

SÁCH HƯỚNG DẪN

BÀI TẬP HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU



Trình độ đào tạo: Đại học

Nhóm biên soạn:

TS. Phạm Văn Hà (*Chủ biên*)
ThS. Trần Thanh Hùng
ThS. Nguyễn Thị Thanh Huyền
ThS. Nguyễn Thị Hương Lan
ThS. Đỗ Ngọc Sơn

Hà Nội - 06/2020

LỜI NÓI ĐẦU

Hệ quản trị cơ sở dữ liệu (DBMS – Database Management System) được tạo ra để quản lý một khối lượng dữ liệu nhất định một cách tự động và có trật tự, bao gồm dữ liệu quan hệ hoặc phi quan hệ. Có rất nhiều hệ quản trị cơ sở dữ liệu (CSDL) được sử dụng hiện nay như Oracle, MySQL, SQL Server, MongoDB,... nhưng thông dụng nhất và được sử dụng nhiều nhất để giảng dạy và học tập là hệ quản trị CSDL Microsoft SQL Server.

Trong khuôn khổ của cuốn sách hướng dẫn *Bài tập hệ quản trị cơ sở dữ liệu* này, chúng tôi mong muốn cung cấp một tài liệu giúp người học có được những kiến thức cơ bản làm cơ sở cho kỹ thuật xử lý dữ liệu trên SQL Server và cung cấp cho người học hệ thống bài tập đa dạng trải rộng đầy đủ trên toàn bộ kiến thức SQL, có hướng dẫn giải mẫu các bài tập và có các bài tập tự giải cho sinh viên, giúp các em sinh viên hiểu sâu hơn về lý thuyết và có kỹ năng tốt hơn về thực hành trên SQL Server.

Xuyên suốt cuốn bài tập thực hành này là một CSDL duy nhất được tạo và quản trị từ đầu đến cuối chương trình, giúp ích cho các em sinh viên có được một cái nhìn tổng quan, bao quát hơn về một hệ thống quản lý đầy đủ, xuyên suốt. Qua đó giúp cho các em có thể xây dựng được một hệ CSDL quản lý tương tự và áp dụng trong việc lập trình các chương trình quản lý sau này.

Cuốn sách hướng dẫn *Bài tập hệ quản trị cơ sở dữ liệu* được biên soạn bởi các tác giả:

- TS. Phạm Văn Hà (chủ biên),
- ThS. Trần Thanh Hùng,
- ThS. Nguyễn Thị Thanh Huyền,
- ThS. Nguyễn Thị Hương Lan,
- ThS. Đỗ Ngọc Sơn.

Cuốn sách bao gồm có 10 bài thực hành xuyên suốt nội dung giáo trình, trong đó mỗi bài được chia làm hai phần: phần một nhắc lại lý thuyết đã học, phần hai là phần bài tập thực hành, trong đó bao gồm phần thực hành mẫu giúp cho các bạn sinh viên có thể hiểu được quy trình thực hành, và phần tự thực hành giúp các em sinh viên củng cố kiến thức, kỹ năng.

Mặc dù đã có nhiều cố gắng trong việc biên soạn cuốn bài tập làm sao cho chi tiết và đầy đủ nhất giúp cho các em sinh viên dễ dàng học tập và áp dụng nhất, tuy nhiên cuốn sách vẫn đang còn nhiều thiếu sót cần bổ sung, rất mong có được các ý kiến phản hồi từ các bạn đồng nghiệp và các em sinh viên giúp cho chúng tôi hoàn thiện cuốn sách hơn ở các lần tái bản sau.

CÁC TÁC GIẢ

MỤC LỤC

LỜI NÓI ĐẦU	1
BÀI THỰC HÀNH SỐ 1	6
BÀI THỰC HÀNH SỐ 2	23
BÀI THỰC HÀNH SỐ 3	30
BÀI THỰC HÀNH SỐ 4	34
BÀI THỰC HÀNH SỐ 5	38
BÀI THỰC HÀNH SỐ 6	43
BÀI THỰC HÀNH SỐ 7	50
BÀI THỰC HÀNH SỐ 8	57
BÀI THỰC HÀNH SỐ 9	64
BÀI THỰC HÀNH SỐ 10.....	75
MỘT SỐ ĐỀ ÔN TẬP THAM KHẢO	83
TÀI LIỆU THAM KHẢO	93

BÀI THỰC HÀNH SỐ 1

1. Cho CSDL QLBanHang bao gồm các bảng sau:

Sanpham(masp, mahangsx, tensp, soluong, mausac, giaban, donvitinh, mota)

Hangsx(mahangsx, tenhang, diachi, sodt, email)

Nhanvien(manv, tenv, gioitinh, diachi, sodt, email, phong)

Nhap(sohdn, masp, manv, ngaynhap, soluongN, dongiaN)

Xuat(sohdx, masp, manv, ngayxuat, soluongX)

- Trong đó: Bảng Sanpham – Sản phẩm, Hangsx – Hàng sản xuất, Nhanvien – Nhân viên, Nhập – Nhập, Xuat – Xuất.

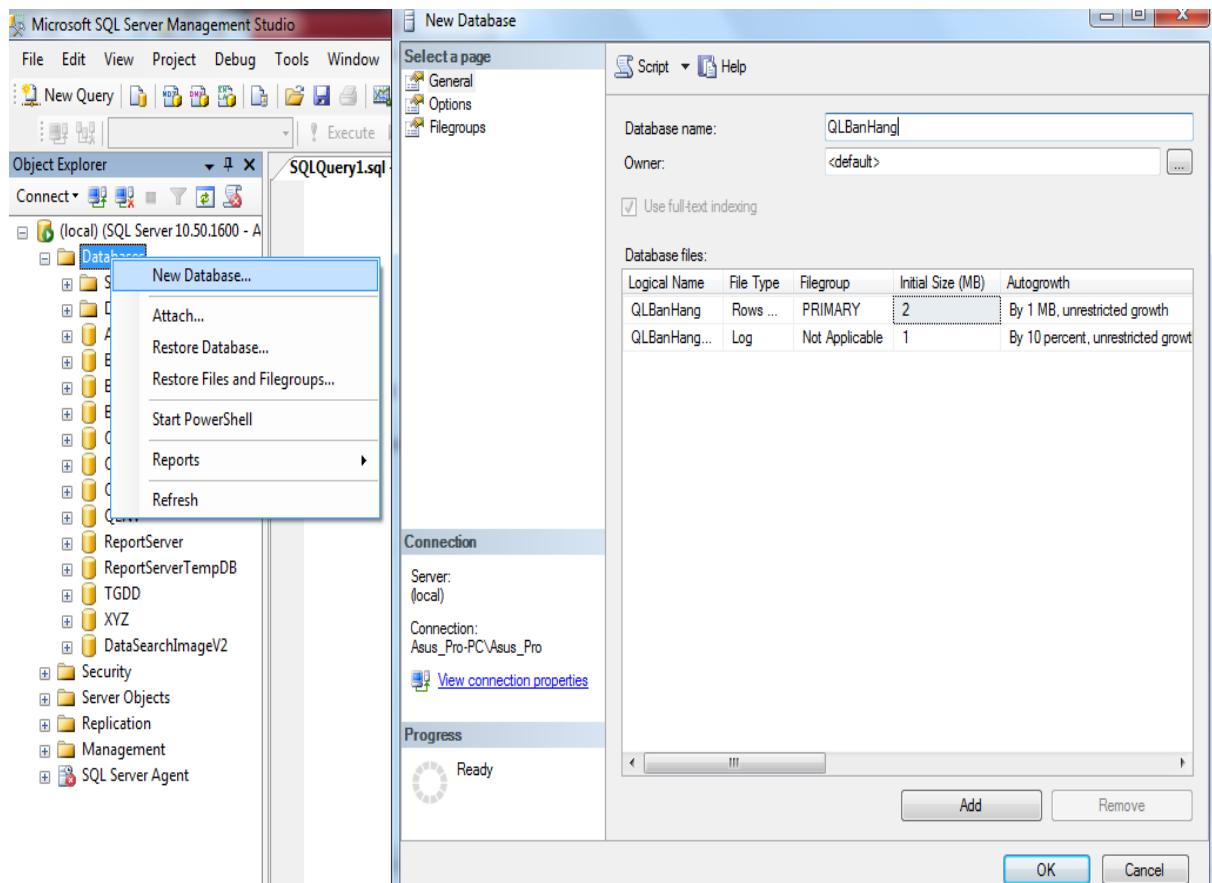
- Hãy tạo/xóa/đọc thông tin CSDL quản lý bán hàng.
- Tạo các bảng dữ liệu, đưa ra lược đồ quan hệ (Database Diagrams).
- Nhập dữ liệu cho các bảng trên.
- Đưa ra dữ liệu vừa nhập

Yêu cầu: Thực hiện các công việc trên bằng Câu lệnh (Query Analyse - QA) và bằng công cụ (Enterprise Management - EM).

2. Hướng dẫn thực hành:

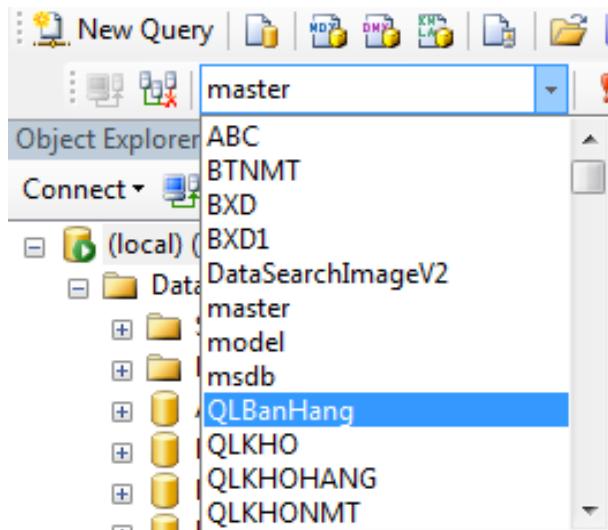
A. Sử dụng EM:

- Tạo/xóa cơ sở dữ liệu bằng EM:



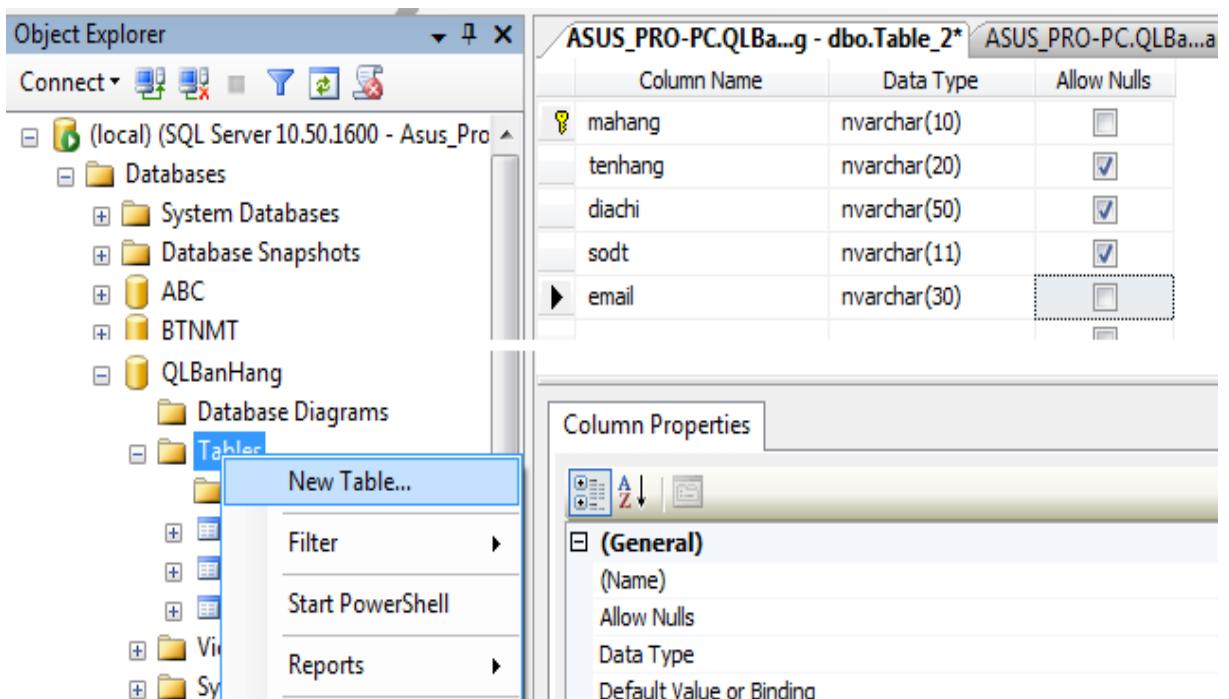
Hình 1.1. Tạo mới CSDL bằng New Database wizard

- Chọn các trường:
 - Initial Size (MB): Kích thước khởi tạo
 - Autogrowth: Mức tăng trưởng
 - path: Đường dẫn nơi lưu file.
- Xóa: Click chuột phải vào tên CSDL vừa tạo (QLBanHang), chọn Delete.
- Sau khi đã tạo xong CSDL QLBanHang, ta phải sử dụng CSDL QLBanHang làm CSDL hiện hành, để tạo các công việc khác.



Hình 1.2. Chọn CSDL hiện hành

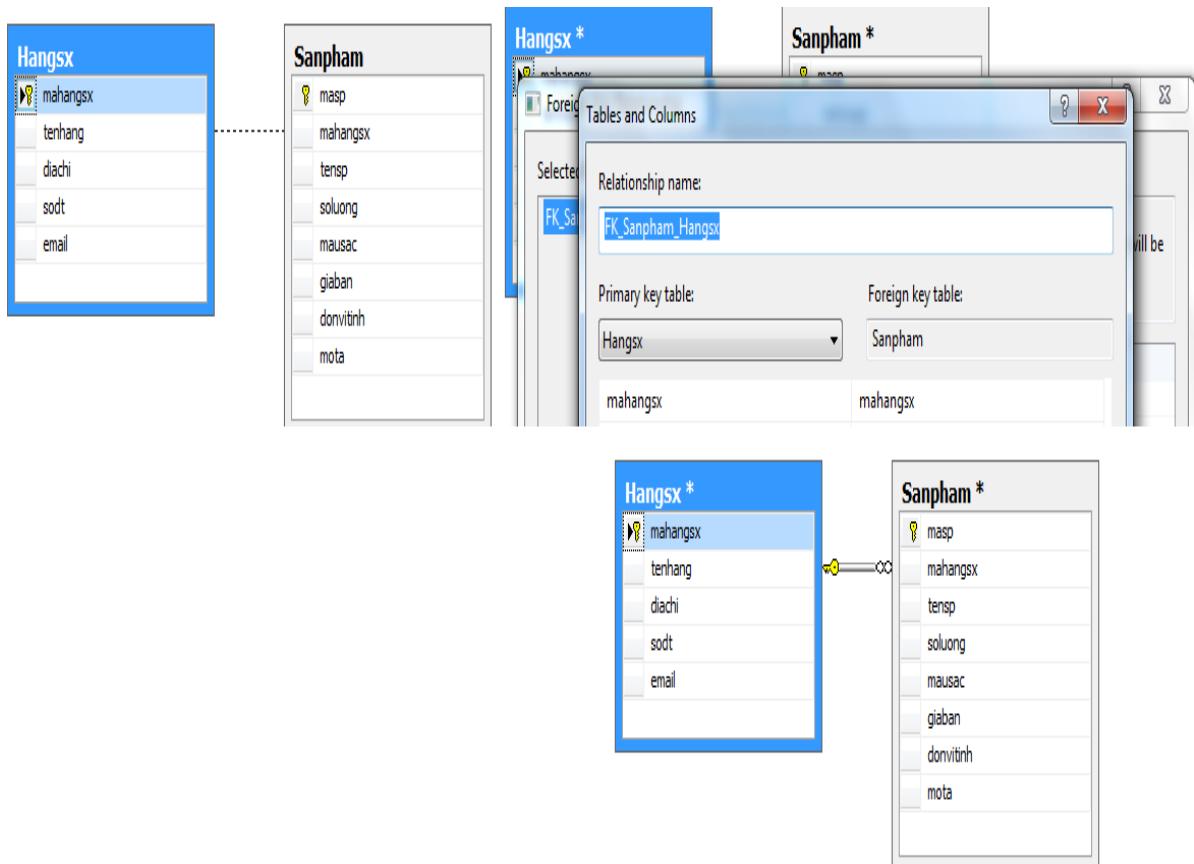
- Tạo/ xóa bảng bằng EM:



Hình 1.3. Tạo mới bảng wizard

- Tạo các bảng theo thứ tự bất kỳ, chưa tạo liên kết.
 - Một số kiểu dữ liệu thông dụng:
 - Int – số nguyên

- Float - số thực
- Double – số thực động
- Char(n) – Kiểu ký tự không dấu
- Nchar(n) – Kiểu ký tự có dấu định dạng unicode
- Varchar(n) – chuỗi ký tự
- Nvarchar(n) – chuỗi ký tự có dấu định dạng unicode
- Text – chuỗi văn bản
- Ntext – chuỗi văn bản có dấu định dạng unicode
- Money – kiểu tiền tệ
- Date – kiểu ngày/tháng/năm
- Datetime – kiểu ngày/tháng/năm/giờ/phút/giây
- ...
- Liên kết khóa ngoại (Foreign key) cho bảng:
 - ✓ Click chuột phải lên: Database Diagram -> New Database Diagram
 - ✓ Trên màn hình diagram: click chuột phải -> Add tables -> Chọn tất cả các table.
 - ✓ Kéo trường thuộc tính khóa ngoại của bảng này (Click chuột trái và kéo thả) tới thuộc tính của bảng khác mà nó liên kết khóa ngoại.



Hình 1.4. Liên kết bảng diagram

- ✓ Lưu lại Diagram vừa tạo liên kết.
- Xóa bảng: Sau khi đã tạo liên kết diagram, xóa bảng thì phải xóa liên kết, click chuột phải trên liên kết ở diagram -> Delete relationship from database -> xóa

hết liên kết tới bảng cần xóa, sau đó vào Database -> click chuột lên bảng cần xóa -> delete.

- Sửa lại nội dung bảng:

- ✓ Chọn bảng cần sửa nội dung -> Design -> sửa -> lưu lại bảng đã sửa.

Column Name	Data Type	Allow Nulls
masp	nvarchar(10)	<input type="checkbox"/>
mahangsx	nvarchar(10)	<input checked="" type="checkbox"/>
tensp	nvarchar(20)	<input checked="" type="checkbox"/>
soluong	int	<input checked="" type="checkbox"/>
mausac	nvarchar(20)	<input checked="" type="checkbox"/>
giaban	float	<input checked="" type="checkbox"/>
donvitinh	nvarchar(10)	<input checked="" type="checkbox"/>
mota	nvarchar(MAX)	<input checked="" type="checkbox"/>

Hình 1.5. Chỉnh sửa cấu trúc bảng wizard

- Nhập/xóa dữ liệu cho các bảng:

- ✓ Nhập dữ liệu theo nguyên tắc: Bảng 1 nhập trước, bảng n nhập sau, xóa dữ liệu theo chiều ngược lại.
- ✓ Bảng có thuộc tính khóa ngoại (bảng n), ở trường khóa ngoại sử dụng lại dữ liệu ở bảng (bảng 1) mà nó liên kết tới.

- Nhập/xóa dữ liệu cho bảng bằng EM:

	mahangsx	tenhang	diachi	sdt	email
*	H01	Samsung	Seoul - Korea Public	0111-012	samsung@gmail.com.kr
NULL	NULL	NULL	NULL	NULL	NULL

Hình 1.6. Nhập dữ liệu mới cho bảng wizard

- Xem dữ liệu các bảng bằng EM:

```

/*
***** Script for SelectTopNRows *****
*/
SELECT TOP 1000 [mahangsx]
      ,[tenhang]
      ,[diachi]
      ,[sdt]
      ,[email]
  FROM [QLBanHang].[dbo].[Hangsx]

```

Hangsx	tenhang	diachi	sdt	email
Samsung	Seoul - Korea Public	0111-012	samsung@gmail.com.kr	

Hình 1.7. Lấy dữ liệu bảng wizard

B. Sử dụng QA:

- Để đọc thông tin các CSDL đã có trước đó trong hệ thống:

```

use master
go
select * from Sysdatabases

```

Để lấy thông tin về 1 CSDL cụ thể:

```
use master
go
select * from Sysdatabases where name='QLBanHang'
```

Để xóa 1 CSDL có sẵn, để tạo CSDL mới:

```
use master
go
if(exists(select * from Sysdatabases where name='QLBanHang'))
    Drop database QLBanHang
```

2. Tạo/xóa CSDL QLBanHang:

- Tạo DB bằng QA:

- Chọn New Query --> Trên màn hình Editor ta soạn:

```
create database QLBanHang
on primary(
    name='QLBanHang_Dat',
    filename= 'D:\QLBanHang.mdf',
    size=2MB,
    maxsize=10MB,
    filegrowth=20%)
log on(
    name= 'QLBanHang_log',
    filename= 'D:\QLBanHang.ldf',
    size=1MB,
    maxsize=5MB,
    filegrowth=1MB
)
go
```

- Bôi đen bó lệnh (Batch) trên và thực thi (Chọn button Execute) → Refresh trên Explore Database ta sẽ thấy được CSDL vừa tạo.

- Sử dụng CSDL vừa tạo làm CSDL hiện hành:

```
Use QLBanHang
Go
```

3. Tạo/xóa bảng dữ liệu:

- Tạo/ xóa bảng bằng QA:

- Tạo/xóa các bảng dữ liệu:
 - ✓ Nguyên tắc
 - Tạo bảng 1 trước, tạo bảng nhiều sau (Bảng chứa khóa ngoại).
 - Xóa các bảng theo nguyên tắc ngược lại.
 - ✓ Với các nguyên tắc này các bảng trên được tạo theo thứ tự sau:

- HangSX
 - Nhanvien
 - Sanpham
 - Nhaph
 - Xuat
- ✓ Xác định khóa chính (Primary key), khóa ngoại (Foreign key) của mỗi bảng để tạo liên kết.
- ✓ Nếu khóa chính là khóa có nhiều thuộc tính thì nên tạo constraint ràng buộc khóa chính:
- Constraint PK1 primary key(A,B,C)
- ✓ Trường thuộc tính khóa ngoại và thuộc tính khóa ở bảng liên kết với nó phải có dữ liệu, kích thước, khai báo giống nhau, nếu không giống nhau sẽ bị báo lỗi.
- ✓ Các tên ràng buộc khóa chính, khóa ngoại, check, trong quá trình tạo bảng không được trùng nhau.
- ✓ Ví dụ:

```

create table Hangsx(
    mahangsx nvarchar(10) not null primary key,
    tenhang nvarchar(20) not null,
    diachi nvarchar(30) not null,
    sodt nvarchar(20) not null,
    email nvarchar(30) not null
)
create table Sanpham(
    masp nvarchar(10) not null primary key,
    mahangsx nvarchar(10) not null,
    tensp nvarchar(20) not null,
    soluong int default 0 not null,
    mausac nvarchar(20),
    giaban money,
    donvitinh nvarchar(10),
    mota nvarchar(max),
    constraint fk_sanpham_hangsx foreign key(mahangsx) references
Hangsx(mahangsx) on update cascade on delete cascade
)

```

- Xóa bảng:
 - ✓ Nguyên tắc bảng nào tạo trước thì xóa sau, tức là bảng liên kết n xóa trước, bảng liên kết 1 xóa sau.
 - ✓ Để xóa 1 bảng, ta phải xóa các liên kết khác tới bảng đó.

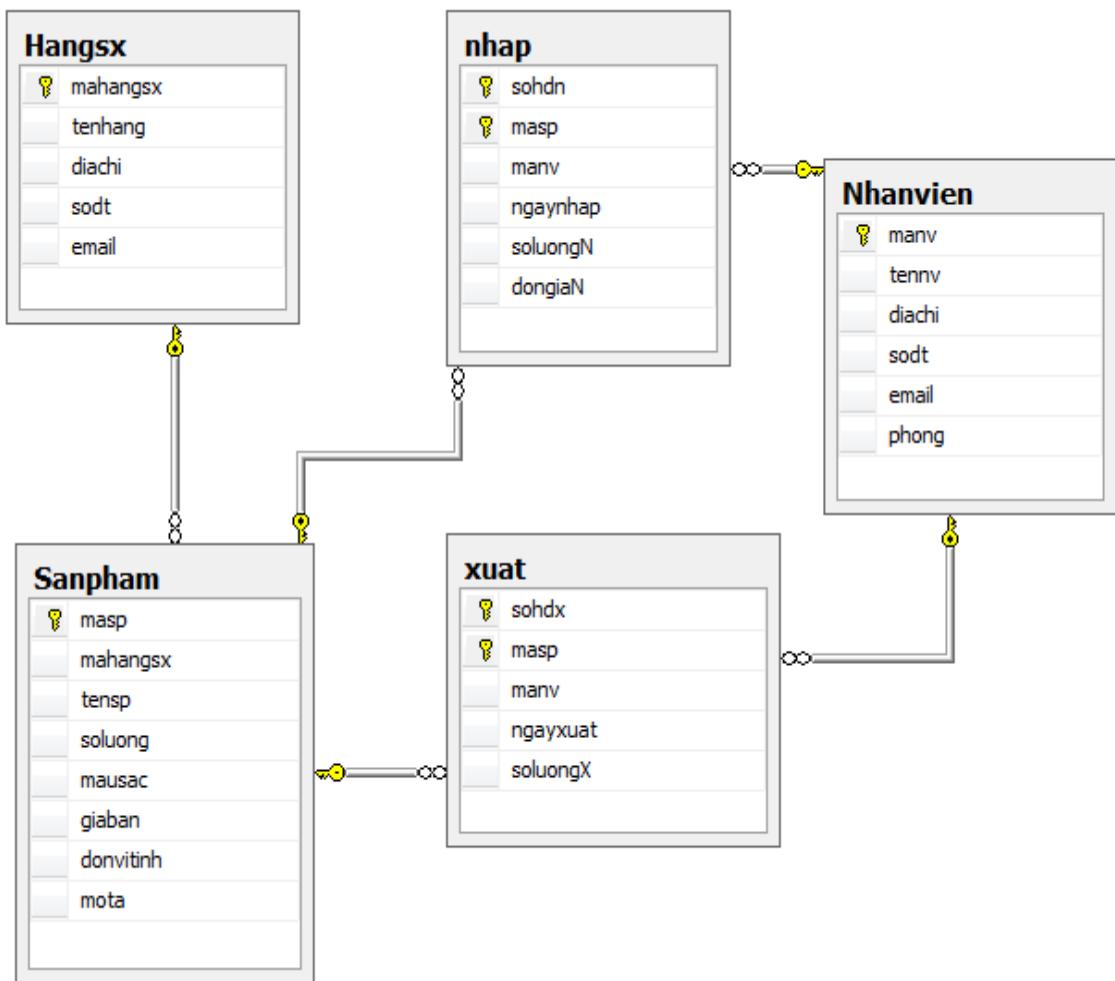
ALTER TABLE Sanpham DROP CONSTRAINT FK_Sanpham_Hangsx

- Sau đó mới xóa bảng:

Drop table Tenbang

4. Tạo lược đồ quan hệ (Database Diagrams)

- Sau khi đã tạo xong các bảng và các liên kết khóa chính, khóa ngoại, đưa ra lược đồ quan hệ Database Diagrams đã nói ở phần tạo liên kết khóa ngoại bằng EM ở trên.



Hình 1.8. Lược đồ quan hệ Diagram CSDL QLBanHang

5. Nhập/xóa dữ liệu cho các bảng:

- Nhập dữ liệu theo nguyên tắc: Bảng 1 nhập trước, bảng n nhập sau, xóa dữ liệu theo chiều ngược lại.
- Bảng có thuộc tính khóa ngoại (bảng n), ở trường khóa ngoại sử dụng lại dữ liệu ở bảng (bảng 1) mà nó liên kết tới.
- Nhập/xóa dữ liệu cho bảng QA:

- Nhập dữ liệu cho bảng QA:

```
Insert into Hangsx values('H01','Samsung','Seoul-Korea
Public','0111-012','samsung@gmail.com.kr')
```

- ✓ Chú ý: chuỗi ký tự viết trong dấu ‘’, còn chữ số không cần.
 - Xóa dữ liệu bằng QA:

```
Delete from bảng where điều-kiện-xóa
```

6. Xem dữ liệu các bảng:

- Xem dữ liệu các bảng bằng QA:

Select * from tenbang

3. Thực hành tạo CSDL QLBanHang trên với 2 phương pháp EM và QA, với thiết kế bảng như sau:

Bảng 1.1. Bảng Sanpham

Sanpham				
Trường	Kiểu dữ liệu	Kích thước	Khóa	Mô tả
<u>masp</u>	Nchar	10	Primary key	Mã sản phẩm
mahangsx	nchar	10	Foreign Key – References Hangsx(mahangsx)	Mã hãng sản xuất
tensp	Nvarchar	20		Tên sản phẩm
soluong	Int			Số lượng
mausac	Nvarchar	20		Màu sắc
giaban	Money			Giá bán
donvitinh	Nchar	10		Đơn vị tính
mota	Nvarchar	Max		Mô tả

Bảng 1.2. Bảng Hangsx

Hangsx				
Trường	Kiểu dữ liệu	Kích thước	Khóa	Mô tả
<u>Mahangsx</u>	Nchar	10	Primary key	Mã hãng sản xuất
Tenhang	Nvarchar	20		Tên hãng sản xuất
Diachi	Nvarchar	30		Địa chỉ hãng
Sodt	Nvarchar	20		Số điện thoại
email	Nvarchar	30		email

Bảng 1.3. Bảng Nhanvien

Nhanvien				
Trường	Kiểu dữ liệu	Kích thước	Khóa	Mô tả
<u>Manv</u>	Nchar	10	Primary key	Mã nhân viên
Tenv	Nvarchar	20		Tên nhân viên
Gioitinh	Nchar	10		Giới tính
Diachi	Nvarchar	30		Địa chỉ
Sodt	Nvarchar	20		Số điện thoại
email	nvarchar	30		email
Phong	nvarchar	30		Phòng ban

Bảng 1.4. Bảng Nhap

Nhap				
Trường	Kiểu dữ liệu	Kích thước	Khóa	Mô tả
<u>Sohdn</u>	Nchar	10	Primary key(sohdn,masp)	Số hóa đơn nhập
<u>Masp</u>	Nchar	10	Foreign key – references Sanpham(masp)	Mã sản phẩm
Manv	Nchar	10	Foreign key – references Nhanvien(manv)	Mã nhân viên
Ngaynhap	date			Ngày nhập
soluongN	int	30		Số lượng nhập
dongiaN	money			Đơn giá nhập

Bảng 1.5. Bảng Xuat

Xuat				
Trường	Kiểu dữ liệu	Kích thước	Khóa	Mô tả
Sohdx	Nchar	10	Primary key(sohdx,masp)	Số hóa đơn xuất
Masp	Nchar	10	Foreign key – references Sanpham(masp)	Mã sản phẩm
Manv	Nchar	10	Foreign key – references Nhanvien(manv)	Mã nhân viên
Ngayxuat	date			Ngày nhập
soluongX	int	30		Số lượng nhập

- Bộ dữ liệu cho các bảng nhập theo mẫu:

Bảng 1.6. Dữ liệu bảng Hangsx

Hangsx				
<u>mahangsx</u>	<i>tenhang</i>	<i>diachi</i>	<i>sodt</i>	<i>email</i>
H01	Samsung	Korea	011-08271717	ss@gmail.com.kr
H02	OPPO	China	081-08626262	oppo@gmail.com.cn
H03	Vinfone	Việt nam	084-098262626	vf@gmail.com.vn

Bảng 1.7. Dữ liệu bảng Nhanvien

Nhanvien						
<u>manv</u>	<i>tennv</i>	<i>gioitinh</i>	<i>diachi</i>	<i>sodt</i>	<i>email</i>	<i>phong</i>
NV01	Nguyễn Thị Thu	Nữ	Hà Nội	0982626521	thu@gmail.com	Kế toán
NV02	Lê Văn Nam	Nam	Bắc Ninh	0972525252	nam@gmail.com	Vật tư
NV03	Trần Hòa Bình	Nữ	Hà Nội	0328388388	hb@gmail.com	Kế toán

Bảng 1.8. Dữ liệu bảng sanpham

Sanpham							
<u>masp</u>	<i>mahangsx</i>	<i>tensp</i>	<i>soluong</i>	<i>mausac</i>	<i>giaban</i>	<i>donvitinh</i>	<i>mota</i>
SP01	H02	F1 Plus	100	Xám	7000000	Chiếc	Hàng cận cao cấp
SP02	H01	Galaxy Note11	50	Đỏ	19000000	Chiếc	Hàng cao cấp
SP03	H02	F3 lite	200	Nâu	3000000	Chiếc	Hàng phổ thông
SP04	H03	Vjoy3	200	Xám	1500000	Chiếc	Hàng phổ thông
SP05	H01	Galaxy V21	500	Nâu	8000000	Chiếc	Hàng cận cao cấp

Bảng 1.9. Dữ liệu bảng Nhap

Nhap					
<u>sohdn</u>	<u>masp</u>	<i>manv</i>	<i>ngaynhap</i>	<i>soluongN</i>	<i>donggiaN</i>
N01	SP02	NV01	02-05-2019	10	17000000
N02	SP01	NV02	04-07-2020	30	6000000
N03	SP04	NV02	05-17-2020	20	1200000
N04	SP01	NV03	03-22-2020	10	6200000
N05	SP05	NV01	07-07-2020	20	7000000

Bảng 1.10. Dữ liệu bảng Xuat

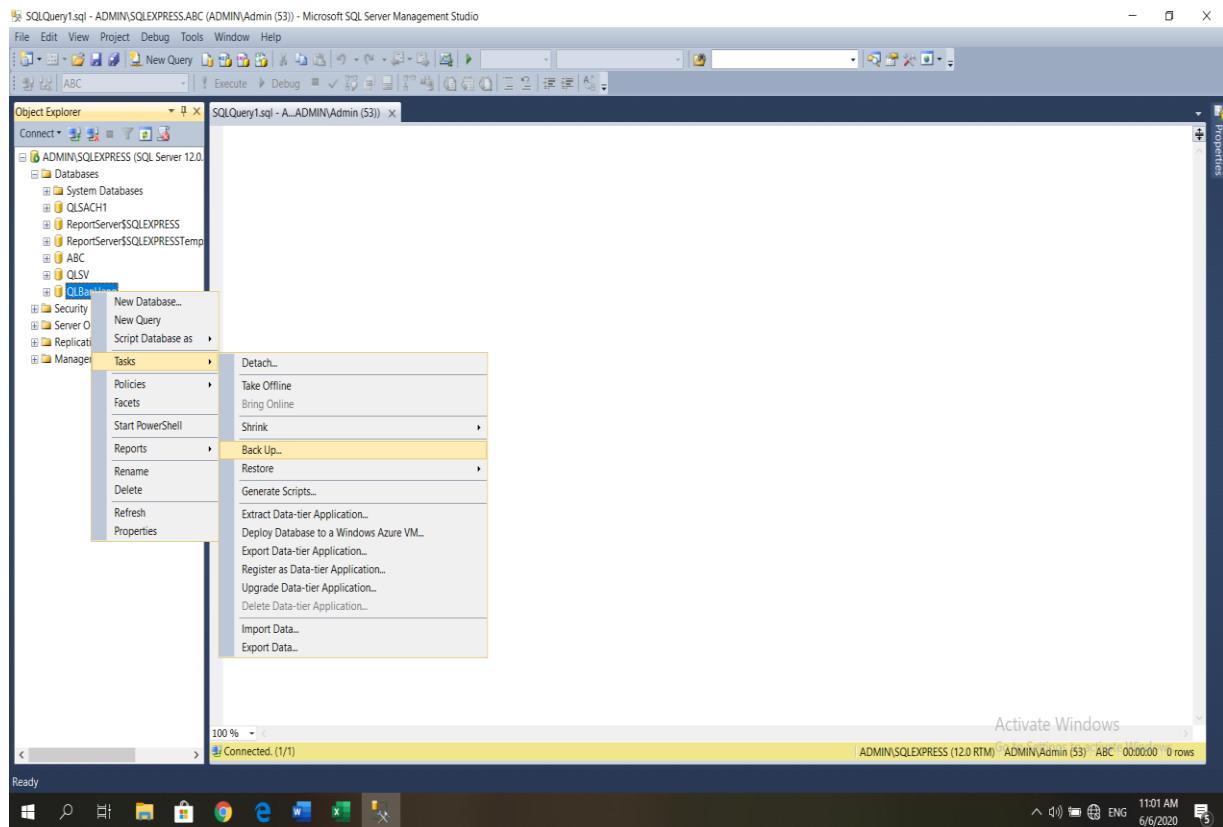
Xuat	<u>sohdX</u>	<u>masp</u>	<u>manv</u>	<u>ngayxuat</u>	<u>soluongX</u>
X01	SP03	NV02	06-14-2020	5	
X02	SP01	NV03	03-05-2019	3	
X03	SP02	NV01	12-12-2020	1	
X04	SP03	NV02	06-02-2020	2	
X05	SP05	NV01	05-18-2020	1	

4. Restore và Backup dữ liệu:

- Sau khi đã tạo xong CSDL QLBanHang, để sử dụng lại CSDL trong các buổi thực hành tiếp theo hoặc lấy CSDL đó cài vào máy khác mà không phải mất công tạo lại thì chúng ta cần phải sao lưu và phục hồi dữ liệu.

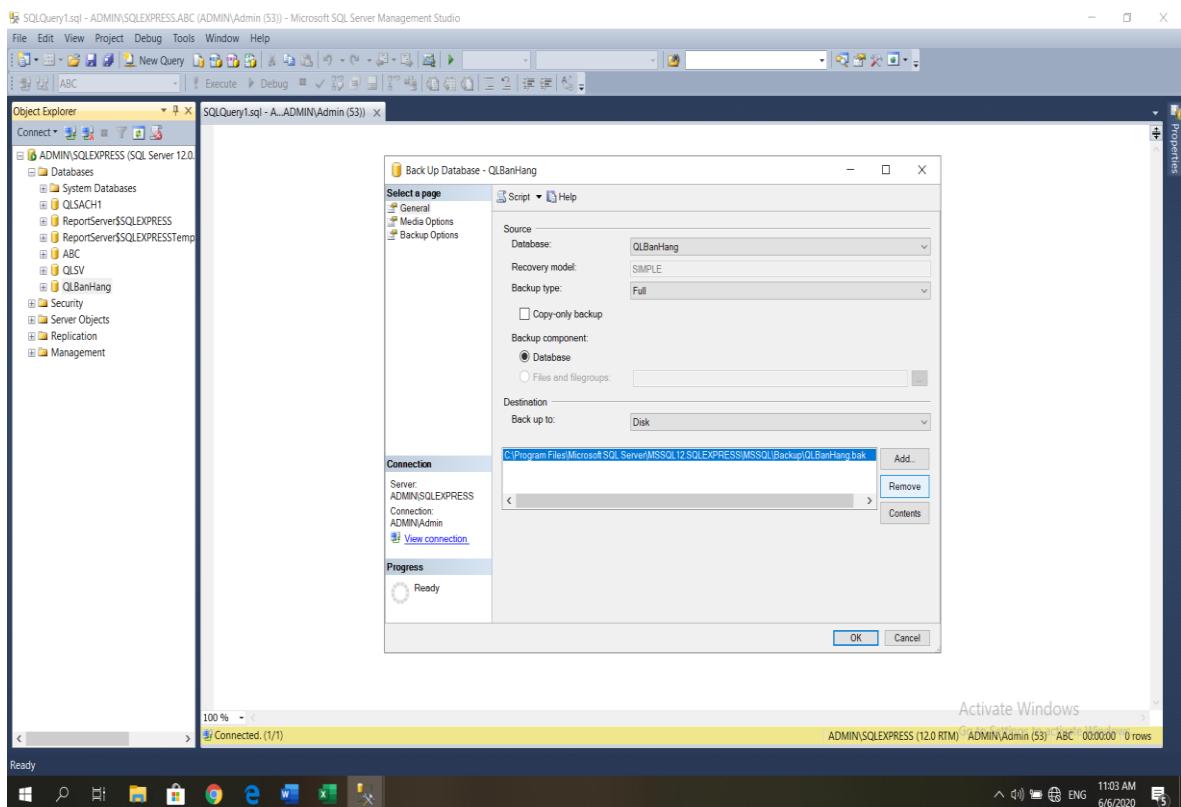
A. Backup dữ liệu: Là quá trình sao lưu dữ liệu

- Chọn CSDL cần sao lưu (QLBanHang) -> Click chuột phải -> Tasks -> Backup



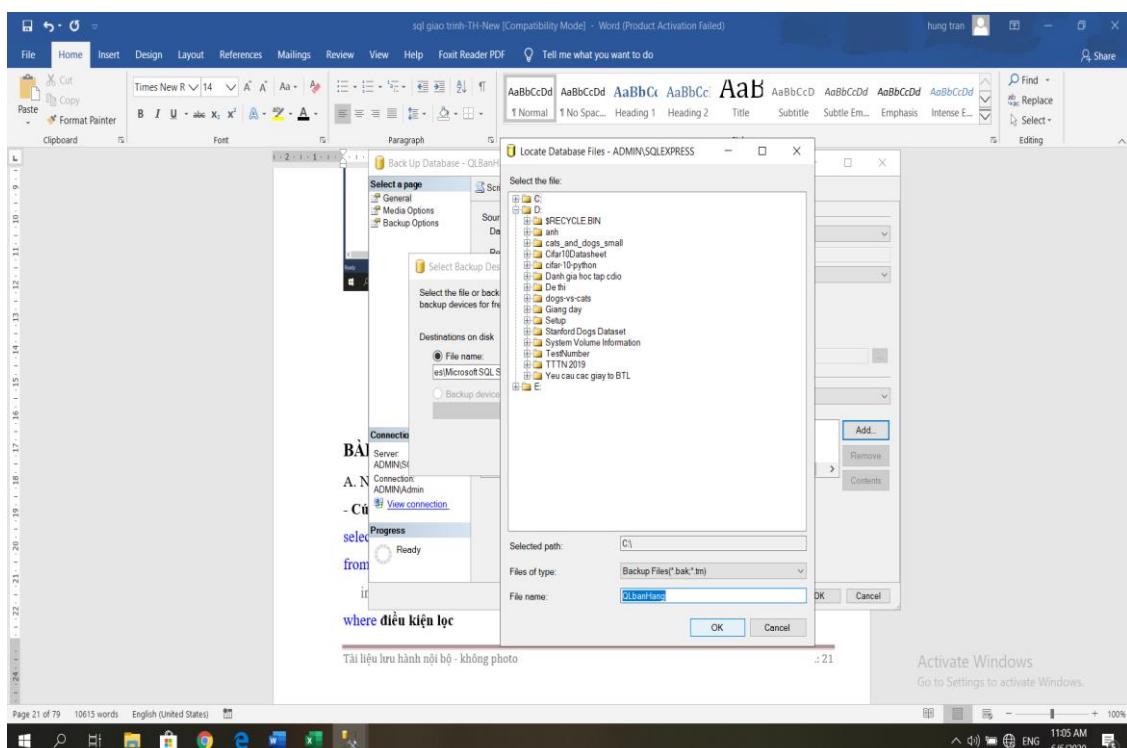
Hình 1.9. Back up dữ liệu

- Back up to: Chọn đường dẫn Lưu file, Nếu muốn lưu nơi khác (Khác với mặc định) -> ta Remove đường dẫn mặc định đi, và chọn nơi lưu và tên file lưu trữ (Add).



Hình 1.10. Add đường dẫn nơi Back up dữ liệu

- Chọn đường dẫn nơi lưu trữ mới và Tên file lưu trữ (.bak)

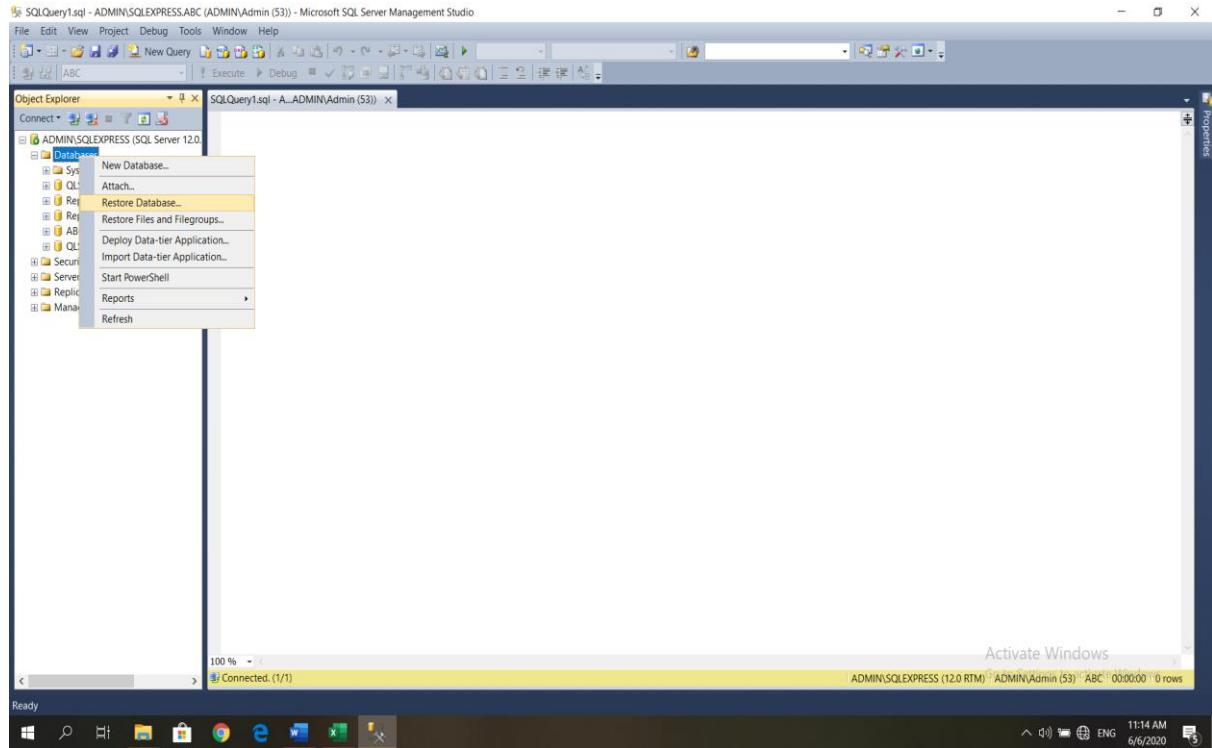


Hình 1.11. Nơi lưu file backup và nhập tên file back up mới

B. Restore dữ liệu: Là quá trình khôi phục dữ liệu

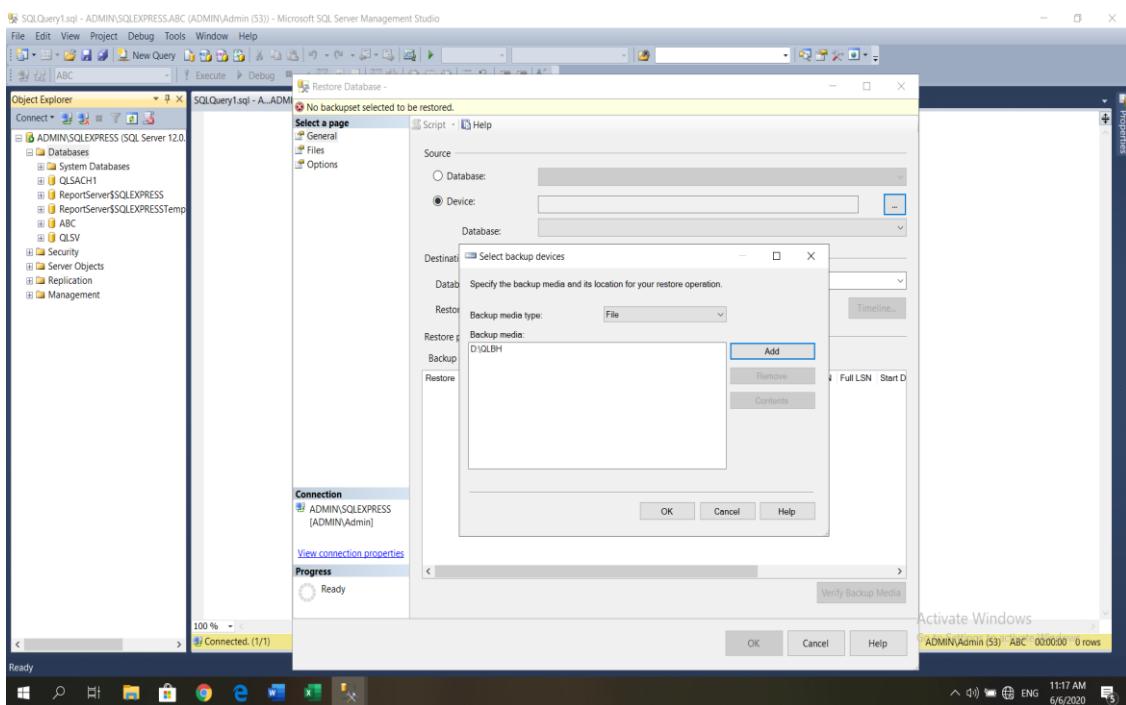
- Phải chắc chắn rằng trong CSDL chưa có CSDL (QLBanHang) mà bạn đang muốn khôi phục trên hệ thống.

- Click chuột phải trên Databases -> Restore Database

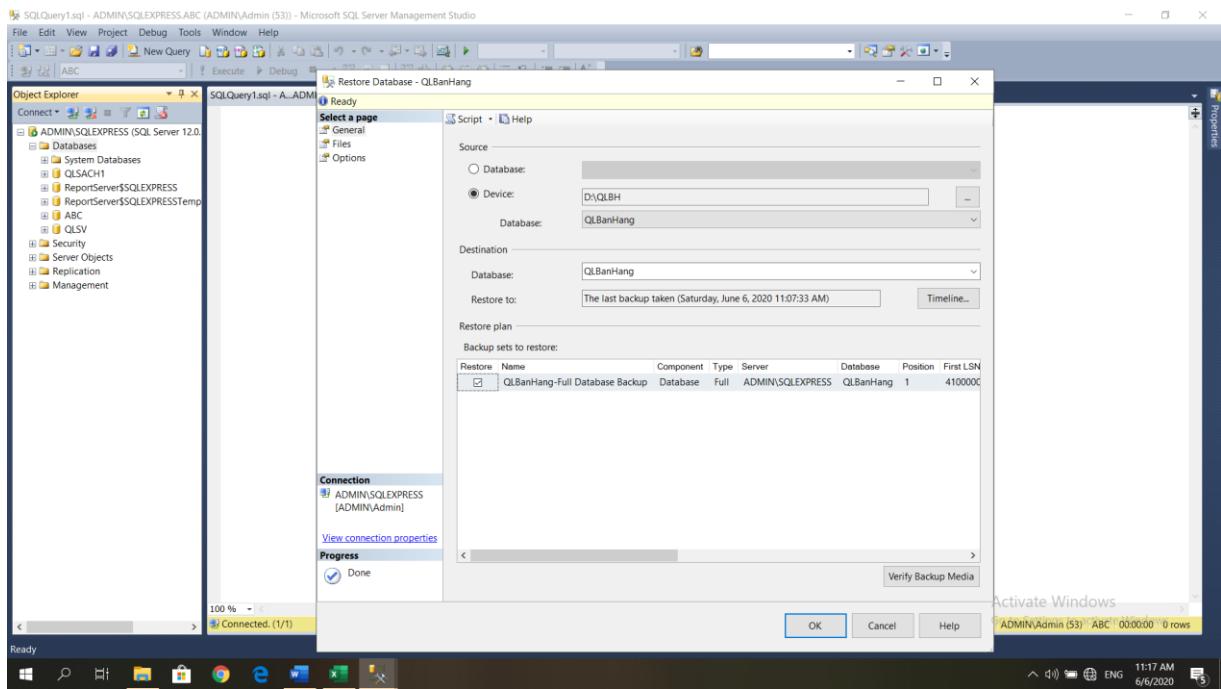


Hình 1.12. Restore dữ liệu

- Chọn Device (Trên thiết bị - ổ đĩa) -> Tìm đường dẫn tới file .bak chúng ta vừa Backup.



Hình 1.13. Chọn đường dẫn file restore



Hình 1.14. Tiến hành Restore CSDL

- Kết quả chúng ta Restore thành công CSDL QLBanHang vừa tạo.

GHI CHÚ:

BÀI THỰC HÀNH SỐ 2

A. Nhắc lại lý thuyết về câu lệnh T-SQL:

- Cú pháp câu lệnh select:

```
select */cột1,2,.../distinct cột/top n/top n percent as 'tên mới'  
from bang1 inner join bang2 on bang1.cot = bang2.cot  
    inner join bang3 on ...  
where điều kiện lọc  
group by cột cần nhóm  
having điều kiện nhóm  
order by cột cần sắp xếp ASC/DESC
```

- Ví dụ: cho 2 bảng dữ liệu

```
sv(masv,tensv,que,gt,tuoi,malop)  
lop(malop,tenlop,phong)
```

1. đưa ra thông tin sv

```
select * from sv
```

2. đưa ra tensv, tuoi

```
select tensv, tuoi from sv
```

3. đưa ra các sv nữ

```
select tensv from sv where gt=N'nữ'
```

- Chữ N'Nữ' – Ký tự 'N' đầu biểu diễn chuỗi unicode.

4. đưa ra 10 sv tuổi cao nhất

```
select top 10 tensv,tuoi  
from sv  
order by tuoi DESC
```

--> thường áp dụng:

- đưa ra 10 tin mới nhất trong ngày
- đưa ra 10 sản phẩm bán chạy nhất

5. Đưa ra 20% sv nữ thấp tuổi nhất đến cao

```
select top 20 percent tensv  
from sv  
where gt=N'nữ'  
order by tuoi ASC
```

6. đưa ra quê quán của các sinh viên, các thông tin trùng nhau chỉ hiển thị một lần

```
select distinct que
```

```
from sv
```

7. kết nối 2 hay nhiều bảng

```
--đưa ra tensv, que, tenlop, phong của các sv  
select tensv,que,tenlop,phong  
from sv inner join lop on sv.malop=lop.malop
```

8. chú ý: nếu 1 trường trong select thuộc vào 2 hay nhiều nguồn dữ liệu thì phải chỉ rõ nguồn nào:

- Ví dụ: đưa ra tensv,que,malop,tenlop,phong của các sv nữ.
-

```
select tensv,que,lop.malop,tenlop,phong  
from sv inner join lop on sv.malop=lop.malop  
where gt=N'nữ'
```

9. một số hàm ngày tháng hay dùng

getdate() -- ngày/tháng/năm/giờ/phút/giây hiện hành

vd: select GETDATE()

- Lấy ngày: day(getdate())
- Lấy tháng: month(getdate())
- Lấy năm year(getdate())

vd: đưa ra tensv, tuoi của các sv

```
select tensv, YEAR(getdate())-YEAR(ngaysinh)as 'Tuoi'  
from sv
```

---Hay dùng trong việc tính toán:

--quản lý thư viện trong việc tính số ngay mượn sách

--quản lý Khách Sạn tính số ngày mượn phòng

10. hàm like '%s%'

-- vd: đưa ra các bạn nữ tên Hằng

```
select tensv from sv  
where gt=N'Nữ' and tensv like N'%Hằng'
```

-- vd: đưa ra các bạn nữ họ Nguyễn

```
select tensv from sv  
where gt=N'Nữ' and tensv like N'Nguyễn%'
```

-- vd: đưa ra các bạn nữ đệm 'thị'

```
select tensv from sv  
where gt=N'Nữ' and tensv like N'%Thị%'
```

----Câu lệnh này thường hay dùng trong ô search – Tìm kiếm sản phẩm,...

11. hàm case .. when .. then

vd:

```
select masv, tensv, tuoi, case
    when gt=N'nam' then N'Anh trai mưa'
    when gt=N'nữ' then N'Em gái mưa'
    end as N'Biệt danh'
    ,que
from sv
```

---Hay sử dụng trong việc chuyển từ số -> chữ:

vd: lưu 0-Nam, 1-Nữ

hoặc: 1-4:F, 4-6: C, 6-8: D,.....

12. hàm thống kê:

- sum(cột) – hàm tính tổng,
- count(*/mã) – hàm đếm,
- avg(cột) – Hàm tính trung bình,
- min(cột) – Hàm tìm giá trị nhỏ nhất,
- max(cột) – Hàm tìm giá trị lớn nhất.
- ví dụ: đưa ra tổng các sv trong lớp

```
select COUNT(*) as N'tong sv'
from sv
```

- đưa ra tổng các sv nữ

```
select COUNT(*) as N'tong'
from sv
where gt=N'nữ'
```

--> where thực hiện đầu tiên --> bảng tạm toàn nữ --> count(*)

13. Hàm liệt kê IN/Not IN – Thay cho các câu lệnh OR nhiều lần.

- đưa ra danh sách các bạn sinh viên quê HN, HY, HP, BG

```
Select * from sv
Where que IN(N'Hà Nội', N'Hưng yên', N'Hải phòng', N'Bắc giang')
```

- đưa ra danh sách các bạn sinh viên không phải quê HN, HY, HP, BG

```
Select * from sv
Where que Not IN(N'Hà Nội', N'Hưng yên', N'Hải phòng', N'Bắc giang')
```

14) Hàm between ... and ...

- Hàm này thường dùng thay thế hàm AND so sánh trong khoảng giá trị.

- đưa ra các bạn sinh viên Nữ có tuổi từ 18 đến 20

```
Select masv, tensv  
From sv  
Where gt=N'Nữ' and year(getdate())-year(ngaysinh) between 18 and  
20
```

B. Từ CSDL QLBanHang đã được tạo ra ở bài thực hành số 1:

Sanpham(masp, mahangsx, tensp, soluong, mausac, giaban, donvitinh, mota)

Hangsx(mahangsx, tenhang, diachi, sodt, email)

Nhanvien(manv, tenv, gioitinh, diachi, sodt, email, phong)

Nhap(sohdn, masp, manv, ngaynhap, soluongN, dongiaN)

Xuat(sohdx, masp, manv, ngayxuat, soluongX)

Hãy thực hiện các câu lệnh T-SQL sau:

1. Hiển thị thông tin các bảng dữ liệu trên.

```
select * from sanpham  
select * from hangsx  
select * from nhanvien  
select * from Nhap  
select * from xuat
```

2. Đưa ra thông tin masp, tensp, tenhang,soluong, mausac, giaban, donvitinh, mota của các sản phẩm sắp xếp theo chiều giảm dần giá bán.

```
select masp, tensp, tenhang, soluong, mausac, giaban, donvitinh,  
mota  
from sanpham inner join hangsx on sanpham.mahangsx =  
hangsx.mahangsx  
order by giaban DESC
```

3. Đưa ra thông tin các sản phẩm có trong cửa hàng do công ty có tên hãng là samsung sản xuất.

```
select masp, tensp, soluong, mausac, giaban, donvitinh, mota  
from sanpham inner join hangsx on sanpham.mahangsx=hangsx.mahangsx  
where tenhang = 'SAMSUNG'
```

4. Đưa ra thông tin các nhân viên Nữ ở phòng ‘Kế toán’.

```
select * from nhanvien  
where gioitinh = N'Nữ' and phong = N'Kế toán'
```

5. Đưa ra thông tin phiếu nhập gồm: sohdn, masp, tensp, tenhang, soluongN, dongiaN, tiennhap=soluongN*dongiaN, mausac, donvitinh, ngaynhap, tenv, phong. Sắp xếp theo chiều tăng dần của hóa đơn nhập.

```
select sohdn, masp, tensp, tenhang, soluongN, dongiaN,
soluongN*dongiaN as N'Tiền nhập', mausac, donvitinh, ngaynhap,
tennv, phong
from nhap inner join sanpham on nhap.masp = sanpham.masp
inner join nhanvien on nhap.manv = nhanvien.manv
order by sohdn ASC
```

6. Đưa ra thông tin phiếu xuất gồm: sohdx, masp, tensp, tenhang, soluongX, giaban, tienxuat=soluongX*giaban, mausac, donvitinh, ngayxuat, tennv, phong trong tháng 10 năm 2018, sắp xếp theo chiều tăng dần của sohdx.

```
select sohdx, masp, tensp, tenhang, soluongX, giaban,
soluongX*giaban as N'Tiền xuất', mausac, donvitinh, ngayxuat, tennv,
phong
from xuat inner join sanpham on xuat.masp = sanpham.masp
inner join nhanvien on xuat.manv = nhanvien.manv
where month(ngayxuat)=10 and year(ngayxuat)=2018
order by sohdx ASC
```

7. Đưa ra các thông tin về các hóa đơn mà hãng samsung đã nhập trong năm 2017, gồm: sohdn, masp, tensp, soluongN, dongiaN, ngaynhap, tennv, phong.

```
select sohdn, masp, tensp, soluongN, dongiaN, ngaynhap, tennv,
phong
from nhap inner join sanpham on nhap.masp = sanpham.masp
inner join nhanvien on nhap.manv = nhanvien.manv
inner join hang on sanpham.mahangsx = hang.mahangsx
where tenhang = 'SAMSUNG' and year(ngaynhap)= 2017
```

8. Đưa ra Top 10 hóa đơn xuất có số lượng xuất nhiều nhất trong năm 2018, sắp xếp theo chiều giảm dần của soluongX.

```
select Top 10 sohdx, ngayxuat, soluongX
from xuat
where year(ngayxuat)=2018
order by soluongX DESC
```

9. Đưa ra thông tin 10 sản phẩm có giá bán cao nhất trong cửa hàng, theo chiều giảm dần giá bán.

```
select top 10 masp, tensp, giaban
from sanpham
order by giaban DESC
```

10. Đưa ra các thông tin sản phẩm có giá bán từ 100.000 đến 500.000 của hãng samsung.

```
select masp, tensp, soluong, mausac, giaban, donvitinh, mota
```

```
from sanpham inner join hangsx on sanpham.mahangsx =
hangsx.mahangsx
where tenhang = 'SAMSUNG' and giaban between 100.000 and 500.000
```

11. Tính tổng tiền đã nhập trong năm 2018 của hãng Samsung.

```
select sum(soluongN*dongiaN) as N'Tổng tiền nhập'
from nhap inner join sanpham on nhap.maspx = sanpham.maspx
inner join hangsx on sanpham.mahangsx =
sanpham.mahangsx
where year(ngaynhap)=2018 and tenhang = 'Samsung'
```

12. Thống kê tổng tiền đã xuất trong ngày 2/9/2018.

```
select sum(soluongX*dongia) as N'Tổng tiền xuất'
from nhap inner join sanpham on nhap.maspx = sanpham.maspx
where ngayxuat = '2/9/2018'
```

13. Đưa ra sohdn, ngaynhap có tiền nhập phải trả cao nhất trong năm 2018.

```
select sohdn,ngaynhap
from nhap
where year(ngaynhap)=2018
and soluongN*dongiaN = (select max(soluongN*dongiaN)
from nhap
where year(ngaynhap)=2018
)
```

14. Đưa ra 10 mặt hàng có soluongN nhiều nhất trong năm 2019.

15. Đưa ra masp,tensp của các sản phẩm do công ty 'Samsung' sản xuất do nhân viên có mã 'NV01' nhập.

16. Đưa ra sohdn,masp,soluongN,ngayN của mặt hàng có masp là 'SP02', được nhân viên 'NV02' xuất.

17. Đưa ra manv,tennv đã xuất mặt hàng có mã 'SP02' ngày '03-02-2020'.

GHI CHÚ:

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

BÀI THỰC HÀNH SỐ 3

A. Nhắc lại lý thuyết về câu lệnh T-SQL:

- Cú pháp câu lệnh select:

```
select */cột1,2,.../distinct cột/top n/top n percent as 'tên mới'  
from bang1 inner join bang2 on bang1.cot = bang2.cot  
    inner join bang3 on ...  
where điều kiện lọc  
group by cột cần nhóm  
having điều kiện nhóm  
order by cột cần sắp xếp ASC/DESC
```

- Ví dụ: cho 2 bảng dữ liệu

```
sv(masv, tensv, que, gt, tuoi, malop)  
lop(malop, tenlop, phong)
```

13) hàm group by ... having

-- đưa ra thống kê mỗi quê có bao nhiêu sv

```
select que,COUNT(*) as N'tong sv'  
from sv  
group by que
```

--> group by trên bảng gốc sv thành các bảng con chứa các sv mỗi quê --> sau đó mới thực hiện count(*)).

-- Thống kê xem mỗi quê có bao nhiêu sv nữ:

```
select que,COUNT(*) as N'tong nu'  
from sv  
where gt=N'nữ'  
group by que
```

--> chú ý: where thực hiện trước --> group by

--> lọc nữ trên bảng gốc sv --> thành 1 bảng có toàn nữ trong bảng tạm --> sau đó mới group nữ theo quê thành các bảng con --> sau đó mới count(*)).

-- Đưa ra các quê có nhiều hơn 10 nữ để thành lập hội đá banh

```
select que,COUNT(*) as N'tong nu'  
from sv  
where gt=N'nữ'  
group by que  
having COUNT(*)>=10  
--> where đầu tiên --> group by --> having --> sum/count/max/min
```

-- lọc bảng sv --> thành bảng tạm toàn nữ --> group by bảng tạm này theo quê thành các bảng con chứa nữ của mỗi quê --> sau đó count(*) trên mỗi bảng con --> having lọc ra các quê thỏa mãn yêu cầu.

- Nếu:

```
select que,COUNT(*) as N'tong nu'  
from sv  
where gt=N'nữ' and COUNT(*)>=10  
group by que
```

--> SAI: các hàm thống kê không sử dụng được trong where.

- Nếu:

```
select que,COUNT(*) as N'tong nu'  
from sv  
group by que  
having gt=N'nữ' and COUNT(*)>=10
```

--> không báo lỗi, nhưng sai kết quả.

-- group by theo quê của bảng sv --> count(*) trên các bảng con chưa lọc nữ--> kết quả sai.

B. Từ CSDL QLBanHang đã được tạo ra ở bài thực hành số 1:

Sanpham(masp, mahangsx, tensp, soluong, mausac, giaban, donvitinh, mota)

Hangsx(mahangsx, tenhang, diachi, sodt, email)

Nhanvien(manv, tennv, gioitinh, diachi, sodt, email, phong)

Nhap(sohdn, masp, manv, ngaynhap, soluongN, dongiaN)

Xuat(sohdx, masp, manv, ngayxuat, soluongX)

Hãy thực hiện các câu lệnh T-SQL sau:

1. Hãy thống kê xem mỗi hàng sản xuất có bao nhiêu loại sản phẩm

```
select hangsx.mahangsx, tenhang, count(*) as N'Số lượng sp'  
from sanpham inner join hangsx on sanpham.mahangsx =  
hangsx.mahangsx  
group by hangsx.mahangsx, tenhang
```

2. Hãy thống kê xem tổng tiền nhập của mỗi sản phẩm trong năm 2018.

```
select nhap.masp,tensp, sum(soluongN*dongiaN) as N'Tổng tiền  
nhập'  
from nhap inner join sanpham on nhap.masp = sanpham.masp  
where year(ngaynhap)=2018  
group by nhap.masp,tensp
```

3. Hãy thống kê các sản phẩm có tổng số lượng xuất năm 2018 là lớn hơn 10.000 sản phẩm của hãng Samsung.

```
select sanpham.masp,tensp,sum(soluongX) as N'Tổng xuất'
from xuat inner join sanpham on xuat.masp = sanpham.masp
inner join hangsx on hangsx.mahangsx =
sanpham.mahangsx
where year(ngayxuat)=2018 and tenhang = 'SAMSUNG'
group by sanpham.masp,tensp
having sum(soluongX) >=10.000
```

4. Thống kê số lượng nhân viên Nam của mỗi phòng ban.

5. Thống kê tổng số lượng nhập của mỗi hãng sản xuất trong năm 2018.

6. Hãy thống kê xem tổng lượng tiền xuất của mỗi nhân viên trong năm 2018 là bao nhiêu.

7. Hãy đưa ra tổng tiền nhập của mỗi nhân viên trong tháng 8 – năm 2018 có tổng giá trị lớn hơn 100.000

8. Hãy đưa ra danh sách các sản phẩm đã nhập nhưng chưa xuất bao giờ.

```
select sanpham.masp,tensp
from sanpham inner join nhap on sanpham.masp = nhap.masp
where sanpham.masp not in (select masp from xuat)
```

9. Hãy đưa ra danh sách các sản phẩm đã nhập năm 2018 và đã xuất năm 2018.

10. Hãy đưa ra danh sách các nhân viên vừa nhập vừa xuất.

11. Hãy đưa ra danh sách các nhân viên không tham gia việc nhập và xuất.

GHI CHÚ:

BÀI THỰC HÀNH SỐ 4

A. Nhắc lại lý thuyết về View:

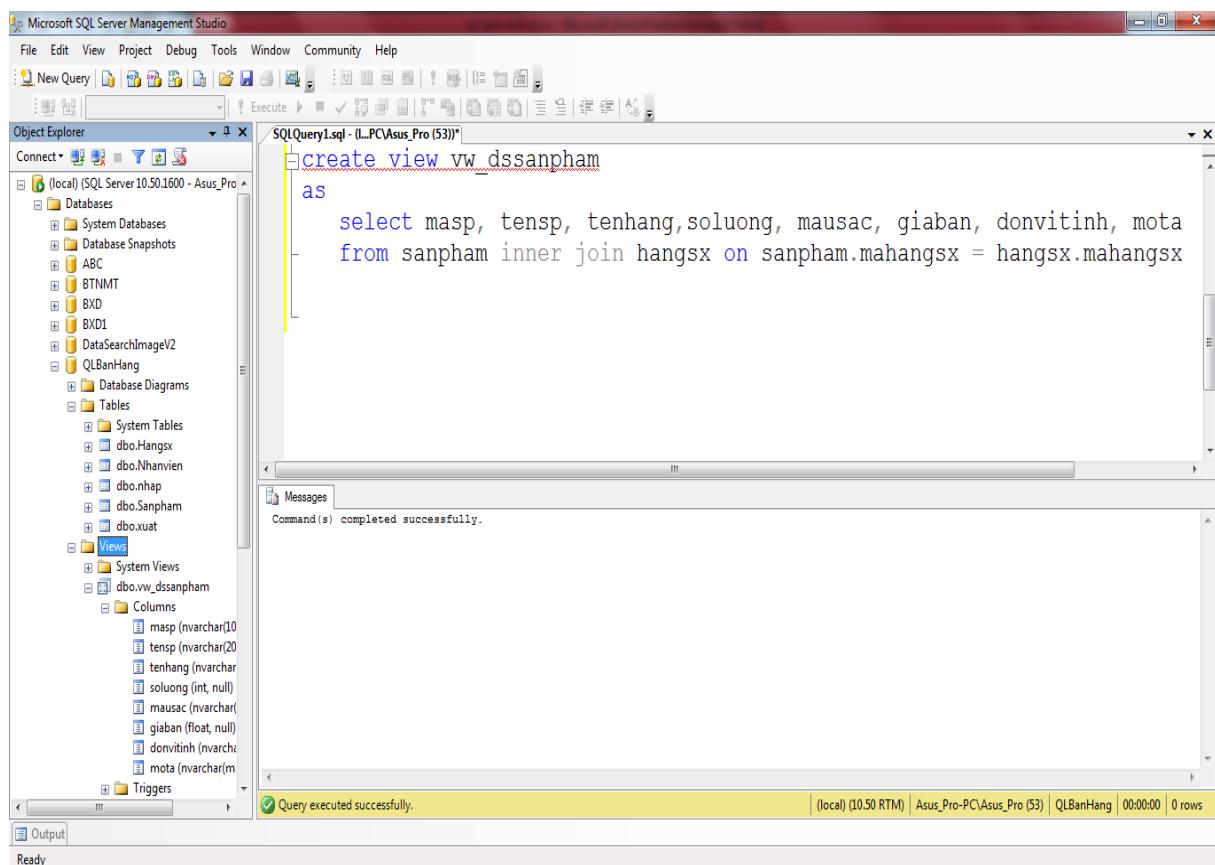
- Khi tạo câu lệnh SQL – khi tắt chương trình câu lệnh sẽ mất đi, không được lưu lại trong CSDL vừa tạo, cho nên ta khó kiểm soát các câu lệnh và sử dụng lại nó cho các việc khác, cho nên phải có 1 cấu trúc lưu lại các câu lệnh sql – đó là các view (bảng ảo – hay khung nhìn).
- Cú pháp view

```
create view tenview  
as  
câu-lệnh-SQL
```

- Khi thực thi view ta chạy cả đoạn lệnh trên
- Khi gọi view cú pháp như sau:

```
select * from tenview
```

- Nơi lưu trữ view trong SQL Server:

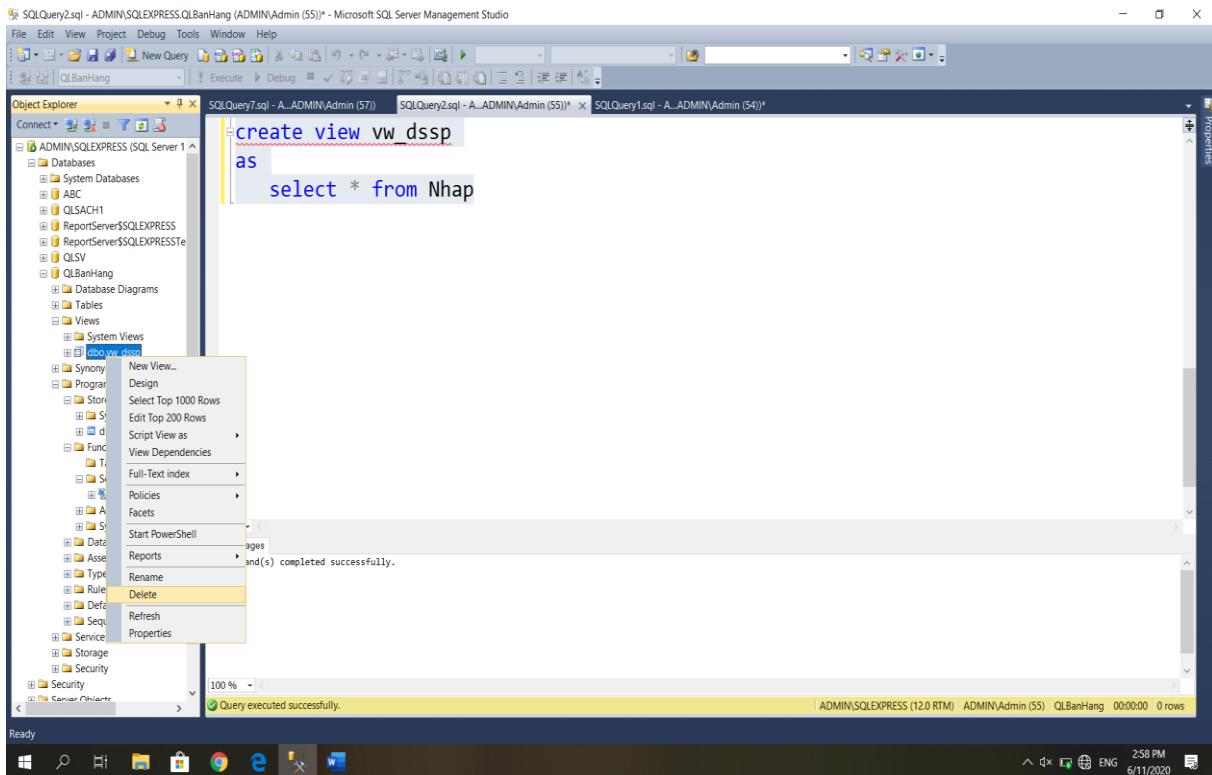


Hình 4.1. Nơi lưu trữ Views trong CSDL

- Khi xóa view

```
drop view tenview
```

✓ Hoặc xóa wizard:



Hình 4.2. Xóa view wizard

- Khi thay đổi nội dung view

```
alter view tenview
```

```
as
```

Câu-lệnh-SQL-thay-đổi

Chú ý:

- ✓ View có thể dùng làm bảng tạm sử dụng trong các câu truy vấn khác.
- ✓ View thường hay được sử dụng trong tạo các reports
- ✓ Trong view các cột đều cần phải có tên, cho nên nếu có thêm cột mới trong quá trình tính toán, ta phải đặt tên cho cột thông qua lệnh as.
- ✓ View không sử dụng cú pháp order by trong xây dựng view. Ta có thể sử dụng order by khi gọi view.

- Ở đây chúng ta có thể xóa, sửa view trên EM.

B. Từ CSDL QLBanHang đã được tạo ra ở bài thực hành số 1:

Sanpham(masp, mahangsx, tensp, soluong, mausac, giaban, donvitinh, mota)

Hangsx(mahangsx, tenhang, diachi, sodt, email)

Nhanvien(manv, tenv, gioitinh, diachi, sodt, email, phong)

Nhap(sohdn, masp, manv, ngaynhap, soluongN, dongiaN)

Xuat(sohdx, masp, manv, ngayxuat, soluongX)

Hãy thực hiện các câu lệnh Tạo view sau:

- Hãy tạo view từ các câu lệnh T-SQL ở bài thực hành số 2 và 3. Sau đó gọi thực thi các view và kiểm tra kết quả chạy.

GHI CHÚ:

BÀI THỰC HÀNH SỐ 5

A. Nhắc lại lý thuyết về Hàm (Function):

- FUNCTION: là hàm có giá trị trả về - tức là hàm có return.

- Có 2 loại Function:

- ✓ Hàm trả về giá trị kiểu vô hướng (Scalar valued Function)
- ✓ Hàm trả về giá trị kiểu bảng (Table valued Function)

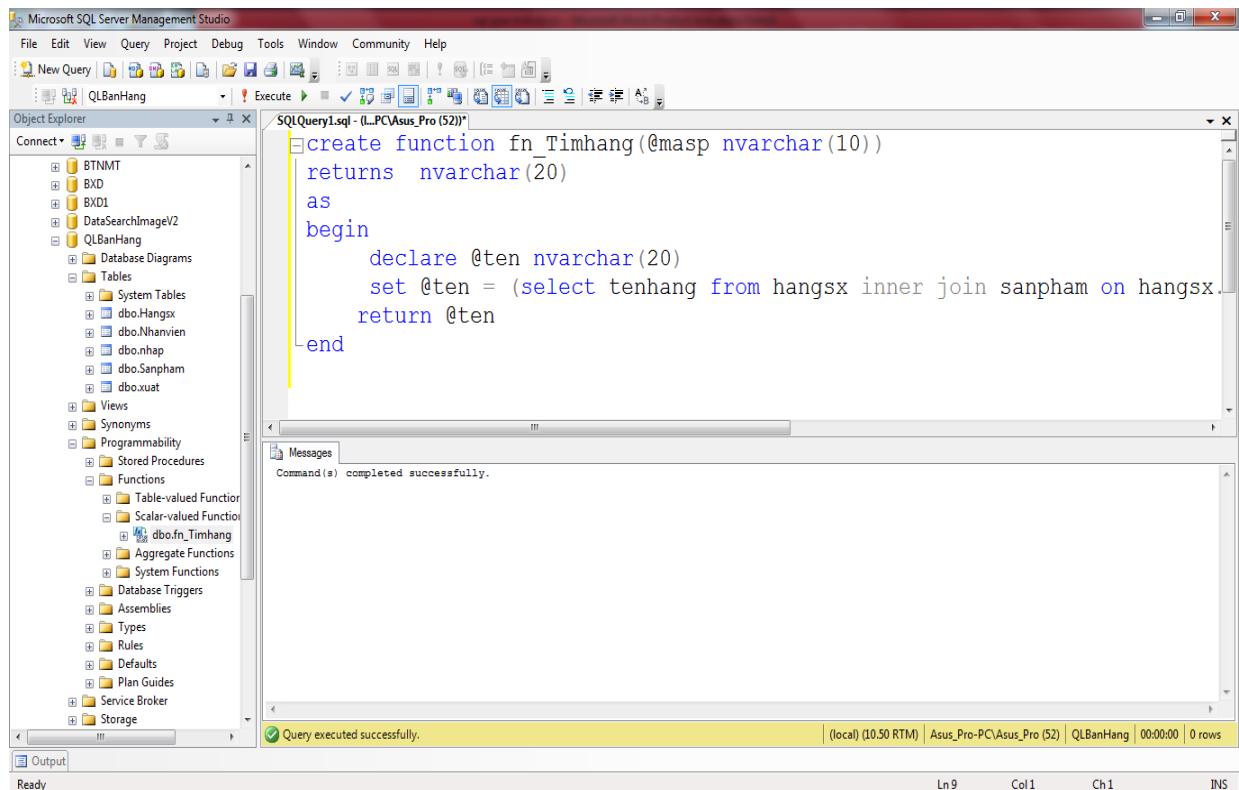
1. Scalar valued Function:

- Cú pháp hàm:

```
create function Tenham(@Thambien1 Kieuudl1,,,)  
returns kieudltrave  
as  
begin --bắt đầu khối lệnh  
    declare @bien kieudltrave      --Khai báo biến cục bộ  
    -- Xử lý thay đổi trên hàm  
    return @bien  
end --kết thúc khối lệnh
```

- Gọi hàm: select dbo.Tenham(doiso1,doiso2, , ,)

- Nơi lưu hàm:

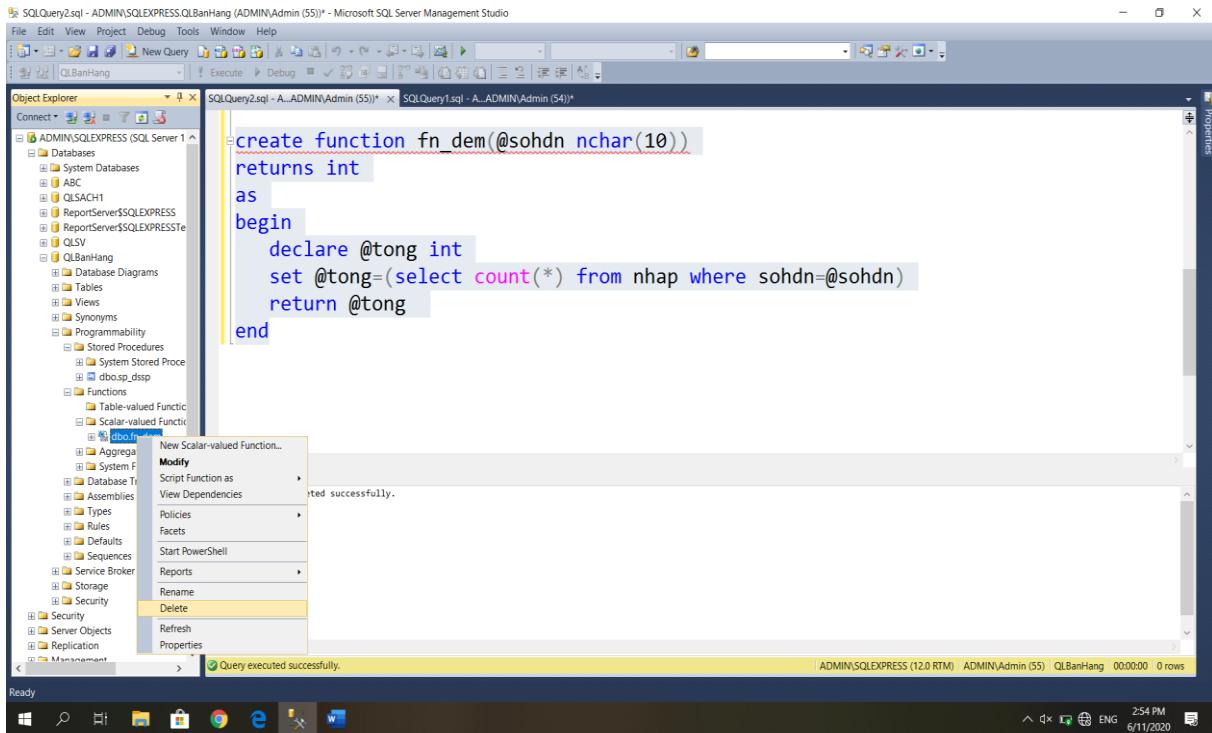


Hình 5.1. Nơi lưu trữ hàm trong CSDL

- Xóa hàm:

Drop function tenham

✓ Hoặc: click chuột phải trên tên hàm -> Delete.

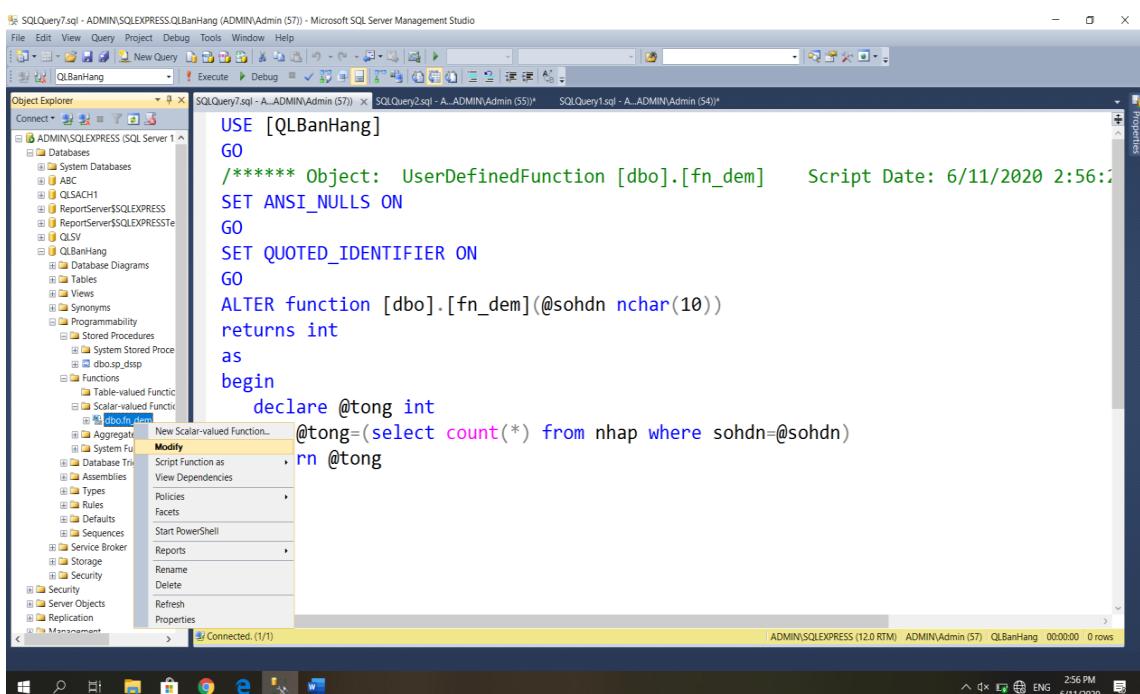


Hình 5.2. Xóa hàm wizard

- Sửa lại nội dung hàm:

Alter function Tenham... -- Các thay đổi nội dung hàm

✓ Hoặc:



Hình 5.3. Sửa lại nội dung hàm

B. Từ CSDL QLBanHang đã được tạo ra ở bài thực hành số 1:

Sanpham(masp, mahangsx, tensp, soluong, mausac, giaban, donvitinh, mota)

Hangsx(mahangsx, tenhang, diachi, sodt, email)

Nhanvien(manv, tenvn, gioitinh, diachi, sodt, email, phong)

Nhap(sohdn, masp, manv, ngaynhap, soluongN, dongiaN)

Xuat(sohdx, masp, manv, ngayxuat, soluongX)

Hãy tạo các hàm sau:

1. Hãy xây dựng hàm đưa ra tên hangsx khi nhập vào masp từ bàn phím

```
create function fn_Timhang(@masp nvarchar(10))
returns nvarchar(20)
as
begin
    declare @ten nvarchar(20)
    set @ten = (select tenhang from hangsx inner join sanpham
                on hangsx.mahangsx = sanpham.mahangsx
                where masp = @masp)
    return @ten
end
```

■ Gọi hàm:

```
select dbo.fn_Timhang('sp01')
```

2. Hãy xây dựng hàm đưa ra tổng giá trị nhập từ năm nhập x đến năm nhập y, với x, y được nhập vào từ bàn phím.

```
create function fn_thongkenhaptheonam(@x int,@y int)
returns int
as
begin
    declare @tongtien int
    select @tongtien = sum(soluongN*dongiaN)
    from nhap
    where year(ngaynhap) between @x and @y
    return @tongtien
end
```

■ Gọi hàm:

```
select dbo.fn_thongkenhaptheonam(2016,2018)
```

3. Hãy viết hàm thống kê tổng số lượng thay đổi nhập xuất của tên sản phẩm x trong năm y, với x,y nhập từ bàn phím.

```
create function fn_thongkenhapxuat(@tensp nvarchar(20),@nam int)
```

```

returns int
as
begin
    declare @tongnhap int
    declare @tongxuat int
    declare @thaydoi int
    select @tongnhap = sum(soluongN) from nhap inner join sanpham
    on nhap.masp = sanpham.masp
    where tensp = @tensp and year(ngaynhap)=@nam
    select @tongxuat = sum(soluongX) from xuat inner join sanpham
    on xuat.masp = sanpham.masp
    where tensp = @tensp and year(ngayxuat)=@nam
    set @thaydoi = @tongnhap - @tongxuat
    return @thaydoi
end

```

■ Gọi hàm:

```
select dbo.fn_thongkenhapxuat('sp1',2018)
```

4. Hãy xây dựng hàm đưa ra tổng giá trị nhập từ ngày nhập x đến ngày nhập y, với x, y được nhập vào từ bàn phím.
5. Hãy xây dựng hàm đưa ra tổng giá trị xuất của hãng tên hãng là A, trong năm tài khóa x, với A ,x được nhập từ bàn phím.
6. Hãy xây dựng hàm thống kê số lượng nhân viên mỗi phòng với tên phòng nhập từ bàn phím.
7. Hãy viết hàm thống kê xem tên sản phẩm x đã xuất được bao nhiêu sản phẩm trong ngày y, với x,y nhập từ bàn phím.
8. Hãy viết hàm trả về số điện thoại của nhân viên đã xuất số hóa đơn x, với x nhập từ bàn phím.
9. Hãy viết hàm thống kê tổng số lượng thay đổi nhập xuất của tên sản phẩm x trong năm y, với x,y nhập từ bàn phím.
10. Hãy viết hàm thống kê tổng số lượng sản phẩm của hãng x, với tên hãng nhập từ bàn phím.

GHI CHÚ:

BÀI THỰC HÀNH SỐ 6

A. Nhắc lại lý thuyết về Hàm (Function):

Table Valued Function:

- Hàm này trả về 1 bảng dữ liệu với điều kiện lọc là các đối số được truyền vào trong hàm, hàm này khác với view là hàm trả về được dữ liệu tùy biến.

- Cú pháp hàm:

```
create function Tenham(@Thambien1 Kieudl1,,,)  
returns @tenbang table(  
    truong1 kieudl1,  
    truong2 kieudl2,  
    ...  
)  
as  
begin  
    ... xử lý trên hàm  
    ...  
    insert into @tenbang  
        select truong1, truong2, ...  
        from ...  
        where ...  
        ...  
    ...  
    return  
end
```

- Gọi hàm: `select * from Tenham(doiso1, doiso2, , ,)`

- Nơi lưu hàm:

```

create function fn_DSSPtheohangsx (@tenhang nvarchar(20))
returns @bang table(
    masp nvarchar(10), tensp nvarchar(20),
    soluong int, mausac nvarchar(20),
    giaban float, donvitinh nvarchar(10), mota nvarchar(max)
)
as
begin
    insert into @bang
        select masp, tensp, soluong, mausac, giaban, donvitinh, mota
        from sanpham inner join hangsx
            on sanpham.mahangsx = hangsx.mahangsx
        where tenhang = @tenhang
    return
end

```

Messages
Command(s) completed successfully.

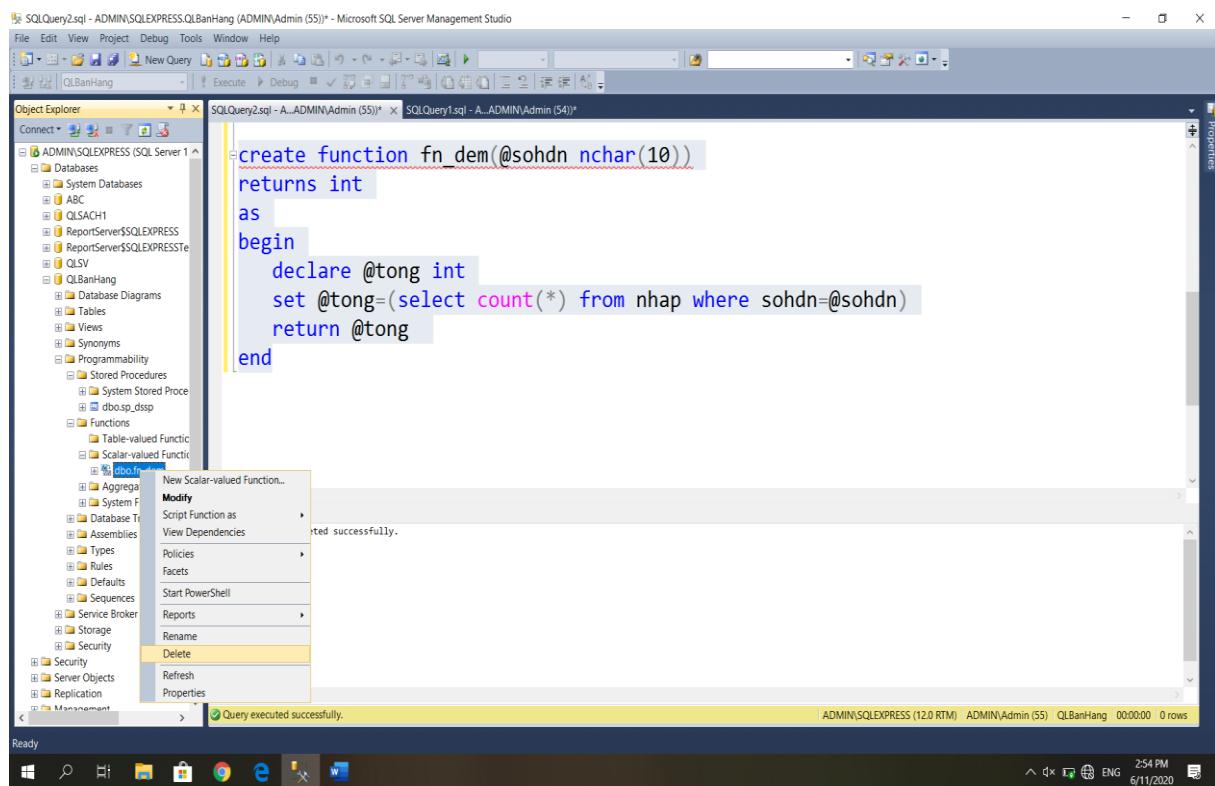
Query executed successfully. | (local) (10.50 RTM) | Asus_Pro-PC\Asus_Pro (52) | QLBanHang | 00:00:00 | 0 rows

Hình 6.1. Nơi lưu trữ hàm trong CSDL

- Xóa hàm:

`drop function tenham`

✓ Hoặc: click chuột phải tên hàm -> Delete.

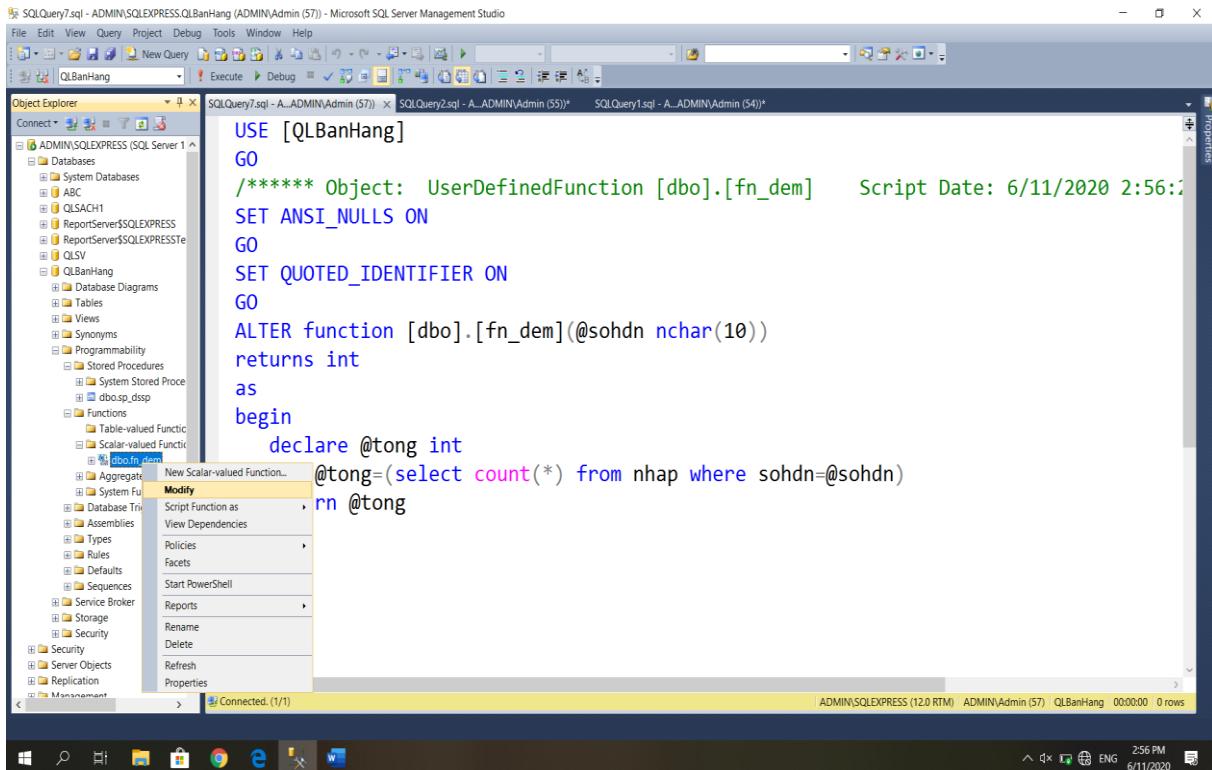


Hình 6.2. Xóa hàm wizard

- Sửa lại nội dung hàm:

alter function Tenham... -- Các thay đổi nội dung hàm

✓ Hoặc:



Hình 6.3. Sửa lại nội dung hàm

B. Từ CSDL QLBanHang đã được tạo ra ở bài thực hành số 1:

Sanpham(masp, mahangsx, tensp, soluong, mausac, giaban, donvitinh, mota)

Hangsx(mahangsx, tenhang, diachi, sotd, email)

Nhanvien(manv, tenv, gioitinh, diachi, sotd, email, phong)

Nhap(sohdn, masp, manv, ngaynhap, soluongN, dongiaN)

Xuat(sohdx, masp, manv, ngayxuat, soluongX)

Hãy tạo các hàm sau:

1. Hãy xây dựng hàm đưa ra thông tin các sản phẩm của hãng có tên nhập từ bàn phím.

```
create function fn_DSSPtheohangsx(@tenhang nvarchar(20))
returns @bang table(
    masp nvarchar(10),
    tensp nvarchar(20),
    soluong int,
    mausac nvarchar(20),
    giaban money,
    donvitinh nvarchar(10),
```

```

        mota nvarchar(max)
    )
as
begin
    insert into @bang
        select masp, tensp, soluong, mausac, giaban,
donvitinh, mota
            from sanpham inner join hangsx
            on sanpham.mahangsx = hangsx.mahangsx
            where tenhang = @tenhang
    return
end

```

■ Gọi hàm:

```
select * from fn_DSSPtheohangsx('samsung')
```

2. Hãy viết hàm đưa ra danh sách các sản phẩm và hãng sản xuất tương ứng đã được nhập từ ngày x đến ngày y, với x,y nhập từ bàn phím.

```

create function fn_DSSPnhaptheongay(@x date,@y date)
returns @bang table(
        masp nvarchar(10),
        tensp nvarchar(20),
        tenhang nvarchar(20),
        ngaynhap date,
        soluongN int,
        dongiaN float
    )
as
begin
    insert into @bang
        select sanpham.masp, tensp, tenhang, ngaynhap,
soluongN, dongiaN
            from nhap inner join sanpham
            on nhap.masp = sanpham.masp inner join hangsx
            on sanpham.mahangsx = hangsx.mahangsx
            where ngaynhap between @x and @y
    return
end

```

■ Gọi hàm:

```
select * from fn_DSSPnhaptheongay ('2/9/2018','3/9/2018')
```

3. Hãy xây dựng hàm đưa ra danh sách các sản phẩm theo hãng sản xuất và 1 lựa chọn, nếu lựa chọn = 0 thì đưa ra danh sách các sản phẩm có soluong = 0, ngược lại lựa chọn =1 thì đưa ra danh sách các sản phẩm có soluong >0.

```
create function fn_DSSPtheoSL(@tenhang nvarchar(20), @Flag int)
returns @bang table(
    masp nvarchar(10),
    tensp nvarchar(20),
    tenhang nvarchar(20),
    soluong int,
    mausac nvarchar(20),
    giaban money,
    donvitinh nvarchar(10),
    mota nvarchar(max)
)
as
begin
    if(@flag=0)
        insert into @bang
            select masp, tensp, tenhang, soluong, mausac,
giaban, donvitinh, mota
            from sanpham inner join hangsx
            on sanpham.mahangsx = hangsx.mahangsx
            where tenhang = @tenhang and soluong=0
    else
        if(@flag =1)
            insert into @bang
                select masp, tensp, tenhang, soluong,
mausac, giaban, donvitinh, mota
                from sanpham inner join hangsx
                on
sanpham.mahangsx = hangsx.mahangsx
                where tenhang = @tenhang and soluong >0
    return
end
```

■ Gọi hàm:

```
select * from fn_DSSPtheoSL('Samsung',0)
select * from fn_DSSPtheoSL('Samsung',1)
```

4. Hãy xây dựng hàm đưa ra danh sách các nhân viên có tên phòng nhập từ bàn phím.
5. Hãy tạo hàm đưa ra danh sách các hãng sản xuất có địa chỉ nhập vào từ bàn phím (Lưu ý – Dùng hàm like để lọc).

6. Hãy viết hàm đưa ra danh sách các sản phẩm và hãng sản xuất tương ứng đã được xuất từ năm x đến năm y, với x,y nhập từ bàn phím.
7. Hãy xây dựng hàm đưa ra danh sách các sản phẩm theo hãng sản xuất và 1 lựa chọn, nếu lựa chọn = 0 thì đưa ra danh sách các sản phẩm đã được nhập, ngược lại lựa chọn =1 thì đưa ra danh sách các sản phẩm đã được xuất.
8. Hãy xây dựng hàm đưa ra danh sách các nhân viên đã nhập hàng vào ngày được đưa vào từ bàn phím.
9. Hãy xây dựng hàm đưa ra danh sách các sản phẩm có giá bán từ x đến y, do công ty z sản xuất, với x,y,z nhập từ bàn phím.
10. Hãy xây dựng hàm không tham biến đưa ra danh sách các sản phẩm và hãng sản xuất tương ứng.

GHI CHÚ:

BÀI THỰC HÀNH SỐ 7

A. Nhắc lại lý thuyết về thủ tục lưu trữ (Stored Procedure):

- Stored procedure: là thủ tục lưu trữ, không trả về giá trị, tương tự hàm void trong C/C++, chỉ thực thi chứ không trả về. Khi gọi thủ tục (stored procedure) không có return.

- Thủ tục có 2 dạng:

- ✓ Thủ tục không có output
- ✓ Thủ tục có output

Thủ tục không có output:

- Cú pháp thủ tục:

```
create proc tenthutuc (@thambien1 kieudl1,@thambien2 kieudl2,...)
as
begin
    --Khai báo các biến cục bộ sử dụng trong thủ tục
    declare @bien1 kieudl1
    ...
    -- Xử lý trong thủ tục
end
```

- Gọi thủ tục: exec tenthuthuc DOISO1,DOISO2,...

- Nơi lưu thủ tục:

```

create proc sp_NhapHangSX (@ma nvarchar(10),
                           @ten nvarchar(20),
                           @dc nvarchar(20),
                           @dt nvarchar(11),
                           @email nvarchar(30))
as
begin
    if(exists(select * from Hangsx where tenhang = @ten))
        print 'Tên hàng đã tồn tại, mời bạn nhập tên khác!!!'
    else
        insert into Hangsx values(@ma,@ten,@dc,@dt,@email)
end

```

Messages

Command(s) completed successfully.

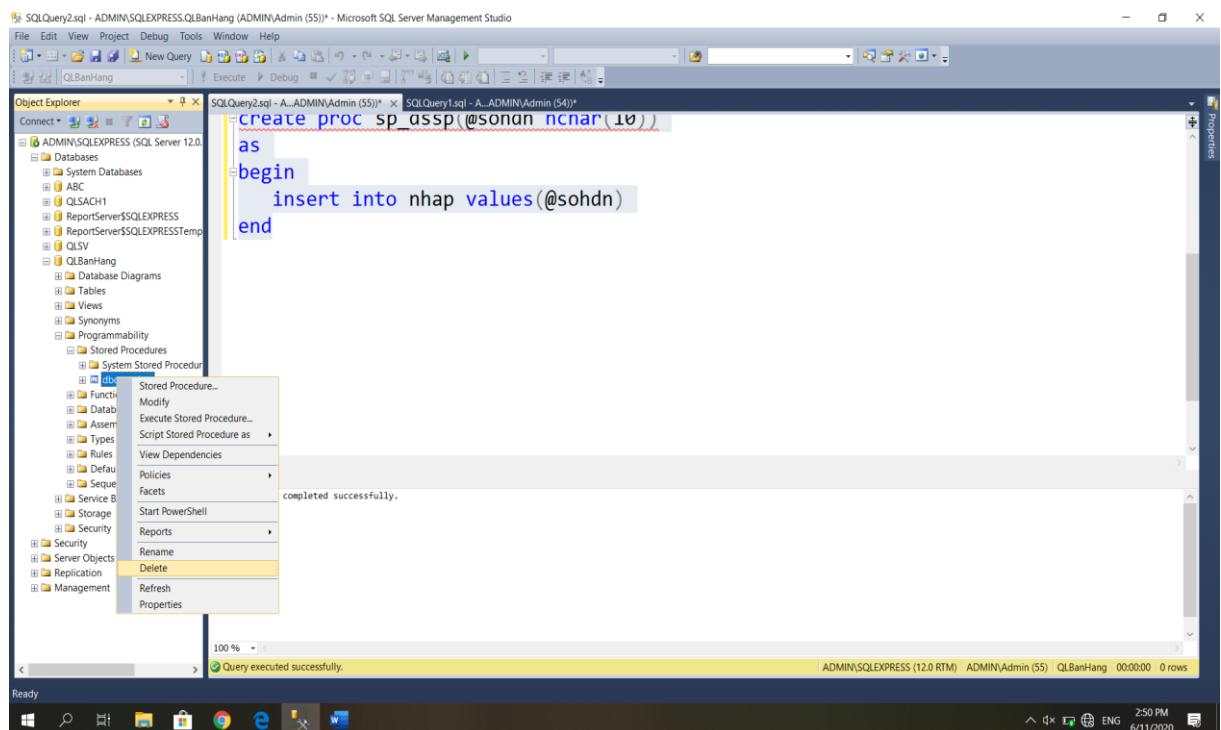
Query executed successfully. | (local) (10.50 RTM) | Asus_Pro-PC\Asus_Pro (55) | QLBanHang | 00:00:00 | 0 rows

Hình 7.1. Nơi Lưu trữ thủ tục trong CSDL

- Xóa thủ tục:

Drop proc tenthutuc

✓ Hoặc: click chuột phải trên tên thủ tục -> Delete.

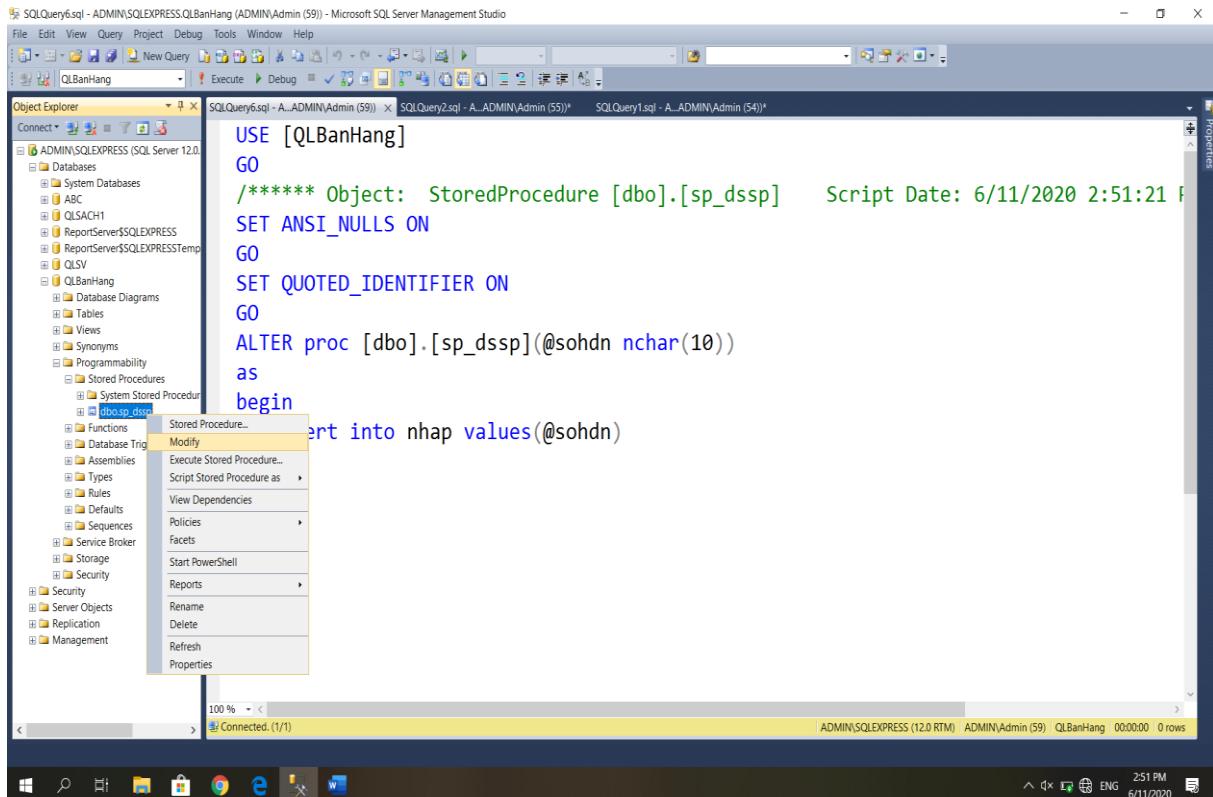


Hình 7.2. Xóa thủ tục wizard

- Sửa lại nội dung thủ tục:

Alter proc tenthetutuc... – Các thay đổi nội dung thủ tục

✓ Hoặc:



Hình 7.3. Sửa lại nội dung thủ tục

B. Từ CSDL QLBanHang đã được tạo ra ở bài thực hành số 1:

Sanpham(masp, mahangsx, tensp, soluong, mausac, giaban, donvitinh, mota)

Hangsx(mahangsx, tenhang, diachi, sodt, email)

Nhanvien(manv, tennv, gioitinh, diachi, sodt, email, phong)

Nhap(sohdn, masp, manv, ngaynhap, soluongN, dongiaN)

Xuat(sohdx, masp, manv, ngayxuat, soluongX)

Hãy tạo các thủ tục lưu trữ sau:

- Tạo thủ tục nhập liệu cho bảng Hangsx, với các tham biến truyền vào mahangsx, tenhang, diachi, sodt, email. Hãy kiểm tra xem tenhang đã tồn tại trước đó hay chưa? Nếu có rồi thì không cho nhập và đưa ra thông báo.

```
create proc sp_NhapHangSX(
    @ma nvarchar(10),
    @ten nvarchar(20),
    @dc nvarchar(20),
    @dt nvarchar(11),
    @email nvarchar(30))
as
```

```

begin
    if(exists(select * from Hangsx where tenhang = @ten))
        print N'Tên hàng đã tồn tại, mời bạn nhập tên khác!!!'
    else
        insert into Hangsx values(@ma,@ten,@dc,@dt,@email)
end

```

■ Gọi thủ tục:

-- Trường hợp đúng

```
Exec sp_NhapHangSX 'ct05', 'Nokia', 'Findland', '76897484',
'nokia@gmail.com.fl'
```

-- Trường hợp bị trùng tên hàng.

```
Exec sp_NhapHangSX 'ct06', 'Nokia', 'Potugal', '12234455',
'ronaldo@gmail.com.pg'
```

2. Tạo thủ tục nhập dữ liệu cho bảng sản phẩm với các tham biến truyền vào masp, tenhangsx, tensp, soluong, mausac, giaban, donvitinh, mota. Hãy kiểm tra xem nếu masp đã tồn tại thì cập nhật thông tin sản phẩm theo mã, ngược lại thêm mới sản phẩm vào bảng sanpham.

```

create proc sp_NhapSP(
    @masp nvarchar(10),
    @tenhang nvarchar(20),
    @tensp nvarchar(20),
    @soluong int,
    @mausac nvarchar(20),
    @giaban money,
    @donvitinh nvarchar(10),
    @mota nvarchar(max))

as
begin
    if(not exists(select * from hangsx where tenhang=@tenhang))
        print N'Không tồn tại tên hàng này'
    else
        begin
            declare @mahangsx nvarchar(10)
            select @mahangsx = mahangsx
            from hangsx where tenhang = @tenhang
            if(exists(select * from sanpham where masp=@masp))
                update sanpham set mahangsx = @mahangsx,
                    tensp = @tensp,
                    soluong = @soluong,
                    mausac = @mausac,

```

```

        giaban = @giaban,
        donvitinh = @donvitinh,
        mota = @mota
    where masp =@masp
else
    insert into sanpham values(@masp, @mahangsx,
@m tensp, @soluong, @mausac, @giaban, @donvitinh, @mota)
end
end

```

■ Gọi thủ tục:

-- Trường hợp đúng: Cập nhật

```
exec sp_NhapSP '01', N'Nokia', 'a11100', 5, N'xanh', 2000000, N'Chiếc', N'Sản phẩm
phổ thông'
```

-- Trường hợp đúng: Thêm mới

```
exec sp_NhapSP '01', N'Nokia', 'a11100', 5, N'xanh', 1950000, N'Chiếc',
N'Sản phẩm phổ thông'
```

-- Trường hợp sai:"Không có sản phẩm Noki"

```
exec sp_NhapSP '02', N'Noki', 'a11100', 5, N'xanh', 2000000,
N'Chiếc', N'Sản phẩm phổ thông'
```

3. Viết thủ tục xóa dữ liệu bảng hangsx với tham biến là tenhang. Nếu tenhang chưa có
thì thông báo, ngược lại xóa hangsx với hàng bị xóa là tenhang. (Lưu ý: xóa hangsx thì
phải xóa các sản phẩm mà hangsx này cung ứng).

```

create proc sp_xoaHangSX(@tenhang nvarchar(20))
as
begin
    if(not exists(select * from hangsx where tenhang =
@tenhang))
        print N'Không tồn tại hàng sx cần xóa'
    else
        begin
            declare @mahangsx nvarchar(10)
            select @mahangsx = mahangsx from hangsx where
                tenhang = @tenhang
            delete from sanpham where mahangsx = @mahangsx
            delete from hangsx where mahangsx = @mahangsx
        end
    end

```

-- Trường hợp đúng:

```
exec sp_xoaHangSX N'Nokia'
```

-- Trường hợp sai:

```
exec sp_xoaHangSX N'Samsung'
```

4. Viết thủ tục nhập dữ liệu cho bảng nhân viên với các tham biến manv, tenv, gioitinh, diachi, sotd, email, phong, và 1 biến cờ Flag, Nếu Flag = 0 thì cập nhật dữ liệu cho bảng nhân viên theo manv, ngược lại thêm mới nhân viên này.
5. Viết thủ tục nhập dữ liệu cho bảng Nhap với các tham biến sohdn, masp, manv, ngaynhap, soluongN, donggiaN. Kiểm tra xem masp có tồn tại trong bảng Sanpham hay không? manv có tồn tại trong bảng nhanvien hay không? Nếu không thì thông báo, ngược lại thì hãy kiểm tra: Nếu sohdn đã tồn tại thì cập nhật bảng Nhap theo sohdn, ngược lại thêm mới bảng Nhap.
6. Viết thủ tục nhập dữ liệu cho bảng xuat với các tham biến sohdx, masp, manv, ngayxuat, soluongX. Kiểm tra xem masp có tồn tại trong bảng Sanpham hay không? manv có tồn tại trong bảng nhanvien hay không? soluongX <= Soluong? Nếu không thì thông báo, ngược lại thì hãy kiểm tra: Nếu sohdx đã tồn tại thì cập nhật bảng Xuat theo sohdx, ngược lại thêm mới bảng Xuat.
7. Viết thủ tục xóa dữ liệu bảng nhanvien với tham biến là manv. Nếu manv chưa có thì thông báo, ngược lại xóa nhanvien với nhanvien bị xóa là manv. (Lưu ý: xóa nhanvien thì phải xóa các bảng Nhap, Xuat mà nhân viên này tham gia).
8. Viết thủ tục xóa dữ liệu bảng sanpham với tham biến là masp. Nếu masp chưa có thì thông báo, ngược lại xóa sanpham với sanpham bị xóa là masp. (Lưu ý: xóa sanpham thì phải xóa các bảng Nhap, Xuat mà sanpham này cung ứng).

GHI CHÚ:

BÀI THỰC HÀNH SỐ 8

A. Nhắc lại lý thuyết về thủ tục lưu trữ (Stored Procedure):

Thủ tục có output:

- Lệnh print chỉ xuất trong SQL Server, không xuất trong chương trình lập trình quản lý kết nối tới SQL server. Cho nên ta cần 1 biến lỗi trả về sau khi gọi thủ tục để chương trình lập trình có thể đọc được biến trả về đó nhằm xuất ra các thông báo (Message).

- Cú pháp thủ tục:

```
create proc tenthutuc (
    @thambien1 kieudl1,
    ...,
    @bientrave Kieudl OUTPUT)
as
begin
    --Khai báo các biến cục bộ sử dụng trong thủ tục
    declare @bien1 kieudl
    ...
    --Xử lý trong thủ tục
    set @bientrave = giatri
    return @bientrave
end
```

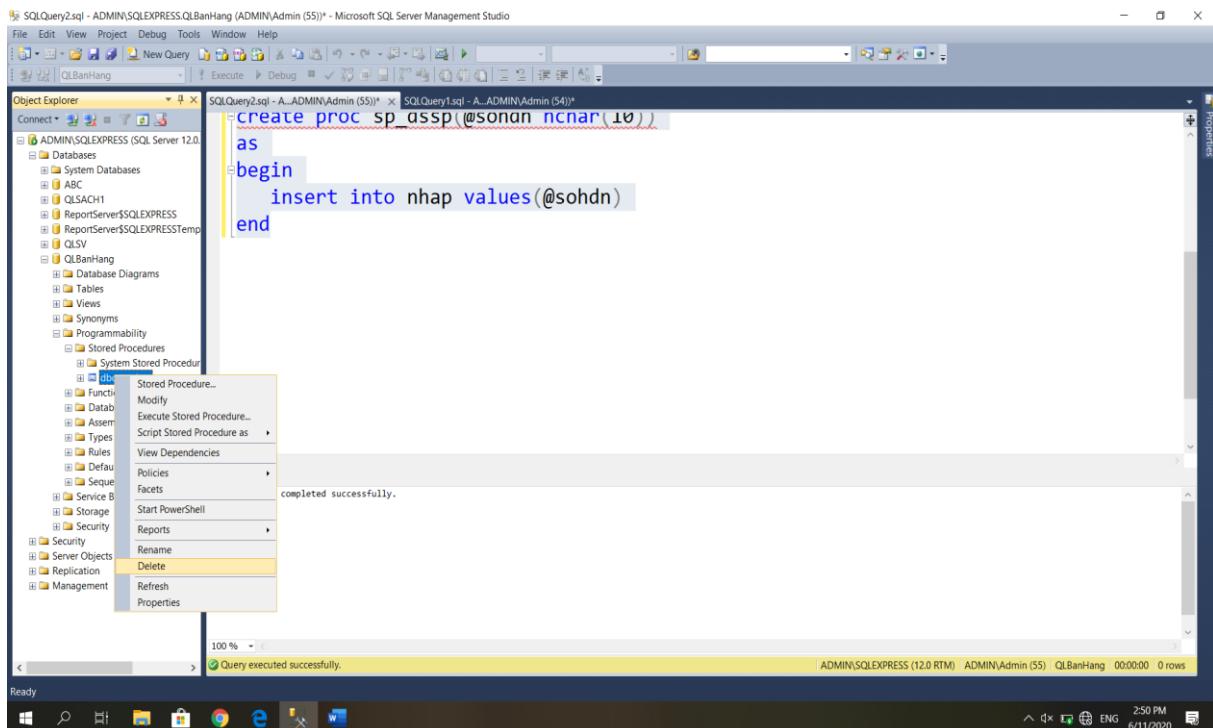
- Gọi thủ tục:

```
declare @bientrave kieudl
exec tenthutuc DOIS01,DOIS02,...,@bientrave OUTPUT
select @bientrave
```

- ✓ Chọn khối lệnh và thực thi.
- Nơi lưu thủ tục: Cùng nơi với thủ tục không có output.
- Xóa thủ tục:

```
drop proc tenthutuc
```

- ✓ Hoặc: click chuột phải trên tên thủ tục -> Delete.

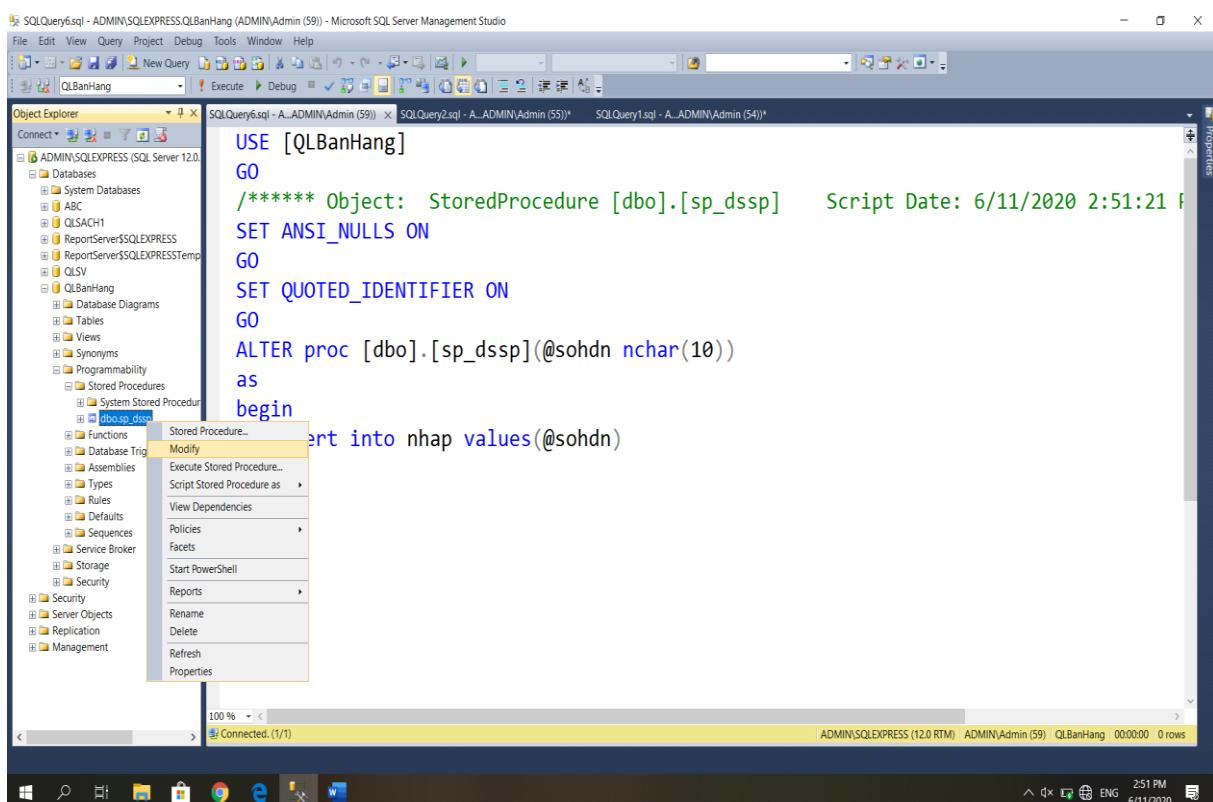


Hình 8.1. Xóa thủ tục wizard

- Sửa lại nội dung thủ tục:

Alter proc Tenthetutuc... -Các thay đổi nội dung thủ tục

✓ Hoặc:



Hình 8.2. Sửa lại nội dung thủ tục

B. Từ CSDL QLBanHang đã được tạo ra ở bài thực hành số 1:

Sanpham(masp, mahangsx, tensp, soluong, mausac, giaban, donvitinh, mota)

Hangsx(mahangsx, tenhang, diachi, sodt, email)

Nhanvien(manv, tenvn, gioitinh, diachi, sodt, email, phong)

Nhap(sohdn, masp, manv, ngaynhap, soluongN, dongiaN)

Xuat(sohdx, masp, manv, ngayxuat, soluongX)

Hãy tạo các thủ tục lưu trữ có output sau:

1. Viết thủ tục thêm mới nhân viên bao gồm các tham số: manv, tenvn, gioitinh, diachi, sodt, email, phong và 1 biến Flag, Nếu Flag=0 thì nhập mới, ngược lại thì cập nhật thông tin nhân viên theo mã. Hãy kiểm tra:

- gioitinh nhập vào có phải là Nam hoặc Nữ không, nếu không trả về mã lỗi 1.

- Ngược lại nếu thỏa mãn thì cho phép nhập và trả về mã lỗi 0.

```
create proc sp_ThemNhanvien(
    @manv nvarchar(10),
    @tenvn nvarchar(20),
    @gioitinh nvarchar(10),
    @diachi nvarchar(30),
    @sodt nvarchar(20),
    @email nvarchar(30),
    @phong nvarchar(20),
    @Flag int,
    @KQ int output
)
as
begin
    if(@gioitinh<>'Nam' and @gioitinh<>'Nữ')
        set @KQ=1
    else
        begin
            set @KQ=0
            if(@Flag =0)
                insert into nhanvien values(@manv,
@tenvn, @gioitinh, @diachi, @sodt, @email,@phong)
            else
                update nhanvien set
                    tenvn = @tenvn,
                    gioitinh = @gioitinh,
                    diachi = @diachi,
```

```

        sodt = @sodt,
        email = @email,
        phong = @phong
    where manv=@manv
end
end

```

■ Gọi thủ tục:

```

declare @bien int
--Thêm mới
exec sp_ThemNhanvien 'nv01', N'Nguyễn Văn A', N'Nam', N'Hà Nội',
'099887666', 'vanA@gmail.com', N'Hành chính', 0, @bien output
--Cập nhật
exec sp_ThemNhanvien 'nv01', N'Nguyễn Văn A', N'Nữ', N'Hà Nội',
'099887666', 'vanA@gmail.com', N'Tổ chức', 1, @bien output
--Sai thông tin gioitinh
exec sp_ThemNhanvien 'nv01', N'Nguyễn Văn A', N'NonSexual', N'Hà
Nội', '099887666', 'vanA@gmail.com', N'Tổ chức', 1, @bien output
select * from Nhanvien
select @bien

```

2. Viết thủ tục thêm mới sản phẩm với các tham biến masp, tenhang, tensp, soluong, mausac, giaban, donvitinh, mota và 1 biến Flag. Nếu Flag=0 thì thêm mới sản phẩm, ngược lại cập nhật sản phẩm. Hãy kiểm tra:

- Nếu tenhang không có trong bảng hangsx thì trả về mã lỗi 1
 - Nếu soluong <0 thì trả về mã lỗi 2
 - Ngược lại trả về mã lỗi 0.
-

```

create proc sp_ThemMoiSP(
    @masp nvarchar(10),
    @tenhang nvarchar(20),
    @tensp nvarchar(20),
    @soluong int,
    @mausac nvarchar(20),
    @giaban money,
    @donvitinh nvarchar(10),
    @mota nvarchar(max),
    @Flag int,
    @KQ int output
)
as
begin
    if(not exists(select * from hangsx where tenhang=@tenhang))

```

```

        set @KQ=1
    else if(@soluong <0)
        set @KQ=2
    else
        begin
            set @KQ = 0
            declare @mahangsx nvarchar(10)
            set @mahangsx = (select mahangsx
                                from hangsx
                                where tenhang=@tenhang)
            if(@Flag=0)
                insert into sanpham values(@masp,
@mahangsx, @tensp, @soluong, @mausac, @giaban, @donvitinh, @mota)
            else
                update sanpham set
                    mahangsx=@mahangsx,
                    tensp=@tensp,
                    soluong=soluong,
                    mausac=@mausac,
                    giaban=@giaban,
                    donvitinh=@donvitinh,
                    mota=@mota
                where masp = @masp
        end
    end

```

■ Gọi thủ tục:

```

declare @bien int
--Thêm mới
exec sp_Themmoisp 'sp01', N'SAMSUNG', N'Galaxy J3', 200, N'Xanh',
2500000, N'Chiếc', N'Hàng cận cao cấp', 0, @bien output
--Cập nhật
exec sp_Themmoisp 'sp01', N'SAMSUNG', N'Galaxy J3', 170, N'Xanh',
2340000, N'Chiếc', N'Hàng cận date', 1, @bien output
--Tên hàng không có
exec sp_Themmoisp 'sp01', N'SÂMSUT', N'Galaxy J3', 200, N'Xanh',
2500000, N'Chiếc', N'Hàng cận cao cấp', 0, @bien output
--Số lượng <0
exec sp_Themmoisp 'sp01', N'SAMSUNG', N'Galaxy J3', -200, N'Xanh',
2500000, N'Chiếc', N'Hàng cận cao cấp', 0, @bien output
select * from Sanpham
select @bien

```

3. Viết thủ tục xóa dữ liệu bảng nhanvien với tham biến là manv. Nếu manv chưa có thì trả về 1, ngược lại xóa nhanvien với nhanvien bị xóa là manv và trả về 0. (Lưu ý: xóa nhanvien thì phải xóa các bảng Nhap, Xuat mà nhân viên này tham gia).
4. Viết thủ tục xóa dữ liệu bảng sanpham với tham biến là masp. Nếu masp chưa có thì trả về 1, ngược lại xóa sanpham với sanpham bị xóa là masp và trả về 0. (Lưu ý: xóa sanpham thì phải xóa các bảng Nhap, Xuat mà sanpham này cung ứng).
5. Tạo thủ tục nhập liệu cho bảng Hangsx, với các tham biến truyền vào mahangsx, tenhang, diachi, sodt, email. Hãy kiểm tra xem tenhang đã tồn tại trước đó hay chưa, nếu rồi trả về mã lỗi 1? Nếu có rồi thì không cho nhập và trả về mã lỗi 0.
6. Viết thủ tục nhập dữ liệu cho bảng Nhap với các tham biến sohdn, masp, manv, ngaynhap, soluongN, dongiaN. Kiểm tra xem masp có tồn tại trong bảng Sanpham hay không, nếu không trả về 1? manv có tồn tại trong bảng nhanvien hay không nếu không trả về 2? ngược lại thì hãy kiểm tra: Nếu sohdn đã tồn tại thì cập nhật bảng Nhap theo sohdn, ngược lại thêm mới bảng Nhap và trả về mã lỗi 0.
7. Viết thủ tục nhập dữ liệu cho bảng xuat với các tham biến sohdx, masp, manv, ngayxuat, soluongX. Kiểm tra xem masp có tồn tại trong bảng Sanpham hay không nếu không trả về 1? manv có tồn tại trong bảng nhanvien hay không nếu không trả về 2? soluongX <= Soluong nếu không trả về 3? ngược lại thì hãy kiểm tra: Nếu sohdx đã tồn tại thì cập nhật bảng Xuat theo sohdx, ngược lại thêm mới bảng Xuat và trả về mã lỗi 0.

GHI CHÚ:

BÀI THỰC HÀNH SỐ 9

A. Nhắc lại lý thuyết về Trigger – Bẫy lỗi:

Trigger là dạng đặt biệt của Store Procedure (SP), dùng khai báo ràng buộc dữ liệu cho một table, View và tự động thực hiện khi một trong 3 phát biểu Insert, Update, Delete thay đổi dữ liệu trên table đó. Trigger không được gọi trực tiếp như SP, không có tham số và giá trị trả về như SP.

Trigger chỉ được thực hiện khi phát biểu cập nhật đã thỏa mãn các ràng buộc đã khai báo trên Table. Lợi ích chính của triggers là chúng có thể chứa các xử lý phức tạp trên các table có dữ liệu liên quan với table đang cập nhật.

Dựa vào ứng dụng của Trigger, có 3 loại Trigger như sau: Insert Trigger; Update Trigger; Delete Trigger

- Cú pháp Tạo Trigger:

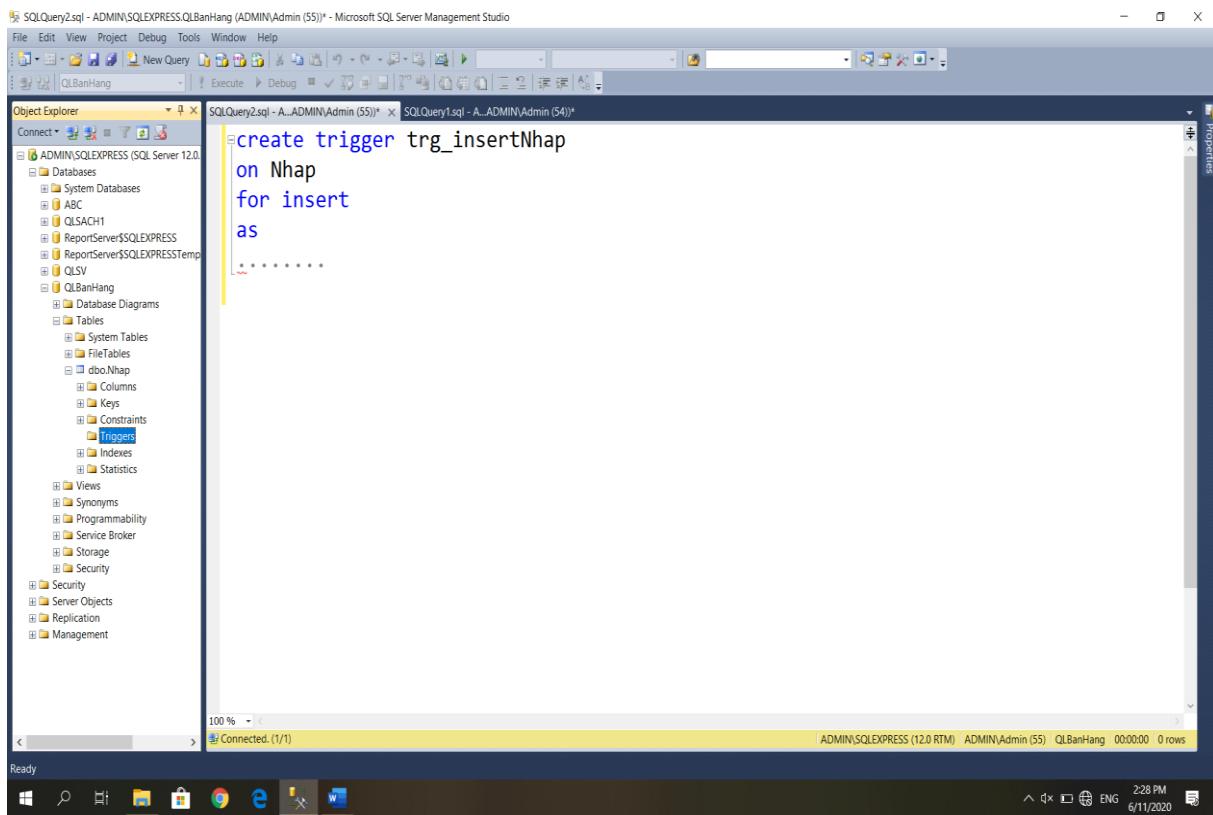
```
CREATE TRIGGER <trigger_name> ON <table name> [WITH ENCRYPTION]
    AFTER | FOR DELETE, INSERT, UPDATE
    AS    <Các phát biểu T-sql>
```

- trigger_name : Tên Trigger phải phân biệt.
- ON <tablename> : tên table mà Trigger sẽ thực hiện. Không sử dụng Trigger cho View.
- WITH ENCRYPTION : Mã hóa Trigger, không cho xem và sửa đổi câu lệnh tạo Trigger..
- FOR DELETE, INSERT, UPDATE - Dùng chỉ định những phát biểu cập nhật nào nào trên Table sẽ kích hoạt Trigger.

- Khi thực hiện Trigger, SQL sẽ tạo các bảng tạm: INSERTED và DELETED

- **Khi Insert** mẫu tin mới vào Table thì mẫu tin mới đó cũng lưu trong table INSERTED
- **Khi Delete** mẫu tin trong table: Thì các mẫu tin bị xoá đó được di chuyển sang table Deleted.
- **Khi Update** mẫu tin trong table: thì table đó và table Inserted đều chứa mẫu tin có nội dung mới, còn Deleted chứa mẫu tin có nội dung cũ.

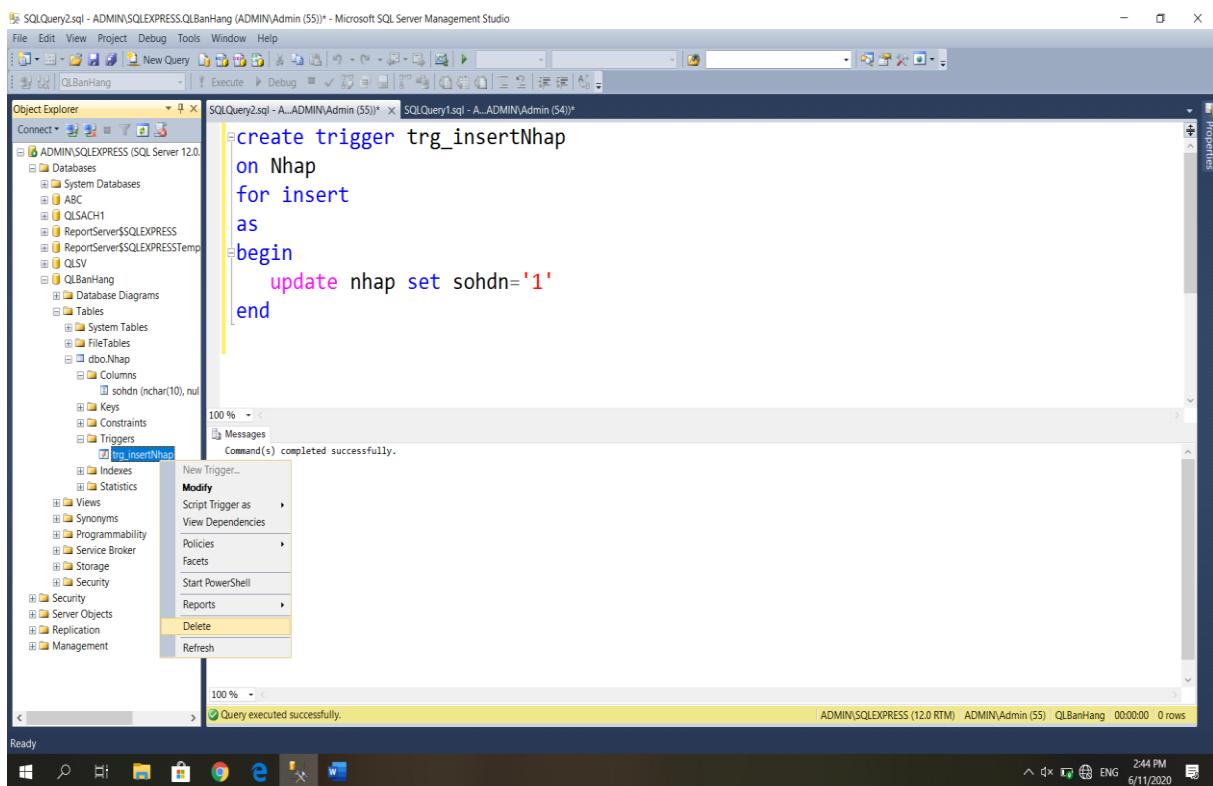
- Nơi Lưu trữ trigger: Trigger được gắn kèm với bảng mà trigger sẽ thực hiện.



Hình 9.1. Nơi lưu trữ trigger

- Xóa trigger:

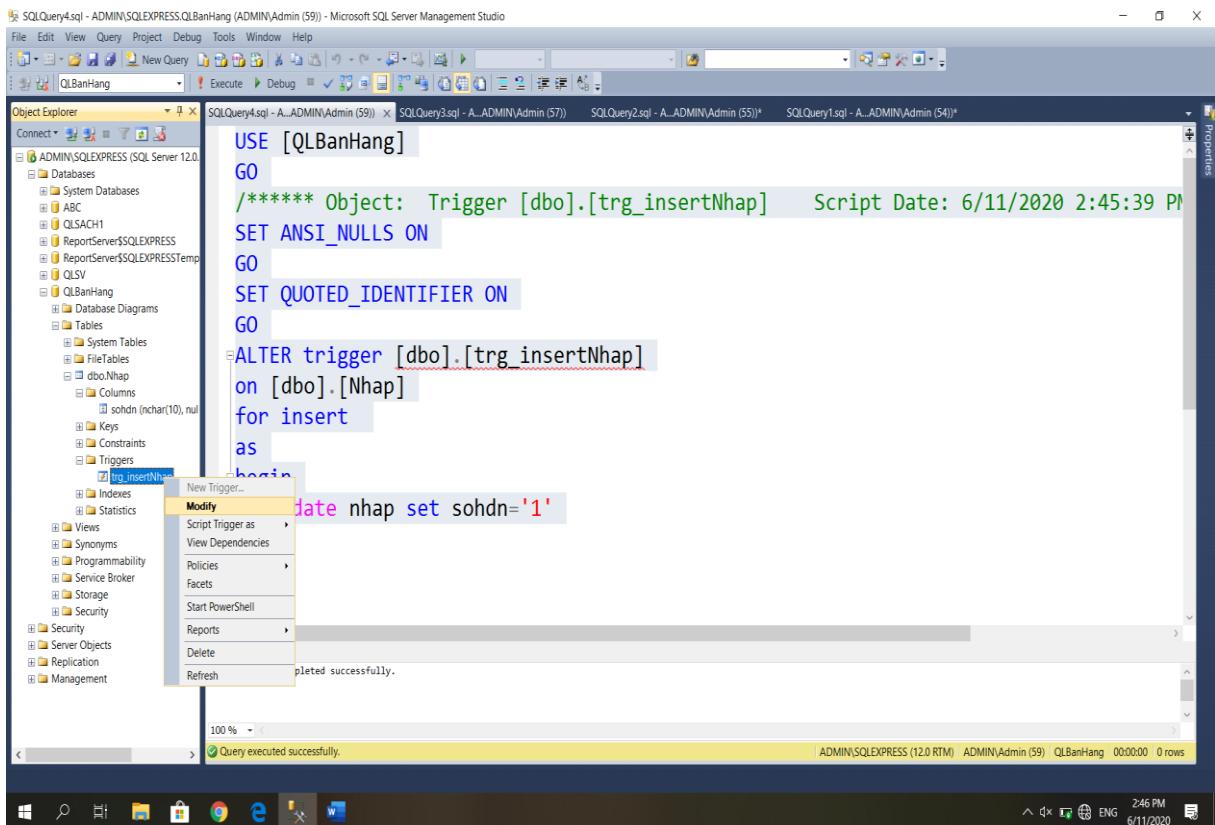
- ✓ drop proc tentrigger
- ✓ Hoặc: click chuột phải trên tên trigger -> Delete.



Hình 9.2. Xóa trigger wizard

- Sửa lại nội dung Trigger:

- ✓ alter Trigger tentrigger... - Các thay đổi nội dung trigger
- ✓ Hoặc:



Hình 9.3. Sửa lại nội dung trigger wizard

❖ *Tóm lại, việc phân chia, tổ chức các kiểm tra ràng buộc toàn vẹn dữ liệu phức tạp hoặc các cập nhật dữ liệu tự động trong đối tượng trigger sẽ làm cho các xử lý được tập trung tại máy chủ và độc lập với ngôn ngữ lập trình tại máy trạm. Điều này làm cho tốc độ của các ứng dụng theo mô hình khách chủ được nhanh hơn.*

B. Từ CSDL QLBanHang đã được tạo ra ở bài thực hành số 1:

Sanpham(masp, mahangsx, tensp, soluong, mausac, giaban, donvitinh, mota)

Hangsx(mahangsx, tenhang, diachi, sode, email)

Nhanvien(manv, tennv, gioitinh, diachi, sode, email, phong)

Nhap(sohdn, masp, manv, ngaynhap, soluongN, dongiaN)

Xuat(sohdx, masp, manv, ngayxuat, soluongX)

Hãy tạo các Trigger kiểm soát ràng buộc toàn vẹn và kiểm tra ràng buộc dữ liệu sau:

1. Tạo trigger kiểm soát việc nhập dữ liệu cho bảng nhập, hãy kiểm tra các ràng buộc toàn vẹn: masp có trong bảng sản phẩm chưa? manv có trong bảng nhân viên chưa? kiểm tra các ràng buộc dữ liệu: soluongN và dongiaN>0? Sau khi nhập thì soluong ở bảng Sanpham sẽ được cập nhật theo.

```
create trigger trg_Nhap
on Nhaph
for insert
as
begin
    declare @masp nvarchar(10),@manv nvarchar(10)
    declare @sln int, @dgn float
    select @masp=masp, @manv=manv, @sln=soluongN, @dgn = dongiaN
    from inserted
    if(not exists(select * from sanpham where masp = @masp))
        begin
            raiserror(N'Không tồn tại sản phẩm trong danh mục sản
phẩm',16,1)
            rollback transaction
        end
    else
        if(not exists(select * from nhanvien where manv= @manv))
            begin
                raiserror(N'Không tồn tại nhân viên có mã
này',16,1)
                rollback transaction
            end
        else
            if(@sln<=0 or @dgn<=0)
                begin
                    raiserror(N'Nhập sai soluong hoặc dongia', 16, 1)
                    rollback transaction
                end
            else
                --Bây giờ mới được phép nhập, khi này cần thay đổi soluong
                --trong bảng Sanpham
                update Sanpham set soluong = soluong + @sln
                from sanpham where masp = @masp
end
```

■ Thực thi trigger:

-- Gọi dữ liệu 3 bảng liên quan

```
select * from sanpham
select * from Nhanvien
```

```
select * from nhap
```

	masp	mahangsx	tensp	soluong	mausac	giaban	donvitinh	mota
1	sp01	H01	Galaxy J3	200	Xanh	2500000	Chiếc	Hàng cận cao cấp
	manv	tennv	diachi	gioitinh	sodt	email	phong	
1	nv01	Nguyễn Văn A	Hà Nội	Nam	099887666	vanA@gmail.com	Hành chính	
	sohdn	masp	manv	ngaynhap	soluongN	dongiaN		

Hình 9.4. Dữ liệu ban đầu của các bảng

-- Nhập sai:

```
insert into nhap values('nh01', 'sp01', 'nv01', '3/7/2018', 0,  
1500000)
```

-- Báo lỗi

Messages

```
Msg 50000, Level 16, State 1, Procedure trg_Nhap, Line 21  
Nhập sai soluong hoặc dongia  
Msg 3609, Level 16, State 1, Line 1  
The transaction ended in the trigger. The batch has been aborted.
```

Hình 9.5. Màn hình thông báo lỗi ràng buộc trigger

-- Nhập đúng:

```
insert into nhap values('nh01', 'sp01', 'nv01', '3/7/2018',  
300,1500000)  
select * from nhap  
select * from sanpham
```

-- Màn hình kết quả:

	masp	mahangsx	tensp	soluong	mausac	giaban	donvitinh	mota
1	sp01	H01	Galaxy J3	500	Xanh	2500000	Chiếc	Hàng cận cao cấp
	sohdn	masp	manv	ngaynhap	soluongN	dongiaN		
1	nh01	sp01	nv01	2018-03-07 00:00:00.000	300	1500000		

Hình 9.6. Kết quả thay đổi sau khi thực thi trigger

- Như vậy soluong sanpham đã thay đổi theo soluongN trong bang Nhap.
- Test ràng buộc toàn vẹn không được thực thi bởi vì kiểm tra toàn vẹn khóa ngoại đã được sqlserver kiểm tra trước khi chạy trigger, việc kiểm tra toàn vẹn trong trigger thích hợp khi chúng ta tạo ra các bảng mà không tạo kết nối, dùng trigger để kiểm tra toàn vẹn.

2. Tạo trigger kiểm soát việc nhập dữ liệu cho bảng xuất, hãy kiểm tra các ràng buộc toàn vẹn: masp có trong bảng sản phẩm chưa? manv có trong bảng nhân viên chưa? kiểm tra các ràng buộc dữ liệu: soluongX < soluong trong bảng sanpham? Sau khi xuất thì soluong ở bảng Sanpham sẽ được cập nhật theo.

```
create trigger trg_xuat
on xuat
for insert
as
begin
    declare @masp nvarchar(10),@manv nvarchar(10)
    declare @slx int, @soluong float
    select @masp=masp,@manv=manv,@slx=soluongX from inserted
    if(not exists(select * from sanpham where masp = @masp))
        begin
            raiserror(N'Không tồn tại sản phẩm trong danh mục sản
phẩm',16,1)
            rollback transaction
        end
    else
        if(not exists(select * from nhanvien where manv= @manv))
            begin
                raiserror(N'Không tồn tại nhân viên có mã
này',16,1)
                rollback transaction
            end
        else
            begin
                select @soluong = soluong
                from sanpham where masp = @masp
                if(@slx>@soluong)
                    begin
                        raiserror(N'Không đủ số lượng sản phẩm để
xuất',16,1)
                        rollback transaction
                    end
            end
    end
```

```

        end
    else
--Bây giờ mới được phép nhập, khi này cần thay đổi
--soluong trong bảng Sanpham
        update Sanpham set soluong = soluong - @slx
        from sanpham where masp = @masp
    end
end

```

■ Thực thi trigger:

```

select * from sanpham
select * from Nhanvien
select * from xuat
--Sai số lượng
insert into xuat values('x01', 'sp01', 'nv01', '3/7/2018', 600)
--Đúng
insert into xuat values('x01', 'sp01', 'nv01', '3/7/2018', 170)
select * from sanpham
select * from xuat

```

3. Tạo trigger kiểm soát việc xóa phiếu xuất, khi phiếu xuất xóa thì số lượng hàng trong bảng sanpham sẽ được cập nhật tăng lên.

```

create trigger trg_XoaXuat
on xuat
for delete
as
begin
    declare @masp nvarchar(10)
    declare @slx int
    select @masp=masp,@slx = soluongX from deleted
    if(not exists(select * from sanpham where masp = @masp))
        begin
            raiserror(N'Không tồn tại sản phẩm trong danh mục sản
phẩm',16,1)
            rollback transaction
        end
    else
        update sanpham set soluong = soluong + @slx
        from sanpham where masp = @masp
end

```

■ Thực thi trigger:

```
--Test
select * from Sanpham
select * from xuat
delete from xuat where sohdx = 'x01'
select * from Sanpham
select * from xuat
```

	masp	mahangsx	tensp	soluong	mausac	giaban	donvitinh	mota
1	sp01	H01	Galaxy J3	330	Xanh	2500000	Chiếc	Hàng cận cao cấp

	sohdx	masp	manv	ngayxuat	soluongX
1	x01	sp01	nv01	2018-03-07 00:00:00.000	170

	masp	mahangsx	tensp	soluong	mausac	giaban	donvitinh	mota
1	sp01	H01	Galaxy J3	500	Xanh	2500000	Chiếc	Hàng cận cao cấp

	sohdx	masp	manv	ngayxuat	soluongX
--	-------	------	------	----------	----------

Hình 9.7. Kết quả sau khi thực thi trigger

4. Tạo trigger cho việc cập nhật lại số lượng xuất trong bảng xuất, hãy kiểm tra xem số lượng xuất thay đổi có nhỏ hơn soluong trong bảng sanpham hay ko? số bản ghi thay đổi >1 bản ghi hay không? nếu thỏa mãn thì cho phép update bảng xuất và update lại soluong trong bảng sanpham.

```
create trigger trg_capnhatXuat
on xuat
for update
as
begin
    --Hàm hệ thống trả về số record sẽ bị cập nhật
    if(@@rowcount>1)
        begin
            raiserror(N'Không được cập nhật nhiều hơn 1 bản ghi
cùng lúc',16,1)
            rollback transaction
        end
    else
        begin
            declare @truoc int, @sau int, @slco int
            declare @masp nvarchar(10)
            select @truoc = soluongX,@masp = masp from deleted
            select @sau =soluongX from inserted
```

```

select @slco = soluong from sanpham where masp = @masp
-- Ở đây chỉ cần gọi if(update(soluongX))
if(@truoc<>@sau)
begin
    --Kiểm tra xem có phải cột soluongX được update
    if(@slco < (@sau - @truoc))
        begin
            raiserror(N'Không đủ hàng xuất',16,1)
            rollback transaction
        end
    else
        update sanpham
            set soluong = soluong - (@sau-@truoc)
        from sanpham
        where masp=@masp
    end
end

```

■ Thực thi trigger:

```

select * from Sanpham
select * from xuat
--Xuất trên 2 sản phẩm, không thể cập nhật thế này được
update xuat set soluongX=20 where sohdx='x02'
--Phải chỉ rõ sản phẩm nào
update xuat set soluongX=20 where sohdx='x02' and masp='sp01'
select * from Sanpham
select * from xuat
-- Khi này số lượng trong bảng sanpham sẽ thay đổi theo

```

The screenshot shows the execution results of the trigger. The first result set is for the 'Sanpham' table, which contains three rows of product information. The second result set is for the 'xuat' table, which contains five rows of shipping information.

	masp	mahangsx	tensp	soluong	mausac	giaban	donvitinh	mota
1	sp01	H01	Galaxy J3	450	Xanh	2500000	Chiếc	Hàng cao cấp
2	sp02	H01	Galaxy Note9	296	Tím	9000000	Chiếc	Hàng Cao cấp
3	sp03	H01	Galaxy A3	988	Trắng	1300000	Chiếc	Hàng Phổ thông

	sohdx	masp	manv	ngayxuat	soluongX
1	x01	sp01	nv01	2018-03-07 00:00:00.000	10
2	x02	sp01	nv01	2018-03-07 00:00:00.000	30
3	x02	sp02	nv01	2018-03-07 00:00:00.000	20
4	x03	sp02	nv01	2018-03-07 00:00:00.000	7
5	x03	sp03	nv01	2018-03-07 00:00:00.000	6

Hình 9.8. Kết quả sau khi thực thi Trigger

5. Tạo trigger cho việc cập nhật lại số lượng Nhập trong bảng Nhập, Hãy kiểm tra xem số bản ghi thay đổi >1 bản ghi hay không? nếu thỏa mãn thì cho phép update bảng Nhập và update lại số lượng trong bảng sanpham.
6. Tạo trigger kiểm soát việc xóa phiếu nhập, khi phiếu nhập xóa thì số lượng hàng trong bảng sanpham sẽ được cập nhật giảm xuống.

GHI CHÚ:

BÀI THỰC HÀNH SỐ 10

A. Nhắc lại lý thuyết về BẢO MẬT (SECURITY)

Mỗi CSDL có 1 hay nhiều users được chỉ định quyền truy xuất dữ liệu. Người quản trị có thể cấp quyền truy xuất CSDL bằng cách tạo một tài khoản đăng nhập (login) SQL Server cho User, thêm User vào CSDL và gán quyền cho User trên CSDL đó.

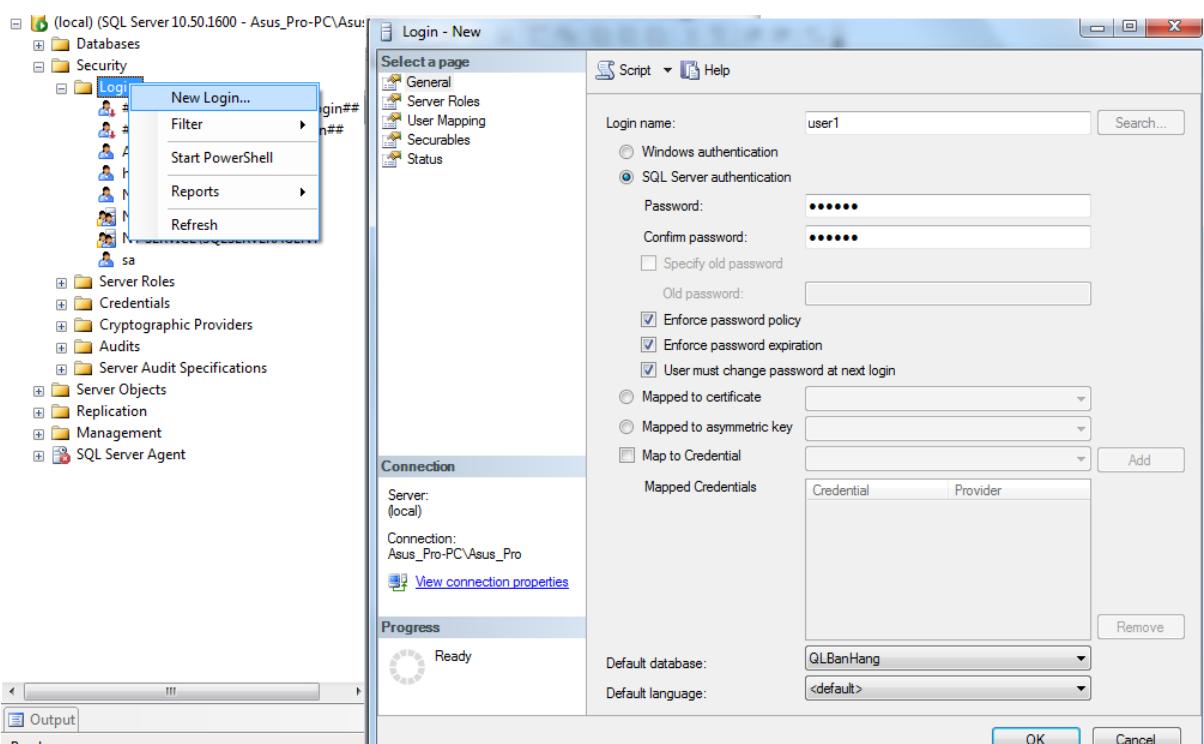
Bao gồm các loại quyền:

- Quyền truy cập vào SQL Server
 - Quyền truy xuất CSDL
 - Quyền thực hiện trên các đối tượng của CSDL
 - Quyền xử lý dữ liệu
- **Có 2 cách tạo/ cấp quyền cho CSDL:**
- ✓ Sử dụng EM
 - ✓ Sử dụng QA

❖ Để đơn giản hơn trong cách sử dụng, chúng tôi đơn giản hóa bằng việc sử dụng EM trong việc tạo bảo mật đối với CSDL trong chương này.

- Tạo Tài Khoản Đăng Nhập (Login Account):

- (1) Mở mục Security, click phải mục Login và chọn New Login...
- (2) Trên trang General, nhập tên đăng nhập, chế độ xác nhận, CSDL mặc định.



Hình 10.1. Tạo tài khoản đăng nhập Login

- (3) Trang Server Roles: Chọn vai trò quản trị mức Server cho tài khoản đăng nhập

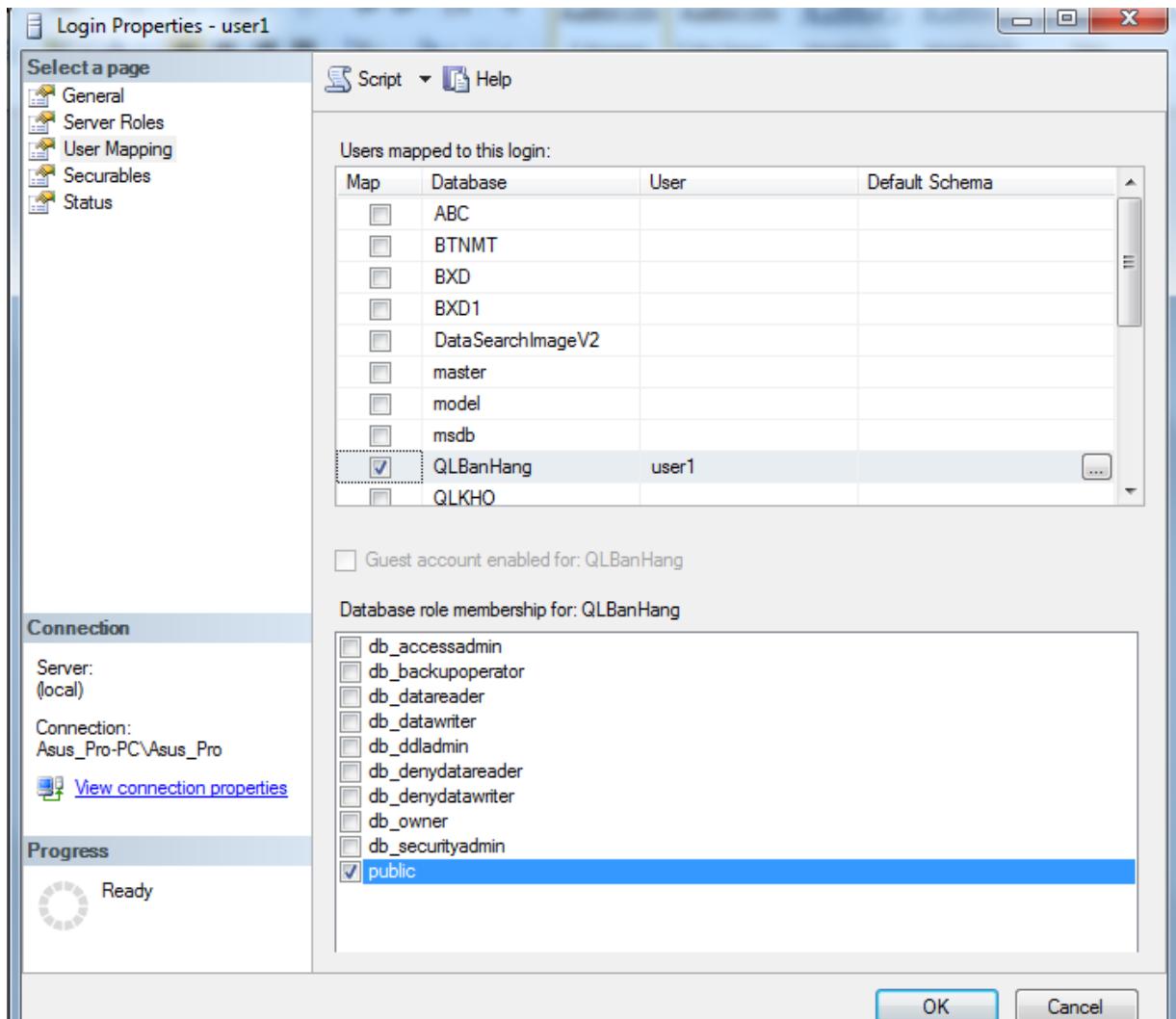
Bảng 10.1. Server Roles: Chọn vai trò quản trị mức Server cho tài khoản đăng nhập

System Administrators	Đặc quyền cao nhất; cho phép thực hiện mọi tác vụ trên SQL
Security Administrators	Quản lý các server logins.
Server Administrators	Cho phép bạn định cấu hình những cài đặt server-wide.
Setup Administrators	Cho phép thêm và xóa các linked servers, và truy xuất vào SP
Database Creator	Tạo và hiệu chỉnh databases.
Disk Administrators	Quản lý các files trên đĩa.
Process Administrators	Quản lý tiến trình đang chạy trong một thê hiện của SQL Server.
Bulk Administrators	Thực hiện phát biểu BULK INSERT.

(4) Trang User Mapping: Chọn CSDL được phép truy xuất và vai trò của nó trong từng CSDL được chọn.

Bảng 10.2. Chọn CSDL được phép truy xuất và vai trò của nó trong từng CSDL được chọn.

Fixed Database	Role Description
Public	Vai trò chung cho tất cả người dùng.
db_owner	Quyền cao nhất trong database.
db_accessadmin	Điều khiển truy xuất, cài đặt hoặc xóa user accounts.
db_datareader	Đọc tất cả dữ liệu trên database.
db_datawriter	Thêm, sửa, xóa dữ liệu trên các tables người dùng trong database.
db_ddladmin	Thêm, sửa, xóa các đối tượng objects (runs all DDLs).
db_securityadmin	Quản lý các roles, các thành viên của role, giấy phép trong database.
db_backupoperator	Cho phép back up database.
db_denydatareader	Từ chối quyền truy vấn dữ liệu trong database.
db_denydatawriter	Từ chối quyền thay đổi dữ liệu trong database.



Hình 10.2. Chọn CSDL được phép truy xuất và vai trò của nó trong từng CSDL được chọn

Sau khi tạo login, nó tự động nhập vào tập Users của mỗi database được chọn, với tên User trùng với tên Login. Bạn có thể thay đổi vai trò của nó trên từng CSDL bằng cách sửa đổi thuộc tính của Login, hoặc chuyển đến tập Users của database và thay đổi thuộc tính hoặc xoá user nào mà bạn không muốn cho truy xuất data của bạn.

Các Login được lưu trong table SysLogins của CSDL Master:

```
if exists(select 1 From Master..SysLogins Where Name = 'Login')
```

Các User trong một CSDL được lưu trong table SysUsers của CSDL đó

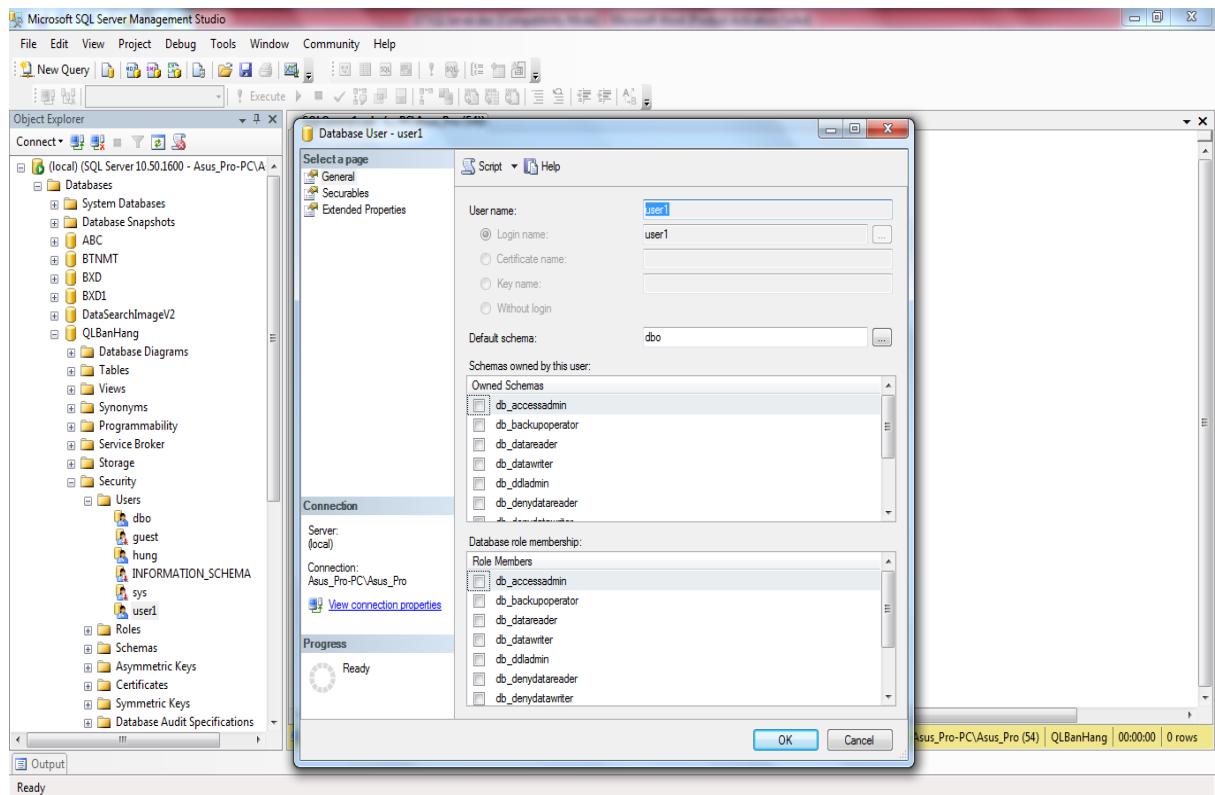
```
if exists(select 1 From SysUsers Where Name = 'User01')
```

- **Thay đổi thuộc tính cho Login:**

Bấm đúp vào tên Login hoặc click phải và chọn mục Properties

- **Cấp Quyền Thực Thi Trên Mọi CSDL:**

Chọn database, trong mục Users bấm đúp vào tên User cần hiệu chỉnh (Login-ID).



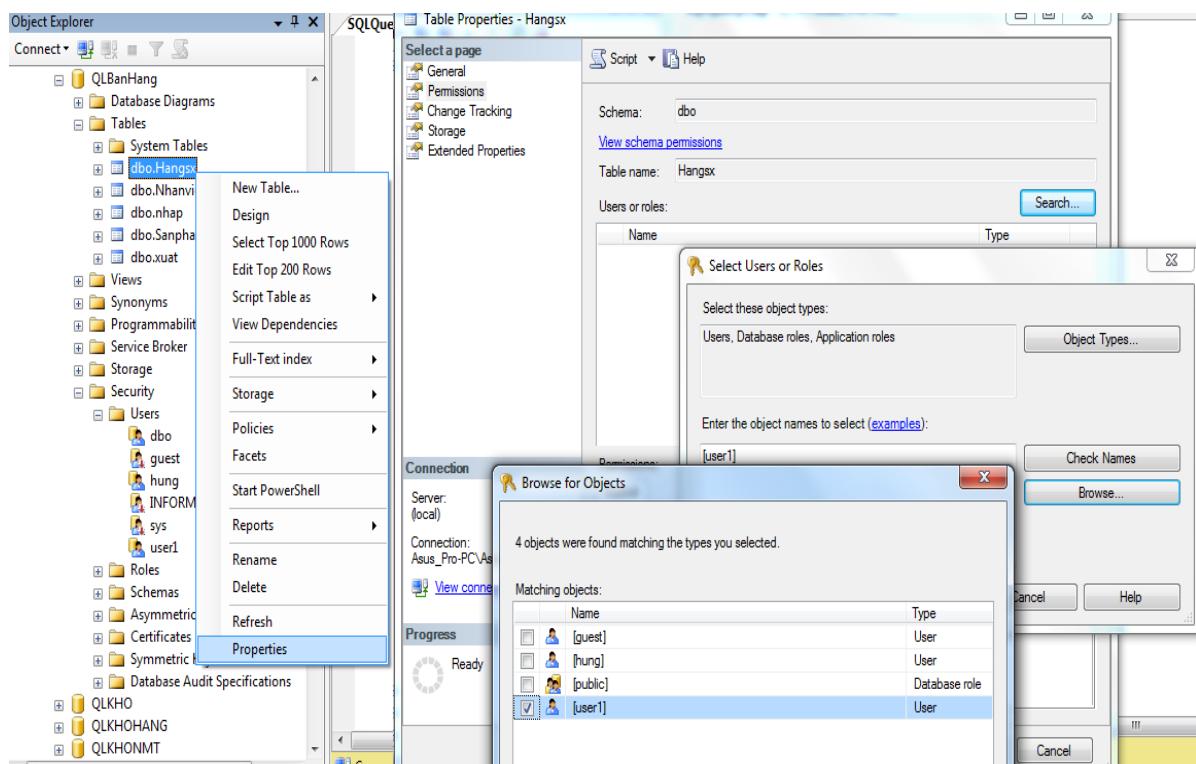
Hình 10.3. Cấp Quyền Thực Thi Trên Mô hình CSDL

Click nút Permission để chỉ định quyền truy cập dữ liệu trên từng Table, View. Quyền kiểm tra RB tham chiếu (DRI - Declarative Referential Integrity). Quyền thực hiện các thủ tục lưu trữ.

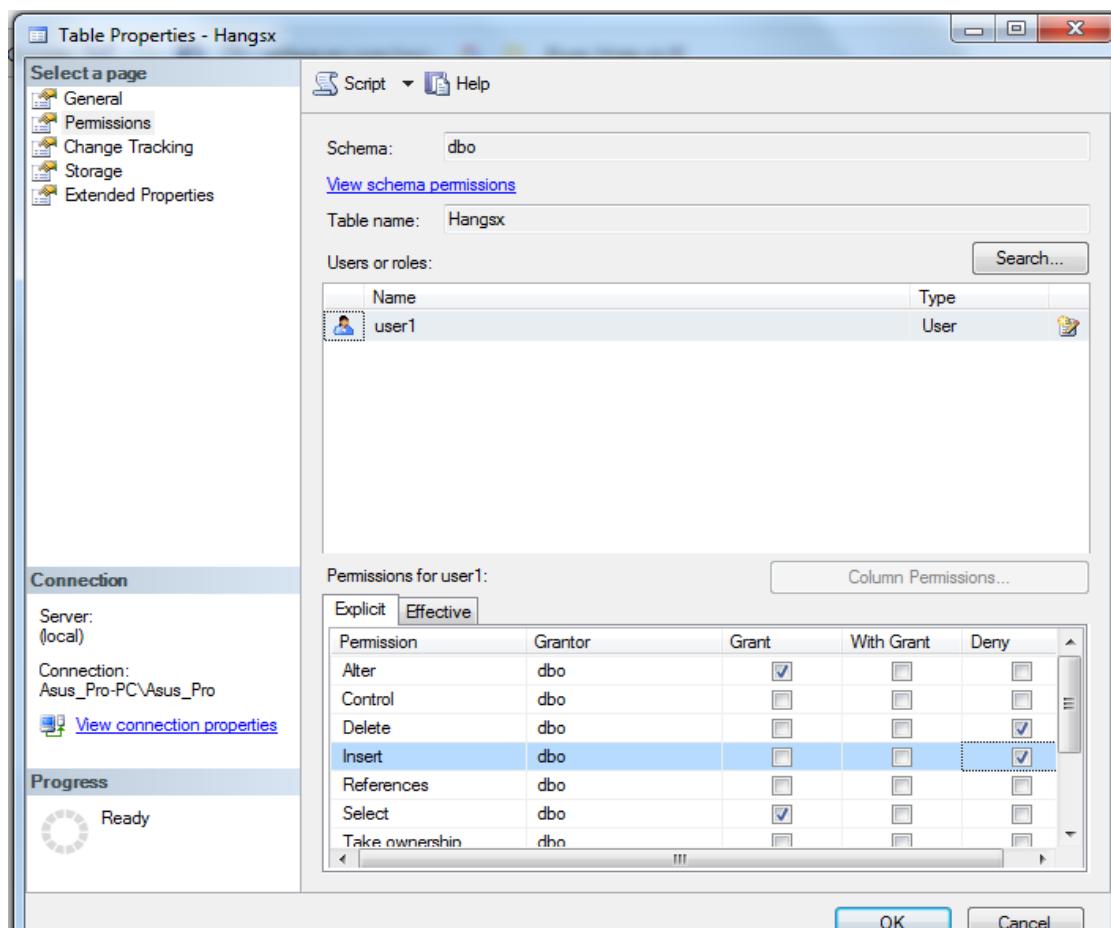
Chú thích: Quyền kiểm tra RB tham chiếu được sử dụng khi Table A được cấp quyền Update hoặc Insert. Table A có RB FOREIGN KEY với table B, mà table B không được cấp quyền SELECT.

- **Cấp Quyền Tạo Đôi Tượng Trên CSDL:**

- Click phải vào tên CSDL, chọn Properties
- Trong HT Properties, chọn trang Permissions
- Đánh dấu chọn các phát biểu được cần cấp quyền thực hiện cho các User.



Hình 10.4. Cấp Quyền Tạo Đối Tượng Trên CSDL



Hình 10.5. Đánh dấu chọn các phát biểu được càn cấp quyền thực hiện cho các User

- **Xóa User trên một CSDL:**

Click phải vào tên user và chọn Delete.

- **Tạo User với Login đã có:**

Click phải vào mục **Users** và chọn **New Database User...**

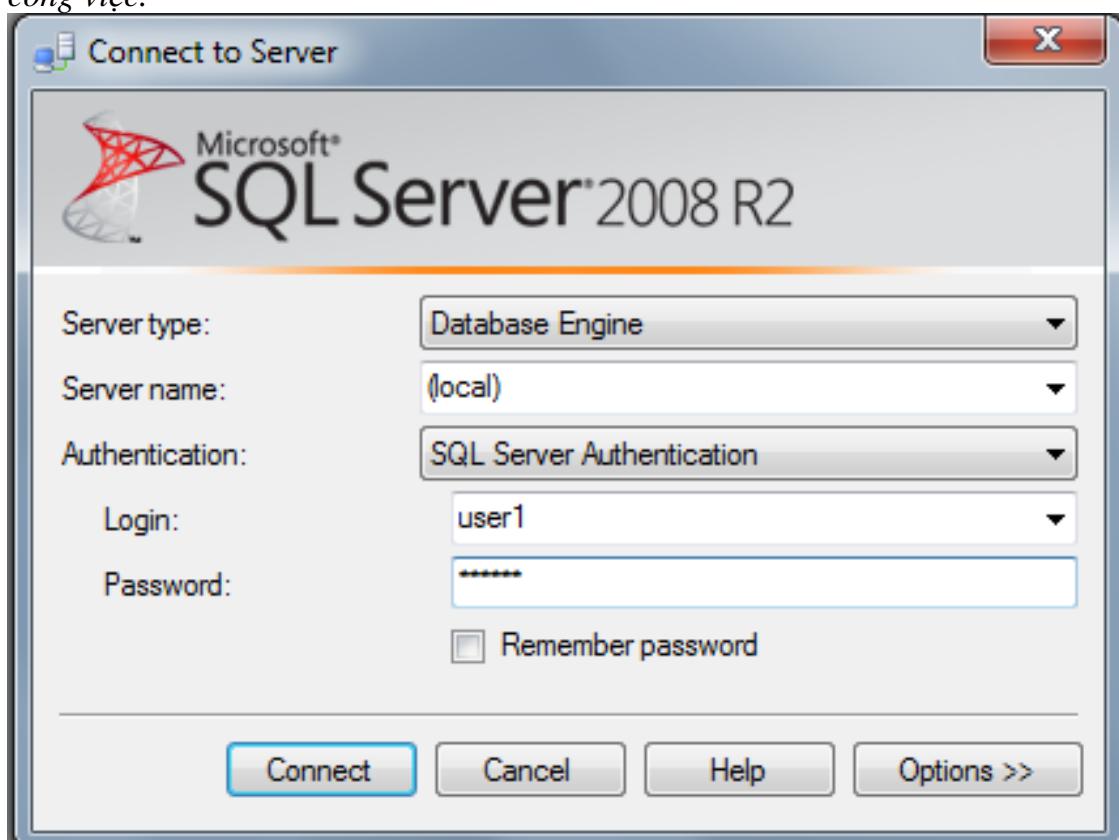
Chú ý: Bạn có thể thay đổi tên User bằng cách xóa và tạo lại với login cũ.

- **Xóa Tài Khoản Đăng Nhập:**

Click phải vào tên Login và chọn Delete

Với cách này sẽ xóa bỏ các user trong các CSDL đã chọn cho tài khoản này.

- ❖ Sau khi tạo xong các tài khoản và cấp quyền cho các user với các table, view. Chúng ta disconnect với tài khoản “sa”, đăng nhập với tài khoản mới và kiểm tra công việc.



Hình 10.6. Đăng nhập với tài khoản mới và kiểm tra công việc

B. Từ CSDL QLBanHang đã được tạo ra ở bài thực hành số 1:

Bảng 10.3. Tạo bảng đăng nhập

Tên login	Tên user	Tên role
TruongLV	TruongLV	PhongKT
AnLV	AnLV	

Hãy tạo 1 Login mới có quyền đăng nhập và có mọi quyền với CSDL này, và tạo các user mới, cấp quyền cho các user này với các table, view.

- Tạo login và user

Bảng 10.4. Bảng phân quyền login và user

Tên user/Tên nhóm	Quyền
PhongKT	Tạo database, tạo bảng
AnLV	Xem toàn bộ cơ sở dữ liệu, chỉ thêm dữ liệu trên bảng PhongMay và cho phép cấp lại quyền

- Phân quyền

➔ Gợi ý

```
-- Câu lệnh cấp quyền tạo cơ sở dữ liệu  
grant alter any database to Tên_Login/Tên_Role  
-- Câu lệnh cấp quyền tạo bảng  
grant create table to Tên_User/Tên_Role
```

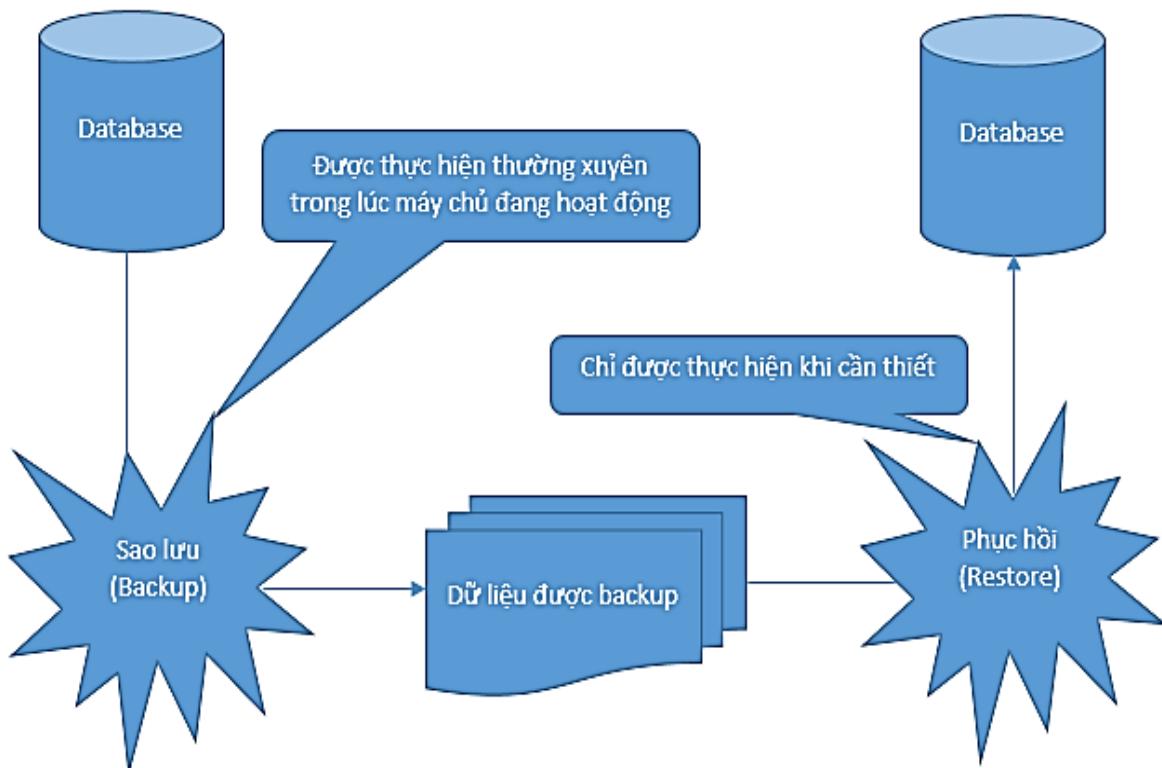
GHI CHÚ:

PHỤ LỤC: SAO LUU VÀ PHỤC HỒI DỮ LIỆU TRONG SQL SERVER

Sao lưu và phục hồi dữ liệu trong sql server là một trong những thao tác quan trọng mà người quản trị cơ sở dữ liệu phải thực hiện.

Nếu như thao tác sao lưu (Backup database) được thực hiện để lưu dữ liệu và được thực hiện thường xuyên thì thao tác phục hồi dữ liệu (Restore database) chỉ được thực hiện khi nào máy chủ bị sự cố như hư ổ cứng hoặc dữ liệu bị mất do người dùng vô tình hoặc cố ý xoá,...

Ngoài ra thao tác phục hồi dữ liệu cũng được thực hiện để sao chép database từ máy chủ này sang máy chủ khác.



Hình 11.1. Mô hình sao lưu và phục hồi dữ liệu

A. Các loại sao lưu dữ liệu

- Loại 1: Full backup

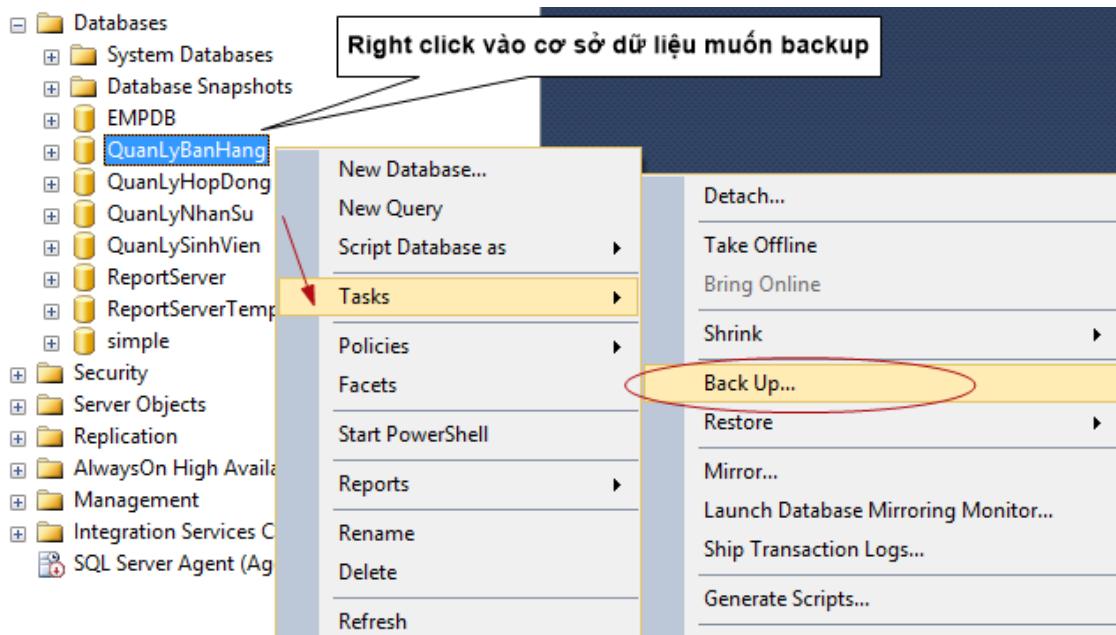
Loại backup này sẽ backup dữ liệu đầy đủ nhất, vì vậy máy chủ sẽ mất nhiều thời gian để thực hiện nếu database lớn

Cú pháp:

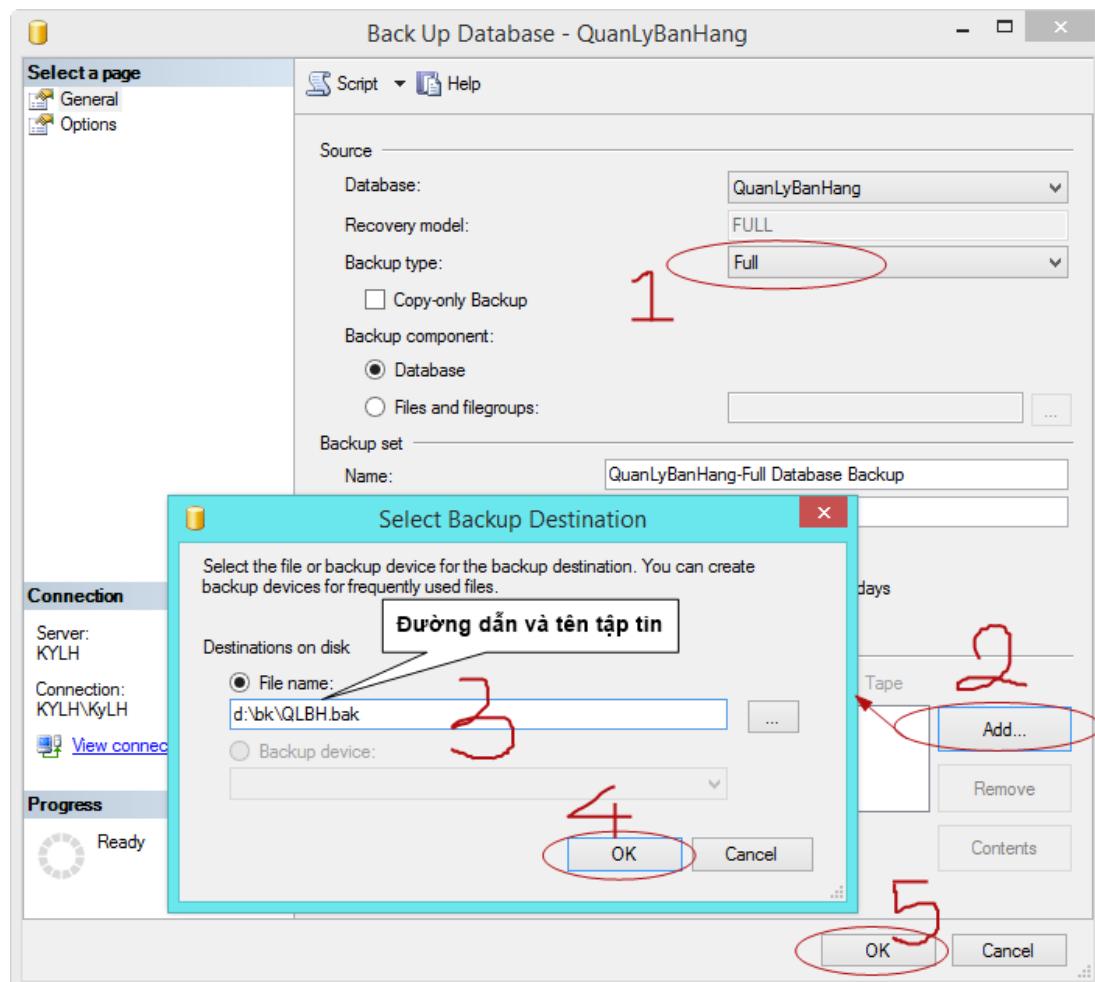
```
backup database Tên_Cơ_Sở_Dữ_Liệu  
to disk = 'Đường_dẫn\Tên_file.bak'
```

Ví dụ câu lệnh full backup cơ sở dữ liệu QuanLyBanHang và lưu vào “d:\bk” (Lưu ý chúng ta phải tạo thư mục bk trong ổ đĩa D trước khi thực thi câu lệnh sau)

backup database QuanLyBanHang to disk = 'd:\bk\QLBH.bak'



Hình 11.2. Thực hiện full backup bằng giao diện



Hình 11.3. Chọn full backup, tên tập tin

- Loại 2: Different backup

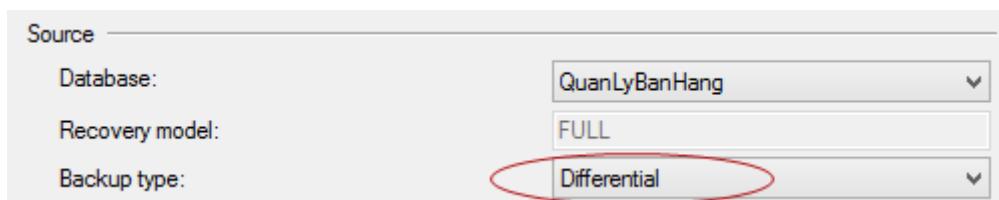
Backup những dữ liệu phát sinh tính từ lần backup full gần nhất. Như vậy trước khi chúng ta thực hiện different backup thì full backup phải được thực hiện trước. Chính vì vậy, khi chúng ta sử dụng loại backup này sẽ tiết kiệm được thời gian backup dữ liệu.

Cú pháp:

```
backup database Tên_Cơ_Sở_Dữ_Liệu  
to disk = 'Đường_dẫn\Tên_File.bak' with differential
```

Ví dụ:

```
backup database QuanLyBanHang to disk = 'D:\sql\QLBH_Diff.bak'  
with differential
```



Hình 11.4. Thực hiện different backup bằng giao diện

- Loại 3: Log backup

Backup dữ liệu sử dụng tập tin log và log backup được sử dụng để backup dữ liệu tại thời điểm cụ thể ví dụ như mỗi giờ trong ngày.

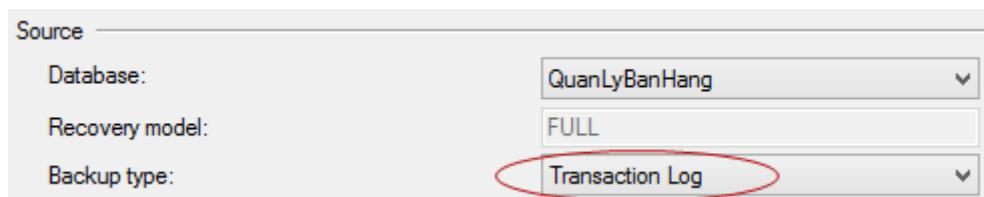
Cú pháp:

```
backup log Tên_Cơ_Sở_Dữ_Liệu to disk = 'Đường_dẫn\Tên_File.trn'
```

Ví dụ:

```
backup log QuanLyBanHang to disk = 'D:\sql\QLBH.trn'
```

Thực hiện log backup bằng giao diện



Hình 11.5. Thực hiện Log backup bằng giao diện

- Kịch bản backup

Kịch bản backup này được thực hiện trên cơ sở dữ liệu QLBanHang

- Thực hiện full backup vào lúc 3h sáng ngày chủ nhật (1 lần trong ngày)
- Thực hiện different backup vào lúc 2h sáng ngày thứ 4 (1 lần trong ngày)

- Thực hiện log backup vào các thời điểm như 2h10', 2h20', 2h40' và 2h50' (4 lần trong một ngày sau thời điểm different backup)

Giả sử máy chủ bị hư ổ cứng vào lúc 2h55' ngày thứ 4. Hỏi kịch bản phục hồi và tình trạng dữ liệu sau khi khôi phục? Câu trả lời ở ngay bên dưới

Đầu tiên chúng ta sẽ thực hiện khôi phục dữ liệu về thời điểm **3h** sáng ngày chủ nhật bằng cách sử dụng bản full backup. Tiếp theo chúng ta sẽ khôi phục dữ liệu về thời điểm **2h** sáng thứ 4 sử dụng bản different backup.

Cuối cùng sử dụng bản log backup ở thời điểm **2h50'** của ngày thứ 4 để đưa dữ liệu về thời điểm 2h50'.

Như vậy dữ liệu chỉ được khôi phục trở lại ở thời điểm 2h50' ngày thứ 4 và chúng ta sẽ bị mất dữ liệu từ 2h51' trở về sau.

B. Phục hồi dữ liệu

- Phục hồi dữ liệu từ bản full và different backup

```
restore database Tên_Cơ_Sở_Dữ_Liệu from disk = 'Đường
dẫn\Tên_File.bak' [With NoRecovery]
```

- Phục hồi dữ liệu từ bản log backup

```
restore log Tên_Cơ_Sở_Dữ_Liệu from disk = 'Đường dẫn\Tên_File.trn'
[With NoRecovery]
```

- Lưu ý:

- Để có thể thực hiện khôi phục dữ liệu từ nhiều bản backup, chúng ta phải chỉ định With NoRecovery ở các câu lệnh restore ngoài trừ câu lệnh restore cuối cùng.
- Ngoài ra chúng ta có thể xem dữ liệu được khôi phục sau mỗi lần thực thi câu lệnh restore bằng cách sử dụng With Standby thay cho With NoRecovery.

C. Sao lưu và phục hồi dữ liệu trong sql server – Ví dụ mẫu

- Sao lưu dữ liệu của cơ sở dữ liệu QLBanHang

Thực hiện full backup:

```
backup database QLBanHang to disk = 'd:\bk\QLBH.bak'
```

Thêm dữ liệu cho bảng NHANVIEN:

```
Insert into NHANVIEN values ('NV04', N'Mai Hạnh Phúc', N'Nữ', N'Hà
Nam', '0928272722', 'phuc@gmail.com', N'Kế toán')
```

Thực hiện different backup:

```
backup database QLBanHang to disk = 'D:\bk\QLBH_Diff.bak' with
differential
```

Tiếp tục thêm dữ liệu cho bảng NHANVIEN:

```
Insert into NHANVIEN values ('NV05', N'Sau Bất Hạnh', N'Nữ', Bắc
Giang', '0998282828', 'hanh@gmail.com', N'Thủ Kho')
```

Thực hiện log backup:

```
backup log QLBanHang to disk = 'D:\bk\QLBH.trn'
```

- Phục hồi dữ liệu cho cơ sở dữ liệu QLBanHang

Giả lập máy chủ bị hư ổ cứng bằng cách xoá cơ sở dữ liệu QLBanHang, sau đó tiến hành khôi phục dữ liệu. Chú ý câu lệnh restore cuối cùng không có With NoRecovery

--Khôi phục dữ liệu từ full backup

```
restore database QuanLyBanHang from disk = 'D:\bk\QLBH.bak' With  
NoRecovery
```

--Khôi phục dữ liệu từ different backup

```
restore database QuanLyBanHang from disk = 'D:\bk\QLBH_Diff.bak'  
With NoRecovery
```

--Khôi phục dữ liệu từ log backup

```
restore log QLBanHang from disk = 'D:\bk\QLBH.trn'
```

D. Với CSDL QLBanHang trên

Câu 1: Thêm dữ liệu và backup

- a. Thêm dữ liệu cho bảng NhanVien sau đó thực hiện full backup
- b. Thêm dữ liệu cho bảng NhanVien sau đó thực hiện different backup
- c. Thêm dữ liệu cho bảng NhanVien sau đó thực hiện log backup (lần 1)
- d. Thêm dữ liệu cho bảng NhanVien sau đó thực hiện log backup (lần 2, *sử dụng lại tên file đã tạo ở lần 1*)

Câu 2: Khôi phục dữ liệu

- a. Xoá cơ sở dữ liệu QLBanHang
- b. Khôi phục dữ liệu với tì chọn With Standby thay cho With NoRecovery

GHI CHÚ:

MỘT SỐ ĐỀ ÔN TẬP THAM KHẢO

ĐỀ 1:

Câu 1: Tạo csdl QLSACH gồm 3 bảng sau:

- + Bảng nhà xuất bản: **NHAXUATBAN(MaNXB, TenNXB, SoLuongXB)**
- + Bảng tác giả: **TACGIA(MaTG, TenTG)**
- + Bảng sách: **SACH(MaSach, TenSach, NamXB, SoLuong, DonGia, MaTG, MaNXB)**
- + Nhập dữ liệu: 2 nhà xuất bản, 2 tác giả và 6 sách.

Câu 2: Tạo **View** thống kê tổng số lượng sách xuất bản theo từng nhà xuất bản, gồm các thông tin: MaNXN, TenNXB, Tổng số lượng XB.

Câu 3: Tạo 1 hàm đưa ra masach, tensach, tenTG, dongia của nhà xuất bản có tên nhập từ bàn phím với năm xb của sách từ x đến y với x, y được nhập từ bàn phím

Câu 4: Tạo thủ tục xóa sách ra khỏi bảng sách với mã sách là tham số truyền vào

-Tạo Test cho mỗi loại- Nếu không có test sẽ 0 điểm

ĐỀ 2

Câu 1: Tạo csdl QLHANG gồm 3 bảng sau:

- + Bảng danh mục hàng: **Hang(MaHang,TenHang,DVTinh,SL)**
- + Bảng hoá đơn bán hàng: **HDBan(MaHD,NgayBan,HoTenKhach)**
- + Bảng hàng bán: **HangBan(MaHD,MaHang,DonGia,SoLuong)**
- + Nhập dữ liệu: 2 Hang, 2 HDBan và 4 HangBan.

Câu 2: Tạo **View** thống kê tổng tiền hàng bán theo từng mặt hàng, gồm các thông tin: MaHang,TenHang,Tổng tiền (tiền=SoLuong*DonGia)

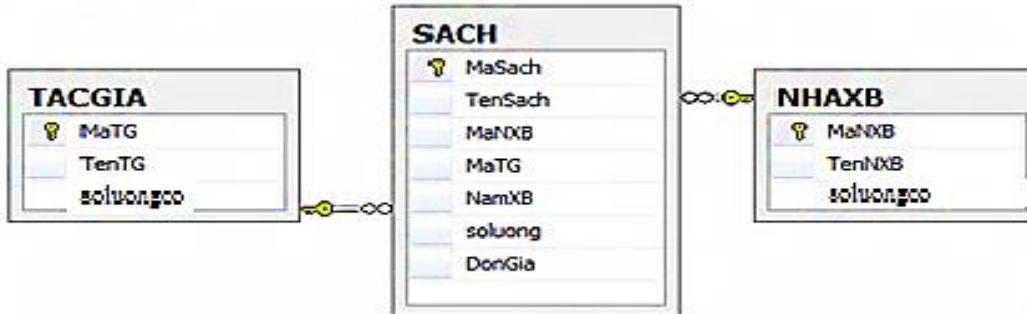
Câu 3: Tạo 1 hàm đưa ra mahang, tenhang, soluong, dongia của khách hàng có tên khách nhập từ bàn phím với ngày bán từ x đến y với x, y được nhập từ bàn phím

Câu 4: Tạo thủ tục xóa đơn hàng ra khỏi bảng HangBan với maHD nhập từ bàn phím

-Tạo Test cho mỗi loại- Nếu không có test sẽ 0

ĐỀ 3

Câu 1(2 đ): Tạo csdl QLSACH gồm 3 bảng:



Xây dựng mô hình quan hệ cho 3 bảng trên, hãy nhập 3 tacgia, 3 nxb, 4 đầu sách

Câu 2(2 đ): Hãy tạo thủ tục đưa ra **thống kê tiền bán theo tên nxb** gồm Manxb, Tênnxb, TienBan (TienBan=SoLuong*DonGia) biết Tên nxb nhập từ bàn phím (lưu ý: một tên NXB có thể xuất bản nhiều sách). Nếu không có tên nxb đưa ra thông báo

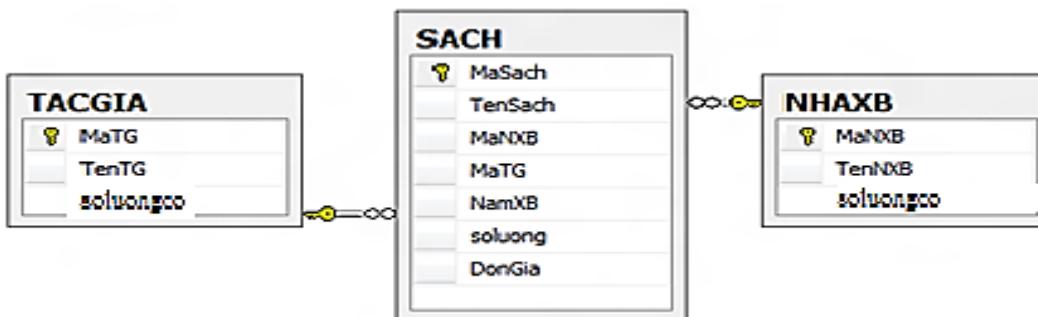
Câu 3(2 đ): Hãy tạo hàm đưa ra **thống kê tiền bán theo mã TG** biết TienBan (TienBan=SoLuong*DonGia) với tham số truyền là Tên TG (lưu ý: một tác giả có thể xuất bản nhiều lần).

Câu 4(2đ): Viết Trigger nhập cho bảng **SACH** sao cho khi nhập Manxb hợp lệ (MaNXB có ở bảng **nhaxb**) thì số lượng có ở bảng **nhaxb** được **cập nhật lại** theo số lượng nhập mới. Trái lại thì Roll Back đồng thời đưa ra thông báo cho người dùng biết “Mã nxb chưa có mặt trong bảng **nhaxb**“.

Chú ý: SV viết sẵn lệnh thực thi cuối mỗi phần code bài làm

ĐỀ 4

Câu 1(2 đ): Tạo csdl QLSACH gồm 3 bảng:



Xây dựng mô hình quan hệ cho 3 bảng trên, hãy nhập 3 tacgia, 3 nxb, 4 đầu sách

Câu 2(2đ): Hãy tạo hàm đưa ra thống kê tiền bán theo tên TG, gồm Masach, Tênsach, TenTG, TienBan (TienBan=SoLuong*DonGia) với tham số truyền là TenTG(lưu ý: một tác giả có thể xuất bản nhiều sách).

Câu 3(2đ): Hãy tạo thủ tục thống kê tiền bán theo tên TG và tên NXB, gồm Masach, Tênsach, TenTG, TienBan (TienBan=SoLuong*DonGia) với tham số truyền là TenTG và tên NXB (lưu ý: một tác giả có thể xuất bản nhiều sách). Nếu tenTG và tên nxb không có đưa ra thông báo

Câu 4(2đ): Viết Trigger nhập cho bảng **SACH** sao cho khi nhập MaTG hợp lệ (MaTG có ở bảng **tacgia**) thì số lượng có ở bảng **tacgia** được **cập nhật lại** theo số lượng nhập mới. Trái lại thì Roll Back đồng thời đưa ra thông báo cho người dùng biết “MãTG chưa có mặt trong bảng **tacgia**“.

GHI CHÚ:

TÀI LIỆU THAM KHẢO

- Sách, giáo trình chính:

- [1]. Giáo trình SQL Server – Đỗ Ngọc Sơn, Phan Văn Viên - Tài liệu lưu hành nội bộ của Trường Đại học Công nghiệp Hà Nội, 2015.
- [2]. Bài tập thực hành SQL Server – Trần Thanh Hùng – Tài liệu lưu hành nội bộ của Trường Đại học Công nghiệp Hà Nội, 2015.
- [3]. Giáo trình SQL Server 2008 – Trung tâm tin học Nhất nghệ (ebook)

- Sách, tài liệu tham khảo:

- [1]. James R, Groff, Paul N. Weinberg, SQL: The Complete Reference, McGraw Hill/Osborne, 2012.
- [2]. Diana Lorentz, SQL Reference, Oracle Corporation, 2011.
- [3]. Marcilina S. Garcia, Jamie Reding, Edward Whalen, Steve Adrien DeLuca, SQLServer 2008 Administrator's Companion, Microsoft Press, 2010.
- [4]. C. J. Date, Hugh Darwen, A Guide to the SQL Standard, Addison-Wesley Publishing, 2012.