

TRƯỜNG ĐH CÔNG NGHỆ GIAO THÔNG VẬN TẢI

HỆ CƠ SỞ DỮ LIỆU

Giảng viên: Nguyễn Thị Kim Huệ

PHẦN 2:

NGÔN NGỮ ĐỊNH NGHĨA DỮ LIỆU
(Data Definition Language - DDL)

1. GIỚI THIỆU

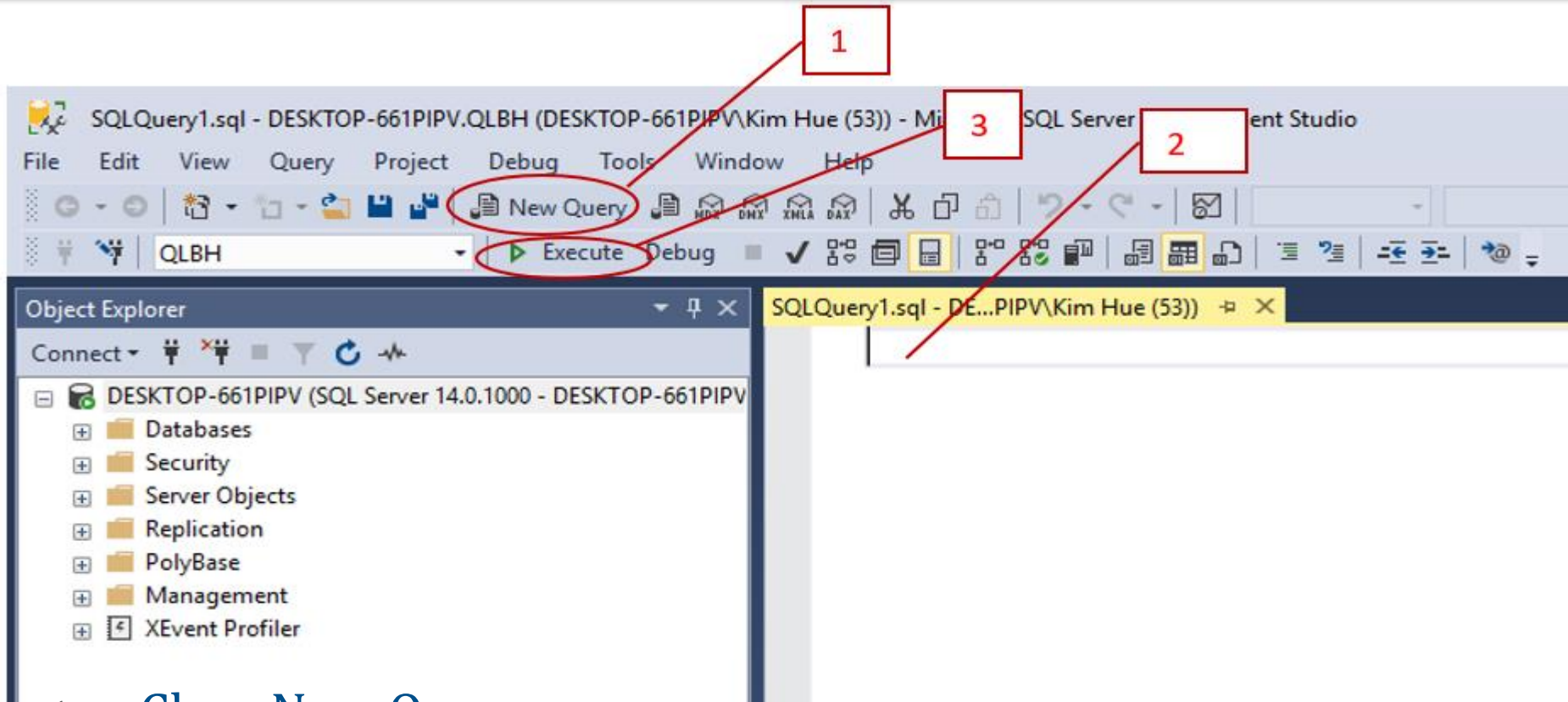
Ngôn ngữ định nghĩa dữ liệu bao gồm các lệnh:

- **CREATE**: định nghĩa và tạo mới đối tượng CSDL
- **ALTER**: thay đổi định nghĩa của đối tượng CSDL
- **DROP**: Xóa đối tượng CSDL đã có

- Có 2 cách để thực hiện:

- + Cách 1: dùng công cụ SQL Server Object Explorer
- + Cách 2: Dùng Query Editor

Dùng Query Editor



1. Chọn New Query
2. Cửa sổ gõ các câu lệnh SQL
3. Chọn Excute để chạy các câu lệnh SQL được chọn trong phần 2

TẠO CSDL

- Cú pháp cơ bản:

CREATE DATABASE *<database-name>*
[ON <filespec>]

- Trong đó:
 - *<database-name>*: Tên CSDL
 - *<filespec>* ::= (**NAME** = logical_file_name,
FILENAME = os_file_name
[, **SIZE** = *n* [KB|MB|GB|TB]]
[, **MAXSIZE** = *max* [KB|MB|GB|TB] | Unlimited]
[, **FILEGROWTH** = *growth* [KB|MB|GB|TB|%]
)

LƯU Ý:
Trong các cú pháp, phần trong **[]** có thể có hoặc không

TẠO CSDL

- **<filespec>**: để điều khiển các thuộc tính cho file được tạo
 - Name: chỉ định tên logical cho file
 - Filename: chỉ định tên, đường dẫn file hệ điều hành (file vật lý)
 - Size: chỉ định kích thước của file, tối thiểu là 3MB
 - Maxsize: chỉ định kích thước tối đa lớn nhất mà file có thể phát triển đến, có từ khóa unlimited chỉ định file được phát triển cho đến khi đĩa bị đầy
 - Filegrowth: chỉ định độ tự động gia tăng của file

TẠO CSDL

- Khi được tạo, 1 Database bao gồm ít nhất 2 file
 - .mdf: lưu trữ các đối tượng trong database
 - .ldf: Lưu trữ quá trình cập nhật, thay đổi dữ liệu, hỗ trợ phục hồi dữ liệu.

TẠO CSDL

- Ví dụ 1: Tạo CSDL có tên là “Quan_ly_SV”

Create Database Quan_ly_SV

B. SỬ DỤNG CSDL:

Cú pháp: **USE <tên CSDL>;**

- Ví dụ: USE QLSV;

C. ĐỔI TÊN CSDL (tên vật lý)

ALTER Database <Tên_CSDL> modify name = <tên mới>

VD: ALTER Database QLSV modify name= QL_SV;

D. XÓA CSDL: 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 <tên CSDL>;**

+ VD: DROP Database Quan_ly_SV;

Chú ý: Không dùng cách xóa thông thường để xóa tệp dữ liệu của SQL mà phải dùng lệnh Drop.

TẠO BẢNG DỮ LIỆU

- Bảng dữ liệu là cấu trúc quan trọng nhất trong CSDL quan hệ.
- Trong bảng, dữ liệu được tổ chức dưới dạng các dòng và cột.
 - Mỗi 1 dòng là 1 bản ghi duy nhất trong bảng
 - Mỗi một cột là một trường

MaSV	TenSV	GT	Ngaysinh	Que	Lop
1	Trần Bảo Trọng	Nam	1995-12-14	Hà Giang	L02
2	Lê Thùy Dương	Nữ	1995-12-14	Hà Nội	L03
3	Trần Phương Thảo	Nam	1996-03-30	Quảng Bình	L01
4	Lê Trường An	Nữ	1999-02-21	Ninh Bình	L01
5	Phạm Thùy Dương	nữ	1999-02-03	Hà Nội	L01
6	Nguyễn An	Nam	1986-02-02	Hà Nội	L03

Ví dụ: bảng trong CSDL quan hệ

CÁC RÀNG BUỘC KHI TẠO BẢNG

- **Ràng buộc toàn vẹn** là những điều kiện bất biến mà tất cả các bộ của những quan hệ có liên quan trong CSDL đều phải thoả mãn ở mọi thời điểm.
- RBTV được áp dụng nhằm đảm bảo tính đúng đắn và hợp lệ của dữ liệu.
- **Ví dụ:**
 - Ngày sinh của sinh viên phải nhỏ hơn ngày nhập học
 - Điểm của sinh viên phải từ 0 đến 10 là một qui định
 - Giới tính của sinh viên chỉ có thể là nam hoặc nữ.
 - Sinh viên đăng ký học với những môn học thuộc khoa mà sinh viên đó học

CÁC RÀNG BUỘC KHI TẠO BẢNG

- RBTV có thể áp dụng ở mức cột hoặc mức bảng:
RB mức cột: chỉ áp dụng ở mức cột, có thể khai báo ngay hoặc sau khi khai báo cột.
RB mức bảng: áp dụng cho toàn bộ bảng, chỉ có thể khai báo sau khi khai báo cột.
- SQL cung cấp 7 công cụ để thực hiện RBTV bao gồm: **Default, Not Null, Unique, Primary Key, Foreign Key, Check.**

CÚ PHÁP TẠO BẢNG

```
CREATE TABLE <table-name>
(
    column1 data-type [RBTV],
    [column2 data-type [RBTV],]
    ...
    [columnn data-type [RBTV],]
    [Constraint <name_RB> RBTV (column1, column2, ...)]
)
```

CÚ PHÁP TẠO BẢNG

- Trong đó:

Table-name: là tên bảng cần tạo, tuân thủ nguyên tắc định danh, không quá 128 ký tự

Column: tên cột cần tạo trong bảng, mỗi bảng có ít nhất một cột


Data-type: xác định kiểu dữ liệu được lưu trữ trong cột, *Kiểu dữ liệu là thuộc tính bắt buộc*

RBTV: gồm các ràng buộc về khuôn dạng dữ liệu hay các ràng buộc về bảo toàn dữ liệu, có thể: **NOT NULL, NULL, UNIQUE, DEFAULT, PRIMARY KEY, IDENTITY, CHECK,...**

Constrain: dùng khi có nhiều hơn một RBTV cùng loại, đặc biệt là với RBTV khóa chính.

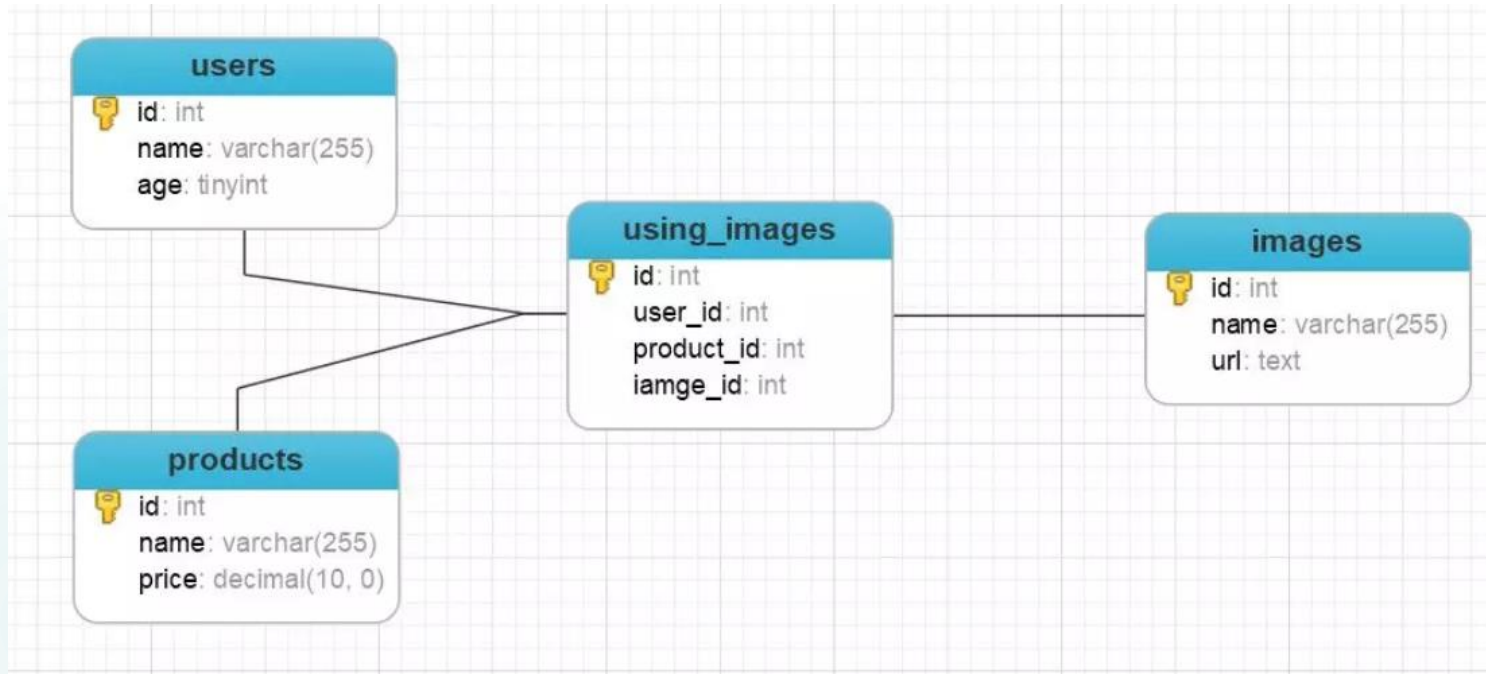
VÍ DỤ TẠO BẢNG

```
CREATE TABLE NHANVIEN (  
    MANV INT,  
    TENNV VARCHAR(50),  
    NS DATE,  
    DCHI VARCHAR(50),  
    GT VARCHAR (3),  
    LUONG INT,  
    MAP INT  
)
```



Bảng chỉ
gồm tên
cột và
kiểu dữ
liệu cột,
chưa có
các
RBTV

Lưu ảnh



Lưu đường dẫn tương đối

Message	Result1	Profile	Status

RÀNG BUỘC CHECK

- Sử dụng nhằm chỉ định điều kiện hợp lệ đối với dữ liệu.

- Cú pháp: **<column> <data type> check (dk)**

Hoặc: **[CONSTRAINT tên-ràng-buộc] CHECK (điều kiện)**

Điều kiện: là một biểu thức so sánh

```
CREATE TABLE DIEM (  
    MASV  VARCHAR(9)  
    MAMH  VARCHAR(50),  
    Diem1 INT  CHECK (Diem1 between  
    0 and 10),  
    Diem2 INT  CHECK (Diem2 >=0 and  
    Diem2<=10)  
)
```

Ràng buộc mức cột

```
CREATE TABLE DIEM (  
    MASV  VARCHAR(9),  
    MAMH  VARCHAR(50),  
    Diem1 INT,  
    Diem2 INT,  
    constraint R1  CHECK ( (Diem1>=0 and  
    Diem1<=10) and ( Diem2 between 0 and  
    10))  
)
```

Ràng buộc mức bảng

Ràng buộc NOT NULL

- *Bắt buộc phải nhập giá trị cho cột khi thêm dữ liệu vào bảng*
- Cú pháp:

<column> <data type> NOT NULL

Ví dụ:

```
CREATE TABLE SINHVIEN (  
    MASV    VARCHAR(9)    NOT NULL  
    TENSV   VARCHAR(50)   NOT NULL,  
)
```

→ Khi nhập dữ liệu, cột MASV và cột TENSX không được bỏ trống.

Ràng buộc PRIMARY KEY

- Được dùng để định nghĩa **KHÓA CHÍNH** của bảng
- Cú pháp:

<column> < data type> PRIMARY KEY

hoặc

[CONSTRAINT tên-ràng-buộc] PRIMARY KEY([d/s cột])

- Ví dụ:

```
CREATE TABLE DIEM (  
    MASV VARCHAR(9) PRIMARY KEY,  
    MAMH VARCHAR(50),  
    Diem1 INT,  
    Diem2 INT )
```

Ràng buộc PRIMARY KEY

- Chú ý: khi bảng khóa chính gồm nhiều cột → Dùng Constraint

- Ví dụ:

```
CREATE TABLE PHANCONG (  
    MASV    varchar(4),  
    MAMH    varchar(100),  
    Constraint PK_Ma PRIMARY KEY( MASV, MAMH)  
)
```

Ràng buộc DEFAULT

- Xác định giá trị mặc định ban đầu cho từng cột
- Cú pháp:

<Tên cột> <data type> DEFAULT <gt;

Hoặc

[CONSTRAINT tên-ràng-buộc] DEFAULT (giá-trị)

- Ví dụ:

```
CREATE TABLE VATTU(  
    MAVTU varchar(4) primary key,  
    TenVtu varchar(100),  
    SoLuong int Default (100)  
)
```

Ràng buộc DEFAULT

- Ví dụ:

```
CREATE TABLE SINHVIEN(  
    MASV varchar(4) primary key ,  
    TenSV varchar(100),  
    GioiTinh varchar(3) Default ( 'Nam' )  
)
```

Ràng buộc UNIQUE

- Dùng khi quy định một cột nào đó cho phép chỉ nhập một giá trị duy nhất cho từng dòng
- Cú pháp:

<column> <type> UNIQUE

Hoặc:

[CONSTRAINT Tên_RB] UNIQUE (DS cột)

- Trong một bảng chỉ có một ràng buộc khoá chính nhưng có thể có nhiều ràng buộc dữ liệu duy nhất

Ràng buộc UNIQUE

- Ví dụ:

```
CREATE TABLE VATTU
```

```
( MAVTU varchar(4) primary key,
```

```
TenVtu varchar(100) UNIQUE )
```

```
CREATE TABLE VATTU
```

```
( MAVTU varchar(4) not null,
```

```
TenVtu varchar(100),
```

```
Constraint UN_Vtu UNIQUE (MaVtu,TenVtu) )
```


Ràng buộc FOREIGN KEY

- Đảm bảo việc nhập dữ liệu cho một cột thuộc tính nào đó **phù hợp tham chiếu** tới một bảng quan hệ khác

- Cú pháp:

REFERENCES *tên-bảng-tham-chiếu(d/s cột tham chiếu)*

hoặc

[CONSTRAINT tên-ràng-buộc] FOREIGN KEY([d/s cột])

REFERENCES *tên-bảng-tham-chiếu(d/s cột tham chiếu)*

[ON DELETE CASCADE | NO ACTION | SET NULL | SET DEFAULT]

[ON UPDATE CASCADE | NO ACTION | SET NULL | SET DEFAULT]

- Việc định nghĩa ràng buộc Foreign Key gồm:
 - Tên cột và danh sách cột của bảng tham gia vào khóa ngoại
 - Tên bảng được tham chiếu bởi khóa ngoại và danh sách cột được tham chiếu đến trong bảng tham chiếu

Ràng buộc FOREIGN KEY

- Cách xử lý bản ghi được sử dụng trong các trường hợp:

CASCADE: tự động xóa (cập nhật) nếu bản ghi được tham chiếu bị xóa (cập nhật)

NO ACTION: (*mặc định*) nếu bản ghi trong bảng tham chiếu đang được tham chiếu bởi một bản ghi bất kỳ trong bảng được định nghĩa thì bản ghi đó không được phép xóa hoặc cập nhật

SET NULL: cập nhật lại khóa ngoại của bản ghi thành giá trị NULL (*nếu cột phép nhận giá trị NULL*)

SET DEFAULT: cập nhật lại khóa ngoại của bản ghi nhận giá trị mặc định (*nếu có giá trị mặc định*)

Ràng buộc FOREIGN KEY

- Ví dụ

CREATE TABLE CTHOADON

(SOHD Varchar(3) **REFERENCES** HOADON(SOHD),
MAVTU Varchar(3) **REFERENCES** VATTU(MAVTU),
Soluong int,
Dongia float)

CREATE TABLE CTHOADON

(SOHD Varchar(3),
MAVTU Varchar(3),
Soluong int,
Dongia float,
FOREIGN KEY (SOHD) **REFERENCES** HOADON(SOHD),
FOREIGN KEY (MAVTU) **REFERENCES** VATTU(MAVTU)
)

Ràng buộc IDENTITY

- Cú pháp:

[**constraint name**] **IDENTITY** [(*start, step*)],

- ❑ Trong một bảng chỉ có tối đa một cột được chỉ định làm cột định danh.
- **start**: là số mà SQL Server sử dụng để cấp phát cho mẫu tin đầu tiên. Mặc định là 1.
- **step**: là chỉ số mà SQL Server cộng lên để cấp phát cho từng mẫu tin kế tiếp. Mặc định là 1.

Ví dụ áp dụng

- Tạo CSDL “Quản lý nhân sự”

- Gồm các bảng:

NHAN VIEN (MaNV, HoTen, GT, NS, QQ, DT, MaPB)

PHONG BAN (MaPB, TenPB, DienThoai) *

CHUC VU (MaCV, TenCV, HSPC)

BACLUONG (MaBL, HSL, HSPC)

DC_PB (MaPB, DiaChi)

NV_CHUCVU (MaNV, MaCV, NgayQD)

- Yêu cầu ràng buộc:

Giới tính chỉ nhận Nam và Nữ

Chức vụ chỉ có “GD”, “PGD”, “TP”, “PP”, “NV”

HSL mặc định là 2.54, HSL từ 2.54 đến 12

HSPC từ 0.4 đến 1.2

NS nhỏ hơn ngày của hệ thống (getdate())

Tạo các ràng buộc khóa chính, khóa ngoại và not null

SỬA CẤU TRÚC BẢNG

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

Thêm, sửa, xóa cột

Thêm cột mới

ALTER TABLE Tên_bảng

ADD Tên_cột Kiểu_dữ_liệu [RBTv]

ALTER TABLE VATTU

ADD Soluong int

Xóa cột

ALTER TABLE Tên_bảng

DROP COLUMN Tên_cột [...]

ALTER TABLE VATTU

DROP COLUMN TenVtu

Sửa đổi kiểu dữ liệu của cột

ALTER TABLE Tên_bảng

ALTER COLUMN Tên_cột
Kiểu_dl_mới

(Kiểu dữ liệu mới phải lớn hơn
kiểu dữ liệu cũ)

ALTER TABLE VATTU

ALTER COLUMN TenVtu Nvarchar(30)

THÊM RÀNG BUỘC CHO CỘT

❑ Cú pháp:

ALTER TABLE Tên_bảng

ADD CONSTRAINT Tên_ràng_buộc Loại_ràng_buộc

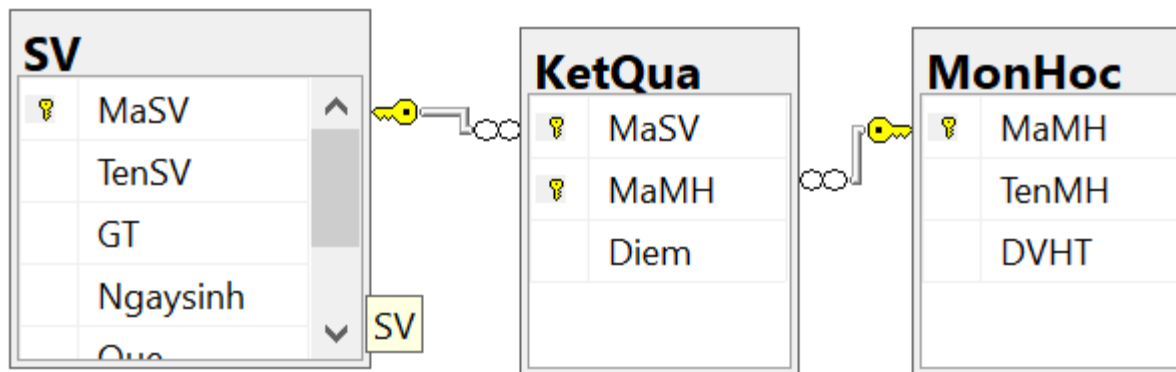
❑ Ví dụ:

ALTER TABLE VATTU

ADD CONSTRAINT CK_NgayNhap

CHECK (Ngaynhap <= GetDate())

Thêm ràng buộc khóa ngoại



```
alter table KetQua
add
  constraint ct_kq_sv
  foreign key (MaSV)
  references SV(MaSV),
  constraint ct_kq_MonHoc
  foreign key (MaMH)
  references MonHoc(MaMH)
```

HỦY RÀNG BUỘC ĐÃ ĐẶT

❑ Cú pháp

ALTER TABLE Tên_bảng

DROP CONSTRAINT Tên_ràng_buộc

❑ Ví dụ:

ALTER TABLE VATTU

DROP CONSTRAINT CK_NgayNhap

Lưu ý

- Khi thêm cột vào bảng đã có ít nhất 1 bản ghi thì cột mới thêm phải cho phép nhận giá trị NULL hoặc phải có giá trị mặc định
- Muốn xóa cột có tồn tại ràng buộc/có tham chiếu khóa ngoài thì phải xóa ràng buộc hoặc khóa ngoài
- Nếu thêm ràng buộc cho bảng đã có dữ liệu mà ràng buộc không thỏa mãn với các dữ liệu đã có thì nó sẽ không thêm vào được.

BẬT TẮT CÁC RÀNG BUỘC

❑ Cú pháp

ALTER TABLE Tên_bảng

NOCHECK CONSTRAINT ALL | Tên_constraint [...]

ALTER TABLE Tên_bảng

CHECK CONSTRAINT ALL | Tên_constraint [...]

ĐỔI TÊN CỘT

❑ Cú pháp

EXEC SP_Rename '*Tên_bảng.Tên_cột*', '*Tên_mới*', '**COLUMN**'

▪ Ví dụ:

EXEC SP_RENAME 'VATTU.MAVTU', 'MAVATTU', 'COLUMN'

ĐỔI TÊN BẢNG

❑ Cú pháp

EXEC sp_rename 'Tên_bảng', 'Tên_mới'

❑ Ví dụ:

EXEC SP_RENAME 'VATTU', 'VT'

Ví dụ - Thay đổi cấu trúc bảng

- *Thêm cột NGHENGHIEP có kiểu CHAR với độ rộng cho phép 20*
- *Thay đổi độ rộng cột thành 50 ký tự*
- *Thêm ràng buộc NOT NULL với trường NGHENGHIEP*
- *Xóa cột NGHENGHIEP*

XÓA BẢNG

- Cú pháp:

DROP TABLE Danh_sách_tên_các_bảng

- Ví dụ:

DROP TABLE VatTu, CTHoaDon

- 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