

CHƯƠNG 2: TẠO VÀ QUẢN TRỊ CƠ SỞ DỮ LIỆU

Giảng viên: Dương Quang Huy

NỘI DUNG

1. Cơ sở dữ liệu
2. Bảng (Table)
3. Tạo các ràng buộc
4. Chỉ mục

1. Cơ sở dữ liệu

❖ Dữ liệu:

- Hiểu một cách đơn giản, dữ liệu (data) là tập hợp thông tin (số liệu, miêu tả, hình ảnh, video,...) được thu thập bằng quan sát, đo lường, nghiên cứu hoặc phân tích.
- Dữ liệu được tổ chức dưới dạng đồ thị, biểu đồ hoặc bảng.



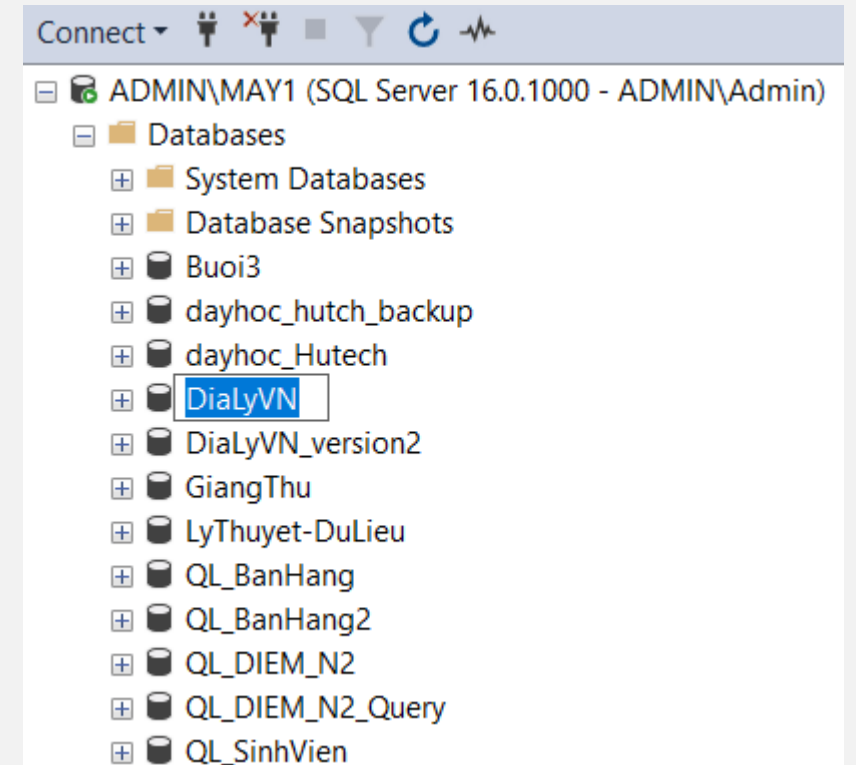
Dữ liệu (Data)

- Chắc hẳn bạn đã từng xem dự báo thời tiết trên các kênh tin tức. Họ liệt kê nhiệt độ tối thiểu, nhiệt độ tối đa, dự đoán và đo lường mưa. Biểu diễn dạng bảng được chỉ ra dưới đây.

Thành phố	Nhiệt độ cao nhất	Nhiệt độ thấp nhất	Khả năng có mưa
Hà Nội	30	25	20%
Hồ Chí Minh	37	29	50%
Đà Nẵng	35	28	40%

Cơ sở dữ liệu

- **Một cơ sở dữ liệu (Database)** là một tập hợp **các dữ liệu có liên quan** với nhau, chứa thông tin của một tổ chức nào đó (như một trường học, một ngân hàng, một công ty, một nhà máy,...), được lưu trữ trên các thiết bị nhớ để đáp ứng nhu cầu khai thác thông tin của nhiều người dùng với nhiều mục đích khác nhau.



Các cơ sở dữ liệu được SQL hỗ trợ

❖ SQL Server hỗ trợ hai loại cơ sở dữ liệu:

□ Cơ sở dữ liệu hệ thống

- Được sử dụng để lưu thông tin về SQL Server.
- Được sử dụng để quản lý các cơ sở dữ liệu người dùng

□ Cơ sở dữ liệu người dùng định nghĩa

- Người sử dụng SQL Server có thể tạo các cơ sở dữ liệu người dùng định nghĩa
- Mục đích của các cơ sở dữ liệu này là để lưu dữ liệu của người sử dụng

Các cơ sở dữ liệu của hệ thống

Cơ sở dữ liệu	Mô tả
master	Cơ sở dữ liệu này lưu trữ các tất cả các thông tin về hệ thống của SQL Server.
msdb	Cơ sở dữ liệu này được sử dụng bởi dịch vụ SQL Server Agent để nhắc nhở các thực hiện công việc theo lịch biểu và các công việc khác.
model	Cơ sở dữ liệu này được sử dụng như là cơ sở dữ liệu mẫu cho tất cả các cơ sở dữ liệu được người dùng tạo ra sau này trong SQL Server.
tempdb	Các kết quả tạm thời, đối tượng tạm thời được tạo ra trong quá trình xử lý tính toán được lưu trong cơ sở dữ liệu này

CSDL người dùng định nghĩa

❖ Để tạo một cơ sở dữ liệu người dùng, cần các thông tin sau:

- Tên của cơ sở dữ liệu
- Người sở hữu hoặc người tạo cơ sở dữ liệu
- Kích thước của cơ sở dữ liệu
- Các tập tin và nhóm các tập tin được sử dụng để lưu trữ cơ sở dữ liệu

CSDL người dùng định nghĩa

❖ Cấu trúc lưu trữ của một CSDL

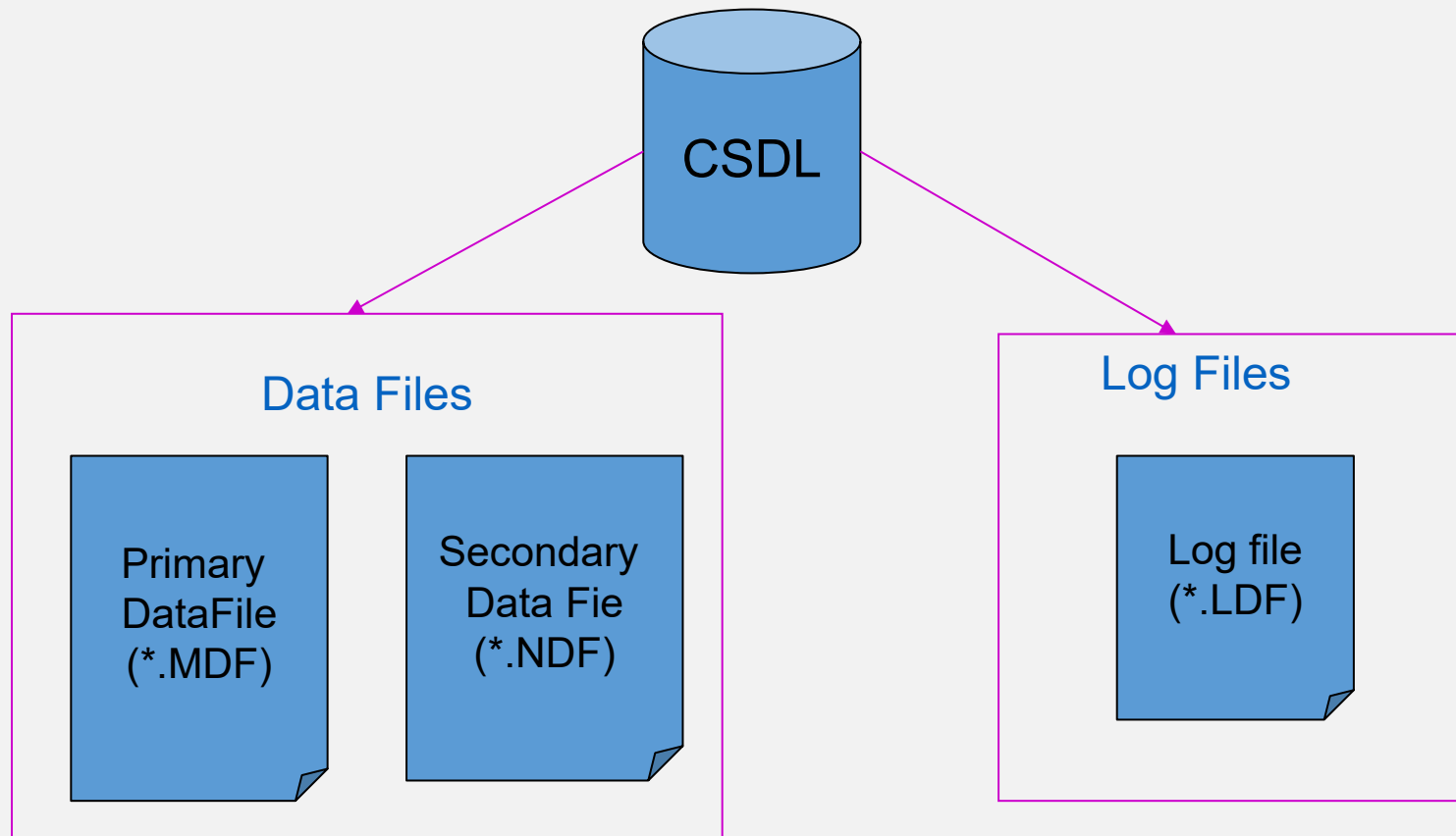
❑ Một CSDL trong SQL Server tối thiểu sẽ dùng **hai tập tin vật**

lý để lưu trữ dữ liệu:

- Một dùng lưu trữ dữ liệu (Data file)
- Một dùng để lưu trữ các vết giao tác (Transaction log file)

CSDL người dùng định nghĩa

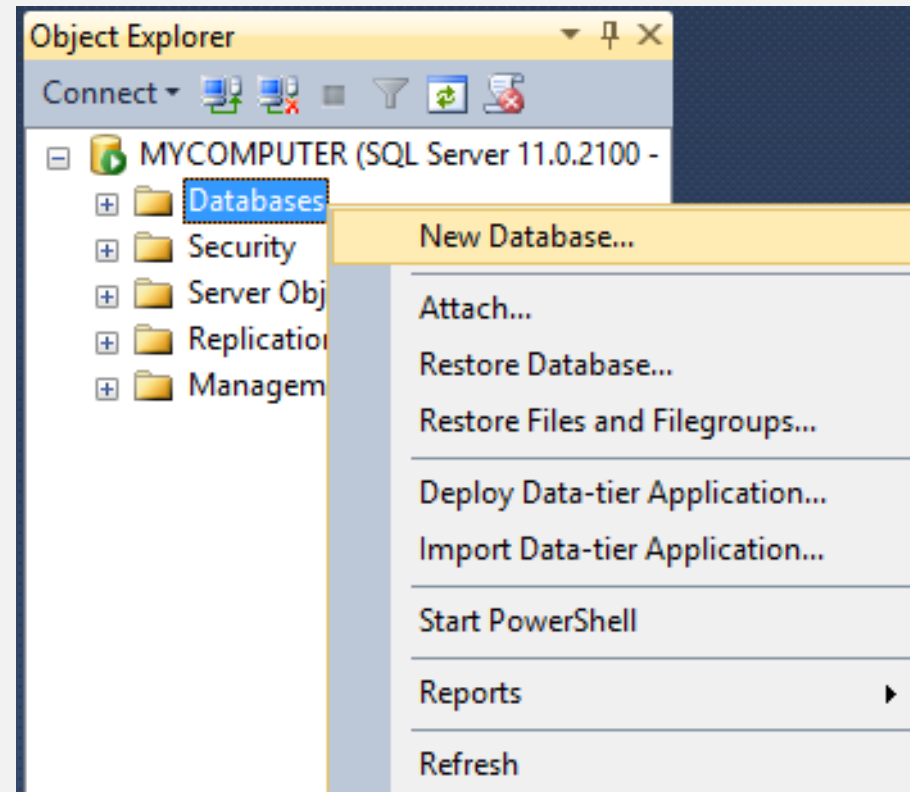
Thường được tổ chức lưu trữ trong 3 tập tin:



Tạo mới CSDL trong SQL Server

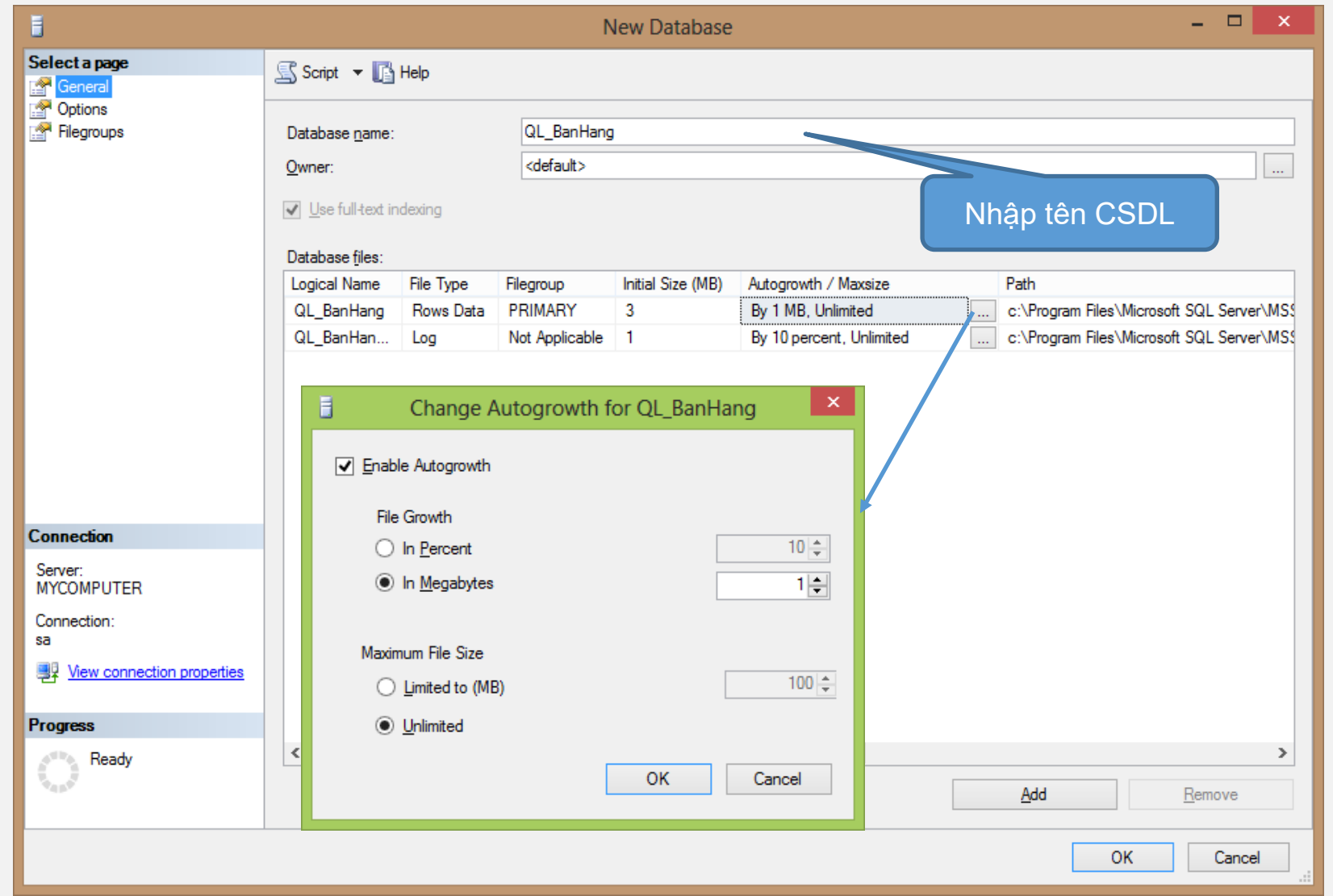
❖ Cách 1:

- ❑ Click chuột phải vào Databases chọn "New Database"



Tạo mới CSDL trong SQL Server

❖ Xuất hiện hộp thoại:



Tạo mới CSDL trong SQL Server

❖ Cách 2: Tạo CSDL bằng cách viết lệnh

❑ Cú pháp:

```
create database Ten_CSDL
```

❑ Ví dụ:

```
create database QuanLyDiemSinhVien
```

Tạo mới CSDL trong SQL Server

```
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 mới CSDL trong SQL Server

□ Ví dụ:

```
CREATE DATABASE QLBanHang
On Primary
(
    Name=QLBanHang_data,          -- Tên file logic: QLBanHang_data
    FileName='D:\QLBanHang.mdf', -- Tên file vật lý: QLBanHang.mdf
    Size=50MB,                    -- Kích thước tập tin ban đầu:50MB
    MaxSize= Unlimited,          -- Kích thước tối đa: Không giới hạn
    FileGrowth=10MB              -- Kích thước tăng trưởng: 10 MB
)
Log On
(
    Name=QLSV_log,
    FileName='D:\QLBanHang.ldf',
    Size=50MB,
    MaxSize=100MB,
    FileGrowth=10MB
);
```

Cơ sở dữ liệu

❖ **Lệnh xóa cơ sở dữ liệu**

❑ Cú pháp:

```
drop database Ten_CSDL
```

❑ Ví dụ:

```
drop database QuanLyDiemSinhVien
```

Cơ sở dữ liệu

❖ **Lệnh gọi cơ sở dữ liệu để làm việc**

❑ Cú pháp:

`use Ten_CSDL`

❑ Ví dụ:

`use QuanLyDiemSinhVien`

Thay đổi kích thước các tập tin của CSDL

- ❑ Tăng kích thước tập tin dữ liệu lên 60MB

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

- ❑ Giảm kích thước tập tin dữ liệu xuống 20MB

```
USE QL_BanHang  
DBCC SHRINKFILE (NAME = QL_BanHang , 20)
```

- ❑ Chỉnh sửa kích thước tăng trưởng 5% mỗi khi database bị đầy

```
ALTER DATABASE QL_BanHang  
MODIFY FILE (NAME = QL_BanHang, FILEGROWTH = 5%)
```

2. Bảng (Table)

❖ Bảng sinh viên trong một Cơ sở dữ liệu

SinhVien

MaSV	HoTen	Lop	Khoa
51703097	Nguyễn Văn Nam	Cd24Ct2	CNTT
51703079	Ngô Văn Tùng	Cd24Ct2	CNTT
51704520	Thúy Vi	Cd24Ct5	KT
51704230	Đình Thành	Cd23Ct7	ĐT

Bảng (Table)

❖ Tạo bảng bằng lệnh

□ Cú pháp:

```
CREATE TABLE <tên bảng>
(
    <tên_cột_1> <kiểu dữ liệu> [NOT NULL],
    <tên_cột_2> <kiểu dữ liệu> [NOT NULL],
    ..... ,
    <tên cột n> <kiểu dữ liệu> [NOT NULL]
)
```

Bảng (Table)

❖ Tạo bảng bằng lệnh

□ Ví dụ:

```
CREATE TABLE LOP
```

```
(
```

```
    MaLop      char(6)          NOT NULL,
```

```
    TenLop     nvarchar(30)     NOT NULL,
```

```
    SiSo       int              NOT NULL
```

```
)
```

Bảng (Table)

□ Giá trị mặc định:

➤ Cú pháp:

```
CREATE TABLE <tên bảng>
(
    <tên cột 1> <kiểu dữ liệu> DEFAULT Giá_trị
    <tên cột 2> <kiểu dữ liệu> [NOT NULL] ,
    ..... ,
    <tên cột n> <kiểu dữ liệu n> [NOT NULL]
)
```

- **Giá trị:** là giá trị cụ thể sẽ được gán vào cột

Bảng (Table)

Ví dụ 1:

```
CREATE TABLE LOP
```

```
(
```

```
    MaLop      char(6)          NOT NULL,
```

```
    TenLop     nvarchar(30)     NOT NULL,
```

```
    SiSo      int      Default 0
```

```
)
```

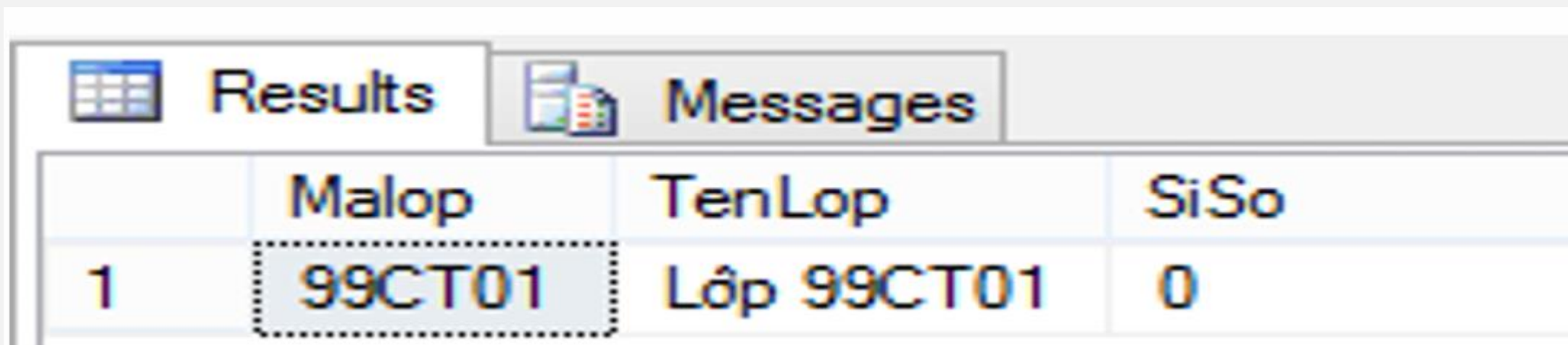
Bảng (Table)

-- Cập nhật dữ liệu

```
Insert into LOP(MaLop, TenLop) Values ('99CT01',N'Lớp 99CT01')
```

-- Truy vấn dữ liệu

```
Select * from LOP
```



The screenshot shows the SQL Server Enterprise Manager interface. The 'Results' pane is active, displaying a table with the following data:

	Malop	TenLop	SiSo
1	99CT01	Lớp 99CT01	0

Bảng (Table)

Ví dụ 2:

```
CREATE TABLE DangKy
```

```
(
```

```
    MaSV        char(10)          NOT NULL,
```

```
    MaMH        char(10)          NOT NULL,
```

```
    NgayDK      datetime         Default '2013-02-25'
```

```
)
```

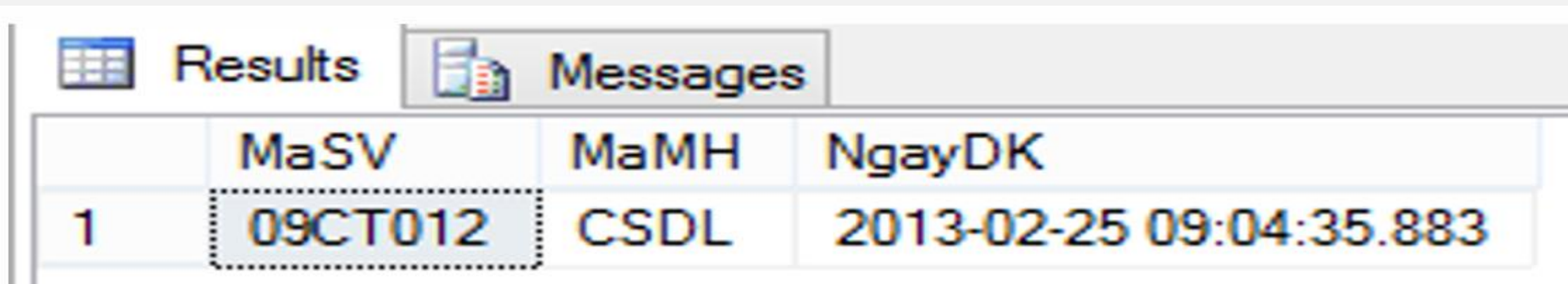
Bảng (Table)

-- Cập nhật dữ liệu

```
Insert into DangKy(MaSV, MaMH) values('09CT012','CSDL')
```

-- Truy vấn dữ liệu

```
Select * from DangKy
```



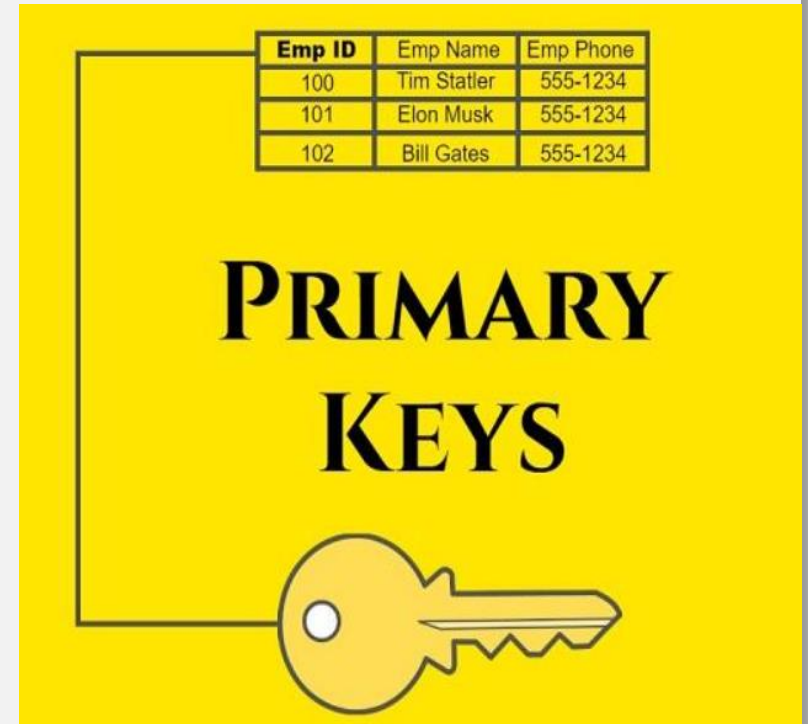
The screenshot shows a database interface with two tabs: "Results" and "Messages". The "Results" tab is active, displaying a table with the following data:

	MaSV	MaMH	NgayDK
1	09CT012	CSDL	2013-02-25 09:04:35.883

Khóa chính (Primary Key)

➤ Khóa chính (Primary Key):

- ❑ Tập hợp của một hay nhiều cột trong bảng
- ❑ Dữ liệu tại cột này phải khác rỗng (Not null)
- ❑ Dữ liệu tại cột này phải là duy nhất trong bảng



Khóa chính (Primary Key)

Ví dụ 1:

Bảng sinh viên có cột MaSV là khóa chính

MaSV	HoTen	Lop	Khoa
51703097	Nguyễn Văn Nam	Cd24Ct2	CNTT
51703079	Ngô Văn Tùng	Cd24Ct2	CNTT
51704520	Thúy Vi	Cd24Ct5	KT
51704230	Đình Thành	Cd23Ct7	ĐT

Khóa chính (Primary Key)

Ví dụ 2:

Bảng Kết Quả có 2 cột MaSV và MaMH là khóa chính

MaSV	MaMH	Diem
51703097	CSDL	7
51703079	CSDL	8
51704520	TKW	3
51704230	AV	6

Khóa chính (Primary Key)

❖ Tạo khóa chính bằng lệnh

❑ Cú pháp tạo **khóa chính** trong khi tạo bảng:

```
create table Tên_Bảng  
(  
    Tên_Cột_1 Kiểu_dữ_liệu Primary Key,  
    Tên_Cột_2 Kiểu_dữ_liệu [NOT NULL],  
    ...  
)
```

Khóa chính (Primary Key)

□ Ví dụ:

```
create table LOP
```

```
(
```

```
    MaLop      char(6)          PRIMARY KEY,
```

```
    TenLop     nvarchar(30)     NOT NULL,
```

```
    SiSo       int           Default 0
```

```
)
```

Khóa chính (Primary Key)

❖ Tạo khóa chính bằng lệnh

❑ Cú pháp tạo khóa chính sau khi tạo bảng:

```
ALTER TABLE Tên_bảng ADD  
    CONSTRAINT PK_Tên_bảng_TenCot PRIMARY  
    KEY (Danh_sách_cột_khóa_chính)
```

Khóa chính (Primary Key)

□ Ví dụ:

```
CREATE TABLE LOP
(
    MaLop    char(6)          NOT NULL,
    TenLop   nvarchar(30)     NOT NULL,
    SiSo     smallInt Default 0,
)
Alter table LOP Add
    CONSTRAINT PK_LOP_MaLop PRIMARY KEY (MaLop)
```

Khóa ngoại (Foreign Key)

❖ Khóa ngoại

- ❑ Tập hợp của một hay nhiều cột trong bảng (trong 1 bảng có thể có 1 hoặc nhiều khóa ngoại), mà các cột này là khóa chính của các bảng khác tham chiếu đến
- ❑ Dữ liệu tại các cột này phải lấy từ bảng mà có cột làm khóa chính tham chiếu đến.

#	Name	Type	Collation	Attributes	Null	Default
<input type="checkbox"/>	1 id 	int(11)			No	None
<input type="checkbox"/>	2 order_id 	int(11)			No	None
<input type="checkbox"/>	3 product_id	varchar(155)	utf8_unicode_ci		No	None
<input type="checkbox"/>	4 name	varchar(1250)	utf8_unicode_ci		Yes	NULL
<input type="checkbox"/>	5 num	int(11)			No	None

Khóa ngoại (Foreign Key)

Ví dụ 1:

Bảng sinh viên có cột MaKH là khóa ngoại tham chiếu đến bảng Khoa

Bảng Khoa

MaKH	TenKhoa
CNTT	Công nghệ thông tin
KT	Kinh tế
VL	Vật Lý
TH	Tin Học

Bảng Sinh Viên

MaSV	HoTen	Lop	MaKH
51703097	Nguyễn Văn Nam	Cd24Ct2	CNTT
51703079	Ngô Văn Tùng	Cd24Ct2	CNTT
51704520	Thúy Vi	Cd24Ct5	KT
51704230	Đình Thành	Cd23Ct7	VL

Khóa ngoại (Foreign Key)

❖ Tạo khóa ngoại bằng lệnh

❑ Cú pháp tạo **khóa ngoại** trong khi tạo bảng:

```
CREATE TABLE Tên_bảng
```

```
(
```

```
Tên_cột_1    Kiểu_dữ_liệu    [NOT NULL],
```

```
Tên_cột_2    Kiểu_dữ_liệu    [NOT NULL]
```

```
[,...]
```

```
CONSTRAINT  FK_Tên_bảng_TenCot  FOREIGN KEY  (Cột_Khóa_Ngoại)
```

```
REFERENCES Tên_bảng_tham_chiếu(cột_tham_chiếu)
```

```
)
```

Khóa ngoại (Foreign Key)

□ Ví dụ:

```
CREATE TABLE HOADON(  
    MSHD          INT          NOT NULL,  
    MSNV          CHAR(6)     NOT NULL,  
    MSKH          INT          NOT NULL,  
    NGAYHD       DATETIME    NOT NULL,  
    TONGTIEN     FLOAT        DEFAULT 0,  
    CONSTRAINT PK_HOADON_MSHD PRIMARY KEY(MSHD)  
    CONSTRAINT FK_HOADON_MSNV FOREIGN KEY(MSNV) REFERENCES  
    NHANVIEN(MANV)
```

)

Khóa ngoại (Foreign Key)

❖ Tạo khóa ngoại bằng lệnh

❑ Cú pháp tạo khóa ngoại sau khi tạo bảng:

ALTER TABLE Tên_bảng **ADD**

CONSTRAINT FK_Tên_bảng_TenCot

FOREIGN KEY (Cột_Khóa_Ngoại)

REFERENCES Tên_bảng_tham_chiếu(cột_tham_chiếu)

Khóa ngoại (Foreign Key)

□ Ví dụ:

```
CREATE TABLE HOADON(  
    MSHD          INT          NOT NULL,  
    MSNV          CHAR(6)      NOT NULL,  
    MSKH          INT          NOT NULL,  
    NGAYHD       DATETIME     NOT NULL,  
    TONGTIEN     FLOAT        DEFAULT 0,  
    CONSTRAINT PK_HOADON_MSHD PRIMARY KEY(MSHD)  
    CONSTRAINT FK_HOADON_MSNV FOREIGN KEY(MSNV) REFERENCES  
    NHANVIEN(MANV)  
)
```

Khóa ngoại (Foreign Key)

ALTER TABLE HOADON ADD

CONSTRAINT FK_HOADON_MSKH FOREIGN KEY(MSKH) REFERENCES

KHACHHANG(MSKH)

Thay đổi cấu trúc bảng

- ❖ Thêm một cột mới vào bảng

`ALTER TABLE Tên_bảng ADD Tên_cột Kiểu dữ liệu`

- Ví dụ:

`ALTER TABLE Khách_Hang ADD DienThoai nvarchar(10)`

- ❖ Sửa đổi kiểu dữ liệu cột trong bảng

`ALTER TABLE Tên_bảng ALTER COLUMN Tên_cột Kiểu dữ liệu mới`

- Ví dụ:

`ALTER TABLE Khách_Hang ALTER COLUMN DienThoai Char(12)`

Thay đổi cấu trúc bảng

- ❖ Xóa một cột trong bảng

```
ALTER TABLE Tên_bảng DROP COLUMN Tên_cột
```

- Ví dụ:

```
ALTER TABLE Khách_Hang DROP COLUMN DienThoai
```

- ❖ Sửa tên bảng trong CSDL

```
EXEC sp_Rename 'Tên bảng cũ', 'Tên bảng mới'
```

- Ví dụ:

```
EXEC sp_Rename 'KHACH_HANG', 'KHACHHANG')
```

Thay đổi cấu trúc bảng

❖ Xóa bảng trong CSDL

DROP TABLE Tên_bảng

□ Ví dụ: Xóa bảng KhachHang

DROP TABLE KhachHang

Cập nhật dữ liệu

❖ Thêm (Insert)

❖ Sửa (Update)

❖ Xóa (Delete)

CÂU LỆNH INSERT INTO

❖ Dùng để thêm dữ liệu vào bảng

❖ Cú pháp:

Insert into <Tên bảng> (Tên cột 1, Tên cột 2, [...])

Values (Giá trị 1, Giá trị 2, ...)

Chú ý:

-> Với giá trị kiểu chuỗi và kiểu ngày nằm trong dấu 'nháy đơn'

-> Thứ tự các giá trị phải trùng với thứ tự các cột

CÂU LỆNH INSERT INTO

➤ Ví dụ 1:

Insert into KhachHang (mskh,tenkh,phai,diachi)

Values (1, N'Long', N'Nam', N'305 Đại lộ 3')

➤ Ví dụ 2:

Insert into KhachHang (mskh,tenkh,phai,diachi,dienthoai)

Values (2, N'Thư', N'Nữ', N'Lý Thường Kiệt', Null)

CÂU LỆNH INSERT INTO

➤ Ví dụ 3:

Insert into KhachHang

Values (3, N'Thành', N'Nam', N'Cà Mau', '0945454545')

➤ Ví dụ 4:

Insert into KhachHang

Values (3, N'Thành', N'Nam', N'Cà Mau', '0945454545'),

(4, N'Thanh', N'Nữ', N'Cà Mau', '0945454545'),

(5, N'An', N'Nam', N'HCM', '0945454545')

CÂU LỆNH UPDATE

- ❖ Dùng để thay đổi dữ liệu của bảng
- ❖ Cú pháp:

Update Tên_Bảng

Set Tên_cột_1 = Giá trị,

Tên_cột_2 = Giá trị,

...

[Where Điều kiện]

CÂU LỆNH UPDATE

❖ Ví dụ 1: Thay đổi địa chỉ và số điện thoại của khách hàng có mã số = 9

Update KhachHang

Set DiaChi = N'215 Điện Biên Phủ',

DienThoai = '0945464665',

Where MSKH = 9

CÂU LỆNH UPDATE

❖ Ví dụ 2: Giảm giá mặt hàng 'Mouse HP' xuống 90k so với ban đầu.

Update MatHang

Set DonGia = DonGia - 90000

Where TenMH like N'Mouse HP'

CÂU LỆNH DELETE

❖ Dùng để xóa dữ liệu của bảng

❖ Cú pháp

Delete From Tên_Bảng

[Where <Điều kiện>]

❖ Ví dụ: Xóa khách hàng có mã số 9

Delete From KhachHang

Where MSKH = 9

3. Tính toàn vẹn trong cơ sở dữ liệu

❖ Các ràng buộc trong SQL Server được quản lý bởi hai đối tượng: **CONSTRAINT** và **TRIGGER**

❑ **Constraint:** Quản lý các ràng buộc như:

- Ràng buộc dữ liệu duy nhất
- Ràng buộc dữ liệu tồn tại
- Ràng buộc miền giá trị

❑ **Trigger:** Quản lý các ràng buộc phức tạp, liên quan đến dữ liệu trên nhiều bảng

Ràng buộc dữ liệu duy nhất

❖ Kiểm tra duy nhất dữ liệu: Có hai loại

Primary Key (khoá chính):

Dữ liệu khác rỗng và duy nhất.

Unique:

Dữ liệu có thể rỗng và dữ liệu tồn tại duy nhất

❖ Định nghĩa ràng buộc khoá chính

Tạo **khóa chính** trong khi tạo bảng

```
CREATE TABLE Tên_bảng  
(  
    Tên_cột_1 Kiểu_dữ_liệu [NOT NULL],  
    Tên_cột_2 Kiểu_dữ_liệu [NOT NULL] [...]  
    [CONSTRAINT PK_Tên_bảng_TenCot] PRIMARY  
    KEY (D.sách_cột_khóa_chính)  
)
```

Tạo **khóa chính** sau khi tạo bảng

```
ALTER TABLE Tên_bảng ADD  
    CONSTRAINT PK_Tên_bảng_TenCot PRIMARY  
    KEY (Danh_sách_cột_khóa_chính)
```

❖ Định nghĩa ràng buộc duy nhất

- ❑ Cách tạo ra **Unique Constraint** cũng tương tự như **Primary Key Constraint** chỉ việc thay chữ **Primary Key** thành **Unique**.

```
CREATE TABLE Tên_bảng  
(  
    Tên_cột_1 Kiểu_dữ_liệu [NOT NULL],  
    Tên_cột_2 Kiểu_dữ_liệu [NOT NULL] [...]  
    [CONSTRAINT UQ_Tên_bảng_TenCot ]  
    UNIQUE (D.sách_cột_tạo_duy_nhất)  
)
```

❖ Ví dụ 1:

```
CREATE TABLE LOP
(
    MaLop    char(6)          NOT NULL,
    TenLop   nvarchar(30)    NOT NULL ,
    SiSo     smallInt Default 0,
    CONSTRAINT PK_LOP_MaLop PRIMARY KEY (MaLop),
    CONSTRAINT UQ_LOP_TenLop UNIQUE (TenLop)
)
```

❖ Ví dụ 2:

```
CREATE TABLE LOP
(
    MaLop    char(6)          NOT NULL,
    TenLop   nvarchar(30)    NOT NULL,
    SiSo     smallInt Default 0,
)
Alter table LOP Add
    CONSTRAINT PK_LOP_MaLop PRIMARY KEY (MaLop),
    CONSTRAINT UQ_LOP_TenLop UNIQUE (TenLop)
```

Kiểm tra tồn tại Dữ liệu

- ❑ Định nghĩa **khóa ngoại** trong khi tạo bảng

```
CREATE TABLE Tên_bảng
```

```
(
```

```
    Tên_cột_1 Kiểu_dữ_liệu [NOT NULL],
```

```
    Tên_cột_2 Kiểu_dữ_liệu [NOT NULL] [...]
```

```
    [CONSTRAINT FK_Tên_bảng_TenCot] FOREIGN
```

```
    KEY (Cột_Khóa_Ngoại) REFERENCES Tên_bảng_tham_chiếu(cột_tham_chiếu)
```

```
)
```

- ❑ Định nghĩa **khóa ngoại** sau khi tạo bảng

```
ALTER TABLE Tên_bảng ADD
```

```
    CONSTRAINT FK_Tên_bảng_Tên_cột
```

```
    FOREIGN KEY (Cột_khóa_ngoại)
```

```
    REFERENCES Tên_bảng_tham_chiếu(Cột_tham_chiếu)
```

❖ Ví dụ:

```
CREATE TABLE HOADON
(
    MSHD          INT          NOT NULL,
    MSNV          CHAR(6)      NOT NULL,
    MSKH          INT          NOT NULL,
    NGAYHD        DATETIME    NOT NULL,
    TONGTIEN     MONEY        DEFAULT 0,
    CONSTRAINT PK_HOADON PRIMARY KEY(MSHD),
    CONSTRAINT FK_HOADON_MSNO FOREIGN
        KEY(MSNV) REFERENCES NHANVIEN(MSNV)
)
ALTER TABLE HOADON ADD
    CONSTRAINT FK_HOADON_MSKH FOREIGN KEY(MSKH)
    REFERENCES KHACHHANG(MSKH)
```

Kiểm tra ràng buộc miền giá trị(CHECK)

❖ Định nghĩa **CHECK** trong khi tạo bảng

```
CREATE TABLE Tên_bảng(  
    Tên_cột_1    Kiểu_dữ_liệu    [NOT NULL],  
    Tên_cột_2    Kiểu_dữ_liệu    [NOT NULL],  
    ...,  
    CONSTRAINT CK_Tên_Bảng_Tên_cột  
    CHECK (Biểu_thức_luận_lý)  
)
```

Kiểm tra ràng buộc miền giá trị(CHECK)

❖ Ví dụ: Kiểm tra DONGIA > 1000

```
CREATE TABLE CHITIET_HD
```

```
(
```

```
    MSHD      INT          NOT NULL,
```

```
    MSMH      CHAR(6)      NOT NULL,
```

```
    SOLUONG   INT,
```

```
    DONGIA    FLOAT,
```

```
    CONSTRAINT CK_CHITIET_HD_DONGIA
```

```
    CHECK (DONGIA > 1000)
```

```
)
```

Kiểm tra ràng buộc miền giá trị(CHECK)

❖ Thực thi:

```
Insert into CHITIET_HD
```

```
Values ('HD01','SP01',2,1500)
```

⇒ Thành công

```
Insert into CHITIET_HD
```

```
Values ('HD02','SP02',2,900)
```

⇒ Thất bại

Kiểm tra ràng buộc miền giá trị(CHECK)

❖ Định nghĩa CHECK sau khi tạo bảng

```
ALTER TABLE Tên_bảng ADD
```

```
CONSTRAINT CK_Tên_Bảng_Tên_cột
```

```
CHECK (Biểu_thức_luận_lý)
```

Kiểm tra ràng buộc miền giá trị(CHECK)

❖ Ví dụ: Kiểm tra SOLUONG > 0

```
CREATE TABLE CHITIET_HD
(
    MSHD      INT           NOT NULL,
    MSMH      CHAR(6)       NOT NULL,
    SOLUONG   INT,
    DONGIA    FLOAT,
    CONSTRAINT CK_CHITIET_HD_DONGIA
CHECK (DONGIA > 1000)
)
```

Kiểm tra ràng buộc miền giá trị(CHECK)

❖ Ví dụ: Kiểm tra SOLUONG > 0

```
ALTER TABLE CHITIET_HD ADD
```

```
CONSTRAINT CK_CHITIET_HD_SOLUONG
```

```
CHECK (SOLUONG > 0)
```

Kiểm tra ràng buộc miền giá trị(CHECK)

❖ Thực thi:

```
Insert into CHITIET_HD
```

```
Values ('HD01','SP01',2,100)
```

⇒ Thành công

```
Insert into CHITIET_HD
```

```
Values ('HD02','SP02',-3,900)
```

⇒ Thất bại

Kiểm tra ràng buộc miền giá trị (RULE)

❑ Tạo RULE.

```
CREATE RULE Tên_Rule AS Biểu_thức
```

Tên_Rule: Kiểm tra miền giá trị dữ liệu được tạo mới
Biểu_thức.

Biểu_thức: Có 3 thành phần:

- Tên biến hình thức phải bắt đầu bằng @
- Toán tử so sánh (>, <, >=, <=, =, ...)
- Giá trị so sánh

Kiểm tra ràng buộc miền giá trị (RULE)

Ví dụ 1: Tạo Rule kiểm tra miền giá trị $SL > 0$.

```
CREATE RULE rule_Soluong_Duong  
AS @Soluong > 0;
```

Ví dụ 2: Tạo Rule kiểm tra miền giá trị chỉ thuộc trong các từ: cái, bộ, thùng, hộp.

```
CREATE RULE rule_DonViTinh  
AS @dvt IN (N'Cái', N'Bộ', N'Thùng', N'Hộp')
```

Kiểm tra ràng buộc miền giá trị (RULE)

- ❑ Áp dụng **RULE** kiểm tra vào bảng

Cú pháp:

```
EXEC sp_bindrule Tên_Rule, Tên_đối_tượng
```

Ví dụ: Áp dụng Rule có tên rule_DonViTinh cho cột đơn vị tính trong bảng HangHoa.

```
EXEC sp_bindrule rule_DonViTinh, 'HangHoa.DonViTinh'
```

Kiểm tra ràng buộc miền giá trị (RULE)

- ❑ Hủy áp dụng **RULE** kiểm tra trong bảng

```
EXEC sp_unbindrule Tên_đối_tượng
```

Ví dụ: Hủy bỏ Rule kiểm tra DonViTinh trong bảng HangHoa

```
EXEC sp_unbindrule 'HangHoa.DonViTinh'
```

- ❑ Xóa **RULE**

```
DROP RULE Tên_Rule
```

Kiểm tra ràng buộc miền giá trị (RULE)

Ví dụ: Xóa Rule `rule_DonViTinh` vừa tạo

```
DROP RULE rule_DonViTinh
```

Chú ý: Phải hủy bỏ áp dụng Rule mới xóa được Rule

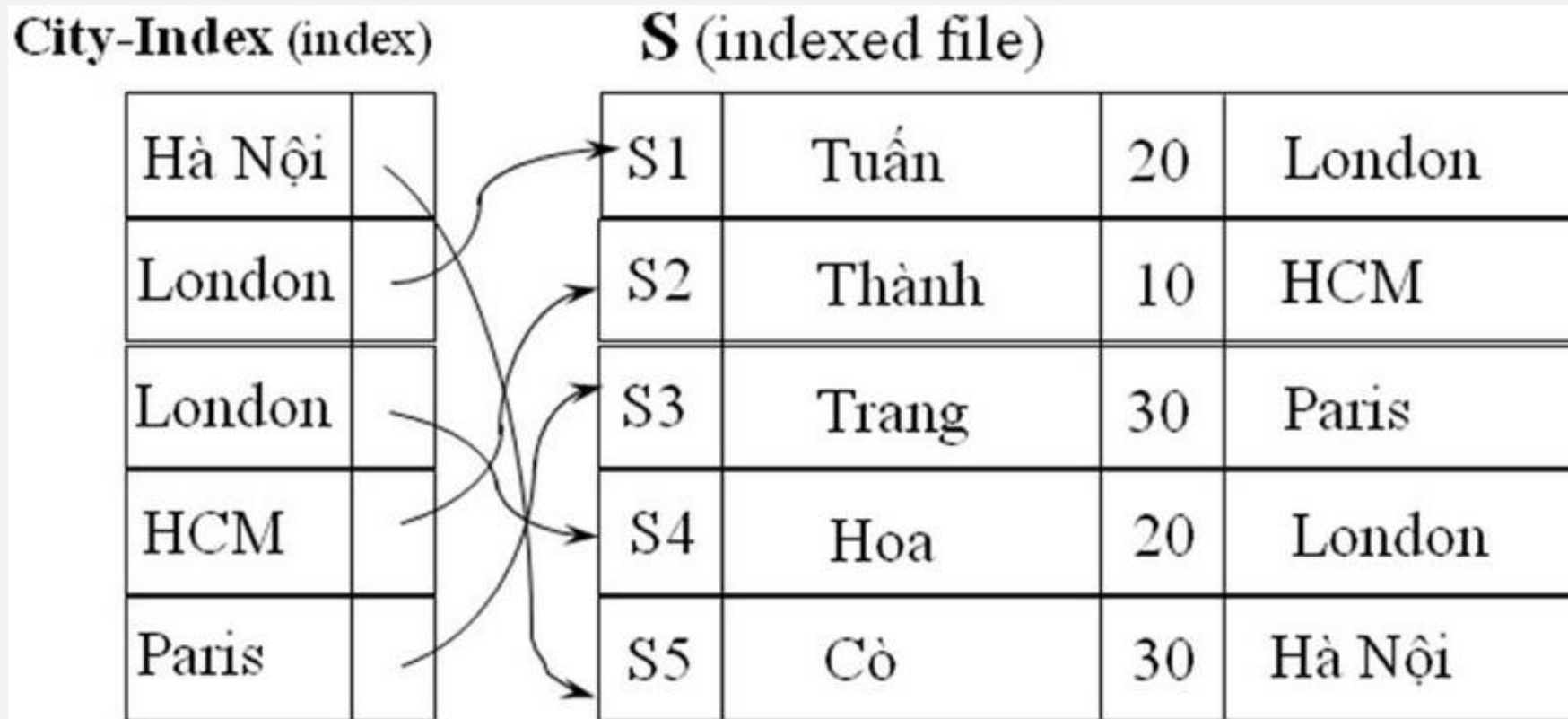
Chỉ mục(Index)

- ❖ **Khái niệm:** Index là một đối tượng trong Datatable (hệ quản trị cơ sở dữ liệu) – là 1 cấu trúc dữ liệu được sử dụng nhằm **tăng tốc độ tìm kiếm** trong 1 table, nó giống như phần mục lục của mỗi cuốn sách.

3. HƯỚNG DẪN SỬ DỤNG INCM SERVER	12
3.1 Khởi động INCM Server	12
3.2 Các biểu tượng và ý nghĩa của nó	12
3.3 Các chức năng điều khiển hệ thống.....	13
3.4 Tính tiền và In hoá đơn.....	17
3.5 Quản lí Hội viên sử dụng thẻ	19
3.6 Quản lí Thực đơn và Phục vụ khách hàng	20
3.7 Quản lí Thu/ Chi và Bảng cân đối thu chi	22
3.8 Tổng kết doanh số.....	23
3.9 Kiểm soát theo dõi.....	24
4. QUẢN TRỊ INCM SERVER	25

Chỉ mục(Index)

- ❖ **Cấu trúc:** Chỉ mục bao gồm ít nhất 2 phần là **Khóa tìm kiếm** và **con trỏ** tới mẫu tin tương ứng trong bảng dữ liệu.



Chỉ mục(Index)

❖ Cú pháp tạo Index:

```
CREATE [CLUSTERED |NONCLUSTERED] INDEX index_name  
ON Table_name (column_name[,column_name]...)
```

❖ Ví dụ: Tạo Index cho cột TenMH trong bảng HangHoa

```
CREATE INDEX index_mh ON HangHoa(TenMH)
```

❖ Cú pháp xóa Index:

```
DROP INDEX Table_name.index_name
```

Chỉ mục (Index)

- ❖ Ví dụ: Xóa Index trong bảng HangHoa

```
DROP INDEX HangHoa.index_mh
```