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

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.

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 lí dữ liệu trong một CSDL để xuất ra những thông tin có nghĩa.

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. 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.

1.3. 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.

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 củangười 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.

1.4. Kiểu dữ liệu trong SQL

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.

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

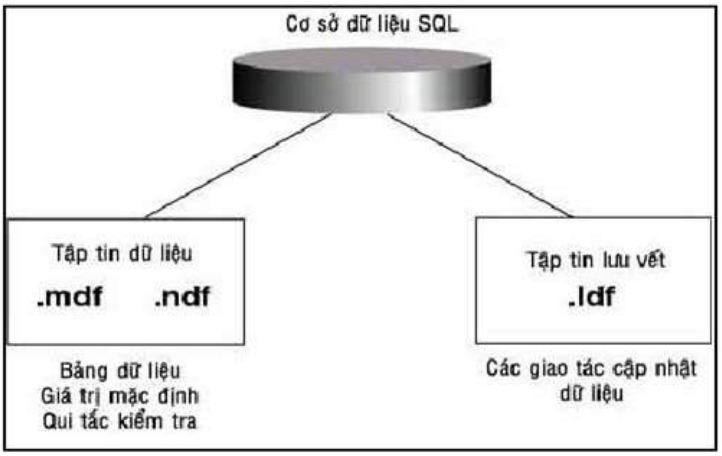
Những giá trị không xác định được biểu diễn trong cơ sở dữ liệu quan hệ bởi các giá trị NULL. Đây là giá trị đặc biệt và không nên nhầm lẫn với chuỗi rỗng (đối với dữ liệu kiểu chuỗi) hay giá trị không (đối với giá trị kiểu số). Giá trị NULL đóng một vai trò quan trọng trong các cơ sở dữ liệu và hầu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

1.5. 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.

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.

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).

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

2.2. Tạo CSDL

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

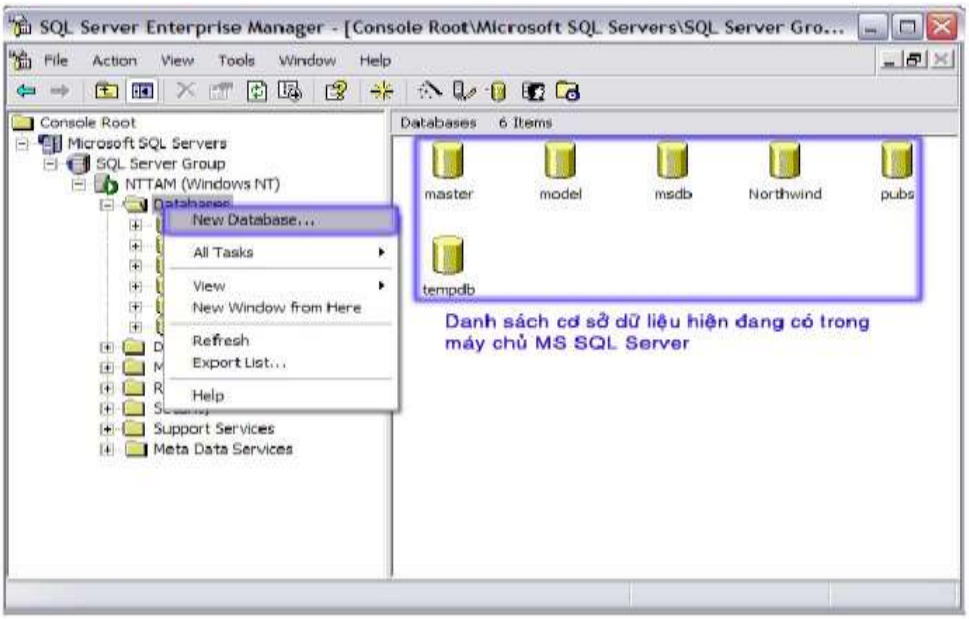
Microsoft SQL Server dùng để quản lý bên trong. Tương ứng mỗi loại tập tin dữ liệu sẽ có một tên tập tin riêng biệt.

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).

Việc tăng trưởng kích thước tập tin dữ liệu (File growth): 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.

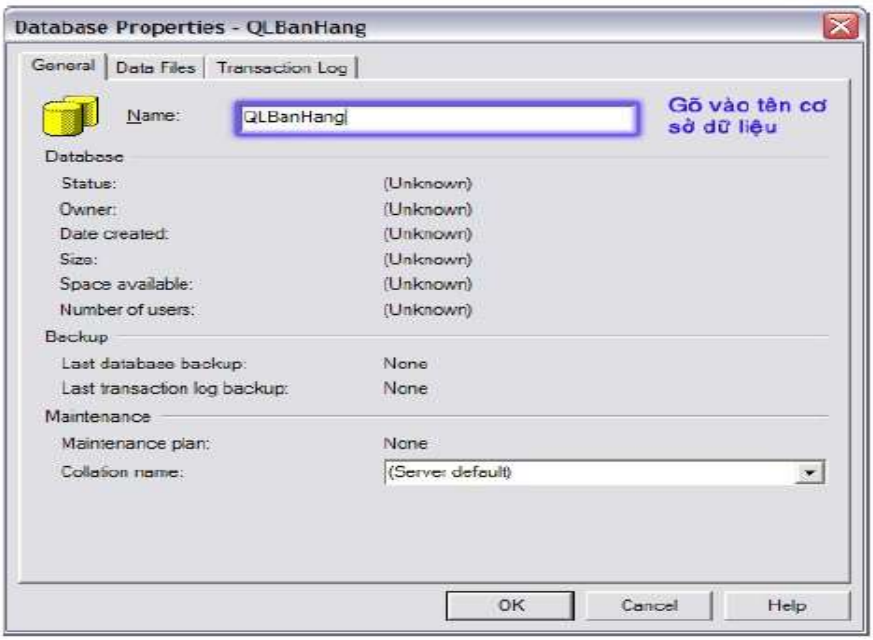
Kích thước tối đa tập tin dữ liệu (Maximum file size) : là việc qui định sự tăng trưởng tự động kích thước của các tập tin dữ liệu nhưng có giới hạn (restrict file growth) đến MB nào đó hoặc là không có giới hạn (un-restrict file growth).

Bước1: 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



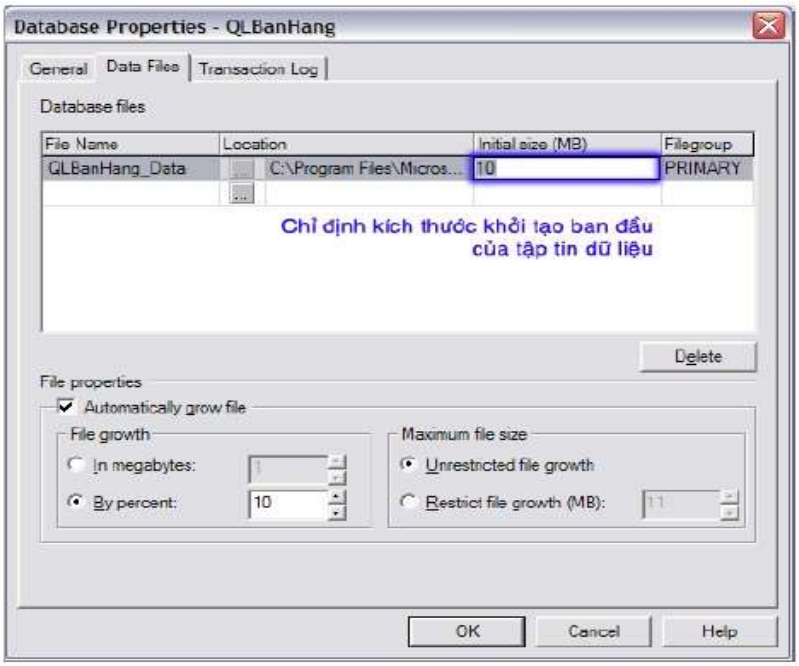
Chọn chức năng New Database

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 Genneral 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

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

Để hủy bỏ cơ sở dữ liệu QLBanHang, thực hiện câu lệnh DROP DATABSE như sau: DROP DATABASE QLBanHang Hoặc nhấn phím Delete trên tên của cơ sở dữ liệu này trong tiện ích Enterprise Manager.

2.4. Tạo bảng dữ liệu

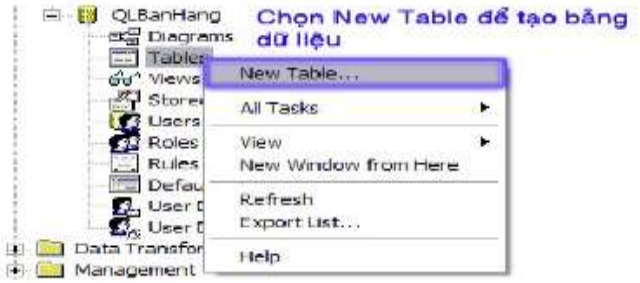
Câu lệnh CREATE TABLE được sử dụng để định nghĩa một bảng dữ liệu mới trong cơ sở dữ liệu. Khi định nghĩa một bảng dữ liệu mới, ta cần phải xác định được các yêu cầu sau đây:

- 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.

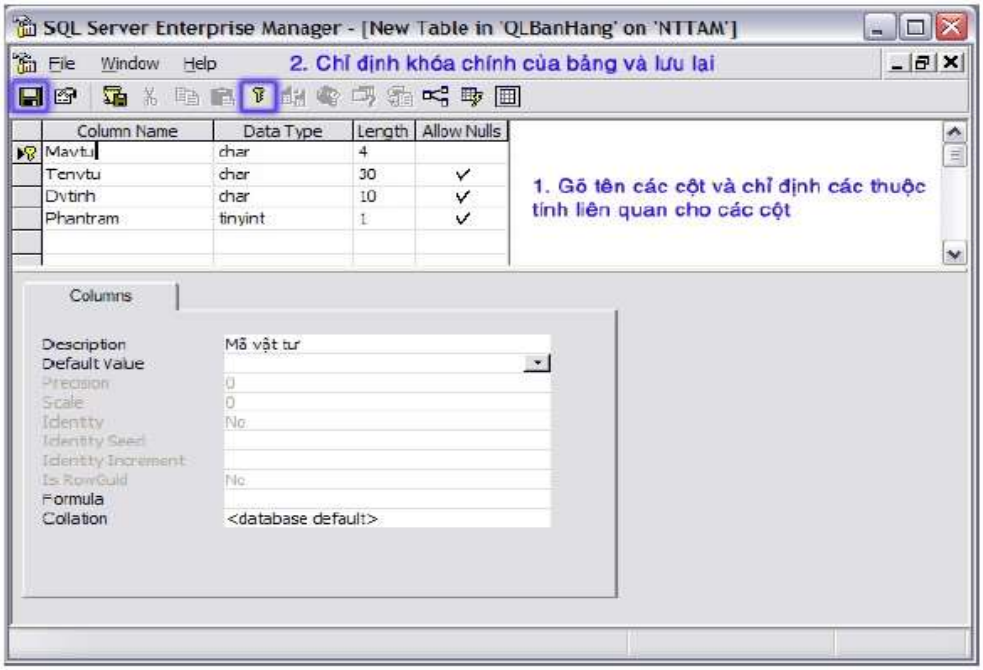
2.5. Tạo cấu trúc bảng dữ liệu bằng EM

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...



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])

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.
- 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 NGAY 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. Ví dụ: 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))

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:

[CONTRAINT tên_ràng_buộc] CHECK (điều kiện)

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 Van A',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 B',6,10.5)

do cột DIEMTOAN nhận giá trị 10.5 không thoả mãn điều kiện của ràng buộc.

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.

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. 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)]

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 Văn','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 khóa 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 (

malop NVARCHAR(10),

tenlop NVARCHAR(10),

khoa SMALLINT NULL,

makhoa NVARCHAR(10),CONSTRAINT pk_lop PRIMARY KEY (malop),

CONSTRAINT unique_lop_tenlop UNIQUE(tenlop))

Ràng buộc FOREIGN KEY

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.

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

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.

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), CONSTRAINT fk_diemthi_masv FOREIGN KEY(masv) REFERENCES sinhvien(masv))

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 khóa 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_côt | ALTER COLUMN tên_cột kiểu_dữ_liêu

[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 (madv INT NOT NULL PRIMARY KEY, tendv NVARCHAR(30) NOT NULL)

CREATE TABLE nhanvien (ma NVARCHAR(10), hoten NVARCHAR(30), ngaysinh DATETIME, diachi CHAR(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)

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

ALTER TABLE nhanvien DROP CONSTRAINT CHK_NHANVIEN_DIENTHOAI

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

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.

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.