

FPT POLYTECHNIC



Bài 2:
TẠO CƠ SỞ DỮ LIỆU QUAN HỆ
& NGÔN NGỮ TRUY VẤN T-SQL

- Tổng quan về hệ quản trị CSDL SQL Server
- Các công cụ SQL Server
 - SQL Server Configuration Manager
 - SQL Server Management Studio
 - SQL Books Online
- Làm quen với công cụ SQL Management Studio

1. Kiểu dữ liệu

2. Tạo CSDL quan hệ

3. Làm quen với T-SQL

KIỂU DỮ LIỆU

- SQL Server định nghĩa 26 *kiểu dữ liệu*, được chia làm bốn nhóm

Nhóm	Mô tả
Chuỗi	Các chuỗi dữ liệu ký tự .
Số	Các số nguyên, số có dấu phẩy động, tiền tệ và các kiểu dữ liệu số khác.
Thời gian (ngày/giờ)	Ngày, giờ, hoặc cả ngày và giờ.
Kiểu khác	Các giá trị chuỗi và nhị phân lớn, XML, dữ liệu hình học, dữ liệu địa lý, dữ liệu phân cấp.

■ Dùng để lưu các ký tự

	Kiểu	Byte	Mô tả
Lưu ký tự thường	char[(n)]	n	<ul style="list-style-type: none"> - Chuỗi độ dài cố định. - n là số ký tự, nằm giữa 1 và 8.000. Mặc định là 1.
	varchar[(n)]	n	<ul style="list-style-type: none"> - Chuỗi độ dài biến đổi. - Số lượng byte được sử dụng để lưu trữ chuỗi phụ thuộc vào độ dài thực của chuỗi. - n là số lượng ký tự lớn nhất, nằm giữa 1 và 8.000. Mặc định là 1.
Lưu ký tự UNICODE	nchar(n)	2n	<ul style="list-style-type: none"> - Chuỗi độ dài cố định chứa dữ liệu ký tự Unicode. - n là số lượng ký tự, nằm giữa 1 và 4.000. Mặc định là 1.
	nvarchar(n)	2n	<ul style="list-style-type: none"> - Chuỗi độ dài biến đổi chứa dữ liệu ký tự Unicode. - Số lượng byte được sử dụng để lưu trữ chuỗi phụ thuộc vào độ dài thực của chuỗi. Hai byte cần thiết để lưu trữ mỗi ký tự. - n là số lượng ký tự lớn nhất, nằm giữa 1 và 4.000. Mặc định là 1.

	Kiểu	Byte	Mô tả
Số nguyên	bigint	8	Số nguyên lớn từ -9.223.372.036.854.775.808 tới 9.223.372.036.854.775.807.
	int	4	Số nguyên từ -2.147.483.648 tới 2.147.483.647.
	smallint	2	Số nguyên nhỏ từ -32.768 tới 32.767.
	tinyint	1	Số nguyên dương rất nhỏ, từ 0 tới 255.
	bit	1	Số nguyên với các giá trị 1 hoặc 0.
Số thập phân	decimal[(p[,s])]	5-17	Số thập phân với độ chính xác (p) và tỷ lệ (s) cố định từ $-10^{38} + 1$ tới $10^{38} - 1$.
	numeric[(p[,s])]	5-17	Đồng nghĩa với decimal.
	money	8	Giá trị tiền tệ với bốn vị trí thập phân từ -922.337.203.685.477,5808 tới 922.337.203.685.477,5807. Đồng nghĩa với decimal(19,4).
	smallmoney	4	Giá trị tiền tệ với bốn vị trí thập phân từ -214.748,3648 tới 214.748,3647. Đồng nghĩa với decimal(10,4).
Số thực	float[(n)]	4 hoặc 8	Số dấu phẩy động độ chính xác kép từ $-1,79 \times 10^{308}$ tới $1,79 \times 10^{308}$.
	real	4	Số dấu phẩy động độ chính xác đơn, từ $-3,4 \times 10^{38}$ tới $3,4 \times 10^{38}$. Đồng nghĩa với float(24).

■ Kiểu dữ liệu ngày/giờ trong phiên bản trước SQL Server 2008

Kiểu	Byte	Mô tả
datetime	8	Ngày và giờ từ 1/1/1753 tới 31/12/9999, với độ chính xác 3,33 mili giây.
smalldatetime	4	Ngày và giờ từ 1/1/1900 tới 6/6/2079, với độ chính xác một phút.

■ Kiểu dữ liệu ngày/giờ trong SQL Server 2008

Kiểu	Byte	Mô tả
date	3	Chỉ có ngày (không chứa phần giờ) từ 1/1/0001 tới 31/12/9999.
time(n)	3-5	Chỉ có giờ (không chứa phần ngày) từ 00:00:00,0000000 tới 23:59:59,9999999, với độ chính xác 0,0000001 giây; n là số từ 0 tới 7 được sử dụng cho độ chính xác phân đoạn giây.
datetime2(n)	6-8	Ngày từ 1/1/0001 tới 31/12/9999, với giá trị thời gian từ 00:00:00,0000000 tới 23:59:59,9999999.
datetimeoffset(n)	8-10	Mở rộng của kiểu datetime2, thêm múi giờ (time zone) được biểu diễn bởi giá trị từ -14 tới +14.

Các kiểu dữ liệu giá trị lớn

Kiểu	Mô tả	Kiểu dữ liệu sử dụng trong phiên bản trước SQL 2005 tương ứng
varchar(max)	<ul style="list-style-type: none"> - Làm việc tương tự như kiểu varchar - Nhưng có thể lưu trữ tới 2.147.483.648 byte dữ liệu. 	text
nvarchar(max)	<ul style="list-style-type: none"> - Làm việc tương tự như kiểu nvarchar - Có thể lưu trữ tới 2.147.483.648 byte dữ liệu. 	ntext
varbinary(max)	<ul style="list-style-type: none"> - Lưu trữ dữ liệu nhị phân độ dài biến đổi tối đa tới 2.147.483.648 byte. - Số lượng byte được sử dụng để lưu trữ phụ thuộc vào độ dài thực sự của dữ liệu. 	image

■ Kiểu dữ liệu chuỗi:

- Là kiểu dữ liệu thường được sử dụng nhất.
- Kiểu **char** và **nchar** dùng để lưu trữ chuỗi có độ dài cố định. Ví dụ: Mã sinh viên
- Kiểu **varchar**, **nvarchar** dùng để lưu trữ chuỗi có độ dài thay đổi.
- Kiểu **nchar**, **nvarchar** dùng để lưu trữ kí tự Unicode. Hệ thống hỗ trợ đa ngôn ngữ nên sử dụng kiểu này.

■ Kiểu dữ liệu số

- Dùng lưu trữ các dữ liệu số, sử dụng trong tính toán
- Trong một số trường hợp thường sử dụng kiểu chuỗi làm kiểu dữ liệu thay thế khi lưu trữ dữ liệu số như: số điện thoại, số chứng minh, mã ZipCode

■ Kiểu dữ liệu thời gian

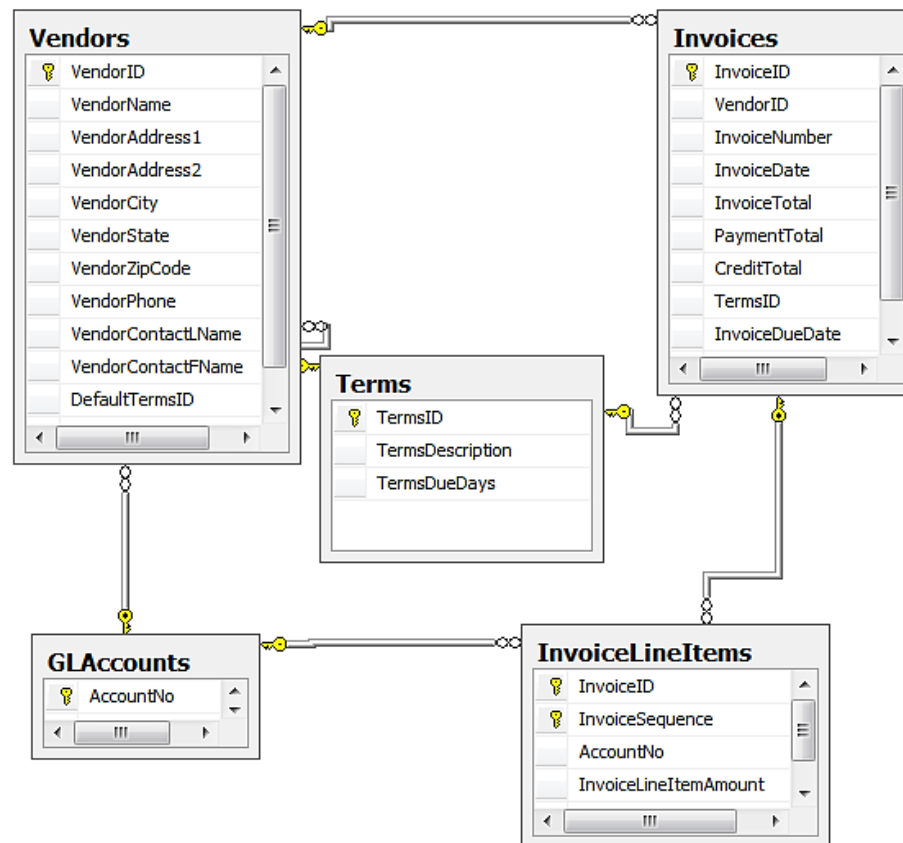
- Phiên bản SQL Server 2008 hỗ trợ cả những kiểu dữ liệu thời gian của các phiên bản trước



- Các kiểu dữ liệu giá trị lớn
 - Sử dụng để lưu ảnh cùng các dạng ký tự cũng như dữ liệu nhị phân lớn.
 - Nên tránh sử dụng các kiểu dữ liệu text, ntext và image (được sử dụng trước SQL Server 2005)

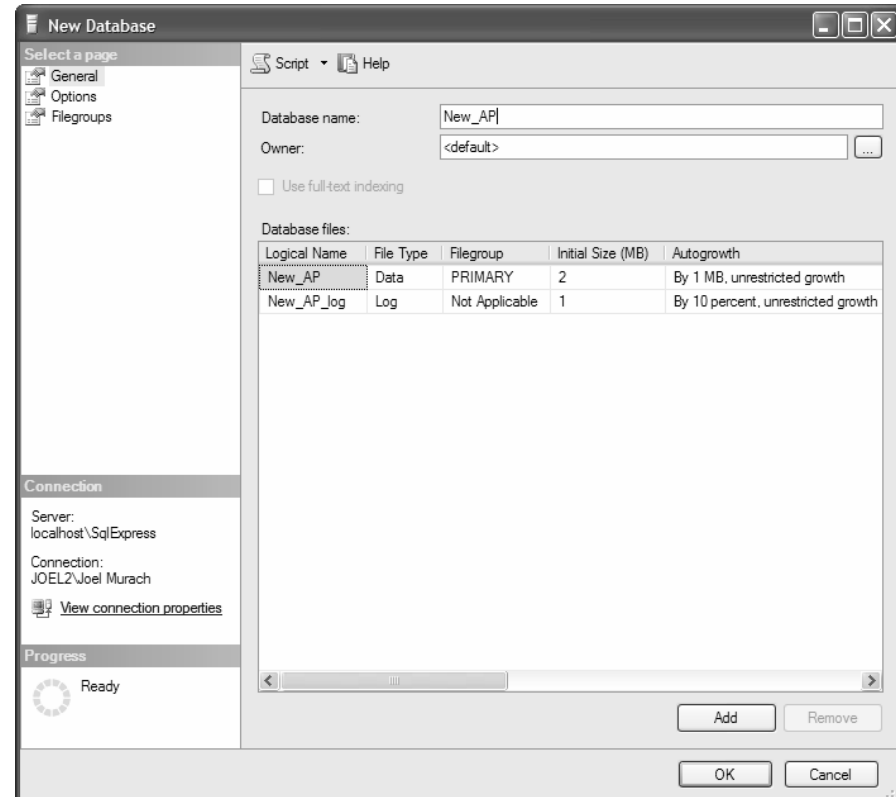
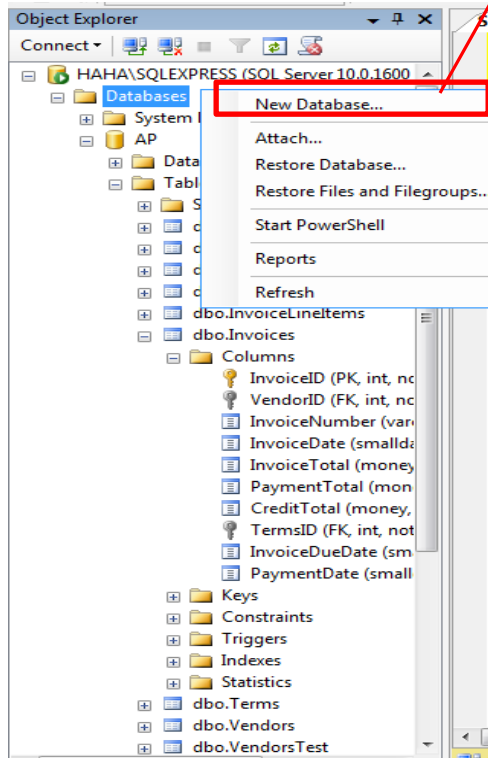
- SQL Server cho phép sử dụng các từ đồng nghĩa với kiểu dữ liệu chuẩn ANSI thay thế cho kiểu dữ liệu của SQL Server.
 - Khi sử dụng từ đồng nghĩa, SQL Server tự động chuyển về kiểu dữ liệu SQL Server tương ứng
 - Tuy nhiên không nên sử dụng từ đồng nghĩa, trừ từ đồng nghĩa Rowversion
 - Tham khảo phần **Kiểu dữ liệu chuẩn ANSI và kiểu tương ứng trong SQL Server**

- Khám phá cách sử dụng các kiểu dữ liệu của các bảng trong cơ sở dữ liệu mẫu **AP**.



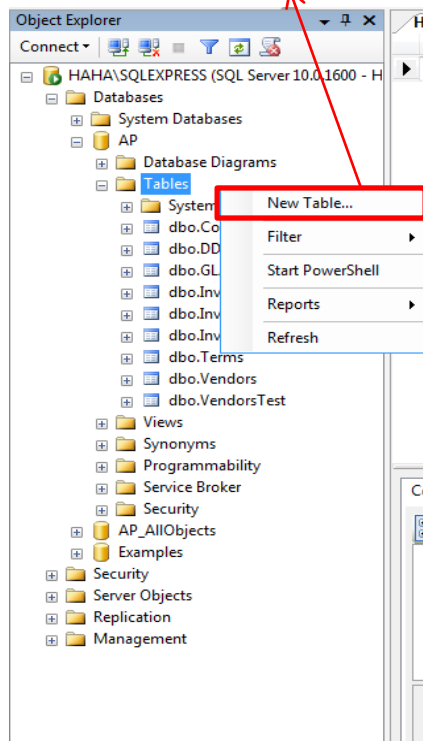
TẠO CSDL QUAN HỆ

Tạo CSDL mới

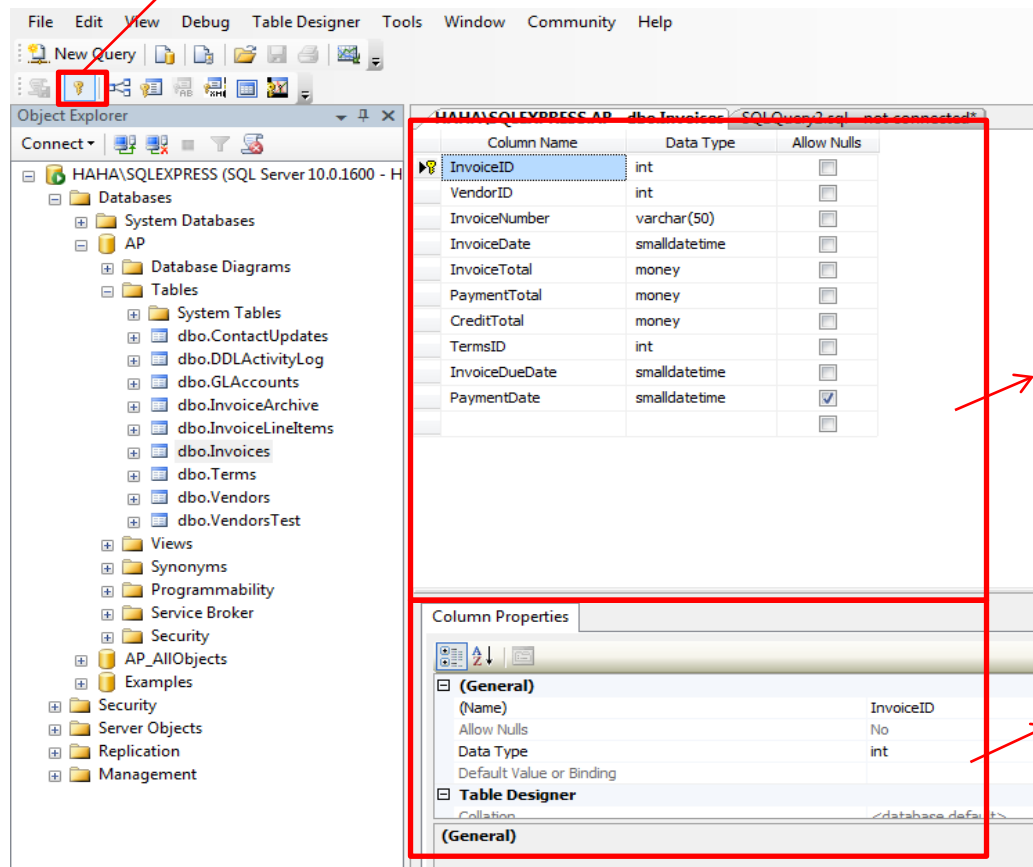


Hướng dẫn tạo bảng

Tạo bảng mới



Thiết lập khóa chính cho cột được chọn

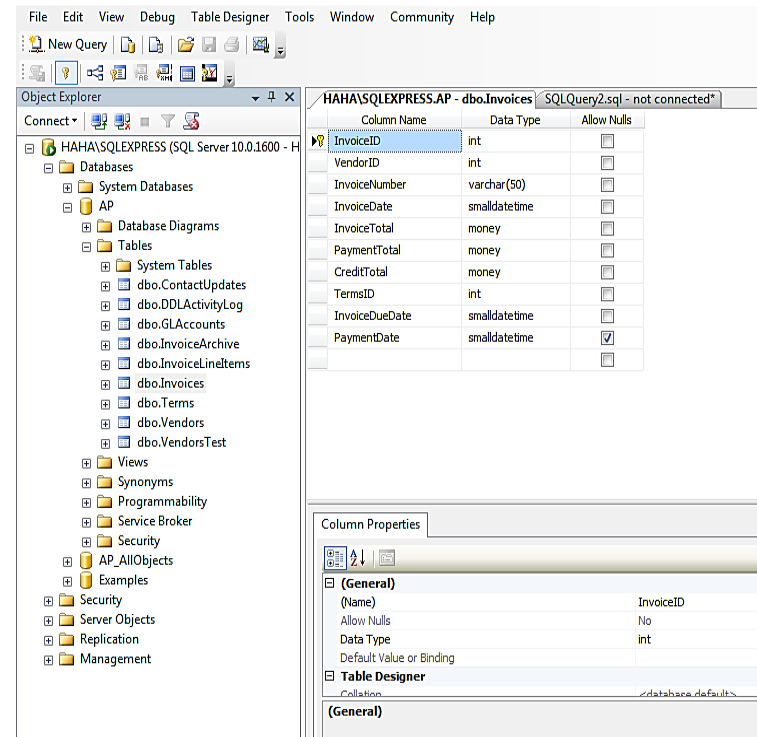
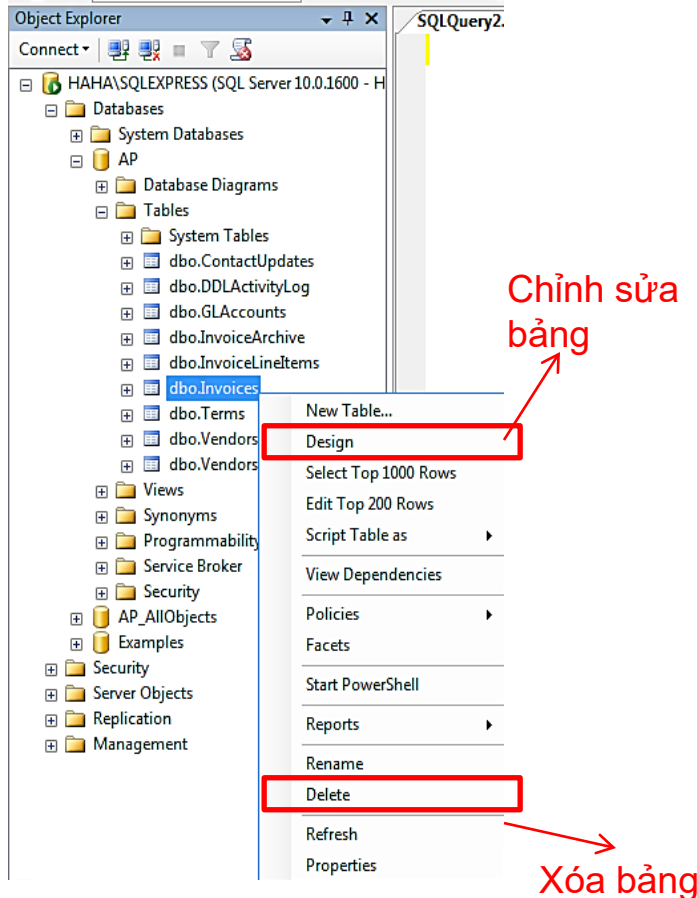


Nhập tên các cột, kiểu dữ liệu, chấp nhận thuộc tính NULL tại đây

Thay đổi thuộc tính cho cột tại đây

Xóa và chỉnh sửa bảng

- Chỉnh sửa bảng: Thay đổi tên cột, thuộc tính cột rồi chọn nút Save .



Các hướng dẫn làm việc với bảng khác

- Thanh công cụ làm việc với bảng.
- Chỉ hiển thị khi ở chế độ Table Design

Nút Manage Indexes and Keys:

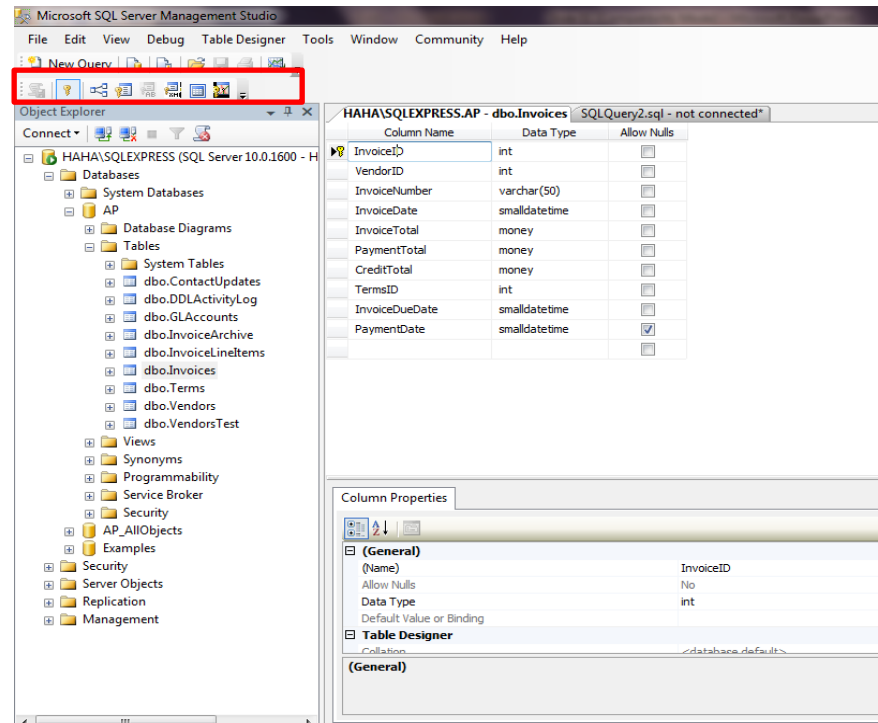
Làm việc với Index và khóa



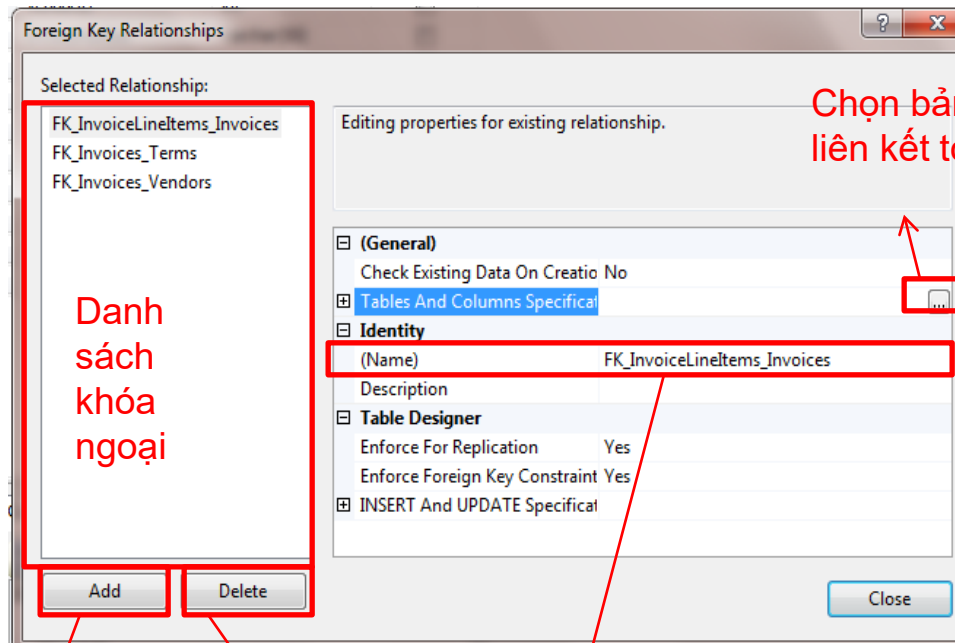
Thiết lập khóa chính

Nút Relationships :
Làm việc với khóa ngoại

Nút Manage Check Constrains:
Kiểm tra ràng buộc cho cột



- Click vào button **Relationships** trên thanh công cụ



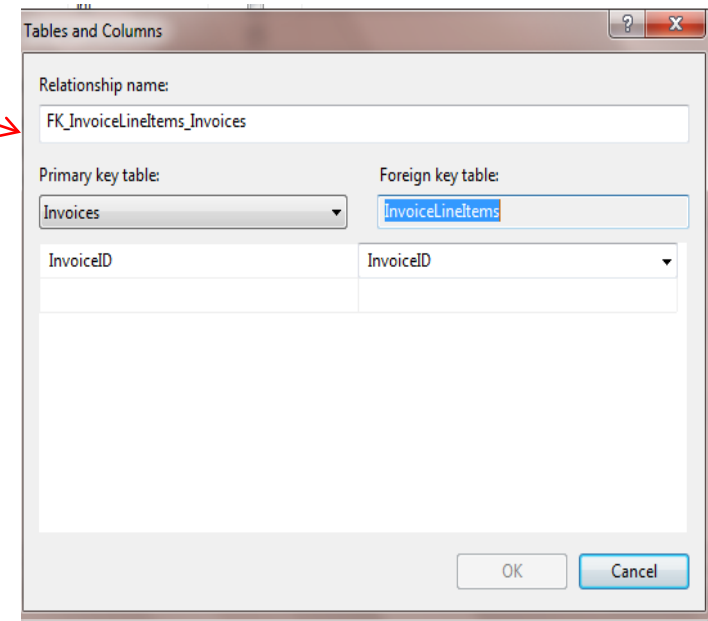
Chọn bảng liên kết tới

Danh sách khóa ngoại

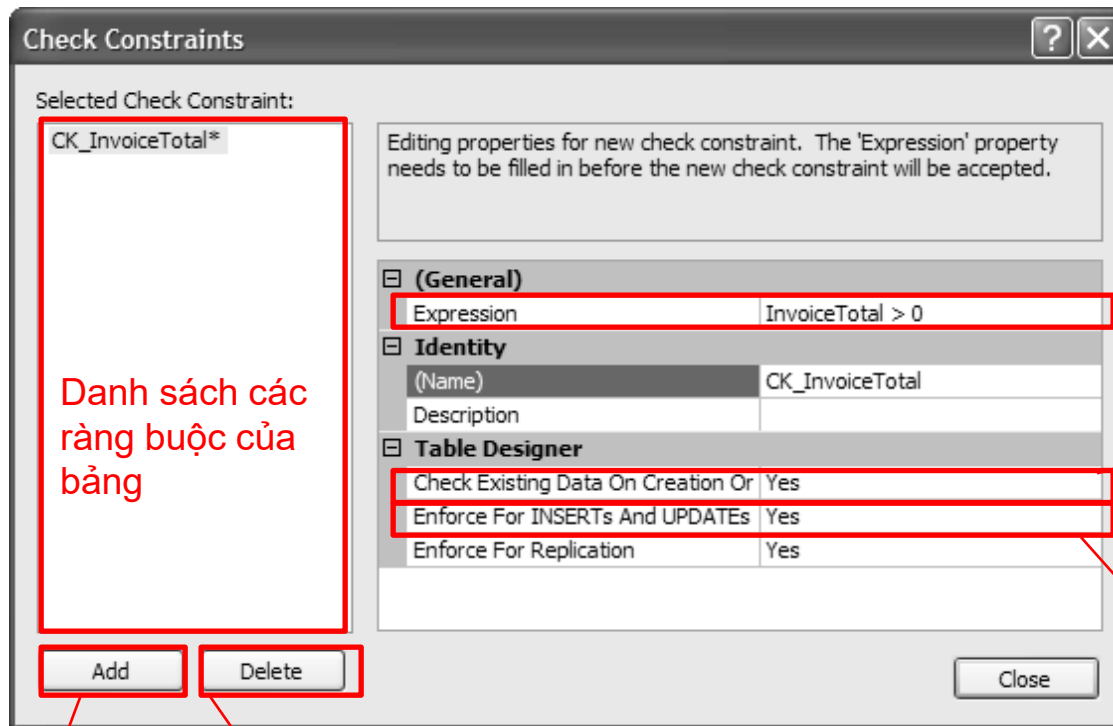
Thêm khóa ngoại mới

Xóa khóa ngoại được chọn

Chỉ định tên cho khóa ngoại



- Click vào nút **Manage Check Constraints** trên thanh công cụ



Danh sách các
ràng buộc của
bảng

Nhập biểu thức
ràng buộc cho
cột

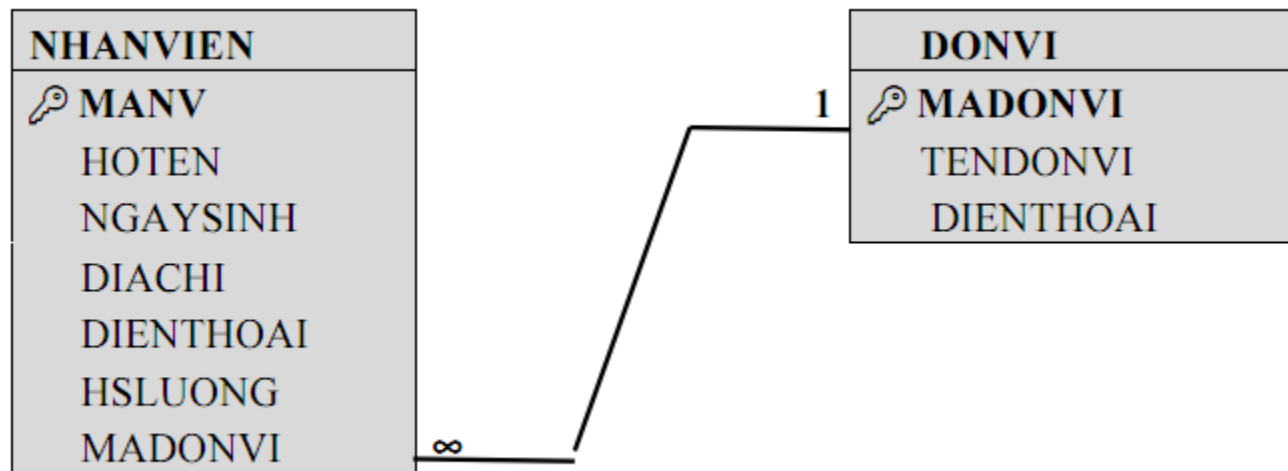
Đặt là **No** để
SQL Server
không kiểm tra
dữ liệu đang tồn
tại với điều kiện
ràng buộc

Đặt là **No** để
SQL Server
không ép buộc
thao tác INSERT
và UPDATE

Thêm một
ràng buộc

Xóa ràng buộc
được chọn trong
danh sách

- Demo tạo CSDL Quản lý nhân viên (Case study “Hướng dẫn sử dụng Management Studio để thiết kế CSDL”)
 - Chọn kiểu dữ liệu phù hợp cho các cột
 - Tạo bảng & quan hệ giữa các bảng



LÀM QUEN VỚI T-SQL

- **Nội dung đã học trong môn CSDL**
 - Ngôn ngữ định nghĩa dữ liệu (DDL)
 - Ngôn ngữ thao tác dữ liệu (DML)
 - Ngôn ngữ điều khiển dữ liệu (DCL)

- Các tính năng lập trình mới
 - Biến
 - Các cấu trúc điều khiển (IF...ELSE, WHILE, CASE...WHEN)
 - Chú thích
 - Định danh (Identifier): Là các đối tượng CSDL như
 - Bảng
 - View, Stored Procedure... (Học trong các bài sau)
 - Hàm
 - Hàm thư viện (Học trong các bài sau)
 - Hàm người dùng định nghĩa (Học trong các bài sau)
- Chú ý: T-SQL không phân biệt chữ hoa, chữ thường

Đã học

**Sẽ học
trong
các
phần
sau**

Câu lệnh	Mô tả
CREATE DATABASE	Tạo mới một CSDL
CREATE TABLE	Tạo mới một bảng trong CSDL hiện tại
CREATE INDEX	Tạo một index mới cho bảng được chỉ định trước
CREATE FUNCTION	Tạo hàm người dùng định nghĩa
CREATE PROCEDURE	Tạo một Stored Procedure (Thủ tục được lưu trữ)
CREATE TRIGGER	Tạo mới một trigger
CREATE VIEW	Tạo mới một view

Đã học



**Sẽ học
trong
các
phần
sau**



Câu lệnh	Mô tả
ALTER TABLE	Sửa thiết kế bảng
ALTER FUCTION	Sửa hàm người dùng định nghĩa
ALTER TRIGGER	Sửa trigger
ALTER VIEW	Sửa view
ALTER PROCEDURE	Sửa Stored Procedure

Đã học

**Sẽ học
trong
các
phần
sau**

Câu lệnh	Mô tả
DROP DATABASE	Xóa một CSDL
DROP INDEX	Xóa một index
DROP FUNCTION	Xóa một function
DROP PROCEDURE	Xóa một procedure
DROP TRIGGER	Xóa một trigger
DROP VIEW	Xóa một view

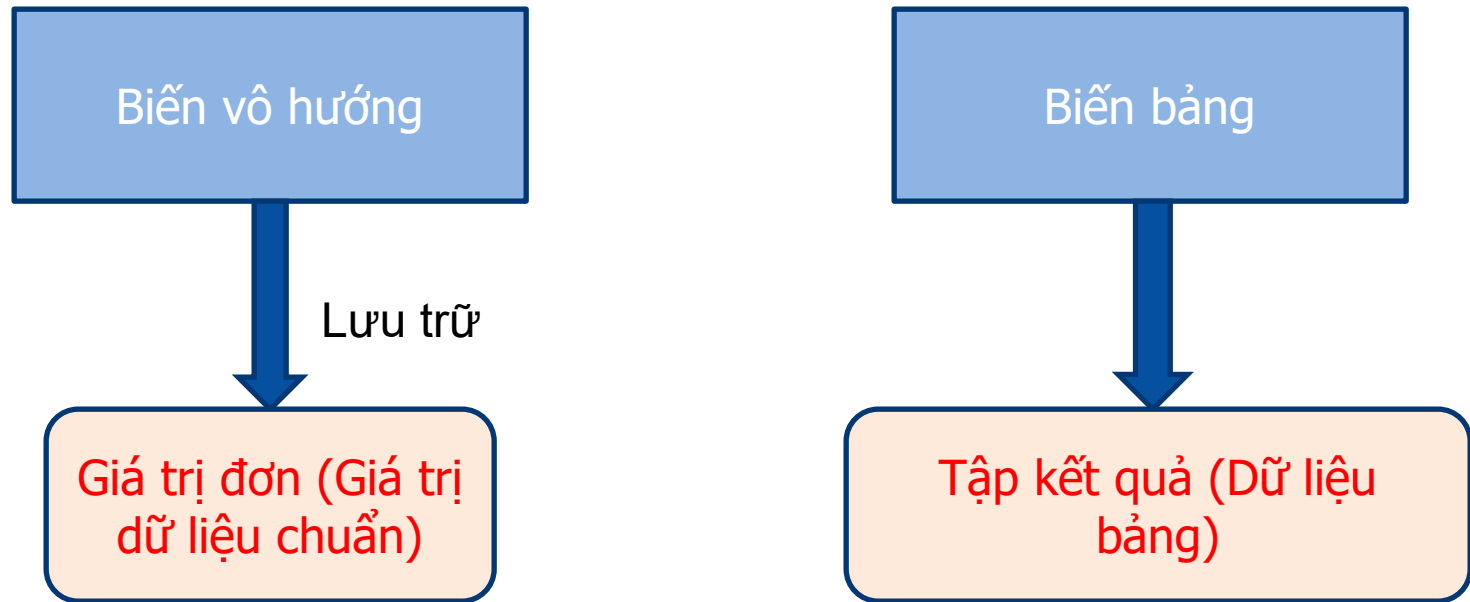
Giống các
câu lệnh
SQL đã
học trong
môn
CSDL

Câu lệnh	Mô tả
SELECT	Truy xuất dữ liệu từ một hoặc nhiều bảng
INSERT	Thêm một hàng dữ liệu vào bảng
UPDATE	Cập nhật dữ liệu
DELETE	Xóa dữ liệu

- Sinh viên ôn tập lại các câu lệnh
 - CREATE DATABASE
 - CREATE TABLE
 - CREATE INDEX
 - ALTER TABLE
 - SELECT
 - INSERT
 - UPDATE
 - DELETE

- Cú pháp chú thích một dòng đơn
 - Viết chú thích một dòng đơn ở đây
- Cú pháp chú thích nhiều dòng
 - /*
 - Chú thích nhiều dòng
 - */

■ Hai loại biến





■ Khai báo

- Cú pháp

DECLARE @<tên biến 1> <kiểu dữ liệu> [, @<tên biến 2> <kiểu dữ liệu>]...

■ Gán giá trị

- Cú pháp của câu lệnh SET

SET @<tên biến> = <biểu thức>

- Sử dụng để gán giá trị cho một biến

- Cú pháp câu lệnh SELECT

SELECT @<tên biến> = <biểu thức cột>

- SELECT có thể dùng để gán giá trị cho 2 hoặc nhiều biến

■ Truy xuất giá trị biến:

- **SELECT @<tên biến>**

■ Ví dụ 1: Gán giá trị cho biến sử dụng câu lệnh SET

-- Khai báo biến

```
DECLARE    @VendorIDVar int
           @MaxInvoice money,
           @MinInvoice money,
```

-- Giá trị ID của nhà cung cấp

-- Tổng số tiền hóa đơn thanh toán lớn nhất

-- Tổng số tiền hóa đơn thanh toán nhỏ nhất

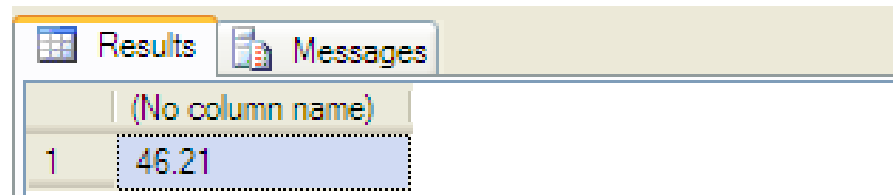
-- Gán giá trị cho biến @VendorIDVar

```
SET       @VendorIDVar = 95
```

-- Gán giá trị cho biến @MaxInvoice bằng tổng số tiền hóa đơn lớn nhất tương ứng với nhà cung cấp @VendorIDVar

```
SET       @MaxInvoice = (SELECT MAX(InvoiceTotal) FROM Invoices WHERE
                        VendorID = @VendorIDVar)
```

SELECT @MaxInvoice -- Truy xuất giá trị biến



Results	
	(No column name)
1	46.21

■ Ví dụ 2: Gán giá trị cho biến sử dụng câu lệnh SET

-- Gán giá trị cho các biến được khai báo ở ví dụ 1 sử dụng câu lệnh SELECT

```
SELECT @MaxInvoice = MAX(InvoiceTotal), @MinInvoice = MIN(InvoiceTotal)
FROM Invoices
WHERE VendorID = @VendorIDVar
```

SELECT @MaxInvoice, @MinInvoice -- Truy xuất giá trị biến

	(No column name)	(No column name)
1	46.21	16.33

■ Cú pháp khai báo biến bảng

DECLARE @<tên bảng> TABLE

**(<tên cột 1> <kiểu dữ liệu> [<Các thuộc tính cột>]
[, <tên cột 2> <kiểu dữ liệu> [<Các thuộc tính cột>]]...
[, <Các thuộc tính bảng>])**

Phân
định
nghĩa
cột &
các
thuộc
tính
bảng

- Phần định nghĩa cột và các thuộc tính bảng
 - Có cú pháp giống như câu lệnh định nghĩa một bảng mới (**CREATE TABLE**)
- Biển bảng được sử dụng giống như bảng chuẩn SQL
 - Có thể tham chiếu đến biển bảng trong câu lệnh **SELECT, INSERT, UPDATE, DELETE** như bảng chuẩn
 - **Ngoại lệ:** Không thể sử dụng câu lệnh **SELECT INTO** với biển bảng

-- Khai báo biến bảng @BigVendors lưu dữ liệu VendorID & VendorName của các nhà cung cấp

```
DECLARE @BigVendors TABLE
        (VendorID int,
         VendorName varchar(50))
```

-- Chèn dữ liệu vào biến bảng

```
INSERT @BigVendors
SELECT VendorID, VendorName
FROM Vendors
WHERE VendorID IN (SELECT VendorID FROM Invoices WHERE InvoiceTotal > 5000)
```

SELECT * FROM @BigVendors -- Truy xuất dữ liệu từ biến bảng

	VendorID	VendorName
1	72	Data Reproductions Corp
2	99	Bertelsmann Industry Svcs. Inc
3	104	Digital Dreamworks
4	110	Malloy Lithographing Inc

■ Cú pháp

IF <biểu thức điều kiện>
{<Câu lệnh> | BEGIN...END}
[ELSE
{<Câu lệnh> | BEGIN...END}]

■ Chú ý:

- Nếu thực thi hai hoặc nhiều câu lệnh trong mệnh đề IF hoặc ELSE: Bạn cần bao các câu lệnh này trong khối **BEGIN...END**

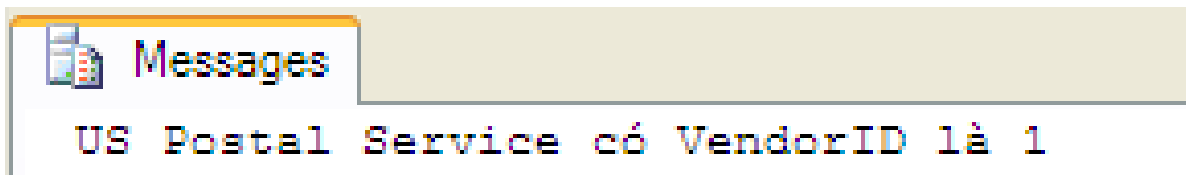
Sử dụng câu lệnh IF...ELSE

```

DECLARE      @vendorID int
SELECT  @vendorID = ISNULL(VendorID,0)
FROM VENDORS
WHERE VendorName ='US Postal Service'

IF @vendorID = 0
    PRINT 'Không tồn tại nhà cung cấp US Postal Service'
ELSE
    PRINT 'US Postal Service có VendorID là ' + CAST(@vendorID AS
VARCHAR)

```



■ Cú pháp

WHILE <biểu thức điều kiện>
{<câu lệnh> | BEGIN...END}
[BREAK]
[CONTINUE]

■ Chú ý:

- Nếu thực thi hai hoặc nhiều câu lệnh trong mệnh đề WHILE: Bạn cần bao các câu lệnh này trong khối BEGIN...END

- Ví dụ: sử dụng vòng lặp WHILE để tính tổng các số từ 1 đến 10

```
DECLARE @a int ,  
          @Tổng int  
  
SET @a = 1  
SET @Tổng = 0  
  
WHILE (@a <= 10)  
BEGIN  
    SELECT @Tổng = @Tổng + @a  
    SET @a = @a + 1  
END  
  
SELECT @Tổng
```

- Trả về giá trị dựa vào điều kiện được chỉ định.
- Có 2 loại hàm CASE

● 1. Cú pháp hàm CASE đơn giản

CASE <biểu thức>

WHEN <biểu thức 1> THEN <biểu thức kết quả 1>

[WHEN <biểu thức 2> THEN <biểu thức kết quả 2>]...

[ELSE <biểu thức kết quả mệnh đề else>]

END

Ví dụ:

```
SELECT InvoiceNumber, TermsID,
CASE TermsID
WHEN 1 THEN 'Net due 10 days'
WHEN 2 THEN 'Net due 20 days'
WHEN 3 THEN 'Net due 30 days'
WHEN 4 THEN 'Net due 60 days'
WHEN 5 THEN 'Net due 90 days'
END AS Terms
FROM Invoices
```

	InvoiceNumber	TermsID	Terms
6	963253261	3	Net due 30 days
7	963253237	3	Net due 30 days
8	125520-1	1	Net due 10 days

• 2. Cú pháp hàm tìm kiếm CASE

CASE

```
WHEN <biểu thức điều kiện 1> THEN <biểu thức kết quả 1>
[WHEN <biểu thức điều kiện 1> THEN <biểu thức kết quả 1>]...
[ELSE <biểu thức kết quả mệnh đề else>]
```

END

Ví dụ:

```
SELECT InvoiceNumber, InvoiceTotal, InvoiceDate, InvoiceDueDate,
CASE
    WHEN DATEDIFF(day, InvoiceDueDate, GETDATE()) > 30
        THEN 'Over 30 days past due'
    WHEN DATEDIFF(day, InvoiceDueDate, GETDATE()) > 0
        THEN '1 to 30 days past due'
    ELSE 'Current'
END AS Status
FROM Invoices
WHERE InvoiceTotal - PaymentTotal - CreditTotal > 0
```

	InvoiceNumber	InvoiceTotal	InvoiceDate	InvoiceDueDate	Status
9	134116	90.36	2008-07-28 00:00:00	2008-08-17 00:00:00	1 to 30 days past due
10	0-2436	10976.06	2008-07-31 00:00:00	2008-08-30 00:00:00	Current
11	547480102	224.00	2008-08-01 00:00:00	2008-08-31 00:00:00	Current

■ Các kiểu dữ liệu

- SQL hỗ trợ 26 kiểu dữ liệu cơ bản, được chia thành 4 nhóm
 - Kiểu dữ liệu chuỗi
 - Kiểu dữ liệu số
 - Kiểu dữ liệu thời gian
 - Kiểu dữ liệu giá trị lớn

- Hai cách để tạo CSDL quan hệ trên SQL Server:
 - Sử dụng các câu lệnh định nghĩa dữ liệu (DDL)
 - Sử dụng công cụ Management Studio
- T-SQL giống với ngôn ngữ SQL chuẩn đã học, gồm có:
 - Ngôn ngữ định nghĩa dữ liệu (DDL)
 - Ngôn ngữ thao tác dữ liệu (DML)
 - Ngôn ngữ điều khiển dữ liệu (DCL)

■ T-SQL cung cấp các tiện ích lập trình

● Biến

- Biến vô hướng: Lưu trữ giá trị đơn (các kiểu dữ liệu chuẩn)
- Biến bảng: Lưu trữ tập kết quả từ câu lệnh truy vấn

● Các cấu trúc điều khiển (IF...ELSE, WHILE, CASE...WHEN)

● Chú thích

● Định danh (Identifier): Là các đối tượng CSDL như

- Bảng
- View, Stored Procedure... (Học trong các bài sau)

● Hàm

- Hàm thư viện (Học trong các bài sau)
- Hàm người dùng định nghĩa (Học trong các bài sau)

XIN CẢM ƠN!