

TRƯỜNG CAO ĐẲNG NGHỀ TỈNH BÀ RỊA VŨNG TÀU  
Khoa Công nghệ thông tin



# **GIÁO TRÌNH HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU MS SQL SERVER**

*Giáo viên: Trần Thị Thùy Dung*

## BÀI 1

### TỔNG QUAN VỀ MÔ HÌNH CLIENT/SERVER

**Thời lượng:** 5 giờ (3 giờ Lý thuyết, 2 giờ Thực hành)

#### Mục tiêu bài học

- Hiểu rõ mô hình CSDL Client/Server.
- Nắm vững các đặc trưng của mô hình Client/Server.
- Nghiêm túc, tự giác trong học tập.

#### Nội dung chính

- Các kiến thức tổng quan về CSDL.
- Các giai đoạn phát triển của một hệ quản trị CSDL.
- Giới thiệu về mô hình Client server và các hệ quản trị CSDL phục vụ cho mô hình Client/Server.
- Các đặc trưng của mô hình Client/server

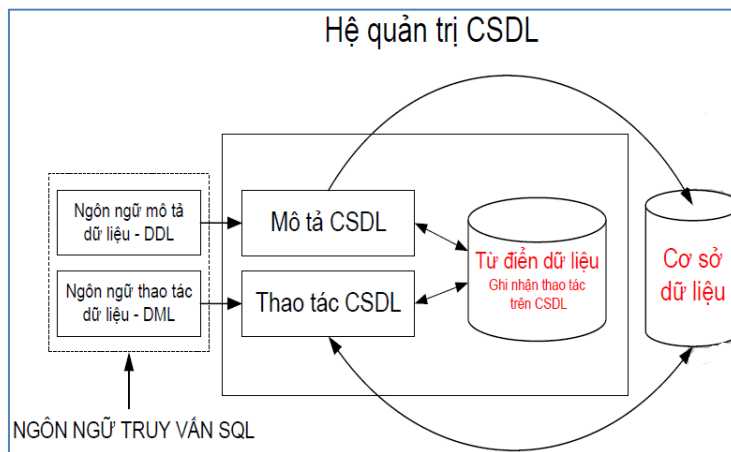
#### Nội dung chi tiết

##### 1. Các kiến thức tổng quan về CSDL

Một hệ CSDL (DB system) bao gồm một **CSDL** (Database) và một **hệ quản trị CSDL** (DBMS)

CSDL (CSDL) là một tập hợp dữ liệu được tổ chức và lưu trữ theo một cấu trúc chặt chẽ nhằm phục vụ *nhiều* đối tượng sử dụng với nhiều mục đích khác nhau.

Hệ quản trị CSDL (Database Management System - DBMS) là một *công cụ phần mềm* tổng quát *nhằm hỗ trợ việc lưu trữ, truy xuất và quản trị CSDL*. Nó cung cấp cho người dùng và ứng dụng một môi trường thuận tiện và sử dụng hiệu quả tài nguyên dữ liệu. Nói cách khác, Hệ quản trị CSDL là *phần mềm chuyên dụng* để giải quyết tốt các tình huống mà CSDL đặt ra như: bảo mật, cạnh tranh trong truy xuất.



*Hình 1: Kiến trúc của một hệ quản trị CSDL*

## **2. Các giai đoạn phát triển của một hệ quản trị CSDL.**

Những năm 1975-1976, IBM lần đầu tiên đưa ra hệ quản trị CSDL kiểu quan hệ mang tên SYSTEM-R với ngôn ngữ giao tiếp CSDL là SEQUEL (Structured English **Q**Uery **L**anguage), đó là một ngôn ngữ con để thao tác với CSDL.

Năm 1976 ngôn ngữ SEQUEL được cải tiến thành SEQUEL2. Khoảng năm 1978-1979 SEQUEL2 được cải tiến và đổi tên thành **Ngôn ngữ truy vấn có cấu trúc (Structured Query Language - SQL)** và cuối năm 1979 được cải tiến thành SYSTEM-R.

Năm 1986 Viện Tiêu Chuẩn Quốc Gia Mỹ (American National Standards Institute - ANSI) đã công nhận và chuẩn hóa ngôn ngữ SQL, và sau đó Tổ chức Tiêu chuẩn Thế giới (International Standards Organization - ISO) cũng đã công nhận ngôn ngữ này. Đó là chuẩn SQL-86.

Tới nay SQL đã qua 3 lần chuẩn hóa lại (1989, 1992, 1996) để mở rộng các phép toán và tăng cường khả năng bảo mật và tính toàn vẹn dữ liệu.

Ngôn ngữ CSDL được cài đặt khác nhau đối tùy theo các hệ quản trị CSDL khác nhau, tuy nhiên đều phải theo một chuẩn (Standard) nhất định. Hiện nay, đa phần các ngôn ngữ truy vấn CSDL dựa trên chuẩn SQL-92.

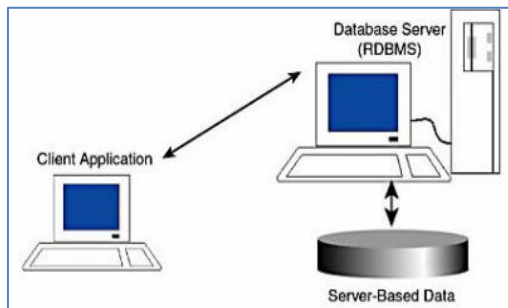
## **3. Giới thiệu về mô hình Client/Server và các hệ quản trị CSDL phục vụ cho mô hình Client/Server.**

SQL là một hệ quản trị CSDL nhiều người dùng kiểu Client/Server. Đây là hệ thống cơ bản dùng lưu trữ dữ liệu cho hầu hết các ứng dụng lớn hiện nay. Mô hình Client/Server trên SQL được sử dụng để điều khiển tất cả các chức năng mà một hệ quản trị CSDL cung cấp cho người dùng các khả năng

- Định nghĩa dữ liệu
- Truy xuất và thao tác dữ liệu
- Điều khiển truy cập
- Đảm bảo toàn vẹn dữ liệu

## **4. Các đặc trưng của mô hình Client/server**

Một ứng dụng kiểu Client/Server bao gồm 2 phần: Một phần chạy trên Server (máy chủ) và phần khác chạy trên các Workstations (máy trạm).



*Hình 2: Mô hình Client/Server trên SQL Server*

Phần Server: (Máy chủ) chứa các CSDL, cung cấp các chức năng phục vụ cho việc tổ chức và quản lý CSDL, cho phép nhiều người sử dụng cùng truy cập dữ liệu. Điều này không chỉ tiết kiệm mà còn thể hiện tính nhất quán về mặt dữ liệu. Tất cả dữ liệu đều được truy xuất thông qua server, không được truy xuất trực tiếp. Do đó, có độ bảo mật cao, tính năng chịu lỗi, chạy đồng thời, sao lưu dự phòng...

Phần Client (Máy khách): Là các phần mềm chạy trên máy trạm cho phép người sử dụng giao tiếp CSDL trên Server.

Hệ thống máy tính Client/Server có 5 mô hình kiến trúc dựa trên cấu hình phân tán về truy nhập dữ liệu, gồm:

- Mô hình CSDL tập trung (Centralized database model)
- Mô hình CSDL theo kiểu file - server (File - server database model)
- Mô hình xử lý từng phần CSDL (Database extract proceSQL Servering model)
- Mô hình CSDL Client/Server (Client/Server database model)
- Mô hình CSDL phân tán (Distributed database model)

## **5. Bài tập**

Câu 1: Phân biệt các khái niệm CSDL, hệ quản trị CSDL, hệ CSDL, SQL, T-SQL

Câu 2: Cho ví dụ về một số hệ quản trị CSDL theo mô hình Client/Server ?

Câu 3: Hãy trình bày các đặc trưng của mô hình Client/Server?

## **BÀI 2**

### **CẤU HÌNH CSDL CLIENT/SERVER**

**Thời lượng:** 9 giờ (4 giờ Lý thuyết, 4 giờ Thực hành, 1 giờ Kiểm tra)

#### **Mục tiêu bài học**

- Phân tích được các tầng cấu trúc của mô hình Client/Server.
- Phân biệt được các mô hình CSDL.
- Nghiêm túc, tự giác trong học tập.
- Đảm bảo an toàn cho người và thiết bị.

#### **Nội dung chính**

- Tổng quan về cấu trúc Client/Server
- Các tầng cấu trúc.
- Các mô hình dữ liệu của hệ thống Client/Server.

#### **Nội dung chi tiết**

##### **1. Tổng quan về cấu trúc Client/Server**

Một CSDL phải đảm bảo được **tính độc lập** giữa dữ liệu và chương trình ứng dụng. Vì vậy, trong mô hình CSDL Client/Server, hệ thống Server lưu trữ CSDL ở trên máy A, thì hệ thống Client chạy các chương trình ứng dụng phải ở trên máy khác.

Trong mô hình này, các thành phần **xử lý ứng dụng** trên hệ thống Client đưa ra yêu cầu cho **phần mềm CSDL** trên máy **Client**, phần mềm này sẽ **kết nối** với **phần mềm CSDL** chạy trên **Server**. Phần mềm CSDL trên Server sẽ **truy nhập** vào CSDL và gửi trả kết quả cho máy Client.

##### **2. Các tầng cấu trúc**

Theo kiến trúc ANSI-PARC, một CSDL có 3 mức biểu diễn :

###### ***a. Mô hình dữ liệu mức thấp (Mức vật lý hay mức trong)***

Đưa ra các khái niệm mô tả chi tiết về cách thức dữ liệu được lưu trữ trong máy tính nên còn được gọi là mức lưu trữ CSDL.

Tại mức này, vấn đề cần giải quyết là, dữ liệu gì và được lưu trữ như thế nào, lưu ở đâu (đĩa từ, băng từ, track, sector)? Cần các chỉ mục gì? Việc truy xuất là tuần tự (Sequential Access SQL Server) hay ngẫu nhiên (Random Access SQL Server) đối với từng loại dữ liệu.

Những người hiểu và làm việc với CSDL tại mức này là người quản trị CSDL (Administrator), những người sử dụng (NSD) chuyên môn.

Ví dụ : Mô hình quan hệ, mô hình mạng, mô hình phân cấp.

###### ***b. Mô hình dữ liệu mức cao (Mức quan niệm)***

Cung cấp các khái niệm gần gũi với người dùng. Mô hình tự nhiên và giàu ngữ nghĩa.

Tại mức này sẽ giải quyết cho câu hỏi CSDL cần phải lưu giữ bao nhiêu loại dữ liệu? Đó là những dữ liệu gì? Mối quan hệ giữa các loại dữ liệu này như thế nào?

Từ thế giới thực các chuyên viên tin học qua quá trình khảo sát và phân tích, cùng với những người sẽ đảm nhận vai trò quản trị CSDL, sẽ xác định được những loại thông tin gì

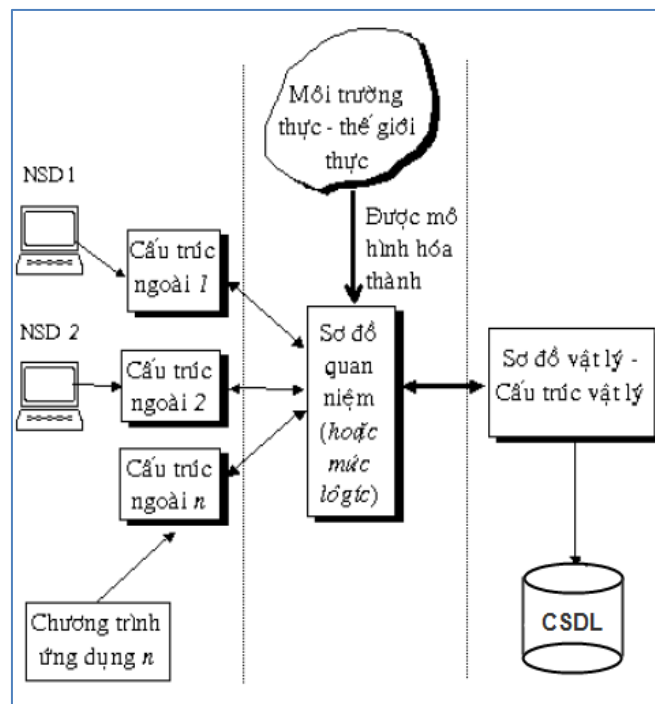
được cho là cần thiết phải đưa vào CSDL, đồng thời mô tả rõ mối liên hệ giữa các thông tin này. Có thể nói cách khác, CSDL mức quan niệm là một sự biểu diễn trừu tượng CSDL mức vật lý; hoặc ngược lại, CSDL vật lý là sự cài đặt cụ thể của CSDL mức quan niệm.

Ví dụ : Mô hình thực thể kết hợp (ERD), mô hình đối tượng.

### c. Mô hình dữ liệu mức thực hiện

Đưa ra các khái niệm người dùng có thể hiểu được nhưng không nằm quá xa với dữ liệu được tổ chức thực sự trên máy. Đây là mức của người sử dụng và các chương trình ứng dụng.

Mỗi người sử dụng hay mỗi chương trình ứng dụng có thể được "nhìn" (View) CSDL theo một góc độ khác nhau. Có thể "nhìn" thấy toàn bộ hay chỉ một phần hoặc chỉ là các thông tin tổng hợp từ CSDL hiện có. Người sử dụng hay chương trình ứng dụng có thể hoàn toàn không được biết về cấu trúc tổ chức lưu trữ thông tin trong CSDL, thậm chí ngay cả tên gọi của các loại dữ liệu hay tên gọi của các thuộc tính. Họ chỉ có thể làm việc trên một phần CSDL theo cách "nhìn" do người quản trị hay chương trình ứng dụng quy định, gọi là khung nhìn (View).



Hình 3 : Kiến trúc tổng quát (ANSI – PARC) của một CSDL.

## 3. Các mô hình dữ liệu của hệ thống Client/Server

### a. Mô hình CSDL tập trung (Centralized database model)

Trong mô hình này, các thành phần xử lý ứng dụng, phần mềm CSDL và bản thân CSDL đều ở trên một bộ xử lý.

Ví dụ Người dùng máy tính cá nhân có thể chạy các chương trình ứng dụng có sử dụng phần mềm CSDL Oracle để truy nhập tới CSDL nằm trên đĩa cứng của máy tính cá nhân đó. Từ khi các thành phần ứng dụng, phần mềm CSDL và bản thân CSDL cùng nằm trên một máy tính thì ứng dụng đã thích hợp với mô hình tập trung.

Hầu hết công việc xử lý luồng thông tin chính được thực hiện bởi nhiều tổ chức mà vẫn phù hợp với mô hình tập trung. Ví dụ một bộ xử lý mainframe chạy phần mềm CSDL IMS hoặc DB2 của IBM có thể cung cấp cho các trạm làm việc ở các vị trí phân tán sự truy nhập nhanh chóng tới CSDL trung tâm. Tuy nhiên trong rất nhiều hệ thống như vậy, cả 3 thành phần của ứng dụng CSDL đều thực hiện trên cùng một máy mainframe do vậy cấu hình này cũng thích hợp với mô hình tập trung.

***b. Mô hình CSDL theo kiểu file - server (File - server databasemodel)***

Trong mô hình CSDL theo kiểu file - server các thành phần ứng dụng và phần mềm CSDL ở trên một hệ thống máy tính và các file vật lý tạo nên CSDL nằm trên hệ thống máy tính khác. Một cấu hình như vậy thường được dùng trong môi trường cục bộ, trong đó một hoặc nhiều hệ thống máy tính đóng vai trò của server, lưu trữ các file dữ liệu cho hệ thống máy tính khác thâm nhập tới. Trong môi trường file - server, phần mềm mạng được thi hành và làm cho các phần mềm ứng dụng cũng như phần mềm CSDL chạy trên hệ thống của người dùng cuối coi các file hoặc CSDL trên file server thực sự như là trên máy tính của người chính họ.

Mô hình file server rất giống với mô hình tập trung. Các file CSDL nằm trên máy khác với các thành phần ứng dụng và phần mềm CSDL; tuy nhiên các thành phần ứng dụng và phần mềm CSDL có thể có cùng thiết kế để vận hành một môi trường tập trung.

***c. Mô hình xử lý từng phần CSDL (Database extract proceSQL Servering model)***

Một CSDL ở xa có thể được truy nhập bởi phần mềm CSDL, được gọi là xử lý dữ liệu từng phần. Với mô hình này, người sử dụng có thể tại một máy tính cá nhân kết nối với hệ thống máy tính ở xa nơi có dữ liệu mong muốn. Người sử dụng có thể tác động trực tiếp đến phần mềm chạy trên máy ở xa và tạo yêu cầu để lấy dữ liệu từ CSDL đó. Người sử dụng cũng có thể chuyển dữ liệu từ máy tính ở xa về chính máy tính của mình và vào đĩa cứng và có thể thực hiện việc sao chép bằng phần mềm CSDL trên máy cá nhân.

Với cách tiếp cận này, người sử dụng phải biết chắc chắn là dữ liệu nằm ở đâu và làm như thế nào để truy nhập và lấy dữ liệu từ một máy tính ở xa. Phần mềm ứng dụng đi kèm cần phải có trên cả hai hệ thống máy tính để kiểm soát sự truy nhập dữ liệu và chuyển dữ liệu giữa hai hệ thống. Tuy nhiên, phần mềm CSDL chạy trên hai máy không cần biết rằng việc xử lý CSDL từ xa đang diễn ra vì người sử dụng tác động tới chúng một cách độc lập.

***d. Mô hình CSDL Client/Server (Client/Server database model)***

Mô hình CSDL Client/Server gần giống như mô hình file - server, tuy nhiên mô hình Client/Server có rất nhiều thuận lợi hơn mô hình file - server.

Xét ví dụ sau đây: Một người dùng cuối muốn tạo ra một văn tin để lấy dữ liệu tổng số, yêu cầu đòi hỏi lấy dữ liệu từ 1000 bản ghi.

Với cách tiếp cận File-Server nội dung của tất cả 1000 bản ghi phải đưa lên mạng, vì phần mềm CSDL chạy trên máy của người sử dụng phải truy nhập từng bản ghi để thỏa mãn yêu cầu của người sử dụng.

Với cách tiếp cận CSDL Client/Server, chỉ có lời văn tin khởi động ban đầu và kết quả



cuối cùng cần đưa lên mạng. Phần mềm CSDL chạy trên máy lưu giữ CSDL sẽ truy nhập các bản ghi cần thiết, xử lý chúng và gọi các thủ tục cần thiết để đưa ra kết quả cuối cùng.

❖ *Front-end software*

Trong mô hình CSDL Client/Server, thường nói đến các phần mềm front-end software và back-end software. Front-end software được chạy trên một máy tính cá nhân hoặc một workstation và đáp ứng các yêu cầu đơn lẻ riêng biệt, phần mềm này đóng vai trò của Client trong ứng dụng CSDL Client/Server và thực hiện các chức năng hướng tới nhu cầu của người dùng cuối cùng, phần mềm Front-end software thường được chia thành các loại sau:

- End user database software: Được thực hiện bởi người sử dụng cuối trên chính hệ thống của họ để truy nhập các CSDL cục bộ nhỏ cũng như kết nối với các CSDL lớn hơn trên CSDL Server.
- Simple query and reporting software: Được thiết kế để cung cấp các công cụ dễ dùng hơn trong việc lấy dữ liệu từ CSDL và tạo các báo cáo đơn giản từ dữ liệu đã có.
- Data analysis software: Cung cấp các hàm về tìm kiếm, khôi phục, chúng có thể cung cấp các phân tích phức tạp cho người dùng.
- Application development tools: Cung cấp các khả năng về ngôn ngữ mà các nhân viên hệ thống thông tin chuyên nghiệp sử dụng để xây dựng các ứng dụng CSDL.
- Database administration Tools: Các công cụ này cho phép người quản trị CSDL sử dụng máy tính cá nhân hoặc trạm làm việc để thực hiện việc quản trị CSDL như định nghĩa các CSDL, thực hiện lưu trữ hay phục hồi.

❖ *Back-end software*

Phần mềm này bao gồm phần mềm CSDL Client/Server và phần mềm mạng chạy trên máy đóng vai trò là Server CSDL.

***e. Mô hình CSDL phân tán (Distributed database model)***

Cả hai mô hình File - Server và Client/Server đều giả định là dữ liệu nằm trên một bộ xử lý và chương trình ứng dụng truy nhập dữ liệu nằm trên một bộ xử lý khác, còn mô hình CSDL phân tán lại giả định bản thân CSDL có ở trên nhiều máy khác nhau.

**4. Bài tập**

Câu 1: Hãy so sánh mô hình dữ liệu tập trung và mô hình dữ liệu phân tán ?

Câu 2: Cho ví dụ về mô hình dữ liệu tập trung và mô hình dữ liệu phân tán hiện nay?



## **BÀI 3**

### **HỆ QUẢN TRỊ CSDL MS SQL SERVER**

**Thời lượng:** 10 giờ (5 giờ Lý thuyết, 5 giờ Thực hành)

#### **Mục tiêu bài học**

- Trình bày lịch sử phát triển của hệ quản trị CSDL MS SQL Server
- Cài đặt được phần mềm hệ quản trị CSDL MS SQL Server
- Sử dụng được các công cụ hỗ trợ của hệ quản trị CSDL MS SQL Server
- Cấu hình được hệ thống quản trị CSDL trên Server nội bộ.
- Nghiêm túc, tự giác trong học tập.
- Đảm bảo an toàn cho người và thiết bị.

#### **Nội dung chính**

- Giới thiệu hệ quản trị CSDL MS SQL Server.
- Cài đặt MS SQL Server
- Các công cụ của MS SQL Server
- Làm việc với công cụ Enterprise Manager

#### **Nội dung chi tiết**

##### **1. Giới thiệu hệ quản trị CSDL MS SQL Server**

SQL viết tắt của Structured Query Language (ngôn ngữ truy vấn có cấu trúc), là công cụ sử dụng để tổ chức, quản lý và truy xuất dữ liệu được lưu trữ trong các CSDL. SQL là một hệ thống ngôn ngữ bao gồm tập các câu lệnh sử dụng để tương tác với CSDL quan hệ.

Ngôn ngữ truy vấn có cấu trúc và các hệ quản trị CSDL quan hệ là một trong những nền tảng kỹ thuật quan trọng trong công nghiệp máy tính. Hiện nay SQL được xem là ngôn ngữ chuẩn trong CSDL. Các hệ quản trị CSDL quan hệ thương mại hiện có như Oracle, SQL Server, Informix, DB2,... đều chọn SQL làm ngôn ngữ cho sản phẩm của mình.

SQL là một hệ quản trị CSDL nhiều người dùng kiểu Client/Server. Đây là hệ thống cơ bản dùng lưu trữ dữ liệu cho hầu hết các ứng dụng lớn hiện nay. Mô hình Client/Server trên SQL được sử dụng để điều khiển tất cả các chức năng mà một hệ quản trị CSDL cung cấp cho người dùng các khả năng:

- **Định nghĩa dữ liệu:** SQL cung cấp khả năng định nghĩa các CSDL, các cấu trúc lưu trữ và tổ chức dữ liệu cũng như mối quan hệ giữa các thành phần dữ liệu.
- **Truy xuất và thao tác dữ liệu:** Người dùng có thể dễ dàng thực hiện các thao tác truy xuất, bổ sung, cập nhật và loại bỏ dữ liệu trong các CSDL.
- **Điều khiển truy cập:** SQL có thể được sử dụng để cấp phát và kiểm soát các thao tác của người sử dụng trên dữ liệu, đảm bảo sự an toàn cho CSDL
- **Đảm bảo toàn vẹn dữ liệu:** SQL định nghĩa các ràng buộc toàn vẹn trong CSDL nhờ đó đảm bảo tính hợp lệ và chính xác của dữ liệu trước các thao tác cập nhật cũng như các lỗi của hệ thống.

SQL Server sử dụng ngôn ngữ lập trình và truy vấn CSDL **Transact-SQL (T-SQL)**, một phiên bản của Structured Query Language. Ngôn ngữ lập trình và truy vấn T-SQL cho phép truy xuất dữ liệu, cập nhật và quản lý hệ thống CSDL quan hệ. Mỗi máy chủ chỉ có một hệ quản trị CSDL SQL Server.

## **2. Cài đặt MS SQL Server**

SQL Server 2005 có nhiều phiên bản khác nhau, trong đó bản ExpressSQL Server là bản thấp nhất, được Microsoft cung cấp miễn phí cho người dùng với mục đích học tập và ứng dụng vào những ứng dụng nhỏ, không yêu cầu cao về các tính năng khác ngoài việc lưu trữ và xử lý đơn giản.

### ***a. Yêu cầu về phần cứng và hệ điều hành sử dụng***

• **Hệ điều hành tối thiểu:** Windows 2000 Service Pack 4; Windows Server 2003 Service Pack1; Windows XP Service Pack 2

• **Phần cứng:**

- Máy tính chip Pentium III 600 MHz trở lên (Cấu hình đề nghị: Chip 1 GHz hoặc cao hơn.)
- Tối thiểu 192 MB RAM (Cấu hình đề nghị: 512 MB RAM.)
- Ổ cứng còn trống tối thiểu 525 MB

• **Bộ cài đặt:**

Để cài đặt SQL Server 2005 ExpressSQL Server, máy tính phải có Windows Installer 3.1 trở lên, download về tại địa chỉ:

<http://go.microsoft.com/fwlink/?LinkId=46459>

Microsoft .Net Framework 2.0

○ Hệ điều hành 32bit: <http://go.microsoft.com/fwlink/?LinkId=55731>

○ Hệ điều hành 64bit: <http://go.microsoft.com/fwlink/?LinkId=55734>

File cài đặt SQL Server 2005 ExpressSQL Server

<http://www.microsoft.com/downloads/details.aspx?familyid=220549b5-0b07-4448-8848-dcc397514b41&displaylang=en>

SQL Server Management Studio ExpressSQL Server

<http://www.microsoft.com/downloads/details.aspx?familyid=C243A5AE-4BD1-4E3D-94B8-5A0F62BF7796&displaylang=en>

Sau khi download về, lưu vào một thư mục nào đó để bắt đầu tiến hành cài đặt (VD: D:\SQL)

Trong hướng dẫn dưới đây, các thành phần trên có tên file cài đặt lần lượt là: *WindowsInstaller-KB893803-v2-x86.exe*: Windows Installer 3.1; *dotnetfx.exe*: Microsoft .Net Framework 2.0; *SQLEXPR.EXE*: SQL Server 2005 ExpressSQL Server; *SQLServer2005\_SQL\_SERVERMSEE.msi*: Công cụ quản lý SQL Server Management Studio ExpressSQL Server

## ***b. Tiến hành cài đặt MS SQL Server***

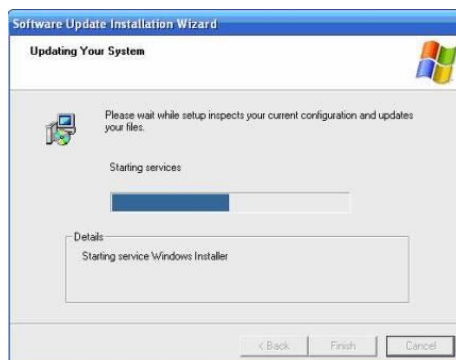
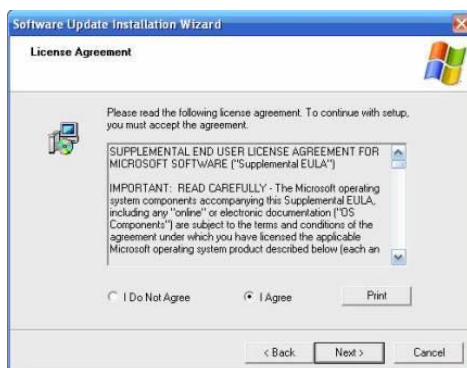
### **Bước 1: Cài Windows Installer 3.1**

**Nhấn vào file WindowsInstaller-KB893803-v2-x86.exe**

Nhấn Next



Chọn I agree, nhấn Next

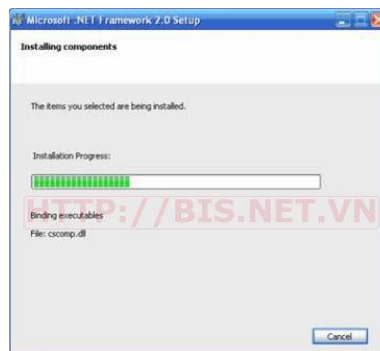
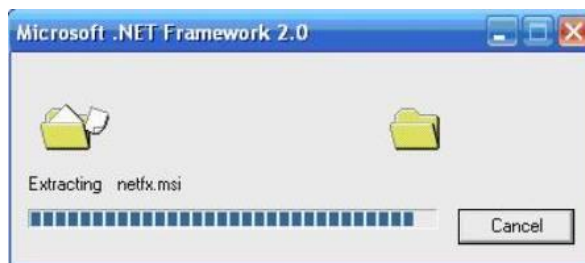


Click Finish

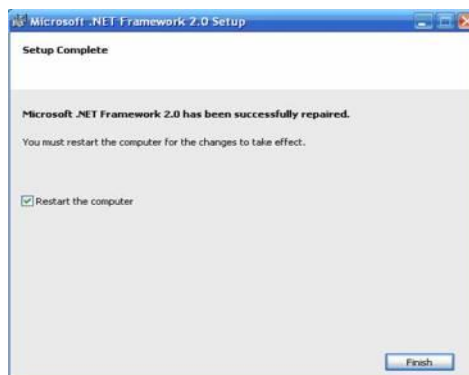


## **Bước 2: Cài Framework 2.0**

**File cài: dotnetfx.exe**



**Nhấn Finish**

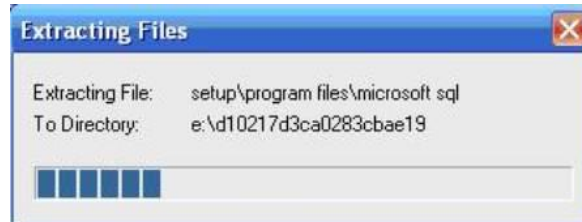


**Máy sẽ khởi động lại**

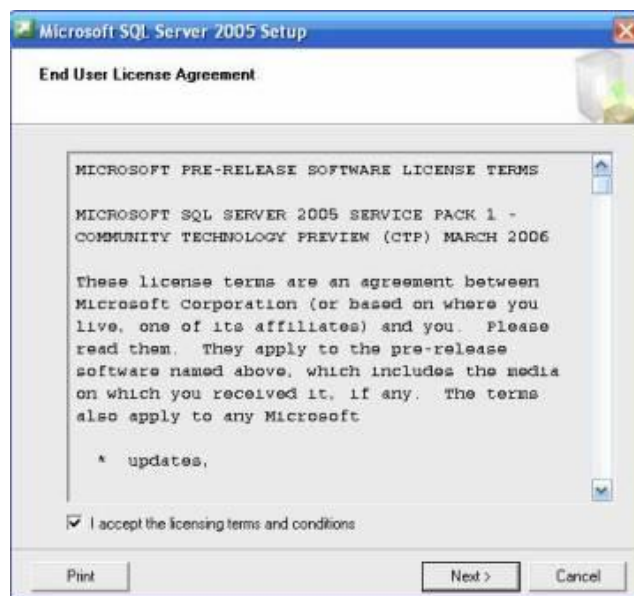
Sau đó tiếp tục bước 3

### **Bước 3: Cài Microsoft SQL Server 2005**

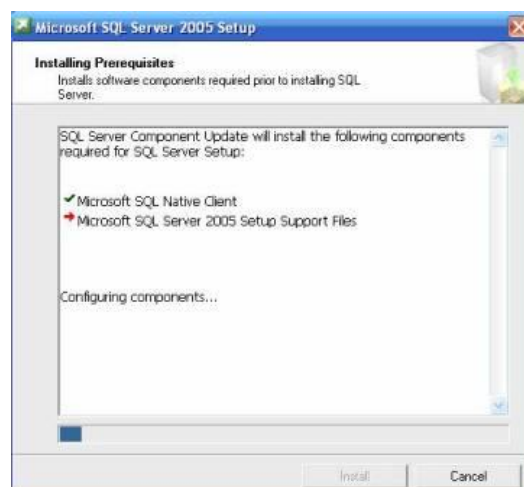
**Nhấn vào file SQLEXPRESS.EXE**



Trên màn hình tiếp theo, đánh dấu vào mục: "I accept the licensing terms and conditions", nhấn Next



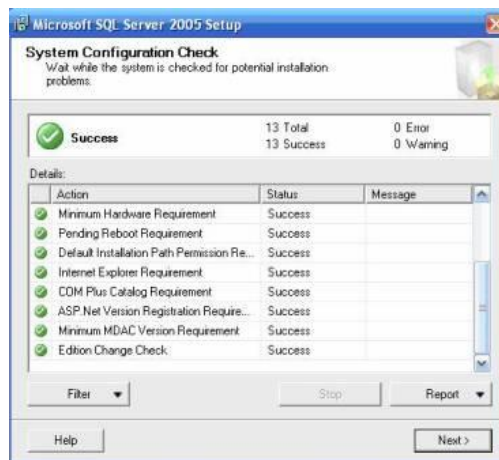
Trên màn hình tiếp theo, nhấn Install



Nhấn Next



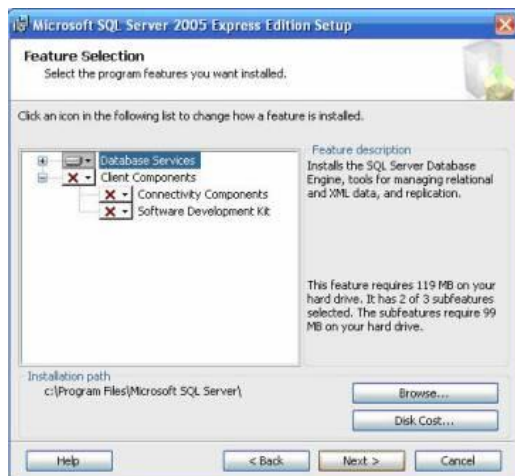
Tiếp tục nhấn Next



Trên màn hình tiếp theo, bỏ dấu check mở mục "Hide advanced configuration options", nhấn Next.



Nhấn Next trên màn hình tiếp theo



Khai báo Instance name, chọn Default Instance (Server Name sẽ trùng với tên máy)

Nhấn Next



Nhấn Next trên cửa sổ tiếp theo





Trên cửa sổ tiếp theo, chọn chế độ chứng thực user

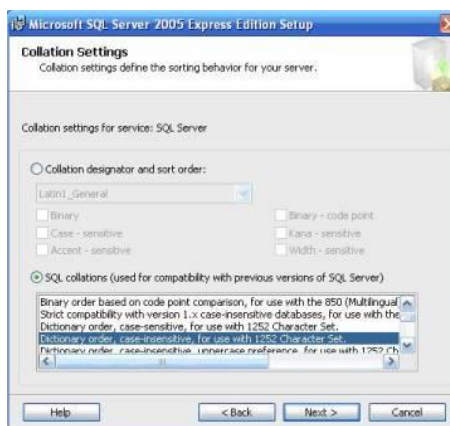
- Windows Authentication Mode: Sử dụng chứng thực của Windows
- Mixed Mode: Kết hợp cả chứng thực của Windows và chứng thực của SQL Server

Ở đây có thể để mặc định: "Windows Authentication Mode"

Nhấn Next



Nhấn Next



Nhấn Next



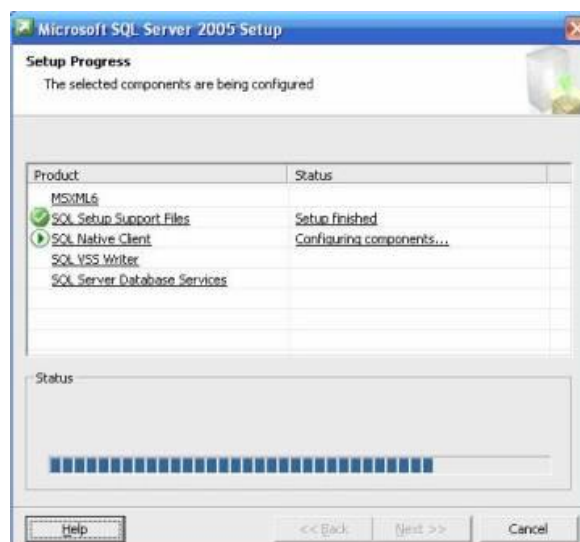
Nhấn Next



Click Install để bắt đầu cài đặt



Quá trình cài đặt có thể mất 5 -> 10 phút



Khi màn hình báo cài đặt thành công như hình dưới, click Next



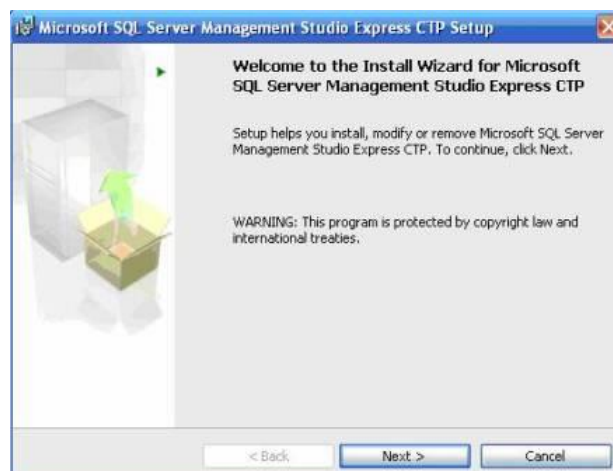
Tiếp tục click Finish



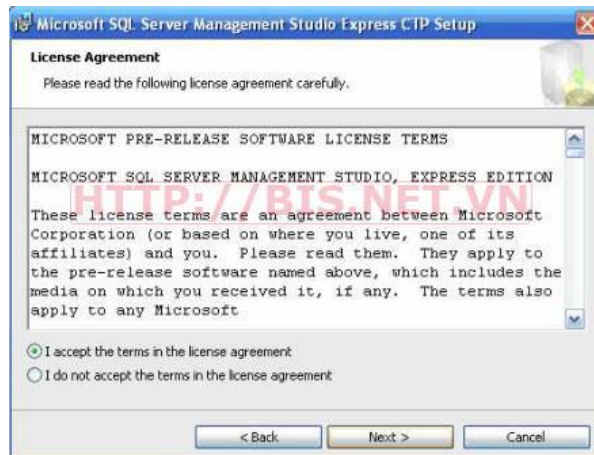
#### Bước 4: Cài Microsoft SQL Server Management Studio Express SQL Server:

File cài đặt SQLServer2005\_SQL SERVERMSEE.msi

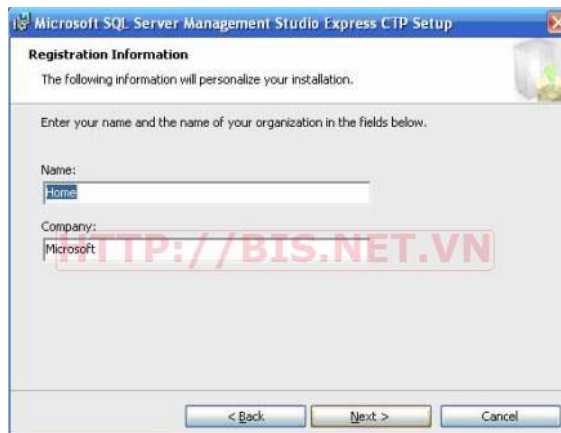
Nhấn Next



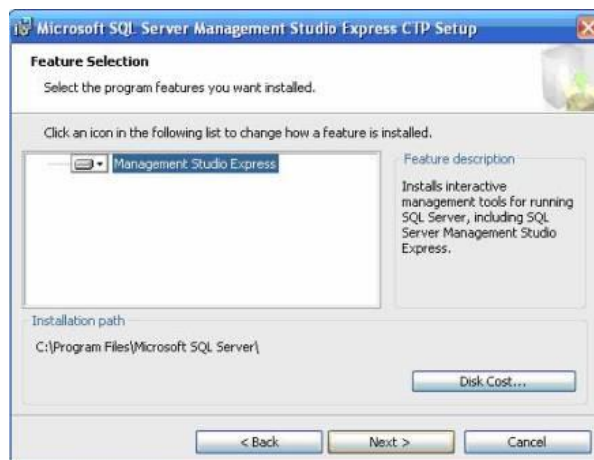
Click chọn "I accept the terms in the license agreement", nhấn Next



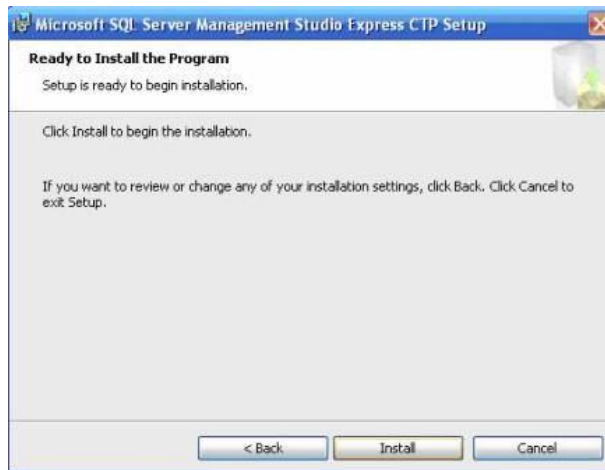
Gõ tên người dùng, đơn vị sử dụng, sau đó nhấn Next



Màn hình tiếp theo, nhấn Next



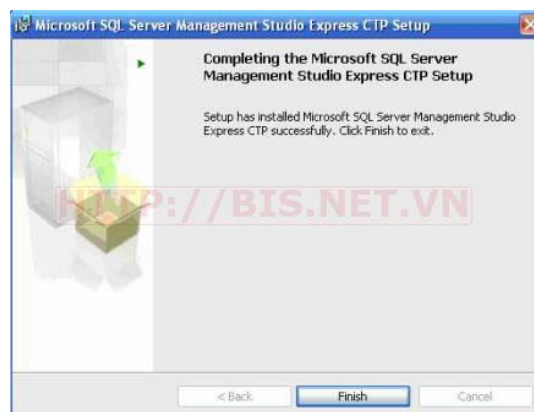
Click Install



Đợi trong vài phút



Nhấn Finish



Đến đây ta đã cài đặt xong SQL Server 2005 ExpressSQL Server và các công cụ để quản lý CSDL.

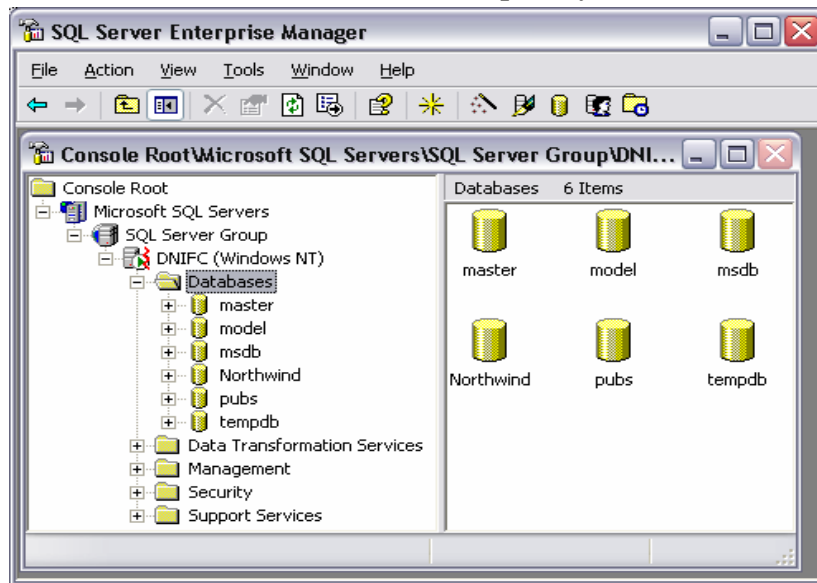
### 3. Các công cụ của MS SQL

#### a. English Query

Đây là một dịch vụ giúp cho việc truy vấn dữ liệu bằng tiếng Anh.

#### b. Trình Enterprise Manager

Đây là môi trường trực quan sử dụng hệ thống thực đơn Enterprise Manager. Sử dụng hệ thống thực đơn (menu) để thực hiện các yêu cầu. Chủ yếu các yêu cầu tạo, thêm, xoá, sửa dữ liệu. Quản lý lịch trình Backup dữ liệu. Quản lý người dùng đang truy cập CSDL. Tạo, xóa quyền Login User. Định cấu hình cho Server. Tạo và quản lý tìm kiếm.



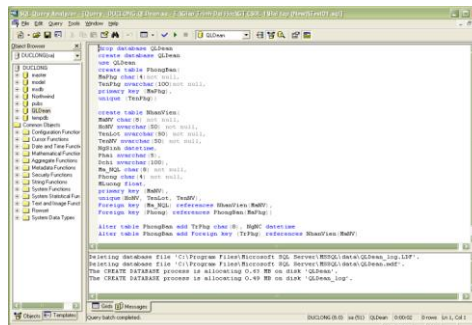
Hình 6: Giao diện SQL Server Enterprise Manager

#### c. Công cụ lập trình - Query Analyzer (ISQL):

Là giao diện chính để chạy các truy vấn Transact-SQL hoặc thủ tục lưu trữ.

Query Analyzer cho phép thực hiện 32 kết nối riêng rẽ cùng một lúc. Mỗi kết nối có một thanh tiêu đề nhận dạng các yếu tố sau:

- Máy tính được đăng nhập
- CSDL đang sử dụng
- Thông tin đăng nhập
- Tên File truy vấn đang mở
- Số của cửa sổ được hiển thị



Hình 7: Giao diện Query Analyzer.

#### d. Tiện ích mạng Client / Server Network

Cung cấp các thư viện nghi thức kết nối mạng (Network-Libraries) cho phép các máy trạm có thể truy cập CSDL trên máy Server: *Named Pipes; TCP/IP; Multiprotocol; NW Link IPX/SPX*

#### e. Books Online

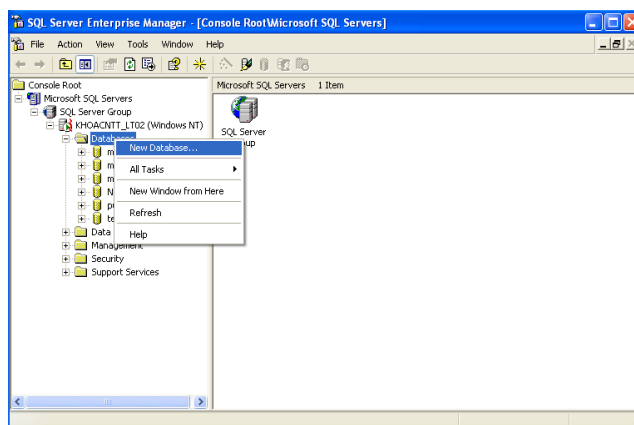
Sách hướng dẫn trực tuyến được lưu dưới dạng HTML đã được biên dịch, nên có thể xem chúng bằng các trình duyệt Web.

### 4. Làm việc với công cụ Enterprise Manager

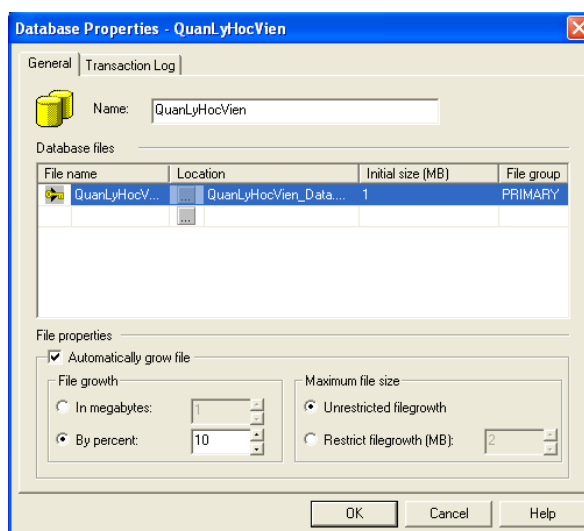
Đây là một công cụ cho ta thấy toàn cảnh hệ thống CSDL một cách rất trực quan. Nó rất hữu ích đặc biệt cho người mới học và chưa thông thạo SQL.

#### a. Tạo CSDL trong SQL Server

Vào menu Start/Programs/Microsoft SQL Server/Enterprise Manager, cửa sổ sau sẽ xuất hiện:



Nhấn chuột phải vào mục Databases /New database..., một cửa sổ sẽ hiện ra yêu cầu nhập tên CSDL



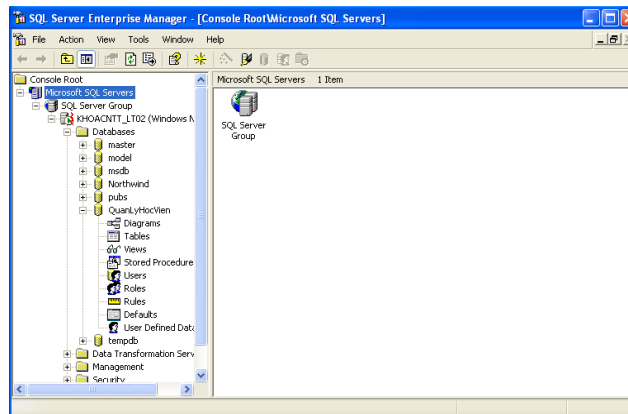
Nhập tên CSDL vào hộp Name (ví dụ QuanLyHocVien)

Vào tùy chọn **In megabytes** thiết lập kích thước lưu trữ giới hạn và dung lượng lưu trữ.

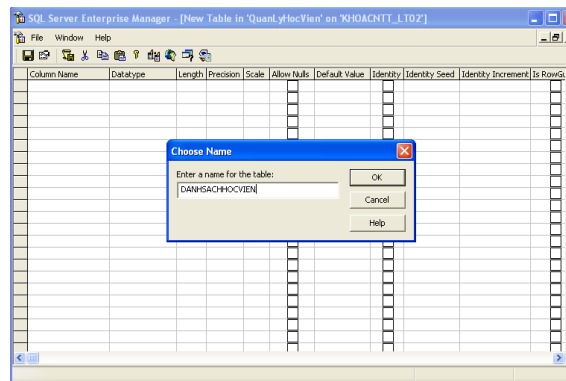


### **b. Tạo bảng trong CSDL**

Để tạo bảng cho một CSDL ta nhấp chuột vào dấu cộng (+) bên trái CSDL tương ứng, ta có danh sách các thành phần của CSDL



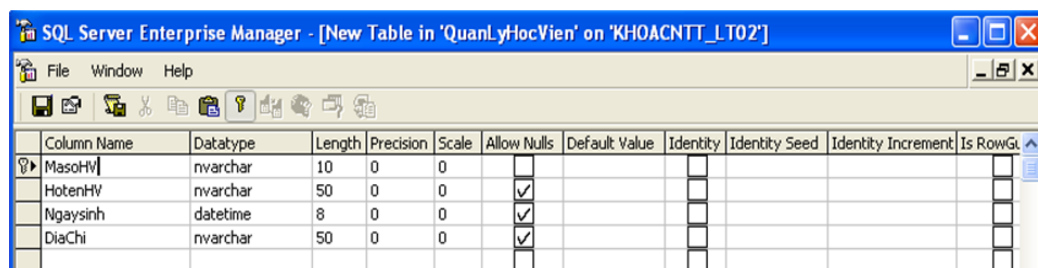
Nhấp chuột phải vào thành phần Tables (bảng)/ New Table.../ để đặt tên bảng.



Sau khi nhập tên bảng nhấn OK để xác nhận, nếu muốn bỏ qua thao tác tạo bảng nhấn nút Cancel.

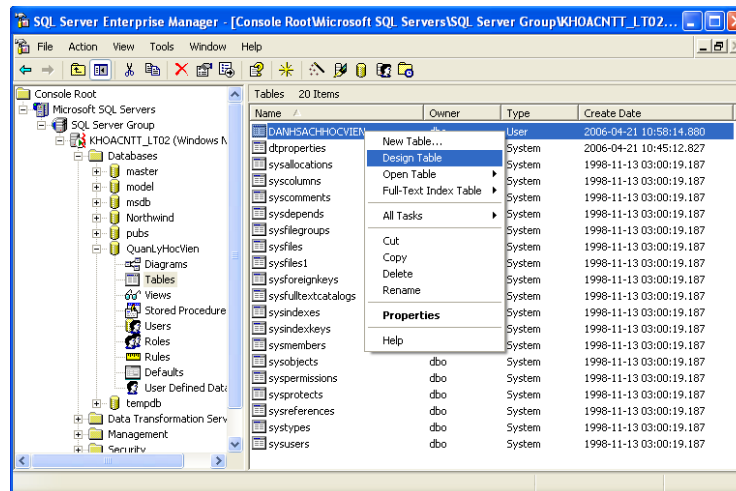
Sau khi tạo bảng mới, ta thiết kế bảng bằng cách nhập vào tên trường vào cột Column Name, chọn kiểu dữ liệu trong cột Datatype, đặt kích thước dữ liệu của trường trong cột Length, bỏ chọn Allow Nulls nếu muốn trường tương ứng không được để trống, chọn Identity nếu muốn tạo chỉ mục.

Muốn tạo khóa chính cho trường nào thì chọn dòng tương ứng sau đó nhấp vào biểu tượng chìa khóa trên thanh công cụ. Có thể kết hợp với phím Shift và Ctrl nếu muốn chọn nhiều dòng.



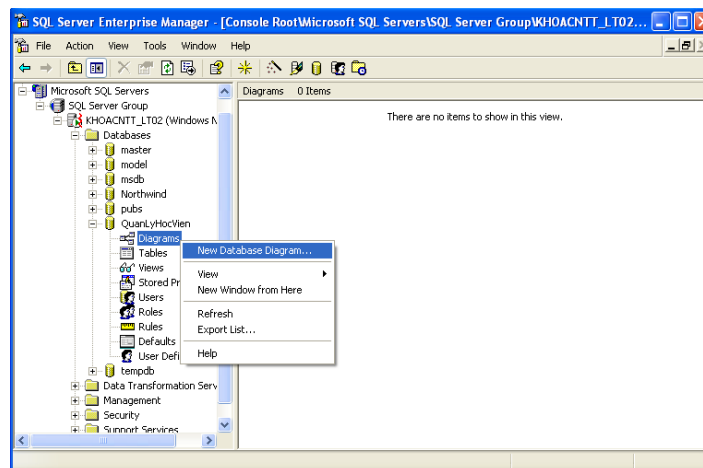
Sau khi thiết kế bảng xong, ta nhấp chuột vào biểu tượng đĩa mềm trên thanh công cụ để lưu bảng và nhấp vào dấu (X) bên dưới của cửa sổ Enterprise Manager để đóng bảng lại.

Để sửa đổi cấu trúc bảng đã tạo chúng ta nhấp chuột phải vào bảng tương ứng chọn Design Table

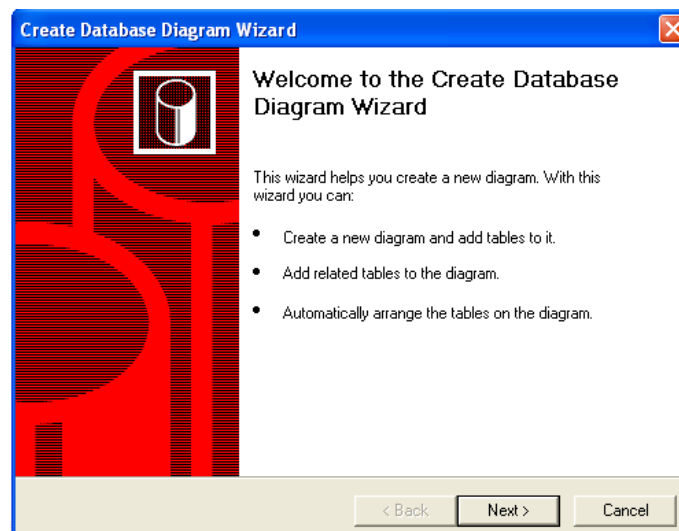


### ***c. Tạo mối quan hệ cho các bảng trong CSDL***

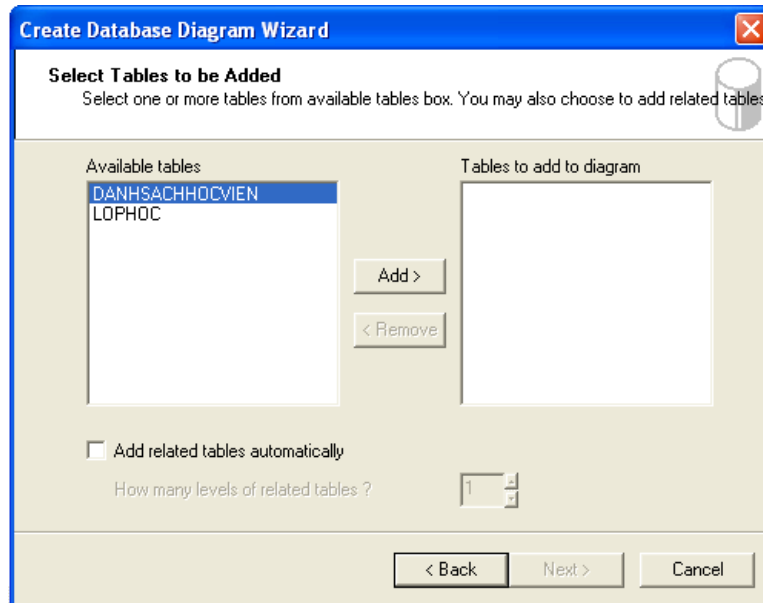
Để tạo quan hệ cho một CSDL trong SQL Server, chúng ta nhấp chuột phải vào thành phần Diagrams của CSDL tương ứng chọn New Database Diagram...



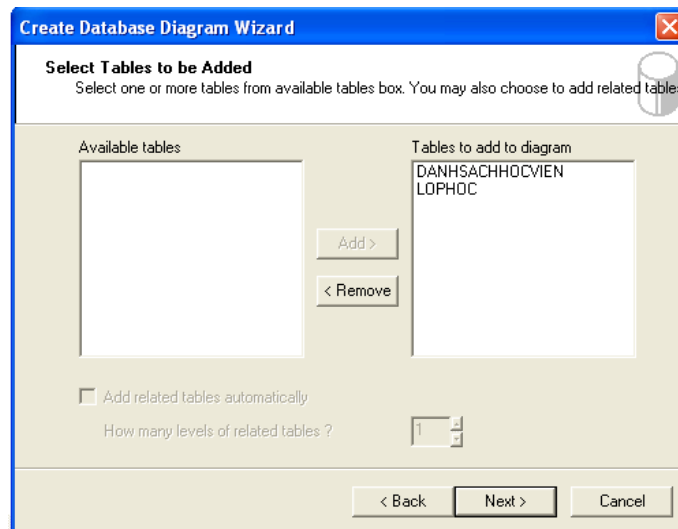
Sau đó sẽ hiển thị một cửa sổ mới



Nhấn nút Next để tiếp tục, một cửa sổ sẽ hiện ra:



Chọn bảng cần thiết lập quan hệ và nhấn nút Add để đưa các bảng này sang cửa sổ bên phải, chọn vào tùy chọn *Add related tables automatically* nếu muốn SQL tự động thiết lập quan hệ.



Bấm Next để tiếp tục và Finish để hoàn thành việc tạo mối quan hệ.

#### d. Nhập dữ liệu cho bảng

Trong cửa sổ Enterprise Manager, chọn CSDL rồi chọn Tables.

Nhấn chuột phải lên bảng cần nhập dữ liệu, chọn Open Table

- Chọn Return all rows nếu muốn nhập dữ liệu với hiện trạng xem tất cả các dòng
- Chọn Return Top nếu muốn xem một số dòng đầu tiên
- Chọn Query nếu muốn định nghĩa dữ liệu theo ý của người dùng.

#### 5. Nhận xét

Có thể nói rằng: SQL là một ngôn ngữ hoàn thiện được sử dụng trong các hệ thống CSDL và là một thành phần không thể thiếu trong các hệ quản trị CSDL. SQL *không phải là một*

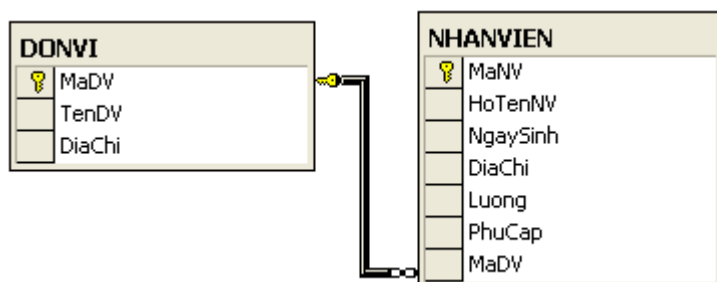
**ngôn ngữ lập trình** nhưng các câu lệnh mà SQL cung cấp có thể được nhúng vào trong các ngôn ngữ lập trình nhằm xây dựng các ứng dụng tương tác với CSDL.

SQL là ngôn ngữ có tính khai báo. Người dùng chỉ cần mô tả các yêu cầu cần phải thực hiện trên CSDL mà không cần phải chỉ ra cách thức thực hiện các yêu cầu như các ngôn ngữ lập trình.

## 6. Bài tập

**Bài 1: Sử dụng Enterprise Manager của MS SQL thực hiện các yêu cầu sau**

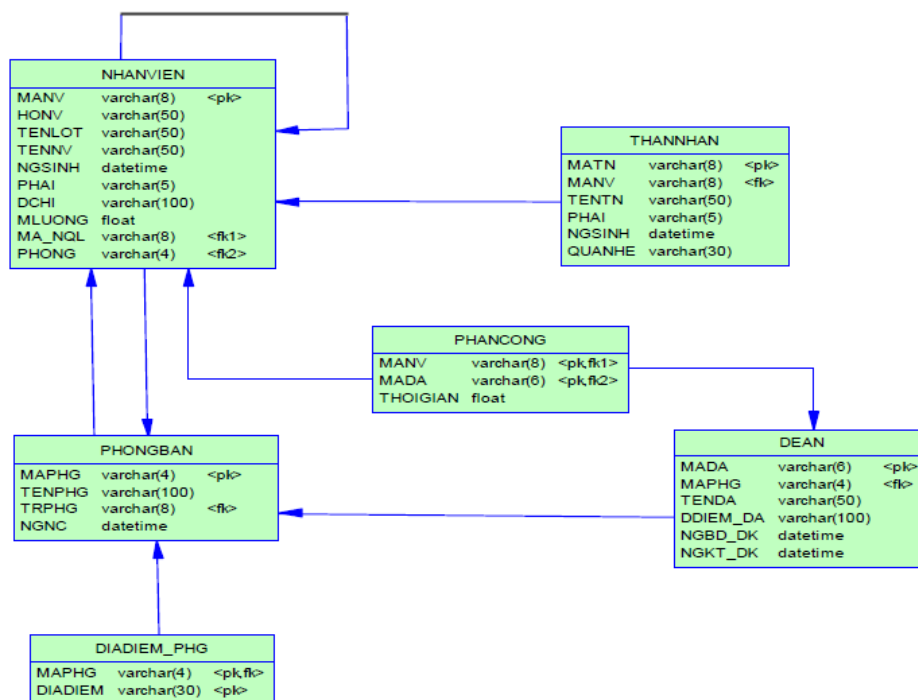
1. Tạo một CSDL tên QuanLyNhanVien.
2. Tạo và thiết lập quan hệ cho hai bảng như sơ đồ sau:



3. Nhập dữ liệu cho hai bảng, mỗi bảng ít nhất 10 bản ghi.

**Bài 2: Sử dụng Enterprise Manager của MS SQL thực hiện các yêu cầu sau**

1. Tạo CSDL có tên QLDeAn
2. Xây dựng các bảng và mối quan hệ dựa trên mô hình dữ liệu sau



## **BÀI 4**

### **CÁC THAO TÁC TRÊN MS SQL SERVER**

**Thời lượng:** 36 giờ (12 giờ Lý thuyết, 23 giờ Thực hành, 1 giờ Kiểm tra)

#### **Mục tiêu bài học**

- Sử dụng được phần mềm quản trị CSDL MS SQL Server
- Thiết kế được các bảng dữ liệu với các kiểu dữ liệu phù hợp
- Phân biệt các mối quan hệ 1-1, 1-n, n-n và thiết lập được quan hệ cho các bảng.
- Thực hiện được các truy vấn Select, Insert, Update và Delete của ngôn ngữ SQL Server trong công cụ Query Analyzer.
- Thực hiện các thao tác an toàn với máy tính

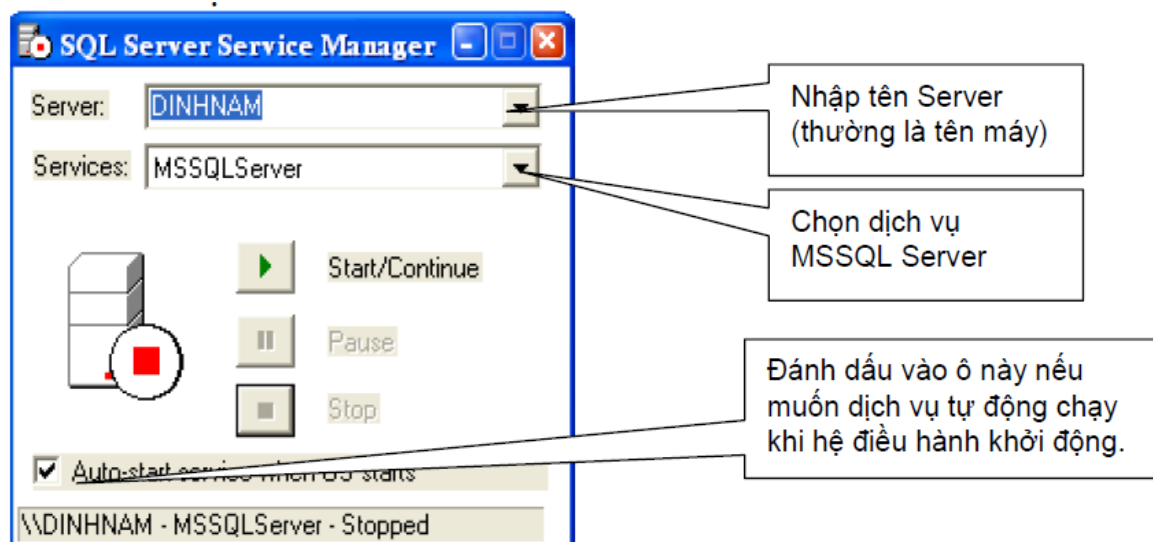
#### **Nội dung chính**

- Đăng nhập vào MS SQL Server.
- Các kiểu dữ liệu trong MS SQL Server
- Tạo CSDL trong MS SQL Server
- Tạo bảng trong MS SQL Server
- Tạo quan hệ trong MS SQL Server
- Nhập dữ liệu trong MS SQL Server
- Truy vấn SQL Server

#### **Nội dung chi tiết**

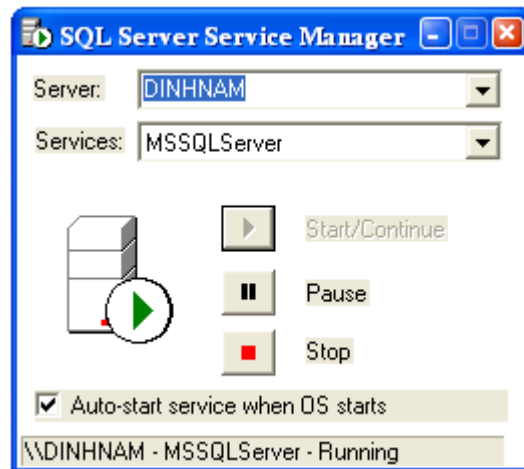
##### **1. Đăng nhập vào MS SQL Server**

Sau khi cài đặt thành công, việc đăng nhập sẽ được thực hiện qua các bước như sau:  
Vào Start/Programs/ Microsoft SQL Server/Service Manager cửa sổ sau sẽ xuất hiện:



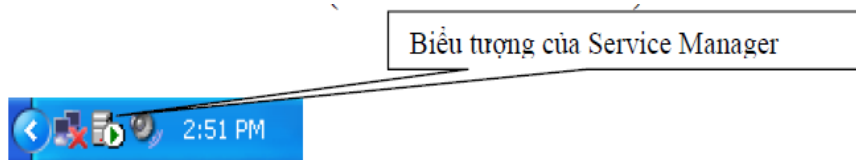
*Hình 8: Đăng nhập SQL Server*

Nhấn vào nút Start/Continue để bắt đầu chạy dịch vụ của SQL Server. Nếu thành công, cửa sổ sau sẽ xuất hiện:



Hình 9: Chạy dịch vụ của SQL Server

- Sau đó chúng ta có thể đóng cửa sổ này lại, một biểu tượng sẽ xuất hiện ở góc phải dưới của màn hình (trên thanh toolbars).



## 2. Kiểu dữ liệu trong MS SQL

Tên kiểu	Mô tả
CHAR (n)	Kiểu chuỗi với độ dài cố định
NCHAR (n)	Kiểu chuỗi với độ dài cố định hỗ trợ UNICODE
VARCHAR (n)	Kiểu chuỗi với độ dài chính xác
NVARCHAR (n)	Kiểu chuỗi với độ dài chính xác hỗ trợ UNICODE
INTEGER	Số nguyên có giá trị từ $-2^{31}$ đến $2^{31}-1$
INT	Như kiểu Integer
TINYTINT	Số nguyên có giá trị từ 0 đến 255.
SMALLINT	Số nguyên có giá trị từ $-2^{15}$ đến $2^{15}-1$
BIGINT	Số nguyên có giá trị từ $-2^{63}$ đến $2^{63}-1$
NUMERIC (p,s)	Kiểu số với độ chính xác cố định.
DECIMAL (p,s)	Tương tự kiểu Numeric
FLOAT	Số thực có giá trị từ $-1.79E+308$ đến $1.79E+308$
REAL	Số thực có giá trị từ $-3.40E + 38$ đến $3.40E + 38$

MONEY	Kiểu tiền tệ
BIT	Kiểu bit (có giá trị 0 hoặc 1)
DATETIME	Kiểu ngày giờ (chính xác đến phần trăm của giây)
SMALLDATETIME	Kiểu ngày giờ (chính xác đến phút)
TIMESTAMP	
BINARY	Dữ liệu nhị phân với độ dài cố định (tối đa 8000 bytes)
VARBINARY	Dữ liệu nhị phân với độ dài chính xác (tối đa 8000 bytes)
IMAGE	Dữ liệu nhị phân với độ dài chính xác
TEXT	Dữ liệu kiểu chuỗi với độ dài lớn
NTEXT	Dữ liệu kiểu chuỗi với độ dài lớn và hỗ trợ UNICODE

### 3. Tạo CSDL trong MS SQL Server

Mở giao diện Query Analyzer. Gõ cú pháp lệnh

CREATE DATABASE <tên CSDL>

Sau đó bôi đen, bấm F5 hoặc  để thực thi câu lệnh.

Ví dụ: Create Database QLBHocsinh

CREATE DATABASE QLBanHang

Dùng cú pháp DROP DATABASE <tên CSDL> để xóa CSDL

### 4. Tạo bảng trong MS SQL Server

Mở giao diện Query Analyzer. Gõ cú pháp lệnh

```
CREATE TABLE < Tên quan hệ> (  
<Thuộc tính> <Kiểu> [<Kích thước>] [NOT NULL],....  
PRIMARY KEY (<Khoá chính>)  
[UNIQUE (<Khoá chỉ định>),...]  
[FOREIGN KEY (<Khoá ngoại>) REFERENCES <bảng>,...]  
[CHECK < Điều kiện ràng buộc>, ...]  
[CONSTRAINT <Tên ràng buộc> <kiểu ràng buộc>] )
```

Trong đó

- Not null: thuộc tính không lấy giá trị rỗng. Lưu ý thuộc tính khoá mặc nhiên bị ràng buộc not null.

- Primary key(<khoá chính>): khai báo khoá chính của bảng.

- Unique (<khoá>): khai báo các khoá chỉ định khác (nếu có)

- Foreign key(<khoá ngoại>) ...: khai báo khoá ngoại của bảng

- Check <điều kiện ràng buộc>: mô tả các ràng buộc miền giá trị

Sau đó bôi đen, bấm F5 hoặc  để thực thi câu lệnh.



Lưu ý: Trước khi tạo bảng phải chọn CSDL chứa bảng bằng lệnh  
USE <Tên CSDL>

Ví dụ:

Tạo bảng với lược đồ quan hệ sau: HANGHOA (MaHG, TenHG, DVT)  
Mã hàng hóa là khóa chính. Tất cả các thuộc tính không được rỗng.

Use QLBanHang

**Create Table** Hanghoa

```
(
MaHG varchar(10) Not Null,
TenHG nvarchar(50) Not Null,
DVT varchar(5) Not Null,
Primary key (MaHG)
)
```

Ví dụ 2: Tạo bảng NhanVien có tham chiếu khóa ngoại đến PhongBan

```
CREATE TABLE NhanVien(
    MaNV char(8) not null,
    HoNV nvarchar(50) not null,
    TenLot nvarchar(50) not null,
    TenNV nvarchar(50) not null,
    NgSinh datetime,
    Phai nvarchar(5),
    Dchi nvarchar(100),
    Ma_NQL char(8) not null,
    Phong char(4) not null,
    MLuong real,
    Primary Key (MaNV),
    Foreign Key (Ma_NQL) References NhanVien(MaNV),
    Foreign Key (Phong) References PhongBan(MaPhg)
)
```

QUAN HỆ CON

Khai báo khóa chính

Khai báo khóa ngoại

QUAN HỆ CHA

## 5. Tạo quan hệ giữa các bảng trong CSDL

Ngoài việc tạo quan hệ giữa các bảng trong quá trình tạo bảng, ta có thể thêm các ràng buộc sau khi tạo bảng theo cú pháp sau

```
ALTER TABLE <Tên_bảng> ADD
CONSTRAINT <Ten_RBTV> <RBTV>,
CONSTRAINT <Ten_RBTV> <RBTV>,
...
```

Hoặc xóa bỏ bằng cú pháp

```
ALTER TABLE <Tên_bảng> DROP <Ten_RBTV>
```

Sau đó bôi đen, bấm F5 hoặc  **Execute** để thực thi câu lệnh.

Ví dụ: Tạo quan hệ giữa bảng PhongBan và Nhanvien trong QLDeAn

```
ALTER TABLE PhongBan ADD CONSTRAINT fk_PhongBan
FOREIGN KEY (TrPhg) REFERENCES NhanVien(MaNV)
```

## 6. Nhập dữ liệu trong MS SQL Server

### a. Thêm dữ liệu

```
INSERT INTO <TÊN BẢNG>
(
    <TÊNTHUỘCTÍNH1>,
    <TÊNTHUỘCTÍNH2>,...)
VALUES(<GIÁ TRỊ 1>, <GIÁ TRỊ 2>, ...)
```

Lưu ý : Các giá trị trong VALUES(...) phải tương ứng với các thuộc tính trong tên bảng. Nếu có thuộc tính nào không được khai thì giá trị của bộ mới được thêm vào ứng với thuộc tính đó sẽ được đặt bằng Null

### b. Sửa dữ liệu

```
UPDATE <TÊN BẢNG> SET
<THUỘC TÍNH 1> = <GIÁ TRỊ 1>,
<THUỘC TÍNH 2> = <GIÁ TRỊ 2>,
...
<THUỘC TÍNH n> = <GIÁ TRỊ n>
WHERE <ĐIỀU KIỆN>
```

Ví dụ:

```
UPDATE MONHOC SET
TCLT = 45
WHERE MAMH = 'MH001'
```

### c. Xóa một trường dữ liệu

```
DELETE FROM <TÊN BẢNG>
WHERE <ĐIỀU KIỆN>
```

Lưu ý Trong việc sửa, xóa trường dữ liệu cần phải có điều kiện where để tránh cập nhật dữ liệu sai

## 7. Truy vấn SQL

Là ngôn ngữ rút trích dữ liệu thỏa một số điều kiện nào đó. Được sử dụng để truy xuất dữ liệu từ các dòng và các cột của một hay nhiều bảng, khung nhìn. Câu lệnh này có thể dùng để thực hiện phép chọn, phép chiếu và phép nối. Ngoài ra, câu lệnh này còn cung cấp khả năng thực hiện các thao tác truy vấn và thống kê dữ liệu phức tạp khác

Cú pháp chung

<b>SELECT</b>	<Danh sách các cột được chọn>
<b>FROM</b>	< Danh sách các bảng>
<b>WHERE</b>	<Điều kiện chọn>
<b>GROUP BY</b>	<Danh sách cột cần nhóm>
<b>HAVING</b>	< Điều kiện nhóm>
<b>ORDER BY</b>	< Danh sách cột cần sắp xếp>

**a. Mệnh đề FROM**

Mệnh đề FROM trong câu lệnh SELECT được sử dụng nhằm chỉ định các bảng và khung nhìn cần truy xuất dữ liệu. Sau FROM là danh sách tên của các bảng và khung nhìn tham gia vào truy vấn, tên của các bảng và khung nhìn được phân cách nhau bởi dấu phẩy.

Ví dụ: Câu lệnh dưới đây hiển thị danh sách các khoa trong trường

```
SELECT * FROM khoa
```

**b. Danh sách chọn trong câu lệnh SELECT**

Danh sách chọn trong câu lệnh SELECT được sử dụng để chỉ định các trường, các biểu thức cần hiển thị trong các cột của kết quả truy vấn. Các trường, các biểu thức được chỉ định ngay sau từ khoá SELECT và phân cách nhau bởi dấu phẩy.

*i. Chọn tất cả các cột trong bảng*

Ví dụ: Liệt kê tất cả các cột trong bảng Lop

```
SELECT * FROM lop
```

*ii. Tên cột trong danh sách chọn*

Ví dụ: Liệt kê danh sách malop, tenlop, namnhaphoc, khoa trong bảng Lop

```
SELECT malop,tenlop,namnhaphoc,khoa
```

```
FROM lop
```

Lưu ý: Nếu truy vấn được thực hiện trên nhiều bảng/khung nhìn và trong các bảng/khung nhìn có các trường trùng tên thì tên của những trường này nếu xuất hiện trong danh sách chọn phải được viết dưới dạng: tên\_bảng.tên\_trường

Ví dụ:

```
SELECT malop, tenlop, lop.makhoa, tenkhoa
```

```
FROM lop, khoa
```

```
WHERE lop.malop = khoa.makhoa
```

*iii. Thay đổi tiêu đề các cột*

Ta sử dụng cách viết: tiêu\_đề\_cột = tên\_trường

hoặc: tên\_trường AS tiêu\_đề\_cột hoặc: tên\_trường tiêu\_đề\_cột

Ví dụ: Câu lệnh dưới đây:

```
SELECT 'Mã lớp'= malop,tenlop 'Tên lớp',khoa AS 'Khoá' FROM lop
```

*iv. Sử dụng cấu trúc CASE trong danh sách chọn*

Cấu trúc này có cú pháp như sau:

CASE biểu\_thức

WHEN biểu\_thức\_kiểm\_tra THEN kết\_quả

[ ... ]

[ELSE kết\_quả\_của\_else]

END hoặc: CASE

WHEN điều\_kiện THEN kết\_quả

[ ... ]

[ELSE kết\_quả\_của\_else] END

Ví dụ: Để hiển thị mã, họ tên và giới tính (nam hoặc nữ) của các sinh viên, ta sử dụng câu lệnh

```
SELECT masv,hodem,ten, CASE gioitinh
```

```
WHEN 1 THEN 'Nam' ELSE 'Nữ'
```

```
END AS gioitinh FROM sinhvien hoặc:
```

```
SELECT masv,hodem,ten, CASE
```

```
WHEN gioitinh=1 THEN 'Nam' ELSE 'Nữ'
```

```
END AS gioitinh
```

```
FROM sinhvien
```

Kết quả của hai câu lệnh trên đều như nhau.

v. Hằng và biểu thức trong danh sách chọn

Ví dụ: câu lệnh dưới đây cho biết tên và số tiết của các môn học

```
SELECT tenmonhoc,sodvht*15 AS sotiet
```

```
FROM monhoc
```

Nếu trong danh sách chọn có sự xuất hiện của giá trị hằng thì giá trị này sẽ xuất hiện trong một cột của kết quả truy vấn ở tất cả các dòng

Ví dụ: Câu lệnh

```
SELECT tenmonhoc,'Số tiết: ',sodvht*15 AS sotiet
```

```
FROM monhoc
```

vi. Loại bỏ các dòng dữ liệu trùng nhau trong kết quả truy vấn

Ta chỉ định thêm từ khóa DISTINCT ngay sau từ khoá SELECT.

Ví dụ: Hai câu lệnh dưới đây

```
SELECT khoa FROM lop và:
```

```
SELECT DISTINCT khoa FROM lop
```

vii. Giới hạn số lượng dòng trong kết quả truy vấn

Ta chỉ định thêm mệnh đề TOP ngay trước danh sách chọn của câu lệnhSELECT.

Ví dụ: Câu lệnh dưới đây hiển thị họ tên và ngày sinh của 5 sinh viên đầu tiên trong danh sách:

```
SELECT TOP 5 hodem,ten,ngaysinh
FROM sinhvien
```

Ngoài cách chỉ định cụ số lượng dòng cần hiển thị trong kết quả truy vấn, ta có thể chỉ định số lượng các dòng cần hiển thị theo tỷ lệ phần trăm bằng cách sử dụng thêm từ khoá PERCENT như ở ví dụ dưới đây.

Ví dụ: Câu lệnh dưới đây hiển thị họ tên và ngày sinh của 10% số lượng sinh viên hiện có trong bảng SINHVIEN

```
SELECT TOP 10 PERCENT hodem,ten,ngaysinh
FROM sinhvien
```

### c. *Chỉ định điều kiện truy vấn dữ liệu*

Mệnh đề WHERE trong câu lệnh SELECT được sử dụng nhằm xác định các điều kiện đối với việc truy xuất dữ liệu. Sau mệnh đề WHERE là một biểu thức logic và chỉ những dòng dữ liệu nào thỏa mãn điều kiện được chỉ định mới được hiển thị trong kết quả truy vấn.

\* Trong mệnh đề WHERE thường sử dụng:

- + Các toán tử kết hợp điều kiện (AND, OR)
- + Các toán tử so sánh
- + Kiểm tra giới hạn của dữ liệu (BETWEEN/ NOT BETWEEN)
- + Kiểm tra khuôn dạng dữ liệu.
- + Các giá trị NULL

#### i. *Các toán tử so sánh*

Toá	ý nghĩa
=	Bằng
>	Lớn hơn
<	Nhỏ hơn
>=	Lớn hơn hoặc
<=	Nhỏ hơn hoặc
<>	Khác
!>	Không lớn hơn
!<	Không nhỏ hơn

#### ii. *Kiểm tra giới hạn của dữ liệu*

Để kiểm tra xem giá trị dữ liệu nằm trong (ngoài) một khoảng nào đó, ta sử dụng toán tử BETWEEN (NOT BETWEEN) như sau:

Cách sử dụng	Ý nghĩa
giá_trị BETWEEN a AND b	$a \leq \text{giá\_trị} \leq b$
giá_trị NOT BETWEEN a	$(\text{giá\_trị} < a) \text{ AND } (\text{giá\_trị} > b)$

**Ví dụ:** Câu lệnh dưới đây cho biết họ tên và tuổi của các sinh viên có tên là Bình và có tuổi nằm trong khoảng từ 20 đến 22

```
SELECT hodem,ten,year(getdate())-year(ngaysinh) AS tuoi
FROM sinhvien
WHERE ten='Bình' AND
YEAR(GETDATE())-YEAR(ngaysinh) BETWEEN 20 AND 22
```

iii. Danh sách (IN và NOT IN)

Ví dụ: Để biết danh sách các môn học có số đơn vị học trình là 2, 4 hoặc 5, thay vì sử dụng câu lệnh

```
SELECT * FROM monhoc
WHERE sodvht=2 OR sodvht=4 OR sodvht=5
```

ta có thể sử dụng câu lệnh `SELECT * FROM monhoc WHERE sodvht IN (2,4,5)`

iv. Toán tử LIKE và các ký tự đại diện

Từ khoá LIKE (NOT LIKE) sử dụng trong câu lệnh SELECT nhằm mô tả khuôn dạng của dữ liệu cần tìm kiếm. Chúng thường được kết hợp với các ký tự đại diện sau đây:

% :Chuỗi ký tự bất kỳ gồm không hoặc nhiều ký tự

\_ : Ký tự đơn bất kỳ

[] Ký tự đơn bất kỳ trong giới hạn được chỉ định (ví dụ [a-f]) hay một tập (ví dụ [abcdef])

[^] Ký tự đơn bất kỳ không nằm trong giới hạn được chỉ định ( ví dụ [^a-f] hay một tập (ví dụ [^abcdef])).

Ví dụ: Câu lệnh dưới đây

```
SELECT hodem,ten FROM sinhvien
WHERE hodem LIKE 'Lê%'
```

Câu lệnh:

```
SELECT hodem,ten FROM sinhvien
WHERE hodem LIKE 'Lê%' AND ten LIKE '[AB]%'
```

v. Giá trị NULL

Là một giá trị đặc biệt trong SQL Server dùng để chỉ một giá trị dữ liệu không chứa gì cả.

**d. Tạo mới bảng dữ liệu từ kết quả của câu lệnh SELECT**

Câu lệnh `SELECT ... INTO` có tác dụng tạo một bảng mới có cấu trúc và dữ liệu được xác định từ kết quả của truy vấn. Bảng mới được tạo ra sẽ có số cột bằng số cột được chỉ định trong danh sách chọn và số dòng sẽ là số dòng kết quả của truy vấn

Ví dụ:

```
SELECT hodem,ten, YEAR(GETDATE())-YEAR(ngaysinh) AS tuoi  
INTO tuoisv  
FROM sinhvien
```

Lưu ý: Nếu trong danh sách chọn có các biểu thức thì những biểu thức này phải được đặt tiêu đề.

***e. Sắp xếp kết quả truy vấn***

Câu lệnh ở ví dụ trên có thể được viết lại như sau:

```
SELECT hodem,ten,gioitinh, YEAR(GETDATE())-YEAR(ngaysinh) AS tuoi FROM  
sinhvien  
WHERE ten='Bình' ORDER BY 3, 4
```

***f. Phép hợp***

Cú pháp như sau:

Câu\_lệnh\_1

UNION [ALL] Câu\_lệnh\_2

[UNION [ALL] Câu\_lệnh\_3]

...

[UNION [ALL] Câu\_lệnh\_n]

[ORDER BY cột\_sắp\_xếp]

[COMPUTE danh\_sách\_hàm\_gộp [BY danh\_sách\_cột]]

Trong đó

*Câu\_lệnh\_1 có dạng*

SELECT danh\_sách\_cột

[INTO tên\_bảng\_mới]

[FROM danh\_sách\_bảng|khung\_nhìn]

[WHERE điều\_kiện]

[GROUP BY danh\_sách\_cột]

[HAVING điều\_kiện]

*và Câu\_lệnh\_i (i = 2,...,n) có dạng*

SELECT danh\_sách\_cột

[FROM danh\_sách\_bảng|khung\_nhìn]

[WHERE điều\_kiện]

[GROUP BY danh\_sách\_cột]

[HAVING điều\_kiện]



***g. Phép nối***

Khi cần thực hiện một yêu cầu truy vấn dữ liệu từ hai hay nhiều bảng, ta phải sử dụng đến phép nối. Một câu lệnh nối kết hợp các dòng dữ liệu trong các bảng khác nhau lại theo một hoặc nhiều điều kiện nào đó và hiển thị chúng trong kết quả truy vấn.

Xét bảng KHOA và LOP

o Chọn ra dòng trong bảng KHOA có tên khoa là Khoa Công nghệ Thông tin, từ đó xác định được mã khoa (MAKHOA) là DHT02.

o Tìm kiếm trong bảng LOP những dòng có giá trị trường MAKHOA là DHT02 (tức là bảng MAKHOA tương ứng trong bảng KHOA) và đưa những dòng này vào kết quả truy vấn

Như vậy, để thực hiện được yêu cầu truy vấn dữ liệu trên, ta phải thực hiện phép nối giữa hai bảng KHOA và LOP với điều kiện nối là MAKHOA của KHOA bằng với MAKHOA của LOP.

***i. Sử dụng phép nối***

Ví dụ: Câu lệnh dưới đây hiển thị danh sách các sinh viên với các thông tin: mã sinh viên, họ và tên, mã lớp, tên lớp và tên khoa

```
SELECT masv,hodem,ten,sinhvien. malop, tenlop, tenkhoa
```

```
FROM sinhvien,lop,khoa
```

```
WHERE sinhvien.malop = lop.malop AND lop.makhoa=khoa.makhoa
```

***ii. Các loại phép nối***

Phép nối bằng và phép nối tự nhiên

Ví dụ: Câu lệnh dưới đây thực hiện phép nối bằng giữa hai bảng LOP và KHOA

```
SELECT *
```

```
FROM lop,khoa
```

```
WHERE lop.makhoa=khoa.makhoa
```

Ví dụ: Để thực hiện phép nối tự nhiên, câu lệnh trong ví dụ trên được viết lại:

```
SELECT malop,tenlop,khoa, hedaotao,namnhaphoc,siso,lop.makhoa,tenkhoa,dienthoai
```

```
FROM lop,khoa
```

```
WHERE lop.makhoa=khoa.makhoa
```

hoặc viết dưới dạng ngắn gọn hơn:

```
SELECT lop.*,tenkhoa,dienthoai FROM lop,khoa
```

```
WHERE lop.makhoa=khoa.makhoa
```

***h. Thống kê dữ liệu với GROUP BY***

Ngoài khả năng thực hiện các yêu cầu truy vấn dữ liệu thông thường (chiều, chọn, nối,...) như đã đề cập như ở các phần trước, câu lệnh SELECT còn cho phép thực hiện các thao tác truy vấn và tính toán thống kê trên dữ liệu như: cho biết tổng số tiết dạy của mỗi giáo viên, điểm trung bình các môn học của mỗi sinh viên,...

Mệnh đề GROUP BY sử dụng trong câu lệnh SELECT nhằm phân hoạch các dòng dữ liệu trong bảng thành các nhóm dữ liệu, và trên mỗi nhóm dữ liệu thực hiện tính toán các giá trị thống kê như tính tổng, tính giá trị trung bình,...

Các hàm gộp được sử dụng để tính giá trị thống kê cho toàn bảng hoặc trên mỗi nhóm dữ liệu. Chúng có thể được sử dụng như là các cột trong danh sách chọn của câu lệnh SELECT hoặc xuất hiện trong mệnh đề HAVING, nhưng không được phép xuất hiện trong mệnh đề WHERE

Các hàm gộp mà SQL cung cấp

SUM([ALL | DISTINCT] biểu\_thức) Tính tổng các giá trị.

AVG([ALL | DISTINCT] biểu\_thức) Tính trung bình của các giá trị

COUNT([ALL | DISTINCT] biểu\_thức) Đếm số các giá trị trong biểu thức.

COUNT(\*) Đếm số các dòng được chọn. MAX(biểu\_thức) Tính giá trị lớn nhất  
MIN(biểu\_thức) Tính giá trị nhỏ nhất

Trong đó:

- o Hàm SUM và AVG chỉ làm việc với các biểu thức số.

- o Hàm SUM, AVG, COUNT, MIN và MAX bỏ qua các giá trị NULL khi tính toán.

- o Hàm COUNT(\*) không bỏ qua các giá trị NULL.

Ví dụ: Để thống kê trung bình điểm lần 1 của tất cả các môn học, ta sử dụng câu lệnh như sau:

```
SELECT AVG(diemlan1) FROM diemthi
```

Câu lệnh dưới đây cho biết tuổi lớn nhất, tuổi nhỏ nhất và độ tuổi trung bình của tất cả các sinh viên sinh tại Dalat:

```
SELECT MAX(YEAR(GETDATE())-YEAR(ngaysinh)), MIN(YEAR(GETDATE())-  
YEAR(ngaysinh)), AVG(YEAR(GETDATE())-YEAR(ngaysinh))  
FROM sinhvien
```

```
WHERE noisinh=' Dalat'
```

### **Thống kê dữ liệu trên các nhóm**

Trong trường hợp cần thực hiện tính toán các giá trị thống kê trên các nhóm dữ liệu, ta sử dụng mệnh đề GROUP BY để phân hoạch dữ liệu vào trong các nhóm. Các hàm gộp được sử dụng sẽ thực hiện thao tác tính toán trên mỗi nhóm và cho biết giá trị thống kê theo các nhóm dữ liệu.

Ví dụ: Câu lệnh dưới đây cho biết sĩ số (số lượng sinh viên) của mỗi lớp

```
SELECT lop.malop,tenlop,COUNT(masv) AS siso
```

```
FROM lop,sinhvien
```

```
WHERE lop.malop=sinhvien.malop
```

```
GROUP BY lop.malop,tenlop
```

còn câu lệnh:

```
SELECT    sinhvien.masv,hodem,ten,    sum(diemlan1*sodvht)/sum(sodvht)    FROM
sinhvien,diemthi,monhoc
```

```
WHERE    sinhvien.masv=diemthi.masv    AND    diemthi.mamonhoc=monhoc.mamonhoc
GROUP BY sinhvien.masv,hodem,ten
```

cho biết trung bình điểm thi lần 1 các môn học của các sinh viên

Lưu ý: Trong trường hợp danh sách chọn của câu lệnh SELECT có cả các hàm gộp và những biểu thức không phải là hàm gộp thì những biểu thức này phải có mặt đầy đủ trong mệnh đề GROUP BY, nếu không câu lệnh sẽ không hợp lệ.

Ví dụ: Dưới đây là một câu lệnh sai

```
SELECT lop.malop,tenlop,COUNT(masv) FROM lop,sinhvien
```

```
WHERE lop.malop=sinhvien.malop
```

```
GROUP BY lop.malop
```

do thiếu trường TENLOP sau mệnh đề GROUP BY.

### **Chỉ định điều kiện đối với hàm gộp**

Mệnh đề HAVING được sử dụng nhằm chỉ định điều kiện đối với các giá trị thống kê được sản sinh từ các hàm gộp tương tự như cách thức mệnh đề WHERE thiết lập các điều kiện cho câu lệnh SELECT.

## **8. Bài tập**

## **BÀI 5**

### **THIẾT KẾ, BẢO TRÌ VÀ PHÁT TRIỂN MÔ HÌNH CLIENT/SERVER**

**Thời lượng:** 15 giờ (5 giờ Lý thuyết, 9 giờ Thực hành, 1 giờ Kiểm tra)

#### **Mục tiêu bài học**

- Đọc được hồ sơ thiết kế CSDL
- Thực hiện được các thao tác tạo người sử dụng, phân quyền cho người sử dụng
- Chuyển đổi được CSDL đơn giản từ hệ quản trị CSDL MS SQL sang MS Access hoặc sang các hệ quản trị CSDL khác và ngược lại.
- Sao lưu và phục hồi được CSDL
- Thực hiện các thao tác an toàn với máy tính

#### **Nội dung chính**

- Đọc hồ sơ thiết kế CSDL
- Bảo mật CSDL
- Chuyển đổi CSDL từ các nguồn CSDL
- Sao lưu dự phòng cơ sở dữ liệu
- Bảo trì CSDL

#### **Nội dung chi tiết**

##### **1. Đọc hồ sơ thiết kế CSDL**

Bước đầu tiên của việc thiết kế CSDL là chúng ta phải biết cách đọc hồ sơ thiết kế CSDL. Việc đọc được sơ đồ này đòi hỏi chúng ta phải nắm rõ các ký hiệu và quy định đã được trình bày trong môn học CSDL.

##### **2. Bảo mật CSDL**

###### **a. Các khái niệm**

Bảo mật là một trong những yếu tố đóng vai trò quan trọng đối với sự sống còn của CSDL. Hầu hết các hệ quản trị CSDL thương mại hiện nay đều cung cấp khả năng bảo mật CSDL với những chức năng như:

- Cấp phát quyền truy cập CSDL cho người dùng và các nhóm người dùng, phát hiện và ngăn chặn những thao tác trái phép của người sử dụng trên CSDL.
- Cấp phát quyền sử dụng các câu lệnh, các đối tượng CSDL đối với người dùng.
- Thu hồi (huỷ bỏ) quyền của người dùng.

Bảo mật dữ liệu trong SQL được thực hiện dựa trên ba khái niệm chính sau đây:

- **Người dùng CSDL (Database user):** Là đối tượng sử dụng CSDL, thực thi các thao tác trên CSDL như tạo bảng, truy xuất dữ liệu,... Mỗi một người dùng trong CSDL được xác định thông qua tên người dùng (User ID). Một tập nhiều người dùng có thể được tổ chức trong một nhóm và được gọi là nhóm người dùng (User Group). Chính sách bảo mật CSDL có thể được áp dụng cho mỗi người dùng hoặc cho các nhóm người dùng.

- **Các đối tượng CSDL (Database objects):** Tập hợp các đối tượng, các cấu trúc lưu trữ được sử dụng trong CSDL như bảng, khung nhìn, thủ tục, hàm được gọi là các đối tượng

CSDL. Đây là những đối tượng cần được bảo vệ trong chính sách bảo mật của CSDL.

- **Đặc quyền (Privileges):** Là tập những thao tác được cấp phát cho người dùng trên các đối tượng CSDL. Chẳng hạn một người dùng có thể truy xuất dữ liệu trên một bảng bằng câu lệnh SELECT nhưng có thể không thể thực hiện các câu lệnh INSERT, UPDATE hay DELETE trên bảng đó.

SQL cung cấp hai câu lệnh cho phép chúng ta thiết lập các chính sách bảo mật trong CSDL:

- Lệnh GRANT: Sử dụng để cấp phát quyền cho người sử dụng trên các đối tượng CSDL hoặc quyền sử dụng các câu lệnh SQL trong CSDL.

- Lệnh REVOKE: Được sử dụng để thu hồi quyền đối với người sử dụng.

#### ***b. Cấp phát quyền***

Câu lệnh GRANT được sử dụng để cấp phát quyền cho người dùng hay nhóm người dùng trên các đối tượng CSDL. Câu lệnh này thường được sử dụng trong các trường hợp sau:

- Người sở hữu đối tượng CSDL muốn cho phép người dùng khác quyền sử dụng những đối tượng mà anh ta đang sở hữu.

- Người sở hữu CSDL cấp phát quyền thực thi các câu lệnh (như CREATE TABLE, CREATE VIEW,...) cho những người dùng khác.

#### ***i. Cấp phát quyền cho người dùng trên các đối tượng CSDL***

Chỉ có người sở hữu CSDL hoặc người sở hữu đối tượng CSDL mới có thể cấp phát quyền cho người dùng trên các đối tượng CSDL. Câu lệnh GRANT trong trường hợp này có cú pháp như sau:

```
GRANT ALL [PRIVILEGES]| các_quyền_cấp_phát  
[(danh_sách_cột)] ON tên_bảng | tên_khung_nhìn  
|ON tên_bảng | tên_khung_nhìn [(danh_sách_cột)]  
|ON tên_thủ_tục  
|ON tên_hàm  
TO danh_sách_người_dùng | nhóm_người_dùng  
[WITH GRANT OPTION ]
```

Trong đó:

ALL[PRIVILEGES]	Cấp phát tất cả các quyền cho người dùng trên đối tượng CSDL được chỉ định. Các quyền có thể cấp phát cho người dùng bao gồm: <ul style="list-style-type: none"> <li>• Đối với bảng, khung nhìn, và hàm trả về dữ liệu kiểu bảng: SELECT, INSERT, DELETE, UPDATE và REFERENCES.</li> <li>• Đối với cột trong bảng, khung nhìn: SELECT và UPDATE</li> <li>• Đối với thủ tục lưu trữ và hàm vô hướng: EXECUTE</li> </ul> Quyền REFERENCES được sử dụng nhằm cho phép tạo khóa ngoài tham chiếu đến bảng cấp phát.
các_quyền_cấp_phát	Danh sách các quyền cần cấp phát cho người dùng trên đối tượng CSDL được chỉ định. Các quyền phân cách nhau bởi dấu phẩy
tên_bảng  tên_khung_nhìn	Tên của bảng hoặc khung nhìn cần cấp phát quyền
danh_sách_cột	Danh sách các cột của bảng/khung nhìn cần cấp phát quyền.
tên_thủ_tục	Tên của thủ tục được cấp phát cho người dùng.
tên_hàm	Tên hàm (do người dùng định nghĩa) được cấp phát quyền.
danh_sách_người_dùng	Danh sách tên người dùng nhận quyền được cấp phát. Tên của các người dùng được phân cách nhau bởi dấu phẩy
WITH GRANT OPTION	Cho phép người dùng chuyển tiếp quyền cho người dùng khác

Ví dụ minh họa cách sử dụng câu lệnh GRANT để cấp phát quyền cho người dùng trên các đối tượng CSDL.

- Cấp phát cho người dùng có tên thuchanh quyền thực thi các câu lệnh SELECT, INSERT và UPDATE trên bảng LOP

```
GRANT SELECT,INSERT,UPDATE ON lop
TO thuchanh
```

- Cho phép người dùng thuchanh quyền xem họ tên và ngày sinh của các sinh viên (cột HODEM,TEN và NGAYSINH của bảng SINHVIEN)

```
GRANT SELECT (hodem,ten,ngaysinh)
ON sinhvien
TO thuchanh
Hoặc
GRANT SELECT
ON sinhvien(hodem,ten,ngaysinh) TO thuchanh
FROM sinhvien
```

Với quyền được cấp phát như trên, người dùng thuchanh có thể thực hiện câu lệnh **SELECT hoden,ten,ngaysinh** trên bảng SINHVIEN

Nhưng câu lệnh **SELECT \* FROM sinhvien** lại không thể thực hiện được

Trong trường hợp cần cấp phát tất cả các quyền có thể thực hiện được trên đối tượng CSDL cho người dùng, thay vì liệt kê các câu lệnh, ta chỉ cần sử dụng từ khóa ALL PRIVILEGES (từ khóa PRIVILEGES có thể không cần chỉ định).

Câu lệnh **GRANT ALL ON diemthi TO thuchanh** cấp phát cho người dùng thuchanh các quyền SELECT, INSERT, UPDATE, DELETE VÀ REFERENCES trên bảng DIEMTHI

Khi ta cấp phát quyền nào đó cho một người dùng trên một đối tượng CSDL, người dùng đó có thể thực thi câu lệnh được cho phép trên đối tượng đã cấp phát. Tuy nhiên, người dùng đó không có quyền cấp phát những quyền mà mình được phép cho những người sử dụng khác. Trong một số trường hợp, khi ta cấp phát quyền cho một người dùng nào đó, ta có thể cho phép người đó chuyển tiếp quyền cho người dùng khác bằng cách chỉ định tùy chọn WITH GRANT OPTION trong câu lệnh GRANT.

*ii. Cấp phát quyền thực thi các câu lệnh*

Ngoài chức năng cấp phát quyền cho người sử dụng trên các đối tượng CSDL, câu lệnh GRANT còn có thể sử dụng để cấp phát cho người sử dụng một số quyền trên hệ quản trị CSDL hoặc CSDL. Những quyền có thể cấp phát trong trường hợp này bao gồm:

- Tạo CSDL: CREATE DATABASE.
- Tạo bảng: CREATE TABLE
- Tạo khung nhìn: CREATE VIEW
- Tạo thủ tục lưu trữ: CREATE PROCEDURE
- Tạo hàm: CREATE FUNCTION
- Sao lưu CSDL: BACKUP DATABASE

Câu lệnh GRANT sử dụng trong trường hợp này có cú pháp như sau:

GRANT ALL | danh\_sách\_câu\_lệnh

TO danh\_sách\_người\_dùng

**Ví dụ:** Để cấp phát quyền tạo bảng và khung nhìn cho người dùng có tên là thuchanh, ta sử dụng câu lệnh như sau:

GRANT CREATE TABLE,CREATE VIEW

TO thuchanh

Với câu lệnh GRANT, ta có thể cho phép người sử dụng tạo các đối tượng CSDL trong CSDL. Đối tượng CSDL do người dùng nào tạo ra sẽ do người đó sở hữu và do đó người này có quyền cho người dùng khác sử dụng đối tượng và cũng có thể xóa bỏ (DROP) đối tượng do mình tạo ra.

Khác với trường hợp sử dụng câu lệnh GRANT để cấp phát quyền trên đối tượng CSDL, câu lệnh GRANT trong trường hợp này không thể sử dụng tùy chọn WITH GRANT OPTION, tức là người dùng không thể chuyển tiếp được các quyền thực thi các câu lệnh đã được cấp phát.

*c. Thu hồi quyền*

Câu lệnh REVOKE được sử dụng để thu hồi quyền đã được cấp phát cho người dùng. Tương ứng với câu lệnh GRANT, câu lệnh REVOKE được sử dụng trong hai trường hợp:

- Thu hồi quyền đã cấp phát cho người dùng trên các đối tượng CSDL



- Thu hồi quyền thực thi các câu lệnh trên cơ sở dữ liệu đã cấp phát cho người dùng.

*i. Thu hồi quyền trên đối tượng cơ sở dữ liệu:*

Cú pháp câu lệnh REVOKE sử dụng để thu hồi quyền đã cấp phát trên đối tượng cơ sở dữ liệu có cú pháp như sau:

```
REVOKE [GRANT OPTION FOR]
ALL [PRIVILEGES] | các_quyền_cần_thu_hồi
[(danhsáchcột)] ON tên_bảng | tên_khung_nhìn
|ON tên_bảng | tên_khung_nhìn [(danhsáchcột)]
|ON tên_thủ_tục
|ON tên_hàm
FROM danhsách_người_dùng
[CASCADE]
```

Câu lệnh REVOKE có thể sử dụng để thu hồi một số quyền đã cấp phát cho người dùng hoặc là thu hồi tất cả các quyền (ALL PRIVILEGES).

**Ví dụ:** Thu hồi quyền thực thi lệnh INSERT trên bảng LOP đối với người dùng thuchanh.

```
REVOKE INSERT
ON lop
FROM thuchanh
```

Giả sử người dùng thuchanh đã được cấp phát quyền xem dữ liệu trên các cột HODEM, TEN và NGAYSINH của bảng SINHVIEN, câu lệnh dưới đây sẽ thu hồi quyền đã cấp phát trên cột NGAYSINH (chỉ cho phép xem dữ liệu trên cột HODEM và TEN)

```
REVOKE SELECT
ON sinhvien(ngaysinh)
FROM thuchanh
```

Khi ta sử dụng câu lệnh REVOKE để thu hồi quyền trên một đối tượng CSDL từ một người dùng nào đó, chỉ những quyền mà ta đã cấp phát trước đó mới được thu hồi, những quyền mà người dùng này được cho phép bởi những người dùng khác vẫn còn có hiệu lực. Nói cách khác, nếu hai người dùng khác nhau cấp phát cùng các quyền trên cùng một đối tượng CSDL cho một người dùng khác, sau đó người thu nhất thu hồi lại quyền đã cấp phát thì những quyền mà người dùng thứ hai cấp phát vẫn có hiệu lực.

*ii. Thu hồi quyền thực thi các câu lệnh:*

Việc thu hồi quyền thực thi các câu lệnh trên cơ sở dữ liệu (CREATE DATABASE, CREATE TABLE, CREATE VIEW,...) được thực hiện đơn giản với câu lệnh REVOKE có cú pháp:

```
REVOKE ALL | các_câu_lệnh_cần_thu_hồi
FROM danhsách_người_dùng
```

**Ví dụ:** Để không cho phép người dùng thuchanh thực hiện lệnh CREATE TABLE trên cơ sở dữ liệu, ta sử dụng câu lệnh:

REVOKE CREATE TABLE

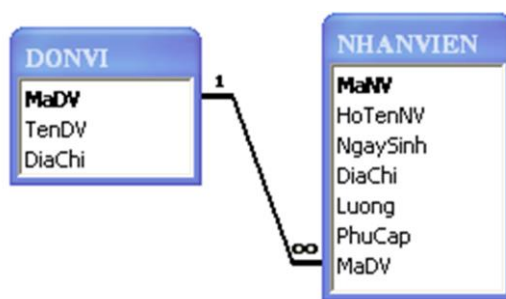
FROM thuchanh

### 3. Chuyển đổi cơ sở dữ liệu từ các nguồn cơ sở dữ liệu

Việc chuyển đổi cơ sở dữ liệu có thể thực hiện qua lại giữa các hệ quản trị cơ sở dữ liệu. Việc chuyển đổi chỉ thông qua một số bước đơn giản. Trong giới hạn của module này, chúng ta sẽ tìm hiểu về cách chuyển đổi qua lại giữa nguồn cơ sở dữ liệu SQL Server và cơ sở dữ liệu Access. Đối với các cơ sở dữ liệu khác, chúng ta cũng có cách làm tương tự.

#### d. *Tạo cơ sở dữ liệu SQL Server từ nguồn cơ sở dữ liệu Access*

Trước tiên, để thuận tiện cho việc thực hành chúng ta tạo một cơ sở dữ liệu Access có tên QuanLyNhanVien (quản lý nhân viên) với hai bảng và quan hệ như sau:

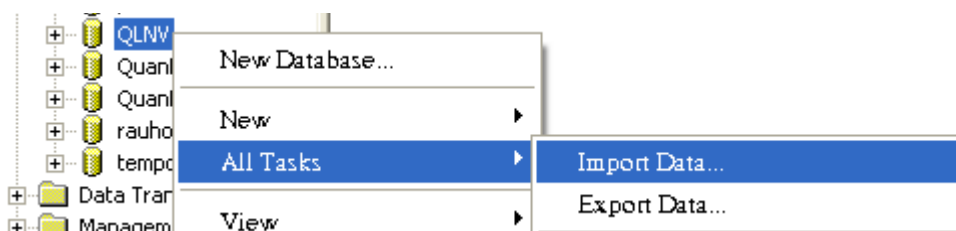


*Hình quan hệ giữa hai bảng*

Tiếp theo, chúng ta nhập một số bản ghi cho từng bảng và đóng cơ sở dữ liệu lại.

#### **Các bước kết xuất dữ liệu từ Access:**

- 1 - Tạo một cơ sở dữ liệu SQL mới, ở đây chúng ta đặt tên là **QLNV**.
- 2 - Click chuột phải lên cơ sở dữ liệu All Tasks/ Import Data...



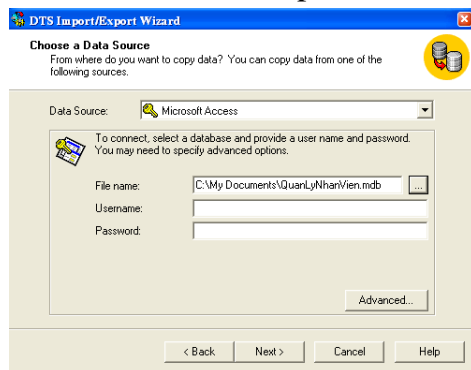
*Hình: Import Data*

một cửa sổ sẽ hiện ra như sau:



Hình 5.3: Data Transformation Services

3 – Nhấn Next để tiếp tục, cửa sổ tiếp theo sẽ hiện ra:



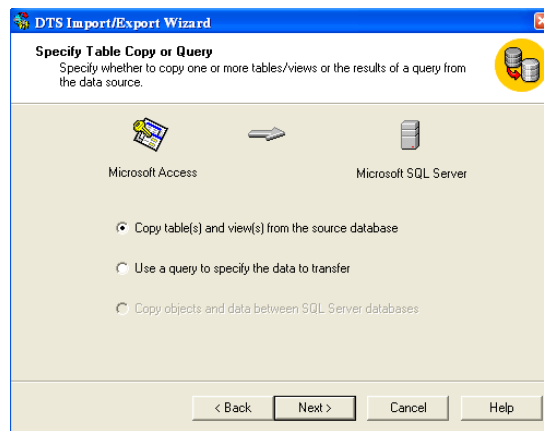
Hình 5.4: Choose a data source

4 – Chọn nguồn cơ sở dữ liệu. Trong hộp Data Source chọn Microsoft Access, trong hộp File name chọn cơ sở dữ liệu Access đã được tạo trước (QuanLyNhanVien.mdb) như hình trên, nhấn Next để tiếp tục.

5 – Chọn cơ sở dữ liệu đích. Để mặc định như hình dưới đây, nếu bạn chọn kết nối với tài khoản của SQL Server thì đánh dấu vào ô **Use SQL Server Authentication**, trong hộp **Username** nhập tài khoản đăng nhập quản trị cơ sở dữ liệu như hình dưới đây:



Nhấn Next để tiếp tục, cửa sổ mới hiện ra:



6 – Để mặc định như hình trên nếu muốn sao chép toàn bộ bảng và truy vấn từ cơ sở dữ liệu nguồn, nhấn Next để tiếp tục.

7 – Đánh dấu vào bảng và truy vấn cần sao chép, nhấn Next / Next để tiếp tục:

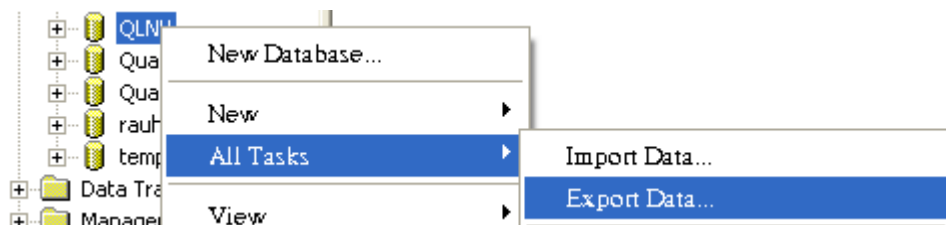


8. Nhấn **Finish**. Đợi trong vài giây, phụ thuộc vào dung lượng cơ sở dữ liệu nguồn mà thời gian này có thể lâu hơn, sau đó nhấn **Done** để hoàn tất công việc.

#### ***e. Xuất cơ sở dữ liệu SQL Server sang cơ sở dữ liệu Access***

Nhấn chuột phải lên cơ sở dữ liệu SQL Server ☐ chọn All Tasks ☐

Export Data ...



Thực hiện lần lượt 8 bước tương tự như trên, chỉ thay đổi CSDL nguồn và đích.

#### **4. Sao lưu dự phòng cơ sở dữ liệu**

Trong phần này chúng ta sẽ bàn về cách sao lưu cơ sở dữ liệu (backup database).

Thuật ngữ	Giải Thích
Backup	Quá trình copy toàn bộ hay một phần của database, transaction log, file hay file group hình thành một backup set. Backup set được chứa trên backup media (tape or disk) bằng cách sử dụng một backup device (tape drive name hay physical filename)
Backup Device	Một file vật lý (như C:\SQLBackups\Full.bak) hay tape drive cụ thể (như \\.\Tape0) dùng để record một backup vào một backupmedia.
BackupFile	File chứa một backup set
BackupMedia	Disk hay tape được sử dụng để chứa một backup set. Backup Media có thể chứa nhiều backup sets (ví dụ như từ nhiều SQLServer 2000 backups và từ nhiều Windows 2000 backups).
BackupSet	Một bộ backup từ một lần backup đơn được chứa trên backup Media.

Chúng ta có thể tạo một backup device cố định (permanent) hay tạo ra một backup file mới cho mỗi lần backup. Thông thường chúng ta sẽ tạo một backup device cố định để có thể dùng đi dùng lại đặc biệt cho việc tự động hóa công việc backup. Để tạo một backup device dùng Enterprise Manager bạn chọn **Management->Backup** rồi **Right-click->New Backup Device**. Ngoài ra bạn có thể dùng **sp\_addumpdevice** system stored procedure như ví dụ sau:

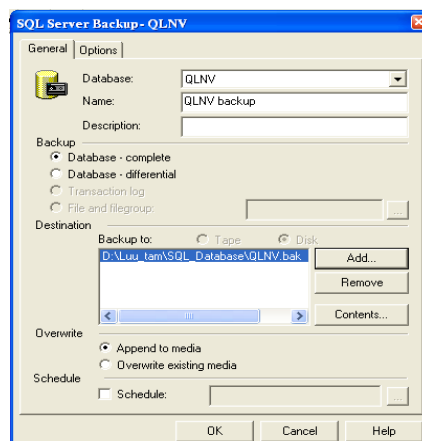
```
USE Master
```

```
Go
```

```
Sp_addumpdevice 'disk', 'FullBackupDevice',
```

```
'E:\SQLBackups\Full.bak'
```

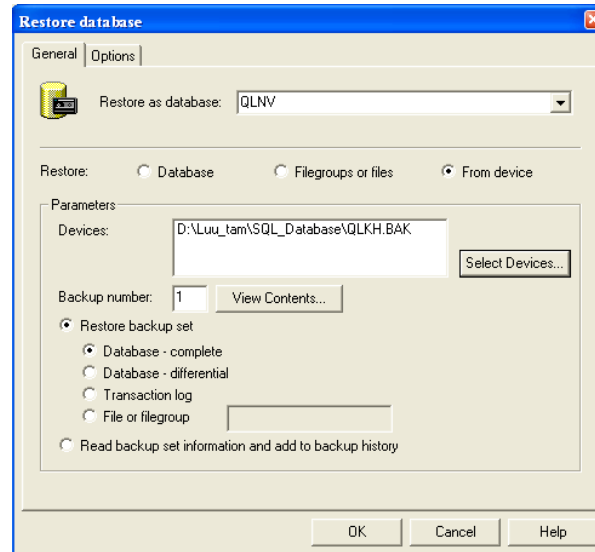
Để backup database bạn có thể dùng Backup Wizard hoặc click lên trên database muốn backup sau đó **Right-click->All Tasks->BackupDatabase...** sẽ hiện ra window như hình vẽ sau:



## 5. Bảo trì cơ sở dữ liệu

Việc sao lưu dự phòng cơ sở dữ liệu nhằm phục hồi lại khi cần thiết, đó là một quá trình bảo trì cơ sở dữ liệu. Trước khi phục hồi cơ sở dữ liệu (restore database) ta phải xác định được thứ tự file cần phục hồi. Các thông tin này được SQL Server chứa trong **msdb**. Việc phục hồi lại dữ liệu có thể thực hiện như sau:

**Right-click->All Tasks->Restore database...** sẽ thấy window như hình vẽ sau:



Nếu chúng ta muốn phục hồi cơ sở dữ liệu từ một tập tin khác của SQL Server hay từ một server khác bạn có chọn tùy chọn **From device** và chọn backup device (file backup) tương ứng.

## 6. Bài tập

1. Tạo hai tài khoản đăng nhập là **nguoisudung1**, **nguoisudung2** và cho phép hai tài khoản này sử dụng cơ sở dữ liệu QuanLyNhanVien.
2. Cấp phát tất cả các quyền cho **nguoisudung1**, cấp quyền **SELECT** cho **nguoisudung2** trên hai bảng của cơ sở dữ liệu QuanLyNhanVien.
3. Thu hồi quyền **Delete** đối với **nguoisudung1** trên bảng **DONVI**.
4. Sao lưu cơ sở dữ liệu QuanLyNhanVien và lưu trên thư mục **C:\BackUp\Data** với tên **QLNV.bak**.
5. Xóa cơ sở dữ liệu QuanLyNhanVien khỏi hệ thống và phục hồi lại như cũ nhờ tập tin **QLNV.bak**.