

# BÀI THỰC HÀNH SỐ 1 (4 tiết)

## ÔN TẬP CÁC KỸ NĂNG THIẾT KẾ CƠ SỞ DỮ LIỆU

### I. Mục tiêu

Bài thực hành này giúp sinh viên củng cố và nắm vững kiến thức về các câu lệnh truy vấn SQL. Ngoài ra, sau khi hoàn thành bài học, sinh viên sẽ tích lũy cho mình những kinh nghiệm trong việc phân tích, thiết kế một cơ sở dữ liệu.

Sau bài thực hành này, sinh viên cần nắm rõ những vấn đề sau:

- Các lệnh truy vấn SQL: SELECT, INSERT, UPDATE, DELETE
- Cách tạo các giao dịch để gom nhóm các lệnh SQL
- Cách sử dụng chương trình Enterprise Manager tích hợp trong bộ SQL Server 2000 để thiết kế một cơ sở dữ liệu.
- Cách sử dụng chương trình Query Analyzer trong bộ SQL Server để viết và thực thi các lệnh truy vấn.
- Cách tạo các ràng buộc (Constraints), chỉ mục (Index), khung nhìn (View), Trigger, hàm (Functions) và thủ tục lưu trữ (Stored Procedure)

### II. Nội dung lý thuyết

#### 1. Các lệnh truy vấn SQL

**Câu lệnh SELECT** được sử dụng để truy xuất dữ liệu từ các dòng và các cột của một hay nhiều bảng, khung nhìn. Các thành phần trong câu lệnh SELECT nếu được sử dụng phải tuân theo đúng thứ tự như trong cú pháp. Nếu không, câu lệnh sẽ được xem là không hợp lệ.

Cú pháp chung của câu lệnh SELECT có dạng:

```
SELECT [ALL | DISTINCT] [TOP n] danh_sách_chọn  
[INTO tên_bảng_mới]  
FROM danh_sách_bảng/khung_nhìn  
[WHERE điều_kiện]  
[GROUP BY danh_sách_cột]  
[HAVING điều_kiện]  
[ORDER BY cột_sắp_xếp]  
[COMPUTE danh_sách_hàm_gộp [BY danh_sách_cột]]
```

**Câu lệnh INSERT** dùng để bổ sung một dòng dữ liệu mới vào bảng, ta sử dụng một trong hai cú pháp như sau:

```
INSERT INTO tên_bảng [(danh_sách_cột)]  
VALUES (danh_sách_trị)  
  
INSERT INTO tên_bảng [(danh_sách_cột)] câu_lệnh_SELECT
```

**Câu lệnh UPDATE** trong SQL được sử dụng để cập nhật dữ liệu trong các bảng. Câu lệnh này có cú pháp như sau:

```
UPDATE tên_bảng
SET tên_cột = biểu_thức
[, ..., tên_cột_k = biểu_thức_k]
[FROM danh_sách_bảng]
[WHERE điều_kiện]
```

**Câu lệnh DELETE** dùng để xóa dữ liệu trong một bảng. Cú pháp của câu lệnh này như sau:

```
DELETE FROM tên_bảng
FROM danh_sách_bảng ]
[ WHERE điều_kiện ]
```

**Lệnh BEGIN TRANSACTION** hoặc viết tắt là **BEGIN TRANS** dùng để bắt đầu một giao dịch. Để xác nhận giao dịch, ta thực hiện lệnh **COMMIT TRANSACTION** hoặc **COMMIT TRANS** hay **COMMIT**. Để quay ngược giao dịch để hủy bỏ các lệnh, ta gọi lệnh **ROLLBACK TRANSACTION** hoặc **ROLLBACK TRANS** hay **ROLLBACK**. Phần giữa lệnh **BEGIN** và **COMMIT** (hoặc **ROLLBACK**) là các lệnh SQL.

## 2. Tạo khung nhìn, trigger

**Câu lệnh CREATE VIEW** được sử dụng để tạo ra khung nhìn và có cú pháp như sau:

```
CREATE VIEW tên_khung_nhìn[(danh_sách_tên_cột)]
AS
    câu_lệnh_SELECT
```

Một khung nhìn (view) có thể được xem như là một bảng “ảo” trong cơ sở dữ liệu có nội dung được định nghĩa thông qua một truy vấn (câu lệnh SELECT).

**Câu lệnh CREATE TRIGGER** được sử dụng để định nghĩa trigger và có cú pháp như sau:

```
CREATE TRIGGER
tên_trigger ON tên_bảng
FOR {[INSERT] [,] [UPDATE] [,] [DELETE]}
AS
    [IF UPDATE(tên_cột)
    [AND UPDATE(tên_cột) |OR UPDATE(tên_cột)]
    ...]
    các_câu_lệnh_của_trigger
```

Một trigger là một đối tượng chứa một tập các câu lệnh SQL và tập các câu lệnh này sẽ được thực thi khi trigger được gọi. Các trigger được “gọi” tự động khi xảy ra những thao tác làm thay đổi dữ liệu trong các bảng.

Mỗi một trigger được tạo ra và gắn liền với một bảng nào đó trong cơ sở dữ liệu. Khi dữ liệu trong bảng bị thay đổi (tức là khi bảng chịu tác động của các câu lệnh INSERT, UPDATE hay DELETE) thì trigger sẽ được tự động kích hoạt.

### 3. Tạo thủ tục lưu trữ và hàm

Một thủ tục là một đối tượng trong cơ sở dữ liệu bao gồm một tập nhiều câu lệnh SQL được nhóm lại với nhau nhằm làm đơn giản hoá các thao tác trên cơ sở dữ liệu, thực thi lệnh nhanh hơn...

**Câu lệnh CREATE PROCEDURE** dùng để tạo thủ tục lưu trữ với cú pháp như sau:

```
CREATE PROCEDURE tên_thủ_tục [ (danh_sách_tham_số) ]  
[ WITH RECOMPILE | ENCRYPTION | RECOMPILE , ENCRYPTION ]  
AS  
    Các_câu_lệnh_của_thủ_tục
```

Hàm là đối tượng cơ sở dữ liệu tương tự như thủ tục. Điểm khác biệt giữa hàm và thủ tục là hàm trả về một giá trị thông qua tên hàm còn thủ tục thì không.

**Câu lệnh CREATE FUNCTION** dùng để định nghĩa một hàm với cú pháp như sau:

```
CREATE FUNCTION tên_hàm ( [danh_sách_tham_số] )  
RETURNS ( kiểu_trả_về_của_hàm )  
AS  
BEGIN  
    các_câu_lệnh_của_hàm  
END
```

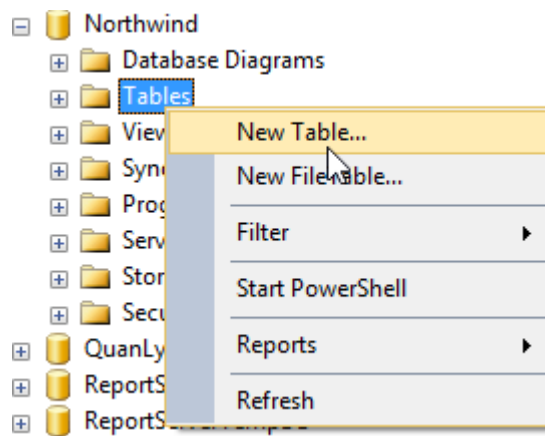
## III. Hướng dẫn thực hành

Trong suốt nội dung thực hành của môn Lập trình cơ sở dữ liệu, chúng ta sẽ xây dựng một chương trình quản lý bán hàng dựa trên cơ sở dữ liệu Northwind. Đây là cơ sở dữ liệu được thiết kế bởi Microsoft.

### 1. Thiết kế bảng

Trong cơ sở dữ liệu Northwind, thiết kế thêm bảng Khuyến Mãi với tên Promotions để lưu thông tin về các chương trình khuyến mãi của công ty. Mỗi chương trình khuyến mãi có các thông tin sau: Tên chương trình khuyến mãi, ngày bắt đầu, ngày kết thúc, mức khuyến mãi (tính theo % giảm giá).

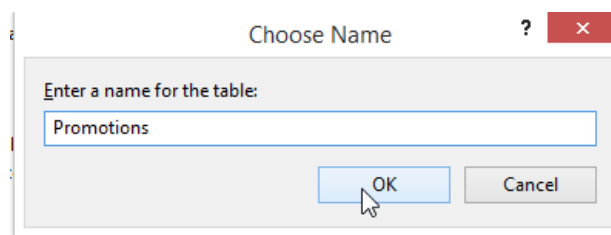
- Mở chương trình Enterprise Manager của MS SQL Server. Đăng nhập vào MSSQL SQL server
- Restore database northwind (theo dõi hướng dẫn trong file restoredatabase đi kèm)
- Trong CSDL northwind, mở thẻ tables và click phải chọn new table (hình 1)
- Trong cửa sổ mới, thiết kế bảng khuyến mãi (Promotion) như hình 2.  
Lưu ý promoID chứa giá trị tự tăng từ 1 với bước nhảy 01. Trên thanh công cụ, nhấn nút Save để lưu bảng khuyến mãi vừa tạo
- Trong cửa sổ Choose Name, nhập Promotions vào ô *Enter a name for the table* (hình 3).
- Click phải vào thẻ table và chọn refresh. Click bảng promotion (hình 4).



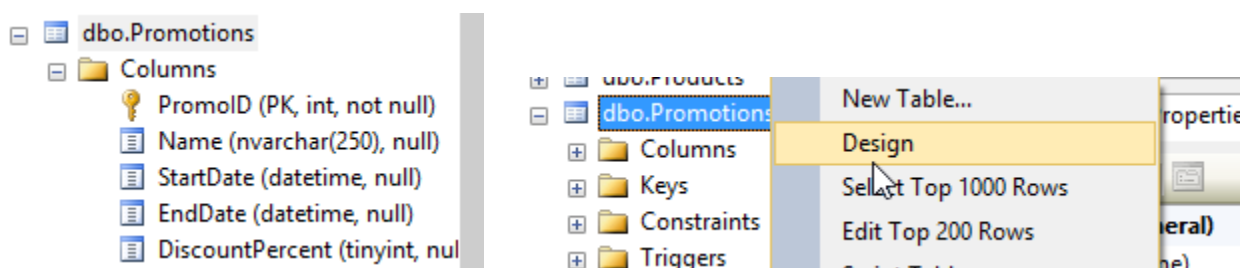
Hình 1. Sử dụng Enterprise Manager để tạo bảng mới

	Column Name	Data Type	Allow Nulls
🔑	PromoID	int	<input type="checkbox"/>
	Name	nvarchar(250)	<input checked="" type="checkbox"/>
	StartDate	datetime	<input checked="" type="checkbox"/>
	EndDate	datetime	<input checked="" type="checkbox"/>
	DiscountPercent	tinyint	<input checked="" type="checkbox"/>

Hình 2. Thiết kế bảng khuyến mãi (Promotions)



Hình 3. Đặt tên cho bảng mới.

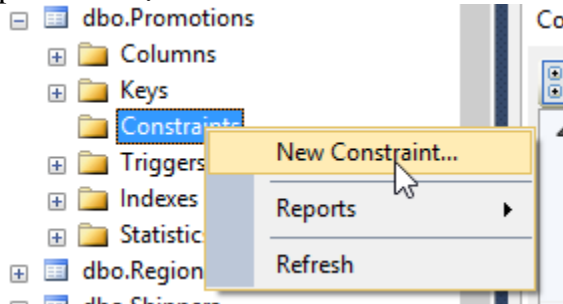


Hình 4. Thuộc tính của bảng Promotions bằng cách (1) mở thẻ columns trong bảng promotions. Hoặc (2) click phải vào bảng promotions và chọn design table

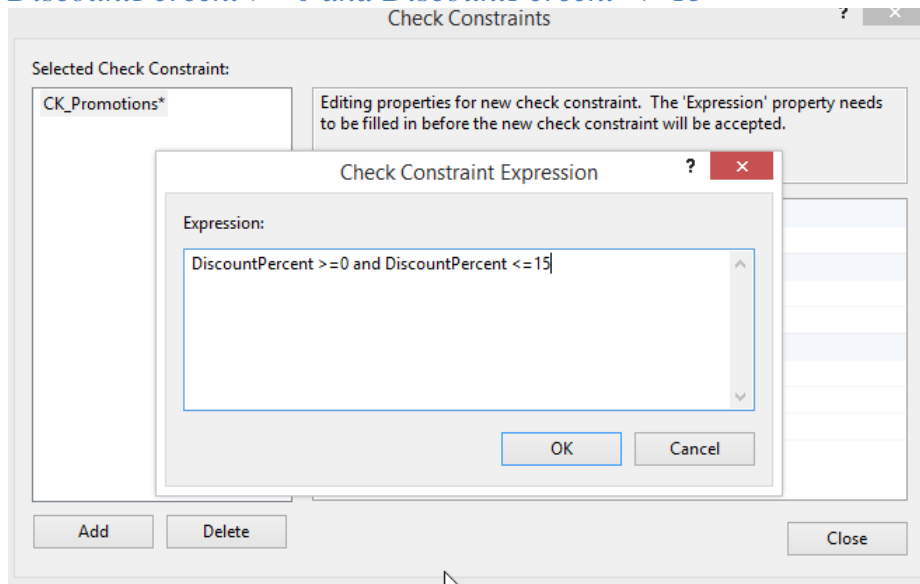
## 2. Tạo ràng buộc và chỉ mục

Tạo ràng buộc cho bảng Promotions trên cột giảm giá (DiscountPercent) để giá trị của nó nằm trong đoạn [0..15].

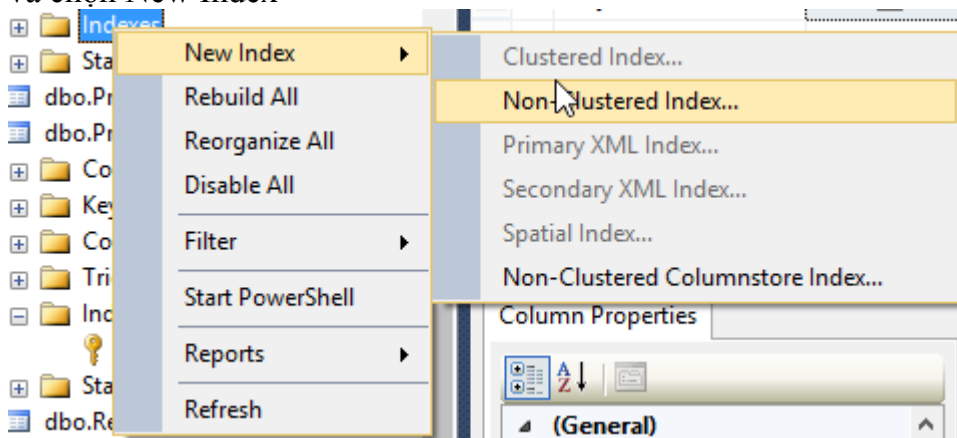
- Trong bảng Promotions, mở chọn thẻ Constraints. Click phải và chọn New constraints



- Trong ô expression và nhập expression cho discountpercent  
*DiscountPercent >=0 and DiscountPercent <=15*



- Để quản lý và tạo chỉ mục, click phải tại thẻ indexes trong bảng promotions và chọn New Index



### 3. Tạo Stored Procedure và hàm

Tạo hàm kiểm tra ngày hiện tại có nằm trong một khoảng thời gian nào đó hay không.

Trong cửa sổ query nhập lệnh sau

```
CREATE FUNCTION [CurrentDateAround]
(
    @start datetime, @end datetime, @curr datetime
)
RETURNS tinyint
AS
BEGIN
    DECLARE @var tinyint;
    IF ((@curr >= @start) AND (@curr <= @end))
        SET @var = 1;
    ELSE
        SET @var = 0;
    RETURN @var;
END
```

- Nhấn F5 Check Syntax để kiểm tra cú pháp.

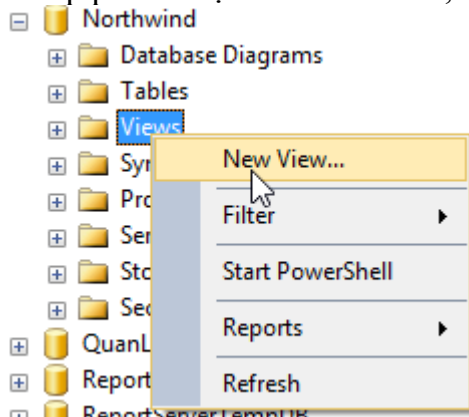
Tạo thủ tục lấy danh sách các chương trình khuyến mãi được áp dụng trong ngày hiện tại.  
Trong cửa sổ query chạy đoạn mã sau:

```
CREATE PROCEDURE GetPromotionForDate
AS
SELECT PromoID, Name, DiscountPercent
FROM Promotions
WHERE dbo.CurrentDateAround(StartDate,EndDate, getdate())=1
```

### 4. Tạo Khung nhìn

Tạo khung nhìn cho biết thông tin chi tiết các hóa đơn gồm: số hóa đơn, tên khách hàng, tên nhân viên bán hàng và ngày bán hàng.

- Nhấp phải chuột vào thẻ Views, chọn New View



- Trong hộp thoại Add Table, nhấp đôi vào 3 bảng Customers, Employees và Orders.



#### IV. Bài tập thực hành

1. Thiết kế thêm bảng để quản lý tình hình **thu chi** (IncOut) của công ty gồm các thông tin: Ngày, Giờ, Số tiền, Lý do, Loại (Thu hay Chi), Nhân viên thực hiện.
2. Thiết kế thêm bảng nhật ký hệ thống (SystemLog) để lưu lại các hoạt động của người dùng trên chương trình. Bảng này chứa các thông tin sau: Ngày giờ, loại thao tác (Xóa, Thêm, Cập nhật, Sao lưu), người thực hiện, dữ liệu bị ảnh hưởng.
3. Tạo các ràng buộc trên bảng IncOut và SystemLog sao cho:
  - a. Loại chỉ là thu hoặc chi.
  - b. Số tiền là không âm.
  - c. Loại thao tác là một trong 4 công việc Xóa, Thêm, Cập nhật, Sao lưu
4. Tạo chỉ mục cho cột Lý do (của bảng IncOut) và Dữ liệu bị ảnh hưởng (của bảng SystemLog)
5. Viết các lệnh SQL để thêm, xóa, cập nhật một vài mẫu tin trong hai bảng trên.
6. Tạo khung nhìn (View) chứa thông tin chi tiết về các sản phẩm được bán trong ngày, trong tuần, trong tháng, trong quý, trong năm.
7. Tạo các thủ tục để thêm, xóa, cập nhật dữ liệu cho bảng IncOut và SystemLog.
8. Tạo các thủ tục để xem tình hình thu chi của công ty trong ngày, trong tháng, trong năm.
9. Tạo các thủ tục để xem danh sách các hoạt động trên hệ thống theo ngày, hoặc.
10. Viết hàm trả về tổng thu nhập trong ngày, tháng hoặc năm.
11. Viết hàm số lượng sản phẩm được bán trong ngày.
12. Tạo các trigger để tự động điền thông tin vào bảng SystemLog khi có sự thay đổi về mặt dữ liệu trong các bảng khác.