

Trường Đại Học Công Nghiệp Tp.Hồ Chí Minh  
Khoa Công Nghệ Thông Tin

BÀI TẬP THỰC HÀNH

# BẢO MẬT CƠ SỞ DỮ LIỆU SQL SERVER

Lưu Hành Nội Bộ  
Năm 2023

## MỤC LỤC

Bài Thực Hành Tuần 1, 2, 3 .....	2
LOGIN – USER – ROLES .....	2
Bài Thực Hành Tuần 4 .....	31
AUDTING .....	31
Bài Thực Hành Tuần 5 .....	38
MÃ HÓA VÀ GIẢI MÃ.....	38
Bài thực hành tuần 6.....	52
Replication .....	52
Bài tập thực hành 7 .....	54
Mirroring và log shipping .....	54
Bài thực hành tuần 8 .....	57
PowerShell .....	57
Bài thực hành tuần 9.....	60
POLICY-BASED.....	60
Bài tập thực hành tuần 10 .....	81
Ôn Tập – Kiểm tra .....	81

## Bài Thực Hành Tuần 1, 2, 3

### LOGIN – USER – ROLES

#### Mục tiêu:

- Phân quyền theo các mô hình DAC, MAC, RBAC
- Tạo, sửa, xóa được các login
- Tạo, sửa, xóa được các user
- Tạo, sửa, xóa được các roles
- Thực hiện được việc cấp quyền, thu hồi và từ chối quyền cho các user và roles

#### PHẦN 1: LÝ THUYẾT

1. Cho biết các mục tiêu chính của bảo mật? Các mức bảo mật mà SQL Server hỗ trợ.
2. SQL Server hỗ trợ bao nhiêu chế độ chứng thực? Sự khác biệt? Để thay đổi chế độ chứng thực của một thể hiện SQL Server, bạn phải thực hiện như thế nào?
3. Cho biết logins, users là gì?
4. Cho biết Roles là gì? Có mấy loại, mức độ như thế nào? Liệt kê các Roles mà SQL Server có hỗ trợ, nếu bạn là thành viên của Roles đó thì bạn có quyền hạn như thế nào? Cho biết Permissions là gì?
5. Cho biết đặc điểm, ưu và khuyết điểm của ba mô hình DAC, MAC, RBAC

#### PHẦN 2: THỰC HÀNH

##### BÀI 1

1. Tạo cơ sở dữ liệu QLTV tham số tùy ý
2. Tạo các LOGIN Minh, Huy, Le, Linh, An, và Binh:
  - a. Password lần lượt là tên username viết hoa.

Hướng dẫn:

1. Tạo CSDL QLTV  
`create database QLTV`
2. Tạo các users Minh, Huy, Le, Linh, An, và Binh:
  - a) Password lần lượt là tên username viết hoa.  
`use QLTV`  
`go`  
`create login Minh with password='MINH'`  
`go`

```

create user Minh for login Minh
go
use QLTV
go
create login Huy with password='HUY'
go
create user Huy for login Huy
go
use QLTV
go
create login Le with password='LE'
go
create user Le for login Le
go
use QLTV
go
create login Linh with password='LINH'
go
create user Linh for login Linh
go
use QLTV
go
create login An with password='AN'
go
create user An for login An
go
use QLTV
go
create login Binh with password='BINH'
go
create user Binh for login Binh

```

### 3. Cho bảng Sach

```

Sach(MaSach INT PRIMARY KEY,
     TenSach NVARCHAR(40)
)
Tạo bảng Sach

```

```
Create table Sach
(
    ID INT PRIMARY KEY,
    Name NVARCHAR(2)
)
```

Làm các bước sau:

- a. Tạo các role sau: DataEntry, Supervisor, và Management.

```
go
use QLTV
go
create role DataEntry
create role Supervisor
create role Management
--xem thông tin các role
sp_helprole
```

Gán Minh, Huy, và Linh vào role DataEntry, gán Le vào role Supervisor, và gán An và Binh vào role Management.

```
go
EXEC sp_addrolemember 'DataEntry','Minh'
go
EXEC sp_addrolemember 'DataEntry','Huy'
go
EXEC sp_addrolemember 'DataEntry','Lym'
--role Supervisor
go
EXEC sp_addrolemember 'Supervisor','Le'
--role Management
go
EXEC sp_addrolemember 'Management','An'
go
EXEC sp_addrolemember 'Management','Binh'
```

- b. Cho role DataEntry các quyền SELECT, INSERT, và UPDATE trên bảng Sach.

```
go
use QLTV
go
grant select,insert,update on Sach to DataEntry
```

- c. Cho role Supervisor các quyền SELECT và DELETE trên bảng Sach.

```
go
use QLTV
go
grant select,delete on Sach to Supervisor
```

- d. Cho role Management quyền SELECT trên bảng Sach.

```
go
use QLTV
go
grant select on Sach to Management
```

- e. Lần lượt kiểm tra kết quả và giải thích các lệnh đã thực hiện được tương ứng với phân quyền đã cấp cho các role

--Dang nhap bang login Minh thuc hien lenh select , insert, update cua role DataEntry

```
use QLTV
go
select * from Sach
go
insert into Sach values (1,'Toán')
go
select * from Sach
go
update Sach
set Name=N'Hình học'
where ID = 1
go
select * from Sach
go
delete from Sach where ID=1
```

--Dang nhap bang login Le thuc hien các lệnh sau của role Supervisor

```
use QLTV
go
select * from Sach
go
update Sach
set Name=N'Giải tích'
where ID = 1
go
select * from Sach
delete from Sach where ID=1
```

```

go
select * from Sach
-- Dang nhap bang user AN thuc hien các lệnh sau của role Supervisor
use QLTV
go
select * from Sach
go
update Sach
set Name=N'Giải tích'
where ID = 1
go
select * from Sach
delete from Sach where ID=1
go
select * from Sach

```

4. Tạo một user mới tên NameManager với password là pc123. Gán quyền update cho user này trên cột TenSach của bảng Sach.

```

--tao user
go
use QLTV
go
create login NameManager with password='pc123'
go
create user NameManager for login NameManager
--gan quyen
go
use QLTV
go
grant update on Sach(Name) to NameManager
--user chay thu quyen dc cap
go
use QLTV
go
update Sach
set Name=N'Văn'
where ID=1
go

```

```
update Sach
set ID=3
where Name=N'Văn'
```

–Lệnh xem các quyền

```
SELECT
  [UserName] = CASE memberprinc.[type]
    WHEN 'S' THEN memberprinc.[name]
    WHEN 'U' THEN ulogin.[name] COLLATE Latin1_General_CI_AI
  END,
  [UserType] = CASE memberprinc.[type]
    WHEN 'S' THEN 'SQL User'
    WHEN 'U' THEN 'Windows User'
  END,
  [DatabaseUserName] = memberprinc.[name],
  [Role] = roleprinc.[name],
  [PermissionType] = perm.[permission_name],
  [PermissionState] = perm.[state_desc],
  [ObjectType] = obj.type_desc,--perm.[class_desc],
  [ObjectName] = OBJECT_NAME(perm.major_id)
FROM
  --Role/member associations
  sys.database_role_members members
JOIN
  --Roles
  sys.database_principals roleprinc ON roleprinc.[principal_id] =
members.[role_principal_id]
JOIN
  --Role members (database users)
  sys.database_principals memberprinc ON memberprinc.[principal_id] =
members.[member_principal_id]
LEFT JOIN
  --Login accounts
  sys.login_token ulogin on memberprinc.[sid] = ulogin.[sid]
LEFT JOIN
  --Permissions
```



```

sys.database_permissions perm ON perm.[grantee_principal_id] =
roleprinc.[principal_id]
LEFT JOIN
--Table columns
sys.columns col on col.[object_id] = perm.major_id
AND col.[column_id] = perm.[minor_id]
LEFT JOIN
sys.objects obj ON perm.[major_id] = obj.[object_id]

```

6. Thực hiện các bước sau:

- a. Cho phép user Minh quyền cấp quyền cho các user khác
- b. Gán tất cả các quyền mà Minh có cho Binh. Binh có quyền INSERT và UPDATE trên bảng QLTV không?

--a. Cho phép user Minh quyền cấp quyền cho các user khác

```

go
use QLTV
go
grant select,insert,update on Sach to Minh with grant option
-- Minh gan quyen cho user #
go
use QLTV
go
grant insert on Sach to An

-- An chạy thu quyền dc cấp
go
use QLTV
go
insert into Sach values (2,'Hóa')
go
select * from Sach

```

--b. Gán tất cả các quyền mà Minh có cho Binh. Binh có quyền INSERT và UPDATE trên bảng Sach không?

--dang nhap bang user Minh

```

go
use QLTV
go
grant select,insert,update on Sach to Binh

```

```
--Binh chay thu quyen dc cap
use QLTV
go
select * from Sach
go
insert into Sach values (4,N'Lý')
go
select * from Sach
go
update Sach
set Name='Sinh'
where ID= 4
go
select * from Sach
--=> Binh có quyền INSERT và UPDATE trên bảng Sach
```

## BÀI 2

4. Tạo cơ sở dữ liệu QLTV tham số tùy ý. Trong CSDL QLTV có các bảng dữ liệu sau:

**NhomSach**(**MaNhom** char(5), TenNhom nvarchar(25))

**NhanVien**(**MaNV** char(5), HoLot nvarchar(25), TenNV nvarchar(10), Phai nvarchar(3), NgaySinh smalldatetime, DiaChi nvarchar(40))

**DanhMucSach**(**MaSach** char(5), TenSach nvarchar(40), TacGia nvarchar(20),

MaNhom char(5), DonGia Numeric(5), SoLuongTon numeric(5))

**HoaDon**(**MaHD** char(5), NgayBan SmallDatetime, MaNV char(5))

**ChiTietHD**(MaHD char(5), MaSach char(5), SoLuong numeric(5))

Dữ liệu cho các table trên

NhomSach

MANHOM	TENNHOM
N001	Kỹ thuật trồng trọt

Sach

MaSH	TenSach	TacGia	MaNH	DonGia	SoLuongTon
S111	Đèn không hắt bóng	Dzunichi Watanabe (Cao Xuân Hạo dịch)	N001	55000	45

S112	Kỹ thuật trồng hoa phong lan	Nguyễn Lâm Hùng	N001	45000	35
S113	Kỹ thuật chăm sóc hoa mai	Lê Xuân A	N007	35000	15
S114	Kỹ thuật chăm sóc cây cam	Trần Ha	N001	24000	12

Các Table khác sinh viên tự thêm dữ liệu vào

- Tạo các users Minh, Huy, Le, Linh, và Binh. Password lần lượt là tên username viết hoa.
- Cho bảng ma trận phân quyền như sau:

R: Read -Xem, U: Update – Sửa, D: Delete – Xoá, I: Insert – Thêm

	Nhom Sach	Nhan Vien	DanhMuc Sach	HoaDon	ChiTietHD	Khác
Minh QLTV	R, I, U, D	Owner	R, I, U, D	R, I, U, D	R, I, U, D	Được cấp quyền cho người khác
Huy QLNVTV	R	R, I, U, D	R			
Le QLBH	R, U	R	R, U	R	R	
Linh NVBH	R	R	R	R, I, U, D	R, I, U, D	Các quyền này do QLTV cấp
Bình NVKho	R, I, U, D	R	R, I, U, D			Các quyền này do QLTV cấp

- Viết các lệnh phân quyền cho Minh, Huy, Le, Linh, Bình theo ma trận phân quyền trên. Chú ý Minh là sở hữu table NhanVien. Bạn viết lệnh phân quyền cho phép Minh tạo và thực hiện được các lệnh cho Table NhanVien
- Đăng nhập vào từng Login và thực hiện các lệnh cho từng người dùng. Các lệnh sinh viên tự nghĩ và thực hiện đủ các lệnh trong cả hai trường hợp là người dùng thực hiện được và không thực hiện được. Giải thích cho từng lệnh.
- Thu hồi quyền sửa trên bảng DanhMucSach cho người dùng Le. Viết lệnh kiểm tra tương ứng.
- Thu hồi quyền sửa và xoá trên bảng NhomSach và DanhMucSach cho người Bình. Viết lệnh kiểm tra tương ứng.
- Viết lệnh từ chối quyền xoá trên bảng NhanVien cho người tên Huy. Viết lệnh kiểm tra tương ứng

### BÀI 3: PHÂN QUYỀN THEO VAI TRÒ

- 1) Tạo CSDL ThuVien
- 2) Thực hiện các lệnh import hay Select...into các Table từ CSDL QLTV ở BÀI 1 vào CSDL ThuVien
- 3) Tạo các users Minh, Huy, Le, Linh, và Binh. Password lần lượt là tên username viết hoa.
- 4) Tạo các role sau: QLBH, NVKHO, QLNVTV, QLTV, NVBH cho CSDL ThuVien
- 5) Gán các người dùng Minh, Huy, Le, Linh, Binh vào các Role tương ứng theo ma trận phân quyền trong BÀI 1.
- 6) Lần lượt đăng nhập vào từng Login và thực hiện các lệnh cho từng người dùng. Các lệnh sinh viên tự nghĩ và thực hiện đủ các lệnh trong cả hai trường hợp là người dùng thực hiện được và không thực hiện được. Giải thích cho từng lệnh.
- 7) Tạo một user mới tên Lan với password là Lan123. Gán quyền update cho user này trên cột TenSach của bảng Sach. Thực hiện lệnh kiểm tra tương ứng.
- 8) Thu hồi quyền cho Role NVBH. Thực hiện lệnh kiểm tra tương ứng
- 9) Thu hồi quyền của người dùng có tên Lan
- 10) Tạo một user mới tên Lan với password là Lan123. Gán quyền update cho user này trên cột TenSach của bảng Sach. Thực hiện lệnh kiểm tra tương ứng. Viết lệnh DENY cho người dùng này

## BÀI 4

CSDL mẫu: **AdventureWorks2008**

Hướng dẫn: Attach file AdventureWorks2008\_Data.mdf, AdventureWorks2008\_Log.ldf để tạo cơ sở dữ liệu mẫu **AdventureWorks2008**

*Giả sử ngoài CSDL do SQL server hỗ trợ, còn có CSDL **AdventureWorks2008**,.... Bạn tạo login chỉ có quyền được cho (không được có quyền cao hơn), tạo xong bạn phải đăng nhập vào SQL Server bằng chính login vừa tạo, thực hiện kiểm tra quyền bằng cách thực hiện các câu lệnh ứng với quyền được phép và các câu lệnh ứng với quyền không được phép.*

## II. TẠO CÁC ROLES, LOGINS, GÁN CÁC QUYỀN BẰNG THAO TÁC TRONG SQL SERVER MANAGEMENT STUDIO (SSMS).

### 1. Tạo login dạng **SQL Server Authentication**

- a. Tạo một login có tên là tên của bạn, login có:
  - Chế độ chứng thực là SQL Server Authentication, password tùy ý, CSDL mặc định là **AdventureWorks2008**.
  - Không thuộc Server Roles nào cả
  - Chỉ cho truy xuất đến duy nhất CSDL là **AdventureWorks2008** và không thuộc Database Roles nào cả ngoại trừ **Public**

#### Kiểm tra:

- Ở SSMS, kiểm tra xem tên login của bạn có nằm trong nhánh **Security\Login** không? kiểm tra xem tên login của bạn có nằm trong nhánh User của CSDL **AdventureWorks2008** không? Xem thuộc tính (properties) của nó.
  - Kết nối vào SSMS bằng login vừa tạo
  - Trong mục database bạn có thể nhìn thấy được những database nào? Tại sao?
  - Dùng câu lệnh **SELECT ... FROM...** để xem các mẫu tin trong bảng **Production.Product**, bạn xem được không? Tại sao?
- b. Hiệu chỉnh login ở trên, cho phép login thuộc database Roles tên là **db\_DataReader** trong CSDL **AdventureWorks2008**

#### Kiểm tra:

- Dùng câu lệnh **SELECT ... FROM...** để xem các mẫu tin trong bảng **Production.Product**, bạn xem được không? Tại sao?
  - Dùng câu lệnh **INSERT ... VALUES** để chèn một mẫu tin mới vào bảng **Production.Product**, bạn có chèn được không? Tại sao? Muốn chèn được bạn phải làm gì? Thực hiện thử xem sao.
- c. Tương tự như vậy, lần lượt tìm hiểu các database Roles còn lại.

### 2. Tạo login dạng **Windows Authentication**

- a. Quay về hệ điều hành tạo một **local user account** hoặc **domain user account** được phép kết nối đến máy Server của SQL Server. User account này có tên là **Nhanvien1**.
- b. Cho phép **Nhanvien1** trở thành login của SQL Server, login này chỉ thuộc vào database Roles là **db\_datareader** của CSDL là **AdventureWorks2008**. (*Lưu ý: phải chọn Windows Authentication*)
- c. Bạn hãy thử kết nối Server thông qua công cụ SSMS bằng login vừa tạo và kiểm tra quyền của login đối với **AdventureWorks2008**.  
(*Hướng dẫn: đóng hết các ứng dụng đang chạy, log off user hiện kết nối đến máy, log on vào máy bằng user account vừa tạo, kết nối vào SSMS bằng login*)
3. Login được tạo ở câu 1 và câu 2 có thể thực hiện tạo được Table, view, ... trong CSDL **AdventureWorks2008** hay không? Muốn tạo được cần có điều gì?
4. Tạo một Database Role có tên là **NVHoaDon** của CSDL **AdventureWorks2008**, Role này có quyền hạn như sau:
  - Được phép chèn, cập nhật dữ liệu trong hai bảng **Purchasing.PurchaseOrderHeader** và **Purchasing.PurchaseOrderDetail**
  - Chỉ được phép xem (Select) trên bảng **Purchasing.WorkOrder**
5. Tạo 3 login dạng SQL Server Authentication, có tên lần lượt là **NVHD1**, **NVHD2**, **NVHD3**. Các Login này chỉ thuộc duy nhất DataBase Role là **NVHoaDon** đã tạo ở trên. Đăng nhập vào từng login **NVHD1**, **NVHD2**, **NVHD3**, ứng với mỗi login thực hiện các công việc sau:
  - Xem thông tin các bảng **Purchasing.PurchaseOrderHeader**, **Purchasing.PurchaseOrderDetail**, **Purchasing.WorkOrder**
  - Chèn vào các bảng **Purchasing.PurchaseOrderHeader**, **Purchasing.PurchaseOrderDetail**, **Purchasing.WorkOrder**, mỗi bảng 1 record với dữ liệu tùy ý, chú ý các ràng buộc khóa ngoại
  - Xóa một record bất kỳ trong mỗi bảng sau **Purchasing.PurchaseOrderHeader**, **Purchasing.PurchaseOrderDetail**, **Purchasing.WorkOrder**.
  - Nếu thực hiện lệnh Update cho 3 bảng **Purchasing.PurchaseOrderHeader**, **Purchasing.PurchaseOrderDetail**, **Purchasing.WorkOrder** có thực hiện được không? Giải thích và cho ví dụ minh họa trong cả 2 trường hợp được hoặc không được.
6. Tạo 3 login dạng SQL Server Authentication, có tên lần lượt là **QLKho1**, **QLKho2**, **QLKho3**. Các login này có cùng một quyền hạn là được phép chèn, xóa dữ liệu trên bảng **Production.Product**; cập nhật duy nhất cột **ListPrice** trong bảng **Production.Product**. Chỉ được phép xem (Select) trên bảng **Production.WorkOrder**. Cho ví dụ kiểm tra các trường hợp đã cấp quyền cho mỗi login thông qua các lệnh insert, update, delete, select

7. Bạn chọn một giải pháp đơn giản nhất để cho phép các login đã tạo ở trên được phép xem thông tin trong bảng **HumanResources.Employee**.
8. Tạo hai login thuộc dạng **SQL Server Authentication**, có tên lần lượt là **PTUD1**, **PTUD2**.
  - a) Với login **PTUD1** có các quyền như sau:
    - Được phép tạo các đối tượng của database
    - Được phép truy xuất và hiệu chỉnh các đối tượng database
  - b) Với login **PTUD2** có các quyền như sau:
    - Được phép tạo các đối tượng của database
    - Được phép truy xuất và hiệu chỉnh các đối tượng database
  - c) **Ứng với mỗi login thực hiện các lệnh sau:**
    - 1) Tạo Table UngDung(MaUD int primary key, TenUD nvarchar(30))
    - 2) Thêm cột TacGia nvarchar(30) vào bảng UngDung
    - 3) Tăng độ rộng cho cột TenUD lên 50 ký tự
    - 4) Thêm vào UngDung 2 record có dữ liệu tùy ý
    - 5) Tạo thủ tục cho phép xem thông tin của một ứng dụng bất kỳ
    - 6) Xóa dữ liệu có trong bảng UngDung
    - 7) Chạy thủ tục đã tạo ở câu e
    - 8) Xóa thủ tục câu e

Bạn hãy đưa ra kết quả và nhận xét sau khi thực thi mỗi lệnh.

# I. TẠO CÁC ROLES, LOGINS, GÁN CÁC QUYỀN BẰNG T\_SQL THÔNG QUA CÁC THỦ TỤC HỆ THỐNG.

Chú ý sau mỗi câu bạn thực hiện kiểm tra lại các lệnh bạn vừa thực hiện

1. Tạo một login dạng Windows Authentication có tên là **GD1** (vào hệ điều hành Window tạo user GD1 trước khi tạo ).
2. Tạo hai login dạng SQL Server Authentication tên là **PGD1** và **PGD2** có password tùy ý.
3. Bạn hãy tạo một user-defined role với tên là **QLSP** có các quyền sau: thêm, xóa, sửa trên bảng **Production.Product**. Tạo 3 user ứng với 3 login trên, thực hiện thêm 3 user là thành viên của role **QLSP**.
4. Giả sử bạn muốn cấm 1 cách tường minh quyền thêm, xóa, sửa trên bảng **Production.Product** đối với user **PGD1**, cho dù user này là thành viên của role có các quyền trên (quyền thêm, xóa, sửa trên bảng **Production.Product**) thì user này cũng bị cấm. Các user khác không bị ảnh hưởng. Bạn thực hiện thế nào?
5. Ở câu 4 bạn đã cấm quyền thêm, xóa, sửa trên bảng **Production.Product** đối với user **PGD1**. Bạn muốn khôi phục lại quyền thêm, xóa, sửa trên bảng **Production.Product** đối với user **PGD1**. Bạn thực hiện thế nào?

6. Ở câu 3 bạn đã cấp quyền cho role **QLSP**: thêm, xóa, sửa trên bảng **Production.Product**. Bạn muốn cấm quyền thêm, xóa, sửa trên bảng **Production.Product** đối với role này. Bạn thực hiện thế nào? Các user là thành viên của role **QLSP** có các quyền gì ở lúc này?
7. Tạo hai login dạng SQL Server Authentication có tên là **NghiepVu1**, **NghiepVu2**. Tạo 2 user **NghiepVu1**, **NghiepVu2** ứng với 2 login trên, 2 user này có các quyền sau: xem và hiệu chỉnh cột ListPrice trong bảng Production.Product ; xem, hiệu chỉnh, xóa dữ liệu trong bảng Production.WorkOrder và Production.Product, chỉ được phép xem (Select) trên bảng Purchasing.WorkOrder.

## BÀI 5: ỨNG DỤNG VÀO BÀI TOÁN CỤ THỂ

### BÀI 5.1.

**Câu 1:** Sử dụng tài khoản với quyền quản trị thực hiện tạo login, user và cấp quyền

1.1 Tạo login tên admin1, mật khẩu Abc12345

1.2 Tạo user thuộc cơ sở dữ liệu master (Databases → System Databases → master)

1.3 Cấp quyền tạo cơ sở dữ liệu, tạo bảng và ~~quyền tạo login cho admin1~~

**Câu 2:** Sử dụng tài khoản admin1 và thực hiện các yêu cầu sau

#### 2.1 Tạo CSDL QuanLyNhanSu

File	Size	MaxSize	FileGrowth
Data	100	Không giới hạn	50
Log	300	Không giới hạn	100

#### 2.2 Tạo bảng NhanVien và LuongNV thuộc CSDL QuanLyNhanSu

##### Bảng NhanVien

Tên cột	Kiểu dữ liệu	Số kí tự	Ghi chú
MaNv	Varchar	20	Khóa chính
TenNv	Nvarchar	100	
NgaySinh	Varchar	10	
NoiSinh	nvarchar	50	

##### Bảng LuongNV

Tên cột	Kiểu dữ liệu	Số kí tự	Ghi chú
MaNv	varchar	20	Khóa chính
NamThang	varchar	7	
Luong	Float		

#### 2.3 Tạo login chứng thực SQL Server (SQL Server Authentication)

Tên login	Mật khẩu
LyNT	Abc12345



HungNT	Abc12345
--------	----------

## 2.4 Tạo user

Tên user	Tên login
LyNT	LyNT
HungNT	HungNT

## 2.5 Cấp quyền

Tên user	Tên bảng	Quyền được cấp
LyNT	NhanVien, LuongNV	Thêm, xóa dữ liệu
HungNT	NhanVien	Chỉ được phép xem MaNV, TenNV và cấp quyền Cập nhật dữ liệu

## 2.6. Kiểm tra cấp quyền

Thực hiện các lệnh sau với user LyNT và HungNT

- 1) Thêm vào bảng nhân viên dòng dữ liệu ('A01', 'Nguyễn Anh Linh', '1/2/88', 'TPHCM')
- 2) Xem thông tin bảng nhân viên
- 3) Sửa dữ liệu nơi sinh cho nhân viên này thành Hà Nội
- 4) Xóa nhân viên này khỏi bảng nhân viên

**Các lệnh trên có thực hiện được không? Lệnh nào không thực hiện được giải thích tại sao không thực thi được?**

- 5) Thêm quyền cập nhật dữ liệu cho bảng LuongNV cho user HungNT. Sau đó thực hiện lại các lệnh trên. Nhận xét.

## BÀI 5.2

### Câu 1: Tạo database tên QuanLyDaoTao

File	Size	MaxSize	FileGrowth
Data	100	Không giới hạn	10
Log	300	Không giới hạn	30

### Câu 2: Tạo các bảng thuộc CSDL QuanLyDaoTao

#### Bảng Lop

Tên cột	Kiểu dữ liệu	Số kí tự	Ghi chú
MaLop	Varchar	20	Khóa chính
TenLop	Nvarchar	100	

#### Bảng SinhVien

Tên cột	Kiểu dữ liệu	Số kí tự	Ghi chú
MaSv	Varchar	20	Khóa chính

TenSv	Nvarchar	100	
NgaySinh	Varchar	10	
NoiSinh	Nvarchar	50	
MaLop	Varchar	20	Khóa ngoại

**Bảng MonHoc**

Tên cột	Kiểu dữ liệu	Số kí tự	Ghi chú
MaMh	Varchar	20	Khóa chính
TenMh	Nvarchar	100	
SoGio	Int		

**Bảng DiemTP**

Tên cột	Kiểu dữ liệu	Số kí tự	Ghi chú
MaSv	Varchar	20	Khóa chính
MaMh	Varchar	20	
Diem	Float		

**Câu 3:** Thiết lập ràng buộc dữ liệu

**Câu 4:** Thêm dữ liệu cho các bảng

**Bảng Lop**

MaLop	TenLop
CN0201	Khóa 2001
CN0202	Khóa 2002

**Bảng SinhVien**

MaSv	TenSv	NgaySinh(dd/mm/yyyy)	NoiSinh	MaLop
sv01	Nguyễn Văn Hưng	12/02/1988	Hồ Chí Minh	CN0201
sv02	Lê Hùng	17/03/1990	Bình Dương	CN0201
sv03	Lê Hùng	02/12/1991	Bình Dương	CN0202

**Bảng MonHoc**

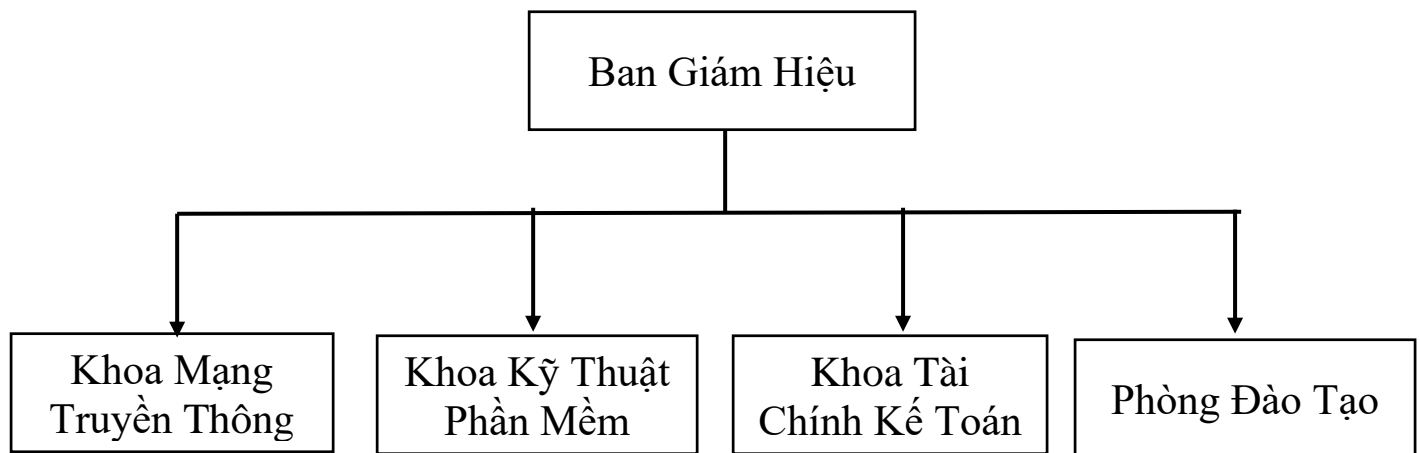
MaMh	TenMh	SoGio
THVP	Tin học văn phòng	45
THDC	Tin học đại cương	45
CSDL	Cơ sở dữ liệu	30

**Bảng DiemTP**

MaSv	MaMh	Diem
sv01	THVP	8.0
sv01	THDC	7.0
sv01	CSDL	6.0

sv02	THVP	9.0
sv02	THDC	4.0
sv02	CSDL	7.0
sv03	THVP	5.0
sv03	THDC	5.0
sv03	CSDL	5.0

### CÂU 6: CẤP QUYỀN



6.1 Tạo các nhóm quyền thuộc CSDL QuanLyDaoTao tương ứng với các phòng ban như mô tả trên

6.2 Cấp quyền cho các phòng ban trên CSDL QuanLyDaoTao

1. Ban Giám Hiệu được phép xem tất dữ liệu tất cả các bảng của database
2. Khoa Mạng Truyền Thông được phép xem trên bảng SINHVIEN, LOP
3. Khoa Kỹ thuật phần mềm được phép xem, thêm, xóa và cập nhật dữ liệu trên bảng SINHVIEN, LOP, MONHOC
4. Khoa Tài chính kế toán chỉ được phép xem dữ liệu trên bảng DiemTP
5. Phòng Đào tạo được phép xem, thêm, xóa và cập nhật dữ liệu trên tất cả các bảng của database

6.3. Tạo danh sách nhân sự cho các phòng ban (**Lưu ý tất cả tài khoản phải đổi mật khẩu trong lần đăng nhập đầu tiên**)

#### Ban Giám Hiệu

Tên	Mật khẩu
AnhNH	Abc12345
HoangNT	Abc12345

### Khoa Mạng Truyền Thông

Tên	Mật khẩu
TrungDM	Abc12345
CongND	Abc12345
DangNS	Abc12345

### Khoa Kỹ thuật phần mềm

Tên	Mật khẩu
ThuatDV	Abc12345
DatDT	Abc12345
NguyenTT	Abc12345

### Khoa Tài chính kế toán

Tên	Mật khẩu
TuanTV	Abc12345
DieuNT	Abc12345
GiangNN	Abc12345

### Phòng Đào tạo

Tên	Mật khẩu
TramNTH	Abc12345
ThuyLT	Abc12345

5.4 Chọn một thành viên đại diện cho mỗi phòng ban, bạn thực hiện các lệnh insert, update, delete, select cho mỗi table, giải thích từng lệnh mà bạn đã thực hiện được.

## BÀI 6: PHÂN QUYỀN THEO VAI TRÒ

### Lý thuyết

**Phân quyền theo Role là gì và làm như thế nào ?**

- Phân quyền theo **Role** là cách gọi chung của mình cho việc bạn nhóm nhiều thành viên trong 1 tổ chức có cùng một quyền hạn thực thi công việc. Lúc đó ta có thể nhóm họ vào 1 group để dễ dàng trao quyền hạn.

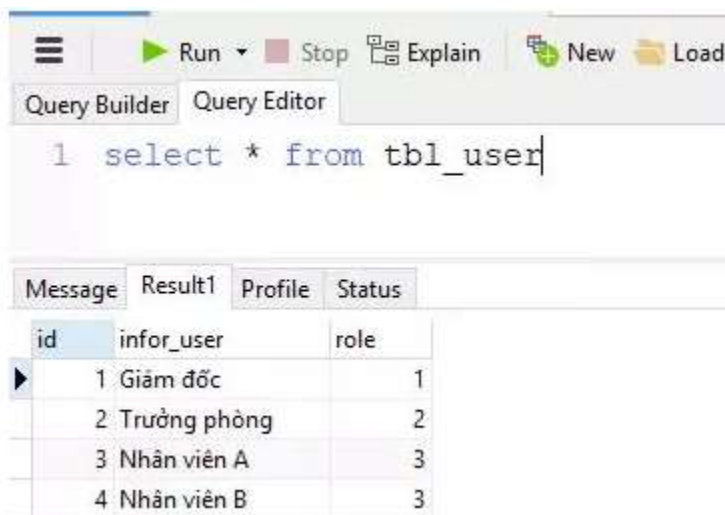
Ví dụ

#### 1. Phân quyền theo cấp bậc

- Loại hình này chúng ta thường thiết kế db đơn giản như sau



- Khi đó dữ liệu bạn dùng sẽ có dạng như thế này.



- role ở đây là 1, 2, 3 tức là có 3 mức quyền hạn và lớn nhất hay bé nhất còn tùy thuộc vào quy định của mỗi công ty. Ví dụ

```

1 declare @role tinyint
2 set @role = (select role from tbl_user where id = 1)
3
4 if @role = 1
5     -- Do some thing
6     print N'Bạn có quyền admin'
7 else if @role = 2
8     print N'Bạn có quyền trưởng phòng'
9 else
10    print N'Bạn chỉ là nhân viên thường thôi'
    
```

143 %

Messages

Bạn có quyền admin

### - Ưu điểm

Việc sử dụng kiểu phân quyền này dễ dàng cho những người mới bắt đầu. Những nhóm quyền được lập lên nhanh chóng có thể sử dụng luôn, Và việc phải check cũng tương đối là đơn giản, bạn chỉ cần 1 câu

```
select count(*) from tbl_... where id = ? and role = ?
```

hoặc




```
select role from tbl_... where id = ?
```

### - Nhược điểm

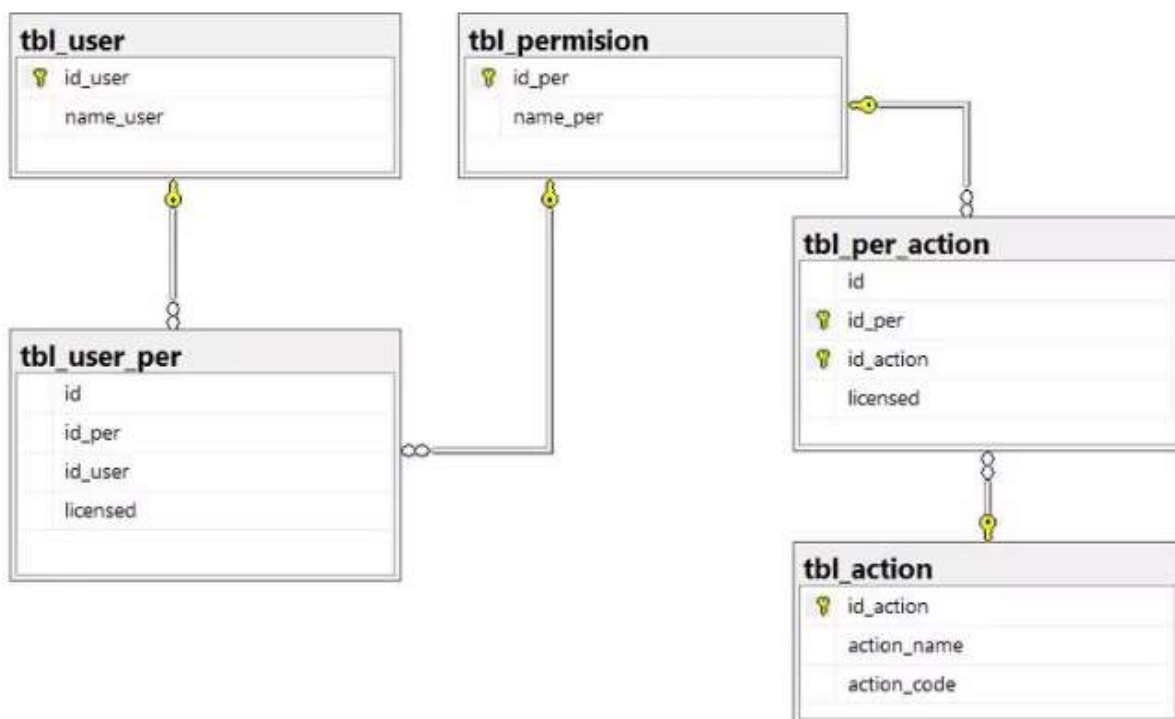
- Rất khó có thể mở rộng dự án
- Trong thực tế không phải lúc nào cũng có 3 role. Nó có thể phát sinh nhiều role kì dị.  
Ví dụ: **Thư ký giám đốc ngoài quyền đuổi việc ra còn lại nó sẽ có quyền của giám đốc** vậy trường hợp này thuộc role 1 hay 2 ?.
- Rất khó để phân quyền chi tiết.

## 2. Phân quyền theo chức năng

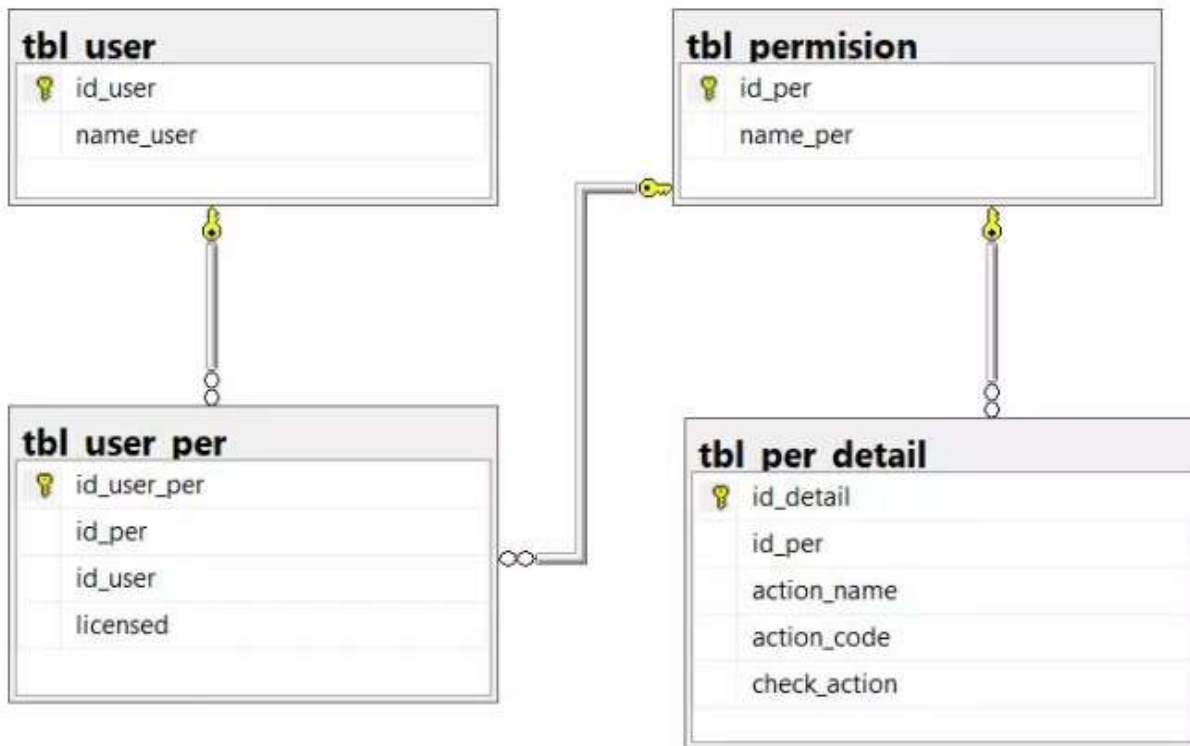
Loại phân quyền này được sử dụng rất nhiều trong thực tế. Nó rất hiệu quả và dễ thao tác đối với người cấp quyền.

Command Permission	Create	Copy	Paste	Cut	Move	Rename	Delete
 Full Access	✓	✓	✓	✓	✓	✓	✓
 Read	✗	✓	✗	✗	✗	✗	✗
 No Access	✗	✗	✗	✗	✗	✗	✗

Ta thiết kế db đơn giản trong ví dụ này như sau:



Nhưng để dễ thực hiện chúng ta sẽ tóm gọn 2 bảng **tbl\_action** và **tbl\_per\_action** thành bảng **tbl\_per\_detail** để dễ thao tác. Và ta có một Database như sau.



Chi tiết của việc thiết kế DB như sau:

- **tbl\_user**: bảng lưu người dùng bao gồm các thuộc tính như ID, Name,... Bảng không có khóa ngoại.
- **tbl\_permission**: bảng chứa nhóm quyền hạn. bao gồm các thuộc tính, ID nhóm quyền hạn, tên nhóm quyền hạn.
- **tbl\_permission\_detail**: là bảng sẽ chứa những quyền hạn cụ thể dành cho nhóm quyền hạn. Trường **action\_name** không cần thiết bạn có thể bỏ. Trường **action\_code** là để khi lập trình mình định nghĩa một thao tác nhất định bằng code, ví dụ quyền sửa thì code nó là **EDIT** chẳng hạn.
- **tbl\_per\_relationship**: là bảng lưu mối liên hệ giữa người dùng và nhóm quyền hạn. Mục đích của bảng này không phải là để một người dùng có nhiều nhóm quyền mà để không phải truy vấn lại bảng user chứa thông tin nhạy cảm như username và password. Bạn cũng có thể **bỏ qua** bảng này và liên hệ trực tiếp giữa bảng user và permission luôn, nhưng mình khuyên bạn nên sử dụng thêm bảng này vì có nhiều trường hợp **user có nhiều quyền hạn**.

## 1. Kiểm tra dữ liệu trong các bảng

- **tbl\_user**



SQLQuery2.sql - TUA...viblo\_test (sa (57))\* X SQLQuery1.sql - TUA...viblo\_test (sa (

```
1 select * from tbl_user
```

143 %

Results Messages

	id_user	name_user
1	1	Tuân Phạm
2	2	Tôi là ai
3	3	Ahihi

- **tbl\_permission**

SQLQuery2.sql - TUA...viblo\_test (sa (57))\* X SQLQuery1.sql - TUA...viblo\_test (sa (

```
1 select * from tbl_permission
```

143 %

Results Messages

	id_per	name_per
1	1	Full
2	2	Admin
3	3	Read only
4	4	Edit
5	5	Create

- **tbl\_per\_detail**

SQLQuery2.sql - TUA...viblo\_test (sa (57))\* SQLQuery1.sql - TUA...viblo\_test (sa (59))\*

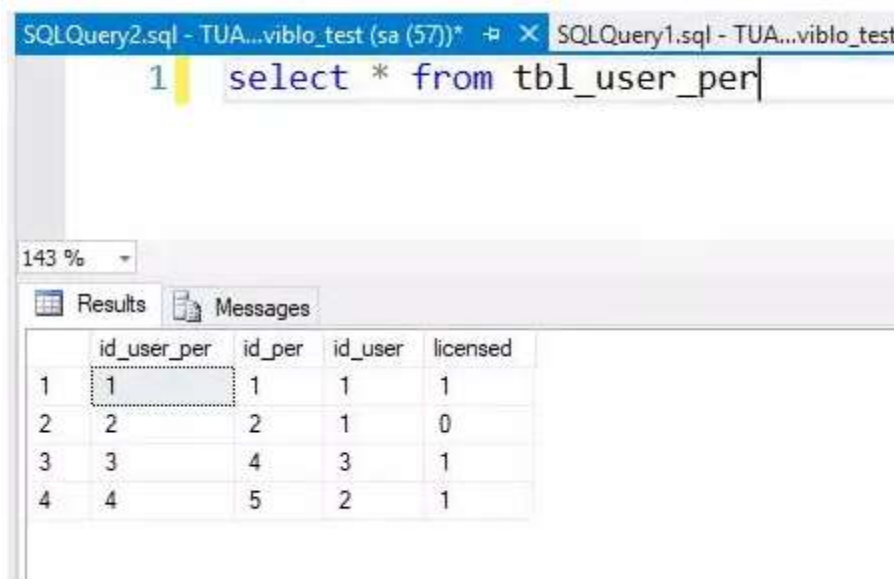
```
1 select * from tbl_per_detail
```

143 %

Results Messages

	id_detail	id_per	action_name	action_code	check_action
1	1	1	Create post	CREATE	1
2	2	1	Edit post	EDIT	1
3	3	1	Delete post	DELETE	1
4	4	1	View post	VIEW	1
5	5	2	Create post	CREATE	1
6	6	2	Edit post	EDIT	1
7	7	2	Delete post	DELETE	0
8	8	2	View post	VIEW	1
9	9	3	Create post	CREATE	0
10	10	3	Edit post	EDIT	0
11	11	3	Delete post	DELETE	0
12	12	3	View post	VIEW	1
13	13	4	Create post	CREATE	0
14	14	4	Edit post	EDIT	1
15	15	4	Delete post	DELETE	0
16	16	4	View post	VIEW	1
17	17	5	Create post	CREATE	1
18	18	5	Edit post	EDIT	0

- **tbl\_user\_per**



## 2. Làm một số ví dụ

- Kiểm tra quyền của người dùng ví dụ: Hãy kiểm tra quyền của user có id là 1:

```
DECLARE @result NVARCHAR(1000)
```

```
SET @result = N'Những quyền hiện tại của user ('
```

```
select @result = @result + name_user + ') là: ' from tbl_user where id_user = 1
```

```
select @result = @result + action_name + ', ' from tbl_user as u
```

```
    join tbl_user_per as up on u.id_user = up.id_user
```

```
    join tbl_permission as p on up.id_per = p.id_per
```

```
    join tbl_per_detail as pd on p.id_per = pd.id_per
```

```
    where u.id_user = 1 and up.licensed = 1 and pd.check_action = 1
```

```
select @result = substring(@result, 0, len(@result))
```

```
print @result
```

- kết quả

```

1 DECLARE @result NVARCHAR(1000)
2 SET @result = N'Những quyền hiện tại của user ('
3
4 select @result = @result + name_user + ') là: ' from tbl_user where id_user = 1
5 select @result = @result + action_name + ', ' from tbl_user as u
6     join tbl_user_per as up on u.id_user = up.id_user
7     join tbl_permission as p on up.id_per = p.id_per
8     join tbl_per_detail as pd on p.id_per = pd.id_per
9     where u.id_user = 1 and up.licensed = 1
10 select @result = substring(@result, 0, len(@result))
11
12 print @result
    
```

Messages

Những quyền hiện tại của user (Tuân Phạm) là: Create post, Edit post, Delete post, View post

- Kiểm tra xem user 2 có quyền xóa bài viết không ?

**DECLARE** @result bit

**select** @result = check\_action **from** tbl\_user **as** u

**join** tbl\_user\_per **as** up **on** u.id\_user = up.id\_user

**join** tbl\_permission **as** p **on** up.id\_per = p.id\_per

**join** tbl\_per\_detail **as** pd **on** p.id\_per = pd.id\_per

**where** u.id\_user = 2 **and** up.licensed = 1 **and** action\_code = 'DELETE'

**begin**

**if** @result = 1

**print** N'Bạn CÓ quyền xóa post'

**else**

**print** N'Bạn KHÔNG có quyền xóa post'

end

- kết quả

```

1 DECLARE @result bit
2 select @result = check_action from tbl_user as u
3     join tbl_user_per as up on u.id_user = up.id_user
4     join tbl_permission as p on up.id_per = p.id_per
5     join tbl_per_detail as pd on p.id_per = pd.id_per
6     where u.id_user = 2 and up.licensed = 1 and action_code = 'DELETE'
7
8 begin
9     if @result = 1
10        print N'Bạn CÓ quyền xóa post'
11    else
12        print N'Bạn KHÔNG có quyền xóa post'
13 end
    
```

Messages

Bạn KHÔNG có quyền xóa post

- Đó là một số ví dụ đơn giản tron tình huống này khi sử dụng phân quyền theo nhóm (group)

### 3. Kết luận

#### - Ưu điểm

- Việc phân quyền này rất dễ thao tác đối với những admin khi họ muốn chuyển nhóm quyền hoặc thỏa sức sáng tạo trong việc tạo ra những quyền mới từ những quyền ban đầu. Ví dụ như người dùng vừa có thể **EDIT** và **DELETE**,....
- Ngoài ra việc thực hiện những câu query cũng rất dễ dàng cho những lập trình viên.

#### - Nhược điểm

- Vấn đề sử dụng quyền hành động rất dễ khi chúng ta làm việc trên 1 group, nhưng nếu trong chương trình của bạn có nhiều group và phân cấp nhiều tầng thì nó lại là một vấn đề nan giải khác, khi bạn không chỉ phải check quyền hành động mà bạn còn phải check xem quyền hành động này của người dùng có thể áp dụng được trong group khác hay không?..

### 3. Phân quyền theo *Hành động* của các *nhóm Group* theo những *cấp bậc* khác nhau

- Đây là loại phân quyền phức tạp nhất nhưng lại là quan trọng nhất, bởi các lý do sau đây:
- Các tổ chức sử dụng phần mềm để thực hiện thao tác của họ đều có phân cấp rõ ràng
- Trong những tổ chức có những người nắm full quyền của nhiều nhóm
- Có những thành viên thuộc nhiều nhóm
- Có những thành viên tuy chỉ là nhân viên nhưng lại có quyền của các sếp (thư ký)
- Chính vì có nhiều trường hợp như vậy những lập trình viên sinh ra được rất nhiều case trong code.

Giải quyết vấn đề này bạn có thể tìm hiểu cách thực hiện phân quyền trong odoo.

- Phân quyền theo model: Người dùng được thao tác thực hiện với những bảng dữ liệu nào

Ví dụ: Admin có thể thực hiện với thao tác với bảng user của họ

- phân quyền theo raw: Người dùng được thực hiện việc thao tác với các raw được chỉ định

Ví dụ: Leader A chỉ có thể thực hiện thao tác với những thành viên của mình trong bảng user

- phân quyền theo column: Người dùng sẽ được quyền thao tác với những column đó

Ví dụ: Chỉ giám đốc mới có thể đuổi việc nhân viên, ở đây ta sẽ có 1 column tên là `is_working` để biết việc nhân viên đó còn đi làm hay không

```

4  --• Quản lý xem các báo cáo thống kê: gồm quản lý QL, thuộc role db_datareader.
5  --Thực hiện các yêu cầu sau:
6  --a.   Tạo các login; tạo các user khai thác CSDL AdventureWorks2008R2 cho các nhân viên và quản lý
7  --(tên login trùng tên user). (1đ)
8
9  ---a
10 create login TN with password='123'
11 go
12 create login NV with password='123'
13 go
14 create login QL with password='123'
15 ---
16 create user TN for login TN
17 go
18 create user NV for login NV
19 go
20 create user QL for login QL
    
```

```

22
23 --b.Phân quyền: -Admin: chỉ cấp quyền cần thiết cho TN và QL; từ chối cấp quyền sửa cho NV.
24 ---Trưởng nhóm TN: chuyển tiếp tất cả các quyền mình có cho NV (1đ)
25 --Các nhân viên sửa, xóa và xem số liệu: gồm trưởng nhóm TN và nhân viên NV, chỉ được làm
26 --việc trên bảng EmployeePayHistory.
27 use AdventureWorks2008R2
28
29 alter role [db_datareader] add member QL
30
31 go
32 grant update,delete,select
33 on [HumanResources].[EmployeePayHistory]
34 to TN with grant option
35 go
36

```

Login bằng tài khoản TN

```

1 use AdventureWorks2008R2
2 select * from HumanResources.EmployeePayHistory
3
4 ---Trưởng nhóm TN: chuyển tiếp tất cả các quyền mình có cho NV (1đ)
5 grant update,delete,select
6 on HumanResources.EmployeePayHistory
7 to NV

```

## BÀI TẬP THỰC HÀNH

- 1) Dựa vào phần lý thuyết bạn hãy thực hiện 1 ví dụ phân quyền cho người dùng truy cập vào các bảng của 1 cơ sở dữ liệu do bạn tự xây dựng.
- 2) Xây dựng 1 Website ứng dụng cách phân quyền trên vào Form đăng nhập vào Website và tùy theo người dùng đăng nhập sẽ hiện các chức năng tương ứng với vai trò của mỗi người dùng.



## Bài Thực Hành Tuần 4

### AUDTING

#### Mục tiêu:

- Kiểm soát quá trình hoạt động của một CSDL

## PHẦN 1: LÝ THUYẾT

### CÂU HỎI

- 1) Xác định mục đích của giám sát nội bộ (internal audit) và giám sát bên ngoài (external audit).
2. Xác định những người có thể được đưa vào một nhóm giám sát cho cả chính thức và giám sát không chính thức.
3. Giải thích mục tiêu của giám sát viên. Cho ví dụ minh họa.
4. Liệt kê các tài liệu có thể sẽ được xem xét trong giai đoạn lập kế hoạch và chuẩn bị của một giám sát bên ngoài chính thức.
6. Liệt kê các phần thông tin tiêu biểu có trong báo cáo giám sát.
7. Xác định các tài liệu có thể sẽ được xem xét trong giai đoạn lập kế hoạch và chuẩn bị của giám sát không chính thức một cơ sở dữ liệu cụ thể.
8. Liệt kê các thành phần hỗ trợ cơ sở dữ liệu sẽ yêu cầu giám sát để đảm bảo độ tin cậy của kho dữ liệu.
9. Xác định và giải thích các lĩnh vực khác nhau cho giám sát bảo mật cơ sở dữ liệu.
10. Giải thích mục đích của Aud\_trail trong Oracle.
11. Mô tả sự khác biệt giữa giám sát cấp cơ sở dữ liệu và giám sát cấp ứng dụng.

### BÀI TẬP NHÓM (Sinh viên làm nhóm và thuyết trình)

#### A. Tình huống 1: Tổ chức kiểm toán

Sử dụng Internet. Tìm và mô tả ít nhất một công ty mà việc giám sát bảo mật sẽ được yêu cầu phải tuân thủ quy định của tổ chức.

#### B. Tình huống 2: Giám sát nội bộ

Cung cấp danh sách giám sát nội bộ mà trường học hoặc công ty hiện tại của bạn đã thực hiện.

#### C. Tình huống 3: Kiểm toán cơ sở dữ liệu

Sử dụng Internet. Tìm và mô tả một công cụ tự động hỗ trợ giám sát cơ sở dữ liệu (Không nên cài đặt công cụ.)



D. Tình huống 4: Kiểm toán Oracle

Sử dụng Internet. Xác định các bước để tạo một kiểm toán tùy chỉnh trong SQL Server và Oracle.

E. Tình huống 5: Kiểm toán MySQL

Sử dụng Internet. Xác định ứng dụng của bên thứ ba có thể hỗ trợ giám sát cơ sở dữ liệu MySQL.

F. Tình huống 6: Giám sát trong Microsoft SQL Server

Sử dụng trang web SQL Server [www.microsoft.com/sqlserver/2008/en/us/](http://www.microsoft.com/sqlserver/2008/en/us/). Xác định các bước để tạo kiểm toán tùy chỉnh cho Microsoft SQL Server.

## PHẦN II: THỰC HÀNH

### BÀI 1

1. Tạo cơ sở dữ liệu QLBH, các tham số tùy ý
2. Tạo bảng SanPham (MaSP int identity(1,1) primary key, TenSP nvarchar(25), DonGia int check (DonGia >= 0), SLTK int default(0))
3. Nhập dữ liệu vào cho bảng SanPham, sinh viên thêm khoảng 5 sản phẩm có giá trị tùy ý.
4. Quản trị viên hệ thống muốn giám sát (Audit) hành động đọc (select) và thêm mới (Insert) dữ liệu trong bảng SanPham. Bạn hãy viết mã lệnh T\_SQL thực hiện yêu cầu giám sát trên.
5. Quản trị viên hệ thống thấy rằng “Hành động đơn giá giảm từ 30% trở lên so với giá ban đầu là hành động đáng ngờ”. Bạn hãy viết mã lệnh Trigger ghi nhận lại những hành động đáng nghi ngờ này.
6. Quản trị viên hệ thống muốn giám sát hành động cập nhật (update), xóa (delete) dữ liệu trong bảng sản phẩm. Bạn hãy viết mã lệnh T-SQL để thực hiện yêu cầu giám sát trên.
7. Quản trị viên hệ thống thấy rằng:”hành động cập nhật số lượng sản phẩm tăng từ 100 trở lên là hành động đáng nghi ngờ”. Bạn hãy viết mã lệnh Trigger ghi nhận lại những hành động đáng nghi ngờ này.

Chú ý: Bài làm phải chứa các đoạn code kiểm thử cho các hành động trong trường hợp:

- hệ thống có giám sát.
- hệ thống không có giám sát.

### BÀI 2

1. Tạo cơ sở dữ liệu QLBH, các tham số tùy ý
2. Tạo bảng SanPham (MaSP int identity(1,1) primary key,

TenSP nvarchar(25),  
 DonGia int check (DonGia >= 0),  
 SLTK int default(0))

3. Nhập dữ liệu vào cho bảng SanPham, sinh viên thêm khoảng 5 sản phẩm có giá trị tùy ý.
4. Tạo 1 trigger giám sát khi người dùng thay đổi đơn giá của bảng sản phẩm. Viết lệnh kiểm tra giám sát vừa thực hiện. Dữ liệu giám sát gồm Masp, TenSp, DonGiaCu, DonGiaMoi, câu lệnh thực hiện, ai thực hiện
5. Sửa lại Trigger của câu 4 chỉ giám sát khi thay đổi giá mới lớn hơn hay bằng 30% giá cũ, Kiểm tra giám sát vừa thực hiện. Dữ liệu giám sát gồm Masp, TenSp, DonGiaCu, DonGiaMoi, câu lệnh thực hiện, ai thực hiện.
6. Tạo Login Hai pass =HAI. Tạo người dùng tên HAI. Cấp quyền cho người dùng này được phép xem, thêm, xoá, sửa. Đăng nhập vào login HAI, thực hiện lệnh Update thay đổi Dongia theo trường hợp câu 4, câu 5. Cho biết kết quả giám sát.
7. Viết lệnh tạo 1 trigger giám sát cho các lệnh thêm trên bảng SanPham. Việc giám sát sẽ gồm ngày giờ thực hiện lệnh, lệnh gì, ai thực hiện, dữ liệu mới thêm là gì.
8. Viết lệnh tạo 1 trigger giám sát cho các lệnh xoá trên bảng SanPham. Việc giám sát sẽ gồm ngày giờ thực hiện lệnh, lệnh gì, ai thực hiện, dữ liệu bị xoá là gì.
9. Viết lệnh tạo 1 trigger giám sát cho các lệnh xem trên bảng SanPham. Việc giám sát sẽ gồm ngày giờ thực hiện lệnh, câu lệnh, ai thực hiện, dữ liệu xem là gì.
10. Đăng nhập vào login HAI, thực hiện lệnh Thêm, Xoá, Xem. Cho biết kết quả giám sát.

### BÀI 3

Thực hiện tạo giám sát sự đăng nhập thông qua window application log theo các lệnh sau

1. Tạo Audit server (lưu file trong application)
 

```
CREATE SERVER AUDIT KiemTraDoiTuong
TO FILE(FILEPATH='T:\BMCSDL\AUdit\AuditFile') /* substitute in here network
drive */
WITH (ON_FAILURE=FAIL_OPERATION, QUEUE_DELAY=0);
```
2. Bật lên (Enable) Audit Server
 

```
ALTER SERVER AUDIT KiemTraDoiTuong WITH (STATE=ON);
```
3. Tạo Server Specifacatetionc
 

```
CREATE SERVER AUDIT SPECIFICATION ThucThiKiemTraDoiTuong FOR
SERVER AUDIT KiemTraDoiTuong
add (AUDIT_CHANGE_GROUP)
```
4. Bật lên Server Specifacatetion

```
ALTER SERVER AUDIT SPECIFICATION ThucThiKiemTraDoiTuong WITH (STATE=ON);
```

5. Thay đổi đường dẫn

```
ALTER SERVER AUDIT KiemTraDoiTuong WITH (STATE=OFF);
ALTER SERVER AUDIT KiemTraDoiTuong TO FILE(FILEPATH='D:\Audit');
ALTER SERVER AUDIT KiemTraDoiTuong WITH (STATE=ON);
```

6. Kiểm tra (Test )

drop server audit Giam\_Sat\_Tong //thử xóa hay tạo audit thì sẽ ghi lại

7. Truy cập file

```
SELECT * FROM sys.server_file_audits
SELECT * FROM sys.fn_get_audit_file('D:\Audit\*', NULL, NULL);
SELECT * FROM sys.dm_server_audit_status
```

Results

Messages

	audit_id	name	audit_guid	create_date	modify_date	principal_id	type	type_desc	on_failure	on_failure_desc	is
1	65540	KiemTraTrongDB	F22F2276-6A30-45C7-A344-5A0AD9354994	2015-11-25 14:27:21.607	2015-11-25 14:27:21.607	1	FL	FILE	2	FAIL OPERATION	0
2	65541	KiemTraTao_Them	A13EF30D-87D2-40BB-918D-F6438D691F4E	2015-11-25 14:59:05.457	2015-11-25 14:59:05.457	1	FL	FILE	2	FAIL OPERATION	0
3	65542	KiemTraTao_Them_for_User	E28C26AF-D472-4F28-A1FE-04F631243D64	2015-11-25 15:26:13.350	2015-11-25 15:26:13.350	1	FL	FILE	2	FAIL OPERATION	0
4	65543	KiemTraDoiTuong	49708D7C-F798-49CC-8872-59D9180EFEE1	2015-11-26 03:49:27.380	2015-11-26 03:49:27.380	1	FL	FILE	2	FAIL OPERATION	1

11

	local_sid	target_database_principal_name	server_instance_name	database_name	schema_name	object_name	statement	additional_information
1			HOAI-PC					<action_info xmlns="http://schemas.microsoft.com
2			HOAI-PC	master		Giam_Sat_Tong	drop server audit Giam_Sat_Tong	

11

	audit_id	name	status	status_desc	status_time	event_session_address	audit_file_path	audit_file_size
1	65543	KiemTraDoiTuong	1	STARTED	2015-11-26 03:51:00.3440000	0x00000001F2283871	D:\Audit\KiemTraDoiTuong_49708D7C-F798-49CC-8872...	7168

Quan sát kết quả và cho nhận xét

## BÀI 4

### Câu 1:

Tạo giám sát về sự thay đổi dữ liệu trong một bảng nào đó (lưu trong file và đọc từ file ra )

1. Tạo bảng NguoiLaoDong(maNLD char(10) primary key, hoTen nvarchar(50))

2. Tạo audit sever

```
CREATE SERVER AUDIT KiemTraTao_Them
```

TO FILE(FILEPATH='D:\Audit1') /\* substitute in here network drive \*/  
WITH (ON\_FAILURE=FAIL\_OPERATION, QUEUE\_DELAY=0);

3. Enable

ALTER SERVER AUDIT KiemTraTao\_Them WITH (STATE=ON);

4. Tạo Database audit specificate

CREATE DATABASE AUDIT SPECIFICATION KiemTraTao\_Them  
FOR SERVER AUDIT KiemTraTao\_Them  
ADD (SELECT , INSERTON [dbo].[NguoiLaoDong] BY dbo )  
WITH (STATE = ON) ;  
GO

5. Kiểm tra (Test)

select \* from NguoiLaoDong

Insert into NguoiLaoDong(maNLD,hoTen) values ('NLD5000','Hoai-Yen')

Insert into NguoiLaoDong(maNLD,hoTen) values ('NLD6000','Hoai-Yen sua cua user')

6. Đọc file

SELECT \* FROM sys.dm\_server\_audit\_status

SELECT \* FROM sys.fn\_get\_audit\_file('D:\Audit1\\*', NULL, NULL);

SELECT \* FROM sys.dm\_server\_audit\_status

Results Messages

	audit_id	name	audit_guid	create_date	modify_date	principal_id	type	type_desc	on_failure	on_failure_desc	is
1	65540	KiemTraTrongDB	F22F2276-6A30-45C7-A344-5A0AD9354994	2015-11-25 14:27:21.607	2015-11-25 14:27:21.607	1	FL	FILE	2	FAILURE OPERATION	0
2	65541	KiemTraTao_Them	A13EF30D-37D2-40BB-918D-F643BD691F4E	2015-11-25 14:59:05.457	2015-11-25 14:59:05.457	1	FL	FILE	2	FAILURE OPERATION	0
3	65542	KiemTraTao_Them_for_User	E28C26AF-D472-4F28-A1FE-04F631243D64	2015-11-25 15:26:13.350	2015-11-25 15:26:13.350	1	FL	FILE	2	FAILURE OPERATION	0
4	65543	KiemTraDoiTuong	49708D7C-F798-49CC-8872-59D918DEFE1	2015-11-26 03:49:27.380	2015-11-26 03:49:27.380	1	FL	FILE	2	FAILURE OPERATION	1

11

principal_id	target_database_principal_name	server_instance_name	database_name	schema_name	object_name	statement	additional_information
1		HOAI-PC					<action_info xmlns="http://schemas.microsoft.com
2		HOAI-PC	master		Giam_Sat_Tong	drop server audit Giam_Sat_Tong	

11

audit_id	name	status	status_desc	status_time	event_session_address	audit_file_path	audit_file_size
1	65543	KiemTraDoiTuong	1	STARTED	2015-11-26 03:51:00.3440000	0x00000001F2283871	D:\Audit\KiemTraDoiTuong_49708D7C-F798-49CC-8872_7168

Đọc kết quả và cho nhận xét.

**Câu 2:** Tạo giám sát về sự thay đổi của bảng [Order Detail] trong cơ sở dữ liệu Northwind khi thực hiện các lệnh Insert, Update, Delete, Select.

**Câu 3:**

- 1) Tạo bảng ACCOUNTS thuộc schema của user ACCMASTER

ACCNO	ACCNAME	BAL
-----	-----	-----
1	Alex	10000
2	Bill	15000
3	Charlie	20000
4	David	25000

- 2) Hiện thực chính sách: giám sát khi một user nào đó truy xuất vào bảng ACCOUNTS và xem số dư lớn hơn hoặc bằng 20000.

**Câu 4:**

- 1) Tạo user mới với username là TenBan. Phân quyền create table và create procedure cho user vừa mới tạo.
- 2) Thực hiện giám sát các hành vi xem, thêm, sửa, xóa dòng trên bất kì bảng nào của user TenBan.
- 3) Đăng nhập vào tài khoản user TenBan. Thực hiện chuỗi hành động sau
  1. Tạo một bảng KHACHHANG (MaKH int, TenKH nvarchar(40), Pass nchar(10))
  2. Nhập vào 1 dòng dữ liệu bất kỳ.
  3. Update giá trị vừa insert vào.
  4. Xem tất cả dữ liệu của bảng KHACHHANG.
  5. Xóa tất cả dữ liệu trong bảng KHACHHANG.
  6. Xóa bảng KHACHHANG.
- 4) Đăng nhập vào user system, kiểm tra những hành vi nào được giám sát lại. Hành vi tạo bảng và xóa bảng của user TenBan có bị giám sát không? Nếu có hãy giải thích lý do, nếu không hãy tạo câu lệnh giám sát hành vi tạo bảng và xóa bảng của user TenBan.

**BÀI 5: ỨNG DỤNG VÀO PROJECT (Sinh viên làm nhóm và thuyết trình)**

Tạo và triển khai giám sát: Bạn đã được thuê làm giám sát viên chính trong công ty của riêng bạn. Bạn là người tạo và thực hiện một lịch trình giám sát bên trong cơ sở dữ liệu cho tổ chức. Thực hiện các yêu cầu sau:

1. Tạo một bảng bao gồm lịch giám sát trong 12 tháng. Bao gồm các cột xác định ước tính thời gian cho mỗi cuộc giám sát được liệt kê.
2. Tạo một kế hoạch và liệt kê danh sách tất cả các thành phần trong công ty phải giám sát.
3. Lập kế hoạch chi tiết và chuẩn bị những mục cần thiết cho mỗi giám sát.
4. Xác định phạm vi cho mỗi giám sát và xác định những giám sát đặc biệt nào

cần được giải quyết.

5. Tạo một danh sách ít nhất năm hoạt động giám sát cho mỗi lần giám sát.
6. Mô tả giám sát nào chỉ có giải quyết ở Oracle.
7. Mô tả giám sát nào chỉ có giải quyết ở MySQL
8. Mô tả giám sát nào chỉ có giải quyết ở SQL Server

## Bài Thực Hành Tuần 5

### MÃ HÓA VÀ GIẢI MÃ

#### Mục tiêu:

- Thực hiện được mã hóa và giải mã trong SQL Server

#### BÀI 1

Thực hiện mã hóa và giải mã theo các lệnh sau, bạn hãy giải thích ý nghĩa, chức năng của từng lệnh đã thực hiện

Encryption

USE master

GO

CREATE DATABASE EncryptTest

go

USE EncryptTest

GO

CREATE TABLE TestTable (FirstCol INT, SecondCol VARBINARY(256))

go

/\* Create Database Master Key \*/

CREATE MASTER KEY ENCRYPTION

BY PASSWORD = 'SQLAuthority'

GO

/\* Create Encryption Certificate \*/

CREATE CERTIFICATE EncryptTestCert

WITH SUBJECT = 'SQLAuthority'

GO

/\* Create Symmetric Key \*/

CREATE SYMMETRIC KEY TestTableKey

WITH ALGORITHM = TRIPLE\_DES ENCRYPTION

```
BY CERTIFICATE EncryptTestCert
GO
```

```
OPEN SYMMETRIC KEY TestTableKey DECRYPTION BY CERTIFICATE
EncryptTestCert
GO
--UPDATE TestTable
--SET EncryptSecondCol = ENCRYPTBYKEY(KEY_GUID('TestTableKey'),SecondCol)
--GO
INSERT                                INTO                                TestTable
values(1,ENCRYPTBYKEY(KEY_GUID('TestTableKey'),'Hello'))
INSERT                                INTO                                TestTable
values(2,ENCRYPTBYKEY(KEY_GUID('TestTableKey'),'123456'))
INSERT                                INTO                                TestTable
values(3,ENCRYPTBYKEY(KEY_GUID('TestTableKey'),'gogogo'))
go
SELECT * FROM TestTable
GO
```

```
/* Decrypt the data of the SecondCol */
```

```
OPEN SYMMETRIC KEY TestTableKey DECRYPTION BY CERTIFICATE
EncryptTestCert
SELECT      CONVERT(VARCHAR(50),DECRYPTBYKEY(SecondCol))      AS
DecryptSecondCol
FROM TestTable
GO
CLOSE SYMMETRIC KEY TestTableKey
GO
```

## BÀI 2

Thực hiện mã hóa và giải mã theo các lệnh sau, bạn hãy giải thích ý nghĩa, chức năng của từng lệnh đã thực hiện

```
/* Create Database */
USE master
GO
```



```
CREATE DATABASE EncryptTest
ON PRIMARY ( NAME = N'EncryptTest', FILENAME = N'C:\EncryptTest.mdf')
LOG ON ( NAME = N'EncryptTest_log', FILENAME = N'C:\EncryptTest_log.ldf')
GO
```

**/\* Create table and insert data in the table \*/**

```
USE EncryptTest
```

```
GO
```

```
CREATE TABLE TestTable (FirstCol INT, SecondCol VARCHAR(50))
```

```
GO
```

```
INSERT INTO TestTable (FirstCol, SecondCol)
```

```
SELECT 1,'First'
```

```
UNION ALL
```

```
SELECT 2,'Second'
```

```
UNION ALL
```

```
SELECT 3,'Third'
```

```
UNION ALL
```

```
SELECT 4,'Fourth'
```

```
UNION ALL
```

```
SELECT 5,'Fifth'
```

```
GO
```

**/\* Check the content of the TestTable \*/**

```
USE EncryptTest
```

```
GO
```

```
SELECT *
```

```
FROM TestTable
```

```
GO
```

Kết quả:

	FirstCol	SecondCol
1	1	First
2	2	Second
3	3	Third
4	4	Fourth
5	5	Fifth

**/\* Create Database Master Key \*/**

```
USE EncryptTest
```

```
GO
```

```
CREATE MASTER KEY ENCRYPTION
```

```
BY PASSWORD = 'SQLAuthority'
```

```
GO
```

```

/* Create Encryption Certificate */
USE EncryptTest
GO
CREATE CERTIFICATE EncryptTestCert
WITH SUBJECT = 'SQLAuthority'
GO
/* Create Symmetric Key */
USE EncryptTest
GO
CREATE SYMMETRIC KEY TestTableKey
WITH ALGORITHM = TRIPLE_DES ENCRYPTION
BY CERTIFICATE EncryptTestCert
GO
/* Encrypt Data using Key and Certificate
Add Columns which will hold the encrypted data in binary */
USE EncryptTest
GO
ALTER TABLE TestTable
ADD EncryptSecondCol VARBINARY(256)
GO
/* Update binary column with encrypted data created by certificate and key */
USE EncryptTest
GO
OPEN SYMMETRIC KEY TestTableKey DECRYPTION
BY CERTIFICATE EncryptTestCert
UPDATE TestTable
SET EncryptSecondCol =ENCRYPTBYKEY(KEY_GUID('TestTableKey'),SecondCol)
GO
/* DROP original column which was encrypted for protect the data */
USE EncryptTest
GO
ALTER TABLE TestTable
DROP COLUMN SecondCol
GO
/* Check the content of the TestTable */
USE EncryptTest

```

GO

SELECT \*

FROM TestTable

GO

Kết quả câu truy vấn

	FirstCol	EncryptSecondCol
1	1	0x003B444CCFA3B040AEE7FF980C76B82301000000CEFE6F...
2	2	0x003B444CCFA3B040AEE7FF980C76B823010000007C11369...
3	3	0x003B444CCFA3B040AEE7FF980C76B82301000000BA407B...
4	4	0x003B444CCFA3B040AEE7FF980C76B823010000004623AC...
5	5	0x003B444CCFA3B040AEE7FF980C76B823010000002886EB...

**/\* Decrypt the data of the SecondCol \*/**

USE EncryptTest

GO

OPEN SYMMETRIC KEY TestTableKey DECRYPTION

BY CERTIFICATE EncryptTestCert

SELECT CONVERT(VARCHAR(50),DECRYPTBYKEY(EncryptSecondCol)) ASDecryptSecondCol

FROM TestTable

GO

	FirstCol	DecryptSecondCol
1	1	First
2	2	Second
3	3	Third
4	4	Fourth
5	5	Fifth

**/\* Clean up database \*/**

USE EncryptTest

GO

CLOSE SYMMETRIC KEY TestTableKey

GO

DROP SYMMETRIC KEY TestTableKey

GO

DROP CERTIFICATE EncryptTestCert

GO

DROP MASTER KEY

GO

```
USE [master]
GO
DROP DATABASE [EncryptTest]
GO
```

### BÀI 3

Thực hiện mã hóa và giải mã theo các lệnh sau, bạn hãy giải thích ý nghĩa, chức năng của từng lệnh đã thực hiện.

#### 1) Mã hóa mức cột:

```
USE AdventureWorks2008R2;
GO
```

--If there is no master key, create one now.

```
IF NOT EXISTS (SELECT * FROM sys.symmetric_keys WHERE symmetric_key_id =
101) CREATE MASTER KEY ENCRYPTION BY PASSWORD =
'Th15i$aS7riN&ofR@nD0m!T3%t'
```

```
GO
```

#### Mã hóa cột sử dụng mật khẩu - Encrypting Columns Using a Passphrase

```
select top 5 * from Sales.CreditCard
```

```
go
```

CreditCardID	CardType	CardNumber	ExpMonth	ExpYear	ModifiedDate
1	SuperiorCard	33332664695310	11	2006	2007-08-30
2	Distinguish	55552127249722	8	2005	2008-01-06
3	ColonialVoice	77778344838353	7	2005	2008-02-15
4.	ColonialVoice	77774915718248	7	2006	2007-06-21
5	Vista	11114404600042	4	2005	2007-03-05

```
USE AdventureWorks2008R2;
```

```
GO
```

```
select CreditCardID, CardType, CardNumber_encrypt = CONVERT(varbinary(256),
CardNumber), ExpMonth, ExpYear, ModifiedDate into Sales.CreditCard_encrypt from
Sales.CreditCard where 1 = 2
```

```
declare @passphrase varchar(128)
```

```
set @passphrase = 'unencrypted credit card numbers are bad, um-kay'
```

```
insert Sales.CreditCard_encrypt ( CardType, CardNumber_encrypt, ExpMonth, ExpYear,
```

```
select top 5 CardType, CardNumber_encrypt = EncryptByPassPhrase(@passphrase,
CardNumber), ExpMonth, ExpYear, ModifiedDate from Sales.CreditCard
```

```
select * from Sales.CreditCard_encrypt
```

```
go
```

CreditCardID	CardType	CardNumber_encrypt	ExpMonth	ExpYear	ModifiedDate
-----	-----	-----	-----	-----	-----
1	SuperiorCard	0x010000007C65089E...	11	2006	2007-08-30
2	Distinguish	0x010000000C624987...	8	2005	2008-01-06
3	ColonialVoice	0x01000000AA8761A0...	7	2005	2008-02-15
4	ColonialVoice	0x010000002C2857CC...	7	2006	2007-06-21
5	Vista	0x0100000095F6730D...	4	2005	2007-03-05

```
declare @passphrase varchar(128)
```

```
set @passphrase = 'unencrypted credit card numbers are bad, um-kay'
```

```
select CreditCardID, CardType, CardNumber = convert(nvarchar(25),
DecryptByPassPhrase(@passphrase, CardNumber_encrypt)), ExpMonth, ExpYear,
ModifiedDate from Sales.CreditCard_encrypt
```

```
GO
```

CreditCardID	CardType	CardNumber_encrypt	ExpMonth	ExpYear	ModifiedDate
-----	-----	-----	-----	-----	-----
1	SuperiorCard	33332664695310	11	2006	2007-08-30
2	Distinguish	55552127249722	8	2005	2008-01-06
3	ColonialVoice	77778344838353	7	2005	2008-02-15
4	ColonialVoice	77774915718248	7	2006	2007-06-21
5	Vista	11114404600042	4	2005	2007-03-05

```
-- The first step is to create the certificate with the CREATE CERTIFICATE command:
```

```
USE AdventureWorks2008R2;
```

```
CREATE CERTIFICATE BillingDept01 WITH SUBJECT = 'Credit Card Billing'
```

```
GO
```

```
USE AdventureWorks2008R2;
```

```
CREATE SYMMETRIC KEY BillingKey2010 WITH ALGORITHM = AES_256
ENCRYPTION BY CERTIFICATE BillingDept01;
GO
```

```
USE AdventureWorks2008R2;
```

```
Truncate table Sales.CreditCard_encrypt
```

```
USE AdventureWorks2008R2;
```

```
-- First, decrypt the key using the BillingDept01 certificate
```

```
OPEN SYMMETRIC KEY BillingKey2010 DECRYPTION BY CERTIFICATE
BillingDept01
```

```
-- Now, insert the rows using the symmetric key
```

```
-- encrypted by the certificate
```

```
insert Sales.CreditCard_encrypt ( CardType, CardNumber_encrypt, ExpMonth, ExpYear,
ModifiedDate )
```

```
select      top      5      CardType,      CardNumber_encrypt      =
EncryptByKey(KEY_GUID('BillingKey2010'), CardNumber), ExpMonth, ExpYear,
ModifiedDate from Sales.CreditCard
```

```
select * from Sales.CreditCard_encrypt
```

```
go
```

CreditCardID	CardType	CardNumber_encrypt	ExpMonth	ExpYear	ModifiedDate
1	SuperiorCard	0x0046C380E7A27749...11		2006	2007-08-30
2	Distinguish	0x0046C380E7A27749... 8	2005		2008-01-06
3	ColonialVoice	0x0046C380E7A27749...7	2005		2008-02-15
4	ColonialVoice	0x0046C380E7A27749...7	2006		2007-06-21
5	Vista	0x0046C380E7A27749...4	2005		2007-03-05

### Giải mã

```
USE AdventureWorks2008R2;
```

```
OPEN SYMMETRIC KEY BillingKey2010 DECRYPTION BY CERTIFICATE
BillingDept01
Select CardType, CardNumber = convert(nvarchar(25),
DecryptByKey(CardNumber_encrypt)), ExpMonth, ExpYear, ModifiedDate from
Sales.CreditCard_encrypt
```

Go

CreditCardID	CardType	CardNumber_encrypt	ExpMonth	ExpYear
1	SuperiorCard	33332664695310	11	2006
2	Distinguish	55552127249722	8	2005
3	ColonialVoice	77778344838353	7	2005
4	ColonialVoice	77774915718248	7	2006
5	Vista	11114404600042	4	2005

CLOSE SYMMETRIC KEY BillingKey2010

```
select name, pvt_key_encryption_type, issuer_name, subject, expiry_date =
CAST(expiry_date as DATE), start_date = CAST(start_date as DATE) from sys.certificates
go
name pvt_key_encryption_type issuer_name subject expiry_date start_date
```

```
BillingDept01 MK Credit Card Billing
Credit Card Billing 2011-05-01 2010-05-01
```

```
select name, key_length, key_algorithm, algorithm_desc, create_date = CAST(create_date as
DATE), modify_date = CAST(create_date as DATE), key_guid from sys.symmetric_keys
go
name key_length key_algorithm algorithm_desc
create_date modify_date key_guid
```

```
##MS_DatabaseMasterKey## 128 D3 TRIPLE_DES
2010-04-30 2010-04-30 A3550B00-6BAE-41E2-A1BC-D784DC35779E
BillingKey2010 256 A3 AES_256
2010-04-30 2010-04-30 10C5C800-0B4C-44C2-9F71-5415007C2E81
```

--If the usage of the key and certificate are no longer needed, they should be dropped from the database:

DROP SYMMETRIC KEY BillingKey2010

DROP CERTIFICATE BillingDept01

## 2) Mã hóa dữ liệu trong suốt

### Implementing Transparent Data Encryption

Các bước để thực hiện mã hóa dữ liệu trong suốt:

- Create a master key.
- Create or obtain a certificate protected by the master key.
- Create a database encryption key and protect it by the certificate.
- Configure the database to use encryption.

USE master;

GO

--Create the master key which is stored in the master database

CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'mystrongpassword\$\$';

GO

-- Create a certificate that is also stored in the master -- database. This certificate will be used to encrypt a user database

CREATE CERTIFICATE MyCertificate with SUBJECT = 'Certificate stored in Master Db'

GO

-- Create a Database Encryption Key (DEK) that is based

-- on the previously created certificate

-- The DEK is stored in the user database

USE AdventureWorks2008R2

GO

CREATE DATABASE ENCRYPTION KEY WITH ALGORITHM = AES\_256

ENCRYPTION BY SERVER CERTIFICATE MyCertificate

GO

--Turn the encryption on for the AdventureWorks2008R2

ALTER DATABASE AdventureWorks2008R2 SET ENCRYPTION ON

GO

--Xem quá trình mã hóa

SELECT DBName = DB\_NAME(database\_id), encryption\_state FROM sys.dm\_database\_encryption\_keys ;

GO

DBName encryption\_state

-----

tempdb 3

AdventureWorks2008R2 3

Khi TDE có hiệu lực (enabled =true), mã hóa sẽ áp dụng các file liên quan đến CSDL là:

Database Data Files (.mdf or .ndf)

Database Log Files (.ldf)



Database Backups (All database backups, including full, differential, and log, are encrypted)  
Tempdb (If any databases on a server are encrypted with TDE, the tempdb database is also encrypted).

### Managing TDE in SSMS

Right-click on the database in the Object Explorer for which you want to configure TDE and select Tasks; then select Manage Database Encryption.

If you are setting up the initial configuration for TDE in a database, you see a dialog like that shown in Figure 12.2.

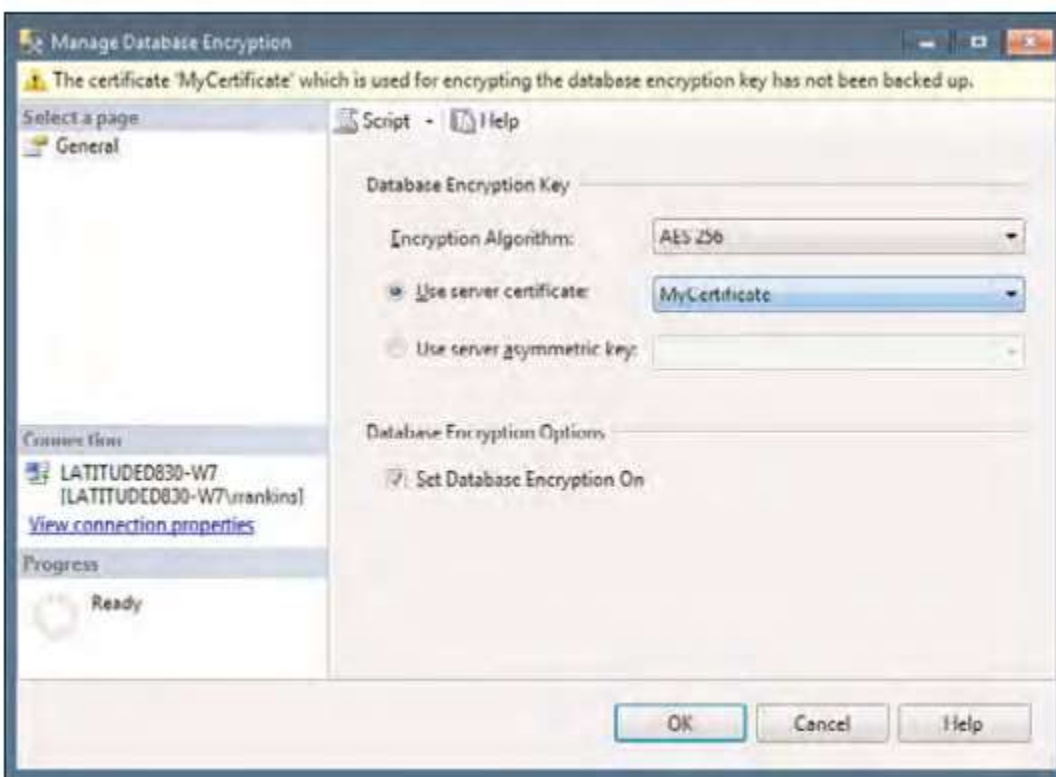


FIGURE 12.2 Enabling TDE in SSMS.

You specify the encryption algorithm to be used and the server certificate used to protect the database encryption key.

When you are ready to enable TDE for the database, put a check mark in the Set Database Encryption On check box.

If TDE is already enabled for a database, the dialog changes to provide you with options to re-encrypt the database encryption key and to regenerate the DEK using a different encryption algorithm. You can also enable/disable database encryption (see Figure 12.3). A second page displays the current TDE properties and encryption state of the database (see Figure 12.4).

## BÀI 4

Thực hiện mã hóa và giải mã theo các lệnh sau, bạn hãy giải thích ý nghĩa, chức năng của từng lệnh đã thực hiện

### Backing Up TDE Certificates and Keys

The most important issue to consider when using TDE is that you must back up and retain the certificate and private key associated with the encryption. If these things are lost or unavailable, you are not able to restore or attach the encrypted database files on another server.



FIGURE 12.3 Modifying TDE properties in SSMS.

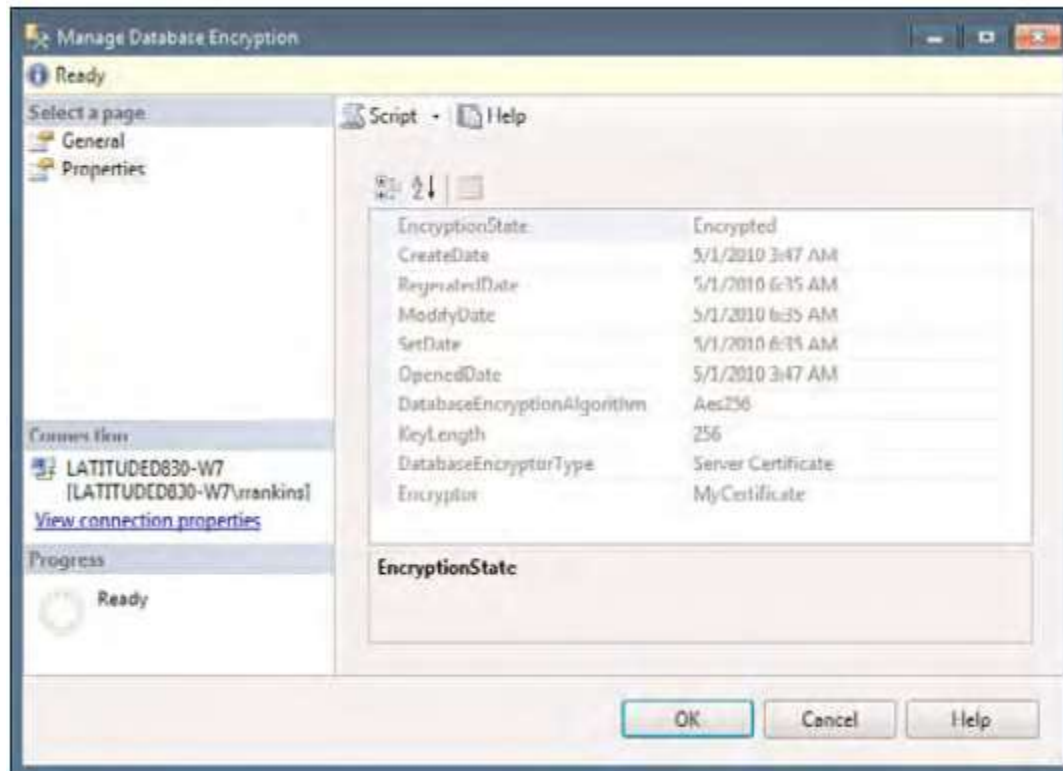


FIGURE 12.4 Viewing TDE properties in SSMS

--Backing up the certificate, private key, and master key for the server is relatively straightforward.

BACKUP MASTER KEY TO FILE = 'c:\mssql2008\backup\masterkey'

ENCRYPTION BY PASSWORD = 'somekeybackuppasword\$'

--Backing up the certificate and associated private key also uses the **BACKUP** command. The following example backs up the certificate created in Listing 12.1:

BACKUP CERTIFICATE MyCertificate TO FILE = 'c:\mssql2008\backup\MyCertificate'  
WITH PRIVATE KEY ( FILE = 'c:\mssql2008\backup\MyCertificatePrivateKey' ,  
ENCRYPTION BY PASSWORD = 'somecertbackuppasword\$\$' )

--If you want to restore a database backup on another server instance, a master key for the server must exist. If one does not exist, you can create one by using the `CREATE MASTER KEY ENCRYPTION` syntax. After creating the master key, you are able to create the TDE certificate from a backup of the certificate from the original SQL Server instance, as shown in the following example:

```
CREATE CERTIFICATE MyCertificate FROM FILE =
'c:\mssql2008\backup\MyCertificate' WITH PRIVATE KEY (FILE =
'c:\mssql2008\backup\MyCertificatePrivateKey',
DECRYPTION BY PASSWORD = 'somecertbackuppassword$$')
```

After the certificate is restored on the other server instance, you can restore the encrypted database backup. At this point, the restore can be performed just as you would restore any unencrypted database backup. The restored database is also encrypted and behaves like the original TDE database. TDE is a relatively simple and effective way to encrypt and protect your data. Other encryption methods that exist with SQL Server can protect different elements of your database. Encryption can be applied to columns of data, an entire table, as well as the communication that occurs between databases and the clients that access them. The level of encryption and need to use it depend on the type of data you are securing.

## BÀI 5

Ứng dụng bài toán thực tế

- 1) Tạo CSDL QLDA các thông số tùy ý
- 2) Tạo các Table sau:  
 Phongban(mapb, tenpb, mota)  
 NhanVien(Manv, tennv, pass, phai, DienThoai, email, mapb)  
 Duan(Mada, TenDa, NgayBD, KinhPhi, MaPB)  
 Thamgia(Manv, Mada, NgayTG, MucLuong, CongViec)  
 Kiểu dữ liệu sinh viên tự qui định. Yêu cầu tạo đầy đủ các ràng buộc khóa chính và khóa ngoại
- 3) Theo bạn, bạn sẽ mã hóa theo cột nào trong bảng NhanVien? Vì sao? Bạn hãy thực hiện mã hóa và giải mã cho cột mà bạn đã chọn theo hai cách dùng password và dùng chứng chỉ. Cho biết kết quả sau khi mã hóa và giải mã.
- 4) Thực hiện mã hóa và giải mã dữ liệu trong suốt cho cơ sở dữ liệu QLDA

## Bài thực hành tuần 6

### Replication

#### 1. Chuẩn bị CSDL để replication

- a. Attach CSDL AdventureWork2008
- b. Dùng Database Copy wizard để tạo 2 bản copy của CSDL AdventureWork2008, lần lượt đặt tên là AdvWrksSnapshotRepl và AdvWrksTransRepl trên instance thứ nhất và Adv2 trên instance thứ hai
- c. Xác định instance thứ nhất là distributor của replication

#### 2. Tạo snapshot replication

- a. Tạo 1 publication mới với replication loại snapshot trên instance thứ 1 của SQL Server cho CSDL AdvWrksSnapshotRepl, với các tính năng sau:
  - Sử dụng account của Windows student cho các Agent
  - Publish 10 bảng liên quan đến customer trong CSDL
  - Tạo snapshot tức thì (snapshot immediately) và lên lịch tạo snapshot sau mỗi 2 giờ.
- b. Tạo 1 subscription mới ứng với publication vừa tạo trên instance thứ hai, với chạy tất cả agent trên distributor (Run all agents at the distributor) , lịch đồng bộ là run continuously, tạo 1 DB mới cho subscription có tên là Pub1 và khởi tạo subscription ngay tức thì

#### 3. Tạo transaction replication

- a. Tạo một publication mới dùng replication loại transactional trên instance thứ nhất cho CSDL AdvWrksTransRepl với các tính năng sau:
  - Publish các bảng về Product trong CSDL.
  - Tạo snapshot immediately và lập lịch chạy 1 lần sau mỗi 2 giờ
  - Sử dụng account của Windows student cho các Agent
- b. Tạo một subscription mới trên instance số 2 tương ứng với publication vừa tạo và tạo 1 CSDL mới tên là Pub2 làm subscription

#### 4. Thay đổi dữ liệu trên CSDL dùng làm publisher trên instance số như sau:

- a. Bổ sung 2 bản ghi mới vào bảng Production.Product như sau:

**SET IDENTITY\_INSERT [Production].[Product] ON**

**insert Production.Product**

**([ProductID],[Name],[ProductNumber],[MakeFlag],[FinishedGoodsFlag]**  
**,[SafetyStockLevel],[ReorderPoint],[StandardCost],[ListPrice]**  
**,[DaysToManufacture],[SellStartDate],[ModifiedDate])**

## VALUES

```
(1999,N'Lap top Vaio','CD-1111',0,0,500,200,99,0,33,getdate(), getdate())
```

```
insert Production.Product
```

```
 ([ProductID],[Name],[ProductNumber],[MakeFlag],[FinishedGoodsFlag]  
 ,[SafetyStockLevel],[ReorderPoint] ,[StandardCost],[ListPrice]  
 ,[DaysToManufacture],[SellStartDate] ,[ModifiedDate])
```

values

```
(2999,N'Galaxy 03','CD-2222',0,0,500,200,99,0,33,getdate(),  
 getdate())
```

Cập nhật lại publication thứ hai, kiểm tra CSDL Pub2 xem đã được đồng bộ với CSDL AdvWrksTransRepl chưa?

b. Mở Replication Monitor kiểm tra trạng thái của các replication vừa tạo

## 5. Tạo merge replication

- Tạo distributor là instance thứ hai.
- Tạo 1 publication với CSDL là Adv2 trên instance thứ hai, loại Merge Replication.
- Tạo 1 subscription với CSDL mới tên m1 trên instance thứ nhất, loại pull subscription
- Kiểm tra việc đồng bộ giữa 2 CSDL publication và subscription:
  - Tạo 1 bảng dữ liệu mới tùy ý trong CSDL m1
  - Tạo 1 bảng dữ liệu mới tùy ý trong CSDL Adv2.
  - Kiểm tra xem dữ liệu có được cập nhật đồng thời trong 2 CSDL

## 6. Chọn lệnh Launch Replication Monitor: xem các replication đã tạo

## Bài tập thực hành 7

### Mirroring và log shipping

CSDL mẫu: AdventureWorks2008

#### 1. Thực hiện mirroring:

Trong bài này, các bạn giả lập 2 server chạy trên cùng 1 máy tính bằng cách cài 2 instance cùng chạy trên một máy tính:

Instance thứ 1 là primary server (nằm phía trên của cây thư mục)

Instance thứ 2 là secondary server (nằm phía dưới của cây thư mục)

#### Bước chuẩn bị:

- Attach file AdventureWorks2008\_Data.mdf, AdventureWorks2008\_Log.ldf để tạo cơ sở dữ liệu mẫu AdventureWorks2008 trên primary server, chọn mode recovery của DB này là Full, sau đó thực hiện full backup.

- Restore backup này trên secondary server (được chọn làm mirror server) với tùy chọn WITH NORECOVERY. Lưu ý DB được restore phải cùng tên với DB trên instance thứ nhất

#### Thực hiện mirroring

- Mở properties của DB AdventureWorks2008 trên primary server, chọn trang Mirroring
- Nhấp nút **Configure Security**, xuất hiện **Configure Database Mirroring Security Wizard**
- Thực hiện mirror (trình tự các bước như slide bài giảng) và kiểm tra kết quả trên mirror server

#### 2. Thực hiện log shipping

#### Bước chuẩn bị:

##### Chuẩn bị cho phía Primary Server:

- **Bước 1:** Tạo thư mục **T:\Backup\_log** (để copy transaction log của database phía Primary Server vào thư mục này)
- **Bước 2:** Click chuột phải vào thư mục **Backup\_log** → chọn properties → chọn tab **Sharing** → chọn nút Share → chọn every one ở ngay combo box sổ xuống → chọn ở chế độ share read/write hoặc read only
- **Bước 3:** Vào My Computer để xem tên máy tính của bạn là gì, xem ở mục Computer name, ví dụ như tên máy tính của bạn là **NguyenVanA**. Click vào biểu



tượng Windows ở góc phía trái, bên dưới màn hình máy tính, gõ vào \\ **NguyenVanA\ Backup\_log** xem thử có hiện ra đúng thư mục bạn vừa share ở bước 1 không, nếu đúng sang mục tiếp theo.

- **Bước 4:** Chuẩn bị CSDL AdventureWorks2008 để thực hiện log shipping

### Chuẩn bị cho phía **Secondary Server**:

- Tạo thư mục **T:\Copy\_transaction\_log** (thư mục này để copy transaction log từ thư mục chia sẻ ở bước 1 của **Primary Server** về)

### Thực hiện log shipping:

Trong bài này, các bạn giả lập 2 server chạy trên cùng 1 máy tính bằng cách cài 2 instance cùng chạy trên một máy tính

Instance thứ 1 là primary server (nằm phía trên của cây thư mục)

Instance thứ 2 là secondary server (nằm phía dưới của cây thư mục)

1. Tại instance thứ 1 (phía primary server ), nhấn chuột phải vào CSDL AdventureWorks2008, chọn **Properties**.
2. Ngay trang **Select a page**, chọn **Transaction Log Shipping**.
3. Nhấn chọn vào check box **Enable this as a primary database in a log shipping configuration**.
4. Ngay trang **Transaction log backups**, nhấn vào **Backup Settings**.
5. Ở hộp thoại **Network path to the backup folder**, gõ vào \\NguyenVanA\ Backup\_log
6. Ở hộp thoại **If the backup folder is located on the primary server, type a local path to the folder**, gõ vào: T:\Backup\_log
7. Xác định các tham số **Delete files older than** và **Alert if no backup occurs within** ở hộp thoại tương ứng
8. Chú ý lịch biểu backup được liệt kê bên dưới hộp **Schedule** bên dưới mục **Backup job**. Nếu bạn muốn chỉnh lại lại lịch biểu, bạn nhấn vào nút **Schedule...**
9. SQL Server 2008 Enterprise hỗ trợ backup dạng nén. Khi cấu hình 1 log shipping, bạn có thể lựa chọn 1 trong các tùy chọn sau: **Use the default server setting**, **Compress backup**, hoặc **Do not compress backup**.
10. Chọn **OK**.
11. Ở trang **Secondary server instances and databases**, nhấn nút **Add**.
12. Nhấn vào nút **Connect** và kết nối đến instance của SQL Server mà bạn sử dụng làm secondary server
13. Ở hộp thoại **Secondary Database**, chọn CSDL AdventureWorks2008
14. Ở Tab **Initialize Secondary database**, chọn các tùy chọn để khởi tạo secondary database.
15. Ở tab **Copy Files**, ở hộp **Destination folder for copied files**, gõ đường dẫn của thư mục chứa các file transaction logs được chép từ primary server sang. Thư mục này nằm tại secondary server.

Trong bài này, các bạn gõ vào: **T:\Copy\_transaction\_log**



16. Chú ý lịch biểu copy được liệt kê bên dưới hộp **Schedule** bên dưới mục **Copy job**. Nếu bạn muốn chỉnh lại lịch biểu, bạn nhấn vào nút **Schedule...**Lịch này gần giống hoặc có thể giống với lịch backup ở bước 8.
17. Ở tab **Restore**, ở mục **Database state when restoring backups**, chọn **Standby mode** và nhấn chọn vào check box **disconnect users in the database when restoring backups**.
18. Chú ý lịch biểu restore được liệt kê bên hộp **Schedule** box dưới mục **Restore job**. Nếu bạn muốn chỉnh lại lịch biểu, bạn nhấn vào nút **Schedule...**Lịch này gần giống hoặc có thể giống với lịch backup ở bước 8.
19. Nhấn vào nút **OK**
20. Ở trang **Monitor server instance**, nếu có server giám sát thì bạn chọn vào check box **Use a monitor server instance** và chọn nút **Settings**. Trong bài này, các bạn không chọn mục này.
21. Nhấn nút **OK**.
22. Ở hộp thoại **Database Properties** , nhấn nút **OK** để bắt đầu tiến trình cấu hình log shipping.

Thực hiện việc kiểm tra log shipping bằng cách kiểm tra xem thử các **transaction log** có ở **thư mục phía primary server** không và các **transaction log có copy và restore** ở phía **secondary server** theo đúng lịch đã cấu hình hay không.

Để dễ theo dõi, ta nên lên lịch 1,2 phút copy 1 lần

## Bài thực hành tuần 8

### PowerShell

CSDL mẫu: AdventureWorks2008

#### Bài 1

1. Thực hiện việc **khởi động PowerShell** theo các cách sau:
  - a. Start ⇔ Programs ⇔ Windows PowerShell 1.0, chọn Windows PowerShell.
  - b. Từ dấu nhắc lệnh của DOS, gõ vào lệnh “powershell” hoặc gõ vào lệnh sqlps.exe
  - c. Khởi động PowerShell trong SQL server Management studio
2. **Nhận biết môi trường PowerShell** bằng dấu nhắc lệnh PS>
3. **Dùng lệnh của PowerShell** để xem nội dung của thư mục hiện hành.
4. Dùng lệnh **get-process** để trả về 1 tập các process trong hệ thống, **sắp xếp giảm dần theo size** của workingset hay dung lượng bộ nhớ của mỗi tiến trình đang sử dụng, sau đó **chọn top 10 process lớn nhất**.

#### Bài 2:

1. Tạo và thực thi PowerShell script để hiển thị thông tin các server đang chạy trên SQL Server.

#### Hướng dẫn:

1. Tạo file script với tên **getsysinfo.ps1** với nội dung sau:

```
#getsysinfo.ps1
# Use WMI queries to retrieve information about the computer, operating
# system and disk devices

gwmi -query "select * from Win32_ComputerSystem" | select Name, Model,
Manufacturer, Description, DNSHostName, Domain, DomainRole,
PartOfDomain, NumberOfProcessors, SystemType, TotalPhysicalMemory,
UserName, Workgroup | format-list

gwmi -query "select * from Win32_OperatingSystem" | select Name,
Version, FreePhysicalMemory, OSLanguage, OSProductSuite, OSType,
ServicePackMajorVersion, ServicePackMinorVersion | format-list

gwmi -query "select * from Win32_PhysicalMemory" | select Name,
Capacity, DeviceLocator, Tag | format-table -AutoSize

gwmi -query "select * from Win32_LogicalDisk where
DriveType=3" | select Name, FreeSpace, Size | format-table -AutoSize
```

2. Viết lệnh để xem **chính sách thực thi (execution policy)** hiện hành ở PowerShell.
3. Nếu chính sách thực thi hiện hành là **Restricted** (mặc định) , viết lệnh để chỉnh chỉnh sách thực thi trong PowerShell sang **RemoteSigned**.
4. Gõ vào dấu nhắc lệnh trong môi trường PowerShell đường dẫn file script vừa tạo trên để thực thi file script, quan sát kết quả.
2. Thực thi file script **CreateDB.ps1** để tạo database tại môi trường PowerShell (**sử dụng SMO**) .
3. Thực thi file script **EmployeeExtract.ps1** để xuất ra danh sách 25 nhân viên đầu tiên trong bảng Person.Person của CSDL AdventureWorks2008 (**sử dụng ADO.NET**).
4. Thực thi file script **Backup.ps1** để thực hiện việc backup trên các CSDL người dùng công việc sau:
  - Thực hiện việc full backup CSDL nếu IsSystemObject = False và IsMirroringEnabled=False
  - Nếu mô hình phục hồi không phải là SIMPLE thì thực hiện việc backup transaction log
5. Thực thi file script **CreateTable.ps1** để tạo bảng, khóa chính, khóa ngoại, các chỉ mục.

6. Thực thi file script **Scripting.ps1** để tạo script cho các đối tượng trong CSDL AdventureWorks2008.
7. Thực thi file script **dept\_birthdays.ps1** để trích thông tin các nhân viên từng phòng ban thông qua cột DepartmentID, và tạo 1 file vật lý riêng lẻ cho mỗi phòng ban. Các file này là text file với dấu ‘,’ là ký tự phân cách cho các cột.
8. Chạy Powershell trong SQL Server Agent: tạo 1 Powershell job để chạy các script vừa tạo.

## Bài thực hành tuần 9

### POLICY-BASED

#### BÀI 1

#### 3) **Creating a Central Management Server** (Tạo một máy chủ quản lý trung tâm)

**Thực hiện theo các bước sau để đăng ký máy chủ quản lý trung tâm:**

1. Trong SQL Server Management Studio, mở menu View và nhấp vào Registered Servers.
2. Trong cửa sổ Registered Servers, click Database Engine, click chuột phải vào Central Management Servers, chọn Register Central Management Server.
3. Trong cửa sổ New Server Registration dialog, xác định tên của Central Management Server.
4. Nếu cần thiết, xác định thêm thuộc tính kết nối trong Tab Connection Properties hay click nút Save.

**Các bước Đăng ký SQL Server instances để liên kết với Central Management Server.**

1. Click phải vào Central Management Server mà bạn muốn liên kết với SQL Server instance.
2. Chọn New Server Registration.
3. Trong cửa sổ New Server Registration, xác định tên của SQL Server Instance và thuộc tính kết nối trong Connection Properties, click nút Save
4. Lặp lại bước 1 tới bước 3 cho tất cả các SQL Server instances mà bạn muốn đăng ký với Central Management Server.

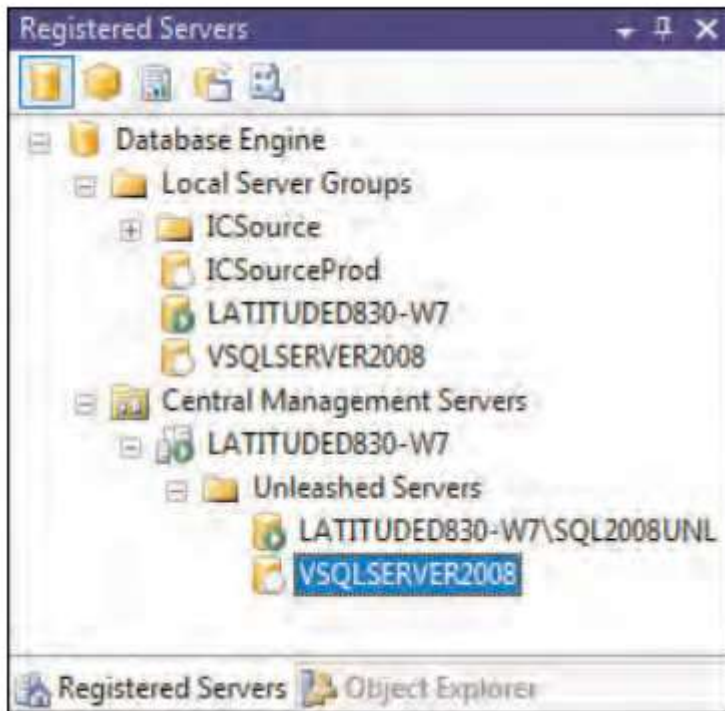


FIGURE 22.2 Central Management Server with Registered SQL Server instances.

#### 4) Importing and Evaluating Policies to the Central Management Server

##### Import policies for multiple instances:

- Right-clicking the Central Management Server or Server Group → select Import Policies.

##### Evaluate the policies:

- Right-clicking the Central Management Server or Server Group → select Evaluate.

#### 5) Implementing Policy-Based Management

Có 6 bước để thực thi và điều hành Policy-Based Management: .

- Creating a condition based on a facet .
- Creating a policy based on that condition .
- Creating a category .
- Creating a Central Management Server .
- Subscribing to a category .
- Exporting or importing a policy

#### G. Creating a Condition Based on a Facet

##### 1. Creating a Condition

- 1) In **Object Explorer**, click the plus sign to expand the server where you want to create a Policy-based Management condition.

- 2) Click the plus sign to expand the **Management** folder.
- 3) Click the plus sign to expand **Policy Management**.
- 4) Click the plus sign to expand the **Facets** folder.
- 5) Right-click the facet in which you want to create a new condition and select **New Condition**.
- 6) In the **Create New Condition** dialog box, in the **Name** box, type the name of the new condition.
- 7) Confirm the correct facet in the **Facet** list, or select a different facet.
- 8) Under **Expression**, construct condition expressions by selecting a facet property in the **Field** box, together with its associated operator and value. When you add multiple expressions, the expressions can be joined by using **And** or **Or**

For this example,

- a. use the @Name property.
- b. In the Operator drop-down box, select the NOT LIKE operator.
- c. In the value text box, enter 'tbl%'.

AndOr	Field	Operator	Value
	@Name	NOT LIKE	'tbl%'
AND	Len (@Name)	<=	50
AND	@Name	NOT LIKE	'%s'

Click OK to finalize the creation of the condition. You may have to click on the Field text box again for the OK button to be enabled.

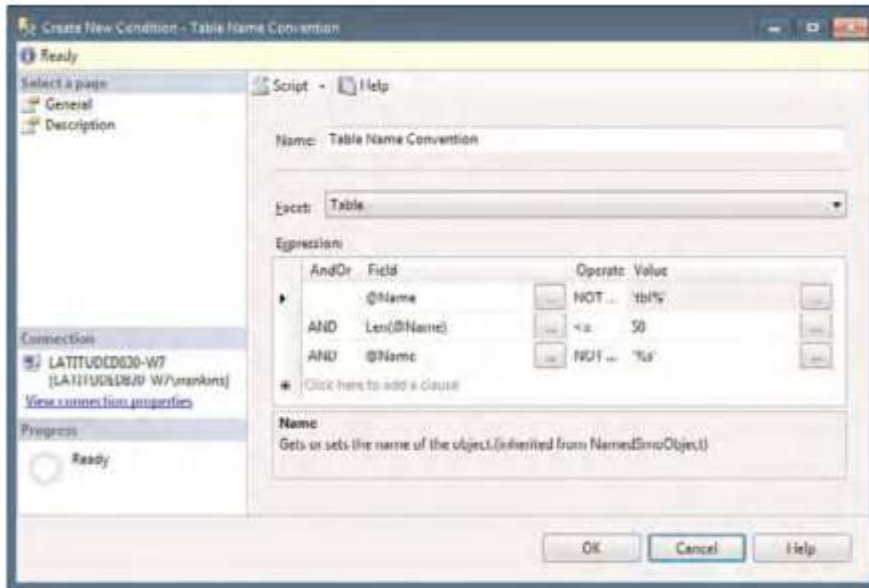


FIGURE 22.3 Creating a condition based on a facet.

## 2) To delete a condition

1. In **Object Explorer**, click the plus sign to expand the server that contains the condition that you want to delete.
2. Click the plus sign to expand the **Management** folder.
3. Click the plus sign to expand **Policy Management**.
4. Click the plus sign to expand the **Conditions** folder.
5. Right-click the condition that you want to delete and select **Delete**.
6. In the **Delete Object** dialog box, ensure that the correct condition is selected and then click **OK**.

## 3) To view or modify a condition's properties

1. In **Object Explorer**, click the plus sign to expand the server that contains the condition that you want to view or modify.
2. Click the plus sign to expand the **Management** folder.
3. Click the plus sign to expand **Policy Management**.
4. Click the plus sign to expand the **Conditions** folder.
5. Right-click the condition that you want to view or edit and select **Properties**.
6. When finished, click **OK**.

## 4) To view a condition's properties

1. In **Object Explorer**, connect to an instance of Database Engine.
2. On the Standard bar, click **New Query**.
3. Copy and paste the following example into the query window and click **Execute**.

Copy



```
USE msdb;
GO
SELECT name,
       description,
       facet,
       expression,
       is_name_condition,
       obj_name
FROM syspolicy_conditions;
GO
```

## H. Creating a Policy

### 1. Creating a Policy

1. In Object Explorer, expand the Management folder, expand the Policy Management folder, and then click on Policies.
2. Right-click on the Policies folder and select New Policy.
3. On the General tab of the Create New Policy dialog, enter a name for the new policy, such as Check Table Naming Conventions.
4. In the Check Condition drop-down box, select a condition, such as the one created in the previous example, or select New to generate a new condition from scratch.
5. The Against Targets section indicates which objects the policy should be evaluated against. For example, you could create a new condition that applies to a specific database, all databases, a specific table, all tables, or to databases created after a specific date. In the Action Targets section, indicate which targets this condition should apply to.
6. Specify the Evaluation Mode by selecting one of the options in the drop-down menu. The options include On Demand, On Schedule, On Change Log Only, and On Change Prevent. NOTE If On Schedule is selected for the Evaluation Mode, specify a schedule from the predefined list or enter a new schedule.
7. The final drop-down box is Server Restriction. You can restrict which servers you do not want the policy to be evaluated against or enforced on by creating a server condition. Create a server restriction or leave the default setting None. An example of the policy settings for checking table name conventions is displayed in Figure 22.4.

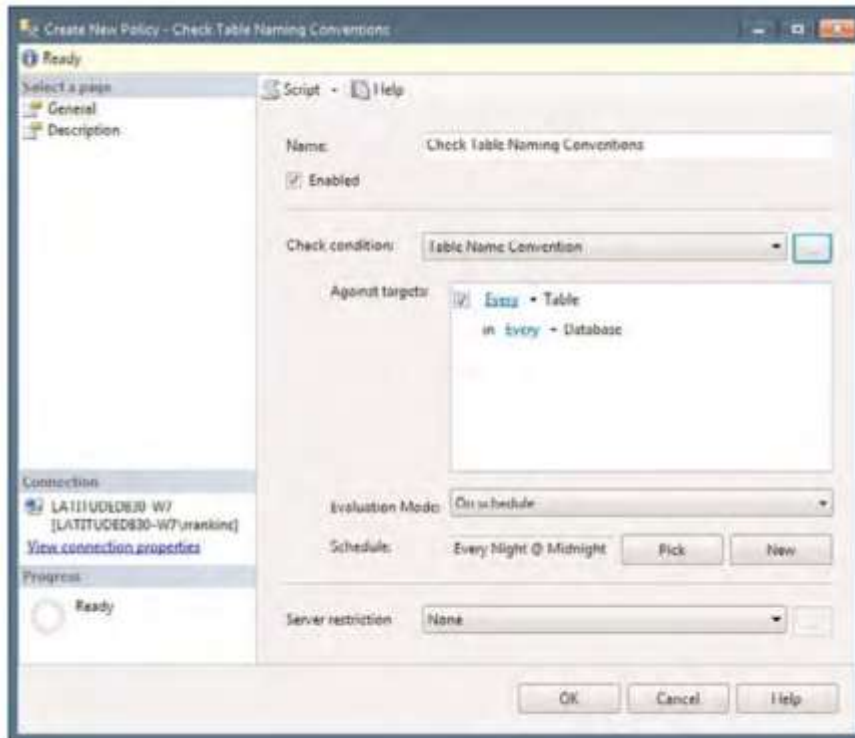


FIGURE 22.4 The Create New Policy dialog.

Before you close the Create New Policy dialog, ensure that the policy is enabled (the Enabled check box is selected) and then click on the Description page. The Description page allows you to categorize your policy, but it also allows you to display a custom text message when a policy is violated and a hyperlink where the DBA/developer can go for more information about the policy. 9. Click OK to finalize the creation of the new policy.

## 2) To view all of the facets in an object

1. In Object Explorer, right-click an instance of SQL Server, instance object, database, or database object, and then click **Facets**.
2. In the **View Facets -object\_name** dialog box, in the **Facet** list, select a facet to view its properties. For more information on the available options in this dialog box, see [View Facets Dialog Box](#).
3. When finished, click **OK**.

## 3) To copy a facet state to an XML file

1. In Object Explorer, right-click an instance of SQL Server, instance object, database, or database object, and then click **Facets**.
2. In the **View Facets -object\_name** dialog box, click **Export Current State as Policy**.
3. In the **Export as Policy** dialog box, type the path and name of the file; or use the Browse (...) button to locate the file, and then type the name of the XML file. For more

information on the available options in this dialog box, see [Export As Policy Dialog Box](#)

4. When finished, click **OK**.

#### 4) Create and manage policies, to:

1. Select a Policy-Based Management facet that contains the properties to be configured.
2. Define a condition that specifies the state of a management facet.
3. Define a policy that contains the condition, additional conditions that filter the target sets, and the evaluation mode.
4. Check whether an instance of SQL Server is in compliance with the policy.

For failed policies, Object Explorer indicates a critical health warning as a red icon next to the target and the nodes that are higher in the Object Explorer tree.

#### I. Creating a Category

- 1) Click on the Description page in the Create New Policy dialog. Policies can be placed in the default category or a specific category, or you can create a new category. Specifying a category is illustrated in Figure 22.6.
- 2) You can also create categories by right-clicking on Policy Management and selecting Manage Categories. If you choose to create a new category, click on the New button. This presents a dialog that allows you to name the category. By default, this policy is parked in the new category.
- 3) You can also select which category you want policies to belong to by selecting a specific category in the drop-down box. After you categorize your policies, you can select which categories you want your database to subscribe to.
- 4) Right-click on the Policy Management folder and select Manage Categories. The Manage Policy Categories dialog (illustrated in Figure 22.7) appears.
  - For example, if you have thirdparty software that does not follow your naming standards, you should ensure that the policies that enforce your naming standards are not in the default category. Then selectively have each of your user databases on your server subscribe to these databases.

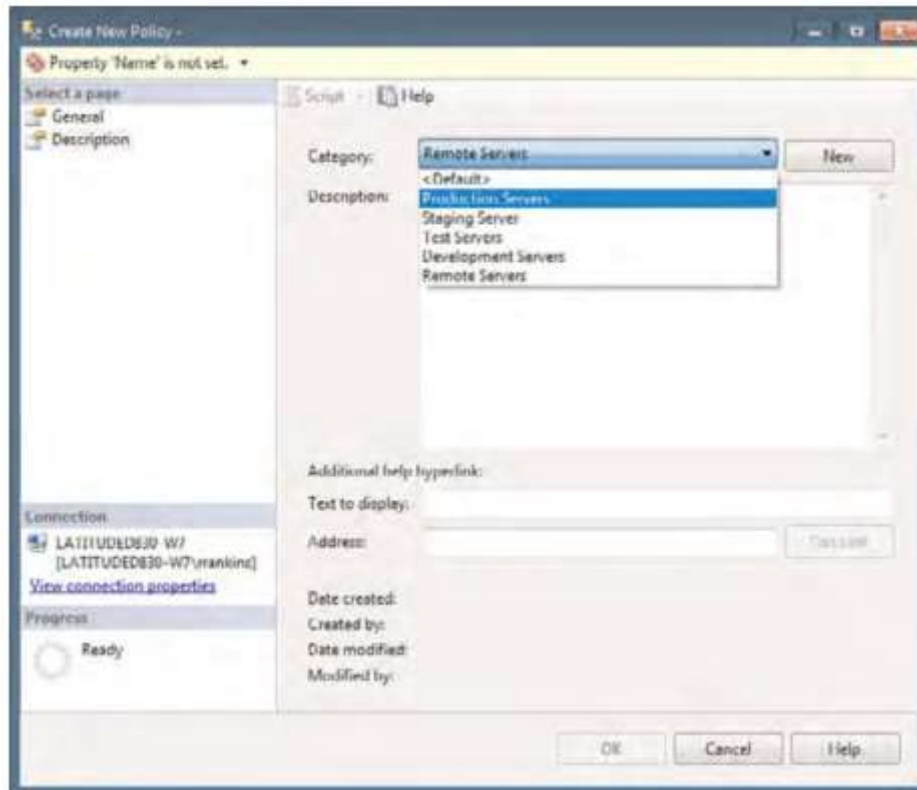


FIGURE 22.6 The category selection dialog.

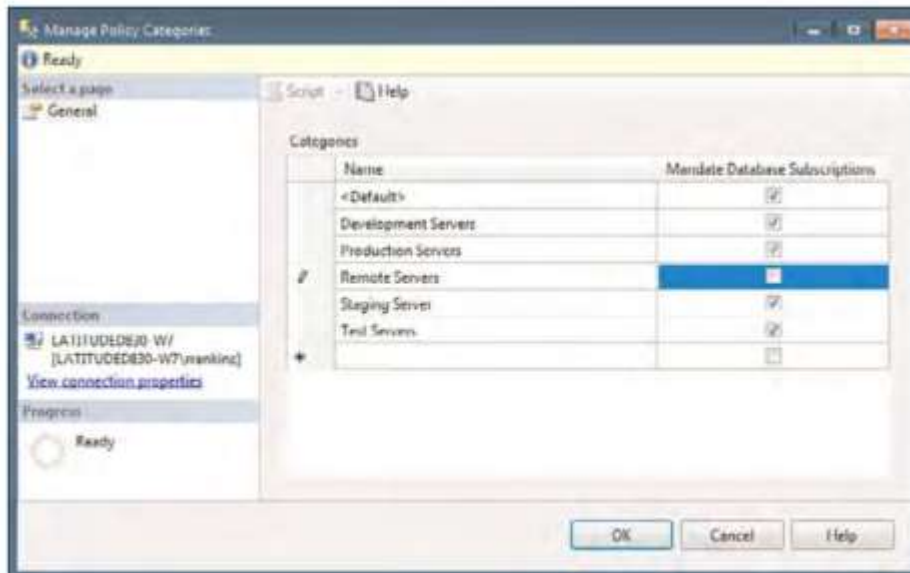


FIGURE 22.7 The Manage Policy Categories dialog.

## J. Evaluating Policies

### To evaluate a policy from an object

1. In Object Explorer, right-click a server instance, a database, or a database object, point to **Policies**, and select **Evaluate**.

2. In the **Evaluate Policies** dialog box, select one or more policies and click **Evaluate** to run the policy in evaluation mode. This generates a compliance report for the target set but does not reconfigure SQL Server or enforce future compliance. For targets that do not comply with the selected policies and have properties that can be reconfigured by Policy-Based Management, you can enforce policy compliance by clicking **Apply**.

## K. Importing and Exporting Policies

### Using SQL Server Management Studio

#### 1) To export a policy

1. In Object Explorer, click the plus sign to expand the server that contains the Policy-Based Management policy that you want to export.
2. Click the plus sign to expand the **Management** folder.
3. Click the plus sign to expand **Policy Management**.
4. Click the plus sign to expand the **Policies** folder.
5. Right-click the policy that you want to export and select **Export Policy**.
6. In the **Export Policy** dialog box, type the path and name of the file in the address bar. Alternately, find a suitable location for the file in the dialog box's navigation pane, and then type the name of the XML file in the **File Name** box.
7. When finished, click **Save**.

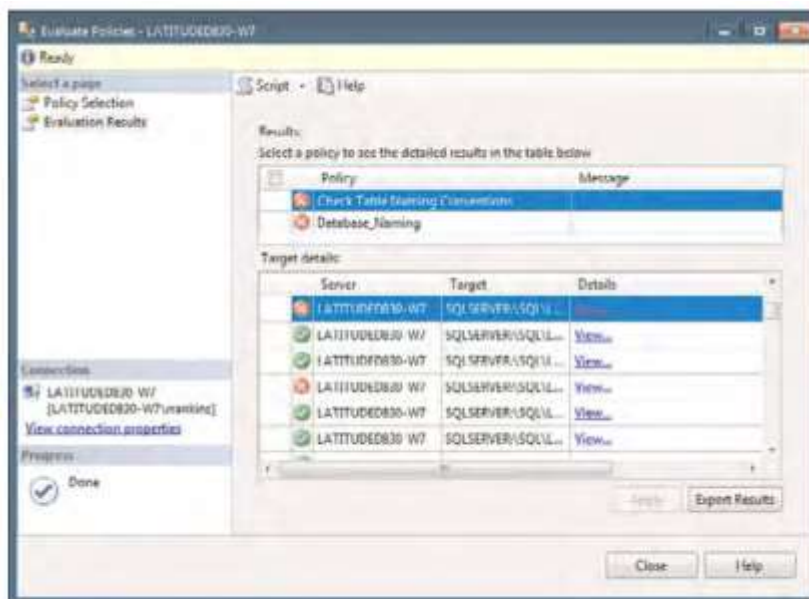


FIGURE 22.8 The Evaluation Results pane.

## 2) To import a policy instance

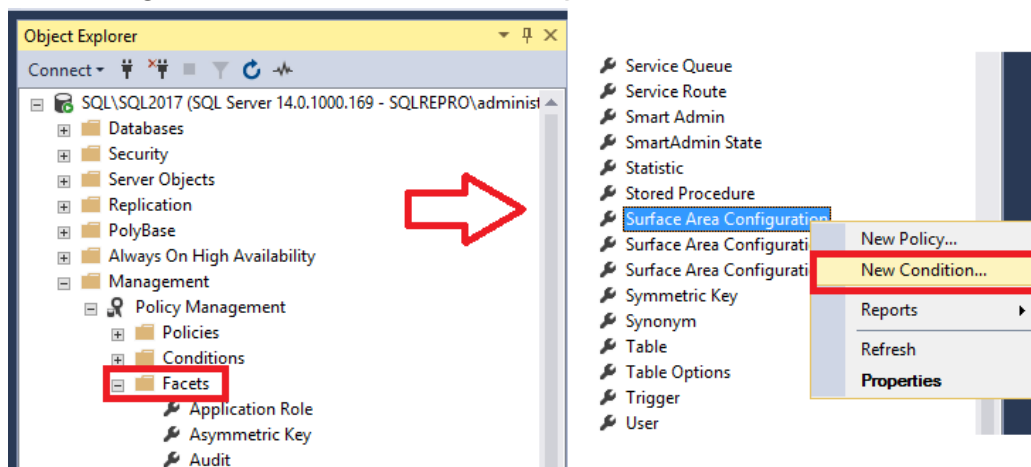
1. In **Object Explorer**, click the plus sign to expand the server where the newly-imported policy instance will reside.
2. Click the plus sign to expand the **Management** folder.
3. Click the plus sign to expand **Policy Management**.
4. Right-click the **Policies** folder and select **Import Policy**.
5. In the **Import** dialog box, type the path and name of the file; or use the Browse (...) button to locate the XML file that contains the policy, and then select the file.
6. When finished, click **OK**.

## BÀI 2

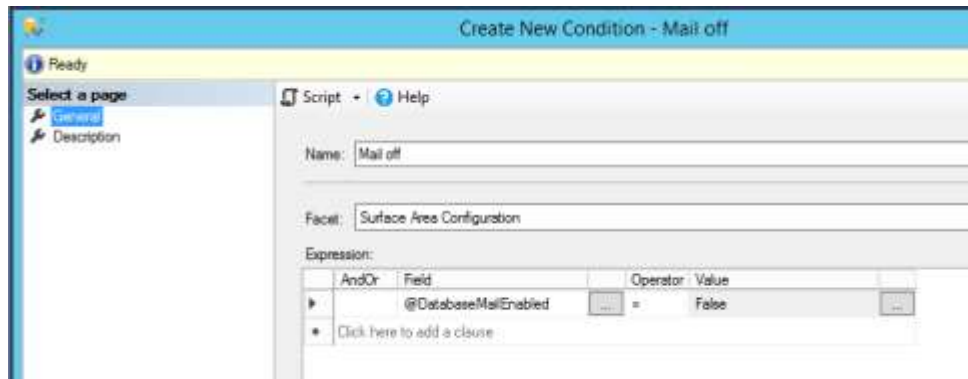
### Create and Apply an Off By Default Policy

#### Create the mail-off condition

1. In Object Explorer, expand **Management**, expand **Policy Management**, expand **Facets**, right-click **Surface Area Configuration**, and then click **New Condition**.

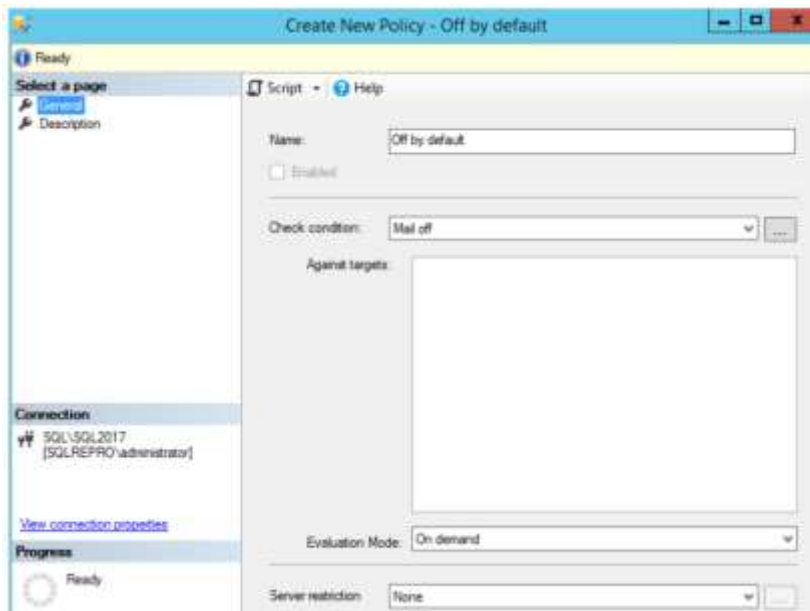


2. In the **Create New Condition** dialog box, in the **Name** box, type **Mail Off**.
  - a. In the **Facet** box, confirm that **Surface Area Configuration** facet is selected.
  - b. In the **Expression** area, in the **Field** box, select **@DatabaseMailEnabled**, in the **Operator** box select **=**, and in the **Value** select **False**.
  - c. On the **Description** page, type a description of the condition, and then click **OK** to create the condition.



### Create the off-by-default policy

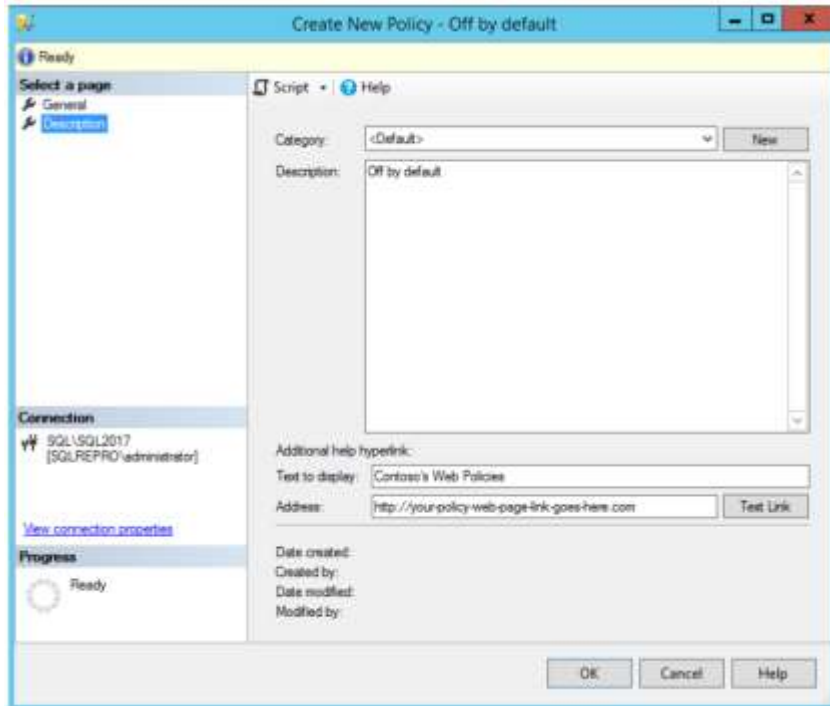
1. In Object Explorer, right-click **Surface Area Configuration**, and then click **New Policy**.
2. In the **Create New Policy** dialog box, in the **Name** box, type **Off By Default**.
  - a. Leave the **Enabled** checkbox unchecked. The **Enabled** checkbox applies to automated policies, and this policy will be executed on demand.
  - b. In the **Check condition** checkbox, scroll down to the **Surface Area Configuration** area, and then select **Mail Off** as the condition to check.
  - c. The **Against targets** box will be blank because this is a server-scoped policy.
  - d. In the **Evaluation Mode** checkbox, select **On demand** as the evaluation mode.
  - e. In the **Server restriction** checkbox, select **None**.
  - f. On the **Description** page, type a description of the policy.



On the description page, you can provide a hyperlink to a Web page for your policies in the **Additional help hyperlink** area. In the **Text to display** box, type the text that will appear for the hyperlink.

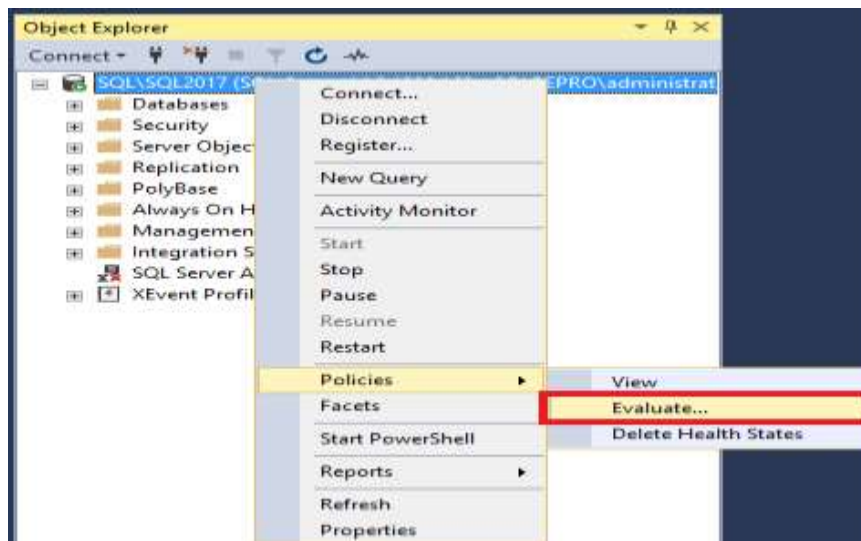


- In the **Address** box, type a hyperlink to a Help page, such as the home page for the IT department of your company.
- To confirm the address by opening the Web page, click **Test Link**.
- Click **OK**.



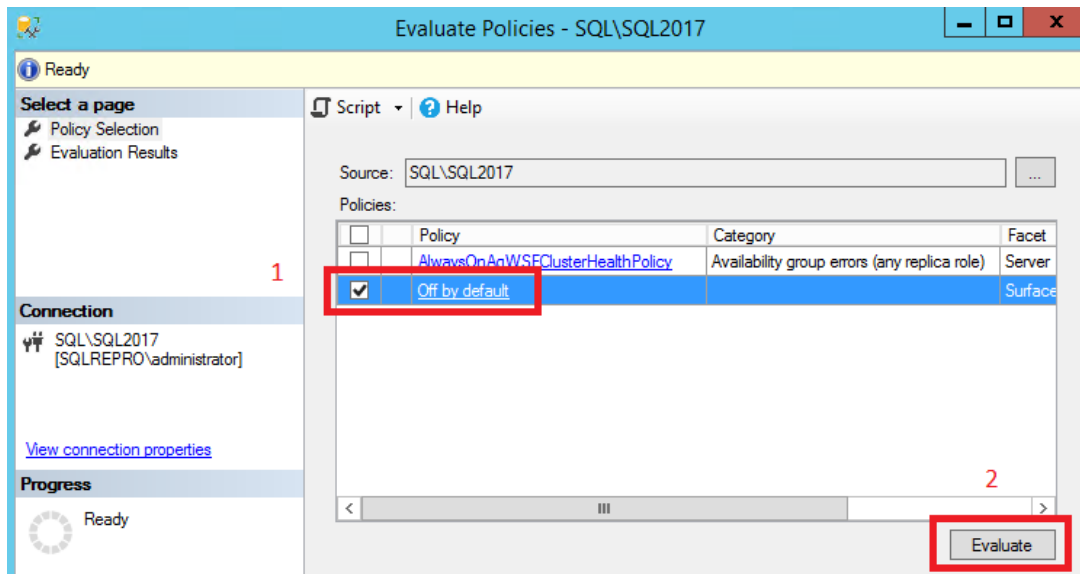
### Configure server to run off-by-default policy

- In Object Explorer, right-click your instance of SQL Server, point to **Policies**, and then click **Evaluate**.

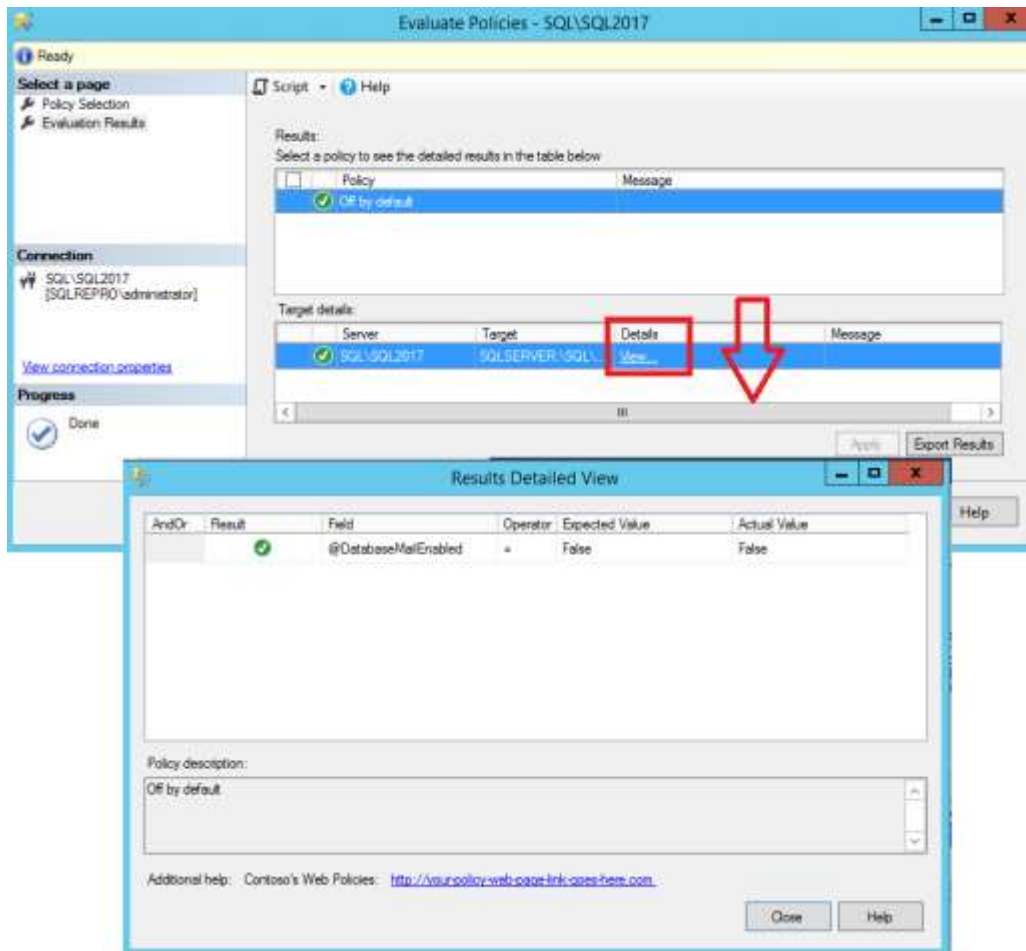




2. In the **Evaluate Policies** dialog box you can select policies from another instance of SQL Server or from a file. For this step, leave **Source** set to your instance of the Database Engine.
  - a. In the **Policies** section, select the **Off By Default** policy.
  - b. To see whether the server is in compliance with the policy, click **Evaluate**.
  - c. In the **Results** area, you will see a green circle with a check mark if the Database Engine complies with the policy. You will see a red circle with an X if the Database Engine does not comply with the policy.



7. In the **Target Details** area, you will see additional information in the **Message** column if an error occurs. In the **Message** column, click **View** to see a report that contains the results of the check for each facet property that was checked.



6. The policy description is displayed at the bottom of the page, and the **Additional help** section displays the hyperlink that you have configured for the policy. Click the message hyperlink to open the Web page that you specified when you created the policy.
7. Close the browser, and then close the **Results Detailed View** dialog box.
8. If the server is out of compliance and you want to disable Database Mail, click **Apply** in the **Evaluation Results** page.
9. Close both the **Results Detailed View** and the **Evaluate Policies** dialog boxes.

### BÀI 3

#### Create and Apply a Naming Standards Policy

##### Create the Finance database

1. In Management Studio, open a query window and execute the following statement:

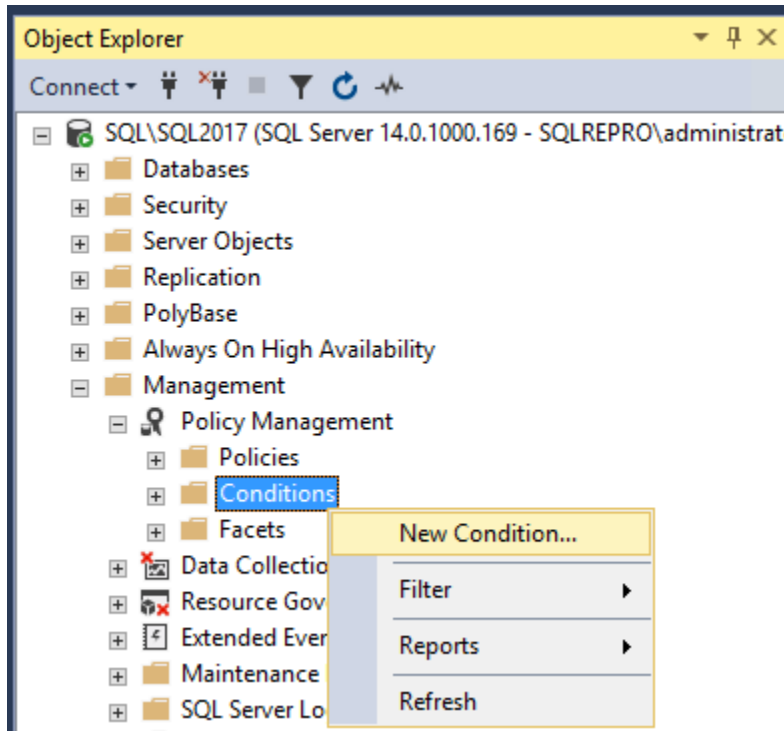
SQL Copy

```
CREATE DATABASE Finance ;
GO
```

2. In Object Explorer, click **Databases**, and then press F5 to refresh the list of databases.

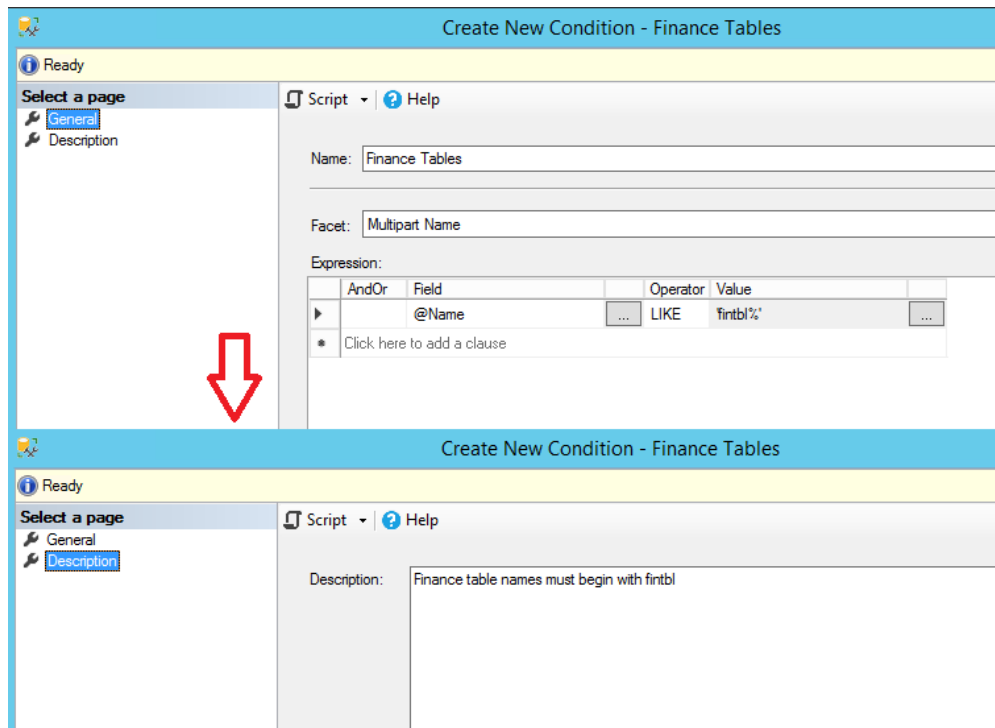
### Create the Finance tables condition

1. In Object Explorer, expand **Management**, expand **Policy Management**, right-click **Conditions**, and then click **New Condition**.



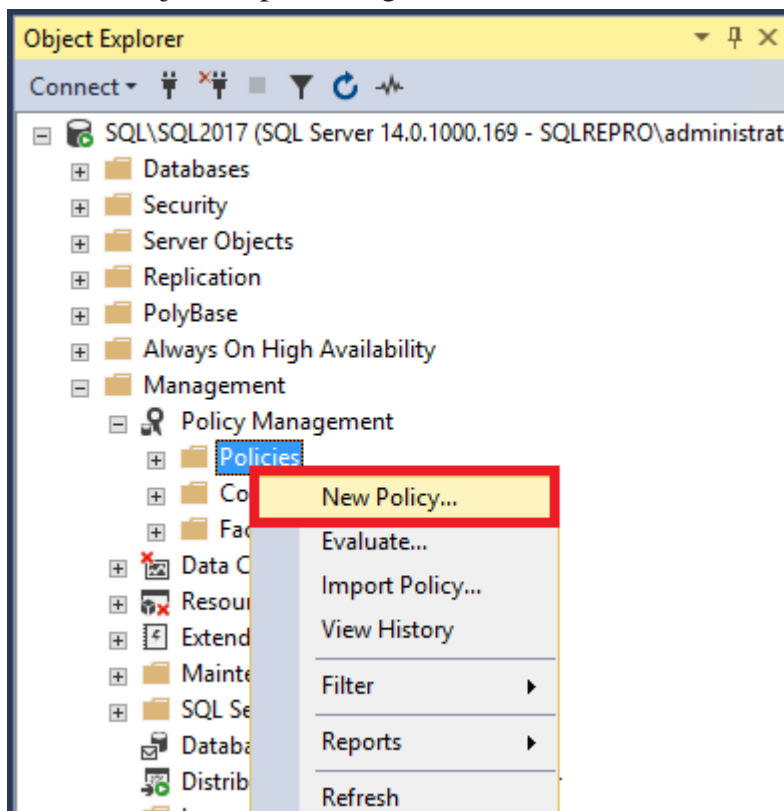
In the **Create New Condition** dialog box, in the **Name** box, type **Finance Tables**.

- a. In the **Facet** list, select **Multipart Name**.
- b. In the **Expression** area, in the **Field** box, select **@Name**; in the **Operator** box, select **Like**; and in the **Value** box, type 'fintbl%' to force all table names to start with the letters **fintbl**.
- c. On the **Description** page, type **Finance table names must begin with fintbl**, and then click **OK** to create the condition.



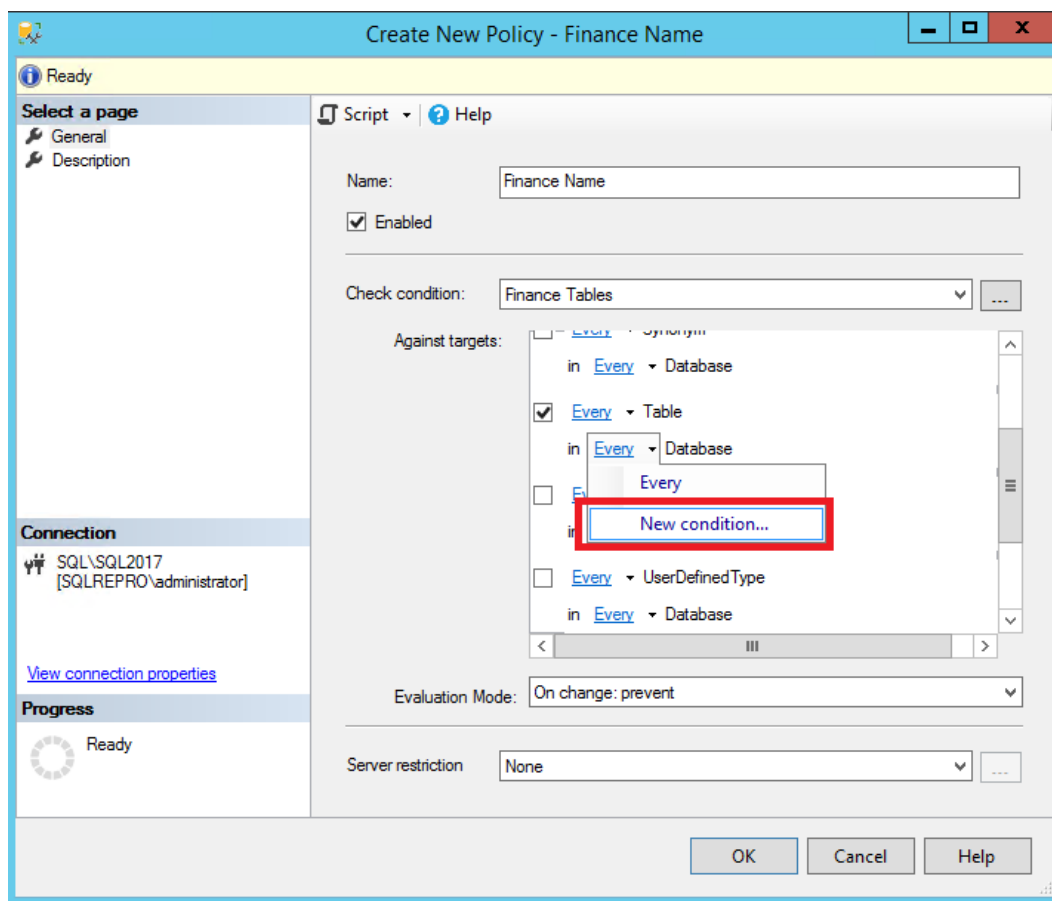
### Create the Finance name policy

1. In Object Explorer, right-click **Policies**, and then click **New Policy**.

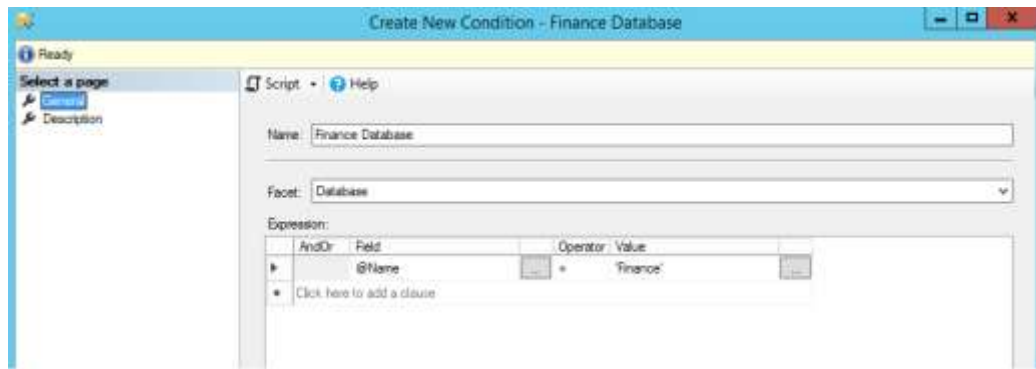


2. In the **Create New Policy** dialog box, in the **Name** box, type **Finance Name**.

- a. the **Check condition** list, select **Finance Tables**. This is in the **Multipart Name** area.
- b. In the **Against** area you will see a list of the database objects that could apply this policy. Select the check box for **Every Table**.
- c. Select the **Enabled** list. (The **Enabled** box does not apply to **On demand** policies.)
- d. In the **Evaluation Mode** list, select **On change: prevent**. This will enforce the policy by creating a database trigger on the Finance database.
- e. In the **Server restriction** list, select **None**.
- f. On the **Description** page, add the description 'Table names in the Finance database must contain 'fintbl%'.'
- g. Go back to the **General** page, and in the **Every Database** area, expand **Every**, and then click **New condition**.



2. In the **Create New Condition** dialog box, in the **Name** box, type **Finance Database**.
  - a. In the **Expression** box, complete the expression to include `@Name = 'Finance'`, and then click **OK** to close the condition page.



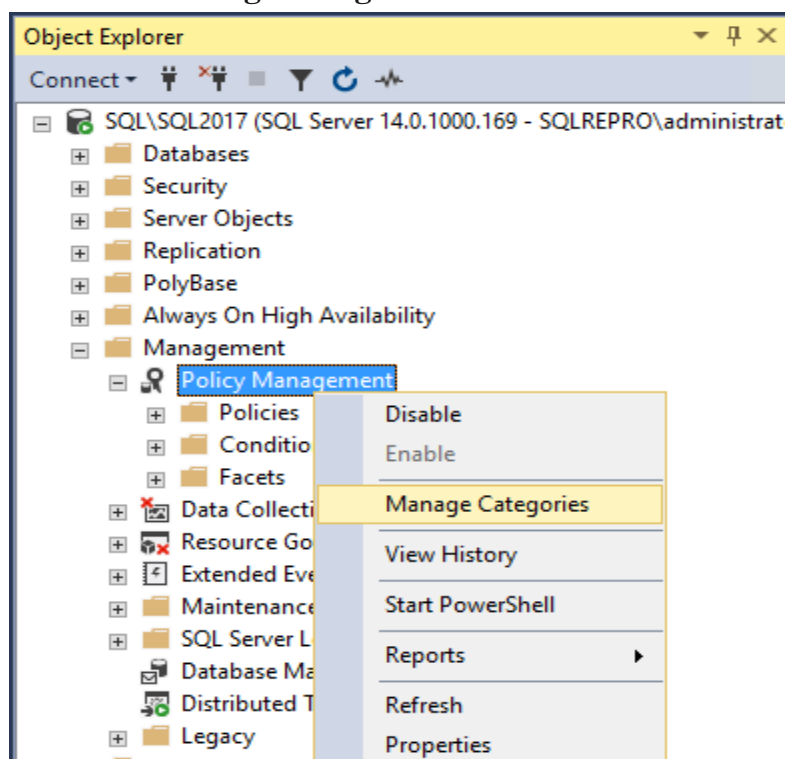
### Note

You might have to tab out of the **Value** box to enable the **OK** button.

Click **OK**.

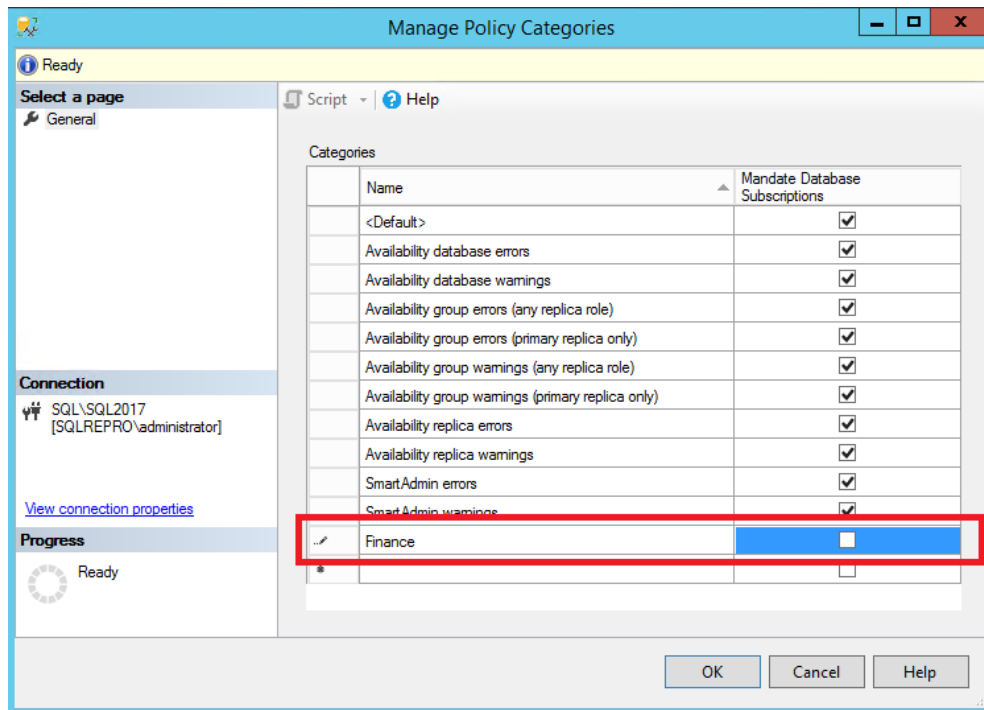
### Create the Finance policy category

1. In Object Explorer, expand **Management**, right-click **Policy Management**, and then click **Manage Categories**.
2. In Object Explorer, expand **Management**, right-click **Policy Management**, and then click **Manage Categories**.



3. In the **Manage Policy Categories** dialog box, under **Name**, type **Finance** in the blank box, and then clear **Mandate Database Subscriptions**. **Mandate Database Subscriptions** will force every database in the instance to subscribe to the policies that

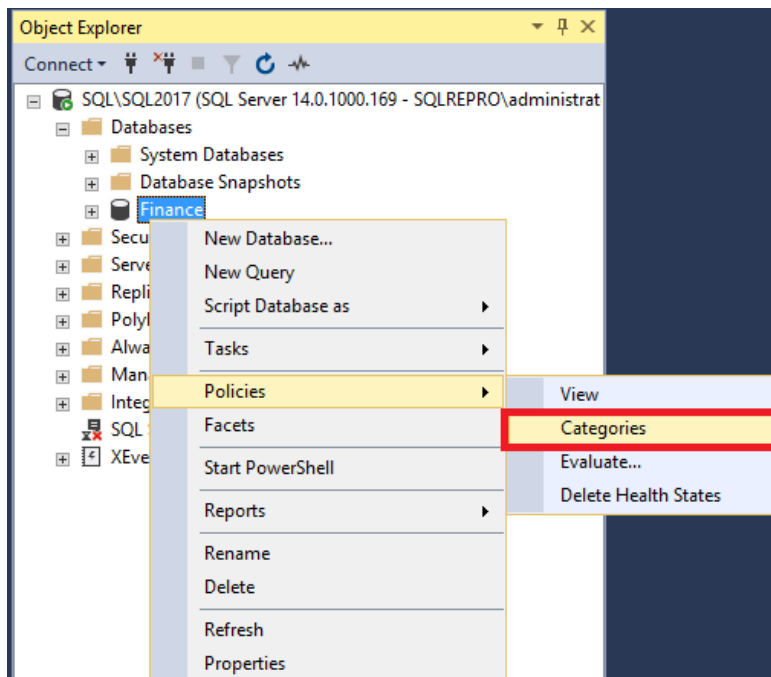
belong to this policy category. For this lesson, only the Finance database should subscribe to the Finance Name policy.



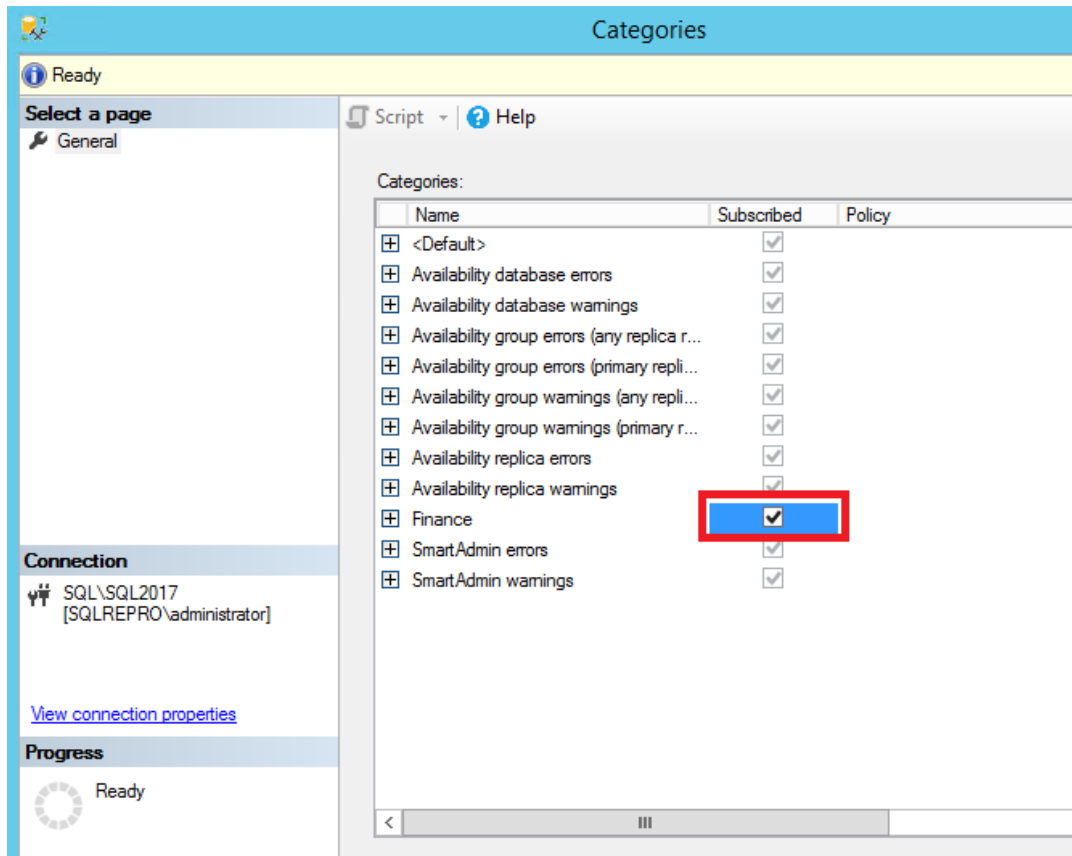
Click **OK**.

### Subscribe to the Finance policy category

1. In Object Explorer, expand **Databases**, right-click **Finance**, point to **Policies**, and then click **Categories**.



2. Select the **Subscribed** checkbox for the **Finance** category.



3. Click **OK**.

### Test the enforcement of the Finance Name policy

1. In Management Studio, open a query window. Execute the following statements that try to create a table that violates the **Finance Name** policy. The table violates the policy because the table name does not begin with the letters **fin**tbl.

SQL Copy

```
USE Finance ;
GO
CREATE TABLE NewTable
(Col1 int) ;
GO
```

Notice that the policy prevents the table from being created and returns an informational message that provides the policy name.

Copy

Policy 'Finance Name' has been violated by  
'SQLSERVER:\SQL\SQL\SQL2017\Databases\Finance\Tables\dbo.NewTable'.  
This transaction will be rolled back.



Policy condition: '@Name LIKE 'fintbl%'

Policy description: 'Tables names in the Finance database must contain 'fintbl%'.

Additional help: " : "

Statement: 'CREATE TABLE NewTable  
(Col1 int)'.

Msg 515, Level 16, State 2, Procedure msdb.sys.sp\_syspolicy\_execute\_policy, Line 69  
[Batch Start Line 2]

Cannot insert the value NULL into column 'target\_query\_expression', table  
'msdb.dbo.syspolicy\_policy\_execution\_history\_details\_internal'; column does not allow  
nulls. INSERT fails.

The statement has been terminated.

To provide a valid name, modify the code as follows and run the statement again.

SQL

USE Finance ;

GO

CREATE TABLE fintblNewTable

(Col1 int) ;

GO

This time, the table is created.

### Apply the policy to the whole server

1. Currently, only the Finance database subscribes to the Finance policy category. In many cases, it is easier to apply the policy category to the whole server. In Object Explorer, expand **Management**, right-click **Policy Management**, and then click **Manage Categories**.
2. In the **Manage Policy Categories** dialog box, locate the Finance category, and select the **Mandate Database Subscriptions** checkbox for the Finance category.
3. Click **OK**. Now the Finance category applies to all databases, but the condition that you have created restricts the Finance Name policy to the Finance database. This shows how you can use complex combinations of conditions to target policies in ways that will apply correctly on many servers.

**Bài tập thực hành tuần 10**

**Ôn Tập – Kiểm tra**