

TABLE (BẢNG)

Các kiểu dữ liệu

Số Chính Xác - Exact Numerics

☐ Số nguyên

- *Bigint*: (8 bytes) giá trị từ -2^{63} đến $2^{63}-1$
- *Int*: (4 bytes) giá trị từ -2^{31} đến $2^{31} - 1$
- *SmallInt*: (2 bytes) giá trị từ -2^{15} đến $2^{15} - 1$
- *Tinyint*: (1 byte) giá trị từ 0 đến 255

☐ Luận lý

- *Bit*: (1 byte) có giá trị 0, 1

☐ Số thực

- *Decimal*(n, d): $-10^{38} + 1$ đến $10^{38} - 1$; $n \leq 38$; d mặc định bằng 0
- *Numeric*(n, d): giống như decimal.

☐ Tiền tệ

- *SmallMoney*: (4 byte) -214,748.3648 đến 214,748.3647
- *Money*: (8 byte) -922,337,203,685,477.5808 đến 922,337,203,685,477.5807

Số gần đúng - Approximate Numerics

- *Float*: (8 byte) $-1.79E + 308$ đến $1.79E + 308$
- *Float*(n): Nếu $1 \leq n \leq 24$ (4 byte) giá trị $-3.40E + 38$ đến $3.40E + 38$
Nếu $25 \leq n \leq 53$ (8 byte) giá trị từ $-1.79E + 308$ đến $1.79E + 308$
- *Real*: (4 byte) giá trị từ $-3.40E + 38$ đến $3.40E + 38$

Các kiểu dữ liệu

Chuỗi ký tự (không theo Unicode) - Character Strings

- *Char*(n): độ dài cố định, tối đa là 8000 ký tự
- *Varchar*(n): độ dài không cố định, tối đa là 8000 ký tự
- *Varchar*(max): độ dài không cố định, tối đa 2,147,483,647 ký tự
- *Text*: độ dài không cố định, tối đa là $2^{31} - 1$ (2,147,483,647) ký tự, nên sử dụng *Varchar*(max) thay cho kiểu *Text*.

Lưu ý: Bao trong dấu nháy đơn khi thể hiện chuỗi

Chuỗi ký tự Unicode - Unicode Character Strings

- *nChar*(n): độ dài cố định, tối đa là 4000 ký tự.
- *nVarchar*(n): độ dài không cố định, tối đa là 4000 ký tự.
- *nvarchar*(max): độ dài không cố định, tối đa 1,073,741,823 ký tự
- *nText*: độ dài không cố định, tối đa là $2^{30} - 1$ (1,073,741,823) ký tự.

Lưu ý: Bao trong dấu nháy đơn và bắt đầu bằng chữ N khi thể hiện chuỗi

ví dụ: N'khoa công nghệ thông tin'

Các kiểu dữ liệu

Số nhị phân - Binary Strings

- *Binary(n)*: độ dài cố định (tối đa 8000 bytes).
- *Varbinary(n)*: độ dài thay đổi (tối đa 8000 bytes).
- *Image*: độ dài thay đổi, tối đa $2^{31}-1$ bytes (2,147,483,647).

Các kiểu dữ liệu khác

- *Cursor*: kiểu con trỏ
- *Table*: dùng lưu trữ các tập dữ liệu cho lần xử lý sau.
- *Sql_variant*: có thể lưu trữ nhiều kiểu dữ liệu khác nhau trong cùng một bảng như int, char, varchar,... nhưng không chứa dữ liệu kiểu text, ntext, timestamp và sql_variant.
- *XML*: dùng lưu trữ và xử lý dữ liệu dạng XML.
- *Timestamp*: là kiểu dữ liệu có kích thước 8 byte, lưu trữ dạng số nhị phân do hệ thống tự sinh ra, cột khai báo kiểu này sẽ được tự động cập nhật với giá trị phân biệt mỗi khi thêm một mẫu tin mới.
- *Hierarchyid*: là kiểu hệ thống mới có thể lưu giá trị để thể hiện các nút trong kiến trúc cây.

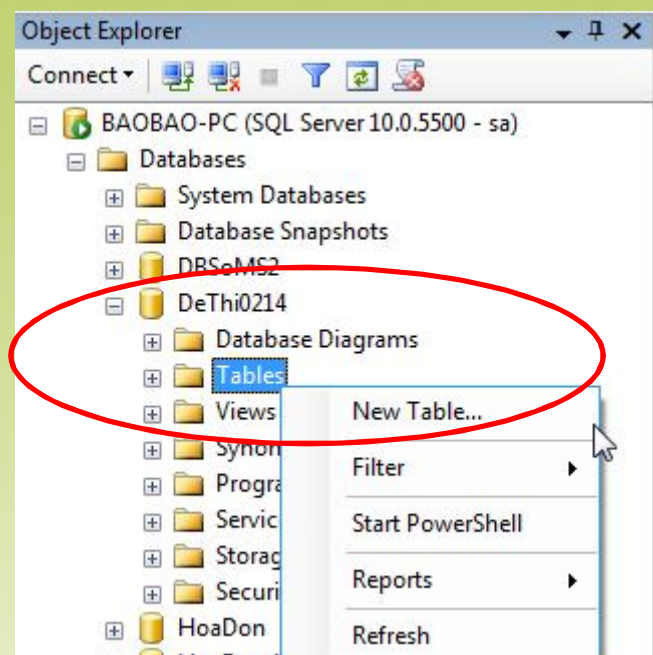
Các kiểu dữ liệu

Ngày giờ

- Date*: YYYY-MM-DD, giá trị từ ‘0001-01-01’ đến ‘9999-12-31’
- Time*: hh:mm:ss[.nnnnnnn], giá trị từ ‘00:00:00.0000000’ đến ‘23:59:59.9999999’
- Datetime*: YYYY-MM-DD, giá trị từ ‘1753-01-01’ đến ‘9999-12-31’
- Datetime2*: YYYY-MM-DD, giá trị từ ‘0001-01-01’ đến ‘9999-12-31’ (có từ SQL 2008)
- Smalldatetime*: YYYY-MM-DD hh:mm:ss, giá trị từ ‘1900-01-01’ đến ‘2079-06-06’

Lưu ý: Bao trong dấu nháy đơn khi thể hiện

Tạo Table – Sử dụng SQL Server Management Studio



BAOBAO-PC.DeThi0214 - dbo.Table_1

Column Name	Data Type	Allow Nulls
		<input type="checkbox"/>

Tạo Table – Sử dụng SQL Server Management Studio

Khai báo tên cột

Cho phép nhận giá trị NULL hay không?

Column Name	Data Type	Allow Nulls
MSDoi	int	<input checked="" type="checkbox"/>
TenDoi	nvarchar(50)	<input type="checkbox"/>
Phai	nvarchar(5)	<input type="checkbox"/>

Column Properties	
(General)	
(Name)	MSDoi
Allow Nulls	No
Data Type	int
Default Value or Binding	
Table Designer	
Collation	<database default>
Computed Column Specification	
Condensed Data Type	int
Description	
Deterministic	Yes
DTS-published	No
Full-text Specification	
Has Non-SQL Server Subscriber	No
Identity Specification	
Indexable	Yes
Is Columnset	No
Is Sparse	No
Merge-published	No
Not For Replication	No

Khai báo kiểu dữ liệu của cột

Các đặc tính của cột

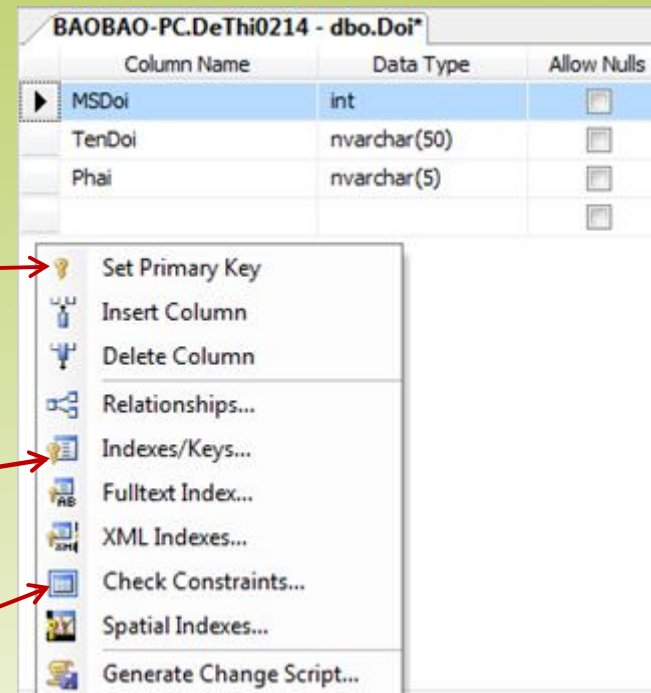
Tạo Table – Sử dụng SQL Server Management Studio

Khai báo một số RBTV

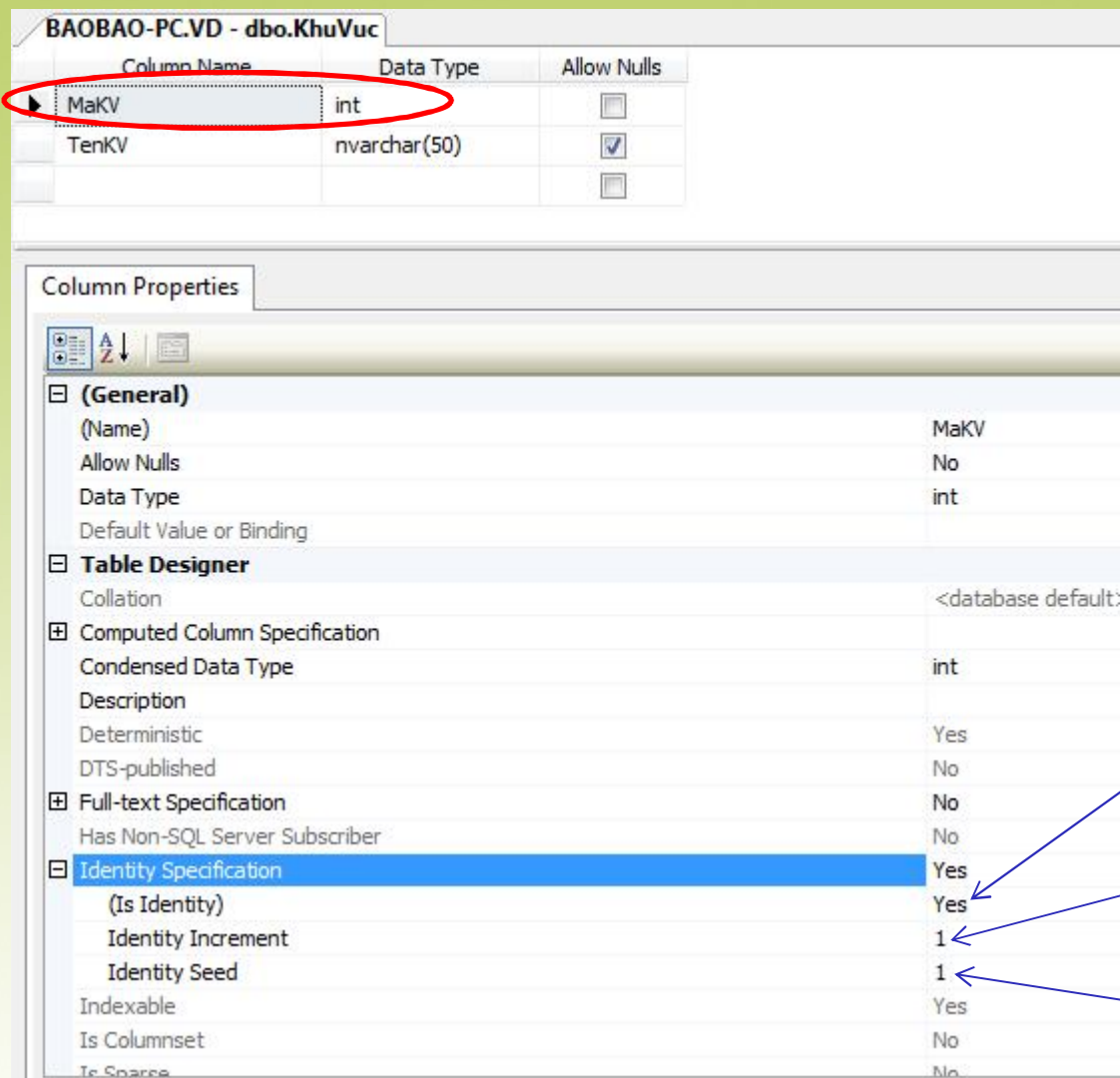
Primary key

Index/Unique Key

Check Constraints



Tạo Table – Sử dụng SQL Server Management Studio



Khai báo cột có giá trị tự động tăng khi thêm mới dữ liệu. (với các cột có kiểu dữ liệu kiểu số nguyên)

(Is Identity): **Yes**

Identity Increment:
Giá trị tăng thêm mỗi lần thêm dòng

Identity Seed: *Giá trị bắt đầu*

Tạo Table – Sử dụng T-SQL Server

❑ Cú pháp:

```
CREATE TABLE tên_bảng  
(  
    Định nghĩa từng cột trong bảng | Tên cột As Biểu thức  
    [Các_ràng_buộc_trên_bảng]  
)
```

Trong đó:

–**Định nghĩa từng cột trong bảng**

• <Tên cột> <kiểu dữ liệu>

• **Tên cột As Biểu thức**: Định nghĩa cột được tính toán dữ liệu từ các cột khác.

–**Các_ràng_buộc_trên_bảng**: cần chú ý một số ràng buộc:

- Not null
- Check
- Unique
- Primary key
- Default
- Foreign key

Khai báo một số Ràng Buộc Toàn Vẹn trên cột

❑ *Nguyên tắc chung:*

- ✓ RBTV có thể liên quan đến một cột, có thể liên quan đến nhiều cột
- ✓ RBTV có thể được khai báo trong lúc tạo quan hệ (CREATE TABLE)
- ✓ RBTV có thể thiết lập bổ sung sau khi các quan hệ đã được tạo (ALTER TABLE)
- ✓ Có thể xây dựng các thủ tục kiểm tra RBTV (Trigger, Procedure,...)
- ✓ Mỗi RBTV có tên tường minh, tên RBTV phải là duy nhất.
- ✓ Việc xóa hay thay đổi một RBTV chỉ được thực hiện khi biết rõ tên của RBTV đó.

Khai báo một số Ràng Buộc Toàn Vẹn (RBTV)

- ❑ **Khai báo RBTV trên một cột:** là những RBTV liên quan đến một cột, sau khi định nghĩa cột, có thể khai báo luôn RBTV cho cột.

✓ Ví dụ:

```
CREATE TABLE NHANVIEN  
(  
    MaNV char(8) NOT NULL PRIMARY KEY,  
    HoNV nvarchar(30),  
    TenNV nvarchar(20) NOT NULL,  
    GioiTinh nvarchar(5) NOT NULL Check(GioiTinh In (N'Nam', N'Nữ'))  
    LCB int Check ( LCB>0)  
)
```

Khai báo một số Ràng Buộc Toàn Vẹn (RBTV)

- ❑ **Khai báo RBTV trên một cột:** là những RBTV liên quan đến một cột, sau khi định nghĩa cột, có thể khai báo luôn RBTV cho cột.

✓ Ví dụ:

```
CREATE TABLE SanPham  
(  
    MaSP Smallint Not Null Primary Key,  
    TenSP varchar(30) Not Null Unique,  
    SoTon Real Default 0 Check (SoTon >=0)  
)
```

Khai báo một số Ràng Buộc Toàn Vẹn (RBTV)

- ❑ **Khai báo RBTV trên một cột:** là những RBTV liên quan đến một cột, sau khi định nghĩa cột, có thể khai báo luôn RBTV cho cột.

✓ Ví dụ:

```
CREATE TABLE SanPham
```

```
(
```

```
    MaSP SmallInt Not Null Constraint pk_MaSP Primary Key,
```

```
    TenSP varchar(30) Constraint uni_TenSP Unique,
```

```
    SoTon Real Constraint df_SoTon Default 0
```

```
)
```

Khai báo một số Ràng Buộc Toàn Vẹn (RBTV)

- ❑ **Khai báo RBTV trên nhiều cột của Table (table constraint):** là các RBTV liên quan đến nhiều cột, hay nhiều table. Khai báo RBTV trong phần riêng, sau khi đã định nghĩa xong Table.

✓ Ví dụ:

```
CREATE TABLE CTHD
```

```
(
```

```
    MaHD Int Not Null,
```

```
    MaSP SmallInt Not Null,
```

```
    SoLuong Int Default 0 Check (SoLuong > 0),
```

```
    Constraint pk_CTHD Primary Key (MaHD, MaSP)
```

```
)
```

Khai báo một số Ràng Buộc Toàn Vẹn (RBTV)

- ❑ **Khai báo RBTV trên nhiều cột của Table (table constraint):** là các RBTV liên quan đến nhiều cột, hay nhiều table. Khai báo RBTV trong phần riêng, sau khi đã định nghĩa xong tất cả các cột.

✓ Ví dụ:

```
CREATE TABLE HOADON  
(  
    MaHD Char(10) NOT NULL PRIMARY KEY,  
    NgayLapHD Date NOT NULL,  
    NgayGiao Date,  
    MaKH char(5) NOT NULL,  
    Check (NgayGiao>=NgayLapHD)  
)
```


Khai báo một số Ràng Buộc Toàn Vẹn (RBTV)

- ❑ **Khai báo RBTV trên nhiều cột của Table (table constraint):** Trường hợp cột là biểu thức (cột ảo): dữ liệu lấy từ những cột khác thông qua biểu thức.

✓ Ví dụ:

```
CREATE TABLE CTHD
```

```
(
```

```
    MaHD int NOT NULL,
```

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

```
    SL int NOT NULL,
```

```
    DonGia int,
```

```
    ThanhTien As SL*DonGia,
```

```
    PRIMARY KEY (MaHD,MaMH)
```

```
)
```

BAOBAO-PC.VD - dbo.CTHD					
	MaHD	MaMH	SL	DonGia	ThanhTien
	1	BN	20	18000	360000
	2	GK	50	65000	3250000
	3	BI	30	380000	11400000
➤*	NULL	NULL	NULL	NULL	NULL



Khai báo một số Ràng Buộc Toàn Vẹn (RBTV)

❑ **Cú pháp:** Cột có giá trị tự tăng **Identity**

IDENTITY(seed, Increment): Để chỉ định cột có giá trị tự động tăng cho dữ liệu kiểu số nguyên.

- Seed: giá trị khởi đầu
- Increment: giá trị tăng mỗi khi thêm dữ liệu mới.

1
2
3
4
5
...

Identity(1,1)

1
4
7
10
13
...

Identity(1,3)

1
3
5
7
9
...

Identity(1,2)

Cột có giá trị tự tăng **Identity**

✓ Ví dụ:

```
CREATE TABLE SanPham
```

```
(
```

```
    MaSP int NOT NULL IDENTITY(1,1),
```

```
    TenSP nvarchar(50) NOT NULL,
```

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

```
    DonGia int,
```

```
    Check (DonGia>0),
```

```
    PRIMARY KEY (MaSP)
```

```
)
```

Khai báo một số Ràng Buộc Toàn Vẹn (RBTV)

- Khai báo ràng buộc tham chiếu (khóa ngoại)

MAKHOA	TENKHOA
CK	Cơ khí
CNTT	Khoa CNTT
KH	Khoa khoa học
KT	Khoa Kinh tế

MANGANH	TENNGANH	MAKHOA
CK	Cơ khí NL	CK
HTT	Hệ thống thông tin	CNTT
KT	Kế toán	KT
MMT	Mạng máy tính	CNTT
QT	Quản trị KD	KT
TD	Tự động hóa	CK

Đại số quan hệ

MAKHOA
CK
CNTT
KH
KT

$$= \text{KHOA}[\text{MAKHOA}] \supseteq \text{NGANH}[\text{MAKHOA}] =$$

MAKHOA
CK
CNTT
KT
CNTT
KT
CK

Hay: $\pi_{\text{MAKHOA}}(\text{KHOA}) \supseteq \pi_{\text{MAKHOA}}(\text{NGANH})$

Khai báo một số Ràng Buộc Toàn Vẹn (RBTV)

❑ Khai báo ràng buộc tham chiếu (khóa ngoại)

... FOREIGN KEY (< khóa ngoại K>

REFERENCES <table chứa K - khóa chính>[(khóa chính - K)]

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

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

Khai báo một số Ràng Buộc Toàn Vẹn (RBTV)

- ✓ Ví dụ: Khai báo ràng buộc tham chiếu (khóa ngoại)

```
CREATE TABLE NGANH
```

```
(
```

```
    MANGANH char(8) PRIMARY KEY,  
    TENGANH NVARCHAR(50)
```

```
)
```

```
GO
```

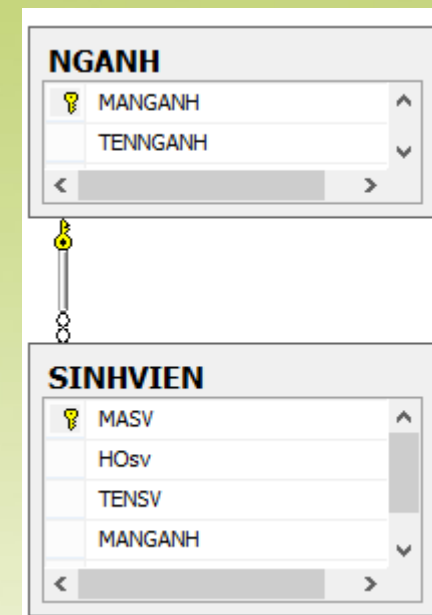
```
CREATE TABLE SINHVIEN
```

```
(
```

```
    MASV CHAR(8) PRIMARY KEY,  
    HOSV NVARCHAR(30),  
    TENSU NVARCHAR(20) NOT NULL,  
    MANGANH char(8),  
    FOREIGN KEY (MANGANH) REFERENCES NGANH
```

```
)
```

```
GO
```



Khai báo một số Ràng Buộc Toàn Vẹn (RBTV)

- ❑ Định nghĩa ràng buộc tham chiếu kèm với mệnh đề **ON DELETE CASCADE**

MAKHOA	TENKHOA
CK	Cơ khí
CNTT	Khoa CNTT
KH	Khoa khoa học
KT	Khoa Kinh tế

MANGANH	TENNGANH	MAKHOA
CK	Cơ khí NL	CK
HTT	Hệ thống thông tin	CNTT
KT	Kế toán	KT
MMT	Mạng máy tính	CNTT
QT	Quản trị KD	KT
TD	Tự động hóa	CK

CREATE TABLE SINHVIEN

(

MASV CHAR(8) PRIMARY KEY,

HOSV NVARCHAR(30),

TENSV NVARCHAR(20) NOT NULL,

MANGANH CHAR(8),

FOREIGN KEY (MANGANH) REFERENCES NGANH ON DELETE CASCADE

)

GO

Khai báo một số Ràng Buộc Toàn Vẹn (RBTV)

- ❑ Định nghĩa ràng buộc tham chiếu kèm với mệnh đề **ON DELETE NO ACTION**



MAKHOA	TENKHOA
CK	Cơ khí
CNTT	Khoa CNTT
KH	Khoa khoa học
KT	Khoa Kinh tế

MANGANH	TENNGANH	MAKHOA
CK	Cơ khí NL	CK
HTT	Hệ thống thông tin	CNTT
KT	Kế toán	KT
MMT	Mạng máy tính	CNTT
QT	Quản trị KD	KT
TD	Tự động hóa	CK

CREATE TABLE SINHVIEN

(

MASV CHAR(8) PRIMARY KEY,
HOSV NVARCHAR(30),
TENSX NVARCHAR(20) NOT NULL,
MANGANH CHAR(8),

FOREIGN KEY (MANGANH) REFERENCES NGANH ON DELETE NO ACTION

)

GO

Khai báo một số Ràng Buộc Toàn Vẹn (RBTV)

- ❑ Định nghĩa ràng buộc tham chiếu kèm với mệnh đề **ON DELETE SET NULL**

MAKHOA	TENKHOA
CK	Cơ khí
CNTT	Khoa CNTT
KH	Khoa khoa học
KT	Khoa Kinh tế

MANGANH	TENNGANH	MAKHOA
CK	Cơ khí NL	CK
HTT	Hệ thống thông tin	null
KT	Kế toán	KT
MMT	Mạng máy tính	null
QT	Quản trị KD	KT
TD	Tự động hóa	CK

CREATE TABLE SINHVIEN

(

MASV CHAR(8) PRIMARY KEY,

HOSV NVARCHAR(30),

TENSV NVARCHAR(20) NOT NULL,

MANGANH CHAR(8),

FOREIGN KEY (MANGANH) REFERENCES NGANH ON DELETE SET NULL

)

GO

Khai báo một số Ràng Buộc Toàn Vẹn (RBTV)

- ❑ Định nghĩa ràng buộc tham chiếu kèm với mệnh đề **ON UPDATE ...** Tương tự như định nghĩa ràng buộc tham chiếu với mệnh đề **ON DELETE....**

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



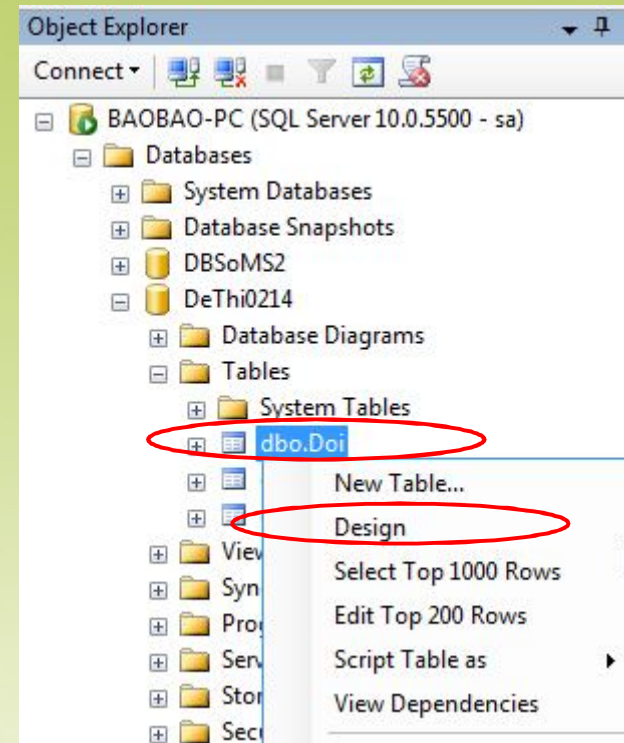
Thay đổi cấu trúc bảng

Thay đổi cấu trúc Table - Sử dụng SQL Server Management Studio

❑ Mở Object Explorer,

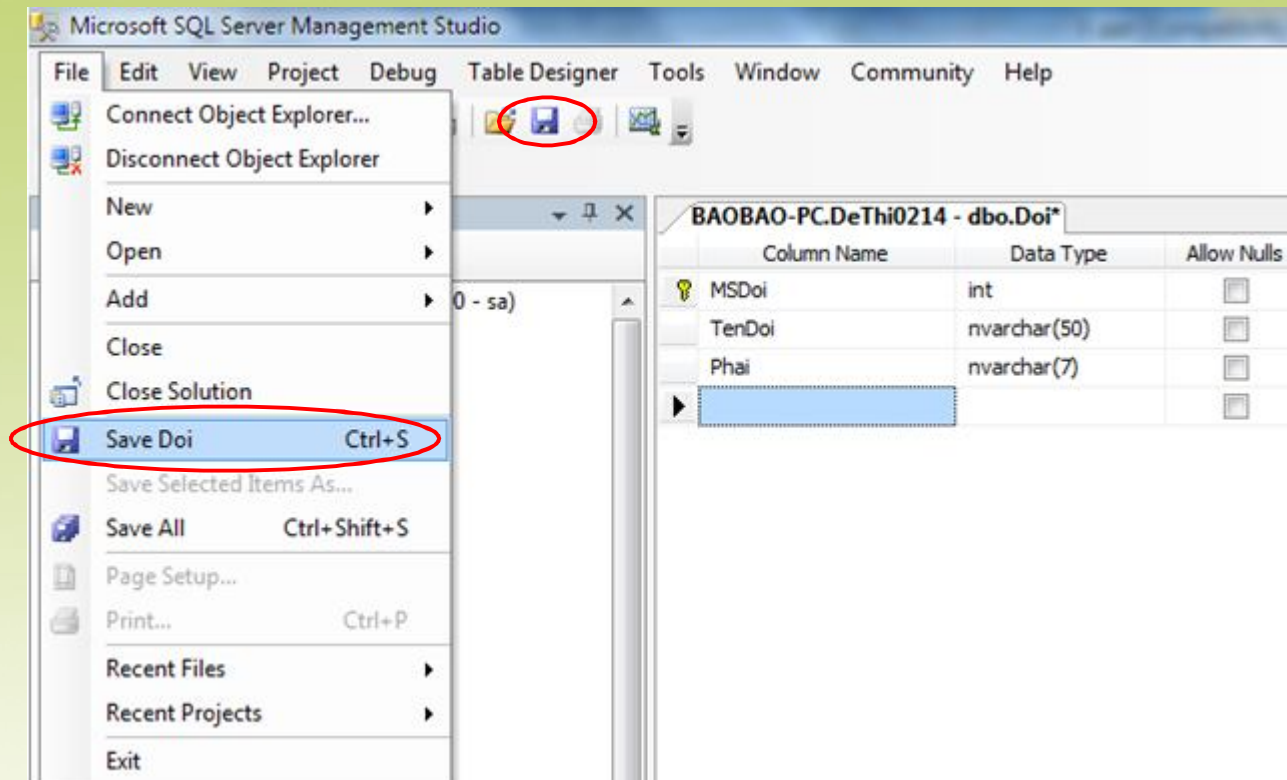
Trong Database, chọn
table cần thay đổi cấu
trúc

Right click → Design, sau
đó tiến hành sửa đổi
như lúc tạo.



Thay đổi cấu trúc Table - Sử dụng SQL Server Management Studio

- ❑ Sau khi thực hiện xong, phải chọn lưu các thay đổi.



Thay đổi cấu trúc Table - Sử dụng T - SQL

❑ Thêm cột

```
ALTER TABLE <table_name> ADD { <column_definition> |  
<column_name> AS <expression> |  
<table_constraint> } [ ,...n ]
```

❑ Xóa cột

```
ALTER TABLE <table_name> DROP COLUMN <column_name>
```

❑ Thay đổi định nghĩa cột

```
ALTER TABLE <table_name> ALTER COLUMN <column_name>  
<type_name> [ NULL | NOT NULL ]
```

RBTV được thiết lập bổ sung sau khi đã tạo quan hệ

- ❑ **Bổ sung RBTV:** Có thể bổ sung RBTV sau khi đã tạo lược đồ quan hệ bằng lệnh ALTER TABLE

```
ALTER TABLE R ADD [CONSTRAINT [<tên rbtv1>]] <rbtv1>,  
CONSTRAINT [<tên rbtv2>] <rbtv2>,...
```

- ❑ **Xóa RBTV**

```
ALTER TABLE <table_name> DROP CONSTRAINT <constraint_name>
```

Thay đổi cấu trúc Table - Sử dụng T - SQL

- ✓ Ví dụ: Giả sử đã tạo lược đồ quan hệ **KetQua**(MaSV, MaMH, Diem) như sau:

Create Table KetQua

```
(  
    MaSV char(10) Not Null,  
    MaMH char(5) Not Null,  
    Diem int Not Null,  
)
```

- ✓ Bổ sung thêm RBTV: $0 \leq \text{Diem} \leq 10$, dùng lệnh ALTER TABLE

Alter Table KetQua **Add Constraint** chk_diem **Check** (Diem \geq 0 **And** Diem \leq 10)

- ~~✗~~ **Lưu ý:** trường hợp thêm RBTV **Default** <value> cho một cột

ALTER TABLE <table_name> **ADD CONSTRAINT** {<constraint_name>}

DEFAULT <value> **FOR** <column_name> [**WITH VALUES**]

Ví dụ:

ALTER TABLE SanPham **ADD CONSTRAINT** df_soton **DEFAULT** 0
FOR SoTon **WITH VALUES**

Một số lưu ý

- ✂ Khi thay đổi kiểu dữ liệu của cột (thay đổi kiểu dữ liệu, kích thước)

`ALTER TABLE table_name ALTER COLUMN col_name <các thay đổi>`

Ví dụ: `Alter Table SanPham Alter Column TenSP nvarchar(30) NULL`

- ❑ Lưu ý, khi cột đã có dữ liệu hoặc có ràng buộc thì không thể:

- Giảm kích thước của cột nhỏ hơn kích thước dữ liệu hiện có.
- Thay đổi kiểu dữ liệu của cột
- Thay đổi kiểu của cột đã dùng trong cột tính toán.
- Từ NULL thành NOT NULL.

Một số lưu ý

✎ Khi thêm cột có ràng buộc **NOT NULL** vào bảng đã có dữ liệu:

❑ Cách 1: thực hiện qua 3 bước:

- Thêm cột với thuộc tính NULL,
- Điền đầy đủ các giá trị cho cột,
- Đổi lại thuộc tính của cột thành NOT NULL.

❑ Cách 2: Sử dụng **WITH VALUES**.

```
ALTER TABLE SanPham ADD SoTon Int NOT NULL  
CONSTRAINT df_soton DEFAULT 0 WITH VALUES
```

Một số lưu ý

✂ Khi xóa cột

35

ALTER TABLE table_name **DROP COLUMN** col_name [...]

Ví dụ: **ALTER TABLE** NhanVien **DROP COLUMN** NoiSinh

❑ *Lưu ý: Xóa ràng buộc hoặc ngừng ràng buộc trước khi xóa cột có ràng buộc CHECK, DEFAULT, FOREIGN KEY, UNIQUE, PRIMARY KEY...*

Một số lưu ý

- ✂ Sử dụng, tạm dừng ràng buộc

```
ALTER TABLE table_name {CHECK | NOCHECK} CONSTRAINT {ALL |  
constraint_name}
```

- ❑ Ví dụ, tạm ngưng kiểm tra ràng buộc.

```
ALTER TABLE SanPham NOCHECK CONSTRAINT chk_DonGia
```

- ❑ Ví dụ, kích hoạt kiểm tra constraint.

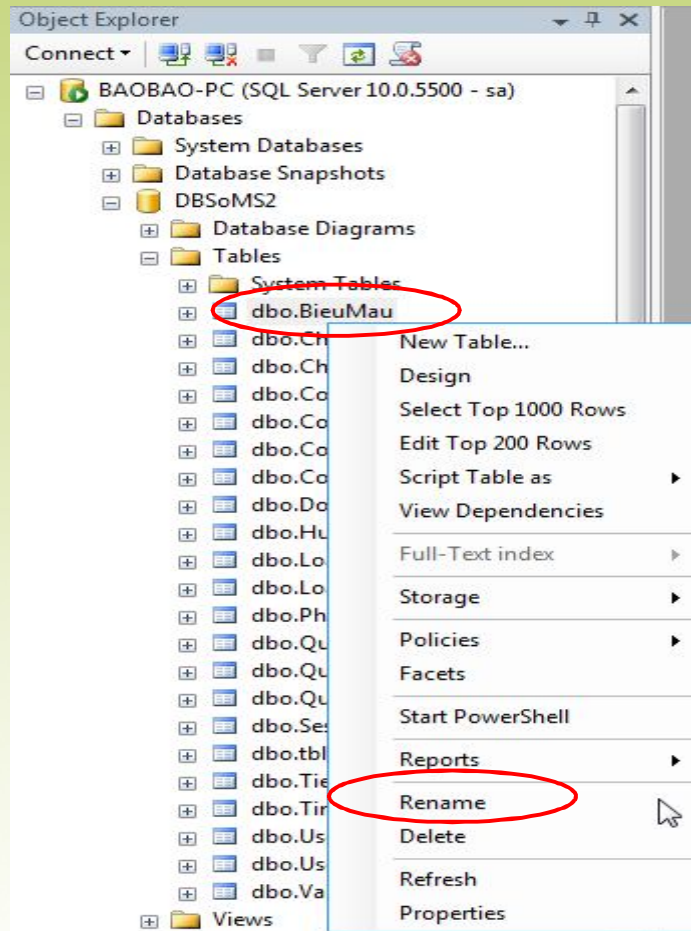
```
ALTER TABLE SanPham CHECK CONSTRAINT chk_DonGia
```

Đổi tên Table

❑ Đổi tên Table

Sử dụng:

SQL Server Management Studio



Sử dụng:

T-SQL

Exec **sp_Rename** 'old_tb_name','new_tb_name'

✓ Ví dụ: Đổi tên table SanPham thành SP

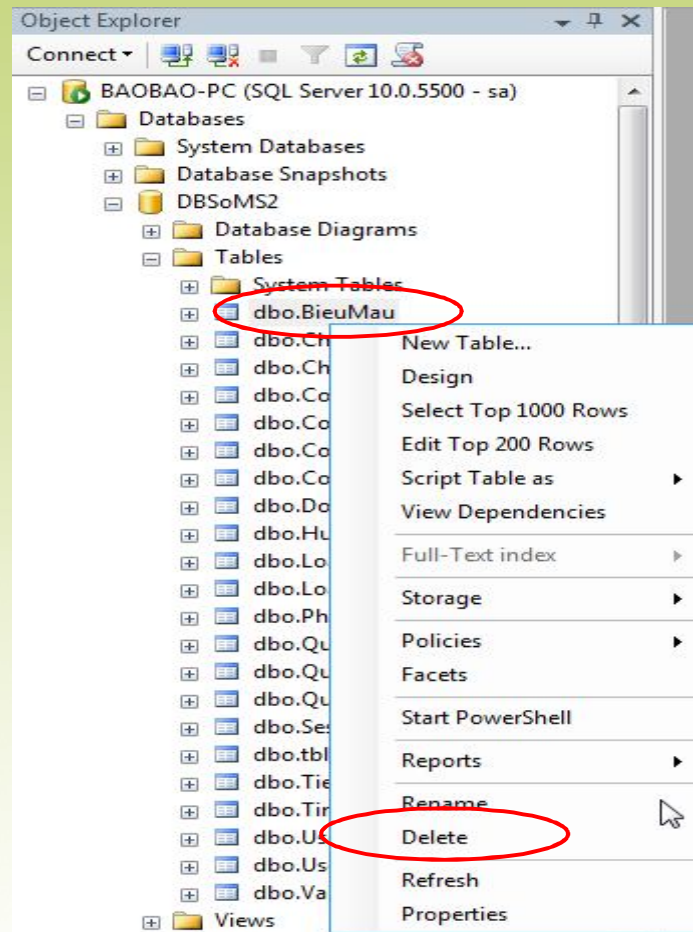
Exec **sp_Rename** 'SanPham','SP'

Xóa Table

❑ Xóa Table

Sử dụng:

SQL Server Management Studio



Sử dụng:

T-SQL

DROP TABLE <table_name>

- ✓ Ví dụ: Xóa table NhanVien
Use QLBanHang
Go
Drop Table 'NhanVien'


Lệnh thêm (Insert) dòng vào Table

❑ Dạng 1:

INSERT INTO <table> (A_1, A_2, \dots, A_n) **VALUES** (v_1, v_2, \dots, v_n)

Trong đó: v_i tương ứng là giá trị trên cột A_i của dòng được thêm vào.

SinhVien

	MASV	nchar(8)	
	HOSV	nchar(30)	
	TENS	nchar(20)	
	GIOITINH	nchar(10)	
	NAMSINH	int	

Insert Into SinhVien (MaSV, HoSV, TenSV, GioiTinh, NamSinh)
Values ('101011', **N**'Trần Gia', **N**'Tuệ', **N**'Nữ', 1996)


Lệnh thêm (Insert) dòng vào Table

□ Dạng 2:

INSERT INTO <table> VALUES (v_1, v_2, \dots, v_m)

Trong đó: Ngầm định, thứ tự các cột trong Table phải tương ứng với các giá trị cần thêm vào.

SinhVien

	MASV	nchar(8)	
	HOSV	nchar(30)	
	TENSV	nchar(20)	
	GIOITINH	nchar(10)	
	NAMSINH	int	

Insert Into SinhVien Values ('101011', N'Trần Gia', N'Tuệ', N'Nữ', 1996)

Lệnh thêm (Insert) dòng vào Table

❑ Dạng 3:

INSERT INTO <table> (A₁, A₂, ..., A_n) <lệnh **Select** sql>

Trong đó:

- Chèn vào <table> các dòng lấy từ kết quả của một câu lệnh Select SQL
- <lệnh Select SQL> phải trả về kết quả gồm các cột tương ứng các cột được liệt kê

✓ *Ví dụ:* Giả sử ta đã có quan hệ: **SV_TH**(MaSV, HoSV, TenSV, PhaiNu, NgaySinh)
Lấy danh sách sinh viên ngành tin học (TH) từ quan hệ **SinhVien** và thêm vào quan hệ **SV_TH**

Lệnh SQL:

Insert Into SV_TH(MaSV, HoSV, TenSV, PhaiNu, NgaySinh) **Select** MaSV, HoSV, TenSV, PhaiNu, NgaySinh **From** SinhVien **Where** MaNganh = 'TH'

Lệnh thêm (Insert) dòng vào Table

❑ Dạng 4:

INSERT INTO <table> <lệnh Select sql>

Trong đó:

- Chèn vào <table> các dòng lấy từ kết quả của một câu lệnh Select SQL
- <lệnh Select SQL> phải trả về kết quả gồm các cột tương ứng các cột được liệt kê

✓ Ví dụ: Giả sử ta đã có quan hệ: **SV_TH**(MaSV, HoSV, TenSV, PhaiNu, NgaySinh)
Lấy danh sách sinh viên ngành tin học (TH) từ quan hệ **SinhVien** và thêm vào quan hệ **SV_TH**

Lệnh SQL:

Insert Into **SV_TH** Select MaSV, HoSV, TenSV, PhaiNu, NgaySinh From **SinhVien**
Where MaNganh = 'TH'

Lệnh thêm (Insert) dòng vào Table

- ❑ **Lưu ý:** Database Engine tự động cung cấp giá trị cho các cột
 - Có thuộc tính IDENTITY
 - Có khai báo Default.
 - Có cho phép Null.
 - Cột tính toán

Lệnh sửa (Update) dữ liệu trên dòng của Table

❑ Cú pháp:

```
UPDATE <table> SET A1=<new value 1>, A2=<new value 2>,...  
[FROM <tables_source>]  
[WHERE <điều kiện>]
```

Trong đó:

- <table>: Tên bảng cần cập nhật dữ liệu
- A_i=<new value i>: Thay giá trị trên cột A_i bởi giá trị <new value i>
- <điều kiện>] : Chỉ những bộ thoả điều kiện mới được cập nhật.

Lưu ý:

- Nếu bỏ qua phát biểu *Where* <điều kiện> thì tất cả các dòng sẽ chịu ảnh hưởng.
- Trường hợp khi điều kiện cập nhật dữ liệu có liên quan đến nhiều bảng, mệnh đề *FROM* được sử dụng.

Lệnh sửa (Update) dữ liệu trên dòng của Table

- ✓ Ví dụ: Sửa lại điểm thi lần 1 môn SQL của sinh viên có mã số 101011 thành 10

```
Update KetQua Set Diem = 10 Where MaSV='101011' And MaMH='SQL'  
And LANHI = 1
```

Lệnh xóa (Delete) các dòng dữ liệu trên Table

❑ Cú pháp 1:

DELETE FROM <table_name> [**WHERE** <điều kiện>]

Lưu ý: Xóa các dòng trong < table_name> thỏa <điều kiện>, nếu bỏ qua phát biểu **WHERE** <điều kiện> thì sẽ xóa tất cả các dòng trong < table_name>

✓ Ví dụ: **Delete From** HoaDon **Where** Year(NgayLapHD) = 1992

❑ Cú pháp 2:

DELETE <table_name> **FROM** <Join Table> [**WHERE** <điều kiện>]

✓ Ví dụ: Xóa các CTHD của các hóa đơn đã bán trong ngày hôm nay.

Delete CTHD **From** CTHD, HoaDon **Where** CTHD.MaHD = HoaDon.MaHD

And NgayLHD = **Getdate()**

Lưu ý: Biến hệ thống @@ROWCOUNT

- ❑ Biến @@ROWCOUNT: Trả về số dòng bị ảnh hưởng bởi câu lệnh cập nhật dữ liệu gần nhất.
- ✓ Ví dụ: cập nhật dữ liệu với UPDATE và dùng @@ROWCOUNT để xác định số dòng đã được thay đổi.

Create Proc CapNhatDVT_SP

@masp int

As

UPDATE SanPham Set DVT = 'Kg' WHERE MaSP = @masp

IF @@ROWCOUNT = 0

Print 'Không có dòng nào được cập nhật!'

Go

Lưu ý: Biến hệ thống @@ERROR

❑ Biến @@ERROR chứa mã lỗi cho phát biểu vừa thực hiện. Nó tự động xóa và đặt lại mã lỗi với mỗi phát biểu được thực hiện. Giá trị 0 nếu phát biểu thành công. Có thể dùng biến @@ERROR để xác định lỗi hoặc làm điều kiện kết thúc thủ tục.

✓ Ví dụ: Create Proc ThemCTHD

@mahd char(5),

@masp int,

@sl int,

@dongia int

As

Insert Into CTHD(MaHD, MaSP, SL, DonGia) Values (@mahd,
@masp, @sl, @dongia)

If (@@Error <> 0)

Begin

Print (N'Lỗi thêm chi tiết hóa đơn')

Return 0

End

Go