



# Create and managing Database, Table

# Tạo và quản lý CSDL & Table



# Mục tiêu bài học

- Tạo và quản lý Database
- Tạo và quản lý Table
- Các ràng buộc dữ liệu (Data Constraint)



# Tạo mới Database



Cú pháp:

```
CREATE DATABASE database_name
```

Trong đó:

❑ **database name** : Là tên của CSDL.

Ví dụ: Tạo CSDL có tên là EmployeesMS

```
CREATE DATABASE EmployeesMS
```

# Tạo mới Database

Cú pháp với các tùy chọn file lưu trữ:

```
CREATE DATABASE database_name  
[ ON [ PRIMARY ] <filespec> ]  
[ LOG ON <filespec> ]
```



# Tạo mới Database



Trong đó:

- ❑ **ON** : Chỉ định các file trên đĩa được sử dụng để lưu trữ các phần dữ liệu của database, data files.
- ❑ **PRIMARY** : File đầu tiên được chỉ định trong <filespec> của nhóm filegroup dùng làm file lưu trữ chính. Một CSDL chỉ có thể duy nhất một file primary.
- ❑ **LOG ON** : Chỉ định file log lưu trữ ở bộ nhớ vật lý.

# Tạo mới Database



Cú pháp tạo **filespec**

```
(  
    NAME = logical_file_name ,  
    FILENAME = { 'os_file_name' | 'filesystem_path' } ,  
    [ , SIZE = size [ KB | MB | GB | TB ] ]  
    [ , MAXSIZE = { max_size [ KB | MB | GB | TB ] | UNLIMITED } ]  
    [ , FILEGROWTH = growth_increment [ KB | MB | GB | TB | % ] ]  
)
```

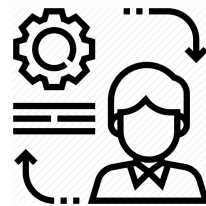
# Tạo mới Database

Trong đó :

- ❑ **NAME** : Chỉ định tên logical cho file. NAME được yêu cầu khi FILENAME được chỉ định.
- ❑ **FILENAME**: Chỉ định đường dẫn đến file lưu trữ vật lý.
- ❑ **SIZE**: Chỉ định kích thước file khởi tạo ban đầu kèm theo đơn vị.
- ❑ **MAXSIZE**: Chỉ định kích thước lớn nhất mà file có thể phát triển đến.
- ❑ **FILEGROWTH**: Chỉ định độ tự động gia tăng của file.



# Tạo mới Database



Ví dụ: tạo Database MyDB với tập tin dữ liệu chính là MyDB\_Data.mdf, dung lượng khởi tạo là 1MB và tối đa là 10MB và độ gia tăng kích thước là 10%.  
Tập tin bản ghi log là MyDB\_Log.ldf với dung lượng ban đầu là 2MB và kích thước tối đa không giới hạn, độ gia tăng dung lượng là 10MB.

# Tạo mới Database

Giải ví dụ

```
USE master
GO
CREATE DATABASE MyDB
ON (
    Name=MyDB_Data,
    FileName='D:\MyDB_Data.mdf',
    Size=10MB,
    MaxSize=100MB,
    FileGrowth=10%
)
```

```
LOG ON (
    Name=MyDB_log,
    FileName='D:\MyDB_Log.ldf',
    Size=2MB,
    MaxSize=UNLIMITED,
    FileGrowth=10%
)
```

# Sửa đổi Database

Cú pháp:

```
ALTER DATABASE database_name  
MODIFY NAME = new_database_name  
<filespec>
```



# Sửa đổi Database

Ví dụ: Xóa thay đổi tên database MyDBT thành MyDB



```
USE master
```

```
GO
```

```
ALTER DATABASE MyDBT MODIFY NAME = MyDB
```

```
GO
```

# Xóa Database



Cú pháp:

```
DROP DATABASE database_name
```

Ví dụ: Xóa database MyDB

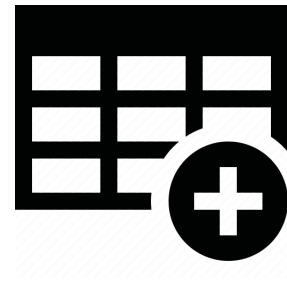
```
USE master
```

```
GO
```

```
DROP DATABASE MyDB
```

```
GO
```

# Tạo mới Table



Cú pháp:

```
CREATE TABLE table_name (  
    < column_definition >  
    < table_constraint >  
)
```

# Ràng buộc (CONSTRAINT)



- Ràng buộc (constraint) là những quy tắc được áp dụng cho các cột trong table. Được sử dụng để xác thực giá trị, dữ liệu nhập vào, đảm bảo tính chính xác cho dữ liệu.
- Một số loại ràng buộc phổ biến có sẵn trong SQL:

NOT NULL	Bảo đảm một cột không thể có giá trị NULL
DEFAULT	Cung cấp một giá trị mặc định cho cột khi không được xác định

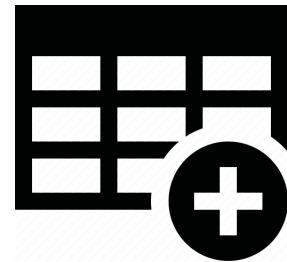
# Ràng buộc (CONSTRAINT)



UNIQUE	Bảo đảm tất cả giá trị trong một cột là khác nhau
PRIMARY KEY	Thiết lập khóa chính, tập giá trị cột là duy nhất
FOREIGN KEY	Thiết lập khóa ngoại, tạo mối liên kết đến bảng khác qua giá trị của cột.
CHECK	Bảo đảm tất cả giá trị trong một cột thỏa mãn các điều kiện nào đó



# Tạo mới Table



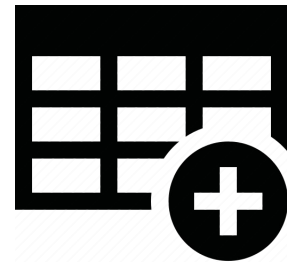
< column\_definition > có cú pháp

```
column_name data_type  
[ , ...n ]
```

Ví dụ:

```
MaSo int, HoTen nvarchar(100), GioiTinh bit
```

# Tạo mới Table



< column\_definition > có thêm ràng buộc:

column\_name data\_type

IDENTITY | NULL | NOT NULL | PRIMARY KEY | UNIQUE |

DEFAULT constant\_expression

Ví dụ:

MaSo int **IDENTITY PRIMARY KEY**, HoTen nvarchar(100) **NOT NULL**

# Tạo mới Table

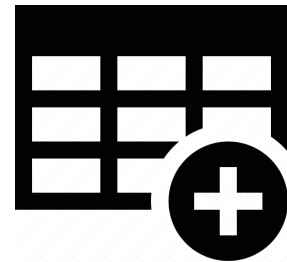
Ví dụ tạo bảng **Lop** trong csdl có tên là **QLDiemSV**:

```
Use QLDiemSV
```

```
Go
```

```
CREATE TABLE Lop (  
    MaLop char(10) PRIMARY KEY,  
    TenLop nvarchar(50) NOT NULL,  
    GioiTinh bit DEFAULT 1  
)
```

# Tạo mới Table



< table\_constraint > có cú pháp

```
CONSTRAINT constraint_name
```

```
[ { PRIMARY KEY | UNIQUE } { ( column [ ,...n ] ) } ]
```

```
| FOREIGN KEY ( column [ ,...n ] )
```

```
REFERENCES ref_table [ ( ref_column [ ,...n ] ) ] }
```

# Tạo mới Table

Ví dụ:

Use QLDiemSV Go

```
CREATE TABLE Lop (
```

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

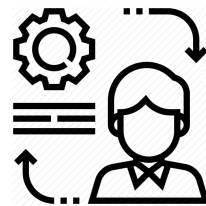
```
    MaKhoa char(10) NOT NULL,
```

```
    CONSTRAINT pk_Lop PRIMARY KEY(MaLop),
```

```
    CONSTRAINT fk_Lop_MaKhoa FOREIGN KEY(MaKhoa)
```

```
    REFERENCES Khoa(MaKhoa)
```

```
)
```



# Ràng buộc (CONSTRAINT)

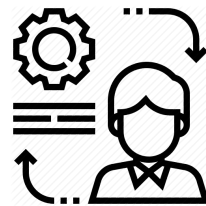
1. Ví dụ thiết lập ràng buộc không được NULL cho cột **Ten** trong bảng **SinhVien**:

➤ Trường hợp tạo mới trong table:

```
CREATE TABLE SinhVien (  
    Ten nvarchar(50) NOT NULL,  
    ...  
)
```

➤ Trường hợp sửa đổi table:

```
ALTER TABLE SinhVien MODIFY Ten nvarchar(50) NOT NULL
```



# Ràng buộc (CONSTRAINT)

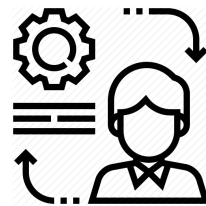
2. Ví dụ thiết lập ràng buộc DEFAULT cho cột GiớiTinh có giá trị là 1 (là Nam) trong bảng SinhVien:

- Trường hợp tạo mới trong table:

```
CREATE TABLE SinhVien (  
    GiớiTinh bit DEFAULT 1  
)
```

- Trường hợp sửa đổi table:

```
ALTER TABLE SinhVien  
ADD CONSTRAINT df_gioi_tinh DEFAULT 1 FOR GiớiTinh
```

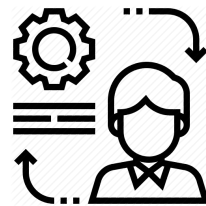


# Ràng buộc (CONSTRAINT)

3. Ví dụ thiết lập ràng buộc PRIMARY KEY cho cột MaSV trong bảng SinhVien

```
CREATE TABLE SinhVien (  
    MaSV varchar(10),  
    CONSTRAINT pk_ma_sv PRIMARY KEY (MaSV)  
)
```

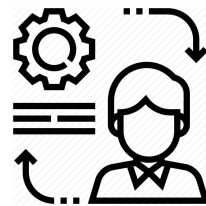




# Ràng buộc (CONSTRAINT)

4. Ví dụ thiết lập ràng buộc FOREIGN KEY cho cột MaLop trong bảng SinhVien nối với cột MaLop bên bảng Lop:

```
CREATE TABLE SinhVien (  
    MaLop varchar(10),  
    CONSTRAINT fk_ma_lop FOREIGN KEY (MaLop)  
    REFERENCES Lop(MaLop)  
)
```



# Ràng buộc (CONSTRAINT)

5. Ví dụ thiết lập ràng buộc CHECK, kiểm tra cột Tuổi trong bảng SinhVien phải nhận giá trị từ 14 trở lên đối với Nam hoặc 16 tuổi đối với Nữ:

```
CREATE TABLE SinhVien (  
    Tuổi int NOT NULL,  
    CONSTRAINT check_mini_age  
    CHECK (  
        (Tuổi >= 14 AND GiớiTinh = 1) OR (Tuổi >= 16 AND GiớiTinh = 0)  
    )  
)
```

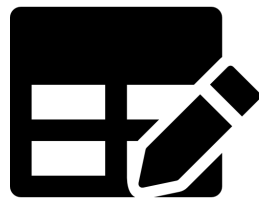


# Ràng buộc (CONSTRAINT)

Một số lưu ý đối với ràng buộc CHECK:

- Ràng buộc kiểm tra trong bảng phải tham chiếu tới cột trong bảng đó, không thể tham chiếu tới cột ở bảng khác
- Có thể định nghĩa trong lệnh CREATE TABLE hoặc ALTER TABLE

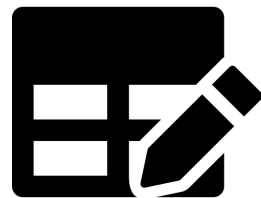
# Sửa đổi Table



Cú pháp

```
ALTER TABLE table_name {  
    ALTER COLUMN column_name { type_name [ NULL | NOT NULL ] }  
    | ADD column_name { type_name [ NULL | NOT NULL ] }  
    | DROP { [ CONSTRAINT ] constraint_name | COLUMN column_name } [  
        ,...n ]  
    } [ ; ]
```

# Sửa đổi Table



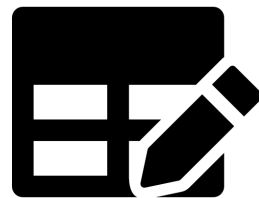
Ví dụ: Thực hiện thêm mới column MaGVCN vào table LOP

```
ALTER TABLE Lop  
ADD MaGVCN char(10) NOT NULL
```

Ví dụ: Thực hiện chuyển kiểu dữ liệu của Khoa thành varchar(10)

```
ALTER TABLE Lop  
ALTER COLUMN Khoa varchar(10)
```

# Sửa đổi Table



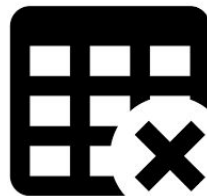
Ví dụ: Thực hiện xóa ràng buộc có tên pk\_Lop trong table LOP

```
ALTER TABLE Lop  
DROP CONSTRAINT pk_Lop
```

Ví dụ: Thực hiện xóa cột TenGV CN trong table Khoa

```
ALTER TABLE Khoa  
DROP COLUMN TenGV CN
```

# Xóa Table



Cú pháp

```
DROP TABLE table_name
```

Ví dụ: Xóa table Lớp

```
DROP TABLE LOP
```

# Tổng kết

- Tạo mới / Sửa đổi / Xóa CSDL
- Tạo mới / Sửa đổi / Xóa Table

