



HỌC VIỆN KỸ THUẬT MẬT MÃ

KHOA CÔNG NGHỆ THÔNG TIN

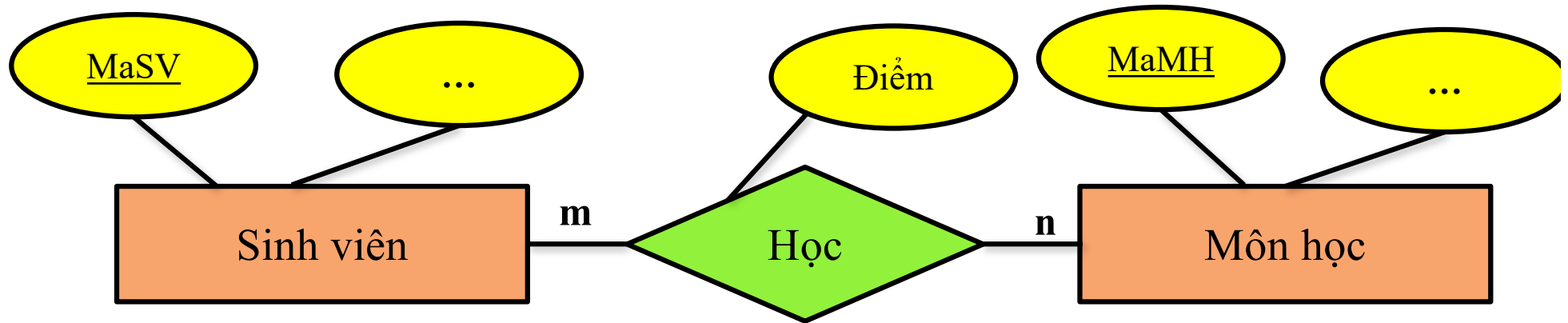
Bài giảng: CSDL & bảng trong SQL Server



“Tri thức là chìa khóa mở cánh cửa tương lai”



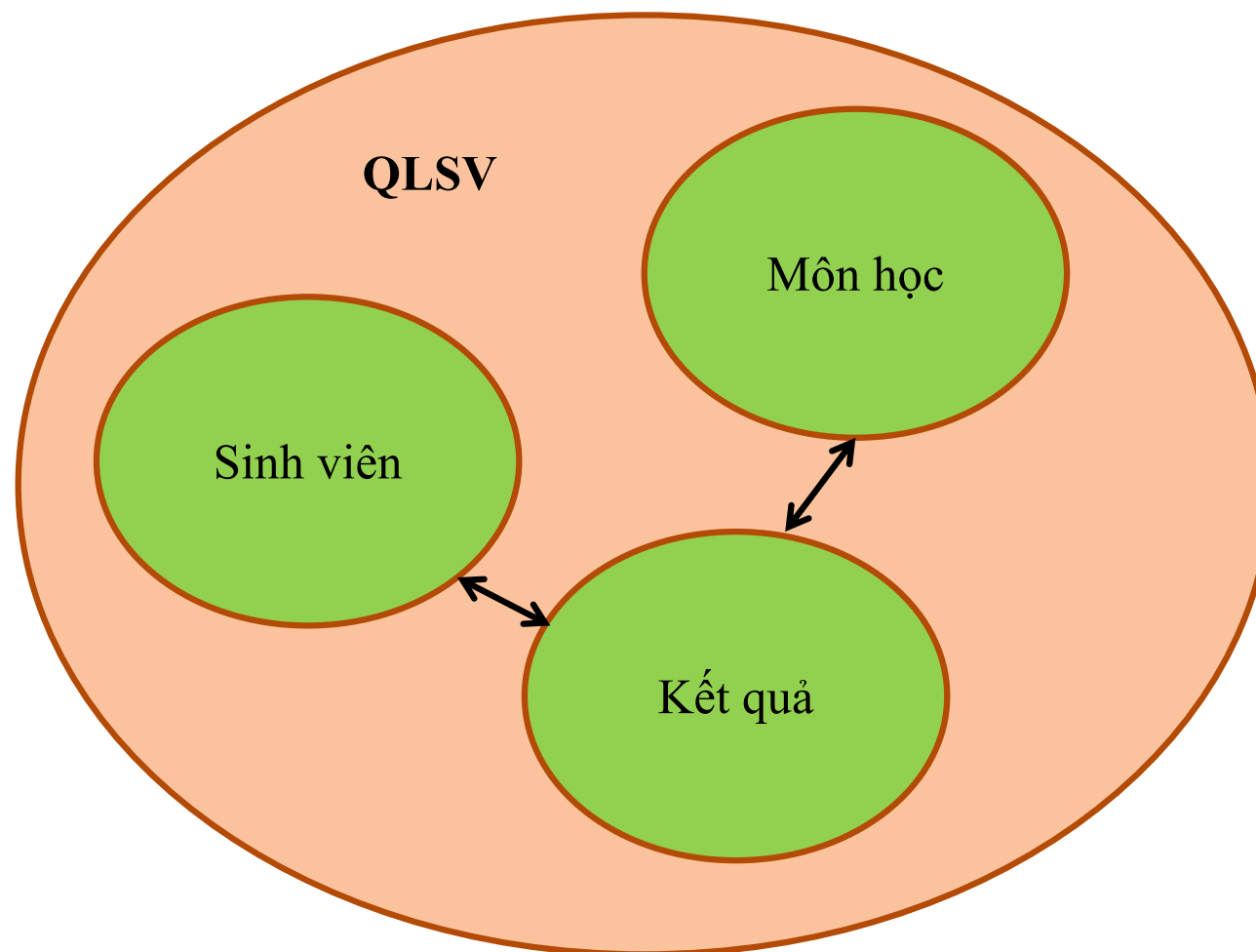
Nhắc lại kiến thức cũ



Sinhvien(MaSV, hotenSV, gioitinh, ngaysinh, quequan, lop)

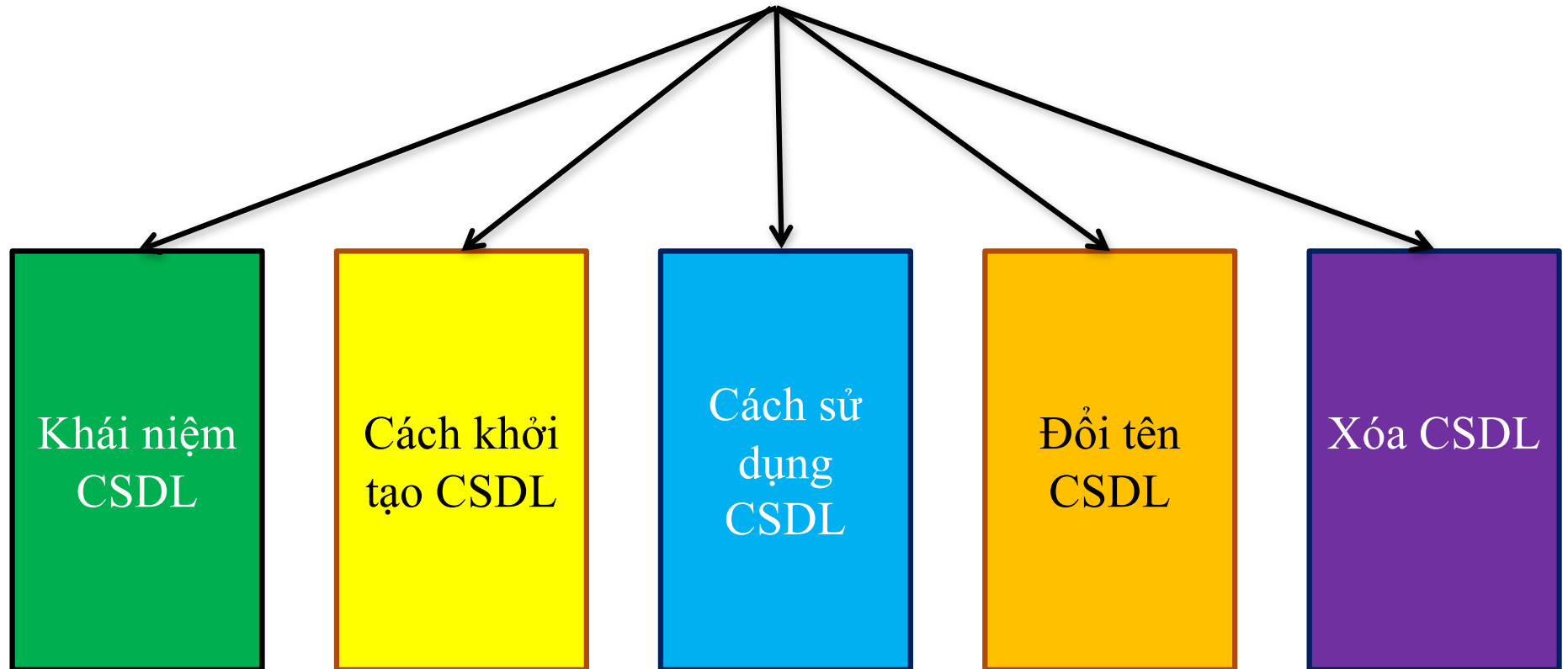
Monhoc(MaMH, TenMH, DVHT)

Ketqua(MaSV, maMH, Diem)





Làm việc với cơ sở dữ liệu





Cơ sở dữ liệu trong SQL

➤ Nhắc lại kiến thức cũ

- ✓ Một Database là tập hợp của rất nhiều dữ liệu phản ánh thế giới thực hoặc một phần của thế giới thực.
- ✓ Có cấu trúc, được lưu trữ tuân theo quy tắc dựa trên lý thuyết toán học.
- ✓ Các dữ liệu trong Database có liên quan với nhau về một lĩnh vực cụ thể, được tổ chức đặc biệt cho việc lưu trữ, tìm kiếm và trích xuất dữ liệu.
- ✓ Được các hệ cơ sở dữ liệu khai thác xử lý, tìm kiếm, tra cứu, sửa đổi, bổ sung hay loại bỏ dữ liệu trong Database.



Cơ sở dữ liệu trong SQL Server

Ở mức logic: Một DATABASE gồm nhiều bảng (TABLE), mỗi bảng được xác định bằng một tên, bảng chứa dữ liệu có cấu trúc và các ràng buộc (CONSTRAINT) định nghĩa trên các bảng.

Ngoài ra, Database còn có khung nhìn (VIEW), các thủ tục/ hàm....

Ở mức vật lý: DATABASE của SQL Server được lưu trữ dưới 3 loại tập tin:

Tập tin dữ liệu (Data-file): gồm 1 tập tin lưu trữ dữ liệu chính (*.mdf) chứa các dữ liệu khởi đầu và các tập tin dữ liệu thứ cấp (*.ndf) chứa dữ liệu phát sinh hoặc không lưu hết trong tập tin lưu trữ chính.

Tập tin nhật ký thao tác (.ldf)* chứa thông tin giao tác, thường dùng để khôi phục Database nếu xảy ra sự cố.



Khởi tạo Database trong SQL Server

Có hai cách khởi tạo Database:

Cách 1: *Tạo Database bằng giao diện (UI)*

Cách 2: *Tạo Database bằng code*



Khởi tạo Database trong SQL Server

Cách 1: Tạo Database bằng giao diện (UI)

Chuột phải vào **Database/ New Database**, xuất hiện cửa sổ như hình

Logical Name	File Type	Filegroup	Initial Size (MB)	Autogrowth / Maxsize	Path
	ROWS ...	PRIMARY	8	By 64 MB, Unlimited	C:\Program Files\Microsoft SQL Server\90\Shared\...
_log	LOG	Not Applicable	8	By 64 MB, Unlimited	C:\Program Files\Microsoft SQL Server\90\Shared\...



Khởi tạo Database trong SQL Server

Cách 2: Tạo Database bằng code

- ✓ Tạo database mặc định
- ✓ Tạo database theo yêu cầu



Khởi tạo Database trong SQL Server

➤ *Tạo database mặc định*

Cú pháp

CREATE DATABASE <Tên_Database>

Ví dụ

CREATE DATABASE QLSV

Các bước thực hiện

Bước 1: Kết nối SQL Server với SQL Server 2019 Management Studio

Bước 2: Chọn New Query hoặc (Ctrl + N)

Bước 3: Gõ code



Khởi tạo Database trong SQL Server

➤ *Tạo database theo yêu cầu*

Cú pháp

```
create database tên_cSDL
on primary --Tạo file data
(
    name = tên_cSDL,
    filename = 'đường_dẫn\tên_cSDL.mdf',
    size = kích_thước_ban_đầu,
    maxsize = kích_thước_tối_đa,
    filegrowth = kích_thước_tăng
)
log on --Tạo file log
(
    name = tên_cSDL_log,
    filename = 'đường_dẫn\tên_cSDL.ldf',
    size = kích_thước_ban_đầu,
    maxsize = kích_thước_tối_đa,
    filegrowth = kích_thước_tăng
)
```



Khởi tạo Database trong SQL Server

Ví dụ Tạo Database **QLSV** với tập tin dữ liệu chính là **QLSV.mdf**, đặt tại thư mục **D:** với dung lượng khởi tạo là **5MB**, tối đa là **50MB** và độ gia tăng kích thước cho phép là **10%**.

```
Create Database QLSV
On Primary
(
Name = QLSV,
FileName = 'D:\QLSV.mdf',
Size = 5MB,
Maxsize = 50MB,
FileGrowth = 10%
)
Log on
(
Name = QLSVlog,
FileName = 'D:\QLSVlog.ldf',
Size = 5MB,
Maxsize = Unlimited,
FileGrowth = 10%
)
```



Làm việc với Database trong SQL Server

➤ Sử dụng Database

Cú pháp `USE <ten_Database>`

Ví dụ `USE QLSV`

➤ Đổi tên Database

`ALTER DATABASE <ten_Database> MODIFY NAME = <ten_moi>`

Ví dụ Đổi tên CSDL “QLSV” thành “QL_SV”

`ALTER DATABASE QLSV MODIFY NAME = QL_SV`

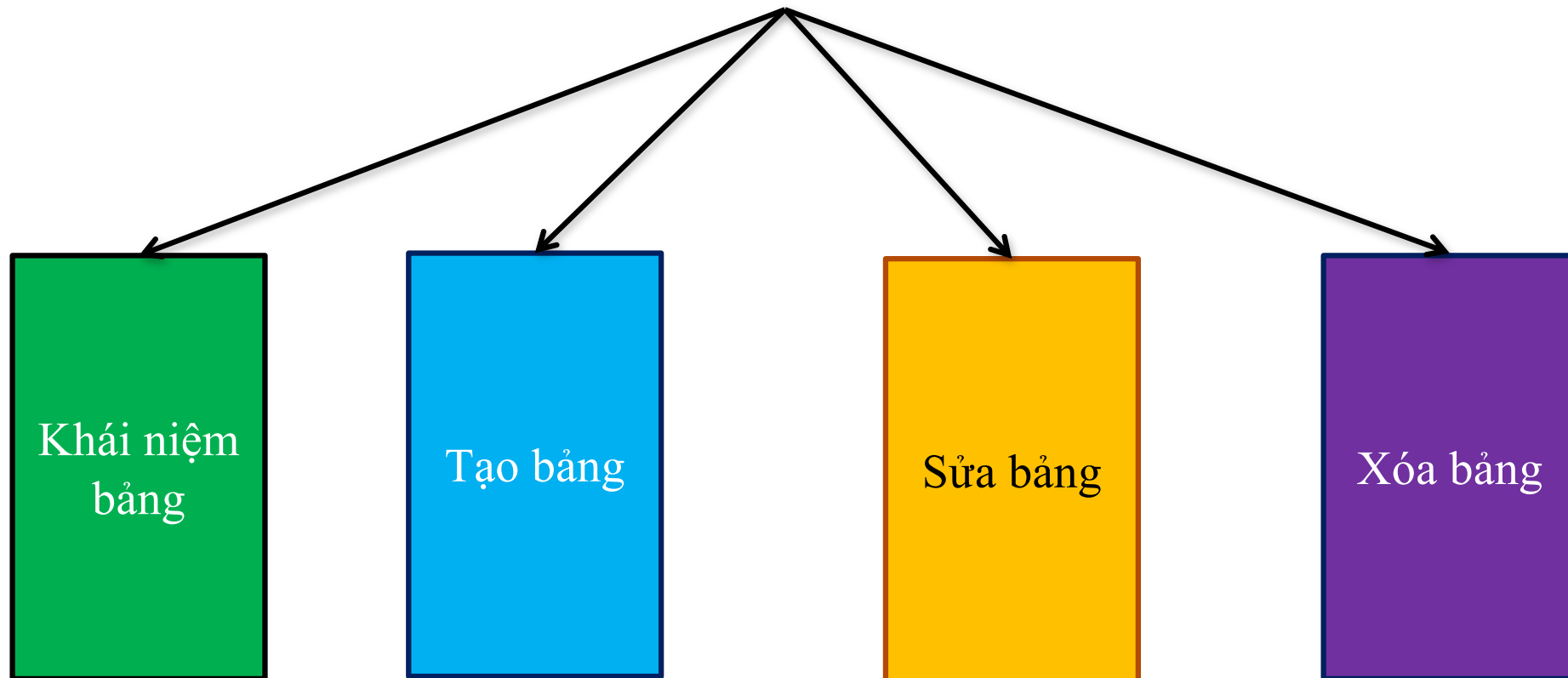
➤ Xóa Database Khi sử dụng lệnh xóa, CSDL sẽ bị xóa khỏi vùng lưu trữ, muốn tạo thì phải thực thi lại lệnh

Cú pháp `DROP DATABASE <ten_Database>`

Ví dụ `DROP DATABASE QL_SV`



Làm việc với bảng





Bảng (Table)

➤ Ý nghĩa

- ✓ Là đối tượng được Database sử dụng để tổ chức và lưu trữ dữ liệu.
- ✓ Mỗi Table trong Database có thể liên kết với một hoặc nhiều Table khác, ở một hoặc nhiều thuộc tính. Một Database bao gồm nhiều Table, giữa các Table có mối liên hệ với nhau thể hiện qua KHÓA CHÍNH & KHÓA NGOẠI.
- ✓ Mỗi bảng được xác định duy nhất bởi tên bảng

Mỗi Table bao gồm:

- ✓ Cột hay còn gọi là các **TRƯỜNG THUỘC TÍNH**. Biểu diễn cho một tính chất của thực thể.
- ✓ Dòng hay còn gọi là các **BẢN GHI**. Biểu diễn cho một thực thể (ứng với một đối tượng)



Tạo bảng

➤ Cú pháp

CREATE TABLE <Tên bảng>

(

<Tên trường 1> <Kiểu dữ liệu> [RBTv] [,...],

<Tên trường 2> <Kiểu dữ liệu> [RBTv] [,...],

<Tên trường n> <Kiểu dữ liệu> [RBTv] [,...]

)

➤ Lưu ý

1. Sau mỗi trường cần có dấu phẩy (,) để ngăn cách với thông tin trường tiếp theo;
2. Sau trường cuối cùng **không cần** có dấu phẩy.



Tạo bảng

➤ Ví dụ

```
create table SINHVIEN  
(  
MaSV char(10),  
HotenSV nvarchar(50),  
GioiTinh nchar(7),  
NgaySinh date,  
QueQuan nvarchar(20),  
Lop varchar(5),  
)
```



Sửa cấu trúc bảng

- ❖ Sử dụng câu lệnh **ALTER TABLE**
- ❖ Câu lệnh này cho phép thực hiện được các thao tác sau:
 1. Bổ sung một cột mới vào bảng.
 2. Xoá một cột khỏi bảng.
 3. Thay đổi định nghĩa kiểu của một cột trong bảng.
 4. Xoá bỏ hoặc bổ sung các ràng buộc cho bảng



Sửa cấu trúc bảng

1. Thêm cột mới vào bảng

Cú pháp

ALTER TABLE *Tên_bảng*

ADD *Tên_cột* *Kiểu_dữ_liệu* [*RBTV*] [*,...*]

Ví dụ

alter table sinhvien **add** email **char**(30)

Chú ý Cột mới luôn được thêm vào cuối bảng

2. Xóa một cột khỏi bảng

Cú pháp

ALTER TABLE *Tên_bảng*

DROP COLUMN *Tên_cột*

Ví dụ

alter table sinhvien **drop column** lop

Lưu ý: Muốn xóa được cột thì phải xóa ràng buộc của cột đó trước



Sửa cấu trúc bảng

3. *Sửa đổi kiểu dữ liệu của cột*

Cú pháp

ALTER TABLE Tên_bảng

ALTER COLUMN Tên_cột Kiểu_dữ_liệu_mới

4. *Đổi tên cột*

Cú pháp

SP_RENAME 'Tên_bảng.Tên_cột_cũ', 'Tên_cột_mới', 'COLUMN'

5. *Đổi tên bảng*

Cú pháp

SP_RENAME 'Tên_bảng_cũ', 'Tên_bảng_mới'

Lưu ý Có dấu nháy hoặc không đều được



Sửa cấu trúc bảng

6. *Xem cấu trúc của bảng*

Cú pháp

```
SP_HELP 'Tên_bảng'
```

7. *Xem dữ liệu các bản ghi trong bảng*

Cú pháp

```
SELECT * FROM Tên_bảng
```



```
create table SINHVIEN
(
  MaSV char(10) primary key,
  HotenSV nvarchar(50) not null,
  GioiTinh nchar(7),
  NgaySinh date not null,
  QueQuan nvarchar(20),
  Lop varchar(5),
)
```

Sinhvien

MaSV	HotenSV	Gioitinh	Ngaysinh	Quequan	Lop
....				



```
alter table sinhvien add email char(30) unique
```

Sinhvien



MaSV	HotenSV	Gioitinh	Ngaysinh	Quequan	Lop	Email
....					



Chèn dữ liệu vào bảng

➤ Ý nghĩa

Các bảng lưu trữ dữ liệu theo các dòng;

Câu lệnh **INSERT INTO** dùng để thêm một dòng mới vào bảng.

➤ Cú pháp

```
INSERT INTO tenbang(cot1, cot2, cot3,...,)  
VALUES (gt1, gt2, gt3,...)
```

➤ Ví dụ

```
insert into SINHVIEN values ('AT1', N'Cao Thu Huyền',  
N'Nữ', 'AT16A', N'Hà Nội', '10/10/1998', 'abc@gmail.com')
```




Chèn dữ liệu vào bảng

➤ Lưu ý

- Nếu dữ liệu kiểu text và kiểu Date, khi chèn phải có ‘ ’, nếu là kiểu nvachar thì phải có tiếp đầu ngữ N’.
- Dữ liệu kiểu Date yêu cầu nhập **tháng/ngày/năm** hoặc **năm/tháng/ngày** hoặc nhập dưới **dạng chuỗi**.
- Dữ liệu dạng tự tăng thì không cần nhập.
- Cột nhập dữ liệu có thể thay đổi nhưng phải tương ứng với cột đưa giá trị vào



Xem dữ liệu trên bảng

➤ Cú pháp *SELECT * FROM <tenbang>*

*SELECT * FROM Sinhvien*

Results		Messages						
	masv	hotensv	gioitinh	lop	quequan	ngaysinh	sdt	email
1	AT1	Cao Thu Huyền	Nữ	AT16A	Hà Nội	1998-10-10	0987878909	abc@gmail.com
2	AT2	Nguyễn Thị Hải	Nữ	AT16K	Hà Nam	1998-11-15	0987845609	cgd@gmail.com
3	AT3	Hoàng Minh Chiến	Nam	AT16A	Hà Nội	1998-11-10	0967878909	ktm@gmail.com
4	CN1	Trần Mạnh Cường	Nam	CT4A	Thái Bình	1998-12-01	0867878909	123@gmail.com
5	CN2	Lê Văn Minh	Nam	CT4A	Thanh Hóa	1998-09-20	0977878809	sutu@gmail.com
6	DT1	Nguyễn Bảo Lâm	Nam	DT3A	Hà Nội	1998-05-12	0677878909	khk@gmail.com
7	DT3	Vũ Tuấn Đạt	Nam	DT3D	Hà Nam	1998-09-13	0677878959	cao@gmail.com



Cập nhật dữ liệu trên bảng - UPDATE

➤ Ý nghĩa

1. UPDATE là truy vấn được sử dụng để chỉnh sửa những bản ghi đã tồn tại trong bảng.
2. Có thể sử dụng mệnh đề WHERE với lệnh UPDATE để cập nhật các hàng được chọn, nếu không muốn tất cả các hàng trong bảng bị ảnh hưởng.

➤ Cú pháp

```
UPDATE Tên_bảng SET cot1 = gtri1, cot2 = gtri2....,  
WHERE [điều_kiện]
```



Cập nhật dữ liệu trên bảng

➤ Ví dụ

```
UPDATE Sinhvien SET hotensv = N'Trần Quang Huy', lop = 'DT3B'  
WHERE masv = 'DT1'
```

➤ Kết quả

Results		Messages						
	masv	hotensv	gioitinh	lop	quequan	ngaysinh	sdt	email
1	AT1	Cao Thu Huyền	Nữ	AT16A	Hà Nội	1998-10-10	0987878909	abc@gmail.com
2	AT2	Nguyễn Thị Hải	Nữ	AT16K	Hà Nam	1998-11-15	0987845609	cgd@gmail.com
3	AT3	Hoàng Minh Chiến	Nam	AT16A	Hà Nội	1998-11-10	0967878909	ktm@gmail.com
4	CN1	Trần Mạnh Cường	Nam	CT4A	Thái Bình	1998-12-01	0867878909	123@gmail.com
5	CN2	Lê Văn Minh	Nam	CT4A	Thanh Hóa	1998-09-20	0877878909	data@gmail.com
6	DT1	Trần Quang Huy	Nam	DT3B	Hà Nội	1998-05-12	0677878909	khk@gmail.com
7	DT3	Vũ Tuấn Đạt	Nam	DT3D	Hà Nam	1998-09-13	0677878959	cao@gmail.com



Xóa bản ghi trong bảng

➤ Ý nghĩa

1. Lệnh DELETE được dùng để xóa một hoặc nhiều bản ghi từ một bảng trong SQL Server.
2. Lệnh DELETE được sử dụng để xóa các hàng từ một bảng. Nếu muốn xóa một hàng cụ thể từ một bảng, sử dụng mệnh đề WHERE.

➤ Cú pháp

```
DELETE FROM Tên_bảng  
[WHERE điều_kiện]
```

➤ Ví dụ

```
DELETE FROM SINHVIEN WHERE MaSV = 'DT1'
```

➤ Yêu cầu So sánh DROP, DELETE VÀ TRUNCATE



Xóa bảng

➤ Ý nghĩa

Đề xóa bảng khỏi CSDL

➤ Cú pháp

DROP **TABLE** *Danh_sách_tên_các_bảng*

➤ Lưu ý

- Câu lệnh Drop Table không thể thực hiện nếu bảng cần xóa được tham chiếu bởi một **Foreign Key**
- Các ràng buộc, chỉ mục, trigger,.. đều bị xóa, nếu tạo lại bảng thì cũng phải tạo lại các đối tượng này
- Sau khi xóa không thể khôi phục lại bảng và dữ liệu bảng

