



HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU (Database Management Systems)

GV: Lâm Thị Họa Mi

Khoa: Công Nghệ Thông Tin

Email: milth@cntp.edu.vn



CHƯƠNG 2

XÂY DỰNG VÀ KHAI THÁC CƠ SỞ DỮ LIỆU

NỘI DUNG



◆ CƠ SỞ DỮ LIỆU (DATABASE)

◆ BẢNG (TABLE)

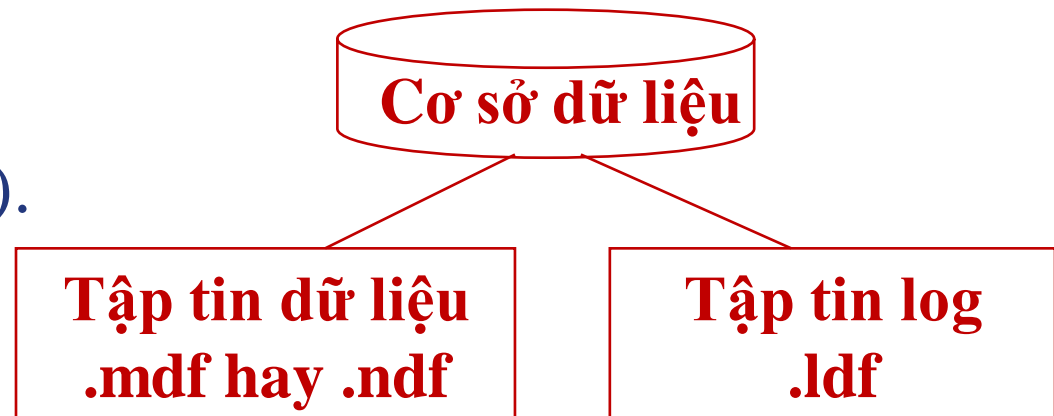
◆ LƯỢC ĐỒ DIAGRAM

◆ BẢNG ẢO (VIEW)

1. CƠ SỞ DỮ LIỆU (DATABASE)



- Mỗi database trong SQL Server chứa duy nhất một data file chính (Primary), có thể có thêm một hay nhiều data phụ (Secondary) và ít nhất một transaction log file (file nhật ký) – mặc dù chỉ cần 1 tập tin log là đủ.
- Phần dữ liệu: tập tin bắt buộc (*.mdf) và tập tin phụ (*.ndf).
- Phần nhật ký (*.ldf).



1. CƠ SỞ DỮ LIỆU (DATABASE)



- **Primary data file (.mdf):**

- file chính chứa data và những system tables.

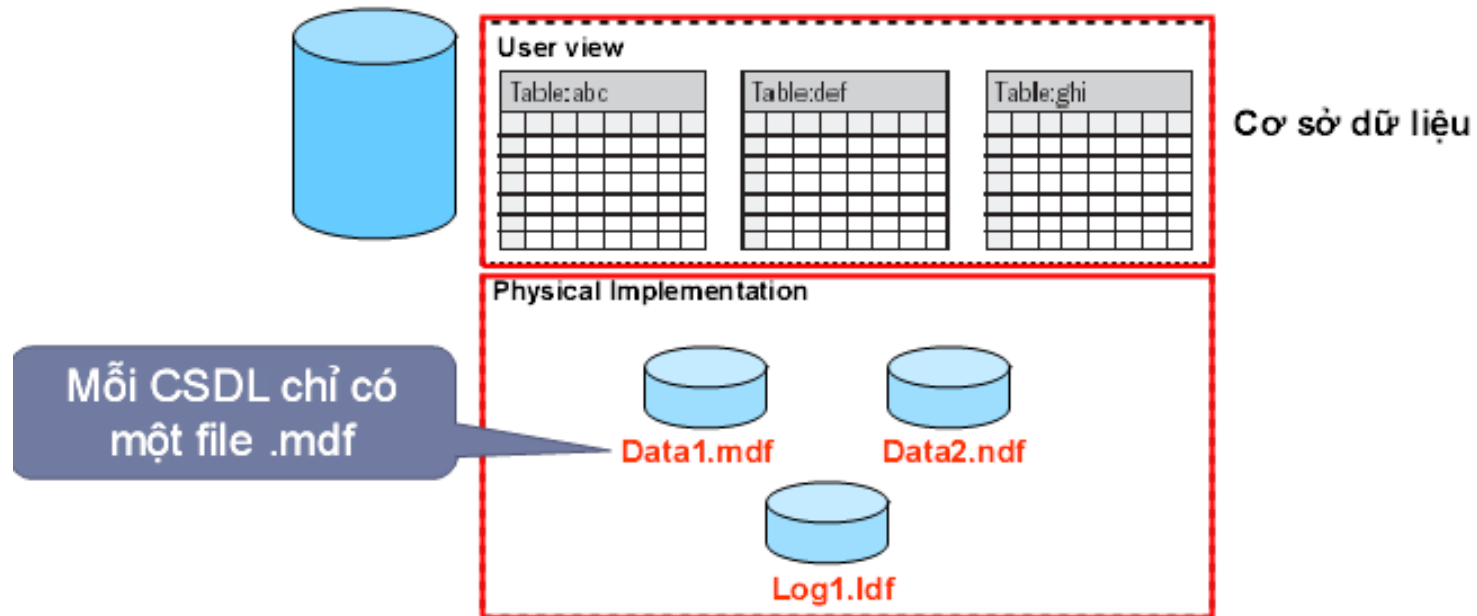
- **Secondary data file (.ndf):**

- file phụ thường chỉ sử dụng khi database được phân chia để chứa trên nhiều đĩa.

- **Transaction log file (.ldf):**

- file ghi lại tất cả những thay đổi diễn ra trong một database và chứa đầy đủ thông tin để có thể roll back hay roll forward khi cần.

1. CƠ SỞ DỮ LIỆU (DATABASE)



- **Về mặt logic:** dữ liệu trong CSDL được lưu trong các đối tượng của CSDL.
- **Về mặt vật lý:** CSDL được lưu trên 2 hoặc nhiều tập tin.

1. CƠ SỞ DỮ LIỆU (DATABASE)



- Một database bao gồm tối thiểu hai file:
 - **.mdf**: lưu trữ các đối tượng trong database như table, view, ...
 - **.ldf**: lưu trữ các quá trình cập nhật/ thay đổi dữ liệu
 - Hỗ trợ phục hồi dữ liệu
 - Hỗ trợ backup/restore dữ liệu.

1. CƠ SỞ DỮ LIỆU (DATABASE)

Các bước tiến hành tạo database: hoạch định vị trí và không gian lưu trữ cho database

- Thông thường để tránh xảy ra rủi ro người ta thường lưu các tập tin transaction log ở ổ đĩa khác với ổ đĩa lưu trữ các tập tin dữ liệu (data file).
- Cần dự tính dung lượng dữ liệu ban đầu cho database sử dụng đủ trong 1 khoảng thời gian nhất định (dung lượng cho data file và dung lượng cho transaction log file). Và thông thường không chọn Autogrowth (tự động tăng trưởng) và Autoshrink (tự động nén).

1. CƠ SỞ DỮ LIỆU (DATABASE)



Nếu chọn Autogrowth (hay Autoshrink) thì chúng ta có thể sẽ gặp 2 vấn đề sau:

- **Performance hit:** Ảnh hưởng đáng kể đến khả năng làm việc của SQL Server. Do nó phải thường xuyên kiểm tra xem có đủ khoảng trống cần thiết hay không và nếu không đủ nó sẽ phải mở rộng bằng cách dành thêm khoảng trống từ đĩa cứng và chính quá trình này sẽ làm chậm đi hoạt động của SQL Server.
- **Disk fragmentation:** Việc mở rộng trên cũng sẽ làm cho data không được liên tục mà chứa ở nhiều nơi khác nhau trong đĩa cứng điều này cũng gây ảnh hưởng lên tốc độ làm việc của SQL Server.

1. CƠ SỞ DỮ LIỆU (DATABASE)



❑ Tạo một CSDL

- Có thể tạo một CSDL bằng:
 - + Dùng SQL Server Manangement Studio.
 - + Dùng lệnh T-SQL.
- Để tạo một CSDL người dùng, các thông tin sau được yêu cầu:
 - Tên CSDL
 - Kích thước của CSDL
 - Các tập tin và các nhóm tập tin để lưu CSDL.

1. CƠ SỞ DỮ LIỆU (DATABASE)



❑ Tạo một CSDL bằng SQL Server Management Studio

1. CƠ SỞ DỮ LIỆU (DATABASE)



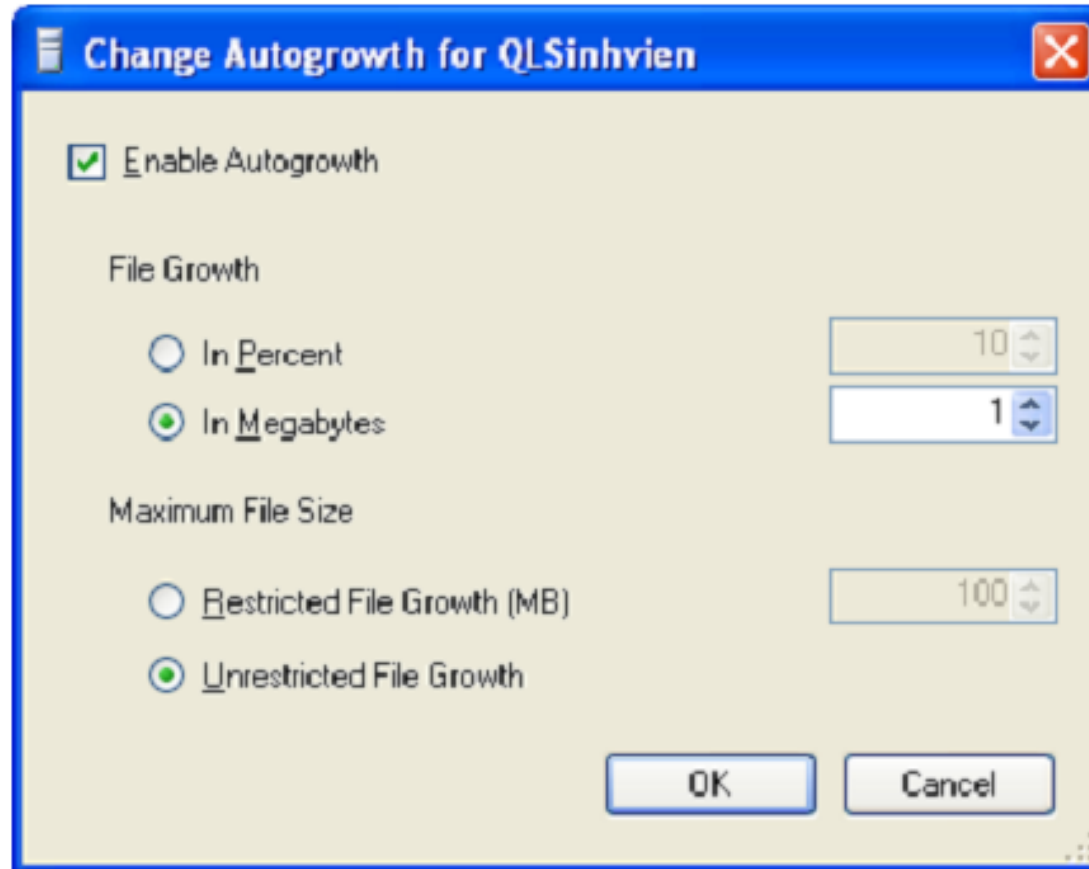
The screenshot shows the 'Properties' dialog box for the 'QLSinhvien' database. The 'Files' tab is selected, displaying a table of database files. The table has columns: Logical Name, File Type, Filegroup, Initial Size (MB), Autogrowth, and Path. Two files are listed: 'QLSinhvien' (Primary data file) and 'QLSinhvien_log' (Log file). The 'Autogrowth' column for the primary file is set to 'By 1 MB, unrestricted growth', and for the log file, it is 'By 10 percent, unrestricted growth'. The 'Path' column shows the file locations on the disk. Callouts highlight specific features: 'Tên file dữ liệu' points to the 'Logical Name' column; 'Đường dẫn vật lý của file dữ liệu' points to the 'Path' column; 'Kích thước ban đầu của tập tin' points to the 'Initial Size (MB)' column; and 'Tùy chọn file tự động tăng trưởng được bật' points to the 'Autogrowth' column.

Logical Name	File Type	Filegroup	Initial Size (MB)	Autogrowth	Path
QLSinhvien	Rows ...	PRIMARY	2	By 1 MB, unrestricted growth	d:\SETUP\Microsoft SQL Server\MS
QLSinhvien_log	Log	Not Applicable	1	By 10 percent, unrestricted growth	d:\SETUP\Microsoft SQL Server\MS

Callouts:

- Tên file dữ liệu
- Đường dẫn vật lý của file dữ liệu
- Kích thước ban đầu của tập tin
- Tùy chọn file tự động tăng trưởng được bật

1. CƠ SỞ DỮ LIỆU (DATABASE)



1. CƠ SỞ DỮ LIỆU (DATABASE)

❑ Tạo một CSDL bằng CREATE DATABASE

Cú pháp:

CREATE DATABASE <Tên_database>

ON PRIMARY

(name= tên logic của file dữ liệu

Filename= 'đường dẫn đến nơi lưu trữ trên đĩa',

Size=<dung lượng file>[KB|MB|GB|TB],

Maxsize=<dung lượng tối đa [KB|MB|GB|TB]|UNLIMITED>,

Filegrowth=<tỉ lệ tăng trưởng>

) [, (...), ..., (...)]

1. CƠ SỞ DỮ LIỆU (DATABASE)



❑ Tạo một CSDL bằng CREATE DATABASE

Cú pháp:

LOG ON

```
(      name= tên logic của file transaction log,  
      Filename= 'đường dẫn đến nơi lưu trữ trên đĩa',  
      Size=<dung lượng>,  
      Maxsize=<dung lượng tối đa>,  
      Filegrowth=<tỉ lệ tăng trưởng>  
);
```

GO

1. CƠ SỞ DỮ LIỆU (DATABASE)



❑ Tạo một CSDL bằng CREATE DATABASE

Giải thích:

- *Tên_database*: tham chiếu đến toàn bộ CSDL.
- *Name*: chỉ định tên logic sẽ sử dụng bên trong SQL Server để tham chiếu đến tập tin CSDL vật lý trên đĩa cứng.
- *File name*: đường dẫn và tên tập tin gắn liền với vị trí của dữ liệu trên đĩa cứng (nó phải là ổ đĩa cứng cục bộ).

1. CƠ SỞ DỮ LIỆU (DATABASE)



❑ Tạo một CSDL bằng CREATE DATABASE

Giải thích:

- *FileGrowth*: chỉ định độ tăng nào được sử dụng đối với việc tự động phát triển của tập tin CSDL.
- *Log On*: mô tả tên, nơi mà *tập tin nhật ký* chuyển tác được tìm kiếm và kích thước của chúng là bao nhiêu.

1. CƠ SỞ DỮ LIỆU (DATABASE)



❑ Tạo một CSDL bằng CREATE DATABASE

Ví dụ 1:

Tạo một CSDL dành ra 20MB lúc đầu cho phần dữ liệu và 5MB cho phần nhật ký. Các tập tin có thể phát triển lên đến 100MB cho phần dữ liệu và 15MB đối với nhật ký.

1. CƠ SỞ DỮ LIỆU (DATABASE)



❑ Tạo một CSDL bằng CREATE DATABASE

Ví dụ 1:

```
CREATE DATABASE DB_NHANVIEN  
ON PRIMARY
```

```
(
```

```
    name = DBNV_Primary,
```

```
    Filename= 'D:\ThuchanhSQL\DBNV_Primary.mdf',
```

```
    Size = 20MB,
```

```
    Maxsize=100MB,
```

```
    Filegrowth= 10MB
```

```
)
```

1. CƠ SỞ DỮ LIỆU (DATABASE)



❑ Tạo một CSDL bằng CREATE DATABASE

Ví dụ 1:

LOG ON

(

name= DBNV_Log,

Filename='D:\ThuchanhSQL\DBNV_Log.ldf',

Size= 5MB,

Maxsize= 15MB ,

Filegrowth= 1MB

);

GO

1. CƠ SỞ DỮ LIỆU (DATABASE)

❑ Tạo một CSDL bằng CREATE DATABASE

Ví dụ 2:

```
CREATE DATABASE DB_SINHVIEN
```

```
ON DDIMADV
```

Lưu ý: Nếu không chỉ định một transaction log file thì SQL sẽ tự động tạo ra một log file với kích thước ban đầu là **bằng $\frac{1}{4}$ dung lượng file .mdf.**

```
Size = 20MB,
```

```
Maxsize=100MB,
```

```
Filegrowth= 10MB
```

```
);
```

1. CƠ SỞ DỮ LIỆU (DATABASE)

❑ Tạo một CSDL bằng CREATE DATABASE

Ví dụ 3: Tạo CSDL tên QLTV

- Tập tin chính tên logic là **qltv_data**, đường dẫn là **d:\qltv_data.mdf**
- Tập tin phụ thứ nhất có tên logic **qltv_data1**, đường dẫn là **d:\ qltv_data1.ndf**
- Tập tin phụ thứ hai có tên logic **qltv_data2**, đường dẫn là **d:\ qltv_data2.ndf**
- Tập tin log có tên logic là **qltv_log**, đường dẫn là **d:\qltv_log.ldf**

1. CƠ SỞ DỮ LIỆU (DATABASE)



❑ Tạo một CSDL bằng CREATE DATABASE

Ví dụ 3: Tạo CSDL tên QLTV

```
CREATE DATABASE qltv
ON primary
( NAME = qltv_data
  FILENAME = 'D:\qltv_data.mdf',
  SIZE = 4 MB,
  MAXSIZE = 10 MB,
  FILEGROWTH = 1MB
),
( NAME = qltv_data1,
  FILENAME = 'D:\qltv_data1.ndf',
  SIZE = 4 MB,
  MAXSIZE = 10 MB,
  FILEGROWTH = 10%
),
( NAME = qltv_data2,
  FILENAME = 'D:\qltv_data2.ndf',
  SIZE = 4 MB,
  MAXSIZE = 10 MB,
  FILEGROWTH = 10%
)
LOG ON
( NAME = qltv_log,
  FILENAME = 'D:\ qltv_log.ldf',
  SIZE = 4 MB,
  MAXSIZE = 10 MB,
  FILEGROWTH = 10%
)
```

1. CƠ SỞ DỮ LIỆU (DATABASE)



❑ **Mở CSDL để làm việc**

Cú pháp:

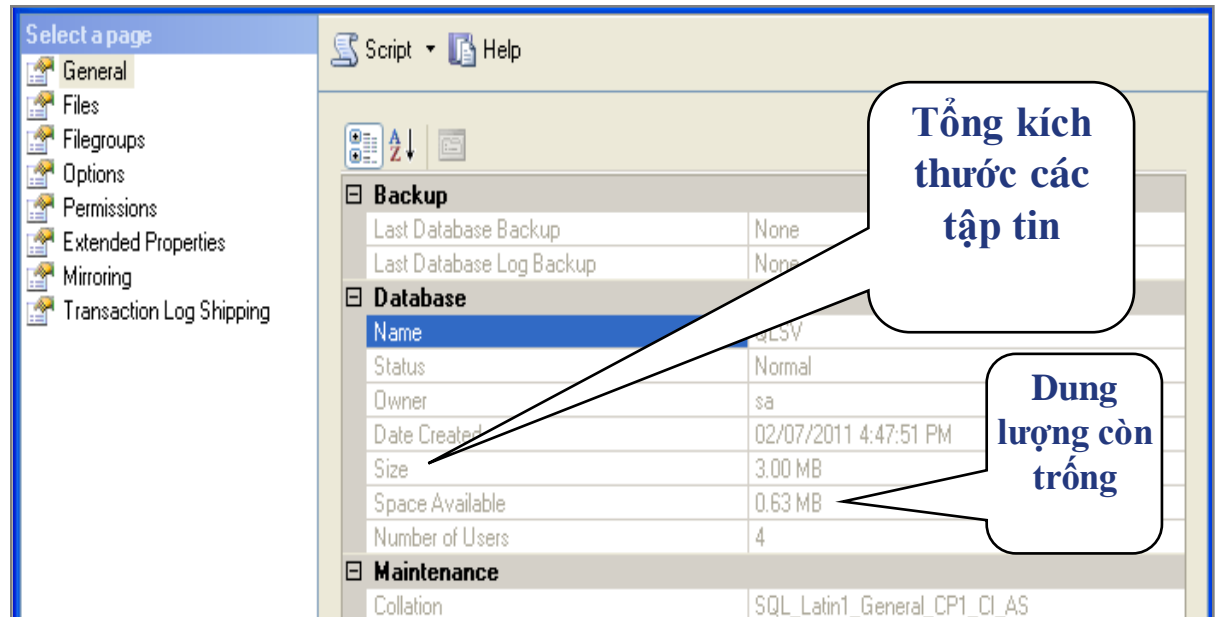
USE <database_name>

1. CƠ SỞ DỮ LIỆU (DATABASE)



❑ Xem kích thước CSDL (SQL Server Management Studio)

Click chuột phải vào cơ sở dữ liệu cần theo dõi kích thước → Chọn Properties → Chọn General → tại mục Database, *Size*



1. CƠ SỞ DỮ LIỆU (DATABASE)



❑ Xem kích thước CSDL

Cú pháp:

Use <database_name>

go

sp_spaceused

```
use QLSV
go
sp_spaceused
```

	database_name	database_size	unallocated space
1	QLSV	3.00 MB	0.63 MB

	reserved	data	index_size	unused
1	1408 KB	664 KB	656 KB	88 KB

1. CƠ SỞ DỮ LIỆU (DATABASE)



❑ Thay đổi kích thước CSDL

Click chuột phải vào cơ sở dữ liệu cần tăng kích thước → Chọn Properties → Chọn mục Files trên cây thư mục bên trái → Thực hiện tăng/giảm kích thước của các tập tin cần thiết trong vùng **Database files**.

Database name: DB_NHANVIEN

Owner: sa

☒ Use full-text indexing

Database files:

Logical Name	File Type	Filegroup	Initial Size (MB)	Autogrowth	Path
DB_PRIMARY	Data	PRIMARY	2	By 10 percent, restricted growth to 5 MB	D:\
DB_SECOND1_1	Data	nhom1	2	By 10 percent, restricted growth to 5 MB	D:\
DB_SECOND1_2	Data	nhom1	2	By 10 percent, restricted growth to 5 MB	D:\
DB_SECOND1_3	Data	nhom2	1	By 5 percent, restricted growth to 3 MB	d:\
DB_Log	Log	Not Applicable	3	By 15 percent, restricted growth to 10 MB	D:\

1. CƠ SỞ DỮ LIỆU (DATABASE)



❑ Thay đổi kích thước CSDL

Tăng kích thước các tập tin của CSDL

Alter database <database_name>

Modify file (name = <tên_file>,
size= <số nguyên > MB)

Lưu ý: Kích thước chỉ định tăng của tập tin phải lớn hơn kích thước hiện hành của tập tin đó trong cơ sở dữ liệu.

1. CƠ SỞ DỮ LIỆU (DATABASE)



❑ Thay đổi kích thước CSDL

Tăng kích thước các tập tin của CSDL

Ví dụ:

Alter database DB_SINHVIEN

Modify file (name= DBSV_Primary, size=20MB)

1. CƠ SỞ DỮ LIỆU (DATABASE)



❑ Hiệu chỉnh thuộc tính của tập tin

Cú pháp:

Alter database <database_name>

Modify file (name = <tên_file>,
 filename = 'đường dẫn',
 size = <số nguyên >,
 maxsize = <số nguyên >,
 filegrowth = <số nguyên >
)

1. CƠ SỞ DỮ LIỆU (DATABASE)



❑ Hiệu chỉnh thuộc tính của tập tin

VD: Hủy khả năng tự tăng trưởng cho
DB_SINHVIENT

Alter database DB_SINHVIENT

Modify file (name= DBSV_Primary,
 filegrowth = 0
)

1. CƠ SỞ DỮ LIỆU (DATABASE)

❑ Thay đổi kích thước CSDL

Giảm kích thước CSDL (bằng cách sử dụng Auto Shrink)

Click chuột phải vào cơ sở dữ liệu cần giảm kích thước
→ Chọn Properties → Chọn mục Option →

Auto Shrink: **True**

Nếu có hơn **25%** không gian trống thì hệ thống sẽ tiến hành thu nhỏ kích thước tập tin để chỉ còn lại **25%** không gian trống

1. CƠ SỞ DỮ LIỆU (DATABASE)

❑ Thay đổi kích thước CSDL

Giảm kích thước CSDL

DBCC ShrinkDatabase

```
(
    <database_name> [,<phần trăm kích
                        thước trống chỉ định>]
);
```

Ví dụ:

DBCC ShrinkDatabase (DB_SINHVIEN, 15);

1. CƠ SỞ DỮ LIỆU (DATABASE)



❑ Thay đổi kích thước CSDL

Giảm kích thước tập tin của CSDL

DBCC ShrinkFile

(
 name=<tên file>[, size=<số nguyên>]);

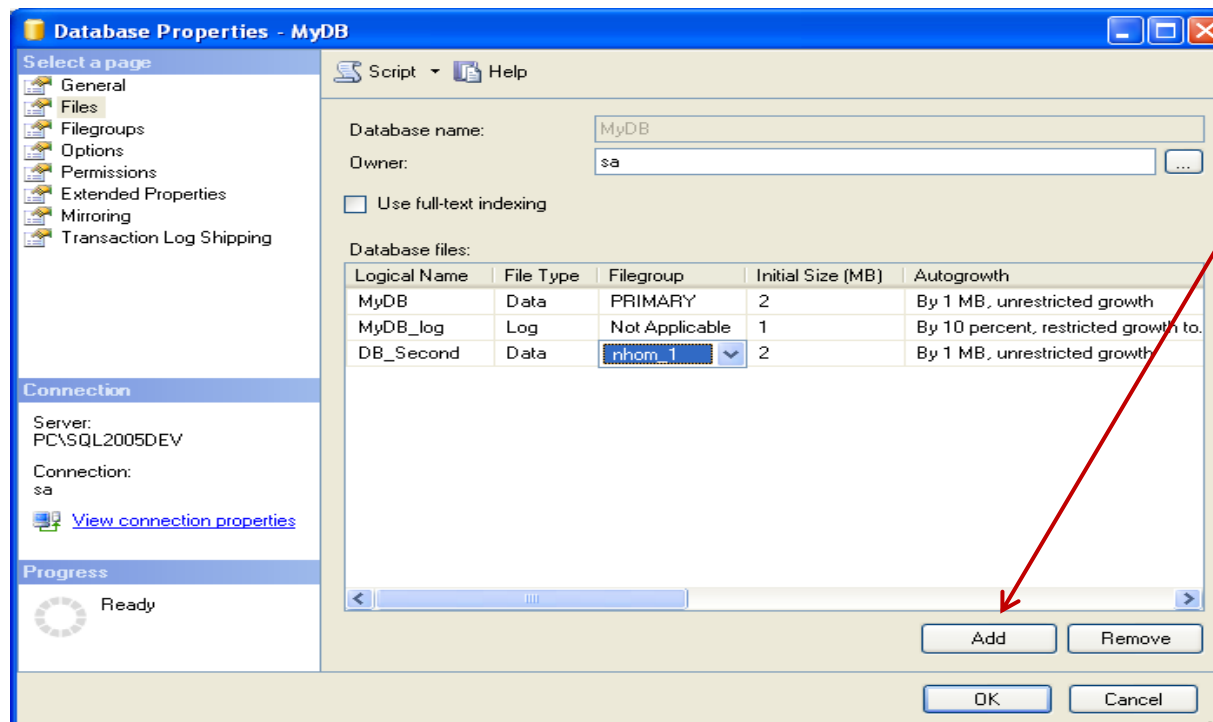
Ví dụ: Giảm kích thước tập tin dữ liệu của CSDL
DB_SINHVIEN xuống còn 10MB.

DBCC ShrinkFile (
 Name = DBSV_Primary, size = 10);

1. CƠ SỞ DỮ LIỆU (DATABASE)

❑ Thêm mới tập tin trong CSDL

Từ cửa sổ Database Properties nhấn nút **Add** để thêm tập tin mới và chỉ định các thông tin cần thiết.



1. CƠ SỞ DỮ LIỆU (DATABASE)



❑ Thêm mới tập tin trong CSDL

ALTER DATABASE <database_name>

Add File [To FileGroup <filegroup_name>]|Add Log File

(

Name= 'tên logic của file dữ liệu',

Filename= 'đường dẫn đến nơi lưu trữ trên đĩa',

Size=<dung lượng file>,

Maxsize=<dung lượng tối đa>,

Filegrowth=<tỉ lệ tăng trưởng>

);

1. CƠ SỞ DỮ LIỆU (DATABASE)



❑ Thêm mới tập tin trong CSDL

Ví dụ:

ALTER DATABASE DB_SINHVIEN

Add File (

Name= DBSV_Primary2,

Filename= 'D:\Thuc hanh SQL\DBSV_Primary2.ndf',

Size=2MB,

Maxsize= 5MB,

Filegrowth= 2MB

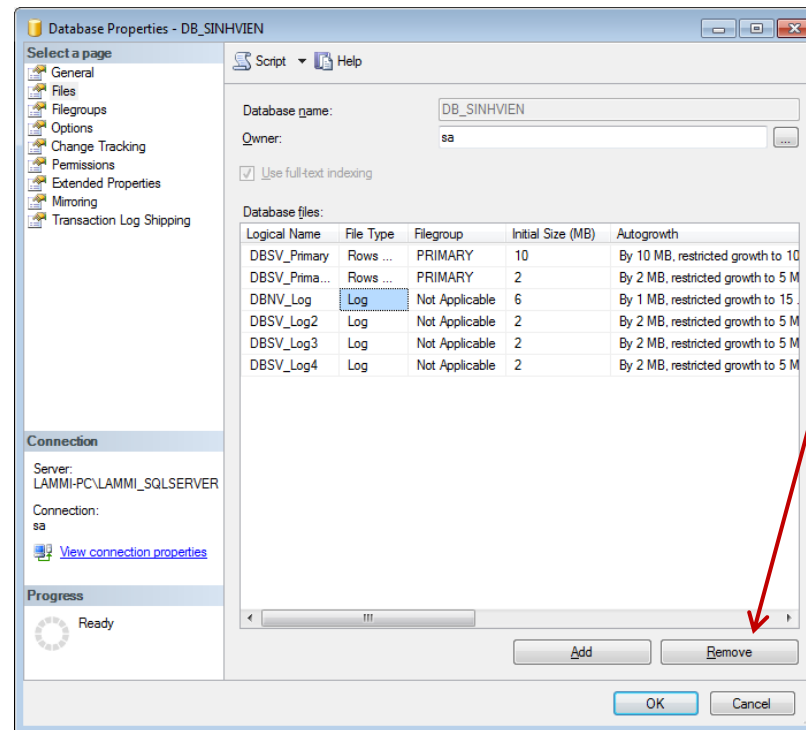
);

1. CƠ SỞ DỮ LIỆU (DATABASE)



❑ Gỡ bỏ tập tin khỏi CSDL

Từ cửa sổ Database Properties chọn tập tin cần hủy và nhấn nút **Remove** để hủy bỏ tập tin.



1. CƠ SỞ DỮ LIỆU (DATABASE)



❑ Gỡ bỏ tập tin khỏi CSDL

ALTER DATABASE <database_name>

Remove File <name>

Lưu ý:

Tập tin phải trống trước khi gỡ bỏ → Sử dụng DBCC ShrinkFile với tùy chọn **EmptyFile** để làm trống một tập tin.

1. CƠ SỞ DỮ LIỆU (DATABASE)



❑ Gỡ bỏ tập tin khỏi CSDL

Ví dụ:

DBCC ShrinkFile

(

DBSV_Primary2, EmptyFile

)

ALTER DATABASE DB_SINHVIEN

Remove File DBSV_Primary2;

2. BẢNG (TABLE)



❑ **Khái niệm:** Bảng là đối tượng lưu trữ dữ liệu chính trong SQL Server, việc tổ chức các bảng dựa vào mô hình cơ sở dữ liệu quan hệ, và được chuẩn hoá dựa vào các dạng chuẩn để dữ liệu được lưu trữ và sử dụng một cách hợp lý, nhất quán.

Ví dụ:

LOP(MALOP, TENLOP, S ISO)

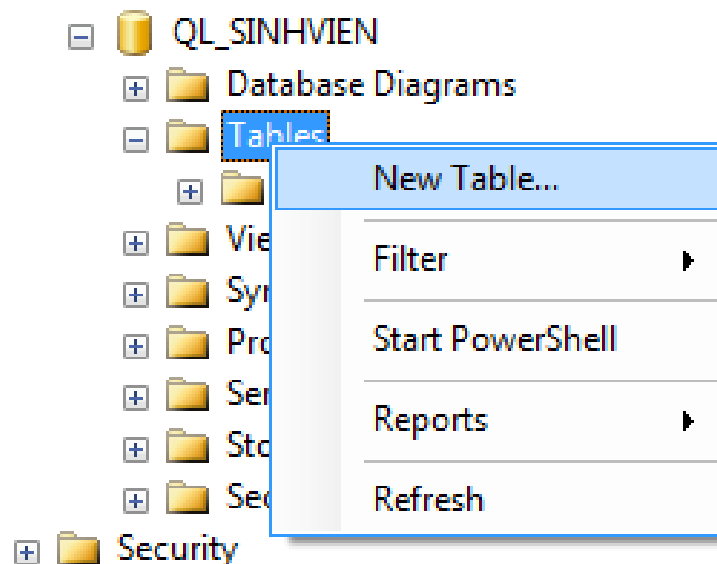
SINHVIEN(MASV, HOTEN, NGSINH, PHAI,
QUEQUAN, MALOP)

2. BẢNG (TABLE)



❑ Tạo cấu trúc bảng bằng công cụ trên SQL Server Management Studio

Mở rộng danh mục một **CSDL** → click phải chuột vào mục *Tables* → chọn *New Table*



2. BẢNG (TABLE)



Microsoft SQL Server Management Studio

File Edit View Project Table Designer Tools Window Community Help

New Query

Object Explorer

- Connect
- LNVSQLEXPRESS (SQL Server 2008)
- Databases
 - System Databases
 - QL_SINHVIEN
 - Database Diagrams
 - Tables
 - Views
 - Synonyms
 - Programmability
 - Service Broker
 - Security
 - Security
 - Server Objects
 - Replication
 - Management

Table - dbo.Table_1*

Column Name	Data Type	Allow Nulls
MALOP	char(10)	<input type="checkbox"/>
TENLOP	varchar(30)	<input checked="" type="checkbox"/>
SISO	int	<input checked="" type="checkbox"/>

Column Properties

(General)

(Name)	MALOP
Allow Nulls	No
Data Type	char
Default Value or Binding	
Length	10

(General)

Properties

[Tbl] dbo.Table_1

(Identity)

(Name)	Table_1
Database Name	QL_SINHVIEN
Description	
Schema	dbo
Server Name	lnv\sqlexpress

Table Designer

Identity Column	
Indexable	Yes
Regular Data	PRIMARY
Replicated	No
Row GUID	
Text/Image	PRIMARY

(Identity)

Ready

2. BẢNG (TABLE)



Set Primary Key – Tạo khóa chính

The screenshot displays the Microsoft SQL Server Management Studio interface. The 'Object Explorer' on the left shows a tree view of databases, with 'QL_SINHVIEN' selected. The 'Table Designer' for 'dbo.Table_1' is open, showing a table with three columns: 'MALOP' (char(10)), 'NAME' (nvarchar(30)), and 'ID' (int). A context menu is open over the 'MALOP' column, with 'Set Primary Key' highlighted. The 'Properties' pane on the right shows the table's configuration, including 'Identity' and 'Table Designer' settings. The 'Table Designer' pane shows the 'General' tab with the following properties:

Property	Value
(Name)	MALOP
Allow Nulls	Yes
Data Type	char
Default Value or Bi	10

The 'Properties' pane on the right shows the following settings for 'Table_1':

Property	Value
(Name)	Table_1
Database Name	QL_SINHVIEN
Description	
Schema	dbo
Server Name	lammi-pc\lammi_sqls
Identity Column	
Indexable	Yes
Lock Escalation	Table
Regular Data Spac	PRIMARY
Replicated	No
Row GUID Column	
Text/Image Filegr	PRIMARY

2. BẢNG (TABLE)



Lưu bảng, đặt tên bảng

A screenshot of a 'Choose Name' dialog box. The title bar says 'Choose Name' and has a question mark icon and a close button. The main area contains the text 'Enter a name for the table:' followed by a text input field. The input field contains the text 'LOP'. At the bottom right, there are two buttons: 'OK' and 'Cancel'.

2. BẢNG (TABLE)



❑ Tạo cấu trúc bảng bằng T – SQL

Để định nghĩa một bảng

- Tên bảng
- Các thuộc tính
 - Tên thuộc tính
 - Kiểu dữ liệu
 - Các RBTV trên thuộc tính

Cú pháp:

2. BẢNG (TABLE)



❑ Tạo cấu trúc bảng bằng T – SQL

```
CREATE TABLE <Tên_bảng>  
(  
    <Tên_cột>    <Kiểu_dữ_liệu> [<RBTV>],  
    <Tên_cột>    <Kiểu_dữ_liệu> [<RBTV>],  
    ...  
    [< RBTV>]  
)
```

2. BẢNG (TABLE)



❑ Tạo cấu trúc bảng bằng T – SQL

Kiểu dữ liệu

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
Int	Số nguyên có giá trị từ -2^{31} đến $2^{31} - 1$
Tinyint	Số nguyên có giá trị từ 0 đến 255.
Smallint	Số nguyên có giá trị từ -2^{15} đến $2^{15} - 1$

2. BẢNG (TABLE)



❑ Tạo cấu trúc bảng bằng T – SQL

Kiểu dữ liệu

Bigint	Số nguyên có giá trị từ -263 đến 263-1
Numeric	Kiểu số với độ chính xác cố định.
Decimal	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)

2. BẢNG (TABLE)



❑ Tạo cấu trúc bảng bằng T – SQL

Kiểu dữ liệu

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	Dữ liệu nhị phân với độ dài chính xác (tối đa 2,147,483,647 bytes)
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)

2. BẢNG (TABLE)



❑ Tạo cấu trúc bảng bằng T – SQL

Ví dụ: **CREATE TABLE NHANVIEN (**
 MANV CHAR(9),
 HONV VARCHAR(10),
 TENNV VARCHAR(10),
 NGSINH DATE,
 DCHI VARCHAR(50),
 PHAI CHAR(3),
 LUONG INT,
 PHG TINYINT
)

2. BẢNG (TABLE)



❑ Lưu ý khi thiết kế 1 table:

- Phải nắm vững về các loại data type.
- Xác định khóa chính chính xác.
- Tránh dùng cột có chứa NULL và nên luôn có giá trị Default cho các cột.
- Phải biết rõ quan hệ (Relationship) giữa các table.

2. BẢNG (TABLE)



❑ Tạo cấu trúc bảng bằng T – SQL

❖ Các loại ràng buộc toàn vẹn (RBTV)

- NOT NULL (không cho rỗng), NULL (cho rỗng)
- UNIQUE (kiểm tra tính duy nhất)
- DEFAULT (giá trị mặc định)
- PRIMARY KEY (khóa chính)
- FOREIGN KEY/ REFERENCES (khóa ngoại)
- CHECK (kiểm tra miền giá trị)

2. BẢNG (TABLE)



❑ Tạo ràng buộc kiểm tra MGT bên trong bảng:

Ví dụ 1: Xét lược đồ quan hệ

HANG (MAHG, TENHG, DONGIA)

Khi tạo bảng cần tạo một ràng buộc kiểm tra MGT cho cột đơn giá khi thêm hay sửa phải >0 .

2. BẢNG (TABLE)



❑ Tạo ràng buộc kiểm tra MGT bên trong bảng:

Ví dụ 1: Xét lược đồ quan hệ

HANG (MAHG, TENHG, DONGIA)

Cách1 (Vị trí 1):

```
CREATE TABLE HANG (  
    MAHG Char(5) Primary Key,  
    TENHG      Varchar(50),  
    DONGIA     Int    CHECK(DONGIA>0)  
)
```

2. BẢNG (TABLE)



❑ Tạo ràng buộc kiểm tra MGT bên trong bảng:

❖ Cách 2: Đặt tên cho ràng buộc (Vị trí 2)

❖ Cú pháp: **CONSTRAINT** <Ten_RBTV> <RBTV>

CREATE TABLE HANG (

 MAHG Char(5) Primary Key,

 TENHG Varchar(50),

 DONGIA Int,

CONSTRAINT ck_dongia CHECK(DONGIA>0)

)

2. BẢNG (TABLE)



❑ Tạo ràng buộc kiểm tra tính duy nhất (Unique):

Ví dụ 2: Xét lược đồ quan hệ

HANG (MAHG, TENHG, DONGIA)

Tạo bảng HANG có kiểm tra tính duy nhất của **TENHANG** (nghĩa là khi thêm hay sửa dữ liệu thì tên hàng phải duy nhất).

2. BẢNG (TABLE)



❑ Tạo ràng buộc kiểm tra tính duy nhất (Unique):

Ví dụ 2: Xét lược đồ quan hệ

HANG (MAHG, TENHG, DONGIA)

Cách1 (Vị trí 1):

```
CREATE TABLE HANG (  
    MAHG Char(5) Primary Key,  
    TENHG      Varchar(50) UNIQUE,  
    DONGIA     Int      CHECK(DONGIA>0)  
)
```

2. BẢNG (TABLE)



❑ Tạo ràng buộc kiểm tra tính duy nhất (Unique):

❖ Cách 2: Đặt tên cho ràng buộc (Vị trí 2)

❖ Cú pháp: **CONSTRAINT** <Ten_RBTV> <RBTV>

CREATE TABLE HANG (

 MAHG Char(5) Primary Key,

 TENHG Varchar(50),

 DONGIA Int CHECK(DONGIA>0),

CONSTRAINT uni_tenhang UNIQUE(TENHG)

)

2. BẢNG (TABLE)



❑ Tạo ràng buộc kiểm tra giá trị mặc định (Default):

Ví dụ 3: Xét lược đồ quan hệ

SINHVIEN (MASV, HOTEN, DIACHI)

Tạo ràng buộc giá trị mặc định thực hiện điền vào giá trị ‘Chưa xác định’ khi để trống cột **DIACHI** lúc thêm dữ liệu.

2. BẢNG (TABLE)



❑ Tạo ràng buộc kiểm tra giá trị mặc định (Default):

Ví dụ 3: Xét lược đồ quan hệ

SINHVIEN (MASV, HOTEN, DIACHI)

Cách1 (Vị trí 1):

```
CREATE TABLE      SINHVIEN (  
    MASV Char(10) Primary Key,  
    HOTEN      Varchar(50),  
    DIACHI Varchar(50) DEFAULT 'Chưa xác định'  
)
```

2. BẢNG (TABLE)



❑ Tạo ràng buộc kiểm tra giá trị mặc định (Default):

❖ Cách 2: Đặt tên cho ràng buộc (Vị trí 2)

❖ Cú pháp: **CONSTRAINT** <Ten_RBTV> <RBTV>

CREATE TABLE **SINHVIEN** (

MASV Char(10) Primary Key,

HOTEN Varchar(50),

DIACHI Varchar(50) **CONSTRAINT** df_diachi

DEFAULT 'Chưa xác định'

)

2. BẢNG (TABLE)



Ví dụ: Viết lệnh T–SQL tạo bảng sau:

NHANVIEN (MANV, HONV, TENNV, NGSINH,
DIACHI, PHAI, LUONG, PHG)

Trong đó:

MANV là khóa chính, **HONV**, **TENNV** không được rỗng, **PHAI** chỉ được nhập giá trị ‘**Nam**’ hay ‘**Nữ**’, **LUONG** được mặc định là **1150000**.

2. BẢNG (TABLE)



Ví dụ:

```
CREATE TABLE      NHANVIEN (  
    MANV           CHAR(9) PRIMARY KEY,  
    HONV           VARCHAR(10) NOT NULL,  
    TENNV          VARCHAR(10) NOT NULL,  
    NGSINH         DATE,  
    DCHI           VARCHAR(50),  
    PHAI           CHAR(3) CHECK (PHAI IN ('Nam', 'Nu')),  
    LUONG          INT   DEFAULT (1150000),  
    PHG            TINYINT )
```


2. BẢNG (TABLE)



❑ Thêm ràng buộc vào **bảng đã tồn tại (Vị trí 3)**

ALTER TABLE <tên_bảng>
ADD [CONSTRAINT
 <tên_ràng_buộc>]

PRIMARY KEY (tên_cột)

UNIQUE (tên_cột)

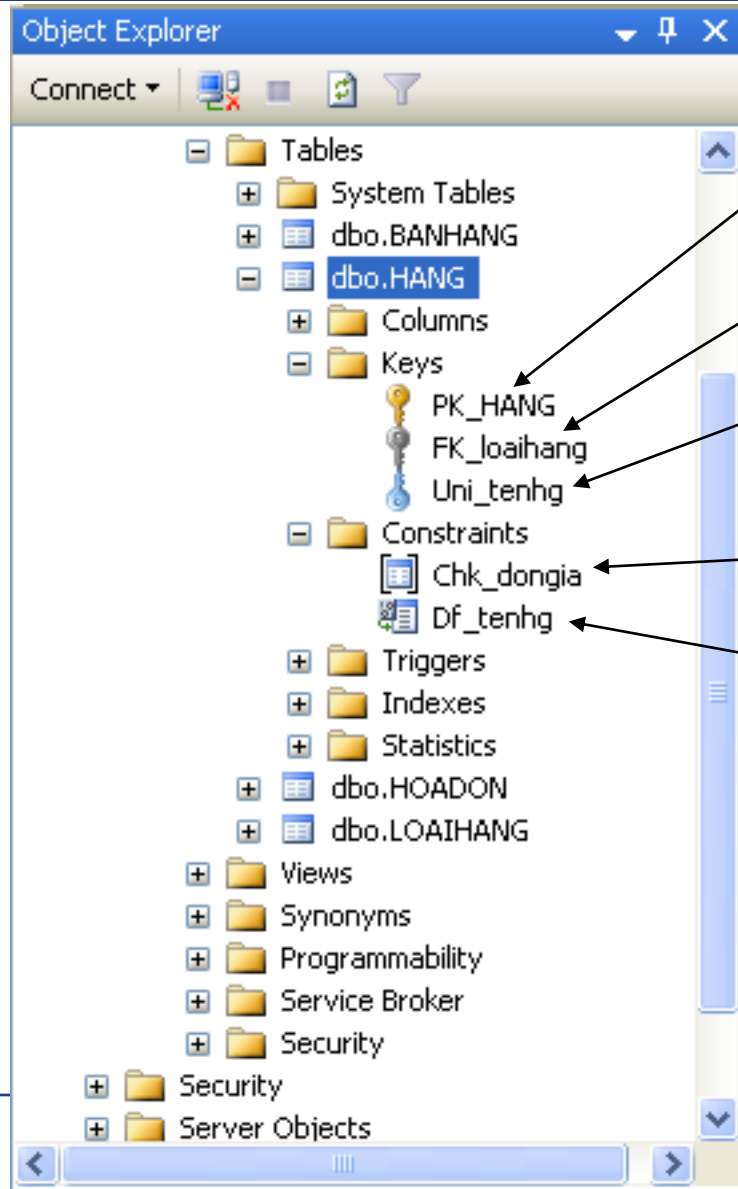
FOREIGN KEY (tên_cột)
REFERENCES tên_bảng
 (cột_là_khóa_chính)

CHECK (tên_cột điều_kiện)

DEFAULT (biểu_thức)
FOR (tên_cột)

2. BẢNG (TABLE)

❑ Quản lý RBTV



Primary key

Foreign key

Unique

Check

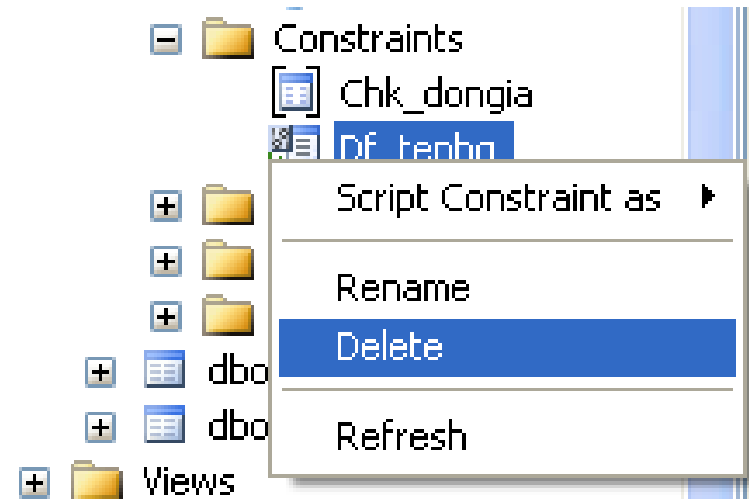
Default

2. BẢNG (TABLE)



❑ Xóa RBTV

Mở thư mục chứa ràng buộc > nhấp chuột phải vào ràng buộc cần xóa > chọn **Delete**.



Hoặc

ALTER TABLE

⟨Tên_bảng⟩

DROP CONSTRAINT

⟨Tên_RBTV [, Tên_RBTV , ...]⟩

BÀI TẬP



❑ **Cho lược đồ CSDL QLHH: quản lý hàng hóa sau**

HangHoa(MaHG, TenHG, DVT, DonGia, CoHang)

Khach(MaKH, HoTen, DChi, DaiLy, SoDT)

HoaDon(SoHD, NgayLap, NgayGiao, TriGia, MaKH)

ChiTiet_HD(SoHD, MaHG, SoLuong, GiaBan)

Yêu cầu: Viết câu lệnh để tạo tất cả các bảng dựa vào các LĐQH trên (với các ràng buộc khóa chính, khóa ngoại; **TenHG**, **HoTen**, **DaiLy** không được rỗng; **CoHang** chỉ nhận giá trị 0 hoặc 1; **DVT** chỉ nhận giá trị: Cái, Lọ, Thố; **NgayLap**, **NgayGiao** nhận giá trị mặc định là ngày hiện tại của hệ thống; **SoDT** có ràng buộc duy nhất; **DonGia**, **TriGia**, **SoLuong**, **GiaBan** phải có giá trị >0).

2. BẢNG (TABLE)



❑ **Lệnh sửa cấu trúc bảng**

Được dùng để

- Thay đổi cấu trúc bảng
- Thay đổi RBTV

Thêm cột

```
ALTER TABLE      <Tên_bảng>
ADD
                   <Tên_cột>      <Kiểu_dữ_liệu>      [<RBTV >]
```

2. BẢNG (TABLE)



❑ Lệnh sửa cấu trúc bảng

Xóa cột

ALTER TABLE	⟨Tên_bảng⟩
DROP COLUMN	⟨Tên_cột⟩

Mở rộng cột

ALTER TABLE	⟨Tên_bảng⟩
ALTER COLUMN	⟨Tên_cột⟩ ⟨Kiểu_dữ_liệu_mới⟩

2. BẢNG (TABLE)



❑ Lệnh sửa cấu trúc bảng

Thêm RBTV

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

Xóa RBTV

```
ALTER TABLE <Tên_bảng>  
DROP <Tên_RBTV>
```

2. BẢNG (TABLE)



❑ Lệnh sửa cấu trúc bảng

```
ALTER TABLE NHANVIEN  
ADD NGHENGHIEP CHAR(20)
```

```
ALTER TABLE NHANVIEN  
DROP COLUMN NGHENGHIEP
```

```
ALTER TABLE NHANVIEN  
ALTER COLUMN NGHENGHIEP CHAR(50)
```


2. BẢNG (TABLE)



❑ Lệnh sửa cấu trúc bảng

```
CREATE TABLE PHONGBAN (  
    TENPB VARCHAR(20),  
    MAPHG INT NOT NULL,  
    TRPHG CHAR(9),  
    NG_NHANCHUC DATETIME  
)
```

```
ALTER TABLE PHONGBAN ADD  
    CONSTRAINT PB_MAPHG_PK PRIMARY KEY (MAPHG),  
    CONSTRAINT PB_TRPHG FOREIGN KEY (TRPHG)  
        REFERENCES NHANVIEN(MANV),  
    CONSTRAINT PB_TENPB_UNI UNIQUE (TENPB)
```

2. BẢNG (TABLE)



❑ Lệnh xóa bảng

Được dùng để xóa cấu trúc bảng

- Tất cả dữ liệu của bảng cũng bị xóa

Cú pháp

DROP TABLE <Tên_bảng>

Ví dụ

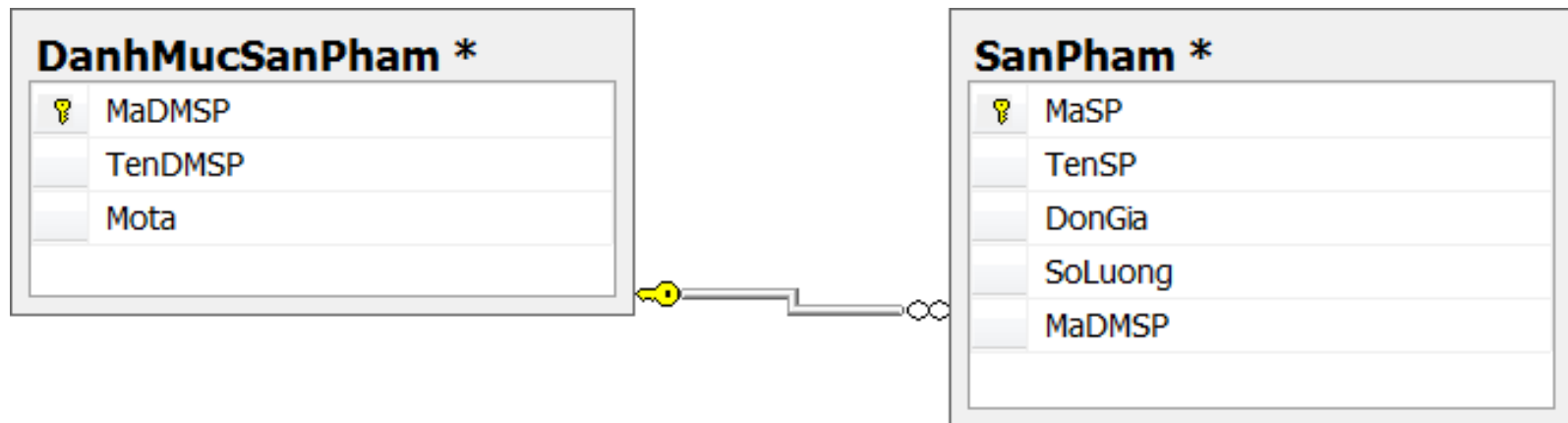
DROP TABLE NHANVIEN

3. LƯỢC ĐỒ DIAGRAM



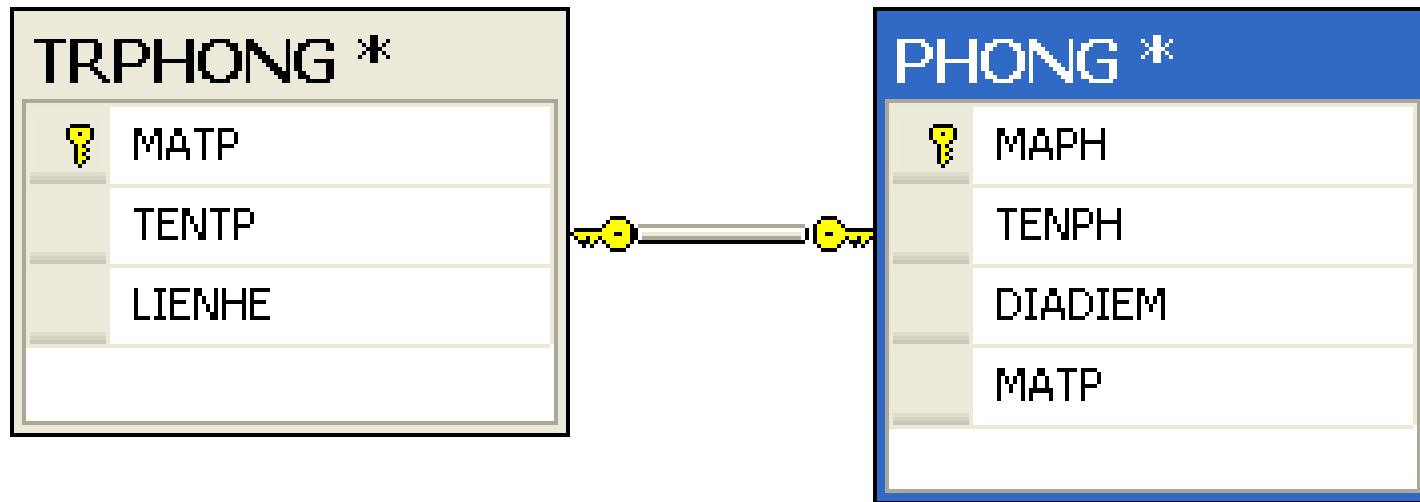
❑ Đối tượng *Diagram* là một lược đồ thể hiện sự liên kết các bảng trong CSDL với nhau. Trên lược đồ *Diagram* chỉ thể hiện 2 loại mối liên kết là **1-n** và **1-1**.

3. LƯỢC ĐỒ DIAGRAM



Liên kết là 1 – n

3. LƯỢC ĐỒ DIAGRAM



Liên kết là 1 – 1

NHẬP DỮ LIỆU VÀO BẢNG



Lưu ý:

Khi nhập liệu vào bảng phải nhập theo đúng thứ tự là bảng có liên kết 1 nhập trước, bảng có liên kết nhiều nhập sau.

Có 3 hình thức nhập dữ liệu vào bảng:

- Nhập trực tiếp
- Nhập bằng lệnh
- Nhập từ một bảng khác

NHẬP DỮ LIỆU VÀO BẢNG



❑ **Nhập trực tiếp:** nhấp chuột phải vào bảng **LOP** > chọn **Edit Top 200 Rows**

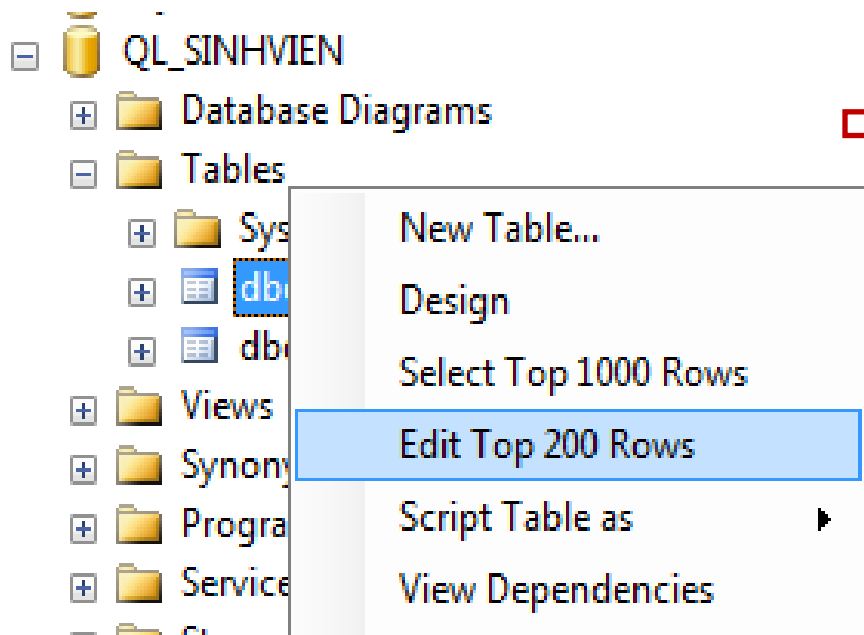


Table - dbo.LOP			
Summary			
	MALOP	TENLOP	SISO
	L001	08CDTH1	60
	L002	08CDTH2	70
▶*	NULL	NULL	NULL

NHẬP DỮ LIỆU VÀO BẢNG



❑ Nhập bằng lệnh T – SQL

```
INSERT INTO <Tên_bảng> [(<danh sách tên cột>)]  
VALUES (<danh sách các giá trị>)
```


NHẬP DỮ LIỆU VÀO BẢNG



❑ Nhập bằng lệnh T – SQL

Lưu ý:

- Thứ tự các giá trị phải trùng với thứ tự các cột
- Có thể thêm giá trị NULL ở những thuộc tính không là khóa chính và NOT NULL
- Câu lệnh **INSERT** sẽ gặp lỗi nếu vi phạm RBTV
 - Khóa chính, Tham chiếu
 - NOT NULL – các thuộc tính có ràng buộc NOT NULL bắt buộc phải có giá trị.

NHẬP DỮ LIỆU VÀO BẢNG



❑ Nhập bằng lệnh T – SQL

Ví dụ:

MANV	HOTEN	NTNS	PHAI	MA_NQL	MaPH	LUONG
001	Vuong Ngoc Quyen	22/10/1957	Nu		QL	3.000.000
002	Nguyen Thanh Tung	09/01/1955	Nam	001	NC	2.500.000
003	Le Thi Nhan	18/12/1960	Nu	001	DH	2.500.000
004	Dinh Ba Tien	09/01/1968	Nam	002	NC	2.200.000
005	Bui Thuy Vu	19/07/1972	Nam		DH	2.200.000
006	Nguyen Manh Hung	15/09/1973	Nam	002	NC	2.000.000
007	Tran Thanh Tam	31/07/1975	Nu	002	NC	2.200.000
008	Tran Hong Minh	04/07/1976	Nu	004	NC	1.800.000

NHẬP DỮ LIỆU VÀO BẢNG



❑ Một số lưu ý

- Thêm ký tự **N** trước chuỗi Unicode (Chuỗi nhập có dấu tiếng Việt).

Ví dụ:

Insert into NHANVIEN

Values ('NV01', **N**'Nguyễn Văn Trường', 'Nam')

NHẬP DỮ LIỆU VÀO BẢNG



❑ Một số lưu ý

Đã có ràng buộc khóa ngoại

Cách 1:

B1: Nhập PHONGBAN


B2: Nhập DEAN

Cách 2:

B1: Nhập DEAN, nhập **phong** = **NULL**

B2: Nhập PHONGBAN

B3: Cập nhật DEAN

DEAN	
	tenda
	mada
	ddiemda
	phong

PHONGBAN	
	tenphong
	maphong
	trphong
	ngnhanhuc

NHẬP DỮ LIỆU VÀO BẢNG



PHONGBAN

tenphong

maphong

trphong

ngnhanchuc

NHANVIEN *

honv

tenlot

tennv

manv

ngaysinh

diachi

phai

luong

manql

phong

đã có ràng buộc khóa ngoại

❖ Cách 1:

B1: Nhập NHANVIEN, nhập
phong=NULL

B2: Nhập PHONGBAN

B3: Cập nhật thuộc tính **phong**
trong NHANVIEN

NHẬP DỮ LIỆU VÀO BẢNG



PHONGBAN

tenphong

maphong

trphong

ngnhanchuc

NHANVIEN *

honv

tenlot

tennv

manv

ngaysinh

diachi

phai

luong

manql

phong

đã có ràng buộc khóa ngoại

❖ Cách 2:

B1: Nhập PHONGBAN, nhập
trphong=NULL

B2: Nhập NHANVIEN

B3: Cập nhật thuộc tính **trphong**
trong PHONGBAN

NHẬP DỮ LIỆU VÀO BẢNG



❑ Một số lưu ý

MaGV	TenGV	NgaySinh	GioiTinh	DienThoai	MaGVQuanLi
GV00001	Nguyễn Văn An	1981-01-02 00:...	Nam	NULL	GV00002
GV00002	Nguyễn Thị Như Lan	1984-12-02 00:...	Nữ	NULL	GV00005
GV00003	Trần Minh Anh	1986-03-23 00:...	Nam	0909123999	GV00002
GV00004	Trương Tường Vi	1988-02-01 00:...	Nữ	0998990909	GV00008
GV00005	Hà Anh Tuấn	1986-12-03 00:...	Nam	0909909000	GV00008
GV00006	Trần Anh Dũng	1979-04-04 00:...	Nam	NULL	GV00010
GV00007	Trần Duy Tân	1978-01-04 00:...	Nam	NULL	GV00002
GV00008	Nguyễn Thị Linh	1979-07-08 00:...	Nữ	0938079700	GV00009
GV00009	Trần Thị Kiều	1977-01-03 00:...	Nữ	NULL	NULL
GV00010	Trần Phương Loan	1978-04-30 00:...	Nữ	NULL	NULL

NHẬP DỮ LIỆU VÀO BẢNG



❑ Một số lưu ý

- Nhập dữ liệu khi đã có ràng buộc khóa ngoại

Column Name	Condensed Type
MaGV	nchar(10)
TenGV	nvarchar(50)
NgaySinh	datetime
GioiTinh	nvarchar(10)
DienThoai	nchar(10)
MaGVQuanLi	nchar(10)

❖ Cách 2:

B1: Nhập GIAOVIEN đặt **MaGVQuanLi=NULL**.

B2: Cập nhật cột **MaGVQuanLi** trong bảng GIAOVIEN.

NHẬP DỮ LIỆU VÀO BẢNG



❑ Nhập từ một bảng khác

INSERT INTO SV_08CDTH1

SELECT MASV, HOTEN

FROM SINHVIEN, LOP

WHERE SINHVIEN.MALOP=LOP.MALOP AND

TENLOP='08CDTH1'

3. BẢNG (TABLE)



❑ Cập nhật dữ liệu vào bảng

UPDATE ⟨Tên_bảng⟩

SET ⟨Tên_thuộc tính1⟩ = ⟨GT1⟩, ..., ⟨Tên_thuộc
tínhN⟩ = ⟨GTN⟩

[**Where** ⟨Điều kiện⟩]

3. BẢNG (TABLE)



❑ Xóa dữ liệu khỏi bảng

DELETE FROM <Tên_bảng>

[**Where** <Điều kiện chọn>]

Lưu ý: Câu lệnh **Delete** chỉ xóa dữ liệu của bảng, **cấu trúc bảng** tạo bởi câu lệnh **Create Table** vẫn giữ nguyên.