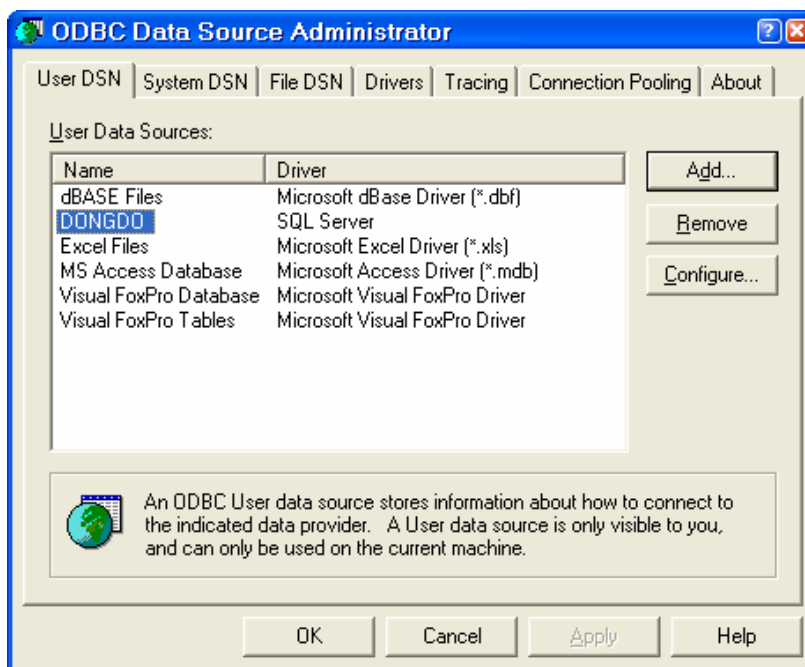
- Nhấn Next, chọn hộp chọn Change the default database to -> chọn cơ sở dữ liệu (vì các cơ sở dữ liệu sẽ xem xét bài sau, thì bây giờ chọn một cơ sở dữ liệu ví dụ có tên là Pubs, đây là cơ sở dữ liệu ví dụ do SQL Server thiết lập làm mẫu).



- Nhấn Next -> Finish.



ki m tra k t n i có thành công không b n nh n Test Data Source sau o nh n Ok k t thúc và thu c màn hình sau:



Trên danh sách các Data Source có tên DONGDO v a c t o, Data Source t o ra s c s đ ng trong ng đ ng client.

C u hình OLE DB.

OLE DB là ph ãng th ãc khá quen thu ãc ãi v ãi ãng ãi l p ãnh CSDL (l p ãnh trãnh trãnh Desktop ho ãc trãnh Internet). OLE DB s ã ãng v ãi nhi u h ã qu n trã CSDL khác nhau, m ãi h ã qu n trã có cú pháp riãng và ch ãnh driver ãi u khi nã cho nó. V ãi SQL Server th ãnh th ãnh s ã ãng 2 ph ãng th ãc k ã t n ãi s ã ãng OLE DB:

- Microsoft OLE DB Provider for SQL Server (SQLOLEDB): Kh ãnh s ã ãng ODBC, xác ãnh driver cho SQL Server.
- Microsoft OLE DB Provider for ODBC: S ã ãng ODBC ã t o (trong ph ãnh trã c).

QUẢN TRỊ CSDL LIU

Trong chương này ta sẽ xem xét cấu trúc vật lý, tạo, xóa, sửa tham số của CSDL Liu.

CẤU TRÚC CSDL LIU.

Chúng ta khi nghiên cứu về quản trị CSDL SQL Server bắt đầu sẽ xem xét về các hệ quản trị CSDL như DBase hoặc Access, và hệ quản trị CSDL như trên Microsoft Liu khi sử dụng (thực hiện các CSDL) sẽ mở kết nối tệp tin của CSDL, tệp tin của CSDL sẽ có mở tệp tin chính (ví dụ *.dbf hoặc *.mdb) và tệp tin phụ như khi ta thao tác ta chỉ cần quan tâm đến tệp tin chính. Nên trong các ứng dụng thông thường ta thường dùng các thao tác mở (open) mở tệp tin chính của CSDL và đóng (close) đóng tệp tin chính của CSDL mà không cần quan tâm đến việc kết nối CSDL của (không có phương thức kết nối).

SQL Server quản lý trực tiếp các CSDL, danh sách máy Server sẽ gồm danh sách các tên CSDL, tên các CSDL là duy nhất, không trùng nhau. Máy CSDL SQL Server sẽ quản lý các cấu trúc vật lý của nó. Chính thức cách thức quản lý như trên mà vì hệ quản trị CSDL Liu có một số đặc điểm sau:

- + Client khai thác CSDL trực tiếp thì thực hiện kết nối đến Server quản trị CSDL đó.

- + Chỉ thực hiện khai thác về các CSDL có tên trong danh sách các CSDL mà Server quản lý.

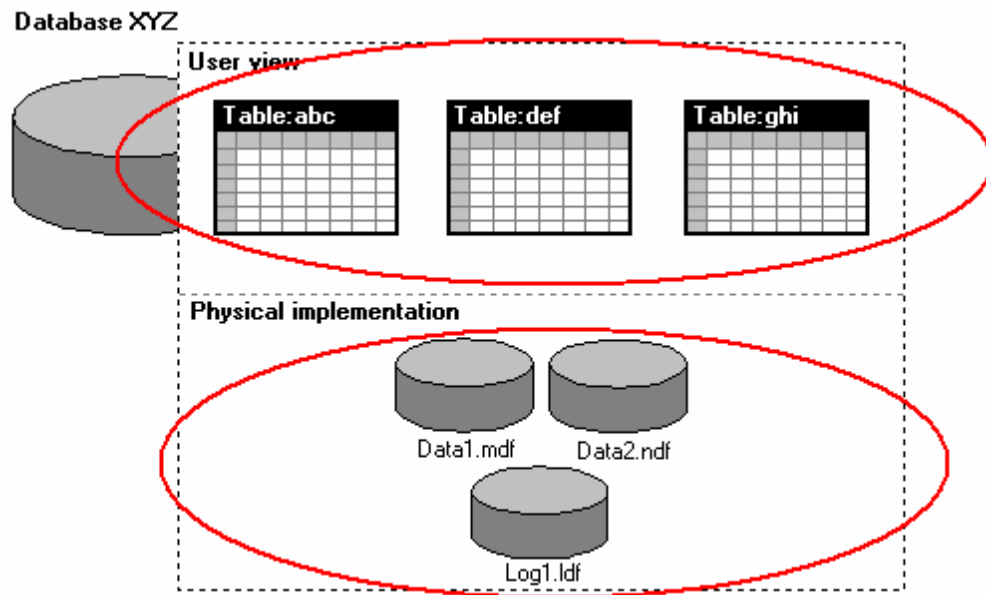
- + Không có các phương thức mở CSDL trực tiếp tệp tin như Dbase hoặc Access.

- + Khi kết nối đến Server, Client chỉ thực hiện các quy định khai thác theo quy định sẵn trong CSDL (phân quyền trong CSDL).

Cấu trúc CSDL Liu.

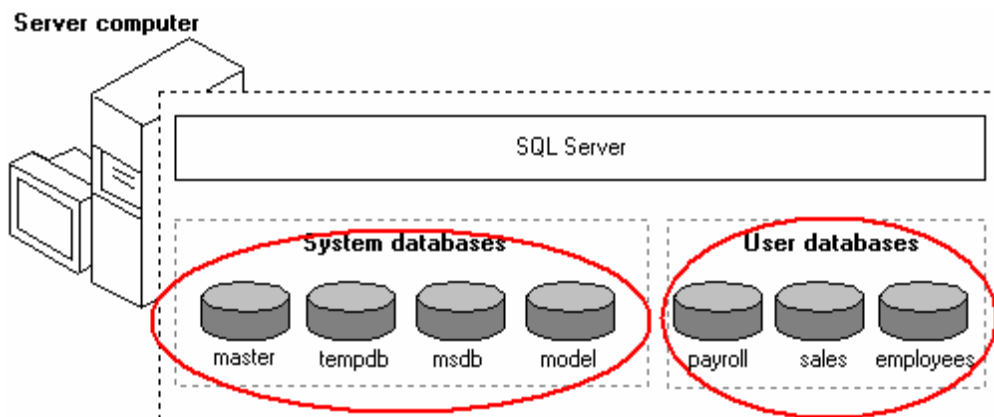
CSDL Liu trong SQL Server lưu trữ theo 2 phần: phần dữ liệu (gồm mở tệp tin bất kỳ *.mdf và các tệp tin phụ *.ndf) và phần nhật ký (*.ldf). Như vậy một CSDL Liu có ít nhất 2 tệp tin.

Cấu trúc logic trong CSDL gồm các table, view và các object khác. Sau đây là cấu trúc mở CSDL.



S qu n tr c s đ li u c a SQL Server.

C s đ li u trong SQL Server chia thành 2 lo i: C s đ li u h th ng (do SQL Server sinh ra khi cài t) và c s đ li u ng i dùng (do ng i dùng t o ta).



C s đ li u h th ng g m:

- Master: L u tr các thông tin login account, c u hình h th ng, thông tin qu n tr các CSDL, là CSDL quan tr ng nên th ng c sao l u b o m an toàn cho h th ng.
- Tempdb: Ch a các table t m th i và các th t c c l u tr t m th i. Các table và th t c nói trên c l u tr trong CSDL này ph c v cho các user.

- Model: c s d ng khi template c s d ng cho các CSDL c t o trên m t h th ng.
- Msdb: S d ng b i SQL Agent.

T p tin c a các CSDL nói trên nh sau:

T p tin CSDL	Tên t p tin v t lý	Kích th c ng m nh
master primary data	Master.mdf	11.0 MB
master log	Mastlog.ldf	1.25 MB
tempdb primary data	Tempdb.mdf	8.0 MB
tempdb log	Templog.ldf	0.5 MB
model primary data	Model.mdf	0.75 MB
model log	Modellog.ldf	0.75 MB
msdb primary data	Msdbdata.mdf	12.0 MB
msdb log	Msdblog.ldf	2.25 MB

C u trúc v t lý c a CSDL.

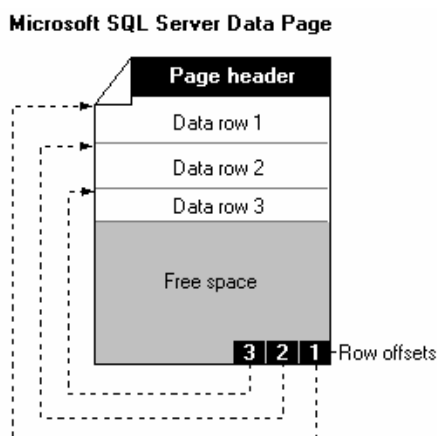
Nh các c u trúc các CSDL h qu n tr CSDL thông th ng (Dbase, Access), SQL Server c ng qu n lý t p tin đ li u c a CSDL đ ng v t lý theo trang (page) và phân o n (extent).

Page.

SQL Server qu n lý m t page có kích th c là 8KB, nh v y 1MB có 128 page, trong m i trang có 96 byte ch a thông tin c a trang. Có 8 ki u page nh sau:

Tên	N i dung
Data	Ch a t t c các ki u đ li u lo i tr text , ntext và image
Index	Các khóa Index.
Text/Image	Text , ntext , and image data.
Global Allocation Map, Secondary Global Allocation Map	Ch a các thông tin nh v c a các extent.
Page Free Space	Ch a thông tin kho ng tr ng c a page.
Index Allocation Map	Ch a các thông tin v Extent ã s d ng cho Index và Page.
Bulk Changed Map	Ch a thông tin v các l nh BACKUP LOG.
Differential Changed Map	Ch a các thông tin l nh BACKUP DATABASE.

ivi các t p tin nh t ký (*.ldf), các b n ghi c ghi l i liên t c, không phân trang.



D li u trong m t trang s b t u l u tr t sau ph n thông tin Header, và l u tr liên ti p, m i hàng có kích th c t i a là 8060byte. Riêng ivi d li u ki u text, ntext, image ây là ki u d li u ph c t p và có kích th c l n, SQL Server s có chỉ n l c qu n lý khác, phân tran riêng nh m t ng hi u qu truy v n d li u.

D li u trong SQL Server c l u tr trên a và t o ch m c Index theo c u trúc d li u ki u B-Tree Plus (có th tham kh o thêm trong nh ng n i dung c u trúc d li u nâng cao).

Extent.

Extent là n v dùng ch a các table và index, m i extent có 8 page hay 64KB. SQL Server có 2 ki u extent:

- Uniform: Ch dùng l u tr cho m t i t ng,.
- Mixform: Có th dùng l u tr 8 i t ng.

C u trúc Extent nh sau:



File.

T p tin l u tr m t CSDL trong SQL Server có 3 lo i.

Primary data file: Là file chính l u tr d li u (*.mdf = Master Data File), m i CSDL có m t file primary, l u tr i m b t u c a m t CSDL và các i m k t n i n các file l u tr ti p theo (secondary).

Secondary data file: Là t p tin l u tr d li u sau Primary data file, m t CSDL có th có nhi u t p tin secondary. Lo i t p tin này cho phép m t CSDL có th phân tán d li u nhi u n i trên máy tính ho c trên m ng.

Log file: Là lo i t p tin l u tr thông tin nh t ký c a CSDL.

Gi s t o m t CSDL có tên MyDB, thông th ng h th ng ng m nh các t p tin nh sau:

MyDB_primary

c:\Program Files\Microsoft SQL Server\MSSQL\Data\MyData1.mdf

Primary data file

MyDB_secondary1

c:\Program Files\Microsoft SQL Server\MSSQL\Data\MyData2.ndf

Secondary data file

MyDB_secondary2

c:\Program Files\Microsoft SQL Server\MSSQL\Data\MyData3.ndf

Secondary data file

MyDB_log1

c:\Program Files\Microsoft SQL Server\MSSQL\Data\MyLog4.ldf

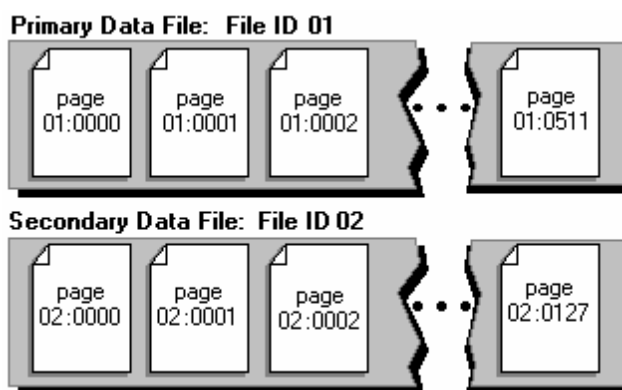
Log file

MyDB_log2

c:\Program Files\Microsoft SQL Server\MSSQL\Data\MyLog5.ldf

Log file

Các tệp tin lưu trữ dữ liệu phân thành từng trang, các trang ảnh hưởng liên tiếp theo từng file:



File group.

SQL Server sử dụng công cụ file group giúp người dùng dễ dàng quản lý file, các file lưu trữ dữ liệu của một CSDL có thể nhóm thành từng nhóm, gồm 2 kiểu nhóm chính:

- Primary: Là nhóm bắt buộc có, dùng xác định cho file primary (*.mdf) và những file khác.
- User-defined: Nhóm do người dùng tạo ra, đặt tên tùy ý.

QUẢN LÝ CSDL LIU.

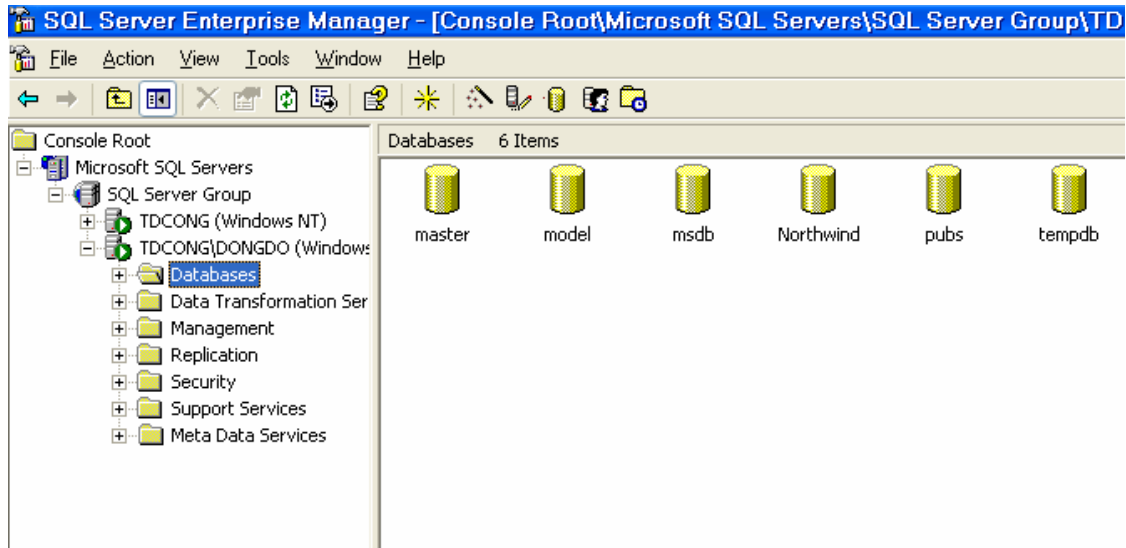
Tổng số dữ liệu.

Theo lý thuyết cơ sở dữ liệu, trước khi tạo CSDL ta phải thực hiện phân tích các thông tin liên quan mật thiết sử dụng CSDL cho bài toán của mình: Tên CSDL, các table, ràng buộc, ... tuân theo các chuẩn CSDL (phần này sẽ bàn kỹ trong bài sau)

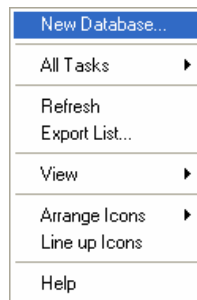
Trong các thao tác với CSDL và một số công việc khác sẽ gồm 2 phần: Phần thao tác theo công cụ wizard và câu lệnh T-SQL.

T o theo công c :

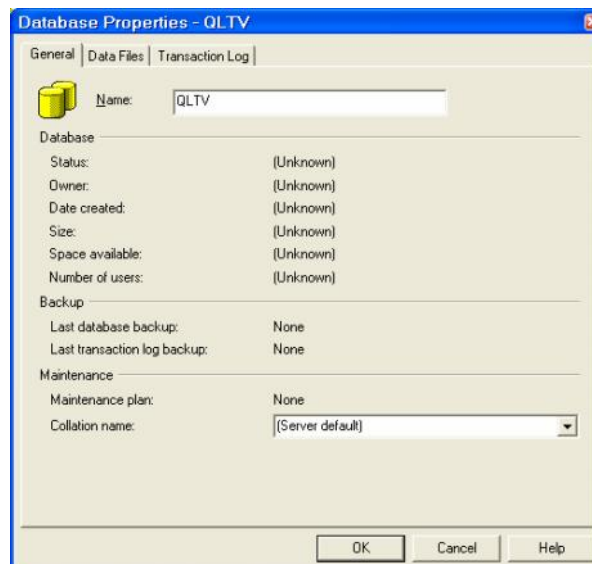
- Vào Enterprise Manager -> Databases.



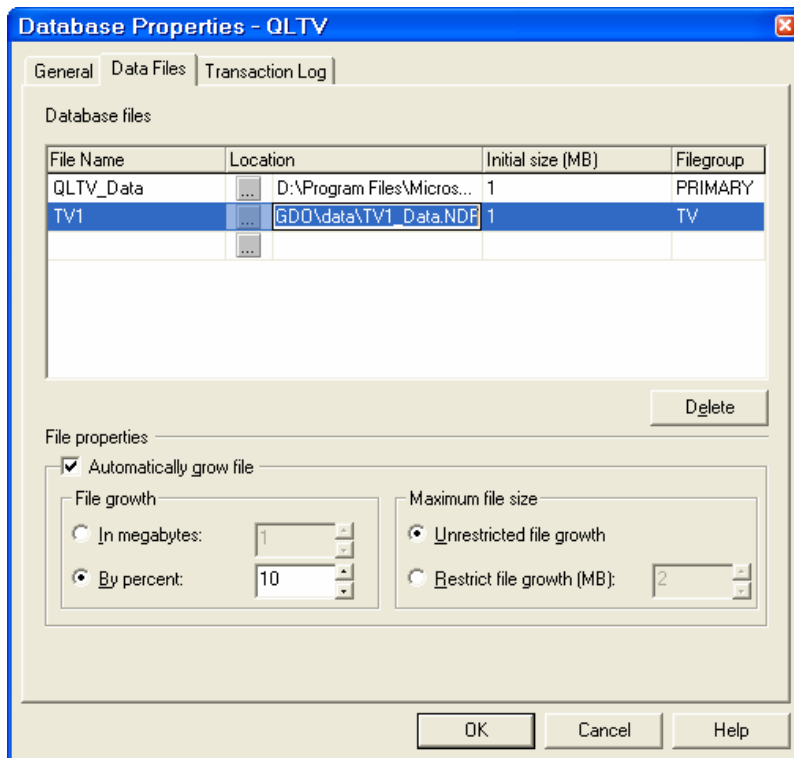
- Nhấn nút phải chuột/hoặc menu Action -> New Database...



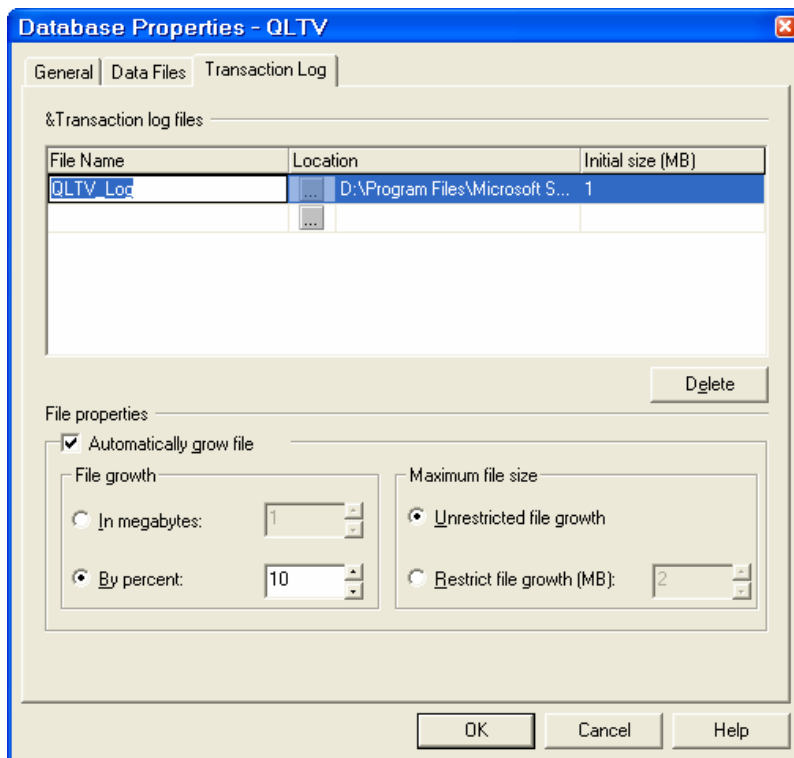
- Nhập tên CSDL.



- Xác định tên logic, tên vật lý, tên nhóm catalog và các tham số khác.



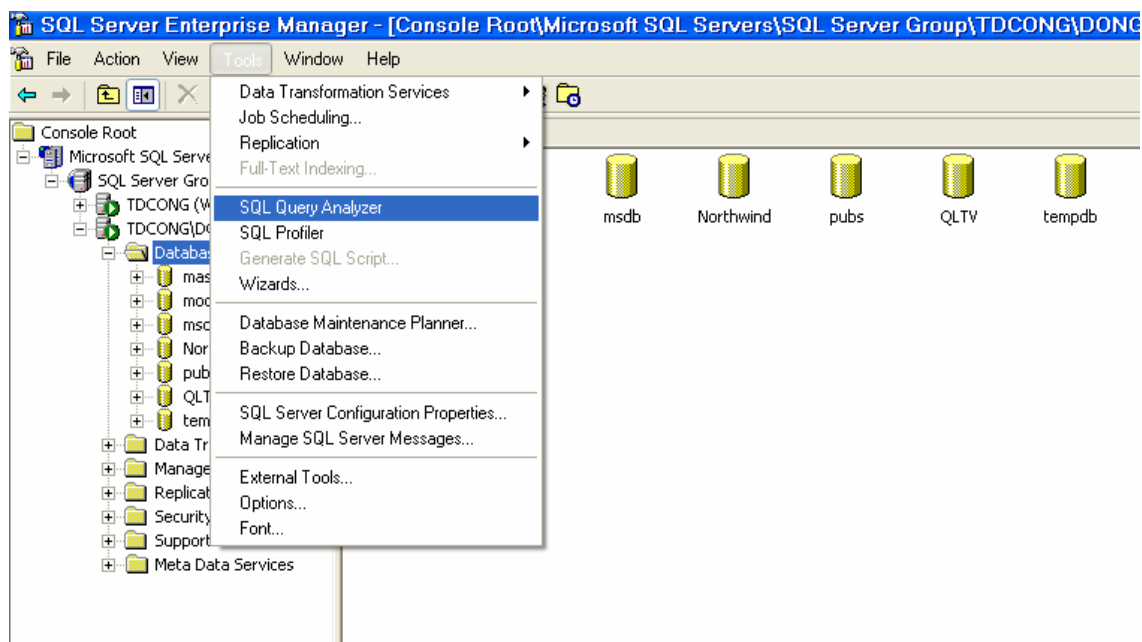
- Xác định tên logic, vật lý, tham số khác và tính năng.



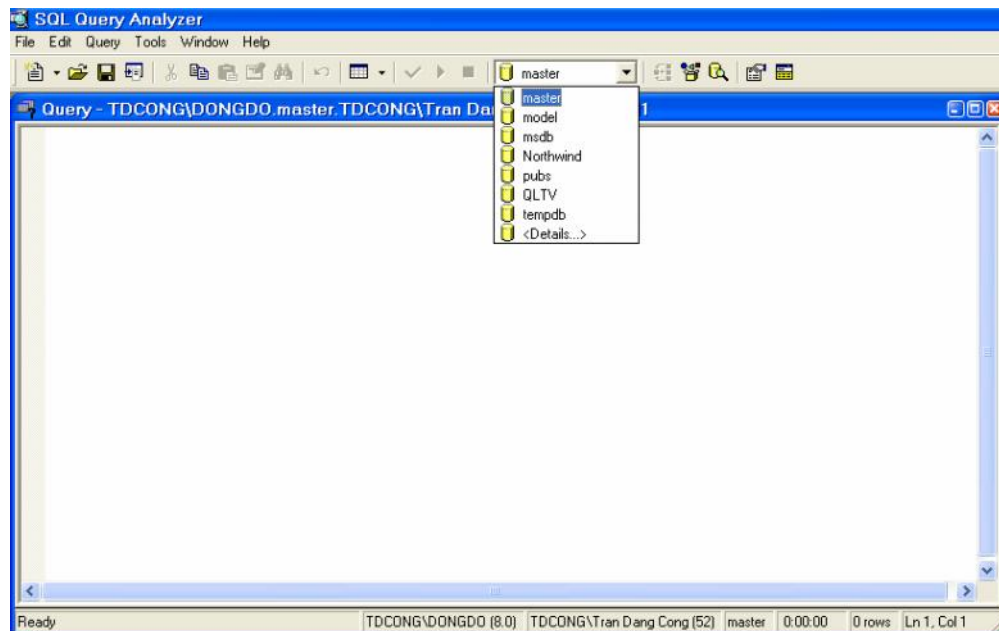
T o theo câu l nh.

S d ng câu l nh Create Database t o CSDL, công c th c hi n l nh:

- Trong Enterprise Manager -> Databases -> Tools -> SQL Query Analyzer



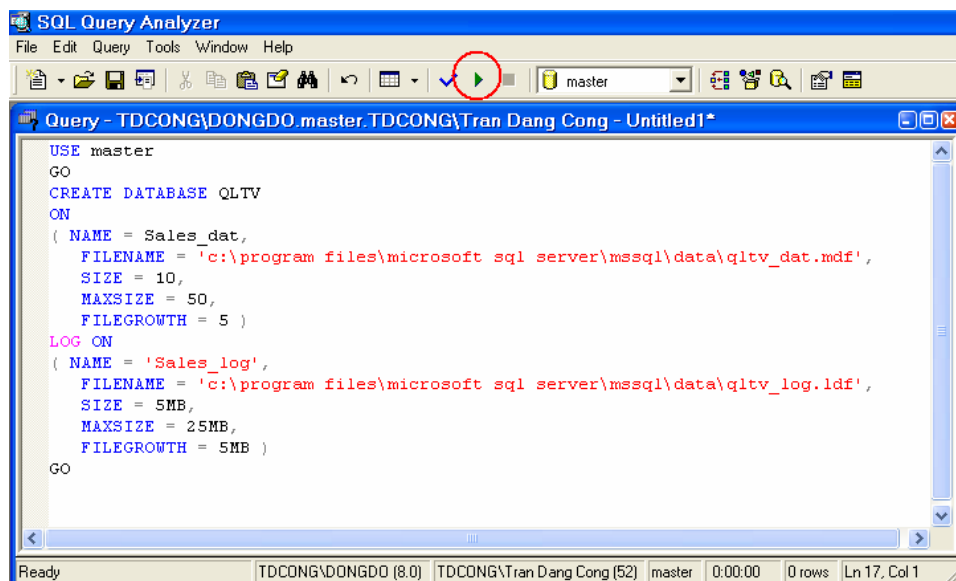
- Ch n CSDL Master.



- So n l nh trong c a s l nh:

```
USE master
GO
CREATE DATABASE QLTV
ON
( NAME = Sales_dat,
  FILENAME = 'c:\program files\microsoft sql
server\mssql\data\qltv_dat.mdf',
  SIZE = 10,
  MAXSIZE = 50,
  FILEGROWTH = 5 )
LOG ON
( NAME = 'Sales_log',
  FILENAME = 'c:\program files\microsoft sql
server\mssql\data\qltv_log.ldf',
  SIZE = 5MB,
  MAXSIZE = 25MB,
  FILEGROWTH = 5MB )
GO
```

- Nh n F5 ho c nút th c hi n.



Công c SQL Query Analyzer cho phép b n th c hi n t ng câu l nh b ng cách bôi en vào o n l nh c n th c hi n sau ó nh n F5 ho c nút th c hi n.

Khi t o CSDL m i thì n ph i ng v trí CSDL Master, khi mu n th c hi n l nh v i m t CSDL c th ã có nào ó b n ph i ch n vào CSDL ó và th c hi n l nh.

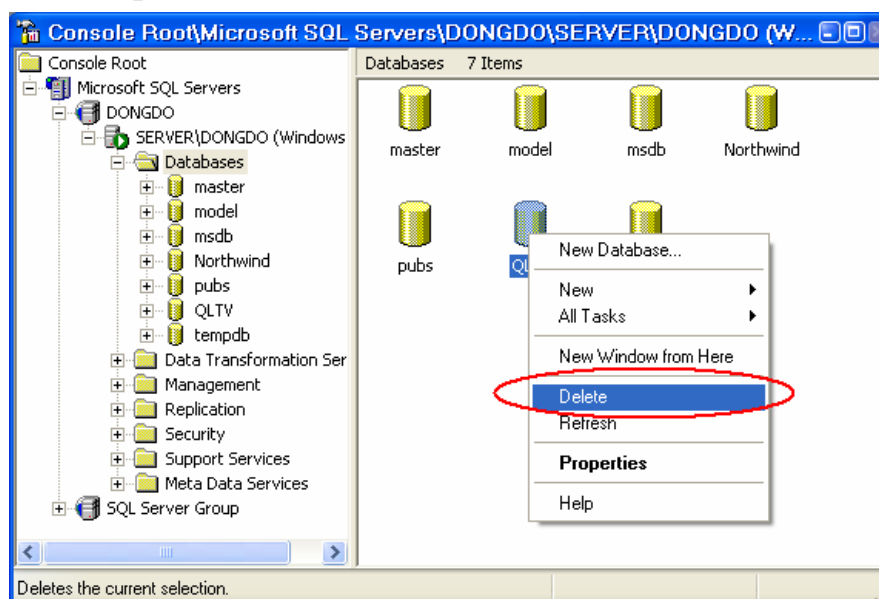
Trong số những lệnh trên lệnh User Master thể hiện chính CSDL Master bằng câu lệnh; lệnh Use xác định CSDL thể hiện.

Lệnh Go xác định câu lệnh kết thúc và bắt đầu câu lệnh khác, câu lệnh cuối luôn là dòng lệnh. Trong lệnh T-SQL một số lệnh khác nhau vẫn có thể nằm trên một dòng lệnh nên trong một số tình huống kích bản câu lệnh không cần sử dụng lệnh Go.

Xóa cơ sở dữ liệu.

Xóa theo công cụ .

- Chọn vào CSDL.
- Nhấn nút phích cắm -> Delete.



- Chọn Yes.

Xóa theo câu lệnh.

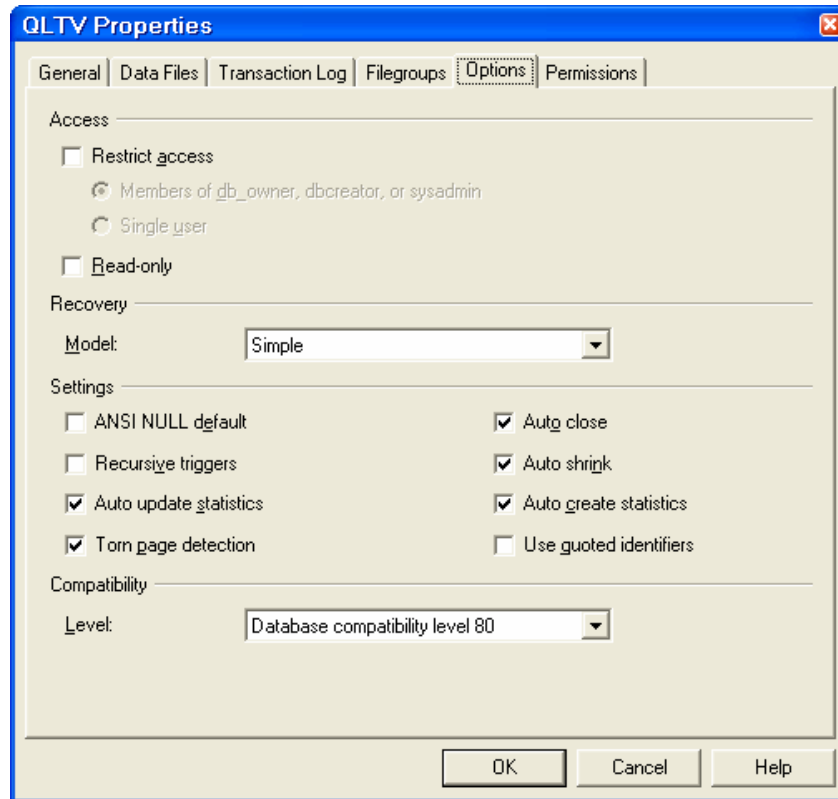
- Sử dụng lệnh Drop Database
Drop Database QLTV

Sửa tham số .

Sửa theo công cụ .

- Chọn CSDL.
- Nhấn nút phích cắm

- Ch n Properties.



- Thay i tham s khi c n thi t:

+ Restrict access: Ng n truy nh p.

+ Read only: t thu c tính ch c.

S a theo câu l nh.

- S d ng câu l nh Alter Database, ví d sau th c hi n thêm t p tin secondary vào CSDL.

```
ALTER DATABASE QLTV
ADD FILE
(
  NAME = QLTV_newfile,
  FILENAME = 'c:\Program Files\Microsoft SQL
Server\MSSQL\Data\newf.ndf',
  SIZE = 5MB,
  MAXSIZE = 100MB,
  FILEGROWTH = 5MB
)
GO
```

- S a tham s d a vào th t c h th ng sp_dboption:

+ S a i thu c tính read only:

```
USE master
```

```
EXEC sp_dboption 'qltv', 'read only', 'TRUE'
```

+ S a thu c tính autoshring

:

```
USE master
```

```
EXEC sp_dboption 'qltv', autoshring, TRUE
```

+ S a thu c tính single user:

```
USE master
```

```
EXEC sp_dboption 'qltv', single_user
```

M i câu l nh liên quan b n có th tra c u, tham kh o trong Book Onlines.

B NG D LI U – TABLE

CÁC CHU N T C.

Trong thi t k c s d li u, vi c tuân th ng t ngèo nh ng chu n là vi c h t s c quan tr ng, nó giúp cho vi c qu n tr d li u có hi u qu , kh c ph c d th a, thu n l i trong qu n tr d li u l n, hi u qu v i d li u ph c t p.

G m 3 chu n c b n:

Chu n th nh t.

Chu n th nh t xác nh c u trúc c a m t b ng không th ch a các tr ng l p l i.

Ta có th l y ví d nh sau gi s mu n l u tr thông tin m t quy n sách, m i quy n sách có th có m t ho c nhi u tác gi tham gia biên so n, n u không tuân theo chu n th nh t nh n u trên thì trong m t b ng d li u sách có th có nhi u tr ng d li u xác nh thông tin tác gi .

ID	Tên sách	NXB	Tác gi 1	Tác gi 2

Trong ví d trên b n nh n th y th ng tin Tác gi c l p l i 2 l n, kh ph c b ng cách t o ra m t b ng l u tr danh sách tác gi c a sách (s b n trong chu n sau).

Chu n th hai.

Chu n th hai xác nh trong các hàng d li u, m i c t u ph thu c vào c t khóa chính. Ta xem xét m t tr ng h p vi ph m chu n th hai nh sau:

Gi s xét tình hu ng sinh viên m n sách trong m t th vi n, vi c m n sách c nh t ký theo b ng nh sau:

Id_sach	Id_Sinhvien	Ngày m n	S c kh e sinh viên

Xem xét trong b ng trên ta th y m i hàng ph thu c vào khóa id_sach và id_sinhvien, nh ng thông tin S c kh e sinh viên không ph thu c vào id_sach, nên thông tin này c n chuy n sang b ng v thông tin c a sinh viên.

Chu n th ba.

Chu n th ba xác nh b n ghi tuân th theo chu n th hai và không có b t k ph n ph thu c chuy n ti p nào. Ph n ph thu c chuy n ti p t n t i khi m t b ng ch a m t c t c tr ng. C t này không ph i là khóa nh ng v n xác nh các c t khác.

Ta xem xét m t ví d vi ph m chu n nh sau:

Gi s trong th vi n có m t b ng li t kê sách t n trong kho, khi sinh viên m n sách s l ng sách mà sinh viên m n s t ng, n u nh t ký m n sách c th c hi n theo b ng sau:

Id_sach	Id_Sinhvien	Ngày m n	S l ng ã m n

B ng trên b n th y m i l n sinh viên m n sách s l ng sách có mã id_sach mà sinh viên có mã id_sinhvien s t ng lên và t ng s là S l ng ã m n, thông tin này là thông tin tích l y theo id_sach, id_sinhvien, ngày m n.

Theo b ng trên ta th y không vi ph m chu n th hai nh ng vi ph m chu n th ba vì c t S sách ã m n là c t ph thu c chuy n ti p, c t này c n ph i c chuy n sang b ng khác là b ng Sinh viên m n sách:

Id_sach	Id_Sinhvien	S l ng ã m n

Khi nào c n chu n t c.

M t c s d li u c n c chu n t c khi:

- D li u l n, phân tán.
- Không xác nh rõ nhóm d li u.
- D li u ph c t p.
- B c u tiên khi xây d ng ng d ng.

Khi nào không c n chu n t c hóa.

M t s tình hu ng s không c n chu n t c hóa, n u theo nh thi t k theo chu n thì vì c a ra m t m u tin truy v n có th ph i th c hi n truy xu t t nhi u b ng v i nhau, i u này có ngh a ta ph i th c hi n k t h p các b ng v i nhau (tuy theo lu t) nên th i gian truy xu t có th r t l n mà yêu c u th c t t ra trong tình

hu ng này là ph i nhanh, thì truy xu t theo m t b ng ã có s n là nhanh h n, sau ây là m t s tr ng h p không c n chu n tác hóa (tùy theo tình hu ng):

- Thông tin tính toán.
- Thông tin s ki n.
- S phân ho ch.

THI T K B NG D LI U.

Table (b ng d li u) là m t thành ph n c b n c a CSDL, m t CSDL c thi t k t m t ho c nhi u b ng d li u, m i b ng d li u c c u trúc t các hàng và c t d li u, m i hàng dùng mô t m t i t ng, v n , s ki n,... c t th hi n thu c tính c a các i t ng, s ki n,... c a hàng. D li u cùng c t có cùng ki u (data type). Ngoài các hàng, c t b ng còn có các khóa, liên k t, ràng bu c,...

Tr c khi b t tay vào thi t l p b ng d li u tr c h t ta ph i xác nh xem b ng s xây d ng nh th nào, d a trên m t s thông tin sau:

- Ki u d li u trong b ng.
- Các c t, ki u d li u t ng ng (và dài n u c n thi t).
- C t nào cho phép giá tr NULL (là giá tr mà ph n d li u thu c hàng, c t xác nh không c gán giá tr nào, vì v y nên 2 ph n t có cùng giá tr NULL là không b ng nhau).
- Giá tr ng m nh (là giá tr mà khi ch a nh p vào nó nh n giá tr này).
- Ch s Index, khóa chính, khóa ngoài.

Ki u d li u.

SQL Server g m nh ng ki u d li u sau:

Binary: Là ki u d li u ch a d ng s h hexa, g m 3 ki u d li u Binary, Varbinary, Image.

Text: Là ki u ký t , ch a ch cái, ký hi u, s , g m nh ng ki u d li u sau:

- Char: Ki u ký t , khi xác nh dài thì dài trong CSDL s xác nh theo dài t tr c mà không theo dài d li u th c có, không s d ng v i ký t d ng Unicode, dài t i a là 8000.

- Nchar: T ng t nh Char nh ng s d ng v i ký t Unicode, dài t i a 4000.

- Nvarchar: Tên gọi NChar nhưng kích thước trong CSDL sẽ là kích thước của dữ liệu nhị phân, không tính theo kích thước thực, kích thước tối đa là 4000.

- Varchar: Tên gọi Nvarchar nhưng không hỗ trợ Unicode.

- Text: Kiểu văn bản, chứa ký tự xuống dòng, lưu trữ theo dạng văn bản, có kích thước lớn, có thể lên đến vài Gb, cách quản lý dữ liệu theo dạng con trỏ và cách thức chèn và cập nhật khác, dữ liệu này không hỗ trợ cho Unicode.

- Ntext: Tên gọi Text nhưng có hỗ trợ Unicode.

Data/Time: Kiểu dữ liệu ngày, thời gian, ngày và thời gian, gồm 2 kiểu:

- DateTime: Kiểu ngày và thời gian.

- SmallDateTime: Chỉ ngày hoặc thời gian.

Numeric: Dữ liệu số, gồm các kiểu dữ liệu sau:

- Int, smallint, tinyint, bigint: Số nguyên

- Float, real, decimal, numeric: Số thập phân.

Monetary: Tiền tệ:

- Money, Smallmoney.

Bit: Kiểu số 0, 1.

Sql_variant: Là kiểu dữ liệu xác định theo kiểu dữ liệu khác, một cột dữ liệu có thể chứa dữ liệu kiểu này có thể lưu trữ nhiều dữ liệu có kiểu khác nhau trong cùng một bảng. Ví dụ có thể lưu trữ nhiều kiểu dữ liệu **int**, **binary**, **char**, nhưng không chứa dữ liệu kiểu **text**, **ntext**, **image**, **timestamp**, **sql_variant**.

Timestamp: Là kiểu dữ liệu có kích thước 8 byte, lưu trữ dữ liệu số phân độ chính xác sinh ra, mỗi giá trị timestamp trong CSDL là duy nhất.

Table: Là kiểu dữ liệu các bit lưu trữ tập hợp các hàng (dòng bảng), mục đích sử dụng chính là lưu trữ tập hợp các hàng sau truy vấn.

Text in row.

Như xem xét trước, dữ liệu kiểu char, varchar có độ dài tối đa là 8000byte, dữ liệu kiểu text, ntext có 2 kiểu lưu trữ: lưu trữ trực tiếp, lưu trữ quản lý theo kiểu con trỏ.

- `iv` là `iv` theo kiểu `text`, kích thước `iv` là 8000, `ivntext` là 4000 (kích thước ký tự mã Unicode là 2 byte, mã không Unicode là 1 byte).

- Lưu trữ, quản lý theo container kích thước lên đến GB.

Lưu trữ dữ liệu theo kiểu container tiên tiến **Text in row** với trạng thái **On**, thuộc tính này giúp việc lưu trữ dữ liệu image.

Sử dụng `sp_tableoption` thay đổi thuộc tính, thuộc tính thay đổi theo bảng dữ liệu.

Giới thiệu cách `text in row` như sau:

```
Sp_tableoption N'TacGia', 'text in row', 'ON'
```

Tắt cách `text in row` như sau:

```
Sp_tableoption N'TacGia', 'text in row', 'OFF'
```

Cập nhật dữ liệu khi thuộc tính `text in row` là `ON`, ta phải dùng lệnh `READTEXT`, `UPDATETEXT`, `WRITETEXT` (sử dụng câu lệnh như sau).

Auto number.

Loại dữ liệu `int`, `bigint` khi thêm cột mới, cột kiểu này không sử dụng dữ liệu. Dữ liệu kiểu này tăng lên vì khi thêm hàng dữ liệu chèn thêm giá trị tăng theo hàm `NewID()`.

Ràng buộc dữ liệu.

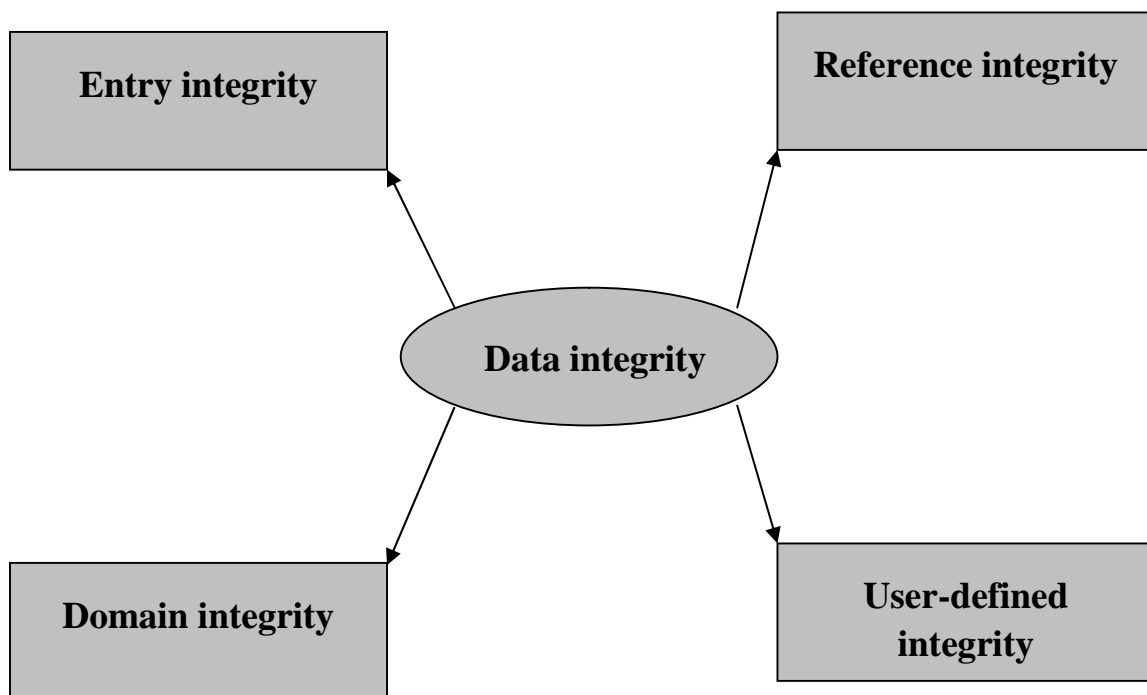
Có một CSDL khi lưu trữ dữ liệu có tính chính xác cao, nhanh và thuận tiện trong khai thác dữ liệu thì toàn vẹn dữ liệu là vấn đề quan trọng. Khi ràng buộc dữ liệu khi nhập vào CSDL sẽ kiểm soát, tính y thống nhất của cơ sở dữ liệu.

Có nhiều kiểu ràng buộc dữ liệu, một CSDL có thể gồm một hoặc nhiều ràng buộc, ràng buộc có thể trên một bảng, trên nhiều bảng.

Toàn vẹn dữ liệu chia thành 4 loại:

- Toàn vẹn thực thể (Entry integrity): Mỗi thực thể xác định theo một khóa, khi bị khóa ta hoàn toàn có thể xác định thực thể tương ứng. Khóa này coi là khóa chính.

- Toàn vẹn theo miền (Domain integrity): Là loại toàn vẹn có hiệu lực với các cột dữ liệu trong một phạm vi nào đó, ví dụ dữ liệu ngày là một dạng của toàn vẹn miền, ràng buộc theo khóa check ngày là toàn vẹn theo miền.
- Toàn vẹn tham chiếu (Referential integrity): Khi một bảng có quan hệ với một bảng khác theo một mối quan hệ, trong mối quan hệ đó sẽ có một khóa chính (nhập phần tử vào trong đó) và một khóa ngoài, khóa ngoài sẽ là khóa tham chiếu của khóa chính, giá trị của khóa ngoài sẽ thuộc tập các giá trị của khóa chính hoặc giá trị NULL. Ràng buộc kiểu quan hệ (Relationship) gọi là toàn vẹn kiểu tham chiếu.
- Toàn vẹn do người dùng định nghĩa (User-defined integrity): Là toàn vẹn do người dùng định nghĩa, quy định dữ liệu nhập vào theo quy cách, giá trị kiểm soát chặt chẽ, toàn vẹn kiểu này cũng có thể xây dựng trên các các toàn vẹn trước.



Bên dưới toàn văn nói trên ta có thể thấy những quy tắc ràng buộc trong SQL Server như sau:

Kiểu toàn văn	Công cụ trong SQL Server
Entry integrity	<ol style="list-style-type: none"> Ràng buộc Primary key Ràng buộc Unique Cột Identity
Domain integrity	<ol style="list-style-type: none"> Giá trị mặc định Default Ràng buộc khóa ngoài Foreign Key Ràng buộc Check Thuộc tính NOT NULL
Referential integrity	<ol style="list-style-type: none"> Ràng buộc Foreign Key Ràng buộc Check
User-defined integrity	<ol style="list-style-type: none"> Rules Stored procedures Triggers

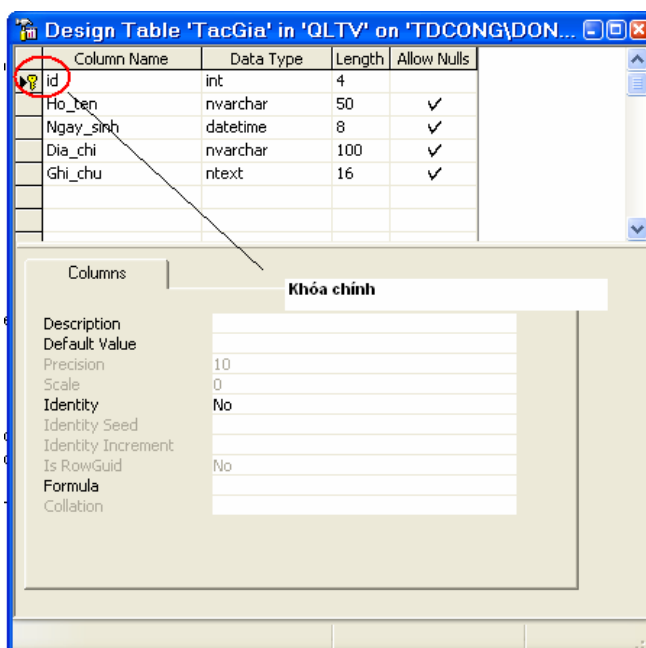
Các khóa.

Khóa chính – Primary Key.

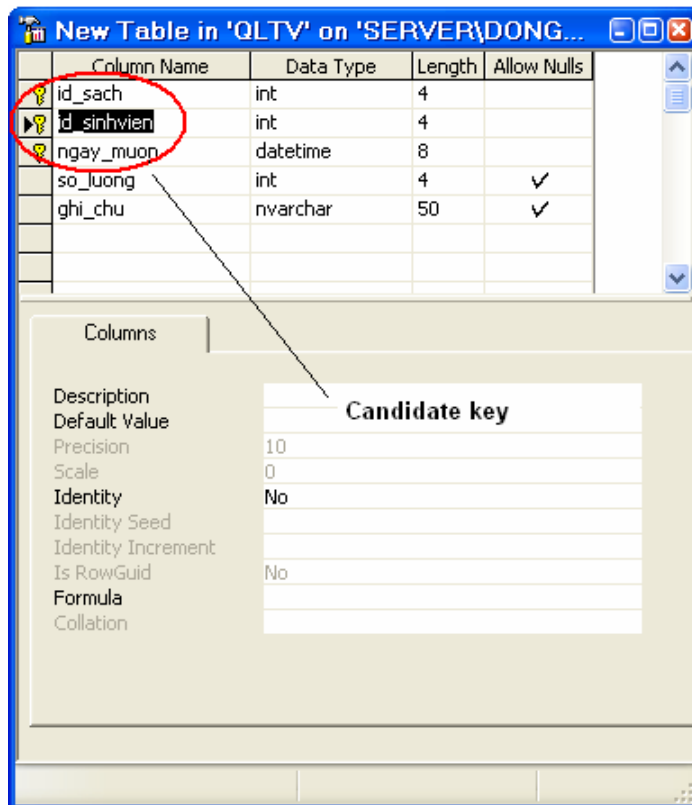
Là một thuộc tính dữ liệu xác định duy nhất trong một bảng, giá trị khóa chính luôn khác NULL.

Ví dụ: Bảng danh sách tác giả viết sách.

Trong ví dụ trên bảng dữ liệu có khóa chính là một cột dữ liệu id, khi cột xác định là khóa chính bên cạnh nút chọn khóa, thuộc tính Allow Nulls không có ảnh hưởng.



Ví dụ: Bảng dữ liệu lưu trữ thông tin nhât ký mượn sách.



Trong ví dụ trên bảng dữ liệu có khóa chính cột hợp 3 cột dữ liệu id_sach, id_sinhvien, ngay_muon, ba cột trên xác định duy nhất một sinh viên mượn một quyển sách trong một ngày (giới hạn quy định xác định ngày), các cột tham gia khóa chính gọi là candidate key.

Khóa ngoài.

Theo chuẩn thiết kế CSDL, khi lưu trữ thông tin sách phải có một cột chứa thông tin nhà xuất bản. Một nhà xuất bản có thể xuất bản nhiều quyển sách và một quyển sách chỉ xuất bản một nhà xuất bản. Nên trong thiết kế ta phải có:

+ Bảng dữ liệu lưu trữ danh sách các nhà xuất bản: Có khóa chính là địa chỉ cho nhà xuất bản.

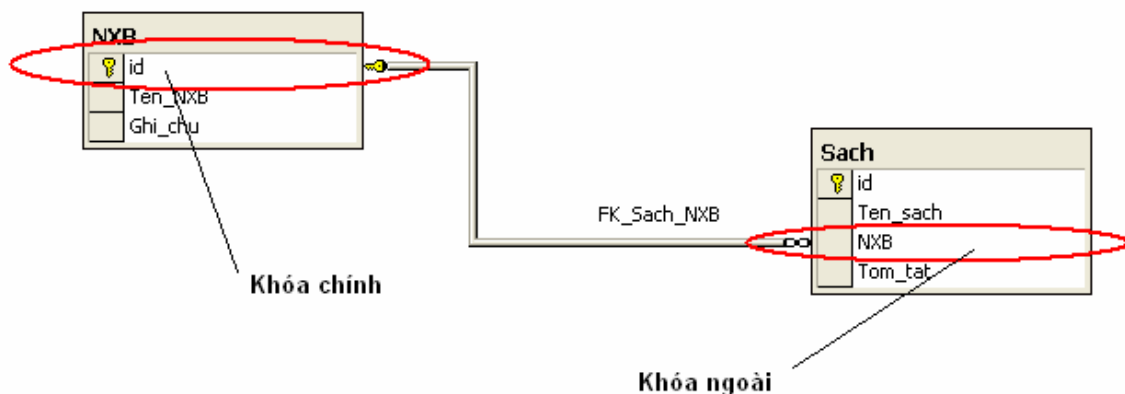
+ Bảng dữ liệu lưu trữ sách: Có chứa thông tin nhà xuất bản.

+ Quan hệ giữa nhà xuất bản và sách: Mã khóa nhà xuất bản thuộc bảng nhà xuất bản và thông tin nhà xuất bản thuộc bảng sách, cột thông tin nhà xuất bản thuộc bảng sách tham gia quan hệ trên gọi là khóa ngoài (Foreign key).

Cột dữ liệu là khóa ngoài có thể có quan hệ với nhiều khóa chính nhiều bảng, một bảng có thể có nhiều khóa ngoài, khóa ngoài có thể có giá trị NULL, giá

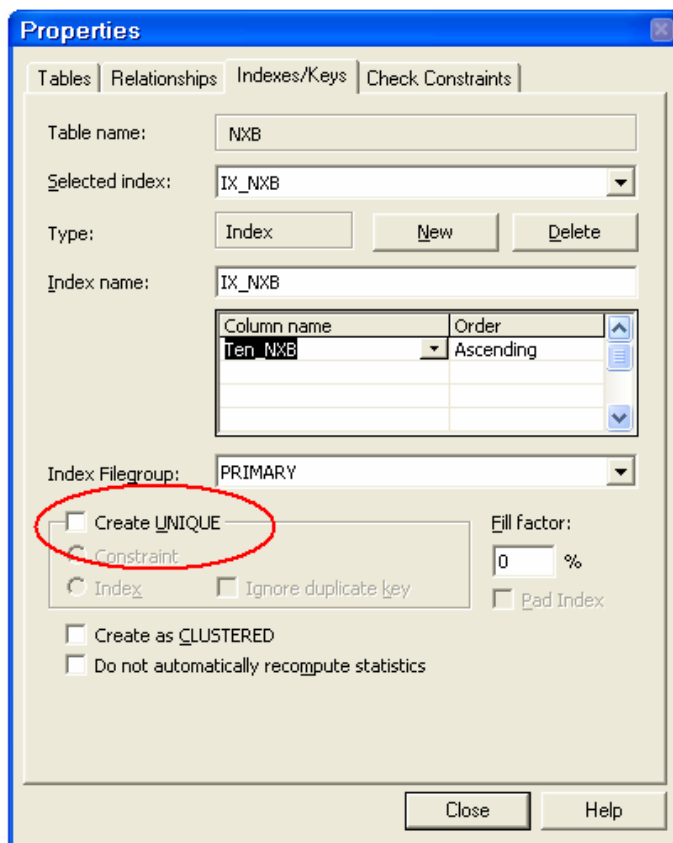
tr c a khóa ngoài luôn n m trong t p giá tr c a khóa chính trong m i quan h ã thi t l p.

Khóa ngoài và khóa chính ph i có cùng ki u d li u, cùng kích th c.



Ràng bu c Unique.

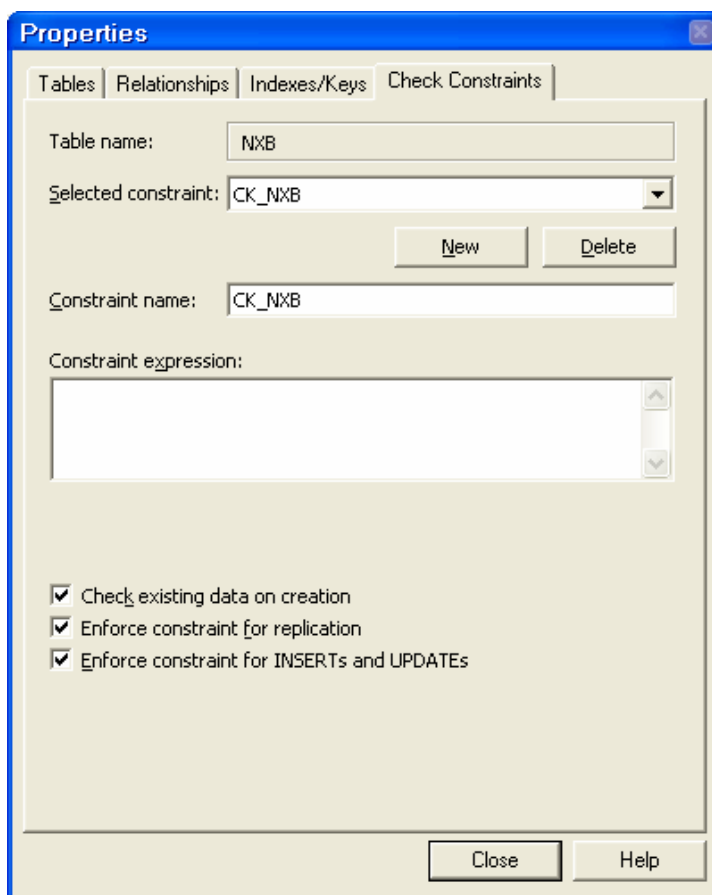
Unique là ràng bu c xác nh trên m t ho c t h p c t d li u, c t ho c t h p c t d li u c xác nh ràng bu c lo i này là duy nh t.



Một bảng dữ liệu có thể có nhiều ràng buộc duy nhất, một cột trong bảng này cho phép nhận giá trị NULL, ràng buộc duy nhất có thể sử dụng làm tham chiếu cho khóa ngoài.

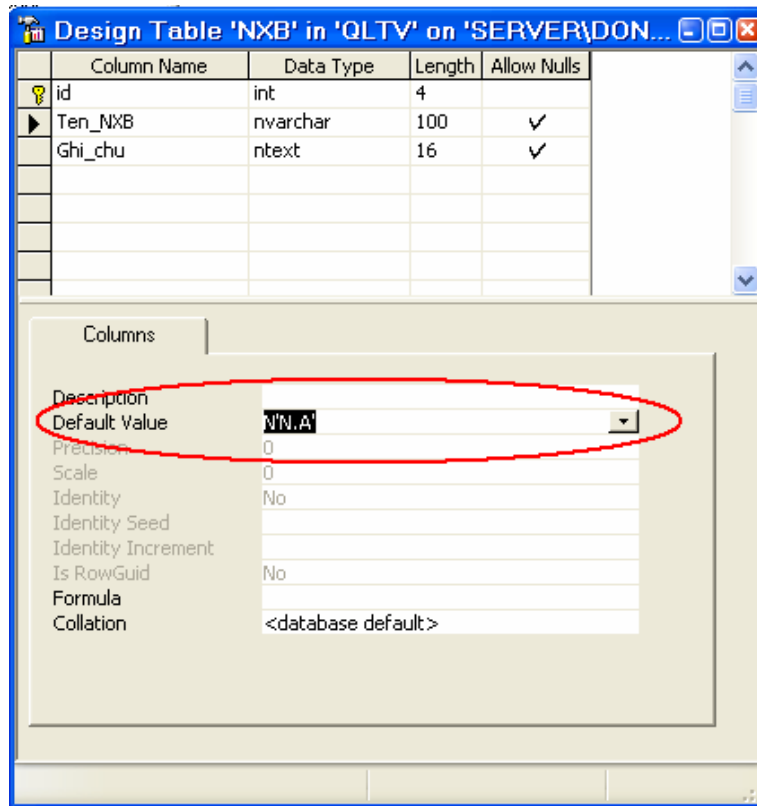
Ràng buộc Check.

Là ràng buộc kiểm tra dữ liệu nằm trong một phạm vi nào đó. Ràng buộc này sẽ kiểm tra dữ liệu khi nhập vào.



Giá trị mặc định – Default.

Giá trị gán cho cột dữ liệu khi thêm bản ghi và chọn dữ liệu vào cột này.



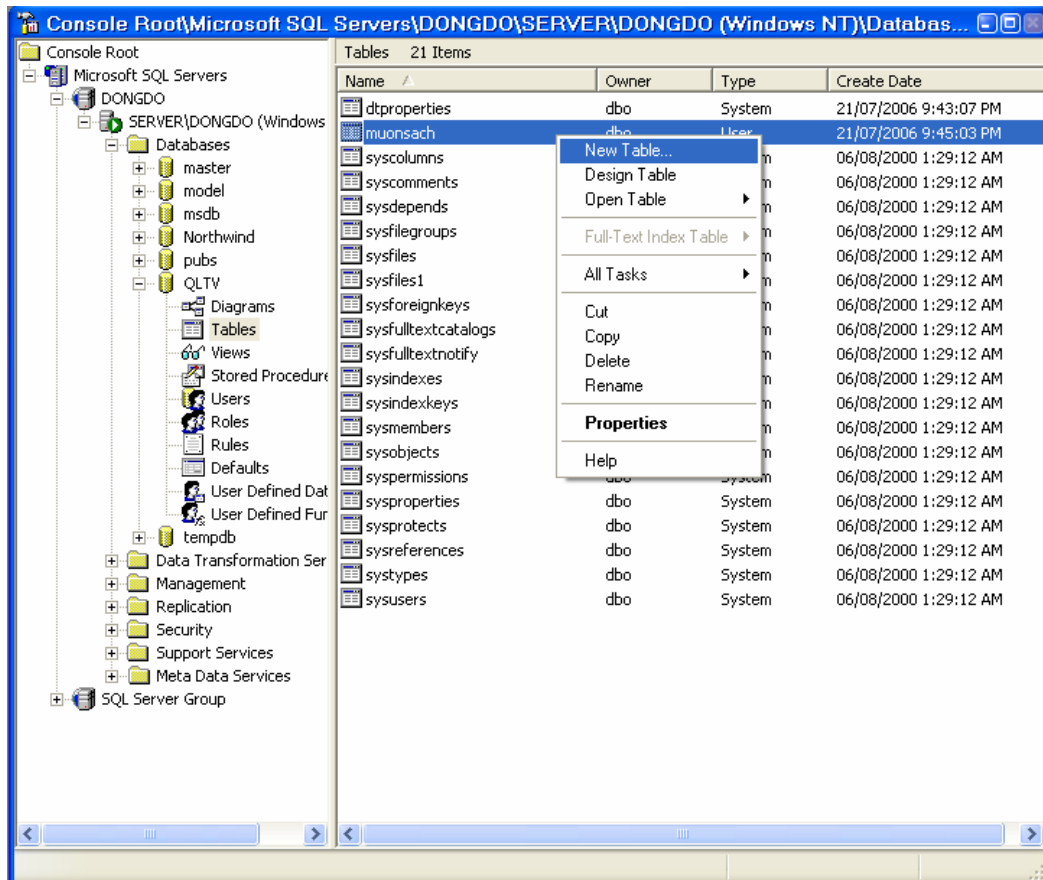
T O B N G D L I U .

Sau khi ã xác nh ÿ các thông tin thi t k CSDL, b c ti p theo là th c hi n t o c u trúc CSDL. t o c u trúc CSDL b c quan tr ng là t o b ng d li u.

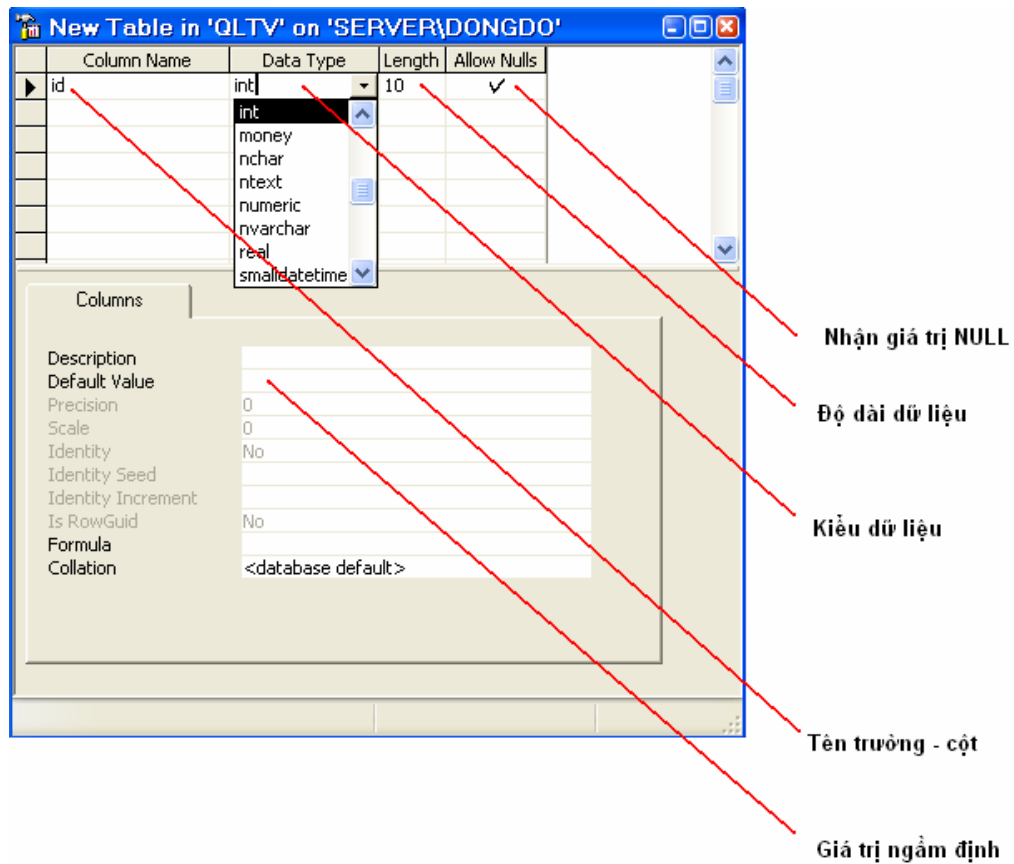
Khi t o CSDL h th ng t ng t o ra m t s b ng d li u ng m nh, các b ng d li u này s cung c p, qu n lý thông tin qu n tr c a CSDL, cung c p m t s hàm h th ng tr giúp b g i dùng.

T o b ng công c .

- Ch n CSDL
- Ch n Tables
- Nh n ph i chu t c a s bên ph i

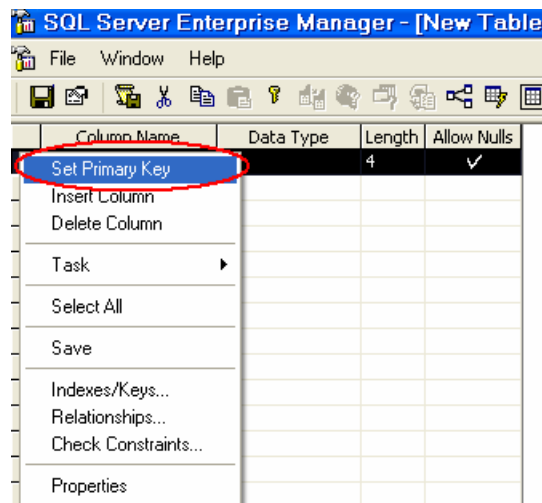


- Chọn New Table.



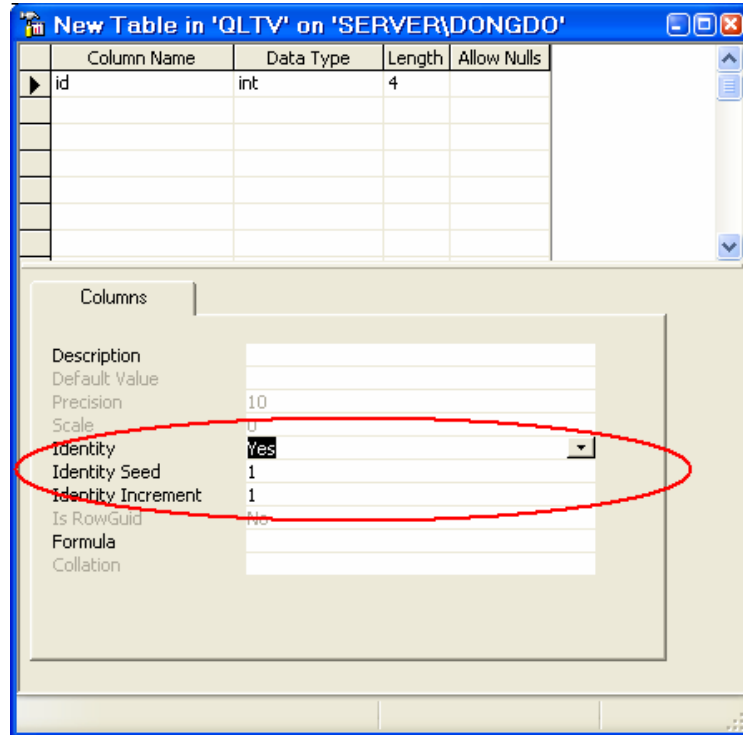
t khóa chính.

xác định khóa chính ta thực hiện chọn nhúng cột tham gia khóa bằng cách gõ phím shift và chọn chuột -> nhúng chuột phải -> chọn Set primary key.



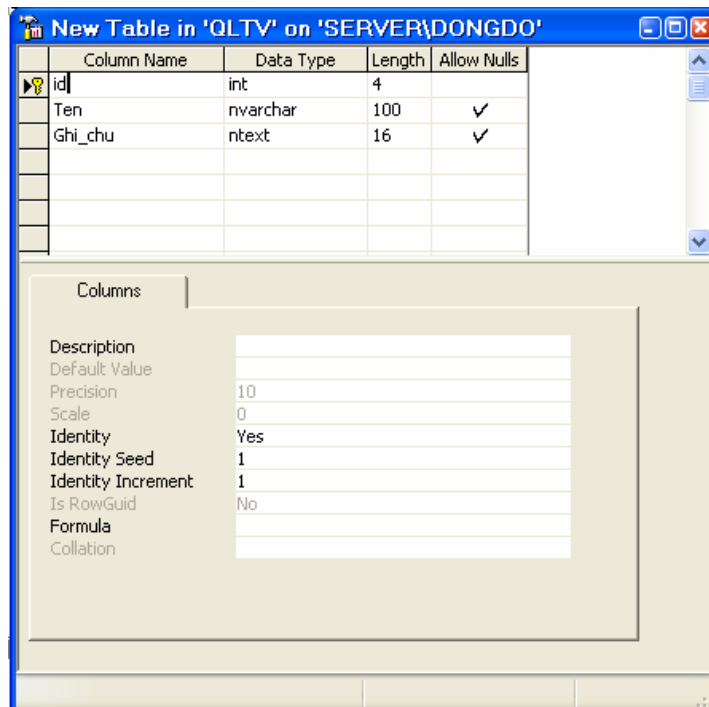
Xác định Identity.

- Chọn cột id -> Chọn yes trong mục Identity -> Identity seed (giá trị khởi đầu) -> Identity increment (bước tăng).



Tạo bảng bán hàng.

Giới thiệu bảng bán hàng tên NXB có cấu trúc như sau:



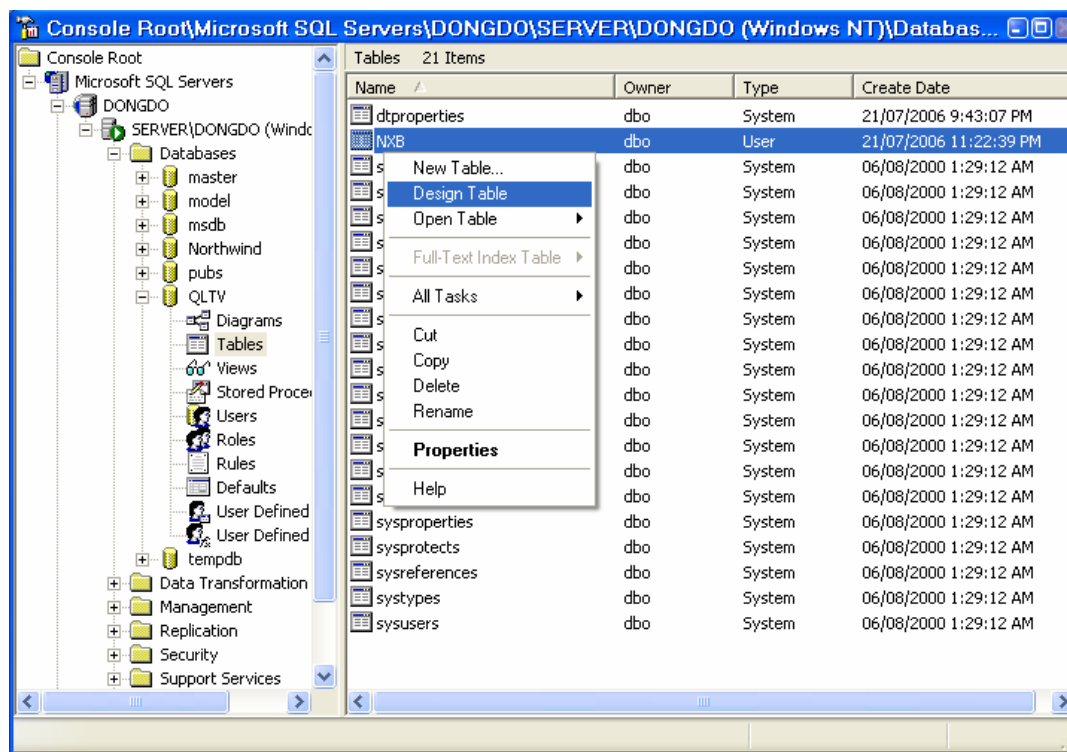
Sử dụng lệnh Create table, kich b n câu lệnh như sau:

```
Create table NXB(id int not null primary key identity(1,1), Ten Nvarchar(100), Ghi_chu Ntext)
```

S a c u trúc b ng.

Sử dụng công cụ .

- Chọn bảng cần sửa ở cửa sổ SQL Enterprise.
- Nhấn phải chuột -> chọn Design Table.



- Th c hi n s a c u trúc b ng.

S a d ng câu l nh.

s a c u trúc b ng d li u ta s d ng câu l nh Alter table.

- Thêm m t c t vào b ng ã có:

```
ALTER TABLE NXB ADD Dia_chi NVARCHAR(100) NULL
```

- Xóa c t t b ng ã có.

```
ALTER TABLE NXB_Drop column_Dia_chi
```

Xóa bảng.

Sử dụng công cụ .

- Chọn bảng
- Nhấn chuột phải
- Chọn Delete -> Yes.

Sử dụng lệnh. (Drop Table)

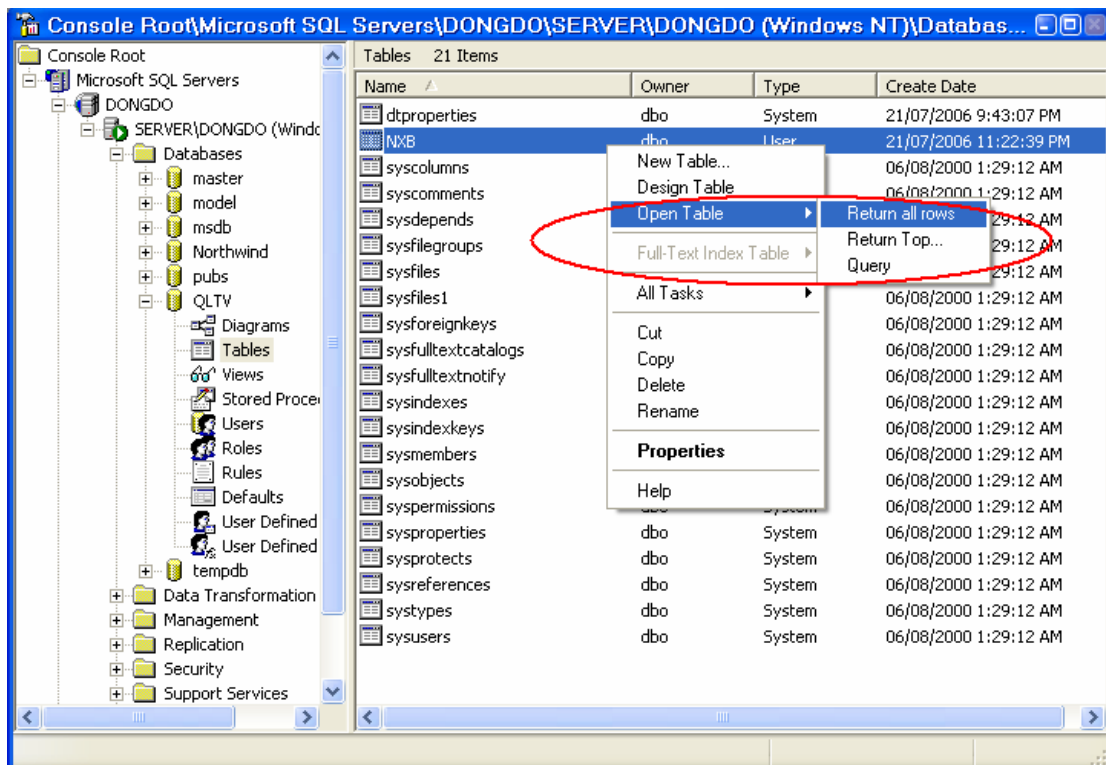
Drop Table NXB

Bảng dữ liệu có tham gia mối quan hệ Relationship khi xóa bảng cần chú ý: Nếu bảng chứa khóa ngoài thì việc xóa thành hiện bình thường, nếu bảng chứa khóa chính của mối quan hệ thì không xóa được.

Nhập dữ liệu vào bảng.

Sử dụng công cụ .

- Chọn bảng dữ liệu
- Nhấn chuột phải -> Open Table -> Return all rows



- Nhập dữ liệu theo đúng quy cách kiểu dữ liệu, ràng buộc dữ liệu.

id	Ten	Ghi_chu	dia_chi
1	Giáo dục	Thuộc Bộ giáo dục - Đào tạo	Trần Hưng Đạo - Hà Nội

Vì cơ sở dữ liệu, xóa các thành phần trình bày. Vì vậy các cột là dữ liệu, thông tin không cần nhập dữ liệu. Luôn luôn nhập dữ liệu để chuyển đổi sang hàng khác.

Sử dụng câu lệnh.

Sử dụng lệnh Insert into.

Insert into NXB(Ten, Dia_chi) values(N'Kim Hưng', N'hà Nội')

Nếu cần dữ liệu hỗ trợ Unicode thì trình bày giá trị nhập vào bên phải thêm kèm ký tự N (như ví dụ trên).

Tạo, sửa ràng buộc, khóa.

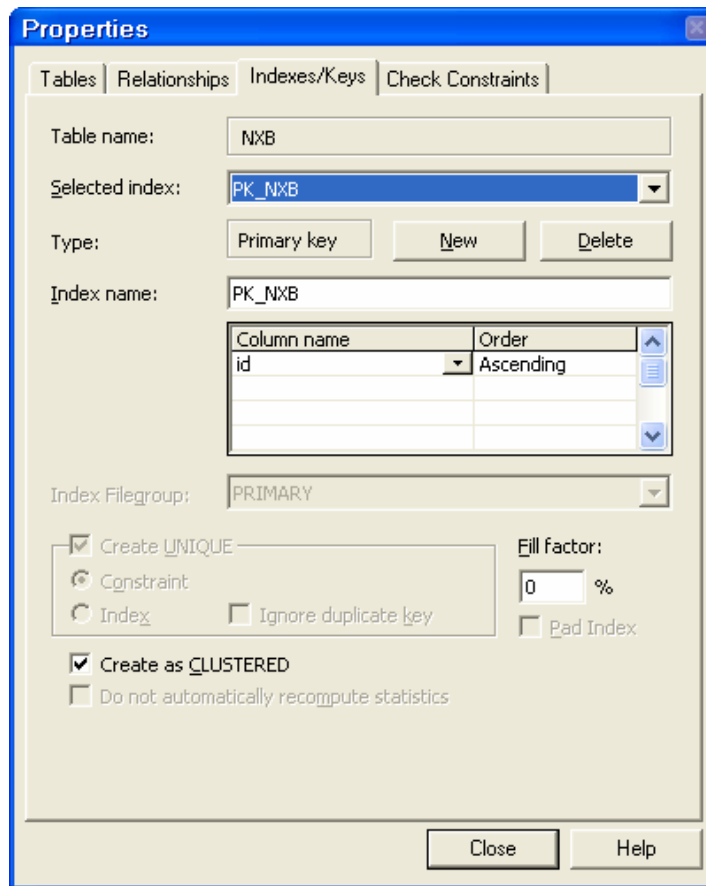
Phần này nhằm thể hiện thao tác với các ràng buộc, khóa: relationship, check, unique,...

Sử dụng công cụ.

- Chọn chức năng Design table.
- Chọn biểu tượng Manage Indexes/Keys...

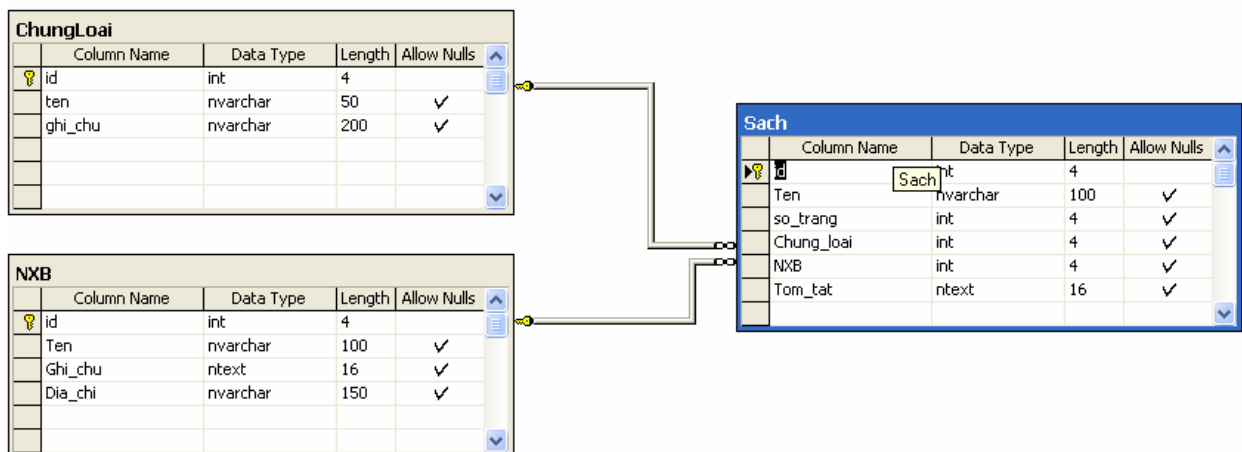
Column Name	Data Type	Length	Allow Nulls	Manage Indexes/Keys...
id	int	4		
Ten	nvarchar	100	✓	
Ghi_chu	ntext	16	✓	
dia_chi	nvarchar	100	✓	

- Ch n b ng t ng ng.



S d ng câu l nh.

c th h n ta th c hi n theo ví d có s c u trúc sau:




```
Create Table NXB(id int not null primary key  
identity(1,1), Ten Nvarchar(100), Ghi_chu Ntext,  
Dia_chi nvarchar(150))
```

Go

```
Create Table ChungLoai(id int not null primary key  
identity, ten nvarchar(50), ghi_chu nvarchar(200))
```

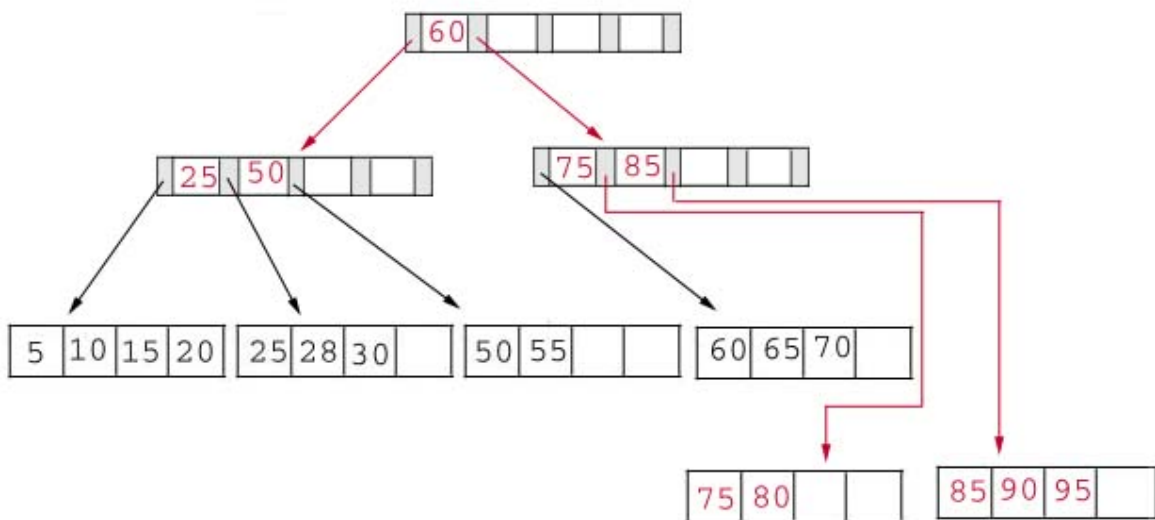
Go

```
Create Table Sach(id int not null primary key identity,  
Ten nvarchar(100), so_trang int default(0), Chung_loai  
int references Chungloai(id), NXB int references  
NXB(id), Tom_tat ntext)
```

KHÓA INDEX

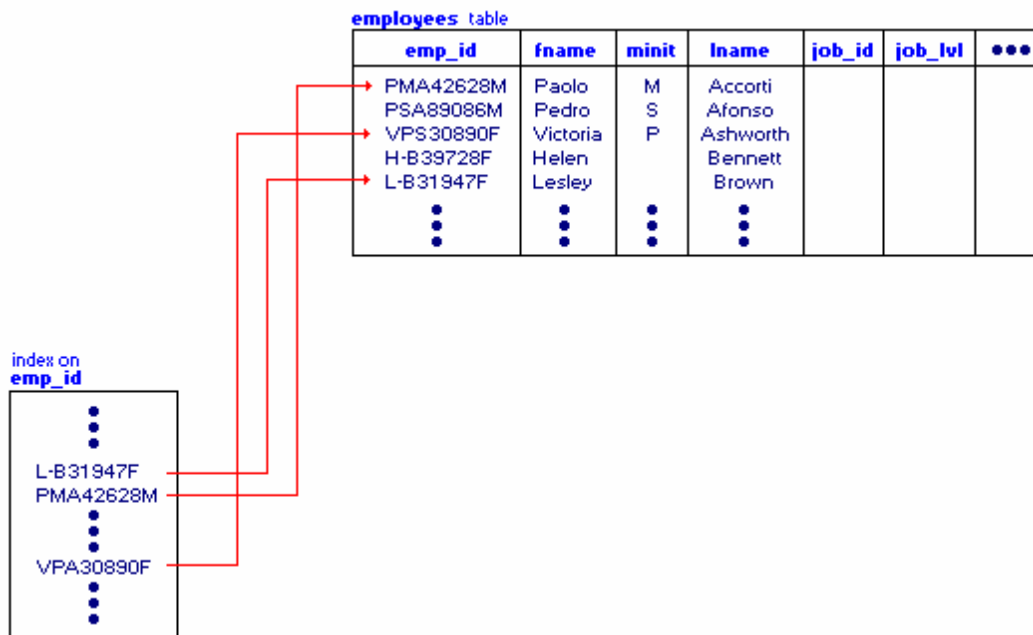
THIẾT KẾ KHÓA INDEX.

Index là một khóa quan trọng trong CSDL để tìm kiếm dữ liệu. Index có thể là một hoặc nhiều cột dữ liệu. Các giá trị của khóa Index sẽ được sắp xếp và lưu trữ theo một danh sách (bảng khác). Mỗi giá trị trong khóa Index là duy nhất trong danh sách, mỗi giá trị khóa Index sẽ liên kết đến giá trị trong bảng dữ liệu (liên kết đến con trỏ). Vì vậy, dữ liệu của bảng có khóa Index sẽ được tìm kiếm theo cấu trúc cây B-Tree nhằm tối ưu truy vấn dữ liệu.



Khi tìm kiếm một giá trị trong cột dữ liệu, mà cột này tham gia tạo khóa Index, đầu tiên câu lệnh xác định vị trí của giá trị nằm trong khóa Index bằng phép duyệt cây, sau đó thực hiện tìm theo liên kết đến bản ghi chứa giá trị tương ứng trong bảng.

S ví d d i ây g m khóa Index c t o t c t emp_id c a b ng employees.



Vi c thi t k khóa Index d a trên nhu c u truy v n, chèn d li u trên m t b ng, xác nh d a vào m t s tham s sau:

- + C t th ng c s d ng làm khóa truy v n d li u (xác nh c t tham gia khóa Index).
- + T p l nh th ng s d ng truy v n c n t c cao (xác nh t p c t tham gia truy v n).
- + D li u nh p vào b ng có khóa Index c n nhanh h n hay truy v n c n nhanh h n (xác nh t clustered ho c nonclustered).
- + L ng d li u nh p ng lo t nhi u hay ít (xác nh tham s fillfactor).

Clustered Index.

Khi khóa t thu c tính Clustered, d li u c a b ng s c s p x p v t lý trên a, nh v y khi thi t k khóa d ng này d li u c chèn và s tìm úng v trí trên a l u tr (vùng a dành cho b ng d li u), chính vì v y mà có th x y ra tr ng h p ph i d ch chuy n danh sách các giá tr ã có a. Nh ng vì c t o khóa Index d ng này s không c n s p x p giá tr d ng logic mà khi truy nh p a ã b o m d li u c s o x p.

B ng d li u ch có th t o t i a m t khóa Lustered Index.

Nonclustered Index.

Dữ liệu Index không sắp xếp đúng về mặt lý mà chỉ sắp xếp logic, dữ liệu cần bắt đầu lưu trữ giá trị khóa Index để sắp xếp, nhanh trong nhập dữ liệu.

Unique Index.

Xác định dữ liệu cần tham gia khóa Index không lặp lại.

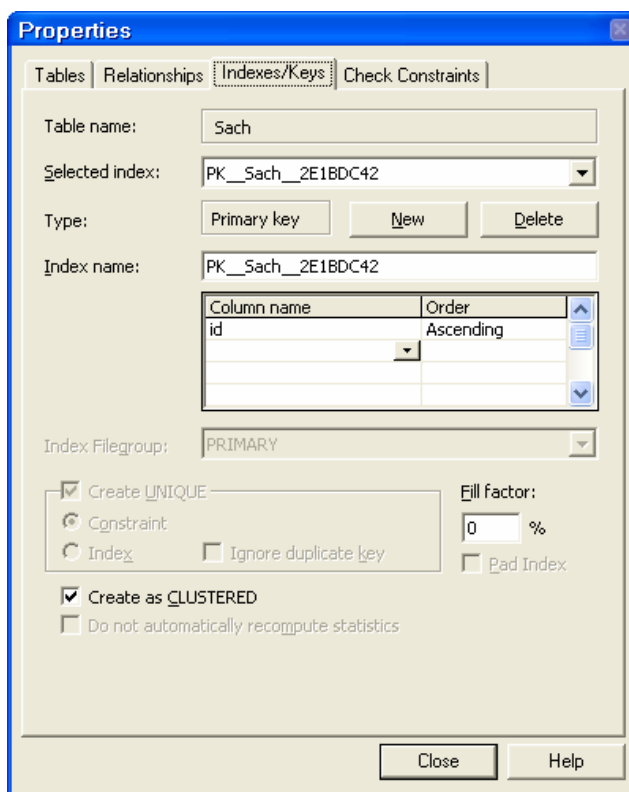
Fill Factor.

Khi tạo khóa Index, dữ liệu tham gia tạo khóa Index sẽ phân theo mức của B-Tree, các mức phân theo page dữ liệu, giá trị Fill factor xác định phần không trống của page theo từng phần trăm. Nếu không trống này mà tạo bố trí cấu trúc Index, tốc độ truy cập thông tin trong cây sẽ chậm.

TẠO KHÓA INDEX.

Tạo theo công cụ.

- Chọn chức năng Design table
- Vào bảng Index manager.



- New

- Chọn các cột tham gia tạo khóa Index
- Các tham số .

Tạo theo câu lệnh.

- Sử dụng trong câu lệnh Create Table, Alter Table.
- Sử dụng lệnh Create Index.

```
CREATE [ UNIQUE ] [ CLUSTERED | NONCLUSTERED ] INDEX index_name  
ON { table | view } ( column [ ASC | DESC ] [ ,...n ] )
```

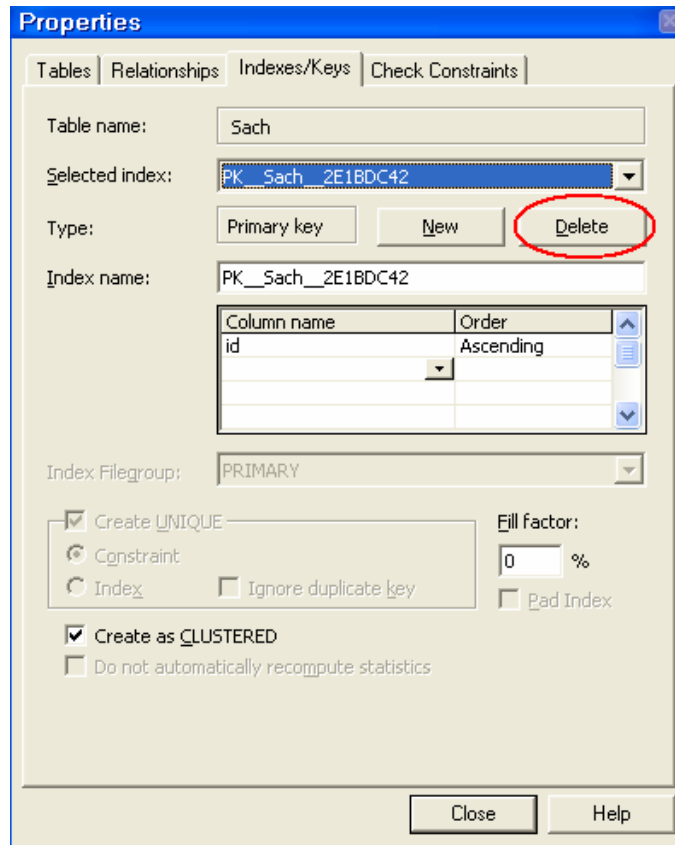
Ví dụ :

```
CREATE INDEX sach_idx ON sach (id)
```

XÓA INDEX.

Sử dụng công cụ .

- Vào Index amnager
- Chọn khóa Index -> Delete



Sử dụng câu lệnh.

Sử dụng lệnh Drop Index.

Drop Index Sach(sach_idx)

KHUNG NHÌN – VIEW

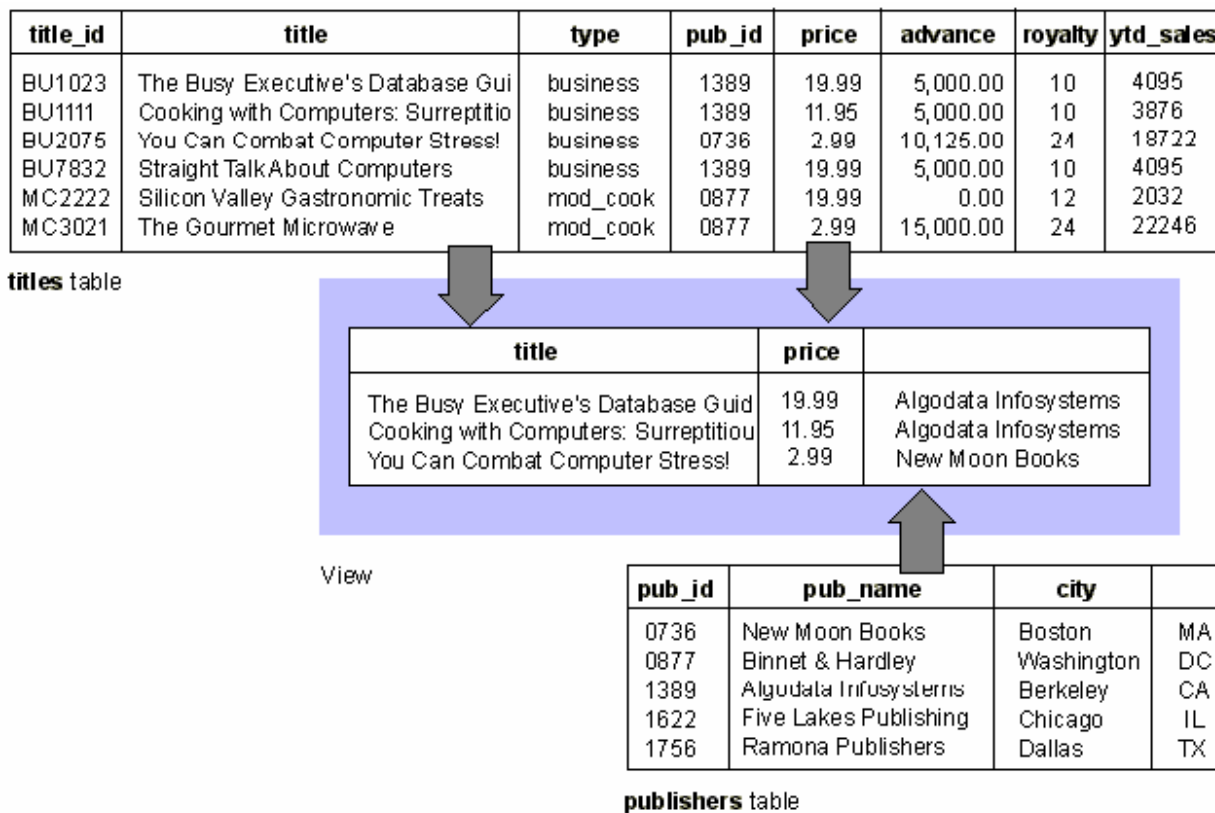
KHÁI NIỆM KHUNG NHÌN.

Khung nhìn (View) là một bảng tạm thời, có cấu trúc như một bảng, khung nhìn không lưu trữ dữ liệu mà nó chỉ tạo ra khi sử dụng, khung nhìn là một công cụ CSDL.

Khung nhìn chỉ tạo ra các câu lệnh truy vấn dữ liệu (lệnh Select), truy vấn tạm thời hoặc chỉ để lưu trữ dữ liệu.

Khung nhìn chỉ sử dụng khai thác dữ liệu như một bảng dữ liệu, chia sẻ nhiều người dùng, an toàn trong khai thác, không ảnh hưởng dữ liệu gốc.

Có thể chỉ định truy vấn dữ liệu trên cấu trúc của khung nhìn.

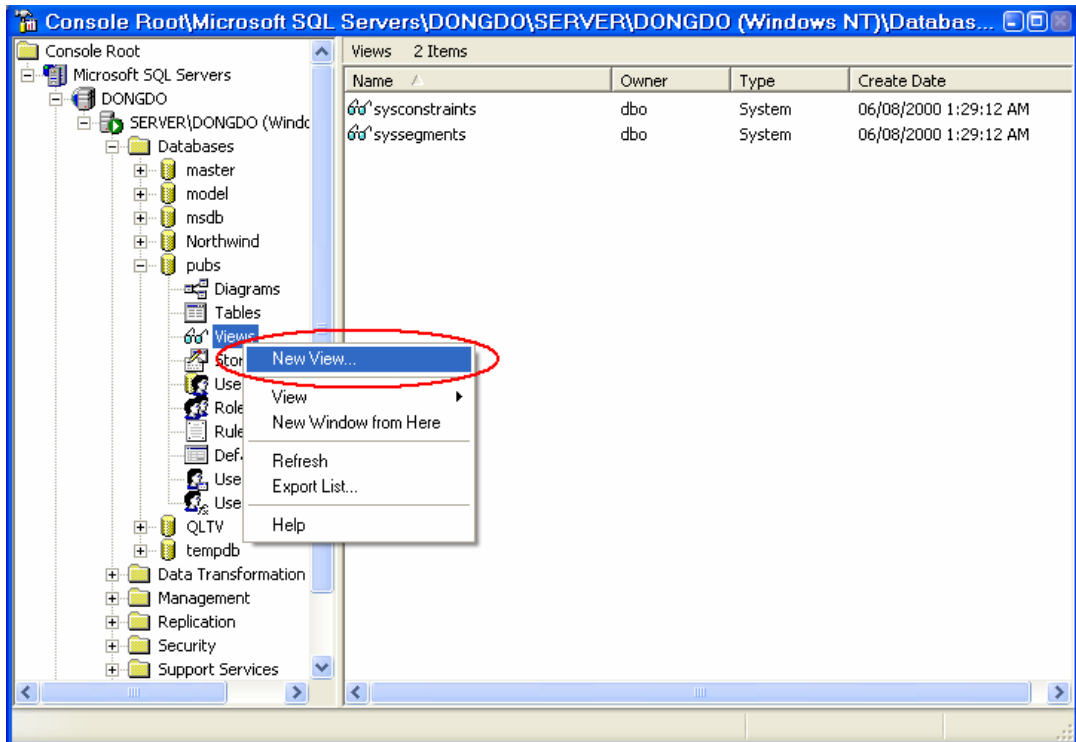


TỔNG KẾT KHUNG NHÌN.

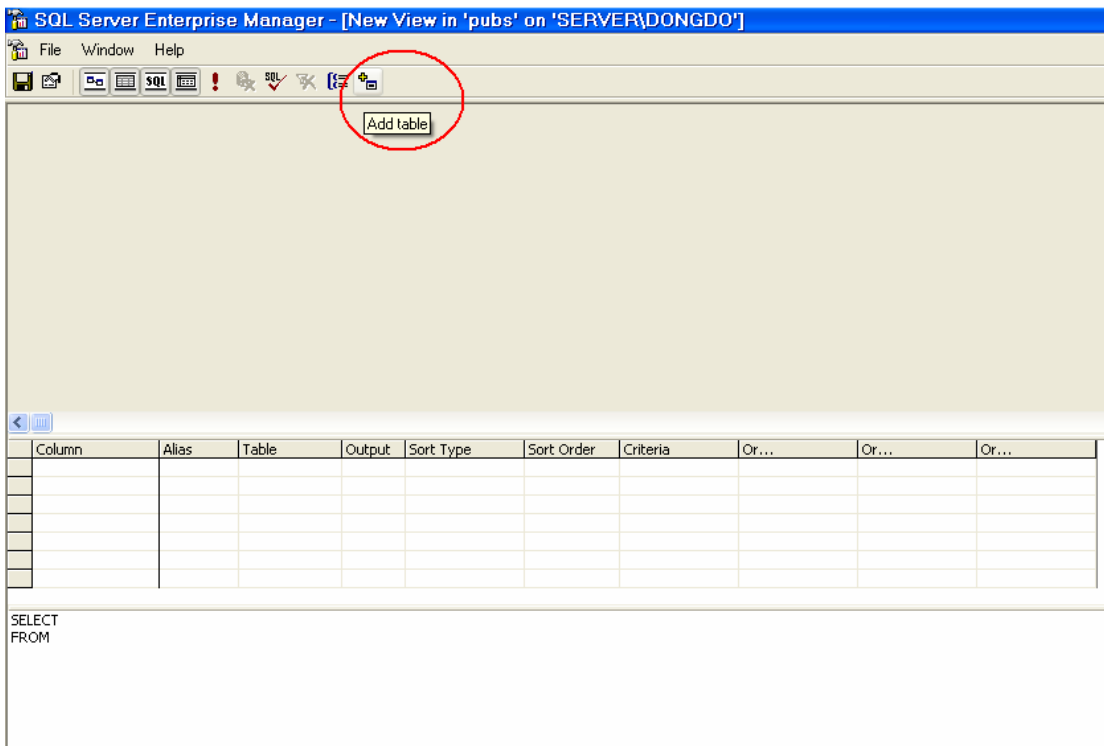
Sử dụng công cụ .

- Chọn chức năng Views của CSDL.

- Nhấn phím chuột.

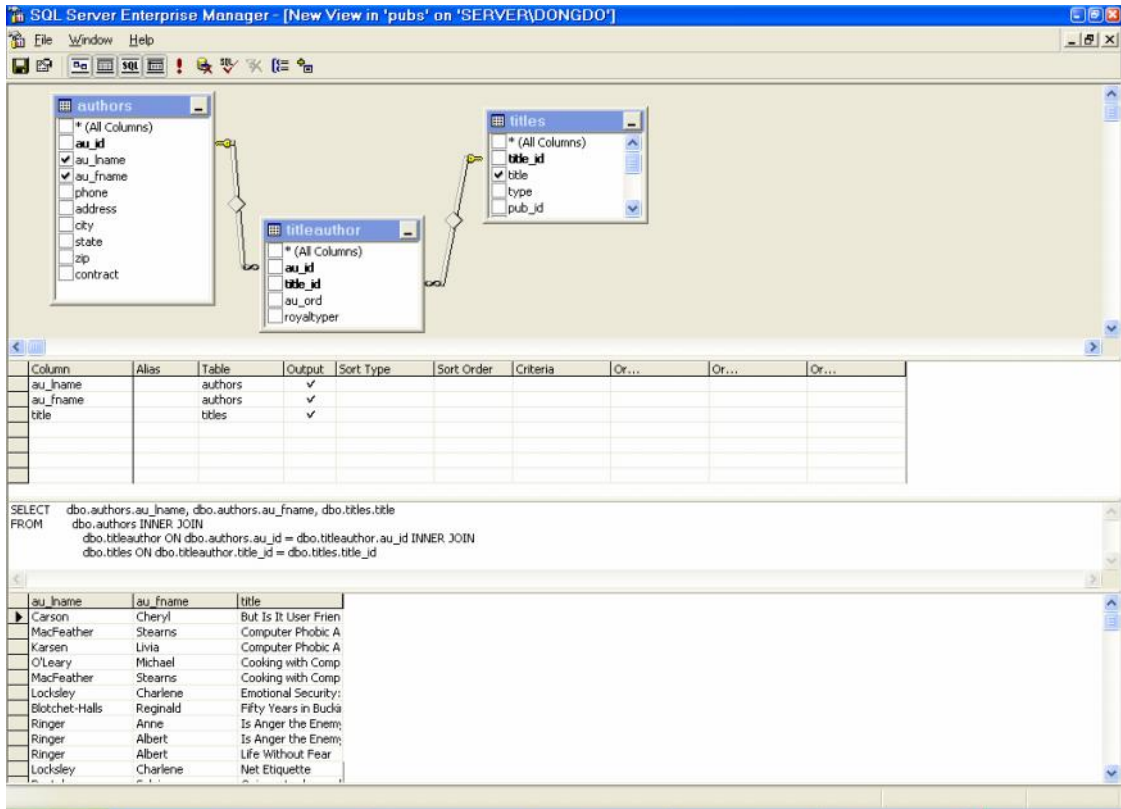


- Chọn New View.



- Thêm các bảng tham gia câu lệnh truy vấn để liệt kê cho View

- So n l nh truy v n ho c ánh d u các c t tham gia t o View.



- S a i l nh Select theo ý mu n.
- Ghi k ch b n -> t tên view.

T o theo câu l nh.

S d ng l nh Create View:

CREATE VIEW VIDU as

SELECT dbo.authors.au_lname, dbo.authors.au_fname, dbo.authors.title

FROM dbo.authors INNER JOIN

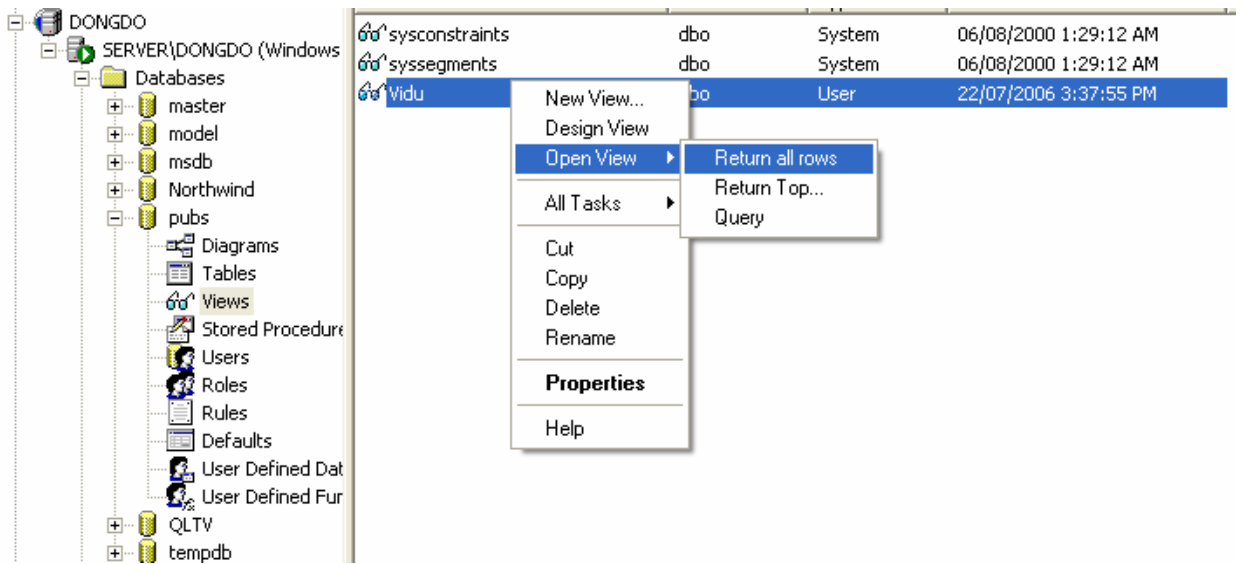
dbo.titleauthor ON dbo.authors.au_id = dbo.titleauthor.au_id INNER

JOIN

dbo.authors ON dbo.titleauthor.title_id = dbo.authors.title_id

S D NG VIEW.

- Ch n View
- Nh n nút ph i chu t.



Th c hi n các ch c n ng t ng t table.

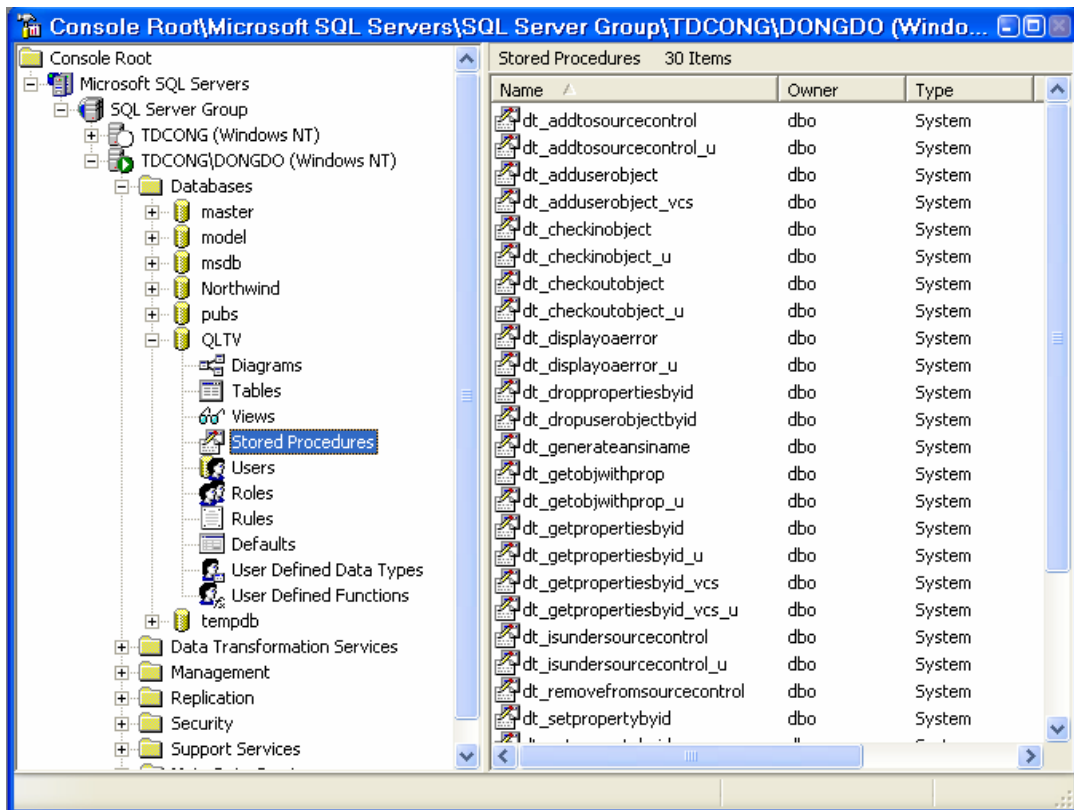
TH T C L U T R

KHÁI NI M TH T C L U T R VÀ HÀM.

Th t c l u t r có thu t ng Stored Procedure, là m t i t ng c a CSDL t ng t nh khung nhìn, th t c l u t r có th t o r a t công c và câu l nh. TH t c c th c hi n nh câu l nh (có th th c hi n t SQL Query analyzer, các v trí g i câu l nh T-SQL).

Th t c l u t r c k t c u t m t k ch b n câu l nh T-SQL, th t c có nh ng c i m c b n sau:

- + Truy n tham s .
- + G i th t c khác.
- + Tr v các giá tr tham s , chuy n giá tr tham s cho các th t c c g i.
- + Tr v giá tr tr ng thái th t c là thành công hay không thành công.



Th t c l u tr có nhi u u i m s o v i th c h i n c â u l n h T-SQL t các máy khách:

+ L p trình theo module: Th t c c thi t l p trong t ng CSDL m t l n, có th g i th c h i n n h i u l n trong m t ng d ng, có th g i t n h i u ng d ng.

+ Th c h i n n h a n h h n: Khi c n th c h i n m t l ng l n c â u l n h T-SQL, th t c l u tr th c h i n n h a n h h n v i khi máy ch nh n c n h i u c u l n h cùng m t lúc u p h i k i m t r a t í n h p l quy n c a tài kho n t máy khách và các tham s khác. Khi th t c c n g i n h i u l n trên các máy khách thì th t c th c h i n m t l n u tiên, nh ng l n sau máy khách s ch y th t c ã c biên d ch.

+ Làm gi m l u l ng trên m ng: Thay cho v i máy khách p h i g i n h i u dòng l n h t các ng d ng n máy ch , khi s d ng th t c thì nó ch c n g i m t l n h, t ó d n n l u l ng thông tin l n h truy n qua m ng gi m.

+ An ninh b o m t h n: Khi không mu n cho m t user tr c t i p khai thác m t i t ng hay b ng d li u nào ó, mà c n cho user ó c khai thác thì th t c có th giúp b n g á n quy n khai thác cho ng i ó. V i c g á n quy n khai thác nh nói trên s giúp cho v n an ninh b o m t trong CSDL t t h n.

PHÂN LO I TH T C L U TR .

Th t c l u tr c phân thành 5 lo i nh sau:

System Stored Procedure.

Là th t c c l u tr t ng CSDL Master, th t c lo i này c b t u b ng ch sp_ th t c lo i này th ng c s d ng trong qu n tr CSDL và an ninh b o m t.

Ví d : Mu n bi t t t c các t i n trình ang th c h i n b i user nào:

sp_who @loginame='sa'

K t qu :



spid	ecid	status	loginame	hostname	blk	dbname	cmd
1	1	0 background	sa		0	NULL	LAZY WRITER
2	2	0 sleeping	sa		0	NULL	LOG WRITER
3	3	0 background	sa		0	master	SIGNAL HANDLER
4	4	0 background	sa		0	NULL	LOCK MONITOR
5	5	0 background	sa		0	master	TASK MANAGER
6	6	0 background	sa		0	master	TASK MANAGER
7	7	0 sleeping	sa		0	NULL	CHECKPOINT SLEE
8	8	0 background	sa		0	master	TASK MANAGER
9	9	0 background	sa		0	master	TASK MANAGER

Local Stored Procedure.

ây là lo i th t c th ã ng dùng nh t, n m trong CSDL do ng i dùng t o ra, th c hi n m t công vi c nào ó. Th t c lo i này th ã ng c t o b i DBA (Database Administrator) ho c ng i l p trình.

Temporary Stored Procedure.

Có ch c n ng t ã ng t nh Local Stored Procedure nh ã ng th t c lo i này t h y khi k t n i t o ra nó ã ng t ho c SQL Server ã ng ng ho t ã ng và nó c t o ra trên CSDL TempDB.

Extended Stored Procedure.

ây là lo i th t c s ã d ã ng ch ã ng trình ngo i vì ã c biên d ch thành DLL. Tên th t c c b t u b ã ng xp_. Ví d th t c xp_sendmail dùng g i mail, th t c xp_cmdshell dùng th c hi n l ã nh c a DOS (xp_cmdshell 'dir c:\').

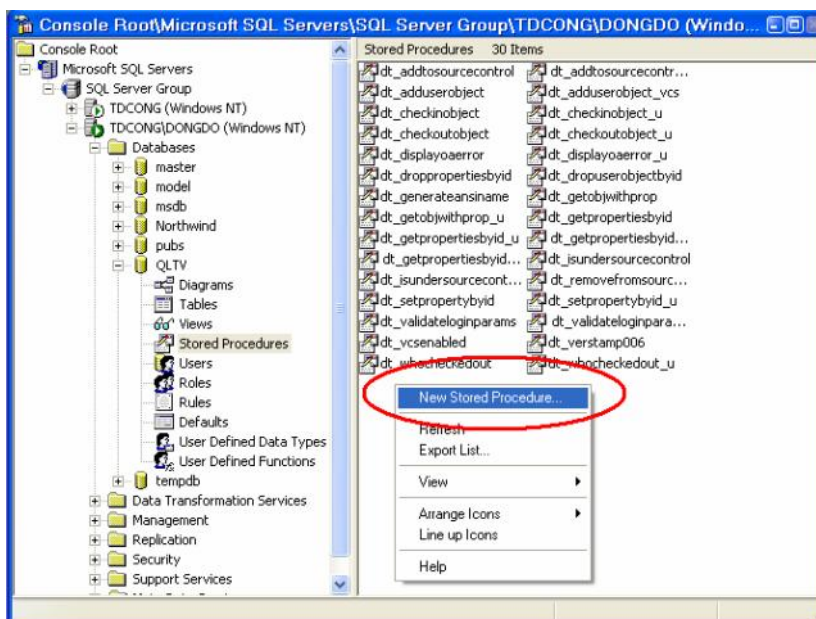
Remote Stored Procedure:

Là lo i th t c s ã d ã ng th t c c a m t server khác.

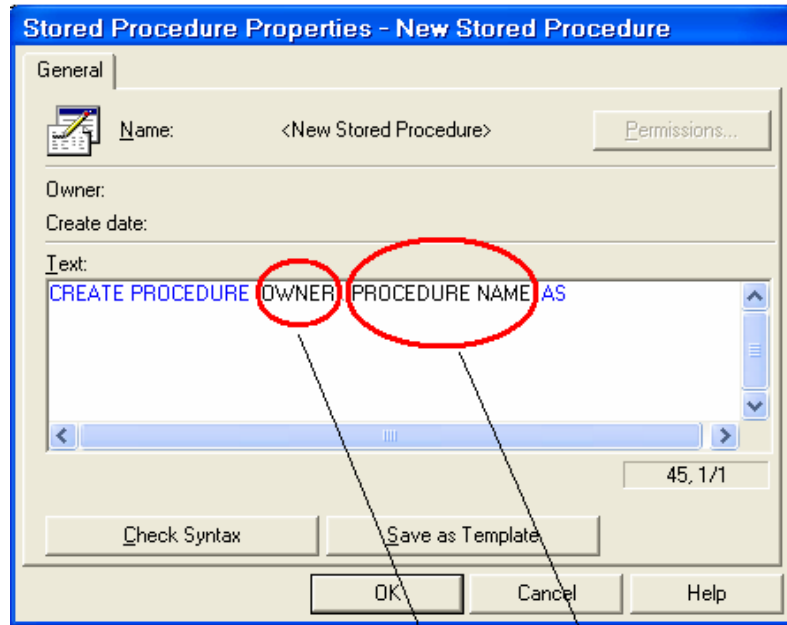
THI T L P TH T C L U T R .

S ã d ã ng công c .

- Ch ã n CSDL c ã n t o th t c trong Enterprise Manager -> Stored Procedures
- Nh ã n nút ph i chu t -> New Stored Procedure...



- t tên th t c, xác nh role ng i khai thác, so n k ch b n câu l nh.



Tên thủ tục

Role xác định quyền người dùng

S d ng câu l nh.

S d ng l nh Create Procedure, ti n xem xét ta xét theo các ví d , các ví d d i ây th c hi n t o th t c và thao tác v i CSDL pubs ti n trong d li u m u, tìm hi u cú pháp câu l nh T-SQL b n xem ph n câu l nh T-SQL trong cùng tài li u này.

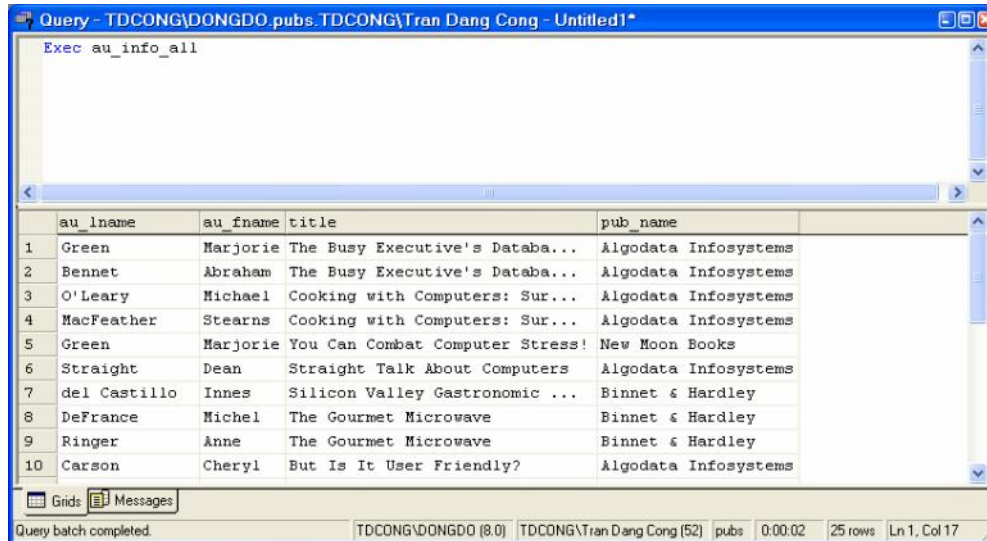
Th t c không có tham s .

Th t c sau s th c hi n li t kê t t c các tác gi , sách và nhà xu t b n mà tác gi vi t sách.

Use Pubs

```
CREATE PROCEDURE au_info_all
AS
SELECT au_lname, au_fname, title, pub_name
FROM authors a INNER JOIN titleauthor ta
ON a.au_id = ta.au_id INNER JOIN titles t
ON t.title_id = ta.title_id INNER JOIN publishers p
ON t.pub_id = p.pub_id
GO
```

K t qu th c hi n:



	au_lname	au_fname	title	pub_name
1	Green	Marjorie	The Busy Executive's Databa...	Algodata Infosystems
2	Bennet	Abraham	The Busy Executive's Databa...	Algodata Infosystems
3	O'Leary	Michael	Cooking with Computers: Sur...	Algodata Infosystems
4	MacFeather	Stearns	Cooking with Computers: Sur...	Algodata Infosystems
5	Green	Marjorie	You Can Combat Computer Stress!	New Moon Books
6	Straight	Dean	Straight Talk About Computers	Algodata Infosystems
7	del Castillo	Innes	Silicon Valley Gastronomic ...	Binnet & Hardley
8	DeFrance	Michel	The Gourmet Microwave	Binnet & Hardley
9	Ringer	Anne	The Gourmet Microwave	Binnet & Hardley
10	Carson	Cheryl	But Is It User Friendly?	Algodata Infosystems

Th t c có tham s .

Th t c sau th c hi n l c tìm tác gi có tên, h truy n theo tham s .

```
USE pubs
```

```
GO
```

```
CREATE PROCEDURE au_info
```

```
    @lastname varchar(40),
```

```
    @firstname varchar(20)
```

```
AS
```

```
SELECT au_lname, au_fname, title, pub_name
```

```
FROM authors a INNER JOIN titleauthor ta
```

```
    ON a.au_id = ta.au_id INNER JOIN titles t
```

```
    ON t.title_id = ta.title_id INNER JOIN
```

```
publishers p
```

```
    ON t.pub_id = p.pub_id
```

```
WHERE au_fname = @firstname
```

```
    AND au_lname = @lastname
```

```
GO
```

Cách truy n tham s :

+ Gán giá tr theo th t :

```
EXECUTE au_info 'Dull', 'Ann'
```

+ Gán giá tr theo tên bi n

```
EXECUTE au_info @lastname = 'Dull', @firstname =  
'Ann'
```

+ Gán giá tr theo tên bi n, không theo th t

```
EXECUTE au_info @firstname = 'Ann', @lastname = 'Dull'
```

Th t c có tham s tùy l a theo giá tr a vào.

Ví d này s c p n vi c truy n tham s theo m u, giá tr tham s c ng m nh khi t o th t c và th t c khi th c hi n s ki m tra giá tr tham s nh p vào.

```
USE pubs
```

```
GO
```

```
CREATE PROCEDURE au_info2
```

```
    @lastname varchar(30) = 'D%',
```

```
    @firstname varchar(18) = '%'
```

```
AS
```

```
SELECT au_lname, au_fname, title, pub_name
```

```
FROM authors a INNER JOIN titleauthor ta
```

```
    ON a.au_id = ta.au_id INNER JOIN titles t
```

```
    ON t.title_id = ta.title_id INNER JOIN publishers p
```

```
    ON t.pub_id = p.pub_id
```

```
WHERE au_fname LIKE @firstname
```

```
    AND au_lname LIKE @lastname
```

```
GO
```

Tham s % xác nh giá tr tùy ý nh p vào tham s , tham s D% xác nh giá tr u tiên c a chu i ph i b ng ch D. Khi ng m nh các giá tr nh trên tham có không c truy n giá tr s t nh n giá tr ng m nh.

Cách truy n tham s nh sau:

+ Không truy n tham s :

```
EXECUTE au_info2
```



```

+ Ch  truy n tham s      u, tham s      sau s      nh n giá tr
ng m      nh.
    EXECUTE au_info2 'Wh%'
+ Ch  truy n m t tham s , tham s      xò n l i s      nh n giá
tr ng m      nh.
    EXECUTE au_info2 @firstname = 'A%'
+ Tham s      th      nh t xác      nh giá tr      m t ký t      thu c v
trí có [CK] ch      nh n ký t      'C' ho c 'K', [OE] ch      nh n
giá tr      'O' ho c 'E'.
    EXECUTE au_info2 '[CK]ars[OE]n'
+ Xác      nh rõ giá tr      tham s
    EXECUTE au_info2 'Hunter', 'Sheryl'
+ Xác      nh ki u giá tr      tham s .
    EXECUTE au_info2 'H%', 'S%'

```

Th t c s d ng tham số lấy giá tr ra (tham tr).

Ví d sau s mô t k thu t s d ng tham tr , nh trong các ví d tr c ta s d ng tham s truy n giá tr vào tên tham s b t u b ng l ch @, tham s c b t u b ng 2 ch @@. S c s d ng trên nhi u dòng l nh, s d ng cùng t khóa OUTPUT xác nh là tham tr l y giá tr ra.

Ví d sau th c hi n truy n tham s vào và l y giá tr ra:

```

USE pubs
GO
CREATE PROCEDURE titles_sum @@TITLE varchar(40) = '%',
@@SUM money OUTPUT
AS
SELECT 'Title Name' = title
FROM titles
WHERE title LIKE @@TITLE
SELECT @@SUM = SUM(price)
FROM titles
WHERE title LIKE @@TITLE
GO

```

Tham s b t u b ng 2 ký t @@ xác nh c s d ng cho nhi u câu l nh, s d ng cùng t khóa OUTPUT xác nh là bi n tham tr .. Ví d trên s d ng bi n @@Title xác nh i u ki n a ra tên sách ây là lo i bi n truy n vào s

dùng cho hai câu lệnh Select, biến @@Sum xác định là biến tham tr dùng l y giá tr ra.

Cách sử dụng tham số như sau:

```
DECLARE @@TOTALCOST money
EXECUTE titles_sum 'The%', @@TOTALCOST OUTPUT
IF @@TOTALCOST < 200
BEGIN
    PRINT ' '
    PRINT 'All of these titles can be purchased for less
than $200.'
END
ELSE
    SELECT 'The total cost of these titles is $'
        + RTRIM(CAST(@@TOTALCOST AS varchar(20)))
```

Ví dụ trên sử dụng biến @@TOTALCOST vào vị trí biến @@SUM trong th t c. Kết quả th c hiện như sau:

Title Name

The Busy Executive's Database Guide
The Gourmet Microwave
The Psychology of Computer Cooking

(3 row(s) affected)

Warning, null value eliminated from aggregate.

All of these titles can be purchased for less than \$200.

Th t c sử dụng biến OUTPUT khi u con tr (Cursor).

Ví dụ sau là th t c có biến khi u Cursor, biến này sẽ quản lý m t b ng d li u truy v n b ng câu l nh Select.

```
CREATE PROCEDURE titles_cursor @titles_cursor CURSOR
VARYING OUTPUT
```

```

AS
SET @titles_cursor = CURSOR
FORWARD_ONLY STATIC FOR
SELECT *
FROM titles

```

```

OPEN @titles_cursor
GO

```

Con trỏ của vào biến kiểu Cursor có tên @Titles_cursor, hướng di chuyển Forward (tiền) và Static. Số dòng biến như ví dụ sau:

```

USE pubs
GO
DECLARE @MyCursor CURSOR
EXEC titles_cursor @titles_cursor = @MyCursor OUTPUT
WHILE (@@FETCH_STATUS = 0)
BEGIN
    FETCH NEXT FROM @MyCursor
END
CLOSE @MyCursor
DEALLOCATE @MyCursor
GO

```

Biến con trỏ của vào biến @MyCursor, khi mở con trỏ vị trí bắt đầu tiên của bảng xác định. Trong ví dụ trên số dòng vòng lặp duy nhất bảng, vì các lý do liên quan đến hiệu suất trong vòng lặp.

Thật ra thì tính năng này không phải là câu lệnh.

Ví dụ sau sử dụng tính năng WITH ENCRYPTION để ẩn mã nguồn trong thủ tục vring là đúng.

```

CREATE PROCEDURE encrypt_this
WITH ENCRYPTION
AS
SELECT *
FROM authors

```

GO

Khi s d ng th t c h th ng sp_helptext xem n i dung th t c:

```
EXEC sp_helptext encrypt_this
```

K t qu nh sau:

The object's comments have been encrypted.

S A, XÓA TH T C

S d ng công c .

- Ch n th t c c n s a, xóa -> th c hi n s a n i dung ho c ch c n ng xóa.

S d ng câu l nh.

- S a s d ng l nh Alter Procedure
- Xóa s d ng l nh Drop Procedure

B n c có th t tìm hi u v User Defined Function tong Book Online, là i t ng g i là hàm thu c CSDL, có ch c n ng và cách th c ho t ng g n gi ng th t c.

TRIGGER

KHÁI NI M TRIGGER.

Trigger là m t th t c c bi t mà vi c th c thi c a nó t ng khi có s ki n x y ra, các s ki n g i th t c c bi t này c nh ngh a trong câu l nh, thông th ng c th c hi n v i các s ki n liên quan n Insert, Update, Delete d li u.

Trigger c s d ng trong vi c b o m toàn v n d li u theo quy t c xác nh, c qu n lý theo b ng d li u ho c khung nhìn.

NH NG TR NG H P S D NG TRIGGER.

- S d ng Trigger khi các bi n pháp toàn v n d li u nh Constraint, rule,... không b o m. Khác v i các công c bao m toàn v n d li u ã nêu, các công c này s th c hi n ki m tra tính toán v n tr c khi ã d li u vào CSDL (còn g i là Declarative Data Integrity), còn Trigger th c hi n ki m tra tính toán v n khi công vi c ã th c hi n r i (còn g i là Procedural Data Integrity).

- Khi CSDL ch a c chu n hóa (Normalization) thì có th x y ra d li u th a, ch a nhi u v trí trong CSDL thì yêu c u t ra là d li u c n c p nh t th ng nh t trong m i n i. Trong tr ng h p này ta ph i s d ng Trigger.

- Khi thay i day chuy n d li u gi a các b ng v i nhau (khi d li u b ng này thay i thì d li u trong b ng khác c ng c thay i theo).

C I M C A TRIGGER.

- M t trigger có th th c hi n nhi u công vi c (theo k ch b n), có th nhi u s ki n kích ho t th c thi trigger, có th tách r i các s ki n trong m t trigger.

- Trigger không c t o trên b ng temprate hay system.

- Trigger ch th c thi t ng thông qua các s ki n mà không th c hi n b ng tay.

- Trigger s d ng c v i khung nhìn.

- Khi trigger th c thi theo các s ki n Insert ho c Delete thì d li u khi thay i s c chuy n sang các b ng Inserted Table, Deleted Tabla, là 2 b ng t m th i ch ch a trong b nh , các b ng này ch c s d ng v i các l nh trong trigger.

Các b ng này th ng c s d ng khôi ph c l i ph n d li u ã thay i (roll back).

- Trigger chia thành 2 lo i Instead of và After: Instead of là lo i trigger mà ho t ng c a s ki n g i nó s b qua và thay vào nó là các l nh th c hi n trong trigger. After (t ng ng v i t khóa For) ây là lo i ng m nh, khác v i lo i Instead of thì lo i trigger này s th c hi n các l nh trong nó sau khi ã th c hi n xong s ki n g i nó.

T O TRIGGER.

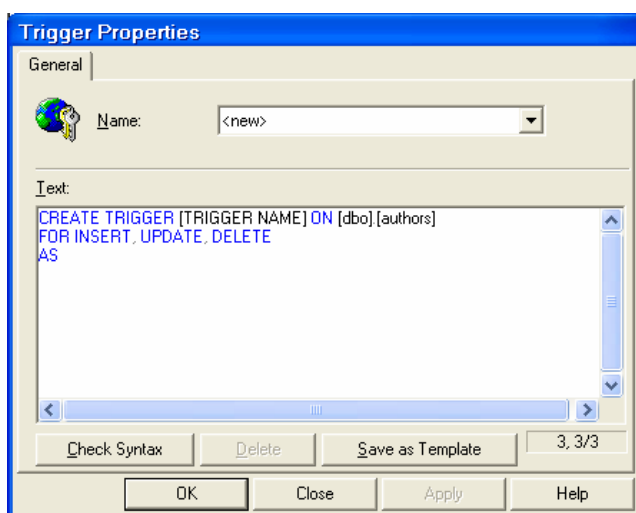
T o trigger c th c hi n thông công c và câu l nh:

T o trigger b ng công c .

- Ch n b ng d li u ho c khung nhìn.
- Nh n nút ph i chu t.
- Ch n All tasks -> Manage Triggers...



- So n k ch b n t o trigger.



(Cú pháp c th h n b n xem trong ph n tiếp theo)

T o trigger b ng câu l nh.

S d ng l nh Create Trigger, cú pháp chung nh sau:

```

CREATE TRIGGER trigger_name
ON { table | view }
[ WITH ENCRYPTION ]
{
  { { FOR | AFTER | INSTEAD OF } { [ INSERT ] [, ] [ UPDATE ] }
    [ WITH APPEND ]
    [ NOT FOR REPLICATION ]
  AS
  [ { IF UPDATE ( column )
    [ { AND | OR } UPDATE ( column ) ]
    [ ...n ]
  | IF ( COLUMNS_UPDATED ( ) { bitwise_operator } updated_bitmask )
    { comparison_operator } column_bitmask [ ...n ]
  } ]
  sql_statement [ ...n ]
}
}

```

Các tham số cần biết:

- + *trigger_name*: Tên trigger.
- + *table/view*: Tên bảng hoặc khung nhìn.
- + For/After/Instead Of: Loại trigger.
- + { [DELETE] [,] [INSERT] [,] [UPDATE] }: Sự kiện khi trigger được kích hoạt.
- + *sql_statement* [...*n*]: Khối lệnh các câu lệnh xử lý của trigger.

Các câu lệnh sau không thể thực thi trong khối lệnh các câu lệnh xử lý của trigger:

ALTER DATABASE	CREATE DATABASE	DISK INIT
DISK RESIZE	DROP DATABASE	LOAD DATABASE
LOAD LOG	RECONFIGURE	RESTORE DATABASE
RESTORE LOG		

có thể bạn ta xét một số ví dụ sau:

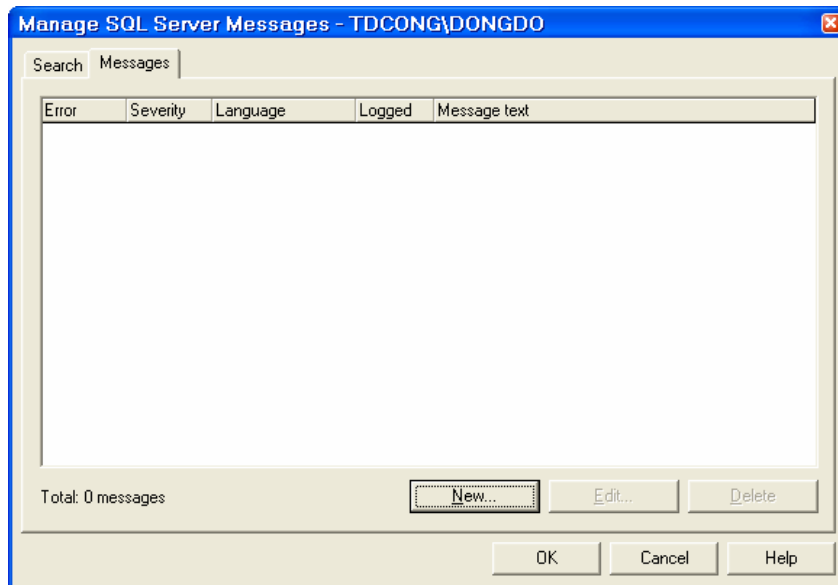
Ví dụ tạo trigger thông báo.

```
CREATE TRIGGER reminder
ON titles
FOR INSERT, UPDATE
AS RAISERROR (50001, 16, 10)
GO
```

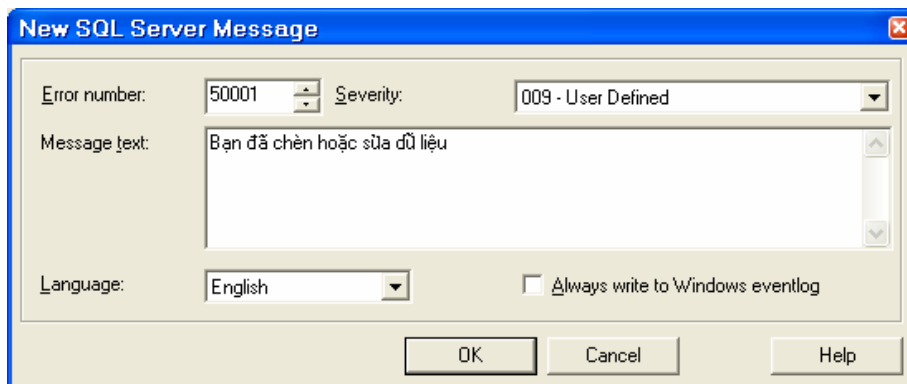
Ví dụ trên tạo trigger thông báo cho các client khi thực hiện thêm hoặc sửa dữ liệu trên bảng Titles, mã thông báo là 50001, là mã thông báo do người dùng định nghĩa.

Để tạo thông báo bạn thao tác như sau:

- Vào menu Tools -> Manage SQL Server Messages...



- Chọn bảng Messages -> New...



- Nhập mã, số định nghĩa, kiểu thông báo (Severity), mã thông báo sử dụng trong các ứng dụng hoặc câu lệnh yêu cầu.

Ví dụ tạo trigger thông báo gửi Email khi có thay đổi.

```
CREATE TRIGGER reminder
ON titles
FOR INSERT, UPDATE, DELETE
AS
    EXEC master..xp_sendmail 'MaryM',
        'Don''t forget to print a report for the
        distributors.'
GO
```

Ví dụ tạo trigger kiểm soát kho giá trị giữa 2 bảng.

Ví dụ sau sẽ tạo trigger thể hiện kiểm soát phạm vi mức lương của nhân viên và chèn vào có thu được giá trị nằm trong bảng mức lương hay không.

```
CREATE TRIGGER employee_insupd
ON employee
FOR INSERT, UPDATE
AS
DECLARE @min_lvl tinyint,
        @max_lvl tinyint,
        @emp_lvl tinyint,
        @job_id smallint
SELECT @min_lvl = min_lvl,
        @max_lvl = max_lvl,
        @emp_lvl = i.job_lvl,
        @job_id = i.job_id
FROM employee e INNER JOIN inserted i ON e.emp_id =
i.emp_id
    JOIN jobs j ON j.job_id = i.job_id
IF (@job_id = 1) and (@emp_lvl <> 10)
BEGIN
    RAISERROR ('Job id 1 expects the default level of
10.', 16, 1)
    ROLLBACK TRANSACTION
END
ELSE
IF NOT (@emp_lvl BETWEEN @min_lvl AND @max_lvl)
```

```
BEGIN
    RAISERROR ('The level for job_id:%d should be
between %d and %d.',
    16, 1, @job_id, @min_lvl, @max_lvl)
    ROLLBACK TRANSACTION
END
```

S A, XÓA TRIGGER.

S d ng công c .

- Ch n trigger trong m c Manage Triggers...
- Th c hi n s a n i dung ho c xóa.

S a, xóa theo câu l nh.

- S d ng l nh Alter trigger s a.
- S d ng l nh Drop Trigger xóa.

(B n có th tìm hi u chi ti t h n trong Book Online)

XU T – NH P D LI U

Trong chương này bạn sẽ tìm hiểu kỹ thuật trao đổi dữ liệu với các môi trường ngoài Server của SQL, có thể với một CSDL khác, một Server SQL khác, một hệ thống CSDL khác hoặc nối ghép tập tin CSDL.

SERVER LIÊN KẾT – LINKED SERVER.

Trong phần này các hệ thống CSDL liên kết khác (Access, Oracle), hệ thống cung cấp công cụ liên kết với hệ thống CSDL khác. Khi liên kết đã thực hiện thì lập, với quy định của user liên kết bạn có thể thực hiện khai thác dữ liệu liên kết trên SQL Server.

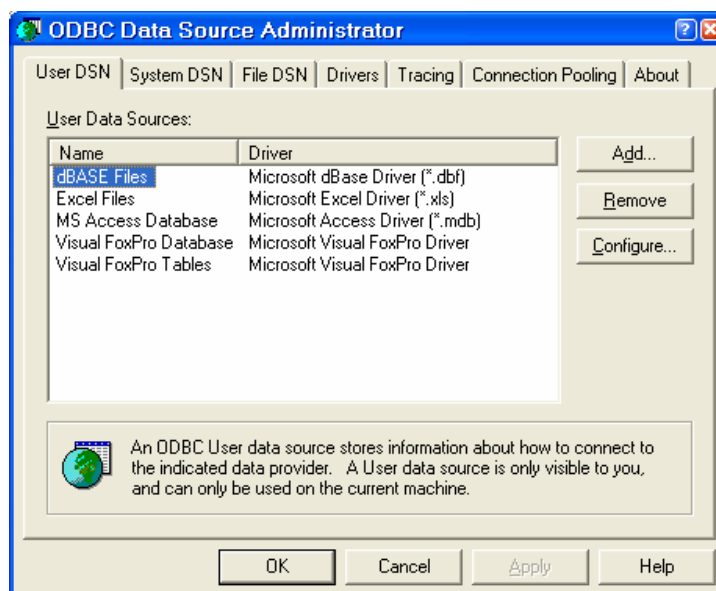
TỔNG QUAN ODBC.

ODBC viết tắt của Open DataBase Connectivity, là công cụ của Windows cung cấp với mục đích làm môi trường trao đổi dữ liệu giữa các CSDL, giữa nhiều hệ thống CSDL với nhau.

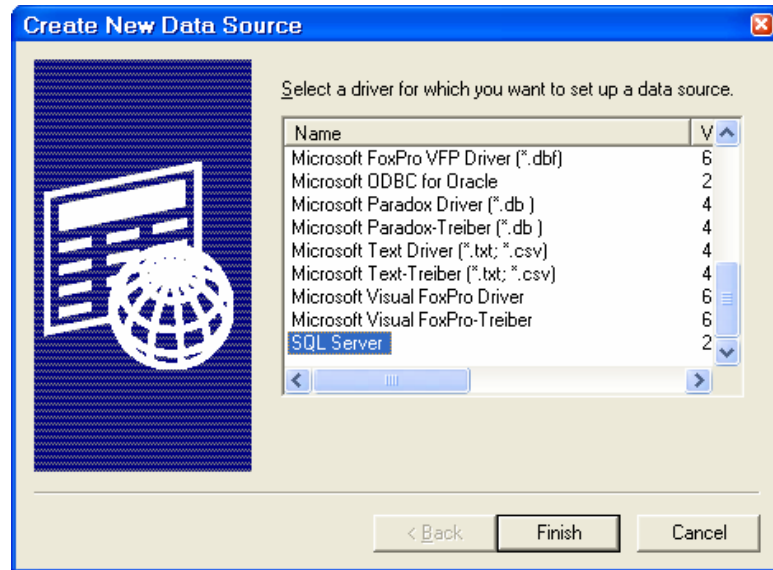
ODBC tổng hợp chung nhiều kết nối CSDL và quản lý. Khi thực hiện khai thác dữ liệu thông qua ODBC, quản lý liên kết theo tên ODBC, quy định khai thác thực hiện khi kết nối ODBC đến CSDL.

Cách tạo ODBC:

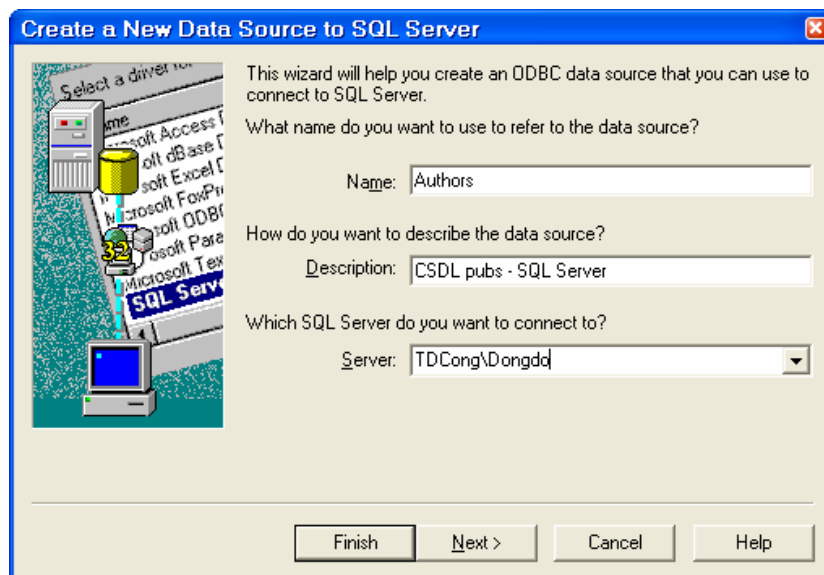
- Chọn ODBC trong Control panel.



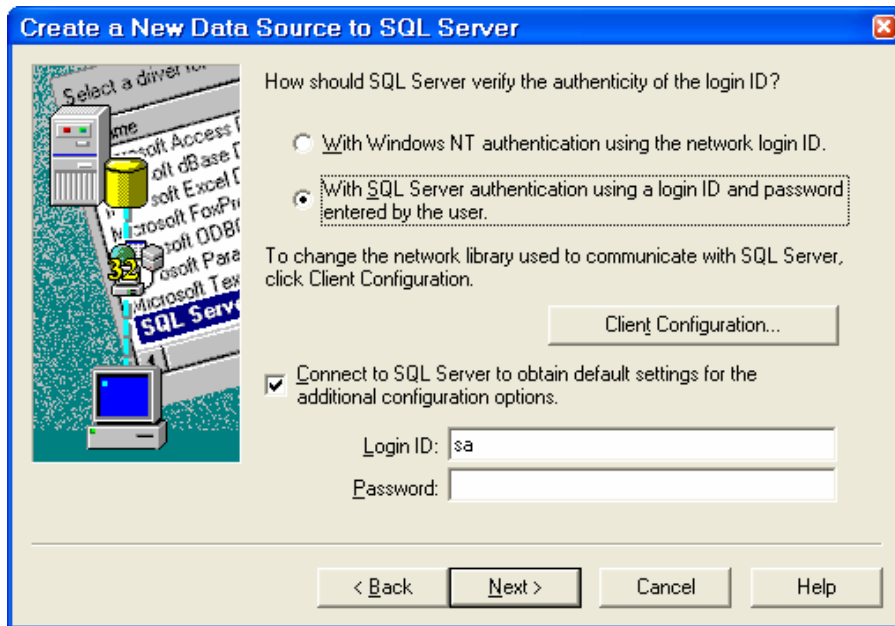
- Chọn bảng User DSN -> Add...



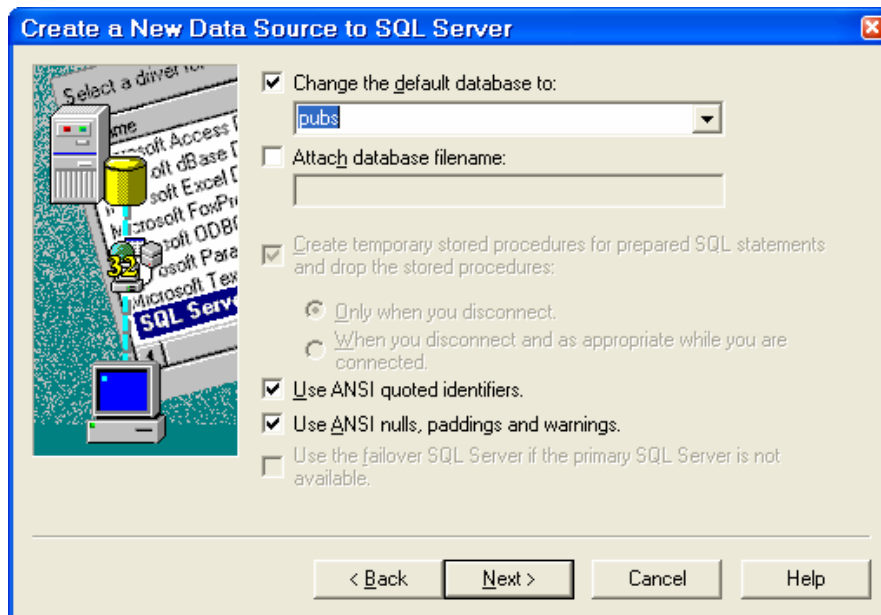
- Chọn Driver có ảnh quản trị CSDL của CSDL cần liên kết ODBC.
- Chọn Finish.



- Nhập tên ODBC (tên này sẽ sử dụng cho ứng dụng khác, nên nhập theo chuẩn chung để sử dụng), các tham số khác (đây trên hình sử dụng Driver của SQL Server nên bổn phải chỉ Server).
- Next.

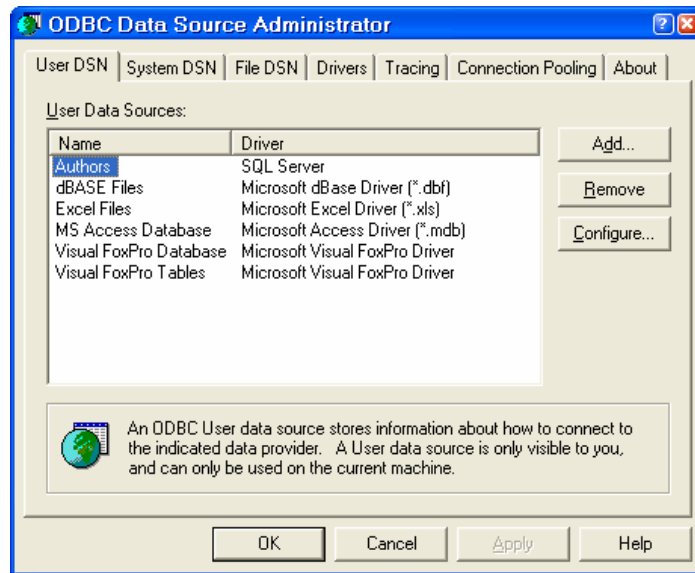


- Nhập Login ID, mật khẩu -> Next.



- Chọn 'Change the default database to' -> Chọn 'pubs' -> Next -> Finish

Sau khi tạo xong trong danh sách xuất hiện ODBC bạn vào, tạo danh sách bạn có thể xóa ODBC khi cần thiết.

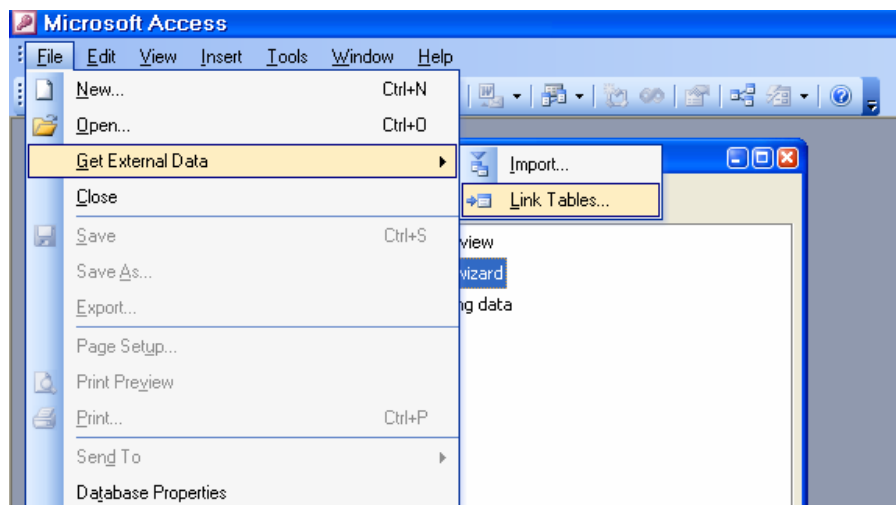


T o liên k t t Access.

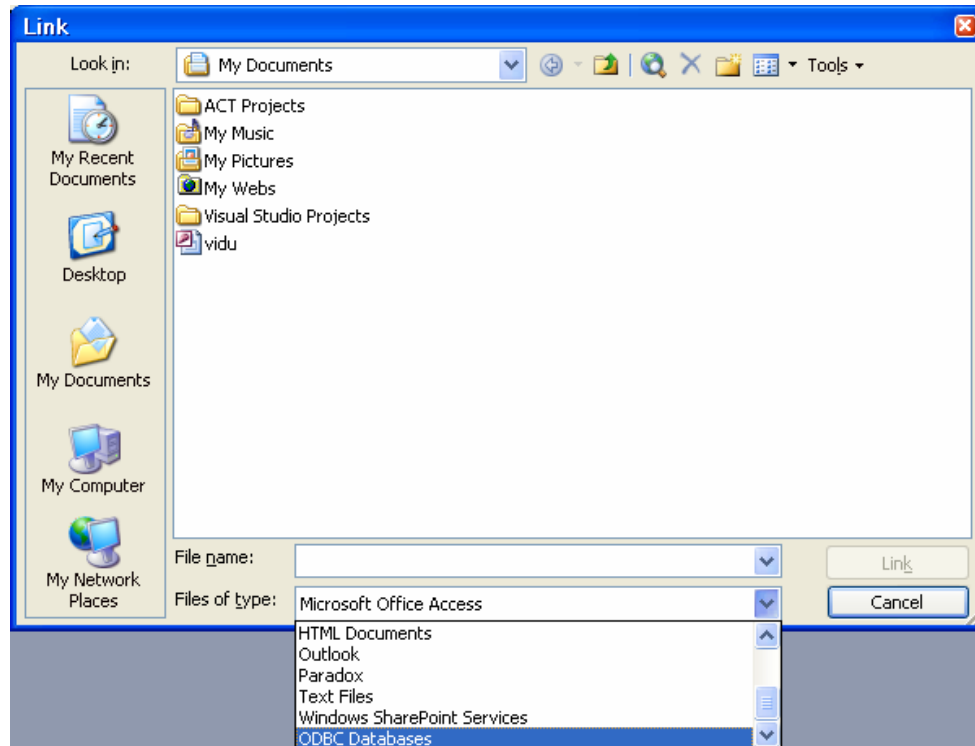
T h qu n tr CSDL Access b n có th t o liên k t n các h qu n tr CSDL khác (Access, Dbase,...), ho c thông qua ODBC. Trong ví d minh h a s d ng liên k t t Access v i ODBC (i v i SQL Server ho c Oracle, My SQL thì Access ph i liên k t thông qua ODBC vì các h qu n tr CSDL này không th c hi n khai thác d li u qua t p tin ch khai thác thông qua tên CSDL, mà Access ch th c hi n theo ph ng th c m t p tin).

Các b c th c hi n nh sau:

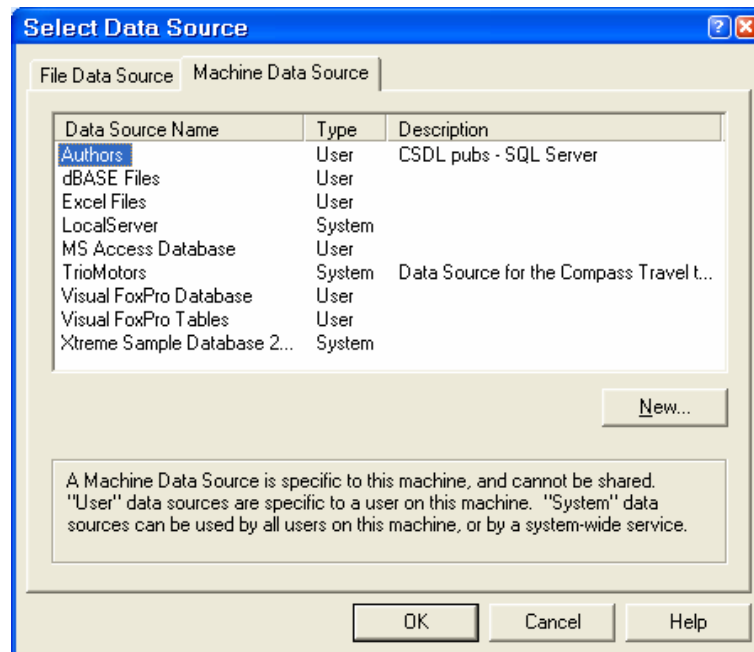
- M h qu n tr CSDL Access.
- M ho c t o CSDL m i t Access
- Ch n File -> Get External Data -> Link Tables.



- Chọn ODBC Databases.



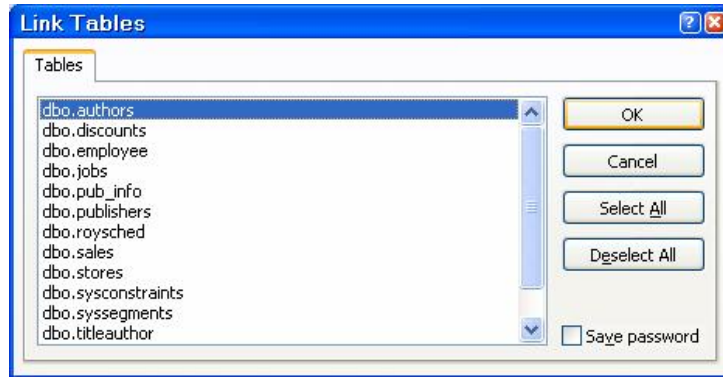
- Chọn ODBC cần liên kết (Authors).



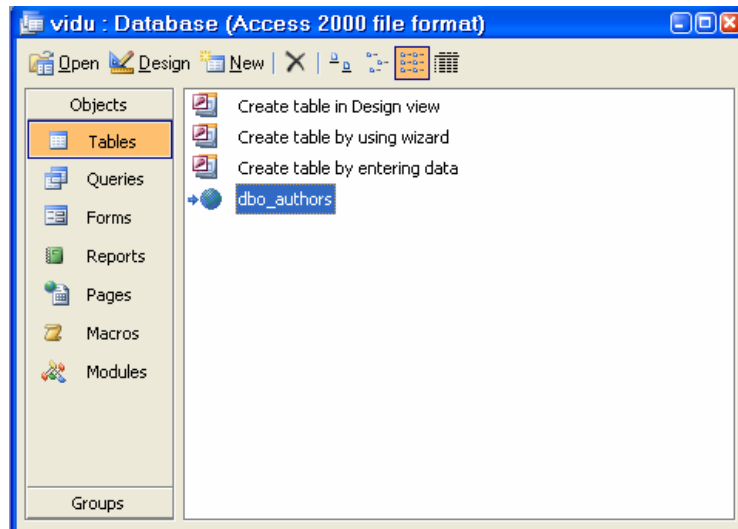
- Ok.

- Nhập Login ID và mật khẩu.

- Chọn bảng hoặc khung nhìn cần liên kết trong danh sách.



- Nhấn Ok, danh sách các bảng trong Access sẽ khai thác từng một bảng khác.



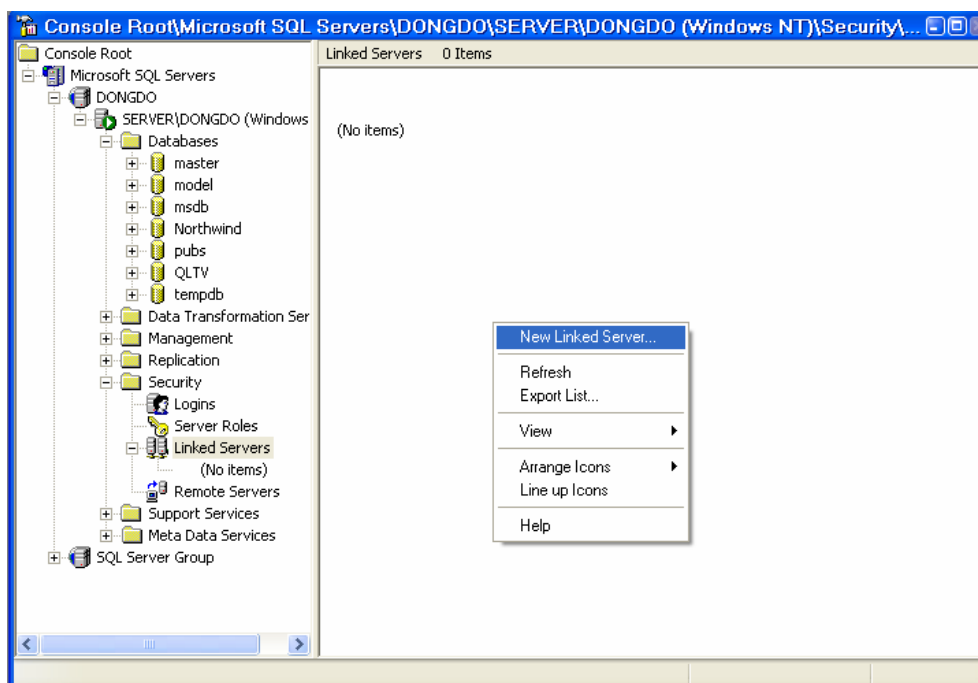
au_id	au_lname	au_fname	phone	address	city	state
409-56-7008	Bennet	Abraham	415 658-9932	6223 Bateman St	Berkeley	CA
648-92-1872	Blotchet-Halls	Reginald	503 745-6402	55 Hillsdale Bl.	Corvallis	OR
238-95-7766	Carson	Cheryl	415 548-7723	589 Darwin Ln.	Berkeley	CA
722-51-5454	DeFrance	Michel	219 547-9982	3 Balding Pl.	Gary	IN
712-45-1867	del Castillo	Innes	615 996-8275	2286 Cram Pl. #	Ann Arbor	MI
427-17-2319	Dull	Ann	415 836-7128	3410 Blonde St.	Palo Alto	CA
213-46-8915	Green	Marjorie	415 986-7020	309 63rd St. #4	Oakland	CA
527-72-3246	Greene	Morningstar	615 297-2723	22 Graybar Hou	Nashville	TN
472-27-2349	Gringlesby	Burt	707 938-6445	PO Box 792	Covelo	CA
846-92-7186	Hunter	Sheryl	415 836-7128	3410 Blonde St.	Palo Alto	CA
756-30-7391	Karsen	Livia	415 534-9219	5720 McAuley S	Oakland	CA
486-29-1786	Locksley	Charlene	415 585-4620	18 Broadway Av	San Francisco	CA
724-80-9391	MacFeather	Stearns	415 354-7128	44 Upland Hts.	Oakland	CA
893-72-1158	McBadden	Heather	707 448-4982	301 Putnam	Vacaville	CA
267-41-2394	O'Leary	Michael	408 286-2428	22 Cleveland Av	San Jose	CA
807-91-6654	Panteley	Sylvia	301 946-8853	1956 Arlington F	Rockville	MD
998-72-3567	Ringer	Albert	801 826-0752	67 Seventh Av.	Salt Lake City	UT
899-46-2035	Ringer	Anne	801 826-0752	67 Seventh Av.	Salt Lake City	UT

T o Server liên k t – Linked Server.

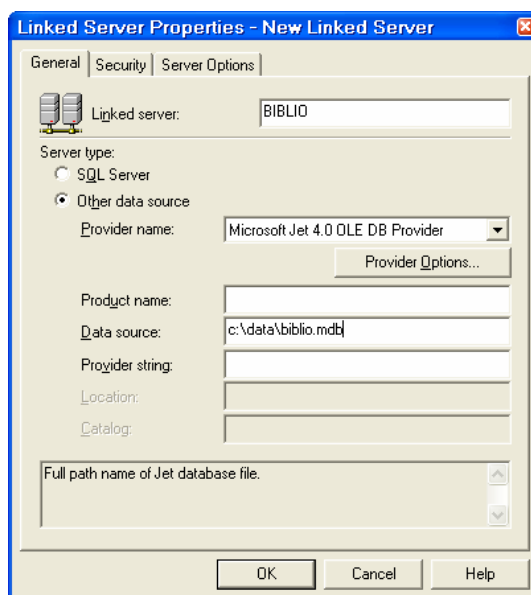
T SQL Server có th t o liên k t tr c ti p n các h qu n tr CSDL khác (Access, SQL Server, Oracle, My SQL,...) mà không c n thi t ph i thông qua ODBC nh Access ã xét tr c.

T o b ng công c .

- Vào m c Security -> Linked Server.



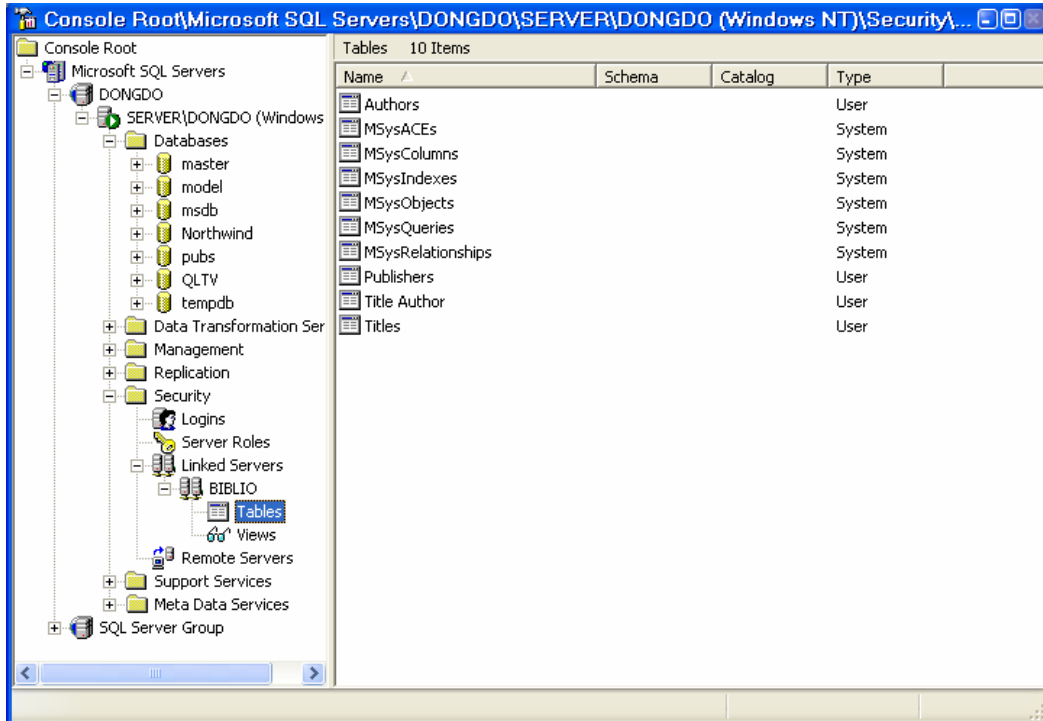
- Nh n nút ph i chu t -> New Linked Server.



- Nh p các tham s :

+ Tên Server.

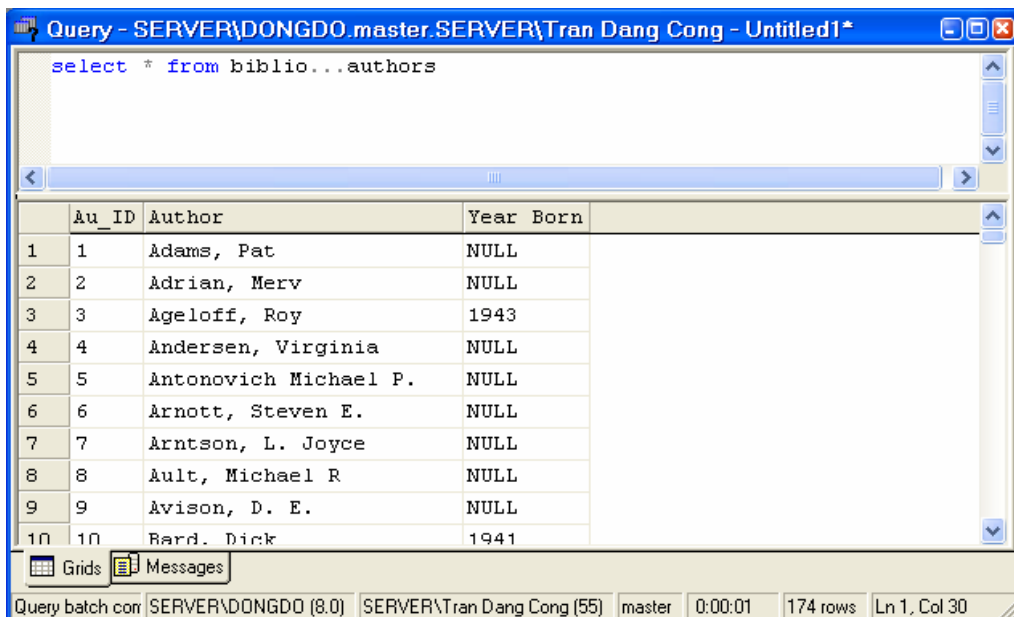
+ Provider (Driver của hệ quản trị CSDL cần thiết để liên kết, trong ví dụ minh họa thể hiện với Access).



- Thể hiện khai thác thông qua câu lệnh, trong câu ISQL phải chỉ định, ví dụ

```
Select * from Biblio...Authors
```

để thể hiện liệt kê toàn bộ danh sách các bản ghi của bảng authors.



T o b ng câu l nh.

S d ng l nh `sp_addlinkedserver` t o server liên k t.

Ví d t o Linked Server Biblio:

```
sp_addlinkedserver 'Biblio',  
                  'Access 97', 'Microsoft.Jet.OLEDB.4.0',  
                  'c:\data\biblio.mdb'
```

(ng d n ph i phù h p v i Server)

Xóa Linked Server.

- S d ng công c : Ch n Linked Server c n xóa -> th c hi n xóa.

- S d ng l nh :

```
sp_dropserver [ @server = ] 'server'  
                [ , [ @droplogins = ] { 'droplogins' | NULL} ]
```

S D NG BCP VÀ BULK INSERT NH P D LI U.

Bcp là câu l nh d ng command prompt, dùng xu t (export) và nhâp (import) d li u gi a SQL Server và t p tin (d ng text ho c excel). Các t p tin tham gia xu t nh p ph i có c u trúc d li u ki u b ng (hàng, c t), b ng d li u c a SQL Server khi th c hi n nh p d li u ph i có c u trúc t ng ng có s n.

Bulk insert là câu l nh t ng t bcp nh ng ch th c hi n import d li u mà không export.

Cú pháp l nh bcp.

L nh bcp c th c hi n t i c a s l nh (command prompt).

```
bcp {[[database_name.][owner.]{table_name | view_name} | "query"}  
      {in | out | queryout | format} data_file  
      [-m max_errors] [-f format_file] [-e err_file]  
      [-F first_row] [-L last_row] [-b batch_size]  
      [-n] [-c] [-w] [-N] [-V (60 | 65 | 70)] [-6]  
      [-q] [-C code_page] [-t field_term] [-r row_term]
```

[-i input_file] [-o output_file] [-a packet_size]
[-S server_name[instance_name]] [-U login_id] [-P password]
[-T] [-v] [-R] [-k] [-E] [-h "hint [,...n]"]

(b n tìm hi u thêm trong book online)

Ví d s d ng l nh bcp.

+ S d ng l nh có t khóa out copy toàn b d li u t m t b ng ho c khung nhìn ra t p tin.

```
bcp pubs..titleview out titleview.txt -c -Sservername -  
Username -Ppassword
```

+ S d ng l nh Select copy m t t p ra t p tin, có t khóa queryout.

```
bcp "SELECT au_fname, au_lname FROM pubs..authors ORDER  
BY au_lname" queryout c:\Authors.txt -c -Sservername -  
Username -Ppassword
```

K t qu th c hi n: N i dung t p tin Authors.txt

Abraham	Bennet
Reginald	Blotchet-Halls
Cheryl	Carson
Michel	DeFrance
Innes	del Castillo
Ann	Dull
Marjorie	Green
Morningstar	Greene
Burt	Gringlesby
Sheryl	Hunter
Livia	Karsen

Charlene Locksley
 Stearns MacFeather
 Heather McBadden
 Michael O'Leary
 Sylvia Panteley
 Albert Ringer
 Anne Ringer
 Meander Smith
 Dean Straight
 Dirk Stringer
 Johnson White
 Akiko Yokomoto

M t s tham s c b n:

- Out: Copy toàn b m t Table ho c view ra t p tin.
- Queryout: Copy t p d li u c truy v n theo câu l nh.
- c: Ch ra r ng câu l nh dùng ki u ký t phân nh các c t, n u không ch thì câu l nh t nh n tab (\t) phân nh và dùng new line xu ng dòng m i.

Cú pháp l nh Bulk Insert.

L nh Bulk Insert g n gi ng l nh bcp nh ng Bulk Insert ch s d ng nh p d li u vào SQL Server (Insert), l nh này c th c hi n b ng SQL Query Analyzer.

Cú pháp chung:

```
BULK INSERT [ [ 'database_name'. ] [ 'owner' ]. ] { 'table_name' FROM 'data_file' }
  [ WITH
  (
    [ BATCHSIZE [ = batch_size ] ]
    [ [ , ] CHECK_CONSTRAINTS ]
    [ [ , ] CODEPAGE [ = 'ACP' | 'OEM' | 'RAW' | 'code_page' ] ]
    [ [ , ] DATAFILETYPE [ =
```

```

    { 'char' | 'native' | 'widechar' | 'widenative' } ] ]
  [ [ , ] FIELDTERMINATOR [ = 'field_terminator' ] ]
  [ [ , ] FIRSTROW [ = first_row ] ]
  [ [ , ] FIRE_TRIGGERS ]
  [ [ , ] FORMATFILE = 'format_file_path' ]
  [ [ , ] KEEPIDENTITY ]
  [ [ , ] KEEPNULLS ]
  [ [ , ] KILOBYTES_PER_BATCH [ = kilobytes_per_batch ] ]
  [ [ , ] LASTROW [ = last_row ] ]
  [ [ , ] MAXERRORS [ = max_errors ] ]
  [ [ , ] ORDER ( { column [ ASC | DESC ] } [ ,...n ] ) ]
  [ [ , ] ROWS_PER_BATCH [ = rows_per_batch ] ]
  [ [ , ] ROWTERMINATOR [ = 'row_terminator' ] ]
  [ [ , ] TABLOCK ]
)
]

```

Ví dụ thể hiện copy toàn bộ dữ liệu tệp tin newpubs.dat vào bảng publishers2 (bảng này đã có cấu trúc sẵn), tệp tin kết thúc bằng ký tự xuống dòng '\n' (xuống dòng để đổi dòng và chuyển xuống dòng).

```

BULK INSERT pubs..publishers2 FROM 'c:\newpubs.dat'
WITH (
    DATAFILETYPE = 'char',
    FIELDTERMINATOR = ',',
    ROWTERMINATOR = '\n'
)

```

DETACH VÀ ATTACH C S D L I U.

Mục này sẽ nói về kỹ thuật và cách ghép tệp tin CSDL về Server. Giả sử bạn đã có các tệp tin của CSDL (gồm tệp tin dữ liệu và nhật ký có thể copy về vị trí khác).

Copy tệp tin của CSDL.

Trước tiên ta xem kỹ thuật copy các tệp tin CSDL sang một vị trí khác (mà vẫn giữ vị trí), sau khi copy sang vị trí khác bạn có thể sao chép sang Instance mới. Các bước thể hiện như sau:

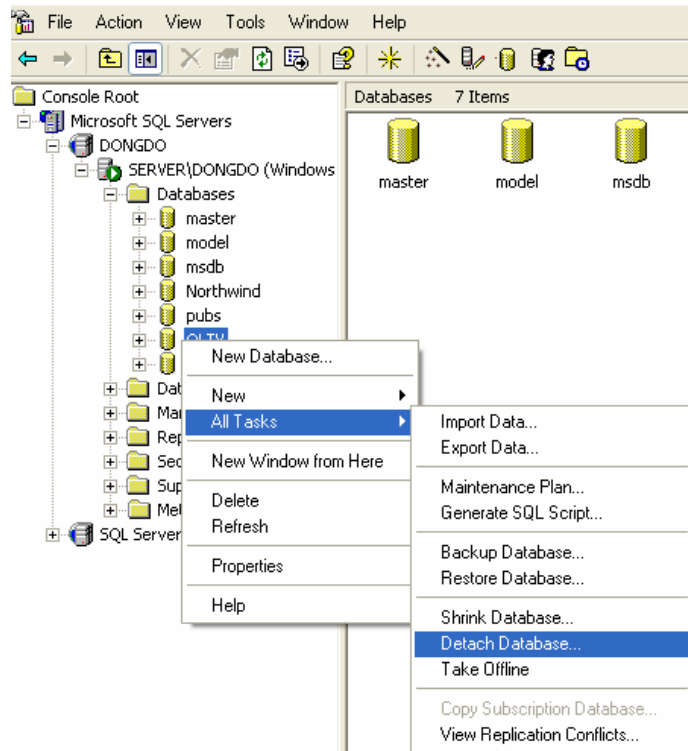
- Stop dịch vụ SQL của Instance có CSDL.

- Copy các tệp tin của CSDL sang vị trí cần thiết.
- Start dịch vụ SQL của Instance thì phần này mới làm việc.

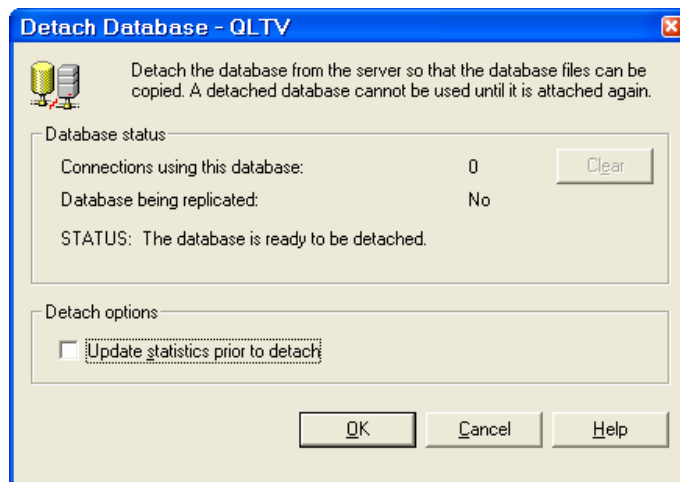
Detach cơ sở dữ liệu.

Là bước cuối cùng để tách CSDL khỏi Instance, Instance không quản lý CSDL nữa. Việc khác với xóa CSDL là các tệp tin của CSDL vẫn còn.

- Chọn CSDL cần detach. -> All tasks -> Detach Database



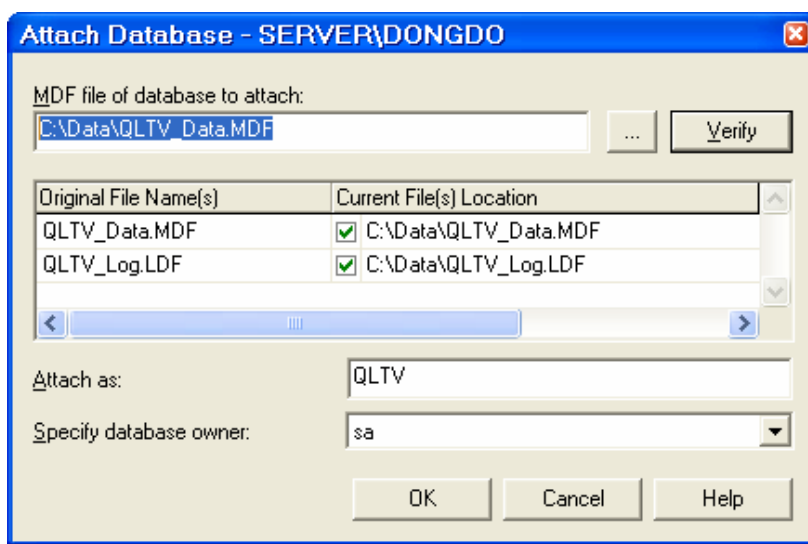
- Nhấn Ok.



Attach tệp tin CSDL vào Instance.

Mục này giới thiệu cách thu thập tệp tin CSDL vào Instance, là bước tiếp theo của các bước Copy và Dettach. Các bước thực hiện như sau:

- Chọn Instance cần Attach CSDL -> Databases -> all tasks -> attach database...
- Chọn nút browse (...)



- Chọn tệp tin mdf của CSDL cần attach.
- Đặt tên CSDL.
- Xác định User owner.
- Ok.

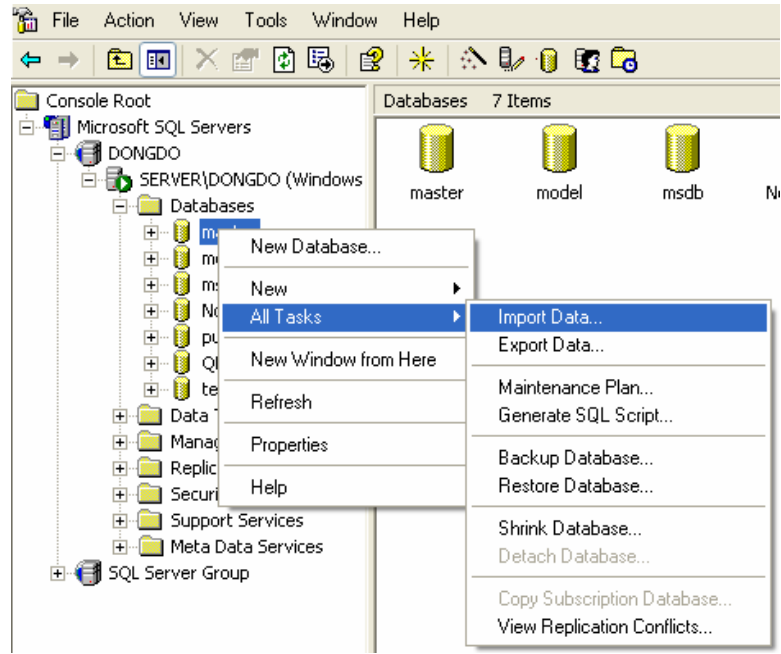
IMPORT VÀ EXPORT C S D L I U.

Phần này sẽ trình bày cách thu thập và xuất dữ liệu CSDL với các hệ quản trị CSDL khác hoặc Instance, CSDL khác của SQL Server.

Import – Nhập dữ liệu.

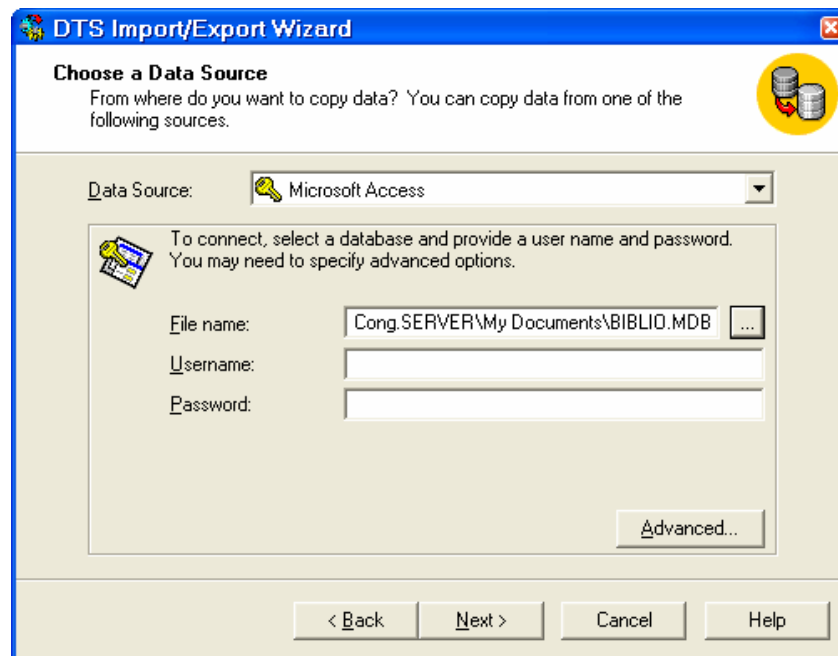
Dùng nhập dữ liệu ngoài vào CSDL hệ quản trị CSDL khác hoặc CSDL khác của SQL Server.

- Chọn Databases -> All tasks -> Import Data...

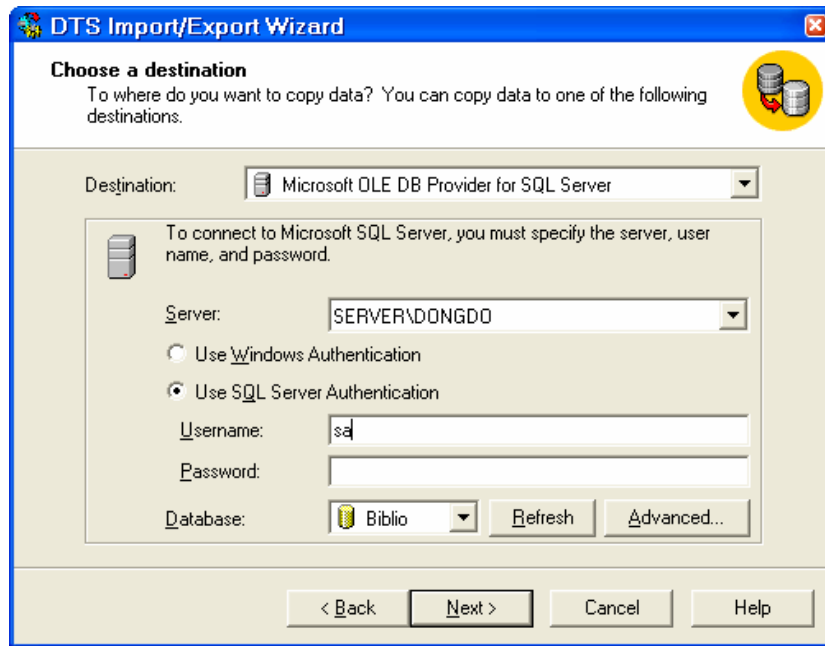


- Next -> Chọn Data Source (Có thể là SQL Server, Oracle, Access,...), trong ví dụ minh họa chọn Access.

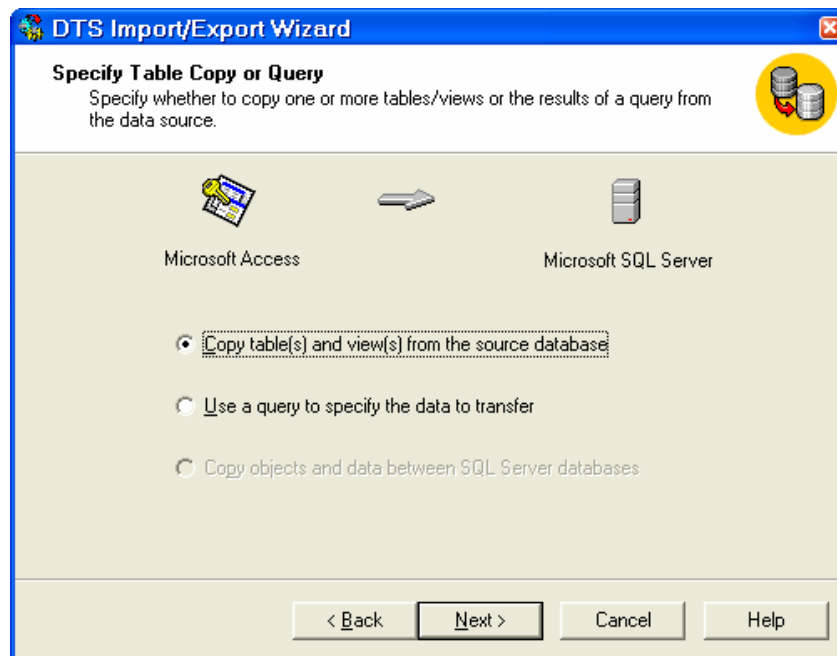
- Chọn tệp tin (file name) -> Next



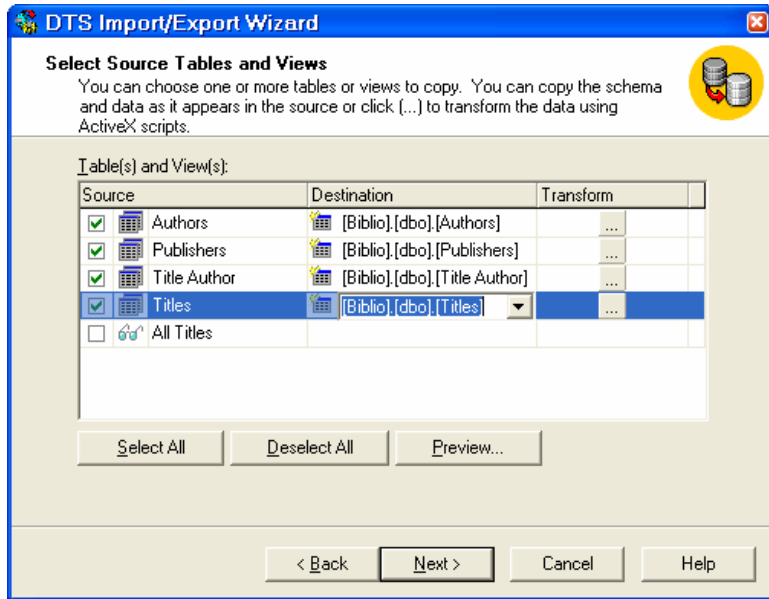
- Chọn Instance cần chuyển dữ liệu vào, user name., tên CSDL (có thể có thể tạo mới bằng cách chọn New) -> Next



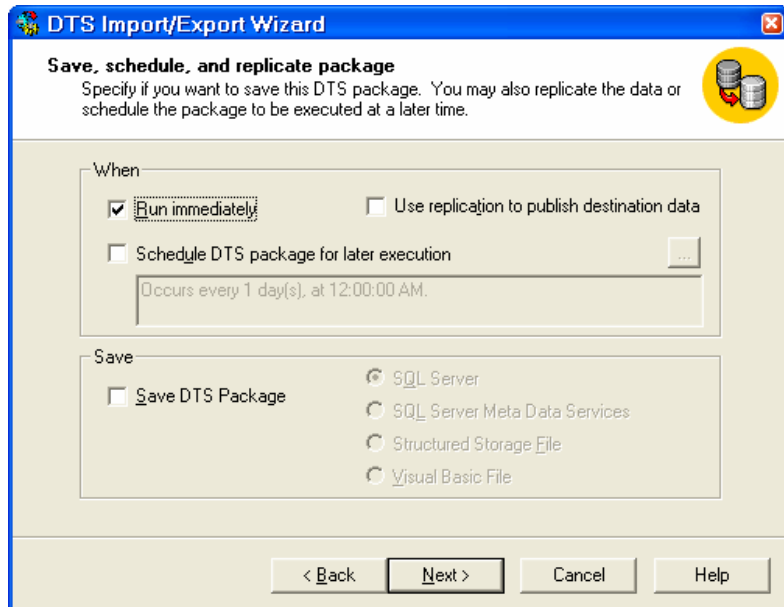
- Chọn cách chuyển toàn bộ bảng dữ liệu hay thông qua câu lệnh truyền v n (trong ví dụ minh họa chọn bảng dữ liệu) -> Next



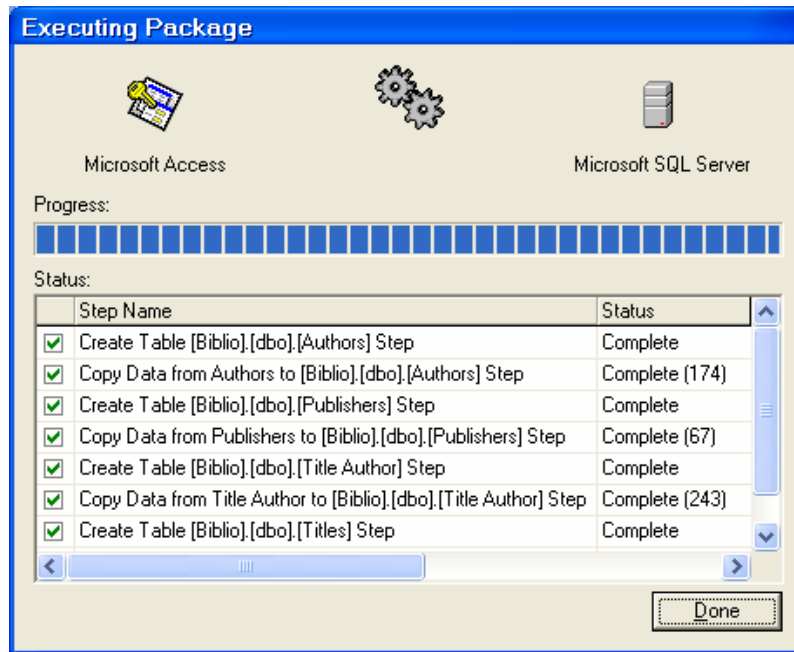
- Chọn các bảng, khung nhìn cần Import (có thể là chọn một số chức năng khác có thể hiện, bên cạnh tìm hiểu), tên các bảng, khung nhìn của SQL Server nhận dữ liệu -> Next.



- Chọn tiếp chọn nguồn thì click ngày hay theo lịch → Next → Finish

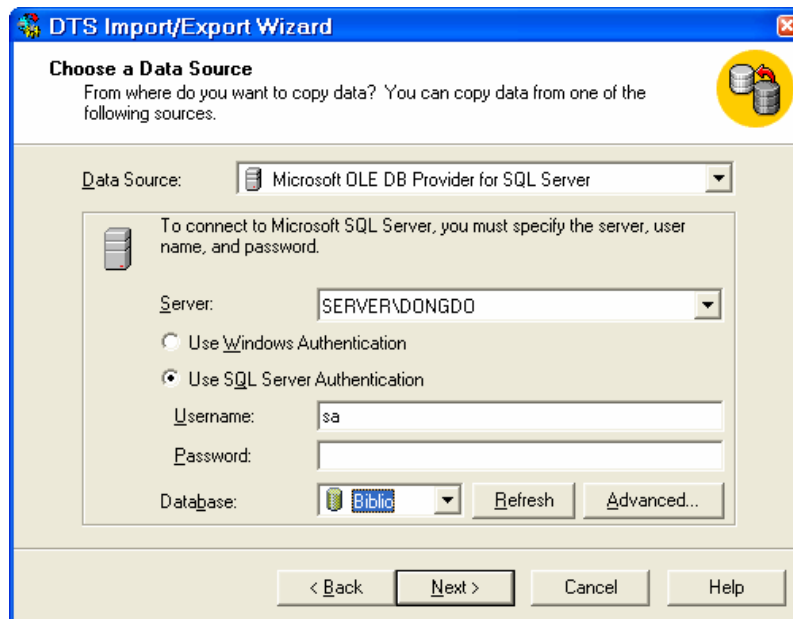


- Xem thông báo sau khi chuyển → Done



EXPORT – XU T D LI U.

Phần này giới thiệu kỹ thuật xuất dữ liệu từ một CSDL của SQL Server ra một hệ quản trị CSDL khác hoặc một CSDL khác của SQL Server. Trong trình Import cũng như Export thì các hiển Data Source là SQL Server, còn Destination là hệ quản trị CSDL khác hoặc CSDL khác của SQL Server (phần này bạn có thể xem xét).



SAO L U, KHÔI PH C D LI U

Chương này sẽ giới thiệu kỹ thuật sao l u (backup) và khôi ph c (restore) d li u, là kỹ thuật th ng c s d ng b o m an toàn d li u phòng tr ng h p CSDL b h ng, nh t ký d li u. Ch c n ng này c th c hi n b ng 2 ph ng pháp: B ng công c và câu l nh T-SQL.

NH NG LÝ DO PH I SAO L U VÀ KHÔI PH C D LI U.

Trong quá trình th c hi n qu n tr CSDL SQL Server thì m t s nguyên nhân sau ây b t bu c b n ph i xem xét n k thu t sao l u và khôi ph c d li u:

- + a b h ng (ch a các t p tin CSDL).
- + Server b h ng.
- + Nguyên nhân bên ngoài (thiên nhiên, h a ho n, m t c p,...)
- + User vô tình xóa d li u.
- + B vô tình hay c ý làm thông tin sai l ch.
- + B hack.

CÁC LO I BACKUP.

Backup d li u trong SQL Server g m các lo i sau:

- + Full Database Backups: Copy toàn b CSDL (các t p tin bao g m các b ng, khung nhìn, các i t ng khác).
- + Differential Database Backups: Copy nh ng d li u thay i trong Data file k t l n full backup g n nh t.
- + File or file group backups: Copy m t file n hay file group.
- + Differential File or File Group Backups: Th c hi n nh Differential Database nh ng copy ph n d li u thay i c a file n ho c file group.
- + Transaction log backups: Ghi nh n t t c các transaction ch a trong transaction log file k t l n transaction log backup g n nh t. V i lo i sao l u này ta có th khôi ph c d li u t i m t th i i m.

CÁC MÔ HÌNH PH C H I D L I U.

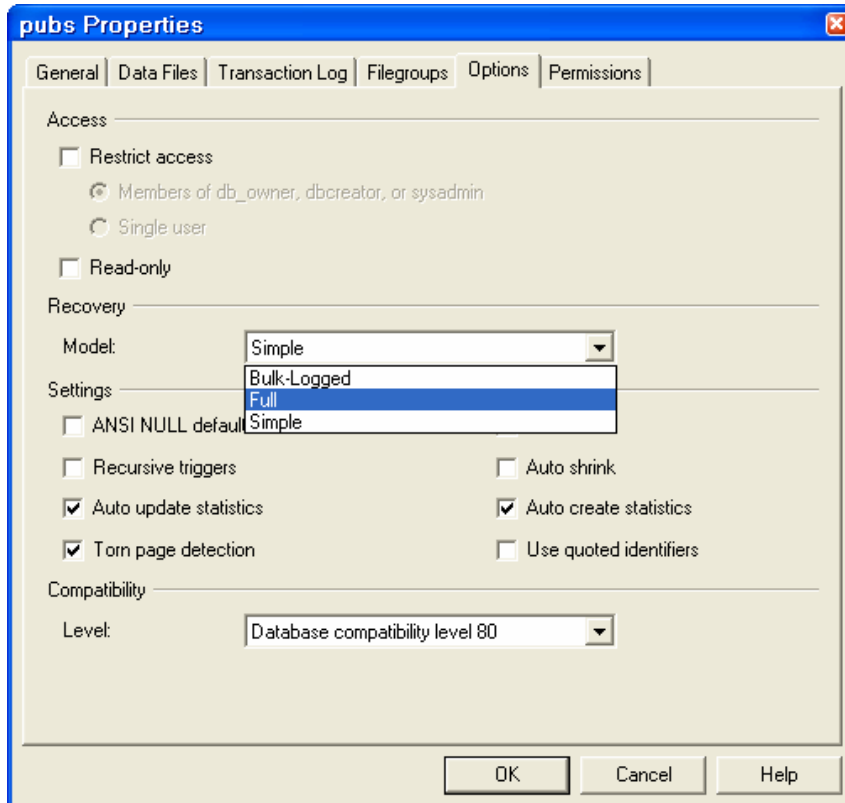
+ Full Recovery model: Là mô hình ph c h i toàn b h o t ng giao d ch c a d li u (Insert, Update, Delete, h o t ng b i l nh bcp, bulk insert). V i mô hình này ta có th ph c h i d li u t i m t th i i m trong quá kh ã c l u trong transaction log file.

+ Bulk-Logged Recovery Model: Mô hình này c th c thi cho các thao tác bcp, bulk insert, create index, writetext, updatetext, các h o t ng này ch nh t ký s k i n vào log b i t mà không sao l u toàn b d li u, chi t i t nh trong full recover. Các s k i n Insert, Update, Delete v n c nh t ký và khôi ph c bình th ng.

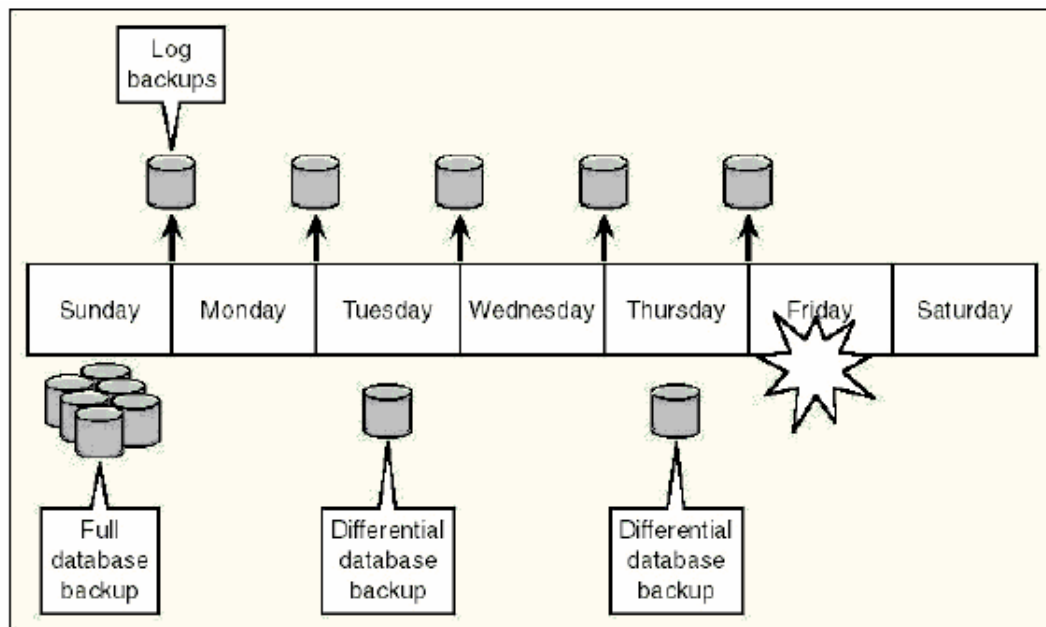
+ Simple Recovery Model: V i mô hình này b n ch ph c h i l i th i i m backup g n nh t mà không theo th i i m khác trong quá kh .

Cách t mô hình khôi ph c:

- Ch n CSDL.
- Nh n nút ph i chu t -> Properties -> Options -> Recovery



Xét ví dụ sau: Giả sử ta có một CSDL cần backup theo chỉ định như hình vẽ:



Nhìn hình trên ta thấy CSDL cần lập lịch Full Database Backup vào ngày chủ nhật, Differential Database Backup vào ngày thứ ba và thứ năm, còn Log Database Backup vào 5 ngày trong tuần, ngày thứ sáu có sự cố vì CSDL data file bị hỏng, vì vậy phát ra là phải phải chi dữ liệu và CSDL hoạt động bình thường. Ta phải làm các bước sau:

- + Thực hiện Backup log file (giả sử log file không bị hỏng).
- + khôi phục Full Database của ngày chủ nhật.
- + Phải chi Differential Database của ngày thứ năm.
- + khôi phục Transaction log backup ngày thứ năm.

SAO L U C S D L I U - BACKUP DATABASE.

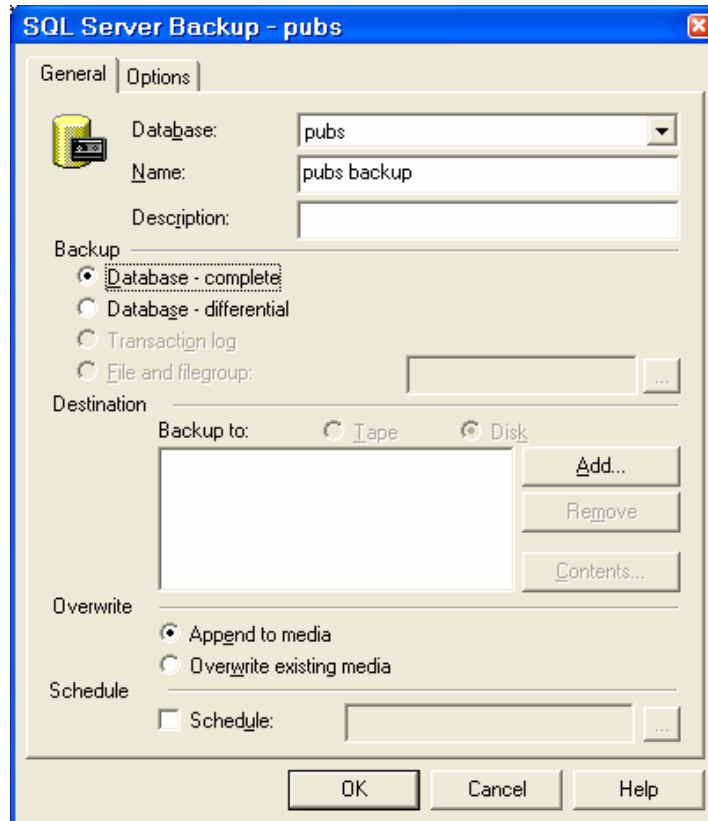
Trước khi xem xét kỹ thuật sao lưu CSDL, ta thấy như một số thuật ngữ thông dụng như sau:

- + Backup: Là quá trình copy toàn bộ hoặc một phần database, transaction log, file, file group thành một tập backup set được chứa trong backup media (disk hoặc tape) bằng cách sử dụng một backup device (tape drive name hoặc physical filename).
- + Backup Device: Một file vật lý hoặc một drive tape.

- + Backup file: Một file chứa Backup set.
- + Backup media: LÀ Disk hoặc tape.
- + Backup set: Một bộ backup một lần backup trên backup media.

Các bước thực hiện backup như sau:

- Chọn CSDL cần backup.
- Nhấn phải chuột -> All Tasks -> Backup Database...



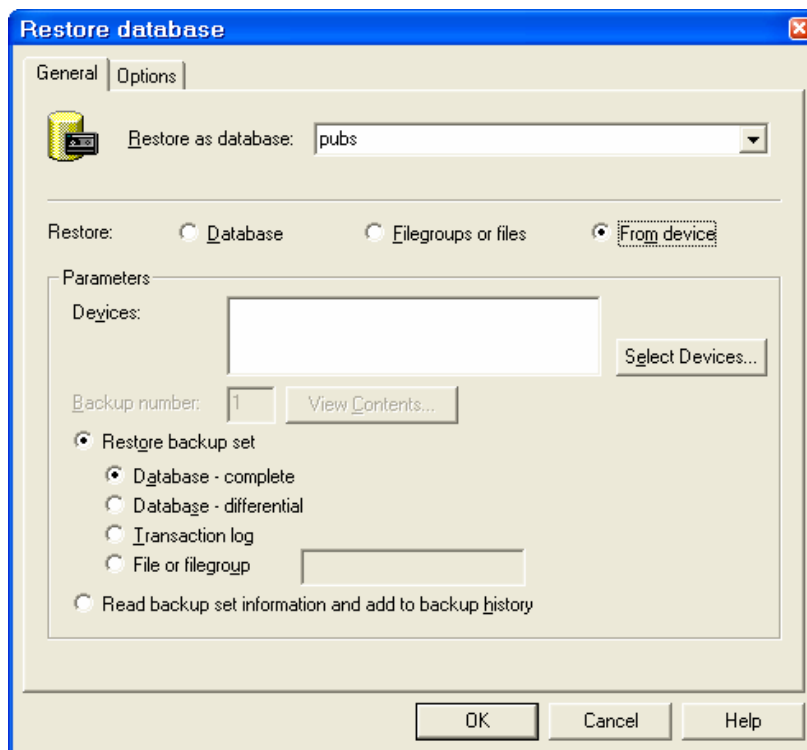
- Nhập các tham số, lựa chọn kiểu.

KHÔI PHỤC DỮ LIỆU – RESTORE DATABASE.

Là chức năng thực hiện khôi phục dữ liệu đã sao lưu, tùy theo chỉ định các backup mà bạn có thể phục hồi dữ liệu ở bất kỳ thời điểm nào, thu hồi dữ liệu trong quá khứ bất kỳ thời nào. Khôi phục dữ liệu thực hiện theo thứ tự backup, thông tin này có lưu trữ trong msdb

Các bước thực hiện như sau:

- Chọn menu Databases -> Nhấn nút phíchút -> All Tasks -> Restore Database...



- Nhập tham số, chọn mô hình khôi phục.

PHÂN QUY N, B O M T

Chương này sẽ giới thiệu về các thuật ngữ phân quyền, quản lý người dùng, và các mô hình bảo mật cho CSDL.

CH B O M T – SECURITY MODE.

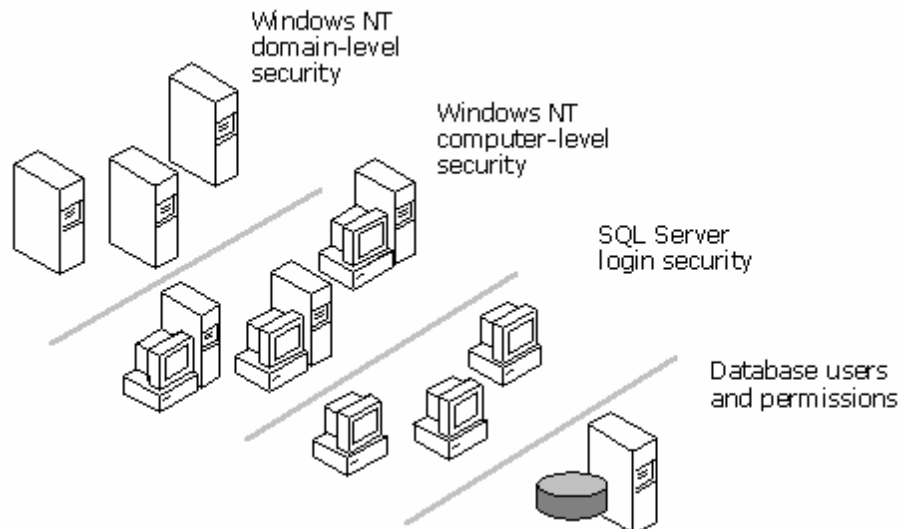
Như đã đề cập trong phần cài đặt SQL Server, SQL Server có 2 chế độ bảo mật:

+ Windows Authentication Mode (Windows Authentication)

+ Mixed Mode (Windows Authentication and SQL Server Authentication)

Windows Authentication.

Là chế độ bảo mật mà người dùng truy cập SQL Server phải là người dùng của Windows. Khi Server được thiết lập chế độ bảo mật này, người dùng phải là người dùng của Windows quản lý mật khẩu truy cập.



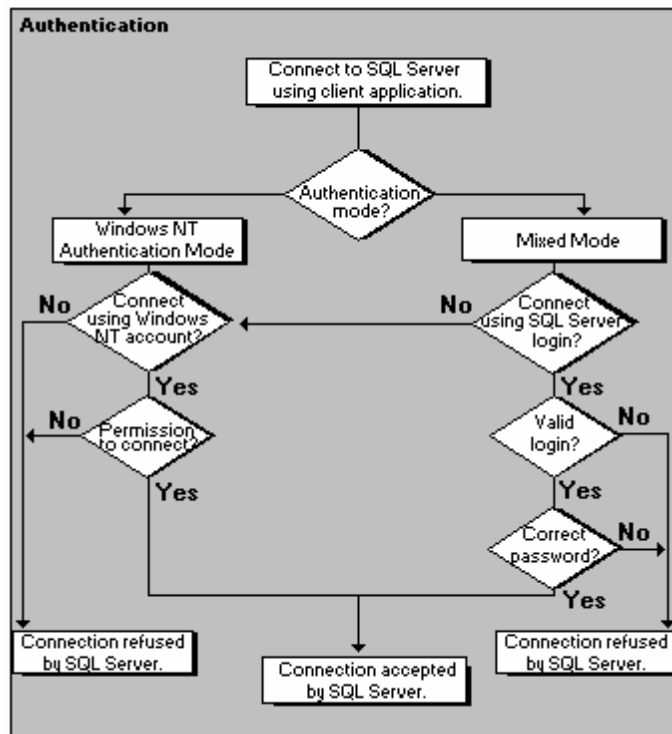
Nhìn trên hình ta thấy khi thiết lập chế độ bảo mật này người sử dụng muốn khai thác SQL Server phải thông qua 4 bước xác thực (1- Domain, 2- Computer, 3- SQL Server, 4- Database).

SQL Server Authentication.

Khi thi t l p ch b o m t này, nh ng User c quy n khai thác ph i là nh ng User do qu n tr SQL Server t o ra, mà nh ng user c a Windows không c khai thác.

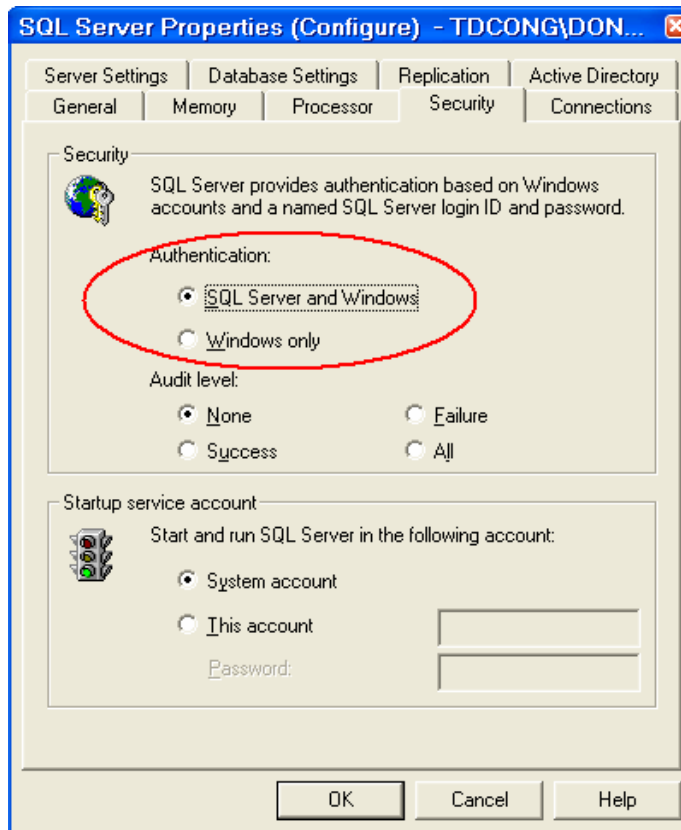
Tuy nhiên, SQL Server cho phép thi t l p hai ch Windows Authentication Mode (Windows Authentication) và Mixed Mode (Windows Authentication and SQL Server Authentication), ch Mixed Mode là s k t h p c a Windows Authentication và SQL Server Authentication, ch này c user c a Windows và SQL Server có th thi t l p truy nh p SQL Server.

SQL Server Security Decision Tree



t ch .

- Nh n ph i chu t ch n tên Server (Instance).
- Ch n Properties.
- Ch n b ng Security.



- Chọn chế độ b o m t -> Ok

SERVER ROLE, DATABASE ROLE.

Role là i t ̣ng xác ̣nh nhóm thu ̣c tính ̣ g ̣n quy ̣n cho các user tham gia khai thác SQL Server.

Server Role.

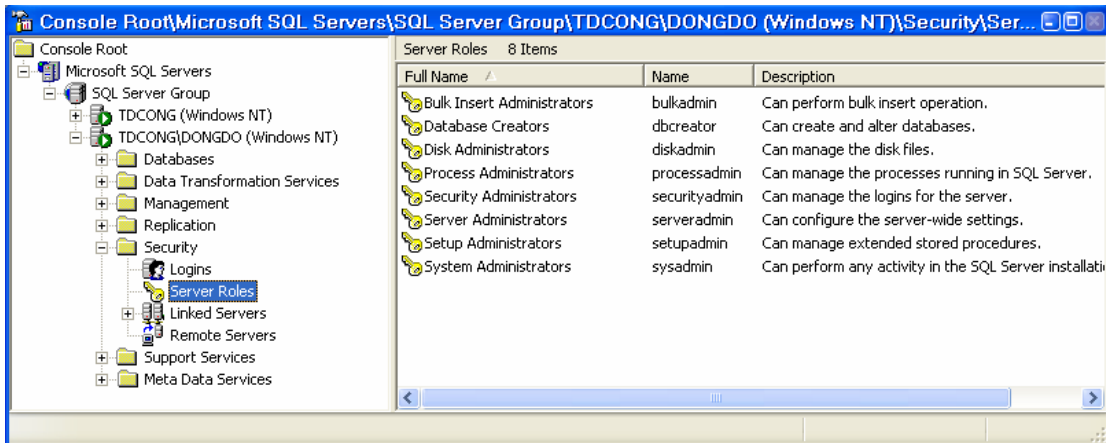
Nhóm các quy ̣n th ̣c hi ̣n qu ̣n tr ̣ h ̣ th ̣ng, g ̣m các nhóm sau:

- + Bulk Insert Administrators: ̣c phép th ̣c hi ̣n Bulk Insert.
- + Database Creators: ̣c phép t ̣o và s ̣a ̣ i c ̣ u trúc CSDL.
- + Disk Administrators: Có th ̣ qu ̣n tr ̣ các file trên ̣ a.
- + Process Administrator: Qu ̣n tr ̣ các d ̣ ch v ̣ ̣ ang ch ̣ y c ̣ a SQL Server.
- + Security Administrators: Qu ̣n tr ̣ h ̣ th ̣ng b o m t.
- + Setup Administrators: Qu ̣n tr ̣ các th ̣ t c m ̣ r ̣ ng (xp_).

+ System Administrators: Quyền truy cập thính SQL Server.

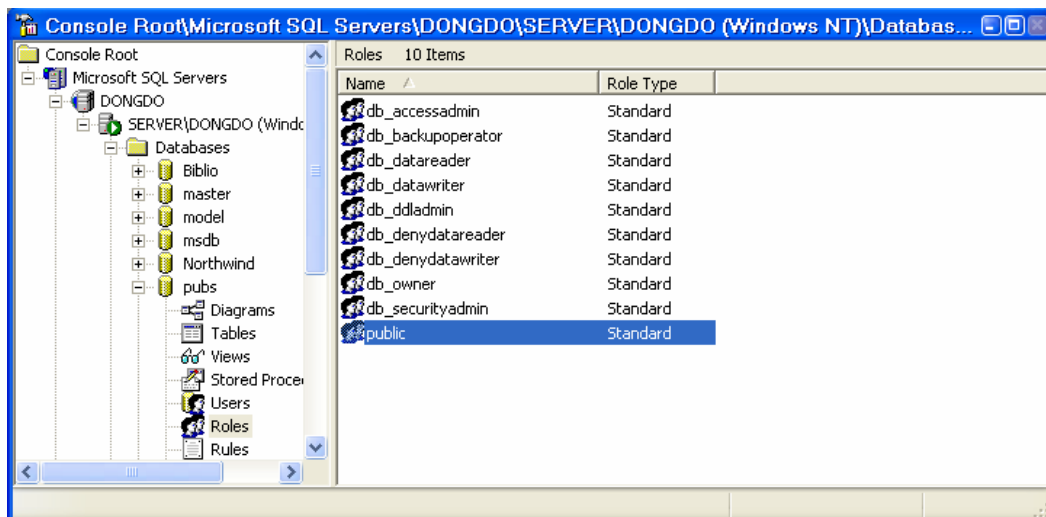
Xem tiếp theo sau:

- M r ng Server (nh n d u '+' ph n tên Server).
- M r ng Security.
- Ch n Server Roles:



Database Role.

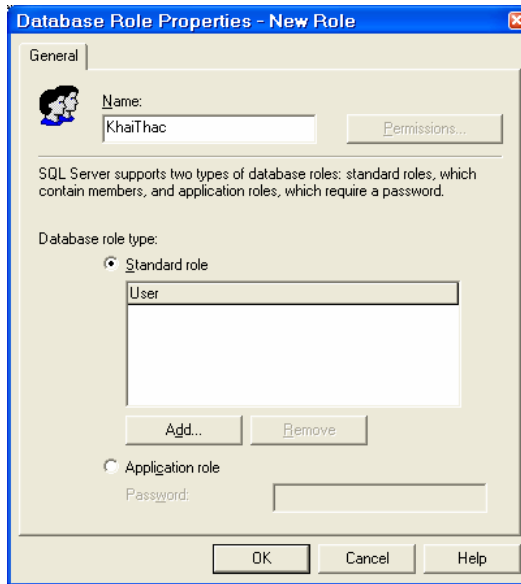
Role là i t ng mà thông qua nó ng i qu n tr có th gán quy n khai thác cho ng i s d ng. Role do CSDL qu n lý, khi t o CSDL h th ng t t m t s Role ng m nh.



Ng i nh ng Role ng m nh ta có th t o Role m i.

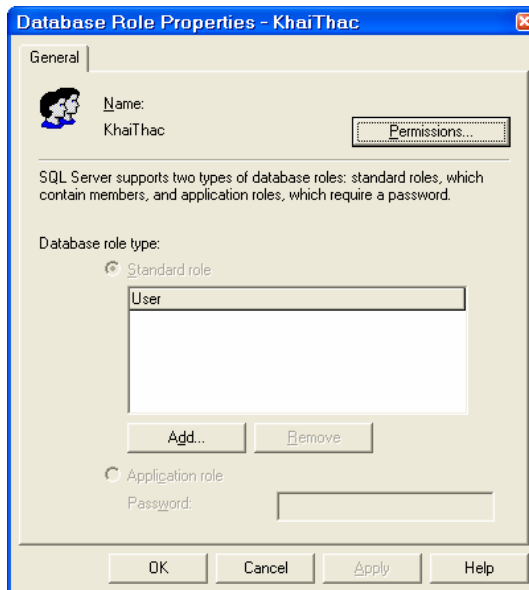
T o Role theo công c .

- Ch n Roles trong CSDL -> Nh n ph i chu t -> New Database Role..
- t tên, ch n user (ch n user có th làm sau).
- Nh n Ok.



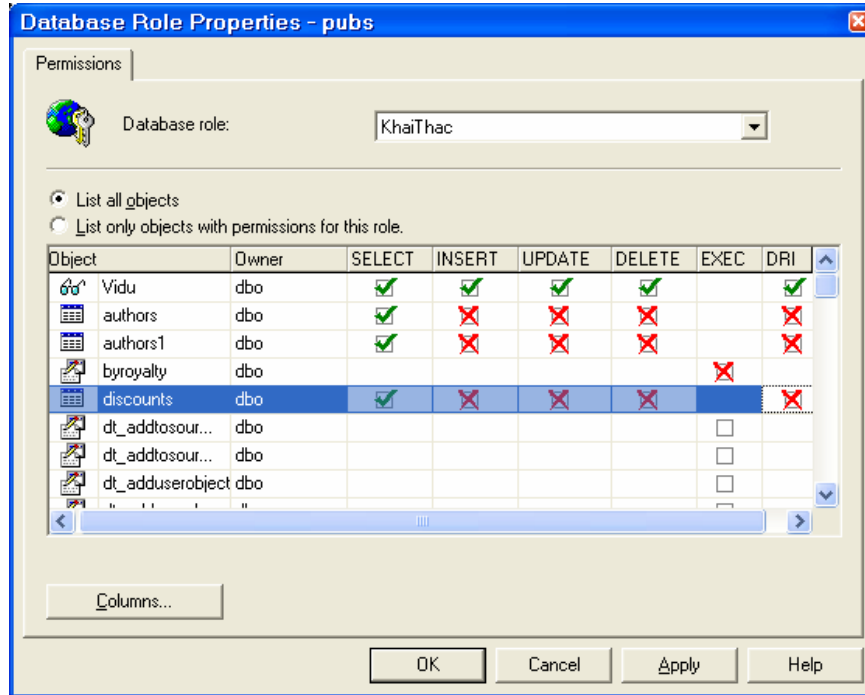
Sau khi t o xong, th c hi n gán quy n khai thác cho Role.

- Ch n Role c n gán quy n.



- Ch n Permissions...

- t các quy n cho t ng i t ng trong CSDL.



N u ch n quy n nh n ô ch n xu t hi n d u ch n màu xanh, n u c m nh n ô ch n xu t hi n d u màu . Có th t quy n khai thác i v i role cho t ng c t c a b ng d li u.

M i thao tác xóa, s a c th c hi n nh các i t ng khác.

T o theo câu l nh.

S d ng câu l nh

```
sp_addrole [ @rolename = ] 'role'
[ , [ @ownername = ] 'owner' ]
```

Ví d : Thêm Role có tên Managers:

```
EXEC sp_addrole 'Managers'
```

QUẢN LÝ NGƯỜI DÙNG.

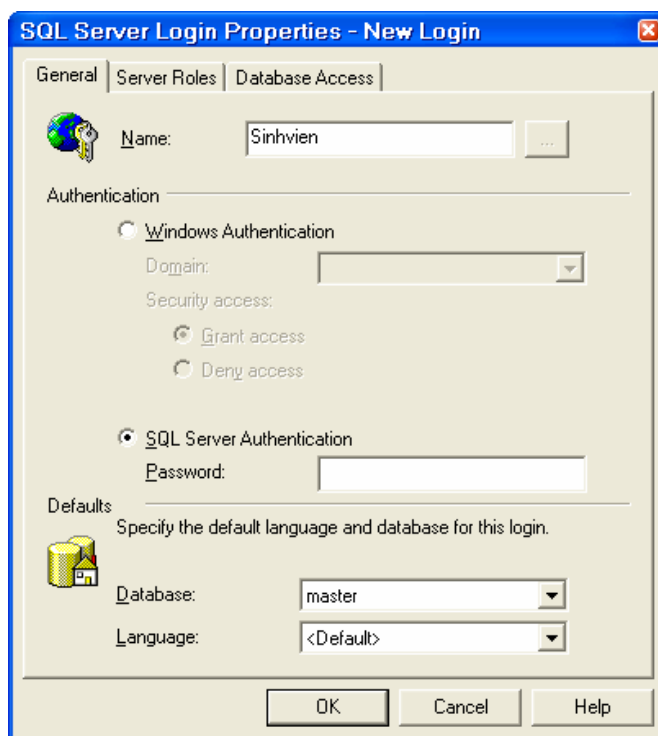
Người dùng trong SQL Server được chia thành 2 mức: Người truy cập vào SQL Server gọi là Login, người khai thác CSDL gọi là User.

Login.

Là một người quản lý truy cập vào SQL Server, tùy theo cách bố trí của SQL Server mà người login là account của Windows NT hay của SQL Server, login do Server quản lý trực tiếp.

Tạo người dùng.

- Chọn chức năng Security của Server -> Logins
- Nhấn phải chuột -> New Login...



- Nhập các tham số: Nếu chọn Account của Windows NT thì bạn có thể chọn trong danh sách. Nếu tạo login của SQL Server thì bạn nhập tên mới, mật khẩu, chọn login thuộc server role nào, có thể gán quyền truy cập khai thác CSDL nào.

Tạo người dùng câu lệnh. Sử dụng câu lệnh

```
sp_addlogin [ @loginame = ] 'login'  
[ , [ @passwd = ] 'password' ]  
[ , [ @defdb = ] 'database' ]  
[ , [ @deflanguage = ] 'language' ]
```



```
[ , [ @sid = ] sid ]  
[ , [ @encryptopt = ] 'encryption_option' ]
```

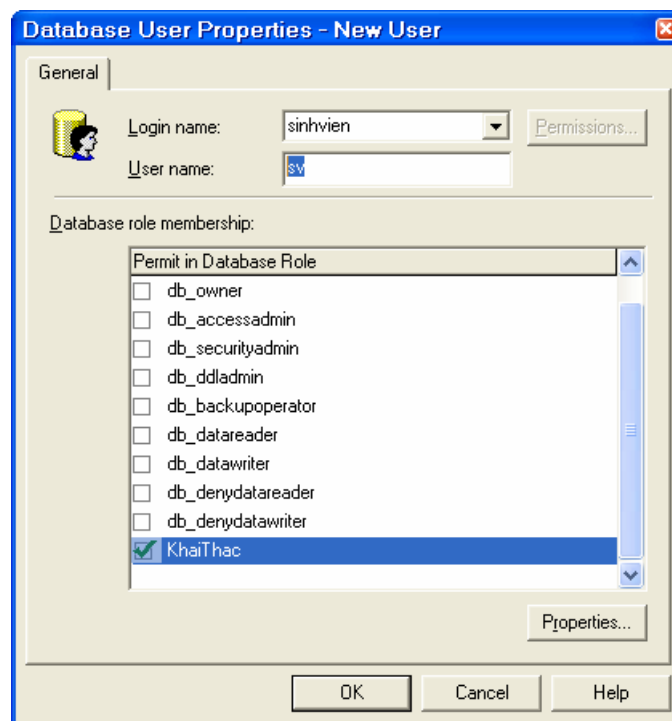
Ví dụ : Tạo login có tên 'Albert', mật khẩu 'corporate'
EXEC sp_addlogin 'Albert', 'food', 'corporate'

Một thao tác sửa, xóa cũng thể hiện những tính năng khác.

User.

User là một tính năng khai thác CSDL, nếu login chỉ xác định truy cập vào SQL Server thì User là login ID tham gia khai thác CSDL, user do CSDL quản lý trực tiếp.

- Chọn CSDL -> users
- Nhấn phải chuột -> new user...



- Chọn Login, nhập user name, chọn role mà user thuộc -> Ok

Các thao tác xóa, sửa cũng thể hiện những tính năng khác, gán quyền cho user bên có thể chỉnh lại user và tạo cho CSDL sau đó vào những vào Permissions.

NHÂN BẢN D L I U

Chương này b n s gi i thi u v i b n k thu t làm gi m l u l ng d li u giao d ch v i SQL Server khi ã c u hình nhi u Server trên m ng.

GI I THI U V NHÂN B N D L I U.

Nhân b n d li u tên ti ng anh g i là Replication, là công c c s d ng copy m t ho c nhi u CSDL n m t ho c nhi u server (SQL Server) khác, các Server c t trong m ng máy tính n i b (LAN), ng i khai thác có th th c hi n truy nh p n CSDL có trong Server c chuy n d li u n. D li u gi a các máy c th c hi n ng b v i nhau theo l ch ho c theo s ki n, khi có yêu c u. Nhân b n d li u có nh ng u i m sau:

- + D li u c l u tr nhi u n i, hi u qu trong vi c có nhi u ng d ng cùng truy nh p, khai thác.

- + Thích h p các ng d ng phân tích d li u OLTP c a DataWare House.

- + Có th khai thác d li u khi không k t n i n Server.

- + Gi m thi u xung kh c do s l ng l n các giao d ch trên m ng.

- + Là m t gi i pháp an toàn khi Server b l i ho c b o d ng.

Mô hình nhân b n.

D ch v nhân b n d li u g m các thành ph n c b n sau: Publisher, Distributor, Subscribers, Publications, Articles, Subscriptions.

Publisher: Là server cung c p d li u nhân b n cho các server khác. M t publisher có th thi t l p nhi u b d li u nhân b n (g i là publication).

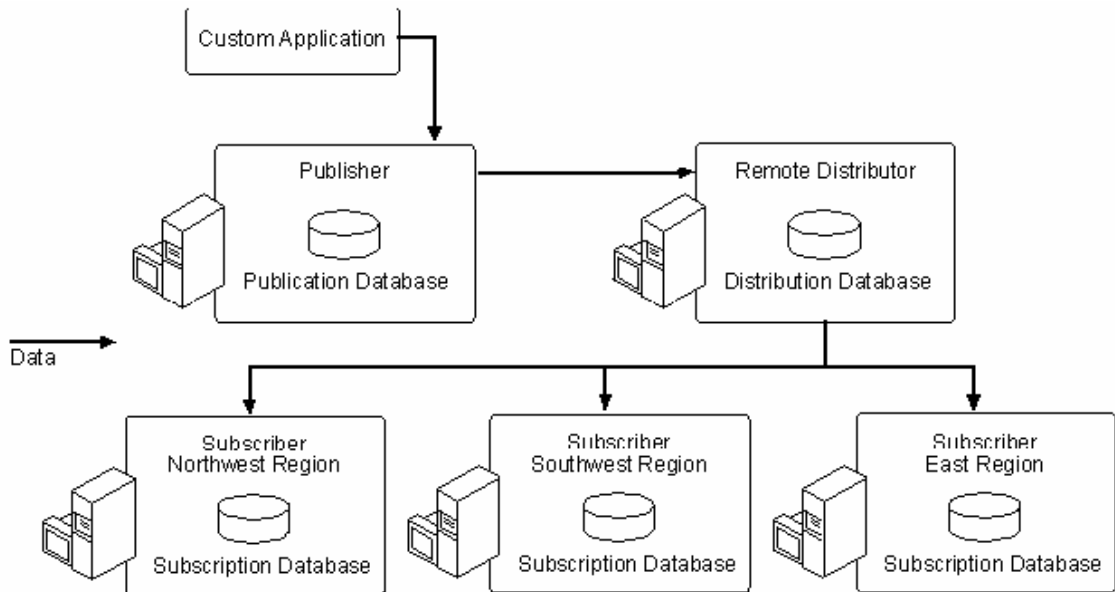
Distributor: Là server qu n lý các thông tin nhân b n, l u tr d li u trong các giao d ch th c hi n nh n và chuy n d li u t Publisher n các Subscriber. Remote distributor là server tách r i kh i publisher và c c u hình là distributor. Local distributor là m t server c c u hình là Publisher và Distributor.

Subscriber: Là server nh n d li u nhân b n. Subscriber g n li n v i publication (là máy ch nh n d li u nhân b n c a m t b d li u c u hình nhân b n).

Article: Là m t b ng, t p d li u ho c i t ng c a CSDL c u hình nhân b n.

Publication: Là m t t p g m m t ho c nhi u article.

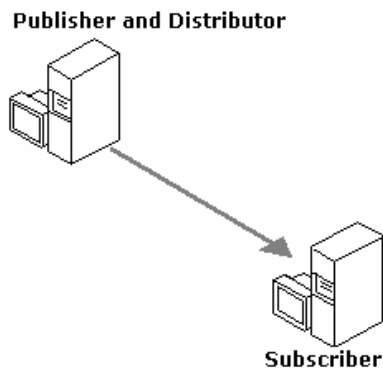
Subscription: Là m t giao d ch yêu c u b n sao b d li u ho c các i t ng c a CSDL th c hi n nhân b n. Trong m i giao d ch publisher th c hi n d y (push subscription) d li u, subscriber th c hi n kéo (pull subscription).



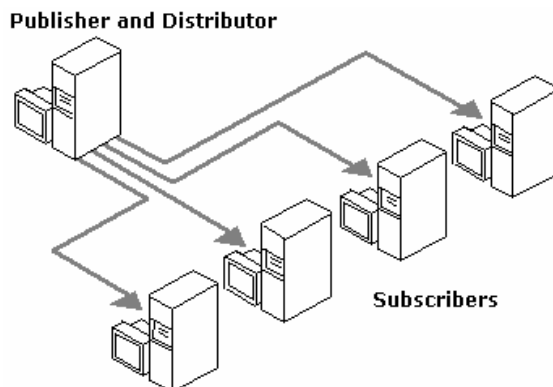
Nhân bản dữ liệu có thể hiển thị theo những mô hình cụ thể sau:

+ *Central Publisher*: Là mô hình Publisher và Distributor thiết lập trên một máy. Gồm các mô hình sau:

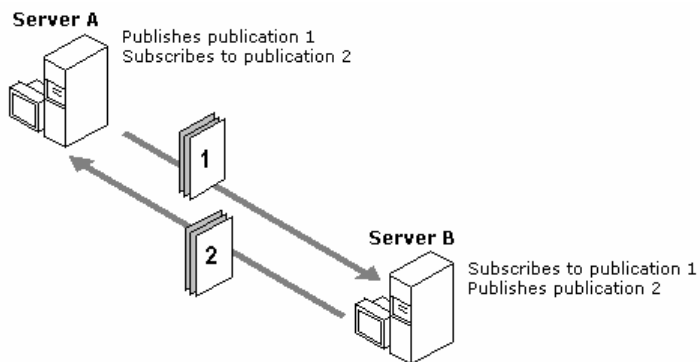
- Một Publishers và một Subscriber:



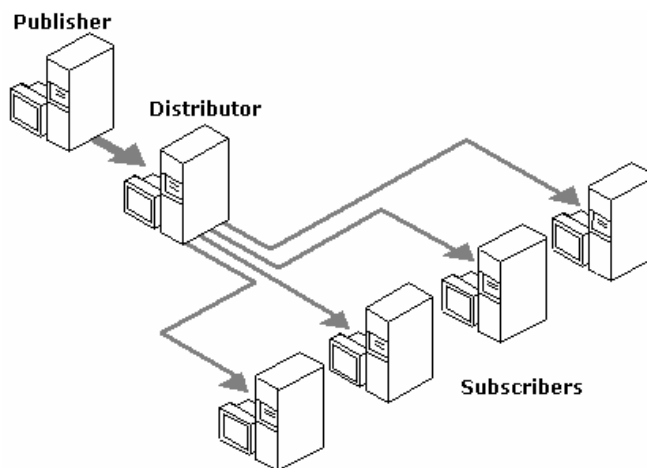
- Một Publisher và nhiều Subscriber.



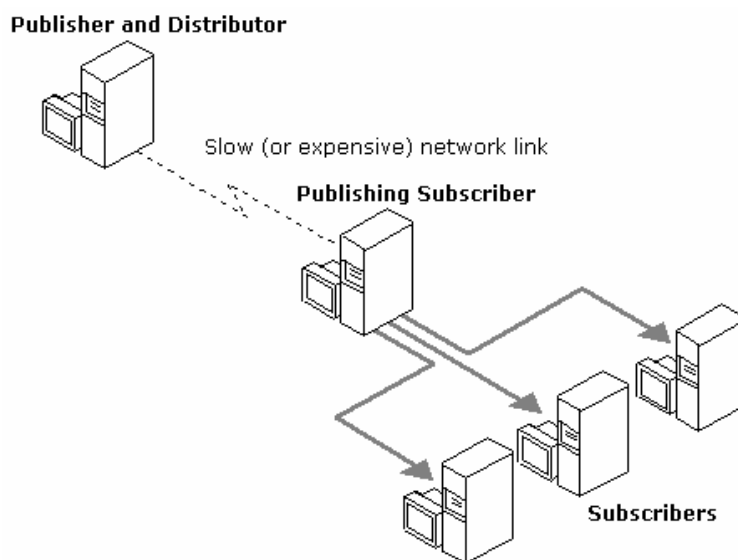
- Publisher và Subscriber có thể tồn tại trên một máy:



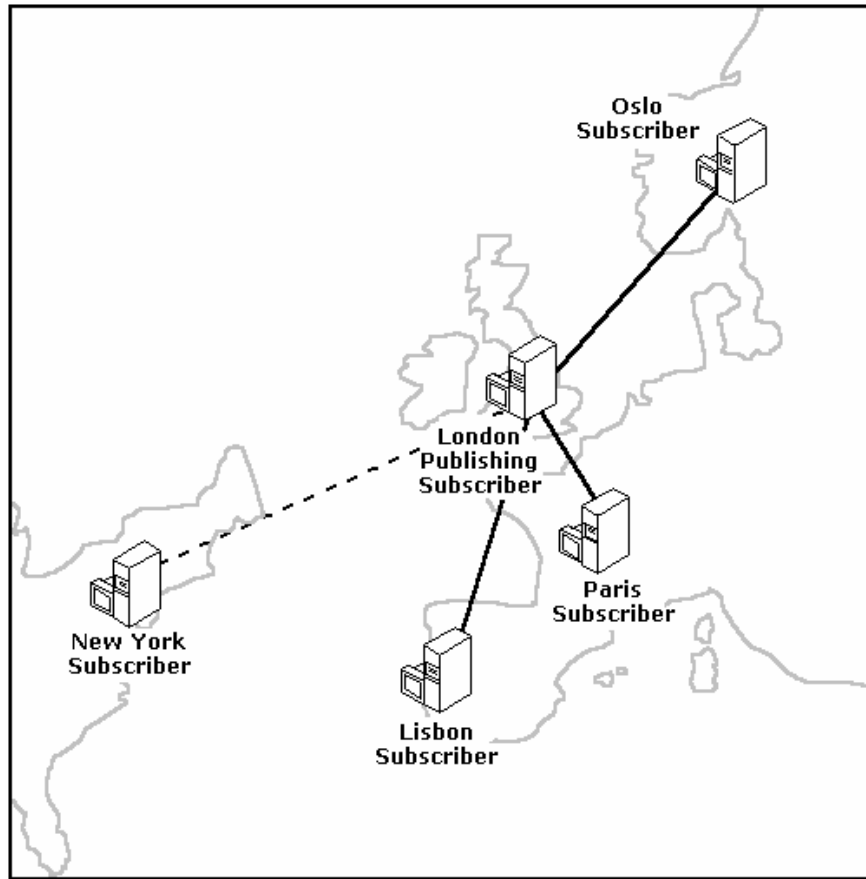
+ *Publisher và Distributor không thể tồn tại trên một máy:*



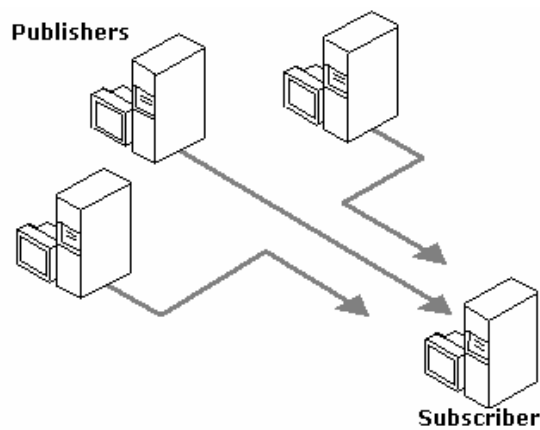
+ *Republisher*: Là mô hình Publisher xuất bản dữ liệu đến Subscriber, sau đó Subscriber có thể tồn tại là Publisher xuất bản dữ liệu đến Subscriber khác.



ng truy n gi a hai máy c thi t l p là Publisher có th t c th p, phù h p v i v trí xa nhau. Ví d mô hình gi a các vùng cách xa nhau:



+ *Central Subscriber*: Là mô hình Subscriber thi t l pn nh n d li u xu t b n t nhi u Publisher.



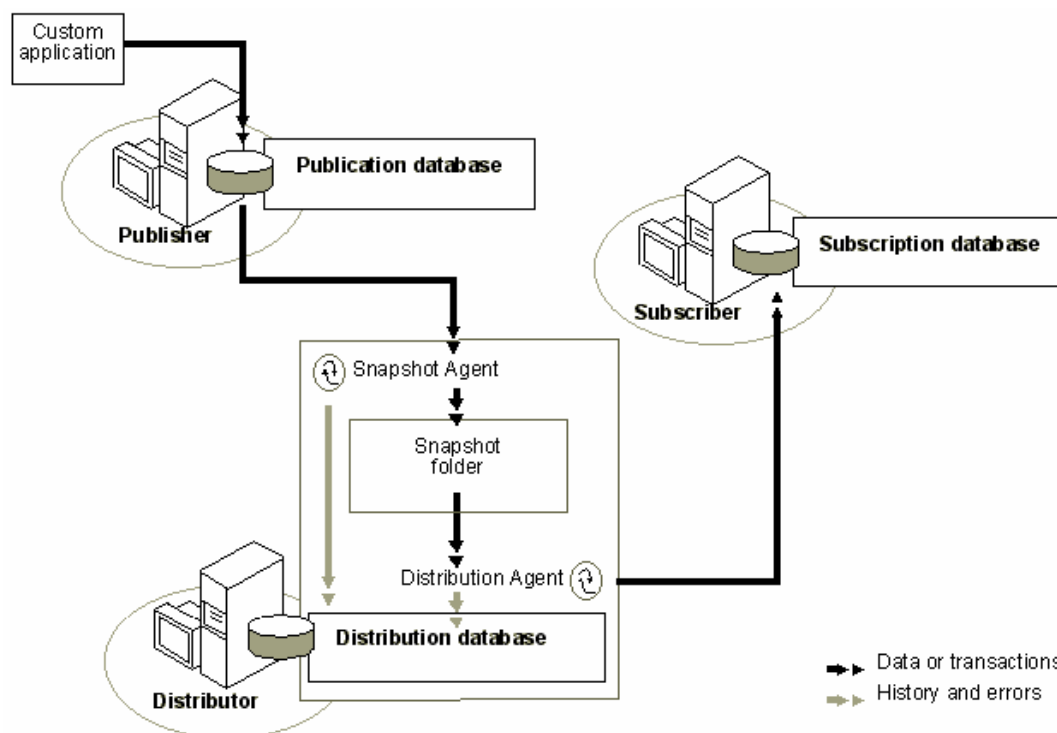
Những kiểu nhân bản dữ liệu.

Có 3 kiểu nhân bản dữ liệu Snapshot, Transaction, Merge.

Snapshot replication: Là kiểu nhân bản thực hiện sao chép, phân tán dữ liệu hoặc các đối tượng của CSDL tới một thời điểm.

Snapshot thường sử dụng cho những tình huống sau:

- + Dữ liệu thường là tĩnh, ít thay đổi.
- + Nhân bản một lần dữ liệu.



Transaction replication: Là kiểu nhân bản mà bắt đầu bằng nhân bản snapshot, sau đó sẽ thực hiện nhân giao dịch dữ liệu theo các sự kiện insert, update, delete và những thay đổi liên quan đến thực hiện stored procedure, index view.

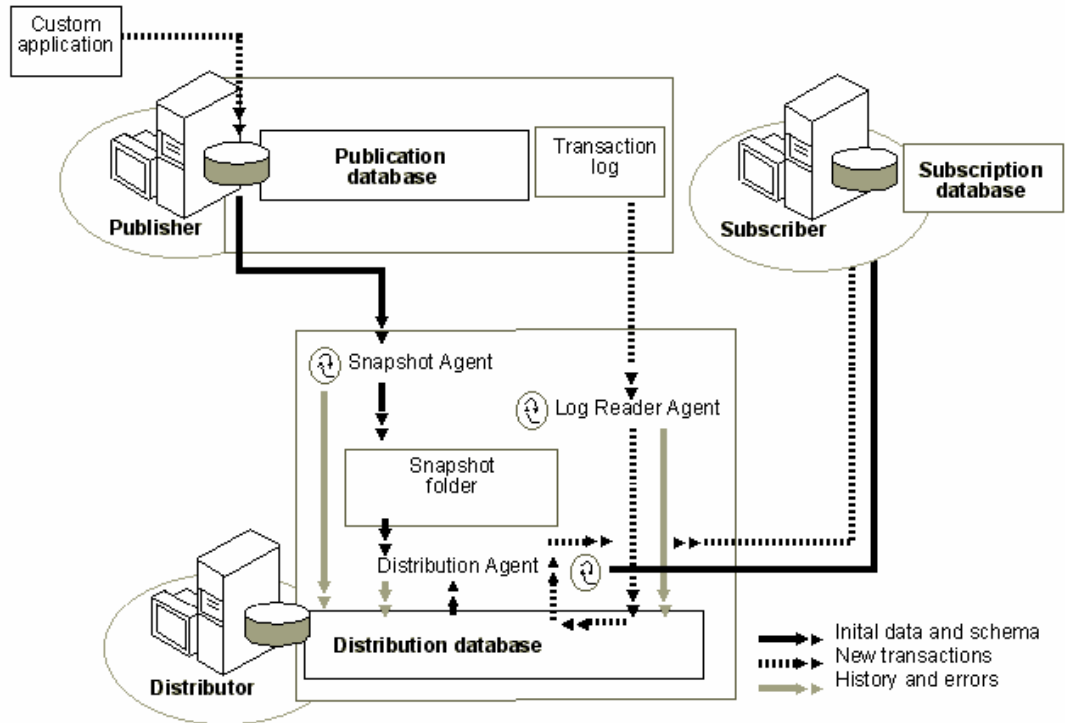
Nhân bản kiểu này cho phép thực hiện load dữ liệu tức thời, cho phép user sử dụng dữ liệu nhân bản tại subscriber và chuyển dữ liệu về lại Publisher hoặc Subscriber khác, dữ liệu sử dụng này có thể coi là dữ liệu thực thời.

Nhân bản kiểu này có thể thực hiện khi:

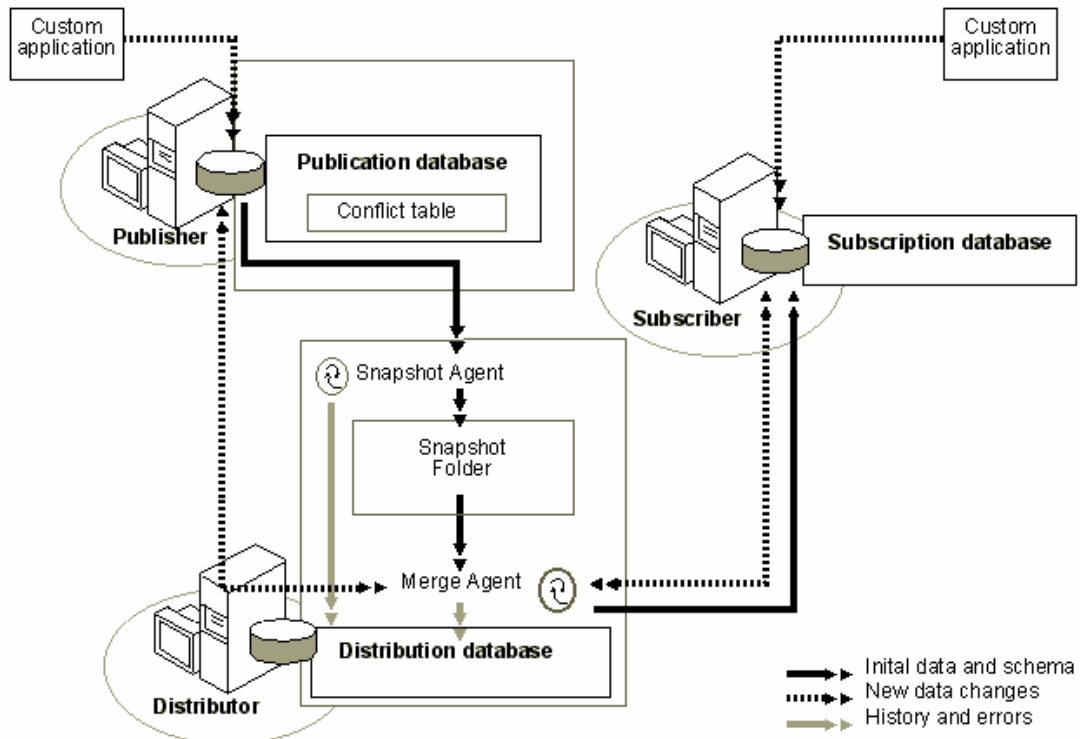
+ Muốn sử dụng dữ liệu thực thời chuyển đến Subscriber, thời gian thực hiện theo giây, hoặc thực thời.

+ Cần giao dịch trên toàn bộ hệ thống nhân bản dữ liệu (dữ liệu có thể chuyển đến tất cả các Subscriber hoặc không chuyển đến Subscriber nào).

+ Subscriber không xuyên kết nối với Publisher.



Merge replication: Là kỹ thuật nhân bản dữ liệu cho phép thực hiện nhân số dữ liệu trên nhiều Subscriber, có thể kết nối (online) hoặc không kết nối (offline) với



Publisher. D li u s c ng b theo l ch ho c theo yêu c u, d li u c p nh t có th i i m sau s c ch p nh n.

Ki u nh n b n này th c hi n khi:

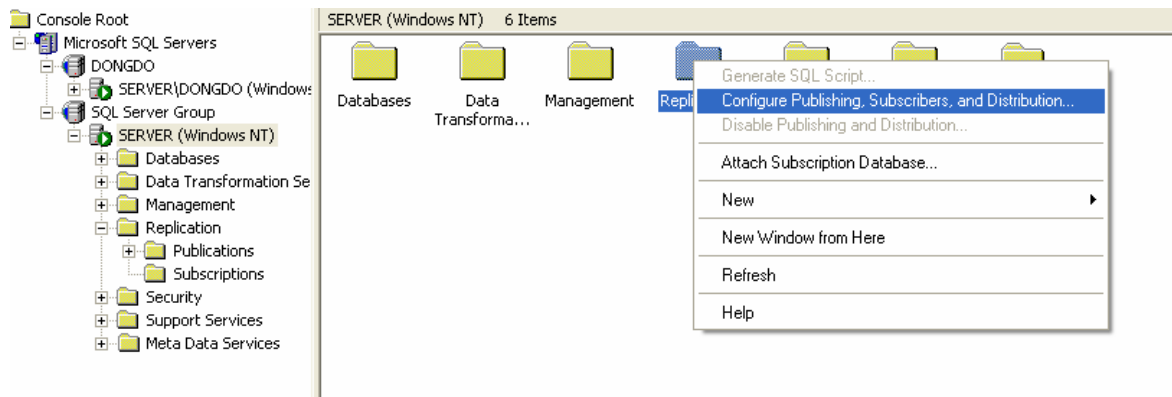
+ Nhi u Subscriber có nhu c u c p nh t d li u và chuy n d li u c p nh t n Publisher ho c Subscriber khác.

+ Subscriber yêu c u nh n ho c chuy n d li u khi offline, ng b d li u v i các Subscriber và Publisher sau.

C U HÌNH PUBLISHER VÀ DISTRIBUTOR.

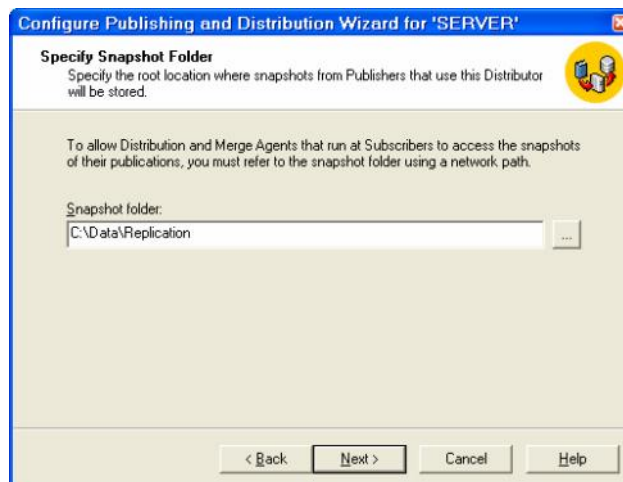
Tr c khi th c hi n c u hình các máy thành Publisher hay Distributor ta ph i th c hi n ch y d ch v SQL Server Agent trong ch c n ng Service manager. các b c c u hình nh sau:

- Ch n Server c n c u hình -> Replication
- Nh n ph i chu t -> Configure Publishing Subscription and Distribution...

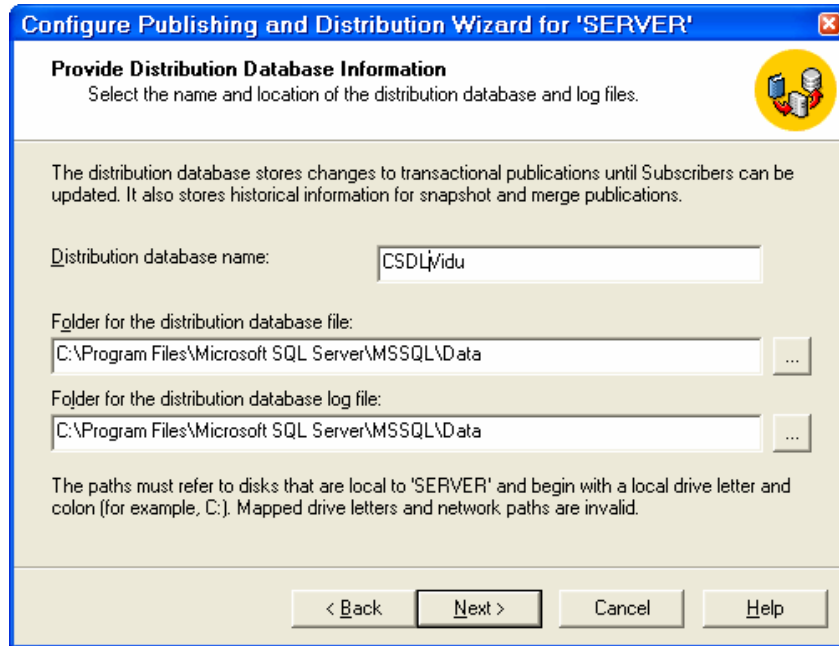


- Th c hi n thao các b c:

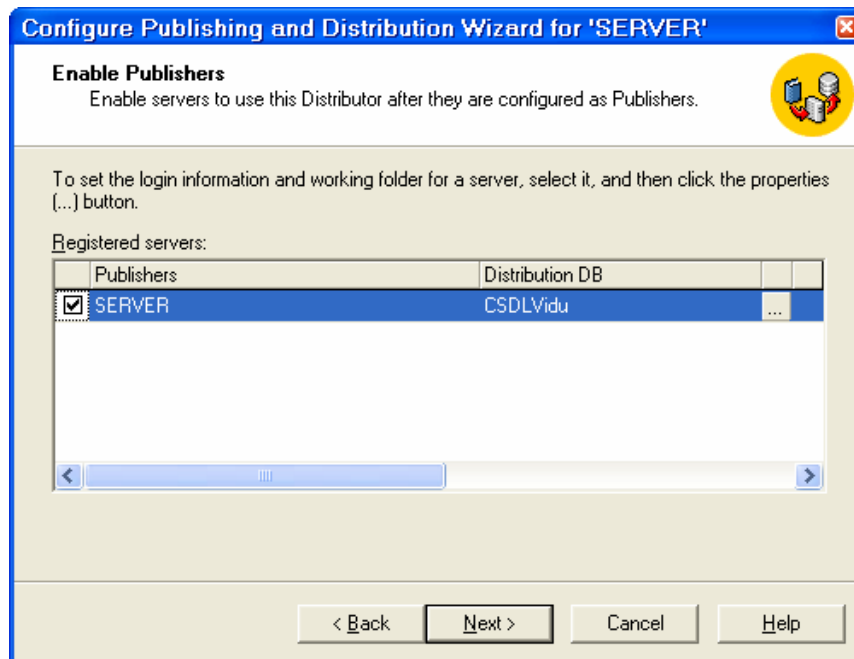
+ Ch n th m c Snapshot: Th m c này s s d ng cho



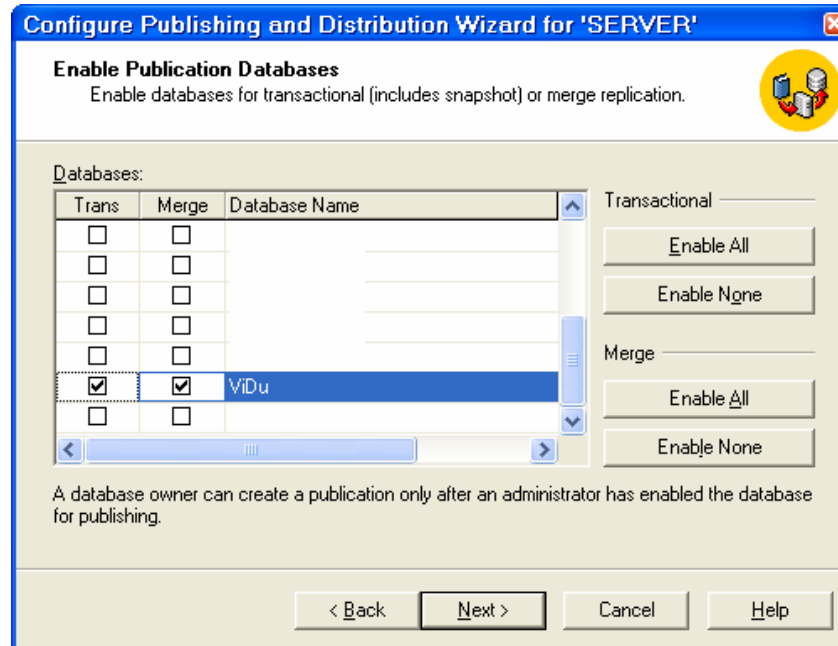
- t tên CSDL c a Distribution.



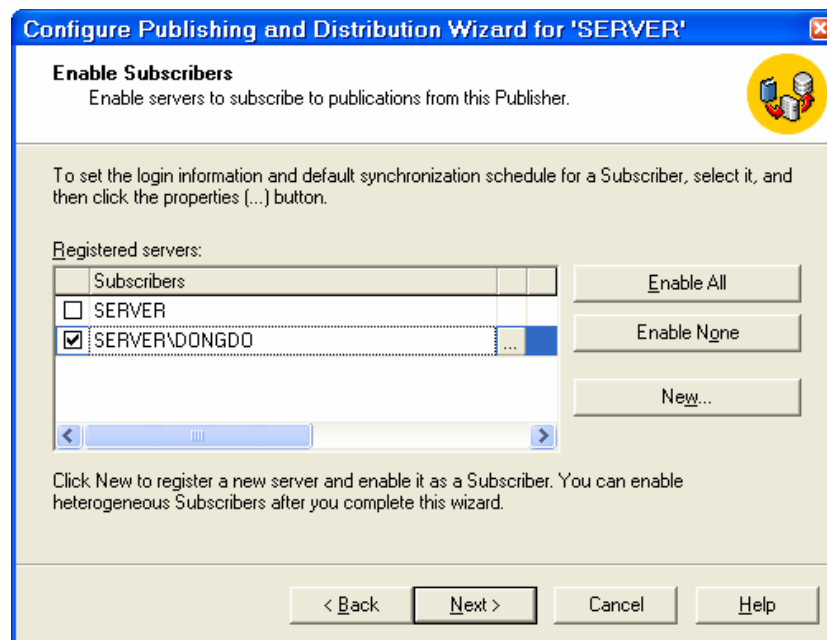
- Ch n Server c u hình thành Publisher.



- Chọn CSDL tham gia nhân bản, kiểu nhân bản.



- Chọn Server để cấu hình là Subscriber của Publisher đang thiết lập.



- Kết thúc.

TỔNG KẾT

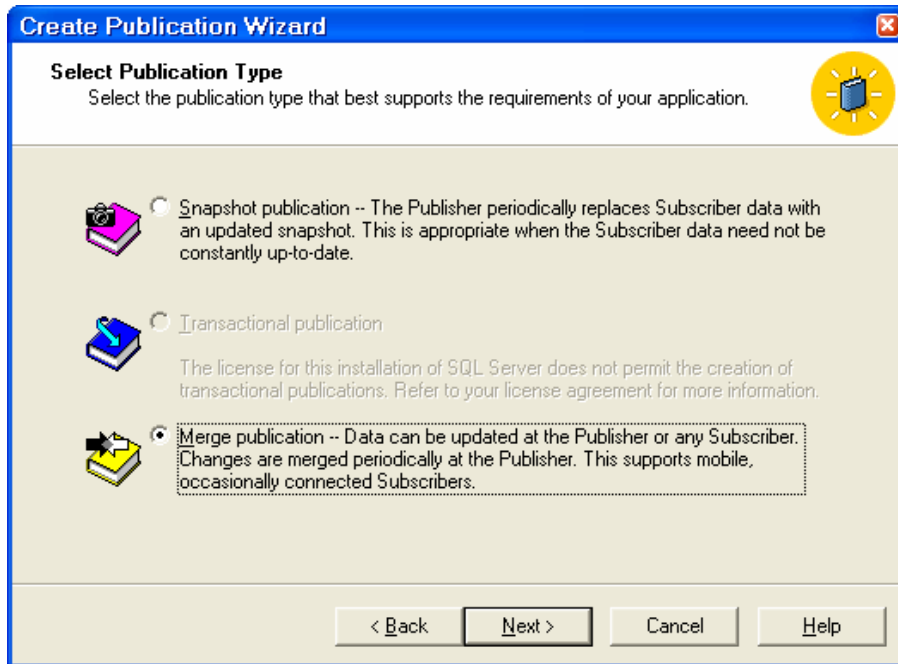
Bài này sẽ hướng dẫn cách thiết lập Publication, cách thức như sau:

+ Chọn Publication trong Replication của Publisher.

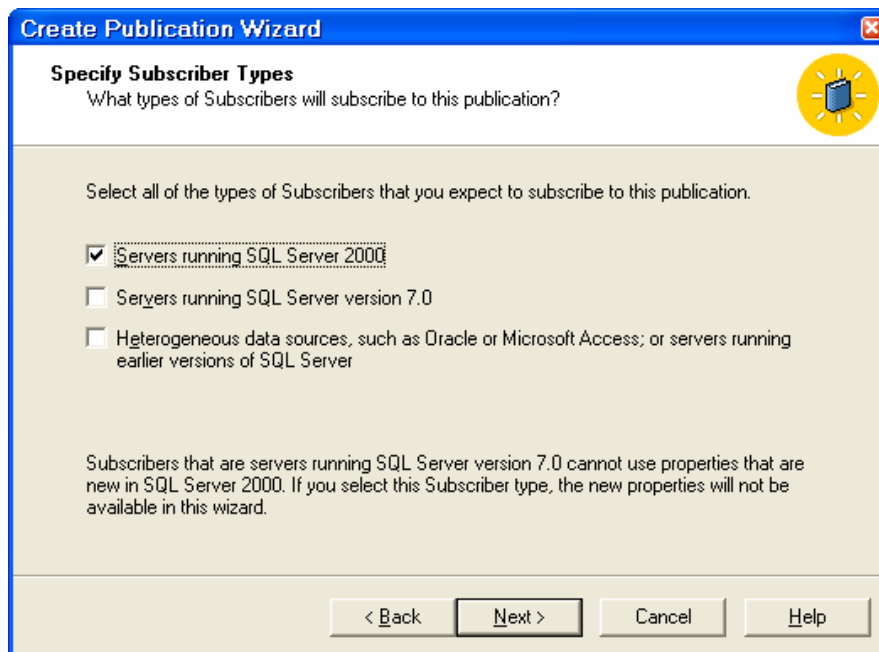
+ Nhấn phím tắt -> New Publication...

+ Th c hi n theo các b c:

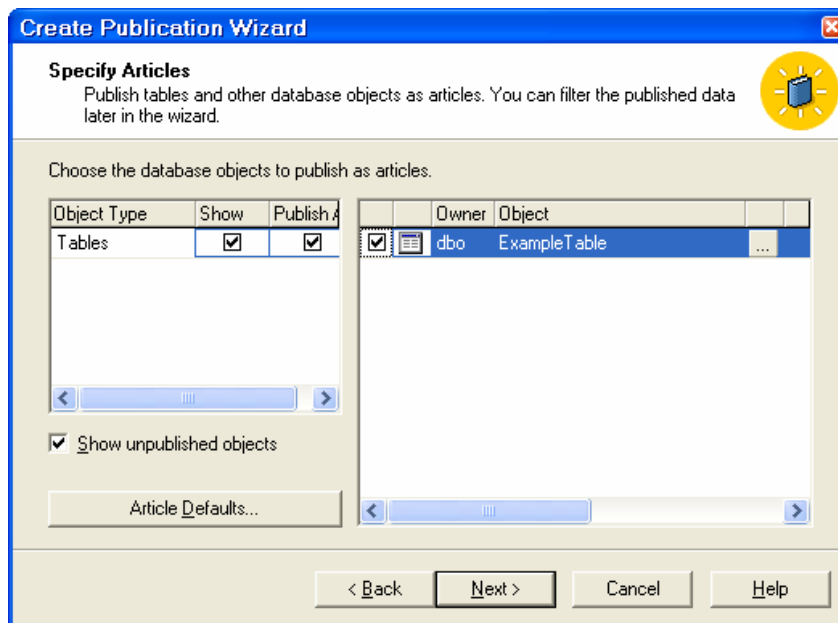
- Ch n CSDL c n xu t b n d li u ho c i t ng.
- Ch n ki u nhân b n (trong ví d này th c hi n ki u Merge)



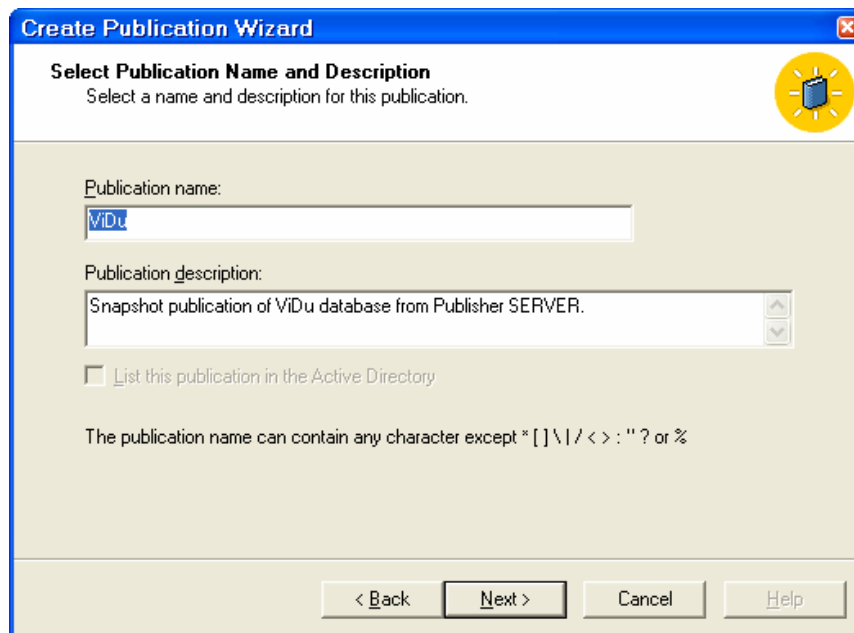
- Ch n phiên b n SQL Server c a Subscriber.



- Ch n Article tham gia Publication.



- Đặt tên cho Publication.



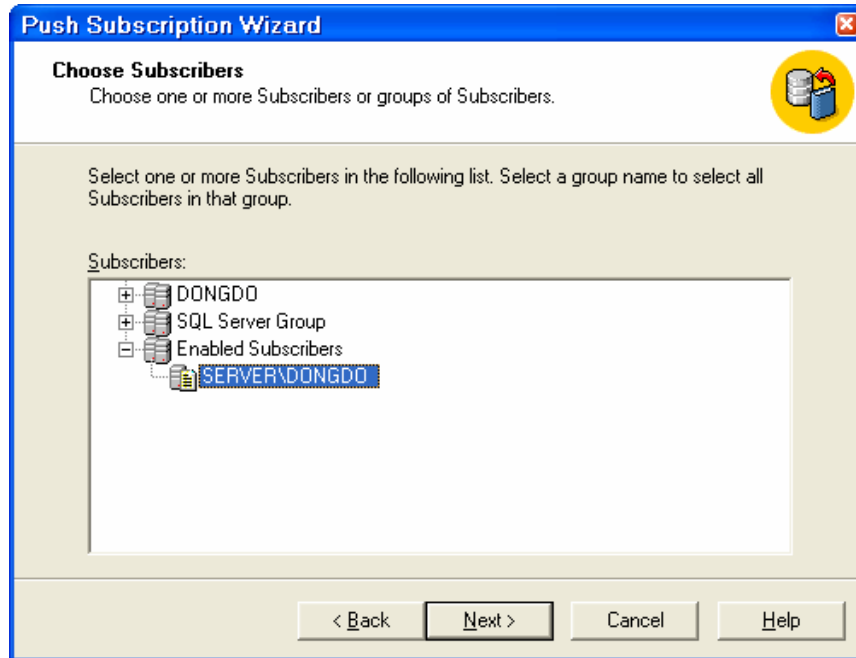
- Kết thúc.

TỔNG KẾT PUSH SUBSCRIPTION.

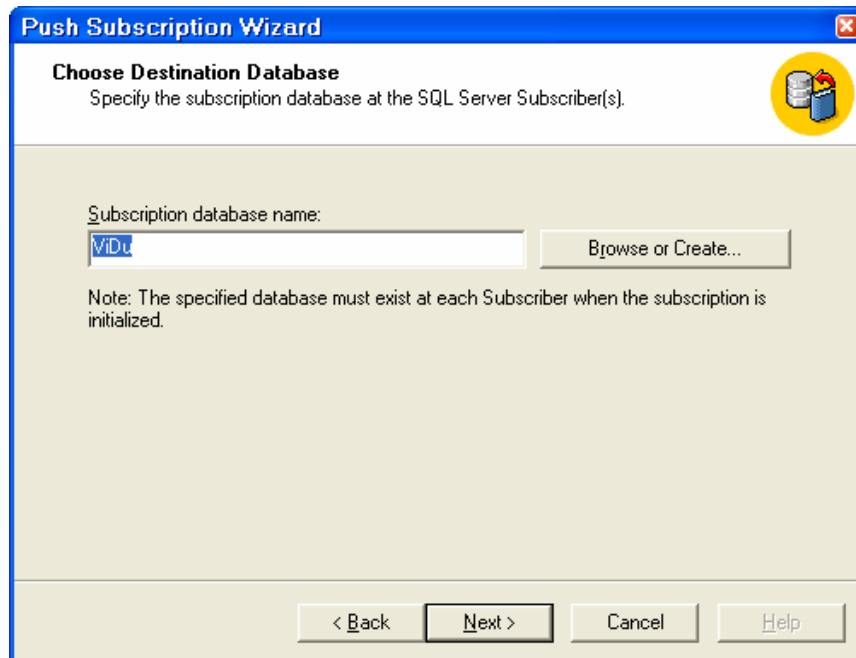
Bước này thể hiện một cách đẩy (push) từ Publisher (Distributor trong ví dụ này) đến Subscriber, các bước thể hiện trên Publisher. Các bước thể hiện như sau:

- Chọn Publication của Publisher -> Nhấn nút Push new Subscription...

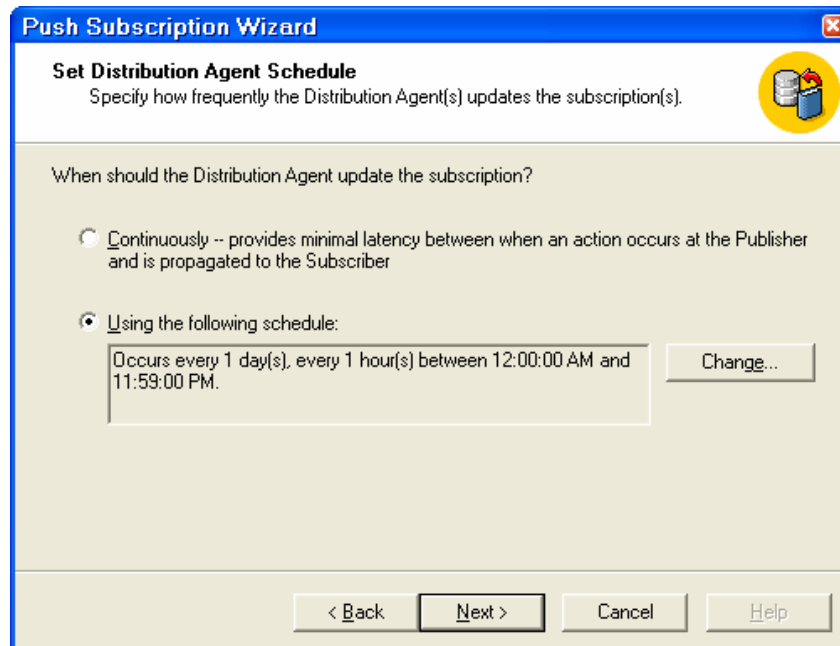
- Chọn Subscriber.



- Chọn CSDL trên Subscriber nếu đã có, nếu chưa có thì click nút 'Browse or Create...' để tạo mới.



- Chọn lịch thời gian bắt đầu lưu.



- Kết thúc. Sau khi thiết lập xong trên Subscriber sẽ có CSDL theo tên đã tạo.

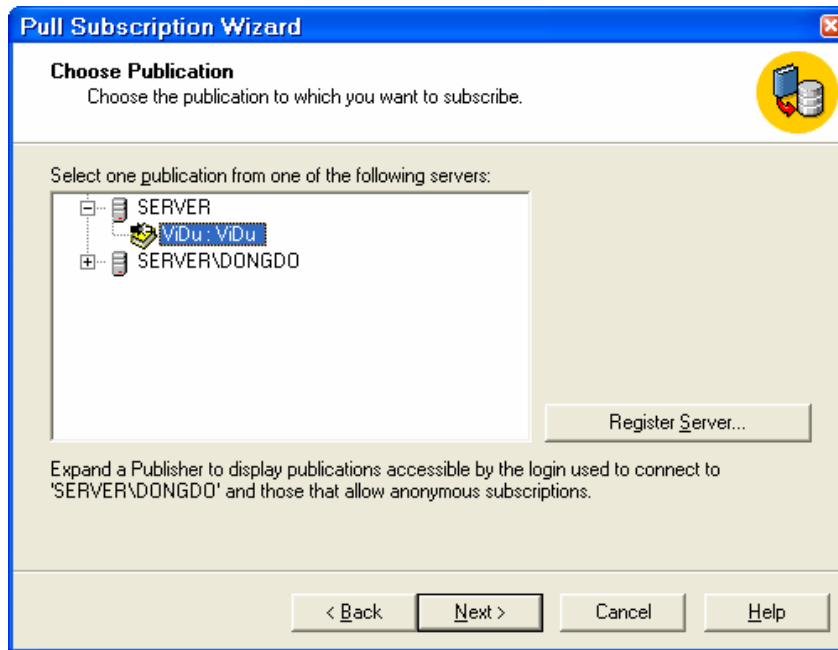
TẠO PULL SUBSCRIPTION.

Bước này thể hiện tạo công việc kéo dữ liệu nhân bản từ Publisher về Subscriber, thể hiện trên Subscriber.

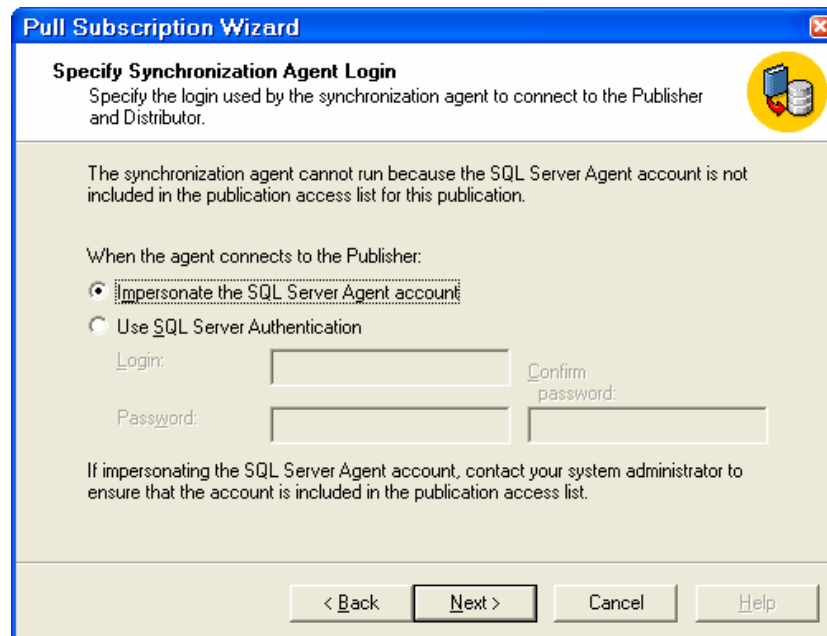
- Chọn Subscription của Subscriber -> Nhấn phải chuột -> New Pull Subscription...

- Thể hiện theo các bước:

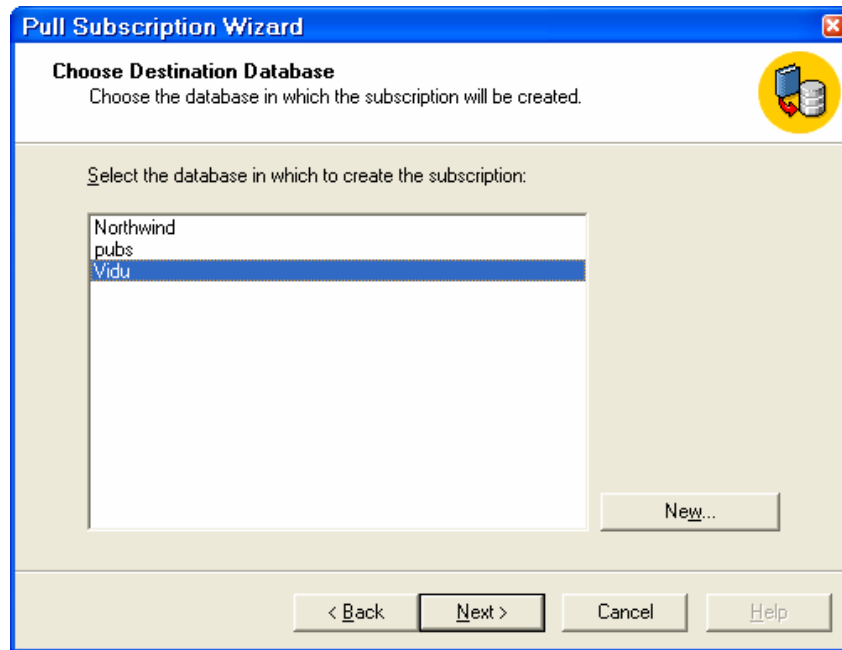
+ Chọn Publication.



- Chọn Agent tham gia kết nối Publisher.



- Chọn CSDL đích.



- Th c hi n ti p các b c và k t thúc.

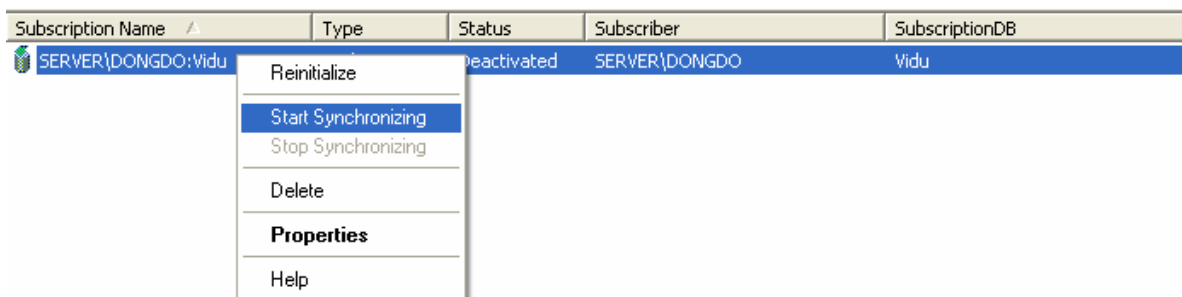
N u ã t o Push Subscription v i m t CSDL s không c t o Pull Subscription v i CSDL ó.

TH C HI N NG B D LI U.

Sau khi thi t l p theo các mô hình nhân b n xong, b n có th th c hi n ng b d li u b ng cách:

- Th c hi n theo l ch.

- Theo yêu c u: Ch n Subscription (Push ho c Pull) -> Nh n ph i chu t -> Start Synchronizing



Sau khi th c hi n xong d li u s c ng b gi a Publisher và Subscriber. Ngoài th c hi n theo công c b n có th tìm hi u th c hi n nhân b n theo câu l nh T-SQL ho c Stored Procedure.

Ph n 2. CÂU L NH T-SQL

Trong ph n này s gi i thi u c u trúc, k thu t so n k ch b n l nh T-SQL, i v i các h qu n tr CSDL Foxfro, Access thì câu l nh th c hi n truy v n, khai thác CSDL là ngôn ng truy v n SQL (Structure Query Language), các l nh c th c hi n theo t ng câu l nh mà không th c hi n theo k ch b n ho c theo t p h p nhi u câu l nh v i nhau. i v i h qu n tr CSDL Oracle thì ngôn ng truy v n đ li u là SQL/PL (SQL Plus), còn SQL Server ngôn ng có tên Transact-SQL vì t t t là T-SQL.

NH NGH A D LI U (DATA DEFINITION LANGUAGE - DDL).

Ph n này s xem xét các l nh liên quan n t o m i, s a i, xóa các i t ng liên quan n Table, View và các i t ng khác.

T o k i u d li u m i.

T o k i u d li u đ ng user-defined.

Cú pháp:

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

Ví d :

```
sp_addtype ssn, 'varchar(11)', 'NOT NULL'
```

Xóa k i u d li u ã t o.

Cú pháp:

```
sp_droptype [ @typename = ] 'type'
```

Ví d :

```
Sp_droptype ssn
```

T o ràng bu c (Constraint).

T o ràng bu c c th c hi n trong 2 câu l nh Create Table ho c Alter Table: Check, Default, Foreign Key, Primary Key, Unique.

Xét m t s ví d sau:

+ T o m t Check. trong b ng authors.

```
ALTER TABLE authors ADD CONSTRAINT chau_id CHECK(au_id
LIKE '[0-9][0-9][0-9]-[0-9][0-9]- [0-9][0-9] [0-9][0-9]')
```

+ T o Check trong b ng Publishers.

```
ALTER TABLE publishers ADD chpub_id CHECK(pub_id IN ('1389',
'0736', '0877', '1622', '1756') OR pub_id LIKE '99[0-9][0-9]')
```

+ T o ràng bu c Default.

```
ALTER TABLE authors ADD DEFAULT 'UNKNOWN' for au_lname
```

+ T o ràng bu c Foreign Key.

```
ALTER TABLE titles ADD CONSTRAINT FK_pub_id FOREIGN
KEY(pub_id) REFERENCES publishers(pub_id)
```

+ T o ràng bu c Primary Key.

```
ALTER TABLE authors ADD CONSTRAINT UPKCL_auind PRIMARY
KEY CLUSTERED (au_id)
```

+ T o ràng bu c Unique.

```
ALTER TABLE stores ADD CONSTRAINT UNC_name_city UNIQUE
NONCLUSTERED(store_name, city)
```

Xóa ràng bu c.

S d ng Drop trong các câu l nh Create Table ho c Alter Table.

+ Ví d xóa Constraint s d ng câu l nh Alter Table.

```
ALTER TABLE authors DROP CONSTRAINT UPKCL_auind
```

Hi n th ràng bu c.

sp_helpconstraint titltes

T o b ng.

t o b ng d li u có th s d ng 2 câu l nh Create Table ho c Select Into.

+ T o b ng t m th i local (là b ng ch hi n v i phiên hi n th i, tên b ng c b t u b ng m t d u #).

```
CREATE TABLE #MyTempTable (cola INT PRIMARY KEY)
INSERT INTO #MyTempTable VALUES (1)
```

+ T o b ng t m th i global (hi n v i t t c các phiên, tên b ng c b t u b ng 2 d u #).

```
CREATE TABLE ##MyTempTable (cola INT PRIMARY KEY)
```

```

INSERT INTO ##MyTempTable VALUES (1)

+T o b n g d l i u.
/* ***** jobs table ***** */
CREATE TABLE jobs
(
    job_id smallint
        IDENTITY(1,1)
        PRIMARY KEY CLUSTERED,
    job_desc varchar(50) NOT NULL
        DEFAULT 'New Position - title not formalized
yet',
    min_lvl tinyint NOT NULL
        CHECK (min_lvl >= 10),
    max_lvl tinyint NOT NULL
        CHECK (max_lvl <= 250)
)

/* ***** employee table */
CREATE TABLE employee
(
    emp_id empid
        CONSTRAINT PK_emp_id PRIMARY KEY NONCLUSTERED
        CONSTRAINT CK_emp_id CHECK (emp_id LIKE
            '[A-Z][A-Z][A-Z][1-9][0-9][0-9][0-9][0-9][FM]'
or
            emp_id LIKE '[A-Z]-[A-Z][1-9][0-9][0-9][0-9][0-9][0-9][FM]'),
    fname varchar(20) NOT NULL,
    minit char(1) NULL,
    lname varchar(30) NOT NULL,
    job_id smallint NOT NULL
        DEFAULT 1
        REFERENCES jobs(job_id),
    job_lvl tinyint
        DEFAULT 10,
    pub_id char(4) NOT NULL
        DEFAULT ('9952')
        REFERENCES publishers(pub_id),
    hire_date datetime NOT NULL
        DEFAULT (getdate())
)

/* ***** publishers table *** */

```

```

CREATE TABLE publishers
(
    pub_id char(4) NOT NULL
        CONSTRAINT UPKCL_pubind PRIMARY KEY CLUSTERED
        CHECK (pub_id IN ('1389', '0736', '0877',
'1622', '1756')
            OR pub_id LIKE '99[0-9][0-9]'),
    pub_name varchar(40) NULL,
    city varchar(20) NULL,
    state char(2) NULL,
    country varchar(30) NULL
        DEFAULT('USA')
)

```

Xóa bảng.

Sử dụng lệnh Drop Table.

+ Xóa bảng trong CSDL hiện tại:

```
Drop Table MyTable
```

+ Xóa bảng trong CSDL khác.

```
DROP TABLE pubs.dbo.authors2
```

Đổi tên bảng.

Sử dụng thủ tục sp_rename

+ Đổi tên bảng:

```
Sp_rename titles, books
```

Sửa cấu trúc bảng.

Sử dụng lệnh Alter Table.

+ Thêm cột vào bảng.

```

CREATE TABLE doc_exa ( column_a INT)
GO
ALTER TABLE doc_exa ADD column_b VARCHAR(20) NULL
GO
EXEC sp_help doc_exa
GO
DROP TABLE doc_exa
GO

```

+ Xóa cột khỏi bảng.

```

CREATE TABLE doc_exb ( column_a INT, column_b
VARCHAR(20) NULL)

```

```

GO
ALTER TABLE doc_exb DROP COLUMN column_b
GO
EXEC sp_help doc_exb
GO
DROP TABLE doc_exb
GO

```

T o Index.

S d ng l nh Create Index.

+ T o Index.

```

SET NOCOUNT OFF
USE pubs
IF EXISTS (SELECT name FROM sysindexes
           WHERE name = 'au_id_ind')
  DROP INDEX authors.au_id_ind
GO
USE pubs
CREATE UNIQUE CLUSTERED INDEX au_id_ind
  ON authors (au_id)
GO

```

Xem thông tin Index.

S d ng th t c sp_helpindex

+ Xem Index c a b ng authors.

```
sp_helpindex authors
```

Xóa Index.

S d ng l nh Drop Index.

+ Xóa Index c a b ng authors.

```
DROP INDEX authors.au_id_ind
```

T o khung nhìn.

S d ng l nh Create View.

+ T o View.

```

USE pubs
IF EXISTS (SELECT TABLE_NAME FROM
           INFORMATION_SCHEMA.VIEWS
           WHERE TABLE_NAME = 'titles_view')

```

```

    DROP VIEW titles_view
GO
CREATE VIEW titles_view
AS
SELECT title, type, price, pubdate
FROM titles
GO

```

Xóa khung nhìn.

Sử dụng lệnh Drop View.

+ Xóa khung nhìn.

```

USE pubs
IF EXISTS (SELECT TABLE_NAME FROM
INFORMATION_SCHEMA.VIEWS
WHERE TABLE_NAME = 'titles_view')
    DROP VIEW titles_view
GO

```

Đổi tên khung nhìn.

Sử dụng lệnh thay đổi tên sp_rename.

+ Đổi tên view.

```

sp_rename titles_view, view_titles

```

THAO TÁC VỚI DỮ LIỆU (DATA MANIPULATION LANGUAGE - DML).

Phần này sẽ xem xét các câu lệnh thao tác với dữ liệu như Insert, Select, Delete.

Lệnh Insert - Chèn dữ liệu vào bảng.

Sử dụng câu lệnh Insert.

+ Chèn dữ liệu vào tất cả các cột, theo thứ tự trong bảng.

```

IF EXISTS(SELECT TABLE_NAME FROM
INFORMATION_SCHEMA.TABLES
WHERE TABLE_NAME = 'T1')
    DROP TABLE T1
GO
CREATE TABLE T1 ( column_1 int, column_2 varchar(30))
INSERT T1 VALUES (1, 'Row #1')

```

+ Chèn dữ liệu vào các cột không theo thứ tự .

```
IF EXISTS(SELECT TABLE_NAME FROM
INFORMATION_SCHEMA.TABLES
WHERE TABLE_NAME = 'T1')
DROP TABLE T1
GO
CREATE TABLE T1 ( column_1 int, column_2 varchar(30))
INSERT T1 (column_2, column_1) VALUES ('Row #1',1)
```

+ Chèn dữ liệu sử dụng giá trị mặc định.

```
IF EXISTS(SELECT TABLE_NAME FROM
INFORMATION_SCHEMA.TABLES
WHERE TABLE_NAME = 'T1')
DROP TABLE T1
GO
CREATE TABLE T1
( column_1 int identity,
column_2 varchar(30)
CONSTRAINT default_name DEFAULT ('column default'),
column_3 int NULL,
column_4 varchar(40)
)
INSERT INTO T1 (column_4)
VALUES ('Explicit value')
INSERT INTO T1 (column_2,column_4)
VALUES ('Explicit value', 'Explicit value')
INSERT INTO T1 (column_2,column_3,column_4)
VALUES ('Explicit value',-44,'Explicit value')
SELECT *
FROM T1
```

+ Chèn dữ liệu vào bảng có cột dữ liệu IDENTITY.

Ví dụ sau sẽ thể hiện chèn dữ liệu vào bảng có cột dữ liệu IDENTITY, cột có dữ liệu IDENTITY sẽ tự động gán giá trị khi hàng mới được tạo, nên người nhập không nhập và sai. Tuy nhiên với thứ tự đúng câu lệnh SET IDENTITY_INSERT nhập giá trị.

```

IF EXISTS(SELECT TABLE_NAME FROM
INFORMATION_SCHEMA.TABLES
WHERE TABLE_NAME = 'T1')
DROP TABLE T1
GO
CREATE TABLE T1 ( column_1 int IDENTITY, column_2
varchar(30))
INSERT T1 VALUES ('Row #1')
INSERT T1 (column_2) VALUES ('Row #2')
SET IDENTITY_INSERT T1 ON
INSERT INTO T1 (column_1,column_2)
VALUES (-99,'Explicit identity value')
SELECT *
FROM T1

```

Lệnh Insert - Chèn dữ liệu vào bảng bằng cách sử dụng SELECT.

Câu lệnh này cho phép chèn dữ liệu vào bảng bằng cách sử dụng câu lệnh VALUES, nhúng giá trị chèn vào câu lệnh SELECT. Điều này rất hữu ích khi chèn dữ liệu có kiểu Nchar, Nvarchar hoặc Unicode thì khi chèn dữ liệu cần phải thêm tiền tố N, ví dụ Lname=N'John Smith'.

+ *Chèn dữ liệu từ truy vấn các cột trong lệnh SELECT.*

```

USE pubs
INSERT INTO MyBooks
SELECT *
FROM titles
WHERE type = 'mod_cook'

```

+ *Chèn dữ liệu từ truy vấn một số cột.*

```

USE pubs
INSERT INTO MyBooks
SELECT title_id, title, type
FROM titles
WHERE type = 'mod_cook'

```

Lệnh Update – Sửa dữ liệu.

Lệnh Update sử dụng các giá trị trong bảng hoặc View, xem xét cú pháp thông qua các ví dụ sau.

+ *S a d l i u s d n g l ã h Update s d n g m ã h SET.*

```
UPDATE Northwind.dbo.Products
SET UnitPrice = UnitPrice * 1.1
WHERE CategoryID = 2
```

Ho c g n giá tr tr c ti p:

```
UPDATE authors
    SET authors.au_fname = 'Annie'
    WHERE au_fname = 'Anne'
```

Ho c g n giá tr NULL cho m t c t.

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

+ *S a d l i u s d n g m ã h Where xác ã h hàng c s a d l i u.*

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

+ *S a d l i u s d n g m ã h From, s d n g thông tin t m t b ã h khác.*

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

Ho c ví d giá tr xác ã h là t ã h b ã h khác.

```
UPDATE titles
    SET ytd_sales =
    (SELECT SUM(qty)
```

```

        FROM sales
            WHERE sales.title_id = titles.title_id
            AND sales.ord_date IN (SELECT MAX(ord_date)
FROM sales))
    FROM titles, sales

```

+ S a d l i u s d n g m n h Top, xác nh s l n g hàng u tiên c s a d l i u.

```

UPDATE authors
SET state = 'ZZ'
FROM (SELECT TOP 10 * FROM authors ORDER BY au_lname)
AS t1
WHERE authors.au_id = t1.au_id

```

L ãnh WriteText – S a d l i u Text, Image.

L ãnh WriteText c s d n g c p nh t c t có ki u Text ho c Image. D ãnh ki u Text và Image th ãnh có kích th ãnh c l n, có th ãnh n Gyga byte, nên làm vi c v i ki u d ãnh này ph i s d n g con tr . s d n g c l ãnh này tr c h t ãnh ãnh i qu n tr (Administrator) ph i t thu c tính select into/bulk copy là true, th c ãnh ãnh t ãnh sau:

```

USE master
EXEC sp_dboption 'pubs', 'select into/bulkcopy', 'TRUE'

```

V i c t d ãnh ki u Text, Image ta có th ãnh gán giá tr NULL ho c s d n g các l ãnh WriteText, UpdateText gán giá tr , khi s d n g các l ãnh trên, hàng d ãnh ãnh có c t c n chèn ãnh t n t i (khô ãnh ng th i v i câu l ãnh Insert). Riêng i v i c t d ãnh ki u Text b n có th ãnh s d n g l ãnh thêm d ãnh ãnh ãnh ãnh các c t ki u chu i khác ãnh ãnh kích th ãnh c c a d ãnh l i u t i a ch ãnh c 4096 ký t .

+ Th c ãnh ãnh chèn o n v n b n vào c t d ãnh l i u ki u Text.

```

DECLARE @ptrval binary(16)
SELECT @ptrval = TEXTPTR(pr_info)
FROM pub_info pr, publishers p
WHERE p.pub_id = pr.pub_id
    AND p.pub_name = 'New Moon Books'

```

```
WRITETEXT pub_info.pr_info @ptrval 'New Moon Books
(NMB) has just released another top ten publication.
With the latest publication this makes NMB the hottest
new publisher of the year!'
```

GO

Xem ví dụ trên ta thấy, chèn dữ liệu vào cột Text hoặc Image ta phải sử dụng con trỏ kiểu binary hoặc varbinary, con trỏ xác định vào cột text, image và hàng target cũng có trong bảng dữ liệu, sau đó sử dụng lệnh WriteText gán giá trị. Trong thực tế khi thực hiện lệnh này ta thường thực hiện thông qua thủ tục lưu trữ của CSDL, giá trị được gán qua biến. Lệnh WriteText thường được sử dụng khi cột dữ liệu đó là NULL hoặc là toàn bộ dữ liệu đã có (không chèn thêm).

Lệnh UpdateText – Sửa dữ liệu Text, Image.

Lệnh UpdateText có chức năng thay đổi dữ liệu kiểu Text, Image, tuy nhiên UpdateText khác WriteText, UpdateText có thể sửa, xóa dữ liệu theo target hoặc thêm dữ liệu vào phần dữ liệu đã có của cột dữ liệu.

+ *Cú pháp chung:*

```
UPDATETEXT { table_name.dest_column_name dest_text_ptr }
  { NULL | insert_offset }
  { NULL | delete_length }
  [ WITH LOG ]
  [ inserted_data
    | { table_name.src_column_name src_text_ptr } ]
```

Trong đó:

- Insert_offset: Xác định vị trí theo byte dữ liệu sẽ chèn vào hoặc bắt đầu xóa.
- Delete_length: Xác định dài dữ liệu sẽ xóa tính từ vị trí insert_offset.

Vì vậy, xóa, sửa dữ liệu chỉ có thể thực hiện thông qua các tham số insert_offset, delete_offset, ví dụ muốn sửa dữ liệu, ưu tiên phải xác định vị trí bắt đầu của dữ liệu (insert_offset) và dài dữ liệu cần sửa, bắt đầu từ vị trí cần xóa dữ liệu mới sẽ chèn vào.

+ Ví dụ sử dụng lệnh với kiểu Text.

```

USE pubs
GO
EXEC sp_dboption 'pubs', 'select into/bulkcopy', 'true'
GO
DECLARE @ptrval binary(16)
SELECT @ptrval = TEXTPTR(pr_info)
      FROM pub_info pr, publishers p
      WHERE p.pub_id = pr.pub_id
      AND p.pub_name = 'New Moon Books'
UPDATETEXT pub_info.pr_info @ptrval 88 1 'b'
GO
EXEC sp_dboption 'pubs', 'select into/bulkcopy',
'false'
GO

```

Cursor - i u khi n con tr .

Cursor là ki u bi n xác nh con tr cho m t t p d li u, là k t qu c a câu l nh Select. Cursor c k t h p cùng l nh Fetch xác nh v trí hàng trong t p d li u. Cursor có 2 ki u Cursor thông th ng và Scroll Cursor.

Các thao tác th c hi n v i Cursor:

- + Declare: Khai báo.
- + Open: M con tr làm vi c v i t p d li u.
- + Fetch: D ch chuy n v trí hàng trong t p d li u.
- + Close: óng con tr .
- + DeAllocate: Gi i phóng con tr .

+ Ví d s d ng Curcor, li t kê danh sách các hàng c a b ng Authors.

```

USE pubs
GO
DECLARE authors_cursor CURSOR FOR
SELECT au_lname FROM authors
WHERE au_lname LIKE "B%"
ORDER BY au_lname

OPEN authors_cursor

-- Perform the first fetch.
FETCH NEXT FROM authors_cursor

```

```

-- Check @@FETCH_STATUS to see if there are any more
rows to fetch.
WHILE @@FETCH_STATUS = 0
BEGIN
    -- This is executed as long as the previous fetch
succeeds.
    FETCH NEXT FROM authors_cursor
END

```

```

CLOSE authors_cursor
DEALLOCATE authors_cursor
GO

```

+ Ví dụ sử dụng Cursor, giá trị cột c và a vào biến.

```

USE pubs
GO

```

```

-- Declare the variables to store the values returned
by FETCH.

```

```

DECLARE @au_lname varchar(40), @au_fname varchar(20)

```

```

DECLARE authors_cursor CURSOR FOR
SELECT au_lname, au_fname FROM authors
WHERE au_lname LIKE "B%"
ORDER BY au_lname, au_fname

```

```

OPEN authors_cursor

```

```

-- Perform the first fetch and store the values in
variables.
-- Note: The variables are in the same order as the
columns
-- in the SELECT statement.

```

```

FETCH NEXT FROM authors_cursor
INTO @au_lname, @au_fname

```

```

-- Check @@FETCH_STATUS to see if there are any more
rows to fetch.

```

```

WHILE @@FETCH_STATUS = 0
BEGIN

    -- Concatenate and display the current values in the
    variables.
    PRINT "Author: " + @au_fname + " " + @au_lname

    -- This is executed as long as the previous fetch
    succeeds.
    FETCH NEXT FROM authors_cursor
    INTO @au_lname, @au_fname
END

CLOSE authors_cursor
DEALLOCATE authors_cursor
GO

```

+ Ví dụ sử dụng Scroll Cursor, cho phép sử dụng các phương thức: LAST, PRIOR, RELATIVE, ABSOLUTE.

```

USE pubs
GO

-- Execute the SELECT statement alone to show the
-- full result set that is used by the cursor.
SELECT au_lname, au_fname FROM authors
ORDER BY au_lname, au_fname

-- Declare the cursor.
DECLARE authors_cursor SCROLL CURSOR FOR
SELECT au_lname, au_fname FROM authors
ORDER BY au_lname, au_fname

OPEN authors_cursor

-- Fetch the last row in the cursor.
FETCH LAST FROM authors_cursor

-- Fetch the row immediately prior to the current row
in the cursor.
FETCH PRIOR FROM authors_cursor

```

```

-- Fetch the second row in the cursor.
FETCH ABSOLUTE 2 FROM authors_cursor

-- Fetch the row that is three rows after the current
row.
FETCH RELATIVE 3 FROM authors_cursor

-- Fetch the row that is two rows prior to the current
row.
FETCH RELATIVE -2 FROM authors_cursor

CLOSE authors_cursor
DEALLOCATE authors_cursor
GO

```

Lệnh Delete – Xóa dữ liệu.

Sử dụng lệnh Delete xóa dữ liệu, kết hợp cùng mệnh đề WHERE để xóa một hay nhiều hàng dữ liệu trong bảng.

+ *Xóa tất cả các hàng của bảng.*

```

USE pubs
DELETE authors

```

+ *Xóa một tập các hàng.*

```

USE pubs
DELETE FROM authors
WHERE au_lname = 'McBadden'

```

+ *Xóa một hàng từ vị trí con trỏ.*

```

USE pubs
DELETE FROM authors
WHERE CURRENT OF complex_join_cursor

```

Trong ví dụ trên con trỏ đã có tên complex_join_cursor.

+ Xóa các hàng đã xóa và liên kết các bảng.

```
/* SQL-92-Standard subquery */
USE pubs
DELETE FROM titleauthor
WHERE title_id IN
    (SELECT title_id
     FROM titles
     WHERE title LIKE '%computers%')

/* Transact-SQL extension */
USE pubs
DELETE titleauthor
FROM titleauthor INNER JOIN titles
    ON titleauthor.title_id = titles.title_id
WHERE titles.title LIKE '%computers%'
```

+ Xóa dữ liệu dựa trên khóa Top.

```
DELETE authors
FROM (SELECT TOP 10 * FROM authors) AS t1
WHERE authors.au_id = t1.au_id
```

Lệnh Truncate Table – Xóa dữ liệu toàn bộ.

Tương tự như câu lệnh Delete, lệnh Truncate Table sẽ xóa dữ liệu toàn bộ, thao tác này giống như Delete khi không có điều kiện Where như lệnh Truncate Table thì chỉ nhanh hơn.

```
TRUNCATE TABLE authors
```

Lệnh Group – Nhóm lệnh.

Lệnh Group không tham gia thao tác với CSDL, lệnh Group xác định nhóm các lệnh với nhau, nhóm lệnh xác định vị trí ưu tiên hoặc tất cả lệnh Group có số lệnh Group tiếp theo. Khi gặp lệnh Group nhóm lệnh sẽ thực hiện ngay trên SQL Server thì chỉ cần.


```

USE pubs
GO
DECLARE @MyMsg VARCHAR(50)
SELECT @MyMsg = 'Hello, World.'
GO -- @MyMsg is not valid after this GO ends the batch.

-- Yields an error because @MyMsg not declared in this
batch.
PRINT @MyMsg
GO

SELECT @@VERSION;
-- Yields an error: Must be EXEC sp_who if not first
statement in
-- batch.
sp_who
GO

```

Control-of-Flow - i u khi n lu ng.

T ng t nh các ngôn ng l p trình thi t k ng d ng, T-SQL cho phép thi t l p k ch b n câu l nh, cho phép s d ng các l nh i u khi n kh i, lu ng, vòng l p, i u ki n, r nhánh,... Sau ây là b ng các l nh:

T khóa	Mô t
BEGIN...END	Kh i l nh
GOTO	L nh nh y
IF...ELSE	L nh i u ki n
RETURN	Thoát
WAITFOR	Ch th c hi n l nh
WHILE..BREAK..CONTINUE	Vòng l p, thoát kh i vòng l p, quay l i l p
CASE	R nhánh
DECLARE	Khai báo
PRINT	In thông báo
RAISEERROR	Tr l i mã l i
EXECUTE (EXEC)	Th c hi n l nh

TRUY VẤN DỮ LIỆU.

Trong bài trước ta đã xem xét những câu lệnh thao tác với dữ liệu như Insert, Update, Delete, phần này ta sẽ xem xét các câu lệnh khai thác truy vấn dữ liệu như Select, các phép Join,...

Lệnh Use - Chọn Cơ sở dữ liệu.

Sử dụng lệnh Use chọn CSDL trong kịch bản câu lệnh.

```
Use Pubs
```

Select - Truy vấn tất cả các cột trong bảng.

Lệnh Select sử dụng truy vấn dữ liệu để liệt kê tất cả các cột trong bảng, từ khóa WHERE, ORDER BY, GROUP BY, HAVING, JOIN, UNION, và các hàmaggregate.

```
USE Northwind
GO
SELECT *
FROM Shippers
GO
```

Order by - Truy vấn sắp xếp danh sách theo thứ tự.

ASC là sắp xếp tăng, DESC là sắp xếp giảm, khi xác định sắp xếp tăng hoặc giảm có thể không cần từ khóa ASC mà hệ thống sẽ mặc định là ASC.

```
USE Northwind
GO
SELECT *
FROM Shippers
ORDER BY CompanyName DESC
GO
```

Truy vấn một số cột, xác định thứ tự các cột.

```

USE Northwind
GO
SELECT OrderID, ProductID, UnitPrice, Quantity,
Discount
FROM [Order Details]
ORDER BY OrderID ASC
GO

```

i tên các c t khi truy v n.

```

USE Northwind
GO
SELECT OrderID as [Order ID], ProductID as [Product
ID], UnitPrice as [Unit Price], Quantity, Discount
FROM [Order Details]
ORDER BY OrderID ASC
GO

```

L nh Case - Phân l p d li u.

Case là câu l nh r nhánh, th ng c s d ng phân l p d li u trong câu l nh Select.

Ví d s d ng l nh Case n gi n:

```

USE pubs
GO
SELECT Category =
CASE type
WHEN 'popular_comp' THEN 'Popular Computing'
WHEN 'mod_cook' THEN 'Modern Cooking'
WHEN 'business' THEN 'Business'
WHEN 'psychology' THEN 'Psychology'
WHEN 'trad_cook' THEN 'Traditional Cooking'
ELSE 'Not yet categorized'
END,
CAST(title AS varchar(25)) AS 'Shortened Title',
price AS Price
FROM titles
WHERE price IS NOT NULL

```

```
ORDER BY type, price
COMPUTE AVG(price) BY type
GO
```

Ví d s d ng l nh Case tìm ki m:

```
USE pubs
GO
SELECT      'Price Category' =
            CASE
                WHEN price IS NULL THEN 'Not yet priced'
                WHEN price < 10 THEN 'Very Reasonable Title'
                WHEN price >= 10 and price < 20 THEN 'Coffee
Table Title'
                ELSE 'Expensive book!'
            END,
            CAST(title AS varchar(20)) AS 'Shortened Title'
FROM titles
ORDER BY price
GO
```

K t qu th c hi n nh sau:

Price Category	Shortened Title
-----	-----
Not yet priced	Net Etiquette
Not yet priced	The Psychology of Co
Very Reasonable Title	The Gourmet Microwav
Very Reasonable Title	You Can Combat Compu
Very Reasonable Title	Life Without Fear
Very Reasonable Title	Emotional Security:
Coffee Table Title	Is Anger the Enemy?
Coffee Table Title	Cooking with Compute
Coffee Table Title	Fifty Years in Bucki
Coffee Table Title	Sushi, Anyone?
Coffee Table Title	Prolonged Data Depri
Coffee Table Title	Silicon Valley Gastr
Coffee Table Title	Straight Talk About
Coffee Table Title	The Busy Executive's

```
Expensive book!           Secrets of Silicon V
Expensive book!           Onions, Leeks, and G
Expensive book!           Computer Phobic And
Expensive book!           But Is It User Frien
```

(18 row(s) affected)

tên cho cột.

Sử dụng độ phân xác định tên cột, tối đa là 30 ký tự.

```
SELECT 'sum'= SUM(ytd_sales) FROM titles
```

Khi cần thể hiện độ phân định giá trị cho tên cột ta cần sử dụng 2 dấu li n nhau. Ví dụ 'I don't understand.'

Chuỗi ký tự trong kết quả truy vấn.

Sử dụng độ phân định trong chuỗi ký tự.

```
SELECT 'The publisher''s name is', publisher=pub_name
FROM publishers
```

Các giá trị tính toán.

Sử dụng các ký hiệu tính toán để sử dụng các phép toán +, -, *, /, %.

```
SELECT title_id, ytd_sales*2 FROM titles
```

Truy vấn kiểu dữ liệu Text, Image.

Truy vấn kiểu dữ liệu Text, Image có thể sử dụng 2 lệnh Select hoặc ReadText. Khi sử dụng lệnh Select truy vấn kiểu dữ liệu này thì chỉ truy vấn được dữ liệu có độ dài xác định như trong câu lệnh SET TEXTSIZE.

```
SET TEXTSIZE 25
```

```
SELECT pub_id, pr_info FROM pub_info
```

Ng m nh kích th c s d ng cho truy v n là 4096 (4K).

T khóa Distinct – Truy v n các hàng khác nhau theo c t.

truy v n các hàng đ li u khác nhau theo c t ta s d ng t khóa Distinct.

```
USE pubs
SELECT DISTINCT au_id
FROM titleauthor
```

Xác nh b ng trong m nh From.

```
USE pubs
SELECT p.pub_id, p.pub_name
FROM publishers p
```

M nh Where.

M nh Where xác nh i u ki n các hàng c truy v n, bi u th c trong m nh Where xác nh theo bi u th c logic. Các phép toán, câu l nh xác nh g m:

- Các phép toán so sánh: =, <>, <, >, !<, !>.
- T khóa xác nh ph m vi: Between, Not Between.
- Danh sách: In, Not In.
- Theo m u ình d ng: Like, Not Like.
- Giá tr NULL: Is Null, Is Not Null.
- Các phép toán logic: And, Or.

+ *T khóa Between:*

```
SELECT UnitsInStock, ProductID, ProductName
```

```
FROM Northwind.dbo.Products
WHERE UnitsInStock BETWEEN 15 AND 25
ORDER BY UnitsInStock
```

+ *T khóa Not Between.*

```
SELECT UnitsInStock, ProductID, ProductName
FROM Northwind.dbo.Products
WHERE UnitsInStock NOT BETWEEN 15 AND 25
ORDER BY UnitsInStock
```

+ *T khóa In, Not In.*

```
USE pubs
```

```
SELECT au_lname, state
FROM authors
WHERE state IN ('CA', 'IN', 'MD')
```

```
USE pubs
```

```
SELECT au_lname, au_fname
FROM authors
WHERE au_id IN
    (SELECT au_id
     FROM titleauthor
     WHERE royaltyper < 50)
```

```
USE pubs
```

```
SELECT au_lname, au_fname
FROM authors
WHERE au_id NOT IN
    (SELECT au_id
     FROM titleauthor
     WHERE royaltyper < 50)
```

+ *T khóa Like.*

T khóa Like c s d ng t ng t nh phép so sánh, phép Like c th c hi n cho d li u ki u chu i, phép Like c xem là phép so sánh theo nh d ng c a chu i, vì c nh d ng xác nh theo m t s t khóa sau:

- % Xác nh b t k chu i ký t nào ho c không có ký t nào t i v trí.
- _ M t ký t b t k nào ó.
- [] M t ký t nào ó n m trong ph m vi, ví d [a-f].
- [^] Xác nh m t ký t không thu c ph m vi nào ó, ví d [^a-f].

Ví d s d ng t khóa Like v i %:

```
USE pubs
GO
SELECT phone
FROM authors
WHERE phone LIKE '415%'
ORDER by au_lname
GO
```

Ví d t khóa Not Like v i %:

```
USE pubs
GO
SELECT phone
FROM authors
WHERE phone NOT LIKE '415%'
ORDER BY au_lname
GO
```

Ví d s d ng t khóa Like v i m nh Escape: Escape c s d ng lo i b m t ký t ho c chu i kh i phép so sánh.

```
USE pubs
GO
```



```

IF EXISTS(SELECT TABLE_NAME FROM
INFORMATION_SCHEMA.TABLES
WHERE TABLE_NAME = 'mytbl2')
DROP TABLE mytbl2
GO
USE pubs
GO
CREATE TABLE mytbl2
(
c1 sysname
)
GO
INSERT mytbl2 VALUES ('Discount is 10-15% off')
INSERT mytbl2 VALUES ('Discount is .10-.15 off')
GO
SELECT c1
FROM mytbl2
WHERE c1 LIKE '%10-15!% off%' ESCAPE '!'
GO

```

Ví dụ sử dụng toán tử khóa Like với []:

```

USE pubs
GO
SELECT au_lname, au_fname, phone
FROM authors
WHERE au_lname LIKE '[CK]ars[eo]n'
ORDER BY au_lname ASC, au_fname ASC
GO

```

+ *Giá trị NULL.*

Giá trị NULL có thể nhập bằng cách gõ toán tử null hoặc gán theo câu lệnh. tìm giá trị NULL trong bảng sử dụng toán tử khóa Is Null hoặc Is Not Null.

```

SELECT title_id, type, advance
FROM pubs.dbo.titles
WHERE advance IS NULL

```

T O B N G B N G L N H SELECT INTO.

Lệnh Select Into truy vấn dữ liệu, dữ liệu được đưa vào một bảng mới. Nếu thực hiện select into/bulkcopy thì có thể tạo bảng cùng, nếu thực hiện không thì ta có thể tạo bảng mới.

```
SELECT Shippers.*, Link.Address, Link.City,
      Link.Region, Link.PostalCode
INTO NewShippers
FROM Shippers
      JOIN LinkServer.DB.dbo.Shippers AS Link
      ON (Shippers.ShipperID = Link.ShipperID)
```

L N H COMPUTE BY.

Khi thể hiện với các hàm tính toán SUM, AVG, MIN, MAX, COUNT thì nên sử dụng với các mệnh đề GROUP BY, COMPUTE BY (không áp dụng các hàm tính toán với dữ liệu kiểu Text, Image).

+ *Sử dụng Group By*: Từ khóa Group By sử dụng nhóm theo cột, có thể kết hợp các hàm tính toán.

```
USE Northwind
SELECT OrdD.ProductID AS ProdID,
      SUM(OrdD.Quantity) AS AmountSold
FROM [Order Details] AS OrdD JOIN Products as Prd
      ON OrdD.ProductID = Prd.ProductID
      AND Prd.CategoryID = 2
GROUP BY OrdD.ProductID
```

+ *Sử dụng mệnh đề Compute*: Tính toán toàn bộ giá trị.

```
USE pubs
SELECT type, price, advance
FROM titles
ORDER BY type
COMPUTE SUM(price), SUM(advance)
```

+ S d ng m nh Compute By: Tính toán theo nhóm (t ng t Group By).

```
USE pubs
SELECT type, price, advance
FROM titles
ORDER BY type
COMPUTE SUM(price), SUM(advance) BY type
```

TOÁN T UNION.

Toán t Union th c hi n h p 2 t p v i nhau, phép toán này th c hi n ch l y i di n khi có hai hàng c a hai t p trùng nhau.

Gi s có 2 b ng d li u nh sau:

Table1			Table2	
ColumnA	ColumnB		ColumnC	ColumnD
char(4)	int		char(4)	int
-----	---		-----	---
abc	1		ghi	3
def	2		jkl	4
ghi	3		mno	5

Th c hi n toán t Union:

```
SELECT * FROM Table1
UNION
SELECT * FROM Table2
```

K t qu th c hi n:

```
ColumnA  ColumnB
-----  -----
abc      1
def      2
ghi      3
jkl      4
```

Khi sử dụng từ khóa ALL toàn bộ các hàng của hai tệp dữ liệu sẽ được liệt kê, không loại bỏ những hàng trùng nhau.

```
SELECT * FROM TableA
UNION ALL
(
  SELECT * FROM TableB
  UNION
  SELECT * FROM TableC
)
GO
```

Sử dụng toán tử Union với Select Into:

```
USE Northwind
IF EXISTS(SELECT TABLE_NAME FROM
INFORMATION_SCHEMA.TABLES
  WHERE TABLE_NAME = 'CustomerResults')
  DROP TABLE CustomerResults
GO
USE Northwind
SELECT ContactName, CompanyName, City, Phone INTO
CustomerResults
FROM Customers
WHERE Country IN ('USA', 'Canada')
UNION
SELECT ContactName, CompanyName, City, Phone
FROM SouthAmericanCustomers
ORDER BY CompanyName, ContactName ASC
GO
```

TRUY VẤN DỮ LIỆU TỪ NHIỀU BẢNG.

Truy vấn dữ liệu từ nhiều bảng sẽ xác định theo quan hệ giữa các cột của các bảng với nhau. Có thể truy vấn thông qua điều kiện liên kết trong mệnh đề Where hoặc từ khóa Join.

Theo i u ki n liên k t.

S d ng i u ki n liên k t theo c t gi a các b ng, thông tin c n truy v n c t nhi u b ng khác nhau, truy v n c các thông tin nh trên ph i xác nh i u ki n liên k t gi a các b ng.

+ *Liên k t b ng nhau.*

```
SELECT P.ProductID,  
       S.SupplierID,  
       S.CompanyName  
FROM Suppliers AS S, Products AS P  
WHERE S.SupplierID = P.SupplierID  
      AND P.UnitPrice > $10  
      AND S.CompanyName LIKE N'F%'
```

i v i câu l nh truy v n theo i u ki n liên k t nói trên, các hàng ch a giá tr Null c a c t tham gia liên k t s không c li t kê, câu l nh này t ng ng v i l nh Inner Join (s xem trong ph n sau).

+ *Liên k t không b ng nhau.*

Liên k t d ng này s d ng các phép toán so sánh >, >=, <, <=, <>, !>, !<

```
USE pubs  
SELECT p.pub_name, p.state, a.au_lname, a.au_fname,  
       a.state  
FROM publishers p, authors a  
      WHERE a.state > p.state and  
p.pub_name = 'New Moon Books'  
ORDER BY au_lname ASC, au_fname ASC
```

+ *T liên k t b ng nhau.*

T liên k t trong m t b ng, câu l nh d ng này th ng c s d ng trong v i c xác nh nh ng c p giá tr nào các c t trong b ng có quan h v i nhau theo liên k t.

```

USE pubs
SELECT au1.au_fname, au1.au_lname, au2.au_fname,
au2.au_lname
FROM authors au1, authors au2
    WHERE au1.zip = au2.zip and au1.city = 'Oakland'
ORDER BY au1.au_fname ASC, au1.au_lname ASC

```

+ *T liên kết không bằng nhau.*

```

USE pubs
SELECT au1.au_fname, au1.au_lname, au2.au_fname,
au2.au_lname
FROM authors au1, authors au2
WHERE au1.zip = au2.zip
    AND au1.city = 'Oakland'
    AND au1.state = 'CA'
    AND au1.au_id < au2.au_id
ORDER BY au1.au_lname ASC, au1.au_fname ASC

```

+ *Truy vấn để liệt kê những nhà xuất bản dựa trên.*

Truy vấn dạng này thể hiện mối liên kết theo từng cặp các bảng với nhau.

```

USE pubs
SELECT a.au_lname, a.au_fname, t.title
FROM authors a, titleauthor ta, titles t
    WHERE a.au_id = ta.au_id
    AND ta.title_id = t.title_id
    AND t.type = 'trad_cook'
ORDER BY t.title ASC

```

+ *Liên kết ngoài trái.*

Những mối liên kết nói trên, những hàng có cột là Null sẽ không có giá trị, câu lệnh liên kết ngoài sẽ trả ra những hàng có giá trị Null nói trên. Xác định liên kết ngoài bằng toán tử *.

```

USE pubs
SELECT a.au_fname, a.au_lname, p.pub_name

```

```

FROM authors a, publishers p
WHERE a.city != p.city
ORDER BY p.pub_name ASC, a.au_lname ASC, a.au_fname ASC

```

Bảng authors sẽ ra các nhà hàng có các thành phố là Null, khi đó chúng ta có pub_name, kết quả như sau:

au_fname	au_lname	pub_name
Reginald	Blotchet-Halls	NULL
Michel	DeFrance	NULL
Innes	del Castillo	NULL
Ann	Dull	NULL
Marjorie	Green	NULL
Morningstar	Greene	NULL
Burt	Gringlesby	NULL
Sheryl	Hunter	NULL
Livia	Karsen	NULL
Charlene	Locksley	NULL
Stearns	MacFeather	NULL
Heather	McBadden	NULL
Michael	O'Leary	NULL
Sylvia	Panteley	NULL
Albert	Ringer	NULL
Anne	Ringer	NULL
Meander	Smith	NULL
Dean	Straight	NULL
Dirk	Stringer	NULL
Johnson	White	NULL
Akiko	Yokomoto	NULL
Abraham	Bennet	Algodata Infosystems
Cheryl	Carson	Algodata Infosystems

(23 row(s) affected)

+ Liên kết ngoài phụ.

```

USE pubs
SELECT a.au_fname, a.au_lname, p.pub_name
FROM authors AS a, publishers AS p
WHERE a.city = p.city
ORDER BY p.pub_name ASC, a.au_lname ASC, a.au_fname ASC

```

Kết quả thực hiện:

au_fname	au_lname	pub_name
Abraham	Bennet	Algodata
Infosystems		
Cheryl	Carson	Algodata
Infosystems		
NULL	NULL	Binnet & Hardley
NULL	NULL	Five Lakes
Publishing		
NULL	NULL	GGG&G
NULL	NULL	Lucerne Publishing
NULL	NULL	New Moon Books
NULL	NULL	Ramona Publishers
NULL	NULL	Scotney Books

(9 row(s) affected)

Liên kết Join – Truy vấn nhiều bảng.

Phần trên đã xem xét kỹ thuật truy vấn dữ liệu nhiều bảng sử dụng liên kết, cũng như các phép toán so sánh, *=, =* SQL Server cung cấp câu lệnh Join thay thế các phép toán nói trên.

+ *Inner Join* – Liên kết trong.

Thay vì xác định liên kết trong mệnh đề Where thì đây ta chỉ cần xác định liên kết trong mệnh đề From.

Liên kết bảng:

```
USE pubs
SELECT *
FROM authors AS a INNER JOIN publishers AS p
ON a.city = p.city
ORDER BY a.au_lname DESC
```

Liên kết không bảng:

```
USE pubs
```



```

SELECT p.pub_name, p.state, a.au_lname, a.au_fname,
a.state
FROM publishers p INNER JOIN authors a
    ON a.state > p.state
WHERE p.pub_name = 'New Moon Books'
ORDER BY au_lname ASC, au_fname ASC

```

+ *T liên kết trong bảng.*

T liên kết bảng:

```

USE pubs
SELECT au1.au_fname, au1.au_lname, au2.au_fname,
au2.au_lname
FROM authors au1 INNER JOIN authors au2
    ON au1.zip = au2.zip
WHERE au1.city = 'Oakland'
ORDER BY au1.au_fname ASC, au1.au_lname ASC

```

T liên kết không bảng:

```

USE pubs
SELECT au1.au_fname, au1.au_lname, au2.au_fname,
au2.au_lname
FROM authors au1 INNER JOIN authors au2
    ON au1.zip = au2.zip
WHERE au1.city = 'Oakland'
    AND au1.state = 'CA'
    AND au1.au_id < au2.au_id
ORDER BY au1.au_lname ASC, au1.au_fname ASC

```

+ *Liên kết nhiều bảng.*

```

USE pubs
SELECT a.au_lname, a.au_fname, t.title
FROM authors a INNER JOIN titleauthor ta

```

```

    ON a.au_id = ta.au_id JOIN titles t
    ON ta.title_id = t.title_id
WHERE t.type = 'trad_cook'
ORDER BY t.title ASC

```

+ Liên kết ngoài trái - *LEFT OUTER JOIN*.

Liên kết ngoài trái là phép toán *=.

```

USE pubs
SELECT a.au_fname, a.au_lname, p.pub_name
FROM authors a LEFT OUTER JOIN publishers p
    ON a.city = p.city
ORDER BY p.pub_name ASC, a.au_lname ASC, a.au_fname ASC

```

+ Liên kết ngoài phải - *RIGHT OUTER JOIN*.

Liên kết ngoài phải là phép toán =*.

```

USE pubs
SELECT a.au_fname, a.au_lname, p.pub_name
FROM authors a RIGHT OUTER JOIN publishers p
    ON a.city = p.city
ORDER BY p.pub_name ASC, a.au_lname ASC, a.au_fname ASC

```

+ Liên kết ngoài 2 phía - *FULL OUTER JOIN*.

Là phép liên kết trái hoặc phải.

```

USE pubs
SELECT a.au_fname, a.au_lname, p.pub_name
FROM authors a FULL OUTER JOIN publishers p
    ON a.city = p.city
ORDER BY p.pub_name ASC, a.au_lname ASC, a.au_fname ASC

```

Kết quả như sau:

au_fname	au_lname	pub_name
Reginald	Blotchet-Halls	NULL
Michel	DeFrance	NULL
Innes	del Castillo	NULL
Ann	Dull	NULL
Marjorie	Green	NULL
Morningstar	Greene	NULL
Burt	Gringlesby	NULL
Sheryl	Hunter	NULL
Livia	Karsen	NULL
Charlene	Locksley	NULL
Stearns	MacFeather	NULL
Heather	McBadden	NULL
Michael	O'Leary	NULL
Sylvia	Panteley	NULL
Albert	Ringer	NULL
Anne	Ringer	NULL
Meander	Smith	NULL
Dean	Straight	NULL
Dirk	Stringer	NULL
Johnson	White	NULL
Akiko	Yokomoto	NULL
Abraham	Bennet	Algodata Infosystems
Cheryl	Carson	Algodata Infosystems
NULL	NULL	Binnet & Hardley
NULL	NULL	Five Lakes Publishing
NULL	NULL	GGG&G
NULL	NULL	Lucerne Publishing
NULL	NULL	New Moon Books
NULL	NULL	Ramona Publishers
NULL	NULL	Scotney Books

(30 row(s) affected)

+ *Giá trị Null và phép Join.*

Giá trị Null không xác định trong phép so sánh của mệnh đề Where (chỉ sử dụng với các phép so sánh Is Null hoặc Is Not Null), trong phép Join ta có thể xác định như nhau giữa 2 giá trị Null. Xét ví dụ sau:

Giả sử có 2 bảng dữ liệu có giá trị như sau:

table1		table2	
a	b	c	d
-----	-----	-----	-----
1	one	NULL	two
NULL	three	4	four
4	join4		

Th c hi n phép Join nh sau:

```
SELECT *
FROM table1 t1 JOIN table2 t2
ON t1.a = t2.c
ORDER BY t1.a
```

K t qu th c hi n:

a	b	c	d
-----	-----	-----	-----
4	join4	4	four

(1 row(s) affected)

TRUY V NT NGH P.

Vi c s d ng các hàm tính toán nh SUM, AVG,... th ng c th c hi n theo các m nh WHERE, GROUP BY, HAVING. Khi xác nh i u ki n có s d ng các hàm tính toán thì ph i s d ng m nh HAVING mà không c s d ng trong m nh WHERE.

Các hàm tính toán có th tóm t t nh sau:

SUM([ALL DISTINCT])	Tính t ng t t c ho c nh ng hàng khác nhau.
AVG([ALL DISTINCT])	Tính trung bình t t c ho c nh ng hàng khác nhau.
COUNT([ALL DISTINCT])	m s hàng t t c ho c nh ng hàng khác nhau.
COUNT(*)	m các hàng c l a ch n.
MAX()	Tính giá tr l n nh t.

MIN()

Tính giá trị nhỏ nhất.

Các hàm SUM, AVG chỉ làm việc với dữ liệu số, các hàm SUM, AVG, COUNT, MAX, MIN bỏ qua giá trị Null, hàm COUNT(*) đếm các hàng có giá trị Null.

Sử dụng hàm tính toán.

+ Tính tổng toàn bộ.

```
USE pubs
SELECT SUM(ytd_sales)
FROM titles
```

+ Tính tổng, trung bình có điều kiện.

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

Mệnh Group By.

Group by chỉ thể hiện nhóm các hàng theo giá trị cụ thể xác định, các hàm tính toán sẽ chỉ thể hiện theo nhóm nói trên.

```
USE Northwind
SELECT OrdD.ProductID AS ProdID,
       SUM(OrdD.Quantity) AS AmountSold
FROM [Order Details] AS OrdD JOIN Products as Prd
     ON OrdD.ProductID = Prd.ProductID
     AND Prd.CategoryID = 2
GROUP BY OrdD.ProductID
```

Kết quả thể hiện như sau:

ProdID	AmountSold
3	328
4	453
5	298
6	301
8	372
15	122
44	601
61	603
63	445
65	745
66	239
77	791

(12 row(s) affected)

M nh Having.

Having c s d ng cùng v i các hàm tính toán xác nh i u ki n l c các hàng, th ng c k t h p cùng m nh Group By th c hi n các hàm tính toán theo nhóm.

+ *Having v i hàm SUM.*

```
USE pubs
SELECT pub_id, total = SUM(ytd_sales)
FROM titles
GROUP BY pub_id
HAVING SUM(ytd_sales) > 40000
```

+ *Having v i hàm Count.*

```
USE pubs
SELECT pub_id, total = SUM(ytd_sales)
FROM titles
GROUP BY pub_id
HAVING COUNT(*) > 5
```

+ *Having v i m nh Where.*

```
SELECT pub_id, SUM(advance) AS AmountAdvanced,  
       AVG(price) AS AveragePrice  
FROM pubs.dbo.titles  
WHERE pub_id > '0800'  
      AND price >= $5  
GROUP BY pub_id  
HAVING SUM(advance) > $15000  
      AND AVG(price) < $20  
ORDER BY pub_id DESC
```

+ *Having thay cho m nh Where.*

```
SELECT titles.pub_id, AVG(titles.price)  
FROM titles INNER JOIN publishers  
      ON titles.pub_id = publishers.pub_id  
GROUP BY titles.pub_id  
HAVING publishers.state = 'CA'
```

TRUY V N L NG NHAU.

Phần này sẽ xem xét các câu lệnh truy vấn liên quan nhau, trong câu lệnh truy vấn Select có câu lệnh truy vấn Select khác trong điều kiện xác định câu lệnh Select ngoài. Thông thường các câu lệnh dùng này đi cùng các toán tử IN, NOT IN, EXITST, NOT EXIST, ANY, ALL.

Truy vấn liên quan với phép boolean.

```
USE pubs  
SELECT title, price  
FROM titles  
WHERE price =  
      (SELECT price  
      FROM titles  
      WHERE title = 'Straight Talk About Computers')
```

u tiên câu lệnh xác định hàng trong lệnh Select trong, lệnh truy vấn này phải
a ra kết quả duy nhất.

Truy vấn với khóa IN.

Kiểm tra tên trong tập các giá trị truy vấn.

```
USE pubs
SELECT distinct pub_name
FROM publishers
WHERE pub_id IN
  (SELECT pub_id
   FROM titles
   WHERE type = 'business')
```

Hàng số tên trong kho:

```
USE pubs
SELECT DISTINCT au_lname, au_fname
FROM authors
WHERE 100 IN
  (SELECT royaltyper
   FROM titleauthor
   WHERE titleauthor.au_id = authors.au_id)
```

Truy vấn với khóa Exist.

Kiểm tra tên từ hàng dữ liệu truy vấn.

```
USE pubs
SELECT DISTINCT pub_name
FROM publishers
WHERE EXISTS
  (SELECT *
   FROM titles
   WHERE pub_id = publishers.pub_id
   AND type = 'business')
```


Truy vấn với hàm All.

Kiểm tra với tất cả các hàng.

```
USE pubs
SELECT t1.type
FROM titles t1
GROUP BY t1.type
HAVING MAX(t1.advance) >= ALL
    (SELECT 2 * AVG(t2.advance)
     FROM titles t2
     WHERE t1.type = t2.type)
```

Truy vấn với hàm Any.

Kiểm tra tồn tại bất kỳ hàng nào.

```
USE pubs
SELECT title
FROM titles
WHERE advance > ANY
    (SELECT advance
     FROM publishers INNER JOIN titles
     ON titles.pub_id = publishers.pub_id
     AND pub_name = 'Algodata Infosystems')
```

Truy vấn với hàm Some.

Kiểm tra với ít nhất một hàng.

```
USE pubs
SELECT t1.type
FROM titles t1
GROUP BY t1.type
HAVING MAX(t1.advance) >= SOME
    (SELECT 2 * AVG(t2.advance)
     FROM titles t2
     WHERE t1.type = t2.type)
```

Nhiệm vụ Select lồng nhau.

```
USE pubs
SELECT au_lname, au_fname
FROM authors
WHERE au_id IN
    (SELECT au_id
     FROM titleauthor
     WHERE title_id IN
         (SELECT title_id
          FROM titles
          WHERE type = 'popular_comp'))
```

UPDATE, DELETE, INSERT VÀ LẬP NHỮNG TRUY VẤN LỒNG NHAU.

Vì có thể chỉ ra các lập luận thao tác và dữ liệu có thể kết hợp lại để truy vấn lồng nhau xác định phạm vi dữ liệu để thao tác.

Kết quả ví dụ Select.

```
UPDATE titles
SET price = price * 2
WHERE pub_id IN
    (SELECT pub_id
     FROM publishers
     WHERE pub_name = 'New Moon Books')
```

Kết quả ví dụ Join.

```
UPDATE titles
SET price = price * 2
FROM titles INNER JOIN publishers ON titles.pub_id =
publishers.pub_id
AND pub_name = 'New Moon Books'
```

Xóa dữ liệu kết hợp với Select.

```
DELETE sales
WHERE title_id IN
  (SELECT title_id
   FROM titles
   WHERE type = 'business')
```

Xóa dữ liệu với phép Join.

```
DELETE sales
FROM sales INNER JOIN titles ON sales.title_id =
titles.title_id
  AND type = 'business'
```

L NHIỆM READTEXT – CTEXT, IMAGE.

Lệnh ReadText cho phép chỉ định dữ liệu kiểu Text, Image và chuyển vào một bit.

```
USE pubs
GO
DECLARE @ptrval varbinary(16)
SELECT @ptrval = TEXTPTR(pr_info)
  FROM pub_info pr INNER JOIN publishers p
  ON pr.pub_id = p.pub_id
  AND p.pub_name = 'New Moon Books'
READTEXT pub_info.pr_info @ptrval 1 25
GO
```

Ví dụ trên chỉ định dữ liệu từ cột pr_info bắt đầu ở vị trí 1, dài 25 byte.

THAO TÁC DỮ LIỆU NGOÀI.

Nội dung phần này sẽ giới thiệu câu lệnh, kỹ thuật truy vấn dữ liệu của hệ quản trị CSDL khác hoặc Instance khác.

Lệnh OpenRowSet.

Lệnh OpenRowSet sử dụng truy vấn phân phối xa và ngôn ngữ truy vấn là OLE DB, kết nối này có thể thực hiện các lệnh Insert, Update, Delete, Select và insert bảng phân phối. Quy định thực hiện trong câu lệnh thực hiện theo user kết nối trong câu lệnh.

+ *OPENROWSET* với lệnh *SELECT* và *Microsoft OLE DB Provider for SQL Server*.

```
USE pubs
GO
SELECT a.*
FROM OPENROWSET('SQLOLEDB','seattle1';'sa';'MyPass',
    'SELECT * FROM pubs.dbo.authors ORDER BY au_lname,
    au_fname') AS a
GO
```

Ví dụ trên thực hiện kết nối Instance có tên seattle1, user có tên sa, mật khẩu MyPass.

+ *OPENROWSET* với *OLE DB Provider for ODBC*.

```
USE pubs
GO
SELECT a.*
FROM OPENROWSET('MSDASQL',
    'DRIVER={SQL
Server};SERVER=seattle1;UID=sa;PWD=MyPass',
    pubs.dbo.authors) AS a
ORDER BY a.au_lname, a.au_fname
GO
```

+ *Microsoft OLE DB Provider for Jet*. Lệnh dùng này thực hiện kết nối trong Access.

```
USE pubs
GO
SELECT a.*
FROM OPENROWSET('Microsoft.Jet.OLEDB.4.0',
```

```
'c:\MSOffice\Access\Samples\northwind.mdb';'admin';'my
wd', Orders)
    AS a
GO
```

+ *OPENROWSET* v i *INNER JOIN* m t b ng khác.

```
USE pubs
GO
SELECT c.*, o.*
FROM Northwind.dbo.Customers AS c INNER JOIN
    OPENROWSET('Microsoft.Jet.OLEDB.4.0',
```

```
'c:\MSOffice\Access\Samples\northwind.mdb';'admin';'my
wd', Orders)
    AS o
    ON c.CustomerID = o.CustomerID
GO
```

L nh OpenDataSource.

L nh OpenDataSource th c hi n m d li u ngoài Instance, không c n n linked_server.

+ *K t n i n Instance khác.*

```
SELECT *
FROM OPENDATASOURCE(
    'SQLOLEDB',
    'Data Source=ServerName;User
ID=MyUID;Password=MyPass'
    ).Northwind.dbo.Categories
```

+ *K t n i n Excel.*

```

SELECT *
FROM OpenDataSource( 'Microsoft.Jet.OLEDB.4.0',
  'Data Source="c:\Finance\account.xls";User
  ID=Admin;Password=;Extended properties=Excel
  5.0')...xactions

```

L nh OpenQuery.

L nh OpenQuery th c hi n thao tác v i d li u ngoài thông qua LinkedServer.

```

EXEC sp_addlinkedserver 'OracleSvr',
  'Oracle 7.3',
  'MSDAORA',
  'ORCLDB'
GO
SELECT *
FROM OPENQUERY(OracleSvr, 'SELECT name, id FROM
joe.titles')
GO

```

M T S HÀM C B N.

Hàm h th ng.

DB_ID	Tr v ID c a CSDL khi bi t tên.
DB_NAME	Tr v tên CSDL khi bi t ID.
HOST_ID	Tr v ID c a máy ch .
HOST_NAME	Tr v tên máy ch
SUSER_ID	Tr v ID User c a Server khi bi t tên
SUSER_NAME	Tr v tên User c a Server khi bi t ID.
USER_ID	Tr v ID User khi bi t tên
USER_NAME	Tr v tên User khi bi t ID

Hàm thao tác v i chu i.

+ *SUBSTRING* - L y chu i nh trong chu i.

SUBSTRING (*expression* , *start* , *length*)

S d ng v i chu i ký t :

```
USE pubs
SELECT au_lname, SUBSTRING(au_fname, 1, 1)
FROM authors
ORDER BY au_lname
```

S d ng v i text, ntext, image:

```
USE pubs
SELECT pub_id, SUBSTRING(logo, 1, 10) AS logo,
      SUBSTRING(pr_info, 1, 10) AS pr_info
FROM pub_info
WHERE pub_id = '1756'
```

+ *CHARINDEX* – Tr v v trí b t u m t m u trong chu i.

CHARINDEX (*expression1* , *expression2* [, *start_location*]) – Tìm v trí xu t hi n chu i *expression1* trong *expression2*.

Ví d tìm chu i 'wonderful' trong c t notes c a b ng titles:

```
USE pubs
GO
SELECT CHARINDEX('wonderful', notes)
FROM titles
WHERE title_id = 'TC3218'
GO
```

+ *PATINDEX* – Tr v v trí xu t hi n c a m u trong chu i.

PATINDEX ('%pattern%' , *expression*)

Ví d tr m v trí xu t hi n m u '%wonderful%':

```
USE pubs
GO
SELECT PATINDEX('%wonderful%', notes)
FROM titles
WHERE title_id = 'TC3218'
GO
```

Ví d tr m v trí xu t hi n m u '%won_erful%':

```
USE pubs
GO
SELECT PATINDEX('%won_erful%', notes)
FROM titles
WHERE title_id = 'TC3218'
GO
```

+ *STR* – Chuy n d li u ki u s s áng chu i.

STR (*float_expression* [, *length* [, *decimal*]])

Ví d chuy n s sang chu i có dài 6, làm tròn sau d u ph y 1 s .

```
SELECT STR(123.45, 6, 1)
GO
```

K t qu là chu i '123.5'

Ví d s d ng v i hàm Floor l y giá tr nguyên nh h n c a m t s th c:


```
SELECT STR (FLOOR (123.45), 8, 3)
GO
```

Kết quả là '123.000'

+ *STUFF* – Chèn một chuỗi vào một chuỗi khác.

Hàm Stuff thực hiện xóa chuỗi nhúng trong một chuỗi sau đó thực hiện chèn một chuỗi mới vào vị trí bắt đầu.

```
STUFF ( character_expression , start , length , character_expression )
```

Ví dụ :

```
SELECT STUFF('abcdef', 2, 3, 'ijklmn')
GO
```

Kết quả thực hiện:

aijklmnef

+ *SOUNDEX* – Trả về hàm phát âm.

Hàm SOUNDEX sử dụng so sánh phát âm giữa 2 chuỗi, ví dụ sau sẽ cho 2 mã SOUNDEX như nhau:

```
SELECT SOUNDEX ('Smith'), SOUNDEX ('Smythe')
```

Kết quả thực hiện:

S530 S530

+ *Defference* – So sánh giá trị hàm SOUNDEX giữa 2 chuỗi: Giá trị trả về từ 0 đến 4, 4 là giá trị gần nhau nhất. Ví dụ sau so sánh giữa 2 chuỗi:

```
SELECT DIFFERENCE('Smithers', 'Smythers')
GO
```

Kết quả thực hiện: 4

```
SELECT DIFFERENCE('Anothers', 'Brothers')
GO
```

Kết quả thực hiện: 2

+ UNICODE – L y mã unicode ký t u tiên trong chu i.

+ NCHAR – Chuy n mã unicode thành ký t .

Các hàm DateTime.

+ GETDATE: Tr v ngày, gi hi n t i.

+ DATEPART: Tr v giá tr ngay ho c tháng ho c n m c a m t bi u th c ngày.

DATEPART (*datepart* , *date*)

Giá tr datepart theo b ng sau:

Datepart	D ñng rút g n
year	yy, yyyy
quarter	qq, q
month	mm, m
dayofyear	dy, y
day	dd, d
week	wk, ww
weekday	dw
hour	hh
minute	mi, n
second	ss, s
millisecond	ms

```
SELECT DATEPART(m, 0), DATEPART(d, 0), DATEPART(yy, 0)
```

+ SET DATFIRST: t ngày u tiên trong tu n.

+ SET DATEFORMAT: t ñh d ñng ki u DateTime ñh p d li u.

```

SET DATEFORMAT mdy
GO
DECLARE @datevar datetime
SET @datevar = '12/31/98'
SELECT @datevar
GO

```

+ DAY, MONTH, YEAR: Lấy giá trị ngày, tháng, năm.

+ ISDATE: Kiểm tra xem dữ liệu có hợp lệ DateTime không.

+ DATEDIFF: Xác định chênh lệch giữa 2 giá trị DateTime.

```
DATEDIFF ( datepart , startdate , enddate )
```

Ví dụ: Xác định số ngày đã phát hành sách.

```

USE pubs
GO
SELECT DATEDIFF(day, pubdate, getdate()) AS no_of_days
FROM titles
GO

```

+ DATEADD – Xác định giá trị DateTime mới khi thay đổi một khoảng thời gian.

```
DATEADD ( datepart , number , date )
```

```

USE pubs
GO
SELECT DATEADD(day, 21, pubdate) AS timeframe
FROM titles
GO

```

Các hàm chuyển đổi.

+ CONVERT

+ CAST

TRANSACTION – PHIÊN GIAO DỊCH.

Transaction là một đơn vị công việc trong nó bao gồm nhiều việc nhỏ, các việc này chỉ thực hiện thành công thì Transaction thành công, nếu không thì trong quá trình thực hiện của Transaction sẽ bị hủy bỏ. Nếu trong quá trình có phát sinh lỗi thì Transaction sẽ rollback (Roll Back hoặc Cancel), để lại không có kết quả. Một phiên giao dịch có 4 tính chất ACID (Atomicity, Consistency, Isolation, Durability).

Atomicity – Nguyên tử: Một phiên giao dịch là một đơn vị công việc nguyên tử, tất cả đều diễn ra thành công hoặc không thực hiện.

Consistency- Nhất quán: Giao dịch sẽ không thể hiện nếu có một thao tác xung đột với một logic hoặc quan hệ. Tính nhất quán rất quan trọng vì mô hình ứng dụng client/server, vì mô hình ứng dụng này thì mỗi máy có thể có nhiều giao dịch thực hiện đồng thời, nếu một giao dịch nào đó không nhất quán thì tất cả các giao dịch khác sẽ thực hiện sai, dẫn đến sai lệch vì phạm toàn vẹn dữ liệu.

Isolation – Tách biệt: Mỗi một phiên giao dịch đồng thời, các phiên giao dịch khác nhau sẽ không ảnh hưởng đến nhau khi diễn ra cùng một lúc (kết thúc phiên). Giữa có 2 phiên giao dịch có tác động

Durability - Bền vững: Sau khi giao dịch hoàn tất, dữ liệu trở về trạng thái bền vững.

Một phiên giao dịch sẽ xác định bởi, kết thúc:

Bắt đầu phiên giao dịch.

Phiên giao dịch có 3 loại: explicit transaction, implicit transaction, autocommit transaction.

Explicit transaction: Là kiểu phiên giao dịch rõ ràng, bắt đầu bằng lệnh BEGIN TRANSACTION, kết thúc bằng lệnh COMMIT hoặc ROLLBACK.

BEGIN DISTRIBUTED TRANSACTION

[*transaction_name* | @*tran_name_variable*]

tên giao dịch:

```

DECLARE @TranName VARCHAR(20)
SELECT @TranName = 'MyTransaction'

BEGIN TRANSACTION @TranName
GO
USE pubs
GO
UPDATE roysched
SET royalty = royalty * 1.10
WHERE title_id LIKE 'Pc%'
GO

COMMIT TRANSACTION MyTransaction
GO

```

ảnh d u trong giao d ch:

```

BEGIN TRANSACTION RoyaltyUpdate
    WITH MARK 'Update royalty values'
GO
USE pubs
GO
UPDATE roysched
    SET royalty = royalty * 1.10
    WHERE title_id LIKE 'Pc%'
GO
COMMIT TRANSACTION RoyaltyUpdate
GO

```

Autocommit transaction: Mọi câu lệnh thực hiện tự động khi nó kết thúc, không cần câu lệnh commit khi kết thúc phiên giao dịch.

Implicit transaction: Là phiên giao dịch, thực hiện này thông qua hàm API hoặc lệnh SET IMPLICIT_TRANSACTIONS ON. Khi phiên giao dịch kết thúc, câu lệnh T-SQL tiếp theo sẽ khởi tạo phiên giao dịch mới.

```
SET IMPLICIT_TRANSACTIONS { ON | OFF }
```

Setting the value of Implicit transaction:

```
USE pubs
GO
```

```
CREATE table t1 (a int)
GO
INSERT INTO t1 VALUES (1)
GO
```

```
PRINT 'Use explicit transaction'
BEGIN TRAN
INSERT INTO t1 VALUES (2)
SELECT 'Tran count in transaction' = @@TRANCOUNT
COMMIT TRAN
SELECT 'Tran count outside transaction' = @@TRANCOUNT
GO
```

```
PRINT 'Setting IMPLICIT_TRANSACTIONS ON'
GO
SET IMPLICIT_TRANSACTIONS ON
GO
```

```
PRINT 'Use implicit transactions'
GO
-- No BEGIN TRAN needed here.
INSERT INTO t1 VALUES (4)
SELECT 'Tran count in transaction' = @@TRANCOUNT
COMMIT TRAN
SELECT 'Tran count outside transaction' = @@TRANCOUNT
GO
```

```
PRINT 'Use explicit transactions with
IMPLICIT_TRANSACTIONS ON'
GO
BEGIN TRAN
INSERT INTO t1 VALUES (5)
SELECT 'Tran count in transaction' = @@TRANCOUNT
COMMIT TRAN
SELECT 'Tran count outside transaction' = @@TRANCOUNT
GO
```

```
SELECT * FROM t1
GO
```

```
-- Need to commit this tran too!
DROP TABLE t1
COMMIT TRAN
GO
```

K t thúc phiên giao d ch.

S d ng l nh Commit trong phiên giao d ch.

```
COMMIT [ TRAN [ SACTION ] [ transaction_name | @tran_name_variable ] ]
```

+ *Commit m t phiên giao d ch.*

```
BEGIN TRANSACTION
USE pubs
GO
UPDATE titles
SET advance = advance * 1.25
WHERE ytd_sales > 8000
GO
COMMIT
GO
```

+ *Commit nhi u phiên giao d ch l ng nhau.*

```
CREATE TABLE TestTran (Cola INT PRIMARY KEY, Colb CHAR(3))
GO
BEGIN TRANSACTION OuterTran -- @@TRANCOUNT set to 1.
GO
INSERT INTO TestTran VALUES (1, 'aaa')
GO
BEGIN TRANSACTION Inner1 -- @@TRANCOUNT set to 2.
GO
INSERT INTO TestTran VALUES (2, 'bbb')
GO
BEGIN TRANSACTION Inner2 -- @@TRANCOUNT set to 3.
GO
```

```

INSERT INTO TestTran VALUES (3, 'ccc')
GO
COMMIT TRANSACTION Inner2 -- Decrements @@TRANCOUNT to 2.
-- Nothing committed.
GO
COMMIT TRANSACTION Inner1 -- Decrements @@TRANCOUNT to 1.
-- Nothing committed.
GO
COMMIT TRANSACTION OuterTran -- Decrements @@TRANCOUNT to 0.
-- Commits outer transaction OuterTran.
GO

```

Hủy bỏ và quay lại phiên giao dịch.

Sử dụng lệnh RollBack Transaction hủy bỏ những thay đổi hiển và quay lại phiên giao dịch.

```

ROLLBACK [ TRAN [ SACTION ]
    [ transaction_name | @tran_name_variable
    | savepoint_name | @savepoint_variable ] ]

```

LOCK – KHÓA.

Khi 2 hay nhiều người cùng truy nhập cùng thời điểm CSDL, SQL Server sử dụng khóa xác định hoạt động cho một người và không xác định cho người khác. Khóa là việc ngăn không cho những người khác đi vào mà không bằng người khác sử dụng.

Hệ thống SQL Server sử dụng khóa để ngăn, bản có thể thi hành tất cả các cách có hiệu quả hơn bằng việc tìm hiểu về khóa và chọn khóa cho ứng dụng của bạn.

Tìm hiểu về khóa.

Khoá gồm các loại sau:

Kiểu khóa	Mô tả
Shared	Là khóa không làm thay đổi, ghi dữ liệu, dùng cho lệnh Select
Update	Khoá cho phép sửa đổi dữ liệu
Exclusive	Khoá về các thao tác Update, Insert, Delete

Một số phạm vi khóa như sau:

Tên	Mô t
Page	Trang d li u 2K ho c trang ch m c Index, th ng c dùng
Extent	Nhóm các trang có kích th c 8k, ch dùng v i tr ng h p xác nh
Table	C b ng d li u, g m d li u và index
Intent	Là ki u c bi t t ki u khoá c a trang hi n t i trên b ng

B ng xác nh hi u l c c a các ki u khoá

	Shared	Update	Exclusive
Shared	Yes	Yes	No
Update	Yes	No	No
Exclusive	No	No	No

Ví d : Khi t ch khoá là Exclusive thì nh ng phiên giao d ch khác không th yêu c u b t c lo i khoá nào n khi hoá Exclusive b b .

Xem thông tin v khoá.

xem thông tin v khoá ng s d ng trong SQL Server ta làm nh sau:

- Ch n i t ng c n xem khoá
- Th c hi n th t c sp_lock

Ch n ki u khoá.

Khoá oc t trong các câu l nh nh : SELECT, INSERT, UPDATE, và DELETE , sau ây là b ng mô t các ki u khoá i v i ph ng th c nói trên

Tên	Mô t
NOLOCK	c s d ng v i câ l nh Select, ng i oc có th c d li u khi d li u g c khi ch a c ghi d li u m i trong giao d ch ang s d ng
HOLDLOCK	Khoa Shared c gi n khi phiên giao d ch c hoàn t t khi khoá ch a c gi i phóng
UPDLOCK	Dùng c p nh t d li u c a ki u khoá Shared trong quá trình c b ng d li u và c gi n khi k t thúc l nh c a phiên giao d ch. Khoa này dùng khi c p nh t d li u, ng n không cho ng i khác c n khi phiên giao d ch c p nh t c hoàn t t
TABLOCK	Dùng khoá Shared trên m t b ng d li u, cho phép nh ng ng i khác c d li u nh ng ng n không cho c p nh t
PAGELOCK	s d ng ki u khoá Shared ph m vi trang d li u (Page), ây là

	lo i khoá ng m nh
TABLOCKX	Dùng ki u khoá Exclusive trên m t b ng d li u, ng n ng i khác c và c p nh t d li u t b ng và gi n khi k t thúc l nh, phiên giao d ch

- Cách t khoá nh sau: Dùng l nh SET
- Ví d 1:

```
USE pubs
GO
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE
GO
BEGIN TRANSACTION
SELECT au_lname FROM authors WITH (NOLOCK)
GO
```

- V d? 2:
Select * from authors(UPDLOCK)

t m c khoá.

Dùng t, i u khi n khoá trong các giao d ch c a SQL Server

- Cú pháp:

```
SET TRANSACTION ISOLATION LEVEL
{ READ COMMITTED
| READ UNCOMMITTED
| REPEATABLE READ
| SERIALIZABLE
}
```

Trong ó:

- Read Committed: Dùng ki u khoá Shared trong quá trình c d li u
- Read Uncommitted: Không t khoá Shared và khoá Exclusive, có th c d li u g c khi ang có phiên giao ch s a i d li u
- RepeatTable Read: Khoá t t c d li u ang c s d ng trong truy v n, ng n nh ng ng i khác s a d li u nh ng ng i khác có th chèn thêm d li u m i vào b ng (hàng m i)
- Serializable: t khoá trong m t t p d li u (khoá ph m vi) ng n không cho ng i khác có th s a, thêm hàng m i vào t p d li u n khi giao d ch k t thúc, t ng nh HoldLock trong l nh Select
- Ví d :

```
SET TRANSACTION ISOLATION LEVEL REPEATABLE READ
```

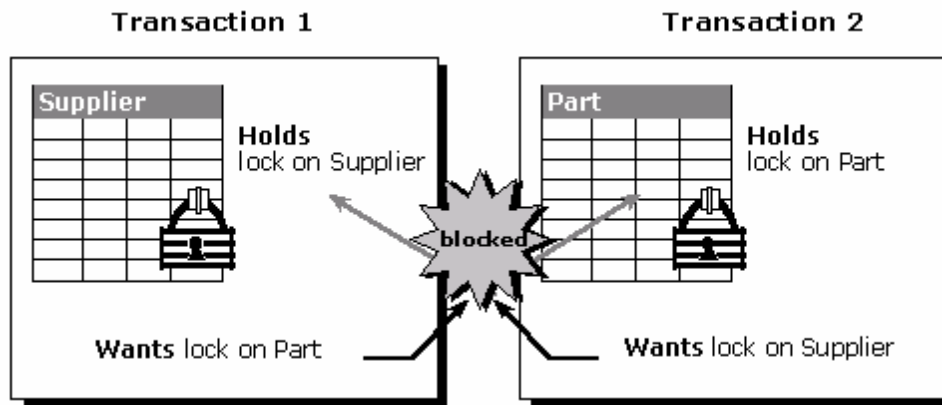
```

GO
BEGIN TRANSACTION
SELECT * FROM publishers
SELECT * FROM authors
...
COMMIT TRANSACTION

```

Khoá chết (DeadLock).

Trong hệ quản trị CSDL quan hệ nối riegel và các hệ quản trị khác nói chung, vì các xu hướng nhiều luồng dữ liệu đồng thời trong CSDL là thường xuyên xảy ra, một giao dịch có thể lấy dữ liệu từ nhiều nguồn dữ liệu khác nhau, hai giao dịch trong cùng CSDL có thể cùng chung một nguồn dữ liệu nào đó nên vì các giao dịch này tất các mức khoá khác nhau cho các nguồn dữ liệu mà nó nắm giữ là không thể tránh khỏi, ví dụ tên sau mô tả giao chéo về nguồn dữ liệu trong giao dịch



Trong giao dịch 1 và 2 sử dụng các bảng dữ liệu mức khoá Exclusive, ngay giao dịch 1 chỉ thể hiện được khi giao dịch 2 thể hiện xong hoặc quay lại trạng thái ban đầu, ngược lại giao dịch 2 cũng chờ giao dịch 1 thể hiện xong hoặc quay lại trạng thái ban đầu. Chính vậy thì cả 2 giao dịch sẽ không bao giờ kết thúc được phiên giao dịch của mình. Phần chung của khoá nói trên gọi là khoá chết, và các khoá theo kiểu (block).

GRAND – GÁN QUYỀN.

Loại Grand thể hiện gán quyền cho user hoặc role của SQL Server. Người thể hiện Grand phải có quyền thể hiện phân quyền cho user. Có 2 hình thức gán quyền: gán quyền thể hiện câu lệnh, gán quyền thao tác với dữ kiện.

Gán quyền thao tác câu lệnh.

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

Các câu lệnh:

- CREATE DATABASE
- CREATE DEFAULT
- CREATE FUNCTION
- CREATE PROCEDURE
- CREATE RULE
- CREATE TABLE
- CREATE VIEW
- BACKUP DATABASE
- BACKUP LOG

SQL Server ng m nh m t s nhóm có quy n th c hi n câu l nh nh sau:

	dbcreator	processadmin	securityadmin	serveradmin	bulkadmin
ALTER DATABASE	X				
CREATE DATABASE	X				
BULK INSERT					X
DBCC				X (1)	
DENY			X (2)		
GRANT			X (2)		
KILL		X			
RECONFIGURE				X	
RESTORE	X				
REVOKE			X (2)		
SHUTDOWN				X	

	db_ owner	db_ datareader	db_ datawriter	db_ ddladmin	db_ backup operator	db_ security admin
ALTER DATABASE	X			X		
ALTER FUNCTION	X			X		
ALTER PROCEDURE	X			X		
ALTER TABLE	X (1)			X		
ALTER TRIGGER	X			X		
ALTER VIEW	X (1)			X		
BACKUP	X				X	
CHECKPOINT	X				X	
CREATE DEFAULT	X			X		
CREATE FUNCTION	X			X		
CREATE INDEX	X (1)			X		
CREATE PROCEDURE	X			X		
CREATE RULE	X			X		
CREATE TABLE	X			X		
CREATE TRIGGER	X (1)			X		
CREATE VIEW	X			X		
DBCC	X				X (2)	
DELETE	X (1)		X			
DENY	X					X
DENY on object	X					
DROP	X (1)			X		
EXECUTE	X (1)					
GRANT	X					X
GRANT on object	X (1)					
INSERT	X (1)		X			
READTEXT	X (1)	X				
REFERENCES	X (1)			X		
RESTORE	X					
REVOKE	X					X
REVOKE on object	X (1)					

SELECT	X (1)	X				
SETUSER	X					
TRUNCATE TABLE	X (1)			X		
UPDATE	X (1)		X			
UPDATE STATISTICS	X (1)					
UPDATETEXT	X (1)		X			
WRITETEXT	X (1)		X			

Các user c gán quy n có th là user c a SQL Server ho c user c a Windows NT.

Ví d gán quy n thao tác câu l nh cho 3 user (trong ó có 2 user c a SQL Server và 1 user c a Windows NT):

```
GRANT CREATE DATABASE, CREATE TABLE
TO Mary, John, [Corporate\BobJ]
```

Ví d gán quy n thao tác cho role và user:

```
USE pubs
GO
```

```
GRANT SELECT
ON authors
TO public
GO
```

```
GRANT INSERT, UPDATE, DELETE
ON authors
TO Mary, John, Tom
GO
```

Gán quy n thao tác i t ng.

Là vi c gán quy n cho các user ho c role có quy n thao tác v i các i t ng c a SQL Server.

Ví d gán quy n thao tác cho Role:

```
GRANT CREATE TABLE TO Accounting
```

Ví dụ gán quyền gán quyền thao tác cho user khác: Ví dụ Jean là dbo của bảng Plan_data, Jean thực hiện gán quyền với chức năng GRAND_OPTION cho role accounting, Jill thực hiện role nói trên và Jill gán quyền chức năng Select cho Jack, Jack không là thành viên của Accounting.

```
/* User Jean */  
GRANT SELECT ON Plan_Data TO Accounting WITH GRANT  
OPTION
```

```
/* User Jill */  
GRANT SELECT ON Plan_Data TO Jack AS Accounting
```

Thực thi sp_grantlogin.

Là thực thi thực hiện gán quyền truy cập cho user của Windows NT hoặc nhóm user của Windows NT.

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

Ví dụ gán quyền truy cập SQL Server cho BobJ.

```
EXEC sp_grantlogin 'Corporate\BobJ'
```

Thực thi sp_grantdbaccess.

Gán quyền khai thác cho user của SQL Server hoặc Windows NT.

```
sp_grantdbaccess [@loginame =] 'login'  
[,[@name_in_db =] 'name_in_db' [OUTPUT]]
```

Ví dụ gán quyền khai thác cho user của Windows và lấy theo tên máy.

```
EXEC sp_grantdbaccess 'Corporate\GeorgeW', 'Georgie'
```

REVOKE – T C QUY N.

Revoke là câu lệnh tắt quy n khai thác c a user.

T c quy n c th c hi n câu l nh.

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

Ví dụ tắt c quy n khai thác v i 2 user:

```
REVOKE CREATE TABLE FROM Joe, [Corporate\BobJ]
```

Ví dụ tắt c quy n khai thác 2 câu l nh v i các user:

```
REVOKE CREATE TABLE, CREATE DEFAULT  
FROM Mary, John
```

T c quy n khai thác c a user v i i t ng.

Ví dụ tắt c quy n th c hi n l nh Select trong role Budget_data i v i Mary:

```
REVOKE SELECT ON Budget_Data TO Mary
```

DENY – T CH I QUY N.

Là câu lệnh tắt ch i quy n i v i user, user ch th c hi n c quy n khi có ch nh rõ ràng.

Ví dụ tắt ch i quy n th c hi n l nh v i các user:

```
DENY CREATE DATABASE, CREATE TABLE  
TO Mary, John, [Corporate\BobJ]
```


Ví dụ gán quyền khai thác cho role, sau đó th c hi n t ch i th c hi n c a các user trong role:

```
USE pubs  
GO
```

```
GRANT SELECT  
ON authors  
TO public  
GO
```

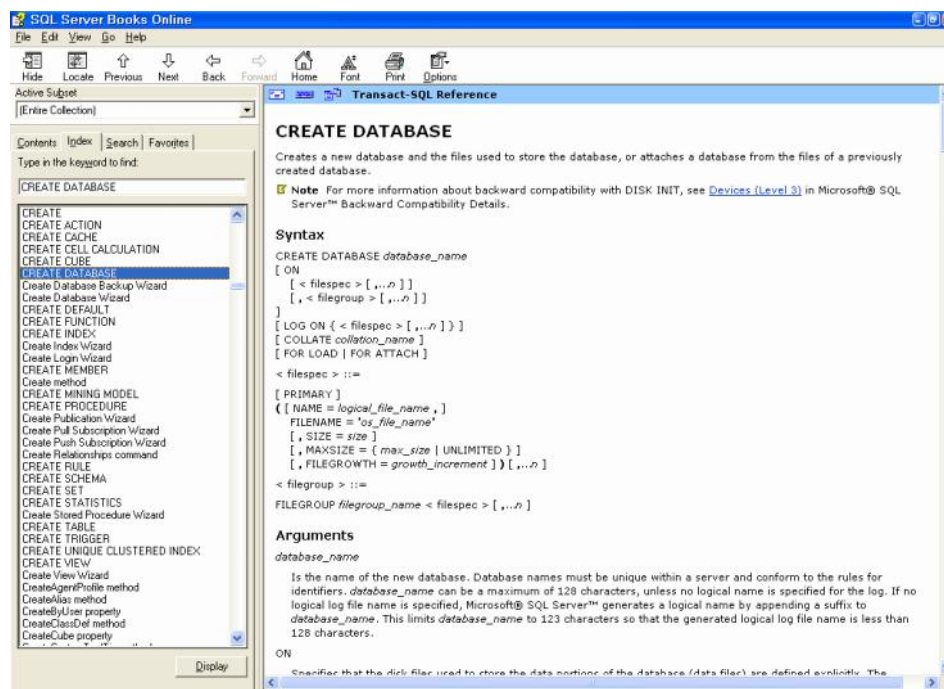
```
DENY SELECT, INSERT, UPDATE, DELETE  
ON authors  
TO Mary, John, Tom
```

Ví dụ t ch i quy n c a role:

```
DENY CREATE TABLE TO Accounting
```

TR GIÚP.

Trong quá trình th c hi n s o n l nh T-SQL b n có th th c hi n tra c u l nh trong Book Online.



Phần 3. PHÁT TRIỂN NG D NG V I SQL SERVER

Trong phần này ta sẽ xem xét kỹ thuật phát triển ứng dụng với SQL Server từ các ngôn ngữ lập trình (Visual Basic, C++, VBScript,...). Các ứng dụng khai thác CSDL của SQL Server thể hiện các bước sau:

- + Kết nối ứng dụng với SQL Server.
- + Xây dựng cơ sở dữ liệu.
- + Thể hiện các lệnh khai thác hoặc thao tác của SQL Server.
- + Khai thác dữ liệu thông qua công cụ có sẵn.
- + Ngắt kết nối.

GIỚI THIỆU.

Thi thoảng ứng dụng là ví dụ thể hiện tổ giao diện (API – Application Program Interface) giao tiếp với SQL Server, ví dụ thể hiện kết nối thể hiện thông qua các công cụ ADO, URL, OLE DB, ODBC, Embedded SQL for C, DB-Library. Khi sử dụng các công cụ kết nối dữ liệu thao tác dữ liệu bằng hoặc truy vấn tài liệu XML.

- + Dữ liệu dữ liệu bằng thể hiện thông qua các công cụ kết nối ADO, OLE DB, ODBC, Embedded SQL for C, DB-Library.

- + Dữ liệu thể hiện thông qua tài liệu XML thông qua các công cụ ADO, URL, OLE DB.

KẾT NỐI VỚI SQL SERVER BẰNG ADO.

ADO viết tắt của thuật ngữ ActiveX Data Object là công cụ giao tiếp với dữ liệu của nhiều hệ quản trị CSDL khác nhau, SQL Server là một ví dụ cho ví dụ giao tiếp.

ADO sử dụng với CSDL quan hệ hoặc sử dụng với CSDL phân tán, khi đó gọi là ADO MD (ADO Multi Dimension). ADO sử dụng kết nối kiểu OLE DB hoặc các thể hiện kết nối COM (Component Object Model).

OLE DB sử dụng 2 phương thức Microsoft OLE DB Provider for SQL Server (SQLOLEDB) và Microsoft OLE DB Provider for ODBC (MSDASQL).

ADO có thể thực hiện tất cả ngôn ngữ lập trình Visual Basic, ASP, C++.

Cấu trúc ngữ cảnh sử dụng ADO.

ADO gồm các thành phần cơ bản sau: Application, ADO, OLE DB Provider, Data Source.

Thành phần	Chức năng
Application	Giữ các đối tượng, thành phần, phương thức và các thuộc tính của ADO. Thông qua các thành phần này người dùng gửi các câu lệnh SQL và nhận kết quả xử lý.
ADO	Quản lý việc trao đổi dữ liệu giữa người dùng và OLE DB
OLE DB provider	Xử lý các lệnh gửi người dùng qua ADO, kết nối với Data Source. Processes all ADO calls from the application, connects to a data source, passes SQL statements from the application to the data source, and returns results to the application.
Data source	Contains the information used by a provider to access a specific instance of data in a DBMS.

Khi thực hiện lập trình người dùng với SQL Server sử dụng ADO, người lập trình phải thực hiện các thao tác sau:

- + Kết nối nguồn dữ liệu (data source).
- + Gửi câu lệnh SQL đến nguồn dữ liệu.
- + Xử lý kết quả nhận về từ câu lệnh gửi.
- + Xử lý các lỗi và thông báo.
- + Ngắt kết nối nguồn dữ liệu.

Ở mức độ người dùng phần mềm sử dụng ADO có thể sử dụng một số thao tác sau:

- + Sử dụng con trỏ (cursor) để lưu khi cần vị trí trong tập kết quả.
- + Thực hiện thao tác lưu trữ trên Server.
- + Thực hiện hàm toán học trên Server.
- + Quản lý các phép truy vấn mà có nhiều tập kết quả.

- + Yêu cầu kết thúc hoạt động phiên giao dịch.
- + Quản lý các thao tác văn bản (text, image).
- + Thực hiện các thao tác văn bản XML sử dụng phép truy vấn XPath.

Kết nối SQL Server.

Kết nối SQL Server, các công việc cần thực hiện như sau:

- + Cấu hình kết nối.
- + Thiết lập nguồn kết nối người dùng.
- + Xác định OLE DB provider.
- + Thực hiện truy vấn.
- + Quản lý các phiên làm việc trên kết nối.

Khi sử dụng SQLOLEDB ta phải thực hiện một số thuộc tính sau cho kết nối:

- + **Initial Catalog:** Xác định CSDL.
- + **Data Source:** Xác định tên Server.
- + **Integrated Security:** Xác định cách xác thực, nếu là **SSPI** thì xác thực là Windows Authentication, hoặc xác định **User ID, Password** cách xác thực SQL Server Authentication.

Ví dụ thực hiện kết nối SQL Server một số thuộc tính tiêu biểu Visual Basic:

```
' Initialize variables.
Dim cn As New ADODB.Connection
. . .
Dim ServerName As String, DatabaseName As String, _
    UserName As String, Password As String

' Put text box values into connection variables.
ServerName = txtServerName.Text
DatabaseName = txtDatabaseName.Text
```

```

UserName = txtUserName.Text
Password = txtPassword.Text

' Specify the OLE DB provider.
cn.Provider = "sqloledb"

' Set SQLOLEDB connection properties.
cn.Properties("Data Source").Value = ServerName
cn.Properties("Initial Catalog").Value = DatabaseName

' Decision code for login authorization type:
' Windows NT or SQL Server authentication.
If optWinNTAuth.Value = True Then
    cn.Properties("Integrated Security").Value = "SSPI"
Else
    cn.Properties("User ID").Value = UserName
    cn.Properties("Password").Value = Password
End If

' Open the database.
cn.Open

```

Víd k t n i n SQL Servers d ng chu i k t n i:

```

' Initialize variables.
Dim cn As New ADODB.Connection
Dim provStr As String

' Specify the OLE DB provider.
cn.Provider = "sqloledb"

' Specify connection string on Open method.
ProvStr =
"Server=MyServer;Database=northwind;Trusted_Connection=
yes"
cn.Open provStr

```

Víd k t n i s d ng ODBC:

```
Dim cn As New ADODB.Connection

cn.ConnectionTimeout = 100
' DSN connection. You can use variables for the
parameters.
cn.Open "MyDataSource", "sa", "MyPassword"
' Alternative syntax follows:
' cn.Open "DSN=DataSourceName;UID=sa;PWD=Password;"

cn.Close
```

Ví dụ kết nối trực tiếp Driver của SQL Server:

```
Dim cn As New ADODB.Connection

' Connection to SQL Server without using ODBC data
source.
cn.Open "Driver={SQL
Server};Server=Server1;Uid=SA;Pwd=;Database=northwind"

cn.Close
```

Thao tác truy vấn.

Thao tác truy vấn sử dụng đối tượng Command.

cmd.Execute(NumRecords, Parameters, Options)

Đối tượng Command có thể thực hiện nhiều kiểu câu lệnh (Select, Update, Insert, Delete, Create, Drop), riêng kiểu Select kết quả thực hiện là một recordset.

Set rs = cmd.Execute(NumRecords, Parameters, Options)

Kiểu lệnh thao tác trong Command được xác định theo option của lệnh, gồm một số kiểu sau:

Tên kiểu	Mô tả
adCmdFile	Tên file chứa đối tượng recordset

adCmdStoreProc	Stored procedure
adCmdTable	Tên bảng
adCmdTableDirect	Tên bảng mà các cột được truy vấn
adCmdText	Câu lệnh SQL
adCmdUnknown	Chưa xác định
adCmdUnspecified	Chưa xác định tham số cho lệnh

Thích hiện truy vấn thông qua đối tượng connection.

```
Dim cn As New ADODB.Connection
. . .
Dim rs As New ADODB.Recordset

cmd1 = txtQuery.Text
Set rs = cn.Execute(cmd1)
```

Thích hiện truy vấn có sử dụng tham số.

Khi thích hiện các thủ tục có tham số truy vấn vào các ứng dụng phi truy vấn tham số, trong phần này sẽ giới thiệu một ví dụ sử dụng đối tượng parameter.

Tổng kết:

```
USE NORTHWIND
GO
drop proc myADOParaProc
GO
CREATE PROC myADOParaProc
@categoryid int(4)
AS
SELECT * FROM products WHERE categoryid = @categoryid
GO
```

Sử dụng đối tượng parameter truy vấn tham số là sử dụng nguyên xác định categoryID:

```
Dim cn As New ADODB.Connection
Dim cmd As New ADODB.Command
Dim rs As New ADODB.Recordset
Dim prm As ADODB.Parameter
```

```

Dim fld As ADODB.Field
Dim provStr As String

' Connect using the SQLOLEDB provider.
cn.Provider = "sqloledb"

' Specify connection string on Open method.
provStr =
"Server=MyServer;Database=northwind;Trusted_Connection=yes"
cn.Open provStr

' Set up a command object for the stored procedure.
Set cmd.ActiveConnection = cn
cmd.CommandText = "myADOParaProc"
cmd.CommandType = adCmdStoredProc
cmd.CommandTimeout = 15

' Set up a new parameter for the stored procedure.
Set prm = Cmd.CreateParameter("CategoryID", adInteger,
adParamInput, 4, 7)
Cmd.Parameters.Append prm

' Create a recordset by executing the command.
Set rs = cmd.Execute
Set Flds = rs.Fields

' Print the values for all rows in the result set.
While (Not rs.EOF)
    For Each fld in Flds
        Debug.Print fld.Value
    Next
    Debug.Print ""
    rs.MoveNext
Wend

' Close recordset and connection.
rs.Close

cn.Close

```

it ng Recordset.

S d ng it ng Recordset l u tr k t qu c a l nh Select.


```

Dim cn As New ADODB.Connection
Dim rs As ADODB.Recordset
. . .
cmd1 = txtQuery.Text
Set rs = New ADODB.Recordset
rs.Open cmd1, cn
rs.MoveFirst
. . .
' Code to loop through result set(s)

```

it ng Field.

S d ng it ng field là các c t c a Recordset, thông qua nó ta có th lấy giá tr, thu c tính c a c t.

```

Dim rs As New ADODB.Recordset
Dim fld As ADODB.Field
Dim cn As ADODB.Connection
Dim cmdText As String

cn.Provider = "sqloledb"
cn.Properties("Data Source").Value = "MyServerName"
cn.Properties("Initial Catalog").Value = "northwind"
cn.Properties("Integrated Security").Value = "SSPI"
cn.Open

cmdText = "select * from authors"

rs.Open cmdText, cn
Set Flds = rs.Fields
Dim TotalCount As Integer
TotalCount = Flds.Count

For Each fld In Flds
    Debug.Print fld.Name
    Debug.Print fld.Type
    Debug.Print fld.Value
Next
rs.Close

```

S d ng con tr .

Khi sử dụng iterator Recordset của ADO, ta có thể sử dụng nhiều kiểu con tr khác nhau xác định kiểu khóa, kiểu khi cần vị trí,...

```
Dim rs As New ADODB.Recordset
. . .
rs.Open "SELECT * FROM titles", , adOpenDynamic,
adLockOptimistic

rs.Close
```

Con tr nói trên gồm những thuộc tính cơ bản sau: **CursorType**, **CursorLocation**, **LockType**, **CacheSize**.

Thuộc tính	Mô tả
CursorType	<ul style="list-style-type: none"> - adOpenForwardOnly: Kiểu mặc định. Xác định kiểu con tr chỉ sử dụng: - adOpenForwardOnly: Chỉ đọc, chỉ có thể cập nhật dữ liệu trên hàng dữ liệu hiện tại. - adOpenStatic: Trạng thái tĩnh, khi mở kiểu này hệ thống sẽ cung cấp một bản sao dữ liệu (snapshot), dữ liệu thay đổi trên bảng cơ sở dữ liệu không thể hiện trên snapshot dữ liệu này. - adOpenKeyset: Theo vị trí tùy chọn, khi di chuyển hàng cập nhật con tr kiểu này sẽ chỉ ra hàng dữ liệu cũ, hàng dữ liệu cũ khó và bản có thể cập nhật, lý do dữ liệu hàng cũ. - adOpenDynamic: Động, con tr kiểu này giống với keyset cursor, nhưng con tr kiểu này phản ánh những thay đổi trên bảng cơ sở dữ liệu.
CursorLocation	<ul style="list-style-type: none"> - adUseServer: Kiểu mặc định. - adUseClient: Nếu true là ta chỉ có thể mở trạng thái tĩnh.
LockType	<ul style="list-style-type: none"> - adLockReadOnly: Kiểu mặc định. <p>Xác định kiểu khóa trong quá trình cập nhật dữ liệu (adLockPessimistic, adLockOptimistic, adLockBatchOptimistic).</p>
CacheSize	<p>Giá trị mặc định: 1</p> <p>Xác định số hàng trong bộ nhớ cache trong một thời điểm.</p>

Các phương thức di chuyển hàng dữ liệu.

Khi sử dụng iterator Recordset bạn có thể di chuyển vị trí của hàng dữ liệu bằng các phương thức **MoveFirst**, **MoveLast**, **MoveNext**, **MovePrevious**.
Ánh xạ vị trí theo phương thức **Bookmark**, phương thức **clone** tạo một bản sao recordset.

Quản lý phiên làm việc.

Trong phần câu lệnh T-SQL ta đã xem xét việc sử dụng khi cần thực hiện làm việc (transaction), tuy nhiên ta có thể sử dụng iterator connection của ADO sử dụng khi cần thực hiện phiên làm việc như trong kịch bản như nói trên bằng việc sử dụng các phương thức **BeginTrans**, **CommitTrans**, **RollbackTrans**. Xét ví dụ sau:

```
Dim cn As New ADODB.Connection
Dim rs As New ADODB.Recordset

. . .
' Open connection.
cn.Open

' Open titles table.
rs.Open "SELECT * FROM titles", Cn, adOpenDynamic,
adLockPessimistic

. . .
' Begin the transaction.
rs.MoveFirst
cn.BeginTrans

' User loops through the recordset making changes.
. . .
' Ask if the user wants to commit all the changes made.
If MsgBox("Save all changes?", vbYesNo) = vbYes Then
    cn.CommitTrans
Else
    cn.RollbackTrans
End If
```

Thực hiện các lệnh DDL.

Thực hiện các lệnh DDL như **CREATE TABLE**, **DROP TABLE**, **ALTER TABLE**. Bạn có thể sử dụng iterator command của ADO, xét ví dụ sau:

```
Dim Cn As New ADODB.Connection
```

```

Dim Cmd As New ADODB.Command

' If the ADOTestTable does not exist, go to AdoError.
On Error GoTo AdoError

' Connect using the SQLOLEDB provider.
cn.Provider = "sqloledb"
cn.Properties("Data Source").Value = "MyServerName"
cn.Properties("Initial Catalog").Value = "northwind"
cn.Properties("Integrated Security").Value = "SSPI"
cn.Open

' Set up command object.
Set Cmd.ActiveConnection = Cn
Cmd.CommandText = "DROP TABLE ADOTestTable"
Cmd.CommandType = adCmdText
Cmd.Execute

Done:
    Cmd.CommandText = "SET NOCOUNT ON"
    Cmd.Execute
    Cmd.CommandText = "CREATE TABLE ADOTestTable (id
int, name char(100))"
    Cmd.Execute
    Cmd.CommandText = "INSERT INTO ADOTestTable
values(1, 'Jane Doe')"
    Cmd.Execute
    Cn.Close
Exit Sub

AdoError:
    Dim errLoop As Error
    Dim strError As String

    ' Enumerate Errors collection and display
properties of
    ' each Error object.
    Set Errs1 = Cn.Errors
    For Each errLoop In Errs1
        Debug.Print errLoop.SQLState
        Debug.Print errLoop.NativeError
        Debug.Print errLoop.Description

```

Next

GoTo Done

End Sub

Quy n lý d li u ki u l n – Text, image.

D li u ki u text, ntext, image là ki u d li u ph c t p, vì c quy n lý, khai thác không c th c hi n thông th ng, ADO h tr các ph ng th c riêng th c hi n.

Thay vì c, c p nh t d li u tr c ti p thì d li u ki u này c thao tác theo o n (chunk) b ng cách s d ng các ph ng th c **AppendChunk**, **GetChunk**.

Tr c khi th c hi n b n ph i t tham s b ng cách th c hi n l nh sau:

```
EXEC sp_dboption 'pubs', 'Select into/bulkcopy', 'True'
```

Xét ví d sau trên CSDL Pubs:

- Copy b ng pub_info sang b ng m i

```
USE pubs
SELECT * INTO pub_info_x
FROM pub_info
GO
```

- Th c hi n chèn d li u vào b ng:

```
Public Sub AppendChunkX()

Dim cn As ADODB.Connection
Dim rstPubInfo As ADODB.Recordset
Dim strCn As String
Dim strPubID As String
Dim strPRInfo As String
Dim lngOffset As Long
Dim lngLogoSize As Long
Dim varLogo As Variant
Dim varChunk As Variant

Const conChunkSize = 100
```

```

' Open a connection.
Set cn = New ADODB.Connection
strCn = "Server=srv;Database=pubs;UID=sa;Pwd=;"

cn.Provider = "sqloledb"
cn.Open strCn

'Open the pub_info_x table.
Set rstPubInfo = New ADODB.Recordset
rstPubInfo.CursorType = adOpenDynamic
rstPubInfo.LockType = adLockOptimistic
rstPubInfo.Open "pub_info_x", cn, , , adCmdTable

'Prompt for a logo to copy.
strMsg = "Available logos are : " & vbCr & vbCr

Do While Not rstPubInfo.EOF
    strMsg = strMsg & rstPubInfo!pub_id & vbCr & _
        Left(rstPubInfo!pr_info,
            InStr(rstPubInfo!pr_info, ",") - 1) & vbCr &
vbCr
    rstPubInfo.MoveNext
Loop

strMsg = strMsg & "Enter the ID of a logo to copy:"
strPubID = InputBox(strMsg)

' Copy the logo to a variable in chunks.
rstPubInfo.Filter = "pub_id = '" & strPubID & "'"
lngLogoSize = rstPubInfo!logo.ActualSize
Do While lngOffset < lngLogoSize
    varChunk = rstPubInfo!logo.GetChunk(conChunkSize)
    varLogo = varLogo & varChunk
    lngOffset = lngOffset + conChunkSize
Loop

' Get data from the user.
strPubID = Trim(InputBox("Enter a new pub ID:"))
strPRInfo = Trim(InputBox("Enter descriptive
text:"))

```

```

' Add a new record, copying the logo in chunks.
rstPubInfo.AddNew
rstPubInfo!pub_id = strPubID
rstPubInfo!pr_info = strPRInfo
lngOffset = 0 ' Reset offset.

Do While lngOffset < lngLogoSize
    varChunk = LeftB(RightB(varLogo, lngLogoSize - _
        lngOffset), conChunkSize)
    rstPubInfo!logo.AppendChunk varChunk
    lngOffset = lngOffset + conChunkSize
Loop

rstPubInfo.Update

' Show the newly added data.
MsgBox "New record: " & rstPubInfo!pub_id & vbCr & _
    "Description: " & rstPubInfo!pr_info & vbCr & _
    "Logo size: " & rstPubInfo!logo.ActualSize

rstPubInfo.Close
cn.Close

End Sub

```

K t n i t ASP.

Trong ác ví d̄ saiū ây th̄ c hi n làm vi c v i SQL Server t̄ ASP, s̄ d̄ ng ngôn nḡ l̄ p̄ tr̄nh VBScript, làm c̄ ví d̄ này b̄ n̄ c̄ ph̄ i có ki n th̄ c v̄ thi t̄ k̄ Web site (HTML).

Thi t̄ k̄ form k̄ t̄ n̄ i:

```

<%@LANGUAGE="VBSCRIPT" CODEPAGE="65001"%>
<html>
<head>
<title>Login SQL Server example</title>
<meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
<style type="text/css">
<!--

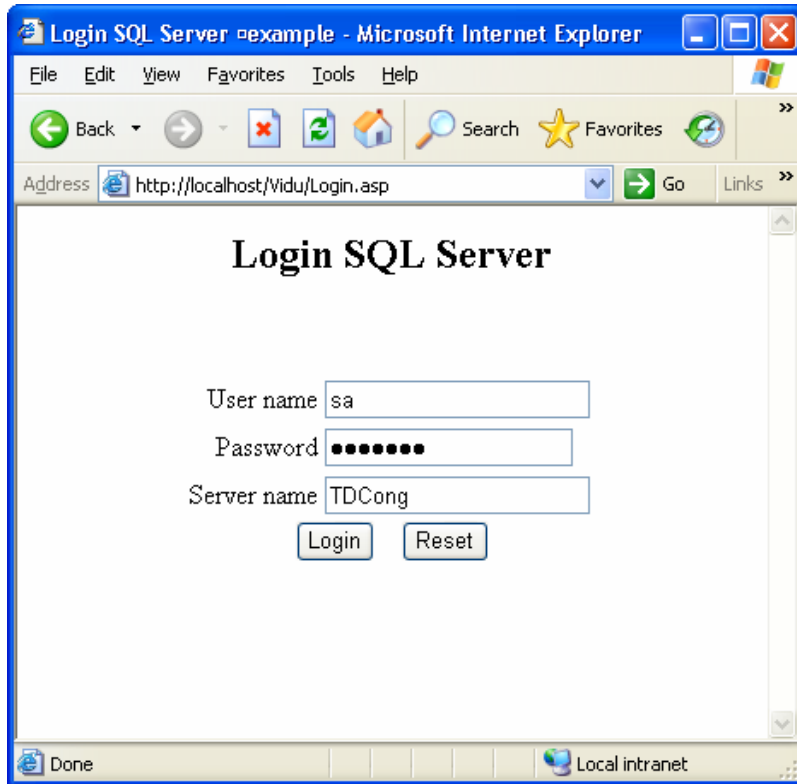
```



```

frmlogin.method="post"
frmlogin.action="connect.asp"
frmlogin.submit
end sub
</script>

```



T p tin connect.asp: Là t p tin c g i t form login.asp, th c hi n nh n tham s c a form login.asp, k t n i n SQL Server.

```

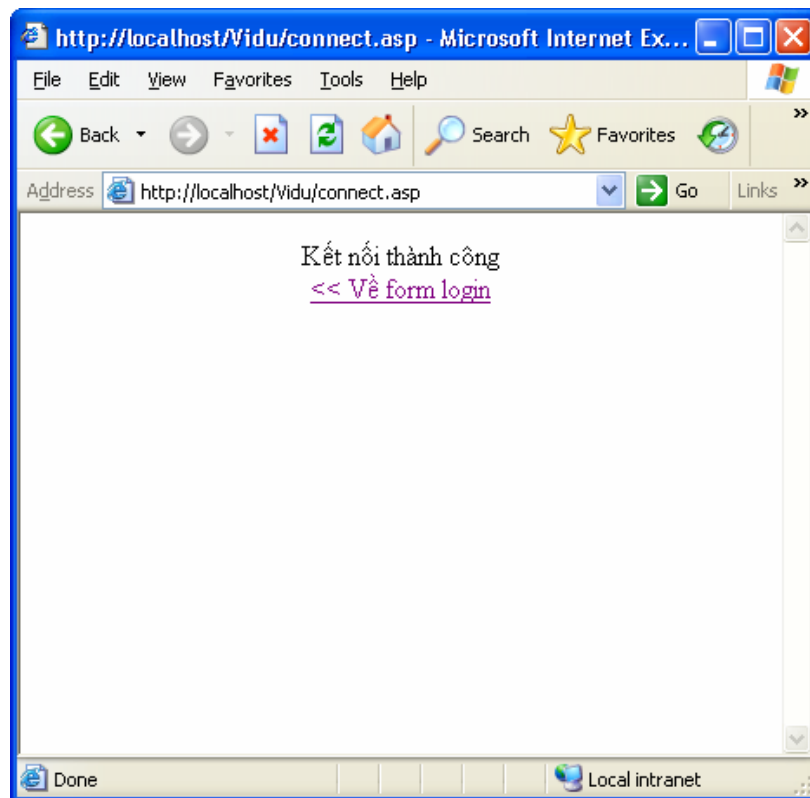
<%@LANGUAGE="VBSCRIPT" CODEPAGE="1252"%>
<%
dim username, password, servername, txt
username=request.Form("txtUser")
password = request.Form("txtPassword")
servername=request.Form("txtServer")
txt= "Provider=SQLOLEDB; "
txt=txt & " Data Source=" & servername & ";"
txt=txt & " Initial Catalog=pubs; "
txt=txt & " User ID=" & username & ";"
txt=txt & " PWD=" & password

```

```

Set cn=Server.CreateObject("ADODB.Connection")
cn.Open txt
%>
<div align="center"><span class="style1">K&#7871;t
n&#7889;i th&agrave;nh c&ocirc;ing</span>
<%
cn.close
%>
<br>
<a href="Login.asp"> V&#7873; form login</a>
</div>

```



Li t kê danh sách.

li t kê danh sách (có th l y d li u b ng cách truy v n tr c ti p ho c thông qua khung nhìn - view) tr c h t ph i t o m t recordset l u tr k t qu truy v n, t recordset ta có th l y d li u và t vào v trí t ng ng c n thi t.

+ *Khai báo Recordset:*

```

Set rs=Server.CreateObject("ADODB.Recordset")
rs.ActiveConnection =cn

```

```
rs.Source = "Select * from Authors"
rs.Open
```

+ *L y giá tr :*

```
Rs.fields("au_id")
```

+ *óng Recordset:*

```
Rs.close
```

+ *Ví d li t kê danh sách b ng cách truy v n tr c ti p:*

```
<%@LANGUAGE="VBSCRIPT" CODEPAGE="65001"%>
<html>
<head>
<title>Danh sach</title>
<meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
<style type="text/css">
<!--
.style5 {
    font-family: Arial, Helvetica, sans-serif;
    font-weight: bold;
    font-size: 14px;
}
.style6 {font-family: Arial, Helvetica, sans-serif}
.style7 {
    font-size: 14px;
    font-weight: bold;
}
.style9 {font-family: Arial, Helvetica, sans-serif;
font-size: 14px; }
.style12 {font-size: 12px}
-->
</style>
</head>
<%
dim username, password, servername, txt
username="sa"
```

```

password = ""
servername="TDCong"
txt= "Provider=SQLOLEDB; "
txt=txt & " Data Source=" & servername & ";"
txt=txt & " Initial Catalog=pubs; "
txt=txt & " User ID=" & username & ";"
txt=txt & " PWD=" & password
Set cn=Server.CreateObject("ADODB.Connection")
cn.Open txt

Set rs=Server.CreateObject("ADODB.Recordset")
rs.ActiveConnection =cn
rs.Source ="Select * from Authors"
rs.Open
%>
<body>
<p align="center"><strong>LIST OF AUTHORS
</strong></p>
<p>&nbsp;</p>
<table width="100%" border="0.2">
  <tr bgcolor="#999999">
    <td width="5%"><div align="center"
class="style5">No</div></td>
    <td width="14%"><div align="center" class="style6
style7">au_id</div></td>
    <td width="14%"><div align="center"
class="style9"><strong>au_lname</strong></div></td>
    <td width="14%"><div align="center"
class="style9"><strong>au_fname</strong></div></td>
    <td width="14%"><div align="center"
class="style9"><strong>phone</strong></div></td>
    <td width="26%"><div align="center"
class="style9"><strong>address</strong></div></td>
    <td width="13%"><div align="center"
class="style9"><strong>city</strong></div></td>
  </tr>
  <%
    i=0
    do while not rs.eof and not rs.bof
      i=i+1
      if i mod 2<>0 then
%>

```

```

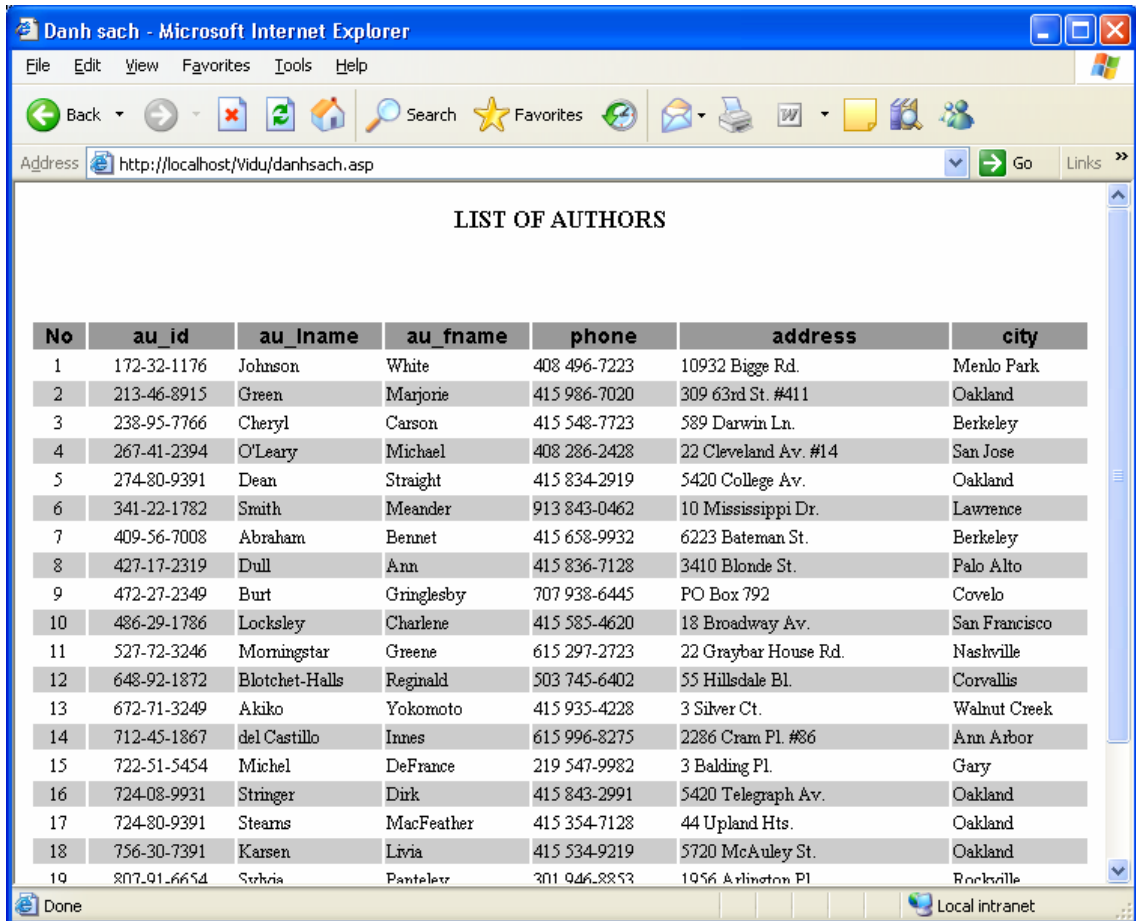
    <tr bgcolor="#FFFFFF">
        <td><div align="center"><span
class="style12"><%=i%></span></div></td>
        <td><div align="center"><span
class="style12"><%=rs.fields("au_id")%></span></div></t
d>
        <td><span
class="style12"><%=rs.fields("au_fname")%></span></td>
        <td><span
class="style12"><%=rs.fields("au_lname")%></span></td>
        <td><span
class="style12"><%=rs.fields("Phone")%></span></td>
        <td><span
class="style12"><%=rs.fields("Address")%></span></td>
        <td><span
class="style12"><%=rs.fields("City")%></span></td>
    </tr>
    <%
    else
    %>
    <tr bgcolor="#CCCCCC">
        <td><div align="center"><span
class="style12"><%=i%></span></div></td>
        <td><div align="center"><span
class="style12"><%=rs.fields("au_id")%></span></div></t
d>
        <td><span
class="style12"><%=rs.fields("au_lname")%></span></td>
        <td><span
class="style12"><%=rs.fields("au_fname")%></span></td>
        <td><span
class="style12"><%=rs.fields("Phone")%></span></td>
        <td><span
class="style12"><%=rs.fields("Address")%></span></td>
        <td><span
class="style12"><%=rs.fields("City")%></span></td>
    </tr>
    <%
    end if
    rs.movenext
loop
%>

```

```

</table>
</body>
<%
    rs.close
%>
</html>

```



K T N I V I SQL SERVER B NG SQL-DMO.

SQL DMO vi t t t c a c m t SQL Distributed Management Objects, s đ ng th vi n liên k t ng (dll) k t n i n SQL Server.

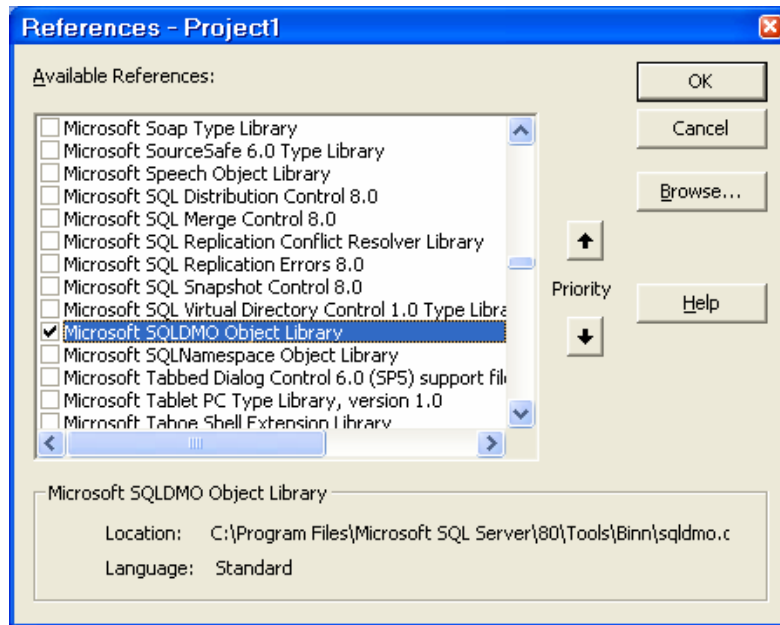
SQL DMO th c hi n liên k t nhúng (OLE Automation), các i t ng SQL Server c th c hi n nhúng các i t ng c a SQL Server vào ng d ng, khai thác các i t ng thông qua thu c tính, s ki n và các ph ng th c làm vi c c a nó.

SQL DMO hỗ trợ phát triển ứng dụng ngôn ngữ lập trình Visual Basic, C++, khi đóng gói các thành viên liên kết ứng dụng cùng, cài đặt ứng dụng thành viên sẽ cài đặt trong Windows, nên khi chuyển ứng dụng bản không cần thì lập môi trường Client Connectivity.

Các tệp tin cần bổ sung cho SQL DMO: sqldmo.dll, sqldmo80.hlp, sqldmo.rll, sqldmo.h (C++), sqldmoid.h (C++), sqldmo.sql. Trong phần này sẽ giới thiệu kỹ thuật kết nối ứng dụng Visual Basic 6.0.

Khai báo thành viên trong project.

- Vào menu Project -> References
- Chọn Microsoft SQL DMO Object Library -> Ok



Khai báo đối tượng.

Sau khi thành hiện khai báo thành viên trong project, ta có thể khai báo biến kiểu đối tượng (object) hoặc kiểu đối tượng của SQL DMO.

Ví dụ khai báo biến kiểu SQL Server:

```
Dim oSQLServer As SQLDMO.SQLServer
```

Kết nối SQL Server.

k t n i n SQL Server ta s d ng ph ng th c k t n i c a i t ng SQL Server, có 3 tham s Servername, LoginName, Password.

```
Dim oSQLServer As SQLDMO.SQLServer
Set oSQLServer = New SQLDMO.SQLServer
oSQLServer.Connect "ServerName", "LoginName",
"Password"
```

Th c hi n l i k t n i:

Trong nhi u tr ng h p b n mu n ng t k t n i h i n t i và th c hi n l i k t n i l y tr ng thái SQL Server h i n t h i (t ng t ng tác làm t i – Reresh).

```
oSQLServer.DisConnect
```

```
oSQLServer.ReConnect
```

Làm vi c v i các i t ng.

SQL DMO t o i t ng k th at nh ng i t ng con c a nó, ví d SQL Server k th at các i t ng Database <- Table <- Column,...

Xác nh bi n v i CSDL:

```
Dim oDatabase as new SQLDMO.Database
Set oDatabase = oSQLServer.Databases("Northwind")
```

L y danh sách tên các CSDL vào h p tho i:

```
Dim nDatabase as Integer
For nDatabase = 1 to oSQLServer.Databases.Count
    Combo1.AddItem oSQLServer.Databases(nDatabase).Name
Next nDatabase
```

Các i t ng u c k th à t các i t ng con, các i t ng con t o thành m t t p h p, t p h p nói trên có th th c hi n các ph ng th c Add, Remove,...v i t ng i t ng.

Ví d remove b ng kh i CSDL:

```
oServer.Databases("Northwind").Tables.Remove("Orders",
"anne")
```

Th c hi n l nh SQL:

Các i t ng (SQL Server, Database) có th th c hi n các l nh SQL thông qua các ph ng th c ExecuteImmediate và ExecuteWithResults.

Ví d th c hi n l nh thao tác:

```
oSQLServer.ExecuteImmediate "Create Database Example"
```

Ví d th c hi n l nh truy v n:

```
Dim rs As QueryResults  
Set rs = oDatabase.ExecuteWithResults("Select *  
from Authors")
```

Ví d l y d li u t m t truy v n:

```
For i = 1 To rs.Rows  
    For j = 1 To rs.Columns  
        MsgBox rs.GetColumnString(i, j)  
    Next j  
Next i
```

Các ph ng th c th c hi n k t n i có th h tr theo t ng ngôn ng l p trình, h tr nhi u trong vi c l p trình t Visual Basic, ASP, C, C++. B n có th tham kh o các ví d trong th m c Sample c a SQL Server. Các ví d s c p nhi u ngôn ng l p trình, nhi u s ki n khác nhau.