



Bài 1. Tổng quan về Hệ quản trị CSDL và xây dựng CSDL trên hệ quản trị CSDL SQL Server

- Mục đích, yêu cầu: Cung cấp cho sinh viên kiến thức tổng quan về Tổng quan về Hệ quản trị CSDL và cách thức xây dựng CSDL trên hệ quản trị CSDL SQL Server.
- Hình thức tổ chức dạy học: Lý thuyết, trực tiếp + trực tuyến + tự học
- Thời gian: Lý thuyết(trên lớp: 3; online: 2) Tự học, tự nghiên cứu: 10
- Nội dung chính:

Tổng quan về Hệ quản trị CSDL và xây dựng CSDL trên hệ quản trị CSDL SQL Server

I. Tổng quan về DBMS và SQL Server

Chương này trình bày một cách nhìn khái quát về cơ sở dữ liệu (CSDL/DB), về hệ quản trị cơ sở dữ liệu (HQTCSDL/DBMS) và về hệ cơ sở dữ liệu (HCSDL/DBS). Các đòi hỏi khi xây dựng một HQTCSDL đó cũng chính là những chức năng mà một HCSDL cần phải có.

Trong chương này chúng tôi cũng muốn giới thiệu tổng quan về ngôn ngữ hỏi có cấu trúc (SQL) và các hệ quản trị cơ sở dữ liệu quan hệ là một trong những nền tảng kỹ thuật quan trọng trong công nghiệp máy tính. Cho đến nay, có thể nói rằng SQL đã được xem là ngôn ngữ chuẩn trong cơ sở dữ liệu. Các hệ quản trị cơ sở dữ liệu quan hệ thương mại hiện có như Oracle, SQL Server, Informix, DB2,... đều chọn SQL làm ngôn ngữ cho sản phẩm của mình

Vậy thực sự SQL là gì? Tại sao nó lại quan trọng trong các hệ quản trị cơ sở dữ liệu? SQL có thể làm được những gì và như thế nào? Nó được sử dụng ra sao trong các hệ quản trị cơ sở dữ liệu quan hệ? Chương này sẽ cung cấp cho chúng ta cái nhìn tổng quan về SQL và một số vấn đề liên quan.

Ta tìm hiểu DBMS trên một HQCSDL cụ thể: SQL Server 2000. Do vậy chương này giới thiệu cài đặt SQL Server 2000 và các thành phần của nó, giúp chúng ta chủ



động khai thác trong nắm bắt và tạo lập ứng dụng.

1.1. Một số khái niệm

Một cơ sở dữ liệu - CSDL(DataBase): Là một kho dữ liệu được tổ chức theo một nguyên tắc nào đó. Đó là một tập hợp các tập tin có liên quan với nhau, được thiết kế nhằm làm giảm thiểu sự dư thừa dữ liệu, đảm bảo tính tin cậy khi truy xuất dữ liệu. Các tập tin này chứa các thông tin biểu diễn các đối tượng trong một ứng dụng thế giới thực.

CSDL lưu giữ thông tin của một trường đại học như : khoa, giảng viên, sinh viên, khóa học,...

Thông thường, một cơ sở dữ liệu sẽ bao trùm tất cả các thông tin của một ứng dụng, không nên đặt hai cơ sở dữ liệu vào một ứng dụng.

Hệ quản trị cơ sở dữ liệu DBMS(DataBaseManagement System): là một hệ thống gồm một CSDL và các thao tác trên CSDL. Đó là hệ thống chương trình, công cụ cho phép quản lý và tương tác với CSDL. Trên đó người dùng có thể định nghĩa, thao tác, và xử lý dữ liệu trong một CSDL để xuất ra những thông tin có nghĩa. Ví dụ 1-5 : một DBMS có thể quản trị cơ sở dữ liệu của một trường đại học cũng như những cơ sở dữ liệu có ý nghĩa khác như : cơ sở dữ liệu phục vụ tổng thu nhập quốc gia,

một cơ sở dữ liệu liên hợp quốc về dữ liệu địa lý thế giới,v.v...

- Một hệ cơ sở dữ liệu (HCSDL/ DBS: DataBase System) là một phần mềm cho phép xây dựng một HQTCSDL.

Các vấn đề cần xử lý của hệ cơ sở dữ liệu

Một số điểm bất lợi chính của việc lưu giữ *thông tin có tổ chức* trong hệ thống xử lý file thông thường mà hệ HCSDL cần lưu ý:

- **Dư thừa dữ liệu và tính không nhất quán (Data redundancy and inconsistency):**
Do các file và các trình ứng dụng được tạo ra bởi các người lập trình khác nhau,



nên các file có định dạng khác nhau, các chương trình được viết trong các ngôn ngữ lập trình khác nhau, cùng một thông tin có thể được lưu giữ trong các file khác nhau. Tính không thống nhất và dư thừa này sẽ làm *tăng chi phí truy xuất và lưu trữ*, hơn nữa, nó sẽ dẫn đến tính không nhất quán của dữ liệu: *các bản sao của cùng một dữ liệu có thể không nhất quán*.

- **Khó khăn trong việc truy xuất dữ liệu:** Môi trường của hệ thống xử lý file thông thường không cung cấp các công cụ cho phép truy xuất thông tin một cách hiệu quả và thuận lợi.

- **Sự cô lập dữ liệu (Data isolation) :** Các giá trị dữ liệu được lưu trữ trong cơ sở dữ liệu phải thỏa mãn một số các ràng buộc về tính nhất quán của dữ liệu (ràng buộc nhất quán - consistency constraints).

Trong hệ thống xử lý file thông thường, rất khó khăn trong việc thay đổi các chương trình để thỏa mãn các yêu cầu thay đổi ràng buộc. Vấn đề trở nên khó khăn hơn khi các ràng buộc liên quan đến các hạng mục dữ liệu nằm trong các file khác nhau.

- **Các vấn đề về tính nguyên tử (Atomicity problems):**

Tính nguyên tử của một hoạt động (giao dịch) là: *hoặc nó được hoàn tất trọn vẹn hoặc không có gì cả*. Điều này có nghĩa là một hoạt động (giao dịch) *chỉ làm thay đổi* các dữ liệu bền vững khi nó đã hoàn tất (kết thúc thành công) nếu không, giao dịch không để lại một dấu vết nào trên CSDL. Trong hệ thống xử lý file thông thường khó đảm bảo được tính chất này.

- **Tính bất thường trong truy xuất cạnh tranh :** Một hệ thống cho phép nhiều người sử dụng cập nhật dữ liệu đồng thời, có thể dẫn đến kết quả là dữ liệu không nhất quán. Điều này đòi hỏi một sự giám sát. Hệ thống xử lý file thông thường không cung cấp chức năng này.

- **Vấn đề an toàn (Security problems):** một người sử dụng hệ cơ sở dữ liệu không cần thiết và cũng không có quyền truy xuất tất cả các dữ liệu. Vấn đề này đòi hỏi hệ thống phải đảm bảo được tính phân quyền, chống truy xuất trái phép ... Các bất lợi nêu trên đã gợi mở sự phát triển các DBMS. Phần sau của giáo trình sẽ đề cập đến các quan niệm và các thuật toán được sử dụng để phát triển một hệ cơ sở dữ liệu nhằm *giải quyết các vấn đề nêu trên*.



Hầu hết các hệ quản trị CSDL đều thực hiện các chức năng sau :

- Lưu trữ dữ liệu
- Tạo ra và duy trì CSDL
- Cho phép nhiều người dùng truy xuất đồng thời
- Hỗ trợ tính bảo mật và riêng tư
- Cho phép xem và xử lý dữ liệu lưu trữ
- Cho phép cập nhật và lưu trữ dữ liệu sau khi cập nhật
- Cung cấp một cơ chế chỉ mục (index) hiệu quả để truy cập nhanh các dữ liệu lựa chọn
- Cung cấp tính nhất quán giữa các bản ghi khác nhau
- Bảo vệ dữ liệu khỏi mất mát bằng các quá trình sao lưu (backup) và phục hồi (recovery).

1.2. Tổng quan về cơ sở dữ liệu quan hệ

Mô hình dữ liệu quan hệ

Mô hình dữ liệu quan hệ được Codd đề xuất năm 1970 và đến nay trở thành mô hình được sử dụng phổ biến trong các hệ quản trị cơ sở dữ liệu thương mại. Nói một cách đơn giản, một cơ sở dữ liệu quan hệ là một cơ sở dữ liệu trong đó tất cả dữ liệu được tổ chức trong các bảng có mối quan hệ với nhau. Mỗi một bảng bao gồm các dòng và các cột: mỗi một dòng được gọi là một bản ghi (bộ) và mỗi một cột là một trường (thuộc tính). Hai hay nhiều bảng có thể có liên kết nếu chúng có một hay nhiều trường chung)

Hình 1.1 minh họa cho ta thấy được 3 bảng trong một cơ sở dữ liệu



Bảng KHOA

MAKHOA	TENKHOA	DIENTHOAI
DHT01	Khoa Toán cơ - Tin học	054822407
DHT02	Khoa Công nghệ thông tin	054826767
DHT03	Khoa Vật lý	054822462
DHT04	Khoa Hoá học	
...	...	

Bảng LOP

MALOP	TENLOP	KHOA	HEDAOTAO	NAMNHAPHOC	SISO	MAKHOA
C24101	Toán K24	24	Chính quy	2000	5	DHT01
C24102	Tin K24	24	Chính quy	2000	8	DHT02
C24103	Lý K24	24	Chính quy	2000	7	DHT03
C24301	Sinh K24	24	Chính quy	2000	5	DHT05

Bảng SINHVIEN

MASV	HODEM	TEN	NGAYSINH	GIOTINH	NOISINH	MALOP
0241010001	Ngô Thị Nhật	Anh	Nov 27 1982	0	Quảng Ninh, Quảng Bình	C24101
0241010002	Nguyễn Thị Ngọc	Anh	Mar 21 1983	0	Tân Kỳ, Nghệ An	C24101
0241010003	Ngô Việt	Bắc	May 11 1982	1	Yên Khánh, Ninh Bình	C24101
0241010004	Nguyễn Đình	Bình	Oct 6 1982	1	Huế	C24101
0241010005	Hồ Đăng	Chiến	Jan 20 1982	1	Phong Điền, TTHuế	C24101
0241020001	Nguyễn Tuấn	Anh	Jul 15 1979	1	Đo Linh, Quảng Trị	C24102
0241020002	Trần Thị Kim	Anh	Nov 4 1982	0	Phong Điền, TTHuế	C24102
0241020003	Võ Đức	Ẩn	May 24 1982	1	Huế	C24102
0241020004	Nguyễn Công	Bình	Jun 6 1979	1	Thăng Bình, Quảng Nam	C24102
0241020005	Nguyễn Thanh	Bình	Apr 24 1982	1	Huế	C24102
...

Các bảng trong một cơ sở dữ liệu

Bảng (Table)

Như đã nói ở trên, trong cơ sở dữ liệu quan hệ, bảng là đối tượng được sử dụng để tổ chức và lưu trữ dữ liệu. Một cơ sở dữ liệu bao gồm nhiều bảng và mỗi bảng được xác định duy nhất bởi tên bảng. Một bảng bao gồm một tập các dòng và các cột: mỗi một dòng trong bảng biểu diễn cho một thực thể (trong hình 1.1, mỗi một dòng trong bảng SINHVIEN tương ứng với một sinh viên), và mỗi một cột biểu diễn cho một tính chất của thực thể (chẳng hạn cột NGAYSINH trong bảng SINHVIEN biểu diễn cho ngày sinh của các sinh viên được lưu trữ trong bảng).

Như vậy, liên quan đến mỗi một bảng bao gồm các yếu tố sau:

- Tên của bảng: được sử dụng để xác định duy nhất mỗi bảng trong cơ sở dữ liệu.



- Cấu trúc của bảng: Tập các cột trong bảng. Mỗi một cột trong bảng được xác định bởi một tên cột và phải có một kiểu dữ liệu nào đó (chẳng hạn cột NGAYSINH trong bảng SINHVIEN ở hình 1.1 có kiểu là DATETIME). Kiểu dữ liệu của mỗi cột qui định giá trị dữ liệu có thể được chấp nhận trên cột đó.
- Dữ liệu của bảng: Tập các dòng (bản ghi) hiện có trong bảng.

Khoá của bảng

Trong một cơ sở dữ liệu được thiết kế tốt, mỗi một bảng phải có một hoặc một tập các cột mà giá trị dữ liệu của nó xác định duy nhất một dòng trong một tập các dòng của bảng.

Tập một hoặc nhiều cột có tính chất này được gọi là khoá của bảng.

Việc chọn khoá của bảng có vai trò quan trọng trong việc thiết kế và cài đặt các cơ sở dữ liệu quan hệ. Các dòng dữ liệu trong một bảng phải có giá trị khác nhau trên khoá.

Bảng MONHOC trong hình dưới đây có khoá là cột MAMONHOC

MAMONHOC	TENMONHOC	SODVHT
HO-001	Hoá đại cương	3
TI-001	Tin học đại cương	4
TI-002	Ngôn ngữ C	5
TI-003	Lý thuyết hệ điều hành	4
TI-004	Cấu trúc dữ liệu và giải thuật	4
TO-001	Đại số tuyến tính	4
TO-002	Giải tích 1	4
TO-003	Bài tập Đại số	2
TO-004	Bài tập Giải tích 1	2
VL-001	Vật lý đại cương	3

Bảng MONHOC với khoá chính là MAMONHOC

Một bảng có thể có nhiều tập các cột khác nhau có tính chất của khoá (tức là giá trị của nó xác định duy nhất một dòng dữ liệu trong bảng). Trong trường hợp này, khoá được chọn cho bảng được gọi là khoá chính (primary key) và những khoá

còn lại được gọi là khoá phụ hay là khoá dự tuyển (candidate key/unique key).

Mối quan hệ và khoá ngoài

Các bảng trong một cơ sở dữ liệu không tồn tại độc lập mà có mối quan hệ mật thiết với nhau về mặt dữ liệu. Mối quan hệ này được thể hiện thông qua ràng buộc giá trị dữ liệu xuất hiện ở bảng này phải có xuất hiện trước trong một bảng khác. Mối quan hệ giữa các bảng trong cơ sở dữ liệu nhằm đảm bảo được tính đúng đắn và hợp lệ của dữ liệu trong cơ sở dữ liệu.

Trong hình 1.3, hai bảng LOP và KHOA có mối quan hệ với nhau. Mối quan hệ này đòi hỏi giá trị cột MAKHOA của một dòng (tức là một lớp) trong bảng LOP phải được xác định từ cột MAKHOA của bảng KHOA.

MAKHOA	TENKHOA	DIENTHOAI
DHT01	Khoa Toán cơ - Tin học	054822407
DHT02	Khoa Công nghệ thông tin	054826767
DHT03	Khoa Vật lý	054823462
...

MALOP	TENLOP	KHOA	HEDAOTAO	NAMNHAPHOC	SISO	MAKHOA
C24101	Toán K24	24	Chính quy	2000	5	DHT01
C25101	Toán K25	25	Chính quy	2001	5	DHT01
C25102	Tin K25	25	Chính quy	2001	6	DHT02
C24102	Tin K24	24	Chính quy	2000	8	DHT02
...

Mối quan hệ giữa hai bảng LOP và KHOA trong cơ sở dữ liệu

Mối quan hệ giữa các bảng trong một cơ sở dữ liệu thể hiện đúng mối quan hệ giữa các thực thể trong thế giới thực. Trong hình 1.3, mối quan hệ giữa hai bảng LOP và KHOA không cho phép một lớp nào đó tồn tại mà lại thuộc vào một khoa không có thật.

Khái niệm khoá ngoài (Foreign Key) trong cơ sở dữ liệu quan hệ được sử dụng để biểu diễn mối quan hệ giữa các bảng dữ liệu. Một hay một tập các cột trong một bảng mà giá trị của nó được xác định từ khóa chính của một bảng khác được gọi là



khoá ngoài.

Trong hình 1.3, cột MAKHOA của bảng LOP được gọi là khoá ngoài của bảng này, khoá ngoài này tham chiếu đến khoá chính của bảng KHOA là cột MAKHOA.

1.3. Giới Thiệu SQL Server

SQL Server là một hệ thống quản trị cơ sở dữ liệu quan hệ (Relational Database Management System (RDBMS)) sử dụng Transact-SQL để trao đổi dữ liệu giữa Client computer và SQL Server computer. Một RDBMS bao gồm databases, database engine và các ứng dụng dùng để quản lý dữ liệu và các bộ phận khác nhau trong RDBMS.

SQL Server được tối ưu để có thể chạy trên môi trường cơ sở dữ liệu rất lớn (Very Large Database Environment) lên đến Tera-Byte và có thể phục vụ cùng lúc cho hàng ngàn user. SQL Server có thể kết hợp "ăn ý" với các server khác như Microsoft InternetInformation Server (IIS), E-Commerce Server, Proxy Server....

1.4. Cài Đặt SQL Server (Installation)

Khi install cần lưu ý các điểm sau:

Ở màn hình thứ hai bạn chọn **Install Database Server**. Sau khi install xong SQL Server bạn có thể install thêm Analysis Service nếu thích.

Ở màn hình **Installation Definition** chọn **Server and Client Tools**.

Sau đó nên chọn kiểu **Custom** và **chọn tất cả** các bộ phận của SQL Server. Ngoài ra nên

chọn các giá trị mặc định (default)

Ở màn hình **Authentication Mode** nhớ chọn **Mixed Mode**. Lưu ý vì SQL Server có thể dùng chung chế độ bảo mật (security) với Win NT và cũng có thể dùng chế độ bảo mật riêng của nó. Trong Production Server người ta thường dùng Windows Authentication vì độ an toàn cao hơn và dễ dàng cho người quản lý mạng



và cả cho người sử dụng. Nghĩa là một khi bạn được chấp nhận (authenticated) kết nối vào domain thì bạn có quyền truy cập dữ liệu (access data) trong SQL Server. Tuy nhiên ta nên chọn Mixed Mode để dễ dàng cho việc học tập.

Sau khi install bạn sẽ thấy một icon nằm ở góc phải bên dưới màn hình, đây chính là Service Manager. Có thể Start, Stop các SQL Server services dễ dàng bằng cách doubleclick vào icon này.

1.5. Các thành phần quan trọng trong SQL Server

SQL Server được cấu tạo bởi nhiều thành phần như Relational Database Engine, Analysis Service và English Query.... Các thành phần này khi phối hợp với nhau tạo thành một giải pháp hoàn chỉnh giúp cho việc lưu trữ và phân tích dữ liệu một cách dễ dàng

Relational Database Engine - Cái lõi của SQL Server:

Đây là một engine có khả năng chứa data ở các quy mô khác nhau dưới dạng table và support tất cả các kiểu kết nối (data connection) thông dụng của Microsoft như ActiveX Data Objects (ADO), OLE DB, and Open Database Connectivity (ODBC). Ngoài ra nó còn có khả năng tự điều chỉnh (tune up) ví dụ như sử dụng thêm các tài nguyên (resource) của máy khi cần và trả lại tài nguyên cho hệ điều hành khi một user log off.

Replication - Cơ chế tạo bản sao (Replica):

Giả sử bạn có một database dùng để chứa dữ liệu được các ứng dụng thường xuyên cập nhật. Khi bạn muốn có một database giống hệt như thế trên một server khác để chạy báo cáo (report database) (cách làm này thường dùng để tránh ảnh hưởng đến performance của server chính). Vấn đề là report server của bạn cũng cần phải được cập nhật thường xuyên để đảm bảo tính chính xác của các báo cáo. Ta không thể dùng cơ chế back up and restore trong trường hợp này. Vậy cần xử lý thế nào? Lúc đó cơ chế replication của SQL Server sẽ được sử dụng để bảo đảm cho dữ



liệu ở 2 database được đồng bộ (synchronized).

Data Transformation Service (DTS) - Một dịch vụ chuyển dịch data hiệu quả .

Nếu bạn làm việc trong một công ty lớn trong đó data được chứa trong nhiều nơi khác nhau và ở các dạng khác nhau cụ thể như chứa trong Oracle, DB2 (của IBM), SQL Server, Microsoft Access....Bạn chắc chắn sẽ có nhu cầu di chuyển data giữa các server này (migrate hay transfer) và không chỉ di chuyển bạn còn muốn định dạng (format) nó trước khi lưu vào database khác, khi đó bạn sẽ thấy DTS giúp bạn giải quyết công việc trên dễ dàng như thế nào.

Analysis Service- Một dịch vụ phân tích dữ liệu rất hay của Microsoft

Dữ liệu (Data) chứa trong database sẽ chẳng có ý nghĩa gì nhiều nếu như bạn không thể lấy được những thông tin (Information) bổ ích từ đó. Do đó Microsoft cung cấp cho bạn một công cụ rất mạnh giúp cho việc phân tích dữ liệu trở nên dễ dàng và hiệu quả bằng cách dùng khái niệm hình khối nhiều chiều (multi-dimension cubes) và kỹ thuật khai phá dữ liệu (data mining).

English Query - Đây là một dịch vụ giúp cho việc query data bằng tiếng Anh "trơn" (plain English).

MetaData Service : Dịch vụ này giúp cho việc chứa đựng và "xào nấu" Meta data dễ dàng hơn. Thế thì Meta Data là cái gì vậy? Meta data là những thông tin mô tả về cấu trúc của data trong database như data thuộc loại nào String hay Integer..., một cột nào đó có phải là Primary key hay không....Bởi vì những thông tin này cũng được chứa trong database nên cũng là một dạng data nhưng để phân biệt với data "chính thống" người ta gọi nó là Meta Data. Phần này phải xem thêm trong một thành phần khác của SQL Server là **SQLServerBooks Online**.

SQL Server Books Online - Rất hữu dụng và không thể thiếu (được đính kèm theo SQL Server).

SQL Server Tools - Đây là một bộ đồ nghề của người quản trị cơ sở dữ liệu (DBA),



gồm:

Enterprise Manager - Đây là một công cụ cho ta thấy toàn cảnh hệ thống cơ sở dữ liệu một cách rất trực quan. Nó rất hữu ích đặc biệt cho người mới học và không thông thạo lắm về SQL.

Query Analyzer - Đối với một DBA giỏi thì hầu như chỉ cần công cụ này là có thể quản lý cả một hệ thống database mà không cần đến những thứ khác. Đây là một môi trường làm việc khá tốt vì ta có thể đánh bất kỳ câu lệnh SQL nào và chạy ngay lập tức đặc biệt là nó giúp cho ta debug stored procedure dễ dàng.

SQL Profiler - Nó có khả năng "chụp" (capture) tất cả các sự kiện hay hoạt động diễn ra trên một SQL server và lưu lại dưới dạng text file rất hữu dụng trong việc kiểm soát hoạt động của SQL Server.

Ngoài một số công cụ trực quan như trên chúng ta cũng thường hay dùng **osql** và **bcp** (bulk copy) trong command prompt.

1.6. SQL là ngôn ngữ cơ sở dữ liệu quan hệ

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

Tên gọi *ngôn ngữ hỏi có cấu trúc* phần nào làm chúng ta liên tưởng đến một công cụ (ngôn ngữ) dùng để truy xuất dữ liệu trong các cơ sở dữ liệu. Thực sự mà nói, khả năng của SQL vượt xa so với một công cụ truy xuất dữ liệu, mặc dù đây là mục đích ban đầu khi SQL được xây dựng nên và truy xuất dữ liệu vẫn còn là một trong những chức năng quan trọng của nó. SQL được sử dụng để điều khiển tất cả các chức năng mà một hệ quản trị cơ sở dữ liệu cung cấp cho người dùng bao gồm:

- **Định nghĩa dữ liệu** : SQL cung cấp khả năng định nghĩa các cơ sở dữ liệu, 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** : Với SQL, người dùng có thể dễ dàng thực hiện các thao tác truy xuất, bổ sung, cập nhật và loại bỏ dữ liệu trong các cơ sở dữ liệu.
- **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 cơ sở dữ
- **Đảm bảo toàn vẹn dữ liệu** : SQL định nghĩa các ràng buộc toàn vẹn trong cơ sở dữ liệu 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.

Như vậy, có thể nói rằng SQL là một ngôn ngữ hoàn thiện được sử dụng trong các hệ thống cơ sở dữ liệu và là một thành phần không thể thiếu trong các hệ quản trị cơ sở dữ liệu. Mặc dù SQL không phải là một ngôn ngữ lập trình như C, C++, Java,... song 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 cơ sở dữ liệu.

Khác với các ngôn ngữ lập trình quen thuộc như C, C++, Java,... SQL là ngôn ngữ có tính khai báo. Với SQL, người dùng chỉ cần mô tả các yêu cầu cần phải thực hiện trên cơ sở dữ liệu mà không cần phải chỉ ra cách thức thực hiện các yêu cầu như thế nào.

Chính vì vậy, SQL là ngôn ngữ dễ tiếp cận và dễ sử dụng.

1.7. Sơ lược về SQL

Tổng quan về T- SQL

Bản thân SQL không phải là một hệ quản trị cơ sở dữ liệu, nó không thể tồn tại độc lập. SQL thực sự là một phần của hệ quản trị cơ sở dữ liệu, nó xuất hiện trong các hệ quản trị cơ sở dữ liệu với vai trò ngôn ngữ và là công cụ giao tiếp giữa người sử dụng và hệ quản trị cơ sở dữ liệu.

Trong hầu hết các hệ quản trị cơ sở dữ liệu quan hệ, SQL có những vai trò như sau:



- **SQL là ngôn ngữ hỏi có tính tương tác:** Người sử dụng có thể dễ dàng thông qua các trình tiện ích để gửi các yêu cầu dưới dạng các câu lệnh SQL đến cơ sở dữ liệu và nhận kết quả trả về từ cơ sở dữ liệu
- **SQL là ngôn ngữ lập trình cơ sở dữ liệu:** Các lập trình viên có thể nhúng các câu lệnh SQL vào trong các ngôn ngữ lập trình để xây dựng nên các chương trình ứng dụng giao tiếp với cơ sở dữ liệu
- **SQL là ngôn ngữ quản trị cơ sở dữ liệu :** Thông qua SQL, người quản trị cơ sở dữ liệu có thể quản lý được cơ sở dữ liệu, định nghĩa các cấu trúc lưu trữ dữ liệu, điều khiển truy cập cơ sở dữ liệu,...
- **SQL là ngôn ngữ cho các hệ thống khách/chủ (client/server):** Trong các hệ thống cơ sở dữ liệu khách/chủ, SQL được sử dụng như là công cụ để giao tiếp giữa các trình ứng dụng phía máy khách với máy chủ cơ sở dữ liệu.
- **SQL là ngôn ngữ truy cập dữ liệu trên Internet :** Cho đến nay, hầu hết các máy chủ Web cũng như các máy chủ trên Internet sử dụng SQL với vai trò là ngôn ngữ để tương tác với dữ liệu trong các cơ sở dữ liệu.
- **SQL là ngôn ngữ cơ sở dữ liệu phân tán :** Đối với các hệ quản trị cơ sở dữ liệu phân tán, mỗi một hệ thống sử dụng SQL để giao tiếp với các hệ thống khác trên mạng, gửi và nhận các yêu cầu truy xuất dữ liệu với nhau.
- **SQL là ngôn ngữ sử dụng cho các cổng giao tiếp cơ sở dữ liệu :** Trong một hệ thống mạng máy tính với nhiều hệ quản trị cơ sở dữ liệu khác nhau, SQL thường được sử dụng như là một chuẩn ngôn ngữ để giao tiếp giữa các hệ quản trị cơ sở dữ liệu SQL chuẩn bao gồm lệnh thường được sử dụng nhấ t khoảng 40 câu lệnh.

Các bảng phía dưới liệt kê danh sách các câu trong số các câu lệnh của SQL. Trong các hệ quản trị cơ sở dữ liệu khác nhau, mặc dù các câu lệnh đều có cùng dạng và cùng mục đích sử dụng song mỗi một hệ quản trị cơ sở dữ liệu có thể có một số thay đổi nào đó. Điều này đôi khi dẫn đến cú pháp chi tiết của các câu lệnh có thể sẽ khác nhau trong các hệ quản trị cơ sở dữ liệu khác nhau.



Các câu lệnh của SQL đều được bắt đầu bởi các từ lệnh, là một từ khoá cho biết chức năng của câu lệnh (chẳng hạn SELECT, DELETE, COMMIT). Sau từ lệnh là các mệnh đề của câu lệnh. Mỗi một mệnh đề trong câu lệnh cũng được bắt đầu bởi một từ khoá (chẳng hạn FROM, WHERE,...).

Câu lệnh:

```
SELECT masv, hodem, ten FROM sinhvien WHERE
malop='C24102'
```

dùng để truy xuất dữ liệu trong bảng SINHVIEN được bắt đầu bởi từ lệnh SELECT, trong câu lệnh bao gồm hai mệnh đề: mệnh đề FROM chỉ định tên của bảng cần truy xuất dữ liệu và mệnh đề WHERE chỉ định điều kiện truy vấn dữ liệu.

1.8. Quy tắc sử dụng tên trong SQL

Các đối tượng trong cơ sở dữ liệu dựa trên SQL được xác định thông qua tên của đối tượng. Tên của các đối tượng là duy nhất trong mỗi cơ sở dữ liệu. Tên được sử dụng nhiều nhất trong các truy vấn SQL và được xem là nền tảng trong cơ sở dữ liệu quan hệ là tên bảng và tên cột.

Trong các cơ sở dữ liệu lớn với nhiều người sử dụng, khi ta chỉ định tên của một bảng nào đó trong câu lệnh SQL, hệ quản trị cơ sở dữ liệu hiểu đó là tên của bảng do ta sở hữu (tức là bảng do ta tạo ra). Thông thường, trong các hệ quản trị cơ sở dữ liệu này cho phép những người dùng khác nhau tạo ra những bảng trùng tên với nhau mà không gây ra xung đột về tên. Nếu trong một câu lệnh SQL ta cần chỉ đến một bảng do một người dùng khác sở hữu (hiển nhiên là phải được phép) thì tên của bảng phải được viết sau tên của người sở hữu và phân cách với tên người sở hữu bởi dấu chấm:

tên_người_sở_hữu.tên_bảng

Một số đối tượng cơ sở dữ liệu khác (như khung nhìn, thủ tục, hàm), việc sử dụng tên cũng tương tự như đối với bảng.



Ta có thể sử dụng tên cột một cách bình thường trong các câu lệnh SQL bằng cách chỉ cần chỉ định tên của cột trong bảng. Tuy nhiên, nếu trong câu lệnh có liên quan đến hai cột trở lên có cùng tên trong các bảng khác nhau thì bắt buộc phải chỉ định thêm tên bảng trước tên cột; tên bảng và tên cột được phân cách nhau bởi dấu chấm.

Ví dụ dưới đây minh họa cho ta thấy việc sử dụng tên bảng và tên cột trong câu lệnh SQL

```
SELECT masv, hodem, ten, sinhvien.malop, tenlop FROM
dbo.sinhvien, dbo.lop WHERE sinhvien.malop = lop.malop
```

1.9. Kiểu dữ liệu

Chuẩn ANSI/ISO SQL cung cấp các kiểu dữ liệu khác nhau để sử dụng trong các cơ sở dữ liệu dựa trên SQL và trong ngôn ngữ SQL. Dựa trên cơ sở các kiểu dữ liệu do chuẩn ANSI/ISO SQL cung cấp, các hệ quản trị cơ sở dữ liệu thương mại hiện nay có thể sử dụng các dạng dữ liệu khác nhau trong sản phẩm của mình. Bảng 1.2 dưới đây liệt kê một số kiểu dữ liệu thông dụng được sử dụng trong SQL.

Một số kiểu dữ liệu thông dụng trong 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 -231 đến 231 - 1
▪ INT	Như kiểu Integer
▪ TINYTINT	Số nguyên có giá trị từ 0 đến 255.
▪ SMALLINT	Số nguyên có giá trị từ -215 đến 215 - 1
▪ BIGINT	Số nguyên có giá trị từ -263 đến 263-1
	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)
- 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
- TEXT Dữ liệu kiểu chuỗi với độ dài lớn (tối đa 2,147,483,647 ký tự)
- NTEXT Dữ liệu kiểu chuỗi với độ dài lớn và hỗ trợ UNICODE (tối đa 1,073,741,823 ký tự)

Câu lệnh dưới đây định nghĩa bảng với kiểu dữ liệu được qui định cho các cột trong bảng

```
CREATE TABLE NHANVIEN (
MANV NVARCHAR(10) NOT NULL,
HOTEN NVARCHAR(30) NOT NULL,
GIOITINH BIT,
NGAYSINH
SMALLDATETIME,
NOISINH NCHAR(50) ,
HSLUONG DECIMAL(4,2) ,
MADV INT )
```

Giá trị NULL

Một cơ sở dữ liệu là sự phản ánh của một hệ thống trong thế giới thực, do đó



các giá trị dữ liệu tồn tại trong cơ sở dữ liệu có thể không xác định được. Một giá trị không xác định được xuất hiện trong cơ sở dữ liệu có thể do một số nguyên nhân sau:

- Giá trị đó có tồn tại nhưng không biết.
- Không xác định được giá trị đó có tồn tại hay không.
- Tại một thời điểm nào đó giá trị chưa có nhưng rồi có thể sẽ có.
- Giá trị bị lỗi do tính toán (tràn số, chia cho không,...)

Những giá trị không xác định được biểu diễn trong cơ sở dữ liệu quan hệ bởi các giá trị NULL. Đây là giá trị đặc biệt và không nên nhầm lẫn với chuỗi rỗng (đối với dữ liệu kiểu chuỗi) hay giá trị không (đối với giá trị kiểu số). Giá trị NULL đóng một vai trò quan trọng trong các cơ sở dữ liệu và hầu hết các hệ quản trị cơ sở dữ liệu quan hệ hiện nay đều hỗ trợ việc sử dụng giá trị này.

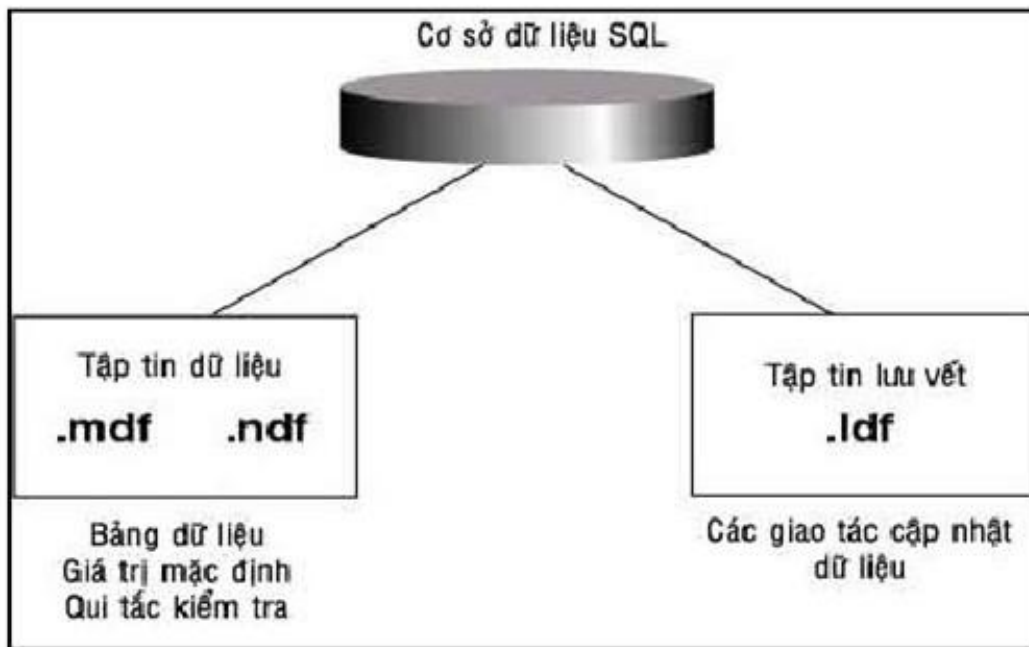
1.10. Các tập tin vật lý lưu trữ cơ sở dữ liệu

Mặc dù phải quản lý nhiều đối tượng bên trong cơ sở dữ liệu nhưng Microsoft SQL Server chỉ tổ chức hai loại tập tin để lưu trữ.

Một cơ sở dữ liệu trong Microsoft SQL Server tối thiểu sẽ dùng hai (2) tập tin vật lý để lưu trữ dữ liệu:

Datafile: dùng lưu trữ dữ liệu.

Transaction log file : dùng để lưu trữ các hành động thực hiện trên cơ sở dữ liệu trong quá trình sử dụng. Các hành động thực hiện trên CSDL gọi là các giao tác.



Các loại tập tin lưu trữ dữ liệu của SQL Sever 2000

Các loại tập tin lưu trữ dữ liệu của SQL Sever 2000

Các tập tin lưu trữ cơ sở dữ liệu bên trong Microsoft SQL Server được phân chia thành ba loại tập tin vật lý khác nhau:

Tập tin dữ liệu chính (Primary Data File) : Đây là tập tin chính dùng để lưu trữ các thông tin hệ thống của cơ sở dữ liệu và phần còn lại dùng lưu trữ một phần dữ liệu. Phần mở rộng của tập tin này thông thường là *.MDF.

Tập tin dữ liệu thứ yếu(Secondary Data Files) : Đây là tập tin dùng lưu trữ các đối tượng dữ liệu không nằm trong tập tin dữ liệu chính. Loại tập tin này không bắt buộc phải có khi tạo mới cơ sở dữ liệu. Phần mở rộng của tập tin này thông thường là *.NDF.

Tập tin lưu vết (Log Files): Đây là tập tin dùng lưu vết các giao tác – là những hành động cập nhật dữ liệu (thêm, sửa, xóa) vào các bảng do người sử dụng tác động trên cơ sở dữ liệu. Tập tin sẽ này hỗ trợ cho phép các bạn có thể hủy bỏ (rollback) các thao tác cập nhật dữ liệu đã được thực hiện hay giúp SQL Server phục hồi dữ liệu



trong các trường hợp gặp sự cố như mất điện,... Phần mở rộng của tập tin này thông thường là *.LDF.

Kết chương

Như vậy, SQL (viết tắt của *StructuredQueryLanguage*) là hệ thống ngôn ngữ được sử dụng cho các hệ quản trị cơ sở dữ liệu quan hệ. Thông qua SQL có thể thực hiện được các thao tác trên cơ sở dữ liệu như định nghĩa dữ liệu, thao tác dữ liệu, điều khiển truy cập, quản lý toàn vẹn dữ liệu... SQL là một thành phần quan trọng và không thể thiếu trong hệ quản trị cơ sở dữ liệu quan hệ.

SQL ra đời nhằm sử dụng cho các cơ sở dữ liệu theo mô hình quan hệ. Trong một cơ sở dữ liệu quan hệ, dữ liệu được tổ chức và lưu trữ trong các bảng. Mỗi một bảng là một tập hợp bao gồm các dòng và các cột; mỗi một dòng là một bản ghi và mỗi một cột tương ứng với một trường, tập các tên cột cùng với kiểu dữ liệu và các tính chất khác tạo nên cấu trúc của bảng, tập các dòng trong bảng chính là dữ liệu của bảng.

Các bảng trong một cơ sở dữ liệu có mối quan hệ với nhau. Các mối quan hệ được biểu diễn thông qua khoá chính và khoá ngoài của các bảng. Khoá chính của bảng là tập một hoặc nhiều cột có giá trị duy nhất trong bảng và do đó giá trị của nó xác định duy nhất một dòng dữ liệu trong bảng. Một khoá ngoài là một tập một hoặc nhiều cột có giá trị được xác định từ khoá chính của các bảng khác.

II. Ngôn ngữ định nghĩa dữ liệu

2.1. Ngôn ngữ định nghĩa dữ liệu

Các câu lệnh SQL đã đề cập đến trong chương 2 được sử dụng nhằm thực hiện các thao tác bổ sung, cập nhật, loại bỏ và xem dữ liệu. Nhóm các câu lệnh này được gọi là ngôn ngữ thao tác dữ liệu (DML). Trong chương này, chúng ta sẽ tìm hiểu nhóm các câu lệnh được sử dụng để định nghĩa và quản lý các đối tượng CSDL như bảng, khung nhìn, chỉ mục,... và được gọi là ngôn ngữ định nghĩa dữ liệu (DDL).



Về cơ bản, ngôn ngữ định nghĩa dữ liệu bao gồm các lệnh:

- CREATE: định nghĩa và tạo mới đối tượng CSDL.
- ALTER: thay đổi định nghĩa của đối tượng CSDL.
- DROP: Xóa đối tượng CSDL đã có.

2.2. Tạo CSDL

Sau khi có khái niệm về cách thức tổ chức các tập tin vật lý để lưu trữ dữ liệu trong Microsoft SQL Server, chúng ta sẽ tự tạo một cơ sở dữ liệu cho riêng mình nhằm lưu trữ các dữ liệu riêng biệt và đưa vào khai thác các dữ liệu đó. Cách dễ nhất để các bạn tạo ra một cơ sở dữ liệu là sử dụng tiện ích Enterprise Manager. Chỉ những người với vai trò là quản trị hệ thống (sysadmin) thì mới có thể tạo lập cơ sở dữ liệu. Do đó các bạn có thể đăng nhập vào với tên tài khoản người dùng là sa để thực hiện việc tạo cơ sở dữ liệu mới cho ứng dụng của mình. Trước khi giới thiệu từng bước tạo lập cơ sở dữ liệu, phần kế tiếp mà chúng tôi muốn trình bày là các thuộc tính của một cơ sở dữ liệu trong Microsoft SQL Server. Các thuộc tính nhằm giúp các bạn hiểu rõ thêm về bên trong cơ sở dữ liệu của Microsoft SQL Server, chúng gồm có:

Tên cơ sở dữ liệu(database name) : là duy nhất trong một Microsoft SQL Server, độ dài tối đa là 123 ký tự. Theo chúng tôi các bạn nên đặt tên cơ sở dữ liệu gợi nhớ.

Thí dụ:

QLBanhang (Quản lý bán hàng), QlHocsinh (Quản lý học sinh)...

Vị trí tập tin (File location) : là tên và đường dẫn vật lý của các loại tập tin dữ liệu dùng để lưu trữ cơ sở dữ liệu của Microsoft SQL Server. Thông thường các tập tin này sẽ được lưu tại thư mục C:\MSSQL\DATA.

Tên tập tin (File name) : là tên logic của mỗi loại tập tin dữ liệu tương ứng mà hệ thống Microsoft SQL Server dùng để quản lý bên trong. Tương ứng mỗi loại tập tin dữ liệu sẽ có một tên tập tin riêng biệt.



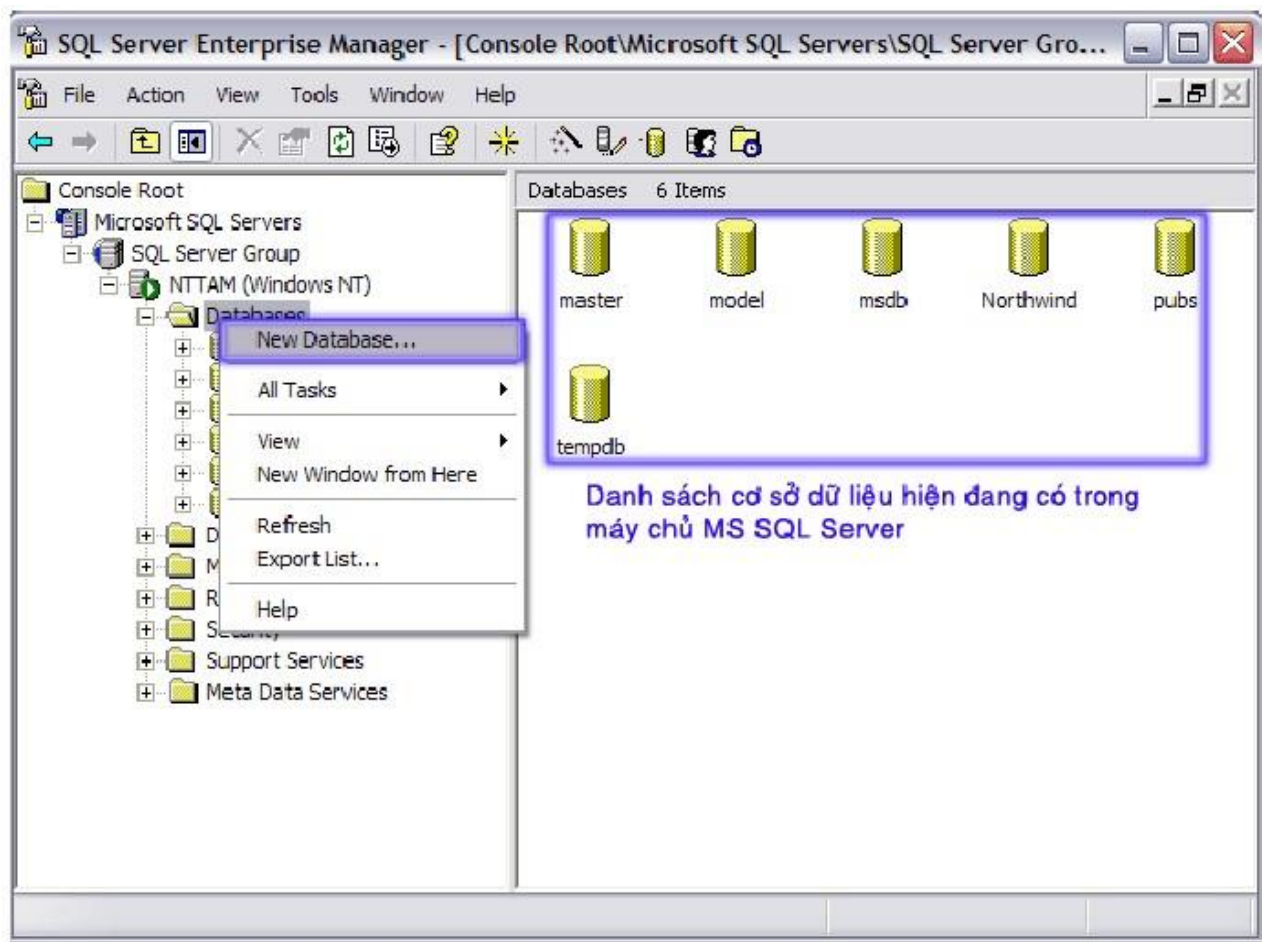
Kích thước ban đầu(Initial size) : là kích thước khởi tạo của tập tin dữ liệu khi cơ sở dữ liệu mới được tạo lập. Đơn vị tính là MegaByte (MB). Thông thường kích thước ban đầu của một cơ sở dữ liệu mới tối thiểu phải bằng với kích thước của cơ sở dữ liệu Model, bởi vì Microsoft SQL Server sẽ lấy cơ sở dữ liệu Model làm khuôn dạng mẫu khi hình thành một cơ sở dữ liệu mới.

Việc tăng trưởng kích thước tập tin dữ liệu (File growth) : là các qui định cho việc tăng trưởng tự động kích thước tập tin dữ liệu, bởi vì các dữ liệu sẽ được lưu trữ ngày càng nhiều hơn so với kích thước ban đầu khi tạo lập. Việc tăng trưởng sẽ tự động làm tăng kích thước tập tin dữ liệu theo từng MB hoặc theo tỷ lệ phần trăm (by percent) của kích thước hiện hành khi các dữ liệu bên trong Microsoft SQL Server lưu trữ gần đầy so với kích thước tập tin vật lý hiện thời. Mặc định kích thước tập tin dữ liệu sẽ được tăng tự động 10% khi dữ liệu lưu trữ gần đầy.

Kích thước tối đa tập tin dữ liệu (Maximum file size) : là việc qui định sự tăng trưởng tự động kích thước của các tập tin dữ liệu nhưng có giới hạn (restrict file growth) đến MB nào đó hoặc là không có giới hạn (un-restrict file growth). Trong trường hợp nếu các bạn chọn có giới hạn kích thước của tập tin dữ liệu thì chúng ta phải biết tự thêm vào các tập tin dữ liệu mới khi dữ liệu lưu trữ đã bằng với kích thước tối đa của tập tin dữ liệu. Các tập tin dữ liệu mới này chính là loại tập tin thứ yếu (Secondary data file) và chúng ta có thể lưu trữ các tập tin vật lý này tại các đĩa cứng khác có bên trong Microsoft SQL Server. Đây cũng là một trong nét đặc trưng của mô hình cơ sở dữ liệu phân tán (distributed database). Đối với các CSDL thực tế, việc xác định các tham số về kích thước ban đầu rất quan trọng vì nhiều lý do. Để đảm bảo có đủ không gian lưu trữ dữ liệu, bạn cần dành trước cho CSDL phòng khi những ứng hay CSDL khác sử dụng hết đĩa cứng. CSDL có kích thước nhỏ cũng sẽ ảnh hưởng tới tốc độ do SQL Server cần phải thực hiện nhiều lần thao tác mở rộng kích thước tập tin CSDL khi có dữ liệu thêm mới. Ngoài ra, đa số các dữ liệu trong CSDL thực tế theo thời gian không thể xóa bỏ mà cần phải lưu trữ (backup) lại trước.

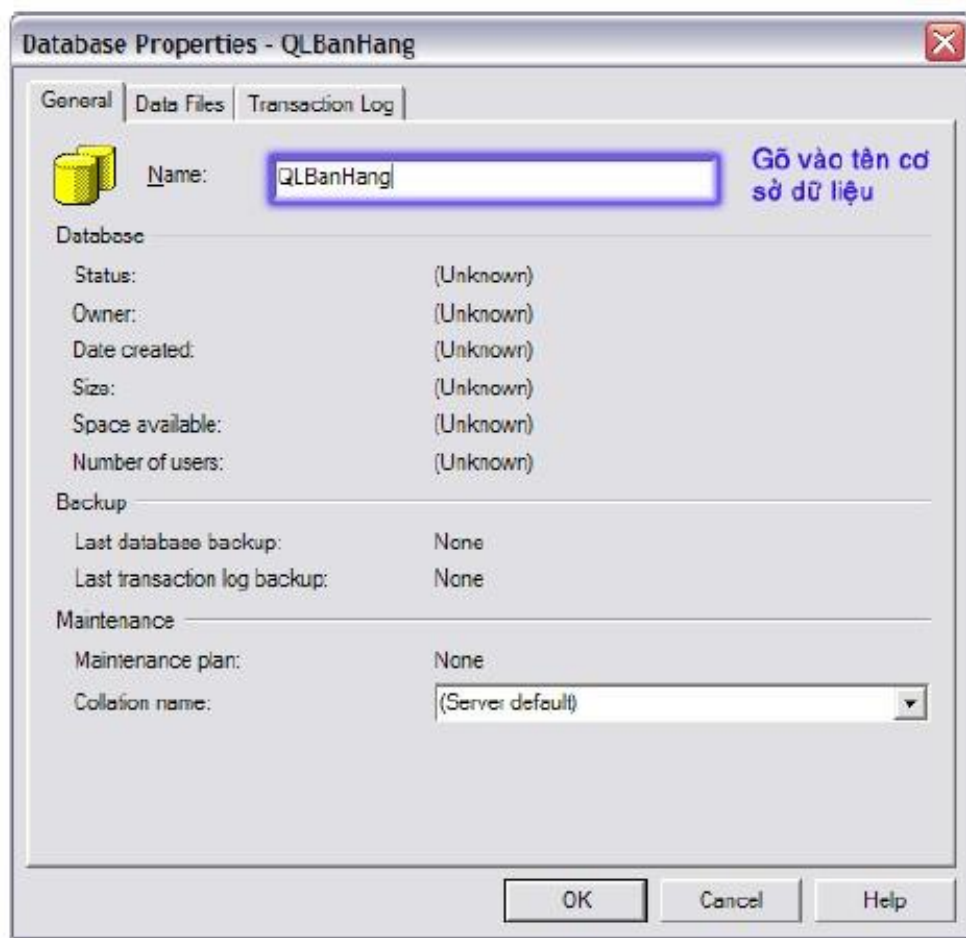
Việc lưu trữ và phục hồi (restore) dữ liệu cũng ảnh hưởng bởi kích thước các tập tin do chúng phải đủ nhỏ để lưu trên các đĩa CD- ROM hay băng từ. Các bước mà chúng tôi mô tả bên dưới sẽ giúp các bạn tạo ra một cơ sở dữ liệu mới bằng tiện ích Enterprise Manager.

Bước 1: Khởi động ứng dụng Enterprise Manager, chọn một (1) Microsoft SQL Server đã được đăng ký quản trị trước đó. Chọn chức năng New Database... trong thực đơn tắt sau khi nhấn chuột phải trên đối tượng Database



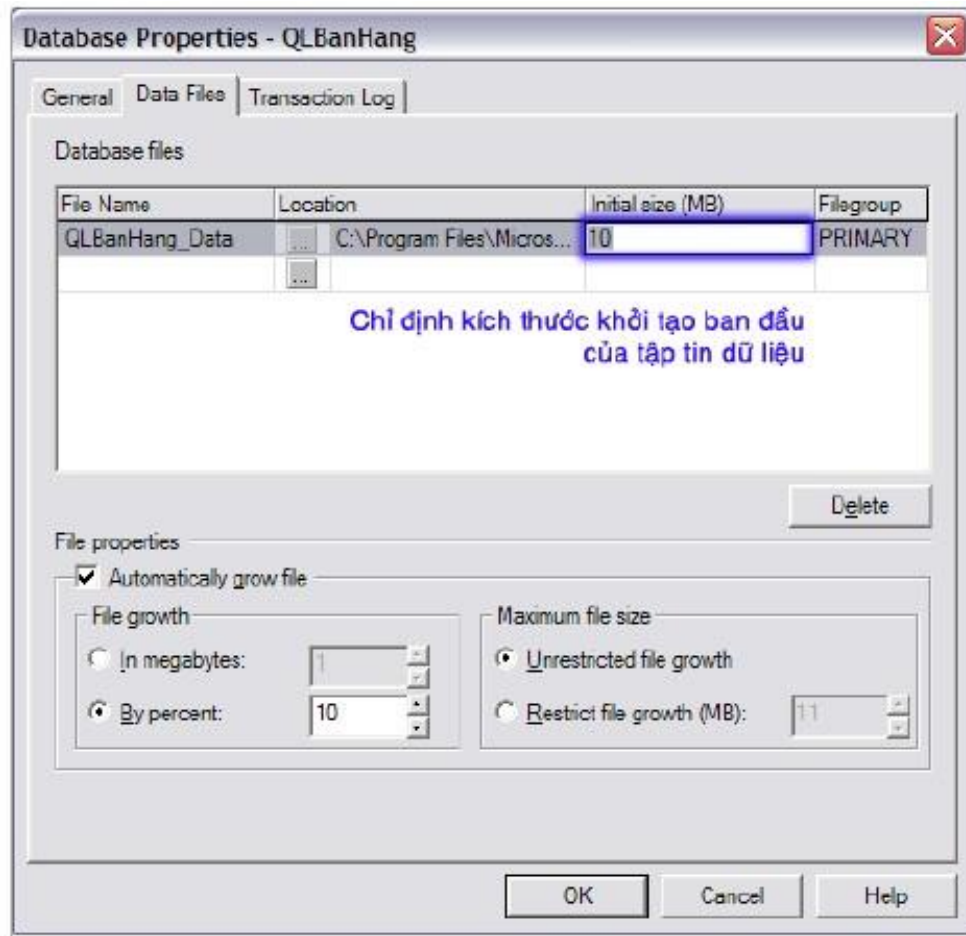
Chọn chức năng New Database

Bước 2: Trong màn hình các thuộc tính của cơ sở dữ liệu (Database Properties) tại trang General gõ vào tên cơ sở dữ liệu muốn tạo mới.



Trang General với các thuộc tính của CSDL

Bước 3: Trong màn hình các thuộc tính của cơ sở dữ liệu (Database Properties) tại trang Data Files, chỉ định kích thước ban đầu khi khởi tạo của tập tin dữ liệu chính, kế tiếp thay đổi các thuộc tính khác (nếu cần). Chuyển sang trang Transaction Log để thay đổi các thuộc tính của tập tin lưu vết theo cách tương tự.



Các thuộc tính trong trang Transaction Log

Tùy thuộc vào kích thước của cơ sở dữ liệu mà thời gian thực hiện tạo cơ sở dữ liệu sẽ nhanh hoặc lâu. Ngoài ra chúng ta còn có thể tạo mới một cơ sở dữ liệu bằng câu lệnh CREATE DATABASE được thực hiện trong tiện ích Query Analyzer. Các thành phần trong câu lệnh này hoàn toàn giống với các thuộc tính của cơ sở dữ liệu mà chúng tôi đã giới thiệu trong phần trên.

Để tạo ra một cơ sở dữ liệu có tên QLBanHang với kích thước ban đầu lúc khởi tạo của tập tin dữ liệu chính là 50MB, tự động tăng kích thước lên 10% khi dữ liệu bị đầy, kích thước tăng trưởng tập tin dữ liệu tối đa không quá 200MB. Và tập tin lưu vết với kích thước ban đầu lúc khởi tạo là 10MB, tự động tăng kích thước tập tin lên 5MB khi dữ liệu bị đầy, kích thước tăng trưởng tập tin không giới hạn. Các



bạn sẽ thực hiện câu lệnh CREATE DATABASE như sau:

```
CREATE DATABASE QLBanHang
ON PRIMARY
( NAME=QLBanHang_Data,
  FILENAME='C:\MSSQL7\DATA\QLBANHANG.MDF',
  SIZE=50MB,
  MAXSIZE=200MB,
  FILEGROWTH=10%)
LOG ON
( NAME=QLBanHang_Log,
  FILENAME='C:\MSSQL7\DATA\SAMPLE.LDF',
  SIZE=10MB,
  MAXSIZE=UNLIMITED,
  FILEGROWTH=5MB)
```

2.3. Xoá cơ sở dữ liệu đã có

Một cơ sở dữ liệu sau khi tạo xong sau một thời gian dài mà các bạn không còn khai thác dữ liệu bên trong đó thì các bạn có thể hủy bỏ để làm cho dung lượng đĩa trống được tăng lên. Tuy nhiên phải chắc rằng các thông tin dữ liệu trong cơ sở dữ liệu mà các bạn dự định xóa sẽ không còn hữu ích về sau nữa. Bởi vì chúng ta không thể khôi phục khi đã xóa. Để hủy bỏ cơ sở dữ liệu trong Microsoft SQL Server chúng ta có nhiều cách thực hiện: sử dụng câu lệnh DROP DATABASE, nhấn phím Delete hoặc nhấn chuột trên biểu tượng Delete và xác định đồng ý hủy bỏ cơ sở dữ liệu đã chọn trong tiện ích Enterprise Manager. Để hủy bỏ cơ sở dữ liệu QLBanHang, thực hiện câu lệnh DROP DATABASE như sau: DROP DATABASE QLBanHang. Hoặc nhấn phím Delete trên tên của cơ sở dữ liệu này trong tiện ích Enterprise Manager và xác nhận là đồng ý (chọn Yes) để hủy bỏ cơ sở dữ liệu QLBanHang. Hệ thống Microsoft SQL Server không cho người sử dụng có thể hủy bỏ các cơ sở dữ liệu hệ thống như là: Master, Model, Tempdb bởi vì các cơ sở dữ liệu luôn được hệ thống Microsoft SQL Server sử dụng. Ngoài ra để hủy bỏ một cơ sở dữ liệu thành



công thì phải đảm bảo không còn người sử dụng nào đang truy cập vào cơ sở dữ liệu đó. Trong trường hợp khi thực hiện hủy bỏ cơ sở dữ liệu đang còn người sử dụng truy cập thì hệ thống sẽ hiển thị thông báo bên dưới.

2.4. Tạo bảng dữ liệu

Như đã nói đến ở chương 1, bảng dữ liệu là cấu trúc có vai trò quan trọng nhất trong cơ sở dữ liệu quan hệ. Toàn bộ dữ liệu của cơ sở dữ liệu được tổ chức trong các bảng, những bảng này có thể là những bảng hệ thống được tạo ra khi tạo lập cơ sở dữ liệu, và cũng có thể là những bảng do người sử dụng định nghĩa.

MAKHOA	TENKHOA	DIENTHOAI	TRUONGKHOA
DHT01	Khoa Toán cơ - Tin học	054822407	Trần Lộc Hùng
DHT02	Khoa Công nghệ thông tin	054826767	Nguyễn Mậu Hân
DHT03	Khoa Vật lý	054823462	Trương Văn Chương
DHT04	Khoa Hoá học	054823951	Nguyễn Văn Hợp
DHT05	Khoa Sinh học	054822934	Đỗ Quý Hai
DHT06	Khoa Địa lý - Địa chất	054823837	NULL
DHT07	Khoa Ngữ văn	054821133	Hoàng Tất Thắng
DHT08	Khoa Lịch sử	054823833	Nguyễn Văn Mạnh
DHT09	Khoa Mác - Lê Nin	054825698	Đoàn Đức Hiếu
DHT10	Khoa Luật	054821135	Đoàn Đức Lương

Hình 2.4: Bảng trong CSDL quan hệ

Trong các bảng, dữ liệu được chức dưới dạng các dòng và cột. Mỗi một dòng là một bản ghi duy nhất trong bảng và mỗi một cột là một trường. Các bảng trong cơ sở dữ liệu được sử dụng để biểu diễn thông tin, lưu giữ dữ liệu về các đối tượng trong thế giới thực và/hoặc mối quan hệ giữa các đối tượng. Bảng trong hình 3.1 bao gồm 10 bản ghi và 4 trường là MAKHOA, TENKHOA, DIENTHOAI và TRUONGKHOA.

Câu lệnh CREATE TABLE được sử dụng để định nghĩa một bảng dữ liệu mới trong cơ sở dữ liệu. Khi định nghĩa một bảng dữ liệu mới, ta cần phải xác định được các yêu cầu sau đây:



- Bảng mới được tạo ra sử dụng với mục đích gì và có vai trò như thế nào trong cơ sở dữ liệu.
- Cấu trúc của bảng bao gồm những trường (cột) nào, mỗi một trường có ý nghĩa như thế nào trong việc biểu diễn dữ liệu, kiểu dữ liệu của mỗi trường là gì và trường đó có cho phép nhận giá trị NULL hay không.
- Những trường nào sẽ tham gia vào khóa chính của bảng. Bảng có quan hệ với những bảng khác hay không và nếu có thì quan hệ như thế nào.
- Trên các trường của bảng có tồn tại những ràng buộc về khuôn dạng, điều kiện hợp lệ của dữ liệu hay không; nếu có thì sử dụng ở đâu và như thế nào.

2.5. Tạo cấu trúc bảng dữ liệu bằng EM

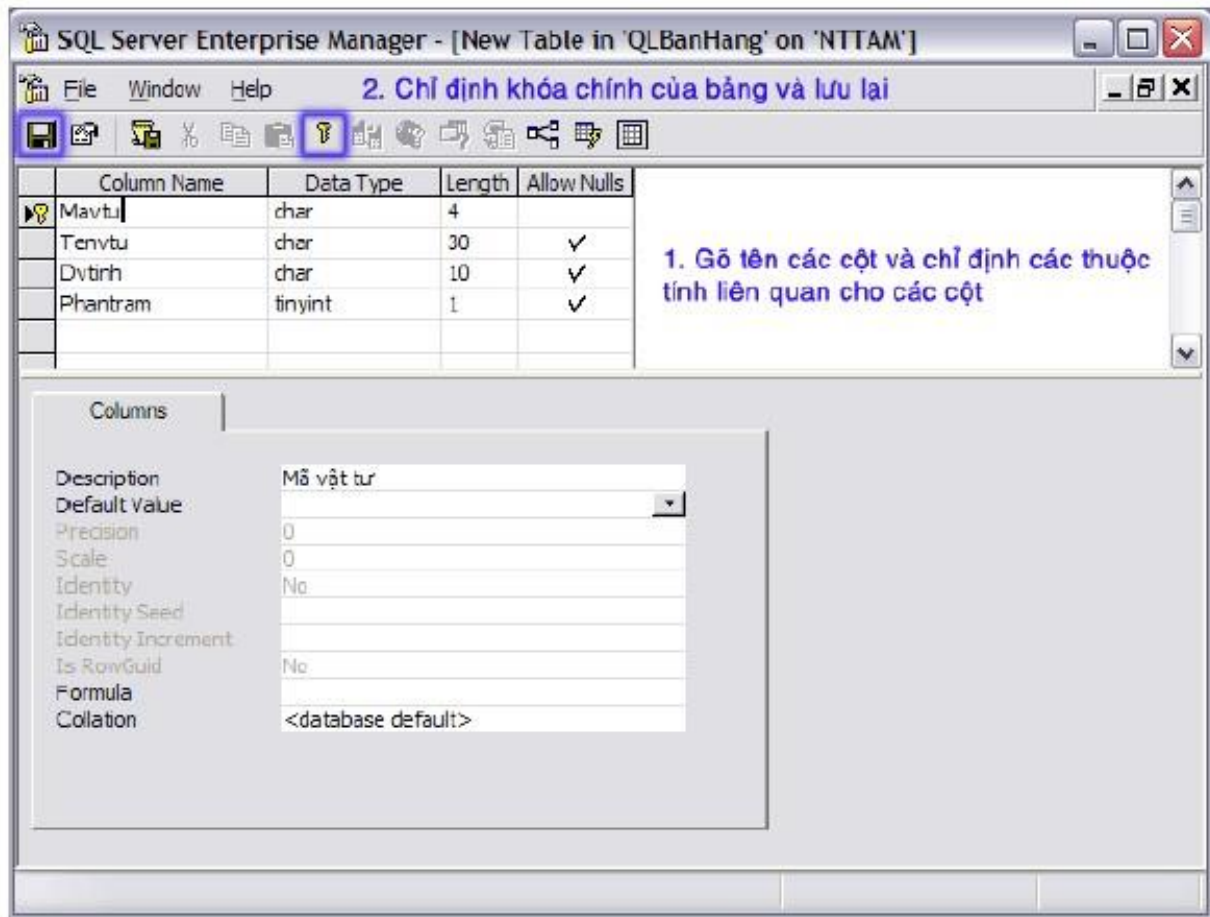
Sau khi xem xét và hiểu được các thuộc tính liên quan đến cấu trúc của bảng, trong phần này chúng tôi sẽ hướng dẫn các bạn các cách để tạo cấu trúc bảng dữ liệu mới. Để tạo cấu trúc bảng chúng tôi hướng dẫn các bạn hai (2) cách thực hiện. Đầu tiên là tạo cấu trúc bảng bằng tiện ích Enterprise Manager. Kể từ bây giờ chúng tôi xem như các bạn đã đăng ký quản trị một Microsoft SQL Server và bên trong Microsoft SQL Server này, cơ sở dữ liệu quản lý bán hàng (QLBanHang) đã được tạo lập. Các bảng dữ liệu và những đối tượng khác ở các phần trình bày kế tiếp sẽ được tạo ra bên trong cơ sở dữ liệu QLBanHang này. Các bước thực hiện việc tạo bảng dữ liệu trong Enterprise Manager như sau:

Bước 1: Trong ứng dụng Enterprise Manager, mở rộng cơ sở dữ liệu để thấy các đối tượng bên trong. Nhấn chuột phải trên đối tượng Tables, chọn chức năng New Table... trong thực đơn tắt.



Tạo bảng dữ liệu

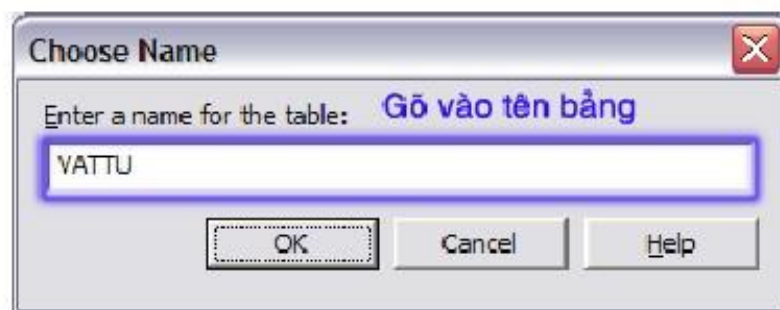
Bước 2: Trong màn hình thiết kế cấu trúc bảng (design table), lần lượt gõ vào tên các cột bên trong bảng, chọn lựa các kiểu dữ liệu tương ứng thích hợp và chỉ định các thuộc tính cần thiết cho các cột bên trong bảng.



Màn hình xây dựng cấu trúc bảng

Bước 3: Định nghĩa khóa chính cho bảng và lưu lại cấu trúc bảng vừa định nghĩa. Đóng màn hình thiết kế cấu trúc bảng lại để kết thúc quá trình tạo cấu trúc bảng bằng tiện ích

EM



Màn hình chỉ định tên bảng mới



2.6. Tạo cấu trúc bảng bằng T-SQL

Câu lệnh CREATE TABLE có cú pháp như sau:

```
CREATE TABLE tên_bảng ( tên_cột thuộc_tính_cột
các_ràng_buộc [... ,tên_cột_n thuộc_tính_cột_n
các_ràng_buộc_cột_n] [...các_ràng_buộc_trên_bảng] )
```

Tên_bảng Tên của bảng cần tạo. Tên phải tuân theo qui tắc định danh và không được vượt quá 128 ký tự. **tên_cột** Là tên của cột (trường) cần định nghĩa, tên cột phải tuân theo qui tắc định danh và không được trùng nhau trong mỗi một bảng. Mỗi một bảng phải có ít nhất một cột. Nếu bảng có nhiều cột thì định nghĩa của các cột (tên cột, thuộc tính và các ràng buộc) phải phân cách nhau bởi dấu phẩy.

Thuộc_tính_cột Mỗi một cột trong một bảng ngoài tên cột còn có các thuộc tính bao gồm:

- Kiểu dữ liệu của cột. Đây là thuộc tính bắt buộc phải có đối với mỗi cột.
- Giá trị mặc định của cột: là giá trị được tự động gán cho cột nếu như người sử dụng không nhập dữ liệu cho cột một cách tường minh. Mỗi một cột chỉ có thể có nhiều nhất một giá trị mặc định.
- Cột có tính chất IDENTITY hay không? tức là giá trị của cột có được tự động tăng mỗi khi có bản ghi mới được bổ sung hay không. Tính chất này chỉ có thể sử dụng đối với các trường kiểu số.
- Cột có chấp nhận giá trị NULL hay không

Đây là phần ví dụ , xác định khoảng của ví dụ đó rồi chọn style

Khai báo dưới đây định nghĩa cột STT có kiểu dữ liệu là int và cột có tính chất IDENTITY:

```
stt INT IDENTITY
```

hay định nghĩa cột NGÀY có kiểu datetime và không cho phép chấp nhận giá trị

NULL:

```
ngay DATETIME NOT NULL
```



và định nghĩa cột *SOLUONG* kiểu *int* và có giá trị mặc định là 0:

```
soluong INT DEFAULT (0)
```

Các_ràng_buộc: Các ràng buộc được sử dụng trên mỗi cột hoặc trên bảng nhằm các mục đích sau:

Quy định khuôn dạng hay giá trị dữ liệu được cho phép trên cột (chẳng hạn qui định tuổi của một học sinh phải lớn hơn 6 và nhỏ hơn 20, số điện thoại phải là một chuỗi bao gồm 6 chữ số,...). Những ràng buộc kiểu này được gọi là ràng buộc CHECK Đảm bảo tính toàn vẹn dữ liệu trong một bảng và toàn vẹn tham chiếu giữa các bảng trong cơ sở dữ liệu. Những loại ràng buộc này nhằm đảm bảo tính đúng của dữ liệu như: số chứng minh nhân dân của mỗi một người phải duy nhất, nếu sinh viên học một lớp nào đó thì lớp đó phải tồn tại,... Liên quan đến những loại ràng buộc này bao gồm các ràng buộc PRIMARY KEY (khóa chính), UNIQUE (khóa dự tuyển) và FOREIGN KEY (khóa ngoài) Các loại ràng buộc này sẽ được trình bày chi tiết hơn ở phần sau.

Câu lệnh dưới đây định nghĩa bảng NHANVIEN với các trường MANV (mã nhân viên), HOTEN (họ và tên), NGAYSINH (ngày sinh của nhân viên), DIENTHOAI (điện thoại) và HSLUONG (hệ số lương)

```
CREATE TABLE nhanvien (
manv NVARCHAR(10) NOT NULL, hoten
NVARCHAR(50) NOT NULL,
ngaysinh DATETIME NULL,
dienthoai NVARCHAR(10) NULL,
hsluong DECIMAL(3,2)
DEFAULT (1.92) )
```

Trong câu lệnh trên, trường MANV và HOTEN của bảng NHANVIEN không được NULL (tức là bắt buộc phải có dữ liệu), trường NGAYSINH và DIENTHOAI sẽ nhận giá trị NULL nếu ta không nhập dữ liệu cho chúng còn trường HSLUONG sẽ nhận giá trị mặc định là 1.92 nếu không được nhập dữ liệu.



Nếu ta thực hiện các câu lệnh dưới đây sau khi thực hiện câu lệnh trên để bổ sung dữ liệu cho bảng NHANVIEN

```
INSERT INTO nhanvien VALUES ('NV01', 'Le Van A', '2/4/75', '886963', 2.14)
```

```
INSERT INTO nhanvien (manv, hoten) VALUES ('NV02', 'Mai Thi B')
```

```
INSERT INTO nhanvien (manv, hoten, dienthoai) VALUES ('NV03', 'Tran Thi C', '849290')
```

Ta sẽ có được dữ liệu trong bảng NHANVIEN như sau:

MANV	HOTEN	NGAYSINH	DIENTHOAI	HSLUONG
NV01	Le Van A	1975-02-04 00:00:00.000	886963	2.14
NV02	Mai Thi B	NULL	NULL	1.92
NV03	Tran Thi C	NULL	849290	1.92

Ràng buộc CHECK

Ràng buộc CHECK được sử dụng nhằm chỉ định điều kiện hợp lệ đối với dữ liệu. Mỗi khi có sự thay đổi dữ liệu trên bảng (INSERT, UPDATE), những ràng buộc này sẽ được sử dụng nhằm kiểm tra xem dữ liệu mới có hợp lệ hay không.

Ràng buộc CHECK được khai báo theo cú pháp như sau:

```
[CONSTRAINT tên_ràng_buộc] CHECK (điều kiện)
```

Trong đó, điều_kiện là một biểu thức logic tác động lên cột nhằm qui định giá trị hoặc khuôn dạng dữ liệu được cho phép. Trên mỗi một bảng cũng như trên mỗi một cột có thể có nhiều ràng buộc CHECK.

Câu lệnh dưới đây tạo bảng DIEMTOTNGHIEP trong đó qui định giá trị của cột DIEMVAN và DIEMTOAN phải lớn hơn hoặc bằng 0 và nhỏ hơn hoặc bằng 10

```
CREATE TABLE diemtotnghiep (
Hoten NVARCHAR(30) NOT NULL,
Ngaysinh DATETIME,
Diemvan DECIMAL(4,2) CONSTRAINT chk_diemvan
```




```
CHECK(diemvan>=0 AND diemvan<=10) ,
Diemtoan DECIMAL(4,2) CONSTRAINT chk_diemtoan
CHECK(diemtoan>=0 AND diemtoan<=10) )
```

Như vậy, với định nghĩa như trên của bảng DIEMTOTNGHIEP, các câu lệnh dưới đây là hợp lệ:

```
INSERT INTO diemtotnghiep(hoten,diemvan,diemtoan)
VALUES('Le Thanh Hoang',9.5,2.5)
INSERT INTO diemtotnghiep(hoten,diemvan)
VALUES('Hoang Thi Mai',2.5) )
```

còn câu lệnh dưới đây là không hợp lệ:

```
INSERT INTO diemtotnghiep(hoten,diemvan,diemtoan)
VALUES('Tran Van Hanh',6,10.5)
```

do cột DIEMTOAN nhận giá trị 10.5 không thỏa mãn điều kiện của ràng buộc. Trong ví dụ trên, các ràng buộc được chỉ định ở phần khai báo của mỗi cột. Thay vì chỉ định ràng buộc trên mỗi cột, ta có thể chỉ định các ràng buộc ở mức bảng bằng cách khai báo các ràng buộc sau khi đã khai báo xong các cột trong bảng.

Câu lệnh

```
CREATE TABLE lop ( malop NVARCHAR(10) NOT NULL , tenlop
NVARCHAR(30) NOT NULL , khoa SMALLINT NULL , hedaotao
NVARCHAR(25) NULL CONSTRAINT chk_lop_hedaotao CHECK
(hedaotao IN ('chính quy','tại chức')), namnhaphoc INT
NULL CONSTRAINT chk_lop_namnhaphoc CHECK
(namnhaphoc<=YEAR(GETDATE())) , makhoa NVARCHAR(5)
```

có thể được viết lại như sau:

```
CREATE TABLE lop ( malop NVARCHAR(10) NOT NULL , tenlop
NVARCHAR(30) NOT NULL , khoa SMALLINT NULL , hedaotao
NVARCHAR(25) NULL, namnhaphoc INT NULL , makhoa
NVARCHAR(5), CONSTRAINT chk_lop CHECK
(namnhaphoc<=YEAR(GETDATE())) AND hedaotao IN ('chính
```



```
quy', 'tại chức')) )
```

Ràng buộc PRIMARY KEY

Ràng buộc PRIMARY KEY được sử dụng để định nghĩa khoá chính của bảng. Khoá chính của một bảng là một hoặc một tập nhiều cột mà giá trị của chúng là duy nhất trong bảng. Hay nói cách khác, giá trị của khoá chính sẽ giúp cho ta xác định được duy nhất một dòng (bản ghi) trong bảng dữ liệu. Mỗi một bảng chỉ có thể có duy nhất một khoá chính và bản thân khoá chính không chấp nhận giá trị NULL. Ràng buộc PRIMARY KEY là cơ sở cho việc đảm bảo tính toàn vẹn thực thể cũng như toàn vẹn tham chiếu. Để khai báo một ràng buộc PRIMARY KEY, ta sử dụng cú pháp như sau:

```
[CONSTRAINT tên_ràng_buộc] PRIMARY KEY  
[(danh_sách_cột)]
```

Nếu khoá chính của bảng chỉ bao gồm đúng một cột và ràng buộc PRIMARY KEY được chỉ định ở mức cột, ta không cần thiết phải chỉ định danh sách cột sau từ khoá PRIMARY KEY. Tuy nhiên, nếu việc khai báo khoá chính được tiến hành ở mức bảng (sử dụng khi số lượng các cột tham gia vào khoá là từ hai trở lên) thì bắt buộc phải chỉ định danh sách cột ngay sau từ khoá PRIMARY KEY và tên các cột được phân cách nhau bởi dấu phẩy.

Câu lệnh dưới đây định nghĩa bảng SINHVIEN với khoá chính là MASV

```
CREATE TABLE sinhvien (  
masv NVARCHAR(10) CONSTRAINT pk_sv_masv PRIMARY KEY,  
hodem NVARCHAR(25) NOT NULL ,  
ten NVARCHAR(10) NOT NULL ,  
ngaysinh DATETIME,  
gioitinh BIT,  
noisinh NVARCHAR(255),  
malop NVARCHAR(10) )
```

Với bảng vừa được tạo bởi câu lệnh ở trên, nếu ta thực hiện câu lệnh:



```
INSERT INTO sinhvien (masv, hodem, ten, gioitinh, malop)
VALUES ('0261010001', 'Lê Hoàng Phương', 'Anh', 0, 'C26101')
```

một bản ghi mới sẽ được bổ sung vào bảng này. Nhưng nếu ta thực hiện tiếp câu lệnh:

```
INSERT INTO sinhvien (masv, hodem, ten, gioitinh, malop)
VALUES ('0261010001', 'Lê Huy', 'Đan', 1, 'C26101')
```

thì câu lệnh này sẽ bị lỗi do trùng giá trị khoá với bản ghi đã có.

Câu lệnh dưới đây tạo bảng DIEMTHI với khoá chính là tập bao gồm hai cột MAMONHOC và MASV

```
CREATE TABLE diemthi ( ma NVARCHAR(10) NOTNULL,
Diemlan1NUMERIC(4, 2), Diemlan2NUMERIC(4, 2),
CONSTRAINT pk_diemthi PRIMARY KEY (mamonhoc, masv) )
```

Lưu ý :

- Mỗi một bảng chỉ có thể có nhiều nhất một ràng buộc PRIMARY KEY.
- Một khoá chính có thể bao gồm nhiều cột nhưng không vượt quá 16 cột.

Ràng buộc UNIQUE

Trên một bảng chỉ có thể có nhiều nhất một khoá chính nhưng có thể có nhiều cột hoặc tập các cột có tính chất như khoá chính, tức là giá trị của chúng là duy nhất trong bảng. Tập một hoặc nhiều cột có giá trị duy nhất và không được chọn làm khoá chính được gọi là khoá phụ (khoá dự tuyển) của bảng. Như vậy, một bảng chỉ có nhiều nhất một khoá chính nhưng có thể có nhiều khoá phụ.

Ràng buộc UNIQUE được sử dụng trong câu lệnh CREATE TABLE để định nghĩa khoá phụ cho bảng và được khai báo theo cú pháp sau đây:

```
[CONSTRAINT tên_ràng_buộc] UNIQUE [(danh_sách_cột)]
```

Giả sử ta cần định nghĩa bảng LOP với khoá chính là cột MALOP nhưng đồng



thời lại không cho phép các lớp khác nhau được trùng tên lớp với nhau, ta sử dụng câu lệnh như sau:

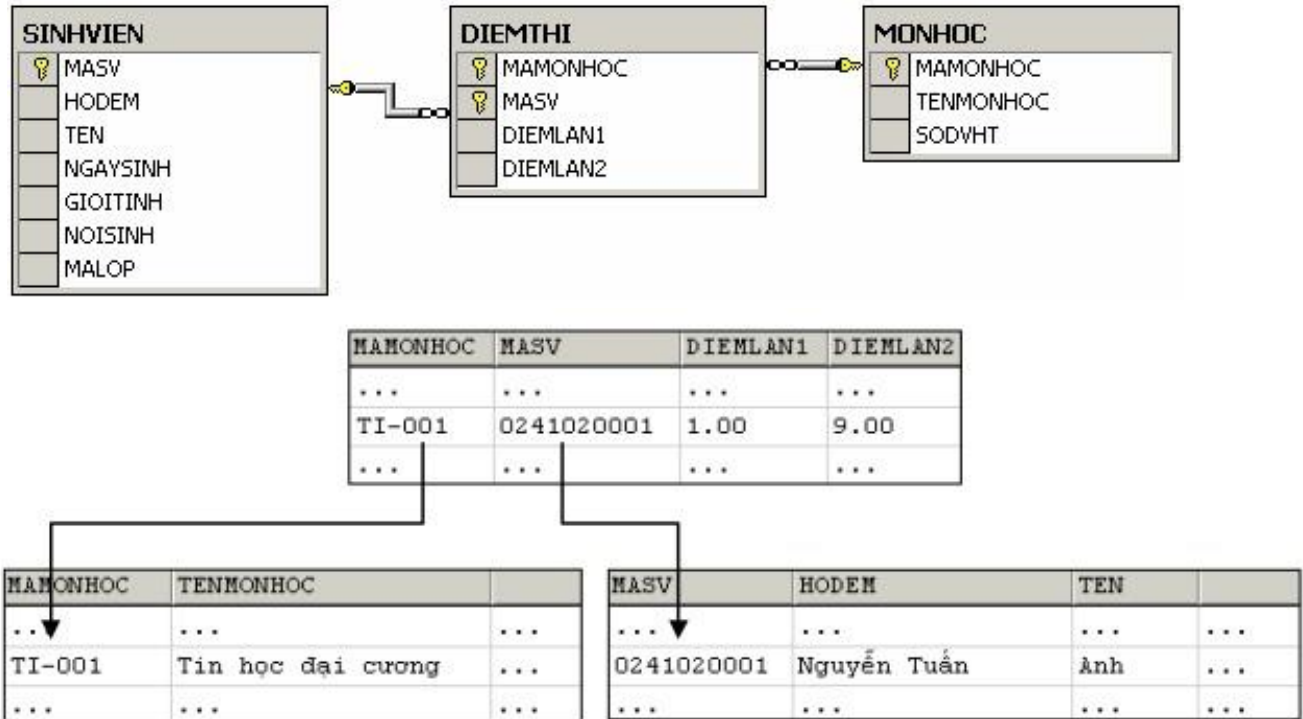
```
CREATE TABLE lop (  
ma NVARCHAR(10) ,  
ten NVARCHAR(10) ,  
khoa SMALLINT NULL ,  
hedaotao NVARCHAR(10) NULL ,  
namnhaphoc INT NULL ,  
makhoa NVARCHAR(10) CONSTRAINT pk_lop PRIMARY KEY  
(malop) ,  
CONSTRAINT unique_lop_tenlop UNIQUE(tenlop) )
```

Ràng buộc FOREIGN KEY

Các bảng trong một cơ sở dữ liệu có mối quan hệ với nhau. Những mối quan hệ này biểu diễn cho sự quan hệ giữa các đối tượng trong thế giới thực. Về mặt dữ liệu, những mối quan hệ được đảm bảo thông qua việc đòi hỏi sự có mặt của một giá trị dữ liệu trong bảng này phải phụ thuộc vào sự tồn tại của giá trị dữ liệu đó ở trong một bảng khác.

Ràng buộc FOREIGN KEY được sử dụng trong định nghĩa bảng dữ liệu nhằm tạo nên mối quan hệ giữa các bảng trong một cơ sở dữ liệu. Một hay một tập các cột trong một bảng được gọi là khoá ngoại, tức là có ràng buộc FOREIGN KEY, nếu giá trị của nó được xác định từ khoá chính (PRIMARY KEY) hoặc khoá phụ (UNIQUE) của một bảng dữ liệu khác.

Hình dưới đây cho ta thấy được mối quan hệ giữa 3 bảng DIEMTHI, SINHVIEN và MONHOC. Trong bảng DIEMTHI, MASV là khoá ngoại tham chiếu đến cột MASV của bảng SINHVIEN và MAMONHOC là khoá ngoại tham chiếu đến cột MAMONHOC của bảng MONHOC.



Mối quan hệ giữa các bảng

Với mối quan hệ được tạo ra như hình trên, hệ quản trị cơ sở dữ liệu sẽ kiểm tra tính hợp lệ của mỗi bản ghi trong bảng DIEMTHI mỗi khi được bổ sung hay cập nhật. Một bản ghi bất kỳ trong bảng DIEMTHI chỉ hợp lệ (đảm bảo ràng buộc FOREIGN KEY) nếu giá trị của cột MASV phải tồn tại trong một bản ghi nào đó của bảng SINHVIEN và giá trị của cột MAMONHOC phải tồn tại trong một bản ghi nào đó của bảng MONHOC.

Ràng buộc FOREIGN KEY được định nghĩa theo cú pháp dưới đây:

```
[CONSTRAINT tên_ràng_buộc] FOREIGN KEY[(danh_sách_cột)]
REFERENCES
tên_bảng_tham_chiếu(danh_sách_cột_tham_chiếu)
[ON DELETE CASCADE|NO ACTION|SET NULL|SET DEFAULT]
[ON UPDATE CASCADE|NO ACTION|SET NULL|SET DEFAULT]
```

Việc định nghĩa một ràng buộc FOREIGN KEY bao gồm các yếu tố sau:



- Tên cột hoặc danh sách cột của bảng được định nghĩa tham gia vào khoá ngoài.
- Tên của bảng được tham chiếu bởi khoá ngoài và danh sách các cột được tham chiếu đến trong bảng tham chiếu.
- Cách thức xử lý đối với các bản ghi trong bảng được định nghĩa trong trường hợp các bản ghi được tham chiếu trong bảng tham chiếu bị xoá (ON DELETE) hay cập nhật (ON UPDATE). SQL chuẩn đưa ra 4 cách xử lý:
 - CASCADE: Tự động xoá (cập nhật) nếu bản ghi được tham chiếu bị xoá (cập nhật).
 - NO ACTION: (Mặc định) Nếu bản ghi trong bảng tham chiếu đang được tham chiếu bởi một bản ghi bất kỳ trong bảng được định nghĩa thì bản ghi đó không được phép xoá hoặc cập nhật (đối với cột được tham chiếu).
 - SET NULL: Cập nhật lại khoá ngoài của bản ghi thành giá trị NULL (nếu cột cho phép nhận giá trị NULL).
 - SET DEFAULT: Cập nhật lại khoá ngoài của bản ghi nhận giá trị mặc định (nếu cột có qui định giá trị mặc định).

Câu lệnh dưới đây định nghĩa bảng DIEMTHI với hai khoá ngoài trên cột MASV và cột MAMONHOC (giả sử hai bảng SINHVIEN và MONHOC đã được định nghĩa)

```
CREATE TABLE diemthi ( ma NVARCHAR (10) ma NVARCHAR(10)
diemlan1 NUMERIC(4,2) diemlan2 NUMERIC(4,2) CONSTRAINT
pk_diemthi PRIMARY KEY(mamonhoc,masv), CONSTRAINT
fk_diemthi_mamonhoc FOREIGN KEY(mamonhoc) REFERENCES
monhoc(mamonhoc) ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT fk_diemthi_masv FOREIGN KEY(masv) REFERENCES
sinhvien(masv) ON DELETE CASCADE ON UPDATE CASCADE )
```

Lưu ý:

- Cột được tham chiếu trong bảng tham chiếu phải là khoá chính (hoặc là khoá phụ).
- Cột được tham chiếu phải có cùng kiểu dữ liệu và độ dài với cột tương ứng trong khoá ngoài.



- Bảng tham chiếu phải được định nghĩa trước. Do đó, nếu các bảng có mối quan hệ vòng, ta có thể không thể định nghĩa ràng buộc FOREIGN KEY ngay trong câu lệnh CREATE TABLE mà phải định nghĩa thông qua lệnh ALTER TABLE.

2.7. Sửa đổi định nghĩa bảng

Một bảng sau khi đã được định nghĩa bằng câu lệnh CREATE TABLE có thể được sửa đổi thông qua câu lệnh ALTER TABLE. Câu lệnh này cho phép chúng ta thực hiện được các thao tác sau:

- Bổ sung một cột vào bảng.
- Xoá một cột khỏi bảng.
- Thay đổi định nghĩa của một cột trong bảng.
- Xoá bỏ hoặc bổ sung các ràng buộc cho bảng Cú pháp của câu lệnh ALTER TABLE như sau: ALTER TABLE tên_bảng

```
ADD định_nghĩa_cột | ALTER COLUMN tên_cột kiểu_dữ_liệu
[NULL | NOT NULL] | DROP COLUMN tên_cột | ADD
CONSTRAINT tên_ràng_buộc định_nghĩa_ràng_buộc | DROP
CONSTRAINT tên_ràng_buộc
```

Các ví dụ dưới đây minh hoạ cho ta cách sử dụng câu lệnh ALTER TABLE trong các trường hợp. Giả sử ta có hai bảng DONVI và NHANVIEN với định nghĩa như sau:

```
CREATE TABLE donvi ( madvINTNOT NULLPRIMARY KEY,
tendvNVARCHAR(30)NOT NULL ) CREATE TABLE nhanvien ( ma
NVARCHAR(10) hoten NVARCHAR(30) ngaysinh DATETIME,
diachiCHAR(30)NOT NULL )
```

Bổ sung vào bảng NHANVIEN cột DIENTHOAI với ràng buộc CHECK nhằm qui định điện thoại của nhân viên là một chuỗi 6 chữ số:

```
ALTER TABLE nhanvien ADD dienthoai NVARCHAR(6)
CONSTRAINT chk_nhanvien_dienthoai CHECK (dienthoai LIKE
```




```
'[09][09][09][09][09][09]')
```

Bổ sung thêm cột MADV vào bảng NHANVIEN:

```
ALTER TABLE nhanvien ADD madv INT NULL
```

Định nghĩa lại kiểu dữ liệu của cột DIACHI trong bảng NHANVIEN và cho phép cột này chấp nhận giá trị NULL:

```
ALTER TABLE nhanvien ALTER COLUMN diachi NVARCHAR(100)
NULL
```

Xoá cột ngày sinh khỏi bảng NHANVIEN :

```
ALTER TABLE nhanvien DROP COLUMN ngaysinh
```

Định nghĩa khoá chính (ràng buộc PRIMARY KEY) cho bảng NHANVIEN là cột MANV:

```
ALTER TABLE nhanvien ADD CONSTRAINT pk_nhanvien PRIMARY
KEY (manv)
```

Định nghĩa khoá ngoài cho bảng NHANVIEN trên cột MADV tham chiếu đến cột MADV của bảng DONVI :

```
ALTER TABLE nhanvien ADD CONSTRAINT _nhavien_madv
FOREIGN
KEY (madv) REFERENCES donvi (madv) ON DELETE CASCADE ON
UPDATE CASCADE
```

Xoá bỏ ràng buộc kiểm tra số điện thoại của nhân viên

```
ALTER TABLE nhanvien DROP CONSTRAINT
CHK_NHANVIEN_DIENHOTOAI
```

Lưu ý:

- Nếu bổ sung thêm một cột vào bảng và trong bảng đã có ít nhất một bản ghi thì cột mới cần bổ sung phải cho phép chấp nhận giá trị NULL hoặc phải có giá trị mặc định.



- Muốn xoá một cột đang được ràng buộc bởi một ràng buộc hoặc đang được tham chiếu bởi một khoá ngoài, ta phải xoá ràng buộc hoặc khoá ngoài trước sao cho trên cột không còn bất kỳ một ràng buộc và không còn được tham chiếu bởi bất kỳ khoá ngoài nào.
- Nếu bổ sung thêm ràng buộc cho một bảng đã có dữ liệu và ràng buộc cần bổ sung không được thoả mãn bởi các bản ghi đã có trong bảng thì câu lệnh ALTER TABLE không thực hiện được.

2.8. Xoá bảng

Khi một bảng không còn cần thiết, ta có thể xoá nó ra khỏi cơ sở dữ liệu bằng câu lệnh DROP TABLE. Câu lệnh này cũng đồng thời xoá tất cả những ràng buộc, chỉ mục, trigger liên quan đến bảng đó. Câu lệnh có cú pháp như sau:

```
DROP TABLE tên_bảng
```

Trong các hệ quản trị cơ sở dữ liệu, khi đã xoá một bảng bằng lệnh DROP TABLE, ta không thể khôi phục lại bảng cũng như dữ liệu của nó. Do đó, cần phải cẩn thận khi sử dụng câu lệnh này.

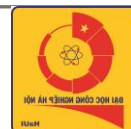
Câu lệnh DROP TABLE không thể thực hiện được nếu bảng cần xoá đang được tham chiếu bởi một ràng buộc FOREIGN KEY. Trong trường hợp này, ràng buộc FOREIGN KEY đang tham chiếu hoặc bảng đang tham chiếu đến bảng cần xoá phải được xoá trước.

Khi một bảng bị xoá, tất cả các ràng buộc, chỉ mục và trigger liên quan đến bảng cũng đồng thời bị xoá theo. Do đó, nếu ta tạo lại bảng thì cũng phải tạo lại các đối tượng này.

Giả sử cột MADV trong bảng DONVI đang được tham chiếu bởi khoá ngoài *fk_nhanvien_madv* trong bảng NHANVIEN. Để xoá bảng DONVI ra khỏi cơ sở dữ liệu, ta thực hiện hai câu lệnh sau:

Xoá bỏ ràng buộc *fk_nhanvien_madv* khỏi bảng NHANVIEN:

```
ALTER TABLE nhanvien DROP CONSTRAINT fk_nhanvien_madv
```



Xoá bảng DONVI:

```
DROP TABLE donvi
```

Tài liệu tham khảo:

[1]. Giáo trình SQL Server – Đỗ Ngọc Sơn, Phan Văn Viên - Tài liệu lưu hành nội bộ của Trường Đại học Công nghiệp Hà Nội, 2015.

[2]. Giáo trình hệ quản trị cơ sở dữ liệu - Đỗ Ngọc Sơn; Phan Văn Viên; Nguyễn Phương Nga - NXB Khoa học Kỹ thuật

[3]. Bài tập Hệ quản trị Cơ sở dữ liệu – Phạm Văn Hà, Trần Thanh Hùng, Đỗ Ngọc Sơn, Nguyễn Thị Thanh Huyền – Trường Đại học Công nghiệp Hà Nội, 2020.