

Lab 10

Lập trình Transact-SQL2012

Phần I – Hướng dẫn thực hành từng bước

1. Mục tiêu

- Hiểu biết về lập trình Transact-SQL.
- Hiểu các thành phần trong cú pháp lập trình.
- Hiểu luồng câu lệnh.
- Khai báo hàm.

2. Thực hiện

Lập trình Transact-SQL là một phần mở rộng của ngôn ngữ thủ tục cho SQL. Lập trình Transact-SQL mở rộng bằng cách thêm các chương trình con có cấu trúc lập trình tương tự các ngôn ngữ lập trình. Lập trình T-SQL có những cú pháp, quy tắc kiểm soát để đảm bảo các câu lệnh lập trình có thể làm việc với nhau. Những mệnh đề điều kiện, vòng lặp thường gặp ở nhiều ngôn ngữ lập trình cũng được sử dụng trong T-SQL giúp việc lập trình trở lên thuận tiện dễ dàng hơn.

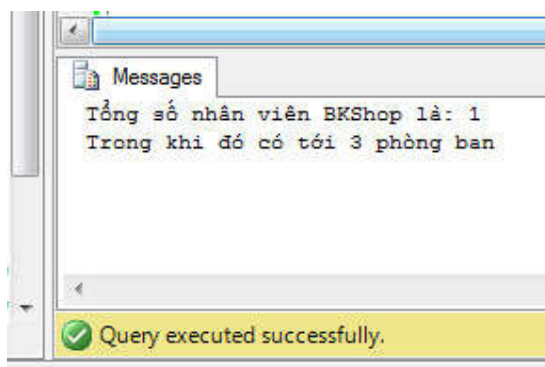
Các dòng lệnh, cú pháp thực thi tạo bảng, view, trigger, procedure...đồng thời thêm, sửa, xóa chính là việc sử dụng T-SQL theo dạng khối lệnh.

Bài thực hành 1: Khai báo biến tong_so_nv rồi thực hiện truy vấn lấy về tổng số nhân viên trong csdl BKShop đã tạo ở bài trước.

Bước 1: Viết lệnh SQL:

```
DECLARE
    @tong_so_nv int,
    @tong_so_pb int
SELECT @tong_so_nv = COUNT(*) FROM NhanVien
SET @tong_so_pb = (SELECT COUNT(*) FROM PhongBan)
PRINT N'Tổng số nhân viên BKShop là: ' +
CONVERT(varchar,@tong_so_nv)
PRINT N'Trong khi đó có tới ' + CONVERT(varchar,@tong_so_pb) + N'
phòng ban'
GO
```

Bước 2: Thực thi câu lệnh trên:



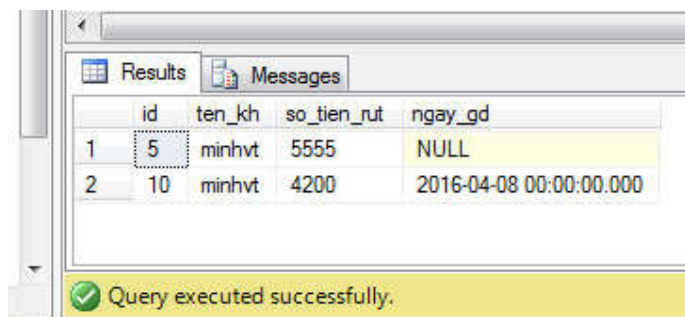
Kết quả

Bài thực hành 2: Tạo từ đồng nghĩa để định danh một đối tượng khác (bảng, view....) ở máy chủ khác bằng lệnh T-SQL.

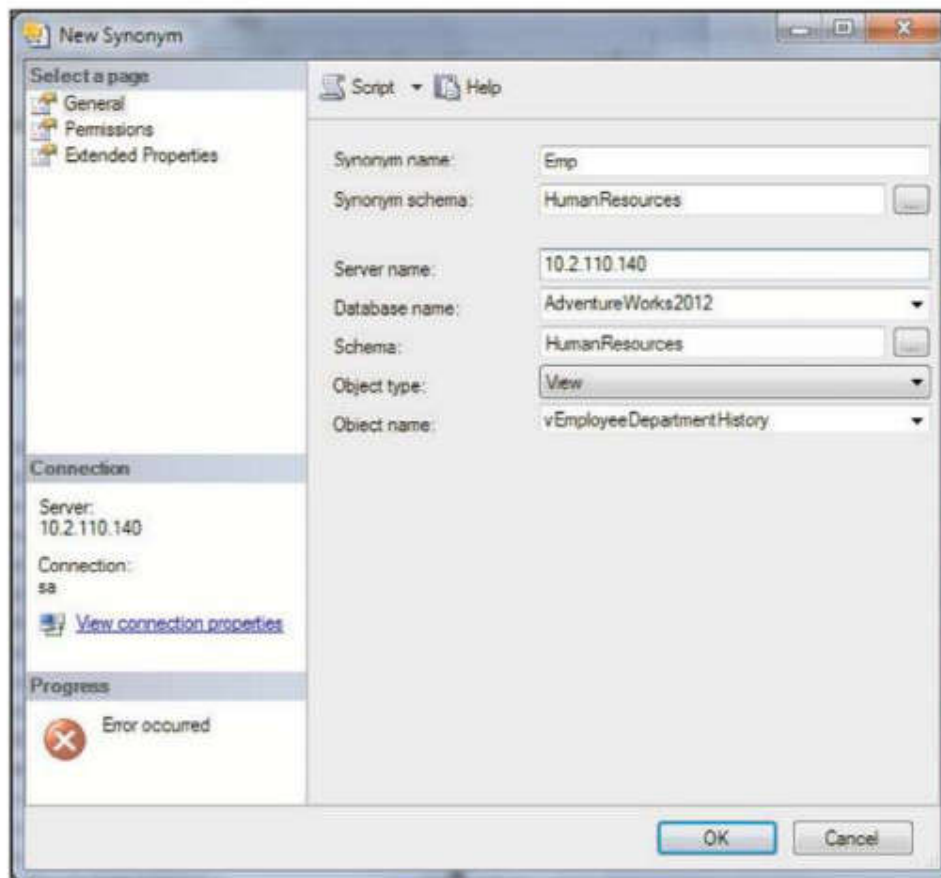
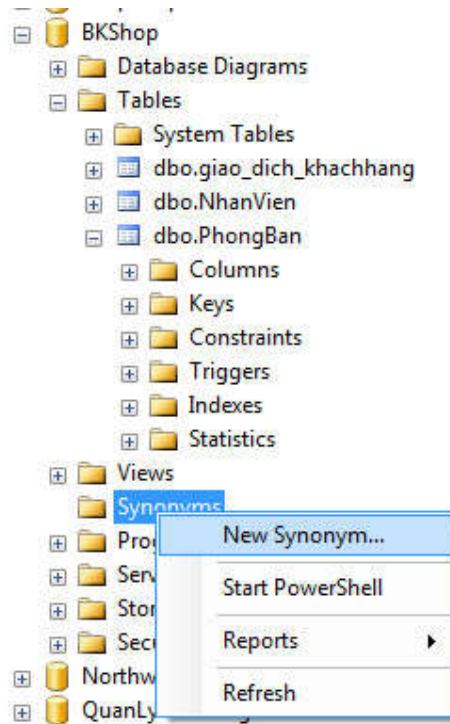
Bước 1: Gõ lệnh SQL:

```
CREATE SYNONYM GDKH FOR BKShop.dbo.giao_dich_khachhang
GO
SELECT * FROM GDKH
GO
```

Bước 2: Thực thi câu lệnh:



Tình huống trên vẫn còn đơn giản, thực tế việc sử dụng từ đồng nghĩa để rút gọn lời gọi từ csdl khác rất hiệu quả. Sau đây là cách tạo bảng công cụ:

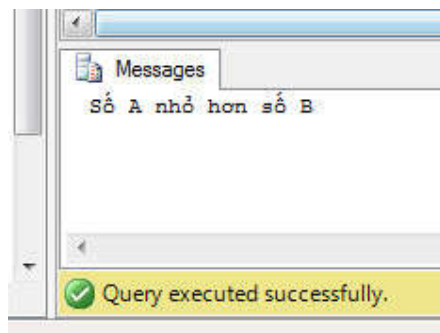


Bài thực hành 3: Viết lệnh T-SQL tạo 2 số rồi gán giá trị cho chúng, so sánh giá trị và in ra kết quả.

Bước 1: Gõ lệnh SQL:

```
BEGIN
    DECLARE
        @soA int,
        @soB int
    SET @soA = 6
    SET @soB = 8
    IF @soA > @soB
        PRINT N'Số A lớn hơn số B'
    ELSE
        PRINT N'Số A nhỏ hơn số B'
END
```

Bước 2: Thực thi lệnh SQL:



Lệnh thực thi và hiển thị kết quả. IF-ELSE là một trong số lệnh điều khiển luồng trong T-SQL như sau đây:

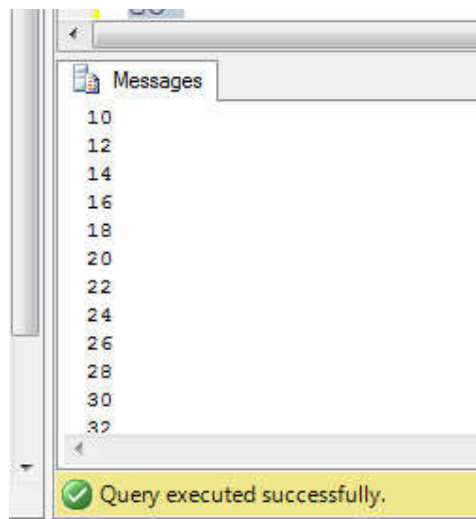
Control-Of-Flow Language Keywords
RETURN
THROW
TRY....CATCH
WAITFOR
WHILE
BEGIN....END
BREAK
CONTINUE
GOTO label
IF...ELSE

Bài thực hành 4: In ra màn hình tất cả số chẵn từ 10 đến 95.

Bước 1: Gõ lệnh SQL:

```
DECLARE @flag int
SET @flag = 10
WHILE (@flag <=95)
BEGIN
    IF @flag%2 =0
        PRINT @flag
        SET @flag = @flag + 1
        CONTINUE;
    END
GO
```

Bước 2: Thực thi lệnh SQL:



Thử thực hiện lệnh SQL với chút sửa đổi để xem hiệu quả lệnh BREAK và CONTINUE.

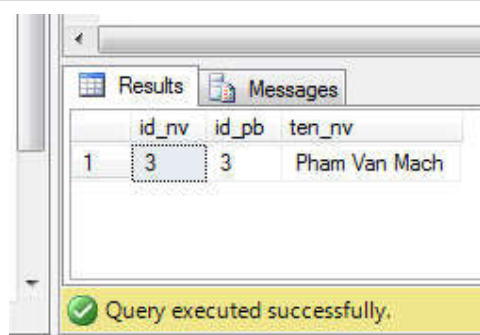
```
DECLARE @flag int
SET @flag = 10
WHILE (@flag <=95)
BEGIN
    SET @flag = @flag + 1
    IF @flag < 30
        CONTINUE; -- Từ khóa có tác dụng chuyển vòng lặp kế tiếp
    IF @flag = 60
        BREAK; -- Dừng ngay lập tức vòng lặp
    IF @flag%2 =0
        PRINT @flag
END
GO
```

Bài thực hành 5: Tạo hàm truy vấn lấy tất cả nhân viên phòng kỹ thuật trong csdl BKShop.

Bước 1: Gõ lệnh SQL:

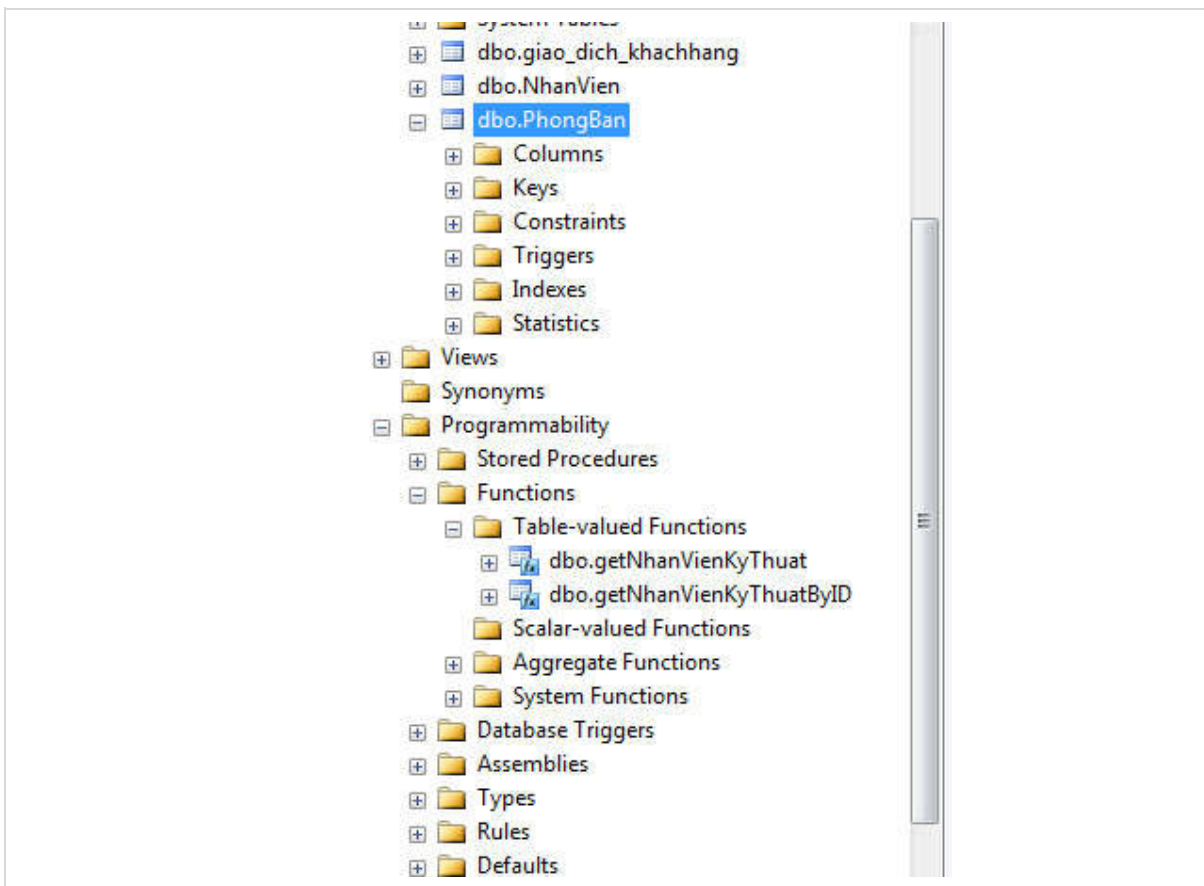
```
CREATE FUNCTION getNhanVienKyThuat ()
RETURNS TABLE
AS
RETURN (
    SELECT * FROM NhanVien WHERE id_pb = 3
)
GO
SELECT * FROM getNhanVienKyThuat ()
GO
```

Bước 2: Thực thi lệnh :



Sửa câu lệnh trên để hàm có thể nhận thêm tham số truyền vào, giúp tăng tính uyển chuyển của mã code.

```
CREATE FUNCTION getNhanVienKyThuatByID(@id int)
RETURNS TABLE
AS
RETURN (
    SELECT * FROM NhanVien WHERE id_pb = @id
)
GO
SELECT * FROM getNhanVienKyThuatByID(3)
```

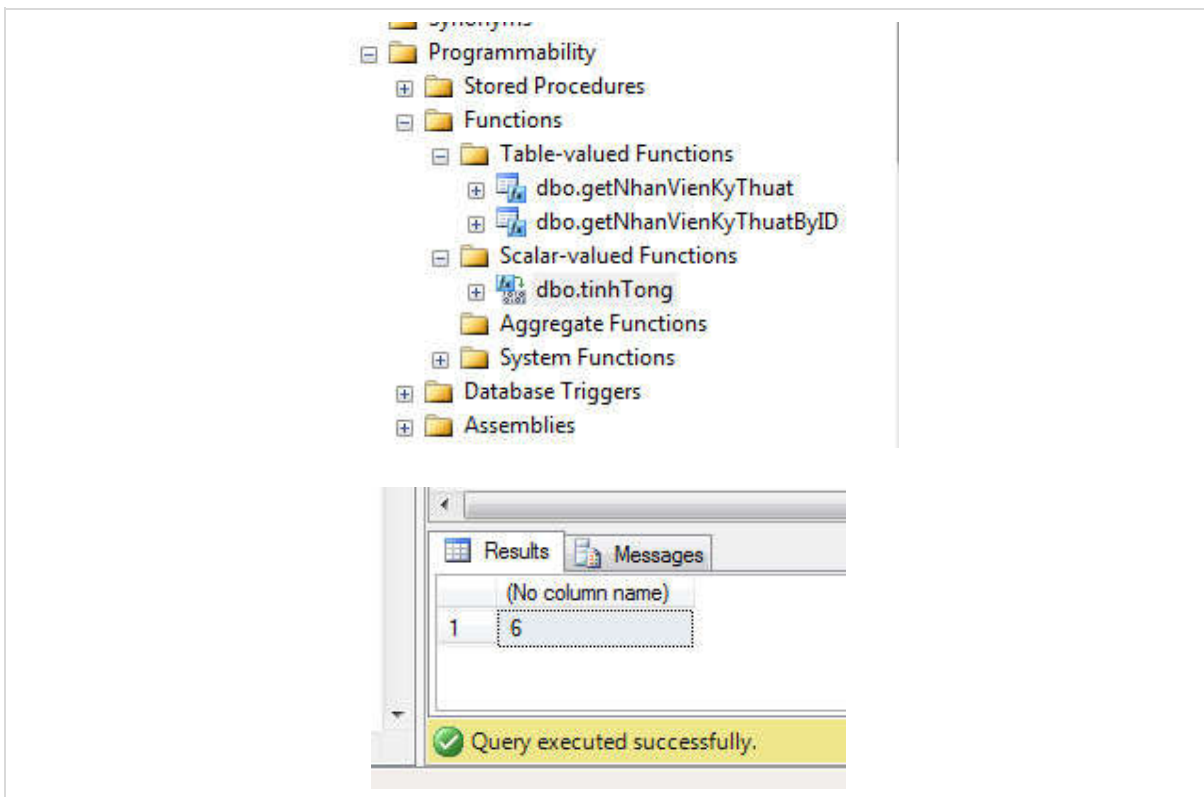


Bài thực hành 6: Tạo hàm thực thi tính tổng 2 số truyền vào.

Bước 1: Gõ lệnh SQL:

```
GO
CREATE FUNCTION tinhTong(@soA int, @soB int)
RETURNS int
AS
BEGIN
    RETURN @soA + @soB
END
GO
SELECT dbo.tinhTong(2,4)
GO
```

Bước 2: Thực thi lệnh :



Phần II - Bài tập tự làm

Thực hành làm bài Assignment theo hướng dẫn của giảng viên.