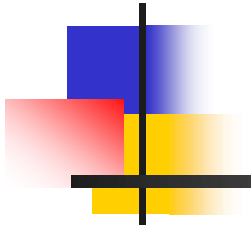




Chương 3

Bảng (Table)

GV: TS. Phạm Thị Thu Thúy
Email: thuthuy@ntu.edu.vn



Nội dung chương 3

- Các kiểu dữ liệu trong SQL
- Câu lệnh định nghĩa dữ liệu
 - Tạo bảng
 - Câu lệnh cập nhật dữ liệu
 - Câu lệnh thay đổi cấu trúc bảng
 - Xóa bảng



Các kiểu dữ liệu trong SQL

- Kiểu dữ liệu là một thuộc tính, nó chỉ định kiểu của dữ liệu và dung lượng có thể lưu trữ của một đối tượng
- SQL Server hỗ trợ một số kiểu dữ liệu được cài sẵn như sau:



Các kiểu dữ liệu trong SQL (tt)

Exact numbers	int	<ul style="list-style-type: none">- sử dụng 4 byte trong bộ nhớ máy tính.- thường được dùng để lưu trữ giá trị số nguyên
	smallint	<ul style="list-style-type: none">-sử dụng 2 byte trong bộ nhớ máy tính.- lưu trữ các số nguyên từ -32768 đến 32767.
	Tinyint	<ul style="list-style-type: none">-Chiếm 1byte trong bộ nhớ-Có giá trị từ 0 đến 255
	Bigint	<ul style="list-style-type: none">- Sử dụng 8bytes trong bộ nhớ máy tính-lưu trữ các số nguyên từ -2^{63}(-223372036854775807) đến $2^{63}-1$
	numeric(p,d)	<ul style="list-style-type: none">-Kiểu số với độ chính xác cố định-Biểu diễn số gồm p chữ số và 1 dấu chấm, có d chữ số bên phải dấu chấm thập phân
	money	<ul style="list-style-type: none">-sử dụng 8 byte trong bộ nhớ máy tính.-Biểu diễn giá trị dữ liệu tiền tệ từ $(-2^{63}/10000)$ đến $(2^{63}-1)$.



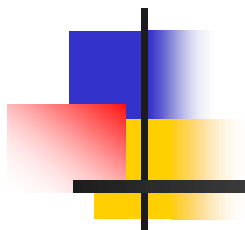
Các kiểu dữ liệu trong SQL (tt)

Approximate numerics	float(n)	<ul style="list-style-type: none">- sử dụng 8 byte trong bộ nhớ máy tính.- Biểu diễn các số dấu chấm động từ $-1.79E+308$ đến $1.79E+308$
	real	<ul style="list-style-type: none">- sử dụng 4 byte trong bộ nhớ máy tính.- biểu diễn các số dấu chấm động có độ chính xác từ $-3.4E+38$ đến $3.4E+38$
Date and time	datetime	<ul style="list-style-type: none">- Biểu diễn ngày và giờ- Được lưu trữ như là 2 số integer, chiếm 2 bytes, chính xác đến phần trăm của giây.
	smalldatetime	<ul style="list-style-type: none">- Biểu diễn ngày và giờ- Chính xác đến phút



Các kiểu dữ liệu trong SQL (tt)

Character String	char	- Lưu trữ dữ liệu ký tự, được cố định kích thước và không hỗ trợ Unicode
	varchar	- Lưu trữ dữ liệu ký tự, độ dài có thể thay đổi và không hỗ trợ Unicode
Unicode Types	text	- Lưu trữ dữ liệu chuỗi, độ dài lớn và không hỗ trợ Unicode
	ntext	- Lưu trữ dữ liệu ký tự, độ dài lớn và có hỗ trợ Unicode
	nchar	- Lưu trữ dữ liệu ký tự, được cố định kích thước và có
	nvarchar	- Lưu trữ dữ liệu ký tự, độ dài có thể thay đổi và có hỗ trợ Unicode

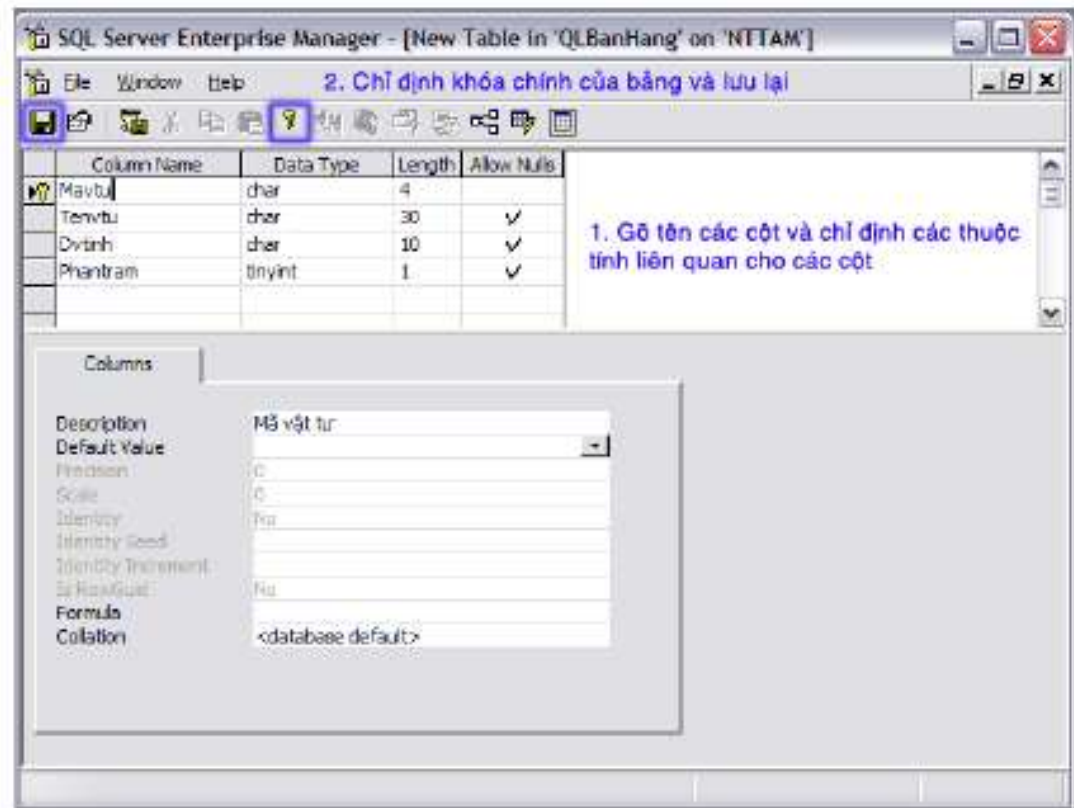


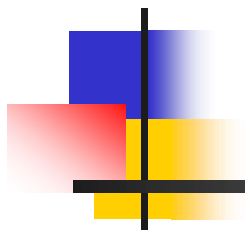
Các kiểu dữ liệu trong SQL (tt)

- Kiểu dữ liệu Date/ Time mới trong SQL server 2012
 - Date: Kiểu ngày
 - Time: Kiểu giờ

Tạo bảng – Khái niệm bảng

- Tên bảng
- Tên cột
- Kiểu dữ liệu
 - Độ dài dữ liệu
 - Số ký số lưu trữ
 - Số số lẻ lưu trữ
- Thuộc tính trên cột
 - Allow null
 - Identity
 - Default value





Tạo bảng

Cú pháp:

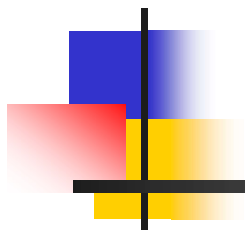
```
CREATE TABLE <tên bảng>(  
    <tên cột 1><kiểu dữ liệu>[CONSTRAINT<tên 1>],  
    ( <tên cột 2><kiểu dữ liệu>[CONSTRAINT<tên  
2>],  
    ...  
    ( <tên cột n><kiểu dữ liệu>[CONSTRAINT<tên n>]  
    [,CONSTRAINT <tên 1>][, CONSTRAINT <tên 2>]  
    ...  
    [,CONSTRAINT <tên n>]  
);
```



Tạo bảng (tt)

Trong đó, cú pháp khai báo của các ràng buộc toàn vẹn như sau:

- `[CONSTRAINT <tên RBTV>] NULL|NOT
NULL|UNIQUE[(<tên cột i>,<tên cột j>...)]|
PRIMARY KEY[(<tên cột i>,<tên cột j>...)]|
FOREIGN KEY [[(<tên cột i>,<tên cột
j>...)] REFERENCES <tên bảng>(<tên cột
i>,<tên cột j>...)]|
CHECK (<điều kiện>)`



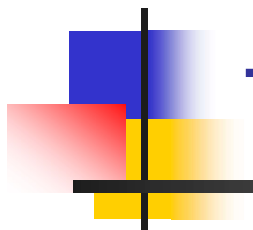
Tạo bảng (tt)

Ví dụ: Tạo bảng với lược đồ quan hệ sau:

HANGHOA (MaHH, TenHH, DVT)

- Mã hàng hóa là khóa chính, tên hàng và đơn vị tính. Tất cả không được rỗng.

```
Create Table Hanghoa (  
MaHH varchar(10) Not Null Primary key,  
TenHH nvarchar(50) Not Null,  
DVT varchar(5) Not Null  
)
```



Tạo bảng – cột được phép null

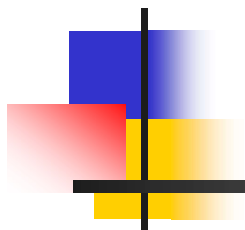
- Đặc trưng về khả năng null của một cột quyết định các hàng trong bảng có thể chứa giá trị null cho cột đó.
- Khả năng Null của một cột có thể được định nghĩa khi tạo một bảng.
 - Từ khóa NULL được sử dụng để chỉ ra rằng giá trị null là được phép trong cột
 - Từ khóa NOT NULL được sử dụng để chỉ ra rằng giá trị null là không được phép



Tạo bảng – cột được phép null

- Ví dụ:

```
CREATE TABLE Nhanvien(  
    manv char(5) NOT NULL,  
    tennv nvarchar(30),  
    manqly char(5) NULL
```



Tạo bảng – Định nghĩa DEFAULT

- Ràng buộc default có thể được tạo ra tại thời điểm tạo bảng hoặc thêm sau khi bảng được tạo.
- Với một cột, chỉ có thể tạo được một giá trị default.
- Giá trị default có thể là một hằng, một hàm hệ thống, một biến toàn cục, hoặc một hàm do người dùng định nghĩa.



Tạo bảng – Định nghĩa DEFAULT (tt)

- Định nghĩa default trong khi tạo bảng:

```
CREATE TABLE <tên bảng> <tên cột> <KDL>  
[NULL|NOT NULL]  
[CONSTRAINT <tên ràng buộc>] DEFAULT  
    <biểu thức>
```

Ví dụ:

```
CREATE TABLE StoreProduct(  
    ProductID int NOT NULL,  
    Name varchar(40) NOT NULL,  
    Price money NOT NULL DEFAULT (100)  
)
```


Tạo bảng - Định nghĩa DEFAULT (tt)

- Định nghĩa default đối với một bảng đã tồn tại:

ALTER TABLE <tên bảng>

**ADD [CONSTRAINT <tên ràng buộc>] DEFAULT
<biểu thức> FOR <tên cột>**

Tạo bảng - 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 hai thành phần:
 - Giá trị khởi đầu
 - Giá trị tăng
-

Tạo bảng - Thuộc tính IDENTITY (tt)

- **Cú pháp:**

- `CREATE TABLE <table_name> (column_name data_type [IDENTITY [(seed_value, increment_value)] NOT NULL)`
 - Trong đó, **seed_value** là giá trị khởi đầu .
 - **increment_value** là giá trị tăng.

- **Ví dụ:**

```
CREATE TABLE ContactPhone (  
    Person_ID int IDENTITY(500,1) NOT NULL,  
    MobileNumber bigint NOT NULL  
)
```

Tạo bảng – Ràng buộc

- **Ràng buộc là thuộc tính được chỉ định cho một cột hoặc một tập hợp các cột trong bảng để ngăn ngừa các giá trị không nhất quán được nhập vào.**
 - **SQL Server hỗ trợ các loại ràng buộc sau:**
 - **PRIMARY KEY**
 - **UNIQUE**
 - **FOREIGN KEY**
 - **CHECK**
 - **NOT NULL**
-

Tạo bảng – Ràng buộc

- Primary Key (Khoá chính)

- Ràng buộc PRIMARY KEY được sử dụng để tạo một khóa chính và đảm bảo toàn vẹn thực thể bảng.
- Cú pháp để thêm một khóa chính trong khi tạo bảng:

- **CREATE TABLE** <tên bảng> (
 <tên cột> <kiểu dữ liệu> PRIMARY KEY [ds cột]
)

- **CREATE TABLE** <tên bảng> (
 <tên cột> <kdl> [ds cột] CONSTRAINT <tên RB>
 PRIMARY KEY)

Tạo bảng – Ràng buộc - Primary Key (Khoá chính) (tt)

- Ví dụ: Tạo bảng PhongBan với khóa chính là maphong

```
CREATE TABLE PhongBan (  
    maphong      int      PRIMARY KEY, tenphong  
    nvarchar(20), trphong char(5) ,  
    diadiem nvarchar(20) )
```

Hoặc có thể viết như sau:

```
CREATE TABLE PhongBan (  
    maphong int constraint pk_pb PRIMARY KEY,  
    tenphong nvarchar(20), trphong char(5), diadiem  
    nvarchar(20) )
```

Tạo bảng – Ràng buộc

- UNIQUE - Khóa duy nhất

- **Ràng buộc khóa duy nhất:**
 - được sử dụng để bảo đảm rằng chỉ các giá trị duy nhất được nhập vào trong cột hoặc một tập hợp các cột (→ cho phép nhà phát triển chắc chắn rằng không có các giá trị trùng lặp được nhập vào)
 - Được sử dụng trong câu lệnh create table để định nghĩa khóa phụ cho bảng
 - Ràng buộc UNIQUE cho phép null
-

Tạo bảng – Ràng buộc

- UNIQUE - Khóa duy nhất (tt)

- Cú pháp:

- CREATE TABLE <tên bảng> ([ds cột,] <tên cột> <kdl> [CONSTRAINT <tên RB>] UNIQUE [,ds cột])

- Ví dụ:

```
CREATE TABLE KhachHang (  
    makh int PRIMARY KEY,  
    tenkh nvarchar(30) ,  
    diachi nvarchar(30) ,  
    sodt char(10) UNIQUE )
```

Tạo bảng – 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
- Ràng buộc khóa ngoại được sử dụng để đảm bảo toàn vẹn tham chiếu
- Cú pháp:
 - CREATE TABLE <tên bảng 1>(
[ds cột,]
<tên cột> <kdl> [CONSTRAINT <tên RB>] FOREIGN KEY REFERENCES
<tên bảng 2>(cột làm khóa chính) [, ds cột])

Tạo bảng – Ràng buộc

- Khóa ngoại (foreign key)

- Ví dụ:

```
CREATE TABLE Nhanvien(  
    manv char(5) NOT NULL,  
    tennv nvarchar(30),  
    manqly char(5),  
    phong int foreign key references  
        PhongBan (maphong)  
)
```

- Hoặc có thể viết như sau:

Tạo bảng – Ràng buộc

- Khóa ngoại (foreign key)

- Ví dụ: (tt)
- Hoặc có thể viết như sau:

```
CREATE TABLE Nhanvien (  
    manv      char(5)      NOT      NULL, tennv  
                nvarchar(30) ,  
    manqly    char(5) , phong int  
    constraint fk_nv_pb foreign key(phong)  
    references PhongBan (maphong)  
)
```

Tạo bảng – Ràng buộc

- Ràng buộc kiểm tra (Check)

- **Ràng buộc check**
 - Được sử dụng để chỉ định điều kiện hợp lệ đối với dữ liệu
- **Cú pháp:**
 - `CREATE TABLE <Tên bảng> (
[ds cột,] <tên cột> <kdl> [CONSTRAINT
<tên RB>] CHECK (<điều kiện>)`
 - Ví dụ:

```
Create table KetQua(  
    masv int, mamh char(4),  
    diem real,  
    lanthi int check(lanthat<3)
```



Ví dụ - RBTV

```
CREATE TABLE NHANVIEN (  
    HONV VARCHAR(10) NOT NULL,  
    TENLOT VARCHAR(20) NOT NULL,  
    TENNV VARCHAR(10) NOT NULL,  
    MANV CHAR(9) PRIMARY KEY,  
    NGSINH DATETIME,  
    DCHI VARCHAR(50),  
    PHAI CHAR(3) CHECK (PHAI IN ('Nam', 'Nu')),  
    LUONG INT DEFAULT (10000),  
    MA_NQL CHAR(9),  
    PHG INT  
)
```



Ví dụ - RBTv

```
CREATE TABLE PHONGBAN (  
    TENPB VARCHAR(20) UNIQUE,  
    MAPHG INT NOT NULL,  
    TRPHG CHAR(9),  
    NG_NHANCHUC DATETIME DEFAULT (GETDATE())  
)
```

```
CREATE TABLE PHANCONG (  
    MA_NVIENT CHAR(9) FOREIGN KEY (MA_NVIENT)  
        REFERENCES NHANVIEN(MANV),  
    SODA INT REFERENCES DEAN(MADA),  
    THOIGIAN DECIMAL(3,1)  
)
```




Ví dụ - Đặt tên cho RBTV

```
CREATE TABLE NHANVIEN (  
    HONV VARCHAR(10) CONSTRAINT NV_HONV_NN NOT  
    NULL,  
    TENLOT VARCHAR(20) NOT NULL,  
    TENNV VARCHAR(10) NOT NULL,  
    MANV CHAR(9) CONSTRAINT NV_MANV_PK PRIMARY KEY,  
    NGSINH DATETIME,  
    DCHI VARCHAR(50),  
    PHAI CHAR(3) CONSTRAINT NV_PHAICHK  
        CHECK (PHAI IN ('Nam', 'Nu')),  
    LUONG INT CONSTRAINT NV_LUONG_DF DEFAULT (10000),  
    MA_NQL CHAR(9),  
    PHG INT  
)
```



Ví dụ - Đặt tên cho RBTV

```
CREATE TABLE PHANCONG (  
    MA_NVIEN CHAR(9),  
    SODA INT,  
    THOIGIAN DECIMAL(3,1),  
    CONSTRAINT gidoK PRIMARY KEY (MA_NVIEN, SODA),  
    CONSTRAINT PC_MANVIEN_FK FOREIGN KEY (MA_NVIEN)  
        REFERENCES NHANVIEN(MANV),  
    CONSTRAINT PC_SODA_FK FOREIGN KEY (SODA)  
        REFERENCES DEAN(MADA)  
)
```

Nội dung chương 3

- Các kiểu dữ liệu trong SQL
 - **Câu lệnh định nghĩa dữ liệu**
 - Tạo bảng
 - **Câu lệnh cập nhật dữ liệu**
 - **Câu lệnh thay đổi cấu trúc bảng**
 - **Xóa bảng**
-

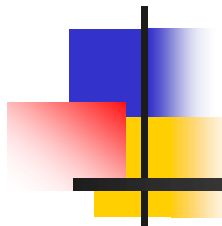
Câu lệnh cập nhật dữ liệu

- Là câu lệnh dùng để cập nhật dữ liệu.
 - Bao gồm các câu lệnh: **INSERT, UPDATE, DELETE.**
-



Lệnh INSERT

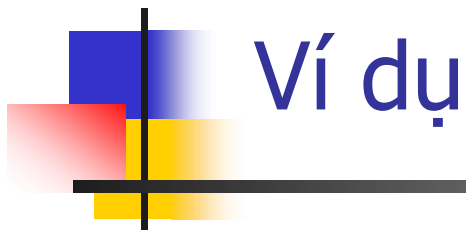
- Dùng để thêm 1 hay nhiều dòng vào bảng
- Để thêm dữ liệu
 - Tên quan hệ
 - Danh sách các thuộc tính cần thêm dữ liệu
 - Danh sách các giá trị tương ứng



Lệnh INSERT (tt)

- Cú pháp (thêm 1 dòng)

**INSERT INTO <tên bảng>(<danh sách các thuộc tính>)
VALUES (<danh sách các giá trị>)**



```
INSERT INTO NHANVIEN(HONV, TENLOT, TENNV, MANV)
VALUES ('Le', 'Van', 'Tuyen', '635635635')
```

```
INSERT INTO NHANVIEN(HONV, TENLOT, TENNV, MANV,
DCHI)
VALUES ('Le', 'Van', 'Tuyen', '635635635', NULL)
```

```
INSERT INTO NHANVIEN
VALUES ('Le', 'Van', 'Tuyen', '635635635', '12/30/1952', '98
HV', 'Nam', '37000', 4)
```




Lệnh INSERT (tt)

- Nhận xét

- Thứ tự các giá trị phải trùng với thứ tự các cột
- Có thể thêm giá trị NULL ở những thuộc tính không là khóa chính và NOT NULL
- Câu lệnh INSERT sẽ gặp lỗi nếu vi phạm RBTV
 - Khóa chính
 - Tham chiếu
 - NOT NULL - các thuộc tính có ràng buộc NOT NULL bắt buộc phải có giá trị



Lệnh INSERT (tt)

- Cú pháp (thêm nhiều dòng)

INSERT INTO <tên bảng>(<danh sách các thuộc tính> <câu truy vấn con>



Ví dụ

```
CREATE TABLE THONGKE_PB (  
    TENPHG VARCHAR(20),  
    SL_NV INT,  
    LUONG_TC INT  
)
```

```
INSERT INTO THONGKE_PB(TENPHG, SL_NV, LUONG_TC)  
SELECT TENPHG, COUNT(MANV), SUM(LUONG)  
FROM NHANVIEN, PHONGBAN  
WHERE PHG=MAPHG  
GROUP BY TENPHG
```



Lệnh DELETE

- Dùng để xóa các dòng của bảng
- Cú pháp

```
DELETE FROM <tên bảng>  
[WHERE <điều kiện>]
```



```
DELETE FROM NHANVIEN  
WHERE HONV='Tran'
```

```
DELETE FROM NHANVIEN  
WHERE MANV='345345345'
```

```
DELETE FROM NHANVIEN
```



Ví dụ 24

- Xóa đi những nhân viên ở phòng 'Nghien cuu'



Lệnh DELETE (tt)

- Nhận xét

- Số lượng số dòng bị xóa phụ thuộc vào điều kiện ở mệnh đề WHERE
- Nếu không chỉ định điều kiện ở mệnh đề WHERE, tất cả các dòng trong bảng sẽ bị xóa
- Lệnh DELETE có thể gây ra vi phạm RB tham chiếu
 - Không cho xóa
 - Xóa luôn những dòng có giá trị đang tham chiếu đến
 - CASCADE
 - Đặt NULL cho những giá trị tham chiếu



Lệnh DELETE (tt)

MANV	HONV	TENLOT	TENNV	NGSINH	DCHI	PHAI	LUONG	MA_NQL	PHG
333445555	Nguyen	Thanh	Tung	12/08/1955	638 NVC Q5	Nam	40000	888665555	5
987987987	Nguyen	Manh	Hung	09/15/1962	Ba Ria VT	Nam	38000	333445555	5
453453453	Tran	Thanh	Tam	07/31/1972	543 MTL Q1	Nu	25000	333445555	5
999887777	Bui	Ngoc	Hang	07/19/1968	33 NTH Q1	Nu	38000	987654321	4
987654321	Le	Quynh	Nhu	07620/1951	219 TD Q3	Nu	43000	888665555	4
987987987	Tran	Hong	Quang	04/08/1969	980 LHP Q5	Nam	25000	987654321	4
888665555	Pham	Van	Vinh	11/10/1945	450 TV HN	Nam	55000	NULL	1

MA_NVIEN	SODA	THOIGIAN
333445555	10	10.0
888665555	20	20.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
453453453	1	20.0



Lệnh DELETE (tt)

TENPHG	MAPHG	MA_NVIEN	NG_NHANCHUC
Nghien cuu	5	333445555	05/22/1988
Dieu hanh	4	987987987	01/01/1995
Quan ly	1	888665555	06/19/1981

MANV	HONV	TENLOT	TENNV	NGSINH	DCHI	PHAI	LUONG	MA_NQL	PHG
333445555	Nguyen	Thanh	Tung	12/08/1955	638 NVC Q5	Nam	40000	888665555	NULL
987987987	Nguyen	Manh	Hung	09/15/1962	Ba Ria VT	Nam	38000	333445555	NULL
453453453	Tran	Thanh	Tam	07/31/1972	543 MTL Q1	Nu	25000	333445555	NULL
999887777	Bui	Ngoc	Hang	07/19/1968	33 NTH Q1	Nu	38000	987654321	4
987654321	Le	Quynh	Nhu	07620/1951	219 TD Q3	Nu	43000	888665555	4
987987987	Tran	Hong	Quang	04/08/1969	980 LHP Q5	Nam	25000	987654321	4
888665555	Pham	Van	Vinh	11/10/1945	450 TV HN	Nam	55000	NULL	1



Lệnh UPDATE

- Dùng để thay đổi giá trị của thuộc tính cho các dòng của bảng
- Cú pháp

```
UPDATE <tên bảng>  
SET <tên thuộc tính>=<giá trị mới>,  
    <tên thuộc tính>=<giá trị mới>,  
    ...  
[WHERE <điều kiện>]
```



Ví dụ

```
UPDATE NHANVIEN  
SET  NGSINH='08/12/1965'  
WHERE MANV='333445555'
```

```
DELETE NHANVIEN  
SET  LUONG=LUONG*1.1
```



Ví dụ 25

- Với đề án có mã số 10, hãy thay đổi nơi thực hiện đề án thành 'Vung Tau' và phòng ban phụ trách là phòng 5

```
UPDATE DEAN  
SET DIADIEM_DA='Vung Tau', PHONG=5  
WHERE MADA=10
```



Lệnh UPDATE

- Nhận xét

- Những dòng thỏa điều kiện tại mệnh đề WHERE sẽ được cập nhật giá trị mới
- Nếu không chỉ định điều kiện ở mệnh đề WHERE, tất cả các dòng trong bảng sẽ bị cập nhật
- Lệnh UPDATE có thể gây ra vi phạm RB tham chiếu
 - Không cho sửa
 - Sửa luôn những dòng có giá trị đang tham chiếu đến
 - CASCADE

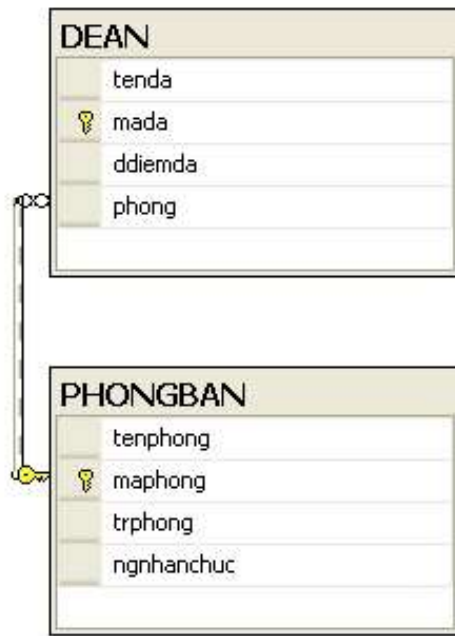
Câu lệnh cập nhật dữ liệu (tt)

- Một số lưu ý:
 - Thêm ký tự N trước chuỗi Unicode
 - Ví dụ:

```
insert            into      NhanVien
values ( 'NV01' ,N' Nguyễn      văn      Trường' ,
        'Nam' )
```
 - Thuộc tính NOT NULL
 - Nếu thuộc tính được khai báo là NOT NULL thì bắt buộc phải có giá trị khi nhập 1 bộ vào bảng

Câu lệnh cập nhật dữ liệu (tt)

- Một số lưu ý (tt)
 - Nhập dữ liệu khi đã có ràng buộc khóa ngoại



- Trường hợp 1:

- Cách 1:

- B1: Nhập PHONGBAN

- B2: Nhập DEAN

- Cách 2:

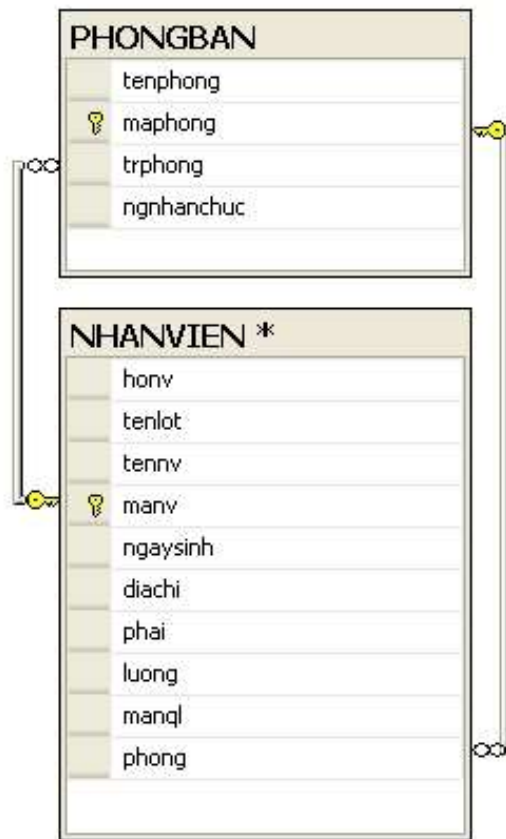
- B1: Nhập DEAN, nhập phong = null

- B2: Nhập PHONGBAN

- B3: Cập nhật DEAN

Câu lệnh cập nhật dữ liệu (tt)

- Một số lưu ý (tt)
 - Nhập dữ liệu khi đã có ràng buộc khóa ngoại



- Trường hợp 2:

- Cách 1:

- B1: Nhập NHANVIEN, đặt phong = null
 - B2: Nhập PHONGBAN
 - B3: Cập nhật thuộc tính phong của NHANVIEN

- Cách 2:

- B1: Nhập PHONGBAN, đặt trphong = null
 - B2: Nhập NHANVIEN
 - B3: Cập nhật PHONGBAN

Câu lệnh cập nhật dữ liệu (tt)

- Một số lưu ý (tt)
 - Nhập dữ liệu khi đã có ràng buộc khóa ngoại



- Trường hợp 3:

- Cách 1:

- Những nhân viên có manql là null thì nhập trước
 - Sau đó nhập những nhân viên mà đã nhập thông tin người quản lý nhân viên đó.

- Cách 2:

- B1: Nhập NHANVIEN đặt manql = null
 - B2: Cập nhật manql của NHANVIEN

Nội dung chương 3

- Các kiểu dữ liệu trong SQL
- **Câu lệnh định nghĩa dữ liệu**
 - Tạo bảng
 - Câu lệnh cập nhật dữ liệu
 - **Câu lệnh thay đổi cấu trúc bảng**
 - **Xóa bảng**



Lệnh sửa bảng

- Được dùng để
 - Thay đổi cấu trúc bảng
 - Thay đổi RBTV

- Thêm cột

```
ALTER TABLE <Tên_bảng> ADD COLUMN  
    <Tên_cột> <Kiểu_dữ_liệu>  
[<RBTV>]
```

- Xóa cột

```
ALTER TABLE <Tên_bảng> DROP COLUMN  
<Tên_cột>
```

- Mở rộng cột

```
ALTER TABLE <Tên_bảng> ALTER COLUMN  
    <Tên_cột> <Kiểu_dữ_liệu_mới>
```

Các lệnh về thay đổi cấu trúc bảng

- Thêm, xóa, sửa một cột (thuộc tính)

- Thêm một cột vào bảng có sẵn

- Cú pháp

```
ALTER TABLE <tên bảng> ADD <tên cột 1>  
                <kd11>[,<tên cột 2> <kd12>, ...]
```

- Ví dụ: Thêm cột ngaysinh vào bảng NhanVien

```
ALTER TABLE NhanVien ADD ngaysinh date
```

Các lệnh về thay đổi cấu trúc bảng

- Thêm, xóa, sửa một cột (thuộc tính) (tt)

- **Chỉnh sửa cột trong bảng**

- **Cú pháp**

ALTER TABLE <table_name>

**ALTER COLUMN <tên cột 1> <kd11>
[, <tên cột 2> <kd12> ...]**

- **Ví dụ: Sửa cột tennv trong bảng NhanVien thành cột có kiểu dữ liệu là nvarchar(20)**

**ALTER TABLE NhanVien ALTER column tennv
nvarchar(20)**

Các lệnh về thay đổi cấu trúc bảng

- Thêm, xóa, sửa một cột (thuộc tính) (tt)

- **Xóa cột trong bảng**

- **Cú pháp**

```
ALTER TABLE <table_name> DROP COLUMN  
<column_name1> [, <column_name2>
```

- **Ví dụ: Xóa cột diadiem trong bảng PhongBan**

```
ALTER TABLE PhongBan DROP COLUMN  
diadiem
```

Thêm ràng buộc khóa chính

- Thêm RB khóa chính

- Cú pháp

```
ALTER TABLE <tên bảng> ADD CONSTRAINT  
<tên RB> PRIMARY KEY (<DS các cột làm  
khóa chính>)
```

- Ví dụ: Thêm khóa chính cho bảng KetQua

```
ALTER TABLE KetQua ADD CONSTRAINT pk_kq  
PRIMARY KEY (masv, mamh)
```

Thêm ràng buộc khóa ngoại

- Thêm RB khóa ngoại

– Cú pháp

```
ALTER TABLE <tên bảng> ADD CONSTRAINT  
<tên RB> FOREIGN KEY (<DS các cột làm  
khóa ngoại>) REFERENCES <tên bảng tham  
chiếu>(<ds cột>)
```

- Ví dụ: Thêm RB khóa ngoại cho bảng KetQua

```
ALTER TABLE KetQua ADD CONSTRAINT  
fk_kq_sv FOREIGN KEY (masv) REFERENCES  
SinhVien(masv)
```


Thêm ràng buộc kiểm tra (check)

- Thêm RB kiểm tra

- Cú pháp

```
ALTER TABLE <tên bảng> ADD CONSTRAINT  
<tên RB> CHECK(<điều kiện>)
```

- Ví dụ: Thêm RB kiểm tra đảm bảo thuộc tính phái trong bảng NhanVien chỉ nhận giá trị “nam” hoặc “nữ”

```
ALTER TABLE NhanVien ADD CONSTRAINT  
ck_phai CHECK(phai in ( 'nam' , 'nu' ) )
```

Xóa ràng buộc khóa chính, khóa ngoại, check

- Cú pháp

```
ALTER TABLE <tên bảng> DROP CONSTRAINT  
                <tên RB>
```

- Ví dụ:

- Xóa RB kiểm tra trong bảng NhanVien

```
ALTER TABLE NhanVien  
DROP CONSTRAINT ck_phai
```

- Xóa RB khóa ngoại trong bảng KetQua

```
ALTER TABLE KetQua  
DROP CONSTRAINT fk_kq_sv
```



Ví dụ - Thay đổi cấu trúc bảng

```
ALTER TABLE NHANVIEN ADD  
    NGHENGHIEP CHAR(20)
```

```
ALTER TABLE NHANVIEN DROP COLUMN  
    NGHENGHIEP
```

```
ALTER TABLE NHANVIEN ALTER COLUMN  
    NGHENGHIEP CHAR(50)
```



Ví dụ - Thay đổi RBTV

```
CREATE TABLE PHONGBAN (  
    TENPB VARCHAR(20),  
    MAPHG INT NOT NULL,  
    TRPHG CHAR(9),  
    NG_NHANCHUC DATETIME  
)
```

```
ALTER TABLE PHONGBAN ADD  
    CONSTRAINT PB_MAPHG_PK PRIMARY KEY (MAPHG),  
    CONSTRAINT PB_TRPHG FOREIGN KEY (TRPHG)  
        REFERENCES NHANVIEN(MANV),  
    CONSTRAINT PB_NGNHANCHUC_DF DEFAULT (GETDATE())  
        FOR (NG_NHANCHUC),  
    CONSTRAINT PB_TENPB_UNI UNIQUE (TENPB)
```

Nội dung chương 3

- Các kiểu dữ liệu trong SQL
- **Câu lệnh định nghĩa dữ liệu**
 - Tạo bảng
 - Câu lệnh cập nhật dữ liệu
 - Câu lệnh thay đổi cấu trúc bảng
 - **Xóa bảng**

Xóa bảng

- **Cú pháp:**
`Drop table <tên bảng>`
- **Ví dụ: Xóa bảng NhanVien**
`Drop table NhanVien`

Xóa bảng (tt)

- Lưu ý khi xóa bảng có liên quan đến khóa ngoại:
 - Nếu không có tham chiếu vòng thì tiến hành xóa bảng chứa khóa ngoại trước sau đó xóa bảng còn lại, hoặc xóa khóa ngoại rồi sau đó tiến hành xóa các bảng.
 - Nếu có tham chiếu vòng thì xóa một khóa để làm mất tham chiếu vòng rồi tiến hành như trường hợp 1.
 - Có thể sử dụng câu lệnh sau làm vô hiệu hóa các RB trên bảng trước khi xóa bảng:

```
Alter      table    <tên    bảng>    nocheck  
constraint all
```



Lệnh tạo miền giá trị

- Tạo ra một kiểu dữ liệu mới kế thừa những kiểu dữ liệu có sẵn
- Cú pháp

```
CREATE DOMAIN <Tên_kdl_mới> AS  
<Kiểu_dữ_liệu>
```

- Ví dụ

```
CREATE DOMAIN Kieu_Ten AS  
VARCHAR(30)
```


Ví dụ tổng hợp

- **Tạo csdl QLDT gồm các lược đồ sau:**
 - **SINHVIEN (MaSV, Hoten, Namsinh, QQ, Hocluoc)**
 - **DETAI (MaDT, TenDT, Chunhiem, Kinhphi)**
 - **SV_DT (MaSV, MaDT, NoiAD, KQ)**
- **Ta có các câu lệnh SQL sau:**

```
Create database QLDT
Use QLDT
Create table SinhVien(
    masv int primary key, hoten nvarchar(30) ,
    namsinh datetime, qq nvarchar(20), hocluoc
float)
```

Ví dụ tổng hợp

```
Create table DeTai(  
    madt      int      primary      key,      tendt  
    nvarchar(30) ,  
    chunhiem nvarchar(30) , kinhphi int)
```

```
Create table SV_DT(  
    masv      int      foreign      key      references  
    SinhVien(masv) ,  
    madt      int      foreign      key      references  
    DeTai(madt) ,  
    noiAD      nvarchar(20) , kq float ,  
    constraint      pk_svdt          primary  
    key(masv , madt) )
```

Bài tập

- Cho lược đồ csdl quản lý hàng hóa như sau:
 - HangHoa(MaHG, TenHG, DVT, Dongia, Cohang)
 - Khach(MaKH, Hoten, Diachi, Daily, sodienthoai)
 - HoaDon(SoHD, Ngaylap, Ngaygiao, Trigia, MaKH)
 - Chitiet_HD(SoHD, MaHG, Soluong, Giaban)
- Thực hiện các công việc sau:
 1. Viết câu lệnh để tạo tất cả các bảng dựa vào các LĐQH trên (có đầy đủ RB khóa chính, khóa ngoại và not null).
 2. Trong bảng Khach, viết câu lệnh sửa cột diachi thành nvarchar(30), giả sử trước đó cột này có kiểu char(20).
 3. Viết câu lệnh kiểm tra RB trên bản Chitiet_HD. (Soluong>0)
 4. Viết câu lệnh không kiểm tra RB trên bảng HoaDon.s
 5. Thêm RB kiểm tra cho thuộc tính cohang trong bang HangHoa là 0 hoặc 1.
 6. Thêm RB duy nhất cho thuộc tính sodienthoai trong bảng Khach.