

Lập trình trên SQL Server

TS. LẠI HIỀN PHƯƠNG

EMAIL: LHPHUONG@TLU.EDU.VN

Nội dung

- Các kiểu dữ liệu trong SQL Server
- Cơ sở dữ liệu trong SQL Server
- Bảng trong SQL Server
- Các hàm trong SQL Server
- Câu lệnh điều khiển
- Thủ tục và hàm người dùng

Các kiểu dữ liệu trong SQL Server

Giới thiệu chung

- Kiểu dữ liệu (Data type) quy định về cấu trúc, miền giá trị của dữ liệu có thể nhập vào và tập các phép toán/toán tử có thể tác động lên miền giá trị đó
- Trong SQL, kiểu dữ liệu quyết định dạng dữ liệu của từng Cột (thuộc tính dữ liệu) cũng như kích thước sử dụng trong vùng nhớ của thuộc tính đó.
- Cần xác định kiểu dữ liệu thích hợp cho từng thuộc tính dữ liệu để đảm bảo tối ưu bộ nhớ trong quá trình sử dụng
 - Ví dụ:
 - thuộc tính « Ngày Sinh » cần kiểu dữ liệu ngày tháng
 - Thuộc tính « Họ Tên » cần kiểu dữ liệu dạng chuỗi ký tự
 - Thuộc tính « Lương » cần kiểu dữ liệu số

Danh sách các kiểu dữ liệu trong SQL Server

| Kiểu dữ liệu | Kích thước | Miền giá trị dữ liệu lưu trữ |
|--------------------------------------|------------|--------------------------------------|
| > Các kiểu dữ liệu dạng số nguyên | | |
| Int | 4 bytes | từ -2,147,483,648 đến +2,147,483,647 |
| SmallInt | 2 bytes | từ -32768 đến +32767 |
| TinyInt | 1 byte | từ 0 đến 255 |
| Bit | 1 byte | Nhận giá trị 0, 1 hoặc Null |
| > Các kiểu dữ liệu dạng số thập phân | | |
| Decimal, Numeric | 17bytes | từ -10^{38} đến $+10^{38}$ |
| > Các kiểu dữ liệu dạng số thực | | |
| Float | 8 bytes | từ $-1.79E+308$ đến $+1.79E+308$ |
| Real | 4 bytes | từ $-3.40E+38$ đến $+3.40E+38$ |

Danh sách các kiểu dữ liệu trong SQL Server (tiếp)

> Các kiểu dữ liệu dạng chuỗi có độ dài cố định

| | | |
|---------|---------|---|
| Char(N) | N bytes | Kiểu ký tự, cấp phát tĩnh N ô nhớ Từ 1 đến 8000 ký tự, mỗi ký tự là một byte |
|---------|---------|---|

> Các kiểu dữ liệu dạng chuỗi có độ dài biến đổi

| | | |
|------------|---------|--|
| VarChar(N) | N bytes | Kiểu ký tự, cấp phát động không vượt quá N ô nhớ Từ 1 đến 8000 ký tự, mỗi ký tự là 1 byte |
| Text | | Kiểu văn bản, chuỗi ký tự, cấp phát động theo độ dài chuỗi ký tự nhập vào |

> Các kiểu dữ liệu dạng chuỗi dùng font chữ Unicode

| | | |
|-------------|-----------|---|
| Nchar(N) | 2*N bytes | Kiểu ký tự, cấp phát tĩnh N ký tự Từ 1 đến 4000 ký tự, mỗi ký tự là 2 bytes |
| NVarChar(N) | 2*N bytes | Kiểu ký tự, cấp phát động không vượt quá N ký tự Từ 1 đến 4000 ký tự, mỗi ký tự là 2 bytes |
| NText | | Kiểu văn bản, chuỗi ký tự, cấp phát động theo độ dài chuỗi ký tự nhập vào |

Danh sách các kiểu dữ liệu trong SQL Server (tiếp)

> Các kiểu dữ liệu dạng tiền tệ

| | | |
|------------|---------|--|
| Money | 8 bytes | từ -922,337,203,685,477.5808 đến +922,337,203,685,477.5807 |
| SmallMoney | 4 bytes | từ -214,748.3648 đến + 214,748.3647 |

> Các kiểu dữ liệu dạng ngày và giờ

| | | |
|---------------|---------|------------------------------|
| DateTime | 8 bytes | từ 01/01/1753 đến 31/12/9999 |
| SmallDateTime | 4 bytes | từ 01/01/1900 đến 06/06/2079 |

> Các kiểu dữ liệu dạng chuỗi nhị phân (Binary String)

| | | |
|-----------|---------|------------------------------|
| Binary | N bytes | từ 1 đến 8000 bytes |
| VarBinary | N bytes | từ 1 đến 8000 bytes |
| Image | N bytes | từ 1 đến 2,147,483,647 bytes |

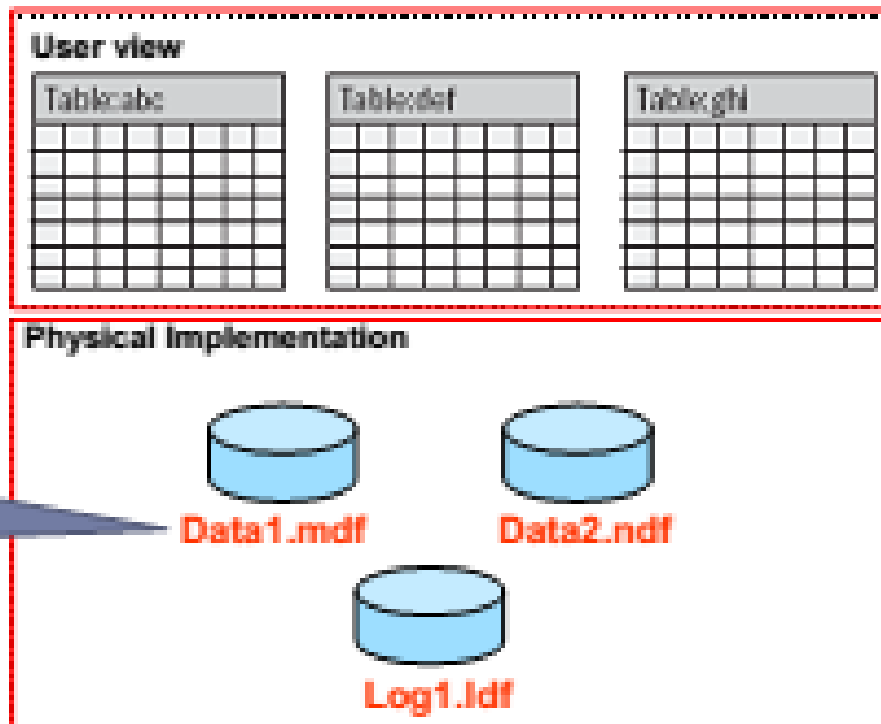
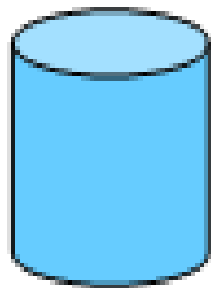
Cơ sở dữ liệu trong SQL Server

Giới thiệu chung

- SQL Server quản lý trực tiếp các CSDL, mỗi CSDL SQL Server sẽ quản lý các cấu trúc vật lý của nó
- Mỗi Server quản lý một danh sách các CSDL, tên các CSDL là duy nhất, không trùng nhau.
- Đặc điểm của việc quản trị CSDL:
 - Để khai thác CSDL, client phải kết nối đến Server quản trị CSDL đó
 - Chỉ khai thác các CSDL có trong server
 - Không có các phương thức mở CSDL trực tiếp từ tệp tin
 - Client chỉ thực hiện khai thác theo quy định đã định sẵn trong CSDL

Cấu trúc CSDL trong SQL Server

- Về mặt logic, CSDL gồm các table, view và các đối tượng
- Về mặt vật lý, CSDL được lưu trữ trên 2 hay nhiều tệp tin



Cơ sở dữ liệu

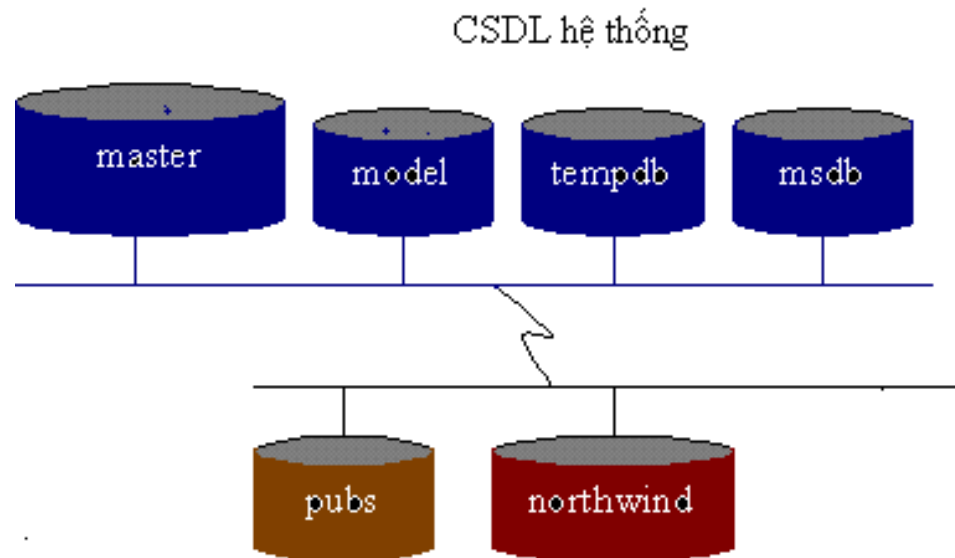
Mỗi CSDL chỉ có một file .mdf

Cấu trúc CSDL

- CSDL trong SQL Server lưu trữ theo 2 phần:
 - Phần dữ liệu: gồm
 - 1 tập tin bắt buộc (primary data file, *.mdf): là file chính chứa dữ liệu và các bảng, view hệ thống
 - 1 hoặc nhiều tập tin phụ (secondary data file *.ndf): là file phụ thường chỉ được sử dụng khi CSDL được phân chia để lưu trữ trên nhiều đĩa
 - Phần nhật ký:
 - 1 transaction log file (*.ldf): file ghi lại tất cả những thay đổi diễn ra trong 1 CSDL, chứa đầy đủ những thông tin để quay lui, backup hay phục hồi dữ liệu khi cần

Các loại CSDL trong SQL Server

- Có 3 kiểu CSDL trong SQL Server
 - CSDL hệ thống (system databases): master, model, tempdb, msdb
 - CSDL mẫu: pubs, northwind
 - CSDL do người dùng định nghĩa (user databases)



Các CSDL hệ thống

- **Master:** Chứa tất cả thông tin cấp hệ thống (system-level information) bao gồm thông tin về các CSDL trong hệ thống như vị trí các data files, các login account, các thiết lập cấu hình hệ thống của SQL Server.
- **Tempdb:** chứa tất cả các table hay stored procedure được tạo ra tạm thời trong quá trình làm việc bởi user hay SQL Server engine. Các table hay stored procedure này sẽ biến mất khi ta khởi động lại SQL Server hay khi ta disconnect.

Các CSDL hệ thống (tiếp)

- **Model:** CSDL này đóng vai trò như một bảng mẫu (template) cho các CSDL khác. Nghĩa là khi một CSDL của người dùng được tạo ra thì SQL Server sẽ copy toàn bộ các đối tượng hệ thống (tables, stored procedures,...) từ CSDL Model sang CSDL người dùng vừa mới tạo.
- **Msdb:** được sử dụng cho SQL Server Agent để lập lịch các công việc và các cảnh báo (schedule alerts and jobs)

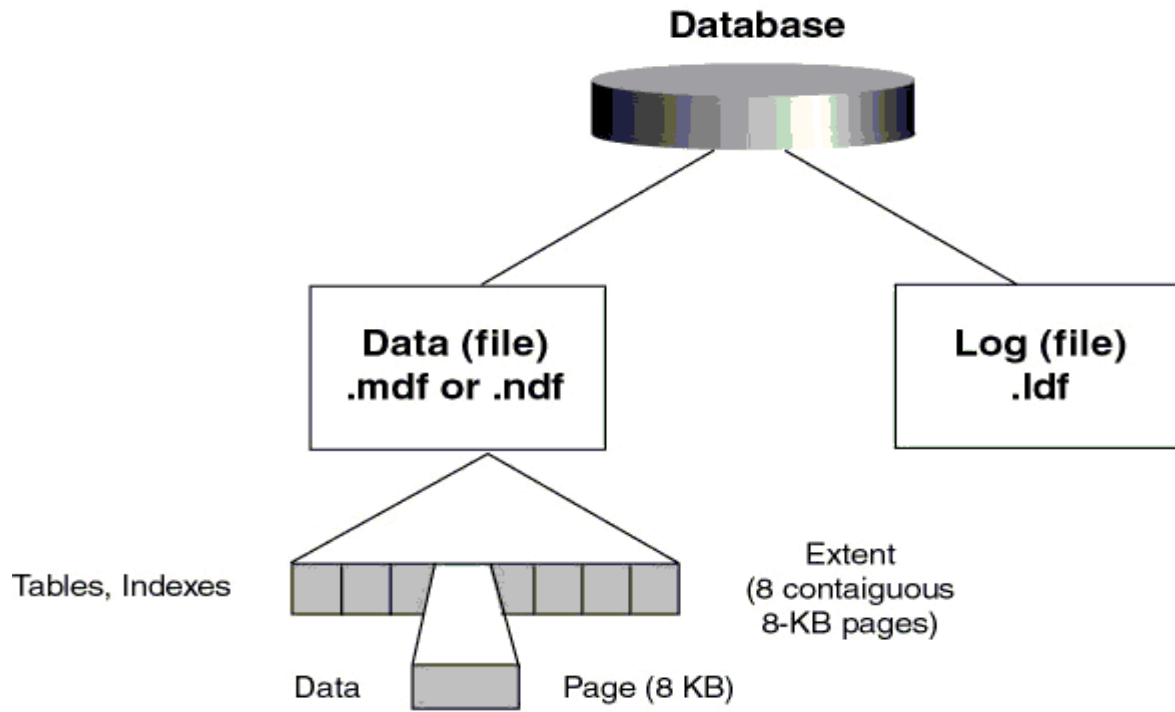
Các CSDL hệ thống (tiếp)

- Tập tin của các CSDL hệ thống như sau

| Tập tin CSDL | Tên tập tin vật lý | Kích thước ngầm định |
|---------------------|---------------------------|-----------------------------|
| master primary data | Master.mdf | 11.0 MB |
| master log | Mastlog.ldf | 1.25 MB |
| tempdb primary data | Tempdb.mdf | 8.0 MB |
| tempdb log | Templog.ldf | 0.5 MB |
| model primary data | Model.mdf | 0.75 MB |
| model log | Modellog.ldf | 0.75 MB |
| msdb primary data | Msdbdata.mdf | 12.0 MB |
| msdb log | Msdblog.ldf | 2.25 MB |

Cấu trúc vật lý của CSDL

- SQL Server quản lý tập tin dữ liệu của CSDL ở dạng vật lý theo trang (page) và phân đoạn (extent)



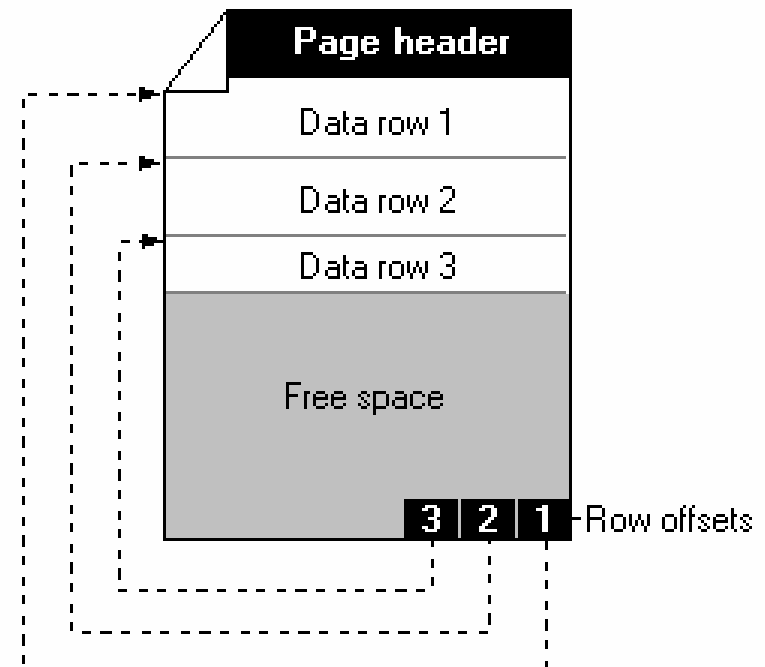
Max row size = 8060 bytes

Cấu trúc vật lý của CSDL (tiếp)

■ Trang (Page):

- Chỉ phần trang với các tập tin dữ liệu; các bản ghi được ghi lại liên tục không phân trang đối với các tập tin nhật ký (*.ldf)
- Mỗi page có kích thước 8KB
- Dữ liệu trong một trang sẽ bắt đầu lưu trữ sau phần thông tin Header, lưu trữ liên tiếp, mỗi hàng có kích thước tối đa 8060 byte.
- Dữ liệu trong SQL Server được lưu trữ trên đĩa và tạo chỉ mục Index theo cấu trúc B-tree Plus

Microsoft SQL Server Data Page

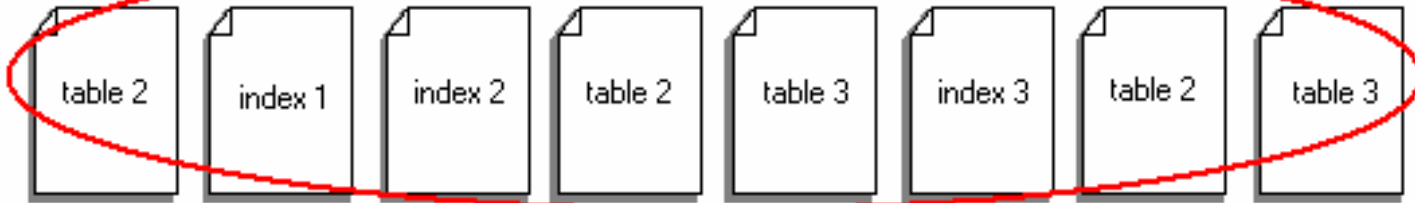


Cấu trúc vật lý của CSDL (tiếp)

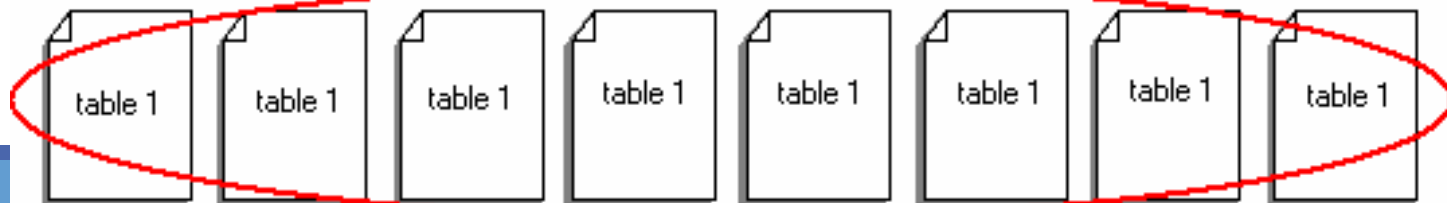
■ Phân đoạn (Extent):

- Extent là đơn vị dùng chứa các table và index, mỗi extent có 8 page hay 64KB.
- SQL Server có hai kiểu extent:
 - Uniform extent: chỉ dùng lưu trữ dữ liệu cho 1 đối tượng, 1 bảng
 - Mixed extent: dùng để lưu trữ dữ liệu cho nhiều bảng trên cùng 1 extent

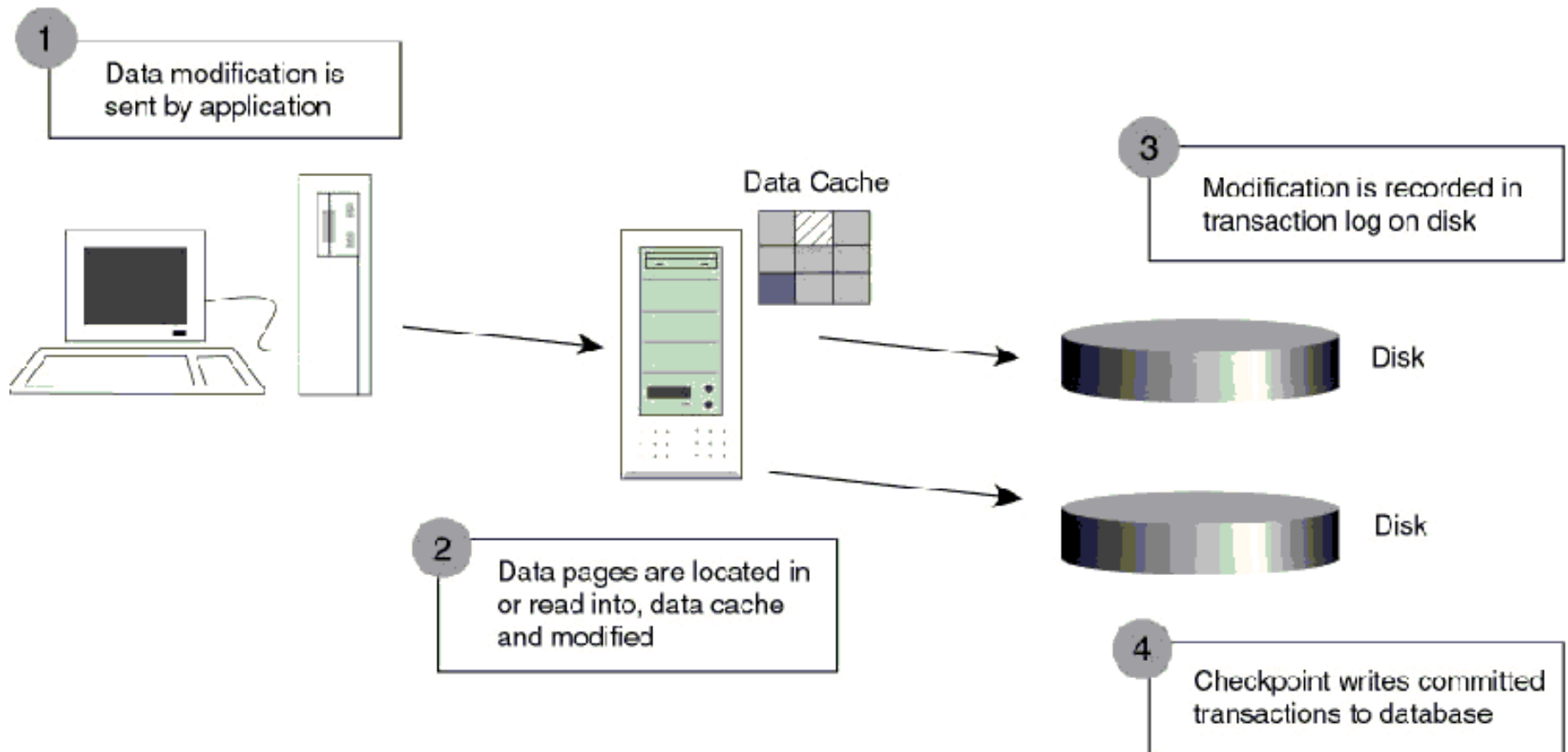
Mixed extent



Uniform extent



Quy trình thao tác dữ liệu



Cấu trúc logic của CSDL SQL Server

- Hầu như mọi thứ trong SQL Server được tổ chức thành những đối tượng (objects) ví dụ như tables, views, stored procedures, indexes, constraints,...
- Những system objects trong SQL Server thường có bắt đầu bằng chữ sys hay sp

Cấu trúc logic của CSDL SQL Server (tiếp)

- Một số system objects thường dùng:

| System Stored Procedure | Ứng dụng |
|---|---|
| Sp_help ['object'] | Cung cấp thông tin về một database object (table, view...) hay một data type. |
| Sp_helpdb ['database'] | Cung cấp thông tin về một database cụ thể nào đó. |
| Sp_monitor | Cho biết độ bận rộn của SQL Server |
| Sp_spaceused ['object', 'updateusage'] | Cung cấp thông tin về các khoảng trống đã được sử dụng cho một object nào đó |
| Sp_who ['login'] | Cho biết thông tin về một SQL Server user |

Cấu trúc logic của CSDL SQL Server (tiếp)

■ Ví dụ:

- Sp_helpdb 'Northwind' sẽ cho thông tin về CSDL Northwind

| name | db_size | owner | dbid | created | status |
|-----------|---------|-------|------|------------|---|
| <hr/> | | | | | |
| Northwind | 3.94 MB | sa | 6 | Aug 6 2000 | Status=ONLINE, Updateability=READ_WRITE, |

Quản lý CSDL trong SQL Server

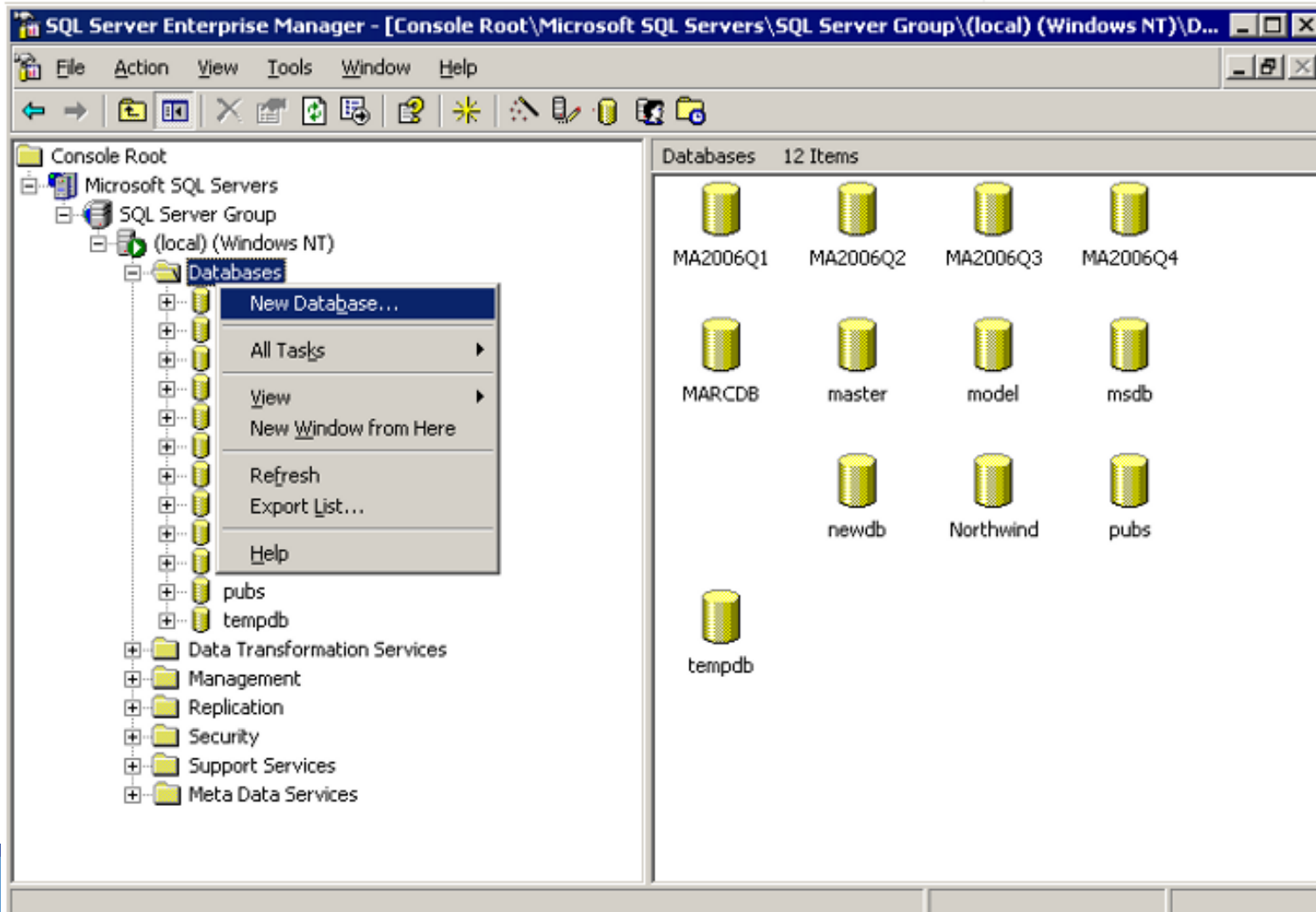
Tạo CSDL

- Để tạo CSDL, phải phân tích các thông tin liên quan mục đích sử dụng CSDL cho bài toán:
 - Tên của CSDL
 - Các bảng, các ràng buộc
 - Người sở hữu hoặc người tạo CSDL
 - Kích thước CSDL
 - Các tập tin hoặc nhóm tập tin cần dùng để lưu trữ CSDL
- Có 2 cách tạo CSDL:
 - Trong SQL Server Management Studio
 - Bằng câu lệnh T-SQL

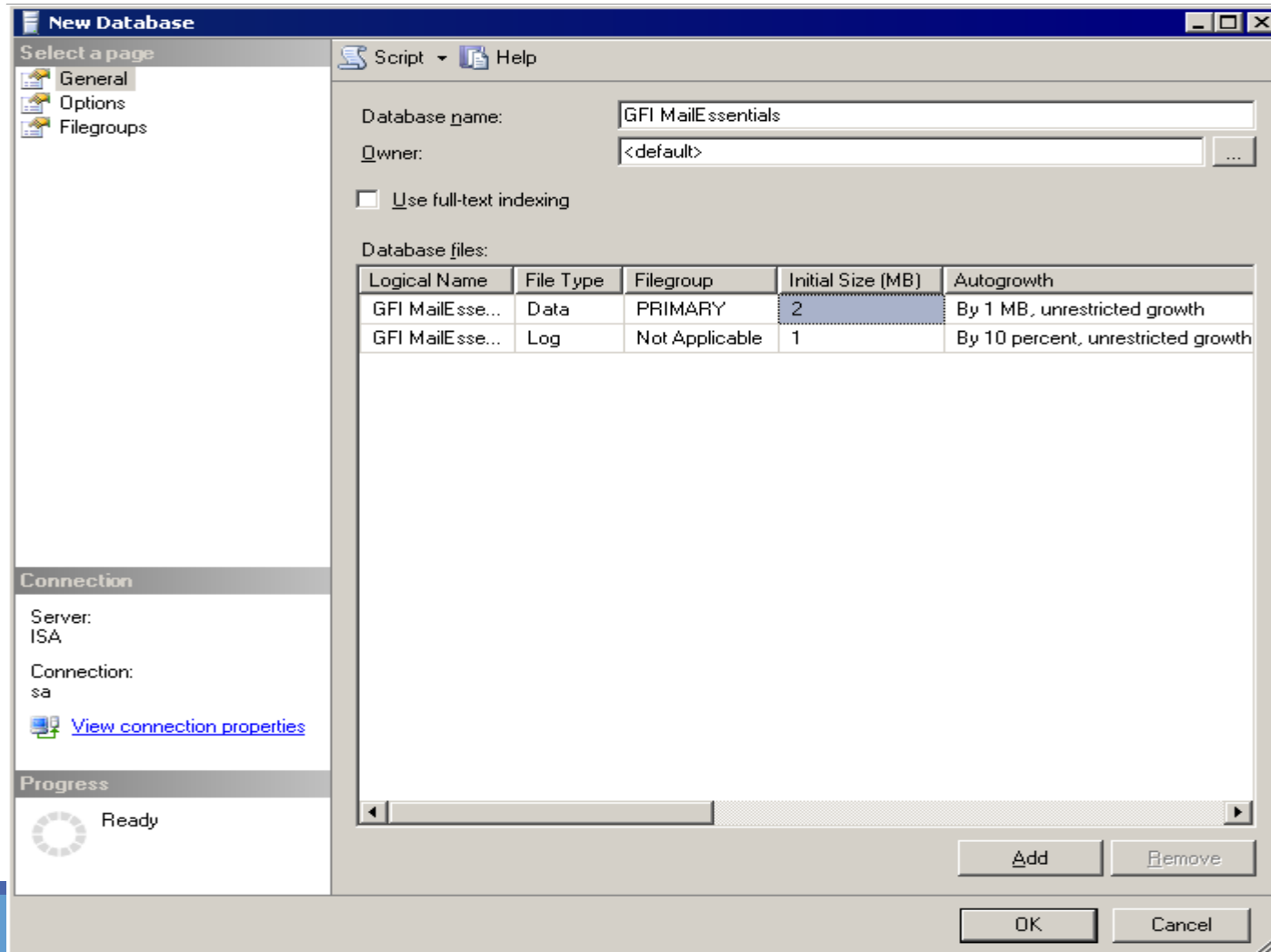
Tạo CSDL bằng SQL Server Management Studio

- Mở SQL Server Management Studio
- Kết nối bằng tài khoản người quản trị CSDL
- Kích chuột phải vào « Databases » và chọn « New Database »
- Nhập tên CSDL
- Kích đúp chuột vào CSDL vừa tạo để mở hộp thoại chi tiết CSDL
- Ở thẻ General, ta có các thông tin chung về CSDL
- Kích chuột vào các thẻ và hộp thích hợp, thực hiện các thay đổi và nhấn OK

Tạo CSDL bằng SQL Server Management Studio (tiếp)



Tạo CSDL bằng SQL Server Management Studio (tiếp)



Tạo CSDL bằng câu lệnh T-SQL

```
CREATE DATABASE Tên_CSDL
```

```
On Primary
```

```
(
```

```
    Name=logical_file_name ,
```

```
-- Tên file logic
```

```
    FileName='os_file_name'
```

```
-- Tên file vật lý
```

```
    Size=size [ KB | MB | GB | TB ] ,
```

```
-- Kích thước tập tin ban đầu
```

```
    MaxSize=max_size [ KB | MB | GB | TB ] | UNLIMITED], -- Kích thước tối đa
```

```
    FileGrowth=growth_increment
```

```
-- Kích thước tăng trưởng
```

```
)
```

```
Log On
```

```
(
```

```
    Name= logical_file_name,
```

```
    FileName='os_file_name',
```

```
    Size=size,
```

```
    MaxSize=max_size,
```

```
    FileGrowth= growth_increment
```

```
);
```

Tạo CSDL bằng câu lệnh T-SQL (tiếp)

Tên_CSDL: tên của CSDL

On Primary: mô tả primary file của CSDL

- *Name*: tên primary file
- *File name*: đường dẫn của primary file
- *Size*: kích thước của primary file
- *MaxSize*: kích thước lớn nhất của primary file
- *Filegrowth*: chỉ định độ tăng nào được sử dụng đối với việc tự động phát triển của primary file

Log On: mô tả file nhật ký của CSDL

Tạo CSDL bằng câu lệnh T-SQL (tiếp)

- **Ví dụ:** Tạo CSDL tên là QLSV dành ra 10MB lúc đầu cho phần dữ liệu và 5MB cho phần nhật ký chuyển tác . Các tập tin có thể phát triển lên đến 50 MB với phần dữ liệu, 25 MB với phần nhật ký chuyển tác. Các file được lưu trữ trong D:\HQTCSDL\ThucHanh

Tạo CSDL bằng câu lệnh T-SQL (tiếp)

CREATE DATABASE QLSV

On Primary

(Name=QLSV_Data,

Filename= 'D:\HQTCSDL\ThucHanh\QLSV_dat.mdf',

Size=10 MB,

MaxSize=50MB)

Log On

(Name = QLSVLog,

FileName='D:\HQTCSDL\ThucHanh\QLSV_log.ldf',

Size=5MB,

MaxSize=25MB)

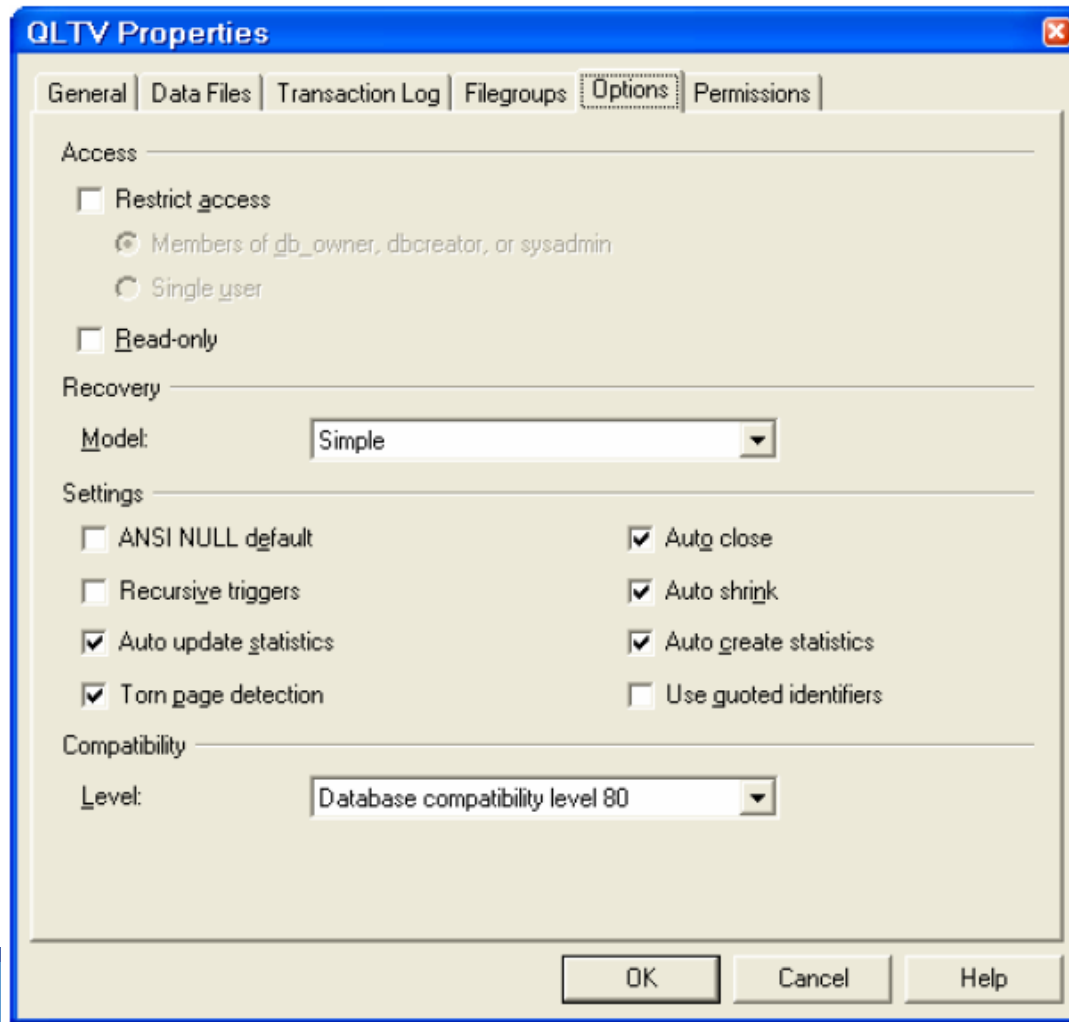
Xóa CSDL

- Xóa bằng SQL Server Management Studio
 - Trong Database, kích chuột phải vào CSDL muốn xóa rồi chọn « Delete »
 - Sau đó chọn « Yes »
- Xóa bằng câu lệnh T-SQL
 - Cú pháp: Drop DATABASE Tên_CSDL
 - Ví dụ: Drop DATABASE QLSV

Sửa tham số CSDL

- Sửa bằng SQL Server Management Studio
 - Trong Database, kích chuột phải vào CSDL muốn sửa rồi chọn « Properties »
 - Sửa các tham số cần thiết sau đó chọn « OK »
- Sửa bằng câu lệnh T-SQL
 - Sử dụng câu lệnh ALTER DATABASE TênCSDL

Sửa tham số CSDL (tiếp)



Đổi tên CSDL

- Cú pháp 1

```
ALTER DATABASE database_name  
MODIFY NAME = new_database_name
```

- Cú pháp 2: Thực thi thủ tục lưu trữ hệ thống **sp_renamedb** để đổi tên CSDL

sp_renamedb 'oldName', 'newName'

- Chú ý: muốn đổi tên CSDL, phải tắt hết các ứng dụng, các cửa sổ lệnh, view... hiện thời đang truy cập vào CSDL
- Ví dụ: Đổi tên CSDL QLSV thành QLSV_new

Đổi tên CSDL (tiếp)

- Ví dụ: Đổi tên CSDL QLSV thành QLSV_new

- Cú pháp 1:

```
ALTER DATABASE QLSV  
    MODIFY NAME = QLSV_new
```

- Cú pháp 2:

```
sp_renamedb 'QLSV', 'QLSV_new'
```

Hiệu chỉnh thuộc tính của tập tin CSDL

► Cú pháp:

```
ALTER DATABASE database_name
MODIFY FILE (
    NAME = logical_file_name ,
    FILENAME = 'os_file_name' ,
    SIZE = size ,
    MAXSIZE = max_size ,
    FILEGROWTH = size
)
```

Hiệu chỉnh thuộc tính của tập tin CSDL

- Ví dụ 1: Tăng kích thước của tập tin dữ liệu của CSDL QLSV lên 60MB
- Ví dụ 2: Hủy bỏ thuộc tính tự tăng trưởng của CSDL QLSV

Hiệu chỉnh thuộc tính của tập tin CSDL

- Ví dụ 1: Tăng kích thước của tập tin dữ liệu của CSDL QLSV lên 60MB

- Cú pháp:

```
ALTER DATABASE database_name  
MODIFY FILE (NAME = logical_file_name, SIZE = size)
```

- Thực hiện:

```
ALTER DATABASE QLSV  
MODIFY FILE (NAME = QLSV_data, SIZE = 60MB)
```

Hiệu chỉnh thuộc tính của tập tin CSDL

- Ví dụ 2: Hủy bỏ thuộc tính tự tăng trưởng của CSDL QLSV

- Cú pháp:

```
ALTER DATABASE database_name  
MODIFY FILE (NAME = logical_file_name, FILEGROWTH = size
```

- Thực hiện:

```
ALTER DATABASE QLSV  
MODIFY FILE (NAME = QLSV_data, FILEGROWTH = 0)
```


Hiệu chỉnh thuộc tính của tập tin CSDL

■Cú pháp giảm kích thước tập tin của CSDL

```
USE database_name  
DBCC SHRINKFILE (NAME = logical_file_name , size)
```

■Ví dụ: Giảm kích thước tập tin dữ liệu của CSDL QLSV xuống còn 20MB

```
USE QLSV  
DBCC SHRINKFILE (NAME = QLSV_data , 20)
```

Bổ sung thêm tập tin dữ liệu và tập tin nhật ký cho CSDL

Cú pháp

```
ALTER DATABASE database_name  
ADD FILE  
    (  
        NAME = logical_file_name ,  
        FILENAME = 'os_file_name' ,  
        SIZE = size ,  
        MAXSIZE = max_size ,  
        FILEGROWTH = max_size  
    )
```

Bổ sung thêm tập tin dữ liệu và tập tin nhật ký cho CSDL

■ Ví dụ: Thêm tập tin secondary vào CSDL QLSV

```
ALTER DATABASE QLSV
ADD FILE (
    NAME = QLSV_data2,
    filename = 'D:\HQTCSDDL\ThucHanh\QLSV_data2.ndf',
    size = 2,
    maxsize = 15,
    filegrowth = 2)
```

Gỡ bỏ một tập tin khỏi CSDL

■Cú pháp

```
ALTER DATABASE database_name  
    REMOVE FILE logical_file_name
```

- Chú ý:** tập tin phải trống trước khi được gỡ bỏ.
 - Sử dụng **DBCC SHRINKFILE** với tùy chọn **EMPTYFILE** để làm trống tập tin

Gỡ bỏ một tập tin khỏi CSDL (tiếp)

- **Ví dụ:** gỡ bỏ tập tin QLSV_data khỏi CSDL QLSV

```
USE QLSV GO
```

```
DBCC SHRINKFILE (  
    QLSV_data, EMPTYFILE  
)
```

```
GO
```

```
ALTER DATABASE QLSV  
    REMOVE FILE QLSV_data
```

Tạo bản chụp CSDL (Database snapshots)

- Bản chụp CSDL là bản chỉ đọc, khung nhìn tĩnh của CSDL SQL Server.
- Sau khi đã tạo bản chụp CSDL, trong trường hợp người dùng thực hiện một thao tác nhằm lẫn với CSDL gốc, CSDL gốc có thể phục hồi về trạng thái trước đó khi mà bản chụp đã được tạo.

Tạo bản chụp CSDL (Database snapshots)

■Cú pháp:

```
CREATE DATABASE database_snapshot_name
ON
(
NAME = logical_file_name,
FILENAME = 'os_file_name'
) [ ,...n ]
AS SNAPSHOT OF source_database_name [;]
```

Bảng trong SQL Server

Giới thiệu chung

- Bảng (Table) là nơi lưu trữ dữ liệu thật sự của CSDL. Dữ liệu được lưu trữ trên bảng theo dạng hàng, cột:
 - Mỗi cột tương ứng với một thuộc tính (attribute) của bảng
 - Mỗi hàng tương ứng với một bản ghi (record) của bảng
- Mỗi CSDL trong SQL Server có thể lưu được 2 tỷ bảng
- Mỗi bảng có thể có tối đa 1024 cột

Giới thiệu chung (tiếp)

- Số lượng các hàng và tổng kích thước của bảng được giới hạn bởi dung lượng cho phép của kho lưu trữ
- Kích thước tối đa cho mỗi hàng là 8060 bytes
- Mỗi bảng có thể chứa đến 249 chỉ mục loại nonclustered và chỉ mục loại clustered

Các bất thường khi cập nhật dữ liệu

- Ví dụ: với CSDL là gồm 1 bảng hóa đơn (INVOICE) cho khách hàng như sau:

| Acme Industries INVOICE | | | | |
|---------------------------------|--------------------------------|-----------------|-------------------------------|------------------------|
| Customer Number: 1454837 | | | Terms: Net 30 | |
| Customer: W. Coyote | | | Ship Via: USPS | |
| General Delivery | | | | |
| Falling Rocks, AZ 84211 | | | Order Date: 12/01/2008 | |
| (599) 555-9345 | | | | |
| <u>Product No.</u> | <u>Description</u> | <u>Quantity</u> | <u>Unit Price</u> | <u>Extended Amount</u> |
| SPR-2290 | Super Strength Springs | 2 | 24.00 | \$48.00 |
| STR-67 | Foot Straps, leather | 2 | 2.50 | \$5.00 |
| HLM-45 | Deluxe Crash Helmet | 1 | 67.88 | \$67.88 |
| SFR-1 | Rocket, solid fuel | 1 | 128,200.40 | \$128,200.40 |
| ELT-1 | Emergency Location Transmitter | 1 | 79.88 | **FREE GIFT** |
| TOTAL ORDER AMOUNT: | | | | \$128,321.28 |

Các bất thường khi cập nhật dữ liệu (tiếp)

- Ví dụ: Bảng hóa đơn (INVOICE) như sau:

INVOICE

| Customer Number | Customer | Terms | Ship Via | Order Date | Product No. | Description | Quantity | Unit Price | Extended Amount |
|-----------------|----------|--------|----------|------------|-------------|--------------------------------|----------|------------|-----------------|
| 1454837 | W.Coyote | Net 30 | USPS | 12/01/2008 | SPR-2290 | Super Strength Springs | 2 | 24.00 | \$48.00 |
| 1454837 | W.Coyote | Net 30 | USPS | 12/01/2008 | SRT-67 | Foot Straps, leather | 2 | 2.50 | \$5.00 |
| 1454837 | W.Coyote | Net 30 | USPS | 12/01/2008 | HLM-45 | Deluxe Crash Helmet | 1 | 67.88 | \$67.88 |
| 1454837 | W.Coyote | Net 30 | USPS | 12/01/2008 | SFR-1 | Rocket, Solid fuel | 1 | 128,200.4 | \$128,200.4 |
| 1454837 | W.Coyote | Net 30 | USPS | 12/01/2008 | ELT-1 | Emergency Location Transmitter | 1 | 79.88 | \$79.88 |

Các bất thường khi cập nhật dữ liệu (tiếp)

- Bất thường khi thêm dữ liệu:
 - Không thể thêm một khách hàng vào CSDL nếu khách hàng không mua một mặt hàng nào
- Bất thường khi cập nhật dữ liệu:
 - Muốn thay đổi địa chỉ của khách hàng, phải thay đổi trên tất cả các hóa đơn của khách hàng đó bởi vì địa chỉ của khách hàng lưu trữ dư thừa trong tất cả các hóa đơn của khách hàng
- Bất thường khi xóa dữ liệu:
 - Nếu xóa hóa đơn cuối cùng của khách hàng thì thông tin về khách hàng đó bị mất

Chuẩn hóa CSDL

- Trong thiết kế CSDL, việc tuân thủ ngặt nghèo những chuẩn tắc chuẩn hóa CSDL giúp cho việc quản trị dữ liệu có hiệu quả, khắc phục dư thừa dữ liệu, loại bỏ các bất thường khi cập nhật CSDL.
- Cần có các bước chuẩn hóa dữ liệu từ một CSDL chưa chuẩn hóa sang chuẩn hóa:
 - Dạng chưa chuẩn hóa (unnormalized form – UNF): quan hệ chưa chuẩn hóa là quan hệ chứa các bộ dữ liệu bị lặp lại giá trị.

Chuẩn hóa CSDL (tiếp)

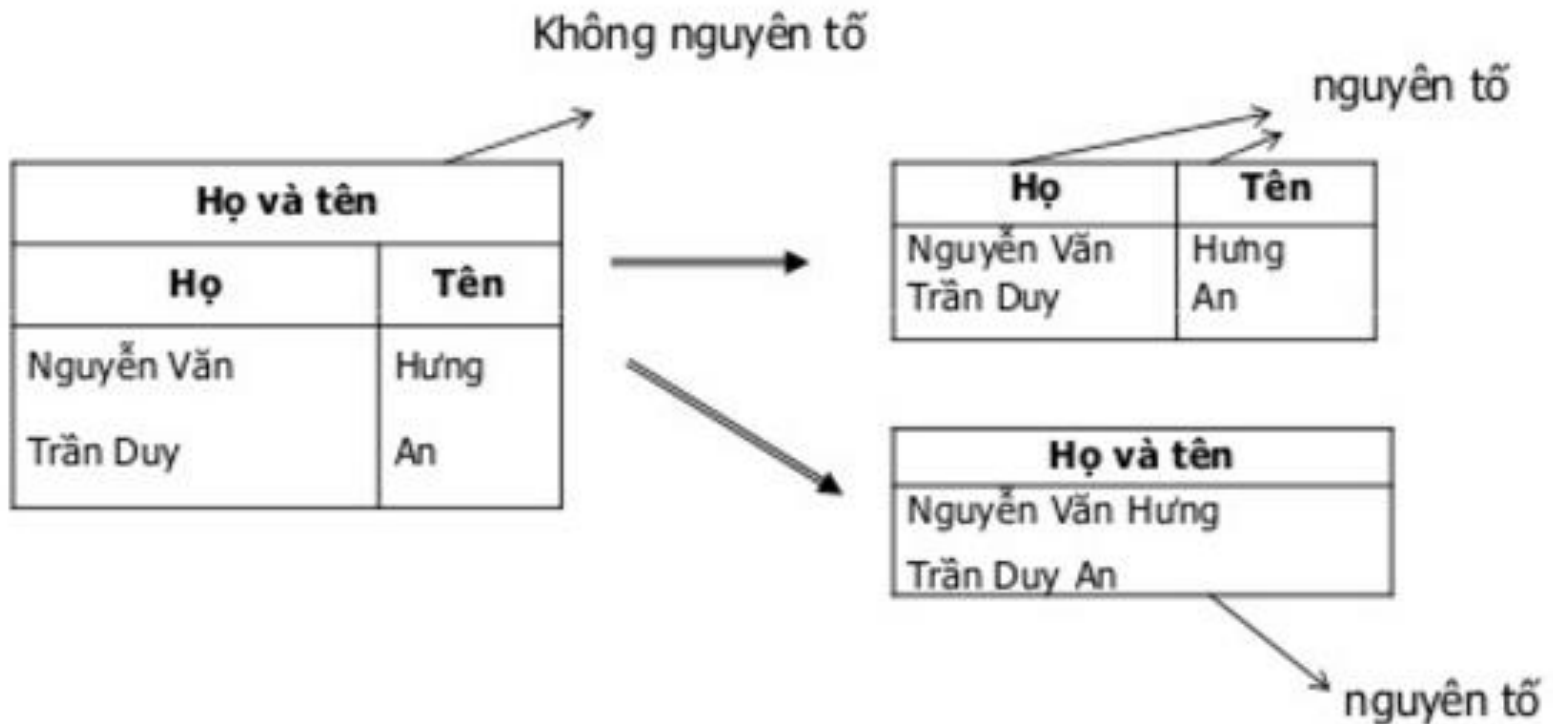
- Chuẩn hóa là quá trình phân rã lược đồ quan hệ nhằm đảm bảo các lược đồ quan hệ thỏa mãn hai tính chất
 - Trùng lặp dữ liệu ít nhất
 - Khả năng gây ra bất thường khi cập nhật được giảm thiểu
- Các dạng chuẩn:
 - Dạng chuẩn 1 (1NF)
 - Dạng chuẩn 2 (2NF)
 - Dạng chuẩn 3 (3NF)

Dạng chuẩn 1 (1NF)

- Lược đồ quan hệ R của một CSDL được gọi là 1NF nếu tất cả các thuộc tính của R thỏa mãn cả 3 điều kiện sau:
 - Là nguyên tố (không là danh sách các giá trị phức hợp)
 - Giá trị của các thuộc tính trên các bộ là đơn vị, không chứa nhóm lặp
 - Không có một thuộc tính nào mà giá trị có thể tính toán được từ một số thuộc tính khác

Dạng chuẩn 1 (1NF) (tiếp)

- Chuyển các thuộc tính không nguyên tố thành nguyên tố



Dạng chuẩn 1 (1NF) (tiếp)

- Loại bỏ các thuộc tính lặp hoặc đa trị:
 - tách các thuộc tính lặp/đa trị sang bảng mới, khóa của bảng mới là khóa của bảng ban đầu và khóa của các thuộc tính lặp/đa trị

| DONVI | Mã số DV | Tên DV | Mã số NQL | Địa điểm |
|-------|----------|------------|-----------|----------------------------|
| | 5 | Nghiên cứu | NV002 | Nam định, Hà Nội, Bắc Ninh |
| | 4 | Hành chính | NV014 | Hà Nội |
| | 1 | Lãnh đạo | NV061 | Hà Nội |

Địa điểm: là thuộc tính đa trị

Bảng DONVI

| Mã số DV | Tên DV | Mã số NQL |
|----------|------------|-----------|
| 5 | Nghiên cứu | NV002 |
| 4 | Hành chính | NV014 |
| 1 | Lãnh đạo | NV061 |

Bảng DONVI ĐỊA ĐIỂM

| Mã số DV | Địa điểm |
|----------|----------|
| 5 | Nam định |
| 5 | Hà Nội |
| 5 | Bắc Ninh |
| 4 | Hà Nội |
| 1 | Hà Nội |

Dạng chuẩn 1 (1NF) (tiếp)

- Loại bỏ các thuộc tính lặp hoặc đa trị:

| NHÂNVIÊN_DỰÁN | MãsốDA | TênDA | Tênnhânviên | Sốgiờ |
|---------------|--------|-------|-------------|-------|
| | 1 | DA01 | Văn | 15 |
| | | | Nam | 20 |
| | 2 | DA02 | Nam | 10 |
| | | | Thanh | 12 |
| | 3 | DA03 | Bằng | 28 |
| | | | Thanh | 20 |
| | | | Văn | 12 |

Thuộc tính lặp là
{Tênnhânviên, Sốgiờ}

NHÂNVIÊN_DỰÁN

| <u>MãsốDA</u> | <u>Tênnhânviên</u> | <u>Sốgiờ</u> |
|---------------|--------------------|--------------|
| 1 | Văn | 15 |
| 1 | Nam | 20 |
| 2 | Nam | 10 |
| 2 | Thanh | 12 |
| 2 | Bằng | 28 |
| 3 | Thanh | 20 |
| 3 | Văn | 12 |

DỰÁN

| <u>MãsốDA</u> | <u>TênDA</u> |
|---------------|--------------|
| 1 | DA01 |
| 2 | DA02 |
| 3 | DA03 |

Dạng chuẩn 1 (1NF) (tiếp)

- Bảng INVOICE được viết thành quan hệ sau (thêm InvoiceNumber làm thuộc tính khóa cho bảng Invoice)

R(InvoiceNumber, Customer Number, Name, Address, City, State, Zip Code, Phone, Terms, Ship Via, Order Date, {Product Number, Description, Quantity, Unit Price, Extended Amount}, Total Order Amount)

- Quan hệ được tách thành

INVOICE(InvoiceNumber, Customer Number, Name, Address, City, State, Zip Code, Phone, Terms, Ship Via, Order Date, Total Order Amount)

INVOICE LINE ITEM (InvoiceNumber, Product Number, Description, Quantity, Unit Price, Extended Amount)

Dạng chuẩn 2 (2NF)

- Quan hệ R được gọi là đạt dạng chuẩn 2 nếu đã là 1NF và tất cả các thuộc tính không khóa đều phụ thuộc hàm đầy đủ vào khóa:
 - Thuộc tính không khóa là thuộc tính không tham gia vào bất kỳ khóa nào
- Chuẩn hóa từ 1NF – 2NF:
 - Tách các thuộc tính không khóa phụ thuộc vào một bộ phận của khóa chính thành một bảng riêng với khóa chính là bộ phận khóa mà chúng phụ thuộc vào

Dạng chuẩn 2 (2NF) (Tiếp)

■ Ví dụ:

| <u>MSKH</u> | TÊNKH | TP | PVC | <u>MSMH</u> | TÊNMH | ĐG | SL |
|-------------|-------|-----|-----|-------------|-------|-----|-----|
| S1 | An | HCM | 01 | P1 | Táo | 650 | 300 |
| S1 | An | HCM | 01 | P2 | Chanh | 500 | 200 |
| S1 | An | HCM | 01 | P3 | Cam | 450 | 400 |
| S2 | Hòa | HN | 02 | P1 | Táo | 650 | 100 |
| S2 | Hòa | HN | 02 | P3 | Cam | 450 | 300 |
| S3 | Thanh | NT | 03 | P2 | Chanh | 500 | 200 |
| S4 | Trang | NT | 03 | P2 | Chanh | 500 | 210 |

Tập phụ thuộc hàm:

$F = \{$

| | |
|-------------------|-------------------------|
| <u>MSKH</u> | \rightarrow TÊNKH, TP |
| <u>MSMH</u> | \rightarrow TÊNMH, ĐG |
| <u>MSKH, MSMH</u> | \rightarrow SL |
| TP | \rightarrow PVC} |

Dạng chuẩn 2 (2NF) (Tiếp)

- Ví dụ: quan hệ trên được tách thành

R₁

| <u>MSKH</u> | TÊNKH | TP | PVC |
|-------------|-------|-----|-----|
| S1 | An | HCM | 01 |
| S2 | Hoà | HN | 02 |
| S3 | Thanh | NT | 03 |
| S4 | Trang | NT | 03 |

$F_{R1} = \{MSKH \rightarrow TÊNKH, TP\}$

$TP \rightarrow PVC\}$

MH

| <u>MSMH</u> | TÊNMH | ĐG |
|-------------|-------|-----|
| P1 | Táo | 650 |
| P2 | Cam | 500 |
| P3 | Chanh | 450 |

$F_{MH} = \{MSMH \rightarrow TÊNMH, ĐG\}$

ĐH

| <u>MSKH</u> | <u>MSMH</u> | SL |
|-------------|-------------|-----|
| S1 | P1 | 300 |
| S1 | P2 | 200 |
| S1 | P3 | 400 |
| S2 | P1 | 100 |
| S2 | P3 | 300 |
| S3 | P2 | 200 |
| S4 | P2 | 210 |

$F_{DH} = \{MSKH, MSMH \rightarrow SL\}$

Dạng chuẩn 2 (2NF) (tiếp)

- CSDL INVOICE ở dạng chuẩn 1NF

R1-INVOICE(InvoiceNumber, Customer Number, Name, Address, City, State, Zip Code, Phone, Terms, Ship Via, Order Date, Total Order Amount)

R2-INVOICE LINE ITEM (InvoiceNumber, Product Number, Description, Quantity, Unit Price, Extended Amount)

Dạng chuẩn 2 (2NF) (tiếp)

- Ta thấy quan hệ R2-INVOICE LINE ITEM có phụ thuộc hàm vào một phần khóa

{Product Number} -> {Description, Unit Price}

- Quan hệ này được tách thành 2 quan hệ

PRODUCT (Product Number, Description, Unit Price)

R2-INVOICE LINE ITEM (InvoiceNumber, Product Number, Quantity, Extended Amount)

Dạng chuẩn 3 (3NF)

- Quan hệ R được gọi là đạt dạng chuẩn 3 nếu thỏa mãn các điều kiện sau:
 - R phải đạt dạng chuẩn 2NF
 - Mọi thuộc tính không khóa không phụ thuộc bắc cầu vào khóa (tức là tất cả các thuộc tính phải được suy ra trực tiếp từ khóa)

R_1

| <u>MSKH</u> | TÊNKH | TP | PVC |
|-------------|-------|-----|-----|
| S1 | An | HCM | 01 |
| S2 | Hoà | HN | 02 |
| S3 | Thanh | NT | 03 |
| S4 | Trang | NT | 03 |

$F_{R1} = \{MSKH \rightarrow TÊNKH, TP$

$TP \rightarrow PVC\}$

Dạng chuẩn 3 (3NF) (tiếp)

- Quy tắc chuẩn hóa về 3NF:
 - Loại bỏ các thuộc tính phụ thuộc bắc cầu ra khỏi quan hệ và tách chúng thành một quan hệ riêng có khóa chính là thuộc tính bắc cầu
 - Các thuộc tính còn lại lập thành 1 quan hệ có khóa chính là quan hệ ban đầu

| <u>MSKH</u> | TÊNKH | TP |
|-------------|-------|-----|
| S1 | An | HCM |
| S2 | Hòa | HN |
| S3 | Bình | NT |
| S4 | Trang | NT |

$$F_{KH} = \{MSKH \rightarrow TÊNKH, TP\}$$

| <u>TP</u> | PVC |
|-----------|-----|
| HCM | 01 |
| HN | 01 |
| NT | 03 |

$$F_{VC} = \{TP \rightarrow PVC\}$$

Dạng chuẩn 3 (3NF) (tiếp)

Cho quan hệ sau:

NHÂNVIÊN_DONVI(Mã số NV, Họ tên NV, Ngày sinh, Địa chỉ, Mã số DV, Tên DV, Mã số NQL)

Có chứa các phụ thuộc hàm dưới đây:

{Mã số NV} → {Họ tên NV, Ngày sinh, Địa chỉ, Mã số DV, Tên DV, Mã số NQL} Mã số NV là khóa chính của bảng

{Mã số DV} → {Tên DV, Mã số NQL}

Chứa phụ thuộc hàm bắc cầu vào thuộc tính không phải khóa

| <u>Mã số NV</u> | Họ tên NV | Ngày sinh | Địa chỉ | Mã số DV | Tên DV | Mã số NQL |
|-----------------|-----------|-----------|-----------------|----------|---------------|-----------|
| NV01 | Tùng | 1/1/1983 | 16 Thụy Khuê HN | DV01 | Phòng tin học | NV02 |
| NV02 | Cúc | 2/2/1980 | 10 Hàng Đào HN | DV01 | Phòng tin học | NV02 |
| NV03 | Trúc | 3/3/1983 | Đà Nẵng | DV02 | Hành chính | NV05 |
| NV04 | Mai | 5/5/1982 | Hà Giang | DV02 | Hành chính | NV05 |
| NV05 | Sen | 6/6/1986 | Hải Phòng | DV02 | Hành chính | NV05 |
| NV06 | Hồng | 6/8/1984 | Hà Nam | DV03 | Kinh doanh | NV06 |
| | | | | | | |

- Quan hệ này được tách thành các quan hệ sau

ĐƠN VỊ (Mã số DV, Tên DV, Mã số NQL)

NHÂN VIÊN (Mã số NV, Họ tên NV, Ngày sinh, Địa chỉ, Mã số DV)

Dạng chuẩn 3 (3NF) (tiếp)

- CSDL INVOICE ở dạng chuẩn 2NF

INVOICE(InvoiceNumber, Customer Number, Name, Address, City, State, Zip Code, Phone, Terms, Ship Via, Order Date, Total Order Amount)

PRODUCT (Product Number, Description, Unit Price)

INVOICE LINE ITEM (InvoiceNumber, Product Number, Quantity, Extended Amount)

Dạng chuẩn 3 (3NF) (tiếp)

- Ta thấy quan hệ INVOICE có chứa thuộc tính bắc cầu
{Customer Number} -> {Name, Address, City, State, Zip Code, Phone}

- Quan hệ này được tách thành 2 quan hệ

Customer(Customer Number, Name, Address, City, State, Zip Code, Phone)

INVOICE(InvoiceNumber, Customer Number, Terms, Ship Via, Order Date, Total Order Amount)

Thiết kế bảng dữ liệu

- Khi thiết kế bảng dữ liệu, cần xác định các thông tin sau:
 - Các cột, kiểu dữ liệu tương ứng (và độ dài nếu cần thiết)
 - Cột nào cho phép giá trị NULL
 - Giá trị ngầm định (là giá trị mà khi chưa nhập vào nó nhận giá trị này)
 - Các ràng buộc dữ liệu
 - Chỉ số Index, khóa chính, khóa ngoài

Ràng buộc dữ liệu

- Bằng cách thiết lập các ràng buộc thông tin trên CSDL, dữ liệu khi nhập vào CSDL sẽ được kiểm soát, độ tin cậy thông tin sẽ được đảm bảo.
- Có 4 loại toàn vẹn dữ liệu:
 - Toàn vẹn thực thể (Entry integrity)
 - Toàn vẹn theo miền (Domain integrity)
 - Toàn vẹn dạng tham chiếu (Referential integrity)
 - Toàn vẹn do người dùng định nghĩa (User-defined integrity)

Ràng buộc dữ liệu (tiếp)

- Toàn vẹn thực thể (Entity integrity): mỗi thực thể được xác định theo một khóa, khi biết khóa ta hoàn toàn có thể xác định được thực thể tương ứng. Khóa như vậy coi là khóa chính.
 - Ví dụ: Bảng SinhVien(MaSV, HoTen, NgaySinh, DiaChi) có MaSV là khóa chính.
- Toàn vẹn theo miền (Domain integrity): là loại toàn vẹn có hiệu ứng với các cột dữ liệu trong một phạm vi nào đó, ví dụ kiểu dữ liệu, ràng buộc theo khóa check là các dạng của toàn vẹn theo miền.
 - Ví dụ: Lương nhân viên phải lớn hơn lương tối thiểu

Ràng buộc dữ liệu (tiếp)

- Toàn vẹn dạng tham chiếu (Referential integrity): một mối quan hệ giữa hai bảng có một khóa chính và một khóa ngoài, khóa ngoài là khóa tham chiếu của khóa chính và giá trị của khóa ngoài sẽ thuộc tập các giá trị của khóa chính hoặc giá trị NULL.
 - Quan hệ giữa bảng SinhVien và bảng KetQua(MaSV, MaMon, Diem) xác định chỉ những sinh viên của trường mới có kết quả thi.
- Toàn vẹn do người dùng định nghĩa (User-defined integrity): là toàn vẹn do người dùng định nghĩa, quy định dữ liệu nhập vào theo quy cách, giá trị được kiểm soát chặt chẽ.
 - Ví dụ: MaSV phải chứa 10 chữ số.

Ràng buộc dữ liệu (tiếp)

- Các khóa, quy tắc, ràng buộc trong SQL Server tương ứng:

| Kiểu toàn vẹn | Công cụ trong SQL Server |
|------------------------|--|
| Entry integrity | <ol style="list-style-type: none">1. Ràng buộc Primary key2. Ràng buộc Unique3. Cột Identity |
| Domain integrity | <ol style="list-style-type: none">1. Giá trị ngầm định Default2. Ràng buộc khóa ngoài Foreign Key3. Ràng buộc Check4. Thuộc tính NOT NULL |
| Referential integrity | <ol style="list-style-type: none">1. Ràng buộc Foreign Key2. Ràng buộc Check |
| User-defined integrity | <ol style="list-style-type: none">1. Rules2. Stored procedures3. Triggers |

Các khóa

■ **Khóa chính (Primary Key):**

- Là một hoặc tổ hợp nhiều cột dữ liệu được sử dụng để định danh duy nhất mỗi bản ghi trong bảng.
- Dữ liệu của khóa chính phải có tính duy nhất và không có giá trị NULL
- Ví dụ: Bảng KetQua(**MaSV**, **MaMon**, Diem, GhiChu) có khóa chính {MaSV, MaMon} xác định duy nhất một sinh viên chỉ có 1 điểm thi duy nhất đối với một môn học.

■ **Khóa ngoại (Foreign Key)** của một bảng được xem như con trỏ trỏ tới khóa chính của bảng khác.

- Ví dụ: nếu trường MaSV trong bảng KetQua được sử dụng để tạo ràng buộc tham chiếu đến bảng SinhVien(**MaSV**, HoTen, NgaySinh, DiaChi) thì MaSV được gọi là khóa ngoại của bảng KetQua. Người sử dụng không thể nhập một giá trị MaSV không có trong bảng SinhVien vào bảng KetQua

Các khóa (tiếp)

- **Ràng buộc Unique:** ngăn cản việc hai bản ghi có giá trị đồng nhất trong một hoặc một tổ hợp cột dữ liệu cụ thể.
- **Ràng buộc Check:** là ràng buộc khống chế dữ liệu nằm trong một phạm vi nào đó. Ràng buộc này sẽ kiểm tra dữ liệu khi nhập vào.
- **Giá trị ngầm định – Default:** Gán giá trị mặc định cho một dữ liệu khi thêm bản ghi. Nếu người dùng không nhập dữ liệu vào cột thì giá trị mặc định sẽ được sử dụng cho cột đó

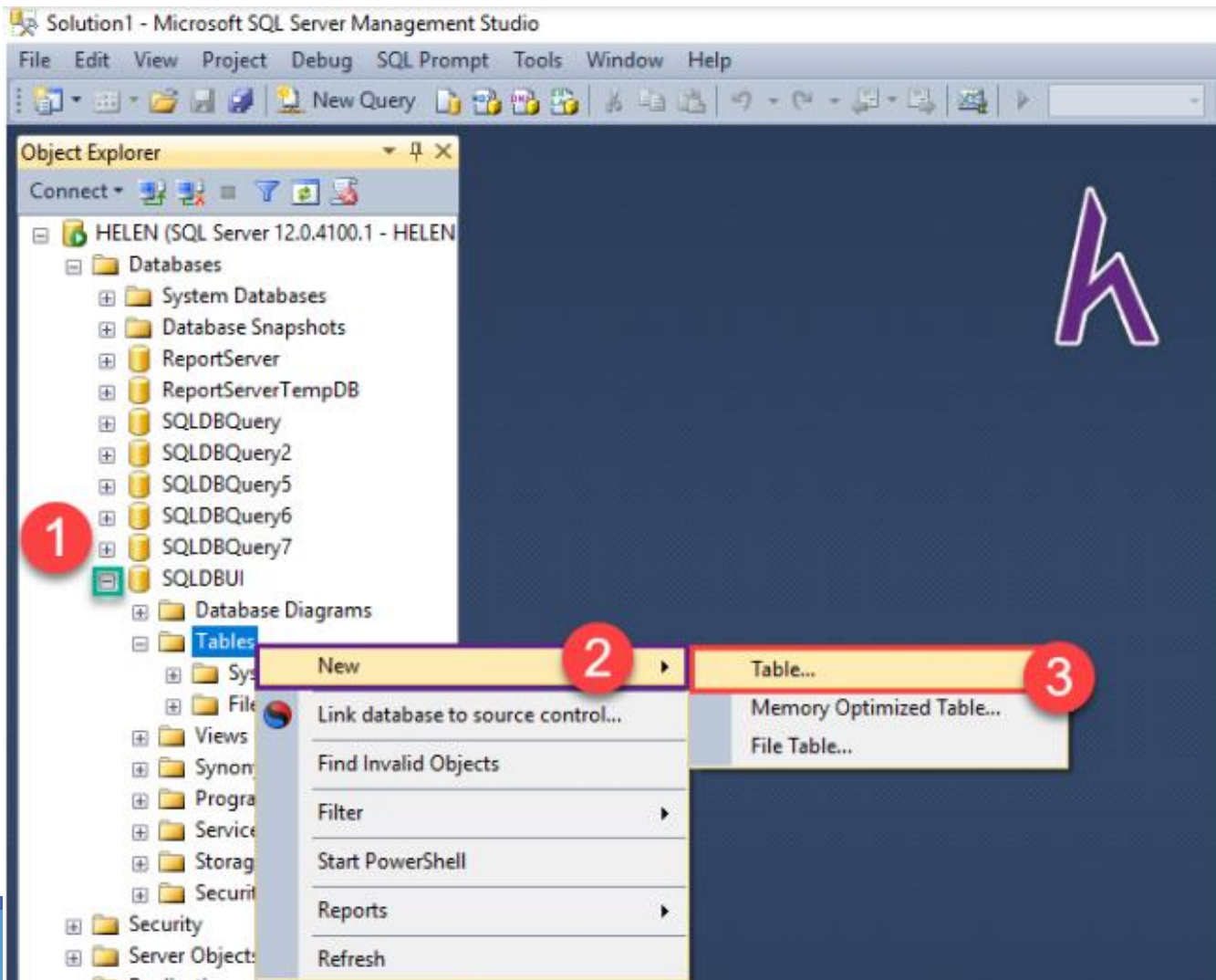
Tạo bảng dữ liệu

- Có 2 cách:
 - Sử dụng SQL Server Management Studio
 - Sử dụng câu lệnh T-SQL

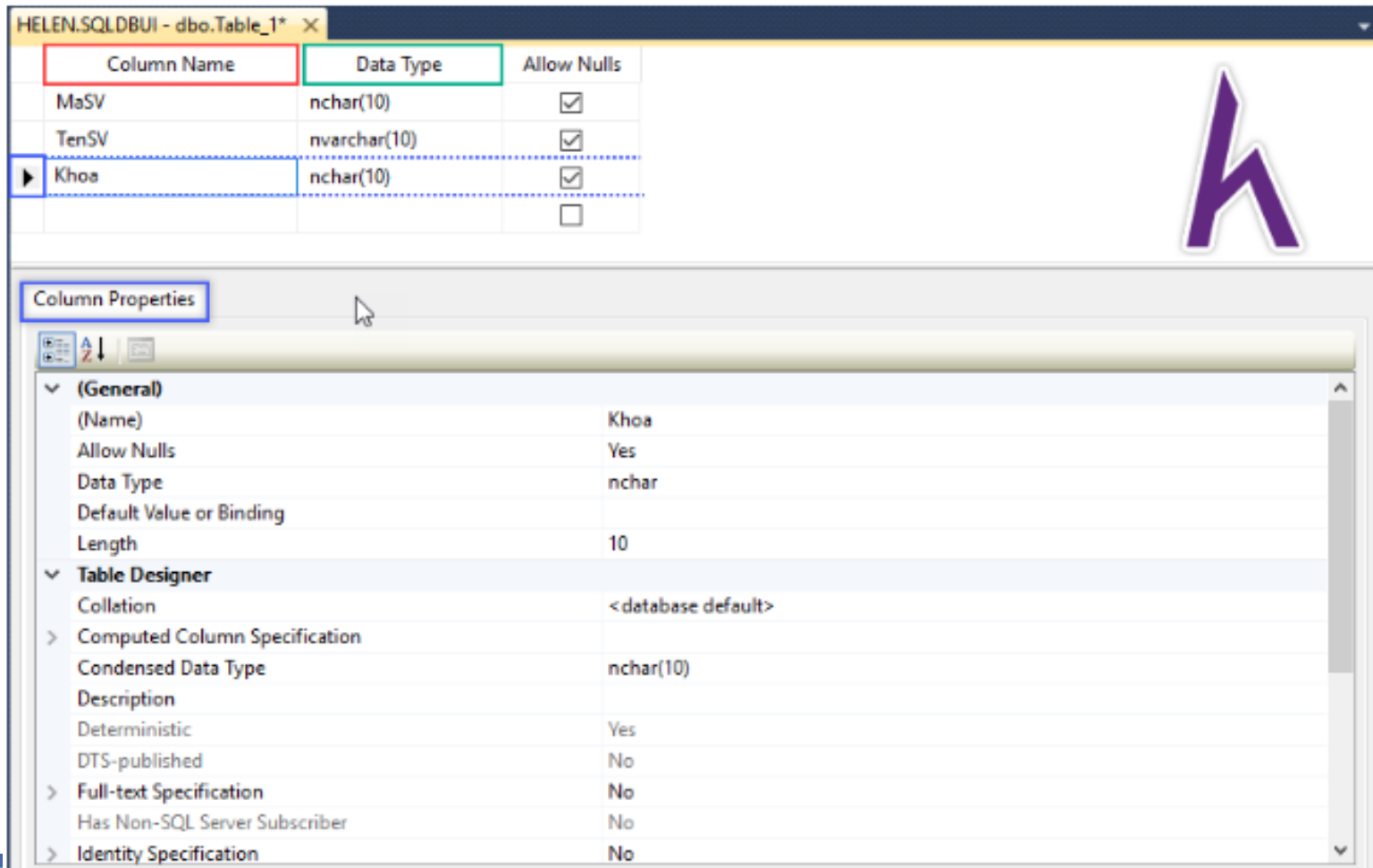
Tạo bảng sử dụng SQL Server Management Studio

- Nhấp dấu (+) bên trái Database cần tạo Bảng, tìm folder Table
- Nhấp chuột phải vào Table, chọn New-> Table
- Trong cửa sổ thông tin chi tiết của bảng, thêm các cột dữ liệu cần thiết trong đó:
 - Column Name: tên trường thuộc tính trong bảng
 - Data Type: Kiểu dữ liệu của các trường thuộc tính
 - Allow Nulls: xác định trường có nhận giá trị Null hay không
- Bước 4: Lưu bảng vừa tạo bằng cách chọn Save, nhập Tên bảng và nhấn OK.

Tạo bảng sử dụng SQL Server Management Studio (tiếp)

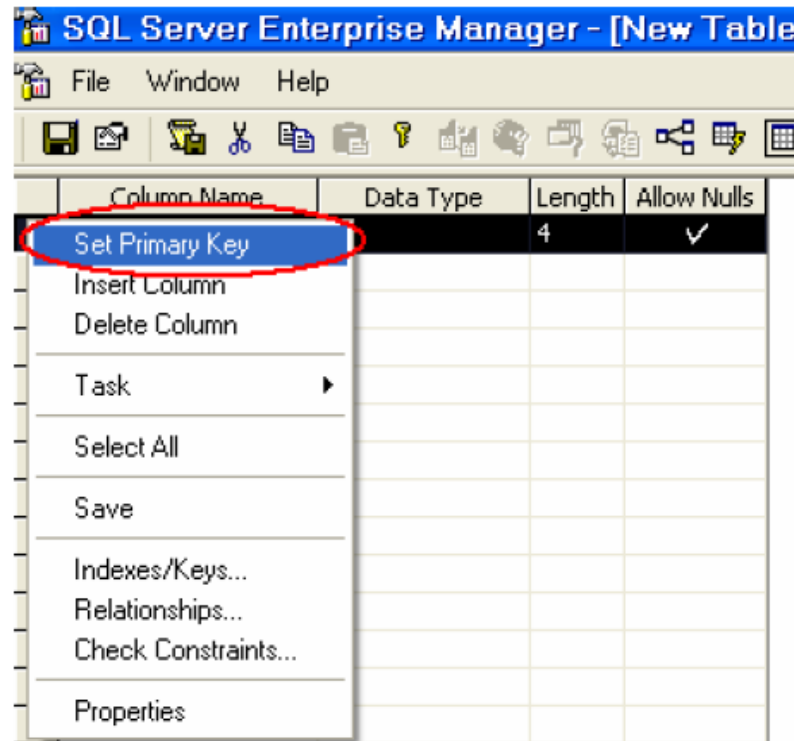


Tạo bảng sử dụng SQL Server Management Studio (tiếp)



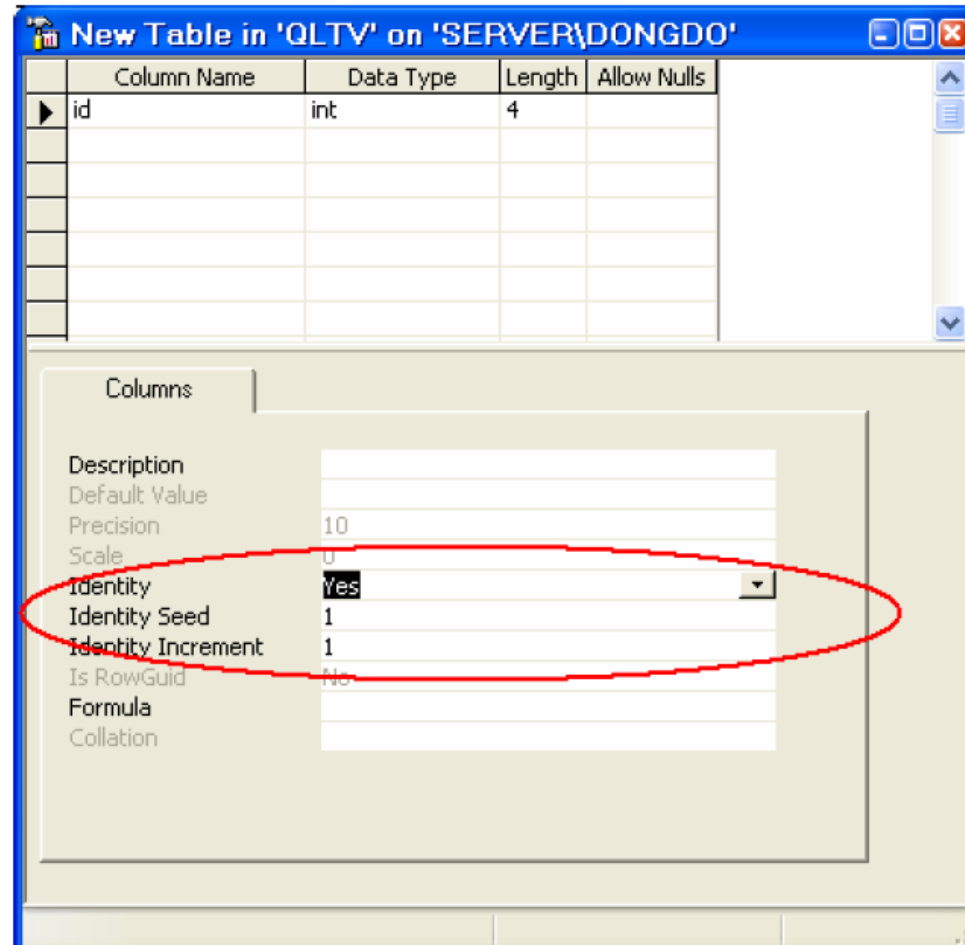
Tạo bảng sử dụng SQL Server Management Studio (tiếp)

- **Đặt khóa chính:** chọn những cột tham gia khóa, nhấn chuột phải, chọn Set primary key



Tạo bảng sử dụng SQL Server Management Studio (tiếp)

- **Xác định identity:** Chọn cột dữ liệu:
 - Đặt Yes cho Identity
 - Đặt Identity seed: giá trị ban đầu
 - Đặt Increment: bước tự động tăng



Tạo bảng bằng T-SQL

■Cú pháp:

[**USE** Tên_CSDL]

CREATE TABLE Tên_Bảng(
 Tên_cột_1 Kiểu_dữ_liệu [Tùy_chọn],
 Tên_cột_2 Kiểu_dữ_liệu [Tùy_chọn],
 ...
);

Tạo bảng bằng T-SQL (tiếp)

- **Ví dụ:** tạo trong cơ sở dữ liệu QLSV bảng SinhVien gồm các trường sau:
 - **MaSV** có kiểu dữ liệu số nguyên
 - **HoTen** có kiểu dữ liệu chuỗi ký tự, cấp phát động không vượt quá 30 ký tự
 - **NgaySinh** có kiểu dữ liệu Date
 - **DiaChi** có kiểu dữ liệu chuỗi ký tự, cấp phát động, không vượt quá 100 ký tự
 - **GioiTinh** có kiểu dữ liệu chuỗi ký tự, cấp phát động, không vượt quá 3 ký tự

Tạo bảng bằng T-SQL (tiếp)

- **Ví dụ:** bảng SinhVien được tạo như sau

```
USE QLSV
```

```
CREATE TABLE SinhVien(  
    MaSV int,  
    HoTen nvarchar(30),  
    NgaySinh date,  
    DiaChi nvarchar(100),  
    GioiTinh nvarchar(3)  
);
```

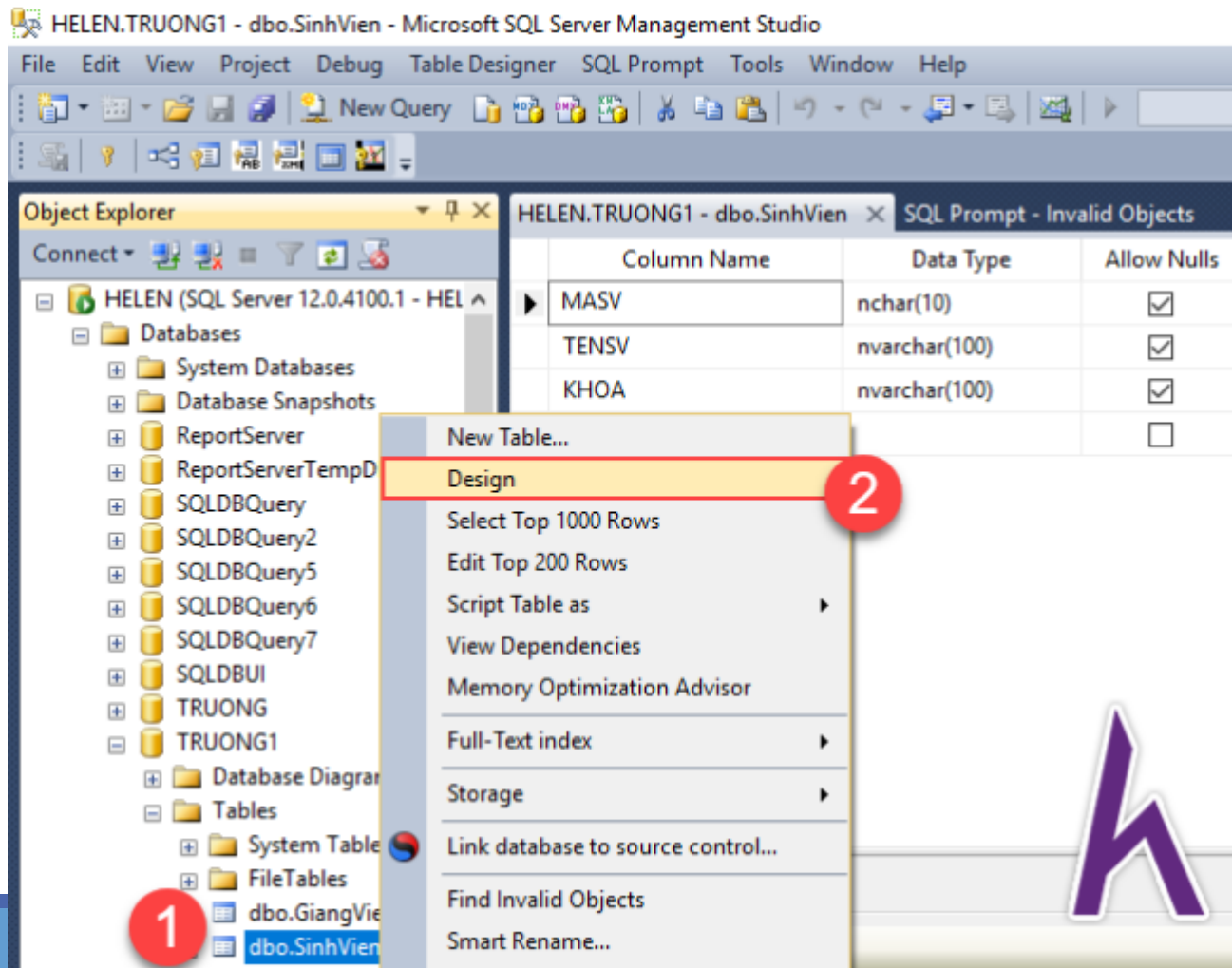
Xóa bảng dữ liệu

- Sử dụng SQL Server Management Studio:
 - Nhấn chuột phải vào tên bảng cần xóa
 - Chọn Delete và nhấn OK
- Sử dụng câu lệnh T-SQL
 - Cú pháp: `DROP TABLE Tên_bảng_cần_xóa`
 - Ví dụ:
 - Xóa bảng SinhVien: `DROP TABLE SinhVien;`
 - Xóa hai bảng SinhVien và KetQua:
`DROP TABLE SinhVien, KetQua;`

Sửa bảng dữ liệu

- Sử dụng SQL Server Management Studio:
 - Nhấn chuột phải vào tên bảng cần xóa
 - Chọn Design
 - Tiến hành thay đổi và nhấn OK
- Sử dụng câu lệnh T-SQL:
 - Sử dụng câu lệnh ALTER TABLE

Sửa bảng dữ liệu (tiếp)



Sửa bảng dữ liệu – ALTER TABLE

■ Thêm cột dữ liệu:

- Cú pháp:

[USE Tên_CSDL]

ALTER TABLE Tên_Bảng

ADD

Tên_cột_1 Kiểu_dữ_liệu [Tùy_chọn],

Tên_cột_2 Kiểu_dữ_liệu [Tùy_chọn],

...

;

Sửa bảng dữ liệu – ALTER TABLE

- **Thêm cột dữ liệu:**

- Ví dụ: thêm cột GhiChu có kiểu dữ liệu nvarchar(100) và cột SoDienThoai có kiểu dữ liệu nvarchar(11) vào bảng SinhVien

```
ALTER TABLE SinhVien  
ADD  
GhiChu nvarchar(100),  
SoDienThoai nvarchar(11);
```

Sửa bảng dữ liệu – ALTER TABLE

- **Sửa cột dữ liệu:**

- Cú pháp:

- [**USE** Tên_CSDL]

- ALTER TABLE** Tên_Bảng

- ALTER COLUMN** Tên_cột Kiểu_dữ_liệu [Tùy_chọn];

Sửa bảng dữ liệu – ALTER TABLE

- **Sửa cột dữ liệu:**

- Ví dụ: Sửa cột GhiChu trong bảng sinh viên số lượng ký tự là 50 thay vì 100 như trước

```
ALTER TABLE SinhVien
```

```
ALTER COLUMN
```

```
GhiChu nvarchar(50);
```

Sửa bảng dữ liệu – ALTER TABLE

- **Xóa cột dữ liệu:**

- Cú pháp:

- [USE Tên_CSDL]

- ALTER TABLE Tên_Bảng

- DROP COLUMN Tên_cột1 [, Tên_cột2,...];

- Ví dụ: Xóa cột dữ liệu GhiChu, SoDienThoai trong bảng SinhVien

- ALTER TABLE SinhVien

- DROP COLUMN GhiChu, SoDienThoai;

Tùy chọn Cột được phép null

- Cột được phép null:
 - Khả năng Null của một cột quyết định các hàng trong bảng có thể chứa giá trị Null cho cột đó.
 - Khả năng null của một cột có thể được định nghĩa khi tạo một bảng hoặc định dạng một bảng:
 - Từ khóa NULL : cho phép nhận giá trị NULL
 - Từ khóa NOT NULL: không cho phép nhận giá trị NULL

Tùy chọn Cột được phép null (tiếp)

- **Ví dụ:** viết lại câu lệnh tạo bảng SinhVien trong đó
 - các cột MaSV, HoTen và GioiTinh không cho phép nhận giá trị NULL
 - các cột còn lại cho phép nhận giá trị NULL

```
USE QLSV
```

```
CREATE TABLE SinhVien(  
    MaSV int NOT NULL,  
    HoTen nvarchar(30) NOT NULL,  
    NgaySinh date NULL,  
    DiaChi nvarchar(100) NULL,  
    GioiTinh nvarchar(3) NOT NULL  
);
```


Tùy chọn Cột được phép null (tiếp)

- **Ví dụ:** Viết câu lệnh ALTER TABLE cho bảng SinhVien sao cho
 - không cho phép cột GioiTinh nhận giá trị NULL
 - Cho phép cột DiaChi nhận giá trị NULL

```
ALTER TABLE SinhVien
```

```
ALTER COLUMN GioiTinh nvarchar(3) NOT NULL;
```

```
ALTER TABLE SinhVien
```

```
ALTER COLUMN DiaChi nvarchar(100) NULL;
```

Tùy chọn Định nghĩa DEFAULT

- Định nghĩa DEFAULT được sử dụng để định sẵn giá trị mặc định cho một cột.
 - Nếu không có giá trị nào được xác định trong khi khởi tạo, giá trị mặc định được gán cho cột đó.
 - DEFAULT có thể được định nghĩa khi tạo bảng hoặc khi sửa cấu trúc của bảng (ALTER Table) bằng cách thêm tùy chọn **DEFAULT giá_trị_mặc_định** vào phần định nghĩa cột

Tùy chọn Định nghĩa DEFAULT (tiếp)

- Ví dụ: viết lại câu lệnh tạo bảng SinhVien trong đó
 - Đặt giá trị mặc định cho cột giới tính là « Nam »

```
USE QLSV
```

```
CREATE TABLE SinhVien(  
    MaSV int NOT NULL,  
    HoTen nvarchar(30) NOT NULL,  
    NgaySinh date NULL,  
    DiaChi nvarchar(100) NULL,  
    GioiTinh nvarchar(3) NOT NULL DEFAULT 'Nam'  
);
```

Tùy chọn Định nghĩa DEFAULT (tiếp)

- Ví dụ: viết câu lệnh ALTER TABLE cho bảng SinhVien trong đó
 - Đặt giá trị mặc định cho cột giới tính là « Nam »

```
ALTER TABLE SinhVien  
ADD DEFAULT 'Nam' for GioiTinh;
```

Tùy chọn Thuộc tính Identity

- Thuộc tính IDENTITY của SQL Server được sử dụng để tạo ra các cột nhận dạng, chúng chứa các giá trị tự động phát sinh tuần tự để nhận dạng duy nhất mỗi hàng trong một bảng.
- Một thuộc tính nhận dạng có 2 phần:
 - Giá trị khởi đầu: seed_value
 - Giá trị tăng: increment_value
- Thuộc tính nhận dạng cho một cột có thể được định nghĩa khi tạo bảng hoặc sửa bảng bằng cách thêm tùy chọn

IDENTITY (seed_value, increment_value)

Tùy chọn

Thuộc tính Identity (tiếp)

- Ví dụ: viết lại câu lệnh tạo bảng SinhVien để thêm tùy chọn IDENTITY với giá trị khởi đầu là 0 và giá trị tăng là 1 cho thuộc tính MaSV

```
USE QLSV
```

```
CREATE TABLE SinhVien(  
    MaSV int NOT NULL IDENTITY (0,1),  
    HoTen nvarchar(30) NOT NULL,  
    NgaySinh date NULL,  
    DiaChi nvarchar(100) NULL,  
    GioiTinh nvarchar(3) NOT NULL DEFAULT 'Nam'  
);
```

Tùy chọn

Thuộc tính Identity (tiếp)

- **Ví dụ:** viết câu lệnh ALTER TABLE cho bảng SinhVien để thêm tùy chọn IDENTITY với giá trị khởi đầu là 0 và giá trị tăng là 1 cho thuộc tính MaSV

```
ALTER TABLE SinhVien  
ADD MaSV Int Identity(0,1);
```

- **Chú ý:** câu lệnh trên thêm thuộc tính MaSV vào bảng

Ràng buộc Khóa chính – Primary Key

- Để thiết lập khóa chính khi tạo bảng, sử dụng cú pháp

[**USE** Tên_CSDL]

CREATE TABLE Tên_Bảng(
Tên_cột1 Kiểu_dữ_liệu [Tùy_chọn],
Tên_cột2 Kiểu_dữ_liệu [Tùy_chọn],
...

PRIMARY KEY (Tên_cột_khóa_chính1, Tên_cột_khóa_chính1, ...)

);

Ràng buộc Khóa chính – Primary Key (Tiếp)

■ Hoặc cú pháp

[**USE** Tên_CSDL]

CREATE TABLE Tên_Bảng(
Tên_cột1 Kiểu_dữ_liệu [Tùy_chọn],
Tên_cột2 Kiểu_dữ_liệu [Tùy_chọn],
...
CONSTRAINT TenRangBuoc **PRIMARY KEY** (Tên_cột_khóa_chính1, Tên_cột_khóa_chính1, ...)
);

Ràng buộc

Khóa chính – Primary Key (tiếp)

- Ví dụ: Tạo bảng KetQua(MaSV, MaMon, Diem) với khóa chính gồm 2 trường MaSV và MaMon

```
CREATE TABLE KetQua(  
  MaSV int,  
  MaMon int,  
  Diem real,  
  PRIMARY KEY (MaSV, MaMon)  
);
```

```
CREATE TABLE KetQua  
(  
  MaSV int,  
  MaMon int,  
  Diem real  
  CONSTRAINT KhoaChinh PRIMARY KEY(MaSV, MaMon)  
);
```

Ràng buộc Khóa chính – Primary Key (tiếp)

- Để thiết lập khóa chính cho một bảng đã tạo:

[**USE** Tên_CSDL]

ALTER TABLE Tên_Bảng ADD PRIMARY KEY
(Tên_cột_khóa_chính1, Tên_cột_khóa_chính1, ...);

Hoặc cú pháp

[**USE** Tên_CSDL]

ALTER TABLE Tên_Bảng ADD CONSTRAINT Tên_ràng_buộc
PRIMARY KEY (Tên_cột_khóa_chính1, Tên_cột_khóa_chính1, ...);

Ràng buộc

Khóa chính – Primary Key (tiếp)

- Ví dụ: Giả sử bảng kết quả đã được tạo, thêm khóa chính bằng lệnh ALTER TABLE

```
ALTER TABLE KETQUA ADD PRIMARY KEY(MaSV, MaMon);
```

```
ALTER TABLE KETQUA ADD CONSTRAINT KhoaChinh PRIMARY KEY(MaSV, MaMon);
```

Ràng buộc Khóa duy nhất – UNIQUE

- Ràng buộc khóa duy nhất (UNIQUE) được sử dụng để đảm bảo rằng chỉ các giá trị duy nhất được nhập vào trong cột hoặc một tập hợp các cột
- Cách thêm ràng buộc khóa duy nhất khi tạo bảng

```
CREATE TABLE Tên_Bảng(  
    Tên_cột1 Kiểu_dữ_liệu [Tùy_chọn],  
    Tên_cột2 Kiểu_dữ_liệu [Tùy_chọn] UNIQUE,  
    ...  
);
```

Ràng buộc Khóa duy nhất – UNIQUE (tiếp)

- Cách thêm ràng buộc khóa duy nhất khi tạo bảng

```
CREATE TABLE Tên_Bảng(
```

```
...
```

```
UNIQUE (tên_cột_unique_1, tên_cột_unique_2,...)
```

```
);
```

```
CREATE TABLE Tên_Bảng(
```

```
...
```

```
CONSTRAINT Tên_ràng_buộc UNIQUE (tên_cột_unique_1,  
tên_cột_unique_2,...)
```

```
);
```

Ràng buộc Khóa duy nhất – UNIQUE (tiếp)

- Cách thêm ràng buộc khóa duy nhất khi sửa bảng

ALTER TABLE Tên_Bảng ADD UNIQUE (tên_cột_unique_1, tên_cột_unique_2,...);

ALTER TABLE Tên_Bảng ADD CONSTRAINT Tên_ràng_buộc
UNIQUE (tên_cột_unique_1, tên_cột_unique_2,...);

Ràng buộc Khóa duy nhất – UNIQUE (tiếp)

- Ví dụ: thêm ràng buộc duy nhất cho 2 cột MaSV và MaMon của bảng KETQUA

```
ALTER TABLE KETQUA  
ADD UNIQUE (MaSV, MaMon);
```


Ràng buộc

Khóa ngoại – FOREIGN KEY

- Một khóa ngoại trong một bảng là một cột, nó chỉ đến một khóa chính trong một bảng khác
- Cách thêm ràng buộc khóa ngoại khi tạo bảng

```
CREATE TABLE Tên_Bảng( ...
```

```
FOREIGN KEY (Tên_cột_khóa_ngoại) REFERENCES  
Tên_bảng_tham_chiếu(tên_khóa_chính_bảng_tham_chiếu)  
);
```

```
CREATE TABLE Tên_Bảng( ...
```

```
CONSTRAINT Tên_ràng_buộc FOREIGN KEY (Tên_cột_khóa_ngoại)  
REFERENCES Tên_bảng_tham_chiếu(tên_khóa_chính_bảng_tham_chiếu)  
);
```

Ràng buộc

Khóa ngoại – FOREIGN KEY (tiếp)

- Cách thêm ràng buộc khóa ngoại khi sửa bảng

```
ALTER TABLE Tên_Bảng ADD FOREIGN KEY (Tên_cột_khóa_ngoại)  
REFERENCES Tên_bảng_tham_chiếu(tên_khóa_chính_bảng_tham_chiếu);
```

```
CREATE TABLE Tên_Bảng  
  
ADD CONSTRAINT Tên_ràng_buộc  
  
FOREIGN KEY (Tên_cột_khóa_ngoại)  
  
REFERENCES Tên_bảng_tham_chiếu(tên_khóa_chính_bảng_tham_chiếu);
```

Ràng buộc

Khóa chính – Primary Key (tiếp)

- Ví dụ: Giả sử bảng KETQUA đã được tạo, thêm khóa ngoại bằng lệnh ALTER TABLE sao cho tham chiếu trường MaSV đến trường khóa chính MaSV trong bảng SinhVien

```
ALTER TABLE KETQUA
```

```
ADD FOREIGN KEY (MaSV) REFERENCES SinhVien(MaSV);
```

Ràng buộc Kiểm tra – CHECK

- Ràng buộc kiểm tra CHECK kiểm tra giới hạn giá trị có thể được nhập vào trong cột.
- Cách thêm ràng buộc CHECK

```
CREATE TABLE Tên_Bảng(
```

```
...
```

```
Tên_cột_kiểm_tra Kiểu_dữ_liệu CHECK (Điều_kiện));
```

```
ALTER TABLE Tên_Bảng ADD CONSTRAINT Tên_ràng_buộc CHECK (Điều_kiện));
```

Ràng buộc Kiểm tra – CHECK (tiếp)

- Ví dụ: thêm ràng buộc kiểm tra cho bảng SinhVien sao cho GioiTinh chỉ nhận giá trị 'Nam' hoặc 'Nữ'

```
ALTER TABLE SinhVien  
ADD CONSTRAINT KTGioiTinh CHECK (GioiTinh=N'Nam' or GioiTinh=N'Nữ');
```

Xóa ràng buộc

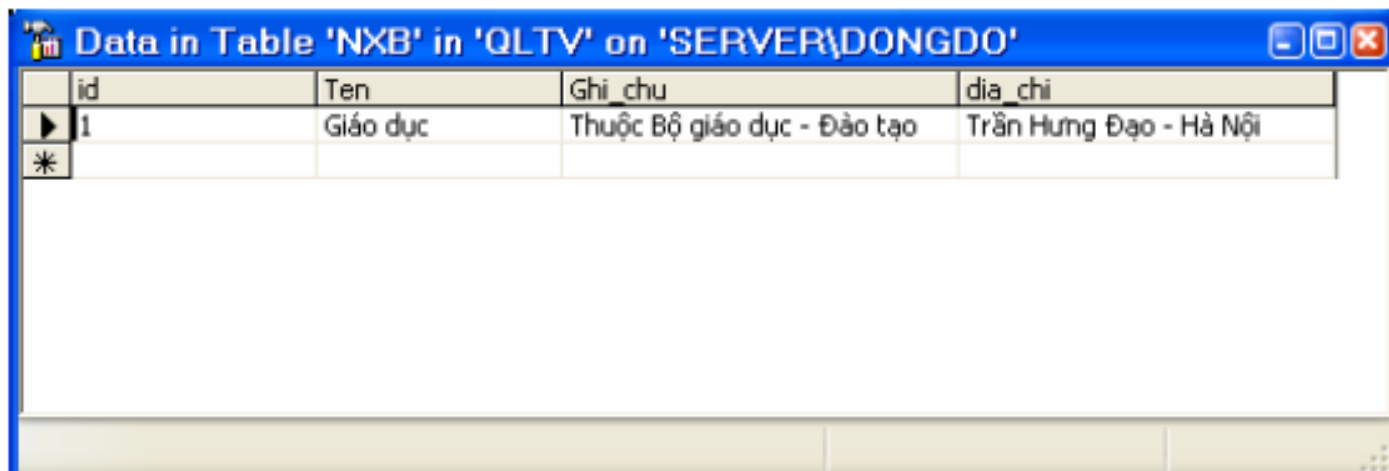
- Để xóa ràng buộc trong một bảng, ta dùng cú pháp sau
- Cách thêm ràng buộc CHECK

ALTER TABLE Tên_Bảng DROP CONSTRAINT Tên_ràng_buộc;

Làm việc với bản ghi trong SQL Server

Sử dụng SQL Server Management Studio

- Chọn bảng dữ liệu
- Nhấn chuột phải, chọn Open Table – Return all rows
- Nhập dữ liệu theo quy cách kiểu dữ liệu, ràng buộc nếu có
- Sửa, xóa dữ liệu nếu muốn



The screenshot shows a window titled "Data in Table 'NXB' in 'QLTV' on 'SERVER\DONGDO'". The window displays a table with four columns: id, Ten, Ghi_chu, and dia_chi. The first row contains the values 1, Giáo dục, Thuộc Bộ giáo dục - Đào tạo, and Trần Hưng Đạo - Hà Nội. Below the table is a large empty area for data entry, and a status bar at the bottom.

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

T-SQL: Nhập dữ liệu vào bảng

■ Cú pháp

```
INSERT INTO Tên_Bảng(Thuộc_tính_1, Thuộc_tính_2,...)  
VALUES(giá_trị_thuộc_tính_1, giá_trị_thuộc_tính_2,...);
```

- **Ví dụ:** thêm vào bảng SinhVien một bản ghi sinh viên với thông tin họ tên là 'Nguyễn Văn A', sinh ngày 14/9/1996, địa chỉ ở 175 Tây Sơn (Chú ý: chữ N dùng để nhập tiếng Việt)

```
insert into SinhVien(HoTen,NgaySinh,DiaChi)  
values(N'Nguyễn Văn A','19960914',N'175 Tây Sơn');
```

```
insert into SinhVien(HoTen,NgaySinh,DiaChi)  
values(N'Nguyễn Văn B','09/14/1998',N'175 Tây Sơn');
```

T-SQL: Cập nhật dữ liệu của bảng

■ Cú pháp

UPDATE Tên_Bảng

SET Tên_cột = biểu thức

[,..., Tên_cột_k = biểu_thức_k]

[**FROM** danh_sách_bảng]

[**WHERE** điều_kiện]

- **Ví dụ:** sửa ngày sinh của sinh viên 'Nguyễn Văn A' thành 15/9/1995

```
update SinhVien
```

```
set NgaySinh='09/15/1995'
```

```
where HoTen=N'Nguyễn Văn A';
```

T-SQL: Xóa dữ liệu của bảng

■ Cú pháp

```
DELETE FROM Tên_Bảng  
[FROM danh_sách_bảng]  
[WHERE điều_kiện]
```

■ Ví dụ: Xóa khỏi bảng KetQua tất cả điểm thi của sinh viên 'Nguyễn Văn A'

```
delete from KetQua  
from SinhVien  
where SinhVien.MaSV = KetQua.MaSV  
and SinhVien.HoTen = N'Nguyễn Văn A';
```